12. The World's 50 Most Innovative Companies [e-resource]: (Periodical e-publication) // Business Week. – 17 April 2008. – Available at: http://bwnt.businessweek.com/interactive_reports/innovative_companies/

13. Duet for Microsoft Office and SAP [e-resource]: (Sap community network portal). – [Last visited: 5 October 2009]. – Available at:http://www.sdn.sap.com/irj/sdn/go/portal/prtroot/docs/webcontent/uuid/40e264f4-db9c-2910-cd8c-912e80e2a41a

R. Buniak

OVERVIEW OF THE NEW TREND IN THE SOFTWARE ENGENEERING BASED ON THE DESIGN THINKING APPROACH

Nowadays, one of the key differentiators in the IT-business market is the ability to innovate, constantly produce new ideas. Design thinking is one of the new and most perspective approaches for this. In comparison with a classic analytical thinking, it operates on a multidisciplinary level, is based on the user's demands and is characterized by searching for simplicity and creativity in solutions. Design thinkers analyze any new task from the three points of view: business viability, technical feasibility and users' desirability. The challenge solving process in the proposed method usually consists of the three main stages: analyzing the problem, synthesizing and iterative prototyping. Business experience of many famous corporations (like Google, Microsoft, Amazon, and others) shows the high effectiveness and productivity of the design thinking approach usage, as only due to the constant innovative solutions these companies are leaders in their business sphere.

УДК 004.042

Апостол А. В.

СИСТЕМИ ОПЕРАТИВНОГО АНАЛІЗУ ДАНИХ

У цій статті розглянуто концепції технології OLAP та систем, що її застосовують. OLAP – технологія обробки інформації, яка передбачає складання і динамічну публікацію звітів і документів, її використовують аналітики для швидкої обробки складних запитів до бази даних. В роботі зроблено порівняння систем OLAP та OLTP, висвітлено методики, які пришвидшують роботу зі складними запитами. Детально проаналізовано технологію OLAP кубу як одну з таких методик. Акцентовано на особливому способі зберігання даних, стовпчикових базах даних. Такий вид збереження інформації надає ряд переваг під час обробки складних запитів, порівняно з традиційним рядковим представленням. Цю теорію доводить приклад, в якому порівнюється швидкодія C-Store, однієї з найпопулярніших стовпчикових БД, та комерційної рядкової БД.

Схеми баз даних

Кількість даних у глобальних мережах стрімко зростає практично щодня. На початку сучасної комп'ютерної ери файлові сховища задовольняли усі потреби людей щодо зберігання інформації. Потужності таких сховищ вистачало для того, аби обробляти інформацію з відносно високою швидкістю. Але уже через декілька років цей метод був неприйнятним. Створення систем керування базами даних (СКБД) свого часу стало значним проривом у галузі зберігання і обробки інформації. СКБД дали можливість оперувати величезними масивами даних, задовольняючи прийнятні часові обмеження. Навіть сьогодні СКБД залишаються основним напрямом розвитку сфери зберігання даних. В основі традиційних СКБД лежали таблиці рядкової структури. Рядкова база серіалізує всі значення одного рядка, потім значення іншого і так далі. Цей підхід вважається досить практичним, якщо мова йде про вставлення чи оновлення даних. Такі бази, оптимізовані для оновлення, називають **операційними базами даних,** і викорис-

товують у **системах оперативної обробки транзакцій (OLTP)** [1, 2]. Проте сучасні інформаційні технології тяжіють більше до зчитування даних, аніж до їх запису. Згадати лише про системи прийняття рішень та інші аналітичні застосування, де значення зчитуються у різних комбінаціях з багатьох таблиць для того, аби сформувати кінцевий масив необхідних даних. Бази, оптимізовані для читання, називають **сховищами даних.** Вони застосовуються у **системах оперативного аналізу (OLAP)** [2] (Див. табл. 1).

