

МНОЖИНИ ТА ВІДНОШЕННЯ ЯК ОСНОВА РЕЛЯЦІЙНИХ БАЗ ДАНИХ

Вінницький національний технічний університет

Анотація

У доповіді показано, що реляційні бази даних ґрунтуються на поняттях множин і відношень. Таблиця розглядається як множина кортежів, а зв'язки між таблицями – як відношення між множинами. Операції SQL інтерпретуються як реалізація операцій над множинами, що забезпечує узгодженість даних. Нормалізація подається як наслідок математичної коректності моделі.

Ключові слова: множина, відношення, реляційна модель, кортеж, домен, SQL.

Abstract

The report shows that relational databases are based on set theory and relations. A table is viewed as a set of tuples, and links between tables as relations. SQL operations correspond to set operations, ensuring data consistency. Normalization is presented as a consequence of mathematical correctness.

Keywords: set, relation, relational model, tuple, domain, SQL.

Вступ

Реляційні бази даних є фундаментальним інструментом сучасних інформаційних систем, забезпечуючи впорядковане зберігання, цілісність та логічну узгодженість даних. Їхня ефективність пояснюється тим, що реляційна модель ґрунтується на формальних поняттях дискретної математики: множинах, відношеннях та операціях над ними. У моделі, запропонованій Е. Ф. Коддом, таблиця розглядається не як технічний контейнер, а як математична структура – множина впорядкованих елементів. Саме ця властивість дозволяє описати поведінку реляційних БД незалежно від конкретних мов програмування чи способів фізичного зберігання даних.

Множина як модель організації даних

У реляційній моделі таблиця розглядається як множина впорядкованих записів, де кожен запис описує один об'єкт предметної області. Такий запис подається у вигляді впорядкованого набору атрибутів ($a_1, a_2, \dots, a_n$) і саме ця впорядкованість дозволяє інтерпретувати його як єдиний елемент множини. Кожен атрибут має власний допустимий набір значень, тобто належить певній множині можливих значень. Таку множину допустимих значень для конкретного атрибута називають доменом. Іншими словами, якщо a_i – це значення деякого поля запису, то воно береться не довільно, а лише зі свого домену D_i , який визначає припустимі значення та забезпечує цілісність таблиці.

Таким чином, таблиця подається як множина

$$R = \{(a_1, a_2, \dots, a_n) \mid a_i \in D_i\},$$

тобто як набір впорядкованих елементів, де кожен елемент належить строго заданим множинам значень. Саме така інтерпретація робить можливим подальше використання математичних операцій над множинами в реляційних базах даних та гарантує єдині правила взаємодії з даними.

Відношення між множинами як основа зв'язків

У дискретній математиці відношення визначається як підмножина декартового добутку двох множин:

$$R \subseteq A \times B.$$

У реляційних базах ця конструкція реалізується через первинні та зовнішні ключі, що формують відповідності між множинами таблиць. Наприклад:

$$F : \text{Student} \rightarrow \text{Group}, f(s) = g$$

інтерпретує належність студента до групи як відображення між множинами. Типи цього відображення (1:1, 1:N, M:N) визначають структуру зв'язків, а також поведінку каскадних оновлень і контролю цілісності.

Таким чином, структура реляційної БД не виглядає як довільний набір таблиць: це сукупність множин, зв'язаних відношеннями, де математичні властивості відповідностей визначають логіку роботи системи.

Операції над множинами та їх відповідники у SQL

Якщо таблиці реляційної бази даних розглядати як множини кортежів, а зв'язки між ними – як відношення між множинами, то операції обробки даних у SQL логічно постають не як довільні інструкції програмування, а як реалізація базових операцій над множинами. Тому багато дій, які виконуються в запитах, мають прямі математичні аналоги. Це забезпечує передбачуваність поведінки даних та пояснює, чому SQL працює саме так, а не інакше.

У таблиці 1 подано зіставлення поширених операцій над множинами з їхніми SQL-відповідниками.

