УДК 004.652.4

А.Ю. Берко, В.А. Висоцька
Національний університет "Львівська політехніка",
кафедра інформаційних систем та мереж

ЗАСТОСУВАННЯ OLAP-ТЕХНОЛОГІЇ В СИСТЕМАХ ЕЛЕКТРОННОГО БІЗНЕСУ

Описано основні архітектури сховищ даних у системах електронного бізнесу, розглянуто деякі загальні принципи їхньої побудови. Докладно описано способи звітування в системах електронної комерції за допомогою OLAP-систем. Проаналізовано основні проблеми побудови OLAP-систем електронної комерції та запропоновано методи вирішення цих проблем.

In the given article main problems of electronical business are analyzed. New methods for solution of discussed problems are proposed.

1. Вступ

Бурхливий розвиток електронного бізнесу спричинив підвищення вимог щодо інформаційних систем електронної комерції, які мають:

- Аналізувати інформацію в тимчасовому аспекті;
- Формувати довільні запити до системи;
- Опрацьовувати більші обсяги даних;
- Інтегрувати дані з різних систем.

На жаль, у більшості систем електронної комерції інформація актуальна тільки на момент звертання до бази даних, у наступний момент часу за тим же запитом можна одержати зовсім інший результат. Інтерфейс таких систем розрахований на проведення певних жорстких операцій і можливості одержання результатів на нерегламентований (ad-hoc) запит сильно обмежені. Можливість опрацювання більших масивів даних має також обмеження через налагодження СКБД на виконання коротких транзакцій і неминучого вповільнення роботи інших користувачів.

Виникнення цієї потреби спричинило появу нової технології організації баз даних – технології сховищ даних. Існує три найпоширеніші підходи до побудови сховищ даних.

1. Віртуальне сховище даних – це система, що подає інтерфейси й методи доступу до системи, які стимулюють роботу з даними в цій системі як зі сховищем даних. Віртуальне сховище даних можна організувати, створивши ряд запитів (view) у базі даних або застосувавши спеціальні засоби доступу, наприклад продукти класу Desktop OLAP, до яких належать зокрема BusinessObjects, Brio Enterprise та інші.

Головними перевагами такого підходу є:

- Простота й мала вартість реалізації;
- Єдина платформа із джерелом інформації;
- Відсутність мережевих з'єднань між джерелом інформації й сховищем даних.

Однак недоліків у нього набагато більше, ніж переваг. Створюючи віртуальне сховище даних, отримуємо не сховище як таке, а його ілюзію. Структура зберігання даних і саме зберігання даних не терпить змін, і залишаються проблеми:

- Продуктивності;
- Трансформації даних;
- Інтеграції даних з іншими джерелами;
- Відсутності історії;
- Чистоти даних;

• Залежності від доступності основного БД;

• Залежності від структури основного БД.

2. Дворівнева архітектура сховища даних передбачає побудову вітрин даних (data mart) без створення центрального сховища, при цьому інформація надходить із невеликої кількості систем, які реєструють, і обмежена конкретною предметною галуззю. Будуючи вітрини даних за основними принципами побудови сховищ даних, їх можна вважати сховищами даних у мініатюрі. Плюсами вітрин даних є:

• Простота й мала вартість реалізації;

• Висока продуктивність за рахунок фізичного поділу аналітичних систем, що реєструють, виділення завантаження й трансформації даних в окремий процес, оптимізований під аналіз структури зберігання даних;

• Підтримка історії;

• Можливість додавання метаданих.

3. Побудову повноцінного корпоративного сховища даних зазвичай виконують у трирівневій архітектурі. На першому рівні розташовані різноманітні джерела даних – внутрішні системи, що реєструють, довідкові системи, зовнішні джерела (дані інформаційних агентств, макроекономічні показники). Другий рівень містить центральне сховище даних, куди стікається інформація від всіх джерел з першого рівня і, можливо, оперативний склад даних (ОСД). Оперативний склад не містить історичних даних і виконує дві основні функції. По-перше, він є джерелом аналітичної інформації для оперативного керування й, по-друге, тут готують дані для подальшого завантаження до центрального сховища. Під підготовкою даних розуміють їхнє перетворення й здійснення певних перевірок. Наявність ОСД просто необхідна за різного регламенту надходження інформації із джерел. Третій рівень в описуваній архітектурі являє собою набір предметно-орієнтованих вітрин даних, джерелом інформації для яких є центральне сховище даних. Саме з вітринами даних і працює більшість кінцевих користувачів.

2. Зв'язок висвітленої проблеми із важливими науковими чи практичними завданнями

Використання сховища даних у банківській системі фінансового керування

АБС + Data Warehousing + OLAP = Повна система фінансового керування для банків. Автоматизовані банківські системи (АБС) електронного бізнесу являють собою системи оперативної обробки транзакцій (OLTP – on-line transaction processing), оптимізовані під виконання банківських операцій і містять деякі засоби аналізу й одержання звітів для керівництва. Функції ж аналізу й керування високого рівня – такі, як аналіз прибутковості – зазвичай реалізуються поза АБС.

Найчастіше використовують засоби й системи, розроблені третьою фірмою й виконані в іншій програмній технології й на інший СКБД. Природно, що в такій ситуації виникає безліч проблем, пов'язаних із забезпеченням прозорої й зручної для користувача процедури обміну даними, що не дає змоги ефективно використати обидві системи. Складність забезпечення ефективної взаємодії обумовлена, зокрема, тим, що банки почали процес автоматизації з установки АБС, а надалі нарощували її засобами аналізу й керування. Спектр продуктів, запропонований у цей час фірмами-лідерами комп'ютерного ринку, є прекрасною базою для створення інтегрованого рішення для банків, від АБС (OLTP-систем) до засобів підтримки прийняття рішень. А використовуючи засоби OLAP, за допомогою яких ефективно вирішують завдання аналізу, прогнозування й планування, можна розробити повну й закінчену систему банківського аналізу й керування.

У статті описано закінчену систему керування банком, що базується на інтегрованому підході з використанням концепції сховища даних (Data Warehouse) та оперативної аналітичної обробки (OLAP).

