УДК 004.6

ТОМАШЕВСЬКИЙ В.М.,
ЯЦИШИН А.Ю.

МАТЕМАТИЧНА МОДЕЛЬ ЗАДАЧІ ПРОЕКТУВАННЯ ГІБРИДНИХ СХОВИЩ ДАНИХ З ВРАХУВАННЯМ СТРУКТУР ДЖЕРЕЛ ДАНИХ

У статті розглядається питання формулювання математичної моделі задачі проектування гібридних сховищ даних (ГСД). Приводиться опис існуючих рішень та описується придатність їх застосування. Пропонується математична модель, яка включає опис джерел даних і сховища даних, а також оптимізаційних параметрів та рівнянь.

In paper mathematical model formulation problem for hybrid data warehouse (HDW) building is discussed. An overview of existing solutions is described and their applicability is stated. Author introduces a mathematical model which includes data sources and warehouses description. Also, this model includes optimization parameters and equations.

Вступ

При проектуванні сховищ даних постає питання коректної побудови структур даних для ефективного його функціонування.

В різних наукових публікаціях були запропоновані різні математичні моделі для формалізації задачі проектування сховищ даних. Однак модель, яка б враховувала особливості задачі проектування гібридних сховищ з врахуванням структур джерел даних, запропонована не була.

При цьому, актуальним та невирішеним на сьогодні є питання автоматизованого чи напівавтоматизованого проектування гібридних сховищ даних у цілому. Ця проблема випливає з практичної необхідністі автоматизованої побудови сховищ даних з оптимальною швидкодією за вибраним критерієм оптимальності сховища.

Ціллю статті є запропонувати математичну модель для задачі автоматизованого проектування гібридних сховищ даних, б враховувала і розміщення даних у сховищі, і оптимізаційні фактори (матеріалізовані подання, індекси, схеми розбиття), частоту виконання запитів до сховища та структури джерел даних.

Питаннями застосування генетичних алгоритмів до побудови сховищ даних займалися та оптимізації сховищ даних Wen-Yang Lin,I-Chung Kuo [1], Chuan Zhang, Xin Yao, Jian Yang [2,5], Ladjel Bellatreche,Kamel Boukhalfa [3], J.-T. Horng, Y.-J. Chang, B.-J. Liu [4], Goran Velinov, Danilo Gligoroski, Margita Kon Popovska [6], Michael Lawrence [7], Бакуліна М.А. [8]

У праці [9] розглянуто основні принципи побудови та функціонування сховищ даних. У статтях [10] і [11] розглядаються питання проектування гібридних сховищ даних.

Виклад основного матеріалу

У дисертації Бакуліної М.А. [8] розглядаються методи та алгоритми автоматизації проектування структур сховищ даних для аналітичної обробки числових показників. У роботі ставляться цілі прискорення процесу проектування сховища даних та підвищення швидкості аналітичної обробки даних сховища даних.

Розглядаються наступні задачі :

1) Розробка єдиної математичної моделі бази даних і сховища даних
2) Розробка математичної моделі багатомірного сховища даних
3) Розробка математичної моделі операцій над багатомірним сховищем даних
4) Розробка математичної моделі структури даних в сховищі даних, що відповідає вимогам OLAP-систем по швидкодії
5) Розробка алгоритмів, що автоматизукть процес побудови сховищ даних на основі запропонованих моделей
6) Розробка алгоритмів OLAP на основі запропонованої структури
7) Розробка програмної системи, що здійснює автоматизацію проектування сховища даних і оперативний аналіз реляційного сховища даних на основі запропонованих алгоритмів.

Для розв'язання вишенаведених задач застосовуються методи тензорної алгебри, кратномасштабного аналізу, вейвлет-перетворень і сигнатуроного пошуку.

Математична модель

Математична модель сховища базується на тензорній моделі.

