MONGODB

Path ---> C:\Program Files\MongoDB\Server\6.0\bin

How to run mongodb and setup:

1.) Add the path to envoirement variables,
As you set the path for java or jdk path.

Check the version –

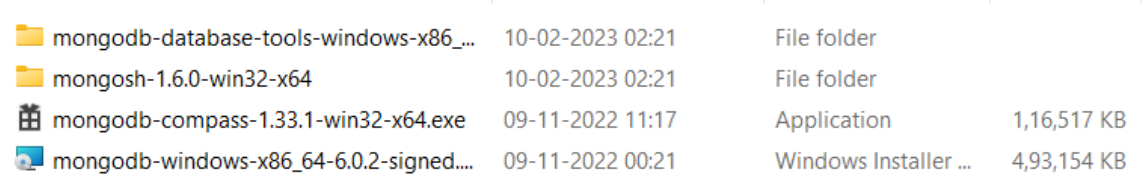
```
C:\Program Files\MongoDB\Server\6.0\bin>mongod --version
db version v6.0.2
Build Info: {
    "version": "6.0.2",
    "gitVersion": "94fb7dfc8b974f1f5343e7ea394d0d9deedba50e",
    "modules": [],
    "allocator": "tcmalloc",
    "environment": {
        "distmod": "windows",
        "distarch": "x86_64",
        "target_arch": "x86_64"
    }
}
```

```
C:\Program Files\MongoDB\Server\6.0\bin>mongod
{"t":{"$date":"2023-02-08T02:13:30.583+05:30"},"s":"I",  "
 specification","attr":{"spec":{"incomingExternalClient":{
t":{"minWireVersion":0,"maxWireVersion":17},"outgoing":{"m
```

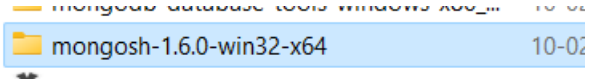
2.) C directory me data or db naam ka folder banana h, pahle data uske andar db.

C:\data\db

3.) Agar version 6.0 ya use upr ka ho to hume “mongosh” download karna padega and uska path set karna pdega, tbhi code run hoga.

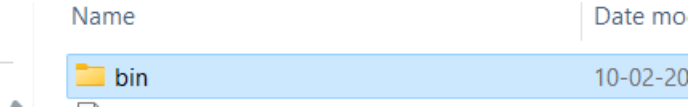


i)



ii)

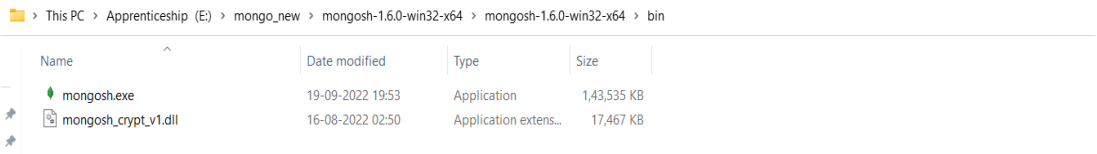
iii)



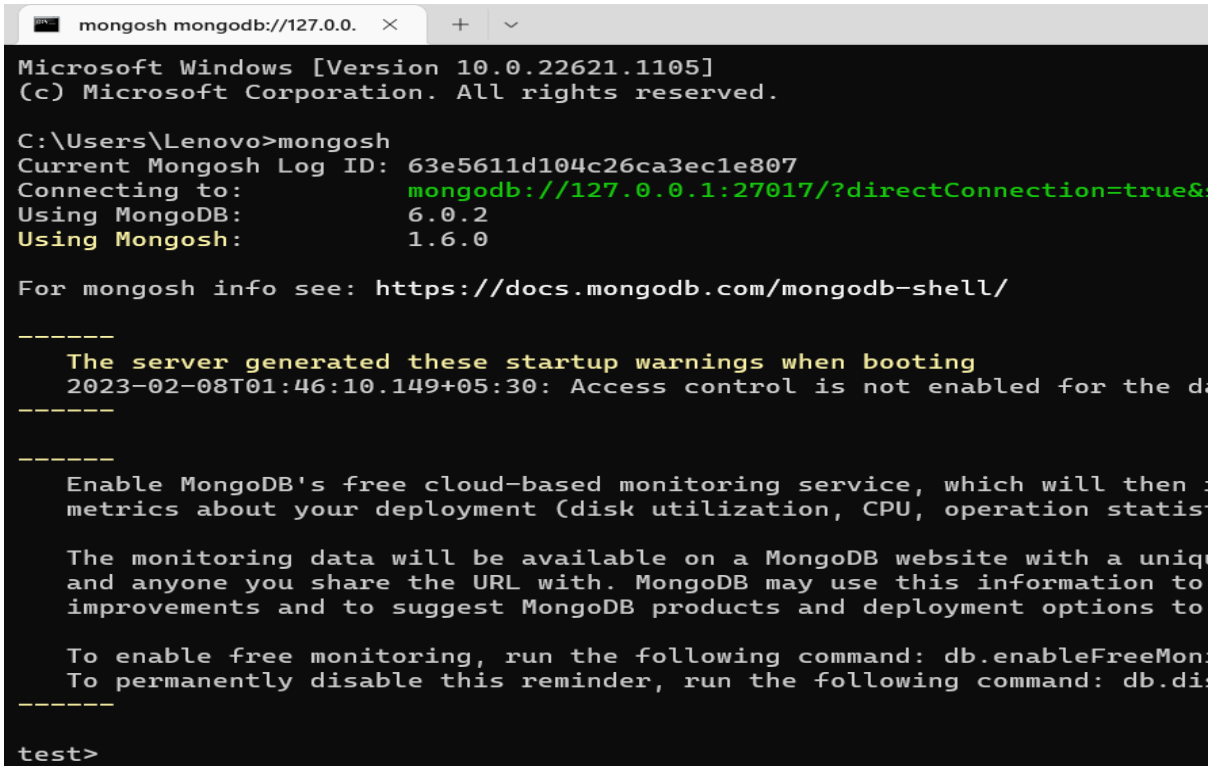
iv)

v) copy path

E:\mongo_new\mongosh-1.6.0-win32-x64\mongosh-1.6.0-win32-x64\bin



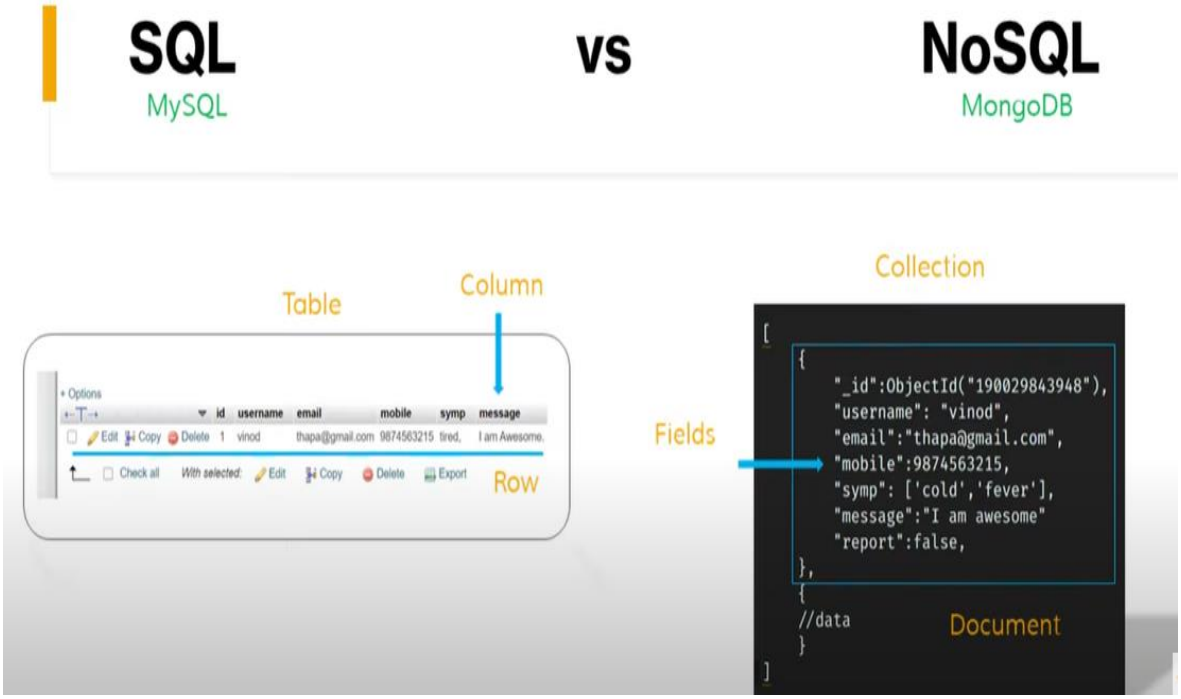
vi) Run command on cmd → (mongosh)



Successfully run.