Опис підходу

АБС електронного бізнесу – це система OLTP, що автоматизує щоденні рутинні банківські операції. Структура даних АБС спроектована для швидкого й ефективного виконання елементарних дій, з яких і складаються банківські операції (увести проводку, укласти угоду, нарахувати відсотки й

т.п.). Вона відбиває структуру бізнес-процесів банку й забезпечує ефективні методи уведення, зміни й доступу до даних. Більшість даних, з якими працює АБС, – це поточні, "сьогоднішні" дані: для виконання операцій зазвичай потрібні тільки ці значення параметрів і лише до них забезпечується швидкий і зручний доступ. Модель даних АБС підтримує операційні процеси в банку й подає сутності бізнес-процесів; їхні атрибути зручно вводити й модифікувати, але не аналізувати. Набір атрибутів множинний, але не багатомірний: атрибути, що мають той самий зміст для аналітиків, можна подавати різними полями в таблицях, що описують сутності. Більше того, іноді для їхнього одержання виявляється необхідно провести обчислення. Наприклад, для банківських операцій на валютному або фондовому ринках або ж на ринку міжбанківських кредитів використовують різні формули обчислення прибутковості угоди. У повнофункціональних АБС ці операції проводять, як правило, у різних модулях. Тому одержання аналітичної характеристики "прибутковість за типами інструментів" вимагатиме додаткових зусиль.

З іншого боку, для аналітичної обробки необхідні регулярні багатомірні структури, а стандартні математичні методи (і пакети) використовують багатомірні матриці. Проведення статистичного аналізу вимагає доступу до великого обсягу даних. Час – найістотніший вимір, і зазвичай аналіз проводять з метою виявлення певних тенденцій: спочатку аналізують дані про минуле й сьогодення, а потім виявлену тенденцію екстраполюють на майбутній період часу. Отже, два типи діяльності – повсякденні операції банку й аналітична обробка – потребують як різних структур даних, так і різних процедур доступу й обробки інформації. Переважно операції банку й аналітичну обробку реалізують у двох окремих системах (пакетах), і для забезпечення взаємодії між ними потрібен інтерфейсний модуль. Найчастіше це програма, що здійснює односпрямоване (від АБС до аналітичної системи) перетворення й інтеграцію даних. Однак такий інтерфейсний модуль можна розглядати як необхідний "інтелектуальний" проміжний шар (middleware) між АБС і системою OLAP, що володіє декількома важливими властивостями систем підтримки сховища даних. Саме цей шар, набагато потужніший, ніж просто перетворювач даних, здатний істотно розширити можливості всієї системи, а також забезпечити користувачеві додаткові зручності. Цей «інтелектуальний» проміжний шар необхідний, зокрема, через розходження в природі об'єктів: прості в системі OLTP (АБС) і складні (складові агрегування) у системі OLAP. Для проведення високорівневого об'єктно-орієнтованого аналізу потрібно побудувати на основі елементарних понять АБС (рахунку, проводки й т.п.) складні об'єктно-орієнтовані об'єкти, необхідні для подальшої обробки в системі OLAP. За відсутності такого шару засоби OLAP подекуди працюватимуть намарно, оскільки аналітикові доведеться оперувати не фінансово-економічними, а обліково-бухгалтерськими категоріями.

3. Аналіз останніх досліджень і публікацій, в яких започатковано розв'язання проблеми і на які спираються автори

Наведемо перелік стандартних для сховищ даних операцій перетворення й інтеграції даних:

1. Завантаження даних з різних джерел: АБС, зовнішніх систем і т.п., зокрема консолідація даних з філій;

2. Перевірка й фільтрація даних (на щастя, ці процедури відносно прості, тому що всі дані належать одній і тій самій предметній галузі з обмеженою множиною понять і типів об'єктів);

3. Визначення метаданих (об'єктно-орієнтованих об'єктів);

4. Агрегація даних.

Однак крім стандартних операцій, в "інтелектуальному" шарі бажано мати:

- Інтерактивне визначення метаданих (об'єктів).
- Інтерактивний аналіз і контроль, моніторинг параметрів АБС і обумовлених користувачем показників.
- Визначення об'єктів, що залежать від часу.
- Зміна даних у сховищі.

Отже, у повній системі керування можна виділити три рівні; розходження між ними полягає в об'єктах, з якими ці рівні працюють, і виконуваних над ними функціях (див. рис. 1): АБС (система OLTP), проміжний рівень та система OLAP.

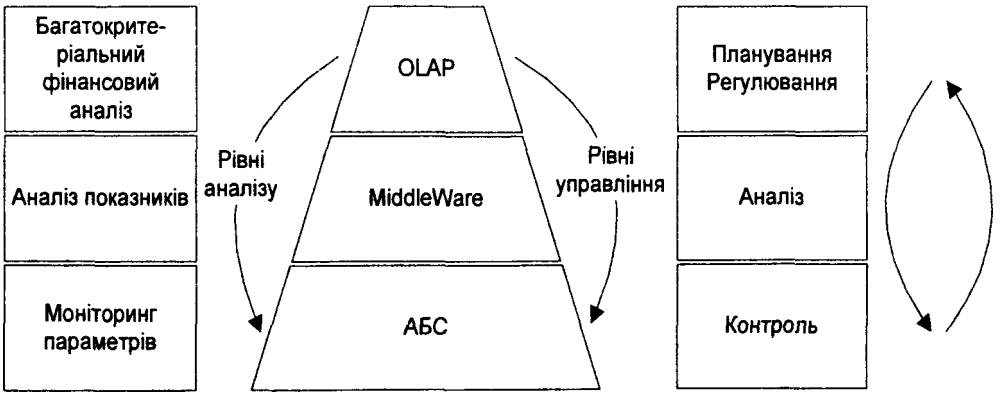

Рис. 1. Повна система фінансового керування для банків

Схема фінансового аналізу й керування

Перший рівень (АБС) оперує елементарними поняттями, такими як рахунок, проводка, угода й т.п. Цього понятійного апарата досить для виконання функцій бухгалтерського обліку й повсякденних банківських операцій. Поняття вищого рівня, такі як прибутковість за продуктом або прибутковість підрозділу, достатність капіталу банку й т.п., визначаються на другому рівні. Цей рівень є критично важливим для успішної реалізації всієї системи керування банком. Об'єкти, обумовлені на цьому рівні, можуть або забезпечувати консолідоване зведення елементарних даних, або являти собою нові, "неопераційні" поняття, які доступні для аналізу на цьому й наступному рівнях. Третій рівень (аналітичний інструментарій OLAP) містить потужні засоби перегляду даних, а також засоби моделювання й статистичної обробки. Для простих аналітичних запитів зазвичай достатньо таких засобів перегляду, як багаторівнева деталізація, знаходження найбільших і найменших значень і т.п.; у складніших випадках можна використати засоби моделювання й проведення аналізу.

