XI Konferencja PLOUG
Kościelisko
Październik 2005

Porównanie wydajności hurtowni danych ROLAP i MOLAP w Oracle 10g

Bartosz Bębel, Julusz Jezierski, Robert Wrembel

Politechnika Poznańska, Instytut Informatyki

{Bartosz.Bebel, Juliusz.Jezierski, Robert.Wrembel}@cs.put.poznan.pl

Streszczenie
Istotnym czynnikiem branym pod uwagę przy wyborze technologii składowania i przetwarzania danych w hurtowni danych jest wydajność. Do dyspozycji stoją dwa rozwiązania: wykorzystanie relacyjnego serwera bazy danych ogólnego przeznaczenia (ROLAP) albo dedykowanego serwera przetwarzania danych wielowymiarowych (MOLAP). System Oracle od kilku wydań wspiera oba te rozwiązania. Celem tego artykułu jest przestawienie wyników testów wydajności przetwarzania danych z wykorzystaniem obu rozwiązań, przeprowadzonych w oparciu o bazę danych z testu wydajnościowego TPC-H.

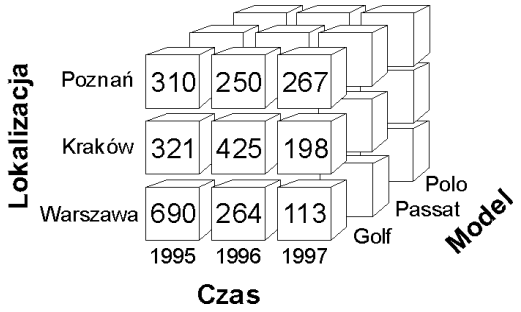
Informacja o autorach
Bartosz Bębel jest pracownikiem naukowym Instytutu Informatyki Politechniki Poznańskiej; obszar zainteresowań to magazyny danych i projektowanie systemów informatycznych. Kieruje projektem Sokrates - system wspierania dydaktyki na wyższej uczelni. Od 2000 r. prowadzi szkolenia w Oracle University.
Juliusz Jezierski urodził się w roku 1968 roku w Poznaniu. W roku 1992 ukończył studia na Wydziale Elektrycznym Politechniki Poznańskiej. Po zakończeniu studiów został zatrudniony w Instytucie Informatyki swojej macierzystej uczelni. Obecnie prowadzi zajęcia dydaktyczne, bierze udział w szeregu prac naukowych z zakresu baz danych oraz administruje systemy informatyczne działające w oparciu o serwery Oracle.
Robert Wrembel pracuje na stanowisku adiunkta w Instytucie Informatyki Politechniki Poznańskiej. Zawodowo zajmuje się bazami danych i magazynami (hurtowniami) danych. W latach 1996-2002 brał udział w realizacji 7 projektów informatycznych zarówno naukowo-badawczych, jak i typowo komercyjnych opartych o Oracle. Od 1998 roku prowadzi szkolenia w Centrum Edukacyjnym Oracle Polska. Jest autorem ponad 70 publikacji, krajowych i zagranicznych, w tym trzech książek dotyczących systemu Oracle. Od roku 1999 zasiada w Zarządzie Stowarzyszenia Polskiej Grupy Użytkowników Systemu Oracle.

1. Wprowadzenie

Na efektywność przetwarzania zapytań analitycznych w hurtowni ma wpływ m.in. zastosowany model danych, czyli reprezentacji danych w systemie. W praktyce wykorzystuje się dwa podstawowe modele reprezentacji i przechowywania danych w hurtowni: *relacyjny*, zwany również ROLAP (ang. Relational OLAP) i *wielowymiarowy*, zwany również MOLAP.

Hurtownia danych w technologii **ROLAP** jest implementowana w postaci tabel, których schemat posiada najczęściej strukturę gwiazdy (ang. star schema) lub płatka śniegu (ang. snowflake schema) lub konstelacji faktów (ang. fact constellation) [BWZ04].

W technologii **MOLAP** do przechowywania danych wykorzystuje się wielowymiarowe tablice, popularnie zwane kostkami (ang. multidimensional arrays, datacubes). Tablice te zawierają wstępnie przetworzone (m.in. zagregowane) dane pochodzące z wielu źródeł. Przykładowa 3-wymiarowa tablica została przedstawiona na rysunku 1. Zawiera ona trzy wymiary: *Lokalizacja*, *Czas* i *Samochody VW*, a poszczególne komórki kostki, tzw. miara (ang. measure), przechowują informację np. o łącznej liczbie sprzedanych sztuk wybranych modeli w poszczególnych latach, w wybranych miastach.



