**What is a Database?**

Basically, the database is an organized collection of Data, which is usually managed by DBMS. Basically, if we focus on our computer system is basically used to store data like the hard disk in it is also used to store data, but the problem is that it is not in an organized form, So to make data organized we use DBMS which are software like MongoDB, PostgreSQL, Oracle .that are used to manage data.

Now remember the software that we create cannot store data permanently like we create an array of users and it stores that array of data in memory but it is temporary, so it basically stores it on RAM (primary storage) which is not permanent because it is volatile memory. In the long term it cannot store data ,it can only process data.

Working principle that how data is fetched from database

Server process

client request the request but database it not save the it stores

data it sends it data.

to database.

Now when data is stored in the database so we can easily access it and manage it.

Now we have various kinds of DB but all are used to store data, but in different way it can be in form of tables , schemas etc.

What is MongoDB?

1-It’s a No-SQL based database, no-SQL means a DB which not use SQL for query.

2-its scalable, open source , high-performance , document-oriented database.

[**What is a Document Database?**](https://www.digitalocean.com/community/conceptual-articles/an-introduction-to-document-oriented-databases#what-is-a-document-database)

Breaking free from thinking about databases as consisting of rows and columns, as is the case in a table within a relational database, document databases store data as *documents*. You might think of a document as a self-contained data entry containing everything needed to understand its meaning, similar to documents used in the real world.

The following is an example of a document that might appear in a document database like MongoDB. This sample document represents a company contact card, describing an employee called Sammy:

Sammy's contact card document

{

"\_id": "sammyshark",

"firstName": "Sammy",

"lastName": "Shark",

"email": "sammy.shark@digitalocean.com",

"department": "Finance"

}

Copy

Notice that the document is written as a JSON object. [JSON](https://www.digitalocean.com/community/tutorials/an-introduction-to-json) is a human-readable data format that has become quite popular in recent years. While many different formats can be used to represent data within a document database, such as XML or YAML, JSON is one of the most common choices. For example, MongoDB adopted JSON as the primary data format to define and manage data.

Basically , document-oriented DB means asa DB jisma rows and columns ka form ma data store na ho like jo MS-access hai usme data rows and column ka format ma hota hai bcuz that is SQL based DB , but in No-SQL data is stored in JSON object.

## ****Why is DBMS Required?****

Database management system, as the name suggests, is a management system that is used to manage the entire flow of data, i.e, the insertion of data or the retrieval of data, how the data is inserted into the database, or how fast the data should be retrieved, so DBMS takes care of all these features, as it maintains the uniformity of the database as well does the faster insertions as well as retrievals.

## ****Why is RDBMS Required?****

RDBMS on the other hand is a type of DBMS, as the name suggests it deals with relations as well as various key constraints. So here we have tables which are called schema and we have rows which are called tuples/data. It also aids in the reduction of data redundancy and the preservation of database integrity.

Relational Database Management System is an **advanced** version of a DBMS.

RDBMS : it is also DB management system , but it is used to manipulate relational databases.

|  |  |
| --- | --- |
| DBMS | RDBMS |
| [DBMS](https://www.geeksforgeeks.org/introduction-of-dbms-database-management-system-set-1/) stores data as file. | [RDBMS](https://www.geeksforgeeks.org/rdbms-architecture/) stores data in tabular form. |
| Data elements need to access individually | Multiple data elements can be accessed at the same time |
| No relationship between data. | Data is stored in the form of tables which are related to each other. |
| It stores data in either a navigational or hierarchical form. | It uses a tabular structure where the headers are the column names, and the rows contain corresponding values. |

For more check : <https://www.geeksforgeeks.org/difference-between-rdbms-and-dbms/?ref=previous_article>

NO SQL and SQL

|  |  |
| --- | --- |
| SQL | NO-SQL |
| Its RDBMS-based database, which means it stores data in form of a table. | NON-relational or distributed |
| In it, we create predefined schemas,and based on that schema we create tables | Data stored in JSON format (also known as collection, bcuz it store data in form of documents,record ) |
| Vertically Scalable (now it means data is vertically increasing,bcuz it is in table format) | Horizontal scalable (its mean data increase horizontally) |

**MongoDB installation**

**‘ mongoDB community server ’** means that we are installing core mongoDB database in our computer takay mein apna computer ma mongodb ko run kr sako bina kisi external service ko ya mongodb atlas ko use kiya. MongoDB is a general-purpose document database. With the Community Edition, you can self-manage and host it locally or in the cloud.

