Machireddy Dhamini

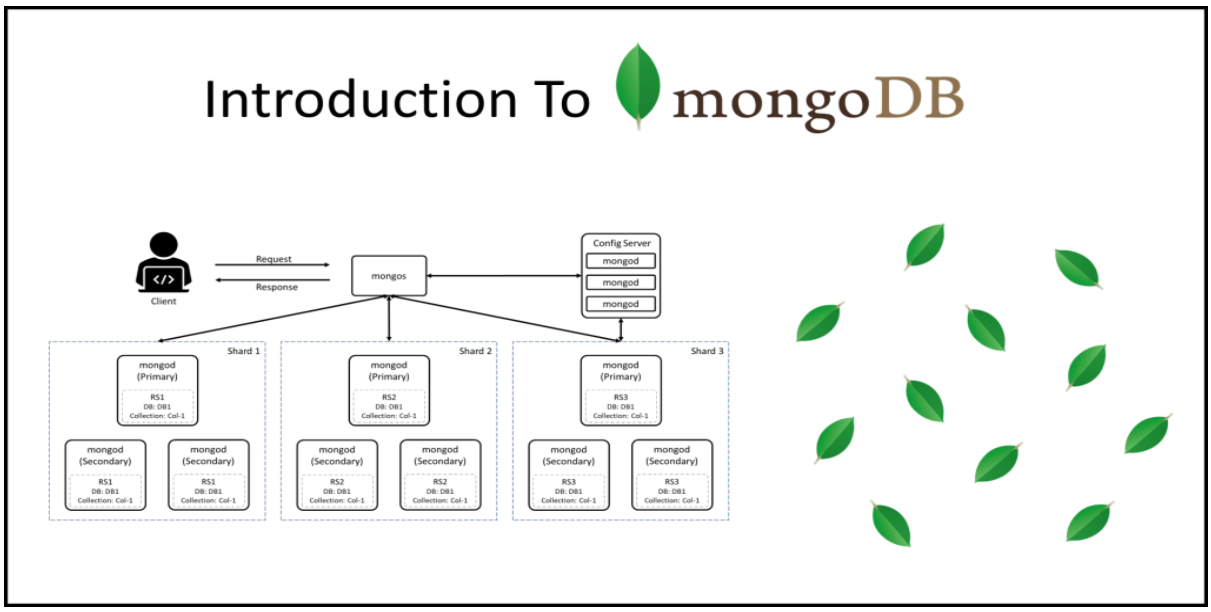
MONGODB (NOSQL)

MongoDB and the Importance of NoSQL in Data Engineering

Introduction to MongoDB

What is MongoDB?

MongoDB is a leading, open-source NoSQL database system built to handle high volumes of diverse data types. Unlike traditional relational databases, MongoDB uses a flexible data model based on BSON (Binary JSON), making it ideal for agile development environments and rapidly changing data structures. It enables developers to store nested arrays and documents directly within a record, offering a more intuitive structure for many applications.



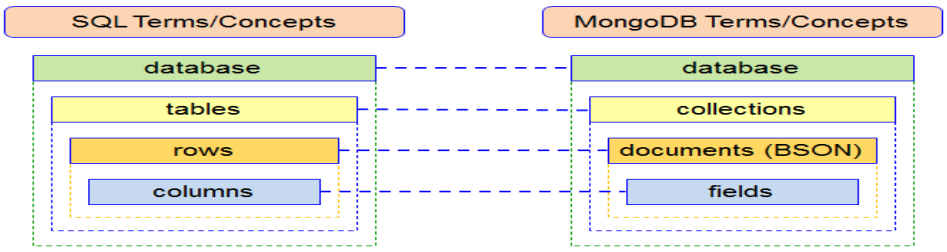
Key Features of MongoDB:

- Document-Oriented: Each record (document) contains all related data in a single, nested structure.
- Schema-less Design: No fixed schema is required, allowing you to store different fields in different documents.
- Horizontal Scalability: Built-in sharding allows distribution of data across multiple servers.
- High Availability: Replica sets ensure redundancy and automatic failover.
- Powerful Query Language: Rich queries, indexing, and aggregation support.
- Cloud Integration: MongoDB Atlas offers seamless integration with cloud environments.

