УДК 004.652.5

Фісун М. Т., Горбань Г. В.

СТВОРЕННЯ OLAP-КУБІВ В ОБ'ЄКТНО-ОРІЄНТОВАНИХ СКБД ЗА ДОПОМОГОЮ АЛГОРИТМУ DWARF

У статті розглянуто аспекти проектування OLAP-системи в об'єктно-орієнтованих базах даних: представлення фактів і вимірювань у них і можлива реалізація OLAP-куба у вигляді дерева за допомогою алгоритму DWARF. Дано опис класів, що використовуються в даному алгоритмі.

Ключові слова: *бази даних, SQL, NoSQL, реляційні СКБД, об'єктно-орієнтовані СКБД, вимір, ієрархія, клас, таблиця фактів, багатомірні бази даних, OLAP, B-дерево, DWARF.*

В статье рассмотрены аспекты проектирования OLAP-системы в объектно-ориентированных базах данных: представление фактов и измерений в них и возможная реализация OLAP-куба в виде древовидной структуры с помощью алгоритма DWARF. Данноописаниеклассов, используемыхвданномалгоритме.

Ключевые слова: *базы данных, SQL, NoSQL, реляционные СУБД, объектно-ориентированные СУБД, измерение, иерархия, класс, таблица фактов, многомерные базы данных, OLAP, B-дерево, DWARF.*

The paper deals with aspects of designing of OLAP-system in object-oriented databases: measurement and presentation of the facts in them and a possible implementation of OLAP-cube in a tree structure using an algorithm DWARF. A description of the classes used in this algorithm is present.

Key words: *database, SQL, NoSQL, relational DBMS, object-oriented DBMS, dimension, hierarchy, class, table of facts, multidimensional databases, OLAP, B-Tree, DWARF.*

Вступ. На даний час у системах підтримки прийняття рішень (СППР) важливе місце посідають технології оперативного (OLAP) та інтелектуального (Data Mining) аналізу даних [1]. Дані технології вже реалізовані в деяких комерційних системах управління базами даних (СКБД), таких як Microsoft SQL Server, Oracle і інших, які представляють собою реляційні СКБД.

Реляційні бази даних (БД) отримали великий успіх і на даний момент є найпоширенішими. Це обґрунтовано тим, що дані в них зберігаються в табличному форматі і доступні для кожного, хто знайомий зі структурованою мовою запитів SQL. Проте з часом інформаційні системи ставлять такі вимоги до обробки даних, з якими реляційні БД уже нездатні впоратися. Прикладом цього може служити зберігання зображень, звукових файлів і відеозаписів, слабко структурованих та розсіяних даних. Істотним недоліком реляційних БД є те, що зберігання реальних структур даних у вигляді стовпців і рядків таблиць є достатньо непростим завданням, оскільки при декомпозиції даних може створитись достатньо велика кількість таблиць, відношення між якими складно використовувати і запам'ятати. Таким чином, традиційні методи управління даними, засновані на попередньому визначенні схеми даних і посилань між даними, вже не відповідають сучасним вимогам обробки і аналізу великих різнотипних слабко структурованих даних. Ці види СКБД називаються NoSQL [2]. Вони не є повним запереченням мови SQL і реляційної моделі, підхід NoSQL виходить з того, що SQL – це важливий і дуже корисний інструмент, але при цьому він не може вважатися універсальним. Однією з проблем, на яку вказують для класичних реляційних БД, є проблема при роботі з даними дуже великого об'єму, зі слабко структурованими або неструктурованими даними, особливо у системах з високим навантаженням. Основна мета підходів NoSQL – розширити можливості БД там, де SQL недостатньо гнучкий, і не витісняти його там, де він справляється зі своїми завданнями дуже ефективно.

У літературі частіше за все виділяють такі види нереляційних (NoSQL) СКБД:

- стовпчико-орієнтовані СКБД;
- СКБД типу «ключ-значення»;
- бази даних на основі графів;
- документо-орієнтовані СКБД;
- багатовимірні бази даних;
- об'єктні СКБД.

У статті розглядаються останні два типи СКБД, взаємозв'язки між моделями даних цих СКБД та зв'язок із реляційними базами даних під кутом зору побудови OLAP-систем [3].