Check the database :

```
test> show dbs
admin    40.00 KiB
config  108.00 KiB
local    80.00 KiB
mydb     80.00 KiB
test>
```



For Creating new database:-

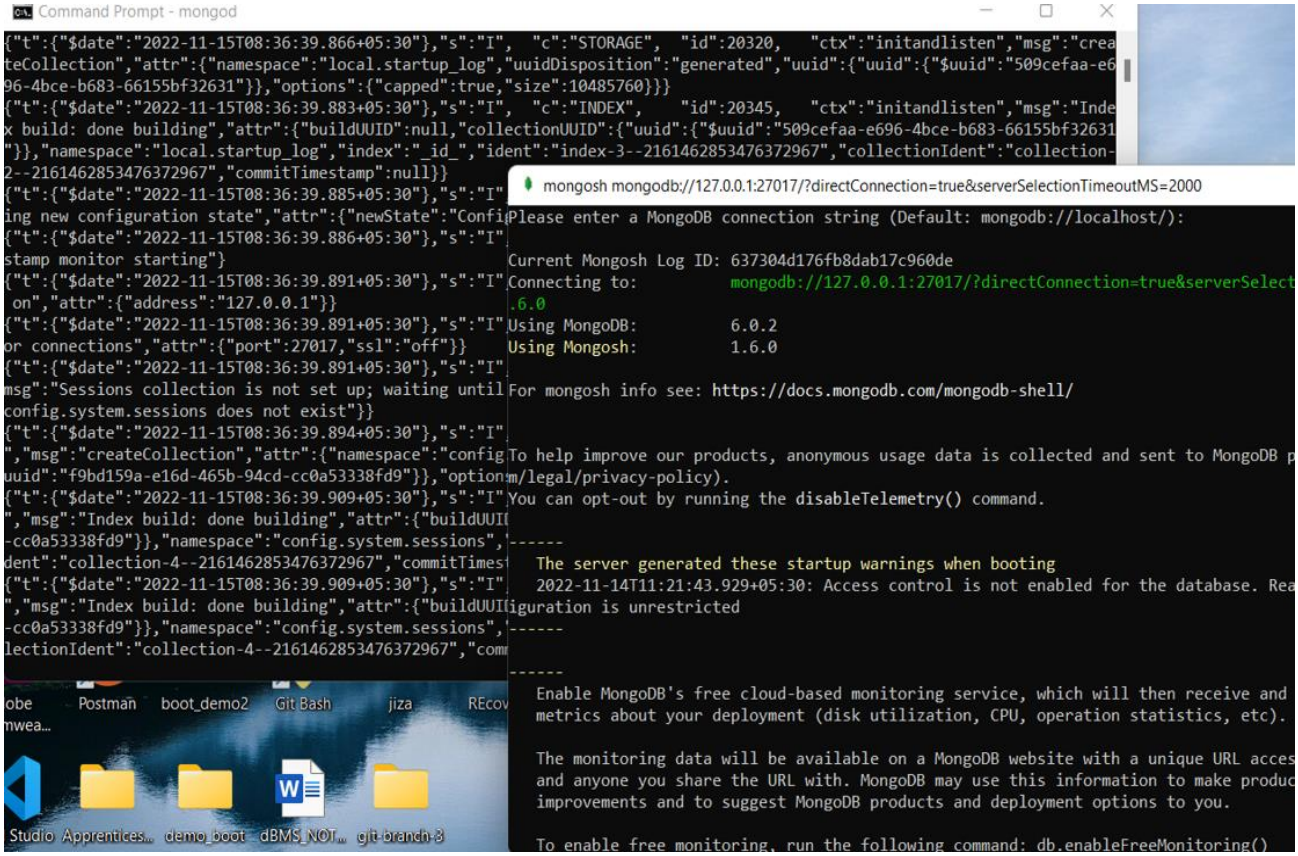
use databaseName

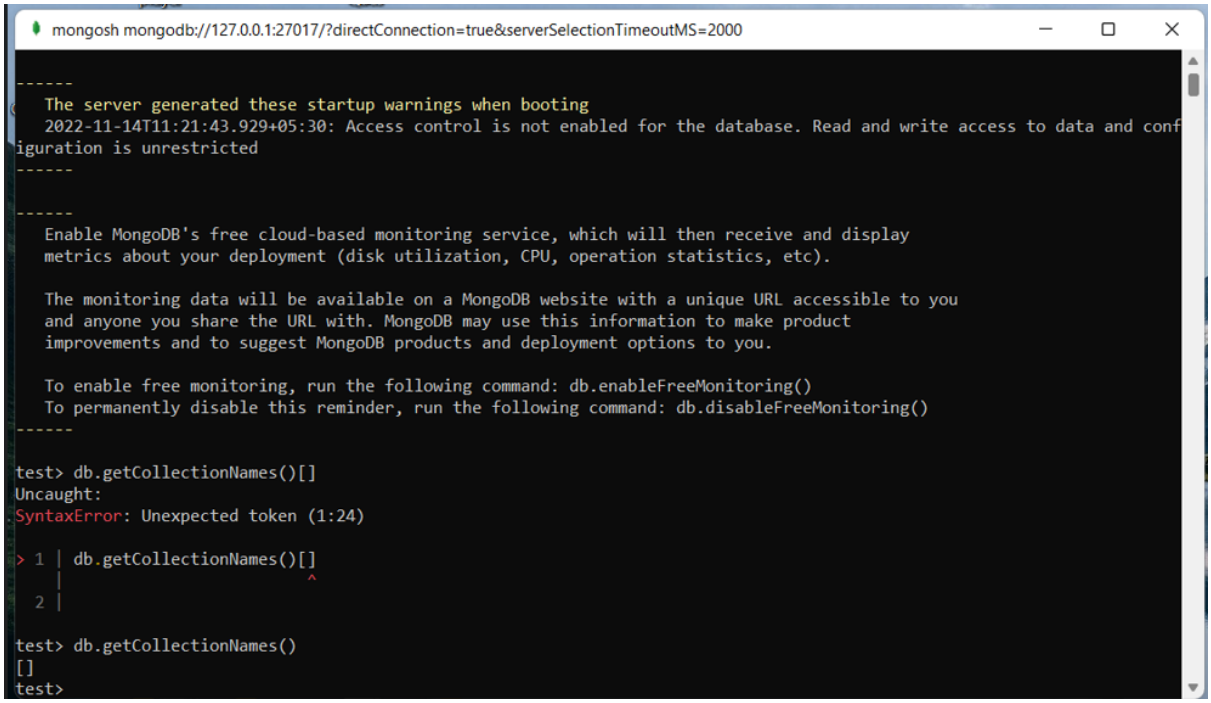
Check the current database:- db

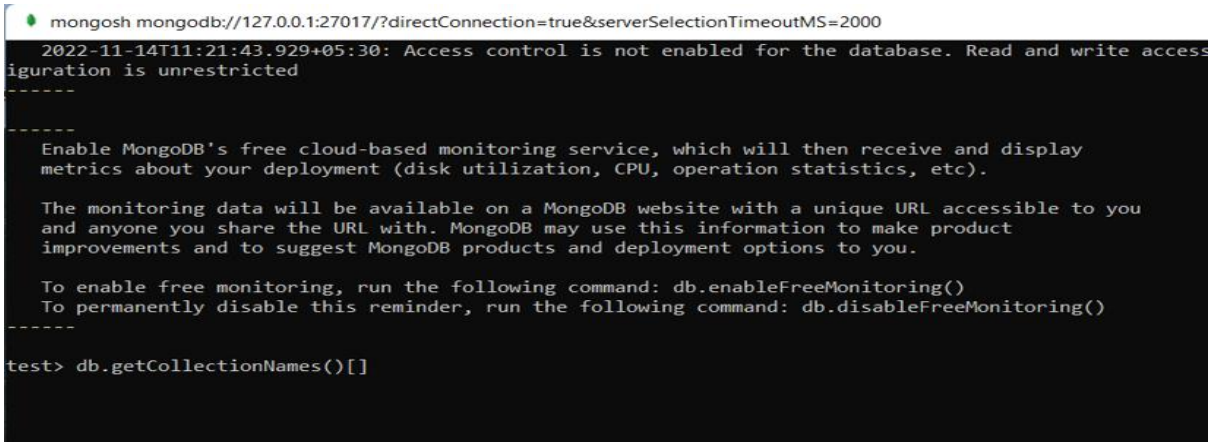
Database ke andar kitne collection hai?

