RELATIONSHIPS IN RELATIONAL DATABASES

Sayfieva Gulrukhsor Komil kizi,

Qarshi State Technical University,

Computer engineering student

Annotation. *In relational databases, relationships are used to link different tables based on common attributes. These relationships define how data in one table is associated with data in another, ensuring data integrity and enabling efficient querying and data manipulation. The three primary types of relationships in relational databases are one-to-one, one-to-many, and many-to-many. Each type of relationship is implemented using keys, such as primary keys and foreign keys, to establish these connections between tables. Understanding and properly implementing these relationships is crucial for database design, as it ensures that data is organized logically, minimizes redundancy, and maintains consistency across the system. This article explores the types of relationships in relational databases, their applications, and how they contribute to the effective management of data in various domains such as e-commerce, finance, and healthcare.*

Keywords: *Relational Databases, Data Relationships, One-to-One, One-to-Many, Many-to-Many, Primary Keys, Foreign Keys, Data Integrity, Database Design, E-Commerce, Data Management.*

Аннотация. *В реляционных базах данных отношения используются для связи различных таблиц на основе общих атрибутов. Эти отношения определяют, как данные в одной таблице связаны с данными в другой, обеспечивая целостность данных и позволяя эффективно выполнять запросы и манипулировать данными. Три основных типа отношений в реляционных базах данных — это «один к одному», «один ко многим» и «многие ко многим». Каждый тип отношений реализуется с использованием ключей, таких как первичные ключи и внешние ключи, для установления этих связей между таблицами. Понимание и правильная реализация этих отношений имеют* *решающее значение для проектирования базы данных, поскольку это гарантирует, что данные организованы логически, минимизирует избыточность и поддерживает согласованность во всей системе. В этой статье рассматриваются типы отношений в реляционных базах данных, их приложения и то, как они способствуют эффективному управлению данными в различных областях, таких как электронная коммерция, финансы и здравоохранение.*

Ключевые слова. *реляционные базы данных, связи данных, «один к одному», «один ко многим», «многие ко многим», первичные ключи, внешние ключи, целостность данных, проектирование баз данных, электронная коммерция, управление данными.*

In the world of database management, the relational database model remains one of the most popular and widely used systems for data storage and retrieval. One of the key features that define relational databases is the concept of relationships between tables. These relationships enable a structured and efficient way of organizing and managing large sets of data. By linking data stored in multiple tables, relational databases ensure data integrity, reduce redundancy, and enable complex queries. The understanding of relationships—such as one-to-one, one-to-many, and many-to-many—is essential for proper database design, data manipulation, and retrieval. This article will delve into the different types of relationships found in relational databases, how they are implemented using keys, and their practical applications in real-world scenarios.

Relational databases store data in tables (also known as relations), where each table consists of rows and columns. Each row represents a unique record, and each column represents an attribute of the data. These tables are related to each other through certain attributes, typically using keys. A key is an attribute or a set of attributes that uniquely identifies a record in a table.

The two main types of keys used to establish relationships are:

- **Primary Key (PK):** A primary key is a unique identifier for a record in a table. Each table can only have one primary key, which ensures that each record is distinct.
- **Foreign Key (FK):** A foreign key is an attribute in one table that links to the primary key in another table. The foreign key establishes a relationship between the two tables.

In relational database design, there are three main types of relationships that can be defined between tables:

A one-to-one relationship exists when a record in one table is associated with exactly one record in another table. This relationship is relatively rare and is usually used when one table holds data that is closely related to another, but needs to be stored separately for performance or organizational reasons.

Example:

Imagine a database for a library system. One table stores information about books, and another table stores additional details about a book's digital version (such as an eBook). Since each book has only one digital version, this would be a one-to-one relationship.

- **Table 1 (Books):** BookID (PK), Title, Author
- **Table 2 (eBook):** eBookID (PK), BookID (FK), FileSize, Format

In this case, the BookID in the eBook table is a foreign key that references the primary key in the Books table, creating a one-to-one relationship between the two tables.

When to Use One-to-One:

- To separate sensitive or optional data (such as user preferences or additional details).
- When different tables are required to store large or frequently updated data (improving performance and reducing table size).

A one-to-many relationship is one of the most common types of relationships in relational databases. It occurs when a single record in one table can be related to multiple records in another table. This type of relationship is implemented by having a foreign key in the "many" table that references the primary key in the "one" table.

