Rachel Theis

Module 1.3 Assignment

CSD 310

10/27/24

**In the context of relational databases, what are relationships? Describe at least two, and provide an example of their use.**

Relationships in relational databases refer to the associations between different tables that exist in a single relational database (Zivkovic, 2024).

There are three main types of relationships in relational databases: one-to-one, one-to-many, and many-to-many (Zivkovic, 2024). Below is a brief explanation of each type.

One-to-one: This type of relationship has two tables that have only one record, and every primary key value relates to none or one record in the related table (Zivkovic, 2024). One-to-one relationships do not usually naturally occur in a relational database and are often forced from established business rules (Zivkovic, 2024).

An example of a one-to-one relationship is two tables, one with customer names and the other with customer contact information. Here, each entry corresponds only once (Zivkovic, 2024).

One-to-many: One-to-many relationships are the most common type of relationship found in a relational database (Zivkovic, 2024). In a one-to-many relationship, one table can have multiple corresponding relationships to the second table, but the second table can only have one corresponding relationship to the first (Zivkovic, 2024). For example, in a retail setting, a customer table and an order table have a one-to-many relationship (Zivkovic, 2024). A single customer can have many orders, but an order can only correspond to one customer.

Many-to-many: Many-to-many relationships are the most complex of these three relationships and refer to tables that both have multiple relationships (Zivkovic, 2024). In many-to-many relationships, a third junction table is necessary (Zivkovic, 2024). An example of a many-to-many relationship is a table of books and a table of authors. Here, one author can have written several books, and books can have multiple authors (Zivkovic, 2024).

**What are the advantages of relational databases? What are the advantages of NoSQL databases?**

Relational Databases:

**Standardized schema:** The standardized schema of SQL is rigid, which can be limiting in some cases. However, this standardization lends itself to data consistency, integrity, and compliance (Edwards, 2021).

**Larger user community:** The SQL language is mature and continues to be extremely relevant for databases. Developers and database users can leverage the large and knowledgeable community to collaborate, innovate, and find resources for challenging projects (Edwards, 2021).

**No code required:** SQL is a very user-friendly language and can be managed and queried through keywords (Edwards, 2021).

**ACID Compliance:** Due to SQLs rigid and standardized structure, SQL databases can be ACID compliant in situations where applications can have no room for error and data integrity is essential, like in transaction processing (Edwards, 2021).

NoSQL Databases:

**Continuous Availability:** In NoSQL databases, data is distributed across multiple data centers rather than having a single failure point (Edwards, 2021). This lends to database stability and ensures always-up databases.

**Query Speed:** Joins are not necessary for NoSQL databases because data is not normalized. This can make the database faster (Edwards, 2021).

**Agility:** Unlike relational databases, NoSQL databases are intended to flex to meet developer and project needs. NoSQL databases do not necessitate predefined schemas and even columns and rows (Edwards, 2021). The agility of NoSQL databases can minimize the amount of time spent in the planning stages and allow developers to begin the development stage sooner (Edwards, 2021).

**Low Cost**: NoSQL databases are more cost-effective to scale, since they are expanded horizontally rather than vertically, minimizing the need for expensive hardware upgrades (Edwards, 2021). This advantage makes NoSQL databases a good fit for cloud computing and rapidly growing data (Edwards, 2021).

**What are the disadvantages of relational databases? What are the disadvantages of NoSQL databases?**

Relational Databases:

**Hardware:** Hardware is a costly aspect of relational databases. Relational databases are normally scaled vertically, which requires additional investment in RAM, CPU, and SSD capabilities as the database is scaled (Edwards, 2021). Additionally, hardware can quickly become obsolete as new releases are pushed (Edwards, 2021). Maintenance and everyday operations can also be costly from a hardware perspective.

**Data Normalization:** The need for data normalization can slow down a database, especially in complex cases where elements like lookups and joins are involved (Edwards, 2021).

**Rigidity:** Rigidity can be considered both a pro and a con in relational databases. While rigidity promotes data integrity and consistency, it can also be limiting. Due to the rigidity, the planning process of relational databases is a substantial investment since the database will be inflexible and needs to be thoroughly planned upfront (Edwards, 2021).

**Resource-intensive Scaling:** While relational databases are scalable, doing so can be expensive and require a great deal of expertise (Edwards, 2021). SQL databases are typically vertically scaled, which requires additional hardware investments and is a time-consuming process (Edwards, 2021).

NoSQL Databases:

**No standardized language:** NoSQL databases do not have a standardized language, leading to a sharper learning curve for new developers (Edwards, 2021).

**Smaller user community**: NoSQL databases are less mature than relational databases, meaning that solving undocumented issues can be more challenging than in relational databases (Edwards, 2021).

**Inefficiency with complex queries**: Due to the flexible nature of NoSQL databases, making complex queries can be inefficient due to the lack of a standard interface (Edwards, 2021).

**Data retrieval inconsistencies:** Since NoSQL databases use distributed data, inconsistencies in data retrieval can be inconsistent. This is a disadvantage that disqualifies NoSQL databases from ACID compliance (Edwards, 2021).

**Identify at least two features of MySQL and two features of MongoDB, and describe what they are and how they are used.**

MySQL

Scalability: MySQL can only be scaled vertically, which prioritizes data integrity and consistency but is expensive and costly in terms of time (Deshpande, 2023).

Language: MySQL supports SQL (Deshpande, 2023)

Community Support: “There are about 222k repositories and 7 Million commits on GitHub for support on MySQL” (Deshpande, 2023).

MongoDB:

Scalability: MongoDB can be scaled horizontally or vertically, which can make scaling more cost and time-effective, especially in situations with rapidly growing datasets (Deshpande, 2023).

Community Support: “There are about 177k repositories and 923k commits on GitHub for support on MongoDB” (Deshpande, 2023).

Language: MongoDB supports JSON Query Language (Deshpande, 2023).

**References**

Deshpande, C. (2023, February 14). *MongoDB Vs. MySQL: Which One is Better*. Simplilearn.com. https://www.simplilearn.com/tutorials/mongodb-tutorial/mongodb-vs-mysql

Edwards, R. (2021, September 29). *SQL versus NoSQL: Pros and Cons*. SQL versus NoSQL: Pros and Cons | Datastax. https://www.datastax.com/blog/sql-vs-nosql-pros-cons

Zivkovic, M. (2024, February 12). *The Beginner’s Guide to Database Relationships | Luzmo*. Www.luzmo.com. https://www.luzmo.com/blog/database-relationships