Microsoft®

Microsoft® SQL Server® 2008 Analysis Services Krok po kroku

Scott Cameron,
Hitachi Consulting

przekład: Małgorzata Dąbkowska-Kowalik, Witold Sikorski

Microsoft® SQL Server® 2008 Analysis Services Krok po kroku
Edycja polska Microsoft Press
Original English language edition © 2009 by Hitachi Consulting
Tytuł oryginału: Microsoft® SQL Server® 2008 Analysis Services Step by Step

Polish edition by APN PROMISE Sp. z o.o. Warszawa 2009

APN PROMISE Sp. z o.o., biuro: ul. Kryniczna 2, 03-934 Warszawa
tel. +48 22 35 51 642, fax +48 22 35 51 699
e-mail: mspress@promise.pl

Wszystkie prawa zastrzeżone. Żadna część niniejszej książki nie może być powielana ani rozpowszechniana w jakiejkolwiek formie i w jakikolwiek sposób (elektroniczny, mechaniczny), włącznie z fotokopiowaniem, nagrywaniem na taśmy lub przy użyciu innych systemów bez pisemnej zgody wydawcy.

Microsoft, Microsoft Press, Excel, IntelliSense, Internet Explorer, MSDN, MSN, SharePoint, Silverlight, SQL Server, Visual Studio, Windows oraz Windows Server są zarejestrowanymi znakami towarowymi Microsoft Corporation.

Wszystkie inne nazwy handlowe i towarowe występujące w niniejszej publikacji mogą być znakami towarowymi zastrzeżonymi lub nazwami zastrzeżonymi odpowiednich firm odnośnych właścicieli.

Przykłady firm, produktów, osób i wydarzeń opisane w niniejszej książce są fikcyjne i nie odnoszą się do żadnych konkretnych firm, produktów, osób i wydarzeń. Ewentualne podobieństwo do jakiejkolwiek rzeczywistej firmy, organizacji, produktu, nazwy domeny, adresu poczty elektronicznej, logo, osoby, miejsca lub zdarzenia jest przypadkowe i niezamierzone.

APN PROMISE Sp. z o.o. dołożyła wszelkich starań, aby zapewnić najwyższą jakość tej publikacji. Jednakże nikomu nie udziela się rękojmi ani gwarancji.
APN PROMISE Sp. z o.o. nie jest w żadnym wypadku odpowiedzialna za jakiekolwiek szkody będące następstwem korzystania z informacji zawartych w niniejszej publikacji, nawet jeśli APN PROMISE została powiadomiona o możliwości wystąpienia szkód.

ISBN: 978-83-7541-053-2

Przekład: Małgorzata Dąbkowska-Kowalik, Witold Sikorski
Redakcja: Marek Włodarz
Korekta: Magdalena Kalina-Swoboda
Skład i łamanie: MAWart Marek Włodarz

Spis treści

Podziękowania ..... **vii**

Wprowadzenie ..... **ix**

Jak znaleźć właściwy dla siebie punkt startowy ix
Konwencje i funkcje stosowane w książce x
Wymagania dotyczące sprzętu i oprogramowania xi
Pliki ćwiczeń i konfiguracja SQL Server 2008 xii
Znajdowanie dodatkowej informacji online xv
Wsparcie dla książki xv

Część I Istota analizy biznesowej a Analysis Services

1 Analiza biznesowa: podstawy analizy danych 3

Wprowadzenie do analizy biznesowej 3
Wielowymiarowa analiza danych 5
Atrybuty w analizie danych 5
Hierarchie w analizie danych 10
Wymiary w analizie danych 12
Wymiarowa hurtownia danych 17
Tabela faktów 17
Tabele wymiarów 19
Klucze zastępcze i wolno zmienne wymiary 20
Alternatywne struktury tabel 22
Wielowymiarowa baza danych OLAP 24

2 Omówienie OLAP i Analysis Services 27

Pojęcie OLAP 27
Trwale krótki czas reakcji 28
Zapytania oparte na metadanych 30
Wzory w stylu z arkusza kalkulacyjnego 32
Pojęcie usług Analysis Services 34
Analysis Services i szybkość działania 34
Analysis Services i metadane 35
Analysis Services i platforma Microsoft Business Intelligence 39
Narzędzia Analysis Services 41

3 Dostęp do danych źródłowych 43

Tworzenie rozwiązania BI 43
Tworzenie źródła danych 46

Tworzenie widoku źródła danych 48
Modyfikowanie widoku źródła danych 53

Część II Podstawy projektowania

4 Tworzenie wymiarów 65

Przegląd danych dotyczących wymiarów 66
Tworzenie wymiaru standardowego 70
Wdrażanie bazy danych Analysis Services 74
Modyfikowanie wymiaru 78
Tworzenie wymiaru czasu 89
Tworzenie wymiaru nadrzędny-podrzędny 102

5 Tworzenie kostki 113

Podgląd danych kostki 113
Tworzenie koski za pomocą kreatora 116
Wymiary pełniące role 121
Wdrażanie i przeglądanie kostki 125
Użycie projektanta Cube Designer do modyfikowania kostki 136
Dodawanie do kostki przyjaznych nazw 136
Formatowanie miar 137
Modyfikacja interakcji wymiarów i grup miar 140
Dodawanie miar i grup miar do kostki 143
Tworzenie elementu obliczanego 146
Ponowne wdrożenie i przeglądanie kostki 149

6 Tworzenie zaawansowanych miar i obliczeń 153

Używanie funkcji agregacji 154
Użycie MDX do pobierania wartości z kostki 166
Tworzenie elementów obliczanych 171
Stosowanie formatowania warunkowego 173
Obliczanie udziału 174
Tworzenie elementów obliczanych poza wymiarem Measures 180
Skrypty obliczeniowe 184
Tworzenie kluczowych wskaźników wydajności (KPI) 187

7 Zaawansowane projektowanie wymiarów 193

Korzystanie z wymiarów 194
Tworzenie wymiarów wskaźnikowych 194
Tworzenie wymiaru faktów 201
Tworzenie wymiaru wiele-do-wielu 207

Część III Projektowanie zaawansowane

8 Praca z Account Intelligence ... 215
Projektowanie kostki analizy finansowej ... 216
Praca z Account Intelligence ... 217
Tworzenie wymiaru Account ... 218
Praca z nieaddytywnymi miarami finansowymi ... 227
Tworzenie nieaddytywnego elementu obliczanego ... 227
Praca z wymiarem Scenario ... 231
9 Obsługa konwersji walut i wielu języków ... 235
Obsługa konwersji obcych walut ... 235
Obsługa tłumaczenia języków obcych ... 247
10 Interakcja z kostką ... 253
Implementowanie akcji ... 253
Tworzenie akcji standardowych ... 254
Tworzenie akcji eksploracji szczegółów ... 258
Łączenie z raportami usług Reporting Services ... 263
Użycie zapisu zwrotnego do modyfikacji danych Analysis Services ... 272
Włączanie zapisu zwrotnego wymiaru ... 272
Włączanie zapisu zwrotnego kostki ... 275
11 Pobieranie danych z usług Analysis Services ... 281
Tworzenie perspektyw ... 282
Tworzenie zapytań MDX ... 285
Dostęp do usług Analysis Services za pomocą Excel 2007 ... 290
Tworzenie raportów usług Reporting Services ... 308
12 Implementowanie zabezpieczeń ... 317
Pojęcie ról ... 317
Zabezpieczanie dostępu administracyjnego ... 319
Przyznawanie dostępu administracyjnego na poziomie serwera ... 319
Przydzielanie dostępu administracyjnego na poziomie bazy danych ... 322
Zabezpieczanie dostępu do danych ... 325
Przyznawanie dostępu do kostki ... 325
Ograniczanie dostępu do elementów wymiaru ... 326
Ograniczanie dostępu do komórek ... 334
13 Projektowanie agregacji ... 339
Pojęcie projektu agregacji ... 339
Użycie kreatora Aggregation Design ... 342
Relacje atrybutów, hierarchie zdefiniowane przez użytkownika oraz projekt agregacji ... 349
Projektowanie agregacji ... 355

Zmiana liczby partycji ... 356
Zastosowanie kreatora Usage-Based Optimization Wizard ... 359

Część IV Zarządzanie środowiskiem produkcyjnym

14 Zarządzanie partycjami oraz przetwarzaniem bazy danych ... 369

Praca z magazynami danych ... 370
Pojęcie trybów magazynowania wymiarów ... 371
Tryby magazynowania partycji ... 372
Zmienianie danych w hurtowni ... 375
Zarządzanie przetwarzaniem usług Analysis Services ... 377
Przetwarzanie wymiaru ... 378
Przetwarzanie kostki ... 382
Aktywne buforowanie ... 385
Praca z partycjami ... 391
Pojęcie strategii partycji ... 391
Tworzenie partycji ... 392

15 Zarządzanie wdrażaniem ... 401

Przegląd wdrażania ... 401
Mechanizmy wdrażania ... 402
Wdrażanie za pomocą Business Intelligence Development Studio ... 402
Wdrożenie przy użyciu kreatora Deployment Wizard ... 405
Interfejs użytkownika kreatora Deployment Wizard ... 405
Pojęcie skryptów wdrażania ... 409
Migracja baz danych i przywracanie po awarii ... 411
Odłączanie i dołączanie bazy danych Analysis Services ... 411
Synchronizowanie baz danych ... 413
Tworzenie kopii zapasowej i przywracanie bazy danych ... 413

16 Zaawansowane monitorowanie oraz narzędzia zarządzania ... 417

Monitorowanie Analysis Services przy użyciu Reliability And Performance Monitor ... 418
Monitorowanie usług Analysis Services za pomocą SQL Server Profiler ... 426
Dynamic Management Views w usługach Analysis Services ... 432

Indeks ... 436

O autorze ... 454

Podziękowania

Jestem bardzo wdzięczny za pomoc i wsparcie, jakie podczas pisania książki „SQL Server 2008 Analysis Services Krok po kroku" otrzymałem z Hitachi Consulting. Kevin Davis zarządzał projektem, zapewniał codzienną motywację i wykonał niezwykłą pracę z harmonogramem, koordynując wszystkie wysiłki współautorów i autorów współpracujących oraz komunikację z Microsoft Press. Współautorzy Dave DuVarney, Joe Kasprzak i Bryan Smith oraz autorzy współpracujący Harlan Smith, Phillip Duong, Shay Jones, Robert Brawn i Renee De Voursney hojnie poświęcili swój osobisty czas i zmniejszyli moje obciążenie pracą. Reed Jacobsen, mój wieloletni nauczyciel oraz Stacia Misner, Paul Turley, Richard Osbourn i Ryan Clay wnieśli swoją wiedzę techniczną. Hilary Feier dodawała otuchy. Ona też oraz Carr Krueger, Drew Naukum i Lance Baldwin, zaaranżowali wsparcie ze strony Hitachi Consulting, co dało mi czas na napisanie tej książki. David Han i Mary Gianopoulos wykonywali dodatkowe zarządzanie projektem.

Doceniam też, że firma Microsoft Press dała mi możliwość napisania tej książki i wdzięczny jestem za wsparcie, jakie otrzymałem od zespołu wydawców. Podziękowania otrzymuje Ken Jones, menedżer programu i Sally Stickney, edytor, którzy sponsorowali i inicjowali projekt; Lynn Finnel, edytor projektu, która okazała wiele cierpliwości i prowadziła zespół redaktorów. Składam też podziękowania redaktorom technicznym, którymi byli John Welch, Matthew Roche i Phillippe Freddi, i pomagali mi przy zapewnieniu, aby wyjaśnienia i procedury były poprawne i zrozumiałe oraz edytorom: Megan Smith-Creed, Becka McKay, Julie Hotchkiss i Tom Lewis.

Dziękuję wreszcie moje żonie. Tarya cierpliwie wytrzymała wiele dni, kiedy budzik dzwonił wcześnie rano.

Przepraszam, jeśli kogoś pominąłem. Wszystkim, którzy przyczynili się do powstania tej książki serdecznie dziękuję.

Scott Cameron
Marzec 2009

Wprowadzenie

Microsoft SQL Server 2008 Analysis Services to wielowymiarowy komponent przetwarzania analitycznego online (OLAP) dostępny w Microsoft SQL Server 2008, który integruje dane relacyjne i OLAP na potrzeby rozwiązań analizy biznesowej (Business intelligence, BI). Celem tej książki jest pokazanie sposobów korzystania z narzędzi i funkcji Analysis Services, aby łatwo tworzyć, zarządzać i współdzielić kostki OLAP w ramach organizacji. Ćwiczenia realizowane krok po kroku pomagają w przygotowaniu się do własnego wdrożenia rozwiązań BI.

Pomocą w poznaniu wielu funkcji Analysis Services jest podział tej książki na cztery części. Część I, „Podstawowe informacje o analizie biznesowej (BI) i Analysis Services", omawia BI, analizę wielowymiarową oraz OLAP, a także wyjaśnia, jak usługi Analysis Services implementują korzyści płynące z OLAP. Część II, „Podstawy projektowania,", uczy projektowania źródeł danych, widoków źródeł danych, wymiarów i kostek. W część III, „Projektowanie zaawansowane", pokazano, jak korzystać z wyrażeń wielowymiarowych (Multidimensional Expressions, MDX) i agregować funkcje w celu wykonywania złożonych obliczeń i podsumowań oraz tworzenia kluczowych wskaźników wydajności (key performance indicators, KPI). Ponadto w tej części opisano funkcje Analysis Services służące do zaawansowanego projektowania wymiarów, analizy finansowej, globalizacji i wielu interakcji rozszerzających możliwości analityczne kostek. Poznajemy tam także sposoby tworzenia raportów analitycznych za pomocą Microsoft Office Excel i SQL Server 2008 Reporting Services. W części IV, „Zarządzenie eksploatacją", wyjaśniono, jak kontrolować zabezpieczenia w dostępie do kostek i ograniczać dane dostępne dla określonego użytkownika, jak projektować partycje w celu zarządzania skalowalnością bazy danych oraz jak zarządzać bazami danych Analysis Services i je monitorować.

Jak znaleźć właściwy dla siebie punkt startowy

Książka ta obejmuje pełny cykl życia rozwiązania Analysis Services od projektu do wdrożenia. Osoby odpowiedzialne za wybrany wycinek działań mogą przeczytać tylko rozdziały odnoszące się do ich sytuacji, pomijając pozostałą część książki. Poniższa tabela pomoże w ustaleniu najlepszego punktu wyjścia.

