Slide 1: Data Model

MongoDB:

Document-oriented database

Schema-less design allows flexibility

Documents stored in collections

SQL:

Relational database management system (RDBMS)

Tables with rows and columns

Defined schema enforces structure

Slide 2: Scalability

MongoDB:

Designed for horizontal scalability

Sharding enables distribution of data across multiple servers

Better suited for handling large volumes of unstructured data

SQL:

Vertical scalability is common

Scaling can be challenging with increasing data volume and transaction load

Primarily suited for structured data and traditional ACID transactions

Slide 3: Query Language

MongoDB:

Queries expressed using JSON-like syntax

Rich query language with support for CRUD operations, aggregation, and geospatial queries

SQL:

Queries written in SQL (Structured Query Language)

Standardized language for managing relational databases

Supports complex joins, subqueries, and transaction management

Slide 4: Use Cases

MongoDB:

Well-suited for applications requiring flexible schemas and fast development cycles

Ideal for content management systems, real-time analytics, and mobile applications

SQL:

Preferred for applications with structured data and complex transactions

Commonly used in traditional enterprise systems, financial applications, and legacy systems

Slide 5: Conclusion

Summary of key differences between MongoDB and SQL databases

Considerations for choosing between NoSQL and SQL based on project requirements

Emphasize the importance of understanding the data model, scalability needs, and query requirements

This presentation provides a high-level comparison between MongoDB and SQL databases, highlighting their respective strengths and best use cases.