Список літератури

1. Буй Д. Б. Модели, методы и алгоритмы оптимизации запросов в базах данных (обзор) / Д. Б. Буй, В. Г. Скобелев // Радіоелектронні і комп'ютерні системи. – 2014. – № 2. – С. 43–58.
2. Мендекович Н. А. Обзор развития методов лексической оптимизации запросов / Н. А. Мендекович, С. Д. Кузнецов // Труды ИСП РАН. – 2012. – Т. 23. – С. 195–214.
3. Редько В. Н. К основаниям теории реляционных моделей баз данных / В. Н. Редько, Д. Б. Буй // Кибернетика и системный анализ. – 1996. – № 4. – С. 3–12.
4. Реляційні бази даних: табличні алгебри та SQL-подібні мови / В. Н. Редько, Ю. Й. Брона, Д. Б. Буй, С. А. Поляков. – К. : Академперіодика, 2001. – 198 с.
5. Codd E. F. A Relational Model of Data for Large Shared Data Banks / E. F. Codd // Comm. of ACM. – 1970. – Vol. 13, № 6. – P. 377–387.
6. Codd E. F. The Relational Model for Database Management: version 2 / E. F. Codd. – Addison-Wesley, 1990. – 538 p.
7. Knuth D. E. The Art of Computer Programming, Volume 4, Fascicle 0: Introduction to Combinatorial Algorithms and Boolean Functions / D. E. Knuth. – Addison-Wesley, 2008. – 240 p.

O. Senchenko

ABOUT DERIVATION OF RENAMING IN TABLE ALGEBRAS WHEN UNIVERSAL DOMAIN IS FINITE

In this paper the derivation of renaming in table algebras is considered. It is proved that when universal domain is finite, this operation is derived with respect to the operations of projection, selection and join, with the use of one constant table.

Keywords: database, table algebra, derivation.

Матеріал надійшов 30.04.2015

УДК 004.655

Буй Д. Б., Глушко І. М.

РОЗШИРЕННЯ СИГНАТУР РЕЛЯЦІЙНИХ (ТАБЛИЧНИХ) АЛГЕБР КОДДА: СУЧАСНИЙ СТАН

Статтю присвячено огляду літератури з розширень реляційної (табличної) алгебри. Розглянуто додаткові операції, що розширюють можливості реляційної алгебри: агрегування, групування, сортування, напівз'єднання, зовнішні з'єднання та ін. Проаналізовано використання null-значень при оперуванні невизначеною та неповною інформацією в базах даних. Оскільки існує ряд прикладних задач, особливістю яких є множинність і повторюваність даних, також приділено увагу питанню розширення можливостей баз даних за рахунок використання мультимножин.

Ключові слова: реляційні бази даних, реляційна (таблична) алгебра, розширення реляційної алгебри, мультимножина.

Вступ

Системи управління базами даних (СУБД) у наш час використовуються практично в усіх сферах людської діяльності, які пов'язані зі збереженням та обробкою інформації. Прогрес, досягнутий у галузі технологій баз даних (БД), значною мірою ґрунтується на реляційній моделі, запропонованій Е. Коддом (E. Codd) у 70-х роках XX ст. Традиційно вважається, що класична реляційна (таблична) алгебра лежить в основі більшості СУБД і мов запитів, які підтримують реляційну модель. Реляційна алгебра була розроблена Е. Коддом у вигляді сукупності операторів над таблицями [35] (1970 р.). Е. Кодд запропонував 8 операцій реляційної алгебри: традиційні операції над множинами (об'єднання, перетин, різниця) та спеціальні операції над таблицями (проекція, декартове з'єднання, з'єднання (theta-, equi-), ділення, селекція) [35]. Він також довів реляційну повноту цієї алгебри, вказавши алгоритм перетворення довільного виразу реляційного числення на кортежах (формальне визначення реляційного числення подане в цій же статті) на семантично еквівалентний алгебраїчний вираз [37]. Детальний опис реляційного числення зі змінними-кортежами Кодда можна знайти у [26].

Додаткові операції табличної алгебри

Набір операцій реляційної алгебри, запропонований Е. Коддом, з часом був розширений відповідно до потреб мов запитів. Крім вказаних вище операцій, до сигнатури реляційної алгебри нині також відносять операції перейменування та активного доповнення [25, с. 20–22, с. 35–37; 28, с. 33–34].

У ході розвитку комерційних реляційних СУБД виникла потреба у використанні агрегатних функцій, які дозволяють знаходити сумарні, середні, максимальні, мінімальні та інші значення елементів у стовпці (стовпцях) таблиці. Так, Е. Клуг (A. Klug) [49] (1982 р.) розширив реляційну алгебру та реляційне числення такими агрегатними функціями і показав еквівалентність отриманих при цьому двох формальних мов. У статті [49] подано точні означення агрегатних функцій, які не використовують поняття «дублікати». Реляційну алгебру поповнено новою операцією агрегатного утворення (aggregate formation). Синтаксис реляційного числення в порівнянні з [37] модифіковано, а саме: не дозволяються вирази, які будують відношення, що містять нескінченну кількість рядків. Показано, що набір запитів, які виразні в алгебрі, такий самий, як і набір запитів, виразних у численні.

П. Грей (P. Gray) (1989 р.) демонструє можливості реляційної алгебри АСТРІД (ASTRID), яка є розширенням реляційної алгебри Кодда [13]. АСТРІД містить деякі додаткові операції: розширення (extend) та групування (group_by), які призначені для отримання обчислюваних даних. Так, результатом застосування операції розширення є нова таблиця, що містить додаткові стовпці, значення в яких обчислюються, виходячи із значень у передуючих стовпцях. Ці додаткові значення функціонально залежать від ключа вхідної таблиці, який стає ключем і результуючої таблиці. Синтаксис оператора розширення в алгебрі АСТРІД у термінах БНФ (Бекуса – Наура форма) має такий вигляд:

<таблиця> := <таблиця> extend_by [<список-присвоювань>]

<список-присвоювань> ::= <ім'я-атрибута [1]> := <вираз> | <список-присвоювань>,

де ::=, |, [,] – метасимволи, що використовуються стандартно.

Операція group_by, за словами автора, поводить себе як проекція в поєднанні з розширенням. П. Грей описує виконання цієї операції таким неформальним чином: по-перше, вхідна таблиця розбивається на підтаблиці, до однієї підтаблиці потрапляють рядки, в яких збігаються значення всіх атрибутів зі списку проекції; по-друге, здійснюється проекція таблиці на ці атрибути, причому для кожного рядка в результуючій таблиці обчислюється похідне значення за допомогою деякої вбудованої функції, що застосовується до відповідної підтаблиці. Отримана таблиця, як і при застосуванні операції розширення, матиме додаткові стовпці, які містять ці обчислені значення. Синтаксис оператора групування в алгебрі АСТРІД такий:

<таблиця> := <таблиця> group_by [<список-імен-атр>
creating <список-присвоювань>]

<список-імен-атр> ::= <ім'я-атрибута> | <ім'я-атрибута>, <список-імен-атр>

<список-присвоювань> ::= <ім'я-атрибута> := <вираз> | <список-присвоювань>.

Мова АСТРІД містить вбудовані функції COUNT, SUM, MAX, MIN та дві логічні функції ANY та ALL, які використовуються для формування запитів з кванторами існування та загальності відповідно. Вказано, що квантор загальності можна виразити, використовуючи вбудовану функцію ALL разом із групуванням або ж за допомогою операції різниці множин. Також описана велика кількість різних способів оптимізації виразів реляційної алгебри. Автор зазначає, що мова АСТРІД дає змогу сформувати будь-який запит, виражений мовою SQL.

Методи формування запитів, які використовуються в реляційній алгебрі та реляційному

---

[1] Замість терміна «ім'я атрибута», на нашу думку, краще використовувати термін «атрибут», так само, як у [28].

численні, зовні виглядають по-різному. П. Грей у своїй книзі вивчає ці відмінності та розглядає декілька способів перетворення запитів з одного представлення в інше. Описано, як транслюються в реляційну алгебру вирази реляційного числення (як числення кортежів, так і числення на доменах) та як здійснюється зворотна трансляція. Показано, що еквівалентність реляційної алгебри та реляційного числення зберігається й тоді, коли до цих формалізмів додаються операції групування та розширення.