Kim jesteś	Kroki do wykonania
Użytkownik informacji, który korzysta z OLAP przy podejmowaniu decyzji	**1.** Instalacja plików ćwiczeń zgodnie z opisem w punkcie „Instalacja i użytkowanie plików ćwiczeń". **2.** Praca z częściami I i II, aby zaznajomić się z podstawowymi możliwościami Analysis Services.

Kim jesteś	Kroki do wykonania
	3. Przejrzenie ciekawszych rozdziałów z części III, aby zrozumieć dodatkowe funkcje, które mogą spełnić analityczne potrzeby użytkownika.
Analityk BI, który projektuje modele OLAP i prototypy dla BI	**1.** Instalacja plików ćwiczeń zgodnie z opisem w punkcie „Instalacja i użytkowanie plików ćwiczeń”. **2.** Przejrzenie części I, aby zobaczyć pojęcia BI i nauczyć się sposobu implementacji OLAP przez Analysis Services. **3.** Praca z częścią II, aby rozwinąć umiejętności potrzebne do utworzenia prototypowej kostki. **4.** Przejrzenie interesujących rozdziałów z części III i IV, aby poznać zaawansowane funkcje Analysis Services i zrozumieć, w jaki sposób użytkownicy uzyskują dostęp do kostki oraz nauczyć się zarządzania kostką po wdrożeniu jej do eksploatacji.
Administrator, który utrzymuje zasoby serwera lub procesy migracji produkcji	**1.** Instalacja plików ćwiczeń zgodnie z opisem w punkcie „Instalacja i użytkowanie plików ćwiczeń”. **2.** Przejrzenie części I, II i III, aby zrozumieć funkcje włączone do Analysis Services. **3.** Praca z częścią IV, aby nauczyć się sposobu zarządzania kostką i zabezpieczenia dostępu do kostki i jej zawartości na serwerze, a także konfigurowania, monitorowania i zarządzania komponentami oraz wydajnością serwera.
Architekt BI, który projektuje i wdraża rozwiązania analityczne	**1.** Instalacja plików ćwiczeń zgodnie z opisem w punkcie „Instalacja i użytkowanie plików ćwiczeń”. **2.** Wykonanie części I, aby zapoznać się z korzyściami płynącymi z Analysis Services. **3.** Praca z częściami II i III, aby nauczyć się jak tworzyć wymiary i kostki oraz jak implementować zaawansowane techniki projektowania. **4.** Wykonanie części IV, aby zrozumieć sposób projektowania kostek, które implementują funkcje zabezpieczeń, wydajności i przetwarzania Analysis Services.

Konwencje i funkcje stosowane w książce

Książka prezentuje informacje przy użyciu konwencji, dzięki której informacje są czytelne i łatwe do śledzenia. Przed rozpoczęciem lektury warto przeczytać poniższą listę, która tłumaczy konwencje i wskazuje pomocne funkcje.

Konwencje

- Każde ćwiczenie składa się z serii zadań. Każde zadanie jest prezentowane jako ciąg ponumerowanych kroków (1, 2 i tak dalej).

- Uwagi zatytułowane „wskazówka” podają dodatkowe informacje lub alternatywne metody służące pomyślnemu wykonaniu danego kroku.
- Uwagi zatytułowane „ważne” zwracają uwagę na informacje, które trzeba sprawdzić przed kontynuowaniem ćwiczenia.
- Tekst wpisywany przez użytkownika jest **pogrubiony**.
- Znak plus (+) między nazwami dwóch klawiszy oznacza, że trzeba nacisnąć te klawisze jednocześnie. Na przykład Naciśnij „Alt+Tab” oznacza, że trzeba przytrzymać klawisz Alt i nacisnąć klawisz tabulatora.

Wymagania dotyczące sprzętu i oprogramowania

Aby wykonać ćwiczenia praktyczne zawarte w tej książce, potrzebny jest następujący sprzęt i oprogramowanie:

- Wersja 32- lub 64-bitowa SQL Server 2008 wydanie Enterprise lub SQL Server 2008 wydanie Developer. Trzeba zainstalować następujące komponenty SQL Serwer 2008: Database Engine, Analysis Services i Reporting Services. Podczas instalacji dodajemy nasz ID logowania z systemu Windows do listy administratorów systemów SQL Server i Analysis Services.

> **Wskazówka** W pełni funkcjonalną wersję próbną SQL Server 2008 Enterprise Edition, działającą przez 180 dni, można pobrać z witryny: *http://www.microsoft.com/sqlserver/2008/en/us/trial-software.aspx*.



- SQL Server 2008 Enterprise Edition wymaga systemu Microsoft Windows Server 2008 lub Windows Server 2003 SP2. SQL Server 2008 Developer Edition wymaga systemu Microsoft Windows Server 2008, Windows Server 2003 SP2, Windows Vista lub Windows XP SP2.
- Procesor 1.0 GHz Pentium III (zalecany jest 2.0 GHz lub szybszy dla 32-bitowej wersji SQL Server 2008.
- Procesor 1.4 GHz (zalecany 2.0 GHz lub szybszy) AMD Opteron, AMD Athlon 64, Intel Xeon z obsługą Intel EM64T, Intel Pentium IV z obsługą EM64T lub 1.0 Ghz albo szybszy procesor Itanium dla 64-bitowego SQL Server 2008.
- Dostępna pamięć fizyczna RAM 512 MB (zalecane 2 GB lub więcej).
- Karta VGA lub wyższej rozdzielczości, o minimalnej rozdzielczości 1 024 x 768 pikseli.
- Stacja CD-ROM lub DVD-ROM.
- Mysz Microsoft lub inne kompatybilne urządzenie.

Szczegółowe wymagania dotyczące instalacji SQL Server 2008 można znaleźć w artykule „Hardware and Software Requirements for Installing SQL Server 2008", na SQL Server Books Online (*http://msdn.microsoft.com/en-us/library/ms143506.aspx*).

Ćwiczenia „krok po kroku", zawarte w tej książce i towarzyszące im pliki do ćwiczeń były testowane za pomocą Windows Vista Enterprise Service Pack 1 i Microsoft SQL Server 2008 Developer Edition. W przypadku użycia innej wersji systemu operacyjnego lub SQL Server 2008 mogą wystąpić pewne drobne różnice.

Pliki ćwiczeń i konfiguracja SQL Server 2008

Obraz dysku CD towarzyszącego książce jest dostępny na stronie wydawcy przy opisie książki w zakładce Dodatkowe informacje, pod adresem:

http://www.ksiazki.promise.pl/aspx/produkt.aspx?pid=51059

Na podstawie tego obrazu można wykonać fizyczny dysk CD lub zainstalować go jako napęd wirtualny. Dysk ten zawiera zawiera pliki do ćwiczeń, których będziemy używać podczas wykonywania zadań praktycznych. Pliki te i szczegółowe instrukcje w kolejnych lekcjach umożliwiają naukę poprzez praktykę, co stanowi łatwy i efektywny sposób nabywania i zapamiętywania nowych umiejętności. Płyta CD zawiera także bazę danych SQL Server 2008, która jest źródłem danych dla rozwiązań analitycznych tworzonych w tej książce.

Instalacja plików do ćwiczeń

Instalacja plików ćwiczeń na swoim komputerze wymaga wykonania podanych niżej kroków.

1. Wkładamy dysk do stacji CD-ROM/montujemy obraz dysku jako napęd wirtualny.



Uwaga Umowa licencyjna użytkownika końcowego otworzy się automatycznie. Jeśli umowa nie pojawi się, trzeba na pulpicie otworzyć Computer (Komputer) lub menu Start, dwukrotnie kliknąć ikonę stacji CD-ROM, a następnie dwukrotnie kliknąć StartCD.exe.

2. Pobieramy umowę licencyjną użytkownika końcowego. Jeśli akceptujemy warunki, wybieramy akceptację i klikamy Next.

   Pojawi się menu z opcjami związanymi z tą książką.

3. Klikamy pliki Install Practice (Instaluj ćwiczenia).

4. Wykonujemy pojawiające się polecenia.

   Pliki ćwiczeń zostaną skopiowane z płyty CD na dysk twardy. Domyślnym folderem instalacji jest C:\Microsoft Press\Analysis Services 2008 SBS. Jeśli instalujemy

pliki ćwiczeń w innej lokalizacji, trzeba będzie ją podać podczas pracy nad ćwiczeniami. Każdy rozdział książki tłumaczy, kiedy i jak używać plików ćwiczeń dla danego rozdziału. Gdy trzeba będzie użyć danego pliku ćwiczeń, w książce znajdą się instrukcje, jak go otworzyć.

Wskazówka W folderze C:\Microsoft Press\Analysis Services 2008 SBS\Answers znajduje się oddzielny folder dla każdego rozdziału, w którym wprowadzamy zmiany do plików ćwiczeń. Po ukończeniu rozdziału pliki z tych folderów są kopiami plików przykładowych. Można się do nich odwołać, aby sprawdzić wyniki wszystkich ukończonych ćwiczeń z rozdziału.



5. Usuwamy płytę CD ze stacji CD-ROM.

Konfiguracja SQL Server 2008

Ćwiczenia z tej książki pozwolą utworzyć bazę danych Analysis Services i zapełnić ją za pomocą danych zapisanych w bazie danych SQL Server. W ćwiczeniach przyjmuje się, że na naszym komputerze jest zainstalowany SQL Server 2008. Poniższe kroki poprowadzą nas przez proces konfiguracji SQL Server i Analysis Services, aby można było wykonywać ćwiczenia. Wykonanie tych kroków wymaga uprzedniej instalacji plików ćwiczeń; trzeba być także członkiem grupy Administrators na swoim komputerze.

Usługi Analysis Services będą wymagały połączenia z SQL Server, trzeba więc wcześniej poznać nazwę loginu użytkownika systemu Windows dla Analysis Services, a następnie przyznać uprawnienia dla SQL Server na tę nazwę użytkownika.

1. W pasku narzędzi Microsoft Windows klikamy Start. W polu Search (Szukaj) wpisujemy **services**, a następnie w grupie Programs wybieramy Services (Usługi). Jeśli pojawia się okno dialogowe User Account Control (Kontrola konta użytkownika), klikamy Continue. [Jeśli korzystamy z Windows XP lub Windows Server 2003, klikamy Start, wybieramy Control Panel (Panel sterowania), potem Administrative Tools (Narzędzia administracyjne), a następnie Services.]
2. W oknie Services znajdujemy kolumnę Name (Nazwa) SQL Server Analysis Services i zapisujemy wartość w kolumnie Log On As (Zaloguj się jako).

Ważne Nazwa loginu użytkownika Analysis Services będzie złożona z dwóch części w formacie *NazwaDomeny\NazwaUżytkownika* lub .*NazwaUżytkownika,* lub też może to być *Usługa Sieciowa* lub *Usługa Lokalna*. Nazwy tej użyjemy w kroku 5.



3. Zamykamy okno Services.
4. W pasku narzędzi Microsoft Windows klikamy przycisk Start, a następnie klikamy Computer. W Windows Explorer przechodzimy do C:\Microsoft Press\Analysis

Services 2008 SBS\Setup. Prawym przyciskiem klikamy SSAS_Login.sql i wybieramy Open. W oknie dialogowym Connect To Database Engine (Połącz z mechanizmem bazy danych) klikamy Cancel.

Otwiera się Microsoft SQL Server Management Studio z plikiem SSAS_Login.sql wyświetlonym w trybie edycji. Ten plik skryptu SQL tworzy login SQL Server dla Analysis Services (o ile nie istnieje), a następnie nadaje temu loginowi uprawnienia. Trzeba dokonać edycji skryptu i podać nazwę loginu użytkownika Analysis Services, którą poznaliśmy w kroku 2.

5. W drugim wierszu pliku znajdujemy DOMAIN\ACCOUNT i zastępujemy go dwuczęściową nazwą ustaloną w kroku 2.



Ważne Jeśli w kroku 2. ustalimy, że nazwa loginu użytkownika Analysis Services to *Usługa sieciowa* lub *Usługa lokalna*, to będzie trzeba zastąpić DOMAIN\ACCOUNT przez NT Authority\Network Service lub NT Authority\Local Service.

Jeśli nazwa użytkownika poznana w kroku 2. ma format *NazwaDomeny\NazwaUżytkownika*, trzeba użyć nazwy dwuczęściowej. Jeśli poznana nazwa użytkownika ma format .*NazwaUżytkownika*, trzeba zastąpić kropkę (.) nazwą komputera lokalnego. Na przykład na komputerze autora login Analysis Services ma postać .\AS_Service. Nazwa komputera to Hitachi, więc trzeba zastąpić DOMAIN\ACCOUNT przez Hitachi\AS_Service. W tym przypadku drugi wiersz skryptu powinien wyglądać tak:

```
SET @user ='Hitachi\AS_Service';
```

6. W menu File wybieramy Exit. W oknie dialogowym klikamy Yes, aby zapisać zmiany w SSAS_Login.sql.
7. W Windows Explorer klikamy prawym przyciskiem Setup.cmd i wybieramy Open.

   Ten plik poleceń przyłącza bazę danych Microsoft SQL Server 2008, która stanowi źródło danych dla tworzonego i używanego w książce rozwiązania analitycznego. Potem wykonuje skrypt zmodyfikowany w kroku 5.

Możemy teraz zacząć wykonywać kolejne ćwiczenia.

Odinstalowanie plików ćwiczeń

Aby usunąć z komputera pliki ćwiczeń, należy wykonać podane niżej kroki.

1. W Control Panel otwieramy Add Or Remove Programs (Dadaj lub usuń programy).
2. Z listy Currently Installed Programs (Zainstalowane program) wybieramy Microsoft SQL Server 2008 Analysis Services Step by Step.
3. Klikamy Remove (Usuń).

4. W celu usunięcia plików ćwiczeń postępujemy zgodnie z pojawiającymi się instrukcjami.

Znajdowanie dodatkowej informacji online

W miarę pojawiania się nowego lub zaktualizowanego materiału uzupełniającego tę książkę, jest on umieszczany w Internecie, w witrynie Microsoft Press Online Developer Tools. Materiały, które można tam znaleźć, obejmują aktualizacje zawartości książki, artykuły, łącza do wspomagającej zawartości, erraty, przykładowe rozdziały i wiele więcej. Witryna jest dostępna pod adresem: *www.microsoft.com/learning/books/online/developer* i jest okresowo aktualizowana.

