**To clear the screen:**

cls

**To show the Databases:**

show dbs

test> show dbs

admin 40.00 KiB

config 108.00 KiB

local 72.00 KiB

**to switch to datadase:**

test> use admin

switched to db admin

admin> use testing1

switched to db testing1

**To Create the Collection**

testing1> db.createCollection("testings")

{ ok: 1 }

testing1>

**To Drop/Delete the Database:**

testing1> show dbs

admin 40.00 KiB

config 108.00 KiB

local 72.00 KiB

testing1 8.00 KiB

**testing1> db.dropDatabase()**

{ ok: 1, dropped: 'testing1' }

testing1>

after this:

**testing1> show dbs**

admin 40.00 KiB

config 108.00 KiB

local 72.00 KiB

testing1>

**Insert into the Collection**

testing1> db.testings.insertOne({name:"Awais Ali", age:23,gpa:3.6})

{

acknowledged: true,

insertedId: ObjectId('669e0be311586d1e11c4e49b')

}

testing1>

**Find(): to see the total number of collection:**

testing1> db.testings.find()

[

{

\_id: ObjectId('669e0be311586d1e11c4e49b'),

name: 'Awais Ali',

age: 23,

gpa: 3.6

}

]

testing1>

**To Insert More than one collection:**

testing1> db.testings.insertMany([{name:'Sherry', age:20, gpa:3.5}, {name:'Jahanzaib', age:26, gpa:4.0}, {name:'Talal', age:15, gpa:2.5}])

{

acknowledged: true,

insertedIds: {

'0': ObjectId('669e0cc811586d1e11c4e49c'),

'1': ObjectId('669e0cc811586d1e11c4e49d'),

'2': ObjectId('669e0cc811586d1e11c4e49e')

}

}

**diff Datatype in Mongo:**

name

"Soban Shafeeq" **String**

age

22 **Int**

gpa

2.2  **double**

fullTime

false **boolean**

registerDate

2024-07-22T07:50:13.974+00:00 **Date()**

gradutionDate

null **Null**

courses

**Array** (3)

0

"English"

1

"Urdu"

2

"Computer"

address

**Object**

street

"Dhoke khaba"

city

"islamabad"

**Sort the Data by Alphabets**

testing1> db.testings.find().sort({name:1})

[

{

\_id: ObjectId('669e0be311586d1e11c4e49b'),

name: 'Awais Ali',

age: 23,

gpa: 3.6

},

{

\_id: ObjectId('669e0cc811586d1e11c4e49d'),

name: 'Jahanzaib',

age: 26,

gpa: 4

},

**Limit() method: to limit the number of document:**

testing1> db.testings.find().limit(2)

[

{

\_id: ObjectId('669e0be311586d1e11c4e49b'),

name: 'Awais Ali',

age: 23,

gpa: 3.6

},

{

\_id: ObjectId('669e0cc811586d1e11c4e49c'),

name: 'Sherry',

age: 20,

gpa: 3.5

}

]

**find() with parameter:**

testing1> db.testings.find({gpa:4})

[

{

\_id: ObjectId('669e0cc811586d1e11c4e49d'),

name: 'Jahanzaib',

age: 26,

gpa: 4

}

]

**To get only specfice field using projection:**

testing1> db.testings.find({}, {name:true, \_id:false})

[

{ name: 'Awais Ali' },

{ name: 'Sherry' },

{ name: 'Jahanzaib' },

{ name: 'Talal' },

{ name: 'Soban Shafeeq' }

]

testing1>

**find({Query}, {Projection}): query act as where clause and projection work as select clause**:

**Updating the Documents:**

db.collectionName.updateOne({filter}, {Update})

**filter use for criteria**

**update what are the modification**

**replace the value**

testing1> db.testings.updateOne({name:"Sherry"},{$set: {fullTime:true}})

{

acknowledged: true,

insertedId: null,

matchedCount: 1,

modifiedCount: 1,

upsertedCount: 0

}

**UpdateMany(filter,Update);**

testing1> db.students.updateMany({}, {$set: {fulltime:truullTime:true}})

true}})