Відношення формалізуються наступним правилом:

1) відношенню $R(A_1, A_2, ..., A_n, B_1, B_2, ..., B_K, C_1, C_2, ..., C_m)$ R(Ax,A2 присвоється довільна велика літера, наприклад R;

2) n ключовим атрибутам $A_1, A_2, ..., A_n$ присвоється n різних довільних индексів, например $a_1, a_2, ..., a_n$;

3) атрибутам $B_1, B_2, ..., B_K$ и $C_1, C_2, ..., C_m$ присвоється $(k+m)$ різних індексів, що відмінних від індексов п.2, наприклад $b_1, b_2, ..., b_k, c_1, c_2, ..., c_m$;

4) $a_1, a_2, ..., a_n$ будуть відповідати коваріантним (нижнім) індексам тензору, що визначають вхідний потік даних, $b_1, b_2, ..., b_k, c_1, c_2, ..., c_m$ будуть відповідати контраваріантним (верхнім) індексам тензору, що визначають вихідний потік даних (див. рисунок 1); тензор буде мати вигляд $R^{b_1 b_2 ... b_k c_1 c_2 ... c_m}_{a_1 a_2 ... a_n}$

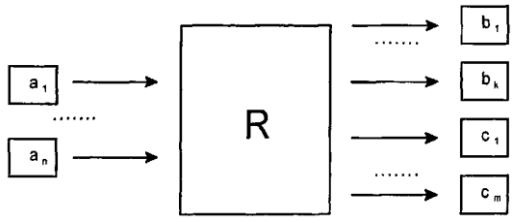

Рис. 1. Графічне представлення тензору

Запити до сховища даних формалізуються наступним правилом.

Поток запитів до сутності сутності бази даних позначається e^, де під «^» розуміється індекс (індекси), що позначають вхідні дані запиту. В загальному випадку e можно описати матрицею, в якій одиничні елементи розміщені на позиціях, що дорівнюють значенню «^».

Тензорна модель запиту описується виразом:

$$e^{jl} * P^k_{jl} = e^k, \quad (1)$$

яка має наступне матричне відображення:

$$\left\| e_{jl} \right\|_{n \times n} \times \left\| k_{jl} \right\|_{n \times n} = \begin{Vmatrix} 0 & 0 & . & 0 \\ 0 & 0 & . & 0 \\ . & . & 1_{jl} & . \\ 0 & 0 & . & 0 \end{Vmatrix} \times \begin{Vmatrix} k_{11} & k_{12} & . & k_{1n} \\ k_{21} & k_{22} & . & k_{2n} \\ . & . & k_{jl} & . \\ k_{n1} & k_{n2} & . & k_{nn} \end{Vmatrix} = k_{jl} \quad (2)$$

Якщо два тензори $R^{b_1 b_2 ... b_k c_1 c_2 ... c_m}_{a_1 a_2 ... a_n}$ та $U^{d_1 d_2 ... d_q}_{b_1 b_2 ... b_k}$ містять однойменні індекси $b_1, b_2, \ldots, b_k$, причому у тензору $R^{b_1 b_2 ... b_k c_1 c_2 ... c_m}_{a_1 a_2 ... a_n}$ індекси $b_1, b_2, ..., b_k$ – контраваріантні, а у тензора $U^{d_1 d_2 ... d_q}_{b_1 b_2 .. b_k}$ – коваріантні, то відношення $U(B_1, B_2, ..., B_k; D_1, D_2, ..., D_q)$ знаходиться в зв'язку один до одного з відношенням $R(A_1, A_2, ..., A_n; B_1, B_2, ..., B_k; C_1, C_2, ..., C_m)$ та індекси $b_1, b_2, ..., b_k$ в тензорі $R^{b_1 b_2 ... b_k c_1 c_2 ... c_m}_{a_1 a_2 ... a_n}$ позначають зовнішні ключі.

У моделі також визначені наступні операції над багатомірним сховищем даних:

1) зріз:

$$s_a(Z^k_{adx}) = Z^k_{a_m dx}, \quad 3)$$

де a_m – елемент деякої множини $a=\{a_1, a_2, a_3, ..., a_m, ..., a_n\}$.