Опис компонентів АБС

Хоча викладений підхід можна застосовувати до будь-яких АБС, для систем з розвинутою структурою даних він надає виняткові можливості для аналізу. (Зазначимо, що цей факт дуже часто не беруть до уваги: потужні аналітичні системи типу SAS запускають працювати над АБС, що не має таких понять, як угода, центр прибутку, банківські продукти тощо. Об'єктами аналізу стають всі ті ж проводки та залишки на рахунках. А від аналітика потребують відповіді на питання щодо прибутковості того або іншого фінансового інструмента...)

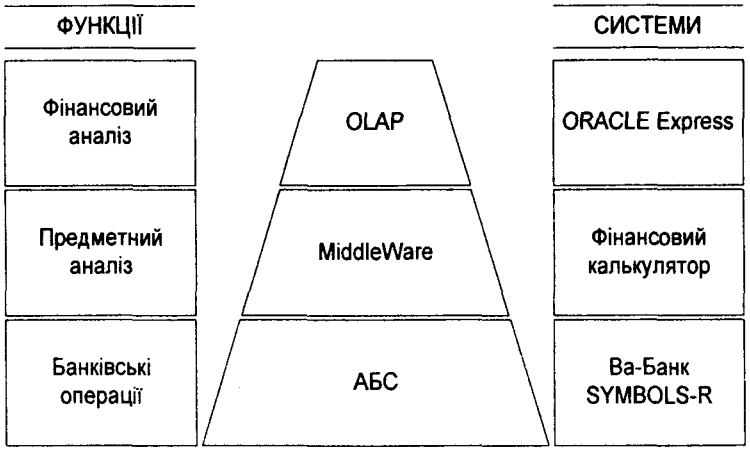

Рис.2. Схема фінансового аналізу та керування АБС

Проміжний рівень (middleware)

Загальні принципи організації цього шару, викладені нижче, застосовні загалом до будь-якої АБС.

Middleware – це засіб підтримки сховища фінансових даних (СФД). Крім стандартних функцій перетворення й інтеграції даних, він реалізує перераховані вище властивості "інтелектуального" шару, які не просто корисні, але й необхідні:

1. Інтерактивне визначення об'єктів;
2. Інтерактивний аналіз і контроль;
3. Визначення залежних від часу об'єктів;
4. Можливість вносити зміни в дані, що зберігаються.

Перша й друга властивості забезпечують інтерактивне завдання об'єктів, аналіз і моніторинг. Чому вони так важливі? Річ у тому, що багато показників мають складну структуру (формулу обчислення), але водночас винятково важливо контролювати їхнє значення протягом операційного дня. Основні показники такого роду переважно обчислюються в АБС і можуть бути проконтрольовані. Однак, щоб найефективніше керувати банком, необхідно розширити перелік таких показників. Використовуючи першу властивість, користувач може у будь-який момент визначити новий показник, а за допомогою другого – здійснювати моніторинг значень і вживати коригуючих дій.

Третя властивість забезпечує підтримку показників, описаних формулами, змінними в часі. Часто зустрічається ситуація, коли коефіцієнти формул і навіть самі формули розрахунку багатьох обов'язкових показників змінюються. Тому для забезпечення коректності звітів і результатів аналізу за тривалий проміжок часу необхідно мати засіб опису залежності формули від часу (історію зміни формули). З теоретичної точки зору потрібна ще одна розмірність: "тимчасова версія" об'єкта. Наявність таких засобів забезпечує додаткову гнучкість під час проведення аналізу.

Четверта властивість, важлива сама по собі для багатьох реалізацій сховищ даних, виявляється абсолютно необхідною, адже дає змогу вирішити проблеми заключних оборотів, виправлення помилок (будь-якої природи) і забезпечення правильного функціонування аналітичних пакетів за введення законів і правил обліку, що діють "заднім числом".

4. Виділення невирішених раніше частин загальної проблеми, котрим присвячується означена стаття

Ускладнюють аналітичну діяльність банків електронного бізнесу загалом такі проблеми:

- Дані розрізнені як на фізичному, так і на логічному рівні, тому що вони розподілені між різними автономними автоматизованими системами.
- Дані важкодоступні для аналізу, тому що вони зберігаються в структурах, не призначених для нестандартних запитів або аналітичних звітів.
- Дані не погоджені: інформація з тим же самим значенням зберігається по-різному в різних транзакційних базах даних, різні коди ідентифікації часто використовуються для тих же самих об'єктів, довідкові дані можуть змінюватися в часі, але ці зміни зазвичай не фіксуються в транзакційних системах.

Для усунення цих проблем необхідно створити інформаційно-аналітичну систему, засновану на сучасних інформаційних технологіях сховищ даних та інтерактивної аналітичної обробки.

5. Формулювання цілей статті

- Опис моделі централізованого сховища для інформаційного забезпечення різних аналітичних служб електронного банку;
- Розробка технології й автоматизація процесу збирання, відбору, узгодження й перевірки інформації, що надходить із різних джерел – систем оперативного рівня;
- Розробка технології й реалізації процедур доставки аналітичної інформації в оперативному режимі й у формі, зручній для сприйняття;
- Розробка оперативних процедур підготовки статистичних і аналітичних звітів відповідно до прийнятих стандартів;

Проектувати й розробляти інформаційну аналітичну систему електронного банку необхідно за такими принципами побудови:

- Поетапне впровадження з метою мінімізації ризику й керування складністю системи.
- Модульна структура програмного забезпечення, основний модуль якої містить всі обов'язкові компоненти, реалізує додаткові функціональні можливості – факультативні модулі.
- Відкритість рішення, необхідна для додавання як нових структур даних, так і нових функцій.
- Максимальна незалежність від діючих внутрішніх систем з особливою увагою до стандарту інтерфейсу обміну й автоматизацією процесу витягу даних і завантаження їх у сховище.

6. Виклад основного матеріалу дослідження з повним обґрунтуванням отриманих наукових результатів

Архітектура інформаційного сховища для систем електронного бізнесу

У базі даних інформаційного сховища перебувають основні дані, тобто дані предметної області, нормативно-довідкова інформація й службова інформація, необхідна для функціонування додатків інформаційного сховища (рис. 3).

