Projektowanie schematów logicznych dla magazynów danych

Bartosz Bębel, Mikołaj Morzy
Politechnika Poznańska, Instytut Informatyki
ul. Piotrowo 3A, 60-965 Poznań
Bartosz.Bebel@cs.put.poznan.pl, Mikolaj.Morzy@cs.put.poznan.pl

Abstrakt. Technologia magazynów danych pozwala na aktywne wykorzystanie informacji posiadanych przez przedsiębiorstwo do podejmowania strategicznych decyzji, określania trendów rynkowych lub zarządzania zasobami. W artykule przedstawiono model tworzenia korporacyjnego magazynu danych i szczegółowo opisano metodologię projektowania schematów baz danych dla potrzeb magazynów danych. Poszczególne zagadnienia zostały zobrazowane praktycznymi przykładami.

1. Wstęp

Większość współczesnych przedsiębiorstw stara się wykorzystywać najnowsze technologie do zwiększenia konkurencyjności i odniesienia sukcesu ekonomicznego. Jedną z popularnych metod jest aktywne wykorzystanie informacji gromadzonych na przestrzeni lat przez te przedsiębiorstwa. Możliwość dokładnej analizy tych informacji pozwala na poprawę jakości procesu podejmowania decyzji oraz na zwiększenie dochodowości poprzez szybsze reagowanie na zmiany zachodzące w otoczeniu przedsiębiorstwa.

Podstawową przeszkodą na drodze aktywnej analizy danych i informacji zgromadzonych przez przedsiębiorstwo jest niekompatybilność licznych systemów transakcyjnych obsługujących bieżącą działalność przedsiębiorstwa. Systemy obsługi bieżącej nie są dostosowane do potrzeb procesu wspomagania decyzji i często zawierają niespójne informacje. W celu przezwyciężenia tych problemów w ostatnich latach rozwinęła się technologia magazynów danych (ang. *data warehouse*).

Magazyn danych jest zbiorem kluczowych informacji wykorzystywanych do zarządzania przedsiębiorstwem. Informacjami mogą być dane pomagające decydować o poziomie zapasów w magazynie, dane demograficzne do przeprowadzania kampanii promocyjnych, czy też podsumowania i dane zbiorcze wspomagające podejmowanie strategicznych decyzji na poziomie makroekonomicznym. Magazyn danych to nie tylko tematycznie zorientowane dane, ale także cały proces ekstrakcji informacji z heterogenicznych źródeł danych (systemy transakcyjne, sieć WWW, arkusze kalkulacyjne, pliki tekstowe) do relacji w bazie danych, oraz przetwarzania danych do postaci atrakcyjnej dla analityków i decydentów. Podsumowując, magazynem danych nazywamy dane (metadane, fakty, wymiary, agregaty) oraz procesy (ładowanie, odświeżanie, odpytywanie), które wspólnie udostępniają dane i umożliwiają decydentom podejmowanie strategicznych decyzji.

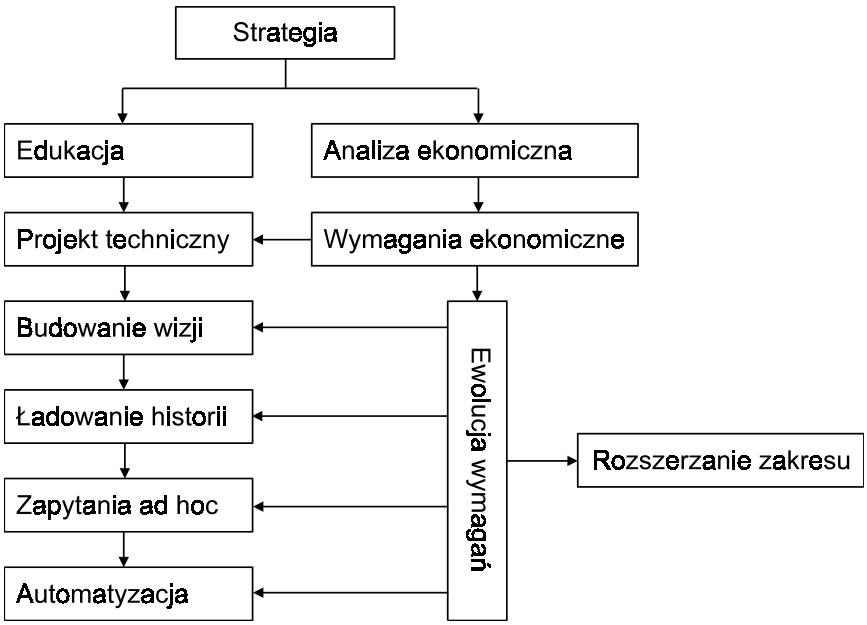
Niniejszy artykuł jest zorganizowany następująco. W rozdziale 2 przedstawiono szczegółowo proces konstruowania magazynu danych. Rozdział 3 poświęcony jest metodom tworzenia logicznego schematu bazy danych, ze szczególnym uwzględnieniem projektowania relacji faktów, relacji wymiarów i relacji zbiorczych. Rozdział 4 stanowi krótkie podsumowanie. Wiele zagadnień zostało zobrazowanych przykładami zaczerpniętymi z dziedziny telefonii komórkowej. Schemat logiczny magazynu danych dla firmy telekomunikacyjnej, do którego odnoszą się przykłady, został przedstawiony w załączniku A.

2. Proces konstruowania magazynu danych

Rozwiązania stosowane w dziedzinie magazynów danych różnią się zasadniczo od rozwiązań stosowanych w tradycyjnych systemach transakcyjnych. Podstawowa różnica polega na tym, że magazyny danych nigdy nie są statyczne, lecz nieustannie zmieniają się by odzwierciedlić ewolucję przedsiębiorstwa i jego zmieniające się potrzeby. W praktyce oznacza to, że magazyny danych muszą być projektowane w elastyczny sposób, który będzie pozwalał na łatwe wprowadzanie modyfikacji do struktury magazynu danych. Kłopot polega na tym, że w momencie konstruowania magazynu danych przyszłe potrzeby i wymagania są w ogólności nieznane.

Z powodu nieustannie zmieniających się wymagań proces konstrukcji magazynu danych różni się fundamentalnie od metodologii stosowanej powszechnie w przypadku systemów transakcyjnych. Tradycyjne podejście zakładające rozpoczęcie projektowania architektury i struktury systemu po uprzednim zakończeniu fazy analizy może łatwo doprowadzić do „paraliżu przez analizę", ponieważ najczęściej pełne zebranie wszystkich wymagań jest niemożliwe. W rzeczywistości konieczne jest konstruowanie magazynu danych na podstawie aktualnie dostępnych wymagań oraz tego, co projektanci mogą przewidzieć na temat przyszłych zastosowań, wymagań, profilów zapytań i innych parametrów projektowanego magazynu danych.

Na rysunku 1 przedstawiono ogólny schemat procesu tworzenia magazynu danych.



Rys. 1. Proces tworzenia magazynu danych

- Strategia

Magazyn danych jest strategiczną inwestycją przedsiębiorstwa, a koszty jego budowy mogą być bardzo wysokie. Projekt powinien mieścić się w ramach szerszej strategii informatycznej przedsiębiorstwa, ponieważ w przeciwnym wypadku jego finansowanie może okazać się trudne.

- Analiza ekonomiczna

Celem tej fazy jest identyfikacja wszystkich zysków, jakie przedsiębiorstwo osiągnie poprzez implementację magazynu danych. Nawet jeśli zakładane zyski nie są wymierne, powinny być jasno określone na samym początku projektu.

- Edukacja i prototyp

Magazyn danych stwarza nowe możliwości, ale też wymaga od użytkowników nowych umiejętności. Dobrym pomysłem jest stworzenie prototypu, za pomocą którego przyszli użytkownicy posiądą wymagane umiejętności i nabiorą zaufania do nowej technologii. Prototyp nie powinien być zaczątkiem magazynu danych, tzn. nie powinien być dalej rozwijany, ponieważ architektura prototypu jest najprawdopodobniej nieskalowalna w kontekście całego magazynu danych.

- Wymagania ekonomiczne

W trakcie tej fazy określa się logiczny model informacji przechowywanej w magazynie danych, systemy źródłowe z których czerpie się informacje, reguły ekonomiczne stosujące się do informacji, profile zapytań kierowanych do magazynu danych, itp. Prawidłowe zrozumienie krótko- i średnioterminowych potrzeb przedsiębiorstwa pozwala na zbudowanie efektywnego magazynu danych, który będzie odpowiadał aktualnym wymaganiom i będzie mógł ewoluować w momencie pojawienia się nowych wymagań. Ponadto należy przeznaczyć nieco wysiłku na określenie prawdopodobnych potrzeb długoterminowych. Znajomość takich potrzeb pozwala na elastyczną konstrukcję magazynu danych.

