190031249 **P.MOHITH**

EXPERIMENT-11
PRE-LAB

1. **What are NoSQL databases? What are the different types of NoSQL databases?**

NoSQL Database is a non-relational Data Management System, that does not require a fixed schema. It avoids joins, and is easy to scale. The major purpose of using a NoSQL database is for distributed data stores with humongous data storage needs. NoSQL is used for Big data and real-time web apps.

NoSQL database stands for "Not Only SQL" or "Not SQL." Though a better term would be "NoREL", NoSQL caught on.

Here are the four main types of NoSQL databases:

- Document databases
- Key-value stores
- Column-oriented databases
- Graph databases

2. **While creating Schema in MongoDB what are the points need to be taken in consideration?**

Points need to be taken in consideration are:-

- Design your schema according to user requirements
- Combine objects into one document if you use them together. Otherwise, separate them
- Do joins while write, and not when it is on read
- For most frequent use cases optimize your schema
- Do complex aggregation in the schema

3. **What is the syntax to create a collection and to drop a collection in MongoDB?**

To Create a new collection in MongoDB:

- **Syntax 1:**
  >db.createCollection("collection name");
  example:
  >db.createCollection("cse");
  >db.createCollection("eee");
- **Syntax 2:**

  >db.createCollection("collectionname",{capped:true, size:number (bytes), max: number });

this is used to set the collections as a CAPPED COLLECTION.
when creating capped collection, second argumnet is mandatory;
capped:true -->pupose is, u r settting this collections as Capped by given value as "true"
size:number -->purpose is, setting the maximum size of the collection
max:number -->pupose is, to set the maximum number documnets that can hold in this collection.

example:

```
>db.createCollection("cse",{capped:true, size: 500000, max:5});
```

To delete a collection in Mongodb

Basic syntax:

```
db.COLLECTION_NAME.drop()
```

4. Explain what are indexes in MongoDB?

An index in MongoDB is a special data structure that holds the data of few fields of documents on which the index is created. Indexes improve the speed of search operations in database because instead of searching the whole document, the search is performed on the indexes that holds only few fields. On the other hand, having too many indexes can hamper the performance of insert, update and delete operations because of the additional write and additional data space used by indexes.

5. Mention what is the basic syntax to use index in MongoDB?

Basic Syntax:

```
db.collection_name.createIndex({field_name: 1 or -1})
```

The value 1 is for ascending order and -1 is for descending order.

INLAB

Construct Queries using MongoDB on Case Study 9 (MILITARY DATABASE)

1. **Create a query to create a collection called Army with given attributes defined in skilling session 4**

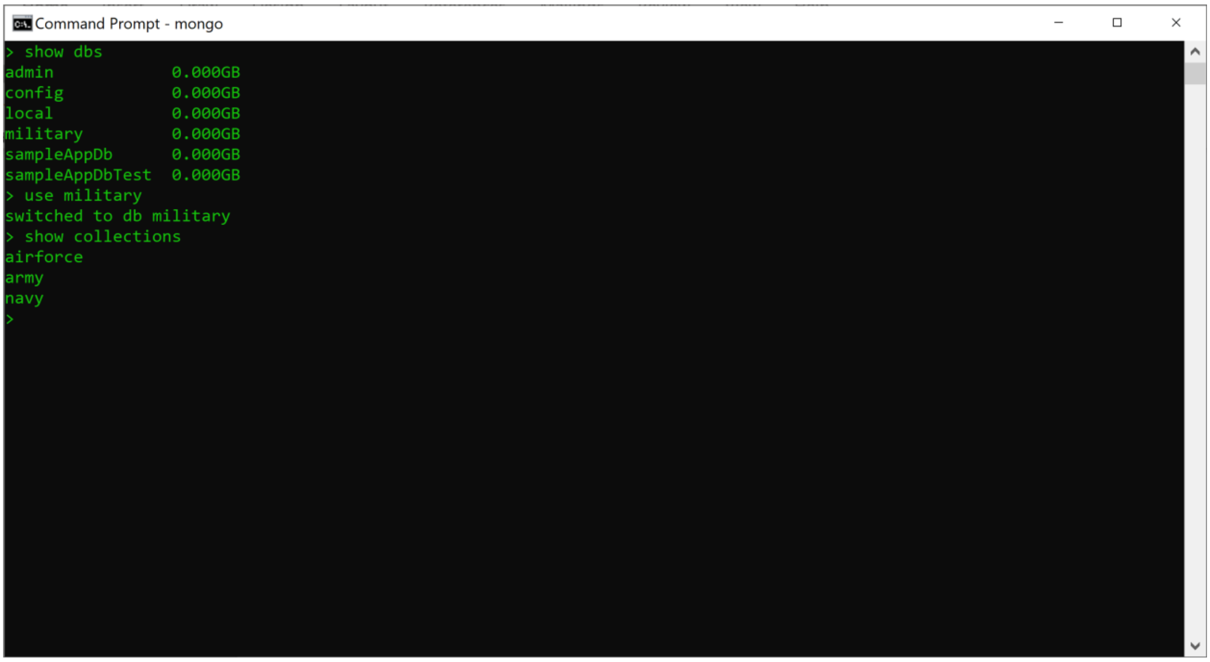
Basic Syntax:
>db.createCollection("collection name");
Here Collection Name is Army So,
>db.createCollection("army");

