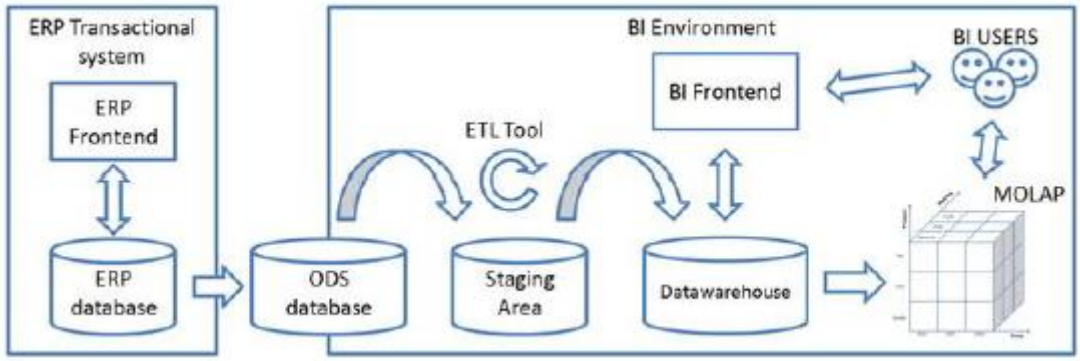
1. Business Intelligence dla każdego

W końcu biznes wymaga inteligencji. Taka jest prawda, zwłaszcza jeśli udajesz, że prowadzisz swój biznes poprawnie, ponieważ bez inteligencji nie da się prowadzić dochodowego biznesu. Mówiąc z perspektywy książki informatycznej Business Intelligence (BI) należy do świata Analytics. Business Intelligence to zestaw narzędzi i procesów, które pomagają podejmować decyzje w oparciu o dokładne dane, oszczędzając czas i wysiłek. Główną ideą narzędzia BI jest możliwość łatwej analizy danych w oparciu o koncepcje biznesowe bez posiadania wiedzy technicznej na temat narzędzi bazodanowych lub innych źródeł zawierających dane. Narzędzia BI udają, że wydobywają wiedzę z naszych przechowywanych danych w oparciu o trzy główne filary: niezawodność, dostępność i atrakcyjne doświadczenie użytkownika. Wyobraź sobie, że jesteś dyrektorem generalnym małej firmy zajmującej się produkcją ciastek i na podstawie raportów sprzedaży według analizowanych produktów stwierdzasz, że z każdym miesiącem sprzedaż ciasteczek Cream Chocolate spada. Liczby, które widzisz, stanowią obecnie około połowy kwoty, którą sprzedawano na początku roku. Jako dyrektor generalny masz różne możliwości: usuń ciastko Cream Chocolate z katalogu; zmienić formułę Cream Chocolate; ustaw premię dla swojego działu handlowego, jeśli sprzedadzą ten produkt; lub zwolnij menedżera marki, ponieważ jej dział przynosi Ci straty, co znajduje odzwierciedlenie w analizie zysków i strat. Ale co się stanie, jeśli prawdziwym problemem jest to, że ten produkt zmienił swój wewnętrzny kod z powodu zmiany formuły kremu, a katalog produktów nie jest poprawnie zaktualizowany w Twoim systemie BI - i nie odzwierciedla prawidłowo sprzedaży z nowym kodem? Twoje poprzednie decyzje byłyby błędne, ponieważ opierasz je na błędnej analizie danych. Dlatego niezawodność jest podstawowym wymogiem we wszystkich projektach IT, ale jest szczególnie istotna w narzędziach BI, ponieważ można ich używać do podejmowania głównych decyzji, od strategicznego zarządzania firmą po podstawowe działania operacyjne. Na tej podstawie obowiązkowe jest, aby dane oferowane przez nasz system BI były spójne, a każdy wymiar analizy musi zapewniać prawidłowe wyniki w oparciu o jakość danych. Teraz wyobraź sobie, że pracujesz na linii montażowej samochodu; jest piątek o 20:00 i musisz złożyć zamówienie, aby uzupełnić swój magazyn różnymi elementami przed powrotem do domu. Uruchamiasz pulpit nawigacyjny magazynu, który sugeruje ilość każdej sztuki, o którą chcesz zapytać, a źródłem tej informacji są informacje ze środy po południu, ponieważ proces codziennego odświeżania jeszcze się nie zakończył. W przyszłym tygodniu zatrzymasz się na linii montażowej z powodu brakujących kół lub będziesz miał całkowicie zalany magazyn, ponieważ poprosiłeś o 100 zderzaków, które dotarły w zeszły czwartek. Podobny powód może spowodować ten sam rezultat, jeśli nie możesz uzyskać dostępu do systemu w wymaganym czasie z powodu pewnych prac konserwacyjnych na platformie i musisz oszacować zamówienie na podstawie tego, czego Twoim zdaniem brakuje. Nasz system musi być dostępny dla naszych użytkowników, kiedy nasi użytkownicy muszą z niego korzystać. Wydaje się to oczywistym warunkiem, ale istnieją dwa główne czynniki, które mogą spowodować, że nie uda nam się osiągnąć tego celu. Nasz system musi być stabilny, działać poprawnie w godzinach pracy, a dane muszą być aktualizowane odpowiednio do naszych docelowych konsumentów i ich wymagań. Ostatnią główną cechą systemu, który zamierzamy zbudować, jest to, że nasz dostęp do dostępnych danych musi być przyjazny dla użytkownika i dostosowany do oczekiwań i możliwości konsumentów. Nie możesz zapewnić eksperckiej analizy danych za pomocą narzędzia, które nie może wchodzić w interakcję z informacjami, a z drugiej strony możesz mieć kłopoty, jeśli Twój personel handlowy, który nie ma pojęcia o komputerach, będzie musiał opublikować zapytanie SQL, aby uzyskać dane, które wymagają analizy. Aby móc zapewnić narzędzie przyjazne dla użytkownika, musisz najpierw poznać swojego użytkownika i zgodzić się z jego wymaganiami w oparciu o jego potrzeby. Również dostarczone rozwiązanie musi być dostosowane do możliwości użytkownika.

Osiągnięcie tych trzech punktów nie jest łatwym zadaniem, ale są one podstawą do dostarczenia rentownego i trwałego rozwiązania BI wewnątrz Twojej firmy lub dla Twoich klientów.

Co to jest analiza biznesowa?

Oprócz krótkiego wprowadzenia, które omówiliśmy powyżej, i biorąc pod uwagę, że będziemy mówić głównie o Business Intelligence, chcielibyśmy bardziej szczegółowo przeanalizować, jak odpowiedzieć na te dwa proste pytania: Co oznacza BI? Jak właściwie zrozumieć koncepcję BI? BI ma wiele definicji w wielu publikacjach. Definicja BI w Wikipedii jest następująca: „Inteligencja biznesowa to zestaw teorii, metodologii, architektur i technologii, które przekształcają surowe dane w znaczące i przydatne informacje do celów biznesowych". Naszym zdaniem jest to bardzo ciekawa definicja, ponieważ pokazuje pełny obraz rozwiązań BI, a nie tylko zwykłe skupienie się na narzędziach front-endowych, o których wspominają niektóre definicje. Ponieważ posiadanie rozwiązania BI oznacza podążanie za pewnymi teoriami w definiowaniu procesu, tak że określony model danych stosuje metodologie, które pomagają osiągnąć wydajność podczas projektu wdrożenia, a następnie późniejszej konserwacji, które określają poprawną architekturę, która zapewnia odpowiedni zwrot z inwestycji w oparciu o korzyści, jakie uzyskasz z projektu BI, a na koniec wybierz zestaw technologii, który spełni Twoje wymagania, specyfikacje i możliwości ekonomiczne. Na rysunku



możesz zobaczyć schemat głównych komponentów BI. Zapamiętaj to, ponieważ zrozumienie tego będzie przydatne. Na rysunku widać, że głównym źródłem informacji całego systemu jest ERP (pomimo tego, jak zobaczymy, może być wiele innych źródeł), wtedy mamy bazę danych ODS, która zawiera bezpośrednią ekstrakcję z ERP; może to być baza danych lub niektóre tabele w naszej bazie danych, ale ta koncepcja zwykle istnieje, wykorzystując bezpośrednie ekstrakcje z ERP, aby nie przeciążać systemu źródłowego. Za pomocą naszego narzędzia ETL przeniesiemy informacje z ODS do bazy danych Staging Area, gdzie będziemy je przetwarzać, a na koniec umieścimy je w hurtowni danych, do której uzyskamy dostęp za pomocą naszego narzędzia front-end BI. Jest całkiem możliwe, że mamy tylko bazę danych, a rozróżnienie między ODS, Staging Area i Datawarehouse to tylko używane przez nas tabele lub różne schematy w bazie danych. Wreszcie możemy mieć system MOLAP, który pomoże nam zrealizować budżet na kolejny rok. Zobaczymy szczegóły każdego elementu. Przeanalizujemy niektóre teorie podczas tego wprowadzenia. Istnieją również inne interesujące koncepcje definiujące BI. Jednym z nich jest skupienie się na korzyściach, jakie firma może odnieść z wdrożenia tego rodzaju systemu, o ile możesz być bardziej wydajny w zadaniach administracyjnych związanych ze zbieraniem informacji i wykorzystać swój czas na analizę informacji w celu wyciągnięcia wniosków. Należy również zauważyć, że informacje, którymi zarządzamy, mogą pochodzić ze źródeł wewnętrznych i zewnętrznych, aby móc analizować, w jaki sposób wykonujemy nasze działania, ale także porównywać się z naszymi konkurentami, jeśli publikują informacje lub próbują analizować dane z naszych rynków docelowych do którego będziemy mieli dostęp. Inną ciekawą koncepcją jest możliwość przewidywania przyszłości. Nie mówimy o czarownicach ani wróżbitach; mówimy o znalezieniu prawidłowych wzorców, które

pozwolą nam przewidzieć, jaka może być nasza sprzedaż, jeśli warunki pozostaną takie same. Co może się stać, jeśli nowy konkurent wejdzie na nasz główny rynek i ukradnie nam 20% udziału w rynku? Lub co może być wynikiem zwiększenia o 25% naszego zespołu sprzedaży o 25%? W tej koncepcji kluczową cechą jest uzyskanie umiejętności wykrywania, które zmienne są ze sobą skorelowane, a które z nich są prawie niezależne. Interaktywność jest również jednym z elementów, które mogą dać ci wyobrażenie o tym, czym jest BI. To naprawdę interesujące, że analitycy biznesowi mogą badać i przeglądać dane, aby móc odkryć te ukryte wzorce, które mogą dać ci wgląd w najbliższą przyszłość. Kolejnym elementem, o którym warto wspomnieć przy różnych definicjach BI, jest wiedza, która jest wynikiem zastosowania technik BI do dużych ilości danych przechowywanych w naszych bazach danych lub uproszczenia formuły: jeśli łączysz Dane + Analiza, otrzymujesz Wiedzę.

Ewolucja BI

Koncepcja Business Intelligence, o której mowa tu, pojawiła się po raz pierwszy i została opisana przez Howarda Dresnera w 1989 roku. Opisał Business Intelligence jako „koncepcje i metody usprawniające podejmowanie decyzji biznesowych za pomocą systemów wsparcia opartych na faktach". Od późnych lat 90. użycie tego terminu zostało uogólnione i można znaleźć niezliczone odniesienia do BI w książkach technicznych i artykułach internetowych. Pojawienie się BI jest bezpośrednio związane z konsolidacją systemów transakcyjnych na całym świecie. Zanim wszędzie zainstalowano usługi transakcyjne, narzędzia komputerowe były wykorzystywane głównie do analizy wysokiego poziomu; ilość informacji zapisywanych w systemach była na tyle mała, że można je było analizować bezpośrednio, bez potrzeby stosowania dodatkowego narzędzia. Gdy w scenariuszach biznesowych pojawiły się systemy transakcyjne, ilość danych do zarządzania wzrosła wykładniczo. Pomyśl o firmie detalicznej, która miała comiesięczne informacje o zakupionych jednostkach danego produktu i zapasie, który pozostał w sklepie; teraz ma informacje o każdym bilecie dowolnego klienta, ze szczegółowymi produktami, które zakupili. Mogą uzyskać relacje między produktami; mogą analizować metody płatności; jeśli ich klienci płacą kartami, mogą uzyskać nazwę klientów i mogą przeanalizować, ile razy dany klient odwiedza nasz sklep, jaki rodzaj produktów kupuje i wiele innych analiz. A to tylko przykład; możesz przełożyć ten przykład na swoją firmę i zrozumieć, dlaczego potrzebujesz BI we własnym biznesie.

Uwaga: większość ostatnich odniesień dotyczy specjalistycznego oprogramowania obsługującego funkcje BI, a wiele firm konsultingowych ma dedykowane zespoły i projekty BI tylko po to, by spełniać wymagania programistyczne dotyczące narzędzi BI, traktując resztę rozwiązania jako pomocnicze komponenty samo narzędzie BI. Bądź więc ostrożny, jeśli myślisz o zatrudnieniu wsparcia doradczego, upewniając się, że ich oszacowanie kosztów projektu zawiera wszystkie wymagane elementy do Twojej prośby.

Od strategii do taktyki

BI doświadczyło również pewnych zmian w zakresie swoich projektów, rozprzestrzeniając się w różnych organizacjach, od raportów najwyższego kierownictwa i pulpitów nawigacyjnych po codzienne analizy operacyjne. Korzyści płynące z BI zostały udowodnione przez menedżerów najwyższego szczebla, którzy zauważyli, że BI oferuje ich organizacjom wiele możliwości czerpania zysków z narzędzi BI wdrażających projekty BI od dołu do góry ich firm. Na przestrzeni lat możemy zauważyć, że wdrożenia odeszły od wdrożeń strategicznych, które pomagają top managerom w decyzjach, które muszą podjąć, aby właściwie kierować firmami. Obejmuje to wszystkie środowiska wewnątrz organizacji, w tym te najniższe, aby ułatwić pracownikom podejmowanie decyzji, np. które produkty

muszę poprosić o uzupełnienie do magazynu lub jaki kolor najlepiej ubrać manekina w sklepie, w którym pracuję . BI przeszło od wspomagania decyzji strategicznych do taktycznych.

Uwaga: początkowe implementacje BI były znane pod różnymi akronimami, które ujawniają charakter docelowych użytkowników tych początkowych wdrożeń. Jednym z początkowych akronimów określających tego rodzaju systemy był DSS, czyli Decisional Support System, co pokazuje nam, że naszą grupą docelową będą decydenci. Z pewnością każda osoba w naszej firmie będzie podejmować decyzje, ale najważniejsze decyzje są zwykle podejmowane przez menedżerów, liderów i dyrektorów. Innym ciekawym akronimem, który wyraża to samo, jest EIS (Executive Information System), który w tym przypadku zawiera bezpośrednio nazwiska docelowych użytkowników platformy BI: kadry kierowniczej firmy.

Big Data

Obecnie najważniejszy nurt BI odnosi się do koncepcji Big Data i Data Lake. Sam Big Data opiera się na możliwości wykorzystania narzędzi BI i możliwości analitycznych do wydobywania informacji z niewiarygodnie ogromnej ilości danych generowanych każdego dnia przez naszych pracowników, klientów i użytkowników platform na wielu różnych platformach, takich jak portale społecznościowe, ogłoszenia o pracę w sieci, fora, blogi, aplikacje i zasoby mobilne, urządzenia mobilne, informacje GPS itp., które są zapisane w nieustrukturyzowanych systemach i których nie można zaatakować za pomocą standardowego uzbrojenia hurtowni danych; a wynika to z natury hurtowni danych, co przeanalizujemy w następnych sekcjach. DWH opiera się na ustrukturyzowanej bazie danych, która zawiera jednorodne informacje ładowane do naszego systemu wykorzystujący procesy ETL, które zapewniają integralność danych, wielokrotne kontrole i walidacje; a tego rodzaju procesy są zbyt złożone, aby można je było odczytać ze źródeł Big Data, ponieważ moc obliczeniowa wymagana do przeprowadzenia tego rodzaju analizy jest zbyt wysoka. W systemie Big Data dokładność nie jest tak krytyczna jak w scenariuszu DWH. Wykorzystanie systemu Big Data do analizy logów Facebooka i pominięcie komentarza, który mógłby dostarczyć informacji od potencjalnego klienta wśród 1 miliarda użytkowników to coś, co możemy zaakceptować, ale jeśli pominiesz wiersz w swoim systemie księgowym, wygeneruje to niezgodność księgową. Big Data może służyć jako źródło zaopatrzenia naszego systemu BI: jest to dodatkowy komponent w warstwie infrastruktury i nie zastąpi naszej analizy sprzedaży, finansów ani operacji, którą możemy mieć na miejscu. Aby móc wspierać wymagania Big Data, wymagana jest nowa koncepcja inna niż DWH. W tym scenariuszu możemy zlokalizować Data Lake. Ideą tej koncepcji jest to, że nie musisz przetwarzać całej ilości dostępnych danych, aby stworzyć ustrukturyzowane źródło danych dla swojego systemu BI. Zamiast tego powinieneś uzyskać bezpośredni dostęp do swojego źródła danych, aby wyłowić potrzebne informacje (pomysł na jezioro pochodzi z wędkowania; pomysłowy, prawda?).

Internet przedmiotów

Można go traktować również jako źródło do analizy Big Data, jednak Internet of Things chciałbym omówić osobno ze względu na możliwości, jakie może on zaoferować projektom BI. Internet rzeczy jest związany z niewiarygodną ilością informacji, które można wydobyć z przychodzących urządzeń elektronicznych, które będą używane w wielu elementach na całym świecie. Teraz mamy samochody z dostępem do Internetu, lodówki, które mogą nam powiedzieć, czego brakuje w środku, maszyny sprzątające, które mogą wysłać nam SMS-a, gdy skończą, czy roboty sprzątające, które można zaprogramować do uruchamiania ze smartfona. Wyobraź sobie, ile informacji mogłoby to dostarczyć do analizy w środowiskach Big Data. Ten Internet Rzeczy zapewnia nieskończone możliwości badań i rozwoju, w których wiedza wydobyta z informacji dostarcza danych wejściowych, które będą niezwykle interesujące do analizy.

Charakterystyka BI

W ramach cech, które już skomentowaliśmy, istnieją dodatkowe i tematy związane z BI, na których chcielibyśmy skupić się bardziej szczegółowo, i które wymagają osobnych części.

Ukryte istotne informacje

Wykonując ponownie ćwiczenie wyobraźni, wyobraź sobie, że jesteś kierownikiem Sales Force w swoim kraju w swojej firmie i że masz pod swoją kontrolą pięciu przedstawicieli handlowych różnych narodowości: jeden z nich jest Chińczykiem, a drugi Włochem. Pozostali to rodowici mieszkańcy kraju. Mają przypisane regiony na podstawie liczby klientów, których muszą odwiedzić, oraz wielkości regionu w Twoim kraju. Robią regularne wizyty, aby odwiedzić wszystkich klientów z minimalną częstotliwością jednej wizyty na trzy miesiące. W tym scenariuszu przygotowujesz skonsolidowany raport dla wszystkich krajów i widzisz, że co trzy miesiące przynosi szczyt całkowitej sumy sprzedaży dla całego kraju. Wstępny wniosek, jaki możesz wyciągnąć, jest taki, że pod koniec kwartału Twój przedstawiciel handlowy sprzedaje więcej ze względu na jakiś cel, który sobie przyświeca lub że Twoi klienci mają kwartalną częstotliwość zakupów. Postanawiasz zmienić obiektywny proces na comiesięczny, aby zmotywować swoich pracowników do rozłożenia sprzedaży na cały rok, ale może to wywrzeć zbyt dużą presję i jeden z nich może zachorować. Zamiast wstępnej konkluzji i reakcji, aby potwierdzić, że podstawową przyczyną wahań sprzedaży jest okresowość, przechodzisz do analizy regionów i stwierdzasz, że są dwa regiony po pięć, które mają wyższy szczyt na wykresie, a pozostałe regionów są stabilne. Postanawiasz wyszukać przedstawiciela handlowego każdego regionu i widzisz, że są to regiony z pracownikami zagranicznymi. Porównując między nimi widać, że przez dwa miesiące w kwartale są poniżej średniej sprzedaży, ale w trzecim już nad resztą. Tak więc zmienność nie wynika tylko z okresowości, ale zależy również od pracownika. W tym momencie chcesz przeprowadzić głębszą analizę, aby zbadać, dlaczego są one bardziej zmienne w porównaniu z resztą. Następnie możesz przejrzeć miasta odwiedzane miesięcznie i docenić, że w trzecim miesiącu kwartału zawsze odwiedzają te same miasta na podstawie kwartalnej częstotliwości sprzedaży. Następnie możesz użyć analizy pochodzącej z Narodowego Instytutu Statystyki dotyczącej imigracji w poszczególnych miastach i skorelować z miastami o najwyższej sprzedaży dla tych dwóch pracowników, wykrywając, że głównym krajem pochodzenia imigracji w miastach, w których chiński pracownik ma lepsze wyniki sprzedaży, jest Chiny, a drugim krajem źródłowym w miastach, w których Włoch jest liderem sprzedaży, są Włochy. Z tymi wynikami mogłoby być interesujące przetestowanie reorganizacji przypisania klienta poprzez przypisanie, jeśli to możliwe, klienta do kogoś w Twojej organizacji z tej samej narodowości, ponieważ lepiej się rozumieją w negocjacjach, które prowadzą, zamiast robić to jak teraz przez geografia. Z tego przykładu można się dowiedzieć, że wiedza oprócz informacji może być ukryta wśród miliardów wierszy danych, a aby z tych danych wydobyć ważne informacje, trzeba wiedzieć nie tylko technicznie, jak się nimi bawić i umieć w pełni wykorzystać funkcjonalności Twojej platformy BI, ale także wymagana jest wiedza o tym, czego szukasz, w połączeniu z odrobiną intuicji.

Precyzja

Kiedy zarządzasz tysiącami, milionami lub miliardami wierszy w celu przeprowadzenia analizy, możesz pomyśleć, że jeśli stracisz kilka danych, nie wpłynie to na ogólną kwotę, ponieważ bardziej interesują Cię trendy i ewolucja danych niż posiadanie dokładnych danych do momentu trzecia liczba dziesiętna. Być może masz rację, ale bardzo ważne jest również dopasowanie danych do źródła informacji, ponieważ musisz oferować klientom niezawodność. Wyjaśnijmy przykład, aby pokazać znaczenie małych danych. W firmie produkującej produkty konsumenckie, takie jak środki czystości, masz zdefiniowany z klientami proces zarządzania zamówieniami zwrotu w przypadku produktu, który jest w złym stanie lub bezpośrednio uszkodzony. Może to spowodować małe zlecenia zwrotu kwot, które

wpływają do systemu transakcyjnego i są przekazywane do DWH. Możesz zignorować te polecenia w swoim DWH, ponieważ zarządzanie danymi głównymi jest zupełnie inne i przeprowadzając ogólną analizę, nie zauważyłbyś żadnej różnicy; pokazujesz sprzedaż brutto w wysokości 2 234 555,34 USD zamiast rzeczywistych 2 232 434,96 USD. Ale to 0,09% sprzedaży brutto, które zostało pominięte, wiąże się z wysokimi kosztami zarządzania zamówieniami zwrotów, jeśli nie weźmiesz ich pod uwagę, możesz przegapić interesujące dane dla Twojej organizacji, które mogą przełożyć się na Twoje przychody netto z dużą marżą, ponieważ niektóre zwroty od ważnych klientów wiążą się z pewnymi karami, które są dodawane do kosztów zarządzania.

Istotne KPI

Kluczowy wskaźnik wydajności (KPI) jest powiązany z miarą danych, które są przydatne do mierzenia ewolucji Twojej firmy. Określenie, które KPI muszę monitorować, aby mieć pewność, że nie przegapię żadnej istotnej miary mojej firmy, jest jednym z krytycznych kroków we wdrażaniu projektu BI. Podczas ich definiowania należy wziąć pod uwagę wiele kwestii, ale najważniejszą z nich jest trafność metryki. Oczywiście trafność miernika będzie zależała od grupy docelowej; jeśli definiujesz je ze swoim działem sprzedaży, najbardziej odpowiednie metryki będą dotyczyły sprzedaży, sprzedanych jednostek lub wolumenu, głównie sprzedaży brutto i sprzedaży netto, ponieważ jest całkiem możliwe, że te metryki są bezpośrednio powiązane do Twoich wyników i celów sprzedażowych, ale jeśli jesteś z działu finansowego, będziesz bardziej zainteresowany kosztami i przychodami netto. Jeśli pracujesz w dziale obsługi klienta, interesują Cię zwroty ze sprzedaży, jakość sprzedawanych towarów itp. Ale to nie jedyny ważny parametr, gdy chcesz mieć pokaźny zestaw KPI. W przypadku działu sprzedaży wyobraź sobie, że zostałeś przeniesiony od jednego klienta do drugiego. Twój stary klient kupował 3 miliony dolarów rocznie, a nowy kupuje 8 milionów dolarów. Czy ten numer jest dobry czy zły? Czy ta liczba sama w sobie może powiedzieć, jaka jest wydajność na tego klienta? Na pewno nie. Najlepsze analizy BI, gdy mierzysz swoją ewolucję, opierają się na wartościach procentowych; trzeba to z czymś porównać Jak te dane mają się do sprzedaży z zeszłego roku? Jak porównuje się te dane z moimi celami rocznymi? Takim rodzajem analizy interesuje się większość odbiorców informacji; chcą otrzymywać informacje, że można łatwo zinterpretować.

Na czas

Ile razy słyszałeś typowe zdanie typu „Proszę dostarczyć raport sprzedaży na mój pulpit do jutra rano przed godziną 09:00” lub „Ta informacja powinna była zostać dostarczona naszemu kierownikowi wczoraj”? Wspominaliśmy już o tym, jak ważne jest dostarczanie informacji we właściwym czasie, kiedy jest to wymagane, ale chcielibyśmy rozwinąć ten pomysł. Aby informacje były dostarczane na czas, musimy najpierw dowiedzieć się, co oznacza dla naszych klientów terminowość dla każdego zestawu informacji. W definicji zakresu projektu, oś tymczasowa jest jednym z głównych tematów do oceny w celu sprawdzenia, czy zakres projektu jest realistyczny, czy nie. Należy wziąć pod uwagę różne elementy, które mogą mieć na to wpływ, takie jak wolumetria do przeniesienia, pojemność sprzętu, złożoność procesu i zależności w zakresie ładowania informacji. W tej definicji istotne jest również rozważenie przyszłej ewolucji wszystkich tych tematów: w jaki sposób można zwiększyć wolumetrię do zarządzania i wolumetrię docelową (w zależności od strategii procesu może to zależeć nie tylko od informacji do załadowania do bazy danych, ale także od rozmiaru tabel docelowych, zwłaszcza jeśli używasz strategii aktualizacji lub scalania); jak może zwiększyć wydajność sprzętu, jeśli mamy skalowalny system lub jeśli mamy najnowocześniejszy sprzęt i nie możemy się rozwijać bez przebudowy pełnej platformy z nowymi serwerami, aby kontrolować wzrost złożoności procesów i sprawdzać, czy wcześniejsze procesy nie będą cierpieć z powodu opóźnień, które mogłyby wpłynąć na nasze procesy ETL. Wszystkie te rozważania są szczególnie istotne, jeśli mówimy o procesach dziennych lub śródziennych, ale nie są tak istotne, gdy harmonogram ETL jest tygodniowy lub miesięczny

Analiza firmy : Cykl życia i ciągłość

Poprawa

Cóż, jesteśmy w szóstym miesiącu po uruchomieniu naszego projektu BI i co miesiąc analizujemy, które są bestsellerami w naszym katalogu produktów, a które sprzedają się najgorzej. Widzimy, że podobnie jak w zeszłym miesiącu dany produkt pojawia się na dole listy i dzieje się to miesiąc po miesiącu dla tego samego produktu, ale naszym zdefiniowanym działaniem jest podążanie za ewolucją. W przyszłym miesiącu jest całkiem możliwe, że ten sam produkt pojawi się na dole listy sprzedaży. Czego brakuje w naszym systemie BI? Wydaje się jasne &helip; działanie. Musimy działać zgodnie z wynikami, które uzyskaliśmy z naszej analizy, a następnie sprawdzić wyniki naszych działań. Znajdujemy się w cyklu życia analityki firmy, gdzie nasz system BI może pomóc nam zrozumieć cały proces.

* Analizuj dane: Kiedy mówimy o analizie danych, nie możemy poprzestać na wstępnej analizie. W poprzednim przykładzie nie możemy poprzestać na określeniu, które produkty mają najgorszy wskaźnik sprzedaży, musimy zbadać, co jest tego główną przyczyną lub czy istnieje więcej niż jeden czynnik, który może wpłynąć na wyniki, które zaobserwowaliśmy. Musimy spróbować skorelować informacje z inną analizą, spróbować zobaczyć inne wizualizacje informacji, przeanalizować informacje w różnych wymiarach, próbując wyizolować, na czym polega problem tego produktu. Być może jest to kwestia złego zaprojektowania produktu, wykorzystania niewłaściwego kanału sprzedaży, zastosowanej reklamy nie jest wystarczająco dobra lub jest skierowana do niewłaściwego targetu; może przedstawiciel handlowy tego produktu ma złe wyniki dla wszystkich produktów, które ma w portfolio lub jest to produkt tymczasowy, który staramy się wyprzedać w jego głównym sezonie – próbujemy sprzedawać lody zimą w naszym lokalnym biuro na biegunie północnym.

* Zdefiniuj plan działania: Po określeniu, co jest podstawową przyczyną wykrytego problemu lub co jest naszą mocną stroną w naszym bestsellerze, musimy zdefiniować zestaw działań, które pomogą nam poprawić/ulepszyć to, co jest nie tak. Jeśli zobaczymy, że mamy w katalogu naprawdę zły produkt, zdecydujemy się usunąć go z katalogu; jeśli region nie będzie właściwy, zdecydujemy się przenieść przedstawiciela handlowego do lepszego regionu; jeśli sprzedajemy lody na biegunie północnym, lepiej zamknąć naszą firmę. Nasz plan działania musi zawierać przede wszystkim działania, odpowiedzialność i terminy.

* Działaj: Plan działania nie ma żadnej użyteczności, jeśli go nie przestrzegasz. Musimy więc postępować zgodnie z nim w ramach zaplanowanego kalendarza.

* Sprawdź wyniki: Musimy zmierzyć, w jaki sposób zmieniamy KPI, które oceniamy, aby zobaczyć, czy nasze działania przynoszą pożądane rezultaty: czy powodują odwrotny skutek niż oczekiwano lub czy moje działania nie nie mają żadnego wpływu na wynik.

Uwaga: Głównym wynikiem dobrej analizy jest dobry plan działania. Musimy spróbować dostarczyć narzędzie BI, które pozwoli Twoim analitykom łatwo opracować plan działania i zmierzyć wyniki naszych działań w celu wdrożenia planu działania.

Korzyści z BI

Korzystanie z BI może przynieść Twojej firmie wiele korzyści. Jeśli Twoje narzędzie BI zostało poprawnie wdrożone i osiągnąłeś poprawną wydajność w swoim systemie BI, uzyskasz korzyści, które klasyfikujemy jako bezpośrednie i pośrednie.

Korzyści bezpośrednie

Za bezpośrednie korzyści uważamy wszystkie te korzyści, które daje korzystanie z narzędzia BI, które są bezpośrednio związane z wdrożeniem BI. Z pewnością uzyskasz te korzyści, jeśli będziesz w stanie wdrożyć rozwiązanie BI z dobrą strategią i wydajnością. Jednym ze sposobów, aby to umożliwić, jest przestrzeganie zaleceń, które staramy się przekazać w tej książce. Jeśli zrobisz to w ten sposób, możesz uzyskać:

Optymalizacja zasobów: jeśli próbujesz wdrożyć rozwiązanie BI, jest całkiem możliwe, że obecnie Twój system analityczny jest czymś w rodzaju ponownej kompilacji różnych zespołów w Excelu, skonsolidowanej w jednym skoroszycie, z którego wyodrębniasz różne wykresy, tabele przestawne i wysokie -stosunki poziomów. Istnieje możliwość, że bardzo często znajdziesz jakiś błąd w formacie źródłowych arkuszy Excela i będziesz musiał zmodyfikować dziesiątki stron, ponieważ musisz dodać nową kolumnę w swojej analizie lub popełniłeś błąd w agregacji, co wymaga ponownego wykonania całej ekstrakcji. Rozwiązanie BI powinno bezpośrednio dostarczać informacje w wymaganym formacie; powinieneś być w stanie zapisywać raporty w formacie, którego potrzebujesz, aby otrzymywać najlepsze produkty lub porównywać z ostatnim okresem, i powinieneś być w stanie zaplanować zadanie, aby otrzymywać te raporty codziennie, co tydzień lub co miesiąc.<br

Oszczędność kosztów: wynikająca z wcześniej skomentowanej optymalizacji zasobów, uzyskasz oszczędności, szczególnie w zasobach ludzkich, które są inwestowane w bieżący proces analizy ręcznej. Projekt BI pozwoli Ci również zaoszczędzić wiele innych kosztów dzięki wdrożeniu raportowania analizy kosztów w Twoim rozwiązaniu BI, które zapewni Ci wiele punktów działania w celu optymalizacji procesów i operacji.

Uwaga: Spróbuj zmierzyć ilość czasu, jaki Twoja firma inwestuje w gromadzenie, agregowanie i formatowanie informacji, które pozwalają poznać wynik bieżącego ćwiczenia, okres zamknięcia miesiąca lub dzienną sprzedaż; zsumuj cały ten czas, mnożąc koszt jednostkowy zasobów ludzkich, a następnie porównaj go z całkowitym kosztem projektu BI. W ten sposób będziesz mógł przeanalizować bezpośredni zwrot z inwestycji (ROI), jaki uzyskasz z wdrożenia projektu BI.

SVOT: akronim od Single Version of Truth; jedną z korzyści płynących z posiadania systemu referencyjnego, który zawiera wszystkie dane wymagane do analizy, jest to, że wszyscy będą korzystać z tego samego systemu, dzięki czemu wszystkie działy będą uzyskiwać te same dane, zamiast wielu arkuszy Excel z ręcznymi ekstrakcjami, ręcznymi modyfikacjami i personalnymi rozważania. Wszystkie działy uzgodnią, jaką łączną wielkość sprzedaży osiągniesz w każdym miesiącu lub jaki był ubiegłoroczny przychód netto.

Jedna osoba odpowiedzialna za informacje: W ramach wdrażania projektu BI zdefiniujesz osobę odpowiedzialną za BI lub dział, który będzie Twoim pojedynczym punktem kontaktowym w zakresie wszystkich informacji. Nie będziesz już zależny od informacji napływających z różnych kanałów, plików Excela wypełnianych i formatowanych przez wielu pracowników, działów czy wymiany e-maili pomiędzy wszystkimi działami kontrolingu sprzedaży. Dział IT będzie centralizował proces dostarczania informacji.

Analiza samoobsługowa: jeśli jesteś informatykiem w małej firmie i odpowiadasz za wyodrębnianie i formatowanie danych dla użytkowników biznesowych, jest to jedna z najbardziej interesujących korzyści, jakie może zapewnić wdrożenie rozwiązania BI. Będą mogli samodzielnie generować raporty. Stworzysz infrastrukturę, która pozwoli im przeprowadzać wszelkie analizy, o których mogą pomyśleć, pod warunkiem posiadania dostępnych informacji, których potrzebują. Ale będą w stanie samodzielnie formatować dane, drążyć dostępne informacje, obracać dane, segmentować dane, filtrować informacje, wprowadzać dowolne modyfikacje, na które chcesz im zezwolić.

Możliwości szczegółowej analizy: Jeśli jesteś w stanie wdrożyć solidny model danych z integralnością danych, zweryfikowanymi tabelami wyszukiwania i pełnymi wymiarami, otrzymasz solidny system, który pozwoli ci analizować informacje na najbardziej szczegółowym poziomie, jaki możesz sobie wyobrazić. Z pewnością, aby móc to zrobić, musisz użyć poziomu sprzętu, który odpowiada wymaganiom dotyczącym oczekiwanego czasu odpowiedzi, ilości danych i równoległości użytkowania. Ale jeśli sprzęt jest wystarczająco wydajny, będziesz mieć możliwość dostarczenia systemu analitycznego, który zapewni zagregowaną analizę na wysokim poziomie najbardziej szczegółowych dostępnych informacji, zawsze wykorzystując funkcje BI do filtrowania ilości informacji, które system zwraca użytkownikowi.

Korzyści pośrednie

Wywodzące się z poprzedniej grupy korzyści, ale także wynikające z konfiguracji samego rozwiązania BI, będziesz mieć możliwość uzyskania innych korzyści niematerialnych, które nie są bezpośrednie, ale które można uznać za konsekwencję Twojego projektu BI. Projekt BI da Ci narzędzia do podejmowania właściwych decyzji i działania zgodnie z tymi decyzjami; jeśli to zrobisz, będziesz w stanie osiągnąć zestaw korzyści. Na poniższej liście pokazujemy tylko kilka przykładów, które uważamy za ogólnie istotne, ale może to być dla Ciebie podstawa do określenia, na czym skupiają się Twoje działania wynikające z analizy BI. Wzrost sprzedaży: Będziesz w stanie przeanalizować, którzy klienci kupują Twój produkt i przeanalizować, czy znajdziesz wśród nich wspólne wzorce, aby skoncentrować swoje strategie marketingowe na tych klientach, którzy spełniają te wzorce. Dzięki temu będziesz mieć możliwość analizy swoich produktów wiele cech produktu, które mogą dać kombinację atrybutów maksymalizujących lub minimalizujących wyniki sprzedaży, a zobaczysz, które promocje sprzedaży są bardziej skuteczne, porównując je ze sprzedażą w pozostałej części roku. Podsumowując, będziesz mieć w zasięgu ręki narzędzia, które pozwolą Ci w prosty sposób przeprowadzić złożone analizy

Redukcja kosztów: Opierając się na projekcie BI, możesz zredukować koszty swojej firmy w oparciu o wiele perspektyw, o ile posiadanie potężnego narzędzia analitycznego pomoże ci usprawnić proces kontroli kosztów. Gdy opracujesz projekt BI z wymaganymi informacjami, będziesz mógł przeanalizować wszystkie dostępne koszty w Twojej organizacji, koszty operacyjne, zasoby ludzkie, wynajem i leasingu, koszty finansowe, wydatki pracownicze, koszty wydziałowe itp.

Powiernictwo klientów : Możesz analizować reakcje klientów na kampanię marketingową. Możesz również analizować wykorzystanie kart lojalnościowych, będziesz w stanie zweryfikować ewolucję pozyskiwania klientów, a BI pozwoli Ci połączyć informacje pochodzące od Twojego operatora transakcyjnego i logistycznego, aby zweryfikować, czy dostarczają Twoje produkty na czas, wśród wielu

inna analiza, o której możesz pomyśleć.

Powinowactwo produktów: jeśli skoncentrujesz się na wymiarze produktu, będziesz w stanie przeanalizować, które produkty są ze sobą powiązane na podstawie preferencji klienta. Będziesz mógł analizować, które pary produktów kupują Twoi klienci, jakie relacje nie są oczywiste, stosując podstawowe logiczne myślenie. Można sobie wyobrazić, że jeśli klient kupuje monitor komputerowy, to całkiem możliwe, że interesuje go mysz i klawiatura, ale może trudniej jest określić, które tytuły książek lub filmy mogą mieć związek, jeśli nie jest to bezpośrednio związane z kategorią, autorem lub reżyserem.

Segmentacja klientów: Segmentacja umożliwia grupowanie klientów na podstawie ich cech. Istnieją pewne cechy bezpośrednio z nimi związane, które zwykle pochodzą z danych podstawowych klienta w Twoim systemie transakcyjnym lub CRM (moduł relacji z klientami), takie jak wiek, miasto, adres, dochód netto, liczba dzieci lub wszelkie preferencje, które możesz uzyskać od dowolnego narzędzia do

zbierania informacji. Ale możesz też zastosować segmentację opartą na metrykach, takich jak liczba nabytych produktów, jakie są jego preferowane kategorie, jaka jest jego metoda płatności, jakie są jego preferowane kanały dystrybucji, pochodząca od niego sprzedaż brutto, okres zakupu lub dowolne inne pole który wymaga wcześniejszego obliczenia wyprowadzonego z podstawowego modelu, który wdrażasz.

Analiza demograficzna : W związku z segmentacją klientów, szczególnie istotna jest analiza demografii interesującego Cię regionu. Możesz zacząć od informacji pochodzących z wewnętrznych źródeł zebranych za pomocą jakiegoś narzędzia CRM związanego z Twoimi klientami lub możesz przeanalizować informacje dostarczane przez zewnętrzne źródła, takie jak publiczne instytucje statystyczne, które publikują ogólne informacje o krajach, regionach, miastach lub okręgach. Możesz połączyć analizę demograficzną z wcześniejszą segmentacją klientów i skupić się na tych dzielnicach, w których okolica może być bardziej zainteresowana Twoim produktem. Również na podstawie demografii możesz zdecydować, gdzie otworzyć sklep lub gdzie zlokalizować punkt informacji dla klientów.

Analiza geograficzna: Można to uznać za segmentację klientów lub analizę demograficzną, ale wolimy zachować ją jako oddzielną korzyść ze względu na nowe potężne wizualizacje map, które pozwalają zlokalizować na mapie liczbę klientów, sprzedaż lub inne dane które chcesz analizować.

Analiza procesu produkcyjnego: Analizując proces produkcyjny, linię montażową lub łańcuch dystrybucji, możesz uniknąć nadprodukcji i nadwyżki zapasów, stosując techniki just-in-time za pomocą rozwiązania BI.Możesz także przeanalizować stracony czas, czas oczekiwania, wąskie gardła, ilość zasobów zużywanych przez wszystkie etapy procesu produkcyjnego lub jakie zależności można zoptymalizować/usunąć z przepływu pracy produkcyjnej. Analiza jakości: Możesz także skoncentrować swoją analizę na jakości produktów lub usług, które oferujesz, analizując liczbę incydentów według typologii, źródło incydentów według etapów linii produkcyjnej, analizuj opinie klientów lub sprawdzaj terminowość rozwiązywania i dostarczania projektów.

Analiza produktywności pracowników: Twoja firma będzie łatwiej definiować cele i analizować ich realizację, korzystając z funkcji BI, aby uzyskać dane dotyczące zrealizowanych wizyt przedstawicieli handlowych, efektywności tych wizyt, liczby zamówień na pracownika, wartości sprzedaży brutto, liczby wyprodukowanych sztuk , liczba spowodowanych usterek na pracownika, zużyte zasoby na pracownika, zrealizowane szkolenia czy dni robocze.

Cele

Możesz połączyć i połączyć wszystkie te korzyści i analizy wskaźników KPI, aby zdefiniować i realizować różne cele strategiczne dla swojej organizacji, definiując w ten sposób, która strategia poprowadzi Twoją firmę do sukcesu. Poniżej znajdziesz kilka typowych celów, ale tak jak na liście korzyści, jest to tylko zestaw przykładów; powinieneś być w stanie zdefiniować własne cele w oparciu o misję firmy.

Zoptymalizuj wydajność firmy: Zwiększając sprzedaż, zmniejszając koszty lub łącząc oba te czynniki, możesz zmaksymalizować wydajność swojej firmy. Ostatecznie dla wyników naprawdę ważny jest dochód netto, więc wszelkie wskaźniki uwzględnione w obliczeniach dochodu netto mogą rosnąć lub maleć, aby zmaksymalizować wydajność. Aby działać na podstawie któregokolwiek z tych wskaźników, możesz użyć swojego rozwiązania BI, które pomoże Ci wykryć sposób, w jaki to robisz i śledzić Twoje działania w oparciu o dowolne konkretne wskaźniki.

Wykrywanie rynków niszowych: jest to cel, który można zdefiniować w szczególności na podstawie analizy danych zewnętrznych, takich jak dane demograficzne i analizy konkurencji. Dzięki połączeniu

analizy zewnętrznej z wiedzą o tym, co Twoja firma może zaoferować rynkowi, możesz znaleźć nisze rynkowe dla konkretnego produktu, który znajduje się w Twoim katalogu lub który można opracować dla tego rynku.

Popraw zadowolenie klientów : Wydaje się dość oczywiste, że monitorując KPI, które pozwalają mierzyć jakość usług oferowanych klientom, będziesz w stanie je poprawić. Również Twoje narzędzie BI może służyć do raportowania klientom, w jaki sposób wykonujesz tę jakość usług lub inne raporty, które mogą być dla nich przydatne. Może również poprawić relacje z twoimi dostawcami, o ile możesz również dostarczać dostawcy informacji przydatnych do rozwoju jego produktów i marketingu. Przykładem tego celu może być producent odzieży, który sprzedaje swoje produkty do różnych sieci hipermarketów; jeśli hipermarket wysyła informacje o stanie sprzedaży produktów producenta, producent może ocenić, czy projektuje zgodnie z fajną linią, czy też jego produkty zostaną zwrócone, ponieważ hipermarket nie jest w stanie sprzedać żadnych sukienek.

Popraw zadowolenie pracowników: niektóre z przedstawionych wskaźników KPI i korzyści dotyczą kontroli pracowników, na przykład produktywności, sprzedaży na pracownika lub kontroli kosztów, co może powodować u pracownika poczucie, że jest on wysoce kontrolowany i monitorowany, co nie zawsze jest przyjemne dla nich. Ale z drugiej strony możesz zaoferować wiele innych korzyści płynących z BI; poprawa produktywności może dać pracownikowi czas do poświęcenia się ciekawszym zadaniom.

Skrócenie czasu poświęcanego na powtarzalne zadania, takie jak zbieranie i formatowanie informacji, pozwoli pracownikom Twojej firmy skupić się na analizie danych; wszystkie opcje oferowane przez platformę BI ułatwią im pracę; a także mogą dowiedzieć się o możliwościach narzędzi BI, temat, który poprawi ich programy nauczania.

Kto może skorzystać z BI?

Na podstawie wszystkich korzyści, które analizowaliśmy w poprzedniej sekcji, wydaje się oczywiste, że prawie każda firma może w różnym stopniu skorzystać z BI, od podstawowego operatora, który może podejmować decyzje w oparciu o przyjazny interfejs użytkownika, po wypchnięcie zdefiniowanej przycisk do dyrektora generalnego, który może zdecydować, na którym rynku inwestować w ciągu najbliższych czterech lat; w końcu każda osoba w naszej organizacji może w pewnym momencie zostać poproszona o podjęcie decyzji i może użyć BI, aby pomóc mu zdecydować, które opcje wybrać. W każdym razie podajmy kilka przykładów według działów:

* Ogólne kierownictwo: To był pierwotny zespół docelowy dla narzędzi BI, więc są oni członkami organizacji, którzy mogą czerpać większe korzyści z narzędzi BI. Dobre narzędzie BI musi być nastawione na łatwe pokazywanie istotnych informacji, na które należy zwrócić uwagę, szybkie skupienie się na celu oraz wyciąganie wniosków i podejmowanie decyzji w krótkim czasie. Podsumowując sposób na życie zespołu wykonawczego.

* Dział sprzedaży: Sprzedaż to najważniejsze zadanie w każdej firmie. Możesz wyprodukować najlepszy produkt w swojej kategorii, ale jeśli go nie sprzedasz, Twoja firma zbankrutuje. Tak więc historycznie te zespoły były drugim krokiem we wdrażaniu BI. Analizuj trendy sprzedaży, wykorzystuj BI jako wsparcie do definiowania celów sprzedażowych, a także do śledzenia, jaki jest status na rzeczywistych vs obiektywnych KPI to główne analizy, na których się skupisz.

* Zespół finansowy: Sprzedaż prawie nieskończona, ale z kosztem jednostkowym wyższym niż cena, spowodowałaby nieskończone straty. Dlatego kolejnym krokiem we wdrożeniach BI jest zwykle analiza finansowa firmy, aby skupić się na poprawie wydajności firmy.

* Dział zakupów: wywodzący się z analizy finansowej i kosztowej, a zwłaszcza dla firm produkcyjnych lub odsprzedających, jest również dość istotny w celu obniżenia kosztów nabycia surowców i towarów. Aby pomóc Ci w tym trudnym zadaniu, możesz również skorzystać z funkcji BI, ponieważ zawsze potrzebujesz go do analizy informacji w celu wyciągnięcia wniosków.

* Dział kadr: Inne ważne koszty wynikające z analizy kosztów finansowych to wynagrodzenia, diety, wydatki i inne koszty osobowe naszych pracowników. Gdy dział finansowy naciska na obniżenie kosztów zasobów ludzkich, wsparcie narzędzia BI może pomóc w analizie danych pracowników.

* Operatorzy: Każda osoba w dowolnym zespole operacyjnym w firmie może mieć możliwość podjęcia decyzji: od prośby o uzupełnienie zapasów, po zaoferowanie klientowi produktu pasującego do reszty przedmiotów, które ma w koszyku. Ponownie, posiadanie dobrego rozwiązania BI może ułatwić te zadania każdemu, kto bierze udział w procesie decyzyjnym.

Komponenty Platformy BI

W tej sekcji przeanalizujemy, jakie są główne komponenty całej platformy BI, co posłuży jako teoretyczne wprowadzenie do kolejnych rozdziałów, w których dogłębnie dowiesz się, jak zainstalować i wdrożyć rozwiązanie BI oparte na głównym oprogramowaniu open source narzędzia.

Źródło ERP: Najczęstszym źródłem informacji w rozwiązaniu BI i zwykle pierwszym, które należy wykorzystać, jest Planowanie Zasobów Przedsiębiorstwa. Tego rodzaju narzędzie transakcyjne służy do kontrolowania i zapisywania głównych operacji zachodzących wewnątrz firmy. Ten komponent sam w sobie nie może być uważany za komponent BI; ale jak na razie jest źródło projektu, więc o tym wspomnieliśmy. W zależności od włączonych funkcji ERP w Twojej firmie będziemy w stanie wydobyć główne informacje o sprzedaży, finansach, operacjach, zasobach ludzkich czy zapasach. Aby wydobyć informacje mamy dwie możliwości: dostęp przez jakiś interfejs API lub dostęp bezpośrednio do bazy danych zawierającej informacje ERP. Nasze preferowane podejście to pierwsze, jeśli narzędzie na to pozwala; w ten sposób nie będziesz tak zależny od zmiany struktury tabel w bazie danych. Interfejs komunikacyjny, który umożliwia ekstrakcję danych, prawdopodobnie będzie trzymał się ewolucji ERP, podczas gdy nikt nie zapewni, że struktura tabel pozostanie taka sama. Z drugiej strony struktura tabeli może być poufna, więc musisz zbadać, które tabele i kolumny zawierają informacje, które chcesz wyodrębnić podczas korzystania z interfejsu API, o których będziesz mieć informacje z funkcjami interfejsu i parametrami do wyodrębnienia wymaganych danych. Głównym problemem, jaki można znaleźć w użyciu API, jest wydajność, jaką oferuje ten interfejs, ponieważ wydajność ERP jest zwykle zoptymalizowana pod kątem operacji transakcyjnych o niskim wolumenie, ale uruchomienie ekstrakcji ETL może zarządzać tysiącami wierszy w jednej ekstrakcji. Będziesz musiał ocenić możliwości i wydajność obu metod ekstrakcji, aby móc zdecydować, która opcja jest najlepsza.

Baza danych

Do przechowywania danych. To jest główny cel Bazy Danych. To może być wystarczające wprowadzenie do baz danych, ale przejdźmy do rozwinięcia tego pomysłu nieco bardziej. Zapewne wiesz, czym jest baza danych, więc nie ma sensu zaczynać tutaj od definicji z Wikipedii. Skupmy się na wykorzystaniu bazy danych w BI rozwiązaniu. Baza danych to ogólna nazwa, której używamy w odniesieniu do technologii przechowywania danych, ale w przypadku środowiska BI zwykle mówimy o niej jako Datawarehouse. Z pewnością jest to podstawowy składnik tej architektury; bez bazy danych nie byłoby nic. Zawsze możesz zrobić własny ETL z procedurami ładowania, własnym rozwiązaniem do raportowania z jakąś aplikacją internetową pobierającą dane z bazy danych; ale bez bazy danych nie byłbyś w stanie nic zrobić. Głównym celem bazy danych będzie zawieranie informacji, które będą dostępne z narzędzia BI, ale także będzie zawierała informacje pomocnicze do ładowania danych, które

w zależności od narzędzi, których używamy do ETL i front-endu, będą wymagały od Ciebie zapisywanie obiektów wewnętrznych do bazy danych, która może być taka sama jak ta zawierająca dane główne lub inna. Podczas definiowania opcji konfiguracji bazy danych należy wziąć pod uwagę kilka kwestii:

* Istnieją parametry do optymalizacji dla hurtowni danych, które muszą być ustawione w inny sposób niż transakcyjna baza danych.

* Musisz określić, jaka jest wymagana dostępność dla użytkowników.

* Będziesz musiał określić, jaka jest częstotliwość ładowania i które okno ładowania jest dostępne do odświeżenia informacji.

* Będziesz musiał narysować definicję środowiska; możesz mieć pośredniczącą bazę danych pomiędzy transakcyjną a hurtownią danych, która może służyć jako ODS (Operational Data Storage).

* Będziesz musiał zdefiniować zasady tworzenia kopii zapasowych zgodnie z częstotliwością ładowania. Nie ma sensu zapisywać codziennych kopii zapasowych, jeśli dane zmieniają się raz w miesiącu.

ETL: Komponent ETL, będący akronimem słów Extraction, Transformation i Load, ma dość opisową nazwę. Jak możesz sobie wyobrazić, jego główne funkcje to Extract, Transform i Load . Co to znaczy? Nie będziesz używać swoich tabel transakcyjnych do opracowywania analizy modelu za pomocą narzędzia BI; wykorzystasz w tym celu swoją hurtownię danych. Będziesz więc musiał wyodrębnić informacje ze źródłowego systemu ERP i załadować je do hurtowni danych. W połowie tego procesu będziesz musiał dostosować informacje do wymaganej struktury hurtowni danych, tworząc tabele faktów z pożądanymi polami, tabele relacji, które muszą respektować jedną do wielu relacji między wymiarami, zapewniając, że wszystkie możliwe wartości łączenia pola są obecne w tabeli i że pola, które łączą się z resztą tabel, nie mają wartości pustych lub że tabele przeglądowe zawierają wszystkie możliwe wartości faktów i wartości tabel relacji. W ramach tych podstawowych operacji jest całkiem możliwe, że masz przekształcenia, takie jak rachunek zagregowanych tabel, proces robienia codziennego zdjęcia jakiejś tabeli, aby zachować historię jej ewolucji, obciążenie tabel bezpieczeństwa, które pozwalają na zabezpieczenie na poziomie wiersza dostępu lub wszelkie inne przekształcenia, których możesz potrzebować w procesie uzupełniania hurtowni danych. W rozdziale 5 tej książki przeanalizujemy ten proces i niektóre narzędzia open source.

Narzędzie front-end: czasami traktowane jako samo rozwiązanie BI, będziesz potrzebować rozwiązania front-end, które umożliwi użytkownikom interakcję z danymi zapisanymi w hurtowni danych. To narzędzie BI będzie głównym kanałem komunikacji między użytkownikami a danymi, dlatego zdecydowanie zaleca się, aby oba połączenia działały poprawnie. Twoi użytkownicy będą wymagać wystarczającej wiedzy na temat narzędzia BI, a narzędzie BI musi być przyjazne dla użytkownika. Z drugiej strony narzędzie BI musi być w pełni kompatybilne do pracy z Twoją bazą danych; musisz zapewnić certyfikowaną interoperacyjność między obydwoma komponentami zgodnie ze specyfikacjami dostawcy. W dalszej części tego rozdziału przeanalizujemy, jakie są główne możliwości narzędzia BI, a także w rozdziale 8 tej książki zobaczymy kilka rozwiązań open source, które można wykorzystać do rozpoczęcia projektu BI.

Narzędzie do budżetowania: Na podstawie decyzji, które podejmiesz z wykorzystaniem danych z narzędzia front-end, zdefiniujesz działania, które powinny znaleźć odzwierciedlenie w budżecie na nadchodzące okresy. Czasami tego rodzaju narzędzia nie są brane pod uwagę jako część BI, ale uważamy, że są one ważnym narzędziem, które finalizuje cykl życia analityki firmy i zawiera również funkcje analizy danych, dlatego zdecydowaliśmy się uwzględnić je w naszej definicji platformy jako komponent BI. Jako narzędzie do budżetowania możesz po prostu mieć arkusz zawierający cel na

przyszły rok, ale my przeanalizujemy bardziej zaawansowane narzędzia, które zapewnią Ci dodatkowe funkcje do tego celu, takie jak analiza WhatIf lub udostępnianie danych. Więcej informacji zobaczymy później w sekcji MOLAP

Lokalizacja platformy BI

Kiedy już wiemy, które komponenty będą częścią naszej platformy BI, musimy zdecydować, gdzie je ulokować. Aby podjąć taką decyzję, musimy zastanowić się, jaka jest polityka naszej firmy wobec pozostałych serwerów, które posiadamy; w przypadku wyboru rozwiązania licencjonowanego musimy wziąć pod uwagę różne ceny licencji dla wersji onpremise i cloud, biorąc pod uwagę zarówno licencje na system operacyjny, jak i licencje na narzędzia. Ponadto nasza firma może nałożyć pewne ograniczenia bezpieczeństwa dotyczące przesyłania poufnych danych do chmury oraz możliwe zmiany na naszej platformie. Głównie będziemy mieli trzy opcje:

On-Premise: będziesz mieć serwery zlokalizowane w Twoim CPD, w sieci firmowej i całkowicie dedykowane Tobie. Będziesz musiał zadbać o konserwację i aktualizację wersji zainstalowanego systemu operacyjnego i oprogramowania.

Chmura: będziesz korzystać ze współdzielonej infrastruktury serwerów wirtualnych kupując tylko pojemność i zapominając o utrzymaniu systemu operacyjnego i oprogramowania; zrobi to firma działająca w chmurze. Możesz mieć zawsze zaktualizowaną platformę do najnowszej wersji bez dbania o procesy aktualizacji.

Hybrydowy: Możesz myśleć w rozwiązaniu hybrydowym, umieszczając niektóre serwery w swoim CPD, a inne w chmurze. W tym przypadku najczęstszym podejściem jest posiadanie interfejsu BI w chmurze, o ile zwykłe ograniczenia bezpieczeństwa są silniejsze dla baz danych, więc być może twoja polityka bezpieczeństwa nie pozwala na posiadanie bazy danych w chmurze, ale o ile twoje Narzędzie BI front-end nie zapisuje żadnych danych, może znajdować się w dowolnym miejscu.

Koncepcje BI

W świecie BI istnieje wiele koncepcji, które mogą mieć różne nazwy w różnych narzędziach BI, ale ostatecznie odnoszą się do tego samego tematu. W tej części przyjrzymy się różnym koncepcjom, zaczynając od koncepcji hurtowni danych, poprzez model logiczny zestawu tabel z hurtowni danych, które są wykorzystywane bezpośrednio do raportowania, następnie zobaczymy model fizyczny i jego powiązania z logicznym oraz jakie elementy modelu są odwzorowywane w narzędziach BI i jak możemy je wykorzystać.

Hurtownia danych

Jedną z najważniejszych koncepcji związanych z rozwiązaniem BI jest Datawarehouse (DWH). DWH jest podstawą tradycyjnych systemów BI, ponieważ jest to miejsce, w którym znajdują się dane, które chcesz przeanalizować, zwykle obsługiwane przez system baz danych. Idea poniżej DWH polega na tym, że możesz zbierać dane z wielu źródeł, oczyszczać je, zapewniać ich integralność, zapewniać spójność i kompletność, aby mieć niezawodną implementację BI. Możesz mieć wiele źródeł ze swojej firmy, dane pochodzące z systemu transakcyjnego do standardowego przepływu Zamówienia/Dostawy/Faktury, dane pochodzące z narzędzia magazynowego, które kontroluje zapasy, które masz w środku twojego magazynu, dane pochodzące od firmy logistycznej dostarczającej Twoje produkty, ręczne grupowanie klientów lub produktów wykonane przez Twoich analityków, dane pochodzące od Twoich klientów detalicznych pokazujące, jak sprzedają Twoje produkty itp. Możesz mieć heterogeniczne dane w tych różnych źródeł, ale podstawą technik DWH jest powiązanie wszystkich tych danych w jakiś sposób z kompletnymi tabelami przeglądowymi, w których masz

wszystkie możliwe wartości dla swoich wymiarów, bez przerw w hierarchii i z unikalnymi relacjami między tabelami. Aby to osiągnąć, będziesz potrzebować wdrożenia procesów integralności, które mogą wymagać pewnych ręcznych działań wstecznych w celu poprawienia niektórych danych transakcyjnych lub głównego źródła klienta/produktu. Oczyszczanie danych jest wymagane, ponieważ dość często system transakcyjny pozwala na wprowadzanie informacji bez zbytniej kontroli. Ostatecznie jest to kwestia braku koordynacji między systemem transakcyjnym a narzędziem BI. Można to łatwo wyjaśnić na prawdziwym przykładzie, który znaleźliśmy u jednego z naszych klientów. W tej firmie dość elastycznie wykorzystują bardzo popularne narzędzie transakcyjne, w którym można je ustawić wśród wielu innych cech, jakim jest przypisany do danego klienta członek zespołu sprzedaży. Ta relacja, oparta na projektowaniu procesu przez reguły biznesowe, jest wyjątkowa; nie powinno być klienta z przypisanym więcej niż jednym pracownikiem działu sprzedaży, ale system transakcyjny nie ma tego ograniczenia – pozwala ustawić więcej niż jednego. Kiedy otrzymujemy w systemie BI osobę przypisaną do klienta, widzimy dwie różne możliwości, co może powodować powielanie danych podczas analizy zysków klientów w hierarchii sił sprzedaży. To, co zrobiliśmy, to wdrożenie procesu, który gwarantuje, że masz tylko jedno zadanie; wybieramy jeden losowo, ale informujemy osobę odpowiedzialną za zarządzanie danymi podstawowymi o poprawienie przypisania w transakcji, aby móc opublikować wiarygodną informację dla tego klienta przy kolejnym ładowaniu informacji. Inny przykład z życia wzięty, który widzieliśmy w przypadku różnych klientów reguły biznesowej, która jest ustalana przez odpowiedzialną firmę, ale czasami nie jest przestrzegana przez operatorów biznesowych, dotyczy przepływu Zamówienie-Dostawa-Faktura lub z powodu jakiegoś problemu technicznego, który może mieć problemy z integralnością. W tym przypadku główny problem, który znaleźliśmy, jest natury technicznej. U tego klienta istnieje automatyczny system zapisywania danych w bazie DWH. Za każdym razem, gdy ktoś wprowadza zamówienie, dostawę lub fakturę do systemu transakcyjnego, jest to automatycznie przesyłane do DWH. Reguła biznesowa mówi, że nie możesz mieć faktury bez powiązania z dostawą lub dostawy bez powiązania z zamówieniem. Ale jeśli z powodu jakiegoś problemu technicznego otrzymasz dostawę, ale nie otrzymałeś wcześniej powiązanego zamówienia, system BI odrzuca dostawę na stół kwarantanny, dopóki nie otrzymamy zamówienia w naszym systemie, a także ostrzega nas na koniec miesiąca o odrzuconych fakturach lub dostawach. Te dwa przykłady to tylko niektóre z niemal nieskończonych możliwości, jakie można znaleźć w naturze różnych środowisk biznesowych, które można znaleźć u swojego operatora.

Uwaga: Wdrażając system DWH, musisz dokładnie zrozumieć, jakie reguły biznesowe mają zastosowanie do Twojego modelu, i wprowadzić ograniczenia techniczne, aby wymusić realizację tych reguł, określając również działania naprawcze, które należy wykonać.

Innym pomysłem związanym z DWH jest możliwość przechowywania szerszej historii danych w stosunku do danych, do których masz dostęp w systemie źródłowym, o ile jednym z podstawowych wymagań stawianych systemowi BI jest możliwość analizy ewolucji Twoich danych w przeszłości oszacować, jaka może być przyszła ewolucja, a jeśli chcesz to zrobić konsekwentnie, będziesz potrzebować jak najwięcej informacji. DWH jest zwykle wypełniany dziennymi, tygodniowymi lub miesięcznymi procesami, więc zwykle informacje, o które można zapytać, mają pewne opóźnienie wynoszące co najmniej kilka godzin, a wynika to ze zwykłych procesów ETL, które są używane do wypełniania danych , a także ograniczenia i kontrole, które muszą wprowadzić, aby zapewnić wiarygodność danych. Istnieje również tendencja do posiadania intraday BI, ale jest to możliwe tylko w zależności od relacji między dostępnym sprzętem, optymalizacją procesów ETL i ilością danych, które chcesz uwzględnić w swoim DWH. Wewnątrz hurtowni danych znajdują się różne grupy tabel:

Tabele wejściowe: zawierają informacje bezpośrednio ze źródła danych, które próbujesz przeanalizować, czasami nazywane tabelami ODS, ponieważ zawierają informacje pobrane bezpośrednio z bazy danych ODS; Przechowywanie danych operacyjnych; który zwykle jest klonem, pełnym lub częściowym, systemu transakcyjnego, czasami nazywane są tabelami wejściowymi.

Tabele tymczasowe: Zawierają informacje tylko podczas procesu ETL, ponieważ są zwykle używane do rozwiązywania niektórych procesów obliczeniowych, których ze względu na problemy z wydajnością, złożoność lub inne przyczyny techniczne nie można rozwiązać w jednym kroku ETL. To są tabele, które będziemy uważać za obszar przejściowy. Mogą znajdować się w osobnej bazie danych lub w hurtowni danych.

Tabele końcowe: Są to zestawy tabel, które zostaną opublikowane w narzędziu BI do analizy. Ta grupa tabel będzie bezpośrednio związana z modelami logicznymi i fizycznymi, które przeanalizujemy w następnych sekcjach.

DataMart

Ideą DataMartu jest wyodrębnienie informacji z konkretnego obszaru wewnątrz firmy. Podczas gdy DWH przechowuje całe informacje, DataMart będzie zawierał informacje wydziałowe. W zależności od strategii możesz zdefiniować DataMart jako część DWH lub Twój DataMart może być umieszczony w oddzielnej bazie danych, a DWH jest poprzednim etapem Twojego DataMart. Możesz odizolować swój DataMart od reszty środowiska, umieszczając go na innym serwerze, w innej instancji bazy danych, w innej bazie danych, w innym schemacie bazy danych w tej samej bazie danych lub po prostu oddzielając go w sposób logiczny (przez nazwy tabel z przedrostkami lub sufiksami). Nie ma ogólnej rekomendacji, aby przeprowadzić to oddzielenie, będzie to zależeć od ilości danych, którymi zarządzasz w DataMart w porównaniu z całym środowiskiem, budżetu na wdrożenie DataMart, poziomu izolacji, na który możesz sobie pozwolić, oraz parametryzację, którą umożliwia implementacja bazy danych. Wszystkie poprzednie rozważania, które wyjaśniliśmy w odniesieniu do DWH, można zastosować do DataMart. Jedyna różnica polega na tym, że Datamart zawiera tylko podzbiór danych. W dalszej części tego rozdziału omówimy projekty wewnątrz obiektów BI; DataMart może być bezpośrednio powiązany z projektem.

Model logiczny

Całkiem możliwe, że jeśli jesteś częścią zespołu IT w swojej firmie i chcesz zacząć od zdefiniowania struktury DWH wymaganej do obsługi analizy BI, którą chcesz wykonać, całkiem możliwe, że zaczniesz myśleć o tabelach, pola, typy danych, klucze podstawowe, klucze obce, widoki i inne techniczne rzeczy, zaczynając od utworzenia przykładowej tabeli, która na końcu jest konwertowana na ostateczną tabelę. Ale potem możesz zobaczyć brakujące pola, nie wiesz, jak je wypełnić, jak są powiązane z tym samym polem w innych tabelach itp. Raczej nie zalecamy rozpoczynania od logicznej definicji modelu, który chcesz wdrażać w oparciu o byty i relacje między nimi, zamiast bezpośrednio zaczynać od definicji technicznej. Dzięki modelowi logicznemu możesz zobaczyć, które tabele są powiązane z innymi, jakie pola można wykorzystać do łączenia tych tabel, a także możesz łatwo sprawdzić, czy wymagania analizy biznesowej odpowiadają proponowanemu modelowi i jego głównemu celowi ; służy do interakcji i tłumaczenia tych wymagań biznesowych na strukturę bazy danych. Wewnątrz modelu logicznego, a zatem odzwierciedlonego w modelu fizycznym, możemy znaleźć głównie trzy rodzaje tabel:

Tabele faktów : są to zazwyczaj największe tabele w modelu, ponieważ zawierają dane biznesowe do analizy, podsumowania i agregacji na podstawie pożądanych pól. Zawierają szczegółowe informacje o sprzedaży, kosztach, krokach operacyjnych, ruchach księgowych, danych dotyczących zasobów ludzkich, wizytach przedstawicieli handlowych lub innych danych, które chcesz przeanalizować. Nie

jest wymagane posiadanie maksymalnego poziomu szczegółowości we wszystkich tabelach faktów, w rzeczywistości zaleca się wstępne obliczenie niektórych tabel zbiorczych, aby uzyskać lepszy czas odpowiedzi z interfejsu raportowania.

Tabele relacji: Te tabele służą do łączenia wielu pojęć między nimi. Mogą one opierać się na bezpośrednich relacjach służących do definiowania hierarchii logicznych, jako harmonogram zawierający relacje między dniem, tygodniem, miesiącem, kwartałem i rokiem, lub mogą być wykorzystywane do wiązania niezależnych pojęć, takich jak produkt i klient, w oparciu o źródłowa tabela faktów, jak np. produkt i klienci, którzy mają jakikolwiek rejestr sprzedaży w całej historii DWH.

Tabele przeglądowe : Nazywane również tabelami głównymi, zawierają głównie identyfikator pojęcia i opis tego pojęcia, a także mogą zawierać identyfikatory wyższej hierarchii w przypadku powiązanych ze sobą różnych atrybutów. W poprzednim przykładzie dotyczącym czasu można mieć powiązaną tabelę wyszukiwania dla dnia zawierającą identyfikator dnia, format daty i identyfikator miesiąca, a następnie można znaleźć tabelę miesiąca zawierającą identyfikator miesiąca, opis miesiąca i powiązany identyfikator roku. Czasami do zdefiniowania relacji można użyć tabeli przeglądowej.

Model relacyjny

Modele logiczne są oparte na modelu relacyjnym, o ile są zwykle zlokalizowane w relacyjnej bazie danych. Nazywa się relacyjnym, ponieważ podstawą modelu są relacje między danymi, zwykle zapisywane w tabelach. Tabela jest zdefiniowana przez kolumny, a każdy wiersz danych jest relacją istniejącą między różnymi polami. Definiując prosty przykład, możesz rozważyć tabelę sprzedaży, w której masz kod produktu, kod klienta i kwotę sprzedaży sprzedanej w tym miesiącu. Z tej tabeli możesz pobrać wszystkie relacje między produktem a klientem, które mają jakiś związek ze sprzedażą. Możesz także łączyć tabele za pomocą instrukcji wyboru SQL (Structured Query Language), aby łączyć informacje z więcej niż jednej tabeli i uzyskiwać relacje pochodne. Ale więcej informacji zobaczymy w rozdziale 3 poświęconym SQL. Istnieje wiele klasyfikacji modeli, ale w tej książce omówimy główne typy modeli używane w hurtowniach danych, modele znormalizowane i zdenormalizowane, modele gwiazdy i płatka śniegu.

Model znormalizowany

Model znormalizowany ma na celu zredukowanie do minimum redundancji danych, optymalizację kosztów przechowywania poprzez uniknięcie wielokrotnego powtarzania tych samych danych. Ten rodzaj modelu jest wysoce zalecany do rozwiązań transakcyjnych i może być również używany w modelach DWH, ale należy zdawać sobie sprawę z jego ograniczeń. Ponieważ będziesz mieć związek tylko raz, nie możesz pozwolić na wiele zadań. Jeśli zmienisz relację, wpłynie to na wszystkie wiersze w DWH.

Model zdenormalizowany

Model zdenormalizowany chce poprawić wydajność zapytań, unikając łączenia w czasie wykonywania. Aby to zrobić, wymaga powtarzania danych wzdłuż tabel, aby zminimalizować liczbę połączeń wymaganych do rozwiązania żądania użytkownika.

Uwaga: Najbardziej typową sytuacją w DWH jest sytuacja pośrednia między wysoką normalizacją a wysoką denormalizacją. W zależności od charakteru hierarchii i atrybutów użyjesz strategii znormalizowanej, strategii zdenormalizowanej lub strategii pośredniej.

Model gwiazdy

Model gwiezdny jest rodzajem modelu relacyjnego szeroko stosowanego w hurtowniach danych, szczególnie w przypadku małych DataMartów, gdzie masz tabele faktów (omówimy to w następnych sekcjach), które zawierają informacje o Twoich danych sprzedażowych, operacyjnych lub finansowych, a następnie mieć również kilka tabel przeglądowych połączonych z tabelą faktów przez niektóre kluczowe pola. Te tabele przeglądowe będą zawierać opisy lub inne pojęcia związane z polami kluczowymi.

Model płatka śniegu

Pomyśl teraz o płatku śniegu. Zwizualizujesz płatek z dużym rdzeniem, a następnie gałęzie, które dzielą się na mniejsze gałęzie. W podobny sposób możesz pomyśleć o modelu danych płatka śniegu, w którym pośrodku znajdują się duże tabele faktów, połączone z nimi przez niektóre kluczowe tabele, które zawierają podstawę do wyodrębnienia hierarchii głównych atrybutów, a następnie możesz znaleźć bezpośrednio połączoną tabelę wzorcową wyszukiwania do tych tabel kluczowych lub mniejszych tabel kluczowych, które wiążą dane z innymi tabelami głównymi

Model fizyczny

Po zdefiniowaniu modelu logicznego przejdziesz do spraw technicznych i będziesz musiał określić, które pola zawierają twoje tabele; które będą bezpośrednio powiązane z podmiotami modelu logicznego; jaki jest typ danych dla wszystkich tych pól; które pola będą używane do łączenia tabel; które będą unikalnymi kluczami tabel; jeśli Twój model będzie wymuszał integralność danych przy użyciu kluczy obcych lub jeśli wolisz unikać ograniczeń fizycznych w swoim modelu i opracować logiczne kontrole w celu ułatwienia procesu ETL.

Uwaga: Ważne jest zdefiniowanie nomenklatury dla tabel i pól. Ta nomenklatura może być bardzo ścisła, ale zalecamy posiadanie pośredniego poziomu nomenklatury między bardzo opisowymi nazwami a nazwami tylko technicznymi. W pełni otwarte nazwy mogą prowadzić do nazw tabel, takich jak SALES_AT_CUSTOMER_AND PRODUCT_LEVEL lub SALES_AT_CUSTOMER_AND PRODUCT_LEVEL_INCLUDING_PLANT. Jeśli spróbujesz naprawić bardzo ścisłą nomenklaturę, możesz znaleźć nazwy takie jak XP55GES001FT, które nie dają ci pojęcia, co może zawierać ta tabela. Ale z opcją pośrednią możesz naprawić, że twoje tabele faktów zaczynają się od F_, tabele relacji od R_, tabele wyszukiwania od L_, a następnie pozwalają niektórym znakom spróbować ustawić pewne znaczące nazwy, takie jak F_SALES_CUS_PRD lub F_SALES_CUS_PRD_PLT.

Jeśli chodzi o nazewnictwo pól, zaleca się przestrzeganie tych dwóch prostych zasad:

1. Nie ustawiaj tej samej nazwy pola dla różnych pojęć: zwykle narzędzia BI rozpoznają, kiedy masz tę samą nazwę pola i próbują połączyć informacje przez to pole, powodując utratę części danych. Musisz wziąć pod uwagę, że coś, co wydaje się być podobne, jak Region, może powodować niezgodność danych, jeśli mówisz o regionie klienta lub regionie pracownika, ponieważ w niektórych przypadkach mogą się one różnić.

2. Ustaw tę samą nazwę pola i typ dla tej samej koncepcji: podstawowa przyczyna jest taka sama, zwykle narzędzia BI rozpoznają to samo pole jako to samo pojęcie, ale ścisłe przestrzeganie tego zmniejszy złożoność modelu, ponieważ nie musisz zapamiętać różne nazwy pól w różnych tabelach, aby dołączyć, jeśli mówisz o tych samych informacjach.

Chcemy zwrócić uwagę na znaczenie tych dwóch pomysłów z kilkoma dodanymi prawdziwymi przykładami. Wewnątrz klienta, który analizował wizyty swoich pracowników w sklepach, w których sprzedają swoje produkty, mieliśmy informację o tym, kto odwiedził i nazwaliśmy ją EMPLOYEE_ID. Wszyscy klienci powinni mieć przydzielonego jednego pracownika, ale tylko jednego. Wydaje się jasne,

że można mieć tabelę relacji między klientem a pracownikiem, a ponieważ byliśmy wtedy bardzo oryginalni, nazwaliśmy EMPLOYEE_ID pole związane z pracownikiem, który przypisał klienta. Ale co się stało, gdy próbowaliśmy zrobić analizę wizyt na pracownika? Narzędzie wykryło, że EMPLOYEE_ID jest dostępne w obu tabelach i zinterpretowało, że są takie same, więc połączyło obie tabele przez to pole i traciliśmy informację, czy wizytę wykonał inny pracownik, czy zmieniło się przypisanie klienta.

Główne obiekty BI

W różnych narzędziach BI możemy znaleźć podobne obiekty, które mogą mieć różne nazwy, ale są powiązane z podobną ideą. W tej sekcji postaramy się przeanalizować, które to obiekty, ale powinieneś skupić się na idei stojącej za obiektem, a nie na samej nazwie obiektu.

Projekt

Jako projekt traktujemy zestaw informacji, który ma wystarczająco dużo podobieństw, aby można je było wspólnie analizować w oparciu o niektóre kluczowe pola relacji. W różnych narzędziach usłyszysz o projektach, wszechświatach, środowiskach, książkach, wdrożeniach, ale wszystkie odnoszą się do tej samej koncepcji, grupy tabel, które mają sens razem analizować, ponieważ mają jakiś związek. Koncepcja projektu w BI jest bezpośrednio powiązana z koncepcją DataMart w bazie danych, którą już zdefiniowaliśmy jako zestaw powiązanych tabel, więc przeanalizujemy w projekcie BI bazę danych DataMart. Projekty są zwykle odizolowane od siebie i często narzędzia BI nie pozwalają na transport obiektów lub łączenie informacji pochodzących z różnych projektów, więc musisz być ostrożny przy podejmowaniu decyzji, czy umieszczasz nową analizę w tym samym projekcie, czy też wolisz utwórz osobną, aby nie wymagać ponownej analizy, gdy zdasz sobie sprawę, że informacje muszą być analizowane w połączeniu z istniejącą.

Tabela

Najczęściej stosowana strategia polega na tym, że tabele bazy danych są odwzorowywane na pewnego rodzaju tabelę logiczną lub definicję tabeli, która jest własnym obiektem narzędzia BI, a następnie są wykorzystywane jako podstawa do projektowania pozostałych obiektów. Większość narzędzi BI będzie miała tego rodzaju logiczne definicje tabel wstawianych do swoich katalogów obiektów. To mapowanie może pochodzić z tabeli lub z widoku, a czasami samo narzędzie umożliwia definiowanie własnych widoków (lub selekcji) na poziomie narzędzia BI, zamiast robić to na poziomie bazy danych. Wszystkie tabele mają co najmniej jedno pole, które w kolejnych sekcjach podzielimy na dwa typy.

Fakt

Fakty to jeden z rodzajów pól, które możemy znaleźć w naszych tabelach. Są to zazwyczaj pola liczbowe, a ich główną cechą jest to, że można je agregować. Podstawą analizy są fakty i dlatego zwykle nazywane są danymi pomiarowymi. Łącząc fakt z formułą, którą chcemy zastosować do podsumowania danych (jakiś przykład zobaczymy, gdy będziemy mówić o możliwościach grupowania), otrzymamy miary, które również w zależności od narzędzia można znaleźć jako metryki, wskaźniki lub KPI. W zależności od narzędzia sam obiekt faktu nie istnieje i odwzorowujesz bezpośrednio pole za pomocą formuły, której chcesz użyć do jego agregacji. Przykładami faktów mogą być sprzedaż brutto, koszt, przychód, poświęcony czas, wielkość zapasów lub wykorzystana przestrzeń.

Wymiar

Jeśli niektóre pola są używane do agregacji, pozostałe pola podają poziom, na którym chcesz przeprowadzić tę agregację. Pola te są również nazywane danymi opisowymi, ponieważ opisują sposób dystrybucji danych i ilość analizowanych danych. Nazewnictwo różnych narzędzi to Oś, Atrybut,

Wymiar i można je pogrupować w Hierarchie, które zdefiniują relacje między tymi atrybutami lub wymiarami. W tym przypadku możemy myśleć o dacie, kliencie, produkcie, zakładzie, regionie, kraju, biurze sprzedaży lub jakimkolwiek innym polu, które pozwala podzielić informacje.

Raporty

Na końcu drogi głównym efektem działania narzędzia BI będą raporty na poziomie umożliwiającym użytkownikom końcowym analizę informacji. W zależności od narzędzia BI, ale także od charakterystyki i złożoności raportów, możemy znaleźć różne koncepcje, takie jak Raport, Dokument, Pulpit nawigacyjny, Widok, Analiza, Książka, Skoroszyt itp., ale wszystkie odnoszą się do końcowego interfejsu, który będzie służyć do analizy informacji. Mogą przejść od prostej siatki sprzedaży według kategorii do zestawu interaktywnych wizualizacji, które można przeglądać, przeglądać, obracać, drukować lub wysyłać e-mailem.

Podejścia BI

Istnieją różne podejścia BI, które zapewniają różne funkcje. Niektóre narzędzia BI łączą więcej niż jedno podejście w jedną platformę, ale zwykle nie obejmują wszystkich funkcjonalności. Musisz zdefiniować zakres swojego projektu, zanim będziesz mógł właściwie wybrać, które części rozwiązania będą potrzebne.

Zapytanie i raportowanie

Query and Reporting (Q&R) to początkowy etap typowych wdrożeń BI, o ile głównym celem systemu BI jest dostarczanie informacji w jakimś kompleksowym formacie do ich analizy. Te informacje pochodzą z bazy danych, więc aby uzyskać te informacje, użyjemy Zapytania, które jest żądaniem do hurtowni danych w celu pobrania stamtąd danych. Z drugiej strony zwracane informacje są sformatowane w formacie przyjaznym dla użytkownika, aby były czytelne dla analityków. Mówiąc o Q&R mamy na myśli nie tylko możliwość dostępu do hurtowni danych w celu wydobycia z niej informacji, ale także narzędzie, które pozwala je analizować, zagłębiać się w szczegóły, sprawdzać czy wyróżniają się liczby, sprawdzać nietypowe zachowania odbiegające od normy trendy, filtruj otrzymywane informacje i formatuj je zgodnie ze standardami firmy. Główną zaletą narzędzia Q&R jest to, że nie musisz znać języka SQL, aby móc wyodrębnić informacje z bazy danych; ten SQL jest generowany automatycznie przez narzędzie BI. W niektórych przypadkach, jeśli masz do tego umiejętności techniczne, narzędzie może dać ci możliwość modyfikowania, dostrajania lub bezpośredniego pisania od zera uruchamianego SQL, na wypadek gdyby model w narzędziu BI nie był w pełni dostosowane do poniższej bazy danych lub że potrzebujesz uzyskać informacje z innego zestawu tabel, który nie jest odwzorowany w katalogu BI. Ta możliwość będzie zależała również od używanego narzędzia BI; nie wszystkie pozwalają na tworzenie i modyfikowanie zapytań w SQL. W zależności od możliwości narzędzia BI i modelu Twojej organizacji, będą również tworzyć własne metryki, niestandardowe wymiary lub złożone filtry i grupowania. Możliwości BI przeanalizujemy w następnej sekcji. To rozwiązanie Q&R jest szczególnie przeznaczone dla analityków, którzy będą w stanie uzyskać większe korzyści z tej elastyczności tworzenia raportów ad hoc niż większość komponentów w organizacji.

Udostępnianie informacji

Analitycy w Twojej firmie będą bardzo zadowoleni z możliwości korzystania z raportów i zapytań ad hoc oraz przeprowadzania skomplikowanych analiz tworząc złożone metryki, ale zdecydowana większość pracowników w Twojej firmie będzie bardzo zadowolona, jeśli będzie miała bezpośredni dostęp do już utworzonych raportów lub uzyska swoje raporty w swoich skrzynkach pocztowych lub w

udostępnionej lokalizacji. Standardowym podejściem w środowisku BI jest posiadanie niewielkiego podzbioru pracowników z umiejętnościami analityka, którzy będą czerpać zyski z rozwiązania Q&R, ale następnie posiadanie grupy programistów (mogą to być te same grupy analityków), którzy będą tworzyć raporty do udostępniania z resztą organizacji albo za pośrednictwem tego samego narzędzia BI, aby użytkownicy mogli łączyć się z narzędziem BI w celu generowania raportów również z włączonymi funkcjami BI, albo za pośrednictwem jakiejś usługi dystrybucyjnej, takiej jak poczta e-mail lub folder udostępniony. Ta dystrybucja może być zwykle wykonana za pomocą narzędzia BI, ponieważ większość z nich pozwala na automatyczne dostarczanie wiadomości e-mail.

Pulpit nawigacyjny

Mówiąc o kokpitach BI, możesz pomyśleć o desce rozdzielczej samolotu. Można tam znaleźć wiele wskaźników pochodzących z różnych źródeł, które dają przegląd osiągów całego samolotu, od głównych silników po klapy skrzydłowe. Podobnie jak w desce rozdzielczej samolotu, deska rozdzielcza BI jest szczególnie istotna, aby mieć informacje o alertach, o ile trzeba zwrócić szczególną uwagę na te metryki, które są poza standardowym zakresem roboczym. Jak skomentowaliśmy w poprzednich sekcjach mówiących o istotności KPI, musimy skoncentrować nasze pulpity nawigacyjne na tych wskaźnikach, które są naprawdę istotne dla przeprowadzanej przez nas analizy, a także muszą być naprawdę znaczące w porównaniu z poprzednimi latami lub celami. Pulpity nawigacyjne BI mogą również oferować pewne funkcje, takie jak dynamiczne selektory, panele informacyjne, zależne siatki lub wykresy, w których kliknięcie części wykresu powoduje filtrowanie zależnej wizualizacji; przejść do szczegółowych informacji; przejdź do powiązanego pulpitu nawigacyjnego; wszystkie opcje formatu, których możesz wymagać kolorów, czcionki, stylu, obrazuy i kształtu; wiele układów informacji; podpowiedzi; lub osadzanie multimediów wśród innych funkcji.

Import danych

Jednym z głównych trendów w platformach BI jest możliwość szybkiej analizy informacji poprzez umożliwienie użytkownikowi dostępu do kanału importu własnych plików danych do interfejsu BI w celu wdrożenia szybkich dashboardów z wieloma wizualizacjami, dając użytkownikowi samoobsługę BI. Ta możliwość skraca również czas opracowywania projektów BI, które w przeszłości były dużymi projektami z długimi terminami realizacji. W zależności od narzędzia będziesz mógł importować pliki w różnych formatach (Excel, CSV, tekst), łączyć się z własnymi źródłami danych, łączyć się z interfejsami usług internetowych, takich jak Xquer, lub korzystać z plików znajdujących się na współdzielonych platformach chmurowych.

Wykrywanie danych

Ściśle powiązane z Importem danych jest podejście Data Discovery, które polega na zestawie wizualizacji specjalnie przeznaczonych do wyszukiwania trendów i wyjątków w bardzo łatwy sposób za pomocą bardzo intuicyjnego interfejsu. Ten rodzaj interfejsu jest bezpośrednią ewolucją pulpitów nawigacyjnych poprzez uproszczenie elementów sterujących i menu, ograniczenie dozwolonych opcji, ale skupienie się na mocniejszych. Być może użytkownik nie dysponuje bardzo precyzyjną siatką wyrównania, aby tworzyć idealne wykresy, ale może z łatwością powiązać jeden wykres z drugim i filtrować oba za pomocą prostych paneli filtrujących. Główną ideą Data Discovery jest umożliwienie użytkownikowi tworzenia własnych dashboardów bez znajomości narzędzi BI, tylko z bardzo intuicyjnym interfejsem z głównymi funkcjonalnościami.

MOLAP

Bazy danych wielowymiarowego przetwarzania analitycznego OnLine (MOLAP) są zorientowane na lepszą wydajność procesu zapytania. Zwykle są postrzegane jako sześcian, o ile mają wiele wymiarów. Tak, ściśle mówiąc, sześcian ma tylko trzy wymiary, ale jest to ilustracyjny sposób ich narysowania, aby odróżnić je od relacyjnych tabel relacyjnej bazy danych. Zamiast wielu tabel z faktami, relacjami i informacjami wyszukiwania, bazy danych MOLAP zawierają wszystkie informacje razem, wstępnie obliczone dla wszystkich wymiarów, na wszystkich poziomach różnych wymiarów, biorąc pod uwagę przecięcie ze wszystkimi poziomami pozostałych wymiarów. To wstępne obliczenie ma dwa główne implikacje dotyczące wydajności. Pobieranie informacji z bazy danych jest bardzo sprawne, ale z drugiej strony ładowanie informacji do bazy danych MOLAP może zająć bardzo dużo czasu; innymi słowy, jeśli chcesz mieć rozsądny czas ładowania, będziesz musiał zmoderować poziom szczegółowości swoich kostek MOLAP. Trudno znaleźć bazę danych MOLAP o takim samym, ani podobnym poziomie szczegółowości jak hurtownia danych. Bazy danych MOLAP zazwyczaj mają również możliwość zapisywania w nich danych przez użytkowników końcowych, a ta funkcja otwiera możliwość wykorzystania ich jako narzędzi do planowania i budżetowania. Dość często można znaleźć bazę danych MOLAP z wymiarem o nazwie Scenariusz lub Wersja, który może być używany do przechowywania rzeczywistych danych z bieżącego roku i prognozy na następny rok, a także z różnymi wersjami przyszłej prognozy, aby móc przeprowadzić analizę WhatIF, ponieważ na przykład, co by się stało, gdybym w przyszłym roku zwiększył sprzedaż o 15%, inwestując 10% więcej w reklamę? Teraz spróbujmy zwiększyć 5% w tym produkcie, ale 15% w innym produkcie. Teraz zobaczmy, czy obniżę koszty produkcji o 11% itd. Możesz przeprowadzić i porównać wiele symulacji różnych scenariuszy przed opublikowaniem celów firmy na następny rok. Czasami MOLAP nie jest uważany za klasyczną funkcjonalność BI, ponieważ BI miał być narzędziem analitycznym rzeczywistości bez możliwości pisania, ale obecnie uważamy, że dość ważna jest możliwość rysowania scenariuszy bawiących się różnymi zmiennymi, a kiedy już to zrobisz zdefiniowałeś, jaki jest twój cel, zamknij cykl życia firmy, próbując go osiągnąć.

Eksploracja danych

Jeśli widzieliśmy ostatnio, że MOLAP pozwala wyobrazić sobie, jaka może być przyszłość, zobaczymy, że eksploracja danych pozwoli ci ją przewidzieć. Eksploracja danych pomoże Ci to przewidzieć na podstawie możliwości wykrywania ukrytych trendów i wzorców. Jak możesz sobie wyobrazić, te możliwości predykcyjne opierają się na przeszłych informacjach i formułach prognozowania, więc chociaż możesz wykonać bardzo dobrą robotę, definiując je, istnieje możliwość, że nieoczekiwany fakt może całkowicie zmienić scenariusz, powodując zupełnie inne wyniki niż twoje planowanie. Dzięki eksploracji danych próbujesz znaleźć model, który uzasadnia relację między wejściem a wyjściem twojego procesu, dzięki czemu możesz lepiej zrozumieć swój biznes. Wyobraź sobie, że prowadzisz firmę zajmującą się parkowaniem samochodów. Aby wymodelować Twój biznes, zacznijmy od relacji pomiędzy posiadanymi parkingami a miesięczną kwotą faktury. Zaczynasz z 10 miejscami parkingowymi, a Twoja miesięczna kwota faktury to 2000 $. Po roku możesz dokupić 10 dodatkowych miejsc parkingowych, a Twoja miesięczna faktura wzrośnie do 4000 USD. Na podstawie tych danych zależność między liczbą miejsc parkingowych a miesięczną kwotą faktury wydaje się być liniowa. Decydujesz się więc na budowę pełnego parkingu budowlanego na 500 miejsc parkingowych, ale potem widzisz, że nie jesteś w stanie wynająć więcej niż 40, a także cena wynajmu tych parkingów spada. Twój model nie był poprawny, ponieważ nie wziąłeś pod uwagę gęstości zaludnienia na swoim obszarze, liczby istniejących parkingów poza twoim itp. Szczególnie interesujące jest znalezienie tych wzorców, aby zwrócić uwagę na nietypowe wartości, wyniki zerowe i rozproszenie danych, ponieważ mogą udzielić ci informacji zwrotnej na temat poprawności twojego modelu. Więc jeśli chcesz uzyskać jak najdokładniejsze wyniki, musisz być bardzo wyczerpujący podczas definiowania modelu eksploracji danych, uwzględniając jak najwięcej zmiennych, aby uzyskać model biznesowy, który pozwoli ci

przewidzieć przyszłe wyniki. Podstawowe obliczenia prognozy opierają się na dwóch głównych składnikach: poprzednich wynikach i trendzie. Następnie musisz rozpocząć analizę tego, jak różnią się one w zależności od różnych wymiarów. Zwykle możesz zacząć od czasu, aby sprawdzić, czy występuje sezonowość, cykl w ciągu miesiąca lub wiele lat. Następnie możesz spróbować sprawdzić, czy produkt wpływa w jakiś sposób ze względu na cykl życia produktu, który jest korelacją między inwestycjami w reklamę a przychodami netto, lub jakąkolwiek inną zmienną, którą chcesz wziąć pod uwagę.

Przychodzące podejścia

Trudno powiedzieć, dokąd możemy dojść w ciągu najbliższych dziesięciu lat w podejściach i opcjach BI. Jak skomentowano w sekcji Ewolucja BI, obecnie głównym celem jest opracowywanie rozwiązań Big Data, aby móc uzyskać dostęp do trendów w nieskończonej ilości informacji, które są generowane wszędzie przez każde urządzenie elektroniczne. Będzie to wymagało dostosowania narzędzi BI, aby miały wystarczającą moc do analizy tej ilości danych, a także zmiany filozofii klasycznego Datawarehouse, aby móc zarządzać taką ilością informacji bez rygorów reguł Datawarehouse.

Możliwości BI

W zależności od narzędzia, którego używasz, zobaczysz te możliwości z różnymi szczegółami, na przykład możesz znaleźć narzędzie, które pozwala drążyć dane, wystarczy dwukrotnie kliknąć nagłówek pola, inne, które pozwalają również wybrać sposób drążenia lub inne, że wiercenie jest ustawiane przez administratorów i nie można wyjść z określonych ścieżek, ale mają one głównie pewne cechy, które postaramy się przeanalizować w tej sekcji. Zaraz skończysz to nudne wprowadzenie, aby zacząć od bardziej technicznych rozdziałów, zwłaszcza od rozdziału 4 do rozdziału 10, które będą znacznie bardziej interaktywne. Daj spokój!

Drążenie

Widzieliśmy już kilka przykładów, które wykorzystują możliwości drążenia, aby dojść do pewnych wniosków, ale chcielibyśmy zdefiniować, jakie jest znaczenie drążenia w środowisku BI. Drill to możliwość poruszania się po danych w modelu w celu uzyskania bardziej szczegółowych informacji o niektórych pojedynczych wierszach. Jest to szczególnie przydatne, gdy próbujesz odkryć, co powoduje niezwykłą ilość danych, która wyróżnia się na tle pozostałych wartości, które widzisz. Kiedy analizujesz raport według pracownika i jeden z pracowników sprzedaje trzy razy więcej jednostek niż średnia lub inny pracownik sprzedaje mniej niż połowę, możesz przejść do hierarchii produktów, aby zobaczyć, co jest pierwotną przyczyną różnicy, lub przejść do szczegółów czas, aby sprawdzić, czy pracownik był poza biurem przez dwa tygodnie z powodu urlopu, lub przeszukać kraje lub regiony, aby sprawdzić, czy bestseller obejmuje jakiś zamożny obszar itp. Przykład użycia arkusza programu Excel z tabelą przestawną do zilustruj, że może to być następujący, w którym możesz drążyć z projektu RUN, aby przeanalizować, które zadania są wykonywane w tym projekcie.

Obracanie

Dane przestawne odnoszą się do możliwości przenoszenia pojęć z wierszy do kolumn i odwrotnie, przenoszenia niektórych pojęć do przestrzeni filtrów, przenoszenia niektórych pojęć w celu generowania różnych stron informacyjnych, ogólnie możliwości gry polami w celu uporządkowania danych które wyświetlasz w przyjaznym formacie. Na przykład łatwiej jest zrozumieć raport, który analizuje trendy, jeśli masz koncepcję czasu w kolumnach, jeśli masz ją w wierszach lub w przypadku wykresu liniowego, znacznie łatwiej jest zobaczyć ewolucję, jeśli czas jest na X oś. Ale może w połączeniu z drążeniem wykryjesz miesiąc, w którym wszyscy Twoi klienci mają wzrost sprzedaży, a następnie przeniesiesz atrybut czasu do obszaru filtrowania, wybierzesz miesiąc ze wzrostem i

przeniesiesz atrybut produktu ze strony do obszar wierszy, aby je porównać, przenosząc klienta do kolumn, aby zobaczyć relację między klientem a produktem w danym miesiącu.

Wizualizacje

W wielu przypadkach graficzne przedstawienie danych ułatwia prawidłowe zrozumienie danych ukrytych za zwykłą siatką informacji. Dlatego wszystkie narzędzia BI oferują graficzne wizualizacje danych, które w zależności od narzędzia mogą być bardziej zaawansowane i ewoluować niż inne. Większość narzędzi, a może wszystkie, oferuje zestaw wykresów, takich jak grafika słupkowa, grafika liniowa lub grafika powierzchniowa, wszystkie pionowe i poziome; wykresy kołowe, rozproszone lub pierścienie są również powszechne w większości narzędzi. Następnie możesz znaleźć konkretną zaawansowaną wizualizację, która jest dostarczana tylko przez niektóre narzędzia, takie jak chmura słów, mapa cieplna, wykresy relacji, wykresy kaskadowe, sunburst, ułożone słupki, wykres strumieniowy lub wiele innych wizualizacji, które można wykorzystać do lepszego zrozumienia raport.

Uwaga: niektóre narzędzia umożliwiają korzystanie z niestandardowych wizualizacji, które można opracowywać i importować samodzielnie. Jeśli masz jakieś specjalne możliwości wizualizacji lub jakikolwiek rodzaj dostosowania, powinieneś sprawdzić dokumentację swojego dostawcy, aby zobaczyć, czy zezwalają na tego rodzaju dodatkową wizualizację.

Wizualizacje map

We wszystkich możliwych opcjach wizualizacji istnieje rodzaj wizualizacji, który naszym zdaniem wymaga dla nich dodatkowej sekcji. Pochodzące z uniwersalizacji możliwości GPS w urządzeniach mobilnych, które są wykorzystywane do gromadzenia danych w połączeniu z możliwościami integracji map z platformami takimi jak Google Maps, ESRI Maps lub CartoDB wśród wielu innych opcji komercyjnych; możesz mieć w swoim narzędziu BI wizualizację mapy, która ułatwia poprawność zrozumienia, w jaki sposób dystrybuowana jest Twoja sprzedaż w całym kraju lub w jaki sposób Twoje siły sprzedaży obejmują określony obszar.

Sortowanie

Inną funkcją, której będziesz potrzebować w swojej analizie, są opcje sortowania. Jest to również oferowane przez większość narzędzi, ale mogą również występować pewne różnice w opcjach, na które pozwalają. Możesz sortować rosnąco lub malejąco, według jednego lub wielu pól, na podstawie pól numerycznych, alfanumerycznych lub dat, na podstawie wymiarów lub metryk, a także z możliwością sortowania wyświetlanych stron.

Grupowanie

Istnieją dwa rodzaje grupowania w świecie BI. Pierwsza jest podstawowa dla analizy BI; nie możesz szczegółowo przeanalizować setek milionów wierszy, więc musisz zebrać te informacje w coś czytelnego. Aby to zrobić, wszystkie narzędzia BI zapewniają możliwości grupowania przy użyciu języka SQL, który jest używany do wysyłania zapytań do bazy danych. Aby uzyskać pogrupowane informacje, możesz wybrać szczegółowe pola, które chcesz przeanalizować, i użyć funkcji agregacji, takich jak suma, średnia, maksimum, minimum, ostatnia, mediana itp. - zwykle w odniesieniu do pól liczbowych. Otrzymujesz więc wynik formuły zastosowanej na wszystkich dostępnych informacjach na pożądanym poziomie wyjściowym. Ten pierwszy typ grupowania zostanie przeanalizowany z dalszymi szczegółami w rozdziale SQL. Ale istnieje również inne grupowanie uważane za grupowanie elementów, ponieważ nie jest to grupowanie techniczne oparte na wartościach pola; zamiast tego jest to grupa tych wartości do niestandardowej agregacji. Wyjaśnijmy to na przykładzie, który pokaże ci, o czym mówimy. W swoim DWH masz pole, które klasyfikuje region twojego klienta, oraz drugie, które pokazuje, gdzie

znajduje się kraj. Z drugiej strony w twojej strukturze odpowiedzialnej za sprzedaż masz zespół odpowiedzialny za regiony północne i inny zespół odpowiedzialny za regiony południowe. Możesz mieć raport sprzedaży według kraju lub sprzedaży według regionu, ale być może chcesz pogrupować niektóre regiony, aby pokazać zagregowaną wartość ich zestawu i porównać ją z resztą, a pole kraju nie jest dla ciebie przydatne, ponieważ potrzebujesz dodatkowych szczegółów. Dzięki funkcjom grupowania możesz grupować regiony południowe i północne, aby wyświetlać informacje tylko na tym poziomie szczegółowości, ani w regionie, ani na poziomie kraju. Aby wykonać tego rodzaju grupowanie elementów, możesz znaleźć różne opcje w różnych narzędziach. Niektóre z nich umożliwiają grupowanie informacji w locie na podstawie wyników raportu za pośrednictwem tego samego interfejsu raportu, a inne narzędzia umożliwiają tworzenie atrybutów pochodnych lub wymiarów w celu wyświetlenia informacji pogrupowanych na tym poziomie podczas korzystania z tych obiektów ; czasami te narzędzia umożliwiają grupowanie informacji na podstawie wyników metryki, a także prostsze narzędzia BI będą wymagały przeniesienia tej możliwości do silnika bazy danych, więc musisz opracować dodatkowe tabele/widoki w bazie danych, aby powiązać różne elementy .

Filtracja

Kiedy Twoje dane bazowe składają się z setek milionów wierszy, setek pól w tysiącach tabel, jest więcej niż możliwe, że nie wykorzystasz tak dużej ilości informacji w jednym raporcie. Zwykle raporty są filtrowane według jednego lub kilku pól. Ogólnie rzecz biorąc, chcesz przeanalizować informacje dzienne, zamknięcie miesiąca finansowego, porównać je z tym samym miesiącem poprzedniego roku, zobaczyć ewolucję ostatnich dwóch lat, ale nie uzyskać w raporcie całej historii hurtowni danych. Będziesz filtrować według dnia, miesiąca lub roku. Możesz także wymagać filtrowania według kategorii produktów, za które odpowiadasz, usuwając z danych źródłowych informacje związane z wewnętrznym kodem klienta, którego Twój zespół używa do przydzielania bezpłatnych próbek lub ograniczając informacje do dziesięciu najlepszych klientów w sprzedaży dla każdego regionu. Ponownie, możliwości filtrowania będą się różnić w zależności od platformy. W zależności od platformy będziesz mógł zakwalifikować się na podstawie wymiaru lub atrybutu (równy, większy, niższy, pomiędzy, na liście, nie na liście, jest pusty, nie jest pusty, zawiera, zaczyna się itp.), wybrać spośród różnych wartości tego wymiaru lub przefiltruj wynik a obliczenia, które mogą być wynikiem na poziomie raportu lub na innym poziomie (możesz przefiltrować dziesięciu najlepszych klientów, aby przeanalizować ich sprzedaż, lub przefiltrować dziesięciu najlepszych klientów i przeprowadzić analizę według produktu, aby zobaczyć, które są ich ulubionymi produktami). W zależności od narzędzia BI będziesz mieć możliwość dynamicznego filtrowania w czasie wykonywania. Możesz mieć stały raport ze zmiennym filtrem, co pozwoli ci, używając tego samego raportu, wyodrębnić informacje o różnych klientach lub regionach.

Wyrażenia warunkowe

Wewnątrz wyrażeń warunkowych możemy znaleźć wiele formuł, takich jak przypadek, a jeśli połączy się je z logicznymi, takimi jak i, lub, i nie, to da to możliwość tworzenia złożonych obliczeń, które można wykorzystać do segmentacji klientów lub przypisania celów pracownikom. Wyobraź sobie, że wychodząc z Twojej transakcji, otrzymujesz informacje finansowe. Otrzymujesz różne wartości kosztów, a znak pochodzący z transakcji ma różne znaczenia w zależności od kosztów. Kiedy przetwarzasz zachętę do sprzedaży, jest to wartość ujemna, ale gdy otrzymujesz koszty operacyjne, jest to wartość dodatnia. Jeśli chcesz mieć metrykę, która sumuje wszystkie koszty, będziesz potrzebować zdefiniowania takiej formuły:

Case when cost_type = 'SI' then - Cost_value else +

cost_value

Uwaga: Aby opracować złożone obliczenia, będziesz potrzebować wyrażeń warunkowych. Oczywiście tego typu warunki można zastosować w bazie danych, ale jeśli pozwala na to Twoje narzędzie BI, będziesz mieć możliwość zrobienia tego na poziomie BI. Zawsze istnieje równowaga między złożonością rozwiązaną w bazie danych a złożonością rozwiązaną w narzędziu BI. W większości przypadków będzie to zależeć od preferencji programisty; można to również zdefiniować w wytycznych dla programistów i książce najlepszych praktyk.

Sumy częściowe

Gdy uruchamiasz raporty płaszczyzny z więcej niż jednym atrybutem lub wymiarem w obszarze podziału, ponieważ wymagasz przeprowadzenia połączonej analizy różnych koncepcji, możliwe, że chcesz poznać część szczegółów każdej kombinacji atrybutów, które są sumami częściowymi każdej wartości jednego lub więcej atrybutów. Również na przykładzie pochodzącym z Excela możesz zobaczyć, że będziesz mógł wiedzieć, jaka jest łączna liczba godzin dziennie, wliczając wszystkich współpracowników w projekcie.

Administracja

Wszystkie platformy BI muszą być zarządzane przez jakiś zespół administracyjny. Ponownie funkcjonalności i możliwości platformy będą się znacznie różnić w zależności od narzędzia. Wszystkie muszą mieć w taki czy inny sposób definicję bezpieczeństwa, zarządzanie użytkownikami, zarządzanie grupami, zarządzanie rolami, które określą, jaki dostęp jest dozwolony dla każdego użytkownika, z jakich uprawnień i funkcjonalności może korzystać oraz z jakich danych użytkownik jest pozwolono zobaczyć. Ale oprócz podstawowego zarządzania bezpieczeństwem, będziesz miał możliwość ustawienia pewnych limitów dostrajających wydajność serwera, parametrów konfiguracyjnych wpływających na sposób dostępu do bazy danych, parametrów konfiguracji webowej w przypadku narzędzia z komponentem webowym, konfiguracji klastra w przypadku wysokie wymagania wydajnościowe, integracja bezpieczeństwa z uwierzytelnianiem LDAP lub Single Sign On. Będziesz także musiał ocenić, jakie są Twoje wymagania administracyjne, aby upewnić się, że narzędzie BI, o którym myślisz, jest właściwe.

Interfejs użytkownika

Innym tematem, który należy ocenić, są interfejsy oferowane przez narzędzie BI w celu uzyskania dostępu i doświadczenia użytkownika końcowego. W celu uzyskania dostępu do informacji znajdziesz narzędzia, które oferują Twoim użytkownikom możliwość dostępu poprzez instalację klient-serwer, instalację jednostanowiskową, dostęp sieciowy do połączenia z systemem lub oferują otrzymywanie informacji z niektóre narzędzia dystrybucyjne, głównie e-mail lub współdzielony zasób sieciowy. Oczywiście istnieją narzędzia, które oferują je wszystkie i inne narzędzia, do których dostęp jest bardziej ograniczony. Ten temat powinien być również uwzględniony w twoim panelu decyzyjnym, ponieważ może to spowodować odrzucenie użycia jakiegoś narzędzia ze względu na twoje wytyczne dotyczące bezpieczeństwa lub ponieważ zwiększa złożoność projektu, na przykład w przypadku konieczności opracowania instalacji pakietu w porozumieniu z twoim Najlepsze praktyki instalacyjne.

Uwaga: Podsumowując, najlepszą strategią znalezienia najodpowiedniejszej opcji jest utworzenie panelu decyzyjnego w arkuszu lub slajdzie w celu oceny wszystkich możliwości różnych platform, które są obowiązkowe dla Twojego projektu, które warto mieć i które są bez znaczenia; ustaw ocenę dla każdej sekcji, a następnie wybierz najlepsze możliwe narzędzie, które odpowiada Twojemu budżetowi.

Wniosek

Jeśli dotarłeś tutaj, powinieneś mieć teraz ogólny zarys tego, czym jest rozwiązanie BI, jakie są jego główne komponenty i funkcjonalności, jakie są jego najważniejsze możliwości oraz główne kamienie milowe jego ewolucji; powinieneś znać również pewne pojęcia dotyczące BI i jego głównego komponentu, jakim jest hurtownia danych. Przejdźmy teraz do szczegółowej analizy, jaka jest nasza zalecana metodologia wdrażania platformy BI, i przeanalizujemy bardziej szczegółowo, jak zainstalować i wdrożyć główne części BI w Twojej organizacji. Ciesz się!

2.Metody Agile dla projektów BI

Załóżmy, że jesteś liderem projektu BI. Załóżmy również, że stosujesz typowe podejście do wdrażania projektu hurtowni danych z powiązanym narzędziem raportowania, procesem ETL odczytującym ze źródłowej transakcyjnej bazy danych i wstawiającym dane do swojego DWH. Zgodnie z typowym podejściem powinieneś zebrać specyfikacje od kluczowych użytkowników, pomyśleć o bardzo solidnym modelu danych, który służy do osiągnięcia tych specyfikacji, zainstalować wszystkie komponenty, wyodrębnić wszystkie potrzebne dane z różnych źródeł danych, zweryfikować integralność wszystkich tych danych dla wszystkich pól zdefiniuj wymagane raportowanie, a następnie będziesz mógł pokazać użytkownikowi kluczowemu wynik. Cały proces mógł trwać miesiące, a może nawet lata. Kiedy sprawdzasz z kluczowym użytkownikiem, jaki był wynik, Twój użytkownik mógł zmienić zdanie co do tego, czego potrzebuje, lub może Twój kluczowy użytkownik zmienił zdanie i ma zupełnie inne pomysły na to, co wykorzystać w swoich raportach. Możesz stracić dużo czasu i wysiłku na rozwój, nie osiągając żadnych rezultatów, a to może stanowić realne ryzyko przy tradycyjnym podejściu do zarządzania projektami hurtowni danych. Aby uniknąć tego ryzyka, możesz spróbować użyć tego, co uważamy za najlepsze metodyki Agile stosowane w projektach BI. W każdym razie, nie zawsze stosowanie metodologii Agile jest najlepszą strategią kontynuowania projektu. Tradycyjne metodologie opierają się na sekwencyjnych procesach, intensywnym planowaniu i dużych kosztach zarządzania projektami, więc mogą być przydatne, gdy masz zamkniętą definicję wymagań projektowych, jasną strategię rozwoju projektu i długoterminowe projekty do opracowania. Tłumaczymy, jak pracować z metodologiami Agile, ponieważ jesteśmy przekonani, że jeśli zamierzasz realizować projekt BI od zera, z użytkownikami, którzy nigdy nie korzystali z narzędzia BI, całkiem możliwe, że będziesz musiał wprowadzić zmiany w trwającego projektu, współpracować z użytkownikami końcowymi, aby sprawdzić, czy wynik projektu jest zgodny z ich oczekiwaniami, stworzyć wstępne rozwiązanie, a następnie dodawać nowe funkcjonalności w kolejnych wersjach. W tych warunkach użycie Agile może pomóc w realizacji projektu. Istnieje również podejście pośrednie, które łączy niektóre tradycyjne cechy z Agile, wykorzystując je do wykonywania niektórych tradycyjnych zadań. Oczywiście Agile działa również wtedy, gdy masz zamkniętą definicję wymagań projektowych, więc możesz używać Agile do rozwiązywania zarówno dynamicznych, jak i statycznych wymagań projektowych. Chcielibyśmy również zauważyć, że Agile nie jest metodologią samą w sobie, jest modelem do stworzenia Twojej metodologii; to zestaw reguł, które musisz dostosować do swojego środowiska i technologii, aby dokładnie określić parametry procesów roboczych. Ważne jest, aby powiedzieć, że być może, jeśli jesteś w małej firmie, niektóre zalecenia wewnątrz metodologii nie mają zastosowania. Istnieje pewna metodologia, która obejmuje codzienne spotkania, jeśli jesteś jedynym facetem w IT i będziesz odpowiedzialny za opracowanie całego rozwiązania, nie będziesz potrzebować wewnętrznych spotkań programistów w celu śledzenia statusu zadań, o ile będziesz być jedyną osobą na tym spotkaniu, chyba że masz wiele osobowości i tak czy inaczej w tym przypadku trudno byłoby spotkać wszystkich razem. W przypadku, gdy Twój zespół nie uzasadnia wdrażania pewnych rzeczy z metodologii, po prostu je zignoruj i weź stąd metody i działania, które uważasz za interesujące, aby poprawić swoją produktywność i odnieść sukces w swoim projekcie. Aby umożliwić Ci podjęcie decyzji, czy jesteś zainteresowany stosowaniem metodologii Agile, zróbmy krótkie wprowadzenie do tego, czym są metodologie Agile i jak można je wykorzystać do poprawy zadowolenia klienta w projekcie BI.

Uwaga: ta część to zbiór rekomendacji, które uważamy za interesujące, wyciągniętych z naszego doświadczenia z metodologiami Agile, zwłaszcza Scrum i Kanban, więc nie traktuj ich

jako czysto Agile, ponieważ to, co przeczytasz, to osobiste przemyślenia na ich temat.

Wprowadzenie do metodyk Agile

W niektórych przypadkach nazwa pojęcia, które próbujesz zdefiniować, jest na tyle znacząca, że może być autoopisowa. Opisujemy teraz jedną z takich sytuacji. Z naszego punktu widzenia główną cechą metodologii Agile jest to, że są one zwinne. Ale co to znaczy - zwinny pod względem projektów programistycznych? Zasadniczo istnieją cztery obszary, na których musi koncentrować się metodologia Agile . jeśli spojrzymy na zasady Manifestu Agile. Oto cztery zasady manifesty Agile, które doceniliśmy:

- Osoby i interakcje ponad procesami i narzędziami
- Działające oprogramowanie ponad obszerną dokumentację
- Współpraca z klientem ponad negocjacje kontraktów
- Reagowanie na zmiany zamiast podążania za planem

Z tych wartości manifestu Agile, dwie są dla nas szczególnie istotne: skupienie się na zadowoleniu klienta i dostosowywaniu się do zmian; ta ostatnia koncepcja sama w sobie jest definicją zwinności. Wewnątrz projektu Agile nie martwimy się zmianami, które mogą pojawić się w specyfikacjach, uważamy je za normalne; w rzeczywistości są zdrowe, o ile użytkownicy mogą dostosować swoje potrzeby, gdy zobaczą, co może zrobić narzędzie BI. Powinniśmy mieć w naszej firmie użytkowników o zdolnościach widzących, jeśli oczekujemy od nich, że są w stanie wiedzieć, czego chcą, nie wiedząc o możliwościach, jakie oferuje Twoje narzędzie. Aby postępować zgodnie z metodykami Agile, powinieneś skupić się na członkach zespołu i ich wiedzy, a nie na zdefiniowanym procesie rozwoju. Wolisz, aby wszyscy członkowie zespołu byli świadomi tego, co robi reszta, jak to zrobić i czy jest jakaś blokada, którą mogą rozwiązać inni członkowie zespołu, dzieląc się doświadczeniem w celu ulepszenia technik zamiast silnych zasad rozwoju lub bezużyteczne narzędzia kontrolne. Będziesz także zainteresowany posiadaniem produktu do pokazania klientom, a nie wymaganą dokumentacją. Dokumentacja to drugorzędne zadanie, które należy wykonać, ale priorytetem jest posiadanie działającego systemu bez awarii. Lepiej mieć solidny mały projekt niż niespójny duży system. Lepiej mieć kilka niezawodnych funkcji niż setki funkcji, które od czasu do czasu zawodzą, ale powinny działać, jeśli sprawdzisz w dokumentacji. Tutaj możemy przyjrzeć się koncepcji Incremental Delivery, o ile będziemy zainteresowani posiadaniem małego projektu początkowego, ale z dodawanymi funkcjonalnościami co kilka tygodni. Zależy nam na tym, aby rozwiązanie było dostępne jak najszybciej, aby uzyskać opinie klientów na początkowym etapie całego projektu, aby móc korygować nieporozumienia i zmieniać funkcjonalności w oparciu o oczekiwania klientów. Aby to osiągnąć, będziemy potrzebować ścisłej współpracy z naszymi klientami. Jak zobaczysz w kolejnych sekcjach, klient jest włączony w cykl życia każdej iteracji rozwoju, który jest podstawą rozwoju Agile. Życie to zmiana. Nic nie jest statyczne. Zwinne dostosowywanie się do zmian to główna zaleta, jaką metodologie Agile mogą zaoferować, aby osiągnąć sukces w projekcie BI. Dobrze jest mieć plan, ale dostosowanie się do zmiany da klientowi większą satysfakcję niż podążanie za planem. Ostatecznie to Twój klient będzie korzystał z Twojego systemu, a korzystanie z systemu jest głównym wskaźnikiem sukcesu Twojego projektu. Agile to także optymalizacja. Za Agile kryje się intencja optymalizacji naszych zasobów, aby zmaksymalizować postrzeganą wartość dla naszych klientów; mogą to być wewnętrzni lub zewnętrzni użytkownicy końcowi platformy BI. Postaramy się ograniczyć lub bezpośrednio uniknąć wszystkich tych zadań, które nie dodają wartości dla klienta. Z naszego doświadczenia wiemy, że wysoki odsetek funkcjonalności

wymaganych w początkowych wymaganiach dużego projektu nigdy nie jest wykorzystywany przez klientów, wiele wymiarów lub metryk nie jest używanych w raportach, wiele raportów nie jest wykonywanych po wstępnej walidacji, a także wielokrotnie wykryliśmy, że wielu żądanych użytkowników, którzy mają dostęp do platformy, nigdy nie logował się na naszych platformach. Z Agile postaramy się uniknąć wszystkich tych bezużytecznych wysiłków, aby spróbować skupić się na tych funkcjonalnościach, których naprawdę chce prawdziwy użytkownik. Wprowadzam tutaj pojęcie rzeczywistego użytkownika, ponieważ znaleźliśmy również wymagania projektowe definiowane przez informatyków, przez działy, które będą inne niż te, które będą korzystać z narzędzia lub przez menedżerów bardzo wysokiego szczebla, którzy czasami nie są świadomi potrzeby wykonywać codzienne zadania pracowników, więc rzeczywisty użytkownik musi mieć bezpośrednią wiedzę na temat tych codziennych potrzeb, aby zdefiniować projekt, który od początku będzie w stanie objąć najczęściej używane funkcjonalności. Gile to także synonim współpracy. Nie ma lidera zespołu programistów (może się to wydawać pośrednio na podstawie doświadczenia niektórych członków zespołu), ale chodzi o to, że wszystko jest omawiane, oceniane i analizowane przez cały zespół, a następnie poszczególne osoby wykonują zadania rozwojowe, ale reszta jest zrobiona współpraca z całym zespołem. Kolejną zaletą Agile, jeśli jesteś w stanie wdrożyć ją we właściwy sposób, jest to, że spróbujesz dostosować wymagania dotyczące obciążenia pracą z istniejącą grupą zadaniową, aby nie stresować swojego zespołu programistów, unikając również szczytów obciążenia pracą. Spróbuj zmniejszyć wielozadaniowość w Twoim zespole, aby umożliwić programistom skupienie się na przydzielonym zadaniu. Pomysł polega na tym, że każdy członek zespołu wykonuje po prostu szeregowane zadania, jedno po drugim.

Zwinne podejście

Istnieje wiele metodologii Agile, które są zgodne z zasadami Agile, które właśnie widzieliśmy w poprzedniej sekcji, takie jak Scrum, Kanban, Lean Software Development, Adaptive Software Development, metody Crystal Clear, Extreme Programming, Feature-Driven Development i inne, ale ta rozdział nie pretenduje do miana wyczerpującej analizy Agile, bo treści starczyłoby na jeszcze jedną czy dwie książki; więc skupimy się na dwóch głównych metodach, Scrum do zarządzania projektami i Kanban do utrzymania rozwiązania. Istnieje również mieszanka znana jako Scrumban, która ma pewne cechy obu metodologii.

Nasza rekomendowana mieszanka Scruma i Kanbana

Opierając się na zasadach Agile, istnieje kilka elementów, które naszym zdaniem powinny być wspólne dla metodologii Scrum i Kanban. Pierwszym z nich jest panel wizualizacji, o ile ważne jest, aby każdy w zespole miał dostęp do statusu każdego zadania. Panel wizualizacji jest obowiązkowy dla Kanbana ale opcjonalne dla Scruma, który uwzględnia tylko zaległości zadań, ale zdajemy sobie sprawę, że posiadanie go na fizycznej tablicy jest bardzo pozytywne. Uważamy również, że bardzo interesującą funkcją są regularne, codzienne spotkania. Są one obowiązkowe dla Scruma i opcjonalne dla Kanbana, ale i tak uważamy to za bardzo interesującą funkcję. Oba podobieństwa wynikają z pierwszej wartości z manifestu Agile, aby nadać priorytet indywidualnej ewolucji i interakcji między nimi w stosunku do narzędzi, których używamy, na końcu priorytetem jest wiedza, którą zdobywają. Spotkania muszą mieć dużą cykliczność, jeśli chodzi o przepływ wiedzy między członkami zespołu i ważne, żeby były krótkie, ale częste. Idealnym okresem jest codzienne 15-minutowe spotkanie na początku dnia i powinno to być obowiązkowe. Również w Scrumie będziesz mieć różne okresowe spotkania, jak zobaczymy w następnej sekcji. Te codzienne spotkania można doskonale przeprowadzić stojąc w kącie biura, a szczególnie interesujące jest to, że odbywają się przed drugim wspólnym elementem, panelem wizualizacyjnym. Ważne jest, aby mieć tablicę lub ścianę z uporządkowanymi wszystkimi zadaniami,

które się tam znajdują aby zobaczyć w bardzo graficzny sposób, które tematy są w toku, które z nich z backlogu są najważniejsze, aby je podnieść i czy istnieje jakakolwiek zależność między zadaniami itp. Standardowym sposobem lokalizowania zadań jest uporządkowanie według kolumn na podstawie statusu zadań i grupowania według wierszy w zależności od tematu lub obszaru. Istnieją również narzędzia, które pozwalają uporządkować wszystkie zadania, więc ta graficzna wizualizacja może być na tablicy lub przy użyciu projektora lub ekranu. Na codziennych spotkaniach cały zespół przegląda status zadań i przesuwa zadania wzdłuż kolumn, gdy stan się zwiększa. Liczba kolumn i wierszy zostanie dostosowana w zależności od potrzeb projektu, ale powinny mieć co najmniej trzy kolumny: Backlog, In Progress i Done. W Backlogu zlokalizujemy wszystkie zadania oczekujące na wykonanie, w kolumnie W toku zlokalizujemy wszystkie zadania, w które zaangażowany jest jeden z członków zespołu lub klient, a na końcu przejdziemy do Gotowe wszystkie zakończone zadania . Gdy pojawi się nowa wersja oprogramowania/projektu, czyścimy kolumnę Gotowe i zaczynamy od nowa. Przyjrzymy się szczegółowo procesowi w każdym podejściu Agile.

Tworzenie projektów w Scrumie

Scrum był jedną z pierwszych metodyk, które pojawiły się jako próby łagodzenia problemów typowego zarządzania projektami z podejściem sekwencyjnym i wysoce zorganizowanymi projektami w zmieniającym się środowisku. Została zdefiniowana w 1986 roku przez Hirotakę Takeuchi i Ikujiro Nonakę. Chcieli wdrożyć elastyczną strategię rozwoju skupiając się na zdefiniowaniu wspólnego celu, a więc skupiając się na zespole deweloperskim. Istnieje kilka koncepcji wewnątrz Scruma, niektóre z nich są wspólne z innymi podejściami, ale zgodnie z komentarzem, skupimy się na tej metodologii. Aby wyjaśnić opis metodologii, która może być bardzo abstrakcyjna, zilustrujemy cały opis metodologii na podstawie tego samego fikcyjnego przykładu, projektu opracowanego dla firmy zajmującej się sklepami z narzędziami (HSC), która chce wdrożyć system BI do analizy spostrzeżeń firmy. Jest to średnia firma z 5 sklepami i 70 pracownikami, 20 z nich pracuje w centrali, 50 z nich pracuje w sklepach, w zespole sprzedaży. Każdy sklep ma kierownika sklepu, który koordynuje wszystkie wymagane przez sklep czynności, takie jak uzupełnianie zapasów, zarządzanie kasami, koordynacja z zespołami centralnymi do zarządzania zasobami ludzkimi, inicjatywami handlowymi i ofertami itp. Dział IT składa się z 3 pracowników:

* Menedżer IT: odpowiedzialny za koordynację zadań IT, kontakt z działem zakupów w celu zakupu urządzeń technicznych i zakupów oprogramowania oraz kontakt z dostawcami zewnętrznymi w razie potrzeby.

* Technik terenowy: odpowiedzialny za konserwację i naprawę komputerów i sieci w centrali i sklepach.

* Technik danych: odpowiedzialny za administrowanie serwerami centralnymi, takimi jak system ERP, poczta e-mail, współdzielone dyski sieciowe itp.

W tej firmie użyjemy Scruma do wdrożenia systemu BI, który pomoże firmie analizować dane, zaczynając od analizy sprzedaży, ale po zakończeniu będzie chciał uwzględnić zarządzanie zapasami, dane dotyczące zamówień i dane dotyczące zasobów ludzkich.

Role

Aby zdefiniować, co kto robi, zacznijmy od zdefiniowania ról, które mamy w Scrumie. Powiążemy je również z naszą przykładową strukturą.

Właściciel Produktu

Jest to podstawowa rola ze strony klienta. Będzie odpowiedzialny za zdefiniowanie, co chcą zawrzeć w projekcie, co jest priorytetem dla każdej funkcjonalności, jakie są warunki uznania funkcjonalności za w pełni działającą i akceptację rozwoju. Będzie współpracował z zespołem programistów w celu zdefiniowania, przeanalizowania i uszczegółowienia każdego wymagania. W naszym przykładzie rolę tę przejmie kierownik Kontrolingu Sprzedaży firmy.

Mistrz Scruma

Scrum Master będzie osobą odpowiedzialną za koordynację wszystkich opracowań, upewniając się, że zespół programistów rozumie i przestrzega zasad Agile; przeszkoli zespół, aby pomóc w jego autoorganizacji; pomóc usunąć blokady i przeszkody; i staraj się unikać ryzykownych przerw, które mogą przeszkadzać zespołowi w jego głównym zadaniu. Na przykładzie będzie reprezentowany przez kierownika IT.

Zespół deweloperski

Zespół programistów będzie siłą roboczą do wykonania wszystkich działań wymaganych w rozwoju. Wycenią każde wymaganie, analizując, ile może kosztować i jak je rozwinąć. Zorganizują wewnętrznie, kto w zespole podejmie się każdego zadania, a także przeanalizują blokady, które mogą pojawić się podczas procesu rozwoju, aby spróbować ich uniknąć. Pod koniec rozwoju przeanalizują, jak ulepszyć proces rozwoju dla kolejnych wymagań. Wewnątrz tego zespołu możemy zdefiniować różne role podrzędne, takie jak architekt rozwiązania, programiści, testerzy, analitycy funkcjonalni, grafik itp. Zapewnią one wewnętrzną jakość realizowanych rozwiązań, aby uniknąć skutków w przyszłości. Idealnie byłoby, gdyby ich profil był ogólny, co oznacza, że każdy członek zespołu może wykonać dowolne zadanie, ale nie zawsze jest to możliwe. W tym przykładzie rola ta zostanie przejęta przez kierownika IT, technika danych i dwóch programistów wynajętych z firmy zewnętrznej do pomocy we wstępnym rozwoju.

Interesariusze

Jest to bardzo ogólna koncepcja, ponieważ będzie wielu interesariuszy zainteresowanych tym, co robimy. Nie są zaangażowani w proces rozwoju, ale muszą być informowani o decyzjach projektowych io postępach projektu. W naszym przykładzie za interesariuszy uznamy cały zespół zarządzający, w tym dyrektora generalnego.

Użytkownicy

Będą oni końcowymi konsumentami informacji, które dostarczamy. W naszym przykładzie liczba użytkowników będzie rosła, zaczynając od zespołu wsparcia sprzedaży i kierowników sklepów, a po dodaniu nowych funkcjonalności będziemy obejmować jako użytkowników zespół HR, zespół finansowy, magazynierów odpowiedzialnych za wszystkie sklepy oraz zespół zakupów.