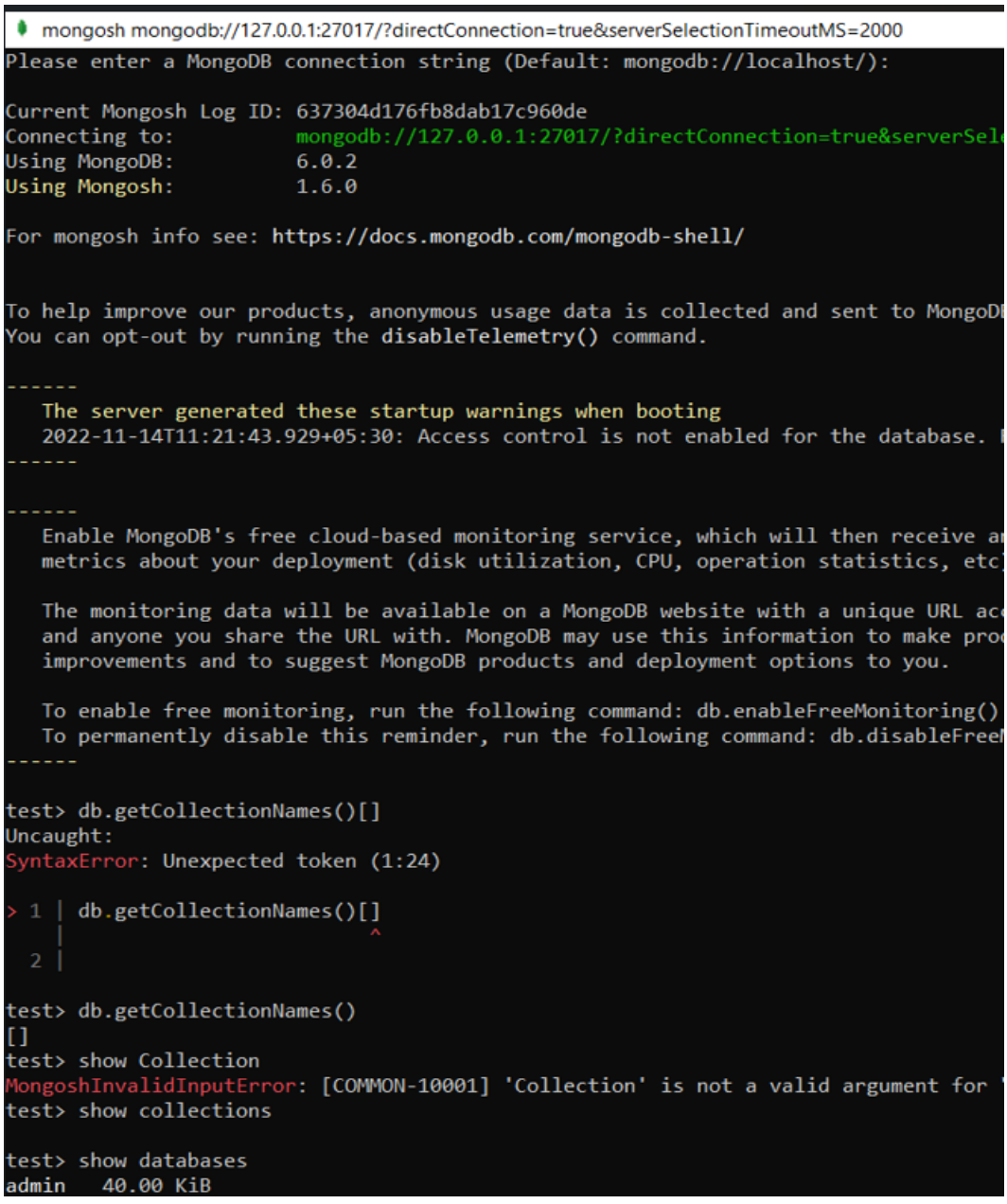
Show collections

2nd way……


Command Prompt - mongod
mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
Please enter a MongoDB connection string (Default: mongodb://localhost/):
Current Mongosh Log ID: 637304d176fb8dab17c960de
Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelect
.6.0
Using MongoDB: 6.0.2
Using Mongosh: 1.6.0
For mongosh info see: https://docs.mongodb.com/mongodb-shell/
To help improve our products, anonymous usage data is collected and sent to MongoDB p
You can opt-out by running the disableTelemetry() command.
The server generated these startup warnings when booting
2022-11-14T11:21:43.929+05:30: Access control is not enabled for the database. Rea
iguration is unrestricted
Enable MongoDB's free cloud-based monitoring service, which will then receive and
metrics about your deployment (disk utilization, CPU, operation statistics, etc).
The monitoring data will be available on a MongoDB website with a unique URL acces
and anyone you share the URL with. MongoDB may use this information to make produc
improvements and to suggest MongoDB products and deployment options to you.
To enable free monitoring, run the following command: db.enableFreeMonitoring()
Postman
boot_demo2
Git Bash
jiza
Studio
Apprentices...
demo_boot
dBMS_NOT...
git-branch-3

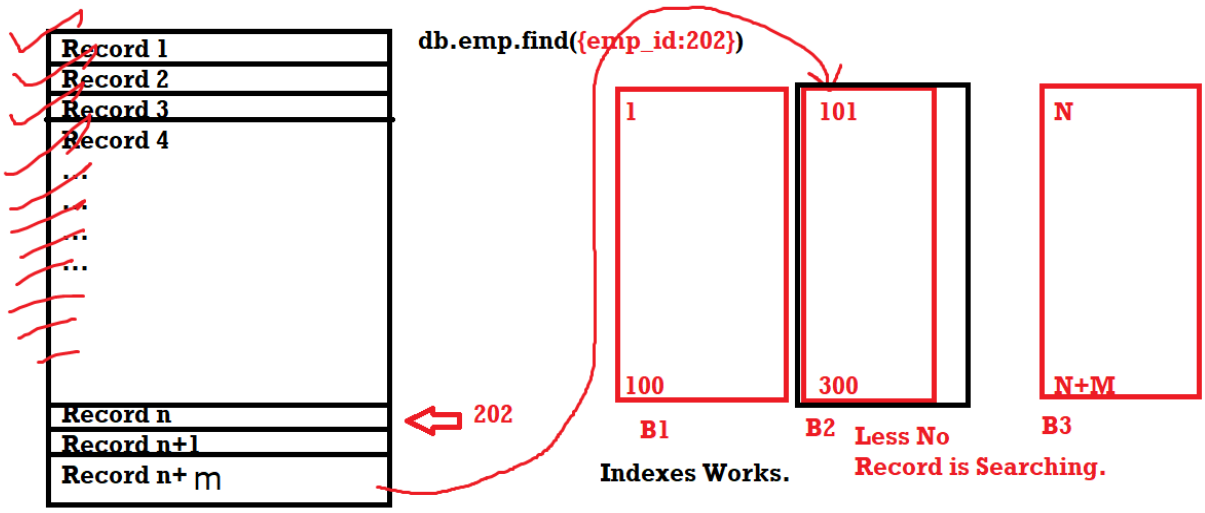

mongosh mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000
The server generated these startup warnings when booting
2022-11-14T11:21:43.929+05:30: Access control is not enabled for the database. Read and write access to data and conf
iguration is unrestricted
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).
The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.
To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
test> db.getCollectionNames()[]
Uncaught:
SyntaxError: Unexpected token (1:24)
> 1 | db.getCollectionNames()[]
2 |
test> db.getCollectionNames()
[]
test>