2. **Create a query to construct a collection called Navy with given attributes defined in skilling session 4**

Basic Syntax:
>db.createCollection("collection name");
Here Collection Name is Navy So,
>db.createCollection("navy");

3. **Create a query to create a collection with the Airforce details with given attributes defined in skilling session 4**

Basic Syntax:
>db.createCollection("collection name");
Here Collection Name is Airforce So,
>db.createCollection("airforce");



```
> show dbs
admin            0.000GB
config           0.000GB
local            0.000GB
military         0.000GB
sampleAppDb      0.000GB
sampleAppDbTest  0.000GB
> use military
switched to db military
> show collections
airforce
army
navy
>
```

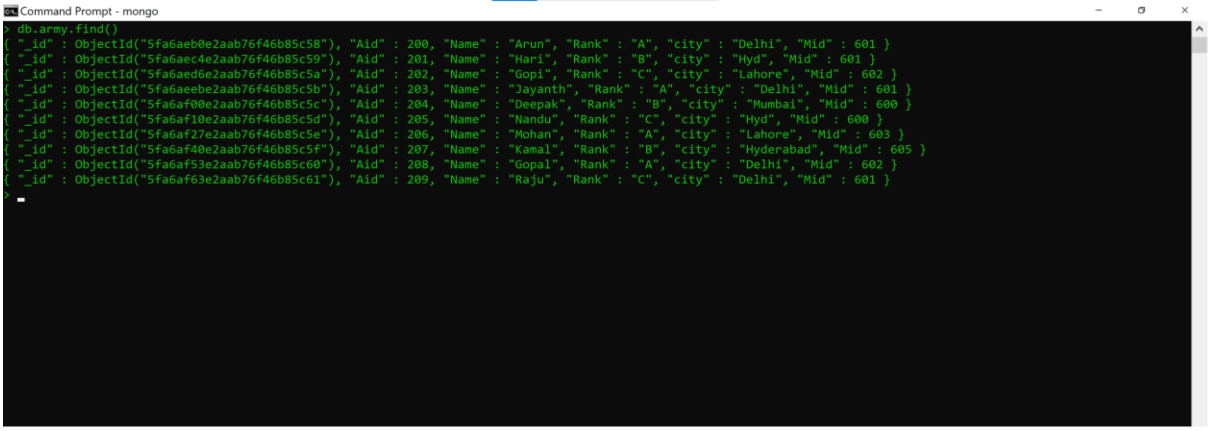
4. Construct a query to insert data into above created collections.

Basic Syntax:

>db.COLLECTION_NAME.insert(document)