Core Concepts of MongoDB

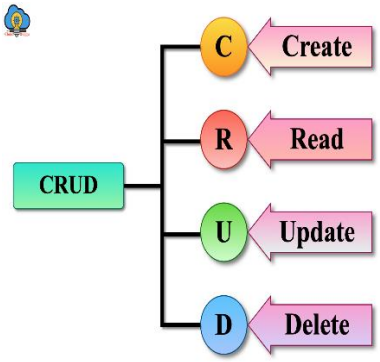
1. Collections and Documents

- Documents: The smallest unit of data, stored in BSON format. Fields can include arrays, other documents, and even deeply nested objects.
- Collections: Analogous to tables in RDBMS. A collection holds multiple documents without enforcing a schema.



2. CRUD Operations

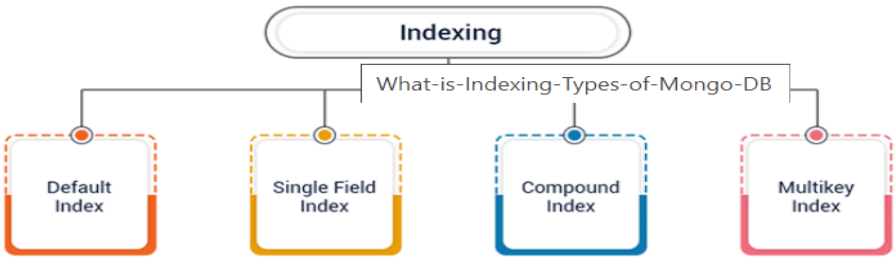
MongoDB is the **Document-Based** distributed database system that provides rich functionality to read and write the data and these operations are achieved through MongoDB **CRUD**. The MongoDB **CRUD** stands for **CREATE, READ, UPDATE and DELETE** which are used to perform the operation such as create, update, read and delete on MongoDB Documents. Using these operations we can easily interact with the database and perform various tasks.



3. Indexing

Indexes speed up query execution. MongoDB supports:

- **Single Field**: Speeds up queries on one field.
- **Compound**: Optimizes queries using multiple fields.
- **Multikey**: Indexes array fields for efficient array queries.
- **Geospatial**: Enables location-based queries on coordinate data.



4. Aggregation Framework

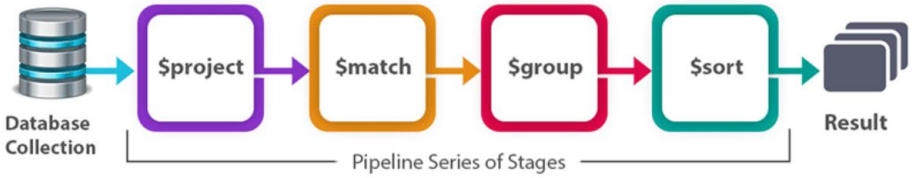
Aggregation pipelines process documents in stages. It's used for transforming and computing aggregated values using $match, $group, $sort, etc.

Example:

[{$match: {status: "active"}}, {$group: {_id: "$category", total: {$sum: 1}}}]

5. Sharding and Replication

- Sharding: Enables horizontal scaling by partitioning data across shards.
- Replication: Replica sets replicate data across nodes to ensure high availability.



Why NoSQL is Essential in Data Engineering

1. Rise of Big Data

The explosion of data from mobile apps, IoT devices, logs, and social media has led to the need for databases that can handle volume, velocity, and variety. NoSQL databases like MongoDB are purpose-built to meet these demands.

2. Flexible Schema Design

NoSQL allows changes to data structures without major migrations. This is crucial for iterative development and fast deployment.

