УДК 004.658: 652.3

ДВОРЕЦЬКИЙ М.Л.

ПРОЕКТУВАННЯ ТА ОЦІНКА ОПТИМАЛЬНОСТІ СТРУКТУРИ СХОВИЩА ДАНИХ ТА БАГАТОВИМІРНОЇ БД

У статті розглянуто проектування структури сховища даних на базі аналізу щодо вимог кінцевих замовників інформації та запропоновано алгоритм визначення оптимальності структури СД та багатовимірної БД, що базується на кластерному аналізі.

In article is discussed planning of data warehouse structure based on analysis of the requirements of eventual customer of information. Also is offered the algorithm, witch determinate an optimum structure of data warehouse and multidimensional DB, based on a cluster analysis.

Загальна постановка проблеми та її зв'язок із науково-практичними задачами. Прагнення об'єднати в одній архітектурі СППР можливості OLTP-систем і систем аналізу, вимоги до яких багато в чому суперечливі, привело до появи концепції *сховищ даних* (СД).

Концепція СД так чи інакше обговорювалася фахівцями в області інформаційних систем досить давно. Перші статті, присвячені саме СД, з'явилися в 1988 р., їхніми авторами були Девлін і Мерфі. У 1992 р. Уільман Г. Інмон докладно описав дану концепцію у своїй монографії "Побудова сховищ даних" [1].

Сховище даних – предметно-орієнтований, інтегрований, немінливий, підтримуючий хронологію набір даних, організований для цілей підтримки прийняття рішень.

Зупинимося на основних проблемах створення СД:

- необхідність інтеграції даних з неоднорідних джерел у розподіленому середовищі;
- потреба в ефективному збереженні й обробці дуже великих обсягів інформації;
- необхідність наявності багаторівневих довідників метаданих;
- підвищені вимоги до безпеки даних.

Зниження витрат на створення СД можна домогтися, створюючи його спрощений варіант *вітрину даних* (Data Mart).

Вітрина даних (ВД) це спрощений варіант СД, що містить тільки тематично об'єднані дані [2].

ВД максимально наближена до кінцевого користувача і містить дані, тематично орієнтовані на нього (наприклад, ВД для працівників відділу маркетингу може містити дані, необхідні для маркетингового аналізу). ВД істотно менше по об'єму, ніж СД, і для її реалізації не потрібно великих витрат. Вони можуть бути реалізовані як самостійно, так і разом із СД.

Самостійні ВД часто з'являються у великих організаціях з великою кількістю незалежних підрозділів, що вирішують власні аналітичні задачі.

Останнім часом усе більш популярною стає ідея сполучити СД і ВД в одній системі. У цьому випадку СД використовується як єдине джерело інтегрованих даних для усіх ВД. СД являє собою єдине централізоване джерело інформації для всієї предметної області, а ВД є підмножинами даних зі сховища, організованими для представлення інформації тематичних розділів даної області [3]. Кінцеві користувачі мають можливість доступу до детальних даних сховища, якщо даних у вітрині недостатньо, а також для одержання більш повної інформаційної картини. Перевагами такого підходу є:

- простота створення і наповнення ВД, оскільки наповнення походить з єдиного стандартизованого надійного джерела очищених даних з СД;
- простота розширення СППР за рахунок додавання нових ВД;
- зниження навантаження на основне СД.

Основними засобами названого вище аналізу в останні роки стали засоби оперативної аналітичної обробки (on-line Analytical Processing – OLAP).

Кінцевою метою використання OLAP-технологій є аналіз даних і представлення результатів цього аналізу у вигляді, зручному для сприйняття інформації та прийняття рішень. Основна ідея OLAP полягає у побудові багатовимірних кубів, які будуть доступні для аналітичних запитів, що будуть корисними для прийняття рішень [4].

Основними складовими сховища даних є таблиця фактів (fact table) та таблиці вимірів (dimension tables). Таблиця фактів, як правило, містить відомості про об'єкти та події, сукупність яких буде у подальшому аналізуватися. Вона, як правило, містить унікальний складений ключ, що поєднує первинні ключі таблиць вимірів. Найчастіше це цілочисельні значення або значення типу дата/час – адже таблиця фактів може містити мільони записів і зберігати у ній текстові поля, як правило, не вигідно – краще помістити їх у менші за об'ємом таблиці вимірів. При цьому як ключові, так і деякі не ключові поля мають відповідати майбутнім вимірам OLAP-кубу. Окрім цього, таблиця фактів має одне або декілька числових полів, на основі яких у подальшому будуть отримані агрегатні дані [5].