У книзі [46] (1989 р.) представлено 6 основних операцій реляційної алгебри, додаткові операції, які є похідними від основних, та операції, які дозволяють виразити спеціальні види запитів, їх автори називають неалгебраїчними. Основні операції реляційної алгебри поділено на дві групи. До першої групи віднесено операції об'єднання, різниці та декартового з'єднання, до другої – реляційно-орієнтовані операції (проекція, селекція та з'єднання (natural, theta-)). Похідні алгебраїчні операцій включають перетин, ділення, зовнішнє з'єднання (outer join) та напівз'єднання (semijoin).

Зупинимося більш детально на операції напівз'єднання $\ltimes$. Операція напівз'єднання двох таблиць R_1 та R_2 визначає таблицю R_3[2], яка має ті самі атрибути, що й таблиця R_1, та містить ті рядки таблиці R_1, які входять у (природне (natural) за термінологією [28]) з'єднання таблиць R_1 та R_2. Автори зазначають, що цю операцію також можна сформулювати за допомогою операцій проекції та з'єднання: $R_1 \ltimes R_2 = \pi_{X_1}(R_1 \bowtie R_2)$, де X_1 – множина атрибутів таблиці R_1.

До неалгебраїчних операцій належать доповнення, операція Split, операція агрегування та транзитивне замикання. Опишемо операцію Split. Автори вказують на те, що ця операція насправді не належить до операцій реляційної алгебри, оскільки, застосовуючи її до однієї таблиці, в результаті отримаємо дві таблиці. Одна з отриманих таблиць міститиме рядки, які задовольняють вказану умову, а друга – ті рядки, які цю умову не задовольняють.

Операції агрегування розглядаються як множинно-орієнтовані операції, що спочатку розбивають таблицю на підмножини (підтаблиці) відповідно до значень атрибута (або атрибутів), потім виконують функціональні обчислення для кожної підмножини (підтаблиці) і, нарешті, будують результуючу таблицю, формуючи один рядок для кожної підмножини (підтаблиці). Звернемо увагу, що ця схема обчислень використовується для запитів третього типу (з групуванням) мови SQL [28, с. 180–188]. Зауважимо також, що для семантики таких конструкцій треба вводити до розгляду сукупності з повтореннями, тобто мультимножини.

Автори розглядають функцію $AGG(R;X;B)$, де AGG один із функціональних символів (AVG, MAX, MIN, COUNT, SUM), R – таблиця, X – множина атрибутів таблиці R і B – атрибут таблиці R. Значенням цієї функції є таблиця $S(X, AGG\_B)$ схеми $X \bigcup \{B\}$, кожен рядок якої складається із значень, які належать множині значень атрибутів X у рядках таблиці-аргументу та результату застосування агрегатної функції до відповідних значень атрибута B, результат обчислень зберігається в атрибуті B. Множина X може бути порожня, тоді результуюча таблиця – це таблиця з єдиним атрибутом B та єдиним рядком, який містить результат застосування агрегатної функції до значень атрибута B по всіх рядках таблиці-аргументу.

Отже, тут також наявна стандартна схема групування: спочатку таблиця розбивається на підтаблиці згідно зі значеннями атрибутів групування X, потім для кожної підтаблиці формується мультимножина значень атрибута B, до якої застосовується агрегатна функція.

Ще однією неалгебраїчною операцією, яку розглядають Дж. Ґардарін (G. Gardarin) та П. Вальдуріз (P. Valduriez) в [46], є транзитивне замикання (transitive closure). У результаті застосування транзитивного замикання до таблиці R схеми $\{A_1, A_2\}$, де атрибути A_1, A_2 мають однаковий домен (значень), отримаємо таблицю тієї самої схеми, що містить усі рядки таблиці R та всі рядки, послідовно виведені застосуванням властивості транзитивності (transitivity) до таблиці: якщо існують рядки $\{<A_1, a>, <A_2, b>\}$ та $\{<A_1, b>, <A_2, c>\}$, то маємо й рядок $\{<A_1, a>, <A_2, c>\}$. Автори вказують на невиразність транзитивного замикання в реляційній алгебрі та зазначають, що ця операція широко використовується в дедуктивних БД.

Параметром операції виступає кортеж атрибутів $<A_1, A_2>$, бо порядок атрибутів суттєвий. Узагалі кажучи, формальне означення операції можна дати в такий спосіб. Нехай $U \stackrel{def}{=} \{<r(A_1), r(A_2)> \mid r \in R\}$ – вказане бінарне відношення, індуковане таблицею R, а U^* – його транзитивне замикання в традиційному розумінні [29], наприклад, $U^* = \bigcup_{i=1,2,\ldots} U^i$, де $U^1 \stackrel{def}{=} U$, $U^{i+1} = U \bullet U^i$, $i = 1, 2, \ldots$, а $\bullet$ – звичайна композиція

[2] На відміну від [28], позначення якої в цій праці будуть базовими, тут R_1, R_2, R_3 – таблиці (relations), а не схеми таблиць.

бінарних відношень. Тоді для транзитивного замикання R^* маємо: $R^* \overset{def}{=} \{r \mid \exists u_1, \exists u_2 (< u_1, u_2 > \in U) \wedge r(A_1) = u_1 \wedge r(A_2) = u_2)\}$. Задати R^* також можна через найменший розв’язок відповідного рівняння: $X = R \cup R \underset{r_1(A_2)=r_2(A_1)}{\otimes} X$ [8].

Таким чином, транзитивне замикання двоатрибутних таблиць моделює класичне транзитивне замикання бінарних відношень.

Операції транзитивного замикання також приділяє увагу Д. Мейєр (D. Maier) (1983 р.) [25]. Автор показує, що не існує виразу реляційної алгебри, який би задавав транзитивне замикання двоатрибутної таблиці.

Отже, визначення повноти реляційної алгебри можна оспорити, оскільки є принаймні одна операція над таблицями, що не виразна в реляційній алгебрі. Якщо говорити точно, то операцію транзитивного замикання не можна побудувати з сигнатурних операцій реляційної алгебри та селекторних функцій операцією суперпозиції в розумінні [22]. Цей результат залишається в силі, якщо дозволити користуватися константами [6].

Крім реляційної алгебри, у книзі [25] розглянуто ще три системи запитів: реляційне числення кортежів, реляційне числення доменів та табло. Показано, що і числення кортежів, і числення доменів еквівалентні за виразною силою реляційній алгебрі. Для кожного з цих числень представлено дві інтерпретації – необмежену та обмежену – і введено клас «безпечних» виразів, для яких обидві інтерпретації збігаються. Автор зазначає, що табло запитів не володіють такою виразною силою, як реляційна алгебра, проте вони можуть представити будь-який алгебраїчний вираз, що містить операції вибору (селекції) по рівності, проекції і з’єднання, а також після виконання деяких розширень – і операції різниці та об’єднання. Табло запитів цікаві тим, що їх можна оптимізувати для зменшення до мінімуму кількості з’єднань у початковому алгебраїчному виразі.

У кінці 1990-х років К. Дж. Дейт (C. J. Date) та Х. Дарвен (H. Darwen) запропонували новий «мінімальний» варіант алгебри – Алгебру A. Базисом цієї алгебри [3] є операції реляційного заперечення (доповнення), реляційної кон’юнкції (або диз’юнкції) та проекції (видалення атрибута). Реляційні аналоги логічних операцій визначаються в термінах відношень на основі звичайних теоретико-множинних операцій і дозволяють виражати операції перетину, декартового з’єднання, природного з’єднання, об’єднання відношень і т.д. Операції перейменування атрибутів, з’єднання загального вигляду, різниці відношень виражаються шляхом комбінування базових операцій.

Огляд Алгебри A Дейта – Дарвена здійснено С. Д. Кузнєцовим у [21]. Автор математично строго описує базові операції цієї алгебри: реляційне доповнення <NOT>, видалення атрибута <REMOVE>, перейменування атрибута <RENAME>, реляційна кон’юнкція <AND>, реляційна диз’юнкція <OR> – та розглядає їхні властивості. Для кожної операції наводяться приклади. Показано, що п’яти базових операцій Алгебри A достатньо, щоб виразити всі операції алгебри Кодда, та продемонстровано, що кількість базових операцій можна скоротити до трьох: <sh> (або <pi>), <RENAME>, <REMOVE>, де <sh> – реляційний варіант штриха Шеффера, <pi> – реляційний варіант стрілки Пірса, а то й до двох, оскільки операція перейменування атрибута <RENAME> теж є надлишковою.