2) обертання:

$$s_a(Z^k_{adx}) = Z^k_{a_m dx}, \quad 4)$$

$$Rot(Z^k_{adx}) = Z^k_{dxa} = Z^k_{xad} = Z^k_{ndx} \quad 5)$$

3) агрегація:

$$DU(Z^k_{adx}) = Z^K_{ADX}, \quad 6)$$

де $A=\{a_1, a_2, a_3, ..., a_n\}$, $D=\{d_1, d_2, d_3, ..., d_m\}$, $X=\{x_1, x_2, x_3, ..., x_m\}$, $K=\sum_{a=1}^{n}\sum_{d=1}^{m}\sum_{x=1}^{t} k_{adx}$

4) деталізація:

$$DD(Z^K_{ADX}) = Z^{K_1}_{ADx_1}, \quad 7)$$

де $A=\{a_1, a_2, a_3, ..., a_n\}$, $D=\{d_1, d_2, d_3, ..., d_m\}$, x_1=const, $K_1=\sum_{a=1}^{n}\sum_{d=1}^{m} k_{adx_1}$

Ця стаття пропонує непогану математичну модель сховища даних та алгоритм його проектування, однак не розглядає проектування сховища даних за певним критерієм, що не дозволяє говорити про оптимальність сховища даних.

Стаття [1] описує генетично-жадібний алгоритм для вибору куба даних OLAP. Задачею є вибрати куби даних з N вимірами з 2N можливих. Хромосома складена з 1 для вибраних кубів, 0 – для не вибраних. За функцію придатності взята функція вартості, яка дорівнює

$$\sum_{i=1}^{n} f_{c_i} E(c_i.M) + g_u \sum_{c \in M} U(c, M) \quad (8)$$

де f_{c_i} – частота звернення до куба і, $E(c_i.M)$ – вартість оцінки куба і при поточній матеріалізації М, g_u – частота вставки в базове відношення, $U(c,M)$ – вартість обслуговування куба і при поточній матеріалізації М. Селекція вибирається шляхом генерації випадкового числа від 0 до 1 і вибору більш придатної хромосоми, якщо це число перевищує 0,75. Схрещування відбувається з імовірністю (точкою схрещування) 0,7. Мутація відбувається з імовірністю pm. Крім генетичного алгоритму, використовується жадний алгоритм для тих рішень, які не є допустимими. Обробка недопустимих рішень відбувається так: якщо пара батьконащадок мають тих самих предків і вони матеріалізовані, тоді треба роз матеріалізувати вузол-нащадок, перерахувати загальну вартість і замінити старе рішення новим.

Дане рішення може бути використано як елемент МАР для вибору кубів у багатовимірній базі даних, однак воно не вирішує питання побудови цього сховища.

В статті [2] виріщується питання вибору представлень для матеріалізації шляхом «об'єднання» планів виконання запиту. Хромосома складена так: 1 – якщо представлення матеріалізується, 0 – якщо ні. Функція придатності дорівнює $C_{\max} - c(x)$, де $c(x)$ – функція вартості, а $C_{\max}$ – максимальне значення $c(x)$ в популяції або в останніх k поколіннях. Використовується одно точкове схрещування з випадково вибраною точкою схрещування від 1 до n, де n – довжина хромосоми. Мутація відбувається за допомогою інвертування біта від 1 до n, що випадково вибирається в хромосомі.

Дане рішення може бути використано для вибору матеріалізованих представлень.

У статті [3] розглядаються проблема вибору схеми розбиття сховища даних, вико ристовується горизонтальне розбиття. Кожен атрибут фрагментації бути предсттавлений масивом з n елементами, де n відповідає числу його піддоменів. Значення цих елементів знходться між 1 і n. Якщо два елементи мають ті ж значення, то вони будуть об'єднані для формування одного. Це кодування може бути використане для представлення фрагментів таблиць вимірів і таблиці фактів. Для селекції використовується метод колеса рулетки (він виділяє сектор колеса, що відповідає і-й хромосомі і створює нащадка, якщо згенероване число знаходиться в околі 0 зупиняється усередині призначеного сектора рядка). Використовується двоточковий механізм схрещування. Значення придатності визначається за допомогою призначення балів по двох показниках : поріг і виконання запитів. Поріг визначає, чи перевищує кількість отриманих фрагментів дорівнює ± 5 відсотків порогу, то будуть призначені 55 балів, інакше менше балів. Виконання запитів визнчає, чи перевищує вартість запиту встановлене значення, що обчислюється за допомогою функції вартості, якщо це так, то призначається менше 3 балів за запит. Функцією вартості є

$$Cost(Q) = \sum_{j=1}^{N} valid(Q, S_i) \prod_{i=1}^{M_j} \left(\frac{Sel_F^{p_i} \times \|F\| \times L}{PS} \right), \quad (9)$$

де M_j, F, L і PS позначають відповідно кількість предикатів вибірки, що визначають фрагмент факту підсхеми типу «зірка» $Sel_F^{p_i}$, кількість кортежів, присутніх в таблиці фактів F, розмір у байтах кортежу таблиці F та розмір сторінки файлової системи (у байтах).

