

Лекція 2.

СХОВИЩА ДАНИХ ТА OLAP - ТЕХНОЛОГІЇ

2.1. Концепція сховищ даних

На початку восьмидесятих років минулого століття, в період бурхливого розвитку реєструючих інформаційних систем, виникло розуміння обмеженості можливостей їх застосування для аналізу даних і побудови на їх основі систем інтелектуального аналізу даних. Реєструючі системи створювалися для автоматизації рутинних операцій по веденню бізнесу - виписки рахунків, оформлення договорів, перевірки стану складу і т. п. Основними вимогами до таких систем були забезпечення транзакційності змін, що вносилися, і максимізація швидкості їх виконання. Саме ці вимоги визначили вибір реляційних СУБД і моделі представлення даних "суть-зв'язок" в якості основного технічного рішення при побудові реєструючих систем.

Для менеджерів і аналітиків у свою чергу були потрібні системи, які б дозволяли:

- аналізувати інформацію в часовому аспекті;
- формувати довільні запити до системи;
- обробляти великі об'єми даних;
- інтегрувати дані з різних реєструючих систем.

Очевидно, що реєструючі системи не задовольняли жодному з вищезгаданих вимог. У реєструючій системі інформація актуальна тільки на момент звернення до бази даних, в наступний момент часу по тому ж запиту можна отримати абсолютно інший результат. Інтерфейс реєструючих систем розрахований на проведення жорстко певних операцій і можливості отримання результатів на нерегламентований (ad-hoc) запит сильно обмежені. Можливість обробки великих масивів даних також мала через налаштування СУБД на виконання коротких транзакцій і неминучого уповільнення роботи решти

користувачів. Відповіддю на виниклу потребу стала поява нової технології організації баз даних - технології сховищ даних (Data Warehouse) [4, 9, 45, 66].

За спостереженнями дослідницької компанії Forrester Research [71], більшість крупних компаній стикаються з наступною проблемою: вони накопичують величезну кількість інформації, яка ніколи не використовується. Практично в будь-якій організації реально функціонує безліч транзакційних систем, орієнтованих на оперативну обробку даних (кожна для конкретного класу задач) і безперервно поповнюють численні бази даних. Окрім цього, часто підприємства володіють величезними об'ємами інформації, що зберігається в так званих успадкованих системах. Всі ці дані розподілені по мережах персональних комп'ютерів, зберігаються на мейнфреймах, робочих станціях і серверах. Таким чином, інформація є, але вона розосереджена, неузгоджена, неструктурована, часто надмірна і не завжди достовірна. Тому в більшості організацій ці дані до цих пір не можуть бути використані для ухвалення критичних бізнес-рішень. На вирішення цього протиріччя і направлена концепція сховищ даних (Data Warehouse) (рис. 2.1.).

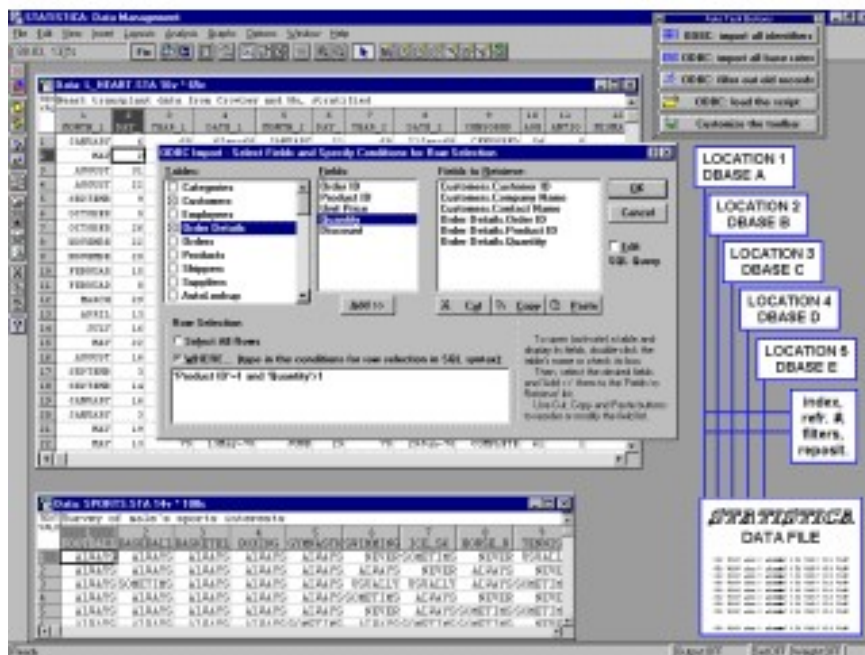


Рис. 2.1. Сховище даних SENS компанії StatSoft Enterprise Systems.

Зберігання даних є наріжним каменем систем бізнес-аналітики. 87% компаній мають одне або більше функціонуючих сховищ даних, причому більше половини мають декілька подібних сховищ. Фактично, кожні чотири компанії мають двадцять або більше сховищ, які деяким способом логічно інтегровані, 18% цих сховищ має об'єм в терабайт або більше, 30% респондентів чекає, що об'єм їх сховища даних зросте удвічі або навіть у декілька разів за подальші два або три роки. Різноманітність джерел даних, що наповнюють сховища, просто дивує. Природно, кожне сховище використовує транзакційні системи як джерело даних. Проте відомо декілька незвичайних джерел даних, такі як Web-сайти електронної комерції (51%), прив'язки до баз даних, розташованих у замовників або бізнес-партнерів (49%), сервіси розширеного доступу до даних, які забезпечують характеристичну інформацію по замовниках (47%), витягання інформації з Web-сайтів (41%), сервіси пошуку/виявлення (34%) і канали новинних даних від крупних новинних агентств (33%). Нарешті, 47% компаній мають одну або більш вітрин даних, і 49% використовують накопичувачі операційних даних.

Білл Інмон, автор концепції, в своїй класичній статті «Що таке сховища даних» (1996 р. с. 4) визначає сховища даних як *«предметно орієнтовані, інтегровані, незмінні, такі, що підтримують хронологію набори даних, організовані для цілей підтримки управління»*. Він розглядає сховища як «єдине і одноосібне джерело істини», «центр всесвіту» систем підтримки прийняття рішень (СППР). «З сховищ даних, - пише він, - інформація перетікає в різні відділи, фільтруючись відповідно до заданих настройок СППР. Ці окремі бази даних для ухвалення рішень називаються вітринами даних» [16, 37, 51].

У основі концепції сховищ даних лежить ідея об'єднання корпоративних даних, розсіяних по системах оперативної обробки даних, історичних архівах і інших зовнішніх джерелах. Ці джерела можуть містити дані, які не використовуються безпосередньо в системах обробки інформації, але що є життєво необхідними для СППР: законодавча база (включаючи податкові прогнози), плани розвитку галузей, статистичні дані, електронні довідники. Як

показує практика, рішення, прийняте на основі лише внутрішніх даних, найчастіше виявляється некоректним.

Мета концепції сховищ даних - прояснити відмінності в характеристиках даних в операційних і аналітичних системах, визначити вимоги даним, що поміщаються в сховище, визначити загальні принципи і етапи його побудови, основні джерела даних, дати рекомендації по рішенню потенційних проблем, що виникають при їх вивантаженні, очищенні, узгодженні, транспортуванні і завантаженні до цільової бази даних сховища.

Предметом концепції сховищ даних є не аналіз даних, а власне дані, тобто концепція їх підготовки для подальшого аналізу. В той же час концепція сховища даних визначає не просто єдиний логічний погляд на корпоративні дані, а реалізацію єдиного інтегрованого джерела даних.

Розглянемо порівняльні характеристики даних, що застосовуються в інформаційних системах, орієнтованих на операційну і аналітичну обробку даних (таб. 2.1.).

Таблиця 2.1.

Порівняння характеристик даних.

Характеристика	Операційні	Аналітичні
Частота оновлення	Висока частота, маленькими порціями	Мала частота, великими порціями
Джерела даних	В основному внутрішні	В основному зовнішні
Об'єми зберігаємих даних	Сотні мегабайт, гігабайти	Гігабайти та терабайти
Вік даних	Поточні (за період від декількох місяців до одного року)	Поточні та історичні (за період в декілька років, десятки років)
Призначення	Фіксація, оперативний пошук і перетворення даних	Зберігання деталізованих і агрегованих історичних даних, аналітична обробка, прогнозування і моделювання

Згідно цієї таблиці сховища даних мають переваги порівняно з використанням оперативних систем або баз даних:

- на відміну від оперативних систем, сховище даних містить інформацію за весь необхідний часовий інтервал - аж до декількох десятиліть - в єдиному інформаційному просторі, що робить такі сховища ідеальною основою для виявлення трендів, сезонних залежностей і інших важливих аналітичних показників;

- як правило, інформаційні системи підприємства зберігають і представляють аналогічні дані по-різному. Наприклад, одні і ті ж показники можуть зберігатися в різних одиницях вимірювання. Одна і та ж продукція або одні і ті ж клієнти можуть іменуватися по-різному. У системах сховищ невідповідності в даних усуваються на етапі збору інформації і занурення її в єдину базу даних. При цьому організуються єдині довідники, всі показники в яких приводяться до однакових одиниць вимірювання;

- дуже часто оперативні системи внаслідок помилок операторів містять деяку кількість невірних даних. На етапі поміщення в сховище даних інформація заздалегідь обробляється. Дані за спеціальною технологією перевіряються на відповідність заданим обмеженням і при необхідності коректуються (очищаються). Технологія забезпечує побудову аналітичних звітів на основі надійних даних і своєчасне сповіщення адміністратора сховища про помилки у вхідній інформації;

- універсалізація доступу до даних. Сховище даних надає унікальну можливість отримувати будь-які звіти про діяльність підприємства на основі одного джерела інформації. Це дозволяє інтегрувати дані, що вводяться і вже накопичені в різних оперативних системах, легко і просто порівнювати їх. При цьому в процесі створення звітів користувач не зв'язаний відмінностями в доступі до даних оперативних систем;

- прискорення отримання аналітичних звітів. Отримання звітів за допомогою засобів, що надаються оперативними системами, - спосіб неоптимальний. Ці системи витрачають значний час на агрегацію інформації (розрахунок сумарних, середніх, мінімальних, максимальних значень). Крім того, в поточній базі оперативної системи знаходяться тільки найнеобхідніші і

свіжіші дані, тоді як інформація за минулі періоди поміщається в архів. Якщо дані доводиться отримувати з архіву, тривалість побудови звіту зростає ще в два три рази. Слід також враховувати, що сервер оперативної системи часто не забезпечує необхідну продуктивність при одночасній побудові складних звітів і введенні інформації. Це може катастрофічно позначатися на роботі підприємства, оскільки оператори не зможуть оформляти накладні, фіксувати відвантаження або отримання продукції в той час, коли виконується побудова чергового звіту. Сховище даних дозволяє вирішити ці проблеми. По-перше, робота сервера сховища не заважає роботі операторів. По-друге, в сховищі крім детальної інформації містяться і наперед розраховані агреговані значення. По-третє, в сховищі архівна інформація завжди доступна для включення в звіти. Все це дозволяє значно скоротити час створення звітів і уникнути проблем в оперативній роботі;

- побудова довільних запитів. Інформацію в сховищі даних недостатньо тільки централізувати і структурувати. Аналітикові потрібні засоби візуалізації цієї інформації, інструмент, за допомогою якого легко отримувати дані, необхідні для ухвалення своєчасних рішень. Одна з головних вимог будь-якого аналітика - простота формування звітів і їх наочність. У разі оперативних систем побудова звітів часто позбавлена гнучкості; щоб створити новий звіт, доводиться задіювати фахівців інформаційного відділу, які об'єднують дані декількох систем. У разі ж використання сховища даних вирішення проблеми надає технологія OLAP (On-Line Analytical Processing). Ця технологія забезпечує доступ до даних в термінах, звичних для аналітика. Технологія OLAP базується на концепції багатовимірного представлення даних. Дійсно, кожне числове значення, що міститься в сховищі даних, має до декількох десятків атрибутів (наприклад, кількість продажів певним менеджером в певному регіоні на певну дату і т.п.). Таким чином, можна вважати, що робота йде з багатовимірними структурами даних (багатовимірними кубами), в яких числові значення розташовані на перетині декількох вимірювань. Саме цей підхід використовується в OLAP-системах. Вони надають гнучкі засоби

навігації по багатовимірних структурах - так звані OLAP-маніпуляції. З їх допомогою аналітик може отримувати різні зрізи даних, "крутити" дані.

Як видно з перерахованих переваг використання технології сховищ даних, велика їх частина може істотно спростити, підвищити швидкість і якісно поліпшити процес Data Mining. Таким чином, комплексне впровадження цих технологій дає розробникам і користувачам незаперечні переваги перед використанням розрізнених баз даних різних інформаційних систем при створенні систем підтримки прийняття рішень [3, 66].

В той же час, на сьогоднішній день є, принаймні, три істотні проблеми, пов'язані зі сховищами даних. Вони полягають в управлінні брудними даними, оптимальному виборі джерела даних, а також в продуктивності і масштабованості операцій, заснованих на скануванні.

З наслідками брудних даних ми стикаємося в своєму повсякденному житті. Ми отримуємо масу поштових відправлень з орфографічними помилками в іменах, безліч послань з різними варіантами одного і того ж імені, масу листів, адресованих людям, які давно переїхали, банківські повідомлення про численні витратні операції (тоді як гроші з рахунку знімалися лише одного разу) і т.д. До брудних даних відносяться відсутні, неточні або даремні дані з погляду практичного застосування (наприклад, представлені в невірному форматі, не відповідному стандарту). Брудні дані можуть з'явитися з різних причин, таких як помилка при введенні даних, використання інших форматів або одиниць вимірювання, невідповідність стандартам, відсутність своєчасного оновлення, невдале оновлення всіх копій даних, невдале видалення записів-дублікатів і т.д. Очевидно, що результати запитів, здобичі даних або бізнес-аналіза над сховищем, що містить велике число брудних даних, не можуть вважатися надійними і корисними. Представлені сьогодні на ринку засоби очищення даних (наприклад, продукти компаній Vality/Ascential Software, Trillium Software і First Logic) допомагають виявляти і автоматично коректувати деякі найбільш важливі типи даних, особливо, імена і адреси людей (з використанням національного каталогу імен і адрес). Проте цим засобам

належить пройти ще довгий шлях, оскільки сьогодні вони не вміють працювати зі всіма типами брудних даних, і далеко не всі компанії використовують навіть наявні засоби. Більш того, більшість підприємств не упроваджують надійні методики і процеси, що гарантують високу якість даних в сховищі. Недостатня увага, що приділяється якості даних, обумовлена відсутністю розуміння типів і об'єму брудних даних, проникаючих в сховища; впливу брудних даних, ухвалення рішень і виконуваних дій; а також тим фактом, що продукти очищення даних, представлені на ринку, не дуже добре рекламуються або дуже дорого стоять. Для того, щоб почати приділяти необхідну увагу якості даних в своїх сховищах, підприємствам, перш за все, потрібно розібратися в різноманітні можливих брудних даних, джерелах їх появи і методах їх виявлення і очищення.

Поза сумнівом, що більшість організацій сьогодні не роблять достатніх зусиль для забезпечення високої якості даних в своїх сховищах. Для забезпечення високої якості даних потрібно мати процес, методології і ресурси для відстежування і аналізу якості даних, методологію для запобігання або виявлення і очищення брудних даних і методології для оцінки вартості брудних даних і витрат на забезпечення високої якості даних. У Ewha Women's University розроблений прототип інструментального засобу DAQUM (Data Quality Measurement), призначений для відстежування більшості типів брудних даних і приписування різним типам брудних даних кількісної міри якості даних залежно від особливостей програм.

Іншою проблемою сховищ даних є проблема вибору джерел даних. Сьогодні проєктувальники сховищ даних проєктують схему бази даних цільового сховища даних з використанням засобів моделювання баз даних. Схема бази даних складається з таблиць, стовпців (полів) таблиць, типів даних і обмежень стовпців, а також зв'язків між таблицями. Проєктувальники також визначають відображення (перетворення) схем джерел інформації на схему цільового сховища даних. Але як проєктувальники можуть переконатися в тому, що сховище даних містить всі дані, потрібні програмним комплексам і не

містить ніяких даних, які додаткам не потрібні? Сьогодні це ґрунтується на основі припущень досвідчених проектувальників. Їм доводиться виявляти потреби даних (таблиці і стовпці), опитуючи розробників програмних додатків, бізнес-аналітиків і адміністраторів баз даних. Після початкового створення сховища часто виявляється, що в ньому відсутні дані, потрібні для отримання відповідей на деякі запити, і присутні дані, які ніколи не потрібні додаткам. Хоча вартість зберігання може бути відносно невеликою, всі поля, потрібні і непотрібні, зберігаються в одних і тих же записах і прочитуються і записуються спільно, що уповільнює швидкість вибірки, збільшує час обробки, а також приводить до неефективного використання середовища зберігання.

Існує багато пропозицій по моделюванню сховища даних у вигляді репозитарія результатів всіх виконуваних запитів. Автори цих пропозицій намагаються знайти алгоритми, які вибиратимуть (для завантаження в сховищі даних) підмножину початкових даних, що мінімізує загальний час відповідей на запити. Деякі з авторів намагаються також мінімізувати вартість оновлення сховища даних. Іншими словами, вони ґрунтуються на припущенні, що всі запити до сховища даних можна наперед дізнатися або передбачити і що можна наперед дізнатися або передбачити всі можливі зміни сховища даних, а отже, і початкових джерел даних.

Ідеальний спосіб вибірки даних для завантаження в сховище даних полягає в тому, що перш за все визначаються всі запити, які генеруватимуться всіма програмними додатками, що виконуються над сховищем даних, і визначаються таблиці і поля, що фігурують в цих запитах. Визначення всіх запитів до створення сховища даних є важким завданням. Проте це може стати можливим після початкового створення сховища даних за рахунок реєстрації протягом розумного проміжку часу всіх запитів, що поступають від додатків. Аналіз зареєстрованих запитів може бути використаний для тонкої настройки сховища даних і видалення даних, до яких додатки не здійснюють доступ.

Потенційно корисним і практичним є засіб, який аналізує потреби додатків в даних, автоматично зіставляє ці потреби з схемами джерел даних і

видає рекомендації по складу оптимального підбору джерел даних, які потрібно завантажити в сховищі даних, щоб в ньому знаходилися всі потрібні дані і не знаходилися які-небудь непотрібні. Таким засобом є MaxCentra. Функціонування MaxCentra спирається на наявність попередньої побудованої бази знань ключових слів, яка представляє потреби програмних додатків в даних. Ключові слова в основному є неявними вказівками таблиць і полів, до яких здійснюватиметься доступ при виконанні запитів, що генеруються додатком. Такий список ключових слів може бути забезпечений бізнес-аналітиками або розробниками програмних додатків, або ж він може бути отриманий автоматично шляхом аналізу запитів від додатків, що виконуються над неоптимізованим сховищем даних. MaxCentra відштовхується саме від цього і за підтримки і сприянні проєктувальників дозволяє отримати оптимальну схему бази даних для сховища даних.

Для сховищ даних існує також проблема продуктивності та масштабованості. У системах реляційних баз даних для вибірки невеликої кількості потрібних записів без повного сканування таблиці або бази даних використовуються різні методи доступу, такі як індекси на основі хешування або В+-дерев. Такі методи доступу вельми ефективні при вибірці по одному ключовому полю (або невеликому числу полів), коли результати є малою частиною таблиці. Прикладами подібних запитів є: «Знайти всіх 25-річних людей» або «Знайти всіх 25-річних інженерів-програмістів». Для швидкої відповіді на такі запити можна створити індекси: на стовпці «вік» таблиці «співробітники» або на стовпцях «вік» і «посада» таблиці «співробітники» відповідно. Проте методи доступу в загальному випадку не допомагають при відповіді на запити, результатами яких є значна частина таблиці. Прикладами є запити: «Знайти всіх співробітників жіночого полу» або «Знайти молодих співробітників». Крім того, методи доступу не приносять користі, якщо значення стовпця часто змінюється, оскільки такі зміни вимагають перебудови методів доступу.

Крім подібних «простих» запитів існують два класи операцій, для яких методи доступу в системах реляційних баз даних стають безсилими. До першого класу відносяться операції «агрегації», що передбачають групування всіх записів таблиці і застосування до згрупованих записів агрегатних функцій (середнього значення, загального числа, суми, мінімального або максимального значення). Цей тип операцій важливий в таких програмних додатках як аналіз даних Web-журналів, сегментації даних про замовників і т.д. На ринку є продукти MaxScan і Ab Initio, призначені для вирішення проблем продуктивності і масштабованості при виконанні даного типу операцій. У MaxScan застосовується метод зберігання таблиць по полях, методи хешування для групування і агрегації записів, а також методи паралельної обробки. Продуктивність і масштабованість при виконанні агрегатних операцій в 10-20 разів перевищують показники систем реляційних баз даних. Пакет Ab Initio є засобом ETL, в якому в механізмі трансформації даних використовуються методи підвищення продуктивності.

До іншого класу відноситься операції «переміщення файлів», що читають і/або записують файли цілком. Цей тип операцій важливий на етапі «перетворення даних», що вимагає великих тимчасових витрат при створенні сховища даних або на етапі «підготовки даних» при автоматичному витяганні знань (здобичі даних) з наявних джерел. Етап перетворення даних включає трансформацію формату і представлення даних в заданих полях (зміна одиниці вимірювання, зміна формату дати і часу, зміна аббревіатури і т.д.), злиття двох або більше полів в одне, розщеплювання поля на два або більше полів, сортування таблиці, побудову узагальненої таблиці з таблиць, що містять деталізовані дані, створення нової таблиці шляхом з'єднання двох або більше таблиць, розщеплювання таблиці на дві або більше і т.д. До етапу підготовки даних відноситься перетворення даних заданого поля в цифровий код (у нейронних мережах), перетворення безперервних цифрових даних в заданому полі в категоричні дані (наприклад, вік, що перевищує 60 років, вважається «пенсійним»), додавання до запису нового поля, взяття з таблиці зразків даних,

реплікація в таблиці деяких записів (для досягнення бажаного розподілу записів) і т.д.

Сьогодні операції переміщення файлів знаходяться в майже повній залежності від послідовних операцій систем реляційних баз даних над файлами, тобто читання одного або більше файлів, створення тимчасового файлу і запису результуючого файлу або файлів. Частота виконання подібних операцій і об'єм використовуваних даних може зробити виправданим застосування сервера перетворення/підготовки даних. У ідеалі такий сервер може складатися з декількох паралельно працюючих процесорів. Незалежно від конфігурації процесора на ньому повинні виконуватися програмні засоби перетворення/підготовки даних, спроектовані для паралельної обробки. Коли це виправдано, потрібно застосовувати конвеєрну паралельну обробку, при якій отримання часткових результатів однієї операції ініціюють виконання іншої операції без потреби очікування завершення першої операції. Конвеєрна паралельна обробка усуває потребу в записі в тимчасовий файл повних результатів однієї операції і в їх читанні наступною операцією, що дозволяє заощадити декілька процесів обробки файлів.

Технологія сховищ даних зародилася в США, а потім отримала визнання по всьому світу. Перші сховища даних з'явилися ще на початку 80-х років минулого століття (тоді вони називалися бази атомарних даних), але тільки до кінця 80-х років була повною мірою усвідомлена необхідність інтеграції корпоративної інформації і належного управління нею, а також з'явилися технічні можливості для створення відповідних систем, спочатку названих "сховищами інформації" (Information Warehouse - IW) і лише пізніше отримавших свою сучасну назву "сховища даних" (Data Warehouse - DW) . На початку 90-х, з появою технологій витягання, перетворення і завантаження даних (Extraction, Transmission and Loading - ETL) і оперативної аналітичної обробки (On-Line Analytic Processing - OLAP) почалося активне розповсюдження сховищ даних в комерційному секторі. Цьому процесу сприяло також опублікування першої книги Білла Інмона (W.H. Inmon. Building

the Data Warehouse, QED/Wiley, 1991, 312 p.), який отримав загальне визнання як "батько концепції сховища даних". Незабаром технологія сховищ даних перетворилася на розвинену архітектуру, відому як фабрика корпоративної інформації (Corporate Information Factory - CIF).

Іншим важливим напрямом, який сприяв появі сховищ даних, була поява систем підтримки ухвалення рішень (DSS — Decision Support Systems) та інформаційних систем для вирішень (EIS — Executive Information Systems). Не претендуючи на володіння власним інтелектом, на відміну від всього того, що обговорювалося в 60–70-і роки і що залишилося в області вічних ідей, системи DSS і EIS виявилися практично корисними: вони стали прототипами сучасних систем розтину даних, онлайнної аналітики, нової дисципліни корпоративного управління на основі знань. Системи DSS були спочатку налаштовані на менеджмент середньої ланки, а EIS намагалися дати більш загальне і багатовимірне бачення над полем даних — для керівників корпоративного рівня. Саме ці дві технологічні ідеї, що глибоко перетиналися між собою, з'явилися, найімовірніше, прямими предками сучасної концепції сховищ даних.

Розглянемо типові проблеми, що вирішуються за допомогою сховищ даних. До них відносяться, зокрема, аналіз клієнтської бази, аналіз продажів і аналіз доходів, а також управління пасивами і активами та інші.

Аналіз клієнтської бази дозволяє сформулювати цільові сегменти клієнтів і використовувати цю інформацію при продажі банківських продуктів і послуг. Цільові сегменти формуються на основі демографічних і фірмографічних відомостей, фінансових показників (наприклад, обороту або прибутку), галузевих ознак і інших параметрів клієнтів. Одним з найбільш важливих питань є виділення сегментів прибуткових клієнтів, націлене на їх подальше утримання. Зокрема, за рахунок детальнішої сегментації підрозділи маркетингу починають краще розуміти потреби клієнтів і можуть використовувати ці дані при проведенні маркетингових кампаній. Аналіз клієнтської бази і сегментація дають можливість наблизитися до реалізації концепції індивідуального

маркетингу і ефективніше застосовувати систему управління взаєминами з клієнтами.