Rys. 1. Przykładowa trójwymiarowa tablica opisująca miarę *liczba_sztuk* w trzech wymiarach: *Lokalizacji*, *Czasu* i *Modelu*

Wybór właściwego modelu do właściwego rodzaju analizy danych będzie miał wpływ na szybkość przetwarzania danych. W ramach niniejszego artykułu zostaną przedstawione charakterystyki efektywnościowe modelu ROLAP i MOLAP w implementacji *Oracle10g*.

2. MOLAP w Oracle10g

2.1. Obiekty modelu wielowymiarowego

Logicznymi składnikami modelu wielowymiarowego *Oracle OLAP* są: zmienne potocznie zwane kostkami, miary, wymiary, hierarchie, poziomy i atrybuty.

Kostka jest zbiorem *komórek*, określanych mianem *faktów* i reprezentujących elementarne jednostki informacji, będące przedmiotami analiz (np. sprzedaż produktu). Atrybutami faktów są *miary*, będące najczęściej wartościami numerycznymi (np. cena jednostkowa sprzedaży). Miary, umieszczone w tej samej kostce, mają identyczne powiązania do innych logicznych obiektów schematu wielowymiarowego i mogą być razem analizowane i prezentowane.

Wymiary tworzą krawędzie kostki i są informacjami *referencyjnymi*, określającymi kontekst analiz miar. W modelu wielowymiarowym każda miara jest powiązana z kilkoma wymiarami, dlatego każda wartość miary jest określana przez kilka *wystąpień* (instancji) wymiarów. Wymiar najczęściej posiada *strukturę hierarchiczną*, określającą sposób agregacji wartości skojarzonych z nim miar. Typowa hierarchia wymiaru składa się z *poziomów*, jednak są sytuacje, w których

hierarchia wymiaru nie posiada poziomów, gdyż powiązania referencyjne typu nadrzędny-podrzędny pomiędzy wystąpieniami wymiaru nie pozwalają na zdefiniowanie poprawnych poziomów. Taki wymiar określany jest często mianem *wymiaru bazującego na wartościach* (ang. value-based dimension). Jeśli wymiar nie posiada hierarchii i poziomów, nosi nazwę *wymiaru płaskiego* (ang. flat dimension).

Ostatni element modelu wielowymiarowego, *atrybut*, dostarcza dodatkowych informacji o instancjach wymiaru.

2.2. Przestrzeń analityczna

Dane modelu wielowymiarowego *Oracle OLAP* umieszczane są w tzw. *przestrzeniach analitycznych* (ang. analytic workspaces). W instancji bazy danych może istnieć wiele przestrzeni analitycznych, każda z nich jest własnością określonego użytkownika bazy danych, natomiast pozostali użytkownicy mogą uzyskać dostęp do przestrzeni po nadaniu im odpowiednich uprawnień. *Oracle10g* składuje przestrzenie analityczne w tabelach specjalnego schematu relacyjnego, tabele te mogą być zarządzane w taki sam sposób jak zwykłe tabele bazy danych. Obiekty fizyczne, implementujące obiekty logiczne, obecne w modelu wielowymiarowym danej przestrzeni analitycznej, umieszczane są w rekordach tabel jako wartości atrybutów typu LOB.

Firma Oracle dostarcza narzędzie o nazwie *Analytic Workspace Manager*, które przy wykorzystaniu graficznego interfejsu użytkownika umożliwia tworzenie i zarządzanie przestrzeniami analitycznymi. Opis tych przestrzeni jest realizowany wg. standardu *CWM2* (Common Warehouse Metamodel) [OMG, VVS00]. Dostęp do metadanych z poziomu SQL jest możliwy przy wykorzystaniu *Aktywnego Katalogu* (ang. Active Catalog), będącego zbiorem perspektyw relacyjnych.