Р. Елмасрі (R. Elmasri) та С. Наватхе (S. Navathe) (2000 р.) виділяють типи запитів, які не можна виразити в традиційній реляційній алгебрі, і пропонують додаткові реляційні операції, що дозволяють їх виразити та, як пишуть автори, збільшують виразну силу реляційної алгебри [43]. До цих запитів належать запити, які задають математичні агрегатні функції на наборах значень БД, і запити, які передбачають групування рядків таблиці за значеннями деяких з її атрибутів та застосування агрегатної функції незалежно до кожної групи. Для задання цих типів запитів автори визначають операцію агрегатної функції (AGGREGATE FUNCTION operation): <grouping attributes> <function list> (*R*), де <grouping attributes> – список атрибутів таблиці *R*, за якими відбувається групування, <function list> – це список пар (<function>, <attribute>). Для кожної пари <function> – це одна з агрегатних функцій SUM, AVERAGE, MAXIMUM, MINIMUM, COUNT, а <attribute> – це атрибут таблиці *R*, до значень якого застосовується відповідна агрегатна функція. Подібне означення агрегатної функції можна знайти в [58] (2011 р.).

Ще один тип операцій, який узагалі не можна визначити в традиційній реляційній алгебрі і про який коротко згадується в [43], – рекурсивне замикання. У цій книзі на прикладі показано, як можна задати деякі види рекурсивних запитів. Операція, яка дозволяє обчислювати рекурсивні взаємозв’язки так довго, як триватиме рекурсія,

[3] Щоб не суперечити традиційній алгебраїчній термінології, краще було б говорити про базис сигнатури або просто про сигнатуру.

має назву «транзитивне замикання». Дослідження рекурсивних запитів у SQL, виявлення структури таких запитів та задання їх семантики здійснено у [4].

Один із розділів книги [27] (2001 р.) присвячено різним операціям над таблицями, які реалізовані за допомогою реляційних операторів і розширень цих операторів, що є в мові SQL. Р. Райордан (R. Riordan) розглядає такі додаткові реляційні операції, як агрегування (summarize) та розширення (extend). Операція агрегування повертає результати, що містять сумарні дані, згруповані за вказаним значенням атрибутів. Цю операцію автор реалізує за допомогою виразу GROUP BY SQL-оператора SELECT. Розглянуто агрегатні функції AVERAGE, COUNT, SUM, MAXIMUM і MINIMUM. Операція розширення дозволяє створювати віртуальні атрибути, значення яких обчислюються на основі констант, збережених у БД, проте які не зберігаються (в БД) як певні значення, що записуються у файл БД на фізичному рівні.

Крім того, у книзі описано три додаткові оператори, які підтримуються продуктами корпорації Microsoft: TRANSFORM, ROLLUP і CUBE. Кожен з цих операторів являє собою окремий спосіб агрегування і представлення даних. Оператор TRANSFORM підтримується механізмом БД Microsoft Jet та використовується для створення перехресних запитів (crosstab query). Оператор ROLLUP – це логічне розширення операції GROUP BY, яке дозволяє отримати сумарні значення для груп або підгруп. Оператор CUBE концептуально схожий на ROLLUP, але, на відміну від нього, обчислює не тільки сумарні значення для кожного стовпця, що перераховані в списку атрибутів оператора GROUP BY, а й сумарні значення для додаткових груп. Оператор CUBE формує результуючий набір, який являє собою багатовимірний куб. При використанні умови GROUP BY з оператором CUBE потрібно враховувати такі факти і норми [34]:

- якщо в умові GROUP BY є n стовпців або виразів, СУБД SQL Server у наборі результатів повертає 2n-1 можливу комбінацію;
- значення NULL у наборі результатів указують, що ці особливі рядки створені в результаті роботи оператора CUBE;
- з оператором CUBE не можна використовувати ключове слово ALL.

Оператори ROLLUP та CUBE підтримуються СУБД SQL Server 6.5 (і вище).

До додаткових реляційних операцій Г. Ґарсіа-Моліна (H. Garcia-Molina) та ін. (див., наприклад, [11] (2004 р.) або [45] (2008 р.)) відносять операції сортування, розширеної проекції, групування, агрегування та операції зовнішніх з'єднань. Дамо коротку характеристику цих операцій у термінах вказаних джерел. Оператори агрегування дозволяють обчислювати різні загальні показники, що стосуються вмісту того чи іншого стовпчика таблиці: суми (оператор SUM), мінімальні (оператор MIN), максимальні (оператор MAX) та середні значення (оператор AVERAGE), а також кількості рядків (оператор COUNT). Відзначено, що оператори агрегування не належать до компетенції реляційної алгебри як такої, але використовуються в контексті оператора групування.

Оператор групування $\gamma_L(R)$ поєднує в собі можливості групування та агрегування. Нижній індекс оператора γ задає список L елементів, кожен з яких є або одним з атрибутів таблиці R, до яких застосовується оператор γ (це так звані групуючі атрибути), або оператором агрегування, який застосовується до атрибутів таблиці. Схема виконання оператора $\gamma_L(R)$ така:

1. Розбити множину рядків таблиці R на групи, кожна з яких включає всі рядки, що містять певні (зокрема, однакові) значення групуючих атрибутів зі списку L. Якщо групуючі атрибути не задані, таблиця R розглядається як єдина група.

2. Для кожної групи створити по одному рядку, який містить:

a. значення групуючих атрибутів, що відповідають групі;

b. результати агрегування, обчислені за вказаними атрибутами всіх рядків групи, заданих у списку L.

Розглянемо оператор сортування. Оператор сортування $\tau_L(R)$, де R – деяка таблиця, а L – список атрибутів R, перетворює таблицю у список рядків, які впорядковані відповідно до деякого критерію. Результатом обчислення виразу $\tau_L(R)$ є та сама таблиця R, але її рядки впорядковані за допомогою списку L. Наприклад, якщо L – список атрибутів $A_1, A_2, ..., A_n$, то рядки спочатку впорядковуються за значеннями атрибута A_1. Потім рядки, які мають однакові значення для атрибута A_1, впорядковуються, враховуючи значення атрибута A_2. Якщо існують рядки, у яких збігаються значення для атрибутів A_1 і A_2, то вони сортуються за значеннями атрибута A_3 і т. д. Після того як рядки, у яких збігаються значення для атрибутів $A_1, A_2, \ldots, A_{n-1}$, впорядковані за вмістом A_n, значення інших атрибутів до уваги не беруться. Таким чином, результатом виконання оператора сортування є список рядків, а не множина. Цей оператор також докладно розглянутий у [28, с. 159–165].

Зауважимо тільки, що вказане вище відношення на рядках таблиці, взагалі кажучи, є передпорядком (а не порядком), побудованим за логічною схемою лексикографічного добутку відношень.

До додаткових операцій реляційної алгебри Г. Ґарсіа-Моліна та ін. також відносять операцію розширеної проекції. Список L оператора розширеної проекції $\pi_L(R)$ може містити не тільки атрибути таблиці R, а й вирази вигляду $x \to y$, що передбачають перейменування атрибута x таблиці R на атрибут y, або вирази вигляду $E \to z$, де E – вираз, що містить атрибути таблиці R, константи й арифметичні та рядкові оператори, а z – новий атрибут, який містить значення, отримане в результаті обчислення виразу E. Результатом застосування операції розширеної проекції є таблиця з атрибутами зі списку L (враховуючи варіанти їх перейменування). Автори зазначають, що рядки-дублікати, отримані внаслідок виконання операції розширеної проекції, не вилучаються.

Операції зовнішніх з’єднань детально розглянемо в подальшому.

К. Дж. Дейт (2005 р.) розглядає декілька додаткових реляційних операцій: напівз’єднання (semijoin), напіврізницю (semiminus), розширення (extend), агрегування (summarize), а також описує операції групування (group), розгрупування (ungroup) та транзитивне замикання (tclose) [15]. Синтаксис виразів реляційної алгебри базується на використанні мови Tutorial D. Коротко опишемо згадані вище операції в термінах [15].

Операцію напівз’єднання таблиць К. Дж. Дейт визначає як і в [19; 46].

Операція напіврізниці між таблицями a і b (у вказаному порядку, бо операція не комутативна) aSEMIMINUSb визначена через операції різниці та напівз’єднання: aMINUS(aSEMIJOINb), де MINUS – операція різниці двох таблиць однієї схеми. Отже, результуюча таблиця містить ті рядки таблиці a, які не входять у (природне) з’єднання таблиць a і b.

Операція EXTEND породжує таблицю, ідентичну заданій, з точністю до того, що вона містить додатковий атрибут, значення якого отримано внаслідок деяких скалярних обчислень. Призначення операції розширення EXTEND полягає в підтримці обчислювальних можливостей реляційної алгебри.