Аналіз продажів допомагає виявляти тенденції, планувати продажі по продуктах, клієнтах, підрозділах і, виходячи з результатів збуту, будувати механізми стимулювання клієнтських і продуктових підрозділів. Завдяки використанню сховища даних можна отримати інтегроване уявлення про результати продажів і узяти цю інформацію на озброєння при формуванні планів.

Аналіз доходів актуальний для будь-якого банку, причому понад усе затребуваний аналіз в розрізі клієнтів. Дуже важливо також мати уявлення про розподіл доходів по продуктах і послугах, каналах надання послуг і підрозділах банку. Аналіз доходів в розрізі клієнтів і продуктів дозволяє формувати “унікальні” пропозиції для кожного “унікального” клієнта з метою максимізації прибутку в довгостроковій перспективі. Він сприяє формуванню цінової політики банку, виділенню сегментів, продуктів і послуг, які стратегічно важливі для нього. Наприклад, банк Chase Manhattan Bank має сховище об'ємом більше 560 Гбайт, компанія Mastercard Online - 1,2 Тбайт. Коли всі дані містяться в єдиному сховищі, вивчення зв'язків між окремими елементами даних може бути пліднішим, а результатом аналізу стають нові знання.

Управління активами і пасивами. За допомогою сховища даних можна проводити ефективний аналіз активів і пасивів і управляти не тільки ними, але і миттєвою ліквідністю банку на основі інструментального і портфельного підходів. Ці завдання вирішуються при мінімальних витратах на підготовку спеціальних даних і з урахуванням лише обмеженого об'єму інформації, що збирається з джерел у філіалах. Програмний комплекс забезпечує завантаження з інформаційних джерел семи типів і дозволяє формувати декілька десятків звітів.

Існують різні типи сховищ даних, які мають свою специфіку.

Фінансові сховища даних. В більшості випадків фінансові сховища даних це сховища, які організації будують в першу чергу. Створення фінансового сховища - дуже привабливе рішення, оскільки:

- фінансові дані завжди знаходяться в центрі «мозоку» організації;
- у більшості організацій фінансові дані представляють найменші об'єми даних з тих, що є;
- фінанси охоплюють всі аспекти функціонування компанії;
- фінансові дані за своєю природою мають структуру, на яку безпосередньо впливає повсякденна практика обробки фінансової інформації.

З цих причин фінанси стають переважною областю побудови корпоративного сховища даних. Проте, фінансові сховища даних мають серйозні, властиві тільки цьому типу сховищ, недоліки. Перший з них полягає в тому, що в організаціях чекають, що відомості з фінансових сховищ з точністю до однієї копійки співпадатимуть з даними існуючого фінансового середовища. Раз у раз можна почути, що «це фінансове сховище явно несправне, тому що в звіті, який я отримав вчора, було вказано, що доходи складають 145,998.32 доларів, коли ж я виконав той же звіт у фінансовому сховищі даних, то отримав величину, рівну 139, 762.01 долларам. Цьому сховищу просто не можна вірити». Очікування того, що інформація у фінансовому сховищі даних повинна точнісінько співпасти з цифрами з поточного фінансового звіту, є глибоко помилковим. Люди (тобто фінансові працівники), які так думають, просто не розуміють, що, коли дані переходять з операційного середовища у фінансове сховище даних, відбувається трансформація. А коли дані перетікають з світу додатків в реальний світ компанії, їх розглядають в іншому вимірюванні. А ось, що точно відбувається при такому переході даних з одного світу в іншій:

- міняються звітні періоди. У операційному середовищі звітний період завершується в кінці місяця, в середовищі сховища даних закінчується на корпоративному календарі;
- міняються схеми угруповання і кодування рахунків. У операційному середовищі дані розраховуються відповідно до одного плану бухгалтерських

рахунків, а у фінансовому середовищі всієї компанії може бути абсолютно інший набір схеми угруповання і кодування;

- міняються класифікації даних. Так, в операційному середовищі Північна Америка складається всього з 48 континентальних штатів, в глобальному сховищі даних Північна Америка включає також Канаду, Мексику, Аляску і острови Карибського басейну.

- міняються валюти. Операційні грошові кошти відповідають тій валюті, в якій вони звертаються: гривни, євро, фунти, долари і так далі. У глобальному середовищі гроші перетворюються до однієї загальної валюти: доларам або євро.

Сховища даних в області страхування. Сховища даних в області страхування за деякими невеликими виключеннями схожі на інші сховища. Перше виключення (і це особливо справедливо відносно страхування життя) полягає в тому, що тривалість існування наявних сховищ дуже велика. Такі сховища містять дані, які є старими, дуже старими. Причина, по якій страхові компанії вимушені цікавитися такими даними - актуарна обробка даних. Практично для кожної справи приводиться довід, що діяльність, якою організація займалася в 1950 році, практично не пов'язана з сьогоdnішнім заняттям. І часто цей довід звучить правдоподібно.

Друга відмінність цих сховищ визначається датами, які зберігаються в цьому бізнесі. Середовище страхування - відрізняється наявністю величезного числа дат, пов'язаних з бізнесом, чим який-небудь інший видом діяльності. Так, у сфері роздрібної торгівлі є декілька важливих дат: дата продажу, дата появи на складі, можливо, дата виробництва. У банківській справі істотна дата транзакції. У телекомунікації - це дата телефонного дзвінка. У страхуванні ж присутні дати всіляких типів.

Нарешті, третя відмінність полягає в тому, що ці сховища даних використовують свій робочий цикл ділової активності. Більшість організацій має вельми обмежений і короткий економічний цикл. Так, в банках це обналічування чека. У торгівлі - покупка виробу. У телефонній компанії - дзвінок. Проте, в страхуванні їм може бути заявка на страхове відшкодування,

яка може бути задоволена через п'ять років. Або закриття поліса може супроводжуватися двомісячним відстроченням. Резюмуючи, можна сказати, що швидкість, з якою функціонує страхування, відрізняється від швидкості, характерної для інших галузей. Ця різниця в швидкості відбивається в сховищі даних. У інших сховищах транзакції просто збираються і обробляються. В області страхування транзакція може відкладатися на невизначений термін, а її різні частини можуть відбиватися в сховищі даних. Результатом цього є абсолютно особливий підхід при проектуванні і впровадженні таких сховищ даних.

Сховища даних для управління людськими ресурсами. Сховища даних для управління людськими ресурсами мають вельми істотні відмінності від інших сховищ. Перша відмінність - число предметних областей. Таке сховище даних неминуче має одну важливу предметну область - це працівник. Практично все інше підпорядковане цій області або займає другорядне положення. Більшість же інших сховищ даних мають декілька базових предметних областей. Проте, основна відмінність сховищ даних для управління людськими ресурсами полягає в тому, що такі сховища взагалі-то використовують дуже мало транзакцій. Так, є дата, коли суб'єкт стає працівником, дата, коли людина звільняється, а також річні надбавки і підвищення. Але, окрім транзакцій фонду заробітної плати і інших рідкісних, згенерованих працівником транзакцій, в такому сховищі практично більше нічого і немає. Порівняйте сферу управління людськими ресурсами з комунікацією або банківським середовищем, і різниця в числі транзакцій стане очевидною. Ця різниця в темпах транзакцій між тією, що розглядається і іншими сферами діяльності є причиною виникнення певної складності, яка полягає в тому, що в області управління людськими ресурсами спостерігається тенденція до об'єднання операційної обробки людських ресурсів і обробки людських ресурсів для систем ухвалення рішення в одне середовище. У інших же галузях спокуса зробити таку архітектурну помилку вельми невелика.

Глобальні сховища даних. Глобальні сховища даних призначені для глобального представлення корпорації. Розрізняють три типи таких сховищ:

- географічно превалююча обробка даних. Наприклад, необхідно інтегрувати бізнес в Гонконзі з бізнесом в Парижі, який у свою чергу слід інтегрувати з Ріо-де-Жанейро, а той - з Нью-Йорком.
- функціонально превалююча обробка даних. Виробнича діяльність повинна бути інтегрована з постачанням, яке необхідно інтегрувати з продажами, а ті - з дослідженнями і так далі.
- галузева превалююча обробка даних. Наприклад, потрібно інтегрувати друкарську справу з консалтингом, який підлягає інтеграції з бізнесом у сфері медичного устаткування, а той із спеціалізацією в області програмного забезпечення.

Особливість глобального сховища даних полягає в тому, що на глобальному рівні часто дуже мало загальних вимірювань. Єдине загальне вимірювання - це гроші. І інтеграція бізнесу може бути досягнута тільки з його допомогою. Інші ж вимірювання можуть мати або не мати сенсу на глобальному рівні. Так, клієнт, продукт, постачальник, транзакція - всі ці класичні предметні області можуть бути як присутніми, так і бути відсутнім в глобальній інтегрованій сфері - глобальному сховищі даних. Крім цього, глобальне сховище даних схильне до того, від чого інші сховища захищені - від руйнівної дії змін. Якщо в інших сховищах зміни базових даних трапляються нечасто, то для цього типу сховищ вони відбуваються постійно і в самій основі. Так, у будь-який момент може бути відкрите нове родовище нафти, наприклад, у Венесуелі. У наступну хвилину в Перу спалахне революція. А потім, завдяки розвитку технології, стануть доступними поклади нафти в Луїзіані. Услід за цим послідує санкції ОПЕК. У Мексиці буде змінено законодавство. І так далі. Якщо розглядати ситуацію в глобальному аспекті, то видно, що зміни носять постійний характер. Тому структура і технологія, використовувана для розміщення і обслуговування глобального сховища даних, повинні дозволяти підтримувати ці безперервні зміни.

Сховища даних з можливостями Data Mining/Data Mining і Exploration.

Сховища даних, що підтримують технологію Data Mining і Exploration є гібридом класичних сховищ. Такі сховища використовуються для виконання могутньої статистичної обробки даних. Ці сховища є дуже детальними, глибоко історичними, оптимізованими для статистичного аналізу. Крім того, для таких сховищ характерна орієнтація на який-небудь проект. Це означає, що, на відміну від всіх інших типів сховищ даних, їх перестають використовувати відразу після закінчення аналізу, ради якого вони створювалися. Ще одна важлива відмінність сховищ даних з можливостями Data Mining/Data Mining і Exploration полягає в тому, що ці сховища дуже часто включають зовнішні дані. Такі дані дуже корисні з погляду забезпечення бізнес-перспективи, яку не так легко побачити без їх участі.

Сховища даних в області телекомунікацій. Відмітна особливість цих сховищ полягає в тому, що вони в значній мірі визначаються даними, що згенеровані в деталях на рівні дзвінка. Зрозуміло, в галузі телекомунікації присутня безліч інших типів даних. Але жодна інша область сховищ даних не зумовлюється в такому ступені розміром однієї предметної області - деталями на рівні дзвінка. Існують багато способів зберігання деталей на рівні дзвінка:

- зберігання деталей на рівні дзвінка тільки за декілька місяців;
- зберігання безлічі деталей на рівні дзвінка, розміщених на різних носіях;
- резюмування або агрегація деталей на рівні дзвінка;
- зберігання тільки відібраних деталей на рівні дзвінка, і так далі.

На жаль, не дивлячись на різноманітність методів обробки, для даного сховища даних обробка може бути виконана тільки над деталями на рівні дзвінка. А робота на підсумковому або агрегованому рівні просто неможлива.

2.2. Технології побудови сховищ даних

Ідея, покладена в основу технологій інформаційних сховищ, полягає в тому, що проводити оперативний аналіз безпосередньо на базі інформаційних систем неефективно. Натомість, всі необхідні для аналізу дані витягуються з декількох традиційних баз даних (в основному, реляційних), перетворюються і потім поміщаються в одне джерело даних – сховище даних [4, 9, 11, 49].

В процесі занурення дані:

- очищаються - усунення непотрібної інформації;
- агрегуються - обчислення сум, середніх;
- трансформуються - перетворення типів даних, реорганізація структур зберігання;
- об'єднуються із зовнішніх і внутрішніх джерел - приведення до єдиних форматів;
- синхронізуються - відповідність одному моменту часу.

Сьогодні, технології побудови сховищ даних є основою для створення повноцінних інтелектуальних систем аналізу даних, орієнтованих на рішення слабо структурованих задач прийняття рішень, оскільки вони містять дані, що володіють наступними властивостями:

Цілісністю і внутрішнім взаємозв'язком. Хоча дані занурюються з різних джерел, але вони об'єднані єдиними законами іменування, способами вимірювання атрибутів і т.д. Це має велике значення для корпоративних організацій, в яких одночасно можуть експлуатуватися різні по своїй архітектурі обчислювальні системи, що представляють однакові дані по-різному. Наприклад, можуть використовуватися декілька різних форматів представлення дат, або один і той же показник може називатися різним чином, наприклад, "вірогідність доведення інформації" і "вірогідність отримання інформації". В процесі занурення подібні невідповідності усуваються автоматично.

Предметною орієнтованістю. Локальні бази даних містять мегабайти інформації, абсолютно не потрібної для аналізу (адреси, поштові індекси, ідентифікатори записів і т.п.). Подібна інформація не заноситься в сховище, що обмежує спектр розглядаємих при ухваленні рішення даних до мінімуму.

Відсутністю часової прив'язки. Оперативні системи охоплюють невеликий інтервал часу, що досягається за рахунок періодичної архівації даних. Сховища даних, навпаки, містять історичні дані, накопичені за великий інтервал часу (роки, десятиліття).

Доступністю виключно для читання. Модифікація даних не проводиться, оскільки вона може привести до порушення цілісності сховища даних. Оскільки не потрібно мінімізувати час занурення, то структура сховища може бути оптимізована для обробки певних запитів, що досягається за рахунок денормалізації реляційної схеми, попередньої агрегації і побудови найбільш доречних індексів.

Інтегрованість означає, що дані задовольняють вимогам всього підприємства, а не одній функції бізнесу. Цим сховище даних гарантує, що однакові звіти, що згенерували для різних аналітиків, міститимуть однакові результати.

Незмінність означає, що, потрапивши один раз в сховищі, дані там зберігаються і не змінюються. Дані в сховищі можуть лише додаватися.

Всі дані, які містяться в сховищі, можна розділити на наступні категорії:

- метадані (дані про даних);
- агреговані дані;
- детальні дані.

Важливою особливістю систем інтелектуального аналізу даних на основі сховищ даних є метадані. Це різного роду системні словники, що дозволяють контролювати склад і структуру інформації в сховищі, управляти процесами завантаження і розрахунків і т.п. Без прошарку управлінських даних сховище з часом загрожує перетворитися на велике електронне звалище.

Ключовою відмінністю інформації сховища є не тільки спосіб його наповнення (з різних зовнішніх і внутрішніх джерел), але і модель зберігання даних, особливо це стосується агрегованої інформації. У сховищі інформація розміщується в денормалізованому вигляді у формі класичної «сніжинки» або «зірки». Такий підхід дозволяє істотно понизити час відгуку бази даних при виконанні запитів. Не вдаючись до технологічних особливостей проектування, відзначимо, що відбувається це за рахунок деякої надмірності зберігання даних. Така форма зберігання інформації в корпоративних сховищах є загальносвітовою практикою. Разом з тим частина детальних даних цілком може зберігатися в нормалізованих таблицях. Вимога до спеціальних структур зберігання обумовлена деяким спеціальним способом використання даних, про що доцільно сказати докладніше. Використовувати інформацію, накопичену в сховищі, можна як за допомогою традиційних звітів, так і з використанням динамічних запитів до бази даних. Існують також абсолютно специфічні способи використання інформації, призначені спеціально для аналітичних завдань. До них відносяться так звані OLAP-технології (On-line Analytical Processing) і технології інтелектуального аналізу даних. З практичної точки зору рішучий крок, який робить OLAP-технологія, полягає в тому, щоб, відмовившись від зайвої спільності, зробити процес аналізу максимально швидким. В рамках цієї технології передбачається, що склад і структура показників для аналізу відомий наперед і міняється дуже рідко (для деяких видів систем - практично не міняється). Користувач може виконувати над даними в такому багатовимірному уявленні набір OLAP-операцій підйому (консолідації по деяких напрямках), спуску (деталізації по деякому напрямку), повороту (зміни напрямку сортування). Детальні дані в системах сховищ даних найчастіше є джерелом для інтелектуального аналізу.

Таким чином, дані, занурені в інформаційне сховище даних, організовуючись в інтегровану цілісну структуру, володіють природними внутрішніми зв'язками, набувають нових властивостей, що дає їм можливість набутти статус інформації.

Розглянемо характеристики інтеграції даних в сховищі даних. Як вже відомо, метою інтеграції даних є отримання єдиної та цільної картини бізнес-даних. Для її досягнення застосуємо модель, яка включає додатки, продукти, технології та методи:

- додатки - це рішення, створені постачальниками відповідно до вимог клієнтів, які використовують один або більше продуктів інтеграції даних;
- продукти - це готові комерційні рішення, що підтримують одну або більше технології інтеграції даних;
- технології реалізують один або більше методів інтеграції даних;
- методи - це підходи до інтеграції даних, незалежні від технологій.

Існує три основні методи інтеграції даних: консолідація, федералізація і розповсюдження (рис. 2.2).

Консолідація даних. При використанні цього методу дані збираються з декількох первинних систем і інтегруються в одне постійне місце зберігання. Таке місце зберігання може бути використане для підготовки звітності і проведення аналізу, як у випадку застосування сховища даних, або як джерело даних для інших додатків, як у випадку впровадження операційного складу даних. При використанні цього методу зазвичай існує деяка затримка між моментом оновлення інформації в первинних системах і часом, коли ці зміни з'являються в кінцевому місці зберігання. Залежно від потреб бізнесу таке відставання може складати декілька секунд, годин або багато днів. Термін *«режим, наближений до реального часу»* часто використовується для опису кінцевих даних, оновлення яких відстає від джерела на декілька секунд, хвилин або годин. Дані, що не відстають від джерела, вважаються даними *«в режимі реального часу»*, але це важко досягнути при використанні методу консолідації даних.

Перевагою консолідації даних є те, що цей підхід дозволяє здійснювати трансформацію значних об'ємів даних (реструктуризацію, узгодження, очищення і/або агрегацію) в процесі їх передачі від первинних систем до кінцевих місць зберігання. Деякі складнощі, пов'язані з даним підходом, - це

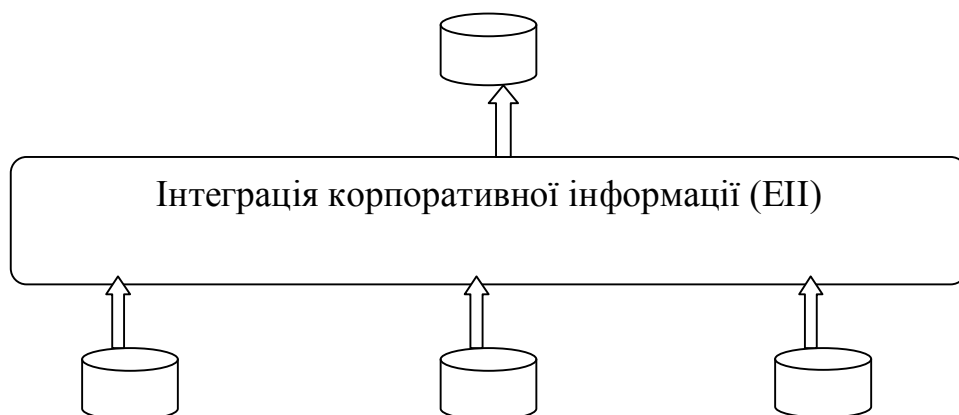
значні обчислювальні ресурси, які потрібні для підтримки процесу консолідації даних, а також істотні ресурси пам'яті, необхідні для підтримки кінцевого місця зберігання. Але з урахуванням постійного вдосконалення апаратних засобів це не проблема.

Консолідація даних - це основний підхід, який використовується програмними додатками сховищ даних для побудови і підтримки оперативних складів даних і корпоративних сховищ. Консолідація даних також може знайти застосування для створення залежної вітрини даних, але в цьому випадку в процесі консолідації використовується тільки одне джерело даних (наприклад, корпоративне сховище). У середовищі сховищ даних однією з найпоширеніших технологій підтримки консолідації є технологія ETL (витягання, перетворення і завантаження - extract, transform, and load). Ще одна поширена технологія консолідації даних - управління змістом корпорації (Enterprise Content Management - ECM). Більшість рішень ECM направлені на консолідацію і управління неструктурованими даними, такими як документи, звіти і веб-сторінки.



Федералізація даних

Віртуальна картина бізнесу



Розповсюдження даних

Розподілені дані



Рис. 2.2. Методи інтеграції даних.

Федералізація даних. Процес забезпечує єдину віртуальну картину одного або декількох первинних файлів даних. Якщо бізнес-додаток генерує запит до цієї віртуальної картини, то процесор федералізації даних витягує дані з відповідних первинних складів даних, інтегрує їх так, щоб вони відповідали віртуальній картині і вимогам запиту, і відправляє результати бізнес-додатку від якого прийшов запит. За визначенням, процес федералізації даних завжди

полягає у *витяганні* даних з первинних систем на підставі зовнішніх вимог. Всі необхідні перетворення даних здійснюються при їх витяганні з первинних файлів. Інтеграція корпоративної інформації (Enterprise Information Integration - ЕІ) - це приклад технології, яка підтримує федеральний підхід до інтеграції даних. Один з ключових елементів федеральної системи - це метадані, які використовуються процесором федералізації даних для доступу до первинних даних. В деяких випадках ці метадані можуть складатися виключно з визначень віртуальної картини, які ставляться у відповідність («мепіруються») первинним файлам. У більш передових рішеннях метадані також можуть містити детальну інформацію про кількість даних, що знаходяться в первинних системах, а також про шляхи доступу до них. Така розширена інформація може допомогти федеральному рішенню оптимізувати доступ до первинних систем.

Вважається, що основна перевага федерального підходу - той факт, що він забезпечує доступ до поточних даних і позбавляє від необхідності консолідувати первинні дані в новому сховищі даних. Але слід пам'ятати, що федералізація даних не дуже добре підходить для витягання і узгодження великих масивів даних або для тих додатків, де існують серйозні проблеми з якістю даних в первинних системах. Ще один істотний чинник - потенційний вплив на продуктивність і додаткові витрати на доступ до численних джерел даних під час виконання програми.

Федеральна архітектура дуже корисна для крупних транснаціональних корпорацій і є вельми зручним підходом для підтримки балансу між необхідністю автономії місцевих підрозділів компанії і їх гнучкості, з одного боку, і стандартизації і централізованого контролю, які здійснює центральний офіс, - з іншою. При цьому федеральним сховищем може бути як єдине фізичне федеральне сховище, так і федерація дрібніших спеціалізованих сховищ даних.

Розповсюдження даних. Додатки розповсюдження даних здійснюють копіювання даних з одного місця в інше. Ці додатки зазвичай працюють в оперативному режимі і проводять переміщення даних до місць призначення, тобто залежать від певних подій. Оновлення в первинній системі можуть

передаватися в кінцеву систему синхронно або асинхронно. Синхронна передача вимагає, щоб оновлення в обох системах відбувалися під час однієї і тієї ж фізичної транзакції. Незалежно від використовуваного типу синхронізації, метод розповсюдження гарантує доставку даних в систему призначення. Така гарантія - це ключова відмінна ознака розповсюдження даних. Більшість технологій синхронного розповсюдження даних підтримують двосторонній обмін даними між первинними і кінцевими системами. Прикладами технологій, що підтримують розповсюдження даних, є інтеграція корпоративних додатків (Enterprise Application Ntegration - EAI) і тиражування корпоративних даних (Enterprise Data Replication - EDR).

Великою перевагою методу розповсюдження даних є те, що він може бути використаний для переміщення даних в режимі реального часу або близькому до нього. Інші достоїнства включають гарантовану доставку даних і двостороннє розповсюдження даних. Метод розповсюдження даних може також використовуватися для урівноваження робочого навантаження, створення резервних копій і відновлення даних, зокрема у разі надзвичайних ситуацій. Практичне застосування цього методу відрізняється чималою різноманітністю як в плані продуктивності, так і відносно можливостей реструктуризації і очищення даних. Деякі корпоративні продукти розповсюдження даних можуть підтримувати переміщення і реструктуризацію крупних масивів даних, тоді як продукти EAI часто мають обмежені можливості пересування великої кількості даних і їх реструктуризації. Одна з причин подібної відмінності - той факт, що в центрі архітектури тиражування корпоративних даних лежать дані, а в центрі технології EAI - повідомлення або транзакції.

Гібридний підхід. Методи, які використовуються додатками інтеграції даних, залежать як від потреб бізнесу, так і від технологічних вимог. Достатньо часто додаток інтеграції даних використовує так званий гібридний підхід, який включає декілька методів інтеграції. Хороший приклад такого підходу - інтеграція даних про клієнтів (Customer Data Ntegration - CDI), метою якої є

забезпечення узгодженої картини інформації про клієнтів. Найпростіший підхід до CDI - це створення консолідованого сховища даних про клієнтів, який містить дані, отримані з первинних систем. Відставання інформації в консолідованому сховищі залежатиме від режиму консолідації даних (оперативний або пакетний) і від частоти оновлення цієї інформації. Інший підхід до CDI - це федералізація даних, коли визначаються віртуальні бізнес-представлення даних про клієнтів в первинних системах. Ці уявлення використовуються бізнес-додатками для доступу до поточної інформації про клієнтів в первинних системах. При федеральному підході також може використовуватися довідковий файл метаданих для зв'язку інформації про клієнтів на основі загальних ключових елементів.

Гібридний підхід, що використовує як консолідацію, так і федералізацію даних, також може мати місце. Загальні дані про клієнтів (ім'я, адреса і т.д.) можуть бути консолідовані в одному сховищі, а дані, які відносяться до певного первинного додатку (наприклад, замовлення), можуть бути федералізовані. Такий гібридний підхід може бути розширений за рахунок розповсюдження даних. Якщо клієнт оновлює своє ім'я і адресу під час транзакції в Інтернет-магазині, то ці зміни можуть бути відправлені до консолідованого сховища даних, а звідти поширені в інші первинні системи, такі як база даних про клієнтів роздрібного магазину.