Isse yeh faida hua kay we can configure Database by our own, means that we can do configuration in it.

‘ mongoDB community server ’ ,it is used to manage, update data in mongoDB, we can say that if we have installed ‘ mongoDB community server ’ it means we installed the code that will run mongoDB on our computer.

Steps for installation:

1-run setup which you have installed

2-then go inside the bin folder, then first we will run mongod (mongo demon) in CMD. Mostly it generate shutdown it is due to the path not defined bcuz by default there is a path define i.e : “c: data/db ” . now to solve this error write “mongod –help” by this we get all commands then find the command for changing directory which is ‘—dbpath’ .

3-then write in terminal like this : mongod --dbpath (here write path name which is defined from c drive)

e.g : mongod –dbpath /db

now basically jo humara ‘mongod’ also known as mongodb demon , now basically ‘mongod’ is running our database means jo be code hai mongodb ka wo sara run ho rha hai with the help of mongod.

**MongoDB Compass:** it is basically a tool for visualizing our database which is running on terminal with the help of mongod. also known as GUI for database.

**Connecting Mongodb with mongoDB compass:**

1-just for checking first we will stop the database which is running on CMD by using command : ctrl+c, now when we put the database connection string or we can also called it as URl , so it raise error bcuz we have stopped our database on CMD. (this step is not necessary just for learning purpose)

2-then again we run the database by using the same command i.e : mongod –dbpath /db .

3-now when we put database connection string in Mongodb compass so it make connection bcuz database is in running.

**Now see the structure**

**In RDBMS structure is like :**

Database table row

Now its telling that in RDMS we have Database , then in it we have table ,then we have row (which represent data).

**In NO-SQl (mongodb) structure is like :**

database collection document(JSON

Object

In mongodb , there are collections in place of tables and document in place of row.

Now remember in MongoDB there is no need of schema as in SQL databases we create schema which means kay schema basically ek table ka structure ka ban jata hai jo bs ek dafa phr ussi ki base pa data store hoga I mean like jo fields usma bnayi hai bs unka A/C hi data store later on asa nhi ka koi or field be add krdein. But in NO-SQL mongodb there is no schema which means that agar ek document(JSON object) humna bnaya jisma name and age ki property di or ek dusra document bnaya usko name,address and phone ki property dedi so it will accept bcuz its dynamic.

What are collections in NO-SQL ?

Remember at least we have to create one collection after creating database. Collection is like jasa SQL ma ek database ma number of tables hotay hain issi tarah No-SQL ma tables ki jaga collection hoti hai like agar humna employees ka data store krna hai toh hum simply ek collection bnyega ‘employees’ ka name say phr further usme document ki form ma data store krenga like in table we store it in form of ‘row’.MongoDB assigns every document a unique id.

Now remember basically hota iss tarah sa haka jab hum mongo community server dpwnload krtay hain toh usma jo bin ki folder hota hai usma 3 ‘.exe ’ ki files milti thi which are : mongo , mongod , mongos.

These all 3 files are tools to run mongoDB , where mongod(mongo daemon) is used to compile or we can say run the core code on terminal of mongoDB , then mongo or mongos is used to interact with MongoDB on terminal by using commands and for GUI we use mongo compass but its better to use tool like mongos or mongo to interact with DB on terminal better than GUI. But now problem is that in the server upper than 6.0version so there is no mongo.exe file in bin so that’s why must install ‘mongosh’ which help you to interact with ur DB which is running on terminal with the help of mongod. also define your mongoDB path in system variable for that:

1-Go to setting of computer

2-System environment vriable

3-find there path field and edit it and Add new path i.e: c:\programs\mongodb\bin

By doing above step use yeh hog aka ab jab huma mongosh ya mongo ko run krna hoga toh kisi be directory ma jaka mongosh ya mongo likhein ga toh wo waha par mongoDB run krdega , but we will use mongosh bcuz mongo is not available in version >=6.0 . basically mongosh or mongo command run the MongoDB shell .

