Przegląd procesu ETL dla księgarni

Pliki:

Księgarnia Integration.zip – projekt typu Integration Services Project dla Księgarni Księgarskiego

Źródła.zip – pliki z danymi źródłowymi:
rachmistrz-schema.sql – tworzenie tabel dla systemu rachmistrz
1.rachmistrz.sql – dane dla systemu rachmistrz dla pierwszego importu (T0-T1)
2.rachmistrz.sql – przyrost danych dla systemu rachmistrz dla drugiego importu (T1-T2)
1.Prezes.xlsx – „exel prezesa" dla pierwszego importu (T0-T1)
2.Prezes.xlsx – „exel prezesa" dla drugiego importu (T0-T2)
locations.csv – plik pomocniczy, zawiera mapowanie kodów pocztowych na województwa

Skrypty.zip – pomocnicze skrypty SQL:
ksiegarnia.sql – tworzenie tabel dla hurtowni
tworzenie tabel pomocniczych.sql – tworzenie tymczasowych tabel używanych w procesie ETL
usuwanie tabel pomocniczych.sql – usuwanie tych tabel
czas.sql – wypełnianie tabeli czas w hurtowni
swieta.sql, wakacje.sql – wypełnianie tabel pomocniczych – terminy wakacji i świąt

Uruchomienie procesu:

W projekcie są dwa pakiety:
Dane wstępne.dtsx – będzie wykonywany raz – wstawia dane wymiarów, które są znane i nie ma sensu ich wyłuskiwać ze źródeł (w tym przypadku czas – będą to 24 wpisy dla poszczególnych godzin, data – wstawiane wszystkie daty z zakresu używanego w źródłach (1980-2015), oraz tabela Śmieci – zakładamy, że lista możliwych stanowisk oraz form płatności się nie będzie zmieniać).
Import danych.dtsx – import właściwych danych ze źródeł do hurtowni. Będzie wykonywany cyklicznie (w naszym przypadku 2 razy, dla danych z czasu T1 oraz T2).

Kroki:

1. Jeśli nie stworzono bazy danych dla hurtowni, zrobić to (księgarnia.sql).
2. Jednokrotne uruchomienie Dane wstępne.dtsx – otworzyć pakiet, dla wszystkich menadżerów połączeń ustawić właściwe lokalizacje plików oraz parametry połączenia (dla źródła TEST1.auxiliary „test connection" się nie powiedzie, baza zostanie dopiero stworzona)
3. Wybrać Start. Wszystkie zadania powinny się pomyślnie wykonać. Dla „Wprowadzenie dat" można sprawdzić Data flow. W Management studio można sprawdzić, że wypełnione zostały tabele Data, Czas oraz Śmieci.
4. Ustawianie źródeł danych na czas T1: stworzyć bazę dla systemu rachmistrz (rachmistrz-schema.sql) oraz wypełnić ją danymi do punktu T1 (1.rachmistrz.sql).
5. Otworzyć pakiet Import danych.dtsx i dla wszystkich menadżerów połączeń ustawić właściwe parametry połączenia bądź lokalizacje plików (dla Excel Connection Manager - Prezes.xlsx wybrać plik 1.Prezes.xlsx).
6. Wybrać Start. Pozostałe tabele powinny zostać wypełnione.

7. Ustawianie źródeł danych na czas T2: uzupełnić bazę systemu rachmistrz danymi do punktu T2 (2.rachmistrz.sql, nie usuwać dotychczasowych danych), zmienić plik w menadżerze połączenia Excel Connection Manager - Prezes.xlsx z 1.Prezes.xlsx na 2.Prezes.xlsx
8. Ponownie wybrać start dla pakietu Import danych.dtsx. Dane z czasu T1-T2 zostaną uzupełnione.

Ogólny przegląd projektu:

Pakiet Dane wstępne.dtsx:

Menadżer połączenia TEST1.auxiliary ma we właściwościach ustawione {Delay validation: True}, ponieważ baza auxiliary jest tworzona przez zadanie Tworzenie tabel pomocniczych i przed jego wykonaniem nie da się z tą bazą połączyć.

Zielone strzałki w Control flow oznaczają poprzedzanie zadań (które zadanie musi być wykonane dopiero po zakończeniu jakiegoś innego), w Data flow zielone strzałki oznaczają przepływ danych.

Role poszczególnych zadań:

Tworzenie tabel pomocniczych – tworzy tabele dla wakacji i świąt, które po uzupełnieniu będą wykorzystane do wstawienia dat do hurtownii.

