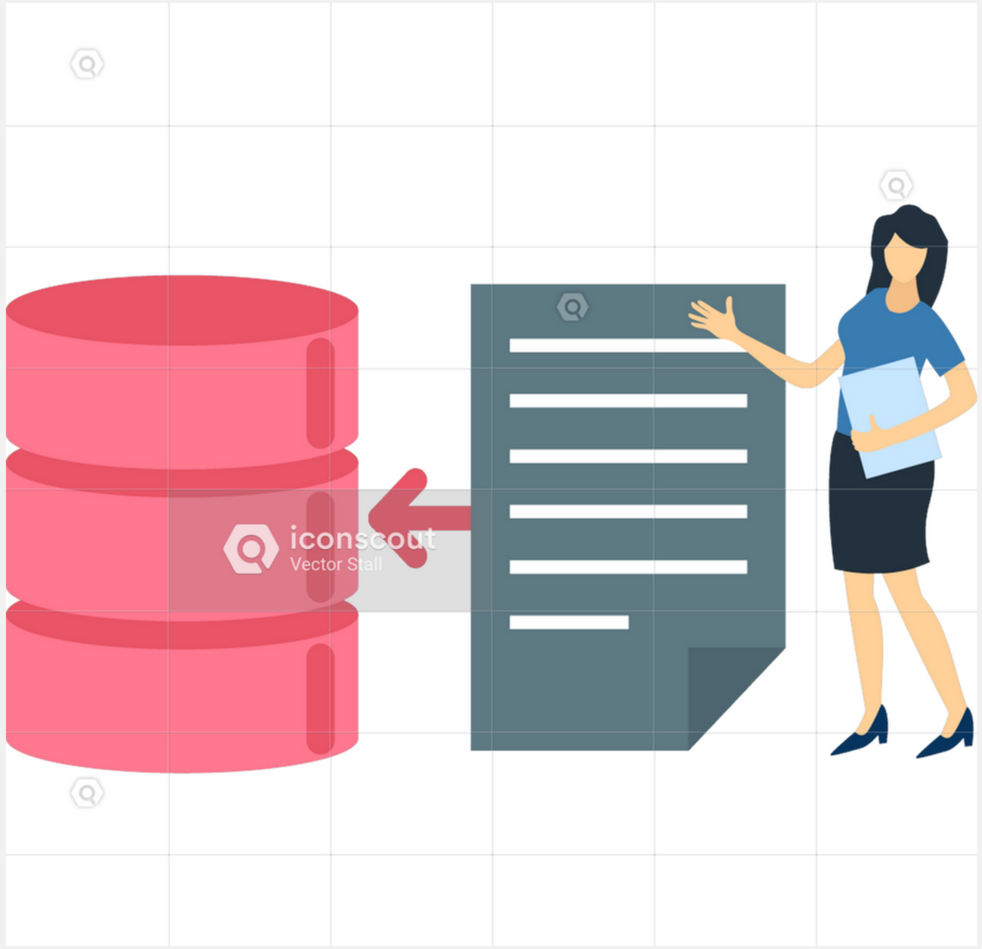

Introduction to MongoDB

Understanding NoSQL Database

What is a Database (DB)?

- A collection of similar kinds of data.
- Data collection/storage or collection of data.

What is Data?

- Collection of information.

What is Information?

- Meaningful data or processed data.

Types of Databases

SQL (Structured Query Language) Databases

- Databases that use SQL queries for operations.
- Examples: MySQL, PostgreSQL, Oracle, SQL Server.
- SQL itself is not a database, it is a query language used for interacting with relational databases.

NoSQL (Non-Relational) Databases

- Databases that do not use SQL for querying.
- Examples: MongoDB.
- Data stored in formats like JSON, key-value pairs.

What is DBMS (Database Management System)?

- A software that manages your database.
- Helps in storing, retrieving, updating, and managing data efficiently.

Types of DBMS

RDBMS (Relational Database Management System)

- Data stored in the form of tables (rows & columns).
- Maintains relationships between data.
- Examples: MySQL, PostgreSQL, Oracle.

user_id	first_name	last_name	age
1	Joe	Doe	29
2	Jane	Dan	31
3	Potter	Paul	39
4	Pil	Passot	41

Table: Users

NRDBMS (Non-Relational Database Management System)

- Data stored in the form of objects, key-value pairs, or documents.
- More flexible than RDBMS.
- Examples: MongoDB, Firebase, Cassandra.

What is CAP Theorem?

What is a Distributed Database?

- A distributed database is a system storing data across multiple servers instead of a single server.
- Ensures faster access and reliability
- Prevents data loss by distributing copies across locations
- Used by companies like Google, Amazon, and Facebook.
- **Example:** Amazon Warehouses
- If Amazon had only one warehouse in Delhi, delivery to Mumbai would be slow
- By having multiple warehouses, delivery is faster for different locations

What is CAP Theorem?