Here for Army

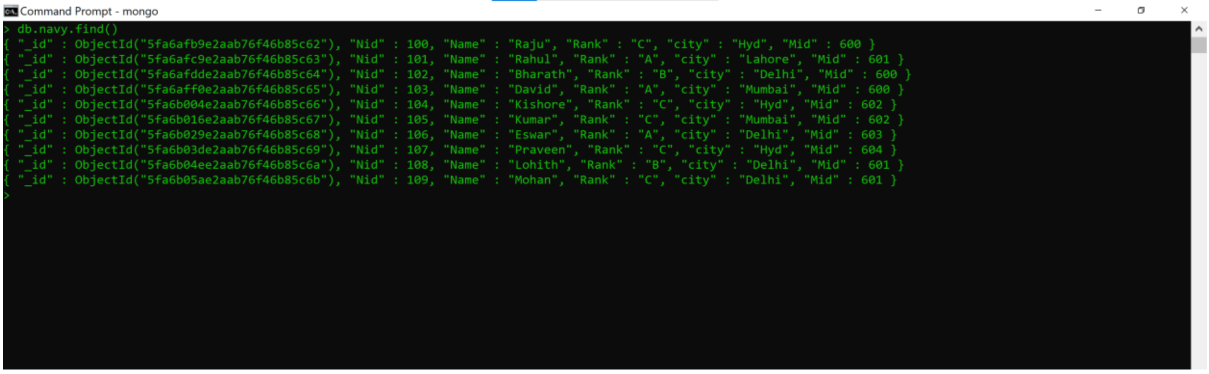
>db.army.insert({Aid:200,Name:”Arun”,Rank:”A”,city:”Delhi”,Mid:601})



```
> db.army.find()
{ "_id" : ObjectId("5fa6aeb0e2aab76f46b85c58"), "Aid" : 200, "Name" : "Arun", "Rank" : "A", "city" : "Delhi", "Mid" : 601 }
{ "_id" : ObjectId("5fa6aec4e2aab76f46b85c59"), "Aid" : 201, "Name" : "Hari", "Rank" : "B", "city" : "Hyd", "Mid" : 601 }
{ "_id" : ObjectId("5fa6aed6e2aab76f46b85c5a"), "Aid" : 202, "Name" : "Gopi", "Rank" : "C", "city" : "Lahore", "Mid" : 602 }
{ "_id" : ObjectId("5fa6aeebe2aab76f46b85c5b"), "Aid" : 203, "Name" : "Jayanth", "Rank" : "A", "city" : "Delhi", "Mid" : 601 }
{ "_id" : ObjectId("5fa6af00e2aab76f46b85c5c"), "Aid" : 204, "Name" : "Deepak", "Rank" : "B", "city" : "Mumbai", "Mid" : 600 }
{ "_id" : ObjectId("5fa6af10e2aab76f46b85c5d"), "Aid" : 205, "Name" : "Nandu", "Rank" : "C", "city" : "Hyd", "Mid" : 600 }
{ "_id" : ObjectId("5fa6af27e2aab76f46b85c5e"), "Aid" : 206, "Name" : "Mohan", "Rank" : "A", "city" : "Lahore", "Mid" : 603 }
{ "_id" : ObjectId("5fa6af40e2aab76f46b85c5f"), "Aid" : 207, "Name" : "Kamal", "Rank" : "B", "city" : "Hyderabad", "Mid" : 605 }
{ "_id" : ObjectId("5fa6af53e2aab76f46b85c60"), "Aid" : 208, "Name" : "Gopal", "Rank" : "A", "city" : "Delhi", "Mid" : 602 }
{ "_id" : ObjectId("5fa6af63e2aab76f46b85c61"), "Aid" : 209, "Name" : "Raju", "Rank" : "C", "city" : "Delhi", "Mid" : 601 }
> 
```

Here for Navy

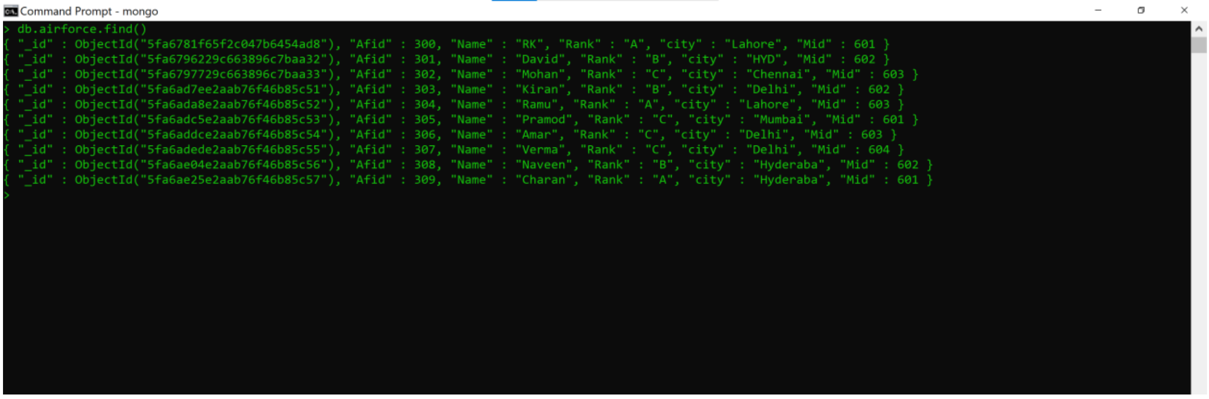
>db.navy.insert({Nid:100,Name:”Raju”,Rank:”C”,city:”Hyderabad”,Mid:600})



```
> db.navy.find()
{ "_id" : ObjectId("5fa6afb9e2aab76f46b85c62"), "Nid" : 100, "Name" : "Raju", "Rank" : "C", "city" : "Hyd", "Mid" : 600 }
{ "_id" : ObjectId("5fa6afc9e2aab76f46b85c63"), "Nid" : 101, "Name" : "Rahul", "Rank" : "A", "city" : "Lahore", "Mid" : 601 }
{ "_id" : ObjectId("5fa6afdde2aab76f46b85c64"), "Nid" : 102, "Name" : "Bharath", "Rank" : "B", "city" : "Delhi", "Mid" : 600 }
{ "_id" : ObjectId("5fa6aff0e2aab76f46b85c65"), "Nid" : 103, "Name" : "David", "Rank" : "A", "city" : "Mumbai", "Mid" : 600 }
{ "_id" : ObjectId("5fa6b004e2aab76f46b85c66"), "Nid" : 104, "Name" : "Kishore", "Rank" : "C", "city" : "Hyd", "Mid" : 602 }
{ "_id" : ObjectId("5fa6b016e2aab76f46b85c67"), "Nid" : 105, "Name" : "Kumar", "Rank" : "C", "city" : "Mumbai", "Mid" : 602 }
{ "_id" : ObjectId("5fa6b029e2aab76f46b85c68"), "Nid" : 106, "Name" : "Eswar", "Rank" : "A", "city" : "Delhi", "Mid" : 603 }
{ "_id" : ObjectId("5fa6b03de2aab76f46b85c69"), "Nid" : 107, "Name" : "Praveen", "Rank" : "C", "city" : "Hyd", "Mid" : 604 }
{ "_id" : ObjectId("5fa6b04ee2aab76f46b85c6a"), "Nid" : 108, "Name" : "Lohith", "Rank" : "B", "city" : "Delhi", "Mid" : 601 }
{ "_id" : ObjectId("5fa6b05ae2aab76f46b85c6b"), "Nid" : 109, "Name" : "Mohan", "Rank" : "C", "city" : "Delhi", "Mid" : 601 }
>
```

