Krok 9: Wyodrębnij/przekształć/załaduj projekt

Omówiono następujące tematy:

* Kwestie do rozważenia przy projektowaniu wyodrębniania/przekształcania/ładowania (ETL) Typowe strategie wdrażania BI, takie jak hurtownie danych, operacyjne magazyny danych, hurtownie danych przedsiębiorstwa i hurtownie internetowe

* Jak ponownie sformatować, uzgodnić i wyczyścić dane źródłowe dla trzech różnych zestawów programów ETL: ładowanie początkowe, ładowanie historyczne i ładowanie przyrostowe Różne podejścia do wyodrębniania danych z operacyjnych plików źródłowych i źródłowych baz danych

* Typowe problemy z danymi źródłowymi napotykane podczas transformacji, takie jak zduplikowane klucze podstawowe, niespójne wartości danych, różne formaty danych i wbudowana logika procesu

* Zagadnienia dotyczące obciążenia, takie jak integralność referencyjna i indeksowanie

* Dokument mapowania źródła do celu, schemat przebiegu procesu i obszar pomostowy

* Osiem kroków, które należy wykonać podczas oceny produktów i dostawców ETL

Krótkie opisy działań związanych z projektowaniem ETL, rezultatów wynikających z tych działań oraz zaangażowanych ról

* Ryzyko niewykonania kroku 9

Rzeczy do rozważenia

Narzędzia

* Czy wybraliśmy narzędzie ETL, czy piszemy programy ETL od zera?

* Czy narzędzie ETL będzie działać na platformie, na której znajdują się pliki źródłowe? Na oddzielny serwer?

* Czy mamy osobne narzędzie do czyszczenia danych? Czy uruchomimy go przed czy w trakcie procesu ETL?

* Czy mamy wydajne narzędzie sortowania?

Inscenizacja ETL

* Jak duże jest nasze okno etapowe ETL? Ile godzin na dobę? Na tydzień? Czy mamy mniejsze okno na koniec miesiąca z powodu innych procesów na koniec miesiąca? O ile mniejszy?

* Czy możemy dopasować nasz proces ETL do tych okien, czy będziemy musieli przejechać kilka dni lub nocy?

* Ile elementów danych źródłowych musimy wyodrębnić? I jak do wielu plików źródłowych i źródłowych baz danych musimy mieć dostęp?

Przebieg procesu ETL

* Ile programów możemy uruchomić równolegle, aby skrócić czas wykonania ETL?

* Jak długo potrwa początkowe ładowanie? Czy zrobiliśmy prototyp?

* Ile czasu zajmie wczytanie danych historycznych? Ile uszu historii musimy załadować?

* Czy wiemy, jak długo będą działać obciążenia przyrostowe?

* Czy powinniśmy wstawiać wiersze, czy korzystać z systemu zarządzania bazą danych?

(DBMS) narzędzie do ładowania?

* Czy powinniśmy użyć narzędzia do ładowania innej firmy, aby przyspieszyć ten proces? Czy mamy już narzędzie do ładowania innej firmy, czy musimy je kupić?

* Kiedy i jak będą archiwizowane dane? Na dysku? Na taśmie? Czy musimy w tym czasie napisać programy archiwalne, czy możemy to odłożyć dodatkowy wysiłek programistyczny w stosunku do przyszłej wersji?

Rozważania dotyczące wydajności

* Jak wpłynie na wydajność obciążenia ETL, jeśli pozostawimy referencję integralność (RI) włączoną?

* Jak wysokie byłoby ryzyko uszkodzenia danych, gdybyśmy wyłączyli RI? Ile sprawdzania RI chcemy wykonać w programach ETL?

Pojednanie

* W ilu punktach procesu ETL musimy liczyć rekordy wejściowe i wyjściowe?

* Czy układy rekordów i struktury baz danych są inne w starym pliku historycznym i bazie danych niż w plikach bieżących i bazy danych? Jak je pogodzić?

* Czy musimy uzgodnić zmienione kody? Ponownie wykorzystany i przedefiniowane pola?

* Ile elementów danych musimy uzgodnić? Ile kodów? Ile kwot?

* Czy brudne dane zostaną odrzucone? Jak zostanie to odzwierciedlone w sumach uzgodnień?

* Czy liczniki obciążeń i sumy uzgodnień będą przechowywane jako metadane?

Wskaźniki jakości

* Jak będą liczone błędy jakości danych? Jakie wskaźniki jakości danych musimy skompilować w programach?

* Czy będziemy przechowywać te metryki jako metadane w repozytorium metadanych, czy wydrukować je w raporcie?

Dane źródłowe dla aplikacji BI będą pochodzić z różnych platform, którymi zarządzają różne systemy operacyjne i aplikacje. Celem procesu ETL jest połączenie danych z tych heterogenicznych platform w standardowy format dla docelowych baz danych BI w środowisku wspomagania decyzji BI.

Strategie wdrażania

Istnieje kilka rodzajów strategii wdrażania wspierających decyzje BI z każdą możliwą kombinacją docelowych baz danych BI (np. operacyjne magazyny danych i hurtownie danych przedsiębiorstwa; hurtownia internetowa i zbiorcze bazy danych; hurtownie eksploracyjne i bazy danych do eksploracji danych; zbiorcze zbiorcze dane i operacyjne targi]). Zdecydowanie najpopularniejszą strategią wdrożeniową jest środowisko data mart. Bez względu na wybraną strategię wdrożenia, istnieje właściwy i zły sposób jej realizacji. Niewłaściwym sposobem jest zbudowanie zbioru autonomicznych docelowych baz danych BI, z których każda posiada własny niezależny proces ETL. Takie podejście nie zapewni zintegrowanego i uzgodnionego środowiska wspomagania decyzji BI, ponieważ tworzenie

oddzielnych procesów ETL nie różni się niczym od tworzenia tradycyjnych systemów wspomagania decyzji. Właściwym sposobem wdrożenia wybranej strategii jest zbudowanie środowiska wspierającego podejmowanie decyzji BI, w którym wszystkie docelowe bazy danych BI są zintegrowane i uzgadniane. Podczas tworzenia tego środowiska bardzo ważne jest, aby przeprowadzić wspólne przekształcenia danych dla wszystkich docelowych baz danych BI tylko raz i uzgodnić te przekształcenia z powrotem z operacyjnymi plikami źródłowymi i źródłowymi bazami danych. To zademonstruje ważność danych w różnych docelowych bazach danych BI. Ważne jest również uzgodnienie wszystkich danych w różnych docelowych bazach danych BI w celu wykazania spójności danych między różnymi docelowymi bazami danych BI. Oba procesy uzgadniania najlepiej można osiągnąć przy skoordynowanym wysiłku ETL dla wszystkich docelowych baz danych BI.