Remember that by default jab hum computer pa mongod ko run krtay hain toh wo localhost:27017 pa instance ko run krta hai or uska A/C database show hojati hai, but agr koi or instance sa connect kreinga toh uska A/C database show hojagi. (instance means jo hum mongod ki command lag ka ek code run hota hai wo instance kehlata hai)

Commands for MongoDB shell so we can interact with DB on terminal rather on GUI where we use mongo compass.

1-show dbs = by using this command all databases will be shown.

2-use dbName = This command is used for using anyone database from all. Like I have two DB that are company1 and company2, so if I want to use or do any operation in the company1 DB so I just write the ‘use company1’ command.

**CRUD operations**

C – create (this Is related to creation of DB, collections, documents , fields.)

R – Read (it means to fetch/query data from DB, it can be a single document or multiple documents etc..)

U – Update (It is used for updating the data )

D – Delete (it is used for deleting the data)

For more understanding check Mongodb docs .

Also check = c: \learning MongoDB, for more CRUD operations.

**What is Mongo Atlas?**

So, basically, it is also a tool to work with Mongodb but its most benefit is that we don’t have to deploy MongoDB in our computer if we are working with Mongodb atlas, deployment sa mtlb hai jasa humna mongoDb ko install kia then phr ‘mongod’ ko run krka core MongoDB ka code-base ko run kia or then phr ‘mongosh’ ka through mongoDb shell ko run kia, so this all steps we don’t need if we are working with MongoDB atlas. Now basically hota yeh haka jab hum direct MongoDB ko apni computer/machine ma deploy krka chalatay hain toh uski koi gurantee nahi hoti kay uski kya speed rahegi , computer kahin slow toh nahi hoajyega kiu ka its not properly managed so that’s why mongoDB atlas is helpful bcuz it’s a managed cloud-based database provided by MongoDB. We can also called it as MongoDB on the cloud which means that jo humara mongoDB abhi local computer ma chl raha tha wo ab cloud pa chalega.

Steps to create database using Mongo atlas:

1-sign-in on Mongo atlas

2-create organization

3-create project

4-create cluster (cluster in MongoDB atlas is basically a No-SQL database as a service offering in the public cloud, which is managed by mongoDB service.)

5-copy the connection string of the cluster which you have created (select mongo compass for connection ).

6-paste that connection string in mongo compass and now your cluster is opened on mongo compass now you can create databases,collection in it.

Remember as we are using MongoAtlas so we don’t need to take tension that what if our computer become slow, etc . so kya humari site slow hojayegi, NO bcuz now we are using cloud-based database not our local database which we deployed on our computer.

Learn mongoDB query operator/selector from mongoDB official docs.

# **Aggregation Operations**

Aggregation operations process multiple documents and return computed results. You can use aggregation operations to:

* Group values from multiple documents together.
* Perform operations on the grouped data to return a single result.
* Analyze data changes over time.

Aggregation basically hum use krtay hain for suppose hum asi query bnana chahatay hain jis ma kuch stages hon like we have 5 pizza doc now we want that first of all find the pizzas whose size is medium , then find those whose flavor is pepperoni , then those who are pepperoni put some extra cheese on them.

Now we can see that in above query we have 3 stages first we have to find medium size pizza, then pepperoni flavor , then add some cheese .so in these type of cases we use Aggregation operation.

To perform this operation we can use aggregation pipeline :

* Each stage performs an operation on the input documents. For example, a stage can filter documents, group documents, and calculate values.
* The documents that are output from a stage are passed to the **next stage**.
* An aggregation pipeline can return results for groups of documents. For example, return the medium pizza, find pepperoni flavor, and then add extra cheese on that.