Швидкість збільшення об'єму таблиць вимірів має бути незначною по відношенню до швидкості збільшення об'єму таблиці фактів, наприклад, додавання нового запису у таблицю вимірів, яка характеризує товари, відбувається лише при появі нового товару, що не продавався раніше.

При висвітленні питань розробки структури СД [6-8] та засобів синхронізації із оперативними джерелами даних [1-2,6,8-9] приділяється увага таким питанням, як завантаження, вилучення, перетворення даних, що отримуються із оперативних джерел. Тому актуальним залишається питання розробки структури СД виходячи із вимог кінцевого замовника інформації до вітрин даних.

Огляд публікацій та аналіз невирішених питань. Існує цілий ряд публікацій [1-2, 4, 6-8], які висвітлюють питання розробки структури СД та багатовимірних БД виходячи із аспектів синхронізації із оперативними джерелами даних.

Крім того, приводяться загальні підходи при проектуванні СД у вигляді однієї таблиці фактів та декількох таблиць вимірів [1, 4-6] із подальшим створенням на його базі OLAP-кубів.

Але при цьому не достатньо уваги приділяється питанням структури СД в аспекті вимог до звітів кінцевих замовників інформації. Крім того, не розглядається питання оптимізації структури СД.

Метою досліджень є розробка структури СД та багатовимірної БД на базі аналізу щодо вимог кінцевих замовників інформації та оптимізація структури СД виходячи із

показників швидкодії оновлення даних та формування звітів та об'єму інформаційної бази.

Результати досліджень. На першому кроці проектування структури СД визначаємося із вимогами до звітної інформації, та формалізуємо їх у вигляді наступних множин. По перше, кожен користувацький звіт представляємо в наступному вигляді:

$$R_i = D_i \text{ x } M_i \quad (1)$$

де D_i – множина вимірів і-го звіту, M_i – множина показників і-го звіту

Загальну множину користувацьких звітів визначимо наступним чином:

$$R = \{R_1, R_2, R_3, \dots, R_i, \dots, R_n\} \quad (2)$$

де n – загальна кількість користувацьких звітів

Далі визначаємо загальні множини вимірів та показників, які є об'єднанням множин вимірів і показники відповідно для множини всіх звітів:

$$D = \bigcup_{i=1}^{n} D_i \quad (3)$$

$$M = \bigcup_{i=1}^{n} M_i \quad (4)$$

Наступним кроком є виключення із множини R таких звітів R_j , які є підмножиною деякого звіту R_i. Тобто звіт R_j має бути виключеним, якщо істинне наступне твердження:

$$R_j \subset R_i \quad (5)$$

де $R_j \subset R$ та $R_i \subset R$

Будемо вважати, що множина j-го звіту R_j є підмножиною і-го звіту R_i, якщо множина вимірів j-го звіту D_j є підмножиною множини вимірів D_i і-го звіту і множина показників j-го звіту M_j є підмножиною множини показників M_i і-го звіту, тобто якщо істинні наступні твердження:

$$D_j \subset D_i \quad (6)$$

де $D_j \subset D$ та $D_i \subset D$

$$M_j \subset M_i \quad (7)$$

де $M_j \subset M$ та $M_i \subset M$

Надалі при розробці структури СД, необхідно визначитись із основними його складовими – таблицями фактів та таблицями вимірів.

Із переліком таблиць вимірів визначитись досить нескладно. Цілком імовірно, що в різних звітах будуть зустрічатися однакові виміри, тобто множини D_1, D_2, …, D_i, … D_n можуть між собою перетинатись. Тому кількість тіблиць вимірів буде відповідати кількості елементів множини D, що є об'єднанням множин вимірів всіх звітів.

Однак технологія OLAP використовує ще поняття загального виміру – виміру, що використовується більш, ніж однією таблицею фактів. Цю множину визначемо, як об'єднання попарних перерізів множин вимірів всіх користувацьких звітів, тобто:

$$D_{shared} = \bigcup_{i=1}^{n} (D_i \bigcap D_{i+1}) \quad (8)$$

де n – загальна кількість звітів.