Proces tworzenia przestrzeni analitycznej składa się z następujących kroków:

1. Zdefiniowanie przestrzeni analitycznej: określenie nazwy nowej przestrzeni oraz wskazanie schematu, w którym przestrzeń będzie utworzona; po zatwierdzeniu podanych informacji następuje utworzenie we wskazanym schemacie zbioru relacji i innych obiektów w standardowej formie bazodanowej,
2. Zdefiniowanie obiektów logicznych, które utworzą wielowymiarowy schemat: kostek, miar, wymiarów, poziomów, hierarchii i atrybutów, obiekty logiczne zostają zaimplementowane w postaci fizycznych obiektów bazy danych,
3. Określenie źródeł danych dla poszczególnych obiektów modelu (tabel lub perspektyw schematów relacyjnych),
4. Załadowanie danych ze źródeł danych bezpośrednio do schematu wielowymiarowego (krok ten jest powtarzany cyklicznie, w miarę zachodzenia zmian danych w źródłach modelu).

Poniższe sekcje omawiają dokładnie procesy definiowania logicznych obiektów przestrzeni analitycznej, określania źródeł danych dla obiektów oraz ładowania danych ze źródeł.

2.3. Definiowanie obiektów logicznych przestrzeni analitycznej

2.3.1. Definiowanie wymiarów

Pierwszym krokiem procesu definiowania logicznych obiektów przestrzeni analitycznej jest utworzenie wymiarów. Wyróżniamy dwa rodzaje wymiarów: *wymiary użytkownika* oraz *wymiary czasowe*.

Wymiar użytkownika jest standardowym wymiarem przestrzeni analitycznej, określającym kontekst analiz i może być wymiarem płaskim, wymiarem bazującym na wartościach lub standardowym wymiarem z hierarchią poziomów. Każde wystąpienie wymiaru musi być unikalną wartością, natomiast sam wymiar do zapewnienia unikalności może wykorzystywać *klucze naturalne* lub *klucze sztuczne*. Klucze naturalne są wartościami pobieranymi bezpośrednio z tabel źródło-

wych bez żadnych zmian, wartości te muszą być unikalne w zbiorze wszystkich poziomów wymiaru. Unikalność nie musi być jednak wymuszana w tabelach źródłowych dla wymiaru, gdyż każdy poziom może być połączony z inną kolumną tabeli źródłowej. Klucze naturalne są wymagane dla wymiarów płaskich i bazujących na wartościach. Z kolei klucze sztuczne wymuszają unikalność instancji poziomów przez dodanie do wartości pobranej z kolumny tabeli źródłowej przedrostka będącego nazwą poziomu docelowego dla wartości. Wymiar z kluczem sztucznym musi posiadać przynajmniej jedną hierarchię poziomów.

Specjalnym rodzajem wymiaru jest wymiar czasowy. Jeśli przestrzeń analityczna ma wspierać analizy przebiegów czasowych (ang. time-series analysis), np. porównania z wcześniejszymi okresami czasowymi, wymiar czasowy musi umożliwiać pełną definicję okresów czasowych. To pociąga za sobą konieczność istnienia w tabeli źródłowej dla wymiaru czasowego kolumn, określających datę końca oraz rozpiętość okresu czasowego (ang. time span). Jeśli te informacje nie są dostępne, wówczas wymiar przechowujący czas może zostać zdefiniowany jako zwykły wymiar użytkownika, jednak w tym przypadku nie będzie możliwości realizacji analiz bazujących na czasie.

2.3.2. Definiowanie poziomów

Definiowanie poziomów przebiega tylko w przypadku, gdy już zdefiniowany wymiar jest wymiarem bazującym na poziomach. W tej sytuacji poziomy mogą być powiązane zależnościami typu nadrzędny-podrzędny lub jeden-do-wielu. Dla każdego poziomu konieczne jest zidentyfikowanie źródła danych dla instancji wymiaru na tym poziomie.

2.3.3. Definiowanie hierarchii

Wymiar może posiadać jedną lub więcej hierarchii (jak już wspomniano, mogą również istnieć wymiary nie posiadające hierarchii). Najczęstszym typem hierarchii jest hierarchia bazująca na poziomach. Dla tego rodzaju hierarchii *Oracle OLAP* wspiera następujące typy hierarchii:

- *hierarchia normalna* – składa się z jednego lub większej liczby poziomów agregacji. Instancje poziomu podrzędnego połączone są z instancjami poziomu nadrzędnego zależnościami wiele-do-jednego, przez co instancje poziomów podrzędnych „zwijają się" do instancji poziomów nadrzędnych, które z kolei zwijają się do instancji swoich poziomów nadrzędnych, itd. aż do poziomu szczytowego,
- *hierarchia wadliwa* (ang. ragged hierarchy) – zawiera co najmniej jedną instancję poziomu z inną bazą, przez to tworząc „wadliwy" poziom bazowy hierarchii,
- *hierarchia z brakującym poziomem* – zawiera co najmniej jedną instancję, której instancja nadrzędna umieszczona jest wyżej niż jeden poziom w górę w hierarchii wymiaru, przez to powstaje „dziura" w hierarchii.

