

РОБОТА З РЕЛЯЦІЙНОЮ МОДЕЛЛЮ ДАНИХ

Карпішен Б.С., асистент
Перепелиця А.С., здобувач РВО доктор філософії
Харківський національний автомобільно-дорожній університет
м. Харків, Україна, moonustineyes@gmail.com

Анотація: Для вирішення проблеми створення та використання виробничого потенціалу потрібно оптимально розподіляти ресурси використовуючи комп'ютерну техніку. У роботі розглядається поняття моделі даних та їх структури. Розглянуто способи взаємодії з реляційними моделями даних.

Ключові слова: інформаційні системи, реляційні бази даних, проектування, реляційна модель, кортежі

Рівень організації доступу до даних, а також високий ступінь автоматизації обробки даних складають основу сучасної інформаційної індустрії. Структуровані за певними моделями дані несуть семантичну інформацію про предметну область, вимагають методів і засобів для підтримки своєї цілісності та повинні розглядатися як важливий ресурс суспільства. Вирішення проблеми створення та використання інтенсивного виробничого потенціалу пов'язане з проблемою оптимального розподілу ресурсів (включаючи природні ресурси та інформацію) на основі застосування методів сучасної математики, програмного забезпечення та комп'ютерної техніки. Важливість єдиної концепції взаємопов'язаного розвитку між виробництвом, споживанням і охороною навколишнього середовища зумовлює необхідність розвитку інформаційної бази для вирішення складних завдань. Сьогодні цією основою є застосування концепцій баз даних в інформаційних системах і системах підтримки прийняття рішень, а перспективною є її реалізація на основі реляційної моделі.

Модель даних – це засіб абстракції, який забезпечує можливість інтерпретації даних, що заносяться або вилучаються з бази даних (БД), відповідно до уявлень користувачів про предметну область (ПрО) [1], яку відображає БД. У моделях даних виділяють три функціональні компоненти: структурну – визначення логічної структури даних; операційну – задання припустимих операцій над даним; обмежень цілісності – задає припустимі стани БД і окремих структурних одиниць даних, а також припустимі переходи між такими станами. Механізми кожної структурної компоненти (СКБД) конструюються на основі певної моделі даних, реалізуючи відповідну мову обробки даних. Мовні засоби СКБД зазвичай включають засоби визначення даних які реалізують структурний компонент і декларативні обмеження цілісності, та засоби маніпулювання даними, що реалізують операції й процедурні обмеження цілісності. У реляційній моделі даних (РМД) основною структурною одиницею даних є n -арне або n -місцеве відношення.

Поняття відношення добре відомо в класичній математиці, однак у контексті моделей даних воно має ряд особливостей [2].

Формально структурний компонент РМД задається четвіркою:

$$SRM = \langle R, D, A, dom \rangle, \quad (1)$$

де R – множина імен відношень; D – множина доменів; A – множина імен атрибутів; dom – відображення з A в D .

Кожний елемент SRM, де вихідними компонентами для структурної специфікації в РМД є домени, імена атрибутів і відношення. Кожний елемент D_i множини D називається доменом. Домен – це іменованій набір однорідних атомарних значень. Загалом, вибір домену визначається уявленнями користувача бази даних про домен, що моделюється. Наприклад, це може бути набір прізвищ, набір дат, набір номерів телефонів тощо. Область визначення також може бути нескінченною множиною. У сучасних реляційних СУБД домени часто вказуються за допомогою вбудованих типів даних, таких як NULL, NUMERIC, SYMBOL, STRING тощо [3]. Для будь-якого елемента A_i множини A визначена множина значень атрибута, що збігається з одним з доменів. Таким чином, в РМД задається відображення $dom: A \rightarrow D$ та пари $\langle A_i, dom A_i \rangle$, яка названа атрибутом з іменем A_i і областю значень dom_i . При цьому кожен атрибут відображається в один і тільки один домен. Атрибути – це визначені властивості об'єктів (сутностей і явищ) у домені з певними значеннями для кожного об'єкта. Таким чином, кожен атрибут має ім'я та приймає своє значення з певного домену. У СКБД відображення атрибутів домену вказується при описі структури зв'язку. Користувачі самостійно призначають імена атрибутам, виходячи зі своїх інформаційних потреб.

Нехай, наприклад, у моделі визначений домен YEAR = {1990, ..., 2000, ...}, і відображення dom зіставляє атрибуту $\langle Year-Date \rangle$ домен $\langle PİK \rangle$; тоді областю значень dom (Рік-Дата) атрибута є множина значень {1990, 1991, ..., 2000, ...}. Вираз $R_i(A_i, \dots, A_n)$ у структурній специфікації РМД називається носієм відношення. Кожній схемі відношення R_i модель зіставляє множину кортежів декартового добутку $r_i^* = dom A_1 \cdot \dots \cdot dom A_n$.

