

PRACTICAL-2

Aim : Commands

1. Create User

Ans :-

```
db.createUser({ user: "guru", pwd: "guru007", roles:[{role: "userAdminAnyDatabase", db:"admin"}] })
```

Output:

```
guru> db.createUser({user:"guru",pwd:"guru007",roles:[{role:"userAdminAnyDatabase",db:"admin"}]});  
{ ok: 1 }
```

2. Create Database

Ans :-

```
use emp
```

Output :-

```
test> use emp  
switched to db emp  
emp>
```

3. Create Collection

Ans :-

```
db.createCollection("emp")
```

Output :-

```
emp> db.createCollection("emp")  
{ ok: 1 }  
emp>
```

4. Insert Data into Database

Ans :-

```
db.emp.insert({"empid":1,"empname":"martin"})
```

Output :-

```
emp> db.emp.insert({"empid":1,"empname":"martin"})
DeprecationWarning: Collection.insert() is deprecated. Use insertOne, insertMany, or bulkWrite.
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63ec658e989c1d7779dae5d8") }
}
emp>
```

```
db.emp.insert({"empid":2,"empname":"neha","age":23})
```

```
emp> db.emp.insert({"empid":2,"empname":"neha","age":23})
{
  acknowledged: true,
  insertedIds: { '0': ObjectId("63ec65d3989c1d7779dae5d9") }
}
emp> |
```

```
var myemp=[{"empid":3,"empname":"smith"}, {"empid":4,"empname":"John"}]
db.emp.insert(myemp)
```

```
emp> var myemp=[{"empid":3,"empname":"smith"}, {"empid":4,"empname":"John"}]
emp> db.emp.insert(myemp)
{
  acknowledged: true,
  insertedIds: {
    '0': ObjectId("63ec6690989c1d7779dae5da"),
    '1': ObjectId("63ec6690989c1d7779dae5db")
  }
}
emp>
```

```
db.emp.find().pretty()
```

```
emp> db.emp.find().pretty()
[
  {
    _id: ObjectId("63ec658e989c1d7779dae5d8"),
    empid: 1,
    empname: 'martin'
  },
  {
    _id: ObjectId("63ec65d3989c1d7779dae5d9"),
    empid: 2,
    empname: 'neha',
    age: 23
  },
  {
    _id: ObjectId("63ec6690989c1d7779dae5da"),
    empid: 3,
    empname: 'smith'
  },
  {
    _id: ObjectId("63ec6690989c1d7779dae5db"),
    empid: 4,
    empname: 'John'
  }
]
emp>
```

5. Update and Delete Data

Ans :-

```
db.emp.update({"empid":1},{ $set: {"empname": "xyz" }})
```

Output :-

```
emp> db.emp.update({"empid":1},{ $set: {"empname": "xyz" }})
DeprecationWarning: Collection.update() is deprecated. Use updateOne, updateMany, or bulkWrite.
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
db.emp.update({"empid":1},{ $set: {age:40} })
```

```
emp> db.emp.update({"empid":1},{ $set: {age:40} })
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

```
db.emp.find({"empid":1}).pretty()
```

```
emp> db.emp.find({"empid":1}).pretty()
[
  {
    _id: ObjectId("63ec658e989c1d7779dae5d8"),
    empid: 1,
    empname: 'xyz',
    age: 40
  }
]
```

➤ **Remove Particular Field :-**

Ans :-

```
db.emp.update({"empid":1},{ $unset:{age:40}})
```

Output :-

```
emp> db.emp.update({"empid":1},{ $unset:{age:40}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

➤ **Update value of age by 5 :**

Ans :-

```
db.emp.update({"empid":1},{ $inc:{age:5}})
```

Output :-

```
emp> db.emp.update({"empid":1},{ $inc:{age:5}})
{
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

➤ **Removing whole Document :-**

Ans :-