Wypełnienie świąt, Wypełnienie wakacji – uzupełniają odpowiednio tabele świąt i wakacji.

Usuwanie tabel pomocniczych – po wykorzystaniu tabele świąt i wakacji są usuwane (tak jak i cała baza auxiliary).

Wszystkie te zadania są typu Execute SQL task i korzystają z zewnętrznych plików.

Są też 2 zadania Execute SQL task korzystające z poleceń SQL podanych bezpośrednio. Są to Wypełnianie innych oraz Wypełnianie czasu, które wypełniają odpowiednio tabele Śmieci oraz Czas.

Ostatnie zadanie (Wprowadzenie dat) jest typu Data Flow Task i służy do wypełnienia tabeli Data. Dwukrotne kliknięcie tego zadania spowoduje przejście do zakładki data flow, gdzie będzie zobrazowany stworzony przepływ danych.

Przepływ Wprowadzenie dat:

Aby móc swobodnie sprawdzać ustawienia poszczególnych komponentów (w szczególności lookupów) należy przed rozpoczęciem oglądania przepływu danych wykonać zadanie Tworzenie tabel pomocniczych i dla menadżera połączeń TEST1.auxiliary wykonać Test Connection – program nie będzie zgłaszał błędów przy oglądaniu mappingów kolumn; po zakończeniu prac należy oczywiście wykonać Usuwanie tabel pomocniczych).

Przepływ danych zaczyna się od źródła typu OLE DB. Wykorzystuje ono polecenie SQL generujące daty z wybranego zakresu (@Start = '1980-01-01', @End = '2015-12-31'). Ponadto na podstawie wygenerowanej daty tworzone są dodatkowe pola takie jak Rok, Miesiąc, Dzień tygodnia i Dzień pracujący.

Następnie dla danej daty określane jest czy jest to święto – Lookup o nazwie Święto. Wyszukujemy w tabeli swieta wpisów o takiej samej dacie jak rozpatrywana. Jeśli zostanie znalezione, to do

przepływu danych dodawane są pola święto oraz wolne. Jeśli nie, to wiersze bez odpowiadających im wpisów w tabeli swieta trafiają do „No match output”. W tym celu trzeba ustawić w lookup, żeby wiersze bez pary wysyłał na „No match output” – domyślnie komponent będzie zgłaszał błąd.

Dla dni świątecznych sprawdzane jest, czy są to święta ustawowo wolne (kolumna wolne) i jeśli są, to wartość pola „Dzień pracujący” ustawiana jest na "Dzień wolny", jeśli nie, to pozostawiana jest dotychczasowa wartość – komponent Dzień wolny. Dla dni nieświątecznych dodawana jest pole święto o wartości "Zwykły dzień" – komponent Brak święta. Następnie oba przepływy są łączone (komponent Union All).

Następny lookup (Dzień przedświąteczny) wykonuje analogiczną operację, ale dla dnia przed świętem. Stąd też źródłem dla lookupa jest nie tabela, ale zapytanie które daty z tabeli swieta cofa o 1 a do nazwy święta dostawia „Jutro ”.

Ostatni lookup (Lookup) przeprowadza operację określenia, czy dana data jest w wakacje. Operacja byłaby analogiczna do poprzednich, ale tu trzeba sprawdzić, czy sprawdzana data zawiera się pomiędzy początkiem a końcem jakichkolwiek wakacji. Aby to zrobić w lookap trzeba użyć Custom query. Aby to uzyskać w zakładce general Cache mode trzeba ustawić na partial. Następnie w zakładce Colums ustawiamy mapping. Jak dokładnie wygląda nie jest istotne, ważne aby użyta została kolumna Data, która będzie parametrem w docelowym zapytaniu. Następnie w zakładce adwanced wybieramy Modify the SQL statement i tam podajemy docelowe zapytanie – za znak zapytania zostanie podstawiona wartość pola data rozpatrywanego wiersza (przycisk parameters…).

Na koniec data jest wstawiana do tabeli w hurtowni (OLE DB Destination). Proszę zwrócić uwagę, że nie jest wprowadzana żadna wartość dla klucza surogatowego (będzie wygenerowana automatycznie). Proszę to przyjąć za standard – wypełnianie klucza surogatowego w procesie ETL będzie uznawane za błąd.

