**Botiwa F Konan**

**Module 1**

**Assignment 1-3**

**Relationships in Relational Databases**

In relational databases, relationships represent the connections between different tables. These relationships are typically defined through primary keys and foreign keys. A primary key is a unique identifier for a record in a table. In contrast, a foreign key is a field in one table that uniquely identifies a row of another table or the same table. For example, consider a database with two tables: **Customers** and **Orders**. The **Customers** table has a primary key **CustomerID**, and the **Orders** table has a foreign key **CustomerID** to link each order to a specific customer, establishing a one-to-many relationship.

**Advantages of Relational Databases**

Relational databases offer strong consistency and data integrity through ACID (Atomicity, Consistency, Isolation, Durability) compliance. They are excellent for complex queries involving multiple tables and support structured data and relationships efficiently. SQL provides powerful querying capabilities that allow for precise data manipulation and retrieval.

**Advantages of NoSQL Databases**

NoSQL databases provide flexibility in data models, accommodating unstructured and semi-structured data. They are designed to scale out horizontally, making them suitable for large-scale data and high-velocity applications. NoSQL databases often offer better performance for specific workloads, such as real-time data processing and large-scale distributed systems.

**Disadvantages of Relational Databases**

Relational databases can struggle with scalability as they primarily scale vertically. They might also face performance issues with very large datasets or high-velocity data streams due to the complexity of maintaining ACID properties. Schema rigidity can be a limitation when dealing with evolving data requirements.

**Disadvantages of NoSQL Databases**

NoSQL databases may lack the rigorous consistency models provided by relational databases, as many opt for eventual consistency. They often require more complex data modeling for relationships and might not support multi-table transactions natively. Additionally, the variety of NoSQL databases can lead to steep learning curves and integration challenges.

**Features of MySQL and MongoDB**

**MySQL:**

1. **ACID Compliance**: MySQL supports ACID transactions, ensuring reliable and secure transactions. This feature is crucial for applications requiring precise and reliable data management, such as banking systems.
2. **Replication**: MySQL offers master-slave replication, allowing data from one MySQL database server (master) to be replicated to one or more MySQL database servers (slaves). This is used for improving data availability and read performance.

**MongoDB:**

1. **Document-Oriented Storage**: MongoDB stores data in flexible, JSON-like documents. This allows for a more dynamic schema, making it easier to store and query complex hierarchical data structures.
2. **Horizontal Scalability**: MongoDB supports sharding, which distributes data across multiple servers. This feature is essential for handling large datasets and high-throughput applications by allowing horizontal scaling.

Understanding the strengths and weaknesses of both relational and NoSQL databases is crucial for selecting the right database technology to meet specific application requirements.