Таблиця 1. **Порівняння систем OLAP та OLTP**

OLTP	OLAP
Наперед визначені запити	Спонтанні запити
Прості запити	Складні запити
Високоселективні терміни запитів – невеликі набори в результаті пошуку	Низькоселективні терміни запитів – великі набори в результаті пошуку
Детальне видобування рядків – видобування великої кількості стовпчиків	Агрегація та групування – видобування невеликої кількості стовпчиків
Оновлення та видобування	Здебільшого видобування
Оновлення в реальному часі	Заплановані оновлення
Короткі транзакції	Довгі транзакції
Велика кількість конкурентних користувачів (1000+) – велика кількість транзакцій	Невелика кількість користувачів (100+) – небагато транзакцій

Проблеми OLTP

Бази даних OLTP нормалізовані. Нормалізація дає ряд переваг для обробки транзакцій (швидкі вставлення, оновлення та видалення). Однак саме вона й викликає проблеми, пов'язані зі зчитуванням даних:

- Порівняно з обробкою однієї великої таблиці, з'єднання (joins) та запити, що використовують декілька таблиць, виконуються повільно.
- Кількість індексів у таблиці недостатня для здійснення швидкої вибірки.
- Дані організовані таким чином, що спонтанні запити від користувачів з нетехнічною освітою не підтримуються (наприклад, для того, щоб отримати звіт по базі даних, необхідно виконати багато з'єднань таблиць).

Нормалізація – одна з основних причин, що роблять неможливим використання баз даних OLTP у системах, спрямованих на читання даних [4].

Для того, аби вирішити ці проблеми, OLAP пропонує багатовимірну схему (**куби**), яка:

- Денормалізована для пришвидшення роботи з запитами.
- Більше спрямована на кінцевого користувача, аніж традиційні схеми представлення даних.

OLAP куби

OLAP куб – багатовимірне відображення даних (наприклад, продукти переглядаються за регіоном, поділом чи групою покупців). Це підхід полягає у відображенні даних у вигляді кубу, де кожна грань відповідає єдиному виміру (регіон, поділ, група) [5].

Аналітик може зрозуміти значення, що містяться в базі даних, використовуючи багатовимірний аналіз. При зіставленні змісту даних з власною ментальною моделлю ймовірність непорозуміння чи неправильного трактування інформації значно знижується. Аналітик може переміщуватися базою, здійснювати пошук певної підмножини даних, змінюючи їх орієнтацію і виконуючи певні обчислення. Процес пошуку даних, ініційований користувачем, з використанням поворотів, проникнення вглиб і вгору часом називають «Slice and Dice». Стандартні операції включають зріз (slice), багатошаровий зріз (dice), проникнення вглиб (drill down), згортку (roll up), а також вісь повороту (pivot) [3]:

- *зріз*: це підмножина багатовимірного масиву, що відповідає одному значенню для одного чи кількох членів вимірів, які не потрапили до підмножини;
- *багатовимірний зріз*: операція багатовимірного зрізу – це зріз по більше, ніж двох вимірах кубу даних;
- *проникнення вглиб/вгору*: ця операція є спеціальним аналітичним методом, за якого користувач проглядає різні рівні даних, починаючи з найбільш загального і закінчуючи найбільш деталізованим (згори донизу), або навпаки;
- *згортка*: при згортці обчислюються всі відношення між даними для одного або декількох вимірів. Для цього може бути наперед визначена формула;
- *вісь повороту*: використовується для зміни орієнтації виміру.

Технічне визначення OLAP кубу

У теорії баз даних **OLAP куб –** абстрактне представлення проекції відношення реляційної бази даних. За такого відношення порядку N проекцію X, Y та Z можна вважати ключем, а W – залишковим атрибутом. Запишемо це відношення у вигляді функції:

$$W: (X, Y, Z) \rightarrow W,$$

де X, Y та Z відповідають осям куба, а значення W, в яке переходять всі триплети (X, Y, Z), відповідає елементові даних, розміщеному в кожній клітинці кубу [3].

