

XI Konferencja PLOUG
Kościelisko
Październik 2005

Oracle10g, Sybase IQ, SQL Server 2000 – ocena funkcjonalności serwerów baz danych w zastosowaniach hurtowni danych

Jędrzej Błasiński, Robert Wrembel

Politechnika Poznańska, Instytut Informatyki

Robert.Wrembel@cs.put.poznan.pl

Streszczenie

Na rynku serwerów baz danych dla zastosowań hurtowni danych dostępnych jest kilka komercyjnych produktów, tj. Oracle9i/10g, SQL Server, SAS BI server, NCR Teradata, Sybase IQ, Hyperion, SAP Business Warehouse. Najpopularniejszymi z nich, stosowanymi w Polsce są dwa pierwsze. Producenci pozostałych systemów również są obecni na polskim rynku i coraz intensywniej promują swoje produkty. Celem niniejszego artykułu jest omówienie podstawowej funkcjonalności serwerów Oracle9i/10g, SQL Server, Sybase IQ i ich porównanie w kontekście hurtowni danych. Przedstawiane wyniki są rezultatem prac implementacyjno-testowych zrealizowanych w Instytucie Informatyki Politechniki Poznańskiej z wykorzystaniem ww. systemów.

Informacja o autorach

Jędrzej Błasiński jest absolwentem Politechniki Poznańskiej, kierunku informatyka.

Robert Wrembel: por. referat "Porównanie wydajności hurtowni danych ROLAP i MOLAP w Oracle 10g".

1. Hurtownie danych – zarys technologii

Z punktu widzenia technicznego, hurtownia danych jest bazą danych. Posiada ona jednak charakterystykę odmienną od typowych baz danych dla zastosowań ewidencyjnych. Podstawowe różnice dotyczą m.in. rozmiaru przechowywanych danych, liczby użytkowników i typu operacji przez nich wykonywanych. Ponieważ hurtownia integruje dane pochodzące z innych systemów, a dodatkowo zawiera dane historyczne i zagregowane, więc jej rozmiar jest znacznie większy od rozmiarów typowych baz danych. Przykładowo, największe polskie hurtownie mają rozmiary rzędu kilkudziesięciu TB.

Na hurtowni danych pracują aplikacje analityczne - OLAP (ang. On-line Analytical Processing) uruchamiane przez decydentów, których jest niewiele w porównaniu z użytkownikami tradycyjnych systemów. Większość operacji realizowanych przez tego typu aplikacje obejmuje złożone zapytania wykorzystujące łączenie, filtrowanie, agregowanie, wymagające dostępu do milionów rekordów. Przykładami takich zapytań mogą być: *Jaki jest trend sprzedaży towarów z branży elektroniki użytkowa w ostatnich kilku tygodniach?* *Jaki jest rozkład sprzedaży odtwarzaczy DVD w województwie wielkopolskim?* Tak więc, przetwarzanie w aplikacjach analitycznych charakteryzuje się operacjami odczytu dużych wolumenów danych, przetwarzanych następnie przez złożone funkcje analityczne. Kluczowym parametrem efektywnościowym dla tego typu systemów jest czas odpowiedzi na zapytania użytkowników.

Na efektywność przetwarzania zapytań analitycznych w hurtowni ma wpływ m.in.:

- zastosowany model danych, czyli reprezentacji danych w systemie i związana z nim struktura samej hurtowni, czyli tzw. jej schemat;
- sposób zasilania/odświeżania hurtowni danymi pochodzącymi ze źródeł zewnętrznych;
- mechanizmy optymalizacji zapytań analitycznych (zaawansowane optymalizatory kosztowe, perspektywy zmaterializowane, techniki przepisowywania zapytań z wykorzystaniem tych perspektyw i indeksowanie danych, przetwarzanie równoległe);
- efektywne składowanie dużych wolumenów danych (partycjonowanie i kompresja);
- wsparcie języka SQL dla przetwarzania analitycznego.

Na światowym rynku serwerów baz danych dla zastosowań hurtowni danych dostępnych jest kilka komercyjnych produktów, tj. Oracle9i/10g, SQL Server, SAS BI server, NCR Teradata, Sybase IQ, Hyperion, SAP Business Warehouse. Najpopularniejszymi z nich stosowanymi w Polsce są dwa pierwsze, jednak producenci pozostałych systemów również są obecni na polskim rynku i coraz intensywniej promują swoje produkty. Wszystkie wspomniane systemy oferują wspomnianą wyżej funkcjonalność na mniej lub bardziej zaawansowanym poziomie.

Celem niniejszego artykułu jest omówienie podstawowej funkcjonalności serwerów *Oracle9i/10g*, *SQL Server*, *Sybase IQ* i ich porównanie w kontekście hurtowni danych. Przedstawiane wyniki są rezultatem prac implementacyjno-testowych zrealizowanych w Instytucie Informatyki Politechniki Poznańskiej z wykorzystaniem ww. systemów.

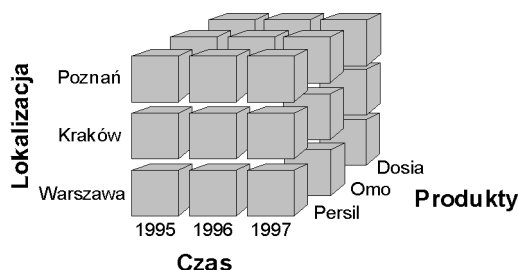
2. Model danych i schemat hurtowni

W praktyce wykorzystuje się trzy następujące modele reprezentacji i przechowywania danych w hurtowni: *relacyjny*, zwany również ROLAP (ang. Relational OLAP), *wielowymiarowy*, zwany również MOLAP lub MDOLAP (ang. Multidimensional OLAP) i *hybrydowy* - HOLAP (ang. hybrid OLAP).

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).

Schemat gwiazdy posiada następujące cechy: (1) istnieje jedna tabela faktów, (2) tabele wymiarów są zdenormalizowane, tj. nie spełniają wymogów trzeciej postaci normalnej. Jeśli wymiary są znormalizowane, wówczas schemat hurtowni danych ma postać **płatka śniegu**. Schemat gwiazdy lub płatka śniegu, w którym ta sama tabela wymiaru jest powiązana z kilkoma tabelami faktów nazywa się schematem **konstelacji faktów**.

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 wielowymiarowa tablica została przedstawiona na rysunku 1. Zawiera ona trzy wymiary: *Lokalizacja*, *Czas* i *Produkty*, a poszczególne komórki kostki, tzw. miara (ang. measure), przechowują informację np. o sprzedaży wybranych produktów w poszczególnych latach, w wybranych miastach.



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

Połączeniem ROLAP i MOLAP jest model **HOLAP**. W modelu tym dane elementarne są składowane w standardowych tabelach, natomiast dane zagregowane są przechowywane w kostkach.

2.1. Oracle

Baza danych *Oracle* wspiera implementowanie dowolnych schematów hurtowni danych. Można do tego celu wykorzystać oprogramowanie *Designer6i/9i/10g*, które posiada opcję projektowania hurtowni danych. Dla zaprojektowanego schematu zostaną wygenerowane zarówno skrypty tworzące tabele jak i skrypty tworzące indeksy właściwe dla zaprojektowanego schematu.

