УДК 004.652.43
© *С. Кулик*[1], *В. Овсяк*[1], 2010

СИНТАКСИС І СЕМАНТИКА ОПИСУ БАЗОВИХ ПОНЯТЬ РЕЛЯЦІЙНОЇ МОДЕЛІ ТА ОПЕРАЦІЙ

У роботі зроблено опис уніфікованої системи позначень понять реляційної моделі, наведена семантика побудови виразів, адаптовано синтаксис до комп'ютерних засобів створення неформатованого тексту у рядок, закладений опис базових операцій з множинами та елементами множин.

Description of the compatible system of denotations of concepts of relation model are presents, semantic of construction of expressions, syntax adapted to computer facilities of creation of not formatted text in a line, description of base operations with sets and elements of sets are considered here.

1. ФОРМУЛЮВАННЯ ЗАДАЧІ

Модель реляційної бази даних утворена структурною, маніпуляційною та цілісною частинами. Опис моделі реляційної бази даних засобами алгебри алгоритмів підвищить рівень автоматизації програмування прикладних реляційних баз даних. У зв'язку з обмеженням обсягу публікації у даній роботі розглядається формалізація засобами алгебри алгоритмів тільки маніпуляційної та цілісної частин реляційної моделі.

2. БАЗОВІ ПОНЯТТЯ

Опис маніпуляційної частини це перш за все побудова складних математичних виразів на основі понять відношення, кортежа, атрибута та оперування множинами і їх елементами [1].

Для ефективного і зрозумілого абстрактного опису моделей створимо уніфіковану систему позначень для внутрішньої ідентифікації об'єктів баз даних (БД).

Для створюваної системи позначень є суттєвими такі властивості відношень[2]:

Кожен кортеж містить точно одне значення для кожного атрибуту;

У відношенні відсутні дублікати кортежів.[3, с. 51]

Для унітермів алгебри алгоритмів [4] введемо такі позначення понять реляційної моделі:

---

[1] Українська академія друкарства

Відношення - велика літера R. Ідентифікація відношення за унікальною назвою, заданою на абстрактному рівні нижнім індексом. Наприклад: R_1, R_3, R_s.

Атрибут - велика літера A. Ідентифікація за ключем, заданому на абстрактному рівні нижнім індексом (A_1, A_2, …, A_n).

Кортеж маленька літера t. Ідентифікація за ключем, заданому на абстрактному рівні нижнім індексом (t_1, t_6, …, t_n).

Первинний ключ - велика літера K з встановленням належності до відношення за ключем відношення у вигляді верхнього індексу (наприклад, K^2) та ідентифікації за власним ключем, заданому на абстрактному рівні нижнім індексом, наприклад $K^2{}_1$.

Вторинний (зовнішній) ключ - велика літера Z з встановленням належності до певного відношення за ключем відношення у вигляді верхнього індексу (наприклад, Z^3) та ідентифікації у межах відношення за власним ключем, заданим на абстрактному рівні нижнім індексом, наприклад $Z^3{}_2$;

Значення компонента кортежу - маленька літера v, у секвентній трійці $\overline{\overline{\overline{A_i} ; D_i} ; v_i}$. Ідентифікується тільки засобами оператора доступу $R_1.A_5.v$ (описано нижче).

Нагадаємо, що поняття відношення, домена, атрибута, кортежа означені на основі математичного поняття множини. По-перше множини не є впорядковані але в них можна штучно задавати відношення порядку. По-друге усі елементи множини є різними [2].