Oracle OLAP wspiera również tworzenie hierarchii bazujących na wartościach (takich, w których nie jest możliwe wyróżnienie poprawnych poziomów), w tym wypadku należy jednak pamiętać, że wymiar, posiadający taką hierarchię, musi korzystać z kluczy naturalnych.

2.3.4. Definiowanie atrybutów

Atrybuty pozwalają na przechowywanie dodatkowych informacji, opisujących instancje wymiarów. Wyróżniamy *atrybuty definiowane automatycznie* oraz *atrybuty użytkownika*.

Atrybuty definiowane automatycznie są tworzone przez *Analytic Workspace Manager* podczas tworzenia wymiaru. Każdy wymiar posiada atrybuty, pozwalające na umieszczenie długiego oraz krótkiego opisu instancji wymiaru. Atrybuty te wykorzystywane są przez narzędzia, służące do przeprowadzania analiz danych modelu wielowymiarowego. Wymiar czasowy jest automatycznie uzupełniany atrybutami określającymi datę końca okresu oraz długość okresu.

Atrybuty użytkownika są tworzone przez użytkownika i pozwalają na uzupełnienie definicji wymiaru o dodatkowe informacje.

2.3.4. Definiowanie kostek

Pierwszym krokiem procesu definiowania kostki jest określenie jej nazwy oraz wskazanie zdefiniowanych wcześniej wymiarów, które utworzą krawędzie kostki. Kolejny krok to zdefiniowanie miar, jakie mają być obecne w kostce. Oprócz zwykłych miar, których wartości będą pobierane z kolumn tabel źródłowych, istnieje możliwość zdefiniowania w kostce tzw. *miar wyliczanych*. Wartości miar wyliczanych nie są pobierane z tabel źródłowych, a są wynikiem formuł zdefiniowanych przez użytkownika i składowanych w przestrzeni analitycznej. *Oracle OLAP* udostępnia szeroki zestaw funkcji, pogrupowanych w zbiory:

- podstawowa arytmetyka: dodawanie, odejmowanie, mnożenie, dzielenie, proporcja;
- zaawansowana arytmetyka: suma kumulowana, pozycja, wariancja, udział, i inne;
- porównania: różnica z poprzednim okresem, procentowa różnica z poprzednim okresem, wartość przyszła i inne;
- ramy czasowe: moving average, moving maximum, moving sum, i inne.

Wartości miar wyliczanych nie są składowane, ale wyliczane na bieżąco w przypadku. gdy użytkownik umieści je w zapytaniu analitycznym.

2.4. Określanie źródeł danych dla obiektów przestrzeni analitycznej

Źródłami danych dla logicznych obiektów przestrzeni analitycznej: wymiarów i miar, są tabele, znajdujące się w tym samym schemacie relacyjnym, lub umieszczone w różnych schematach. Określenie źródeł odbywa się po utworzeniu definicji obiektów przy pomocy narzędzia *Analytic Workspace Manager*. Narzędzie pozwala na wskazanie kolumny w tabeli w schemacie relacyjnym, z której dany obiekt logiczny będzie pobierał dane. Należy podkreślić, że narzędzie nie pozwala na realizację transformacji danych (np. określenie, że źródłem danych dla miary *LiczbaSztuk* ma być zsumowana wartość kolumny *LiczbaSztuk* z tabeli *SprzedażAktualna* z kolumną *LiczbaSztuk* z tabeli *SprzedażHistoryczna*). Jeśli takie transformacje są konieczne, należy w schemacie relacyjnym utworzyć perspektywy, które te transformacje zrealizują, i dopiero kolumny perspektyw wskazać jako źródła danych dla obiektów przestrzeni analitycznej.