Деякі аспекти побудови системи OLAP в об'єктно-орієнтованих базах даних. Основою будь-якої OLAP-системи є куб, який являє собою впорядкований багатовимірний масив і споруджуваний на фактичних даних, які можуть перебувати в кількох таблицях фактів і зазвичай мають кілька вимірів [4; 5].Основними реалізаціями систем OLAP є MOLAP (багатовимірний OLAP), ROLAP (реляційний OLAP) і HOLAP (гібридний OLAP), що використовує як багатовимірні, так і реляційні БД. У MOLAP таблиця фактів містить усі комбінації значень усіх вимірювань і відповідні їм значення мір. На відміну від MOLAP, таблиця фактів у ROLAP не містить усі можливі варіанти комбінацій вимірювань; вона містить унікальний складовий ключ, який об'єднує первинні ключі таблиць вимірів.

Типовими схемами зв'язку таблиці фактів із таблицями вимірів є схеми «зірка» та «сніжинка». У схемі «зірка» таблиця фактів є ненормалізованою, оскільки дана схема не дозволяє врахувати можливі рівні ієрархії вимірів. Для цього застосовується схема «сніжинка», головною особливістю якої є зберігання інформації про один вимір в декількох зв'язаних таблицях. Однак головною вадою такої схеми є ускладнення структури баз даних, яка в цьому випадку буде містити достатньо багато таблиць фактів.

При порівнянні реляційної моделі даних з об'єктною моделлю між ними можна знайти достатньо багато аналогій. Так, реляційне поняття таблиці відповідає класу, рядок таблиці – екземпляру класу, стовпець – властивості класу, зовнішній ключ – посиланню на об'єкт і т. д. [6]. Таким чином, можна зробити висновок, що таблиця фактів в об'єктній моделі даних являє собою клас, властивостями якого є міри та посилання на таблиці вимірювань, які, у свою чергу, також є класами.

Структуру таблиці фактів і таблиць вимірів для об'єктної моделі даних можна показати на прикладі бази даних деякої торгівельної мережі, яка має магазини в різних регіонах країни, у які поставляються товари різних категорій. Факти про продаж певного товару в певному магазині за певний місяць фіксуються в таблиці фактів, що має назву «Продажі». Відповідно в даній системі буде 3 виміри: «Дата», «Магазини» і «Товари». Мірою в таблиці фактів є дохід за продаж.

При цьому кожний із вимірів має кілька рівнів ієрархії. Таким чином, нижнім рівнем ієрархії «Дата» є «Місяць», вище знаходяться рівні «Квартал» і «Рік». У свою чергу, для ієрархій «Магазини» і «Товари» нижніми рівнями є відповідно «Магазин» і «Товар», а верхніми – «Регіон» (кожен магазин знаходиться в певному регіоні) і «Категорія товару» (кожен товар можна віднести до певної категорії). На рисунку 1 показана схема вищевказаної бази даних.