На сьогоднішній день існує два основні підходи до архітектури сховищ даних. Це так звана корпоративна інформаційна фабрика (Corporate Information Factory - CIF) Білла Інмона і сховище даних з архітектурою шини (Data Warehouse Bus - BUS) Ральфа Кимболла. Розглянемо кожний з них докладніше.

Corporate Information Factory. На рис. 2.3 представлений підхід, використовуваний в сховищах даних з архітектурою CIF. Колись цей підхід був відомий під назвою корпоративного сховища даних (Enterprise Data Warehouse - EDW). Робота такого сховища починається з скоординованого витягання даних з джерел. Після цього завантажується реляційна база даних з третьою нормальною формою, що містить атомарні дані. Отримане нормалізоване

сховище використовується для того, щоб наповнити інформацією додаткові репозиторії презентаційних даних, тобто даних, підготовлених для аналізу. Ці репозиторії, зокрема, включають спеціалізовані сховища для вивчення і «здобичі» даних (Data Mining), а також вітрини даних [16, 71].

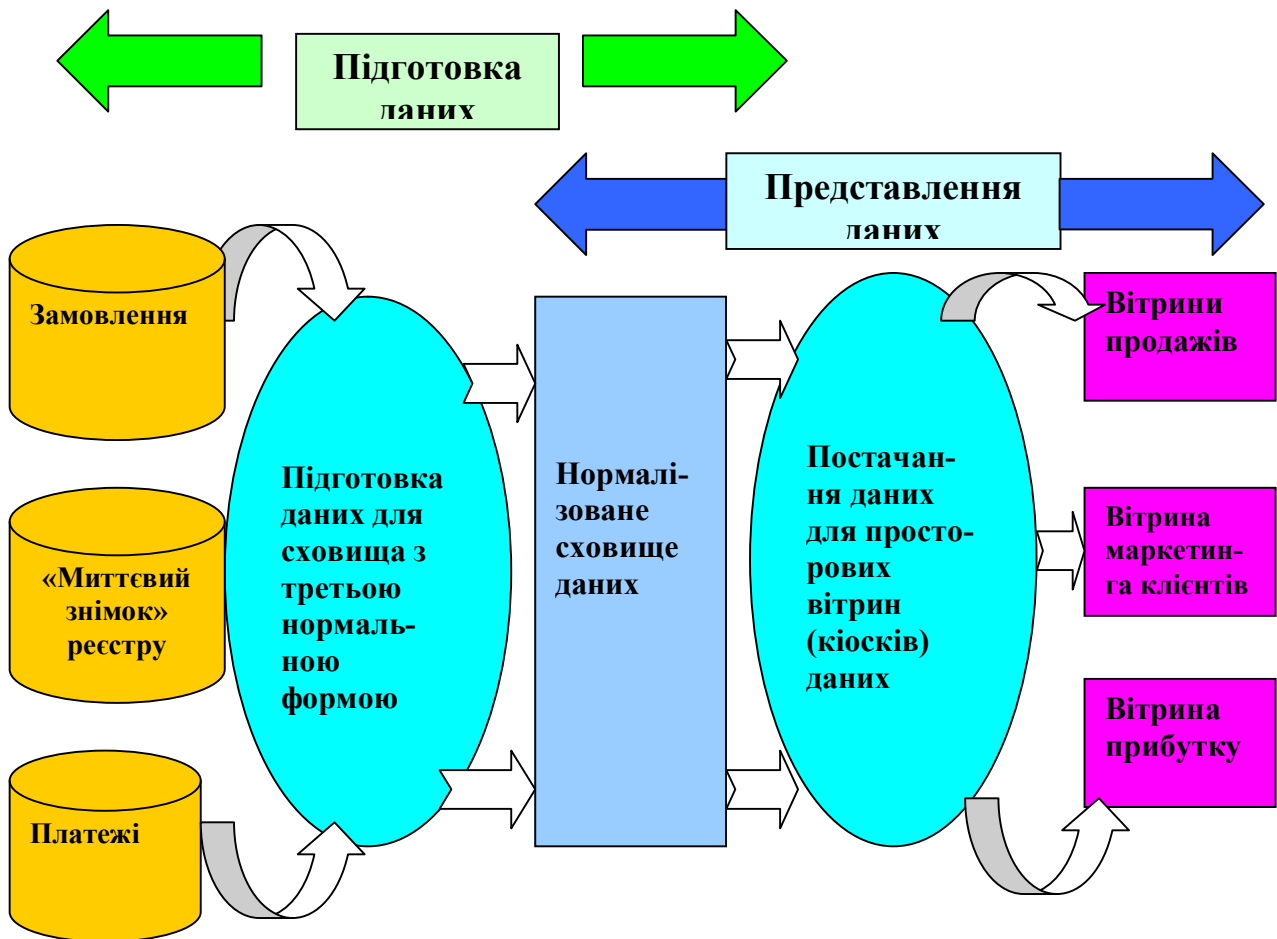


Рис. 2.3. Нормалізоване сховище даних з просторовими вітринами підсумкових даних (CIF).

При такому сценарії кінцеві вітрини даних створюються для обслуговування бізнес-відділів або для реалізації бізнес-функцій і використовують просторові моделі для структуризації сумарних даних. Атомарні дані залишаються доступними через нормалізоване сховище даних. Очевидно, що структура атомарних і сумарних даних при такому підході

істотно розрізняється. В якості відмітних характеристик підходу Білл Інмона до архітектури сховищ даних можна назвати наступні:

- використання реляційної моделі організації атомарних даних і просторовою - для організації сумарних даних;
- використання ітеративного або «спірального» підходу при створенні великих сховищ даних, тобто «будівництво» сховища не відразу, а по частинах. Це дозволяє при необхідності вносити зміни в невеликі блоки даних або програмних кодів і позбавляє від необхідності перепрограмувати значні об'єми даних в сховищі. Те ж саме можна сказати і про потенційні помилки: вони також будуть локалізовані в межах порівняльного невеликого масиву без ризику зіпсувати все сховище;
- використання третьої нормальної форми для організації атомарних даних, що забезпечує високий ступінь детальної інтегрованих даних і, відповідно, надає корпораціям широкі можливості для маніпулювання ними і зміни формату і способу представлення даних в міру необхідності;
- сховище даних - це проект корпоративного масштабу, що охоплює всі відділи і обслуговує потреби всіх користувачів корпорації.
- сховище даних - це не механічна колекція вітрин даних, а фізично цілісний об'єкт.

Data Warehouse Bus. Рисунок. 2.4. представляє альтернативний підхід до архітектури сховищ даних, відомий як сховище з архітектурою шини або підхід Ральфа Кимболла [62, 71].

У цій моделі первинні дані перетворюються в інформацію, придатну для використання, на етапі підготовки даних. При цьому обов'язково приймаються до уваги вимоги до швидкості обробки інформації і якості даних. Як і в моделі Білла Інмона, підготовка даних починається з скоординованого витягання даних з джерел. Ряд операцій здійснюється централізовано, наприклад, підтримка і зберігання загальних довідкових даних, інші дії можуть бути розподіленими. Область уявлення просторово структурована, при цьому вона може бути централізованою або розподіленою.

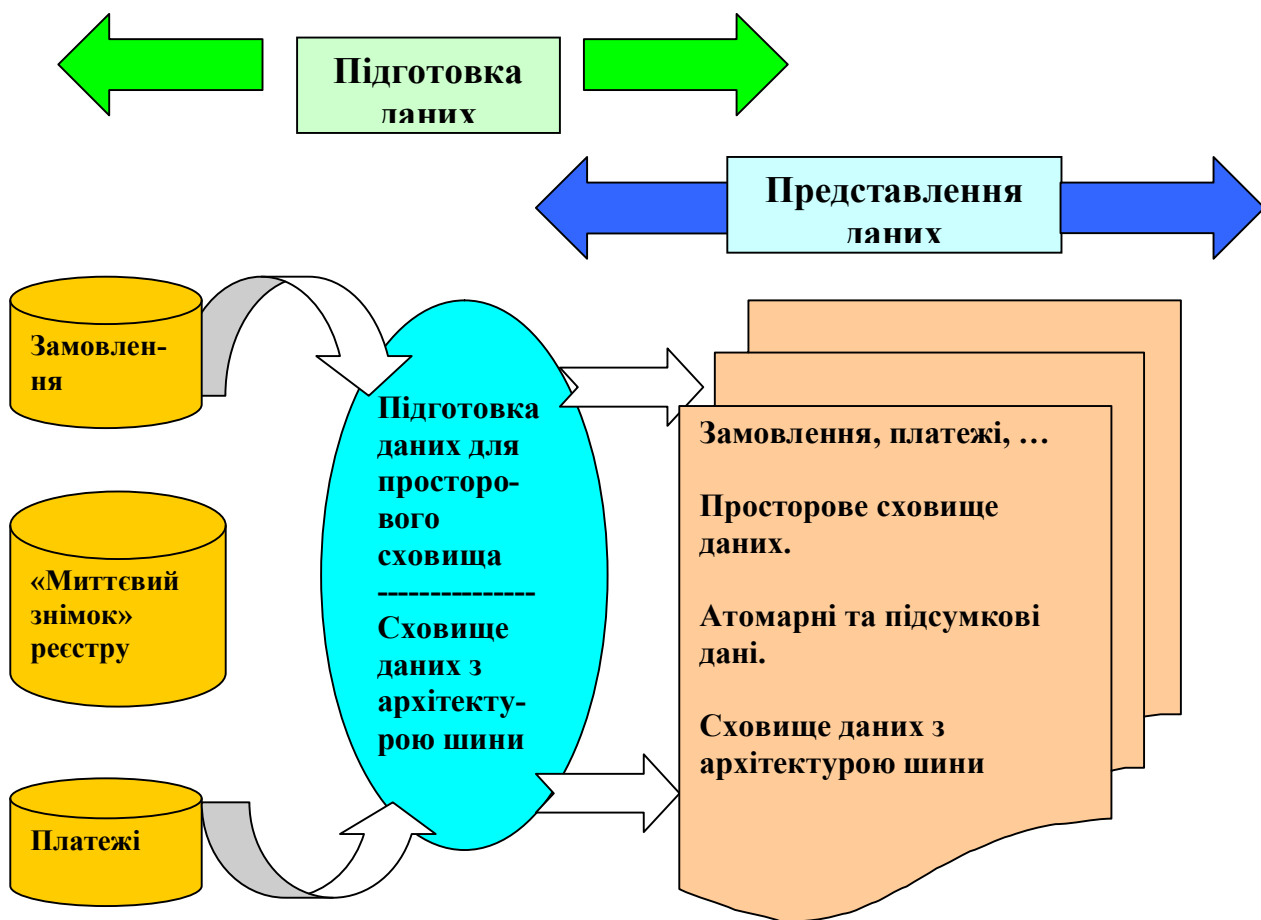


Рис. 2.4. Просторове сховище даних.

Просторова модель сховища даних містить ту ж атомарну інформацію, що і нормалізована модель, але інформація структурована по-іншому, щоб полегшити її використання і виконання запитів. Ця модель включає як атомарні дані, так і загальну інформацію (агрегати в зв'язаних таблицях або багатовимірних кубах) відповідно до вимог продуктивності або просторового розподілу даних. Запити в процесі виконання звертаються до все більш низького рівня деталізації без додаткового перепрограмування з боку користувачів або розробників додатку.

На відміну від підходу Білла Інмона, просторові моделі будуються для обслуговування бізнес-процесів (які, у свою чергу, пов'язані з бізнес-показниками), а не бізнес-відділа. Наприклад, дані про замовлення, які повинні бути доступні для загально корпоративного використання, вносяться до просторового сховища даних тільки один раз, на відміну від CIF-підходу, в

якому їх довелося б тричі копіювати у вітрини даних відділів маркетингу, продажів і фінансів. Після того, як в сховищі з'являється інформація про основні бізнес-процеси, консолідовані просторові моделі можуть видавати їх перехресні характеристики. Матриця корпоративного сховища даних з архітектурою шини виявляє і підсилює зв'язки між показниками бізнес-процесів (фактами) і описовими атрибутами (вимірюваннями).

Підсумовуючи все вищесказане, можна відзначити типові риси підходу Ральфа Кимболла:

- використання просторової моделі організації даних з архітектурою «зірка» (star scheme);
- використання дворівневої архітектури, яка включає стадію підготовки даних, недоступну для кінцевих користувачів, і сховище даних з архітектурою шини як таке. До складу останнього входять декілька вітрин атомарних даних, декілька вітрин агрегованих даних і персональна вітрина даних, але воно не містить одного фізично цілісного або централізованого сховища даних;
- сховище даних з архітектурою шини володіє наступними характеристиками: воно просторове, воно включає як дані про транзакції, так і сумарні дані, воно включає вітрини даних, присвячені тільки одній предметній області або що мають тільки одну таблицю фактів (fact table), воно може містити безліч вітрин даних в межах однієї бази даних;
- сховище даних не є єдиним фізичним репозиторієм (на відміну від підходу Білла Інмона). Це «віртуальне» сховище. Це колекція вітрин даних, кожна з яких має архітектуру типу «зірка».

До теперішнього часу технології сховищ даних досягли рівня зрілості - і як дисципліна, і як ринок технологій. Попит на них високий як ніколи. Про це, наприклад, свідчить той факт, що в 2007 р. дані засоби вперше потрапили в десятку головних пріоритетів директорів по інформаційним технологіям (згідно опиту компанії Gartner Group Inc.). Багато корпорацій вже володіють інфраструктурою сховищ даних і зараз займаються її удосконаленням і вирішенням проблем, з якими не вдалося справитися на стадії впровадження.

Частина корпорацій також прагнуть досягти наступних рівнів в розвитку цих засобів і акцентують увагу на 10 основних тенденціях, які сьогодні впливають на підхід корпорацій до впровадження сховищ даних. Ці тенденції включають як ті поліпшення, які компанії вносять до своєї стратегії і інфраструктури, так і нові технології і ініціативи, сприяючі подальшому розвитку сховищ даних [71].

Тенденція № 1: серйозне відношення до якості даних. Мало хто займатиметься вирішенням проблеми якості даних тільки ради якості як такого. Так що ж примушує корпорації приймати конкретні заходи по поліпшенню якості даних, а не просто говорити про це? По-перше, низька якість даних коштує гроші в тому сенсі, що веде до зниження продуктивності, ухвалення неправильних бізнес-рішень і неможливості отримати бажаний результат, не дивлячись на істотні інвестиції в корпоративні додатки. По-друге, низька якість даних може утруднити виконання вимог законодавства. За прогнозом компанії META Group, щорічне зростання ринку програмного забезпечення і послуг у сфері якості даних складатиме 20-30% аж до 2010 р. Цей прогноз співпадає з вже існуючими тенденціями: компанії дійсно мають намір робити конкретні дії для вирішення проблем якості даних. Враховуючи ту різноманітність засобів, яка зараз існує на ринку, для будь-якої компанії дуже важливо правильно вибрати такі методологію і інструменти, які дозволять практично підійти до вирішення цих проблем.

Тенденція № 2: стандартизація і консолідація інфраструктури. Корпорації стали усвідомлювати проблему роз'єднаності своїх рішень у сфері сховищ даних зовсім недавно, буквально пару років назад. По-перше, виявилось, що усунення дублюючих один одного інструментів сховищ даних або вітрин даних може понизити витрати на ліцензії і підтримку. Крім того, поліпшення доступу до інформації - це також істотна вигода від усунення незалежних структур даних, хоча вона і не піддається прямим кількісним оцінкам. Але здійснити реальну стандартизацію і консолідацію інфраструктури сховищ даних набагато складніше, ніж просто декларувати такий намір. Ця

діяльність включає політичні і організаційні аспекти, які такі ж проблематичні, як і технологічна сторона процесу.

Тенденція № 3: використання зарубіжних трудових ресурсів в області сховищ даних. На відміну від американських компаній, тенденція перенесення виконання проектів в області сховищ даних за межі країни з метою економії засобів мало актуальна для України, де заробітна плата і так менше, ніж в розвинених країнах. Тому тут ми детально не зупинятимемося на цій тенденції, відзначимо лише два основні моменти.

1. Така тенденція дійсно існує в США протягом декількох останніх років і активно обговорюється бізнес-аналітиками.

2. Перші результати цього процесу достатньо суперечливі. Разом з економією засобів компанії стикаються і з труднощами, пов'язаними з якістю роботи, комунікаційними проблемами і т.д. У результаті компанії далеко не завжди отримують ту економію засобів, на яку вони розраховували.

Тенденція № 4: стратегічний підхід до інформації. Поволі, але вірно корпорації починають розглядати інформацію як стратегічну складову бізнесу. Поки мало хто утілює ідею «інформація як капітал» в практику, але в багатьох організаціях вже є співробітники, які усвідомлюють стратегічну цінність інформації. Менеджери вищої ланки також стали прихильно відноситися до цієї ідеї. Доказом даної тенденції є той факт, що сховища даних стають важливою частиною інших проектів з великими перспективами для бізнесу. Компанії можуть не впроваджувати систем інтелектуального аналізу даних в масштабах всієї корпорації, але вони використовують прийоми таких систем і сховища даних в ключових корпоративних проектах, які направлені на оптимізацію бізнес-процесів і зниження витрат.

Отже, з чого ж починається відношення до інформації як до капіталу? Перший крок - це розробка інформаційної стратегії і архітектури, а також впровадження основоположних стандартів управління даними. Це вимагає участі як ІТ-відділу, так і менеджерів, оскільки тут однаково важливі і технологічні, і організаційні і політичні аспекти. ІТ-фахівці повинні працювати

разом з менеджерами, щоб зрозуміти вимоги останніх і з'ясувати, які саме дані потрібні для вирішення основних питань бізнесу.

Тенденція № 5: виконання вимог законодавства як двигун розвитку сховищ даних. Для виконання законодавчих вимог необхідний доступ до даних. Як результат усвідомлення цього, в США зросли інвестиції в рішення, що надають доступ до інформації (або що забезпечують безпеку даних) і гарантують її точність. Необхідність виконання вимог законодавства стала основним стимулом для проектів, пов'язаних з сховищами даних, позначивши реальні негативні наслідки відсутності доступу до якісних даних. Багато корпорацій можуть випробувати спокусу проігнорувати сховища даних і звернутися до своїх систем планування ресурсів підприємства (Enterprise Resource Planning - ERP) і фінансових систем, але прозорість і можливість контролю бізнесу, які надають рішення систем аналізу даних і сховищ даних, залишаються поза конкуренцією.

Тенденція № 6: подальший розвиток проблеми інтеграції корпоративних даних. Для передових корпорацій вже не стоїть питання, які технології вибрати для інтеграції даних, - вони зробили свій вибір. Багато хто вибрав технології витягання, перетворення і завантаження даних (Extraction, Transmission, Loading - ETL) як можливий спосіб для інтеграції корпоративних даних. Тепер перед цими корпораціями встає інше питання: як використовувати ті інструменти ETL, які вони вибрали. Таким чином, проблема інтеграції даних піднімається на наступний рівень. Для повноцінної інтеграції даних організаціям необхідно здійснювати управління метаданими і довідковими даними, а також забезпечити якість даних. Фактично, їм потрібний ще один інтегруючий засіб, який може зв'язати воедино вибрані ними інструменти і збільшити вартість кінцевої продукції компанії. В той же час, багато корпорацій все ще знаходяться в процесі пошуку відповідних інструментів для інтеграції даних, вибираючи між ETL-технологіями, інтеграцією корпоративних додатків (Enterprise Application Integration - EAI) і web-сервісами. Але треба мати на увазі, що межі між цими технологіями стають все

більш розпливчатими: у ETL-технологіях з'являються можливості інструментів EAI для роботи в режимі реального часу, а в засобах EAI розширюються можливості для перетворення даних. Навіть для тих корпорацій, які все ще не визначилися з вибором інтеграційних технологій, буде корисно заглянути вперед і подумати про ті кроки, які доведеться зробити для впровадження вибраної технології і її успішного використання.

Тенденція № 7: навчання кінцевих користувачів. Рішення сховищ даних виявляться незатребуваними, якщо кінцевим користувачам не пояснити, які дані тепер їм доступні, і не переконати їх, що ці дані точні і повноцінні. Багато організацій вважають, що буде досить навчити кінцевих користувачів загальним прийомам роботи з цими інструментами. Але, на думку відомої аналітичної корпорації Gartner Group Inc., «набагато важливіше навчити користувачів інтелектуальному аналізу даних». Користувач, навчений роботі з інструментами, але що не знає, як їх використовувати в контексті конкретного середовища сховища даних, не зможе отримати бажані аналітичні результати. Відповідно, такий користувач або звернеться в IT-відділ, або взагалі відмовиться від використання даних засобів. Недостатнє визнання користувачами і усвідомлення компаніями цінності попередніх проектів сховищ даних примушують багато організацій визнати значущість комплексних навчальних програм для кінцевих користувачів.

Тенденція № 8: управління довідковими даними. У кожній корпорації є набір даних, які забезпечують важливу інформацію, необхідну для ідентифікації і конкретного визначення ключових об'єктів, таких як споживачі, продукція, постачальники і т.д. Ці дані називаються довідковими даними (master data), і зараз вони стають головною турботою для все більшого числа організацій. Управління довідковими даними, на перший погляд, виглядає простим завданням. Дійсно, що може бути складного для організації ідентифікувати своїх споживачів або визначити кожен вид своєї продукції? Як і у багатьох випадках, що відносяться до сфери інтелектуального аналізу даних і сховищ даних, те, що здається зовсім простим, в реальності може виявитися

вельми складним. Розповсюдження корпоративних додатків укупі з переважанням в багатьох організаціях ізольованих структур даних призводить до того, що довідкові дані виявляються розсіяними по всій корпорації. Різні області бізнесу можуть по-різному визначати і ідентифікувати такі об'єкти, як «споживач» або «продукт», і, можливо, зберігати їх довідкові дані в роз'єднаних базах даних. Таким чином, тенденція до інтеграції і оптимальної організації корпоративних систем зробила управління довідковими даними пріоритетним завданням. Але треба мати на увазі, що будь-яке технічне рішення, яке буде вибрано для цього завдання, повинне не тільки включати конкретні управлінські інструменти, але і приймати до уваги організаційні і політичні аспекти, якимось: хто є «власником» цих довідкових даних і хто повинен визначати їх.

Тенденція № 9: поява нових учасників ринку сховищ даних. Зростання ринку сховищ даних не пройшло непоміченим серед постачальників в інших областях технології. Серед нових учасників цього ринку є як невеликі компанії, що починають, так і вже достатньо відомі фірми, але найбільш могутні з них - це постачальники систем планування ресурсів підприємства (ERP системи) і систем управління відносинами з клієнтами (Customer Relationship Management - CRM), наприклад, такі американські компанії, як SAP, Oracle і Siebel. Ці компанії бачать великі можливості на ринку сховищ даних, по-перше, оскільки неухильно зростає оборот цього ринку, а по-друге, тому що багато корпорацій виявили, що їх ERP і CRM системи функціонують не зовсім так, як очікувалося, в сенсі можливостей доступу до даних і їх аналізу. Постачальники ERP і CRM систем приходять на цей ринок, пропонуючи своїм клієнтам засоби інтелектуального аналізу даних і сховищ даних, щоб допомогти їм в отриманні необхідних даних, і статистика показує, що ці постачальники досягають успіху на новому терені.

Тенденція № 10: використання сховищ даних як робочого інструменту. Компанії усвідомлюють стратегічну роль сховищ даних, але вони також хочуть використовувати інформацію, отриману за допомогою своїх даних, для того,

щоб ухвалювати і тактичні рішення. Наприклад, роздрібні компанії зацікавлені в тому, щоб зрозуміти, як вони можуть використовувати інформацію, отриману з ланцюга постачань, для ухвалення своєчасних рішень. Таким чином, сховища даних можуть зіграти роль як в оперативному функціонуванні корпорації, так і у визначенні стратегічного напрямку її розвитку.

2.3. Вітрини та кіоски даних

У найбільш загальному виді сховища даних можуть бути розбиті на два типи: корпоративні сховища даних (Enterprise Data Warehouses) і кіоски або вітрини даних (Data Marts) [9, 37, 49, 51].

Корпоративні сховища даних містять інформацію, що відноситься до всієї корпорації і зібрану з безлічі оперативних джерел для консолідованого аналізу. Зазвичай такі сховища охоплюють цілий ряд аспектів діяльності корпорації і використовуються для ухвалення як тактичних, так і стратегічних рішень. Корпоративне сховище містить детальну і узагальнену інформацію, його об'єм може досягати від 50 Гбайт до одного або декількох терабайт. Вартість створення і підтримки корпоративних сховищ може бути дуже високою. Зазвичай їх створенням займаються централізовані відділи інформаційних технологій, причому створюються вони зверху вниз, тобто спочатку проектується загальна схема, і тільки тоді починається заповнення даними. Такий процес може займати декілька років.

Кіоски або вітрини даних містять підмножину корпоративних даних і будуються для відділів або підрозділів усередині організації (рис. 2.5). Кіоски даних часто будуються силами самого відділу і охоплюють конкретний аспект, що цікавить співробітників даного відділу. Кіоск даних може отримувати дані з корпоративного сховища (залежний кіоск) або, що поширеніше, дані можуть поступати безпосередньо з оперативних джерел (незалежний кіоск). Кіоски і

сховища даних будуються за схожими принципами і використовують практично одні і ті ж технології.

Багато компаній, що усвідомлюють необхідність розробки корпоративного сховища даних, все ж таки не в силах справитися зі всіма завданнями виділення, стандартизації і об'єднання терабайт даних. Натомість вони вважають за краще будувати кіоски (або вітрини) даних (Data Marts) - спеціалізовані сховища даних, присвячені тільки одному напрямку діяльності організації. Кіоск (вітрина) даних - це, найчастіше, найбільш керований різновид сховища даних. Його безперечний недолік полягає в тому, що без сховища даних, яке охоплювало б інформацію всього підприємства, неможливо порівнювати і аналізувати дані по всіх відділах і процесах. У багатьох компаніях вже зрозуміли, що кіоски (вітрини) даних можуть послужити хорошу службу і навіть стати єдиною можливим рішенням для виконання термінових аналітичних завдань, але створення спеціалізованих кіосків без попередньої розробки корпоративної інфраструктури сховища даних може згодом привести до великих утруднень.

