УДК 004.652
В.М. Барбарук, Л.В.Барбарук
Технологічний інститут Східноукраїнського національного університету ім'я Володимира Даля (м.Сєвєродонецьк)

ЗАСОБИ ПРЕДСТАВЛЕННЯ БАГАТОМІРНИХ ДАНИХ В ІНФОРМАЦІЙНО-АНАЛІТИЧНИХ СИСТЕМАХ

Розглянута задача застосування сховищ даних в інформаційно-аналітичних системах. Показано переваги і недоліки використання реляційних СУБД і систем оперативної аналітичної обробки даних. Запропоновано представляти багатомірну інформацію за допомогою зіркоподібних реляційних моделей OLAP, що дозволяє позбутися від проблеми оптимізації зберігання розріджених матриць, яка гостро стоїть перед багатомірними СУБД. Наведено приклад побудови зіркоподібних схем даних для організації системи оперативної аналітичної обробки даних медичної інформаційної системи.

БАЗА ДАНИХ, ТАБЛИЦЯ, КУБ, ВИМІР, ФАКТ

Реляційна модель даних, яка була запропонована Коддом в 1970 році, і за яку десятиліття через він отримав премію Тьюрінга, є основою сучасної багатомільярдної галузі баз даних. За останні десять років склалася багатомірна модель даних, яка використовується, коли метою є саме аналіз даних, а не виконання транзакцій. Технологія багатомірних баз даних — ключовий фактор інтерактивного аналізу більших масивів даних з метою підтримки ухвалення рішення.

По Кодду, багатомірне концептуальне представлення (multi-dimensional conceptual view) являє собою множинну перспективу, що полягає з декількох незалежних вимірів, уздовж яких можуть бути проаналізовані певні сукупності даних. Одночасний аналіз по декільком вимірам визначається як багатомірний аналіз. Кожний вимір включає направлення консолідації даних, що полягають із серії послідовних рівнів узагальнення, де кожний вищий рівень відповідає більшому ступеню агрегації даних по відповідному до виміру. Так, вимір "Лікар" може визначатися напрямком консолідації, що полягають із рівнів узагальнення " управління охорони здоров'я - поліклініка - відділення – медичний працівник". Вимір "Час" може навіть включати два напрямки консолідації - "рік - квартал - місяць - день" і "тиждень - день", оскільки рахунок часу по місяцях і по тижнях несумісний. У цьому випадку стає можливим довільний вибір бажаного рівня деталізації інформації з кожного з вимірів. Операція спуска (drilling down) відповідає руху від вищих щаблів консолідації до нижчих; напроти, операція підйому (rolling up) означає рух від нижчих рівнів до вищих (рис. 1).

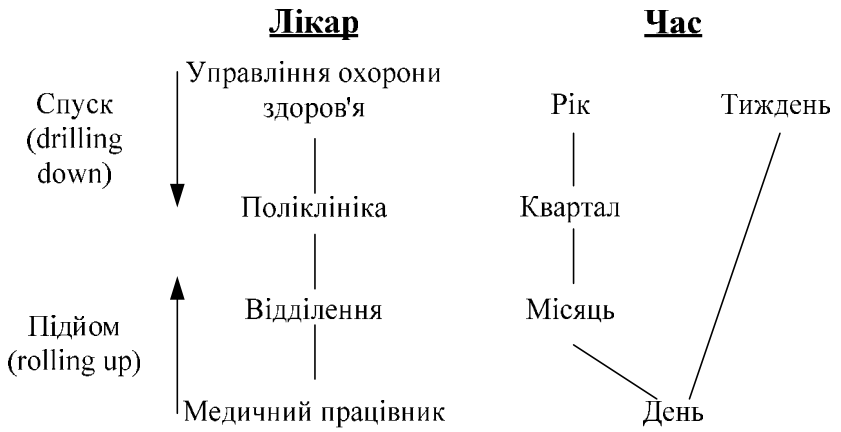

Рис.1. Виміри і напрямки консолідації даних

Багатомірні моделі розглядають дані або як факти з відповідними чисельними параметрами, або як текстові виміри, які характеризують ці факти. У медичній статистиці, приміром, звернення пацієнта — це факт, симптоми і скарги — параметри, а поставлений діагноз, дата і місце обстеження — вимірювання. Запити агрегують значення параметрів по всьому діапазону вимірювання, і в підсумку одержують такі величини, як загальне місячне число звернень пацієнтів

з даними симптомами. Багатомірні моделі даних мають три важливі області застосування, пов'язаних із проблематикою аналізу даних:

– сховища даних інтегрують для аналізу інформації з декількох джерел організації / підприємства;

– системи оперативної аналітичної обробки (online analytical processing — OLAP) дозволяють оперативно одержати відповіді на запити, що охоплюють більші обсяги даних у пошуках загальних тенденцій;

– додатка видобутку даних служать для виявлення знань за рахунок напівавтоматичного пошуку раніше невідомих шаблонів і зв'язків у базах даних.