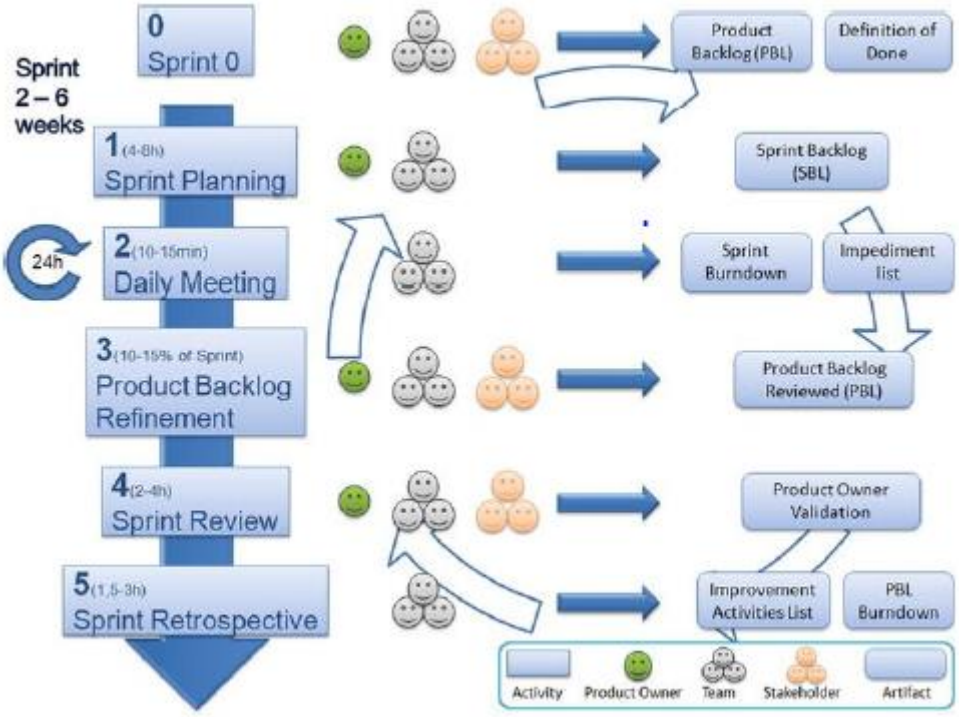
Asystent Właściciela Produktu

Ta rola jest opcjonalna; będzie to zależało od wiedzy Agile Właściciela Produktu. Będzie odpowiedzialny za ocenę Właściciela Produktu pod kątem zasad i zasad Agile. W naszym przykładzie nie rozważymy tego.

Sprint

Koncepcja sprintu lub iteracja to jedno z najważniejszych pojęć w Scrumie, także w większości metodologii Agile. Można to również potraktować jako iterację. Sprint to pojedynczy cykl rozwoju wewnątrz projektu, który zapewni w pełni użyteczną wersję projektu dostępną dla klientów do

przetestowania. Jest to okres próbowania od 2 do 6 tygodni, w zależności od wielkości zadania projektu, z zamiarem ograniczenia go do minimum. Docelowy rozmiar zadania powinien wynosić od 2 do 8 godzin obciążenia pracą, aby łatwo wykrywać zadania, które miały status „W toku" dłużej niż jeden dzień. Aby osiągnąć ten cel, jest całkiem możliwe, że musimy podzielić zadania na podzadania, aby zmieścić się w tym maksymalnym rozmiarze zadania wynoszącym 8 godzin. Wyjaśnimy, jak to zrobić w następnej sekcji Techniki segmentacji projektów. Wybierzemy wystarczającą liczbę zadań, aby uwzględnić je w sprincie, starając się, aby zespół był zajęty przez cały sprint, ale upewniając się, że zespół nie zostanie przeciążony. Sprint jest podzielony na trzy części: inicjalizacja sprintu w celu określenia, jakie zadania mają zostać uwzględnione, opracowanie tych zadań, które zapewnią następną wersję projektu oraz zakończenie sprintu w celu opublikowania wyników klientowi i oceny występ zespołu. Z tej organizacji sprintu wynikają spotkania wyjaśnione w następnej sekcji. Na rysunku



możesz zobaczyć diagram przedstawiający organizację sprintu, jakie są jego spotkania, kto powinien asystować oraz oczekiwany rezultat każdego spotkania. W przykładzie, który śledzimy wstępnie zdefiniujemy czas trwania sprintu na 4 tygodnie i postaramy się go skrócić podczas kolejnych sprintów, więc początkowo będziemy mieć siłę roboczą 144 godzin na osobę aż do pierwszego i ostatniego dnia każdego sprintu jest przeznaczona na rozpoczęcie i zakończenie sprintu. Jeśli mamy 3 osoby w zespole, będziemy mieli do dyspozycji do 432 godzin programistycznych do zarządzania w sprincie.

Konkretne spotkania Scrumowe

W odniesieniu do sprintu znajdują się cztery powiązane spotkania, które są specyficzne dla Scruma, również współdzielone z inną metodologią, a także kilka wstępnych spotkań w celu zdefiniowania zakresu projektu i wspólnego codziennego spotkania. Sprint zero meeting : Celem tego spotkania lub spotkań jest określenie zakresu projektu i jakich funkcjonalności oczekuje użytkownik, aby zdefiniować przegląd wymagań projektowych. Z tego spotkania powinniśmy wydobyć Backlog Produktu, czyli listę wymagań, które należy opracować, aby móc dostarczyć działającą wersję projektu. Ta lista wymagań musi zawierać opis każdego elementu Backlogu Produktu, czyli znaczenie tego PBI dla klienta, jaki jest koszt wdrożenia go przez zespół programistów i czy istnieje jakiekolwiek ryzyko związane z

wymaganiem. Również na tym spotkaniu powinniśmy otrzymać Definicję Ukończenia, czyli listę kontrolną warunków wynikających z umowy między klientem a deweloperem, której rezultatem jest oczekiwany wynik dla wszystkich wymagań. W spotkaniu tym powinien wziąć udział również właściciel produktu, zespół deweloperski i przynajmniej ktoś reprezentujący wszystkich interesariuszy. W naszym przykładzie zdefiniujemy na spotkaniu zerowym Sprintu, że chcemy uwzględnić następujący Backlog Produktu zawarty w Tabeli 1

Product Backlog Item	Business value (1-10)	Development cost	Risks
Sales data coming from ERP system	10	200 hours	Unknown source structure
Master data hierarchies from ERP system	9	240 hours	Lack of definition of the required levels
Customized masterdata groups	7	160 hours	Extra technology to connect
Warehouse Stock information	8	320 hours	Different masterdata values
Financial information	9	400 hours	N/A
HR information	7	320 hours	Confidential restrictions

* Spotkanie planowania sprintu: Na tym spotkaniu wybieramy umowę z Właścicielem Produktu, listę Elementów Backlogu Produktu, które będziemy rozwijać w sprincie, traktowanych również jako Backlog Sprintu. Elementy te można podzielić na zadania, aby zespół mógł zarządzać rozmiarem zadania. To spotkanie jest podzielone na dwie części: pierwsza obejmuje właściciela produktu i zespół programistów, a druga obejmuje tylko zespół programistów.

- Spotkanie „Co”: Pierwsza część tego spotkania powinna zająć od dwóch do czterech godzin, aby zdefiniować wszystkie uwzględnione PBI i uzyskać pełne zrozumienie od zespołu programistycznego, do czego odnosi się PBI, wprowadzając szczegóły dotyczące dokładnego wyniku wymagania lub funkcjonalność.

- Spotkanie „How to”: Druga część tego spotkania powinna zająć również od dwóch do czterech godzin i powinniśmy ocenić, jak spełnić wymagania właściciela produktu, które należy podzielić na zadania każdego PBI, który z zespołu deweloperskiego jest w za każde zadanie i jak długo spodziewamy się, że to zadanie może zająć. Jeśli niektóre zadania wyjściowe wymagają wyceny wymagającej dużego nakładu pracy, musimy spróbować ponownie je podzielić. Oszacowanie nakładu pracy musi być wykonane wspólnie przez cały zespół starający się osiągnąć jak największy postęp, co może wymagać rozwiązania problemów związanych z rozwiązywaniem problemów.

Mamy zaległe obciążenie szacowane na 432 godziny. Nasza wycena pracochłonności dla dwóch pierwszych zadań to 440 godzin, więc w ramach pierwszego sprintu zespół spróbuje zaimplementować pierwszy element z Backlogu Produktu i spróbuje ukończyć drugi z pomocą kierownika IT. W tym celu postaramy się podzielić każdy element na zadania, aż do osiągnięcia rozsądnej wielkości zadania. W Tabeli 2 pokazujemy tylko początkowy podział pierwszego zadania. Powinniśmy podążać za podziałem próbując zdefiniować pojedyncze podzadania z rozmiaru od 2 do 8 godzin.

Product Backlog Item	Task	Development cost	Who
Sales data coming from ERP system	Analyze information source	16 hours	Data technician
	Define database model	24 hours	Data technician
	Implement ETL process	40 hours	External 1
	Define data views	16 hours	External 1
	Create logical model	16 hours	External 2
	Define BI objects	32 hours	External 2
	Create BI reports	24 hours	External 2
	Technical validation	16 hours	Data technician
	Process automation	16 hours	Data technician

* Codzienne spotkanie: Zgodnie z komentarzem w podejściu ogólnym, zespół programistów będzie miał 15-minutowe codzienne spotkanie rano, aby skomentować bieżące zadania. Na tym spotkaniu każdy w zespole powinien odpowiedzieć sobie na trzy pytania:

- Co zrobiłeś od wczorajszego spotkania?

- Co dzisiaj robisz?

- Jakie punkty blokujące znalazłem lub mogą się pojawić?

Idea maksymalnego rozmiaru zadania wynoszącego osiem godzin polega na tym, że podczas codziennego spotkania można wykryć, czy jakieś zadanie zostało zawieszone, o ile nie powinno się wykonywać tego samego zadania na dwóch kolejnych codziennych spotkaniach. Z tego spotkania powinniśmy prowadzić listę przeszkód, które mogą się pojawić. Wracając ponownie do naszego przykładu, możemy wykryć, że nie mamy wymaganych poświadczeń, aby uzyskać dostęp do danych ERP, lub że potrzebujemy wsparcia od dostawcy ERP, aby zrozumieć strukturę tabeli. Być może jakiś składnik rozwiązania BI, taki jak narzędzie do raportowania BI, zwłaszcza w początkowej fazie rozwoju, oczekuje na dostarczenie; zabrakło nam miejsca w naszym rozwojowym systemie bazodanowym lub wielu innych przykładowych blokad, które mogą się pojawić.

* Udoskonalanie Rejestru Produktu: To spotkanie można zaplanować w dowolnym momencie sprintu, ale należy je przeprowadzić dla wszystkich sprintów, aby przejrzeć pozostały Backlog Produktu, czy są jakieś dodane zadania, które należy uwzględnić na liście, jakiś priorytet zmiana itp. Podobnie jak w przypadku spotkania zerowego sprintu, wymagana jest asysta Właściciela Produktu i zespołu deweloperskiego na spotkaniu, a oczekiwanym wynikiem jest przegląd i aktualizacja Backlogu Produktu.

* Przegląd sprintu: Po zakończeniu sprintu dokonamy wraz z właścicielem produktu i interesariuszami przeglądu wszystkich wdrożonych PBI. Da to naszemu klientowi możliwość przeglądu swoich oczekiwań, zmiany priorytetów zadań, zmiany wszystkiego, co uważa, lub dodania nowych wymagań w oparciu o rzeczywisty wynik dotyczący tego, co może zrobić nasze narzędzie. Wtedy podczas spotkania dopracowującego backlog produktu będzie mógł przełożyć naszą pracę. W przykładzie HSC możemy stwierdzić, że kierownik ds. wsparcia sprzedaży wykrył, że hierarchia produktów z transakcji nie pasuje do grup finansowych wymaganych do wyodrębnienia danych o przychodach netto, ponieważ muszą one grupować według produktów z różnymi stawkami podatkowymi i jest to nie ustawiono w wyodrębnionych przez nas tabelach ERP. Więc kiedy część sprzedażowa działa, kiedy próbujemy dodać informacje finansowe, musimy zmodyfikować obciążenie masterdat dodaj kilka dodanych pól.

* Retrospektywa sprintu: Jest to wewnętrzne spotkanie zespołu deweloperskiego, które pozwoli zespołowi przejrzeć to, co zrobiliśmy, jak rozwiązaliśmy problemy, które się pojawiły, co możemy zrobić, aby uniknąć ponownego pojawienia się problemów i jak uniknąć blokad, które może zamrozić nasz rozwój. To spotkanie pozwoli na ciągłe doskonalenie całego zespołu poprawiając produktywność każdego z nas. Wynikiem tego spotkania powinna być lista ulepszeń, które powinniśmy zastosować, aby uzyskać lepszą wydajność naszej fazy rozwoju. Z tego spotkania mogliśmy zobaczyć, że wolelibyśmy mieć rozmiar sprintu wynoszący trzy tygodnie i zmniejszyć rozmiar zadania, ponieważ chcemy mieć większą interakcję z właścicielem produktu. Lub możemy ulepszyć część metody, na przykład dodać nowy status do walidacji zadania, lub zdecydować, że spotkania będą odbywać się na projektorze zamiast fizycznej tablicy.

Wydanie

Zwykle będziemy pracować z co najmniej dwoma środowiskami: jednym do programowania i drugim do produkcji, a może trzema, dodatkowym środowiskiem, w którym użytkownik sprawdzi poprawność wykonanego rozwoju za pomocą ekstrakcji danych produkcyjnych. Po wykonaniu i zatwierdzeniu sprintu musimy promować wszystkie zmiany w środowisku produkcyjnym, aby móc czerpać korzyści z opracowanych nowych wymagań. Wiemy, że wynikiem wszystkich sprintów jest zestaw wymagań składających się na w pełni działające rozwiązanie dostarczane Właścicielowi Produktu do walidacji, ale przejście do produkcji może wymagać pewnych wysiłków, takich jak ponowna kompilacja, pobieranie klienta przez wszystkich użytkowników, ponowne ładowanie danych itp. ., i możliwe, że zdecydujemy się poczekać i zamrozić część rozwoju, dopóki nie będziemy mieli większej ilości wymagań, aby przejść do produkcji. Koncepcja wydania polega na przeniesieniu do produkcji zestawu rozwiązań, które mogą pochodzić z jednego lub wielu sprintów. Dostarczymy wersję, gdy będziemy w stanie zgromadzić wystarczającą liczbę funkcji o wysokiej wartości dla użytkowników końcowych, aby zaakceptowali zmianę swojego obecnego narzędzia. Ponownie w naszym przykładzie wprowadzimy do produkcji po opracowaniu pierwszych trzech elementów: danych sprzedaży, danych podstawowych z ERP i dostosowanych danych podstawowych, o ile jest to obowiązkowe dla kierownika ds. wsparcia sprzedaży.

Artefakty używane w Scrumie

Z objaśnienia sprintu widzieliśmy kilka narzędzi, które pomogą nam iść do przodu z naszą metodologią Scrum. Istnieją również inne narzędzia, które pomogą nam monitorować i śledzić rozwój projektu. Wszystkie wymienione i wyjaśnione narzędzia są uważane za artefakty w nomenklaturze Scruma.

Historia użytkownika

Ważne jest, aby mieć techniczną definicję tego, co jest wymagane do opracowania, ale ważne jest również, aby użytkownik wyjaśnił, co chce uzyskać, aby móc poprawnie zrozumieć, jakie są prawdziwe wymagania techniczne. Historyjka użytkownika jest krótką definicją wymagania i oczekuje się, że zostanie napisana zgodnie z szablonem historii użytkownika, który obejmuje rolę osoby żądającej, jakie jest wymaganie i jakie są korzyści z jego wdrożenia. Przykładem szablonu może być to:

As < role >,

I want < requirement >

so that < benefit >

Przechodząc ponownie do naszego przykładu firmy sprzętowej, moglibyśmy postrzegać je jako historie użytkowników:

Jako właściciel produktu chcę mieć możliwość drążenia hierarchii klientów, aby użytkownicy mogli analizować szczegóły sprzedaży poprzez organizację klientów ERP. Jako HR Manager chcę mieć dostępne dane o wynagrodzeniach i premiach, aby nasz dział mógł analizować informacje w podziale na hierarchię firmy, analizując koszty HR, urlopy i szkolenia. Jako kierownik finansowy chcę mieć informacje o zyskach i stratach, aby nasz dział mógł analizować PNL poprzez główną hierarchię produktów. Jako Sales Force Manager chcę analizować wyniki sprzedaży w poszczególnych sklepach, aby móc analizować sprzedaż według kategorii, grup klientów i kampanii promocyjnych.

Właściciel produktu i zespół programistów przeanalizują i posortują według priorytetów wszystkie te historie użytkowników, włączając je do Backlogu Produktu, a następnie zespół programistów przeanalizuje, jak technicznie postępować, aby spełnić to wymaganie. Istnieje procedura o nazwie INVEST, która sprawdza, czy historia użytkownika jest wystarczająco dobra, aby potraktować ją jako punkt wyjścia do rozwoju. Ta procedura odpowiada inicjałom słów:

Niezależne: Powinniśmy być w stanie opracować historyjkę użytkownika w oderwaniu od reszty oczekujących historyjek użytkownika.

Do negocjacji: Ponieważ historia użytkownika nie jest ściśle zdefiniowana, możemy negocjować z właścicielem produktu, jak postępować z rozwojem historii użytkownika.

Wartościowe: historia użytkownika zapewni użytkownikowi końcowemu pewną wartość dodaną. Szacunkowe: Z definicją, którą wykonał użytkownik, możemy określić, ile wysiłku będziemy potrzebować, aby wdrożyć tę historyjkę użytkownika.

Mały: Wynik tej oceny powinien być możliwy do zarządzania w sprincie.

Testowalne: istnieje sposób sprawdzenia, czy osiągnęliśmy oczekiwaną długość użytkownika dla tej historii użytkownika.

Historia dewelopera

Historia programisty jest powiązana z historią użytkownika i jest bardziej szczegółowym wyjaśnieniem w krótkich zdaniach, co należy zrobić, aby spełnić wymagania historii użytkownika. Jest to pierwszy krok szczegółowej analizy, której będziemy wymagać dla każdego elementu Backlogu Produktu, aby był on wystarczająco jasny, aby rozpocząć rozwój (aby włączyć go do Backlogu Sprintu). Zgodnie z nazwą, historia programisty jest dostarczana przez programistów i jest napisana językiem, który zarówno programiści, jak i Właściciel Produktu mogą w pełni zrozumieć. Wyjaśnimy to jaśniej na przykładzie naszej ukochanej firmy HSC. W tym przykładzie przeanalizujemy bardziej szczegółowo, jaka jest historia programisty dla historii użytkownika związanej z menedżerem HR w poprzednim przykładzie. Najpierw, podobnie jak w historii użytkownika, zdefiniujemy szablon historii dewelopera.

The < data module >

will < Action/feature >

that will allow < requirement >

related to the < related user story >

Za pomocą tego szablonu zdefiniujemy nasze historie programistów, aby wspierać historię użytkowników HR

Model HR otrzyma dane z systemu META4, które pozwolą na analizę dostępności danych HR firmy związanych z HR user story. Model HR będzie zawierał hierarchię pracowników które pozwolą drążyć

poziomy hierarchii HR związane z historią użytkownika HR. Model HR zapewni raport podzielony na strony według sklepu, który umożliwi analizę kosztów HR, szkoleń i urlopów na sklep w odniesieniu do historii użytkownika HR.

Również w tym przypadku fani metodologii, którzy lubią definiować akronimy, zdefiniowali jeden, aby sprawdzić, czy historia programisty jest wystarczająco dobra, aby rozpocząć rozwój, a jego nazwa to DILBERT'S:

Możliwy do wykazania: Po opracowaniu historii programisty powinniśmy być w stanie pokazać Właścicielowi Produktu, że działa ona zgodnie z oczekiwaniami.

Niezależne: podobnie jak w przypadku historii użytkownika, musimy zdefiniować historie programisty, które można opracować bez wpływu na inną historię programisty, a programowanie powinno zapewniać funkcjonalność działającą samodzielnie.

Warstwowa: Ta cecha jest bardzo skoncentrowana w rozwoju Business Intelligence, ponieważ większość zmian po stronie BI musi uwzględniać rozwój ETL, bazy danych i interfejsu użytkownika.

Wartość biznesowa: użytkownik musi docenić nasz rozwój , wynikającye z wartości user story.

Szacunkowa: ponownie wychodząc z charakterystyki historii użytkownika, historia programisty musi być wyraźnie możliwa do oszacowania. W tym przypadku różnica polega na tym, że należy to zrobić z dokładniejszym oszacowaniem, o ile deweloper pisze, co zamierza opracować.

Możliwość dopracowania: Historia programisty musi być napisana w bardziej konkretny sposób, niż historia użytkownika, ale znowu można ją przejrzeć i dostosować podczas opracowywania.

Testowalny i mały: te dwie cechy są dokładnie takie same jak w historii użytkownika.

Rejestr Produktu

Backlog produktu to lista wymagań na wysokim poziomie, bez szczegółowych specyfikacji kwalifikowanych według priorytetu użytkownika i kosztu rozwoju, które muszą być zdefiniowane na początku projektu i przeglądane po każdym sprincie, więc nie jest to stała lista; może ewoluować i zmieniać się w trakcie projektu. Właściciel produktu jest odpowiedzialny za tworzenie i aktualizowanie tej listy. W tym przypadku odwołujemy się do przykładu wyjaśnionego na konkretnych spotkaniach Scrumowych, aby zrozumieć, czym jest Backlog Produktu oraz w jakim momencie jest tworzony i utrzymywany.

Definicja Wykonania

Definicja wykonania to lista kontrolna, którą musi spełnić każdy element, aby został uznany za wystarczająco dobry, aby można go było przenieść do środowiska produkcyjnego. Jest to walidacja jakości, która ma na celu upewnienie się, że wszystkie wymagania zostały opracowane z zachowaniem pożądanych standardów jakości. Definicja wykonanej listy kontrolnej zawiera pozycje takie jak: opracowany kod odpowiada standardom kodyfikacyjnym, testy zostały pomyślnie zakończone, testy wydajnościowe zakończyły się sukcesem, uzyskaliśmy akceptację klienta, czy dokumentacja jest zakończona.

Kiedy zacząć - gotowa definicja

Ciekawą ideą dodaną do oryginalnej metodologii Scrum jest definicja gotowości lub definicja gotowości. Jest to analogia do Definicji Ukończenia, czyli kiedy nasz rozwój można uznać za zakończony, ale związana z definicją, którą Właściciel Produktu musi zrobić dla PBI, które chce opracować. Kiedy

definiujemy Backlog Produktu, PBI jest definiowane na bardzo wysokim poziomie, więc nie jest to coś, co można zacząć rozwijać. Aby być kandydatem do włączenia do rejestru sprintu, ten PBI musi być w pełni zdefiniowany i określony zgodnie z parametrami listy Definicja lub Gotowy. Niektóre przykłady listy kontroli, które mogą zawierać Definicja lub Gotowe, mogą obejmować pytania takie jak:

* Czy przedmiot może zostać ukończony w sprincie, który zamierzamy rozpocząć?

* Czy ten element został oszacowany przez zespół programistów?

* Czy ten przedmiot sam w sobie stanowi wartość dodaną? A może trzeba to zrobić z innymi pozycjami z listy?

Musimy uzgodnić z właścicielem produktu, jakie wymagania muszą spełniać definicję funkcjonalności, aby zostać zaakceptowanym do rozpoczęcia rozwoju.

Wykres wypalania Backlogu Produktu

Jest to wykres, który pokazuje na osi X liczbę sprintów, a na osi Y liczbę elementów Backlogu Produktu oczekujących na wdrożenie lub oczekiwany wysiłek włożony w ich opracowanie. Trend powinien być spadkowy, ale może ulec pewnemu wzrostowi z powodu pewnych zmian w PB, które obejmują nowe PBI, aby spełnić nową specyfikację zdefiniowaną przez Właściciela Produktu.

Backlog sprintu

Na początku sprintu wybieramy listę PBI, które zostaną opracowane w ramach sprintu. Ten wybór to backlog sprintu. Lista elementów do opracowania musi być podzielona na zadania, które powinny zostać opracowane w maksymalnym okresie 2 dni z zaleceniem, aby zadanie mieściło się w przedziale od 2 do 8 godzin prac rozwojowych. Rejestr sprintu nie powinien być modyfikowany, chyba że istnieje poważna blokada, która utrudnia ukończenie niektórych PBI. To, co musimy na bieżąco aktualizować, to informacje dotyczące każdego zadania, czy jest jakaś blokada, ile godzin pozostało do zakończenia rozwoju i kto ma przydzielone to zadanie. Również na tej liście powinniśmy unikać zbytniego dzielenia zadań, które nie wchodzą w zakres mikrozarządzania.

Wykres wypalania backlogu sprintu

Ten wykres liniowy jest oszacowaniem pozostałego wysiłku do ukończenia sprintu. Teoretycznie powinna to być linia malejąca, która przecina linię zerową na końcu sprintu. Ale o ile pozostały wysiłek jest weryfikowany każdego dnia, może się różnić, jeśli rozpoczynając zadanie wykryjemy, że nie zostało ono poprawnie oszacowane. Zwykle wyznaczamy dwie linie, jedną z teoretycznym trendem, a drugą z rzeczywistym postępem.

Lista przeszkód

Podczas codziennych spotkań wykryjemy przeszkody lub blokady, które mogą zagrozić realizacji danego zadania lub pełnego PBI. Mistrz Scrum musi przechowywać te blokady na liście i śledzić wszystkie tematy, aby móc je odblokować. Na tej liście musimy poinformować o blokadzie, powiązanym PBI, statusie blokady, planowanej dacie rozwiązania i osobie odpowiedzialnej za jej rozwiązanie.

Lista ulepszeń

Po każdym sprincie dokonamy przeglądu naszej aktywności na spotkaniu retrospektywnym. Tam cały zespół powinien zasugerować wszelkie ulepszenia, które jego zdaniem są istotne dla wyników zespołu.

Ta lista jest również prowadzona przez Scrum Mastera i powinna zawierać element akcji, status, zaplanowaną datę zakończenia oraz osobę odpowiedzialną za jego rozwiązanie.

Ograniczanie czasu spędzonego za pomocą Timeboxingu

Czas w naszym życiu jest ograniczony. Jest również ograniczona w naszej pracy, a także w naszych projektach. Musimy ograniczyć czas, który poświęcamy na każde zadanie, aby nie tracić zbyt wiele czasu poświęconego na to samo zadanie. Praktyka Timebox określa granice każdego rodzaju zadań, aby zapewnić, że realizacja projektu nie zawiesza się. Ta technika timeboxingu jest stosowana na wszystkich poziomach, więc staramy się określić, ile czasu ma zająć cały projekt, ile czasu potrzebujemy na wydanie wydania, czyli nasz okres sprintu, a następnie ustalamy, ile czasu musimy spędzić w jednym zadania takie jak przyjmowanie wymagań, opracowywanie kodu, czyli każdy rodzaj czasu trwania spotkania, opracowywanie dokumentacji itp. Ta metodologia jest wysoce oparta na czasie, ponieważ wszystkie działania mają ograniczenia czasowe. Za tą strategią stoi teoria równoważenia trzech elementów: wysiłku, wymagań i harmonogramu. Zawsze możemy ulepszyć dwa z nich, równoważąc drugi. Możemy użyć mniej wysiłku, jeśli zmniejszymy wymagania lub zwiększymy harmonogram. Jeśli chcesz mieć więcej wymagań, możemy to zrobić, poświęcając więcej czasu lub zasobów. Jeśli chcesz skrócić harmonogram, możemy to zrobić, dodając zasoby lub zmniejszając wymagania. To, co Scrum proponuje w ramach tego ograniczenia czasowego, to próba ustalenia jednego z parametrów i maksymalizacji pozostałych w oparciu o priorytety klienta. Ustalamy, że w sprincie spędzimy 4 tygodnie. Jeżeli mamy 5 osób oddelegowanych do rozwoju to możemy zainwestować ustaloną ilość godzin. Dlatego nasz klient musi nadać priorytet zadaniom, które chce mieć dostępne, a my skupimy się na nich. Zwiększymy satysfakcję klienta, bo pomimo braku niektórych funkcjonalności, nasz klient może zacząć walidować i wykorzystywać najważniejsze dla niego zadania. W ten sam sposób ustalamy upływający czas na spotkania. Musimy zacząć od najważniejszych tematów do omówienia i nie spędzać na spotkaniu więcej czasu niż jest to wymagane. Kiedy szukamy informacji do wdrożenia nowej funkcjonalności, robimy dokumentację lub cokolwiek innego, mamy na to maksymalną ilość czasu i musimy zrobić wszystko, co w naszej mocy, aby rozwinąć zadanie. Ideą takiego podejścia jest to, że musimy uzyskać wyniki wystarczająco dobre, aby były zgodne ze standardami jakości, ale nie doskonałe. Wszystko, co doskonałe, kosztowałoby zbyt wiele czasu i wysiłku, więc byłoby sprzeczne z naszymi celami szybkiego dostarczania rzeczy.

Spike

Czasami jest jakieś zadanie, które trzeba wykonać, ale nie dodaje bezpośredniej wartości dla klienta. Opierając się na standardach Scrum, wszystkie nasze działania lub PBI same w sobie muszą zapewniać wartość dodaną, ale tworzenie oprogramowania, a w tym przypadku rozwój BI, może wymagać zrobienia czegoś, na czym tak naprawdę nie zależy naszym klientom. Aby móc włączyć je do Backlogu Produktu/Backlogu Sprintu, Scrum umożliwia zarządzanie nim przez koncepcję Spike. Pokażmy przykład. Wyobraź sobie, że użytkownik poprosił Cię o możliwość drążenia z jednego raportu do innego z bardziej szczegółowym filtrowaniem dynamicznym wybranego przez siebie klienta. Opracowałeś rozwiązanie BI przy użyciu narzędzia BI, które może wykonać tego rodzaju drążenie, ale nie w obecnej wersji, którą zainstalowałeś na swojej platformie, ale w nowej. Aktualizacja platformy do nowej wersji może być uznana za skok, jako coś, co jest wymagane, ale użytkownik nie prosi o to bezpośrednio. Innym przykładem może być sytuacja, w której w celu spełnienia danego wymagania zespół musi użyć funkcji, z której nigdy wcześniej nie korzystał. Jeśli chcesz odnieść sukces w spełnieniu tego wymogu, możliwe, że Twój zespół wymaga przeszkolenia w zakresie tej funkcji lub weryfikacji koncepcji, aby mieć pewność, że wiedzą, jak zastosować ją do potrzeb klienta.

Konserwacja z Kanbanem

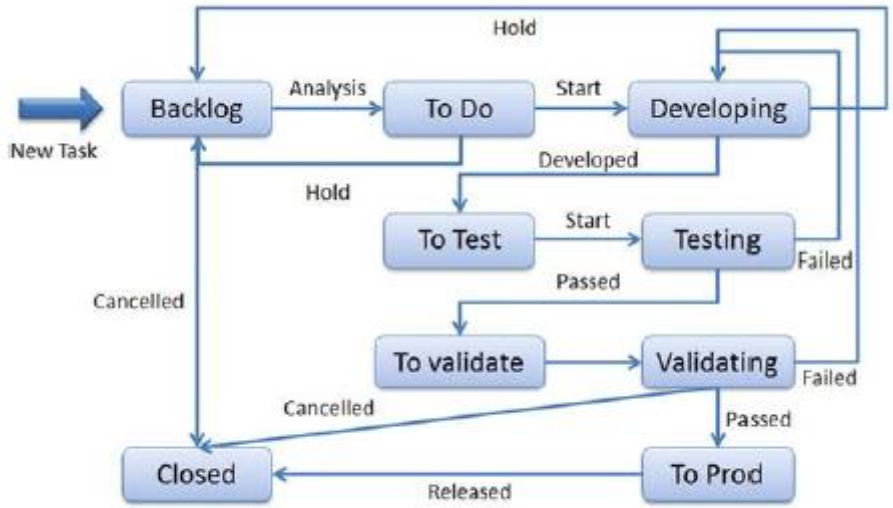
Po opracowaniu głównych funkcjonalności wymaganych przez klientów można śmiało powiedzieć, że Twój system BI wymaga konserwacji, poprawek, drobnych usprawnień i administracji. Aby przeprowadzić tę konserwację, naszym preferowanym podejściem z metodologii Agile jest Kanban, który pozwala na utrzymanie aplikacji z podejściem ciągłego doskonalenia, koncentrując się na małych zmianach w rozwiązaniu, aby dostosować je do nadchodzących wymagań.

Koncepcje Kanbana

Istnieje kilka interesujących koncepcji z teorii Kanbana, które musisz znać, aby zwinnie zarządzać swoim projektem.

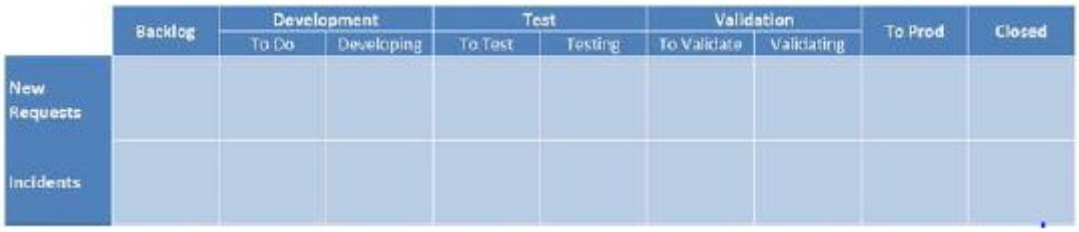
Przepływ pracy zadania

Kanban definiuje przepływ pracy zadania, definiując różne statusy, jakie zadanie powinno przyjąć, od początkowego zaległości do rozwoju, testowania, walidacji i zakończenia. Ta lista statusów może się różnić w zależności od naszego środowiska i być może w zależności od typu zadania, ale ważne jest, aby były one poprawnie zdefiniowane, aby zmaksymalizować wydajność zespołu. Na rysunku możesz znaleźć przykład przepływu pracy dla Kanbana.



Tablica zadań i zarządzanie wizualne

Aby mieć kontrolę nad wszystkimi zadaniami, wykrywać wąskie gardła, wykrywać blokady i łatwo komunikować się wewnętrznie z zespołem, czyli statusem wszystkich zadań i postępem, jaki zespół wykonał na co dzień, ważne jest, aby mieć tablica, która zawiera wszystkie zadania wewnątrz, używając karty do każdego zadania i przesuwając ją do przodu, o ile zadanie przechodzi przez inny status zdefiniowane w przepływie pracy. Tablica będzie zorganizowana z kolumną dla każdego statusu i jednym lub kilkoma wierszami w zależności od typów zadań, priorytetów, incydentów vs. próśb lub jakiejkolwiek potrzeby, którą mamy w naszym projekcie. Rysunek przedstawia przykład tablicy zadań Kanban.



Zarządzanie wizualne jest bardzo ważną cechą Kanbana. Ważne jest, aby mieć fizyczną tablicę, a przynajmniej projektor lub ekran, który jest stale widoczny, aby cały zespół mógł w każdej chwili zobaczyć, w jakim jesteśmy stanie i jakie dalsze kroki są wymagane. Jako metodologia Agile, Kanban udaje, że codziennie odbywa 15-minutowe spotkanie z całym zespołem, aby skomentować status zadania i jakie są zmiany statusu każdego zadania.

Praca w toku

Jednym z najważniejszych parametrów w naszym systemie Kanban będzie Work In Progress czyli WIP. Dla każdego statusu ustawimy WIP, określając maksymalną liczbę zadań, które możemy mieć w określonym statusie. Możliwe, że nie wszystkie kolumny wymagają WIP i trzeba go dostosować do projektu, nad którym pracujemy; posiadanie wysokiego WIP spowoduje, że będziemy przeciążeni, dlatego zaczynamy od wielozadaniowości; a jedną z idei rozwoju Agile jest to, że musimy skoncentrować się na pojedynczym zadaniu w danym momencie, aby być bardziej wydajnym w naszym rozwoju. Moglibyśmy również cierpieć z powodu długich czasów oczekiwania: długich kolejek w statusie pośrednim i utraty jakości. Jeśli zdefiniujemy niski WIP, być może mamy osoby, które nie mają nic przypisanego.

Czas realizacji

W systemie Kanban czas realizacji to ilość czasu, jaką potrzebujemy na opracowanie zadania, ponieważ przesuwamy je do stanu początkowego, aż do jego zamknięcia. Nie uwzględnia czasu zaległości. Mierząc średni czas realizacji, będziemy mogli zobaczyć, jak skuteczna jest strategia Kanban. Chodzi o to, że zadania nie pozostają w statusie w nieskończoność, więc jeśli wykryjemy, że kolumna ma więcej zadań niż powinna, musimy przenieść zasoby, aby pomóc tym, które mają blokady, aby uniknąć wąskich gardeł.

Uwaga: Bardzo ważne jest, aby zauważyć, że Kanban ustala pewne podstawowe zasady, ale musi być dostosowany do każdego środowiska, kiedy próbujemy skonfigurować system, oraz że musi być elastyczny i stale dostosowywany, aby poprawić wydajność zespołu. Będziemy musieli dostroić każdy WIP o każdym statusie, zdefiniować różne tablice zadań Kanban dla różnych typów działań, dostosować status, jaki ma dany typ działania, a wszystkie te adaptacje powinny być dokonywane za pomocą okresowych spotkań retrospektywnych z całym zespołem, który pozwoli nam udoskonalić sposób, w jaki pracujemy.

Mieszanka obu metodologii, Scrumban

Nasze początkowe rozważania dotyczące mieszania pewnych cech obu metodologii wcale nie są czymś, co wymyśliliśmy; przeszukując teorię Agile, można znaleźć metodologię Scrumban. Scrumban stosuje obie metody, ale jest mniej rygorystyczny niż Scrum, który udaje, że ma uniwersalnie wymiennych członków zespołu, więc dopuszcza pewną wyspecjalizowaną rolę, która, jak zobaczysz w następnej sekcji, lepiej pasuje do rozwoju BI; i jest trochę bardziej zorganizowany niż Kanban, używając iteracji do zorganizowania pracy, jak w Scrumie. Scrumban pasuje lepiej niż Scrum w projektach o dużej zmienności, a także w projektach utrzymaniowych lub projektach, w których możemy spodziewać się

wielu błędów ze względu na zmienność informacji źródłowych: jak na przykład projekt oparty na narzędziach Big Data. Jak wspomniano we wstępie do tego rozdziału, nie pretenduje do miana wyczerpującej analizy metodologii Agile, dlatego zachęcamy do ich zbadania; będziesz mógł znaleźć setki stron w Internecie z informacjami na ich temat. Rozważamy bardziej interesujące skupienia na osobliwościach Agile dla BI, co, mam nadzieję, zobaczysz, jeśli będziesz kontynuować czytanie.

Specyfika Scruma dla BI

Zasady i metodologie omówione powyżej są ogólne dla SCRUM w ramach każdego procesu tworzenia oprogramowania, ale istnieją pewne cechy szczególne, które mają wpływ na projekty BI, które są specyficzne dla tego środowiska; łatwo zrozumiesz, co mamy na myśli, przechodząc do tej sekcji.

Sprint 0 - Dłuższa analiza początkowa

Ok, podjęliśmy decyzję o rozpoczęciu rozwoju naszej hurtowni danych, mamy zgodę najwyższego kierownictwa i chcemy zastosować metodologię Scrum. W tym celu zbierzemy wszystkie historie użytkowników; stworzymy nasz Product Backlog, szacując na bardzo wysokim poziomie potrzeby i koszt każdego zadania; niektóre z nich wybierzemy na podstawie priorytetu Właściciela Produktu; uzupełnimy Backlog Sprintu, dokładniej analizując potrzeby każdego PBI; i zaczniemy się rozwijać. Po czterech sprintach dochodzimy do analizy jednego PBI o niskim priorytecie, który był na liście od początku i miał niski priorytet, ale duży wpływ, który spowoduje przebudowę całej wykonanej do tej pory pracy. Dzieje się tak, ponieważ wdrażając bazę danych hurtowni danych, należy zdefiniować model logiczny oparty na jednostkach, które muszą mieć jasno określone relacje; które pojęcia lub atrybuty mają relacje; jakie są rodzaje relacji: relacje jeden-do-wielu, wiele-do-wielu; który jest pojedynczym kluczem do powiązania tabel; a na koniec musimy zdefiniować klucze, aby móc uruchamiać poprawne instrukcje SQL, które powinny zwracać spójne informacje. Musisz również wziąć pod uwagę, że spodziewasz się tam zapisanej długiej historii, dużych tabel z milionami wierszy informacji, które są sprawdzane, formatowane, przekształcane lub cokolwiek innego, czego wymaga Twój proces, aby spełnić Twoje potrzeby. W tej sytuacji za każdym razem, gdy przenosisz zmianę do produkcji, może wymagać przetworzenia dużej ilości informacji, jeśli nie masz solidnego rozwiązania dla swojego schematu tabeli. Tak więc jedną z głównych cech charakterystycznych zastosowania Scruma w projekcie Business Intelligence jest konieczność wcześniejszego myślenia i głębszej analizy implikacji historyjki użytkownika w rozwiązaniu technicznym, które próbujesz dostarczyć. Ponownie, aby zilustrować, jaki może być problem nieprzestrzegania tych zaleceń, pokażmy przykład z firmy HSC. Jest to przykład pochodny od rzeczywistego, którym musieliśmy zarządzać w naszym doświadczeniu w innej firmie, z tą różnicą, że w tamtym momencie nie używaliśmy Scruma. W naszych przykładowych historiach użytkowników znajdujemy cztery różne prośby o dodanie nowych danych do systemu, ale wyobraźmy sobie, że masz ich 40 lub 50. Podążamy za priorytetem zdefiniowanym przez użytkownika i zaczynamy wdrażać pierwsze:

Jako właściciel produktu chcę mieć możliwość drążenia hierarchii klientów, aby użytkownicy mogli analizować szczegóły sprzedaży poprzez organizację klientów ERP.

Moglibyśmy pomyśleć o prostym modelu logicznym obsługującym to wymaganie w oparciu o schemat gwiaździsty z zaledwie czterema tabelami: fakty dotyczące sprzedaży oraz wymiary klienta, produktu i czasu. Podobnie jak w modelu tabelarycznym, powiązane wymiary w narzędziu BI mają swój odpowiedni identyfikator, który określa klucz do łączenia między tabelami. Podążamy za innymi historiami użytkowników, które dodają poziomy do każdej hierarchii; istnieje inna historyjka użytkownika, która wymaga poprawy szybkości, dlatego do naszego modelu dodajemy zagregowane tabele, nowe wymiary analizy oraz coraz więcej tabel i atrybutów. Po kilku sprintach postanawiamy uwzględnić w naszej analizie tę historyjkę użytkownika:

Jako Sales Force Manager chcę analizować wyniki sprzedaży w poszczególnych sklepach, aby móc analizować sprzedaż według kategorii, grup klientów i kampanii promocyjnych.

Nasze początkowe oszacowanie na wysokim poziomie było, ok, przejdźmy do dodania sklepu do tabeli faktów dotyczących sprzedaży i dodajmy tabelę faktów promocyjnych z wyszukiwaniem sklepów. Ale potem zaczynamy szczegółową analizę:

* Sklepy nie mają tej samej liczby produktów; to zależy od wielkości sklepu i lokalizacji. Z tego powodu kategoria niektórych produktów może się różnić w zależności od sklepu, ponieważ niektóre narzędzia są uważane za Ogród w przypadku dużego sklepu z naciskiem na rolnictwo, a za Inne narzędzia w innym sklepie w centrum miasta. Więc kategoria zależy od sklepu. Aby rozwiązać ten problem, musimy dodać sklep do wyszukiwania produktów i ponownie przetworzyć cały wymiar.

* Grupy klientów są również definiowane przez sklep na podstawie rodzaju klientów mieszkających w pobliżu sklepu. Również kod klienta może się powtarzać wzdłuż sklepów, więc aby mieć unikalny identyfikator klienta, musimy dodać sklep do wyszukiwania klientów i ponownie przetworzyć cały wymiar.

* Myślisz na czas … czas nie może się zmieniać. Relacje w hierarchii czasowej są takie same dla wszystkich sklepów… Znowu się mylisz. Aby przeanalizować liczbę dni, które mamy na daną kampanię, musimy policzyć dni robocze. Ponieważ sklepy znajdują się w różnych miastach i regionach, mają różne lokalne święta. Więc znowu musimy dodać sklep do wymiaru czasu i ponownie przetworzyć cały wymiar czasu.

* Twój model logiczny również wymaga przeglądu, aby uwzględnić magazyn terenowy w definicji klucza dla klienta, grupy klientów, kategorii produktu, dnia, typu i wielu innych koncepcji utworzonych do analizy.

Uwaga: Podsumowując tę sekcję, będziemy musieli poświęcić więcej czasu na szczegółowe przeanalizowanie, które historie użytkowników mogą wymagać dużych modyfikacji w naszym modelu. Na koniec dnia powinniśmy postarać się o jak najpełniejszą definicję schematu przed rozpoczęciem programowania. Na pewno pojawią się wymagania, które będą wymagały dostosowań i zmian, ale powinniśmy starać się zapobiegać im w jak największym stopniu.

Segmentacja projektów BI

Po spotkaniach Sprint 0 będziesz mieć wiele próśb do rozważenia od różnych interesariuszy, dla różnych działów, przy użyciu różnych narzędzi do programowania na różnych warstwach BI przez różne role programistów lub przynajmniej kogoś z różnymi obszarami wiedzy programistycznej. Tak więc pod koniec oceny wymagań przeprowadzonej na tych spotkaniach Sprintu 0 prawdopodobnie będziesz miał dużo pracy do wykonania dla wielu zespołów, które o to proszą. W tym momencie masz dwie możliwości: uciec daleko i zostać ogrodnikiem w małym miasteczku lub spróbować zacząć od tego niesamowicie dużego nakładu pracy. Jeśli już doszedłeś do tego miejsca w książce, jestem prawie pewien, że wybierzesz drugą opcję. Jak więc zacząć? Jak omówiono w poprzedniej sekcji, powinieneś zainwestować dużo czasu w analizę wszystkich przychodzących historyjek użytkowników. Z tej analizy powinieneś otrzymać schemat logiczny bazy danych, szkic raportów, które użytkownicy chcą otrzymywać z Twojego systemu BI, a które powinny być główną funkcjonalnością Twojej platformy. Dzięki temu przeglądowi wszystkich oczekujących historii użytkowników do wdrożenia powinieneś być w stanie podzielić je na mniejsze powiązane grupy, aby móc zapewnić działające rozwiązanie po

każdym sprincie. Aby poprawnie zdecydować, jak pogrupować wszystkie powiązane zadania, należy przeanalizować wymagania pod trzema różnymi punktami widzenia:

Obszar funkcjonalny : Musisz podzielić zadania na mniejsze grupy powiązane powiązanym obszarem funkcjonalnym, do którego należą. Możesz więc sklasyfikować oczekujące zmiany jako związane z finansami, związane ze sprzedażą, związane z operacjami, a także komponenty obejmujące różne środowiska, takie jak dane klientów lub dane produktów. W celu przeprowadzenia prawidłowej analizy obszarów funkcjonalnych pomocnym narzędziem będzie analiza modelu logicznego.

Analiza techniczna: mając przegląd wszystkich wymaganych pól do analizy informacji, możesz stwierdzić, że istnieją początkowo niepowiązane historie użytkowników, które mają to samo techniczne źródło i/lub cel. Wyobraź sobie, że Twój kierownik ds. sprzedaży poprosił o wyświetlenie informacji o produktach sklasyfikowanych według rodziny produktów, a Twój kierownik finansowy chce zobaczyć, jaką wagę dajesz swoim klientom na promocję danego produktu. Oba pola informacyjne wydają się być całkowicie niezależne, ale możesz zdać sobie sprawę, że oba pochodzą z tej samej tabeli w Twoim systemie ERP i zamierzasz umieścić je w tej samej tabeli w miejscu docelowym, więc połączenie obu historyjek użytkownika obniży całkowity koszt co do kosztu wykonania ich osobno. Aby prawidłowo przeprowadzić analizę techniczną należy skupić się na fizycznym modelu systemu.

Doświadczenie użytkownika końcowego: w narzędziu BI możesz mieć złożone raporty, które mieszają informacje z różnych obszarów funkcjonalnych, dane pochodzące z różnych systemów i umieszczone w naszym magazynie danych w różnych celach. Ale być może Twój kluczowy użytkownik chce mieć wszystkie te informacje razem w jednym raporcie, więc aby dostarczyć w pełni działający ekran, możesz zdecydować, że powinieneś opracować w tym samym czasie osobną analizę, aby móc zapewnić gotowy rozwój.

Po tej drugiej analizie, aby pogrupować wszystkie zadania, możesz zobaczyć, że niektóre z tych podzielonych na segmenty historii użytkowników są zbyt duże, aby można je było uwzględnić w sprincie, a także powiązane zadania mają wycenę nadmiernego nakładu pracy, aby zmieścić się w oczekiwanym maksymalnym rozmiarze 8 godzin pracy. Aby móc podążać za metodologią Agile, będziesz musiał podzielić segmenty na mniejsze części. Aby to zrobić, możesz powtórzyć analizę funkcjonalną, ale skupiając się na wybraniu z modelu logicznego minimum części, które mężczyzna ma ochotę wspólnie opracować, aby uzyskać mniej tabel i pól do zarządzania, a także możesz uzgodnić z właścicielem produktu, że niektóre części raportu będą wyświetlane z komunikatem „W trakcie opracowywania” aż do następnego sprintu. Ostatecznie byłby to sposób na segmentację według tabel, o ile wybierasz niektóre tabele, które zostaną dodane do modelu, wraz z powiązanym rozwojem BI i ETL. Możesz także podzielić cały projekt, dzieląc załadowane informacje na segmenty. Możesz załadować informacje dla pewnego okna czasowego (bieżący rok, bieżący miesiąc) i pozostawić dane historyczne dla następnego sprintu lub możesz załadować informacje kraj po kraju w środowisku wielonarodowym lub firma po firmie w klastrze firm lub klient po kliencie lub, w przykładzie, który śledziliśmy przez cały rozdział, sklep po sklepie. Jeśli te podziały nie wystarczą, możesz pomyśleć o wybraniu tylko niektórych kolumn swoich tabel, specjalnie pogrupowaniu ich według tabel źródłowych lub środowisk. A jeśli powoduje to, że zadania są nadal zbyt długie, możesz pomyśleć o podzieleniu ich według typu, więc najpierw możesz załadować metryki podstawowe, następnie zaimplementować pochodne z tych metryk, a następnie przejść do agregacji itp.

Działania front-end vs. back-end

Inną cechą charakterystyczną projektów BI są różnice w rozwoju działań frontendowych i backendowych. Aby móc wyświetlać określone informacje w narzędziu front-end BI, konieczne będzie

załadowanie informacji do bazy danych back-end. Z drugiej strony ładowanie informacji do bazy danych, ale bez żadnego raportu front-end do ich analizy, nie ma sensu w rozwiązaniu BI. Tak więc działania front-end i back-end są ze sobą ściśle powiązane; w rzeczywistości większość historyjek użytkownika będzie miała historie deweloperskie z obu typów, o ile wymagane są obie czynności, ale mają one zupełnie inny charakter. Działania backendowe to praca w cieniu, brudna robota, której nikt bezpośrednio nie widzi, podczas gdy raporty frontendowe będą dostępne dla wszystkich, narzędzie BI będzie interfejsem użytkownika umożliwiającym dostęp do informacji na backendzie. Z tego powodu wymagania będą zupełnie inne. Narzędzie front-end jest używane bezpośrednio przez klientów końcowych, więc będziesz musiał wziąć to pod uwagę, aby zdefiniować główne zasady. Nie możesz mieć bardzo ścisłych zasad nomenklatury i konwencji nazewnictwa w interfejsie użytkownika, ponieważ Twoi użytkownicy zwykle nie mają wiedzy technicznej, a nazwy raportów, opisy i obrazy muszą być zorientowane na użytkownika, aby umożliwić użytkownikowi łatwe poruszanie się po narzędziu. Twój interfejs użytkownika powinien być intuicyjny i łatwy w użyciu; będzie to ważniejsze niż rozwiązanie techniczne i możliwości, które oferujesz. Zamiast tego komponent zaplecza ma większe wymagania techniczne, więc możesz ustawić konwencję nazewnictwa, która zmusza programistów do przestrzegania nomenklatury w obiektach bazy danych, obiektach ETL, procedurach programistycznych, stosować najlepsze praktyki dostrajania wydajności itp. W tej części będziesz miał silniejszą pozycję, aby zmusić programistów do przestrzegania reguł, a kilku programistów do ich nauki, również z umiejętnościami technicznymi, które ułatwi ten cel. Na froncie Twoi klienci mają moc decyzyjną, więc musisz dostosować się do ich wymagań. Będziesz miał wielu użytkowników, być może kontaktują się z Tobą przez jakiegoś kluczowego użytkownika, na przykład właściciela produktu, ale na końcu będziesz potrzebować rozwiązania zrozumiałego dla wielu różnych osób, o różnych poziomach wiedzy i z różnych funkcjonalnych światów, podczas gdy dostęp do zaplecza powinni mieć tylko ludzie z głębszą wiedzą na temat tego, co robią.

Izolacja roli - Specyficzna wiedza

Zwinne tworzenie oprogramowania udaje, że ma w pełni elastyczne zespoły, w których wszystkie komponenty zespołu mają wiedzę programistyczną, a historie programistów może wymyślać każdy z nich, ale jak można się domyślić, na podstawie poprzedniej sekcji, narzędzia programistyczne i umiejętności dla dowolny komponent platformy BI są zupełnie inne. Będziesz potrzebować umiejętności technicznych do tworzenia struktur baz danych i procesów ETL, z myśleniem matematycznym, zdolnym do definiowania struktur, relacji, procedur ładowania i zrozumienia modelu relacyjnego, gdy mówisz o narzędziach zaplecza, znajomości definicji schematu w celu zdefiniowania BI narzędzia, modele, umiejętności projektowania funkcjonalnego i graficznego w celu zdefiniowania raportów końcowych, a znalezienie kogoś, kto ma wszystkie te różne umiejętności, jest trudne i kosztowne, więc prawdopodobnie będziesz potrzebować różnych podzespołów w swoim zespole skupionych na różnych obszarach rozwoju. Tak więc członkowie Twojego zespołu nie będą w pełni wymienni, aby pracować w różnych historiach użytkowników, co zwiększy złożoność zarządzania zespołem. Aby móc ruszyć z całym projektem będziesz potrzebować w swoim zespole przynajmniej kogoś, kto posiada którąś z poniższych umiejętności:

* Architekt danych: odpowiedzialny za zdefiniowanie modelu fizycznego, struktur tabel, typów pól, modelu bazy danych itp.

* Programista bazy danych: ta rola będzie rozwijać wymagane struktury zdefiniowane przez architekta danych wewnątrz bazy danych.

* Deweloper ETL: będzie odpowiedzialny za zdefiniowanie procesów ładowania z systemów źródłowych do docelowej bazy danych.

* Data modeler: ta rola będzie wymagać wiedzy o tym, jak zdefiniować i zaimplementować w narzędziu BI model logiczny, który umożliwi narzędziu BI wygenerowanie poprawnego kodu SQL do uruchomienia w bazie danych. Te zadania ról będą się różnić w zależności od wybranego narzędzia; istnieją złożone narzędzia BI, które pozwalają tworzyć złożone modele z setkami funkcjonalności oraz prostsze narzędzia, które nie będą wymagały dużych możliwości modelarza danych.

* Programista front-end: Będziesz także potrzebował w swoim zespole kogoś z wiedzą funkcjonalną, aby zrozumieć potrzeby klienta i powielić je w systemie BI. Ta rola będzie wymagała również dobrych umiejętności komunikacyjnych, o ile będzie głównym interaktorem z klientem końcowym.

* Projektant graficzny: W tym przypadku posiadanie grafika w zespole nie jest obowiązkowe, ale ponieważ możliwości raportowania stają się coraz bardziej atrakcyjne wizualnie, zaleca się, aby w zespole był ktoś z umiejętnościami projektowania graficznego, który może współpracować w definicji raportów, do których użytkownik będzie miał dostęp.

Te różne role i umiejętności mogą utrudniać wymianę zespołu programistów z ról, które mają, jeśli chcesz mieć zespoły multidyscyplinarne, będziesz musiał zainwestować w szkolenie zespołu, co z drugiej strony jest zawsze godne polecenia. Zdajemy sobie również sprawę, że jeśli pracujesz w małej firmie próbującej wdrożyć rozwiązania BI, możliwe, że będziesz jedyną osobą odpowiedzialną za robienie wszystkich rzeczy, więc raczej nie zalecamy kontynuacji czytania tej książki o ile postaramy się dać ci podstawę, kontynuuj rozwój wewnątrz wszystkich komponentów BI. Z pewnością przedstawimy przegląd typowych typów obiektów, ale będziesz musiał uzyskać głębszą wiedzę na temat narzędzi, których będziesz używać do każdego komponentu. Możesz znaleźć kilka książek poświęconych w całości każdemu dostawcy oprogramowania, więc w tej książce nie możemy pokazać wszystkich opcji dla wszystkich różnych narzędzi, byłoby to prawie niemożliwe do osiągnięcia.

Typy historii programistów w BI

Ta sekcja jest ściśle powiązana z poprzednią sekcją, ponieważ będziesz potrzebować różnych ról w swoim zespole, aby móc kontynuować opracowywanie różnych historii programistów. Określamy tutaj historie deweloperów, a nie historie użytkowników, ponieważ na koniec dnia użytkownik będzie chciał mieć narzędzie dostępne w narzędziu front-endowym i aby móc zapewnić użytkownikowi tę funkcjonalność, najczęściej będziesz musiał podzielić tę historię użytkownika na wiele historii programistów dotyczące modelowania danych, ETL, schematu narzędzia BI oraz raportów lub pulpitów nawigacyjnych narzędzia BI. Nie wszystkie historie użytkowników będą miały wszystkie komponenty; być może żądanie użytkownika wykonuje historyczne obciążenie dla istniejącej analizy, a Twoim zadaniem będzie ponowne uruchomienie procesów ETL i walidacja wyników, bez żadnych modyfikacji narzędzii BI; być może masz jakieś żądanie, które wymaga tylko modyfikacji istniejącej relacji między wymiarami danych, która dotyczy tylko twojego schematu BI, lub zmiany typu kolumny w bazie danych, aby móc pomieścić dłuższe opisy dla istniejącego pojęcia. Ostatecznie celem wszystkich tych modyfikacji będzie pokazanie wyniku w raporcie BI, ale sama modyfikacja być może nie wymaga tworzenia żadnego raportu.

Historie deweloperów modelowania danych

W przypadku historii programistów zajmujących się modelowaniem danych będziesz musiał skoncentrować się na rozwoju bazy danych. Oznacza to, że na pewno będziesz potrzebować umiejętności bazodanowych i ewentualnie architekta danych. Historie modelowania danych określą, jaka jest pożądana struktura tabeli, i na tej podstawie programiści będą pracować z narzędziami do projektowania baz danych, aby stworzyć całą wymaganą infrastrukturę. Aby zweryfikować proces modelowania danych, najpierw przetestujemy za pomocą zapytań, czy oczekiwane sprzężenia

zwracają poprawne dane, sprawdzimy integralność danych w systemie i musimy upewnić się, że każda kolumna została zdefiniowana z najbardziej odpowiednim typem kolumny i rozmiar. Musimy również wziąć pod uwagę, w jaki sposób zmiana może zostać zastosowana do istniejącego środowiska produkcyjnego, na przykład, jeśli nasz transport obejmuje zmianę typu kolumny w istniejącej tabeli, będziemy musieli sprawdzić, czy potrzebujemy tabeli pomocniczej, aby zachować istniejącą Informacja; lub jeśli chcemy utworzyć nową tabelę w produkcji, musimy sprawdzić, czy musi ona zostać utworzona z informacjami o środowiskach deweloperskich lub testowych użytkowników, czy też musi być utworzona pusta, aby wypełnić ją powiązanym procesem ETL.

Historie programistów ETL

Historie programistyczne ETL opierają się na sposobie, w jaki zamierzasz uzupełnić informacje do bazy danych, więc w analizie, aby móc je uruchomić, będziesz potrzebować wiedzy, jakie jest źródło informacji, jak możesz połączyć się z tym źródłem informacji, jaka jest oczekiwana częstotliwość ładowania, które tabele docelowe należy wypełnić, w jaki sposób zdefiniowano obciążenie, przyrostowe lub całkowite, oraz jakie są wymagane kontrole w celu sprawdzenia, czy informacje zostały załadowane poprawnie. Ponownie musimy potwierdzić, że nasz rozwój jest prawidłowy. Aby zweryfikować historie programistów ETL, będziemy wymagać sprawdzenia, czy informacje w źródle są takie same jak w celu, czy nasz proces nie powiela informacji, czy nie tracimy informacji na żadnym etapie ładowania, czy wydajność ładunku i czas ładowania są prawidłowe, że ustawiliśmy ten ładunek we właściwej pozycji w stosunku do pozostałych ładunków, a jeśli takie istnieją, że nasz rozwój wpisuje się w definicję najlepszej praktyki dla ETL w naszej Spółce.

Historie deweloperów modeli BI

W scenariuszu deweloperskim mającym na celu modyfikację modelu BI zostanie zdefiniowany zestaw obiektów narzędzi BI, które będą musiały zostać zaimplementowane, aby mogły być używane w interfejsie raportowania. Typy obiektów różnią się znacznie w różnych narzędziach; istnieją narzędzia, które wymagają dużych nakładów na modelowanie i które są zwykle bardziej niezawodne, przeznaczone do dużych wdrożeń, a inne narzędzia są łatwiejsze w zarządzaniu i mają niewiele wymagań dotyczących modelowania. Tak więc w zależności od narzędzia zdecydujesz, czy wysiłek związany z opowiadaniami modelu BI będzie duży, czy mały. Kiedy modyfikujemy definicję modelu BI, musimy upewnić się, że modyfikacja, której dokonujemy, nie spowoduje niepożądanego efektu w istniejącej strukturze. Musimy więc zweryfikować zmiany, które wprowadzamy, ale także musimy mieć zestaw ogólnych walidacji zdefiniowanych w narzędziu BI, aby upewnić się, że nie zmienią się one, gdy zmodyfikujemy istniejący model. Oczekiwanym rezultatem opracowania modelu BI jest to, że kiedy użyjesz tych obiektów w narzędziu, wygenerują one oczekiwane zapytanie SQL, więc będziesz musiał sprawdzić, czy korzysta z oczekiwanych pól, łącząc tabele przy użyciu odpowiednich pól kluczowych, bez niepożądane użycie stołu.

Historie deweloperów raportów i pulpitów nawigacyjnych

Jako najnowszy rodzaj deweloperskich historii standardowego systemu BI, chcielibyśmy omówić ostatni krok, czyli raporty i pulpity nawigacyjne, z którymi użytkownik będzie miał do czynienia. Aby móc opracować raport, będziemy musieli wiedzieć, jakie informacje nasz użytkownik chce zobaczyć, w jakiej kolejności, w jakim formacie, czy będzie jakiś obraz, ścieżki wiercenia, informacje firmowe, czyli szczegółowość wymaganych danych, czy istnieje jakieś pole grupujące, jakie są filtry, które mają zastosowanie do raportu, czy istnieje jakikolwiek selektor do interakcji z informacjami, ile paneli informacji ma zostać wyświetlonych oraz wszelkie inne szczegóły, które mogą być istotne dla raportu . Ponownie będzie to zależeć od narzędzia, którego zamierzasz użyć i jakie możliwości ci ono oferuje. Będzie to ta część, która zostanie bezpośrednio zweryfikowana przez właściciela produktu i klienta, ale

będzie miała niejawną weryfikację pozostałych komponentów rozwiązania BI, o ile uruchamia raport, który użytkownik może zweryfikować, jeśli baza danych ma został poprawnie zamodelowany, czy ETL poprawnie ładuje informacje i czy nasz model został poprawnie zdefiniowany w narzędziu BI. W ramach walidacji musimy również upewnić się, że wykonanie raportu jest zgodne z oczekiwanym.

Historie programistów MOLAP

MOLAP wykracza poza standardowe rozwiązanie BI, ale może być również włączony do rozwoju Agile, gdy potrzebujemy opracować dowolną bazę danych MOLAP. Historie programistów MOLAP muszą określać źródło informacji; jakie wymiary firmy mają znaleźć się w bazie; w jaki sposób zagreguje wszystkie wymiary; który interfejs będzie używany do ładowania informacji; czy informacje zostaną załadowane ręcznie, automatycznie lub w połączeniu z obydwoma; w jaki sposób zostanie zintegrowany z resztą platformy BI; czy zapiszemy informacje z MOLAP do hurtowni danych; czy będziemy odczytywać rzeczywiste informacje z hurtowni danych; oraz jaki jest okres użytkowania i przewidywana częstotliwość obciążenia narzędzia. Walidacja rozwoju MOLAP powinna uwzględniać, czy dane agregują się poprawnie, czy metryki zostały poprawnie zdefiniowane i czy wydajność ładowania informacji jest akceptowalna. Jako raportowanie BI, to narzędzie będzie również używane przez użytkownika końcowego, więc kluczowi użytkownicy i właściciele produktów będą również współpracować przy walidacji rozwoju MOLAP

Zwinne narzędzia zarządzania

Istnieje kilka narzędzi, które mogą nam pomóc w zarządzaniu wszystkimi rzeczami, o których mówiliśmy, przydzielaniu zadań, posiadaniu tablicy w naszym komputerze, tworzeniu profili użytkowników, grup, ustalaniu czasu realizacji zadań itp. Idziemy aby pokazać Ci, jak korzystać z jednego z nich, a także informacje o innych narzędziach, które mogą Cię zainteresować.

Trello

Trello to narzędzie typu open source, które umożliwia tworzenie własnych paneli roboczych, w których można zlokalizować tablicę zadań zarówno dla metodologii Scrum, jak i Kanban. Na stronie internetowej http://www.trello.com można uzyskać dostęp do narzędzia. Uzyskując dostęp do swojej tablicy, zobaczysz różne karty, które na koniec dnia są kolumnami, które pozwolą ci uporządkować różne statusy, jakie może przyjąć zadanie. Wewnątrz każdego zadania możesz zobaczyć różne pola, takie jak kto przydzielił to zadanie (w Trello jest brane pod uwagę, kto jest członkiem), jaki termin ma zostać dostarczony, opis zadania, komentarze, etykiety, dodaj załączniki i ciekawą funkcję Trello dodaj listy kontrolne, które pomogą Ci zdefiniować kroki, walidacje lub wymagania lub Definicję ukończenia. Gdy zadanie zmienia swój status, możesz po prostu przeciągnąć i upuścić zadanie w Trello, aby przenieść je do następnego stanu. Możesz także subskrybować zadanie, aby otrzymywać wiadomości e-mail za każdym razem, gdy zadanie zostanie zaktualizowane. Z drugiej strony masz możliwość tworzenia użytkowników i zespołów, a następnie możesz przypisać tablicę do zespołu. Uważamy, że to narzędzie jest dość proste w użyciu, ale jednocześnie całkiem kompletne, aby śledzić całą wykonaną pracę. Możesz także uaktualnić do wersji płatnej, która zapewni Ci zaawansowaną funkcjonalność i dodatkowe miejsce do zapisywania załączników.

Oprogramowanie JIRA

Jest to jedno z najpopularniejszych programów do zarządzania rozwojem z perspektywy Agile. Możesz tylko ocenić oprogramowanie za darmo, ale możesz je zdobyć dość tanio (10 USD), jeśli chcesz je zainstalować na swoim serwerze i od 10 USD miesięcznie, jeśli chcesz mieć je w chmurze. Mamy duże doświadczenie w korzystaniu z tego narzędzia do zarządzania i jest ono bardziej kompletne niż Trello,

ale o ile zapewnia więcej opcji, zarządzanie nim może być nieco trudniejsze. Można go znaleźć na stronie dostawcy (Atlassian):

https://www.atlassian.com/software/jira

Możesz dostosować swoją tablicę, aby zdefiniować kolumny i wiersze, aby zlokalizować zadania, ale możesz dodać do 140 pól do zadań, aby móc je następnie grupować i klasyfikować, dzięki czemu będzie to bardziej elastyczne; ale jeśli użyjesz wielu z tych pól, zwiększysz ogólne koszty utrzymania tego oprogramowania. Na rysunku 2.13 możesz zobaczyć niektóre z dostępnych pól, które można skonfigurować dla każdego zadania; możesz skonfigurować ponad 100 pól. Ponieważ JIRA jest zorientowana na użycie Kanbana, możesz zdefiniować WIP każdej kolumny, a tło kolumny jest zaznaczone na czerwono, gdy masz więcej zadań w kolumnie niż zdefiniowany WIP. Tablica działa bardzo podobnie do Trello, ponieważ możesz przeciągnąć i upuścić zadanie, aby zmienić status, przypisać zadania dowolnemu członkowi zespołu, dodać załączniki lub śledzić konkretne zadanie opracowane przez innych członków zespołu. Oprócz tego możesz również zdefiniować swoje dashboardy, aby analizować sposób, w jaki pracujesz, określać średni czas dostawy, śledzić ostatnią aktywność na platformie, zużyty czas, kalendarz dat dostaw przychodzących i wiele innych gadżetów, które można umieścić na pulpicie nawigacyjnym. Istnieje wiele innych opcji w JIRA, ale nie chcielibyśmy spędzać zbyt wiele czasu na tym temacie, ponieważ mamy wiele innych komponentów i narzędzi do analizy. Zachęcamy do oceny JIRA, Trello lub dowolnego innego oprogramowania do zarządzania rozwojem, które znajdziesz i sprawdzenia, które z nich lepiej dostosowuje się do Twoich potrzeb.

Uwaga: W ramach podsumowania tego rozdziału chcielibyśmy zalecić przestrzeganie zasad Agile, ale dostosowanie ich do Twojej organizacji i Twoich potrzeb. Każda metodologia zarządzania powinna być czymś, co ułatwia nam życie, a nie czymś, co tylko zwiększa złożoność i koszty zarządzania naszymi projektami. Zachowaj prostotę, a przynajmniej tak prostą, jak tylko potrafisz.

Wniosek

W tej części zobaczyliśmy, jak wykorzystać metodologie Agile w zarządzaniu projektami, dodając do projektu warstwę zarządzania, która może pomóc w zorganizowaniu całej liczby zadań wymaganych do dostarczenia systemu BI ze wszystkimi komponentami. Jak zauważono w rozdziale, nie lubimy być bardzo surowi w zakresie zasad metodologii, dlatego proponujemy dostosować je do swojego projektu za pomocą tych narzędzi, które mają dla Ciebie sens. W kolejnych rozdziałach zapoznamy się ze szczegółowymi informacjami na temat wszystkich komponentów BI, zaczynając od podstaw języka SQL, które ułatwią interakcję z bazą danych.

3. Podstawy SQL

Zanim zaczniesz korzystać z relacyjnych baz danych i prawdopodobnie jeszcze bardziej skomplikowanych rzeczy, musisz znać standardowy język używany do interakcji z nimi. Chociaż możesz obejść się bez znajomości języka SQL i pracy z bazami danych (pomyśl o programie Microsoft Access), prędzej czy później będziesz musiał się go nauczyć. W tej części zobaczymy wprowadzenie do SQL. Skoncentrujemy się tylko na podstawowych tematach, dzięki czemu nie zgubisz się w szczegółach, które nie są potrzebne w pierwszych etapach Twojego projektu. Ci, którzy mają już przyzwoite umiejętności SQL, mogą przejść do następnej części, ale nawet w takim przypadku zdecydowanie zalecamy zrobienie krótkiego przeglądu.

Co to jest SQL?

SQL oznacza ustrukturyzowany język zapytań. Jak sama nazwa wskazuje, jest to język używany do wysyłania zapytań do bazy danych, ale oprócz wykonywania zapytań ma wiele innych funkcji, co zobaczymy w tej części. W tej części zobaczymy wprowadzenie do SQL. Skoncentrujemy się tylko na podstawowych tematach, dzięki czemu nie zgubisz się w szczegółach, które nie są potrzebne w pierwszych etapach Twojego projektu. Jedną rzeczą, którą musimy wiedzieć o SQL, jest to, że nie jest on podobny do innych języków programowania, takich jak Java, C# i Python. Dzieje się tak, ponieważ jest to język deklaratywny, a nie proceduralny. Oznacza to, że zamiast opisywać obliczenia, które należy wykonać, aby uzyskać to, czego chcesz, po prostu określasz, czego chcesz, bez określania, jak to uzyskać. SQL został pierwotnie stworzony z myślą o modelu relacyjnym. Jednak późniejsze dodatki dodały funkcje, które nie są częścią modelu relacyjnego, więc w razie potrzeby możesz robić rzeczy w inny sposób. Wiele aktualnych wersji baz danych obsługuje zarówno metody relacyjne, jak i obiektowo-relacyjne, które obejmują obsługę programowania obiektowego i bardziej złożonych typów danych niż zwykłe podstawowe. Wkrótce zobaczymy, czym jest tryb relacyjny i jakie są jego implikacje; nie martw się. Tam, gdzie może się to wydawać nieco ograniczające i prawdopodobnie masz rację, ponieważ sam SQL prawdopodobnie nie wystarczy do wykonania wszystkich obliczeń, o których myślisz, zwłaszcza jeśli myślisz o aplikacjach dla dużych przedsiębiorstw, jest łatwy w użyciu w połączeniu z innymi językami , zwłaszcza te proceduralne. Prawie każdy język programowania ma jakieś złącze specyficzne dla bazy danych do interakcji lub używa jakiegoś rodzaju sterownika ODBC, który pozwala łączyć się z relacyjną bazą danych i wykonywać zapytania. Zróbmy krótkie wprowadzenie do historii SQL. SQL pojawił się po raz pierwszy w 1970 r.1, ale najnowsza wersja pochodzi z 2011 r. Nie wszystkie bazy danych mają tę samą wersję; nawet nie wszystkie z nich implementują całą wersję, ale czasami tylko jej podzbiór. Czasami powoduje to problemy podczas próby przeniesienia kodu, który działa dla określonej wersji bazy danych, do kodu pochodzącego od innego dostawcy.

Model relacyjny

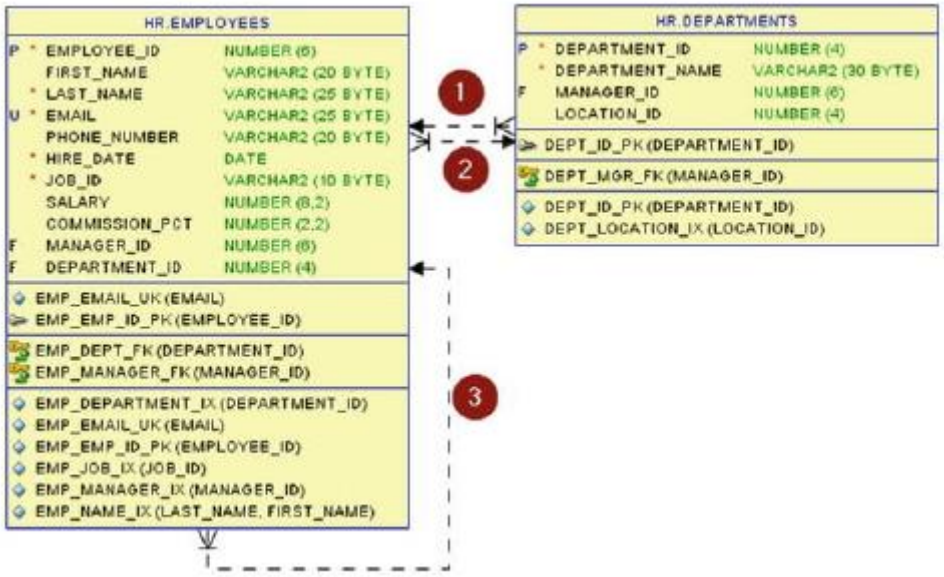
SQL jest standardowym językiem używanym do interakcji z relacyjnymi bazami danych, ale nie implementuje wszystkich koncepcji modelu relacyjnego. Po wielu latach nie ma jeszcze w pełni udanej implementacji modelu relacyjnego. Model relacyjny ma zasadniczo trzy koncepcje:

* Tabele, które służą do przedstawiania informacji oraz przechowywania kolumn i wierszy, których celem jest reprezentowanie atrybutów i rekordów (pomyśl o arkuszu kalkulacyjnym Excel).

* Klucze, aby skutecznie zarządzać i wdrażać relacje oraz jednoznacznie identyfikować rekordy w tabeli (więcej później, gdy omawiamy klucze podstawowe i obce).

* Operacje, które wykorzystują model i mogą generować inne relacje i obliczenia spełniające zapytania.

Aby pokazać przykład dotyczący tych pojęć, weźmy próbkę relacyjnej bazy danych; równie dobrze mogą to być typowe tabele Pracowników i Działów, wyjaśniające nieco niektóre z tych pojęć przy użyciu dwóch tabel ze schematu Oracle HR. Procedura ich instalacji polega zasadniczo na klonowaniu repozytorium github i uruchomieniu kilku poleceń, które uruchomią skrypt. Pełne instrukcje masz na stronie powitalnej repozytorium. Jednakże, jeśli chcesz przetestować, możesz także zdecydować się na pobranie bazy danych Oracle Enterprise Edition ze strony internetowej Oracle, która jest bezpłatna według strony internetowej właściciela, gdy tylko użycie ma na celu „opracowywanie, testowanie, prototypowanie i demonstrowanie twojej aplikacji, a nie w jakimkolwiek innym celu."



Na rysunku możesz zauważyć, że mamy dwie tabele. Pierwsza część wylicza kolumny tabel i ich typy danych (zobaczymy je za chwilę). Dla działów widzimy, że mamy kolumnę o nazwie DEPARTMENT_ID, składającą się z liczby 4 cyfr. „P" z przodu oznacza, że ta kolumna jest kluczem podstawowym. Następnie mamy kolumnę DEPARTMENT_NAME, która jest nazwą działu; MANAGER_ID to kolumna identyfikująca identyfikator pracownika (w tabeli Pracownicy), który jest kierownikiem działu, który jest kluczem obcym łączącym tabelę działów z tabelą pracowników i wymusza, aby kierownik działu musi być prawidłowym pracownikiem znalezionym w tabeli Pracownicy; oraz LOCATION_ID, który jest identyfikatorem lokalizacji działu (jak możesz pomyśleć, że może to być prawdopodobnie klucz obcy do „brakującej" tabeli lokalizacji. Tak, masz rację!). Kolumny, które są używane w relacjach z innymi tabelami lub identyfikują wartości tabeli, tutaj DEPARTMENT_ID i MANAGER_ID są zwykle przedstawiane osobno na diagramie, w postaci kluczy z nazwami ograniczeń, przedstawiającymi nazwę kolumny, do której się odnoszą w nawiasy. Również wszelkie indeksy znalezione w tabeli potrzebne do przyspieszenia zapytań lub zapewnienia prawidłowej implementacji relacji są również przedstawione w ostatniej części definicji tabeli. Są jeszcze inne ważne rzeczy, których można się nauczyć z tej liczby. Są to kropkowane linie między tabelami, które definiują relacje. To są:

1. DEPT_MGR_FK, czyli w zasadzie relacja, która mówi nam, że pracownik o numerze MANAGER_ID jest jedynym managerem z Tabeli Pracownicy. Ta relacja jest kluczem obcym i zwykle jest zgodna z konwencją nazywania jej z zakończeniem „_FK".

2. EMP_DEPT_FK, który jest w przeciwnym kierunku, mówi nam, że DEPARTMENT_ID znaleziony w tabeli Pracownicy jest jednym i jedynym DEPATMENT_ID znalezionym w tabeli departamentów.

3. EMP_MANAGER_FK. Jest to jak dotąd najbardziej skomplikowane do zrozumienia, ale definiuje relację menedżera, tak że KIEROWNIK pracownika musi być jednocześnie członkiem tabeli PRACOWNIK, co uniemożliwia przypisanie menedżera, który nie jest częścią tabeli PRACOWNIK przy stole, czy w naszym świecie domenowym, pracownik spoza firmy.

To jest zrozumiałe, czyli w zasadzie wszystko, co musimy wiedzieć o modelu relacyjnym. Oczywiście temat jest znacznie szerszy, ale dla naszych celów zaufaj nam, że to w zasadzie wszystko, co musisz wiedzieć.

Bazy danych i dostawcy baz danych

Istnieje wiele typów baz danych: relacyjne, obiektowe, bez SQL… ale generalnie hurtownie danych zostały utworzone z relacyjnymi bazami danych. Wynika to z faktu, że łatwo jest nauczyć się języka SQL i przeprowadzać z nim analizy. W dzisiejszych czasach wiele systemów, zwłaszcza tych, które zajmują się dużymi ilościami danych, ma tendencję do korzystania z innego typu baz danych, baz danych NoSQL, które są inne i zwykle używają pary klucz/wartość jako sposobu przechowywania danych. Podczas gdy te bazy danych mogą być bezschematowe i bardziej praktyczne, nie przestrzegają zasad ACID dla transakcji, których przestrzega większość relacyjnych baz danych. Jest to kompromis, który trzeba zapłacić, aby zyskać prędkość i móc zapewnić lepszą wydajność podczas pracy z dużą ilością danych oraz móc działać w sposób rozproszony.

Uwaga: obecnie NoSQL i specjalnie dystrybuowane bazy danych są popularnym tematem wraz z nadejściem BigData i faktycznie używają innych języków niż ich relacyjne odpowiedniki. Ale nawet przy tym znajdziesz wiele podobieństw między nimi a językiem SQL. Tak więc, pomimo tego, że myślisz o zaimplementowaniu bazy danych NOSQL do swojego projektu lub innego typu bazy danych, ten rozdział powinien być wart przeczytania.

Głównymi dostawcami baz danych jest firma Oracle, która twierdzi, że ma udział w rynku wynoszący 98,5% firm z listy Fortune 500. Choć może się to wydawać zdumiewające, należy podkreślić, że zazwyczaj duże firmy mają różnych dostawców baz danych. Prawdopodobnie łatwo jest zobaczyć duże firmy posiadające Oracle, SQL Server, a nawet MySQL, Postgree i bazy danych typu open source, takie jak MariaDB i wiele innych. DB2 firmy IBM i SQL Server firmy Microsoft to również ważne komercyjne relacyjne bazy danych, na które należy zwrócić uwagę. Znajdziesz je w wielu firmach choć pewnie nie są tak rozbudowane jak Oracle. Warto zauważyć, że wszystkie komercyjne bazy danych mają zwykle jakąś zmniejszoną lub ograniczoną wersję do bezpłatnego użytku, a nawet z pozwoleniem na dystrybucję udzielonym przez ich własną firmę. Wersje te mają zwykle pewne ograniczenia, ale jako mała lub średnia firma nie powinno to stanowić dla Ciebie problemu i zawsze możesz przejść na wersję płatną, jeśli Twoja firma się rozwinie. Jeśli odejdziesz od komercyjnych baz danych, Oracle ma inny RDBMS, który jest open source, o nazwie MySQL. Chociaż MySQL został zakupiony przez Oracle, nadal możesz go pobrać i używać za darmo. Tę bazę danych zobaczysz głównie w wielu projektach internetowych, zwłaszcza, że łatwiej jest ją zintegrować z językami internetowymi, takimi jak php i tym podobne. Istnieją inne bardzo interesujące bazy danych typu open source. Jednym z nich jest postgresql, który w dzisiejszych czasach zyskuje coraz większą popularność. Kolejnym jest rozwidlenie MySQL, zwane MariaDB, które było kontynuowane przez głównych ludzi, którzy stworzyli MySQL i których projekt rozpoczął się, gdy Oracle kupił oryginalną bazę danych. Jedną z najlepszych rzeczy MariaDB jest to, że jest w 100% kompatybilna z MySQL. Od kilku lat do MariaDB dodano różne i nowe funkcje, które odbiegają od oryginalnego MySQL. Te zwykle nie mają ograniczeń, tylko te wynikające z ograniczeń implementacji i architektury. Wszystkie te bazy danych mogą współpracować z wieloma systemami operacyjnymi, w tym Linux i Windows w głównych częściach ich odpowiednich wersji.

Uwaga: Wybierając bazę danych do realizacji swojego projektu, nie kieruj się wyłącznie pieniędzmi, ale również umiejętnościami swoich pracowników. Oracle jest bardzo dobry i wersja Express może ci się przydać, ale jeśli z jakiegoś powodu po jakimś czasie będziesz musiał przejść na edycje płatne, licencje nie są tanie. Migracja bazy danych od jednego dostawcy do innego może być uciążliwa, jeśli Twój system jest duży lub Twoje aplikacje korzystają z określonych funkcji jednego dostawcy. W takich przypadkach możesz rozważyć skorzystanie z rozwiązania w chmurze z płatną bazą danych, w której cena może być kontrolowana. Bazy danych typu open source są również bardzo dobrą alternatywą.

Zgodność z ACID

Relacyjne bazy danych zwykle obsługują tak zwany zestaw właściwości ACID, jeśli chodzi o transakcje. Baza danych zgodna z ACID oznacza, że gwarantuje:

* Niepodzielność, co oznacza, że instrukcje mogą grupować się w transakcje, co oznacza, że możesz kontrolować, że jeśli jedna się nie powiedzie, transakcja się nie powiedzie, co oznacza, że żadne zmiany nie zostaną zapisane w bazie danych.

* Spójność, co oznacza, że baza danych jest zawsze w prawidłowym stanie, nawet po awarii lub nieprzestrzeganiu ograniczeń. Zwykle najłatwiejszym sposobem myślenia o tym jest tabela z ograniczeniem, które sprawdza, czy żaden pracownik nie może mieć pensji większej niż 10 000 USD. Jeśli zaczniesz wprowadzać pracowników i wynagrodzenia do tej tabeli jako część transakcji, jeśli dodasz jedną z pensją 20 000 USD, cała transakcja powinna zakończyć się niepowodzeniem, a dane tabeli powinny zostać przywrócone tak, jak przed rozpoczęciem tej transakcji. To samo dotyczy przypadku niepowodzenia podczas realizacji transakcji. Albo transakcja została zatwierdzona i utrwalona, lub jeśli nie została jeszcze ukończona, zostanie wycofana.

* Izolacja, co oznacza, że za każdym razem, gdy wysyłasz zapytanie do bazy danych, pobierane jest to, co było przed rozpoczęciem jakiejkolwiek uruchomionej transakcji lub po zakończeniu dowolnej uruchomionej transakcji. Pomyśl na przykład, że masz duży stół i wysłałeś do bazy danych instrukcję aktualizacji wynagrodzeń pracowników, aby ustalić dwukrotność ich wynagrodzenia. Jeśli transakcja nie została jeszcze zakończona podczas zapytania do tabeli, powinna zostać wyświetlona poprzednia wersja tabeli, w której tabela nie została jeszcze zaktualizowana. Po zakończeniu transakcji wszelkie kolejne zapytania do tabeli powinna zostać wyświetlona zaktualizowana tabela, w której pracownicy mają podwójną pensję, ale generalnie nie jest pożądane uzyskiwanie miksu, ponieważ prowadzi to do błędów.

* Trwałość oznacza, że raz zatwierdzona transakcja jest zatwierdzona na zawsze, nawet w przypadku niepowodzenia. Tak więc baza danych musi zapewnić, dzięki różnym mechanizmom, że zmiany są zawsze zapisywane na dysku, albo poprzez bezpośrednie przechowywanie danych (zwykle nieefektywne), albo przez przechowywanie pliku wektorowego definiującego zmiany zastosowane do danych, aby baza danych po przywróceniu , jest w stanie odtworzyć te zmiany.

Rodzaje instrukcji SQL

W tym momencie prawdopodobnie już zastanawiasz się, co zrobisz, aby sprawdzić sumę pieniędzy zafakturowanych dla konkretnego klienta lub określonego regionu. To dobrze, a wkrótce nauczysz się, jak to robić. Jednak podczas pracy z relacyjną bazą danych istnieją inne stwierdzenia, być może równie ważne jak zapytania. Oprócz wysyłania zapytań do bazy danych, istnieją inne rodzaje instrukcji, które są zwykle używane przed zapytaniem, aby przygotować bazę danych do przechowywania danych, które będą później wyszukiwane. Zróbmy szybkie podsumowanie i przejrzyjmy je:

* Instrukcje DML lub Data Manipulation Language. Należy zauważyć, że ta grupa zawiera instrukcje Select do wykonywania zapytań w tabelach, ale nie ogranicza się tylko do zapytań, ponieważ istnieją inne zestawy instrukcji, które należą do tej grupy, takie jak INSERT, używane do wstawiania rekordów w tabeli; UPDATE do aktualizacji tych zapisów; i DELETE, aby usunąć rekordy na podstawie jednego lub wielu warunków. W wielu miejscach zobaczysz komunikat MERGE. Jest to specyficzna instrukcja, która nie została zaimplementowana we wszystkich bazach danych i jest mieszanką INSERT + UPDATE. Czasami jest to również nazywane i UPSERT z tego powodu.

* Instrukcje DDL lub Data Definition Language, które obejmują instrukcje używane do modyfikowania struktury tabeli, tworzenia ich lub usuwania. Należą do nich instrukcja CREATE do tworzenia obiektów, takich jak tabele i indeksy, ale także wiele innych, instrukcja DROP do ich usuwania, ALTER do ich modyfikowania oraz specjalna instrukcja o nazwie TRUNCATE używana do usuwania wszystkich danych z tabeli.

* Wyciągi transakcji, takie jak COMMIT, aby zatwierdzić wszystkie zmiany dokonane w ramach transakcji, ROLLBACK, aby cofnąć wszystkie wprowadzone zmiany. Należy zauważyć, że jeśli polecenie COMMIT nie zostało wykonane na końcu transakcji, po rozłączeniu, ze względu na mechanizmy spójności zaimplementowane przez prawie wszystkie bazy danych, zmiany zostaną utracone i nie zostaną utrwalone w plikach bazy danych. Niektóre bazy danych implementują również polecenie CHECKPOINT, które wyznacza punkty kontrolne w bazie danych, ale obecnie nie jest to dla nas zbyt przydatne.

Uwaga: Należy zachować ostrożność w przypadku opcji automatycznego zatwierdzania, która jest domyślnie włączona w niektórych relacyjnych bazach danych, takich jak MySQL i SQL Server. Może to być dobre, jeśli zapomnisz zatwierdzić transakcję na koniec transakcji, ale może mieć straszne konsekwencje, jeśli zepsujesz ją instrukcją aktualizacji lub usunięcia. Naszą sugestią jest wyłączenie automatycznego zatwierdzania i pamiętaj, aby zawsze zatwierdzać. Instrukcja TRUNCATE, ponieważ wszystkie instrukcje DDL mają niejawne polecenie COMMIT. Jeśli przez pomyłkę obetniesz tabelę, nie ma możliwości jej odzyskania, chyba że przywrócisz bazę danych do poprzedniego punktu kontrolnego. Podczas gdy w przypadku niektórych baz danych możliwe jest nawet odzyskanie aktualizacji lub usunięcia wprowadzonych przez pomyłkę, nie jest to możliwe w przypadku obciętych baz danych. Bądź bardzo ostrożny podczas uruchamiania instrukcji TRUNCATE!

Typy danych SQL

Język SQL definiuje zestaw typów danych dla wartości kolumn. Nie wszystkie bazy danych implementują te same typy danych i nie wszystkie bazy danych używają tej samej nazwy dla zgodnych typów danych. Większość baz danych obsługuje jednak najpopularniejsze typy danych. W kolejnych podrozdziałach dokonujemy ich przeglądu, wyjaśniamy, kiedy ich używać i analizujemy ich specyfikę. Ważne jest, aby użyć poprawnego typu danych, ponieważ wybranie nieprawidłowego typu, oprócz marnowania miejsca, może prowadzić do niskiej wydajności z powodu jawnych lub nawet niejawnych konwersji typów.

Liczbowe typy danych

Istnieje wiele liczbowych typów danych. W większości baz danych są one reprezentowane przez typy Integer, Float, Real, Decimal i BigInt. Istnieją inne typy danych, takie jak Smallint dla małych liczb. Wszystkie one są używane do przechowywania liczb i musisz wybrać, która jest bardziej odpowiednia

dla każdej kolumny tabeli, zasadniczo biorąc pod uwagę maksymalną wartość, jaką można przejść, oraz czy potrzebne są pozycje dziesiętne i do jakiego stopnia precyzja jest dla Ciebie ważna.

Tekstowe typy danych

Tekst jest zwykle reprezentowany przez typy danych Char (stała długość) lub nowsze typy danych Varchar (zmienna długość). Typy danych String i CLOB (Character LOB) są również dostępne w innych bazach danych. Zalecamy trzymanie się typu danych Varchar, chyba że istnieje bardzo ważny powód, aby używać innych typów danych, ponieważ zazwyczaj nowe bazy danych nie marnują miejsca na przechowywanie mniejszej wartości w kolumnie varchar zdefiniowanej do przechowywania większej liczby znaków.

Typy danych daty

Wszystkie wartości daty, godziny i znacznika czasu należą do tej kategorii. Prawdopodobnie tutaj pojawia się najwięcej różnic w stosunku do różnych dostawców baz danych. Ważne jest, aby sprawdzić dokumentację swojej bazy danych, aby upewnić się, że używasz właściwego typu. Większość baz danych obsługuje również znaczniki czasu ze strefami czasowymi, więc jeśli planujesz obsługiwać różne regiony w swojej aplikacji i chcesz korzystać z tych różnych stref czasowych, przed wybraniem typu danych zapoznaj się z instrukcją.

Inne typy danych

W tym obszarze możemy znaleźć inne ważne typy danych, takie jak typ danych binarny, bool lub bitowy używany do przechowywania informacji binarnych (prawda lub fałsz); typy danych LOB lub RAW używane do przechowywania dużych informacji, takich jak duże fragmenty tekstów, obrazów lub plików w formacie binarnym; typy XML do przechowywania informacji XML w obsługujących je bazach danych, dzięki czemu można wydajnie wyszukiwać dane; typy zdefiniowane przez użytkownika poprzez łączenie typów podstawowych; oraz wiele innych typów danych, które nie mieszczą się w żadnej z poprzednich kategorii.

Pobieranie danych z tabeli

Kilka akapitów temu przedstawiliśmy tabele Pracownicy i Działy. Wykorzystamy je w tej części do wykonania kilku instrukcji i sprawdzenia oczekiwanych rezultatów. Zaczniemy od najbardziej podstawowej operacji, jaką można wykonać na tabeli, za pomocą instrukcji SELECT. Najpierw chcemy pokazać przegląd wybranej instrukcji. W kolejnych podrozdziałach przyjrzymy się każdemu blokowi po kolei i omówimy jego zastosowanie. Wszystkie typowe instrukcje wyboru SQL mają następującą postać:

SELECT list_of_columns (separated by commas, or *)

FROM list_of_tables (separated by commas)

[WHERE set_of_conditions (separated by operators)]

[GROUP BY grouping_of_columns (separated by

commas)]

[HAVING

set_of_conditions_applied_to_the_grouping_of_columns

(separated by operators)]

[ORDER BY ordering_conditions (separated by

commas)] ;

Jeśli to jest wystarczająco jasne, możemy zacząć od uzyskania wszystkich informacji z tabeli za pomocą specjalnej instrukcji SELECT *, a następnie zobaczymy inne sposoby użycia opcji z instrukcją Select i użyjemy projekcji, aby wybrać tylko określone kolumny z bazy danych.v

Instrukcja *SELECT

Składnia instrukcji select jest bardzo prosta. Zwykle musimy określić, z jakich kolumn chcemy pobrać informacje i z której tabeli lub tabel, a następnie opcjonalny predykat w klauzuli where, aby zastosować dowolny filtr w danych, których chcemy użyć, aby wiersze, które nie pasują do tych warunki zostaną odfiltrowane z wyniku. Typowa instrukcja select wygląda następująco:

SELECT

column1, column2, columnn

FROM

schema.table1

WHERE

column1 = 'Albert';

Wybrane kolumny są częścią klauzuli projekcji. Kolumny w klauzuli where są filtrami lub kolumnami używanymi do łączenia z innymi tabelami. Zobaczmy najpierw najbardziej podstawowe stwierdzenie:

```
   SELECT
         *
   FROM
         hr.departments;

   DEPARTMENT_ID
DEPARTMENT_NAME                MANAGER_ID LOCATION_ID
   -------------- ------------------------------ ------
---- -----------
             10
Administration                      200      . 1700
             20
Marketing                           201        1800
             30
Purchasing                          114        1700
             40 Human
Resources                      203        2400
             50
Shipping                            121        1500
             60
IT                                  103        1400
   ... (output truncated)
   27 rows selected
```

Za pomocą tej instrukcji select * mówimy bazie danych, aby pobierała wszystkie kolumny i wszystkie dane z tabeli departamentów bez stosowania żadnego filtra. Ponieważ pobieramy wszystkie dane, zostanie wykonane PEŁNE SKANOWANIE tabeli, co oznacza, że cała tabela zostanie odczytana przez bazę danych i zwrócona. Żaden indeks, bez względu na to, czy istniał, nie zostanie użyty do odzyskania tego, chyba że zdarzy się nietypowa sytuacja, w której wszystkie kolumny tabeli będą znajdować się w indeksie (więcej na ten temat w dalszej części rozdziału dotyczącego wydajności).

Instrukcja select column

Instrukcja select column jest szczególnym przypadkiem instrukcji select *. W takim przypadku używasz predykatu projekcji, aby wybrać tylko te kolumny, które Cię interesują. Jeśli to możliwe, używaj go zamiast select *, ponieważ ten ostatni jest bardzo niebezpieczny. Pomyśl, co może się stać, jeśli dodasz kolumnę do tabeli. Teraz wybór zamiast zwracania n kolumn zwróci n+1. To samo, jeśli usuniesz niektóre. Na przestrzeni lat widzieliśmy wiele błędów w procesach polegających na używaniu gwiazdki zamiast nazywania kolumn. Również z punktu widzenia wydajności użycie select * nie jest zalecanym rozwiązaniem, ponieważ uniemożliwisz optymalizatorowi użycie indeksów, jeśli są dostępne, ale zostanie to omówione później. Typowa instrukcja select służąca do pobierania nazw działów z tabeli departamentów jest następująca:

SELECT

department_name

FROM

hr.departments;

DEPARTMENT_NAME

Administration

Marketing

Purchasing

Human Resources

Shipping

IT

..... (output truncated)

27 rows selected

Instrukcja select count (*) lub count (column).

Podczas opracowywania możemy być zainteresowani łatwym policzeniem liczby wierszy zwracanych przez zapytanie. Można to osiągnąć za pomocą instrukcji select count. Jeśli użyjesz gwiazdki, policzy wszystkie wiersze z tabeli, ale z kolumną pominie wartości NULL, czyli wartości, które nie istnieją dla tego konkretnego wiersza. Zdania, w przypadku gdy istnieją wszystkie wartości, powinny dawać ten sam wynik, ale różne bazy danych implementują się wewnętrznie inaczej, a niektóre z nich mogą mieć lepszą wydajność niż inne.

SELECT

COUNT(department_name)

FROM

hr.departments;

COUNT(DEPARTMENT_NAME)

27

Whereas:

SELECT

COUNT(*)

FROM

hr.departments;

COUNT(*)

27

Otrzymujemy więc dokładnie ten sam wynik.

Wybierz odrębną klauzulę

Gdy nie chcemy brać pod uwagę odrębnych wartości dla określonego wyboru, możemy użyć klauzuli odrębnej. Spowoduje to pobranie tylko jednego na parę kolumn. Jeśli wybierzemy tylko jedną kolumnę, pobierzemy tylko różne wartości dla tej kolumny; ale jeśli zapytanie składa się z projekcji kilku kolumn, zwróci różne krotki, co oznacza, że zwróci różne wartości, ale biorąc pod uwagę wszystkie wybrane kolumny. Oto przykład:

SELECT

COUNT(first_name)

FROM

hr.employees;

FIRST_NAME

107

But, for example:

SELECT

COUNT(distinct first_name)

FROM

hr.employees;

COUNT(DISTINCTFIRST_NAME)

91

Jeśli jednak zapytamy parę imię, nazwisko, zobaczymy, że nie ma pracowników o dokładnie takim samym nazwisku (imię + nazwisko są takie same):

```
SELECT DISTINCT

first_name, last_name

FROM

hr.employees;

FIRST_NAME LAST_NAME

-------------------- -------------------------

Ellen Abel

Sundar Ande

Mozhe Atkinson

... (output truncated)

107 rows selected
```

Sortowanie

Czasami chcesz, aby wynik zapytania był sortowany według określonej kolumny. Sortowanie może odbywać się rosnąco lub malejąco. W SQL osiąga się to za pomocą słów kluczowych ORDER BY. Klauzula ORDER BY zawsze znajduje się na końcu i sortuje wszystkie rekordy, które zostały wybrane i nie zostały odfiltrowane. Rzućmy okiem na kilka przykładów: Jeśli nie określono, domyślnie kolejność sortowania jest rosnąca:

```
SELECT

department_name

FROM

hr.departments

ORDER BY

department_name;

DEPARTMENT_NAME

------------------------------

Accounting

Administration

Benefits

Construction

... (output truncated)

27 rows selected
```

Uwaga: Domyślnie sortowanie jest numeryczne w przypadku pól liczbowych, alfabetyczne w przypadku pól tekstowych i chronologiczne podczas sortowania pola daty. Ale możemy określić kierunek sortowania:

```
SELECT
department_name
FROM
hr.departments
ORDER BY
department_name desc;
```

```
DEPARTMENT_NAME
------------------------------
Treasury
Shipping
Shareholder Services
Sales
... (output truncated)
27 rows selected
```

Możemy sortować wiersze według wielu kolumn jednocześnie, więc pierwsza kolumna w klauzuli zostanie posortowana, potem druga i tak dalej:

```
SELECT
first_name, last_name
FROM
hr.employees
ORDER BY
first_name, last_name;
```

```
FIRST_NAME LAST_NAME
-------------------- -------------------------
Adam Fripp
Alana Walsh
Alberto Errazuriz
Alexander Hunold
Alexander Khoo
```

... (output truncated)

107 rows selected

Możemy nawet zdecydować się na sortowanie niektórych kolumn rosnąco, a innych malejąco:

```
SELECT
first_name, last_name
FROM
hr.employees
ORDER BY
first_name asc, last_name DESC;
```

```
FIRST_NAME LAST_NAME
-------------------- -------------------------
Adam Fripp
Alana Walsh
Alberto Errazuriz
Alexander Khoo
Alexander Hunold
... (output truncated)
107 rows selected
```

Czasami jednak łatwiej jest po prostu określić pozycję kolumny w predykacie projekcji, więc poniższy przykład będzie podobny do poprzedniego przykładu:

```
SELECT
first_name, last_name
FROM
hr.employees
ORDER BY 1 ASC, 2 DESC;
```

```
FIRST_NAME LAST_NAME
-------------------- -------------------------
Adam Fripp
Alana Walsh
Alberto Errazuriz
Alexander Khoo
```

Alexander Hunold

... (output truncated)

107 rows selected

Jeśli mamy w kolumnach jakieś wartości null i chcemy je przenieść na początek lub na koniec, możemy użyć NULLS LAST:

SELECT

First_name, last_name, commission_pct

FROM

hr.employees

ORDER BY

3 DESC NULLS LAST,1,2;

FIRST_NAME LAST_NAME COMM

ISSION_PCT

-------------------- ------------------------- ---- ----------

John Russell

,4

Allan McEwen

,35

Janette King

,35

Patrick Sully

,35

Alberto Errazuriz

,3

... (output truncated)

Vance Jones

William Gietz

Winston Taylor

107 rows selected

I to na razie wszystko, co musimy wiedzieć o sortowaniu.

Uwaga: Oprócz porządkowania, relacyjne bazy danych zwykle implementują pewne predykaty, aby wybrać górne X wierszy. Może to być bardzo przydatne w środowisku hurtowni danych, jeśli chcesz

wyświetlić 5 największych klientów lub 10 regionów z największą sprzedażą. W zależności od bazy danych jest to realizowane jako klauzule TOP, LIMIT, FETCH FIRST X ROWS ONLY; ponieważ nie ma jasnego standardu, nie będziemy go tutaj omawiać, ale zachęcamy do sprawdzenia instrukcji dostawcy, aby uzyskać więcej informacji.

Filtracja

Do tej pory widzieliśmy kilka podstawowych zapytań, ale nie są one jeszcze zbyt przydatne. Po prostu wybieramy wszystkie rekordy z tabeli. Ale zwykle chcesz wybrać tylko niektóre rekordy, które spełniają warunek. Obliczenie tego warunku lub zestawu warunków może być czasem skomplikowane. Przez większość czasu będziesz pytać o swoje dane, ile wystawiłem faktury za dany region? Kim są pracownicy pracujący w części łańcucha dostaw firmy? Jaki procent zysku mam dla określonej grupy produktów? Wszystkie te pytania wymagają pewnego rodzaju filtrowania, ponieważ kierujesz obliczenia do określonej grupy.

Klauzula Where

Predykatem do filtrowania wierszy, które zostaną uwzględnione w wyniku, jest klauzula WHERE. Istnieje jednak ogromny zestaw operatorów, z których można korzystać. Przejrzyjmy je. Jeśli pamiętasz z poprzednich przykładów, mieliśmy w tabeli 107 pracowników, zobaczmy, jak obliczyć pracowników, którzy zarabiają więcej niż 10 000 USD za pomocą zapytania. Zapytanie będzie wyglądać następująco, wykorzystując to, czego się do tej pory nauczyliśmy:

SELECT

count(*)

FROM

hr.employees

WHERE

salary>10000;

15 rows selected

COUNT(*)

15

Możemy więc teraz wywnioskować, że mamy 15 pracowników na 107, którzy zarabiają więcej niż $10,000$.

Operatory

W naszym przykładzie użyliśmy operatora większego niż, aby porównać wartość wynagrodzenia w tabeli z określoną liczbą. Ale jest wielu innych operatorów. Oto lista najczęściej używanych:

Operator : Znaczenie : Przykład

= : Równe wynagrodzenie : = 10000

< : Mniej niż pensja : < 10000

<= : Mniejsze lub równe Wynagrodzeniu : <=10000

> : Większa niż pensja : > 10000

>= : Większe lub równe Wynagrodzeniu : >= 10000

<> : Inaczej niż Wynagrodzenie : <> 10000

IN () : Jedna z wartości na liście : Wynagrodzenie w (10000, 11000, 12000)

NOT IN () : Żadna z wartości na liście : Wynagrodzenie nie w (10000, 11000,12000)

BETWEEN x i y : Pomiędzy tym zakresem : Wynagrodzenie między 10000 i 11000

LIKE : Dopasowuje częściowe słowo (tekst). Pobiera wszystkich pracowników, których imię to Pete, Peter lub PeteXXX. Symbol wieloznaczny % będzie pasował do dowolnego znaku (znaków): Imię, na przykład „Pete%"

NOT LIKE : Wyklucz częściowe dopasowania: Imię inne niż „Pete%"

IS NULL : Określona kolumna ma wartość NULL : Wynagrodzenie ma wartość NULL

IS NOT NULL : Określona kolumna ma wartość : Wynagrodzenie nie jest puste

Uwaga: pamiętaj, że jeśli porównujesz z ciągiem tekstowym lub znakiem, musisz użyć pojedynczych cudzysłowów, aby ująć ciąg lub znak, z którym porównujesz. Poniższe porównanie nie jest poprawnym porównaniem: gdzie imię = Piotr. Właściwym sposobem jest ujęcie tekstu w pojedyncze cudzysłowy: gdzie imię = „Piotr".

Oprócz nich istnieje kilka innych operatorów, które są podobne do operatora IN(). Są to operatory, które porównują grupę wierszy: ANY, ALL, EXISTS, a nazwy są oczywiste. W podrozdziale dotyczącym podzapytań zobaczymy pewne zastosowanie.

Operatory logiczne

Istnieje inny zestaw operatorów zwanych operatorami logicznymi. Ten zestaw operatorów jest szeroko stosowany, ponieważ działają one jak klej między różnymi warunkami lub innymi operatorami. Dla czytelników przyzwyczajonych do programowania w dowolnym języku wyniki te będą bardzo znajome

Operator : Znaczenie: Przykład

AND: Oba warunki muszą być spełnione, aby zwróciły wartość true: Wynagrodzenie > 10000 AND imię, takie jak „Pete%"

OR: Przynajmniej jeden z warunków musi być spełniony, aby zwrócić wartość true:

Wynagrodzenie > 10000 OR imię jak „Pete%"

NOT: Warunek musi być fałszywy, aby zwrócił prawdę: NOTNIE (wynagrodzenie > 10000)

Jeśli pamiętasz ze szkoły, musisz znać pierwszeństwo operatorów. Oznacza to, że czasami musimy użyć nawiasów, aby określić, który operator występuje jako pierwszy. Na przykład w operatorze NOT określiliśmy nawiasy, aby nakazać bazie danych wykonanie najpierw porównania Wynagrodzenie > 10000, a następnie operatora NOT. W tym przypadku nie jest to konieczne, ponieważ domyślnie najpierw stosowane są operatory porównania, a następnie operator logiczny NOT. Ale musisz znać

zasady pierwszeństwa. W każdym razie zalecamy używanie nawiasów, jeśli to możliwe, ponieważ kod jest znacznie bardziej przejrzysty i można łatwo wiedzieć, które warunki należy sprawdzić w pierwszej kolejności, i uniknąć błędów. Możemy zagwarantować, że zaoszczędzi to dużo czasu na debugowaniu źle działających zapytań. Zwykle najpierw wykonywane są operacje arytmetyczne. Tak więc dodawanie, odejmowanie, mnożenie lub dzielenie zostanie wykonane przed jakimkolwiek innym warunkiem. Później stosowane są operatory porównania. Po nich LIKE, IN i NULL. Następnie instrukcja Between, następnie operator porównania <>, a na samym końcu operatory NOT, AND i OR w tej kolejności. Pozostawienie tych operatorów logicznych na koniec ma sens, ponieważ, jak wyjaśniliśmy wcześniej, są one używane głównie jako łącznik między innymi operatorami lub grupami warunków. Jeśli dotarłeś do tego momentu, to jest to bardzo dobra wiadomość. Nauczyłeś się prawdopodobnie najważniejszej części pisania zapytań. To oczywiście jeszcze nie wystarczy, ponieważ sprawy mogą się znacznie skomplikować, a pisanie dobrych, wydajnych i przejrzystych zapytań SQL wymaga trochę czasu. Ale jesteśmy na dobrej drodze.

Grupowanie danych

Czasami nie chcesz pobierać pojedynczych wierszy z bazy danych. Chcesz pogrupować według określonych danych i pobrać tylko sumy, liczbę grup, średnie i tak dalej. Jest to bardzo typowe w środowisku hurtowni danych. Pomyśl o menedżerze, który chce odzyskać sprzedaż z określonego dnia. W zależności od wielkości Twojej firmy pobieranie danych sprzedaży pojedynczo nie będzie miało sensu, ponieważ prawdopodobnie bardziej interesuje Cię konkretny szczegół niż weryfikacja wszystkich zakupów klientów. W takich przypadkach musimy mieć możliwość grupowania rekordów według określonych grup, aw SQL osiąga się to za pomocą klauzuli GROUP BY. Istnieje lista funkcji agregujących lub zestawów, których również musisz się nauczyć. Określają one operację, którą chcesz obliczyć na grupie.

Operator : Znaczenie : Przykład

MAX () : Zwraca wartość maksymalną zestawu. : MAKS (wynagrodzenie)

MIN () : Zwraca minimum zestawu. : MIN (pensja)

SUMA () : Zwraca sumę lub sumę częściową zestawu. : SUMA (wynagrodzenie)

AVG () : Zwraca średnią zestawu. : średnia (wynagrodzenie)

COUNT () : Zlicza rekordy. Możesz również określić warunek wewnątrz. : LICZBA (pensja > 10000)

Aby móc uwzględnić kolumny w klauzuli group by, musimy upewnić się, że znajdują się one również w części projekcyjnej zapytania. Oznacza to, że nie możemy dodać kolumny w klauzuli select, że nie ma jej w klauzuli group by. Niektóre bazy danych pozwalają to zrobić, modyfikując niektóre parametry, ale zwykle jest to coś, czego chcemy uniknąć, więc miej to na uwadze. Wiedząc o tym, możemy obliczyć np. średnią wynagrodzeń pracowników według ich działów, aby zobaczyć, w którym dziale firmy najlepiej pracować, jeśli oczywiście zależy nam tylko na pieniądzach!

```
SELECT
      department_id, trunc(avg(salary))
FROM
      hr.employees
GROUP BY
      department_id
ORDER BY


      2 DESC;

DEPARTMENT_ID                      TRUNC(AVG(SALARY
))
------------ ------------------------------------
--
          90                                   193
33
         110                                   101
50
          70                                   100
00
... (output truncated)
          10                                    44
00
          30                                    41
50
          50                                    34
75
12 rows selected
```

Jest więc jasne, że osoby przypisane do działu o id = 90 są wyraźnie tymi, które zarabiają więcej, podczas gdy te przypisane do działu o id = 50 otrzymują niższe odcinki wypłat. Można to łatwo wytłumaczyć, ponieważ dział o id = 90 to dział Kierownictwa.

Uwaga: Zapomnij na chwilę o używaniu funkcji TRUNC() . Dodaliśmy je, aby uzyskać wyraźny wynik i nie mieć wyników z dużymi miejscami po przecinku. W zależności od każdej bazy danych ta funkcja ma różne nazwy, ale w zasadzie instruuje bazę danych, aby pozbyła się części dziesiętnej liczby poprzez obcięcie danych wyjściowych. Uwaga, to nie oznacza tego samego, co Runda. Jeszcze raz; sprawdź instrukcję sprzedawcy, aby uzyskać więcej informacji.

Oczywiście czasami to nie wystarcza i chcesz przefiltrować konkretną grupę, bo np. interesują Cię tylko te, które mają co najmniej 5 pracowników. Wymaga to użycia klauzuli HAVING, tak jakbyś myślał, że nie ma możliwości filtrowania tego za pomocą klauzuli where. Kluczową kwestią jest tutaj zrozumienie, że każdy filtr w klauzuli Where filtruje wiersze, ale w tym przypadku musimy odfiltrować grupy z danych wyjściowych, a nie pojedyncze wiersze. Potrzebna jest więc klauzula posiadania. Oto przykład: Po pierwsze, liczba pracowników na dział:

```
SELECT
      department_id, COUNT(*)
FROM
      hr.employees
GROUP BY
      department_id
ORDER BY
      2 DESC;

DEPARTMENT_ID   COUNT(*)
------------- ----------
           50         45
           80         34
          100          6
           30          6
           60          5
           90          3
           20          2
          110          2
           40          1
           10          1
                       1
           70          1
12 rows selected
```

A następnie spójrzmy na nasze działy, które mają przydzielonych więcej niż 5 pracowników:

```
SELECT
      department_id, COUNT(*)
FROM
      hr.employees
GROUP BY
      department_id


HAVING
      COUNT(*) > 5
ORDER BY 2 DESC;

DEPARTMENT_ID   COUNT(*)
------------- ----------
           50         45
           80         34
           30          6
          100          6
 4 rows selected
```

Ważne jest również, aby zrozumieć, że klauzula „having” jest stosowana po zastosowaniu wszystkich filtrów w klauzuli „where”. Tak więc w poprzednim przykładzie, jeśli odfiltrujemy pracowników, którzy zarabiają więcej niż 9 000, a następnie usuniemy grupy, które mają mniej niż 5 pracowników, wynik może być inny, ponieważ niektórzy pracownicy mogli już zostać usunięci z grup na podstawie warunku Wynagrodzenie > =9.000, wpływając na całkowitą liczbę pracowników dla tej konkretnej grupy. Zobaczmy przykład:

```
SELECT
      department_id, COUNT(*)
FROM
      hr.employees
WHERE
      salary < 9000
GROUP BY
      department_id
HAVING
      COUNT(*) >= 5
ORDER BY 2 DESC;

DEPARTMENT_ID   COUNT(*)
------------- ----------
           50         45
           80         17
           30          5
3 rows selected
```

Jak więc widać, straciliśmy dwie grupy. Wynika to z faktu, że id_działu 100 i 60 miały co najmniej dwóch i jednego pracownika, z których każdy zarabiał co najmniej 9 000 USD. Ponadto działy 80 i 30 zmniejszyły liczbę członków spełniających warunek wynagrodzenia <9000 USD, zmieniając wynik zapytania.

Korzystanie z podzapytań

Wiele razy chcesz określić filtr, ale ten filtr jest obliczany na podstawie jakiegoś parametru, który nie jest oczywisty lub nie można go wcześniej ustawić, ponieważ zależy od twoich własnych danych. Jak widzieliście w poprzednich przykładach, możemy znaleźć dla pracowników zarabiających ponad 10 000 USD. To jest ok, ale co się dzieje, jeśli chcemy szukać pracowników zarabiających więcej niż średnia w firmie, aby zobaczyć, czy na to zasługują? Najwyraźniej to, czego nauczyliśmy się do tej pory, nie wystarczy, ponieważ średnie wynagrodzenie w firmie może się zmieniać od czasu do czasu. Pomyśl o podwyżkach płac, zatrudnionych lub zwolnionych pracownikach itd., które zmienią tę liczbę. Jest to bardzo powszechna sytuacja i musimy wprowadzić pojęcie podzapytania. Aby to ułatwić, najpierw będziemy szukać średniej pensji firmy, a następnie przetestujemy wszystkich pracowników pod kątem tej pensji, jeśli wynosiła ona około 10 000 USD. Więc jedyne, co musimy zrobić, to znaleźć sposób, aby najpierw obliczyć tę średnią pensję, a potem reszta będzie taka, jak pokazano wcześniej. Pokażmy przykład, jak to obliczyć. Na początek napiszmy zapytanie, jak znaleźć wszystkich pracowników zarabiających powyżej 10 000 USD. Będzie to coś podobnego do następującego:

```
  SELECT
        first_name, last_name, salary
  FROM
        hr.employees
  WHERE
        salary > 10000
  ORDER BY 3 DESC;

  FIRST_NAME           LAST_NAME
SALARY
  -------------------- ------------------------- ----
------
  Steven               King
 24000
```

```
  Neena                Kochhar
17000
  Lex                  De
Haan                       17000
  John                 Russell
14000
  Karen                Partners
13500
  Michael              Hartstein
13000
  Shelley              Higgins
12000
  Alberto              Errazuriz
12000
  Nancy                Greenberg
12000
  Lisa                 Ozer
11500
  Gerald               Cambrault
11000
  Den                  Raphaely
11000
  Ellen                Abel
11000
  Eleni                Zlotkey
10500
  Clara                Vishney
10500
  15 rows selected
```

To połowa pracy, ponieważ teraz musimy zmodyfikować nasze zapytanie, aby zamiast tego kierować się na kwotę 10 000 USD obliczającą pracowników, którzy zarabiają więcej niż średnia firmy. Zrobienie tego jest łatwiejsze niż myślenie o tym, jak obliczyć to drugie. Jeśli do tej pory zrozumiałeś wszystko, co omówiliśmy, powinno to być dla ciebie łatwe

SELECT

TRUNC(AVG(salary))

FROM

hr.employees;

TRUNC(AVG(SALARY))

6461

Uwaga: w tym przypadku nie potrzebujemy grupy według wyrażenia, ponieważ nie chcemy żadnej grupy, rozważamy firmę jako całość. Gdybyśmy chcieli wykonać te same obliczenia według działów, potrzebowalibyśmy trochę pogrupowania według id_działu, ale zobaczymy to później, ponieważ podzapytanie zwraca więcej niż jeden wiersz (w rzeczywistości jeden na dział), a nadal musimy zobaczyć coś innego aby móc odpowiedzieć na to pytanie.

Ok, to wszystko! Mamy średnią pensję całej firmy, więc teraz pozostaje tylko kwestia zastosowania koncepcji podzapytania. Jak widać mamy tutaj zapytanie główne, to które oblicza pracowników, których pensja jest większa niż jedna kwota oraz podzapytanie, które jest potrzebne do obliczenia średniej firmy. Pozostaje więc tylko kwestia ich wspólnego spisywania. Użyjemy nawiasów w miejscu gdzie i dodamy podzapytanie tak, jakby miało jakąkolwiek wartość:

```
SELECT
      first_name, last_name, salary
FROM
      hr.employees
WHERE
      salary > (SELECT
                        TRUNC(AVG(salary))
                  FROM
                        employees)
ORDER BY 3 DESC;
```

```
And the result is the following:
FIRST_NAME           LAST_NAME
SALARY
-------------------- ------------------------- ----
------
Steven               King
 24000
Neena                Kochhar



 17000
  Lex                  De
Haan                       17000
  ... (output truncated)
  David                Lee
  6800
  Susan                Mavris
  6500
  Shanta               Vollman
  6500
  51 rows selected
```

Oblicz liczbę pracowników, którzy przekraczają średnią pensję

Wykluczanie pracowników z Department_id = 90 (Kierownictwo) Chcemy, abyś ty też spróbował. Jak myślisz, jak moglibyśmy policzyć liczbę pracowników według działów, którzy przekraczają średnią pensję? Chcielibyśmy wyłączyć z średniej kadry kierowniczej. Spróbuj zastosować tę samą metodologię, którą wyjaśniliśmy w tym punkcie, aby znaleźć rozwiązanie.

1. Spróbuj wymyślić zapytanie, które to spełni. WSKAZÓWKA: Powinno to być zapytanie zawierające identyfikator działu i liczbę pracowników.

2. Spróbuj pomyśleć o podzapytaniu, które jest do tego potrzebne. WSKAZÓWKA: Tym razem musimy obliczyć średnią wynagrodzeń firmy, filtrując najpierw pracowników należących do id_działu = 90.

3. Połącz oba zapytania i napisz wymagane zapytanie. Nie oszukuj i spróbuj samodzielnie przemyśleć rozwiązanie! Jeśli po pewnym czasie nie możesz, tutaj jest dla odniesienia:

```
SELECT
      department_id, count(*)
FROM
      hr.employees
WHERE
      salary > (SELECT
                        TRUNC(AVG(salary))


                  FROM
                        hr.employees
                  WHERE
                        department_id <>90)
GROUP BY
      department_id
ORDER BY 2 DESC;
DEPARTMENT_ID   COUNT(*)
------------- ----------
           80         34
          100          6
           50          4
           90          3
          110          2
           40          1
           60          1
           20          1
                       1
           30          1
           70          1

 11 rows selected
```

Łączenie tabel

Do tej pory nauczyliśmy się całkiem sporo o tym, jak pracować z relacyjną bazą danych. Jednak dość często zdarza się, że musimy pracować z kilkoma tabelami jednocześnie. Więc to, co widzieliśmy do tej pory, jest mało przydatne. Zwykle w hurtowni danych będziesz mieć kilka tabel. Niektóre z nich będą tabelami wyszukiwania, zawierającymi podstawowe dane Twoich klientów, dostawców, produktów itd., podczas gdy inne będą tabelami faktów, zawierającymi informacje o sprzedaży, kosztach, odcinkach wypłat pracowników itd. Więcej na ten temat dowiemy się w następnych rozdziałach, ale zacznij o tym myśleć. Na razie koncentrujemy się na naszych dwóch przykładowych tabelach, Pracownicy i Działy. Prawdopodobnie w pewnym momencie chcesz policzyć na przykład liczbę pracowników w każdym dziale. W tym momencie możesz pomyśleć o rozwiązaniu niezbyt eleganckim, ale to zadziała, czyli wybraniu kolumny DEPARTMENT_ID z tabeli pracowników i dodaniu funkcji agregacji, takiej jak count (*), a następnie zastosowaniu grupy w kolumnie DEPARTMENT_ID. Chociaż to zadziała, mamy tutaj pewne wady (i pewne zalety, szczerze mówiąc). Najważniejsze jest to, że to rozwiązanie nie podaje nam nazw działów. Mamy tylko identyfikatory, więc w zależności od tego, jakie informacje chcemy przedstawić, jest to niedopuszczalne. Zaletą jest to, że wysyłanie zapytań tylko do jednej tabeli jest zawsze szybsze niż pobieranie danych z więcej niż jednej, oczywiście biorąc pod uwagę, że stosowane są te same filtry. Istnieje wiele rodzajów sprzężeń, ale składnia jest bardzo podobna. Przedstawimy składnię ANSI 92 Join i poprzednią, ANSI89. Chociaż ANSI 92 jest obsługiwany we wszystkich bazach danych, poprzedni może nie być, zwłaszcza jeśli chodzi o łączenia zewnętrzne. Zależy to całkowicie od tego, którego używasz, i chociaż zalecamy trzymanie się standardu ANSI 92, który jest obsługiwany we wszystkich bazach danych, prawdą jest również, że jesteśmy bardziej przyzwyczajeni do starego. Składnia zapisu łączenia obejmującego dwie tabele w ANSI89 jest następująca:

SELECT list_of_columns (separated by commas, or *)

FROM table1, table2

[WHERE table1.column1=table2.column1 ...]

Whereas in the new syntax the format is as follows:

SELECT list_of_columns (separated by commas, or *)

FROM table1

JOIN table2

ON (table1.column1=table2.column1)

Jak widać, oba formaty są podobne. Aby wygenerować łączenie, w starej składni używamy przecinka, natomiast w nowszej składni używamy słowa kluczowego JOIN (lub podobnego, w zależności od typu łączenia, więcej później), a następnie dodajemy klauzulę ON, aby określić łączone kolumny z dwóch stolików. Zamiast tego, w pierwszym przypadku połączone kolumny są określone w tej samej klauzuli gdzie, co może powodować zamieszanie, ponieważ czasami może być trudno zobaczyć, które kolumny są połączone i jakie są warunki zastosowane w klauzuli gdzie. Większość zwolenników ANSI92 używa tego argumentu jako głównego w swojej obronie nowszej składni.

Rodzaje sprzężeń

Jak powiedzieliśmy wcześniej, istnieje wiele rodzajów sprzężeń. Bardzo ważne jest, aby wiedzieć, co chcemy wybrać, ponieważ zmusi nas to do użycia jednego lub drugiego rodzaju łączenia. Zaczniemy od najbardziej podstawowego, a te najbardziej skomplikowane przyjrzymy się przykładom z tabel Pracownicy i Działy, które widzieliśmy wcześniej.

Połączenie kartezjańskie

W rzeczywistości nie jest to łączenie, ale jest również nazywane łączeniem (łączenie krzyżowe). Jest to relacja „bez relacji" między dwiema tabelami. Z tego powodu jest to rodzaj łączenia, którego najczęściej chcesz uniknąć, ponieważ zwykle dochodzi do niego przez pomyłkę, przy błędnym określeniu klauzul łączenia. Jak sama nazwa wskazuje, ten typ łączenia jest iloczynem kartezjańskim, który obejmuje łączenie (lub łączenie) każdego pojedynczego wiersza z pierwszej tabeli z każdym pojedynczym wierszem drugiej tabeli. Jak już zauważyłeś, całkowita liczba rekordów jest iloczynem liczby rekordów z pierwszej tabeli przez liczbę rekordów z drugiej. Chociaż prawdą jest, że zwykle będziesz chciał uniknąć tego łączenia, może to być przydatne w niektórych przypadkach, na przykład, gdy wiesz, że dwa stoły nie mają ze sobą nic wspólnego, ale mimo to chcesz do nich dołączyć, lub gdy jeden ze stołów, który nie ma nic wspólnego z innymi, zawiera tylko jeden wiersz i chcesz dołączyć dwie tabele razem, generując nową tabelę z kolumnami obu tabel. Mając kilka rekordów z tabeli pracowników

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID	SALARY	COMMISSION_PCT	MANAGER_ID	DEPARTMENT_ID
1	100	Steven	King	SKING	515.123.4567	17/06/87	AD_PRES	24000	(null)	(null)	90
2	101	Neena	Kochhar	NKOCHHAR	515.123.4568	21/09/89	AD_VP	17000	(null)	100	90
3	102	Lex	De Haan	LDEHAAN	515.123.4569	13/01/93	AD_VP	17000	(null)	100	90

i kilka rekordów z tabeli departamentów ,

	DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
1	10	Administration	200	1700
2	20	Marketing	201	1800
3	30	Purchasing	114	1700
4	40	Human Resources	203	2400

możemy utworzyć między nimi sprzężenie krzyżowe:

SELECT

*

FROM

hr.employees, hr.departments;

2.889 records selected

Or using the newer ANSI92 syntax:

SELECT

*

FROM

hr.employees

CROSS JOIN

hr.departments;

2.889 records selected

Jak widać na ostatnim rysunku, kolumna ID_działu pochodząca z tabeli pracowników i kolumna ID_działu pochodząca z tabeli działów nie pasują do siebie. Jest to oczywiste, ponieważ nie dodaliśmy takiego warunku, więc silnik SQL zasadniczo łączy każdego pracownika, niezależnie od jego działu, ze wszystkimi działami w firmie. Na rysunku widać, że pracownicy przypisani do działów 90 i 60 są powiązani z działem 10 i tak dalej.

Połączenie wewnętrzne

Najczęściej używanym łączeniem jest łączenie wewnętrzne. Ten rodzaj łączenia polega na łączeniu dwóch tabel, które mają co najmniej jedną wspólną kolumnę. W klauzuli where lub w klauzuli ON dodajemy warunek table1.column1 = table2.column1 i tak dalej, aby określić wszystkie kolumny, których chcemy użyć do łączenia. Spowoduje to utworzenie wyniku z kolumnami, które wybraliśmy w

projekcji z dwóch (lub więcej) tabel połączonych za pomocą kolumn, które określiliśmy w klauzulach Where lub ON. Zobaczmy przykład: wiemy, że identyfikator działu jest wspólny dla dwóch kolumn i powiedziano nam, że mamy uzyskać listę pracowników i przypisane im nazwy działów. Można to osiągnąć za pomocą następujących zapytań:

```
SELECT
employees.first_name, employees.last_name,
departments.department_name
FROM
hr.employees, hr.departments
WHERE
employees.department_id =
departments.department_id;
```

or using the ANSI92 syntax:

```
SELECT
employees.first_name, employees.last_name,
departments.department_name
FROM
hr.employees
JOIN
hr.departments
ON
(employees.department_id =
departments.department_id);
```

Rezultatem, jak widać, tym razem nie jest już robienie iloczynu kartezjańskiego, ponieważ wprowadzamy poprawną klauzulę łączenia, więc całkowita liczba pobranych rekordów będzie zwykle mniejsza niż wynik mnożenia. Jeśli jedna z tabel (najmniejsza) ma unikalne rejestry dla danej kolumny, liczba zwróconych rekordów będzie równa liczbie rekordów, które mamy w większej tabeli, chyba że jedna z tabel zawierała rekord pusty lub wartość identyfikator działu nie istnieje w tabeli działów, ponieważ wtedy dopasowanie jest niemożliwe. W rzeczywistości mamy pracownika z nieprzypisanym działem w naszej tabeli, więc suma zamiast 107 wynosi 106, jak widać w wynikach zapytania.

```
   FIRST_NAME           LAST_NAME                 DEPA
RTMENT_NAME
   -------------------- ------------------------- ----
-------------------------
   Jennifer             Whalen                    Admi
nistration
   Pat                  Fay                       Mark
eting
   Michael              Hartstein                 Mark


 eting
    Sigal                Tobias                    Purc
 hasing
    106 rows selected
```

Istnieje specjalny przypadek łączenia wewnętrznego, zwany łączeniem naturalnym. Naturalne łączenie wykonuje wewnętrzne równoważne połączenie dwóch tabel bez określania kolumn łączenia. Być może zastanawiasz się, jak silnik to osiąga. To jest łatwe. Używa nazw kolumn i typów danych kolumn, więc każda kolumna, która ma dokładnie taką samą nazwę i ten sam typ danych, jest automatycznie dodawana do łączenia przez silnik. Ponieważ w naszym przykładzie kolumna id_działu jest wspólna dla obu tabel i ma ten sam typ danych, możemy przepisać poprzednie łączenie, aby zamiast tego użyć składni łączenia naturalnego:

SELECT

*

FROM

hr.employees

NATURAL JOIN

hr.departments;

32 rows selected

Jednak zdecydowanie odradzamy używanie naturalnych połączeń, ponieważ zdasz sobie sprawę, że numery rekordów nie pasują, a my starannie wybraliśmy tę próbkę, aby ostrzec Cię o niebezpieczeństwach związanych z naturalnym połączeniem i niepożądanymi skutkami ubocznymi. Jeśli wrócimy do definicji tabeli pracowników i działów, zobaczymy, że oprócz kolumny id_działu, współdzielą one również kolumnę id_kierownika. Więc ponieważ używamy naturalnego łączenia, to łączenie kolumny manager_id jest również automatycznie dodawane w klauzuli join i jest to coś, co nie ma tutaj żadnego sensu, ponieważ kierownik działu niekoniecznie oznacza, że jest kierownikiem wszystkich pracowników zatrudnionych w tym samym dziale.