Мутація відбувається з нормою 6-30%.

Дане рішення може бути використано як елемент МАР для вибору схеми розбивки бази даних (як реляційної, так і багатовимірної) , але тим самим вирішується лише питання оптимізації сховища даних.

У статті [4] розглядається застосування генетичного алгоритму до вибору матеріалізованих представлень. Хромосома складається з двох частин – вибраних планів виконання для кожного з запитів, та вибраних представлень для матеріалізації. Селекція відбувається за допомогою вибору числа від 1 до $\sqrt{n}$, потім це число підноситься до квадрату і вибирається індивід з отриманим порядковим номером, з упорядкованих індивідів по вартості у порядку її збільшення. Використовується одно точкове схрещування шляхом схрещування обох частин хромосоми: рядка планів виконання запитів та рядка представлень для матеріалізації, кожна схрещується окремо. Мутація відбувається шляхом інвертування бітів в рядку представлень, а також змінює числа в рядку планів виконання, підтримуючи допустимість рішення.

Дане рішення також може бути використано для вибору матеріалізованих представлень .

У статті [5] також розглядається проблема вибору матеріалізованих представлень. В якості хромосоми взято номери планів виконання запитів, об'єднані в бінарний рядок. Функція придатності дорівнює $C_{\max} - c(x)$, де $c(x)$ – функція вартості, а $C_{\max}$ – максимальне значення

c(x) в популяції або в останніх k поколіннях. Використовується одно точкове схрещування. Мутація відбувається шляхом випадкового вибору плану виконання запиту, з усіх доступних для вибраного для мутації запиту. Використовується турнірна селекція, де порівнюються індивіди, і кращий з них проходить до наступної популяції. Розмір турніру від 4 до 7 індивідів.

Дане рішення є більш раннім варіантом рішення [2] і також може бути використано для вибору матеріалізованих представлень.

Стаття [6] описує вибір індексів, матеріалізованих представлень для покращення реляційних баз даних. Ця стаття описує вибір індексів, матеріалізованих представлень для покращення реляційних баз даних. Використовується генетично-жадний алгоритм. Хромосома формується слідуючим чином. Об'єкт (просторові відношення, агрегатні представлення та індекси) представляється масивом бітів, тобто по частинах, які будуть об'єднані в хромосому. Просторові представлення представляються (як особливий тип представлень для матеріалізації) з їх різними варіантами . Кількітсь бітів, необхідних для представлення кожного просторового відношення з всіма його варіантами дорівнює k +1 , де k – кількість елементів додаткового набору атрибутів. Тому, перший біт використовується для представлення просторового відношення а інші n біт для представленнядодаткових атрибутів. Всі просторові відношення мають бути матеріалізовані,тому кожне з них представляється 1. Атрибут в додатковому наборі , при додаванні до його просторового відношення представляється 1,інакше 0. Агреговані представлення представляються схожим чином, тобто для кожного представлення, один біт вико ристовується для його представлення (1 – якщо вибране, 0 – не вибране для матеріалізації) і k бітів використовуються для представлення його додаткових атрибутів , тобто варіантів пред-ставлень. Атрибути додаткового набору агрегованих представляються у схожий спосіб з атрибутами додаткових наборів просторових відношень. Для кожного агрегованого представлення атрибути міри представляєься n бітами. Атрибут міри, якщо він додається до відповідного представлення, представляється 1, інакше 0.Для кожного представлення має бути доданий по якнайменше один атрибут міри. Після представлення кожного подання відбувається представлення їх можливих індексів. Кожний індекс представляється одними бітом , тобто 1 – якщо індекс вибраний, 0 – не вибраний. Кількість і порядок індексів визначаються попередньо.