У спеціалізованих СУБД, заснованих на багатомірному представленню даних, дані організовані не у формі реляційних таблиць, а у вигляді впорядкованих багатомірних масивів:

1) гіперкубів (усі збережені в БД комірки повинні мати однакову мірність, тобто перебувати в максимально повному базисі вимірів) або

2) полікубів (кожна змінна зберігається із власним набором вимірювань, і всі пов'язані із цим складності обробки перекладаються на внутрішні механізми системи).

Використання багатомірних БД у системах оперативної аналітичної обробки має наступні переваги.

1) У випадку використання багатомірних СУБД пошук і вибірка даних здійснюється значно швидше, чим при багатомірному концептуальному погляді на реляційну базу даних, тому що багатомірна база даних денормалізована, містить заздалегідь агреговані показники та забезпечує оптимізований доступ до запитуваних гнізд.

2) Багатомірні СУБД легко справляються із завданнями включення в інформаційну модель різноманітних вбудованих функцій обмеження, тоді як об'єктивно існуючі обмеження мови SQL роблять виконання цих завдань на основі реляційних СУБД досить складним, а іноді й неможливим.

З іншого боку, є істотні обмеження.

1) Багатомірні СУБД не дозволяють працювати з більшими базами даних. До того ж за рахунок денормалізації та попередньо виконані агрегації обсяг даних у багатомірній базі, як правило, відповідає (по оцінці Кодда [1]) в 2.5-100 раз меншому обсягу вихідних деталізованих даних.

2) Багатомірні СУБД у порівнянні з реляційними дуже неефективно використовують зовнішню пам'ять. У переважній більшості випадків інформаційний гіперкуб є сильно розрідженим, а оскільки дані зберігаються в упорядкованому вигляді, невизначені значення вдається вилучити тільки за рахунок вибору оптимального порядку сортування, що дозволяє організувати дані в максимально більші безперервні групи. Але навіть у цьому випадку проблема вирішується тільки частково. Крім того, оптимальний з погляду зберігання розріджених даних порядок сортування швидше за все не буде збігатися з порядком, який найчастіше використовується в запитах. Тому в реальних системах доводиться шукати компроміс між швидкодією і надмірністю дискового простору, зайнятого базою даних.

Отже, використання багатомірних СУБД виправдане тільки при наступних умовах.

1) Обсяг вихідних даних для аналізу не занадто великий (не більш декількох гігабайт), тобто рівень агрегації даних досить високий.

2) Набір інформаційних вимірювань стабільний (оскільки будь-яка зміна в їхній структурі майже завжди вимагає повної перебудови гіперкуба).

3) Час відповіді системи на нерегламентовані запити є найбільш критичним параметром.

4) Потрібне широке використання складних вбудованих функцій для виконання кроссмерних обчислень над комірками гіперкуба, у тому числі можливість написання користувацьких функцій.

У якості мір у тривимірному кубі, зображеному на рис. 2, використане число звернень пацієнтів, а в якості вимірювань - час, діагнози і регіон. Виміри представлені на певних рівнях угруповання: діагнози групуються по категоріях, медичні заклади - по містах, а дані про час виконання операцій - по місяцях.

Березень
Лютий
Січень

	Сєвєродонецьк	Лисичанськ	Рубіжне
A00-T98	12000	11000	12500
A00-B99	3000	3500	2900
G00-G99	500	400	520
K00-K99	1300	1200	1300

Рис. 2. Приклад кубу

Навіть тривимірний куб складно відобразити на екрані комп'ютера так, щоб були видні значення мір, що цікавлять. Для візуалізації даних, що зберігаються в кубі, застосовуються, як правило, звичні двовимірні, тобто табличні, представлення, що мають складні ієрархічні заголовки рядків і стовпців.