Uwaga : Używanie naturalnych połączeń może wprowadzać w błąd i być źródłem błędów. Używaj ich tylko wtedy, gdy masz pewność, że mogą być bezpiecznie używane. Zachęcamy do nieużywania ich z następującego powodu: zastanów się, czy mamy dwie tabele, które mają tylko jedną kolumnę, ale w którymś momencie ktoś zmodyfikuje jedną z kolumn, wprowadzając kolumnę, która wcześniej istniała

w drugiej tabeli: Spowoduje to, że join, aby dodać do niego nowe kolumny, powodując potencjalnie niechciany wynik. To samo można zastosować w przypadku połączeń krzyżowych. Używaj go ostrożnie.

Połączenie zewnętrzne

Czasami przydatne jest łączenie tabel, które mają częściowo wspólne wiersze, a w wyniku łączenia chcesz mieć wszystkie wiersze, które istniały w jednej z tabel źródłowych lub w obu z nich, bez względu na to, czy nie mają one odpowiednika w inny połączony stół. Zwłaszcza, gdy pracujemy na hurtowniach danych, jest to zwykle źródłem problemów (brakujące rekordy po złączeniu) i chociaż nasze procesy i skrypty ETL powinny sobie z tym poradzić, w niektórych obliczeniach może być konieczne upewnienie się, że nie stracimy rekordów, które nie istnieją w żadnej z zaangażowanych tabel w złączeniu. Wyobraź sobie, że Twoja firma notuje sprzedaż od początku swojej działalności. Z jakiegokolwiek powodu masz sprzedaż powiązaną z produktami, których już nie sprzedajesz, ale robiłeś to w przeszłości. Wyobraź sobie, że straciłeś wszystkie szczegóły tego produktu, a nawet nie masz już wpisu w swoim systemie ERP lub transakcyjnym. Jeśli połączysz tabelę produktów z tabelą sprzedaży, nie będzie korespondencji między sprzedażą tych produktów a informacjami o produkcie. Oznacza to, że jeśli dołączysz do tych dwóch stołów, stracisz wszystkie rekordy sprzedaży. Doprowadzi to do problemów i zamieszania, ponieważ jeśli obliczysz sprzedaż zagregowaną według kategorii produktów lub całości, a nawet według produktu, „zapomnisz" policzyć te sprzedaże. Jest to czasem niepożądane. Jak możesz pomyśleć, możesz stworzyć fałszywy wpis produktu w tabeli produktów, z poprawnym identyfikatorem produktu, więc wtedy zadziała łączenie wewnętrzne. Może to być rozwiązanie, ale czasami nie jest to łatwe ze względu na liczbę utraconych rekordów lub może to być niepraktyczne. W takich konkretnych przypadkach na ratunek przychodzą łączenia zewnętrzne. Zasadniczo istnieją trzy typy sprzężeń zewnętrznych: lewe sprzężenia zewnętrzne (lub lewe złączenia dla zwięzłości), prawe sprzężenia zewnętrzne (lub prawe złączenia) i pełne złączenia zewnętrzne. Te trzy mają tę samą składnię, z wyjątkiem słów kluczowych join, ale zachowują się zupełnie inaczej.

Uwaga : Złączenia zewnętrzne są szczególnie interesujące do wykorzystania w procesie ETL, aby nie stracić żadnego rejestru podczas ładowania, ale raczej nie zaleca się używania fikcyjnych wpisów dla tabel końcowych w celu ich uzupełnienia.

Lewe złącze zewnętrzne (lub złącze lewe)

W lewym łączeniu rekordy, które nie pasują z pierwszej tabeli, pojawiają się w wyniku łączenia, natomiast te, które nie pasują z drugiej tabeli, nie pojawiają się. Wynikiem złączenia będzie wówczas kompozycja między złączeniem wewnętrznym (wszystkie rekordy, które pojawiają się w obu tabelach) i tymi, które pojawiają się tylko w pierwszej tabeli. Aby to lepiej zrozumieć, skorzystajmy jeszcze raz z naszych zaprzyjaźnionych stolików pracowników i działów. Jeśli przypomnisz sobie tabelę Działy, każdy dział ma kierownika. Cóż, to nie do końca prawda. Istnieją działy, które nie mają menedżera, więc zamiast tego w kolumnie identyfikator_kierownika w tabeli działów występuje wartość pusta. Jeśli klucz obcy nie odwołuje się do wartości, jak w tym przypadku, jeśli wykonamy połączenie wewnętrzne, te działy zostaną odfiltrowane, ponieważ wartość null nie będzie pasować do żadnego pracownika w naszej tabeli (nie mamy pracownika zerowego, prawda?). Zobaczmy przykład lewego sprzężenia zewnętrznego, w którym zwrócimy wszystkie działy, które mają menedżerów, wraz z nazwiskiem menedżera, ale także wszystkie działy, które nie mają menedżera. Tak więc wynikiem połączenia powinno być tych samych 27 działów wraz z nazwiskiem kierownika dla tych, które je mają.

```
SELECT

d.department_name, e.First_Name || ' ' ||
```

e.Last_Name as Manager

FROM

hr.departments d

LEFT JOIN

hr.employees e

ON (d.manager_id=e.employee_id);

DEPARTMENT_NAME MANAGER

------------------------------ --------------------

Executive Steven King

IT Alexander Hunold

Finance Nancy Greenberg

Purchasing Den Raphaely

Shipping Adam Fripp

..... (output truncated)

Benefits

Shareholder Services

Control And Credit

Corporate Tax

Treasury

27 rows selected

Jak widać z poprzedniego fragmentu, mamy 27 rekordów, ponieważ mamy 27 działów, a niektóre z nich pokazują zerowego menedżera. Dzieje się tak, ponieważ Outer Join dodaje rekordy, które mają wpis w tabeli działów, ale nie w tabeli pracowników (menedżer).

Prawe łączenie zewnętrzne (lub prawe łączenie)

W ten sam sposób mamy lewe łączenie, mamy prawe łączenie. Pomysł jest ten sam, ale tym razem rekordy, które zostaną dodane do wyniku łączenia wewnętrznego, to te, które istnieją w prawej tabeli (drugiej tabeli) łączenia, podczas gdy te, które są obecne tylko w lewej tabeli (pierwsza tabela złączenia) zostaną utracone. Wyobraźmy sobie, że teraz chcemy wiedzieć, w jakim dziale pracownik jest kierownikiem. Jak myślisz, nie wszyscy pracownicy są menedżerami, więc jeśli powtórzymy to samo złączenie, które zrobiliśmy w poprzednim przykładzie, tym razem będziemy mieć wszystkie rekordy z prawej części złączenia (tabela pracowników) i z którego działu zarządzanie:

```
   SELECT
         d.Department_name, e.First_Name || ' ' ||
e.Last_Name as Manager
   FROM
         hr.departments d
   RIGHT JOIN
         hr.employees e
   ON (d.manager_id=e.employee_id);

   DEPARTMENT_NAME                MANAGER
   ------------------------------ --------------------
--------------------------
   Administration                 Jennifer Whalen
   Marketing                      Michael Hartstein
   Purchasing                     Den Raphaely
   Human Resources                Susan Mavris
   Shipping                       Adam Fripp
   IT                             Alexander Hunold
   Public Relations               Hermann Baer
   Sales                          John Russell
   Executive                      Steven King



 Finance                        Nancy Greenberg
 Accounting                     Shelley Higgins
 ... (output truncated)
                                Mozhe Atkinson
                                Alberto Errazuriz
                                Allan McEwen
                                Douglas Grant

   107 rows selected.
```

Jak widać tym razem są pracownicy, którzy niczym nie zarządzają.

Pełne połączenie zewnętrzne (lub pełne połączenie)

Wyobraź sobie, że chcesz, aby zarówno Left Join, jak i Right Join zostały wykonane w tym samym czasie. Wtedy na ratunek przychodzi Full Outer Join lub Full Join. Wyobraź sobie, że chcesz listę działów i ich kierowników, ale jednocześnie chcesz wszystkich pracowników i dział, którym zarządzają. Oczywiście potrzebujesz kombinacji obu. Zobaczmy przykład:

```
SELECT
      d.Department_name, e.First_Name || ' ' ||
e.Last_Name as Manager
FROM
      hr.departments d
FULL OUTER JOIN
      hr.employees e
ON (d.manager_id=e.employee_id);

DEPARTMENT_NAME                MANAGER
------------------------------ --------------------
-------------------------
Executive                      Steven King
                               Neena Kochhar
                               Lex De Haan
IT                             Alexander Hunold
                               Bruce Ernst
... (output truncated)
Payroll
Recruiting


Retail Sales

123 rows selected.
```

Jak widać mamy teraz Działy z Kierownikiem, Pracowników, którzy nie zarządzają żadnym działem oraz Działy bez Kierownika.

Uwaga: lewe i prawe łączenie jest znacznie częściej używane niż pełne łączenie zewnętrzne, zwłaszcza w środowisku hurtowni danych. Ale ważne jest również, aby wiedzieć, że zawsze istnieje możliwość połączenia obu w jedno oświadczenie.

Aliasy tabeli

Czasami my, jako ludzie, jesteśmy trochę leniwi. Ciągłe odwoływanie się do nazw tabel za pomocą ich nazw jest trudne, a ponadto, jeśli ta sama tabela jest używana kilka razy w instrukcji select, jest myląca. Na szczęście język SQL rozwiązuje ten problem, umożliwiając programistom nadawanie tabelom pseudonimów lub aliasów. Następujące stwierdzenia są podobne:

SELECT

first_name, last_name, department_name

FROM

hr.employees, hr.departments

WHERE

employees.department_id =

departments.department_id;

and

```
SELECT

first_name, last_name, department_name

FROM

hr.employees e, hr.departments d

WHERE

e.department_id = d.department_id;
```

Jedyna różnica polega na tym, że dodaliśmy dwa aliasy tabel, aby odwoływać się do ich oryginalnych tabel przy użyciu aliasu lub pseudonimu. Alias należy dodać po nazwie tabeli, a następnie można go użyć w klauzuli where i kolejnych klauzulach, aby odnieść się do oryginalnych tabel.

Skorelowane podzapytania

Widzieliśmy wcześniej, jak działa podzapytanie. Poradziliśmy jednak, że niektóre zapytania podrzędne muszą używać sprzężeń, aby zewnętrzna tabela mogła obliczyć określone obliczenia. Wyobraźmy sobie, że chcemy odzyskać pracowników, których wynagrodzenie jest powyżej średniej wynagrodzeń wszystkich osób w ich działach. Nie możemy tego zrobić bezpośrednio za pomocą podzapytania, ponieważ musimy obliczyć średnią pensję dla każdego działu, a następnie porównać każdego pracownika z tą średnią pensją. Ale te dwa zapytania są ze sobą powiązane, ponieważ dział, w którym przebywa pracownik, musi być taki sam, jak ten, który obliczamy dla wynagrodzenia, więc skutecznie potrzebujemy łączenia. To jeden z najczęstszych przykładów. Zobaczmy, jak to rozwiązać. Zapytanie będzie się składało, jak wspomniano, z dwóch części. Jedno, zwane zapytaniem zewnętrznym, wybierze pracowników spełniających warunek, a drugie zapytanie, zwane zapytaniem wewnętrznym, będzie zapytaniem obliczającym średnie wynagrodzenie na dział. Relacja między zapytaniem wewnętrznym i zewnętrznym zostanie określona w klauzuli where zapytania wewnętrznego, ponieważ zapytanie zewnętrzne nie może odwoływać się do kolumn zapytania wewnętrznego, chyba że znajdują się one w klauzuli FROM, co nie ma miejsca (są one w WHERE lub klauzula filtrująca). Zapytanie będzie wyglądać mniej więcej tak. Zwróć uwagę na aliasy tabeli wewnętrznej i zewnętrznej, jak wyjaśniono w poprzednim akapicie:

```
   SELECT
         first_name || ' ' || last_name EMP_NAME,
salary
   FROM
         hr.employees emp_outer
   WHERE
         salary > (SELECT
                           AVG(salary)
                  FROM
                           employees emp_inner
                  WHERE
                           emp_inner.department_id =
emp_outer.department_id)
   ORDER BY 2 DESC;

   EMP_NAME
SALARY


   ---------------------------------------- ----
------
   Steven
King                                      24000
   John
Russell                                    14000
   Karen
Partners                                   13500
   Michael
Hartstein                                 13000
   ... (output truncated)
   Renske
Ladwig                                     3600
   Jennifer
Dilly                                     3600
   Trenna
Rajs                                       3500

    38 rows selected
```

Ustaw operacje

Łączenie danych z różnych tabel jest bardzo przydatne, ale są też inne operacje, których trzeba się nauczyć. Co się stanie, jeśli chcesz połączyć dane z dwóch tabel, które mają identyczny układ? Pomyśl na przykład o dwóch różnych tabelach zawierających dane sprzedażowe z 2016 i 2017 roku. Musi istnieć sposób na ich „połączenie” i wykorzystanie jako jednej tabeli. Na szczęście jest jeden. Przedstawimy kilka operatorów zbiorów, które ułatwią te i inne zadania. Zacznijmy od operatora unii.

Union i Union Wszystkich Operatorów

Operator Unii, jak wprowadzono wcześniej, łączy wynik jednego zapytania z wynikiem innego. Warto jednak wiedzieć, że są pewne wymagania. Dwie tabele lub zapytania muszą zwracać identyczną liczbę kolumn, a także mieć te same typy danych w każdej kolumnie. Gwarantuje to, że wynik może być konkatenacją wyników obu zapytań lub tabel, a dane zostaną wyrównane i umieszczone w kolumnie, która musi być. Przykład sprzedaży przedstawiony w poprzednim akapicie jest najbardziej przejrzysty, aby zrozumieć, jak zachowuje się instrukcja UNION lub UNION ALL.

Aby pokazać to na przykładzie, musimy trochę popracować nad danymi HR. Wróćmy do naszej tabeli pracowników i utworzymy dwie nowe tabele na podstawie wynagrodzenia, jakie otrzymuje pracownik. Stworzymy jedną tabelę dla pracowników zarabiających mniej lub równo niż 6000 USD, a drugą dla pracowników zarabiających powyżej 6000 USD.

Uwaga: używamy tutaj instrukcji tworzenia, aby pokazać ten przykład. To stwierdzenie zostanie omówione później w następnych sekcjach, więc nie martw się, jeśli nie rozumiesz go dobrze w tym momencie; po prostu wykonaj przykładową instrukcję i postępuj zgodnie z instrukcjami.

Uruchommy następujące dwie instrukcje:

```
CREATE TABLE

hr.employeesLTE6000

AS SELECT

*

FROM

hr.employees

WHERE

salary <=6000;

and

CREATE TABLE

hr.employeesGT6000

AS SELECT

*

FROM

hr.employees

WHERE

salary >6000;
```

the output should be something like that:

```
Table HR.EMPLOYEESLTE6000 created.

Table HR.EMPLOYEESGT6000 created.
```

Mamy więc teraz w naszym schemacie HR dwie nowe tabele, jedną dla pracowników zarabiających więcej niż 6000 USD, a drugą dla pracowników zarabiających mniej niż 6000 USD. Ponieważ obie tabele mają taką samą liczbę kolumn i te same typy danych, możemy użyć instrukcji union, aby połączyć je z powrotem:

```
SELECT

*

FROM

hr.employeesLTE6000

UNION
```

SELECT

*

FROM

hr.employeesGT6000;

Wynikiem selekcji jest 107 pracowników, których mamy w naszej oryginalnej tabeli pracowników. Wyobraź sobie teraz następujący nowy zestaw tabel:

-HR.EMPLOYEESLTE6000, który obejmuje wszystkich pracowników zarabiających mniej niż lub równo 6000 USD. Oraz nową tabelę o nazwie

-HR.EMPLOYEESGTE6000, która zawiera pracowników zarabiających co najmniej 6000 USD. Stwórzmy brakującą tabelę:

CREATE TABLE

hr.employeesGTE6000

AS SELECT

*

FROM

hr.employees

WHERE

salary >=6000;

Table HR.EMPLOYEESGTE6000 created.

And amend slightly our previous query to use the new table:

SELECT

*

FROM

hr.employeesLTE6000

UNION

SELECT

*

FROM

hr.employeesGTE6000;

Wynik jest taki sam. W porządku. Ale co się stanie, gdy zamiast tego użyjemy tego samego zapytania z operatorem UNION ALL? Zobaczmy:

SELECT

*

FROM

hr.employeesLTE6000

UNION ALL

SELECT

*

FROM

hr.employeesGTE6000;

109 rows selected;

Ups! Mamy problem. Mamy dwa rekordy więcej; mamy zduplikowane dane! Może się zdarzyć, że naprawdę chciałeś to zrobić, ale prawdopodobnie tak nie będzie. Właśnie zduplikowaliśmy dane dla pracowników, którzy zarabiają dokładnie 6000 USD, ponieważ są w obu tabelach. Tak więc NION usuwa duplikaty, podczas gdy UNION ALL nie. Być może zastanawiasz się, używajmy UNION zawsze zamiast UNION ALL. To częściowo prawda. Większość ludzi tak robi, ale zazwyczaj nie jest to dobra decyzja. Ponieważ UNION usuwa duplikaty, jest to proces najdroższy do wykonania dla silnika bazy danych UNION ALL. Używaj ich więc mądrze. Jeśli nie zależy Ci na duplikatach, zawsze używaj UNION ALL, ponieważ zawsze będzie szybciej. Jeśli zależy Ci na zduplikowanych rekordach, użyj UNION, która zawsze odrzuca powtarzające się wiersze.

Uwaga: Union statement nie gwarantuje, że wiersze będą dodawane z jednej tabeli po drugiej ani sortowane w zależności od tabeli, z której pochodzą. Wiersze z pierwszej tabeli mogą pojawiać się na początku, na końcu lub mieszać się z wierszami z drugiej tabeli. Jeśli z jakiegokolwiek powodu chcesz zlecić konkretne zlecenie, musisz użyć klauzuli ORDER BY na końcu instrukcji.

Operator przecięcia

Operator przecięcia działa w taki sam sposób jak operator logiczny o tej samej nazwie. Zasadniczo odczytuje dane z obu zapytań lub tabel i przechowuje tylko te wiersze, które pojawiają się dokładnie w obu tabelach. Ponownie konieczne jest spełnienie tych samych warunków wstępnych, co w przypadku deklaracji Union i Union All. Zobaczmy przykład wykorzystujący wcześniej utworzone tabele:

```
SELECT
      *
FROM
      hr.employeesLTE6000
INTERSECT
SELECT
      *
FROM
      hr.employeesGTE6000;
```

As you may see, two employees are returned, which are exactly the two that appear in both tables, having $6,000 as a salary.

```
EMPLOYEE_ID FIRST_NAME           LAST_NAME
----------- -------------------- ------------------
-------
EMAIL                     PHONE_NUMBER         HIRE
_DATE JOB_ID         SALARY
------------------------- -------------------- ----
----- ---------- ----------
COMMISSION_PCT MANAGER_ID DEPARTMENT_ID
-------------- ---------- -------------
         104 Bruce                Ernst
BERNST                    590.423.4568         21/0
5/07  IT_PROG          6000
                        103            60

         202 Pat                  Fay
PFAY                      603.123.6666         17/0
8/05  MK_REP           6000



                        201            20
```

Operator minusa

Operator minus , znany również jako wyjątkiem w niektórych silnikach baz danych (ale uwaga, ponieważ składnia może się nieco zmienić), jest operatorem, który odejmuje od pierwszej tabeli rekordy, które są również zawarte w drugiej tabeli. Zobaczmy przykład, który to zilustruje: będziemy odejmować od jednej z naszych poprzednich tabel, PRACOWNICYGTE600, która zawiera wszystkich pracowników zarabiających 6000 lub więcej, te z tabeli PRACOWNICYGT6000, która zawierała Pracowników zarabiających więcej niż 6000. Tak więc wynik operacja odejmowania powinna być nowym zestawem wierszy zawierających tylko pracowników, którzy zarabiają dokładnie 6000, czyli tych, którzy będą obecni tylko w pierwszej tabeli. Sprawdźmy to:

```
  SELECT
        *
  FROM
        employeesGTE6000
  MINUS
  SELECT
        *
  FROM
        employeesGT6000;
  and the result are the two employees that exactly
earn 6000:
  EMPLOYEE_ID
FIRST_NAME           LAST_NAME                 EMAIL
                   PHONE_NUMBER         HIRE_DATE
JOB_ID         SALARY COMMISSION_PCT MANAGER_ID
DEPARTMENT_ID
  ----------- -------------------- ------------------
------- ------------------------- --------------------
--------- ---------- ---------- -------------- -------


--- -------------
          104
Bruce                Ernst                     BERNST
                   590.423.4568         21/05/07  IT_P
ROG          6000                       103
 60
          202
Pat                  Fay                       PFAY
                   603.123.6666         17/08/05  MK_R
EP           6000                       201
  20
```

Praca z datami

Daty są ważne w hurtowniach danych. Chociaż możemy z nimi pracować i przechowywać je tak, jakby były dowolnymi ciągami tekstowymi, nie jest to dobre z punktu widzenia wydajności, ponieważ w pewnym momencie może zaistnieć potrzeba operowania nimi i przekonwertowania ich na format daty. Widzieliśmy, że wszystkie bazy danych obsługują różne typy danych w definicjach kolumn. Tutaj, w tym akapicie, zobaczymy, jak operować funkcjami związanymi z datami. W zależności od dostawcy bazy danych funkcje mogą mieć inną nazwę, ale zasadniczo robią to samo.

Funkcje SYSDATE, NOW(), CURRENT_DATE i GETDATE().

Wszystkie silniki relacyjnych baz danych posiadają funkcję pobierania aktualnej daty systemu. Istnieją różne funkcje zwykle do pobierania tylko części daty, daty i części czasu, milisekund od określonego punktu w czasie i tym podobnych. W zależności od bazy danych funkcja może być wywoływana inaczej, ale wszystkie działają w ten sam sposób. Znajomość bieżącej daty jest zwykle ważna, gdy chcemy wykonywać operacje na datach, takie jak dodawanie, odejmowanie, filtrowanie, a jeszcze ważniejsza, gdy chcemy odświeżyć lub ponownie załadować dane na podstawie kryteriów daty. Wyobraźmy sobie, że chcemy odświeżyć dane tylko dla bieżącego miesiąca, czyli ostatnich 30 dni. Musimy porównać daty w naszych hurtowniach danych z rzeczywistą datą bieżącą. Najpierw musimy znaleźć sposób na pobranie aktualnych danych z naszej bazy danych. Na przykład w Oracle jest to realizowane za pomocą zarezerwowanego słowa sysdate. W SQL Server można to zrobić za pomocą funkcji GETDATE(), podczas gdy w mysql możemy użyć funkcji NOW(), aw Postgres możemy użyć wyrażenia ANSI CURRENT_DATE. Spójrzmy na kilka przykładów:

SELECT

SYSDATE

FROM

DUAL; --(This is Oracle's)

SYSDATE

18/06/16

SELECT

NOW(); --(This is MYSQL's, but you can also

do select SYSDATE();)

2016-06-18 12:46:42

SELECT

CURRENT_DATE --(This is Postgres)

date

2016-06-18

Dodawanie dni do daty

W ten sam sposób, w jaki możemy uzyskać aktualną datę, możemy operować polami daty. Możemy dodawać dni, miesiące i lata, odejmować je i usuwać części daty. Ponownie, zależy to od każdej implementacji bazy danych, dlatego zachęcamy do sprawdzenia instrukcji dostawcy bazy danych lub sprawdzenia jej w Internecie w celu uzyskania bardziej szczegółowych informacji o tym, jak z nimi pracować. W tym rozdziale pokażemy kilka przykładów użycia Oracle Express Edition i mysql. W przypadku Oracle uruchamiamy zapytanie takie jak to:

SELECT

SYSDATE, SYSDATE + INTERVAL '1' DAY

FROM

dual;

SYSDATE SYSDATE+INTERVAL'1'DAY

-------- ----------

18/06/16 19/06/16

Podczas gdy w przypadku mysql musimy go nieco zmienić:

SELECT

SYSDATE(), SYSDATE() + INTERVAL '1' DAY ;

SYSDATE() SYSDATE() + INTERVAL '1' DAY

---------- ----------

2016-06-18 15:22:33 2016-06-19 15:22:33

Za pomocą tej samej procedury możemy odjąć dowolną wartość. Zwróć uwagę, że słowo kluczowe INTERVAL jest bardzo wygodne, ponieważ możemy zmienić interwał DAY o dowolny potrzebny nam przedział czasu: MIESIĄC, ROK, GODZINA, MINUTA, SEKUNDA…

Wyrażenia warunkowe

Język SQL ma wbudowane pewne wyrażenia warunkowe. Wyrażenia warunkowe sprawdzają warunek i wykonują akcję lub inną w zależności od tego, czy wynik testu jest prawdziwy, czy fałszywy. Podobnie jak w przypadku każdego języka programowania, warunek testu musi być warunkiem boolowskim, więc wynik jego oceny zawsze będzie prawdziwy lub fałszywy.

Wyrażenie przypadku

Wyrażenie Case jest bardzo przydatną instrukcją sterującą do częściowego wykonywania obliczeń lub zbierania danych z określonej kolumny na podstawie warunku. Składnia instrukcji case jest następująca:

CASE

WHEN BooleanExpression1 THEN Branch1

...

WHEN BooleanExpressionN THEN BranchN

ELSE BranchN+1

END (alias for the column)

Wyobraźmy sobie teraz, że chcemy pobrać w kolumnie z wartością zależną od daty. Zapytamy o numer miesiąca i przetłumaczymy go na nazwę miesiąca. Oczywiście istnieją lepsze podejścia do tego, ale dla celów ilustracyjnych, to będzie dobre.

SELECT

CASE

WHEN to_char(sysdate,'mm')=06 THEN

'June' ELSE 'Another Month'

END CurrentMonth

FROM

dual;

CURRENTMONTH

June

This is Oracle syntaxis using sysdate and the dual pseudotable but the same

can be written in for example, mysql/mariadb:

SELECT

CASE

WHEN month(now())=06 THEN 'June' ELSE

'Another Month'

END CurrentMonth;

CurrentMonth

June

Wyrażenia Decode() lub IF().

Wyrażenie decode jest również używane jako instrukcja warunkowa. Nie zachęcamy do korzystania z nich, chyba że jesteś do tego przyzwyczajony lub jesteś przyzwyczajony do programowania w językach, które mają klauzule if/else, ponieważ czasami może to być nieco tajemnicze, co utrudnia debugowanie. Można go użyć w zamian za obudowę, a tam, gdzie jest bardziej zwarty, zwykle trudno jest zrozumieć, czy jest ich wiele zagnieżdżonych. Składnia jest następująca: DECODE (Statement,result1,branch1, …resultn, branchn, [else_branch]). Zobaczmy przykład, odpowiednik tego, który zrobiliśmy dla poprzedniego polecenia CASE:

SELECT

DECODE(to_char(sysdate,'mm'), 06, 'June',

'Another Month') CurrentMonth FROM

dual;

CURRENTMONTH

1

June

And in mysql/mariadb instead of decode we will be using

IF(BooleanExpression,if_case,else_case):

SELECT

IF(month(now())=06, 'June', 'Another Month')

CurrentMonth;

CurrentMonth

June

Wniosek

Jedna część do nauki języka SQL to za mało. Chcieliśmy jednak, abyś przedstawił powszechnie używany język interakcji z relacyjnymi bazami danych, abyś mógł zacząć myśleć o pisaniu własnych zapytań. W tym rozdziale zobaczyliśmy krótkie wprowadzenie do relacyjnych baz danych, rodzaje dostępnych instrukcji, typy danych, sposób pobierania i zliczania danych, sortowania, filtrowania i grupowania, a następnie bardziej zaawansowane instrukcje, w tym instrukcje zagnieżdżone lub skorelowane podzapytania. Zalecamy przeczytanie nieco więcej o języku SQL lub zakup jednej z wielu dostępnych książek, które dadzą Ci przewagę w pierwszym kroku potrzebnym do pomyślnego zbudowania rozwiązania BI. Jeśli twój kod jest dobry i dobrze napisany oraz wykonuje obliczenia, do których jest uprawniony, kolejne kroki będą znacznie łatwiejsze i będą działać znacznie szybciej niż kiepski projekt lub wadliwy kod. W kolejnych rozdziałach zaczniemy widzieć komponenty, które będą zgodne z naszą hurtownią danych, jak logicznie definiować encje, a następnie przełożymy to na wymagania do zbudowania schematu potrzebnego do przechowywania naszych danych. Wiemy, że ten rozdział był trochę gęsty, ale obiecujemy, że kolejne będą bardziej praktyczne, ponieważ zaczniemy pracować nad naszym rozwiązaniem od zera i jesteśmy pewni, że wiele się z niego nauczysz i że pomogą zbudujesz własne rozwiązanie BI.

4. Inicjalizacja projektu - instalacja bazy danych i źródła ERP

Widzieliśmy wiele bardzo teoretycznych aspektów BI. Zaczęliśmy od ogólnego wprowadzenia, części poświęconej zarządzaniu projektami, a także części poświęconej wprowadzeniu do SQL. Ale koniec końców, do tej pory nie zrobiliśmy nic praktycznego, więc czas zabrać się za rozwój rozwiązania. Zanim jednak zaczniemy, musimy zainstalować nasz źródłowy ERP, skąd będą pochodzić dane. Po zainstalowaniu i skonfigurowaniu jeszcze trochę pokombinujemy, a następnie wybierzemy naszą bazę danych do przechowywania hurtowni danych.

Potrzeba danych

Wszystkie systemy BI wymagają pewnego źródła danych. Dane te mogą być ustrukturyzowane lub nieustrukturyzowane. Ponadto dane te mogą pochodzić z wielu źródeł i mieć wiele kształtów. Ponadto możemy napotkać pewne problemy z jakością danych, ponieważ możemy mieć brakujące i/lub nadmiarowe dane. Ale zawsze istnieje punkt wyjścia, w którym generowane są dane źródłowe. Ten tekst nauczy Cię, jak wyodrębniać, przekształcać, ładować i raportować swoje dane. Ale jest pierwszy wstępny krok, który polega na pozyskiwaniu tych danych. W tej książce zdecydowaliśmy się użyć Odoo (wcześniej znanego jako OpenERP) jako źródła naszych danych. Tak, korzystanie z dowolnych dostępnych bezpłatnie przykładowych schematów baz danych dostępnych w Internecie lub dostarczanych w pakiecie z prawie każdą bazą danych mogło być łatwiejsze. I tak znowu, użyjemy niektórych, aby pokazać ci kilka zaawansowanych punktów w dalszej części książki, takich jak indeksowanie i tak dalej. Ale chcemy pokazać prawdziwą aplikację i kompletny potok danych. A w prawdziwym życiu zazwyczaj oznacza to połączenie z systemem ERP. Większość projektów BI zaczyna pozyskiwać dane z SAP. SAP nie jest jednak przeznaczony dla małych i średnich firm. Złożoność systemu jest przytłaczająca. a na domiar złego ceny licencji są wysokie. Jest więc prawdopodobne, że będziesz musiał celować w inne systemy ERP. Jest ich wiele w przystępnej cenie, większość z nich działa nawet w chmurze, a płacisz na bieżąco (głównie na podstawie liczby użytkowników korzystających z nich lub zakupionych modułów lub na podstawie użycia). Uważamy jednak, że istnieją dobre projekty open source, dostępne za darmo, które mogą odpowiadać potrzebom Twojej firmy.

Konfigurowanie systemu za pomocą Odoo ERP

Jeśli masz już zainstalowany ERP w swojej firmie lub używasz jakiegoś programu do zarządzania klientami, wystawiania faktur itp. Gratulacje! masz już swoje źródło danych. Źródłem danych będzie rzeczywisty ERP lub rzeczywista transakcyjna baza danych, w której program lub programy, z których korzystasz, gromadzą dane. Dla tych, którzy jeszcze się nad tym zastanawiają, w tej części przedstawiamy, jak zainstalować i bawić się Odoo, aby zbierać dane do rozwoju naszego systemu BI. Jak wyjaśniono we wstępie, do zainstalowania Odoo użyjemy komputera z systemem Linux. Zdecydowaliśmy się na serwer Ubuntu w wersji 16.04 LTS. Dzieje się tak dlatego, że jest to wersja Long Term Support (LTS), co oznacza, że będzie obsługiwana przez dłuższy czas i powinna być bardziej stabilna. Instalacja Odoo może być nieco skomplikowana ze względu na różne wymagania wstępne, więc jest to raczej zadanie dla administratora systemu. Jeśli go nie masz lub nie możesz sobie pozwolić na jego wynajęcie, zalecamy skorzystanie z uproszczonej wersji, którą przedstawimy w kolejnych akapitach. Jeśli chcesz zainstalować go od zera, przed instalacją upewnij się, że zainstalowałeś bazę danych PostgreSQL, serwer WWW Apache i że wersja Pythona jest zgodna z wersją wymaganą przez instalowaną wersję Odoo.

Pakiet Bitnami Odoo

Do naszego wdrożenia użyjemy samodzielnego instalatora Bitnami, który zainstaluje i skonfiguruje dla nas Odoo. Ten pakiet zawiera bazę danych PostgreSQL, serwer Apache i pliki programu. Dla tych, którzy

są bardziej zainteresowani wypróbowaniem go najpierw i nie chcą wdrażać pakietu na istniejącej maszynie, Bitnami oferuje również maszynę wirtualną ze wszystkim już zainstalowanym, w tym z serwerem Ubuntu. Aby pobrać pakiet Bitnami, otwórz przeglądarkę i sprawdź następujący adres URL: https://bitnami.com/stack/odoo/installer. Jeśli chcemy zainstalować pakiet samodzielnie, to musimy wybrać opcję instalatorów z paska menu. A następnie wybierz ten odpowiedni dla naszego systemu operacyjnego. Dla tych, którzy chcą najpierw wypróbować, zamiast wybierać opcję Instalatory w menu, wybierz Maszyny wirtualne i pobierz obraz Ubuntu VM zawierający zainstalowany program. Aby użytkownik i hasło w pakiecie mogli zalogować się do urządzenia, przeczytaj uważnie stronę .

Pobieranie i instalacja Odoo

Odtąd omówimy podstawy instalacji pakietu . Po znalezieniu odpowiedniej wersji uzyskamy adres URL. Ponownie, w momencie pisania link do pobrania jest następujący:

https://downloads.bitnami.com/files/stacks/Odoo/9.0.20160620-1/bitnami-Odoo-9.0.20160620-1-linux-x64-installer.run

Ale mogło się to zmienić, jeśli dostępna jest nowa wersja. Za pomocą tego adresu URL przejdziemy do naszego systemu Linux i użyjemy wget lub curl, aby pobrać instalator. Najpierw przechodzimy do katalogu /tmp, aby tam pobrać pakiet:

cd / tmp

Następnie wywołujemy wget, aby pobrać instalator:

wget

I rozpocznie się pobieranie. Po kilku sekundach lub minutach (w zależności od połączenia sieciowego) plik zostanie pobrany do urządzenia.

Przed uruchomieniem pliku musimy uczynić go wykonywalnym, więc robimy chmod

+x:

bibook@bibook:/tmp$ chmod +x bitnami-Odoo-9.0-3-linux-x64-installer.run

Tam uruchomimy instalator, uruchamiając następujące czynności, używając sudo do uruchomienia go jako root, ponieważ będziemy go potrzebować do zainstalowania go w innym katalogu

bibook@bibook:/tmp$ sudo ./bitnami-Odoo-9.0-3-linux-x64-installer.run

I postępujemy zgodnie z instrukcjami na ekranie. Po uruchomieniu instalator zasugeruje ścieżkę w katalogu domowym. Używamy /opt, aby go zainstalować, więc zmień go na wypadek, gdybyś go nie miał. Pod koniec procesu instalacji instalator zapyta nas, czy chcemy uruchomić usługi i komponenty Odoo. Wybieramy tak (Y).

Pliki konfiguracyjne Bitnami i Odoo

W tym podrozdziale przejrzymy miejsce na wszystkie pliki konfiguracyjne i skrypty startowe, abyśmy mogli kontrolować naszą instalację Odoo. W przypadku, gdy chcemy zatrzymać lub rozpocząć później na stosie, istnieje skrypt rządzący całą instalacją o nazwie ctlscript.sh znajdujący się na górze ścieżki instalacyjnej, w naszym przypadku /opt/odoo-9.0-3. Uruchomienie skryptu jako użytkownik root pozwala nam uruchamiać i zatrzymywać powiązane usługi:

```
bibook@bibook:/opt/odoo-9.0-3$ sudo ./ctlscript.sh
usage: ./ctlscript.sh help
       ./ctlscript.sh (start|stop|restart|status)
       ./ctlscript.sh (start|stop|restart|status)
Postgresql
       ./ctlscript.sh (start|stop|restart|status)


Apache
       ./ctlscript.sh (start|stop|restart|status)
openerp_background_worker
       ./ctlscript.sh (start|stop|restart|status)
openerp_gevent

  help       - this screen
  start      - start the service(s)
  stop       - stop  the service(s)
  restart    - restart or start the service(s)
  status     - show the status of the service(s)
```

Ale w tym momencie mamy już uruchomione usługi, więc nie musimy nic robić. Oprócz usług jest jeszcze jeden ważny plik, który należy znać, czyli domyślna konfiguracja usług, w tym porty, adresy URL, ścieżki lokalne i inne ważne rzeczy. Plik nazywa się properties.ini i ponownie potrzebujemy uprawnień administratora, aby go zobaczyć. Można go znaleźć w katalogu głównym. Domyślnie porty używane przez aplikację to standardowe porty używane przez zainstalowane usługi. W naszym przypadku serwer Apache nasłuchuje na porcie 80, podczas gdy baza danych PostgreSQL nasłuchuje na domyślnym porcie 5432. Oprócz tych plików konfiguracyjnych usługi istnieją inne ważne pliki. Być może najważniejszym z nich jest plik konfiguracyjny aplikacji, który można znaleźć w /opt/odoo-9.0-3/apps/odoo/conf/openerp-server.conf Jest to ważny plik, ponieważ zawiera kilka losowo wygenerowanych haseł niezbędnych do połączyć się z aplikacją. W tym pliku będziemy mogli znaleźć nazwę użytkownika i hasło do bazy danych, nazwę bazy danych oraz hasło lub konto administratora. Wartości, które musimy zanotować z tego pliku, wraz z ich znaczeniem, znajdują się w tabeli

Wartość: Opis

admin_passwd : Domyślne hasło administratora do logowania w interfejsie WWW

db_host : Adres IP serwera bazy danych, którym będzie host lokalny

db_name : Nazwa bazy danych, domyślnie bitnami_openerp

db_password : domyślne hasło do bazy danych

db_port : domyślny port, 5432

db_user : użytkownik bazy danych, domyślnie bn_openerp

Instalowanie psql i sprawdzanie połączenia z bazą danych

Ostatnim zadaniem do wykonania przed połączeniem z programem jest instalacja klienta psql, dzięki czemu możemy uzyskać dostęp do bazy danych PostgreSQL z linii poleceń. Jeśli używamy Ubuntu, można to osiągnąć, uruchamiając następujące polecenia:

sudo apt-get install postgresql-client-common postgresql-clientv

Następnie możemy przetestować połączenie i sprawdzić, czy wszystko działa poprawnie, łącząc się z klientem psql:

bibook@bibook:/opt/Odoo-9.0-3/apps/Odoo/conf$ psql

-U bn_openerp -h host lokalny -d bitnami_openerp

Zostaniemy poproszeni o wartość db_password; wprowadzamy go i powinniśmy zobaczyć monit klienta psql. Jeśli do tego dojdziemy, wszystko jest w porządku.

Password for user bn_openerp:

psql (9.5.2, server 9.4.6)

Type "help" for help.

bitnami_openerp=#

Teraz skończyliśmy. Jesteśmy gotowi do odwiedzenia naszego nowo zainstalowanego ERP!

Uwaga: Jeśli próbujesz uzyskać dostęp do bazy danych z komputera zewnętrznego, musisz najpierw włączyć dostęp zdalny. W takim przypadku musisz zmodyfikować plik konfiguracyjny PostgreSQL, aby baza danych nasłuchiwała pod wszystkimi interfejsami (*) i zezwoliła użytkownikom na zdalne logowanie

Dostęp do aplikacji

Po wykonaniu wszystkich czynności konfiguracyjnych możemy zalogować się do aplikacji. Domyślnie możemy zalogować się za pomocą adresu URL hosta lokalnego: http://localhost Zostanie wyświetlony ekran powitalny z prośbą o podanie danych uwierzytelniających. W tym momencie musimy użyć nazwy użytkownika wybranej podczas instalacji oraz użyć wartości admin_passwd z pliku konfiguracyjnego openrp-server.conf. Po wejściu właśnie zalogowaliśmy się przy użyciu konta administratora.

Konfigurowanie i instalowanie modułów

Domyślnie instalacja konfiguruje nasze Odoo z włączonymi kilkoma modułami, ale może się zdarzyć, że będziemy chcieli je dostosować, wybierając moduły do zainstalowania, wybierając, czy chcemy zainstalować przykładowe dane i tak dalej. Jak widać w dolnej części obrazka, dostępna jest opcja Zarządzaj bazami danych, która pozwala nam rozpocząć nowy projekt Odoo od zera. Wybierzemy tę opcję, ponieważ pozwoli nam to założyć nową bazę danych Odoo i zainstalujemy moduł Sales, aby używać go jako przykładu podczas rezerwacji. Jednocześnie w nowej instalacji możemy polecić Odoo załadowanie niektórych danych testowych. Jest to interesujące, ponieważ możemy go użyć do celów ilustracyjnych, chociaż niestety wstawia bardzo mało rekordów. Okno dialogowe tworzenia bazy danych pyta nas o hasło główne, nazwę nowej bazy danych, język bazy danych i hasło administratora. Mamy również pole wyboru, aby załadować przykładowe dane, więc będziemy je sprawdzać. Hasło główne to admin_passwd natomiast hasło administratora może być wybrane przez nas. Ten proces zajmuje trochę czasu, ponieważ program tworzy i wypełnia nową bazę danych, więc prosimy o cierpliwość i poczekanie do końca. Po zainstalowaniu bazy danych możemy wrócić do ekranu głównego i zalogować się nazwą użytkownika: admin i hasłem, które ustawiliśmy dla administratora w oknie dialogowym nowej bazy danych. Po zalogowaniu system wyświetli listę aplikacji dostępnych do zainstalowania. W tym projekcie książki będziemy używać głównie modułu sprzedaży, więc zainstalujmy go i skonfigurujmy, naciskając Zainstaluj w module Zarządzanie sprzedażą na ekranie głównym. Ponownie będziemy musieli trochę poczekać, ponieważ niektóre skrypty muszą utworzyć tabele potrzebne dla modułu i wstępnie wypełnić niektóre z nich danymi testowymi. Czas na przerwę

na kawę! Po chwili instalacja zostanie zakończona i zobaczymy, że domyślnie moduł Sales Management zainstalował również jako warunek wstępny moduł Discus i Fakturowanie. To idealnie, ponieważ będziemy pracować z fakturami. Teraz nadszedł czas, aby sprawdzić dane testowe i zobaczyć naszych klientów oraz powiązane z nimi faktury. Jeśli teraz odświeżymy główny ekran, zobaczymy, że mamy nowe opcje w górnym menu, w tym Sprzedaż, więc klikamy na to i przechodzimy do Klienci, aby przejrzeć naszą bazę klientów. Jak zobaczysz, mamy teraz kilku klientów dostępnych . Klikając na każdą firmę, pojawia się nowy ekran pokazujący nam kontakty firmy i kilka innych ciekawych opcji, takich jak dane klienta, suma zafakturowana, liczba faktur i tak dalej. Po zainstalowaniu i sprawdzeniu, czy aplikacja działa poprawnie, możemy przystąpić do nauki z modelu danych. Odoo ma złożony model danych, ale na szczęście kod źródłowy jest czysty, dostęp do bazy danych nie jest skomplikowany, aw Internecie jest kilka przydatnych zasobów, które nieco uszczegóławiają model. Dzięki tym fragmentom powinniśmy być w stanie zrozumieć podstawy podstawowych tabel i dwóch modułów, nad którymi będziemy pracować: Zarządzanie sprzedażą i Fakturowanie. W poniższych akapitach szczegółowo omówimy ważne tabele i model, do którego należą tabele; sprawdzimy kolumny i wiersze tych tabel i porównamy to, co jest zapisane w bazie danych, z tym, co widzimy w aplikacji. Jest jeszcze jedna rzecz do zrobienia. Przykładowe dane, które zainstalowaliśmy, są nieco ubogie i nie obejmują wszystkich możliwych scenariuszy. Na przykład nie ma opłaconych faktur: wszystkie mają status oczekujący. Potrzebujemy trochę więcej danych, aby móc wykonać raportowanie, więc utworzymy kilka rekordów i dołączymy je do bieżących danych testowych. W tym celu będziemy musieli utworzyć dane testowe, ale zajmiemy się tym później, gdy zaczniemy pobierać dane z Odoo.

Wybór naszej bazy danych hurtowni danych

Istnieje wiele opcji wyboru systemu RDBMS, który ma stać się naszą bazą danych hurtowni danych. Albo to, albo możemy użyć bazy danych NoSQL. Ale to jeszcze nie jest trend, a większość narzędzi do raportowania, w tym te, których będziemy używać, jest zaprojektowana do pracy z relacyjnymi bazami danych. Chociaż ten paradygmat może ulec zmianie w przyszłości, prawie wszystkie obecnie hurtownie danych oparte są na relacyjnych bazach danych. Istnieje dobra strona internetowa do śledzenia najczęściej używanych baz danych, dzięki czemu możesz sam zobaczyć, jakie trendy są obecnie na rynku i jakie opcje należy rozważyć. Kiedy już jesteśmy świadomi, że potrzebujemy relacyjnej bazy danych, nadszedł właściwy czas na jej wybór. Gdybyśmy mieli przyzwoity budżet i dużo danych do przetworzenia, prawdopodobnie będziemy patrzeć na komercyjną bazę danych. Te wydają się być potężne i pomimo faktu, że istnieją pewne bazy danych typu open source, które również dobrze się skalują, sensowne byłoby trzymanie się wersji prywatnej. Pomyśl szczególnie o wsparciu, aktualizacjach i rozwiązywaniu problemów, a także o liczbie ekspertów dostępnych na rynku, co jest bardzo ważnym czynnikiem, który musisz wziąć pod uwagę, decydując się na użycie jednej technologii lub aplikacji zamiast drugiej. Jeśli zdecydujemy się na małe wdrożenie, sensowne może być użycie PostgreSQL, jak widzieliśmy wcześniej; to ten, który jest używany do przechowywania metadanych Odoo i jest bezpośrednio dołączony do pakietu Bitnami. Jeśli zdecydowaliśmy się na ręczną instalację Odoo, mieliśmy również możliwość dostarczenia bazy danych MySQL dla metadanych. W większości przypadków ta decyzja dotyczy tego, jakie technologie już posiadamy w firmie (lub licencje) i jaka jest główna wiedza specjalistyczna, jaką posiadamy w firmie w zakresie baz danych. Nie ma sensu zakładać hurtowni danych PostgreSQL, jeśli nie mamy w firmie nikogo, kto mógłby się nią zająć, gdy pojawią się problemy (zaufaj mi, zrobią to!). Zalecamy zachowanie prostoty i wdrożenie bazy danych, do której jesteśmy przyzwyczajeni. Jeśli pracujemy w środowisku bazodanowym Oracle to instalacja bazy danych Oracle ma sens, ale czasami jest to niewykonalne. Jako mała firma możemy zainstalować jedną z bezpłatnych (i ograniczonych) wersji komercyjnej bazy danych, którą widzieliśmy w części 3. Oracle XE może być dobrym wyborem, ale należy zachować ostrożność, ponieważ, jak widzieliśmy w części 3, pewne ograniczenia pamięci, procesora i miejsca, więc myślenie o przyszłości zależy od naszego

uzasadnienia biznesowego, aby zdecydować, czy to wystarczy, czy nie. To samo dotyczy Microsoft SQL Server Express. Zakładając, że jesteśmy małą firmą i nie chcemy wydawać pieniędzy na żadną licencję, opcje ograniczają się do jednej z bezpłatnych wersji komercyjnych narzędzi lub open source lub bezpłatnej bazy danych. W naszym przypadku będziemy instalować MySQL/MariaDB dla naszego wdrożenia. Jest kilka dobrych powodów, aby to zrobić:

* Jest to bezpłatna baza danych (lub open source w przypadku MariaDB).

* Jest najbardziej rozbudowany i łatwy w administrowaniu i utrzymaniu. Jest wielu ludzi, którzy wiedzą, jak to zrobić, i mnóstwo informacji w sieci.

* Dobrze się skaluje

Ale jak zawsze wybierz to, co najbardziej Ci odpowiada. W naszym przypadku będzie to MariaDB, ponieważ jest to prawdziwie otwarta baza danych, a najlepsze jest to, że jest kompatybilna z MySQL, więc przy użyciu tych samych narzędzi lub sterowników, które musisz połączyć z MySQL, można również użyć MariaDB. Zrezygnowaliśmy z używania PostgreSQL dołączonego do pakietu Bitnami Odoo jako bazy danych hurtowni danych, ponieważ chcemy, aby źródłowa baza danych ERP była odizolowana od bazy danych hurtowni danych. W bardzo małym wdrożeniu prawdopodobnie nie jest to bardzo ważne, ale myślenie w większym wdrożeniu może spowodować niepowodzenie całego projektu. Dostęp do bazy danych PostgreSQL będzie możliwy kilka razy na sekundę prawdopodobnie przez narzędzie ERP (transakcyjne). Więc istnieje pewna presja na tę bazę danych. Jeśli połączymy również nasze narzędzia BI i procesy ETL z tą samą bazą danych, może to mieć wpływ na narzędzie transakcyjne. Jest to niepożądana konsekwencja korzystania z tej samej bazy danych, ponieważ chcemy zapewnić jak najwyższą dostępność, zwłaszcza dla naszych programów operacyjnych. Ponieważ hurtownie danych są zwykle mniej wrażliwe niż nasze programy operacyjne, zainstalujemy dedykowaną bazę danych, która będzie działać jako hurtownia danych dla naszej platformy. Jest to dość powszechny scenariusz we wdrożeniach BI.

Pozyskiwanie i instalacja MariaDB

Istnieje wiele sposobów instalacji MariaDB. W przypadku systemu Linux kilka dystrybucji ma go już jako domyślną bazę danych w narzędziach menedżera oprogramowania. Inni nadal mają MySQL i będą tęsknić za repozytoriami MariaDB, a inni mogą zmusić cię do ręcznej instalacji pakietów, używając prekompilowanego pakietu lub samodzielnie kompilując źródła. Pomimo tego, co powiedzieliśmy w poprzednich podrozdziałach, kiedy wyjaśniliśmy, że naprawdę powinieneś zainstalować bazę danych hurtowni danych na osobnej maszynie, ponieważ mamy małe środowisko testowe, będziemy instalować ją na tej samej maszynie. Możemy to zrobić, ponieważ nie mamy żadnego innego serwera MySQL ani MariaDB działającego na tej maszynie, ponieważ pakiet Bitnami używa bazy danych PostgreSQL zamiast MySQL/MariaDB i używają innych portów, więc nie musimy nawet martwić się o nich. Dlatego dla uproszczenia tym razem będziemy używać tej samej maszyny, ale pamiętajmy o wcześniejszych rozważaniach.

Instalacja dla Windowsa

Jeśli korzystasz z systemu Windows , możesz bezpośrednio pobrać pliki binarne ze strony internetowej MariaDB i zainstalować je w zwykły sposób, na przykład dla aktualnej wersji w momencie pisania tej książki wyglądało to następująco:

https://downloads.mariadb.org/mariadb/

Będziesz musiał pobrać pakiet msi dla systemu Windows (32- i 64-bitowy, niezależnie od tego, jaki masz gust).

Instalacja na Linuksie

Chociaż możesz tam również pobrać pliki źródłowe dla Linuksa, będziemy pobierać już skompilowane pakiety dla naszej dystrybucji. Wyjaśniamy tutaj, jak dodać repozytoria MariaDB do obecnego pakietu menedżera oprogramowania dla kilku dystrybucji, najpopularniejszych, ale poszukaj instrukcji na stronie internetowej MariaDB dla każdego innego rodzaju wdrożenia. W przypadku obu dystrybucji będziemy używać narzędzia do konfiguracji repozytorium dostępnego na stronie internetowej MariaDB i dostępnego pod następującym adresem URL: https://downloads.mariadb.org/mariadb/repositories.

Instalacja w Ubuntu

Instalacja w Ubuntu jest prosta, ponieważ będziemy używać menedżera pakietów apt-get. Jednak domyślnie w starszych instalacjach Ubuntu musimy dodać repozytoria MariaDB, ponieważ nie są one domyślnie dostępne. Jeśli używamy aktualnej wersji lub nowszej niż 14.04, możemy pominąć tę część, ponieważ te pliki binarne są już załadowane na liście dystrybucyjnej pakietów. W przypadku wcześniejszych wersji musimy je najpierw dodać. Musimy przejść do narzędzia konfiguracji repozytorium i na liście dystrybucyjnej wybrać Ubuntu, wybrać wersję, którą mamy, wersję MariaDB i lustro, którego będziemy używać. Tak więc pierwszy krok będzie polegał na wykonaniu następujących poleceń w celu dodania repozytorium do naszej pamięci podręcznej apt:

sudo apt-get install software-properties-common

sudo apt-key adv --recv-keys --keyserver

hkp://keyserver.ubuntu.com:80 0xF1656F24C74CD1D8

sudo add-apt-repository 'deb

[arch=amd64,i386,ppc64el]

http://tedeco.fi.upm.es/mirror/mariadb/repo/10.1/ubunt

u xenial main'

Po zakończeniu możemy przystąpić do aktualizacji naszej pamięci podręcznej apt i zainstalowania programu, tak jak zrobimy to z aktualną wersją dystrybucji:

sudo apt update

sudo apt install mariadb-server

Gdy wykonamy następujące czynności, pojawi się nowy ekran z prośbą o pobranie nowego oprogramowania, a my potwierdzamy za pomocą „Y”. Po kilku sekundach (lub minutach), w zależności od mocy komputera, wszystko powinno zostać zainstalowane.

Po zakończeniu instalacji nasz serwer MariaDB działa, ale nadal należy wykonać kilka kroków. Jeśli instalacja jest w starej wersji i instalujesz MariaDB 5.5, musisz wykonać dodatkowy krok. Po instalacji musisz uruchomić (ten sam) skrypt, co w MySQL, aby zabezpieczyć instalację. Skrypt nazywa się ysql_secure_installation (wywołaj go sudo mysql_secure_installation) i powinien być dostępny z twojej ścieżki. Ten skrypt umożliwia zmianę hasła roota i usunięcie niektórych niepotrzebnych użytkowników.

Być może zostałeś wcześniej poproszony o ustawienie hasła; w takim przypadku możesz pominąć pierwszą część skryptu. Pozostałe rzeczy, o które zostaną poproszeni, to:

Change the root password? [Y/n]

Remove anonymous users? [Y/n]

Disallow root login remotely? [Y/n]

Remove test database and access to it? [Y/n]

Reload privilege tables now? [Y/n]

Sugerujemy odpowiedź Y, aby zmienić hasło roota, chyba że ustawiłeś je podczas instalacji, usunąć wszystkich anonimowych użytkowników, wyłączyć zdalne logowanie, JEŚLI instalacja Odoo nie została przeprowadzona na osobnym komputerze, w tym przypadku odpowiedź Nie; w przeciwnym razie będziesz musiał później dostosować uprawnienia dla użytkowników, aby usunąć testowe bazy danych, ponieważ nie są one potrzebne i przeładować tabele uprawnień po operacjach. Następnie nadszedł czas, aby spróbować zalogować się na serwerze i możemy użyć narzędzia poleceń, aby przetestować łączność i konfigurację. Nie jest to konieczne, ale zalecamy ponowne uruchomienie usług. Można to osiągnąć za pomocą następujących poleceń:

sudo service mysql stop && sudo service mysql start

Następnie możesz sprawdzić stan usługi, wykonując:

sudo service mysql status

I połącz się z bazą danych z klientem:

mysql -u root –p

Program poprosi o hasło roota, które właśnie ustawiłeś w poprzednich krokach. Jeśli zobaczysz coś w rodzaju banera MYSQL „Welcome to the MariaDB monitor…”, wszystko jest w porządku i gotowe do stworzenia pierwszej bazy danych!

Uwaga: w nowszych wersjach MariaDB może nie być konieczne uruchamianie skryptu sekurytyzacji po zakończeniu instalacji, ponieważ niektóre z tych zadań, takie jak zmiana hasła roota, są już wbudowane w główny instalator. W takim przypadku po prostu pomiń tę sekcję dokumentu i przejdź od razu do przetestowania łączności.

Instalacja MariaDB w Centos

Omówiliśmy już instalację MariaDB w Windows i Ubuntu, ale dla innych osób korzystających z Centos, Fedory lub Redhat Enterprise, tutaj omawiamy podstawy instalacji Domyślnie Centos 7 jest dostarczany z instalacją MariaDB 5.5 (taką samą, którą zainstalowaliśmy w Ubuntu), więc proces będzie podobny, ale w tym przypadku przy użyciu menedżera pakietów yum. Czas zacząć!

```
   yum install mariadb-server mariadb
   ================================================
==========================
    Package                    Architecture    Versio
n          Repository  Size
   ================================================
==========================
   Installing:
    mariadb                    x86_64     1:5.5.47-
1.el7_2      updates     8.9 M
    mariadb-server             x86_64     1:5.5.47-
1.el7_2      updates      11 M
   Instalando para las dependencias:
    perl-Compress-Raw-Bzip2    x86_64     2.061-
3.el7           base         32 k
    perl-Compress-Raw-Zlib     x86_64     1:2.061-
4.el7          base         57 k
    perl-DBD-MySQL             x86_64     4.023-
5.el7           base        140 k
    perl-DBI                   x86_64     1.627-
4.el7           base        802 k
    perl-IO-Compress           noarch     2.061-
2.el7           base        260 k
    perl-Net-Daemon            noarch     0.48-
5.el7            base         51 k
    perl-PlRPC                 noarch     0.2020-
14.el7         base         36 k

   Transaction summary
   ================================================
==========================
   Install 2 Packages (+7 Dependant packages)



  Total size to download: 21 M
  Total size installed: 107 M
  Is this ok [y/d/N]:y
```

Następnie instalacja zostanie zakończona, a następnie możemy uruchomić serwer za pomocą:

systemctl start mariadb

A teraz możemy zabezpieczyć naszą instalację w taki sam sposób, jak w przypadku instalacji Ubuntu.

mysql_secure_installation

Odpowiadamy tak samo jak w wersji Ubuntu i po tym jesteśmy gotowi do uruchomienia klienta i przetestowania połączenia

mysql -u root -p

I znowu powinniśmy zobaczyć słynne „Witamy w monitorze MariaDB…” Jeśli nie, czytaj dalej, aby rozwiązać problemy.

Uwaga: ponownie zauważ, że jeśli zdecydowaliśmy się zainstalować MariaDB z jednego z oficjalnych repozytoriów, spowoduje to zainstalowanie wersji 10.1 z niewielkimi zmianami w procedurze instalacji

Rozwiązywanie problemów z łącznością

Możliwe, że domyślnie nie możemy połączyć się z komputera zewnętrznego z naszą nową instalacją MySQL. Niektóre pakiety wyłączają to, określając adres sprzężenia zwrotnego jako adres powiązania dla silnika MySQL. W takim przypadku rozwiązanie jest dość proste, musimy edytować plik /etc/mysql/my.cnf za pomocą naszego pożądanego edytora (vi, nano…) i szukać adresu, który określa, którego IP nasłuchuje baza danych , który będzie adresem IP sprzężenia zwrotnego, mniej więcej tak:

bind-address: 127.0.0.1

I zmień to, wiążąc go ze wszystkimi interfejsami lub z adresem IP interfejsu, który chcesz powiązać (zwykle eth0), ale może mieć inną nazwę, stąd plik powinien teraz czytać

bind-address: 0.0.0.0

Lub

bind-address: (wpisz tutaj ip twojej sieci interfejs, z którym chcesz się połączyć)

Potem pozostaje nam tylko zrestartować usługę…

sudo service mysql restart

I możemy przejść, określając adres IP, z którym chcemy się połączyć mysql -u root -p (twoje hasło tutaj bez spacji) - h (tutaj twoje ip ze spacją)

tj.

mysql -u root -pp4ssw0rd -h 192.168.2.10

Tworzenie naszej pierwszej bazy danych

Nadszedł czas, aby połączyć się z naszą instalacją MariaDB i zacząć się z nią bawić. Celem jest przekształcenie tej bazy danych w naszą nową hurtownię danych, z której będziemy pobierać dane z wielu źródeł. System transakcyjny, w naszym przypadku nasza nowa aplikacja Odoo, będzie zawierał szczegóły dotyczące klientów, produktów, zamówień, faktur itp., które chcemy przeanalizować. Te wraz z innymi informacjami zewnętrznymi zostaną wyodrębnione i załadowane do naszego

hurtownia danych. W kolejnych rozdziałach zobaczymy, jak wyodrębnić te informacje, i opracujemy kilka procesów, które zajmą się nimi w całej książce, ale najpierw musimy przygotować naszą bazę danych. Na potrzeby naszej realizacji postanowiliśmy stworzyć dwie bazy danych. Nie mamy bazy danych ODS, więc będziemy nazywać naszą inscenizację, nawet jeśli nie ma żadnych/kilka transformacji ze źródłowej bazy danych. W tym celu z dwóch baz danych, które mają zostać utworzone, jedna będzie tak zwaną bazą pomostową, która będzie emulować bazę danych ODS, w której zwykle umieszczane są surowe dane, a druga będzie zawierała bieżącą hurtownię danych. Jest to powszechna konfiguracja, ponieważ zwykle potrzebujemy miejsca, aby najpierw umieścić nasze dane, które będą pochodzić z wielu źródeł, zanim zostaną przekształcone w ostateczny kształt. W niektórych przypadkach możliwe jest napisanie skryptu ETL, który wykona całą tę pracę i może bezpośrednio umieścić dane w hurtowni danych, ale jest to zawsze dobre rozwiązanie. Argumentów przeciw jest wiele:

* Czasami trzeba mieszać dane z wielu źródeł, co utrudnia sprawę, ponieważ to samo narzędzie nie zawsze może mieszać wszystkie pojedyncze elementy danych.

* Wydajność jest zwykle gorsza, jeśli pracujemy z zewnętrznymi źródłami danych, a także dostępność może być zagrożona.

* Łatwiej jest wyleczyć się z błędów. Zwykle dane źródłowe są w formacie nieprzetworzonym i jak widzieliśmy w poprzednich rozdziałach, mogą nie być poprawne. Musimy trochę posprzątać, a wstawianie bezpośrednio do naszych tabel końcowych może naruszyć już sformatowane dane i znacznie utrudnić odzyskiwanie, a nawet spowodować utratę danych.

Mając to na uwadze, możemy przystąpić do tworzenia naszych dwóch baz danych i do tego nie będziemy bardzo oryginalni; nazwijmy je staging i dwh, co oznacza DataWareHouse. Polecenia, aby to osiągnąć, są następujące:

Najpierw musimy ponownie połączyć się z naszą bazą danych na wypadek rozłączenia:

mysql -u root -p

Utworzymy dwóch użytkowników z uprawnieniami tylko do tych samych baz danych. Każdy użytkownik będzie miał uprawnienia tylko do własnej bazy danych. Później w kolejnych rozdziałach utworzymy użytkowników, którzy będą mieli uprawnienia do interakcji między dwiema bazami danych, ale w tej chwili ich nie potrzebujemy.

Pokaż polecenia baz danych i tabel

Zacznijmy od utworzenia dwóch baz danych. Ponieważ jesteśmy zalogowani jako root, nie będziemy mieli żadnych problemów z utworzeniem dwóch baz danych. Polecenia, które musimy uruchomić, są następujące:

CREATE DATABASE IF NOT EXISTS dwh;

CREATE DATABASE IF NOT EXISTS staging;

Te dwa polecenia utworzą nam dwie bazy danych, ale nadal nic więcej. Żaden użytkownik, ponieważ jeszcze ich nie utworzyliśmy, nie będzie miał dostępu (z wyjątkiem naszego użytkownika root), a nawet jeśli użytkownicy już istnieją, nie będzie miał dostępu, ponieważ nie nadaliśmy im uprawnień. Dodajmy dwóch użytkowników, a następnie uprawnienia dla każdego z nich, aby uzyskać dostęp do własnej bazy danych.

Tworzenie użytkowników dla dwóch nowych baz danych

Stworzyliśmy bazy danych, teraz potrzebujemy użytkowników. Tworzenie użytkowników w MariaDB jest dość proste, ale musimy określić, czy użytkownik będzie lokalny (będzie miał dostęp do bazy danych tylko z tego samego serwera), czy będzie to użytkownik zdalny. W zależności od wybranej konfiguracji musimy wykonać jedną z dwóch opcji: Jeśli nasz użytkownik będzie miał dostęp do bazy danych lokalnie, możemy utworzyć użytkownika lokalnie. Oświadczenia są następujące:

CREATE USER 'dwh'@'localhost' IDENTIFIED BY

'p4ssw0rd';

CREATE USER 'staging'@'localhost' IDENTIFIED BY

'p4ssw0rd';

Jeśli nasi użytkownicy potrzebują dostępu z zewnątrz, co jest prawdopodobne, możemy dodać symbol wieloznaczny (%) i będą mogli łączyć się z dowolnego miejsca. Obaj użytkownicy mogą współistnieć w tej samej bazie danych.

CREATE USER 'dwh'@'%' IDENTIFIED BY 'p4ssw0rd';

CREATE USER 'staging'@'%' IDENTIFIED BY 'p4ssw0rd';

Dzięki temu nasi użytkownicy są gotowi do dostępu. Ale w tym momencie nie będą mogli operować na bazach danych. Przydzielmy każdemu wymagane uprawnienia do ich baz danych.

Uwaga

Później, kiedy przejdziemy do części ETL i Reporting, ci użytkownicy będą wykorzystywani, więc jest bardzo prawdopodobne, że użytkownicy będą potrzebować dostępu z zewnątrz, więc weź to pod uwagę. W każdym razie można to zmienić w razie potrzeby, więc nie martw się o to teraz, ale po prostu miej to pod ręką na wypadek problemów z połączeniem.

Udziel uprawnień do baz danych

Stworzyliśmy bazy danych i użytkowników, więc teraz czas na połączenie tych dwóch pojęć. W tym celu musimy zezwolić użytkownikom na manipulowanie bazami danych. Struktura uprawnień w MariaDB/MySQL jest dość łatwa do zrozumienia. Ma dobrze zdefiniowany format:

GRANT [type of permission] ON [database name].

[table name] TO '[username]'@'%';

Aby cofnąć uprawnienia, składnia jest bardzo podobna, zmieniając tylko słowa kluczowe:

REVOKE [type of permission] ON [database name].

[table name] FROM '[username]'@'%;

Zwróć uwagę na symbol wieloznaczny % na końcu, który musi zostać zastąpiony przez localhost, jeśli pracujemy z użytkownikami o dostępie lokalnym. W naszym przypadku przyznamy uprawnienia naszym lokalnym i zewnętrznym użytkownikom:

MariaDB [(none)]> GRANT ALL PRIVILEGES ON dwh.* TO

'dwh'@'localhost';

Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON dwh.* TO

'dwh'@'%';

Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON staging.*

TO 'staging'@'localhost';

Query OK, 0 rows affected (0.00 sec)

MariaDB [(none)]> GRANT ALL PRIVILEGES ON staging.*

TO 'staging'@'%';

Query OK, 0 rows affected (0.00 sec)

Gdy to zostanie osiągnięte, możemy przystąpić do testowania naszych użytkowników, w tym celu opuszczamy naszą sesję root i testujemy obu użytkowników. Teraz zamiast tego należy nawiązać połączenie z nowym użytkownikiem:

mysql -u dwh -pp4ssw0rd

Wszystko idzie dobrze i powinniśmy zobaczyć naszego klienta zalogowanego:

Welcome to the MariaDB monitor. Commands end with

; or \g.

Your MariaDB connection id is 14

Server version: 5.5.47-MariaDB MariaDB Server

Copyright (c) 2000, 2015, Oracle, MariaDB

Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear

the current input statement.

Następnie możemy sprawdzić, które bazy danych użytkownik widzi

```
MariaDB [(none)]> show databases;
+--------------------+
| Database           |
+--------------------+
| information_schema |
| dwh                |
+--------------------+
2 rows in set (0.00 sec)
```

And then change to our database:

```
MariaDB [(none)]> use dwh;
Database changed
```

And then check that no tables are present yet:

```
MariaDB [dwh]> show tables;
Empty set (0.00 sec)
```

A następnie przejdź do naszej bazy danych:

MariaDB [(none)]> use dwh;

Database changed

A następnie sprawdź, czy nie ma jeszcze żadnych tabel:

MariaDB [dwh]> show tables;

Empty set (0.00 sec)

Następnie możemy wyjść i przetestować drugiego użytkownika, wykonując dokładnie tę samą procedurę:

MariaDB [dwh]> exit;

Bye

Połączymy się teraz jako użytkownik pomostowy i sprawdzimy to dopiero po zobaczeniu bazy danych pomostowych:

<img src="images/maria2.png">

```
   [root@localhost anogues]# mysql -u staging -
pp4ssw0rd
   Welcome to the MariaDB monitor.  Commands end with
; or \g.
   Your MariaDB connection id is 15
   Server version: 5.5.47-MariaDB MariaDB Server

   Copyright (c) 2000, 2015, Oracle, MariaDB
Corporation Ab and others.

   Type 'help;' or '\h' for help. Type '\c' to clear
the current input statement.

   MariaDB [(none)]> show databases;
   +--------------------+
   | Database           |
   +--------------------+
   | information_schema |
   | staging            |
   +--------------------+
   2 rows in set (0.00 sec)

   MariaDB [(none)]> use staging;
   Database changed
   MariaDB [staging]> show tables;
   Empty set (0.00 sec)

   MariaDB [staging]> exit;
   Bye
```

Uwaga

Przedstawiliśmy wszystkie polecenia przy użyciu interakcji klienta z wiersza poleceń. Niedoświadczonym użytkownikom może się opłacać wypróbowanie narzędzia GUI do interakcji z bazami danych MySQL/MariaDB. Jest ich wiele za darmo, będąc oficjalnym środowiskiem roboczym MySQL, prawdopodobnie najbardziej znanym: https://www.mysql.com/products/workbench/, który działa w kilka systemów operacyjnych, ale istnieją inne dobre programy, takie jak HeidiSQL i dbForge dla systemów Windows, które są bezpłatne lub przynajmniej mają darmową edycję.

Analiza źródła danych

Nasza hurtownia danych jest już gotowa do działania. Nadszedł więc czas, aby zacząć dokładniej przyglądać się oprogramowaniu Odoo i jego strukturze bazy danych, ponieważ byłoby to źródłem

większości naszych danych w części ETL. Jeśli pamiętasz z początku tej części, zainstalowaliśmy oprogramowanie Odoo, używając PostgreSQL jako bazy danych do przechowywania metadanych aplikacji. Tutaj zobaczymy mały przegląd klienta PostgreSQL, dzięki czemu możemy sprawdzić źródłową bazę danych i tabele Odoo. Tutaj przedstawimy tylko przegląd, które tabele rozważymy później w naszym systemie, i wyjaśnimy podstawową relację między nimi a modułem, do którego należą.

Sprawdzanie naszego modelu

Możemy zacząć od bezpośredniego połączenia z bazą danych i rozpoczęcia przeglądania tabel. Jednak w zależności od aplikacji, które zainstalowaliśmy, znajdziemy wiele i wiele tabel. Na szczęście istnieje bardzo dobry internetowy zasób modułów Odoo na następującej stronie:

http://useopenerp.com/v8

Póki co dostępna dokumentacja dotyczy wersji 8.0, my będziemy używać wersji 9.0, ponieważ w naszym przypadku nie jest to bardzo ważne, ponieważ różnice są minimalne. Jak widać w sieci, aplikacja Odoo jest podzielona na zestaw kategorii, czyli modułów, które są podobne do aplikacji w Odoo. Chociaż nie tłumaczą się bezpośrednio, ponieważ istnieją pewne wspólne kategorie, które są używane przez wiele aplikacji i są uważane za rdzeń Odoo, niektóre z nich mają podobieństwa. Na przykład w Odoo możemy znaleźć kategorię Zarządzanie sprzedażą oraz aplikację Zarządzanie sprzedażą, ale na przykład nie możemy znaleźć kategorii Fakturowanie, podczas gdy w Odoo mamy aplikację do fakturowania. Wynika to z faktu, że Fakturowanie należy do kategorii Zarządzanie sprzedażą. Jeśli klikniemy dalej w link Zarządzanie sprzedażą, możemy zaobserwować następujący adres URL modułu:

http://useopenerp.com/v8/module/sales-management

Możemy również zobaczyć szczegóły dotyczące modułu Zarządzanie sprzedażą w Odoo. Możemy zobaczyć krótki opis aplikacji i przepływów obsługiwanych przez ten moduł wraz z kilkoma zrzutami ekranowymi menu i opcji objętych tym modułem, wszystkimi klasami Pythona zaangażowanymi w diagram klas UML, na wypadek, gdybyśmy chcieli dostosować ten moduł przez modyfikowanie kodu bezpośrednio w aplikacji i jeszcze kilka rzeczy, szczególnie interesujących dla naszego projektu, sekcja Modele. Model to jedna funkcjonalność zaimplementowana w Odoo. Mamy na przykład model sale.order, który zawiera wszystkie informacje dotyczące funkcjonalności zamówień aplikacji. Na przykład dla tego modułu możemy zobaczyć wszystkie kolumny, które są odwzorowane na różne tabele bazy danych i będą zawierać wymaganą logikę do zaimplementowania tej funkcjonalności. Należy również zauważyć, że model może być używany przez wiele aplikacji. Diagram Modelu jest również ważny, ponieważ dostarcza nam informacji o relacjach między innymi modelami i klasami, ale w tym momencie bardziej interesują nas kolumny i typy tych kolumn; więc patrząc na tę sekcję, możemy zacząć opracowywać listę pól, które będziemy mogli wyodrębnić i skopiować do naszego obszaru przejściowego. Na przykład, w oparciu o potrzeby biznesowe, jesteśmy zainteresowani wydobywaniem informacji o naszych zamówieniach. Aby wesprzeć naszą analizę, nie jesteśmy zainteresowani wyodrębnieniem całego modelu zamówień, ponieważ prawdopodobnie znajdą się tam informacje, których nie potrzebujemy. Pierwszym krokiem będzie wyodrębnienie informacji zbiorczych dla tabel modeli Odoo do naszego obszaru przejściowego. Na przykład możemy rozważyć skopiowanie następujących pól z modelu zamówienia:

Nazwa kolumny: Opis

nazwa : Nazwa zamówienia

stan L Status zamówienia: wersja robocza, wysłane, anulowane, w trakcie realizacji, wykonane…

Date_order : Data realizacji zamówienia

User_id : Klucz do Sprzedawcy, który przyjął zamówienie w modelu res.users

Partner_id L Klucz do identyfikacji naszego klienta w modelu res.partner

Order_line : jeden lub więcej wierszy zamówień produktów, które odwołują się do modelu sale.order.line

Otrzymując te pola będziemy już mogli obliczyć w naszej hurtowni danych np. sumę zamówień klienta, sumę zamówień generowanych przez handlowca, liczbę sztuk na zamówienie jako średnią, sumę zamówień miesięcznie oraz na przykład procent zamówień, które faktycznie kończą się zaakceptowaną fakturą, żeby wymienić tylko kilka możliwych analiz. Prawdopodobnie, aby uzyskać bardziej znaczące informacje, musimy wyodrębnić więcej informacji z innych modeli. Na przykład prawdopodobne jest, że zamiast ich wewnętrznych identyfikatorów będziemy potrzebować nazwisk naszych pracowników lub klientów, prawda? Będzie to wymagało od nas nieco więcej ćwiczeń i wyszukania potrzebnych kolumn w res.users (dla naszych pracowników) i res.partners (dla naszych klientów), aby wyodrębnić wszystkie ich nazwiska, adresy i wszelkie istotne informacje potrzebujemy dla nich.

Uwaga

Próba zrozumienia modelu danych, zwłaszcza tak dużego jak ten, bez posiadania zbyt dużej wiedzy na temat UML, relacji między jednostkami i baz danych, może początkowo być nieco onieśmielająca. Ale przykłady, które zrobimy , są łatwe do zrozumienia i wszystkie będą prowadzone, więc nie denerwuj się, jeśli jesteś teraz trochę przytłoczony.

Spoglądając na res_users szybko zauważymy, że nie ma tam nazwisk pracowników. Stanowi to problem dla naszej próbki, ponieważ ich potrzebujemy. Musimy więc szukać gdzie indziej. Na szczęście znowu model przychodzi nam z pomocą i udaje nam się ich zlokalizować. Podążając za informacjami ze strony modelu widzimy następującą sekcję, po definicji kolumny wszystkich dostępnych kolumn dla użytkowników res_users: „Kolumny odziedziczone z bazy dodatków > res > res_partner.py > class res_partner”.

W tej chwili musimy skupić się tylko na ostatnim kroku zdania. Daje nam to wskazówkę, gdzie się udać i znaleźć potrzebne dane. W tym przypadku system mówi nam dokładnie, że musimy wyszukać dane w tabeli res_partner. A patrząc na model widzimy, że tabela res_partner zawiera kolumnę name, która będzie identyfikować naszego pracownika. Ale jednocześnie ta tabela jest również używana dla klientów, których możesz się zastanawiać, jak wyjaśniliśmy wcześniej. Masz rację. Dla Odoo wszyscy użytkownicy, którzy tworzą część naszego systemu, a nawet organizacje, są uważani za członków modelu res_partner. Oznacza to, że wszystkie firmy, w tym nasza własna, wszyscy nasi pracownicy, wszyscy nasi klienci, bez względu na to, czy są to firmy, czy osoby fizyczne, będą miały co najmniej jeden wpis w tabeli res_partner. Fakt ten oznacza, że kiedy będziemy pracować nad ekstrakcją, będziemy musieli rozważyć wykonanie wielu połączeń do tej tabeli, oczywiście przy użyciu różnych kluczy i kolumn, w zależności od tego, jakie informacje chcemy wyodrębnić. Ale tym zajmiemy się dopiero później.

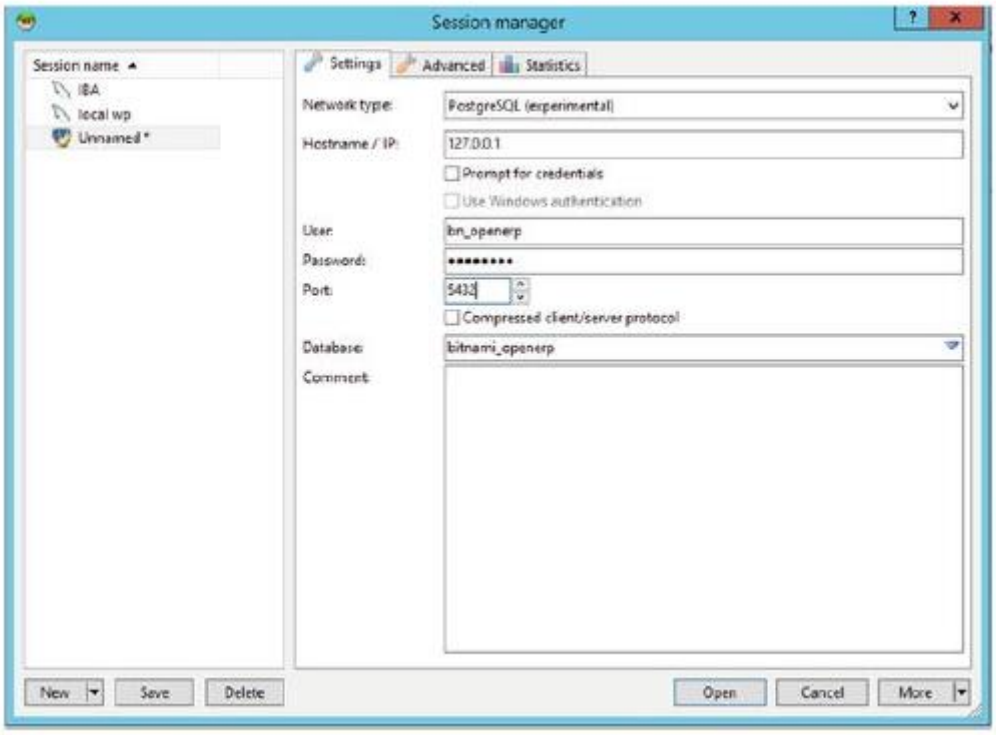
Konfigurowanie połączenia PostgreSQL z serwerem

Jeśli pamiętasz z pierwszej części tego rozdziału, kiedy instalowaliśmy pakiet Bitnami, zdecydowaliśmy się na użycie dołączonej bazy danych PostgreSQL. Oznacza to, że metadane wszystkich aplikacji i modułów Odoo będą przechowywane w bazie danych PostgreSQL. To dobre ćwiczenie, ponieważ w

dalszych rozdziałach, kiedy będziemy pracować nad częścią ETL, zobaczymy, jak połączyć się z różnymi bazami danych, ponieważ nasza hurtownia danych będzie w MariaDB. Jednak w tym rozdziale bardziej interesuje nas możliwość łączenia się z metadanymi aplikacji i przeglądania naszej bazy danych. Zaczynajmy. Istnieje wiele zastosowań do tego celu iw różnych smakach. Możemy zdecydować się na ponowne użycie wiersza poleceń lub skorzystać z aplikacji GUI. Chociaż linia poleceń jest czasami przydatna, jeśli zasadniczo interesuje nas przeglądanie relacji i zawartości danych, dostrojenie klienta konsolowego do ładnego wyświetlania jest bardzo trudne. Z tego powodu tym razem użyjemy narzędzia GUI do sprawdzenia naszych metadanych Odoo. Istnieje wiele narzędzi GUI, które są bezpłatne i będą pasować do rachunku. W rzeczywistości PostgreSQL jest dostarczany z pakietem. My jednak wolimy używać HeidiSQL. Dobrą rzeczą jest to, że chociaż HeidiSQL jest przeznaczony dla baz danych MySQL i MariaDB, obsługuje również bazy danych PostgreSQL. I jest open source, więc można go używać za darmo!

Konfigurowanie połączenia PostgreSQL z serwerem

Jeśli pamiętasz z pierwszej części tego rozdziału, kiedy instalowaliśmy pakiet Bitnami, zdecydowaliśmy się na użycie dołączonej bazy danych PostgreSQL. Oznacza to, że metadane wszystkich aplikacji i modułów Odoo będą przechowywane w bazie danych PostgreSQL. To dobre ćwiczenie, ponieważ w dalszych rozdziałach, kiedy będziemy pracować nad częścią ETL, zobaczymy, jak połączyć się z różnymi bazami danych, ponieważ nasza hurtownia danych będzie w MariaDB. Jednak w tym rozdziale bardziej interesuje nas możliwość łączenia się z metadanymi aplikacji i przeglądania naszej bazy danych. Zaczynajmy. Istnieje wiele zastosowań do tego celu iw różnych smakach. Możemy zdecydować się na ponowne użycie wiersza poleceń lub skorzystać z aplikacji GUI. Chociaż linia poleceń jest czasami przydatna, jeśli zasadniczo interesuje nas przeglądanie relacji i zawartości danych, dostrojenie klienta konsolowego do ładnego wyświetlania jest bardzo trudne. Z tego powodu tym razem użyjemy narzędzia GUI do sprawdzenia naszych metadanych Odoo. Istnieje wiele narzędzi GUI, które są bezpłatne i będą pasować do rachunku. W rzeczywistości PostgreSQL jest dostarczany z pakietem. My jednak wolimy używać HeidiSQL. Dobrą rzeczą jest to, że chociaż HeidiSQL jest przeznaczony dla baz danych MySQL i MariaDB, obsługuje również bazy danych PostgreSQL. I jest open source, więc można go używać za darmo! Aby zainstalować HeidiSQL, możemy kilka razy kliknąć Dalej. Jeśli chcesz zmienić ścieżkę lub inną konfigurację, możesz to zrobić. Jeśli podczas instalacji postępowałeś zgodnie z domyślnymi krokami, prawdopodobnie parametry połączenia będą takie same, jak te pokazane na rysunku



To jest ekran powitalny, który pojawia się, gdy po raz pierwszy otwierasz HeidiSQL i jest to miejsce do zdefiniowania wszelkich nowych połączeń, które chcesz utworzyć. Nasze połączenie byłoby z silnikiem PostgreSQL, więc upewnij się, że jest to ten wybrany w rozwijanym polu Typ sieci.

Po połączeniu zobaczymy kilka baz danych, z których większość jest wewnętrzna w bazie danych, ale wyróżnia się jedna o nazwie publiczna. Musimy kliknąć ten, a pojawi się lista tabel. Jednym z nich jest res_partner. Jeśli klikniesz go dwukrotnie, wejdziemy w tryb tabeli. W tym momencie nie chcemy edytować tabeli, więc nie zapisuj przypadkowo wprowadzonych zmian (jeśli w ogóle!). W tym widoku jesteśmy w stanie zobaczyć kolumny, relacje z innymi tabelami, a nawet przeglądać dane, a wszystko to za pomocą kilku kliknięć myszką. Karta danych będzie później bardzo przydatna, aby sprawdzić, czy dane, które wyodrębniamy, pasują do danych w systemie źródłowym. Lista kolumn daje nam wskazówki dotyczące kolumn, które możemy potrzebować wyeksportować, a żółte i zielone klucze przed numerami kolumn pokazują, czy kolumna jest kluczem podstawowym (żółty), czy kluczem obcym (zielony) odwołującym się do kolumny w innej tabeli . Informacje te uzupełniają informacje, które możemy znaleźć na stronie modelu i pomogą nam przez cały czas dokładnie wiedzieć, jakie informacje potrzebujemy pobrać i skąd je uzyskać.

Wniosek

Zobaczyliśmy, jak zainstalować system operacyjny, w naszym przypadku pakiet Odoo. Widzimy również, jak wykonać podstawową konfigurację aplikacji. Następnie zainstalowaliśmy bazę danych MariaDB do przechowywania naszego magazynu danych i utworzyliśmy dwie wymagane bazy danych: bazę danych ODS, którą nazwaliśmy staging; oraz baza danych hurtowni danych. Po instalacji naszej hurtowni danych przyjrzeliśmy się strukturze Odoo, gdzie przechowuje informacje i jak uzyskać do nich dostęp z narzędzia GUI. Jeśli wykonaliśmy wszystkie kroki opisane w tym rozdziale, jesteśmy gotowi, aby przejść do bardziej zaawansowanych koncepcji i wreszcie zacząć bawić się naszymi danymi i je analizować. Z tego powodu ważne jest, aby wykonać wszystkie kroki opisane tutaj, ponieważ kolejne części zakładają, że instalacje i konfiguracje omówione w tej części zostały już zaimplementowane.

5. Modelowanie danych dla rozwiązań BI

Otrzymałeś od swojego szefa prośbę o wdrożenie hurtowni danych na serwerze bazy danych, który niedawno zainstalowałeś. Jesteś wybraną osobą do kierowania i być może rozwoju, ale zawsze będzie to zależeć od wielkości i zasobów Twojej firmy, rozwiązania, które musi umożliwiać Twojej firmie analizę danych. Tak więc, po przeprowadzeniu wcześniejszej analizy w oparciu o wymagania użytkownika, mając bazę danych dostępną do działania i wystarczającą ilość informacji do zasilenia systemu, należy rozpocząć logiczny i fizyczny rozwój rozwiązania bazodanowego, do którego dostęp będzie można uzyskać z narzędzia BI. Jeśli postępowałeś zgodnie z instrukcjami z poprzedniej części, będziesz mieć ERP do wykorzystania jako źródło informacji; zainstalowana wybrana baza danych dostępna do alokacji danych firmy; a tutaj zaproponujemy Ci zestaw technik i rozwiązań, które pozwolą Ci przeanalizować przykładowe dane, które trafiają do systemu Odoo w celach testowych. Model, który zamierzamy zaproponować to podstawowa analiza w układzie płatka śniegu, która pozwoli ci wypełnić podstawową tabelę faktów, niektóre tabele wymiarów i pochodne wyszukiwania, aby móc drążyć informacje od analizy najwyższego poziomu do najwyższy dostępny poziom szczegółowości, coś podobnego do modelu płatka śniegu z Części 1. Kiedy już będziemy mieć ten prosty model początkowy, przeanalizujemy niektóre konfiguracje i struktury, które pozwolą zaoferować klientom większą funkcjonalność (jak zwykle mogą być klientami wewnętrznymi lub zewnętrznymi). Wszystkie te punkty zobaczymy dzięki połączeniu wyjaśnienia proponowanego rozwiązania z głębszą analizą modelowania danych, a nie teorii, jak wyjaśniono w części wprowadzającej.

Konwencja nazewnictwa i nomenklatura

Przed przystąpieniem do projektowania bazy danych proponujemy zdefiniowanie podstawowej konwencji nazewnictwa, aby ułatwić zarządzanie bazą danych. Kiedy zaczynasz od opracowania modelu, możesz pomyśleć, że najbardziej intuicyjnym sposobem nadawania nazw tabelom jest po prostu nadanie nazwy zawartym w nich informacjom. I to prawda, ale kiedy twój model zaczyna dorastać, możesz napotkać pewne problemy. Wyobraź sobie, że mamy firmę, która sprzedaje narzędzia hurtownikom, a także mamy kanał sprzedaży detalicznej do bezpośredniej sprzedaży ludziom. Analizując nasze środowisko sprzedażowe pochodzące z rozwiązania ERP, którego używamy do zarządzania sprzedażą do hurtowników, możemy mieć tabelę zawierającą naszych różnych klientów, których mamy w tym module sprzedażowym ERP i możemy po prostu pomyśleć o nazwaniu jej KLIENCI. Opracowujemy całą analizę modułu obejmującą wiele wymiarów, tabele faktów, procesy związane z ETL, narzędzie BI nad tym dostępem do bazy danych a gdy mamy zaimplementowany cały moduł sprzedażowy chcemy analizować dane pochodzące z naszego CRM gdzie mamy informacje dla klientów detalicznych. W tym momencie będziemy mieli inną koncepcję klientów. To, co uważamy za klientów w sprzedaży, to hurtownicy dla CRM, a klienci w CRM nie istnieli w sprzedaży. Natura klientów nie jest taka sama, ani źródło ani pola, które masz dostępne, więc nie ma sensu mieszać wszystkich klientów w jednym stole. Możesz nazwać tabelę CRM_CUSTOMERS lub RETAIL_CUSTOMERS lub jakimkolwiek innym wynalazkiem, ale wtedy zaczniesz komplikować nazwy. Z drugiej strony możesz pomyśleć, że lepiej mieć różne stoły dla klientów i stosować konwencję nazewnictwa, ale, och! Musisz zmodyfikować procesy ETL, odniesienia do baz danych, interfejs narzędzia BI i wszelkie inne zależności od tabeli KLIENCI. Więc uwierz nam, zainwestuj trochę czasu przed rozpoczęciem prac nad zdefiniowaniem konwencji nazewnictwa, która przyniesie korzyści w przyszłości. Podczas naszego doświadczenia widzieliśmy różne konwencje nazewnictwa u różnych klientów i zalecamy stosowanie podejścia pośredniego, zgodnie ze strategią, która pozwala identyfikować tabele i środowiska, ale dając możliwość nadawania znaczących nazw obiektom bazy danych. Jeśli zmusisz się do przestrzegania bardzo ścisłej nomenklatury, będziesz potrzebować słownika, aby przetłumaczyć, jakie jest użycie każdej tabeli i odpytywać go za każdym razem, gdy chcesz napisać nowe zapytanie SQL.

Poniżej mamy zamiar zaproponować konwencję nazewnictwa opartą na tych rozważaniach, ale oczywiście możesz dostosować ją do swoich referencji lub potrzeb. W naszej konwencji nazewnictwa proponujemy używanie nazwy z różnymi częściami oddzielonymi znakiem „_”. Nazwy obiektów bazy danych byłyby zgodne z następującą nomenklaturą:

OT_ST_ENV_DESCRIPTIVE_NAME_SUFFIX

Gdzie każda część pasuje do następującego wyjaśnienia. Możesz zobaczyć przykład na końcu samego wyjaśnienia:

Pierwsza część, Object Type (OT), będzie miała maksymalnie 2 litery, które powinny wskazać, jaki to typ obiektu:

T: Tabele

V: Podgąd

PR: Procedury

PK: Pakiety

MV: Podglądy zmaterializowane

F: Funkcje

TR: Wyzwalacze

Następna część, Subtype (ST), o rozmiarze jednej litery, jest opcjonalną częścią specjalnie używaną w tabelach i widokach, która wskazuje, jaki jest typ tabeli lub widoku. W tym ćwiczeniu rozważymy trzy główne typy, powiązane z typami tabel zdefiniowanymi w części 1:

F: Tabela faktów

L: Tabela wyszukiwania

R: Tabela relacji

Część ENV jest związana ze środowiskiem, do którego należy tabela. Ma od 2 do 4 znaków i jest zmienny w zależności od potrzeb. Jako przykład proponujemy użycie:

MD: Dane podstawowe dotyczące klientów, produktów, zakładów, sklepów, czasu lub wszelkie inne dane, które mają charakter wielofunkcyjny.

SAL: Dane dotyczące sprzedaży związane z Twoim systemem fakturowania.

FIN: Dane finansowe związane z systemem księgowym.

OP: Dane operacyjne związane z Twoim procesem produkcyjnym.

HR: podzbiór informacji o zasobach ludzkich.

STG: Tabele pomostowe pochodzące z ERP lub innych źródeł.

Część DESCRIPTIVE_NAME: jest tak opisowa, że potrzeba znacznie więcej wyjaśnień, powinno ono zawierać nazwę pozwalającą na łatwe rozpoznanie, które informacje zawarte są w tabeli ze słowami (lub akronimami) oddzielonymi znakiem „_”. Część SUFFIX jest również częścią opcjonalną, którą szczególnie polecamy używać do oznaczania tabel tymczasowych. Jeśli musisz załadować określoną tabelę, potrzebujesz trzech kroków i zalecamy, aby wszystkie te kroki miały taką samą nazwę jak tabela

końcowa, ale dodając na końcu TMP1, TMP2 i TMP3 jako sufiksy. Więc jeśli mamy tabelę (T), która jest wyszukiwaniem (L) miesiąca rozliczeniowego ze środowiska sprzedaży (SAL), nazwalibyśmy tę tabelę: T_L_SAL_BILLING_MONTH, w tym przypadku bez sufiksu.

Uwaga: podczas nazywania obiektów w bazie danych należy sprawdzić, jaki jest maksymalny rozmiar nazwy dozwolony w bazie danych. Na przykład w Oracle maksymalny rozmiar nazwy tabeli to 30 znaków.

Etapy modelowania

W części 1 widzieliśmy przegląd modelowania danych, biorąc pod uwagę głównie dwa etapy procesu, zdefiniowanie modelu logicznego poprzez ustawienie zestawu jednostek, typów relacji między nimi i atrybutów, które istnieją w dowolnej tabeli, a następnie przejście do modelu fizycznego kiedy już zdefiniowaliśmy wszystkie nazwy pól, typy, precyzję i inne czynniki fizyczne dotyczące tworzenia i lokalizacji tabeli. Teraz dodamy dwa kroki do procesu definiowania naszego modelu, modelowania biznesowego i modelowania wymiarowego. Cały proces modelowania można zobaczyć na rysunku

Wracając do części 2, tę analizę można uznać za część początkowego sprintu zerowego, który sugerowaliśmy na początku projektu. Opiszmy każdy krok bardziej szczegółowo. Aby pomóc Ci w fazie modelowania, zdecydowanie zalecamy skorzystanie z niektórych narzędzi do modelowania dostępnych do tego celu; istnieje również wolne oprogramowanie lub bezpłatne edycje oprogramowania komercyjnego, które mogą pomóc w zdefiniowaniu modeli logicznych i fizycznych. W dalszej części tego rozdziału ocenimy niektóre z tych bezpłatnych produktów do modelowania oprogramowania, aby pokazać, jak używać ich w pomocny sposób.

Model biznesowy

Modelowanie biznesowe jest bezpośrednio związane z ludźmi biznesu, kluczowymi użytkownikami, właścicielami produktów lub kimkolwiek, kto jest rozmówcą od strony biznesowej (może ty, drogi czytelniku). Jest to opis tego, co musi zapewnić Twój projekt BI. W odniesieniu do koncepcji Części 2, byłoby to zmapowane na historie użytkowników i historie deweloperów, więc jest to definicja tego, co chcesz analizować, jakich wymiarów chcesz użyć do analizy i jaką szczegółowość chcesz zastosować do jej analizy . Może to być tekst opisowy, definicja graficzna lub kombinacja obu w języku zrozumiałym dla użytkowników biznesowych. Przydatnym narzędziem, które może być pomocne w tej analizie, jest macierz granulacji, w której można sprawdzić, które wymiary są używane w każdej analizie i na jakim poziomie; możesz zobaczyć przykład na rysunku

	Sales	Finance	Customer Service	Stocks
Customer	Ship to code	Level 3	Ship to code	
Product	Single Product	Family	Single Product	Single Product
Time	Day	Month	Day	Day
Plant			Plant	Plant
Employee	Employee		Employee	
Currency	Currency	Currency		

Aby uzyskać te informacje, pierwszym krokiem będzie zawsze rozmowa z zespołami biznesowymi w celu uzyskania ich wymagań, a następnie przejrzenie dostępnych informacji, aby sprawdzić, czy możesz spełnić określone wymagania. W tej macierzy granulacji widać, że w naszym modelu będziemy mieli sześć wymiarów: Klient, Produkt, Czas, Zakład, Pracownik i Waluta; i będziemy mieli cztery obszary analizy: sprzedaż, finanse, obsługa klienta i zapasy. Wewnątrz macierzy lokalizujemy dostępne wymiary, które będą poziomem wymiaru, który będziemy mieli dla każdego obszaru analizy.

Model logiczny

Po zebraniu wszystkich wymagań i zestawieniu ich w naszej analizie biznesowej będziemy mogli określić, które podmioty muszą zostać uwzględnione w naszym systemie. W tym celu będziemy musieli szczegółowo przeanalizować koncepcje, które nasi kluczowi użytkownicy oczekują do analizy, po prostu sprawdzając, czy mamy możliwość uzyskania ich z naszego systemu źródłowego. Wydaje się całkiem logiczne, że jeśli mają jakieś wymagania dotyczące analizy, to dlatego, że używają ich w źródłowym ERP, ale nie powinieneś akceptować tego jako prawdy, dopóki tego nie zweryfikujesz. Czasami nie będziesz mieć możliwości bezpośredniego dostępu do źródłowego ERP, ale do pośredniego środowiska pomostowego, w którym masz tylko wyodrębnione niektóre tabele i niektóre pola i nie masz dostępu do tych wymaganych do zaimplementowania powiązanej funkcjonalności. Czasami użytkownik chce przeanalizować informacje na poziomie zagregowanym, które konsoliduje w pliku Excel, który należy uwzględnić w systemach źródłowych. W naszym przykładzie zdefiniujemy różne encje powiązane z każdym wymiarem, jak widać na rysunku .

Dimension	Entities
Sales	Sales
	Status
Time	Time
	Month
	Quarter
	Year

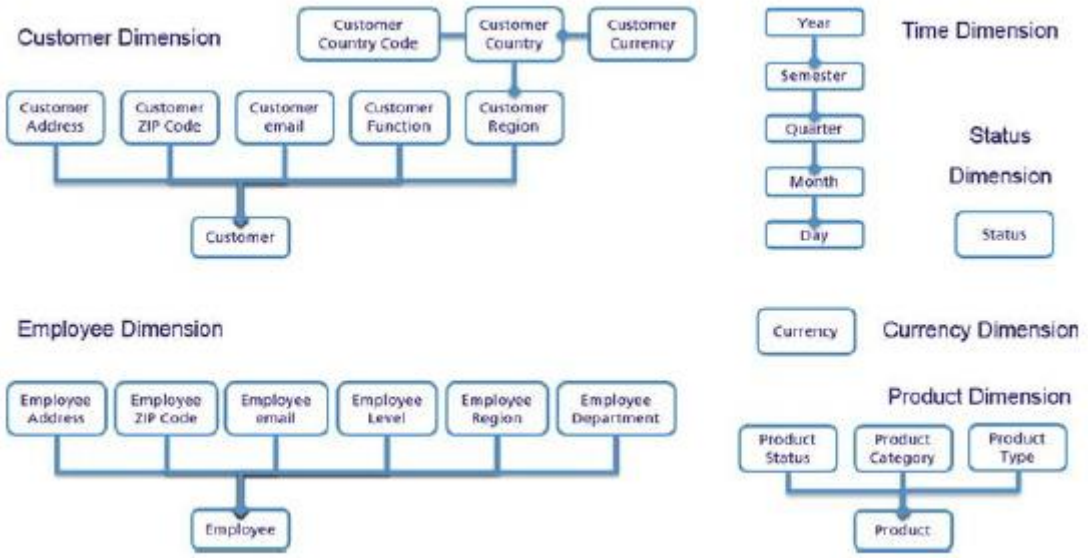
Dimension	Entities
Customer	Customer
	Customer Country
Product	Product
	Category
Employee	Employee
	Employee Country
Currency	Currency

Tutaj możemy zobaczyć podmiot sprzedaży, który odnosi się do informacji bezpośrednio związanych z procesem sprzedaży i różnymi wymiarami zdefiniowanymi w poprzednim kroku z powiązanymi z nimi podmiotami. W każdym z nich podmiot główny będzie zawierał powiązanie z centralnym podmiotem sprzedażowym, a pozostałe będą lubiane do tego podmiotu głównego każdego wymiaru. Na przykład encja Czas będzie powiązana z encją Sprzedaż, a encje Miesiąc, Kwartał i Rok będą powiązane z encją Czas. Również w definicji modelu logicznego zdefiniujemy, w jaki sposób te podmioty są powiązane, czy jest to relacja jeden do jednego, jeden do wielu, czy wiele do wielu. We wstępnej propozycji modelu, który definiujemy, wszystkie relacje to jeden do wielu. Analizując niektóre zaawansowane opcje, zauważymy pewien wyjątek dla relacji między sprzedażą a walutą, który będzie wiele do wielu. Pozostała część encji wymiaru głównego będzie powiązana z encją sprzedaży w relacji jeden-tomany, innymi słowy, będziemy mieć wiele wierszy dla klienta w encji sprzedaż, ale tylko jeden wiersz w encji wymiaru klienta. Podobnie jednostki drugorzędne wymiaru będą miały relację jeden-do-wielu z jednostką główną. Idąc za przykładem, będziemy mieć wielu klientów w jednym kraju, więc wiele wierszy dla danego kraju w encji klienta, ale jeden wpis dla kraju w encji Kraj klienta. Wreszcie, w modelu logicznym musimy zdefiniować, które atrybuty są umieszczane w każdej jednostce, innymi słowy, jakie cechy różnych koncepcji biznesowych będą dostępne dla każdej jednostki. Mówiąc o

produktach, możemy pomyśleć o kolorze, rozmiarze, kategorii, zakładzie produkcyjnym lub jakiejkolwiek innej koncepcji związanej z produktem. Mówiąc o klientach, możemy myśleć o regionie, mieście, adresie, kraju, e-mailu itp. Dla wymiaru czasowego możemy mieć dzień, kwartał, tydzień, semestr, rok, dzień tygodnia, miesiąc roku lub inne atrybuty związane z czas.

Model wymiarowy

Podczas procesu modelowania logicznego będziesz musiał określić, które fakty będziemy musieli wziąć pod uwagę i jakie są nasze wymiary, które można uznać za grupy powiązanych atrybutów z określoną relacją między nimi. Czasami jest to uważane za część modelu logicznego, ale czasami, zwłaszcza w przypadku dużej złożoności, można to zrobić w osobnej analizie, uzyskując w rezultacie model wymiarowy dla swoich danych. Podczas definiowania typów relacji między atrybutami zobaczysz, że możesz mieć te same typy relacji, co w przypadku encji, jeden do jednego, jeden do wielu i wiele do wielu, ale w tym przypadku bierzemy pod uwagę oczekiwaną szczegółowość dla pól wewnątrz tabele, a nie szczegółowość tabel. Najwyraźniejszym wynikiem analizy wymiarowej jest schemat hierarchiczny, jak pokazano na rysunku



W tym przypadku różne typy relacji mają podobne znaczenie, ale z różnicą koncepcyjną. Gdy istnieje relacja jeden-do-wielu między dwoma atrybutami, oznacza to, że na przykład w przypadku Region klienta vs. Klient można mieć wielu klientów zlokalizowanych w regionie, ale tylko region przypisany do każdego klienta . W przypadku kodu kraju klienta vs. kraju klienta, zdefiniowana relacja jest jeden do jednego, więc kraj może mieć tylko kod kraju, a kod kraju może należeć tylko do kraju, rzeczywisty związek, o ile odpowiada temu kodowi kraju na międzynarodowy numer kierunkowy. Na koniec mamy wymiary Waluta i Status, które w tej początkowej wersji są zdefiniowane jako pojedynczy wymiar atrybutu. Dodatkowo do tej analizy graficznej powinniśmy otrzymać listę faktów, które chcemy przeanalizować w naszym projekcie, wraz z opisem źródła i warunków. Powinniśmy otrzymać coś podobnego do pokazanego na rysunku

Entity	Fact	Description
Sales	Quotation Quantity	Quantity ordered when order is in Quotation status
	Ordered Quantity	Quantity ordered when order is in Confirmed status
	Delivered Quantity	Quantity delivered to the customer
	Invoiced Quantity	Quantity invoiced to the customer
	Invoiced Amount	Amount invoiced to the customer
	TAX	TAX applied to the invoiced amount
	Total Invoiced	Sum of amount and tax
Product	Product price	Product Price from catalog

Uwaga: Myśląc o kolejnym rozdziale, związanym z implementacją procesu ETL, możemy rozwinąć jak najwięcej informacji podczas tworzenia modelu danych. W takim przypadku możemy podać więcej informacji technicznych, jeśli są dostępne, takich jak nazwa tabeli źródłowej i pole.

Model fizyczny

W końcu dochodzimy do modelu fizycznego. To ostatni krok tej części. Rozmawialiśmy z kluczowymi użytkownikami, aby zrozumieć i udokumentować ich potrzeby, określiliśmy, które podmioty logiczne będą zaangażowane w nasz model i jakie są relacje między nimi, analizowaliśmy, jakie wymiary i fakty będziemy mieć w naszych modelach oraz atrybuty używane w naszych wymiarach; teraz wykonamy ostatni krok modelowania danych przed przystąpieniem do tworzenia obiektów w bazie danych. Szczególnie przydatne w tym kroku jest użycie narzędzia do modelowania danych, ponieważ po zdefiniowaniu struktury tabel, kolumn, typów kolumn, kluczy podstawowych, kluczy obcych itd. w interfejsie graficznym, w większości z nich będziemy mieli możliwość generowanie skryptów tworzenia dla wszystkich tych obiektów do wykonania w bazie danych lub bezpośrednio możliwość tworzenia powiązanych obiektów. Wewnątrz naszego modelu fizycznego zdefiniujemy jako główne obiekty modelu, które będą tabelami służącymi do lokalizowania informacji i będą one bezpośrednio powiązane z jednostkami, a każda jednostka zostanie zmapowana do tabeli fizycznej. W zależności od opcji dostępnych w naszej bazie danych będziemy mieli możliwość zdefiniowania niektórych parametrów związanych z każdą tabelą, takich jak partycjonowanie, lokalizacja, schemat lub właściciel, kompresja i wiele innych funkcji, które będą się różnić w zależności od oprogramowania bazy danych, a także od modelu wybrane oprogramowanie. Kolejnym krokiem będzie zdefiniowanie, które pola będą zawierały nasze tabele, które zdefiniowaliśmy w naszym systemie. Będą one ściśle powiązane z atrybutami modelu logicznego i wymiarowego. Dla pól zdefiniujemy nazwę zgodnie z naszymi konwencjami nazewnictwa; typ pola, zazwyczaj będą to numery; znak lub data, z różnymi podtypami w zależności od bazy danych; zdefiniujemy również rozmiar i precyzję pola oraz inne parametry, takie jak między innymi to, czy mogą być puste, czy nie, które również zależą od bazy danych i narzędzia do modelowania danych. Dzięki tabelom i polom mamy podstawę do zdefiniowania działającego projektu, ale są jeszcze inne elementy, które mogą nam pomóc poprawić wydajność i integralność danych w naszym systemie. Są to indeksy, klucze podstawowe i klucze obce. Indeks to posortowana struktura, która zawiera różne wartości pola oraz wskaźnik, gdzie te dane się znajdują, poprawiając czas odpowiedzi systemu, gdy szukamy danej wartości. Klucz podstawowy to indeks, który również definiuje zbiór pól, których kombinacje wartości identyfikują pojedynczy wiersz w tabeli, nie może istnieć ta sama powtarzalna wartość kombinacji pól klucza podstawowego w więcej niż jednym wierszu. W tabeli przeglądowej atrybutu, w której mamy identyfikator i opis, kluczem podstawowym byłby identyfikator, o ile spodziewamy się mieć tylko jeden wiersz na wartość atrybutu, co zapewni, że nie będziesz mieć zduplikowanych informacji jeśli wybierzesz opcję dołączania do dowolnej tabeli za pomocą tego wyszukiwania. Z drugiej strony w tabeli faktów może się zdarzyć, że nie będziemy w stanie zdefiniować klucza podstawowego, będzie to zależało od naszego projektu, ale zwykle w środowiskach

hurtowni danych będziesz miał zagregowane informacje na pożądanym poziomie, co spowoduje domyślnie, że twój klucz jest ustawiony kolumn, które zostały uwzględnione w klauzuli group by. W naszym systemie możemy również mieć klucze obce, które są fizyczną równoważnością relacji między podmiotami. Korzystanie z kluczy obcych ma swoje zalety i wady, więc jeśli chcesz je zaimplementować, musisz wcześniej wiedzieć, jakie problemy możesz napotkać. Główną zaletą jest to, że zapewniają integralność danych w tabelach. Jeśli w tabeli sprzedaży masz klucz obcy przy identyfikatorze produktu w tabeli wymiarów produktu, upewnisz się, że nie masz danych związanych z nieistniejącym kodem produktu, więc połączenie z tabelą produktów nie spowoduje utraty danych . Główną wadą jest to, że mogą komplikować proces ETL, a także powodować pewne spowolnienie podczas ładowania danych. Kiedy masz włączony klucz obcy, nie możesz obcinać tabeli, do której się odwołuje, ani usuwać rejestrów, które zawierają informacje powiązane z tabelami zależnymi, co wydaje się logiczne, ale jak zobaczymy w następnej części, czasami proces ETL jest łatwiejszy w zarządzaniu przez obcięcie i pełne ponowne załadowanie niektórych tabel odnośników i wymiarów. Zamiast tego będziemy wymagać użycia instrukcji wstawiania/aktualizacji lub instrukcji łączenia, jeśli Twoja baza danych i narzędzie ETL mają taką możliwość. Istnieją również dostawcy baz danych, którzy umożliwiają wyłączenie ograniczeń kluczy podstawowych w celu ponownego załadowania tabel referencyjnych i włączenie ich ponownie po zakończeniu ładowania.

Strategie ładowania danych

Zanim przejdziemy do modelu fizycznego, który proponujemy na podstawie naszego przykładu, skomentujmy coś na temat strategii ładowania danych dla naszych tabel, które powinniśmy mieć w tym momencie jasne, szczególnie istotne, jeśli połączymy to z skomentowanym ograniczeniem, że użycie obcych klucze mogą spowodować Będziemy mieli głównie trzy typy tabel w zależności od sposobu ich załadowania:

Tabele ręczne/stałe: nie będą uwzględniane w żadnym procesie ETL; będą mieli stałe informacje, dopóki nie zaktualizujemy ich ręcznie. Są przydatne w przypadku atrybutów statycznych lub bardzo słabo zmieniających się, takich jak firma, kraj lub w naszym modelu tabele wyszukiwania walut.

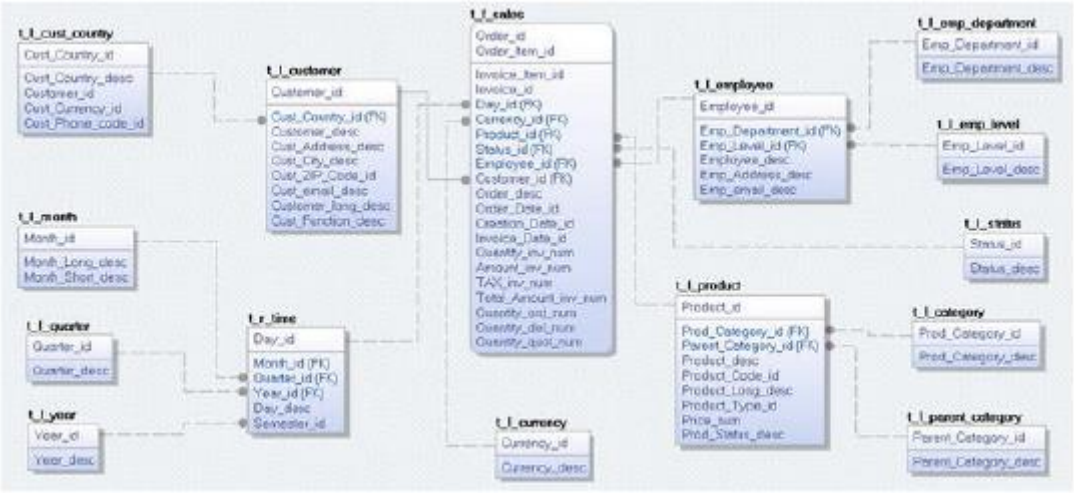
Tabele przyrostowe : W tym przypadku będziemy przesyłać nowe rejestry, które pojawiają się w każdym ładunku. Może mamy dostępne tylko nowe informacje do załadowania, a może mamy całą historię, ale wolimy po prostu ładować nowe wiersze zamiast za każdym razem przeładowywać tabelę od zera. Aby móc mieć wiarygodne informacje w środku, będziemy musieli mieć klucz podstawowy tabeli, abyśmy mogli wykryć, które informacje są nowe, a które są już wstawione do naszej tabeli. W przypadku istniejących rejestrów będziemy mieli dwie możliwości: zaktualizować stare rejestry o nowe informacje dla pól niebędących kluczami lub zachować je bez zmian. Pierwsza opcja jest w tym przypadku najczęstszą opcją.

Tabele pełnego obciążenia : Jest to zwykle najszybszy i najprostszy sposób opracowania ETL, a szczególnie przydatne jest ponowne ładowanie tabel przeglądowych lub tabel wymiarów. Może być również wymagane użycie pełnego procesu przeładowania niektórych zagregowanych tabel faktów, jeśli możemy otrzymać informacje z przeszłości lub jeśli agregacja opiera się na zmieniających się relacjach atrybutów. Jeśli mamy zagregowaną tabelę faktów na poziomie grupy klientów i kategorii, możliwe, że będziemy musieli w pełni ponownie załadować tę tabelę, jeśli klienci mogą przechodzić z jednej grupy klientów do drugiej lub jeśli produkt może zostać ponownie sklasyfikowany w różnych kategoriach. W naszym przykładowym modelu użyjemy kombinacji strategii ładowania tabel, w zależności od wymiaru. Przeanalizujemy je bardziej szczegółowo w następnych sekcjach, ale z reguły będziemy mieć ładowanie przyrostowe dla naszych tabel faktów, ładowanie przyrostowe dla większości tabel wymiarów, ale czyszczenie niektórych danych i pełne ładowanie dla tabel

przeglądowych te klucze, które już istnieją w tabelach wymiarów, aby nie mieć wartości bez powiązanych informacji.

Definiowanie naszego modelu

W naszym początkowym modelu zamierzamy zaimplementować projekt płatka śniegu z niskim poziomem normalizacji. Ale co to oznacza? Jak pokazano w części 1, podejście typu płatek śniegu polega na tym, aby centralne tabele faktów były dostępne za pośrednictwem niektórych tabel relacji, które są połączone z tabelami odnośników lub z mniejszymi tabelami relacji. Rozmiar tabeli maleje, gdy jestem oddzielony od centralnej tabeli faktów, więc zapamiętuje fraktalną strukturę płatka śniegu. Jak skomentowano również w części 1, normalizacja w modelu bazy danych jest stosowana w celu zminimalizowania zajmowanego miejsca w bazie danych poprzez unikanie powtarzających się opisów i pól relacji. Zaproponujemy model, który ma pewną normalizację, ale nie jest nadmierny. Na przykład w przypadku hierarchii danych głównych użyjemy jednej tabeli relacji, aby powiązać pole kluczowe z resztą pojęć, zamiast używać każdego wyszukiwania pojęć, aby powiązać je z jego rodzicem. W przypadku hierarchii czasowej będziemy mieli dzień koncepcyjny, który będzie kluczem relacji między tabelą faktów a relacją pierwszą. Następnie w tabeli relacji będziemy mieli relację między dniem a miesiącem, kwartałem a rokiem. Moglibyśmy mieć związek między Miesiącem a Rokiem w tabeli Miesiąc, ale wolimy używać tabeli relacji, aby zmniejszyć liczbę łączeń wymaganych do rozwiązania zapytań z narzędzia BI. Na rysunku możemy zobaczyć początkowy model zdefiniowany w celu wsparcia podstawowej analizy.

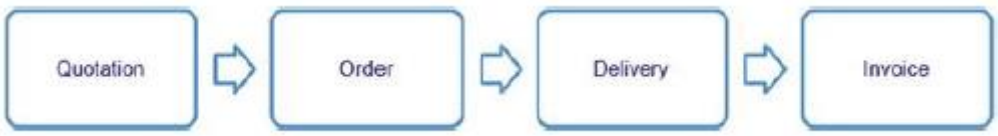


Zwykle będziesz musiał przeanalizować wszystkie wymagania użytkowników, aby zdefiniować model, który jest w stanie rozwiązać wszystkie zapytania użytkowników, ale w tym przypadku, o ile działamy jako nasi użytkownicy, wykonaliśmy proces odwrotny. Przeanalizowaliśmy informacje dostępne w modelu Odoo i na tej podstawie zaproponowaliśmy przykładowy model, który naszym zdaniem może być przydatny, ale nie jest to zwykła procedura, ponieważ zwykle sprawdzasz źródło, aby zebrać informacje, które Twój użytkownik prosi. Zdefiniowanie modelu płatka śniegu będzie miało pewne cechy szczególne, które nasi użytkownicy muszą zaakceptować i musimy to jasno określić. Kiedy agregujemy przez zdefiniowane wymiary, będziemy mieli potencjalny widok danych, to znaczy na podstawie aktualnej sytuacji wymiaru, a nie sytuacji, która była w momencie faktu. W naszym przykładzie produkt jest powiązany z jedną kategorią; nie możesz mieć produktu należącego do więcej niż jednej kategorii. W oparciu o to założenie zmiany kategorii produktów wpłyną na całą historię faktów dotyczących produktu. Na przykładzie sklepów żelaznych wyobraź sobie, że sprzedajesz swoim

klientom truciznę na ślimaki. W początkowej klasyfikacji miała kategorię Produkty Chemiczne, ale Wasza firma zdecydowała się stworzyć nową kategorię dla Ogród i przeniosła truciznę na ślimaki z Chemia do Ogrodu. Cała historia sprzedaży trucizn zostanie przeklasyfikowana, a łączna ilość produktów chemicznych zmniejszy się. Ta cecha musi zostać zaakceptowana i podpisana przez głównego użytkownika i właściciela produktu, aby uniknąć dyskusji po latach. Możesz pomyśleć, że to nielogiczne, Twoja sprzedaż nie powinna się zmieniać, ale jedną z funkcji Twojego systemu BI jest posiadanie informacji statystycznych, ewolucji wskaźników KPI i analizowanie trendów, które pozwalają nam prognozować, a jeśli chcesz przeprowadzić dokładną prognoza, musisz mieć wizję tego, jak byłyby Twoje dane historyczne, gdybyś miał obecną sytuację hierarchii; tutaj mamy koncepcję „wizji potencjalnej". Ale dlaczego chcemy mieć potencjalną wizję? Wyobraź sobie, że jesteś producentem żywności z 20 markami spożywczymi i myślisz o sprzedaży całej marki, aby uzyskać środki finansowe. Aby podjąć właściwą decyzję, przeanalizujesz swoje dane historyczne porównując wyniki różnych marek. Następnie w ciągu ostatnich sześciu miesięcy Twoja firma przenosiła różne produkty, które posiadasz: czekoladę z ciasteczkami z marki MyCookies do marki MyChocolate. Jeśli przeanalizujesz historyczny trend według marki, zobaczysz, że w ciągu ostatnich sześciu miesięcy Twoja sprzedaż spadała, więc możesz zdecydować się na sprzedaż marki MyCookies. Ale jeśli przeanalizujesz to z potencjalnym spojrzeniem, zobaczysz, jak trend historyczny wykorzystuje obecne hierarchie, dzięki czemu możesz mieć porównywalny widok na różne marki i będziesz w stanie podejmować bardziej trafne decyzje. Przeanalizujmy bardziej szczegółowo model danych proponowany do alokacji naszego systemu BI i przyczyny każdej związanej z tym decyzji.

Wymiar sprzedaży

Jako centralną tabelę dla naszego modelu wybraliśmy tabelę Sales, ponieważ jest to najczęstsza analiza, którą należy rozpocząć, gdy zaczynasz wdrażać analizę BI. W tej centralnej tabeli będziemy mieć większość pól faktów, które zostaną zmapowane na metryki i kluczowe wskaźniki wydajności, związane z kwotami, które można podsumować, takie jak ilość (liczba jednostek) i kwota sprzedaży. Informacje będą pochodzić z tabel zamówień i faktur z naszego serwisu transakcyjnego, które ogólnie będziemy traktować jako Dokumenty Sprzedaży. Zdefiniujemy różne statusy każdego dokumentu sprzedaży, w oparciu o status sprzedaży, jak pokazano na rysunku.



Będziemy mieć również różne pola dla ilości w oparciu o ten status dokumentu, który zostanie zdefiniowany na poziomie wiersza, o ile możemy mieć ofertę zawierającą wiersz, który nie jest ostatecznie zamówiony lub ilość zamówienia jest niższa niż cytowany jeden, niektóre pozycje zamówienia mogą nie zostać dostarczone z powodu braku zapasów lub z innego powodu, a być może niektóre z dostarczonych artykułów nie mogą zostać zafakturowane, ponieważ mają jakąś wadę. Ilości mogą się więc różnić dla danej linii dokumentu. Źródłem tej tabeli będą niektóre tabele ERP, które będą zawierały informacje o zamówieniach i fakturach, a także szczegóły pozycji zamówień i faktur; szczegóły dotyczące procesu ETL zostaną przeanalizowane w następnym rozdziale. Będziemy mieć podane wspólne pola od pierwszego etapu dokumentu sprzedaży, stan oferty i inne pola, które zostaną wypełnione tylko wtedy, gdy dokument sprzedaży przejdzie przez obieg. Będzie też jakieś pole, które będzie miało inne źródło w zależności od statusu dokumentu jako identyfikator dnia. Jako wspólne pola będziemy mieli identyfikator klienta, identyfikator pracownika, identyfikator produktu oraz

walutę. Wtedy ze stanu początkowego, notowania, że na końcu jest to zamówienie, które nie zostało potwierdzone, będą dostępne identyfikatory zamówienia i wiersza zamówienia, opis zamówienia i datę utworzenia. Będziemy mogli również poinformować o liczbie produktów związanych z tą wyceną. W tym statusie dokumentu sprzedaży identyfikator dnia będzie informowany o dacie utworzenia, która jako jedyna może być uznana za wiarygodną w tym statusie. Identyfikator statusu, jak możesz sobie wyobrazić, zostanie ustawiony na Cytowany. Po potwierdzeniu zamówienia będziemy mogli podać datę zamówienia i zamawianą ilość. Zaktualizujemy również identyfikator dnia o wartość daty zamówienia oraz zmienimy status na Zamówione. Status dostawy jest opcjonalny na platformie Odoo, więc w tym przykładzie nie będziemy go uwzględniać w tej tabeli; po prostu załadujemy ilość dostawy z ilością zamówienia, gdy otrzymamy zamówienie. W przypadku, gdy w Twojej firmie istnieje obieg pracy, który wymaga zarządzania dokumentem dostawy i uwzględniania rzeczywistej ilości dostawy, możesz po prostu dodać wymagane kolumny do tej analizy. Ostatnim statusem, który weźmiemy pod uwagę w tym procesie, jest Zafakturowany. Gdy zamówienie zostanie zafakturowane, otrzymamy z systemu transakcyjnego identyfikatory faktury i pozycji faktury, datę faktury oraz wymagane pola numeryczne do naszej analizy, zafakturowaną ilość, zafakturowaną kwotę, podatki i łączną kwotę wraz z podatkami. W tym momencie zaktualizujemy status na Zafakturowano oraz identyfikator dnia o wartość daty wystawienia faktury. Być może najdziwniejszym tematem, który widzisz w tym procesie, jest koncepcja połączonego dnia w zależności od statusu. I być może będziesz miał rację, jest to komplikacja analizy, ale uwzględniliśmy ją głównie z dwóch powodów. Po pierwsze, aby pokazać, że czasami możemy mieć pola obliczeniowe, które nie są bezpośrednim wyodrębnieniem z naszego ERP, mamy możliwości przetwarzania w celu wdrożenia pochodnych koncepcji z początkowego źródła. Po drugie, mamy możliwość gry z różnymi datami i chcemy przeprowadzić pełną analizę przy użyciu jednego pola. Zwykle najważniejszą koncepcją daty, którą należy wziąć pod uwagę, jest data faktury, ponieważ to ona wpływa na oficjalne wyniki firmy, ale gdy mamy tylko status Zamówione, nie będziemy informować o dacie faktury, więc jeśli przefiltrujemy daty faktury, stracimy te wiersze w naszej analizie. Możemy użyć podwójnego lub potrójnego filtrowania przy użyciu dwóch lub trzech pól, ale wolimy, aby w tej strategii pole dnia było zawsze informowane o najbardziej znaczącej opcji w każdym statusie. Ale istnieją również pewne przyczyny techniczne poniżej tego. Ponieważ data faktury jest zwykle najważniejszym polem, większość raportów będzie filtrowana według daty faktury. W celu optymalizacji wydajności zapytań jedną z opcji jest możliwość wykorzystania indeksów lub partycjonowania z wykorzystaniem tego pola daty iw obu przypadkach należy o tym poinformować. Z pewnością możemy podać wartość fikcyjną, aby uniknąć wartości zerowych, ale bardziej sensowne jest użycie alternatywnego pola daty, które pokaże nam powiązane informacje i może być użyte do wyodrębnienia całkowitej ilości produktów zafakturowanych w tym miesiącu, w tym również zamówienia, które zostały otrzymane, ale nadal nie są zafakturowane, filtrując tylko po jednym polu daty. Jeśli chodzi o pola ilościowe, będą one zawierać jaka była początkowa kwota wyceny, jaka ilość została ostatecznie zamówiona, jaką ilość dostarczyliśmy, a także jaka została zafakturowana. W idealnym systemie wszystkie powinny być takie same, ale jeśli mamy dostępne wszystkie te ilości, możemy przeanalizować nasz proces sprzedaży, porównując rozbieżności między wszystkimi tymi polami.

Tabela stanu

Różne statusy dokumentu sprzedaży zostaną zapisane w małej ręcznej tabeli, którą wypełnimy ręcznie. Rozpatrujemy ją wewnątrz wymiaru sprzedażowego, o ile jest to atrybut bezpośrednio związany ze sprzedażą. Wewnątrz tej tabeli zapiszemy tylko identyfikator statusu i opis statusu. Będzie używany głównie do drążenia dostępnych statusów w narzędziu do raportowania, aby uniknąć konieczności przechodzenia do tabeli faktów w celu dotarcia do nich.

Wymiar walutowy

Wymiar waluty będzie się składał tylko z bardzo prostej tabeli, tylko z identyfikatorem waluty i opisem waluty, aby w razie potrzeby pokazać ją z długą nazwą. Będziemy przechowywać tę tabelę ręcznie, o ile oczekuje się, że będzie to bardzo wolno zmieniający się wymiar. W tym przypadku wolimy zachować go jako osobny wymiar, mimo to można go rozważyć w ramach wymiaru sprzedażowego, ponieważ w kolejnych sekcjach zaproponujemy pewną ewolucję tego wymiaru, badając inne możliwości modelowania danych.