Remember if we want that kay jasa hum chahtay hain hum jab mongoDB apna local computer pa chla rahay thay Mongo shell pa by using ‘mongosh’ tool , to issi tarah usko hum jab mongoDB atlas pa run kray toh usko be jasa upper humna sikha ka kis tarah mongo Compass ko connect krna hai mongoDB atlas say but ab hum yeh chahtay hain hum uss mongoDB atlas ko mongo shell pa be chalayein toh uska lia simply jaha mongo atlas ma humna connection ka lia mongo compass use kia tha waha uski jaga mongo shell ko select krna hai or phr jo wo Db\_conn\_string provide kreinga usko apna CLI pa paste krna hai or bs phr wo CLI pa hi password puchega waha wo btana hai or bs phr wo connect hoja ga.

Lookup

Basically yeh jo lookup ka function hota hna yeh jasa sql may left join hota hna yebi uski tarah work krta hai hai, but basic yeh joining ka liya use hota hai

Like two collection user and order , toh hum agr chahtay hain kay jo user ka record hai wo order mabi nazar aye toh automatic si baat hai humey unka darmayain ek connection/join bnana hoga , so here come lookup. Ab lookup yeh krega kay dono collection kay document ma say koi specific fields lega or unki base pa dono ko apas ma join krdega, like jo user collection hai usme jo \_id field hai and order ki collection ma jo user\_id field hai un dono field ki base pa unko apas ma join krdia which means that ab jo user ka data hai wo order wali collection ka document mabi agya but srf un ids ka liya jo user collection may \_id and order collection may user\_id match hui.

* Lookup ko mostly aggregation pipeline ka sth use kia jata hai.

Before understanding syntax remember right now we are doing lookup query in order collection bcuz we want the user record in order collection.

Syntax:

{

$lookup: {

from: yaha par wo collection name ayega jis ka sth join krna hai , in our case it is “user” ,

localfield: yaha order collection may say jis be field ko user collection ma compare krwana hai wo ayegi, in our case it is “user\_id” ,

foreignfield: yaha user collection may say jis be field ko user\_id wali field sa match krna hai wo ayegi, in our case it is “\_id” ,

as: yaha par finally wo field ka name ayega jisma jo lookup ka result ayega wo store hoga, toh yaha par kuch be name de sktay hain, in our case hum usko “user\_details” dedenga, so finally jo be user ka record user collection sa milega wo order collection may user\_details ma store hojayega.

}

}

* Now remember Lookup jo result return krta hai wo array ki form may hota hai , like ab jo user ka data “user\_detail” field ma store hua hoga wo ek array of object hoga.
* Now agr hum chahtay hain kay user\_detail ma array of object na store ho blka simply jo user data wala object hai wo store hojaye toh humna jo uper query likhi hai usi ko extend krsktay hain like this:

{

$lookup: {

from: ‘user ‘,

localField : ‘user\_id’ ,

foreignField: ‘\_id ’ ,

as: ‘ user\_detail ’

},

$addField:{ //ab yeh basically field ko add krega orders collection ka jo doc hai usme

user\_detail:{ //yeh jo user\_detail hai wo field ka name hai jo add hogi.

$arrayElemAt: [ “$user\_detail”, 0 ] //or yaha basically user\_detail field ko object dia hai or usme yeh keh rhy hain jo user\_detail wali field uper as a result of lookup hna , usme jo array hai usme jo 0 index pa element hna wo yaha par store krdo.

}

}

}

Toh iss tareeka say ab wo jo user\_detail ma array of obj tha wo srf ab ek object ki tarah store hogya.

But yaha question haka iss tarah toh phr order collection ka doc ma two field with same name hojayegi which is user\_detail nhi asa nhi hoga bcuz jab hum aggregation pipeline ka use kr rhay hain toh wo jo pehlay lookup ka through user\_detail field hogi uski replacement ma yeh jo new user\_detail field add ki hi wo ajagi.

## Pattern matching using $regex operator

This operator provides regular expression capabilities for pattern matching stings in the queries. Or in other words, this operator is used to search for the given string in the specified collection. It is helpful when we don’t know the exact field value that we are looking in the document. For example, a collection containing 3 documents i.e.,

{

name: "Tony",

position: "Backend developer"

}

{

name: "Bruce",

position: "frontend developer"

}

{

name: "Nick",

position: "HR Manager"

}

and we are looking for developer information. So, with the help of the $regex operator, we create a pattern(i.e., {position: {$regex: “developer”}}) that will return only those documents that contain developer string.