W przypadku implementacji relacyjnej zalecane jest zdefiniowanie dodatkowych logicznych obiektów bazy danych opisujących ww. schematy. Obiektami tymi są **wymiar** (ang. dimension), **hierarchia wymiaru** (ang. dimension hierarchy) i **zależności funkcyjne** (ang. functional dependencies). Obiekty takie są wspierane przez bazę danych *Oracle*. Tworzy się je albo za pomocą graficznej aplikacji *Enterprise Manager* albo za pomocą poleceń SQL. Wspomniane obiekty są wykorzystywane w procesie optymalizacji zapytań analitycznych.

W systemie *Oracle* dane wielowymiarowe są przechowywane w tzw. **przestrzeni analitycznej** (ang. analytic workspace). Definiowanie tej przestrzeni i zarządzanie nią realizuje się albo z wykorzystaniem oprogramowania *Analytic Workspace Manager*, albo *Warehouse Builder*, albo z poziomu SQL wykorzystując do tego celu pakiety systemowe *CWM2*, *DBMS_AWM*, *DBMS_AW*, *DBMS_AW_UTILITIES*, *OLAP_TABLE*. Zbiór pakietów rodziny *CWM2* umożliwia zarządzanie informacjami słownikowymi (metadanymi) opisującymi przestrzeń analityczną. *DBMS_AWM* zawiera procedury tworzenia przestrzeni analitycznej, natomiast *DBMS_AW* zawiera procedury umożliwiające operowanie na danych wielowymiarowych, m.in. wczytywanie danych z tabel, analizę danych, wymianę danych pomiędzy przestrzenią analityczną, a tabelami. *DBMS_AW_UTILITIES* udostępnia procedury zarządzania miarami w przestrzeni analitycznej.

OLAP_TABLE umożliwia prezentowanie danych wielowymiarowych w postaci tabelarycznej i stanowi interface do modelu relacyjnego.

2.2. SQL Server2000

W ramach oprogramowania *SQL Server2000* dostępny jest serwer nazwie *Analysis Services*, który oferuje trzy modele przechowywania danych tj. MOLAP, ROLAP i HOLAP.

W modelu MOLAP dane są pobierane do *Analysis Services*. Klucze wymiarów w tabeli faktów są skompresowane, ponadto zakładane są na nich indeksy. Główną zaletą modelu jest szybkość pobierania wyników znajdujących się od razu w systemie. Istotną wadą jest natomiast brak połączenia wielowymiarowej tablicy z magazynem danych, co znacznie komplikuje proces aktualizacji danych. Ponadto, model ten przechowuje tylko podstawowe typy danych takie jak tekst czy liczby.

Model ROLAP wykorzystuje dane pogrupowane w tabelach zbiorczych umieszczonych w hurtowni danych. Zawierają one różne poziomy agregacji w zależności od opcji wybranych podczas tworzenia kostek danych. *Analysis Services* buduje tabele zbiorcze z kolumnami dla każdego wymiaru i każdej miary oraz indeksuje każdą kolumnę określającą wymiar. Model ten eliminuje problemy występujące w przypadku modelu MOLAP kosztem spadku efektywności zapytań.

W modelu HOLAP, łączącym zalety modeli MOLAP i ROLAP, centralna baza danych pozostaje magazynem danych, zagregowane dane są natomiast przechowywane na serwerze *Analysis Services*.

Server *Analysis Services* umożliwia tworzenie wymiarów czterech następujących typów:

- regularne (ang. regular),
- wirtualne (ang. virtual),
- nadrzędny-podrzędny (ang. parent-child),
- eksploracyjne (ang. data mining).

Najprostszym typem wymiaru jest wymiar **regularny**. Jego zagregowane dane znajdują się w kostkach z nim związanych. Wymiar ten posiada liczbę poziomów zgodną z liczbą kolumn wybranych podczas jego definiowania. Tworzony jest w oparciu o wiele tabel modelu relacyjnego.

Wymiary **wirtualne** są oparte o dane pochodzące z już utworzonych wymiarów. Ich dodanie do kostki danych nie zwiększa jej rozmiarów ponieważ nie przechowuje on żadnych agregatów – są one wyliczane w pamięci. Zapytania operujące na kostkach zawierających wirtualne wymiary są jednak mniej efektywne.

Wymiar **nadrzędny-podrzędny** umożliwia reprezentację hierarchii, właściwych dla operacji self-join. Charakterystycznym przykładem zależności tego typu są np. zależności hierarchiczne jednostek organizacyjnych, związki komponent-element składowy, modelowane za pomocą jednej tabeli i powiązania tej tabeli samej z sobą.

Specyficznymi strukturami są wymiary **eksploracyjne**. Mogą być one tworzone jedynie w oparciu o modele eksploracji danych dostępne w *Analysis Services* i umożliwiają analizę wyników otrzymanych po zastosowaniu wybranych algorytmów eksploracyjnych.

2.3. Sybase IQ

Ze względu na specyficzną technikę składowania danych w technologii ROLAP stosowaną w tym serwerze, model wielowymiarowy nie jest stosowany, ponieważ, jak podaje producent, zastosowana technika składowania jest znacznie szybsza od tradycyjnych.

3. Zasilanie/odświeżanie hurtowni

Bezpośrednio po zaimplementowaniu i uruchomieniu systemu hurtowni danych jest ona pusta. Należy ją wypełnić danymi pochodzącymi z różnych źródeł. Proces ten, w literaturze znany jako ETL (Extraction Translation Loading) składa się z trzech następujących faz: odczytu danych ze źródeł (Extraction), transformacji ich do wspólnego modelu wykorzystywanego w hurtowni wraz z usunięciem wszelkich niespójności (Translation), wczytanie danych do hurtowni (Loading).

Głównym problemem w dostępie do źródeł zewnętrznych z hurtowni jest heterogeniczność tych źródeł. Nawet jeśli źródła są bazami danych, to bardzo często są to bazy danych pochodzące od różnych producentów, a co za tym idzie, posiadające różną funkcjonalność, reprezentację danych i dialekt języka SQL. Z tego względu, dostęp z jednej bazy danych do innej musi być realizowany za pomocą dedykowanego oprogramowania dla tych baz. Oprogramowanie to nosi nazwę **gateway'a**. Jego zadaniem jest m.in. transformowanie dialektów SQL i reprezentacji (typów) danych przesyłanych między bazami danych. Kolejnym rozwiązaniem jest zastosowanie sterowników ODBC/OLE DB/JDBC. Wszystkie serwery baz danych istniejące na rynku, zarówno komercyjne jak i niekomercyjne, umożliwiają dostęp do swoich danych za pomocą tych sterowników.

Proces zasilania hurtowni danymi początkowymi jest czasochłonny ze względu na liczbę danych, która musi zostać do niej wczytana. Pierwsze zasilanie jest realizowane jako tzw. **odświeżenie pełne** (ang. complete refreshing). W tym procesie wszystkie dane ze źródeł są wczytywane do hurtowni. W trakcie eksploatacji hurtowni, jest ona odświeżana cyklicznie. Podczas tego procesu dąży się do tego, aby do hurtowni trafiały jedynie te dane ze źródeł, które uległy zmianie od czasu poprzedniego odświeżania. Jest to tzw. **odświeżanie przyrostowe** (ang. incremental refreshing).