Wymiar klienta

W przypadku klientów będziemy mieli dwie różne tabele, które pozwolą nam zdefiniować różne koncepcje klienta. Pierwsza tabela relacji pozwoli nam połączyć się z tabelą faktów poprzez identyfikator klienta, a następnie będziemy mieli kilka pól związanych z klientem, które pozwolą nam zapisać niektóre cechy związane z klientem, takie jak opis klienta, adres klienta, miasto klienta, kod pocztowy, identyfikator kraju, adres e-mail, długi opis i funkcja, jaką ten kontakt z klientem pełni w swojej firmie. Dostępna będzie również inna tabela, która będzie wyszukiwać kraje, które będą miały kod kraju, który będzie międzynarodowym kodem dla krajów, nazwę kraju, kod kraju (dwuliterowy kod międzynarodowy), walutę tego kraju i jego telefon przedrostek numeru. Źródłem informacji dla tego wymiaru będzie również system ERP, w którym zidentyfikowaliśmy kilka tabel zawierających informacje dotyczące klienta i kraju. Będziemy musieli zapisać informacje o wszystkich aktywnych klientach, a także o tych historycznych, którzy mogą być nieaktywni, ale mają sprzedaż w naszym systemie, w przeciwnym razie moglibyśmy utracić informacje po dezaktywacji klienta. Musimy zapewnić spójność danych, aby wszystkie możliwe wartości dla klienta były zawarte w tabeli klientów, a także, że nie mamy zduplikowanych identyfikatorów klientów, więc jeśli jest jakieś pole, które zmienia się w tabeli klientów, musimy zmodyfikować istniejącego rejestru, nie możemy dodawać nowych wierszy związanych z istniejącym identyfikatorem klienta. Również w następnych sekcjach porozmawiamy o dodawaniu informacji o geolokalizacji, które to zrobią ,pozwalają nam zlokalizować nasze informacje o sprzedaży na mapie.

Wymiar pracowniczy

Aby zlokalizować dane naszych pracowników, główna tabela wymiarów będzie miała prostszą strukturę niż nasz wymiar klienta; będzie zawierał identyfikator pracownika, opis, adres e-mail i adres domowy, a także dział, do którego należy i poziom, który ma, który może być od stażysty do dyrektora generalnego w przykładowych danych, które mamy w Odoo. W ramach głównej tabeli wymiarów będziemy mieć dwie małe tabele, które będą tabelami przeglądowymi, jedną dla działu, a drugą dla poziomu, o ile w poprzedniej tabeli będziemy mieli tylko identyfikator działu i poziomu. Podczas ładowania tabeli wymiarów będziemy to robić przyrostowo, aktualizując istniejące wiersze i wstawiając nowe, a dla tabel działów i poziomów będziemy obcinać i wstawiać dane tylko na podstawie istniejących wartości tabeli t_l_employee.

Wymiar produktu

Nasz wymiar produktu będzie zawierał wszystkie informacje związane z produktami, które sprzedajemy w naszej branży. W rzeczywistości, jak wyjaśniono w przypadku tabeli klientów, musi ona zawierać wszystkie produkty, które mamy w naszym katalogu, a także te, które są poza katalogiem, ale mają pewną historyczną sprzedaż. Musimy więc upewnić się, że wszystkie produkty, które mamy w naszej tabeli faktów, znajdują się w tabeli relacji między produktami, aby zapewnić integralność danych i że nie ma zduplikowanych wierszy według identyfikatora produktu. W odniesieniu do pól hierarchii produktowej będziemy mieli do dyspozycji identyfikator produktu; że jest to wewnętrzne pole

numeryczne ERP; jako klucz tabeli będziemy mieli krótki i długi opis produktu, kod produktu służący do jego identyfikacji w interfejsie ERP, identyfikator kategorii, który pozwoli nam połączyć się z tabelą wyszukiwania kategorii, rodzaj i status produktu oraz cena produktu. Zdefiniowaliśmy również inną tabelę jako wyszukiwanie kategorii, która będzie zawierała kategorię i jej opis. Chcielibyśmy zauważyć w tym wymiarze, że możemy mieć potencjalny pogląd na niektóre koncepcje, takie jak kategoria, jak wyjaśniono we wstępie, ale także możesz mieć potencjalny pogląd na niektóre fakty, takie jak wielkość sprzedaży przy zastosowaniu bieżącej ceny do danych historycznych.

Wymiar czasu

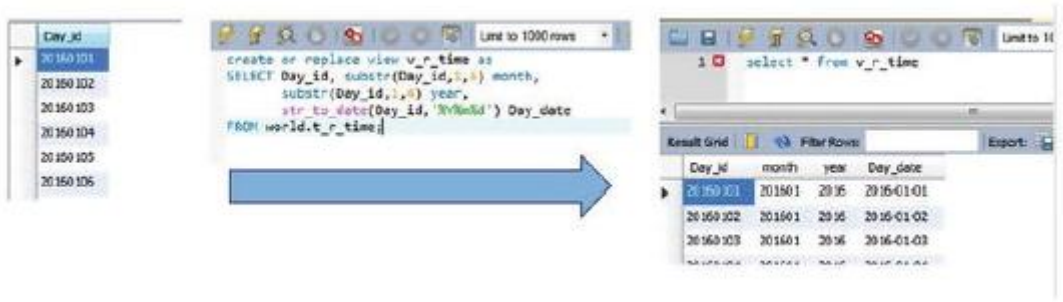
Jednym z podstawowych wymiarów obecnych w prawie wszystkich modelach analizy danych jest wymiar czasu. W zależności od narzędzi BI można zauważyć, że wymiar ten nie jest tworzony fizycznie w bazie danych, ale wyliczany za pomocą własnych funkcjonalności BI lub funkcji bazodanowych na podstawie pola daty znajdującego się w tabeli faktów. W tym przypadku woleliśmy stworzyć go fizycznie w bazie danych, o ile nie naprawiamy żadnego konkretnego narzędzia BI lub bazy danych; po prostu sugerujemy niektóre i być może ta, którą wybierzesz, nie pozwala na wdrożenie tej funkcji, więc na wszelki wypadek przejdziemy do tej fizycznej definicji. W tym przykładzie zdefiniowaliśmy podstawowy wymiar czasu z pięcioma atrybutami: dzień, miesiąc, kwartał, semestr i rok, ale moglibyśmy również użyć innych atrybutów, takich jak tydzień, dzień tygodnia, tydzień roku, miesiąc roku i inni. Do zlokalizowania atrybutów daty użyjemy pól numerycznych, ponieważ ułatwia to obsługę na poziomie bazy danych, ale będziemy mieli dostępny opis daty w formacie daty, aby można go było użyć w narzędziu BI, które zwykle ma możliwości transformacji daty. Oprócz tych komentowanych tematów, w wymiarze czasowym możemy mieć wiele szczegółów, takich jak sposób definiowania tygodnia i związek tygodnia z rokiem i miesiącem; lub Twoja firma może używać równoległego okresu obrachunkowego innego niż naturalny rok i naturalny miesiąc i powinieneś wymagać wdrożenia alternatywnej hierarchii czasu opartej na tej koncepcji okresu obrachunkowego.

Badanie możliwości modelowania danych

Do tego momentu wiesz, jak modelować prostą strukturę płatka śniegu z bezpośrednimi relacjami między tabelami, co pozwoli ci rozwiązywać zapytania agregujące w odniesieniu do niektórych atrybutów. Nic, czego nie można by zdefiniować w pliku programu Excel przy użyciu niektórych formuł do łączenia powiązanych informacji. Teraz nadszedł czas, aby zobaczyć inne funkcje i możliwości, które mogą pomóc w przeprowadzeniu bardziej zaawansowanej analizy w systemie BI, zapewniając potężne opcje i elastyczność w analizie.

Wyświetl warstwę

Opierając się na naszym ostatnio widzianym modelu fizycznym, załóżmy, że chcemy przeanalizować sprzedaż na podstawie pola Data zamówienia, zdefiniowanego jako pole liczbowe. Ale w naszym modelu BI mamy możliwość korzystania z wielu funkcji opartych na polach daty, takich jak operowanie datami, pobieranie pierwszego dnia miesiąca, czy tworzenie dynamicznego filtra, który daje skumulowaną sprzedaż w ciągu ostatnich trzech miesięcy. Moglibyśmy użyć funkcji bazy danych do przekształcenia pola w typ daty, aby uzyskać całą funkcjonalność narzędzia BI lub stworzyć pole, które dostraja rok lub miesiąc. I możemy to zrobić za pomocą widoku bazy danych, bez zapisywania obliczonych danych w bazie danych, po prostu używając jakiejś funkcji w czasie wykonywania. Na rysunku możesz zobaczyć, jak z pojedynczego pola liczbowego w tabeli zawierającej daty w formacie RRRRMMDD możemy utworzyć wiele atrybutów związanych z datami.



Korzystanie z warstwy widoku między tabelą fizyczną a mapą wykonaną w naszym narzędziu BI pozwoli nam na pewną elastyczność w dodawaniu dodatkowej logiki poniżej naszego systemu BI. Możemy wymusić łączenie z tabelą parametrów, aby poprawić wydajność łączenia, możemy użyć go do dostosowania niektórych nazw pól lub precyzji, aby wyrównać je z innymi tabelami w środowisku, możemy zastosować filtr w kodzie, aby usunąć niektóre niepożądane dane ze wszystkich analizy opartej na tej tabeli. Możliwości, które oferują użycie pośredniej warstwy widoku, jest wiele, ale należy z nich korzystać ostrożnie, aby nie wpłynąć na system niepożądanymi efektami. Możesz na przykład użyć widoków do filtrowania zwróconych wartości, stosując dowolny warunek, jaki możesz wymyślić, ale widzieliśmy przykład widoku wyszukiwania dla atrybutu, który zwrócił trzy wiersze z dziewięciu istniejących w tabeli, ponieważ były one używali wartości, ale aby zobaczyć tylko te wartości, które miały powiązane dane w tabeli faktów, łączyli się z całą tabelą faktów zawierającą miliony wierszy. Tak więc, aby pokazać tylko trzy z dziewięciu wierszy, widok przeszukiwał miliony, z logiczną utratą wydajności. Również za pomocą widoków możesz zwiększyć złożoność całego rozwiązania, mając kolejną warstwę, w której możesz dodawać warunki, obliczenia i formuły, możliwe jest, że poczujesz się zagubiony na wypadek, gdybyś musiał przeanalizować jakiś incydent. Opierając się również na prawdziwym przypadku, odziedziczyliśmy odpowiedzialność za zarządzanie działającą platformą BI i musieliśmy przeanalizować niezgodność między systemem źródłowym a raportem BI. Przeanalizowaliśmy cały proces ETL, przeszukując wszystkie etapy, w których moglibyśmy utracić informacje, dopóki nie zobaczyliśmy, że informacje zostały poprawnie załadowane w tabeli końcowej. Jednak informacje nie były zgodne, więc przeanalizowaliśmy bardzo złożony raport, który wykonywał wieloetapowe sprawdzanie poprawności, dzięki czemu nie straciliśmy żadnego rejestru w żadnym sprzężeniu SQL. W końcu zdaliśmy sobie sprawę, że ktoś ustawił niewłaściwy filtr w kodzie widoku, więc zwracane dane różniły się między tabelą a powiązanym widokiem.

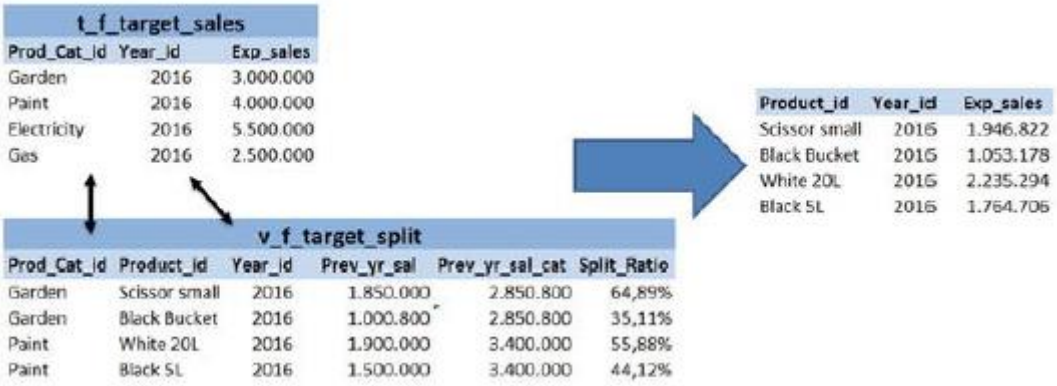
Widoki zmaterializowane

Mówiąc o widokach, większość baz danych ma określone typy widoków zwane widokami zmaterializowanymi. Te typy widoków to obecnie tabela przechowująca informacje, ale oparta na instrukcji select zależnej od innych tabel. Są więc podobne do fizycznej tabeli, w której wstawiamy wynik powiązanego wyboru, aby go wstępnie obliczyć, na koniec dnia można je po prostu uznać za końcowy krok ETL, który należy wykonać, gdy zmieni się zawartość zależnej tabeli. Główną zaletą tego rodzaju obiektów jest to, że mają one powiązane pewne funkcjonalności na poziomie bazy danych, które pozwalają odświeżyć informacje za pomocą prostych instrukcji, w zależności od systemu bazy danych można wykonać tylko częściowe odświeżenie, aby poprawić wydajność odświeżania, i masz elastyczność tworzenia lub modyfikowania ich w oparciu tylko o instrukcję select, taką jak standardowe widoki, dzięki czemu masz pewne zalety obu typów obiektów, widoków i tabel.

Podział danych

Czasami możemy znaleźć się w sytuacji, że nasi użytkownicy potrzebują danych, których nie ma na pożądanym poziomie w systemie źródłowym i jesteśmy zmuszeni „wymyślić” te informacje. To wydaje

się być czymś z Business Intelligence i bardziej związanym z magią, ale możemy zapewnić, że możesz być w takiej sytuacji. Wyjaśnijmy to na kilku przykładach. Kiedy mówimy o wynalezieniu, nie mamy na myśli tylko dodawania losowych danych do tabeli, odnosimy się do stworzenia jakiegoś modelu, który pozwala nam analizować informacje na innym, bardziej szczegółowym poziomie niż w źródle informacji. Wyobraźmy sobie, że mamy zdefiniowany proces budżetowania w naszej firmie, który określa zestaw celów sprzedażowych, które firma powinna osiągnąć, aby uzyskać niezawodny stan na przyszłość. W tego rodzaju procesach budżetowania normalne jest, że nie masz zdefiniowanych informacji na najniższym możliwym poziomie, biorąc za przykład hierarchię produktów, nie będziesz mieć celu zdefiniowanego dla żadnego pojedynczego produktu; będziesz mieć oczekiwaną sprzedaż według kategorii na następny rok. Ale potem, w przyszłym roku, będziesz chciał porównać, jak postępujesz w swoim celu, abyś mógł podzielić ogólny cel dla kategorii na różne produkty w tej kategorii na podstawie sprzedaży z poprzedniego roku, wstępnie obliczając współczynnik podziału, który pozwoli ci zobaczyć dane na pożądanym poziomie. Na rysunku możesz zobaczyć przykład pokazujący wymaganą strukturę do implementacji tego modelu oraz przykładowe dane pokazujące, jak powinna być zdefiniowana.



t_f_target_sales

Prod_Cat_id	Year_id	Exp_sales
Garden	2016	3.000.000
Paint	2016	4.000.000
Electricity	2016	5.500.000
Gas	2016	2.500.000

Product_id	Year_id	Exp_sales
Scissor small	2016	1.946.822
Black Bucket	2016	1.053.178
White 20L	2016	2.235.294
Black 5L	2016	1.764.706

v_f_target_split

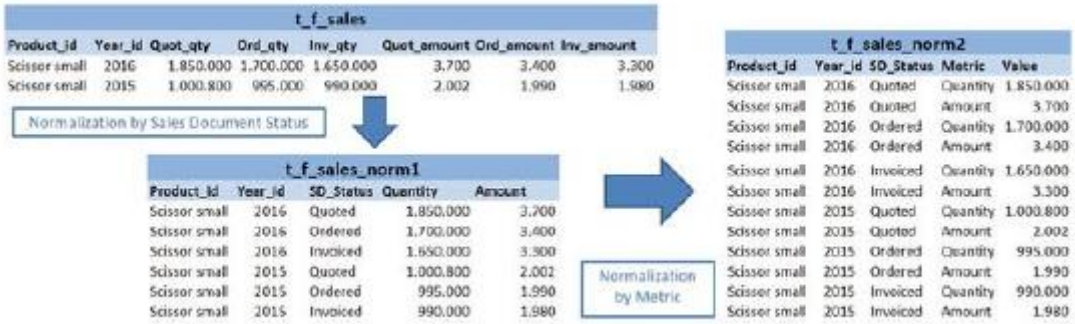
Prod_Cat_id	Product_id	Year_id	Prev_yr_sal	Prev_yr_sal_cat	Split_Ratio
Garden	Scissor small	2016	1.850.000	2.850.800	64,89%
Garden	Black Bucket	2016	1.000.800	2.850.800	35,11%
Paint	White 20L	2016	1.900.000	3.400.000	55,88%
Paint	Black 5L	2016	1.500.000	3.400.000	44,12%

W tym przykładzie pokazujemy strukturę, która pozwala nam podzielić cel sprzedaży na kategorię na cel sprzedaży na produkt, w oparciu o firmę, która ma cztery kategorie i dla uproszczenia analizujemy dwie z nich, przypisując tylko dwa produkty w każdej kategorii. W celu wyliczenia wskaźnika stworzyliśmy widok, który wylicza sprzedaż na poziomie produktu za cały poprzedni rok i przypisuje ją do tego roku. Również ten widok dokonuje agregacji na poziomie kategorii, aby móc obliczyć współczynnik, którego użyjemy do podziału celu. W rezultacie nasze narzędzie BI powinno być w stanie dołączyć do tabeli celów i widoku proporcji oraz uzyskać oczekiwaną sprzedaż na produkt w bieżącym roku.

Normalizacja i denormalizacja faktów

Skomentowaliśmy już, że nasz model podąża za strukturą płatka śniegu z pewnym poziomem normalizacji. W części wprowadzającej widzieliśmy już różnicę między znormalizowaną a zdenormalizowaną strukturą i widzieliśmy już, że normalizacja pomaga nam zachować integralność podczas denormalizacji zapewnia lepszą wydajność zapytań SQL. Ale teraz chcemy porozmawiać o normalizacji i denormalizacji faktów. W przypadku normalizacji zmieniamy strukturę tabeli tak, aby zawierała jak najmniej kolumn liczbowych, w miarę możliwości tylko jedną kolumnę faktów, a następnie definiujemy różne atrybuty, które pozwalają nam rozdzielić kolumny zmieniające informacje na wiersze. Jak zwykle łatwiej pokazać to na przykładzie. W tym przykładzie, który możemy zobaczyć na rysunku , pokazujemy dwa poziomy normalizacji: pierwszy definiujący nowy atrybut o nazwie Status

dokumentu sprzedaży, który może być wyceniony, zamówiony lub zafakturowany oraz przypisanie do kolumn Stan wyceny podana ilość i wyceniona ilość, do kolumn Stan zamówienia ilość zamówiona i kwota zamówienia oraz do kolumn Stan faktury Ilość zafakturowana i kwota faktury. W drugim kroku pokazanym na rysunku utworzyliśmy metrykę konceptualną i przenieśliśmy ją do wierszy, pozostawiając tylko kolumnę z wartością metryki.



Teraz możesz się zastanawiać, dlaczego muszę normalizować? Mogą istnieć różne powody, które można uzasadnić w zależności od wymagań biznesowych. Podczas korzystania z narzędzi Business Intelligence czasami łatwiej jest zarządzać metryką atrybutu niż wieloma metrykami. W następnej sekcji zobaczymy przykład transformacji fazowej, w której warto mieć metrykę atrybutu, aby użyć jej jako selektora. Inną zaletą stosowania znormalizowanych struktur jest większa elastyczność dodawania nowych wartości znormalizowanych atrybutów. Wyobraź sobie, że chcesz dodać nową metrykę z kosztem produktu w tabeli t_f_sales z Figure. Musiałbyś zmodyfikować strukturę tabeli dodając nową kolumnę i wszystkie wymagane obiekty zależne, takie jak widoki, model BI, proces ETL itp. Jeśli chcesz dodać więcej danych do tabeli t_f_sales_norm2 z tego samego przykładu wystarczy, że dodaj więcej wierszy, a żadne modyfikacje w modelu nie będą wymagane. Wystarczy dodać więcej wierszy w procesie ETL, a dane będą dostępne. Interesujące jest również użycie znormalizowanych struktur, jeśli masz słabo wypełnione kolumny faktów i chcesz uniknąć wartości null w wizualizacji. Typowym scenariuszem takiej sytuacji jest bardzo szczegółowa analiza finansowa, w której występuje wiele rodzajów kosztów związanych z fakturami. Zobaczysz, że pojedyncza faktura zawiera pewne standardowe koszty wynikające z wytworzenia twojego produktu, ale wtedy niektóre z nich będą miały promocyjne rabaty, inne będą miały kupony, zachęty sprzedażowe, umowy zjazdowe itp. Tak więc w jednym rzędzie będziesz mieć tylko kilka kolumny faktów poinformowane, ale z punktu widzenia relacji będziesz miał rejestr, który wiąże fakturę z kosztem, mimo że jest pusty, co może spowodować niepoprawną analizę, jeśli spróbujesz przeanalizować liczbę klientów z rabatami zjazdowymi (tak, możesz filtrować według cost <> null, ale musisz o tym pamiętać). Jeśli w tym przykładzie zmienisz strukturę, tworząc atrybut typu kosztu i tylko kolumnę dla wartości, będziesz mieć tylko rejestry związane ze zjazdem dla tych faktur, które mają informacje o zjeździe, a nie dla wszystkich faktur. Możesz także zdenormalizować tabelę, wykonując proces odwrotny. Główne pytanie, które możesz sobie zadać, może brzmieć: dlaczego? Istnieje kilka powodów, dla których można uzasadnić zastosowanie denormalizacji. Pierwszym jest zmniejszenie liczby wierszy tabeli; szczególnie interesujące jest to, że masz dużą gęstość faktów. Jeśli masz te same klucze tabeli dla atrybutu, możesz przenieść atrybuty do kolumn, tworząc kolumnę dla każdej wartości relacji fakt-atrybut. Wyobraź sobie, że masz dostępne cztery kategorie produktów i sprzedajesz wszystkie cztery kategorie wszystkim swoim klientom. Jeśli utworzysz kolumnę dla każdej kategorii, będziesz mieć jedną czwartą wierszy w tabeli końcowej. Inną interesującą cechą zdenormalizowanej strategii jest możliwość porównywania różnych wartości atrybutów. Idąc za przykładem kategorii, możesz mieć kolumnę podsumowującą wszystkie kategorie i procent z wartością każdej kategorii w stosunku do sumy

wszystkich kategorii. Oczywiście, jeśli normalizacja modelu zapewnia większą elastyczność, denormalizacja powoduje sztywność, więc w tym przykładzie, jeśli chcesz dodać nową kategorię, będziesz musiał zmodyfikować model, aby dodać nowe kolumny i powiązane zależności w systemie BI i procesie ETL.

Transformacje czasu

Drążąc możliwości modelowania dochodzimy do czasu przemian. Głównym celem transformacji czasowej jest pokazanie informacji dotyczących danego miesiąca na tle informacji z innych miesięcy. Jak to zwykle bywa w tej sekcji, ponownie pytanie brzmi: dlaczego? Posiadanie atrybutu transformacji umożliwia łatwe porównywanie różnych okresów w jednym wierszu lub kolumnie. Na rysunku możesz zobaczyć tabelę przekształceń, która jest dołączana do tabeli faktów za pomocą kolumny source_mth_id i kolumny month_id tabeli faktów.

t_r_time_transformations

Month_id	Source_mth_id	Transformation_id
201602	201602	Current Month
201602	201601	Previous Month
201602	201512	Previous Month - 2
201602	201502	Previous Year
201602	201602	Year to Date
201602	201601	Year to Date
201602	201502	Previous Year to Date
201602	201501	Previous Year to Date

t_f_sales

Month_id	Category_id	Invoiced_qty
201501	Garden	30
201502	Garden	35
201503	Garden	30
201511	Garden	40
201512	Garden	50
201601	Garden	32
201602	Garden	38
201603	Garden	33

Invoiced Quantity

Month	Category	Transformation: Current Month	Previous Month	Previous Month - 2	Previous Year	Year to Date	Previous Year to Date
201601	Garden	32	50	40	30	32	30
201602	Garden	38	32	50	35	70	65
201603	Garden	33	38	32	30	103	95

Narzędzie do raportowania pobiera atrybut miesiąca z kolumny Month_id tabeli przekształceń i dodaje kolumnę ilość_faktury z tabeli sprzedaży. W tabeli przekształceń pokazujemy tylko dane za luty 2016, ale powinniśmy ją wypełnić dla wszystkich miesięcy w roku. Takie podejście do transformacji czasowej jest uruchamiane przez narzędzie do raportowania, dzięki czemu nie trzeba mieć wstępnie obliczonych obliczeń w bazie danych, czyli sprzedaży z poprzedniego miesiąca lub sprzedaży od początku roku dla danego klienta, a jest to coś, co obliczane po uruchomieniu raportu, który wykonuje zapytanie SQL. Zapewniasz w ten sposób, że rachunek różniczkowy zostanie odświeżony w tym czasie, ale możliwe są problemy z wydajnością podczas mnożenia dostępnych wierszy, na przykład w przypadku daty początkowej dla grudnia uzyskasz dostęp do 12 rzędów, aby uzyskać jeden. Możesz także myśleć w modelu, który pozwala na przeprowadzanie transformacji czasowych w bardziej efektywny sposób poprzez wstępne obliczanie informacji w tabeli faktów, dzięki czemu możesz dodawać kolumny do swojej tabeli faktów i dodawać tam obliczenia dla poprzedniego miesiąca, poprzedniego roku, itd. W tym celu nie można korzystać z bardzo szczegółowej tabeli faktów bazowych, patrząc na nasz model bazowy, jeśli w tabeli mamy identyfikator faktury, nie ma sensu obliczać kwoty faktury z poprzedniego roku, ponieważ kod faktury jest unikalny i nie zawiera informacji za poprzednie okresy. Tego rodzaju analizy można wstępnie obliczać tylko na poziomach zagregowanych.

Fazy faktów

Kiedy mówimy o fazach faktów, mówimy o różnych poglądach na ten sam fakt. Tak więc dla danego faktu jako ilość zafakturowana masz wartość bieżącą, prognozę roczną z początku roku, przegląd

prognozy śródrocznej, cel określony przez odpowiedzialnych za sprzedaż lub porównania między nimi, zarówno w postaci różnicy, jak i wskaźników. W tej sekcji można również uwzględnić przekształcenia czasowe, o ile można je uznać za fazy faktu. Aby je modelować, mamy różne alternatywy:

* Nowe tabele w modelu: Będziesz musiał użyć tej strategii, jeśli różne fazy faktu są wprowadzane na innym poziomie, na przykład w przypadku faktów prognozowanych, gdzie zwykle nie mamy ich na najwyższym poziomie szczegółowości . Również ta nowa tabela mogłaby zawierać istniejące fakty w sposób zagregowany, aby poprawić wydajność.

* Dodaj kolumny do istniejących tabel: stosując tę zdenormalizowaną strategię uzyskasz lepszą wydajność, ale możesz skomplikować proces ETL i będziesz wymagać modyfikacji schematu narzędzia BI.

* Dodaj wiersze do istniejących tabel: dzięki tej strategii możesz stracić wydajność, ale konserwacja jest łatwiejsza.

W końcu faza jest takim nowym atrybutem, więc możesz użyć tych samych strategii, co w przypadku pozostałych atrybutów, znormalizowanych, używając go jako atrybutu lub zdenormalizowanego modelu tworzącego nowe fakty dla każdej wartości atrybutu.

Rzeczywistość kontra potencjał

Zdefiniowany model podstawowy pozwala nam mieć potencjalny widok danych, o ile jest zgodny ze strukturą płatka śniegu. O ile mamy jedną wersję produktu, jeśli zmienimy kategorię produktową produktu, zmieni się cała historia danych związanych z tym produktem. Ale możemy też mieć możliwość zapisania rzeczywistego widoku, jakim była kategoria produktu w czasie sprzedaży, poprzez zapisanie kategorii w tabeli faktów. Ważne jest, aby zapisać tylko odpowiednie pola dla tej rzeczywistej analizy, o ile zajmie to dużo miejsca w bazie danych, możemy mieć miliony lub miliardy wierszy w naszych tabelach faktów, więc zaoszczędzisz konieczności powtarzania tam wielu informacji, jeśli dodasz wszystkie atrybuty do tabeli faktów. Aby wybrać, które atrybuty są dla Ciebie istotne, wymagana jest ocena różnych tematów. Możesz zapisać niektóre atrybuty związane ze sprzedażą, ale być może będziesz musiał zapisać również niektóre zależności hierarchiczne, aby móc przeprowadzić żądaną analizę. Powtórzmy tutaj przykład z prawdziwego przypadku, który znaleźliśmy u niektórych klientów, który wyjaśniliśmy w części 1, kiedy mówiliśmy o hurtowniach danych. Pracowaliśmy nad projektem, aby przeanalizować wydajność sił sprzedaży, liczbę klientów odwiedzonych przez każdego członka sił sprzedaży, jak długo byli klientami, czynności wykonywane dla każdego klienta i tak dalej. Mieliśmy wymóg posiadania dwóch widoków, rzeczywistych, analizujących, kto odwiedził klienta; i potencjał, analizując, kto miał przypisanego klienta. W ciągu tego samego miesiąca były one zwykle takie same, ale z czasem może się to zmienić. Główny problem, który tam znaleźliśmy, polegał na tym, że dysponowaliśmy pracownikiem, który odwiedził tego klienta w celu opracowania danej czynności, ale nie zapisaliśmy, jakiego przełożonego wyznaczył; lub do jakiego obszaru należał, o ile w tabeli faktów mieliśmy tylko informację o pracowniku, a nie jego hierarchię. Trzeba też pomyśleć o zapisywaniu opisów czy nie. Może chcesz analizować dane z opisem, który miał na początku, a nie takim, jaki może mieć teraz, jeśli się zmienił.

Uwaga: Musisz być ostrożny, jeśli masz załadowane informacje o hierarchii do tabeli faktów, ponieważ nie będą one spójne, więc nie możesz zdefiniować relacji jeden-do-wielu nadrzędny-podrzędny za pomocą tej tabeli, ponieważ w pewnym momencie produkt będzie należeć do kategorii i po pewnym czasie może się to zmienić, więc relacja nie będzie jedna do wielu. Musisz także uważać na opisy; jeśli pojedynczy identyfikator atrybutu ma różne opisy, możliwe, że w narzędziu BI występują pewne niespójności, mimo że wygenerowany kod SQL może być poprawny.

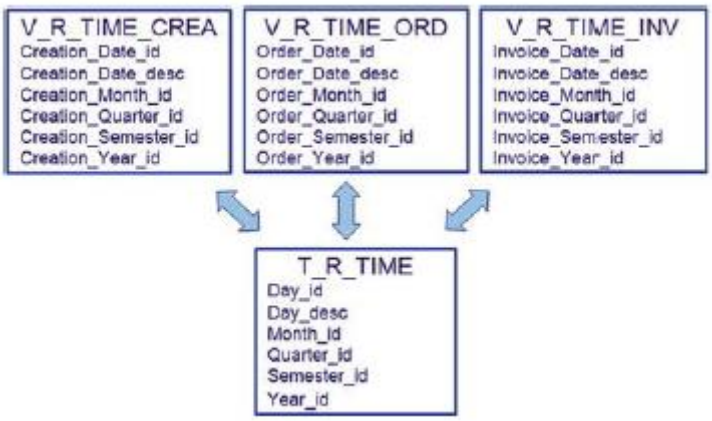
Istnieje również pewne ryzyko związane z dodawaniem wielu opcji dla użytkowników, jednym z nich jest to, że użytkownicy nie mają odpowiedniego przeszkolenia, aby z nich korzystać i mogą wątpić w prawdziwe znaczenie obu widoków, więc niektóre zaawansowane opcje powinny być jasno zdefiniowane w Twoim dokumentacji oraz na kursach szkoleniowych, które powinniście przeprowadzać dla swoich użytkowników.

Hierarchie historyczne

W odniesieniu do poprzedniej sekcji można również zdefiniować pewne tabele hierarchii historycznej, które pozwolą uzyskać rozwiązanie pośrednie między wartością rzeczywistą a potencjalną, dokonując przybliżenia atrybutów rzeczywistych bez zapisywania setek pól w tabelach faktów. Wewnątrz hierarchii historycznej zapiszesz status hierarchii ze zdefiniowaną częstotliwością, codziennie, co tydzień, co miesiąc, co kwartał lub co rok; a definiując relację atrybutu klucza i atrybutu czasu wraz z tabelą faktów będziesz mógł zobaczyć jak wyglądała hierarchia w danym okresie. Standardową procedurą jest posiadanie tabeli zależności bazowych, a następnie kopiowanie jej w każdym okresie, zapisując za każdym razem całe „zdjęcie" tabeli. Aby określić okres, którego użyjesz do hierarchii, będziesz musiał zrównoważyć dokładność przypisania hierarchii z wolumenem, który chcesz zachować, i wynikającą z tego wydajnością. Będzie to zależeć od wielkości tabeli związanej z podstawowymi relacjami, ale przez większość czasu posiadanie miesięcznej podstawy jest wystarczającą dokładnością dla użytkowników. Możesz także zdefiniować strategię informacji data_od/data_do, w której masz ważność każdego wiersza na podstawie daty rozpoczęcia definiowania do daty zakończenia. Głównym problemem w tym przypadku wymagane sprzężenia nie są bezpośrednie i trzeba ocenić odpowiednią hierarchię między klauzulami, co może wpłynąć na wydajność i skomplikować model BI, zwłaszcza jeśli zdefiniujesz go w narzędziu BI. W takim przypadku zalecamy użycie widoków do zdefiniowania relacji między tabelami.

Wiele widoków przy użyciu tabeli, izolacja jednostek

Są chwile, kiedy masz podobne atrybuty, które mogą mieć te same możliwe wartości, więc mogą używać tej samej tabeli jako wyszukiwania, ale są to różne koncepcje. Możemy myśleć o obszarach geograficznych, takich jak regiony, kraje i miasta, które są takie same dla klientów i pracowników, ale mają różne znaczenia. Analiza sprzedaży według regionu klienta nie będzie taka sama jak analiza sprzedaży według regionu pracownika. W naszym podstawowym modelu możemy znaleźć przykład z polami zawierającymi informacje o dacie. Zdefiniowaliśmy wiele pól dat, datę utworzenia, kiedy dokument sprzedaży jest tworzony; datę zamówienia, kiedy zamawiany jest dokument sprzedaży oraz datę wystawienia faktury, kiedy dokument sprzedaży jest fakturowany. Obecnie są one zdefiniowane jako pola informacyjne, o ile nie ma połączenia z wymiarem czasowym przez te pola, więc nie można agregować z obecną definicją do wyższych poziomów czasowych, takich jak miesiąc, kwartał lub rok. Aby to zrobić, będziesz musiał zdefiniować jednostki czasu dla wszystkich tych koncepcji dat; nie można ponownie użyć tych samych jednostek czasu, miesiąca, kwartału i roku do zdefiniowania wielu atrybutów, ponieważ koncepcje są różne, a system BI może wykonywać niepożądane połączenia. Z drugiej strony tworzenie różnych tabel nie ma sensu, ponieważ będą one zawierać te same powiązane informacje, dlatego w tym przypadku zalecamy użycie warstwy widoku do zdefiniowania różnych widoków tych samych tabel, jak pokazano na rysunku.



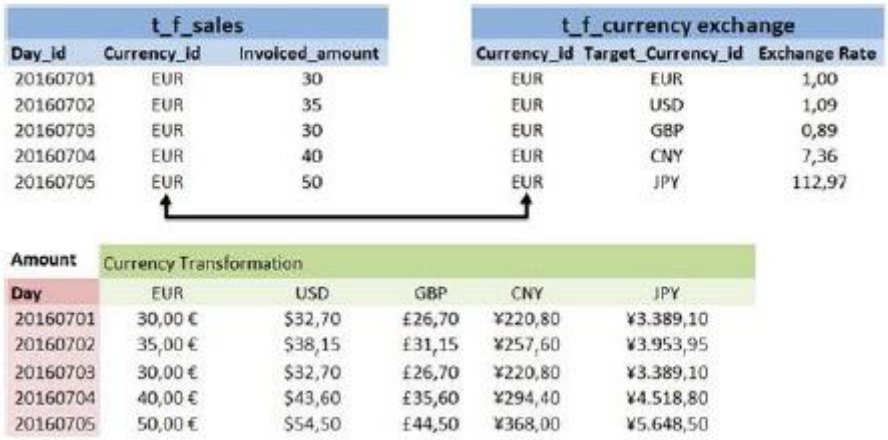
Tam możesz zobaczyć przykład dla głównej tabeli relacji, ale należy to zrobić dla wszystkich tabel związanych z czasem w modelu, t_l_miesiąc, t_l_kwartał i t_l_rok.

Modyfikowanie istniejących struktur, dodawanie kolejnych statusów dokumentów sprzedaży

W obiegu dokumentów sprzedaży zdefiniowaliśmy cztery statusy: Wyceniony, Zamówiony, Dostarczony i Zafakturowany. Moglibyśmy nieco skomplikować ten model, dodając dwa kolejne statusy do tego przepływu pracy, Zapłacono i Zwrócono, o ile Odoo pozwala nam zapisać płatność iw razie potrzeby zwrócić pieniądze klientowi. Spowoduje to zwiększenie złożoności modelu i nowych kolumn zapisujących datę płatności, zapłaconą ilość, zapłaconą kwotę, datę zwrotu, zwróconą ilość i zwróconą kwotę. Ale i ta zmiana pozwoli nam analizować nowe KPI, takie jak średni KPI dni płatności czy wskaźnik zwrotu. Nie uwzględnimy tego ani w modelu podstawowym, więc nie komplikujemy przykładu, który zamierzamy zaimplementować w tej książce, ani w części ETL, aby zbytnio nie rozszerzać tego rozdziału, ale jest to również interesujące do zrozumienia że zdefiniowany przez nas model nie jest statyczny, można go modyfikować w dowolnym momencie. Weź tylko pod uwagę, że jeśli zamierzasz zmodyfikować tabelę, która jest ładowana przyrostowo, będziesz musiał ponownie załadować całą tabelę, jeśli chcesz mieć pełne informacje dla nowych kolumn i prawidłowy status dla starych faktur.

Przekształcenie waluty

Zdefiniowaliśmy małą jednostkę dla waluty, aby mieć opis każdej waluty. Ale załóżmy, że działamy w międzynarodowym środowisku i mamy transakcje w różnych walutach. Aby móc przeprowadzić zagregowaną analizę całkowitej kwoty sprzedaży dla całej firmy, musisz mieć wszystkie informacje w tej samej walucie. Z punktu widzenia modelowania danych najłatwiejszym sposobem na to byłoby posiadanie wszystkich informacji w jednej walucie poprzez przekształcenie ich w procesie ETL, ale być może w pewnym momencie zechcesz zobaczyć je w innej walucie, dlatego zamierzamy zaproponować, abyś miał kilka opcji modelowania, aby spełnić to wymaganie, w zależności od szczegółów, które musisz podać. Potencjalna waluta: Najprostszym sposobem przeprowadzenia transformacji jest zdefiniowanie tabeli zawierającej walutę źródłową, walutę docelową i kurs wymiany. Kurs ten może być ostatnim dostępnym kursem lub średnią z określonego okresu, takiego jak bieżący miesiąc, poprzedni miesiąc, bieżący rok, ostatni rok, ostatnie 12 miesięcy lub cała historia hurtowni danych. Struktura umożliwiająca wykonanie przeliczenia waluty z potencjalnym widokiem jest pokazana na rysunku.



t_f_sales		
Day_Id	Currency_Id	Invoiced_amount
20160701	EUR	30
20160702	EUR	35
20160703	EUR	30
20160704	EUR	40
20160705	EUR	50

t_f_currency exchange		
Currency_Id	Target_Currency_Id	Exchange Rate
EUR	EUR	1,00
EUR	USD	1,09
EUR	GBP	0,89
EUR	CNY	7,36
EUR	JPY	112,97

Amount	Currency Transformation				
Day	EUR	USD	GBP	CNY	JPY
20160701	30,00 €	$32,70	£26,70	¥220,80	¥3.389,10
20160702	35,00 €	$38,15	£31,15	¥257,60	¥3.953,95
20160703	30,00 €	$32,70	£26,70	¥220,80	¥3.389,10
20160704	40,00 €	$43,60	£35,60	¥294,40	¥4.518,80
20160705	50,00 €	$54,50	£44,50	¥368,00	¥5.648,50

Prawdziwa waluta: Podejście w prawdziwej walucie jest podobne do potencjalnego, ale dodaje dzień w tabeli walut. W ten sposób połączenie z tabelą walut odbywa się z uwzględnieniem daty, więc masz kurs wymiany, który był oficjalny w momencie fakturowania rachunku. Analiza walutowa zwykle zawiera wiele zmiennych. Potencjalne podejście wykorzystujące jeden kurs wymiany jest przydatne, gdy analizujesz trendy biznesowe i nie chcesz mieć wpływu na analizę zmian kursów walut. W 2014 roku kurs dolara za euro wzrósł z 0,72 do 0,95, więc można by pomyśleć, że wasz amerykański firmy radziły sobie lepiej w tym roku, jeśli analizujesz dane w euro, lub twoje europejskie firmy osiągają gorsze wyniki, jeśli robisz to, a następnie analizujesz je w dolarach. Jeśli więc do analizy użyjesz statycznego kursu walutowego, unikniesz tego efektu w swojej analizie sprzedaży. Musisz wziąć pod uwagę, gdzie znajduje się twoje konto bankowe i jaka jest używana waluta. Jeśli Twoje konta bankowe znajdują się w kraju źródłowym z walutą kraju, za każdym razem, gdy chcesz skonsolidować pieniądze na jednym koncie, zastosujesz kurs wymiany, który masz w tym momencie, dlatego warto analizować dane z aktualnym kursem wymiany . Z drugiej strony, jeśli otrzymujesz pieniądze w walucie docelowej zaraz po wypłacie, ponieważ Twoje konto bankowe korzysta z waluty docelowej, sprawi to, że sens ma prawdziwą analizę transformacji walutowej. Kolejną zmienną, którą należy wziąć pod uwagę, jest pole daty używane do wymiany. W celu dokładnej analizy należy użyć daty transakcji, kiedy klient zapłaci rachunek… ale w tym przypadku nie mamy jej dostępnej w oryginalnym modelu, więc najlepszą opcją powinno być użycie Dnia (pamiętaj, ta kombinacja wśród Utworzenie, Zamówienie i Data wystawienia faktury w zależności od statusu), przynajmniej jeśli nie dodamy daty płatności zawartej w poprzedniej sekcji. Posiadanie analizy potencjału pozwala również wziąć pod uwagę ogólną sytuację, oceniając, czy kurs wymiany wpływa na Twój biznes. Coś szczególnie interesującego, jeśli prowadzisz firmę z wieloma transakcjami międzynarodowymi. Odnośnie danych zawartych w rzeczywistej tabeli musimy upewnić się, że mamy informacje za wszystkie dni; w przeciwnym razie złączenie z tą tabelą może spowodować utratę niektórych danych. Musimy także określić, skąd czerpiemy te informacje ponieważ jest całkiem możliwe, że nie jest dostępny w twoim systemie ERP, więc jedną z propozycji może być proces analizowania sieci, który eksploruje publiczną sieć w celu zebrania danych dotyczących wymiany walut. Lub jeszcze lepiej, pobierz dane z jakiejś usługi internetowej bezpośrednio w naszych procesach ETL. informacje geograficzne możliwości raportowania i mapowania danych geoprzestrzennych są interesującymi cechami prawie wszystkich narzędzi BI dostępnych na rynku, mniej lub bardziej zintegrowanymi z oprogramowaniem, a większość narzędzi zapewnia pewien sposób graficznego przedstawiania informacji na mapie. Ale aby móc wyświetlać informacje na mapie, będziesz mieć pewne wymagania dotyczące danych, które powinieneś spełnić, sposób, aby narzędzie zrozumiało, gdzie musi zlokalizować figury lub narysować obszary lub punkty. Będziesz miał głównie dwie możliwości: Twoje narzędzie jest w stanie geolokalizować informacje na podstawie opisu lub musisz podać długość i szerokość geograficzną swojemu narzędziu BI, innymi słowy, Twój model musi

pozwalać na zapisanie tych informacji w bazie danych. Aby skorzystać z pierwszej opcji, musisz upewnić się, że opisy, których zamierzasz użyć do zlokalizowania informacji na mapie, odpowiadają oczekiwanym nes przez twoje narzędzie; w przeciwnym razie nie będzie w stanie go zlokalizować. Jeśli chodzi o drugą możliwość, będziemy mieli dwie główne opcje: uzyskać informacje ze źródłowych systemów ERP lub CRM, jeśli nasze siły sprzedaży mają jakieś narzędzie mobilne do zbierania informacji geoprzestrzennych lub użyć narzędzia do geokodowania, aby uzyskać te informacje na podstawie pisemnych adresów. Masz wiele opcji na rynku w zakresie geokodowania, Google Maps geocoding API, ArcGIS, Mapbox - i jest całkiem pewne, że w niedalekiej przyszłości będziesz miał wiele innych możliwości.

Narzędzia do modelowania danych

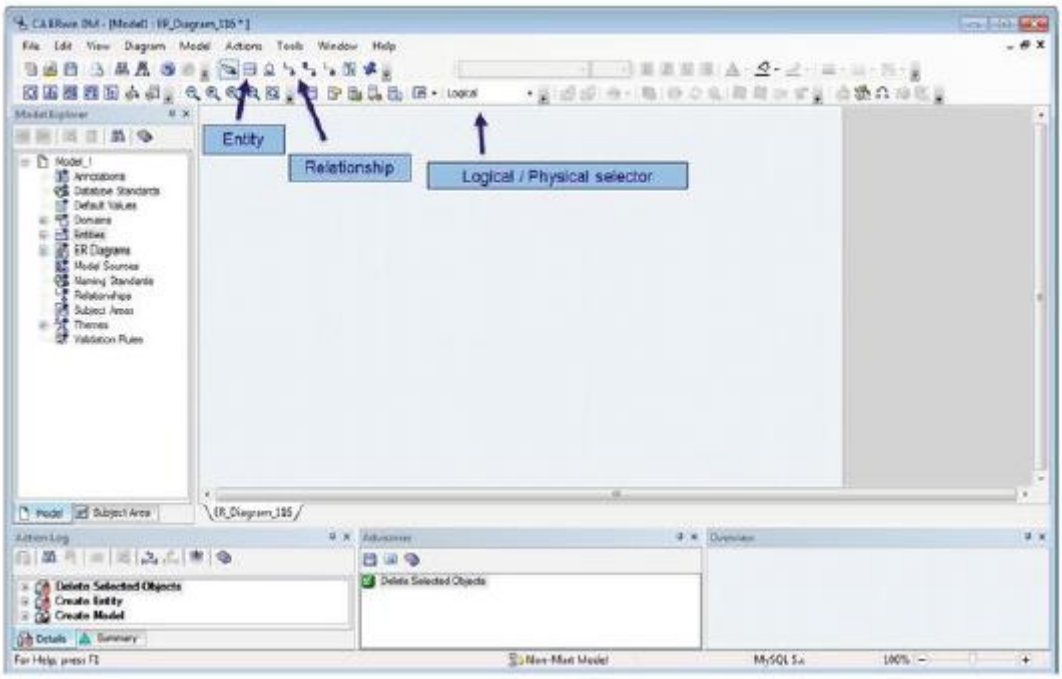
Istnieje wiele narzędzi do modelowania, które mogą pomóc w procesie modelowania. W tej sekcji omówimy dwa z nich, Erwin DataModeler i MySQL Workbench, które znamy i które uważamy za interesujące również w ich darmowej wersji. Jeśli masz budżet na ich komercyjną edycję, będziesz mieć wsparcie i wiele dodanych funkcji, które jeszcze bardziej ułatwią Twoje zadania, ale dzięki ich bezpłatnej edycji będziesz mieć również wystarczającą liczbę funkcji, aby kontynuować bieżący scenariusz modelowania . Jak zwykle w tej książce musimy ostrzec, że nie będzie to wyczerpująca eksploracja możliwości narzędzi do modelowania danych, a jedynie przegląd podstawowych funkcji, które naszym zdaniem będą potrzebne do zdefiniowania modelu.

DataModeler Erwina

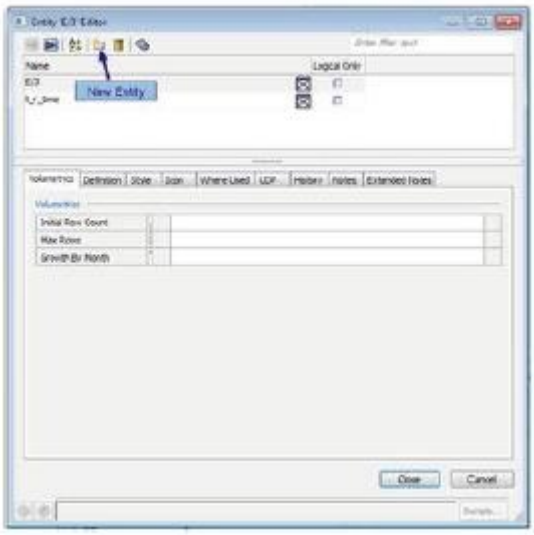
Zacznijmy od Erwin DataModeler, o ile to on posłużył do opracowania naszego podstawowego modelu. To oprogramowanie ma różne wersje, edycję społecznościową za darmo i kilka wersji komercyjnych. Jak można się domyślić w tym projekcie będziemy korzystać z wersji community. Aby go zainstalować, możesz go znaleźć na stronie http://erwin.com/products/data-modeler/communityedition/ , gdzie w prawej części strony znajduje się link Pobierz teraz. Pobierz odpowiednią wersję, 32 lub 64-bitową dla swojego laptopa i uruchom instalację pliku wykonywalnego. Instalację można wykonać klikając obok wszystkich ekranów, o ile domyślnie instalowane są komponenty potrzebne do opracowania modelu logicznego i fizycznego. Po zainstalowaniu będziesz mógł otworzyć narzędzie, dostępne, jeśli zachowałeś domyślną ścieżkę do przycisku Start ➤ CA ➤ Erwin ➤ CA ERwin Data Modeler r9.64 (64-bitowy), przynajmniej jeśli masz komputer 64-bitowy . Po otwarciu tworzenie modelu rozpoczyna się od kliknięcia przycisku Nowy, pierwszego w lewym górnym rogu. Następnie zostaniesz poproszony o menu, jak pokazano na rysunku

W tym kroku będziesz mógł wybrać typ modelu, który chcesz zaimplementować (w tym przypadku wybierzemy Logiczny/Fizyczny, aby utworzyć oba typy); baza danych z której będziesz korzystał, o ile po zdefiniowaniu modelu będziesz miał możliwość uzyskania kodu do tworzenia wszystkich

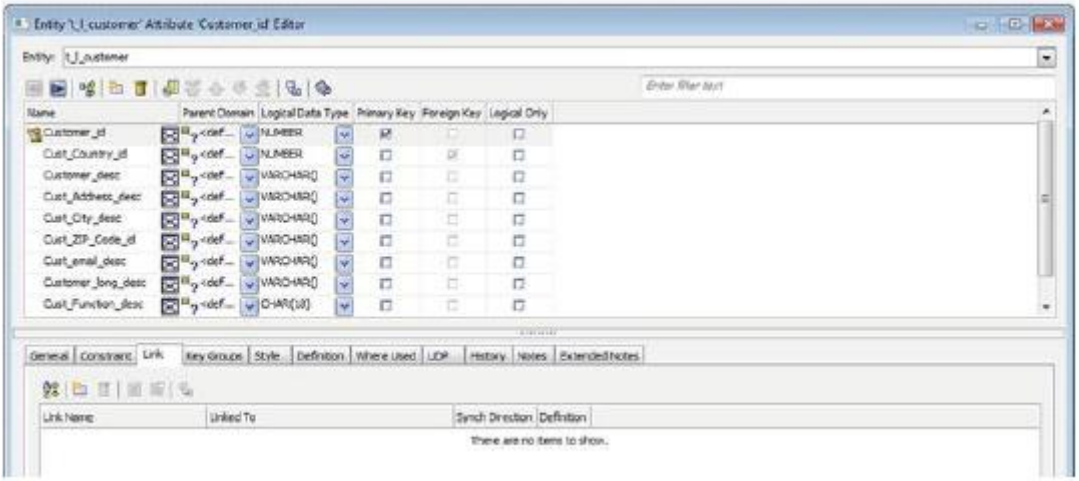
wymaganych tabel (my wybierzemy MySQL, o ile MariaDB używa tego samego kodu, jest to kontynuacja MySQL); oraz niektóre opcje szablonów o ile Erwin umożliwia zapisywanie szablonów w celu ułatwienia tworzenia nowych modeli. Na rysunku



możesz zobaczyć układ Erwina, z paskiem menu u góry ekranu, dwoma paskami przycisków tuż poniżej, a następnie eksploratorem modelu, w którym możesz zobaczyć obiekty modelu, oraz panelem projektu, w którym zdefiniujesz twój model. W dolnej części widoczne są trzy okna: Dziennik akcji, w którym można przejrzeć aktywność wykonaną w modelu, Poradniki z ostrzeżeniami i komunikatami; oraz sekcję przeglądu, w której można zobaczyć przegląd całego modelu i wyśrodkować okno w sekcji, którą należy zmodyfikować. Zauważ, że domyślnie projektujesz w modelu logicznym, ale możesz to zmienić za pomocą selektora logicznego/fizycznego, który masz na pasku przycisków, jak widać na rysunku wcześniejszym. Do narysowania modelu posłużymy się głównie przyciskami Entity i Relationship. Wybieramy Entity , a następnie klikamy w obszarze projektowania dodamy nowy podmiot. Ustawi nazwę jako domyślną, E/1, E/2, E/3... Będziesz mógł zmienić jej nazwę jednym kliknięciem na nazwę w celu ustawienia właściwej nazwy lub podwójnym kliknięciem otworzy kartę edytora jednostek, która pozwoli ci zmienić nazwy wszystkich jednostek na jednym ekranie, a także będziesz mieć wiele kart dla każdej jednostki, aby zmienić niektóre właściwości, niektóre, które wpłyną na skrypt tworzenia tabeli, takie jak Wolumetria; inne wpłyną na format w obszarze projektowania, takie jak Styl lub Ikona, a także możesz uzyskać informacje o zmianach dokonanych w elemencie na karcie Historia. Wewnątrz edytora encji będziesz mógł dodawać encje, klikając przycisk nowej encji, jak widać na rysunku.

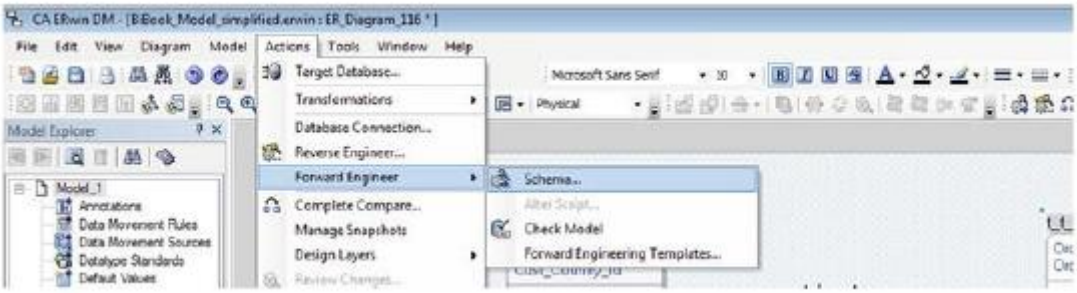


Gdy mamy już wszystkie jednostki zdefiniowane w naszym modelu, dodamy relacje, które będą łączyć nasze tabele. W tym celu należy kliknąć przycisk relacji (jeden do wielu lub wiele do wielu w zależności od rodzaju relacji) i przejść od najmniejszej liczności do większej (od „Jeden" do „Wiele"). Na przykład w naszym modelu od kwartalnika do czasu lub od czasu do sprzedaży itp. Po zdefiniowaniu jednostek można je otworzyć, aby zdefiniować różne atrybuty, które będą powiązane z kolumnami w tabeli fizycznej. W rzeczywistości po dodaniu nowego atrybutu będziesz musiał wybrać typ pola, który będzie używany. Aby otworzyć edytor atrybutów w celu zdefiniowania atrybutów, kliknij obiekt prawym przyciskiem myszy i wybierz Właściwości atrybutu. Otworzy się edytor atrybutów dla wybranego podmiotu, w którym zdefiniujemy atrybuty, które będą zawierały nasz podmiot, w tym typ fizyczny, który zostanie użyty w skrypcie tworzenia bazy danych. Na rysunku zobaczysz definicję atrybutu dla tabeli t_l_customer, w której możemy zobaczyć różne atrybuty i które z nich będą atrybutami kluczowymi, oznaczonymi jako klucz podstawowy.

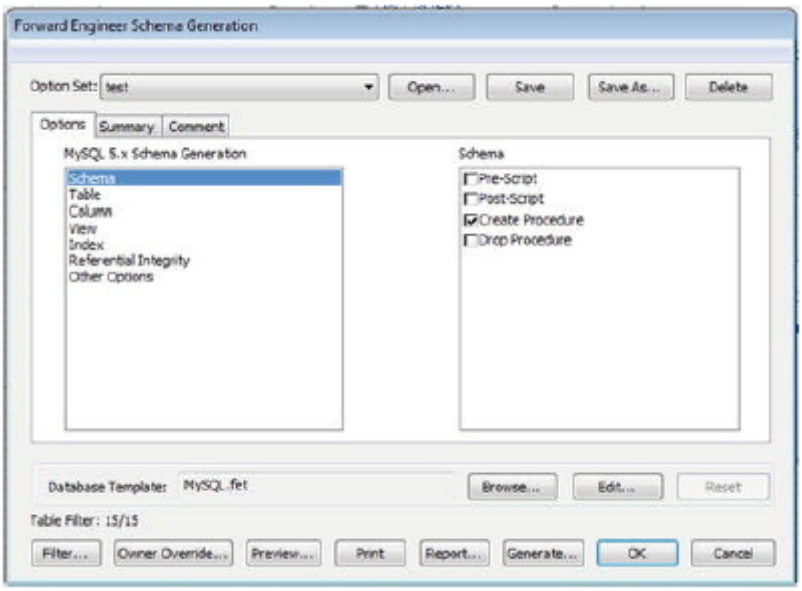


Gdy zdefiniujesz atrybut jako klucz podstawowy, jest on automatycznie definiowany jako klucz obcy w pozostałych tabelach zawierających ten atrybut, a także jest oznaczony jako klucz podstawowy powiązanych tabel (w naszym przykładzie oznacza identyfikator klienta jako klucz podstawowy w tabeli Sprzedaż). W przypadku, gdy ta definicja klucza podstawowego nie jest poprawna w powiązanej tabeli,

można ją usunąć z edytora atrybutów, w tym przypadku z tabeli t_f_sales. Po zdefiniowaniu wszystkich podmiotów i relacji dojdziesz do modelu pokazanego na rysunku wcześniejszym w poprzedniej sekcji „Definiowanie naszego modelu”. Jeśli chcesz zmienić jakąś fizyczną charakterystykę kolumn lub tabel, możesz przełączyć się do widoku Fizyczny, pamiętaj o selektorze Logiczny/Fizyczny z rysunku , a wtedy będziesz mieć możliwość otwarcia właściwości tabeli i kolumny zamiast elementu i atrybutu . Szczególnie interesujące jest to, że kolumna ,zakładka właściwości, podkarta MySQL, w której można zdefiniować dokładny typ kolumny, czy kolumna dopuszcza wartość pustą, czy nie, oraz kilka innych opcji specyficznych dla MySQL. Po ukończeniu modelu w oparciu o Twoje wymagania, możesz łatwo wygenerować skrypt, który utworzy model w bazie danych lub bezpośrednio połączyć model Erwina z bazą danych, a następnie bezpośrednio utworzyć wymagany schemat. Aby to zrobić, musisz być w widoku fizycznym i przejść do menu Akcje ➤ Do przodu Inżynier ➤ Schemat, jak pokazano na rysunku.



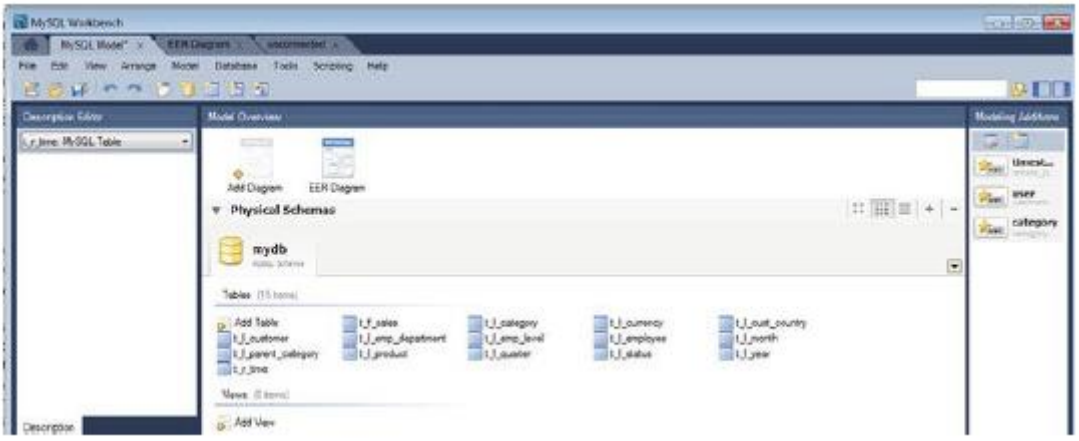
Po wybraniu tej opcji otworzy się nowe menu, w którym możesz sprawdzićktóre obiekty chcesz wygenerować i możesz następnie wybrać, czy chcesz wyświetlić podgląd skryptu, który pozwoli Ci wykonać go w dowolnym miejscu, czy też chcesz go wygenerować bezpośrednio, opcja, która będzie wymagać połączenia z bazą danych w tym momencie. Na rysunku możesz zobaczyć to menu z wieloma opcjami wyboru typów obiektów, opcjami skryptów dla każdego typu, podsumowaniem działań, które zostaną wykonane, oraz wieloma przyciskami do filtrowania obiektów, raportowania, co ma zostać zrobione itp.



Jeśli wybierzesz opcję Generuj, otworzy się nowe menu z pytaniem o parametry połączenia z bazą danych, w tym wersję bazy danych, nazwę użytkownika i hasło oraz używany ODBC, jak widać na rysunku . Po kliknięciu przycisku Połącz rozpocznie się tworzenie obiektów bazy danych

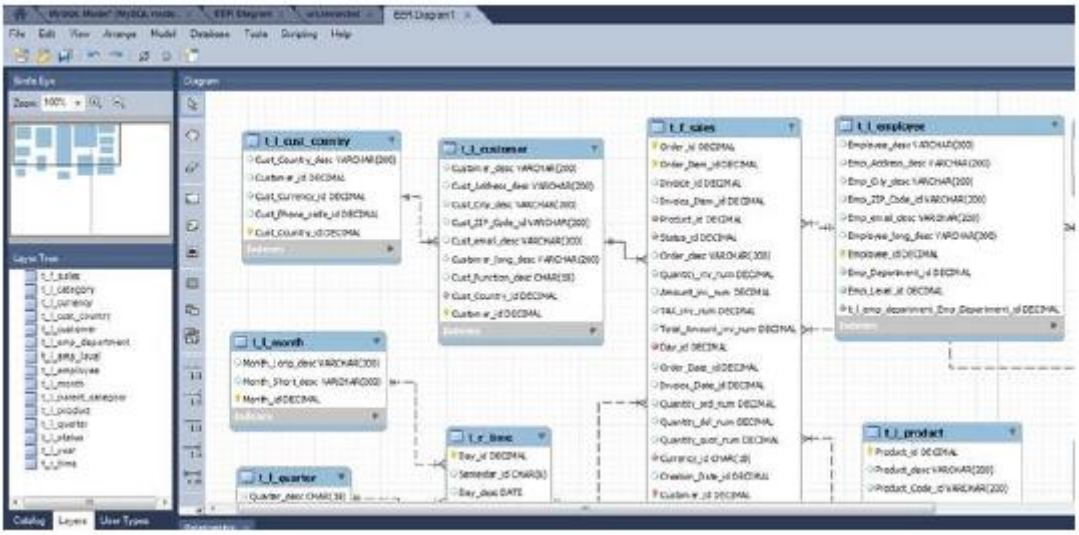
Środowisko pracy MySQL

Szybko przejrzymy narzędzie MySQL Workbench, które pomoże Ci w przyjazny sposób tworzyć diagramy, tabele i widoki, ale jest dedykowane tylko do baz danych MySQL (również do MariaDB o ile jest ewolucją MySQL), więc nie poświęcę na to zbyt dużo czasu. Prawie wszyscy dostawcy baz danych mają swój własny edytor, więc niemożliwe byłoby omówienie wszystkich opcji, które można znaleźć na rynku modelowania danych. Ale o ile baza danych MariaDB pozwala nam korzystać z MySQL Workbench, uważamy, że jest to interesujące narzędzie, o którym warto wiedzieć. Spróbujmy więc pokrótce wyjaśnić, jak go zainstalować i kilka szybkich kroków dotyczących tworzenia modeli za pomocą MySQL Workbench. MySQL Workbench jest dostępny bezpłatnie, ale aby uzyskać pomoc techniczną, jeśli jej potrzebujesz, musisz skorzystać z pomocy technicznej MySQL Enterprise, więc w tym przypadku może to oczywiście przekraczać Twój budżet, w zależności od wielkości budżetu. Produkty MySQL zwykle mają wersję Community, którą można pobrać i używać za darmo, a następnie, jeśli chcesz mieć wsparcie firmy Oracle (która niedawno kupiła MySQL), będziesz musiał uiścić opłatę za wsparcie. W środowisku produkcyjnym raczej nie zalecamy zakupu wsparcia dla bazy danych, ale w przypadku środowiska roboczego wsparcie to nie jest tak istotne. Jeśli wybierzesz MySQL Workbench jako opcję narzędzia do modelowania, pierwszym krokiem powinno być pobranie go. Aby to zrobić, możesz uzyskać bezpośredni dostęp do strony pobierania, http://dev.mysql.com/downloads/ i upewnić się, że jesteś objęty pobieraniem przez społeczność, aby móc korzystać z wersji komercyjnej, za którą powinieneś zapłacić. Po pobraniu i zainstalowaniu będziesz mieć dostęp do narzędzia, które domyślnie znajduje się w przycisku Start ➤ MySQL ➤ MySQL Workbench 6.3 (lub dowolnej wersji, którą pobierzesz). Jeśli klikniesz menu Plik ➤ Nowy model, będziesz mógł utworzyć nowy model. Po zakończeniu możesz dodawać tabele do modelu, aż dodasz je wszystkie, jak widać na rysunku



Również w tym przypadku, o ile skrypt był już tworzony z Erwinem, zaimportowaliśmy go. Możesz to zrobić za pomocą menu Plik ➤ Importuj ➤ Inżynier wsteczny MySQL Utwórz skrypt. Gdy już masz wszystkie tabele w swoim modelu, możesz utworzyć diagram, który pozwoli ci mieć widok twojego

modelu, tak jak to zrobiono w Erwinie, określając również, które tabele są powiązane, łącząc je z relacjami 1-N, używając 1- Przycisk relacji N. Możesz zobaczyć wynik na rysunku 5.22, jak można przypuszczać, jest dość podobny do tego zdefiniowanego w Erwinie.



Przygotowanie ETL

Jeśli postępowałeś krok po kroku zgodnie z instrukcjami zawartymi w tej książce, w tym momencie będziesz miał zdefiniowaną strukturę bazy danych, a jeśli korzystałeś z niektórych narzędzi do modelowania, które właśnie oceniliśmy, z łatwością utworzysz tę strukturę w Baza danych. Następnym krokiem powinno być ich wypełnienie, ale aby to zrobić, musimy wiedzieć, jakie źródło mamy do tego dostępne. Przeanalizujemy proces ich wypełniania w następnym rozdziale, ale zanim tam przeskoczymy, zbadamy lokalizację informacji. Naszym skromnym zdaniem najlepszym podejściem do tego jest strategia odgórna, identyfikująca systemy źródłowe, tabele źródłowe, które są łączeniem między nimi, jeśli potrzebujemy więcej niż jednej tabeli źródłowej do załadowania tabeli docelowej; i wreszcie, jakich pól źródłowych użyjemy i czy istnieje jakaś formuła lub obliczenie, aby osiągnąć pole docelowe. W tym kroku ważne jest również przedstawienie podsumowania strategii ładowania dla każdej tabeli lub wszelkich innych warunków filtra, które powinniśmy wziąć pod uwagę, aby prawidłowo wypełnić tabelę, na przykład jakiego klucza podstawowego powinniśmy użyć lub jakie kontrole integralności powinniśmy wykonać wykonać albo z kluczem obcym, albo z procesami kontrolnymi.

Systemy źródłowe

W naszym przypadku to zadanie jest dość łatwe, o ile wszystkie informacje, które wykorzystamy w naszym podstawowym modelu, będą pochodzić z naszego systemu Odoo ERP, więc to zadanie zostało wykonane szybko. Ale zwykle nie jest to takie łatwe. Możemy mieć różne moduły tego samego systemu ERP, sprzedaż, finanse, operacje, logistyka, zasoby ludzkie itp. Możemy również mieć wiele źródeł, jak skomentowaliśmy w alternatywnych modelach, które wyjaśniliśmy w poprzednich sekcjach, takich jak parsowanie sieci w celu uzyskania dzienny kurs wymiany walut, przy użyciu platformy geokodującej w celu uzyskania informacji o szerokości i długości geograficznej, i moglibyśmy pomyśleć o wielu innych systemach, takich jak narzędzia CRM, zewnętrzni dostawcy informacji, oficjalne statystyki, pliki płaskie, pliki csv lub Excel lub wszelkie inne możliwości o których moglibyśmy pomyśleć. Interesujące jest również posiadanie dostępnych parametrów łączności, takich jak nazwa serwera, baza danych, port, nazwa użytkownika, hasło, nazwa pliku, lokalizacja pliku, format pliku, częstotliwość aktualizacji, a

także informacje kontaktowe w celu rozwiązywania problemów lub wątpliwości, które mogą pojawić się przy definiowaniu procesu ETL .

Tabele źródłowe

Wyjaśniliśmy, jak załadować większość tabel, a ta informacja jest również przydatna, aby umieścić ją w tabeli podsumowującej, która pozwoli Ci łatwo sprawdzić, gdzie się udać, aby uzyskać informacje i jak wdrożyć proces ETL. Ale oprócz tych informacji zobaczmy, które tabele w naszym systemie źródłowym zawierają informacje, których będziemy potrzebować. Więc zdefiniujemy dla każdego wymiaru, które tabele są uwzględnione; jakiego systemu źródłowego będziemy potrzebować do wyszukiwania; w jaki sposób obciążenie jest zdefiniowane dla tej tabeli, przyrostowe lub pełne; które tabele wykorzystamy jako źródła i jakie warunki – mogą to być filtry lub wymagane łączenia między tabelami, jeśli mamy więcej niż jedną tabelę źródłową dla danego celu. Testowym sposobem na zebranie wszystkich tych informacji jest użycie formatu tabelarycznego zawierającego wszystkie te pola, takiego jak ten pokazany na rysunku

Dimension	Table	Source System	Load Type	Source Table/s	Conditions
Sales	t_f_sales	Odoo ERP	Incremental	sale_order sale_order_line sale_order_line_invoice_rel account_invoice account_invoice_line account_invoice_line_tax account_tax	sale_order.id=sale_order_line.order_id sale_order_line.id=sale_order_line_invoice_rel.order_line_id sale_order_line_invoice_rel.invoice_line_id=account_invoice_line.id account_invoice.id=account_invoice_id.invoice_id account_invoice_line=account_invoice_line_tax (outer) account_invoice_line_tax.tax_id=account_tax.id
	t_l_status	Manual	Static	N/A	
Currency	t_l_currency	Manual	Static	N/A	
Customer	t_l_customer	Odoo ERP	Incremental	res_partner	
	t_l_cust_country	Odoo ERP	Full	res_country	
Employee	t_l_employee	Odoo ERP	Incremental	hr_employee	
	t_l_emp_department	Odoo ERP	Full	hr_department	
	t_l_emp_level	Odoo ERP	Full	hr_job	
Product	t_l_product	Odoo ERP	Incremental	product_product product_template	product_product.template_id=product_template.id
	t_l_category	Odoo ERP	Full	product_category	
	t_l_parent_category	Odoo ERP	Full	product_category	
Time	t_r_time	Autogenerated	Full		
	t_l_month	Autogenerated	Full		
	t_l_quarter	Autogenerated	Full		
	t_l_year	Autogenerated	Full		

Pola źródłowe

Na koniec będziemy musieli przejść do przodu w każdej tabeli, aby zbadać, które pole źródłowe musimy wziąć, aby wypełnić każde pole docelowe, jakie reguły biznesowe i formuły musimy zastosować, a także wszelkie warunki, które uważamy za istotne, aby uzyskać prawidłowe obliczenie. Na rysunku możemy znaleźć formułę dla tabeli t_f_sales, która jest najbardziej złożona w naszym modelu.

Table	t_f_sales	
Field Name	**Source Table**	**Source Field**
Order_id	sale_order	id
Order_Item_id	sale_order_line	id
Invoice_id	account_invoice	id
Invoice_Item_id	account_invoice_line	id
Customer_id	sale_order	partner_id
Employee_id	sale_order	user_id
Product_id	sale_order_line	product_id
Status_id	Calculated	if invoice_id is not null then invoiced else if qty_to_invoice <>0 then Ordered else Quoted
Order_desc	sale_order	name
Quantity_inv_num	account_invoice_line	quantity
Amount_inv_num	account_invoice_line	price_sub_total
TAX_inv_num	account_invoice_line account_tax	price_sub_total*amount
Total_Amount_inv_num	Calculated	amount_inv_num + TAX_inv_num
Day_id	Calculated	if invoice_id is not null then account_invoice.date else if qty_to_invoice <>0 then sale_order.date_order else sale_order.creation_date
Order_Date_id	sale_order	date_order
Invoice_Date_id	account_invoice	date
Quantity_ord_num	sale_orde_line	qty_delivered
Quantity_del_num	sale_orde_line	qty_delivered
Quantity_quot_num	sale_orde_line	product_uom_qty
Currency_id	sale_orde_line	currency_id
Creation_Date_id	sale_orde_line	creation_date

Dostarczamy to tylko jako przykład, ale wszystkie tabele w modelu powinny mieć analizę pola źródłowego, taką jak ta pokazana.

Wniosek

Model danych jest podstawą Twojej analizy BI. Poprawny model zdefiniowany w tym momencie projektu pozwoli Ci zaoszczędzić wielu bólów głowy w przyszłości, dlatego gorąco polecamy poświęcić czas na ten punkt, aby upewnić się, że proponowany model z łatwością rozwiąże pytania, które będą zadawać Twoi użytkownicy Twój system BI. Kontynuacja proponowanych kroków zapewni projekt wykonany z wystarczającą jakością i dokumentacją, aby zrozumieć przyczyny struktury modelu; W przeciwnym razie, jeśli po prostu stworzysz tabele, które uznasz za istotne, możemy zapewnić, że w niedalekiej przyszłości stracisz źródło wielu decyzji podejmowanych w modelu. Z pewnością możesz potrzebować projektów i testów Proof of Concept bez przechodzenia przez modele biznesowe, logiczne, wymiarowe i fizyczne, ale kiedy już będziesz wystarczająco jasny, że strategia modelu spełni twoje wymagania, powinieneś wykonać wszystkie kroki, aby jasno zorganizować definicję modelu. Teraz, gdy mamy już zdefiniowany nasz model, przejdźmy do tego, jak wypełnić go niektórymi narzędziami ETL i jak czerpać z tego zyski za pomocą platformy BI.

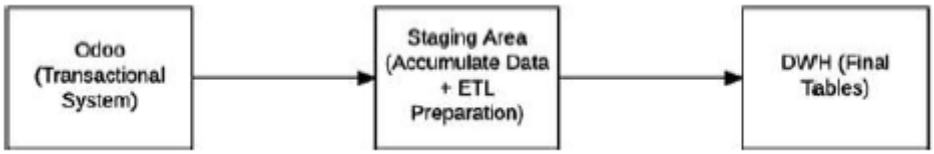
6. Podstawy ETL

Dlaczego potrzebujemy procesu ETL?

Można się zastanawiać, skoro mamy już dane w naszych bazach danych, dlaczego nie skierować naszych narzędzi do raportowania na te bazy i zacząć pracować w raportach i dashboardach, prawda? Cóż, życie zwykle nie jest takie łatwe. Przede wszystkim nigdy nie zaleca się bezpośredniego łączenia z operacyjnym systemem danych. Zapytania generowane przez narzędzia do raportowania, a nawet przez użytkowników, mogą być bardzo czasochłonne podczas zbierania zasobów potrzebnych systemom operacyjnym. Powodów jest jednak więcej. Czasami danych nie da się wykorzystać w taki sposób, w jaki są one przechowywane w systemie operacyjnym i wymaga pewnego rodzaju transformacji, której nie da się wykonać w narzędziu do raportowania lub w czystym SQL. A czasami, uwierzcie nam, zazwyczaj tak jest, zwłaszcza gdy firma ma rozsądną wielkość, musimy zintegrować dane ze źródeł zewnętrznych: dostawców, punktów sprzedaży, różnych baz danych, plików Excel lub zwykłych plików tekstowych, a nawet danych z Internet. W tym scenariuszu potrzebujemy sposobu na połączenie danych, skonsolidowanie ich i przechowywanie w formacie, który możemy przeszukiwać. Ponadto, w większości przypadków będziemy musieli przejść przez etap czyszczenia danych, ponieważ dane z systemów źródłowych nie zawsze są kompletne, poprawne lub nadające się do przechowywania. Takie podejście ma wiele zalet. Oczywiście komplikuje to sprawę, ponieważ wymaga od nas opracowania tych procesów, ale w zamian unikamy uderzania w systemy operacyjne; możemy również modyfikować nasze dane, aby odpowiadały naszym potrzebom, a jeśli chcemy, możemy stosować różne zasady przechowywania danych, co zapewnia nam niespotykany dotąd stopień elastyczności.

Szczegóły rozwiązania

Zanim zagłębimy się w szczegóły, chcemy przedstawić Wam nasz pomysł na końcowy przepływ danych. Mamy nasz system transakcyjny oparty na Odoo, w którym zbierzemy kilka tabel i pobierzemy je na obszar przejściowy. Z tego obszaru przejściowego wykonamy całą agregację czyszczenia, filtrowanie i manipulacje potrzebne do wypełnienia naszych końcowych stolików. Jest to zwykle najczęstsze podejście, jeśli chodzi o pobieranie i ładowanie danych. To podejście jest zwykle wybierane spośród innych, ponieważ nie wywiera presji na system operacyjny.Jeśli zapytamy ponownie, Odoo, system transakcyjny, może mieć wpływ na wydajność. Dzięki temu rozwiązaniu wszystkie operacje będą wykonywane na bazach danych staging i dwh, które nie są częścią systemu operacyjnego, więc obciążenie pojawi się na tych bazach. Zwykle ekstrakcja będzie prowadzona w nocy lub poza godzinami szczytu, więc proces przesyłania danych, który odbywa się między obszarem transakcyjnym a strefą postojową, nie wpływa na system operacyjny na godziny pracy, które mają być godzinami krytycznymi; oraz godziny, w których system jest pod większą presją. Stąd ostateczny schemat rozwiązania będzie podobny do tego, co widać na rysunku



Pakiety ETL typu open source

Mamy dobre bazy danych typu open source, ale prawdopodobnie mamy jeszcze lepsze pakiety ETL typu open source. W tej książce przyjrzymy się dwóm narzędziom i chociaż bardziej skoncentrujemy się na Pentaho Data Integrator (znanym również jako Kettle), zobaczymy również kilka przykładów użycia Talend Open Studio. Wybraliśmy tych dwóch, ponieważ są wystarczająco dojrzali, cały czas

ewoluują i są szeroko stosowane, więc cieszą się dużym wsparciem społeczności. Istnieje kilka komercyjnych narzędzi ETL, ale uwierz nam, jeśli nie masz bardzo konkretnych wymagań, których nie można spełnić za pomocą jednego z bezpłatnych narzędzi, nie ma żadnego powodu, aby za nie płacić. Zwykle są łatwe w zarządzaniu, wydajne i mają niekończący się zestaw wtyczek do łączenia z dowolnym źródłem danych, jaki możesz sobie wyobrazić, wszystko za darmo. Możliwe jest również, że jeśli korzystasz z jakiejś komercyjnej bazy danych w swojej firmie, uwzględniłeś licencję na korzystanie z własnego narzędzia ETL, tak jak ma to miejsce w przypadku Microsoft SQL Server i narzędzia SS Integration Services.

Pobieranie i instalowanie integracji danych Pentaho

Aby rozpocząć, musimy najpierw pobrać i zainstalować narzędzie Pentaho Data Integration (od teraz PDI lub Kettle). Możemy pobrać plik PDI zip na ich stronie internetowej: http://community.pentaho.com/projects/dataintegration/ Kliknij sekcję pobierania, a to przeniesie Cię na dół strony internetowej, gdzie wyświetlana jest najnowsza wersja, która wynosi 6,1 w momencie pisania. Kliknij nazwę wersji, a otworzy się nowe okno, które uruchomi pobieranie. Plik jest dość duży, prawie 700 MB, więc przed rozpoczęciem pobierania upewnij się, że masz przyzwoite połączenie z Internetem. Po zakończeniu pobierania będziemy mieć plik ZIP z kilkoma programami, w rzeczywistości bibliotekami i klasami Java. Na potrzeby tego rozdziału, ponieważ będziemy projektować zadania i transformacje (później), będziemy używać komputera z systemem Windows, z interfejsem graficznym. Bez problemu możemy również korzystać z dowolnego systemu Windows lub Linux, o ile mamy zainstalowane środowisko Java JRE w naszym systemie (wersja 8 załatwi sprawę). Wejdź na stronę Oracle i pobierz ją, jeśli nie masz jej zainstalowanej na swoim komputerze. Później, aby zaplanować zadania, które projektujemy, zainstalujemy PDI na naszym serwerze Linux, ale ponieważ nie zainstalowaliśmy tam żadnego komponentu X Window, niemożliwe byłoby użycie edytora graficznego do zaprojektowania naszych zadań i transformacji. Podsumowując, serwer jest w porządku do uruchamiania zadań, ale do ich projektowania potrzebujemy komputera z interfejsem graficznym. Po rozpakowaniu pliku PDI-ce-6.1.0.1-196.zip (lub coś podobnego, w zależności od aktualnej wersji) w dowolnym miejscu na naszym komputerze, możemy uruchomić edytor GUI, klikając dwukrotnie plik Spoon. Upewnij się, że wybierasz odpowiedni dla swojego środowiska. Jeśli korzystasz z Linuksa, powinieneś wybrać spoon.sh i przed uruchomieniem go uczynić go wykonywalnym za pomocą polecenia chmod. Jeśli jesteś w systemie Windows, po prostu uruchom Spoon.bat. Jeśli skrypt może zlokalizować java w twoim systemie, program zostanie uruchomiony. Jeśli masz problemy z uruchomieniem go w systemie Windows, sprawdź następujące porady na stronie internetowej:

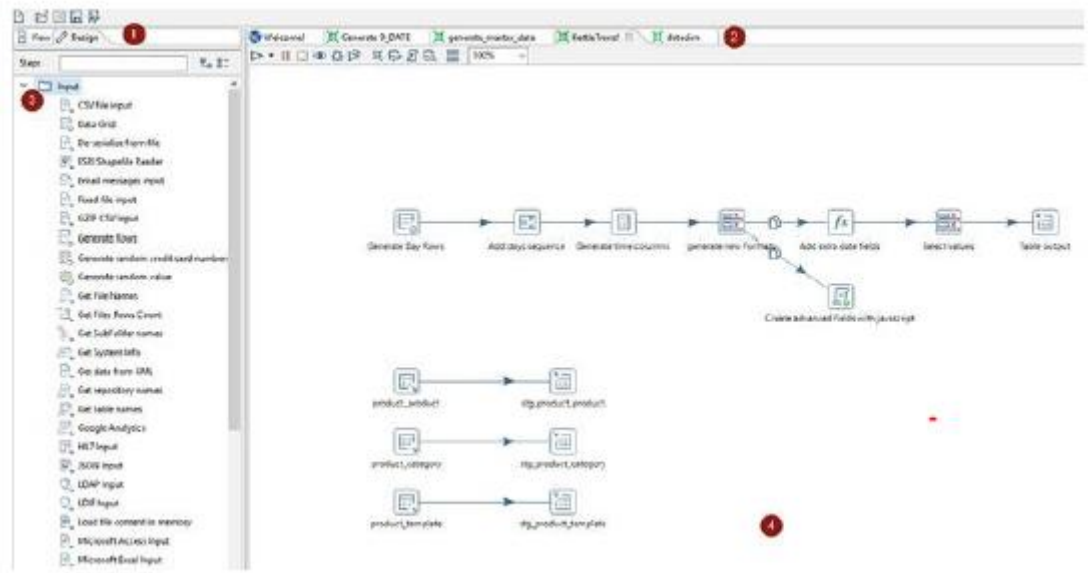
Edytuj plik Spoon.bat i:

Zamień w ostatnim wierszu „start javaw” tylko na „java”.

Dodaj „pauzę” w następnym wierszu.

Zapisz i spróbuj ponownie

Jeśli wszystko idzie dobrze, powinniśmy zobaczyć ekran podobny do pokazanego na rysunku .



Jeśli zobaczysz okno repozytorium z prośbą o utworzenie repozytorium, kliknij na razie anuluj. Utworzymy jeden później. Ważne jest, abyśmy w tym momencie zrozumieli, w jaki sposób rozmieszczone jest główne okno PDI. Cztery główne obszary, o których musimy wiedzieć, to:v

1. Karta Projekt lub karta Widok. Pierwszym jest dodanie nowych operacji do płótna, a drugim wyświetlenie już dodanych komponentów, połączeń itd.

2. Zakładka przekształceń i zleceń pozwala nam przełączać się z jednej przekształcenia lub opracowywanego zlecenia do drugiego.

3. Trzeci to miejsce, w którym możemy wybrać dowolną nową operację i przeciągnąć ją na płótno

4. Czwarta część to płótno. Wszystkie nasze operacje i przepływy muszą być zdefiniowane w kanwie. Kiedy otwieramy PDI po raz pierwszy, powinno mieć puste płótno, ale następnym razem otworzy te, nad którymi pracowaliśmy wcześniej.

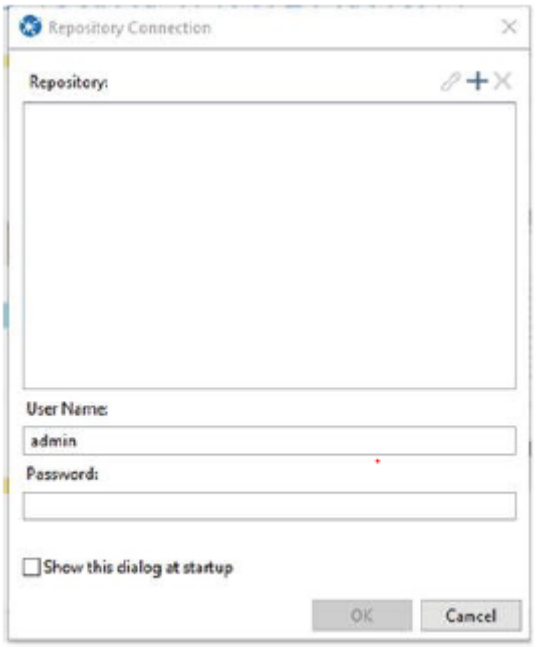
Zrozumienie koncepcji ETL

Aby zrozumieć ogólny projekt ETL, najpierw musimy zrozumieć kilka pojęć i obiektów używanych przez narzędzia ETL; zostały one przedstawione w tym podrozdziale, wraz ze sposobem ich tworzenia lub wykorzystania w PDI. Zwykle wszystkie ETL mają źródła i cele danych, czyli miejsce, w którym dane się znajdują i gdzie musimy się połączyć, aby je odczytać lub zapisać; wykonać na tych danych pewne akcje i manipulacje, które są pogrupowane w komponenty zwykle określane jako transformacja oraz komponent wyższego poziomu, który jest koordynatorem przepływu pomiędzy wszystkimi transformacjami, który jest zwykle nazywany zadaniem lub przepływem pracy. Nadszedł właściwy czas, aby przyjrzeć się tym komponentom dogłębnie.

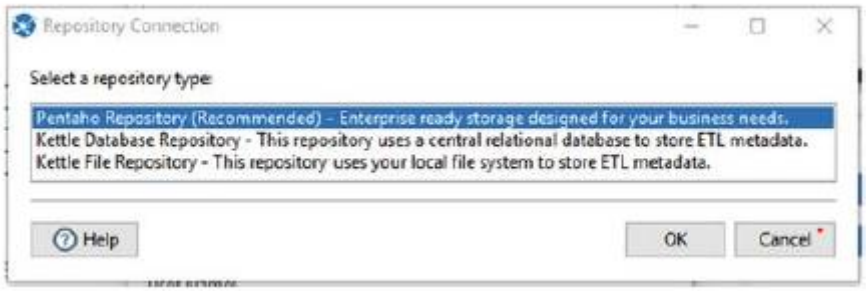
Repozytoria i połączenia

Jak w większości narzędzi ETL, repozytorium to metadane, w których będą przechowywane wszystkie nasze obiekty PDI. Repozytorium przechowuje zadania, transformacje, połączenia i wiele więcej. Jeszcze kilka lat temu w PDI istniały dwa rodzaje repozytoriów, Kettle Database Repository i Kettle File Repository, ale ostatnio dodano kolejny, Pentaho Repository. Ten najnowszy jest nowym zalecanym przez Pentaho, o ile jest najlepszy podczas pracy na dużym wdrożeniu. Ponieważ tak nie jest w naszym przypadku, oceniliśmy dwie opcje, korzystając z repozytorium bazy danych Kettle lub repozytorium

plików Kettle. W środowisku biznesowym sensowne jest przechowywanie repozytorium w relacyjnej bazie danych. Zwykle upraszcza to sprawę, ponieważ nie musimy kopiować zadań i transformacji do folderów, ale pozwala Kettle zarządzać nimi automatycznie. Musimy pamiętać, aby cały czas tworzyć kopię zapasową tego schematu bazy danych, ponieważ jeśli zostanie utracony, stracimy wszystkie wdrożenia Kettle. Ponieważ dopiero poznajemy PDI i nie chcemy tracić czasu na tworzenie nowych schematów baz danych, skorzystamy z łatwiejszego w zarządzaniu repozytorium plików. Zasadniczo zdefiniujemy folder w naszym systemie, aby usunąć zadania i transformacje. Należy jednak pamiętać, że do działania PDI nie jest konieczne tworzenie żadnego repozytorium. Możemy pracować bezpośrednio z plikami, a to zwykle najłatwiejszy sposób w środowisku nieprodukcyjnym. Aby utworzyć repozytorium, wykonaj następujące kroki: Z menu Narzędzia wybierz Repozytorium, a następnie Połącz lub naciśnij CRL-R; spowoduje to, że repozytorium będzie wyglądać tak, jak pokazano na rysunku, teraz nadal będzie puste.



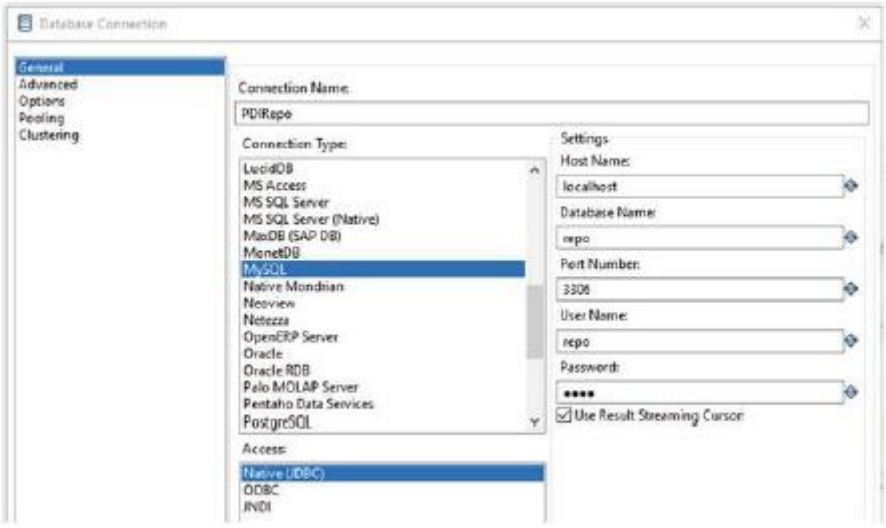
W prawym górnym rogu okna dialogowego znajduje się znak plus; kliknijmy go, a pojawi się nowe okno dialogowe z pytaniem, jaki typ repozytorium chcemy utworzyć. Możesz to zobaczyć na rysunku



Dla opcji repozytorium bazy danych wybierzemy drugą opcję. Aby pracować z repozytorium plików, wybierz trzecie. W naszym projekcie na razie będziemy pracować z plikami, ale zdecydowanie zalecamy korzystanie z opcji bazy danych w środowisku produkcyjnym.

Uwaga: Jeśli wybrałeś repozytorium bazy danych, pamiętaj o utworzeniu schematu bazy danych i przygotowaniu nazwy użytkownika i hasła do tego schematu, ponieważ będą one potrzebne do skonfigurowania repozytorium bazy danych Kettle.

Teraz musimy zdefiniować nasze połączenie z bazą danych. I tu pojawia się pierwszy problem. Nadal nie zdefiniowaliśmy żadnego połączenia. Nie stanowi to problemu, ponieważ możemy go utworzyć, klikając przycisk Nowy na ekranie. Jeśli spojrzysz na Typ połączenia, nie znajdziesz MariaDB, ale to nie jest problem, jak wyjaśniliśmy w poprzednim rozdziale: Sterowniki MySQL są w 100% kompatybilne. Wybierzmy więc MySQL, a następnie jesteśmy gotowi do utworzenia połączenia. Użyję bazy danych o nazwie repo, z repozytorium użytkowników i repozytorium haseł. Sprawdź na rysunku pola, które należy wypełnić.



Um, jeszcze nie skończyliśmy! Teraz nadeszły „złe” wieści. Domyślnie PDI jest dostarczane bez sterowników umożliwiających łączenie się ze zbyt wieloma bazami danych, w tym MariaDB i MySQL. Aby móc nawiązać połączenie, będziemy musieli pobrać dla niego sterownik Java. Jeśli go nie pobierzemy i nie spróbujemy się połączyć lub przetestować połączenia, zobaczymy komunikat podobny do tego na rysunku , informujący, że nie można znaleźć sterownika.

Aby uzyskać sterownik, musimy przejść do strony internetowej MySQL, na której znajduje się sterownik Java, po prostu przejdź do następującej strony: http://dev.mysql.com/downloads/connector/j/ w przeglądarce i pobierz plik zip ze sterownikiem. Jeśli używasz Linuksa, możliwe, że ten sterownik jest już zainstalowany na twoim komputerze; jeśli nie, możesz spróbować uruchomić dowolnego menedżera pakietów, aby go pobrać i zainstalować, lub wykonać tę samą procedurę, aby skopiować go do ścieżki klasy programu. Po pobraniu interesuje nas głównie plik jar o nazwie mniej więcej tak: mysql-connector-java-5.1.39-bin.jar. Następnie skopiuj ten plik do ścieżki lib swojej instalacji PDI. Następnie kliknij Anuluj w oknie dialogowym połączenia z bazą danych w PDI i zamknij PDI, ponieważ w przeciwnym razie nie będzie działać. Ponownie uruchamiamy plik wykonywalny Spoon i tym razem, jeśli skopiowaliśmy plik we właściwe miejsce, możemy przetestować połączenie i powinniśmy zobaczyć komunikat podobny do tego:

Connection to database [PDIRepo] is OK.

Hostname : localhost

Port : 3306

Database name : repo

Mamy gotowe pierwsze połączenie. Należy pamiętać, że połączenia zawsze działają w ten sam sposób. Nie ma znaczenia, czy jest to repozytorium, czy połączenie danych. Procedura ich zakładania jest zawsze taka sama. Upewnij się, że masz sterownik w folderze lib, wybierz odpowiedni smak bazy danych i uzupełnij szczegóły, odpowiednio nazywając połączenie z bazą danych! Jeśli zdecydujemy się na repozytorium plików, musimy tylko wybrać folder nadrzędny, w którym utworzymy nasze repozytorium, którym może być dowolny folder na twoim komputerze; nadaj mu nazwę, a następnie opis. Spójrz na rysunek , aby uzyskać szczegółowe informacje:

Następnie klikamy OK, a repozytorium zostanie utworzone i zostaniemy przekierowani z powrotem do ekranu powitalnego.

Transformacje, rdzeń Kettle

Transformacje wraz z zadaniami, które zobaczymy później, to obiekty, których używamy do definiowania operacji na danych. Zwykle zadanie zawiera jedną lub więcej transformacji, dlatego lepiej jest je najpierw zrozumieć. Transformacja to łańcuch kroków. Krok to czynność, która będzie wykonywana zwykle na danych. Ale nie ogranicza się to tylko do tego. Mamy kroki, które wchodzą w interakcję ze zmiennymi (wewnętrznymi z PDI lub zmiennymi systemowymi), inne, które wchodzą w interakcje z plikami, a jeszcze inne po prostu manipulują danymi, z którymi mamy do czynienia. W Kettle są dziesiątki różnych kroków. Niektóre z nich działają bezpośrednio w silniku PDI, podczas gdy inne działają w systemie plików lub systemie operacyjnym, a nawet z zewnętrznymi lub zdalnymi narzędziami: interakcja z bazami danych, plikami płaskimi, plikami Excel, plikami dostępu, xml,

serwerami WWW… a nawet systemem zmienne. Nie będziemy komentować wszystkich kroków PDI, ponieważ będzie to wymagało co najmniej jednej całej książki, zobaczymy tylko część z nich. Zasadniczo skoncentrujemy się na krokach, których potrzebujemy. Kroki są pogrupowane według kategorii, w zależności od funkcji, którą wykonują. Mamy wejście, wyjście, transformację, przepływ narzędzi, skrypty, wyszukiwanie, łączenie, a nawet kroki Big Data. Zasadniczo będziemy pracować z krokami wejścia i wyjścia, odczytywać i zapisywać dane z bazy danych i plików zewnętrznych, a także użyjemy niektórych kroków transformacji, które pozwolą nam zmodyfikować nasze dane, niektóre kroki przepływu (zwłaszcza krok filtrowania wierszy) i prawdopodobnie użyjemy niektórych kroków narzędzi, które zasadniczo nam na to pozwalają operować plikami: spakować je, wysyłać e-maile, zapisywać pliki dziennika i tak dalej. Istnieje inny zestaw kroków, zwany skryptami, który jest niezwykle potężny. Pozwala nam określić kod bezpośrednio w kroku. Ten kod można napisać w języku SQL, JavaScript lub Java.

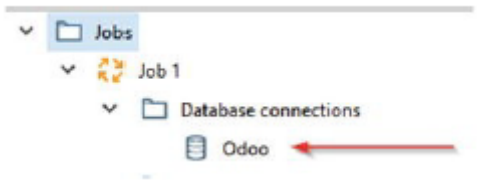
Transformacje muszą być umieszczone w określonej kolejności na płótnie. A następnie muszą być połączone, aby wyjście z poprzednika było wejściem późniejszej transformacji. Czasami możliwe jest określenie więcej niż jednego wyjścia. W takim przypadku PDI zapyta nas, czy chcemy zduplikować dane wyjściowe między późniejszymi zadaniami, czy też chcemy skopiować dane wyjściowe do każdego kolejnego kroku. Możliwe jest również posiadanie wyjść warunkowych. Niektóre kroki dzielą dane wejściowe między dwa wyjścia, w zależności od określonego warunku logicznego. Jeżeli warunek jest spełniony, wejście spełniające warunek zostanie przesłane do jednego kroku, natomiast wejście niespełniające warunku zostanie przekierowane do innego kroku. Zobaczymy to później na przykładzie transformacji.

Zadania, czyli jak zorganizować zestaw przekształceń w przepływie pracy

Zadania w Kettle to kontenery przekształceń i połączonych działań, które są również określane poprzez przeciąganie i upuszczanie kroków na płótnie zadania. Jednak rodzaje kroków, które możemy znaleźć w widoku zadania, różnią się nieco od tych, które możemy znaleźć w widoku transformacji. W widoku zadania mamy kilka kroków pogrupowanych według kategorii. Ogólne, które są głównymi składnikami zadań, Poczta, Zarządzanie plikami, Warunki (do kontroli przepływu), Skrypty, Ładowanie masowe, BigData, Modelowanie, XML, Narzędzie, Przesyłanie plików i Szyfrowanie plików - najważniejsze. Każda praca musi mieć punkt wyjścia. Dobrą praktyką jest użycie kroku sukcesu na końcu. A do debugowania możemy użyć Dummy Step. Zobaczymy więcej później. Krok początkowy możemy znaleźć w grupie ogólnej, więc aby rozpocząć projektowanie pracy, musimy przeciągnąć i upuścić krok początkowy na kanwę. Po wykonaniu tej czynności możemy dodać kroki transformacji, które będą wykonywały większość pracy, i połączyć je, przesuwając mysz nad krokiem, czekając sekundę i klikając znak „wyjdź” w kroku. Po kliknięciu nie klikaj ponownie i przeciągnij i upuść mysz do następnego kroku. Po wykonaniu tej czynności zostanie utworzone łącze między pierwszym krokiem a drugim. Należy zauważyć, że istnieją alternatywne sposoby łączenia kroków. Można wybrać dwa kroki, kliknąć prawym przyciskiem myszy pierwszy (źródłowy) i kliknąć „New Hop”. Spowoduje to utworzenie dokładnie tego samego przeskoku. Po utworzeniu przeskoku możemy go edytować, usunąć lub wyłączyć. Aby to zrobić, po prostu kliknij prawym przyciskiem myszy strzałkę przeskoku, a pojawi się menu kontekstowe z tymi działaniami. Jak wyjaśniliśmy wcześniej, istnieją kroki, które mogą tworzyć warunkowe przeskoki, które są wykonywane przez ocenę wyrażenia iw zależności od wyniku, wykonanie jednego lub drugiego kroku. W ogólnych przypadkach większość kroków będzie oznaczona jako bezwarunkowa, więc nad strzałką pojawi się ikona kłódki. Oznacza to, że krok docelowy zostanie wykonany niezależnie od wyniku kroku poprzedniego. Proces łączenia etapów transformacji jest dokładnie taki sam. Będziemy tworzyć przeskoki we wszystkich naszych Pracach i Transformacjach.

Utwórz i udostępnij połączenie

Aby rozpocząć programowanie z PDI, jeśli chcemy łączyć się z bazami danych, tak jak w naszym przypadku, musimy najpierw utworzyć połączenia. Aby utworzyć połączenie w PDI, możemy wykonać te same kroki, co wcześniej, kiedy wyjaśniliśmy, jak skonfigurować połączenie z metadanymi nowego repozytorium. W tej książce będziemy używać głównie dwóch połączeń. Jeden do PostgreSQL, w którym rezyduje Odoo, a drugi do bazy danych MySQL, która będzie hostować hurtownię danych. Teraz jest dobry moment, aby utworzyć te dwa połączenia i pozwolić im ustawić się na pozostałą część książki. Gdy stworzymy, udostępnimy go, aby wszystkie nasze zadania i transformacje mogły z niego korzystać, bez konieczności definiowania nowego w każdym zadaniu lub transformacji. Zobaczmy, jak to zrobić. Przed rozpoczęciem zainstalowaliśmy już złącze Java MySQL, jeśli postępowaliśmy zgodnie z instrukcjami w sekcji łączenie z repozytorium. Ale do połączenia metadanych Odoo potrzebujemy biblioteki java PostgreSQL. Domyślnie ten jest dostarczany z programem, przynajmniej w najnowszych wersjach, ale jeśli masz problemy z połączeniem, upewnij się, że sterownik znajduje się w odpowiednim folderze. W menu Plik kliknij Nowy, a następnie w nowym menu kliknij Połączenie z bazą danych. Pojawi się menu podobne do poprzedniego. W zakładce Ogólne upewnimy się, że wybieramy połączenie PostgreSQL oraz uzupełnimy znane nam już dane, takie jak nazwa użytkownika, nazwa bazy danych, hasło oraz adres IP serwera, na którym znajduje się baza danych. Następnie nadajemy połączeniu nazwę, na przykład Odoo, a następnie możemy kliknąć przycisk Test, aby upewnić się, że możemy się połączyć bez żadnych problemów. Jeśli wszystko idzie dobrze, powinniśmy zobaczyć pomyślny komunikat informujący, że można nawiązać połączenie. Klikamy OK i nasze połączenie zostanie utworzone. Od tego momentu pozostaje już tylko jeden krok, czyli w zasadzie udostępnienie połączenia. To jest proste zadanie. W lewym widoku upewnij się, że masz wybraną kartę Widok zamiast Projektowanie, a następnie poszukaj folderu o nazwie Praca lub Transformacje, w zależności od tego, czy otworzyłeś pracę lub transformację w kanwie. Wewnątrz znajdziesz swoją obecną pracę lub transformację, aw środku ponownie zobaczysz folder o nazwie Połączenia z bazą danych. Jeśli przeglądasz folder, powinieneś zobaczyć swoje nowe połączenie. Możesz zobaczyć ikonę połączenia z bazą danych na rysunku.



W tym momencie wszystko, co musimy zrobić, to kliknąć prawym przyciskiem myszy połączenie i kliknąć udostępnij. Jeśli wszystko idzie dobrze, jedyne, co zauważymy, to to, że nazwa połączenia jest pogrubiona. Oznacza to, że nasze połączenie jest teraz współdzielone i mimo to, jeśli nie zauważymy niczego szczególnego, będziemy mogli go używać we wszystkich naszych zadaniach i transformacjach. Jeśli nie widzisz ich w innych zadaniach lub transformacjach, zamknij je i otwórz ponownie. Powinieneś być teraz w stanie zobaczyć połączenie.

Uwaga: Domyślnie, jeśli pracujesz w środowisku opartym na repozytorium, połączenia są automatycznie udostępniane, więc nie ma potrzeby wykonywania tych kroków. Jeśli pracujesz z izolowanymi plikami, jest to konieczne.

Globalny obraz

Chcemy przeprowadzić analizę na podstawie tego, jakie produkty sprzedajemy. Pomysł polega na tym, że po przeprowadzeniu tej analizy będzie można odpowiedzieć na niektóre z tych pytań: które produkty są najczęściej sprzedawane, który pracownik osiąga najlepsze wyniki i jaka jest ewolucja naszej sprzedaży rok po roku. Aby móc przeprowadzić tę analizę, musimy zdefiniować proces ETL, który pobiera pewne dane z naszego systemu operacyjnego, przetwarza je i przechowuje w naszej hurtowni danych. Później będziemy mogli podłączyć kilka wymyślnych narzędzi graficznych do naszej małej bazy danych hurtowni danych i przeprowadzić analizę, która pomoże nam uzyskać odpowiedzi na nasze pytania. W tej części skupimy się na pierwszej części, czyli wydobyciu danych z Odoo i załadowaniu ich do naszej bazy danych hurtowni danych. Kroki będą następujące: załaduj wszystkie powiązane tabele produktów, których potrzebujemy do naszej analizy; załadować tabele dotyczące klientów i pracowników, dzięki czemu możemy śledzić produkty sprzedane przez pracownika lub zakupione przez klientów, a nawet razem; dane dotyczące sprzedaży, które utworzą naszą podstawową tabelę faktów; a następnie będziemy potrzebować szeregu czasowego, aby nadać temu wszystkiemu jakiś sens. Jak wyjaśniliśmy, chcemy zobaczyć ewolucję sprzedaży, a także inne trendy czasowe, na przykład sprzedaż sezonową i inne aspekty, dlatego musimy zbudować harmonogram, ponieważ wyodrębnianie tylko dat w prostym formacie nie jest odpowiednie i nie pozwoli nam na agregację według tygodnia, miesiąca, kwartału ani roku. Zacznijmy od tabel związanych z produktami.

Tabele produktów, kategorii produktów i kategorii nadrzędnych produktów

W modelu znajdują się trzy tabele, które są powiązane z produktami i ich kategoriami. Tabela t_l_product, t_l_category i t_l_parent_category. Nie są to dokładnie te same tabele, które możemy znaleźć w metadanych Odoo, ale są one podobne. Aby je wygenerować, potrzebujemy tylko trzech tabel z Odoo, jak widzieliśmy w ostatniej części rozdziału 5. Te tabele nazywają się product_category, product_product i product_template. Bezpośrednio pobierzemy te trzy tabele do obszaru przemieszczania, a następnie zbudujemy zapytania, aby je wygenerować. Możemy to uwzględnić w poprzedniej transformacji lub stworzyć nową. Ponieważ pracujemy tylko z obszarem przejściowym, możemy zgrupować je wszystkie w tej samej transformacji. Przeciągnij i upuść krok wprowadzania danych z tabeli na obszar roboczy i nie łącz go nigdzie. To będzie pierwsza gałąź naszej transformacji. Ponieważ nie wymuszamy kluczy podstawowych w obszarze przejściowym, możemy je ładować równolegle bez żadnych błędów. Kliknij dwukrotnie krok wprowadzania tabeli, wybierz połączenie Odoo z utworzonymi wcześniej metadanymi i kliknij przycisk Get SQL Statement. W nowym oknie dialogowym wyświetl folder Tables i poszukaj pierwszej tabeli o nazwie product_product. Wybierz tabelę i kliknij OK. PDI zapyta nas teraz, czy chcemy uwzględnić nazwy pól w zapytaniu SQL. Jeśli zdecydujemy się ich nie uwzględniać, zostaną one zastąpione gwiazdką, ale zwykle nie jest to dobra praktyka, więc odpowie TAK. Zostanie wygenerowane zapytanie podobne do tego:

```
SELECT

id

, create_date

, weight

, default_code

, name_template

, create_uid

, message_last_post
```

, product_tmpl_id

, barcode

, volume

, write_date

, active

, write_uid

FROM product_product

Teraz nadszedł czas, aby kliknąć przycisk Podgląd, aby upewnić się, że możemy pobrać dane z Odoo. Jeśli wszystko idzie dobrze, zostanie wyświetlona siatka z danymi. Kliknij Zamknij po sprawdzeniu. Przed zamknięciem kroku zmień nazwę kroku na product_product name. W tej chwili pobieramy tylko tabele do naszego obszaru przejściowego, więc nie przeprowadzamy jeszcze żadnej transformacji. W tym celu musimy teraz tylko upuścić krok danych wyjściowych tabeli w obszarze roboczym. Połącz oba kroki, przeciągając i upuszczając strzałkę wyjściową z wejścia tabeli do kroku wyjścia tabeli. Czas skonfigurować kilka znanych rzeczy. W tym celu nazwiemy wyjście i nazwę tabeli docelowej jako stg_product_product. Dobierzemy odpowiednie połączenie i zaznaczymy pole wyboru Truncate Target Table. Teraz prawie skonfigurowaliśmy nasze miejsce docelowe, ale znowu musimy utworzyć strukturę tabeli. Tym razem niestety PDI popełnia błąd. Jeśli naciśniemy przycisk SQL i spróbujemy utworzyć tabelę za pomocą tej instrukcji, zobaczymy, że trzy kolumny dat zostały zdefiniowane jako typy UNKNOWN, jak widać w poniższym fragmencie kodu:

CREATE TABLE stg_product_product

(

id INT

, create_date UNKNOWN

, weight DOUBLE

, default_code LONGTEXT

, name_template LONGTEXT

, create_uid INT

, message_last_post UNKNOWN

, product_tmpl_id INT

, barcode LONGTEXT

, volume DOUBLE

, write_date UNKNOWN

, active BOOLEAN

, write_uid INT

);

W tym momencie mamy kilka możliwych opcji:

* Popraw instrukcję tworzenia tabeli, wybierając odpowiednie typy danych i uruchom ją ręcznie w naszej docelowej bazie danych z pomocą dowolnego klienta.

* Usuń te pola, ponieważ nie potrzebujemy ich dla naszego modelu i nie ma potrzeby pobierania ich z metadanych.

Każda z dwóch określonych opcji powinna działać. Ale ponieważ celem tej książki jest wyjaśnienie rzeczy we właściwy sposób i wyjaśnienie, jak radzić sobie z nieoczekiwanymi rzeczami, wybierzemy pierwsze możliwe rozwiązanie. Zaktualizowana instrukcja tworzenia dla tego jest pokazana poniżej:

```
CREATE TABLE stg_product_product
(
id INT
, create_date DATETIME
, weight DOUBLE
, default_code LONGTEXT
, name_template LONGTEXT
, create_uid INT
, message_last_post DATETIME
, product_tmpl_id INT
, barcode LONGTEXT
, volume DOUBLE
, write_date DATETIME
, active BOOLEAN
, write_uid INT
)
```

Uruchom tę instrukcję w pomostowej bazie danych i ponownie uruchom transformację. Jeśli wszystko idzie dobrze, powinieneś zobaczyć, jak kończy się pomyślnie. Wróć do pomostowej bazy danych i uruchom następujące zapytanie, aby upewnić się, że rekordy zostały dołączone, a formaty dat są zachowane:

```
select * from stg_product_product;
```

Powtórz tę samą procedurę dla tabel product_template i product_category.

Uwaga: Aby ułatwić sobie zadanie, możesz wybrać z wciśniętym klawiszem Control w obu krokach (wejście i wyjście tabeli) oraz skopiować i wkleić w obszarze roboczym. Następnie musisz tylko edytować kilka rzeczy, zmieniając nazwy tabel zarówno w kroku wejściowym, jak i wyjściowym, i jesteś gotowy do pracy.

Prawdopodobnie napotkasz ten sam problem z typem danych UKNOWN, z jakim mieliśmy do czynienia wcześniej. Po prostu zmień je na typ DATETIME w tworzonych skryptach tabeli i jesteś gotowy do pracy.

Tabele klientów i krajów klientów

Sposób, w jaki poradzimy sobie z tymi dwoma tabelami, jest prawie taki sam, jak w przypadku produktów. Pierwszym krokiem będzie pobranie ich do postoju z naszych metadanych Odoo, a następnie zrobimy inną transformację, aby pobrać wymagane dane z tabeli pomostowej i zgromadzić je w naszej hurtowni danych. Ponownie, istnieje wiele strategii, takich jak obcięcie + wstawienie i wstawienie (połączenie między aktualizacją a wstawieniem, istniejące klucze są aktualizowane, wstawiane są nowe) najczęstsze. W książce nie ma wystarczająco dużo miejsca, aby szczegółowo opisać każdą transformację, ale kroki, które należy wykonać, są bardzo podobne do tych w poprzednim zestawie tabel. W takim przypadku początkowo będziemy szukać jednej tabeli w metadanych Odoo: res_partner, która zawiera informacje o pracownikach, własnej firmie, dostawcach, klientach itp. W tej tabeli znajdują się pola, które pomagają nam określić, jakiego rodzaju podmiotem jest każdy wpis w tabeli, na przykład pole company_type, które określa, czy podmiotem jest osoba, czy firma, lub nadmiarowa flaga is_company. Kolumny dostawca, klient i pracownik mogą również służyć do dostosowywania danych, które pobieramy z metadanych, i unikania pobierania danych, których nie potrzebujemy.

Stwórz transformację klienta

Kiedy już znamy procedurę, cały czas są to te same kroki. Jak widać, wyraźnie potrzebujemy trzech tabel, aby móc wypełnić dwie, które mamy w naszej hurtowni danych. Są to hr_employee, res_country i res_currency. Aby Ci pomóc, podamy Ci kilka wskazówek. Jeśli zaczynałeś w res_partner, jesteś na dobrej drodze, ale przeczytanie danych z tej tabeli nie wystarczy. Oto kilka wskazówek, jak postępować:

* Aby uzyskać pozostałe dane dla t_l_cust_country, w res_partner znajduje się kolumna o nazwie country_id będąca kluczem obcym tabeli res_country, z której możemy pobrać kraj pracownika.

* W tej samej tabeli res_country mamy inną kolumnę, currency_id, która jest kluczem obcym do res_currency, w którym możemy uzyskać nazwę waluty faktur klienta.

Pozostawiamy tobie, jak utworzyć transformację, aby pobrać te trzy tabele w obszarze przejściowym, ale jak zwykle kod jest dostępny w repozytorium github, więc zawsze możesz porównać swój wynik z dostarczonym przez nas rozwiązaniem, a jeśli nie w stanie go uruchomić, masz działającą kopię do użycia.

Tabele pracowników i kategorii pracowników oraz działów pracowników

Jeśli pomyślnie zakończyliśmy transformację klienta, pracownik nie będzie miał przed nami tajemnic. Musimy powtórzyć dokładnie te same kroki, które wykonaliśmy poprzednio, ale zamiast używać flag w tabeli res_partner do zbierania klientów, musimy teraz użyć hr_employee do pobrania danych naszych pracowników. Zwróć uwagę, że może być konieczne zainstalowanie aplikacji „Katalog pracowników" w Odoo, nie po to, aby mieć przykładowe dane, ale sprawdzić je w interfejsie, na wszelki wypadek, gdybyśmy chcieli się bawić. Będzie to pomocne w tym ćwiczeniu. Jest jednak dodatkowy krok: aby zebrać opis funkcji pracownika, czyli rolę pracownika, musimy połączyć tabelę hr_employee z hr_jobs za pomocą kolumny job_id, aby zebrać pole name z tej ostatniej tabeli, które jest rolą przez pracownika w firmie. Następnie musimy zebrać trochę danych z działu hr_department, aby zebrać dział pracowników. Ale w tym momencie jedyne, co musimy zrobić, to określić, skąd wziąć te pola, abyśmy

mogli bezpośrednio pobrać wszystkie trzy tabele do naszego obszaru pomostowego, a po ostatecznej transformacji wykonamy łączenia.

Tabela faktów: jak stworzyć transformację dla sprzedaży

Do tej pory większość przekształceń była prosta. Sprawy staną się teraz nieco bardziej skomplikowane, ale nie martw się, stopniowo będziemy zwiększać złożoność. Mamy teraz wszystkie wymiary oprócz czasu. Skoncentrujmy się teraz na uzyskaniu bardziej znaczących danych oraz danych potrzebnych do analizy spostrzeżeń naszej firmy. Aby zrozumieć tę transformację, najpierw musimy zrozumieć, w jaki sposób będziemy podchodzić do sprzedaży w naszej firmie. Wyjaśniliśmy to już w rozdziale 5, ale dla odświeżenia utworzymy zamówienia dla naszych klientów, a kiedy zamówienie będzie gotowe, dostarczymy towar i wystawimy fakturę naszym klientom. Ta faktura jest oczywista i będzie zawierała pewną liczbę produktów zgodnych z liniami produktów oraz łączną kwotę zafakturowaną na linię pozycji, w tym podatki. Następnie mamy wszystkie relacje z naszymi tabelami wymiarów, takie jak produkty, pracownicy, klienci i ręczna tabela statusu. Jest to wymagane, aby powiązać zamówienie i fakturę z naszymi wymiarami, abyśmy mogli zapytać bazę danych, aby dowiedzieć się, kto co sprzedał, komu i jakie produkty są najczęściej sprzedawanymi. Aby skonstruować tę tabelę, musimy pobrać dane z kilku tabel w metadanych Odoo. To są:

* Dla danych zamówienia musimy pobrać dane z sale_order i sprzedaż_zamówienie_online.

* W przypadku faktur linia_konta_faktury i_linia_faktury_konta.

* Dla relacji między dwiema tabelami sale_order_line_invoice_rel.

* Ze względów podatkowych chcemy również pobrać plik account_tax i konto_oline_faktury_podatek.

Przed pobraniem tych tabel tutaj musimy najpierw o tym pomyśleć. Jeśli stoły nie są bardzo duże, możemy w całości pobrać je na naszą inscenizację. Jednak czasami tak nie jest, a nasze stoły mogą być bardzo duże. W takim przypadku zamiast pobierania pełnych tabel możemy zastosować warunek do niektórych pól ze znacznikiem czasu, takich jak pole write_date, i pobrać tylko najnowsze dane. W tym celu postępujemy zgodnie ze standardową procedurą: przeciągamy i upuszczamy krok wprowadzania tabeli do naszego płótna, wybieramy tabelę jak zwykle, a następnie odpowiadamy twierdząco na pytanie, czy chcemy, aby wszystkie kolumny były określone w klauzuli select. Następnie musimy tylko dodać dodatkowe zdanie na końcu zapytania, aby pobrać tylko najnowsze dane. Jeśli procesy ładowania działają dobrze każdego dnia, będziemy pracować nad rozdziałem planowania i możemy zdecydować się na pobieranie danych z poprzedniego dnia. Jako środek bezpieczeństwa, jeśli dobrze współpracujemy z procesami wstawiania aktualizacji, które ładują dane z obszaru pomostowego do hurtowni danych, możemy pobrać dane z dwóch dni, na wypadek, gdybyśmy mieli dzień wolny od pracy i z jakiegokolwiek powodu proces się nie powiódł. Pobaw się trochę z częścią czasową równania. Znasz lepiej niż my swoje wymagania, a także bierzesz pod uwagę ilość danych, które wydobywasz. Oświadczenie powinno być podobne do następującego:

```
SELECT

id

, origin

, create_date

, write_uid

, team_id
```

, client_order_ref

, date_order

, partner_id

, create_uid

, procurement_group_id

, amount_untaxed

, message_last_post

, company_id

, note

, "state"

, pricelist_id

, project_id

, amount_tax

, validity_date

, payment_term_id

, write_date

, partner_invoice_id

, user_id

, fiscal_position_id

, amount_total

, invoice_status

, "name"

, partner_shipping_id

FROM sale_order

where write_date > current_date - INTERVAL '2' DAY;

Mając to na uwadze, możemy zrobić to samo dla reszty zestawu tabel, więc pobieramy tylko te dane, które są naprawdę potrzebne do naszej inscenizacji. Tylko pamiętaj, że musisz upewnić się, że wybrałeś opcję obcinania tabeli docelowej w danych wyjściowych tabeli, aby nie dodawać danych w tabeli pomostowej. W przypadku pozostałych czterech tabel wymienionych na początku tego punktu możemy postępować według dokładnie tego samego schematu. Teraz nadszedł czas, aby przejść do bardziej złożonych rzeczy: tworzenia wymiaru czasu.

Tworzenie wymiaru czasu

Czas przejść do bardziej skomplikowanych spraw. Z tego rozdziału dowiesz się, czego prawie zawsze będziesz potrzebować w każdym procesie ETL. Jeśli pamiętasz z poprzedniego rozdziału, w którym widzieliśmy model ER dla hurtowni danych, są tam cztery tabele czasu. Spośród tych tabel najpierw interesuje nas wypełnienie tabeli t_h_time, która zawiera pięć pól. Są to dzień, miesiąc, kwartał, semestr i rok. Po wypełnieniu tego, pomyślimy o tym jako wypełnieniu pozostałych trzech, ponieważ te zostaną ukończone bezpośrednio z głównego. Zanim zaczniemy pisać transformację w celu wygenerowania tych danych, musimy zdefiniować kilka ważnych rzeczy. Są to:

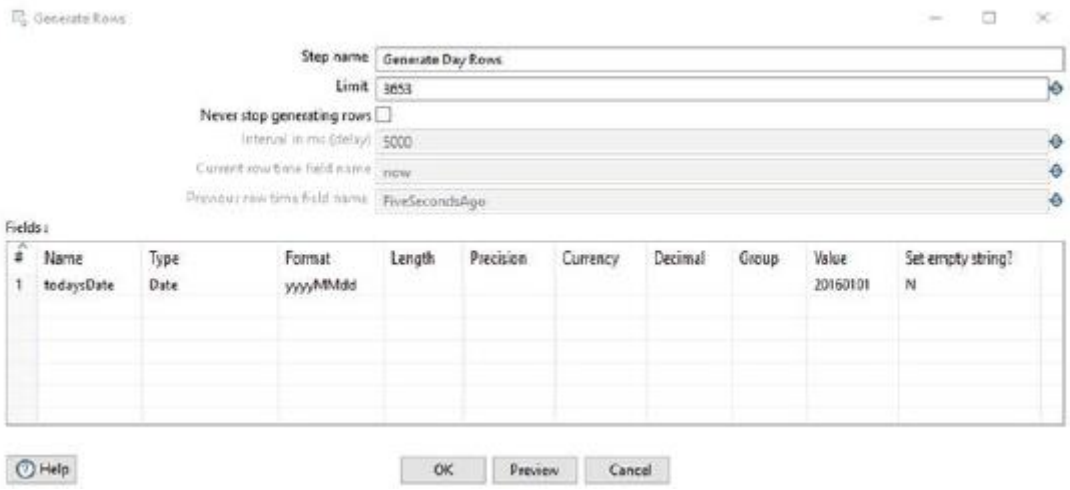
* Zdefiniuj, jaki będzie nasz pierwszy dzień w tabeli, a jaki ostatni. Możemy to zdefiniować dynamicznie lub statycznie. Jeśli zdecydujemy się na pierwszą opcję, będziemy musieli skorzystać z etapu scalania, ponieważ dane będą każdego dnia inne. Aby uprościć sprawę, wybierzemy stały zakres dat, ponieważ nie będziemy codziennie przeładowywać tej tabeli. Będziemy używać strategii ładowania raz, przez cały czas trwania projektu.

* Zdefiniuj formaty pól i klucz podstawowy tabeli. Aby wybrać klucz podstawowy tabeli, nie mamy zbyt wielu możliwości wyboru. To musi być unikatowa wartość, a jedyna z pięciu to kod dnia. Wybranym formatem będzie RRRRMMDD, który jest łatwy do odczytania, i unikamy przechowywania dat w naszym systemie, zwłaszcza jako części klucza podstawowego, co zwykle nie jest dobrą opcją, zarówno ze względu na wydajność, jak i zrozumienie i użyteczność.

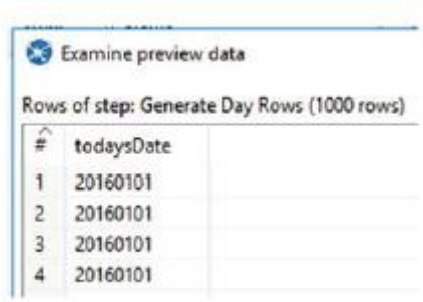
Uświadomienie sobie tych rzeczy pomoże nam i uprości proces ETL dla tabeli wymiarów czasu. Zaczynajmy! Pierwszym krokiem jest otwarcie PDI, jeśli mamy je zamknięte, następnie powinniśmy przejść do Plik, następnie w podmenu kliknąć Nowy, a następnie Nowa transformacja. Otworzy się puste płótno transformacji.

Wygeneruj tyle dni, ile potrzebujemy do przechowywania

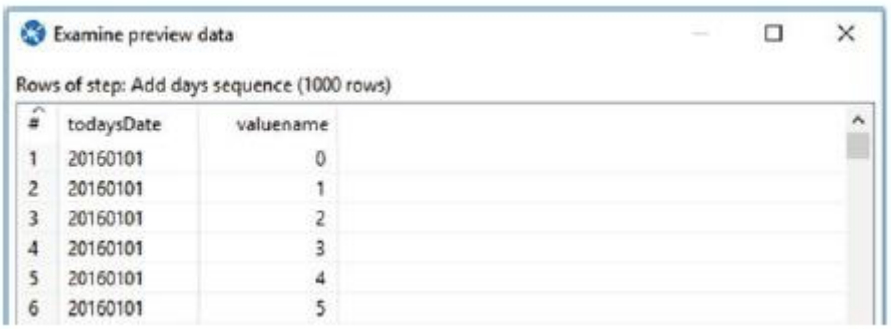
Pierwszym krokiem jest utworzenie wpisu dla każdego dnia, który chcemy zapisać. W poprzednim wprowadzeniu zdecydowaliśmy, że nie będziemy aktualizować tabeli wymiarów czasu. W tym kroku musimy więc uwzględnić długi okres czasu. Biorąc pod uwagę, że otworzyliśmy nasz sklep 1 stycznia 2016 roku, mamy nadzieję, że sklep będzie działał przez co najmniej 10 lat. Musimy więc utworzyć 365 dni (w przybliżeniu) * 10 lat = 3650 wpisów w naszej tabeli. Po jednym na każdy dzień tego 10-letniego okresu. PDI posiada wygodny krok do generowania wierszy na podstawie zakresu wprowadzonego przez dewelopera. Jak być może myślisz, krok nazywa się Generowanie wierszy i można go znaleźć w kroku grupy danych wejściowych. Możemy skorzystać z przeglądarki lub wyszukać go ręcznie w folderze Input. W każdym razie musimy przeciągnąć go z zakładki Projekt na płótno. Gdy mamy krok w naszym kanwie, czas go edytować, aby skonfigurować go do naszych potrzeb. Po prostu kliknij go dwukrotnie, a otworzy się nowy ekran, na którym możemy wypełnić wiele szczegółów i dostroić go.Wypełnimy go, jak pokazano na rysunku



Pierwszym wymaganym polem jest Step Name i powinniśmy je odpowiednio nazwać. Chociaż możemy pozostawić wartość domyślną, zwykle nie jest to dobra praktyka, ponieważ prawdopodobnie w pewnym momencie będziemy musieli zmodyfikować lub ponownie przyjrzeć się tej transformacji i trudno będzie to zrozumieć. Zdecydowaliśmy się nazwać to: Generuj wiersze dnia. Możesz użyć dowolnej nazwy, o której myślisz. Bardzo ważne jest również drugie pole. Będzie to liczba wygenerowanych wierszy. Ponieważ chcemy wygenerować dane z 10 lat, wpiszmy 3653 dni, czyli dokładnie tyle samo dni, które upłynęły między 1 stycznia 2016 r. a 31 grudnia 2025 r. Pozostałe pola pozostawmy nietknięte. Siatka jest wówczas wyświetlana w dolnej części okna dialogowego. Tutaj tworzymy kolumny danych, które chcemy wygenerować. Ponieważ chcemy wygenerować zestaw wierszy dla każdego dnia w kalendarzu, w zasadzie potrzebujemy teraz tylko jednej kolumny, dnia. Później zajmiemy się miesiącem, semestrem, kwartałem i rokiem. Po prostu skup się na przygotowaniu pierwszego dnia, a następnie możemy manipulować danymi, aby wygenerować inne potrzebne nam kolumny. Tak więc jedyne, czego potrzebujemy, to wpis w siatce. W kolumnie nazwa wpiszemy TodaysDate, w Type wpiszemy Date aw formacie wpiszemy rrrrMMdd, który zapisze naszą datę w wygodnym formacie. Nie martw się o typ daty, możemy go później zmienić, a właściwie musimy, ponieważ nadal musimy operować datą, aby uzyskać inne pola. Ostatnią, ale bardzo ważną kolumną jest kolumna Wartość. Wprowadź tutaj wartość początkową w poprawnym formacie rrrrMMdd. W naszym przypadku będzie to 20160101, co przekłada się na 1 stycznia 2016. Przed przejściem do kolejnych kroków i przed kliknięciem OK upewnij się, że okno dialogowe jest takie samo, jak pokazano na rysunku powyżej. Nadszedł czas, aby upewnić się, że wszystkie dane wychodzące z tego kroku są poprawne i zgodne z tym, co zdefiniowaliśmy. Aby przeglądać dane bez ich uruchamiania, możemy kliknąć przycisk Podgląd, a na ekranie pojawi się nowe okno dialogowe z danymi. Domyślnie wyświetlany jest podgląd tylko pierwszych 1000 wierszy. Upewnij się, że odpowiadają one formatowi, który wybraliśmy, jak pokazano na rysunku . Nie martw się, że cały czas powtarzamy dokładnie ten sam rząd, ponieważ tego oczekujemy. Następnie w kolejnym kroku będziemy operować na każdym wierszu, aby wygenerować odpowiednie dane.



