Perform following queries using use, drop, createcollection, dropcollection, insertOne and insertMany method. (Viraj Odedra)

***LAB 8 and 9 :***

Part - A 1. Create a new database named “Darshan”.

use Darshan

2. Create another new database named “DIET”.

use DIET

3. List all databases.

show databases

4. Check the current database.

db

5. Drop “DIET” database.

db.dropDatabase()

6. Create a collection named “Student” in the “Darshan” database.

use Darshan

db.createCollection("Student")

7. Create a collection named “Department” in the “Darshan” database.

db.createCollection("Department")db.createCollection("Department")

8. List all collections in the “Darshan” database.

db.createCollection("Department")

9. Insert a single document using insertOne into “Department” collection. (Dname:’CE’, HOD:’Patel’)

db.Department.insertOne({

Dname : 'IT',

Dname : 'ICT'

})

10. Insert two document using insertMany into “Department” collection. (Dname:’IT’ and Dname:’ICT’)

db.Department.insertMany([

{Dname:'IT'},

{Dname:'ICT'}

])

11. Drop a collection named “Department” from the “Darshan” database.

db.Department.drop()

12. Insert a single document using insertOne into “Student” collection. (Fields are Name, City, Branch, Semester, Age) Insert your own data.

db.Student.insertOne({

Name : 'Pruthviraj',

City : 'Vakaner',

Branch : 'CSE',

Semester : 04,

Age : 05

})

13. Insert three documents using insertMany into “Student” collection. (Fields are Name, City, Branch, Semester, Age) Insert your three friend’s data.

db.Student.insertMany([

{

Name : 'Dhairya',

City : 'Junagadh',

Branch : 'CSE',

Semester : 04,

Age : 19

},

{

Name : 'Viraj',

City : 'Kutiyana',

Branch : 'CSE',

Semester : 04,

Age : 18

}

])

14. Check whether “Student” collection exists or not.

db.getCollectionNames().includes("Student")

15. Check the stats of “Student” collection.

db.Student.stats()

16. Drop the “Student” collection.

db.Student.drop()

17. Create a collection named “Deposit”.

db.createCollection("Deposit")

18. Insert following data in to “Deposit” collection.

db.Deposit.insertMany(

[

{ ACTNO: 101, CNAME: 'ANIL', BNAME: 'VRCE', AMOUNT: 1000.00, CITY: 'RAJKOT' },

{ ACTNO: 102, CNAME: 'SUNIL', BNAME: 'AJNI', AMOUNT: 5000.00, CITY: 'SURAT' },

{ ACTNO: 103, CNAME: 'MEHUL', BNAME: 'KAROLBAGH', AMOUNT: 3500.00, CITY: 'BARODA' },

{ ACTNO: 104, CNAME: 'MADHURI', BNAME: 'CHANDI', AMOUNT: 1200.00, CITY: 'AHMEDABAD' },

{ ACTNO: 105, CNAME: 'PRMOD', BNAME: 'M.G. ROAD', AMOUNT: 3000.00, CITY: 'SURAT' },

{ ACTNO: 106, CNAME: 'SANDIP', BNAME: 'ANDHERI', AMOUNT: 2000.00, CITY: 'RAJKOT' },

{ ACTNO: 107, CNAME: 'SHIVANI', BNAME: 'VIRAR', AMOUNT: 1000.00, CITY: 'SURAT' },

{ ACTNO: 108, CNAME: 'KRANTI', BNAME: 'NEHRU PLACE', AMOUNT: 5000.00, CITY: 'RAJKOT' }

])

19. Display all the documents of “Deposit” collection.

db.Deposit.find()

20. Drop the “Deposit” collection.

db.Deposit.drop()

Part – B

1. Create a new database named “Computer”.

2. Create a collection named “Faculty” in the “Computer” database.

3. Insert a below document using insertOne into “Faculty” collection.

4. Insert below documents using insertMany into “Faculty” collection.

5. Display all the documents of “Faculty” collection.

6. Drop the “Faculty” collection.

7. Drop the “Computer” database

Part – C (Perform following operation using UI)