- Projekt techniczny

W fazie projektu technicznego należy określić całokształt architektury spełniającej długoterminowe wymagania, oraz wszystkie komponenty konieczne do implementacji magazynu danych. Projekt musi określić m.in.: ogólną architekturę systemu, architekturę serwera i tematycznych magazynów danych, najważniejsze elementy schematu bazy danych, okres przechowywania danych w magazynie, strategie archiwizowania i odtwarzania magazynu oraz plan obciążenia sprzętu. Na tym etapie nie tworzy się szczegółowego schematu logicznego magazynu danych, lecz tylko identyfikuje jego najważniejsze komponenty.

- Budowanie wizji

Głównym celem tego etapu jest dostarczenie małego, w pełni funkcjonalnego prototypu. Najczęściej tworzony jest system stanowiący niewielki fragment planowanego magazynu. Gotowy prototyp powinien zaspokajać najbardziej palące potrzeby informatyczne przedsiębiorstwa.

- Ładowanie historii

W trakcie tej fazy magazyn danych nie jest rozbudowywany „wszerz" (dodawanie nowych encji) lecz „w głąb" (poszerzanie horyzontu czasowego informacji przechowywanych w magazynie danych). Jeśli w fazie budowania wizji do magazynu danych załadowano dane obejmujące ostatnie trzy miesiące połączeń telefonicznych, to w fazie ładowania historii do magazynu danych zostaną dodane wszystkie pozostałe informacje o połączeniach na przestrzeni ostatnich lat. Gwałtowny wzrost wolumenów danych, jaki pojawia się w tej fazie, komplikuje zagadnienia związane z pielęgnacją i utrzymaniem magazynu. Na tym etapie należy opracować dokładne strategie tworzenia kopii zapasowych bazy danych, odtwarzania po awarii, partycjonowania danych, itp.

- Zapytania ad hoc

Kolejna faza projektowania magazynu danych polega na dokładnej analizie profili zapytań wydawanych przez użytkowników i strojeniu narzędzi dostępu do bazy danych. Większość użytkowników magazynu danych nie potrafi operować językiem SQL i do wydawania zapytań wykorzystuje różnego rodzaju wizualne generatory zapytań. Zadaniem projektantów magazynu danych jest takie skonfigurowanie tych narzędzi dostępu, aby generowane przez nie plany

wykonania zapytania były możliwie efektywne. Zadanie to może wymagać wprowadzenia licznych zmian w strukturze bazy danych (statystyki, dodatkowe indeksy, dodatkowe perspektywy, itp.), stąd powinno być wykonane jako osobna faza procesu konstrukcji magazynu danych.

- Automatyzacja

Ta faza polega na zautomatyzowaniu możliwie dużej części procesu zarządzania magazynem danych. Wśród składowych tego procesu, które mogą podlegać automatyzacji, wymienić należy: ekstrakcję i ładowanie danych z systemów źródłowych, transformację danych, tworzenie kopii zapasowych, odtwarzanie i archiwizację danych, tworzenie predefiniowanych agregatów, monitorowanie profili najczęstszych zapytań.

- Rozszerzanie zakresu

W fazie rozszerzania zakresu magazyn danych jest rozbudowywany w ten sposób, aby funkcjonalnie objąć nowe wymagania ekonomiczne. Najczęściej zmiany dotyczą dodawania nowych encji oraz wprowadzania nowych źródeł informacji, choć czasem rozszerzenie magazynu danych może polegać tylko na dodaniu nowych perspektyw i agregatów na podstawie informacji, które już są obecne w magazynie. Wysiłek i złożoność tej operacji w pełni usprawiedliwia umieszczenie jej w ramach osobnej fazy procesu konstruowania magazynu danych.

- Ewolucja wymagań

Jak już wspomniano wcześniej, najważniejszą cechą procesu tworzenia magazynu danych jest to, że wymagania funkcjonalne dotyczące magazynu nigdy nie są statyczne, lecz ewoluują wraz z działalnością przedsiębiorstwa i zmieniającymi się warunkami rynku. W celu zapewnienia magazynowi danych maksymalnej elastyczności, system taki powinien być konstruowany przede wszystkim w oparciu o szeroko pojęty model działania przedsiębiorstwa, a nie w oparciu o wymagania konkretnych zapytań. Bardzo istotne jest, aby zmiany wymagań były nieustannie monitorowane i sygnalizowane projektantom.

3. Schemat logiczny magazynu danych

W przypadku magazynów danych najczęściej stosuje się schemat gwiazdy (ang. *star schema*), schemat płatka śniegu (ang. *snowflake schema*) lub schemat hybrydowy (ang. *starflake schema*). Wynika to z tego, że schematy gwiaździste i ich pochodne najlepiej przystają do zapytań wydawanych w systemach wspomagania decyzji. Większość zapytań posiada podobną strukturę: analizują zbiór szczegółowych informacji (transakcje dotyczące rozmów telefonicznych, zdarzenia związane z użytkownikami i klientami) poprzez agregację i grupowanie informacji względem różnych kryteriów (przedziały czasowe, typy abonamentu, lokalizacje geograficzne). Taki profil zapytań wymusza pewną logiczną organizację bazy danych. Szczegółowe informacje opisujące transakcje znajdują się w centrum, zaś otaczają je opisy kryteriów. Centralna relacja zawierająca szczegółowe dane nazywana jest relacją **faktów** (ang. *fact table*). Informacje referencyjne, takie jak hierarchie regionów geograficznych, przedziały czasowe, grupy produktów, itd., umieszczone są w relacjach **wymiarów** (ang. *dimension tables*). Oprócz tego w magazynie danych mogą się znaleźć dane zbiorcze, przechowywane w relacjach **zbiorczych** (ang. *summary tables*). Połączone relacje faktów i wymiarów tworzą **schemat gwiazdy**.

Informacje przechowywane w magazynie danych dzielą się na dwie rozłączne klasy: informacje faktyczne i referencyjne. Informacja faktyczna opisuje fizyczne wystąpienie zdarzenia w świecie rzeczywistym. Zdarzeniem takim może być transakcja w sklepie, połączenie telefoniczne, operacja bankowa lub żądanie wypłacenia ubezpieczenia. W większości magazynów danych informacje faktyczne stanowią ponad 70% zawartości całego magazynu. Ponieważ większość zapytań odwołuje się do informacji faktycznych, relacje faktów muszą być projektowane bardzo starannie. Ich zawartość oraz format przechowywanych w nich danych muszą zostać dokładnie ustalone w fazie zbierania wymagań ekonomicznych. Relacje faktów zawierają najczęściej miliony krotek, są w większości typu numerycznego i po wprowadzeniu do magazynu danych nie ulegają zmianom.

Informacje referencyjne opisują wymiary, według których są analizowane dane faktyczne. Charakter informacji referencyjnych różni się znacząco od informacji faktycznych. Relacje wymiarów są dużo mniejsze od relacji faktów i zawierają setki lub tysiące krotek, najczęściej są łańcuchami znaków (tekstowymi opisami) i często mogą ulegać modyfikacji.

Informacje zbiorcze to zagregowane kopie szczegółowych informacji przechowywanych w relacjach faktów. W przeciwieństwie do faktów informacje zbiorcze mają charakter tymczasowy i ulegają częstym modyfikacjom. Celem wstępnego obliczania informacji zbiorczych jest przede wszystkim przyspieszenie wykonywania wspólnych części zapytań. Liczba relacji zbiorczych, występujących w magazynie danych jest ściśle związana z charakterem magazynu, jego przeznaczeniem, profilami zapytań, itp. Najczęściej magazyny danych zawierają kilkadziesiąt różnych relacji zbiorczych.

Obok informacji faktycznych, referencyjnych i zbiorczych w magazynie danych przechowywane są również metadane, czyli dane opisujące zawartość magazynu. W ramach metadanych przechowywane są szczegółowe informacje o położeniu i charakterystyce każdego z zewnętrznych źródeł danych, definicje wszystkich agregatów, informacje pozwalające na kierowanie zapytań do najbardziej adekwatnych fragmentów magazynu danych. Poza tym w metadanych zawarte są wszystkie informacje niezbędne dla działania magazynu danych: statystyki, szczegóły dotyczące strategii archiwizowania i odtwarzania magazynu, itp.

3.1. Identyfikowanie faktów

Często projektanci mają kłopoty z poprawnym rozróżnieniem faktów i wymiarów. Poniżej przedstawiono metodę pozwalającą na jednoznaczne zidentyfikowanie tych encji, które w magazynie danych będą reprezentowane jako fakty.