Звідси множину вимірів, що не є загальними, визначаємо як різницю множин D та D_{shared}.

$$D_{simple} = D \setminus D_{shared} \quad (9)$$

Отже, таблиці вимірів визначаємо на базі елементів множини D_{simple}, таблиці занальних вимірів – на базі елементів множини D_{shared}.

Далі переходимо до визначення таблиць файтів. Визначення їхнього переліку та складу є дещо складнішим. Кількість таблиць фактів може змінюватить у межах від однієї до кількості елементів множини показників. Тобто сховище даних може бути спроектоване таким чином, що буде складатися із однієї таблиці фактів, або буде включати окрему таблицю фактів для кожного показників, що аналізується. Зрозуміло, що оптимальним найчастіше виявляється розбиття множини показників на декілька групп (кластерів), та представлення кожної группи показників у вигляді окремої таблиці фактів.

Крім того, кожна з таблиць фактів містить зовнішні ключі, що є посиланнями на таблиці вимірів. У найпростішому випадку можливо додати посилання на всі елементи множини вимірів D до кожної таблиці фактів. Але кожен наступний вимір таблиці фактів збільшує об'єм базатовимірної БД на порядок, що не може негативно не вплинути на швидкодію синхронізації даних із оперативними джерелами даних та на формування користувацьких звітів. Тому перелік вимірів кожної таблиці фактів має бути чітко визначеним і в жодному разі не надлишковим.

Тому при розбитті множини показників між таблицями фактів пропонується застосувати елементи кластерного аналізу із побудовою оцінки кожного з варіантів розбиття, виходячи із запобігання надлишкового об'єму багатовимірних БД. Далі коротко наведемо описання агломеративного алгоритму кластеризації у контексті задачі, що розглядається.

На першому кроці вся множина I представляється як множина кластерів:

$$c_1=\{i_1\},\ c_2=\{i_2\},\ \ldots,\ c_m=\{i_m\} \tag{10}$$

На наступному кроці вибираються два найбільш близькі одна одній (наприклад, c_p та c_q*)* і поєднуються в один загальний кластер. Нова множина, що складається вже з *m-1* кластерів, буде:

$$c_1=\{i_1\},\ c_2=\{i_2\},\ \ldots,\ c_p=\{i_p, i_q\},\ldots,c_m=\{i_m\} \tag{11}$$

Повторюючи процес, одержуємо послідовні множини кластерів, що складаються з *(m-2), (m-3), (m-4)* і т.д.

Наприкінці процедури вийде кластер, що складається з *m*-об'єктів і співпадає з первісною множиною.

Для визначення відстані між кластерами можна вибрати різні способи. У залежності від цього одержують алгоритми з різними властивостями.

Існує кілька методів перерахування відстаней з використанням старих значень відстаней для поєднуваних кластерів, що відрізняються коефіцієнтами у формулі:

$$d_{rs}=\alpha_p d_{ps}+ \alpha_q d_{ps} +\beta d_{pq}+\gamma \left| d_{ps}-d_{qs} \right| \tag{12}$$

Якщо кластери *p* та *q* поєднуються в кластер r і потрібно розрахувати відстань від нового кластера до кластера *s,* застосування того чи іншого методу залежить від способу визначення відстані між кластерами, ці методи розрізняються значеннями коефіцієнтів α_p, α_q, β і γ.

Елементами множини I у нашому випадку є користувацькі звіти, тобто

$$I \equiv R \tag{13}$$

При визначенні звітів, які будуть об'єднані в один кластер, мірою близькості виступає кількість спільних вимірів із ваговими коефіцієнтами кожного із них. Тобто міра близькості двох звітів може бути розрахована наступним чином:

$$C_{ij} = \sum_{k=1}^{m} koef_d * D_k^{ij} \tag{14}$$

де

$$B^{\text{що}} = (B_{\text{ш}} \setminus B_{\text{о}}) \bigcap (B_{\text{о}} \setminus B_{\text{ш}}) \quad (15)$$

а $koef_d$ визначається кількістю елементів кожного із вимірів.