Najważniejszą zasadą ETL dla zintegrowanej strategii wdrażania BI jest współdzielenie jednego skoordynowanego procesu ETL. To właśnie odróżnia BI od tradycyjnego podejścia wspomagającego podejmowanie decyzji.

Przygotowanie do procesu ETL

Proces ETL rozpoczyna się od przygotowań do ponownego formatowania, uzgadniania i czyszczenia danych źródłowych.

Ponowne formatowanie: Dane źródłowe znajdujące się w różnych plikach źródłowych i źródłowych bazach danych, każda z własnym formatem, będą musiały zostać ujednolicone we wspólnym formacie podczas procesu ETL.

Uzgadnianie: Ogromna ilość danych w organizacjach wskazuje na oszałamiającą nadmiarowość, która niezmiennie prowadzi do oszałamiających niespójności. Należy je znaleźć i uzgodnić podczas procesu ETL.

Czyszczenie: Brudne dane znalezione podczas analizy danych i prototypowania będą musiały zostać oczyszczone podczas tego procesu.

Przed zaprojektowaniem procesu ETL należy zapoznać się z następującymi kwestiami:

* Nagrywaj układy bieżących i historycznych plików źródłowych

* Bloki opisu danych dla aktualnych i historycznych baz danych źródłowych

* Specyfikacje czyszczenia danych dla źródłowych elementów danych

Większość danych źródłowych dla procesu ETL to bieżące dane operacyjne z systemów operacyjnych, ale część danych źródłowych może być archiwalnymi danymi historycznymi.

Zbiór programów ETL

Początkowe obciążenie: początkowa populacja docelowych baz danych BI z bieżącymi danymi operacyjnymi (1)

Obciążenie historyczne: początkowa populacja docelowych baz danych BI z zarchiwizowanymi danymi historycznymi (2)

Obciążenie przyrostowe: bieżąca populacja docelowych baz danych BI z bieżącymi danymi operacyjnymi (3)

Jeśli wymagania dotyczące danych obejmują kilka lat historii, które należy uzupełnić od początku, należy zaprojektować i opracować trzy zestawy programów ETL. Jeśli zostanie podjęta decyzja o

napisaniu programów ETL w języku proceduralnym (np. C++ lub COBOL), należy przygotować i przekazać programistom ETL specyfikacje transformacji dla trzech zestawów programów. Jeśli będzie używane narzędzie ETL, instrukcje ETL (metadane techniczne) muszą zostać utworzone dla trzech zestawów procesów ładowania. Metadane techniczne ETL będą odzwierciedlać tę samą logikę, która zostałaby napisana w niestandardowych programach, gdyby nie było dostępne żadne narzędzie ETL. Metadane techniczne powinny być przechowywane w repozytorium metadanych.

Początkowe obciążenie

Proces przygotowania programów ładowania początkowego jest bardzo podobny do procesu konwersji systemu, na przykład tego, który wiele organizacji przeprowadza, przenosząc swoje stare systemy operacyjne do produktu do planowania zasobów przedsiębiorstwa (ERP). Ogólnie rzecz biorąc, pierwszym zadaniem procesu konwersji systemu jest odwzorowanie wybranych elementów danych z plików źródłowych lub źródłowych baz danych na najbardziej odpowiednie elementy danych w plikach docelowych lub docelowych bazach danych. „Najbardziej odpowiedni element danych" w docelowym pliku lub docelowej bazie danych to taki, który jest najbardziej podobny pod względem nazwy, definicji, rozmiaru, długości i funkcjonalności do źródłowego elementu danych. Drugim zadaniem procesu konwersji systemu jest napisanie programów konwersji (transformacji) w celu przekształcenia danych źródłowych. Te programy do konwersji muszą również rozwiązywać zduplikowane rekordy, dopasowywać klucze podstawowe i skracać lub powiększać rozmiar danych elementów. Zazwyczaj brakujące w programach do konwersji, a niestety również brak w większości procesów ETL, to czyszczenie i uzgadnianie danych. Organizacje wielokrotnie tracą najlepsze okazje do uporządkowania chaosu danych, gdy nadal „wysysają i przesadzają" dane ze źródła do celu, bez zmian. Ich jedyną troską jest to, że struktura odbierającej bazy danych nie odrzuca danych źródłowych z przyczyn technicznych, takich jak zduplikowane klucze lub naruszenia typu i długości danych. To za mało dla aplikacji BI, ponieważ ludzie biznesu oczekują jakości i spójności danych ze względów biznesowych. Dlatego podczas projektowania procesów ładowania czyszczenie i uzgadnianie danych musi stać się częścią przepływu procesu ETL.