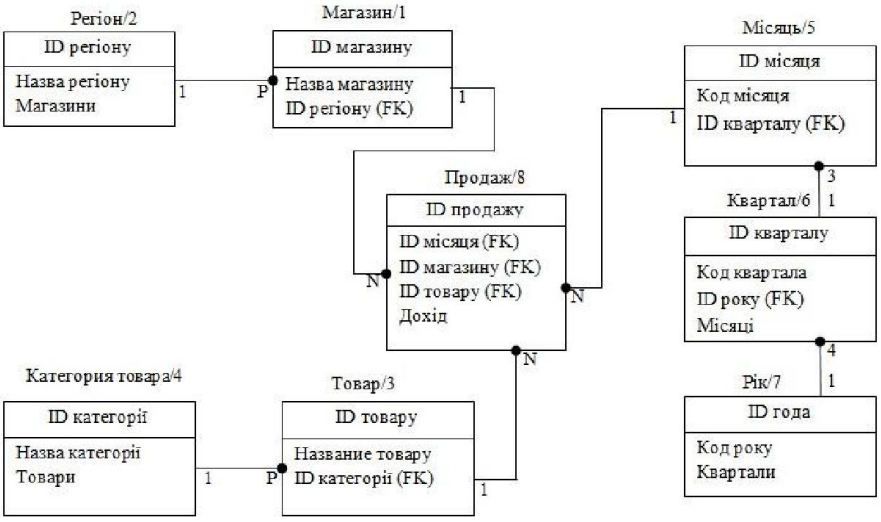

Рис. 1. Схема бази даних торгівельної мережі

У проекті стандарту об'єктних БД визначена мова ODL (ObjectDefinition Language – мова визначення об'єктів), синтаксис якого аналогічний синтаксису C++ і Java, проте точно з ними не збігається [7]. Опис класу «Продажі» мовою ODL буде виглядати так:

```
classFactSales
{
    attributeMonthfactmonth; //посилання на об'єкт класу «Місяць»
    attributeShopfactshop; // посилання на об'єкт класу «Магазин»
    attributeGoodsfactgoods; // посилання на об'єкт класу «Товар»
    attribute float profit; //дохід (міра таблиці фактів)
}
```

Таким чином, у таблиці фактів містяться посилання на екземпляри класів вимірювань нижнього рівня ієрархії. У свою чергу, класи вимірювань у своєму описі містять атрибути зв'язку з екземплярами класів більш високих рівнів ієрархії. Наприклад, для класу «Місяць»:

```
ClassMonth
{
    attributestringCodeMonth; //код місяця в базі даних (наприклад, «01-2013» – січень 2013 року)
    relationshipQuarterquarter
inverseQuarter::months; //опис зв'язку з класом «Квартал» (місяць завжди належить до певного кварталу, у свою чергу квартал має 3 місяці)
}
```

У класі «Квартал» також є вищевказаний зв'язок із класом «Місяць», а також зв'язок із класом «Рік».

```
ClassQuarter
{
attributestringCodeQuarter; // код місяця в базі даних (наприклад, «1 кв-2013» – 1 квартал 2013 року)
relationshipset<Month>months
inverseMonth::quarter; //опис зв'язку з класом «Місяць», у даному випадку атрибут представляє множину екземплярів класу «Місяць», оскільки в кварталі більш ніж 1 місяць
relationshipYearyear
inverseYear::quarters // опис зв'язку з класом «Рік», у якому також визначений відповідний зв'язок
}

Class Year
{
attribute integer CodeYear;
relationship set<Quarter> quarters
inverseQuarter::year; //відповідний зв'язок із класом «Квартал»
}
```

Відповідні зв'язки між собою мають класи інших вимірів. На відміну від таблиці фактів, яка містить посилання на екземпляри класів вимірювань, класи вимірювань містять атрибути зв'язку з класами вимірювань більш високого рівня для того, щоб не порушити цілісність бази даних.

При агрегуванні даних на більш високому рівні до вимірювань можна звертатися за допомогою точкового синтаксису, застосовуваного в об'єктній моделі даних, наприклад:

Month.Quarter.CodeQuarter – код кварталу, у який входить місяць;

Month.Quarter.Year.CodeYear – рік, у який входить місяць (доступ через об'єкт класу «Квартал»).

За допомогою точкового синтаксису можна також звернутися від значень більш високого рівня ієрархії до більш низького, наприклад:

Year.Quarter [n]. CodeQuarter – код n-го кварталу року;

Year.Quarter [n]. Month [m]. CodeMonth – код m-го місяця n-го кварталу року;

Таким чином, стає можливим агрегування за вимірюваннями з рівнями ієрархій великої кількості. При цьому немає надмірності даних, яке виникає при денормалізації таблиці фактів у MOLAP і схемі «зірка» у ROLAP. З іншого боку, агрегування за вищими рівнями ієрархії не буде таким витратним за часом, як при використанні схеми «сніжинка» у ROLAP. Так, об'єктна модель даних може вирішити дві проблеми при проектуванні OLAP, а саме: проблему надмірності даних, що виникає при використанні ненормалізованих таблиць, і проблему збільшення кількості операцій при нормалізації бази даних, що призводить до зниження швидкості роботи системи в цілому.

Побудова багатомірних структур даних в об'єктно-орієнтованих СКБД за допомогою алгоритму DWARF. На даний час зв'язки багатомірних баз даних (MOLAP) і реляційних баз даних досліджено достатньо добре. Хай маємо множини (базові відношення):

$$D_1 = \{d_{11}, d_{12}, \dots, d_{1p}\}, D_2 = \{d_{21}, d_{22}, \dots, d_{2r}\}, \dots, D_n = \{d_{n1}, d_{n2}, \dots, d_{ns}\}.$$

Відношенням R вказаних множин називають підмножину декартового добутку

$$R = \{< d_{1i}, d_{2j}, \dots, d_{nk} > | V(d_{1i}, d_{2j}, \dots, d_{nk}) = true\} \subset D_1 \times D_2 \times \dots \times D_n,$$

де $i = \overline{1..p}$, $j = \overline{1..r}$, $k = \overline{1..s}$;

$< d_{1i}, d_{2j}, \dots, d_{nk} >$ – кортежі відношення *R*;

$V(d_{1i}, d_{2j}, \dots, d_{nk})$ – висловлювання, визначене на кортежах відношення *R*.