В результаті отримуємо n-варіантів розбиття користувацьких звітів на кластери, де n-кількість елементів множини R. Для визначення найкращого з варіантів будуємо оціночну функцію, та знаходимо її мінімальне значення:

$$F = \sum_{i=1}^{n} \sum_{j=1}^{m} Koef_j * D_j * count(M_i) \quad (16)$$

де n – кількість кластерів, $count(M_i)$ – кількість показників і-го кластера, m – кількість вимірів і-го кластера.

Викладений підхід щодо оптимізації структури СД дає схожі результати при застосуванні дивізимних алгоритмів кластеризації. На першому кроці всі елементи входять до одного кластеру C_1=I. Потім вибирається елемент, у якого середнє значення відстані від інших елементів у цьому кластері найбільше. Середнє значення може бути обчислено, наприклад, за допомогою формули

$$B_{CI}=1.\text{Тс}_1 * \sum\sum\text{вХш}_\text{р}\text{б ш}_\text{й}) \text{ ш}_\text{з}\text{б ш}_\text{й} \in C \quad (17)$$

Обраний елемент віддаляється з кластера C_1 і формує перший член другого кластера C_2.

На кожному наступному кроці елемент у кластері C_1, для якого різниця між середньою відстанню до елементів, що знаходяться в C_2, і середньою відстанню до елементів, що залишаються в C_1 найбільша, переноситься в C_2. Переноси елементів із C_1 у C_2 продовжуються доти, поки відповідні різниці середніх не стануть негативними, тобто поки існують елементи, розташовані до елементів кластера C_2 ближче чим до елементів кластера C_1.

У результаті один кластер поділяється на два дочірніх, один із яких розщеплюється на наступному рівні ієрархії. Кожен наступний рівень застосовує процедуру поділу по одному з кластерів, отриманих на попередньому рівні.

Висновки та перспективи подальших досліджень. Розробка структури СД та багатовимірної БД на базі аналізу звітності для кінцевих замовників інформації дозволяє за допомогою теорії множин формалізувати вимоги що інформаційної моделі.

Подальше застосувння операцій реляційної алгебри дозволяє сформулювати основні вимоги до таблиць фактів та вимірів шляхом виключення надлишкових звітів та визначення множин вимірів та показників.

Здійснивши кластеризацію звітів із використанням міри близькості користувацьких звітів, маємо можливість побудувати оціночну функцію для кожного з варіантів та визначити оптимальну структуру СД. Викладений підхід дозволяє спростити процес проектування структури СД та багатовимірної БД та запобігти виникненню надлишкових вимірів у таблицях фактів.

Слід зауважити, що при проведенні кластеризації розглядалась лише група ієрархічних алгоритмів, але в перспективі передбачається проведення кластеризації із використанням алгоритму k-means та порівняння результатів із отриманими вище.

ЛІТЕРАТУРА

1. Барсегян А.А., Куприянов М.С., Степаненко В.В., Холод И.И. Методы и модели анализа данных: OLAP и Data Mining. – СПб., 2004. – 336 с.
2. Архипенков С. Я., Голубев Д. В., Максименко О. Б. Хранилища данных. – Диалог-МИФИ, 2002 г. 528 с.
3. Гарсиа-Молина Г., Ульман Дж.Д., Уидом Дж. Системы баз данных. Полный курс. – М.: Изд. дом "Вільямс", 2003. – 1088 с.
4. Введение в OLAP: часть 2. Хранилища данных. – http://www.olap.ru/basic/olap_intro2.asp

5. Использование OLAP-технологии в CRM-программном комплексе. http://www.interface.ru/fset.asp?Url=/crystal/olapincrm.htm
6. Елманова Е., Федоров А. ВВЕДЕНИЕ В OLAP-технологии MICROSOFT. – Диалог-МИФИ, 2002. – 272 с.
7. Эрик Спирли. Корпоративные хранилища данных. Планирование, разработка и реализация. Т.1. Вильямс, 2001. – 400 с.
8. Аналитические системы и хранилища данных. – http://www.interface.ru/fset.asp? Url=/oracle/acdc
9. Морозов А.А. Системы принятия решений: проблемы и перспективы. Управляющие системы и машины. – 1995. – № 1. – С. 13-21 с.