Here for Airforce

>db.airforce.insert({Afid:300,Name:”RK”,Rank:”B”,city:”Lahore”,Mid:601})



```
> db.airforce.find()
{ "_id" : ObjectId("5fa6781f65f2c047b6454ad8"), "Afid" : 300, "Name" : "RK", "Rank" : "A", "city" : "Lahore", "Mid" : 601 }
{ "_id" : ObjectId("5fa6796229c663896c7baa32"), "Afid" : 301, "Name" : "David", "Rank" : "B", "city" : "HYD", "Mid" : 602 }
{ "_id" : ObjectId("5fa6797729c663896c7baa33"), "Afid" : 302, "Name" : "Mohan", "Rank" : "C", "city" : "Chennai", "Mid" : 603 }
{ "_id" : ObjectId("5fa6ad7ee2aab76f46b85c51"), "Afid" : 303, "Name" : "Kiran", "Rank" : "B", "city" : "Delhi", "Mid" : 602 }
{ "_id" : ObjectId("5fa6ada8e2aab76f46b85c52"), "Afid" : 304, "Name" : "Ramu", "Rank" : "A", "city" : "Lahore", "Mid" : 603 }
{ "_id" : ObjectId("5fa6adc5e2aab76f46b85c53"), "Afid" : 305, "Name" : "Pramod", "Rank" : "C", "city" : "Mumbai", "Mid" : 601 }
{ "_id" : ObjectId("5fa6addce2aab76f46b85c54"), "Afid" : 306, "Name" : "Amar", "Rank" : "C", "city" : "Delhi", "Mid" : 603 }
{ "_id" : ObjectId("5fa6adede2aab76f46b85c55"), "Afid" : 307, "Name" : "Verma", "Rank" : "C", "city" : "Delhi", "Mid" : 604 }
{ "_id" : ObjectId("5fa6ae04e2aab76f46b85c56"), "Afid" : 308, "Name" : "Naveen", "Rank" : "B", "city" : "Hyderaba", "Mid" : 602 }
{ "_id" : ObjectId("5fa6ae25e2aab76f46b85c57"), "Afid" : 309, "Name" : "Charan", "Rank" : "A", "city" : "Hyderaba", "Mid" : 601 }
>
```

POSTLAB

1. Most NoSQL databases support automatic _____ meaning that you get high availability and disaster recovery.

Ans) **replication**

2. Which format is supported by MongoDB?

Ans) In MongoDB, data is stored as documents. These documents are stored in MongoDB in JSON (JavaScript Object Notation) format. JSON documents support embedded fields, so related data and lists of data can be stored with the document instead of an external table

3. Dynamic schema in MongoDB makes______________________easier for applications.

Ans) **polymorphism**

4. Which is the interactive shell for MongoDB called?

Ans) mongo is an interactive JavaScript shell interface to MongoDB, which provides a powerful interface for system administrators as well as a way for developers to test queries and operations directly with the database. mongo also provides a fully functional JavaScript environment for use with a MongoDB.

5. MongoDB stores all documents in ______________________

Ans) **Collections**

6. Which command is used to add a new document to the users collection?

Ans) MongoDB provides the insert () command to insert documents into a collection.

Basic syntax is:

```
>db.COLLECTION_NAME.insert(document)
```

Within the "insert" command, add the required Field Name and Field Value for the document which needs to be created.