Невпорядкованість математичних множин обумовлює відповідну невпорядкованість атрибутів (доменів), відношень та кортежів [1]. Тому якщо більш строго ставитись до їх невпорядкованості то опис множин утворюючих основу реляційної моделі треба задавати як невпорядковані секвентні області. В алгебрі алгоритмів унітерми невпорядкованих секвенцій розділяється не крапкою з комою, а комою.[4, 5, 6]. Наприклад:

$$R.h=\overline{\overline{\overline{A_1,A_2} , A_3},\ldots,A_n}$$

$$R.b=\overline{\overline{\overline{t_1,t_2} , t_3},\ldots,t_m}$$

До системи позначень введемо верхні індекси. Допустимим є використання верхніх індексів для ідентифікації належності кортежів чи атрибутів до відношення. Наприклад:

$$t_7^1 \in R_1, \qquad A_9^2 \in R_2,$$

де верхні індекси означають відповідно належність кортежу до відношення з назвою 1, та належність атрибуту до відношення 2.

Для структур даних реляційної моделі атрибут, кортеж, відношення не означена позиційна адресація та немає поняття «наступність». [1] У роботі будемо використовувати цілі числа як абстрактне зображення ключів, які з цієї причини будуть справляти враження впорядкованості. Треба розуміти що замість чисел ми у абстрактному описі відношень могли застосовувати назви, та розташувати елементи у будь-якому порядку слідування ключів. При будь-якому порядку розташування ключів це будуть ті ж самі відношення. Але такий опис є маловиразним та громіздким.

При проектуванні баз даних поняття відношення прийнято описувати як секвенцію атрибутів побудовану на основі відповідних доменів. Саме атрибут має властивість унікальності своєї назви, що робить його більш зручним засобом практичного опису змісту відношень. Домени ж означені на певних системних типах та означених властивостях [1, с. 45].

Таблиці у програмуванні реалізуються індексними масивами виду A[n][m], де n – кількість стовпців, m – кількість рядків.

Але відношення не є таблицями. За допомогою таблиць відношення просто представлені користувачеві, як у формі найбільш зручній для сприйняття та розуміння.

Для реалізації невпорядкованих множин у програмуванні зокрема використовуються:

Асоціативні масиви (map)

Хеш-таблиці (hashtable)

Колекції (collection): стек, вектор, матриця, множина, черга, список та ін.

Бінарні дерева (b-trees)

Для усіх приведених структур даних є характерним:

– відсутність будь-якої впорядкованості, наприклад, зліва направо чи зверху вниз;

– наявність у кожного елемента пари ключ - значення (key - value);

– ідентифікація та звертання до елементів за ключем;

– відсутність двох однакових елементів за рахунок унікальності ключів;

– при вилученні елемента із множини не утворюється порожній елемент зі старим індексом та не відбувається вилучення із зсувом ключів усіх «наступних» елементів;

– наявність ітератора – вказівника на елементи даних, який підтримує базовий набір операцій: розіменування [←], інкремент/декремент[++]/[--], складання/віднімання [+]/[-] та порівняння[==].

Вони мають різну ефективність пошуку, впорядкування, маніпуляцій але будь-яку із цих структур можна практично застосовувати для реалізації відношень БД. Для реалізації ефективних алгоритмів пошуку можливим є подальший розвиток бінарного дерева. Бінарне дерево пошуку та бінарна купа, які мають відносну впорядкованість елементів за ключем.

Детальнішій огляд структур програмування пристосованих для реалізації реляційної моделі є окремим предметом дослідження та у зв'язку з обмеженням обсягу статті тут не розглядаються.

Для кожного відношення обирається за ознакою унікальності або штучно створюється атрибут A, який є первинним ключем K та кожне значення v якого ідентифікує певний кортеж t відношення R. Докладніше до поняття ключа відношення, його властивостей, обмежень і системи правил ми звернемось окремо при розгляді цілісної частини реляційної моделі.

3. ГРАМАТИКА ОПИСУ УНІТЕРМІВ

На основі алгебри алгоритмів створені спеціальні системи набору і редагування формул алгоритмів, наприклад, системи МОДАЛ і АБСТРАКТАЛ [5,6] не забезпечують набору верхніх та нижніх індексів. Тому їх буде реалізовано як властивості шрифту (нарівні з такими як курсив, напівжирний, підкреслений та ін.). Випадкове пристосування нового стилю до текстового блоку здатне водночас перетворити усі індекси опису на звичайний текст, що не є допустимим.