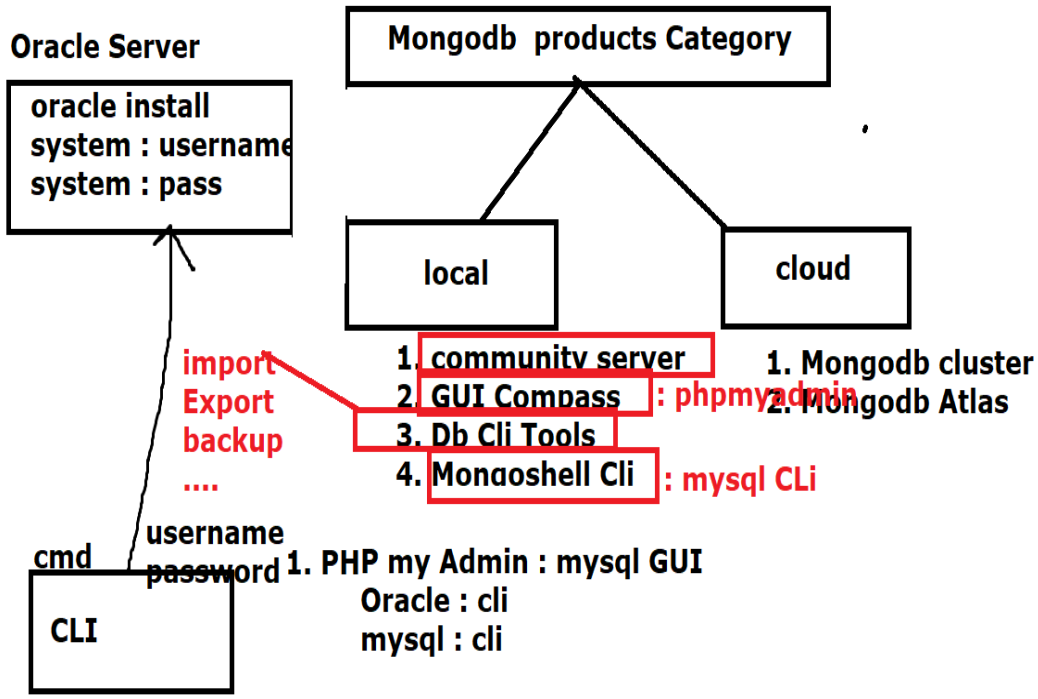


Clear data :- cls , quit()

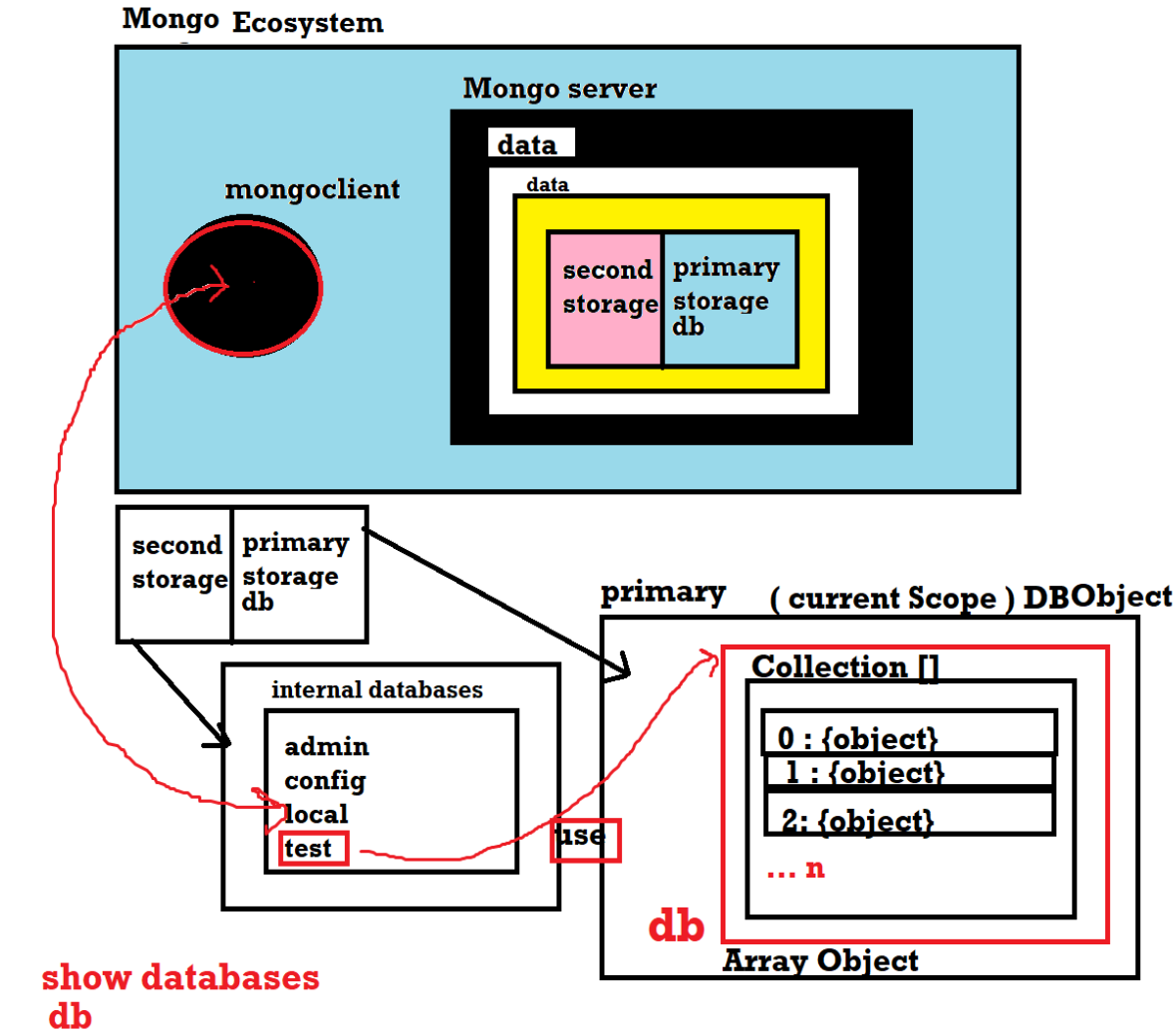
How-Index-works –



mongo-db –



Mongo-Architecture –



RDBMS VS Document Data (Json Data)

student :table

	s_id	s_name	s_class
1	1001	ravi	12
2	1002	shiv	10
3	1003	vikas	11

: columns

: records/rows

SQL → Structured Database.

R1 : 1X3
R2 : 1X3
R3 : 1X3
} order

student : collection : Array

[

Keys

Objects

{ s_id :1001,s_name :ravi ,s_class:12 } :0

{ s_id :1002,s_name :shiv ,s_class:10 } :1

{ s_id :1003,s_name :vikas ,s_class:11} :2

]