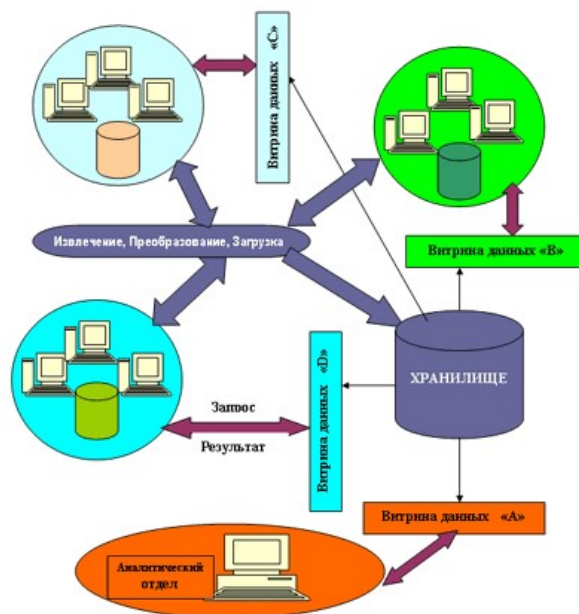


Рис. 2.5. Кіоски (вітрини) даних організації.

За класичним визначенням, вітрина (кіоск) даних (Data Mart) є підмножиною сховища даних, що відображає специфіку підрозділу (бізнес-об'єкт) і що забезпечує підвищену продуктивність. Таким чином, вітрина є ланкою, на якій базується конкретна аналітична система для вирішення свого кола завдань. Проте можлива ситуація, коли деяка область діяльності підприємства практично не корелює з іншими, і можливо побудувати відповідну вітрину даних автономно, без прив'язки до корпоративного сховища. Тоді вітрина поповнюватиметься даними безпосередньо з оперативних систем обробки транзакцій. Такі вітрини даних отримали назву незалежних, на відміну від класичних залежних від сховища даних і поповнюваних з нього вітрин.

У ряді випадків представляється доцільним розвернути вітрину (кіоск) даних замість повністю сформованого сховища. Вітрини даних накладають менші зобов'язання, вони дешевше і простіше в побудові і базуються на дешевших серверах, а не на мультипроцесорних комплексах. При такому підході немає необхідності задіювати цілу інформаційну систему корпорації і підтримувати складні процедури синхронного оновлення вітрини даних при оновленні сховища. В той же час необхідний розуміти, що при такому підході вітрини даних можуть розмножитися в цілі комплекси незалежних інформаційних баз даних, і природно буде поставлено завдання управління індивідуальними стратегіями пошуку, обслуговування і відновлення. З іншого боку, будувати єдине корпоративне сховище на основі множини незалежних вітрин даних значно вигідніше, ніж спираючись на розсіяні по системах обробки транзакцій дані.

Так що ж доцільно застосовувати: єдине сховище, самостійні вітрини (кіоски) даних, сховище із залежними вітринами або інші варіанти? Універсальної відповіді на питання про необхідність застосування того або іншого варіанту не існує. В кожному випадку оптимальний варіант визначається вимогами бізнесу, інтенсивністю запитів, мережевою архітектурою, необхідною швидкістю реакції і іншими умовами.

Розглянемо аргументи «За» і «Проти» використання вітрин (кіосків) даних відносно використання сховищ даних (таб. 2.2).

Таблиця 2.2.

Порівняльні аргументи застосування вітрин (кіосків) та сховищ даних.

	Аргументи «За»
Вартість	Створення навіть декількох вітрин (кіосків) даних обходиться значно дешевше, ніж організація єдиного сховища даних
Терміни	На опис предметної області, взаємних зв'язків між даними, організацію сховища даних і розробку механізмів його поповнення може піти декілька років, тоді як опис якого-небудь одного напрямку діяльності підприємства без урахування різних зв'язків і з невеликим числом джерел надходження інформації займе менше часу.
Розміри	Оскільки вітрини (кіоски) даних зазвичай містять лише дані по певному колу питань і, отже, займають менше місця і вимагають менше технічних ресурсів, то для них менш гостро коштує питання апаратної платформи і вартості устаткування.
Безпека	З вітринами (кіосками) даних зазвичай працює менше число користувачів, чим зі сховищем даних. З'являється можливість контролю прав не тільки на рівні окремих таблиць і записів, а і на рівні доступу до всього додатку, що надійніше.
	Аргументи «Проти»
Дублювання даних	Різні вітрини даних можуть містити одну і ту ж інформацію, якщо цього вимагають їх предметні області. Природно, дублювання інформації ставить перед користувачами і адміністраторами проблему синхронізації даних (тобто їх порівняння і уніфікації).
Розширення	Хоча виробники і стверджують, що вітрини (кіоски) даних поступово можна наростити до рівня сховищ даних (як модульні системи), в це погано віриться. Інакше вони б

	коштували стільки ж, скільки і самі сховища. Представляється, що процес об'єднання незалежних (логічно і фізично) вітрин даних вельми трудомісткий.
Обмеженість	Вітрини (кіоски) даних задумані як склади даних, що містять інформацію по якій-небудь одній темі. Для великих компаній з широким колом вирішуваних задач і різноманітними інтересами, вигідніше мати повноцінне сховище даних, оскільки воно зможе вміщати всі необхідні для їх життєдіяльності відомості.

Розповідаючи про триваючі протягом півроку дослідження (з питання про переважний вибір між кіосками даних і сховищами даних), аналітик Кевін Стрендж, керівник досліджень по програмних структурах компанії Gartner Group, відзначив, що кіоски не обов'язково мають малий розмір, деколи вони вимагають рішення складних адміністративних задач, і причому далеко не завжди обходяться дешевше за сховища даних [37, 51]. Кіоски (вітрини) даних створюються спеціально під конкретний програмний додаток і вирішують специфічні задачі, скажемо проводять аналіз, необхідний для роботи служби технічної підтримки користувачів, тоді як крупні сховища даних не залежать від додатку. Але кіоски даних можуть зберігати сотні гігабайт інформації і не обмежуються загальним стандартом в 50 Гбайт.

Кіоски добре підходять компаніям, які вимушені підтримувати системи підтримки ухвалення рішення і не володіють достатнім досвідом для розробки повномасштабного сховища даних. По словах Стренджа, організації, що вже реалізували кіоски даних, поступово починають об'єднувати їх, і без сумніву до 2010 року більшість організацій, які зараз створюють кіоски, прагнуть групувати їх в крупніші структури.

Компанія Gartner Group також прийшла до висновку, що орієнтація сховищ даних на Web обмежувалася додатками для Intranet і не була

розрахована на використання Internet із-за уразливості захисту і недостатнього рівня продуктивності.

Прийоми моделювання кіосків (вітрин) даних відрізняються від прийомів моделювання сховищ даних через різні вимоги до структур даних. Якщо основною задачею сховища даних є зберігання консолідованої історичної інформації, то вітрина даних будується з урахуванням вимог по доступу до даних і представлення інформації. Як правило, для моделювання вітрин (кіосків) даних використовуються типи моделі під назвою: схема "зірка" і схема "сніжинка". Зупинимося докладніше на кожному з цих типів моделей.

Схема "зірка" - популярний тип моделі даних для вітрин даних. Дана модель характеризується наявністю таблиці фактів, оточеної пов'язаними з нею таблицями розмірностей. Запити до такої структури включають прості об'єднання таблиці фактів з кожною з таблиць розмірностей. Характеризується високою продуктивністю запитів. Проектується для виконання аналітичних запитів. Характеризується невеликою надмірністю даних і високою в порівнянні з нормалізованими структурами продуктивністю. Деякі промислові СУБД і інструменти класу OLAP/Reporting уміють використовувати переваги схеми "зірка" для скорочення часу виконання запитів. На рисунку 2.6 зображений приклад схеми "зірка" для аналізу залишків на рахунках і кількості транзакцій в розрізі часу, клієнтів, рахунків, продуктів, відділень, статусів рахунків. Дана модель дозволяє відповісти на широкий спектр аналітичних питань.

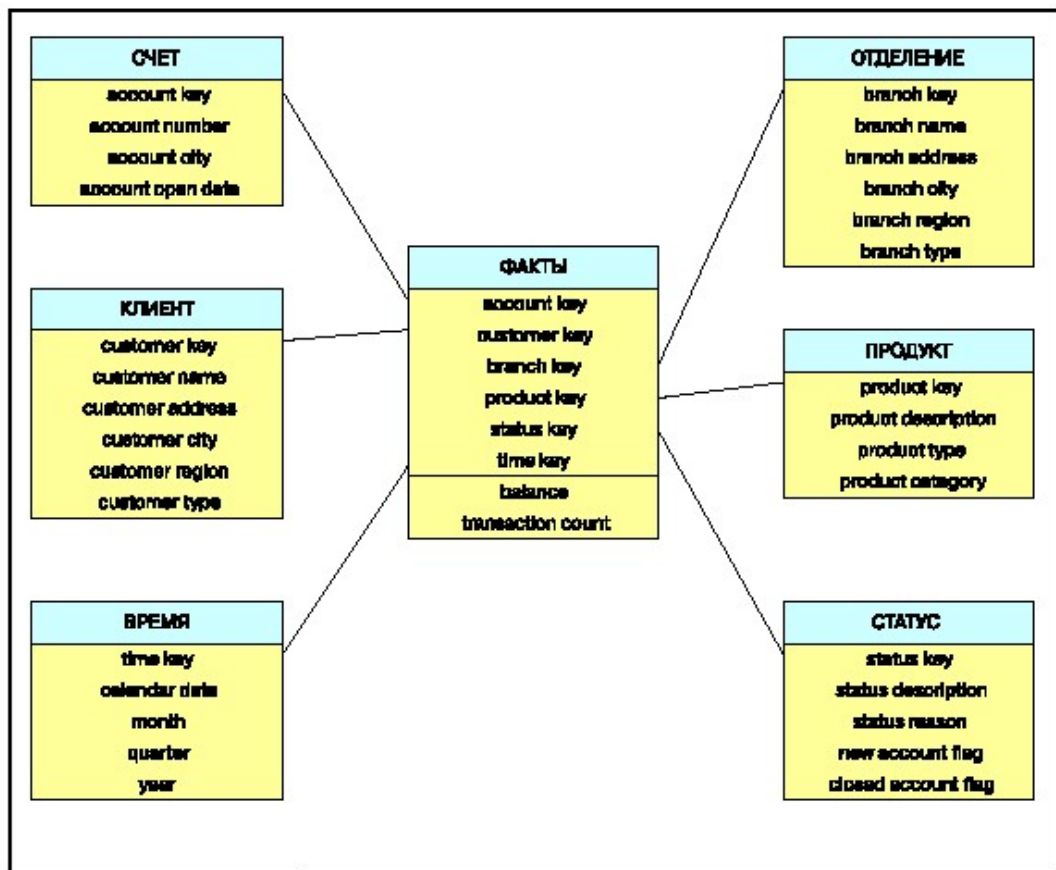


Рис. 2.6. Приклад схеми "зірка" для аналізу залишків на рахунках.

Розглянемо компоненти схеми "зірка".

Розмірності. У технології багатовимірного моделювання розмірність - це аспект, в розрізі якого можна отримувати, фільтрувати, групувати і відображати інформацію про факти. Типові розмірності, що зустрічаються практично в будь-якій моделі:

- Клієнт
- Продукт
- Час
- Географія
- Співробітник

Розмірності, як правило, мають багаторівневу ієрархічну структуру. Наприклад, розмірність ЧАС може мати наступну структуру: РІК – КВАРТАЛ - МІСЯЦЬ - ДЕНЬ.

Факти. Факти - це зазвичай числові величини, що зберігаються в таблиці фактів і є предметом аналізу. Приклади фактів: об'єм операцій, кількість проданих одиниць товару і так далі. Факти мають ряд властивостей, на яких ми коротко зупинимося.

1. Адитивні факти. Адитивність визначає можливість підсумування факту уздовж певної розмірності. Такі факти можна підсумовувати і групувати уздовж всієї розмірності на будь-яких рівнях ієрархії.

2. Напівадитивні факти. Напівадитивний факт — це факт, який можна підсумовувати уздовж певної розмірності, і не можна — уздовж інших. Прикладом може служити залишок на рахунку (або залишок товару на складі). Дану величину не можна підсумовувати уздовж розмірності ЧАС. Проте сума залишків по рахунках уздовж розмірності є предметом для аналізу.

Фахівці рекомендують моделювати напівадитивні факти так, щоб зробити їх більш адитивними. Наприклад, представити відсоток складовими його величинами.

3. Неадитивні факти. Неадитивні факти взагалі не можна підсумовувати. Приклад неадитивного факту — відношення (наприклад, виражене у відсотках).

Таблиці покриття. Таблиці покриття використовуються з метою моделювання поєднання розмірностей, для яких відсутні факти. Наприклад, потрібно знайти кількість категорій продуктів, які сьогодні жодного разу не продавалися. Таблиця фактів продажів не може відповісти на дане питання, оскільки вона реєструє лише факти продажів. Для того, щоб модель дозволяла відповідати на подібні питання, потрібна додаткова таблиця фактів (яка, по суті справи, не містить фактів).

Схема "сніжинка" використовується для нормалізації схеми "зірка" (рис. 2.7). Вона декілька скорочує надмірність в таблицях розмірності. Одним з достоїнств є швидше виконання запитів про структуру розмірності (запити вигляду «Вибрати всі рядки з таблиці розмірності на певному рівні»), які дуже часто виконуються при аналізі даних, і можуть затримувати хід аналізу. Проте основною відмінністю схеми "сніжинка" є не економія дискового простору, а

можливість мати таблиці фактів з різним рівнем деталізації. Наприклад, фактичні дані на рівні дня, а планові — на рівні місяця.

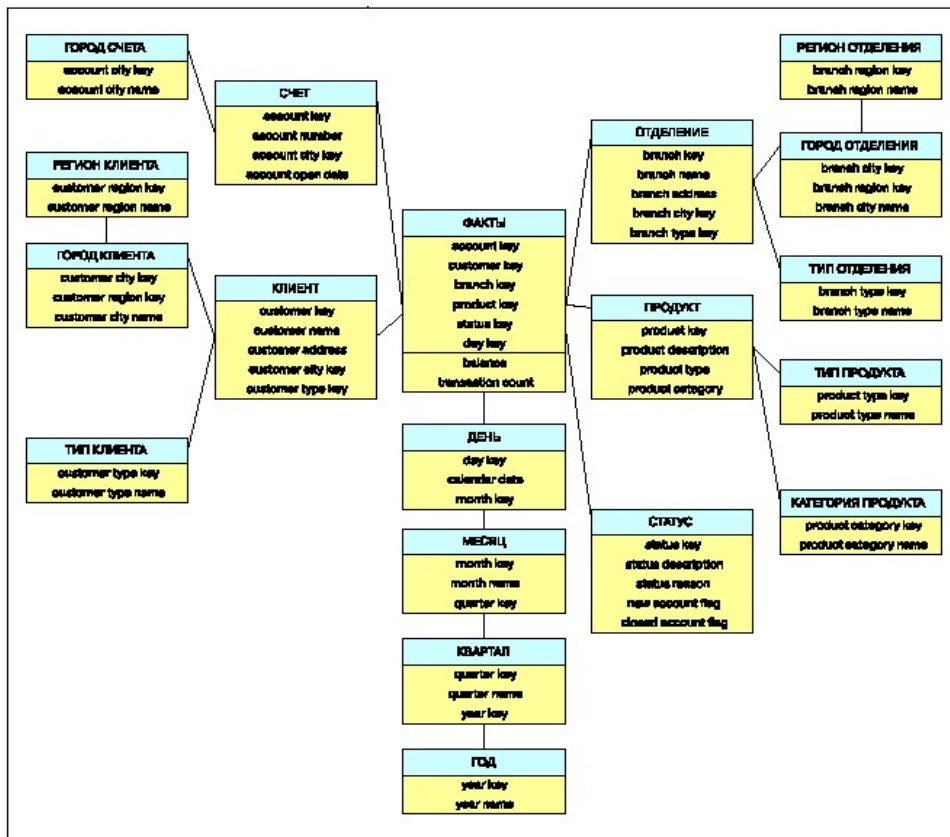


Рис. 2.7. Приклад схеми "сніжинка" для аналізу залишків на рахунках.

Методика побудови вітрин (кіосків) даних з простої теоретичної дисципліни поступово перетворюється на складну науку, повну варіацій і напрямів. Якщо раніше було відомо лише про EDW (Enterprise Data Warehouse), то тепер з'явилися поступово розвиваємі вітрини даних (Incremental Architected Data Mart, ADM), розподілені вітрини (кіоски) даних (Distributed Data Mart, DDM), об'єднані вітрини даних (Federated Data Mart, FDM). Розглянемо деякі з цих нових напрямків.

Системи об'єднаних вітрин даних. У багатьох організаціях склалася практика реалізації багаточисельних сховищ даних. Хоча, за визначенням, існує лише одне сховище даних, а всі останні об'єкти є його підмножиною або вітринами (кіосками) даних, що поступово розвиваються, але не всі організації дотримуються цього правила. Таким чином, в багатьох компаніях існує два,

три, десяток і навіть більше систем сховищ даних. Поширення сховищ даних привело до розвитку архітектури сховища даних підприємства, а саме: до появи об'єднаних систем сховищ даних або вітрин (кіосків) даних.

Система об'єднаних вітрин даних характеризується спільним використанням загальних інформаційних ресурсів, усуваючи, таким чином, надмірність і гарантуючи достовірність інформації по всій організації (рис. 2.8).

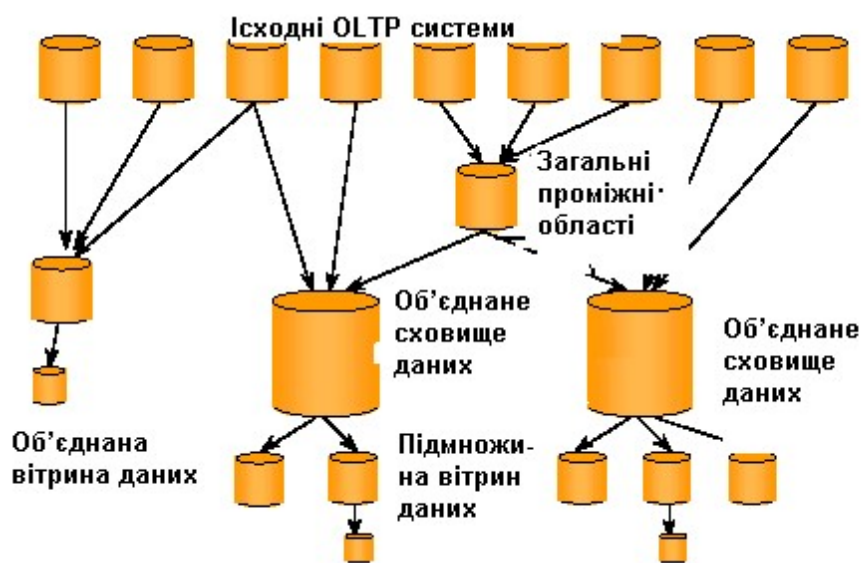


Рис. 2.8. Система об'єднаних вітрин даних.

Позитивними рисами об'єднаних вітрин даних є: загальна семантика бізнес-правила; один набір процесів витягання і бізнес-правил; децентралізовані ресурси і управління; паралельна розробка.

Недоліками такого архітектурного рішення є: необхідність в координуванні робіт; складнощі в подоланні "політичних" моментів і вирішенні питань авторських прав; потрібна узгодженість серед різних відділів по питаннях архітектури, бізнес-правил і семантики; складне технічне середовище; наявність багаточисельних репозиторіїв метаданих.

Непроектуємі вітрини даних. Поява непроектуємих вітрин даних (Non-Architected Data Marts) пояснюється, перш за все, складнощами, пов'язаними з реалізацією систем EDW і FDW. Брудні і швидко отримувані набори даних не піддаються очищенню і, отже, не можуть використовуватися для подальшої

інтеграції з будь-якими іншими джерелами даних систем сховищ даних. Дуже швидко вони перетворюються на застарілі системи, які лише додають проблем, а не вирішують їх. Для цих систем характерні багаточисельні процеси витягання, безліч бізнес-правил, невірогідність інформації (рис. 2.9).

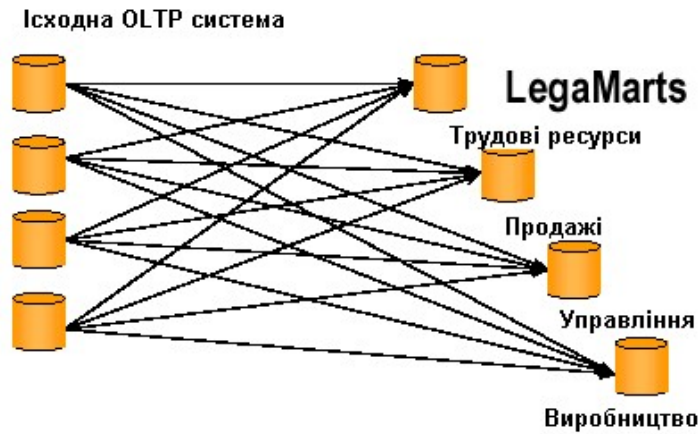


Рис. 2.9. Система непроєктуємих вітрин даних (LegaMart).

Позитивними рисами непроєктуємих вітрин даних є: висока продуктивність; низька вартість. Недоліками таких систем є: недостовірна інформація; багаточисельні процеси витягання; багаточисельні бізнес-правила; підвищена складність при інтеграції.

Система вітрин (кіосків) даних, що поступово розвиваються. Ця архітектура є альтернативою сховища даних підприємства. Для наповнення таких вітрин зазвичай використовується інструментальний засіб класу підприємства, що реалізує стратегію «витягаєш один раз, наповнюєш багато» (рис. 2.10).

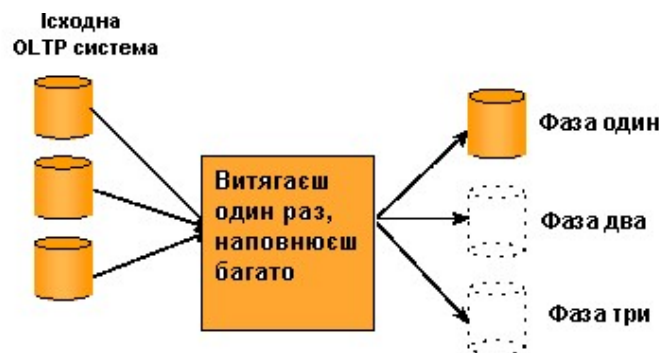


Рис. 2.10. Система поступово розвиваємих вітрин даних.

Достоїнствами таких вітрин даних є: єдиний набір процесів витягання; здійснимий масштаб. Недоліки: найбільш ефективні при використанні інструментального засобу класу підприємства; необхідність в архітектурі вітрин даних підприємства (Enterprise Data Mart Architecture, EDMA).

Методика побудови вітрин (кіосків) даних виявилася напрямом ринку проєктів інтелектуального аналізу даних, що нестримно розвивається, швидко змінюється. Якщо раніше не було механізмів їх ефективного проєктування, і був лише один спосіб їх створення, в даний час можна знайти незчисленне число таких інструментів і ряд технологій життєздатної архітектури таких систем. За умови вибору відповідної архітектури і належного підходу до проєкту можна побудувати систему сховища та вітрин даних, яка забезпечить не лише високе повернення інвестицій, але і значно підвищить ефективність функціонування всього підприємства [37, 70].

Декілька фірм пропонує системи побудови вітрин даних: Informatica (PowerMart Suite), Sagent Technology (Data Mart Solution) і Oracle (DataMart Suite). Для ілюстрації процесу розробки вітрини даних можна розглянути коротко склад і функціональність пакету DataMart Suite [43].

Пакет включає п'ять основних компонентів: Data Mart Designer, Data Mart Builder, Oracle7 Enterprise Server, Web Server і Discoverer 3.0. Data Mart Designer дозволяє описувати структуру вітрини і запам'ятовувати її в репозитарії. На виході Data Mart Designer породжує опис на мові DDL SQL, який потім подається на вхід Oracle7 Enterprise Server. В результаті створюється структура бази даних, що реалізовує вітрину даних. В ході побудови вітрини користувач може застосовувати існуючі описи структур або будувати вітрину «з нуля». Крім того, Data Mart Designer дозволяє будувати додатки для Oracle Web Server на базі PL/SQL. Data Mart Builder витягує дані із зовнішніх джерел і заповнює вітрину. Він володіє наочним спеціалізованим інтерфейсом, що відображає потоки даних при заповненні сховища. Data Mart Builder здатний витягувати дані з реляційних СУБД і CSV-файлів. Web Server

надає відкриту платформу для розробки Web-додатків. Він включає Web Request Broker (WRB), що реалізований на основі технології картриджів і дозволяє розробляти Web-додатки, що вбудовуються в Web Server. Як засоби розробки можуть використовуватися Java, PL/SQL, LiveHTML, C і C++. Discoverer 3.0 - це засіб кінцевого користувача, що дозволяє генерувати звіти, а також виконувати деякі OLAP-операції з вітриною даних. Звіти, побудовані за допомогою Discoverer 3.0, можна експортувати у форматі HTML, роблячи їх доступними для web-браузерів. Discoverer 3.0 також дозволяє створювати і підтримувати таблиці агрегованих даних. Крім цього, DataMart Suite включає готовий додаток Sales Analyzer.

2.4. OLAP - технологія

Вміння швидко і головне правильно приймати рішення має в сучасному бізнесі принципове значення для досягнення успіху. Проте кількість інформації, що впливає на предмет рішення, інколи може бути просто-таки величезною. Як поступати в такій ситуації? Покласти на випадок або все ж таки взятися за повномасштабний аналіз? Досвідчений керівник незмінно вибере другий спосіб, тим більше що сьогодні існує ряд технологій, здатних спростити процес прийняття і моделювання рішень при великій кількості «вхідної» інформації [4, 9, 11, 66].

