

**EXPERIMENT-12****PRELAB****1. What is CRUD?**

CRUD - Create, Read, Update, Delete

When we build an api's we want our models to provide four basic types of functionalities. Computer scientists often refer them as CRUD.

**2. What is a Namespace in MongoDB?**

MongoDB stores BSON objects in the collection. The concatenation of the collection name and database name is called a namespace.

**3. What are the components of the Sharded cluster?**

A sharded cluster consists of shards , mongos, routers and config servers. Shards (upper left ) store the application data only the mongo routers or system administrators should be connecting directly to shards .

**4. What is the syntax of the limit() method and sort() method?****Syntax of limit Method**

```
> db.collection_name.find().limit(number)
```

**Syntax of sort Method**

```
> db.collection_name.find().sort({KEY:1})
```

**5. What is the difference between MongoDB and MySQL?**

MySQL is a relational database management system (RDBMS) from the oracle corporation. Like other relational systems , MySQL stores data in tables and uses structured query language (SQL) for database access. MongoDB is a NoSQL database that stores data as JSON documents.

## INLAB

## Construct Queries using MongoDB on Case Study 7

1. Write a query to create a collection called Property Owner with an attributes defined in skilling session 5.

**Basic Syntax:**

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

Here Collection Name is **Property Owner** So,

>db.createCollection("PropertyOwner",{capped:true, size:5000, max: 5 });

Command Prompt - mongo

```
> db.createCollection("PropertyOwner",{capped:true,size:5000,max:5});  
{ "ok" : 1 }
```

2. Write a query to construct a collection called Business Owner with an attributes defined in skilling session 5

**Basic Syntax:**

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

Here Collection Name is **Business Owner** So,

>db.createCollection("BusinessOwner",{capped:true, size:6000, max: 7 });

Command Prompt - mongo

```
> db.createCollection("BusinessOwner",{capped:true,size:6000,max:7});  
{ "ok" : 1 }
```

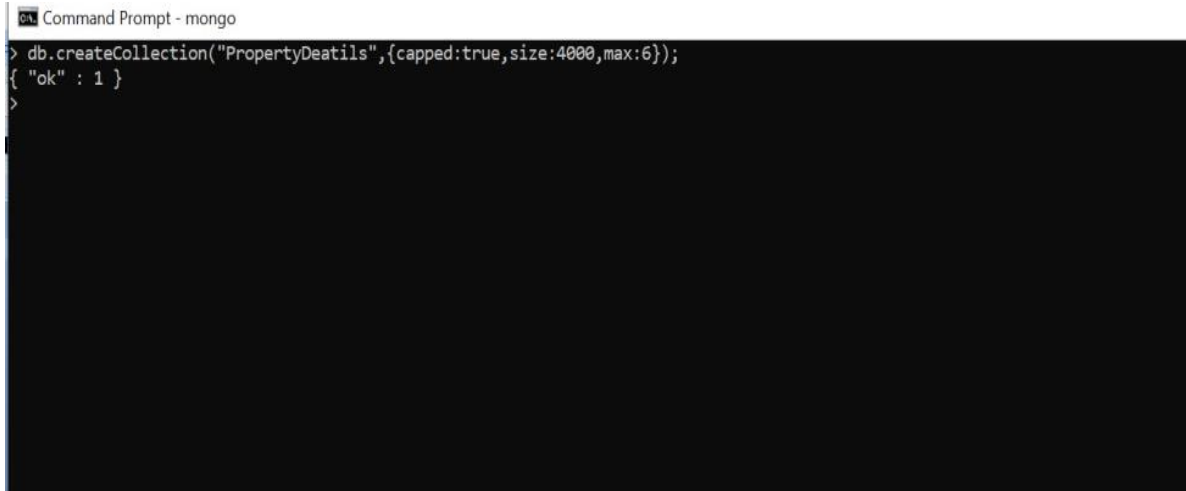
### 3. Write a query to create a collection with the property details with an attributes defined in lab skilling session 5

#### Basic Syntax:

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

Here Collection Name is **Property Details** So,

>db.createCollection("PropertyDetails",{capped:true, size:4000, max: 6 });



```
Command Prompt - mongo
> db.createCollection("PropertyDeatils",{capped:true,size:4000,max:6});
{ "ok" : 1 }
>
```

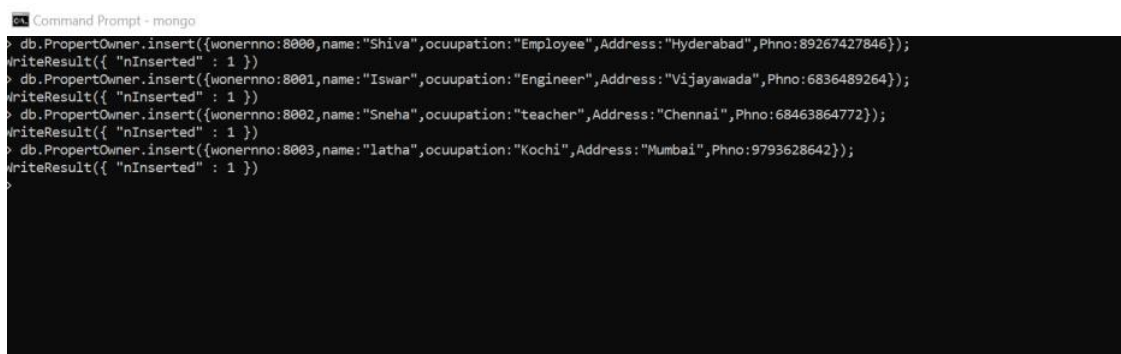
### 4. Construct a query to insert data into above created collections.