Ładunek historyczny

Historyczny proces wczytywania można traktować jako rozszerzenie wstępnego procesu wczytywania, ale ten typ konwersji jest nieco inny, ponieważ dane historyczne są danymi statycznymi. W przeciwieństwie do danych operacyjnych na żywo, dane statyczne spełniły swoje zadanie operacyjne i zostały zarchiwizowane na urządzeniach pamięci masowej offline. Oznacza to, że ponieważ niektóre stare dane wygasają, a niektóre nowe są dodawane na przestrzeni lat, układy rekordów zarchiwizowanych plików zwykle nie są zsynchronizowane z układami rekordów bieżących plików operacyjnych. Dlatego programy do konwersji napisane dla bieżących plików operacyjnych zwykle nie mogą być stosowane do zarchiwizowanych plików historycznych bez pewnych zmian. Na przykład w często zmieniającym się systemie operacyjnym nierzadko zdarza się, że zarchiwizowane akta historyczne przez pięć lat mają pięć (lub więcej) nieco różniących się układami zapisów. Choć różnice w układach płyt mogą nie być drastyczne, to i tak trzeba je pogodzić. Ponadto czystość danych może nie być taka sama we wszystkich zarchiwizowanych plikach. To, co kiedyś było ważne w pliku historycznym, może już nie być ważne. Specyfikacje transformacji danych muszą uwzględniać te różnice i je pogodzić. Wszystkie te czynniki przyczyniają się do tego, że proces ETL może być bardzo długi i bardzo skomplikowany.

Obciążenie przyrostowe

Po opracowaniu procesów zapełniania docelowych baz danych BI danymi początkowymi i historycznymi należy zaprojektować inny proces dla bieżącego obciążenia przyrostowego (co miesiąc,

co tydzień lub codziennie). Obciążenia przyrostowe można osiągnąć na dwa sposoby, wyodrębniając tylko wszystkie rekordy lub delty, jak pokazano w tabeli. Projekt procesu ekstrakcji ETL będzie się różnić w zależności od wybranej opcji.

Wyodrębnij wszystkie rekordy: Wyodrębnij dane źródłowe ze wszystkich operacyjnych rekordów, niezależnie od tego, czy jakiekolwiek wartości danych zmieniły się od ostatniego załadowania ETL, czy nie.

Wyodrębnij tylko różnice : Wyodrębnij dane źródłowe tylko z tych rekordów operacyjnych, w których niektóre wartości danych uległy zmianie od ostatniego obciążenia ETL („zmiana netto").

Wyodrębnienie wszystkich rekordów często nie jest realną opcją ze względu na ogromne ilości danych. Dlatego wiele organizacji wybiera ekstrakty delta (wyodrębnianie tylko rekordów, które uległy zmianie). Projektowanie programów ETL do ekstrakcji delta jest znacznie łatwiejsze, gdy dane źródłowe znajdują się w relacyjnych bazach danych, a znacznik czasu może być użyty do określenia delt. Ale gdy dane są przechowywane w płaskich plikach bez znacznika czasu, proces wyodrębniania może być znacznie bardziej złożony. Być może będziesz musiał odwołać się do odczytania ścieżek audytu operacyjnego, aby określić, które rekordy uległy zmianie. Alternatywą może być wyodrębnienie pełnej kopii pliku źródłowego dla każdego obciążenia, a następnie porównanie nowego ekstraktu z poprzednim, aby znaleźć zmienione rekordy i utworzyć własny plik delta. Inną alternatywą jest poproszenie pracowników systemów operacyjnych o dodanie znacznika czasu systemu do ich plików operacyjnych. Czasami mogą się na to zgodzić, jeśli zmiana ich systemów operacyjnych jest trywialna i nie dotyczy wielu programów. Jednak w większości przypadków menedżerowie operacji nie zgadzają się na to, ponieważ wszelkie zmiany w ich strukturach plików wymagałyby również zmian w programach do wprowadzania danych i aktualizacji. Musiałby zostać napisany dodatkowy kod, aby te programy mogły przechwytywać systemowy znacznik czasu. Zmiana systemów operacyjnych o znaczeniu krytycznym i spędzanie dużej ilości czasu na testowaniu regresji nie byłaby dla nich opłacalna - tylko z korzyścią dla aplikacji BI.

Przetwarzanie usuniętych rekordów

Innym aspektem, który należy dokładnie rozważyć w przypadku obciążeń przyrostowych, jest kwestia usuniętych operacyjnych rekordów źródłowych. Gdy pewne rekordy są logicznie usuwane z plików źródłowych i źródłowych baz danych (oznaczone jako usunięte, ale nie usunięte fizycznie), odpowiednie wiersze nie mogą zostać automatycznie usunięte z docelowych baz danych BI. W końcu jednym z głównych wymagań docelowych baz danych BI jest przechowywanie danych historycznych. Proces ETL musi być zgodny z zestawem reguł biznesowych, które powinny określać, kiedy usunięcie operacyjne powinno być propagowane do docelowych baz danych BI, a kiedy nie. Na przykład, być może usuwany jest rekord operacyjny, ponieważ został wcześniej utworzony przez pomyłkę, rekord jest archiwizowany lub system operacyjny przechowuje tylko transakcje „otwarte", a usuwa te „zamknięte". Najprawdopodobniej reguły biznesowe stwierdzają, że należy usunąć powiązany wiersz z docelowej bazy danych BI tylko w przypadku, gdy rekord został utworzony przez pomyłkę. Ponieważ docelowa baza danych BI przechowuje dane historyczne, reguły biznesowe prawdopodobnie nie pozwolą na usunięcie powiązanego wiersza w pozostałych dwóch wystąpieniach. Kiedy rekordy są fizycznie usuwane z plików źródłowych lub źródłowych baz danych, nigdy byś tego nie wiedział, jeśli wyodrębniasz tylko delty. Programy do ekstrakcji delta są przeznaczone do wyodrębniania tylko tych istniejących rekordów, w których zmieniła się jedna z wartości danych; nie mogą wyodrębnić rekordów, które nie istnieją. Jednym ze sposobów na znalezienie fizycznie usuniętych rekordów jest odczytanie operacyjnych ścieżek audytu. Inną opcją jest wyodrębnienie pełnej kopii pliku źródłowego, porównanie nowego ekstraktu z poprzednim w celu znalezienia usuniętych rekordów, a następnie

utworzenie własnych plików delta. W obu przypadkach, po zidentyfikowaniu usuniętych rekordów, proces ETL musi postępować zgodnie z zestawem reguł biznesowych, aby zdecydować, czy fizycznie usunąć powiązane wiersze z docelowych baz danych BI.

