UI - Assignment (Surendiran S || IT-B || Roll.no: 98)

Sunday, January 19, 2020

1. Show Databases

```
> show dbs
admin       0.000GB
config      0.000GB
local       0.000GB
suren       0.000GB
surendiran  0.000GB
```

2. To create Database

```
> use suren
switched to db suren
```

3. To Select the currently selected Database

```
> db
suren
```

4. Drop a Database

```
> db.dropDatabase()
{ "dropped" : "suren", "ok" : 1 }
```

5. To Create a collection

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

6. To check the created collection

```
> show collections
UI
```

7. Create Collections automatically when you insert some documents

```
> db.UI2.insert({"name" : "suren"})
WriteResult({ "nInserted" : 1 })
> show collections
UI
UI2
```

8. Drop a collection

```
> db.UI.drop()
true
```

9. Insert a Single Document

```
> db.inventory.insertOne(
...    { item: "canvas", qty: 100, tags: ["cotton"], size: { h: 28, w: 35.5, uom: "cm" } }
... )
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5e24409162ce7bbc96e0a0fc")
}
>
```

10. Insert Multiple Documents

```
> db.inventory.insertMany([
...    { item: "journal", qty: 25, size: { h: 14, w: 21, uom: "cm" }, status: "A" },
...    { item: "notebook", qty: 50, size: { h: 8.5, w: 11, uom: "in" }, status: "A" },
...    { item: "paper", qty: 100, size: { h: 8.5, w: 11, uom: "in" }, status: "D" },
...    { item: "planner", qty: 75, size: { h: 22.85, w: 30, uom: "cm" }, status: "D" },
...    { item: "postcard", qty: 45, size: { h: 10, w: 15.25, uom: "cm" }, status: "A" }
... ]);
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("5e243cf462ce7bbc96e0a0f6"),
                ObjectId("5e243cf462ce7bbc96e0a0f7"),
                ObjectId("5e243cf462ce7bbc96e0a0f8"),
                ObjectId("5e243cf462ce7bbc96e0a0f9"),
                ObjectId("5e243cf462ce7bbc96e0a0fa")
        ]
}
```

11. Select All Documents in a Collection

```
> db.inventory.find( {} )
{ "_id" : ObjectId("5e243bfa62ce7bbc96e0a0ef"), "item" : "journal", "qty" : 25, "tags" : [ "blank", "red" ], "size" : { "h" : 14, "w" : 21, "uom" : "cm" } }
{ "_id" : ObjectId("5e243bfa62ce7bbc96e0a0f0"), "item" : "mat", "qty" : 85, "tags" : [ "gray" ], "size" : { "h" : 27.9, "w" : 35.5, "uom" : "cm" } }
{ "_id" : ObjectId("5e243bfa62ce7bbc96e0a0f1"), "item" : "mousepad", "qty" : 25, "tags" : [ "gel", "blue" ], "size" : { "h" : 19, "w" : 22.85, "uom" : "cm" } }
{ "_id" : ObjectId("5e243c0b62ce7bbc96e0a0f2"), "item" : "canvas", "qty" : 100, "tags" : [ "cotton" ], "size" : { "h" : 28, "w" : 35.5, "uom" : "cm" } }
{ "_id" : ObjectId("5e243c2162ce7bbc96e0a0f3"), "item" : "journal", "qty" : 25, "tags" : [ "blank", "red" ], "size" : { "h" : 14, "w" : 21, "uom" : "cm" } }
{ "_id" : ObjectId("5e243c2162ce7bbc96e0a0f4"), "item" : "mat", "qty" : 85, "tags" : [ "gray" ], "size" : { "h" : 27.9, "w" : 35.5, "uom" : "cm" } }
{ "_id" : ObjectId("5e243c2162ce7bbc96e0a0f5"), "item" : "mousepad", "qty" : 25, "tags" : [ "gel", "blue" ], "size" : { "h" : 19, "w" : 22.85, "uom" : "cm" } }
{ "_id" : ObjectId("5e243cf462ce7bbc96e0a0f6"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5e243cf462ce7bbc96e0a0f7"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5e243cf462ce7bbc96e0a0f8"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
{ "_id" : ObjectId("5e243cf462ce7bbc96e0a0f9"), "item" : "planner", "qty" : 75, "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" }, "status" : "D" }
{ "_id" : ObjectId("5e243cf462ce7bbc96e0a0fa"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "A" }
```