Вище наведені обмеження відомих комп'ютерних систем набору і редагування формул алгоритмів вимушують ввести синтаксис і семантику опису унітермів з індексами.

У зв'язку з тим що у більшості комп'ютерних систем розробки (редактори коду, інтегровані середовища розробки), власних засобів для введення верхнього та нижнього індексів не реалізовано, введемо нові позначення індексів унітермів. Позначення індексів унітермів подаємо у вигляді рядка неформатованого тексту. Наприклад, унітерм A^1_2 запишеться так:

A↑[1]↓[2] ,

де ↑[] – ознака верхнього індексу, а ↓[] – ознака нижнього індексу.

Дотримування порядку індексів «спочатку верхні потім нижні» є обов'язковим.

Квадратні дужки дають можливість відрізняти круглі дужки якщо індекс є складним виразом з певним порядком дій.

Наприклад:

$A^{(n-m)/2}{}_2$=A↑[(n-m)/2]↓[2]

Якщо індекс є скаляром то квадратні дужки можна опустити:

$A^1{}_2$=A↑1↓2

Якщо змінна має зарезервовану назву як A, t, R, та у виразі використовується тільки нижній індекс (який вказує на ключ елемента), то знак ↓ можна опустити, система замовченням вважає числову частину назви нижнім індексом:

A_2=A↓2=A2

Така система запису індексів унітермів наближає їх опис до внутрішнього зображення відношень та їх властивостей у комп'ютерних системах.

3.1. ПРАВИЛА ЗАСТОСУВАННЯ ВЕРХНІХ, НИЖНІХ, ЛІВИХ ТА ПРАВИХ ІНДЕКСІВ ЗМІННОЇ

Наведений вище опис індексів дає можливість застосовувати лівий та правий індекси, це має виглядати так:

${}^{K}A^1{}_2$=A↑[K][1]↓[2]

Правий індекс застосовується для ідентифікації, ознаки належності та якщо потрібно – впорядкування.

Верхній індекс 1 означає що атрибут A належить до відношення R_1, тобто 1 – назва ключа відповідного відношення.

Атрибут A має власний ключ 2, який однозначно ідентифікує його у відношенні R_1.

Лівий індекс призначено для типізації елементів та вказування на певні, означені раніше властивості. Наприклад, A↑[K][1]↓[2] означає що атрибут A є первинним ключем у відношенні R_1 та від нього функціонально залежать інші атрибути відношення R_1.

У записі пари [x][y] під знаками верхнього ↑ або нижнього ↓ індексів є впорядкованість зліва направо: перший елемент є лівим індексом, другий правим.

3.2. ІДЕНТИФІКАЦІЯ ЕЛЕМЕНТІВ МНОЖИН ТА ОПЕРАТОРА ДОСТУПУ

Введемо оператор за допомогою якого можна безпосередньо звертатись до будь-яких елементів множини – оператор доступу [.].

Вершину ієрархії у реляційної моделі займає відношення, яке має підпорядковані йому елементи: атрибути, кортежі, ключі та інші змінні.

Через крапку можна звертатись до будь якого елемента секвенції та ідентифікувати підпорядковані елементи. Наприклад:

V^5_2=R5.V2 – зовнішній ключ V_2 відношення R_5,

K_1=R1.К – первинний ключ відношення R_1,

t^3_2=R3.t2=$R_3.t_2$ – другий кортеж відношення R_3,

$R_3.t_2.v$ – звертання до значення кортежу з ключем 2, який належить відношенню з індексом 3.

$R_2.m$ – звертання до змінної яка показує кількість кортежів у відношенні з індексом 2.

R4.A8.D.a0 – звертання до властивості тип, на якому створено домен атрибуту з індексом 8 відношення з індексом 4.

3.3. ТІЛО ТА ЗАГОЛОВОК ВІДНОШЕННЯ

Заголовок (head) відношення R.h є заголовком кортежу. Відношення R має такі самі атрибути (відповідно такі самі назви та типи атрибутів) та такий порядок (ступінь) як заголовок.