Projektowanie programów ekstrakcji

Z perspektywy systemów operacyjnych najbardziej preferowanym sposobem tworzenia fragmentów może być po prostu powielenie całej zawartości operacyjnych plików źródłowych i źródłowych baz danych oraz przekazanie duplikatów zespołowi projektu BI. Jednak programiści ETL mieliby ciężar pracy z ogromnymi plikami, gdy potrzebują tylko podzbioru danych źródłowych. Z perspektywy projektu BI najbardziej preferowanym sposobem tworzenia wyciągów może być sortowanie, filtrowanie, czyszczenie i agregowanie wszystkich wymaganych danych w jednym kroku, jeśli to możliwe, i robienie tego bezpośrednio u źródła. Jednak w niektórych organizacjach, które wpłynęłyby na systemy operacyjne do tego stopnia, że operacyjne funkcje biznesowe musiałyby zostać zawieszone na kilka godzin. Rozwiązanie jest zazwyczaj kompromisem: programy do ekstrakcji są zaprojektowane z myślą o jak najwydajniejszym przetwarzaniu ETL, ale zawsze z naciskiem na jak najszybsze uzyskanie wymaganych danych źródłowych. Celem jest zejście z drogi systemom operacyjnym tak, aby codzienne funkcje biznesowe nie zostały naruszone. Z wielu powodów łatwiej to powiedzieć niż zrobić. Wybieranie i łączenie danych z plików źródłowych i źródłowych baz danych może stanowić wyzwanie ze względu na dużą nadmiarowość danych w systemach operacyjnych. Programy wyodrębniające muszą wiedzieć, które z nadmiarowych plików źródłowych lub źródłowych baz danych są systemami zapisu. Na przykład ten sam element danych źródłowych (np. Nazwa klienta) może istnieć w dziesiątkach plików źródłowych i źródłowych baz danych. Te nadmiarowe wystąpienia muszą zostać posortowane i skonsolidowane, co obejmuje szereg kroków sortowania i scalania, sterowanych przez szereg tabel przeglądowych zawierających odniesienia do określonych kluczy i wartości danych. Innym sposobem tworzenia małych i stosunkowo czystych plików wyodrębniania jest wyodrębnianie tylko tych elementów danych źródłowych, które są potrzebne aplikacji BI i rozwiązywanie tylko tych problemów z jakością danych źródłowych, które dotyczą reguł domeny danych biznesowych, bez próby uporządkowania i konsolidacji nadmiarowe wystąpienia danych. Jednak nawet ten kompromis nie zadziała w wielu dużych organizacjach, ponieważ proces czyszczenia danych spowolniłby proces wyodrębniania, co z kolei spowodowałoby zablokowanie systemów operacyjnych dłużej, niż jest to dopuszczalne. W wielu dużych organizacjach zespół projektowy BI ma szczęście uzyskać od trzech do czterech godzin czasu przetwarzania w systemach operacyjnych, zanim te systemy operacyjne będą musiały „uruchomić" funkcje operacyjne następnego dnia roboczego. Jest to główny powód, dla którego zapełnianie docelowych baz danych BI jest podzielone na trzy oddzielne procesy: wyodrębnianie, przekształcanie i ładowanie.

Projektowanie programów transformacji

Stosując regułę 80/20, 80 procent pracy ETL odbywa się w części „T" (transformacja), gdy wymagana jest rozległa integracja danych i czyszczenie danych, podczas gdy wyodrębnianie i ładowanie stanowi tylko 20 procent procesu ETL.

Problemy z danymi źródłowymi

Projektowanie programów transformacji może stać się bardzo skomplikowane, gdy dane są pobierane z heterogenicznego środowiska operacyjnego. Poniżej opisano niektóre typowe problemy z danymi źródłowymi.

* Niespójne klucze podstawowe: klucze podstawowe rekordów danych źródłowych nie zawsze pasują do nowego klucza podstawowego w tabelach BI. Na przykład może istnieć pięć plików klientów, każdy

z innym kluczem klienta. Te różne klucze klienta zostałyby skonsolidowane lub przekształcone w jeden ustandaryzowany klucz klienta BI. Klucz klienta BI byłby prawdopodobnie nowym kluczem zastępczym („wymyślonym”) i nie pasowałby do żadnego z kluczy operacyjnych.

* Niespójne wartości danych: wiele organizacji powiela wiele swoich danych. Termin duplikat zwykle oznacza, że element danych jest dokładną kopią oryginału. Jednak z biegiem czasu te duplikaty kończą z zupełnie innymi wartościami danych z powodu anomalii aktualizacji (niespójne aktualizacje zastosowane do duplikatów), które muszą zostać uzgodnione w procesie ETL.

* Różne formaty danych: Elementy danych, takie jak daty i waluty, mogą być przechowywane w zupełnie innym formacie w plikach źródłowych niż będą przechowywane w docelowych bazach danych BI. Jeżeli moduły przeliczania dat i walut już istnieją, należy je zidentyfikować; w przeciwnym razie należy opracować logikę tej transformacji.

* Niedokładne wartości danych: Aby skorygować niedokładne wartości danych, należy zdefiniować logikę czyszczenia. Niektóre z logiki czyszczenia danych mogą być bardzo skomplikowane i długotrwałe. Korekta jednego naruszenia danych może zająć kilka stron instrukcji czyszczenia. Czyszczenie danych nie odbywa się jednorazowo – jest to proces ciągły. Ponieważ nowe dane są ładowane do docelowych baz danych BI w każdym cyklu ładowania, algorytmy czyszczenia danych ETL muszą być uruchamiane za każdym razem, gdy dane są ładowane. Dlatego programów transformacji nie można pisać „szybko i brudno”. Zamiast tego muszą być zaprojektowane w przemyślany i dobrze zorganizowany sposób.