Оператор агрегування дозволяє виконувати обчислення в межах значень атрибута таблиці по всіх рядках. До типових різновидів операції агрегування належать COUNT, SUM, AVG, MAX, MIN, ALL, ANY, COUNTD, SUMD і AVGD. Суфікс «D» в операціях COUNTD, SUMD і AVGD означає, що перед виконанням цієї узагальненої операції потрібно видалити всі надлишкові значення, тобто значення, які повторюються (від англ. distinct – різний).

Транзитивним замиканням таблиці a називається таблиця a^+ тієї самої схеми, що є надмножиною таблиці a, тобто рядок $\{X,x \quad Y,y\}$ з’являється в таблиці a^+, якщо він належить таблиці a або існує така послідовність значень $z1,z2,\ldots,zn$, що всі кортежі $\{X,x \quad Y,z1\}$, $\{X,z1 \quad Y,z2\},\ldots$, $\{X,zn \quad Y,y\}$ з’являються в a. Цю операцію в інших позначеннях проаналізовано раніше.

Оператори GROUP (групування) і UNGROUP (розгрупування) потрібні для прямого та оберненого перетворення таблиць, що містять атрибути, значеннями яких є таблиці, в таблиці, що мають атрибути з примітивними доменами значень (по суті, перебувають у першій нормальній формі).

Автор підкреслює, що не всі використовувані операції примітивні. Наприклад, операція агрегування може бути змодельована за допомогою операції розширення.

Додаткові операції, запропоновані К. Дж. Дейтом, а саме агрегування та розширення, також описано у [40].

Як альтернативу реляційній алгебрі К. Дж. Дейт коротко розглядає реляційне числення. Синтаксис реляційного числення введено, взявши за зразок версію числення мови Tutorial D. На прикладі показано, як можна використовувати алгоритм редукції Кодда для перетворення довільного виразу реляційного числення в еквівалентний вираз реляційної алгебри. Також обговорюється, як можна розширити числення кортежів з метою підтримки обчислювальних можливостей, які в реляційній алгебрі забезпечуються операціями EXTEND і SUMMARIZE.

До додаткових операцій, розглянутих у [15], Х. Дарвен (2009 р.) додає дві нові: WRAP та UNWRAP [40]. Операція WRAP дозволяє об’єднати значення деяких атрибутів у кожному рядку таблиці та сформувати в результаті єдине значення з новим атрибутом (цей атрибут відмінний від атрибутів таблиці, до якої застосовується операція), яке саме по собі є самостійним рядком (отже, не перебуває в першій нормальній формі). Операція UNWRAP є оберненою до WRAP.

Різні варіанти операції з’єднання (join) розглянуто у [24] (2006 р.): тетаз’єднання (theta-join), еквіз’єднання (equi-join), напівз’єднання (semi-join), природне з’єднання (natural join), природне напівз’єднання (natural semi-join), природне еквіз’єднання (natural equi-join).

Також О. С. Марков та К. Ю. Лісовський пропонують до операцій реляційної алгебри відносити симетричну різницю (Symmetrical Difference). Ця операція визначається за аналогією з одноіменною операцією в теорії множин, при цьому передбачається, що таблиці, до яких вона застосовується, мають однакові схеми.

У статтях [54] (2003 р.) та [53] (2004 р.) запропоновано нову агрегатну функцію LIST. Ця функція з'єднує значення одного або декількох атрибутів таблиці в єдине значення, яке буде списком цих значень, можливо, впорядкованим. Автор визначає синтаксис та семантику функції LIST і показує, яким чином ця функція може бути корисною на практиці.

Особливе значення Null

Бажання оперувати невизначеною та неповною інформацією в БД призвело до включення невизначених значень (null values) у реляційну модель і мови запитів. Null-значення – це, змістовно кажучи, не значення, а деякий маркер, який вказує на те, що значення невідоме. Уперше розширення реляційної алгебри на таблиці, які можуть містити невизначені значення, було запропоноване Е. Коддом [36]. Він також ввів тризначну логіку для роботи з невизначеними значеннями. Ця логіка має три значення істинності: істина (True), хиба (False) та невідомо (Unknown). Зауважимо, що ця логіка є давно відомою сильною тризначною логікою Кліні, яку було вперше введено в теорії алгоритмів С. К. Кліні в монографії [18] (див. також [9; 10]).

Основні підходи до розширення реляційної моделі БД для маніпулювання неповними або неточними даними розглянуто та проаналізовано в оглядовій роботі [1] (1989 р.). У статті нараховується 48 джерел з цієї теми. Огляд складається з двох частин, які відображають два напрями досліджень: по-перше, дослідження БД з порожніми або невизначеними значеннями, коли повністю відсутня інформація про значення окремих атрибутів на деяких об'єктах, і, по-друге, вивчення БД з неповною, частковою або неточною інформацією про значення атрибутів на об'єктах, розробку способів представлення такої інформації в реляційних БД. У роботі проаналізовано підхід Е. Кодда щодо розширення реляційної алгебри, наведено переваги такого підходу та вказано низку нерозв'язаних проблем, що приводять до некоректності розширеної алгебри. Подоланням цих проблем займалося багато авторів, їхні підходи й розглянуто та описано в [1]. Зокрема, зауважується, що для моделювання неповної та неточної інформації використовується ймовірнісний підхід та апарат теорії нечітких множин (fuzzy sets).

Питання про використання null-значень у теорії реляційних БД остаточно не розв'язане. Основоположник реляційного підходу Е. Кодд вважав null-значення невід'ємною частиною реляційної моделі. Так, до 12 канонічних правил, які має задовольняти кожна реляційна СУБД, він включив правило, що стосується використання саме Null: Systematic Treatment of Null Values – систематична обробка невизначених (Null) значень [39] (у цьому контексті краще говорити про семантичну обробку).

Разом з тим відомий популяризатор реляційного підходу в БД К. Дж. Дейт виступає проти використання null-значень, заявляючи, що «невизначені значення (NULL) і тризначна логіка є помилковими поняттями і їм немає місця в чітких формальних системах, подібних реляційній моделі» [15]. У книзі [15] автор робить спробу обґрунтувати свою думку про те, що поняття невизначених значень є помилковим, та коротко описує альтернативний спосіб обробки відсутньої інформації з використанням спеціальних значень. Також розглянуто наслідки наявності невизначених значень у первинних (primary) і зовнішніх (foreign) ключах та описано операції, які трапляються в контексті невизначених значень та тризначної логіки, а саме операції зовнішнього з'єднання (outer join).

Зауважимо, що спеціальне невизначене значення (Null) у реляційних БД принципово відрізняється від спеціального невизначеного значення (ω), яке використовується в методі нерухомої точки (fix point method) для моделювання часткових функцій тотальними функціями на домені, поповненому спеціальним значенням ω [8; 23; 41]; таке поповнення своєю чергою потрібне для задання денотаційної та операційної семантики рекурсивних програм [23].

Проявом зазначеної різниці є, наприклад, те, що тотальні функції, які моделюють часткові функції, завжди зберігають спеціальне значення, а операції кон'юнкції та диз'юнкції сильної тризначної логіки Кліні – не зберігають (зокрема, $Unknown \vee True = True$, $Unknown \wedge False = False$).

Зупинимося більш детально на операціях зовнішнього з'єднання. Уперше термін і концепцію зовнішнього з'єднання ввів у 1971 р. І. Хез (I. Heath). На думку Е. Кодда [38] (1990 р.), операції зовнішнього з'єднання є невід'ємною частиною реляційної моделі. Розглядають три види операцій зовнішнього з'єднання: ліве зовнішнє з'єднання (left outer join), праве зовнішнє з'єднання (right outer join) та повне зовнішнє з'єднання (full outer join) [11; 19; 20; 27; 28; 43; 46; 58]. Операції зовнішнього з'єднання призначені для врахування тих рядків таблиць-аргументів, які не потрапили в результат вихідного звичайного (внутрішнього, inner) з'єднання. При цьому для розширення рядків до надсхеми використовуються саме значення Null. Таким чином, кожна операція зовнішнього з'єднання логічно підпорядкована своїй операції внутрішнього з'єднання.

Операція лівого (правого) зовнішнього з'єднання приведе до таблиці, в якій будуть наявні всі рядки з таблиці, що розташована ліворуч (праворуч) від оператора зовнішнього з'єднання, а для позначення відсутніх значень буде використовуватися значення Null. Таблиця-результат операції повного зовнішнього з'єднання міститиме рядки з обох таблиць, а для позначення значень атрибутів рядків, які не ввійшли в результат внутрішнього з'єднання, використовуватиметься значення Null.