Example:

Consider a database for an e-commerce platform. One table stores information about customers, and another table stores customer orders. A single customer can place many orders, but each order is associated with only one customer. Thus, there is a one-to-many relationship between the Customers table and the Orders table.

- **Table 1 (Customers):** CustomerID (PK), Name, Email
- **Table 2 (Orders):** OrderID (PK), CustomerID (FK), OrderDate, TotalAmount

In this example, the CustomerID in the Orders table is a foreign key that links to the primary key in the Customers table. A customer can have multiple orders, but each order can only belong to one customer.

When to Use One-to-Many:

- When one entity (e.g., customer, department) can have multiple associated entities (e.g., orders, employees).
- Ideal for most business and organizational data systems, where many records need to be linked to a central record (e.g., customers, employees, products).

A many-to-many relationship occurs when multiple records in one table can be associated with multiple records in another table. This type of relationship is implemented using a junction table, which contains foreign keys from both related tables.

Example:

In a database for a university, there might be a many-to-many relationship between students and courses, as students can enroll in multiple courses, and each course can have multiple students.

- **Table 1 (Students):** StudentID (PK), Name, Major
- **Table 2 (Courses):** CourseID (PK), CourseName, Instructor
- **Junction Table (Enrollments):** StudentID (FK), CourseID (FK), EnrollmentDate

In this case, the Enrollments table is the junction table that connects the Students table and the Courses table, creating a many-to-many relationship. Each student can be enrolled in many courses, and each course can have many students.

When to Use Many-to-Many:

- When an entity can be related to multiple entities in another table, and vice versa.
- Suitable for complex systems like educational platforms, social networks, and product catalogs where multiple associations are needed.

When designing relational databases, it is essential to follow normalization principles to ensure that relationships between tables are logically structured and data redundancy is minimized. The process of normalization involves organizing data to reduce duplication and ensure that each table focuses on a specific type of data. The four main normal forms (1NF, 2NF, 3NF, BCNF) guide the database designer to avoid data anomalies and ensure consistency.

- **1NF (First Normal Form):** Ensures that each column contains atomic values and there are no repeating groups of data.
- **2NF (Second Normal Form):** Eliminates partial dependency by ensuring that all non-key attributes are fully functionally dependent on the primary key.
- **3NF (Third Normal Form):** Removes transitive dependency by ensuring that non-key attributes are not dependent on other non-key attributes.
- **BCNF (Boyce-Codd Normal Form):** A stricter version of 3NF that resolves issues where a non-prime attribute depends on a candidate key.

By normalizing a database, designers ensure that the relationships between tables are accurate, efficient, and maintainable over time.

Relationships play a crucial role in ensuring data integrity in relational databases. By linking tables using primary and foreign keys, relational databases enforce referential integrity, meaning that the data in the foreign key column must correspond to a valid record in the referenced table. This prevents orphaned records and maintains the consistency of the database.

Additionally, the relationships between tables allow for more powerful and efficient querying. SQL (Structured Query Language) joins, such as INNER JOIN, LEFT JOIN, and RIGHT JOIN, leverage these relationships to combine data from multiple tables in a single query, making relational databases an ideal choice for complex data retrieval.

The practical application of relationships in relational databases is evident across various industries:

- **E-Commerce:** Managing customer orders, inventories, and payments with one-to-many relationships between customers and orders, and many-to-many relationships between products and orders.
- **Healthcare:** Relating patients to their medical records, treatments, and prescriptions through one-to-many and many-to-many relationships.
- **Education:** Linking students to courses, grades, and instructors using many-to-many relationships and junction tables.
- **Finance:** Managing transactions, accounts, and financial records with one-to-many relationships between accounts and transactions.

Understanding relationships in relational databases is essential for designing efficient, scalable, and maintainable systems. By carefully structuring relationships—whether one-to-one, one-to-many, or many-to-many—database designers ensure that data is logically organized, redundant data is minimized, and integrity is maintained. As databases continue to evolve and become more complex, the principles behind relational database relationships remain fundamental to effective data management. From e-commerce to healthcare and education, relational databases continue to provide the backbone for data storage and retrieval across a wide range of industries.

