07 Document Databases and MongoDB

History of MongoDB

MongoDB was founded in 2007 by engineers from DoubleClick (acquired by Google) who realized relational databases were not efficient for handling massive-scale data, such as serving 400,000+ ads per second. The name "MongoDB" was derived from "Humongous Database." MongoDB Atlas, the fully managed cloud version, was introduced in 2016.

MongoDB Data Structure

MongoDB is structured hierarchically:

- Database contain multiple collections.
- Collection group of related documents (analogous to tables in relational databases).
- Document is a single JSON-like object stored within a collection.

Relational vs. MongoDB Comparison

In relational databases, a database is a collection of related tables, whereas in MongoDB, a database contains multiple collections.

A table or view in a relational database is equivalent to a collection in MongoDB.

A row in a relational database corresponds to a document in MongoDB, where each document is stored in JSON or BSON format.

A column in a relational database is similar to a field in MongoDB, representing an individual data attribute.

An index functions the same in both relational databases and MongoDB, optimizing query performance.

A foreign key in relational databases is used to establish relationships between tables, whereas in MongoDB, references between documents can be created using embedded documents or manual references.

Instead of performing joins like in relational databases, MongoDB uses embedded documents, where related data is stored within a single document for faster retrieval.

MongoDB Features

- Rich Query Support: Supports all CRUD (Create, Read, Update, Delete) operations.
- Indexing: Supports primary and secondary indices for faster querying.
- Replication: Supports replica sets for high availability with automatic failover.
- Load Balancing: Built-in support for distributing load across multiple nodes.

MongoDB Deployment Options

- MongoDB Atlas: Fully managed cloud-based MongoDB (DBaaS).
- MongoDB Enterprise: Subscription-based, self-managed version with enterprise features.
- MongoDB Community: Open-source, free-to-use, self-managed version.

Interacting with MongoDB

MongoDB Tools

- mongosh (MongoDB Shell): CLI tool for interacting with MongoDB instances.
- MongoDB Compass: Free, open-source GUI tool for managing MongoDB databases.
- DataGrip & Third-Party Tools: Additional GUI-based database management tools.
- MongoDB Libraries for Programming Languages:
    - PyMongo (Python)
    - Mongoose (JavaScript/Node.js)
    - Other libraries for Java, C#, Go, etc.

Running MongoDB in Docker

- Create a container
- Map host port to container port 2701.
- Set an initial username and password for the superuser

Using MongoDB Compass

MongoDB Compass provides a visual interface for working with MongoDB. It allows users to connect to a database, create collections, and run queries.