Schemat relacyjny, zawierający tabele źródłowe dla wymiarów, może być schematem typu gwiazda, płatek śniegu lub dowolnym innym schematem relacyjnym, w którym występują zależności typu nadrzędny-podrzędny pomiędzy tabelami/perspektywami. Jeśli wymiar zawiera dodatkowe atrybuty, również dla nich definiuje się źródła danych wskazując kolumny w odpowiednich tabelach schematu relacyjnego.

Z kolei źródłem danych dla miar, tworzących kostkę, może być dowolna tabela lub perspektywa, zawierająca odpowiednie dane.

2.5. Ładowanie danych ze źródeł do przestrzeni analitycznej

Ostatnim krokiem procesu tworzenia przestrzeni analitycznej jest załadowanie danych ze źródeł relacyjnych do obiektów przestrzeni analitycznej. Po załadowaniu danych konieczne jest również wyliczenie agregatów (o ile są zdefiniowane w przestrzeni analitycznej). Ładowanie danych musi być powtarzane cyklicznie, aby zapewnić stały dopływ aktualnych danych do schematu wielowymiarowego przestrzeni.

Definicja i realizacja procesu ładowania danych odbywa się w narzędziu *Analytic Workspace Manager* w jednym z trzech scenariuszy. Pierwszy z nich to natychmiastowe uruchomienie procesu przez użytkownika. W tym przypadku dane, pobrane z tabel źródłowych, są natychmiast ładо-

wane do obiektów przestrzeni analitycznej. Drugi scenariusz zakłada zdefiniowanie zadań w kolejce zadań SZBD Oracle, które będą odpowiedzialne za cykliczne uruchamianie procesu ładowania danych bez konieczności jego inicjacji przez użytkownika. Wreszcie trzeci scenariusz polega na utworzeniu skryptu SQL, zawierającego polecenia ładowania danych, celem jego późniejszego uruchomienia.

Po zakończeniu procesu ładowania danych, schemat wielowymiarowy przestrzeni analitycznej jest gotowy do realizacji analiz danych.

2.6. Analizy danych

Przestrzeń analityczna *Oracle OLAP* umożliwia realizację szeregu analiz danych, służących pozyskaniu wiedzy pomagającej w podejmowaniu decyzji biznesowych. Analizy można podzielić na *analizy historyczne* oraz *prognozowanie przyszłych trendów*.

Analizy historyczne polegają na wydawaniu zapytań do danych historycznych, zgromadzonych w schemacie wielowymiarowym. Narzędzia do realizacji analiz, dostarczane przez firmę Oracle, to m.in. *Oracle BI Discoverer Plus OLAP* oraz *Oracle BI SpreadSheet Add-In*. *Discoverer Plus OLAP* umożliwia analitykom prezentację danych wielowymiarowych w różnych ujęciach, realizację typowych operacji modelu wielowymiarowego, takich jak drążenie, obracanie, i inne, a także wizualizację danych w postaci różnorakich wykresów. Z kolei *SpreadSheet Add-In* umożliwia pracę z danymi wielowymiarowymi w formie podobnej do pracy z arkuszem programu Microsoft Excel. Oba narzędzia pozwalają na konstruowanie skomplikowanych zapytań analitycznych bez znajomości języka SQL, przy wykorzystaniu szeregu kreatorów i szablonów zapytań.

Do realizacji analiz, polegających na prognozowaniu przyszłości na podstawie danych historycznych, służy znane już nam narzędzie *Analytic Workspace Manager* oraz narzędzie *OLAP Worksheet*. *Oracle OLAP* udostępnia szereg metod prognozowania, takich jak m.in. prosta regresja liniowa, kilka metod regresji nieliniowej czy też metodę Holta-Wintersa. Analityk może wskazać, która z metod ma zostać wykorzystana w prognozowaniu, może również pozostawić wybór systemowi, który spróbuje dobrać metodę najbardziej pasującą do danego przypadku.

Pierwszym krokiem przy prognozowaniu jest zdefiniowanie *przyszłych okresów czasowych*, dla których mają zostać znalezione prognozy. Użytkownik musi zapewnić, aby wymiar czasowy w schemacie wielowymiarowym zawierał odpowiednią perspektywę czasową. Może to zrealizować, dodając odpowiednie dane do tabeli źródłowej dla wymiaru czasowego (konieczne jest następnie uruchomienie procesu ładowania danych) lub dodając instancje wymiaru czasowego bezpośrednio w narzędziu *Analytic Workspace Manager*.