12. Specify Equality Condition

```
> db.inventory.find( { status: "D" } )
{ "_id" : ObjectId("5e24416462ce7bbc96e0a104"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
{ "_id" : ObjectId("5e24416462ce7bbc96e0a105"), "item" : "planner", "qty" : 75, "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" }, "status" : "D" }
>
```

13. Specify Conditions Using Query Operators

```
> db.inventory.find( { status: { $in: [ "A", "D" ] } } )
{ "_id" : ObjectId("5e243cf462ce7bbc96e0a0f6"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5e243cf462ce7bbc96e0a0f7"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5e243cf462ce7bbc96e0a0f8"), "item" : "paper", "qty" : 100, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "D" }
{ "_id" : ObjectId("5e243cf462ce7bbc96e0a0f9"), "item" : "planner", "qty" : 75, "size" : { "h" : 22.85, "w" : 30, "uom" : "cm" }, "status" : "D" }
{ "_id" : ObjectId("5e243cf462ce7bbc96e0a0fa"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "A" }
```

14. Specify AND Conditions

```
> db.inventory.find( { status: "A", qty: { $lt: 30 } } )
{ "_id" : ObjectId("5e24416462ce7bbc96e0a102"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
>
```

15. Specify OR Conditions

```
> db.inventory.find( { $or: [ { status: "A" }, { qty: { $lt: 30 } } ] } )
{ "_id" : ObjectId("5e24416462ce7bbc96e0a102"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5e24416462ce7bbc96e0a103"), "item" : "notebook", "qty" : 50, "size" : { "h" : 8.5, "w" : 11, "uom" : "in" }, "status" : "A" }
{ "_id" : ObjectId("5e24416462ce7bbc96e0a106"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "A" }
>
```

16. Specify AND as well as OR Conditions

```
> db.inventory.find( {
...       status: "A",
...       $or: [ { qty: { $lt: 30 } }, { item: /^p/ } ]
... } )
{ "_id" : ObjectId("5e24416462ce7bbc96e0a102"), "item" : "journal", "qty" : 25, "size" : { "h" : 14, "w" : 21, "uom" : "cm" }, "status" : "A" }
{ "_id" : ObjectId("5e24416462ce7bbc96e0a106"), "item" : "postcard", "qty" : 45, "size" : { "h" : 10, "w" : 15.25, "uom" : "cm" }, "status" : "A" }
>
```

17. Update a single Document

```
> db.inventory.updateOne(    { item: "paper" },    {      $set: { "size.uom": "cm", status: "P" },      $currentDate: { lastModified: true }    } )
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 1 }
```

18. Update Multiple Documents

```
> db.javatpoint.insert(
...    {
...      course: "java",
...      details: {
...         duration: "6 months",
...         Trainer: "Sonoo jaiswal"
...      },
...      Batch: [ { size: "Small", qty: 15 }, { size: "Medium", qty: 25 } ],
...      category: "Programming language"
...    }
... )
WriteResult({ "nInserted" : 1 })
> db.javatpoint.find()
{ "_id" : ObjectId("5e24484262ce7bbc96e0a10a"), "course" : "java", "details" : { "duration" : "6 months", "Trainer" : "Sonoo jaiswal" }, "Batch" : [ { "size" : "Small", "qty" : 15 }, { "size" : "Medium", "qty" :
 25 } ], "category" : "Programming language" }
> db.javatpoint.update({'course':'java'},{$set:{'course':'android'}})
WriteResult({ "nMatched" : 1, "nUpserted" : 0, "nModified" : 1 })
> db.javatpoint.find()
{ "_id" : ObjectId("5e24484262ce7bbc96e0a10a"), "course" : "android", "details" : { "duration" : "6 months", "Trainer" : "Sonoo jaiswal" }, "Batch" : [ { "size" : "Small", "qty" : 15 }, { "size" : "Medium", "qty
" : 25 } ], "category" : "Programming language" }
```