Pakiet Import danych.dtsx:

W pakiecie tym jest 7 zadań typu data flow. 2 z nich dotyczą wstawiania faktów, 5 wstawiania wymiarów. Zielone strzałki ustawiają poprzedzanie. W ogólności aby wstawiać fakty należy najpierw wstawić wszystkie wymiary z nimi związane, aby można było pograć klucze obce (trzeba je pobrać z hurtowni, gdyż nie można ich ustawić na wybrane wartości – w procesie ETL będą otrzymywały wartości, które należy uważać za losowe). W tym przypadku dodatkowo przed wprowadzaniem pracowników należy wstawić księgarnie – pracownik będzie przypisywany do księgarnii, a wstawianie pracownika rozbite jest na 2 etapy ze względu na hierarchię rekurencyjną.

W zadaniu Wprowadzanie księgarni:

Informacje o księgarniach są pobierane z arkusza 1 pliku exel (Księgarnie – Exel). Brakuje w nich danych o województwach – te pobieramy z pliku locations.csv (Województwa - Flat File Source). Dane są złączane po kodzie pocztowym – stąd sortowanie obu przepływów danych po tej kolumnie. Dla danych z pliku csv odrzucane są powtórzenia kodu pocztowego – założono, że wszystkie miejsca o tym samym kodzie pocztowym są w jednym województwie (jeśli będą 2 wpisy z tym samym kodem i różnym województwem, to wybrane będzie losowe z nich).

Dla księgarni potrzebna jest jej wielkość, która jest określana na podstawie liczby pracowników. Stąd pobranie pracowników z arkusza 2 exela prezesa (Pracownicy – Exel); odrzucani są pracownicy, którzy już nie pracują (Conditional Split - Brak daty zwolnienia) a reszta jest zliczana względem nr.

Identyfikacyjnego księgarni (komponent Zliczanie pracowników typu Aggregate). Natej podstawie określana jest wielkość księgarni słownie (Wielkość księgarni) i następnie dokonywane jest złączenie z głównymi danymi o księgarni przy użyciu numeru. Księgarnia jest traktowana jako wolno zmieniający się wymiar – zmienia się pole wielkość księgarni,a nowa wielkosc zastępuje starą – odpowiednio ustawiono komponent Slowly Changing Dimension. Oba komponenty z min połączone zostają dodane automatycznie po skonfigurowaniu Slowly Changing Dimension.

Wprowadzanie pracowników

W przepływie danych dotyczących wprowadzania pracownika są 2 ważne części:

- pobieranie informacji o pracowniku - Pracownicy – Exel i późniejsze komponenty. Komponent Script Component jest to wywołanie skryptu C#, który dostaje na wejściu poszczególne wiersze z przepływu danych i może je przetwarzać. W szczególności, po zdefiniowaniu w zakładce Inputs and Outputs kolumn wyjściowych może określać ich wartości. W przykładzie obliczany jest StażPracy i PrzedziałWiekowy
- uzyskiwanie informacji o kluczu surogatowym księgarni, w której pracuje pracownik: po pobraniu danych z Województwa - Flat File Source oraz Księgarnie – Exel określane jest województwo dla każdej księgarni, tak jak przy wstawianiu informacji o księgarni, a następnie przy użyciu komponentu Lookup pobierane są klucze surogatowe dla każdej z nich (na podstawie nazwy, miasta i województwa) – w razie nie znalezienia klucza surogatowego komponent zgłosi błąd.. Następnie ten przepływ danych jest łączony z przepływem o pracowniku po nr identyfikacyjnym księgarni. Dzięki temu wstawiając pracownika do bazy znany jest klucz obcy do księgarni, w której pracuje.

Pobrane dane wstawiane są do bazy (bez informacji o szefie – następny krok) jako wolno zmieniający się wymiar. W komponencie Slowly Changing Dimension biznes key jest ustawiany an (PESEL, data wstawienia). Data wygaśnięcia oraz ImieiNazwisko ustawione są na Changing attribute – w razie zmiany wartość zostanie nadpisana. Pozostałe wstawiane kolumny na Historical attribute – w razie wystąpienia zmiany w którejkolwiek z nich zostanie stworzony nowy wpis w tabeli Sprzedawca. Stanowisko pracownika nie może się zmieniać – stąd ustawione Fixed attribute dla tej kolumny (w źródle danych pracownik o zmienionym stanowisku jest dodawany jeszcze raz i ma nową datę wstawienia). Aktualność wpisu jest oznaczana przez kolumnę aktualność typu logicznego. Komponenty pod komponentem Slowly Changing Dimension zostały wygenerowane automatycznie na podstawie podanej konfiguracji. Należy również zaznaczyć, że jeśli na wejściu Slowly Changing Dimension znajdzie się wiersz taki sam, jak znajdujący się już w bazie (nie uwzględniając oczywiście klucza surogatowego), to nie zostanie ponownie dodany.