3. Scalability and Performance

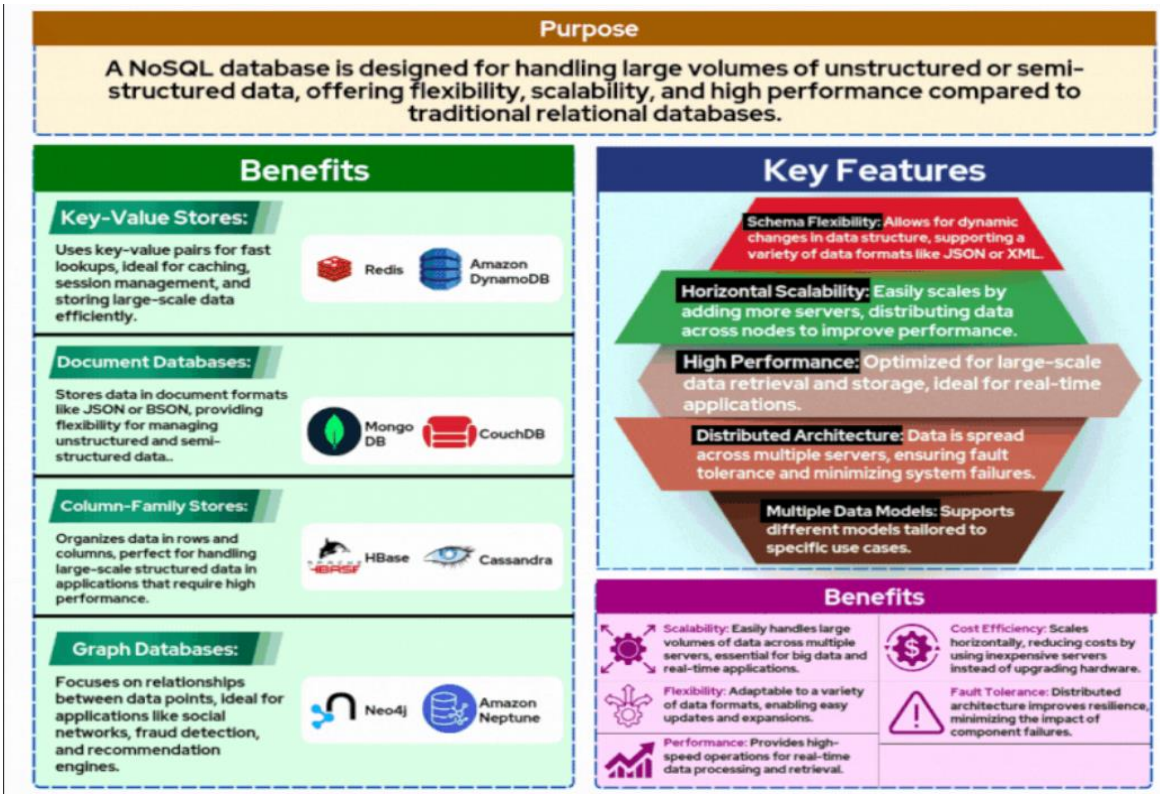
NoSQL databases can horizontally scale across commodity servers, ensuring high throughput and low latency even under large workloads.

4. Data Variety and Complexity

NoSQL databases can store structured, semi-structured (JSON, XML), and unstructured data (images, videos, logs), which are commonly encountered in data engineering pipelines.

5. Cloud-Native Architecture

Designed for distributed computing, NoSQL systems integrate well with cloud platforms and containers, supporting DevOps and microservices architectures.



When to Choose NoSQL:

- Rapid prototyping
- High volume/low latency requirements
- Diverse and changing data types
- Global-scale apps requiring sharding and replication

MongoDB in the Data Engineering Ecosystem

1. ETL Pipelines

MongoDB plays a critical role in modern ETL workflows:

- Extract: Collects data from sources like APIs, Kafka, or logs.
- Transform: Uses aggregation or tools like Apache Beam/Spark.

- Load: Inserts clean data into MongoDB collections.

2. Real-Time Data Processing
MongoDB supports real-time ingestion and querying:

- Seamless integration with Kafka, Spark Streaming
- Native change streams for event-driven apps

3. Integration with Data Lakes
MongoDB can complement data lakes by:

- Acting as a source for batch processing
- Storing cleansed, enriched data

4. Data Modeling Best Practices

- Embed: When data is accessed together often.
- Reference: For large, independent objects.
- Avoid deep nesting: Max document size is 16MB.

5. Monitoring and Security

- MongoDB Atlas provides automated backups, real-time alerts, and analytics.
- Role-based access control (RBAC), TLS encryption, and auditing are supported.

Conclusion

MongoDB and NoSQL databases are redefining how data engineering is done. They bring agility, scalability, and adaptability to the forefront, making them indispensable tools for data engineers tackling modern data challenge

Machireddy Dhamini

MONGODB (NOSQL)