Таке відношення можна представити n-мірною матрицею інциденцій $p \times r \times \dots \times s$. У базах даних, крім самого факту наявності «відношення» між значеннями складових кортежу ($V(d_{1i}, d_{2j}, \dots, d_{nk}) = true$), такому кортежу зіставляються певні властивості q_{ijk}. Для бінарних відношень у мовах реляційних баз даних QBE і SQL реалізовані процедури (оператори) їх перетворення в двовимірні таблиці, які називають перехресними. У QBE – це запит дії CrossTab, у SQL – оператор Transform. Хоча зворотне відображення в реляційних СКБД безпосередньо не реалізовано, його можна реалізувати засобами навіть мови SQL [8]. Саме завдяки ізоморфному відображенню між багатовимірною і реляційною моделями даних OLAP-технології в промисловій реалізації з'явилися вперше саме в складі реляційних СКБД.

Як вже відмічалося, одним із альтернативних до реляційного підходу технологій баз даних є об'єктний підхід. Порівняно з реляційним, даний підхід з'явився дещо пізніше (приблизно середина 1980-х років). Об'єктний підхід дозволяє найбільш природним способом записати об'єкти в БД і забезпечити їх збереження і дії (процедури обробки) із ними. Крім того, можна встановити ізоморфне взаємне відображення між багатомірною та об'єктною моделями даних. Дійсно, будь який куб можна представити B-деревом із кількістю рівнів, що дорівнює кількості вимірів [9]. Хай, наприклад, маємо 3-мірний куб із вимірами *i, j, k* – M_{ijk}, де: $i = \overline{1..p}$, $j = \overline{1..r}$, $k = \overline{1..s}$. При представленні куба у вигляді дерева спочатку створюємо кореневу вершину, потім на 1-му рівні ієрархії дерева

розміщуємо p вершин дерева, що відповідають індексу i, на 2-му рівні – $p \times r$ вершин дерева, що відповідають індексам *(i, j)*, на 3-му рівні – $p \times r \times s$ вершин дерева, що відповідають індексам *(i, j, k)*. Так, для $p = 2$, $r = 3$ і $s = 2$ таке дерево буде виглядати так, як показано на рисунку 2.

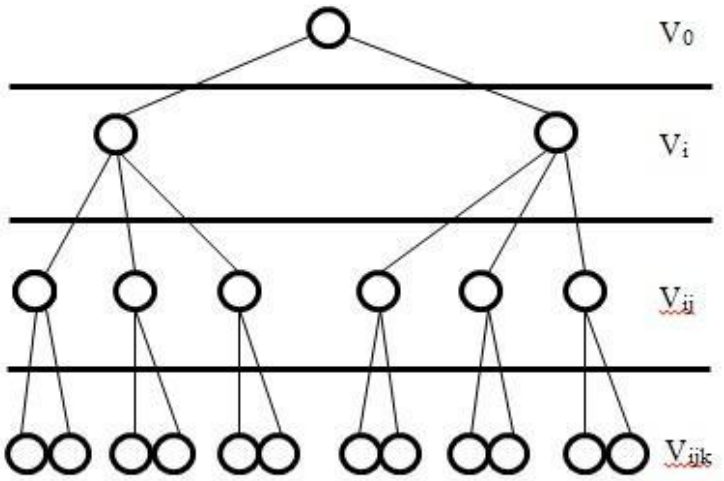

Рис. 2. Представлення куба у вигляді дерева

У висячих вершинах V_{ijk} знаходяться значення комірок куба M_{ijk}. Така ієрархічна структура добре вписується в ієрархічну модель даних, а також в ієрархічну діаграму класів об'єктної парадигми. На нижньому рівні розміщені змінні, над якими можна визначити певні дії, наприклад класичні операції ведення даних: *insert, delete, update, select*. На верхніх рівнях ієрархії в ролі змінних можна визначити певні агреговані значення підлеглого нижчого рівня, наприклад, класичні для баз даних агрегатні значення SUM, COUNT, AVERAGE тощо (узагальнено їх прийнято позначати через ALL), а діями – як функції обчислення агрегатних функцій, так і додаткові, що використовуються в OLAP.