До службової інформації належать:

- метадані, тобто опис наявності й повноти основних даних;
- календар банківських операційних періодів;
- адміністративні дані, необхідні для роботи внутрішніх завдань сховища.

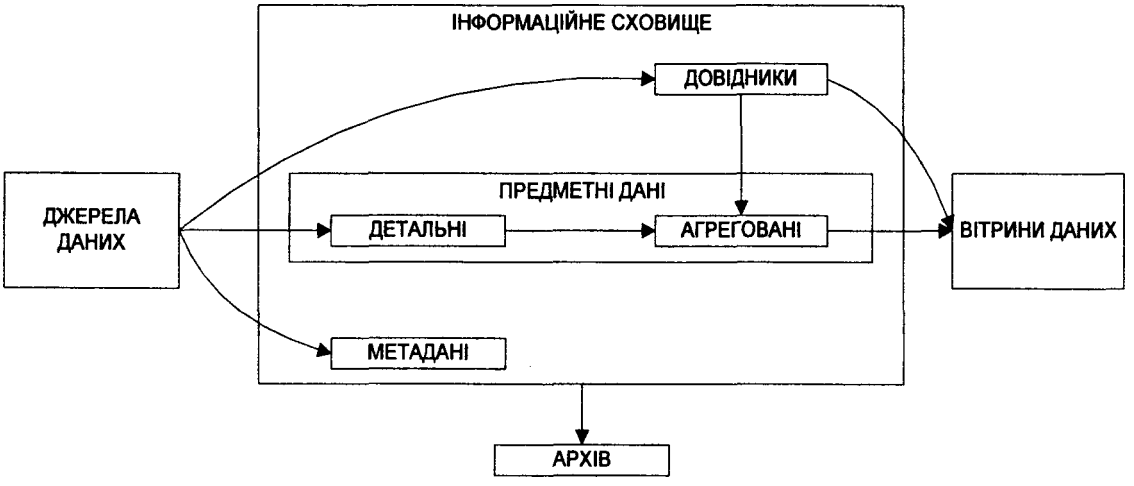

Рис. 3. Інформаційна модель сховищ даних

Метадані містять специфікації й описи структури предметної галузі, дані для адміністрування й супроводу. Спеціальну увагу приділяють інформації щодо завантажених предметних даних, тобто даних для рівня адміністрування й розробки.

Нормативно-довідкова інформація містить загальні класифікатори, централізовано поширювані в системі банку, і регіональні довідники.

Основними розділами інформації предметної галузі є:

- Інформація обліково-операційної системи про діяльності підрозділів;
- Звітність, надавана кредитними організаціями регіону й аналітична інформація на її основі,
- Інформація про курси валют і валютне регулювання.

Інформація організована у вигляді наборів даних за банківськими установами за операційні періоди. Агрегована інформація містить набори даних, отримані з детальних даних за допомогою заданих алгоритмів підсумовування, а також звіти, сформовані на регулярній основі й призначені для зберігання. Для ефективності роботи сховища доцільно виносити історичні детальні дані до архіву після закінчення строку їхнього активного використання. Найчастіше бувають затребувані детальні дані за останні два–три роки. За необхідності звертання до детальних даних архів повинен забезпечувати їхнє відновлення в прийнятний термін.

Робота сховища даних неможлива без автоматизованої системи виконання стандартних внутрішніх завдань: завантаження інформації, додаткової перевірки цілісності, агрегування тощо. Схему взаємодії серверних і прикладних завдань сховища наведено на рис. 4. Для стандартних завдань сховища характерно те, що їхнє виконання задається як за часом (наприклад, щодня), так і за наявності даних. Крім того, виконання одних завдань залежить від успішності виконання інших.

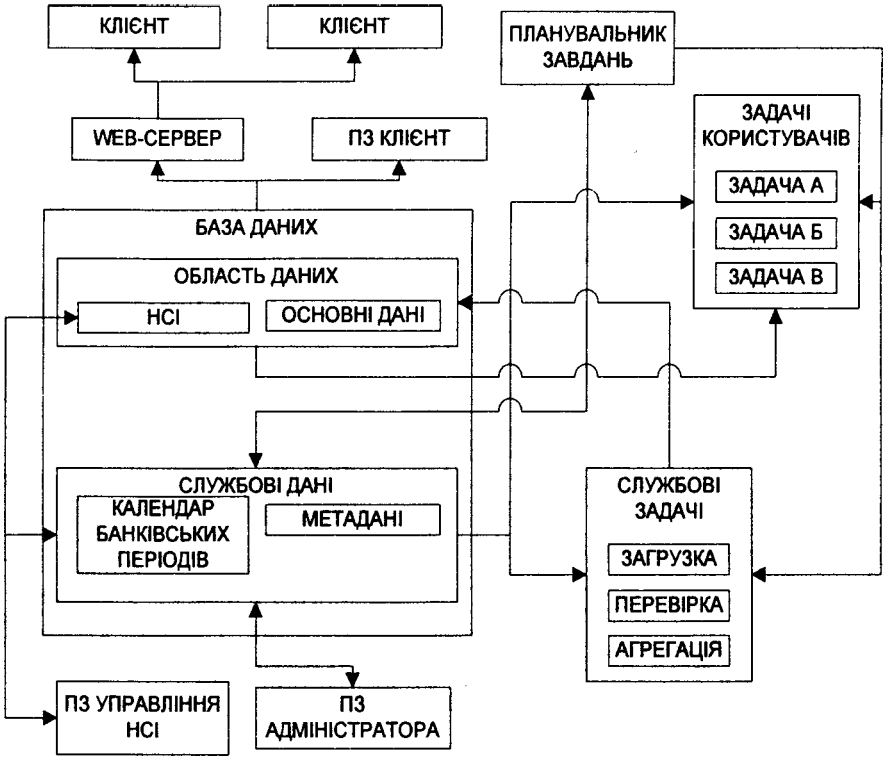

Рис. 4. Архітектура інформаційного сховища електронного бізнесу

Основою системи керування завданнями є ПЗ "Планувальник завдань". Його призначення – формування розкладу завдань на кожний календарний день (за заданими умовами старту) та їхній запуск на виконання. Більшість завдань пов'язана з появою тих або інших даних у сховищі. У випадку успішного завершення завдання Планувальник заповнює метадані про наявність і повноту певного розділу інформації сховища. Кожне завдання прив'язується до конкретного розділу даних адміністратором. Адміністратор також задає періодичність і додаткові умови старту завдання.