Ok, mamy teraz tyle rzędów, ile dni musimy dodać do tabeli końcowej, ale nadal te informacje są dość bezużyteczne. Mamy 1 stycznia 2016, powtarzający się wszędzie i nie tego chcemy. Bez smutków. PDI zapewnia nam kolejny krok transformacji o nazwie Add Sequence. Użyteczność tego kroku jest kluczowa, ponieważ pozwoli nam stworzyć sekwencję, zaczynając od żądanej liczby, w naszym przypadku zero, i operować na przychodzących danych i tej sekwencji. Widzisz to, prawda? Tak! dodamy datę plus kolejny numer i zapiszemy je jako dane wyjściowe nowego dnia. Zaczniemy więc od 1 stycznia 2016 r. i będziemy dodawać jeden dzień więcej do każdego wpisu w danych kroków. Zobaczmy, jak zastosować tę logikę w kroku PDI. Na karcie Projekt wyszukaj folder Przekształć, aw środku znajdziemy krok Dodaj sekwencję. Następnie musimy przeciągnąć i upuścić go na płótno. Przed edycją kroku zawsze dobrze jest powiązać go z poprzednim krokiem, który mamy, ponieważ niektóre kontrole kroku polegają na odczytaniu danych (zwykle metadanych o danych, które pochodzą) z poprzednich kroków, a bez tego nie ma sposób, w jaki PDI może z nich korzystać. Aby połączyć oba kroki, klikamy pierwszy, aw nowym menu, które się pojawi, naciskamy przycisk wyjścia i przeciągamy strzałkę nad drugim krokiem. Po zakończeniu upuszczamy mysz, a łącze zostanie utworzone tak, jak widzieliśmy wcześniej. Możemy teraz dwukrotnie kliknąć drugi krok i skonfigurować go. Ponownie musimy to nazwać, więc tym razem użyjemy „Dodaj sekwencję dni" jako nazwy opisowej. Nazwa wybranej przez nas wartości to nazwa_wartości, ale może to być dowolna nazwa. Pozostałe pola pozostawiamy bez zmian oprócz start at value, które zamiast 1 zmienimy na 0, gdyż inaczej pierwszym dniem naszej tabeli kalendarza będzie 2 stycznia 2016, a to nie jest pożądany wynik . Niestety ten krok nie ma przycisku Podgląd, ale nie powinno to powstrzymać nas przed podglądem. Na szczęście istnieje inny sposób podglądu danych na każdym kroku, choć wymaga to przeprowadzenia transformacji. Kliknij prawym przyciskiem myszy ten drugi krok i poszukaj opcji podglądu. Kliknij go, a pojawi się okno dialogowe. Następnie kliknij Szybkie uruchamianie. Spowoduje to uruchomienie transformacji do tego momentu i pokaże nam wyniki na ekranie. Upewnij się, że wynik jest taki sam jak na rysunku.

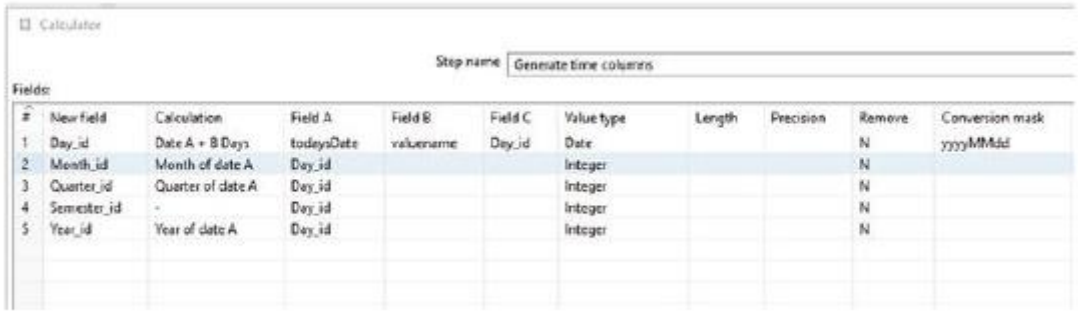


Jak widzimy, dodano nową kolumnę zawierającą wygenerowaną sekwencję oraz poprzednie dane. Możesz sobie wyobrazić, co zrobimy w następnym kroku, prawda? Sprawy zaczynają się już wyjaśniać. Czas nacisnąć przycisk Zatrzymaj, aby zatrzymać wykonanie naszej transformacji.

Uwaga: nigdy nie wahaj się korzystać z opcji podglądu wierszy tyle razy, ile chcesz. Konieczność modyfikacji kilku kroków, ponieważ przenosimy nieprawidłowe dane z poprzednich kroków, może być bardzo denerwująca. Upewnij się, że wynik każdego kroku jest tym, co próbujesz osiągnąć, często korzystając z opcji podglądu. Następnym krokiem, jak być może się zastanawiasz, jest oczywiście dodanie dwóch kolumn do nowej kolumny. To najtrudniejszy krok, ponieważ podsumowuje poprzednie i gdzie będziemy musieli upewnić się, że wszystko jest odpowiednio dostrojone, aby mieć to, czego potrzebujemy. Gdybyśmy zamiast daty wykonali już konwersję na liczbę, dodanie 31 do 20160101 wygenerowałoby niepoprawną datę jako 32 dzień stycznia, który nie istnieje, dlatego najpierw działamy, a potem konwertujemy.

Używanie kalkulatora do obliczania niektórych innych pól

Przejdźmy do interesującego kroku. Jesteśmy gotowi do użycia jednego z najpotężniejszych wbudowanych kroków w PDI. Krok kalkulatora. Poszukaj go w folderze Transform i przeciągnij i upuść na płótnie. Jako nazwę kroku wybraliśmy Generuj kolumny czasu, ale możesz użyć dowolnej nazwy. Tak jak w poprzednim kroku, upewnij się, że między drugim a trzecim krokiem zostało utworzone łącze. W tym momencie powinniśmy mieć łańcuch trzech kroków, połączonych jeden po drugim. Krok kalkulatora jest podobny do tego, co widzieliśmy do tej pory. Posiada siatkę, w której możemy zdefiniować pola. Pola te określają gdzie będą nasze kroki wyjściowe. Aby wygenerować kroki wyjściowe, możemy użyć wbudowanych funkcji i/lub pól z poprzedniego kroku lub zmiennych, które zobaczymy później. Zacznijmy tworzyć nowe pole wyjściowe. Ostateczny wygląd kalkulatora powinien wyglądać tak, jak na rysunku . Ale zobaczmy wyjaśnienie każdego pola.



W nowej kolumnie pola wpisz Day_id. Będzie to nazwa naszej pierwszej kolumny w danych wyjściowych tego kroku. Pod kolumną obliczeń kliknij rozwijane pole i poszukaj operacji o nazwie Data A + B dni, która pozwoli nam dodać liczbę do daty, którą już mamy. Aby to zadziałało, mamy teraz trzy kolumny, zwane Polem A, Polem B i Polem C. Pole A zawsze będzie polem wyjściowym lub wynikiem operacji. Pole A będzie pierwszym parametrem dla operacji, a pole B drugie. Należy zauważyć, że istnieją operatory, które działają tylko z jednym parametrem. W takim przypadku wystarczy użyć kolumny Pola B i pozostawić pustą kolumnę Pola C. Ponieważ nasza operacja to C= A+B, musimy wypełnić wszystkie trzy kolumny. W kolumnie Pole A wybieramy z rozwijanej listy TodaysDate. Tak nazwaliśmy wygenerowaną datę w pierwszym kroku. W polu B wybierzemy valuename, czyli sposób, w jaki nazwaliśmy sekwencję w naszym drugim kroku. W polu C wybieramy tę samą nazwę, którą nadaliśmy naszej nowej kolumnie Day_id, ponieważ chcemy w niej przechowywać wynik obliczeń. W polu Typ wartości wybierzemy datę, ponieważ nadal potrzebujemy formatu daty dla innych operacji na kolumnach. Upewnij się, że kolumna maski konwersji jest ustawiona na rrrrMMdd, ponieważ jest to nasz pożądany format. W ten sposób mamy prawie gotowe pierwsze pole tabeli (pierwsza kolumna). Ale jeśli pamiętasz, tabela docelowa miała pięć pól, więc wciąż mamy cztery do zrobienia. Na szczęście dla nas PDI ponownie udostępnia zestaw wbudowanych funkcji, które pozwolą nam łatwo obliczyć pozostałe pola. Month_id, Quarter_id, Semester_id i Year_id to pozostałe pola, które do tej pory

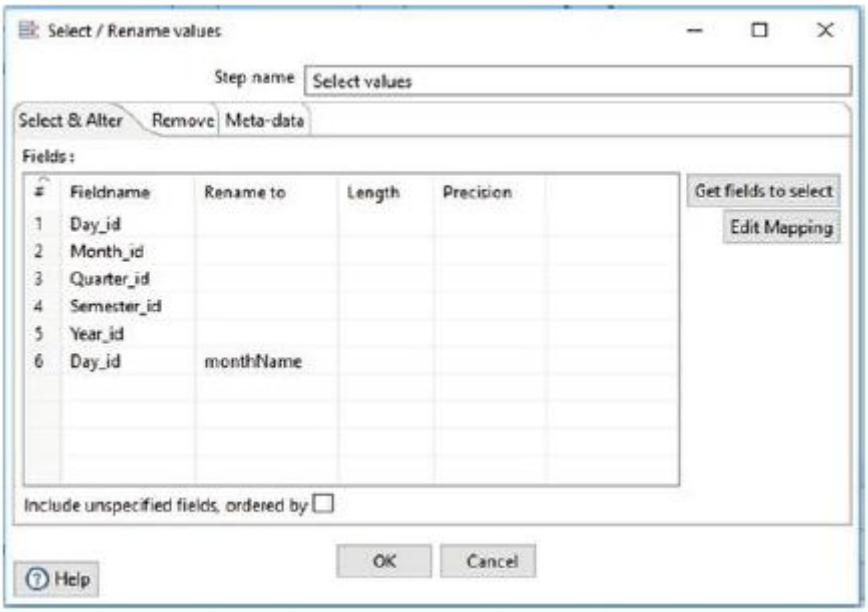
musimy dodać. Nie ma z tym problemu, ponieważ wystarczy podać obliczenia, a wynik jest bezpośredni. W przypadku id_miesiąca interesuje nas numer miesiąca w roku, więc styczeń będzie równy 1, luty będzie równy 2 i tak dalej. W kolumnie obliczeniowej musimy wybrać Month of Date A, a pole A będzie miało wartość Day_id, ponieważ obliczamy ją na podstawie naszej sekwencji dni. Typ wartości będzie liczbą całkowitą, ponieważ jest to oczywiste. Dla Quarter_id użyjemy tej samej logiki. Istnieje obliczenie o nazwie Quarter of Date A, a my również określimy Day_id w kolumnie fieldA. Ponownie typem danych jest liczba całkowita. Z Semester_id mamy problem, ponieważ Kettle niestety nie ma wbudowanej funkcji do obliczania semestru. Musimy więc znaleźć sposób na określenie semestru. Ponownie szkoda, że nie możemy zastosować funkcji MOD w naszej kolumnie Semester, ponieważ przy łatwej operacji bylibyśmy w stanie to obliczyć. Zatem tym razem nie możemy użyć kalkulatora do obliczenia tego pola, więc użyjemy myślnika, co oznacza, że pozostawimy pole puste. Rozwiążemy to później. Year_id ponownie ma łatwe rozwiązanie, ponieważ istnieje obliczenie o nazwie Year of Date A, a jego wynikiem jest liczba całkowita.

Obliczanie bardziej złożonych pól i usuwanie niepotrzebnych

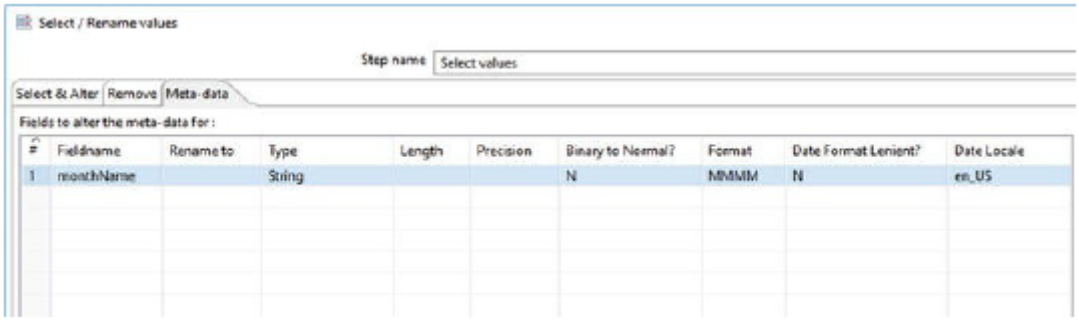
Ok, do tej pory obliczyliśmy prawie wszystko oprócz semestru. Ale jeśli pamiętasz z naszego modelu ERD, mieliśmy cztery harmonogramy. Nadszedł czas, aby przyjrzeć się kolumnom, które musimy wypełnić w innych tabelach. Wcześniej wyjaśniliśmy, że te trzy inne tabele zostaną wypełnione z dużej, którą budujemy, więc musimy mieć wszystkie potrzebne dane również dla tych tabel, zawarte w makrorozkładzie czasowym, który budujemy. Szybkie spojrzenie na diagram pokaże nam, że potrzebujemy opisu miesiąca: styczeń, luty…, a nie tylko liczby, opisu kwartału, który będzie Q1, Q2… i opisu roku. W przypadku roku zwykle nie ma sensu pisanie roku literami, więc w tym celu użyjemy dokładnie tej samej wartości, co numer roku: 2016, 2017 i tak dalej. Tych konwersji nie można wykonać na kroku kalkulatora, więc potrzebujemy czegoś innego. Tutaj przedstawiamy kolejny bardzo ważny krok w PDI. Ten krok jest nazywany wybieraniem wartości, ale zwykle robi znacznie więcej. Pozwala nam przekształcać nasze dane bez konieczności używania jakiegokolwiek fragmentu kodu. Znajdziesz go w grupie Transform. Zobaczmy, jak to działa. Krok Wybierz wartości ma trzy zakładki. Pierwszy, w którym możemy wybrać, które pola z poprzednich kroków chcemy propagować do wyjścia z tego kroku i czy chcemy zmienić nazwę pola, długość lub precyzję. Jest tutaj przydatny przycisk o nazwie Pobierz pola do wyboru, który załaduje wszystkie poprzednie pola do siatki. Następnie możemy usunąć lub dodać, co chcemy, ale zwykle naciśnięcie tego przycisku jest dobrym punktem wyjścia. Upewnij się, że poprzedni krok został już połączony z tym, ponieważ w przeciwnym razie przycisk nie będzie działał. Możemy ponownie użyć przycisku podglądu, aby upewnić się, że wszystko nadal działa. To właściwy czas, aby zobaczyć, jak upuścić pole i nie propagować go do wyjścia z kroku. Jeśli kliknęliśmy przycisk, zobaczysz, że pierwsze pole, którego użyliśmy do wygenerowania naszych dat, nasze sklonowane pole, które zawsze zawiera 20160101, które nazwaliśmy TodaysDate, wciąż podlega naszej transformacji. Ponieważ to pole było używane tylko jako podstawa do dodania numeru kolejnego, a mamy już utworzony dzień, nie jest nam już potrzebny. Tak więc pierwszą rzeczą do zrobienia w tym kroku jest usunięcie pola TodaysDate z siatki. Kliknij go prawym przyciskiem myszy i wybierz Usuń wybrane linie lub naciśnij przycisk Del na klawiaturze. Dokładnie to samo dotyczy pola valuename, które zawiera nasz kolejny numer. Możemy go również usunąć z siatki, ponieważ nie jest już potrzebny.

Uwaga: prawdopodobnie będziesz się zastanawiać, dlaczego nie używamy drugiej zakładki tego kroku o nazwie Usuń. To dobre pytanie i zależy od tego, co wolisz. Jeśli nie wybierzesz pola, to pole nie będzie propagowane, więc jest równoznaczne z jego usunięciem. Której opcji użyć, zależy od Ciebie.

Dobra, podsumujmy. Nadal brakuje nam opisu semestru, opisu kwartału i opisu miesiąca. Zacznijmy od najnowszego. Kolejna funkcja kroku Wybierz wartości znajduje się w trzeciej zakładce. Zakładka Metadane pozwala nam zmienić metadane pola. Metadane pola to wszystkie dane, które wyjaśniają, czym jest pole. W tym przypadku możemy również zmienić typ pola, z numeru na tekst i tak dalej. Tutaj zmienimy nasz dzień w formacie 20160101, aby używać tylko miesiąca, i zmienimy miesiąc z liczby na rzeczywistą nazwę miesiąca w prostym języku angielskim. Aby to osiągnąć, najpierw musimy wrócić do pierwszej zakładki i zduplikować pole Day_id, ponieważ będziemy operować na tym polu i nadal potrzebujemy reprezentacji 20160101, ponieważ potrzebujemy jej dla innej kolumny w bazie danych. Dodajmy więc szóstą nazwę pola w siatce, która będzie używać Day_id jako nazwy pola bazowego, ale zmieni nazwę na MonthName, ponieważ będzie to wynik przekształceń. Kolumny długości i precyzji pozostawiamy puste, ponieważ niczego tam nie zmieniamy. Sprawdź rysunek, aby upewnić się, że pasuje do tego, co tworzysz.



Następnie przechodzimy do trzeciej zakładki, zakładki Metadata i jako nazwy pola używamy nowo utworzonego pola MonthName. Typ tego pola będzie ciągiem, ponieważ planujemy przechowywać tekst, i tutaj pojawia się magia. Kolumna Format pozwala nam określić format wyprowadzanego tekstu. Jeśli użyjemy przycisku rozwijanego, nie zobaczymy go, ale istnieje specjalny format do konwersji części miesiąca daty na tekst. Ten format to MMMM (cztery m wielkimi literami), więc użyjmy tego formatu. Istnieje również inna kolumna, której możemy użyć, kolumna Date Locale, która użyje określonych tutaj ustawień regionalnych do wyświetlenia nazwy. Ponieważ chcemy używać angielskich nazw miesięcy, możemy użyć dowolnego angielskiego ustawienia regionalnego: wystarczy en_us, ale także en_GB i wiele więcej. Wynik powinien być taki sam, jak na rysunku 6.19. Następnie uruchom podgląd, aby upewnić się, że uzyskałeś oczekiwane wyniki.

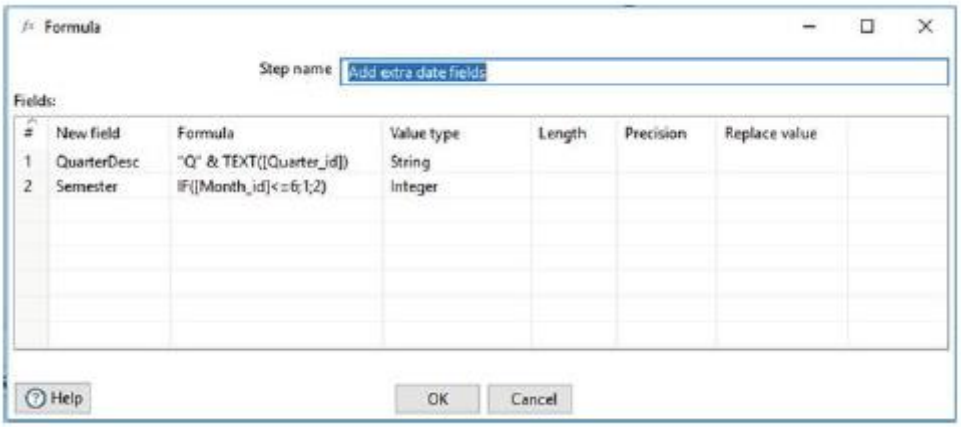


Na diagramie jest jeszcze jedno pole, o którym jeszcze nie mówiliśmy, ponieważ jest prawie takie samo jak to, które mamy. Jest to skrócona nazwa miesiąca. Na szczęście mamy do tego format MMM, więc musimy powtórzyć dokładnie te same kroki, aby uzyskać nazwę miesiąca, ale z tą różnicą, że tym razem nasza kolumna formatu będzie zawierała tylko trzy M. nazwiemy to pole MonthShort. Zostawiamy to jako ćwiczenie dla Ciebie.

Wykonywanie zaawansowanych manipulacji

Z tym dzielą nas już tylko dwa pola. Nazwa semestru i kwartału. Jak w życiu, istnieje wiele sposobów na wykonanie tego zadania. Tak czy inaczej wymaga od nas przedstawienia nowego kroku. Ponieważ oba kroki są uważane za ważne i powinieneś o nich wiedzieć, przedstawię dwa możliwe sposoby ich wykonania. Pierwszy uprawnia do użycia kroku Formula, który można znaleźć w kroku Scripting, a drugi wykorzystuje prawdopodobnie jeden z najpotężniejszych kroków w Kettle, ale także najtrudniejszy do opanowania, czyli zmodyfikowaną wartość JavaScript, którą można również znaleźć w grupie Skrypty. Skupmy się najpierw na kroku formuły, ponieważ jest on łatwiejszy do zrozumienia. Krok formuły Nadszedł czas, aby przeciągnąć i upuścić krok formuły do płótna i utworzyć łącze między ostatnim krokiem w łańcuchu, którym były wybrane wartości, a nowym usuniętym krokiem. W tym kroku mamy dwa zadania. Jednym z nich jest utworzenie opisu kwartału, który będzie wyglądał jak Q1, Q2 itd., a drugi polega na obliczeniu semestru. Skupmy się na pierwszym. Kliknij dwukrotnie nowo dodany krok formuły. Najpierw nazwij krok. Użyliśmy nazwy „Dodaj dodatkowe pola daty". Ponieważ tworzymy Opis kwartału, w nowym polu umieścimy QuarterDesc jako nazwę. W kroku formuły możemy kliknąć dwukrotnie, a pojawi się wyskakujące okienko. W górnej części ekranu możemy wpisać naszą formułę, natomiast w lewej mamy listę dostępnych nam formuł i przekształceń. Jeśli klikniemy na jedną, w dolnej części okna dialogowego zostanie wyświetlona strona pomocy ze składnią i przykładami dla każdej formuły. Trzeba przyznać, że ten krok nie jest łatwy w użyciu, więc przy odrobinie praktyki możesz napisać formułę bezpośrednio w górnym polu. Użyjemy dwóch funkcji: jedna to formuła konkat, która jest wykonywana za pomocą ampersand & słowo kluczowe; a następnie inny, aby przekonwertować liczbę całkowitą, która jest liczbą porządkową ćwiartki, na tekst, ponieważ funkcja konkat działa z dwoma polami tekstowymi. Należy zauważyć, że do pól, które pochodzą z poprzednich kroków, nie można bezpośrednio odwoływać się za pomocą ich nazw, dlatego musimy je dodać w nawiasach kwadratowych []. Ponadto każdy tekst, który chcemy połączyć, musi być zdefiniowany między podwójnymi cudzysłowami. Wynikowa formuła będzie następująca: „Q" & TEXT([Quarter_id]). Jeśli przejrzymy krok, powinniśmy zobaczyć, że mamy już gotowy opis kwartału. Czas przejść do pozostałego pola, jakim jest Semestr. Istnieje wiele sposobów obliczania semestru. Jednym z nich jest funkcja warunkowa, taka jak instrukcja IF lub CASE. Jeśli wartość miesiąca mieści się w przedziale od 1 do 6, to semestr wynosi 1, w przeciwnym razie 2. Inny próbuje dokonać dzielenia liczby całkowitej numeru miesiąca z przedziału 7 i dodać jeden do wyniku. Ponieważ nie jest to najłatwiejsze i

najprostszym podejściem, użyjmy instrukcji warunkowej, aby to obliczyć. Nazwiemy nową dziedzinę Semestr. Aby to obliczyć, użyjemy prostej logiki. Jeśli miesiąc jest mniejszy lub równy 6, to semestr wynosi 1, w przeciwnym razie 2. Formuła powinna wyglądać następująco: IF ([Month_id] <=6; 1; 2). Sprawdź, czy pasuje to do rysunku .

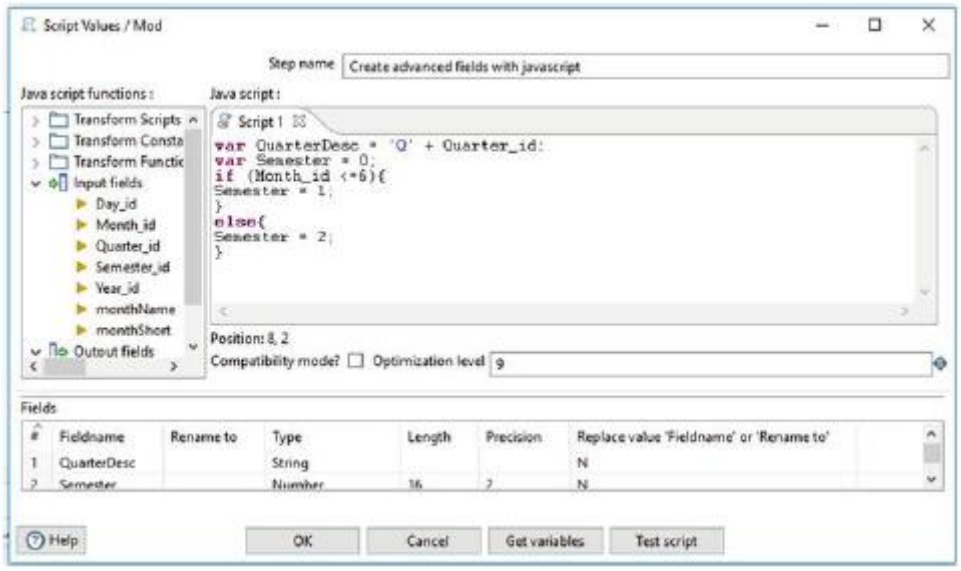


Krok Zmodyfikowana wartość JavaScript

Jest to jeden z najpotężniejszych kroków dostępnych w PDI. Zmodyfikowany krok wartości JavaScript pozwala nam używać kodu JavaScript do wykonywania dowolnych operacji, tworzenia nowych pól i łączenia pól. Oczywiście wymaga to znajomości języka JavaScript, ponieważ musimy używać tego języka, aby wykonać ten krok. Ponieważ celem tej książki nie jest nauczanie języka JavaScript, przedstawimy tylko obliczenia, które wykonaliśmy w tym kroku, aby osiągnąć te same wyniki, co w kroku Formuła. Jeśli przeciągniesz i upuścisz krok i połączysz go w kanwie, możesz zacząć. Kliknij go dwukrotnie, a zobaczysz wpis tekstowy w kroku. Po prostu skopiuj ten kod, a następnie kliknij przycisk na dole o nazwie Pobierz zmienne, który utworzy kroki wyjściowe dla dwóch zdefiniowanych pól.

```
var QuarterDesc = 'Q' + Quarter_id;

var Semester = 0;

if (Month_id <=6){

Semester = 1;

} else{

Semester = 2;

}
```

Jeśli wszystko jest w porządku, powinieneś zobaczyć to samo, co na rysunku a podgląd powinien dać takie same wyniki, jak w kroku kalkulatora.



Naprawa kilku rzeczy

Jeśli przejrzymy naszą transformację, zauważymy, że mamy powtarzające się wartości dla naszych miesięcy i kwartałów. Jak odróżnić Q1 od 2016 i Q1 od 2017? Tak jak jest teraz, nie jest to możliwe bez patrzenia na inne pola w tym samym rzędzie. To samo dzieje się z miesiącem. Będziemy mieli problem z wstawieniem tych informacji, ponieważ nasze tabele wymiarów muszą zawierać unikalne wartości dla każdego rekordu. Zwykle wymuszamy to za pomocą klucza podstawowego, ale czasami w hurtowni danych nie jest to używane i to proces ETL zawiera logikę pozwalającą uniknąć takich sytuacji. Chcemy również uwzględnić w naszych opisach miesięcy rok, aby był bardziej przejrzysty. Jest wiele miejsc, w których można to naprawić. Możemy to naprawić, gdy zbudujemy ETL, który odczytuje dane z obszaru pomostowego i zapisuje je do hurtowni danych, używając wyrażenia SQL do manipulowania polami; możemy wrócić do poprzednich kroków i tam go rozwiązać; lub możemy dodać dodatkowy krok i rozwiązać go od razu. W zależności od przypadku jedno rozwiązanie może być lepsze od drugiego, ale czasami nie ma to znaczenia. Aby nie komplikować sprawy, poprawmy poprzednie kroki.

1. Wróć do kroku wyboru wartości i zmień formaty dwóch miesięcy MMMM na MMMM rrrr i MMM na MMM rrrr. Zwróć uwagę na spację między M i y.

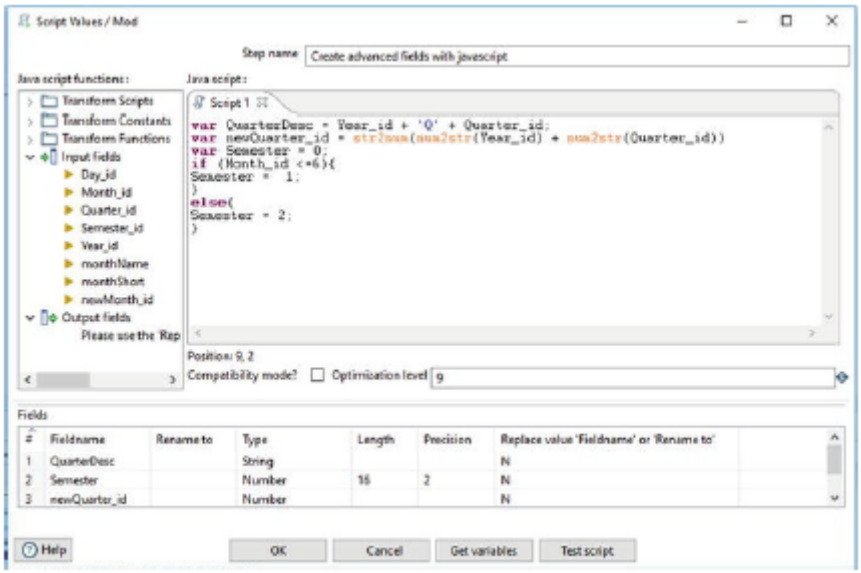
2. Aby dołączyć rok do miesiąca, albo ponownie używamy nowej formuły lub kroku JavaScript na końcu, albo dokonujemy transformacji metadanych za pomocą kroku Wybierz wartości. My zdecydujemy się na to drugie. W naszym kroku wybierania wartości przechodzimy do zakładki Select & Alter i tworzymy nowy wiersz z nazwą pola Day_id i zmieniamy nazwę na newMonth_id. Następnie przechodzimy do zakładki metadane i dodajemy nowy wiersz o nazwie pola newMonth_id typu string i formacie rrrrMM. Spowoduje to utworzenie nowego formatu miesiąca typu RRRRMM, który jest potrzebny do hurtowni danych, jak zobaczymy później.

3. Ustalenie kwartału jest nieco bardziej skomplikowane, ponieważ nie ma dla niego wzoru daty. Ale możemy to zrobić w naszym kroku JavaScript lub formuły.

a. W kroku formuły najpierw zmodyfikuj opis kwartału, dodając rok na początku. Formuła powinna teraz brzmieć:

TEKST([id_roku]) & "Q" & TEKST([id_kwartalu]). Dzięki temu poprawilibyśmy opis, ale jeszcze nie identyfikator. Utwórz nowy wiersz o nazwie newQuarter_id z następującą formułą: TEKST([Year_id]) & TEKST([Quarter_id]).

b. W przypadku kroku JavaScript musimy wykonać podobną poprawkę. Kod JavaScript zostanie zmieniony na następujący i należy dodać nowy wpis w dolnej siatce, aby propagować newQuarter_id, jak na rysunku




```
var QuarterDesc = Year_id + 'Q' + Quarter_id;

var newQuarter_id = str2num(num2str(Year_id) +

num2str(Quarter_id))

var Semester = 0;

if (Month_id <=6){

Semester = 1;

} else{

Semester = 2;

}
```

Semestr nie wymaga żadnej poprawki, ponieważ będziemy przechowywać 1 lub 0, ponieważ nie mamy do tego tabeli wymiarów. Jeśli chcesz uwzględnić również rok, pozostaw to jako ćwiczenie.

Ostatni krok

Teraz, gdy mamy wszystko, czego potrzebujemy, nadszedł czas, aby wygenerować ostatni krok. Cóż, w rzeczywistości dwa ostatnie kroki, ponieważ ten ostatni będzie krokiem wyjściowym bazy danych do przechowywania wyniku tego w naszym schemacie pomostowym. Najpierw musimy przeciągnąć i upuścić kolejny krok wyboru wartości w naszym kanwie. Połączymy go z obecnym łańcuchem kroków i wykorzystamy do podwójnego celu. Po pierwsze, aby wybrać wszystkie pola, które ostatecznie chcemy przechowywać i odrzucić te, które nie są interesujące. Po drugie, możemy również użyć zakładki metadane, aby zmienić typ danych kolumny, jeśli zajdzie taka potrzeba. Możemy również

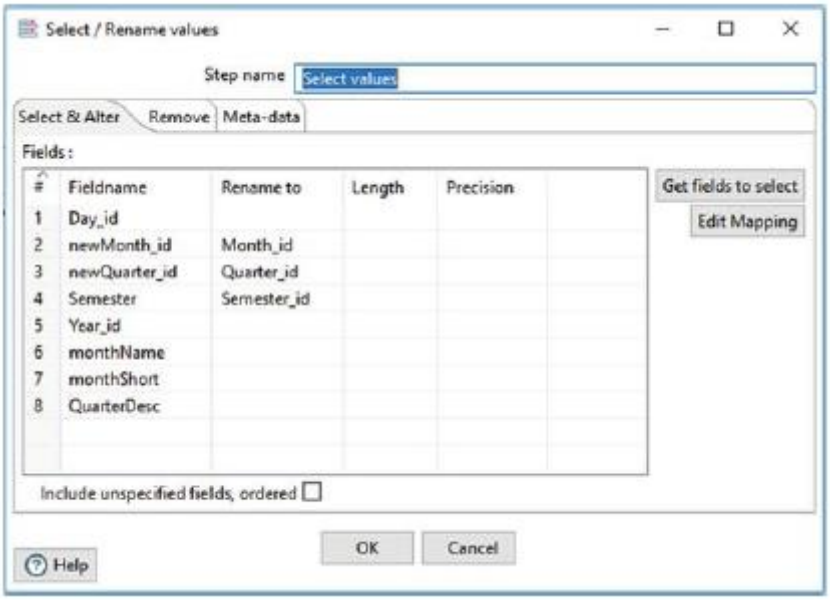
pozwolić, aby ostatni krok, a nawet niejawna konwersja bazy danych, zmieniły typ danych, ale dobrą praktyką jest robienie tego bezpośrednio na ETL, ponieważ inne mogą mieć nieoczekiwane wyniki. Na tym etapie nie musimy propagować kilku pól. Są to Month_id, Quarter_id i Semester_id, ponieważ nie mają pożądanej przez nas formy lub w ogóle nie zawierają danych. Więc albo nie uwzględniamy ich na liście, albo dodajemy je na karcie Usuń. Następnie musimy zmienić nazwy trzech kolumn, aby zastąpić trzy, których nie propagowaliśmy do danych wyjściowych. Zastosuj następujące zmiany nazw:

* Nazwa pola newMonth_id zostanie zmieniona na Month_id

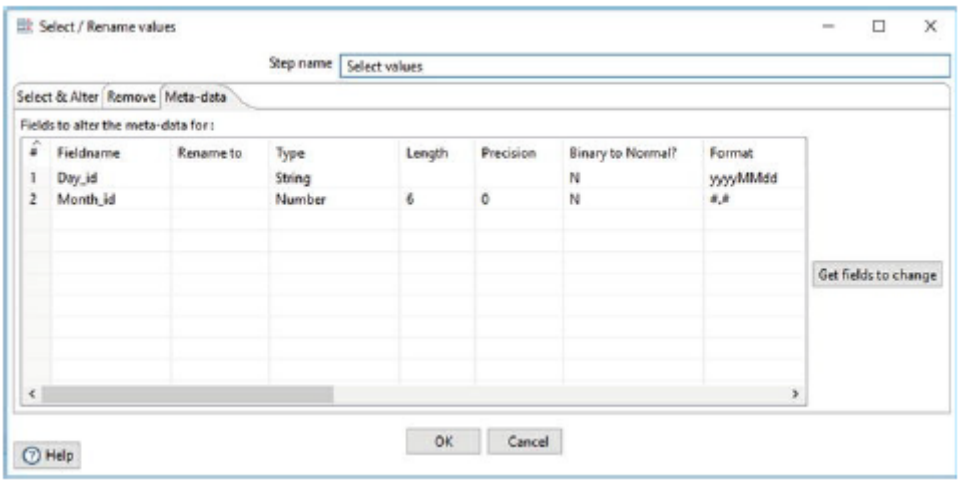
* Nazwa pola newQuarter_id zostanie zmieniona na Quarter_id

* Nazwa pola Semester zostanie zmieniona na Semester_id

Poza tym będziemy też porządkować pola. Nie jest to tak naprawdę obowiązkowe, ale wolimy to zrobić dla jasności. Końcowy stan tego kroku powinien wyglądać tak, jak pokazano na rysunku



Pozostało tylko zmienić teraz dwa pola, więc zapisujemy jedno jako String, chociaż powinno to być liczba całkowita, ale później zrobimy konwersję, czyli Day_id, i kolejne, nowe Month_id, które będzie INT zamiast ciągu znaków. Można je zobaczyć na rysunku



Teraz nadszedł wreszcie moment, aby stworzyć nasz ostatni krok. W tym celu przechodzimy do folderu Wyjście w widoku projektu i przeciągamy i upuszczamy krok wyjściowy tabeli w obszarze roboczym, a następnie łączymy go. W tym kroku musimy skonfigurować kilka rzeczy: odpowiednio nazwać krok; wybranie połączenia do zdefiniowanego przez nas wcześniej obszaru postojowego; wybierając schemat, który w przypadku MySQL/MariaDB jest bazą danych, którą stworzyliśmy, czyli naszą pomostową bazą danych, jako docelową nazwę tabeli ustaw stg_time i zaznacz obciętą tabelę, jak powiedzieliśmy wcześniej, tę tabelę załadujemy tylko raz. Jeśli z jakiegokolwiek powodu chcemy załadować go później, zaznaczając to pole, upewnimy się, że wszystkie poprzednie dane zostaną odrzucone przed dodaniem nowego. Pozostałe opcje możemy pozostawić bez zmian. Jeśli spojrzysz na dolną część okna dialogowego, zobaczysz przycisk o nazwie SQL. Jeśli naciśniesz ten przycisk, zostanie wyświetlona sugerowana instrukcja tworzenia tabeli, a my możemy zdecydować o uruchomieniu jej w naszej pomostowej bazie danych. Jeśli spojrzymy na pola, zobaczymy, że PDI próbowało zdecydować, które typy danych są zdefiniowane dla każdego pola. Czasami nie jest to poprawne lub nieoptymalne, jak tym razem. Ale ponieważ w obszarze przejściowym nie przejmujemy się tym zbytnio, akceptujemy je i naciskamy przycisk wykonania. Tabela zostanie utworzona bez zmian, ale dane nadal nie zostaną załadowane. Wynik powinien wyglądać mniej więcej tak:

SQL executed: CREATE TABLE test.stg_time

(

Day_id TINYTEXT

, Month_id INT

, Quarter_id TINYTEXT

, Semester_id INT

, Year_id INT

, monthName TINYTEXT

, monthShort TINYTEXT

, QuarterDesc TINYTEXT

)

1 SQL statements executed

Możemy teraz zamknąć okno dialogowe, ponieważ wszystko jest w porządku. Następnie wracając do naszej transformacji, nadszedł czas, aby ją uruchomić i załadować dane do obszaru pomostowego. Przejdź do menu akcji i kliknij polecenie Uruchom lub naciśnij klawisz F9. Transformacja rozpocznie się i pojawi się okno dialogowe Uruchom. Pozostaw wszystko jako domyślne i kliknij przycisk Uruchom. Po kilku sekundach transformacja powinna zakończyć się pomyślnie. Na dolnym ekranie PDI poszukaj zakładki Step Metrics i upewnij się, że mamy wygenerowane 3653 wierszy. Możemy teraz przejść do naszej tymczasowej bazy danych i zapytać nową tabelę, aby upewnić się, że dane już tam są:

SELECT *

FROM stg_time

Rezultatem są 3653 rekordy w tabeli. Więc wszystko ok. Po sprawdzeniu ostatecznego wyniku wszystko wygląda w porządku, jak widać na rysunku

Examine preview data

Rows of step: Select values (1000 rows)

#	Day_id	Month_id	Quarter_id	Semester_id	Year_id	monthName	monthShort	QuarterDesc
1	20160101	201601	20161	1	2016	January 2016	Jan 2016	2016Q1
2	20160102	201601	20161	1	2016	January 2016	Jan 2016	2016Q1
3	20160103	201601	20161	1	2016	January 2016	Jan 2016	2016Q1
4	20160104	201601	20161	1	2016	January 2016	Jan 2016	2016Q1
5	20160105	201601	20161	1	2016	January 2016	Jan 2016	2016Q1

Łączenie wszystkiego

Ostatni krok w naszym procesie ETL jest najbardziej złożony. Nie tylko dlatego, że musimy stworzyć złożoną transformację, ale dlatego, że musimy wybrać odpowiednie dane i wykonać różnorodne połączenia między tabelami, aby uzyskać dokładnie te dane, których potrzebujemy, w wymaganym formacie i umieścić je we właściwych tabelach naszej hurtowni danych. W tym celu stworzymy nową transformację. W PID przejdź do Plik ➤ Nowy ➤ Transformacja. Na naszym płótnie pojawi się pusta transformacja.

Tabele czasu

Czas rozpocząć naszą ostateczną transformację. Zaczniemy zapełniać terminarze. Mieliśmy tylko jedną tabelę w obszarze przejściowym, aby wypełnić cztery tabele, które mamy w naszym magazynie danych, koncepcjami związanymi z czasem; musimy wyprowadzić dane z tej tabeli pomostowej. Zaczniemy wypełniać tabelę t_l_year. Tabela t_l_year zawiera tylko rok i jego opis. Jeśli pamiętasz, kiedy omawialiśmy wymiar czasu, powiedzieliśmy, że przechowywanie opisu przez rok nie ma większego sensu, chyba że chcesz przechowywać rok w trybie tekstowym. To nie jest coś, co nas interesuje w tym momencie, więc w tym momencie użyjemy numeru roku w obu polach. Tak, może to wyglądać trochę głupio, ale dzięki temu zachowujemy ten sam schemat dla wszystkich tabel. Aby rozpocząć, po prostu przeciągnij i upuść krok wprowadzania tabeli na obszar roboczy. Wybierz pomostowe połączenie z bazą danych i zapisz to zapytanie:

```
select distinct

Year_id Year_id, Year_id Year_desc

from staging.stg_time;
```

Wybieramy wszystkie różne lata z tabeli i używamy numeru roku w obu polach. Dzięki klauzuli odrębnej upewniamy się, że nie naruszamy żadnego klucza podstawowego w polu year_id ani nie wprowadzamy duplikatów, które mogą później powodować iloczyny kartezjańskie na etapie raportowania. Wszystko idzie dobrze; jeśli przejrzymy dane wyjściowe, powinniśmy zobaczyć 10 wierszy z 10 latami 2016-2025 na ekranie, zduplikowanymi w dwóch kolumnach o różnych nazwach. Więc to jest to! Skończyliśmy na pierwszym stole, łatwe, prawda? Wiemy, że musimy tylko połączyć to z wyjściem tabeli i wybrać naszą bazę danych dwh jako połączenie i wybrać naszą tabelę t_l_year jako miejsce docelowe.

Notatka

Jeśli określiliśmy ograniczenie w naszym narzędziu do modelowania danych dla tabel, w tym przypadku nie możemy użyć tabeli wyjściowej z zaznaczoną opcją obcinania, ponieważ silnik bazy danych będzie narzekał, ponieważ niektóre inne tabele używają tabeli t_l_year jako tabeli głównej. Otrzymamy komunikat o błędzie podobny do tego: „Nie można obciąć tabeli, do której odwołuje się ograniczenie klucza obcego (`dwh`.`t_r_time`, CONSTRAINT `R_11` FOREIGN KEY (`Year_id`) REFERENCES

`dwh`.`t_l_year` (`Id_roku`))". W takim przypadku musimy użyć innego kroku, zwanego Insert/Update lub znanego również jako Upsert lub Merge, który omówimy dla innych części transformacji.

Zauważ, że czasami, jeśli nie określimy nazw pól w instrukcji tabeli wejściowej lub użyjemy takich, które nie pasują do końcowych nazw kolumn w naszej hurtowni danych, mamy możliwość sprawdzenia określonych pól bazy danych w naszej tabeli wyjściowej. Umożliwi nam to użycie w tym samym kroku zakładki o nazwie Pola bazy danych, w której możemy zmapować źródła pochodzenia z nazwami kolumn docelowych. Pomocne może być użycie przycisku Pobierz pola i wtedy pozostaje już tylko kwestia stworzenia relacji między przychodzącymi i wychodzącymi polami lub nazwami kolumn. Następna tabela może być ćwiartką. W tym przypadku używamy tego samego pomysłu, ale używając pól QuarterDesc i QuarterId z naszej tabeli stg_time. Następnie robimy to samo z naszą tabelą t_l_month_table. Ten t_r_time jest nieco trudniejszy. Jeśli pamiętacie, powiedzieliśmy, że nadal nie mamy formatu Day in English w naszej tabeli inscenizacyjnej. Czas więc go wygenerować. Możesz także generować wartości z danych w inscenizacji, stosując wbudowane funkcje w bazie danych lub stosując operacje SQL na danych, które już mamy, aby wygenerować nowe dane. W tym przypadku stosujemy wbudowaną funkcję wyodrębniania nazwy dnia w formacie angielskim z daty zapisanej w formacie daty. Korzystając z pola Day_id, możemy wygenerować naszą pełną nazwę dnia. Po prostu użyj następującego fragmentu kodu jako danych wejściowych tabeli dla naszej tabeli t_r_time:

```
select distinct

Day_id, Semester_id, DATE_FORMAT(Day_id,'%W, %D of

%M %Y') Day_Desc, Month_Id, Quarter_id, Year_id

from staging.stg_time;
```

Upewnij się, że tym razem wybrałeś określone pola bazy danych w kroku tabeli wyjściowej i odpowiednio zmapuj kolumny między danymi wejściowymi i wyjściowymi. I mamy teraz idealną tabelę z wypełnionymi wszystkimi kolumnami.

Tabele produktów

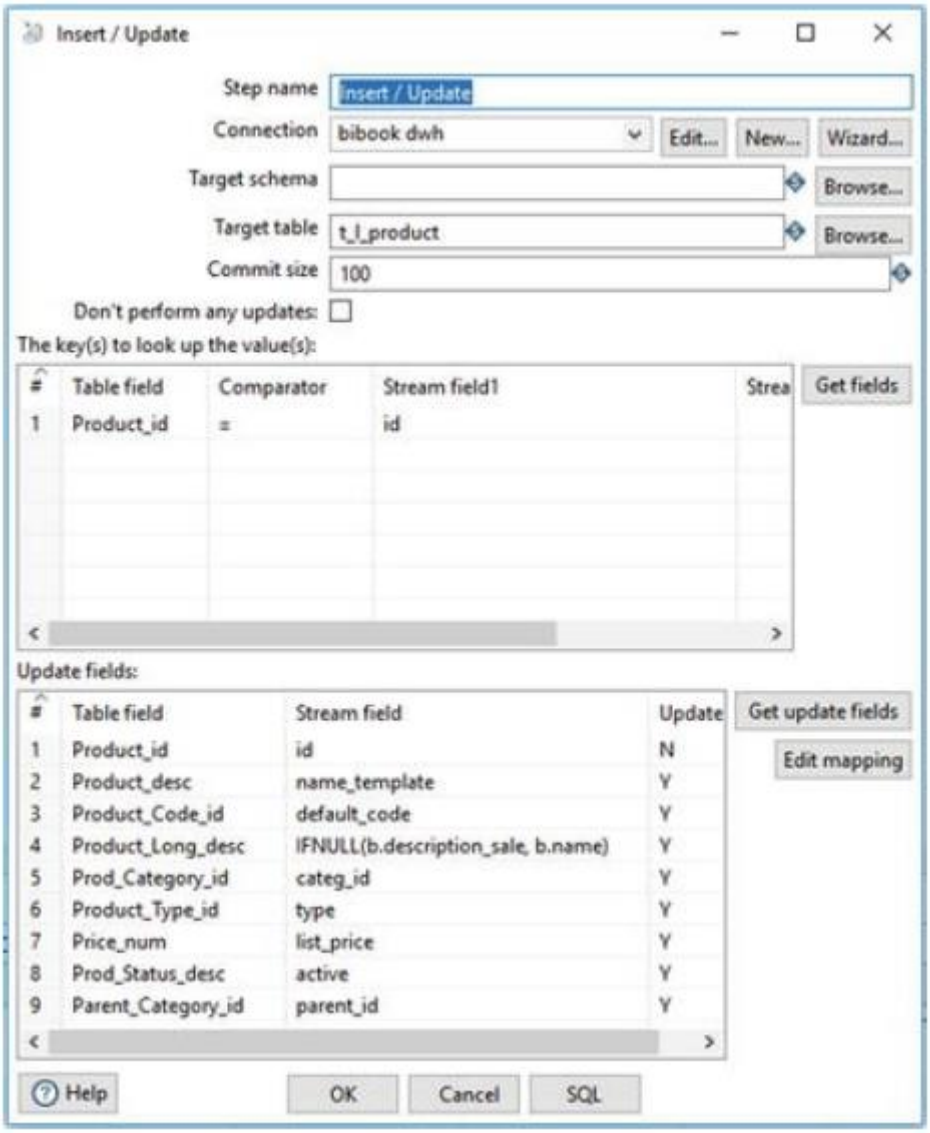
Kolejnym zestawem tabel, które możemy wypełnić, są tabele produktów. Składają się one z t_l_product, t_l_category i t_l_parent_category. t_l_category to bezpośredni wybór z kategorią stg_product_category z inscenizacji. Musimy tylko wybrać potrzebne kolumny: id i name. To samo dotyczy tabeli kategorii t_l_parent. Można się zastanawiać, czy dla tabeli t_l_product możemy zrobić to samo. Poczekaj! To nie wszystko. Pomyśl, co mogłoby się stać, gdybyśmy zaprzestali sprzedaży produktu i usunęli go z naszego serwisu transakcyjnego. Jeśli wybierzemy strategię obcinania + wstawiania, ten produkt, ponieważ nie istnieje już w naszym transakcyjnym, zostanie utracony. Ok, w przypadku przyszłej sprzedaży nie stanowi to problemu, ponieważ już go nie sprzedajemy, ale co stanie się z już załadowanymi informacjami o sprzedaży? Zapewne już znasz odpowiedź: będzie nam brakować tych wyprzedaży, a przynajmniej ines, które się odnoszą

do tego konkretnego produktu, kiedy krzyżujemy tabelę z tabelą produktów. To jest coś, czego musimy unikać. Najpierw skup się na tabeli wejściowej. Musimy połączyć trzy tabele, aby pobrać wszystkie potrzebne dane. Instrukcja join, którą musimy zdefiniować w naszej tabeli wejściowej, jest następująca:

```
SELECT

a.id,
```

```
a.name_template,
a.default_code,
IFNULL(b.description_sale, b.name),
b.categ_id,
b.type,
b.list_price,
b.active,
c.parent_id
FROM staging.stg_product_product a,
staging.stg_product_template b,
staging.stg_product_category c
where a.product_tmpl_id=b.id
and b.categ_id=c.id;
```

Następnie pozostaje tylko kwestia podłączenia go do kroku docelowego. Jeśli przeczytałeś poprzednią notatkę, będziesz świadomy możliwego rozwiązania. Zamiast obcinać i wstawiać wszystkie rejestry od zera, możemy zastosować kolejny krok o nazwie Update/Insert, znany również jako Merge lub Upsert dla tych, którzy mają większe doświadczenie z relacyjnymi bazami danych. Konfiguracja tego kroku jest nieco bardziej skomplikowana, ale nie ma rzeczy niemożliwych. Pierwsze pola są podobne do kroku tabeli wyjściowej. Po prostu nadaj nazwę krokowi, wybierz połączenie dwh, wpisz nazwę tabeli, która ma być używana jako cel: t_l_product, a następnie zwróć uwagę na pierwszą zakładkę o nazwie klucz(e), aby wyszukać wartości. Wyjaśnijmy, co to oznacza, najpierw rozumiejąc ten krok. Na czym polega ten krok, dla każdego wiersza wejściowego próbuje znaleźć pasujący wiersz w tabeli docelowej. Kolumny, które są porównywane, są zdefiniowane w tej pierwszej tabeli. Ponieważ pracujemy głównie według identyfikatorów, powinniśmy tutaj porównać identyfikator produktu z tabeli pomostowej z identyfikatorem produktu w tabeli dwh. Jeśli się zgadzają, oznacza to, że produkt istnieje, więc mamy dwie opcje: nic nie robić, ponieważ produkt już mamy i nie musimy go dodawać; lub zaktualizuj dowolną wartość: w naszym przypadku przechowujemy tylko opis produktu. Równie dobrze może być nazwa produktu uległa zmianie lub wystąpiła literówka, dlatego w takich przypadkach możemy chcieć ją zaktualizować. Pola do aktualizacji należy określić w drugiej tabeli kroku, wzdłuż której będzie znajdować się kolumna wejściowa, która będzie używana do jej aktualizacji. Jeśli porównanie się nie powiedzie, ponieważ wiersz nie działa, wówczas krok wstawi brakujący rekord do naszej tabeli końcowej. Ponieważ niczego nie usuwamy ani nie obcinamy, nie ma ryzyka utraty informacji, które zostały już wyczyszczone z naszego systemu operacyjnego, a mamy system, który będzie na bieżąco dodawać nowe produkty, które się pojawią. Jeśli brzmi to nieco myląco, spójrzmy na rysunek 6.29, aby zobaczyć przykład, jak należy zaimplementować upsert tabeli kategorii produktów.



Podobnie jak w przypadku porad dotyczących wydajności, w sytuacjach, w których występuje wiele wyszukiwań, opłacalne może być zdefiniowanie indeksu w kluczu id w tabeli hurtowni danych, dzięki czemu wyszukiwania są wykonywane znacznie szybciej. Jeśli już zdefiniowaliśmy tę kolumnę jako klucz podstawowy, nie musimy się martwić, ponieważ baza danych podjęła już działania, aby upewnić się, że indeks wymusza klucz podstawowy.

Notatka

Istnieją inne sposoby osiągnięcia tego samego, ale są one trudne do zrozumienia dla początkujących. W grupie działań Datawarehouse znajdziesz krok o nazwie Dimension Lookup Update, który jest bardziej wydajny niż krok, którego użyliśmy, i to samo można wykonać w podobny sposób.

Tabele pracowników i klientów

Po ukończeniu produktu mamy jeszcze dwa zestawy tabel do wypełnienia: są to tabele dotyczące pracowników i klientów. Są nieco podobne i stosują tę samą strategię, co te ujawnione wcześniej. Zachęcamy do spróbowania ich wypełnienia poprzez zdefiniowanie kroków jako ćwiczenia. Oto kilka wskazówek.

Wypełnij tabele pracowników

W odniesieniu do pracowników mamy do wypełnienia trzy tabele: t_L_employee, która będzie zawierała dane dotyczące pracowników; t_l_employee_department, który zawiera dane o działach w firmie; oraz t_l_emp_level, który zawiera dane o możliwym opisie stanowiska każdego pracownika. Te dwa ostatnie mogą być pełne za pomocą obcinania/wstawiania. Możesz również użyć kroku Wstaw aktualizację, jeśli uważasz, że niektóre działy lub stanowiska mogą od czasu do czasu zniknąć. Są to prawdopodobnie małe tabele, więc nie powinno być problemu, i są to bezpośrednie zapytania dotyczące stg_hr_departments i stg_hr_jobs. Upewnij się, że wybrałeś odrębny. W przypadku pierwszego musimy wykonać obciążenie przyrostowe. To, co musimy zrobić, jest bardzo podobne do tego, co zrobiliśmy dla tabeli produktów. Stół jest bezpośrednim wyborem z inscenizacji; i nie musimy dokonywać żadnych transformacji ani obliczać niczego nowego, wystarczy wybrać wymagane pola.

Wypełnij tabele klientów

Jeśli chodzi o tabele klientów, proces jest taki sam jak w przypadku tabeli pracowników. Potrzebujemy tylko dwóch stolików z inscenizacji. Są to: stg_res_partner i stg_res_country. Procedura jest również taka sama, jak omówiliśmy w przypadku poprzednich, z pełnym załadowaniem t_l_cus_country, ponieważ nie oczekujemy modyfikacji listy krajów (choć może tak być!) wstawić

Tabela sprzedaży: t_f_sales

Ostatnią tabelą do załadowania jest tabela faktów. Ta tabela będzie zawierać całą sprzedaż firmy na poziomie linii produktów. Oznacza to, że musimy zebrać przedmioty, jak widzieliśmy wcześniej z siedmiu różnych stołów. Chociaż możemy ładować siedem tabel z inscenizacji jedna po drugiej i wykonywać łączenia w PDI, nie ma to większego sensu, jeśli możemy wykorzystać posiadane umiejętności SQL. Ponownie przeciągniemy i upuścimy krok Tabeli wejściowej na kanwę i wprowadzimy następujące zapytanie, które łączy siedem tabel i zbiera dane wymagane do wypełnienia tabeli faktów:

```
select a.id Order_id, b.id Order_Item_id, d.id

Invoice_id, d.invoice_line_id Invoice_Item_id,

IFNULL(d.product_id,b.product_id) Product_id,

IF(d.id is

NULL,IF(b.qty_to_invoice<>0,'Ordered','Quoted'),'Invoi

ced') Status_id, a.name Order_desc, d.quantity

Quantity_inv_num, d.price_subtotal Amount_inv_num,

d.line_taxes TAX_inv_num,

d.price_subtotal+IFNULL(d.line_taxes,0)

Total_amount_inv_num, DATE(IF(d.id is

NULL,IF(b.qty_to_invoice<>0,a.date_order,a.create_date

),d.date)) Day_id,

DATE(a.date_order) Order_Date_id, DATE(d.date)

Invoice_date_id, b.qty_delivered Quantity_ord_num,
```

```
b.qty_delivered Quantity_del_num, b.product_uom_qty
Quantity_quot_num,
b.currency_id Currency_id, DATE(a.create_date)
Creation_Date_id, a.partner_id Customer_id, a.user_id
Employee_id
from stg_sale_order a
join
stg_sale_order_line b
on (a.id=b.order_id)
left outer join
(select a.id, a.date, b.id invoice_line_id,
price_unit, price_subtotal, discount, product_id,
quantity , c.order_line_id,
e.amount*b.price_subtotal/100 line_taxes
from
stg_account_invoice a
join stg_account_invoice_line b on
(a.id=b.invoice_id)
join stg_sale_order_line_invoice_rel c on
(c.invoice_line_id=b.id)
left join stg_account_invoice_line_tax d on
(b.id=d.invoice_line_id)
left join stg_account_tax e on (d.tax_id=e.id)
)d
on (b.id=d.order_line_id )
```

Następnie dodajemy krok Wstaw/Aktualizuj jako krok docelowy i wypełniamy kluczowe pola, jak pokazano na rysunku 6.30, a także pozostałe pola, jak pokazano na tym samym obrazku.

Na początku może to wyglądać trochę onieśmielająco, ale jest to nic innego jak zastosowanie tego, czego nauczyliśmy się do tej pory, z kilkoma podstawowymi transformacjami, które widzieliśmy w rozdziale SQL. Zakończyliśmy teraz wszystkie transformacje ładowania danych.

Projektowanie pracy

Kiedy mamy już gotowe dwie transformacje, wystarczy zdefiniować zadanie. Poinstruujemy Kettle, aby wykonał pierwszą transformację, aw przypadku błędu wyśle wiadomość e-mail na określoną skrzynkę pocztową, która może być nasza. Jeśli wykonanie pierwszej transformacji zakończyło się OK, nastąpi przejście do drugiej. Ten jest najnowszym, jaki zaprojektowaliśmy i skopiuje dane z przemieszczania do tabel magazynu danych. Ponownie potrzebujemy kroku po wykonaniu transformacji na wypadek, gdyby coś poszło nie tak. Jeśli wszystko zakończyło się OK, możemy również wysłać wiadomość e-mail, aby powiadomić ludzi, że ładowanie zostało zakończone, a dane są już dostępne do analizy. Prawdą jest, że projektowanie przepływu pracy (lub zadania) jest częścią procesu ETL. Postanowiliśmy jednak dodać do tej książki kolejny rozdział, zatytułowany „Planowanie procesów BI: jak organizować i aktualizować uruchomione procesy”. W tej części zobaczymy, jak utworzyć zadanie, zaplanować je i stworzyć wszystkie mechanizmy kontrolne do wysyłania wiadomości e-mail i powiadamiania ludzi. Mamy nadzieję, że to da Wam wytchnienie i naładujecie akumulatory do patrzenia na sprawy na wyższym poziomie, zawodowym.

Open Source Alternatywy dla PDI

Na początku rozdziału omawialiśmy programy Open Source ETL. Do opracowania tego rozdziału wykorzystaliśmy PDI (lub Kettle) firmy Pentaho. Uważamy, że jest to prawdopodobnie najlepsze darmowe/otwarte oprogramowanie ETL na rynku. Jednak w niektórych przypadkach interesujące może być zbadanie alternatyw. Bardzo dobry nazywa się Talend, a jego flagowy produkt Talend Open Studio do integracji danych można pobrać z:

https://www.talend.com/products/talend-open-studio

Idea Open Studio jest bardzo podobna do PDI. Z projektantem graficznym możemy napisać wszystkie potrzebne transformacje, a następnie skompilować. Dane wyjściowe różnią się nieco od PDI tym, że generują kod Java, ale wszystkie koncepcje i metodologie omówione w tym rozdziale mają zastosowanie w ten sam sposób. Powodem, dla którego w tym rozdziale wybraliśmy PDI zamiast Talend, jest przekonanie, że dla nowych użytkowników PDI może być łatwiejsze do zrozumienia, przynajmniej na początku. Tam, gdzie oba oferują funkcję przeciągnij i upuść, uważamy, że interfejs PDI jest bardziej przejrzysty i minimalistyczny. Każdy krok ma podobny zestaw opcji, a większość kroków działa od razu po wyjęciu z pudełka. Wystarczy połączyć wejście z wyjściem i prawie gotowe. Talend ma wiele opcji do skonfigurowania, więc w niektórych przypadkach może być potężny, ale głupotą byłoby nie doceniać PDI, ponieważ jest to również bardzo potężne narzędzie. Możesz ulepszać lub dodawać nowe funkcje za pomocą wtyczek, które są udostępniane w witrynie społeczności lub w Internecie. W sumie to, jakie narzędzie wybrać, jest czasem osobistą decyzją. Jeśli Twój zespół ma już pewną wiedzę na temat któregoś z narzędzi, trzymaj się tego konkretnego rozwiązania. Jeśli w Twojej organizacji nie ma wcześniejszej wiedzy na temat któregokolwiek z tych narzędzi, zalecamy rozpoczęcie od PDI. W każdym razie przed rozpoczęciem sprawdź w Internecie lub na forach pomocy informacje o transformacjach, które musisz opracować przed ich rozpoczęciem, i stwierdzeniu, że są one trudne lub prawie niemożliwe do opracowania za pomocą określonego narzędzia.

Wniosek

Przyjrzeliśmy się, czym jest proces ETL. Poprzednia część służyła jako wprowadzenie do naszego modelu danych i położył podwaliny pod ten rozdział. Wiemy, że jest sporo do przetworzenia, ponieważ napisanie ETL nie jest prostym zadaniem, ale mamy nadzieję, że wszystkie przekształcenia zaprojektowane w tym rozdziale, a także wszystkie zastosowane kroki i metodologie są teraz jasne. Zasadniczo koncepcja polega na przenoszeniu danych i jednoczesnym stosowaniu wymaganych przekształceń. Aby to osiągnąć, aby nie wpływać na systemy produkcyjne, które mogą przetwarzać ważne zamówienia lub fakturować, pobieramy dane poza godzinami szczytu do naszej tymczasowej bazy danych. Tam wybieramy zasadniczo dwie możliwe strategie: albo wykonaj ładowanie przyrostowe za pomocą kroku Wstaw/Aktualizuj, który jest odpowiedni dla dużych tabel, zwłaszcza tabel faktów, albo dla tabel wymiarów, w których rekordy mogą zniknąć z operacji, i pozwól strategii Obcinanie/Wstaw dla tabeli wymiarów i tabel, które nie mają tendencji do dużych zmian i gdzie nie ma ryzyka utraty wartości. Jeśli przeżyłeś tą sekcję, jesteś gotowy, aby odkryć prawdopodobnie najpiękniejszą część projektu BI. Część raportowa to ta, która prezentuje wyniki użytkowników na podstawie danych, z którymi pracujemy. W kolejnych częściach opracujemy raporty i pulpity nawigacyjne, w których użytkownik będzie mógł interaktywnie bawić się danymi, które zgromadziliśmy w tym rozdziale. Będziemy również tworzyć niezbędne procesy i planować je, aby mieć pewność, że nasze dane są aktualne i wszystko działa jak w zegarku.

7. Ulepszenia wydajności

W firmie średniej wielkości jest wysoce prawdopodobne, że ilość danych, które posiadamy, nie jest tak duża. Jeśli jednak jesteśmy firmą detaliczną, jest prawdopodobne, że nasz transakcyjny może mieć dobrą liczbę transakcji, zwłaszcza jeśli mamy dane z kilku lat. Niezależnie od tego, czy dotyczy to Twojej firmy, czy nie, ważne jest, aby mieć pewną ekspozycję, aby poprawić wydajność podczas pracy z bazami danych. To jest powód, dla którego ta część została dodana: aby zapewnić Ci wgląd w to, jak przyspieszyć swoje procesy. Nie jest to zbyt obszerna kompilacja, ale na początek powinna wystarczyć. Przyjrzymy się kilku optymalizacjom wydajności zarówno na poziomie bazy danych, jak i łańcucha ETL. Zacznijmy od tego, gdzie znajdują się dane (baza danych), a następnie przejdźmy do obliczeń.

Optymalizacje baz danych

Istnieje wiele operacji, które można wykonać na bazie danych w celu zwiększenia wydajności. Istnieje również inny zestaw działań, które same w sobie nie poprawią bezpośrednio wydajności, ale mogą pomóc pośrednio, na przykład włączenie kompresji danych. Najpierw przedstawimy listę zaleceń i wyjaśnimy, dlaczego są one ważne podczas pracy z bazami danych.

Unikaj używania dat dla kluczy łączenia (i kluczy podstawowych)

To zalecenie jest szczególnie prawdziwe, gdy dane zawierają również czas lub są polem znacznika czasu lub, co gorsza, jeśli zawierają strefę czasową jako część danych. Te klucze mogą reprezentować problemy podczas dopasowywania dwóch tabel, zwłaszcza jeśli są to różne formaty danych w różnych bazach danych. Każda baza danych ma własną metodę przechowywania dat, a porównanie tych pól może być trudne. Zwykle, jeśli chcemy użyć daty, czasami warto dodać nową kolumnę lub po prostu zapisać ją jako liczbę. Oczywiście musimy szanować pierwotną wartość klucza, więc jeśli zawiera część czasową, a ta część jest reprezentatywna, musimy to wziąć pod uwagę. Używanie ciągów znaków czasami nie jest tak dobre, jak wybór zwykłych liczb całkowitych, zwłaszcza jeśli są one tak długie lub mają zmienną długość. Użycie tak zwanych kluczy zastępczych (kluczy generowanych automatycznie), gdy w tabeli nie ma oczywistego klucza podstawowego, jest zazwyczaj dobrym pomysłem w takich przypadkach. Należy również wziąć pod uwagę, że aby umożliwić partycjonowanie, czasami istnieją ograniczenia dotyczące typu danych pola. Omówimy podział w dalszej części tej części, ale może to być również dobry powód do podjęcia decyzji o konkretnym typie danych kolumny.

Analizuj tabele w swojej bazie danych

Wszystkie silniki baz danych wymagają zaktualizowanych statystyk dotyczących dystrybucji danych, kształtu i innych elementów każdej tabeli w bazie danych. Statystyki te są używane przez optymalizator bazy danych do wybierania kolejności łączenia, określania, czy używać indeksu, czy nie (więcej o indeksach można przeczytać za jakiś czas) oraz przeprowadzania zestawu wewnętrznych optymalizacji zakresu w celu rozwiązania zapytania. Jeśli te statystyki nie są dostępne lub, co gorsza, są nieaktualne, optymalizator może opracować nieoptymalny plan. Oznacza to, że zapytanie będzie potencjalnie działać znacznie gorzej niż powinno. Aby decyzje podejmowane przez optymalizator były jak najbardziej trafne, chcemy aktualizować statystyki naszej tabeli. Jeśli całkowita ilość danych do pobrania przez zapytanie jest niewielka, nie jest to tak ważne, ale gdy tabele zaczynają rosnąć, może to być bardzo ważny czynnik. Zły plan może być o kilka rzędów wielkości gorszy od planu optymalnego. Może to prowadzić do bardzo niskiej wydajności lub nawet zawieszenia zapytania, jeśli mamy do czynienia z dużą ilością danych. Każda baza danych ma własny zestaw narzędzi do zbierania statystyk. W MySQL/MariaDB możemy użyć, podobnie jak w niektórych innych bazach danych, ANALYZE TABLE nazwa_tabeli; Komenda. W Oracle musimy wywołać pakiet dbms_stats i tak dalej dla innych baz danych. Pełna składnia MariaDB jest następująca:

ANALYZE TABLE table_name [,table_name …]

[PERSISTENT FOR [ALL|COLUMNS ([col_name [,col_name

...])] [INDEXES ([index_name [,index_name &helip;])]]

Ale w większości przypadków skorzystamy z analizy podstawowej: dla całej tabeli i wszystkich indeksów zależnych znajduje się następujące stwierdzenie:

ANALYZE TABLE table_name PERSISTENT FOR ALL;

A jeśli używasz starej wersji MariaDB/MySQL, użyj:

ANALYZE TABLE table_name;

Czasami może być konieczne zrobienie tego poza MariaDB CLI lub edytorem GUI. W tym przypadku MariaDB i MySQL mają dołączone narzędzie, które pozwala analizować tabele z zewnątrz. To narzędzie nazywa się mysqlcheck i oprócz analizy tabeli może ją również sprawdzać, optymalizować i naprawiać, w zależności od przełącznika użytego w wywołaniu programu. W przypadku analizy tabeli wywołanie powinno wyglądać następująco:

./client/mysqlcheck [OPTIONS] database [table]

Tak więc do analizy tabeli składnia powinna wyglądać następująco:

./client/mysqlcheck -h host_where_the_db_resides -u

username -ppassword -a database_name table_name

A prawdziwy przykład będzie wyglądał następująco:

bibook@bibook:~$ mysqlcheck -h localhost -u bibook

-pxxxxxx -a dwh t_f_sales

dwh.t_f_sales

OK

I proszę, zauważ jeszcze raz, że nie ma spacji między -p i hasło, jak w przypadku połączenia klienckiego. Ostatnią rzeczą, którą możesz zrobić, oprócz analizy tabel, jest ich defragmentacja. To polecenie jest bardzo ważne, gdy dane z tabeli były wielokrotnie usuwane i ponownie wstawiane. Podobnie na dysku twardym danych może nie być umieszczone w sposób ciągły w silniku pamięci masowej, przez co pobieranie tych danych jest trudniejsze i wolniejsze niż powinno. To polecenie jest również interesujące, aby odzyskać zmarnowane nieużywane miejsce w tabeli. Ponownie, każdy dostawca baz danych ma w tym celu określoną procedurę, ale w przypadku tabel MariaDB można użyć następującej składni:

OPTIMIZE TABLE table_name;

Lub równoważne wywołanie wiersza poleceń

./client/mysqlcheck -h host_where_the_db_resides -u

username -ppassword -o database_name table_name

Przykład jest pokazany poniżej:

bibook@bibook:~$ mysqlcheck -h localhost -u bibook

-pxxxxxx -o dwh t_f_sales

dwh.t_f_sales

note : Table does not support optimize, doing

recreate + analyze instead

status : OK

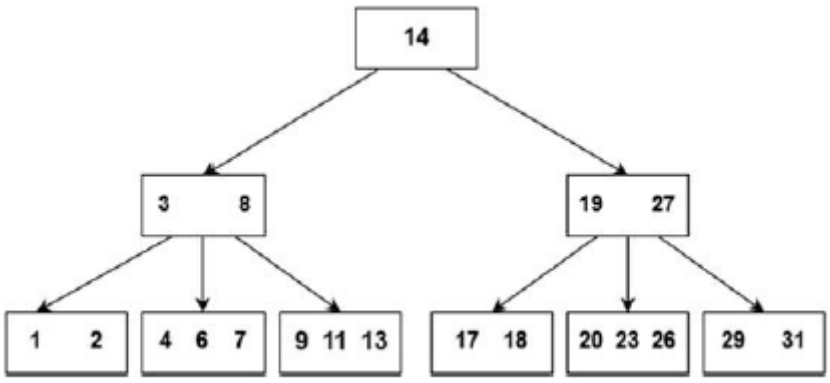
Uwaga: jeśli tabela jest duża, te polecenia mogą zająć trochę czasu, zwłaszcza polecenie optymalizacji, jeśli spróbuje później odtworzyć i przeanalizować. Używaj ich podczas okien konserwacyjnych lub gdy wiesz, że nie wysyłasz zapytań do tabel.

Indeksowanie, czyli jak przyspieszyć zapytania

Niektóre kolumny w tabeli mogą być częścią indeksu. Indeks to struktura danych, która umożliwia szybkie wyszukiwanie wartości. Pomyśl o indeksie książki. Jeśli chcesz znaleźć określony temat lub słowo, przejdź do indeksu i szybko znajdź dokładną stronę, zamiast przeglądać wszystkie strony. W bazach danych koncepcja jest bardzo podobna. Chcielibyśmy tworzyć indeksy dla pól, które są regularnie odwiedzane, aby optymalizator zapytań nie musiał skanować wszystkich danych, ale zaglądał do indeksu, aby wiedzieć, gdzie znajdują się dane i bezpośrednio uzyskiwać dostęp do lokalizacji. Jeszcze lepiej, w zależności od przypadku, jeśli wszystkie potrzebne informacje znajdują się już w indeksie, w niektórych bazach danych dostęp do bloków danych nie jest nawet konieczny. Istnieje kilka różnych typów indeksów, w zależności od sposobu ich implementacji, a niektóre z nich są lepsze od innych. Ponownie, zależy to od dostawcy bazy danych, więc przed próbą ich użycia sprawdź w instrukcji, czy te typy indeksów są dostępne. W tym rozdziale przyjrzymy się zasadniczo dwóm najbardziej powszechnym: standardowym indeksom drzewa B+ oraz indeksom bitmapowym.

Indeksy standardowe lub B+ TREE

Najpopularniejszymi indeksami są indeksy B+ TREE. Elementy indeksu są przechowywane w posortowanej kolejności. Chodzi o to, aby mieć strukturę podobną do drzewa, z korzeniem na górze, a następnie wskaźnikami do liści, które z kolei są wskaźnikami do innych liści, aż do samego poziomu końcowego, który jest tylko liściem bez wskaźników, i to liście zawierające ostatnie wartości. Jeśli dane są uporządkowane, bardzo łatwo jest przejść przez indeks i dotrzeć do liścia z żądaną wartością. Nie będziemy wdawać się w zbyt wiele szczegółów na temat tego, jak fizycznie te indeksy są przechowywane i używane, ponieważ nie jest to celem książki i zależą one od każdej implementacji w każdej pojedynczej bazie danych, więc sprawdź konkretny podręcznik od dostawcy, jeśli są tym zainteresowani. Na potrzeby tego rozdziału wystarczy zrozumieć, że dane są uporządkowane, a to jest świetny przykład przyspieszenia zapytania; a gdy wszystkie żądane dane znajdują się w indeksie, nie ma potrzeby sprawdzania bloków danych. Tylko zdjęcie do zilustrowania, dzięki czemu możesz teraz trochę lepiej zrozumieć, jak to działa. Rysunek przedstawia indeks nad kolumną zawierającą wartości liczbowe z zakresu od 0 do 31, ale nie wszystkie z nich są obecne.



Wiersz będzie zawierał partycję tej listy, więc podczas odpytywania indeksu pierwszą rzeczą do zrobienia jest porównanie szukanej liczby z liczbą w katalogu głównym. Jeśli szukana liczba jest mniejsza niż pierwiastek, pójdziemy lewą ścieżką, w przeciwnym razie pójdziemy prawą ścieżką. Zobaczmy teraz praktyczny przykład. Wyobraź sobie dużą tabelę, zwykle tabelę faktów, którą zaprojektowaliśmy w poprzednich częściach, w której masz szczegółowe informacje o kliencie i szczegóły produktu na poziomie faktury. Jak powiedzieliśmy wcześniej, może to być bardzo duży stół, zwłaszcza jeśli prowadzisz działalność detaliczną, z mnóstwem transakcji każdego dnia. W pewnym momencie nasza tabela hurtowni danych zacznie wymykać się spod kontroli i chociaż możemy zrezygnować z indeksowania kilku kolumn, ponieważ zwykle używamy większości danych do naszej analizy, dla niektórych osób może ich analiza skupia się głównie w kilku kolumnach, np. aby wprowadzić licznik produktów zakupionych przez każdego klienta, a oni chcą mieć do nich szybki dostęp. Równie dobrze może być możliwe utworzenie tabeli faktów pochodnych, na przykład tabel ze skumulowanymi danymi na poziomie tygodniowym lub miesięcznym, i chcemy zindeksować te daty w celu szybkiego wyszukiwania w okresie czasu lub dla określonego klienta.

Uwaga: W przypadku analizy czasowej partycjonowanie (co zobaczymy w dalszej części tego rozdziału) może działać znacznie lepiej niż indeksowanie, zwłaszcza w środowisku hurtowni danych, ponieważ zestaw opcji jest ograniczony. Jeśli mamy bardzo niewielu klientów, którzy generują dużo danych, partycjonowanie może być również wyborem, ale na pierwszy rzut oka bardziej sensowne jest indeksowanie kolumny klientów.

Wróćmy do naszej przykładowej tabeli. W tej chwili ta tabela zawiera prawie 1,5 gigabajta danych i około 6 milionów wierszy. Rośnie w tempie ponad 3000 rekordów dziennie. Więc tak, prowadzimy całkiem przyzwoitą firmę, prawda? W tym przykładzie zindeksowałem kod produktu. To prawdopodobnie nie będzie najbłyskotliwszy pomysł, ponieważ prawdopodobnie będziesz odnosić się do produktu po nazwie, ale jeśli pamiętasz z naszego projektu hurtowni danych w kształcie gwiazdki i płatka śniegu, mieliśmy łączenia przez id. Tak więc bezpośrednie zapytanie do jednej tabeli poprzez pobranie identyfikatora produktu wygląda absurdalnie, ale podczas łączenia tabel, jeśli identyfikatory są indeksowane, znacznie przyspieszamy zapytania. Ale nie idź jeszcze dalej; po prostu zacznij od prostej tabeli i sprawdź moc indeksu.

select * from fact_table where productid = 56894

/* Affected rows: 0 Found rows: 2 Warnings:

0 Duration for 1 query: 0,047 sec. */

Spróbujmy nie używać indeksu i zobaczmy, co się stanie:

select * from fact_table IGNORE INDEX (id_idx)

where productid = 56894

/* Affected rows: 0 Found rows: 2 Warnings:

0 Duration for 1 query: 5,235 sec. */

Jak widać, różnica jest ogromna, prawda? Ok, 5 sekund może nie wyglądać na zbyt wiele, ale wyobraź sobie większe tabele, złączenia i podzapytania z innymi zapytaniami lub tabelami, a to może zacząć rosnąć wykładniczo! Z drugiej strony posiadanie indeksu ma pewien wpływ na wydajność. Jeśli w tabeli znajduje się indeks, operacje DML, takie jak usuwanie, aktualizowanie i wstawianie, będą podlegały pewnym karom. Zwykle nie stanowi to problemu, zwłaszcza gdy te operacje są wykonywane podczas etapów ładowania wsadowego i wpływają tylko na hurtownię danych, ale w systemie operacyjnym dodanie indeksu, który powoduje poważne przeciążenie wstawiania transakcji, jest nie do przyjęcia. Ale tak jest zawsze w życiu: kompromis między kilkoma aspektami, w tym przypadku szybkością podczas wysyłania zapytań do szybkości podczas ładowania wsadowego. W hurtowni danych może to być tego warte, ponieważ zwykle wstawiasz raz, ale wysyłasz wiele zapytań. Dobrą opcją poprawy ładowania danych (którą można wyjaśnić również w następnej sekcji dotyczącej optymalizacji ETL) jest po prostu usunięcie indeksów podczas ładowania i ponowne utworzenie ich po zakończeniu.

Uwaga: Użyliśmy wskazówki IGNORE_INDEX w MySQL/MariaDB. To instruuje optymalizator, aby nie używał indeksu do rozwiązania zapytania. Jeśli nie potrzebujesz już indeksu, upuść go, nawet jeśli nie jest używany; indeks jest utrzymywany, gdy w tej tabeli są wstawiane, aktualizowane lub usuwane elementy, które wpływają na wydajność.

Indeksy map bitowych

Indeksy bitmapowe to inne typy indeksów szeroko stosowane w hurtowniach danych. Indeksy te nie przypominają drzewa, jak indeksy BTREE, ale przechowują różne wartości kolumny, zakodowane w sekwencji bitów. Wyobraźmy sobie, że potrzebujemy analizy w czasie rzeczywistym w naszej tabeli faktów t_f_sales, która staje się bardzo duża tylko po to, aby wykryć, które zamówienia mają status Wycenione, ale nie zostały jeszcze zafakturowane. Możliwymi wartościami dla kolumny Status_id są Cytaty, Zamówione lub Zafakturowane. Stanowi to niewielką możliwą wartość wyników, więc w tym przypadku a indeks mapy bitowej może być lepiej dopasowany niż drzewo btree, przydatne w przypadku kolumn o dużej liczności. Chcemy, aby nasze zapytanie skanowało wszystkie wpisy w tabeli faktów, które mają status Wycenione, abyśmy mogli naprawdę przejść do szczegółów zamówienia i rzucić okiem na wszystkie szczegóły zamówienia, aby nasi pracownicy lub finansiści mogli sprawdzić, dlaczego nie są jeszcze zafakturowane i w razie potrzeby przejdź do wystawienia faktury. W przypadku statusu wyobraź sobie, że mamy tylko cztery możliwe wartości statusu: nieznane, podane, zamówione i zafakturowane. Potrzebujemy 2 bitów, aby przedstawić status jako 2 bity * 2 możliwe wartości na bit (1 lub 0) = 4 kombinacje. Indeks zostanie utworzony przy użyciu struktury podobnej do tej w tabeli

Invoice : Num Status_id : Bitmap

1 : unknown : 00

2 : quoted : 01

3 : quoted : 01

4 : ordered :10

5 : invoiced : 11

6 : ordered : 10

7 : quoted : 01

Dzięki takiej strukturze możemy mieć mały indeks ze wszystkimi możliwymi statusami. Indeksy te zwykle działają dobrze w projekcie schematu gwiazdy lub płatka śniegu, takim jak nasz. Zwykle umieszcza się je na kluczach obcych tabeli faktów, ponieważ odnoszą się one do tabel wymiarów, które zwykle są znacznie mniejszymi tabelami. Na przykład, jeśli mamy tylko kilku klientów, ale generuje to duże zamówienia na faktury, sprytnym rozwiązaniem może być umieszczenie indeksu mapy bitowej w kolumnie obcej, która odwołuje się do identyfikatora klienta w tabeli t_f_sales. Niestety, MariaDB ani MySQL nie obsługują jeszcze w tym momencie indeksów map bitowych. Jest to już wymagana funkcja, ale nie jest jeszcze jasne, kiedy zostanie wdrożona. Prawdopodobnie potrzebujesz relacyjnej bazy danych, takiej jak Oracle, z wersją korporacyjną, aby ją obsługiwać, co prawdopodobnie nie jest w twoim zakresie ze względu na koszty licencji, ale należy o tym pamiętać.

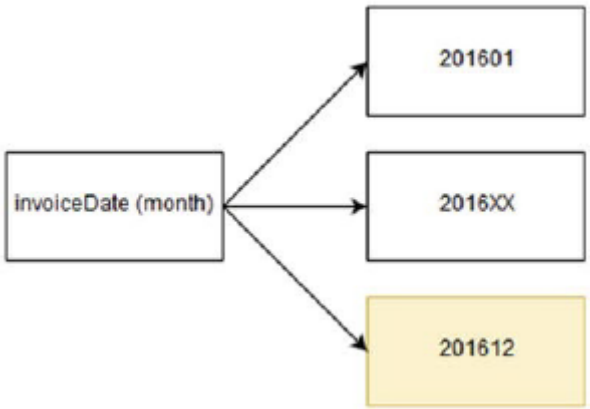
Indeksy FULLTEXT

MariaDB i MySQL obsługują indeksy FULLTEXT. Te indeksy są przydatne, gdy trzeba zaindeksować kolumny z informacjami tekstowymi: kolumny zawierające CHAR, VARCHAR lub dowolne warianty TEXT. Te indeksy są naprawdę potężne, gdy chcesz wyszukiwać kolumny tekstowe, ale nie masz dokładnych informacji zawartych w kolumnach. Indeks PEŁNOTEKSTOWY można umieścić na jednym z tych produktów i znacznie przyspiesza zapytanie, jeśli jest używany ze składnią podobną do następującej: PODAJ.POZYCJĘ (opis_produktu) PRZECIW (wyrażenie [modyfikator_wyszukiwania]). Wyobraź sobie naszą tabelę wymiarów dla produktów, t_l_product. W pewnym momencie wprowadziliśmy nowy produkt, ale z innym opakowaniem i inną liczbą sztuk. Spowoduje to wygenerowanie wielu kodów SKU, po jednym dla każdej konkretnej wersji. Tak, wszystkie produkty są takie same, ale w naszej tabeli produktów będziemy potrzebować wpisu dla każdego konkretnego opakowania. Nazwa produktu prawdopodobnie będzie podobna, ale nazwa opakowania zostanie dodana w opisie produktu. produkt A, 0,5 ml; Produkt A 1l; Produkt Pakiet 4x4; Produkt A promocyjny 1,2l; i tym podobne. Jeśli chcemy przyspieszyć wyszukiwanie po kolumnie opisu produktu, dobrym wyborem jest indeks FULLTEXT. Ten indeks, zamiast przechowywać lub wykorzystywać całą wartość pola opisu w tabeli produktu, będzie przechowywać częściowe słowa opisu. Niestety nie działają one z klauzulami LIKE (indeks nie jest używany), więc mają ograniczone zastosowanie w środowisku hurtowni danych, zwłaszcza na etapie raportowania, chyba że użyjesz narzędzia, które jest naprawdę na tyle mądre, aby z niego skorzystać. W ETL może to mieć pewne zastosowanie, zwłaszcza jeśli zakodujesz zapytanie i dodasz klauzule MATCH i AGAINST, zamiast używać klauzuli LIKE, która jest preferowaną opcją używaną przez większość dostępnych narzędzi raportowania.

Partycjonowanie

Kiedy tabele zaczynają się rozrastać, zazwyczaj trzeba zdefiniować indeksy. Jednak czasami indeksy nie są najlepszą opcją, zwłaszcza gdy chcesz pobrać duży zestaw danych. Indeksy są zwykle wskazywane, gdy pobrane wartości są mniejsze niż 5% całości i są również dobre do swobodnego dostępu. Gdy chcesz pobrać znaczną liczbę rekordów z tabeli i identyfikujesz wzorzec dostępu do danych według określonych kolumn, a czasem nawet zakresu wartości dla tych kolumn, możesz znacznie zwiększyć wydajność, używając partycjonowania. Partycjonowanie polega na wybraniu jednej lub więcej kolumn i podzieleniu oryginalnej tabeli na mniejsze części w zależności od wartości tej kolumny. Partycjonowanie jest szeroko stosowane w przypadku atrybutów czasu, zazwyczaj dnia, miesiąca lub roku. Ale posiadanie partycji na dzień nie oznacza, że musisz mieć partycję na każdy dzień: możesz użyć zakresów, aby określić, że wszystkie dni w miesiącu znajdują się w tej samej partycji. Twoje pytanie może brzmieć: dlaczego używamy codziennego partycjonowania do przechowywania miesięcy?

Dlaczego nie tworzymy partycji według miesięcy? Główną zaletą partycjonowania jest użycie tego samego pola do filtrowania, co do partycjonowania. Musisz więc przeanalizować swoje procesy i raporty, aby sprawdzić, które pole jest najczęściej używane do filtrowania. Jeśli Twoje raporty i procesy są filtrowane według dnia, system partycjonowania będzie działał lepiej, jeśli zostanie podzielony na partycje według dnia, a wtedy możesz używać zakresów, które nie wymagają partycjonowania dla każdego dnia. Zobaczmy przykład partycjonowania. Wyobraź sobie, że mamy naszą tabelę faktów, t_f_sales. Jak wiesz w tym momencie, mamy kilka kolumn, które definiują linię faktury. Jednym z nich są dane do faktury. Wyobraź sobie teraz, że nasz dział finansowy jest bardzo zainteresowany zamówieniami, które zostały zafakturowane w bieżącym miesiącu. W tym momencie interesują nas wszystkie linie produktów, które mają datę faktury odpowiadającą aktualnemu miesiącowi. Jeśli podzielimy tabelę t_f_sales według daty faktury i określimy partycję dla każdego miesiąca, będziemy mogli pobrać rekordy z bieżącego miesiąca bez potrzeby indeksowania, a nawet lepiej, w sposób, który baza danych z łatwością zlokalizuje rejestry należące do tego miesiąca, pomijając wszystkie rejestry niespełniające warunku (tj. z poprzednich miesięcy). Wyobraź sobie, że chcemy uzyskać dostęp do partycji z grudnia 2016 r., Wtedy uzyskamy dostęp do partycji specjalnie na ten miesiąc.



Zobaczymy teraz, jak zaimplementowano partycjonowanie, ale mam nadzieję, że to brzmi dobrze. Jedyną wadą jest to, że czasami zapytanie nie ma standardowego wzorca dostępu. Czasami będziesz szukać po dacie faktury, innym po kliencie, a jeszcze innym po pracowniku lub produkcie. Jak zawsze, nie ma magicznego sposobu, który pasowałby wszystkim, więc należy zdecydować, która kombinacja jest najlepsza z możliwych, aw razie potrzeby przeprowadzić testy. Jak wyraźnie stwierdzono, partycjonowanie dobrze sprawdza się w przypadku dużych tabel, zwłaszcza gdy potrzebujemy pobrać znaczną ilość danych, ale może mieć również inne zastosowania, takie jak łatwość konserwacji, gdy chcemy zajmować się danymi historycznymi, ponieważ możemy przechowywać każdy inna partycja w innym pliku. Przeprowadzanie konserwacji, czyszczenia danych i automatycznego czyszczenia jest łatwiejsze, gdy mamy podzielone na partycje tabele. Oprócz wydajności są to inne powody, dla których partycjonowanie jest szeroko stosowane w hurtowniach danych. Aby utworzyć partycjonowaną tabelę, musimy tylko użyć tej samej instrukcji create table, ale na samym końcu musimy dodać nową klauzulę, aby określić partycjonowanie, jego typ i wyrażenie partition. Coś takiego należy dodać na końcu instrukcji tworzenia tabeli:

PARTITION BY partition_type

(partitioning_expression)

(

PARTITION partition_name VALUES

values_condition,

…

)

Typy partycjonowania

Teraz, gdy rozumiemy, jak działa partycjonowanie, nadszedł czas, aby przedstawić najpopularniejsze typy partycjonowania dostępne w większości silników baz danych. To są:

* Partycjonowanie zakresu;

* Partycjonowanie listy;

* Partycjonowanie skrótu, które nie zostanie tutaj wyjaśnione, ponieważ wykorzystuje matematyczną funkcję skrótu do równomiernego rozłożenia danych na partycje.

Partycjonowanie zakresu

Jak sama nazwa wskazuje, partycjonowanie zakresów polega na tworzeniu zakresu ciągłych wartości, które się nie nakładają, i grupowaniu ich razem w jednej partycji. Te wartości zostaną zdefiniowane przez wyrażenie partycji. Wróćmy do naszego przykładu t_f_sales. Wyobraź sobie, że chcemy teraz podzielić tabelę według daty day_id. Nasi finansiści chcą zobaczyć, jakie produkty są sprzedawane każdego miesiąca, aby sprawdzić, czy są jakieś produkty stacjonarne, które mają więcej zamówień lub sprzedaży w pewnym momencie roku lub w określonym sezonie. Chcemy mieć dostęp do określonego miesiąca i pobrać wszystkie produkty oraz liczbę sprzedanych produktów w tym konkretnym miesiącu. Jak wspomniano wcześniej, pobieranie danych z określonego miesiąca będzie działać lepiej w partycjonowaniu dziennym, jeśli filtrujemy za pomocą instrukcji między od pierwszego dnia miesiąca do ostatniego dnia miesiąca. Jak wynika z wymagań biznesowych, to zapytanie może skorzystać z partycjonowania, gdy dzielimy według daty day_id. Ponownie wybranie daty day_id jako klucza partycjonowania może być odpowiednie dla tego zapytania, ale nie dla innych. Prosimy więc wziąć pod uwagę wszystkie przypadki biznesowe i starać się jak najwięcej wiedzieć z wyprzedzeniem, jaka będzie metoda dostępu do tabeli, ponieważ zły projekt na tym etapie może poważnie wpłynąć na wydajność systemu w przyszłości. Na potrzeby tego ćwiczenia powiedziano nam, że będzie to główny dostęp do tabeli, więc zobaczmy, jak zaimplementować partycjonowanie zakresu w naszej tabeli faktów. Będziemy tworzyć określoną partycję (na podstawie day_id) dla każdego miesiąca. W naszym przypadku będziemy tworzyć partycje na wszystkie 2016 miesięcy. W tym celu zwróć uwagę na dwie rzeczy. Najpierw, aby utworzyć zakres, przekonwertujemy określoną datę na kod dnia. Aby to zrobić, istnieje wiele sposobów, ale MariaDB/MySQL ma funkcję o nazwie TO_DAYS, która zwraca numer dnia od pierwszego dnia pierwszego miesiąca roku 0. Dzięki temu będziemy mogli określić funkcję partycjonowania i naszą górną granice na podstawie wprowadzonych przez nas dat. Ponieważ chcemy mieć razem konkretne dane dla każdego miesiąca, jako górną granicę określamy 1. dzień tego miesiąca. Dzięki temu każda partycja będzie przechowywać rekordy, które nie należą do poprzedniej partycji, do wartości mniejszej niż pierwsza w następnym miesiącu. Drugim aspektem, o którym musimy pomyśleć, jest nazwanie naszych partycji. Tutaj nie ma reguły, więc możemy je nazwać jak chcemy, ale znowu zdrowy rozsądek powinien zwyciężyć, więc nazwijmy przegrody w sensowny sposób. Kod DDL służący do tworzenia tabeli partycjonowanej według zakresu t_f_sales jest następujący:

CREATE TABLE dwh.t_f_sales_byrange (

```
Order_id INT(11) DEFAULT NULL,
Order_Item_id INT(11) DEFAULT NULL,
Invoice_id INT(11) DEFAULT NULL,
Invoice_Item_id INT(11) DEFAULT NULL,
Product_id INT(11) DEFAULT NULL,
Status_id VARCHAR(8) DEFAULT NULL,
Order_desc VARCHAR(255) DEFAULT NULL,
Quantity_inv_num DOUBLE DEFAULT NULL,
Amount_inv_num DOUBLE DEFAULT NULL,
TAX_inv_num DOUBLE DEFAULT NULL,
Total_amount_inv_num DOUBLE DEFAULT NULL,
Day_id DATETIME DEFAULT NULL,
Order_Date_id DATETIME DEFAULT NULL,
Invoice_date_id DATETIME DEFAULT NULL,
Quantity_ord_num DOUBLE DEFAULT NULL,
Quantity_del_num DOUBLE DEFAULT NULL,
Quantity_quot_num DOUBLE DEFAULT NULL,
Currency_id INT(11) DEFAULT NULL,
Creation_Date_id DATETIME DEFAULT NULL,
Customer_id INT(11) DEFAULT NULL,
Employee_id INT(11) DEFAULT NULL
)E
NGINE = INNODB
AVG_ROW_LENGTH = 455
CHARACTER SET utf8mb4
COLLATE utf8mb4_general_ci
PARTITION BY RANGE (TO_DAYS(day_id))
(
PARTITION t_f_sales_201601 VALUES LESS THAN
(TO_DAYS('2016-02-01')),
PARTITION t_f_sales_201602 VALUES LESS THAN
```

```
(TO_DAYS('2016-03-01')),
PARTITION t_f_sales_201603 VALUES LESS THAN
(TO_DAYS('2016-04-01')),
PARTITION t_f_sales_201604 VALUES LESS THAN
(TO_DAYS('2016-05-01')),
PARTITION t_f_sales_201605 VALUES LESS THAN
(TO_DAYS('2016-06-01')),
PARTITION t_f_sales_201606 VALUES LESS THAN
(TO_DAYS('2016-07-01')),
PARTITION t_f_sales_201607 VALUES LESS THAN
(TO_DAYS('2016-08-01')),
PARTITION t_f_sales_201608 VALUES LESS THAN
(TO_DAYS('2016-09-01')),
PARTITION t_f_sales_201609 VALUES LESS THAN
(TO_DAYS('2016-10-01')),
PARTITION t_f_sales_201610 VALUES LESS THAN
(TO_DAYS('2016-11-01')),
PARTITION t_f_sales_201611 VALUES LESS THAN
(TO_DAYS('2016-12-01')),
PARTITION t_f_sales_201612 VALUES LESS THAN
(TO_DAYS('2017-01-01'))
);
```

>> DML succeeded [2,384s]

Wstawmy teraz nasze dane z t_f_sales do nowej podzielonej na partycje tabeli zakresu. W tym celu użyjemy prostej instrukcji insert jako instrukcji select:

```
insert into dwh.t_f_sales_byrange select * from
t_f_sales;
```

>> 36 rows inserted [0,268s]

And now, let’s check with a select that our data is ok:

```
select count(*) from dwh.t_f_sales_byrange;
```

>> 36 rows selected

Wszystko działa dobrze. Dobrą praktyką jest jednak dołączenie partycji catchall na samym końcu, z najwyższą możliwą wartością, z myślą o przyszłości. W przeciwnym razie, jeśli spróbujemy wprowadzić rekord i nie będzie partycji do przechowywania tego rekordu, system zgłosi błąd. Tutaj możesz zobaczyć błąd podczas wstawiania rekordu za styczeń 2017, ponieważ nie utworzono pasującej partycji. Wstawmy fałszywe zamówienie 99 na artykuł 999, zamówione i zafakturowane 1 stycznia 2017 r.:

```
INSERT INTO dwh.t_f_sales_byrange

(Order_id, Order_Item_id, Invoice_id,

Invoice_Item_id, Product_id, Status_id, Order_desc,

Quantity_inv_num, Amount_inv_num, TAX_inv_num,

Total_amount_inv_num, Day_id, Order_Date_id,

Invoice_date_id, Quantity_ord_num, Quantity_del_num,

Quantity_quot_num, Currency_id, Creation_Date_id,

Customer_id, Employee_id)

VALUES

(99, 999, 15, 24, 69, 'Invoiced', 'SO999', 1, 320,

48, 368, '2017-01-01 00:00:00', '2017-01-01 00:00:00',

'2017-01-01 00:00:00', 0, 0, 0, 3, '2017-01-01

00:00:00', 7, 1);
```

A zgłoszony błąd jest następujący:

```
Error (50,1): Table has no partition for value 736695
```

Zasadniczo oznacza to, że system nie był w stanie znaleźć partycji do przechowywania tej wartości. Jak powiedzieliśmy, partycjonowanie wymaga konserwacji. Dlatego powinniśmy wcześniej utworzyć wszystkie wymagane partycje. Jeśli ich nie stworzyliśmy, zawsze możemy dodać je później. Utwórzmy po prostu taką instrukcję na styczeń 2017 r. i spróbujmy ponownie uruchomić instrukcję wstawiania:

```
ALTER TABLE dwh.t_f_sales_byrange ADD PARTITION

(PARTITION t_f_sales_201701 VALUES LESS THAN

(TO_DAYS('2017-02-01')));

>> DML succeeded [0,732s]
```

Teraz próbujemy ponownie wstawić rekord i tym razem powinno działać:

```
INSERT INTO dwh.t_f_sales_byrange(Order_id,

Order_Item_id, Invoice_id, Invoice_Item_id,

Product_id, Status_id, Order_desc, Quantity_inv_num,

Amount_inv_num, TAX_inv_num, Total_amount_inv_num,
```

Day_id, Order_Date_id, Invoice_date_id,

Quantity_ord_num, Quantity_del_num, Quantity_quot_num,

Currency_id, Creation_Date_id, Customer_id,

Employee_id) VALUES

(99, 999, 15, 24, 69, 'Invoiced', 'SO999', 1, 320,

48, 368, '2017-01-01 00:00:00', '2017-01-01 00:00:00',

'2017-01-01 00:00:00', 0, 0, 0, 3, '2017-01-01

00:00:00', 7, 1);

>> 1 row inserted [0,119s]

Partycjonowanie listy

Do tej pory widzieliśmy partycjonowanie zakresu, które jest dobrym kandydatem do pracy z sekwencjami liczb lub dat. Ale czasami nie chcesz pracować z zakresami, ale z określonymi wartościami lub listą wartości. W takich przypadkach partycjonowanie listy przychodzi na ratunek. Składnia jest bardzo podobna do partycjonowania zakresu, ale będziemy określać wartości zamiast zakresu. Na końcu instrukcji CREATE TABLE dodamy następującą klauzulę:

PARTITION BY LIST (partitioning_expression)

(

PARTITION partition_name VALUES IN

(value_list),

…

[PARTITION partition_name DEFAULT]

)

Wyobraźmy sobie teraz nowe wymaganie. Chcemy szybko uzyskać wszystkie zamówienia według rodzaju fakturowanej waluty. Możemy mieć euro, dolary i tak dalej, a każdy facet w firmie zarządza innym regionem, więc faktury, które będzie śledził, różnią się od pozostałych. W tym momencie mamy dwie możliwe waluty: 1 i 3. Wartość 1 to dolary i 3 euro. Ale możemy mieć inne wartości. Musimy zdobyć je wszystkie, ponieważ będziemy musieli utworzyć partycję dla każdego. Zwróć też uwagę, że wartość w wyrażeniu musi być liczbą. Na przykład utworzenie partycjonowania dla Status_id zamówienia: Quoted, Ordered, Invoiced nie zadziała, jeśli nie przełożymy tych statusów na liczby (identyfikatory). Aby zrealizować to wymaganie, zdecydowaliśmy się podzielić tabelę t_f_sales na partycje List, tworząc dwie partycje (po jednej dla każdego identyfikatora waluty). Tutaj, w partycjonowaniu listy, nie mamy partycji typu catch-all, więc musimy się upewnić, że wartość zawsze będzie w jednej z partycji; w przeciwnym razie zostanie wyświetlony błąd. Kod do utworzenia partycji jest następujący:

CREATE TABLE dwh.t_f_sales_bylist (

Order_id INT(11) DEFAULT NULL,

```
Order_Item_id INT(11) DEFAULT NULL,
Invoice_id INT(11) DEFAULT NULL,
Invoice_Item_id INT(11) DEFAULT NULL,
Product_id INT(11) DEFAULT NULL,
Status_id VARCHAR(8) DEFAULT NULL,
Order_desc VARCHAR(255) DEFAULT NULL,
Quantity_inv_num DOUBLE DEFAULT NULL,
Amount_inv_num DOUBLE DEFAULT NULL,
TAX_inv_num DOUBLE DEFAULT NULL,
Total_amount_inv_num DOUBLE DEFAULT NULL,
Day_id DATETIME DEFAULT NULL,
Order_Date_id DATETIME DEFAULT NULL,
Invoice_date_id DATETIME DEFAULT NULL,
Quantity_ord_num DOUBLE DEFAULT NULL,
Quantity_del_num DOUBLE DEFAULT NULL,
Quantity_quot_num DOUBLE DEFAULT NULL,
Currency_id INT(11) DEFAULT NULL,
Creation_Date_id DATETIME DEFAULT NULL,
Customer_id INT(11) DEFAULT NULL,
Employee_id INT(11) DEFAULT NULL
)E
NGINE = INNODB
AVG_ROW_LENGTH = 455
CHARACTER SET utf8mb4
COLLATE utf8mb4_general_ci
PARTITION BY LIST (Currency_id)
(
PARTITION t_f_sales_1 VALUES IN (1),
PARTITION t_f_sales_3 VALUES IN (3)
);
```

Uwaga: Do każdej partycji możemy przypisać różne możliwe wartości. Jeśli facet odpowiedzialny za fakturowanie w euro, funtach brytyjskich i innych walutach europejskich jest ten sam, możemy umieścić wszystkie te identyfikatory walut w tej samej partycji, umieszczając wszystkie możliwe identyfikatory wewnątrz klauzuli in oddzielone przecinkami: WARTOŚCI W (3, 4,5,6) na przykład. Teraz możemy wstawić wartości. Wszystko idzie dobrze, więc nie powinniśmy otrzymać komunikatu o błędzie:

insert into dwh.t_f_sales_bylist select * from

t_f_sales;

>> 36 rows inserted [0,149s]

Rozważania dotyczące partycjonowania

Dobrą praktyką jest zawsze dodanie partycji typu catch-all na końcu. Dzięki temu nie wystąpią takie błędy:

ALTER TABLE dwh.t_f_sales_byrange ADD PARTITION

(PARTITION t_f_sales_max VALUES LESS THAN (MAXVALUE));

>> DML succeeded [0,543s]

Więc teraz nawet próba dodania rekordu za 2018 rok nie zawiedzie:

INSERT INTO dwh.t_f_sales_byrange(Order_id,

Order_Item_id, Invoice_id, Invoice_Item_id,

Product_id, Status_id, Order_desc, Quantity_inv_num,

Amount_inv_num, TAX_inv_num, Total_amount_inv_num,

Day_id, Order_Date_id, Invoice_date_id,

Quantity_ord_num, Quantity_del_num, Quantity_quot_num,

Currency_id, Creation_Date_id, Customer_id,

Employee_id) VALUES

(99, 999, 15, 24, 69, 'Invoiced', 'SO999', 1, 320,

48, 368, '2018-01-01 00:00:00', '2018-01-01 00:00:00',

'2018-01-01 00:00:00', 0, 0, 0, 3, '2018-01-01

00:00:00', 7, 1);

>> 1 row inserted [0,068s]

Teraz wyobraź sobie, że chcemy po prostu zhistoryzować nowo utworzoną partycję. Wyeksportowałeś dane dla starych partycji i chcesz je usunąć z bazy danych, aby zwolnić miejsce na nowe rekordy. Wprawdzie nie ma to większego sensu, ponieważ zawsze będziesz chciał usunąć najstarsze dane, a nie najnowsze, ale to tylko próbka. Uruchommy instrukcję usuwania partycji:

select count(*) from dwh.t_f_sales_byrange where

order_id = 99;

>> 1

ALTER TABLE dwh.t_f_sales_byrange DROP PARTITION

t_f_sales_201701;

>> DML succeeded [0,432s]

select count(*) from dwh.t_f_sales_byrange where

order_id = 99;

>> 0

Jak widać, dane dołączone do tej partycji zniknęły. Zachowaj ostrożność podczas wdrażania czyszczenia, ponieważ nie ma łatwego sposobu na przewinięcie tego.

Używając zdania EXPLAIN

Kiedy piszemy zapytanie, mówimy bazie danych, do których tabel ma uzyskać dostęp, które łączenia mają wykonać, ale nie określamy żadnej kolejności ani nie instruujemy bazy danych, aby używała jakiegokolwiek indeksu ani innej struktury. Baza danych ma pewną swobodę wyboru tego, co uważa za najlepszy sposób rozwiązania zapytania, które do niej przedstawiamy. Jak widzieliśmy w tym rozdziale, aby mieć dobre plany wykonania, potrzebujemy statystyk, być może trzeba będzie zaindeksować niektóre kolumny, a także możemy użyć partycjonowania. Istnieją dalsze optymalizacje, które zobaczymy w tym rozdziale, ale w tym momencie mamy wystarczająco dużo materiału, aby przedstawić zdanie WYJAŚNIJ. To zdanie, poprzedzające dowolne zapytanie, zamiast je wykonać, powie nam dokładnie, jaki plan dostępu wybiera silnik bazy danych. Dzięki temu możemy łatwo sprawdzić, czy użyto indeksu, partycjonowano, czy też wysyłamy masowe zapytanie o wszystkie dane w tabeli. Możemy również zobaczyć kolejność dostępu do tabeli i kilka ciekawszych danych wejściowych. Spróbujmy najpierw uzyskać dostęp do pierwszej podzielonej na partycje tabeli przez określony Day_id:

SELECT *

FROM dwh.t_f_sales_byrange

WHERE day_id BETWEEN STR_TO_DATE('01/09/2016',

'%d/%m/%Y') and STR_TO_DATE('30/09/2016', '%d/%m/%Y');

Spowoduje to zwrócenie nam wszystkich rekordów, które mieliśmy we wrześniu 2016 r. Aby upewnić się, że partycjonowanie działa, możemy wywołać rozszerzoną wersję wyjaśniającą, zwaną wyjaśniającą partycje, która pokaże nam, skąd pobierane są informacje (usunęliśmy kilka dodatkowych informacji):

explain partitions select * from

dwh.t_f_sales_byrange where day_id between

STR_TO_DATE('01/09/2016', '%d/%m/%Y') and

STR_TO_DATE('30/09/2016', '%d/%m/%Y');

Jak widać w tabeli ,

id	select_type	table	partitions	type	Pos_keys	key	rows	Extra
1	SIMPLE	t_f_sales_byrange	t_f_sales_201609	ALL	null	null	9	Using where

kolumna partitions wskazuje, z której kolumny zebrano dane. Silnik był wystarczająco inteligentny, aby określić, do której partycji uzyskano dostęp z tabeli. Teraz spróbujmy uzyskać dostęp do partycjonowanej tabeli za pomocą jednego z pól, które nie jest częścią klucza partycji. Wyobraźmy sobie, że chcemy uzyskać dostęp do listy partycjonowanej tabeli, którą stworzyliśmy dla walut, ale używając dat. Oczekujemy, że baza danych nie będzie korzystała z partycjonowania, więc dostęp do wszystkich partycji będzie możliwy, ponieważ przy dostarczonych informacjach silnik nie może korzystać z partycjonowania. To zachowanie można zobaczyć w tabeli :

id	select_type	table	partitions	type	Pos_keys	rows	Extra
1	SIMPLE	t_f_sales_bylist	t_f_sales_1,t_f_sales_3	ALL	null	36	Using where

explain partitions select * from

dwh.t_f_sales_bylist where day_id between

STR_TO_DATE('01/09/2016', '%d/%m/%Y') and

STR_TO_DATE('30/09/2016', '%d/%m/%Y');

Możesz więc zobaczyć, jak uzyskuje dostęp do obu partycji. Spróbujmy teraz zebrać konkretny identyfikator waluty z partycjonowanej tabeli t_f_sales_bylist. Oczekiwane zachowanie możemy sprawdzić w tabeli

id	select_type	table	partitions	type	Pos_keys	rows	Extra
1	SIMPLE	t_f_sales_bylist	t_f_sales_3	ALL	null	18	Using where

explain partitions select * from

dwh.t_f_sales_bylist where currency_id = 3;

Widzimy więc wyraźnie, że partycjonowanie działa. Uzyskuje dostęp tylko do partycji o wartości 3 w kolumnie currency_id. To samo można zaobserwować w przypadku indeksów i innych aspektów zapytania. Jeśli używany jest indeks, zostanie on również wyświetlony. Stworzyłem indeks według identyfikatora zamówienia w tabeli. Jeśli poprosimy system o pobranie konkretnego zamówienia, zobaczmy, co się stanie. Tabela pokazuje, że tym razem do rozwiązania zapytania używany jest indeks:

id	select_type	table	type	Possible_keys	Key_len	ref	rows	Extra
1	SIMPLE	t_f_sales	ref	idx_t_f_sales_lookup	5	const	4	Using where

explain select * from dwh.t_f_sales where order_id

= 7;

Kolumna możliwe_klucze pokazuje nam, który indeks jest używany do rozwiązania zapytania. Teraz nadszedł czas, aby przyjrzeć się, co się dzieje, gdy w grę wchodzi więcej niż jeden stół. Spójrz na następujące zapytanie:

explain select * from dwh.t_f_sales s, t_l_customer

c, t_l_employee e

where s.Customer_id = c.Customer_id and

s.Employee_id=e.Employee_id

Mając to na uwadze, możemy zobaczyć określoną kolejność łączenia w tabeli.

id	select_type	table	type	Possible_keys	rows	Extra
1	SIMPLE	c	ALL	Null	5	
2	SIMPLE	e	ALL	Null	22	Using join buffer (flat, BNL join)
3	SIMPLE	s	ALL	null	36	Using where; Using join buffer (incremental, BNL join)

Tabela kolumn informuje nas o pierwszym uzyskanym dostępie i tak dalej. Dzięki temu możemy również sprawdzić, czy indeksy są używane i spróbować ustalić, skąd może pochodzić problem.

Uwaga: Dostrajanie zapytania jest złożone i wymaga praktyki. Decydująca jest kolejność tabel w złączeniu, użycie partycjonowania, indeksów i statystyk oraz wiele innych czynników. Więcej informacji na temat wydajności i sposobu interpretacji planów można znaleźć w dokumentacji MariaDB lub MySQL.

Widoki i widoki zmaterializowane

Widoki to pewnego rodzaju obliczenia zapytań, które są używane do tworzenia czegoś w rodzaju wirtualnej tabeli. W rzeczywistości tabela nie istnieje, ponieważ dane są obliczane w czasie zapytania na podstawie kodu SQL widoku. Są przydatne do ukrywania złożoności przed zapytaniami dla użytkownika oraz do obliczania określonych obliczeń bez konieczności ponownego kodowania tego samego zapytania w kółko. Zmaterializowane widoki są o krok dalej i poza przechowywaniem obliczeń i kodu i mogą zawierać rzeczywiste dane. Ten typ widoku jest używany głównie ze względu na wydajność. Jeśli mamy zapytanie obejmujące bardzo duże tabele, uzyskanie wyników zajmie prawdopodobnie zbyt dużo czasu. Powoduje to problem, zwłaszcza w środowisku hurtowni danych, gdy użytkownik biznesowy wykonuje raport. Niedopuszczalne jest proszenie użytkownika o czekanie pół godziny na zakończenie zapytania. Aby uniknąć takich sytuacji, istnieje możliwość stworzenia widoku zmaterializowanego, zawierającego nie tylko logikę zapytania, ale również wynik tych obliczeń. Scenariusz taki jak ten pomaga w rozwiązywaniu skomplikowanych lub czasochłonnych obliczeń, ponieważ obliczenia te można wykonać z wyprzedzeniem. Wyobraź sobie opisywany przez nas przypadek reportażu. Wiemy, że nasz pracownik finansowy będzie codziennie generował konkretny raport, rano i po południu. Wiemy, jak skarżył się wcześniej, że wykonanie raportu trwa zbyt długo. Możemy wstępnie obliczyć wynik tych obliczeń i zapisać je w zmaterializowanym widoku. Następnie musimy tylko zaplanować odświeżenie tych obliczeń (w rzeczywistości odświeżenie zmaterializowanego widoku) w pewnym momencie dnia, zanim wykona on raport. Wadą zmaterializowanego widoku jest to, że zawiera on tylko migawkę danych w momencie ostatniego odświeżenia. Jeśli więc chcemy mieć świeże dane, musimy najpierw odświeżyć widok zmaterializowany. Odświeżenie zajmie tyle samo czasu (i prawdopodobnie trochę więcej) niż czas wykonania zapytania, więc wyraźnie jest to korzystne dla zapytań, które są wykonywane wiele razy dziennie lub gdzie dane nie zmieniają się zbyt często, więc unikamy odświeżania zmaterializowanego widoki cały czas. Niestety, ani MySQL, ani MariaDB nie obsługują obecnie widoków zmaterializowanych. Ale jeśli używasz innego silnika, jest to wyraźny powód, aby go rozważyć. W przypadku MySQL i MariaDB możliwą opcją jest użycie procedur przechowywanych, które emulują kod

widoku i przechowywanie wyników w zwykłej tabeli. Opracowanie pełnej procedury składowanej wykracza poza zakres tej książki, ale może być całkiem przydatne. Masz dobry punkt wyjścia na tej stronie: http://www.mysqltutorial.org/mysql-stored-proceduretutorial.aspx

PORADNIK

Kiedy przeglądaliśmy sekcję Indeksy, zobaczyliśmy wprowadzenie do indeksów. W tym momencie rozdziału użyliśmy wskazówki NO_INDEX, aby poprosić silnik, aby nie używał określonego indeksu. To dobre rozwiązanie - poinstruować optymalizator, aby nie używał indeksu, nawet gdy jest w pełni sprawny - ponieważ wiesz, że będzie działał lepiej niż jego użycie. To dobrze, ale nie jest to przypadek użycia dla miejsca, w którym zbudowano wskazówki. Istnieje wiele wskazówek, a każda grupa wskazówek ma określone zadanie, ale główną ideą, którą należy zrozumieć, jest to, że wskazówki są modyfikatorami zapytań i że ich użycie wpływa na zachowanie silnika optymalizatora. Wpływając na optymalizator, faworyzujesz określone plany. Istnieją głównie dwie kategorie wskazówek w MYSQL i MariaDB: wskazówki optymalizatora i wskazówki indeksu. Niestety, pierwsza grupa występuje tylko w MySQL w wersji 5.7.7 lub nowszej. Większość czytelników zaznajomionych już z bazą danych Oracle uzna je za bardzo interesujące, ponieważ są one dość podobne do baz danych Oracle. Dzięki nim możemy wpływać na działanie optymalizatora, materializować część lub całe podzapytanie, dzięki czemu nie jest ono obliczane za każdym razem dla każdej wartości i tak dalej. Podczas gdy nie są one dostępne na poziomie podpowiedzi w MariaDB (są na poziomie bazy danych, ale będą miały wpływ na wszystkie zapytania, więc może zajść potrzeba zmiany rzeczy między zapytaniami), wciąż mamy sposób na wpłynięcie na optymalizator w MariaDB, określając kolejność łączenia tabel, która jest zwykle jednym z pierwszych aspektów, które należy wypróbować podczas dostrajania zapytania, wraz z metodą łączenia. Słowo kluczowe STRAIGHT_JOIN umieszczone po instrukcji SELECT lub częściej po instrukcji FROM powie optymalizatorowi, aby dołączył tabele w kolejności, w jakiej pojawiają się w zapytaniu. Dzięki tej wskazówce masz dobry sposób na dokładne poinstruowanie optymalizatora w kolejności używanych tabel. Idealnie byłoby zacząć od najbardziej restrykcyjnego stołu i skończyć na mniej restrykcyjnym, unikając pobierania większego na samym początku i przenoszenia wielu (prawdopodobnie) niepotrzebnych rejestrów od samego początku. Ale to zależy od każdego przypadku.

Uwaga: Wskazówki dotyczące optymalizatora nie są dostępne we wcześniejszych wersjach MySQL ani w MariaDB, ale można użyć parametru Optimizer_switch. Używając tego parametru, możesz kontrolować prawie wszystko, w tym typ sprzężeń i tak dalej.

Druga grupa, dostępna w obu bazach, to Wskazówki do indeksów. Są to, jak sama nazwa wskazuje, wskazówki dotyczące kontrolowania użycia indeksów. Mamy trzy możliwości: UŻYJ; IGNORUJ lub WYMUSZ użycie indeksu. Dołączając te słowa kluczowe zaraz po nazwie tabeli, tak jak to zrobiliśmy w przykładach należących do indeksów, możesz kontrolować, czy optymalizator może wybrać indeks, czy nie. Są one potężne, ale wymagają pełnej świadomości tego, co robisz, oraz szczegółowej wiedzy na temat działania indeksowania. Nie zalecamy ich używania, jeśli nie jest to konieczne, ponieważ prawdopodobnie wydajność będzie gorsza niż pozwolenie optymalizatorowi na wybranie właściwego podejścia.

Denormalizacja

Denormalizacja to koncepcja, którą widzieliśmy już w książce, ale warto o niej pamiętać. Denormalizacja to proces zwiększania wydajności poprzez powtarzanie pewnych wartości w bazie danych, co pozwala uniknąć dodatkowych połączeń między tabelami, a także dzięki kontrolowanemu powielaniu danych. Denormalizując bazę danych, zwiększamy wydajność, korzystając z mniejszej liczby operacji łączenia i szybszego wyszukiwania danych kosztem zajmowania większej przestrzeni lub

wielokrotnego przechowywania niektórych danych oraz dodając możliwość wystąpienia niespójności danych. Strategia ta jest szeroko stosowana w hurtowniach danych, ponieważ zwykle będziesz przygotowywać instrukcje select zamiast wstawiania, aktualizowania i usuwania. Zdania DML, podobnie jak poprzednia próba, będą miały więcej danych do przetworzenia w zdenormalizowanej bazie danych, więc będą działać gorzej w tych scenariuszach. Jest to coś do zaakceptowania, ponieważ hurtownia danych powinna mieć znacznie więcej wybranych aktywności niż aktywność DML.

Wyłączanie wyzwalaczy i ograniczeń

Nie jest to zbyt dobra praktyka, ale aby zwiększyć wydajność, może być możliwe wyłączenie ograniczeń i wyzwalaczy w tabelach. Należy to jednak robić ostrożnie, ponieważ wyłączenie ograniczenia, takiego jak klucz podstawowy, może nie powodować problemów podczas dodawania zduplikowanych wartości w tabeli. Więc może to mieć niepożądany skutek. Rób to tylko wtedy, gdy jesteś w pełni świadomy problemów z zabezpieczeniami, które może spowodować to rozwiązanie.

Optymalizacje ETL

Czasami łatwiej jest rozwiązać problemy z wydajnością u źródła danych, w naszym przypadku albo w naszej bazie danych zawierającej system operacyjny, co nie jest zalecane; lub w bazie danych hurtowni danych. Jeśli problem jest związany z wolno działającym raportem lub zapytaniem, należy to naprawić. Ale czasami możemy również napotkać problemy z wydajnością w naszych procesach ETL. Chociaż nie jest to tak złe, jak ich posiadanie, niż w części raportowej implementacji BI, może powodować ogromne problemy, takie jak opóźnienia w ładowaniu danych z poprzedniego dnia lub całkowity brak możliwości posiadania tych danych w hurtowni danych. W tym podrozdziale podamy kilka rad, jak rozwiązać te problemy.

Przenoszenie operacji do bazy danych

Pierwszym pomysłem podczas pracy z ETL jest odciążenie całej możliwej pracy jak najbliżej danych. W naszym przypadku zależy nam na tym, aby baza danych, w tym przypadku inscenizacja, wykonała jak najwięcej prac. Zwykle poprawia to wydajność, ponieważ masz najszybszy dostęp do danych. Jeśli to nie poprawi wydajności, prawdopodobnie wąskie gardło znajduje się w bazie danych. Zapoznaj się z pierwszą częścią tego rozdziału i spróbuj określić, czy opisane przez nas ulepszenia wydajności można wprowadzić do bazy danych.

Sprawdź łącza sieciowe

Jeśli nie ma możliwości przeniesienia pracy do bazy, wówczas będziemy musieli przyjrzeć się ETL. Jeśli korzystałeś z PDI (czajnik), istnieje kilka wskazówek, które mogą poprawić wydajność. Przede wszystkim upewnij się, że połączenie sieciowe między Twoimi źródłami danych (zwykle bazami danych) a hurtownią danych i serwerem ETL jest wystarczająco dobre. Jeśli dokonujemy szybkich transformacji i manipulacji danymi, ale łącza sieciowe są słabe, możemy przez pomyłkę pomyśleć, że to problem z bazą danych lub ETL. W tym celu warto skopiować duży plik między serwerami i obliczyć współczynnik pobierania i wysyłania. Jeśli używamy Linuksa, można to łatwo osiągnąć przez ssh, używając put i get. Jeśli jest to serwer Windows, możemy spróbować skopiować plik z dwóch różnych dysków sieciowych dowolnym protokołem udostępniania lub przez ftp i sprawdzić czasy.

Wskazówki dotyczące wydajności z PDI

Jeśli powyższe wskazówki nie działają lub nie poprawiają sytuacji, przejdźmy do kilku poprawek, które możemy wprowadzić w PDI, aby spróbować poprawić wydajność ETL. Zwiększ liczbę wierszy w zestawie wierszy W każdej transformacji możesz skonfigurować parametr o nazwie liczba wierszy w

zestawie wierszy. Jeśli masz problemy z niską wydajnością, możesz spróbować zwiększyć ten parametr. Domyślnie w nowych wersjach PDI ten parametr jest ustawiony na 10.000, ale starsze wersje miały znacznie niższą wartość. Jak zawsze w przypadku wydajności, jest to kwestia testów. Teoretycznie, jeśli masz transformacje, które przetwarzają ten sam zestaw danych w kółko lub wykonujesz powtarzalną pracę, zwiększenie tej liczby zwiększy przepustowość.

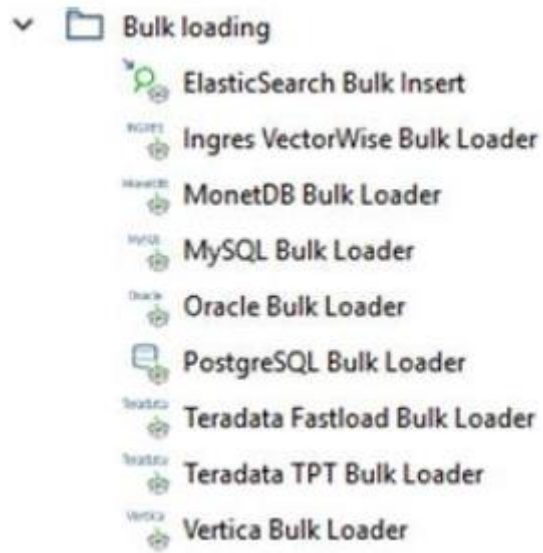
Przekształcenia równoległe

Projektując zadanie, zamiast uruchamiać wszystkie transformacje jedna po drugiej, sprawdź, czy możliwe jest utworzenie gałęzi z krokami lub transformacjami, które nie mają zależności z żadnymi innymi krokami lub transformacjami, które jeszcze nie zostały uruchomione. W takim przypadku te kroki lub przekształcenia są dobrymi kandydatami do zrównoleglenia.

Uwaga: zrównoleglanie rzeczy nie zawsze poprawia wydajność. Teoretycznie robienie rzeczy równolegle może zwiększyć wydajność w większości sytuacji, ale nie zawsze tak jest. Wyobraźmy sobie pierwszy scenariusz, w którym w tym samym czasie pozyskujemy dane ze zdalnej bazy danych i manipulujemy danymi w innej już pozyskanej gałęzi. Pierwszy będzie zależał od przepustowości między źródłową bazą danych a naszym serwerem etl, a drugi dotyczy mocy naszego serwera etl. W takim przypadku zrównoleglenie zadania może przynieść znaczny wzrost wydajności.

Masowe ładowanie i aktualizacje wsadowe

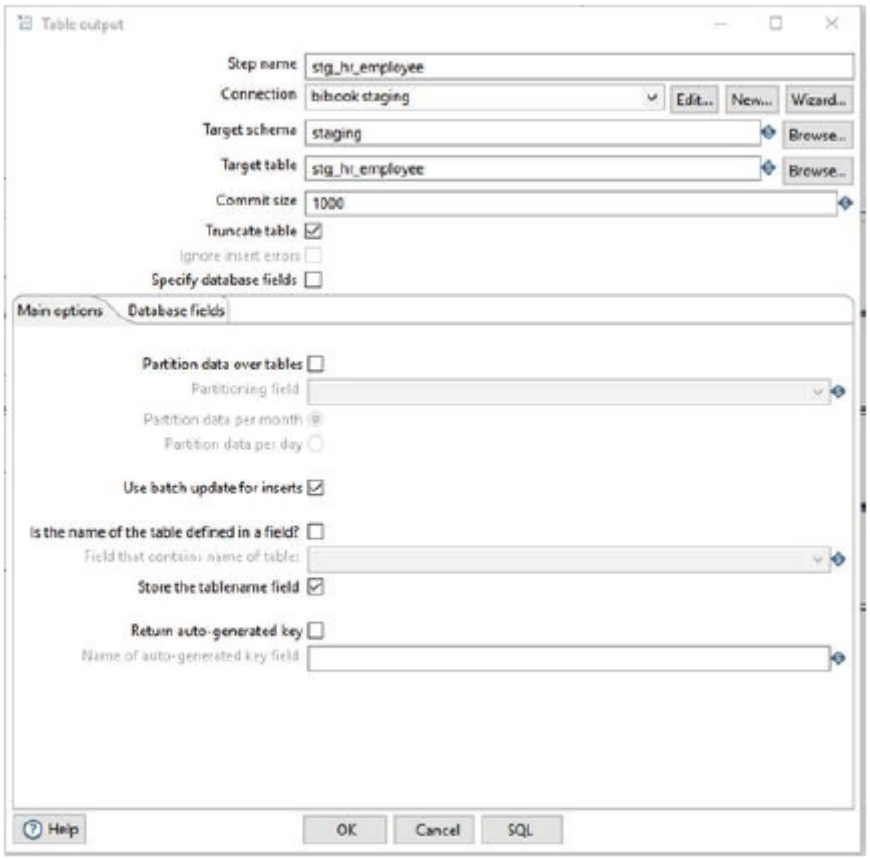
Zwykle wszystkie kroki bazy danych i/lub łączniki bazy danych mają opcję włączenia trybu zbiorczego. Tryb masowy jest przeznaczony do pracy z dużą ilością danych. Zamiast wstawiania jednego rejestru na raz, wstawianie zbiorcze grupuje kilka wstawek w poleceniu, a polecenie jest wysyłane do bazy danych w celu wykonania kilku małych instrukcji naraz. W większości przypadków bardzo dobrym rozwiązaniem jest użycie programu do ładowania masowego, ponieważ grupowanie małych instrukcji w większe oznacza tylko jedną transakcję z bazą danych zamiast wielu, co oznacza mniejsze obciążenie i oczywiście zwiększenie wydajności w znaczny sposób. Rysunek przedstawia dostępne programy ładujące luzem w najnowszych wersjach PDI.