Цільова функція задається наступним чином. Нехай SCM є станом схеми кубу даних SC з набором AVM AV ⊆ представлень-кандидатів на матеріалізацію, де кожний з них представляється його варіантом і набором SIM SI ⊆ індексів-кандидатів. Нехай також всі просторові відношення представляються їх відповідними варіантами. Тоді проблема оптимізації, обмежена витратами на обслуговування є наступною: вибрати стан SCM схеми кубу даних SC, яка мінімізує

$$\tau(SC, SCM, SQ) = \sum_{Q \in SQ} FQ \cdot P(Q, SCM), \qquad (10)$$

при обмеженні U (SC , SCM) ≤ S , де SQ – набір попередньо визначених запитів, Q F – частота запитів, P(Q,SCM) позначає мінімальну вартість обробки запиту Q в стані SCM схеми SC . Нехай U(SC, SCM) – загальна вартість обслуговування , визначена як:

$$U(SC, SCM) = \sum_{R \in R_{SC}} GR \cdot m(R, SCM) + \\ + \sum_{V \in AV_M} GV \cdot m(V, SCM) + \sum_{I \in I_M} GI \cdot m(I, SCM) \qquad (11)$$

де GR, GV and GI – частота оновлення відповідно відношень, представлень та індексів. Нехай m(R, SCM), m(V, SCM) та m(I, SCM) – мінімальна вартість обслуговування відношень, представлень і індексів відповідно в присутності стану SCM. P(Q, SCM) – цільова функція проблеми.

Розглянуті теоретичні рішення дозволяють оптимізувати сховище даних за допомогою індексів, матеріалізованих представлень та фрагментації. Оптимізацією є знаходження оптимальних рішень по заданих цільових функціях, керуючись заданими функціями придатності. Однак ці методи не вико ристовуються в комплексі, а методи не враховують поєднання реляційної та багатовимірної бази даних, а також структури джерел даних.

Тому необхідно запропонувати таку математичну модель, яка б враховувала і розміщення даних у сховищі, і оптимізаційні фактори (матеріалізовані подання, індекси, схеми роз-

биття), частоту виконання запитів до сховища та структури джерел даних.

Для формулювання математичної моделі задачі проектування гібридних сховищ даних з врахуванням структури джерел сховищ даних потрібно зробити формальний опис джерел даних, сховища даних, яке отримуємо в результаті проектування, а також оптимізаційних параметрів та рівнянь оптимізації.

Джерела даних. Джерела даних можуть бути структурованими файлами (розділеними чи XML) та базами даних двох типів: OLAP і OLTP.

У випадку розділених структурованих файлів метадані не задаються. У таких файлах може бути задана лише назви полів даних, що входять у файл. Типи даних не задаються. Формалізуються вони так: $S=\{s_i\}$ – множина атрибутів даних у розподіленому файлі.

У випадку структурованих файлів XML можуть бути задані метадані. У якості метаданих можу бути задані і метадані: назва поля, тип та інші необхідні характеристики. Формалізуються вони аналогічно до розподілених файлів : $X=\{x_i\}$ – множина атрибутів даних у файлі XML.

У випадку реляційних баз даних структура даних задається відношеннями. Відношення позначаються як $R=(r_1,r_2,...,r_1)$, де R – відношення, а r_i – атрибути відношення. Якщо два відношення $R=(r_1,r_2,...,r_1,r_{l+1},...,r_{l+p})$ та $R=(r_l,r_{l+1},...,r_{l+p},...,r_{l+k})$ мають спільні атрибути $r_1,r_{l+1},...,r_{l+p}$,то в одному з відношень цей набір атрибутів приймає унікальні значення, а в іншому – є зовнішнім ключем.

У випадку багатовимірних баз даних структура даних задається набором вимірів $D=\{d_1,d_2,\ldots,d_q\}$ та мір (можуть формулюватися також як вимір) $M=\{m_1,m_2,\ldots,m_u\}$.

Гібридне сховище даних. Відповідно до [11] гібридне сховище даних має наступні характерні особливості. Основними рисами гібридного сховища даних є наявність своєї окремої системи керування та автоматизоване проектування сховища під популяції запитів.

Узагальнене гібридне сховище даних передбачає наступне:

1) Гібридне сховище даних має свою систему керування гібридним сховищем даних (СКГСД), до за допомогою якої здійснюється робота з сховищем (виконання запитів до сховища).