Wsparcie dla książki

Dołożono wszelkich starań, aby książka i zawartość płyty CD były dokładne i najwyższej jakości. W miarę pojawiania się poprawek lub zmian, są one dodawane do artykułu w bazie wiedzy Microsoft Knowledge Base.

Microsoft Press zapewnia obsługę tej książki i płyty CD na poniższej witrynie:

http://www.microsoft.com/learning/support/books/

Pytania i komentarze

Wszelkie komentarze, pytania lub pomysły związane z tą książką lub dołączoną do niej płytą CD, lub też pytania, na które nie można znaleźć odpowiedzi na podanych wyżej witrynach, może wysłać pocztą elektroniczną do Microsoft Press na adres:

mspinput@microsoft.com

lub też pocztą tradycyjną:

Microsoft Press
Attn: *Microsoft SQL Server 2008 Analysis Services Step by Step* Editor
One Microsoft Way
Redmond, WA 98052-6399

Zwracamy uwagę, że pod tymi adresami nie jest oferowane wsparcie techniczne dla oprogramowania firmy Microsoft.

CZĘŚĆ I

Istota analizy biznesowej a Analysis Services

ROZDZIAŁ 1: Analiza biznesowa: podstawy analizy danych 3

ROZDZIAŁ 2: Omówienie OLAP i Analysis Services 27

ROZDZIAŁ 3: Dostęp do danych źródłowych 43

ROZDZIAŁ 1

Analiza biznesowa: podstawy analizy danych

W tym rozdziale nauczymy się, jak:

- Zrozumieć potrzeby i struktury systemu BI.
- Zrozumieć koncepcję wielowymiarowej analizy danych.
- Hurtownia danych implementuje przestrzenny model danych.
- Usługi Analysis Services implementują przestrzenny model danych.

Wprowadzenie do analizy biznesowej

Analiza biznesowa (Business Intelligence, BI) dotyczy informacji wyprowadzonych z danych zawartych w systemach operacyjnych naszej firmy i źródeł zewnętrznych. BI powinna pomagać w szybszym podejmowaniu dobrych decyzji. Przypuśćmy, że zarządzamy niewielką nową firmą, Adventure Works Cycles, która produkuje i sprzedaje rowery, części do nich, stroje i akcesoria na rynkach Ameryki Północnej, Europy i Azji. Firma musi się rozrastać, ale ma zbyt małe zasoby, aby wspomóc rozwój. Trzeba podjąć odpowiednie decyzje, a to wymaga określonych informacji. Bieżące trendy biznesowe można znaleźć na łamach *Wall Street Journal*, zaś przeglądarka jest na stale ustawiona na witrynę *www.bloomberg.com*. Te informacje wraz z doświadczeniem pozwalają na podjęcie subiektywnej decyzji, „na wyczucie". Ale potrzebne są decyzje obiektywne, oparte na liczbach. Potrzebne liczby znajdują się w systemach przetwarzania zamówień, księgowości, kadrach i innych zasobach firmy. Trzeba też uzyskać prognozy rynkowe dotyczące konkurencji i dane na temat przewidywanych kursów walut. Potrzebne jest narzędzie, które połączy te wszystkie informacje i poda gotowe wyniki. Takim narzędziem jest system BI. Jest to rozwiązanie do zbierania danych z wielu źródeł, przekształcania ich tak, aby były spójne, umieszczenia ich w jednej lokalizacji oraz prezentowania umożliwiającego analizę i podejmowanie decyzji.

System BI może składać się z pięciu warstw:

1. Warstwy źródła danych
2. Warstwy transformacji danych
3. Warstwy przechowywania i odzyskiwania danych
4. Warstwy analitycznej
5. Warstwy prezentacji

Warstwa źródła danych składa się z systemów danych, które są używane przez organizację do prowadzenia codziennej działalności, danych w plikach tekstowych, arkuszy Microsoft Office Excel lub baz danych Microsoft Office Access oraz danych pozyskiwanych ze źródeł zewnętrznych. Ponieważ dane pochodzą z wielu różnych źródeł, tworzenie raportów i wykonywanie analiz jest bardzo trudne. Warstwa transformacji danych jest używana do pobierania danych z wielu różnych źródeł, ich modyfikacji, aby były wewnętrznie spójne, oraz ładowania danych do systemu przechowywania danych.

Warstwa przechowywania i odzyskiwania danych jest hurtownią danych utworzonych w systemie zarządzania relacyjną bazą danych. Hurtownia danych jest systemem zapisów i zawiera wartości istotne dla organizacji. Hurtownia danych dojrzałej firmy zawiera dane związane ze wszystkimi aspektami przedsiębiorstwa. Jest to miejsce skomplikowane i pełne ruchu. Dane są uzupełniane co miesiąc, co tydzień, codziennie, a nawet częściej. Raporty i analityczne zapytania są wykonywane w dzień i w nocy. Aby zmniejszyć obciążenie hurtowni danych i uprościć dostęp użytkowników, dane dotyczące pojedynczych obszarów tematycznych są wydobywane z hurtowni, sumowane i ładowane do minihurtowni danych (*data marts*). Są to relacyjne bazy danych lub też wielowymiarowe bazy danych OLAP (*online analytical processing*). Usługa Analysis Services stanowi jeden z przykładów wielowymiarowej bazy danych OLAP.

Celem warstwy analitycznej naszego systemu BI jest przekształcenie danych na informacje i zapewnienie szybkiego i łatwego dostępu do tych informacji dla decydentów. Wielowymiarowe bazy danych OLAP tworzą warstwę analityczną systemu BI. Gdy szczegółowe dane z hurtowni danych zostają załadowane do wielowymiarowej bazy danych OLAP, wstępnie obliczane są zsumowane wartości. Ponieważ są one przechowywane w bazie danych, raporty i zapytania analityczne są wykonywane szybko. Gdy dane są ładowane do wielowymiarowej bazy danych OLAP, dodawane są do nich metadane. Metadane zawarte w bazie danych OLAP obejmują informacje o relacjach i hierarchiach danych, sposobie sortowania i podsumowywania danych oraz o sposobie formatowania ich w celu prezentacji. Metadane zawarte w bazie danych OLAP są tym, co zmienia dane w informacje. W bazie danych OLAP mogą być tworzone i zapisywane złożone obliczenia. Dzięki temu dostęp do informacji jest łatwiejszy, gdyż twórcy raportów i zapytań nie muszą wciąż tworzyć obliczeń, zaś każda osoba w organizacji może korzystać z dokładnie tych samych wzorów obliczeniowych.

Narzędzia do tworzenia raportów i wizualizacji tworzą warstwę prezentacji systemu BI. Aplikacje w warstwie prezentacji systemu BI mogą przeszukiwać hurtownie i minihurtownie danych lub wielowymiarowe bazy danych OLAP, prezentując ich zawartość w różnorodnych formatach. Nasz system BI może wysyłać raporty na telefony komórkowe naszych zewnętrznych przedstawicieli handlowych i pracowników działu dostaw, aby kierować ich codziennymi działaniami. Analityczne diagramy przepływu pracy ilustrują złożone procesy biznesowe, pokazując w każdym węźle informacje, które pozwalają decydentom na podejmowanie wartościowych decyzji. Interakcyjne raporty analityczne z możliwościami typu „slice-and-dice", tabelami przestawnymi

i możliwością eksploracji szczegółów oraz wielowymiarowe wizualizacje danych pomagają menedżerom i analitykom w wyszukiwaniu błędów oraz znajdowaniu informacji, umożliwiając bardziej efektywne działanie organizacji i korzystanie z nowych możliwości. Raporty tablicowe z kluczowymi wskaźnikami pozwalają zarządzającym na szybkie określenie, czy organizacja spełnia swoje cele strategiczne.

Zestawu elementów składających się na system BI nie można traktować dogmatycznie. System BI może mieć wszystkie wymienione warstwy lub tylko kilka z nich. Jeśli na przykład używamy Analysis Services w systemie monitoringu zbliżonym do czasu rzeczywistego, można co kilka minut ładować dane bezpośrednio z bazy danych kontroli systemu eksploatacyjnego do wielowymiarowej bazy danych OLAP. Warstwa prezentacji w systemie czyta dane z Analysis Services i wyświetla diagram pokazujący każdą maszynę linii produkcyjnej i to, czy w danej chwili działa ona dobrze w porównaniu ze średnią liczoną w dłuższym czasie. Taki system BI nie ma hurtowni danych, zaś przekształcanie danych i warstwy analityczne są ze sobą połączone. Złożoność systemu BI nie jest istotna. Naprawdę ważne jest, aby dostarczał on informacji właściwych do podejmowania decyzji, zrozumiałych oraz czytelnych, niezawodnych, a także dawał szybki i łatwy dostęp do nich i możliwość posługiwania się nimi, oraz udostępniał je, gdy będą potrzebne.

Wielowymiarowa analiza danych

Systemu BI używamy, aby wykonywać wielowymiarową analizę danych. Niech ta terminologia nas nie przeraża. Większość z nas wykonuje już od pewnego czasu wielowymiarową analizę danych. Każdy, kto kiedyś czytał raport pokazujący dane liczbowe według różnych atrybutów – na przykład przeglądy według klientów w poszczególnych miesiącach – wykonywał właśnie wielowymiarową analizę danych. W takim przypadku wymiarami używanymi w naszym przeglądzie są klienci oraz miesiące.

W tej części poznamy koncepcje związane z wielowymiarową analizą danych, takie jak atrybuty, hierarchie i wymiary. Przygotuje nas to do rozdziału 2., „Usługi analityczne OLAP", w którym poznamy wielowymiarowe bazy danych OLAP i szczególne zalety Analysis Services.

Atrybuty w analizie danych

Aby lepiej zrozumieć atrybuty i hierarchie, rozważmy scenariusz, w którym prezes firmy Adventure Works Cycles zechce wiedzieć więcej o tym, jak pracuje jego firma. Poprosi on analityka biznesowego o raport. Otrzymaną odpowiedź pokazuje zawartość tabeli 1-1.

Tabela 1-1 Praca firmy Adventure Works

42

Wynik taki wprawia w osłupienie. Wiadomo, że odpowiada na jakieś pytanie, ale na *jakie*? Raport nie ma żadnej wartości. *Liczby bez właściwego kontekstu są danymi, ale nie niosą informacji*. W BI podsumowana wartość liczbowa używana do monitorowania działania organizacji nosi nazwę *miary*. Gdy patrzymy na informacje liczbowe, jedną z pierwszych rzeczy jest ustalenie, jakie miary chcemy oglądać. Miara może reprezentować sprzedaż w dolarach (Sales Dollars), wysyłane jednostki (Shipment Units), liczbę braków na godzinę (Defects Per Hour) lub reakcję na kampanię reklamową (Ad Campaign Responses). Jeśli pytamy o liczbę jednostek sprzedaży firmy Adventure Works, otrzymujemy raport pokazany w tabeli 1-2.

Tabela 1-2 Sprzedaż firmy Adventure Works

Units Sold
70

Dodanie etykiety do liczb przekształciło dane w informacje. Można zobaczyć, że liczba 70 reprezentuje Units Sold, liczbę sprzedanych sztuk. Etykieta to *metadane* – dane opisujące dane. Jednym ze sposobów, w jaki aplikacja BI przekształca dane na informacje, jest dodawanie metadanych. Oglądanie pojedynczej liczby nie mówi nam wiele. Chcemy otrzymać coś bardziej informacyjnego. Na przykład, jak firma daje sobie radę w poszczególnych okresach. Możemy porosić o analizę w rozbiciu na miesiące. Nowy raport pokazuje tabela 1-3.

Tabela 1-3 Units Sold dla Adventure Works według Month (Miesiąc)

Jan 2011	Feb 2011	Mar 2011	Apr 2011
4	14	27	25

Mamy teraz więcej informacji, gdyż możemy zobaczyć więcej metadanych. Jednym z atrybutów każdej wartości Units Sold jest miesiąc, w którym nastąpiła sprzedaż. Nasza firma działa od czterech miesięcy, więc na górze raportu znajdujemy cztery etykiety atrybutu, jakim są miesiące. Zauważmy także, że etykiety miesięcy są uporządkowane chronologicznie, a nie alfabetycznie lub losowo. Kolejność pojawiania się etykiet atrybutu to także metadane, które są istotne dla zrozumienia jak firma dawała sobie radę w kolejnych miesiącach.

Raport miesięczny wciąż nas nie satysfakcjonuje. Firma sprzedaje więcej niż jeden produkt. Jaka jest sprzedaż każdego z produktów w tym okresie? Prosimy więc o nowy raport, który zawiera Units Sold według produktów i miesięcy, co widać w tabeli 1-4.

Tabela 1-4 Units Sold dla Adventure Works według Product i Month

	Jan 2011	Feb 2011	Mar 2011	Apr 2011
Mountain-500 Black, 40	1	3	1	2
Mountain-500 Black, 44		2		1
Mountain-500 Black, 48		1	2	1
Mountain-500 Silver, 40		1	2	1
Mountain-500 Silver, 44		1	1	1
Mountain-500 Silver, 48	2			
Road-750 Black, 44			10	7
Road-750 Black, 48			5	9
Hitch Rack	1	6	6	3

Mamy teraz znacznie więcej informacji, gdyż możemy zobaczyć jeszcze więcej metadanych. Każda wartość pozycji Units Sold ma dwa atrybuty: określony miesiąc i rodzaj sprzedanego produktu. Firma produkuje trzy modele produktu w różnych rozmiarach i kolorach, co daje dziewięć unikatowych produktów. Po lewej stronie raportu pojawia się więc dziewięć etykiet atrybutów: po jednym dla każdego produktu.

Ponieważ to mała firma, można łatwo zrozumieć zasady jej działalności, patrząc na raport, taki jak w tabeli 1-4, gdzie widzimy szczegóły liczbowe. Jednak w miarę rozwoju firmy i zwiększania się liczby wytwarzanych produktów, podobny raport może liczyć wiele stron. Wiele szczegółowych wierszy tego raportu spowoduje, że trudno będzie odczytać trendy w działaniu naszej firmy. Pogląd, iż zbyt wiele szczegółów może utrudnić zrozumienie sposobu działania firmy, jest poparty badaniami psychologicznymi, które wskazują, że większość ludzi może przyswoić około siedmiu elementów lub siedmiu grup elementów. Badania te pokazują, że grupowanie – czyli *agregacja* – stanowi sposób postępowania z nadmiarem informacji. Ponieważ Adventure Works rośnie, będziemy chcieli zgrupować rosnącą liczbę pojedynczych produktów do postaci modelu, podkategorii i kategorii. W optymalnej sytuacji każda grupa będzie zawierać około siedmiu elementów.