{

acknowledged: true,

insertedId: null,

matchedCount: 0,

modifiedCount: 0,

upsertedCount: 0

}

**filter empty means all the doucment in the collection**

**Exists(): check if the field exist: if not exists add the field to the documents**

testing1> db.testings.updateMany({Gender:{$exists:false}},{$set:{Gender:'Male'}})

{

acknowledged: true,

insertedId: null,

matchedCount: 5,

modifiedCount: 5,

upsertedCount: 0

}

testing1> db.testings.find()

[

{

\_id: ObjectId('669e0be311586d1e11c4e49b'),

name: 'Awais Ali',

age: 23,

gpa: 3.6,

fulltime: true,

Gender: 'Male'

},

{

\_id: ObjectId('669e0cc811586d1e11c4e49c'),

name: 'Sherry',

age: 20,

gpa: 3.5,

fullTime: true,

fulltime: true,

Gender: 'Male'

},

**Deleting the Documents:**

**DeleteOne()**

testing1> db.testings.deleteOne({name:"Sherry"})

{ acknowledged: true, deletedCount: 1 }

testing1>

**DeleteMany(): is delete more than one record/documents**

Delete document with the Missing field

testing1> db.testing.deleteMany({registerDate:{$exists:fa

false}})

{ acknowledged: true, deletedCount: 0 }

testing1>

**Comparison Operator:**

start with the $ sign

**$ne (Not Equal)**

testing1> db.testings.find({name:{$ne:"Sherry"}})

[

{

\_id: ObjectId('669e0be311586d1e11c4e49b'),

name: 'Awais Ali',

age: 23,

gpa: 3.6,

fulltime: true,

Gender: 'Male'

},

**$lt (Less Than)**

testing1> db.testings.find({age:{$lt:20}})

[

{

\_id: ObjectId('669e0cc811586d1e11c4e49e'),

name: 'Talal',

age: 15,

gpa: 2.5,

fulltime: true,

Gender: 'Male'

}

]

**$Lte (Less than equall)**

**$gt (Greater than)**

**$gte (Greater than equal to)**

**Values Between two points**

testing1> db.testings.find({gpa:{$gte:2.5, $lte:3.6}})

[

{

\_id: ObjectId('669e0be311586d1e11c4e49b'),

name: 'Awais Ali',

age: 23,

gpa: 3.6,

fulltime: true,

Gender: 'Male'

},

{

\_id: ObjectId('669e0cc811586d1e11c4e49e'),

name: 'Talal',

age: 15,

gpa: 2.5,

fulltime: true,

Gender: 'Male'

},

**In operator**:

**$in (In it)**

one of the value in the array in found in the records

testing1> db.testings.find({name:{$in:["Awais Ali", "Ali","Jahanzaib", "Sherry"]}})

[

{

\_id: ObjectId('669e0be311586d1e11c4e49b'),

name: 'Awais Ali',

age: 23,

gpa: 3.6,

fulltime: true,

Gender: 'Male'

},

{

\_id: ObjectId('669e0cc811586d1e11c4e49d'),

name: 'Jahanzaib',

age: 26,

gpa: 4,

fulltime: true,

Gender: 'Male'

},

**$nin (Not in it)**

**Logical Operator in MongoDb**

Return data that are evaluate either true or false

**$and**

testing1> db.testings.find({$and:[{fullTime:true},{age:{$lte:22}}]} )

[

{

\_id: ObjectId('669e0cc811586d1e11c4e49c'),

name: 'Sherry',

age: 20,

gpa: 3.5,

fullTime: true,

fulltime: true,

Gender: 'Male'

}

]

**$or**

**$nor**

**$not**

**Indexes:**

Explain("ExecutionStats")

Stats about the specifice document

db.collectionsName.createindex(based on what condition)

**getIndexes()**

testing1> db.testings.getIndexes()

[

{ v: 2, key: { \_id: 1 }, name: '\_id\_' },

{ v: 2, key: { name: 1 }, name: 'name\_1' }

]

**Drop the index**

testing1> db.testings.dropIndex("name\_1");

{ nIndexesWas: 2, ok: 1 }

testing1>

indexes slowdown the insertion and deletion!

take up much memory

when indexes when when you are more working in searching the stuff

**How to Create new collections:**

testing1> db.createCollection("test2", {capped:true, size:10000000, max:1000}, {autoIndexId:false})

{ ok: 1 }

testing1> show collections

test2

testings