3.1.1. Wyszukanie transakcji elementarnych

Pierwszym krokiem w identyfikacji faktów jest analiza modelu przedsiębiorstwa i wyszukanie tych transakcji, które są fundamentalne z punktu widzenia działalności przedsiębiorstwa. Przykładami takich transakcji są:

- dla handlu: transakcje zakupu w sklepie, wahania kursów giełdowych,
- dla bankowości: operacje na kontach bankowych, zmiany typów kont, założenie lub likwidacja konta,
- dla firm ubezpieczeniowych: żądania wypłacenia odszkodowania, podpisanie nowej polisy, zmiana warunków polisy,
- dla firm telekomunikacyjnych: połączenia telefoniczne, wpłynięcie opłaty, podłączenie lub odłączenie klienta.

Należy przy tym pamiętać, że nie wszystkie dane szczegółowe muszą koniecznie być faktami. Stopień szczegółowości danych może wynikać z charakterystyki źródła danych, a nie z rzeczywistego modelu informacyjnego przedsiębiorstwa.

3.1.2. Określenie kluczowych wymiarów dla faktów

Następnym krokiem jest identyfikacja podstawowych wymiarów dla każdej potencjalnej relacji faktów. Na podstawie analizy modelu logicznego należy znaleźć te wymiary, które zostaną włączone do relacji faktów jako klucze obce. W niektórych przypadkach konieczna będzie restrukturyzacja modelu logicznego. Przykładowo, jeśli relacją faktów jest relacja reprezentująca operacje na koncie bankowym, to może ona być połączona z relacją *Właściciel-konta* poprzez relacje *Konto* i *Konto-Posiadane-Przez*. Jeżeli większość zapytań analizuje transakcje bankowe z perspektywy poszczególnych właścicieli, to do relacji faktów należy bezwzględnie dodać klucz obcy reprezentujący właściciela konta. Dzięki temu wszystkie zapytania analizujące transakcje bankowe według właścicieli unikną kosztownych operacji połączeń.

3.1.3. Sprawdzenie, czy potencjalny fakt nie jest wymiarem

W systemach obsługi bieżącej wiele relacji źródłowych może zawierać pomieszane fakty i wymiary. Dzieje się tak, ponieważ relacje źródłowe zostały skonstruowane pod kątem spełniania konkretnych wymagań systemu obsługi. Jako przykład rozważmy relację *Klient*. Zawiera ona identyfikator, nazwisko, datę podpisania umowy, datę wygaśnięcia umowy, rodzaj abonamentu. W rzeczywistości faktami są tu daty wystąpienia poszczególnych zdarzeń, zaś tożsamość klienta i typ abonamentu są wymiarami. Dobrym testem pozwalającym na zidentyfikowanie takiej sytuacji jest:

- sprawdzenie, czy potencjalna relacja faktów nie jest relacją wymiarów zawierającą powtarzające się grupy faktów,
- sprawdzenie, czy potencjalna relacja faktów nie będzie w przyszłości ulegała modyfikacjom. Jeśli istnieje prawdopodobieństwo, że krotki w tej relacji będą ulegać w przyszłości modyfikacji, to należy taką relację kandydującą podzielić na fakty i wymiary.

3.1.4. Sprawdzenie, czy potencjalny wymiar nie jest faktem

Niektóre encje mogą być jednocześnie postrzegane jako fakty i wymiary. Encja *Klient* jest faktem w przypadku magazynu danych nakierowanego na marketing i budowanie profili klientów, zaś w magazynie danych dla analizy sprzedaży detalicznej staje się wymiarem. Wybór klasy, do której należy dana encja, zależy od charakteru konstruowanego magazynu danych. W przypadku zaistnienia wątpliwości należy sprawdzić, z ilu różnych wymiarów można postrzegać daną encję. Jeśli takich wymiarów jest więcej niż trzy, to encja jest prawdopodobnie faktem.

3.2. Projektowanie relacji faktów

Tworząc relacje faktów projektant powinien odpowiednio zrównoważyć wartość informacji przechowywanej w takiej relacji i koszt jej utworzenia. Czynniki, takie jak poziom szczegółowości informacji lub horyzont czasowy danych, powinny być skorelowane z kosztem pielęgnowania i modyfikowania relacji faktów. Poniżej przedstawiono kilka technik, które pozwalają znacząco obniżyć koszt utworzenia i pielęgnacji relacji faktów, zachowując jednocześnie jej jakość.

3.2.1. Identyfikacja horyzontu czasowego dla każdej z funkcji

Projektanci magazynów danych często popełniają błąd polegający na założeniu, że szczegółowe informacje faktyczne muszą być przechowywane w magazynie danych przez długi okres czasu, np. przez 10 lat. Powszechny jest także brak różnicowania stopnia szczegółowości przechowywanych danych w zależności od ich wieku. W rzeczywistości takie podejście może negatywnie wpłynąć na efektywność zapytań kierowanych do magazynu danych. Opracowanie strategii stopniowego agregowania danych wraz z ich starzeniem się może znacząco zmniejszyć objętość wolumenu danych przechowywanych w relacji faktów, a co za tym idzie, przyspieszyć wykonywanie zapytań.

Najczęściej szczegółowe dane nie muszą być przechowywane przez okres dłuższy niż kilka lat. Dla raportów rocznych wystarczające powinny być agregaty sumujące fakty z dokładnością do tygodnia. Dla zapytań odczytujących fakty starsze niż 5 lat najprawdopodobniej w zupełności wystarczają podsumowania miesięczne zamiast dziennych, itd. Pomocne dla projektanta może okazać się stworzenie wykresu, na którym dla każdej funkcji (typu zapytania) znajdą się przedziały czasowe, w jakich poszczególne fakty muszą być szczegółowe.

3.2.2. Czy dane szczegółowe można zastąpić próbkowaniem?

Inną metodą znacznego zmniejszenia objętości relacji faktów jest zastąpienie danych szczegółowych przez reprezentatywną próbkę. Reszta danych jest przechowywana wówczas jako dzienne lub tygodniowe agregaty. Ta technika znajduje zastosowanie przede wszystkim w przypadku magazynów danych, dla których znajomość wszystkich szczegółowych faktów jest niepotrzebna. Jeśli firma pragnie przeanalizować schematy zachowań abonentów w okresie letnim, to taka analiza może być z powodzeniem dokonana na próbce zawierającej 15% faktów opisujących

rozmowy między abonentami. Z drugiej strony, jeśli firma pragnie przeprowadzić zogniskowaną kampanię reklamową dotyczącą szczególnego segmentu rynku, próbkowanie może okazać się niewystarczające, ponieważ w próbce znajdzie się zbyt mało faktów opisujących zachowania docelowych abonentów.

3.2.3. Wybór właściwych atrybutów

Kolejnym krokiem jest usunięcie z relacji faktów wszystkich zbędnych atrybutów. Dla każdego atrybutu należy zadać pytania: „Czy ten atrybut wnosi jakąś nową wiedzę o fakcie?", „Czy istnieje jakieś inne miejsce, skąd można wywieść tę daną?", „Czy atrybut ma istotne znaczenie, czy może służy do celów związanych z implementacją systemu?". Dane wywiedzione lub agregaty nie powinny być przechowywane w relacji faktów, ponieważ bardziej efektywne jest generowanie tych informacji „w locie". Wszystkie atrybuty, które są obecne w relacji faktów tylko ze względu na specyficzne potrzeby systemu obsługi bieżącej, z którego pobrano fakty, powinny być bezwzględnie usunięte z magazynu danych.

W przypadku firmy telekomunikacyjnej atrybuty, które powinny się znaleźć w relacji faktów, to: numer inicjujący połączenie, numer docelowy, data połączenia, oraz czas trwania połączenia.

3.2.4. Minimalizacja rozmiarów atrybutów w relacji faktów

Projektanci baz danych mają tendencję do przeszacowywania rozmiarów atrybutów w poszczególnych relacjach. W przypadku magazynu danych zmniejszenie rozmiaru jednego atrybutu o jeden bajt może zaowocować znaczącym zmniejszeniem rozmiaru relacji, a co za tym idzie, poprawą efektywności wykonywania zapytań. Jeśli relacja faktów zawiera informacje o dwóch milionach abonentów, z których każdy dzwoni średnio dwa i pół raza dziennie (zakładamy dwuletni horyzont czasowy dla szczegółowych faktów), to w relacji faktów zawartych jest 3.65 miliarda krotek. Zmniejszenie rozmiaru któregokolwiek z atrybutów o dziesięć bajtów spowoduje zmniejszenie relacji o 34 gigabajty.