Відношенням r_i зі схемою R_i у РМД називається всяка кінцева підмножина кортежів множини r_i^* . Таким чином, відношення в реляційному підході розглядається як реалізація схеми. У кожному момент часу існування БД у ній міститься не більш, ніж одна реалізація схеми. Число n атрибутів відношення називається його ступенем (« n -арність»), а число кортежів – потужністю. Зрозуміло, що ступінь відношення (при незмінній схемі) не змінюється від реалізації до реалізації, тоді як потужності відношень, відповідних різним реалізаціям однієї й тієї ж схеми, можуть бути різними [4].

Відношення реляційної БД являють собою однотипні об'єкти ПрО. Наприклад, множини об'єктів типу EMPLOYEE може відповідати однойменне відношення, тобто множина кортежів (наборів значень атрибутів типу EMPLOYEE). Структурна специфікація (схема) відношення в РМД виглядає як вираз вигляду:

$$\text{ім'я-відношення (ім'я-атрибута-1, \dots, ім'я-атрибута-n)}. \quad (2)$$

Всі імена атрибутів у відношенні різні. Як тільки для кожного атрибута в РМД заданий його домен, для специфікованої схеми можна визначити множину

різних екземплярів реалізацій відношень з цією схемою. Відношення реляційної БД – це кінцева множина n -містних кортежів, утворених значеннями атрибутів, перелічених у схемі. Нехай, наприклад, у РМД задані домени $SURNAME = \{\text{Іванов, Петров}\}$ і $YEAR = \{1990, 1999\}$, та визначено, що атрибуту $\langle Surname_Employee \rangle$ відповідає домен $\langle SURNAME \rangle$, а атрибутам $\langle Year_Data \rangle$ й $\langle Year_hiring \rangle$ відповідає домен $\langle YEAR \rangle$. Тоді для схеми

$$EMPLOYEE(Surname, Year_Data, Year_hiring) \quad (3)$$

є можливість сформувати кортежі:

$\langle \text{Іванов, 1990, 1990} \rangle$, $\langle \text{Іванов, 1990, 1999} \rangle$,
 $\langle \text{Іванов, 1999, 1990} \rangle$, $\langle \text{Іванов, 1999, 1999} \rangle$,
 $\langle \text{Петров, 1990, 1990} \rangle$, ..., $\langle \text{Петров, 1999, 1999} \rangle$.

Вибираючи із цієї множини кожен підмножину навіть якщо вона має значення NULL, можна сформувати конкретні екземпляри відношення. Наприклад, відношенням є множина кортежів $\{\langle \text{Іванов, 1990, 1999} \rangle, \langle \text{Петров, 1990, 1999} \rangle\}$ або одноелементна множина $\{\langle \text{Іванов, 1999, 1990} \rangle\}$.

Таким чином, кожен кортеж із заданою схемою є екземпляром об'єкта моделювання відповідного типу. Кожен елемент кортежу є значенням певного атрибута, який представляє значення відповідного атрибута об'єкта. Ці значення можуть не відповідати «змісту» ПрО, як, у нашому випадку, коли дата народження EMPLOYEE виявляється більше року прийому на роботу.

Зауважимо, що в БД одночасно зберігається лише один екземпляр відношення із заданою схемою. Відношення може бути надано прямокутною таблицею, рядки якої відповідають кортежам відношення, а стовпці – його атрибутам. У таблиці зазвичай виділяється «заголовок» і «тіло». У заголовку таблиці розміщуються імена атрибутів, у тілі – значення атрибутів, тобто власне дані [5]. Нехай, наприклад, задана схема відношення:

$$EMPLOYEE(Surname, Year_Data, Year_hiring). \quad (4)$$

Таблиця 1 – Таблиця EMPLOYEE

Surname	Year_Data	Year_hiring
Іванов	1988	2004
Петренко	1989	2004
...

Як впливає із визначення відношення, імена атрибутів у таблиці унікальні. Порядок їх розташування, не має значення. Формально це означає, що порядок атрибутів у схемі не породжує нової схеми та не призводить до утворення нової її реалізації. У цьому і полягає найбільш суттєва відмінність відношення у РМД від відношень у їхньому математичному розумінні. Аналогічно, зміна порядку розташування рядків у таблиці не призводить до утворення нового відношення. Інакше кажучи, поняття впорядкованості кортежів у РМД відсутнє [6]. Схемою реляційної бази даних називається кінцевий набір схем відношень

$$B = \{R_1, \dots, R_p\}. \quad (5)$$

Відповідно, реляційна база даних (РБД) зі схемою B – це множина реалізацій схем відношення $\{R_1, \dots, R_p\}$:

$$b = \{r_1, \dots, r_p\}. \quad (6)$$

Усяка операція над даними в моделях даних «супроводжується» операцією над типами, тобто над описом структури операндів. Зокрема, у РМД операндами операцій є відношення, задані структурним компонентом моделі, результатом також є відношення; операції визначені при виконанні певних угод відносно типів операндів і результату. У той же час, над типами, тобто схемами відношення, можна провадити свої специфічні операції. Деякі з таких спеціальних операцій будуть розглянуті нижче. Об'єднання (union). Існують R і S – відношення одного типу. Тоді результатом об'єднання

$$R \text{ union } S \quad (7)$$

є множина кортежів, кожен з яких належить R або S . Тип результату дорівнює типу операндів. У випадку розбіжності імен атрибутів відношень-операндів для іменування атрибутів результату використовується лівий операнд. Різниця (difference)- коли прийняті ті ж угоди та позначення, що й вище. Тоді результатом операції

$$R \text{ difference } S \quad (8)$$

є множина кортежів, що належать R і не належать S . Операція не комутативна.