#### Basic Syntax:

>db.COLLECTION\_NAME.insert(document)

Here for PropertyOwner

>db.PropertyOwner.insert({wonerno:8000,name:"Shiva",occupation:"Employee",Address:"Hyderabad",Phno:89267427846})



```
Command Prompt - mongo
> db.PropertyOwner.insert({wonerno:8000,name:"Shiva",ocuupation:"Employee",Address:"Hyderabad",Phno:89267427846});
WriteResult({ "nInserted" : 1 })
> db.PropertyOwner.insert({wonerno:8001,name:"Iswar",ocuupation:"Engineer",Address:"Vijayawada",Phno:6836489264});
WriteResult({ "nInserted" : 1 })
> db.PropertyOwner.insert({wonerno:8002,name:"Sneha",ocuupation:"teacher",Address:"Chennai",Phno:68463864772});
WriteResult({ "nInserted" : 1 })
> db.PropertyOwner.insert({wonerno:8003,name:"latha",ocuupation:"Kochi",Address:"Mumbai",Phno:9793628642});
WriteResult({ "nInserted" : 1 })
>
```

**Basic Syntax:**

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

Here for BusinessOwner

```
>db.BusinessOwner.insert({bno:9801,name:"raja",occupation:"clerk",Address:"Mumabi",Phno:8869264862})
```

```
Command Prompt - mongo
> db.BusinessOwner.insert({bno:9801,name:"raja",occupation:"clerk",Address:"Mumbai",Phno:8869264862});
WriteResult({ "nInserted" : 1 })
> db.BusinessOwner.insert({bno:9802,name:"Sundhar",occupation:"Teacher",Address:"Chennai",Phno:7629462884});
WriteResult({ "nInserted" : 1 })
> db.BusinessOwner.insert({bno:9803,name:"raghu",occupation:"Journalist",Address:"Vijayawada",Phno:975575377});
WriteResult({ "nInserted" : 1 })
>
```

**Basic Syntax:**

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

Here for PropertyDetails

```
>db.PropertyDetails.insert({pno:63589,Address:"hyderabad",Owned_By:8892,Ownes_seen_by:50012})
```

```
Command Prompt - mongo
> db.PropertyDetails.insert({pno:63589,Address:"hyderabad",Owned_By:8892,Ownes_seen_by:50012});
WriteResult({ "nInserted" : 1 })
> db.PropertyDetails.insert({pno:63590,Address:"Bangalore",Owned_By:8893,Ownes_seen_by:50013});
WriteResult({ "nInserted" : 1 })
> db.PropertyDetails.insert({pno:102590,Address:"Vijayawada",Owned_By:8894,Ownes_seen_by:50014});
WriteResult({ "nInserted" : 1 })
> db.PropertyDetails.insert({pno:189590,Address:"Chennai",Owned_By:8895,Ownes_seen_by:50015});
WriteResult({ "nInserted" : 1 })
>
```

## 5. Construct a query which retrieves details of Property that have hired or being hired by each renter.

```
Command Prompt - mongo
> db.PropertyOwner.find().pretty();
{
  "_id" : ObjectId("5fa61ff10163c2c39d139968"),
  "woneranno" : 8000,
  "name" : "Shiva",
  "occupation" : "Employee",
  "Address" : "Hyderabad",
  "Phno" : 89267427846
}
{
  "_id" : ObjectId("5fa620110163c2c39d139969"),
  "woneranno" : 8001,
  "name" : "Iswar",
  "occupation" : "Engineer",
  "Address" : "Vijayawada",
  "Phno" : 6836489264
}
{
  "_id" : ObjectId("5fa620330163c2c39d13996a"),
  "woneranno" : 8002,
  "name" : "Sneha",
  "occupation" : "teacher",
  "Address" : "Chennai",
  "Phno" : 68463864772
}
{
  "_id" : ObjectId("5fa620560163c2c39d13996b"),
  "woneranno" : 8003,
  "name" : "latha",
  "occupation" : "Kochi",
  "Address" : "Mumbai",
  "Phno" : 9793628642
}
>
```

6. Construct a query which retrieves details of property submitted for hire by each property owner.

Command Prompt - mongo

```
> db.BusinessOwner.find().pretty();
{
  "_id" : ObjectId("5fa621070163c2c39d13996c"),
  "bno" : 9801,
  "name" : "raja",
  "occupation" : "clerk",
  "Address" : "Mumbai",
  "Phno" : 8869264862
}
{
  "_id" : ObjectId("5fa6211230163c2c39d13996d"),
  "bno" : 9802,
  "name" : "Sundhar",
  "occupation" : "Teacher",
  "Address" : "Chennai",
  "Phno" : 7629462884
}
{
  "_id" : ObjectId("5fa621600163c2c39d13996e"),
  "bno" : 9803,
  "name" : "raghu",
  "occupation" : "Journalist",
  "Address" : "Vijayawada",
  "Phno" : 975575377
}
```

**POSTLAB**

1) MongoDB process collection of documents using \_\_\_\_\_ operations.

Ans) Map-reduce

2) Which pipeline is used for aggregation in MongoDB?

Ans) MongoDB provides the db.collection.aggregate() method in the mongo shell and the aggregate ccommand to run the aggregation pipeline.

3) In aggregation pipeline, the \_\_\_\_\_ pipeline stage provides access to MongoDB queries.

Ans) \$match

4) To suppress the \_id field from the result set, specify \_\_\_\_\_ in the projection document.

Ans) \_id:0

5) \_\_\_\_\_ command display the list of databases.

Ans) show dbs;

6) Command to check existence of collection is \_\_\_\_

Ans) show collections