У книзі [38] автор описує три види операцій зовнішнього з'єднання: зовнішні еквіз'єднання (outer equi-join), зовнішні природні з'єднання (outer natural join) та зовнішні Т-з'єднання (outer T-join). На прикладі показано, чим зовнішні природні з'єднання відрізняються від зовнішніх еквіз'єднань. Е. Кодд вказує на те, що операцію внутрішнього природного з'єднання (inner natural join) можна виразити через операції проекції та внутрішнього еквіз'єднання (inner equi-join), у той час як для операцій зовнішнього еквіз'єднання та зовнішнього природного з'єднання таке представлення не має місця. Пояснено зв'язок між внутрішнім та зовнішнім еквіз'єднаннями.

Свій погляд на невизначені значення та неповну інформацію в БД пропонує Д. Мейєр [25]. У його книзі розглянуто та проаналізовано підходи Е. Кодда, М. Лакруа (M. Lacroix) з А. Піроттом (A. Pirotte) та К. Заніоло (C. Zaniolo) щодо узагальнення операції з'єднання. Автор, зокрема, на контрприкладах показує, що всі запропоновані операції неасоціативні.

Питання неасоціативності операцій зовнішнього з'єднання також порушено в роботі [44] (1994 р.). У цій статті охарактеризовано ті зовнішні з'єднання, які є асоціативними, та ті, що такими не є. Використовуючи диз'юнктивну нормальну форму, автор показав, що деякі запити об'єднання даних (data merging queries) не можна задати засобами бінарних зовнішніх з'єднань, та запропонував альтернативну процедуру задання цих запитів.

Операціям зовнішнього з'єднання притаманні декілька «неприємних властивостей» [14] (2001 р.). Серед них такі:

1. Зовнішнє Θ-з'єднання не є селекцією від результату декартового добутку [4].

2. Селекція не поширюється на результати операції зовнішнього Θ-з'єднання.

3. У тризначній логіці вираз $A \le B$ – не те саме, що вираз $A < B$ OR $A = B$ (в контексті зовнішнього з'єднання).

4. У тризначній логіці операції Θ-порівняння не є транзитивними.

5. Зовнішнє природне з'єднання не є проекцією внутрішнього з'єднання за еквівалентністю.

Крім зовнішніх з'єднань, деякі автори виділяють «зовнішні» версії інших операторів реляційної алгебри, зокрема об'єднання, перетину та різниці [38; 43]. Р. Елмасрі (R. Elmasri) та С. Наватхе (S. Navathe) описують операцію зовнішнього об'єднання (outer union) таблиць [43]. Операція використовується для об'єднання рядків двох таблиць, схеми яких різні. Ці таблиці повинні бути частково сумісні по об'єднанню, тобто деякі атрибути мають бути однакові для обох таблиць. Передбачається, що список однакових атрибутів містить ключ для обох таблиць. Рядки, які мають той самий ключ у вихідних таблицях, представлені в результуючій таблиці лише один раз і мають значення для кожного атрибута. Атрибути, які є різними для цих таблиць, теж входять до схеми результату, а рядки, що не мають значення для цих атрибутів, поповнюються значенням Null.

У книзі як приклад розглянуто зовнішнє об'єднання двох таблиць: STUDENT(Name, SSN, Department, Advisor) і FACULTY(Name, SSN, Department, Rank). У результаті отримуємо таблицю R(Name, SSN, Department, Advisor, Rank), в яку включено всі рядки обох таблиць. Рядки таблиці STUDENT матимуть значення Null для атрибута Rank, а рядки таблиці FACULTY матимуть значення Null для атрибута Advisor. Рядок, який в обох таблицях має однакові значення для атрибутів Name, SSN, Department, матиме значення для всіх атрибутів, тобто для атрибутів Name, SSN, Department, Advisor, Rank.

Е. Кодд, крім операції зовнішнього об'єднання, задає також операції зовнішнього перетину (outer set intersection) та зовнішньої різниці (outer set difference) [38]. При цьому вводиться поняття, яке є більш загальним за поняття рівності рядків – поняття близького аналога (close

[4] Краще говорити про декартове з'єднання.

counterpart). Рядок таблиці S є близьким аналогом рядка таблиці *T*, якщо виконуються умови [5]:

● таблиці *S* і *T* мають однакові первинні ключі;

● два рядки (один з *S*, а інший з *T*) мають рівні значення первинного ключа;

● попарна рівність відомих значень зберігається для тих атрибутів таблиці *S*, що відповідають атрибутам таблиці *T*.

Перед тим, як застосовувати зовнішні теоретико-множинні операції, потрібно таблиці-аргументи зробити сумісними за об'єднанням. Для цього Е. Кодд задає операцію **extend**, яка кардинально відрізняється від операції розширення, описаної в попередньому підрозділі. Після застосування цієї операції до кожної з таблиць їх схеми поповнюються відсутніми атрибутами, а значеннями цих атрибутів є значення Null.

Задамо кожну зовнішню теоретико-множинну операцію в термінах [38].

Зовнішнє об'єднання $S \backslash \quad / T$ таблиці S з таблицею *T* здійснюється в три кроки:

1) формуємо таблицю $St = S$ **per** T ,

2) формуємо таблицю $Ts = T$ **per** *S*, де **per** позначає операцію **extend**,

3) формуємо таблицю $S \backslash \cup / T \quad St \cup Ts$.

Зовнішня різниця $S \backslash - / T$ між таблицями S та T здійснюється в чотири кроки:

1) формуємо таблицю $St = S$ **per** T ,

2) формуємо таблицю $Ts = T$ **per** S , де **per**, як і раніше, позначає операцію **extend**,

3) формуємо таблицю semi-equi-join $U = St[sem =]Ts$,

4) формуємо таблицю $S \backslash - / T = St - U$, де $sem =$ – це операція напівз'єднання semi-equi-join, описана вище.

Зовнішній перетин $S \backslash \cap / T$ таблиць S і T передбачає п'ять кроків:

1) формуємо таблицю $St = S$ **per** T ,

2) формуємо таблицю $Ts = T$ **per** S ,

3) формуємо таблицю $U = St\,[sem =]Ts$,

4) формуємо таблицю $V = Ts\,[sem =]St$, де $sem =$ – операція semi-equi-join, як і раніше,

5) формуємо таблицю $S \backslash \cap / T = U \cap V$.

Для виключення дублікатів рядків використовується поняття близького аналога.

Показано, що операції зовнішнього об'єднання та зовнішнього перетину комутативні на відміну від операції зовнішньої різниці.

Встановлено зв'язок між зовнішніми множинними операціями: $S \backslash \cup / T = (S \backslash - / T) \cup (S \backslash \cap / T)$ [6].

Як зазначалося на початку, Е. Клуг поповнив реляційну алгебру та реляційне числення агрегатними функціями і показав еквівалентність отриманих при цьому двох формальних мов [49]. Проте він не розглядав випадок, коли агрегатна функція як результат може мати значення Null. Цій проблемі та проблемі еквівалентності реляційної алгебри та реляційного числення, які поповнені Null-значеннями, присвячено статтю [33].

Гюнтер фон Бюльтцінгсловен (Günter von Bültzingsloewen) поповнює реляційне числення Е. Клуга двома логічними зв'язками $\top$ і $\perp$, які відображають істиннісне значення UNKNOWN у TRUE та FALSE відповідно. Множина $\Theta = \{=, \neq, <, \leq, >, \geq\}$ оператора порівняння θ доповнена спеціальним оператором $\equiv$, який приймає значення TRUE, якщо аргументи рівні у звичайному сенсі або обидва ω , і FALSE в інших випадках [7]. Інші оператори приймають значення UNKNOWN, якщо хоч один з аргументів рівний ω , TRUE, якщо всі аргументи істинні, і FALSE, якщо всі аргументи хибні. Реляційна алгебра теж розширена Null-значеннями. Показана еквівалентність розширених таким чином реляційної алгебри та реляційного числення.