2) В гібридному сховищі присутня реляційна БД.

3) В гібридному сховищі присутня багатовимірна БД, яка може містити як атомарні, так і узагальнені дані

4) Поєднуються підходи проектування «зверху вниз» і «знизу вгору». Реляційна і багатовимірна бази даних проектуються в комплексі, для забезпечення оптимального функціонування сховища даних у цілому.

У математичній моделі гібридне сховище даних представляється наступним чином. Воно містить :

1) Атрибути даних: $B=\{b_i\}=$ $=S\cup X\cup R\cup D\cup M$, тобто атрибути джерел повинні бути присутні в сховищі даних, що проектується.

2) Таблиці даних, що складаються з атрибутів даних: $T=\{t_i \mid t_i=\{b_j\}\}$)

3) Області сховища, що містять таблиці даних:

$$A=\{a_i\} \mid a_i=\{t_j\},\ j\in\overline{1,w},\exists b_k\in t_{j_1}, b_k\in t_{j_2})\}$$

Оптимізаційні параметри та рівняння оптимізації

З огляду на необхідність оптимізації гібридного сховища даних я пропоную використати два механізми покращення роботи сховища – застосування індексів і матеріалізованих представлень.

Індексом будемо вважати об’єкт бази даних, який дозволяє отримувати дані з таблиць швидше за допомогою збереження словників даних, які будуються за одним чи декількома стовпцями таблиці, на якій визначається індекс. У реляційних БД поля індексу доцільно визначати за допомогою генетичного алгориму : у хромосому включаються гени, які відображають включення поля в індекс таблиці. В багатовимірних базах даних побудова індексів найчастіше інкапсульована, а тому створення індексів є або неможливим, або зайвим, так як при наявності двох наборів індексів продуктивність бази даних може бути знижена. Однак принцип індексування в алгоритмі може поширюватися і на багатовимірну базу даних. Індексуються ті поля, при індексуванні яких ЦФ на цій таблиці є мінімальною.

Матеріалізованим представленням будемо вважати збережену окрему від баз даних таб-

лицю або таблиці даних, який включає в себе деякі поля цих таблиць, вибір яких здійснюється генетичним алгоритмом. У хромосому включаються гени, які від ображають включення поля в матеріалізоване представлення. Даний підхід повністю застосовний для реляційних баз даних, однак не застосовний для багатовимірних, оскільки розривання полів таблиці, що відповідає зрізу куба, може привести до порушення цілісності даних. Матеріалізуються тільки ті поля, при включенні яких у матеріалізоване представлення ЦФ на цій таблиці є мінімальною.

Для формалізації оптимізаційних параметрів та рівнянь оптимізації звернемося до викладених у роботі [10] результатів. У цій роботі Яцишином А.Ю. було запропоновано алгоритм проектування гібридних сховищ даних. Опишемо математичну модель, запропоновану в цій статті.

Відповідно до постановки задачі критерієм оптимальності сховища є кількість доступів до даних, тобто операцій читання даних, які необхідно провести для виконання запиту до сховища даних. Оскільки спроектувати базу даних під всі запити однаково ефективно неможливо, доцільно проектувати її під конкретні популяції запитів. Тому цільова функція задачі буде розраховуватися під перший запит популяції даних.

Цільова функція задачі має вигляд:

$$z = \sum_{i=1}^{n} A_i + \sum_{j=1}^{n-1} T_j , \qquad (11)$$

де n – кількість таблиць запиту, A_i – кількість доступів до *i*-ї таблиці реляційної бази даних або зрізу багатовимірної, T_j – час виконання операції з'єднання над таблицями $i_1=j$ та $i_2=j+1$

Але в практичній діяльності, враховуючи особливості використання різних баз даних, і те, що той самий запит може виконуватися порізному в залежності від виконаної оптимізації баз даних, а також неможливість отримання інформації про кількість доступів з ядра бази даних, вважаю, що дану ЦФ доцільно замінити на

$$z = \sum_{i=1}^{n} t_i + \sum_{j=1}^{n-1} T_j , \qquad (12)$$

де n – кількість таблиць запиту, t_i – час доступу до *i*-ї таблиці реляційної бази даних або зрізу багатовимірної, T_j – час виконання операції з'єднання над таблицями $i_1=j$ та $i_2=j+1$

Крім того, треба зауважити що t_i та T_j є функціями від розміщення таблиць даних:

$$t_i = f(L_1, L_2, ..., L_n) , \qquad (13)$$

де L_i – ознака розміщення таблиці даних (0 – в реляційній БД , 1 – в багатовимірній БД).