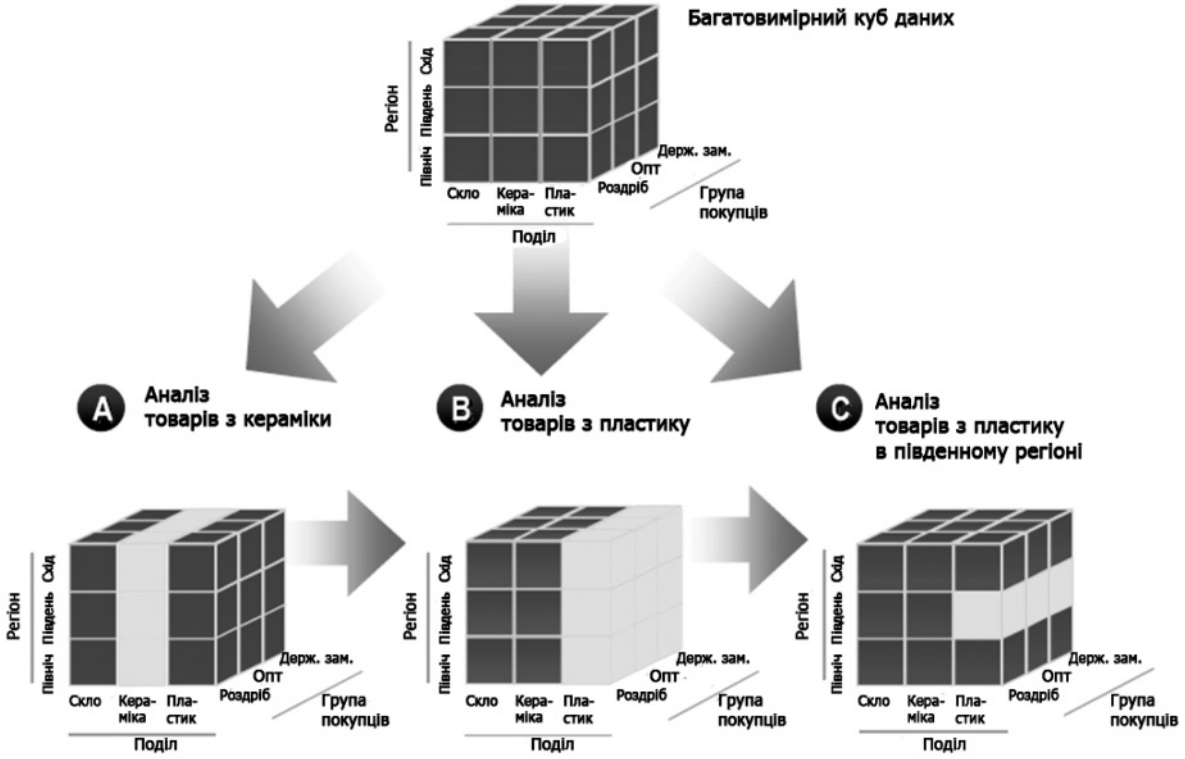


Рис. 1. Приклад багатовимірного аналізу товарів за допомогою OLAP кубу

Оскільки двовимірні пристрої виводу не можуть охопити чотирьох вимірів, доцільно проектувати так звані зрізи кубу даних (проектувати – в класичному векторно-аналітичному сенсі зменшення вимірності, а не в сенсі SQL).

$$W: (X, Y) \rightarrow W.$$

Це може затіняти первинний ключ, однак має деяку семантичну важливість – зріз потрійного функціонального представлення по значенню Z, яке й цікавить аналітика.

Мотивація, що стоїть за представленням OLAP, нагадує парадигму звіту з перехресними посиланнями СКБД 1980'х років. Аналітик може бути зацікавленим у виведенні електронної таблиці, де значення X містяться в рядку $1; значення Y розташовані в стовпчику $A; а значення $W: (X, Y) \rightarrow W$ розміщені в індивідуальних комірках «південно-східніше» від $B2, умовно кажучи. При цьому, $B2 включається. Звичайно, для того, аби відобразити (X, Y, W) триплети, можна використовувати DML (Data Manipulation Language) традиційного SQL. Однак цей формат виводу не такий зручний, як його альтернатива – звіт з перехресними посиланнями. Адже в першому випадку необхідно здійснити лінійний пошук даної пари (X, Y), щоб визначити відповідне значення W, а в другому – просто здійснити пошук перетинів шуканих стовпчика X та рядка Y.