Możemy zacząć tworzenie grup produktów, patrząc na ich wspólne atrybuty. Tabela 1-5 pokazuje, jak można podzielić nazwę produktu na atrybuty nazwy modelu, koloru i rozmiaru.

Tabela 1-5 Atrybuty produktu

Product (Produkt)	Model (Model)	Color (Kolor)	Size (Rozmiar)
Mountain-500 Black, 40	Mountain-500	Black	40
Mountain-500 Black, 44	Mountain-500	Black	44
Mountain-500 Black, 48	Mountain-500	Black	48
Mountain-500 Silver, 40	Mountain-500	Silver	40
Mountain-500 Silver, 44	Mountain-500	Silver	44
Mountain-500 Silver, 48	Mountain-500	Silver	48
Road-750 Black, 44	Road-750	Black	44
Road-750 Black, 48	Road-750	Black	48
Hitch Rack	Hitch Rack		

Mamy więc w raporcie trzy dodatkowe listy etykiet atrybutów produktu, których można użyć do tworzenia grup w raporcie. Ponieważ określenie *lista etykiet atrybutów* jest dość długie, w praktyce BI mówi się o nich po prostu *atrybuty*. Etykiety z każdej listy związane są ze sobą i należą do tego samego atrybutu, więc etykiety nazywamy *elementami*. Tak więc w tym przykładzie atrybut Model ma trzy elementy: Hitch Rack, Mountain-500 i Road-750. Atrybut Color ma dwa elementy: Black i Silver.

Atrybut Product jest *atrybutem kluczowym*. Atrybut kluczowy jednoznacznie identyfikuje elementy wszystkich innych atrybutów. Jeśli oglądamy element atrybutu Product, to znamy elementy wszystkich innych atrybutów tego produktu. Wiemy na przykład, że produkt Mountain-500 Silver ma rozmiar 42 i jest rowerem koloru srebrnego, model Mountain-500. Innym sposobem identyfikacji atrybutu kluczowego jest utworzenie listy powiązanych atrybutów, co widać w tabeli 1-5, w której atrybuty kluczowe są unikatowe. Model nie jest atrybutem kluczowym. Jeśli patrzymy na element Mountain-500 atrybutu Model, nie wiemy jaki jest jego atrybut Color ani Size. Możemy także zobaczyć, że elementy atrybutu Model powtarzają się w tabeli 1-5.

Można teraz wymagać, aby analityk utworzył raport grupujący produkty według modelu, co widać w tabeli 1-6.

Tabela 1.6- Units Sold dla firmy Adventure Works według atrybutów Model, Product i Month

	Jan 2011	Feb 2011	Mar 2011	Apr 2011
Mountain-500	3	8	6	6
Mountain-500 Black, 40	1	3	1	2
Mountain-500 Black, 44		2		1
Mountain-500 Black, 48		1	2	1
Mountain-500 Silver, 40		1	2	1
Mountain-500 Silver, 44		1	1	1
Mountain-500 Silver, 48	2			

Tabela 1.6- Units Sold dla firmy Adventure Works według atrybutów Model, Product i Month

	Jan 2011	Feb 2011	Mar 2011	Apr 2011
Road-750			15	16
Road-750 Black, 44			10	7
Road-750 Black, 48			5	9
Hitch Rack	1	6	6	3
Hitch Rack	1	6	6	3

Nasz raport zawiera teraz informacje podsumowujące. Wartość Units Sold dla każdego Modelu stanowi sumę, czyli inaczej *wartość zagregowaną* wartości Units Sold dla określonego Productu. Elementy atrybutów Model i Product są ustawione *hierarchicznie*, gdzie elementy atrybutu Model tworzą górny poziom hierarchii, zaś elementy atrybutu Product – poziom dolny. Możliwość ustawienia produktów i modeli w sposób hierarchiczny i zagregowania wartości produktu do wartości wspólnych dla modelu tworzy dodatkowe metadane, które zwiększają możliwość podawania informacji przez nasze dane.

Raport w tabeli 1-6 jest zbliżony do czegoś, czego można użyć w sytuacji, gdy firma Adventure Works produkuje wiele produktów. Gdy mamy do czynienia z hierarchią, nie musimy wyświetlać wszystkich jej poziomów. W naszym przypadku chcemy zobaczyć tylko poziom Model, więc prosimy analityka o wyłączenie z raportu poziomu Product. Tabela 1-7 pokazuje raport podsumowujący, który daje nam bardziej zrozumiały ogląd jakości działania firmy.

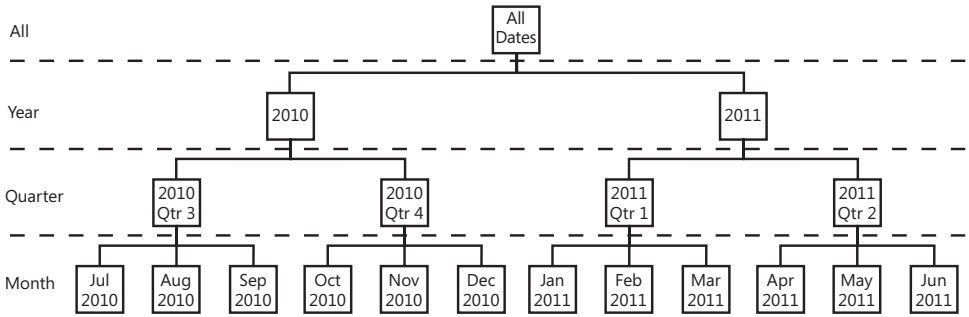
Tabela 1-7 Units Sold dla Adventure Works według Model i Month

	Jan 2011	Feb 2011	Mar 2011	Apr 2011
Mountain-500	3	8	6	6
Road-750			15	16
Hitch Rack	1	6	6	3

Zobaczyliśmy do tej pory, jak atrybut na poziomie szczegółowym można rozdzielić na zbiór powiązanych atrybutów. Podczas wykonywania wielowymiarowej analizy danych możemy pracować z wieloma innymi atrybutami. Na przykład będziemy często tworzyć raporty zawierające datę, pracownika, położenie geograficzne, klienta i inne atrybuty. Wiemy też, jak atrybuty produktu i modeli można zorganizować w prostą hierarchię, a następnie wykorzystać ją do utworzenia raportu grupującego i podsumowującego dane szczegółowe. W kolejnym punkcie znajduje się więcej informacji na temat tworzenia hierarchii.

Hierarchie w analizie danych

Hierarchię tworzymy, organizując dane według poziomów. Hierarchia Product by Model (Produkty według modeli) pokazuje tylko dwa poziomy, ale hierarchia może mieć wiele poziomów. Można, na przykład, utworzyć hierarchię Date (Data) z wieloma poziomami, wykorzystując atrybut Month jako najniższy poziom, a następnie dodając poziomy pośrednie utworzone z elementów członkowskich atrybutów Quarter (Kwartał) i Year (Rok). Można następnie dodać jeszcze jeden poziom na szczycie hierarchii Date jako *poziom All* (Wszystkie). Poziom ten zawiera jeden element członkowski – All Dates (Wszystkie daty) – stanowiący agregację wszystkich elementów poziomu Year. Rysunek 1-1 pokazuje niektóre elementy członkowskie hierarchii Date ustawione na poziomach All, Year, Quarter i Month.



Rysunek 1-1 Hierarchia Date

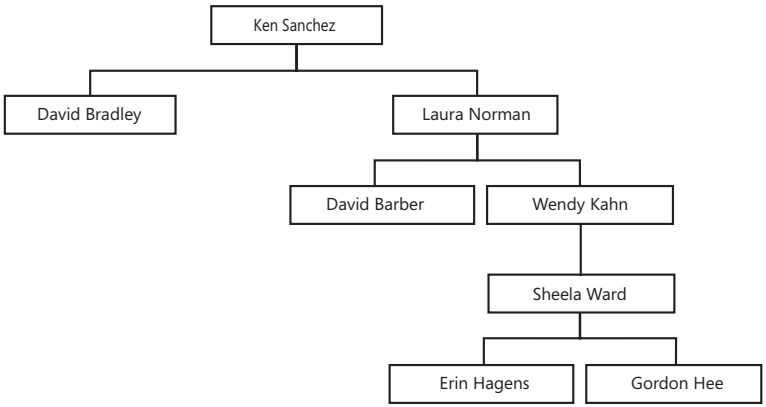
Wszystkie elementy hierarchii można traktować jako połączone relacjami nadrzędny-podrzędny. Element nadrzędny jest powiązany z elementami kolejnego, niższego poziomu. Element tamtego poziomu jest elementem podrzędnym dla powiązanych z nim elementów poziomu o jeden wyższego. Rysunek 1-1 pokazuje, że 2011 jest nadrzędny dla Qtr I 2011 i Qtr II 2011. Można też zobaczyć, że Feb 2011 jest podrzędny dla Qtr I 2011. Hierarchia Date stanowi hierarchię naturalną. W hierarchii tego rodzaju każdy element podrzędny ma tylko jeden element nadrzędny. Hierarchie Size by Color (Wielkość według kolorów) nie jest hierarchią naturalną. Element 40 poziomu Size ma dwa elementy nadrzędne: elementy poziomu Color – Black i Silver.

Rysunek 1-1 pokazuje także, że hierarchia może mieć wiele gałęzi. Gdy dochodzimy do końca gałęzi drzewa, możemy tam znaleźć liście. Na końcu gałęzi hierarchii są *elementy liście*. Hierarchia może mieć element All lub nie mieć go, ale każda hierarchia musi mieć elementy liści.

Każdy atrybut można przekształcić w *hierarchię atrybutów*, która ma tylko dwa poziomy. Używając atrybutu jako poziomu liścia, a następnie dodając poziom All, można utworzyć hierarchię Color, która ma element poziomu All jako All Products (Wszystkie produkty) oraz elementy liści Black i Silver.

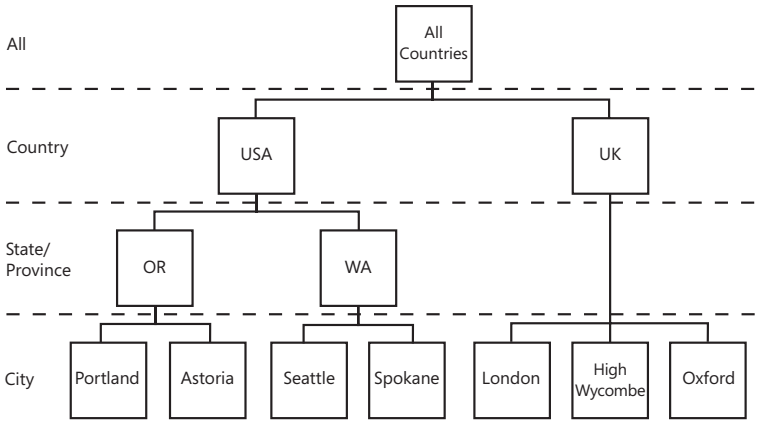
Niektóre hierarchie, jak hierarchia Date, są *zrównoważone*. Jak pokazuje rysunek 1-1, nie ma znaczenia, jaką ścieżką poruszamy się wzdłuż gałęzi, gdyż na każdym poziomie znajdziemy element, zaś każda ścieżka ma taką samą liczbę poziomów. W hierarchii zrównoważonej można łatwo nadawać nazwy poziomom.

Niektóre hierarchie są *niezrównoważone*. Schemat organizacyjny firmy jest często niezrównoważony. Na przykład wiele firm ma dużo więcej pracowników, a więc więcej poziomów zarządzania, na poziomie produkcji niż w dziale kadr. Struktury podziału pracy i plany kont stanowią kolejne przykłady niezrównoważonych hierarchii. W niezbilansowanej hierarchii często trudno jest nazwać poszczególne poziomy, natomiast elementami liści są zawsze te elementy, które nie mają pod sobą żadnych elementów podrzędnych. Rysunek 2-1 pokazuje, że w hierarchii niezrównoważonej niezależnie od ścieżki, jaką poruszamy się wzdłuż gałęzi, na każdym poziomie znajdziemy jakieś elementy. Jednak niektóre ścieżki mogą mieć więcej poziomów od innych. Inaczej mówiąc, elementy liści mogą należeć do różnych poziomów.



Rysunek 1-2 Schemat organizacyjny

W niektórych hierarchiach następuje pewne rozmycie między zrównoważeniem i niezrównoważeniem. Na przykład w hierarchii geograficznej można łatwo nazwać poziomy – All, Country (Kraj), State/Provinece (Stan/Prowincja) i City (Miasto) – ale niektóre kraje mogą nie mieć podziału na stany czy też prowincje. Są to hierarchie *wadliwe* (ang. *ragged*). Rysunek 1-3 pokazuje, że niezależnie od wybranej ścieżki, każda ścieżka ma tę samą liczbę poziomów. Jednak niektóre ścieżki na niektórych poziomach nie mają elementów. W hierarchii wadliwej elementy liści zawsze należą do tego samego poziomu.



Rysunek 1-3 Hierarchia wadliwa

Hierarchie są tworzone przez podział atrybutów na poziomy. Hierarchia może mieć tylko dwa poziomy lub dowolną większą liczbę poziomów. Najniższy poziom hierarchii nazywany jest *poziomem liścia*, potem są poziomy pośrednie, a na końcu poziom najwyższy. W większości hierarchii najwyższym poziomem jest All. Zawiera on jeden element o nazwie All. O elementach hierarchii można myśleć w kategoriach relacji nadrzędny-podrzędny, czyli jest poziom nadrzędny i podrzędny. Jeśli element podrzędny ma tylko jeden element nadrzędny, wtedy hierarchię określamy jako naturalną. Jeśli hierarchia jest zrównoważona, to na każdej jej ścieżce znajduje się tyle samo poziomów, a każdy poziom ma element. W hierarchii niezrównoważonej różne ścieżki hierarchii mogą mieć różną liczbę poziomów, zaś w hierarchii wadliwej każda ścieżka ma tę samą liczbę poziomów, ale nie każdy poziom zawiera jakieś elementy.