Таким чином, між трьома типами моделей: реляційною, багатомірною й об'єктною – існують ізоморфні відображення одна в одну, що дозволяє, у принципі, реалізувати всі три названі моделі даних у рамках однієї СКБД. Такий підхід реалізовано в СКБД Caché [6]. При цьому дві моделі: об'єктна і реляційна – мають широке суспільне визнання, що вилилося у створення відповідних стандартів. Третя модель даних – це модель більш низького рівня абстракції, і хоча в документації Caché її називають багатовимірною, за своєю суттю вона є ієрархічною. Вона становить основу для реалізації як об'єктної, так і реляційної моделі даних у СКБД Caché і має вельми тривалу передісторію в розробках фірми Intersystems (розробник Caché), тому ця багатовимірна модель даних є для цієї СКБД «рідною».

Оскільки ієрархічну структуру покладено в основу як об'єктної, так і багатомірної моделей даних, то представляється можливість створювати OLAP-системи в об'єктно-орієнтованому середовищі. І хоча одним із компонентів СКБД Caché є система інтелектуального аналізу даних DeepSee, дослідження підходів до побудови OLAP-систем в об'єктних СКБД залишаються актуальними. Однією із основних проблем залишається обробка великих обсягів розріджених даних. У [9; 10] наведено низку алгоритмів збереження й обробки ієрархічних структур, що відображають куби розріджених даних. Авторами на продовження роботи [11] досліджено і реалізовано в середовищі СКБД Caché алгоритм багатопозиційного агрегування багатомірного масиву. Існує також інший алгоритм – DWARF, який можна віднести до синтаксичних алгоритмів зберігання OLAP-кубів [9]. Даний алгоритм розпізнає надмірності збереження даних та усуває їх при створенні куба. При цьому куб має вигляд дерева, глибина якого дорівнює кількості вимірів у таблиці фактів.

Основною перевагою даного алгоритму є те, що він може використовуватись як для щільних, так і для розріджених кубів. Якщо куб розріджений, то алгоритм DWARF стискує його синтаксично. Ніяких порожніх комірок у кубі не буде. Також відносно простим є зберігання даних, тому що кожна вершина містить вказівник на свого нащадка.

Але є в цього алгоритму і недоліки. Головним недоліком є неспроможність зробити обертання куба. Для того, щоб це зробити, необхідно заново створити інше дерево, маючи на його вершині вже інший вимір.

Структура куба, що відображає схему бази даних на рис. 1, у результаті виконання алгоритму DWARF матиме вигляд, що представлений на рисунку 3.

