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

Relationships define how data in different tables connect to one another. The three main types are one-to-one, one-to-many, and many-to-many.

In a one-to-one relationship, each record in one table is associated with exactly one record in another. For example, a person can have only one Social Security Number, and that number belongs to only one person.

In a one-to-many relationship, one record in a table can be related to many records in another table. For example, many people can be born on the same date, but each person has only one birth date.

A many-to-many relationship allows multiple records in one table to be related to multiple records in another. This is common in scenarios like students enrolled in multiple classes, and each class having multiple students.

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

Relational databases provide strong data consistency and help reduce redundancy through normalization. The use of SQL enables powerful, structured querying, making them well-suited for complex transactions and data analysis.

NoSQL databases are highly flexible, allowing storage of both structured and unstructured data. They support Agile development and are designed for scalability, especially useful for handling large volumes of data with varied formats.

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

Relational databases are not ideal for handling unstructured data and can lack flexibility in evolving data models. They may also experience performance issues at scale, especially with large, distributed systems.

NoSQL databases, while flexible, lack a universal query language, which can make complex queries harder to manage. They may also sacrifice data consistency (depending on the type) in favor of performance and scalability.

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

**MySQL**:

* **ACID-compliant transactions**: These ensure data integrity during operations like bank transfers, where all steps must succeed or fail as a group.
* **Foreign key constraints**: Used to enforce relationships between tables, ensuring that data remains consistent and connected across the database.

**MongoDB**:

* **Document-oriented storage**: Stores data as flexible, JSON-like documents, making it easy to work with nested data structures.
* **Horizontal scalability (sharding)**: MongoDB can distribute data across multiple servers, allowing it to handle large-scale applications and heavy workloads efficiently.

Sources

IBM. (n.d.). *Relational databases*. IBM Think. Retrieved March 22, 2025, from <https://www.ibm.com/think/topics/relational-databases>

IBM. (n.d.). *MongoDB*. IBM Think. Retrieved March 22, 2025, from <https://www.ibm.com/think/topics/mongodb>

Couchbase. (2023, March 29). *Relational vs. non-relational databases: What’s the difference?* Couchbase Blog. Retrieved March 22, 2025, from <https://www.couchbase.com/blog/relational-vs-non-relational-database/>

Oracle. (n.d.). *MySQL 8.4 features*. MySQL Documentation. Retrieved March 22, 2025, from <https://dev.mysql.com/doc/refman/8.4/en/features.html>