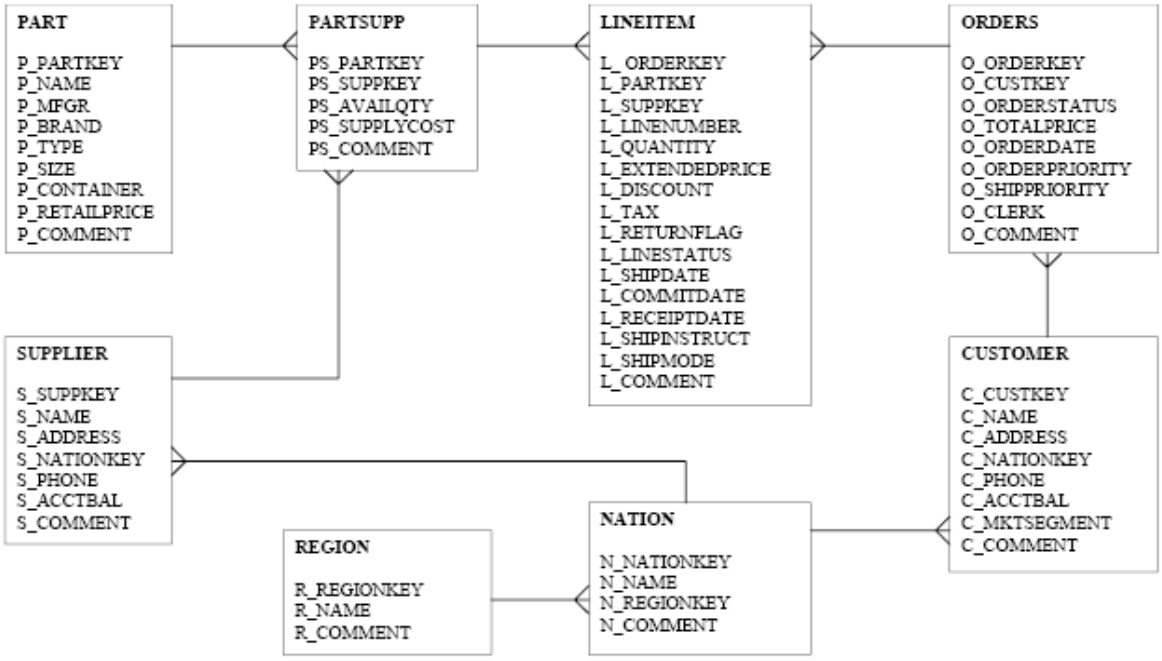
Następnie należy zdefiniować *miarę*, która przechowa znalezione podczas prognozowania wyniki. Miara może być częścią analizowanej kostki lub może należeć do nowej kostki.

Kolejnym krokiem procesu prognozowania jest utworzenie odpowiedniego *programu*, zawierającego komendy sterujące analizą. Komendy te m.in. pozwalają na określenie parametrów procesu prognozowania, takich jak rodzaj użytej metody, ilość okresów czasowych, uruchamiają samo prognozowanie i powodują składowanie uzyskanych wyników we wskazanej miarze. Program może być definiowany w obu wymienionych wcześniej narzędziach, jednak jego uruchomienie jest możliwe jedynie w *OLAP Worksheet*. Program generuje wyniki prognozy, zwykle są to dane na najniższym stopniu szczegółowości, dane mogą być następnie zagregowane wg wskazanych ścieżek agregacji.

3. Testy efektywnościowe

3.1. Scenrariusz testów i środowisko testowe

W celu porównania wydajności hurtowni danych ROLAP i MOLAP w *Oracle10g* przyjęto dwa podstawowe kryteria oceny testów wydajnościowych, tj. **czas ładowania danych** do hurtowni oraz **czas odpowiedzi** na wykonanie przykładowego zapytania analitycznego. Pierwsze kryterium stanowi miarę oceny efektywności przetwarzania z punktu widzenia administratora systemu i opisuje wielkość zasobów (procesory, dyski) niezbędną do przygotowania danych do analizy. Drugie kryterium stanowi miarę oceny efektywności z punktu widzenia użytkownika systemu i opisuje komfort pracy użytkownika za pomocą aplikacji interakcyjnych.