Двовимірне подання куба можна одержати, "розрізавши" його поперек однієї або декількох осей (вимірів): ми фіксуємо значення всіх вимірів, крім двох, - і одержуємо звичайну двовимірну таблицю. У горизонтальній осі таблиці (заголовки стовпців) представлений один вимір, у вертикальній (заголовки рядків) - інший, а в комірках таблиці - значення мір. При цьому набір мір фактично розглядається як один з вимірів - ми або вибираємо для показу одну міру (і тоді можемо розмістити в заголовках рядків і стовпців два виміри), або показуємо кілька мір (і тоді одну з осей таблиці займуть назви мір, а іншу - значення єдиного "нерозрізаного" виміру).

На рис. 3 зображений двовимірний зріз куба для однієї міри - число звернень і двох "нерозрізаних" вимірів - медична заклад і час.

	Сєвєродонецьк	Лисичанськ	Рубіжне
Січень	16800	16100	17220
Лютий	15920	14690	18450
Березень	17120	16950	19370

Рис. 3. Двомірний зріз кубу для однієї міри

На рис. 4 представлено лише один "нерозрізаний" вимір - "Звернення", але зате тут відображаються значення декількох мір - "Число пацієнтів", "Число звернень" і "Середній вік".

	Сєвєродонецьк	Лисичанськ	Рубіжне
Кількість пацієнтів	12982	11245	15851
Число звернень	15920	14690	18450
Середній вік	45	45	44

Рис. 4. Двомірний зріз кубу для декількох мір

Двовимірне представлення кубу можливе і тоді, коли "нерозрізаними" залишаються і більше двох вимірів. При цьому на осях зрізу (рядках і стовпцях) будуть розміщені два або більше виміри "куба, що розріжеться" - див. рис. 5.

	Січень			Лютий		
	Сєвєродонецьк	Лисичанськ	Рубіжне	Сєвєродонецьк	Лисичанськ	Рубіжне
Кількість пацієнтів	705	653	920	687	720	879
Число звернень	860	947	1156	783	1089	1032
Середній вік	45	45	44	45	45	44

Рис. 5. Двомірний зріз кубу з декількома вимірами на одній осі

Безпосереднє використання реляційних БД у системах оперативної аналітичної обробки має наступні передаги.

1) У більшості випадків корпоративні сховища даних реалізуються засобами реляційних СУБД, і інструменти ROLAP дозволяють робити аналіз безпосередньо над ними. При цьому розмір сховища не є таким критичним параметром, як у випадку MOLAP.
2) У випадку змінної розмірності завдання, коли зміни в структуру вимірів доводиться вносити досить часто, ROLAP системи з динамічним поданням розмірності є оптимальним рішенням, тому що в них такі модифікації не вимагають фізичної реорганізації БД.
3) Реляційні СУБД забезпечують значно більш високий рівень захисту даних і гарні можливості розмежування прав доступу.

Головний недолік ROLAP у порівнянні з багатомірними СУБД - менша продуктивність. Для забезпечення продуктивності, порівнянної з MOLAP, реляційні системи вимагають ретельного проробляння схеми бази даних і настроювання індексів, тобто більших зусиль із боку адміністраторів БД. Тільки при використанні зіркоподібних схем продуктивність добре настроєних реляційних систем може бути наближена до продуктивності систем на основі багатомірних баз даних.

Опису схеми зірки (star schema) і рекомендаціям з її застосування повністю присвячені роботи [2, 3, 4]. Її ідея полягає в тому, що є таблиці для кожного виміру, а всі факти містяться в одну таблицю, що індексується множинним ключем, складеним із ключів окремих вимірів (рис. 6). Кожний промінь схеми зірки задає, у термінології Кодда, напрямок консолідації даних по відповідному вимірі.

Рис.6. Приклад схеми "зірка"

У складних завданнях з багаторівневими вимірами має сенс звернутися до розширень схеми зірки - схемі сузір'я (fact constellation schema) і схемі сніжинки (snowflake schema) [3]. У цих випадках окремі таблиці фактів створюються для можливих сполучень рівнів узагальнення різних вимірів (рис.7). Це дозволяє домогтися кращої продуктивності, але часто приводить до надмірності даних і до значних ускладнень у структурі бази даних, у якій виявляється величезна кількість таблиць фактів.