CAP Theorem states that a distributed database cannot achieve all three properties simultaneously:

- **Consistency (C):** All nodes see the same data at the same time.
- **Availability (A)**: The system remains operational even if some nodes fail
- **Partition Tolerance (P):** The system continues working despite network issues

CAP Theorem - You Can Only Choose Two

- Distributed systems can only maintain two out of three CAP properties:

- **CP (Consistency + Partition Tolerance):** SQL Databases (e.g., MySQL, PostgreSQL)
- **AP (Availability + Partition Tolerance):** NoSQL Databases (e.g., MongoDB, Cassandra)

CAP Theorem - You Can Only Choose Two

•Distributed systems can only maintain two out of three CAP properties:

•**CP (Consistency + Partition Tolerance):** SQL Databases (e.g., MySQL, PostgreSQL).
Example:
•WhatsApp (AP Model) : Messages may be slightly delayed but the app is always available.
•SQL (CP): Strong consistency, good for banking.

CAP Theorem - You Can Only Choose Two

- **AP (Availability + Partition Tolerance):** NoSQL Databases (e.g., MongoDB).

- Banking App (CP Model): Transactions are always consistent, but the app may go offline during failures.

- NoSQL (AP): High availability, ideal for social media and real-time applications.

What is MongoDB?

- MongoDB is a NoSQL database.
- It is non-relational, meaning it does not store data in table format.
- MongoDB stores data as objects (documents).

```
{
  "_id": "12345",
  "name": "foo bar",
  "email": "foo@bar.com",
  "address": {
    "street": "123 foo street",
    "city": "some city",
    "state": "some state",
    "zip": "123456"
  },
  "hobbies": ["music", "guitar", "reading"]
}
```

Mongo Shell

- It is a REPL (Read-Eval-Print Loop) or Terminal.
- Used to run MongoDB commands interactively.
- Acts as a powerful interface to interact with your database.

Collections in MongoDB

- Similar to tables in SQL but used in a NoSQL context.
- A collection is a group of MongoDB documents.
- Multiple collections together form a MongoDB database.

Documents in MongoDB

- Inside each collection, you have one or more documents.
- Documents are analogous to rows or entities in SQL databases.
- Each document holds key-value pairs (JSON-like structure).

Relationship Between Database, Collection, and Documents

- Database contains Collections.
- Collections contain Documents.
- Documents store the actual data (fields and values)
- MongoDB is a flexible, schema-less database.
- Collections work like tables, and documents work like rows.
- Mongo Shell is used to run commands directly in MongoDB.

Working with MongoDB

Step 1: Open MongoDB

- Command: Open MongoDB in CMD using -> **mongosh**

Step 2: Check Databases

- Command to list databases: **show dbs**

Step 3: Default Database

- <test> is the default database.
- Use the command db to check the current database.

Step 4: Switching to a Database

- Command: Switch to a specific database using: **use dbname**

Step 5: Collections in MongoDB

- To List Collections: Use the command: **show collections**

Step 6: Drop Database

- Command to delete the current database: **db.dropDatabase()**

Step 7: Create a Database

- Command to create or switch to a database: **use cheerz**.
- (If the database doesn't exist, it will be created once a collection is added)

Step 8: Create a Collection

- Command to create a collection: **db.createCollection("collectionName").**

Step 9: Drop a Collection

- Command to drop a collection: **db.collectionName.drop()**

Step 10: Working with Collections

- To Read All Documents : **db.collectionName.find()**

Step 11 : Inserting Data

- Insert a Single Document: **db.collectionName.insertOne({})**
- Insert Multiple Documents**: db.collectionName.insertMany([{},{},...])**
- (MongoDB auto-assigns a unique ObjectID to each document)

MongoDB CRUD commands

1. Create (Insert)

- Insert One Document: db.collectionName.insertOne({});
- **Insert Multiple Documents**: db.collectionName.insertMany([{}, {}]);

2. Read (Query)

- Find All Documents: db.collectionName.find();
- Find One Document: db.collectionName.findOne({});

3. Update (Modify)

- Update One Document: db.collectionName.updateOne({}, { $set: {} });
- Update Multiple Documents: db.collectionName.updateMany({}, { $set: {} });

4. Delete (Remove)

- Delete One Document: db.collectionName.deleteOne({});
- Delete Multiple Documents: db.collectionName.deleteMany({});

Operators

- $lt -> less than
- $lte -> less than equals to
- $eq -> check equals
- $gt -> greater than
- $gte-> greater than equal
- $and -> accepts an array -> more than one object
- $or -> accepts an array