MongoDB Fundamentals: Core Concepts

1.MongoDB

MongoDB is a NoSQL database management system that uses a document-oriented data model. It is designed to provide scalability, flexibility, and performance for modern applications. MongoDB stores data in flexible, JSON-like documents, making it easy to represent complex data structures and relationships.

2.Documents

It is the basic unit of data and data is stored as key value pairs. Documents are stored in format called BSON and it can be considered as analog of row in relational database

3.Collections

A collection in MongoDB is a group of documents and it can be considered as analog to a table in RDBMS

4.Database

Containers for collection. Analog to Database in RDBMS

5.Schema less or Flexible Schema

Refers to the ability to store documents in collection without enforcing a predefined structure or schema. Documents within the same collection can have different set of fields with different datatypes from one document to another

6.Sharding

Technique used for distributing data across multiple machines or servers to improve scalability and performance.Data is partitioned into smaller chunks called "shards," and each shard is stored on a separate server or replica set.
Shard Key: Field or combination of fields used to partition data across shards
Mongos: routing processes that act as proxies for client connections to the sharded cluster
Config Server: These store metadata about the sharded cluster, including information about which data is stored on each shard and the shard key ranges.

7.Replica set

Group of mongoDB servers that maintain the same dataset/identical copies providing data redundancy,high availability and fault tolerance.It consists of a primary node for writes and secondary nodes for data replication.
PRIMARY node is the only active replica set member that receives write operations from database clients. The PRIMARY node saves data changes in the Oplog(operation logs). Changes saved in the Oplog are sequential ie, saved in the order that they are received and executed.
The SECONDARY nodes is querying the PRIMARY nodes for new changes in the Oplog. If there are any changes, then Oplog entries are copied from PRIMARY to SECONDARY as soon as they are created on the PRIMARY node

8.Read and Write Concern

Settings that determine the level of acknowledgment required from MongoDB servers for read and write operations.
Write Concern
{ w: 1 } :Requires acknowledgment from the primary node that the write operation has been received.
{ w: "majority" } :Requires acknowledgment from a majority of replica set members
{w: 0} : Do not require acknowledgement
{ wtimeout: <milliseconds> } :
Write concerns can also specify a timeout value to control how long MongoDB waits for acknowledgment before timing out.

Read concern
{ readConcern: "local" } : Allows reading data from the local node without waiting for replication.
{ readConcern: "majority" } :Ensures that the data read reflects the latest majority-committed data.
{ readConcern: "available" } : Return data from the most recent replica set member that is available, regardless of whether that data has been replicated to a majority of replica set members.
{ readConcern: "snapshot"} : Provides a higher level of consistency by ensuring that the read operation returns data from a snapshot of a majority-committed state
{ readConcern: "linearizable" } :Provides linearizable consistency.

9.Indexing

Indexes are data structures that improve the performance of read operations by facilitating efficient data retrieval. They allow MongoDB to quickly locate documents in a collection based on the values of one or more fields.Without indexes, MongoDB must scan every document in a collection to return query results
Types of Indexes
Single Field: A single field index indexes the values of a single field within documents in a collection.
Compound Index: Indexes multiple fields within documents in a collection.
Multikey Index: Indexes the elements of an array field within documents in a collection.
Geospatial Index: Indexes spatial data for geospatial queries.
Text Index: Indexes the text content of string fields for full-text search.
Hashed Index: A hashed index hashes the values of a single field using a hash function.

10.Capped collections

They are fixed-size collections that insert and retrieve documents based on insertion order. Capped collections work similarly to circular buffers ie , Once a capped collection reaches its maximum size, it automatically overwrites the oldest documents with new ones.

11.TTL indexes

TTL(Time To Live) indexes are special single-field indexes that MongoDB can use to automatically remove documents from a collection after a certain amount of time.They provide a mechanism for removing data from MongoDB collections based on the value of a designated "expireAfterSeconds" field.
Eg:It can be used for storing machine generated event data, logs, and session information

12.GridFS

GridFS is a specification for storing and retrieving large files in MongoDB such as images, videos, and binary data.

GridFS breaks down large files into smaller "chunks" of a fixed size (typically 255KB by default) and stores each chunk as a separate BSON document in a MongoDB collection

13.Journaling

It is a feature that provides durability in the event of failure by recording operations in a write ahead log[WAL] or Journal.MongoDB writes operations to a journal file (also known as the write-ahead log or WAL) before making changes to the database data files on disk.