Oprócz korzystania z kroku trybu masowego, czasami można również określić kilka parametrów w połączeniu z bazą danych, aby poprawić wydajność. Każda baza danych ma inny zestaw parametrów, więc sprawdź dokumentację dostawcy bazy danych, aby znaleźć parametr i dodać go we właściwościach połączenia.

Uwaga: Istnieje dobre źródło informacji o tym, jak znacznie zwiększyć wydajność baz danych MySQL/MariaDB, dodając kilka parametrów w połączeniu. Te parametry to głównie dwa: useServerPrepStmts i rewriteBatchedStatements. W tym wpisie na blogu masz wyjaśnienie i przewodnik, jak uwzględnić parametry i włączyć kompresję, a także połączenie MariaDB/MySQL PDI: https://anonymousbi.wordpress.com/2014/02/11/increasemysql-output-to-80k-rowssecond-in-pentaho-dataintegration/

Oprócz ładowania zbiorczego istnieją inne parametry, które można zmienić lub dostosować. Dobrą opcją (domyślnie włączoną) w krokach danych wyjściowych tabeli jest włączenie aktualizacji wsadowych. W głównej zakładce kroku tabeli wyjściowej znajduje się pole wyboru, aby włączyć tę funkcję. W tym samym oknie dialogowym pomocne może być również granie z rozmiarem zatwierdzenia. Duży rozmiar zatwierdzenia powinien nieco poprawić wydajność, ale może wydłużyć czas zatwierdzenia transakcji. Również w niektórych bazach danych, jeśli do bazy danych nie jest przypisana wystarczająca ilość pamięci, może to powodować problemy, ponieważ dane te nie zostały jeszcze zapisane w plikach danych i nadal znajdują się w strukturach pamięci w bazie danych, w których może zabraknąć miejsca. Możesz sprawdzić obie opcje na rysunku



Porządkowanie w operacjach scalania

Niektóre ETL działają znacznie szybciej, gdy istnieje operacja scalania lub operacja polegająca na łączeniu danych z różnych źródeł, jeśli dane są już posortowane. Niektóre ETL po prostu sortują dane,

gdy wykonują niektóre z tych operacji, ale jeśli wykryją, że dane są już posortowane w poprzednim kroku, czasami stosuje się optymalizacje, aby uniknąć zmiany kolejności danych. Posortowanie danych w poprzednim kroku może poprawić wydajność na etapie scalania.

Uwaga: w poście na stronie pomocy technicznej Pentaho znajduje się lista kontrolna wydajności, którą warto sprawdzić. Upewnij się, że przeczytałeś ją uważnie, jeśli napotkasz problemy w swoich Akcjach lub Transformacjach.

Wniosek

Do tego momentu widzieliśmy, jak radzić sobie z bazami danych i procesami ETL. Wiemy, że dotarcie tam było dość długą podróżą. Ta część wyszła nieco poza powszechną wiedzę, aby przedstawić kilka aspektów dostrajania wydajności, które mogą znacznie przyspieszyć procesy ETL. Nie zagłębialiśmy się zbytnio w szczegóły, ale przynajmniej będziesz mógł wykonać pierwszy krok rozwiązywania problemów. Przedstawione tu koncepcje mają stanowić podstawowe wytyczne, ale mogą nie wystarczyć do rozwiązania Twoich problemów. Radzimy sprawdzić w Internecie, forach i innych miejscach wsparcia. Nadal czeka nas część dotycząca planowania i orkiestracji. Ale zanim to połączymy, założymy się, że nie możesz się doczekać, aby zobaczyć, co możemy zrobić z danymi, które już zebraliśmy.

8. Interfejs raportowania BI

Robimy postępy i jeśli postępowałeś zgodnie z krokami instalacji i instrukcjami zawartymi w poprzednich częściach, dotarłeś do zabawnej części implementacji platformy BI. Pod koniec będziemy w stanie analizować informacje na naszej platformie BI w graficzny i intuicyjny sposób. Interfejs raportowania BI jest czasami uważany za rozwiązanie BI, ale bez całej dotychczasowej pracy, którą już wykonaliśmy, trudno byłoby nam coś przeanalizować w naszym narzędziu BI. Pod koniec będziesz gotowy odpowiedzieć na pytania biznesowe, takie jak: które produkty osiągają lepsze wyniki, który z pięciu największych klientów ma lepszy stosunek przychodów netto do sprzedaży brutto, które produkty są ze sobą powiązane pod względem sprzedaży, ile jakie mamy zapasy w naszych magazynach i ile miesięcy obejmujemy tym zapasem na podstawie średniej sprzedaży z poprzedniego roku. Na wszystkie te i inne pytania można odpowiedzieć za pomocą narzędzi BI, jeśli opracujemy wymagane wizualizacje. Ogólne funkcjonalności i koncepcje zostały ocenione w Części 1, więc zrobimy krótkie wprowadzenie i skupimy się na samym projekcie części BI. W tym celu skorzystamy z różnych narzędzi, które pozwolą Ci wybrać, które rozwiązanie lepiej odpowiada Twoim potrzebom. Aby mieć dostępne darmowe lub tanie rozwiązanie dla interfejsu BI, mamy głównie dwie możliwości: skorzystać z narzędzia open source lub skorzystać z bezpłatnej edycji narzędzi komercyjnych. Rozwiązania open source pozwalają na wdrożenie bezpłatnego rozwiązania z gwarancją, że pozostanie ono darmowe w przyszłości, ale z ryzykiem braku wsparcia w przypadku, gdy Twój projekt się rozrośnie i będziesz chciał stworzyć skalowalne rozwiązanie dla dużej liczby użytkownicy. Darmowe edycje komercyjnych rozwiązań pozwolą Ci ewoluować z darmowej edycji do komercyjnej i uzyskać wsparcie od firmy (oczywiście po opłaceniu licencji i cenie wsparcia). Mamy pewne doświadczenie z narzędziami open source, takimi jak rozwiązania Pentaho czy Jaspersoft, i podobnie jak w przypadku głównych narzędzi komercyjnych, ci dwaj dostawcy mają również wersję komercyjną. W rozwiązaniu open source nie masz możliwości raportowania ad hoc; możesz napisać tylko kilka zapytań do uruchomienia przez bazy danych, a następnie wykorzystać wyniki w interfejsie zbliżonym do programowania, co nie jest zbyt intuicyjne dla użytkowników końcowych. Wyjaśnimy więc, jak używać bezpłatnych wersji komercyjnych narzędzi, takich jak Microstrategy Desktop, Power BI i Qlik Sense, jako bezpłatnych wersji platform Microstrategy, Power BI PRO i Qlikview.

Jak wybrać narzędzie BI

Rozmawialiśmy również w Części 1 o podejściach BI i zobaczyliśmy, że masz dostępne strategie Query & Reporting do uruchamiania raportów ad hoc, Information Sharing do dostarczania informacji do całej firmy, Dashboarding, gdzie możesz grać na jednym ekranie z wieloma selektory i panele oraz narzędzia do importu danych i wykrywania danych, które umożliwiają użytkownikom końcowym korzystanie z przyjaznego interfejsu umożliwiającego dodawanie własnych danych oraz badanie trendów i wzorców. Tę klasyfikację można połączyć ze strategią projektu, której używasz w swoim projekcie BI, a po przeprowadzeniu tej analizy będziesz mógł zdecydować, które narzędzie BI jest dla Ciebie lepsze. Aby właściwie podjąć decyzję, musisz również wziąć pod uwagę, czy chcesz skorzystać z bezpłatnego rozwiązania, czy też masz budżet na zakup wersji komercyjnej. Jeśli w tym momencie myślisz, że możesz przejść na projekt pilotażowy, a następnie jesteś przekonany, że będziesz miał budżet na licencje, aby przejść do dużego projektu, to opcja komercyjna z darmowym rozwiązaniem na pilota byłaby dobrą strategią . Nie jest naszym celem sprzedawanie Ci żadnego narzędzia komercyjnego, ale zazwyczaj narzędzia komercyjne są solidniejsze, mają wsparcie, więcej inwestycji w rozwój i innowacje, więc powinno to być coś, co powinieneś rozważyć jako ważną opcję, jeśli jesteś w takiej sytuacji. Twoja strategia projektowa może być klasycznym podejściem ze statycznymi raportami opartymi na ustrukturyzowanej hurtowni danych i skoncentrowanymi na dużych zmianach na poziomie bazy danych, która wymaga wiedzy technicznej do opracowania i wdrożenia wymagań użytkowników, lub

nowymi podejściami do odkrywania danych, które koncentrują się na wizualizacjach i dużej dynamice dla użytkownika końcowego, dając mu autonomię poprzez samodzielne importowanie danych i zabawę różnymi wizualizacjami, aż znajdzie wymagane wzorce. Klasyczne projekty raportowe wymagają również rozwoju infrastruktury wewnątrz narzędzia, co powoduje długotrwały i duży wysiłek w rozwoju projektu, aż użytkownik końcowy zobaczy wyniki. Z drugiej strony istnieją bardziej niezawodne rozwiązania, szczególnie dla dużych środowisk z setkami użytkowników i tysiącami wykonań raportów miesięcznie. Z drugiej strony nowe platformy BI skoncentrowane na automatycznej obsłudze użytkowników pozwalają im badać wgląd w dane, dodając możliwości dochodzenia, które ułatwiają zadanie uzyskiwania odpowiednich informacji z naszych danych, takie jak drążenie, stronicowanie, filtrowanie, wybieranie, i krzyżowanie informacji z wielu źródeł. W całej tej książce analizujemy narzędzia dla małych/średnich scenariuszy, więc spodziewamy się, że być może pomyślisz o projekcie pilotażowym przed przystąpieniem do większego projektu. Dzięki temu zobaczymy narzędzia nastawione na eksplorację danych, które sprawdzą się w tych małych/średnich projektach. Po przeprowadzeniu tego pilotażu będziesz mógł ocenić, czy testowany produkt odpowiada Twoim potrzebom, czy wolisz uzyskać licencję, aby rozwinąć ten pilotaż w wybranym narzędziu, lub czy chcesz ocenić inne platformy przeznaczone dla większych środowisk. Podstawowym narzędziem do klasycznej analizy są raporty i dokumenty, w których użytkownicy mogą zobaczyć czyste dane przefiltrowane, uporządkowane, z możliwością eksportu do Excela, z wykorzystaniem BI jako sourcingu dla Excela. Podstawowym narzędziem do analizy danych Discovery jest dashboard. Chociaż można tworzyć proste raporty, które umożliwiają przeprowadzanie mniej lub bardziej szczegółowych analiz statycznych, pulpit nawigacyjny zawiera wiele dostępnych informacji, ale pokazuje tylko części i agregacje informacji, a następnie umożliwia filtrowanie, segmentowanie, agregowanie lub drążenie w poprzek informacji w celu przeprowadzenia sensownej analizy. Podczas gdy raporty i dokumenty są używane jako narzędzie pośrednie do wyodrębniania i analizowania danych, pulpit nawigacyjny może być wynikiem końcowym w tym sensie, że można go pokazać bezpośrednio na spotkaniu i przeprowadzić analizę, aby pokazać swoje ustalenia w danych. Biorąc to pod uwagę, wygląd i projekt dashboardu mają co najmniej takie samo znaczenie jak dane w nim zawarte. Przeanalizujmy kilka najlepszych praktyk, które pomogą Ci opracować przydatne pulpity nawigacyjne. Nie martw się wyborem. Istnieją rozwiązania, zwłaszcza komercyjne, które mają wszystkie dostępne podejścia, dzięki czemu można wykonać projekt pilotażowy, importując Excele, aby zobaczyć, jak to wygląda, a następnie opracować projekt z całą powiązaną strukturą wewnątrz narzędzia BI, tabelami, atrybutami, wymiarami, fakty, hierarchie, metryki, filtry itp. Ponieważ koncentrujemy się na bezpłatnych lub tanich narzędziach, będziemy mieli ograniczone możliwości, wybierając tylko bezpłatne edycje narzędzi komercyjnych, które koncentrują się na umożliwianiu użytkownikom końcowym odkrywania danych za pomocą funkcji pulpitów nawigacyjnych. Nie będziemy chcieli wchodzić w rozwój przy użyciu narzędzi takich jak Jaspersoft czy Pentaho, ponieważ wymagałoby to obszerniejszej książki. Zobaczmy więc, jak tworzyć ładne i przydatne pulpity nawigacyjne za pomocą narzędzi do odkrywania danych w następnej sekcji.

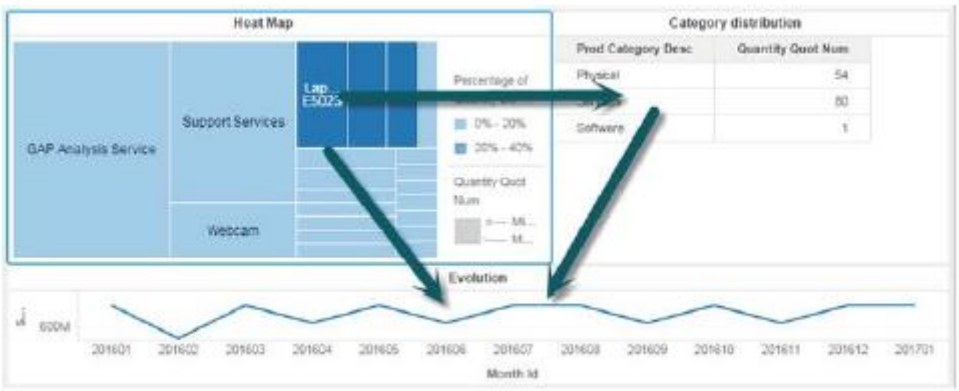
Najlepsze praktyki w Dashboardingu

Chociaż raporty są proste i istnieje niewiele kwestii związanych z projektowaniem, o ile wynikiem jest zwykle tabela lub wykres, podczas tworzenia rozbudowanych dokumentów i pulpitów nawigacyjnych istnieje zestaw zaleceń, które zwykle stosujemy podczas opracowywania. W ramach zaleceń, które przeanalizujemy w tej sekcji, zobaczysz, że niektóre z nich są dość logiczne, a inne wydają się arbitralne. Zwłaszcza ta druga grupa rekomendacji opiera się na naszym doświadczeniu w projektowaniu kokpitów oraz opiniach użytkowników z tych kokpitów. Kokpit musi być intuicyjny, przyjazny dla użytkownika, który nie wymaga żadnych instrukcji obsługi, aby wiedzieć, jak z niego korzystać. Jeśli Twoi użytkownicy potrzebują pomocy w analizie danych w dashboardzie, dzieje się tak dlatego, że

dashboard nie jest wystarczająco przejrzysty. Jeśli zastosujesz się do zestawu poniższych zaleceń, jesteśmy prawie pewni, że uzyskasz pomyślne wyniki i pozytywną opinię od swojego klienta, który będzie końcowym użytkownikiem informacji.

Zaczynając od lewego górnego rogu

Czytamy od lewego górnego rogu do prawej, a następnie do następnego wiersza. Przynajmniej oczekujemy, że przeczytasz tę książkę w ten sposób; inaczej nic z tego nie zrozumiesz. Gdy informacje są wyświetlane w formie graficznej, a ekran zawiera wiele wykresów i siatek danych, zwykle robimy to samo, więc interesujące będzie zlokalizowanie najistotniejszych informacji lub wykresów w lewym górnym rogu ekranu. W ten sam sposób logika analizy powinna być zgodna z tym porządkiem. Jeśli zdefiniujesz wykres, który będzie używany jako selektor reszty pulpitu nawigacyjnego, zaleca się umieszczenie go w lewym górnym rogu, jak pokazano na rysunku



W tym przykładzie obszar wybrany na wykresie Mapa Temperatur będzie filtrował siatkę dystrybucji kategorii i wykres Ewolucji. Również kategoria wybrana w siatce dystrybucji kategorii będzie miała wpływ tylko na wykres ewolucji. Jeśli otworzysz ten dashboard bez żadnej innej instrukcji, trudno będzie ci zrozumieć, że wybierając kategorię możesz zmienić dane wyświetlane na mapie ciepła. Jeśli chcesz tę kategorię, siatka dystrybucji działa jak selektor mapy ciepła, a wykres ewolucji byłby znacznie bardziej intuicyjny, jeśli umieścisz go w lewym górnym rogu.

Łączenie powiązanych informacji

Wewnątrz firmy prawie wszystkie informacje są powiązane w większym lub mniejszym stopniu. Liczba pracowników prawdopodobnie wpłynie na wielkość sprzedaży, zysku lub operacji. Również dane zewnętrzne mogą być powiązane z danymi wewnętrznymi, a populacja regionu będzie miała wpływ na ogólną sprzedaż, jaką nasza firma realizuje w tym regionie. Tak więc próba ustawienia wszystkich powiązanych informacji na jednym pulpicie nawigacyjnym jest w rzeczywistości niemożliwa. W każdym razie musimy spróbować mieć pełną analizę na jednym ekranie zawierającym większość istotnych informacji. Oczywiście nie możesz zobaczyć wszystkich informacji na jednym ekranie, więc spróbujemy użyć selektorów, filtrów, wizualizacji zależnych i drążenia, aby zobaczyć jak najwięcej informacji na jednym ekranie. Jak skomentowaliśmy w części 1, ważne jest, aby wziąć pod uwagę naszą publiczność. Możemy opracować wstępny dashboard z podsumowaniem najważniejszych wskaźników KPI w firmie, które będą udostępniane wszystkim pracownikom, a następnie opracować inne dashboardy, które będą zawierały bardziej konkretne informacje i szczegóły dotyczące pojedynczego obszaru, działu lub zespołu, dodając wymagany poziom szczegółowości w każdym pulpicie nawigacyjnym, który będzie kierowany do jego własnych odbiorców.

Skoncentruj się na istotnych danych

Pojedynczy ekran ma ograniczoną ilość miejsca, dlatego ważne jest, aby koncentrować się tylko na tych KPI, które wnoszą większą wartość dodaną do procesu decyzyjnego. Ważne jest, aby podążać za tymi KPI, które nasze kierownictwo uznało za cele firmy. Opierając się na hipotetycznym celu zwiększenia naszej sprzedaży o 20% w stosunku do roku poprzedniego, będziemy musieli skoncentrować się na pomiarach sprzedaży. Jeśli cel został określony na podstawie ilości, będziemy musieli skoncentrować nasze raportowanie na wielkości sprzedaży, jeśli cel został zdefiniowany na podstawie ilości, będziemy musieli skoncentrować się na ilości, jeśli celem jest ilość, będziemy musieli skupić się na wielkości . Przekreślone informacje mogą być przydatne, ale nieistotne dla analizy, więc można ich uniknąć lub przynajmniej przenieść z początkowego ekranu dashboardu. Jeśli nasze kierownictwo zdefiniowało jako cel obniżenie ogólnych kosztów we wszystkich naszych procesach, musimy skoncentrować się na KPI kosztowych, podzielonych według miejsc powstawania kosztów, elementów kosztów, rodzaju kosztów itp. Jeśli celem naszej firmy jest osiągnięcie udziału w rynku w stosunku do naszego konkurentów, będziemy musieli przeanalizować udział w rynku, nowe produkty na rynku, nowych konkurentów, którzy mogą się pojawić lub czynniki zewnętrzne, które mogą mieć wpływ na udział w rynku, takie jak kwestie regulacyjne. Oprócz skupienia się na odpowiednich danych ważne jest również wykorzystanie znaczących wskaźników KPI. Może być ważne, aby wiedzieć, że mamy 25% udziału w rynku dla danej kategorii produktów, ale będzie to zależeć od udziału głównego konkurenta, więc będziemy musieli zobaczyć porównanie między obydwoma KPI, a nie pojedynczy udział w rynku. Jeśli naszym celem jest obniżenie naszych kosztów, istotne będzie, że obniżyliśmy je poniżej 100 000 USD naszych kosztów globalnych, ale będzie to bardziej istotne, jeśli porównamy to z celem, który ustaliliśmy. Jeśli nasz cel zostanie zwiększony o 20%, nasza sprzedaż będzie adekwatna do rzeczywistego wzrostu sprzedaży w porównaniu z tymi 20% celu.

Zalecenia dotyczące formatowania

Ważne jest jakie dane pokażemy, ale ważne jest też jak to pokażemy. Z biegiem czasu coraz częściej widzimy rolę projektanta graficznego w zespołach programistów BI, który pomaga w projektowaniu pulpitów nawigacyjnych i doświadczeniach użytkowników końcowych oraz upewnia się, że jesteśmy w pełni zgodni z tą strategią. Nie byłby to pierwszy raz, gdy widzimy naprawdę przydatny pulpit nawigacyjny, który pokazujemy użytkownikowi końcowemu, a pierwszy komentarz brzmi: „ok, możemy razem zobaczyć wszystkie wymagane informacje, możemy bawić się informacjami i to będzie być bardzo przydatne dla naszego działu, ale tytuły są w kolorze niebieskim, a nasz kolor korporacyjny jest zielony”. Dlatego chcielibyśmy przedstawić kilka zaleceń dotyczących formatowania. Możesz pomyśleć, że większość poniższych zaleceń jest całkowicie poprawna podczas tworzenia slajdów w programie PowerPoint i miałbyś rację. Ostatecznie dashboard może służyć jako dokument prezentacji, więc w obu przypadkach obowiązują te same zasady.

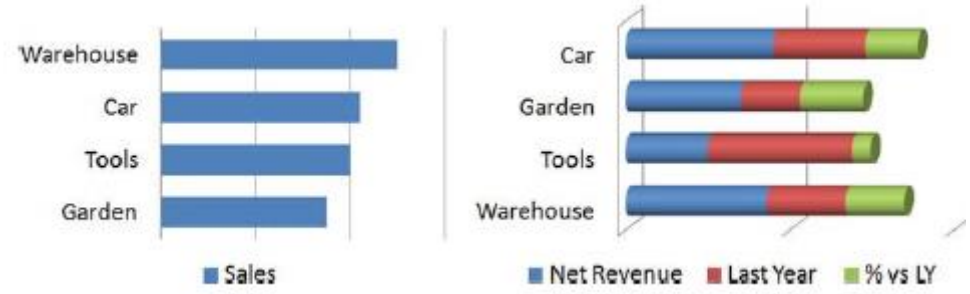
Integracja Korporacyjna

Kierowanie się korporacyjnym stylem projektowania deski rozdzielczej jest najważniejszym elementem formatowania, który musimy wziąć pod uwagę, jeśli chcemy uzyskać pozytywne opinie od naszych klientów wewnętrznych. Jako organizacja będziemy się lepiej identyfikować z dokumentem, ekranem, dashboardem czy programem, jeśli dostrzeżemy w nim obecny styl korporacyjny. Dlatego ważne jest, aby pokazać logo firmy, kolory na dashboardzie są korporacyjne, że używamy korporacyjnej czcionki, korporacyjnych obrazów itp. Jeśli Twoje narzędzie BI na to pozwala, zaleca się również umieszczenie linków do korporacyjnej strony internetowej lub Intranet wewnątrz deski rozdzielczej. Możemy również znaleźć dokumentację dotyczącą korzystania z pulpitu nawigacyjnego w intranecie i zlokalizować wewnętrzne linki do naszego narzędzia BI. Jeśli chodzi o jeden konkretny temat, wybrany rodzaj czcionki, również dodatkową rekomendację, ułatwi czytanie dashboardu, jeśli użyjemy jednego

rodzaju czcionki we wszystkich tekstach, które pojawiają się w środku. Może być konieczna zmiana rozmiaru czcionki, ale wszystkie powinny mieć ten sam typ czcionki.

Powiązane wyrównanie danych

Jak skomentowaliśmy, istnieje kilka zaleceń, które wydają się oczywiste, ale wolimy o nich dyskutować, ponieważ nie byłby to pierwszy raz, kiedy widzimy coś podobnego. Na rysunku 8.2 możesz zobaczyć przykład pulpitu nawigacyjnego z dwoma wykresami umieszczonymi obok siebie, oba na poziomie kategorii.



Ale jak widać, kategorie nie są wyrównane między wykresami. Jest dość oczywiste, że najlepszym sposobem na porównanie obu wykresów jest wyrównanie kategorii, ale w tym przypadku masz wykres posortowany według Sales KPI, a drugi jest posortowany alfabetycznie, więc trudno jest zobaczyć zależność między sprzedażą a przychodami netto dla kategorii; Twój widok musi poruszać się w górę i w dół między wykresami. Powinieneś starać się zachować wyrównanie zarówno w poziomie, jak iw pionie, jeśli szczegóły na wykresach i siatkach są podobne. Oczywiście należy starać się zachować wyrównanie wielkości obiektów i pozycji, a czasami nie jest to łatwe, ponieważ relacja między ramą a figurą wewnętrzną nie jest taka sama. Dla tego wyrównania ważne jest, aby narzędzie BI, którego używasz, umożliwiało przesuwanie obiektów w tym samym czasie, gdy widzisz dane, ponieważ możesz zobaczyć, jak wygląda końcowy wynik. Tego rodzaju technologie są zwykle nazywane WYSIWYG, akronimem What You See Is What You Get, ponieważ końcowy wynik jest taki sam, jak podczas edycji.

Formatowanie warunkowe

Jak skomentowaliśmy w poprzednich sekcjach, ważne jest, aby skupić się tylko na odpowiednich danych i wykorzystaniu odpowiednich KPI, które pozwalają nam porównać rzeczywiste dane z naszym budżetem. Ale co z możliwością użycia formatowania warunkowego, aby zauważyć, że niektóre dane są bardziej istotne niż pozostałe? W większości narzędzi BI istnieje możliwość zastosowania formatowania warunkowego, aby zauważyć, że wartości, które są bardziej istotne niż pozostałe, możemy na przykład zaznaczyć na zielono, które komórki osiągnęły cel, na pomarańczowo te, które spełniają 80% celu, a na czerwono pozostałe. Formatowanie warunkowe ułatwi proces podejmowania decyzji, pomagając określić, gdzie należy działać. Znajdziesz go pod różnymi nazwami w różnych narzędziach BI, progach, alertach lub formatowaniu warunkowym, ale jest to coś, z czego zazwyczaj możesz korzystać. Masz przykład użycia na rysunku


Category distribution

Prod Category Desc	Quantity Quot Num
Physical	54
Services	80
Software	1

Intensywność a kolor

Poprzedni przykład wyraźnie pokazuje odwrotny przykład dotyczący tego, co chcemy teraz skomentować. Ta kombinacja kolorów, żółta, jasnozielona i czerwona, jest zbyt ostra i może również wpływać na rzeczywiste zrozumienie danych poniżej tych kolorów. Jest całkiem możliwe, że teraz znajdziesz inny problem, używając innych kolorów; jeśli pulpit nawigacyjny lub dokument jest wydrukowany w czerni i bieli, rozróżnienie, które komórki są odpowiednie, może być trudne, jeśli nie mieć część o różnych kolorach i różnych odcieniach. Ogólnie zaleca się więc używanie różnych tonów tego samego koloru, jak widać na rysunku 1 na wykresie mapy cieplnej, gdzie używamy różnych tonów niebieskiego (wolimy określić, ponieważ jest całkiem możliwe, że czytając to czarno na białym) zamiast używać różnych kolorów, jak widać na rysunku 3.

Widoczność danych

Kolejną rekomendacją, powiązaną z poprzednią, jest stosowanie kontrastu między wykresami, siatkami, tekstami i komórkami zawierającymi informacje, ramki i tło w celu uzyskania odpowiedniej widoczności danych. Staraj się nie używać nadmiaru struktur ramowych, prostokątów, linii, figur i obrazów, które mogą przeszkadzać użytkownikowi i powodować utratę uwagi na ważnych danych. Również wykresy 3D, cienie lub efekty 3D mogą poprawić jakość graficzną raportu, ale mogą wpływać na analizę danych, ukrywając dane.

Zastosowania graficzne

Wewnątrz najlepszych praktyk chcielibyśmy otworzyć specjalną sekcję, aby porozmawiać o tym, który typ wykresu lepiej pasuje do każdej analizy, o ile zobaczysz, że każdy zestaw danych ma jakiś typ wykresu, który pasuje lepiej niż inne. Jako ogólne zalecenia dotyczące korzystania z wykresów , rozważymy następujące zalecenia:

* Ogranicz użycie wykresów kołowych, ponieważ są one trudne do analizy, tylko w niektórych szczególnych przypadkach mogą być przydatne.

* Staraj się używać wykresów, które efektywnie wykorzystują dostępną przestrzeń, ograniczając użycie legendy, osi lub etykiet danych do tych wykresów, które naprawdę wymagają ich użycia, aby zrozumieć wykres.

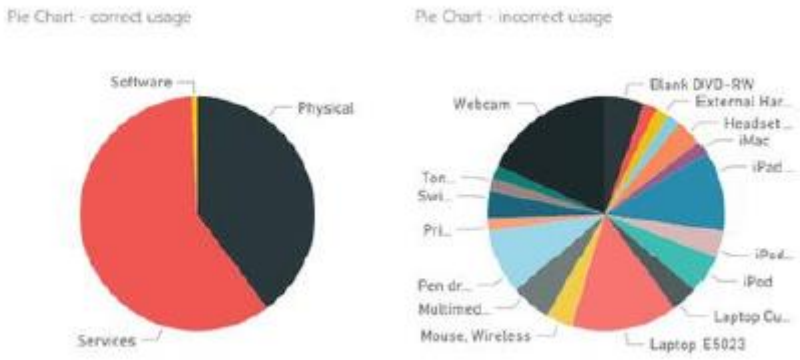
* W przypadku używania etykiet danych staraj się używać ich bez miejsc dziesiętnych lub ograniczaj je do minimum, a jeśli pokazujesz tysiące lub miliony, spróbuj używać notacji K i M (10K zamiast 10 000, 5M zamiast 5 000 000).

* Używaj efektywnych wykresów. Jeśli ze względu na charakter danych najbardziej efektywnym wykresem dla wszystkich wymaganych informacji jest wykres słupkowy, użyj czterech wykresów słupkowych. Nie próbuj urozmaicać typów wykresów pokazanych tylko po to, by urozmaicić.

Przeanalizujmy również niektóre przypadki użycia wykresów, które mogą pomóc w zdefiniowaniu Twojego pulpitu nawigacyjnego.

Wykres kołowy — procentowy udział dla kilku wartości

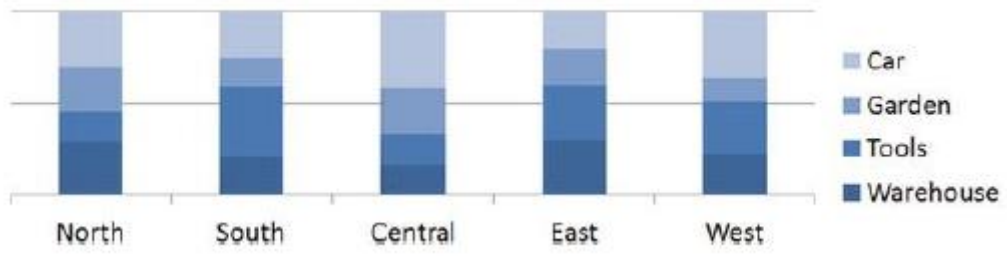
Wykorzystanie wykresu kołowego, jak skomentowano przed chwilą, powinno być ograniczone do pewnych warunków. Musi być ukierunkowana na analizę udziału każdej wartości w ogólnej kwocie (analiza procentowa) dla niewielkiej liczby wartości i jest szczególnie przydatna, jeśli chcesz pokazać, że jedna z wartości jest znacznie wyższa niż pozostałe. Na rysunku możesz zobaczyć podwójny przykład użycia wykresu kołowego, jeden według kategorii produktów, a drugi według produktów.



W pierwszym z nich można łatwo rozróżnić, która kategoria (Usługi) jest bardziej odpowiednia i jak bardzo jest trafna (około 60%), podczas gdy w drugiej naprawdę trudno jest porównać różne produkty, która jest bardziej odpowiednia. Możesz postrzegać kamerę internetową jako najlepiej sprzedający się produkt, ale i tak trudno ją porównać z laptopem E5023, który wydaje się być drugim, a jeśli spróbujesz zlokalizować trzeci, czwarty i piąty, wydają się być naprawdę trudne do zidentyfikowania.

Skumulowany Wykres Słupkowy - Porównanie Procentowe

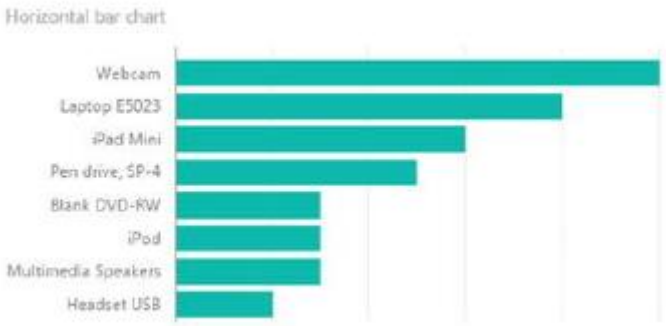
Jeśli chcesz zobaczyć procent sumy, gdy masz wiele wartości i sprawdzić to w wielu kategoriach, dobrą opcją jest skumulowany wykres słupkowy bezwzględny. Ten wykres będzie miał zawsze tę samą wysokość, a następnie sekcje pośrednie będą się różnić w zależności od metryki, jak widać na rysunku



W tym przykładzie widzimy procent sprzedaży dla każdej kategorii w regionach naszych klientów. Łatwiej jest rozróżnić, która kategoria jest najbardziej odpowiednia w każdym regionie i jak zmienia udział kategorii w różnych regionach. Należy wziąć pod uwagę na tym wykresie, że widzimy ogólny rozkład według regionów, więc dopasowujemy się do 100% sumy wszystkich kategorii. Musisz jasno powiedzieć, że nie są to wartości bezwzględne; widać, że samochody w regionie centralnym biorą więcej niż w południowym, ale dotyczy to tylko procentu, może w wartościach bezwzględnych południe ma wyższą sprzedaż niż centralny.

Słupki poziome - porównanie górnych N elementów

Innym typowym wykresem, który możemy zobaczyć w wielu dokumentach, jest poziomy wykres słupkowy. Istnieje kilka przypadków użycia, w których zalecamy korzystanie z tego wykresu, a jednym z nich jest porównanie pierwszych N elementów. Aby łatwo zobaczyć, które najlepsze elementy są skuteczniejsze dla danego atrybutu, zdecydowanie zaleca się posortowanie wykresu według wartości metrycznych, tak aby pierwszy słupek był powiązany z najlepiej sprzedającym się produktem, jak widać na rysunku.



Tutaj możesz łatwo zobaczyć, że kamera internetowa jest najlepiej sprzedającym się produktem, że drugi to Laptop E5023, trzeci iPad Mini itp. O wiele łatwiej to zobaczyć niż na wcześniejszym rysunku z wykresem kołowym.

Pionowe linie, słupki lub obszary – ewolucja w czasie

Zwykle przedstawiamy pojęcie czasu jako linię poziomą, biegnącą od lewej do prawej. Tak więc na każdym wykresie ewolucji, który chcesz pokazać ewolucję określonego kluczowego wskaźnika wydajności w czasie, musisz użyć pionowych słupków, obszarów lub linii, które pokazują postęp czasu w poziomie, przy czym pionowy słupek obok drugiego pokazuje, jak ewoluuje. Chociaż słupki mogą być również używane do porównywania innego typu atrybutów, nieokresowe, linie i obszary są powszechnie używane tylko do analizy czasu. Zwykle wiążemy linię z ewolucją, więc posiadanie wykresu liniowego do porównywania tylko kategorii lub elementów nie jest poprawnie rozumiane przez użytkowników końcowych. Zalecamy również używanie połączonych linii i słupków podczas analizy wielu wskaźników. Zwykłym wykresem, którego używaliśmy w naszych opracowaniach, jest użycie słupka do analizy rzeczywistych danych, a następnie użycie linii do zaznaczenia celu lub odchylenia w odniesieniu do poprzedniego roku. Ten ostatni przykład to ten pokazany na rysunku 8.7, gdzie można zobaczyć sprzedaż w bieżącym roku, sprzedaż w poprzednim roku oraz stosunek jednego do drugiego.

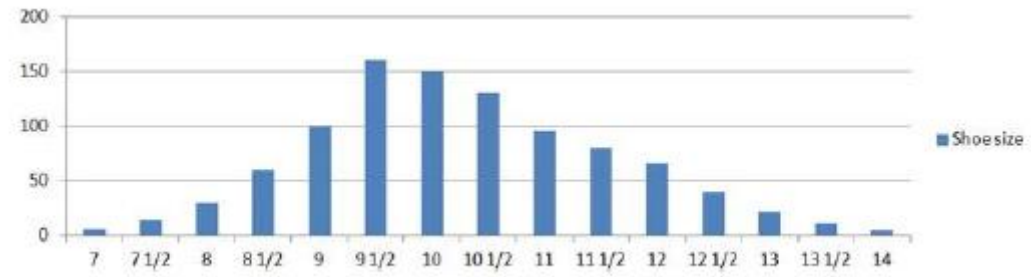
Pionowe słupki - histogram

Za pomocą histogramu próbujemy przeanalizować rozproszenie danych dla metryki, próbując zobaczyć, jaką częstotliwość powtórzeń wartości metryki otrzymujemy. Dzięki analizie histogramu możemy zobaczyć, która wartość pojawia się najczęściej i jaki jest rozrzut tej wartości. Najlepszą reprezentacją analizy histogramu jest pionowy wykres słupkowy. Istnieje wiele przykładów analizy histogramu:

* Analiza kwalifikacji na egzaminie/teście

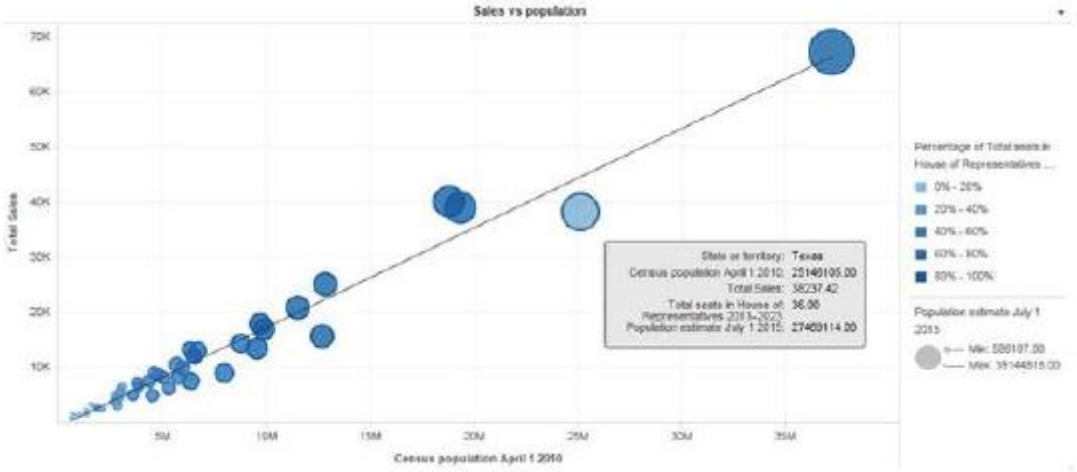
* Analiza średniego wskaźnika zanieczyszczenia w mieście

* Analiza najczęściej używanego rozmiaru buta według modelu buta



Wykres bąbelkowy - Korelacja

Wykres bąbelkowy umożliwia porównanie wielu metryk między nimi; w zależności od możliwości narzędzia, pozwoli porównać do czterech metryk, aby zobaczyć korelację między nimi. Za pomocą koloru można porównać zależność między pozycją pionową i poziomą, zależność między pozycją a rozmiarem oraz zależność między nimi wszystkimi. Na rysunku 8.9 próbujemy przeanalizować sprzedaż we wszystkich stanach USA z populacją każdego stanu i liczbą miejsc w Izbie Reprezentantów.



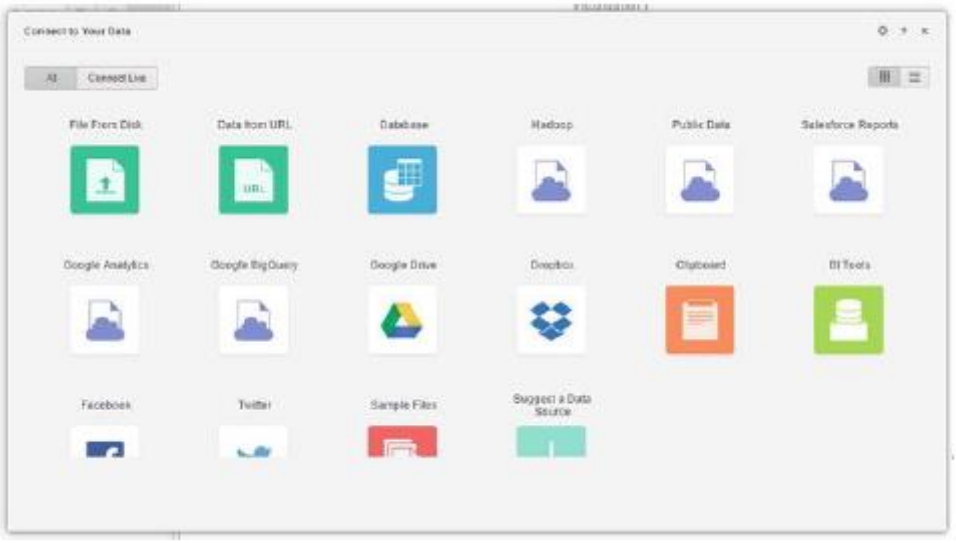
W tym przykładzie widzimy w pozycji poziomej populację każdego stanu, w pozycji pionowej całkowitą sprzedaż w stanie, pokolorowaną według liczby miejsc w Izbie Reprezentantów i wielkości według przewidywanej liczby ludności w ciągu najbliższych pięciu lat. Jak widać, nasza sprzedaż jest dość powiązana z populacją, ponieważ wydaje się, że podąża ona rosnącą linią. Że bąbelki powyżej trendu sprzedają więcej na mieszkańca, te poniżej trendu mają mniejszą sprzedaż na mieszkańca. Również narzędzie BI użyte w tym przykładzie (Microstrategy Desktop) pozwala nam zobaczyć podpowiedź ze szczegółami danych i umieściliśmy kursor myszy nad Teksasem, który wydaje się być poniżej trendu, aby zobaczyć powiązane dane dla bańki.

Narzędzia BI

Do tej pory widzieliśmy pewne teorie na temat tworzenia kokpitów menedżerskich, które wykresy lepiej pasują do niektórych przykładów, które, jak oczekujemy, można ekstrapolować na potrzeby wewnątrz firmy. Chcielibyśmy teraz przyjrzeć się trzem narzędziom: Microstrategy, Power BI i Qlik Sense i jak je zainstalować oraz podstawowym instrukcjom, aby uzyskać z nich jakiś dashboard.

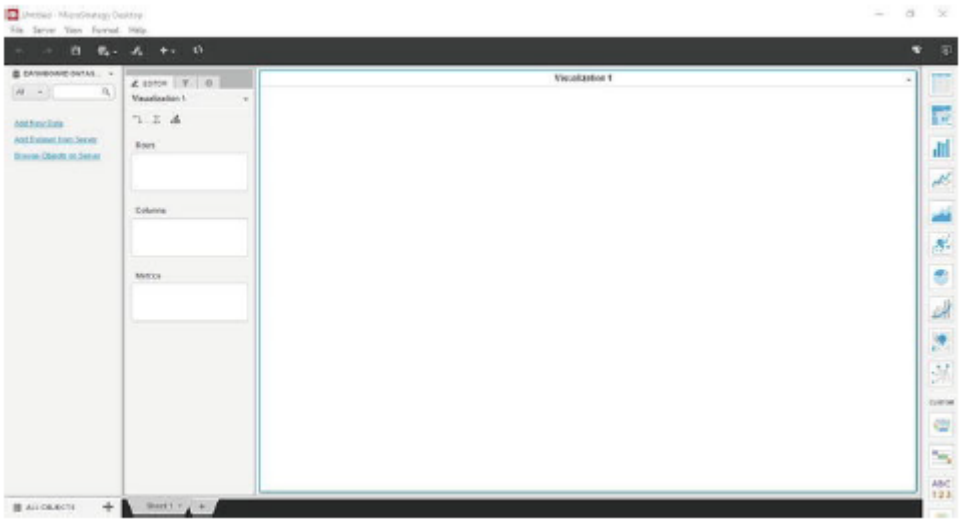
Pulpit Microstrategy

Platforma Microstrategy to jedno z tradycyjnych narzędzi BI, które możemy znaleźć na rynku od ponad 20 lat. Jest to dojrzała technologia, która w wersji Enterprise zawiera wiele komponentów, zaczynając od Intelligence Server, rdzenia procesu; Serwer WWW, z którym łączy się większość użytkowników; Mobile Server, który zapewnia obsługę aplikacji mobilnych na smartfony i tablety; Narrowcast Server i usługi dystrybucji w celu dostarczania informacji w wiadomościach e-mail, drukarkach, udostępnionych zasobach; Report Services, które zapewniają możliwość definiowania zaawansowanych dokumentów i pulpitów nawigacyjnych za pomocą technologii Pixel Perfect; oraz usługi o wysokiej interaktywności i OLAP, które zapewniają funkcje w pamięci w celu poprawy wydajności systemu. Proces projektowania to scenariusz korporacyjny, który może być znacznie bardziej złożony niż ten, który wyjaśnimy dla Microstrategy Desktop, o ile jest to tylko mały element całej architektury, ale wszystkie wyjaśnienia, które są napisane poniżej, są w pełni poprawne dla pulpitów nawigacyjnych należących do Report Services wewnątrz platformy. Jak skomentowano, Microstrategy ma również darmową edycję o nazwie Microstrategy Desktop, która zapewnia część interaktywności Report Services obsługiwanej przez niektóre funkcje Intelligence Server. Jest to instalowalna aplikacja na samodzielne komputery, która umożliwia łączenie się z bazami danych; importować pliki płaskie; połączyć się z plikami sieciowymi; łączyć się z silnikami Big Data; otwórz wiele źródeł, takich jak Facebook lub Twitter; i importować pliki z udostępnionych platform, takich jak Dropbox, Dysk Google, w tym importować dane ze schowka systemu Windows, jak widać na rysunku, który pokazuje ekran początkowy podczas dodawania danych do pulpitu nawigacyjnego.



Ale przed użyciem musisz zainstalować oprogramowanie na swoim komputerze. Aby zainstalować Microstrategy Desktop, pierwszym krokiem, jak możesz sobie wyobrazić, jest pobranie pakietu instalacyjnego ze strony Microstrategy. Znajdziesz go na stronie https://www.microstrategy.com/us/desktop, gdzie po wypełnieniu małego formularza z imieniem i nazwiskiem, firmą oraz e-mailem będziesz mógł pobrać pakiet instalacyjny. Możesz wybrać wersje Windows lub Mac do zainstalowania; oczywiście będzie to zależeć od komputera, na którym będziesz z nim pracować. Po pobraniu wystarczy rozpakować pobrany plik i uruchomić plik instalacyjny o nazwie MicrostrategyDesktop-64bit.exe. Możesz go zainstalować zachowując domyślne opcje instalacji (cóż, oczywiście możesz zlokalizować instalację gdziekolwiek chcesz), a po instalacji będziesz mógł otworzyć narzędzie. Przy pierwszym otwarciu zobaczysz stronę wprowadzającą z linkami do filmów i różnych

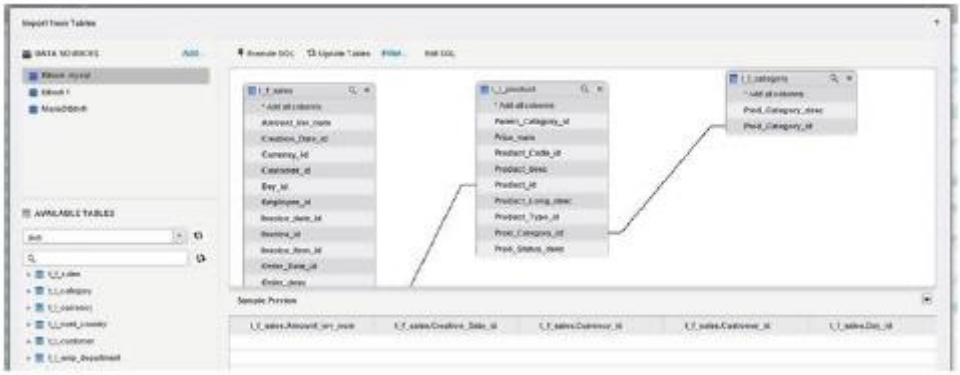
stron pomocy, ale na górze zobaczysz link do Utwórz pulpit nawigacyjny. Jeśli zaznaczysz, że nie chcesz więcej widzieć wskazówek, następnym razem, gdy otworzysz Pulpit, zostaniesz przekierowany do Dashboard Editor, który wygląda jak na rysunku.



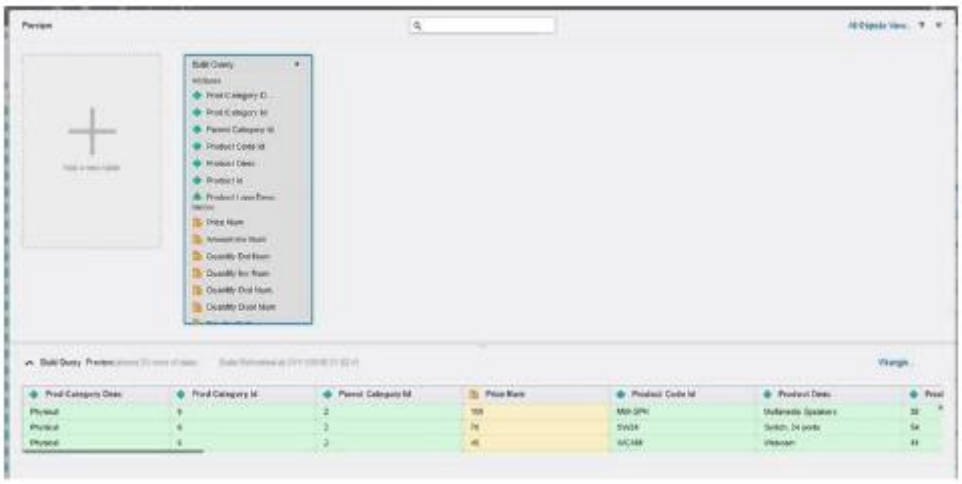
W tym edytorze możemy znaleźć różne obszary do pracy. W górnej części edytora mamy menu główne umożliwiające dostęp do różnych opcji. Pierwszy panel po lewej stronie to miejsce, w którym będziemy mieć nasze źródła danych do wykorzystania w kokpicie, obecnie pustym, dopóki nie wybierzemy żadnego zestawu danych. Drugi panel zawiera edytor wizualizacji, edytor filtrów oraz opcje konfiguracji dla każdej wizualizacji. Główny panel, który pojawia się z tytułem Wizualizacja 1, to panel wizualizacji, w którym możemy zlokalizować jedną lub więcej wizualizacji. Wreszcie po prawej stronie mamy selektor typu wizualizacji, aby wybrać wizualizację, której chcemy użyć w panelu wizualizacji. Aby przeglądać informacje z naszej bazy danych, będziemy potrzebować przynajmniej utworzonego ODBC, które łączy się z bazą danych. Być może już go utworzyłeś, jeśli uzyskujesz dostęp do bazy danych z komputera, na którym zainstalowałeś narzędzie Microstrategy Desktop, jeśli nie, musisz go utworzyć za pomocą odpowiedniego sterownika ODBC (może być wymagana instalacja klienta bazy danych). W naszym przypadku użyjemy naszego już zainstalowanego sterownika ODBC, aby uzyskać dostęp do tabel z użytkownikiem tylko do odczytu. Możesz także połączyć się za pomocą połączenia bez ODBC, ale na koniec dnia oznacza to, że nie będziesz mieć dostępnego ODBC do innych celów, ODBC jest wbudowany w Microstrategy, ale i tak musisz określić bibliotekę ODBC. Sugerujemy, że zawsze możesz użyć MySQL ODBC do połączenia z MariaDB i odwrotnie, więc jeśli masz zainstalowany jeden z nich, powinien działać, aby połączyć się z inną bazą danych.

Zaleca się, aby użytkownik tylko do odczytu pobierał informacje z bazy danych z dostępem do wszystkich tabel, ale tylko z uprawnieniami do odczytu, aby mieć pewność, że w przypadku różnych zespołów zespół narzędzia BI nie zmodyfikuje żadnego obiektu w bazie danych które mogłyby wpłynąć na odpowiedzialność procesu ETL.

Podczas dodawania nowych danych pochodzących z bazy danych Microstrategy Desktop zaoferuje trzy możliwości: budowanie nowego zapytania, pisanie nowego zapytania lub wybór tabel. Za pomocą pierwszej opcji, pokazanej na rysunku, możesz utworzyć zapytanie, wybierając tabele, definiując łączenia między nimi, a następnie wybierając pola każdej tabeli, które chcesz dodać do zbioru danych, a na koniec definiując jeśli istnieje jakakolwiek funkcja do zastosowania w jakimś polu.



Po zakończeniu możesz przejść do trybu SQL, aby zobaczyć, które zapytanie SQL zostanie uruchomione w bazie danych, która jest tym samym edytorem SQL, który otwiera się bezpośrednio po wybraniu drugiej opcji. Po wybraniu tabel, które Cię interesują, będziesz mógł je opublikować w obecnej postaci lub kliknąć przycisk Przygotuj dane. Jak widać na rysunku,



Microstrategy Desktop automatycznie identyfikuje na podstawie typu danych, które pola mogą być atrybutami do analizy danych i które z nich mogą być metrykami do analizy. Jeśli niektóre z nich zostały nieprawidłowo przypisane, możesz to zmienić, przeciągając i upuszczając z Atrybutów do Metryk lub używając prawego przycisku myszy, aby to zmienić. Również na tym ekranie prawym przyciskiem masz dostęp do bardzo rozbudowanej konfiguracji, w której możesz zdefiniować wiele funkcji dla niektórych typów znanych pól. W przypadku pola daty możesz zdefiniować całą hierarchię czasową, Rok, Miesiąc, Miesiąc w roku, Tydzień itp. W przypadku atrybutów geograficznych takich jak Kraj, Miasto lub Region możesz zdefiniować je jako geograficzne, co pozwoli narysować to na mapie. Po zdefiniowaniu wszystkich typów kolumn możesz następnie zaimportować dane, aby móc analizować informacje odłączone od bazy danych za pomocą opcji Importuj jako zestaw danych w pamięci lub możesz zdefiniować tabelę jako łącze, aby za każdym razem, gdy otworzysz Microstrategy Desktop, będzie ponownie uruchom zapytanie odświeżając dane z bazy za pomocą opcji Connect Live.

Za pomocą przycisku Wrangle na ekranie Przygotuj dane możesz dodawać funkcje, przekształcenia i modyfikacje do ładowanych danych, aby dostosować informacje źródłowe do swoich potrzeb w przypadku, gdy dane źródłowe mają jakieś brakujące wymagania. Możesz dodawać nowe kolumny pochodne, łączyć kolumny lub ignorować niektóre kolumny, jeśli nie chcesz umieszczać zbyt wielu informacji w pamięci.

Możliwe, że masz wszystkie informacje w jednym zbiorze danych, ale możliwe jest również, że potrzebujesz połączenia z więcej niż jednym zbiorem danych. W takim przypadku będziesz musiał również zdefiniować, które atrybuty będą definiować relacje między zbiorami danych, po prostu klikając prawym przyciskiem myszy odpowiedni atrybut i wybierając opcję powiązania go z innym zbiorem danych. Możesz także połączyć wiele zestawów danych na ekranie Przygotuj dane, klikając duży przycisk po lewej stronie ekranu Dodaj nową tabelę tabeli, jak pokazano na powyższym rysunku. Każdy zestaw danych może dostarczać z różnych źródeł, ale należy zachować ostrożność, ponieważ wartości atrybutów muszą być zgodne z różnych źródeł, aby uzyskać logiczne wyniki. Możesz także skonfigurować typ łączenia i określić, które zbiory danych są podstawowe, a które drugorzędne; w ten sposób można modyfikować wyniki kombinacji wielu zestawów danych. Jeśli wszystkie z nich są podstawowe, pojawią się wszystkie możliwe kombinacje. Wyobraź sobie, że masz zestaw danych zawierający dane dotyczące sprzedaży, w tym atrybut kraju, oraz inny, który zawiera tylko kraj i populację, aby porównać sprzedaż w przeliczeniu na liczbę osób. Jeśli oba są podstawowe, możesz mieć populację Ugandy, ale nie sprzedajesz w Ugandzie, więc może po prostu chcesz ustawić zbiór danych populacji jako drugorzędny, aby wyświetlać tylko dane dla tych krajów, w których masz dane dotyczące sprzedaży. Po wybraniu wszystkich danych źródłowych potrzebnych do naszej analizy możemy przystąpić do samej analizy. Analiza odbywa się za pomocą wizualizacji. Możesz skorzystać z szerokiego zestawu wizualizacji specjalnie połączonych z opcjami, które masz w zakładce konfiguracji:

Siatka: Pojedyncza tabela z wierszami i kolumnami do analizy danych w formacie tekstowym.

Mapa cieplna: obszar wizualizacji jest podzielony na kwadraty według atrybutu, którego rozmiar jest oparty na metryce, a kolor jest oparty na innej metryce.

Wykres słupkowy: Mogą to być słupki poziome lub pionowe, bezwzględne lub skumulowane, przedstawiające jako wysokość wartość metryczną, podzieloną przez jakiś atrybut z możliwością zmiany szerokości i koloru kolumny za pomocą wartości metrycznych.

Wykres liniowy: Masz podobne możliwości jak słupki, również z możliwością zmiany znaczników.

Wykres warstwowy: podobny do wykresu słupkowego, ale zamiast słupków pokazuje obszary.

Wykres bąbelkowy: możesz porównać do czterech danych podzielonych według co najmniej jednego atrybutu. Jedna metryka określa położenie poziome, następna pionowa, inna metryka pokazuje rozmiar bąbelka, a kolejna metryka kolor.

Wykres kołowy: możesz zdefiniować kąt za pomocą metryki, koloru, partycji, segmentacji poziomej i pionowej na podstawie atrybutów, a także zmienić rozmiar w przypadku zwrócenia wielu wykresów kołowych.

Wykres kombi: Kombinacja słupków, linii i obszarów w formacie poziomym lub pionowym.

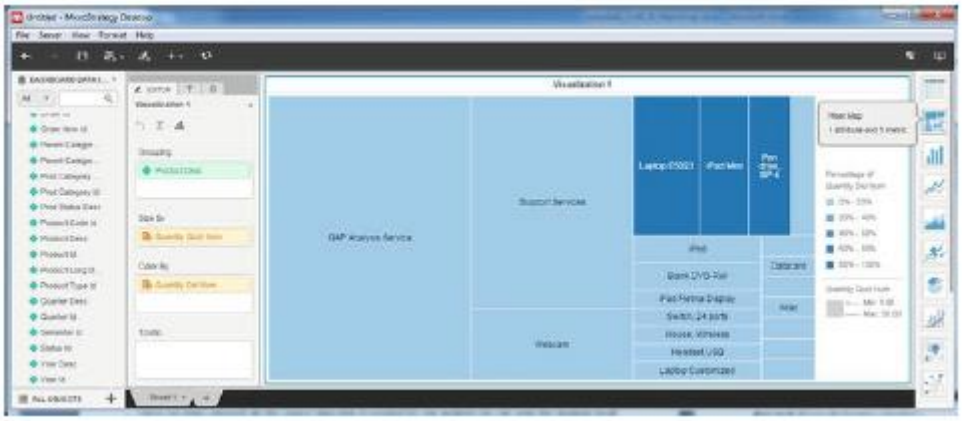
Mapa: Geograficzna lokalizacja metryk na podstawie atrybutów geograficznych, takich jak kraj, region, miasto lub kod pocztowy, do rysowania obszarów, a także przy użyciu szerokości i długości geograficznej do lokalizowania znaczników.

Sieć: Przydatna do zobaczenia relacji między dwoma atrybutami, typowym przykładem jest liczba lotów między lotniskami; możesz zdefiniować atrybuty źródłowe i docelowe, a następnie zastosować kolor i rozmiar na podstawie metryk.

Niestandardowe: Wreszcie masz możliwość dodawania nowych zaawansowanych wizualizacji ze stron open source, które mają dostępne setki niestandardowych wizualizacji, takich jak te dostępne w:

https://github.com/d3/d3/wiki/Gallery

Każda wizualizacja ma swoją własną specyfikację działania, minimalną liczbę atrybutów i metryk, które pozwolą wizualizacji działać. Na rysunku pokazujemy przykład wizualizacji mapy cieplnej.



Po umieszczeniu wskaźnika myszy nad każdym przyciskiem wizualizacji pojawi się podpowiedź z minimalnymi wymaganiami, jak widać na rysunku 8-14. Aby użyć mapy cieplnej, będziesz potrzebować co najmniej jednego atrybutu i jednej metryki. Na tym samym rysunku widzimy panel Edytora, w którym można przeciągać i upuszczać różne atrybuty Grupowania, które określają liczbę obszarów w wizualizacji, w obszarze Rozmiar według można przeciągać i upuszczać niektóre dane, które określają rozmiar każdego możesz użyć opcji Koloruj według, aby zdefiniować podstawę do pokolorowania każdego obszaru, a w obszarze Etykietki narzędzi możesz przeciągać i upuszczać atrybuty i metryki, które będą wyświetlane po umieszczeniu wskaźnika myszy nad jednym z obszarów.

Jak skomentowaliśmy, każda wizualizacja ma swoją własną charakterystykę, więc obszary otwierane w zakładce konfiguracji będą w każdym przypadku inne. Ponieważ w pozostałych częściach nie jest to dogłębna analiza tego, jak korzystać z jednego narzędzia, więc nie zobaczymy szczegółów wymaganych przez każdą wizualizację i wszystkich opcji, których można użyć w Microstrategy desktop, ale spróbujmy podsumować główne czynności, które możesz wykonać:

Wiele wizualizacji w jednym arkuszu: Możesz dodać nowe wizualizacje i domyślnie podzieli obszar wizualizacji na dwie części. Następnie możesz przeciągnąć i upuścić wizualizację, aby podzielić ją w poziomie lub w pionie, i dostosować rozmiar wizualizacji.

Połącz wizualizacje: Możesz użyć wizualizacji do filtrowania danych w pozostałych wizualizacjach, więc kliknięcie jednego obszaru lub komórki jednego wykresu spowoduje odfiltrowanie informacji widocznych w pozostałych wizualizacjach. Aby to zrobić, kliknij strzałkę znajdującą się w prawym górnym rogu wizualizacji i wybierz opcję Użyj jako filtru dla tej wizualizacji, która ma działać jako filtr pozostałych.

Dodaj warunki filtrowania: Warunki filtrowania będą miały zastosowanie do wszystkich wizualizacji na stronie; możesz filtrować według metryk lub atrybutów, definiując również sposób ich interakcji, filtrowania, wykluczania lub komentowania.

Dodaj selektory : Aby przefiltrować wyświetlane informacje, możesz dodać selektory w różnych formatach, pasku przycisków, pasku łączy, liście pól wyboru, slajdach itp. W takim przypadku możesz określić, na którą wizualizację docelową będzie miał wpływ filtr .

Użyj obrazów i tekstu : Możesz także łączyć obrazy i teksty, aby dodawać tytuły, komentarze, wyjaśnienia, łącza i logo.

Wiele stron: Możesz dodać nowe arkusze danych, klikając + arkusz pokazany na dole rysunku.

Formatowanie: Możesz zmienić kolory, linie, czcionkę i rozmiar liter oraz wiele innych opcji, aby dostosować kokpit do kolorów korporacyjnych.

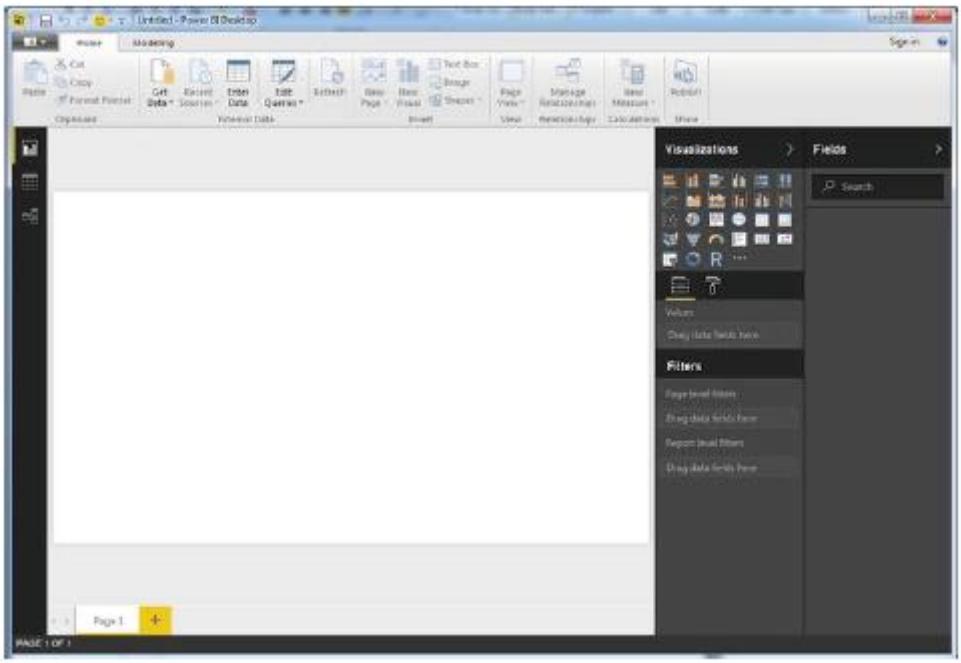
Progi: w większości wizualizacji można zastosować formatowanie warunkowe do komórek, map, słupków i znaczników.

Wiercenie : Możesz zdefiniować drążenie, jeśli twoje atrybuty zostały zdefiniowane jako jeden nadrzędny drugiego (na przykład miesiąc jako rodzic dnia).

Twórz obliczenia pochodne : możesz dodawać formuły, które nie znajdują się bezpośrednio w danych, zliczając elementy atrybutów lub definiując dowolne inne formuły pochodne.

Microsoft Power BI

Microsoft jest najbardziej rozbudowanym dostawcą oprogramowania, więc jest całkiem możliwe, że w Twojej organizacji masz już zainstalowane oprogramowanie Microsoft i całkiem możliwe, że Twój własny komputer korzysta z jakiejś wersji systemu Windows, więc jesteśmy prawie pewni, że nie jest wymagane, abyśmy Cię zapoznali Microsoftu. Ale może nie wiesz, że Microsoft oferuje bezpłatne narzędzie BI do tworzenia raportów i pulpitów nawigacyjnych o nazwie Power BI. Do dyspozycji masz również Power BI PRO, który oferuje Ci możliwość pracy w środowisku współdzielonym, natomiast w darmowej edycji będziesz pracować na swoim komputerze z wersją Power BI Desktop. Wersja PRO zapewni Ci również lepszą wydajność w obciążeniu pulpitu nawigacyjnego i większą przestrzeń danych na użytkownika. Aby go zainstalować, możesz uzyskać dostęp do https://powerbi.microsoft.com/en-us/desktop/ i pobrać pakiet instalacyjny. Na tej stronie znajdziesz również linki do samouczków, które zapewnią Ci głębszą wiedzę na temat korzystania z narzędzia. Kontynuowanie samej instalacji jest dość proste, klikając przycisk Dalej, a na koniec można bezpośrednio otworzyć narzędzie uzyskując dostęp do ekranu, który powinien wyglądać jak na rysunku

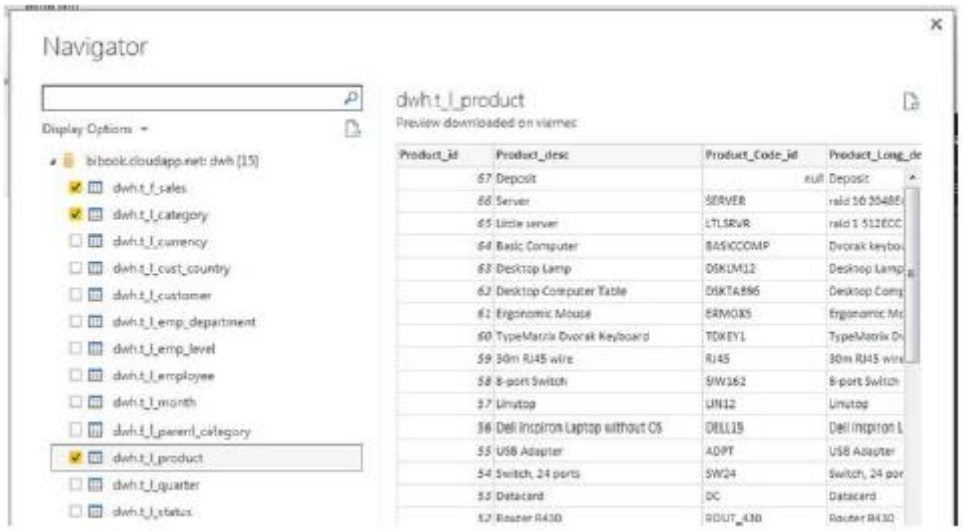


W tym panelu widać przycisk i obszar menu w górnej części ekranu, pionowy pasek przycisków linków po lewej stronie, obszar wizualizacji jako największy pusty obszar, a następnie panel wizualizacji i panel pola. Poniżej znajduje się również selektor stron, w którym można dodać więcej niż jedną stronę, jeśli

jest to wymagane do analizy. Pierwszym krokiem, wspólnym dla wszystkich narzędzi, jest wybranie danych, które chcemy uwzględnić w naszej analizie. W tym celu musimy skorzystać z opcji Pobierz dane. Ponieważ będziemy łączyć się z naszą bazą danych MariaDB, nie zobaczysz żadnej opcji bezpośrednio na liście źródeł danych. Ale jeśli przejdziesz do Więcej, otworzy się nowe menu. W tym menu nie możemy znaleźć opcji MariaDB, ale zgodnie z komentarzem możemy użyć sterownika MySQL, a ten jest dostępny na liście. Jeśli spróbujemy się połączyć, pojawi się ostrzeżenie, ponieważ sterownik NET dla MySQL jest niedostępny, jak pokazano na rysunku .

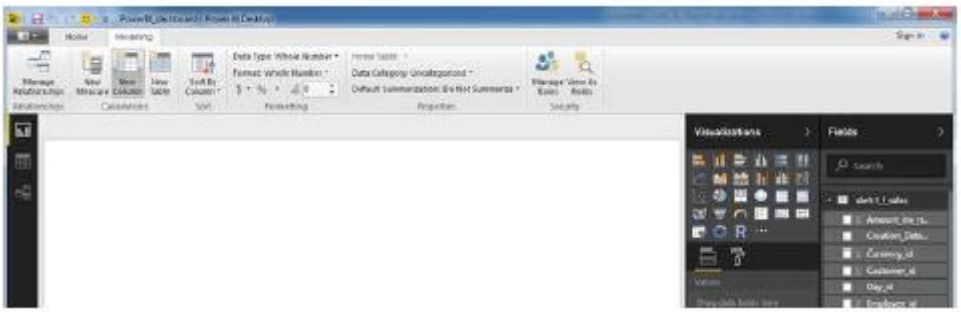
Ale kliknięcie Dowiedz się więcej przekieruje Cię do strony pobierania MySQL, gdzie możesz pobrać sterownik sieciowy.

Po zainstalowaniu (ponownie prosta instalacja Next-Next-Next), jeśli spróbujesz ponownie dodać źródło danych z MySQL, pojawi się nowe okno z pytaniem o nazwę serwera i bazę danych, z którą chcesz się połączyć. Na następnym ekranie poprosi Cię o uwierzytelnienie, a na koniec uzyskasz dostęp do menu Nawigatora, w którym wybieramy te same trzy tabele, co w przykładzie Microstrategy, t_f_sales, t_l_category i t_l_product, jak pokazano na rysunku .



Kolejnym krokiem jest zdefiniowanie powiązań między tabelami. Niektóre z nich są automatycznie łączone; w przykładzie stwierdziliśmy, że relacja między tabelami produktów i kategorii została poprawnie zdefiniowana, ale inne muszą zostać ustawione, na przykład relacja poprzez product_id między tabelą sprzedaży a tabelą produktów nie została utworzona automatycznie. Aby zarządzać relacjami, musisz przejść do przycisku Zarządzaj relacjami w menu. Zobaczysz, jak utworzyć relację między tabelami Produkt i Kategoria, a klikając przycisk Nowy będziesz mógł utworzyć nowe relacje,

jeśli którejś brakuje. W tym przypadku utworzymy nową relację dla pola Product_id między tabelą sprzedaży a tabelą produktów. Inną kwestią, którą zobaczysz, jest to, że domyślnie pola liczbowe są uważane za agregowalne, jak widać w prawej części rysunku dla identyfikatorów waluty, klienta i pracownika (są to te pokazane na rysunku, ale większość atrybuty zostały uznane za metryki).



Aby zmienić to zachowanie, aby były dostępne do analizy, musisz je odpowiednio zmienić, aby usunąć agregację.

Można to zrobić w menu Modelowanie, które jest również pokazane na powyższym rysunku w opcji Podsumowanie domyślne, wybierając wymagane pole i wybierając z menu rozwijanego opcję Nie podsumowuj, pokazaną już na tym rysunku. Zrobimy to samo dla wszystkich pól identyfikatora numerycznego, które widzimy w tej sytuacji. Po zdefiniowaniu naszego trybu z kluczowymi atrybutami bez opcji podsumowania i powiązaniu między tabelami zawierającymi tę samą koncepcję, możemy to zrobić zacznij rysować nasz dashboard od wstawienia wizualizacji w obszarze wizualizacji. W Power BI masz podobne wizualizacje jak w Microstrategy, ale jest też kilka innych, jak widać na poniższej liście, gdzie mówimy o głównych dostępnych wizualizacjach:

Wykres słupkowy: masz poziome i pionowe słupki, skupione i ułożone, a także ułożone w 100%. W takim przypadku możesz naprawić tylko wartość i kolor.

Wykres liniowy: linie poziome można dodać do wizualizacji, Wykres warstwowy: obszary mogą być grupowane i skumulowane.

Mieszany wykres liniowy i słupkowy: słupki można grupować i układać w stosy.

Wykres kaskadowy: to nowość w odniesieniu do domyślnej instalacji w Microstrategy. Możesz zobaczyć, jak zróżnicowana jest metryka zarówno pod względem dodatnim, jak i ujemnym dla danego atrybutu.

Wykres punktowy: jest podobny do wykresu bąbelkowego; definiujesz lokalizację każdego punktu na podstawie dwóch metryk, a następnie możesz zmienić kolor i rozmiar. W takim przypadku możesz również dodać oś odtwarzania, która pokazuje na ruchomym wykresie, jak zmieniają się wartości dla każdej wartości atrybutu na osi odtwarzania.

Wykres kołowy: podstawowy wykres kołowy zdefiniowany przez jakiś atrybut i pewną metrykę.

Mapa drzewa: ta sama koncepcja co Mapa cieplna, kwadratowe obszary, których rozmiar jest oparty na metryce.

Mapa i wypełniona mapa: lokalizacja wartości na mapie na podstawie lokalizacji, długości i szerokości geograficznej. Wypełniona mapa pokazuje obszary, standardowa mapa pokazuje znaczniki.

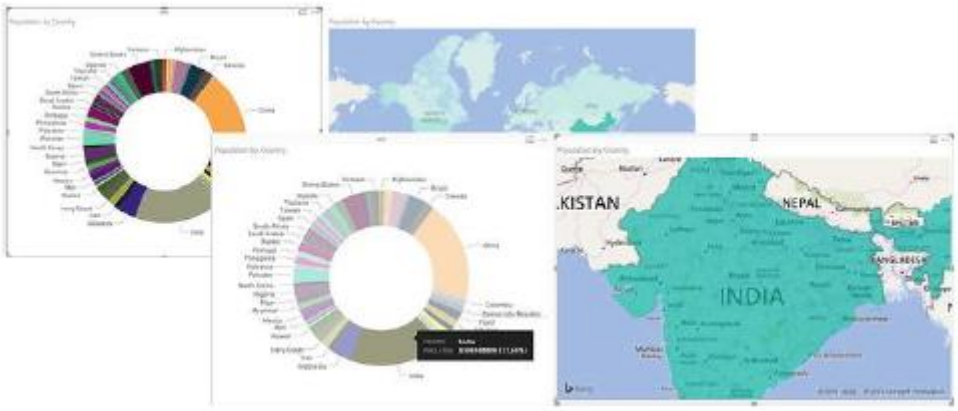
Tabela i macierz: informacje w postaci zwykłego tekstu uporządkowane w wierszach i kolumnach.

Lejek: poziomy pasek ze środkowymi paskami, który pozwala pokazać relacje między elementami w procentach.

Miernik i pączek: podobny do wykresu kołowego, ale o innym kształcie.

Fragmentator: działa jako selektor dla pozostałych wykresów.

Masz również możliwość dodania niestandardowych wizualizacji. W tym narzędziu wszystkie wizualizacje są automatycznie łączone, więc kliknięcie jednego obszaru lub kształtu wizualizacji pokazuje, jak wpływa to na pozostałe. Jak pokazano na rysunku, po kliknięciu segmentu pączka w Indiach mapa po prawej stronie powiększa się, aby pokazać tylko część mapy w Indiach:



Możesz uzyskać dostęp do wszystkich opcji dla każdej wizualizacji za pomocą trzech głównych przycisków, pokazanych na rysunku .

Przycisk z ikoną przedstawiającą dwa pola służy do konfigurowania tego, co jest wyświetlane, co przypisuje efekty do kolorów, rozmiaru itp. Przycisk z rolką pokazuje opcje formatowania, takie jak kolory, tytuły, tło lub obramowania. Wreszcie przycisk z wykresem wewnątrz soczewki ma opcje dodawania analiz, takich jak obliczanie średnich, trendów, linii maksymalnych lub linii stałych. Ten ostatni przycisk ma opcje tylko dla niektórych wizualizacji.

Jeśli chodzi o opcje, które możesz zrobić, aby utworzyć pulpity nawigacyjne w usłudze Power BI, masz do dyspozycji następujące opcje:

Wiele wizualizacji w jednym arkuszu: ten edytor jest bardziej elastyczny w dodawaniu wizualizacji w takim zakresie, w jakim można je zlokalizować i zmienić rozmiar zgodnie z potrzebami, i nie wymaga wykorzystania całej dostępnej przestrzeni.

Dodaj warunki filtrowania: możesz mieć filtry, które mają zastosowanie do wszystkich stron lub do pojedynczej strony.

Użyj obrazów i tekstu: Możesz także łączyć obrazy i teksty, aby dodawać tytuły, komentarze, wyjaśnienia, łącza i logo.

Wiele stron: Możesz dodać nowe arkusze danych, klikając + arkusz.

Edytuj interakcje: Pozwala wybrać sposób działania wykresów po wybraniu obszaru w jednym z nich.

Twórz hierarchie i drążenie: Możesz definiować hierarchie i używać ich do drążenia w danych.

Edytuj relacje: Możesz modyfikować relacje między tabelami również w edytorze graficznym.

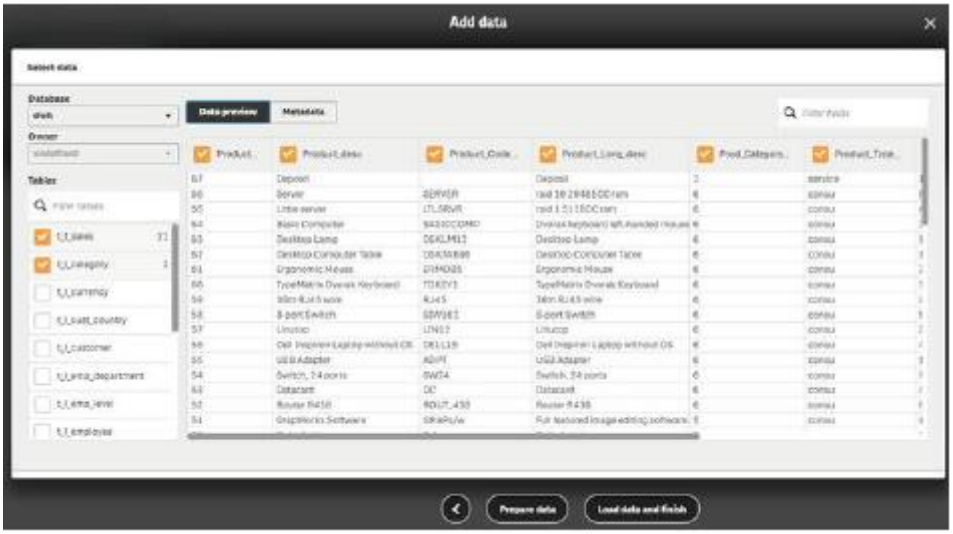
Twórz miary, kolumny i tabele: Korzystając z istniejących pól, możesz tworzyć nowe formuły, atrybuty i zapytania traktowane jako tabele.

Qlik Sense

Qlikview to jeden z pierwszych aktorów, który pojawił się w scenariuszu Business Intelligence z ideą odkrywania danych. Od samego początku narzędzie to koncentrowało się na minimalizowaniu czasu programowania w porównaniu z klasycznymi podejściami BI, dając również użytkownikowi końcowemu możliwość gry z danymi w celu odkrywania trendów, wyjątków i odpowiednich danych w przyjazny dla użytkownika sposób. Pracujemy z Qlikview od kilku lat i jest to dość przyjazne dla użytkownika narzędzie, ale ostatnia wersja, którą zainstalowaliśmy w celach testowych, zaskoczyła nas łatwością obsługi i interfejsem graficznym. W tej sekcji zobaczysz kilka zrzutów ekranu dotyczących narzędzi, które pokazują, jak sądzimy, dość zaawansowany układ. Qlik Sense Desktop to bezpłatne narzędzie opracowane przez Qlikview, które ma ten zaawansowany wygląd i sposób działania, i które pozwoli ci korzystać z wielu funkcjonalności wersji korporacyjnej w sposób lokalny, podobnie jak poprzednie analizowane narzędzia, ale po aktualizacji do wersji korporacyjnej masz możliwość wykorzystania całej mocy platformy komercyjnej. Również bez aktualizacji do opcji komercyjnej będziesz mieć możliwość korzystania z Qlik Sense Cloud za darmo udostępniając swoje dashboardy nawet pięciu użytkownikom. Aby pobrać Qlik Sense Desktop, masz go dostępnego pod tym adresem URL: http://www.qlik.com/us/products/qlik-sense/desktop. Po pobraniu instalacja oprogramowania jest ponownie procesem następny-następny-następny. Aby rozpocząć pracę w Qlik Sense, wystarczy uruchomić narzędzie z menu Start i rozpocząć tworzenie aplikacji. Na pierwszym ekranie zostaniesz zapytany, czy chcesz utworzyć nową aplikację. To jest ekran powitalny, który można odznaczyć, aby nie pojawiał się podczas uruchamiania. Wewnątrz Qlik Sense znajduje się centrum koncepcji, które może zawierać wiele aplikacji, ekran początkowy, który można zobaczyć po usunięciu poprzedniego ekranu początkowego. To centrum można dostosować, klikając przycisk w prawym górnym rogu i wybierając Dev Hub , jak pokazano na rysunku.

Tam możesz stworzyć swój własny styl aplikacji i tworzyć Mashupy z połączonymi informacjami z wielu aplikacji, konfigurować style css, html i tak dalej, aby zintegrować je ze środowiskiem internetowym. Omawiamy to, aby pokazać ci opcję, ale nie będziemy szczegółowo omawiać, jak ją skonfigurować.

Aplikacja jest podobna do dashboardu w Microstrategy i Power BI. Wewnątrz aplikacji możesz mieć wiele arkuszy, a wewnątrz każdego arkusza wiele wizualizacji. Wizualizacje te można tworzyć jako samodzielne obiekty w celu ponownego wykorzystania w wielu arkuszach lub można je osadzać w arkuszu i dostosowywać za każdym razem w inny sposób. Możesz także uporządkować Aplikację, tworząc Stories, które można uznać za prezentacje, które w rzeczywistości można wykorzystać na spotkaniach, ponieważ jej format jest na tyle ładny, aby to zrobić, a także możesz dodawać komentarze i opisy, aby wyjaśnić dane, które pokazujesz. Ale zanim tam dotrzemy, zacznijmy od początku. Pierwszym krokiem jest wybranie danych, które chcemy mieć dostępne do wykorzystania w naszej aplikacji. Wybierając opcję Utwórz nową aplikację i ustawiając nazwę aplikacji, zostaniemy poproszeni o dodanie danych lub otwarcie edytora ładowania danych, również z wieloma przyciskami i menu, które można otworzyć u góry ekranu, na przykład otwórz aplikację przegląd, Menedżera danych, Edytor ładowania danych lub wróć do koncentratora. Również w czarnym pasku znajdują się dwa przyciski do edycji wizualizacji aplikacji oraz modyfikacji tła i kolorów. Edytor ładowania danych otworzy edytor kodu, który pozwoli ci zaprogramować ładowanie przy użyciu niektórych funkcji ETL, transformacji, formuł itp. Jeśli potrzebujesz dostosować swoje ładowanie, możesz znaleźć wiele informacji w dokumentacji produktu i społeczności Qlik. W tym przykładzie użyjemy opcji Dodaj dane, która otworzy menu, w którym możesz wybrać natywne konektory dla wielu baz danych lub, jak to robimy w naszym przypadku, otworzyć istniejący ODBC, który wskazuje na naszą bazę danych. Aby przeprowadzić podobną analizę dla poprzednich narzędzi, wybierzemy te same tabele, t_f_sales, t_l_product i t_l_category. Na rysunku możesz zobaczyć ekran, na którym wybieramy tabele z opcjami podglądu, a także możesz sprawdzić przycisk Metadane, aby zobaczyć, jaka struktura tabeli jest zawarta w każdej tabeli.



Po zatwierdzeniu i wybraniu możesz przejść do Załaduj dane i zakończyć, aby bezpośrednio załadować wszystkie tabele, lub kliknąć przycisk Przygotuj dane, aby zdefiniować relacje między tabelami. Jak można się domyślić, skorzystamy z tej drugiej opcji, aby mieć pewność, że zachowamy poprawną relację między naszymi tabelami. Następnie pojawia się ciekawy ekran: ekran skojarzeń, który pokaże dymek dla każdego stołu. Aby zdefiniować łącza do tabeli, po prostu przeciągnij i upuść jeden z bąbelków na drugi, a relacja domyślnie pojawi się, jeśli będzie w stanie znaleźć pasujące pola. Na rysunku widać ekran Association, a w tym przypadku relację między tabelami sprzedaży i produktów, pole Product_id.

Po zdefiniowaniu asocjacji możesz również przejść do menu Tabele i zmodyfikować niektóre opcje definicji tabeli, takie jak dodanie nowych pól lub przestawienie danych z wierszy na kolumny. Po zdefiniowaniu wymaganych tabel i asocjacji wystarczy kliknąć przycisk Załaduj dane i wejść w projekt arkusza, aby rozpocząć rysowanie naszej wizualizacji aplikacji. Przejdziemy wtedy do edytora arkuszy, pokazanego na rysunku , gdzie możemy rozpocząć definiowanie naszego interfejsu graficznego.

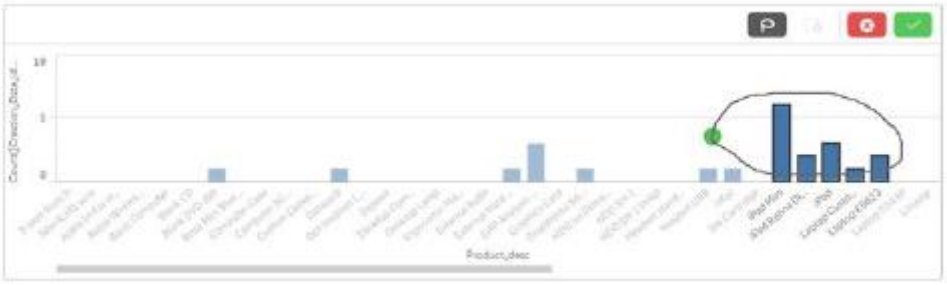
W tym edytorze dostępne są różne obszary: w lewym górnym rogu znajdują się menu umożliwiające poruszanie się po aplikacjach, powrót do menedżera danych, edytora ładowania danych, przeglądarki modelu danych lub powrót do centrum. Kolejny przycisk pozwoli na dodawanie nowych danych i eksport wizualizacji do pdf, powielanie i usuwanie arkuszy oraz dostęp do pomocy. Żadnych komentarzy na temat przycisku Zapisz, ponieważ jest to dość oczywiste.

W następnym podzestawie przycisków masz dostęp do menu Historia, w którym możesz tworzyć nowe historie, które, zgodnie z komentarzem, mogą być używane (i eksportowane) jako prezentacja PowerPoint. Obok menu historii znajduje się przycisk zakładki, w którym możesz zapisywać wybrane dane podczas poruszania się po narzędziu, aby zastosować je w dowolnym arkuszu. W centralnym obszarze masz najpierw pasek przycisków, aby dodać wykresy do szablonu. Na powyższym rysunku widać wszystkie dostępne wizualizacje, przycisk obiektów niestandardowych do dodawania niestandardowych rozszerzeń narzędzia, przycisk elementów głównych, w którym możemy tworzyć wymiary, miary i wizualizacje, oraz przycisk pól, w którym można zobaczyć wszystkie pola dla wszystkich zmapowanych tabel. Do tworzenia wykresów i siatek można bezpośrednio korzystać z pól bazy danych, ale zalecamy użycie wymiarów i miar jako kroku pośredniego, o ile ułatwi to obsługę

aplikacji. Środek centralnego obszaru to szablon, na który przeciągamy i upuszczamy wizualizacje, kształty i teksty; wreszcie w odpowiednim miejscu mamy do dyspozycji panel konfiguracyjny, który w zależności od tego, który element szablonu wybrałeś, pokaże różne opcje. Na powyższym rysunku widzimy opcje konfiguracji arkusza, ale zmienia się to automatycznie po dodaniu nowych wizualizacji. Również opcje będą się różnić w zależności od wybranej wizualizacji.

W dolnym obszarze znajduje się kilka przycisków do wycinania, kopiowania i wklejania obiektów; cofać i ponawiać działania; oraz pokaż i ukryj panel konfiguracyjny. W oparciu o zalecenie utworzenia pośredniej warstwy obiektów, następnym krokiem jest teraz utworzenie wszystkich wymiarów i miar wymaganych do analizy. Jak widać za pomocą tego narzędzia, rozpoznaje pola numeryczne używane do przechowywania dat w formacie RRRRMMDD jako datę i automatycznie tworzy wiele koncepcji dat, rok, kwartał, miesiąc, tydzień i tak dalej. Zatem w tym przypadku tabele t_r_time, t_l_month, t_l_quarter i t_l_year nie będą potrzebne, jeśli będziemy pracować z Qlik Sense. Wymiary, które tworzymy, mogą być pojedyncze lub szczegółowe, w zależności od tego, czy tworzymy je z jednym atrybutem, czy z wieloma powiązanymi. Stworzymy teraz wszystkie wymiary związane z produktem, wiele wymiarów czasowych dla wielu koncepcji dat itp. Stworzymy również wszystkie miary, których chcemy użyć, definiując formułę. Podstawowe formuły będą sumą pól numerycznych i liczby pól wymiarów, ale wtedy możesz użyć SetExpressions i innych parametrów do zdefiniowania filtrowanych metryk, z odrębnymi klauzulami itp. Możesz zobaczyć opcje na rysunku ; jeśli nie masz racji co do dokładnej nazwy pola, możesz wybrać ją z selektorów w prawym obszarze.

Po utworzeniu miar i wymiarów możemy je wykorzystać w wizualizacjach. Aby dodać wizualizacje w arkuszu, wystarczy przeciągnąć je i upuścić w obszarze szablonu, a następnie można przystąpić do ich konfiguracji. Gdy zlokalizujesz wizualizację w obszarze szablonu, pojawi się ona w opcjach wizualizacji panelu konfiguracyjnego, które różnią się w zależności od wykresu, ale głównie są pogrupowane w cztery grupy: Dane, w których określasz zawartość wizualizacji i niektóre opcje danych; Sortowanie, w którym określasz kryteria sortowania wykresu; Dodatki, w których można zdefiniować dodatkowe analizy, takie jak linie trendów; i Wygląd, gdzie można zdefiniować kolory i formaty. Po zakończeniu definiowania szablonu kliknij przycisk Gotowe w prawym górnym rogu, aby rozpocząć analizę. Qlik Sense automatycznie łączy wszystkie wizualizacje, więc kliknięcie dowolnego obszaru wykresu automatycznie filtruje pozostałe wykresy. Aby filtrować, gdy jesteś w trybie wizualizacji, wystarczy kliknąć słupek wykresu słupkowego, komórkę tabeli lub dowolną sekcję dowolnego wykresu, a następnie zostaną wyświetlone przyciski filtrowania, jak pokazano na rysunku 8-26. Będziesz mógł klikać jeden po drugim, aby wybrać i odznaczyć elementy, ale możesz też użyć opcji laso do zaokrąglenia wielu wartości, co widać na rysunku



Wszystko, co filtrujesz, znajduje się w górnym obszarze; wystarczy kliknąć X, aby usunąć filtr zastosowany we wszystkich wizualizacjach. Na rysunku możesz zobaczyć filtry zastosowane do poprzednich wykresów za pomocą pola Product_desc. Samo kliknięcie na X spowoduje zresetowanie wizualizacji do poprzedniego ekranu.

Możesz także wykonać migawki wizualizacji, klikając ikonę kamery, która pojawia się po przesunięciu wskaźnika myszy nad wizualizację i otwarciu wizualizacji w trybie pełnoekranowym, aby drążyć i analizować dane wewnątrz wizualizacji. Po zdefiniowaniu wszystkich arkuszy nadszedł właściwy moment na utworzenie Historii. Jeśli zrobisz to za pomocą przycisku historii, który masz w prawym górnym rogu ekranu, przejdziesz do edytora historii, pokazanego na rysunku .

Tutaj możesz zobaczyć miniatury slajdów po lewej stronie, z dolnym przyciskiem, w którym możesz dodawać nowe slajdy; po prawej stronie widoczny jest również pasek przycisków narzędzi, na którym można wybrać obiekty, które chcesz dodać, obrazy, tekst, kształty, efekty, obiekty multimedialne i arkusze Qlik sense. Po przygotowaniu wszystkich slajdów możesz je zwizualizować na całym ekranie za pomocą zielonego przycisku Play.

Wniosek

Projekt odgrywa ważną rolę w definiowaniu pulpitu nawigacyjnego i musisz mieć jasność, że interfejs BI jest narzędziem graficznym, a jeśli chcesz, aby projekt BI zakończył się sukcesem, musi być użyteczny, przyjazny dla użytkownika i miły dla użytkowników końcowych. Z tego powodu zaczęliśmy przeglądać różne podejścia do narzędzi BI i zapoznać się z wieloma zaleceniami projektowymi dla naszych pulpitów nawigacyjnych i dokumentów, które pozwolą nam zdefiniować przydatne, przyjazne dla użytkownika i ładne pulpity nawigacyjne dla naszych klientów, które chcemy opracować , a także zobaczyć z wieloma przykładami najlepsze praktyki użycia wykresów. Przestrzeganie tych zaleceń jest koniecznością, aby zapewnić powodzenie wdrożenia BI. Ale musisz też wziąć pod uwagę, że funkcjonalności, które Twój

model i narzędzie BI oferują Twoim użytkownikom, muszą spełniać ich wymagania i oczekiwania. Tylko z przedstawionymi najlepszymi praktykami nie można odnieść sukcesu. Dokonaliśmy przeglądu tylko niektórych opcji dostępnych na rynku, które uznaliśmy za interesujące do dodania , ale istnieje wiele narzędzi w różnych wersjach, bezpłatnych, komercyjnych, korporacyjnych, z szerokim zakresem cen, funkcjonalności, konfiguracji i możliwości. Jeśli chodzi o trzy narzędzia wybrane do analizy, Microstrategy, Power BI i Qlik Sense, pokazaliśmy przegląd funkcjonalności, ale mają one znacznie więcej opcji, zwłaszcza jeśli przejdziesz do ich wersji komercyjnych lub korporacyjnych. Funkcjonalności darmowych wersji uważamy za dość podobne. Może Qlik Sense z możliwością tworzenia historii daje wartość dodaną, ale oni robią głównie to samo tylko z kilkoma innymi opcjami. Jeśli masz zamiar wykorzystać je do stworzenia projektu pilotażowego jako poprzedzającego krok instalacji komercyjnej, to musisz wziąć pod uwagę wiele innych czynników, liczbę użytkowników, oczekiwany odzew, oczekiwane funkcjonalności, wykorzystanie chmury, układ serwera lub ilość danych i oceń, które narzędzie, w ramach tych trzech lub poza nimi, lepiej pasuje do Twoich potrzeb. Naszym skromnym zdaniem Power BI jest tańszy i można go uznać za opcję dla małych i średnich projektów, Qlikview jest opcją pośrednią pod względem ilości danych i ceny, a Microstrategy oferuje platformę najbardziej solidną i z szerszymi funkcjonalnościami od administracji i dostarczania danych z tych trzech, ale także najdroższy pod względem licencji. Ale to tylko opinia. Kończąc , jeśli postępowałeś zgodnie z instrukcjami zawartymi w całym tekście, powinieneś mieć platformę BI wraz z jej głównymi komponentami, począwszy od platformy transakcyjnej, bazy danych, procesu ETL do jej wypełnienia oraz narzędzia BI uzyskującego dostęp do modelu danych. Oczywiście cały projekt powinien być zgodny z metodykami Agile, a Twoja baza danych i ETL będą optymalne. Przejdźmy do kolejnych części, które pozwolą nam zasymulować scenariusze za pomocą narzędzia MOLAP, zobaczyć, jak utrzymać te wszystkie rozwiązania, jak zautomatyzować wszystkie procesy, czy wreszcie możliwości przeniesienia rozwiązania do środowiska chmurowego. Mamy nadzieję, że Ci się spodoba!

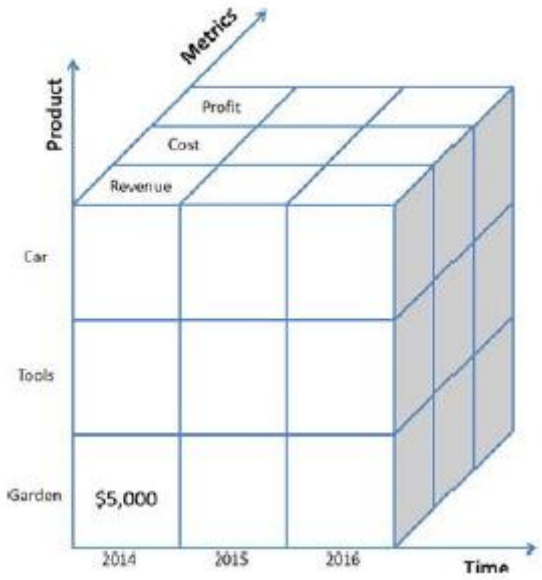
9. Narzędzia MOLAP do budżetowania

Czas na dodatkowy utwór bonusowy! Uważamy narzędzia MOLAP za część całej platformy BI, ale zazwyczaj wykracza to poza zakres zwykłych rozważań BI. Jednak, jak wyjaśniono w Części 1, uważamy to narzędzie za ostatni krok procesu BI wewnątrz firmy, głównie z dwóch powodów: i tak można je wykorzystać do analizy danych przez użytkowników końcowych, więc można je uznać za BI; a zdefiniowanie budżetu i celów na kolejne okresy pomaga firmie skoncentrować się na codziennych działaniach i może być wkładem do systemów BI, gdy np. chcemy porównać scenariusze rzeczywiste i docelowe. Pracujemy w naszym systemie operacyjnym ratującym codzienną działalność firmy; następnie wyodrębniamy go za pomocą narzędzia ETL, ładując do naszej bazy danych informacje, które zostaną przeanalizowane za pomocą narzędzia BI. W tej chwili wyciągamy wnioski z analizowanych danych i zastanawiamy się nad działaniami, które należy podjąć, aby poprawić wyniki naszej firmy, ale przed zastosowaniem chcemy wiedzieć, co każde działanie implikuje w zakresie poprawy przychodów netto. Oprócz tego, że jest ścieżką bonusową, komponent ten jest zwykle wdrażany oddzielnie od reszty środowiska BI. Możliwe, że ten komponent już istnieje w Twojej firmie, gdy rozpoczynasz proces BI, w zależności od dojrzałości procesu każdego działu. O ile tego rodzaju narzędzia są wielokrotnie wykorzystywane do analiz finansowych, możliwe, że Twój dział finansowy już korzysta z jakiegoś narzędzia do budżetowania i symulacji różnych scenariuszy. Z drugiej strony możesz preferować swój proces BI , a gdy system będzie w pełni działał i zbierał rzeczywiste dane, zaczniesz od możliwości tworzenia symulacji „co by było, gdyby", aby zobaczyć, jak wpłyną one na pozostałe wskaźniki KPI w oparciu o pewne przypuszczenia. Więc jeśli zamierzasz postępować zgodnie książką krok po kroku, aby bawić się proponowanymi narzędziami, po prostu zrób to i wypróbuj nasze propozycje. Ale jeśli traktujesz to jako przewodnik po wdrożeniu produktywnego systemu BI, najpierw upewnij się, że żaden inny dział nie korzysta z tego rodzaju narzędzi, a jeśli nie, może lepiej pozostawić tą część w trybie zamrożenia, dopóki nie skonsolidujesz reszta procesu BI. Najpierw musisz przeanalizować rzeczywiste dane swojej firmy, określając, które KPI będą się koncentrować na Twojej firmie, zbierając historyczne trendy do analizy danych, a następnie możesz zacząć myśleć o symulowaniu scenariuszy na przyszłość. W każdym razie przejdźmy dalej, podając kilka przykładów dotyczących użycia narzędzi MOLAP. Wyobraź sobie, że wracasz do przykładu sieci sklepów żelaznych. Ty, jako dyrektor generalny, możesz myśleć o otwarciu nowego sklepu, aby poprawić sprzedaż, i może to być prawdą, jeśli go otworzysz, ale co może się stać z Twoimi wynikami przychodów netto? Możesz spróbować zasymulować wynik na podstawie wielu scenariuszy, grając z wieloma zmiennymi i porównując wyniki. Możesz ocenić, co się stanie, jeśli trzy sklepy, które są bliżej nowego, zmniejszą sprzedaż ze względu na bliskość otwarcia nowego sklepu. Możesz ocenić, jak wpływa to na zjazdy, które otrzymujesz od swoich dostawców, zwiększając ogólną sprzedaż firmy. Możesz również ocenić wpływ na przychody netto potrzeby zapełnienia nowego sklepu poprzez zmniejszenie kosztów zapasów w pozostałych sklepach. Możesz sprawdzić, co się stanie, jeśli lokalizacja sklepu nie jest wystarczająco dobra i sprzedajesz tylko połowę tego, co jest oczekiwane lub jak możesz naciskać na dostawców artykułów ogrodniczych, aby zwiększyli rabaty, ponieważ ten sklep ma szansę podwoić sprzedaż produktów ogrodniczych ponieważ znajduje się w regionie z dużą ilością domów z ogrodami. Z pewnością będziesz w stanie symulować scenariusze sprzedaży, porównując arkusze Excela, a także wiele narzędzi do budżetowania, które pomogą Ci w procesach budżetowania bez wdrażania systemu MOLAP. Ale korzystanie z narzędzi MOLAP da ci możliwość zapisywania informacji w jednym repozytorium współdzielonym przez użytkowników, zapewniając, że nie ma błędów w złożonych plikach Excel, które używają setek formuł i używając złożonych obliczeń w trybie przygotowanym do analizy danych na zagregowanych poziomach z dużą prędkością. MOLAP może wykorzystywać dwie główne strategie: wstępnie obliczone modele, które zawierają dane już wstępnie obliczone na wielu poziomach, które natychmiast zwracają dane, lub modele online, które szybko agregują dane w celu zwrócenia wyników.

I to jest jedna z głównych zalet wielowymiarowych baz danych: informacje można wyszukiwać na wielu poziomach każdego wymiaru kostki lub bazy danych, co zapewnia bardzo szybką wydajność. Ale zanim przejdziemy do szczegółów, jak używać wielowymiarowego narzędzia do analizy danych, przyjrzyjmy się głównym koncepcjom MOLAP. Po przeanalizowaniu ogólnych koncepcji zobaczymy przegląd narzędzia MOLAP, ponieważ w tym przypadku trudno jest znaleźć narzędzia open source lub darmowe wersje komercyjnych, narzędzia MOLAP nie są tak szeroko stosowane jak narzędzia BI. Mamy duże doświadczenie z Hyperion Essbase dostarczanym przez Oracle, ale nie ma darmowej wersji tego produktu, więc przeszukaliśmy rynek i wybraliśmy komercyjne narzędzie, które oferuje bezpłatną wersję swojej aplikacji, PowerOLAP, aby przejść do kilku przykładów dotyczących Możliwości MOLAP. Nie mieliśmy żadnego wcześniejszego doświadczenia z tym narzędziem, ale oceniliśmy je na potrzeby tej książki, a wyniki i wnioski były wystarczająco dobre, aby przejść do niektórych przykładów funkcjonalności MOLAP. Ale w każdym razie zacznijmy tak, jak skomentowaliśmy, od ogólnych koncepcji.

Wielowymiarowe bazy danych

Wielowymiarowe bazy danych mogą być wykorzystywane do celów budżetowania, ale mogą być również wykorzystywane do samej analizy danych. Skupiamy się teraz na procesie budżetowania, ale są też inne narzędzia dla nich. Używamy ich jako narzędzi do budżetowania, ponieważ zwykle zapewniają interfejs użytkownika, który jest w stanie wstawiać dane, podczas gdy typowe narzędzia BI pozwalają tylko na pobieranie danych, a do dodawania danych do bazy danych potrzebny jest jakiś proces ETL. Będą aplikacje, w których będziemy mogli korzystać zarówno z relacyjnych, jak i wielowymiarowych baz danych, takich jak podstawowa analiza danych, ale będą też aplikacje, w których będziemy mogli czerpać korzyści z wielowymiarowej bazy danych i inne scenariusze, w których wady wielowymiarowej bazy danych będą nas zmuszać użyć relacyjnego. Podstawową cechą wymiarowej bazy danych jest to, że możemy wstępnie obliczyć informacje na wszystkich możliwych poziomach dowolnej kombinacji hierarchii, dzięki czemu mają one bardzo dobrą wydajność dla zapytań, o ile dane są już wstępnie obliczone w bazie danych na pożądanym poziomie. W relacyjnej bazie danych OLAP (ROLAP) można zachować to samo zachowanie, ale w większości przypadków konieczne będzie wykonanie pewnej agregacji danych na pożądanym poziomie używanym w zapytaniu. Wielowymiarowe bazy danych są powszechnie znane jako kostki. Jeśli usłyszysz o koncepcji Cube, możesz wizualizować obiekt z trzema osiami, ale oczywiście, mówiąc o wielowymiarowości, będziemy mieć więcej niż trzy wymiary. W każdym razie idea sześcianu jest dość potężna, aby zrozumieć, jak działają. Wyobraź sobie, że masz kostkę Rubika, więc spróbujmy zbudować bazę danych z trzema wymiarami po trzy elementy w każdym. Na jednej osi masz wymiar czasowy, więc będziesz miał dane za lata 2014, 2015 i 2016; na innej osi możesz mieć wymiar produktu, w naszym ograniczonym przykładzie będziemy mieli tylko trzy kategorie, Ogród, Narzędzia i Samochód; i wreszcie w trzecim wymiarze możemy mieć wymiar Metryki, z Przychodami, Kosztami i Zyskami. Następnie wewnątrz każdego kawałka kostki Rubika będzie znajdować się odpowiednia wartość, więc w pierwszym kawałku będziesz mieć, jak widać na przykładzie na rysunku , 5000 $, co odpowiada wartości przychodów dla ogrodu w 2014 roku.



Wiemy, że dodanie więcej niż trzech wymiarów jest trudne do wizualizacji, ale w oparciu o ten obraz możesz teraz zacząć rozwijać go w swoim umyśle, dodając wymiary i elementy do modelu. Możesz zobaczyć wielowymiarową bazę danych jako bazę danych z pojedynczą tabelą, w której każda kolumna jest powiązana z wymiarem, ale biorąc pod uwagę, że w każdej kolumnie wymiaru możesz mieć wartości dla wielu poziomów wymiaru. Będziesz mieć wiersz z danymi dla całego roku, jeden wiersz dla każdego kwartału i jeden wiersz dla każdego miesiąca. Następnie możesz sobie wyobrazić, że metryki lub wymiary miar są kolejnym wymiarem standardowym, więc będziesz mieć wiersz dla każdej metryki (przychód, korzyść, procentowy wzrost korzyści w stosunku do ubiegłego roku itp.) z jednym polem dla wszystkich wartości, co być znormalizowanym sposobem wyobrażania sobie tego lub kolumną dla każdej metryki, biorąc pod uwagę wymiar metryk jako szczególny. Sposób, w jaki każdy wielowymiarowy dostawca przechowuje dane, będzie zależał od każdego dostawcy, a także niektórzy dostawcy umożliwią skonfigurowanie każdego wymiaru w celu zapisywania informacji w trybie rozrzedzonym lub gęstym, który można zidentyfikować za pomocą wierszy i kolumn. Tryb rzadki utworzy nowe wiersze tylko wtedy, gdy istnieje kombinacja klucza, a tryb gęsty utworzy komórkę po prostu tworząc element członkowski, więc można je uznać za kolumny.

Wymiary lub osie do analizy

Jak można się domyślić, w definicji wielowymiarowej bazy danych głównym tematem do zdefiniowania jest to, jakich wymiarów użyjemy w analizie. W takim przypadku możemy przeprowadzić podobną analizę, jak w rozdziale 5, gdzie widzieliśmy, jak modelować naszą hurtownię danych. Nie może być tak samo, o ile w swojej wielowymiarowej bazie danych nie będziesz miał zwykle tego samego poziomu szczegółowości informacji, co w hurtowni danych; być może masz model powiązany z hurtownią danych w hurtowni danych, który można bezpośrednio wykorzystać jako model wielowymiarowy, ale w większości przypadków będziesz musiał ponownie wykonać tę analizę dla wielowymiarowej bazy danych. Pojawią się atrybuty, których użycie w wielowymiarowej bazie danych nie będzie miało sensu, takie jak numer zamówienia i faktury, o ile będą zawierały tylko informacje za rok, klienta, produkt lub scenariusz, więc nie ma sensu porównywać, jak zmienia się numer faktury w zależności od klienta lub wymiaru czasowego, ponieważ będziemy mieć ten numer faktury tylko dla jednego klienta i okresu. W konkretnym przypadku tego rozdziału, w którym chcemy wykorzystać wielowymiarową bazę danych do budżetowania, pojawią się również pewne atrybuty, które należy ewentualnie pominąć; będziesz

miał w swojej bazie informacje na poziomie produkt-klient i jeśli nie masz niewielu klientów i mało produktów w swojej firmie, nie ma sensu próbować definiować budżetu na przyszłoroczną sprzedaż na poziomie produkt-klient. Najprawdopodobniej Twój budżet na przyszły rok będzie realizowany na wyższych poziomach hierarchii produktów, np. kategorii lub rodziny oraz typu klienta lub grupy klientów. Pojęcie wymiaru w tym obszarze jest dość podobne do tego, które widać również w rozdziale 1; wymiar to oś, która pozwala nam podzielić informacje na różne poziomy wymiarów, aby zobaczyć, jak metryki działają w każdym elemencie wymiaru. Główna różnica polega na tym, że w tym przypadku wymiar zawiera członków z wielu poziomów w zdefiniowanej hierarchii, aw czystym wymiarze BI traktujemy go jako grupę powiązanych ze sobą atrybutów. Inna różnica w wykorzystaniu wymiarów między relacyjnymi i wielowymiarowymi bazami danych polega na tym, że podczas gdy w relacyjnej bazie danych można wybrać dowolne pole z dowolnej tabeli, w wielowymiarowej bazie danych konieczne będzie użycie co najmniej jednego członka dla każdego wymiaru w zapytania. Jeśli nie wybierzesz członka danego wymiaru, zwróci ci wartość dla nazwy wymiaru, zwykle zawierającą agregację całego wymiaru, biorąc i tak element wymiaru. Zwykle w większości narzędzi można wyróżnić trzy główne typy wymiarów:

Wymiar metryki : zawiera definicję metryki dla wszystkich metryk, których będziemy używać w modelu. Jest to obowiązkowe o ile bez metryk nie będziesz miał czego analizować. Niektóre narzędzia nie traktują metryk ani miar jako samych wymiarów, ale jako pola kostki, więc wizualizują je w kolumnach, a nie w wierszach. Ale we wszystkich aplikacjach będziesz używać metryk, aby móc pracować z kostkami.

Wymiar czasu : czas ma zazwyczaj pewne określone właściwości i funkcje ułatwiające zarządzanie wymiarami, takie jak akumulacja od początku roku do daty, kumulacja roku do końca lub poprzedni okres, które umożliwiają łatwe obliczanie tego typu agregacji i zmian w widoku czasu. Tego rodzaju zmiany widoku czasu można powiązać z transformacjami czasu.

Standardowy wymiar: Pozostałe wymiary można uznać za standardowe.

Uwaga: Wszystkie metryki będą zwykle wartościami liczbowymi, ale niektóre aplikacje umożliwiają zapisywanie metryk tekstowych i dat, ale nie pozwalają na dowolny tekst, tylko selektor wartości, które są mapowane na liczbę, a daty są zapisywane jako liczba, więc w na koniec dnia zapisują wartości liczbowe.

Wymiar może być podstawowy, mieć tylko kilka elementów, które pozwalają zobaczyć wiele elementów, lub hierarchiczny, zorganizowany w trybie rodzic-dziecko, w którym każdy rodzic jest sumą swoich dzieci. Wewnątrz wymiaru hierarchicznego możemy mieć wiele poziomów hierarchii i grupowania z jednorodną strukturą, w której wszystkie gałęzie wymiaru są zakończone lub z heterogeniczną strukturą, w której będziesz mieć różną liczbę poziomów w każdej gałęzi. Przykładem jednorodnej struktury może być wymiar czasowy, w którym zawsze występuje ta sama liczba poziomów: Rok, Kwartał i Miesiąc. Byłoby bardzo dziwne mieć wymiar czasowy, w którym w czerwcu docierasz do dziennego szczegółu, aw kwietniu docierasz tylko do miesięcznego szczegółu. Typowymi wymiarami heterogenicznymi mogą być metryki, w których można grupować metryki na podstawie agregacji, które mają sens; w przypadku analizy finansowej będziesz mieć pewne koszty, które zostaną pogrupowane w rodzaje kosztów; wtedy będziesz mieć sprzedaż brutto, sprzedaż netto, przychód netto, a wszystkie te metryki będą miały różne poziomy metryk zdefiniowanych jako dzieci i potomkowie. Typ struktury dla każdego wymiaru będzie się różnił w zależności od Twoich potrzeb i będzie miał mniej lub bardziej implikacje w zależności od wybranego narzędzia i dostępności funkcji dla odniesień do poziomów. Aby przejść do wyjaśnienia definicji wymiaru, spróbujmy zaproponować Ci model oparty na przykładzie, z którego korzystaliśmy w dalszej części. W tym przypadku użyjemy

naszego modelu budżetowania do określenia naszych celów w zakresie sprzedaży i przychodów netto, więc będzie to mieszanka analizy sprzedaży i finansów. Podczas gdy większość wymiarów występuje w hurtowni danych, istnieje jeden specyficzny, związany z procesem budżetowania, scenariuszowy. Scenariusz pozwoli Ci mieć wiele wersji tych samych informacji, aby mieć możliwość porównania danych między nimi; w rzeczywistości będziemy mieć już członka wymiaru, który oblicza porównanie między nimi. Kolejna uwaga dotycząca wymiarów wybranych w naszym modelu dotyczy modelu Year One. Zdecydowaliśmy się na osobny wymiar dla roku dotyczący wymiaru Czas, aby łatwo porównywać różne lata, wybierając miesiące w kolumnach i lata w wierszach lub odwrotnie, a także uprościć obsługę wymiaru czasu, o ile każdego roku będziesz należy dodać tylko członka (numer roku), a nie 17 (1 rok, 4 kwartały i 12 miesięcy). W zależności od narzędzia, każdy poziom hierarchii będziesz odnosił się do poziomu numerycznego, zaczynając od zera dla tych, które nie mają dzieci, będziesz miał możliwość odniesienia się do nich jako do pokoleń, zaczynając od generacji zero od góry hierarchię i jeśli Twoje narzędzie na to pozwala, będziesz mógł zdefiniować poziomy hierarchii za pomocą nazw.

Składowe wymiaru

Wszystkie wymiary zostaną złożone przez wiele elementów. Członek to wartość wymiaru, w roku wymiaru będziemy mieć 2014, 2015, 2016, każdego członka hierarchii klientów, którego chcemy mieć w kostce, każdy element produktu, każdą metrykę itp. W zależności od narzędzia, którego używamy , będziesz mógł zdefiniować wiele typów członków: niektóre z nich są stałymi członkami, które przechowują dane, a inne są tylko formułami zdefiniowanymi w wymiarze. Możesz także mieć wskaźniki do innych członków, również nazwanych współdzielonymi członkami, innych członków, którzy mogą być tylko etykietami bez żadnych danych w środku. To, czy możesz mieć różne typy członków, będzie zależeć od wybranego narzędzia. Możesz także klasyfikować członków jako elementy podstawowe lub zagregowane, biorąc pod uwagę, że elementy bazowe będą tymi, które nie mają żadnego dziecka, co w niektórych narzędziach jest również nazywane poziomem zerowym. W naszym przykładzie utworzymy przykładowy model z kilkoma elementami, definiując tylko mały sześcian, aby pokazać, jak to działa, ale który będzie zawierał wystarczającą ilość danych, aby zrozumieć dostępne opcje. Krotka jest zdefiniowana jako przecięcie pojedynczego elementu każdego wymiaru. Używając przykładu kostki Rubika, krotka byłaby każdym kawałkiem sześcianu, a ten jednoznacznie zidentyfikowany przez Garden, 2014 i Revenue ma wartość 5000 $. Sześcian zdefiniowany przez wymiary pokazane na powyższym rysunku jest małym sześcianem, ale bierze się pod uwagę, że jeśli zostanie całkowicie wypełniony, będziesz mieć ponad 100 000 kombinacji danych lub krotek, tylko z tymi małymi wymiarami, aby załadować cały sześcian będzie miał właśnie z tą atrapą kostki plik zawierający około 20 000 wierszy, jeśli ustawisz metryki w kolumnach. Będzie to zależeć od tego, jak zarządza narzędziem i danymi poniżej, ale zwykle nie wszystkie kombinacje zostaną utworzone automatycznie, tylko te z danymi. Ale w przypadku hierarchii, jeśli dodasz dane na przykład w styczniu, zostaną utworzone dane dla kwartału 1 i roku ogółem, więc jeśli masz hierarchię z 10 poziomami i dodasz dane na najniższym poziomie, utworzy to do 10 razy więcej ilość wprowadzonych danych. Należy więc zachować ostrożność podczas definiowania kostki i oceniać wpływ na miejsce na dysku i wydajność ładowania danych przed dodaniem ogromnej liczby elementów w wymiarze. Nie należy uważać wielowymiarowej bazy danych za hurtownię danych; po prostu użyj go do przechowywania ograniczonej i zagregowanej liczby elementów.

Uwaga: Należy zauważyć, że podczas gdy w relacyjnej bazie danych można usunąć element z tabeli wymiarów i załadować go ponownie bez utraty danych faktów, bazy danych MOLAP są definiowane przez elementy wymiaru i dane, które znajdują się na przecięciach elementów (krotki), więc jeśli w procesie przeładowania usuniemy członka wymiaru, dane dotyczące tego członka zostaną całkowicie utracone, dla wszystkich kombinacji z resztą wymiarów. Uważaj więc na strategię ładowania

wymiarów, upewniając się, że nie usuwasz potrzebnych elementów z wymiarów. Próba wyczyszczenia nieużywanych elementów może spowodować utratę informacji dotyczących wcześniejszych scenariuszy.

Udostępnianie właściwości w kostkach

W narzędziu MOLAP zwykle zobaczymy jakiś sposób grupowania kostek, który pozwoli nam dzielić się pewnymi właściwościami. Aby wyjaśnić sens korzystania z tych grup, pokażmy kilka konkretnych przykładów opartych na istniejących narzędziach i ich nazwach. W Essbase odnoszą się do bazy danych dla kostki i aplikacji dla zestawu kostek i można zdefiniować dostęp, parametry, limity i polecenia serwisowe, takie jak start i stop. W PowerOLAP nazywają je odpowiednio kostką i bazą danych, a wymiary można udostępniać w różnych bazach danych. Więcej szczegółów na temat ich używania poznamy w kolejnych sekcjach, analizując sposób ich użycia, zwłaszcza PowerOLAP, który ma dostępną darmową wersję.

Język MDX e

MDX to język, który będzie używany z narzędzi interfejsu raportowania do łączenia się z kostkami w celu pobierania informacji w celu pokazania ich do celów analizy. Język MDX, akronim wyrażenia MultiDimensional, jest podobny do języka SQL omówionego w rozdziale 3, ale podczas gdy SQL był używany do komunikowania się z relacyjną bazą danych, MDX jest używany do komunikowania się z wielowymiarową bazą danych. Mają pewne podobieństwa, ale MDX ogranicza się tylko do odczytu danych z możliwością zastosowania obliczeń i formuł do danych źródłowych, podczas gdy w SQL można również tworzyć obiekty bazodanowe. Istnieją narzędzia MOLAP, które zapewniają przyjazny dla użytkownika interfejs, który pomaga uniknąć używania MDX, o ile bezpośrednio tłumaczą twoje żądania na MDX i pokazują wyniki. Inne narzędzia nie mają tego narzędzia i musisz wchodzić w interakcję z aplikacją za pomocą MDX i innych narzędzi, które umożliwiają mieszane użycie; dostarczają podstawowy MDX na podstawie twoich żądań, a następnie możesz modyfikować i dostrajać ten MDX, ale w tym celu powinieneś przeczytać o wiele więcej dokumentacji MDX niż ta, którą wyjaśnimy w tej sekcji. Chociaż poświęciliśmy cały rozdział (Rozdział 3) omówieniu SQL, nie chcemy wchodzić w szczegóły działania MDX, ponieważ uważamy, że MOLAP jest dodatkową funkcją i możliwe, że go nie potrzebujesz, więc postaramy się dać ci krótki przegląd. Główną operacją w MDX jest wykonywanie selekcji organizujących w kolumnach i wierszach sposób, w jaki otrzymujesz informacje, z możliwością dodania osi do zapytania, takich jak strony, rozdziały i sekcje, ale z ograniczeniem, którego nie można pominąć osie. Nie możesz więc dokonać wyboru zorganizowanego w rozdziały, jeśli nie zawiera on czegoś w kolumnach, wierszach i osiach stron. W instrukcjach select użyjesz klauzuli FROM do zdefiniowania kostki, której używasz do pobierania informacji, i możesz dodać klauzule WHERE podobne do tych, które widzisz w SQL, aby filtrować pobierane informacje, generując wycinek danych. Może to być wybrane zestawienie z kostki Forecast, którą utworzymy w celu przeanalizowania naszych prognoz sprzedaży na przyszłe lata:

```
SELECT

[Metrics].Members ON COLUMNS,

[Time].[Month].Members ON ROWS

[Product].[Category].Members ON PAGES

FROM

[Forecast]
```

WHERE

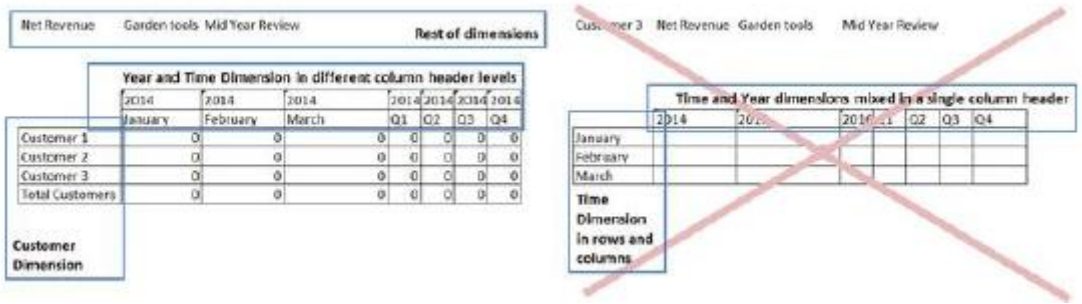
([Year].[2016],[Scenario].[Actual])

W tym zestawieniu wybieramy dane za rok 2016 i scenariusz rzeczywisty, co widać w klauzuli WHERE w zestawieniu select. Wynik zostanie uporządkowany w pięciu kolumnach, po jednej dla każdej metryki zdefiniowanej klauzulą NA KOLUMNACH, w dwunastu wierszach, po jednej dla każdego miesiąca zdefiniowanego w klauzuli NA WIERSZACH i z czterema stronami oraz po jednej dla każdej kategorii zdefiniowanej w klauzuli NA STRONACH. Istnieje kilka wymiarów, które można pominąć w instrukcji select, w tym przykładzie nie ustalamy, dla którego Klienta chcemy wyodrębnić dane lub czy chcemy mieć stronę dla każdego Klienta. Gdy nie mamy żadnego odniesienia do istniejącego wymiaru na kostce, zwróci ona wartość domyślną, zwykle powiązaną z sumą, więc w tym przypadku zwróci dane związane z Total Customer. Również w instrukcjach select możesz dodawać obliczone elementy członkowskie w czasie pobierania lub zapisywać je jako element członkowski, dodając złożoność i wiele funkcji, aby móc uzyskać dane pochodzące z tego, który zapisałeś w bazie danych. Oprócz tego będziesz mógł używać instrukcji sub-select, aby móc wykonywać złożone zapytania.

Uwaga: Jak skomentowaliśmy na początku tej sekcji, nie chcemy przeprowadzać wyczerpującej analizy możliwości MDX, ponieważ jest całkiem możliwe, że nie potrzebujesz ich w swojej implementacji MOLAP, ale jeśli chcesz przeanalizować w głębszy sposób szczegółowo, jak działa MDX, zalecamy zajrzeć do dokumentacji narzędzia do raportowania opartego na kostkach MOLAP, icCube, gdzie masz dość przyjazne odniesienie MDX (być może MDX), dostępne pod następującym linkiem: https:// www.iccube.com/support/documentation/mdx/index.php

Interfejsy danych MOLAP

Często zdarza się, że używając narzędzi MOLAP (PowerOLAP będzie działał w ten sposób, podobnie jak Essbase), instalujesz wtyczkę w aplikacji Microsoft Excel, która umożliwia łączenie, pobieranie i ładowanie danych do bazy danych MOLAP za pomocą Excela, więc głównym interfejsem użytkownika dla tego zestawu narzędzi będzie Twoje już znane i przydatne oprogramowanie dostarczane przez najpopularniejszego dostawcę oprogramowania na świecie. Korzystanie z programu Excel jako interfejsu pozwoli Ci wybrać część danych lub wycinek z całej bazy danych oraz w łatwy sposób odczytywać i zapisywać dane. W tym celu zwykle umieszcza się niektórych członków z jednego lub wielu wymiarów w kolumnach (jeden wymiar w kolumnie), niektórych członków jednego lub wielu wymiarów w wierszach (jeden wymiar w wierszu), a niektórych członków pozostałych wymiarów w początkowe wiersze arkusza Excela pełnią rolę stron Twoich danych, a następnie będziesz mógł uruchamiać, pobierać lub wstawiać polecenia za pomocą menu wtyczek interfejsu. Logicznym ograniczeniem odczytu i zapisu danych jest to, że nie można jednocześnie zlokalizować elementów z tego samego wymiaru na więcej niż jednej osi, wierszach i kolumnach, kolumnach i stronach lub wierszach i stronach. Na rysunku 9.5 pokazujemy przykład, jak wygląda poprawny szablon danych programu Excel i niepoprawny.



Z drugiej strony istnieją narzędzia, które mają własny interfejs dostępu do danych zarówno do ładowania, jak i analizy. W takich przypadkach każdy interfejs będzie miał własne ograniczenia i zasady.

Uwaga: Istnieją narzędzia BI, które pozwalają łączyć się z wielowymiarowymi bazami danych, oferując wszystkie możliwości BI również dla danych w bazach danych MOLAP z możliwością porównywania danych z hurtowni danych z danymi w MOLAP, co jest również przydatne do sprawdzania integralności danych w różnych systemach.

