**Day 5: 30 Mar. 25**

**Mongo Db Database**

Mongo DB is an open source no sql Database which use to store the data using document in the form of JSON.

Limitation of RDBMS database

Like MySQL, Oracle, Db2, Sql Server etc.

1. All RDBMS database are schema base database. Means before storing record in RDBMS we need to create schema mean we need to create table, define number of column table contains with their data types.

Employee

Id Name Salary

int varchar float

1. RDBMS database preferable to store homogeneous elements.

Employee

Id Name Salary PhNumber city

100 Steven 10000 null null

101 Lex 12000 null null

102 John 15000 48477575 null

103 Leena 13000 null Bangalore

1. In RDBMS if we want to store multi attribute value we need to divide into another table.

Sid SName age tech

100 Steven 24 Java, Python

101 John 27 React, Angular, Python

Student table

Course or Technologies table

StudentAndCourse table

1. In RDBMS we store always records in table format.

No SQL Database

In No SQL Database we can store the record any other format apart from table. Like JSON, xml, graph etc

First you need to install the Mongo DB Database.

Window User

In C Drive create the folder as

data which contains db folder

db

To run the mongo DB server we need to open the terminal inside a bin directory

C:\Program Files\MongoDB\Server\7.0\bin

Run the command as

**mongod**

---------------

Non window User

Open the terminal

sudo mkdir -p /data/db

run the command as mongod

sudo chown -R `id -u` /data/db

or

**sudo chown -R mongodb:mongod** /data/db

Open another terminal (window user inside bin directory)

**mongosh** or mongo

Non window user

Open another terminal

mongosh or mongo

In mongo DB first we need to start the server using command as

mongod it use to run the server

mongo old version it is use to run the client terminal

mongosh new version it is use to runt the client terminal

or

we can connect mongo db database using GUI base ie

compass

**MySQL Mongo DB**

**show databases show databases**

it is use to display all database present in your account.

**create database databasename**

**use databasename**; **use databasename;**

in mongo db if database already present it move inside that database. If not present it create and swich inside that database.

**show tables show tables**

**show collections**

In mongo db table is equal to collections. Table is known as collections.

Creating the collection in mongo db.

Mongo Db provided one of the pre defined ie **db.** Which contains lot of pre defined function which help to do some operation on collections or tables.

Mongo Db is **case sensitive**.

**db.createCollection(“Sample”);**

db.createCollection(“sample”);

In mongo DB in a collection we can store more than one documents. **document** is like a **record** in RDBMS database. Document allow use to store the data in **json** format.

Store the document in a collection

db.**Sample**.**insertOne**({**name:”steven”**});

to view the document from a collection

**db.Sample.find();**

**\_id** attribute created for each document to make unique document if we store more than one document with same key-value pairs.

Here \_id is like a PK. If you need to pass custom value for that attribute we can pass but we can’t change \_id attribute name.

Creating Employee collection with few documents

db.Employees.insertMany(

[

{\_id:100,name:"John",age:21,salary:45000,depid:1},

{\_id:101,name:"Bob",age:22,salary:42000,depid:2},

{\_id:102,name:"Charli",age:24,salary:41000,depid:3},

{\_id:103,name:"James",age:21,salary:49000,depid:2},

{\_id:104,name:"Lex",age:22,salary:48000,depid:1},

{\_id:105,name:"Raj",age:26,salary:43000,depid:3},

{\_id:106,name:"Neena",age:29,salary:44000,depid:2},

{\_id:107,name:"Alice",age:28,salary:48000,depid:1},

{\_id:108,name:"Mohan",age:23,salary:47000,depid:2},

{\_id:109,name:"Ajay",age:24,salary:46000,depid:3},

{\_id:110,name:"Vijay",age:25,salary:42000,depid:1}

]);

**Sorting documents from a collection using attribute**

db.Employees.find().sort({age:1}) ascending order number type

db.Employees.find().sort({age:-1}) descending order number type

db.Employees.find().sort({name:1}) ascending order string type

db.Employees.find().sort({name:-1}) descending order string type

limit() this function is use to top document from a collection

db.Employees.find().limit(2);

sort() with limit()