```
db.emp.remove({empid:2})
```

Output :-

```
emp> db.emp.remove({empid:2})
DeprecationWarning: Collection.remove() is deprecated. Use deleteOne, deleteMany, findOneAndDelete, or bulkWrite.
{ acknowledged: true, deletedCount: 1 }
emp>
```

- If no document matching with particular condition exists, then if we set upsert "true", then initially it will also create it :-

Ans :-

```
db.emp.update({empid:3},{empid:5,empname:"heena",{upset:true}});
```

Output :-

```
> db.emp.update({empid:3},{empid:5,empname:"heena",{upset:true}});  
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
```

- Renaming any particular field :-

Ans :-

```
db.emp.update({empid:4},{ $rename: {"empname": "nameemp" } });
```

Output :-

```
emp> db.emp.update({empid:4},{ $rename: {"empname": "nameemp" } });  
{  
  acknowledged: true,  
  insertedId: null,  
  matchedCount: 1,  
  modifiedCount: 1,  
  upsertedCount: 0  
}
```

6. Retrival of Data

```
db.emp.find().forEach(printjson)
```

Output :-

```
emp> db.emp.find().forEach(printjson)
{
  _id: ObjectId("63ec658e989c1d7779dae5d8"),
  empid: 1,
  empname: 'xyz',
  age: 5
}
{
  _id: ObjectId("63ec6690989c1d7779dae5da"),
  empid: 5,
  empname: 'heena'
}
{
  _id: ObjectId("63ec6690989c1d7779dae5db"),
  empid: 4,
  nameemp: 'John'
}
```

```
db.emp.find().pretty()
```

Output :-

```
emp> db.emp.find().pretty()
[
  {
    _id: ObjectId("63ec658e989c1d7779dae5d8"),
    empid: 1,
    empname: 'xyz',
    age: 5
  },
  {
    _id: ObjectId("63ec6690989c1d7779dae5da"),
    empid: 5,
    empname: 'heena'
  },
  {
    _id: ObjectId("63ec6690989c1d7779dae5db"),
    empid: 4,
    nameemp: 'John'
  }
]
```

```
db.emp.find({})
```

```
emp> db.emp.find({})
[
  {
    _id: ObjectId("63ec658e989c1d7779dae5d8"),
    empid: 1,
    empname: 'xyz',
    age: 5
  },
  {
    _id: ObjectId("63ec6690989c1d7779dae5da"),
    empid: 5,
    empname: 'heena'
  },
  {
    _id: ObjectId("63ec6690989c1d7779dae5db"),
    empid: 4,
    nameemp: 'John'
  }
]
```

```
db.emp.find({empname:"xyz"}).forEach(printjson)
```

```
emp> db.emp.find({empname:"xyz"}).forEach(printjson)
{
  _id: ObjectId("63ec658e989c1d7779dae5d8"),
  empid: 1,
  empname: 'xyz',
  age: 5
}
```

```
db.emp.find({empid:{$gt:2}})
```

```
emp> db.emp.find({empid:{$gt:2}})
[
  {
    _id: ObjectId("63ec6690989c1d7779dae5da"),
    empid: 5,
    empname: 'heena'
  },
  {
    _id: ObjectId("63ec6690989c1d7779dae5db"),
    empid: 4,
    nameemp: 'John'
  }
]
```

```
var myemp=db.emp.find({empid:{$gt:2}})
while(myemp.hasNext()){
... print(tojson(myemp.next()));
... }
```

```
> var myemp=db.emp.find( { empid : {$gt:2}})
> while(myemp.hasNext()){ print(tojson(myemp.next())); }
{
  "_id" : ObjectId("63ec6690989c1d7779dae5da"),
  "empid" : 5,
  "empname" : "heena"
}
{
  "_id" : ObjectId("63ec6690989c1d7779dae5db"),
  "empid" : 4,
  "nameemp" : "John"
}
```

7. Display Database

Ans :-

```
show dbs
```

Output :-

```
emp> show dbs
admin      180.00 KiB
config     108.00 KiB
emp        72.00 KiB
guru       72.00 KiB
local      72.00 KiB
res_guru   640.00 KiB
emp>
```