Власне, аби спростити роботу з багатоцільовими даними і не загрузнути в їх океані, а також уміло перетворити набір кількісних показників на якісні, і застосовується метод OLAP – On-Line Analytical Processing (оперативна аналітична обробка). Останній, на відміну від інших способів автоматизації бізнес-діяльності, дає можливість отримати користувачеві «на виході» не готове чітко структуроване рішення, що видається після включення раніше налагодженого майстра обробки форм, а своєрідний матеріал для наочної і, якщо можна так виразитися, творчої оцінки існуючої ситуації. Тому сфера

вживання OLAP-аналізу зазвичай обмежується менеджерським складом підприємств різних розмірів, якому доводиться часто займатися тактичними і стратегічними завданнями на зразок аналізу ключових показників діяльності і сценаріїв розвитку, маркетинговим і фінансово-економічним аналізом груп товарів або послуг, а також довгостроковим прогнозуванням роботи підприємства або його підрозділів. Для цього користувач OLAP-систем отримує в руки потужний і головне дуже гнучкий інструмент створення різних звітів по вибраних їм же розрізах і напрямках. При цьому методики OLAP більш досконаліша за звичні електронні таблиці, адже окрім простих функцій створення таблиць, графіків і діаграм, OLAP-системи дають можливість отримати узагальнені дані по самостійно вибраних критеріях, вмить поглибитися в деталі вибраних напрямів, відфільтрувати, сортувати або відкинути непотрібні цифри або показники. Наприклад, якщо менеджерів продажів компанії потрібно отримати сезонні зведення динаміки продажів вибраній категорії товарів, система запропонує йому всілякі дані про продажі за місяць, квартал, рік, а також знайде і проаналізує їх залежність від зазначених чинників, скажімо, часу проведення маркетингових акцій. Крім того, базуючись на одній лише статистиці продажів, OLAP-система може виявити ефективність роботи різних підрозділів компанії, у тому числі і в розрізі географічної ієрархії їх взаємодії. При цьому параметри, що характеризують успішність підрозділів, вибираються менеджером самостійно і у ряді випадків можуть стати інструментом мотивації успішного персоналу.

Щоб зрозуміти, як працюють OLAP-системи, досить звернутися до її механізмів. Найбільш показове поняття OLAP-технології - гіперкуб (метакуб), що є уможливною фігурою в багатовимірному просторі, утвореному площинами даних, які важливі для діяльності підприємства. При цьому сама OLAP-система виступає саме в ролі гіперкуба, здатного накопичувати в собі всю інформацію, що цікавить керівника. Як ребра куба виступають різні категорії товарів або послуг просуваємих компанією. Наприклад, ціна виробленого або конкурентного товару, компанії-учасники виробничого циклу, підрядчики при

організації послуг, об'єми продажів, географія самої компанії. Важливо відзначити, що градація різних осей квадрантів куба може мати різну структуру, а крім того, самі осі можуть бути взаємозалежними. Так, вісь часу може бути розбита по роках, кварталах, тижнях, а вісь доходів або шкідливих викидів при виробництві - прологарифмована. Інформація, необхідна для аналізу в даний момент, вирізається з гіперкуба перетином площин даних, що використовуються при аналізі, немов його шар або частина. При цьому розрізи можуть проходити як через весь куб, так і обмежуватися певними рамками і межами осей.

Технічно системи оперативного аналізу даних зазвичай функціонують у зв'язці з сховищами та вітринами даних, а клієнтські OLAP-системи встановлюються на будь-яких призначених для користувача комп'ютерах корпоративної інформаційної системи. Рідше OLAP-модулі взаємодіють з іншими системами автоматизації, адже бази даних останніх досить часто мають вельми своєрідний вигляд і набір спеціальних показників. Втім, для сучасного українського підприємства характерна нетипова ситуація, коли є декілька систем автоматизації (для вирішення різних завдань) і, як наслідок, дані зберігаються розрізнено, в результаті відсутній єдиний погляд на управлінську інформацію. Тому в процесі складання звіту беруть участь два фахівці – програміст, що забезпечує запити до баз даних, і економіст, що намагається за допомогою електронних таблиць звести ці дані в звіт, необхідний керівництву. Як показує практика, подібна модель взаємодії користувача звіту (керівника) і самих даних незмінно приводить до ефекту «зіпсованого телефону», не говорячи вже про істотні витрати часу. І з даної точки зору використання OLAP-систем також представляється вельми раціональним, адже використання декількох інформаційних систем незмінно приводить до «надлишку» даних, які можуть бути впорядковані OLAP-системою.

У чому ж відмінність OLAP-системи від сховища даних? З точки зору користувача, відповідь на це питання досить проста: у мірі предметної структурованості інформації. Працюючи з OLAP-додатком, користувач

застосовує звичні економічні категорії і показники – види матеріалів і готової продукції, регіони продажів, об'єм реалізації, собівартість, прибуток і тому подібне. А для того, щоб сформулювати будь-який, навіть досить складний запит, користувачеві не доведеться вивчати SQL. При цьому відповідь на запит буде отримана протягом всього декількох секунд. Крім того, працюючи з OLAP-системою, економіст може користуватися такими звичними для себе інструментами, як електронні таблиці або спеціальні засоби побудови звітів. Таким чином, якщо сховище даних – в основному об'єкт уваги спеціаліста по інформаційним технологіям, то OLAP без перебільшення можна назвати програмним засобом з арсеналу економіста. Адже саме економіст має справу з самими різними аналітичними задачами: маркетинговим аналізом, аналізом продажів, аналізом бюджетних показників, аналізом фінансової звітності і так далі [17, 49, 71].

Цікаво, що розшифровка аббревіатури OLAP – «система оперативного аналізу даних» – не повністю відповідає реальному призначенню і методу роботи цих систем. Отримана назва є наслідком того, що OLAP-модулі оперують багатовимірними моделями, якими, до речі, мислить і людина. Тому OLAP, так само як і людський мозок, вимагає вичерпної інформації по самих різних напрямках і навряд чи розкриє весь свій потенціал при недоліку даних. В будь-якому разі «океан даних» якраз те середовище, де використання інструментів OLAP найбільш виправдано. Слід розуміти, що OLAP - це не окремо взятий програмний продукт, не мова програмування і навіть не конкретна технологія. Якщо постаратися охопити OLAP у всіх його проявах, то можна стверджувати, що це є сукупність концепцій, принципів і вимог, які лежать в основі програмних продуктів, що полегшують аналітикам доступ до даних.

Розглянемо деякі сценарії практичного застосування OLAP-продуктів.

Аналіз фінансових показників діяльності підприємства. Бухгалтерські системи 1С, БЕСТ, Парус, Інфін, RS-Balance та інші день за днем нагромаджують результати обліку господарської діяльності підприємств. Вони

забезпечують розрахунок фінансових показників і випуск звітності для наглядових органів. Проте, фінансова звітність не призначена для управління організацією. Керівника цікавить динаміка залишків і рух фінансів, структура доходів і їх розподіл по клієнтах, товарах, днях тижня, місяцях, кварталах, за рік і так далі. Аби забезпечити керівників управлінськими звітами, OLAP-система налаштовується на базу даних будь-якої облікової системи. У додатку описуються запити на здобуття і розрахунок нових фінансових показників, таблиці і графіки звітів для різних користувачів: фінансових директорів, співробітників планово-економічних служб, бухгалтерів. Важливий поточний звіт можна роздрукувати для обговорення з товаришами по службі або зберегти у вигляді мікрокуба і відіслати директорові або інвесторам. В результаті дані фінансового обліку реально використовуються для управління організацією. Вони миттєво витягуються з бази даних бухгалтерської системи, користувач самостійно управляє формою звіту, отримує фінансові показники в різних розрізах.

Корпоративна звітність. У розподіленій організації філії регулярно передають дані в центральний офіс. Тут дані потрапляють в єдине сховище. Над ними виконуються додаткові розрахунки, для яких в філіях немає даних. Наприклад, загально корпоративні витрати розносяться на філії, зменшуючи тим самим їх прибуток в звіті про прибутки і збитки. Аби філії могли ознайомитися з остаточними звітами після виконання всіх розрахунків і перевірок, запускається генератор кубів, що є програмним модулем OLAP-системи. В результаті для кожної філії генерується окремий мікрокуб, який відправляється по e-mail. Одержувач - співробітник планово-економічного відділу філії - відкриває куб в OLAPBrowser, аналізує, роздруковує і підшиває звіти. Для керівника філією створюється новий, узагальнений мікрокуб. У ньому додатково побудовані діаграми, що наочно показують структуру доходів і витрат, динаміку продажів за останній квартал в розрізі товарів і тому подібне. Таку технологію, наприклад, використовує Укрсиббанк. За допомогою систем генерації і перегляду мікрокубів забезпечується інформаційна підтримка

співробітників 20 філій банку, розподілених по всій Україні. Мікрокуби створюються в головному банку за даними бухгалтерського і управлінського обліку, які інтегровані в єдиному фінансовому сховищі.

Аналіз бюджетних даних. Для ведення фінансового планування і обліку фактичного виконання бюджетів підприємства застосовують прикладні модулі у складі комплексних ERP- систем Галактика, БЕСТ та ін., спеціалізовані програмні комплекси, наприклад, Контур Корпорація, Бюджет холдингу, Інтальов, Бюджетне управління і тому подібне. У всіх випадках для аналізу бюджетних планів, контролю виконання бюджету і аналізу відхилень фактичних показників від планових застосовується OLAP- система (рис 2.11).

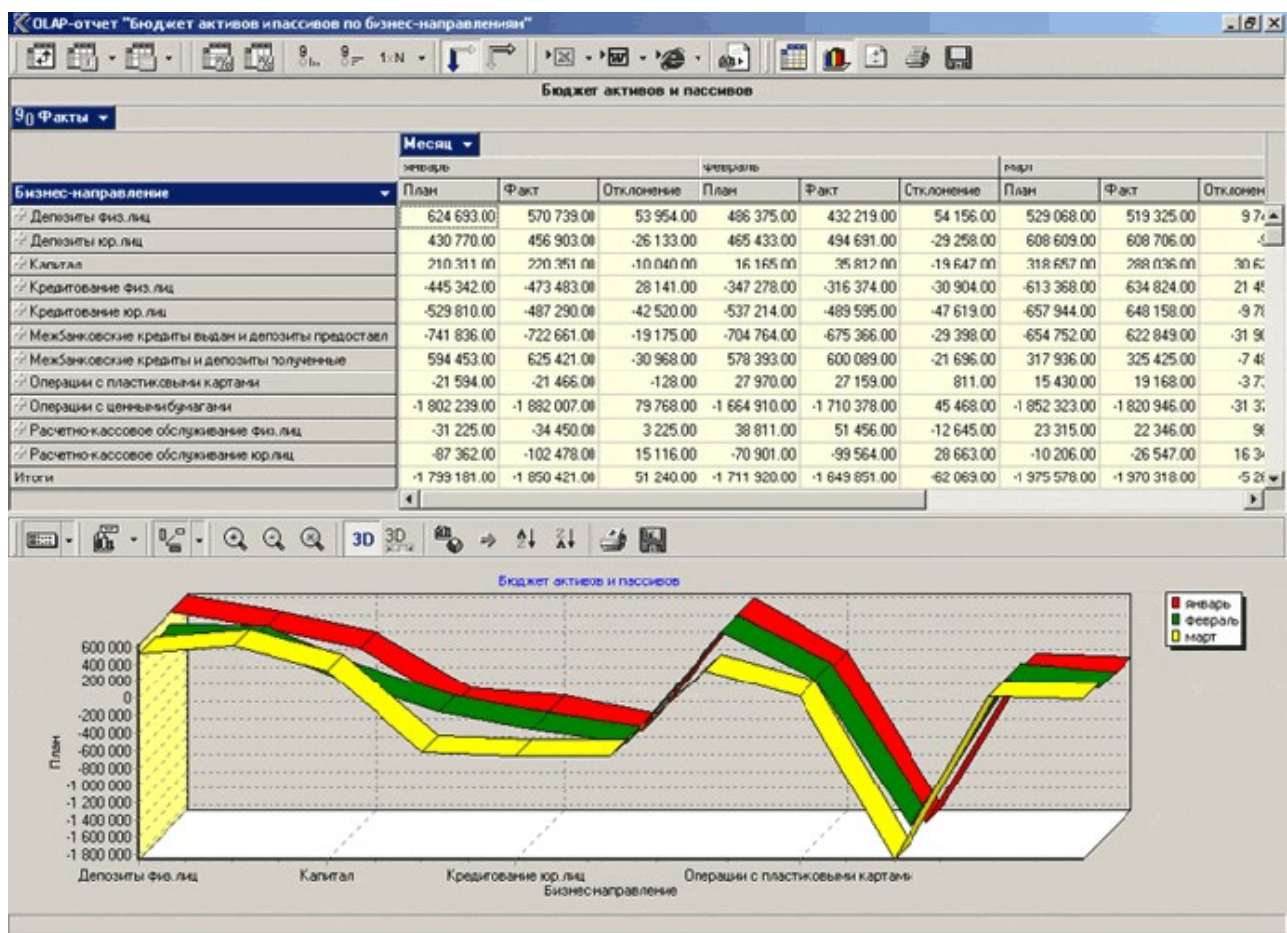


Рис. 2.11. Аналіз бюджетних даних.

Співробітники бюджетно-аналітичних і планово-економічних підрозділів випускають «план-факт» звіти про виконання бюджетів за місяць, квартал, рік,

аналізують бюджетні плани в розрізі центрів фінансової відповідальності і бізнес-напрямів, деталізують значення бюджетних статей. Наприклад, найбільший банк Казахстану «Банк ТуранАлем» випускає всю бюджетну звітність у вигляді OLAP-звітів: фінансовий план і бюджет доходів і витрат в розрізі центрів фінансової відповідальності, бізнес-напрямів, банківських продуктів і так далі.

Аналіз складських даних. Інформація про перебування і рух товарів на складі (товарні запаси, терміни зберігання товарів, постачальники і одержувачі продукції, накладні переміщення товарів) міститься в базі даних OLTP-модуля складського обліку. Аналіз цієї інформації дає відповіді на питання: «Скільки продукції було куплено замовником Івановим в третій декаді вересня?», «Який оптимальний об'єм активних і резервних запасів по даній товарній позиції?», «Чи існують сезонні коливання за даним типом товарів і яка їх амплітуда?» і тому подібне (рис. 2.12).

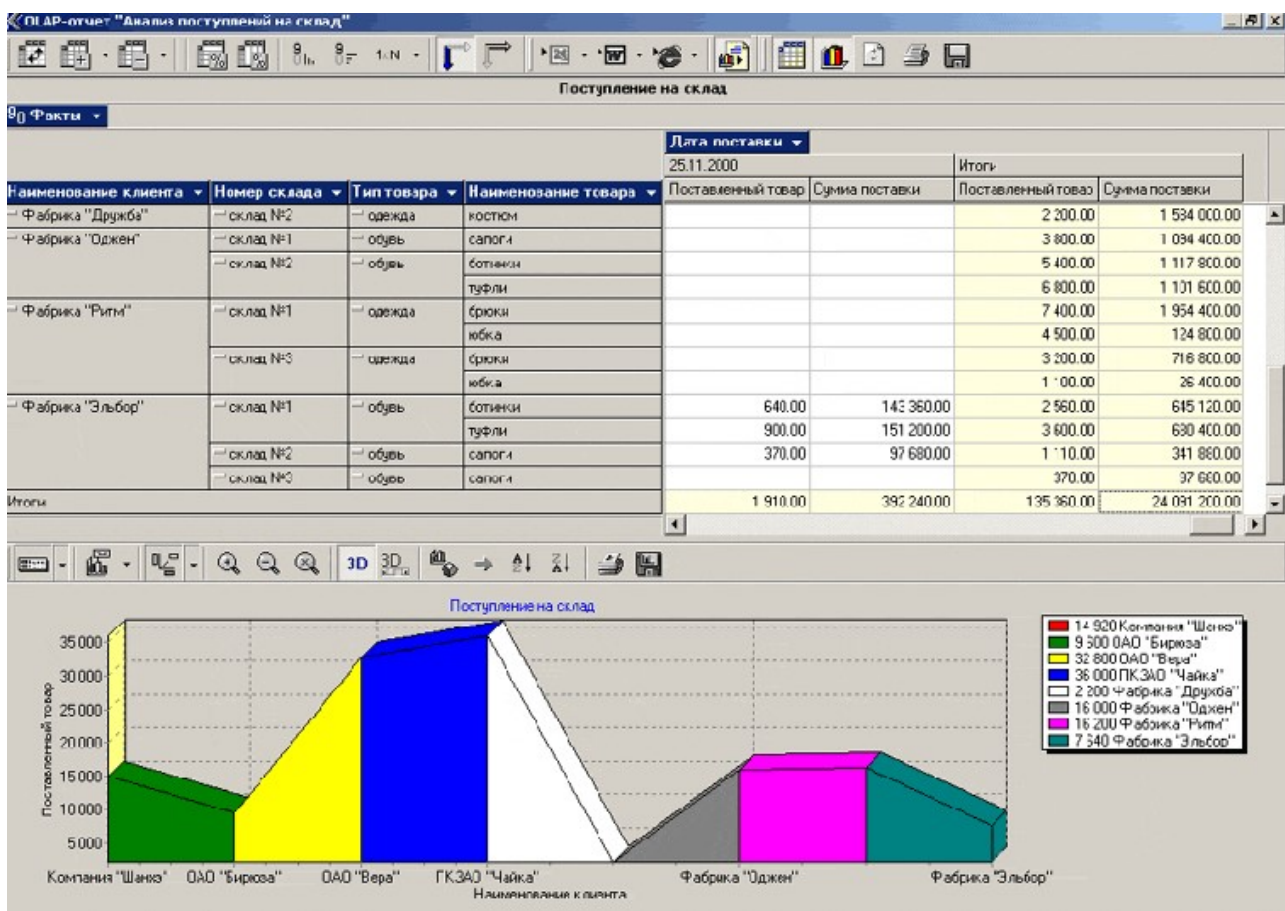


Рис.2.12. Аналіз перебування товару на складі.

Декілька операторів працюють з одним програмним додатком, який розміщується на доступному сервері і безпосередньо звертається до бази даних системи складського обліку. За допомогою OLAP-звітів вони оперативно контролюють поточну ситуацію: залишки товарних запасів в розрізі видів і партій товарів, термінів зберігання, стан відвантаження по одержувачах. Для віддалених підрозділів стан складу відбивається в мікрокубі, який розміщується на захищеній сторінці корпоративного веб-сайту.

Спільно з обліком складських даних вирішується і задача обліку транспортної логістики. Для цього до облікової системи складу підключаються довідники обслуговуючого склади транспорту, ведеться облік його використання по кожному транспортному засобу. Наприклад, у ВАТ «Трьохгорна мануфактура» аналітична система дозволяє контролювати відвантаження товарів з складів по артикулах.

Аналіз відвідуваності Web-сайту. Web-сайт є серйозним маркетинговим інструментом для багатьох компаній. Аналіз поведінки відвідувачів сайту дозволяє оцінити віддачу від маркетингових заходів і рекламних акцій, ефективність застосування on-line сервісів, інтерес до продуктів і послуг компанії і т.д. Для аналізу використовуються дані log-файлів веб-сервера, вивантажені в локальні або реляційні таблиці, або база даних сайту. Оскільки розміри таких баз, як правило, дуже великі, застосовується технологія мікрокубів. Генератор кубів щоночі за розкладом створює мікрокуб з актуальними показниками відвідуваності ресурсу. За його допомогою співробітники відділу маркетингу випускають звіти, які зберігаються в локальній мережі компанії. Запити до мікрокуба виконуються практично з «нульовим» часом очікування, мережевий трафік значно нижчий за рахунок стиснення інформації в мікрокубі. Таке рішення, наприклад, застосовується в компанії «1С: Рарус» для аналізу використання співробітниками ресурсів Інтернет і аналізу дзвінків по мобільному зв'язку.

Публікація маркетингових досліджень. Маркетингові агентства збирають інформацію, обробляють її і продають результати зацікавленим

організаціям. Наприклад, якесь агентство досліджує ринок продуктів глибокого заморожування і продає свої щомісячні звіти виробникам. Якщо в агентстві немає спеціалізованого ПО, результати досліджень можуть зводитися в Excel-файл. Цей файл засобами MS Excel можна зберігати в dbf-таблиці, з якої, у свою чергу, легко створити мікрокуб. Він продається виробникам продуктів по абонементу. Маркетологи підприємств отримують мікрокуб поштою і аналізують ринок, використовуючи програму OlapBrowser. Наприклад, англійська консалтингова компанія Decision Tree Consulting виконує аналіз конкурентного середовища для брендів по замовленнях 200 найбільших компаній, таких як Sony, Toshiba, Panasonic, Nokia, Pioneer, Sanyo, Siemens, Phillips, Hewlett Packard. Результати досліджень поставляються замовникам у вигляді мікрокубів. Індійська консалтингова компанія MARC, яка обслуговує фармацевтичні компанії Індії, також поширює результати досліджень замовникам у вигляді мікрокубів, надаючи їм видалений доступ через Інтернет.

Створення інформаційного сервісу. Електронні біржі і інформаційні агентства публікують на своїх сайтах проспекти біржових індексів, котирування цінних паперів різних емітентів, рейтинги учасників фондового ринку за різними показниками і іншу інформацію у вигляді мікрокубів. Комерсанти знайомляться з актуальними даними з будь-якої точки земної кулі через Інтернет і з допомогою OlapBrowser проводять аналіз архівних і поточних біржових зведень і аналітичних довідок. Підтримка інформації в актуальному стані забезпечується за рахунок генерації мікрокубів за розкладом. Так транснаціональна інвестиційна корпорація Fidelity Investment поставляє своїм клієнтам інформацію про цінні папери у вигляді мікрокубів через Інтернет.

Світовий ринок OLAP-додатків стабільно зростає – на 16% в 2008 році, і по прогнозах – приблизно на стільки ж в 2009 – 2010 роках. У абсолютних цифрах об'єм світового ринку OLAP, за даними «The OLAP Report» [66, с. 7], в 2007 році складав 5,7 млрд. дол., і як очікується, досягне показника в 7 млрд. дол. до 2009 року. Потрібно відзначити комерційну привабливість цього сектора - адже він є в даний час одним з самих швидкозростаючих сегментів ІТ-

ринку. Проте так було не завжди – показник зростання дуже сильно вагався і демонстрував сильну залежність від стану галузі в цілому – від більш ніж 40% зростання (у 1995- 1998 роках) до близько 5% в 2001-2002, в період кризи доткомів в США. Є всі підстави вважати, що і надалі ринок OLAP буде так само сильний схильний до впливу загально галузевих коливань кон'юнктури ІТ-ринку (рис. 2.13).

Ринок OLAP сильно консолідований – на долю трьох найбільших виробників доводилося в 2008 році близько 63% ринку, а на долю найбільших десяти – ледве більше 97% ринку (рис. 2.14). Важливим також є той факт, що нинішній лідер ринку – Microsoft – єдина компанія, яка, за інформацією «The OLAP Report» [66, с.11], з року в рік демонструє зростання своєї долі на ринку (з 11,4% до 31,6% в 2008 р.).

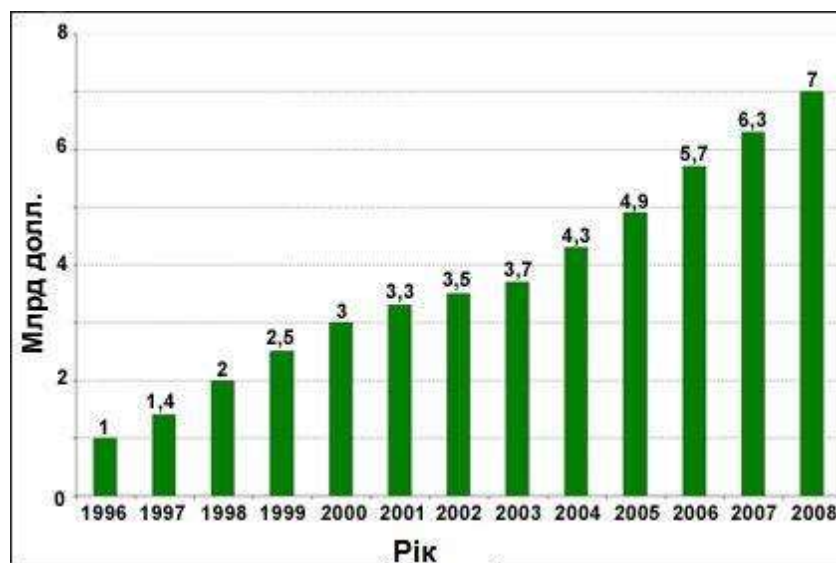


Рис. 2.13. Розвиток ринку OLAP–додатків.

Важливою тенденцією розвитку ринку OLAP-додатків є створення комплексних засобів аналізу даних, що включають як OLAP-засоби, так і інші засоби Data Mining. Поєднання OLAP з такими інструментами, як системи фільтрації і обробки даних, засоби автоматичної класифікації і кластеризації, нейронні мережі т.д., дозволяє створювати потужні аналітичні додатки з широким спектром можливого вживання. Вочевидь, саме такий універсальні додатки надалі домінуватимуть на ринку.

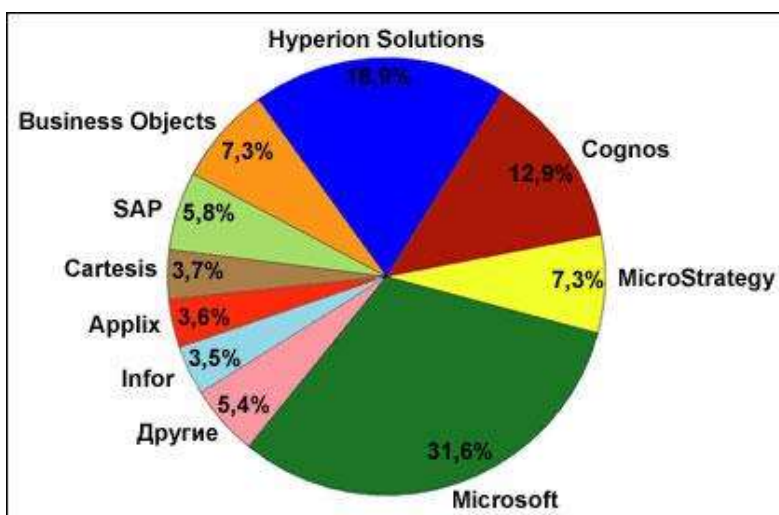


Рис. 2.14. Найбільші виробники OLAP-додатків.