3.1. Oracle

Z instalacją *Oracle9i/10g* jest dostarczane oprogramowanie *Transparent Gateways*, które umożliwia dostęp i wymianę danych z większością głównych komercyjnych systemów relacyjnych, tj. wytwarzanych przez *IBM, Sybase, Microsoft, NCR*. Natomiast definiowanie całego procesu ETL jest wspomagane oprogramowaniem *Warehouse Builder*.

Dane do hurtowni wczytuje się często również z plików tekstowych. W tym zakresie rozwiązania *Oracle9i/10g* obejmują m.in.: pakiet systemowy *UTL_FILE*, oprogramowanie *SQL*Loader*, tabele zewnętrzne i funkcje tablicowe. Pakiet *UTL_FILE* zawiera procedury i funkcje odczytu i zapisu plików. *SQL*Loader* umożliwia wczytywanie informacji z plików tekstowych o różnym formacie. Możliwe jest także weryfikowanie, transformowanie i filtrowanie danych przed ich wczytaniem do bazy. Rozszerzeniem tej funkcjonalności jest możliwość definiowania w bazie danych tzw. *tabel zewnętrznych* (ang. external tables), których źródłem danych są pliki tekstowe systemu operacyjnego. Do tabel takich odwołuje się za pomocą standardowych poleceń SQL *select*, a dane są dynamicznie pobierane z plików skojarzonych z tabelą zewnętrzną. W *Oracle10g* za pomocą tabel zewnętrznych można dodatkowo kopiować dane z bazy danych do plików zewnętrznych. Mechanizm tzw. *funkcji tablicowych* (ang. table functions) udostępnia dane, które najpierw są przetwarzane, a następnie przekazywane przez funkcje napisane w języku PL/SQL. Funkcje te są składowane w bazie danych. Źródłami danych dla tych funkcji mogą być dowolne obiekty bazy danych (np. zwykłe tabele lub tabele zewnętrzne) lub pliki tekstowe. Dane mogą być przetwarzane w bardzo złożony sposób, a algorytmy ich przetwarzania implementuje się w języku proceduralnym. Funkcje tablicowe w zapytaniach wykorzystuje się w sposób identyczny jak tabele.

Jeżeli źródła danych i hurtownia są oparte o bazy Oracle, wówczas do przenoszenia danych ze źródeł do hurtowni można wykorzystać mechanizm eksportu/importu danych. *Eksport* umożliwia zapisanie do binarnego pliku albo danych pojedynczej tabeli, albo całego schematu użytkownika, albo zawartości całej bazy danych, w zależności od wybranej opcji. *Import* umożliwia wczytanie

zawartości pliku eksportu do wskazanej bazy danych. W *Oracle10g* udostępniono mechanizm o nazwie *Data pump*, stanowiący rozszerzenie funkcjonalności standardowego eksportu/importu. W przypadku korzystania z *Data pump* jest możliwy m.in. podział eksportowanych danych na wiele plików, szeregowanie zadań eksportu/importu, ich automatyczne uruchamianie, czasowe wstrzymywanie aktywnych zadań i ich ponowne uruchamianie. Poza tym, mechanizm *Data pump* został zaimplementowany i jest uruchamiany na serwerze, jako pakiet *DBMS_DATAPUMP*, co znacznie przyspiesza jego działanie, w porównaniu ze standardowym mechanizmem export/import (uruchamianym po stronie klienta).

Baza danych *Oracle* posiada dwa mechanizmy umożliwiające propagowanie zmian z bazy produkcyjnej do hurtowni. Pierwszy z nich to *wyzwalacze* (ang. triggers), czyli procedury wyzwalane na skutek operacji modyfikowania danych źródłowych. Procedury te mogą informować moduł monitora lub propagować dane do hurtowni. Projektant hurtowni musi w tym przypadku sam zaimplementować właściwie działające wyzwalacze. Drugi mechanizm wykorzystuje tzw. *migawki* (ang. snapshots) zwane również perspektywami zmaterializowanymi (ang. materialized views).

Perspektywa zmaterializowana jest kopią tabeli lub jej fragmentu pochodzącego z innej bazy danych. W definicji takiej perspektywy określa się najczęściej: częstotliwość jej odświeżania, sposób odświeżania (przyrostowe, pełne), zbiór danych ze źródła który ma być dostępny w perspektywie. System zarządzania hurtownią danych sam kontroluje proces odświeżania perspektywy. Odświeżanie przyrostowe perspektyw zmaterializowanych jest możliwe dopiero po utworzeniu w bazie produkcyjnej tzw. *dziennika perspektywy zmaterializowanej* (ang. materialized view log) dla każdej tabeli, z której perspektywa pobiera dane. Zadaniem dziennika jest rejestrowanie zmian zachodzących w danych źródłowych. Proces odświeżający perspektywy korzysta z informacji zapisanych właśnie w dziennikach. Odświeżanie przyrostowe z wykorzystaniem perspektyw zmaterializowanych i ich dzienników jest jednak możliwe jedynie w przypadku, gdy zarówno bazy źródłowe, jak i sama hurtownia są oparte o *Oracle*. Perspektywy zmaterializowane tworzy się albo z wykorzystaniem oprogramowania *Enterprise Manager* albo z poziomu SQL.

3.2. SQL Server

W zakresie wczytywania danych do hurtowni, *SQL Server* oferuje dwa mechanizmy: polecenie *bulk insert* i oprogramowanie Data Transformation Services (DTS). *bulk insert* jest poleceniem języka Transact SQL umożliwiającym import danych z plików tekstowych w formacie znakowym (ASCII), unicode lub zgodnym z aktualnym formatem *SQL Servera* (native). Funkcjonalność *Data Transformation Services* (DTS) umożliwia zrealizowanie dostępu do źródeł danych (*SQL Server*, *Excel*, *Access*), wczytanie danych, ich transformację i integrację. Połączenia są realizowane w oparciu o sterowniki OLE DB.

Oprogramowanie DTS składa się z następujących modułów:

- *Import/Export Wizard* - umożliwia tworzenie prostych pakietów DTS, wspierający migracje danych oraz proste transformacje;
- *DTS Designer* - interfejs graficzny umożliwiający tworzenie rozbudowanych procesów transferu danych;
- *DTSRun* - konsola znakowa umożliwiająca uruchamianie istniejących pakietów DTS;
- *DTSRunUI* - interfejs graficzny programu *DTSRun*.

Podstawę procesu zarządzania migracją danych stanowią **pakiety DTS**. Są to logiczne kolekcje grupujące zadania DTS, transformacje DTS oraz ograniczenia kolejnościowe między tymi obiektami, utworzone za pomocą narzędzi DTS lub programistycznie. Pakiety te są przechowywane w *SQL Server*. Każdy pakiet zawiera jedno lub więcej zadań, które w zależności od nałożonych ograniczeń mogą być wykonywane sekwencyjnie lub współbieżnie. Przykładowe wywołanie pakietu może obejmować sekwencje następujących czynności: połączenie bazy danych z zewnątrz-

nym źródłem, transfer danych i obiektów, ich transformację oraz powiadomienie administratora o pomyślnym zakończeniu procesu.