* Synonimy i homonimy: nadmiarowe dane nie zawsze są łatwe do rozpoznania, ponieważ ten sam element danych może mieć różne nazwy. Systemy operacyjne są również znane z używania tej samej nazwy dla różnych elementów danych. Ponieważ synonimy i homonimy nie powinny istnieć w środowisku wspomagania decyzji BI, zmiana nazw elementów danych dla docelowych baz danych BI jest częstym zjawiskiem.

* Wbudowana logika procesu: Niektóre systemy operacyjne są bardzo stare. Biegają, ale często nikt nie wie jak! Często zawierają nieudokumentowane i archaiczne relacje między niektórymi elementami danych źródłowych. Istnieje również bardzo duża szansa, że niektóre kody w systemach operacyjnych są wykorzystywane jako przełączniki szyfrujące. Na przykład wartość "00" w elemencie danych Alter-Flag może oznaczać, że przesyłka została zwrócona, a wartość "FF" w tym samym elemencie danych może oznaczać, że chodziło o przebieg na koniec miesiąca. Specyfikacje transformacji musiałyby odzwierciedlać tę logikę.

Transformacje danych

Oprócz przekształcania danych źródłowych ze względu na niezgodny typ i długość danych lub niespójne i niedokładne dane, duża część logiki transformacji będzie obejmować wstępne obliczanie danych do przechowywania wielowymiarowego. Dlatego nie powinno dziwić, że dane w docelowych bazach danych BI będą wyglądały zupełnie inaczej niż dane w systemach operacyjnych. Kilka konkretnych przykładów znajduje się poniżej.

* Niektóre dane zostaną zmienione zgodnie ze standardami nazewnictwa BI (synonimy i homonimy nie powinny być propagowane do środowiska wspomagania decyzji BI). Na przykład element danych Flaga konta może teraz nosić nazwę Product_Type_Code.

* Niektóre elementy danych z różnych systemów operacyjnych zostaną połączone (scalone) w jedną kolumnę w tabeli BI, ponieważ reprezentują ten sam logiczny element danych. Na przykład Cust-Name z pliku CMAST, Customer_Nm z tabeli CRM_CUST i Cust_Acct_Nm z tabeli CACCT można teraz scalić z kolumną Customer_Name w tabeli BI_CUSTOMER.

* Niektóre elementy danych zostaną podzielone na różne kolumny w docelowej bazie danych BI, ponieważ są używane przez systemy operacyjne do wielu celów. Na przykład wartości „A", „B", „C", „L", „M", „N", „X", „Y" i „Z" źródłowego elementu danych Prod-Code mogą być używane przez system operacyjny w następujący sposób: „A", „B" i „C" opisują klientów; „L", „M" i „N" opisują dostawców; a „X", „Y" i „Z" opisują ograniczenia regionalne. W rezultacie kod produktu można teraz podzielić na trzy kolumny:

- Customer_Type_Code w tabeli BI_CUSTOMER

- Supplier_Type_Code w tabeli BI_SUPPLIER

- Regional_Constraint_Code w tabeli BI_ORG_UNIT

* Niektóre elementy danych kodu zostaną przetłumaczone na mnemoniki lub zostaną zapisane. Na przykład:

- „A" można przetłumaczyć na „Korporacja"

- „B" można przetłumaczyć na „Partnerstwo"

- „C" można przetłumaczyć na „Indywidualny"

* Ponadto większość danych będzie agregowana i podsumowywana na podstawie wymaganych wzorców raportowania oraz na podstawie wybranej wielowymiarowej struktury bazy danych (schemat gwiazdy, płatek śniegu). Na przykład na koniec miesiąca można zsumować (zagregowane) elementy danych źródłowych: Saldo-kredytu-hipoteki, Saldo-kredytu-budowlanego i Kwota-kredytu-konsumenta podsumowane według regionu w kolumnie Monthly_Regional_Portfolio_Amount w tabeli faktów BI_PORTFOLIO.

Projektowanie programów ładowania

Ostatnim krokiem w procesie ETL jest załadowanie docelowych baz danych BI, co można zrealizować na dwa sposoby: (1) przez wstawienie nowych wierszy do tabel lub (2) za pomocą narzędzia ładowania DBMS w celu wykonania ładowania zbiorczego . Dużo bardziej efektywne jest korzystanie z narzędzia do ładowania systemu DBMS i większość organizacji wybiera takie podejście. Po zakończeniu etapów wyodrębniania i transformacji nie powinno być zbyt skomplikowane, aby zakończyć proces ETL z etapem ładowania. Jednak nadal konieczne jest podejmowanie decyzji projektowych dotyczących integralności referencyjnej i indeksowania.

Więzy integralności

Ze względu na ogromne ilości danych wiele organizacji woli wyłączyć RI, aby przyspieszyć proces ładowania. Jednak w takim przypadku programy ETL muszą wykonać niezbędne kontrole RI; w przeciwnym razie docelowe bazy danych BI mogą ulec uszkodzeniu w ciągu kilku miesięcy lub nawet tygodni. Działanie zgodnie z ideą, że sprawdzanie RI nie jest potrzebne w aplikacjach BI (ponieważ nie są tworzone żadne nowe relacje danych i ładowane są tylko istniejące dane operacyjne) nie zapobiega uszkodzeniu bazy danych! Często dochodzi do uszkodzenia docelowych baz danych BI, głównie dlatego, że dane operacyjne często nie są właściwie powiązane w pierwszej kolejności, zwłaszcza gdy dane operacyjne nie znajdują się w relacyjnej bazie danych. Nawet jeśli dane operacyjne pochodzą z relacyjnej bazy danych, nie ma gwarancji, że RI zostanie odpowiednio wymuszony, ponieważ zbyt wiele projektów relacyjnych baz danych jest niczym więcej niż niepowiązanymi płaskimi plikami w tabelach.

Gdy RI jest wyłączone podczas procesu ładowania ETL (tak jak powinno, ze względu na wydajność), zaleca się jego ponowne włączenie po zakończeniu procesu ładowania, aby umożliwić systemowi DBMS określenie wszelkich naruszeń RI między zależnymi tabelami.