Крім того, цікаво відзначити такий важливий напрям, як інтеграція OLAP-додатків з системами управління базами даних і ERP-системами. Цей напрям інтеграції активно розвиває Microsoft - Microsoft Analysis Services є додатком до Microsoft SQL Server, а рішення для управління підприємством – такі як Microsoft Dynamics AX (раніше - Microsoft Business Solutions Ахapta) містять вбудовані OLAP-модулі. Схожої практики дотримуються і інші виробники СУБД і ERP- систем – Oracle і SAP. Вочевидь, що кількість і об'єм продажів інтегрованих OLAP-додатків з часом лише збільшуватиметься [11].

Одним з важливих чинників розвитку інтелектуального аналізу даних в сучасному інформаційному світі є рух Open Source. Все більше і більше компаній або використовують продукти з відкритими вихідними кодами, або самі беруть участь і підтримують розвиток таких програмних додатків. Майже для кожного типу програмних продуктів існують їх аналоги в сегменті Open Source, і OLAP-додатки не є виключенням. Вже зараз існує цілий набір додатків, що реалізують всі функції бізнес-аналітики (такі як Pentaho Open BI Suite – повноцінна платформа для корпоративної системи аналізу даних, Palo-Server, і ін.). Головною їх перевагою можна вважати можливість змінити додаток так, щоб він найбільше підходив під конкретні умови роботи, замість того, щоб змінювати структуру ухвалення рішень, що склалася. Це може залучити крупні компанії, що використовують OLAP-додатки.

Також наслідком появи Open Source OLAP-додатків стане більше їх поширення – невеликі компанії, які до цього не застосовували дорогі OLAP-продукти із-за їх ціни, дістануть таку можливість. Внаслідок цього можна чекати появи ринку послуг з установки і підтримки Open Source OLAP-додатків, де основними споживачами будуть компанії малого і середнього бізнесу.

Ринок OLAP-додатків тісно пов'язаний з ринком консалтингових послуг. Навчання, впровадження, підтримка є природним доповненням до програмного забезпечення для аналізу даних. Вже зараз на українському ринку успішно діють компанії, які пропонують комплексні рішення по консалтингу, навчанню і впровадженню цілого спектру продуктів. Проте до того моменту, коли ринок OLAP неможливо буде відокремити від ринку консалтингових послуг, ще далеко. У довгостроковій перспективі компанії-виробники продовжуватимуть орієнтуватися на виробництво програмного забезпечення, залишаючи надання послуг партнерам.

Враховуючи вище перелічені чинники, що впливають на розвиток ринку OLAP-додатків, можна передбачити, що в середньостроковій перспективі ринок зростатиме досить швидкими темпами, але конкуренція при цьому скорочуватиметься із-за процесу злиття компаній, що продовжується. Все більше очікується комплексних рішень для аналізу даних і модулів для систем управління базами даних і ERP-систем, що реалізують функціональність OLAP. Ціна на OLAP-додатки знижуватиметься – як із-за тиску з боку лідерів ринку, так і із-за поширення Open Source OLAP продуктів. В той же час потреба в послугах з впровадження і супроводу OLAP-систем збільшиться, що приведе до збільшення кількості спеціалізованих консалтингових компаній.

Ідея обробки багатовимірних даних не є новою. Фактично вона сходить до 1962 року, коли Кен Айверсон опублікував свою книгу "Мова програмування" ("A Programming Language" - APL). Перша практична реалізація APL була здійснена в кінці 60-х років компанією IBM. APL - це математична мова з багатовимірними змінними і витонченими, хоч і досить

абстрактними операторами. Вона призначалася більше для опису багатовимірних перетворень, ніж для використання як практичної мови програмування. Так, наприклад, в ній не приділялося уваги таким приземленим питанням, як робота з файлами або вивід на друк. У дуже стислій нотації мови використовувалися грецькі символи. Насправді, тексти програм виходили вельми компактними. Вона стала відомою, як «мова тільки для написання», тому що було набагато легше переписати наново програму, ніж виправити раніше збережений текст. APL поглинала машинні ресурси і вимагала великих витрат. Програми дуже поволі виконувалися і обходилися дуже дорого. Проте не дивлячись на невдалий початок, APL не була викинута. Вона використовувалася в багатьох ділових додатках 70-х, 80-х років, які функціонально подібні до сьогоднішніх OLAP систем. Так, IBM розробила операційну систему для APL, названу VSPC, і деякі люди вважали її ідеальним середовищем для персонального використання задовго до появи електронних таблиць. У 80-х роках APL став доступний на персональних машинах, але не знайшов ринкового застосування. Альтернативою було програмування багатовимірних додатків з використанням масивів в інших мовах. Це було дуже важким завданням навіть для професійних програмістів, так що кінцевим користувачам залишалося чекати наступного покоління багатовимірних програмних продуктів.

У 1970 році вперше з'явився прикладний багатовимірний програмний продукт, що використалися в навчальних цілях - Express. Він в повністю переписаному вигляді широко використовується в сучасних OLAP-додатках, проте оригінальні концепції 70-х років залишилися далеко позаду. Сьогодні, Express залишається однією з найбільш популярних OLAP-технологій, і компанії Oracle вдається підтримувати його на рівні сучасних вимог разом з багатьма новими продуктами з архітектурою «клієнт-сервер».

Більше багатовимірних продуктів з'явилися в 80-х роках. На початку десятиліття з'явився Stratagem, в новому обличчі - Acumate, який просувався на

ринку до середини 90-х, але сьогодні, на відміну від Express, використовується дуже обмежено.

Comshare System W був багатовимірним продуктом іншого стилю. Представлений в 1981 році, він першим використовував ідею гіперкуба і був більшою мірою орієнтований на кінцевого користувача в розробці фінансових додатків. Він привніс багато концепцій, які, правда, ще добре не пропрацювали, типу не процедурних правил, повноекранного перегляду багатовимірних даних, редагування даних, інтеграція з реляційними даними (у пакетному режимі). Проте Comshare System W був достатньо важкий для апаратного забезпечення того часу і менш програмованим в порівнянні з іншими продуктами і, відповідно, був менш популярний в середовищі професіоналів. Він також поки використовується, але продається все менше, оскільки не мав тих поліпшень, які очікувалися. В кінці 80-х Comshare випустив в середовищі DOS, а пізніше для Windows, продукт під назвою Commander Prism. Він використовував ті ж концепції, що були закладені в System W. Essbase, продукт компанії Hyperion Solution, хоч і не є прямим нащадком System W, був очевидно під впливом його рішень з своєю орієнтацією на фінансові додатки і організацією гіперкуба з повними попередніми обчисленнями.

Іншим новаторським продуктом на початку 80-х був Metaphor. Він призначався для професійних маркетологів. Цей пакет також запропонував багато нових концепцій, які стали популярними тільки в 90-х роках: такі як, клієнт - серверне обчислення, багатовимірна обробка реляційних даних, розрахований на багато користувачів режим і об'єктно орієнтована розробка додатків. На жаль стандартні персональні комп'ютери не забезпечували тих характеристик, які вимагав Metaphor, і постачальники були вимушені розробляти власні персональні машини і мережі. Надалі Metaphor став працювати вдало і на серійних персональних машинах, проте він ніколи не використав стандартний графічний інтерфейс користувача (GUI).

В кінці 80-х серед інструментів кінцевого користувача для аналізу даних стали домінувати електронні таблиці. Перша багатовимірна електронна

таблиця з'явилася у вигляді Compete. Він просувався на ринок як дуже дорогий продукт для фахівців, але постачальники не забезпечили можливість захоплення ринку цим продуктом, і компанія Computer Associates придбала права на нього разом з іншими продуктами класу «spreadsheet» (електронні таблиці), включаючи Supercalc і 20/20. Основним ефектом від придбання Compete компанією СА було різке зниження ціни і зняття захисту від копіювання, що, природно, сприяло його розповсюдженню. Проте він ще не був вдалим. Протягом декількох років Compete ще зрідка можна було зустріти у вигляді навантаження в деяких комплектах постачання. Пізніше Compete був покладений в основу Supercalc 5, але багатовимірний аспект його не просувався.

Компанія Lotus була наступною, хто спробував увійти на ринок багатовимірних електронних таблиць з продуктом Improv. Це гарантувало, що продажі 1-2-3 не знизяться, але коли той з часом був випущений під Windows, Excel вже став настільки серйозним конкурентом, що продажі Improv не внесли помітних змін в розподілі ринку. Lotus, подібно СА з Compete, скинула ціну на Improv, проте і цього було недостатньо для просування на ринку, і нові розробки в цій області не отримали продовження. Виявилось, що користувачі персональних комп'ютерів віддають перевагу електронним таблицям в оригінальній версії 1-2-3 і не цікавляться новими багатовимірними можливостями, якщо вони не повністю сумісні з їх старими таблицями. Так само концепція невеликих багатовимірних настільних електронних таблиць, пропонованих як продуктивний інструмент для персональних додатків, насправді не виявилися зручними і не прижилися в реальній практиці.

Компанія Microsoft пішла по цьому шляху, додавши PivotTables до Excel. Хоча небагато користувачів Excel отримали вигоду від використання цієї можливості, це, ймовірно, єдиний факт широкого використання можливостей багатовимірного аналізу просто тому, що в світі дуже багато користувачів Excel. Excel 2000 містить витонченішу версію PivotTables, призначену для використання і як настільний інструмент OLAP і як клієнтська частина для

взаємодії з Microsoft OLAP Services. Проте можливості OLAP в Excel 2000 не є базовими, ведучими, вони скоріше вбудовані як додаткова, другорядна можливість.

В кінці 80-х років фірма Sinper увійшла до світу багатовимірних електронних таблиць спочатку з власною електронною таблицею для DOS, а потім приєднаною до версії 1-2-3 для DOS. Перетворений на продукт TM/1, він увійшов до ери Windows як сервера баз даних у форматі Excel і 1-2-3. Трохи пізніше Arbor зробив аналогічну річ, хоча його новий Essbase міг працювати тільки в режимі клієнт-сервер, тоді як продукт фірми Sinper міг так само працювати на локальному комп'ютері. Цей підхід приніс мультирозмірність в електронні таблиці, які є такими популярними серед користувачів. Таким чином, традиційні постачальники власних інструментів представлення даних кінцевому користувачеві пішли по цьому шляху, і такі продукти, як DSS Server, Express, Holos, Gentia, Mineshare, Powerplay, Metacube і Whitelight тепер з гордістю пропонують високо інтегрований доступ до електронних таблиць в своїх серверах додатків. За іронією долі за свої перші шість місяців існування Microsoft OLAP Services був одним з декількох OLAP серверів, що не мають клієнтської частини у вигляді електронної таблиці. Пропозиції компанії Microsoft з'явилися тільки в червні 1999 року в Excel 2000. Проте OLAP@Work, вбудований в Excel, заповнив цей пропуск, і поки що має набагато кращі експлуатаційні характеристики, чим власний інтерфейс Excel компанії Microsoft.

Деякі користувачі вимагають для своїх багатовимірних додатків можливості обробки дуже великих багатовимірних баз даних. І реляційні OLAP-інструменти розвиваються в цьому напрямі, відгукуючись на ці потреби. Вони надають звичайні засоби проглядання багатовимірних даних, а іноді включають інтерфейс кінцевого користувача у вигляді електронної таблиці, навіть якщо всі дані зберігаються в реляційних СУБД. Такі засоби є дуже дорогими для користувача, вони менш продуктивні, чим спеціалізовані інструменти багатовимірного аналізу, але вони забезпечують цю, таку

популярну форму аналізу даних, навіть якщо останні зберігаються не у вигляді багатовимірних структур.

Інші поставщики розвивають те, що сьогодні називається настільним OLAP: невеликі куби, що генеруються з великих баз даних і потім завантажуються в персональний комп'ютер для обробки. Вони дійсно досягають великого успіху. А коли постачальник продає обидві можливості: і інструмент формування реляційних запитів і інструмент багатовимірного аналізу і формування звітів, то досягає більшого успіху у кінцевих користувачів, ніж в інших випадках.

2.5. Основні архітектури OLAP - систем

Системи інтелектуального аналізу даних зазвичай володіють засобами надання користувачеві агрегатних даних для різних вибірок з початкового набору в зручному для сприйняття і аналізу вигляді. Як правило, такі агрегатні функції утворюють багатовимірний (і, отже, не реляційний) набір даних (нерідко званий гіперкубом або метакубом), осі якого містять параметри, а ячейки - залежні від них агрегатні дані - причому зберігатися такі дані можуть і в реляційних таблицях, але в даному випадку ми говоримо про логічну організацію даних, а не про фізичну реалізацію їх зберігання. Уздовж кожної осі дані можуть бути організовані у вигляді ієрархії, що представляє різні рівні їх деталізації. Завдяки такій моделі даних користувачі можуть формулювати складні запити, генерувати звіти, отримувати підмножини даних [4, 9, 45, 71].

Технологія комплексного багатовимірного аналізу даних отримала назву OLAP (On-Line Analytical Processing). OLAP - це ключовий компонент організації сховищ даних. Концепція OLAP була описана в 1993 році Едгаром Коддом, відомим дослідником баз даних і автором реляційної моделі даних. У 1995 році на основі вимог, викладених Коддом, був сформульований так званий тест FASMI (Fast Analysis of Shared Multidimensional Information - швидкий

аналіз розподіленої багатовимірної інформації), що включає наступні вимоги до додатків для багатовимірного аналізу:

- Fast (Швидкий). Надання користувачеві результатів аналізу за прийнятний час (зазвичай не більше 5 с), нехай навіть ціною менш детального аналізу;
- Analysis (Аналіз). Можливість здійснення будь-якого логічного і статистичного аналізу, характерного для даного додатку, і його збереження в доступному для кінцевого користувача вигляді;
- Shared (Розподілений доступ). Розрахований на багато користувачів доступ до даних з підтримкою відповідних механізмів блокувань і засобів авторизованого доступу;
- Multidimensional (Багатовимірність). Багатовимірне концептуальне представлення даних, включаючи повну підтримку для ієрархій і множинних ієрархій (це - ключова вимога OLAP);
- Information (Інформація). Можливість звертатися до будь-якої потрібної інформації незалежно від її об'єму і місця зберігання.

Слід зазначити, що OLAP-функціональність може бути реалізована різними способами, починаючи з простих засобів аналізу даних в офісних додатках і закінчуючи розподіленими аналітичними системами, заснованими на серверних продуктах. Але перш ніж говорити про різні реалізації цієї функціональності, давайте розглянемо, що ж є куб OLAP з логічної точки зору. Як приклад реляційної бази даних, який ми використовуватимемо для ілюстрації принципів OLAP, скористаємося базою даних Northwind, що входить в комплекти постачання Microsoft SQL Server і є типовою базою даних, що зберігає дані про торгові операції компанії, що займається оптовими постачаннями продовольства. До таких даних відносяться відомості про постачальників, клієнтів, компанії, що здійснюють доставку, список товарів, що поставляються і їх категорій, дані про замовлення і замовлені товари, список співробітників компанії [11].

Для розгляду концепції OLAP скористаємося представленням Invoices і таблицями Products і Categories з бази даних Northwind, створивши запит, в результаті якого отримаємо докладні відомості про всі замовлені товари і виписані рахунки:

```
SELECT dbo.Invoices.Country,  
dbo.Invoices.City,  
dbo.Invoices.CustomerName,  
dbo.Invoices.Salesperson,  
dbo.Invoices.OrderDate,  
dbo.Categories.CategoryName,  
dbo.Invoices.ProductName,  
dbo.Invoices.ShipperName,  
dbo.Invoices.ExtendedPrice  
FROM dbo.Products INNER JOIN  
dbo.Categories ON dbo.Products.CategoryID = dbo.Categories.CategoryID INNER  
JOIN  
dbo.Invoices ON dbo.Products.ProductID = dbo.Invoices.ProductID
```

Цей запит звертається до представлення Invoices, що містить відомості про всі виписані рахунки, а також до таблиць Categories і Products, що містить дані про категорії продуктів, які замовлялися, і про самі продукти відповідно. В результаті цього запиту ми отримаємо набір даних про замовлення, що включає категорію і найменування замовленого товару, дату розміщення замовлення, ім'я співробітника, що виписав рахунок, місто, країну і назву компанії-замовника, а також найменування компанії, що відповідає за доставку. Для зручності збережемо цей запит у вигляді уявлення, назвавши його Invoices1. Результат звернення до цього уявлення приведений на малюнку 2.15.

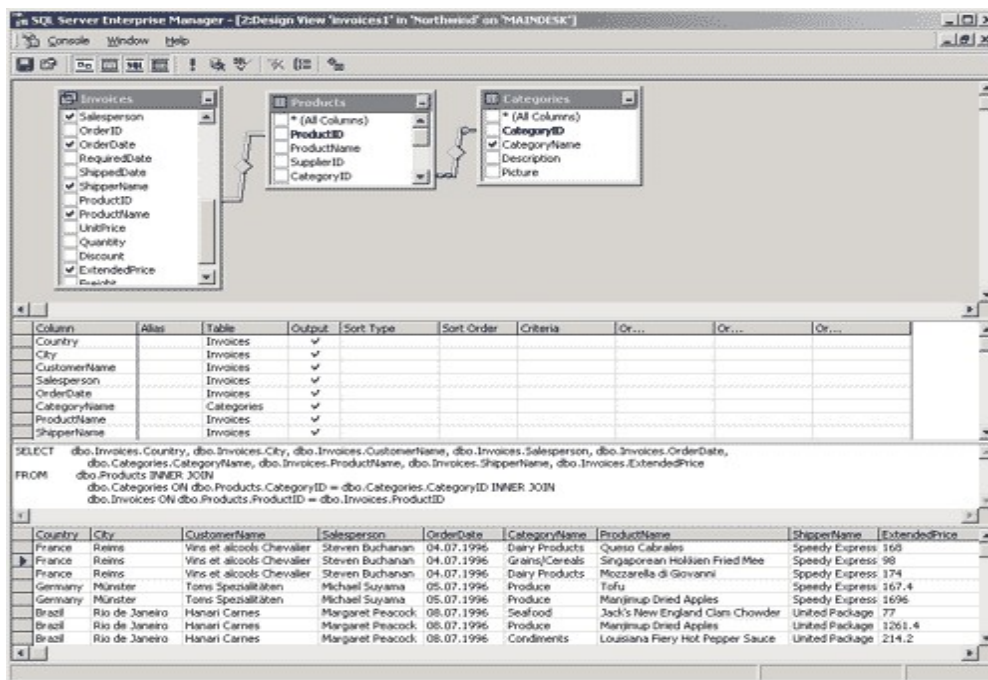


Рис. 2.15. Результат звернення до представлення Invoices 1.

Які агрегатні дані ми можемо отримати на основі цього уявлення?

Звичайно це відповіді на питання типу:

- Яка сумарна вартість замовлень, зроблених клієнтами з Франції?
- Яка сумарна вартість замовлень, зроблених клієнтами з Франції і доставлених компанією Speedy Express?
- Яка сумарна вартість замовлень, зроблених клієнтами з Франції в 1997 році і доставлених компанією Speedy Express?

Переведемо ці питання в запити на мові SQL.

Питання	SQL- запит
Яка сумарна вартість замовлень, зроблених клієнтами з Франції?	SELECT SUM (ExtendedPrice) FROM invoices1 WHERE Country='France'

Яка сумарна вартість замовлень, зроблених клієнтами з Франції і доставлених компанією Speedy Express?	SELECT SUM (ExtendedPrice) FROM invoices1 WHERE Country='France' AND ShipperName='Speedy Express'
Яка сумарна вартість замовлень, зроблених клієнтами з Франції в 1996 році і доставлених компанією Speedy Express?	SELECT SUM (ExtendedPrice) FROM Ord_pmt WHERE CompanyName='Speedy Express' AND OrderDate BETWEEN 'December 31, 1995' AND 'April 1, 1996' AND ShipperName='Speedy Express'

Результатом будь-якого з перерахованих вище запитів є число. Якщо в першому із запитів замінити параметр 'France' на 'Austria' або на назву іншої країни, можна знову виконати цей запит і отримати інше число. Виконавши цю процедуру зі всіма країнами, ми отримаємо наступний набір даних:

Country	SUM (ExtendedPrice)
Argentina	7327.3
Austria	110788.4
Belgium	28491.65
Brazil	97407.74
Canada	46190.1
Denmark	28392.32
Finland	15296.35
France	69185.48
Germany	209373.6
...	...

Отриманий набір агрегатних значень може бути інтерпретований як одномірний набір даних.

Тепер звернемося до другого з приведених вище запитів, який містить дві умови в пропозиції WHERE. Якщо виконувати цей запит, підставляючи в

нього всі можливі значення параметрів Country і ShipperName, ми отримаємо двомірний набір даних наступного вигляду:

Country	ShipperName		
	Federal Shipping	Speedy Express	United Package
Argentina	1 210.30	1 816.20	5 092.60
Austria	40 870.77	41 004.13	46 128.93
Belgium	11 393.30	4 717.56	17 713.99
Brazil	16 514.56	35 398.14	55 013.08
Canada	19 598.78	5 440.42	25 157.08
Denmark	18 295.30	6 573.97	7 791.74
Finland	4 889.84	5 966.21	7 954.00
France	28 737.23	21 140.18	31 480.90
Germany	53 474.88	94 847.12	81 962.58
...

Такий набір даних називається зведеною таблицею (Pivot Table) або кросс-таблицею (Cross Table). Створювати подібні таблиці дозволяють багато електронних таблиць і настільні СУБД - від Paradox для DOS до Microsoft Excel.

Третій з розглянутих вище запитів має вже три параметри в умові WHERE. Варіюючи їх, ми отримаємо трьохмірний набір даних (рис. 2.16). Ячейки куба містять агрегатні дані, відповідні значенням параметрів запиту, що знаходяться на осях куба в пропозиції WHERE. Можна отримати набір двомірних таблиць за допомогою перетину куба площинами, паралельними його граням (для їх позначення використовують терміни Cross Sections і Slices).

	Federal Shipping	Speedy Express	United Package
Argentina	11806.28	9190.48	1263.9
Austria			4039.5
Belgium	1745.42	1207.28	14924.12
Brazil			5208.28
Canada	2952.4		
Denmark	1739.76	1376	
Finland	5470.98	3538.92	2328.46
France	11927.48	9823.43	11052.28
Germany	2208.62	1739.6	4681.16
Ireland		330.9	608
Italy	2139.1		1357.6
Mexico			786
Norway	459		
Poland	1268.3	716.72	285.12
Portugal	236.5	220.3	2235.8
Spain	3021.23	2380	1488.8
Sweden	2490.5		1628.32
Switzerland	5094.88	1520.8	901.2
UK	11192.65	6347.52	14091.93
USA	3925.58	3171.92	
Venezuela	11806.28	9190.48	1263.9

Рис. 2.16. Трьохмірний набір агрегатних даних.

Якщо в пропозиції WHERE міститься чотири або більше параметрів, результуючий набір значень (також званий OLAP-кубом) може бути 4-мірним, 5-мірним і т.д.

Розглянувши, що є багатовимірними OLAP-кубами, перейдемо до деяких ключових термінів і понять, використовуваних при багатовимірному аналізі даних.

Разом з сумами в ячейках OLAP-куба можуть міститися результати виконання інших агрегатних функцій мови SQL, таких як MIN, MAX, AVG, COUNT, а в деяких випадках - і інших (дисперсії, середньоквадратичного відхилення і т.д.). Для опису значень даних в ячейках використовується термін Summary (у загальному випадку в одному кубі їх може бути декілька), для позначення початкових даних, на основі яких вони обчислюються, - термін Measure, а для позначення параметрів запитів - термін Dimension (що перекладається як «вимірювання», коли йдеться про OLAP-куби, і як «розмірність», коли йдеться про сховища даних). Значення, що відкладаються на осях, називаються членами вимірювань (Members).

Кажучи про вимірювання, слід згадати про те, що значення, що наносяться на осі, можуть мати різні рівні деталізації. Наприклад, нас може цікавити сумарна вартість замовлень, зроблених клієнтами в різних країнах, або сумарна вартість замовлень, зроблених іногородніми клієнтами або навіть окремими клієнтами. Природно, результуючий набір агрегатних даних в другому і третьому випадках буде детальнішим, ніж в першому. Відмітимо, що можливість отримання агрегатних даних з різним ступенем деталізації відповідає одній з вимог, що пред'являються до сховищ даних, - вимозі доступності різних зрізів даних для порівняння і аналізу.

Оскільки в розглянутому прикладі в загальному випадку в кожній країні може бути декілька міст, а в місті - декілька клієнтів, можна говорити про ієрархії значень у вимірюваннях. В цьому випадку на першому рівні ієрархії розташовуються країни, на другому - міста, а на третьому - клієнти (рис. 2.17).

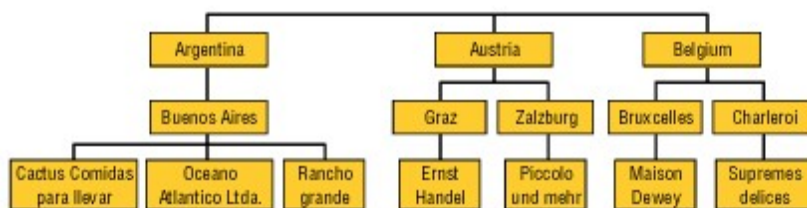


Рис. 2.17. Ієрархія у вимірюванні, пов'язаному з географічним положенням клієнтів.

Відзначимо, що ієрархії можуть бути збалансованими (balanced), як, наприклад, ієрархія, представлена на малюнку 2.17, а також ієрархії, засновані на даних типу «дата-час», і незбалансованими (unbalanced). Типовий приклад незбалансованої ієрархії - ієрархія типу «начальник-підлеглий (її можна побудувати, наприклад, використовуючи значення поля SalesPerson початкового набору даних з розглянутого вище прикладу)», представлений на рис. 2.18. Іноді для таких ієрархій використовується термін Parent-child hierarchy.