Zadania DTS stanowią podstawowy element pakietu DTS. Każde zadanie definiuje określony zakres pracy do wykonania jako element procesu migracji danych. Pakiet DTS dostarcza zestawu predefiniowanych zadań dostępnych zarówno za pomocą graficznego interfejsu przez aplikację *DTS Designer* jak i programistycznie. Przykładowymi zadaniami są: import i eksport, transformacja danych, kopiowanie obiektów bazy danych (np. procedur, perspektyw, wyzwalaczy) ale tylko pomiędzy dwoma bazami *SQL Server*, wykonywanie poleceń języka Transact SQL oraz skryptów.

Transformacje DTS są zbiorem funkcji i operacji wykonywanych na zbiorach danych podczas procesu migracji. Transformacje DTS umożliwiają dokonanie walidacji skomplikowanych danych, konwersji danych, konsolidacji i grupowania.

Przeptywy DTS umożliwiają łączenie dwóch zadań w sekwencje, w której uruchomienie drugiego z nich zależy od statusu wykonania pierwszego.

Odświeżanie hurtowni jest możliwe albo za pomocą cyklicznie uruchamianych pakietów DTS, albo za pomocą mechanizmu replikacji. W zakresie replikacji *SQL Server* wykorzystuje mechanizmy publikatora/publikacji, dystrybutora, subskrybenta/subskrypcji.

Dystrybutor (serwer dystrybucyjny) utrzymuje wszystkie zmienione dane, które są rozsyłane do subskrybentów. Może być zlokalizowany na serwerze publikacji lub innym (w takim przypadku staje się zdalnym serwerem dystrybucyjnym, zawierającym bazę dystrybucyjną). Pojedynczy serwer dystrybucyjny może obsługiwać wiele publikatorów i jest koordynatorem całego procesu.

Subskrybent (serwer subskrybenta) przechowuje kopię lub fragment publikowanej bazy danych. Serwer dystrybucyjny przesyła wszystkie zmiany wprowadzone w publikowanej bazie do kopii utrzymywanej na subskrybentach.

Publikacja stanowi grupę złożoną z jednego lub kilku artykułów i stanowi podstawową jednostkę replikowanych danych. **Artykuł** jest wskaźnikiem do pojedynczej tabeli, zbioru jej rekordów lub kolumn podlegających replikowaniu. Pojedyncza baza danych może zawierać więcej niż jedną publikację. Publikować można dane z tabel, wyniki działania procedur składowanych, a nawet obiekty schematu bazy, w tym indeksy i wyzwalacze. Niezależnie od tego, co podlega replikacji, wszystkie artykuły w publikacji są synchronizowane w tym samym czasie.

SQL Server2000 wykorzystuje agentów replikacji do prowadzenia różnych zadań tego procesu. Usługi te są okresowo uaktywniane. Do najważniejszych z nich należą: agent migawki, agent odczytu dziennika transakcji, agent dystrybucji, agent scalający oraz agent usuwania historii.

W *SQL Server2000* można stosować trzy typy replikacji: (1) migawkową, (2) transakcyjną i (3) scalającą. Każdy z tych typów odnosi się do pojedynczej publikacji. Możliwe jest stosownie różnych kombinacji tych metod dla pojedynczej bazy danych.

Replikacja migawkowa tworzy obraz wszystkich tabel publikacji w określonym momencie i rozsyła go do subskrybentów. Tworzona publikacja jest budowana za każdym razem od początku – jest to więc replikacja pełna. Replikacja migawkowa jest najprostszą pod względem konfiguracji.

Replikacja transakcyjna polega na pobieraniu transakcji z dziennika transakcji publikowanej bazy danych i przesyłaniu ich do bazy subskrypcyjnej. Wszystkie modyfikacje danych zostają zapisane w dystrybucyjnej bazie danych i przesłane są do dowolnej liczby subskrybentów. Proces pobierania zmian z dziennika transakcji publikowanej bazy danych jest bardzo efektywny. Nie wymaga on bezpośredniego odczytu tabel, z wyjątkiem wykonania początkowej migawki, a liczba danych przenoszonych przez sieć jest minimalna. Zalety te sprawiają, że replikacja transakcyjna jest najczęściej wykorzystywaną metodą. Odbiorcy mogą być również skonfigurowani do prowa-

dzenia subskrypcji wymuszonej, co jest szczególnie wskazane dla tych z nich, którzy do sieci włączają się okresowo.

W przypadku **replikacji scalającej**, po zainicjowaniu publikatora i wszystkich subskrybentów możliwa jest zmiana danych zarówno przez serwer publikujący, jak i serwery subskrybentów. Wprowadzone modyfikacje są scalane i rozsyłane do wszystkich zainteresowanych subskrypcją stron, co umożliwia utrzymywanie w każdej bazie prawie identycznych danych. Należy rozwiązywać okresowo występujące konflikty. W czasie ich rozstrzygnięcia stroną uprzywilejowaną jest zawsze publikator, jeżeli zasad nie określono w inny sposób.

3.3. Sybase IQ

W zakresie wczytywania danych z plików tekstowych ASCII *Sybase IQ* udostępnia polecenie *load table*. Przykładowo, wczytanie danych z dwóch plików: *zrodlo2.dmp* i *zrodlo3.dmp* umieszczonych na lokalnym dysku C w katalogu *mojedane* do tabeli *nowe_produkty*. Domyślnym separatorem kolumn jest tabulator, natomiast separatorem wierszy jest znak nowej linii.

```
LOAD TABLE nowe_produkty
( id,
nazwa,
opis,
rozmiar,
kolor '\x09' NULL( 'null', 'brak', 'na' ),
liczebosc PREFIX 2,
cena PREFIX 2 )
FROM 'C:\mojedane\zrodlo2.dmp',
'C:\mojedane\zrodlo3.dmp'
QUOTES OFF
ESCAPES OFF
BLOCKSIZE 100000
FORMAT ascii
DELIMITED BY '\x09'
ON FILE ERROR CONTINUE
ROW DELIMITED BY '\n'
```

Polecenie *insert* rozszerzone o klauzule *location* umożliwia wczytanie danych z baz *Adaptive Server Enterprise*, *Adaptive Server Anywhere*, *Microsoft SQL Server*, *IBM DB2*, a także wcześniejszych wersji *Sybase IQ*. Przykładowo, poniższe polecenie wczytuje dane z innej bazy danych *Sybase IQ*:

```
INSERT INTO pracownicy_poznan
(prac_id, placa, stanowisko)
LOCATION 'poznan.asiqdb'
{ SELECT prac_id, placa, stanowisko
FROM pracownicy
WHERE staz_pracy > 1 }
```

Bardzo wydajnym rozwiązaniem wspierającym migrację danych jest *Sybase Enterprise Connect Data Access* – za jego pośrednictwem istnieje możliwość transferu danych z baz danych *AS/400*, *DB2*, *Oracle*, *Informix*, *SQL Server* oraz innych baz dostępnych przez ODBC. Innymi popularnymi aplikacjami są również, oferowane przez zewnętrznych producentów, *Ascential Data Stage* i *Informatica Metadata Exchange*. Natomiast realizacja procesu migracji danych pomiędzy bazą *SQL Server* a magazynem danych *Sybase IQ* może być w całości oparta na wykorzystaniu pakietów DTS.