Import danych

Posiadanie utworzonej struktury bazy danych ze wszystkimi wymiarami i członkami na miejscu jest dobrym punktem wyjścia, ale po utworzeniu struktury bazy danych będziesz musiał wypełnić ją danymi, w przeciwnym razie cały ten proces byłby bezużyteczny. Aby załadować informacje do baz danych MOLAP będziemy mieli głównie dwie możliwości: wypełnić je ręcznie poprzez interfejs aplikacji (albo Excel, albo własny interfejs, jak pokazano w poprzedniej sekcji); lub zautomatyzować ładowanie danych z niektórych różnych źródeł danych, takich jak pliki proste, pliki programu Excel lub relacyjne bazy danych, konfigurując połączenia ODBC lub JDBC. Import danych przyspieszy i ułatwi ładowanie danych do bazy danych poprzez ustanowienie bezpośredniego połączenia między bazą danych a źródłem danych. Zainicjowanie procesu może wymagać większego nakładu pracy niż bezpośrednie wpisywanie i zapisywanie danych do bazy danych w sposób ręczny, ale usprawni proces ładowania danych w przypadku dużej ilości danych. Import danych jest szeroko stosowany do ładowania scenariuszy rzeczywistych, o ile pochodzą one bezpośrednio z istniejącego systemu, ale podczas zapisywania scenariuszy prognozy, budżetu i prognozy może się zdarzyć, że funkcja importu danych nie będzie tak użyteczna. Można go i tak użyć podczas początkowego ładowania budżetu, a następnie można edytować i zapisywać dane z interfejsu ręcznego, aby dostosować każdą metrykę w zależności od potrzeb. Import danych pozwoli również na ułatwienie aktualizacji wymiarów w oparciu o relacyjne tabele lub pliki poprzez włączenie możliwości dynamicznej zmiany struktury bazy danych poprzez dodawanie członków, aktualizowanie opisów czy zmianę relacji nadrzędnych. Inną możliwą korzyścią płynącą z importu danych, gdy narzędzie MOLAP posiada interfejs wiersza poleceń, jest możliwość zautomatyzowania procesu importu danych za pomocą narzędzia workflow do synchronizacji procesów.

Przygotowanie danych źródłowych

W przypadku korzystania z interfejsu ręcznego do ładowania danych, wystarczy przygotować arkusz Excela z informacjami przygotowanymi do załadowania na określonym poziomie lub wystarczy połączyć się z bazą danych i przesłać tam informacje. Ale jeśli myślisz o jakiejś automatyzacji do ładowania danych do bazy danych MOLAP, całkiem możliwe, że potrzebujesz manipulacji danymi, aby osiągnąć określony poziom. Prawdopodobnie użyjesz jednej z dwóch strategii: utwórz widok z zaznaczeniem, który zwraca dane zagregowane do pożądanego poziomu, lub utwórz tabelę i powiązany proces, aby wypełnić ją w tym samym celu. W większości przypadków każdy z nich będzie wymagał wyboru agregacji, o ile zwykle masz mniejszy poziom szczegółowości w bazie danych MOLAP w porównaniu z hurtownią danych. Jeśli spróbujemy połączyć proponowany model danych z tym, którego użyliśmy w naszej relacyjnej bazie danych, zobaczymy, że mamy tabelę faktów, t_f_sales zdefiniowaną między innymi na poziomie klienta, dnia, produktu, faktury i będziesz chciał podsumuj to do rodziny produktów poprzez tabelę t_l_product (która w rzeczywistości w danych źródłowych nie istniała bezpośrednio w tabeli wymiarów produktu, załóżmy dla tego przykładu, że jest powiązana z kategorią Produkt i w kostce będziemy mieli kategorię opis), będziesz musiał podsumować to do miesiąca za pomocą tabeli t_r_time i będziesz potrzebować niestandardowej agregacji dla hierarchii

klientów, o ile niektórzy klienci zostaną wprowadzeni bezpośrednio, a inni mniejsi klienci zostaną połączeni w grupy. Tak więc zapytanie wymagane do utworzenia widoku lub wypełnienia tabeli byłoby:

```
select date_format(s.day_id, '%M') month,
cat.prod_category_desc category,
'Actual' scenario,
t.year_id,
case when c.customer_desc in ('Customer 1',
'Customer 2','Customer 3')
then c.Customer_desc
else 'Customer Group 2'
end customer,
sum(Amount_inv_num) Gross_sales
from t_f_sales s, t_r_time t, t_l_product p,
t_l_month m, t_l_category cat, t_l_customer c
where s.day_id=t.day_id and
s.product_id=p.product_id and t.month_id=m.month_id
and
p.prod_category_id=cat.prod_category_id and
s.customer_id=c.customer_id
```

Eksport danych

W ten sam sposób, w jaki można skonfigurować procesy importu danych, można skonfigurować procesy eksportu danych, które umożliwiają zbiorczy eksport informacji z bazy danych do plików prostych lub relacyjnych baz danych. Ważną użytecznością tego eksportu danych byłoby sfinalizowanie cyklu życia BI. Po zweryfikowaniu naszego budżetu, przeprowadzając wiele symulacji i wybierając jeden scenariusz jako wynik naszej analizy, możesz wyeksportować te dane do załadowania do hurtowni danych, aby śledzić, jak radzi sobie z wynikami rocznymi w porównaniu z oczekiwanymi KPI, zwłaszcza jeśli Twoja aplikacja BI nie daje możliwości podłączenia go do bazy danych MOLAP. Eksport danych może być wykorzystany do dowolnego innego celu, który wymagałby posiadania informacji w plikach tekstowych lub w bazie danych, takich jak zapisywanie ich w systemie ERP dla dowolnych wymagań, przesyłanie danych w wielu środowiskach MOLAP, wysyłanie go klientom lub jakimkolwiek innym wymogom dotyczącym przepływu danych. Ponownie możesz wyeksportować również strukturę bazy danych, więc dobrze byłoby skonfigurować kopię zapasową struktury, załadować tę strukturę do innego środowiska MOLAP lub wyeksportować strukturę jako próbkę, a następnie użyć jej jako szablonu do dodania kolejnych elementów do swoją bazę danych MOLAP.

Obliczenia

Dane są zwykle importowane na najniższym poziomie każdego wymiaru, a następnie agregowane, o ile większość wymaganych obliczeń to tylko sumy wartości związanych z elementami potomnymi. Ale tym obliczeniem może być dowolna inna formuła, którą można sobie wyobrazić jako stosunki, różnice, mnożenia, procenty lub jakąkolwiek inną formułę wymaganą dla Twojej firmy. Zarówno podstawowa agregacja, jak i inne obliczenia muszą być w jakiś sposób zdefiniowane w narzędziu MOLAP, aby system mógł wiedzieć, jak obliczyć dane dotyczące każdego członka, które nie zostały bezpośrednio przesłane ręcznie lub automatycznie. Ale obliczenia nie są wykonywane tylko od dołu do góry. Wyobraź sobie, że robisz budżet na przyszły rok, a twoja hierarchia produktów ma sześć poziomów: produkt ogółem, kategoria, podkategoria, rodzina, podrodzina i produkt; a cała hierarchia ma 4000 produktów. Zdefiniowanie budżetu przyszłorocznej sprzedaży na poziomie produktu może być koszmarem, w tym niewygodnym, np. jeśli masz 4000 produktów, całkiem możliwe, że w ciągu roku będziesz mieć nowe produkty w swojej hierarchii i niektóre wycofane. Więc w tej sytuacji jest całkiem możliwe, że przygotujesz budżet na poziomie kategorii lub podkategorii, określając budżet na 20, 30 lub 50 elementów, ale nie na 4000 elementów. Ten punkt może być dla Ciebie interesujący, ponieważ masz kalkulację odgórną, która rozkłada całkowitą kwotę oczekiwanej sprzedaży na poziomie podkategorii na poziom produktu, używając niektórych danych jako czynnika rozrzutu, aby dotrzeć do produktu. Ta pojemność jest jedną z najbardziej użytecznych zalet narzędzi MOLAP. Możesz wykonać te obliczenia, używając niektórych skompilowanych funkcji we wszystkich narzędziach MOLAP, które pozwalają (z różną składnią w każdym narzędziu) odwoływać się do danych z powiązanych elementów poprzez hierarchię lub pozycję. Będziesz miał możliwość odniesienia się do POPRZEDNICH elementów na podstawie hierarchii czasowej, aby uzyskać dane z poprzednich miesięcy lub lat, potrzebne w przypadku spreadu, ponieważ rozłożysz Budżet na 2017 rok na podstawie stanu faktycznego z 2016 roku. Będziesz miał możliwość odwołania się do PARENT wartość, aby pomnożyć bieżącą wartość członka przez jego rzeczywiste dane i podzielić ją przez rzeczywiste dane członka nadrzędnego. Będziesz miał możliwość odniesienia się do najwyższego członka hierarchii, aby uzyskać wartość procentowego wzrostu sprzedaży w odniesieniu do bieżącego roku, jaki chcesz osiągnąć w następnym ćwiczeniu oraz wiele innych funkcji, które mogą być przydatne do wykonywania obliczeń MOLAP. Dostępne funkcje i możliwości MOLAP są bardzo podobne do tych oferowanych przez język MDX, ale czasami język skryptowy jest specyficzny dla używanego narzędzia, więc zapoznaj się z dokumentacją narzędzia, aby dowiedzieć się, które funkcje są dostępne w twoim przypadku i specyfikacje dotyczące sposobu ich używać. Teraz, gdy widzieliśmy uzasadnienie korzystania z bazy danych MOLAP w naszym procesie budżetowania i przejrzeliśmy kilka istotnych tematów, aby zrozumieć koncepcje MOLAP, przejdźmy do ich analizy za pomocą PowerOLAP, jednej z opcji dostępnych na rynku.

PowerOLAP

PowerOLAP to produkt oferowany przez firmę PARIS Technologies, który umożliwia korzystanie z funkcjonalności MOLAP w przyjazny dla użytkownika sposób za pomocą prostego interfejsu. Istnieją głównie dwie opcje, komercyjna, która obejmuje komponent serwerowy jako rdzeń systemu obsługującego użytkowników, oraz wersja osobista, która jest dostępna za darmo. To będzie ten, którego użyjemy, aby pokazać, jak wdrożyć system MOLAP, o ile da nam tę samą funkcjonalność projektową, ale po prostu zastosujemy pewne ograniczenia w rozmiarze kostek, z których będziemy mogli korzystać : 10 kostek, 10 wymiarów i 10 000 elementów; oraz niektóre zadania administracyjne, takie jak zarządzanie bezpieczeństwem. Aby uzupełnić naszą bazę danych będziemy mieli głównie dwie możliwości: skorzystanie z własnego narzędzia PowerOLAP lub wtyczki PowerOLAP dodanej do Excela, obie instalowane podczas procesu instalacji. Silnik PowerOLAP stosuje strategię korzystania z modelu online, więc dane są przechowywane na poziomie podstawowym, a następnie agregowane w bardzo szybkiej strategii, gdy je wysyłasz. Jest to przydatne, aby mieć możliwość szybkiego odświeżenia w przypadku modyfikacji danych podstawowych, ponieważ wstępnie obliczone modele wymagają zwykle

dużego nakładu czasu na obliczenia przy użyciu procesów wsadowych, podczas gdy modele online ponownie obliczają informacje w krótkim czasie. Pobieranie nie będzie natychmiastowe, jak w przypadku wstępnie obliczonych modeli, ale chodzi o to, aby mieć zrównoważony scenariusz między czasem ładowania a czasem pobierania.

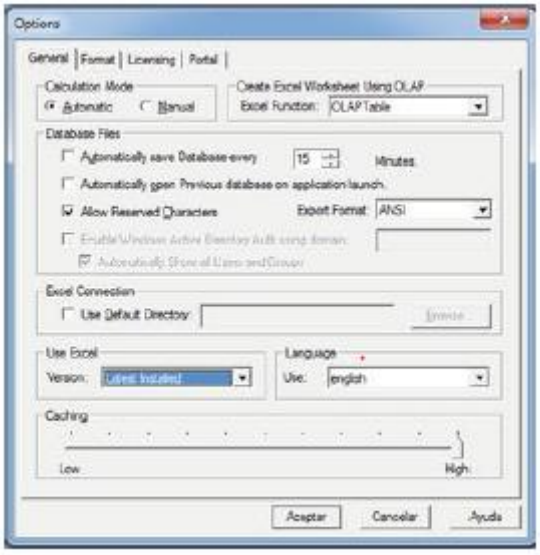
Począwszy od PowerOLAP

Jako pierwszy krok do korzystania z PowerOLAP, jak zwykle będziemy musieli go zainstalować. W tym celu przejdź do strony Paristech pod tym adresem URL http://paristech.com/products/powerolap i kliknij opcję Pobierz PowerOLAP Personal. Poprosi cię o rejestrację, a oni wyślą ci link umożliwiający dostęp do pobierania, i jest całkiem możliwe, że skontaktuje się z tobą ktoś z zespołu sprzedaży PowerOLAP. Rozpocznie się pobieranie pliku zip o nazwie PowerOLAP-Personal-16044.zip lub podobnej, w zależności od wersji dostępnej podczas czytania tych wierszy. Wystarczy rozpakować i zawiera plik wykonywalny. Instalacja jest dość łatwa, Next-Next-Next iw końcu będziesz mieć zainstalowane PowerOLAP Personal na swoim laptopie, oba już skomentowane komponenty, aplikację i dodatek do Excela. Możesz znaleźć aplikację w menu Start ➤ folder PowerOLAP lub podczas uzyskiwania dostępu do programu Microsoft Excel zobaczysz dodatek i stamtąd możesz również uruchomić aplikację za pomocą pierwszego przycisku wstążki zainstalowanej w programie Excel. Po uruchomieniu programu PowerOLAP zobaczysz interfejs podobny do pokazanego na rysunku 9.8, z pięcioma głównymi menu. Strona główna to miejsce, w którym można znaleźć kilka ogólnych opcji, takich jak przyciski wycinania, kopiowania i wklejania , opcje konfiguracji bazy danych i inne opcje wizualizacji, w tym łącze również do Centrum pomocy i Genius, które zawiera łącza do pomocy technicznej, filmów i wersji próbnych. W menu Model znajdziemy dostęp do głównych komponentów projektu bazy danych, wymiarów i kostek, a także dostęp do przycisku sprawdzania składni (cóż, a jeśli masz wersję serwerową, będziesz miał również dostęp do zakładek bezpieczeństwa). Następne menu to menu Slice, które zawiera wszystkie wymagane przyciski do zarządzania plasterkami, aw kilku liniach zobaczymy, czym one są i jakie są główne opcje. Mamy również menu Dane, za pomocą którego możemy połączyć naszą kostkę ze źródłami zewnętrznymi, głównie plikami płaskimi i bazami danych w sposób dwukierunkowy, z możliwością odczytu z bazy danych w celu załadowania danych lub zbudowania wymiarów oraz z możliwością eksportu obu danych i wymiary. Wreszcie mamy menu Narzędzia, w którym możemy uzyskać dostęp do tservers i skonfigurować wiele opcji, ale dostępne tylko w wersji komercyjnej. Skoro już przedstawiliśmy Ci interfejs, zobaczmy, jak to działa. PowerOLAP używa wymiarów, które działają jak osie do analizy informacji, jak już wyjaśniono. W przypadku PowerOLAP wymiary są izolowanymi obiektami, w innych bazach danych są częścią sześcianu; ale tutaj mamy kostki i wymiary niezależne od siebie, więc możesz mieć kostki, które nie używają niektórych wymiarów bazy danych i wymiarów używanych w wielu kostkach.

Uwaga: Zarządzanie członkami wymiaru musi być wykonywane z uwzględnieniem faktu, że wymiar może być używany w wielu kostkach, więc usunięcie członka wymiaru może mieć wpływ na inne kostki, z którymi nie pracujemy w danym momencie. Aby tego uniknąć, możesz użyć cechy Trwałość w każdym członku, którego chcesz zachować w stanie nienaruszonym.

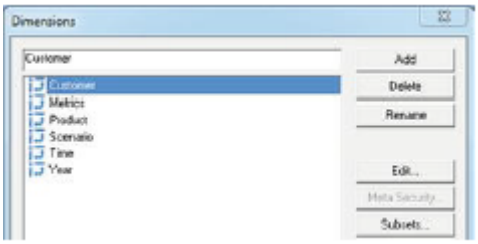
Aby rozpocząć pracę z PowerOLAP, pierwszym krokiem będzie utworzenie bazy danych, która na końcu będzie tylko plikiem na Twoim komputerze, zawierającym całą wymaganą infrastrukturę i dane, z rozszerzeniem .olp. Możesz to zrobić, tworząc w lewym górnym rogu przycisk z pomarańczową kostką i wybierając Nowy. Istnieje kilka opcji konfiguracyjnych dostępnych z menu Plik, które są definiowane na poziomie bazy danych, takie jak tryb obliczeń, opcje automatycznego zapisywania, poziom buforowania lub niektóre opcje formatowania. Możesz je zobaczyć na rysunku 9-9. Z tego ekranu tryb obliczania jest najważniejszym parametrem, jeśli chodzi o określenie, czy zagregowane dane w Twojej

bazie będą obliczane tylko poprzez przesłanie danych, czy też musisz wykonać ręczną aktualizację danych.



Tworzenie wymiarów w PowerOLAP

Po utworzeniu bazy danych następnym krokiem będzie utworzenie żądanych wymiarów. Musimy przejść do menu Model i kliknąć przycisk Wymiar, aby wejść do katalogu wymiarów, gdzie dodamy wszystkie nasze wymiary, wpisując nazwę i klikając przycisk Dodaj przycisk. Możesz zobaczyć wynik na rysunku , gdy stworzymy wszystkie nasze wymiary .



Po dodaniu wymiaru będziemy musieli dodać członków wymiaru i zdefiniować hierarchię. W tym celu naciskamy przycisk Edytuj pokazany również na rysunku powyżej i wchodzimy do Edytora wymiarów, który można zobaczyć na rysunku.

Wybraliśmy ten wymiar, aby Ci pokazać, ponieważ tutaj możesz zobaczyć wiele opcji i szczegółów. Po lewej stronie masz dostępnych członków do użycia w hierarchii, aw prawym obszarze masz kompilację hierarchii. Za pomocą pierwszego przycisku będziesz mógł dodawać nowe elementy do listy, co nie oznacza, że musi być uwzględniany w hierarchii, a po utworzeniu wystarczy przeciągnąć i upuścić element, umieszczając go na żądanym poziomie z żądanym rodzicem - relacje z dziećmi. Domyślne

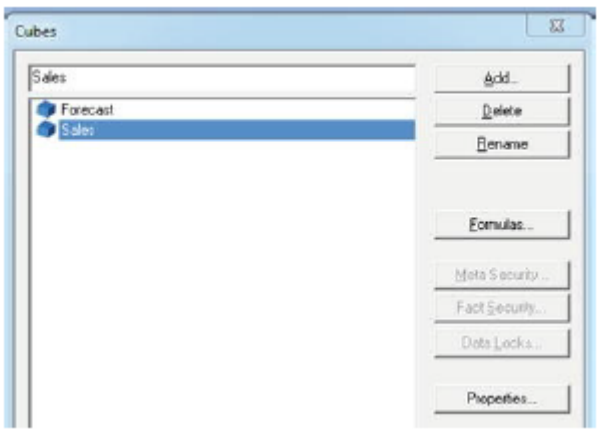
umieszczenie elementu poniżej innego oznacza dwie rzeczy: dane nowego elementu zostaną skonsolidowane na najwyższym poziomie, domyślnie agregując 100% jego wartości; i górny poziom nie jest już uważany za element bazowy, a jego ikona zmienia się z # na Σ. Powiedzieliśmy agregację domyślnie 100%, ponieważ można to zmienić za pomocą ikony balansu zmieniającej wagę wartości elementu. W tym przykładzie zmieniliśmy wartość na -1, aby element Rzeczywisty vs. Prognoza był różnicą między Rzeczywistym a Prognozowanym, ale można również utworzyć obliczenia, takie jak średnia ważona, przy użyciu wymaganej wagi.

Uwaga: Ze względu na strukturę kostek PowerOLAP nie można zapisywać danych bezpośrednio w zagregowanych elementach; dane można zapisywać tylko w elementach bazowych. Aby wykonać obliczenia, takie jak rozłożenie kosztów, musisz mieć prostą hierarchię bez elementów zagregowanych i utworzyć kalkulację do wykonania podziału. Więc w tym przypadku nie byłoby to idealne narzędzie.

Istnieje więcej rzeczy do zrobienia w definicji wymiaru. Na przykład możesz zdefiniować aliasy dla członków, aby mieć różne etykiety odnoszące się do tego samego obiektu, co może pomóc ci odróżnić to, co chcesz zobaczyć w wycinku, od tego, czego potrzebujesz do załadowania danych. Wyobraź sobie, że ładujesz dane faktów dotyczące produktów za pomocą publicznego kodu EAN, ale chcesz zobaczyć w pobieraniu opis produktu, możesz utworzyć oba aliasy i używać jednego w każdym momencie. Możesz się tam dostać za pomocą trzeciego przycisku pokazanego na rysunku 9-11. Możesz także zmodyfikować niektóre właściwości dla każdego elementu, które można następnie wykorzystać w obliczeniach, uzyskując dostęp do piątego przycisku (gwiazdka) i oznaczając elementy jako Trwałe, aby nie utracić danych związanych z elementem w przypadku automatycznego wczytania wymiaru, jak już skomentowano w poprzedniej notce. W końcu masz dostępne przyciski, które mogą ułatwić implementację hierarchii poprzez dodanie członków jako rodzeństwa i dzieci. Możesz także zdefiniować podzbiory elementów w każdym wymiarze, które mogą ułatwić tworzenie hierarchii, a także mogą być wykorzystywane w obliczeniach.

Definicja sześcianu

Po zdefiniowaniu wszystkich wymiarów zdefiniowanie sześcianu jest dość łatwe, wystarczy wejść do katalogu kostek, przechodząc do menu Model i klikając Kostka, uzyskując zakładkę podobną do tej pokazanej na rysunku . Następnie musisz zdefiniować nazwę i kliknąć Dodaj, otwierając nowy ekran, w którym wybierasz wymiary, których chcesz użyć w swojej kostce spośród wszystkich dostępnych wymiarów. Na poziomie kostki możesz zdefiniować zestaw opcji, głównie związanych z buforowaniem w edycji Personal, klikając przycisk Właściwości pokazany na rysunku , a także możesz zdefiniować formuły związane z kostką, klikając przycisk Formuły. Opcje bezpieczeństwa są włączone tylko w wersji komercyjnej.



Plastry

Jak widać na wstępie, możemy mieć tysiące, a nawet miliony krotek (lub komórek) z kilkoma elementami w wymiarach, o ile masz krotkę dla każdej kombinacji elementów wymiaru. Aby ułatwić obsługę danych, będziemy musieli naprawić części bazy danych, w których chcemy ładować i analizować dane; w przeciwnym razie potrzebowalibyśmy ekranu kinowego z super-bardzo-ultra-wysoką rozdzielczością, aby zobaczyć wszystkie komórki na jednym ekranie. Segmentacja bazy danych w celu zarządzania danymi odbywa się za pomocą wycinków, porcji informacji zdefiniowanych przez kolumny, wiersze i strony. Plasterki można tworzyć i używać zarówno z programu Excel, jak iz aplikacji PowerOLAP. Aby uzyskać dostęp do plasterków z PowerOLAP wystarczy przejść do menu Slice i tam można nimi zarządzać, tworząc nowe, otwierając, zamykając, kopiując i zarządzając przyciskami Slice. Przejdźmy do New Slice, a następnie pojawi się menu z selektorem kostek. Wybierzemy jedną z ostatnio utworzonych kostek, a kiedy klikniemy Ok, pojawi się nowy wycinek z aspektem pokazanym na rysunku

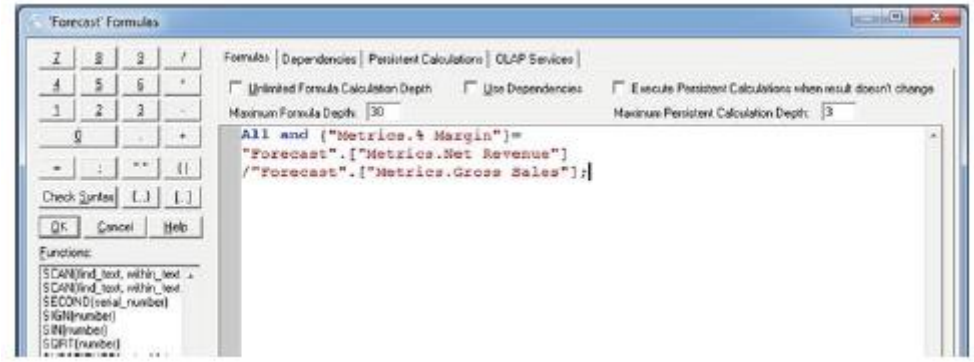
Aby dostosować to, czego chcemy używać i co chcemy pokazać, możemy przesuwać wymiary między obszarami filtrów, etykiet kolumn i etykiet wierszy, a klikając F9, aby odświeżyć, zobaczymy, jak to wygląda. Klikając dwukrotnie wymiar, będziemy mogli wybrać, które elementy każdego wymiaru będą używane w kolumnach, wierszach i filtrach. Po zapisaniu za pomocą przycisku Zarządzaj plasterkiem możesz wykonać pewne czynności, takie jak dodanie ograniczeń do danych, które można wstawić, aby spróbować zminimalizować błędy wprowadzania danych. Tutaj możesz również zaznaczyć, aby ukryć puste i zerowe wiersze. Chociaż może to być interesujące do czytania, nie będzie przydatne do wstawiania danych, o ile będzie całkiem możliwe, że cały wiersz będzie pusty podczas wypełniania kostki, więc nie będziesz go wizualizować i wygrałeś nie można dodawać danych. Po zakończeniu definiowania Plasterka możesz otworzyć go w Excelu, klikając przycisk Worksheet z menu Plasterek. Chcielibyśmy również skomentować, że możesz zobaczyć dane w formacie graficznym, wybierając jeden z wykresów z przycisku Wykres.

Uwaga: Jeśli zdefiniowałeś podzbiory podczas definiowania wymiaru, będziesz mógł ich użyć w tym momencie, aby wybrać elementy, które chcesz użyć w przekroju.

Formuły

W przycisku Formuły, które skomentowaliśmy w części definicji kostki, uzyskujemy dostęp do edytora formuł, który pozwala nam zdefiniować, które obliczenia zostaną zastosowane do kostki. Jeśli nie zdefiniujesz żadnej formuły, dane zostaną zagregowane, podsumowując wartości podstawowe w różnych hierarchiach. To właśnie jest uważane za czystą agregację. Ta agregacja jest najbardziej optymalna z możliwych, ponieważ bierze pod uwagę tylko te komórki, które zawierają informacje, pomijając puste wiersze, o ile narzędzie jest w stanie wykryć brak danych przed zapytaniem o rzeczywistą wartość. Ale możemy również zdefiniować każde obliczenie, o którym możemy pomyśleć.

Możemy mieć wskaźniki, procenty, różnice, średnie, maksimum, minimum lub dowolną inną formułę potrzebną do naszej analizy. Aby dodać formułę, wystarczy kliknąć przycisk Formuły w katalogu kostek . Następnie zobaczymy edytor formuł, jak ten pokazany na rysunku , który zawiera również przykład obliczania metryki na podstawie dwóch innych metryk, % Margin zdefiniowanej jako Przychód netto / Sprzedaż brutto.



Nie chcemy wchodzić w szczegółowe informacje na temat języka programowania dla PowerOLAP, ale tylko kilka komentarzy. W pierwszej części zestawienia ustalamy poziom prętów, na których zostanie zastosowana formuła, w tym przypadku All oznacza cały sześcian (dla pozostałych wymiarów część z Metrics, czyli pręt, który obliczamy), ale możemy wybierz również Agregaty i Szczegóły, aby ustalić elementy zagregowane lub podstawowe, następnie słowo, a następnie element (lub elementy, możesz określić więcej wymiarów oddzielonych andami), który chcemy obliczyć. Po drugiej stronie równości ustawiamy metryki definiujące formułę, w tym przypadku podzielone przychody netto i sprzedaż brutto. Instrukcje muszą kończyć się znakiem „;”.

Z tego edytora ponownie zaznacz niektóre przyciski:

* {…}: służy do wyboru obszaru docelowego i elementów obliczeń.

* […]: służy do wybierania prętów zawartych w formule.

* Sprawdź składnię: aby sprawdzić, czy to, co piszemy, jest poprawne.

Uwaga: Korzystając z funkcji zależności, można zaoszczędzić dużo czasu wykonywania obliczeń. Służy do obliczania wartości metryki wynikowej tylko wtedy, gdy metryka użyta w formule nie jest pusta. Aby z niej skorzystać należy zaznaczyć flagę Użyj zależności i zdefiniować zależność w zakładce Zależności. Więcej informacji o formułach i zależnościach można znaleźć w dokumentacji produktu.

Buforowanie

Dzięki konfiguracji buforowania można zaoszczędzić czas, zwłaszcza w przypadku kosztownych obliczeń, które wykorzystują wszystkie komórki, ponieważ zależności nie wchodzą w grę. Pamięć podręczna zapisuje niektóre zagregowane wartości już obliczone za pomocą algorytmu, aby wybrać, które wartości najlepiej zapisać na podstawie liczby operacji wymaganych do uzyskania danych. Konfigurację buforowania można zdefiniować na poziomie kostki, włączając lub wyłączając ją we właściwościach kostki, a także na poziomie bazy danych, gdzie można zdefiniować poziom buforowania oznaczający liczbę obliczeń wymaganych, aby komórka została uznana za komórkę buforowaną. W podejściu z niskim buforowaniem rozważy większą liczbę operacji, więc zapisze mniej komórek, o ile w najniższym przypadku zapisze tylko te komórki, które wymagają tysięcy obliczeń. Na najwyższym poziomie zapisuje komórkę co 500 jednostek obliczeniowych, więc plik pamięci podręcznej jest

większy. Możemy dostroić poziom pamięci podręcznej, powtarzając testy i uzyskując to, co uważamy za najlepszy stosunek wykorzystanej przestrzeni do czasu pobierania.

Import i eksport danych

Używanie wycinków do wypełnienia kostki może być przydatne w przypadku małych kostek lub procesów budżetowania, w których chcemy stworzyć wiele scenariuszy i porównań między nimi, ale jak skomentowaliśmy w poprzednich sekcjach ogólnych, może to być bezużyteczne, jeśli chcemy wypełnić duże ilości danych w nasza baza danych. Jak widzieliśmy, zwykle istnieje wiele sposobów importowania danych za pomocą narzędzia, które pozwala na łatwe aktualizowanie wymiary i dane faktów, znajdujące się w menu Dane pokazane na rysunku.

Stąd możemy importować i eksportować dane z plików, przenosić dane między kostkami, uruchamiać trwałe obliczenia i tworzyć kostki od zera z relacyjnej bazy danych za pomocą opcji OLAPExchange. Opcja Metadata służy do eksportowania/importowania struktury kostki i formuł, opcja Factdata służy do eksportowania/importowania danych liczbowych do kostki.

Aby załadować metadane i fakty, zdecydowanie zalecamy najpierw wyeksportować plik, aby poznać format, który należy zaimplementować, aby poprawnie załadować pliki. Poniżej znajduje się przykład naszego modelu z tylko kilkoma wierszami danych z polami oddzielonymi tabulatorami (można to wybrać podczas importu), ale jak widać wymaga on nagłówka, aby poprawnie zrozumieć poniższe pola. Następnie za pomocą tego przykładu możesz wygenerować i załadować pełny plik kostki.

SCHEMA TRANSFER

COLUMN M Customer

COLUMN M Metrics

COLUMN M Product

COLUMN M Scenario

COLUMN M Time

COLUMN M Year

STARTINLINE

Customer 3 Net Revenue Garden tools Mid

Year Review February 2014 21,00

Customer 3 Net Revenue Garden tools Mid

Year Review January 2014 12,00

Customer 3 Trade Deals Garden tools Mid

Year Review January 2014 2,00

...

ENDINLINE

Eksportowanie danych faktów umożliwia również zapisywanie danych w relacyjnej bazie danych, umożliwiając dostęp do nich z poziomu narzędzia BI lub wykonywanie dowolnych czynności związanych z zarządzaniem. W czasie eksportu możesz zdefiniować część kostki, którą chcesz wyeksportować, filtrując według dowolnego wymiaru lub używając podzbiorów danych, aby pobrać żądaną porcję danych do relacyjnej bazy danych. Konfiguracja eksportu jest zapisywana przed wykonaniem, więc możesz ją wykonać tyle razy, ile potrzebujesz. Dzięki opcji Transfer Cube w menu Transfer danych możesz kopiować dane między kostkami lub wewnątrz kostki przez jakieś pole. Bardzo często będzie wykonywana kopia z jednego scenariusza do drugiego, aby zapisać wersje swoich danych, aby mieć możliwość porównania po dowolnej symulacji warunków, którą możesz spróbować wykonać. Zapamiętaj wprowadzenie do tego rozdziału: możesz chcieć porównać, co mogłoby się stać z przychodami netto Twojej firmy, gdyby nowy sklep oznaczał zmniejszenie całkowitych kosztów lub zmniejszenie całkowitej sprzedaży dla pozostałych. Mamy wreszcie sekcję opcji połączenia z relacyjną bazą danych i tworzenia od zera kostek i wymiarów. Wybierając żądane tabele do tworzenia wymiarów i ładowania faktów, kostka zostanie utworzona automatycznie, definiując elementy, aliasy elementów, właściwości elementów, atrybuty drążenia wszerz, atrybuty zestawienia, kolejność elementów itp. Po zapisaniu definicji kostki można ją zaktualizować w pełni lub przyrostowo, aby uzyskać lepszą wydajność aktualizacji, także z możliwością aktualizacji pojedynczych wymiarów kostki.

Wniosek

W tej dodatkowej części nauczyłeś się głównych koncepcji teoretycznych baz danych OLAP, które pozwalają przygotowywać cele na kolejny okres oraz tworzyć i porównywać wiele scenariuszy danych do budżetowania tego celu w łatwy sposób, centralizując wszystkie dane na serwerze dostępnym dla wielu użytkowników. Widzieliśmy również przykłady głównych funkcjonalności OLAP z bezpłatną wersją PowerOLAP zainstalowaną na laptopie, przy instalacji pojedynczego użytkownika. Może nie pasuje do twoich potrzeb i potrzebujesz większej instalacji z większymi kostkami i większą liczbą użytkowników, ale w tym celu zalecamy skorzystanie z narzędzia komercyjnego, o ile darmowe wersje nie są wystarczająco wydajne i niektóre otwarte źródła, które mamy oceniane są dość skomplikowane w użyciu, wymagające umiejętności programowania w Javie. Jeśli dotarłeś do tego miejsca bez pominięcia żadnego rozdziału, masz teraz przegląd wszystkich komponentów BI, które są wymagane do posiadania kompletnego rozwiązania BI. Gratulacje za sfinalizowanie całej infrastruktury platformy BI! Ale jeszcze nie skończyliśmy; mamy jeszcze kilka innych obszarów do zobaczenia. W tym momencie powinieneś zainstalować wiele komponentów na wielu serwerach, z wieloma procesami ETL, raportowaniem i ładowaniem do narzędzia MOLAP, które są ze sobą powiązane w logiczny sposób, ale będziemy musieli zorganizować wszystkie te procesy, dodając między nimi zależności z niektórymi narzędziami koordynacji. W prawdziwym scenariuszu będziesz mieć więcej niż jedno środowisko do opracowywania nowych procesów i testowania ich przed przejściem do produkcji. Jest kilka najlepszych praktyk, które chcemy Ci pokazać. Wreszcie, w oparciu o Twoje potrzeby, można zalecić możliwość przejścia do rozwiązań w chmurze dla wszystkich komponentów iw tym obszarze należy również rozważyć kilka kwestii. Wszystkie te tematy zostaną przeanalizowane w dalszych częściach, więc jeśli jesteś zainteresowany wszystkimi tymi punktami, nie wahaj się i śmiało z nimi!

10. Planowanie procesów BI: jak organizować i aktualizować uruchomione procesy

Do tej pory zbudowaliśmy nasze rozwiązanie od bazy danych po warstwę raportowania i system budżetowania. Być może udało Ci się zbudować najlepszy pulpit nawigacyjny, jaki kiedykolwiek widziano, ze wszystkimi kluczowymi wskaźnikami wydajności, które Twoi użytkownicy muszą analizować, ale Twoi użytkownicy będą okresowo potrzebować świeżych danych. Oznacza to, że musimy wprowadzić system działający codziennie; lub z częstotliwością, której potrzebujemy; a w przypadku jakiegokolwiek błędu lub problemu podczas procesu ładowania, otrzymujemy co najmniej alert, więc osoba odpowiedzialna może przejrzeć i przeanalizować, co się stało. Tu najpierw zobaczymy, jak zakończyć nasz projekt ETL, projektując ostateczne zadanie, które uruchomi transformację, zbierze informacje i będzie kontrolować przebieg jej wykonania; i uważaj, jeśli coś pójdzie nie tak. Następnie przejdziemy dalej, aby zobaczyć, jakie mechanizmy ma PDI, aby uruchamiać zadania i transformacje, a następnie przejdziemy do tego, jak zaplanować je. Pod koniec zobaczymy również, jak wykonać niektóre zadania konserwacyjne w bazie danych, aby upewnić się, że nasze procesy działają dobrze po pewnym czasie, poprzez gromadzenie świeżych statystyk i przeprowadzanie kontroli spójności danych.

Kończę ETL

Pierwszym krokiem przed rozpoczęciem etapu planowania naszych procesów jest zakończenie naszego ETL. Kończąc, mamy na myśli zamknięcie naszej transformacji w pracy. Chociaż możliwe jest uruchomienie transformacji Kettle bezpośrednio z wiersza poleceń lub dowolnego innego harmonogramu zadań, zwykle nie jest to dobra opcja. Jeśli chcemy dodać kontrolę przepływu, powiadomienia e-mail w przypadku awarii i inne kontrole, musimy połączyć je w ramach zadania. Ponadto bardzo często występuje więcej niż jedna transformacja. Aby synchronizować, umieszczać zależności, organizować przepływ danych i czynności, które mają być na nich wykonywane, potrzebujemy kontrolera. Ten kontroler to praca. Naszym pierwszym zadaniem w tym rozdziale jest więc oczyszczenie ETL i utworzenie zadania zawierającego nasze transformacje. Zaczynajmy!

Tworzenie pracy w PDI

Mieliśmy dwie transformacje, jedną, która pobiera dane z bazy danych Odoo i przechowuje je w inscenizacji, stosując pierwszą partię transformacji; i jeszcze jeden, który pobiera dane z obszaru ODS i przechowuje je w końcowych tabelach hurtowni danych. Aby utworzyć nową pracę, musimy otworzyć edytor Spoon, kliknąć Plik, następnie Nowy, a na koniec wybrać Job. W przeciwieństwie do transformacji, w których zaczynamy czytać źródło danych (zwykle, nie zawsze, ponieważ czasami czytamy parametry lub inne zmienne, takie jak zmienne systemowe), w zadaniu PDI punktem wyjścia jest krok Start, typowa zielona strzałka, którą możemy poszukaj w dziale ogólnym. Przeciągnij i upuść go na płótno, a następnie znajdź dwa kroki transformacji w tej samej ogólnej kategorii i przeciągnij je również na płótno. Na koniec znajdź krok o nazwie Poczta w folderze Poczta, a także przeciągnij i upuść dwa z nich w obszarze roboczym. Aby utworzyć odpowiedni przepływ , upewnij się, że łączysz obiekty tak, jak pokazano na rysunku

Następnie, aby powiedzieć PDI, która transformacja musi zostać wykonana w każdym kroku, należy dwukrotnie kliknąć krok transformacji iw zależności od tego, czego używasz (zwykłe pliki lub repozytorium), wybrać odpowiednią opcję, aby zlokalizować transformację, której będziesz używać. W moim przypadku, ponieważ pracuję na swoim komputerze i nie przeniosłem jeszcze tych plików na serwer, a tylko do testów, wybieram plik ręcznie w moim systemie plików. Ale jeśli na serwerze planujemy użyć repozytorium, co powinno mieć miejsce, to będziemy musieli to zmienić. Zobaczymy, jak to zrobić później, ale w tym momencie po prostu wybierz plik na swoim komputerze, a to pozwoli nam wypróbować pełną pracę. Ostatnim krokiem jest skonfigurowanie obu etapów wysyłania wiadomości e-mail. Wyślą nam one wiadomość, gdy zadanie zostanie zakończone, określając, czy wszystko poszło dobrze (pierwszy) lub kolejny komunikat określający, czy coś się nie powiodło, wraz z logiem błędu, dzięki czemu możemy szybko sprawdzić z naszej skrzynki pocztowej i udać się tam, gdzie trzeba, aby naprawić błąd: zawieszona baza danych, problem z ETL, problem z jakość danych

Uwaga: Aby skonfigurować etap poczty e-mail, potrzebujemy serwera SMTP, aby móc przekazywać wiadomości e-mail do określonych miejsc docelowych. Większość firm ma wewnętrzny serwer SMTP, ale jeśli tak nie jest, nie martw się. Wiele firm udostępnia e-maile SMTP za darmo. Sprawdź warunki każdego z nich, ale polecam https://www.mailjet.com/, który oferuje do 6.000 e-maili miesięcznie za darmo przy użyciu ich serwerów SMTP.

Najłatwiejszym sposobem skonfigurowania obu etapów dostarczania wiadomości e-mail jest utworzenie pierwszego, a następnie drugiego poprzez skopiowanie pierwszego, zmianę jego nazwy i dostosowanie nowych właściwości. Nie będą dokładnie takie same, ponieważ w kroku błędu zdecydowaliśmy się wysłać również pliki dziennika i wiersz błędu w kroku e-mail, więc gdy błąd zostanie zgłoszony, każdy, kto otrzyma wiadomość e-mail, będzie wiedział dokładnie, który błąd wystąpił zostały wyprodukowane bez konieczności sprawdzania procesu ETL. Te wiadomości e-mail będą kierowane do kontrolera procesu, ale dobrym rozwiązaniem jest również dodanie innego adresu e-mail dla użytkowników biznesowych, aby jeśli obciążenie działało dobrze, wiedzieli, że mają dostępne świeże dane. Z drugiej strony przydatne może być posiadanie tabeli kontrolnej zawierającej datę ostatniego wykonania i stan każdego etapu ładowania oraz dodanie do pulpitu użytkownika końcowego małej siatki zawierającej te informacje, aby wiedział na tym samym ekranie, że analizują dane oraz kiedy został uruchomiony ostatni proces odświeżania. Proponowane również darmowe wersje komponentu BI nie pozwalają na planowanie e-maili i automatycznego odświeżania danych, ale jeśli zdecydujesz się przejść na narzędzie, które umożliwia to automatyczne wykonanie stąd, możemy uruchomić polecenie korzystające z narzędzia BI linii poleceń, aby uruchomić aktualizację danych w narzędziu lub automatyczne wysłanie dołączonych informacji pocztą elektroniczną. Aby skonfigurować krok SMTP, musimy znać szczegóły Twojego dostawcy SMTP. Możesz użyć wewnętrznego serwera SMTP lub zewnętrznego. W tym przypadku zdecydowaliśmy się na korzystanie z usługi mailjet SMTP, a szczegóły potrzebne do skonfigurowania kroku to w zasadzie nazwa hosta serwera SMTP, port, nazwa użytkownika i hasło. Kiedy już uzyskamy te informacje (sprawdź swojego administratora lub technika, jeśli używasz wewnętrznego SMTP), możemy zaczynać. Konfiguracja kroku SMTP jest łatwa. Wystarczy dwukrotnie kliknąć krok, a na pierwszej karcie wystarczy skonfigurować źródłowe konto e-mail, docelowy adres e-mail i nazwę nadawcy. Krok serwera jest nieco bardziej złożony. Tutaj musimy skonfigurować wszystkie szczegóły dotyczące serwera, użytkownika i hasła oraz portu. Jeśli używasz mailjet tak jak ja, pamiętaj o określeniu portu 587, ponieważ mieliśmy problemy z użyciem standardowego 21. Jako login użyj swojej nazwy użytkownika lub KLUCZA API, jeśli używasz mailjet, a jako hasło podane hasło konto lub KLUCZ API. Następnie możesz zdecydować, czy chcesz używać bezpiecznego uwierzytelniania, czy nie. Jeśli wysyłasz poufne dane, powinno to być koniecznością. Możesz wybrać pomiędzy TLS lub SSL. Czasami sprawdź szczegóły, wybierając uwierzytelnianie, aby użyć innego portu. W naszym przypadku nie zaznaczyliśmy pola uwierzytelniania, ale jeśli korzystasz z

mailjet, możesz bez problemu wybrać TLS. W trzeciej zakładce, Wiadomość e-mail, pozostawiamy treść wiadomości e-mail pustą, ale w temacie piszemy: ETL zakończony OK. I pomijamy ostatnią zakładkę, jakby proces zakończył się dobrze, nie chcemy dołączać żadnego pliku. Po tym jesteśmy gotowi do uruchomienia pracy. Ponieważ wciąż go testujemy, możemy uruchomić go bezpośrednio z łyżki. Po zakończeniu otrzymamy wiadomość e-mail podobną do poniższej w naszym polu docelowym, z tematem „ETL zakończył się OK”:

Job:

JobName : KettleJob

Directory : /

JobEntry : sendMail OK

Message date: 2017/01/21 12:26:03.005

Previous results:

Job entry Nr : 2

Errors : 0

Lines read : 0

Lines written : 0

Lines input : 0

Lines output : 0

Lines updated : 0

Lines rejected : 0

Script exist status : 0

Result : true

Path to this job entry:

KettleJob

KettleJob : : start : Start of job execution

(2017/01/21 12:17:45.716)

KettleJob : : START : start : Start of job

execution (2017/01/21 12:17:45.717)

KettleJob : : START : [nr=0, errors=0,

exit_status=0, result=true] : Job execution finished

(2017/01/21 12:17:45.717)

KettleJob : : Odoo to Staging : Followed

unconditional link : Start of job execution

(2017/01/21 12:17:45.718)

KettleJob : : Odoo to Staging : [nr=1, errors=0,

exit_status=0, result=true] : Job execution finished

(2017/01/21 12:21:55.690)

KettleJob : : Staging to DWH : Followed link

after success : Start of job execution (2017/01/21

12:21:55.690)

KettleJob : : Staging to DWH : [nr=2, errors=0,

exit_status=0, result=true] : Job execution finished

(2017/01/21 12:26:02.996)

KettleJob : : sendMail OK : Followed link after

success : Start of job execution (2017/01/21

12:26:02.996)

Czas skonfigurować teraz krok błędu. W takim przypadku zachowamy zasadniczo tę samą konfigurację, ale dołączymy również dziennik do wiadomości e-mail, aby śledzić zgłoszony błąd. W tym celu klikamy dwukrotnie w kroku sendMail KO, a na czwartej zakładce zaznaczamy pole wyboru Dołącz plik(i) do wiadomości i wybieramy opcje Log, Error Line i Error. To nie wszystko; musimy dokonać niewielkiej zmiany w naszym procesie ETL. Aby móc wychwycić dowolny błąd w dowolnym kroku, musimy skonfigurować wyjście błędu w każdym kroku. Ponieważ w tym momencie nie jesteśmy zbytnio zainteresowani innym sposobem radzenia sobie z naszymi dwiema transformacjami, możemy połączyć oba wyjścia błędów z tym samym wysyłaniem kroku poczty. W tym celu utwórz nowy link, przeciągając i upuszczając krok wyjściowy pierwszej transformacji w kanwie, ten, który przenosi dane między bazą danych Odoo a obszarem przejściowym w naszej bazie danych hurtowni danych i połącz go z naszym krokiem sendMail KO, na przykład na rysunku

Następnie na każdym etapie transformacji wymagana jest nowa niewielka zmiana. Musimy włączyć śledzenie dziennika w obu krokach transformacji. Kliknij dwukrotnie pierwszy, przejdź do zakładki ustawień rejestrowania i kliknij przycisk Specify Logfile? Zaznacz to pole i wpisz żądaną nazwę pliku

dziennika w polu Nazwa pliku dziennika, a następnie pozostaw rozszerzenie txt w polu tekstowym Rozszerzenie pliku dziennika. Pole rozwijane Poziom dziennika umożliwia wybranie szczegółowości dziennika wyjściowego. Na razie zostawimy to, co jest (podstawowe), ale tutaj możesz skonfigurować szczegóły dziennika błędów, który chcesz otrzymać. Teraz wyobraź sobie, że wystąpił błąd w procesie ETL. W naszym przypadku jest to problem z połączeniem z bazą danych. Ale może to być wiele innych problemów, takich jak zduplikowane wartości, błędne dane lub niewłaściwy format danych, brakująca tabela, problem z połączeniem z bazą danych lub jakikolwiek inny problem, jaki możesz sobie wyobrazić. Jeśli ETL został zaprojektowany poprawnie, otrzymamy ten błąd w naszej skrzynce pocztowej. Tym razem tematem e-maila będzie „The ETL zakończył KO”, a treść będzie mniej więcej taka:

Job:

JobName : KettleJob

Directory : /

JobEntry : sendMail KO

Message date: 2017/01/21 12:50:29.672

Previous results:

Job entry Nr : 1

Errors : 0

Lines read : 0

Lines written : 0

Lines input : 0

Lines output : 0

Lines updated : 0

Lines rejected : 0

Script exist status : 0

Result : false

Path to this job entry:

KettleJob

KettleJob : : start : Start of job execution

(2017/01/21 12:50:23.022)

KettleJob : : START : start : Start of job

execution (2017/01/21 12:50:23.022)

KettleJob : : START : [nr=0, errors=0,

exit_status=0, result=true] : Job execution finished

(2017/01/21 12:50:23.022)

KettleJob : : Odoo to Staging : Followed

unconditional link : Start of job execution

(2017/01/21 12:50:23.023)

KettleJob : : Odoo to Staging : [nr=1, errors=1,

exit_status=0, result=false] : Job execution finished

(2017/01/21 12:50:29.653)

KettleJob : : sendMail KO : Followed link after

failure : Start of job execution (2017/01/21

12:50:29.654)

Zauważ, że wartość wyniku jest teraz fałszywa. Oznacza to, że zadanie nie zostało ukończone prawidłowo. Interesująca jest również ta linijka: „Odoo to Staging: [nr=1, errors=1, exit_status=0,result=false]”, wyjaśniająca, w której transformacji wystąpił błąd. W naszym przypadku było to w pierwszym. Ale to nie wszystko! Nadal mamy załączony plik do przejrzenia w celu uzyskania dalszych informacji, plik dziennika błędów. Jeśli dokładnie sprawdzimy e-mail, który właśnie otrzymaliśmy, powinniśmy znaleźć załączony plik, zwany dziennikiem błędów, ponieważ jest to nazwa, którą dodaliśmy wcześniej. Jeśli sprawdzimy zawartość pliku, uzyskamy więcej informacji o błędzie, przy czym pierwsze wiersze są bardzo pouczające:

2017/01/21 12:50:23 - Odoo to Staging - Loading

transformation from XML file [OdooToStg.ktr]

2017/01/21 12:50:23 - KettleTransf - Dispatching

started for transformation [KettleTransf]

2017/01/21 12:50:23 - sale_order.0 - ERROR (version

6.1.0.1-196, build 1 from 2016-04-07 12.08.49 by

buildguy) : An error occurred, processing will be stopped:

2017/01/21 12:50:23 - sale_order.0 - Error occurred

while trying to connect to the database

2017/01/21 12:50:23 - sale_order.0 -

2017/01/21 12:50:23 - sale_order.0 - Error

connecting to database: (using class

org.postgresql.Driver)

2017/01/21 12:50:23 - sale_order.0 - Connection

refused.

Przegląd w narzędziach wiersza polecenia PDI

Jak na razie dobrze. Do tego momentu widzieliśmy, jak uruchomić zadanie PDI (lub transformację) i przesłać nam status jego wykonania. To bardzo interesujące, ale wciąż istnieje jedno przypuszczenie, które zakładaliśmy do tego momentu. Uruchamiamy zadania bezpośrednio z łyżki, ale zwykle serwer nie ma interfejsu graficznego, a co więcej, chcemy zaplanować automatyczne uruchamianie zadań, więc posiadanie kogoś do uruchamiania zadań nie wchodzi w grę. Aby rozwiązać ten problem, PDI ma kilka narzędzi wiersza poleceń, które pomagają nam uruchamiać zadania z powłoki (zarówno w systemach Windows, jak i Unix). Te dwa narzędzia nazywają się Pan i Kitchen. Pan to narzędzie do uruchamiania samodzielnych transformacji (bez żadnego zadania), a Kitchen służy do uruchamiania zadań (które mogą zawierać kilka transformacji). Zobaczymy próbkę obu, używając najpierw Kitchen, aby uruchomić naszą pierwszą transformację, a później Pan, aby uruchomić całe zadanie, które zaprojektowaliśmy na początku tej części.

Uruchamianie transformacji z wiersza poleceń za pomocą funkcji Pan

Istnieje kilka parametrów, które można przekazać do pliku wykonywalnego Pan, aby uruchomić transformację. Te parametry można łatwo zwizualizować, jeśli wywołasz plik wykonywalny bez żadnych parametrów. Najważniejsze z nich to:

Options:

/rep : Repository name

/user : Repository username

/pass : Repository password

/trans : The name of the transformation

to launch

/dir : The directory (dont forget the

leading /)

/file : The filename (Transformation in

XML) to launch

/level : The logging level (Basic,

Detailed, Debug, Rowlevel, Error, Minimal, Nothing)

/logfile : The logging file to write to

Uwaga: Możesz sprawdzić wszystkie opcje wraz z wyjaśnieniem, wywołując narzędzie lub czytając dokumentację Pan na stronie internetowej PDI tutaj: https://help.pentaho.com/Documentation/6.0/0L0/0Y0/070/000

W zależności od tego, czy korzystamy z repozytorium, czy nie, musimy wywołać plik wykonywalny Pan z zestawem określonych parametrów. Rzućmy okiem na łatwiejszą opcję, czyli wywołanie ktr (pliku transformacji PDI) bezpośrednio z pliku wykonywalnego. W naszym przypadku transformacja nosi

nazwę OdooToStg.ktr i jest to pierwsza transformacja naszego zadania, kopiująca dane z Odoo do obszaru przemieszczania. Ponieważ nie musimy przekazywać żadnych parametrów do naszej transformacji, określimy tylko dziennik wyjściowy, aby sprawdzić, czy transformacja przebiegła pomyślnie. Można to osiągnąć za pomocą przełącznika /logfile. Zachowamy domyślny poziom rejestrowania. Weź pod uwagę, czy jesteś w systemie Windows lub Unix, ponieważ musisz wywołać odpowiednią wersję Pan (Pan.bat lub pan.sh). Tak więc w przypadku systemu Windows wywołanie powinno wyglądać następująco:

Pan.bat /file OdooToStg.ktr /logfile OdooToStg.log

Dane wyjściowe powinny wyglądać mniej więcej tak (zostały obcięte, aby pokazać tylko najważniejszą część):

…...

2017/01/21 16:07:37 - Pan - Start of run.

2017/01/21 16:07:38 - KettleTransf - Dispatching

started for transformation [KettleTransf]

2017/01/21 16:07:39 - stg_account_tax.0 - Connected

to database [bibook staging] (commit=1000)

2017/01/21 16:07:39 - stg_hr_employee.0 - Connected

to database [bibook staging] (commit=1000)

2017/01/21 16:07:39 - Table output.0 - Connected to

database [bibook staging] (commit=1000)

2017/01/21 16:07:39 - stg_product_template.0 -

Connected to database [bibook staging] (commit=1000)

…

2017/01/21 16:11:50 - Table output.0 - Finished

processing (I=0, O=3653, R=3653, W=3653, U=0, E=0)

2017/01/21 16:11:50 - Pan - Finished!

2017/01/21 16:11:50 - Pan - Start=2017/01/21

16:07:37.716, Stop=2017/01/21 16:11:50.629

2017/01/21 16:11:50 - Pan - Processing ended after

4 minutes and 12 seconds (252 seconds total).

2017/01/21 16:11:50 - KettleTransf -

2017/01/21 16:11:50 - KettleTransf - Step Generate

Day Rows.0 ended successfully, processed 3653 lines. (

14 lines/s)

…

Jeśli w pewnym momencie założymy repozytorium (bez względu na to, czy założymy plik repozytorium lub repozytorium bazy danych) lub nawet repozytorium Pentaho (tylko nowsze wersje), musimy podać parametry repozytorium, aby uruchomić naszą transformację. W tym momencie przypomina to uruchomienie samodzielnego pliku z niewielkimi zmianami. Polecenie brzmiałoby:

Pan.bat /rep pdirepo /user admin /trans OdooToStg

A to uruchomi naszą Transformację z repozytorium. Zauważ, że nie podaliśmy żadnego hasła do repozytorium. Jeśli go podałeś, powinieneś dodać przełącznik /password i podać hasło do repozytorium; w przeciwnym razie Pan zgłosi błąd. Mając to jasne, przejdźmy do uruchamiania zleceń w Kitchen!

Uruchamianie zadań z wiersza poleceń z Kitchen

Uruchamianie zadań w Kitchen jest bardzo podobne do uruchamiania transformacji w Pan. Opcje są w zasadzie takie same jak w Pan. Aby uruchomić zadanie w naszym repozytorium, użyjemy następującego polecenia:

Kitchen.bat /rep pdirepo /user admin /job KettleJob

W przypadku otrzymania komunikatu typu: „BŁĄD: Kuchnia nie może być kontynuowana, ponieważ zadanie nie mogło zostać załadowane”, prawdopodobnie oznacza to, że uruchamiamy zadanie, które nie istnieje lub, co bardziej prawdopodobne, błędnie wpisaliśmy nazwę zadania, lub podajemy nieprawidłowe repozytorium. Aby wyświetlić listę posiadanych repozytoriów, możemy użyć następującego polecenia:

Kitchen.bat /listrep

Co zwróci nam następujący wynik:

List of repositories:

#1 : pdirepo [pdirepo] id=KettleFileRepository

Sprawdziliśmy, czy repozytorium jest w porządku. Aby sprawdzić katalog i nazwę zadania, możemy użyć kolejnych dwóch zestawów poleceń:

Kitchen.bat /rep pdirepo /user admin /listdir

Kitchen.bat /rep pdirepo /user admin /dir /

/listjobs

A ostatnie polecenie pokaże, że mamy tylko jedno zadanie o nazwie KettleJob. I ponowne uruchomienie go z poprawnym katalogiem:

Kitchen /rep pdirepo /user admin /dir / /job

KettleJob

Co w rzeczywistości jest tym samym, co pierwszy fragment kodu, który widzieliśmy, tak jakby zadanie było umieszczone w katalogu głównym „/”, nie trzeba jawnie mówić o tym programowi. Po chwili zobaczymy:

2017/01/22 11:09:27 - KettleJob - Job execution

finished

2017/01/22 11:09:27 - Kitchen - Finished!

2017/01/22 11:09:27 - Kitchen - Start=2017/01/22

11:01:00.126, Stop=2017/01/22 11:09:27.297

2017/01/22 11:09:27 - Kitchen - Processing ended

after 8 minutes and 27 seconds (507 seconds total).

Co oznacza, że nasza praca zakończyła się sukcesem.

Uwaga: Możesz przeglądać zawartość repozytorium z Spoon, przechodząc do menu Narzędzia, następnie do Repozytorium, a następnie klikając Eksploruj. Stamtąd powinieneś być w stanie wyświetlić wszystkie swoje obiekty w repozytorium oraz hierarchie plików i folderów. Co więcej, z podmenu Repozytorium możesz importować (lub eksportować) samodzielne pliki do i z repozytorium.

Widzieliśmy, jak wchodzić w interakcje z narzędziami wiersza poleceń PDI. Jest to ważne, aby wiedzieć, ponieważ umożliwi to efektywne planowanie zadań z dowolnego harmonogramu zadań, o którym możemy pomyśleć. W pozostałej części rozdziału zobaczymy, jak planować zadania za pomocą dwóch najpopularniejszych dostępnych menedżerów zadań: Crona dla systemów Unix i Harmonogramu zadań Windows dla systemów Windows.

Planowanie zadań w Harmonogramie zadań

Nadszedł czas, aby zobaczyć, jak zintegrować nasze zadania w przepływie. Zazwyczaj kilka procesów będzie uruchomionych w pewnym momencie dnia, zwykle poza godzinami szczytu. Tak więc część BI musi przebiegać w jakiejś harmonii między nimi. Ponieważ każda firma (mała czy duża) ma inne potrzeby, przedstawimy łatwy i darmowy sposób na uruchomienie Twoich zleceń. Oczywiście ma to swoje ograniczenia, więc jeśli potrzebujesz ustalić złożone zależności między zadaniami lub potrzebujesz ściślejszej kontroli i integracji wszystkich procesów w swojej organizacji, być może będziesz musiał poszukać bardziej wyspecjalizowanych narzędzi. Ale dla naszych potrzeb będziemy trzymać się harmonogramu zadań dołączonego do systemu operacyjnego.

Planowanie zadania PDI w systemie Windows

W systemie Windows mamy harmonogram zadań. Aby utworzyć nowe zadanie, kliknij „Utwórz podstawowe zadanie”. Nazwij zadanie według własnego uznania i wprowadź dowolny zrozumiały opis. Po tym oknie pojawi się kolejne z prośbą o powtórzenie zadania. W naszym przypadku wybierzemy Codziennie. Czas rozpoczęcia i cykl zostaną ustawione w następnym oknie dialogowym. Ponieważ chcemy, aby dane były ładowane poza godzinami pracy, ustawimy datę początkową na 00:00, czyli w zasadzie o północy, a cykl pozostawimy na co 1 dzień, zgodnie z domyślnym wypełnieniem. W następnym kroku mamy trzy możliwości wyboru rodzaju zadania, które chcemy zaplanować. Pierwszy uruchamia program, drugi wysyła wiadomość e-mail, a trzeci wyświetla komunikat. Zostawimy pierwszy jako wybrany i ponownie klikniemy Dalej. Ostatni krok jest najważniejszy. Harmonogram w zasadzie pyta nas, który program uruchomić, jakie parametry przekazać do pliku wykonywalnego i skąd powinien zostać uruchomiony. Błędne skonfigurowanie tego doprowadzi do problemów z rozpoczęciem naszej pracy, więc upewnij się, że informacje są wypełnione poprawnie. Wypełnimy pola następującymi informacjami:

* W polu Program/skrypt po prostu dodaj ścieżkę i nazwę pliku wykonywalnego Kitchen. W systemie Windows będzie to coś podobnego do tego: D:\dataintegration\Kitchen.bat

* W polu argumentów umieścimy wszystkie przełączniki, których użyliśmy podczas ręcznego wywoływania Kitchen. Są to: /rep pdirepo /user admin /dir / /job KettleJob

* Na początku w sekcji umieścimy katalog, w którym PDI jest nieskompresowane. W naszym przypadku jest to: D:\data-integration

Kliknij teraz obok, aby zakończyć dodawanie zadania. Zanim będziemy mogli uruchomić zadanie, chcemy mieć pewność, że zadanie zostanie uruchomione niezależnie od tego, czy jesteśmy połączeni z systemem. W tym celu kliknij dwukrotnie nazwę zadania w Harmonogramie zadań i przejdź do pierwszej zakładki o nazwie Ogólne. Upewnij się, że wybrałeś opcję Uruchom, niezależnie od tego, czy użytkownik jest zalogowany, czy nie. Jeśli Twoje zadanie wymaga uprawnień administracyjnych, upewnij się, że zaznaczyłeś pole wyboru Uruchom z najwyższymi uprawnieniami. Następnie kliknij OK, a zostaniesz poproszony o podanie poświadczeń, aby uruchomić to zadanie. Zalecamy utworzenie konta do zadań zautomatyzowanych tylko z wymaganymi uprawnieniami systemowymi. Dzięki temu masz zaplanowaną pracę. W wyniku ostatniego uruchomienia można zobaczyć wynik ostatniego wykonania. Zamiast opisu jest pokazany w formacie kodu 0x0 operacja zakończyła się pomyślnie, 0x1 (błąd) i 0x41301 zadanie jest aktualnie uruchomione, najczęstsze wyniki.

Uwaga: jeśli potrzebujesz potężnego harmonogramu zadań w systemie Windows, sprawdź nncron. Nncron jest kompatybilny z crontab i ma wersję lite, która jest wystarczająco potężna i jest darmowa. Strona internetowa to: http://www.nncron.ru/index.shtml

Planowanie zadania PDI w systemach Unix/Linux

Aby zaplanować pracę w systemie Linux, wybraliśmy crona. Są inne, ale wszystkie systemy Unix są dostarczane z cronem. Chociaż może być konieczne nauczenie się kilku parametrów konfiguracyjnych, aby to zrozumieć, jest to bardzo potężne narzędzie o bardzo małej powierzchni. Aby wyświetlić zawartość swojego crona, możesz użyć następującego polecenia:

crontab -l

Jeśli nie masz nic zaplanowanego, całkiem możliwe, że zobaczysz komunikat podobny do następującego:

no crontab for bibook

Plik crontab ma określony format. Zasadniczo plik działa w wierszach. Każdy wiersz określa zadanie do uruchomienia, a następnie każdy wiersz jest podzielony na sześć kolumn. Każda kolumna może być oddzielona od następnej spacją lub tabulatorem. Zalecamy używanie tabulatora jako separatora dla łatwiejszego zrozumienia. Kolumny są używane do:

* Pierwsza kolumna określa minutę, w której zadanie zostanie uruchomione.

* Druga kolumna określa godzinę uruchomienia zadania.

* Trzecia kolumna określa dzień miesiąca, w którym zadanie zostanie uruchomione.

* Czwarta kolumna określa miesiąc, w którym zadanie zostanie uruchomione.

* Piąta kolumna określa dzień tygodnia, w którym zadanie zostanie uruchomione.

* Szósta i ostatnia kolumna określa zadanie do uruchomienia wraz z jego parametrami.

Jak zauważyłeś, możesz chcieć mieć zakres możliwych wartości lub na przykład używać symboli wieloznacznych. jeśli chcesz wykonywać pracę co godzinę, wyjaśnienie wszystkich 24 godzin doby

będzie niepraktyczne. Na szczęście cron ma kilka symboli wieloznacznych, w zasadzie myślnik i gwiazdkę. Pierwszy służy do określenia zakresu, gdzie drugi oznacza „wszystko”. Na przykład, jeśli chcemy określić, że zadanie ma być uruchamiane codziennie o 22:00 w nocy, możemy użyć następującego wyrażenia:

0 22 * * * task_to_run

Na przykład, aby uruchomić nasze zadanie o północy, możemy zaplanować je w następujący sposób:

0 0 * * * /path_do_pdi/kitchen.sh /rep pdirepo

/user admin /dir / /job KettleJob

Poniższa lista wypunktowana bardzo pomoże w zrozumieniu tego:

* * * * * command to execute

Gdzie:

* Pierwsza pozycja określa minuty (0-59) lub wszystkie (*).

* Druga gwiazdka określa godzinę (0-23) lub wszystkie (*).

* Trzecia pozycja określa dzień miesiąca (1-31) lub wszystkie (*).

* Czwarta pozycja określa miesiąc (1-12) lub wszystkie (*).

* Piąta pozycja określa dzień tygodnia (0-6), rozpoczynający się w niedzielę i kończący się w sobotę lub wszystkie (*).

* A na koniec polecenie do wykonania.

Uwaga: Dla tych, którzy mają problemy z wierszem poleceń, dostępny jest edytor graficzny dla crona o nazwie gnome-schedule. Jest dołączany do wielu systemów, w tym Ubuntu. Oczywiście nie zawsze jest możliwe posiadanie GUI na serwerze, ale jeśli nie jest to możliwe, nadal możesz użyć narzędzia na innym komputerze, a następnie wykonać polecenie crontab -l, aby wyświetlić zawartość pliku cron i skopiuj go bezpośrednio na serwer po wykonaniu polecenia crontab -e, które otworzy bieżący plik crontab dla zalogowanego użytkownika w edytorze tekstu (zwykle vi).

Uruchamianie zadań konserwacji bazy danych

Uruchamianie zautomatyzowanych zadań otwiera nowe okno. Zwykle wykonujemy nie tylko zadania produkcyjne, ale także niektóre zadania konserwacyjne, które muszą być uruchamiane od czasu do czasu, aby zapewnić optymalne działanie wszystkich systemów. Możesz mieć pewne zadania porządkowe, powtarzające się zadania poprawiające wydajność i inne zadania. Porozmawiamy przez chwilę o zadaniach konserwacyjnych, a dokładniej zadaniach konserwacyjnych baz danych. Jak widzieliśmy, nasza hurtownia danych działa bez zarzutu. Ale zawartość naszych tabel w bazie danych zmienia się od czasu do czasu. Przydatne może być wykonanie na nim pewnych zadań konserwacyjnych, aby zapewnić jego optymalne działanie. Oprócz proaktywnej pracy DBA lub kogokolwiek, kto nadzoruje część administracyjną, możemy zaplanować pewne powtarzające się zadania: zbieranie statystyk, aby upewnić się, że nasze zapytania są dobre, jak widzieliśmy w części 7, przeprowadzać kontrole integralności, aby potwierdzić, że nie mamy uszkodzonych danych, i zadanie tworzenia kopii zapasowej. Zacznijmy od prawdopodobnie najważniejszego, zadania tworzenia kopii zapasowej.

Tworzenie kopii zapasowej naszej bazy danych MySQL/Maria DB

Aby wykonać kopię zapasową naszej bazy danych (MySQL lub MariaDB), dostępne jest dołączone narzędzie, które pomaga. Narzędzie nazywa się mysqldump i zasadniczo jest to zrzut bazy danych, czyli umieszcza w pliku całą zawartość bazy danych. W zależności od parametrów jakie podamy programowi możemy otrzymać strukturę, dane lub strukturę + dane. Zwykle będziemy używać tego drugiego, ponieważ w przypadku przywracania potrzebujemy obu rzeczy.

The usage of mysqldump is the following:

bibook@bibook:~$ mysqldump

Usage: mysqldump [OPTIONS] database [tables]

OR mysqldump [OPTIONS] --databases [OPTIONS]

DB1 [DB2 DB3...]

OR mysqldump [OPTIONS] --all-databases

[OPTIONS]

For more options, use mysqldump –help

Najpierw utwórzmy katalog eksportu, którego kopię zapasową utworzymy w bezpiecznej lokalizacji:

bibook@bibook:~$ sudo mkdir /opt/export

bibook@bibook:~$ sudo chmod 777 /opt/export

Następnie możemy uruchomić mysqldumpl i umieścić docelowy eksport w nowo utworzonym folderze. Ponownie pamiętaj o prawidłowym utworzeniu kopii zapasowej tego folderu; w przeciwnym razie, jeśli będziesz musiał przywrócić bazę danych, będziesz miał kłopoty. Zacznijmy najpierw od inscenizacji. Chociaż do przemieszczania potrzebujemy tylko struktury, ponieważ dane w tej bazie danych mają być niestabilne, możemy również wykonać kopię zapasową danych.

bibook@bibook:/opt/export$ mysqldump -u bibook -p -

-create-options staging >

/opt/export/db_staging_initial.sql

Enter password:

A po wprowadzeniu hasła kopia zapasowa będzie gotowa. Możemy zrobić ls -la, aby wyświetlić plik kopii zapasowej i potwierdzić, że kopia zapasowa została utworzona pomyślnie:

bibook@bibook:/opt/export$ ls -la /opt/export

total 376

drwxrwxrwx 2 root root 4096 Jan 28 14:46 .

drwxr-xr-x 4 root root 4096 Jan 28 14:41 ..

-rw-rw-r-- 1 bibook bibook 375954 Jan 28 14:46

db_staging_initial.sql

Następnie musimy zrobić to samo dla bazy danych hurtowni danych, dwh:

```
bibook@bibook:/opt/export$ mysqldump -u bibook -p -

-create-options dwh > /opt/export/db_dwh_initial.sql

Enter password:

bibook@bibook:/opt/export$ ls -la

total 644

drwxrwxrwx 2 root root 4096 Jan 28 14:48 .

drwxr-xr-x 4 root root 4096 Jan 28 14:41 ..

-rw-rw-r-- 1 bibook bibook 274328 Jan 28 14:48

db_dwh_initial.sql

-rw-rw-r-- 1 bibook bibook 375954 Jan 28 14:46

db_staging_initial.sql
```

Jak zauważysz, to trochę dziwne, że nasz plik kopii zapasowej obszaru pośredniego jest większy niż nasz magazyn danych. Przez pierwsze dni może to być całkiem możliwe, ponieważ istnieją dane, których nadal nie włączamy do hurtowni danych, a nasza hurtownia danych zawiera tylko kilka danych, ale z czasem nie powinno to mieć miejsca. Teraz nadszedł czas, aby zaplanować to w naszym harmonogramie zadań. Zdecydowaliśmy się na codzienną kopię zapasową, więc aby upewnić się, że wiemy, która kopia zapasowa należy do każdego dnia, musimy oznaczyć nazwę pliku znacznikiem czasu. Może to nie być konieczne, jeśli przenosisz plik poza system, ale jeśli chcesz zostawić kopię, konieczne jest uniknięcie nadpisania. Ponownie ta strategia zależy od rozmiaru hurtowni danych i innych czynników. Jeśli jesteśmy na maszynie uniksowej, edytujmy nasz plik crontab:

```
crontab -e
```

i wpisz następujące polecenie:

```
0 0 * * * mysqldump -

u bibook -p --create-options dwh >

/opt/export/db_dwh_$(date -d "today"

+"%Y%m%d%H%M").sql
```

W ten sposób zaplanujemy tworzenie kopii zapasowej codziennie o godzinie 00:00, a do pliku zostanie dodana data i czas. Następnie sprawdź, czy wszystko jest na swoim miejscu za pomocą polecenia crontab -l. Jutro powinieneś zobaczyć tam pierwszą kopię zapasową.

Przeprowadzaj kontrole i optymalizacje w bazie danych

Widzieliśmy, jak wykonać kopię zapasową naszej bazy danych na wypadek katastrofy. Zobaczmy, jaką proaktywną pracę możemy wykonać, aby uniknąć problemów, takich jak zawieszone ładunki lub zbyt długie procesy z powodu niskiej wydajności. Aby sprawdzić stan naszych tabel w bazie danych i analogicznie do polecenia SQL CHECK TABLE nazwa_tabeli, mamy do dyspozycji narzędzie wiersza poleceń o nazwie mysqlcheck. Składnia jest bardzo podobna do mysqldump:

```
bibook@bibook:/opt/export$ mysqlcheck -u bibook -p
```

--database dwh

Enter password:

dwh.t_f_sales

OK

dwh.t_l_category

OK

dwh.t_l_currency

OK

dwh.t_l_cust_country

OK

dwh.t_l_customer

OK

dwh.t_l_emp_department

OK

dwh.t_l_emp_level

OK

dwh.t_l_employee

OK

dwh.t_l_month

OK

dwh.t_l_parent_category

OK

dwh.t_l_product

OK

dwh.t_l_quarter

OK

dwh.t_l_status

OK

dwh.t_l_year

OK

dwh.t_r_time

OK

Interesujące może być zaplanowanie zadania polegającego na uruchamianiu tego polecenia od czasu do czasu (może codziennie przed utworzeniem kopii zapasowej) lub raz w tygodniu, w zależności od aktywności w bazie danych, i upewnienie się, że dane wyjściowe są prawidłowe. Jeśli wykryjemy tabele, które wymagają naprawy, możemy to zrobić za pomocą tego samego narzędzia wiersza poleceń.

mysqlcheck -u bibook -p --auto-repair --check -- baza danych dwh

Uwaga: Aby uzyskać więcej opcji i różnych przełączników poleceń narzędzia mysqlcheck, możesz przeczytać najnowsze informacje o narzędziu tutaj: https://dev.mysql.com/doc/refman/5.7/en/mysqlcheck.html.

Oprócz kontroli integralności możemy również wykonywać inne rodzaje zadań konserwacyjnych. Obejmują one optymalizację tabel i indeksów w naszej bazie danych oraz zbieranie świeżych statystyk dotyczących danych w naszych tabelach, które pomogą optymalizatorowi bazy danych w wyborze odpowiedniego planu wykonania, gdy zostanie przedstawione zapytanie. Zasadniczo istnieją dwa narzędzia, których możemy użyć, które w rzeczywistości są dowiązaniami symbolicznymi do tego samego narzędzia mysqlcheck, które widzieliśmy wcześniej. Te dwa to mysqlanalyze i mysqloptimize. Przyjrzyjmy się obu. Narzędzie analityczne analizuje rozkład danych w tabeli. Jak wyjaśniono wcześniej, wiedza o danych obecnych w tabeli bardzo pomaga silnikowi bazy danych. Bez tych informacji optymalizator działa na ślepo i przyjmuje założenia, które mogą okazać się bardzo niedokładne. Aby pomóc optymalizatorowi, możemy z wyprzedzeniem przeanalizować tabele. Ogólną zasadą jest ponowna analiza tabeli tylko wtedy, gdy zmieniło się więcej niż % danych, więc nie ma potrzeby codziennego uruchamiania tego polecenia. Możemy zaplanować np. cotygodniowe zadanie analizy wszystkich tabel w hurtowni danych. Ponieważ odczyt tabel jest zablokowany, musimy upewnić się, że ta operacja jest wykonywana poza godzinami szczytu, zwykle po wykonaniu ETL. Polecenie i oczekiwane dane wyjściowe są następujące:

bibook@bibook:/opt/export$ mysqlanalyze -u bibook -

p --database dwh

Enter password:

dwh.t_f_sales

OK

dwh.t_l_category

OK

dwh.t_l_currency

OK

dwh.t_l_cust_country

OK

dwh.t_l_customer

OK

dwh.t_l_emp_department

OK

dwh.t_l_emp_level

OK

dwh.t_l_employee

OK

dwh.t_l_month

OK

dwh.t_l_parent_category

OK

dwh.t_l_product

OK

dwh.t_l_quarter

OK

dwh.t_l_status

OK

dwh.t_l_year

OK

dwh.t_r_time

OK

To narzędzie wiersza poleceń jest analogiczne do polecenia ANALYZE TABLE nazwa_tabeli, które można uruchomić z poziomu klienta MySQL lub Maria DB.

Wreszcie, możemy zoptymalizować tabele. Jeśli nasza baza danych zawiera wiele danych przychodzących i wychodzących, musimy od czasu do czasu reorganizować nasze tabele, aby upewnić się, że przestrzeń jest mądrze wykorzystywana. Niewłaściwa organizacja tabel oznacza nie tylko, że zajmą one znacznie więcej miejsca na dysku niż potrzeba, ale także w konsekwencji, gdy pobierzemy te dane, baza danych będzie dłużej działać, ponieważ ma więcej miejsc, w których można szukać. Mówimy więc nie tylko o oszczędności miejsca, ale także o przyspieszeniu zapytań. Polecenie reorganizacji tabeli to mysqloptimize. Możemy uruchomić polecenie, aby zoptymalizować całą listę tabel w bazie danych. Wywołanie i oczekiwane dane wyjściowe są pokazane tutaj:

bibook@bibook:/opt/export$ mysqloptimize -u bibook

-p --database dwh

Enter password:

dwh.t_f_sales

note : Table does not support optimize, doing

recreate + analyze instead

status : OK

dwh.t_l_category

note : Table does not support optimize, doing

recreate + analyze instead

status : OK

dwh.t_l_currency

note : Table does not support optimize, doing

recreate + analyze instead

status : OK

dwh.t_l_cust_country

note : Table does not support optimize, doing

recreate + analyze instead

status : OK

dwh.t_l_customer

note : Table does not support optimize, doing

recreate + analyze instead

status : OK

dwh.t_l_emp_department

note : Table does not support optimize, doing

recreate + analyze instead

status : OK

dwh.t_l_emp_level

note : Table does not support optimize, doing

recreate + analyze instead

status : OK

dwh.t_l_employee

note : Table does not support optimize, doing

recreate + analyze instead

status : OK

dwh.t_l_month

note : Table does not support optimize, doing

recreate + analyze instead

status : OK

dwh.t_l_parent_category

note : Table does not support optimize, doing

recreate + analyze instead

status : OK

dwh.t_l_product

note : Table does not support optimize, doing

recreate + analyze instead

status : OK

dwh.t_l_quarter

note : Table does not support optimize, doing

recreate + analyze instead

status : OK

dwh.t_l_status

note : Table does not support optimize, doing

recreate + analyze instead

status : OK

dwh.t_l_year

note : Table does not support optimize, doing

recreate + analyze instead

status : OK

dwh.t_r_time

note : Table does not support optimize, doing

recreate + analyze instead

status : OK

Jak widać, w zależności od silnika, którego używamy z bazą danych, wykonywana operacja może się nieznacznie różnić. W naszym przypadku pracujemy z tabelami InnoDB. Podczas korzystania z silnika InnoDB polecenie zamiast tego odtwarza i analizuje tabelę. Ta operacja może zająć dużo czasu, zwłaszcza jeśli tabele są większe. Zalecamy uruchomienie tego polecenia w okresie konserwacji lub w momencie, gdy możemy sobie pozwolić na wyłączenie naszej bazy danych na jakiś czas. Podobnie jak

w poprzednich narzędziach, to polecenie jest analogiczne do polecenia OPTIMIZE TABLE nazwa_tabeli, które można uruchomić z klienta MySQL lub Maria DB.

Wniosek

Prawie kończymy naszą podróż i chociaż widzieliśmy już rdzeń i podstawy projektowania rozwiązania BI, ta Część dodała dodatkową wartość, przedstawiając niektóre zadania konserwacyjne, które są bardzo ważne, aby platforma działała. Dodawanie kontroli w naszych zadaniach ETL jest podstawowym zadaniem. Ludzie oczekują, że będą wiedzieć, kiedy coś pójdzie nie tak. Powinni być informowani i podejmować szybkie działania w celu rozwiązania problemów, które mogą pojawiać się od czasu do czasu. Zobaczyliśmy również, jak poprawić wydajność bazy danych i wykonać kilka podstawowych zadań konserwacyjnych w bazie danych, które poprawią kondycję naszego rozwiązania. Zobaczyliśmy również, jak zaprogramować zadania, aby wykonywały się cyklicznie. Podczas cyklu życia rozwiązania będziesz musiał zaprojektować wiele z nich, więc jest to dobry moment, aby zacząć z tym ćwiczyć. Korzystanie z harmonogramów zadań dołączonych do systemów operacyjnych może być dobre, ale w pewnym momencie będziesz musiał stworzyć bardziej złożone zależności między zadaniami i prawdopodobnie będziesz musiał opracować kilka skryptów lub wypróbować bardziej złożone narzędzia, ale na w tym momencie zarówno cron, jak i program planujący Windows są dobrymi narzędziami na początek. W kolejnych częściach zobaczymy, jak ustrukturyzować więcej niż jedno środowisko do pracy; i jak zapewnić interakcję środowisk między nimi, aby nie tracić opracowań lub danych oraz jak wprowadzić pewien porządek, co jest szczególnie potrzebne, gdy nad tymi samymi rzeczami pracuje więcej niż jedna osoba w tym samym czasie. Zobaczymy również, jak wdrożyć rozwiązanie, które zbudowaliśmy na chmurze, jak zaplanować małą pojemność i wybrać rozwiązanie, które pomoże nam skalować, jeśli nasza firma zacznie działać coraz lepiej.

11. Przejście do środowiska produkcyjnego

Stosując się do naszych zaleceń i kroków wdrożeniowych do tego momentu, będziesz mieć środowisko, które będzie miało wszystkie wymagane elementy do pracy, relacyjną bazę danych na miejscu, system ETL, który pobiera informacje ze źródłowego ERP, platformę BI, która pozwala analizować informacje w prosty sposób oraz platformę MOLAP, która pomaga w określeniu celów na przyszły rok. Ale jesteśmy pewni, że na tym nie poprzestaniesz. Będziesz chciał dodać nową analizę, nowe pola, nowe obliczenia, nowe atrybuty lub nowe hierarchie do swojego systemu. Możliwe, że chcesz zwiększyć objętość, większą szczegółowość, szczegóły dzienne zamiast miesięcznych lub osiągnąć ten sam poziom informacji, co transakcyjny system ERP. Możesz wymagać wielu modyfikacji w swoim systemie analitycznym, które mogą ingerować w strukturę bazy danych i już utworzone raporty, z których korzysta wielu klientów. Aby mieć pewność, że masz wiarygodne dane do zaoferowania każdemu, kto analizuje je w twoim systemie, najlepszym scenariuszem jest posiadanie różnych środowisk do analizy danych i nowych rozwiązań, aby uniknąć niepożądanych wpływów. Ocenimy również niektóre kwestie, które należy wziąć pod uwagę przy przechodzeniu do środowiska produktywnego, takie jak wielowymiarowość serwera, definicja kopii zapasowej, monitorowanie, zarządzanie bezpieczeństwem, wysoka dostępność lub konfiguracja odzyskiwania po awarii. Podobnie jak w pozostałej części, niektóre z tych podsekcji mogłyby wymagać napisania całej książki poświęconej każdej z nich, ale robimy tylko krótkie wprowadzenie, aby wyjaśnić cel każdej sekcji i niektóre zalecenia, ale bez wchodzenia szczegółowo o każdym temacie.

Scenariusz wielośrodowiskowy

Najczęstszym stanem we wszystkich instalacjach jest posiadanie więcej niż jednego środowiska dla każdego komponentu układu serwera. W rzeczywistości jest to ogólna zasada dla wszystkich systemów; to nie jest coś czystego z BI. Gdy wielu użytkowników zacznie uzyskiwać dostęp do systemu do dowolnej funkcjonalności, o której myślisz, nie możesz po prostu testować operacji, nowych wersji oprogramowania, nowych rozwiązań lub zmian w konfiguracji w tym systemie, ponieważ możesz wpłynąć na codzienną pracę całej firmy. W przypadku BI, gdy system działa, a użytkownicy uzyskują do niego dostęp, będziemy również chcieli mieć różne środowiska, aby uniknąć wpływu na nasze modyfikacje systemu używanego przez użytkowników końcowych. Normalną sytuacją jest posiadanie od dwóch do pięciu scenariuszy różnych systemów, jednak najczęstszym podejściem jest posiadanie trzech obszarów pracy. Uważamy, że szczególnie w BI wystarczą trzy środowiska, aby komfortowo pracować ze wszystkimi wymaganymi funkcjonalnościami.

Produkcja: Środowisko to będzie wykorzystywane do rzeczywistej codziennej pracy wszystkich użytkowników końcowych platformy. Będzie to główne środowisko pod względem wymagań dotyczących użytkowania, rozmiaru i dostępności.

Symulacja / Testy akceptacji użytkownika / Integracja / Testowanie: Można znaleźć wiele nomenklatur dla tego środowiska, ale to te, które znaleźliśmy częściej u klientów, których odwiedziliśmy. Tutaj niektórzy kluczowi użytkownicy końcowi będą mogli potwierdzić, że opracowane nowe wymagania są zgodne z wymaganymi przez nich specyfikacjami.

Rozwój: W tym środowisku programiści będą przeprowadzać modyfikacje wymagane dla każdego wymagania użytkownika (lub historyjki użytkownika; pamiętaj o koncepcjach Scruma z Części 2).

W następnych sekcjach zobaczymy, jak tworzyć te scenariusze, jakie kwestie należy wziąć pod uwagę, oraz zalecenia dotyczące użytkowania na podstawie tego, co odkryliśmy podczas naszego doświadczenia.

Wdrażanie produktywnego środowiska

Aby osiągnąć cel posiadania wielu środowisk, będziesz mieć różne opcje; jeśli serwery/komponenty, których użyłeś do zainstalowania wszystkich komponentów, działają poprawnie, możesz uznać je za produktywne, a następnie wdrożyć nowe środowiska do pracy jako symulacyjne i rozwojowe. Może to być prawidłowa procedura, jeśli złagodzisz jej główne niedogodności:

Wydajność sprzętowa : jeśli budowałeś tę platformę na serwerach bez wystarczającej pojemności, ponieważ tylko próbowałeś, powinieneś ponownie rozważyć tę strategię, o ile wydajność produkcyjna musi pasować do wymagań wydajnościowych twojego środowiska. W następnej sekcji omówimy plany pojemności serwerów.

Test i błąd: Jest całkiem możliwe, że podczas instalacji wymaganego rozwoju narzędzi wykonałeś wiele instalacji testowych, wypróbowałeś różne komponenty i rozwiązania, przeprowadzałeś testy podczas modelowania danych, opracowywania narzędzia BI, procesu ETL, i jest to wysoce zaleca się używanie czystego środowiska jako produktywnego; więc jeśli nie byłeś tak czysty i masz wiele obiektów testowych na swojej platformie, może lepiej jest wybrać świeże serwery i zainstalować i wdrożyć tam tylko wymagane komponenty.

Bezpieczeństwo: Zwykle zasady bezpieczeństwa wymagane przez Twoją firmę w środowiskach programistycznych nie są tak rygorystyczne, jak te wymagane w środowiskach produkcyjnych, więc możliwe jest, że będziesz wymagać przeglądu dostępów, procedur instalacyjnych lub użytkowników usług używanych do uruchamiania różnych aplikacji, które będziemy biegać.

Separacja sieci: w związku z poprzednim tematem możliwe jest, że zasady bezpieczeństwa Twojej firmy wymagają oddzielnych sieci dla środowisk produkcyjnych i programistycznych, więc jeśli nie zapytałeś o to podczas tworzenia serwerów, mogły one zostać utworzone w sieci deweloperskiej i nie można ich przenieść do sieci produkcyjnej.

Wszystkie te tematy mogą dotyczyć Ciebie w większym lub mniejszym stopniu w zależności od polityki bezpieczeństwa Twojej firmy, złożoności sieci lub wymagań serwerowych dla Twojego rozwiązania BI, dlatego generalnie wolimy wdrażać od zera środowisko produkcyjne o wystarczającej pojemności, tylko z wymagane komponenty do analizy i przestrzeganie wszystkich zaleceń i zasad, które wymagają produktywnego środowiska. W każdym razie dobrą możliwością jest rozpoczęcie od użycia istniejącego systemu ze wszystkimi tymi ograniczeniami, aby wiedzieć, jakie wykorzystanie platformy jest wykonywane przez użytkowników, jakie ma wymagania dotyczące wydajności, oraz zebranie statystyk z istniejącego systemu w celu prawidłowego stworzenia produktywnego środowiska zwymiarowany.

Plan pojemności serwera

Jednym z głównych tematów, które należy wziąć pod uwagę, przygotowując środowisko produkcyjne, jest zapewnienie mu wymaganej pojemności, aby spełnić Twoje oczekiwania dotyczące wydajności i pamięci masowej. Stworzenie dobrego planu wydajności jest czymś dość złożonym, ponieważ wymaga analizy wielu zmiennych, takich jak okresowość ETL, wymagane procesy równoległe, rodzaj wykonanych raportów, liczba wykonanych raportów, liczba oczekiwanych użytkowników, liczba użytkowników równoległych, maksymalna liczba równoległych wykonań w godzinach szczytu; a większość z tych zmiennych jest dość skomplikowana do przewidzenia podczas tworzenia systemu od zera. Trzeba wziąć pod uwagę również skrzyżowane zależności. Twój system ETL będzie miał wpływ na źródłową i docelową bazę danych, a pojemność docelowej bazy danych musi uwzględniać oczekiwaną wydajność ETL i wykonanie raportu. Możesz mieć lokalizacje w różnych krajach z różnymi ramami czasowymi i możesz mieć równoległe procesy ETL w nocy jednego kraju, wpływając na codzienne

wykonywanie raportów rano innego kraju. Oczywiście łatwiej jest przewidzieć wszystkie te zmienne, jeśli masz poprzednie dane z istniejącego systemu, jak skomentowano w poprzedniej sekcji, ale jest całkiem możliwe, że ich nie masz, więc spróbujmy zdefiniować listę kroków do wykonania w planie mocy. Zobaczmy najpierw ogólne rozważania, a następnie wprowadzimy zmienne, które mogą wpływać na każdy z komponentów platformy. Ogólnie rzecz biorąc (w rzeczywistości te rozważania dotyczą nie tylko BI, jeśli zarządzasz systemami innego typu) będziemy musieli wziąć pod uwagę następujące kwestie:

* Głównym celem planu wydajności jest określenie liczby potrzebnych serwerów (w przypadku środowiska klastrowego) oraz pojemności każdego z nich pod względem pamięci RAM, procesora i miejsca na dysku.

* Musimy skupić się na maksymalnym obciążeniu szczytowym, aby mieć platformę o odpowiednich wymiarach.

* Zalecane jest (choć czasem nie jest to możliwe) uzgodnienie z użytkownikami przyszłych projektów i nadchodzących wymagań.

* Jeśli nie masz tych wymagań podczas planowania pojemności, przejrzyj je za każdym razem, gdy nowy projekt pojawi się w twoim systemie.

* Historia danych wymagana do zapisania w hurtowni danych to krytyczny parametr, który może mieć wpływ na wszystkie komponenty, wielkość bazy danych oraz wydajność procesora ETL i BI.

* Wykorzystanie istniejących systemów do zebrania części danych z testów jednostkowych pozwoli na ekstrapolację wyników na ogólne potrzeby środowiska produkcyjnego.

Planowanie pojemności serwera bazy danych

W celu zdefiniowania planowania pojemności dla komponentu Baza danych konieczne będzie zebranie informacji o różnych KPI:

* Średni rozmiar wiersza dla większych tabel (zwykle zawierających fakty): możesz zebrać te informacje z istniejącego systemu.

* Oczekiwana liczba wierszy na miesiąc/rok: jeśli w istniejącym systemie masz prawdziwe dane, będziesz mógł wiedzieć, ile miejsca zostanie wykorzystane, mimo że możesz nie mieć załadowanej całej historii w swojej bazie danych. Jeśli tak nie jest i masz tylko przykładowe dane użytkownika do zbudowania aplikacji, będziesz potrzebować analizy środowiska źródłowego, aby spróbować zrozumieć, jak duża może być liczba otrzymanych wierszy na podstawie ekstrakcji lub bezpośrednich zapytań do bazy danych ERP.

* Oczekiwana historia hurtowni danych: zgodnie z komentarzem w poprzedniej sekcji historia hurtowni danych będzie miała bezpośredni wpływ na wielkość używaną przez tabele faktów.

* Weź pod uwagę, że prawdopodobnie będziesz potrzebować tabel tymczasowych do zapisywania różnych etapów procesu, wymiarów i tabel przeglądowych, które będą miały domyślny współczynnik wzrostu, który nie powinien być tak duży jak tabele faktów; musisz mieć wystarczająco dużo miejsca na niektóre kopie tabel do tworzenia kopii zapasowych, tabele robocze i tak dalej.

* Będziesz także potrzebować wolnego miejsca na dysku dla kopii zapasowej, tymczasowego obszaru tabel do uruchamiania zapytań oraz rozmiar systemu i katalogu dla schematów technicznych.

* Pod względem wymaganego procesora będziesz musiał przeanalizować szczyt równoległych procesów uruchomionych w bazie danych dla procesu ETL oraz szczyt raportów z BI działających w godzinach dziennych.

Planowanie pojemności serwera ETL

Twój system ETL zapewni Ci informacje dostępne w bazie danych podczas niektórych procesów ładowania zawierających różne instrukcje aktualizacji/wstawiania w schemacie bazy danych. Aby poprawnie przeanalizować, jakie wymagania ma ten system, musisz wiedzieć:

Okres ładowania danych: wymagania nie są takie same, jeśli informacje są ładowane codziennie, co tydzień lub co miesiąc.

Okno ładowania danych: możemy mieć wiele zależności od naszych codziennych procesów i musimy znać oczekiwania biznesowe w zakresie dostępności danych, aby dopasować się do wymagań biznesowych. Wyjaśniono na przykładzie, że jeśli zamierzasz wdrożyć system codziennego raportowania dla swojego zespołu sprzedaży, a oni zaczną odwiedzać klientów o 9:00 rano, będziesz potrzebować dziennego obciążenia dla tego procesu przed 8:00, jeśli automatyczne raportowanie proces trwa godzinę. W codziennej analizie sprzedaży będziesz chciał mieć dane o towarach zafakturowanych do dnia poprzedniego, aby Twój proces nie mógł rozpocząć się wcześniej niż o godzinie 0:00. Ale możesz także polegać na pewnym procesie rozliczeniowym, który kończy się o 2:00, zadaniu tworzenia kopii zapasowej ERP lub bazy danych, które kończy się o 3:00 lub jakimkolwiek innym procesie, który można połączyć. Musisz więc upewnić się, że proces ETL mieści się w dostępnym oknie czasowym.

Ilość danych: Będziesz musiał oszacować ilość danych, które zostaną przeniesione przez system, aby poprawnie zdefiniować jego pojemność. Do tej analizy możesz przyjąć jako odniesienie bieżącą ilość danych ETL i ich przepustowość (wiersze na sekundę).

Procesy równoległe: Musisz wziąć pod uwagę wszystkie procesy równoległe, które będą wykonywane przez ETL, aby zdefiniować pojemność serwera, aby zdefiniować maksymalną liczbę wierszy na sekundę, którymi będziesz musiał zarządzać.

Planowanie pojemności serwera BI

W przypadku serwera BI, o ile jest to bezpośredni interfejs z użytkownikami, będziesz musiał skoncentrować swoje wysiłki na analizie profilu wykonania użytkownika.

Liczba użytkowników: Przy planowaniu wydajności istotne jest, aby wiedzieć, ilu użytkowników ma znajdować się w naszym systemie oraz ilu z nich będzie równolegle wykonywać raporty w systemie (i zwykle w bazie danych).

Liczba raportów: musisz oszacować liczbę i rozmiar raportów dziennych wymaganych przez użytkowników w ramach ogólnej analizy dziennej, a także skupić się na szczytowym czasie wykonywania.

Oczekiwany czas wykonania: Ta wartość może mieć wpływ również na bazę danych, o ile w zależności od strategii BI jest całkiem możliwe, że największa część czasu wykonania leży po stronie serwera bazy danych.

Uwaga: krytycznym punktem do przeanalizowania podczas definiowania naszej topologii sieci, który może mieć wpływ na wrażenia użytkownika końcowego, jest opóźnienie sieci. Ogólnie rzecz biorąc, zdecydowanie zaleca się, aby wszystkie komponenty naszej topologii sieci (ETL, BI i baza danych) były

dość zbliżone pod względem opóźnień sieciowych. Ale w przypadku opóźnienia między serwerem BI a użytkownikami końcowymi może to być bardziej skomplikowane, ponieważ będziesz mieć użytkowników łączących się z wielu źródeł i środowisk. Jest to szczególnie ważne, gdy łączysz się z połączeń zewnętrznych, aby upewnić się, że wydajność VPN jest wystarczająco dobra, aby zapewnić klientowi prawidłowe wrażenia użytkownika końcowego.

Planowanie pojemności serwera MOLAP

W przypadku serwera bazodanowego MOLAP będziemy wymagać od Ciebie zastosowania mixu rozważań pomiędzy serwerem bazodanowym a serwerem BI, ale biorąc pod uwagę, że zdefiniowany rozmiar zostanie przemnożony również przez ilość scenariuszy, które będziemy mieli. Zwykle będzie to wymagało mniej równoległych obciążeń i zapytań niż baza danych hurtowni danych, ale będziemy musieli przeanalizować również liczbę użytkowników, liczbę raportów i liczbę równoległych wykonań zapytań.

Uwaga: wykonanie dobrego planowania pojemności może być bardzo trudne i stać się procesem krytycznym, ale można zminimalizować tę krytyczność procesu, jeśli można wybrać skalowalne serwery z możliwością dodawania dysków, procesorów i gniazd pamięci RAM lub serwerów wirtualnych, które można łatwo dostosować do przychodzących potrzeb. W następnym rozdziale zobaczymy również inną alternatywę, czyli wykorzystanie systemów chmurowych do lokalizacji serwerów.

Koszty licencji i pomocy technicznej

Wyobraźmy sobie możliwą sytuację. Uruchomiłeś swój system BI rok temu. Twój system działa poprawnie; masz zestaw dziennych obciążeń zdefiniowanych w ETL, które działają idealnie na czas, z dokładnością danych, bez żadnych incydentów przez ponad sześć miesięcy, ale nagle Twój proces ETL zwraca wyjątek. Nic się nie zmieniło, dane są poprawnie zapisane, nie ma niespójności w danych, a wszystko, co możesz sprawdzić lub ocenić, wydaje się być w porządku i nie znajdujesz żadnego przydatnego wpisu w swoich badaniach w Google. Twoi użytkownicy sił sprzedaży wdrożyli system oparty na tym procesie, którego używają każdego dnia podczas wizyt u klientów w celu przygotowania warunków handlowych i umów, które mogą wykorzystać w swoich negocjacjach i wymagają wyodrębnienia informacji na potrzeby swoich codziennych wizyt. Będą cię mocno naciskać, abyś miał dostęp do informacji, że błędny proces ETL opóźnia się. Co możesz teraz zrobić? Kogo o to zapytać? Jedną z możliwości jest poproszenie o wsparcie jakiejś firmy konsultingowej, która może pomóc w rozwiązaniu problemu, ale być może nie są one dostępne w niektóre dni. Inną możliwością jest zatrudnienie wsparcia dostawcy, aby pomógł ci w tej sytuacji rozwiązać lub obejść problem. Inna sytuacja. Wyobraźmy sobie, że Twój system BI wdrożony z Microstrategy Desktop lub QlikSense odniósł prawdziwy sukces, masz wielu użytkowników, którzy mają go zainstalowanego na swoich laptopach, więc mogą z niego korzystać za darmo, ale nie mogą dzielić się informacjami między sobą, wydajność wykonania nie są wystarczająco dobre i wymagają wdrożenia pewnych zabezpieczeń danych lub priorytetyzacji zadań, aby w pierwszej kolejności zająć się menedżerami w Twojej organizacji. Tego rodzaju działania można rozwiązać, jeśli przejdziesz do instalacji na serwerze, zamiast używać tylko klientów stacjonarnych do uruchamiania raportów, więc w takim przypadku być może trzeba pomyśleć o zakupie licencji, aby móc przejść do licencjonowanego środowiska opartego na serwerze. Koszt licencji może być ważnym rachunkiem do rozważenia, ale możliwe jest, że będziesz musiał przejść do tego scenariusza, jeśli Twoje rozwiązanie wystarczająco się rozwinie i stanie się wystarczająco ważne dla firmy. W takim przypadku będziesz musiał ocenić ROI (zwrot z inwestycji), który zapewni to rozwiązanie, abyś mógł uzasadnić ten ruch.

Uwaga: zdecydowanie zalecamy oszacowanie kosztów licencji i pomocy technicznej narzędzi, które zamierzasz zainstalować bezpłatnie, jako ważnej zmiennej przy wyborze każdego narzędzia. Być może

widzisz, że jakieś narzędzie jest całkiem atrakcyjne, gdy jest bezpłatne, ale wtedy nie możesz uzyskać budżetu od kierownictwa firmy na przejście do licencjonowanego środowiska i musisz odbudować rozwiązanie przy użyciu innego narzędzia tylko za ten koszt.

Dodawanie środowisk

Po zdefiniowaniu środowiska produkcyjnego nadszedł czas na utworzenie pozostałych środowisk, które będą potrzebne do codziennej pracy. We wstępie wymieniono jeszcze co najmniej dwa, rozwój, w którym stworzymy wszystkie nowe obiekty i modyfikacje istniejących, oraz symulację lub test, w którym przetestujemy te modyfikacje, które zostały wprowadzone w fazie rozwoju, z pewnymi istotnymi danymi, aby potwierdzić, że są działa zgodnie z oczekiwaniami. W zależności od rozmiaru projektu oraz liczby programistów i użytkowników uzyskujących dostęp do systemu, możliwe, że będziemy potrzebować czwartego środowiska, które możemy nazwać Hotfix, Patch, Incidental lub Support; na końcu nazwa nie jest istotna, istotne jest użycie. Widzieliśmy też instalacje z piątym środowiskiem, Sandboxem służącym do testowania nowych funkcji i wersji czy możliwości. Znaleźliśmy również kilka instalacji ze środowiskiem integracyjnym, w którym użytkownicy mogą przeprowadzać testy nowych rozwiązań przed przeniesieniem ich do UAT.

Tworzenie nowego środowiska można zasadniczo wykonać na dwa różne sposoby, zaczynając od pustego środowiska i instalując wszystkie komponenty, a następnie wdrażając wszystkie różne obiekty lub klonując środowiska na jednym lub wielu poziomach. W zależności od kierunku kopiowania i niektórych innych zmiennych preferujemy jedną lub drugą opcję:

Nowe czyste środowisko: Dzięki tej strategii tworzysz pusty system; instalujesz wszystkie wymagane komponenty; i wdrażasz na różnych warstwach, Database, ETL, BI i MOLAP wszystkie obiekty wymagane do analizy, używając skryptów tworzenia, jeśli masz taką możliwość. W ten sposób uzyskasz czystą instalację bez żadnych błędnych wpisów w rejestrze systemu; bez fałszywych plików; a także unikasz transportu obiektów testowych, nieużywanych obiektów, kopii zapasowych danych lub wszelkich obiektów, które nie są wyraźnie wymagane. Jest to najbardziej wydajna strategia pod względem wymiarowania i czyszczenia, ale jest najbardziej kosztowna, ponieważ wymaga pełnej kontroli nad każdym obiektem wymagań. Jest też całkiem możliwe, że będziesz wymagał zmiany niektórych obiektów konfiguracyjnych w systemie: oczywiście nazwy serwera i adresu IP, ale także połączeń ODBC, plików konfiguracyjnych, nazw baz danych itp.

W pełni sklonowane środowisko: Po przeciwnej stronie możesz sklonować wszystkie serwery za pomocą jakiegoś narzędzia do tworzenia kopii zapasowych, przywracając cały serwer na pustym. Jest to najszybszy sposób na stworzenie nowego środowiska pracy, ale może przeciągać niektóre problemy, nieużywane obiekty, nieprawidłowe instalacje, instalowane i odinstalowywane oprogramowanie, które przepuszcza niektóre niepotrzebne pliki w systemie, obiekty bazy danych, które nie są potrzebne itp. Oczywiście będziesz miał wtedy możliwość wyczyszczenia wszystkiego, czego nie potrzebujesz, ale zawsze będą jakieś obiekty, których nie wiesz, czy możesz usunąć, takie jak jakaś biblioteka we wspólnej ścieżce systemu.

Środowisko częściowo sklonowane: rozwiązaniem pośrednim, które naszym zdaniem jest dobrym podejściem, jest zainstalowanie całego wymaganego oprogramowania, skonfigurowanie wszystkich wymaganych połączeń, nazw i parametrów, a następnie przeniesienie obiektów aplikacji z jednego serwera na drugi, zwykle przy użyciu narzędzi klienckich które umożliwiają funkcje kopiowania/wklejania.

Izolacja środowisk

W zależności od polityki bezpieczeństwa Twojej firmy możliwe jest, że Twoje środowiska są całkowicie rozdzielone w różnych segmentach sieci bez widoczności między nimi. Izolacja środowisk pozwala na większą elastyczność systemów programistycznych, ułatwiając rozwój poprzez otwarcie bezpośredniego dostępu do serwerów dla deweloperów, nadanie uprawnień administratora lub otwarcie dostępu do Internetu z serwerów w fazie rozwoju, o ile przy oddzielnych sieciach wygrał incydent bezpieczeństwa w środowisku programistycznym nie wpływają na środowisko produkcyjne. Po zakończeniu całego rozwoju możesz przenieść zmodyfikowane obiekty do Testowania i Produkcji (lub zrobi to Twój zespół administratorów zgodnie z Twoimi instrukcjami) i nikt oprócz zespołu administratorów nie może mieć podwyższonego dostępu do żadnego serwera, bazy danych ani narzędzia. Ta izolacja wpłynie na strategię klonowania i konserwacji, ponieważ w oddzielnych sieciach nie będzie można bezpośrednio przenosić obiektów z jednego środowiska do drugiego; będziesz musiał użyć strategii wykorzystującej system pośredni, aby zlokalizować tymczasowe obiekty w ruchu z jednego środowiska do drugiego.

Zalecenia dotyczące wielu środowisk

W scenariuszu wielośrodowiskowym będziesz musiał wziąć pod uwagę, że Twoja praca polega nie tylko na opracowaniu wymagań pochodzących od użytkowników końcowych, ale także na przestrzeganiu pewnej strategii, która ułatwia przemieszczanie obiektów między środowiskami; w przeciwnym razie będziesz zmuszony powtórzyć niektóre prace rozwojowe lub adaptację w każdym transporcie obiektu. Aby ułatwić życie Tobie i Twojemu administratorowi, w kolejnych podsekcjach proponujemy zestaw zaleceń, których należy przestrzegać w przypadku rozwoju sytuacji.

Parametryzacja

Wszystko, co można sparametryzować, powinno być sparametryzowane. Rozwińmy trochę bardziej to twierdzenie. Musisz ocenić wszystko, co może się zmienić w różnych środowiskach, aby spróbować wyodrębnić z obiektu wszelkie odniesienia do stałej wartości i użyć zmiennej lub parametru zewnętrznego w stosunku do samego obiektu. Podajmy przykład: wyobraź sobie, że opracowujesz proces ETL. Sesję można skonfigurować tak, aby łączyła się z serwerem DevERP (środowiskiem programistycznym dla ERP) w celu odczytu oraz z DevDWH (serwerem hurtowni danych) przy użyciu narzędzia DevUSER w celu zapisywania danych. Jeśli to zrobisz, nie możesz po prostu skopiować i wkleić sesji do Testing, ponieważ po skopiowaniu będziesz wymagać zmiany parametrów połączenia na TestERP i TestDWH. Ale jeśli użyjesz zmiennych ${HOST} i ${DATABASE}, będziesz mógł je zastąpić zewnętrznie w pojedynczym pliku konfiguracyjnym lub przy użyciu zmiennych środowiskowych.

Ścieżki serwera

Powinieneś próbować instalować i lokalizować pliki zawsze w tej samej ścieżce na serwerach, aby ułatwić odwoływanie się do plików w różnych narzędziach, takich jak pliki źródłowe dla ETL, pliki parametrów omówione w poprzedniej sekcji, kostki MOLAP, repozytoria lub każdy inny plik używany w dowolnym narzędziu. Czasami nie jest to możliwe, ponieważ możesz mieć wiele wdrożeń na tym samym serwerze. Może się tak zdarzyć, ponieważ nie żyjemy we wspaniałym świecie z nieograniczonymi zasobami do wdrożenia twojego systemu, a czasami możemy zdefiniować logiczne środowiska współdzielące to samo środowisko fizyczne. Wyobraź sobie, że masz jeden serwer ETL dla środowisk programistycznych i testowych oraz dwa wdrożenia silnika ETL oparte na różnych folderach. Ilekroć chcesz przetransportować przepływ pracy, który ładuje plik w zdefiniowanej ścieżce, będziesz musiał to zmienić podczas przenoszenia go z programowania do testowania. W tym scenariuszu możesz również spróbować użyć ścieżek względnych, aby odnieść się do tego pliku z podstawowej ścieżki środowiska. Innymi słowy, możesz użyć ścieżek względnych podobnych do tych:

./SourceFiles/Area1/Sales_area1.txt

Lub używając zmiennych, które na końcu używają tej samej nazwy ścieżki pod względem konfiguracji:

$INSTALLPATH/SourceFiles/Area1/Sales_area1.txt

Wyrównanie nazw dla obiektów łączności

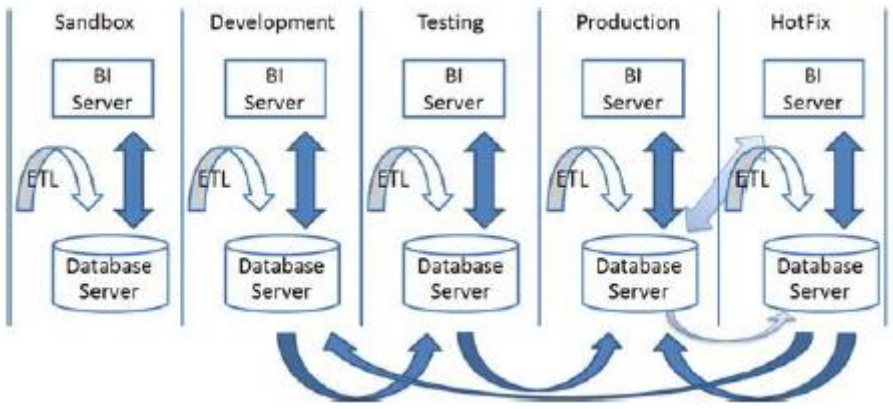
Jak widzieliśmy, podczas instalacji i konfiguracji wszystkich narzędzi będziesz musiał zdefiniować wiele obiektów łączności, takich jak połączenia ODBC, połączenia JDBC lub natywne ciągi połączeń, szczególnie w przypadku narzędzi ETL, BI i MOLAP łączących się w jakiś sposób z silnik bazy danych. Podobnie jak w przypadku pozostałych ścieżek i parametrów, tutaj również wygodnie jest wyrównać nazwy obiektów łączności, dzięki czemu można używać tego samego obiektu w wielu środowiskach, zmieniając tylko konfigurację połączenia w jednym obiekcie. Załóżmy, że tworzysz ODBC wskazujący na bazę danych ERP i nazywasz ją ERP_DB, a swojemu połączeniu ODBC zapisujesz w bazie danych Sales Datamart jako SALES_DB, bez względu na to, czy definiujesz je w środowisku programistycznym, testowym czy produkcyjnym, wskazując odpowiednio na ERP Development, ERP Testing i ERP Production and Sales Datamart Development, Testing i Production. Będziesz mógł korzystać z tego samego workflow odczytu z ERP_DB i zapisu do SALES_DB we wszystkich środowiskach.

Utrzymanie środowiska

Zdefiniowanie strategii wielośrodowiskowej wymaga początkowego wysiłku w celu skonfigurowania i skonfigurowania wszystkich serwerów, które mogą stanowić istotną część całego projektu BI, ale jeśli zastosujesz się do poprzednich zaleceń, będziesz w stanie utrzymać ją minimalizującą wysiłek wymagany do wykonania tego zadania. Ten wysiłek może być minimalny, ale nie oznacza to, że będzie niski, ponieważ będziesz mieć wiele zadań do wykonania, aby system działał w zdrowy sposób. Pozwól, że przedstawimy Ci również niektóre koncepcje związane z utrzymaniem platformy, której wymaga każda pojedyncza platforma.

Cykl życia obiektu

Po zainicjowaniu wielośrodowiskowego systemu możesz zacząć od nowych rozwiązań bez wpływu na rzeczywiste dane. Cykl życia dodawania nowych możliwości w systemie pokazano na rysunku.



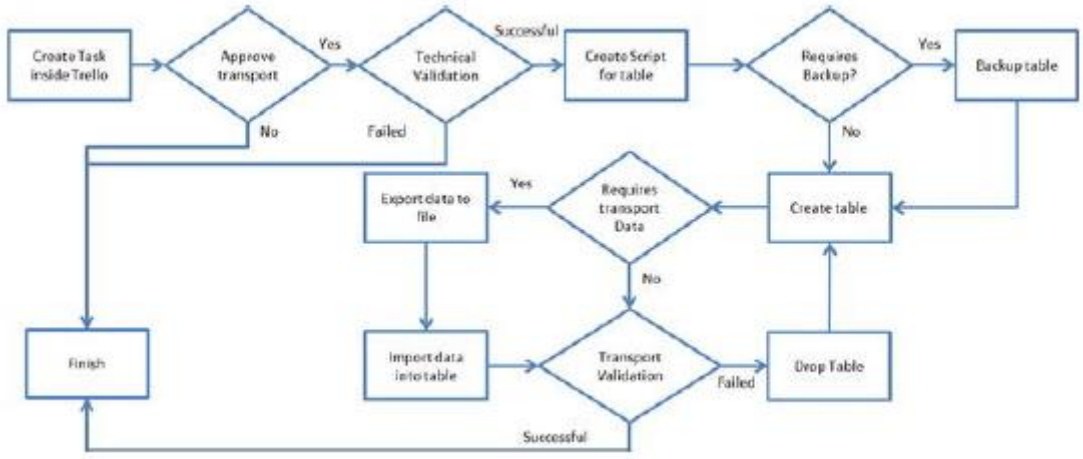
Deweloperzy będą wykonywać swoją pracę w środowisku programistycznym, a po zakończeniu wymagane obiekty zostaną przeniesione do środowiska testowego, gdzie kluczowi użytkownicy mogą zweryfikować, czy program jest zgodny z wymaganiami, a po zatwierdzeniu wymagane modyfikacje zostaną przetransportowane do środowiska produkcyjnego. Na tym wykresie widać również dwa środowiska więcej. Zgodnie z komentarzem, Hotfix będzie używany do rozwiązywania incydentów z

pewnymi zmianami w szybki sposób przy użyciu rzeczywistych lub zbliżonych do rzeczywistych danych. Moglibyśmy mieć środowisko Hotfix tylko dla BI lub tylko dla systemów ETL; w tym przypadku byłyby one połączone z produkcyjną bazą danych. Po rozwiązaniu problemu musimy przejść do środowiska produkcyjnego z wymaganymi modyfikacjami, ale musimy je również zastosować w systemie deweloperskim, aby nie utracić żadnej modyfikacji w przypadku przeniesienia rozwoju, który dotyczy tego samego obiektu. Jest również całkiem normalne, że środowisko Hotfix jest okresowo odświeżane danymi i strukturą produkcyjną.

Uwaga: Zmiana obiektów w innych niż oczekiwane środowiskach może spowodować błędy regresji podczas nadpisywania zmodyfikowanego obiektu na podstawie innej wersji obiektu. Wyobraź sobie, że modyfikujesz typ kolumny w tabeli bazy danych w środowisku produkcyjnym, a następnie tworzysz go ponownie za pomocą skryptu programistycznego, ponieważ dodałeś jeszcze dwie kolumny. Spowoduje to przywrócenie typu kolumny.

Procedura transportowa

Zdecydowanie zalecamy zdefiniowanie i przestrzeganie procedury transportowej dla każdego narzędzia w Twoim środowisku BI, która ułatwi cofanie zmian, zbadanie źródeł ewentualnych problemów lub po prostu śledzenie wprowadzonych zmian w Twoim środowisku. Sama procedura będzie z pewnością zależała od używanego narzędzia, ale może również zależeć od typu obiektu w każdym narzędziu i wymaganych opcji transportu. Powinieneś mieć listę kroków dla każdego typu, określającą ścieżki i lokalizacje obiektów pomocniczych oraz obowiązki każdego kroku. Na rysunku możesz zobaczyć przepływ pracy związany z transportem tabeli bazy danych.



W tym przepływie pracy możemy zobaczyć różne kroki:

1. Rozważamy wykorzystanie Trello do zarządzania zadaniami, więc pierwsze

krokiem będzie otwarcie zadania w Trello przez zespół programistów.

2. Właściciel projektu zatwierdzi transport lub nie.

3. Zespół administratorów sprawdzi, czy tabela jest zgodna ze standardami nazewnictwa,

czy ma poprawne parametry i typy kolumn itp.

4. Jeśli tabela jest zgodna ze standardami, wygenerujemy skrypt do utworzenia tabeli.

5. Po sprawdzeniu, czy wymagana jest kopia zapasowa, wykonamy ją, czy nie (jeśli tabela nie istnieje, to w ogóle nie miałoby sensu; jeśli istnieje, zespół programistów powinien określić), definiując procedurę tworzenia kopii zapasowej, którą moglibyśmy naprawić różne strategie również:

A. Skrypt tabeli do zastąpienia: określenie lokalizacji, w której ma zostać zapisany plik , nazewnictwo skryptów i plików.

B. Kopia tabeli z inną nazwą (_BCK na końcu) lub w kopii zapasowej Bazy danych.

C. Eksportuj tabelę za pomocą narzędzi bazy danych eksportu, definiując również plik eksportu , nazewnictwo i lokalizacja.

6. Następnie zespół administratorów przystępuje do tworzenia tabeli.

7. Jeśli tabela wymaga danych transportowych, powinniśmy wyeksportować/zaimportować te dane za pomocą pliku a plik tymczasowy, który wymaga również określenia nomenklatury i lokalizacji.

8. Po przetransportowaniu musimy zweryfikować, czy transport jest prawidłowy; inaczej odtwarzamy tabelęi i jeśli wszystko jest w porządku, kończymy proces.

To tylko przykład procedury transportowej, ale powinieneś ją zdefiniować dla wszystkich (przynajmniej większość typowych) typów obiektów do transportu między środowiskami.

Uwaga: Ważne jest, aby zaznaczyć w procedurze transportu włączenie kopii zapasowej zmodyfikowanych obiektów, aby móc cofnąć wszelkie wprowadzone zmiany w przypadku, gdy coś zawiedzie podczas transportu lub z powodu nieprawidłowego opracowania.

Okna transportowe

W ramach procedury transportowej zalecamy również umieszczenie okna transportowego, w którym należy realizować transporty z dwoma głównymi celami:

* Minimalizuj wpływ na użytkowników końcowych: wyobraź sobie, że chcesz przenieść zmianę w głównej tabeli faktów, która wymaga usunięcia i ponownego utworzenia, w tym nowych pól i ponownego załadowania całej historii. Pozwoliłoby to na niedostępność danych, a raporty, które je wysyłają, nie pokażą niczego, dopóki operacja transportu nie zostanie zakończona. Jeśli masz zdefiniowane okno transportowe (zwykle poza godzinami pracy lub wkrótce poza godzinami pracy), Twoi użytkownicy mogą wyodrębnić raport poza tym przedziałem czasowym.

* Zminimalizuj nakłady na transporty: Mając zdefiniowane okno do transportu obiektów, możesz zminimalizować nakłady pracy administracyjnej platformy poprzez ułatwienie organizacji pracy administracji.

Automatyzacja Transportu

Idąc dalej w swoim środowisku, możliwe, że dojdziesz do momentu, w którym będziesz musiał dostarczać coraz większą ilość transportów ze względu na rosnącą liczbę wydarzeń. Zalecamy zbadanie możliwości automatyzacji transportu w wierszu poleceń, aby zaimplementować łatwy sposób przenoszenia obiektów i danych między środowiskami, aby można było w jak największym stopniu zautomatyzować to zadanie transportowe, które może być powtarzalnym zadaniem, wymaganym, ale bez dodawania zbyt dużą wartość dla całego rozwiązania BI, którego głównym celem jest tylko analiza danych. Nie zrozum nas źle; ponieważ większość zadań technicznych jest wymagana w każdym środowisku, nieprawidłowa polityka transportowa może powodować wiele problemów, dlatego dość

ważne jest zdefiniowanie prawidłowej procedury, ale ostatecznym celem rozwiązania BI jest analiza danych, a nie prawidłowy transport obiektów między środowiskami . Istnieje sporo narzędzi typu open source lub darmowych, które mogą ci pomóc w tych celach, takich jak Puppet, Chef i biblioteka tkanin Pythona. Narzędzia te automatyzują działania i mogą je powtarzać na wielu komputerach. Są szczególnie interesujące, gdy trzeba powielić tę pracę w różnych środowiskach lub grupach maszyn.

Procedura testowa

Ważną częścią procedury transportowej jest określenie, jaka będzie metodologia testowania w celu sprawdzenia, czy Twój program poprawnie realizuje zdefiniowane wymagania biznesowe. Ważne jest również sprawdzenie, czy jakikolwiek obiekt lub modyfikacja pochodząca z rozwoju nie wpływa na istniejące funkcjonalności, zwykle znane jako testy regresji. Ta ostatnia część jest zwykle ignorowana, gdy masz zupełnie nowy projekt, ale nie lekceważ możliwości wzajemnych powiązań rozwoju, które wpływają na ciebie w produktywnym środowisku. Więc chociaż uważasz, że nowy rozwój nie ma nic wspólnego z tym, co jest na miejscu, nie przegap wykonania testów regresji, aby potwierdzić, że jest to całkowicie prawdziwe. Oto niektóre kwestie do rozważenia podczas definiowania procedury testowania:

* Zestaw definicji testów: które raporty, zapytania lub skrypty należy uruchomić, aby zweryfikować wszystkie testy.

* Zapewnij wykonanie testu: nie byłby to pierwszy raz, gdy znajdujemy pełną procedurę testową, ale nie towarzyszy temu odpowiedzialność.

* Zdefiniuj obowiązki: przypisz osobę odpowiedzialną za wykonanie testów, a także odpowiedzialność za analizę wyników testów. Jeśli nie jest to ta sama osoba, możesz zapewnić wykonanie testów.

* Testowanie powinno obejmować również walidację metodologii i standardów zdefiniowanych w tej książce, takich jak konwencje nazewnictwa lub wszelkie inne ograniczenia, które chcesz zastosować w swoim systemie.

Automatyzacja testów

Podobnie jak w przypadku procedury transportowej, kiedy mówimy o testowaniu, możemy zastosować te same względy. Testowanie jest wymaganym zadaniem technicznym, ale nie ma bezpośredniego wpływu na analizę biznesową. Będziemy więc zainteresowani jak największą automatyzacją wykonywania procedury testowej i walidacją tych wyników wykonania. O ile rozwijamy platformę BI, zdecydowanie sensowne jest automatyczne uwzględnianie wyników testów w niektórych analizach graficznych za pomocą naszej platformy BI, zapisywanie wyników w naszej bazie danych. Interesującymi darmowymi narzędziami do testowania są między innymi AutoIt i Selenium.

Narzędzia monitorowania

Gdy już przejdziesz do środowiska produkcyjnego i Twoi użytkownicy będą codziennie uzyskiwać dostęp do dostarczanych przez Ciebie informacji, ważne jest, aby zapewnić niezawodną usługę, która zapewni użytkownikom dostęp do informacji zawsze, gdy ich potrzebują. Ale jest to zadanie, którego trudno będzie wykonać, jeśli nie masz automatycznego sposobu sprawdzania, czy wszystkie powiązane usługi działają i czy nie ma bezpośredniego ryzyka wystąpienia problemów z powodu braku miejsca na dysku, przeciążenia zajęta pamięć lub stałe użycie procesora powyżej określonego progu. Jest całkiem możliwe, że masz już w swojej firmie zainstalowany system monitorujący, który sprawdza stan innych działających systemów, ale jeśli nie, zalecamy zapoznanie się z narzędziami open source. Na rynku dostępnych jest wiele narzędzi, ale tym, które uważamy za bardziej kompletne, jest Nagios, które

pomaga monitorować główne techniczne wskaźniki KPI serwerów, zapewniając również interfejs graficzny do analizy danych i możliwość powiadamiania Cię za pośrednictwem poczty elektronicznej lub telefonu komórkowego, gdy tylko wykryje jakikolwiek problem na serwerze. Możesz go znaleźć pod adresem:

https://www.nagios.org/projects/nagios-core/

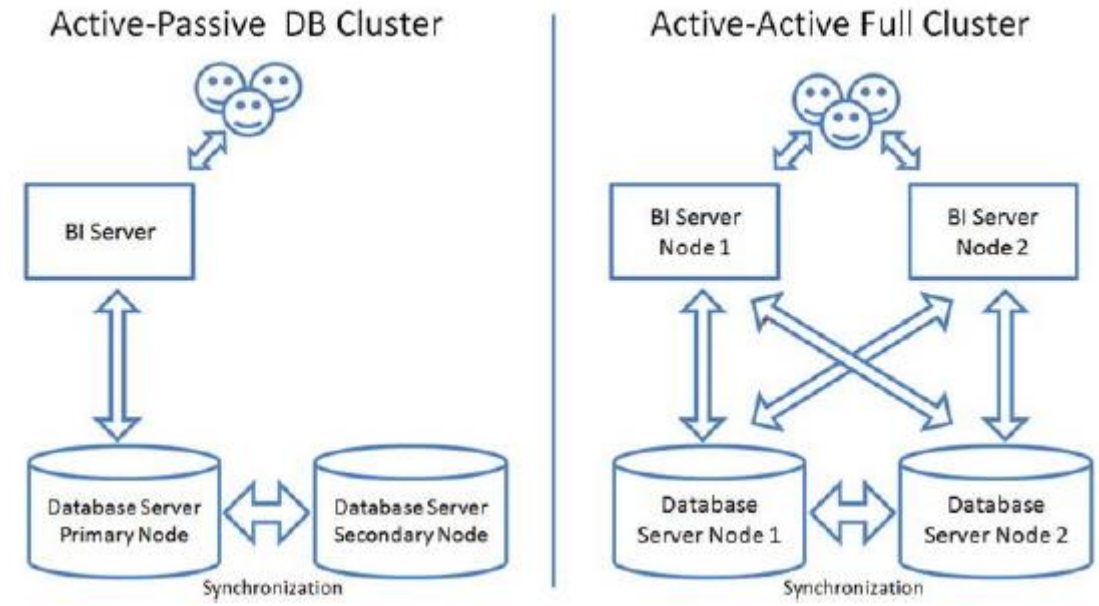
Duża dostępność

Kiedy uznasz swój serwer za produktywny, zastosujesz zestaw rozważań, które powinny zapewnić pewnej niezawodności platformie, takich jak unikanie modyfikacji kodu źródłowego w środowisku produkcyjnym, wymaganie walidacji nowego kodu w środowiskach programistycznych i testowych, ograniczanie dostępu do serwerów produkcyjnych do osób z profilami administracyjnymi lub korzystających z narzędzi monitorujących, jak Nagios skomentował kilka linijek wcześniej, aby zapewnić dostępność usługi. Jeśli Twoja usługa staje się coraz bardziej krytyczna, możesz pomyśleć o zaimplementowaniu konfiguracji wysokiej dostępności poprzez dostarczenie środowiska klastrowego. Ten rodzaj konfiguracji klastrowej ma sens tylko wtedy, gdy mówimy o instalacjach serwerowych, więc w przypadku narzędzi BI przedstawionych w tej książce, które w swojej bezpłatnej wersji zapewniają instalację klienta, nie ma sensu mówić o klastrowaniu. W przypadku narzędzia BI należy przejść na wersję komercyjną, aby mieć taką możliwość. W scenariuszu, który promujemy w całej książce, o wiele bardziej sensowne jest mówienie o klastrowaniu, gdy odnosimy się do części bazy danych. Zarówno MariaDB, jak i MySQL oferują możliwość konfiguracji swoich usług w klastrze. Aby upewnić się, że wiesz, czym jest klaster, pozwól nam szybko przedstawić Ci koncepcję klastra. Ostatecznie to tylko dwa lub więcej serwerów (znanych również jako węzły) uzyskujących dostęp do tych samych danych, co umożliwia dostęp do tych danych z obu węzłów. W ten sposób prawdopodobieństwo wyłączenia obu węzłów jest znacznie mniejsze niż w przypadku posiadania tylko jednego serwera, więc informacje będą dostępne przez większą część czasu. Jest to zasadniczo koncepcja wysokiej dostępności. Istnieją głównie dwa typy konfiguracji klastrów:

* Active-Passive: Jeden serwer świadczy usługi, a drugi węzeł jest zatrzymany, ale jest w stanie gotowości do sprawdzenia, czy serwer główny nie działa, więc musi zacząć świadczyć usługi użytkownikom. Ta opcja oferuje tylko wysoką dostępność, ale nie zwiększa wydajności środowiska.

* Aktywny-aktywny: Oba węzły na serwerze świadczą usługi, więc będziesz mieć możliwość dostępu do jednego lub drugiego węzła. Ta opcja zapewnia również większą pojemność, ponieważ możesz korzystać z obu serwerów jednocześnie, aby uzyskać dostęp do swoich danych.

