**MONGODB**

SQL: Structure query language

RDBMS: Relational database management system

Database: collection of data in a structured way

**What is noSql?**

A nosql database is a key value database

It is not is the form of table.

Different types of nosql are: MongoDB, Amazon DocumenrDb, Google Datastore, Amazon DynamoBd ect.

Extremely useful,powerful and high performance database in largr big data applications, large distributed network architure apps ect.

**What is mongoDB:**

Open and free source,cross platform(any platform can run),

It used Json-like document with schema

**MongoDb Storage or Size we can store:**

The maximum size an individual document can be in MongoDB is **16MB with a nested depth of 100 levels**

**The maximum size of database that we can store is that 1TB to 32TB.**

**What Mongodb is not?**

* It is not a RDMS
* Doesnot have any concept of Joins.
* Not tough or complicated.

**Languages supports:**

PHP,Node Js, Python, java, c#, C++

**Why learn MongoDB?**

* It doesnot contain any schema and does not contain any relational database.
* And easy to under stand.
* It gives the quick response while we search any data.

**IN mongo db :-**

* Database
* Collections
* Documents:

In Document can have any data type – long as it is valid mongodb data type

It is simply key-value pair data

Example: {

“firstname”: “kiran”,

“lastname”:”Kollana”,

“Email”: “Kollanakiran@gamil.com”

},

{

“firstname”: “alekhya”,

“lastname”:”annamraju”,

“Email”: [annamrajualekhya@gamil.com](mailto:annamrajualekhya@gamil.com)

};

**Collections:**

It’s a set of documents

Can have any number of documents

Documents can have any dynamic schema

They can be same or different

No join concept

**Database:**

Sing collection or more collections

**Queries :**

**Creating and dropping data base:**

* Show database; :- it show the data in our database.
* Use databasename(CMR) :use to switch into CMR data base.
* Db: it shows in which database we are in.
* Db.dropDatabase(); :-it used to deleted the database in which we are present.
* To drop a database, first we need to select the DB

---use (database – name>

---db.dropDatabase();

**Creating and dropping collections:**

**Db.createCollection(“< name of collection”);:-creating collections in database.**

**Db.collectionName.drop():- dropped the collection from the database.**

**18/11/22**

**Inserting data:**

**Db.CollectionName.insertOne({}):- it is for inserting data into collection**

**Example:**  **db.Ram.insertOne({ title: " post title",**

**body: "Body of post",**

**category:"News"**

**,date: Date()})**

Db.collectionName.insertMany([**{ title: " post title",**

**body: "Body of post",**

**category:"News"**

**,date: Date()},**

**({ title: " post title",**

**body: "Body of post",**

**category:"News",**

**date: Date()}])**

**To find the Data in the data base:** db.collectionName.find()

**Find it with the single key and value**: db.collectioName.find({Key:”value})

**Find it with the keys** : db.collectionName.find({}, {title: 1, date: 1})

It gives only those fields which we are mentioned in The key.

Empty flower brackets gives the object id.

O/P: [

{

\_id: ObjectId("63771b0e9c297b2f49731ed7"),

title: ' post title',

date: 'Fri Nov 18 2022 11:11:34 GMT+0530 (India Standard Time)'

},

{

\_id: ObjectId("63771dc39c297b2f49731ed8"),

title: 'post title 2',

date: 'Fri Nov 18 2022 11:23:07 GMT+0530 (India Standard Time)'

},

{

\_id: ObjectId("63771dc39c297b2f49731ed9"),

title: 'post title 3',

date: 'Fri Nov 18 2022 11:23:07 GMT+0530 (India Standard Time)'

}

]

**Update the document:**

**To update an existing document we can use the updateone () or updateMany() Method.**

**For update :**

**db.Ram.update({"title":"post title 2"},{$set:{"Ram":"8GB","processor":"Intel i9 12th gen"}})**

**one more update in the Same : db.Ram.updateMany({"Ram": "8GB"},{$set:{"ram":"15Gb"}})**

**Delete concept:** db.Ram.deleteOne({title:"post title 3"})

**Operators:** db.products.find({"tax":{$gte:"1000"}}) : greater then Or equal to.

db.products.find({"tax":{$ne:"1000"}}) : not equal then.

db.products.find({"tax":{$eq:"1000"}}) : greater then.

db.products.find({"tax":{$lt:"1000"}}) : less then.

db.products.find({"tax":{$gt:"1000"}}) : greater then.