**MongoDB: 16-06-23**

* Non structural database
* Different data stored in the same row
* By default keep empty when data is not provided
* Case sensitive
* Data come from two ways:

1. **JASON**: python dictionary (key value pair data)
2. **CSV**: excel sheet data

**mongoDB or mongoD Difference:**

1. **MongoD:**
2. Mongo demon
3. Physical GUI
4. Drag and drop process happen
5. Having own database
6. **Mongo:**
7. Command line utility
8. Mongo shell
9. Through command send the data to database

|  |  |  |
| --- | --- | --- |
| **Sr no** | **SQL TERM** | **MONGO TERM** |
| 1 | database | database |
| 2 | table | collection |
| 3 | rows | documents |
| 4 | columns | fields |

**Commands used :**

1. Check database list: **show dbs**
2. Create new database and switch to another dbs: **use dbsname**

Condition: if we want to see new create dbs then we must have collection in the dbs otherwise dbs is not visible while run show dbs commands

1. Delete database: **db.deletedarabasename()**
2. Create collection: **db.createCollection(‘collection name’)**
3. Delete collection**: db.collectionname.drop()**
4. Insert data in to collection:
5. **db.collectionname.insertone({object})** ….object give in key value pair format

example:

db.shourya.insertOne({

'Name':'vijay',

'Age':35,

'Salary':78000,

'Language':['Python','SQL'],

'Rating':3

})

Output: "acknowledged" : true,

"insertedId" : ObjectId("648bfe1275715637be532757")

1. **db.collectionname.insertmany([{object1,object2}])**

db.shourya.insertMany([

{

'Name':'kb',

'Age':27,

'Salary':89000,

'Language':['Python','SQL','Java'],

'Rating':3.5

},

{

'Name':'shital',

'Age':27,

'Salary':89000,

'Language':['Python','SQL','Java'],

'Certificate':'IBM',

'Rating':3.5

}

])

1. show current database working**: db**
2. show collections in database**: show collections ……….**show collections list in db
3. search data :
4. **find():** finding the data from collection in mongodb

1. **db.collectionname.find( {key value pair in collection})** ………display all data from collection which are matching

2. **db.collectionname.find() ………….**fetch all datafrom collection

1. **findone():**

**db.collectionname.findOne( {key value pair in collection}) ………**display only first entry element from the list of data which are match with condition we are given

1. **limit:**

**db.collectionname.find().limit(no of records want to see)**

1. **count:**

**db.collectionname.find().count() ….**no of records in collection show

1. **update the row:**

**db.colletionname.update({key, value pair},{object which we want to update})**

db.shourya.update({'salary':89000}, …………..old salary

{

'Name':'kb',

'Age':27,

'Salary':99000, …………..updated salary

'Language':['Python','SQL','Java'],

'Rating':3.5

})

1. delete the row:

**db.collectionname.remove({one key value pair in collection})** ……….specific data delete from collection by given just one key value pair

1. **opearator:**
2. **increment:**

**--**used on only on mathematical value

**--db.collectionname.update({key value pair}, {$inc:{mathematical key value pair}})** ……increment value given in second pair

**---**example: db.shourya.update({'Name':'kb'},{$inc:{'Age':3}})

1. **rename:**

**--- db.collectionname.update({key value pair}, {$rename:{key which we want to chanege}})**

**--**example: db.shourya.update({'Name':'kb'},{$rename:{'Salary':'total salary'}}) …..salary change to total salary

1. **less than and less than equal to:**

**--db.collectionname.find({‘key’: {$lt: value}})**

**--**example: db.shourya.find({'Age':{$lt:25}}) ….age less than 25 show data

**-- db.collectionname.find({‘key’: {$lte: value}})**

1. **greater than and greater than equal to:**

**-- db.collectionname.find({‘key’: {$gt: value}})**

**--**example: db.shourya.find({'Age':{$gt:25}}) ….age greater than 25 data show

**-- db.collectionname.find({‘key’: {$gte: value}})**