**Database Setup**

|  |
| --- |
| PS C:\Users\ > mongosh  Current Mongosh Log ID: 65faf4edd00fefe8d1dd02a8  Connecting to: mongodb://127.0.0.1:27017/?directConnection=true&serverSelectionTimeoutMS=2000&appName=mongosh+2.0.2  Using MongoDB: 7.0.2  Using Mongosh: 2.0.2  mongosh 2.1.1 is available for download: <https://www.mongodb.com/try/download/shell>  For mongosh info see: <https://docs.mongodb.com/mongodb-shell/>  The server generated these startup warnings when booting  2024-03-15T18:28:55.484+05:30: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted |

**Collection Creation:** It is used to create collections in mongodb .

**Syntax:** db.createCollection(name,options)

|  |
| --- |
| test> use myDatabase  switched to db myDatabase  myDatabase> db.createCollection("users")  { ok: 1 } |

**Document Insertion:** It is used to insert a document or documents into a collection.

**Syntax:** db.collection.insert() – This is used to insert single document

db.collection.insertMany() – It helps to insert multiple documents in ordered insert. Also all the documents do not need to have same fields.

|  |
| --- |
| myDatabase> db.users.insertMany([ {name: "sheldon" , email:" sheldoncooper@gmail.com", age: 10}, {name:"Missy",email:"missycooper@gmail.com", age: 5}, {name:"Meemaw" , email:"meemaw8@gmail.com", age:61}])  {  acknowledged: true,  insertedIds: {  '0': ObjectId("65faf79cd00fefe8d1dd02a9"),  '1': ObjectId("65faf79cd00fefe8d1dd02aa"),  '2': ObjectId("65faf79cd00fefe8d1dd02ab")  }  } |

**Querying**

**Syntax:** db.collection\_name.find() – It helps to retrive the documents in a specific collection.

In a query operation, we can also use criteria or conditions which can be used to retrieve specific data from the database.

**(i) All users from the users collection**

|  |
| --- |
| myDatabase> db.users.find({})  [  {  \_id: ObjectId("65faf79cd00fefe8d1dd02a9"),  name: 'sheldon',  email: ' sheldoncooper@gmail.com',  age: 10  },  {  \_id: ObjectId("65faf79cd00fefe8d1dd02aa"),  name: 'Missy',  email: 'missycooper@gmail.com',  age: 5  },  {  \_id: ObjectId("65faf79cd00fefe8d1dd02ab"),  name: 'Meemaw',  email: 'meemaw8@gmail.com',  age: 61  }  ] |

**(ii)Users with an age greater than or equal to 30.**

**Syntax:** db.collection\_name.find({< key > : {$gte : < value >}})

|  |
| --- |
| myDatabase> db.users.find({age:{$gte:30}}  )  [  {  \_id: ObjectId("65faf79cd00fefe8d1dd02ab"),  name: 'Meemaw',  email: 'meemaw8@gmail.com',  age: 61  }  ] |

**Update Operation**

**Syntax:** db.collection\_name.updateOne() (or) db.collection\_name.updateMany() —This is used to update all the documents present in the collection that matches the provided queries.

|  |
| --- |
| myDatabase> db.users.updateOne({email:"meemaw8@gmail.com"},{$set:{age: 36}})  {  acknowledged: true,  upsertedCount: 0  } |

**Deletion Operation**

[db.collection.remove()](https://www.mongodb.com/docs/v3.2/reference/method/db.collection.remove/#db.collection.remove) – Used to delete all documents that matches a specified filter

[db.collection.deleteOne()](https://www.mongodb.com/docs/v3.2/reference/method/db.collection.deleteOne/#db.collection.deleteOne) – Used to delete a single document

db.collection.deleteMany() – Used to delete documents that matches a specified filter

|  |
| --- |
| myDatabase> db.users.deleteOne({email:"missycooper@gmail.com"})  { acknowledged: true, deletedCount: 1 } |

**Index Operation**

**Syntax:** db.createIndex(<field>)

It helps to create a unique index on the \_id field during the creation of a collection. The \_id index prevents clients from inserting two documents with the same value for the \_id field. You cannot drop this index.

|  |
| --- |
| myDatabase> db.users.createIndex({email:1})  email\_1  myDatabase> db.users.find({})  [  {  \_id: ObjectId("65faf79cd00fefe8d1dd02a9"),  name: 'sheldon',  email: ' sheldoncooper@gmail.com',  age: 10  },  {  \_id: ObjectId("65faf79cd00fefe8d1dd02ab"),  name: 'Meemaw',  email: 'meemaw8@gmail.com',  age: 36  }  ]  myDatabase> |