Таблиця 1 – Операції між множинами

Операція множин	SQL-еквівалент	Суть
проекція	SELECT	підмножина атрибутів
підмножина	WHERE	елементи, що задовольняють предикат
декартовий добуток	JOIN	поєднання кортежів двох множин
об'єднання $A \cup B$	UNION	формування нової множини
перетин $A \cap B$	INTERSECT	спільні кортежі
різниця $A \setminus B$	EXCEPT	елементи, властиві лише одній множині

Наприклад, оператор JOIN не є «особливим правилом SQL» – це реалізація декартового добутку множин з подальшим відбором кортежів, що задовольняють умову відношення між ними:

```
SELECT S.Name, G.Name  
FROM Student AS S  
JOIN Group AS G ON S.GroupID = G.ID;
```

Цей запит виконує декартовий добуток із подальшим відбором відповідних пар – повністю відповідно до визначення відношення між множинами.

Нормалізація як наслідок властивостей множин і відношень

Оскільки таблиця – це множина впорядкованих елементів, а зв'язки між таблицями – відношення між множинами, некоректне розташування атрибутів порушує математичну структуру і призводить до надлишковості або суперечливості даних. Саме тому нормалізація є не просто технічною рекомендацією, а наслідком вимог до математичної коректності моделі.

Некоректний варіант (порушення моделі множин):

Student(id, name, group, specialty)

Тут значення specialty залежить не від студента, а від групи, що створює транзитивну залежність $id \rightarrow group \rightarrow specialty$.

У такому разі множина не є структурно цілісною, оскільки один атрибут описує інший через проміжну відповідність.

Коректний варіант після нормалізації:

Student(id, name, GroupID)
Group(id, name, specialty)

Після такого розділення таблиці знову виступають множинами впорядкованих елементів, а зв'язки між ними – відношеннями між множинами без логічних суперечностей. Це відновлює математичну коректність структури та усуває транзитивну залежність.

Висновок

Реляційні бази даних функціонують завдяки формальній математичній основі, де таблиці подаються як множини кортежів, а зв'язки між ними – як відношення між множинами. Операції над множинами визначають механізми маніпуляції даними, що реалізуються у SQL, а нормалізація виступає наслідком вимог до математичної коректності структури. Це демонструє, що дискретна математика не є теоретичним доповненням до реляційних БД, а становить їхню концептуальну основу.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Codd E. F. A Relational Model of Data for Large Shared Data Banks // Communications of the ACM. – 1970. – Vol. 13, No. 6. – P. 377–387.
2. Date C. J. An Introduction to Database Systems. – 8th ed. – Boston: Addison-Wesley, 2003. – 1024 p.
3. Silberschatz A., Korth H. F., Sudarshan S. Database System Concepts. – 7th ed. – New York: McGraw-Hill, 2019. – 1376 p.
4. Halpin T., Morgan T. Information Modeling and Relational Databases. – 2nd ed. – Burlington: Morgan Kaufmann, 2008. – 976 p.
5. Войтович С. П., Яремчук О. В. Організація баз даних. – Львів: Новий Світ-2000, 2017. – 368 с.

Липкань Віктор Михайлович – студент групи 1KI-25МС, факультет інформаційних технологій та комп'ютерної інженерії, Вінницький національний технічний університет, м. Вінниця, e-mail: lipkan02@gmail.com.

Науковий керівник: **Добровольська Наталія Вікторівна** – кандидат педагогічних наук, доцент, кафедра обчислювальної техніки, Вінницький національний технічний університет, м. Вінниця

Viktor Mykhailovych Lypkan – student of group 1KI-25MS, Faculty of Information Technology and Computer Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: lipkan02@gmail.com.

Supervisor: **Dobrovol'ska Natalia V.** – Candidate of Pedagogical Sciences, Associate Professor, Department of Computer Engineering, Vinnytsia National Technical University, Vinnytsia