Aplikacją umożliwiającą proces replikacji w *Sybase IQ* jest, oferowany przez firmę Sybase jako osobny produkt, *Sybase Replication Server*. Mechanizm replikacji oparto tutaj na przesyłaniu poleceń modyfikujących dane źródłowe. Proces ten nie jest realizowany w oparciu o wyzwalacze, ale poprzez dedykowane moduły, całkowicie odseparowane od operacji wykonywanych przez źródło danych.

4. Indeksowanie danych

Podstawowymi strukturami danych wykorzystywanymi do optymalizacji zapytań są indeksy. Dla bardzo złożonych zapytań typu OLAP, operujących na ogromnej liczbie danych, standardowe indeksy w postaci B–drzew okazują się nieefektywne ponieważ po pierwsze, nie zapewniają wystarczająco szybkiego dostępu do danych, po drugie, ich rozmiary są zbyt duże, przez co wzrastają koszty ich przetwarzania, przechowywania i utrzymywania. Z tego względu w zastosowaniach hurtowni danych najczęściej stosuje się indeksy bitmapowe i połączeniowe.

4.1. Indeks bitmapowy

Ideą indeksu bitmapowego (ang. bitmap index) jest wykorzystanie pojedynczych bitów do zapamiętania informacji o tym, że dana wartość atrybutu występuje w określonym rekordzie tabeli. Dla każdej unikalnej wartości indeksowanego atrybutu jest przechowywana tablica bitów, zwana *mapą bitową*. Kolejne bity tej mapy odpowiadają kolejnym rekordom indeksowanej tabeli, a liczba bitów mapy bitowej odpowiada liczbie rekordów tej tabeli. Indeks bitmapowy jest zbiorem map bitowych dla wszystkich unikalnych wartości danego atrybutu. Dodatkowo, odwzorowanie rekordów indeksowanej tabeli w poszczególne bity map jest realizowane poprzez przechowywanie w indeksie bitmapowym fizycznych adresów rekordów (ROWID) indeksowanej tabeli. Przykładowy indeks bitmapowy dla atrybutu *kolor* tabeli *Sprzedaż* przedstawiono w tabeli 2.

Sprzedaż				
ROWID	sklep_id	nazwa	...	kolor
x00A1	1010	Omo	...	tak
x00A2	1010	Omo	...	nie
...
x00B6	1010	Omo	...	nie
x00B7	1080	Persil	...	tak
x00B8	1080	Persil	...	nie
x00B9	1030	Dosia	...	tak

Tabela 1. Przykładowa tabela *Sprzedaż*

kolor		
ROWID	tak	nie
x00A1	1	0
x00A2	0	1
...
x00B6	0	1
x00B7	1	0
x00B8	0	1
x00B9	1	0

Tabela 2. Indeks bitmapowy dla atrybutu *kolor*

Indeksy bitmapowe są bardziej efektywne od indeksów w postaci B–drzewa dla atrybutów, które posiadają wąską dziedzinę (kilka do kilkudziesięciu różnych wartości), dla tabel o dużej liczbie rekordów, np. miliony. Indeksy te sprawdzają się dla określonej klasy zapytań. Są to zapytania wykorzystujące dużą liczbę warunków selekcji z operatorami równości oraz zapytania wykorzystujące funkcję *count*. Większa efektywność tych indeksów wynika z: (1) dużej szybkości przetwarzania map bitowych za pomocą operatorów *and*, *or* i *not*, (2) małego rozmiaru indeksów bitmapowych dla atrybutów o wąskiej dziedzinie, (3) możliwości wykonywania operacji logicznych i funkcji *count* bezpośrednio na indeksach bitmapowych (znajdujących się w pamięci operacyjnej), a nie na samych rekordach.

4.2. Indeks połączeniowy

Indeks połączeniowy (ang. join index) łączy z sobą rekordy z różnych tabel posiadające tę samą wartość atrybutu połączeniowego, jest więc strukturą zawierającą zmaterializowane połączenie wielu tabel. Indeks taki posiada strukturę B–drzewa zbudowanego na atrybucie połączeniowym

tabeli (bądź na wielu takich atrybutach). Liście indeksu zawierają wspólne wartości atrybutu połączeniowego tabel wraz z listami adresów rekordów w każdej z łączonych tabel.

Odmianą tego indeksu jest tzw. *indeks bitmapowy* (ang. *bitmap index*). Różni się on od indeksu połączeniowego tym, że w jego liściach zamiast wskaźników do rekordów znajdują się mapy bitowe wskazujące na odpowiednie rekordy łączonych tabel.

4.3. Oracle

Oracle9i/10g umożliwia definiowanie indeksów bitmapowych i bitmapowych połączeniowych (oprócz standardowych indeksów B-drzewo). Ponadto, bitmapowe indeksy połączeniowe są wykorzystywane do optymalizacji tzw. zapytań gwiazdzystych (ang. *star queries*). W tym przypadku optymalizator zapytań przepisuje oryginalne zapytanie użytkownika na zapytanie równoważne z wykorzystaniem właśnie tych indeksów. Skraca się przez to znacznie czas realizacji zapytań gwiazdzystych.

4.4. SQL Server

SQL Server2000 umożliwia tworzenie tradycyjnych indeksów typu B-drzewo i tabel o organizacji indeksowej zwanych indeksami klastrowymi (ang. *clustered indexes*). Dane w tabelach o organizacji indeksowej są przechowywane w strukturze B-drzewo, co zwiększa efektywność zapytań wyszukujących dane według klucza podstawowego.

4.5. Sybase

Każdej kolumnie w tabeli zostaje automatycznie założony domyślny indeks typu *FP* (*Fast Projection*). Przeznaczony jest on do obsługi projekcji, połączeń ad-hoc, obliczeń. Indeks ten wewnętrznie występuje w trzech odmianach: *1 byte-FP*, *2 byte FP*, *flat-FP*. Serwer domyślnie wybiera wersję *flat-FP*. Wybór typu indeksu zależy od liczby unikalnych wartości (poniżej 256, pomiędzy 256 a 65536, powyżej 65536) i można go wskazać za pomocą klauzuli *iq unique* polecenia *create table*. Jeśli w trakcie ładowania danych okaże się, że w danej kolumnie przechowywane jest więcej unikatowych wartości niż zadeklarowano, serwer automatycznie przekształci indeks do właściwego typu.

O ile do projekcji najlepszy jest indeks domyślny (*FP*), przy wyszukiwaniu lepiej sprawdzają się indeksy *LF* (*Low Fast*) – indeks bitmapowy, *HG* (*High Group*) – połączenie mapy bitowej i B-drzewo lub *HNG* (*High Non Group*) – indeks *Bit-Wise*.

Indeks typu *High Group* jest automatycznie zakładany na kolumnach, które stanowią klucz podstawowy lub obcy. Indeks *HG* jest bardzo efektywny w przypadku operacji połączenia i grupowania. Posiada on jednak najbardziej złożoną strukturę, osiąga duże rozmiary, a jego tworzenie jest czasochłonne.

