MongoDB

---

1. Overview

MongoDB is a **NoSQL document-oriented database** designed for high performance, high availability, and easy scalability.

- Stores data as **JSON-like BSON documents** (Binary JSON).
- Schema-less: flexible, allowing different fields and data types in each document.
- Developed by MongoDB Inc., first released in 2009.
- Ideal for modern applications needing flexible data models and fast iteration.

---

2. Key Concepts

Document-Oriented Storage

- Data is stored as **documents**, which are similar to JSON objects.
- A **collection** is a group of documents (similar to a table in relational databases).
- Documents contain **key-value pairs** and can nest complex objects and arrays.

Example document in a users collection:

```
{
  "_id": ObjectId("507f191e810c19729de860ea"),
  "name": "Alice",
  "age": 28,
  "email": "alice@example.com",
  "address": {
    "street": "123 Maple St",
    "city": "Springfield"
  },
  "hobbies": ["reading", "hiking"]
}
```

BSON (Binary JSON)

- MongoDB stores data in BSON, a binary format optimized for speed.
- Supports additional data types not in standard JSON, like Date, Binary, ObjectId.

---

3. Architecture

Components:

- **mongod**: The core database process that handles data storage, queries, and networking.
- **mongos**: Router process used in sharded clusters.
- **Replica Set**: A group of mongod instances that maintain the same data set for redundancy and failover.
- **Sharding**: Horizontal scaling by partitioning data across multiple servers (shards).

---

4. CRUD Operations

Create (Insert)

```
db.users.insertOne({ name: "Alice", age: 28 });

db.users.insertMany([{ name: "Bob" }, { name: "Charlie" }]);
```

Read (Query)

```
db.users.find({ age: { $gt: 25 } });

db.users.findOne({ name: "Alice" });
```

Update

```
db.users.updateOne(
 { name: "Alice" },
 { $set: { age: 29 } }
);


db.users.updateMany(
 { age: { $lt: 30 } },
```

{ $inc: { age: 1 } }

);

Delete

db.users.deleteOne({ name: "Alice" });

db.users.deleteMany({ age: { $lt: 20 } });

---

5. Query Operators

- **Comparison:** $eq, $ne, $gt, $lt, $gte, $lte, $in, $nin
- **Logical:** $and, $or, $not, $nor
- **Element:** $exists, $type
- **Evaluation:** $regex, $text, $where
- **Update:** $set, $unset, $inc, $push, $pull, $addToSet

---

6. Indexing

- Speeds up query performance.
- Types of indexes: Single field, compound, multikey (for arrays), text, geospatial.
- Example:

db.users.createIndex({ email: 1 }); // Ascending index on email

---

7. Replication

- **Replica Sets** maintain multiple copies of data.
- Provides **high availability** and **automatic failover**.
- Primary node handles writes; secondaries replicate data.
- Example: 3-node replica set with 1 primary, 2 secondaries.

---

8. Sharding

- Distributes data across multiple machines.
- Enables horizontal scaling for large datasets.

- Uses a **shard key** to determine data distribution.
- Managed by mongos routers.

---

9. Data Modeling

- Favor embedding related data for fast reads (denormalization).
- Use referencing for relationships with large or changing data sets.
- Example:
    - Embed address inside user document.
    - Reference orders with user IDs.

---

10. Aggregation Framework

- Powerful data processing pipeline.
- Performs operations like filtering, grouping, sorting, reshaping data.
- Common stages: $match, $group, $project, $sort, $limit

Example: Count users by city

```
db.users.aggregate([
 { $group: { _id: "$address.city", count: { $sum: 1 } } }
]);
```

---

11. Security

- Authentication: SCRAM, LDAP, x.509 certificates.
- Authorization: Role-Based Access Control (RBAC).
- Encryption: TLS for data in transit, encryption at rest.
- Auditing and IP whitelisting.

---

12. Tools & Ecosystem

- **Mongo Shell (mongosh)**: Interactive command-line tool.
- **Compass**: GUI for querying and visualizing data.

- **Drivers**: Official drivers for Node.js, Python, Java, C#, etc.
- **Atlas**: Fully-managed cloud database service by MongoDB.

---

13. Use Cases

- Content management systems.
- Real-time analytics.
- Internet of Things (IoT) data storage.
- Mobile and social media apps.
- E-commerce catalogs with flexible schemas.