References:

1. Zarif o‘g‘li K. F. CREATING A TEST FOR SCHOOL EDUCATIONAL PROCESSES IN THE ISPRING SUITE PROGRAM //BOSHLANG ‘ICH SINFLARDA O ‘ZLASHTIRMOVCHILIKNI. – С. 84.

2. O‘G‘Li K. F. Z. CREATING A TEST FOR SCHOOL EDUCATIONAL PROCESSES IN THE ISPRING SUITE PROGRAM //Yosh mutaxassislar. – 2023. – T. 1. – №. 8. – С. 84-87.

3. Kaynarov F. Z. THEORETICAL FOUNDATIONS FOR THE CREATION OF ELECTRONIC TEXTBOOKS FOR DISTANCE EDUCATION //Экономика и социум. – 2024. – №. 2-2 (117). – С. 169-175.

4. Kaynarov F. APPLICATION OF MODERN INFORMATION TECHNOLOGIES IN MEDICINE //International Scientific and Practical Conference on Algorithms and Current Problems of Programming. – 2023.

5. Кайнаров Ф. З. ИННОВАЦИОННЫЕ МЕТОДЫ ПРЕПОДАВАНИЯ ПРИКЛАДНОЙ МАТЕМАТИКИ //Экономика и социум. – 2023. – №. 1-2 (104). – С. 619-622.

6. Daminova B. ACTIVATION OF COGNITIVE ACTIVITY AMONG STUDENTS IN TEACHING COMPUTER SCIENCE //CENTRAL ASIAN JOURNAL OF EDUCATION AND COMPUTER SCIENCES (CAJECS). – 2023. – Т. 2. – №. 1. – С. 68-71.

7. Esanovna D. B. Modern Teaching Aids and Technical Equipment in Modern Educational Institutions //International Journal of Innovative Analyses and Emerging Technology. – Т. 2. – №. 6.

8. Рахимов Н., Эсановна Б., Примкулов О. Ахборот тизимларида мантиқий хулосалаш самарадорлигини ошириш ёндашуви //International Scientific and Practical Conference on Algorithms and Current Problems of Programming. – 2023

9. Даминова Б. Э. СОДЕРЖАНИЕ ПРОФЕССИОНАЛЬНОГО ОБРАЗОВАНИЯ И ТЕНДЕНЦИИ ЕГО ИЗМЕНЕНИЯ ПОД ВЛИЯНИЕМ НОВЫХ СОЦИАЛЬНО-ЭКОНОМИЧЕСКИХ УСЛОВИЙ //Yosh mutaxassislar. – 2023. – Т. 1. – №. 8. – С. 72-77.

10. Кувандиков Ж., Даминова Б., Хафизадинов У. АВТОМАТЛАШТИРИЛГАН ЭЛЕКТРОН ТАЪЛИМ ТИЗИМИНИ ЛОЙИҲАЛАШДА ЎҚУВ ЖАРАЁНИНИ МОДЕЛЛАШТИРИШ //International

Scientific and Practical Conference on Algorithms and Current Problems of Programming. – 2023.

11. Даминова Б. Э. Сравнительный анализ состояния организации многоуровневых образовательных процессов //Экономика и социум. – 2023. – №. 1-2 (104). – С. 611-614.

12. Daminova B. Algorithm of education quality assessment system in secondary special education institution (on the example of guzor industrial technical college) //International Scientific and Practical Conference on Algorithms and Current Problems of Programming. – 2023.

13. Daminova B. FORMATION OF THE MANAGEMENT STRUCTURE OF EDUCATIONAL PROCESSES IN THE HIGHER EDUCATION SYSTEM //Science and innovation. – 2023. – Т. 2. – №. A6. – С. 317-325.

14. Даминова Б. Э., Якубов М. С. Развития познавательной и творческой активности слущателей //Международная конференция" Актуальные проблемы развития инфокоммуникаций и информационного общества. – 2012. – С. 26-27.06.

15. Якубов М., Даминова Б., Юсупова С. Формирование и повышение качества образования с помощью образовательных информационных технологий //International Scientific and Practical Conference on Algorithms and Current Problems of Programming.-2023.

16. Даминова Б. Э. и др. ОБРАБОТКА ВИДЕОМАТЕРИАЛОВ ПРИ РАЗРАБОТКЕ ОБРАЗОВАТЕЛЬНЫХ РЕСУРСОВ //Экономика и социум. – 2024. – №. 2-2. – С. 117.