1. In the context of relational databases, what are relationships?
2. What are the advantages of relational databases?
3. What are the disadvantages of relational databases?
4. Identify at least one feature of MySQL and describe what it is and how it is used.

In relational databases, a relation is a link or connection between an object in one table and an object in another table as ab attribute (Duca, 2021). These links are made clear and concise with the use of primary keys, which also helps with data normalization. Relationships allow for certain complex forms of queries that ask for a lot of data related to a certain object and can be described with different models, such as binary, recursive, or ternary relationships (*Working with ER diagrams*). For example, you could represent the relationship between an employee and their computer terminal as a binary one-to-one relationship because that computer is only ever used by that employee and that employee only ever uses that one computer.

Some advantages of using a relational database really depend on your use cases. However, if you have a well preconceived data schema relational databases may work well for you. They excel with structuring data in a highly organized manner due being run mainly on one single server. This structuring also makes complicated queries, which take advantage of the linked relationships between objects, much more potent. They allow you to get more data about an object from a single query, depending on your schema (Duca, 2021). Schemas will have to be analyzed before implementation for what type of querying will be done with the database, in order to prioritize what attributes and objects are tabled together for easy querying.

This preconceived schema definition is also a weakness in certain regards. Relational databases will take more time, thought, and planning to set up and initialize than a non-relational database because once you implement a data schema it is hard to change. That is to say, once a database has been populated, all that data adds a lot of inertia to changes in a schema, which may add too much overhead for the server to continue operating while the changes are being implemented (Fowler et al., 2012).

One thing that I noticed in MySQL that seems quite useful is stored procedures. Stored procedures are created with the CREATE command and can be viewed in the INFORMATION\_SCHEMA.ROUTINES table for reference. Stored procedures allow you to pre-compile queries, parts of queries, or multiple queries with logic statements, and conditions based on the state of attributes or tables (A.4 MySQL 8.0 FAQ: Stored procedures and functions). This is useful because it allows the database to quickly and efficiently process certain actions that have been defined at an earlier stage, or that may need to be done frequently, improving performance. Stored procedures also allow a level of code reusability and modularity by allowing developers to break down operations into smaller units of logic and store those units for later use.

Sources:

Duca, A. L. (2021, October 5). *Relational vs Non Relational databases*. Medium. https://towardsdatascience.com/relational-vs-non-relational-databases-f2ac792482e3

*Working with ER diagrams*. Studytonight.com. (n.d.). https://www.studytonight.com/dbms/er-diagram.php

Fowler, M., & Sadalage, P. J. (2012). 3.3 Schemaless Databases. In *NoSQL distilled* (pp. 62–65). essay, Addison-Wesley.

*MySQL 8.0 Reference Manual :: A.4 MySQL 8.0 FAQ: Stored procedures and functions*. MySQL. (n.d.). https://dev.mysql.com/doc/refman/8.0/en/faqs-stored-procs.html