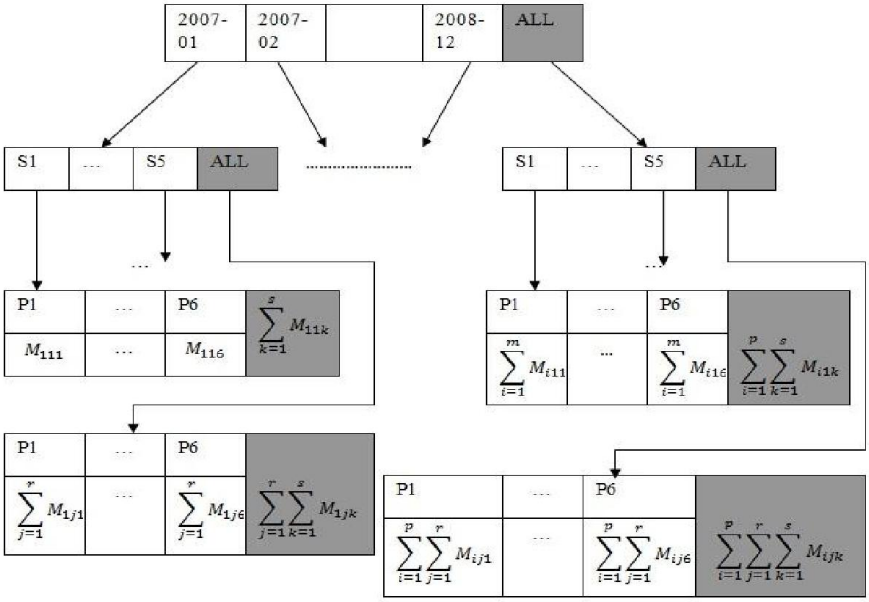

Рис. 3. Схема OLAP-куба у вигляді дерева для бази даних торговельної мережі

Куб, створений у результаті перетворення в деревоподібну структуру виду на рис. 2, являє собою ациклічний орієнтований граф з однією кореневою вершиною, який має n рівнів, де n – число вимірів, у даному випадку – 3.

Коренева вершина V_0 містить комірки виду {ключ, вказівник} для кожного значення першого виміру V_i. Вказівник кожної комірки спрямований до лежачої нижче вершини, що містить усі різні значення наступного виміру, асоційовані з ключем комірки.

Кожна вершина містить спеціальну комірку ALL, зображену сірою областю праворуч від вершини, що містить вказівник і відповідає узагальненим значенням вершини. У ролі агрегованої функції із загальною назвою ALL на рис. 3 використана сумма. Крім того, кожна вершина, в тому числі і листя, містить і комірку ALL, що містить агреговане значення всіх комірок вершини.

Кожне листя L має форму {ключ, агреговане значення} і містить агреговане значення всіх кортежів, які задовольняють шляху від кореня до L. Кожне листя містить і комірку ALL, що містить агреговане значення всіх комірок вершини.

Теоретично при повному (щільному) заповненні куба він буде мати $((p \times r \times s) + (p \times r) + p + 1)$ вершин, але при неповному (нещільному) заповненні куба пам'ять комп'ютера використовується неефективно, а час відповідей на запити може сповільнюватися. Тому алгоритм DWARF обробляє дерево таким чином, що відкидаються «пусті» вершини. Це нагадує процедуру прошивання дерева.

В алгоритмі DWARF можна виділити такі етапи.

1. Формування кореня дерева. При цьому створюються екземпляри класу *Root* у кількості існуючих значень у вимірів першого рівня (наприклад, кількість місяців для виміру «Дата»), значеннями їх властивості *dimvalue* при цьому будуть відповідні значення виміру.

2. Створення відповідних вузлів на наступних рівнях дерева та їх зв'язування з коренем дерева або кореневим вузлом на попередньому рівні.

3. Створення на останньому рівні листя дерева, їх зв'язування з кореневими вузлами та заповнення їх значеннями та агрегування.

4. Для кожного з вузлів передостаннього рівня формування єдиного агрегатного листя.

5. Формування агрегатного вузла для значення «ALL» у корені дерева.

6. Створення листя дерева для агрегатного вузла.

З точки зору об'єктно-орієнтованих СКБД алгоритм DWARF представляється достатньо корисним, оскільки структури даних, описані в даному алгоритмі, легко представити у вигляді класів, а двосторонні зв'язки між класами (в даному випадку це означає можливість пересування деревом не тільки від вершини до листя, а й навпаки) являються передумовою для виконання ефективних запитів даних з OLAP-кубу.

Структури даних куба у вигляді дерева можна представити такими класами: root – вершина, bundle – вузол та leaf – листя.

```
ClassRoot
{
        stringdimvalue; //значення виміру кореневого рівня
                relationshipset<Bundle>bundles
                inverseBundle:root; //зв'язок із вузлами наступного рівня
}

ClassBundle
{
        stringdimvalue; //значення виміру відповідного рівня вузла
        integer level; //рівень дерева
        relationshipRootroot
```

inverseRoot:bundles; //відповідний зв'язок із вершиною дерева, якщо вона є коренем вузла, тобто вузол розташований на 2 рівні

relationshipset<Bundle>daughtbundles

inverseBundle:rootbundle; //зв'язок із дочірніми вузлами

(вузлами наступного рівня дерева, якщо вони існують)

relationship set<Bundle>rootbundle

inverse Bundle:daughtbundles; //відповідний зв'язок із кореневим вузлом

relationship set<Leaf> leaves

inverseLeaf:rootbundle; //зв'язок із листями (структурами, які

мають певне значення міри куба)

floatall; //агреговане значення. Якщо вузол вказує не на листя, а на інший вузол, то дорівнює 0

}

ClassLeaf

{

stringdimvalue; //значення виміру останнього рівня дерева

relationship Bundle rootbundle

inverseBundle:leaves; //відповідний зв'язок із кореневим

вузлом листя

floatvalue; //відповідне значення комірки куба, що відповідає певним значенням вимірів, вказаних у вершині, вузлах та листі

}

На рисунку 4 наведена блок-схема алгоритму DWARF для багатомірних БД.