Rys. 2. Schemat testowej hurtowni danych wg. standardu TPC-H

Na potrzeby eksperymentu zbudowano hurtownię danych ROLAP zasiloną danymi pochodzącymi z testu wydajnościowego TPC-H [TCPH]. Schemat testowej hurtowni danych przedstawiono na rysunku 2. Znaczna złożoność schematu zapytań (8 tabel) dobrze eksponuje różnice w efektywności ROLAP i MOLAP. Należy podkreślić, że zastosowanie schematu z testu TPC-H ma na celu przetestowanie MOLAP i ROLAP z wykorzystaniem realistycznego grafu połączeń, liczebności dziedziny atrybutów oraz selektywności połączeń. Opis schematu źródeł danych dla testu TPC-H oraz rozmiary tabel przedstawiono w tablicy 1.

Eksperyment uruchomiono na komputerze PC Intel 2x1200Mhz, 2 GB RAM, 2 dyski SCSI 10.000RPM, działającego po kontrolą Linuxa dystrybucji Fedora Core 3 na bazie danych Oracle w wersji 10.1.0.3. Parametry instancji Oracle PGA_AGGREGATE_TARGET i SGA_TARGET ustawiono odpowiednio na 192M i 580M.

Tablica 1. Rozmiary tabel źródłowych wykorzystanego testu TPC-H

tabela	Liczba wierszy	objętość[KB]	klucz podstawowy
Region	5	0.4	regionkey
Nation	25	2	nationkey

Supplier	10K	1,300	suppkey
Customer	150K	23,000	custkey
Order	1,500K	158,000	orderkey
Part	200K	23,000	partkey
Lineitem	6,000K	673,000	orderkey+linenumber
PartSupp	800K	111,000	suppkey+partkey

ROLAP zaimplementowano w postaci tabel, zgodnie z opisem z testu wydajności TPC-H, oraz zbioru indeksów B-drzewo zdefiniowanych na kluczach podstawowych i obcych tych tabel.

W MOLAP zaimplementowano 3 główne wymiary (czas, produkt, dostawca), każdy wymiar główny był konkatenacją od trzech do czterech wymiarów podstawowych (np. czas był konkatenacją wymiaru: "cały okres", rok, miesiąc, dzień). Konkatenacja wymiarów prostych posłużyła do zamodelowania hierarchii wymiarów (np. "cały okres" składa się z lat, rok składa się z miesięcy, miesiąc składa się z dni). Fakty zostały zaimplementowane w postaci zmiennej wykorzystującej skompresowany kompozyt wymiarów: czas, produkt i dostawca.

3.2. Wyniki

Pierwszym testem było porównanie czasu ładowania danych do hurtowni (por. tablica 2). W przypadku ROLAP uzyskano czas 2,5 razy krótszy niż w przypadku MOLAP. Różnicę tą należy tłumaczyć złożonymi strukturami danych wykorzystywanymi przez MOLAP, w szczególności skompresowanymi kompozytami, których kompresja znacznie obciążała zasoby systemu.

Tablica 2. Wyniki testów porównawczych implementacji ROLAP i MOLAP

Kryterium [hh:mi:ss]	MOLAP	ROLAP
Ładowanie danych	00:17:01	00:06:31
Materializowanie OLAP	01:30:19	05:47:58
Zapytanie o pojedynczą wartość w kostce	00:00:00.13 00:00:00.04	00:00:00.51 00:00:00.03
Zapytanie o wszystkie ściany i krawędzie	00:01:12 00:00:02	00:00:16 00:00:01
Objętość [bloki bazy danych –8KB]:	135 987	723 968

Drugi test obejmował zapytanie o pojedynczy fakt na podstawie określonych wartości wymiarów. Test ten wykonano dla dwóch przypadków, w pierwszym przypadku zapytanie wykonano przy pustym buforze danych, w drugim przypadku bufor był wypełniony danymi przetwarzanymi przez poprzednie zapytanie. W obu przypadkach czas odpowiedzi zarówno MOLAP i ROLAP wynosi poniżej 1 sekundy. Są to wyniki wystarczające dla wydajnej obsługi zapytań ad-hoc generowanych przez aplikacji interakcyjne.