Wymiary w analizie danych

Aby rozpocząć naukę o wymiarach, przejrzyjmy kilka raportów, pokazanych wcześniej. Tabela 1-8 pokazuje raport Units Sold według Month w firmie Adventure Works, który wcześniej był przedstawiony w tabeli 1-3. Liczba miesięcy – cztery – określa liczbę wartości zawartych w raporcie. Jest to analogiczne do obliczenia odległości w świecie fizycznym: długość linii jest po prostu jej długością.

Tabela 1-8 Units Sold dla firmy Adventure Works według Month

Jan 2011	Feb 2011	Mar 2011	Apr 2011
4	14	27	25

Tabela 1-9 pokazuje raport zawierający sprzedaż firmy Adventure Works dla Units Sold według Model i Month, który wcześniej pojawił się w tabeli 1-7. W raporcie tym miesięczne wartości Units Sold są podzielone według elementów członkowskich atrybutu Model. Potencjalna liczba wartości w raporcie jest równa liczbie modeli

pomnożonej przez liczbę miesięcy, mamy więc do 12 możliwych wartości do rozważenia. Przypomina to obliczenie powierzchni prostokąta: pole powierzchni prostokąta jest równe długości pomnożonej przez szerokość. Raport swoim kształtem przypomina prostokąt.

Tabela 1-9 Units Sold dla firmy Adventure Works według Model i Month

	Jan 2011	Feb 2011	Mar 2011	Apr 2011
Mountain-500	3	8	6	6
Road-750			15	16
Hitch Rack	1	6	6	3

Porównanie z prostokątem dotyczy jednak jedynie działań arytmetycznych związanych z obliczaniem liczby wartości, nie zaś z kształtem raportu. Można łatwo przekształcić raport, umieszczając atrybut Month w wierszach, a wtedy raport przyjmie postać pokazaną w tabeli 1-10.

Tabela 1-10 Units Sold dla firmy Adventure Works według Model i Month

Hitch Rack	Jan 2011	1
Hitch Rack	Feb 2011	6
Hitch Rack	Mar 2011	6
Hitch Rack	Apr 2011	3
Mountain-500	Jan 2011	3
Mountain-500	Feb 2011	8
Mountain-500	Mar 2011	6
Mountain-500	Apr 2011	6
Road-750	Jan 2011	
Road-750	Feb 2011	
Road-750	Mar 2011	15
Road-750	Apr 2011	16

Niezależnie od tego, czy wartości w raporcie tworzą prostokąt jak w tabeli 1-19, czy też ułożone są liniowo, jak w tabeli 1-10, to przy czterech elementach atrybutu Month i trzech elementach atrybutu Model mamy nadal 12 możliwych wartości. Nasz raport ma 12 możliwych wartości, gdyż elementy atrybutu Model i elementy atrybutu Month są niezależne. Każdy model ma swoją własną wartość sprzedaży dla każdego miesiąca – nawet jeśli wynosi ona zero.

Rozważmy nowy raport. Przypuśćmy, że Adventure Works sprzedaje produkty w dwóch różnych stanach i chcielibyśmy znać liczbę sztuk każdego modelu sprzedaną w każdym stanie. Ponadto, poza obejrzeniem tych wielkości według miesięcy, chcemy mieć sumę dla wszystkich miesięcy. Aby wygenerować ten raport, analityk doda

atrybut State (Stan) do wierszy raportu i zastąpi atrybut Month hierarchią atrybutów Month. Nowy raport pokazano w tabeli 1-11.

Tabela 1-11 Units Sold dla firmy Adventure Works według State, Model i Month

		All Months	Jan 2011	Feb 2011	Mar 2011	Apr 2011
WA	Hitch Rack	9		4	3	2
	Mountain-500	18	2	6	5	5
	Road-750	19			9	10
OR	Hitch Rack	7	1	2	3	1
	Mountain-500	5	1	2	1	1
	Road-750	12			6	6

Raport ma teraz dwa elementy w hierarchii atrybutu State trzy elementy atrybutu Model oraz pięć elementów atrybutu Month. Liczba możliwych wartości, 30, jest równa liczbie stanów pomnożonej przez liczbę modeli i przez liczbę miesięcy. Jest to analogiczne do obliczania objętości prostopadłościanu w świecie fizycznym: objętość liczymy, mnożąc długość przez szerokość i przez wysokość.

To oczywiste, że dla tego raportu porównanie dotyczy arytmetyki użytej do obliczeń, a nie fizycznego kształtu. Wartości w tabeli 1-11 są ułożone w prostokąt. Tabela 1-12 pokazuje kilka pierwszych wierszy tego samego raportu, z tym że hierarchię Months umieszczono w wierszach. Raport pokazuje te same informacje, które były w tabeli 1-11, ale wartości ułożone są liniowo.

Tabela 1-12 Units Sold dla firmy Adventure Works według State, Model i Month

WA	Hitch Rack	All Months	9
WA	Mountain-500	All Months	18
WA	Road-750	All Months	19
OR	Hitch Rack	All Months	7
OR	Mountain-500	All Months	5
OR	Road-750	All Months	12
WA	Hitch Rack	Jan 2011	
WA	Mountain-500	Jan 2011	2
WA	Road-750	Jan 2011	
OR	Hitch Rack	Jan 2011	1
OR	Mountain-500	Jan 2011	1
OR	Road-750	Jan 2011	
WA	Hitch Rack	Feb 2011	4
WA	Mountain-500	Feb 2011	6
WA	Road-750	Feb 2011	

Niezależnie od sposobu przedstawienia raportu, State, Model i Month są niezależnymi atrybutami i hierarchiami, zaś sumaryczną liczbę potencjalnych wartości obliczamy, mnożąc elementy pierwszego niezależnego atrybutu (dwa elementy States) przez liczbę elementów drugiego niezależnego atrybutu (trzy elementy Models) przez liczbę elementów trzeciego niezależnego atrybutu (pięć elementów Months).

Aby obliczyć długość, powierzchnię i objętość, mnożymy długości wymiarów obiektu. W celu obliczenia potencjalnych wartości w raporcie mnożymy liczbę elementów w niezależnych atrybutach i hierarchiach raportu. *Niezależne atrybuty i hierarchie są wymiarami raportu*, dlatego praktycy BI zapożyczyli termin *wymiar* z matematyki. Wymiary zawierają atrybuty i hierarchie. Na przykład raport w tabeli 1-11 wyświetla atrybut State należący do wymiaru Geography, atrybut Model należący do wymiaru Product oraz hierarchię atrybutu Month należącą do wymiaru Date. Choć termin *wymiar* pochodzi z matematyki, trzeba pamiętać, że wymiar analizy danych bardzo różni się od wymiaru fizycznego. Użycie terminu *wymiar* nie implikuje niczego na temat fizycznego ustawienia atrybutów i hierarchii w raporcie ani sposobu przechowywania danych wielowymiarowych w bazie danych.

Wymiar może zawierać więcej niż jeden atrybut, ale każdy wymiar ma jeden i tylko jeden atrybut klucza. Wszystkie inne atrybuty powiązane z atrybutem kluczowym należą do tego samego wymiaru. Ponieważ atrybuty należą do wymiarów, elementy członkowskie atrybutów są także elementami członkowskimi wymiarów. Na przykład Product jest kluczowym atrybutem wymiaru Product. Jeśli znamy wartość atrybutu Product, znamy także wartości atrybutów Model, Color i Size, więc atrybuty te należą także do wymiaru Product. Silver jest elementem członkowskim atrybutu Color, zaś 48 jest elementem członkowskim atrybutu Size, więc Silver i 48 są także elementami wymiaru Product.

Wymiary zawierają także hierarchie. W poprzedniej części zatytułowanej „Hierarchie w analizie danych" była mowa o tym, że hierarchie są tworzone przez atrybuty i zorganizowane w postaci poziomów. Atrybuty używane do tworzenia hierarchii muszą być w tym samym wymiarze, więc hierarchia będzie należeć do tego wymiaru. Na przykład hierarchia Product by Model jest zbudowana z atrybutów zawartych w wymiarze Product, więc hierarchia Product by Model także należy do wymiaru Product.

Wymiar, który zawiera miary, jest zbudowany inaczej niż inne wymiary. Nie ma atrybutu Measures (Miary) ani hierarchii utworzonych z miar. Zamiast tego jest tylko wymiar Measures, który zawiera listę etykiet atrybutów identyfikujących miary. Tabela 1-13 pokazuje raport, w którym elementy Units Sold i Sales Dollars (Sprzedaż w dolarach) wymiaru Measures są wyświetlone w kolumnach.

Tabela 1-13 Sprzedaż dla firmy Adventure Works według State, Model i Month

		Jan 2011		Feb 2011		Mar 2011		Apr 2011	
		Units	$	Units	$	Units	$	Units	$
WA	Hitch Rack			4	$480	3	$360	2	$240
	Mountain-500	2	$1,105	6	$3,265	5	$2,775	5	$2,750
	Road-750					9	$4,860	10	$5,400
OR	Hitch Rack	1	$120	2	$240	3	$360	1	$120
	Mountain-500	1	$565	2	$1,105	1	$540	1	$540
	Road-750					6	$3,240	6	$3,240

Units = Sprzedane sztuki, $ = Sprzedaż w dolarach

Raport zawiera czwarty wymiar. Liczba wartości w raporcie jest nadal równa iloczynowi liczby elementów w każdym wymiarze: 2 × 3 × 4 × 2, co daje 48. Jednak nie ma tam – i nie powinno być – żadnej analogii ze świata fizycznego. Pamiętajmy, że wymiary są po prostu wygodnym sposobem grupowania powiązanych ze sobą atrybutów i hierarchii, zaś istnienie czterech wymiarów (lub 20 czy 60) jest tak samo realistyczne jak istnienie trzech wymiarów. Dodawanie wymiarów zwiększa jedynie liczbę wartości wyświetlonych w raporcie czy też przechowywanych w bazie danych.

W świecie fizycznym mierzony obiekt zmienia się zależnie od liczby wymiarów. Na przykład, jednowymiarowy cal jest calem liniowym, dwuwymiarowy – calem kwadratowym, zaś trójwymiarowy – calem sześciennym. Cal sześcienny jest całkiem innym obiektem niż cal kwadratowy czy liniowy. W raporcie jednak mierzony obiekt nie zmienia się w miarę dodawania atrybutów i hierarchii z różnych wymiarów. Wartość liczbowa w raporcie „czterowymiarowym" i wartość liczbowa w raporcie „jednowymiarowym" jest taka sama. W świecie raportów dodatkowy wymiar tworzy po prostu nowy, niezależny sposób podziału miary.

Tabela 1-13 pokazuje, że dodanie czwartego wymiaru nie przekształca prostokątnego raportu w hiperkostkę, co jednak nie oznacza, że jest to czynność trywialna. Przypuśćmy, że zaczynamy od raportu dwuwymiarowego: 30 produktów i 12 miesięcy, czyli 360 możliwych wartości. Dodanie trzech nowych elementów do wymiaru produktu zwiększa liczbę wartości w raporcie do 396, czyli o 10 procent. Dodanie trzeciego wymiaru z trzema nowymi elementami zwiększa natomiast liczbę wartości do 1080, czyli jest to wzrost o 300 procent. Rozważmy taki skrajny przykład: mając 128 elementów w jednym wymiarze raport ma 128 elementów, ale tych samych 128 elementów podzielone na 64 wymiary – z dwoma elementami na wymiar – daje raport, zawierający 18 446 744 073 709 551 616 możliwych wartości!

Wprowadziliśmy tu wielowymiarową analizę danych. Model danych z wymiarami ma miary, wartości liczbowe, które można agregować oraz powiązane relacjami atrybuty i hierarchie zorganizowane w wymiary. Wymiary zapewniają metadane, które przekształcają dane liczbowe na informacje i umożliwiają analizę. Wymiary zapewniają liczbom opisowe etykiety i pozwalają nam poznać sposób grupowania, sortowania, formatowania i agregowania danych.

Model danych z wymiarami można zaimplementować w systemie zarządzania relacyjną bazą danych (RDBMS, SZRBD) lub w wielowymiarowej bazie danych OLAP. System BI zwykle obejmuje obie realizacje. Dane z systemów źródłowych są wybierane, przekształcane i ładowane do hurtowni danych przechowywanej w RDBMS. Następnie dane są wyciągane z hurtowni danych i ładowane do wielowymiarowej bazy danych OLAP. W dwóch kolejnych punktach przedstawiono sposób, w jaki model wymiarowy jest implementowany w obu typach bazy danych.

Wymiarowa hurtownia danych

Wiele systemów BI ma wymiarowe hurtownie danych jako swoją warstwę przechowywania i pobierania danych. Hurtownia danych jest przechowywana w systemie zarządzania relacyjną bazą danych (RDBMS). Na najprostszym poziomie można myśleć o relacyjnej bazie danych jako o zbiorze tabel. Każda tabela ma wiersze i kolumny, jak arkusz w programie Excel. W wymiarowej hurtowni danych wymiary są przechowywane w tabelach wymiarów. Miary są nazywane *faktami* i są przechowywane w *tabelach faktów*.

Tabela faktów

W wymiarowej hurtowni danych tabela, która przechowuje szczegółowe wartości dla miar czyli fakty, nosi nazwę tabeli faktów. Tabela 1-14 ilustruje tę koncepcję na podstawie kilku pierwszych wierszy tabeli faktów FactSales. Przechowuje Units Sold i Sales Dollars według State, Product i Month.

Tabela 1-14 Tabela faktów FactSales

State	Product	Month	UnitsSold	SalesDollars
OR	Hitch Rack	Jan 2011	1	$120.00
OR	Mountain-500 Silver, 40	Jan 2011	1	$565.00
OR	Mountain-500 Silver, 48	Jan 2011	1	$552.50
WA	Mountain-500 Silver, 48	Jan 2011	1	$552.50
OR	Hitch Rack	Feb 2011	2	$240.00
WA	Hitch Rack	Feb 2011	4	$480.00

W przykładowych wierszach z tabeli faktów, pierwsze trzy kolumny – State, Product i Month – to kolumny kluczowe. Wartości w kolumnach kluczowych wiążą fakty w wierszu tabeli faktów z wierszami w każdej tabeli wymiaru. Pozostałe dwie kolumny –UnitsSold i SalesDollars – zawierają fakty liczbowe. Każda kolumna w tabeli faktów jest zwykle albo kolumną kluczową, albo kolumną faktów, ale mogą być także inne kolumny dla celów odwołań – na przykład numery zamówień zakupów lub numery faktur.