Мультимножини

Багато мов, орієнтованих на роботу з БД, вимагають реляційну модель даних з мультимножинною семантикою (multi-set semantics), що передбачає розуміння таблиць (відношень) як мультимножин, тобто сукупностей з дублікатами. Питанню розширення можливостей БД за рахунок використання мультимножин приділяли увагу Л. Лібкін (L. Libkin), Л. Вонг (L. Wong) [51; 52], Дж. Ламперті (G. Lamperti), М. Мельчіорі (M. Melchiori), М. Занелла (M. Zanella) [50], Г. Ґарсіа-Моліна [11; 45], Дж. Ульман (J. D. Ullman), Дж. Уідом (J. Widom) [11; 30; 45; 60], К. Росс (K. Ross), Й. Стояновіч (J. Stoyanovich) [57], А. Зільбешатц (A. Silbeschatz) та ін. [58], а також вітчизняні науковці В. Н. Редько, Ю. Й. Брона, Д. Б. Буй, С. А. Поляков [5; 28]. Огляд літератури з використання мультимножин у базах даних станом на 2010 р. здійснили Д. Б. Буй та Ю. О. Богатирьова у статті [7], яка нараховує 9 джерел з цієї теми. Доповнимо цей огляд ще кількома працями.

У. Даял (U. Dayal) та ін. розширили реляційну модель до мультимножинної реляційної

[5] По суті, вводиться бінарне відношення на декартовому добутку таблиць $S \times T$.

[6] Це співвідношення – аналог добре відомої теоретико-множинної рівності $A \cup B = (A \setminus B) \cup (A \cap B) \cup (B \setminus A)$.

[7] Отже, по суті, йдеться про сильну рівність Кліні або узагальнену рівність $\simeq$ (див., наприклад, [16]).

моделі (multiset relational model), вводячи поняття мультимножинного відношення (multiset relation) [42] (1982 р.). Під мультимножинним відношенням R схеми $\underline{R}$ розуміють скінченну множину пар $\langle t, i \rangle$, де t – кортеж схеми $\underline{R}$, а i – кількість дублікатів кортежу t у відношенні R.

У статті [42] задано операції мультимножинної реляційної алгебри, а саме: селекцію σ, з'єднання $\bowtie$, проекцію π, операцію вилучення дублікатів δ – та розглянуто два підходи до визначення аналогів теоретико-множинних операцій (об'єднання, перетину та різниці). У роботі наведено список властивостей мультимножинної реляційної алгебри, які підкреслюють відмінності між цією алгеброю та традиційною реляційною алгеброю. Розглянемо деякі з цих властивостей:

1) операція з'єднання $\bowtie$ не ідемпотентна (точніше кажучи, основа результату та сама, що й в аргументів, але кількість дублікатів збільшується вдвічі);

2) операція вилучення дублікатів δ ідемпотентна;

3) операція вилучення дублікатів δ дистрибутивна відносно операцій $\bowtie$, $\cup, \cap$;

4) операція вилучення дублікатів δ комутує з операцією селекції σ: $\delta(\sigma R) = \sigma(\delta R)$;

5) операція δ комутує з операцією проекції: $\delta \pi_X R = \pi_X \delta R$.

У роботах [47] (1994 р.) та [48] (1996 р.) запропоновано розширену реляційну алгебру з мультимножинною семантикою. Визначено поняття таблиці (як мультимножини рядків), задано основні операції над таблицями: аналоги теоретико-множинних операцій (об'єднання та різниця), селекція, проекція, декартове з'єднання. Операції перетину та з'єднання розглядаються як похідні від основних операцій. Автори доводять, що їх можна виразити через операції різниці та селекції і декартового з'єднання відповідно. Крім цих операцій, автори вводять три додаткові операції: розширену проекцію, видалення дублікатів, групування та агрегатні функції. Доведено асоціативність декартового з'єднання, з'єднання, перетину та об'єднання і те, що операції проекції та селекції дистрибутивні відносно об'єднання.

Джон Ґрант (John Grant) та ін. пропонують підхід до розв'язання проблеми, що стосується можливості спільно маніпулювати даними в різних БД [56] (1993 р.). Введено модель мультибази даних (multidatabase model), представлено основні поняття та приклад такої БД. Наведемо основні означення.

Мультибаза даних – це сукупність БД, імена яких можуть бути задані явно або неявно. Мультитаблиця (multirelation) – це мультимножина таблиць, динамічно визначених мультитабличним запитом (multirelational query). Таблиці з мультитаблиці зазвичай належать різним БД із мультибази даних. Ці таблиці можуть мати різну кількість рядків та різні імена атрибутів. Мультитабличний запит визначає мультитаблицю з мультитаблиць за допомогою мультитабличних операцій.

У статті також визначено мультитабличну алгебру заданням її операцій: мультипроекція, мультиселекція, мультиоб'єднання, мультиперетин, мультирізниця, мультиз'єднання та мультидекартовий добуток (MPROJECT, MSELECT, MUNION, MINTERSECT, MDIFFERENCE, MJOIN і MPRODUCT відповідно). Опишемо декілька операцій у позначеннях [56].

Мультипроекція мультитаблиці є мультитаблицею: $\mathrm{MPROJECT}(\{R_1,\ldots,R_n\}; A_1,\ldots,A_m) = \{\mathrm{PROJECT}(R_1; A_1,\ldots, A_m),\ldots, \mathrm{PROJECT}(R_n; A_1,\ldots, A_m)\}$, де $\{R_1,\ldots,R_n\}$ – довільна мультитаблиця, $\mathrm{PROJECT}(R_i; A_1,\ldots, A_m)$ – звичайна проекція за атрибутами $A_1,\ldots, A_m$ таблиці R_i, якщо всі A_j – атрибути таблиці R_i; якщо ж деякий атрибут A_j не належить таблиці R_i, то проекція PROJECT $(R_i; A_1,\ldots,A_m)$ не існує. Автори зауважують, що результат може мати менше ніж n таблиць.

Розглянемо операцію мультиз'єднання мультитаблиць: $\mathrm{MJOIN}((\{R_{11},\ldots,R_{1n}\}, \{R_{21},\ldots,R_{2m}\} : C = \{\mathrm{JOIN}(R_{11}, R_{21} : C),\ldots, \mathrm{JOIN}(R_{11}, R_{2m} : C), \mathrm{JOIN}(R_{12}, R_{21} : C),\ldots, \mathrm{JOIN}(R_{1n}, R_{2m} : C)\} = \{\mathrm{JOIN}(R_{1i}, R_{2j} : C) \mid i = 1,\ldots,n, j = 1,\ldots, m\}$, де $\mathrm{JOIN}(R_{1i}, R_{2j} : C)$ – операція з'єднання таблиць, C – умова, за якою відбувається з'єднання. Результат може містити не більше ніж $n \cdot m$ таблиць.

Отже, по суті, мова йде про розширення стандартних операцій над таблицями на мультимножини таблиць за допомогою загальнозначної теоретико-множинної конструкції повного образу; точніше кажучи, йдеться про поширення унарних або бінарних операцій з множини на булеан цієї множини (див. також [3; 17]).

Розділ 4 статті [56] присвячено мультитабличному численню, а саме мультитабличному численню рядків. Це числення зводиться до реляційного числення у випадку, коли мультитаблиця в точності містить одну таблицю. Доведено, що мультитабличне числення не менш виразне за мультитабличну алгебру.

У книзі [58] розглянуто версію реляційної алгебри – мультимножинну реляційну алгебру. Її основні операції (селекцію, проекцію та з'єднання) визначено таким чином (використано позначення вказаної роботи):

1) якщо рядок t_1 таблиці r_1 має c_1 дублікат і задовольняє умову селекції σ_θ, то кількість входжень рядка t_1 у результуючу таблицю $\sigma_\theta(r_1)$ рівна c_1, тобто не змінюється;

2) для кожного екземпляра рядка t_1 таблиці r_1 існує екземпляр рядка $\pi_A(t_1)$ в таблиці $\pi_A(r_1)$, де через $\pi_A(t_1)$ позначено проекцію єдиного рядка t_1 на атрибут A (в загальноматематичних позначеннях мова йде про обмеження вигляду $t_1|\{A\}$ [2]);

3) якщо рядок t_1 таблиці r_1 має c_1 дублікатів і рядок t_2 таблиці r_2 має c_2 дублікатів, то кількість входжень рядка $t_1.t_2$ у таблицю $r_1 \times r_2$ рівна $c_1 \cdot c_2$, де $\times$ – операція з’єднання.

Операції об’єднання, перетину, різниці та агрегатні операції для мультимножинної реляційної алгебри автори не визначають, а лише зазначають, що ці операції можуть бути задані подібно до основних операцій і нічим не відрізняються від відповідних визначень у SQL, які розглянуто в розділі 3.5 цієї книги.