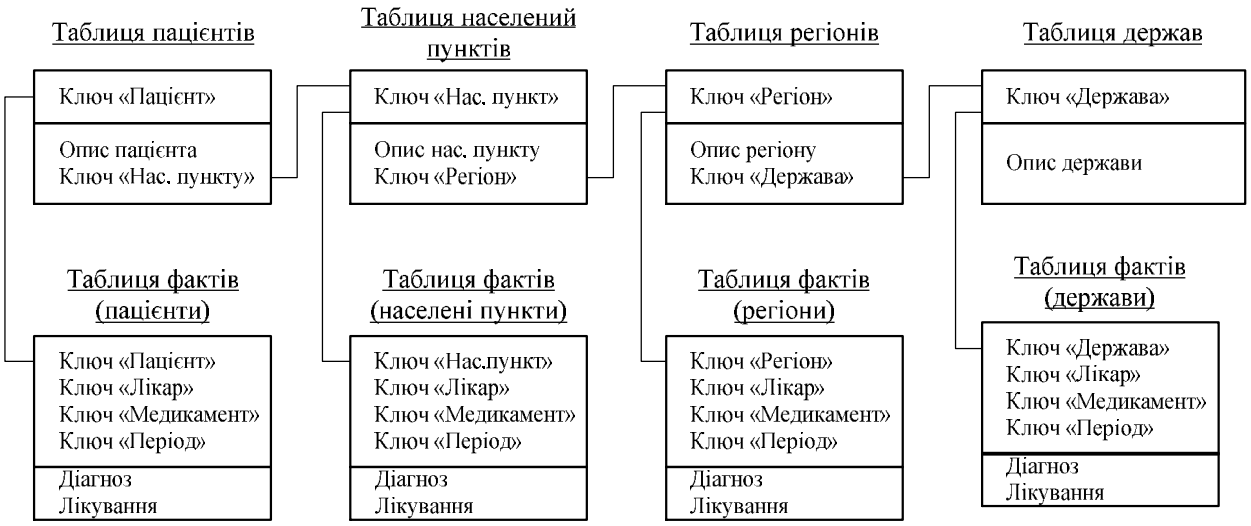

Рис.7. Приклад схеми "сніжинки" (фрагмент для одного виміру)

Збільшення числа таблиць фактів у базі даних може виникати не тільки із множинності рівнів різних вимірів, але і з тієї обставини, що в загальному випадку факти мають різні множини вимірів. При абстрагуванні від окремих вимірів користувач повинен одержувати проекцію максимально повного гіперкубу, причому далеко не завжди значення показників у ній повинні бути результатом елементарного підсумовування. Таким чином, при великій кількості незалежних вимірів необхідно підтримувати множину таблиць фактів, що відповідають кожному можливому сполученню обраних у запиті вимірів, що також приводить до неощадливого використання зовнішньої пам'яті, збільшенню часу завантаження даних у БД схеми зірки із зовнішніх джерел і складностям адміністрування. Частково вирішують цю проблему розширення мови SQL (оператори "GROUP BY CUBE", "GROUP BY ROLLUP" і "GROUP BY GROUPING SETS"); крім того, автори статей [*2,* 4] пропонують механізм пошуку компромісу між надмірністю і швидкодією, рекомендуючи створювати таблиці фактів не для всіх можливих сполучень вимірів, а тільки для тих, значення осередків яких не можуть бути отримані за допомогою наступної агрегації більше повних таблиць фактів (рис. 8).