Тому ЦФ має такий остаточний вигляд:

$$z = \sum_{i=1}^{n} t_i(L_1, L_2, ..., L_n) + \sum_{j=1}^{n-1} T_j(L_1, L_2, ..., L_n) , \qquad (14)$$

де n – кількість таблиць запиту, L_i – ознака розміщення таблиці даних (0 – в реляційній БД , 1 – в багатовимірній БД).

Врахуючи змінні, запишемо наступний вираз:

$$z = \sum_{i=1}^{n} t_i(L_a, \{I_c\}, \{M_c\}) + \sum_{j=1}^{n-1} T_j , \qquad (15)$$

де n – кількість таблиць запиту,a – область, що містить таблицю i, $\{I_c\}$ та $\{M_c\}$ – ознаки індексування та матеріалізації полів c таблиці i. L_a – ознака розміщення областей сховища. $L_a = 1$, якщо всі таблиці області даних a розміщені в багатовимірній БД, та 0, якщо всі таблиці розміщені в реляційній БД.

Індексування полів таблиць представляється змінними I_c. $I_c = 1$, якщо індекс містить поле c, в протилежному випадку – 0.

Матеріалізованість полів таблиць представляється змінними $M_c . M_c = 1$, якщо поле матеріалізується, в протилежному випадку – 0.

На ОДР у роботі [10] накладаються такі обмеження:

1) Забороняється розміщення таблиць, що містять зовнішні ключі та нечислові поля, в багатомірних базах даних, оскільки такі таблиці фактів не підтримуються.

2) Забороняється створення індексів в багатовимірній базі даних, що випливає з природи багатовимірних баз даних, оскільки ця операція найчастіше інкапсульована в двигуні бази даних і виконується автоматично.

3) Забороняється включення зовнішніх ключів в поля, що входять до матеріалізованих представлень таблиць, які містяться в багатовимірній базі даних, оскільки це робить неможливим коректне виконання запитів до багатовимірної з бази даних.

4) Однак таке рішення не дозволяє врахувати такі параметри, як частоти використання відношень у сховищі, а також джерел даних.

Тому необхідно використовувати наступну модель:

$$z = \sum_{i=1}^{n} t_i(L_a,\{I_c\},\{M_c\}) + \sum_{j=1}^{n-1} T_j + (\frac{1}{F_a^{'}} - \frac{1}{\hat{F}_a})L_a \quad (16)$$

$$z = \sum_{i=1}^{n} t_i(L_a,\{I_c\},\{M_c\}) + \sum_{j=1}^{n-1} T_j + (T_a^{'} - \hat{T}_a)L_a, \quad (17)$$

де n – кількість таблиць запиту,a – область, що містить таблицю i, $\{I_c\}$ та $\{M_c\}$ – ознаки індексування та матеріалізації полів cтаблиці i, $F_a = \sum_{i \in a} f_i$ – частота доступу до області а. $F_a^{'}$– встановлене порогове значення частоти доступу до даних області a, $\hat{T}_a$ – середній час виконання запитів, $T_a^{'}$– порогове значення часу виконання запиту у до даних області a .

Показник частоти задається тому, що при високій частоті доступу до даних доцільно розміщувати у реляційній базі даних. Для цього додатково задається порогове значення частоти $F_a^{'}$, яке впливає наступним чином :

- Якщо сумарна частота доступів до області а більше порогового значення, цільова функція збільшується на різницю $(\frac{1}{F_a^{'}} - \frac{1}{F_a})L_a$, у випадку L_a=1 дане значення стає «гіршим» з точки зору оптимізації, так як існують рішення, на яких цільова функція приймає менше значення.
- Якщо сумарна частота доступів до області а менше порогового значення, цільова функція зменшується на різницю $(\frac{1}{F_a^{'}} - \frac{1}{F_a})L_a$, у випадку L_a=1 дане значення стає «краще» з точки зору оптимізації, ніж без врахування фактору частоти.

