MongoDB Interview Q&A; (50 Questions)

Prepared Set for MERN Stack Interviews

Basic MongoDB Questions (15)

Q: What is MongoDB and how is it different from SQL databases?

A: MongoDB is a NoSQL database that stores data in JSON-like documents, unlike SQL databases which store data in tables with fixed schemas.

Q: What is a document in MongoDB?

A: A document is a single record in MongoDB, stored in BSON format.

Q: What is a collection in MongoDB?

A: A collection is a group of MongoDB documents, equivalent to a table in SQL.

Q: Difference between MongoDB and MySQL?

A: MongoDB is schema-less and document-based, MySQL is relational and table-based.

Q: What are BSON documents?

A: BSON is a binary representation of JSON-like documents in MongoDB.

Q: How do you insert data into MongoDB?

A: Using insertOne() for single documents and insertMany() for multiple documents.

Q: What is ObjectId in MongoDB?

A: A unique identifier automatically generated for each document.

Q: Difference between find() and findOne()?

A: find() returns a cursor for multiple documents, findOne() returns a single document.

Q: What are CRUD operations in MongoDB?

A: Create, Read, Update, and Delete operations performed on documents.

Q: What are data types supported in MongoDB?

A: String, Number, Boolean, Date, Array, Object, Null, etc.

Q: How do you update a document in MongoDB?

A: Using updateOne(), updateMany(), or replaceOne() methods.

Q: What is the use of $set operator in update queries?

A: To update specific fields without replacing the entire document.

Q: Difference between embedded documents and referenced documents?

A: Embedded documents store related data inside the same document, referenced documents store references to other documents.

Q: How do you delete a document in MongoDB?

A: Using deleteOne() or deleteMany() methods.

Q: What is the use of limit() and skip() methods?

A: limit() restricts number of results, skip() ignores a number of results from the start.

Hard MongoDB Questions (15)

Q: What are indexes in MongoDB and why are they important?

A: Indexes improve query performance by allowing MongoDB to find documents efficiently.

Q: Difference between single field and compound indexes?

A: Single field indexes index one field, compound indexes index multiple fields.

Q: What is the difference between clustered and non-clustered indexes in MongoDB?

A: MongoDB uses non-clustered indexes; clustered indexes store data sorted by the index key.

Q: How does MongoDB handle relationships?

A: Through embedding documents or referencing documents using ObjectIds.

Q: What is the difference between $lookup and manual joins?

A: $lookup performs server-side joins, manual joins are handled in application code.

Q: Explain the Aggregation Framework in MongoDB.

A: A framework to process data and return computed results using pipelines.

Q: Difference between aggregate() and mapReduce()?

A: aggregate() is more efficient and commonly used; mapReduce() is flexible but slower.

Q: What are capped collections?

A: Fixed-size collections that maintain insertion order and automatically overwrite oldest entries.

Q: What is the difference between updateOne() and updateMany()?

A: updateOne updates the first matching document, updateMany updates all matching documents.

Q: How does MongoDB ensure high availability?

A: Through replica sets with automatic failover.

Q: What is replication in MongoDB?

A: The process of synchronizing data across multiple servers for redundancy.

Q: What is oplog in replication?

A: A special capped collection in the primary that logs all operations for secondaries.

Q: Difference between primary and secondary replica sets?

A: Primary handles writes, secondaries replicate data and can serve reads.

Q: What is write concern in MongoDB?

A: Specifies the level of acknowledgment requested from MongoDB for write operations.

Q: What is read preference in MongoDB?

A: Determines which replica set members to read from (primary, secondary, nearest).

Ultra-Advanced MongoDB Questions (20)

Q: What is sharding in MongoDB?
A: A method to distribute data across multiple servers to scale horizontally.

Q: Difference between horizontal and vertical scaling in MongoDB?
A: Horizontal = adding more servers, vertical = adding resources to a single server.

Q: How does MongoDB distribute data across shards?
A: By using a shard key to divide data into chunks.

Q: What is a config server in sharding?
A: Stores metadata and configuration for a sharded cluster.

Q: Difference between hash-based sharding and range-based sharding?
A: Hash-based distributes data randomly, range-based distributes by key ranges.

Q: What are chunk migrations in MongoDB sharding?
A: Moving chunks of data between shards to balance load.

Q: How does MongoDB handle failover in replica sets?
A: Automatic election of a new primary if current primary fails.

Q: What is journaling in MongoDB?
A: A write-ahead log that ensures data durability.

Q: What are transactions in MongoDB and when to use them?
A: Multi-document ACID operations, used when consistency across documents is needed.

Q: Difference between ACID in MongoDB vs relational databases?
A: MongoDB provides ACID at document level; multi-document ACID with transactions.

Q: How does MongoDB optimize queries internally?
A: Through indexes, query planner, and execution statistics.

Q: What are covered queries in MongoDB?
A: Queries where all fields are in an index, avoiding document fetch.

Q: What is schema design best practice in MongoDB?
A: Design for query patterns, embed related data, avoid joins where possible.

Q: Difference between denormalization and normalization in MongoDB?
A: Denormalization = embedding data; Normalization = referencing data.

Q: How do you monitor and optimize MongoDB performance?
A: Using explain(), profiler, monitoring tools, and indexing strategies.

Q: What is the role of aggregation pipelines in analytics?
A: To transform and compute aggregated results efficiently.

Q: Explain the difference between sharded cluster and replica set.
A: Sharded cluster distributes data; replica set provides redundancy.

Q: What is the impact of large collections on performance?
A: Large collections require proper indexing and shard design to maintain performance.

Q: How does MongoDB handle concurrency and locking?
A: It uses document-level locking to allow high concurrency.

Q: What are some common pitfalls in MongoDB schema design?
A: Over-embedding, large documents, unindexed queries, poor shard key selection.