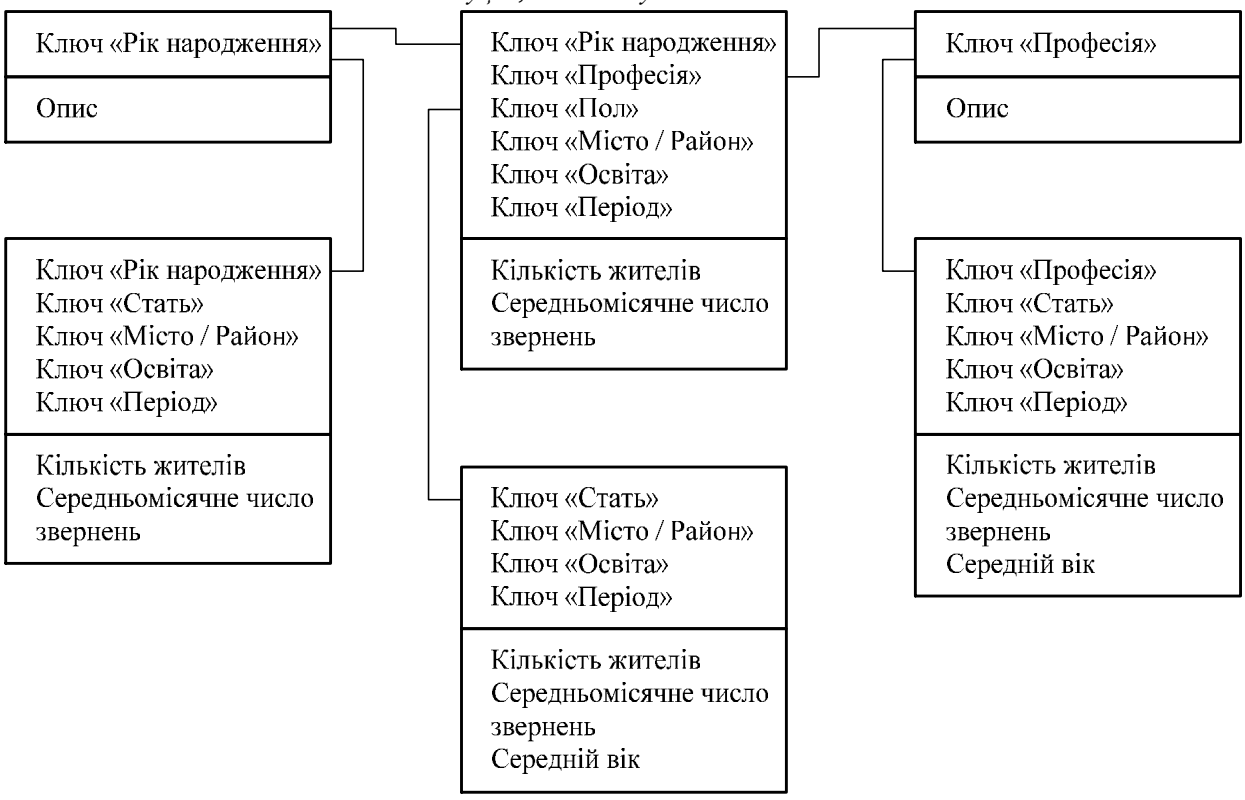

Рис.8. Таблиці фактів для різних поєднань вимірів у запиті

У кожному разі, якщо багатомірна модель реалізується у вигляді реляційної бази даних, варто створювати довгі і "вузькі" таблиці фактів і порівняно невеликі і "широкі" таблиці вимірів. Таблиці фактів містять чисельні значення комірок гіперкуба, а інші таблиці визначають утримуючий їхній багатомірний базис вимірів. Частина інформації можна одержувати за допомогою динамічної агрегації даних, розподілених по незіркоподібних нормалізованих структурах, хоча при цьому варто пам'ятати, що запити, що включають агрегацію, при високонормалізованої структурі БД можуть виконуватися досить повільно.

ВИСНОВКИ

Орієнтація на представлення багатомірної інформації за допомогою зіркоподібних реляційних моделей дозволяє позбутися від проблеми оптимізації зберігання розріджених матриць, яка гостро стоїть перед багатомірними СУБД (де проблема розрідженості вирішується спеціальним вибором схеми). Хоча для зберігання кожної комірки використовується цілий запис, що крім самих значень включає вторинні ключі - посилання на таблиці вимірів, неіснуючі значення просто не включаються в таблицю фактів.

1. Codd E. F., Codd S. B., Salley C. T. Providing OLAP (On-Line Analytical Processing) to User-Analysts: An IT Mandate. - E. F. Codd & Associates, 1993.
2. Harinarayan V., Rajaraman A., Ullman J. D. Implementing Data Cubes Efficiently // SIGMOD Conference. - Montreal, CA. -1996.
3. Raden N. Star Schema. - Santa Barbara, CA: Archer Decision Sciences, Inc., 1995-1996 (http://members.aol.com/nraden/str101.htm).
4. Mumick I. S., Quass D., Mumick B. S. Maintenance of Data Cubes and Summary Tables in a Warehouse. - Stanford University, Database Group, 1996 (http://www-db.stanford.edu/pub/papers/cube-maint.ps).