Na rysunku



można zobaczyć dwa przykłady konfiguracji klastrów, pierwszy oparty na klastrze tylko komponentu Database w podejściu Active-Passive, gdzie serwer BI łączy się z głównym węzłem klastra bazy danych i węzłem drugorzędnym nie działa, czekając na awarie węzła podstawowego. Jeśli wystąpi awaria, węzeł drugorzędny staje się Aktywny, a węzeł główny wyłącza się jako pasywny, podczas gdy serwer BI automatycznie łączy się z węzłem drugorzędnym. Oba węzły muszą być zsynchronizowane, aby w przypadku awarii nie utracić żadnych danych. Tego rodzaju konfiguracja zwykle wymaga pewnego rodzaju współdzielonego dysku do przechowywania informacji z bazy danych. W drugim przykładzie z rysunku widać podwójne klastrowanie na poziomie serwera BI i serwera bazy danych. W tym przypadku serwery baz danych są uruchomione i działają w sposób niewyraźny dla obu serwerów BI. Serwery DB muszą być również zsynchronizowane, aby zachować spójność między nimi, o ile dostęp do obu jest możliwy z obu serwerów BI. Użytkownicy mogą również łączyć się z jednym lub z drugim węzłem. Aby to połączenie było przejrzyste dla użytkownika, zwykle stosuje się komponent równoważenia obciążenia, który przekierowuje żądania użytkownika do jednego lub drugiego węzła w zależności od obciążenia serwera.

Wykorzystanie testowania jako odzyskiwania po awarii produkcji

Jeśli Twoja aplikacja stała się wystarczająco krytyczna, możliwe, że będziesz potrzebować dostępu do informacji dostarczanych przez środowisko BI w prawie każdych możliwych warunkach, ponieważ jest to uważane za Krytyczne w Twoim portfolio aplikacji. W takim przypadku musisz uwzględnić nie tylko możliwość awarii serwera (węzeł dodatkowy pokryje dostępność), ale także możliwość awarii całego centrum danych w przypadku klęski żywiołowej, wypadku, wyłączenia zasilania, terroryzm, czy po prostu duża awaria sieci. Aby postępować zgodnie z polityką Disaster Recovery, musisz mieć dostępne środowisko w innym centrum danych niż to, w którym znajduje się środowisko produkcyjne. Oznacza to, że będziesz potrzebować dostępnego sprzętu na wypadek, gdyby był potrzebny, więc możliwe, że masz nieużywany sprzęt, który tylko czeka, jeśli w produkcyjnym centrum danych wystąpi jakiś duży incydent. Aby przypisać sobie zasługi za ten nieużywany sprzęt, powszechną strategią jest użycie go do zlokalizowania środowiska testowego, dzięki czemu można go użyć do walidacji użytkownika lub testów warunków skrajnych. Oczywiście oznacza to, że w przypadku katastrofy stracisz swoje środowisko testowe; ale oczywiście jest to coś, co w takim przypadku każdy może zaakceptować. W każdym razie istnieje kilka kwestii, które muszą spełniać Twoje środowisko Disaster Recovery, aby być prawdziwą alternatywą dla środowiska produkcyjnego:

* Kompletne środowisko: musimy mieć wszystkie komponenty dostępne w naszej lokalizacji DR, aby zapewnić pełną obsługę Twoich użytkowników. Wydaje się to oczywiste, ale nie byłby to pierwszy raz, kiedy widzimy DR z możliwością zlokalizowania bazy danych i komponentu BI, ale bez żadnego serwera dostępnego dla usługi ETL.

* Podobna wydajność: Niektóre parametry wymiarowości, które analizowaliśmy podczas planowania wydajności, muszą być podobne do tych w Twoim środowisku produkcyjnym. Możesz przetrwać w przypadku katastrofy z mniejszą mocą procesora, ale oczywiście będziesz potrzebować co najmniej takiej samej przestrzeni dyskowej jak środowisko produkcyjne, aby móc przywrócić całą infrastrukturę.

* Zaktualizowane informacje: konieczne będzie zreplikowanie informacji produkcyjnych do środowiska testowego. Może nie bezpośrednio do środowiska testowego, ale do jakiegoś urządzenia w tej samej lokalizacji co testowanie. Innymi słowy, produkcyjny system tworzenia kopii zapasowych musi znajdować się w tym samym centrum danych co środowisko testowe lub musi tam być replikowany.

* Minimalna odległość: nie ma dokładnych specyfikacji, ale zaleca się, aby oba centra danych były oddzielone minimalną odległością, aby mieć pewność, że katastrofa nie wpłynie na obie lokalizacje. Jest to szczególnie wskazane, aby uniknąć klęsk żywiołowych.

* Definicja parametrów RTO i RPO: Recovery Time Objective i Recovery Point Objective określają maksymalny czas, w którym Twoja firma może zaakceptować wyłączenie usługi oraz maksymalną utratę danych, którą można uznać za akceptowalną. W systemie BI dość często spotyka się definicje RTO/RPO 24/72, co oznacza, że będziesz potrzebować usługi BI z powrotem w ciągu 24 godzin, co pozwoli na utratę danych na 72 godziny, innymi słowy, gdy Twoja usługa zostanie wstecz, będziesz musiał uruchomić proces odzyskiwania ETL, który wypełni informacje z poprzednich 3 dni.

Tworzenie kopii zapasowej Twojej platformy

W poprzednich sekcjach komentowaliśmy, że aby dostępne było przywracanie po awarii, konieczne będzie posiadanie niektórych kopii zapasowych danych dostępnych w dodatkowym centrum danych, ale do tej pory prawie nie mówiliśmy o tworzeniu kopii zapasowych. Przejdźmy do szybkiego wyjaśnienia kilku zaleceń dotyczących konfiguracji kopii zapasowych w środowisku BI. Najpierw musimy zdefiniować typ kopii zapasowej wymagany dla każdego komponentu. Serwery BI i ETL zwykle mają instalację oprogramowania, która nie zmienia się, dopóki nie zostanie zainstalowana nowa wersja lub poprawka, oraz repozytorium, w którym zapisywane są obiekty używane w narzędziu. To repozytorium zwykle może być oparte na plikach lub bazach danych i jest całkiem możliwe, że w tym drugim przypadku repozytorium znajduje się na tym samym serwerze DB. W tym przypadku interesować nas będzie właśnie utrzymywanie migawki serwera po każdej instalacji oraz codzienna kopia zapasowa obiektów repozytorium, które zwykle zajmują niewielką ilość miejsca w porównaniu z całym serwerem lub całą bazą danych hurtowni. W przypadku bazy danych polityka tworzenia kopii zapasowych będzie się różnić w zależności od charakteru danych. Zwykle serwery baz danych będą wymagały codziennych kopii zapasowych, aby móc odzyskać obraz z poprzedniego dnia w przypadku uszkodzenia bazy danych lub awarii. Ale może się też zdarzyć, że część Twoich danych wejściowych będzie miała charakter transakcyjny, tzn. że nie będą ładowane w procesie dziennym w nocy, ale mogą być aktualizowane przez całą dobę o dowolnej porze, dlatego Twoja polityka tworzenia kopii zapasowych musi zapewnić Państwu możliwość przywrócenia stanu bazy danych w możliwie najkrótszym czasie. Z drugiej strony możesz mieć dane, które są odświeżane co miesiąc, więc nie ma sensu robić codziennych kopii zapasowych, ponieważ będą one zawierały dokładnie te same informacje do następnego miesiąca. Kiedy definiujesz politykę tworzenia kopii zapasowych, musisz ustalić więcej parametrów, niż tylko częstotliwość tworzenia kopii zapasowych. Nie można pozwolić, aby nieskończona liczba kopii bazy danych czekała tylko na wypadek, gdyby w przyszłości konieczne

było przywrócenie. Parametr określający czas życia każdej kopii zapasowej jest zwykle nazywany przechowywaniem danych. Zwykle miesza się różne retencje danych z różną częstotliwością. Z drugiej strony większość narzędzi do tworzenia kopii zapasowych umożliwia tworzenie pełnych i przyrostowych kopii zapasowych, a więc całej kopii obiektów, które chcesz uwzględnić w kopii zapasowej, lub tylko różnicy od poprzedniej kopii zapasowej. Ważne jest również określenie, które urządzenie będzie kopią zapasową i lokalizacja. Na rysunku możesz zobaczyć przykład zasad tworzenia kopii zapasowych dla poszczególnych środowisk z różnym przechowywaniem danych.

Environment	Periodicity	Type	Data Retention	Device	Datacenter
Production	Daily	Incremental	2 weeks	Tape + Disk	Same
Production	Weekly	Full	2 months	Tape + Disk	Disaster Recovery
Production	Monthly	Full	2 years	Tape	Same
Testing	Weekly	Full	2 months	Tape + Disk	Same
Testing	Monthly	Full	2 years	Tape	Same
Development	Weekly	Code Only	2 months	Disk	Same

Proces historyzacji

Z czasem Twoja baza danych będzie się powiększać. To jest coś, co musisz wziąć pod uwagę. Pomimo faktu, że nie stworzyłeś nowych rozwiązań z nowymi wymaganiami dotyczącymi danych, co uważamy za prawie niemożliwe, wszystkie (przynajmniej większość) twoje tabele faktów będą oparte na czasie, więc każdego dnia, każdego miesiąca, każdego roku będzie miał nowe dane do dodania do systemu, zwiększając rozmiar tabel i bazy danych. Również w tabelach przeglądowych będziesz mieć nowe produkty, nowych klientów, nowe lata; w przeciwnym razie Twoja firma wkrótce umrze i jest całkiem prawdopodobne, że nastąpi również wzrost wegetatywny dla Twoich tabel przeglądowych i wymiarów. Ten wzrost wpłynie bezpośrednio na wykorzystywaną przestrzeń, co można po prostu pokryć kupując więcej dysku i dodając go do serwerów, ale może to również wpłynąć na spadek wydajności, ponieważ nie wszystkie zapytania, które działały doskonale, gdy miałeś 2 lata historia w Twojej hurtowni danych będzie nadal działać dobrze w obliczu danych z 5 lat. Z drugiej strony nie musisz mieć całej możliwej historii dostępnej w codziennie używanych tabelach, aby móc wdrożyć proces czyszczenia, który przenosi, kompresuje lub bezpośrednio usuwa stare dane. Ten proces jest tym, który znamy jako proces historyzacji.

Uwaga: W procesie historyzacji bardzo przydatne jest stosowanie strategii partycjonowanych tabel w bazie danych; ułatwi zadanie usuwania danych, kompresji lub przenoszenia.

Możesz mieć wiele strategii dla tego procesu historyzacji, w zależności od celu, który próbujesz osiągnąć; zasadniczo będziesz mieć dwa: czyszczenie przestrzeni i/lub poprawę wydajności. Z drugiej strony istnieją inne kryteria, które należy spełnić. Możliwe, że ze względu na użytkowników biznesowych musisz mieć dostęp do minimalnej ilości danych, mimo że nie jesteś zainteresowany codziennym odpytywaniem, ponieważ istnieją pewne ograniczenia prawne dotyczące udostępniania danych online przez minimum okres czasu lub zestaw lat, lub mogą poprosić o przechowywanie niektórych zagregowanych danych w jakiejś tabeli przez 10 lat, podczas gdy w najbardziej szczegółowej tabeli wymagane jest przechowywanie ich tylko przez 2 lata. Na tej liście pokazujemy tylko niektóre strategie, ale na końcu musisz dostosować proces historii do pojemności serwera i wymagań biznesowych. Niektóre z nich można łączyć:

Usuwanie danych: Jest to najprostszy sposób na zaoszczędzenie miejsca i kontrolowanie ilości informacji. Uzgadniasz ze swoimi użytkownikami, jaka jest wymagana ilość danych i co roku usuwasz dane starsze niż ten limit po poprzedniej kopii zapasowej.

Przenoszenie do innych tabel: możesz przenieść swoje dane do innego zestawu tabel, a następnie opracować pewne dostosowania, aby uzyskać do nich dostęp, gdy chcesz uzyskać do nich dostęp z interfejsu BI, na przykład zdefiniować niektóre widoki, które łączą stare informacje, do których dostęp ma tylko użytkownik aby uzyskać dostęp do tych starych informacji.

Przenoszenie do innej bazy danych: jeśli masz sposób łączenia informacji z wielu baz danych, czy to za pomocą silnika bazy danych, czy narzędzia BI, pozwala to krzyżować i wyszukiwać informacje tak, jakby to było jedno źródło danych, możesz pomyśleć o przeniesieniu tego informacje z głównej bazy danych.

Tańsza pamięć masowa: W połączeniu z poprzednimi opcjami możesz zapisywać stare dane w tańszej pamięci masowej o niższej wydajności, wzmacniając najnowsze dane, zapisując je w najbardziej wydajnej pamięci masowej.

Kompresja danych: jeśli twój silnik bazy danych na to pozwala, możesz skonfigurować kompresję starych partycji, aby zaoszczędzić miejsce, ale być dostępnym do wysyłania do nich zapytań.

Tylko do odczytu: Możliwe, że chcesz ograniczyć nie tylko informacje dostępne dla zapytań, ale także ograniczyć ilość danych aktualizowanych przez proces ETL. Możesz więc zdefiniować partycje lub tabele zawierające stare informacje jako tylko do odczytu.

Oparte na tabelach: możesz pomyśleć o strategii, która usuwa stare dane z dużych szczegółowych tabel i przechowuje szerszą historię w tabelach zagregowanych. Również w niektórych przypadkach, takich jak analiza migawek, których nie można agregować w czasie, takich jak dystrybucja zapasów lub produktów, możesz mieć różną szczegółowość w zależności od wieku migawki. Możesz mieć codzienną migawkę zapasów z ostatnich dwóch miesięcy i tylko miesięczną migawkę z ostatniego dnia miesiąca z ostatnich dwóch lat.

Uwaga: z zakresu narzędzi, które widzieliśmy w tej książce, komponent bazy danych nie jest jedynym, na który może mieć wpływ ten proces historyzacji. Możesz zastosować go również do baz danych MOLAP, tworząc wiele kostek lub usuwając elementy scenariusza lub czasu z konspektu.

Bezpieczeństwo

Przejście do środowiska produktywnego zwykle wiąże się z przyznaniem dostępu większej liczbie użytkowników. A zapewnienie dostępu większej liczbie użytkowników zwykle wiąże się z koniecznością posiadania różnych profili bezpieczeństwa pochodzących z różnych profili użytkowników. Definiując zabezpieczenia na platformie dla użytkownika zazwyczaj trzeba odpowiedzieć sobie na trzy pytania:

1. Co musi zrobić Twój użytkownik?

2. Jakie obiekty musi widzieć Twój użytkownik?

3. Jakie informacje potrzebuje Twój użytkownik do przeanalizowania?

Z tych trzech pytań widać, że możemy wyprowadzić trzy rodzaje zabezpieczeń. Aby odpowiedzieć na pierwsze pytanie, zdefiniujesz zestaw funkcjonalności, który zaspokoi potrzeby użytkownika i który musi być dostosowany do wiedzy użytkownika. Za pomocą drugiego pytania będziesz mógł ustawić dostęp według obiektu, do których tabel i widoków musi mieć dostęp w bazie danych, które raporty, dokumenty, metryki, filtry lub atrybuty musi zobaczyć w narzędziu BI itp. Za pomocą trzeciego pytania będziesz mógł zastosować filtry, aby ograniczyć na poziomie bazy danych lub w narzędziu BI (w zależności od opcji, które ci oferuje), do których danych ma dostęp. Ale przeanalizujmy to na kilku przykładach. Jeśli chodzi o funkcjonalności, możemy mieć ograniczenia typu użytkownika ze względu na posiadaną licencję, która wymaga ograniczenia dostępu do niektórych funkcjonalności, lub możemy

pomyśleć o scenariuszu, w którym mamy jakieś ograniczenie; wyobraź sobie, że ze względu na politykę bezpieczeństwa nie możemy zapisywać danych w folderach współdzielonych, a nasze narzędzie domyślnie nam na to pozwala, więc musimy ograniczyć tę funkcjonalność do użytkowników, aby zachować standardy bezpieczeństwa. Można sobie wyobrazić również sytuację opartą na profilach użytkowników; pomyśl o użytkowniku biznesowym bez wiedzy technicznej, który w ogóle nie wie, czym jest baza danych ani co oznacza akronim SQL, ale Twoje narzędzie pozwala tworzyć raporty z wykorzystaniem wolnego SQL lub modyfikować wygenerowany SQL, aby dostosować go do Twoich potrzeb. Nie miałoby sensu pozwolić temu profilowi użytkowników na modyfikację tego, ponieważ w najlepszym scenariuszu nie będzie on z tego korzystał, a w najgorszym skończymy z dramatycznymi konsekwencjami. Łatwiej jest zrozumieć dostęp według obiektu. Jeśli masz działy finansowe i sprzedażowe i chcesz zastosować ograniczenia działowe, zezwolisz użytkownikom ds. sprzedaży na przeglądanie tylko raportów sprzedaży, a użytkownikom finansowym na przeglądanie raportów finansowych. Zwykle dostęp przez obiekty ma trzy poziomy dostępu: tylko do odczytu, który umożliwia użytkownikowi przeglądanie raportu lub przeglądanie danych w bazie danych; uprawnienia modyfikujące umożliwiające użytkownikowi zmianę danych w tabeli lub modyfikację definicji raportu; i pełną kontrolę, która pozwala użytkownikowi usuwać, tworzyć, modyfikować lub wykonywać dowolne działania na obiekcie. Ta nomenklatura nie jest wspólna dla wszystkich narzędzi, ale zwykle koncepcja bezpieczeństwa jest podobna. Wreszcie, jeśli mówimy o filtrowaniu danych, do których każdy użytkownik ma dostęp, możemy wymyślić wiele narzędzi do tego celu, ale jednym z najbardziej oczywistych dla nas jest przypadek użycia polegający na opracowaniu raportu dla działu sprzedaży firmy. Zespół działu sprzedaży na podstawowym poziomie powinien widzieć tylko sprzedaż przypisanych mu klientów, ale przełożeni tego zespołu mogą przeglądać informacje dotyczące nich i zespołu zależnego, a kierownik działu sprzedaży może przeglądać ogólną sprzedaż firmy. Aby zastosować bezpieczeństwo danych, możesz je wdrożyć głównie na trzy sposoby:

* Filtrowanie danych w bazie danych: Postępując zgodnie z poprzednim przykładem, będziesz mieć swoje tabele faktów, a następnie tabelę relacji, która zawiera relacje między użytkownikiem a przypisanymi przez niego klientami. Wtedy będziesz miał widok powiązany z każdą tabelą, która dołączy do tabeli faktów i tabeli relacji, a warunek filtrujący pole ID_USER (użyjmy tej nazwy pola) z tabeli relacji musi być równy użytkownikowi połączenia. Z drugiej strony będziesz wymagać odmowy bezpośredniego dostępu do tabel, zezwalając tylko na dostęp do widoków. W ten sposób użytkownicy łączący się z bazą danych zobaczą tylko dane swoich klientów.

* Filtrowanie danych w raportach BI: Możesz zdefiniować ten sam filtr przez użytkowników na poziomie raportu, aby był on stosowany do SQL, gdy raport jest wykonywany. Aby środowisko było naprawdę bezpieczne, nie można pozwolić użytkownikom na edycję raportu, aby mogli usunąć filtr lub tworzyć raporty od podstaw.

* Stosowanie filtra bezpieczeństwa do użytkowników: zastosujesz filtr według działu sprzedaży lub dowolnego innego atrybutu do użytkownika, więc każdy wykonywany raport będzie automatycznie zawierał ten filtr w SQL. Aby skorzystać z tej opcji, która jest łatwiejsza w zarządzaniu, wymagane jest, aby Twoje narzędzie umożliwiało korzystanie z filtrów bezpieczeństwa.

Zarządzanie bezpieczeństwem, korzystanie z ról i grup

W celu ułatwienia utrzymania bezpieczeństwa możesz korzystać z ról i/lub grup; nomenklatura i dokładne znaczenie będą zależeć od narzędzia, aby jednocześnie przyznać uprawnienia podobnym użytkownikom. Poprawmy to zdanie. Zdecydowanie zalecamy korzystanie z ról i grup w celu nadawania użytkownikom dowolnych uprawnień. Bezpośrednie dotacje dla użytkowników powinny być czymś rzadkim, co ma sens tylko w rzadkich przypadkach użytkowników. Więc nasza rekomendacja w tym

przypadku jest dość prosta i jasna… do nadawania uprawnień należy zawsze używać grup użytkowników i ról. Ale możesz pomyśleć, dlaczego muszę używać grupy użytkowników, jeśli w tej grupie jest tylko jeden użytkownik? Możesz myśleć o grupie zabezpieczeń o nazwie General Management i masz tylko jednego dyrektora generalnego, który potrzebuje dostępu do wszystkich danych w firmie. Nasza odpowiedź jest łatwa. Pierwszym powodem jest to, że nigdy nie wiadomo, czy ten warunek jeden do jednego będzie zawsze prawdziwy; być może zdecydujesz się wykorzystać tę grupę do włączenia całego komitetu sterującego. Z drugiej strony Twój dyrektor generalny może się zmienić, więc będziesz musiał przenieść wszystkie zabezpieczenia przyznane poprzedniemu dyrektorowi generalnemu na nowego. Prowadzi nas to więc do drugiego powodu; utrzymanie bezpieczeństwa jest znacznie łatwiejsze przy użyciu grup, mimo że sytuacja początkowa mogłaby być łatwiejsza do zarządzania bezpośrednio z użytkownikami, gdy masz kilku użytkowników do przypisania.

Definicja roli użytkownika

W naszym projekcie będziemy mieć różne funkcje, niektóre z nich związane z rolami, które widzieliśmy w Części 2, związane z rolami Agile, niektóre z nich poza rozwojem projektu, jako kluczowi użytkownicy i użytkownicy końcowi. Będziemy mogli zdefiniować różne role związane z funkcjami, które będzie pełnić każdy użytkownik, a następnie odpowiednio zastosować zabezpieczenia, przyznając funkcjonalności, dostępy i bezpieczeństwo danych w zależności od roli. W tej definicji bezpieczeństwa, zwłaszcza w odniesieniu do funkcjonalności i dostępu do obiektów, będziemy musieli wziąć pod uwagę środowisko; dostęp będzie inny w środowisku programistycznym niż w środowisku testowym lub produkcyjnym. Przyjrzyjmy się najczęstszym rolom w środowisku BI, abyśmy mogli ustawić zabezpieczenia w oparciu o ich funkcje.

Kluczowy użytkownik: będzie odpowiedzialny za organizację wszystkich żądań związanych z projektem BI. Ten użytkownik może być tym samym właścicielem produktu, który widziany jest w Scrumie. Zwykle mamy różnych kluczowych użytkowników według datamart, o ile charakter informacji jest inny dla każdego z nich. Będziesz miał kluczowego użytkownika dla Sprzedaży, innego kluczowego użytkownika dla Finansów, innego kluczowego użytkownika dla Operacji itp. Możliwe, że jest on również odpowiedzialny za tworzenie raportów i publikowanie ich dla pozostałych użytkowników, organizowanie folderów i informacji wewnątrz narzędzia , definiując zabezpieczenia, które muszą być zastosowane dla pozostałych użytkowników, definiując role i uprawnienia itp. Będzie miał pełną kontrolę nad środowiskiem.

Użytkownik końcowy : jest to najszersza grupa użytkowników, a w tej grupie może znajdować się wiele typów użytkowników, od analityków korzystających z raportów ad hoc, tworzenia danych ad hoc i możliwości nawigacji w celu badania różnych danych, po końcowych odbiorców informacji, otrzymywać statyczne raporty w formacie pdf.

Deweloper : Ten użytkownik będzie odpowiedzialny za wdrożenie wymaganej struktury do analizy, więc będzie wymagał dostępu w środowisku programistycznym do wszystkich poziomów platformy BI, ETL, bazy danych i narzędzi BI, aby móc projektować żądane obiekty w wszystkie warstwy.

Administrator: Ten użytkownik będzie odpowiedzialny za przenoszenie obiektów między środowiskami, monitorowanie środowisk, wdrażanie zabezpieczeń użytkowników, instalowanie i konfigurowanie platformy oraz ogólnie za wszystkie zadania związane z zarządzaniem technicznym wymagane do prawidłowego działania platformy.

Jeśli twoje środowisko się rozrośnie, mogą pojawić się nowe role, takie jak menedżerowie usług, wsparcie aplikacji, właściciele platform itp. Istnieje kilka interesujących metodologii, takich jak standardy ITIL, które definiują zestaw zasad, których należy przestrzegać w przypadku utrzymania

infrastruktury IT. Ale ten temat jest kwestią wielu dokumentów, kursów i certyfikatów, które wykraczają poza nasz zakres.

Matryca bezpieczeństwa

Gdy już wiesz, jakie role będą zaangażowane w Twój projekt, warto określić, jakie uprawnienia będą miały w zależności od środowiska, dlatego zalecamy wdrożenie macierzy takiej jak ta pokazana na rysunku , z szczegółowością i szczegółowością, jakiej może wymagać Twój projekt.

		Key User			End User			Developer			Administrator		
Type	Permission	Dev	Test	Prod	Dev	Test	Prod	Dev	Test	Prod	Dev	Test	Prod
Database	Access		Read	Read				Modif	Read		Full	Full	Full
Database	Configuration										Yes	Yes	Yes
Database	Tables		Select	Select				Create	Update		Create	Create	Create
Database	Views		Select	Select				Create	Update		Create	Create	Create
Database	Procedures		Execute	Execute				Create	Execute		Create	Create	Create
ETL	Access		Read	Read				Modif	Read		Full	Full	Full
ETL	Editor							Create	Read		Create	Create	Create
ETL	Monitor		Read	Read				Execute	Execute		Execute	Execute	Execute
BI	Access		Read	Modif				Modif	Read		Full	Full	Full
BI	Configuration										Yes	Yes	Yes
BI	Design tools		Read	Read				Create	Modif		Create	Create	Create
BI	Report creation path		Public	Public			Personell	Public	Public		Public	Public	Public
BI	Drilling		Yes	Yes			Yes	Yes	Yes		Yes	Yes	Yes
BI	Report Distribution		Yes	Yes			Yes	Yes	Yes		Yes	Yes	Yes

Audyt

Chcielibyśmy skomentować ostatni temat związany ze środowiskiem produkcyjnym: istnieje potrzeba zapewnienia, że wszystkie zasady, które zdefiniowaliśmy w poprzednich sekcjach, są spełnione. Aby to zapewnić, zalecamy zdefiniowanie i wykonanie zestawu procesów audytu, który zapewnia poprawność wszystkich powiązanych wytycznych platformy w zakresie konwencji nazewnictwa, lokalizacji, kontroli wydajności, definicji bezpieczeństwa oraz zdefiniowanych metodologii rozwoju, transportu lub jakikolwiek inny uruchomiony proces. Uważamy, że istnieją dwie główne grupy audytów: te, które mają na celu upewnienie się, że stosujemy odpowiednie zabezpieczenia w naszym systemie, oraz te, które mają na celu upewnienie się, że prawidłowo postępujemy zgodnie z wytycznymi lub najlepszymi praktykami zdefiniowanymi w naszym systemie. Nie będziemy robić rozszerzonej analizy procesów audytowych, ale chcielibyśmy zauważyć, że ważne jest przestrzeganie pewnych procesów, które zapewniają przestrzeganie wszystkich reguł, które zdefiniowaliśmy (lub o których przestrzeganie ktoś nas prosi). Zalecamy również jak największą automatyzację tego procesu audytu i wynikających z niego działań.

Audyty bezpieczeństwa

Jak możesz sobie wyobrazić, ta grupa audytów jest najbardziej odpowiednia do wdrożenia, aby uniknąć wycieków danych z Twojej firmy. Nie wyczerpując tematu, przeanalizujmy kilka porad dotyczących bezpieczeństwa, które powinieneś spróbować zweryfikować w swoim środowisku:

* Przede wszystkim powinieneś mieć uruchomiony proces audytu w swojej bazie danych, zapisując przynajmniej wszystkie połączenia w systemie, abyś mógł sprawdzić, kto w danym momencie był podłączony do systemu. Weryfikacja tego dziennika powinna być pierwszym audytem, który należy wdrożyć, aby móc śledzić wszelkie wymagane informacje.

* Korzystanie z użytkowników ogólnych może być konieczne do wdrożenia procesów, takich jak ETL, połączenia między narzędziem BI a hurtownią danych w celu połączenia systemu MOLAP z bazą danych, ale zdecydowanie zalecamy ograniczenie użytkowników ogólnych do procesów technicznych; każde ręczne połączenie z systemem powinno odbywać się z osobistym użytkownikiem i być monitorowane w systemie logów.

* Powinieneś zweryfikować, czy gdy użytkownik opuszcza firmę lub nie potrzebuje dostępu do systemu BI, jego użytkownik jest usuwany lub przynajmniej wyłączany z systemu. Najlepszym podejściem do tego jest użycie systemu uwierzytelniania już wdrożonego w firmie lub przynajmniej zsynchronizowanego z nim, takiego jak Active Directory, LDAP, Novell lub dowolnej innej metody uwierzytelniania używanej w firmie. W ten sposób, gdy użytkownik zostanie wyłączony w celu uzyskania dostępu do ogólnych systemów, zostanie on również wyłączony w celu uzyskania dostępu do twojego systemu. Ale w każdym razie powinieneś potwierdzić, że to prawda.

* Tworzenie modeli baz danych, procesów ETL, raportów i pulpitów nawigacyjnych nie wymaga robienia tego na rzeczywistych danych. Ogólna zasada jest taka, że programiści nie mogą uzyskać dostępu do danych produkcyjnych, o ile nie są użytkownikami biznesowymi, dlatego należy unikać udzielania dostępu każdemu, kto go nie potrzebuje. Z naszego doświadczenia wiemy, że czasami nie da się przeanalizować incydentu lub niezgodności danych bez dostępu do nich, dlatego zwykle definiuje się rolę kierownika projektu lub jakiegoś zespołu wsparcia aplikacji, który ma taki dostęp; ale do początkowego rozwoju zwykle nie potrzebujesz prawdziwych danych. Z drugiej strony, w połączeniu z poprzednią radą, jeśli Twój zespół programistów jest dostarczany z zewnątrz przez firmę konsultingową, możliwe, że Twój proces kontroli nad osobami zewnętrznymi jest inny niż ten nad zespołami wewnętrznymi, więc musisz co jakiś czas walidować do czasu, gdy użytkownicy zewnętrzni nadal potrzebują dostępu.

* Czasami foldery współdzielone są wymagane do udostępniania informacji różnym grupom użytkowników, lokalizowania plików płaskich w celu załadowania ich do bazy danych lub wyodrębniania z nich informacji. Powinieneś również sprawdzić, kto uzyskuje dostęp do tych folderów, unikając przyznawania wszystkim/grupie publicznej.

* Jeśli potrzebujesz zdefiniować model bezpieczeństwa danych w oparciu o widoki bazy danych lub narzędzie BI, upewnij się, że nikt nie może przeskoczyć ograniczeń bezpieczeństwa w dostępie do całego projektu, zapewnij solidność swojego rozwoju bezpieczeństwa i sprawdzaj od czasu do czasu, czy nic się nie zmieniło został przypadkowo zmieniony.

Najlepsze praktyki audytu

Zwykle nie są one tak ważne z punktu widzenia bezpieczeństwa, ale zdefiniowanie pewnych zasad i upewnienie się, że są one przestrzegane, może zaoszczędzić wiele czasu i bólu głowy. Potwierdź, że Twoje zasady są spełnione i mogą ułatwić Ci zarządzanie rzeczami. Skomentujmy też kilka przykładów dobrych praktyk, które powinny zostać poddane audytowi.

* Od czasu do czasu możesz zostać poproszony o zdefiniowanie połączenia między środowiskami, opracowanie narzędzia BI uzyskującego dostęp do produkcyjnej bazy danych z powodu niedostępności rozwojowego środowiska bazy danych z powodu aktualizacji platformy, Twoje środowisko programistyczne ETL połączone z produkcyjną bazą danych w celu sprawdzenia poprawności wydajność obciążenia z rzeczywistą ilością danych lub inne wymaganie, którym można zarządzać za pomocą tego rodzaju obejścia. Krzyżowy dostęp między środowiskami może być przydatny w pewnym momencie do sprawdzenia lub tymczasowego połączenia, ale należy to zapewnić

Twoje połączenia zostaną przywrócone po zakończeniu tego tymczasowego okresu. W poprzedniej sekcji skomentowaliśmy, że programiści uzyskują dostęp do prawdziwych danych, a krzyżowanie się środowisk może również powodować problemy z bezpieczeństwem.

* W Części 5 mówiliśmy o konwencjach nazewnictwa służących do definiowania obiektów bazy danych. Przestrzeganie konwencji nazewnictwa wydaje się być czymś arbitralnym, co moglibyśmy pominąć, ale

jeśli istnieje reguła, należy jej przestrzegać. Wyobraźmy sobie, że ustalamy, że na podstawie wykorzystanej przestrzeni w bazie danych dzielimy koszt platformy na różne działy. Konwencja nazewnictwa tabel polegała na tym, że trzecia część nazwy tabeli określała obszar, SAL dla sprzedaży, FIN dla finansów lub OP dla operacji. Jeśli środowisko nie jest zdefiniowane w tabeli, nie wiemy, kto obciąży tym kosztem, a nasz dział IT będzie wymagał jego przyjęcia. Albo definiujemy maksymalną nazwę bazy danych na 10 znaków i mamy skrypt kopii zapasowej, który ma ograniczenie oparte na tej długości, aby zdefiniować nazwę pliku kopii zapasowej. Tworzymy bazę danych składającą się z 20 znaków, a następnie nazwa kopii zapasowej jest niepoprawna i kopie zapasowe są nadpisywane, pozostawiając tylko jedną dostępną zamiast wymaganej historii.

* Wyobraź sobie, że mamy ograniczone okno ładowania do 4 godzin w nocy, aby uruchomić wszystkie procesy ETL, aby dostarczyć informacje na czas. Będziemy zainteresowani audytem wszystkich czasów ładowania procesów w celu wykrycia, które należy poprawić.

* Zdefiniowanie pewnych zasad lokalizowania plików może ułatwić zadania związane z tworzeniem kopii zapasowych, transportem obiektów między środowiskami, utrzymywaniem czystych środowisk lub aktualizacją wersji oprogramowania. Powinieneś sprawdzić, czy pliki są poprawnie zlokalizowane, aby nie przegapić żadnego obiektu do skopiowania w różnych środowiskach lub aby nie przeciążyć systemu plików lub dysku, ponieważ duże pliki dziennika są zapisywane w niechcianej lokalizacji.

To tylko niektóre przykłady dotyczące procesu audytu, ale chcielibyśmy zwrócić uwagę na potrzebę audytu, aby wszystko zostało wykonane poprawnie. Czasami łatwo jest przestrzegać pewnych zasad podczas wdrażania, ale z czasem, wraz z modyfikacjami i zmianami w zespołach, trudniej jest utrzymać uwagę na czystym środowisku.

Wniosek

Tutaj widzieliśmy, jak zastosować pewne zasady, które pozwalają nam zdefiniować zestaw środowisk do pracy dla różnych ról, które będziemy pełnić w naszej firmie. Będziemy mieć środowisko niezawodne dla użytkowników końcowych, ponieważ będą oni mieli środowisko z modyfikacjami pod kontrolą, które dostarcza informacji potrzebnych do ich codziennej pracy. Będziemy mieć również środowisko dostępne dla programistów do wykonywania ich pracy bez wpływu na informacje o użytkowniku końcowym. Będziemy mieli środowisko do testowania i walidacji wykonanych modyfikacji. Zadbamy o to, aby środowisko było wystarczająco bezpieczne, zapewniając wymagane zabezpieczenia w każdym środowisku. Dodamy warstwę kontrolną w transportach, aby sprawdzić, czy są one wykonywane bez wpływu, i będziemy mieć politykę konserwacji, aby kontrolować wszystkie nasze środowiska poprzez audyt, czy przestrzegane są nasze zasady. W następnym rozdziale, ostatnim, rzucimy okiem na pokrewny temat, aby zrozumieć, jak wdrożyć środowisko chmurowe, bez konieczności początkowej inwestycji w sprzęt i samego wynajmu pojemności chmury. Jest to powiązane, ponieważ pokaże również łatwy sposób definiowania tych wielu środowisk, które ocenialiśmy w tej części.

12. Przeniesienie procesów BI do chmury

Obecnie panuje tendencja do ograniczania inwestycji w maszyny, szczególnie w przypadku małych firm. Dzięki temu kilka czołowych firm dostrzegło okazję biznesową. Wśród nich Amazon, Google i Microsoft. Firmy te zdały sobie sprawę, że większość firm nie może/nie chce sobie pozwolić na drogie zakupy sprzętu, który może stać się przestarzały już po kilku latach. Zamienili się w dostawców usług w chmurze, co w zasadzie oznacza, że można wypożyczyć część maszyny lub całą maszynę i uruchamiać na niej, co się chce, bez konieczności kupowania drogiego sprzętu. Obecnie sprawy poszły dalej i nie tylko można od nich wynajmować infrastrukturę, ale także niektóre inne usługi. Na przykład wszystkie trzy firmy sprzedają niektóre ze swoich baz danych jako usługę, więc zamiast zatrudniać administratora baz danych do dostrajania bazy danych, administrator może wykonać proaktywną pracę, aby zapewnić jej płynne działanie. W tej części zobaczymy, jak przesłać nasze procesy BI, w zasadzie bazę danych i ETL do chmury. Wygląda łatwo, prawda? No cóż, tak jest. Nie możemy jednak bezpośrednio rozpocząć tworzenia naszych zasobów. Istnieje kilka konfiguracji, o których musimy wiedzieć, zanim zaczniemy tworzyć nasze serwery wirtualne i to jest pierwsza rzecz, którą musimy zobaczyć. Zanim jednak to nastąpi, omówimy trochę nasze rozwiązanie w chmurze i przedstawimy możliwe opcje.

Wybór naszego dostawcy usług w chmurze

Jak widzieliśmy, możemy wybierać spośród szerokiej gamy dostawców usług w chmurze . W Internecie znajdziesz wiele. Każdy z nich będzie działał; po prostu wybierz ten, który bardziej Ci odpowiada lub który jest łatwiejszy w użyciu. Oczywiście cena jest czymś ważnym przy podejmowaniu decyzji, ale nie widzieliśmy bardzo dużych różnic między wszystkimi analizowanymi dostawcami. Rozliczenia to kolejny aspekt do rozważenia: niektóre z nich wystawiają faktury za zużycie, zwykle według godzin; a niektóre z nich fakturują stałą miesięczną stawkę, więc warto się nad tym zastanowić. Oceniliśmy dwa rozwiązania, które są najczęściej używane: Amazon AWS i Microsoft Azure.

Rozważania wstępne

Zanim przystąpimy do tworzenia i uruchamiania naszych instancji w chmurze, powinniśmy przemyśleć kilka aspektów, z których najważniejsze to:

* Rozmiar platformy. Liczba serwerów, których potrzebujemy, odpowiednio dobierając ich wymiary, aby obsłużyć obciążenie za pomocą rozwiązania, które zapewnia nam wydajność i skalowalność.

* Zdecyduj, które usługi trafią do chmury, a które pozostaną lokalne.

* Bezpieczeństwo, kontrola dostępu do serwerów oraz integracja z własnym firmowym intranetem, zwłaszcza jeśli część serwerów trzymamy u siebie.

* Lokalizacja danych.

* Śledzenie i rejestrowanie wszystkich usług działających w chmurze przy użyciu kontroli stanu.

* SLA i możliwość (lub nie) wynajęcia jakiegoś planu wsparcia oferowanego przez dostawcę chmury.

* Odzyskiwanie po awarii (Pominęlibyśmy to, ponieważ jest to niewielka zmiana, ale w przypadku poważnej należy to dokładnie zaplanować).

* Oblicz koszt eksploatacji platformy i oblicz oszczędności (jeśli takie istnieją).

Przyjrzymy się trochę tym aspektom, zanim przejdziemy do rzeczywistych szczegółów implementacji

Rozmiar platformy: wydajność i skalowalność

Przeniesienie rozwiązań i infrastruktury do chmury nie jest łatwe. Pierwszą rzeczą, która może przyjść ci do głowy, są wymagania techniczne platformy. Zacząłeś już zastanawiać się we własnym umyśle nad warunkami pojemności i wykorzystania oraz zacząłeś myśleć o wymaganych zasobach. Cóż, jest więcej rzeczy dla aktora, ale wyraźnie jest to jedna. Jedną z zalet rozwiązania w chmurze jest to, że nie trzeba iść na całość ani dokonywać żadnych ogromnych inwestycji na początku. Większość platform chmurowych oferuje rozwiązanie typu pay per use. Jest to prawdopodobnie oczywisty wybór, gdy po raz pierwszy rozpoczynasz inwestycję w rozwiązanie BI. Jeśli projekt zakończy się sukcesem i wniesie wartość dodaną do Twojej firmy, zawsze możesz rozszerzyć swoją analizę i pójść dalej. Ale jeśli z jakiegokolwiek powodu projekt nie przyniesie oczekiwanych rezultatów, nie tracisz całej tej inwestycji w sprzęt, a jedynie wysiłek włożony w rozwój. Jest to ważny aspekt do rozważenia, ponieważ wejście na inwestycję może stanowić istotną barierę w tak małych firmach. Rozmiar platformy może być skomplikowany, ale w tym momencie lepiej byłoby wybrać rozwiązanie typu „płać za użytkowanie", bez ustalonej płatności z góry, aby jeszcze bardziej zmniejszyć ryzyko. Wtedy standardowa/mała instancja posłuży do hostowania hurtowni danych i serwera ETL. Możesz zdecydować się na rozdzielenie obu maszyn, co jest dobrym rozwiązaniem, zwłaszcza jeśli będziesz prowadzić intensywne zadania przetwarzania. Opłacalna może być również opcja przeniesienia narzędzia transakcyjnego do chmury. W takim przypadku zalecamy hostowanie go na osobnej maszynie.

Projekt fizyczny

Decydowanie, które serwery przejdą do chmury, a które pozostaną lokalne, jest również częścią tego etapu. Cały projekt fizyczny obejmuje głównie pracę administratorów systemu: decydowanie o zakresach adresów IP, zaporach ogniowych, dostępie bezpieczeństwa, systemach operacyjnych, sieci platformy i wielu innych. Jest to wyraźnie związane z wielkością platformy, ale rodzi również inne pytania. Czy wszystkie nasze środowiska będą znajdować się w chmurze (programowanie, symulacja i produkcja), czy niektóre zachowamy lokalnie? Naszym zdaniem, jeśli firma nie jest zbyt duża i możesz sobie na to pozwolić, wybralibyśmy rozwiązanie w pełni chmurowe. Posiadanie architektury mieszanej wydaje się być jednym z najnowszych trendów w firmach średniej wielkości, które mogą sobie pozwolić na instalację lokalną. Część z nich ma swoje środowiska testowe w chmurze, podczas gdy środowisko produkcyjne jest hostowane na miejscu. Chociaż to rozwiązanie może również Ci odpowiadać, wymaga więcej pracy związanej z utrzymaniem środowisk. Pomyśl o zasadach tworzenia kopii zapasowych, bezpieczeństwie, konfiguracja sieci jest również bardziej złożona, ponieważ wszystkie sieci wewnętrzne muszą być mieszane, aby zasoby w chmurze mogły uzyskiwać dostęp lub „widzieć" zasoby lokalne i na odwrót. Aby uniknąć tych komplikacji, wdrożymy w chmurze trzy środowiska: programistyczne, symulacyjne i produkcyjne; i najlepiej, gdyby wszystkie trzy znajdowały się w różnych sieciach, aby uniknąć komunikacji między nimi, aby uniknąć błędów ludzkich, to znaczy zapisywania danych z testu do produkcji lub podobnych rzeczy.

Bezpieczeństwo

Bezpieczeństwo jest zawsze najważniejsze. Nie tylko dlatego, że możesz mieć wrażliwe dane, do których nie chcesz, aby osoby nieuprawnione miały dostęp, ale także ze względów prawnych. Z każdym rokiem jurysdykcja w zakresie prywatności staje się coraz trudniejsza. Możesz mieć poważne kłopoty, jeśli nie zabezpieczysz swoich danych. Większość dostawców chmury oferuje pewnego rodzaju zabezpieczenia w pakiecie z ich wdrożeniami. Zwykle uprawnia to zaporę ogniową, którą można skonfigurować tak, aby blokowała wybrane porty, certyfikaty umożliwiające dostęp do twoich maszyn podczas zdalnego łączenia oraz role uprawnień do przypisania różnym aktorom wchodzącym w interakcję z platformą, dzięki czemu możesz ograniczyć ich dostęp i ograniczyć uprawnienia do pracy, którą powinni wykonywać. Integracja tej sieci w chmurze z już działającą infrastrukturą może również stanowić wyzwanie. Musisz dodać nowe wpisy na serwerach DNS, dodać reguły do istniejących zapór

ogniowych i zmodyfikować tablice routingu, aby nowe podsieci były kierowane od wewnątrz. Może to również wpłynąć na bezpieczeństwo, nie tylko w chmurze, ale także w Twojej organizacji, dlatego należy dokładnie przeanalizować wszelkie zmiany.

Lokalizacja danych

Ważny jest również wybór miejsca przechowywania danych. Idealnie byłoby, gdybyś przechowywał dane jak najbliżej swoich klientów (w tym aspekcie bierzesz pod uwagę klienta lub swój dział BI). Większość rozwiązań chmurowych oferuje różne strefy geograficzne do wyboru, gdzie przechowywać dane. Niezależnie od tego, czy nie ma ich w twoim kraju, prawdopodobnie znajdziesz region bliżej domu. Jest to ważne, zwłaszcza w aplikacjach czasu rzeczywistego, ale może mieć również wpływ na wdrożenie BI. W obrębie tej samej strefy geograficznej zazwyczaj znajdują się różne lokalizacje. Są one używane do redundancji i odzyskiwania po awarii. Pamiętaj jednak, że niektóre z tych funkcji mają swoją cenę.

Badania zdrowia

Wszystkie rozwiązania w chmurze zapewniają również niektóre pulpity nawigacyjne do sprawdzania wydajności. Przejrzyj wyświetlane tam dane i podejmij odpowiednie działania, jeśli pojawią się nieoczekiwane wyniki. Zwykle będziesz mieć możliwość dodawania alertów i konfigurowania progów w tych pulpitach nawigacyjnych, aby mieć pewność, że będziesz informowany o wszelkich nieoczekiwanych zachowaniach.

SLA i plany wsparcia

Każdy dostawca chmury powinien gwarantować minimalny poziom działania usługi. W informatyce jest to zwykle regulowane umową o gwarantowanym poziomie usług (lub SLA). Jest to dokument, w którym opisane są najczęstsze mierniki oceny, a dostawca zobowiązuje się zagwarantować ich określoną wartość. Twoim obowiązkiem jest upewnienie się, że te umowy zostały wykonane, a jeśli nie, zażądaj odszkodowania. Podobnie jak w przypadku umowy SLA, niektórzy dostawcy chmury dają użytkownikowi możliwość wykupienia dodatkowego wsparcia. Czasami wiąże się to z lepszymi umowami SLA, większą reakcją zespołów technicznych w przypadku problemów lub zaawansowanym wsparciem wykwalifikowanych inżynierów. Chociaż korzystanie z tych dodatkowych pakietów może być kosztowne, należy je wziąć pod uwagę po wprowadzeniu systemu do produkcji. Jeśli użytkownikom lub firmie zależy na dostępności systemu przez 100% czasu, a jego niedostępność powoduje straty dla firmy, należy podjąć odpowiednie działania, aby te problemy się nie pojawiały, a gdy już się pojawią, zostały rozwiązane tak szybko, jak to możliwe. możliwy. Ale prawdopodobnie na początku nie będziesz nimi zainteresowany.

Oblicz bieżący koszt platformy

Obliczenie kosztu platformy jest skomplikowane. Na szczęście większość dostawców chmury pomaga w publikowaniu kalkulatorów internetowych, gdzie przynajmniej można spróbować obliczyć, ile będzie kosztować wdrożenie chmury. Oczywiście te kalkulatory nie uwzględniają wysiłków rozwojowych, ale przynajmniej można przewidzieć pieniądze potrzebne do zainwestowania w infrastrukturę. Próbkę zobaczymy w dalszej części tej części, analizując jednego z dostawców chmury.

Pierwsze spojrzenie na projekt fizyczny

Po kilku rozmowach z naszym administratorem zdecydowaliśmy, że do naszego pierwszego środowiska (środowiska testowego) potrzebujemy:

* Maszyna, na której ma zostać uruchomiony system transakcyjny (w naszym przypadku Odoo, a ten można już wdrożyć lokalnie, więc jest to kwestia decyzji za lub przeciw przeniesieniu go do chmury).

* Maszyna, na której można uruchomić naszą bazę danych hurtowni danych.

* Maszyna, na której można uruchomić silnik ETL. Jeśli korzystamy z Pentaho, musimy gdzieś zainstalować silnik. Również ta maszyna będzie. Hostuj crontab dla zadań, więc to będzie nasz wykonawca zadań.

* Maszyna, na której w razie potrzeby można uruchomić nasz silnik raportowania. Jeśli pracujesz np. z Power BI, jedyną opcją jest skorzystanie z chmury Microsoft. Jeśli pracujemy z Qlikview Desktop, możemy użyć chmury Qlik lub zainstalować serwer w naszym wdrożeniu. Ponieważ mówimy o środowisku testowym, możemy zrezygnować z jakiegokolwiek wdrożenia serwera i ograniczyć opcję tylko do klientów stacjonarnych, łączących się bezpośrednio z hurtownią danych.

Kiedy to będzie jasne, musimy zdecydować o kilku aspektach. To są:

* Czy chcemy uruchamiać wszystkie nasze maszyny w wirtualnej chmurze prywatnej (VPC)?

* Które porty musimy otworzyć?

* Jaki typ instancji wybierzemy dla każdej maszyny?

Odpowiedź na te pytania jest subiektywna i zależy wyłącznie od wybranej platformy. W celach ilustracyjnych zdecydowaliśmy się na:

* Utwórz VPC. Chociaż w większości przypadków może to nie być najłatwiejszy wybór, ponieważ uczyni nasze środowisko nieco bardziej złożonym, jest to wymagane przez typ instancji, którego planujemy użyć. Więcej o tym później. Brak utworzenia VPC oznacza, że każda maszyna będzie musiała mieć powiązany publiczny adres IP, abyśmy mogli połączyć się z zewnątrz. Jeśli wybierzemy opcję VPC, wszystkie maszyny mogą należeć do tej samej podsieci prywatnej, a następnie możesz chcieć połączyć tę podsieć ze swoją siecią wewnętrzną lub umieścić maszynę do wykonywania NAT między twoją siecią a izolowaną siecią prywatną w Chmura.

* Utwórz maszynę z około 2 GB pamięci RAM do obsługi naszego magazynu danych i tylko jednym procesorem. W tym momencie nie potrzebujemy zbyt dużej mocy, ponieważ jest to testowa baza danych.

* Utwórz kolejną maszynę o pojemności 2 GB, na której uruchomisz integrator danych Pentaho i Odoo.

Teraz nadszedł czas, aby zobaczyć, jak skutecznie wdrożyć to przy jednoczesnym utrzymaniu niskiego budżetu. Przyjrzyjmy się trzem dostawcom chmury, ich cenom i aktualnym ofertom oraz zobaczmy, jak możemy wdrożyć platformę.

Wybór odpowiedniego dostawcy chmury

Do wyboru jest wielu dostawców, ale ograniczymy listę do trzech: AWS (Amazon Web Services), Microsoft Azure i chmura Google. Wszystkie trzy oferują pewnego rodzaju bezpłatną usługę, zwykle ograniczoną do liczby godzin lub okresu czasu. Przyjrzyjmy się ich propozycjom, zaczynając od największej, Amazon Web Services.

Usługi sieciowe Amazon (AWS)

Usługi sieciowe Amazon, czyli AWS, to najpopularniejsze rozwiązanie chmurowe, jakie znajdziesz. Mają ponad 500 produktów i/lub usług (i ta liczba wciąż rośnie) oferowanych klientom, pogrupowanych

według kilku kategorii, takich jak bazy danych, analityka, sieć, pamięć masowa, komputer, urządzenia mobilne, programowanie, Internet rzeczy…

Skupimy się na instancjach Compute i Database, a na końcu rozdziału przyjrzymy się ofercie Analytics. Aby rozpocząć uruchamianie instancji, najpierw musimy zarejestrować konto tutaj:

https://aws.amazon.com/?nc1=h_ls

Dobrą wiadomością jest to, że AWS zapewnia bezpłatną warstwę do testowania usługi. Możliwe, że uda nam się dostosować nasze potrzeby do tej darmowej warstwy, ale będzie to w dużej mierze zależało od wymagań naszej platformy. Obecnie bezpłatna warstwa AWS jest wyjaśniona tutaj: https://aws.amazon.com/free, ale dla ułatwienia, ponieważ oferują one całkiem sporo usług, których w tym momencie nie potrzebujemy, podsumujmy oferty, które mogą bądź dla nas interesujący:

* 750 godzin miesięcznie Amazon Compute Instance (EC2). Będzie to używane do uruchamiania jednego komputera, komputera ETL i programu uruchamiającego zadania.

* 5 GB przestrzeni dyskowej S3. Możemy ich używać do tworzenia kopii zapasowych lub plików statycznych.

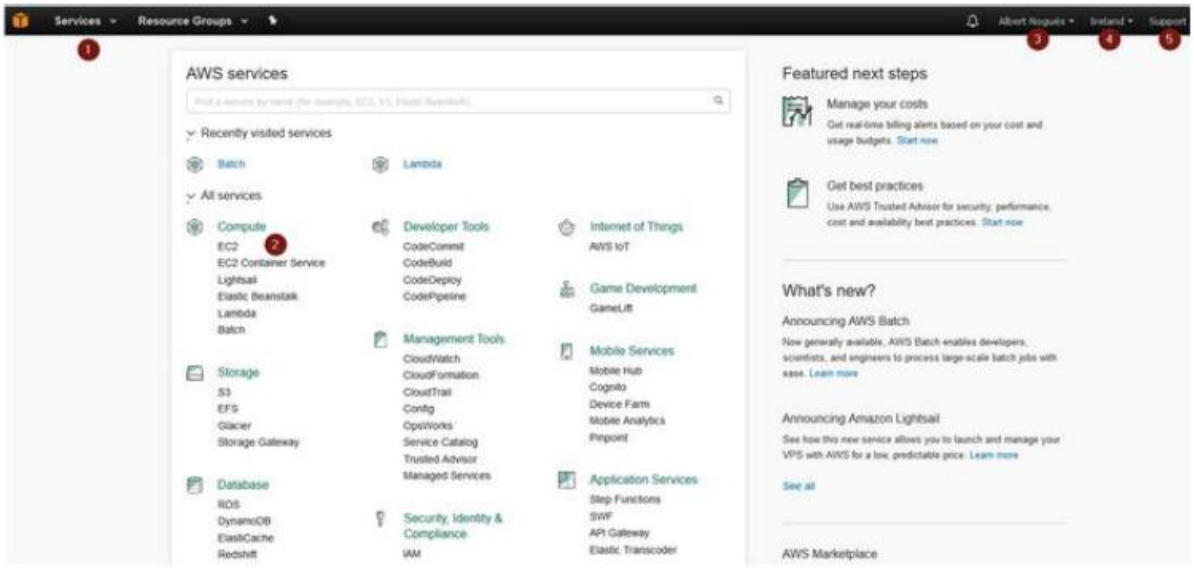
* 750 godzin miesięcznie Amazon RDS do obsługi bazy danych. Będzie to używane w bazie danych Datawarehouse.

* 1 GB SPICE dla AWS Quicksight. Jest to zupełnie nowe narzędzie analityczne opracowane w celu łatwego tworzenia raportów i pulpitów nawigacyjnych na podstawie danych w chmurze AWS.

Niektóre z tych ofert działają wiecznie, na przykład Quicksight, ale najciekawsze, instancja EC2, instancja RDS i pamięć S3, będą działać bezpłatnie tylko przez rok. Więc uważaj jak w pewnym momencie; zostaniesz obciążony. Czas zarejestrować się w AWS i zacząć budować naszą platformę testową. Chodźmy!

Implementacja środowiska deweloperskiego w AWS

Aby zaimplementować nasze rozwiązanie w AWS, zaczniemy od stworzenia instancji EC2 do przechowywania ETL. W tym celu i zakładając, że już zarejestrowałeś się w AWS, musisz przejść do konsoli zarządzania AWS. Jest to centralne miejsce, w którym uruchamiane są wszystkie usługi lub aplikacje. Na początku może to być nieco przytłaczające, ponieważ już powiedzieliśmy, że istnieją setki usług. Ale teraz musimy skupić się tylko na instancjach EC2. Zaloguj się do konsoli AWS wchodząc na adres: https://aws.amazon.com/console/ po zalogowaniu zobaczymy naszą główną konsolę AWS. Jest kilka rzeczy, z którymi musimy się zapoznać i najpierw musimy je wyjaśnić. Rysunek pokazuje je.

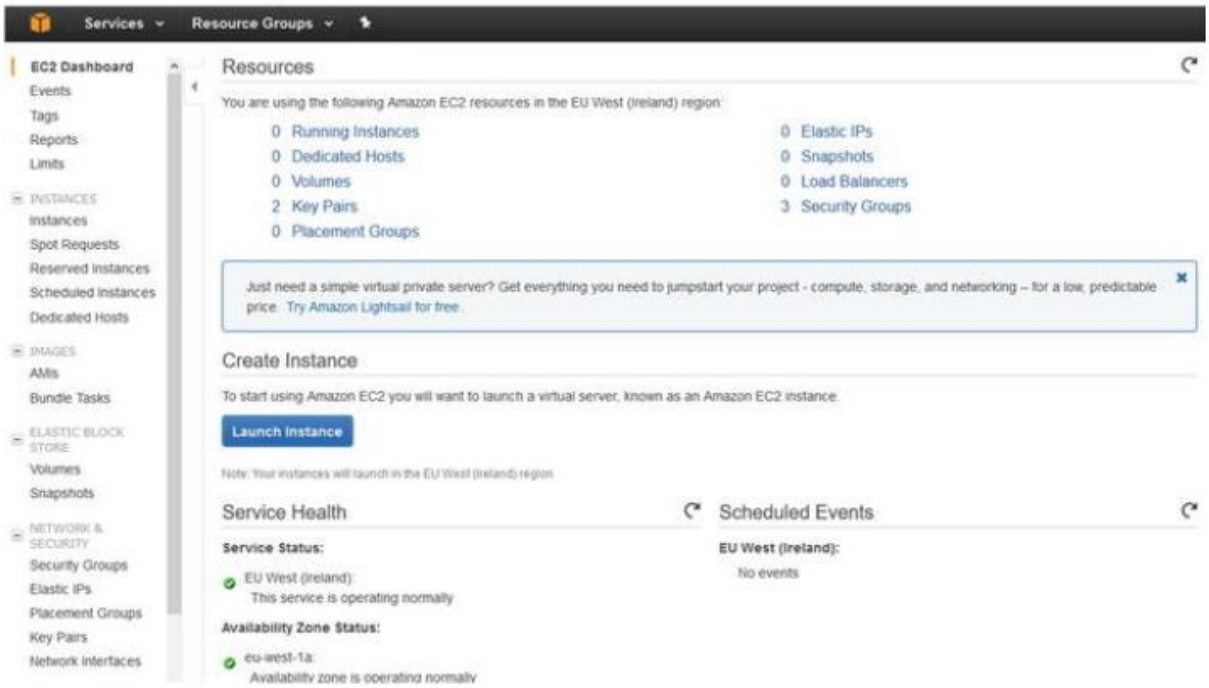


Punkt 1 zaznaczony na rysunku pokazuje menu, w którym można uzyskać dostęp do wszystkich usług. Dostęp do nich z tego menu lub kliknięcie na nazwę usługi w oknie głównym, jak w punkcie 2 pokazanym na obrazku, jest taki sam. Ale znajdujemy łatwy dostęp do nich przez menu, ponieważ wszystko jest uporządkowane i zorganizowane. Nazwa właściciela konta, pokazana w punkcie 3, daje nam dostęp do kilku ważnych sekcji AWS. Wewnątrz nich, oprócz danych konta i niektórych innych konfiguracji, mamy dostęp do pulpitu rozliczeniowego i danych uwierzytelniających. Te dwa aspekty pokazują nam, ile pieniędzy wydaliśmy w bieżącym i przeszłym cyklu rozliczeniowym oraz gdzie te pieniądze zostały wydane. Śledzenie tego jest bardzo ważne, aby koszty były zgodne z oczekiwaniami i aby uniknąć niespodzianek na koniec cyklu rozliczeniowego. Pozwala także skonfigurować alarmy na podstawie progów, aby upewnić się, że otrzymasz powiadomienie w przypadku, gdy wydatki zaczną rosnąć powyżej wcześniej zdefiniowanego progu. Drugim ważnym tematem, na który należy zwrócić uwagę, w tym samym punkcie 3 na obrazku, jest menu poświadczeń bezpieczeństwa. Stąd możemy decydować czy korzystać z uprawnień IAM, czyli Zarządzanie tożsamością i dostępem, aplikacja do przypisywania ról do kont, dzięki czemu można kontrolować jakie uprawnienia są nadawane poszczególnym użytkownikom i jakie jest zalecane rozwiązanie w przypadku konieczności posiadania więcej niż jednej osoby zarządzać środowiskiem, co powinno mieć miejsce w przypadku większości wdrożeń. Możliwe jest jednak trzymanie się jednego konta menedżerskiego, które będzie miało uprawnienia do wszystkiego. W tym miejscu możesz również tworzyć klucze API, na wypadek gdybyś chciał wchodzić w interakcję z API AWS lub uruchamiać usługi z zewnętrznych narzędzi (Skrypty, wywołania API, zewnętrzni dostawcy…). Kolejna ważna sekcja, oznaczona jako czwarty punkt, to wybór regionu, w którym się znajdujemy. Na zdjęciu, które widzisz, używam regionu Irlandii. Możesz poruszać się po wszystkich możliwych regionach, ale upewnij się, że wszystkie usługi znajdują się w tej samej strefie, aby uzyskać lepszą wydajność. Pamiętaj też, że każda strefa może mieć inne ceny usług, więc upewnij się, że je sprawdziłeś. W tym przykładzie wdrożymy wszystko w regionie Irlandii. Wreszcie piątym ważnym punktem jest menu Wsparcie. Z tego miejsca możesz skontaktować się z pomocą techniczną AWS oraz przeglądać fora i strony internetowe z dokumentacją, aby uzyskać więcej informacji lub pomocy.

Uruchamianie instancji EC2

Aby uruchomić instancję EC2, musimy przejść do usługi EC2 w menu Compute. Następnie zobaczymy asystenta zadającego nam kilka pytań w celu wdrożenia instancji. Tutaj możemy skonfigurować i wybrać typ instancji, w tym specyfikację sprzętu, systemy operacyjne, przypisać dowolny adres IP i

szczegóły bezpieczeństwa (porty, klucze dostępu do maszyn…). Główny ekran można zobaczyć na rysunku



Nadszedł czas, aby zacząć definiować naszą pierwszą instancję; w tym celu wystarczy kliknąć niebieski przycisk o nazwie Uruchom instancję. Jeśli skorzystamy z asystenta, będzie się on składał z siedem ekranów . To są:

* Pierwszy, który jest selektorem systemu operacyjnego. Jeśli zamierzasz skorzystać z oferty free tier, możemy skorzystać z jednego z systemów operacyjnych oznaczonych na zdjęciu free tier. W naszym przypadku zdecydowaliśmy się na Amazon Linux AMI, ale nie ma problemu z używaniem Ubuntu AMI lub któregokolwiek innego.

* Drugi ekran pokazuje typ instancji. Tutaj pojawia się pierwszy problem, ponieważ bezpłatna instancja ma tylko 1 GB pamięci RAM. Tutaj możemy zdecydować się na pozostanie przy 1 GB pamięci RAM, co prawdopodobnie bardzo utrudni Odoo współistnienie z PDI lub skorzystanie z większego, nieobjętego darmową warstwą. W naszym przypadku pozostaniemy przy instancji t2.micro, ponieważ planujemy zainstalować Odoo, ale po zainstalowaniu i utworzeniu testowa baza danych zamyka ją i uruchamia tylko PDI. Nawet z tym PDI nie będzie działać zbyt płynnie z zaledwie 1 GB pamięci RAM, ale jedyną opcją rozwiązania tego problemu jest przejście na płatną instancję.

* Trzeci ekran to konfiguracja naszej instancji. W tym momencie musimy utworzyć VPC, ponieważ ta instancja tego wymaga. Zostawiamy 1 jako liczbę instancji do uruchomienia i upewniamy się, że nie klikamy pola wyboru instancji spot. W sekcji sieciowej klikamy przycisk utwórz nowy VPC, który uruchomi nową stronę. Na tej nowej stronie kliknij niebieski przycisk Utwórz VPC i określ następujące parametry: BiBook jako znacznik nazwy, 172.16.1.0/24 jako blok CIDR IPv4, a pozostałe opcje pozostaw jako domyślne. Spowoduje to utworzenie podsieci w zakresie 172.16.1, który jest prywatnym zakresem adresów IP. Po utworzeniu zamknij nowe okno, wróć do asystenta i naciśnij przycisk odświeżania. Tam powinniśmy zobaczyć naszego VPC. Wybieramy to.

* W sekcji podsieci musimy zrobić coś podobnego. Utwórz nową podsieć zawierającą wszystkie adresy IP z zakresu wybranego dla naszego VPC. Po zakończeniu wróć do asystenta i odśwież go. Nowy zakres podsieci powinien pojawić się na liście do wybrania. Wybierz to.

* W sekcji automatycznego przypisania publicznego adresu IP pozostawiamy opcję domyślną, czyli Wyłącz. Jeśli potrzebujemy publicznego adresu IP, aby połączyć się tam później, co w tym momencie nie ma miejsca, zawsze możemy go przypisać (i zapłacić za to).

* Jeśli korzystasz z ról IAM, możemy tam określić tę, którą chcemy. Ponieważ ich nie używam, pozostaw zaznaczoną opcję Brak.

* Pozostałe opcje pozostaw domyślne, w tym wyłączenie

zachowanie. Upewnij się, że NIE wybierasz Zakończ, jak gdyby wybrano, gdy nasza maszyna zostanie zatrzymana, wszystko zostanie zniszczone i nie będzie możliwości odzyskania zawartości.

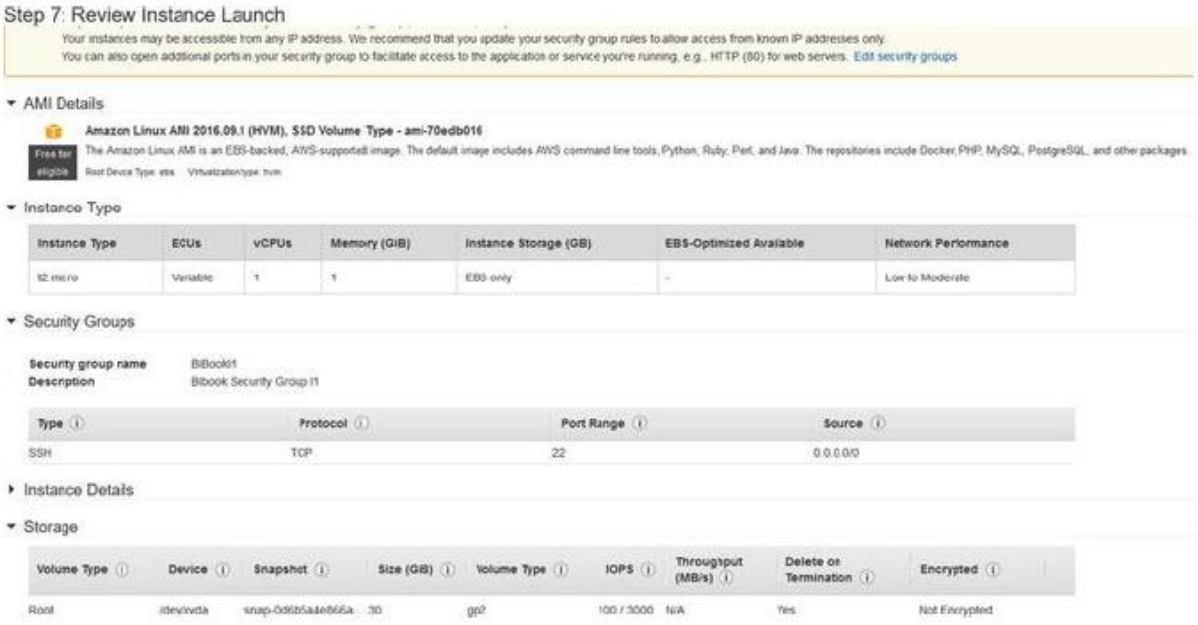
* W sekcji Interfejsy sieciowe upewnij się, że został dodany nowy interfejs o nazwie eth0. Tutaj możesz określić prywatny adres IP w zakresie, który zdefiniowaliśmy wcześniej, lub pozostawić AWS, aby wybrał jeden dla Ciebie. Zostawiam domyślną, a AWS wybierze jedną dla mnie, ale dla łatwości użytkowania może być interesujące wybranie takiej, którą łatwo zapamiętasz.

* Czwarty ekran to sekcja przechowywania. Domyślnie na partycję systemu operacyjnego przydzielono 8 GB. Chociaż może to być dobre, warstwa bezpłatna obejmuje do 30 GB bezpłatnej pamięci SSD. Więc umieścimy 30 zamiast 8. Możesz skonfigurować wiele dysków, ale pamiętaj, że będziesz musiał za nie zapłacić. Zamiast jednak przypisywać je tylko do jednej partycji, możemy zostawić 8 GB na system operacyjny i 22 GB na partycję z danymi. To zależy od Ciebie. W moim przypadku dla uproszczenia przypisuję całe 30 GB tylko do jednego dysku.

* Piąty ekran pozwala zdefiniować kilka tagów identyfikujących naszą maszynę. Zdefiniowałem klucz o nazwie Nazwa i wartość BiBookI1.

* Szósty ekran jest bardzo ważny. Tutaj przypisujemy Security group do naszej instancji. Grupa zabezpieczeń zawiera reguły ruchu przychodzącego i wychodzącego dla naszej zapory. Możemy utworzyć jeden lub ponownie wykorzystać utworzony wcześniej. Domyślnie prosi nas o zestaw adresów IP, aby połączyć się z portem ssh (22). Jeśli pozostawimy domyślne 0.0.0.0, spowoduje to, że instancja będzie dostępna tylko na porcie 22 (SSH) dla całego świata. W idealnej sytuacji chcemy otwierać je tylko dla niektórych adresów IP, takich jak te z naszej firmy, abyśmy mogli zmienić 0.0.0.0 dla zestawu wartości, które chcemy. Nadałem również nazwę BiBookI1 tej grupie bezpieczeństwa i umieściłem BiBook Security Group I1 jako opis.

* Ostatnim krokiem jest etap przeglądu. Upewniamy się, że poprawnie wypełniliśmy wszystkie opcje i możemy uruchomić naszą instancję! Wynik powinien być podobny do rysunku



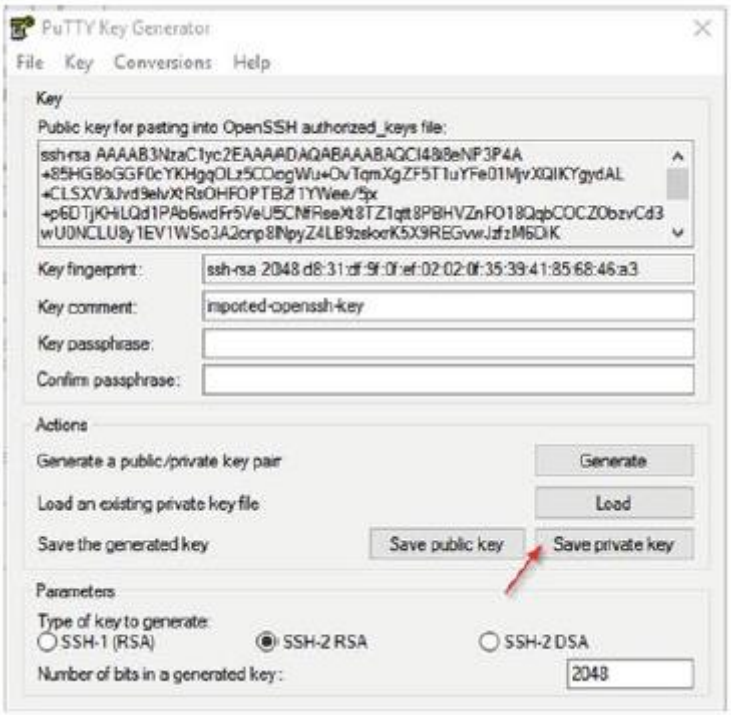
Po zakończeniu pracy z asystentem zostaniemy poproszeni o wybranie pary kluczy lub utworzenie nowej. Są to pary kluczy prywatnych i publicznych. Klucz publiczny zostanie ustawiony na maszynie, natomiast klucz prywatny zostanie pobrany na Twój komputer. Upewnij się, że go nie zgubisz, ponieważ jest potrzebny do połączenia z maszyną. Ponieważ jak dotąd nie mamy żadnego, wybierzmy opcję utwórz nową parę kluczy i nazwijmy ją BiBookKey. Następnie kliknij Pobierz parę kluczy, a twój klucz prywatny, plik .pem, plik tekstowy zawierający twój klucz prywatny, zostanie pobrany na twoją maszynę. Następnie możesz kliknąć niebieski przycisk Uruchom instancję. Ten proces zajmie kilka sekund. Następnie kliknij niebieski przycisk Wyświetl instancje, a zostaniesz przekierowany na stronę instancji EC2 AWS. Jak widać, jeszcze przed uruchomieniem mamy już prywatny adres IP. W moim przypadku nazwa maszyny to: ip-172-16-146-1.eu-west-1.compute.internal i powiązany prywatny adres IP: 172.16.146.1. Aby móc się z nim połączyć, czekamy kilka sekund, aż instancja zostanie zainicjowana i będziemy gotowi do połączenia. Oczywiście w tym momencie mamy tylko prywatny adres IP, więc nie ma możliwości, abyśmy mogli się z nim połączyć. Możemy zacząć wdrażać naszą prywatną sieć wewnątrz, ale nie jest to głównym celem tej książki. Aby umożliwić połączenie z maszyną, w tym momencie utworzymy elastyczny adres IP (publiczny adres IP) i podłączymy go do sieci. Obejmuje to dodatkowe koszty, ale nie tak duże. Kliknij instancję prawym przyciskiem myszy, wybierz opcję Sieć, a następnie Zarządzaj adresami IP. W nowym oknie dialogowym kliknij Przydziel elastyczny adres IP. Spowoduje to otwarcie nowego okna, wybranie VPC w zakresie i kliknięcie Przydziel. System zwróci nam komunikat podobny do następującego: Żądanie nowego adresu powiodło się. Elastyczny adres IP: 34.250.160.101. To jest nasz publiczny adres IP, którego będziemy używać do łączenia. Uderzaj blisko. Otworzy się nowe okno. Elastyczne okno IP. Teraz musimy powiązać nasze publiczne IP z naszą instancją. Wypełnij opcje jedyną możliwą opcją i kliknij Połącz. W tym momencie mamy już wszystko, czego potrzebujemy, aby połączyć się z naszą instancją.

Łączenie z naszą instancją EC2

Teraz, gdy mamy publiczny adres IP, możemy połączyć się z naszą instancją. Jeśli jesteśmy w systemie Linux/Unix, możemy użyć ssh i określić nasz klucz pem. Coś takiego:

ssh -i "BibookKey.pem" ec2-user@ec2-34-250-160-101.eu-west-1.compute.amazonaws.com. Możesz zmienić nazwę DNS według publicznego adresu IP. Jeśli korzystamy z systemu Windows, najpierw

musimy wykonać dodatkowy krok. Obejmuje to konwersję formatu pem w celu przekonwertowania naszego pliku PEM na plik ppk, którego może używać PuTTY; w przeciwnym razie podczas próby połączenia przy użyciu programu PuTTY otrzymamy komunikat o błędzie podobny do następującego: Nie można użyć pliku klucza „C:\Users\Albert\Downloads\BibookKey.pem” (klucz prywatny OpenSSH SSH-2). Aby wykonać tę konwersję, potrzebujemy PuttyGen, dostępnego na tej samej stronie co Putty. Otwórz PuttyGen, kliknij Załaduj istniejący plik klucza prywatnego . Teraz naciśnij przycisk Zapisz klucz prywatny, jak wskazuje strzałka na rysunku.



Spowoduje to utworzenie pliku ppk, który współpracuje z programem Putty. Nadaj mu sensowną nazwę i nie zgub jej. Nadszedł czas, aby otworzyć PuTTY i połączyć się z naszą instancją. Wpisz publiczny adres IP w polu Nazwa hosta w sekcji Sesja, a następnie przejdź do sekcji Uwierzytelnianie w obszarze Połączenie > SSH. W pliku klucza prywatnego do uwierzytelnienia kliknij przeglądaj i wybierz ostatnio utworzony plik ppk. Po tym możemy się połączyć. Jeśli nie określiłeś domyślnego użytkownika logowania, musimy go podać. Ten klucz jest przeznaczony dla dołączonego użytkownika ec2-user. Określ więc użytkownika i naciśnij klawisz Return. Po zakończeniu zostanie wyświetlony ekran podobny do następującego:

```
   login as: ec2-user
   Authenticating with public key "imported-openssh-
key"

       __|  __|_  )
       _|  (     /   Amazon Linux AMI
      ___|\___|___|

   https://aws.amazon.com/amazon-linux-ami/2016.09-
release-notes/
   [ec2-user@ip-172-16-1-67 ~]$
```

Dobre wieści. Zakończyliśmy tworzenie naszej instancji EC2. Teraz nadszedł czas, aby zainstalować programy, takie jak te pokazane w poprzednich rozdziałach tej książki. Następnie przejdźmy do stworzenia naszej instancji RDS do przechowywania naszego magazynu danych MySQL.

Uruchomienie bazy danych RDS

Uruchomienie i uruchomienie instancji to tylko część zadań potrzebnych do skonfigurowania naszego środowiska. Potrzebujemy jeszcze rdzenia, jakim jest nasza hurtownia danych. Aby to osiągnąć, mamy kilka opcji. Możemy samodzielnie uruchomić kolejną instancję i zainstalować bazę danych. Na szczęście istnieje lepsze rozwiązanie w środowisku chmurowym, które wymaga od naszego dostawcy chmury udostępnienia zarządzanej instancji bazy danych. Pozwala to uniknąć kłopotów z instalacją i konfiguracją. Mamy centralny pulpit nawigacyjny, w którym możemy tworzyć lub planować kopie zapasowe i nic więcej. Nie musimy się martwić ustawianiem skomplikowanych parametrów w pliku konfiguracyjnym, instalacją czy konserwacją. Wszystko, co musimy zrobić, to uruchomić instancję RDS i jesteśmy gotowi, aby zacząć z niej korzystać. Przejdź do sekcji Bazy danych i wyszukaj RDS. Kliknij Rozpocznij teraz. Spowoduje to otwarcie asystenta tworzenia instancji RDS. Mamy do wyboru kilka technologii i to jest Twój wybór. Jeśli udajemy, że Odoo działa również w chmurze, możemy stworzyć PostgreSQL dla metadanych Odoo i MySQL/MariaDB dla hurtowni danych. Dla celów ilustracyjnych jako hurtownię danych wybierzemy MariaDB. Asystent przechodzi do drugiego etapu procesu. W tym drugim kroku zostajemy zapytani, jakiego typu instancji MariaDB chcemy. Istnieje opcja o nazwie Dev/Test, która jest uwzględniona w warstwie bezpłatnego użytkowania. Wybierz ten i kliknij Następny krok. W trzecim kroku możemy skonfigurować naszą bazę danych. Jeśli chcemy pozostać przy darmowej warstwie, wybierzmy klasę instancji db.t2.micro, która oferuje bazę danych z 1 vCPU i 1 GB pamięci RAM. Na razie alokujemy 5GB przestrzeni dyskowej, wybieramy wersję, którą chcemy lub pozostawiamy domyślną, jako identyfikator instancji db przypisujemy nazwę BiBook i ustalamy żądaną nazwę użytkownika i hasło. Czwarty krok obejmuje zaawansowaną konfigurację, którą należy poprawnie wypełnić. Zacznijmy od sekcji dotyczącej sieci i bezpieczeństwa. W sekcji sieci i bezpieczeństwa bardzo ważne jest, abyśmy odpowiednio wypełnili te opcje, w przeciwnym razie otrzymamy bezużyteczną instancję, z którą nie możemy się połączyć. Wybierz utworzoną wcześniej VPC o nazwie BiBook. W grupie podsieci wybierz opcję Utwórz nową grupę podsieci bazy danych. Na publicznie dostępnej liście rozwijanej możemy zdecydować, czy chcemy nadać naszej instancji publiczny adres IP, czy nie. Jeśli przypiszemy publiczny adres IP, będziemy mogli łączyć się z zewnątrz, natomiast jeśli odmówimy, będziemy mogli łączyć się tylko z wnętrza naszego VPC. Ponieważ nie planujemy łączyć się z VPC z zewnątrz, wybieramy NIE. Upewnij się, że jeśli chcesz bezpośrednio wchodzić w interakcje z bazą danych z zewnętrznych komputerów, np. firmowych, wybierz Tak i nadaj instancji publiczny adres IP. Ze względów wydajnościowych w strefie dostępności wybierzemy tę samą, w której utworzono naszą instancję, w naszym przypadku eu-west-1c. W ostatnim kroku w tej sekcji upewnij się, że wybrałeś BiBookI1 jako grupę zabezpieczeń VPC. W opcjach bazy danych wpisz DWH jako nazwę bazy danych i pozostaw 3306 jako port bazy danych. Pozostałe opcje pozostaw określone domyślnie. Zmień zasady tworzenia kopii zapasowych zgodnie z potrzebami Twojej firmy, a po zakończeniu przejrzyj sekcję konserwacji i zmień domyślne wartości na takie, które Ci odpowiadają, aby skonfigurować na przykład okna konserwacji aktualizacji na weekendy. Po zakończeniu sprawdź, czy wszystko jest w porządku i kliknij przycisk Uruchom instancję DB. W tym momencie otrzymamy bardzo dziwny błąd. Grupa podsieci DB nie spełnia wymagań pokrycia strefy dostępności. Dodaj podsieci, aby objąć co najmniej dwie strefy dostępności. Obecny zasięg: 1 Musimy ponownie przejść do naszej konfiguracji VPC, sekcji podsieci i utworzyć nową podsieć w innej strefie dostępności, na przykład eu-west-1b i ustawić CIDR na 172.16.2.0/24. Skończ nazywając go BiBookRDS. Odśwież i uderz w instancję LaunchDB. W tym momencie wszystko powinno być w porządku i pojawi się wyskakujące okienko podobne do tego: Nowa grupa podsieci DB (default-vpc-3fca455a) została pomyślnie

utworzona dla ciebie w vpc-3fca455a. Naciśnij przycisk Wyświetl nasze instancje bazy danych i spójrz na już utworzoną instancję. Kolumna statusu pokaże tworzenie, więc musimy chwilę poczekać. W międzyczasie wrócimy do naszej instancji EC2 w Putty i zainstalujemy klienta MySQL.

Uwaga: jeśli pojawi się błąd podobny do poniższego, VPC musi mieć co najmniej 2 podsieci, aby utworzyć grupę podsieci DB. Przejdź do konsoli zarządzania VPC, aby dodać podsieci, co oznacza, że w naszym VPC zdefiniowaliśmy tylko podsieć i zużyłeś wszystkie adresy IP na tym VPC. Nie ma łatwego rozwiązania tego problemu, więc rozwiązanie polega na zakończeniu instancji EC2 i usunięciu podsieci, a następnie odpowiednim ich utworzeniu.

Aby zainstalować klienta, możemy użyć następującego zdania:

```
   [ec2-user@ip-172-16-1-67 ~]$ sudo yum install mysql
   Loaded plugins: priorities, update-motd, upgrade-
helper
   ... (Output truncated)
   Resolving Dependencies
   --> Running transaction check
   ---> Package mysql.noarch 0:5.5-1.6.amzn1 will be
installed
   --> Processing Dependency: mysql55 >= 5.5 for
package: mysql-5.5-1.6.amzn1.noarch
   --> Running transaction check
   ---> Package mysql55.x86_64 0:5.5.54-1.16.amzn1
will be installed
   --> Processing Dependency: real-mysql55-libs(x86-
64) = 5.5.54-1.16.amzn1 for package: mysql55-5.5.54-
1.16.amzn1.x86_64
   --> Processing Dependency: mysql-config for
package: mysql55-5.5.54-1.16.amzn1.x86_64
   --> Running transaction check
   ---> Package mysql-config.x86_64 0:5.5.54-
1.16.amzn1 will be installed
   ---> Package mysql55-libs.x86_64 0:5.5.54-
1.16.amzn1 will be installed
   --> Finished Dependency Resolution

   Dependencies Resolved

   ...  (Output truncated)

   Installed:
     mysql.noarch 0:5.5-1.6.amzn1

   Dependency Installed:
     mysql-config.x86_64 0:5.5.54-
1.16.amzn1   mysql55.x86_64 0:5.5.54-1.16.amzn1
     mysql55-libs.x86_64 0:5.5.54-1.16.amzn1

   Complete!
```

Następnie nadszedł czas, aby wrócić do menu RDS i poszukać punktu końcowego nowo utworzonej bazy danych. Powinno to wyglądać mniej więcej tak:

Endpoint: bibook.c3nefasein6d.eu-est-1.rds.amazonaws.com:3306

Przed próbą połączenia musimy zmodyfikować naszą grupę zabezpieczeń. Jeśli pamiętasz, tylko port 22 mógł się połączyć. Jeśli najedziesz kursorem myszy obok nazwy punktu końcowego, nad polem Brak

uprawnień przychodzących, zobaczysz przycisk edycji grupy zabezpieczeń. Przejdź tam i dodaj nową regułę ruchu przychodzącego dla portu 3306 i otwórz ją dla zakresu adresów IP, którym chcesz zaufać. Nowe reguły przychodzące powinny być takie, jak pokazano na rysunku

Po dodaniu nowej reguły kliknij przycisk Zapisz i wróć do strony instancji RDS. Teraz zamiast Brak uprawnień przychodzących powinieneś zobaczyć coś w stylu (autoryzowany). Możemy teraz połączyć się z naszą instancją, więc wróć do EC2 i wpisz następujące polecenie (zastępując nazwę hosta instancją RDS, nazwę użytkownika root nazwą użytkownika wybraną w asystencie i słowo kluczowe hasło wybranym hasłem).

mysql -h bibook.c3nefasein6d.eu-west-

1.rds.amazonaws.com -u root –ppassword

Wszystko idzie dobrze, klient powinien się połączyć i powinien pojawić się baner MariaDB:

```
   Welcome to the MySQL monitor.  Commands end with ;
or \g.
   Your MySQL connection id is 28
   Server version: 5.5.5-10.0.24-MariaDB MariaDB
Server

   Copyright (c) 2000, 2016, Oracle and/or its


affiliates. All rights reserved.

   Oracle is a registered trademark of Oracle
Corporation and/or its
   affiliates. Other names may be trademarks of their
respective
   owners.

   Type 'help;' or '\h' for help. Type '\c' to clear
the current input statement.

   mysql>
```

Wydaj polecenie show databases i upewnij się, że twoja baza danych DWH pojawi się na liście:

```
mysql> show databases;
+--------------------+
| Database           |
+--------------------+
| dwh                |
| information_schema |
| innodb             |
| mysql              |
| performance_schema |
+--------------------+
5 rows in set (0.01 sec)
```

Tak jak się spodziewaliśmy, wszystko jasne. Mamy teraz skonfigurowaną platformę w chmurze! Teraz nadszedł czas, aby utworzyć wymagane tabele i rozpocząć ich wypełnianie.

Obliczanie cen za pomocą kalkulatora AWS

Posiadanie platformy w chmurze kosztuje. Aby obliczyć lub przewidzieć, ile może Cię to kosztować, AWS oferuje bezpłatny kalkulator online. Kalkulator można znaleźć w poniższym linku:

https://calculator.s3.amazonaws.com/index.html

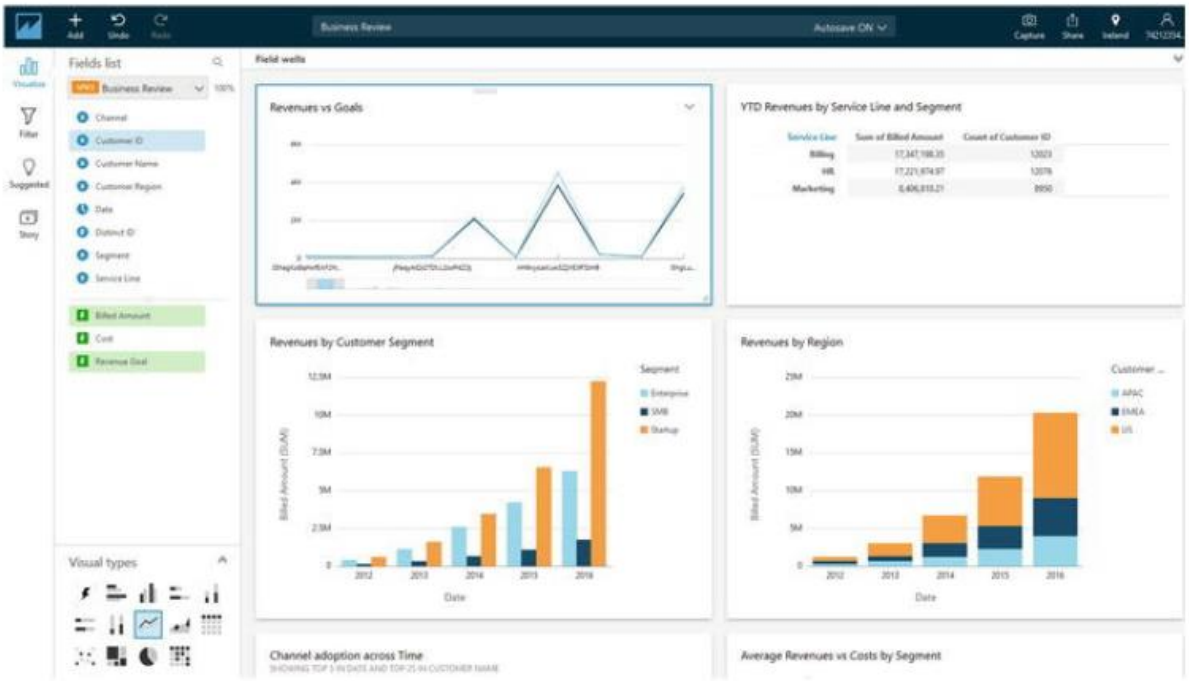
Nie przejmuj się tym, co widzisz! Wystarczy wypełnić kilka miejsc. Przede wszystkim musimy wybrać region geograficzny. W naszym przypadku jest to Europa/Irlandia, ale w innych przypadkach może być inaczej. Wybierz ten, którego planujesz użyć, który powinien być najbliższy Tobie. Po wypełnieniu regionu wystarczy kliknąć znak plus w Compute: Amazon EC2 Instances. Wpisz dowolną znajomą nazwę, dostosuj typ instancji, której chcesz użyć, oraz opcję rozliczeń. W naszym przypadku nie zdecydowaliśmy się na żadną płatność z góry (choć są soczyste rabaty), więc zostaniemy wybrani na żądanie (bez umowy). Na elastycznym IP wybierz 1, a pozostałe pola zostaw jako 0, ponieważ pomimo tego, że będziemy go przypisywać, nie będzie zbyt wielu remapów, a kilka pierwszych jest darmowych. Sekcja Transfer danych jest bardzo ważna. Amazon rozlicza się z danych przychodzących i wychodzących z ich serwerów. Jednak przychodzący transfer danych jest bezpłatny, podczas gdy wychodzący transfer danych jest rozliczany po zużyciu 1 GB na region. Jeśli więc udajesz, że pobierasz dużo danych z serwera na swój komputer, może to nie być dla Ciebie dobre rozwiązanie i lepiej spojrzeć na inne usługi, które oferują przepustowość w niższej cenie. Ponieważ nie planujemy pozwalać ludziom łączyć się bezpośrednio z tym serwerem, tylko programiści będą mieli do niego dostęp do testów, więc planujemy bardzo małe zużycie, powiedzmy 10 GB. Dane przychodzące będą danymi pochodzącymi z petycji od klientów, które nie powinny być bardzo duże, a w każdym razie są bezpłatne. Jeśli spojrzysz teraz na drugą zakładkę, Oszacowanie miesięcznego rachunku, zobaczysz, że trochę pieniędzy jest przydzielanych na różne koncepcje. Jeśli wybierzesz bezpłatną mikroinstancję, również zobaczysz zniżkę. Całkowita przepustowość będzie nas kosztować tylko mniej niż dolara, więc jesteśmy w porządku. Nadal musimy dodać instancję bazy danych RDS, więc spójrz na lewy panel, piątą kartę i przejdź do RDS. Kliknij ponownie znak plus, dodaj fantazyjny opis, podaj szczegóły swojej instancji i dostosuj zużycie przepustowości. W przypadku bazy danych zwróć uwagę, ponieważ większość zastosowań będzie polegać na przesyłaniu danych, więc jest to bezpłatne. Wtedy, jeśli byliśmy wystarczająco mądrzy i umieściliśmy RDS w tej samej strefie dostępności, co nasz silnik aplikacji ETL i raportowania (w razie potrzeby), nie będziemy musieli płacić za transfer danych wewnątrz regionu. Musimy więc rozliczyć się tylko z danych przesłanych na serwer do naszych narzędzi klienckich do raportowania. Większość narzędzi do raportowania przekazuje obliczenia do silników baz danych, co

pozwala zaoszczędzić dużo przepustowości, aw naszym przypadku pieniędzy, ponieważ tylko obliczenia najwyższego poziomu, które muszą być wyświetlane lub przeglądane na pulpicie nawigacyjnym, przechodzą przez sieć. W każdym razie jest to miejsce, w którym nastąpi większość zużycia sieci. Zależy to oczywiście od ilości informacji, które pobieramy z serwera, użytkowników uzyskujących do nich dostęp, sposobu zaprojektowania pulpitów nawigacyjnych oraz poziomu szczegółowości informacji. Aby być bardzo bezpiecznym, zakładamy 50 GB miesięcznie, ale jest to oczywiście subiektywne przypuszczenie. Możesz chcieć dodać inne usługi, z których planujesz korzystać, ale ostatecznie rachunek wynosi mniej niż 40 USD miesięcznie. Możesz także pobawić się i zobaczyć wpływ wynajmu lepszych i mocniejszych serwerów oraz sprawdzić wpływ, jaki będzie to miało na Twój rachunek. Jeśli masz warstwę darmową, zobaczysz, że trzymając się mikroinstancji, będziesz musiał zapłacić tylko za przepustowość i dzierżawę adresu IP w ciągu pierwszego roku. Jest to więc doskonały sposób na rozpoczęcie. Pełne ładunki, które obliczyliśmy, pokazano na rysunku

⊖	Amazon EC2 Service (Europe)		$ 18.30
	Compute:	$ 14.64	
	Elastic IPs:	$ 3.66	
⊖	Amazon RDS Service (Europe)		$ 14.45
	DB instances:	$ 13.18	
	Storage:	$ 1.27	
⊕	AWS Data Transfer In		$ 0.00
⊖	AWS Data Transfer Out		$ 5.31
	Europe (Ireland) Region:	$ 5.31	
⊕	AWS Support (Basic)		$ 0.00
Free Tier Discount:			$ -30.35
Total Monthly Payment:			$ 7.71

Zabawa z AWS Quicksight

W ciągu ostatnich kilku miesięcy Amazon udostępnił AWS Quicksight. Quicksight to szybkie rozwiązanie do raportowania w chmurze i tworzenia pulpitów nawigacyjnych, które jest niezwykle łatwe w użyciu. Quicksight może korzystać z wielu źródeł danych, takich jak pliki Excel i CSV, bazy danych hostowane w AWS, w tym RDS, Aurora, Redshift, a nawet S3 Storage. Ale nie tylko chmura. Możesz także wskazać Quicksight lokalne relacyjne bazy danych, a nawet połączyć się z innymi aplikacjami w chmurze, takimi jak Salesforce. Silnik SPICE jest silnikiem kolumnowym w pamięci, który jest bardzo mocny i szybki. Może skalować się do wielu użytkowników i replikować dane w różnych miejscach. Quicksight oferuje również bardzo piękny interfejs wizualizacji i jest bardzo łatwy w użyciu. Jak większość narzędzi bi, integracja z urządzeniami mobilnymi jest możliwa dzięki niedawno dodanej obsłudze iPhone'a i nadchodzącym wersjom Androida. Możesz udostępniać i pracować w środowisku współpracy, tak jak w wielu innych narzędziach bi, i tworzyć historie podobne do tych, które możesz tworzyć za pomocą QlikSense. W niedalekiej przyszłości planowana jest również integracja z większością dostawców BI, takich jak Tableau, Qlik i TIBCO, więc dzięki tym narzędziom będziesz mógł uzyskać dostęp do danych w silniku SPICE i pracować z nimi bezpośrednio w preferowanym narzędziu BI, zwiększając silniki przetwarzania wbudowane w te narzędzia. Quicksight jest bezpłatny do użytku osobistego do 1 GB danych w silniku SPICE, a do użytku korporacyjnego dostępne są tanie plany. Jak zawsze w przypadku wszystkich usług AWS, płacisz więcej, gdy żądasz więcej. Tak więc oprócz niewielkiej opłaty za użytkownika, miejsce wykorzystywane w silniku SPICE jest również wykorzystywane jako miernik rozliczeniowy, zaczynając od ćwierć dolara za gigabajt miesięcznie. Mały ekran przedstawiający wygląd raportu Quicksight przedstawiono na rysunku



Jak widać na obrazku, po lewej stronie znajduje się podobne menu zawierające wszystkie wymiary i fakty, których można użyć ze źródła danych, oraz duże płótno po prawej stronie. Za pomocą znaku plus na głównym górnym pasku menu możesz dodawać do płótna wizualizacje, które są grafiką lub informacjami tabelarycznymi. Następnie przeciągasz i upuszczasz żądane wymiary i fakty z lewej strony do nowo dodanej wizualizacji i generowana jest grafika. W każdej chwili możesz zmienić kształt i typ wizualizacji wybierając jedną z dostępnych na liście Typy wizualizacji w dolnej części ekranu. Za pomocą kilku kliknięć myszką masz przejrzysty i uporządkowany interfejs zawierający kilka znaczących wykresów do analizy. Po krótkim przetestowaniu Quicksight stwierdzamy, że jest on bardzo podobny do QlikSense pod względem wizualizacji, więc dla użytkowników, którzy mają już doświadczenie z QlikSense, uznają go za przydatny. Oczywiście ukrywa większość złożoności za standardowymi narzędziami BI, ponieważ nie ma edytora skryptów, a wszystkie dostępne informacje są wbudowane w ten ekran. Pod tym względem jest również podobny do silnika asocjacyjnego Qlik. Z pewnością narzędzie do testowania przez osoby mniej doświadczone w narzędziach do raportowania i kokpitów, które chcą uciec od złożoności definiowania skomplikowanych obliczeń w skryptach i chcą mieć łatwy sposób przeglądania swoich danych za pomocą zaledwie kilku kliknięć.

Korzystanie z oprogramowania z AWS Marketplace

Innym podejściem do wdrożenia rozwiązania w chmurze AWS jest użycie urządzenia programowego. Urządzenie programowe to rozwiązanie stworzone przez jakiegoś dostawcę, które obejmuje całą konfigurację potrzebną do tego narzędzia. Następnie dostawca, w zależności od tego, jaki typ lub rozmiar platformy wybierzesz, dobierze dla Ciebie najlepszą infrastrukturę z AWS, która pasuje do potrzeb narzędzia. Ostatecznym kosztem eksploatacji urządzenia będzie koszt zasobów AWS niezbędnych do uruchomienia platformy wraz z kosztami narzędzia w zakresie licencji lub wszelkich innych opłat, które możesz ponieść. Możesz przeglądać urządzenia AWS na jego rynku tutaj:

https://aws.amazon.com/marketplace/

ale jest kategoria wyposażona we wszystkie narzędzia Business Intelligence, z ponad 350 urządzeniami w poniższym linku:

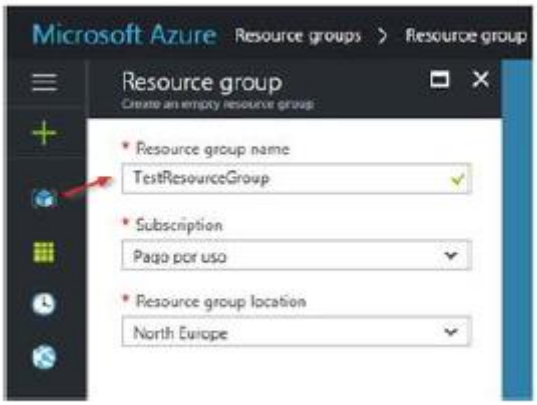
https://aws.amazon.com/marketplace/b/2649336011?ref_=header_nav_category_2649336011 . Na rynku zobaczysz aplikacje takie jak Tibco Spotfire, Tableau i Qlik, a także narzędzia ETL, takie jak Informatica i bazy danych, takie jak Exasol, Teradata, Oracle, MySQL, MariaDB i inne. Po prostu wybierasz ten, który chcesz, środowisko utworzy wymagane zasoby w konsoli AWS, a gdy tylko się pojawią, zostaniesz obciążony. Kiedy nie potrzebujesz już platformy, możesz ją zamknąć i usunąć, a nie będziesz już obciążany, tak jak w przypadku każdego innego zasobu AWS.

Microsoft Azure

Implementacja podobnego rozwiązania na Azure jest jak najbardziej możliwa. Sposób, w jaki działa Azure, przypomina AWS. Rozwiązanie Microsoft jest drugim co do wielkości dostawcą chmury, więc jest to również doskonały wybór. Nazwy usług nieco się zmieniają, ale łatwo je znaleźć. Wdrożenie serwera obejmuje kilka rzeczy, ale pierwszą z nich jest skonfigurowanie subskrypcji, która zwykle będzie subskrypcją płatną za użytkowanie, bardzo podobną do AWS. Możemy zacząć od przejrzenia następującego adresu URL: https://portal.azure.com .

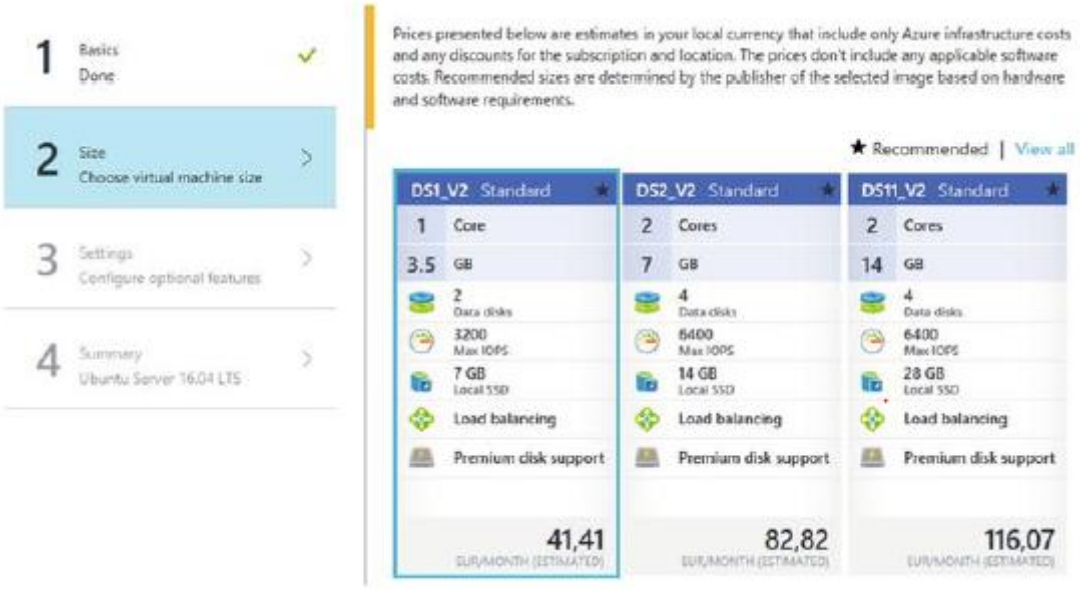
Tworzenie maszyny na platformie Azure

Po skonfigurowaniu subskrypcji (i powiązaniu karty kredytowej) będzie można utworzyć grupę zasobów powiązaną z konkretną subskrypcją utworzoną w poprzednim kroku. Wszystkie zasoby, które utworzysz, będą domyślnie połączone z wybraną grupą zasobów, a te z kolei zostaną połączone z wybraną subskrypcją. Ta grupa zasobów, podobnie jak w AWS, jest połączona z jednym z dostępnych regionów geograficznych. Na poniższym rysunku pokazano, jak utworzyć grupę zasobów.



Jak widać na poprzednim rysunku, grupa zasobów zostanie utworzona w strefie geograficznej Europa Północna. Po ustawieniu grupy zasobów możemy utworzyć nową maszynę. Aby to osiągnąć, w lewym okienku menu szósty przycisk nosi nazwę Maszyny wirtualne (klasyczne). Jeśli naciśniemy ten przycisk, okienko rozwinie się na ekranie głównym i zapyta nas, jaki typ systemu operacyjnego chcemy dla naszej maszyny. W naszym przykładzie wybierzemy Ubuntu, następnie na ekranie pojawi się kilka odmian Ubuntu i wybierzemy wersję LTS, ponieważ są one bardziej stabilne i mają rozszerzoną obsługę. W tym momencie ostatni LTS dostępny na platformie Azure dla Ubuntu to 16.04, więc wybierzemy ten. W rozwijanym polu modelu wdrożenia wybierzemy Klasyczny. Po wybraniu tej opcji zaczynamy konfigurować nasze maszyny. Jest to podzielone na cztery etapy. Pierwszym krokiem jest skonfigurowanie podstawowych ustawień: nazwy maszyny, nazwy użytkownika do logowania, uwierzytelnienia dostępu do maszyny, które w Azure może być poza kluczem publicznym SSH, hasła w przeciwieństwie do domyślnych AWS, które pozwala tylko aby połączyć się za pomocą klucza publicznego. Następnie musisz przypisać tę maszynę do określonej subskrypcji, a jeśli wcześniej utworzyłeś już grupę zasobów, możesz ją również przypisać do grupy zasobów. Jeśli wcześniej nie

utworzyłeś grupy zasobów, możesz to zrobić tutaj automatycznie, a ten komputer zostanie dodany do nowej grupy zasobów. W tym kroku można również określić lokalizację. Niektóre lokalizacje można wybrać w zależności od zdefiniowanej subskrypcji, więc jeśli tak jest, wybierz inną lokalizację. Drugi krok pokazuje wymiarowanie i wycenę maszyny. Domyślnie asystent próbuje wybrać kilka możliwych opcji, ale możesz przełączyć się na wszystkie opcje, klikając Wyświetl wszystko w prawym górnym rogu. Zalecane instancje będą nam odpowiadać w większości przypadków, ponieważ są to instancje ogólnego przeznaczenia. Asystent domyślnie wybiera dla nas trzy, co widać na rysunku



W zależności od potrzebnej mocy, pierwsza instancja może wystarczyć. Pomimo tego, że na ekranie widać tylko 7 GB, instancja ma 2 dyski, które w przypadku dysku magnetycznego są to dyski o pojemności do 50 GB w tej samej cenie co dysk SSD 7 GB. Pamięć masowa premium jest oparta na dyskach SSD, więc na następnym ekranie możesz wybrać, z którym chcesz pozostać. W każdym razie zawsze możesz dodać więcej dysków pamięci później po przyzwoitych kosztach. Jeśli szukasz tańszej opcji, możesz wybrać Wyświetl wszystko i przejść w dół, a znajdziesz podstawowe instancje, a mianowicie A0 i A1, które kosztują od 12 do 17 USD miesięcznie i są dobrym wyborem, jeśli korzystanie z maszyny jest światło. Te dwie maszyny składają się z 0,75 GB i 1,5 GB pamięci i oczywiście nadają się tylko do bardzo lekkich obciążeń. W trzecim kroku możesz skonfigurować jeszcze kilka aspektów nowej maszyny. Zasadniczo w tym kroku konfiguruje się pamięć masową i sieć. W tym kroku wybierana jest usługa w chmurze, do której będzie należeć ta maszyna. Musisz także określić punkty końcowe lub porty, które będą otwarte w tej maszynie. Domyślnie tylko port 22 (ssh) może się łączyć. Możesz swobodnie otwierać potrzebne punkty końcowe. Jako ostatni krok możesz zdefiniować zestaw wysokiej dostępności, ale w tym momencie zrezygnowaliśmy z tego. Ostatnim krokiem jest przegląd wszystkich poprzednich kroków. Jeśli wszystko jest w porządku, przejdź do tworzenia maszyny.

Tworzenie bazy danych na platformie Azure

Tworzenie SQL Database na Azure nie różni się niczym od tego, co widzieliśmy do tej pory. W lewym menu kliknij zielony znak plusa, następnie Bazy danych i na końcu Bazę danych SQL; lub kliknij bezpośrednio na przycisk SQL, który jest piątym przyciskiem pod znakiem plus. Każdy z tych kroków spowoduje utworzenie bazy danych programu SQL Server. Istnieje jednak możliwość wybrania innych silników baz danych, ale w tym celu należy przejść do znaku plus, następnie do baz danych, a przed kliknięciem SQL Database należy kliknąć na Zobacz wszystko w prawym górnym rogu, a następnie

skorzystać z pola wyszukiwania baz danych aby znaleźć żądaną bazę danych, ponieważ nie są one łatwe do znalezienia. Patrz rysunek

Po znalezieniu kliknij MariaDB Server i postępuj zgodnie z menu podobnym do tworzenia maszyny wirtualnej, którą opisaliśmy w poprzednim punkcie. To samo należy zrobić, jeśli chcesz mieć instancję MySQL.

Uwaga: podobnie jak w przypadku AWS, system cen staje się skomplikowany. Na szczęście, jako jego odpowiednik, Microsoft ma kalkulator internetowy, który pomaga we wszystkich cenach. Kalkulator znajduje się pod następującym adresem URL:

https://azure.microsoft.com/enus/pricing/calculator/

Chmura Google

Chmura Google jest trzecim co do wielkości dostawcą. Zwykle uważa się, że chmura Google ma tańsze ceny niż jej rywale. Ponadto mają bardzo dobrą ofertę logowania, zwłaszcza jeśli nadal testujesz lub sprawdzasz koncepcję. W tym momencie oferują 300 USD darmowego kredytu na 12 miesięcy wraz ze wszystkimi zasobami Google w chmurze. To doskonała oferta na początek. Nie mamy wystarczająco dużo miejsca w książce, aby omówić trzeciego dostawcę chmury, ale chcemy zachęcić Cię do wypróbowania ich również. Przy prawidłowym planowaniu maszyn kredyt może wystarczyć na przetestowanie rozwiązania BI w chmurze. Dostęp do bezpłatnej wersji próbnej oraz platformy można uzyskać stąd:

https://console.cloud.google.com/freetrial?pli=1&page=0 .

Aby replikować poprzednią infrastrukturę, którą skonfigurowaliśmy z AWS lub Azure, musisz skupić się na usłudze Compute Engine do tworzenia maszyn wirtualnych i usłudze Cloud SQL do tworzenia bazy danych MYSQL lub PostgreSQL (niestety nie ma jeszcze wsparcia dla MariaDB, ale to nie powinno być dużym problemem). Podobnie jak w przypadku pozostałych dwóch konkurentów Google ma również kalkulator cen. Kalkulator jest dostępny tutaj:

https://cloud.google.com/products/calculator/ i zawiera wszystkie zasoby w chmurze Google.

Rozwiązania chmurowe oparte na dostawcach

Oprócz trzech dostawców usług w chmurze, o których mówiliśmy , istnieje kilka usług w chmurze specyficznych dla dostawców. Usługi te zwykle oferują swoje oprogramowanie jako usługę (SaaS). Z tych, które widzieliśmy w książce, najciekawsze są następujące:

Chmura Oracle: https://cloud.oracle.com/home

Chmura Qliksense: https://www.qlikcloud.com/

Chmura Microstrategy: https://www.microstrategy.com/us/platform/cloud

Chmura Tableau: https://www.tableau.com/products/cloudbi

Salesforce.com to narzędzie CRM, ale obecnie oferuje o wiele więcej niż aplikacje: marketingowe, narzędzia BI, obsługa klienta, aplikacje społecznościowe i czatowe oraz wiele innych. Możesz mieć swój CRM w chmurze oraz rozbudowane dashboardy i raporty wykorzystujące dane generowane przez aplikację z jej poziomu. Jeśli chcesz dowiedzieć się więcej i odkryć narzędzie Salesforce w chmurze, możesz odwiedzić ich stronę internetową pod następującym linkiem: https://www.salesforce.com

Korzystanie z rozwiązania w chmurze jednego z tych dostawców zamiast narzędzia komputerowego pozwoli nam zaoszczędzić trochę zadań konserwacyjnych i administracyjnych. Na przykład aktualizacje wersji są wykonywane automatycznie w chmurze dla wszystkich tych rozwiązań. Dzięki temu administratorzy przenoszą część cięższych zadań administracyjnych na administratorów chmury. Upraszcza to zarządzanie tymi aplikacjami. Również w większości przypadków monitorowanie można skonfigurować bezpośrednio z narzędzia, dzięki czemu otrzymujesz alerty, gdy narzędzie jest niedostępne, a administratorzy narzędzia automatycznie wykonują działania mające na celu przywrócenie narzędzia. Pozostawia to Tobie tylko część programistyczną, a także tworzenie użytkowników i kilka innych aspektów administracyjnych lub poprawiania, podczas gdy większość zadań administracyjnych jest od Ciebie zwolniona.

Wniosek

W tej części zobaczyliśmy, jak wdrożyć środowisko BI w chmurze. Przyjrzeliśmy się dogłębnie głównemu dostawcy: AWS i do pewnego stopnia dwóm jego konkurentom, a mianowicie Azure od Microsoft i Google Cloud od Google. Widzieliśmy, jak przybliżyć koszty prowadzenia całej platformy w chmurze; jakie ma zalety i niedogodności; na jakie ważne rzeczy musimy zwrócić uwagę; oraz kwestie, o których musimy pamiętać, mając infrastrukturę w chmurze, takie jak bezpieczeństwo, monitorowanie i kontrola kosztów. Na koniec tej części zobaczyliśmy, jak niektórzy dostawcy BI stworzyli specyficzne środowisko chmurowe dla swoich produktów oraz jakie zalety może mieć takie rozwiązanie w porównaniu ze standardowymi wdrożeniami w zakresie administracji i konserwacji.

13. Wnioski i dalsze kroki

Cóż, prawie skończyliśmy - jeszcze tylko kilka stron cierpienia. Ale jeśli przybyłeś, to może tutaj nie cierpisz tak bardzo; więc bardzo się cieszymy, że czytasz te strony. Nie zrobiliśmy tego tak źle… Ale teraz, kiedy dotarliśmy do tego punktu, co więcej możemy zrobić? To powinno być pytanie po pomyślnym zakończeniu wdrożenia Twojego systemu BI wraz z podstawową analizą danych znajdujących się w Twojej bazie danych, do których dostęp uzyskujesz z Twojego systemu BI w wielu środowiskach, które mogą znajdować się w całości lub częściowo w chmurze. Oczekujemy, że postępowałeś zgodnie z tekstem, postępując zgodnie z przykładami, pobierając i instalując proponowane oprogramowanie (i być może inne opcje, o których słyszałeś, zwłaszcza dla interfejsu BI istnieje wiele opcji z darmowymi wersjami narzędzi komercyjnych), poznając różnych narzędzi podczas grania zgodnie z naszymi instrukcjami, abyś mógł ukończyć , mając co najmniej wstępną weryfikację koncepcji. Oczywiście zdajemy sobie sprawę, że jeśli przeczytałeś to od końca do końca podczas testowania, nie będziesz mieć produktywnego środowiska wielośrodowiskowego ani wszystkich serwerów w chmurze; rozumiemy, że pierwszą opcją dla tego rodzaju testu jest po prostu laptop, aby zainstalować wszystkie wymagane komponenty do ich oceny. Innymi słowy, jest całkiem możliwe, że po pewnym czasie będziesz mógł wrócić do najnowszych rozdziałów, gdy już pokażesz swoim interesariuszom kilka przykładów tego, co BI może zrobić dla Ciebie i Twojej firmy, więc kiedy wszyscy zgodzą się na kontynuację BI projekt, o którym możesz pomyśleć, aby przenieść to środowisko piaskownicy do prawdziwego środowiska z wieloma serwerami, być może kupić kilka licencji, aby uzyskać lepszą wydajność i funkcjonalność po stronie narzędzia BI; lub myślenie w rozwiązaniu chmurowym, aby zlokalizować wszystkie rzeczy wymagane dla platformy BI, o ile wdrożenie działającego rozwiązania nie polega tylko na wymaganym rozwoju, aby zobaczyć dane, ale także na powiązanym zarządzaniu, aby utrzymać je w ruchu. Twoje pytanie w tym momencie mogłoby brzmieć: czy jest coś jeszcze, co powinieneś rozważyć, biorąc pod uwagę kolejne kroki i przyszłość? Odpowiedź brzmi: tak, zawsze mamy coś więcej do wdrożenia, udoskonalenia i oceny. Znajdujemy się w niesamowitym scenariuszu analizy danych, z nowymi funkcjami i koncepcjami, które pojawiają się każdego dnia i musisz zachować czujność, jeśli chcesz czerpać korzyści ze wszystkich funkcji oferowanych przez technologie Business Intelligence i powiązane oprogramowanie. Nie chcemy zbytnio rozszerzać tego rozdziału, ale chcielibyśmy wspomnieć o kilku zaleceniach dotyczących dokumentacji, których nie mamy skomentowanych w ogóle. Z drugiej strony chcielibyśmy wspomnieć o dwóch trendach, które naszym zdaniem rozprzestrzenią się w organizacjach w ciągu najbliższego roku, czyli oprogramowaniu do zarządzania procesami biznesowymi (BPMS) i Big Data.

Dokumentacja

Nasze zalecenia dotyczące dokumentacji są zgodne z niektórymi zdaniami, które zostały już skomentowane w Części 2, kiedy mówiliśmy o Manifeście Agile. Naszym zdaniem dokumentacja jest ważna, ale nie jest głównym celem projektu. Chcielibyśmy przedstawić tylko kilka zaleceń:

* Zaoszczędź trochę czasu, aby udokumentować wszystkie ważne rzeczy podczas wykonywania działań: Może się to wydawać czymś, co zajmuje Ci czas podczas instalacji, konfiguracji lub programowania, ale jeśli spojrzysz na to z ogólnej perspektywy, zobaczysz, że oszczędzasz czas tak daleko ponieważ nie otwierasz ponownie menu instalacji, aby wykonać zrzuty ekranu, nie pomijasz odpowiednich parametrów w dokumentacji, możesz zauważyć, że rzeczy, które były bardziej skomplikowane do rozwiązania itp.

* Unikaj wyczerpującej dokumentacji: z naszego doświadczenia wynika, że wyczerpująca dokumentacja nigdy nie zostanie przeczytana. Musisz udokumentować tylko te rzeczy, które były istotne podczas twoich operacji, unikając tych, które są już udokumentowane w instrukcji instalacji lub

które możesz łatwiej sprawdzić, uzyskując dostęp do systemu niż przez dostęp do dokumentacji. Nie potrzebujesz dokumentacji mówiącej, że kolumna MONTH_ID tabeli T_L_MONTH to NUMBER(6). Możesz to łatwo sprawdzić w bazie danych.

* Repozytorium dokumentów: użyj jednego współdzielonego miejsca, aby scentralizować całą dokumentację na jednej platformie, gdzie każdy, kto potrzebuje dostępu, może uzyskać wszystkie wymagane dokumenty do swojej codziennej pracy. Unikaj używania lokalnych zasobów do zapisywania dokumentów. Masz wiele platform dostępnych za darmo: Dysk Google, Dropbox, One Drive itp.

* Wersjonowanie: jeśli to możliwe, używaj narzędzia, które umożliwia zapisywanie poprzednich wersji dokumentu bez konieczności kontroli wersji wewnątrz dokumentów. Jest to łatwiejszy sposób zarządzania dokumentacją.

* Lepsze wiele małych dokumentów niż jeden duży: Jest to bardziej związane z naszymi preferencjami, ale widzimy, że łatwiej jest zarządzać małymi dokumentami wyjaśniającymi jeden temat niż dużymi dokumentami zawierającymi setki tematów w środku. W ten sposób będziesz mieć kilka wersji każdego dokumentu, o ile każdy temat będzie wymagał mniej modyfikacji niż duży z setkami stron.

Oprogramowanie BMP

Oprogramowanie do zarządzania procesami biznesowymi oferuje pomoc w poprawie wyników biznesowych, koncentrując się na ulepszaniu procesów biznesowych w oparciu o teorie ciągłego doskonalenia. Ale najpierw wyjaśnijmy, czym jest proces biznesowy, abyś mógł zrozumieć, jak działa system BPM. Proces biznesowy to zbiór czynności lub zadań związanych z uzyskaniem produktu lub usługi, z których Twoja firma ma uzyskać korzyść. Korzyść może być bezpośrednia, jeśli Twoja firma produkuje buty: produkcja buta to proces, projekt buta podlega kolejnemu procesowi, dział zaopatrzenia stosuje własne procesy zakupowe; lub może być pośredni, Twój dział finansowy realizuje proces zamknięcia, dział IT realizuje własne procesy tworzenia użytkownika, tworzenia adresu e-mail lub opracowania ulepszenia BI. Wszystkie firmy, działy i obszary wewnątrz każdej firmy mają swoje własne procesy. Za pomocą narzędzia BPM możesz analizować te procesy, modelować je, dokumentować, organizować iw ten sposób próbować je usprawniać poprzez wykrywanie wąskich gardeł, nieefektywności czy niepożądanych zależności. W tej książce, w konkretnych przykładach w rozdziale 11, mówiącym o transportach, modelowaliśmy już proces, transport tabeli bazy danych. To tylko przykład z działu IT Twojej firmy wewnątrz platformy BI ,system konserwacji. Proces może być również konkatenacją mniejszych procesów. Idąc za tym przykładem, procedura transportowa jest po prostu procesem wewnątrz łańcucha pełnego rozwoju BI, w którym należy uzyskać wymagania użytkowników, przeanalizować je, zamodelować rozwiązanie, utworzyć tabele bazy danych, utworzyć proces ETL, utworzyć interfejs BI raport, przetransportuj całość między środowiskami i zweryfikuj ją. Ideą analizy BPM jest więc rozpoczęcie od procesu wysokiego poziomu, jak pokazano w tym drugim przykładzie, a następnie podzielenie i wprowadzenie szczegółów tego procesu, które zajmują więcej czasu lub są niejasne. Na rynku dostępnych jest wiele programów typu open source BPM, a także bezpłatne wersje komercyjnych. Użyliśmy Bonita soft, ale można znaleźć również Zoho, Processmaker i wiele innych programów, które mogą pomóc we wdrożeniu systemu BPM. Ale jeśli czytasz tę książkę, prawdopodobnie będziesz wdrażać rozwiązanie BI, więc wdrożenie BPM może być jeszcze daleko. To może być sprawa na całą książkę, więc zatrzymajmy się tutaj. Jeśli w przyszłości zdecydujemy się napisać kolejną książkę, to byłaby to dobra kandydatura na ten temat!

Uwaga: uważamy, że oprogramowanie BPM jest bardzo powiązane z oprogramowaniem BI, o ile ma wiele podobieństw. Oba nastawione na analizę, oba nastawione na Business Intelligence w najszerszym znaczeniu tego pojęcia, oba służą poprawie wyników firmy; a podstawowa różnica polega na tym, że BI pomaga zrozumieć dane, a BPM pomaga zrozumieć procesy. Firma korzystająca i

czerpiąca korzyści z systemu BPM jest uważana za bardziej dojrzałą technologicznie niż firma korzystająca tylko z BI; BPM to krok poza zakres dojrzałości technologicznej. BI obsługuje również BPM w zakresie systemu BPM; proponujemy zmiany w sposobie działania firmy, aby po zastosowaniu zmiany można było wykorzystać BI do analizy wyników działań BPM.

Big Data

Wdrożenia BigData już się zaczęły pojawiać i zostały u nas, przynajmniej w dającej się przewidzieć przyszłości. Jeśli nadal nie wiesz, co kryje się za tą fantazyjną nazwą, przedstawimy Ci pierwsze wprowadzenie do BigData. To, co wszyscy rozumiemy na temat BigData, to zestaw narzędzi, które działają razem jako ramy. Narzędzia te pozwalają nam przetwarzać różne rodzaje obciążeń: obciążenie wsadowe, czas rzeczywisty, czas półrzeczywisty oraz różne typy danych: ustrukturyzowane, dane częściowo ustrukturyzowane, dane nieustrukturyzowane, obrazy, filmy. Organizacje zaczęły gromadzić dane, a przetwarzanie i analizowanie tych danych zwykle wykracza poza zakres tradycyjnych narzędzi BI, ponieważ wymaga znacznie większej mocy. Kiedy ta sytuacja jest spełniona, koszty zaczynają stawać się wygórowane, ponieważ skalowanie w pionie jest bardzo drogie. Wyobraź sobie, że masz komputer z 1 TB pamięci RAM. To będzie kosztować strasznie dużo pamięci. Ale z kolei wyobraź sobie 8 komputerów z 128 GB pamięci RAM każdy. Pewnie dużo taniej, prawda? Cóż, może masz dostępne laptopy z petabajtami pamięci RAM, ale w dzisiejszych czasach 1 TB pamięci RAM to duży serwer. W rzeczywistości możliwe jest, że za 10 lat wszystko, co komentujemy w tej sekcji, jest już stare, ale obecnie jest to trend w BI. A więc o to właśnie chodzi w BigData: przetwarzanie ważnych ilości informacji z różnych źródeł i różnych kształtów za pomocą zestawu szerokich narzędzi. Zasadniczo wynika to z 4V: Volume, ponieważ masz dużą ilość danych do przetworzenia; Velocity (Szybkość), ponieważ potrzebujesz przyspieszyć przetwarzanie tych danych przy użyciu wielu maszyn jednocześnie (klaster); Variety (Różnorodność źródeł danych), jak wyjaśniliśmy już w poprzednim rozdziale; oraz Veracity (Prawdziwość), ponieważ przetwarzane dane muszą być przetwarzane tak, aby zapewniały nie tylko znaczącą wartość dla firmy, ale także muszą być godne zaufania, aby osoby podejmujące decyzje w Twojej firmie mogły na nich polegać. Bardziej interesujące z punktu widzenia BI są różne technologie w Hadoop, które mogą pełnić rolę baz danych SQL (lub przynajmniej w pewnym stopniu). Projekt Hive pojawił się jako pierwszy, ale w swojej pierwotnej formie jest powolny. Nie zrozum nas źle, jest to bardzo dobre narzędzie, jeśli przetwarzasz duże zadania wsadowe, ale kiedy musisz rozwiązać zapytania ad hoc, prawdopodobnie z narzędzia do raportowania, czas odpowiedzi jest nie do zaakceptowania. Aby rozwiązać ten problem, niektóre inne silniki, takie jak Spark, który jest jednym z najnowszych i najważniejszych dodatków do stosu Hadoop, oferują krótsze czasy odpowiedzi. Impala, kolejny silnik SQL na szczycie Hadoop, intensywnie wykorzystuje pamięć RAM w maszynach klastrowych, co czyni go bardzo szybkim narzędziem i jednym z najlepiej przystosowanych do rozwiązywania zapytań analitycznych. Jest to więc dobre narzędzie kandydata do użycia w połączeniu z narzędziami do raportowania, a większość z nich już je obsługuje. Z drugiej strony Apache Kudu, projekt, który udaje, że tworzy rozproszoną relacyjną bazę danych na bazie Hadoop, umożliwiając transakcje, modyfikację danych i naśladując wiele tradycyjnych relacyjnych baz danych. Jest to nowe narzędzie, dopiero w fazie beta, ale niektóre organizacje już zaczęły je wdrażać w tym momencie i może bardzo dobrze działać jako chłodnia, na przykład do przechowywania danych historycznych, do których dostęp jest mniejszy, natomiast najnowsza nadal będzie znajdować się w naszej relacyjnej bazie danych. Dzięki takiej strategii możemy zwolnić miejsce w naszej codziennej bazie danych, zachowując nasze stare dane, aby umożliwić nam przeprowadzanie analiz porównawczych, ale na innej maszynie lub silniku. Ale to nie wszystko. Jeśli chcesz eksplorować dane, uczyć się maszynowo, przesyłać strumieniowo informacje, łączyć się z sieciami społecznościowymi i przetwarzać dane w czasie rzeczywistym, masz narzędzia obsługujące te scenariusze w Hadoop. To sprawia, że cały Hadoop jest bardzo interesującą platformą, szerszą w użyciu i bardzo zdolną do wszystkich potrzeb, a jednak w

nadchodzących miesiącach lub latach będzie o wiele więcej. Ponadto trendy cenowe pamięci masowych, a także fakt, że Hadoop może działać na zwykłym sprzęcie, a także spadek cen maszyn, czy to lokalnie, czy w chmurze, obiecują dalsze ulepszenia w tym obszarze. Każdego dnia coraz więcej firm przechowuje znacznie więcej informacji, których prawdopodobnie w tym momencie jeszcze nie zaczęły analizować, ale zaczynają je przechowywać do przyszłych analiz i jest to trend, który nie wykazuje oznak zatrzymania. Cóż, w końcu po tych wszystkich różnych tematach, które widzieliśmy, skończyliśmy - to wszystko, ludzie! Życzymy Ci powodzenia we wdrożeniu BI i mamy nadzieję, że ta lektura Ci się podobała. I mamy nadzieję, że więcej niż przyjemność, masz z tego trochę wiedzy. Mamy nadzieję, że oprócz zdobywania wiedzy, ta wiedza pomoże Ci w skutecznym wdrożeniu systemu BI.

Uwaga: Na stronie internetowej IBM znajduje się dobry zestaw infografik dotyczących niektórych faktów dotyczących 4V: http://www.ibmbigdatahub.com/infographic/four-vs-big-data i tutaj: http://www.ibmbigdatahub .com/infographic/extracting-businessvalue-4-vs-big-data