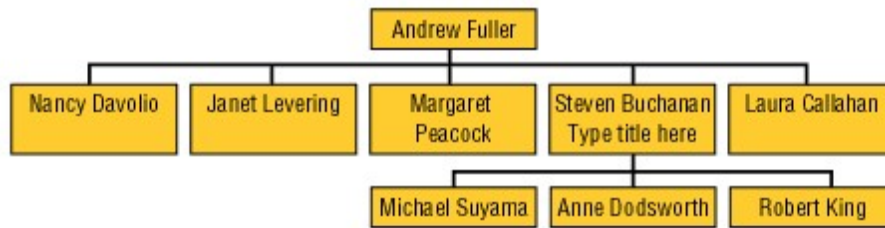


Рис. 2.18. Незбалансована ієрархія.

Існують також ієрархії, що займають проміжне положення між збалансованими і незбалансованими (вони позначаються терміном ragged – «нерівний»). Зазвичай вони містять такі члени, логічні «батьки» яких знаходяться не на безпосередньо вищестоящому рівні (наприклад, в географічній ієрархії є рівні Country, City і State, але при цьому в наборі даних є країни, що не мають штатів або регіонів між рівнями Country і City (рис. 2.19).

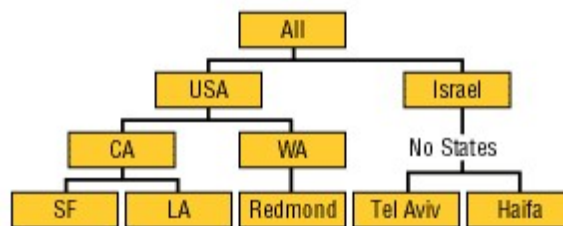


Рис. 2.19. «Нерівна» ієрархія.

Відзначимо, що незбалансовані і "нерівні" ієрархії підтримуються далеко не всіма OLAP-засобами. Наприклад, в Microsoft Analysis Services 2000 підтримуються обидва типи ієрархії, а в Microsoft OLAP Services 7.0 - тільки збалансовані. Різним в різних OLAP-засобах може бути і число рівнів ієрархії, і максимально допустиме число членів одного рівня, і максимально можливе число самих вимірювань.

Багатовимірний аналіз даних може бути проведений за допомогою різних засобів, які умовно можна розділити на клієнтські і серверні OLAP-засоби. Клієнтські OLAP-засоби є додатки, що здійснюють обчислення агрегатних даних (сум, середніх величин, максимальних або мінімальних

значень) і їх відображення, при цьому самі агрегатні дані містяться в кеші усередині адресного простору такого OLAP-засобу. Якщо початкові дані містяться в настільній СУБД, обчислення агрегатних даних проводиться самим OLAP-засобом. Якщо ж джерело початкових даних - серверна СУБД, багато хто з клієнтських OLAP-засобів посилає на сервер SQL-запити і в результаті отримують агрегатні дані, обчислені на сервері [3, 51, 62].

Відзначимо, що клієнтські OLAP-засоби застосовуються, як правило, при малому числі вимірювань (зазвичай рекомендується не більше шести) і невеликій різноманітності значень цих параметрів, - адже отримані агрегатні дані повинні уміщатися в адресному просторі подібного засобу, а їх кількість росте експоненціально при збільшенні числа вимірювань. Тому навіть найпримітивніші клієнтські OLAP-засоби, як правило, дозволяють зробити попередній підрахунок об'єму необхідної оперативної пам'яті для створення в ній багатовимірного куба. Багато клієнтських OLAP-засобів дозволяють зберегти вміст кеша з агрегатними даними у вигляді файлу, що, у свою чергу, дозволяє не проводити їх повторне обчислення. Відзначимо, що нерідко така можливість використовується для відчуження агрегатних даних з метою передачі їх іншим організаціям або для публікації. Типовим прикладом таких відчужуваних агрегатних даних є статистика захворюваності в різних регіонах і в різних вікових групах, яка є відкритою інформацією, що публікується міністерствами охорони здоров'я різних країн і Всесвітньою організацією охорони здоров'я.

Ідея збереження кеша з агрегатними даними у файлі отримала свій подальший розвиток в серверних OLAP-засобах, в яких збереження і зміна агрегатних даних, а також підтримка сховища, що містить їх, здійснюються окремим додатком або процесом, званим OLAP-сервером. Клієнтські додатки можуть запрошувати подібне багатовимірне сховище і у відповідь отримувати ті або інші дані. Деякі клієнтські додатки можуть також створювати такі сховища або оновлювати їх відповідно до початкових даних, що змінилися.

Переваги застосування серверних OLAP-засобів в порівнянні з клієнтськими OLAP-засобами схожі з перевагами застосування серверних СУБД в порівнянні з настольними: у разі застосування серверних засобів обчислення і зберігання агрегатних даних відбуваються на сервері, а клієнтський додаток отримує лише результати запитів до них, що дозволяє в загальному випадку понизити мережевий трафік, час виконання запитів і вимоги до ресурсів, споживаним клієнтським додатком. Відзначимо, що засоби аналізу і обробки даних масштабу підприємства, як правило, базуються саме на серверних OLAP-засобах, наприклад, таких як Oracle Express Server, Microsoft SQL Server 2000 Analysis Services, Hyperion Essbase, продуктах компаній Crystal Decisions, BusinessObjects, Cognos, SAS Institute. Оскільки всі провідні виробники серверних СУБД створюють ті або інші серверні OLAP-засоби, вибір їх достатньо широкий і майже у всіх випадках можна придбати OLAP-сервер того ж виробника, що і у самого сервера баз даних.

OLAP-система складається з множини компонент. На найвищому рівні уявлення система включає джерело даних, OLAP-сервер і клієнта. Джерело даних є засіб, з якого беруться дані для аналізу. Дані з джерела переносяться або копіюються на OLAP-сервер, де вони систематизуються і готуються для швидшого згодом формування відповідей на запити. Клієнт - це призначений для користувача інтерфейс до OLAP-сервера.

Джерела даних. Джерелом в OLAP-системах є сервер, що поставляє дані для аналізу. Залежно від області використання OLAP-продукту джерелом може служити сховище даних, успадкована база даних, що містить загальні дані, набір таблиць, об'єднуючих фінансові дані або будь-яка комбінація перерахованого. Здатність OLAP-продукту працювати з даними з різних джерел дуже важлива. Вимога єдиного формату або єдиної бази, в яких би зберігалися всі початкові дані, не підходить адміністраторам баз даних. Крім того, такий підхід зменшує гнучкість і потужність OLAP-продукту. Адміністратори і користувачі вважають, що OLAP-продукти, що забезпечують витягання даних

не тільки з різних, але і з безлічі джерел, виявляються гнучкішими і кориснішими, ніж ті, що мають жорсткіші вимоги.

Сервер. Прикладною частиною OLAP-системи є OLAP-сервер. Ця складова виконує всю роботу і зберігає в собі всю інформацію, до якої забезпечується активний доступ. Архітектурою сервера управляють різні концепції. Зокрема, основною функціональною характеристикою OLAP-продукту є використання для зберігання даних багатовимірної (ММБД, MDDDB) або реляційної (РДБ, RDB) бази даних.

MOLAP. MOLAP - це Multidimensional On-Line Analytical Processing, тобто Багатовимірний OLAP. Це означає, що сервер для зберігання даних використовує ММБД. Оскільки більшість OLAP-продуктів засновані на ММБД, під OLAP часто розуміють також і MOLAP. Сенс використання ММБД очевидний. Вона може ефективно зберігати багатовимірні за своєю природою дані, забезпечуючи засоби швидкого обслуговування запитів до бази даних. Дані передаються від джерела даних в багатовимірну базу даних, а потім база даних піддається агрегації. Попередній розрахунок - це те, що прискорює OLAP-запити, оскільки розрахунок зведених даних вже проведений. Час запиту стає функцією виключно часу, необхідного для доступу до окремого фрагмента даних і виконання розрахунку. Цей метод підтримує концепцію, згідно якої робота проводиться одного разу, а результати потім використовуються знову і знову. Багатовимірні бази даних є відносно новою технологією. Використання ММБД має ті ж недоліки, що і більшість нових технологій. А саме - вони не так стійкі, як РБД, і в тій же мірі не оптимізовані. Інше слабке місце ММБД полягає в неможливості використовувати більшість багатовимірних баз в процесі агрегації даних, тому потрібний час для того, щоб нова інформація стала доступна для аналізу.

«Вибух» бази даних є феномен багатовимірних баз. Не дивлячись на те, що ця проблема досліджувалася фахівцями, проте, важко пояснити, чому і як це відбувається. Представляється, що це пов'язано з розрідженістю бази даних і попередньою агрегацією даних. Якщо багатовимірна база даних містить

невелике число елементів даних, порівнянне з кількістю забезпечуваних нею рівнів агрегації, кожен фрагмент даних вноситиме більший внесок до всіх отримуваних з нього даних. Коли база даних «вибухає», розмір її стає істотно більше, ніж він повинен бути. Складно визначити умови «вибуху» бази даних або передбачити, чи «вибухне» якась конкретна структура. Одним з підходів, який, схоже, може допомогти вирішити проблему «вибуху», є динамічне управління розрідженими даними. Ця методика дозволяє аналізувати свої власні моделі зберігання і оптимізувати їх з метою запобігання «вибуху» бази даних.

ROLAP. ROLAP - це Relational On-Line Analytical Processing, тобто Реляційний OLAP. Термін ROLAP означає, що OLAP-сервер базується на реляційній базі даних. Початкові дані вводяться в реляційну базу даних, зазвичай по схемі "зірка" або схемі "сніжинка", що сприяє скороченню часу витягання. Сервер забезпечує багатовимірну модель даних за допомогою оптимізованих SQL-запитів. Існує ряд причин для вибору саме реляційною, а не багатовимірної бази даних. РБД - це добре відпрацьована технологія, що має безліч можливостей для оптимізації. До того ж, РБД підтримують крупніші об'єми даних, чим ММБД. Вони якраз і спроектовані для таких об'ємів. Основним аргументом проти РБД є складність запитів, необхідних для отримання інформації з великої бази даних за допомогою SQL. Недосвідчений SQL-програміст міг би з легкістю обтяжити цінні системні ресурси спробами виконати який-небудь подібний запит, який в ММБД виконується набагато простіше.

Прикладний OLAP (HOLAP). Безумовно, це найбільша область, і це, загалом, те, з чим зазвичай зв'язують і що зазвичай розуміють під терміном «OLAP». Прикладний OLAP, як правило, складається з багатовимірних баз даних, доступ до яких відбувається через конкретний додаток, або, можливо, через безліч додатків. Постачальники в даній області ринку в основному пропонують клієнти для бази даних. Клієнт може бути як простим засобом перегляду, так і могутнішим додатком.

Настільний OLAP (DOLAP). Представниками настільного OLAP є продукти, що необов'язково з'єднуються з сервером. Вони можуть запускатися в основному на клієнтській частині, хоча дані у формі куба даних можуть завантажувати і з сервера. Той факт, що куб даних будується і зберігається на машині користувача, дозволяє рекомендувати їх тим, хто часто використовує портативні комп'ютери або хто нечасто запускає настільки складні звіти, що для їх формування необхідна вища швидкість клієнта, а, отже, і могутніший сервер для її забезпечення.

Швидка реалізація запитів є імперативом для OLAP. Це один з базових принципів OLAP - здатність інтуїтивно маніпулювати даними вимагає швидкого витягання інформації. В цілому, чим більше обчислень необхідно провести, щоб отримати фрагмент інформації, тим повільніше відбувається відгук. Тому, щоб зберегти маленький час реалізації запитів, фрагменти інформації, звернення до яких зазвичай відбувається найчастіше, але які при цьому вимагають обчислення, піддаються попередній агрегації. Тобто вони підраховуються і потім зберігаються в базі даних як нові дані. Як приклад типу даних, який допустимо розрахувати наперед, можна привести зведені дані - наприклад, показники продажів по місяцях, кварталах або роках, для яких дійсно введеними даними є щоденні показники.

Різні постачальники дотримуються різних методів відбору параметрів, що вимагають попередньої агрегації і числа заздалегідь обчислюваних величин. Підхід до агрегації впливає одночасно і на базу даних і на час реалізації запитів. Якщо обчислюється більше величин, вірогідність того, що користувач запитає вже обчислену величину, зростає, і тому час відгуку скорочується, оскільки не доведеться запрошувати початкову величину для обчислення. Проте, якщо обчислити всі можливі величини - це не краще рішення - у такому разі істотно зростає розмір бази даних, що зробить її некерованою, та і час агрегації буде дуже великим. До того ж, коли в базу даних додаються числові значення, або якщо вони змінюються, інформація ця повинна відбиватися на заздалегідь обчислених величинах, залежних від нових даних.

Клієнт. Клієнт - це якраз те, що використовується для уявлення і маніпуляцій з даними в базі даних. Клієнт може бути і достатньо нескладним - у вигляді таблиці, що включає такі можливості OLAP, як, наприклад, обертання даних (півотінг) і поглиблення в дані (дріллінг). Воно повинно бути спеціалізованим, але мати такий же простий засіб проглядання звітів або бути таким же могутнім інструментом, як створений на замовлення додаток, спроектований для складних маніпуляцій з даними. Клієнт є настільки важливий, що безліч постачальників зосереджують свої зусилля виключно на розробці клієнта. Все, що включається до складу цих додатків, представляє собою стандартний погляд на інтерфейс, наперед задані функції і структуру, а також швидкі рішення для більшості стандартних ситуацій, наприклад, популярні фінансові пакети. Наперед створені фінансові додатки дозволять фахівцям використовувати звичні фінансові інструменти без необхідності проектувати структуру бази даних або загальноприйняті форми і звіти.

Інструмент запитів/генератор звітів. Інструмент запитів або генератор звітів пропонує простий доступ до OLAP-даних. Вони мають простий у використанні графічний інтерфейс і дозволяють користувачам створювати звіти переміщенням об'єктів методом "drag and drop". Тоді як традиційний генератор звітів надає користувачеві можливість швидко випускати форматовані звіти, генератори звітів, підтримуючі OLAP, формують актуальні звіти. Кінцевий продукт є звіт, що має можливості поглиблення в дані до рівня подробиць, обертання (півотінг) звітів, підтримки ієрархій і ін.

Сьогодні в багатьох напрямках бізнесу за допомогою електронних таблиць проводяться різні форми аналізу корпоративних даних. У якомусь сенсі це ідеальний засіб створення звітів і проглядання даних. Аналітик може створювати макроси, що працюють з даними у вибраному напрямі, а шаблон може бути спроектований таким чином, що, коли відбувається введення даних, формули розраховують правильні величини, виключаючи необхідність неодноразового введення простих розрахунків. Проте, все це дає в результаті «плоский» звіт, що означає наступне: як тільки він створений, важко

розглядати його в різних аспектах. Наприклад, діаграма відображає інформацію за деякий часовий період, - скажімо, за місяць. І якщо хтось бажає побачити показники за день (в протилежність даним за місяць), необхідно буде створити абсолютно нову діаграму. Належить визначити нові набори даних, додати в діаграму нові мітки і внести безліч інших простих, але трудомістких змін. Крім того, існує ряд областей, в яких можуть бути допущені помилки, що в цілому зменшує надійність. Коли до таблиці додається OLAP, з'являється можливість створювати єдину діаграму, а потім піддавати її різним маніпуляціям з метою надання користувачеві необхідної інформації, не обтяжуючи себе створенням всіх можливих уявлень.

Інтернет в ролі клієнта. Новим членом сімейства OLAP-клієнтів є Інтернет. Існує маса переваг у формуванні OLAP-звітів через Інтернет. Найбільш істотною представляється відсутність необхідності в спеціалізованому програмному забезпеченні для доступу до інформації.

Кожен Інтернет-продукт специфічний. Деякі спрощують створення Web-сторінок, але мають меншу гнучкість. Інші дозволяють створювати представлення даних, а потім зберігати їх як статичні HTML-файли. Все це дає можливість проглядати дані через Інтернет, але не більш того. Активно маніпулювати даними з їх допомогою неможливо. Існує і інший тип продуктів - інтерактивний і динамічний, такий, що перетворює такі продукти на повнофункціональні інструменти. Користувачі можуть здійснювати поглиблення в дані, півотінг, обмеження вимірювань і ін. Перш, ніж вибрати засіб реалізації Інтернет, важливо зрозуміти, які функціональні можливості потрібні від Web-рішення, а потім визначити, який продукт найкращим чином утілить цю функціональність.

Додатки. Додатки - це тип клієнта, що використовує бази даних OLAP. Вони ідентичні інструментам запитів і генераторам звітів, описаним вище, але, крім того, вони вносять до продукту ширші функціональні можливості. Додаток, як правило, володіє більшою потужністю, чим інструмент запити.

Середовище. Зазвичай постачальники OLAP забезпечують середовище розробки для створення користувачами власних настроєних додатків. Середовище розробки в цілому є графічним інтерфейсом, що підтримує об'єктно-орієнтовану розробку додатків. До того ж, більшість постачальників забезпечують API, який може використовуватися для інтеграції баз даних OLAP з іншими додатками.

Розглянемо деякі напрями діяльності основних виробників програмних засобів підтримки OLAP - технологій з урахуванням вищеприведенною класифікації архітектур.

Різні постачальники реалізують OLAP на основі власних корпоративних уявлень про те, що повинно входити в ідеальний OLAP-продукт. Постачальники, розглянуті нижче, дотримувалися різних підходів, включаючи і ті, що засновані на багатовимірній базі даних, реляційній базі даних і додатках, що реалізують можливості OLAP на різних рівнях. Процес їх реального функціонування може служити кількісним параметром при розгляді різних підходів до OLAP. З цією метою OLAP Council розробив атестаційне завдання APB-1 для кількісного порівняння роботи різних OLAP-продуктів. OLAP Council є консорціумом, утвореним декількома постачальниками OLAP для підтримки ключових принципів OLAP. Вони визнають, що OLAP є найважливішою технологією для забезпечення корпоративних аналітиків інструментами, потрібними для виконання необхідного їм аналізу. У квітні 1996 р. було випущено перше атестаційне завдання в області OLAP - APB-1. Контрольне завдання визначає параметри бази даних і встановлює набір з 10 запитів, що відображають нормальне використання. Постачальник OLAP створює відповідну базу даних і потім запускає запити. Окреме вимірювання - AQT (середній час запиту, Average Query Time), генерується на основі часу, який витрачається на завантаження бази даних, агрегацію даних і подальший запуск запитів. Правила визначають, що легально і що нелегально - наприклад, чи потрібно розраховувати наперед значення даних, які з розрахованих величин можуть зберігатися і ін.

В області MOLAP архітектури лідером є компанія Oracle [43]. Лінійка продуктів Express та Oracle OLAP Services є корпоративним підходом Oracle до OLAP, включаючи сервер, клієнтську частину, можливості ROLAP і Інтернет-рішення. Express Server був головною OLAP-машиною для Oracle. Він надавав багатовимірну базу даних і був машиною для інших OLAP-продуктів фірми. До того ж, Oracle пропонує Personal Express - локально працюючий сервер Express. Він надає користувачам доступ і можливість працювати з базою даних в автономному режимі. Все це ідеально підходить для мобільних комп'ютерів. Головною особливістю Express Server є здатність використовувати різноманітні джерела даних. Дані для багатовимірної бази даних можуть збиратися з реляційної бази (за допомогою Express Relational Access Manager), багатовимірної бази, табличного або плоского файлу. Користувачі можуть розробляти власні OLAP-додатки для Express за допомогою Express Objects, який забезпечує розробника графічним інтерфейсом і об'єктно-орієнтованим підходом для створення додатків. Oracle також забезпечує безліч шляхів доступу до бази даних. Express Analyzer містить графічний інтерфейс до бази даних і дозволяє користувачеві легко формувати звіти. Крім того, Analyzer може мати загальні об'єкти з Express Objects, а також випускати додатки, розроблені з їх допомогою. Discoverer найточніше можна описати як інструмент запитів до даних. Він простіший, ніж Analyzer, проте найбільш популярний серед засобів запитів до даним. Крім цього, Express містить додаткові таблиці, які можна використовувати спільно з Excel.

У Oracle існує два готові додатки. Це Financial Analyzer і Sales Analyzer. Вони використовують машину і кеш даних Express Server, і містять конфігурації звітів, розроблені для потреб фінансових аналітиків і аналітиків продажів. Вони корисні і користувачам, оскільки визначають важливі функції, що зазвичай беруть участь в обох видах аналізу, який реалізовано в Express Server.

Web Agent і Web Publisher - це засоби створення Інтернет-ресурсів, що надаються Express. Використовуючи будь-який з цих засобів, користувач може

створити динамічний і інтерактивний сайт, що забезпечується застосуванням різних можливостей OLAP. Web Agent в більшій мірі інструмент розробника, він є набором заздалегідь певних процедур, включених в Express Server SPL. Сайти можуть будуватися так само, як будуються призначені для користувача інтерфейси до бази даних. Для кінцевого користувача існує Web Publisher. Web Publisher зв'язаний з Express Analyzer і має можливість для створення власних інтерактивних сайтів тими, хто не має серйозного досвіду програміста. Web Publisher в основному є "майстром", який веде користувача через всі етапи побудови сайту і забезпечує графічний інтерфейс для підтримки його створення.

Arbor Software Corporation - це головний суперник Oracle. Її продуктом є Essbase, а найостаннішим його релізом був Essbase Server 5. Дуже популярний сервер Arbor для різних OLAP-продуктів, що говорить про те, що багато постачальників OLAP не обов'язково випускають повні додатки, а можуть використовувати як базу даних Arbor, і потім створювати інтерфейси до бази даних. Як приклад можна привести Comshare і Web-компоненту для Arbor - Crystalinfo, Seagate Software, що випускається. Останнім часом Arbor уклав партнерську угоду з IBM. Багатовимірне зберігання даних в Arbor Essbase Server буде замінено на DB2 від IBM. Передбачається, що це буде ROLAP-система, але фактично це не так. Це просто OLAP-система без однієї з кращих властивостей багатовимірних баз даних, але з перевагами системи РБД.

В доповнення до таблиць Essbase Spreadsheet Add-in, що забезпечують користувачів можливостями OLAP, Arbor пропонує WIRED for OLAP (засіб аналізу і презентацій), Crystal Info for Essbase (генератор звітів і розкладів) і SQL Drill-Through, що дозволяє користувачам проглядати подробиці даних в початкових реляційних базах. Arbor також випустив Arbor Essbase Adjustment Module. Цей додаток допомагає користувачам в підготовці звітів, що регулярно випускаються. Він сприяє автоматизації форматування звітів і процесів розрахунку. Крім того, існує ще Arbor Currency Conversion Module, здатний

конвертувати різні валюти в національну на основі моделі для відстежування обмінних курсів.

В області ROLAP визнаним лідером є компанія MicroStrategy. Їх філософією є відсутність обмежень на розмір сховища даних, так що немає жодних проблем з його збільшенням. Оскільки вони є виробниками реляційного OLAP, рівень їх аналітичних систем достатньо високий. У MicroStrategy немає OLAP-машини, яка могла б працювати локально, що незручно для користувачів, які часто працюють з ноутбуками або для тих, хто просто вважає за краще працювати автономно. Проте, у них існує продукт, DSS Broadcaster, що дозволяє посилати дані на різні вихідні пристрої. DSS Broadcaster посилає дані за запитом або коли відбувається певна подія. Наприклад, менеджерів може відсилатися щоденне оновлення з сумами прибутку за попередній день. Ця інформація може поступати по електронній пошті, на пейджер або мобільний телефон, а також факсом.

DSS Server є центральним продуктом в лінійці продуктів MicroStrategy. Це могутня машина, що дозволяє іншим агентам діставати доступ до реляційної бази даних в багатовимірному режимі. DSS Server містить різноманітні драйвери баз даних для оптимізації їх під необхідну реляційну базу даних (вони підтримують Oracle, DB2, Sybase, Red Brick, Informix, і інші реляційні бази). До того ж, акцент робиться на здатність їх зростання і включає драйвер для адаптації до дуже великих баз даних (Very Large Databases, VLDBs), розмір яких перевищує терабайти. Природа реляційного OLAP-продукту обмежує MicroStrategy в можливості надання дійсно індивідуального сервера для автономної роботи, проте за допомогою DSS Agent набір даних може завантажуватися і аналіз може виконуватися і в автономному режимі. DSS Agent є клієнт або клієнтський інструмент до DSS Server. Однією з переваг DSS Agent є використання інтелектуальних агентів для автоматизації бізнес-процесів. Наприклад, за допомогою DSS Agent можна створити агента, що знає, коли і де необхідно шукати дані і потім що з ними робити потім (тобто, як проводити їх очищення і куди їх помістити). Використовуючи агентів, можна

автоматизувати безліч звичайних, але часто повторюваних завдань. DSS Executive використовує можливості DSS Agent для реалізації високоякісного генератора звітів і засобу аналізу. Він використовує об'єктно-орієнтований підхід і інтерфейс, що працює за технологією "drag-and-drop" для швидкого створення додатків управлінських інформаційних систем силами самих користувачів. І, нарешті, MicroStrategy пропонує доповнення Excel Add-in, яке може використовуватися для додання таблицям функціональних можливостей OLAP.

Нове покоління Інтернет-орієнтованого OLAP від MicroStrategy представлене DSS Web 5.0. Однією з примітних властивостей DSS Web 5.0 є підтримка Microsoft webcasting standart. Це дозволяє автоматично передавати web-сторінки на комп'ютер користувача. У числі найважливіших можливостей DSS Web можна назвати здатність зберігати карти або діаграми, отримані з Інтернет, майстри звітів і пакети звітів, що настроюються.

Як приклад прикладного OLAP-продукту можна узяти Comshare. Не дивлячись на те, що це додаток, воно доповнює продукт функціональними можливостями OLAP. Comshare Decision проявляє гнучкість щодо використовуваного спільно з ним сервера. Arbor Essbase і Oracle Express - всі ці багатовимірні сервери баз даних можуть використовуватися спільно з Decision.