3.2.5. Wybór pomiędzy kluczami naturalnymi i sztucznymi

Klucze obce występujące w relacji faktów mogą być naturalne (każdy klucz reprezentuje jednoznaczny identyfikator obiektu w świecie rzeczywistym), lub sztuczne (każdy klucz jest generowany automatycznie). Kluczami naturalnymi są np. numer PESEL, data lub numer przedstawicielstwa/sklepu. Kluczami sztucznymi są np. identyfikator klienta (wiążący dany fakt z identyfikatorem w relacji wymiaru *Klienci*), identyfikator daty (wiążący dany fakt z konkretną datą przechowywaną w relacji *Czas*) lub identyfikator sklepu.

Użycie kluczy naturalnych może okazać się bardzo korzystne. Jeśli zapytanie odnosi się bezpośrednio do wartości klucza, to do skonstruowania odpowiedzi wystarczy odczyt samej relacji faktów, bez konieczności dokonywania połączenia z relacją wymiarów. Z drugiej strony, jeśli jakiś identyfikator naturalny ulega zmianie, wówczas taka zmiana musi zostać odzwierciedlona w relacji faktów. Koszt wykonania modyfikacji relacji faktów może być bardzo wysoki i takiej operacji należy za wszelką cenę unikać.

Jeżeli projektant jest absolutnie pewien, że wartości identyfikatorów nie ulegną w przyszłości żadnym zmianom, powinien użyć kluczy naturalnych. W przeciwnym wypadku powinien używać kluczy sztucznych.

3.2.6. Modelowanie czasu w relacji faktów

Czas można modelować podobnie jak wszystkie inne wymiary, tj. poprzez składowanie w relacji faktów sztucznego klucza do relacji wymiarów, w której przechowywane są fizyczne daty. Jednakże bardziej efektywną metodą jest składowanie czasu za pomocą klucza naturalnego bezpośrednio w relacji faktów. Istnieją w ogólności trzy metody składowania czasu w relacji faktów.

- Składowanie fizycznej daty

Ta metoda zakłada składowanie w relacji faktów atrybutu *Data*. Jest bardzo efektywna zarówno w przypadku zapytań odwołujących się do dokładnego znacznika czasowego (ang. *timestamp*), jak i samej daty z dokładnością do dnia. W przypadku dat nie zachodzi niebezpieczeństwo modyfikacji wartości atrybutu w przyszłości, zaś zysk wynikający z przyspieszenia wykonywania zapytań (nie trzeba wykonywać dodatkowego połączenia z relacją wymiarów *Czas*) jest ogromny. Należy przy tym pamiętać, że w środowisku magazynów danych znakomita większość zapytań dotyczy, w ten czy inny sposób, czasu.

- Składowanie przesunięcia od właściwego początku relacji

Jest bardzo prawdopodobne, że relacja faktów podlega partycjonowaniu względem czasu (*patrz 3.3*). Najczęściej poszczególne partycje reprezentują okresy czasowe, np. tydzień, miesiąc lub kwartał. Modelując czas w relacji faktów można wykorzystać fakt partycjonowania poprzez składowanie w tej relacji przesunięcia danego faktu względem właściwego początku partycji. Na przykład, jeżeli relacja faktów jest podzielona na miesięczne partycje, a fakt miał miejsce 9-go dnia miesiąca, to w odpowiedniej partycji zostanie umieszczona liczba 8 (zakładamy, że pierwszy dzień miesiąca odpowiadającego danej partycji ma indeks 0). Taka reprezentacja niesie ze sobą ogromną oszczędność przestrzeni dyskowej, ponieważ teraz do reprezentowania każdej daty w zupełności wystarczą dwa bajty. Co więcej, właściwa data początkowa danej partycji nie musi być nigdzie przechowywana, ponieważ może być na stałe zaszyta w nazwie partycji, np. *Abonenci_lipiec_2000*.

Wadą tego rozwiązania jest fakt, że każde zapytanie operujące na datach fizycznych musi być wpierw poddane konwersji do formy uwzględniającej przesunięcie. Każda taka transformacja niesie ze sobą dodatkowy koszt. Poza tym niektóre narzędzia dostępu mogą być niekompatybilne z takim modelem. W takim wypadku rozwiązaniem może być zdefiniowanie na partycji perspektywy, która dokona odpowiedniej konwersji.

- Składowanie zakresów dat

W przypadku niektórych relacji faktów poszczególne krotki takiej relacji reprezentują fakty, których ważność rozciąga się na pewien okres. Jeśli relacja faktów opisuje stan magazynu, to najprawdopodobniej tylko mała część produktów jest kupowana danego dnia, zaś większość produktów nie zmienia swego stanu magazynowego. W takim wypadku bardziej opłacalne jest odnotowywanie tylko tych faktów (pozycji magazynowych), które uległy zmianom. Każda krotka w relacji faktów opisuje stan jakiegoś produktu, który obowiązywał od dnia *Data_od* do dnia *Data_do*. Jeśli ilość jakiegoś produktu nie ulega zmianie, to zamiast wstawiać nową krotkę do relacji faktów, zwiększa się o 1 wartość atrybutu *Data_do*. Taka technika może prowadzić do znacznych oszczędności przestrzeni dyskowej. Jeśli każdego dnia sprzedawanych jest 10% produktów z katalogu, to wykorzystywanie kodowania dat za pomocą zakresów obowiązywania powoduje zmniejszenie relacji faktów do około 10% objętości początkowej relacji (ponieważ pojawia się dodatkowy atrybut *Data_do*).

To podejście jest obciążone tymi samymi wadami, co składowanie przesunięcia względem początku partycji. Zapytania stają się bardziej skomplikowane, a niektóre narzędzia dostępu nie będą mogły sobie poradzić z taką organizacją składowania czasu. Aby temu zapobiec, możliwe jest wykorzystanie dodatkowej relacji zawierającej krotkę dla każdej kolejnej daty. Relacja faktów jest łączona z relacją zawierającą daty w celu utworzenia produktu kartezjańskiego. Niestety, utworzenie produktu kartezjańskiego jest kosztowne i wymaga przestrzeni tymczasowej. Jeżeli większość zapytań wykorzystuje taką perspektywę, składowanie zakresów dat może okazać się złym wyborem.

Generalnie zaleca się, aby wykorzystywać wyżej opisaną metodę tylko w tych przypadkach, w których narzędzia dostępu mogą bezpośrednio wykorzystywać zakresy dat i nie wymagają tworzenia produktu kartezjańskiego.

3.3. Partycjonowanie faktów

Partycjonowanie polega na dzieleniu logicznej encji na mniejsze podencje. Dzielenie dużych relacji na podrelacje ma na celu m.in.:

- **Zwiększenie efektywności zapytań**: pojedyncze zapytanie wykonuje się dużo szybciej, jeśli zamiast jednej ogromnej relacji musi odczytać zbiór małych partycji. Wiąże się z tym jednak problem automatycznego kierowania zapytań do odpowiednich partycji.
- **Ułatwienie zarządzania relacjami**: relacje faktów często rozrastają się do monstrualnych rozmiarów. Administrowanie relacją o rozmiarze rzędu setek gigabajtów może być trudne lub wręcz niemożliwe. Dotyczy to zmian w strukturze relacji, zakładania indeksów, modyfikowania podzbiorów krotek, itp.
- **Ułatwienie archiwizowania i odtwarzania**: archiwizowanie relacji faktów zawierającej wszystkie aktualne dane może być technicznie niewykonalne bez wyłączenia magazynu danych. Przy wykorzystaniu partycjonowania administrator może pozostawić aktywną tylko aktualną partycję, zaś wszystkie pozostałe oznaczyć jako partycje „tylko do odczytu" i stopniowo je archiwizować.

Istnieje kilka podstawowych metod partycjonowania relacji faktów.

3.3.1. Partycjonowanie według czasu na segmenty o jednakowym rozmiarze

To podstawowa metoda partycjonowania polegająca na podziale relacji faktów na segmenty odpowiadające takim samym przedziałom czasowym. Rozmiar przedziału zależy od charakteru magazynu danych i jego przeznaczenia. Jeżeli większość zapytań kierowanych do magazynu dokonuje raportów miesięcznych, to relacja faktów powinna być podzielona na partycje odpowiadające poszczególnym miesiącom. Należy przy tym pamiętać, że ogólna liczba partycji nie powinna przekroczyć kilkudziesięciu, ponieważ zarządzanie relacją podzieloną na zbyt dużą liczbę segmentów może stać się kosztowne.