Тіло (body) відношення R.b є множина кортежів, які мають сам заголовок; потужність (кардинальність) відношення R.

Фактично заголовок містить означення атрибутів, а тіло містить множину кортежів, кожен елемент яких відповідає атрибутам заголовку.

Опис:

$$R1=\overline{R1.h\,;\,R1.b}=\overline{h1\,;\,b1}$$

де,

$R1.h=\overline{\overline{\overline{A1,A2}\,,A3},\ldots,An}$ – заголовок відношення R1.

$R1.b=\overline{\overline{\overline{t1,t2}\,,t3},\ldots,tm}$ – тіло відношення R1.

Властивості тіла та заголовку [1,2] відношення такі:

Кожна підмножина заголовку є заголовком.

Кожна підмножина тіла є тілом.

3.4. ОПЕРАТОРИ

У термінах нових позначень означимо базові оператори над складовими об’єктами реляційної моделі.

1.Оператори виду елемент–елемент

1.1. Оператори порівняння кортежів

Оператор присвоєння значення [=];

Умовні оператори «дорівнює» [==] та «не дорівнює» [!=];

Операція перевірки кортежу на рівність є основою на якій визначені майже усі операції реляційної алгебри, поняття ключів та функціональної залежності. Означмо її так:

Кортежі t1 та t2 є рівними, (тобто вираз t1 == t2 приймає істинне значення) тоді та тільки тоді, коли вони мають однакові атрибути A1, A2, …, An, та для усіх i (i=1, 2, …, n) значення v1 атрибуту A_i в кортежі t1 дорівнює значенню v2 атрибуту A_i у кортежі t2.

Кортежі є дублікатами один відносно іншого тоді та тільки тоді коли вони рівні у вказаному вище сенсі.

Примітка

До кортежів, відношень та атрибутів не можуть бути застосовані умовні оператори порівняння «більше» [>], «менше» [<], «більше або дорівнює» [>=], «менше або дорівнює» [<=] по вказаним вижче причинам відсутності в них впорядкованості.

1.2. Оператор порівняння атрибутів

Атрибути A↑[1]↓[1] та A↑[2]↓[2] різних відношень R1 та R2 є рівними (A↑[1]↓[1]==A↑[2]↓[2]), якщо вони мають однакову назву та означені на загальному домені D.

3.5. РЕЛЯЦІЙНІ ОПЕРАТОРИ ПОРІВНЯННЯ ВІДНОШЕНЬ

Дорівнює [==];
Не дорівнює [!=];
Підмножина [$\subseteq$];
Строга підмножина [$\subset$];
Надмножина [$\supseteq$];
Строга над множина [$\supset$].

3.6. ОПЕРАЦІЇ ОБ’ЄДНАННЯ ЕЛЕМЕНТІВ: [$\cup$]

3.6.1. Об’єднання кортежів t1 $\cup$ t2

Для кортежів t1 порядку k та t2 порядку n, застосування операції об’єднання t1 $\cup$ t2 (зчеплення) призводить до утворення нового кортежу t3 порядку (k+n) який містить усі компоненти кортежів t1 та t2.

3.6.2. Об’єднання атрибутів A1 $\cup$ A2

Для атрибутів A1 потужності l та A2 потужності m, застосування операції об’єднання A1 $\cup$ A2 (зчеплення) призводить до утворення нового атрибуту A3 порядку (l+m) який містить усі кортежі атрибутів A1 та A2.