19. Remove all Documents

```
> db.javatpoint.remove({})
WriteResult({ "nRemoved" : 1 })
```

20. limit() method

```
> db.javatpoint.find().limit(1)
{ "_id" : ObjectId("5e24496462ce7bbc96e0a10b"), "course" : "java", "details" : { "duration" : "6 months", "Trainer" : "Sonoo jaiswal" }, "Batch" : [ { "size" : "Small", "qty" : 15 }, { "size" : "Medium", "qty" :
 25 } ], "category" : "Programming language" }
```

21. Skip() method

```
> db.javatpoint.find().limit(1).skip(2)
{ "_id" : ObjectId("5e244a8462ce7bbc96e0a10d"), "course" : "java", "details" : { "duration" : "6 months", "Trainer" : "Sonoo jaiswal" }, "Batch" : [ { "size" : "Small", "qty" : 15 }, { "size" : "Medium", "qty" :
 25 } ], "category" : "Programming language" }
>
```

22. sort() method

```
> db.javatpoint.find().sort({"Course":-1})
{ "_id" : ObjectId("5e24496462ce7bbc96e0a10b"), "course" : "java", "details" : { "duration" : "6 months", "Trainer" : "Sonoo jaiswal" }, "Batch" : [ { "size" : "Small", "qty" : 15 }, { "size" : "Medium", "qty" :
 25 } ], "category" : "Programming language" }
{ "_id" : ObjectId("5e24497462ce7bbc96e0a10c"), "course" : "java", "details" : { "duration" : "6 months", "Trainer" : "Sonoo jaiswal" }, "Batch" : [ { "size" : "Small", "qty" : 15 }, { "size" : "Medium", "qty" :
 25 } ], "category" : "Programming language" }
{ "_id" : ObjectId("5e244a8462ce7bbc96e0a10d"), "course" : "java", "details" : { "duration" : "6 months", "Trainer" : "Sonoo jaiswal" }, "Batch" : [ { "size" : "Small", "qty" : 15 }, { "size" : "Medium", "qty" :
 25 } ], "category" : "Programming language" }
```

23. MongoDB statistics

```
> db.stats()
{
        "db" : "suren",
        "collections" : 2,
        "views" : 0,
        "objects" : 4,
        "avgObjSize" : 178.25,
        "dataSize" : 713,
        "storageSize" : 57344,
        "numExtents" : 0,
        "indexes" : 2,
        "indexSize" : 57344,
        "scaleFactor" : 1,
        "fsUsedSize" : 135000981504,
        "fsTotalSize" : 214748360704,
        "ok" : 1
}
```

24. MongoDB help

```
> db.help()
DB methods:
        db.adminCommand(nameOrDocument) - switches to 'admin' db, and runs command [just calls db.runCommand(...)]
        db.aggregate([pipeline], {options}) - performs a collectionless aggregation on this database; returns a cursor
        db.auth(username, password)
        db.cloneDatabase(fromhost) - will only function with MongoDB 4.0 and below
        db.commandHelp(name) returns the help for the command
        db.copyDatabase(fromdb, todb, fromhost) - will only function with MongoDB 4.0 and below
        db.createCollection(name, {size: ..., capped: ..., max: ...})
        db.createUser(userDocument)
        db.createView(name, viewOn, [{$operator: {...}}, ...], {viewOptions})
        db.currentOp() displays currently executing operations in the db
        db.dropDatabase(writeConcern)
        db.dropUser(username)
        db.eval() - deprecated
        db.fsyncLock() flush data to disk and lock server for backups
        db.fsyncUnlock() unlocks server following a db.fsyncLock()
        db.getCollection(cname) same as db['cname'] or db.cname
        db.getCollectionInfos([filter]) - returns a list that contains the names and options of the db's collections
```

25. Clear the Screen

    cls