Indeksowanie

Bazy danych o niskiej wydajności są często wynikiem słabo działających schematów indeksowania. Ze względu na dużą ilość danych w docelowych bazach danych BI konieczne jest posiadanie sprawnie działających indeksów i posiadanie ich wielu. Jednak budowanie wpisów indeksu podczas ładowania tabel BI spowalnia proces ładowania ETL. Dlatego wskazane jest usunięcie wszystkich indeksów przed procesem ładowania ETL, załadowanie docelowych baz danych BI, a następnie odtworzenie indeksów po zakończeniu procesu ładowania ETL i sprawdzeniu RI.

Projektowanie przebiegu procesu ETL

Dokument mapowania źródła do celu

Zanim będzie można zaprojektować (lub ulepszyć) przepływ procesu ETL, należy opracować szczegółowe specyfikacje dotyczące ekstrakcji danych, transformacji i uzgadniania transformacji ETL, biorąc pod uwagę, że będą one dyktować przepływ. Typowym sposobem dokumentowania specyfikacji transformacji ETL jest mapowanie źródła do celu dokumentu, którym może być macierz lub arkusz kalkulacyjny. Dokument mapowania źródło-cel powinien zawierać listę wszystkich tabel i kolumn BI oraz ich typy danych i powinien mapować odpowiednie elementy danych źródłowych do kolumn, wraz z ich typami danych i długościami, powinien pokazywać pliki źródłowe i źródłowe bazy danych, z których wyodrębniane są elementy danych źródłowych. Wreszcie, co najważniejsze, dokument powinien określać logikę transformacji dla każdej kolumny. Dokument ten może być następnie wykorzystany do stworzenia rzeczywistych specyfikacji programowania dla programistów ETL tworzących instrukcje (metadane techniczne) dla narzędzia ETL.

Schemat przebiegu procesu ETL

Po ukończeniu dokumentu mapowania źródło-cel główny programista ETL wraz z administratorem bazy danych i analitykiem jakości danych muszą zaprojektować przepływ procesu ETL. Celem diagramu przepływu procesu ETL jest pokazanie zależności procesów między wszystkimi wyodrębnieniami i scalaniem narzędzi, przekształceń, tymczasowych plików roboczych lub tabel, procesów obsługi błędów, działań uzgadniania i sekwencji ładowania. Programy ETL lub moduły narzędzi ETL będą musiały działać w tej kolejności.

* Wyciągi: Mogą istnieć operacyjne współzależności między kilkoma plikami źródłowymi a źródłem, z którego wyodrębniane są dane. Te współzależności należy zrozumieć, ponieważ określają one czas i kolejność uruchamiania programów wyodrębniania ETL.

* Narzędzia do sortowania i łączenia: Prawie każdy krok wymaga posortowania wyodrębnionych danych w taki sposób, aby można je było połączyć przed dalszym przetwarzaniem. Sortowanie może również znacznie poprawić wydajność.

* Transformacje: Większość danych musi zostać przekształcona z różnych powodów. Ważne jest, aby przeanalizować najbardziej dogodne momenty na wykonanie transformacji. Pamiętaj, że istnieje tylko skoordynowany proces ETL dla środowiska wspomagania decyzji BI. Dlatego przekształcenia mające zastosowanie do wszystkich danych źródłowych, takie jak konwersje typu danych i kodu lub wymuszania domeny danych, powinny być na wczesnym etapie przepływu procesu ETL. Transformacje

specyficzne dla docelowej bazy danych, takie jak podsumowanie agregacji dla określonej hurtowni danych, powinny nastąpić pod koniec przepływu procesu ETL.

* Tymczasowe pliki robocze lub tabele: sortowanie, scalanie i przekształcanie wymaga dużo tymczasowej przestrzeni do przechowywania wyników pośrednich. Te tymczasowe pliki robocze i tabele mogą być dużymi lub większymi oryginalnymi fragmentami. Co więcej, te tymczasowe pliki robocze i tabele tak naprawdę nie są „tymczasowe". miej tę przestrzeń na stałe dostępną dla swojej pomostu.

* Procesy obsługi błędów: podczas procesu ETL wykrywanych jest wiele błędów, ponieważ specyfikacje czyszczenia danych są stosowane do danych źródłowych. W przypadku utworzenia raportów o błędach lub błędnych zapisów w pliku przejściowym, należy je pokazać na diagramie przebiegu procesu ETL.

* Działania uzgadniania: każdy moduł programu, który manipuluje danymi, powinien generować sumy uzgadniania. Może to mieć postać liczby rekordów wejściowych i wyjściowych, liczby określonych domen lub ilości Liczba rekordów jest wystarczająca dla modułów wyodrębniania, sortowania i scalania. Liczby domen są odpowiednio skomplikowanymi specyfikacjami transformacji, takimi jak oddzielanie wartości danych od jednego źródła danych wielu kolumn docelowych. Liczenie kwot jest zwykle wykonywane na wszystkich elementach danych dotyczących kwoty, niezależnie od tego, czy są przenoszone bez zmian, przekształcane do nowego formatu, czy używane w obliczeniach.

* Kolejność ładowania: konieczne jest określenie kolejności, w jakiej tabele mają być ładowane ich potencjalne współzależności oraz ze względu na możliwą relację rekurencyjną na jednej tabeli. tabela wymiarów produktu może wymagać załadowania przed załadowaniem tabeli Sales, jeśli RI jest włączony, dane sprzedaży odwołują się do produktów. Inne tabele mogą być ładowane jednocześnie, co może znacznie przyspieszyć proces ładowania.