3.7. ОПЕРАТОРИ МІЖ ЕЛЕМЕНТАМИ ТА МНОЖИНАМИ

1. Операції додавання елемента до множини

1.1. Додавання кортежу до відношення R+=t

Якщо до відношення R1, яке має потужність m, та тіло $R1.b = \overline{t[1], t[2], t[3], \cdots, t[m]}$ додати кортеж t↓[k] то тіло відношення R1 набуває новий кортеж, та як наслідок відношення збільшує свою потужність m на один:

$$R1.b = \overline{t[1], t[2], t[3], \ldots, t[m], t[k]}$$

1.2. Додавання атрибуту до відношення R+=A

Якщо до відношення R1, яке має порядок n, та заголовок $R1.h = \overline{A[1], A[2], A[3], \ldots, A[n]}$ додати атрибут A↓[k] то заголовок відношення R1 набуває новий атрибут, та як наслідок відношення збільшує свій порядок n на один:

$$R1.h = \overline{A[1], A[2], A[3], \ldots, A[n], A[k]}$$

2. Операції віднімання елемента від множини

2.1. Віднімання кортежу від відношення (вилучення кортежу) R–=t.

Якщо із відношення R1 яке має потужність m, та тіло $R1.b = \overline{t[1], t[2], t[3], \ldots, t[k], \ldots, t[m]}$ видалити кортеж t↓[k] то у тілі відношення R1 робиться пошук потрібного кортежу за ключем k, у випадку знаходження відбувається видалення знайденого кортежу та як наслідок відношення зменшує свою потужність m на один.

$$R1.b = \overline{t[1], t[2], t[3], \cdots, t[m]}$$

2.2. Віднімання атрибуту від відношення (вилучення атрибуту) R–=A

Якщо із відношення R1, яке має порядок n та заголовок $R1.h = \overline{A[1], A[2], A[3], \ldots, A[k], \ldots, A[n]}$ видалити атрибут A↓[k] то у заголовку відношення R1 робиться пошук потрібного атрибуту ключа k, у випадку знаходження відбувається видалення знайденого атрибуту та як наслідок відношення зменшує свій порядок n на один. $R1.h = \overline{A[1], A[2], A[3], \cdots, A[n]}$

2.3. Загально математичні оператори визначення належності елементу до множини є такими

– належить [$\in$];

– не належить [! $\in$].

Приведемо означення належності для кортежу:

Кортеж t належить до відношення R якщо серед множини кортежів тіла відношення $R.b = \overline{t[1], t[2], t[3], \ldots, t[k], \ldots, t[m]}$ є такий t[k] для якого виконується умова t==t[k].

4. ВИСНОВКИ

1. Для точного опису моделі створено універсально систему позначень та для ідентифікаторів прийняті відповідні абстрактні припущення.

2. Створено наочну систему опису індексів при змінних, адаптована до неформатованого тексту.

3. Створена система показала себе спроможною при опису базових операцій.

4. Описані базові операції дають можливість створити опис операцій реляційної алгебри.

5. Для створеної системи позначень характерна гнучкість у виборі найбільш зручних у даній ситуації засобів опису.

1. Дейт К. Дж. Введение в системы баз данных, 8-е издание.: Пер. с англ. — М.: Издательский дом "Вильямс", 2005. — 1328 с. 2. Коннолли Т., Бегг К. Базы данных. Проектирование, реализация и сопровождение. Теория и практика. (3-е издание). 3. Кодд Э. Ф. Расширение реляционной модели для лучшего отражения семантики. /Э. Ф. Кодд //Системы управления базами данных 5/1996.— Переклад російською з оригіналу E.F. Codd. Extending the Database Relational Model to Capture More Meaning. //ACM Transactions on Database Systems, Vol. 4, # 4, December 1979. 4. Owsiak W., Teoria algorytmov abstrakcyjnych i modeowanie matematyczne systemow informacyjnych./ Qwsiak A., Owsiak J.// −Opole: Studia i monografie. z. 176. ”Politechnika opolska”, 2005. − 275 s. 5. Овсяк В.К. Засоби еквівалентних перетворень алгоритмів інформаційно-технологічних систем. / Овсяк В.К.//Доповіді Національної академії наук України, 1996№9,. с. 83-89. 6. Ovsyak V.K. Computation models and algebra of algorithms /Ovsyak V.K.// Інформаційні системи та мережі. Вісник Національного університету «Львівська політехніка». 2008. − №621. − с. 3-18.