Зберігання атрибутів у стовпчиках

Традиційні рядкові системи керування базами даних дуже повільно працюють, коли мова йде про обробку значної кількості запитів на вибірку. Це поклало початок розвитку **стовпчикових СКБД** оптимізованих для читання. На відміну від традиційних рядкових баз даних, стовпчикові бази серіалізують значення одного стовпчика, потім іншого і так далі. Розглянемо приклад:

Номер	Ім'я	Прізвище	Зарплата
1	Василь	Коваленко	65000
2	Іван	Петренко	40000
3	Степан	Омельченко	75000

Рядкова база даних зберігатиме цю таблицю так:

1, Василь, Коваленко, 65000; 2, Іван, Петренко, 40000; 3, Степан, Омельченко, 75000.

На відміну від такого представлення, в стовпчиковій базі ці ж дані будуть зберігатися так:

1, 2, 3; Василь, Іван, Степан; Коваленко, Петренко, Омельченко; 65000, 40000, 75000.

Виникає питання, чи дійсно такий спосіб ефективніший? Безперечно, так. Для того, аби

знайти всі екземпляри атрибута в класичній базі даних, необхідно перевіряти кожен рядок на значення цього атрибута. Що ж до стовпчикової бази даних, до рядкових величин у стовпчику атрибутів можна доступатися напряму [1].

Існує кілька концепцій, пов'язаних з оптимізацією стовпчикових сховищ. Перш за все, вважається, що такі сховища повинні забороняти можливість оновлення величин, а забезпечувати лише можливість вставлення нових (підхід **insert-only**)**.** Запити на оновлення в стовпчикових базах даних вимагають досить багато ресурсів. Це пов'язано зі способом зберігання даних. Як альтернатива запитам оновлення, пропонується використовувати запити вставлення: у випадку, коли необхідно змінити якийсь запис, натомість додається новий, який відрізняється від попереднього лише часовою міткою. Такий підхід має певні переваги. Насамперед, це можливість досить легкого створення журналу змін [6]. Проте ця концепція створює й деякі проблеми. Необхідно брати до уваги постійне зростання кількості даних у базі і своєчасно виконувати видалення непотрібних записів (наприклад, у процесі резервного копіювання бази). Така процедура буде нагадувати функцію «збирач сміття» деяких мов програмування.

Інше вдосконалення полягає у розміщенні бази даних в оперативній пам'яті (підхід **main-memory**)**.** Те, що доступ до оперативної пам'яті значно швидший, аніж доступ до дискового сховища, ні у кого не викликає сумнівів. Тому ідея розміщення БД в ОЗП досить перспективна. Звичайно, реалізуючи цю ідею, можна зіткнутися з рядом проблем, найбільша з яких – об'єм сховища. Кількість даних може бути дуже великою, а об'єм оперативної пам'яті обмеженим. Для того, аби уникнути цієї проблеми, використовують певні технології. Перш за все, пропонується вивантажувати в оперативну пам'ять не повну базу, а деяку підмножину її даних (набори стовпчиків, грані OLAP кубу), яка необхідна для виконання запитів на даний момент часу [6].

Серед теоретиків баз даних існує також ідея проведення компресії даних. Безперечно, стовпчикові бази пристосовані для стискання інформації. Такі БД досить однорідні, а це відкриває значні перспективи для оптимізації даних, які були недоступні в рядкових БД. Наприклад, багато сучасних алгоритмів стискання (таких, як LZW) пристосовані для компресії однорідних даних. В рядкових базах даних ці алгоритми не досягають такого значного ступеню стискання. Ще більшого ефекту можна досягнути, якщо великий відсоток сусідніх рядкових величин однаковий або відсутній (розріджені стовпчики). Проте великий відсоток стискання може бути й недоліком. Чим вищий рівень компресії, тим більш проблемним є читання такої інформації з оперативної пам'яті, адже дані необхідно піддати декомпресії, аби їх прочитати. Тому стовпчикові архітектури дуже часто оптимізовують за допомогою додаткових механізмів, які спрямовані на мінімізацію потреби доступу до стиснутої інформації [1].