Інші розширення табличної алгебри

Вище розглянуто додаткові операції, що розширюють можливості табличної алгебри, проаналізовано питання використання null-значень та мультимножин у табличних БД. Проте існують й інші розширення табличної алгебри, деякі з них описано нижче.

Розширення алгебри та числення, запропоновані Е. Клугом, на атрибути, значеннями яких є таблиці (set-valued attributes), було здійснено Г. Озсойоглу (G. Özsoyoğlu) та ін. [55] (1987 р.). У роботі також представлено три нові оператори табличної алгебри, а саме: pack, unpack та aggregation-by-template. Авторами показано, що отримані при цьому розширенні мови є еквівалентними.

Статтю [59] (2008 р.) присвячено алгебрам БД, а саме: табличній алгебрі, вкладеним табличним алгебрам (nested relation algebra), об’єктно-орієнтованим алгебрам. Зроблено короткий огляд кожної алгебри: визначено структуру її об’єктів, розглянуто відповідні алгебраїчні операції; описано кожну нормальну форму чи інші обмеження, накладені моделлю, вказано деякі найбільш важливі результати про алгебраїчну еквівалентність, зазначено, чи розроблене відповідне числення.

Таким чином, описані вище розширення орієнтовані на табличні БД, що не перебувають у першій нормальній формі.

Джеймс Бредлі (James Bradley) пропонує нову розширену табличну алгебру, придатну для скорочення виразів, написаних природною квантифікованою мовою (natural quantifier language) COOL, а також, імовірно, і будь-якою іншою подібною природною квантифікованою мовою [32] (1996 р.). Ця алгебра містить усі традиційні операції табличної алгебри та три нетрадиційні: group-select, subgroup-select та possibility join. Автор зазначає, що ця розширена таблична БД використовується в проекті побудови інтерфейсу користувача комерційно доступної табличної БД. Цей інтерфейс дасть змогу користувачам маніпулювати розширеною табличною БД, або табличною об’єктно-орієнтованою БД, використовуючи мову COOL.

Розширення табличної алгебри Кодда операціями з рекурсивними об’єктами розглядається у статті В. В. Говорушка та В. Б. Новосельцева [12] (2004 р.). Автори вводять поняття рекурсивної таблиці та задають операції над таблицями. У роботі також доведено теорему про замкнутість розширеної табличної алгебри на основі операції об’єднання. Зазначається, що розроблений апарат може використовуватися як теоретична база при побудові нових структур БД з використанням сучасних засобів проектування.

Хесус М. Альмендрос-Хіменес(Jesus M. Almendros-Jimenez) [31] (2006 р.) пропонує табличну алгебру для декларативних мов. Вона ґрунтується на використанні операцій проекції, селекції, перейменування, декартового з’єднання, об’єднання та з’єднання. Автор зазначає, що цю алгебру можна використовувати для визначення предикатів та функцій декларативної мови.

Висновки

У статті в систематизованому вигляді подано огляд літератури з розширення реляційної (табличної) алгебри: додаткові операції реляційної алгебри, використання null-значень та мультимножин, інші розширення табличних БД. На основі аналізу першоджерел можна зробити такі висновки:

1) для задоволення потреб мов запитів (зокрема, SQL) класична таблична алгебра потребує поповнення її новими операціями: агрегування, напівз’єднання, операціями внутрішніх і зовнішніх з’єднань, зовнішніми множинними операціями;

2) використання null-значень у теорії табличних БД є дискусійним; запропоновано альтернативний спосіб обробки відсутньої інформації;

3) застосування мультимножин у табличних БД є природним для багатьох прикладних задач.

Список літератури

1. Брона И. И. Реляционные базы данных с неполными и неточными значениями (аналитический обзор) / И. И. Брона, Т. А. Малюта, В. В. Пасичник. – Новосибирск : [б. и.], 1989. – 53 с.
2. Буй Д. Б. Властивості відношення конфінальності та устрій множини часткових функцій / Д. Б. Буй, Н. Д. Кахута // Вісник Київського університету. Сер.: фіз.-мат. науки. – 2006. – Вип. 2. – С. 125–135.
3. Буй Д. Б. Властивості теоретико-множинних конструкцій повного образу та обмеження / Д. Б. Буй, Н. Д. Кахута // Вісник Київського університету. Сер.: фіз.-мат. науки. – 2005. – Вип. 2. – С. 232–240.
4. Буй Д. Б. Композиційна семантика рекурсивних запитів в SQL-подібних мовах / Д. Б. Буй, С. А. Поляков // Вісник Київського університету. Сер.: фіз.-мат. науки. – 2010. – Вип. 1. – С. 45–56.
5. Буй Д. Б. Композиційна семантика SQL-подібних мов: мультимножини, рядки, впорядковані таблиці / Д. Б. Буй, С. А. Поляков // Вісник Київського університету. Сер.: фіз.-мат. науки. – 1999. – Вип. 2. – С. 183–194.
6. Буй Д. Б. Рівномірна неперервність сигнатурних операцій табличних алгебр / Д. Б. Буй, Ю. Й. Брона, Н. Д. Кахута // Вісник КНУ. Dynamic system modeling and stability investigation. Theses of conference reports. – К., 2005. – С. 31.
7. Буй Д. Б. Сучасний стан теорії мультимножин / Д. Б. Буй, Ю. О. Богатирьова // Вісник Київського університету. Сер.: фіз.-мат. науки. – 2010. – Вип. 1. – С. 51–58.
8. Буй Д. Б. Теорія програмних алгебр композиційного типу та її застосування : дис. … д-ра фіз.-мат. наук : 01.05.03 – математичне та програмне забезпечення обчислювальних машин і систем / Д. Б. Буй. – К., 2002. – 365 с.
9. Буй Д. Б. Три замечания о трехзначной логике Клини / Д. Б. Буй, С. А. Поляков, Е. В. Шишацкая // The Fourth International Conference "Theoretical and Applied Aspects of Program Systems Development" (TAAPSD'2007, Ukraine, Berdynsk, September 4–9, 2007). – К. : Пульсари, 2007. – С. 47–51.
10. Буй Д. Трехзначные логики Клини и трехэлементные цепи / Д. Буй, Е. Шишацкая // International Book Series "Information Science & Computing" N.1. Supplement to the International Jornal "Information Technologies & Knowledge". – 2008. – Vol. 2. – P. 165–172.
11. Гарсиа-Молина Г. Системы баз данных / Г. Гарсиа-Молина, Дж. Ульман, Дж. Уидом. – М. : Вильямс, 2004. – 1088 с.
12. Говорушко В. В. Расширение алгебры Кодда операциями с рекурсивными объектами / В. В. Говорушко, В. Б. Новосельцев // Вестник Томского государственного университета. Сер.: «Математика. Кибернетика. Информатика». – 2004. – № 284. – С. 18–20.
13. Грей П. Логика, алгебра и базы данных / П. Грей. – М. : Машиностроение, 1989. – 368 с.
14. Дейт К. Дж. Введение в системы баз данных / К. Дж. Дейт. – [7-е изд.]. – М. : Вильямс, 2001. – 1072 с.
15. Дейт К. Дж. Введение в системы баз данных / К. Дж. Дейт. – [8-е изд.]. – М. : Вильямс, 2005. – 1328 с.
16. Катленд Н. Вычислимость. Введение в теорию рекурсивных функций / Н. Катленд. – М. : Мир, 1983. – 256 с.
17. Кахута Н. Д. Застосування теоретико-множинних конструкцій повного образу, обмеження, конфінальності та сумісності в табличних базах даних : дис. … канд. фіз.-мат. наук : 01.05.03 – математичне та програмне забезпечення обчислювальних машин і систем / Н. Д. Кахута. – К., 2010. – 116 с.
18. Клини С. К. Введение в метаматематику / С. К. Клини. – М. : ИЛ, 1957. – 526 с.
19. Коннолли Т. Базы данных. Проектирование, реализация и сопровождение. Теория и практика / Т. Коннолли, К. Бегг. – [3-е изд.]. – М. : Вильямс, 2003. – 1440 с.
20. Крёнке Д. Теория и практика построения баз данных / Д. Крёнке. – [8-е изд.]. – СПб. : Питер, 2003. – 800 с.
21. Кузнецов С. Д. Основы баз данных / С. Д. Кузнецов. – [2-е изд.]. – М. : Интернет-Университет Информационных Технологий : БИНОМ. Лаборатория знаний, 2007. – 484 с.
22. Мальцев А. И. Алгоритмы и рекурсивные функции / А. И. Мальцев. – М. : Наука, 1986. – 368 с.
23. Манна З. Теория неподвижной точки программ / З. Манна // Киб. сб. Нов. сер. – М. : Мир, 1978. – Вып. 15. – С. 38–100.
24. Марков А. С. Базы данных. Введение в теорию и методологию / А. С. Марков, К. Ю. Лисовский. – М. : Финансы и статистика, 2006. – 512 с.
25. Мейер Д. Теория реляционных баз данных / Д. Мейер. – М. : Мир, 1987. – 608 с.
26. Пасічник В. В. Організація баз даних та знань / В. В. Пасічник, В. А. Резніченко. – К. : Видавнича група BHV, 2006. – 384 с.
27. Райордан Р. Основы реляционных баз данных / Р. Райордан. – М. : Русская Редакция, 2001. – 384 с.
28. Реляційні бази даних: табличні алгебри та SQL-подібні мови / В. Н. Редько, Ю. Й. Брона, Д. Б. Буй, С. А. Поляков. – К. : Академперіодика, 2001. – 198 с.
29. Риге Ж. Бинарные отношения, замыкания, соответствия Галуа / Ж. Риге // Киб. сб. : сб. переводов. – М. : Иностр. лит-ра, 1963. – Вып. 7. – С. 129–185.
30. Ульман Дж. Введение в системы баз данных / Дж. Ульман, Дж. Уидом. – М. : Лори, 2000. – 374 с.
31. Almendros-Jimenez J. M. An Extended Relational Algebra for Declarative Programming / Jesus M. Almendros-Jimenez // New Generation Computing. – 2006. – Vol. 24, Issue 2. – P. 129–184.
32. Bradley J. Extended Relational Algebra for Reduction of Natural Quantifier COOL expression / J. Bradley // Journal of Systems and Software. – 1996. – Vol. 33, Issue 1, April. – P. 87–100.
33. Bültzingsloewen G. von. Translating and Optimizing SQL Queries Having Aggregates / Günter von Bültzingsloewen // Proceedings of the 13th VLDB Conference. – Brighton, 1987. – P. 235–243.
34. Celko J. Analytics and OLAP in SQL / J. Celko. – Morgan Kaufmann, 2006. – 208 p.
35. Codd E. F. A Relational Model of Data for Large Shared Data Banks / E. F. Codd // Comm. of ACM. – 1970. – Vol. 13, № 6. – P. 377–387.
36. Codd E. F. Extending the Database Relational Model to Capture more Meaning / E. F. Codd // ACM Transactions on Database Systems. – 1979. – Vol. 4, № 4. – P. 397–434.
37. Codd E. F. Relational Completeness of Data Base Sublanguages / E. F. Codd // Data Base Systems. – NY : Prentice-Hall, 1972. – P. 65–93.
38. Codd E. F. The Relational Model for Database Management: Version 2 / E. F. Codd. – Addison-Wesley, 1990. – 541 p.
39. Codd's rules [Electronic resource]. – Mode of access: http://itsy.co.uk/ac/0405/Sem3/44271_DDI/Lec/3_CoddsRules.htm. – Title from the screen.
40. Darwen H. An Introduction to Relational Database Theory / H. Darwen. – Ventus Publishing Aps, 2009. – 231 p.
41. Davey B. A. Introduction to Lattice and Order / B. A. Davey, H. A. Priestly. – Cambridge : Cambridge University Press, 1990. – 248 p.
42. Dayal U. An Extended Relational Algebra with Control Over Duplicate Elimination / U. Dayal, N. Goodman, R. H. Katz // Proceeding PODS'82 Proceedings of the 1st ACM SIGACT-SIGMOD symposium on Principles of Database Systems. – 1982. – P. 117–123.
43. Elmasri R. Fundamental of Database Systems / R. Elmasri, S. Navathe. – [3rd Edition]. – Addison-Wesley, 2000. – 893 p.