values

No order : Unstructured db

NO-SQL Database

Netstate –

cmd> netstat -a

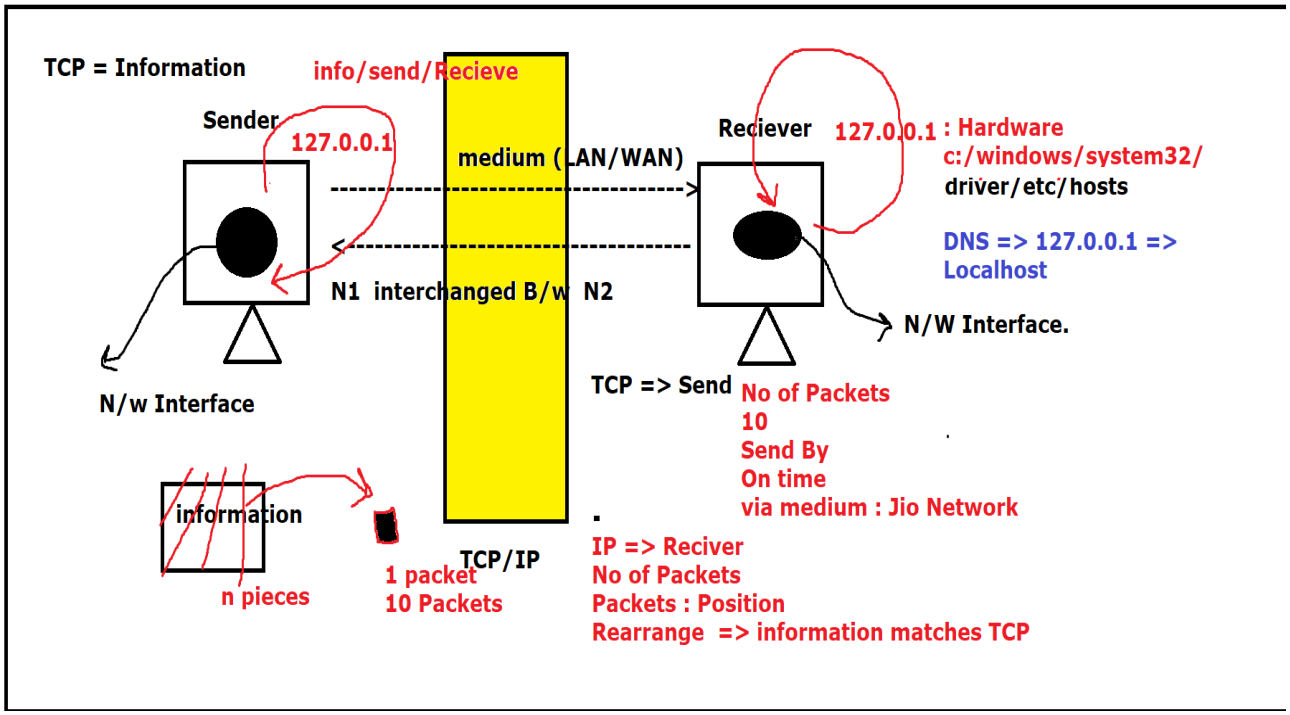
+----------+

Active Connections

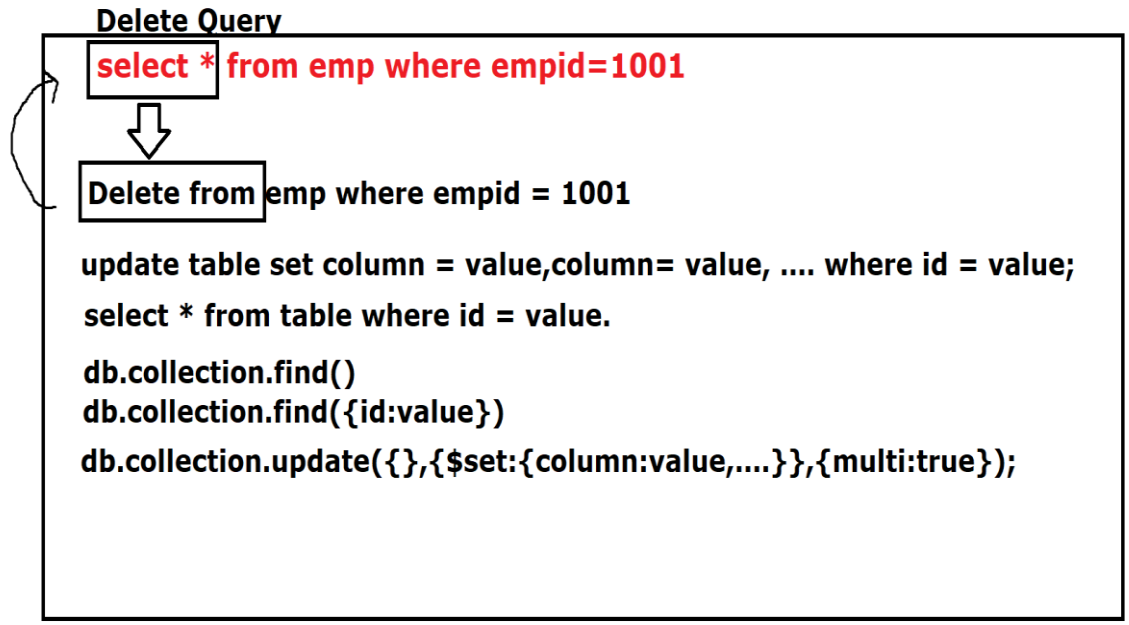
+----------

TCP/IP (proto)	Local Address	Foreign Address	state
	127.0.0.1	127.0.0.1	Establised.
		::1	Listening
			.
			Blocked
UDP	localhost	localhost	

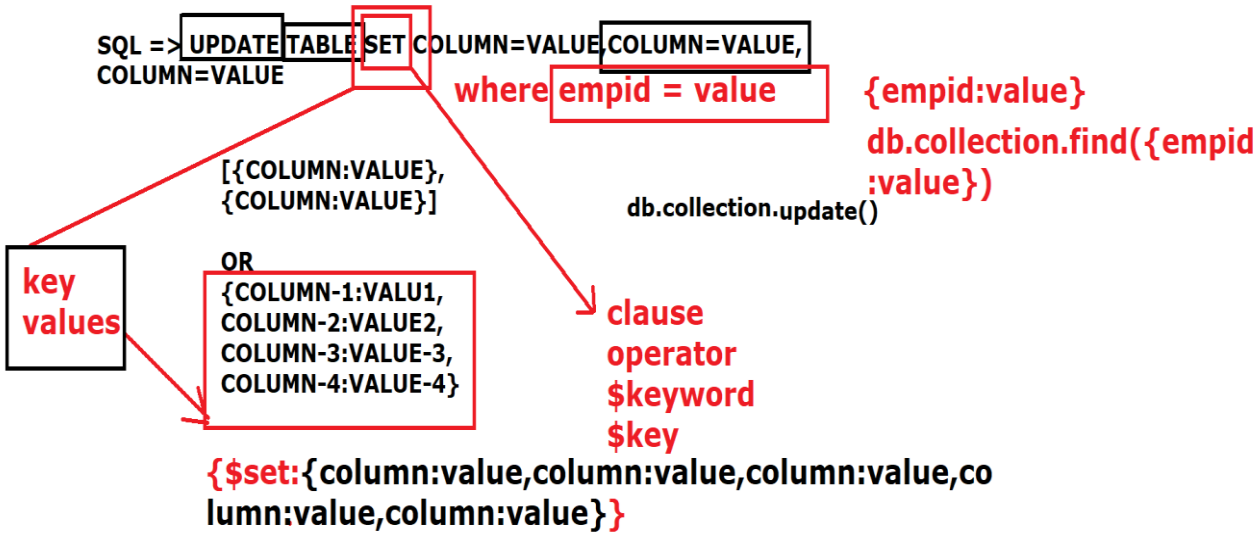
Networks –



Query-for-select-and-update –



update-syntax—



Introduction to Mongodb:-

online compiler :

https://www.jdoodle.com/online-mongodb-terminal/

online/offline =>

oracle

mysql

mongodb => NoSQL database.

1. it is mongodb or mangodb

it is mongodb

Front Where mongodb name came from:-

mongodb name : it conveys Huge and very very Large Data.

it is a product, mongo community.

mongo company not only it makes databases, but other cloud

software.

scope of mongodb:-

What is full stack web-developement

stack : collection of Langauges, or technologies required by any Application.

is called full stack.

Any person who knows all the technologies related to web-developement

is called full stack web-developer.

This stack is langauge dependent:-

1. Python Full stack
2. Java Full Stack
3. .Net Full Stack
4. PHP Full Stack
5. JS Fulls stack

Front End Technology : Common

1. Html/css/Js/Bootstrap/Tailwind/JQuery

Database End Technology

1. Relational Database (Tables Rows and Columns). => SQL

    1. mysql,oracle, SQL Server, sqlite, MS SQL...., Redshift SQL.

2. **Document Database** (Object Database | Unstructed Database | No Schema Database or ArrayObject Database or JSON Database or NOSQL database, BSON Database) => SQL not allowed.

1. mongodb
2. counchDB
3. RedisDB
4. memechachedb
5. Casendra
6. Apache Casino
7. DynamoDB

....

....