Indeksy typu *High Non Group* stosuje się w odniesieniu do pól o dużym zakresie dopuszczalnych wartości, takich jak miary, wykorzystywane do obliczeń i wyszukiwania w ramach zakresów (np. suma wartości i liczby sztuk towarów w cenie poniżej 500 pln).

Istnieje możliwość zakładania kilku indeksów na tej samej kolumnie. *Sybase IQ* w pełni wspiera wykorzystanie indeksu połączeniowego.

5. Perspektywy zmaterializowane i przepisywanie zapytań

Oprócz replikowania danych, inną bardzo ważną i często stosowaną dziedziną zastosowań perspektyw zmaterializowanych jest optymalizacja zapytań analitycznych. Dla tych zastosowań perspektywy zmaterializowane służą do przechowywania wyliczonych danych (najczęściej zagregowanych), których wyznaczenie jest czasochłonne. Materializowanie danych ma w tym przypadku

sens jeżeli w systemie często pojawiają się zapytania identyczne lub podobne do tego, którego wynik zmaterializowano. Jeżeli w systemie pojawi się zapytanie, które może zostać wykonane z wykorzystaniem zmaterializowanych perspektyw, zamiast korzystania ze źródłowych tabel i wyznaczania wyników od początku, wówczas kosztowy optymalizator zapytań skonstruuje odpowiednie zapytanie do tych perspektyw. Jest to tzw. *przepisanie zapytania* (ang. query rewriting). Proces ten jest całkowicie niewidoczny dla użytkownika.

5.1. Oracle

Praktycznie, w każdej hurtowni danych oprócz tabel zawierających dane elementarne znajdują się perspektywy zmaterializowane, przechowujące dane zbiorcze/zagregowane na różnych poziomach. Projektując hurtownię danych, należy zapewnić, aby zbiór tych perspektyw był optymalny, tzn. każda z nich była jak najczęściej wykorzystywana w procesie przepisywania zapytań. Wyznaczenie takiego zbioru perspektyw jest trudne. W tym zakresie baza danych *Oracle10g* oferuje narzędzie o nazwie *Access Advisor* dostępne w postaci pakietu *DBMS_ADVISOR*. Wykorzystanie jego jest możliwe bezpośrednio z poziomu środowiska *SQL*Plus* lub z poziomu oprogramowania graficznego *Enterprise Manager*. *Access Advisor* umożliwia m.in. określenie: (1) właściwego zbioru zmaterializowanych perspektyw przyspieszających wykonywanie wskazanego zbioru zapytań, (2) właściwego zbioru indeksów typu B–drzewo i bitmapowych przyspieszających wykonanie zapytań ze wskazanego zbioru.

5.2. SQL Server

Mechanizmem umożliwiającym optymalizatorowi zapytań przepisanie zapytania w oparciu o dane zmaterializowane są tzw. perspektywy indeksowane. Powstają one w momencie zdefiniowania dla perspektywy indeksu klastrowego typu *unique*. Indeks taki przechowuje poziomie dane udostępniane przez perspektywy. W momencie zaindeksowania, perspektywa przestaje więc być wirtualną tabelą. Optymalizator potrafi automatycznie odwołać się do danych przechowywanych w perspektywie indeksowanej, proces ten jest w pełni transparentny dla użytkownika. Perspektywy indeksowane posiadają silne ograniczenia co do swojej struktury. W celu sprawdzenia czy na danej perspektywie można założyć indeks, należy wykorzystać funkcję systemową *OBJECTPROPERTY* z parametrem *IsIndexable*.

5.3. Sybase IQ

Brak informacji.

6. Przetwarzanie równoległe

Przetwarzanie równoległe (ang. parallel processing) polega na rozbiciu złożonych operacji na mniejsze, które następnie są wykonywane równoległe, np. na wielu procesorach lub komputerach. W efekcie, czas wykonania całej operacji jest krótszy. W przypadku hurtowni danych, najczęściej równoległe przetwarza się zapytania, sortuje dane, wykonuje operacje odczytu i zapisu na dysk, buduje tablice i indeksy oraz wczytuje dane do hurtowni.

6.1. Oracle

Oracle wspiera przetwarzanie równoległe w zakresie wykonywania zapytań, budowania indeksów, wczytywania danych do hurtowni. Użytkownik jawnie może wskazać liczbę procesów dedykowanych do wykonania danego zadania. Przykładowo, poniższe polecenie (w systemie *Oracle*) odczytuje zawartość tabeli *Sprzedaz* z wykorzystaniem 5 równoczesnych procesów.

```
SELECT /*+ PARALLEL(sp, 5) */ sklep_id, sum(ilosc)
```

```
FROM sprzedaż sp
GROUP BY sklep_id;
```

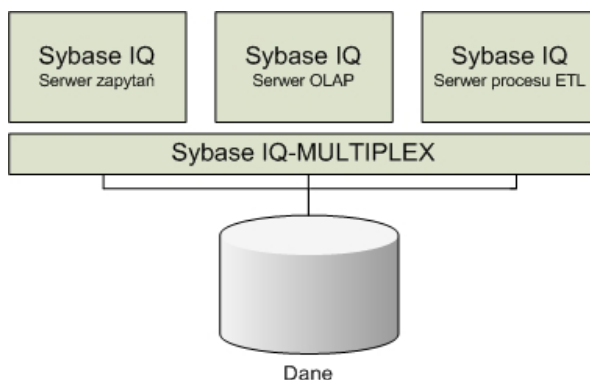
6.2. SQL Server

Procesor zapytań *SQL Server2000* pozwala na równoległe przetwarzanie zapytań. Określana jest optymalna liczba wątków oraz liczba zaangażowanych procesorów. Następnie zadanie jest rozdzielane między wątki. Zapytanie równoległe używa tej samej liczby wątków do czasu zakończenia działania. *SQL Server2000* kontroluje optymalną liczbę procesorów przy każdym pobraniu planu wykonania z bufora.

Przy podejmowaniu decyzji o przetwarzaniu równoległym brane są pod uwagę następujące kryteria: liczba dostępnych procesorów, liczba jednocześnie aktywnych użytkowników, dostępność odpowiedniego zasobu pamięci, typ wykonywanego zapytania, liczba wierszy w aktualnym potoku danych. Konfigurowanie przetwarzania równoległego realizuje się za pomocą parametrów konfiguracyjnych serwera: *Maximum Degree of Parallelism* i *Cost Threshold for Parallelism*. Pierwszy z nich ogranicza liczbę wątków możliwych do wykorzystania w równoległym planie wykonania. Zakres możliwych wartości wynosi od 0 do 32. Wartość ta jest automatycznie ustawiana na 0, co daje możliwość wykorzystania wszystkich dostępnych procesorów. Drugi parametr ustawia górny pułap kosztów, na którego podstawie optymalizator podejmuje decyzję o wykorzystaniu przetwarzania równoległego. Jeżeli wartość kosztu obliczona dla zapytania jest większa niż podana w opcji, generowany jest plan równoległy. Wartość tej opcji jest zdefiniowana jako szacowany w sekundach czas wykonania zapytania na pojedynczym procesorze. Zakres jej wartości wynosi od 0 do 32767.

