Rakshitha devi J

write about sql vs nosql database and features of mongodb.

SQL vs NoSQL Database:

Aspect	SQL (Relational)	NoSQL (Non-relational)
Data Model	Table-based (rows and columns)	Document, Key-Value, Column, or Graph
Schema	Fixed, predefined schema	Flexible, dynamic schema
Query Language	SQL (Structured Query Language)	Varies by database (e.g., JSON-style in MongoDB)
Scalability	Vertical scaling (scale-up)	Horizontal scaling (scale-out)
Transactions	Strong ACID compliance	Often BASE (Eventually consistent)
Joins	Supports complex joins	Limited or no joins
Best Suited For	Structured data, complex queries	Large volumes of semi/unstructured data
Examples	MySQL, PostgreSQL, Oracle, SQL Server	MongoDB, Cassandra, Redis, Couchbase

MongoDB?

MongoDB is an open-source, **NoSQL** (Not Only SQL) database designed for scalability, performance, and ease of use. It stores data in a **document-based** format, typically in **JSON-like structures** called **BSON** (Binary JSON). Unlike traditional relational databases that use tables and rows, MongoDB uses **collections** and **documents**.

- **Collections**: Equivalent to tables in SQL databases, collections are groups of documents.
- **Documents**: Equivalent to rows in SQL databases, but in MongoDB, documents are flexible and can contain any number of fields with varying data types.

MongoDB is designed to handle **unstructured data** and **large volumes of data** that don't fit easily into the rigid structure of relational databases. This flexibility makes it a popular choice for developers working with modern, distributed, and real-time applications.

feature of mongodb:

Schema-less: MongoDB is **schema-less**, meaning that documents within a collection can have different structures. This makes it ideal for storing unstructured or semi-structured data, such as JSON data from web services, social media posts, or sensor data.

Example:

```
{"name": "Alice",
  "age": 28,
  "email": "alice@example.com"}
```

Scalability: MongoDB is designed to scale horizontally by using **sharding**, which distributes data across multiple servers. This makes it suitable for applications with high write loads and large datasets.

Replication: MongoDB supports **replication**, ensuring data availability and redundancy. Data is replicated across multiple servers (nodes), which provides high availability and fault tolerance.

Flexible Data Model: MongoDB uses a **document-oriented model** with a JSON-like format, making it more intuitive to store and manipulate data, especially when dealing with hierarchical or nested data.

Indexing: MongoDB supports powerful indexing, including single field, compound, and geospatial indexes, which significantly improve query performance.

Aggregation Framework: MongoDB offers an **aggregation framework** for performing operations like filtering, sorting, grouping, and transforming data. This allows for more complex queries and computations within the database.