Cloud Paid Database

Note ::

Backend Technology

1. Python Full stack + Django/Fask + DRF (Django Rest Framework)

2. Java (core Java|Advanced Java[Jsp| Servelet|jdbc | struts | sprint | sprint Boot | microservices | maven | log4J])

3. .Net Stack (C#, Asp.net, ADO.net , MVC, Blazor, MVC + .netCore, Entity Framework)

4. PHP (core PHP, Advance PHP (PHPCli,PHPWeb,MVC, MVC Framework [codeIgnitor and Laravel,wordpress,opencart,cms(shopify or woocommerce)]))

5. **<u>Js Technology</u>**

    1. MERN Stack
    2. MEAN Stack
    3. MEVN Stack

    M => Mongodb => NOSQL database.
    X => where A,R,V
        A => Angular-Js/Angular : Front End Application.
        R => React Js
        V => VueJs

    E => Express Js => framework of NodeJs, Backend Service Codes (API)
    N => Node Js => It provides Service Side Runtime Environment for server.

    Mongodb is built on, JS technologies.

    Using Js technologies Later on you can upgrade in Mobile App Developement

React ---mobile--> React Native.
Angular ---mobile---> Ioinic or PhoneGap (Typescript).
VueJs ---mobile----> Native-Script.

These all are used, for mobile App Developement.
R/A/V => Dynamic FrontEnd App (SPA | single page Application)

SPA => core Langauge + Core Framework

****

ReactJs + NextJs

Angular + NestJs

VueJs + NuxtJS.

Products of mongodb:-

1. locally available

1. mongodb community server : Mongodb locally System, creates server.

cmd => mongod => Run keep active mode.

2. mongodb shell : client terminal, you can perform

all database operation by connecting to mongodb server

1. insert,update,select,delete....

same as mysql terminal

test> REPL.

mysql> REPL.

3. mongodb GUI Compass : its same as mongodb shell, but is graphics based same like phpmyadmin.

phpmyadmin GUI

compass GUI

4. mongodb db cli tools : these are externally installed,

for performing database export and import, and local backup.

2. cloud Based

1. Cluster : complete database on cloud, you can also

get remote connection from local system.

commerecially Paid

2. Atlas : free version (1 GB Space for users free cloud)

How to connect to Mongodb:-

****

1. open cmd

2. mongod : server

3. intial start : error cannot write data into default path : c:/data/db/

4. goto c: mkdir data cd data mkdir db

How to Run mongoclient:-

****

1. make sure your mongod server is running.

2. open another instance of cmd.

3. type mongo

mongo shell>

>

or for 6.x version mongodb then you need mongodb shell direct Run and

your mongoclient will start Running.

>db.version()

Important setting about mongo

****

By default port : 27017

dbPath : C:/data/db/

architecture : 64-bit

host : DESKTOP-N1OGJKU <your-pcname>

ip : 127.0.0.1 => localhost => loopback Ip Address.

How to create a new database :-

****

1. use command : database created then select or if not it will create it.
2. how to get current db : db command
3. by default db : test

Note :: during this `use` command, memory => temporary memory => currentScope

of DB Object.

Note :: what is name of database which is present but not shown in list

Ans :: `test` database

Why mongodb does like this?

1. it is a database system, hence it is more resposiblity of the mongodb

that it do waste any memory.

any database without any collections is not written it will not save them in

secondary memory.

By default Object or Internal db of mongodb

****

1. show databases
1. show dbs
    1. admin 0.000GB Round Figure GB.
    2. config 0.000GB
    3. local 0.000GB
    4. test 0.000GB

db.version() >= 6.0 version : memory will shown exactly in use.

db.version() < 6.0 version : memory will be shown in GB`s

Note :: Since the database size variation in GB, we will never be able to know any major changes in database size.

How to see list of collections in Current Database :-

****

1. use <db-name>

2. show collections

    collection-1

    collection-2

    collection-3

    ...

    collection-n

or

3. db.getCollectionNames()

    ["collection-1","collection-2","collection-3","collection-n"]

show collections : Query or statement

db.getCollectionNames() : Procedure or function or sub-routine calls.

Mongo support two types of statements:-

****

1. Query Statements : lower case without semicolon

2. Procedure Statements : functions calls : camelCase without semicolon.

How to see the list of records in a collection : (select * from <table-name>)

****

db.<db-collection-name>.find() : using this we can find the all records stored in a collection.

How to create a new collection : (create table)

****

db.createCollection("<collection-name>")

How to insert a document-object or a Object

****

Note :: here, one object is equivalent to row, or Record

Types of Insert

****

1. Single Insert : Insert as Object {}

2. Bulk Insert : Insert as Array of Object [{},{},{},{}...]

db.<collection-name>.insert()

How to find the All Inserted Records:-

****

db.emp.find()

{ "_id" : ObjectId("61d863df9cc262aad0e158dd"), "emp_id" : 1001, "emp_name" : "ravi", "class" : "12th" }

How mongodb is different from other database in terms of Automicity(Unique-ness)

****

Note :

ObjectID mongodb it is universally unique, because of unix timestamp.

Unix Timestamp : is total milliseconds invested, day by day from Jan 01 1970 12:00:00 Am at UTC.(Universal timestamp/timezone conversion), as per greenwhich

avg time at London. +5:30.

System 32 Bit : 12 hexadecimal Object Id.

System 64 Bit : 24 hexadecimal Object Id.

Server ---> 32 Bit ---> Mysql --> Pk --> integer 11

Server ---> 64 Bit ---> Mysql --> PK --> Integer 11

Server ---> 64 Bit ---> Mysql --> PK --> BigInt 20

"_id" : Primary Key : ObjectID

Object Id : System Architecture and its minimum Architecture 32 Bit

64 Bit ---> 24 Bit

32 bit ---> 12 Bit

Min Architecture --> 32 Bit ---> 12 Bit

Unitary Method : 1 Bit ObjectId ----------> 12/32 => Ratio => 3/8 Bit

Ratio Architecture to ObjectId -> 3:8 Bit.

Questions:- if server specification is as followed

DD4-3X

T1B SSD 100MBS Rd/wr speed

16 GB Ram

Architecture 16 Bit

OS : unix/centos

Processor : 5.5GHz to 5.89 GHZ octa-8z i-core

What is length of Object Id ?

Ans : 6

ObjectID breakup (12 bit/Bytes)

****

4 Bytes : timestamp : hexadecimal Format 01.01.1970 12.00 am to 21-11-2022 08:30:10 AM +5:30 in second timestamp (4x8x2 Bit)

3 Bytes : Mac-address Id (Machine Access code)

2 Byte : process_id (pid)

3 Byte : Random Numbers

61d863df9cc262aad0e158dd => some sort Hashing Algorithms.

In Application, we can store this "_id" store in session for updating and deleting the data.

update student_Table where stdid = '1001';

db.collection.update({_id:"61d863df9cc262aad0e158dd"})

How to see data after bulk insert : db.emp.find()

****

> db.emp.find() : by default this is projection

> db.emp.find().pretty() : by

> db.emp.find().projection() or db.emp.find() in version < 6.x

otherwise in difference, projection(),pretty(),find() in version > 6.x

In SQL, there are two important Concept

****

1. projection : select * from tablename (Query without where clause or any condition.)

2. selection : selection with condition like where clause, limit, group by.

> db.collection.find()

or

> db.collection.find().projection()

Data show in Horizontal Projection

{ "_id" : ObjectId("61d863df9cc262aad0e158dd"), "emp_id" : 1001, "emp_name" : "ravi", "class" : "12th" }

{ "_id" : ObjectId("61d867cc9cc262aad0e158de"), "emp_id" : 1002, "emp_name" : "shyam", "class" : "11th" }

...

...

...