Tabela faktów zawiera kolumnę dla każdej miary. Różne tabele faktów mają różne miary. Tabela faktów Sales może zawierać dwie kolumny miar – jedną dla Dollars, a drugą dla Units. Tabela faktów dla piętra w sklepie może zawierać trzy kolumny miar – dla Units (Sztuki), dla Minutes (Minuty) i dla Defects (Defekty).

Wiersze w tabeli faktów muszą być wszystkie na tym samym poziomie szczegółowości i muszą mieć relację do tego samego zbioru wymiarów. W tabeli FactSales pokazanej jako tabela 1-14, każdy wiersz musi zawierać dane miesięczne i musi mieć wartość dla stanu i dla produktu. Jeśli nasza organizacja ustala kwartalne kwoty sprzedaży dla każdej podkategorii produktu, dane muszą być przechowywane w innej tabeli faktów. Wymiary, z którymi powiązana jest tabela faktów (ma do niej relację), są określane jako *szczegółowość* tabeli faktów. Wymiary, z którymi związana jest tabela faktów, określane są jako *wymiarowość* tabeli faktów. Fakty, które mają różną szczegółowość, czyli różną wymiarowość, muszą być przechowywane w oddzielnych tabelach faktów. Bardzo często w hurtowni danych znajduje się wiele tabeli faktów.

Przykładowe wiersze w tabeli 1-14 ilustrują koncepcję układu tabeli faktów. Tabela faktów prawie zawsze dla każdego elementu wymiaru, zamiast opisowej nazwy, używa wartości całkowitej, zwanej *kluczem wymiaru*. Na przykład tabela 1-15 pokazuje tabelę faktów, w której kolumna State została zastąpiona kolumną StateKey, zaś wartości OR i WA, oznaczające stany, zastąpiono liczbami 1 i 2. W podobny sposób kolumny Product i Month zostały zastąpione przez kolumny kluczowe wymiaru, zaś wartości tekstowe w tych kolumnach zastąpiły wartości klucza wymiaru. Ponieważ tabela faktów może zawierać bardzo duża liczbę wierszy – w hurtowni danych o rozsądnej wielkości tabela faktów z łatwością może osiągnąć setki milionów wierszy – korzystanie z całkowitoliczbowego klucza wymiaru może znacząco zmniejszyć wielkość tabeli faktów.

Tabela 1-15 Tabela FactSales

StateKey	ProductKey	MonthKey	UnitsSold	SalesDollars
1	483	201101	1	120.00
1	591	201101	1	565.00
1	594	201101	1	552.50
2	594	201101	1	552.50
1	483	201102	2	240.00
2	483	201102	4	480.00

Gdy umieszczamy całkowitoliczbowe wartości klucza wymiaru w tabeli faktów, etykiety elementów wymiaru muszą zostać umieszczone w innej tabeli – tabeli wymiaru. Musimy mieć jedną tabelę wymiaru dla każdego klucza wymiaru w tabeli faktów.

Tabele wymiarów

Tabela wymiarów zawiera jeden wiersz dla każdego elementu członkowskiego w atrybucie kluczowym wymiaru. Atrybut kluczowy ma dwie kolumny: jedna z nich zawiera całkowitoliczbowy klucz wymiaru, zaś druga etykietę atrybutu. Tabela 1-16 pokazuje prostą wersję tabeli wymiaru produktu.

Tabela 1-16 Tabela wymiaru DimProduct

ProductKey	Product
483	Hitch Rack
596	Mountain-500 Black, 40
598	Mountain-500 Black, 44
599	Mountain-500 Black, 48
591	Mountain-500 Silver, 40
593	Mountain-500 Silver, 44
594	Mountain-500 Silver, 48
604	Road-750 Black, 44
605	Road-750 Black, 48

W hurtowni danych atrybut kluczowy w tabeli wymiarów musi zawierać unikatową wartość dla każdego elementu członkowskiego wymiaru. Według terminologii relacyjnej bazy danych atrybut kluczowy nazywany jest *kolumną klucza podstawowego*. Każda kolumna klucza wymiaru w tabeli faktów odpowiada kolumnie klucza podstawowego jednej z tabeli wymiaru. Każda wartość klucza, która pojawia się raz w tabeli wymiarów, może wielokrotnie pojawić się w tabeli faktów. Na przykład klucz Product Key 483 dla Hitch Rack, powinien pojawić się tylko w jednym wierszu tabeli wymiaru, ale może pojawić się w wielu wierszach tabeli faktów. Jest to określane jako *relacja jeden-do-wielu*. W tabeli faktów kolumna klucza wymiaru (która znajduje się po stronie „wielu" w relacji jeden-do-wielu) nazywana jest *kolumną klucza obcego*. Relacyjna baza danych używa odpowiadających sobie wartości z kolumny klucza obcego (w tabeli wymiarów) i kolumny klucza obcego (w tabeli faktów) do powiązania rekordów w tabeli faktów z odpowiadającymi im rekordami w tabeli wymiarów.

Tabela wymiarów będzie zwykle miała dodatkowe kolumny zawierające elementy członkowskie innych atrybutów wymiaru. Tabela 1-17 pokazuje tabelę wymiaru produktu z atrybutami Product, Subcategory, Category, Color, Size i List Price.

Tabela 1-17 Tabela wymiaru DimProduct

ProductKey	Product	Subcategory	Category	Color	Size	ListPrice
483	Hitch Rack	Bike Racks	Accessories			120.00
596	Mountain-500 Black, 40	Mountain Bikes	Bikes	Black	40	539.99
598	Mountain-500 Black, 44	Mountain Bikes	Bikes	Black	44	537.99
599	Mountain-500 Black, 48	Mountain Bikes	Bikes	Black	48	534.99
591	Mountain-500 Silver, 40	Mountain Bikes	Bikes	Silver	40	565.00
593	Mountain-500 Silver, 44	Mountain Bikes	Bikes	Silver	44	561.00
594	Mountain-500 Silver, 48	Mountain Bikes	Bikes	Silver	48	552.50
604	Road-750 Black, 44	Road Bikes	Bikes	Black	44	540.00
605	Road-750 Black, 48	Road Bikes	Bikes	Black	48	540.00

Wiele atrybutów wymiarów może być używanych do tworzenia grup rekordów wymiarów, a następnie powiązane fakty mogą być podsumowane dla każdej grupy. Na przykład rekordy wymiaru Product mogą zostać pogrupowane na kategorie Bikes i Accessories, a następnie można obliczyć liczbę Units Sold dla każdej kategorii. Atrybuty, których można użyć do tworzenia grup, nazywane są *agregowalnymi*, zaś wartość zsumowana określana jest jako *agregacja*. Agregacja to bardziej ogólny termin na *wartość sumy*. Określenie wartość sumy sugeruje, że otrzymano ją sumując pojedyncze wartości. Jednak gdy obliczamy agregację, możemy sumować pojedyncze wartości, lecz także liczyć je, znajdować minimum lub maksimum, średnią lub stosować do wartości inne rodzaje obliczeń.

Nie wszystkie atrybuty są agregowalne. Na przykład nie będzie użyteczne tworzenie grup na podstawie atrybutu List Price (Cena katalogowa), a następnie sumowanie liczby sztuk Units Sold dla każdej ceny, gdyż List Price ma wiele odrębnych wartości. Wymiar klienta często zawiera atrybuty, które nie są agregowalne: nazwa ulicy, numer telefonu, adres poczty elektronicznej i tak dalej. Te atrybuty, których nie można agregować, określane są jako *właściwości elementu*. Właściwości elementu są dodawane do raportów, aby zapewnić dodatkowe informacje tam, gdzie są już atrybuty agregowalne. Na przykład raport może wyświetlać sprzedaż według agregowalnego atrybutu Customer, a następnie podawać właściwości elementu jak Telephone Number i E-Mail Address.

Klucze zastępcze i wolno zmienne wymiary

Jednym z powodów, dla którego używamy dla elementów wymiaru klucza całkowitoliczbowego, jest zmniejszenie wielkości tabeli faktów. Ponadto klucz całkowitoliczbowy pozwala na to, aby pozornie podwójne elementy znajdowały się w tabeli wymiarów. Na przykład w wymiarze Customer można mieć dwóch różnych klientów nazywających się John Smith, ale każdy z nich będzie miał przypisany unikatowy klucz klienta Customer Key, co gwarantuje, że każdy element klucza pojawi się tylko raz w tabeli wymiarów.

Oczywiście, ponieważ hurtownia danych jest generowana przez ekstrakcję danych z działającego systemu, dwóch klientów o nazwisku John Smith i tak będzie miało unikatowe klucze. Jeden może być oznaczony C125423A a drugi F234654B. Są to *klucze aplikacji* lub *klucze biznesowe*, gdyż nadawane są przez aplikacje używane do prowadzenia biznesu. Jeśli mamy już unikatowe klucze dla każdego klienta (lub produktu czy regionu), to zastanówmy się, czy hurtownia danych naprawdę musi generować nowe klucze dla swoich celów, czy też może użyć kluczy aplikacji gwarantujących unikatowość?

Najlepsze hurtownie danych generują swoje własne klucze. Te dodatkowe, nadmiarowe unikatowe klucze są nazywane *kluczami zastępczymi*. Czasami ludzie przyzwyczajeni do pracy z eksploatacyjnymi bazami danych mają trudności ze zrozumieniem, dlaczego hurtownia danych ma generować klucze zastępcze, podczas gdy dostępne są natychmiast unikatowe klucze aplikacji. Są trzy powody tworzenia unikatowych kluczy zastępczych w hurtowni danych: klucze zastępcze mogą być liczbami całkowitymi, umożliwiają integrację danych z różnych systemów i pozwalają na stosowanie powoli zmiennych wymiarów.

Klucze zastępcze mogą być całkowite, nawet jeśli klucz aplikacji taki nie jest. W rezultacie tabela faktów hurtowni danych zajmuje mniej miejsca. Mniej miejsca w tabeli faktów potrzeba na zapisanie liczby całkowitej jak 54352 niż ciągu znaków jak C125423A. Jest to jednak najmniej ważny powód stosowania kluczy zastępczych.

Hurtownia danych integruje dane z wielu systemów źródłowych. Często systemy źródłowe mają różne klucze aplikacji dla tego samego elementu wymiaru lub też przeciwnie, ten sam klucz aplikacji dotyczy różnych elementów wymiaru. Jeśli generujemy klucz zastępczy dla każdego elementu wymiaru, nie musimy polegać na kluczach aplikacji. Na przykład w systemie Sales klucz aplikacji produktu A543 może odnosić się do roweru Mountain-100, zaś w systemie produkcji (utworzonym przez całkiem inną grupę ludzi) klucz aplikacji produktu A543 może odnosić się do roweru Road-650. Innym popularnym przykładem jest sytuacja połączenia się dwóch firm. W systemie sprzedaży jednej firmy klient C125423A może oznaczać klienta o nazwisku John Smith, ale w innym systemie sprzedaży ten sam numer C125423A może przypadkiem odnosić się do Abigail Torres. Nawet tak unikatowe wartości, jak amerykański numer ubezpieczeniowy, mogą być przyznane nowej osobie, jeśli rząd uważa, że poprzedni właściciel tego numeru zmarł. Używanie kluczy zastępczych w hurtowni danych przygotowuje hurtownię na takie ewentualności.

Jednym z najważniejszych powodów używania kluczy zastępczych w hurtowniach danych jest obsługa sytuacji, gdy wartość atrybutu zmienia się w czasie. Wymiary, w których atrybuty mogą się zmieniać noszą nazwę *wolno zmiennych wymiarów*. Na przykład w zeszłym roku cena katalogowa Hitch Rack wynosiła 110 $, ale teraz wzrosła do 120 $. W systemie przetwarzania zamówień, cena po prostu uległa zmianie ze 110 na 120 na głównej liście produktów, zaś nowe zamówienia używają nowych cen. W hurtowni danych trzeba rozważyć historię. Czy chcemy udawać, że Hitch Rack zawsze kosztował 120 $, czy też chcemy, aby w hurtowni danych był odzwierciedlony fakt, że w zeszłym roku cena wynosiła 110? Jeśli skorzystamy po prostu z klucza

aplikacji do reprezentowania Hitch Rack, nie mamy wielkiego wyboru. Z drugiej strony, jeśli przewidzieliśmy możliwość utworzenia dla produktu klucza zastępczego, po zmianie ceny katalogowej, tworzymy po prostu nowy rekord z nowym kluczem zastępczym dla droższej wersji tego samego produktu Hitch Rack, pozostawiając klucz aplikacji jako jeszcze jeden atrybut. Możliwość tworzenia wielu instancji jednego produktu, klienta lub innego elementu należącego do wymiaru jest niezwykle ważną korzyścią wynikającą z używania kluczy zastępczych. Jest zwłaszcza istotna w hurtowniach danych, w których utrzymujemy informacje historyczne.

Klucze zastępcze są ważną częścią projektu większości hurtowni danych. Klucz obcy w tabeli faktów i klucz podstawowy w tabeli wymiarów są wtedy pod pełną kontrolą hurtowni danych.

Alternatywne struktury tabel

W działającej bazie danych dane powinny być spójne w całej aplikacji. Jeśli zmienimy adres klienta w jednej części systemu, chcemy, aby zmiana adresu była natychmiast widoczna we wszystkich częściach systemu. Ta potrzeba spójności powoduje, że działające bazy danych dzieli się na wiele tabel, aby każda przechowywana wartość znajdowała się tylko raz w jednej tabeli. Za każdym razem, gdy wartość ta jest potrzebna, można utworzyć połączenie z tabelą, która tę wartość zawiera. Zapewnienie, aby wartość była przechowywana tylko w jednym miejscu, jest elementem procesu nazywanym *normalizacją* i jest bardzo istotne w działających systemach baz danych.

W wymiarze hurtowni danych możemy mieć wiele atrybutów, które tworzą naturalną hierarchię. Projektant bazy danych, który wie jak tworzyć działające bazy danych, będzie chciał normalizować wymiar tak, aby każdy poziom hierarchii miał oddzielną tabelę. Na przykład projektant może znormalizować wymiar Product, tworząc tabelę Subcategory (Podkategoria), w której każda podkategoria pojawia się tylko raz, oraz oddzielną tabelę Category (Kategoria), w której każda kategoria pojawia się tylko raz. Będzie to oczywiście wymagało obcych kluczy w tabelach Product i Subcategory, aby połączyć unikatowe klucze podstawowe odpowiednio w tabelach Subcategory i Category.