**db.Employees.find().sort({salary:-1}).limit(2);**

**skip()** : this function is use skip number for document

**db.Employees.find().skip(4);**

**filter the document like where clause**

db.Employee.find({conditions});

using equal

**db.Employees.find({\_id:100}); \_id condition**

**db.Employees.find({name:'Neena'}); name condition**

**db.Employees.find({salary:42000}); salary condition**

**Mongo DB provided lot of operator all those operator start with pre-fix $ followed by operator name.**

**apply relational operator**

**db.Employees.find({salary:{$gt:44000}}); >**

**db.Employees.find({salary:{$lt:44000}}); >=**

**db.Employees.find({salary:{$gte:44000}}); <**

**db.Employees.find({salary:{$lte:44000}}); <=**

**db.Employees.find({salary:{$eq:44000}}); =**

**db.Employees.find({salary:{$ne:44000}}); !=**

**and/or operator**

**db.Employees.find({$and:[{\_id:100},{name:'John'}]}); and condition**

**db.Employees.find({$or:[{\_id:101},{name:'John'}]}); or condition**

**retrieve specific fields from collections**

**db.Employees.find({conditions},{specificfield})**

**db.Employees.find({},{name:1}); retrieve all name as well as \_id field**

**db.Employees.find({},{name:1,\_id:0}); retrieve all names from document**

**db.Employees.find({},{name:1,salary:1,\_id:0}); retrieve name and alary from document**

**update documents field using the conditions**

**to update we need to use $set operator**

**updateOne : if apply the condition with \_id fields then we need to use updateOne**

**updateMany : if apply condition apart form unique field like \_id then you need to use updateMany**

**db.Employees.updateOne({condition},{updateFieldWithOperator})**

**db.Employees.updateOne({\_id:100},{$set:{salary:55000}});**

**db.Employees.updateMany({depid:1},{$set:{salary:55000}});**

**update with many condition**

**like $and / $or**

**db.Employees.updateMany({$and:[{\_id:101},{depid:2}]},{$set:{salary:45000,age:23}});**

**db.Employees.updateMany({$or:[{\_id:101},{depid:2}]},{$set:{salary:45000,age:23}});**

**Delete document using conditions**

**db.Employees.deleteOne({\_id:110}); to delete using unique field**

**db.Employees.deleteMany({depid:2}); to delete many documents.**

**$exists**

**Display those document which contains particular field or columns**

**db.Sample.find({fname:{$exists:true}});**

**db.Sample.find({age:{$exists:true}});**

**Adding specific field for particular document or specific document.**

**Adding new field for all documents.**

**db.Employees.updateMany({},{$set:{designation:'employee'}});**

**updating with conditions**

**db.Employees.updateMany({\_id:102},{$set:{designation:'Tester'}});**

**remove the field from all document with specific documents.**

**remove all fields from document ie depid**

**db.Employees.updateMany({},{$unset:{depid:1}});**

**remove particular field from a document which contains \_id as 100**

**db.Employees.updateOne({\_id:100},{$unset:{age:1}});**

**storing array value in a fields.**

db.Students.insertMany([

{\_id:1,sname:"John",age:21,result:true,skillSet:["Java","Python"]},

{\_id:2,sname:"Bob",age:23,result:false, skillSet:["Java"]},

{\_id:3,sname:"Charli",age:26,result:true, skillSet:["Python"]},

{\_id:4,sname:"Jamesh",age:25,result:false,skillSet:["HTML","CSS","JavaScript"]},

{\_id:5,sname:"Steven",age:23,result:true,skillSet:[]}

]);

**Searching specific fields from array field attribute**

db.Students.find({skillSet:'Java'});

**adding new value for array fields**

db.Students.updateMany({},{$push:{skillSet:"React JS"}});

**adding new value for array field with conditions**

db.Students.updateMany({result:true},{$push:{skillSet:"Angular Framework"}});

remove array field value

**db.Students.updateMany({skillSet:["React JS"]},{$pop:{skillSet:1}});**