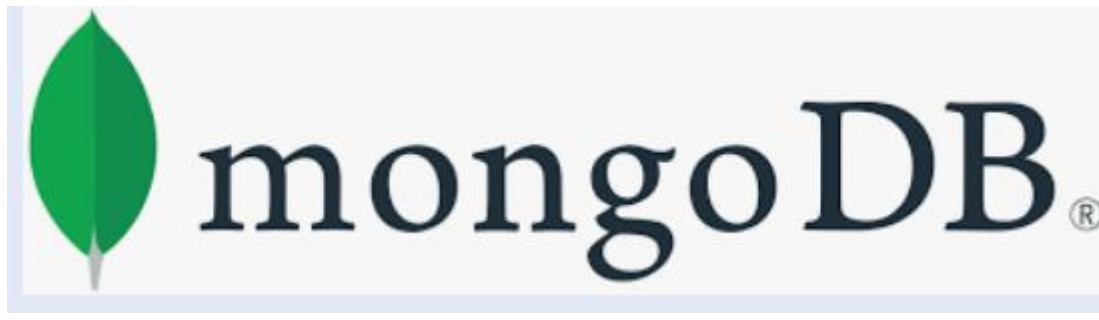


# INTRODUCTION:



MongoDB is a NoSQL database that stores data in flexible, JSON-like documents. Unlike relational databases with fixed schemas, MongoDB offers schema-less storage, allowing for diverse data structures. This flexibility makes it ideal for modern applications with evolving data models.

Known for its speed and scalability, MongoDB excels at handling large datasets and high-traffic environments. Its horizontal scaling allows adding more servers to boost capacity as your data grows.

## **MongoDB supporting feid:**

MongoDB documents are like JSON objects, with fields storing data. These fields can hold various data types like strings, numbers, booleans, and even nested documents and arrays. There's no fixed schema, so documents within a collection can have different structures. This flexibility lets you store diverse data efficiently.

## **WHAT IS DATA BASE:**

Imagine a giant electronic filing cabinet, but instead of folders and papers, it holds organized collections of data. That's essentially what a database is! It's a structured way to store and manage information electronically on a computer system.

Databases keep data organized, making it easy to access, manipulate, and analyze. They typically use a Database Management System (DBMS), a software that acts as the brain of the operation. The DBMS handles tasks like data storage, retrieval, security, and maintenance.

Think of data organization like filing cabinets with drawers and separators. Data is often stored in tables with rows and columns. Rows represent individual entries, like a specific customer in a customer database. Columns represent specific attributes of that entry, like the customer's name, address, or purchase history.

Databases come in different flavors. Relational databases, the most common type, organize data based on relationships between tables. For example, an order table might link to a

customer table and a product table. There are also NoSQL databases, which offer more flexibility for storing data that doesn't fit neatly into relational structures.

The benefits of using databases are numerous. They ensure data accuracy and consistency, prevent data redundancy (having the same data stored in multiple places), and enable efficient searching and analysis. This makes them crucial for businesses, organizations, and anyone who needs to manage large amounts of information.

## FEW COMMANDS IN mongoDB :

- **Manage Databases:** List databases (`show dbs`), switch database (`use <database_name>`)
- **Manage Collections:** List collections (`show collections`), find documents (`db.collection.find()`), filter results (`find({<filter>})`)
- **CRUD Operations:** Insert documents (`.insertOne()`), update documents (`.updateOne()`), delete documents (`.deleteOne()`)
- **Flexible Data:** Schema-less structure allows diverse data within collections.
- **Powerful Queries:** Filter (`$eq`, `$gt`), project (`{_id: 0}`), sort, limit results.
- **Data Processing:** Aggregation pipelines for complex transformations and calculations

## DATA TYPES USED:

1. **String:** The most common type, used for textual data like names, descriptions, or addresses. It must be UTF-8 encoded.
2. **Number:** Handles both integers (whole numbers) and floating-point numbers (decimals). Internally, MongoDB can store them as 32-bit or 64-bit integers depending on the value's size.
3. **Boolean:** Represents true or false values, useful for flags or binary conditions.
4. **Array:** Stores an ordered list of various data types within a single field. Great for collections of items or attributes.
5. **Object (Document):** Allows embedding complex data structures within documents. Nested objects create hierarchical relationships within the data.
6. **Date:** Stores date and time information. You can construct Date objects or use timestamps in milliseconds since the Unix epoch.
7. **ObjectId:** A unique 12-byte identifier automatically generated by MongoDB for each document, useful as the primary key.
8. **Binary Data:** Stores raw binary content like images, audio, or files.
9. **Null:** Represents the absence of a value for a specific field.

These core data types provide flexibility for storing diverse data. Additionally, MongoDB offers:

- **Decimal128:** For high-precision decimal calculations, especially for financial data.
- **Regular Expression:** Stores search patterns for text-based queries.

Understanding these data types is crucial for effectively modeling and storing data in your MongoDB collections.

### Simple Inventory Management with MongoDB

**Scenario:** Imagine a small store that keeps track of its products in a MongoDB collection named `products`. Each product document will have details like:

- `_id` (automatically generated by MongoDB)
- `name` (product name)
- `price` (product price)
- `quantity` (number of items in stock)

### Example:

JavaScript

```
// Insert a document
db.users.insertOne({ name: "Alice", age: 30 })
```

```
// Find all users
db.users.find({})
```

### OUTPUT:

```
{ "_id" : ObjectId("6389d798b4be228888888888"), "name" : "Alice", "age" : 30 }
```