Порівняння швидкодії стовпчикової та рядкової СКБД

Наведемо приклад порівняння швидкодії стовпчикових та рядкових сховищ даних на базі SSBM (Star Schema Benchmark) [7, 8]. Це тип тестування швидкодії сховищ даних, який використовує чисту схему структури даних типу «зірка» і складається з ряду запитів, що дають змогу оцінити можливості СКБД з різних боків. Проводилося порівняння однієї з найпопулярніших стовпчикових БД, C-Store, та комерційної рядкової СКБД.

Схема має такий вигляд, див. рис. 2:

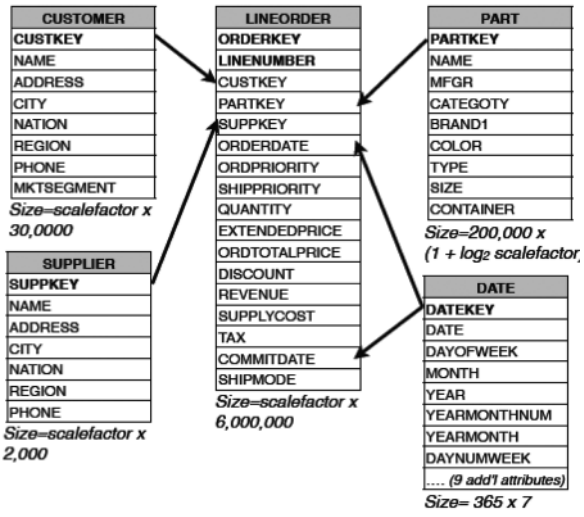
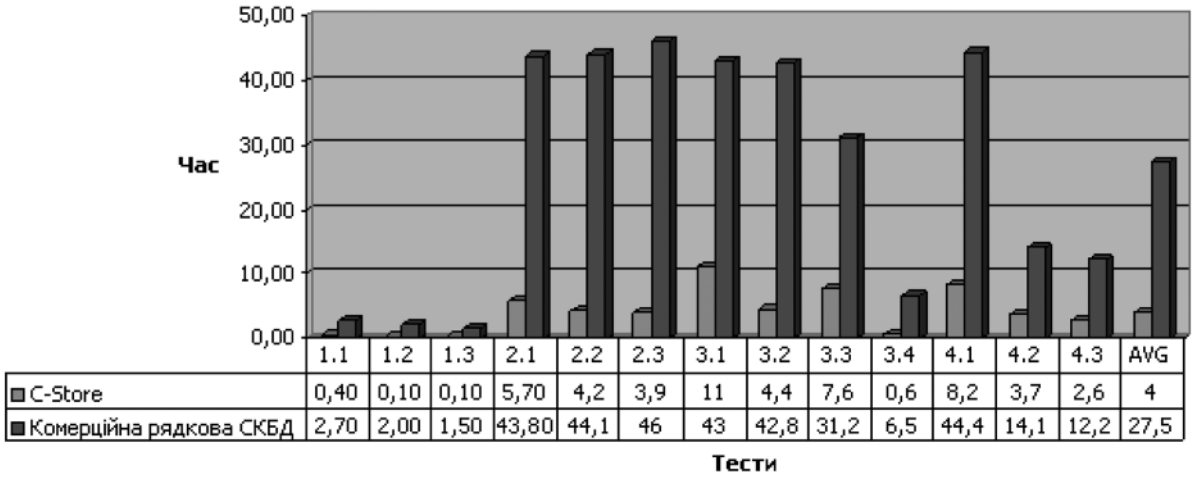


Рис. 2. Схема тестування SSBM

Основна таблиця – LINEORDER, що містить інформацію про індивідуальні замовлення і має складний первинний ключ (ORDERKEY і LINENUMBER). Інші атрибути в цій таблиці також включають зовнішні ключі, що посилаються на таблиці CUSTOMER, SUPPLIER, PART та DATE (для дати замовлення та дати вручення), та інші атрибути замовлення (ціна, пріоритет, кількість тощо). Scalefactor – спеціальна величина, що використовується для масштабування розміру тесту. В цьому прикладі застосовується scalefactor = 10, а основна таблиця має розмір 60000000 кортежів.