> db.collection.find().pretty()

data will be show in, vertical format.

db.version() <=6.x

document-1 or object-1

{ "_id" : ObjectId("61d863df9cc262aad0e158dd"),

"emp_id" : 1001,

"emp_name" : "ravi",

"class" : "12th"

},

document-2 or object-2

{ "_id" : ObjectId("61d867cc9cc262aad0e158de"),

"emp_id" : 1002,

"emp_name" : "shyam",

"class" : "11th"

}

db.version() >=6.x

```
[
        { "_id" : ObjectId("61d863df9cc262aad0e158dd"),
        "emp_id" : 1001,
        "emp_name" : "ravi",
        "class" : "12th"
        },
        { "_id" : ObjectId("61d867cc9cc262aad0e158de"),
        "emp_id" : 1002,
        "emp_name" : "shyam",
        "class" : "11th"
        }
]
```

Note ::

1. select * from tablename where columnname = 'value';

2. db.collection.find({key:"value"});

What are keyword in mongodb:-

****

in mongodb, we have keyword like sql syntax(clauses)

they start with $keyword

Eg:-

AND OR => SQL

$and and $or

in => SQL

$in

from => SQL

$from

Note :: this is not valid, in case of operators.

!= => SQL != not EqualTo

$ne

Note :: trick : stdid = value

stdid:value => equal to

operator => $keyword => as a key..

How to fetch records on the basis of some conditions or predicates

****

syntax:

db.<collection-name>.find({key:value})

SQL : select * from collection-name where key='value';

keyword in Mongo : $keyword

db.collection.find({keyname:{$keyword:value}}) : int

db.collection.find({keyname:{$keyword:"value"}}) : string

!= : $ne

>= : $gte

<= : $lte

< : $lt

> : $gt

syntax :

db.<collection-name>.find({key:{$keyword:value}})

SQL : select * from collection-name where key!='value';

what is difference B/w

db.collection.find()

And

db.collection.find({})

db.collection.find() = db.collection.find({})

Q1 : given, emp(#empid,empname,empsal)

|

properties

1. unique, 2. not null,

****

empid | empname| empsal

****

spi-201| ravi | 100

spi-301| anuj | 200

spi-402| ram | 300

2. Table How many primary key

1PK+other Unique

3. What is composite key.

pk+AK

4. what will be query to query to in mongodb for showing only empid

empid => objectId

Ans : select empid from emp;

db.emp.find({},{_id:1});

or

db.emp.find({},{empname:0,empsal:0});

Note :: _id or ObjectId is by default enabled in mongodb

How to fetch records perticular columns

****

syntax: in order use collection name

db.<collection-name>.find({conditions},{<column-names>:<0 or 1>})

0 : hide

1 : show

Mongodb ---> switch ---> All Columns are shown

emp_id : 1

emp_name : 1

class :1

emp_id : 0

emp_name : 0

class :0

Example:-

> db.student.find({},{name:1,_id:1})

{ "_id" : ObjectId("61ee2d1342556a6a994e343a"), "name" : "ravi" }

{ "_id" : ObjectId("61ee3d3442556a6a994e343b"), "name" : "sandeep" }

{ "_id" : ObjectId("61ee3d3442556a6a994e343c"), "name" : "kuldeep" }

{ "_id" : ObjectId("61ee3d3442556a6a994e343d"), "name" : "pawandeep" }

{ "_id" : ObjectId("61ee3d3442556a6a994e343e"), "name" : "Ratandeep" }

Using And and Or Conditions in Mongodb:-

****

Note :: SQL **[AND] & [OR]** clause but mongodb **[AND] & [OR]** they are operator and every operator in mongodb is a $keyword hence we have two,
$keyword : $and , $or.

Eg:-

> db.collection.find()

> SQL : Select * from collection or tablename.

Eg:-

db.employee.find({condition},{selection in columns})

> db.collection.find({$and:})

SQL : select * from collecton where condition-1 = 'something-1' AND condition-2 = 'something-2';

[{key:value},{key:value},{key:value} .... n] => Array Object will be used

value,

$and is a keyword => key

key:value

{$and:ArrayofObject}

|

condition

where clause

db.collection.find({condition})

db.collection.find({$and:ArrayofObject})

Example:-

{$and:[{condition-key-1:'something-1'},{condition-key-2:'something-2'}]}

Proof that, value of multiple condition should be array

****

```
> db.student.find({$and:{name:"ravi",marks:50}})
Error: error: {
	"ok" : 0,
	"errmsg" : "$and must be an array",
	"code" : 2,
	"codeName" : "BadValue"
}
```

Query : for finding record of student with name and marks

****

```
> db.student.find({$and:[{name:"ravi",marks:50}]})
{ "_id" : ObjectId("61ee2d1342556a6a994e343a"), "roll" : 1001, "name" : "ravi", "marks" : 50 }

>
> db.student.find({$and:[{name:"ravi",marks:50}]},{roll:1,name:1})
{ "_id" : ObjectId("61ee2d1342556a6a994e343a"), "roll" : 1001, "name" : "ravi" }
```

Alternate Query : multiple key and value in single Object

****

```
>  db.student.find({$and:[{name:"ravi"},{marks:50}]},{roll:1,name:1})
```

Query with AND Operator with different key and value as different Object

****

```
> db.student.find({$and:[{marks:80},{stdname:"shubham"}]}).pretty()
{
	"_id" : ObjectId("637c36c66f3828aaff5c8afc"),
	"stdid" : 1003,
	"stdname" : "shubham",
	"stdclass" : "10th",
	"marks" : 80
}
```

And Operator with particular Field

****

```
>
db.student.find({$and:[{marks:80},{stdname:"shubham"}]},{stdclass:0}).pretty()
{
	"_id" : ObjectId("637c36c66f3828aaff5c8afc"),
	"stdid" : 1003,
	"stdname" : "shubham",
	"marks" : 80
}
```

And Operator with particular Field

****

>
db.student.find({$and:[{marks:80},{stdname:"shubham"}]},{stdclass:0,_id:0}).pretty()

{ "stdid" : 1003, "stdname" : "shubham", "marks" : 80 }

>

And Operator with other multiple condition and operators.

****

> db.student.find(

... {$and:[

... {marks:{$gte:30}},{marks:{$lte:80}}

...]}

...)

{ "_id" : ObjectId("637af2f36f3828aaff5c8afb"), "stdid" : 1002, "stdname" : "lakluriya", "stdclass" : "10th", "marks" : 80 }

{ "_id" : ObjectId("637c36c66f3828aaff5c8afc"), "stdid" : 1003, "stdname" : "shubham", "stdclass" : "10th", "marks" : 80 }

{ "_id" : ObjectId("637c36c66f3828aaff5c8afd"), "stdid" : 1004, "stdname" : "Nariyal Anna", "stdclass" : "11th", "marks" : 30 }

>

Note :: you cannot make a combination of inclusion and exclusion in mongodb 6.x

1. either exclude column

2. either include columns

3. donot make its combination other wise

MongodbserverError will be raised saying "Projection Donot Allowed to mix inclusion and exclusion"

Eg:-

db.student.find({},{_id:0,name:0,dept:0}) => valid

db.student.find({},{_id:1,name:1,dept:1}) => valid

db.student.find({},{_id:0,name:0,dept:1}) => In-valid => Inc/Ex Error.

db.student.find({},{dept:id}) => valid.

Example of How to combine Multiple Condition with Mongo-db using or

****

> db.student.find({$or:[{roll:1002},{marks:{$gt:50}}]}).pretty()

How to update the record in, mongodb:-

****

IN SQL, update is type of Query, data level or Record Query

Update are of two types

1. single update (with where Clause)

2. bulk update (without where clause)

in mongodb update work on following manner => no of matching records,

if no of matching records equal = 1, update will work

if no of matching is more than one only the top most record will be updated

so inorder to, update multi-record we need specify the multi as true

syntax:

SQL : update table-name set
column=value,column=value,column=value,column=value where

$condition = ...

db.collection.find({conditon},{configuration})

or

we use $set:[{key:value},{key:value},{key:value},{key:value}]

db.collection.update({condition},{$set:{key:value,key:value,key:value}},{multi:true|false})

db.emp.update({emp_id:{$gte:1007}},{$set:{emp_name:"awnish"}},{multi:true})

How to delete the record in, mongodb:-

****

we use db.collection.remove({condition},true|false)

2nd Argument : true or false bydefault false

db.collection.remove({condition}) => n = 5 => delete => 5

or

db.collection.remove({condition},false) => n=5 => delete => 5

db.collection.remove({condition},true) => n = 5 => detele => 1 Top most.

db.collection.remove({condition},false)

db.collection.remove({condition},true)

db.collection.remove({condition})

Both, will work same only if record is one.

for version 6.0 or above you can this Query

****

db.collection.remove({condition},{single:true}) => 5 Record

vvvp

Difference B/w update and delete

****

by default delete deletes all the matching record

but if you want to delete only single matching records

we specify true

by default in update it updates only first record but if you

want to update all the matching records then, we specify multi=true

Upsert in Mongo-db :-

****

upsert = update + insert

if match => 1 update (found or pre-existing)

if not match => 0 insert (Not found)

record_is_found => find()

```
if(record_is_found){
        then update Query()
}else{
        then insert() Query()
}
```

db.collection.find().count() => 0

db.collection.insert()

db.collection.find().count() => > 1 or more

db.collection.update()

Syntax:-

db.collection.update({condition},{key:value,key:value,key:value},{upsert:true })

upsert => insert Query => columns => set columns + where condition + _id

limiting the Query:-

****

limit(n) method : to limit the Query result set.

db.collection.limit(n)

db.emp.find().limit(n)

Result-set : Output of the Query.