3.3.2. Partycjonowanie według czasu na segmenty o różnych rozmiarach

Jeżeli starsze dane są odczytywane rzadziej niż nowsze, to dobrą metodą jest podzielenie relacji faktów na segmenty o różnych rozmiarach. Najczęściej tworzy się w takim wypadku kilka małych partycji dla danych z ostatnich kilku miesięcy, kilka średnich partycji dla mniej aktualnych danych sprzed pół roku, oraz zbiór dużych partycji dla rzadko odczytywanych danych pochodzących sprzed kilku lat. Główną zaletą tego podejścia jest to, że najczęściej wykorzystywane dane przechowywane są w małych partycjach, co wydatnie przyspiesza ich odczyt. Poza tym ogólna liczba partycji jest utrzymywana na niskim poziomie. Podstawową wadą tej metody jest fakt, że w regularnych odstępach czasu profil partycjonowania się zmienia i administrator musi przenosić bardzo duże ilości danych pomiędzy partycjami, co może okazać się bardzo kosztowne. Tego typu schemat jest zalecany w przypadku, gdy wymagania stawiane przed magazynem danych stanowią mieszankę aktywnej analizy najnowszych danych i eksploracji dużych wolumenów danych historycznych.

3.3.3. Partycjonowanie według innego wymiaru

Czas nie jest jedynym wymiarem, według którego może przebiegać partycjonowanie. Weźmy jako przykład magazyn danych dużej korporacji, której oddziały są rozproszone geograficznie. Jeżeli każdy z oddziałów odczytuje przede wszystkim informacje dotyczące regionu właściwego dla siebie, to bardziej efektywną metodą partycjonowania byłby podział relacji na partycje odpowiadające poszczególnym regionom. W ten sposób każdy z oddziałów unikałby odczytywania danych dotyczących innych oddziałów. Decydując się na takie partycjonowanie relacji faktów projektant musi się upewnić, że wymiar będący podstawą podziału relacji faktów nie ulegnie w przyszłości modyfikacji. Restrukturyzacja schematu partycjonowania związana z modyfikacją wymiaru może być bardzo kosztowna, jeśli nie niemożliwa. Jako wskazówkę można przyjąć tu

zasadę, że partycjonowanie odbywa się tylko według czasu, chyba, że projektant jest absolutnie pewny, że podstawa partycjonowania nie zmieni się w przyszłości.

3.3.4. Partycjonowanie według rozmiaru relacji

W przypadku, gdy żaden wymiar nie jest odpowiedni aby stanowić podstawę do podziału relacji, należy rozważyć schemat partycjonowania według rozmiaru relacji. W takim przypadku za każdym razem, gdy rozmiar relacji faktów zbliża się do pewnego predefiniowanego maksimum, tworzona jest nowa partycja, która od tego momentu staje się partycją aktywną. Partycjonowanie według rozmiaru relacji jest złożone, uciążliwe w zarządzaniu i wymaga specjalnych metadanych, pozwalających na określenie, które krotki są przechowywane w poszczególnych partycjach.

3.3.5. Partycjonowanie pionowe

Partycjonowanie pionowe polega na podziale relacji zawierającej wiele atrybutów na zbiór relacji, z których każda posiada podzbiór atrybutów wyjściowych. Partycjonowanie pionowe jest przydatne w przypadku, gdy projektant chce odseparować zbiór rzadko odczytywanych atrybutów relacji faktów od atrybutów często odczytywanych. Jak już wspomniano wcześniej każde zmniejszenie rozmiaru krotki w relacji faktów ma niebagatelny wpływ na efektywność wykonywania zapytań.

Partycjonowanie pionowe może być wykonane jako **normalizacja** (ang. *normalization*), lub jako **podział krotek** (ang. *row splitting*). Normalizacja jest standardową metodą organizacji baz danych i pozwala na kondensowanie powtarzających się wartości do pojedynczych krotek. W przypadku magazynów danych często obserwuje się tendencję odwrotną, tzn. przeprowadza się denormalizację, w wyniku której zwiększa się zajętość przestrzeni dyskowej, ale jednocześnie można uniknąć kosztownych operacji łączenia relacji podczas wykonywania zapytań. W ogólności projektanci magazynów danych powinni unikać procesu normalizacji relacji. W przypadku podziału krotek zachowywane jest odwzorowanie jeden-do-jeden pomiędzy partycjami. Głównym celem podziału wierszy jest przyspieszenie wykonywania zapytań poprzez zmniejszenie rozmiaru relacji faktów. Oddzielone atrybuty nadal mogą być odczytywane w nowo utworzonej partycji. Podział krotek może okazać się efektywną metodą partycjonowania, pod warunkiem wszakże, że nie występuje konieczność dokonywania połączenia podzielonych części relacji.

3.4. Projektowanie relacji wymiarów

Relacje wymiarów mogą być zaimplementowane na kilka sposobów. Z racji ogólnej charakterystyki informacji referencyjnych relacje wymiarów są zawsze wielokrotnie mniejsze od relacji faktów, stąd ich restrukturyzacja nie jest specjalnie kosztowna. Trzeba jednak pamiętać o tym, że prawidłowo zaprojektowana struktura informacji referencyjnych może znacznie zwiększyć efektywność wykonywania zapytań w magazynie danych.

3.4.1. Wymiary gwiaździste

Wymiary gwiaździste są podstawą konstrukcji schematu gwiaździstego. Zwiększają prędkość wykonywania zapytań poprzez denormalizację wszystkich informacji referencyjnych danego wymiaru do jednej relacji. Większość zapytań analizuje fakty po uprzednim ograniczeniu relacji faktów poprzez nałożenie licznych ograniczeń na relację wymiarów (np. zapytanie sumujące zyski w grupie klientów młodych, mieszkających w dużych miastach i posiadających „żółty" abonament). Ponieważ zapytanie ogranicza zbiór klientów według różnych kryteriów (grupa wiekowa, miejsce zamieszkania, rodzaj abonamentu), można je przyspieszyć poprzez włączenie wszystkich atrybutów dotyczących wymiaru *Klient* do jednej relacji. Wadą tego rozwiązania jest znaczące zwiększenie rozmiaru relacji wymiaru. Jeżeli niektóre z atrybutów wymiaru są odczytywane bardzo rzadko, to koszt zwiększenia rozmiaru relacji może być większy niż zysk wynikający z przyspieszenia wykonywania zapytań. Jeżeli atrybut wymiaru jest używany rzadko, to należy pozostawić go w postaci schematu płatka śniegu.

3.4.2. Wymiary typu „płatek śniegu”

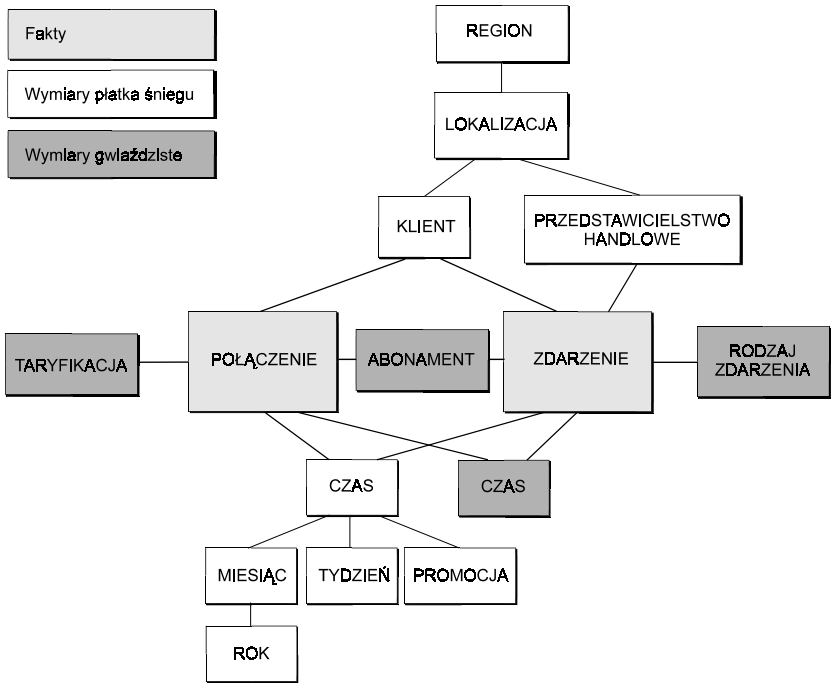
Nie wszystkie encje można sprowadzić do wymiarów gwiaździstych. W pewnych przypadkach encje składające się na wymiar są połączone ze sobą związkami typu wiele-do-wiele, których nie należy denormalizować. Często dla jednej encji występuje wiele różnych hierarchii, reprezentujących różne punkty widzenia na jedne i te same dane. Przykładowo, przedstawicielstwa handlowe firmy X podlegają pod hierarchię opisującą geograficzną lokalizację placówek. Równolegle do podstawowej hierarchii użytkownicy korzystają z dodatkowych hierarchii, klasyfikujących przedstawicielstwa pod względem: typu lokalizacji (supermarket, centrum handlowe, wolnostojące, itp.), rozmiaru placówki, ofert i promocji oferowanych przez placówkę, itd. W takim przypadku denormalizacji do wymiaru gwiaździstego powinna podlegać ta hierarchia, z której korzysta najwięcej zapytań, zaś pozostałe hierarchie powinny pozostać w postaci płatków śniegu. Jeżeli w przyszłości zmieni się profil zapytań (czyli inna hierarchia stanie się „najpopularniejsza”), to nie należy usuwać aktualnie wykorzystywanego wymiaru gwiaździstego (koszt modyfikacji zapytań może być znaczny), lecz wzbogacić go o te atrybuty, które opisują nową hierarchię.