Запити:

Тест 1. Складається з трьох запитів. Запити, а також стовпчики DISCOUNT та QUANTITY



	1.1	1.2	1.3	2.1	2.2	2.3	3.1	3.2	3.3	3.4	4.1	4.2	4.3	AVG
C-Store	0,40	0,10	0,10	5,70	4,2	3,9	11	4,4	7,6	0,6	8,2	3,7	2,6	4
Комерційна рядкова СКБД	2,70	2,00	1,50	43,80	44,1	46	43	42,8	31,2	6,5	44,4	14,1	12,2	27,5

Рис. 3. Результати порівняння C-Store та комерційної рядкової СКБД за схемою SSBM

таблиці LINEORDER, мають обмеження на одновимірні атрибути. Запити спрямовані на виявлення приросту прибутку (продукт з EXTENDEDPRICE та DISCOUNT), який може бути досягнутий, якщо усунути різні рівні знижок для різних кількостей товарів на даний рік. Селективність таблиці LINEORDER для цих трьох запитів складає відповідно 1.9×10^{-2}, 6.5×10^{-4}, 7.5×10^{-5}.

Тест 2. Складається з трьох запитів. Запити мають обмеження на двовимірні атрибути, і вираховують прибуток для певних класів продуктів у певних регіонах, а також групують результат за класом та роком. Селективність LINEORDER для цих трьох запитів вибрана відповідно 8.0×10^{-3}, 1.6×10^{-3}, 2.0×10^{-4}.

Тест 3. Складається з чотирьох запитів, що мають обмеження на три виміри. Запити визначають прибуток в конкретному регіоні за певний проміжок часу, а також групують результат за національністю покупця, національністю постачальника та роком. Селективність LINEORDER для запитів така: 3.4×10^{-2}, 1.4×10^{-3}, 5.5×10^{-5}, 7.6×10^{-7}.

Тест 4. Містить три запити. Запити обмежені трьома вимірами, і обраховують дохід (REVENUE – SUPLY – COST), та групують результат за роком, національністю та категорією (для запиту 1); та за регіоном і категорією (для запитів 2, 3). Селективність таблиці LINEORDER для цих трьох запитів складає відповідно 1.6×10^{-2}, 4.5×10^{-3}, 9.1×10^{-5}.

Тестування дало такий результат, див. Рис. 3: [10].

Як бачимо, стовпчикові СКБД дають можливість досягти пришвидшення роботи з даними, використовуючи запит на вибірку в середньому в 7 разів. Проте відомі випадки, що таке пришвидшення сягало x100 [9].

Висновки

У наш час системи OLAP набувають дедалі більшої популярності. Вони спрямовані на обробку набагато більшої кількості запитів на вибірку, аніж запитів на вставлення. Традиційні транзакційні бази даних мають високий рівень нормалізації, що сповільнює обробку запитів, які використовують декілька таблиць або їх з'єднань одночасно. Аби розв'язати цю проблему, OLAP запропонував багатовимірну схему – OLAP куб, що втілює своєрідний багатовимірний погляд на дані (наприклад, продукти можуть переглядатися за типом, часом чи географією). Дані у ньому представлені у вигляді кубу, кожна грань якого відповідає певному виду (регіон, поділ, група тощо).

Власне, зберігання атрибутів також повинне бути оптимізованим для читання. Пропонується їх зберігати у вигляді стовпчиків. Цей вид баз даних є новим підходом до збереження даних і повинен мати такі характеристики:

- Забороняти оновлення даних; натомість використовувати вставляння.
- Мати стовпчиковий вигляд.
- Розміщувати дані в оперативній пам'яті.

Для збільшення швидкодії та економії пам'яті необхідно втілювати механізми ефективної компресії/декомпресії даних. Правильно побудована стовпчикова СКБД дає можливість досягти пришвидшення виконання запитів на вибірку (порівняно з рядковими СКБД) в середньому в 7–10 разів.