Інструментальні засоби аналізу

Наявність ІАС дає змогу аналітикам, економістам, адміністраторам використати інформацію, вивчати дані, приймати краще рішення й здійснювати те, що вони ніколи не були здатні робити колись. Кінцеві користувачі ІАС розділені на чотири групи:

- одержувачі звітів;
- аналітики;
- група супроводу довідкових даних;
- розроблювачі нових звітів і додатків.

Необхідно використати такі технології аналізу для інформаційної підтримки перших двох груп користувачів:

- Динамічна генерація звітів та їхня поставка кінцевим користувачам за WEB технологією.
- Вітрини даних для багатомірного аналізу.

Для супроводу довідників використовують як автоматизоване завантаження даних (централізовано поширювана інформація), так і "ручну" обробку за допомогою додатків, розроблених за технологією "товстого" клієнта.

Сховище даних дає змогу вирішувати практично будь-які завдання аналізу й керування бізнесом. Високої продуктивності цього рішення досягають за рахунок оптимізованої моделі даних фінансової галузі, для аналітиків доступні бізнес-метадані, набір інструментальних засобів керування сховищами даних, необмежена кількість оперативних і зовнішніх джерел даних. Загальну архітектуру рішення наведено на рис. 5.

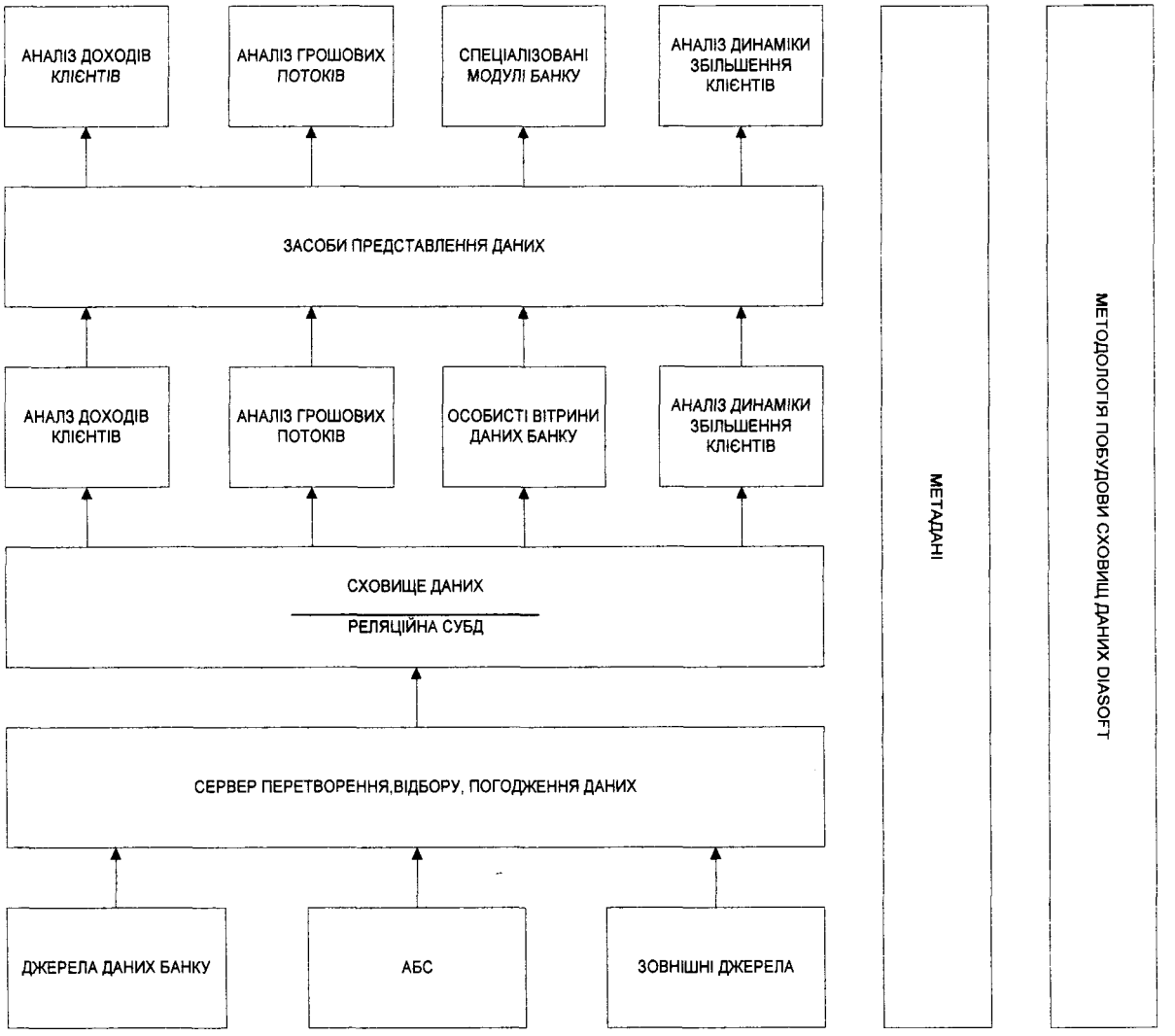

Рис. 5. Архітектура системи керування електронним бізнесом

Математична модель OLAP-системи для електронного бізнесу

Сьогодні в сховищах даних утримується безліч даних, які ніколи не використовуються додатками, виконуваними над цими сховищами даних, і ці непотрібні дані є однією із причин зниження ефективності виконання запитів. Потрібно забезпечити можливість реєстрації повного набору запитів, генерованих всіма додатками, і використання таблиць і полів, що фігурують у запитах, для тонкого налагодження вмісту сховищ даних. У сьогоднішніх сховищах даних для зберігання даних і керування ними значною мірою використовують OLAP-системи. Однак можливостей сьогоднішніх OLAP-систем недостатньо для обробки запитів, орієнтованих на сканування, таких як групування записів і обчислення агрегатів, а також операцій переміщення файлів, які переважають на етапі перетворення даних сховищ даних і етапі підготовки даних під час видобутку даних.

OLAP – динамічний синтез, аналіз та консолідація великих обсягів багатомірних даних, тобто системи аналітичної обробки в реальному часі (on-line analytical processing systems – OLAP-системи).

