Amanda Wedergren

March 22, 2025

Module 1.3 Assignment

There are a few types of relationships in a database. These include one-to-one relationships, one-to-many relationships, and many-to-many relationships. (GeeksforGeeks, 2025) In one-to-many relationships, the items from one table can be linked to only one or many items from another table. An example of this would be a database of customers and their placed orders. While each customer may place more than one order, the master customer record remains the same. In many-to-many relationships, multiple records belonging to a table is associated with multiple records from another table. A wide variety of complex relationships can be created where each item has many related items. An example of this could be a database for music streaming that contains tracks that belongs to multiple playlists and contain multiple tracks. A one-to-one relationship is not as common, but represents one item that corresponds to another single item.

Relational databases are important because they provide ways to efficiently retrieve data, keep its integrity, analyze complex data, and provide scalability to possible expansion. They are also secure, simple to use, and have fast performance. NoSQL, or non-relational databases, provide a simple way to store data from different sources that can be formatted differently. (Foote, 2023) This means that a larger variety of data types can be stored in a NoSQL database. NoSQL databases also have much more scalability compared to relational databases, increasing capacity and many are easy to modify for new forms of data. It is also very developer-friendly, and doesn’t require that data is adapted for storage purposes since it can contain native formats.

Relational and NoSQL databases also have disadvantages. Relational databases can be expensive to use, requiring a user to purchase specific software and/or tools to experience the full benefits. (Foote, 2023) They can also be limited in storage space, which can cause data loss or create problems in the future. In addition to the data sometimes being complex to understand, performance issues can also be of concern, taking more time to process the results. NoSQL databases also have their own disadvantages. They can be difficult to organize and make data untrustworthy. The lack of SQL (as the name suggests) means less mature technology and lack of standardization across multiple brands of NoSQL databases. Each unique system has strengths and weaknesses that must be learned before choosing the NoSQL database that would be best for the situation at hand, which can be incredibly time consuming and confusing.

MongoDB and MySQL are both popular database management systems built for different purposes and have their own unique features. While MondoDB is a NoSQL database designed for having high scalability and handling unstructured data, MySQL is a traditional relational database management system that is ideal for structured data that includes complex relationships. (GeeksforGeeks, 2025) MongoDB allows for flexibility, allowing data to be stored in any structure that can be adjusted over time without disrupting data that already exists. It is also highly scalable and supports complex indexing. MySQL has a predefined schema, in which data is structured and stored in tables that include rows and columns. It is also compliant with ACID (Atomicity, Consistency, Isolation, and Durability) properties for transactions, which helps to ensure data integrity.

References

Foote, K. D. (2023, March 27). *NoSQL databases: Advantages and disadvantages*. DATAVERSITY. https://www.dataversity.net/nosql-databases-advantages-and-disadvantages/

GeeksforGeeks. (2025a, January 24). *MongoDB vs mysql*. https://www.geeksforgeeks.org/mongodb-vs-mysql/

GeeksforGeeks. (2025b, February 12). *Types of relationship in database*. https://www.geeksforgeeks.org/types-of-relationship-in-database/