**Relational vs Non-Relational Database**

1. In the context of relational databases, what are relationships?
   1. Relationships connect two different tables using keys. These relationships help keep data organized and normalized.
2. Describe at least two, and provide an example of their use:
   1. One-to-Many Relationship: This means one record in a table is connected to many records in another.

Example: One teacher can have many students. So, the teacher table links to the student table using the primary key on the teacher (ID)

**Select \* from teacher join students on teacher.teacher\_id = students.teacher\_id**

A diagram of a teacher

AI-generated content may be incorrect.

* 1. Many-to-Many Relationship: This means many records in one table are connected to many records in another. If several students shared books with classmates. You can view all the shared books between multiple students using the following command

**Select \* from students join books on books.student\_id = students.book\_id**

Or a particular book **Select \* from students inner join books on books.student\_id = students.book\_id where books.book\_id = 3**

A screenshot of a computer

AI-generated content may be incorrect.

1. Advantages of Relational Databases
   1. Relational Databases uses a structure of relationships between sets of data which allows you to organize information based on categories.
2. Disadvantages of Relational Databases
   1. It is difficult to manage large sets of data with a complex type of relationships. If students shared books with different graduation years and the books that they shared changed repeatedly between new and old books. If the chapters changed and information on the pages changed but you wanted to established the person that read a prior sentence. You can use NoSQL to manage the scaling of the relationships for support.
3. Advantages of NoSQL Databases
   1. When the relationship between data has scaled in a way where the relationship is unmanageable, you can use NoSQL to join information to handle large amounts of unstructured relationships.
4. Disadvantages of NoSQL Databases
   1. Normalization will be an issue when using NoSQL because there’s no defined way of handling large sets of relationships. Using the prior example, you can’t perform a complex query to view that type of relationships.
5. MySQL Description, Features, Usage:
   1. Description:
      1. MySQL is a relational database that stores data in tables with rows and columns unlike MongoDB. It uses SQL to manage and query information from the database.
   2. Features:
      1. ACID Compliance (Atomicity Consistency Isolation Durability): Most systems support ACID but this is a feature that’s reliable and naturally relatable with MySQL.
      2. MySQL supports join across multiple databases while MongoDB doesn’t have this built-in feature
   3. Usage:
      1. You can use MySQL for the main database on a website such as school database of information, inventory management, ordering website, and more.
6. MongoDB Features:
   1. Description:
      1. MongoDB is a NoSQL database that stores data in a JSON structured system with key-value pairs.
   2. Features:
      1. Flexibility: As stated in the article below, a flexible schema approach is valuable when your data model is subject to change. This allows you to dynamically update the key/value pairs overtime without having to redesign the database.
      2. <https://www.mongodb.com/resources/basics/unstructured-data/schemaless>
   3. Usage:
      1. You would use MongoDB for a particular feature on a website such as a suggestion box or user activity tracking. You wouldn’t use MongoDB for the main repository of information on a website.

References  
General information on SQL: <https://www.w3schools.com/sql/>

Querying: <https://www.w3schools.com/sql/sql_join.asp>

MongoDB: <https://www.mongodb.com/docs/manual/introduction/>

MySQL: <https://dev.mysql.com/doc/refman/8.0/en/>