Trzeci test dotyczył czasu materializacji agregatu. Wybrany agregat był sumą sprzedaży produktów na wszystkich poziomach głównych wymiarów. W ROLAP agregat zaimplementowano w postaci zmaterializowanej perspektywy. Na perspektywie zdefiniowano indeksy bitmapowe na każdej z kolumn opisującej wymiar. W MOLAP do przechowywania agregatów posłużyła ta sama zmienna wykorzystywana do reprezentacji zbioru pojedynczych faktów. W teście tym MOLAP charakteryzuje się ponad 3,5 razy krótszym czasem materializacji agregatu. Znacznie gorszy wynik ROLAP wynika z bardzo skomplikowanego zapytania definiującego zmaterializowaną per-

spektywę. Zapytanie to łączy 4 tabele oraz wykonuje grupowanie wyników trzech operatorów ROLLUP z 2-3 argumentami.

Trzeci test obejmował znalezienie wartości na wszystkich płaszczyznach i krawędziach kostki danych implementującej agregat. Test ten wykonano dla dwóch przypadków, w pierwszym przypadku zapytanie wykonano przy pustym buforze danych, w drugim przypadku bufor był wypełniony danymi przetwarzanymi przez poprzednie zapytanie. Przy pustym buforze ROLAP wykonywał zapytanie 4,5 razy w porównania do MOLAP. Gorszy wynik MOLAP jest spowodowany koniecznością rozkompresowania złożonych struktur danych. Natomiast w przypadku wypełnionego bufora czas odpowiedzi był porównywalny i wynosił 2 sekundy dla MOLAP i 1 sekundę dla ROLAP. Są to wyniki wystarczające dla wydajnej obsługi zapytań ad-hoc generowanych przez aplikacji interakcyjne.

Ostatnim kryterium porównania MOLAP i ROLAP była objętość przechowywanych danych. W tym przypadku, dane zgromadzone w MOLAP zajmują ponad 5 razy mniej miejsca niż te same dane przechowywane w ROLAP. Różnica ta wynika, ze sposobu reprezentowania wymiarów danych. W MOLAP służą do tego dedykowane struktury danych, umożliwiające jednokrotne składowanie tej samej wartości wymiaru. Natomiast w ROLAP wymiary są składowane klasycznie w kolumnach tabeli co powoduje, że pojedyncza wartość wymiaru jest powielana wielokrotnie, w zależności od liczny wystąpień faktu, który jest opisywany przez daną wartość wymiaru.

4. Podsumowanie

W pracy porównano wydajność hurtowni danych ROLAP i MOLAP w Oracle 10g wykorzystując dwa kryteria oceny testów wydajnościowych: czas ładowania danych do hurtowni oraz czas odpowiedzi na wykonanie przykładowego zapytania analitycznego.

Dla pierwszego kryterium całkowity czas ładowania danych (czas ładowania danych + czas materializacji agregatów) jest ponad 3 razy krótszy dla MOLAP w porównaniu do ROLAP. Natomiast dla drugiego kryterium ROLAP charakteryzuje się co najmniej 2 krotnie krótszym czasem odpowiedzi. Należy jednak podkreślić, że czas odpowiedzi zarówno dla ROLAP jak i dla MOLAP jest wystarczająco krótki (1-2 sekund) aby zapewnić komfort pracy użytkownikom z wykorzystaniem aplikacji interakcyjnych.

Podsumowując uzyskane wyniki można stwierdzić, że MOLAP może stanowić atrakcyjną alternatywę dla ROLAP, w szczególności dla wielkich hurtowni danych, gdzie czas załadownia danych jest krytycznym wymaganiem. Przykładowo: nowe dane zebrane z systemów OLTP muszą być załadowane w godzinach nocnych, tak aby były one dostępne następnego dnia w hurtowni danych.

Bibliografia

[BWZ04] Bębel B., Wrembel R., Zadrożna A.: Implementowanie hurtowni danych - zagadnienia technologiczne. Materiały konferencyjne Hurtownie Danych i Business Intelligence, Warszawa, marzec 2004

[JeWr02] Jezierski J., Wrembel R.: Oracle9i Lite: rozwiązanie dla mobilnych baz danych? Materiały konferencyjne PLOUG2002, Zakopane, październik, 2002, ISSN 1641-2117

[OMG] Object Management Group. Common Warehouse Metamodel Specification, v1.1. http://www.omg.org/cgi-bin/doc?formal/03-03-02

[TPCH] Transaction Processing Performance Council; http://www.tpc.org/

[VVS00] Vetterli T., Vaduva A., Staudt M.: Metadata Standards for Data Warehousing: Open Information Model vs. Common Warehouse Metadata. SIGMOD Record, vol. 29, No. 3, Sept. 2000