1. Create a new database named “Computer”.

2. Create a collection named “Faculty” in the “Computer” database.

3. Insert a below documents into “Faculty” collection.

4. Display all the documents of “Faculty” collection.

5. Drop the “Faculty” collection.

6. Drop the “Computer” database

***Lab-9 (Part - A)***

1. Retrieve/Display every document of Deposit collection.

use BANK\_INFO

db.createCollection('Diposit')

db.Deposit.insertMany(

[

{

"ACTNO": 101,

"CNAME": "ANIL",

"BNAME": "VRCE",

"AMOUNT": 1000,

"ADATE": "1995-03-01"

},

{

"ACTNO": 102,

"CNAME": "SUNIL",

"BNAME": "AJNI",

"AMOUNT": 5000,

"ADATE": "1996-01-04"

},

{

"ACTNO": 103,

"CNAME": "MEHUL",

"BNAME": "KAROLBAGH",

"AMOUNT": 3500,

"ADATE": "1995-11-17"

},

{

"ACTNO": 104,

"CNAME": "MADHURI",

"BNAME": "CHANDI",

"AMOUNT": 1200,

"ADATE": "1995-12-17"

},

{

"ACTNO": 105,

"CNAME": "PRMOD",

"BNAME": "M.G. ROAD",

"AMOUNT": 3000,

"ADATE": "1996-03-27"

},

{

"ACTNO": 106,

"CNAME": "SANDIP",

"BNAME": "ANDHERI",

"AMOUNT": 2000,

"ADATE": "1996-03-31"

},

{

"ACTNO": 107,

"CNAME": "SHIVANI",

"BNAME": "VIRAR",

"AMOUNT": 1000,

"ADATE": "1995-09-05"

},

{

"ACTNO": 108,

"CNAME": "KRANTI",

"BNAME": "NEHRU PLACE",

"AMOUNT": 5000,

"ADATE": "1995-07-02"

}

]

)

db.Deposit.find()

2. Display only one document of Deposit collection. (Use: findOne())

db.Deposit.findOne()

3. Insert following document into Deposit collection. (Use: insertOne()) 109 KIRTI VIRAR 3000 3-5-97

db.Deposit.insertOne({

ACTNO: 109,

CNAME: 'KIRTI',

BNAME: 'VIRAR',

AMOUNT: 3000,

ADATE: '1997-5-3'

})

4. Insert following documents into Deposit collection. (Use: insertMany()) 110 MITALI ANDHERI 4500 4-9-95 111 RAJIV NEHRU PLACE 7000 2-10-98

db.Deposit.insertMany([

{

ACTNO: 110,

CNAME: 'MITALI',

BNAME: 'ANDHERI',

AMOUNT: 4500,

ADATE: '1995-9-4'

},

{

ACTNO: 111,

CNAME: 'RAJIV',

BNAME: 'NEHRU PLACE',

AMOUNT: 7000,

ADATE: '1998-10-2'

}

])

5. Display all the documents of ‘VIRAR’ branch from Deposit collection.

db.Deposit.find({BNAME : 'VIRAR'})

6. Display all the documents of Deposit collection whose amount is between 3000 and 5000.

db.Deposit.find({$and : [

{AMOUNT : {$gt : 3000}},

{AMOUNT : {$lt : 5000}}

]})

7. Display all the documents of Deposit collection whose amount is greater than 2000 and branch is VIRAR.

db.Deposit.find({$and : [

{AMOUNT : {$gt : 2000}},

{BNAME : 'VIRAR'}

]})

8. Display all the documents with CNAME, BNAME and AMOUNT fields from Deposit collection.

db.Deposit.find({},{CNAME : 1, BNAME : 1, AMOUNT : 1, \_id : 0 })

9. Display all the documents of Deposit collection on ascending order by CNAME.

db.Deposit.find().sort({CNAME : -1})

10. Display all the documents of Deposit collection on descending order by BNAME.

db.Deposit.find().sort({BNAME : -1})

11. Display all the documents of Deposit collection on ascending order by ACTNO and descending order by AMOUNT.

db.Deposit.find().sort({ACTNO : 1, AMOUNT : -1})

12. Display only two documents of Deposit collection.

db.Deposit.find().limit(2)

13. Display 3rd document of Deposit collection.

db.Deposit.find().skip(2).limit(1)

14. Display 6th and 7th documents of Deposit collection.

db.Deposit.find().skip(5).limit(2)

15. Display the count of documents in Deposit collection.

db.Deposit.countDocuments()