noSQL vs SQL

Slide 1: Introduction

Title: MongoDB vs SQL

- MongoDB: A NoSQL Database
- SQL: A Relational Database Management System

Slide 2: MongoDB

- Type: NoSQL
- Structure: Document-oriented database
- Schéma: Dynamic schema (Schema-less)
- Query Language: MongoDB Query Language (MQL)
- Scalability: Horizontally scalable
- Flexibility: Supports unstructured and semi-structured data
- Use Cases: Big Data, Content Management Systems, Real-time Analytics

Slide 3: SQL

- Type: Relational Database Management System (RDBMS)
- Structure: Table-based database
- Schéma: Statique schema (Schema-on-write)
- Query Language: Structured Query Language (SQL)
- Scalability: Vertically scalable
- Data Integrity: Enforces ACID properties (Atomicity, Consistency, Isolation, Durability)
- Use Cases: Traditional business applications, Financial systems, E-commerce

Slide 4: MongoDB Functionality

- Flexible Schema: Allows for easy modification of data structure without downtime
- Horizontal Scalability: Scales out across multiple servers for high availability and performance
- High Performance: Optimized for read and write operations, especially for large-scale applications
- Geospatial Indexing: Built-in support for geospatial queries and indexes
- JSON-like Documents: Data stored in BSON format, which is similar to JSON, making it easy to work with for developers

Slide 5: SQL Functionality

- ACID Compliance: Guarantees data integrity and consistency with transaction support
- Structured Data: Organizes data into tables with defined relationships, ensuring data integrity and consistency
- Data Integrity Constraints: Supports constraints like primary keys, foreign keys, and unique constraints
- Complex Queries: Allows for complex queries involving joins, subqueries, and aggregations
- Mature Ecosystem: Well-established tools and frameworks, with extensive support and documentation