Декартів добуток (product). Існує R і S – відношення (у загальному випадку) різних типів. Визначимо операцію $R||S$ конкатенації на схемах $R(A_1, \dots, A_m)$ і $S(B_1, \dots, B_n)$, результатом якої є нова схема вигляду

$$T(R .A_1, \dots, R .A_m, S .B_1, \dots, S .B_n). \quad (9)$$

Введемо тепер операцію конкатенації на кортежах. Нехай $r = \langle a_1, \dots, a_m \rangle$ і $s = \langle b_1, \dots, b_n \rangle$ – два кортежі. Тоді $r||s$ – це кортеж виду $\langle a_1, \dots, a_m, b_1, \dots, b_n \rangle$. Нарешті, визначимо декартів добуток двох відношень

$$R \text{ product } S \quad (10)$$

як множина кортежів, утворених конкатенацією різних пар кортежів, перший з яких належить відношенню R , а другий – відношенню S . Тип результату визначається як конкатенація вихідних типів. Селекція (where), або обмеження – це одномісна операція, позначувана як $R \text{ where } F$, яка виділяє з відношення R тільки ті кортежі, що задовольняють деякій умові F . Де F – логічний вираз (формула), який будується за наступними правилами. У найпростішому випадку формула має вигляд

$$A \text{ theta } B, \quad (11)$$

де θ – оператор арифметичного порівняння з множини $\{=, <, <=, >, >=\}$, A і B – операнди, в якості яких виступають імена атрибутів відношення або константи. Операнди повинні бути сумісними за типом даних і ці типи даних повинні відповідати заданому операторові порівняння. У загальному випадку формула будується з найпростіших з використанням логічних зв'язок НЕ (NOT), І (AND), АБО (OR) за звичайними правилами алгебри логіки.

Отже, формула – це функція-вираз, яка приймає одне з двох значень – «істина» або «ні» (хибність) для кожного набору дозволених значень своїх аргументів (операндів порівняння). Кортеж відношення задовольняє формулу, якщо, підставляючи значення його атрибутів замість імен цих атрибутів у формулу, формула буде оцінюватися як істина. Важливо підкреслити різницю в «технології» їх використання в системах БД. З цієї точки зору слід розрізняти операції відбору даних, завдяки яким формується відповідь на запит користувача, і операції відновлення бази даних, завдяки яким формується новий стан бази даних. Наприклад, операції з'єднання в мовах обробки інформації можна використовувати для створення «тимчасового» зв'язку, який відповідає на запит на вибірку, або для додавання нових записів до існуючої таблиці бази даних.

У висновку підкреслимо важливість унікальної концепції розвитку інформаційної бази для вирішення складних проблем сучасності. Сьогодні цією основою є застосування концепцій баз даних в інформаційних системах і системах підтримки прийняття рішень. Ці методи забезпечують ефективну організацію даних і доступ до них. Завдяки обмеженням цілісності, індексу та транзакцій реляційні бази даних забезпечують цілісність даних і безпеку в багатьох програмах. Вважаємо, реляційну модель перспективною для реалізації.

Список літератури

1. Chen, Peter Pin-Shan. 1976. "The Entity-Relationship Model---toward a Unified View of Data." *ACM Transactions on Database Systems* 1 (1): 9–36. <https://doi.org/10.1145/320434.320440>.
2. Rajesh Bordawekar, and Oded Shmueli. "Using Word Embedding to Enable Semantic Queries in Relational Databases." ResearchGate. unknown, May 14, 2017. DOI:10.1145/3076246.3076251
3. Data types. IBM Documentation Help. Ibm.com. 2024. [Online]: <https://www.ibm.com/docs/en/i/7.3?topic=elements-data-types>
4. Pliekhova G, Alisejko O, Kochuieva Z. Application of semantic models and criteria equivalence of data to increase efficiency functioning of economic systems. *Vehicle and electronics Innovative technologies*. 2021 May 1;(19):41–6. doi: 10.30977/VEIT.2021.19.0.41
5. Li N., Bai L. Transforming fuzzy spatiotemporal data from relational databases to XML // *IEEE Access*. 2018. Т. 6. С. 4176–4185. DOI: 10.1109/ACCESS.2018.2790427
6. Гайдаржи В. І. Бази даних в інформаційних системах: підручник / В.І. Гайдаржи [та ін.]. – Київ: Україна, 2018. – 418 с.