Obszar inscenizacji

Obszar pomostowy to miejsce, w którym przebiega proces ETL. Dotyczy to dedykowanej przestrzeni dyskowej, tymczasowych i stałych plików roboczych programu ETL oraz tabel - nawet dedykowanego serwera. Obszar postoju może być scentralizowany i zdecentralizowany. Na przykład może to być centralny obszar pomostowy na komputerze mainframe, jeśli większość danych źródłowych to komputer mainframe. Może również znajdować się na dedykowanym serwerze, na który ładowane są dane źródłowe. Często obszar jest zdecentralizowany. Na przykład zawiły plik mainframe z wieloma przedefiniowanymi klauzulami i wystąpieniami musi zostać spłaszczony za pomocą programu w języku COBOL w obszarze pomostowym na komputerze mainframe, zanim zostanie pobrany z obszaru pomostowego na komputer z systemem UNIX w celu dalszego przetwarzania przez narzędzie ETL. Proces ETL jest zdecydowanie najbardziej skomplikowanym procesem, jaki należy zaprojektować i opracować w każdym projekcie BI. istnieje tylko jeden (logicznie) skoordynowany proces ETL dla środowiska wspomagania decyzji BI, rozszerzanie programów o każdą nową aplikację BI staje się bardzo skomplikowane, a testowanie regresji wymaga więcej czasu. Z tych powodów większość organizacji woli używać narzędzia ETL do wszystkich lub niektórych procesów ETL, zwłaszcza do procesów ekstrakcji i transformacji

Ocena narzędzi ETL

Podczas korzystania z narzędzia ETL specyfikacje transformacji są tłumaczone na instrukcje dla narzędzia ETL. Instrukcje te mogą być następnie przechowywane jako metadane techniczne w repozytorium metadanych. Narzędzie ułatwia rozszerzanie procesu ETL i przeprowadzanie testów

regresji, ponieważ jest mniej interwencji człowieka, a tym samym mniejsze szanse na wprowadzenie błędów.

Podczas oceny produktów ETL postępuj zgodnie z poniższymi krokami.

1. Przeprowadź analizę kosztów i korzyści, aby porównać licencjonowanie (zakup) produktu ETL z tworzeniem procesu ETL we własnym zakresie. Chociaż obie opcje mogą być drogie z różnych powodów (licencjonowanie bardzo wyrafinowanego narzędzia ETL jest drogie, ale także utrzymywanie niestandardowego oprogramowania), pierwszym wyborem powinno być licencjonowanie narzędzia ETL. Jeśli narzędzie ETL nie może obsłużyć wszystkich wymaganych przekształceń, uzupełnij licencjonowany produkt własnym, wyspecjalizowanym kodem dla tych przekształceń. Jeśli Twoje wymagania dotyczące transformacji są proste i masz niewielki budżet, możesz chcieć kupić mniej zaawansowane narzędzie ETL lub rozważyć zbudowanie własnego procesu ETL.

2. Sporządź listę produktów i dostawców ETL, którzy mogą spełnić Twoje wymagania. Weź udział w targach, aby dowiedzieć się więcej o produktach i dostawcach. Caveat emptor — bądź ostrożnym i sceptycznym nabywcą.

Niech Twoje wymagania dotyczące transformacji i oczyszczania — a nie szum wśród dostawców — napędzają proces oceny i wyboru produktów.

3. Porównaj produkty i dostawców ETL z wymaganiami dotyczącymi ważonej transformacji danych. Uwzględnij reguły biznesowe dotyczące czyszczenia danych jako część kryteriów wyboru narzędzia ETL. Na przykład niektóre narzędzia ETL nie mogą czytać plików płaskich i nie mogą wykonywać niektórych bardzo skomplikowanych przekształceń. Jeśli potrzebujesz tych możliwości, musisz zdawać sobie sprawę z ograniczeń narzędzia ETL, ponieważ może być konieczne wykupienie licencji na dodatkowe narzędzie do czyszczenia danych w celu wykonania tych procesów lub rozszerzenie funkcjonalności narzędzia ETL o niestandardowy kod.

4. Oceń każdy produkt ETL obiektywnie i przygotuj kartę scoringową porównującą cechy produktów i ich skuteczność. Reputacja i szybkość reakcji dostawcy są równie ważne jak cechy produktów. Dlatego przygotuj kolejną kartę wyników porównującą dostawców.

5. Sprawdź referencje klientów dostawców, rozmawiając z osobami z organizacji, które już używają rozważanych narzędzi. Jest to najbardziej opłacalny i informacyjny sposób oceny narzędzi.

6. Zawęź listę produktów i dostawców ETL do krótkiej listy dwóch lub trzech kandydatów. W przeciwnym razie zmarnujesz zbyt dużo czasu na porównywanie wszystkich produktów, a podjęcie ostatecznej decyzji może zająć „wieczność”. Możesz wtedy wybrać gorsze narzędzie, aby zakończyć frustrację i opóźnienie.

7. Umów się na prezentacje produktów, ponieważ „widzieć znaczy wierzyć”. Poświęć trochę czasu na przygotowanie przypadków testowych, aby wszyscy dostawcy z krótkiej listy mogli zademonstrować wydajność i skuteczność swoich produktów przy użyciu tych samych przypadków testowych.

8. Przetestuj produkty dostawców, nawet jeśli zabiera to czas z harmonogramu projektu. Testowanie to najlepszy sposób na wykrycie wszelkich usterek, które mogą wystąpić przed użyciem produktu dostawcy w produkcji. Spróbuj wynegocjować 30-dniowy okres próbny.

Działania projektowe ETL

Czynności związane z projektowaniem ETL nie muszą być wykonywane liniowo. Poniższa lista krótko opisuje czynności związane z Krokiem 9, Projektowanie ETL.

1. Utwórz dokument mapowania źródło-cel. Wykorzystaj wyniki analizy danych źródłowych i reguły biznesowe z poprzednich kroków i uwzględnij je w specyfikacjach transformacji. Dokumentuj specyfikacje transformacji w macierzy mapowania źródło-cel lub w arkuszu kalkulacyjnym.

2. Przetestuj funkcje narzędzia ETL. Bardzo ważne jest przetestowanie funkcji narzędzia ETL przed zaprojektowaniem przepływu procesu ETL i przed podjęciem decyzji o konfiguracji obszaru pomostowego. Na przykład bezwartościowe byłoby instalowanie popularnego obecnie narzędzia ETL, które nie potrafi czytać płaskie pliki na komputerze mainframe, jeśli 90 procent danych źródłowych znajduje się w plikach płaskich na komputerze mainframe. Dlatego przetestuj funkcje narzędzia ETL i określ, czy w celu wykonania skomplikowanych i długotrwałych przekształceń, których narzędzie nie może obsłużyć, należy napisać dodatkowy kod.

