Introduction To MongoDB

What is MongoDB

- MongoDB is a document oriented No-SQL general-purpose database designed with flexibility for high volume data storage.
- Because MongoDB takes the best of relational and NoSQL databases, providing a technology foundation that enables customers to meet the demands of almost any class of modern application
- MongoDB stores the data in the form of collections and documents
  - Collection → Table of RDBMS
  - Document → row of RDBMS

What is NoSQL Database

- NoSQL stands for “Not Only SQL”
- NoSQL database are non tabular and doesn’t follow RDBMS standards
- They provide flexible schemas and scale easily with high volumes of data.
- A NoSQL database is used to store large quantities of complex and diverse data, such as product catalogs, logs, user interactions, analytics, and more Designed for scalability and high availability.
- NoSQL databases emerged in the late 2000s as the cost of storage dramatically decreased.
- MongoDB is one of the most established NoSQL databases
- NoSQL databases were invented to solve the problem of storing unstructured and semi-structured data.
- NoSQL databases allow you to store the data without a schema and also support dynamic schema, which decouples the clients from a rigid schema, and is often necessary for modern and experimental applications.

Difference between RDBMS & NoSQL Database

Feature	Relational Database	NoSQL Database
Schema	A relational database follows a rigid schema. The database tables should have a definition of all the desired columns and their types. Any data manipulation that deviates from the schema generates an error.	A NoSQL database does not impose a rigid schema and allows you to store the unstructured data with dynamic structures. This allows an evolving database structure.
Data model/ Storage Structure	The data is stored in tables. Each record is stored as a row that contains information about all the columns. Changing a table can affect the other tables and applications.	The data is stored in different formats, depending on the provider. The standard storage structures are documents, graphs, key-values, and wide columns. There is no alteration required as the database adapts to the dynamic nature of data and the application works seamlessly.
Normalization	Normalization is the process used to remove duplicate data and avoid data anomalies. A relational database prevents data anomalies, using normalization. It requires the data to be stored in different tables and to create relationships among them.	NoSQL databases focus more on fast data retrieval, and the data can be denormalized.
Scaling	Scaling is the ability of a database to grow or reduce in size, depending on the need. Relational databases are hard to scale and are generally scaled vertically, which means increasing the machine compute and storage.	NoSQL databases provide both vertical and horizontal scaling. In horizontal scaling, the data can be distributed across different machines/ clusters.

What is MongoDB

- MongoDB is a document oriented database
- MongpDB is “no-sql” database
- It’s a general purpose “no-sql” database
- Designed for scalability and high availability
- Follows schema-less architecture (Which means a dynamic schema)
- MongoDB works on on-premise , on cloud and mobile devices

Collections and Documents

- MongoDB stores the data in the form of collections and documents.
- Documents are the basis of data storage and it stores the data in the form of "key-value" pair .
- There is no pre-defined schema for documents, which enables the developer to add or remove fields as required.
- for eg: two documents in a single collection can have different fields
- Collections consiste of set of documents and certian built in fuctions like "count" , which will help to get the count of documents in a collection.
- MongoDB allows to create the collection "on the fly", which means the devloper can create the collection along with data insert statement
- MongoDB supports Capped Collection(Fixed size) and non capped collections

Where MongoDB can be used

- E-commerce product catalog.
- Blogs and content management.
- Real-time analytics and high-speed logging, caching, and high scalability.
- Configuration management.
- Maintaining location-based data — Geospatial data.
- Mobile and social networking sites.
- Evolving data requirements.
- Loosely coupled objectives — the design may change by over time.

MongoDB Common Use cases

- Many of its most successful use cases center around the following areas:
- IoT
- Mobile applications
- Real-time analytics
- Personalization
- Catalog management
- Content management

Some companies who uses mongDB

- IBM
- Citrix
- Twitter
- T-Mobile
- Zendesk
- Sony

Benefits of Using MongoDB

- Document oriented
- High performance
- High availability — Replication
- High scalability – Sharding
- Dynamic — No rigid schema.
- Flexible – field addition/deletion have less or no impact on the application
- Heterogeneous Data
- Capped Collections
- No Joins
- Distributed
- Data Representation in JSON or BSON
- Geospatial support
- Document-based query language that's nearly as powerful as SQL
- Cloud distributions such as AWS, Microsoft, RedHat,dotCloud and SoftLayer etc:-. In fact, MongoDB is built for the cloud. Its native scale-out architecture, enabled by 'sharding,' aligns well with the horizontal scaling and agility afforded by cloud computing.

Editions of MongoDB

- MongoDB Community Edition (Free for Production use)
- MongoDB Enterprise Advanced Server (Paid License)
- MongoDB Atlas (For Cloud)
- MongoDB Realm for mobile applications
- Upgrading from community edition to enterprise Edition possible

Difference Between Community and Enterprise Edition

- MongoDB Enterprise provides various features not available in the MongoDB Community edition, such as:
    - In-Memory Storage Engine
    - Auditing
    - Kerberos Authentication
    - LDAP Proxy Authentication and LDAP Authorization
    - Encryption at Rest

MongoDB Atlas

- Is A database as service offering available on AWS , Azue or GPC
- Infrastructure provisioning, setup, and deployment is fully automated with MongoDB Atlas.
- Just select a cloud provider, region, instance size, memory, and additional configurations in the Cluster Builder or via the API.
- Free tier is available for testing

MongoDB deployment Models

- Standalone
  - Standalone installation is a single-machine installation and is meant mainly for development or experimental purposes.
- Replica set
  - A replica set in MongoDB is a group of processes or servers that work together to provide data redundancy and high availability. It requires at least three servers in a cluster. These servers are configured as the primary, secondaries, or arbiters.
- Sharded
  - Sharded deployments allow you to store the data in a distributed way. A shard contains a subset of the data, and each shard must use a replica set to provide redundancy of the data that it holds. Multiple shards working together provide a distributed and replicated dataset.

MongoDB Versioing

- MongoDB versioning has the form X.Y.Z where X.Y refers to either a release series or development series and Z refers to the revision/patch number.
- If Y is even, X.Y refers to a release series; for example, 4.0 release series and 4.2 release series. Release series are **stable** and suitable for production.
- If Y is odd, X.Y refers to a development series; for example, 4.1 development series and 4.3 development series. Development series are **for testing only and not for production**.
- For example, in MongoDB version 4.0.12, 4.0 refers to the release series and .12 refers to the revision.
-

MongoDB Versioing

New Releases

- Changes in the release series (e.g. 4.0 to 4.2) generally mark the introduction of new features that may break backwards compatibility.

Patch Releases

- Changes to the revision number (e.g. 4.0.11 to 4.0.12) generally mark the release of bug fixes and backwards-compatible changes.

Taking an example

- MongoDB 4.4 --> MongoDB stable release 4 of version 4(New feature releases ..sometime contains non backward compatible changes)
- MongoDB 4.4.12 --> MongoDB stable release 4 of version 4 with revision number 14(mostly bug fixes. The changes are backward compatible)

Course Design

What all covered in this course

- This course is designed for MongoDB database administration, meaning more emphasis is given for database administration topics than development activities
- Installation and configuration of MongoDB on Linux
- Different options in starting up mongodb server
- Database creation and management
- Creation of collection
- CRUD Operations

What all covered in this course

- Securing MongoDb
- Authorization and authentication
- User & Roles Management
- Backup and Recovery
- Replication Configuration
- Sharding Configuartion