the key differences between MongoDB and SQL (relational databases) in five slides:

1. Data Model

MongoDB: MongoDB is a NoSQL database that follows a document-oriented data model. It stores data in flexible, semi-structured documents, typically in JSON-like format, allowing for dynamic schema and nested data structures.

SQL: SQL databases are based on a relational data model. They organize data into tables with predefined columns and enforce a fixed schema.

1. : Scalability

MongoDB: MongoDB is designed to scale horizontally, meaning it can distribute data across multiple servers or clusters, providing high scalability for large-scale applications with high read/write demands.

SQL: Traditional SQL databases scale vertically, where you need to upgrade the hardware to handle increasing workload. However, some SQL databases also provide options for horizontal scaling, although it may require additional setup and configuration.

1. : Query Language

MongoDB: MongoDB uses a query language called MongoDB Query Language (MQL), which is a flexible and expressive language for querying document-based data. It supports a wide range of query operators and allows for complex filtering and aggregation operations.

SQL: SQL databases use Structured Query Language (SQL), a standardized language for managing and querying relational databases. SQL offers powerful querying capabilities, including joins, filtering, sorting, and aggregate functions.

1. : Schema Flexibility

MongoDB: MongoDB offers schema flexibility as it does not enforce a strict schema. Documents within a collection can have different structures, allowing for easier adaptation to changing data requirements.

SQL: SQL databases have a fixed schema, meaning the table structure and data types must be defined in advance. Any changes to the schema may require altering the table structure or performing data migrations.

1. Transactions and ACID Compliance

MongoDB: MongoDB introduced multi-document ACID transactions in recent versions, allowing for atomicity, consistency, isolation, and durability across multiple document operations within a single transaction.

SQL: SQL databases have a long history of supporting ACID transactions, ensuring data integrity and consistency during complex operations involving multiple tables.