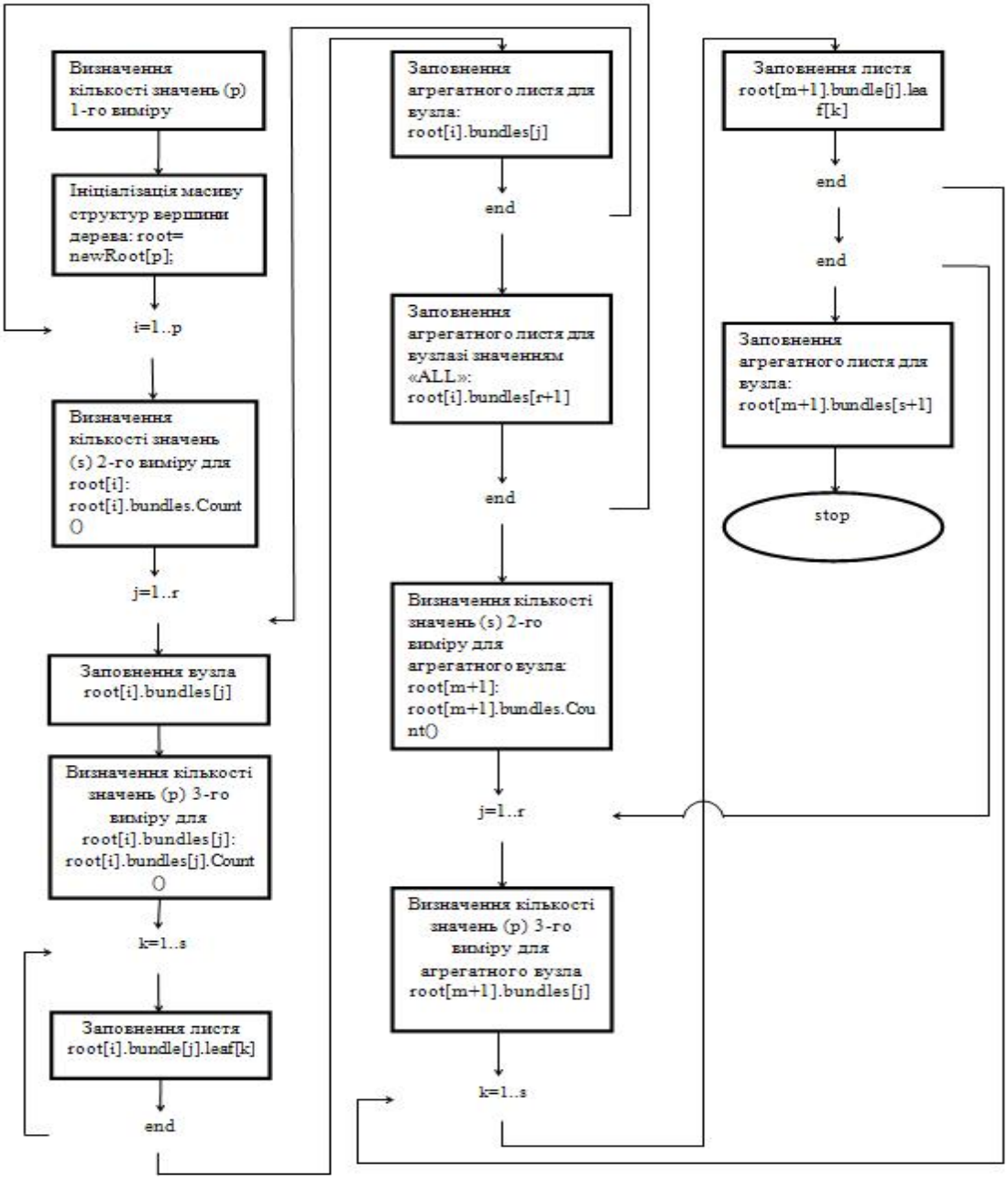

Рис. 4. Блок-схема алгоритму DWARF для куба з 3 вимірами

Висновки та напрями подальших досліджень. На даний час зв'язок OLAP та реляційних БД досліджений достатньо добре, і тому цілком доцільним є постановка питання про зв'язок даної технології з іншими видами БД. У статті розглянуто можливість поєднання OLAP з об'єктно-орієнтованими БД. Так об'єктна модель даних дозволяє усунути проблему надмірності даних та збільшити швидкодію системи шляхом зменшення кількості операцій, яких при використанні реляційних БД може бути достатньо багато.

Одним з варіантів створення OLAP-куба в об'єктно-орієнтованих БД є його представлення у вигляді дерева. Одним з алгоритмів побудови кубу в такому вигляді є алгоритм DWARF, який дозволяє синтаксично стиснути куб. Даний алгоритм ідеально підходить до розріджених кубів, однак його з тим же успіхом можна використати і для щільних кубів.

В подальшому планується проектування підсистеми інтелектуального аналізу даних (Data Mining), яка представляє методи пошуку асоціативних правил у OLAP-кубах в середовищі об'єктно-орієнтованої СКБД.

ЛІТЕРАТУРА

1. Барсегян А. А. Методы и модели анализа данных : OLAP и Data Mining / А. А. Барсегян, М. С. Куприянов, В. В. Степаненко, И. И. Холод. – СПб. : БХВ-Петербург, 2004.
2. Shashank Tiwari. Professional NoSQL. Indianapolis. John Wiley & Sons, Inc. – 2011. – 480 p.
3. Дейт К. Дж. Введение в системы баз данных / К. Дж. Дейт. – Киев – Москва : Диалектика, 1998. – 787 с.
4. Паклин Н. Б. Бизнес-аналитика: от данных к знаниям : [учебное пособие] / Н. Б. Паклин, В. И. Орешков. – 2-е изд., перераб. и доп. – СПб. : Питер, 2010.