Запишемо остаточне формулювання задачі проектування гібридних сховищ даних.

Задані множини атрибутів розділених файлів S, файлів XML X, відношення у реляційній базі даних R, виміри багатовимірної бази даних D та міри M .. Спроектувати гібридне сховище даних, визначивши області сховища даних A, таблиці T та атрибути B та задавши порогові значення частот доступу до даних областей $F_a^{'}$.Знайти такі значення ознак розміщення областей L_a, індексування I_c та матеріалізації M_c, на яких значення цільової функції (17) буде мінімальним серед всіх можливих наборів значень цих змінних.

Висновки

У даній статті було проаналізовано існуючі роботи, що стосуються математичних моделей проектування сховищ даних . Розглянуті теоретичні рішення пропонують моделі проектування сховища даних, оптимізації за допомогою індексів,матеріалізованих представлень та фрагментації. Однак ці методи не використовуються в комплексі, а методи не враховують поєднання реляційної та багатовимірної бази даних.

Тому запропоновано математичну модель, що враховує структуру джерел даних та гібридного сховища даних, оптимізаційні параметри та рівняння.

В подальших дослідженнях слід вивчити особливості функції часу виконання запиту в залежності від її параметрів та запропонувати методи розв’язання задачі проектування гібридних сховищ даних з врахуванням структур сховищ даних.

Перелік посилань

1. Wen-Yang Lin. A Genetic Selection Algorithm for OLAP Data Cube [Текст] / Wen-Yang Lin,I-Chung Kuo – Knowledge and information systems 2004, vol. 6
2. Chuan Zhang.An Evolutionary Approach to Materialized Views Selection in a Data Warehouse Environment. Systems [Текст]/ . Chuan Zhang, Xin Yao, Jian Yang – Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on Volume 31, Issue 3, Aug 2001
3. Ladjel Bellatreche. An Evolutionary Approach to Schema Partitioning Selection in a Data Warehouse [Текст] / Ladjel Bellatreche,Kamel Boukhalfa – Lecture notes in computer science, Congrès DaWaK 2005 : data warehousing and knowledge discovery: International conference on data warehousing and knowledge discovery No7, Copenhagen , DANEMARK 2005 , vol. 3589
4. J.-T. Horng. Applying evolutionary algorithms to materialized view selection in a data warehouse [Текст]/ J.-T. Horng, Y.-J. Chang, B.-J. Liu – Soft Computing – A Fusion of Foundations, Methodologies and Applications, Volume 7, Number 8 / August, 2003

5. Chuan Zhang. Evolving Materialized views in a Data Warehouse [Текст] / Chuan Zhang, Xin Yao, Jian Yang – Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on Volume 2
6. Goran Velinov. Framework for Generalization and Improvement of Relational Data [Текст] / Goran Velinov , Danilo Gligoroski, Margita Kon Popovska – IJCSNS International Journal of Computer Science and Network Security, VOL.8 No.3, March 2008
7. Michael Lawrence. Multiobjective genetic algorithms for materialized view selection in OLAP data [Текст] / Michael Lawrence – Proceedings of the 8th annual conference on Genetic and evolutionary computation,2006
8. Бакулева Марина Алексеевна. Модели и алгоритмы автоматизации проектирования структур хранилищ данных для аналитической обработки числовых показателей [Текст] : диссертация кандидата технических наук : 05.13.12 Рязань, 2007 147 с., Библиогр.: с. 124-131 РГБ ОД, 61:07-5/4771
9. Шаховська Н.Б. Сховища та простори даних: Монографія [Текст]. / Шаховська Н.Б, Пасічник В.В. – Львів: Видавництво Національного університету «Львівська політехніка» , 2009. – 244 с.
10. Яцишин А.Ю. Застосування генетичного алгоритму для проектування гібридних сховищ даних [Текст]. Вісник Національного університету „Львівська політехніка”, секція "Інформаційні системи та мережі",/ Яцишин А.Ю. - м.Львів - 2011
11. Яцишин А.Ю. Підходи та алгоритми проектування гібридних сховищ даних [Текст]. Вісник Національного університету „Львівська політехніка”, секція "Інформаційні системи та мережі" / Яцишин А.Ю – м. Львів - 2010