Jeśli jednak tworzymy raport za pomocą danych z hurtowni danych, wiele połączeń może spowolnić realizację zapytania. Przypuśćmy na przykład, chcemy zobaczyć Sales Amount dla kategorii Bikes w roku 2011. Do wykonać agregację Bikes, musimy połączyć ze sobą każdy wiersz w tabeli faktów z tabelą Product, a następnie z tabelą Subcategory i dalej z tabelą Category. Z kolei agregacja według roku 2011 wymaga połączenia tabeli faktów z tabelą Month, z tabelą Quarter i wreszcie z tabelą Year. Wszystkie te połączenia trzeba wykonać dla wszystkich wierszy tabeli faktów, odrzucić wiersze, które nie są powiązane z kategorią Bikes i z rokiem 2011, a następnie zsumować Sales Amount z pozostałych wierszy. Połączenie całych tabel wymiaru Product i wymiaru Date z tabelą faktów sprawia, że zapytanie wykonuje się znacznie wolniej, niż gdyby wszystkie atrybuty Product znajdowały się w jednej tabeli i wszystkie atrybuty Date też były w jednej tabeli

W konsekwencji w wielu hurtowniach danych wszystkie atrybuty jednego wymiaru są przechowywane w jednej tabeli wymiaru – nawet jeśli oznacza to, że elementy członkowskie atrybutu jak kategorie i lata są przechowywane nadmiarowo. Przechowywanie nadmiarowych wartości w jednej tabeli nazywane jest *denormalizacją* danych. Wykonanie jednego połączenia dla znalezienia Year i jednego dla znalezienia Category może być znacznie szybsze niż wykonanie wielu połączeń w celu otrzymania tych samych danych, więc denormalizacja tabel w hurtowniach danych jest często warta rozważenia.

Utworzenie jednej zdenormalizowanej tabeli dla każdego wymiaru daje w wyniku *schemat gwiazdy*. Gdy robimy diagram schematu gwiazdy, każda tabela faktu jest otaczana przez pojedyncze tabele dla każdego wymiaru, a wynik wygląda trochę jak gwiazda. Normalizacja każdej z tabel wymiaru tak, aby każdy wymiar miał kilka tabel daje w wyniku *schemat płatka śniegu*, gdyż „punkty" gwiazdy są podzielone na małe gałązki, wyglądające jak płatek śniegu. W rzeczywistości to nie hurtownia danych ma schemat gwiazdy lub płata śniegu, gdyż jeden wymiar może być w pełni znormalizowanym wymiarem w schemacie płatka śniegu, zaś inny wymiar tej samej hurtowni danych może być całkiem zdenormalizowanym wymiarem w schemacie gwiazdy.

Jeśli tworzymy hurtownię danych na potrzeby tworzenia raportów bezpośrednio z relacyjnej bazy danych, to powinniśmy uważnie rozważyć, czy każdy wymiar powinien być przechowywany w tabeli wymiaru w schemacie gwiazdy, czy też w wielu tabelach wymiaru płatka śniegu. Jeśli jednak używamy hurtowni danych przede wszystkim jako źródła danych dla Analysis Services, różnica między tabelami wymiarów gwiazdy i płatka śniegu ma mniejsze znaczenie i można przy podejmowaniu decyzji projektowych wziąć pod uwagę inne powody niż to, która struktura bazy danych jest łatwiejsza do tworzenia i aktualizacji.

Wymiar z niezrównoważoną hierarchią, jak wymiar Employee, wymaga nieco innego podejścia do projektu schematu wymiaru w układzie płatka śniegu. W wymiarze Employee elementami atrybutu kluczowego są wszyscy pracownicy. Jednak kierownik każdego pracownika jest także pracownikiem. W przeciwieństwie do standardowego wymiaru płatka śniegu, gdzie element nadrzędny jest na nowym poziomie i otrzymuje nową tabelę, w wymiarze Employee, element nadrzędny po prostu wskazuje inny element atrybutu kluczowego w oryginalnej tabeli wymiaru. Jest to określane jako wymiar *nadrzędny-podrzędny*, gdyż zarówno element nadrzędny, jak i podrzędny znajdują się w tej samej tabeli. W terminologii relacyjnej bazy danych, takie wskazywanie wstecz z jednego atrybutu do kluczowego atrybutu tej samej tabeli określane jest jako *połączenie samoodwołujące się*. Wymiar nadrzędny-podrzędny daje dużą elastyczność w sposobie organizowania hierarchii, ale kierowanie zapytań do takich połączeń może być trudne.

Nauczyliśmy się w tej części, jak wymiarowy model danych jest implementowany w hurtowni danych relacyjnej bazy danych. Hurtownia danych ma tabele faktów, które zawierają miary, zwane też *faktami*, oraz tabele wymiarów zawierające atrybuty. Projektant hurtowni danych może wybrać czy przechowywać hierarchię naturalną

w zbiorze tabel wymiarów schematu płatka śniegu, czy też przechowywać wszystkie atrybuty wymiaru w tabeli schematu gwiazdy. Wymiar z niezrównoważoną hierarchią może być zapisany w tabeli z połączeniem samoodwołującym się.

Nie można używać hurtowni danych relacyjnej bazy danych do implementacji wszystkich aspektów modelu wymiarowego, gdyż ilość metadanych, które można zapisać w relacyjnej bazie danych, jest ograniczona. W tabeli wymiaru nazwy atrybutów (nazwy kolumn) są metadanymi, ale nazwy elementów atrybutów są danymi w wierszach tabeli. Tabela wymiaru w schemacie gwiazdy nie zawiera metadanych dotyczących sposobu organizowania atrybutów w hierarchie. Na przykład, jeśli wymiar Product jest przechowywany w tabeli wymiaru w schemacie pojedynczej gwiazdy, w tabeli nie ma informacji pokazującej, że produkty są agregowane w podkategorie, a te są agregowane w kategorie. Oczywiście, można zorganizować naturalne hierarchie w zbiór tabel w schemacie płatka śniegu, ale nie mamy sposobu wskazania poziomów hierarchii, które nie są hierarchiami naturalnymi. Nie ma metadanych dotyczących sposobu, w jaki elementy atrybutu są posortowane ani określenia, które atrybuty są agregowalne, a które nie. Każda tabela faktów zawiera dane na jednym poziomie szczegółowości, nie ma więc wartości podsumowujących lub zagregowanych, a także nie ma metadanych na temat odpowiedniego typu agregacji, jaki należy zastosować w odniesieniu do faktów. Wreszcie w hurtowni danych dane wymiaru i faktów są przechowywane w oddzielnych tabelach. Gdy zapytanie przeszukuje hurtownię danych, trzeba w nim umieścić instrukcje, jak łączyć tabele wymiarów z tabelami faktów.

Ponieważ hurtownia danych relacyjnej bazy danych nie implementuje w pełni wszystkich aspektów modelu wymiarowego, pisząc raporty i zapytania trzeba wykorzystywać własną wiedzę na temat metadanych, które nie są zapisane w hurtowni danych, aby ominąć ten brak. Ponieważ metadane nie znajdują się w hurtowni danych, użytkownicy muszą włączać je do każdego zapytania i raportu, co prowadzi do pojawiania się ludzkich błędów, zaś raporty i zapytania mogą wyświetlać niespójne dane. Wielowymiarowa baza danych OLAP ma znacznie większe możliwości przechowywania metadanych, może więc być wykorzystana do implementacji modelu wymiarowego.

Wielowymiarowa baza danych OLAP

W systemie BI wielowymiarowa baza danych OLAP znajduje się pomiędzy magazynem danych a warstwą pobierania danych i warstwą prezentacji. Przekształca dane z relacyjnej hurtowni danych w zaimplementowany w pełni model wymiarowy, który pozwala na szybkie i elastyczne tworzenie raportów analitycznych i wizualizacji danych. Ponieważ różne bazy danych OLAP różnią się ze względu na zapewniane funkcje i ich fizyczną implementację, przedstawimy tu sposób, w jaki usługi Analysis Services implementują wymiarowy model danych.

Kolumny zawierające dane liczbowe w tabeli faktów odpowiadają miarom w modelu wymiarowym, więc każda tabela faktów stanowi grupę miar. Usługi Analysis Services

organizują informacje w logiczne struktury nazywane *grupami miar*, co odpowiada pojedynczej tabeli faktów i związanymi z nią wymiarami. Koncepcyjnie grupa miar przypomina tabele faktów, z kilkoma znaczącymi różnicami. Podobnie jak tabela faktów, grupa miar zawiera jedną kolumnę dla każdego wymiaru i jedną kolumnę dla każdej miary. Grupa miar zawiera także wiersz dla każdej możliwej kombinacji elementów każdego wymiaru. Inaczej niż w tabeli faktów, która zawiera dane tylko na jednym poziomie szczegółowości, grupa miar zawiera dane z jednego poziomu szczegółowości oraz zagregowane dane dla wszystkich wyższych poziomów szczegółowości. W tabeli 1-18 pokazano kilka przykładowych wierszy dla grupy miar z wymiarami State, Product i Date.

Tabela 1-18 Grupa miar Sales

State	Product	Date	Units Sold	Sales Amount
All	All	All	70	31 305
WA	All	All	46	21 235
WA	Bikes	All	37	20 155
WA	Road Bikes	All	19	10 260
WA	Road-750 Black, 48	All	13	7 020
WA	Road-750 Black, 48	2011	13	7 020
WA	Road-750 Black, 48	Mar 2011	4	2 160

Dolny wiersz tabeli 1-18 odpowiada wierszowi, który znajdowałby się w tabeli faktów – zawiera ten sam poziom szczegółowości jak tabela faktów. Żaden inny wiersz nie pojawi się w tabeli faktów – każdy z nich ma co najmniej jeden element z wyższego poziomu hierarchii wymiaru. W każdym wierszu, który ma element z wyższego poziomu, dane faktów zostały zagregowane przez zsumowanie szczegółowych danych z niższego poziomu. Tych kilka wierszy ilustruje tylko maleńką próbkę możliwych wartości, nawet w tak małej grupie miar.

Można zauważyć, że w tabeli 1-18 zarówno nagłówki kolumn, jak i dane wymiaru są wytłuszczone, zaś dane numeryczne są w zwykłej czcionce. Jest to celowe i ma pokazać, że zarówno nazwy wymiarów, jak i nazwy elementów wymiarów są metadanymi. Możemy także zobaczyć, że dane wymiarów są zorganizowane w hierarchie. Ponieważ przykład jest prosty, nie pokazuje on wszystkich sposobów, w jaki Analysis Services implementują wymiarowy model danych. W rozdziale 4., „Tworzenie wymiarów", zobaczymy, że można zastosować kolejność sortowania do elementów wymiaru i oznaczyć atrybuty jako agregowalne i nieagregowalne.

Analysis Services pozwala połączyć kilka powiązanych grup miar i zbiór wymiarów w *kostkę*. Nie każda grupa miar zawiera dane z każdego wymiaru, ale jeśli grupa miar znajduje się w kostce, nie może zawierać danych z wymiaru, który nie należy do kostki. Podobnie jak termin *wymiar*, termin *kostka* (dosłownie *sześcian*) został zapożyczony z matematyki. Liczba możliwych wartości w grupie miar stanowi iloczyn

liczby elementów w każdym wymiarze grupy miar, podobnie jak jednostki sześcianu jest iloczynem wymiarów jego brzegów. Tak więc iloczyn liczby elementów we wszystkich wymiarach kostki daje największy możliwy rozmiar jakiejkolwiek grupy miar w kostce. W świecie fizycznym termin *kostka* odnosi się do obiektu trójwymiarowego. W Analysis Services kostka to po prostu dane faktów i wymiarów zapisane na dysku, bez wpływu na to, ile wymiarów zawiera. Kostka może mieć nawet tylko jeden wymiar lub tyle wymiarów, ile nam potrzeba.

W tym rozdziale poznaliśmy powód korzystania z systemu analizy biznesowej oraz jego strukturę. System BI jest używany do wykonania wielowymiarowej analizy danych. Wymiarowy model przekształca dane liczbowe na informacje przez dodanie do liczb metadanych. Metadane są zawarte w wymiarach złożonych z atrybutów i hierarchii, które są powiązane z danymi liczbowymi są też nazywane *miarami*. Hurtownia danych może być wykorzystywana do częściowej implementacji modelu wymiarowego w relacyjnej bazie danych, ale wszystkie korzyści analityczne uzyskujemy, implementując model wymiarowy w wielowymiarowej bazie danych OLAP, takie jak Analysis Services.

W rozdziale 2. poznamy więcej informacji na temat korzyści, jakie dają bazy danych OLAP i określonych zalet Analysis Services.

ROZDZIAŁ 2

Omówienie OLAP i Analysis Services

W tym rozdziale nauczymy się, jak:

- Rozumieć definicję OLAP i korzyści, jakie wielowymiarowa baza danych OLAP może dodać do systemu BI.
- Rozumieć zalety usług Microsoft SQL Server 2008 Analysis Services.
- Rozumieć rolę usługi Analysis Services w platformie BI firmy Microsoft.
- Rozumieć narzędzia tworzenia baz danych Analysis Services i zarządzania nimi.

Analiza biznesowa (BI) stanowi sposób myślenia. Hurtownia danych to ogólna struktura przechowywania danych potrzebnych w dobrych rozwiązaniach BI. Jednak dane w hurtowni danych mają niewielkie zastosowanie bez przekształcenia ich na informacje, których potrzebują decydenci. W tym rozdziale poznamy najpierw zalety analitycznego przetwarzania online (OLAP) – jednej z najlepszych technologii służącej do przekształcania danych na informacje – a następnie dowiemy się, jak w usłudze Microsoft Analysis Services wdrażane są korzyści płynące z OLAP.

Pojęcie OLAP

Pierwsza wersja Analysis Services nosiła nazwę OLAP Services. Choć nazwa ta odzwierciedla cel produktu, a nie stosowaną technologię, sama technologia jest także istotna. Zrozumienie historii terminu *OLAP* może pomóc w zrozumieniu jego znaczenia..