3. Zaprojektuj przebieg procesu ETL. Najtrudniejszym aspektem projektowania ETL jest stworzenie wydajnego przepływu procesów ETL. Ponieważ większość okien do przechowywania danych jest bardzo mała — tylko kilka godzin na dobę — proces ETL musi być maksymalnie uproszczony. To znaczy rozbicie procesu ETL na małe komponenty programu, tak aby jak najwięcej mogło być uruchomionych równolegle.

4. Projektuj programy ETL. Ponieważ większość organizacji wymaga załadowania danych historycznych z kilku lat w pierwszej wersji aplikacji BI, należy wziąć pod uwagę trzy zestawy programów ETL: ładowanie początkowe, ładowanie historyczne i ładowanie przyrostowe. Obciążenie przyrostowe będzie prawdopodobnie obciążeniem delta i dlatego będzie najbardziej skomplikowane do zaprojektowania. Modularyzuj programy ETL tak bardzo, jak to możliwe i twórz specyfikacje programowe dla każdego modułu programu ETL.

5. Skonfiguruj obszar przemieszczania ETL. Określ, czy potrzebujesz scentralizowanego obszaru pomostowego na serwerze dedykowanym, czy też bardziej sensowne byłoby wdrożenie zdecentralizowanego obszaru pomostowego w swoim środowisku. Decydujące czynniki to rodzaj i lokalizacja plików źródłowych i źródłowych baz danych, a także funkcje, możliwości i warunki licencjonowania narzędzia ETL.

Nie twórz oddzielnego obszaru pomostowego dla każdej zbiorczej bazy danych. Zdecentralizowany, skoordynowany obszar pomostowy to nie to samo, co oddzielne, nieskoordynowane obszary pomostowe dla różnych docelowych baz danych BI i różnych aplikacji BI.

Rezultaty wynikające z tych działań

1. Dokument mapowania źródła do celu . Ten dokument zawiera specyfikacje transformacji dla każdej kolumny BI, w tym instrukcje dotyczące czyszczenia danych, sprawdzania RI, uzgadniania i obsługi błędów, a także algorytmy dla agregacji i podsumowań.

2. Schemat przebiegu procesu ETL. Diagram przepływu procesu ETL przedstawia sekwencję procesu i zależności procesu między wszystkimi komponentami procesu ETL, takimi jak moduły programowe, tymczasowe i stałe pliki robocze i tabele oraz narzędzia do sortowania, scalania i ładowania.

3. Dokument projektowy programu ETL. Ten dokument jest tworzony z dokumentu mapowania źródło-cel po określeniu przepływu procesu ETL. Zawiera rzeczywiste specyfikacje programowania dla każdego modułu programu ETL dla ładowania początkowego, ładowania historycznego i ładowania

przyrostowego. Fragmenty tego dokumentu zostaną przekazane różnym programistom ETL w celu zakodowania modułów programu.

4. Obszar inscenizacji. Obszar pomostowy powinien zawierać biblioteki programów z kontrolą wersji oraz wydzieloną przestrzeń na tymczasowe i stałe pliki robocze oraz tabele.

Role zaangażowane w te działania

1.Analityk jakości danych. Współpracując z głównym programistą ETL, analityk jakości danych musi przekazać swoją wiedzę na temat stanu plików źródłowych i źródłowych baz danych odkrytych w kroku 5, Analiza danych. Ponieważ analityk jakości danych zwykle ma doświadczenie w analizie systemów, może on asystować lub nawet przejąć tworzenie dokumentu mapującego źródło do celu.

2.Administrator bazy danych. Administrator bazy danych musi być zaangażowany w projektowanie procesu ETL ze względu na aspekty związane z bazą danych (RI, indeksowanie, klastrowanie i użycie narzędzia do ładowania DBMS). Administrator bazy danych może wnieść cenne dane wejściowe do przebiegu procesu ETL i czasami może skrócić okno przemieszczania ETL o kilka godzin.

3. Główny programista ETL. Główny programista ETL jest odpowiedzialny za cały proces ETL. Z pomocą administratora bazy danych, analityka jakości danych i eksperta merytorycznego, główny programista ETL projektuje przebieg procesu ETL i tworzy dokument projektu programu ETL z rzeczywistymi specyfikacjami programowania dla programistów ETL (lub instrukcjami dla ETL narzędzie).

4. Ekspert merytoryczny. Na tym etapie rolę doradczą pełni ekspert merytoryczny. Ponieważ ekspert merytoryczny był zaangażowany w identyfikację danych źródłowych i znalezienie problemów z jakością danych, powinien on uczestniczyć w tworzeniu źródła do

dokument mapowania docelowego. Ekspert w danej dziedzinie będzie również łącznikiem z przedstawicielem biznesowym, który musi zweryfikować reguły biznesowe stosowane w procesie ETL.

Ryzyko niewykonania kroku 9

Nie jest to opcjonalny krok — nawet jeśli planujesz używać narzędzia ETL. Każdy zespół projektu BI musi ocenić dane źródłowe i dowiedzieć się, jak je ulepszyć, zmienić, ujednolicić i uczynić bardziej użytecznym przed przeniesieniem ich do docelowych baz danych BI. Projekt BI nie przypomina projektu konwersji systemów, w którym po prostu próbujesz przejść z jednej platformy technologicznej na drugą, jednocześnie przesyłając dane jako

jest. Projekt BI jest bardziej podobny do przeprojektowania systemu lub projektu poprawy procesów biznesowych, w którym chcesz zmienić dane. Nie możesz sobie pozwolić na przenoszenie danych ze źródła do celu, a następnie czekanie, aż baza danych odrzuci element danych z przyczyn technicznych. Musisz zaplanować i zaprojektować wymagane zmiany ze względów biznesowych.