Hyperion Software - це також виробник прикладного OLAP-продукту, що випускає виключно OLAP-клієнти. Останнім продуктом був Hyperion MBA, або Multidimensional Business Analyst, що замінив HyperionOLAP. Згідно з останніми даними, Hyperion займав другий за величиною сегмент ринку. Природа продукту гарантує, що велика частина функціональних можливостей заснована на серверній базі даних. Hyperion популярний завдяки своїм додатковим аналітичним можливостям, що реалізуються у формі складних вимірювань, заздалегідь певних функцій і звітності. Метою його є формування могутнього фінансового пакету, що включає OLAP.

Hyperion пропонує два клієнтські OLAP-рішення. Перше, HyperionMBA, використовує OLAP для бізнес-аналіза. Як це зазвичай буває в

OLAP-додатках, Hyperion застосовує в своїх рішеннях складні вимірювання і заздалегідь певні функції для розрахунків і маніпуляцій з валютами. Програма Hyperion Analytic Accounting включає властивості OLAP в розрахункові пакет.

Cognos служить непоганим прикладом настільного OLAP-продукту. Це означає, що велика частина обробки проводиться не на сервері, а локально. Impromptu є інструментом запитів, використовуваним для витягання даних з багатовимірної бази даних. Дані потім поміщаються в Powerplay, яка зберігає куб даних на робочому столі комп'ютера користувача.

2.6. OLAP – системи та Інтернет - технології

Щоб ствердження про необхідність і повсюдне розповсюдження OLAP не здавалися голослівними, розглянемо декілька прикладів рішення задач за допомогою OLAP-інструментів великими зарубіжними підприємствами [66, 71].

Наш перший приклад - одна з найбільших американських енергетичних компаній, Duke Energy, чії активи перевищують 20 мільярдів, а число співробітників складає 22 000 чоловік, що працюють в різних підрозділах компанії по всьому світу. Сферою діяльності компанії є енергетичне забезпечення, обслуговування трубопроводів і постачання електричної енергії, природного газу і зрідженого природного газу. На момент впровадження OLAP компанія припускала вийти на нерегульований урядом енергетичний ринок і відвоювати собі певний шматок світового енергетичного бізнесу. В процесі підготовки до цього нового етапу відділ управління інформацією Duke Energy опрацьовував різні можливості застосування технологій для управління змінними інформаційними потребами підприємства і забезпечення йому гідного місця на ринку і відповідного розвитку. Питання полягало в забезпеченні своїх кінцевих користувачів доступом до інформації, що зберігається в корпоративних базах даних. Фахівці з управління інформацією

з'ясували, що в процесі ухвалення рішень всім підрозділам, починаючи з фінансових відділів і закінчуючи кадровими, важливо мати доступ до такої інформації тоді і там, де вона їм необхідна.

Система, яку Duke Energy використовувала до цього моменту, була складною в плані використання і без серйозного втручання фахівців з управління інформацією не забезпечувала необхідний підприємству рівень точності і своєчасності даних. Плануючи апгрейд систем підтримки прийняття рішень так, щоб співробітники фінансових і інші нетехнічні підрозділи компанії могли якнайповніше використовувати корпоративні дані, Duke Energy ухвалила рішення про впровадження OLAP-системи. Була вибрана система підтримки прийняття рішень від Business Objects. В результаті її впровадження, дані, що знаходяться в оперативній базі PeopleSoft і вітрині даних IBM DB2 HRMS стали доступні для OLAP-аналізу і різноманітних запитів і звітів різних підрозділів компанії. Рішення використовує RDTs (Шаблони швидкого розгортання, Rapid Deployment Templates), що містять зразки звітів, які користувачі без спеціальних технічних знань можуть налаштовувати на свій розсуд для генерації незапланованих запитів до даним і звітів.

В даному випадку компанія визначила, що пропозиція Business Objects найкращим чином відповідає її потребам. Але це не означає, що інші рішення в чомусь гірше вибраного. Кожне підприємство, що ухвалило рішення про впровадження OLAP, в першу чергу розглядає оптимальність пропозицій різних постачальників стосовно своїх власних проблем і потреб. Наприклад, та ж фірма Cognos успішно поставляє свої продукти таким серйозним клієнтам, як Міністерство оборони США і компанії Boeing. Будучи клієнтом Cognos з 1996 року, Міністерство оборони постійно оновлює наявні системи, доповнюючи їх новими можливостями і розробками. У листопаді минулого року фінансовим підрозділом Міністерства (DFAS) була придбана система бізнес-репортінга і аналізу з новими можливостями візуалізації, вартістю 1 мільйон доларів. Нове програмне рішення об'єднує дані з безлічі розрізнених систем, роблячи їх доступними для тисяч співробітників DFAS і інших агентств Міністерства

через захищений Інтернет-портал. «Візуалізатор» перетворить складні дані в зрозумілу зручну для використання інформацію, відображаючи її з використанням багатой графіки, різноманітних презентаційних можливостей і функцій оцінки даних. Роберт Еш, перший віце-президент Cognos, називає дане рішення «ядром електронного Уряду (e-government)», що забезпечує федеральним агентствам можливість використовувати Інтернет для спрощення і прискорення доступу до необхідної їм інформації. Впровадження OLAP-технології дозволило створити таку картину для кінцевих користувачів, що дістали у результаті можливість сформулювати поглиблене уявлення про роботу Агентства, що у свою чергу, сприятиме поліпшенню управління і ухвалення рішень відповідними співробітниками.

Перед Boeing Company стояла декілька інше завдання. Оскільки всі літаки зібрані на замовлення і включають до декількох мільйонів різних компонентів, співробітникам було потрібне серйозне аналітичне рішення для обробки даних таких істотних об'ємів. В рамках спеціальної програми по створенню інфраструктури для репортінга у сфері прийняття рішень в підрозділі комерційної авіації почато впровадження продукту Cognos Impromptu Web Reports, за допомогою якого 11500 співробітників підрозділу зможуть генерувати через Інтернет звіти до бази даних по накладним і специфікаціям на компоненти літальних апаратів. Керівники Boeing Company вважають, що нове рішення заощадить значну кількість людино-годин, що раніше витрачалися на створення таких звітів вручну. Всі разом це закономірно сприятиме поліпшенню процесу прийняття рішень.

Компанія MicroStrategy, як і більшість лідерів цього ринку, також пропонує спеціалізовані рішення для окремих областей бізнесу - фінансів, страхування, охорони здоров'я, урядових організацій і ін. - на базі MicroStrategy Business Intelligence Platform. Одне з таких рішень знайшло застосування в області енергетики. Компанія KeySpan, найбільший на північному сході США постачальник природного газу, що має підрозділи в Брукліне, Бостоні і на Лонг-Айленде, керівник ряду сервісних енергетичних компаній і більш ніж 13

000 співробітників, використовує MicroStrategy Narrowcast Server для забезпечення Нью-Йоркських домовласників і підприємств достатньою кількістю енергії для опалювання і підігріву води. Narrowcast Server, що є ключовим компонентом MicroStrategy Business Intelligence Platform, надає користувачам могутні аналітичні можливості і інтелектуальну систему попередження. Зокрема, система посиляє попередження про зниження або підвищення витрати газу щодо проектного на відповідні адреси електронної пошти і пейджерів. Клієнти KeySpan, «продавці», що поставляють газ кінцевим користувачам, можуть підключитися до інформаційних ресурсів компанії для перегляду і аналізу проектного і реального використання газу і ухвалення обґрунтованих рішень щодо об'ємів газу, необхідних для передачі по трубопроводах в кожен конкретний день. KeySpan використовує технології MicroStrategy для аналізу таких чинників, як історичні дані і погодні умови, щоб спланувати об'єми постачань газу. Narrowcast Server чотири рази на день порівнює фактичний потік газу із запланованим раніше того ж дня і посиляє звіт про непланові ситуації по електронній пошті або пейджеру відповідному клієнтові KeySpan і оперативним підрозділам, контролюючим вентиля трубопроводу. Таким чином компанії-постачальники можуть оптимізувати об'єми газу, що поставляється споживачам залежно від ряду різних чинників.

Одним із сучасних напрямків розвитку систем інтелектуального аналізу даних є об'єднання OLAP з технологією Data Mining і сховищами даних. Ці всі три технології розвиваються у міру того, як компанії починають усвідомлювати цінність даних. Реляційні бази даних свого часу були революційним рішенням, яке дозволило підприємствам збирати дані з щоденних транзакцій у великомасштабні засоби зберігання. За допомогою SQL було можливо виконувати елементарний аналіз цих даних. Коли ж був потрібен складніший аналіз, з'ясувалося, що SQL і РБД зовсім не ідеальне рішення. Таблиці були в змозі забезпечувати гнучкіший аналіз, але мали ряд істотних недоліків. Дані, що підлягають імпорту в таблицю з бази даних і сама таблиця були не в змозі ефективно оперувати великими об'ємами даних. З часом все більше і більше

компаній почали реалізовувати сховища даних і застосовувати до своїх даних засоби Data Mining. Сховище даних забезпечує зберігання очищених корпоративних даних. Дані по транзакціях перевіряються на коректність, категоризуються і потім поміщаються в сховищі. Інструменти Data Mining дозволяють аналітикам підприємств виявити приховані тенденції даних. Інструмент OLAP дає можливість виконувати швидкий і простий аналіз даних. В цілому, користувач-аналітик має уявлення про те, що він збирається знайти в деякому представленні даних. Він просто хоче мати засіб маніпулювання даними щоб найнаочніше відобразити деякі їх аспекти.

Іншим напрямком розвитку таких систем широке застосування OLAP в корпоративних порталах. Корпоративний портал - це «точка доступу», яка забезпечує зовнішніх і внутрішніх користувачів єдиним, безпечним мережевим інтерфейсом з персоніфікованим контентом. Контент слід розуміти в найширшому сенсі, як все, що об'єднує портал в одному призначеному для користувача інтерфейсі: додатки, інформація і інструменти спільної діяльності. До додатків відносяться внутрішні або зовнішні операційні або аналітичні програмно-апаратні засоби. Інформація є структурованими даними, результатами аналізу (звіти, куби, графіки, таблиці і т. п.), внутрішнім або зовнішнім неструктурованим контентом (документи, вміст цифрових накопичувачів, статистику відвідин сайту). Інструменти спільної діяльності - це веб-чати, мережеві конференції, електронна пошта, служби миттєвих повідомлень і т.п. (рис. 2.20).



Рис. 2.20. Корпоративний OLAP портал.

Кажучи про місце OLAP-технологій в порталі, буде цікавим навести точку зору фахівців з консалтингової компанії Object Systems Group. На їх думку, сфера застосування OLAP-інструментів в широкому сенсі - це «просунутий» бізнес-аналіз (Business Intelligence). Ці інструменти можуть звертатися до даних різних об'єктів, співвідносити їх, а також здійснюють пошук інформації (з тематики) в масштабах, непідвладних більшості користувачів, у випадку якби довелось робити те ж саме вручну. На сьогоднішній день, більшість компаній вже мають в своєму розпорядженні багато з компонентів, необхідних для впровадження OLAP-порталів. Ніхто не стане сперечатися, що можливість витягнути будь-яку бізнес-інформацію, задавшись питанням «як це співвідноситься з тим, чим я займаюся?» - могутній

інструмент підтримки прийняття рішень, про який користувачі можуть тільки мріяти.

Важливим напрямком розвитку систем інтелектуального аналізу даних, які набуває широкої популярності є поєднання OLAP – засобів та Web – технологій. Динамічні технології, поява яких стала можлива в результаті розвитку World Wide Web, є прекрасною альтернативою традиційним клієнт-серверним OLAP-методам. За останній час з'явилися цілий ряд OLAP-засобів (їх називають WEB-OLAP або WOLAP), оснащених Web-можливостями. Вони виконують аналітичні функції, такі як агрегація і деталізація (drill-up і drill-down), а також забезпечують високу продуктивність у поєднанні зі всіма перевагами, які дає Web-додаток. Поява Web OLAP-засобів стирає межі, що відокремлюють OLAP-ринок від суміжних категорій програмного забезпечення. Web-платформи інтерактивної звітності по своїй функціональності все більше і більше схожі на стандартні Web OLAP продукти. Більшість WEB-OLAP додатків використовують загальну архітектуру, в якій клієнтський браузер взаємодіє з HTTP-сервером, що пересилає HTML-сторінки. Але крім цього надається ще і проміжне ПО, таке, що зберігається на сервері. Такий компонент може безпосередньо зв'язуватися з Web-браузером або взаємодіяти з HTTP-сервером, який потім повертає браузеру HTML-сторінки з додатковими даними.

WEB-OLAP компонент проміжного рівня виконує набір функцій, які не може забезпечити HTML, а саме:

- взаємодія з базою даних, де знаходиться сховище;
- зберігання станів (попередніх транзакцій бази даних);
- обчислення і буферизація даних, що повертаються на клієнт.

На сьогоднішній день реалізовано декілька різних рішень WEB-OLAP, у тому числі на основі технологій HTML (DHTML), Java, ActiveX, а також їх комбінацій. Розглянемо основні типи таких програмних продуктів:

- HTML (DHTML) - рішення.
- HTML з розширеннями – CGI.

- HTML з використанням Java-апплетів.
- Java або ActiveX - компоненти.

HTML – рішення. Для реалізації OLAP-функціональності в Web-браузері використовується лише HTML. Простим прикладом такого рішення є OLAP-інструмент, що дозволяє користувачеві виконувати обумовлені OLAP-запити або звіти з браузера. В цьому випадку для здобуття даних з автономних OLAP-машин використовуються планувальники, що формують статичні HTML-звіти по HTML-шаблонам. Шаплони створюються так, щоб всі звіти мали погоджений вигляд. Звіти обробляються і передаються в браузер за допомогою Web-сервера. Статичні звіти характеризуються гарною переносимістю, швидко доставляються в браузер, при цьому взаємодії користувача з браузером практично не відбувається. В деяких випадках імітується перехід по вимірах шляхом навігації по звіту. Планувальник може створити набір звітів, зв'язаних між собою гіперпосиланнями. Наприклад, користувач, клацнувши по посиланню під назвою «Третій квартал», перейде до іншого звіту, що містить дані за липень, серпень і вересень.

Існує і інший підхід, коли OLAP-сервер наповнює HTML-шаблон даними в оперативному режимі, тобто по мірі появи запиту користувача через браузер. В цьому випадку на Web-сервері зберігаються лише шаблони звітів і метадані. Ці метадані містять інформацію, необхідну Web-серверу для передачі тих або інших даних в HTML-файл перед тим, як відправити його до браузера.

При будь-якому способі зберігання метаданих шаблонів і звітів інформація збирається Web-сервером згідно коду звіту, що посилається з браузера. Програмне забезпечення Web-сервера використовує метадані звіту, аби витягує відповідні дані з бази. База може зберігатися як на тому ж комп'ютері, що і додаток Web-сервера, так і на іншому. Отримані з бази дані об'єднуються на основі шаблону в звіт і передаються в браузер. Як правило, звіт вже містить за умовчанням певну OLAP-функціональність. При взаємодії із звітом призначений для користувача код посилається разом з іншою інформацією на Web-сервер, що використовує цей код для відстежування

інформації, яку користувач бачить в браузері. Повний набір OLAP-функцій (обертання, агрегація і поглиблення в дані) при такому підході не передбачений.

HTML з розширеннями – CGI. Одне з найслабших місць HTML – неможливість зберігати стану. Пропонований варіант вирішення проблеми – використання CGI (Common Gateway Interface – загальний шлюзовий інтерфейс) або інших Web API (Application Programming Interface - інтерфейс прикладного програмування) для реалізації єднального ПО (middleware). За допомогою цього методу можна забезпечити зберігання станів, а також буферизацію рядків і виконання деяких обчислень над переданими даними. В разі використання такої архітектури переносимість для клієнтських платформ забезпечується за рахунок використання HTML як інтерфейса. Проте і в цьому випадку використання лише HTML накладає обмеження на інтерфейси. Правильне відображення графіків і звітів буде утруднено. Ця проблема вирішується за допомогою Java-апплетів, які володіють великими можливостями по управлінню виведенням інформації на монітор.

HTML з використанням Java-апплетів. Спільне використання HTML і Java-апплетів дозволяє створити відмінне рішення WEB-OLAP. В цьому випадку HTML застосовується для відображення меню і виконання простих інтерфейсних функцій, а за допомогою Java-апплетів забезпечуються складніші компоненти інтерфейсу програми, а також погоджене відображення діаграм, графіків і таблиць. У таких застосуваннях найчастіше реалізовані функції агрегації і деталізації, а також обертання даних. За рахунок широких графічних можливостей Java в порівнянні з HTML, можна представляти дані у вигляді діаграм і змінювати їх в інтерактивному режимі.

Java або ActiveX - компоненти. Наступний підхід передбачає використання Java або ActiveX - компонентів для мінімізації взаємодії між браузером і Web-сервером, а також розширення можливостей користувача по роботі з даними за рахунок якіснішого інтерфейсу. Існує два способи вживання цих компонентів.

У першому випадку Web-сервер заповнює файл даними для звітів, і інтерфейсні компоненти посилаються в браузер разом з відповідним HTML-файлом. Браузер викачує цей файл, а компонент завантажує дані. Таким чином, компоненти забезпечують виконання таких OLAP-функцій, як агрегація, деталізація і обертання за допомогою зручного інтерфейсу, без звернення до сервера.

У другому випадку, інтерфейсний компонент безпосередньо з'єднується з сервером, який в інтерактивному режимі передає дані на клієнт по мірі появи запитів користувачів. В цьому випадку інтерфейсний компонент запрошує дані з Web-сервера, відкриваючи HTTP-потік. Потім, отримавши результати з сервера, він аналізує отриману інформацію і передає дані на об'єкт для відображення.

Історично склалося так, що інтелектуальний аналіз даних (Data Mining) і онлайн аналітична обробка інформації (OLAP) були виключно прерогативою людини, тобто саме люди визначали і створювали аналітичні моделі, а потім використовували отримані з їх допомогою результати. Але з появою обчислювальної моделі Web-служб, які розглядаються як універсальний засіб об'єднання різнорідних систем, картина істотно змінилася, аналітику тепер можна легко пов'язати з іншими обчислювальними завданнями. Іншими словами, люди перестали бути єдиними творцями або споживачами аналітичних сервісів. Як ви розумієте, це відкриває захоплюючі можливості.

Що ж таке аналітична взаємодія «без людей»? Візьмемо, наприклад, «інтелектуального агента», обслуговуючого B2B-обмін в процесі пошуку партнерів по операції. Окрім іншого, пошук передбачає кількісне визначення ризиків - процес, аналогічний оцінці ризиків при видачі кредиту. Кінцеве завдання агента — відповісти на питання, чи годиться партнер X для включення в операцію. Агент визначає важливі критерії (змінні і алгоритми) в запиті на оцінку ризику. Цей запит може включати обчислення тимчасових рядів, ранжирування і схожу на Data Mining процедуру оцінки невідомих параметрів партнерів, але автором і споживачем запиту є агент, а не людина.

Більш того, інтелектуальний агент повинен володіти реальною аналітичною «майстерністю»: на підставі результатів одного запиту створити другий набір запитів, на підставі другого — третій і так далі. З сторони API це нагадує інтерактивний сеанс, а з точки зору користувача — систему з власним «інтелектом». До недавнього часу можливості організації взаємодії аналітичних «движків» були обмежені. Розробникам був доступний ряд створених виробниками API, а також технологія Microsoft OLE DB for OLAP (або ODBO), яку окрім Microsoft підтримували і інші сервери. Але протокол ODBO доступний лише в Win32, що навряд чи могло зробити його універсальним засобом доступу для Web-служб або корпоративних додатків середнього рівня. Java-версію створеного Радою з OLAP (OLAP Council) інтерфейсу MD-API не реалізував жоден постачальник серверних або клієнтських продуктів, тому для серверів додатків на основі відмінних від Win32 технологій мультивендорні API взагалі відсутні. Проте з'являються нові можливості. Перший інтерфейс називається XML for Analysis (XML/A), інший інтерфейс — Java OLAP Interface (JOLAP). Перша перевага цих інтерфейсів — підтримка додатків середнього рівня, створених не на Win32. Зрозуміло, що оба інтерфейси годяться і для інших цілей. Так, для клієнтських застосувань ці API як мінімум забезпечують доступ до раніше недоступній підмножині серверів.

XML/A — це API для доступу до інформації. Як і в OLE DB для OLAP, клієнт може запрошувати багатовимірну інформацію у вигляді набору записів реляційних даних, а також багатовимірних наборів кліток. Проте специфікації досить відкриті і забезпечують підключення інших типів провайдерів і доступ до них без порушення нормативів.

JOLAP заснован на Java, і тому всі аспекти цього API об'єктно-орієнтовані. Важливо, що не визначено жодної текстової мови запитів - замість цього в об'єктній моделі присутні класи, які можна комбінувати для визначення вибірки і розміщення результатів.

Таким чином, закономірною є думка багатьох дослідників та розробників OLAP – систем, що нове покоління засобів інтелектуального

аналізу даних розробляється саме для мережевих користувачів і оптимізовано для застосування в Інтернеті (у зв'язку з чим говорять про Інтернет - аналітику). Ці засоби дозволяють забезпечувати даними і інструментами їх аналізу користувачів мережі, для чого застосовуються особливі технології зберігання багатовимірних даних. До теперішнього часу практично всі провідні постачальники засобів аналізу даних пропонують системи з тонким клієнтом, орієнтовані на роботу в Інтернеті. У деяких з цих продуктів тонкий клієнт використовується для доступу до даних «традиційних» OLAP-серверів, в інших же - для обробки багатовимірних кубів, розміщених на Web-сайтах. Такі куби можна викачати з сайту або переслати по електронній пошті і помістити на локальний або мережевий диск. Об'єм даних, поміщених в мікрокуб стискається приблизно в 40 разів. Ефект досягається за рахунок нетрадиційних для OLAP підходів до зберігання даних. По-перше, в мікрокубах не розміщуються заздалегідь підраховані агрегати. Вони замінюються спеціальними механізмами індексування, які дозволяють обчислювати агрегати «на льоту». По-друге, значення всіх вимірювань зберігаються в мікрокубах в єдиних екземплярах. Нарешті, весь їх вміст архівується.

Одним з найвідоміших прикладів реалізації технологій інтелектуального аналізу даних в Інтернеті є проект провідного американського виробника засобів аналізу компанії Cognos для технологічної біржі Nasdaq. Російський аналог цього рішення - сервісний аналітичний центр для комерційних банків Banklist.ru, що використовує ПО компанії Intersoft Labs.

Рішення Nasdaq & Amex Companies Edition надає аналітичний сервіс учасникам біржі Nasdaq. Підписчики Nasdaq регулярно по електронній пошті отримують локальні куби з біржовими даними. У цих кубах в розрізі індексу Nasdaq, ринкової вартості компаній, річних доходів і інших вимірювань зберігаються показники ліквідності, прибутку і інші дані про діяльність понад 5000 компаній із списку Nasdaq і Amex, акції яких є предметом біржового торгу. Біржові дані поміщаються в куби за допомогою генератора Cognos Powerplay. Для OLAP-аналізу кубів і випуску звітів підписчикам надається

спеціально конфігурований для роботи з кубами OLAP-клієнт Cognos і пакет OLAP-звітів для перегляду і аналізу кубів.

Мета проекту Banklist.ru - забезпечити російським банкам, підприємствам і іншим зацікавленим особам і організаціям швидкий доступ до достовірної фінансової звітності комерційних банків і можливість її аналізу. Це необхідно для підвищення довіри до російських банок як з боку їх клієнтів, так і з боку діючих і потенційних партнерів. Проект створений за ініціативою Асоціації російських банків і реалізований під патронатом ЦБ РФ. Готове рішення організоване як Web-сайт, база даних якого містить мікрокуби аналітичної платформи "Контур". У цих мікрокубах зберігається фінансова інформація: баланси, звіти про прибутки і збитки і розрахунок обов'язкових економічних нормативів діяльності кредитних організацій. Вимірюваннями кубів є регіони, банки, балансові рахунки 1-го і 2-го порядку, статті доходів і витрат, банківські нормативи, а показниками - розгорнені залишки по рахунках, суми доходів, витрат і нормативів. Достовірна звітність 733 російських банків поступає з офіційного сайту ЦБ РФ.

Мікрокуби створюються генератором "Контур CubeMaker". Підписчикам сервісу Banklist.ru надається аналітична складова - OLAP-клієнт "Контур OlapBrowser". Це спеціальний web-браузер для OLAP-аналізу, який забезпечує доступ до мікрокубів через Інтернет і в локальній мережі, перегляд і аналіз даних OLAP-кубів, випуск звітів. З його допомогою можна знайти і відкрити для аналізу будь-який мікрокуб безпосередньо з Web-сайту або з диска призначеного для користувача ПК.

Обидва розглянуті рішення відносяться до класу OLAP-клієнтів з локальним кубом, тобто забезпечують відкладений (off-line) аналіз одного разу підготовлених файлів з даними на ПК користувача. Куби для аналізу можуть зберігатися на Web-сайті (варіант Banklist.ru) або на ПК користувача, як в рішенні для Nasdaq.

Принципова відмінність в технологічній реалізації цих проектів - розміщення метаданих. У разі Nasdaq користувачі отримують персональну

OLAP-систему, жорстко конфігуровану для роботи з PowerCube Nasdaq. Метадані зберігаються у файлах OLAP-звітів, налаштованих на куби Nasdaq. Звіти встановлюються разом із спеціальною запускаючою програмою Cognos Special Edition Launcher і OLAP-клієнтом Cognos Powerplay Special Edition. Це так званий «спеціальний випуск» продуктів Cognos, розроблений для проекту Nasdaq. Передвстановлена конфігурація робочого місця користувача системи аналізу біржових даних Nasdaq є істотним плюсом цього підходу.

«Контур»-мікрокуб містить метадані (опис вимірювань і показників, механізму агрегації і сортування даних, аналітичних інтерфейсів і т. д.) в самому собі. Такий підхід універсальний: зміна складу і даних мікрокубів не вимагає оновлення клієнтського ПО. Один і той же OLAP-клієнт може застосовуватися для аналізу OLAP-кубів різних прикладних сервісів. Це створює передумови для тиражування технології в різних галузях і знижує вартість володіння нею для користувачів.