3.4.3. Wymiary zmieniające się w czasie

Informacje referencyjne, w przeciwieństwie do faktycznych, ulegają częstym zmianom i modyfikacjom. Wynika to z tego, że organizacje zmieniają swój punkt widzenia na posiadane dane w zależności od aktualnych warunków ekonomicznych. Co więcej, same przedsiębiorstwa ulegają wewnętrznym przeobrażeniom i reorganizacjom, co znajduje odbicie w modyfikacji wymiarów w magazynie danych (wymiary opisujące strukturę organizacyjną firmy, wymiary opisujące przydział produktów do grup, itp.). Czasami pojawia się też potrzeba wykonywania zapytań, które operują jednocześnie na faktach grupowanych według aktualnych i przeszłych hierarchii (tzw. zapytania „jak jest, jak było”). W takim wypadku niezbędne staje się przechowywanie zakresów dat w relacjach wymiarów. Wówczas przyporządkowanie krotki wymiaru do pewnej generalizacji jest właściwe tylko w przedziale czasowym *Data_od, Data_do*. Jeżeli jakakolwiek wartość w relacji wymiaru ulega zmianie, to jej poprzednie przyporządkowanie jest zamykane poprzez uzupełnienie wartości *Data_do* i wstawiana jest nowa krotka, opisująca nowe przyporządkowanie danej wartości wymiaru.

3.4.4. Wymiary hybrydowe

W poprzednich akapitach przedstawiono dwie podstawowe organizacje relacji wymiarów: schemat gwiaździsty i schemat typu „płatek śniegu". W rzeczywistych projektach rzadko udaje się wykorzystywać oba schematy w czystej postaci. Najczęściej projektanci wybierają organizację hybrydową (ang. *starflake schema*). W ramach takiej organizacji podstawowa część informacji referencyjnej jest przedstawiona w postaci gwiaździstej (jako zdenormalizowane relacje), a część pomocnicza w postaci „płatka śniegu" (jako znormalizowane hierarchie). Przykład schematu hybrydowego przedstawiono poniżej na rysunku 2.



Rys. 2. Przykład schematu hybrydowego

W miarę eksploatacji magazynu danych wspólne części relacji wymiarów powinny ulegać stopniowemu zanikowi. Wpływa na to lepsze zrozumienie potrzeb użytkownika oraz krystalizacja profilu wykorzystania magazynu. Po pewnym czasie projektant dysponuje wystarczającą wiedzą aby dokonać restrukturyzacji niektórych wymiarów i sprowadzenia ich do bardziej „poprawnej" formy.

3.4.5. Partycjonowanie wymiarów

W pewnych przypadkach relacja wymiaru może urosnąć do rozmiaru, w którym niezbędne stanie się partycjonowanie tego wymiaru. Taka sytuacja może powstać dla wymiaru, który często podlega zmianom i dla którego istnieje wymóg przechowywania wszystkich poprzednich wersji wymiaru w celu dokonywania porównań. Zbyt duży rozmiar relacji wymiaru może bardzo negatywnie wpłynąć na czas wykonywania zapytań.

Podstawą partycjonowania relacji wymiaru powinien być jeden z atrybutów grupujących dla tego wymiaru. Jeśli partycjonowaniu podlega katalog oferowanych produktów o dużym rozmiarze, to podstawą partycjonowania powinna być kategoria produktu. Oczywiście, nie można tworzyć osobnej partycji dla każdego produktu. Podstawa partycjonowania powinna być wybrana na odpowiednim poziomie hierarchii klasyfikacji produktu, tak, aby łączna liczba utworzonych partycji nie przekraczała 50. W rzeczywistości przypadki, w których partycjonowanie wymiaru jest konieczne, są niezwykle rzadkie. Bardziej prawdopodobne jest, że duża relacja wymiaru zawiera ukryte w niej fakty, które powinny być wyodrębnione.

3.5. Projektowanie relacji zbiorczych

Agregacja stanowi bardzo istotny element magazynów danych. Pozwala na efektywne wykonywanie złożonych i kosztownych zapytań w rozsądnym czasie i bez potrzeby znaczących inwestycji w zasoby sprzętowe. Poprawne zaprojektowanie strategii agregacji jest jednak trudne: zbyt wiele agregatów odbije się negatywnie na kosztach zarządzania i pielęgnowania magazynu danych, zbyt mało agregatów nie pozwoli na efektywne wykonywanie zapytań. Ogólnie rzecz biorąc, należy przyjąć zasadę 70-30: w dobrze zaprojektowanym magazynie danych 70% zapytań wykonuje się z prędkością zadowalającą użytkowników, zaś przyspieszenie pozostałych 30% musi odbyć się kosztem znacznych inwestycji w moc obliczeniową sprzętu, na którym działa magazyn danych.

3.5.1. Czym jest agregacja?

Agregacja to wstępne dokonywanie obliczeń, tworzenie danych zbiorczych oraz ich przechowywanie w celu późniejszego wykorzystania. Agregaty nie niosą ze sobą żadnych nowych informacji w tym sensie, że wszystkie obliczenia bazują na danych obecnych w relacjach faktów i wymiarów. Większość strategii agregacji wykorzystuje fakt, że bardzo wiele zapytań operuje na wąskich podzbiorach faktów wyznaczanych przez specyficznie pogrupowane wartości wymiarów. Aby efektywnie realizować proces wspierania decyzji magazyn danych musi dostarczać użytkownikom informacji na odpowiednim poziomie szczegółowości. Bezpośrednia analiza relacji faktów nie pozwala na wyciąganie żadnych wniosków na temat ogólnych trendów i regularności występujących w danych. Dopiero spojrzenie na fakty „z dystansu", np. na poziomie całej grupy klientów lub regionu geograficznego, pozwala na dostrzeżenie istotnych prawidłowości.

Podstawową zaletą stosowania agregacji jest przyspieszenie wykonywania zapytań. Złożone zapytanie odczytuje wyniki skomplikowanych i czasochłonnych obliczeń bezpośrednio z relacji zbiorczej i nie musi tracić czasu na powtarzanie tych obliczeń. Odbywa się to kosztem dokonania wcześniejszych obliczeń i składowania wyniku w relacji zbiorczej oraz, w niektórych przypadkach, pielęgnowania wyliczonej wartości. Widać też wyraźnie, że zyski z agregacji mogą okazać się krótkoterminowe, ponieważ jeśli zmieni się profil zapytań i dana wartość zbiorcza przestanie być używana, to jej dalsze przechowywanie w magazynie danych okaże się bezcelowe. Ostatnia uwaga pokazuje, że projektowanie relacji zbiorczych nie jest czynnością jednorazową. W trakcie życia magazynu danych administrator powinien nieustannie monitorować profile zapytań i, w przypadku odkrycia takiej konieczności, dodawać bądź usuwać pewne relacje zbiorcze.

Jak już powiedziano, zysk z wcześniejszego wyliczania wartości zbiorczych polega na przesunięciu kosztów przetwarzania w czasie, dzięki czemu zmniejsza się koszt wykonywania zapytań. Oczywiście, dla każdej relacji zbiorczej można zdefiniować perspektywę udostępniającą te same dane. W przypadku perspektywy jednak zysk będzie znikomy, ponieważ obecność perspektywy w żaden sposób nie wpłynie na czas wykonania zapytania. Co więcej, wartości wyliczonej za pomocą perspektywy nie można powtórnie wykorzystać i w przypadku powtórzenia zapytania całe przetwarzanie wykona się raz jeszcze od początku.

3.5.2. Które relacje zbiorcze powinny być tworzone?

Wybór informacji, które powinny podlegać agregacji, jest trudnym procesem, ponieważ wraz ze zmianą profilu zapytań zmieniają się także wymagania dotyczące agregatów. W pierwszym kroku trzeba określić początkowy zbiór informacji, które trafią do relacji zbiorczych. Jeżeli magazyn danych jest budowany na bazie starszego systemu komputerowego, to dobrym początkiem jest agregacja tych samych informacji, które podlegały agregacji w poprzednim systemie.

Podstawową metodą odkrywania istotnych agregacji jest stworzenie tabeli ze wszystkimi poziomami wszystkich kluczowych wymiarów. Z tej tabeli należy odczytać wszystkie możliwe kombinacje wymiarów i zbadać, które z tych kombinacji będą odpowiadały zapytaniom użytkowników. Poniżej przedstawiono tabelę zawierającą przekrój poziomów hierarchii dla wymiarów *Taryfikacja*, *Klient* i *Czas*.