Нехай X – деякий скінченний набір багатомірних даних. Будь-яка підмножина $S \subseteq F(X)$ називається подією в наборі багатомірних даних X. Автоматною системою подій в наборі багато-

мірних даних X називатимемо будь-яке сімейство $\{S_i\}$ ($i \in I$, де I – деяка множина індексів) не порожніх подій у цьому наборі багатомірних даних X, таке, що

$$\bigcup_{i=1} S_i = F(X) \setminus \{e\} \text{ і } S_i \cap S_j = \varnothing, \text{ якщо } i \neq j.$$

Нехай $A = (A, X, f, a_0, F)$ – скінченна X – система з множиною станів $\{a_0, a_1, \ldots, a_n\}$, а S_{ij} – подія, яка складається із всіх таких багатомірних даних набору X, що переводить систему A із стану a_i до стану a_j. Для побудови регулярного виразу подій, наведеного в системі A, за рівністю $S(a_0, F) = \bigcup_{a_j \in F} S_{0j}$ достатньо побудувати регулярний вираз для подій S_{ij} $(i, j = 0,1,2,\ldots,n)$.

Сервери багатомірних баз даних на основі OLAP можуть виконувати перераховані нижче основні аналітичні операції:

• **Консолідація** – це такі операції, як просте підсумовування значень ("згортання") чи розрахунок з використанням складних виразів, які мають інші зв'язні дані. Наприклад, показники для відділів компанії можна додати з метою отримання показників для кожного міста, а показники міста – "згорнути" до показників за окремими країнами.

• **Спадний аналіз** ("drill-down") – операція, зворотна до консолідації, яка має відображення докладних відомостей для розглянутих консолідованих даних.

• **Розбиття з поворотом** (slicing and dicing). Ця операція (яка також називається створенням звідної таблиці) дає змогу отримати подання даних з різних точок зору. Наприклад, один зріз даних про прибуток може відображати всі відомості про прибутки від продажу об'єктів нерухомості вказаного типу за кожним містом. Інший зріз може подавати всі дані про прибутки відділів компанії в кожному із міст. Розбиття з поворотом часто виконують паралельно осі часу – з метою аналізу тенденцій та пошуку закономірностей.

Отже, засоби OLAP забезпечують широкі можливості побудови специфічних звітів для організацій, які потребують формування повсякденної звітності.

Динамічні технології, поява яких стала можливою у результаті розвитку "всесвітньої павутини" (World Wide Web), є прекрасною альтернативою традиційним клієнт/серверним OLAP-методам. Останнім часом з'явилися OLAP-засоби (їх називають Web-OLAP. або WOLAP), оснащені Web-можливостями. Вони виконують аналітичні функції, такі як агрегування і деталізація (drill-up і drill-down), а також забезпечують високу продуктивність у сполученні з усіма перевагами, що дає Web-додаток.

З такими додатками значно полегшується задача установки, конфігурування і розгортання. Web-додаток виконують на сервері, тому на клієнтській машині потрібний тільки Web-браузер і під'єднання до Intranet/Internet. Подібна стратегія розгортання особливо зручна для адміністраторів сховищ даних, яким часто доводиться працювати із широким контингентом вилучених користувачів (наприклад, із продавцями або службовцями відділів, розташованих у віддалених офісах), що дуже непросто за використання традиційної клієнт/серверної архітектури.

Використання Web-технологій впливає і на вартість BI-додатків:

• істотно знижуються витрати на розгортання програмного забезпечення в розрахунку на одного користувача;

• компанії можуть скористатися вигіднішими ціновими моделями, оплачуючи використання продукту (яке визначається кількістю одночасно працюючих користувачів), поза залежністю від числа робочих місць або розміру сервера.

Поява Web OLAP-засобів стирає границі, що відокремлюють OLAP-ринок від суміжних категорій програмного забезпечення. Web-платформи інтерактивної звітності за своєю функціональністю усе більше схожі на стандартні Web OLAP продукти.

Internet/Intranet OLAP в електронному бізнесі

Сьогодні багато компаній застосовують intranet-технології для розгортання сховищ даних і надання OLAP-функцій у масштабі всього підприємства, а потім переходять і до технологій

extranet, щоб забезпечити своїх партнерів (постачальників, оптових і роздрібних продавців, представників філій) захищеним доступом до сховища і OLAP-засобів. Такий спосіб розширеної підтримки прийняття рішень дає змогу організаціям, що співпрацюють, спільно використовувати інформаційні ресурси і знаходити нові можливості розвитку бізнесу.

Однак спільний доступ вимагає визначеного рівня конфіденційності і захищеності інформації. Тому, перш ніж упровадити систему підтримки прийняття рішень у extranet, необхідно чітко визначити інформаційні й аналітичні потреби конкретних користувачів. Бізнес-партнерам, що потребують підтримки прийняття тактичних рішень, не потрібні засоби стратегічного аналізу і відповідна інформація. Так, для визначення обсягу постачань на наступний місяць постачальникам необхідні тільки дані про продажі за останній період, а зведення про довгострокові продажі або про плановані доходи їм зовсім не потрібні. Зовнішнім бізнес-партнерам (наприклад, менеджерам-координаторам) не потрібні могутні інструменти нерегламентованого аналізу, необхідні для стратегічного прийняття рішень.

Отже, Web – це дуже продуктивний спосіб передавання інформації всім учасникам процесу прийняття рішень. Застосовуючи технологію Web-OLAP разом з агентами оперативної доставки даних на сервер (push-technology), можна надати менеджерам пріоритетну інформацію, пов'язану з визначеними бізнес-правилами.

Web допомагає вирішувати складні задачі, які виникають під час поширення програмного забезпечення в нецентралізованих середовищах, що зустрічаються, як правило, у більшості компаній. За допомогою Web-браузерів здійснюється не залежна від платформи підтримка інтерактивної OLAP-звітності в міжнародних організаціях, що мають постачальників по всьому світі. Навчання співробітників OLAP-аналізові зводиться до мінімуму за рахунок використання добре знайомих Internet-функцій і методів навігації (наприклад, переходів за гіперпосиланнями для поглиблення в дані).

Архітектура Web-OLAP електронного бізнесу

Для архітектури, що реалізує ідею Web-OLAP, недостатньо простої реалізації клієнт/серверної моделі. З'являються нові технології обробки. Оцінка пропонованих продуктів залежить не тільки від аналізованого обсягу даних, але і від можливості масштабування за кількістю користувачів.