44. Galindo-Legaria C. A. Outerjoins as Disjunctions / C. A. Galindo-Legaria // Series: CWI. Department of Computer Science. – CWI, 1994. – P. 1–17.
45. Garcia-Molina H. Database Systems: The Complete Book / H. Garcia-Molina, J. D. Ullman, J. Widom. – [2nd Edition]. – Prentice Hall, 2008. – 1119 p.
46. Gardarin G. Relational Databases and Knowledge Bases / G. Gardarin, P. Valduriez. – Addison-Wesley, 1989. – 448 p.
47. Grefen Paul W. P. J. A Multi-Set Extended Relational Algebra. A Formal Approach to a Practical Issue / Paul W. P. J. Grefen, Rolf A. De By // 10th International Conference on Data Engineering, ICDE, February 14–18, 1994, Houston, TX, USA. – P. 80–88.
48. Grefen Paul W. P. J. Extending a Multi-Set Relational Algebra to a Parallel Environment / Paul W. P. J. Grefen, J. Flokstra // Distributed and Parallel Databases. – 1996. – Vol. 4. – P. 81–99.
49. Klug A. Equivalence of Relational Algebra and Relational Calculus Query Languages Having Aggregate Functions / A. Klug // J. ACM 29. – 1982. – № 3. – P. 699–717.
50. Lamperti G. On Multisets in Database Systems / G. Lamperti, M. Melchiori, M. Zanella // Multiset Processing: Mathematical, Computer Science, and Molecular Computing Points of View, number 2235 in Lecture Notes in Computing Since. – Berlin : Springer-Verlag, 2001. – P. 147–215.
51. Libkin L. Query Language for Bags and Aggregates Function / L. Libkin, L. Wong // Journal of Computer and System Sciences. – 1997. – Vol. 55, № 1. – P. 241–272.
52. Libkin L. Some Properties of Query Language for Bags / L. Libkin, L. Wong // Proceedings of 4th International Workshop on Database Programming Languages. – NY, 1993. – P. 97–114.
53. Litwin W. Explicit and Implicit LIST Aggregate Function for Relational Databases [Electronic resource] / W. Litwin // Proceedings of IASTED International Conference DATABASES and APLICATIONS, February 17–19, 2004, Innsbruck, Austria. – Mode of access: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.144.8564&rep=rep1&type=pdf. – Title from the screen.
54. Litwin W. Explicit and Implicit LIST Aggregate Function for Relational Databases / W. Litwin // Proceedings of IASTED International Conference DATABASES and APLICATIONS, February 17–19, 2004, Innsbruck, Austria.
55. Özsoyoğlu G. Extended Relational Algebra and Relational Calculus with Set-Valued Attributes and Aggregate Functions / G. Özsoyoğlu, Z. M. Özsoyoğlu, V. Matos // ACM Transactions on Database Systems. – 1987. – Vol. 12, № 4. – P. 566–592.
56. Query Languages for Relational Multidatabases // J. Grant, W. Litwin, N. Roussopoulos, T. Sellis // VLDB Journal. – 1993. – Vol. 2. – P. 153–171.
57. Ross K. A. Symmetric Relations and Cardinality-bounded Multisets in Database Systems / K. A. Ross, J. Stoyanovich // Very Large Database Endowment: international conference, August 31 – September 03, 2004, Toronto, Canada: proceedings. – 2004. – Vol. 30. – P. 912–923.
58. Silbeschatz A. Database System Concepts / A. Silbeschatz, H. Korth, S. Sudarshan. – [6th Edition]. – McGraw-Hill, 2011. –1376 p.
59. Suri Pushpa R. Database Algebra / Pushpa R. Suri, Sudesh Rani // Journal of Theoretical and Applied Information Technology. – 2008. – Vol. 4, № 7. – P. 595–602.
60. Ullman J. D. A First Course in Database Systems / J. D. Ullman, J. Widom. – [3rd Edition]. – Prentice Hall, 2007. – 592 p.

D. Bui, I. Glushko

EXPANSION OF THE SIGNATURE OF CODD'S RELATIONAL (TABLE) ALGEBRAS: CURRENT STATE

Article is devoted to the review of the existing literature about extended relational (table) algebras. Additional operations which expand possibilities of relational algebras are considered. These are aggregation operators, grouping of tuples, sorting operator, semijoin, outer joins and others. Use of null-values is analyzed when operating with uncertain and/or incomplete information in databases. The question of expansion of opportunities of databases due to use of multisets is also considered, because there are many applications the most peculiar feature of which are multiplicity and repeatability of data.

Keywords: relational databases, relational (table) algebra, extending relation algebra, multiset.

Матеріал надійшов 07.05.2015