Tabela 1. Przykładowy przekrój poziomów hierarchii

Taryfikacja	Klient	Czas
Strefa czasowa	Nazwa	Dzień
	Lokalizacja	Tydzień
	Region	Miesiąc
	Kraj	Rok

Z powyższej tabeli można wybrać kombinację *Strefa czasowa-Region-Miesiąc* i utworzyć dla niej odpowiednie relacje zbiorcze, a następnie zapełnić te relacje wyliczonymi wartościami. Taka relacja zbiorcza będzie przechowywać dane o czasie trwania i wartościach rozmów we wszystkich strefach czasowych, zsumowanych dla poszczególnych regionów i miesięcy.

Relacje zbiorcze można podzielić na trzy klasy:

- Wysoki poziom agregacji: udostępniają szersze spojrzenie na całokształt danych, np. sumaryczna sprzedaż przedsiębiorstwa z podziałem na tygodnie i produkty,
- Średni poziom agregacji: bardziej szczegółowe dane na temat konkretnego regionu lub grupy produktów, np. sumaryczna sprzedaż z podziałem na tygodnie i sklepy,
- Niski poziom agregacji: szczegółowe informacje, podobne do faktów, np. dzienna sprzedaż z podziałem na produkty i sklepy.

Relacje zbiorcze z wysokim poziomem agregacji są zazwyczaj małe i powinny zawierać wszystkie konieczne agregaty wraz ze zdenormalizowanymi wymiarami. Relacje zbiorcze z niskim poziomem agregacji są bardzo duże i swą charakterystyką przypominają relacje faktów. W trakcie ich konstruowania należy przestrzegać reguł dotyczących konstrukcji relacji faktów. Relacje zbiorcze ze średnim poziomem agregacji są relacjami „granicznymi", w których przypadku należy rozsądnie balansować między rozmiarem takiej relacji a jej zawartością. Jeśli rozmiar takiej relacji przekroczy 1-2 GB, to prawdopodobnie dane w niej zawarte są zbyt szczegółowe.

3.5.3. Projektowanie relacji zbiorczych

Istotą wykorzystania relacji zbiorczych jest zmniejszenie wolumenu odczytywanych danych poprzez składowanie w relacji zbiorczej maksymalnej liczby wartości częściowych. Nie chodzi tu tylko o zapisywanie do relacji wartości zbiorczych, ale np. o składowanie w relacji zbiorczej informacji referencyjnych celem uniknięcia kosztownych operacji połączenia. W ogólności proces konstruowania relacji zbiorczych jest podobny do procesu konstruowania relacji faktów, z uwzględnieniem pewnych istotnych szczegółów. Składa się z następujących kroków.

- Wybór właściwych wymiarów do agregacji

Relacje zbiorcze są naturalnym rozszerzeniem relacji faktów. Zapytanie, pierwotnie skierowane do relacji faktów, powinno wykorzystać relację zbiorczą bez konieczności modyfikowania treści zapytania, w sensie wykorzystania tych samych wymiarów i faktów. Oznacza to, że w stosunku do relacji faktów relacja zbiorcza powinna zachowywać tę sama strukturę wymiarów (poza agregowanym wymiarem), oferując jednocześnie sumaryczne wartości pewnych faktów. Relacja zbiorcza prezentująca sumaryczny czas rozmów dla poszczególnych rodzajów taryfikacji i rodzajów abonamentu musi zawierać identyfikatory taryfikacji oraz abonamentu. Atrybut *Data* jest niepotrzebny, ponieważ miesiąc, którego dotyczy agregacja, jest zaszyty w nazwie relacji zbiorczej. W tym wypadku oprócz zmniejszenia liczby krotek w relacji zbiorczej zmniejsza się rozmiar każdej z krotek. Jeżeli relacja zbiorcza przechowuje dane o sumarycznym dziennym czasie rozmów w całym kraju, to atrybut *KlientId* staje się niepotrzebny, ponieważ sumowanie pomija podmiotowość poszczególnych abonentów.

Drugą możliwością jest przechowywanie w każdej krotce wartości zagregowanych na pewnym poziomie hierarchii wymiaru. W takim wypadku agregowany wymiar powinien pozostać w relacji zbiorczej. Jeśli w poprzednim przykładzie pojawiłaby się konieczność obliczania sumarycznego

dziennego czasu rozmów na poziomie regionalnym, to wówczas wymiar opisujący lokalizację powinien być na powrót włączony do relacji zbiorczej. Nie powinien być to jednak atrybut *KlientID*, który jest zbyt szczegółowy, lecz atrybut *RegionID*.

- Agregacja wielu wartości

Celem tego kroku jest określenie, które wartości powinny być agregowane i zapisywane do relacji zbiorczych. Robi się to poprzez analizę zapytań i identyfikację agregowanych wartości. Rzadko kiedy użytkownika interesuje tylko jedna wartość zbiorcza. Najczęściej zapytania odczytują zbiór agregatów obliczony dla tych samych wymiarów. Analizując relację zbiorczą zawierającą dane o rozmowach wykonanych w przeciągu tygodnia użytkownik najprawdopodobniej zapyta o: sumaryczny czas połączeń, największą, najmniejszą i średnią liczbę połączeń dla poszczególnych typów abonamentów, itp. Jeżeli większość zapytań odczytuje więcej niż jedną wartość agregowaną wzdłuż tych samych wymiarów, to wartości zbiorcze dla tych agregatów powinny być przechowywane w jednej relacji zbiorczej. Projektanci mają tendencję do zawyżania liczby przechowywanych agregatów („a nuż się przyda w przyszłości"), jednak pamiętać trzeba, że każdy dodatkowy atrybut w relacji zbiorczej wpływa negatywnie na prędkość wykonywania się zapytań. Najczęściej w zupełności wystarcza kilka atrybutów.

- Agregacja wielu faktów w jednej relacji zbiorczej

Czasami okazuje się, że jedno zapytanie odczytuje wiele różnych faktów, agregowanych na tych samych podstawach (tj. na tym samym poziomie hierarchii wymiarów). Tego typu zapytania są częste w przypadku analizy porównawczej, np. wartości rozmów w różnych okresach czasowych, klientów przychodzących do i odchodzących z firmy, itp. Jako przykład weźmy zapytanie analizujące liczbę podpisanych umów w ramach poszczególnych tygodni. Oprócz podstawowych informacji o liczbie umów użytkownik dodatkowo pyta się o: liczbę innych typów umów podpisanych w tym samym okresie, liczbę klientów, która w tym samym okresie zrezygnowała z abonamentu, przewidywaną liczbę umów podpisanych w ramach promocji w danym tygodniu lub zyski osiągnięte w danym tygodniu. Takie zapytanie odczyta cztery relacje: o rozmowach, o promocjach, o przewidywanych obrotach oraz o zdarzeniach (podpisanie lub anulowanie umowy). W rezultacie jedno zapytanie zostanie wykonane jako cztery różne zapytania, z których każde dokona tej samej agregacji dla czterech różnych faktów. Jeżeli podobne zapytania pojawiają się często, to należy rozważyć stworzenie relacji zbiorczej przechowującej agregaty pochodzące z wielu różnych relacji faktów. Należy przy tym pamiętać, że takie rozwiązanie jest efektywne tylko dla agregatów często wykorzystywanych wspólnie. Jeżeli tylko nieliczne zapytania wykorzystują kombinację agregatów różnych faktów, to koszt pielęgnacji takiej relacji zbiorczej przekroczy ewentualny zysk wynikający z przyspieszenia zapytań.

- Wybór poziomu agregacji

Dokonanie agregacji na określonym poziomie hierarchii wymiarów powoduje, że bardziej szczegółowe informacje dotyczące tego wymiaru stają się niedostępne. Z drugiej strony zaniżanie poziomu agregacji powoduje wzrost liczby relacji zbiorczych i zwiększenie kosztu pielęgnacji tych relacji. Dla systemów bankowych częste są zapytania dokonujące agregacji w skali miesiąca (raporty o stanie konta, naliczanie odsetek, rat kredytów, itp.). Stworzenie relacji zbiorczej na poziomie miesiąca powoduje jednak, że obliczenie tygodniowego salda dla konkretnego klienta musi się odbyć na podstawie relacji faktów i nie można do tego celu wykorzystać relacji zbiorczej. Jednak jeśli została zdefiniowana relacja zbiorcza z agregatami na poziomie tygodnia, wyliczenie podsumowań miesięcznych wymaga dokonania agregacji średnio czterech wierszy w relacji zbiorczej. Ta metoda jest szczególnie efektywna, jeżeli dokonanie agregacji „w locie", jak w powyższym przykładzie, wymaga przeliczenia niewielkiej liczby krotek. Tworząc relacje zbiorcze projektanci powinni rozważyć budowanie agregatów o poziom niżej od poziomu wymaganego przez użytkowników, co przy minimalnym koszcie pozytywnie wpływa na elastyczność i funkcjonalność magazynu danych.