6.3. Sybase

Zaproponowane przez firmę Sybase rozwiązanie *Adaptive Server IQ-Multiplex*, korzysta z pojedynczych serwerów *Sybase IQ* wykonujących określone funkcje oraz uzyskujących dostęp do wspólnej, dyskowej pamięci masowej. Możliwe jest skonfigurowanie jednego serwera wyłącznie do obsługi zapytań, drugiego do eksploracji danych, a trzeciego do ładowania i aktualizacji danych, co umożliwia optymalne dostosowanie poszczególnych maszyn do realizowanych zadań. Proces ten określa się jako multipleksowanie.



Rysunek 2. Koncepcja Sybase IQ-Multiplex

7. Składowanie dużych wolumenów danych

W hurtowni danych największe rozmiary osiągają tabele faktów. Przeszukiwanie dużych tabel jest czasochłonne, nawet z wykorzystaniem indeksów. Techniki zmniejszające rozmiary danych adresowane w zapytaniach to partycjonowanie i kompresja.

7.1. Partycjonowanie

Często decydenci są zainteresowani analizą tylko podzbioru rekordów tabeli, np. ilości sprzedanych produktów z grupy kosmetyki w Wielkopolsce. Dla takiego zapytania podział dużej tabeli *Sprzedaż* na mniejsze np. ze względu na województwa, w których dokonano sprzedaży znacznie skróciłoby czas dostępu do wybranego podzbioru danych.

Fizyczny podział tabeli (lub indeksu) na części jest nazywany *partycjonowaniem* (ang. partitioning). Każda z części nazywa się *partycją* (ang. partition). Często jest ona fizycznie umieszczana na osobnym dysku, znajdującym się w tym samym lub wielu węzłach (komputerach) sieci. W praktyce wyróżnia się partycjonowanie poziome (ang. vertical partitioning) i pionowe (ang. horizontal partitioning).

Partycjonowanie poziome polega na rozmieszczaniu całych rekordów w odpowiednich partycjach. Rozmieszczanie to jest realizowane na podstawie wartości wskazanego atrybutu tabeli (indeksu), tzw. *atrybutu partycjonującego*. Fizycznie podzielona tabela (lub indeks) stanowi logiczną całość z punktu widzenia użytkownika. *Partycjonowanie pionowe* polega na rozmieszczaniu kolumn w odpowiednich partycjach.

Głównym celem partycjonowania indeksu jest zwiększenie stopnia współbieżności transakcji i minimalizacja rywalizacji transakcji, poprzez rozproszenie operacji wejścia/wyjścia wykonywanych na indeksie.

7.1.1. Oracle

Baza danych *Oracle9i/10g* wspiera trzy następujące techniki partycjonowania rekordów tabeli: partycjonowanie zakresowe, listowe i haszowe. W *partycjonowaniu zakresowym* (ang. range partitioning) dla każdej partycji określa się zakres wartości atrybutu partycjonującego. Do danej partycji trafiają więc rekordy ze ściśle określonego dla tej partycji zakresu określonego wartością atrybutu partycjonującego. W *partycjonowaniu listowym* (ang. list partitioning) dla każdej partycji określa się zbiór wartości atrybutu partycjonującego. Natomiast w *partycjonowaniu haszowym* (ang. hash partitioning) partycja, w której jest umieszczany rekord jest wyznaczana na podstawie dobieranej przez system funkcji haszowej.

Oracle9i/10g umożliwia tworzenie indeksów partycjonowanych zarówno dla tabel partycjonowanych, jak i niepartycjonowanych. Tabela partycjonowana może natomiast posiadać zarówno indeks partycjonowany, jak i niepartycjonowany.

7.1.2. SQL Server

W tym systemie partycjonowanie jest realizowane za pomocą standardowych niezależnych tabel. Tabelę logiczną partycjonowaną symuluje się za pomocą perspektyw. Indeksy nie są partycjonowane więc dla każdej tabeli utrzymywany jest niezależny indeks.

7.1.3. Sybase IQ

Kluczową rolę w rozwiązaniu *Sybase IQ* odgrywa partycjonowanie pionowe, w którym dane są przechowywane są w kolumnach. Użytkownik uzyskuje dostęp wyłącznie do tych kolumn z danymi, które są potrzebne do realizacji danego zapytania. Dzięki temu zmniejszeniu ulega fizyczne obciążenie kanałów wejścia/wyjścia. Przykładowo, jeśli wykonywane zapytanie odwołuje się jedynie do trzech z dziesięciu kolumn tabeli, system *Sybase IQ* przekazuje z dysku do bufora wejścia/wyjścia tylko te trzy kolumny. Rozwiązania tradycyjne wymagają w takiej sytuacji przetworzenia wszystkich kolumn tabeli.

7.2. Kompresja danych

7.2.1. Oracle

Oracle wspiera kompresję danych w blokach i kompresję indeksów. W bloku nieskompresowanym powtarzające się wartości atrybutów są przechowywane wielokrotnie. Natomiast w bloku skompresowanym, powtarzające się wartości są umieszczane na początku bloku, w przeznaczonych do tego celu obszarach, natomiast w rekordach system umieszcza adresy w ramach bloku do odpowiedniego obszaru.

Oprócz kompresji danych w blokach stosuje się kompresję indeksów B–drzewo i bitmapowych. Kompresja indeksu B–drzewo dotyczy jego liści. W liściu nieskompresowanym są przechowywane m.in. pary wartości: wartość indeksowana W – adres rekordu posiadającego wartość W . Jeżeli indeks założono na atrybucie, którego wartość nie jest unikalna, wówczas wartość indeksowana W , pojawia się w liściach wielokrotnie – tyle razy ile jest rekordów z tą wartością. W przypadku liści skompresowanych jest budowana lista zawierająca: wartość indeksowaną W i adresy wszystkich rekordów posiadających wartość W . W ten sposób wartość indeksowana pojawia się w liściu jeden raz.

Indeksy bitmapowe ulegają kompresji wtedy, gdy liczba zer w mapach bitowych staje się zbyt duża w porównaniu do liczby jedynek. Kompresja taka znacznie zmniejsza rozmiar indeksu bitmapowego.

7.2.2. SQL Server

SQL Server2000 zapewnia automatyczną kompresję danych i indeksów. Dodatkowo serwer *Analysis Services* zapewnia zoptymalizowane przechowywanie agregatów w formacie Microsoft OLAP.

7.2.3. Sybase IQ

Sybase IQ wspiera kompresję danych i indeksów (szczegółowe informacje nie są dostępne).

8. Analiza danych

Dane przechowywane w hurtowni podlegają zaawansowanym analizom. Dla potrzeb takich analiz i złożonego przetwarzania danych, rozszerzono standard języka SQL o klauzule i funkcje umożliwiające zaawansowane grupowanie danych, symulowanie funkcjonalności arkuszy kalkulacyjnych, specjalizowane funkcje statystyczne, finansowe, przewidywania trendów. Ponadto, często serwery baz danych oferują wbudowane mechanizmy eksploracji danych.