Note ::

find()

sort()

pretty()

projection()

limit()

skip()

db.collection.find() => order fix

limit and sort and skip => order not preserve.

SQL Order Query

****

SELECT Query WHERE Clause Group By Having Clause Order By Limit,offset

Mongodb Order

****

db.collection.find().groupBy().having().sort().limit().skip()

Note :: Order not preserved, in case of inserting the record.

Because of No, Schema.

Q1 :- write the Query for sql/mongodb for topmost salary,student topper with max() and without max

max => SQL

SQL:- with max

****

1. select max(marks),studentname from student.

   select max(salary),empname from emp.

2. without max()

   select * from emp order by salary desc

   ****

   raman | 3000

   ravi | 2000 <-----Limit 2

   ram | 1000 <-----Limit 3

   select * from emp order by salary desc limit 1

   ****

   select * from emp order by salary asc

   ram | 1000
   ravi | 2000
   raman | 3000

   db.emp.find().sort({salary:-1}).limit(1)

Example:-

1 |ram | 1000

2 |ravi | 2000

3 |raman | 3000

select max(salary) from student : 3000 ----> 1

Agar 3000 pta hai to Query hogi:-

select * from student where salary = 3000

1. select * from student where salary = (select max(salary) from student)

select max(salary),student.* from student

select * from student where salary != (select max(salary) from student)

Limit 1;

1 |raman | 3000

2 |ravi | 2000

3 |ram | 1000

2 |ravi | 2000

3 |ram | 1000

select * from student where salary != (select max(salary) from student)

AND salary = 2000

2 |ravi | 2000 => *

select max(salary) of student => 3000

select * from student where salary != (select max(salary) from student) AND salary = (select max(salary) from student where salary != (select max(salary) from student));

select * from student where salary != (select max(salary) from student) AND salary > (Select min(salary) from student)

2 |ravi | 2000

3 |ram | 1000

select max(salary) from student => 3000

select min(salary) from student => 1000

select max(salary) from student where salary < select max(salary) from student where salary < (select max(salary) from student where salary = (select max(salary) from student))

Formula:-

Select max(salary) from student where < same Query ...(n-1)Record set.

Select max(salary) from student - 10

< simple 9th

Trick: 10 < 9

10> 9 : true

n+1 > n

n > n-1 => false.

max(n+1) Query < max(n) Query

10 > 9

9+1 > 9

9+1 < 9

Skipping(offset) the Query :-

****

when we run the limit Query Top Record to n record data is fetched.

when we run the skip Query Top Record to n will skipped and Remaining data will be Returned as a result set.

db.collection.find(); //All Records

db.collection.find().skip(n)

How to clear the screen

****

cls

SQL : select * from table order by <column-name> Asc|desc.

Asc (a)=> 1

desc (d) => -1

Note :: By default, it will sorted by asceding with ref to key(column.)

How to order by the data in mongodb

****

since mongodb stores the data in un-structured format, there is no logical arrangement or sequence.

so in-order to manage that, user is resposible to handle the ordering(sort).

db.collection.sort({key:aesc|desc})

aesc = 1

desc = -1

db.collection.find().sort({key:1}) => Ascending order.

db.collection.find().sort({key:-1}) => descending order.

vvvIp********

SQL :- index => 0,1,3,4,5,..... n are not same as array.

[a,b,c,d,e,f,g,h,i,j,k, l]

0 1 2 3 4 5 6 7 8,9,10,11

[8]

[11]->

|

index

arr[9] => j

Example :-

****

for(i=0;i<arr.length;i++){

if(arr[i] == 'j'){

return i;

}

}

select * from products where trending = 1;

|

index

SQL => primary key, unique or index key

index =>

1. clustered.

2. Non-clustered.

Index they have there own, Data structure are of three type

****

1. B tree (slowest)
2. B+ Tree (medium)
3. hash tree (fastest)

https://www.javainuse.com/sql2mongo

How to create the indexes in mongodb

****

Indexes are used or created over the perticular column or key to extract the data or performing the searching

very fast.

in local db or for less number of records creating index will not effect the performance

but definetly if you have cloud server where internet connection and large amount data like 1-lakh record

at time is the constraint(challenge).

in that case you can use the concept of indexes.

How Indexes works:-

****

They organise the groups into different blocks of records called as pages internally in db catalogue

and when ever we want to fetch the records instead of searching it from top to Bottom Line by line

it will direclty jump into that page

Indexes are similar to Book Indexes

Index

****

Unit 1 ----------> Page 1 to 5

Unit 2 ----------> Page 10 to 15

Unit 3 ----------> Page 20 to 25

Unit 4 ----------> Page 100 to 105

Older Version of Mongo:-

****

db.collection.ensureIndex({column_name:1})

1 :create Index

db.collection.ensureIndex() this is replace by createIndex

db.emp.createIndex({emp_id:1})

Joins and Relationships with Multiple Collection set

****

In Mongo we use embedded Data Rather Than Joins :- There is Redundant Duplicated data without Normalisation.

department Table

dept_id department

1. Hr
2. Manager
3. Faculty

Emp Table

empid empname dept_id

1001 Ravi 1

1002 vikas 2

1003 awnish 3

Embedded Data While Inserting we insert entire data as Single Data

Embedded Document Object

{empid:1001,empname:Ravi,dept:{depid:1,department:Hr}}

{empid:1002,empname:vikas,dept:{depid:1,department:Manager}}

{empid:1003,empname:awnish,dept:{depid:1,department:Faculty}}

Pretty Output

****

In order to Increase,display Padding in vertical we pretty function

db.collection.find().pretty()

Concept of Foriegn Key and Primary Key

****

This is not Recommended way to Organise in multiple Object as a collection.

In RDBMS we have two table and we join them by the means of, Joins

using referencial Integrity B/w PK and FK

If you want store the data it is recommended to store the data in form,

in form of Embedded Data Object

we can use lookup Aggregration. to achieve this.

Syntax:

db.collection1.aggregrate({$lookup:{from:"",localField:"",ForiegnField:"",as:""}}
)

As soon as you as clause 2nd Collection will become as field in collection1

Trick:-

****

db.createCollection("<collection>")

db.department.insert();

|

|if collection does not exist then also it will be created

|automatically.

db.dept.insert([

{id:1,name:"Hr"},

{id:2,name:"Faculty"},

{id:3,name:"Admin"},

])

db.emp.insert([

{empid:1001,name:"Ravi",deptid:1},

{empid:1002,name:"Mohan",deptid:1},

{empid:1003,name:"Jainab",deptid:2},

{empid:1004,name:"Tahir",deptid:3},

])

one to Many

```
db.dept.aggregate({$lookup:{from:"emp",localField:"id",foreignField:"deptid",as:"emp"}}).pretty()
```

one emp to one department

```
db.emp.aggregate({$lookup:{from:"dept",localField:"deptid",foreignField:"id",as:"department"}}).pretty()
```

Best Plateforms for best packages

1. Maths & Reasoning : 50%

2. SQL : 30%

3. Programming : 20%

minimum package : 3LPA TO 10LPA |=> Amcat and cocubes (700)

minimum package : 12 LPA to max |=> e-litmus => test PH-Test (780)

minimum package : arc.dev and turing.com or relevel.com

prepinsta..