Більшість Web-OLAP додатків використовують загальну архітектуру, у якій клієнтський браузер взаємодіє з HTTP-сервером, що пересилає HTML-сторінки. Але, крім цього, надається ще і проміжне (сполучне) ПЗ, що зберігається на сервері. Такий компонент може прямо зв'язуватися з Web-браузером або взаємодіяти з HTTP-сервером, який потім повертає браузерові HTML-сторінки з додатковими даними.

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

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

Останнім компонентом цієї архітектури є база даних. Web-браузер не з'єднується з нею прямо: за це відповідає сполучна ланка. Отже, обмежені можливості HTML поряд з лімітованим доступом клієнтської машини на сервер забезпечують достатню безпеку під час видобування даних з реляційної бази.

У більшості випадків (хоча це і не обов'язково) сполучне ПЗ зберігається на тому ж комп'ютері, що і HTTP-сервер та має взаємодіяти з базою, де знаходиться сховище. При цьому необхідно забезпечити права користувачів і мережний доступ так, щоб проміжний компонент міг витягати дані, необхідні для OLAP-додатка. Сполучне ПЗ – потенційно критичний елемент у Web-OLAP архітектурі, який повинен бути спроектований оптимально, щоб не знижувати загальну продуктивність системи. За неправильного конфірування складна програма проміжного рівня може дуже повільно виконувати запити користувачів.

Розміщення проміжного ПЗ між клієнтською частиною і базою даних знижує витрати, зв'язані з переміщенням даних і потенційно може підвищити продуктивність системи.

Такий підхід запобігає розростанню обсягу СД. Централізоване збереження інформації, доступної для всіх користувачів за допомогою OLAP-інструментів, усуває ризик роботи із застарілими даними, тому що всі користувачі будуть звертатися тільки до тих елементів, що знаходяться в центральному сховищі.

Підходи до реалізації Web-OLAP електронного бізнесу

Сьогодні реалізовано безліч різних Web-OLAP рішень, зокрема на основі технологій HTML (DHTML), Java, Active, а також комбінації вищеназваних.

Перелічимо основні типи продуктів:

1. HTML(DHTML)-рішення.
2. HTML з розширеннями – CGI, що пов'язує ПЗ.
3. HTML з використанням Java-аплетів.
4. Java або ActiveX-компоненти.

HTML-рішення

Для реалізації OLAP-функціональності в Web-браузері використовують тільки HTML.

Простим прикладом такого рішення є OLAP-інструмент, що дозволяє користувачеві виконувати визначені OLAP-запити або звіти з браузера; іншої функціональності при цьому не передбачається.

У цьому випадку для одержання даних з автономних OLAP-машин використовуються планувальники, що формують статичні HTML-звіти за HTML-шаблонами. Шаблони створюють так, щоб усі звіти мали погоджений вигляд. Звіти обробляються і передаються в браузер за допомогою Web-сервера. Статичні звіти характеризуються доброю переносимістю, швидко доставляються в браузер, при цьому взаємодії користувача з браузером практично не відбувається. У деяких випадках імітується перехід за вимірами шляхом навігації по звіту. Планувальник може створити набір звітів, зв'язаних між собою гіперпосиланнями. Наприклад, користувач, клацнувши по посиланню за назвою "Третій квартал", перейде до іншого звіту, що містить дані за липень, серпень і вересень.

Існує й інший підхід, частіше використовуваний: OLAP-сервер наповнює HTML-шаблон даними в оперативному режимі, тобто в міру надходження запиту користувача через браузер. У цьому випадку на Web-сервері зберігаються тільки шаблони звітів і метадані. Ці метадані містять інформацію, необхідну Web-серверові для передавання тих або інших даних у HTML-файл перед тим, як відправити його в браузер.

Метадані можуть зберігатися на сервері в двох форматах: у вигляді звичайних HTML-тегів у шаблонах на сервері (у HTML-файлі) або в двійковому форматі, наприклад, як поле в базі даних.

HTML-шаблони також можна відобразити в одному з двох форматів. Якщо для створення шаблонів використовують готовий редактор, то вони зберігаються в звичайних гіпертекстових файлах на Web-сервері. Користувач може клацнути в браузері по посиланню з назвою звіту і HTML-шаблона. Деякі постачальники пропонують інструментальні Web-засоби для створення і збереження метаданих шаблонів разом з метаданими звітів у двійковому форматі. У такому випадку і звіт, і шаблон можуть бути створені в оперативному режимі за допомогою спеціальних процесів Web-сервера.

При будь-якому способі збереження метаданих шаблонів і звітів інформація збирається Web-сервером відповідно до коду звіту, що посилається з браузера. Програмне забезпечення Web-сервера використовує метадані звіту, щоби добути відповідні дані з бази. База може зберігатися як на тому самому комп'ютері, що і додаток Web-сервера, так і на іншому. Отримані з бази дані поєднуються на основі шаблона в звіт і передаються в браузер.

Як правило, звіт уже містить за замовчуванням визначену OLAP-функціональність. Під час взаємодії зі звітом користувацький код посилають разом з іншою інформацією на Web-сервер, що використовує цей код для відстеження інформації, яку користувач бачить у браузері. Код може зберігатися в HTML-файлі як частина HTML-команди, що надходить на сервер у той момент, коли користувач запитує нові дані. Як правило, такі рішення мають обмежені можливості. HTML не може передати точні X і Y координати крапки, де має розташовуватися той або інший елемент звіту, тому залежно від браузера або платформи звіт може виглядати по-різному. Повного набору OLAP-функцій (обертання, агрегування і поглиблення в дані) не передбачено.

HTML з розширеннями – CGI

Одне із найслабших місць HTML – неможливість збереження стану. Пропонований варіант вирішення проблеми – використання CGI (**C**ommon **G**ateway **I**nterface – загальний шлюзовий інтерфейс) або інших Web API (**A**pplication **P**rogramming **I**nterface – інтерфейс прикладного програмування) для реалізації сполучного ПЗ (middleware). За допомогою цього методу можна забезпечити збереження станів, а також буферизацію рядків, що повертаються до клієнта, і виконання деяких обчислень над переданими даними.

У випадку використання такої архітектури переносимість для клієнтських платформ забезпечується використанням HTML як інтерфейсу. Однак залежно від того, як реалізовано додаток Web-сервера, проект може також переноситися з однієї серверної платформи на іншу. Це, зокрема, стосується CGI-додатків, що мають високу переносимість. Про більшість інших Web API цього сказати не можна. Наприклад, серверний додаток, розроблений з використанням API Microsoft Internet Information Server (ISAPI), найімовірніше працюватиме тільки на Internet Information Server.