8.1. Oracle

Oracle9i/10g udostępnia szereg klauzul i funkcji ułatwiających konstruowanie zapytań OLAP. W zakresie wyznaczania agregatów rozszerzono język SQL o klauzule *group by cube*, *group by rollup*, *group by grouping sets* umożliwiające wyliczanie dodatkowych wartości zagregowanych. Innymi słowy, z wykorzystaniem tych klauzul można z danych przechowywanych w tabelach uzyskiwać "2–wymiarowe kostki". Polecenie *select* zostało również rozszerzone o klauzulę *with* umożliwiającą definiowanie w ramach jednego zapytania zmiennych, których wartościami są zbiory wyników innych zapytań. Do takich zmiennych można się odwoływać w zapytaniu głównym. W wersji Oracle10g do polecenia *select* dodano klauzulę *model*, która umożliwia zagnieżdżenie w zapytaniu kodu napisanego w specjalnie do tego celu dedykowanym języku proceduralnym. Możliwe jest m.in. wykorzystanie pętli i zmiennych. Klauzula ta realizuje funkcjonalność arkusza kalkulacyjnego w bazie danych. Wszystkie te rozszerzenia polecenia *select* mają na celu umożliwienie

nie implementowania bardzo złożonych analiz w jednym zapytaniu, którego wykonanie będzie mogło zostać zoptymalizowane przez moduł optymalizatora kosztowego.

Zaawansowane analizy wymagają również stosowania specjalizowanych funkcji analitycznych obliczających m.in.: rankingi, agregaty kumulacyjne, regresje. W tym zakresie, *Oracle9i/10g* oferuje kilkadziesiąt takich funkcji.

Baza danych *Oracle10g* udostępnia również narzędzia programistyczne umożliwiające znajdowanie reguł asocjacyjnych, analizę skupień, klasyfikację i predykcję. Są one dostępne za pomocą interfejsu Java API lub pakietu systemowego *DBMS_DATA_MINING*, składowanych w bazie danych.

8.2. SQL Server

W zakresie analizy danych *SQL Server* udostępnia: wspomniany wcześniej serwer *Analysis Services*, język manipulowania danymi wielowymiarowymi MDX i dwa algorytmy eksploracji danych.

Funkcjonalność *Analysis Services* umożliwia budowanie tabel przestawnych (ang. cross tables), obliczanie agregatów z wykorzystaniem standardowych funkcji, standardowe operacje poruszania się w hierarchii wymiarów (roll-up, drill-down), zmianę orientacji wymiarów (pivoting), zawężanie wymiarów (slice and dice). Ponadto, z wykorzystaniem *Analysis Services* jest możliwa eksploracja danych w oparciu o drzewa decyzyjne (ang. decision trees) i analizę skupień (ang. clustering).

Operacje na danych wielowymiarowych serwera *Analysis Services* realizuje się z wykorzystaniem języka MDX.

8.3. Sybase

Sybase IQ nie posiada swoich narzędzi analitycznych. Istnieje jednak możliwość wykorzystania oprogramowania dostarczonego przez innych producentów. Do najpopularniejszych narzędzi OLAP należą produkty: MicroStrategy, SAS, czy Business Objects. Istnieje również możliwość wykorzystania serwera *Analysis Services*.

Dodatkowo, analizę danych można przeprowadzić bezpośrednio z poziomu SQL, z wykorzystaniem standardowych klauzul i funkcji analitycznych. W tym zakresie *Sybase IQ* oferuje klauzule i funkcje podobne do oferowanych przez *Oracle*.

9. Podsumowanie

W ramach niniejszego projektu przetestowano funkcjonalność trzech popularnych systemów baz danych, tj. *Oracle10g*, *SQL Server 2000* i *Sybase IQ*, pod kątem ich zastosowań dla hurtowni danych. Oceny dokonano pod kątem wsparcia dla: procesu zasilania hurtowni, modeli składowania danych (ROLAP, MOLAP), efektywności przetwarzania danych (indeksowanie, materializowanie, partycjonowanie, przetwarzanie równoległe), oferowanych funkcji analitycznych i eksploracyjnych. Porównanie ww. systemów przedstawiono w Tabeli 1.

Tabela 1. Zestawienie podstawowej funkcjonalności oferowanej przez *Oracle10g*, *SQL Server 2000* i *Sybase IQ*

	Oracle10g	MS SQL Server 2000	Sybase IQ
Model danych i schemat			
ROLAP	schemat gwiazdy, płątka śniegu, konstelacji faktów	schemat gwiazdy, płątka śniegu, konstelacji faktów	schemat gwiazdy, płątka śniegu, konstelacji faktów

MOLAP	analytic workspace, OLAP DML	Analysis Services	brak
Zasilanie/odświeżanie hurtowni (ETL)			
Z poziomu SQL	perspektywy zmaterializowane, tabele zewnętrzne, funkcje tablicowe, perspektywy zmaterializowane	polecenie BULK INSERT, replikacja migawkowa, transakcyjna, scalająca	polecenie LOAD TABLE; polecenie INSERT/LOCATION; wykorzystanie tabel proxy
Narzędzia wspierające	eksport/import, data pump, SQL*Loader	Data Transformation Services	Sybase Enterprise Connect Data Access
Optymalizacja zapytań			
Indeksowanie danych	indeksy B ⁺ -drzewo, bitmapowe, bitmapowe połączeniowe	indeksy klastrowe	indeksy bitmapowe, połączeniowe
Materializowanie	perspektywy zmaterializowane i przepisywanie zapytań	perspektywy zmaterializowane i przepisywanie zapytań	?
Przetwarzanie równoległe	przetwarzanie zapytań, wczytywanie danych, budowanie indeksów	przetwarzanie zapytań	multipleksowanie
Składowanie dużych wolumenów danych			
Partycjonowanie	poziome	poziome – w oparciu o perspektywy	pionowe
Kompresja danych	kompresja tabel i indeksów	kompresja tabel i indeksów	kompresja tabel, agregatów i indeksów
Wsparcie SQL dla przetwarzania analitycznego			
Narzędzia wspierające	Reports, Discoverer, BI Beans	MS SQL Server 2000 Analysis Services	zewnętrzni dostawcy w oparciu o JDBC/ODBC
Język zapytań	SQL, OLAP DML	MDX	SQL
Eksploatacja danych	pakiety systemowe	drzewa decyzyjne, klastrowanie	zewnętrzni dostawcy

Z powyższego zestawienia wynika, że każdy z testowanych systemów oferuje większość funkcjonalności wymaganej w zastosowaniach hurtowni danych. *Oracle10g* oferuje trochę więcej mechanizmów dostępu do zewnętrznych źródeł danych z poziomu bazy danych, bogatszy jest również zbiór obsługiwanych indeksów. W zakresie narzędzi analizy i eksploracji danych, *Sybase IQ* nie posiada własnych rozwiązań i bazuje na produktach dostawców zewnętrznych. *Sybase IQ* odróżnia od konkurentów sposób składowania danych. Poza tym, testowane systemy są porównywalne. Z punktu widzenia użytkownika/administratora, *Oracle* wymaga największej wiedzy i najtrudniej się z niego korzysta. Obsługa *SQL Server* i *Sybase IQ* jest nie wymaga zaawansowanej wiedzy i jest raczej intuicyjna dzięki bogatemu interfejsowi graficznemu.