Wprowadzanie szefów

Określenie kto jest czyim szefem odbywa się w prosty sposób – w każdej księgarni jest jedna osoba na stanowisku dyrektor, pozostali to pracownicy. Zestawiane są więc dla każdej księgarni pary pracownik – dyrektor (dla wszystkich pracowników; Merge Join) i pobierane są odpowiednie klucze surogatowe. Następnie w Slowly Changing Dimension dane te wstawiane są do bazy. Klucze biznesowe są takie jak w poprzednim zadaniu. Tym razem jednak ustawiany jest, jako atrybut historyczny klucz szefa. Pozostałe atrybuty zaś nie mogą się zmienić – w razie zmiany powinny były być zmienione w poprzednim zadaniu. Tym razem jednak ustawione jest wsparcie dla wywiedzionych wierszy (inferred member). Wiersz wywiedziony to taki, który został wstawiony bez pełnej wiedzy o

nim i część kolumn ma pustych. Dla wierszy wywiedzionych komponent Slowly Changing Dimension będzie aktualizował wczytany wpis, a nie tworzył nowy. W przypadku przykładu rozpoznanie czy w wierszu są dane do uzupełnienia określone jest przez kolumnę inferred. Jeśli tak jest, klucz obcy na szefa zostanie dopisany do wiersz, w przeciwnym wypadku, jeśli szef się zmieni, to wstawiany nowy wiersz. W obu przypadkach kolumna inferred ustawiana jest na false, co jednak trzeba było dodać ręcznie do wygenerowanych automatycznie komponentów.

Wprowadzanie książek

Wstawianie danych o książkach jest dość proste – pobierana ze źródła jest informacja o książce(Książka) i jej sprzedaży(Sprzedaż książki). Na podstawie informacji o sprzedaży określane są przedziały cenowe (Przedział cenowy) w których książka o zadanym ISBN występuje. Po złączeniu przepływów sprawdzana jest informacja już znajdująca się w hurtowni – wyszukiwane są informacje o książkach o tym samym numerze ISBN i przedziale cenowym, tylko wiersze, dla których nie znaleziono w hurtowni istniejącego wpisu są do niej dodawane. Przy wstawianiu informacji również w tym przypadku nie ustawia się klucza surogatowego.

Wprowadzanie autorstw

Wprowadzanie faktów w ogólności polega na pobraniu informacji o fakcie ze źródła oraz o wszystkich wymiarach z którymi jest powiązany. Na podstawie informacji o powiązanych wymiarach wyszykuje się właściwe klucze surogatowe z tabeli wymiarów. Może być konieczne obliczenie miar, jeśli w źródle nie są podane wprost.

W przypadku przykładu pobierana jest informacja z tabeli wiele-do-wiele autorstwo i łączona z kluczem surogatowym książki (w źródle danych kluczem książki jest ISBN). Następnie w źródle danych wyszukiwana jest informacja o autorze (Lookup - Autor). Po przetworzeniu uzyskanej informacji określany jest klucz surogatowy autora (Lookup - Autor HD), następnie sprawdzane jest, czy uzyskana informacja nie znajduje się już w bazie (Lookup - Autorstwo HD) i wstawiana jest tylko ta informacja, której brakuje.

Wprowadzanie sprzedaży

Wstawianie faktów sprzedaży przebiega analogicznie. Pobierane są dane o sprzedaży(Sprzedaż) i łączone z rachunkami (Rachunek). Obliczane są miary: łączna cena i zysk (Ceny Zyski). Następnie wyszukiwane są klucze surogatowe dla wielu wymiarów. Dla niektórych może to łączyć się z przetworzeniem posiadanych danych (wyliczenie przedziału cenowego dla książki { Przedział cenowy }, wydobycie z kolumny DATETIME osobno daty i godziny{ Derived Column}).

Dla sprzedawcy pobierany jest klucz surogatowy z aktualnego wpisu (Aktualność = true) o niezwolnionym sprzedawcy (DataWygasniecia = null). Jeśli taki nie występuje w bazie, to wyszukiwany jest wpis o najpóźniejszym czasie zwolnienia dla zadanego numeru PESEL (Sprzedawca przez zwolnieniem HD). Jak zwykle przed wstawieniem jest sprawdzanie, czy informacji już nie ma w hurtowni (Sprzedaż_HD).