- Wybór stopnia włączenia informacji referencyjnych do relacji zbiorczych

Relacje zbiorcze są często modyfikowane i odświeżane. W szczególnym przypadku relacja zbiorcza może być odświeżana po każdym dodaniu nowych faktów do relacji faktów. To sprawia, że w relacjach zbiorczych należy zawsze korzystać z kluczy naturalnych. Podstawowym argumentem przemawiającym przeciwko stosowaniu kluczy naturalnych w relacjach faktów było niebezpieczeństwo konieczności dokonywania restrukturyzacji relacji faktów po zmianie wartości kluczy naturalnych. Ponieważ relacje zbiorcze są nieustannie modyfikowane, powyższy argument traci moc. Samo stosowanie kluczy naturalnych nie chroni jednak przed koniecznością łączenia relacji zbiorczych z relacjami wymiarów. W wielu przypadkach takie operacje mogą być bardzo kosztowne i czasochłonne, szczególnie dla dużych relacji zbiorczych. Jeżeli zapytanie wymaga dostępu zarówno do relacji wymiarów jak i do relacji faktów, to dobrym rozwiązaniem projektowym jest zaszycie niektórych atrybutów wymiaru w relacji zbiorczej. Takie rozwiązanie wyeliminuje potrzebę dokonywania łączenia relacji i pozwala odpowiedzieć na zapytanie tylko na podstawie relacji zbiorczej. Jeśli np. zapytanie odczytuje relację zawierającą sumaryczną tygodniową ilość rozmów w poszczególnych strefach czasowych, zaś w zapytaniu użytkownicy wybierają tylko pewne typy abonamentu (co wymaga odczytania relacji wymiaru *Abonament*), to należy rozważyć włączenie dodatkowego atrybutu do relacji zbiorczej. Ta technika powoduje znaczne zwiększenie rozmiaru poszczególnych krotek i nie powinna być stosowana do relacji zbiorczych o niskim poziomie agregacji (np. do relacji zawierającej dzienne podsumowania sprzedaży).

- Modelowanie czasu w relacjach zbiorczych

Zdecydowanie najlepszą metodą modelowania czasu w relacjach zbiorczych jest przechowywanie dat fizycznych bezpośrednio w relacji. Przechowywanie dat w postaci przesunięcia względem początku relacji lub jako zakresu dat jest nieefektywne. Koszt związany z przeliczaniem przesunięcia na konkretną datę w przypadku małych i średnich relacji zbiorczych jest znaczny i nie posiada żadnych zalet. W przypadku składowania zakresów dat aktualny jest problem narzędzi dostępu, za pomocą których użytkownicy odczytują dane. Większość z tych narzędzi nie potrafi wykorzystywać takiej metody składowania dat.

- Indeksowanie relacji zbiorczych

Obecność indeksu założonego na atrybucie wskazuje systemowi zarządzania bazą danych, że zapytania powinny być kierowane do danego atrybutu. W przypadku relacji zbiorczych zaleca się, aby indeksować wszystkie atrybuty, które prawdopodobnie będą odczytywane. Koszt pielęgnacji indeksów nie powinien być wysoki, zważywszy na niewielkie rozmiary relacji zbiorczych. Istnienie indeksów nie rozwiąże wszystkich problemów, ale dla zapytań skoncentrowanych na małym fragmencie relacji zbiorczej (a takie powinny stanowić większość zapytań) wydajnie zwiększy efektywność systemu.

4. Podsumowanie

Technologia magazynów danych znajduje zastosowanie w coraz większej liczbie przedsiębiorstw. Mimo, że magazyny danych są budowane w oparciu o technologię relacyjnych baz danych, proces konstruowania schematu logicznego magazynu danych jest odmienny od metodologii tworzenia schematów logicznych dla systemów obsługi bieżącej. W artykule przedstawiono najważniejsze różnice i pokazano, w jaki sposób wpływają one na konkretne rozwiązania projektowe.

Podstawowym problemem jest nieustanna ewolucja magazynu danych, wynikająca ze zmiennych warunków, w jakich działa przedsiębiorstwo. Magazyny danych muszą być projektowane w taki sposób, aby oferować użytkownikom elastyczność, łatwość obsługi, wszechstronność i efektywność. Rozmiary magazynów danych oraz specyficzne wymagania, które są związane z charakterystyką zapytań kierowanych do magazynów, powodują, że konstrukcja schematu logicznego magazynu danych jest trudna.

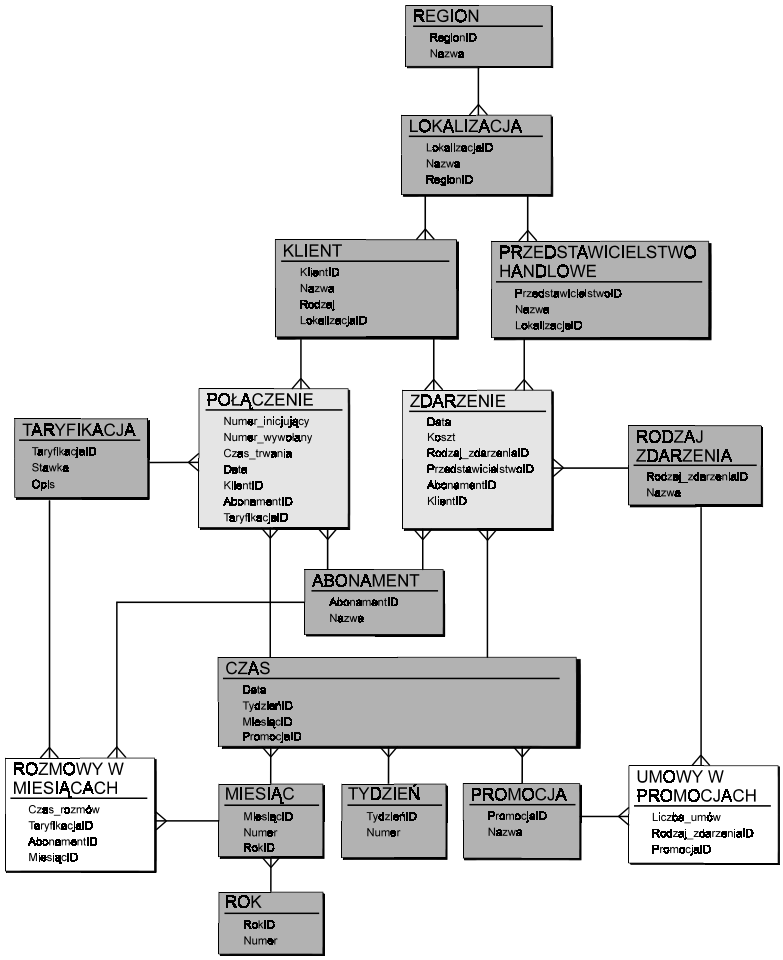
W artykule przedstawiono metody identyfikacji faktów, wymiarów i informacji zbiorczych. Pokazano również, w jaki sposób można zamodelować podstawowe składowe magazynu danych w relacyjnym systemie baz danych. Autorzy mają nadzieję, że przedstawione uwagi i rozwiązania okażą się przydatne dla wszystkich projektantów i administratorów magazynów danych.

Bibliografia

1. Mattison R., Data Warehousing – Strategies, Technologies, and Techniques, McGraw-Hill, 1996, ISBN 0-07-041034-8
2. Anahory S., Murray D., Data Warehousing in the real world, Addison-Wesley, 1997, ISBN 0-201-17519-3
3. Corey M.J., Abbey M., Abramson I., Taub B., Oracle8 Data Warehousing, McGraw-Hill, 1998, ISBN 0-07-882511-3
4. Connolly T., Begg C., Strachan A., Database Systems – A Practical Approach to Design, Implementation, and Management, Addison-Wesley, 1999, ISBN 0-201-34287-1
5. Jarke M., Lenzerini M., Vassiliou V., Vassiliadis P., Fundamentals of Data Warehouses, Springer Verlag, 2000, ISBN 3-540-65365-1

5. Załącznik A

Rysunek 3 przedstawia przykładowy schemat logiczny magazynu danych dla firmy oferującej usługi w dziedzinie telefonii komórkowej. Relacje *Połączenie* i *Zdarzenie* to relacje faktów. Relacje *Rozmowy w miesiącach* i *Umowy w promocjach* są relacjami zbiorczymi. Pozostałe relacje to relacje wymiarów.



Rys. 3. Przykładowy schemat logiczny magazynu danych