5. Харинатх С. и др. Microsoft SQL Server Analysis Services 2008 и MDX для профессионалов. : [пер. с англ.] / С. Харинатх. – М. : ООО «И. Д. Вильямс», 2010.
6. Кирстен В. Постреляционная СУБД Cache 5. Объектно-ориентированная разработка приложений / В. Кирстен, М. Ирингер, М. Кюн, Б. Рериг. – 3-е изд., перераб. и дополн. – М. : ООО «Бином-Пресс», 2008.
7. Харрингтон Д. Проектирование объектно-ориентированных баз данных : пер. с англ. / Д. Харрингтон. – М. : ДМК Пресс, 2001.
8. Фісун М. Т. Алгоритми та програми згортки перехресних таблиць в реляційні таблиці / М. Т. Фісун // Наукові праці. Серія Комп'ютерні технології. – Том 35, Випуск 22. – Миколаїв : Вид. МДГУ, 2004. – С. 133–138.
9. Кудрявцев Ю. Обзор алгоритмов MOLAP [Электронный ресурс] / Ю. Кудрявцев. – 2008. – Режим доступа : http://www.citforum.ru/consulting/BI/molap_overview/node2.shtml.
10. Кормен Т. Алгоритмы: построение и анализ / Т. Кормен [и др.]. – 2-е изд. : пер. с англ. – М. : Издательский дом «Вильямс», 2005.
11. Фісун М. Т. Аналіз особливостей об'єктної та багатовимірної моделей даних в СКБД Caché / М. Т. Фісун, Г. В. Горбань //Вестник Херсонского национального технического университета. – 2011. – № 2 (41). – С. 116–124.

Рецензенти: Мусієнко М. П., д.т.н., професор;
Батрак Ю. А., к.т.н., доцент.

Дата надходження статті до редколегії 10.05.2013 р.

ФІСУН Микола Тихонович, д.т.н., професор, завідувач кафедри інтелектуальних інформаційних систем Чорноморського державного університету імені Петра Могили. Сфера наукових інтересів: інтелектуальні інформаційні системи та CASE-засоби їх створення, системи автоматизованого проектування, бази даних та бази знань.

ГОРБАНЬ Г. В., аспірант, Чорноморський державний університет імені Петра Могили, м. Миколаїв, Україна.