14.Mongodump and Mongorestore

They are command line tools provided by mongoDB for creating and restoring binary backups of mongoDB databases.
mongodump:Create binary backups of MongoDB databases.It connects to a running MongoDB instance and reads data from the databases, creating a binary dump of the data in BSON format.
mongorestore : Restore binary backups created by mongodump back into MongoDB databases

15.Storage Engines

The storage engine is the component of the database that is responsible for managing how data is stored, both in memory and on disk.
WiredTiger Storage Engine: It is a modern, high-performance storage engine designed to provide improved concurrency, compression, and scalability compared to the previous MMAPv1 storage engine. It provides a document-level concurrency model, checkpointing, and compression, among other features.
In-Memory Storage Engine : An In-Memory storage engine is available in MongoDB Enterprise. Rather than storing documents on-disk, it retains them in-memory for more predictable data latencies.

16.Role Based Access Control [RBAC]

MongoDB employs Role-Based Access Control (RBAC) to govern access to a MongoDB system.
Read,readWrite,dbAdmin,dbOwner

17.CRUD Operations

CRUD stands for Create, Read, Update, and Delete, which are the basic operations for managing data in a database.
Create: Inserting new documents into a collection.
Read: Querying and retrieving documents from a collection.
Update: Modifying existing documents in a collection.
Delete: Removing documents from a collection.

18.Aggregation pipelines

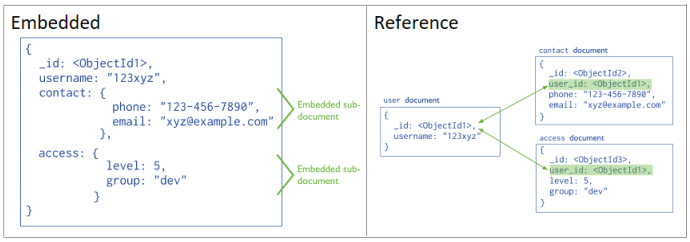
Aggregation Pipeline is a robust framework for performing data transformations and aggregation operations on collections.An aggregation pipeline consists of multiple stages, with each stage representing a transformation operation on the input documents. MongoDB provides a variety of pipeline stages, such as $match, $group, $sort, $project, $unwind, $lookup, etc.,

19.Lookup

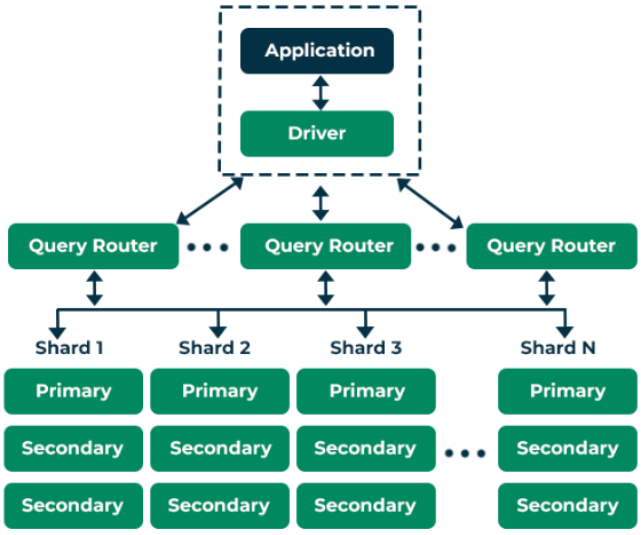
The $lookup allows you to perform a left outer join between documents from two collections based on a common field.For each document in the "local" collection, MongoDB adds a new field containing an array of matching documents from the "foreign" collection.

20.Data Modeling concepts

Embedded and References are two different approaches for modeling relationship between documents in a collection.
Embedded: Embed related data into a single document.
This approach is suitable for one-to-one and one-to-many relationships where the related data is relatively small and does not change frequently.
Embedded data models are often denormalized.
References: References store relationships between data by including links, called references, from one document to another. References result in normalized data models because data is divided into multiple collections and not duplicated.



21.MongoDB Architecture



Reference: Geeks for Geeks

The architecture of MongoDB involves various components working together to provide scalable, high-performance storage and access to data.
Application : The application is any software system or service that interacts with MongoDB to store, retrieve, and manipulate data.

Drivers : Drivers are client libraries that offer interfaces and methods for applications to communicate with MongoDB databases. Drivers will handle the translation of documents between BSON objects and mapping application structures.
Storage Engines: The storage engine is responsible for managing data storage and access within MongoDB.
MongoDB supports multiple storage engines, including WiredTiger (default), MMAPv1, and others. It handles data compression, indexing, concurrency control, and other low-level data management tasks.
Query Routers: In a sharded MongoDB cluster, query routers (mongos processes) act as intermediaries between applications and the shards.Query routers receive client requests and route them to the appropriate shard based on the shard key specified in the query.
Shards: Shards are mongod instances (MongoDB server processes) distributed across multiple servers. Each shard contains a portion of the data based on the shard key, which determines how data is partitioned and distributed across shards.
Primary and Secondary Nodes: The primary node accepts write operations from applications and replicates data to secondary nodes asynchronously.
Secondary nodes replicate data from the primary node and serve read operations.