1. A DBMS for Large Statistical Databases / [Turner, Hammond, Cotton] ; Proceedings of VLDB 1979 – Rio de Janeiro, Brazil.

2. Plattner H. Datawarehouses / Prof. Dr. Hasso Plattner; [Trends and Concepts Lectures]. – Potsdam : HPI, 2008.

3. Mailvaganam H. Introduction to OLAP / H. Mailvaganam. – Slice, Dice and Drill / Hari Mailvaganam (2007). DWreview. Retrieved 2008-03-05.
4. Plattner H. Online Analytical Processing / Prof. Dr. Hasso Plattner; [Trends and Concepts Lectures]. – Potsdam : HPI, 2008.
5. Plattner H. OLAP : Slicing and Dicing / Prof. Dr. Hasso Plattner; [Trends and Concepts Lectures]. – Potsdam : HPI, 2008.
6. Plattner H. Compressed and Optimized OLAP : Storing Attributes in Columns / Prof. Dr. Hasso Plattner; [Trends and Concepts Lectures]. – Potsdam : HPI, 2008.
7. Adjoined Dimension Column Index (ADC Index) to Improve Star Schema Query Performance / [P. E. O'Neil, X. Chen, E. J. O'Neil] ; In ICDE, 2008.
8. The Star Schema Benchmark (SSB) / [P. E. O'Neil, E. J. O'Neil, X. Chen]. – Available at: http://www.cs.umb.edu/poneil/StarSchemaB.PDF.
9. «Column Oriented Database» / Harvard Research Group. – Available at: http://www.hrgresearch.com/ColumnDB.html
10. ColumnStores vs. RowStores: How Different Are They Really? / [Daniel J. Abadi, Samuel R. Madden, Nabil Nachem]. – Available at: http://db.csail.mit.edu/projects/cstore/abadi-sigmod08.pdf.

A. Apostol

ONLINE ANALYTICAL PROCESSING SYSTEMS

In this article, the review of the OLAP technology is presented. OLAP is the technology of the information processing that includes the composing and dynamic publication of the reports and documents. It is used by the analysts for the fast processing of the complex database queries. The comparison of OLAP and OLTP, and the methods that increase the performance of the processing of complex queries are also presented in this work. As one of these methods, the technology of the OLAP cube is thoroughly examined. Much attention is also paid to the special type of data stores, column-oriented databases. In comparison to the traditional row-oriented approach to the storing of data, column-based databases give plenty of advantages while processing the complex queries. This theory is proved by the example in which the performance of C-Store, one of the most popular column-oriented DBMSs, and the performance of the commercial row-oriented DBMS are compared.

УДК 681.3:658.5

Олецький О. В.

ОРГАНІЗАЦІЯ ОНТОЛОГІЧНО-ОРІЄНТОВАНИХ ЗАСОБІВ АВТОМАТИЗОВАНОГО ЕКСПЕРТНОГО ДОБОРУ ІНФОРМАЦІЙНИХ РЕСУРСІВ НА ТЕМАТИЧНОМУ ПОРТАЛІ

Проблему експертного добору інформаційних ресурсів розглянуто в аспекті онтологічно-орієнтованого пошуку на базі системи математичних співвідношень над вузлами онтології в рамках моделі «онтологія–артефакт–користувач–проект». Висвітлено основні компоненти системи, що реалізують такий пошук.

Вступ

Проблема пошуку інформаційних ресурсів, які б найбільш точно відповідали цілям користувача, зважаючи на його індивідуальні особливості, є винятково важливою й актуальною. Протягом тривалого часу розвиваються різні погляди як її розв'язати [1–3 та ін.], але вони здебільшого мають евристичний характер і залучають до розгляду лише окремі аспекти пошуку. Навіть такі ключові поняття, як «релевантність документа запитові», «схожість документів», формалізовано недостатньо.

Окрему увагу слід звернути на веб-орієнтовані інформаційні ресурси, для яких характерні висока інформаційна зв'язність, тематична однорідність, достатньо висока структурованість та якість інформаційного наповнення. До таких ресурсів належать, зокрема, тематичні портали,