Однак і в цьому випадку використання тільки HTML накладає обмеження на інтерфейс. Правильне відображення графіків і звітів буде утруднено. Ця проблема вирішується за допомогою Java-аплетів, що мають великі можливості з керування виведенням інформації на монітор.

HTML з використанням Java-апплетів

Спільне використання HTML і Java-аплетів дає змогу створити інакше Web-OLAP рішення. У цьому випадку HTML застосовують для відображення меню і виконання простих інтерфейсних функцій, а за допомогою Java-аплетів забезпечуються складніші компоненти інтерфейсу програми, а також погоджене відображення діаграм, графіків і таблиць. У таких додатках найчастіше реалізовані функції агрегування і деталізації, а також обертання (pivoting) даних. За рахунок широких графічних можливостей Java порівняно з HTML можна подавати дані у вигляді діаграм і змінювати їх в інтерактивному режимі.

Java або Active X

За наступним підходом використовують Java- або ActiveX-компоненти для мінімізації взаємодії між браузером і Web-сервером, а також розширення можливостей користувача для роботи з даними за рахунок якіснішого інтерфейсу. Існує два способи застосування цих компонентів.

У першому випадку Web-сервер заповнює двійковий файл даними для звітів, і інтерфейсні компоненти посилаються в браузер разом з відповідним HTML-файлом. На клієнті одну з властивостей керівного елемента (Active або Java) задає для браузера назва відповідного файла даних. Браузер стягує цей файл, а компонент завантажує дані. Отже, компоненти забезпечують виконання таких OLAP-функцій, як агрегування, деталізація й обертання за допомогою зручного інтерфейсу, без звертання до сервера. Оскільки передавання даних між клієнтом і сервером здійснюється в двійковому форматі, такий підхід можна вважати безпечнішим.

У другому випадку інтерфейсний компонент прямо з'єднується із сервером, що в інтерактивному режимі передає дані клієнту у міру надходження запитів користувачів. У цьому випадку інтерфейсний компонент запитує дані з Web-сервера, відкриваючи HTTP-потік. Потім, одержавши результати із сервера, він аналізує отриману інформацію і віддає дані на об'єкт для відображення.

Крім того, компоненти дуже зручно використовувати разом зі сполучним ПО-додатком, що здійснює буферизацію даних і виконує обчислення й агрегування перед тим, як передавати дані на клієнт. Потенційно, за рахунок виконання цих дій не на клієнті, а на сервері загальна продуктивність Web-OLAP додатків може підвищитися. Саме таке рішення дає найширший набір OLAP-функцій.

Висновки з даного дослідження та перспективи подальших розвідок у цьому напрямку

Під час реалізації проектів з побудови сховищ даних систем електронної комерції виникають завдання, що не залежать від предметної галузі оброблюваної інформації, до яких належать:

- Проектування структури ієрархічних вимірів;
- Проектування структури повільно мінливих вимірів;
- Проектування й актуалізація агрегатних значень.

Описано застосування сучасних інформаційних технологій і інструментальних засобів під час створення системи підтримки прийняття рішень для керування банком. Наведено реалізацію сховища даних разом з аналітичними додатками для одержання фінансових звітів і проведення аналізу даних. Особливу увагу приділено переміщенню інформації з OLTP-систем обробки даних до інформаційного сховища, зокрема проблемам планування роботи для його початкового завантаження й поповнення.

Розглянуто можливі рішення цих завдань, однак, у всіх розглянутих завданнях існують не вирішені питання, що потребують подальших досліджень. Зокрема, складним для реалізації є випадок ієрархічних вимірів з необхідністю підтримки історії змін, пов'язаних з якими-небудь іншими довідниками. Не розглядалися питання щодо методів відбору даних і алгоритмів завантаження даних до сховища. Ці теми потребують окремого розгляду.

Було виділено три основні проблеми, яким приділяють недостатню увагу під час створення сховищ даних: якість даних, оптимальний вибір джерел даних, їх продуктивність і масштабованість. Сьогодні в сховищах даних утримується безліч даних, які ніколи не використовуються додатками, виконуваними над цими сховищами даних, і ці непотрібні дані є однією із причин зниження ефективності виконання запитів. Потрібно забезпечити можливість реєстрації повного набору запитів, генерованих всіма додатками, і використання таблиць і полів, що фігурують у запитах, для тонкого налагодження вмісту сховищ даних. У сьогоднішніх сховищах даних для зберігання даних і керування ними значною мірою використовують OLAP-системи . Однак можливості сьогоднішніх OLAP-систем не достатні для обробки запитів, орієнтованих на сканування – таких, як групування записів, обчислення агрегатів і операції переміщення файлів, які переважають на етапах перетворення даних сховищ даних та підготовки даних під час видобування даних.

Вимоги законодавства, тиск витрат, прагнення оптимізувати бізнес-процеси й модернізувати корпорації роблять сьогодні технології BI і сховищ даних як ніколи затребуваними. Той факт, що Business Intelligence і сховища даних – це сформована галузь діяльності, а також те, що ринок цих технологій уже досить добре розвинений, переконують корпорації вкладати гроші як для того, щоби впоратися зі старими проблемами, так і для того, щоб підняти рівень своєї організації на вищий рівень розвитку цих передових технологій.

Оперативна аналітична обробка багато років використовувалася переважно бізнес-аналітиками й іншими експертами у галузі обробки даних. Але з недавньою появою Web-OLAP-систем, які спрощують упровадження програмного забезпечення і пропонують користувачеві знайомий інтерфейс браузера, компанії стали впроваджувати складні аналітичні можливості у масштабах усього підприємства.

1. Спирли, Э. Корпоративные хранилища данных. Планирование, розработка, реализация.– Т. 1: Пер. с англ. – М.: "Вильямс", 2001; 2. Дюк В, Самійленко А. Data mining: навчальний курс. – СПб: Питер, 2001; 3. Nigel Pendse, OLAP Architectures: The OLAP Report; 4. http://www.olapreport.com; 5. http://www.bipartner.ru; 6. http://kis.pcweek.ru; 7. http://olap.ru; 8. http://www.platsoft.ru; 9. http://citforum.ru; 10. http://www.elar.ru; 11. http://www.sybase.ru