W roku 1985, E. F. Codd ukuł termin *online transaction processing* (OLTP, *przetwarzanie transakcji online*) i zaproponował 12 reguł definiujących bazę danych OLTP. Jego terminologia została szeroko zaakceptowana jako standard w odniesieniu do baz danych używany w celu zarządzania codziennymi działaniami (transakcjami) firmy. W roku 1993, Codd wprowadził kolejny termin, *online analytical processing* (OLAP, *przetwarzanie analityczne online*), i ponownie zaproponował 12 kryteriów definiujących bazę danych OLAP. Tym razem jego kryteria nie zyskały szerszej akceptacji, w przeciwieństwie do terminu *OLAP* – wiele osób uznało termin OLAP za idealne określenie opisujące bazy danych zaprojektowane w celu ułatwienia analizy danych.

Niektórzy używają określenia OLAP jako synonimu wymiarowej hurtowni danych. Zwykle jednak *OLAP* oznacza specjalizowane narzędzia, które ułatwiają dostęp do hurtowni danych. Jednym określeniem, które jest zawsze związane z OLAP – lecz nie łączy się nigdy z relacyjnymi bazami danych – jest *kostka*. Różne narzędzia OLAP w różny sposób definiują, przechowują i zarządzają kostkami. Jeśli jednak słyszymy określenie *kostka*, to znaczy, że jesteśmy w świecie OLAP.

Jakie zalety ma kostka OLAP w porównaniu z relacyjnymi bazami danych? Zwykle narzędzia OLAP dodają do relacyjnej bazy danych trzy poniższe korzyści:

- Trwale krótki czas reakcji
- Zapytania oparte na metadanych
- Wzory w stylu z arkusza kalkulacyjnego

Zanim zajmiemy się Analysis Services, popatrzmy, jak OLAP ogólnie zapewnia te korzyści.

Trwale krótki czas reakcji

Jednym ze sposobów, w jaki OLAP zapewnia na trwałe krótki czas reakcji, stanowi obliczanie i przechowywanie zagregowanych wartości i wyników obliczeń wzorów podczas ładowania kostki. Istota pomysłu opiera się na tym, że kosztuje nas albo czas obliczeń podczas zapytania, albo możemy za to zapłacić z góry. OLAP pozwala nam zapłacić z góry, więc zapytania działają szybciej. Jeśli chodzi o fizyczne przechowywanie danych, narzędzia OLAP będą należeć do jednego z dwóch typów: modelu arkusza kalkulacyjnego lub modelu bazy danych. Przechowywanie Analysis Services to model bazy danych, jednak warto poznać także niektóre zagadnienia i korzyści modelu OLAP, opartego na arkuszu kalkulacyjnym.

W arkuszu kalkulacyjnym do dowolnej komórki można wstawić wartość lub wyrażenie, zwane formułą. Arkusze są bardzo użyteczne przy wykonywaniu złożonych obliczeń, gdyż pozwalają je dość dokładnie kontrolować. Jednak arkusz może przechowywać tylko ograniczoną liczbę wierszy i kolumn i jest z natury strukturą dwuwymiarową. W porównaniu z arkuszem kostka OLAP może zawierać znacznie więcej danych i zapewnia więcej wymiarów. W modelu OLAP opartym na arkuszu, zagregowane wartości i wyniki obliczeń są fizycznie przechowywane przez bazę danych w kostce. Ma to swoje dobre i złe strony. Dobra jest możliwość wstępnego obliczenia bardzo skomplikowanych wyrażeń i zapisania ich wyników w dowolnej komórce kostki. Ponieważ wyniki takich obliczeń są już gotowe, zapytania wykonywane są bardzo szybko. Z drugiej strony przechowywanie w kostce wszystkich zagregowanych wartości i wyników obliczeń powoduje ogromne zapotrzebowanie na pamięć. Ogranicza to wielkość kostki OLAP w związku z problemem *eksplozji danych*.

Historia człowieka, który wynalazł szachy, pomoże zilustrować problem eksplozji danych. Człowiek ten mieszkał w Indiach i według legendy miał na imię Sessa. Król Indii był pod wrażeniem gry w szachy i poprosił Sessę, aby powiedział, co chce

otrzymać w nagrodę. Jego żądanie było tak skromne, że król poczuł się urażony. Sessa poprosił o jedno ziarnko ryżu za pierwsze pole szachownicy, dwa za drugie, cztery za trzecie i tak dalej, podwajając liczbę ziaren za każde z 64 pól szachownicy. Oczywiście, gdy matematycy królewscy obliczyli, ile ryżu potrzeba, aby zapłacić wynagrodzenie, zdali sobie sprawę, że potrzebny byłby magazyn o wielkości 3 na 5 metrów pomnożony przez dwukrotną odległość do słońca (znali oni system miar i odległość do słońca). Według jednej z wersji legendy król rozwiązał problem, skracając Sessę o głowę. W innej wersji król był łagodniejszy i sprytniejszy. Dał Sessie worek, pokazał mu gdzie jest magazyn i polecił, aby odliczył swoją nagrodę – bez pośpiechu. Problem jaki Sessa przedstawił królowi to postęp geometryczny: gdy wartości rosną w sposób geometryczny, bardzo szybko stają się bardzo duże.

Wielkość kostki rośnie geometrycznie wraz z liczbą wymiarów. Stanowi to problem w przypadku OLAP, w których stosowany jest model arkusza kalkulacyjnego. Ponieważ teoretycznie dane mogą być przechowywane w dowolnej komórce przestrzeni kostki, eksplozja danych staje się poważnym problemem, z którym trzeba sobie poradzić. Im więcej wymiarów dołączymy do kostki i im więcej będzie elementów w każdym wymiarze (ziarnistość), tym większa jest możliwość eksplozji. Narzędzia OLAP oparte na arkuszu kalkulacyjnym obejmują zwykle rozbudowane i skomplikowane techniki zarządzania eksplozją danych; mimo to ich wielkość jest bardzo ograniczona.

Narzędzia OLAP oparte na arkuszu kalkulacyjnym są zwykle powiązane z aplikacjami finansowymi. Większość z nich obejmuje stosunkowo małe bazy danych w połączeniu ze skomplikowanymi, nieaddytywnymi obliczeniami.

Narzędzia OLAP, które przechowują dane kostki za pomocą modelu bazy danych, zachowują się zupełnie inaczej. Nie przechowują one wyników wyrażeń ani zagregowanych wartości. Zamiast tego korzystają z faktu, że większość raportów opiera się na dodawaniu, które jest operacją przemienną. Jeśli na przykład dodajemy liczby 3, 5 i 7, to nie ma znaczenia, czy przed dodaniem liczby 7 dodamy 3 do 5, otrzymując 8, czy też dodamy 5 i 7, otrzymując 12, a potem dodamy 3. W obu przypadkach ostatecznym wynikiem działania jest 15. W czysto relacyjnej bazie danych można szybko otrzymać wyniki zapytania dzięki utworzeniu tabel zagregowanych. W tabeli zagregowanej wstępnie podsumowujemy wartości, które będą potrzebne w raporcie. Na przykład tabela faktów zawierająca tysiące produktów, pięć lat codziennych danych i zapewne kilka innych wymiarów, może zawierać miliony wierszy. Utworzenie kwartalnego raportu sprzedaży według podkategorii produktów może wymagać kilku minut, nawet jeśli mamy tylko 50 podkategorii i 20 kwartałów. Jeśli jednak wstępnie podsumujemy dane w zagregowanej tabeli, która będzie zawierać jedynie podkategorie i kwartały, to będzie ona miała nie więcej niż 1000 wierszy (50 podkategorii razy 20 kwartałów daje maksymalnie 1000 możliwych wierszy), zaś raport żądający sum według podkategorii i kwartałów nie będzie wymagał długiego czasu na wykonanie. Ze względu na przemienną naturę dodawania, raport żądający sum według kategorii i według roku będzie mógł korzystać z tej samej tabeli zagregowanej, także bardzo szybko dając wyniki.

Zapewne największą korzyścią z zapisania OLAP za pomocą modelu bazy danych, jest możliwość uniknięcia eksplozji danych. Ponieważ do zapewnienia szybkich wyników zapytań potrzeba niewielu zagregowanych tabel, można mieć znacznie większe kostki z wieloma wymiarami i atrybutami, w porównaniu z modelem opartym na arkuszu kalkulacyjnym. Największą zapewne wadą OLAP realizowanego za pomocą modelu bazy danych jest brak wewnętrznego sposobu fizycznego przechowywania wartości będących wynikiem obliczeń. Wszystkie wyrażenia muszą być obliczane podczas wykonywania zapytania.

Nawet kostki OLAP zapisane za pomocą modelu bazy danych mogą szybko obliczać wartości niepodlegające prawu łączności. Na przykład Average Selling Price (Średnia cena sprzedaży) nie jest wartością addytywną – nie można po prostu dodać do siebie cen. Ale obliczanie Średniej wartości sprzedaży według Product Category (Kategoria produktów) będzie wymagało zsumowania Sales Amount (Wartość sprzedaży) i Sales Quantity (Ilość sprzedaży) dla każdej kategorii, a następnie podzielenia Wartości sprzedaży przez Ilość sprzedaży. Ponieważ obliczamy prosty stosunek dwóch wartości addytywnych, otrzymujemy wynik niemal tak szybko, jak przy pobieraniu prostej wartości.

Jedną z podstawowych zalet OLAP jest możliwość wcześniejszego obliczenia i zapisania wartości, aby raporty były tworzone bardzo szybko. Różne technologie OLAP korzystają z różnych technik zapisu, które mają różne mocne i słabe strony, lecz dobra baza danych OLAP zapewni odpowiedzi na zapytania znacznie szybciej niż relacyjna baza danych wszędzie tam, gdzie mamy do czynienia z wieloma zsumowanymi wartościami lub złożonymi wzorami obliczeniowymi.

Zapytania oparte na metadanych

Gdy wysyłamy zapytanie do relacyjnej hurtowni danych, korzystamy ze strukturalnego języka zapytań SQL (Structured Query Language). SQL jest świetnym językiem, ale został opracowany przede wszystkim dla systemów transakcyjnych, nie dla aplikacji tworzących raporty. Jednym z problemów z SQL nie jest sam język, lecz fakt, że relacyjne bazy dany dają niewiele informacji na temat sposobu przechowywania struktury danych, a co ważniejsze, co dane oznaczają. Innymi słowy relacyjne bazy danych zawierają bardzo niewiele metadanych. Większość informacji dotyczących baz danych musi pochodzić od nas – osoby piszącej zapytanie w języku SQL.

Z drugiej strony, baza danych OLAP zawiera bardzo wiele metadanych. Na przykład podczas tworzenia kostki OLAP definiujemy nie tylko, jakie są miary, ale także, jak należy je agregować, jaka ma być etykieta każdej z nich, a nawet jak należy formatować liczby. Podobnie, tworząc wymiar z wieloma atrybutami w bazie danych OLAP, efiniujemy, które atrybuty można agregować, oraz czy któreś z atrybutów powinny być wykorzystane do utworzenia hierarchii. Niestety SQL nie zapewnia możliwości napisania zapytania z użyciem dodatkowych metadanych zawartych w bazie danych OLAP. W rezultacie, jeśli źródłem danych jest OLAP, trzeba korzystać z innego języka zapytań: wielowymiarowych wyrażeń, czyli MDX.

Konieczność nauki nowego języka zapytań, którego można używać tylko z usługą Analysis Services może wydawać się niepokojąca. Na szczęście MDX został zaakceptowany przez wielu dostawców OLAP. W roku 2001 firmy Microsoft, Hyperion oraz SAS stworzyły radę, XML for Analysis (XMLA), aby sformułować wspólną specyfikację do komunikacji ze źródłami danych OLAP. Wybranym językiem zapytań zgodnym ze specyfikacją XMLA jest MDX. Większość dużych dostawców OLAP dołączyło do rady XMLA i są teraz dostawcami XMLA. (Więcej informacji na temat XMLA można znaleźć na witrynie komitetu *www.xmla.org*.) Tak więc wszystko, czego nauczymy się z tej książki o zapytaniach MDX, ma zastosowanie do Analysis Services, a także do innych baz danych OLAP.

Jedną z kluczowych zalet standardowego języka zapytań opartego na metadanych OLAP jest możliwość użycia przeglądarki OLAP ogólnego przeznaczenia do realizacji zapytań w dowolnej bazie danych OLAP. Na przykład, w przypadku kostki Microsoft Analysis Services można wybrać narzędzia klienta Microsoft, jak te dostępne w Microsoft Office, lub też wybrać narzędzia wielu innych dostawców. Każde z narzędzi klienckich korzystające z MDX lub XMLA może zrozumieć naszą kostkę i wygenerować sensowne raporty bez konieczności tworzenia zapytań dostosowanych do danego rozwiązania. Innymi słowy, ponieważ instrukcje zapytań MDX oparte są na metadanych zapisanych w kostce, można użyć narzędzi generujących zapytania bez konieczności samodzielnego pisania jakichkolwiek instrukcji zapytań MDX.

Jeśli zachodzi potrzeba napisania zapytań MDX dostosowanych do potrzeb, metadane sprawiają, że jest to łatwiejsze od pisania zapytań w języku SQL. Na przykład, gdy tworzymy w języku SQL zapytanie obliczające sumę Sales Units (Jednostki sprzedaży) dla klientów z każdego miasta, trzeba nadal dodać klauzulę zapewniającą odpowiednie uporządkowanie miast. W zapytaniu języka MDX po prostu podajemy, że potrzebujemy elementów atrybutu City (Miasto) i automatycznie otrzymujemy domyślną kolejność uporządkowania zgodnie z definicją zawartą w metadanych. Innym przykładem może być sytuacja, gdy w relacyjnej bazie danych zawierającej Country (Kraj) i City, nic nie sugeruje, czy miasta należą do krajów, czy kraje do miast. Jeśli chcemy pokazać wszystkie miasta Niemiec, musimy podać zapytanie o miasta, a następnie zawrzeć w nim klauzulę filtrującą dla słowa kluczowego Germany (Niemcy). W kostce OLAP, gdzie Country jest zdefiniowany jako nadrzędny względem City, można po prostu zapytać o elementy podrzędne dla Germany.

Jest to tylko niewielka próbka zapytań o wielkich możliwościach, które można pisać za pomocą języka MDX. Wiele innych zapytań trudno jest napisać w języku SQL – na przykład tabela pokazująca najlepiej sprzedające się produkty jako nagłówki kolumn, zaś przodujące w sprzedaży regiony jako nagłówki wierszy – są dość proste do utworzenia w MDX. Niektóre raporty, których po prostu nie można wykonać w SQL – jak zagnieżdżanie wielu warstw atrybutów jako nagłówki kolumn – można prosto utworzyć za pomocą MDX.