

Mongo-DB Assessment

Name : Chandru V

Reg No : 23BIT0378

1.Insert a single Document into a Collection named student record with fields:name,age and course.

Query :

```
db.StudentInfo.insertOne({name:"chandru",age:18,course:"Maths",mark:70,Roll_No:"23BIT0378"});
```

Output :

```
> db.StudentInfo.insertOne({name:"chandru",age:18,course:"Maths",mark:70,Roll_No:"23BIT0378"})
< {
  acknowledged: true,
  insertedId: ObjectId('685ad904e1d9b7d2d8a80a9d')
}
```

2.Insert multiple documents into employess collection with fields :name,salary,department.

Query:

```
db.EmployeeDetails.insertMany([
  {name:"Chandru",salary:10000,dept:"IT"},
  {name:"Lokesh",salary:20000,dept:"Sales"},
  {name:"Vishwa",salary:30000,dept:"IT"},
  {name:"Ravishanker",salary:40000,dept:"Sales"},
  {name:"Rakesh",salary:50000,dept:"IT"},
  {name:"Akash",salary:60000,dept:"HR"},
  {name:"pharabakaran",salary:70000,dept:"HR"}]);
```

Output :

```

> db.EmployeeDetails.insertMany([
  {name:"Chandru",salary:10000,dept:"IT"},
  {name:"Lokesh",salary:20000,dept:"Sales"},
  {name:"Vishwa",salary:30000,dept:"IT"},
  {name:"Ravishanker",salary:40000,dept:"Sales"},
  {name:"Rakesh",salary:50000,dept:"IT"},
  {name:"Akash",salary:60000,dept:"HR"},
  {name:"pharabakaran",salary:70000,dept:"HR"}
])
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('685af9bfe669c6cacfa34471'),
    '1': ObjectId('685af9bfe669c6cacfa34472'),
    '2': ObjectId('685af9bfe669c6cacfa34473'),
    '3': ObjectId('685af9bfe669c6cacfa34474'),
    '4': ObjectId('685af9bfe669c6cacfa34475'),
    '5': ObjectId('685af9bfe669c6cacfa34476'),
    '6': ObjectId('685af9bfe669c6cacfa34477')
  }
}

```

3.Display the documents who have salary graterthan 30000

Query:

```
db.EmployeeDetails.find({salary:{$gt:30000}},{_id:0});
```

Output :

```
> db.EmployeeDetails.find({salary:{$gt:30000}},{_id:0});  
< {  
  name: 'Ravishanker',  
  salary: 40000,  
  dept: 'Sales'  
}  
{  
  name: 'Rakesh',  
  salary: 50000,  
  dept: 'IT'  
}  
{  
  name: 'Akash',  
  salary: 60000,  
  dept: 'HR'  
}  
{  
  name: 'pharabakaran',  
  salary: 70000,  
  dept: 'HR'  
}
```

4.Find all documents from Product collections.

Query :

```
db.ProductDetails.find();
```

Output:

```
> db.ProductDetails.find();
< {
  _id: ObjectId('685a9b2b119cba058c7a6b8c'),
  product: 'Computer',
  product_ID: '23CID456',
  price: 10000
}
{
  _id: ObjectId('685a9b2b119cba058c7a6b8d'),
  product: 'Fan',
  product_ID: '28FID456',
  price: 20000
}
```

```
{
  _id: ObjectId('685a9b2b119cba058c7a6b8e'),
  product: 'Car',
  product_ID: '21CID456',
  price: 30000
}
{
  _id: ObjectId('685a9b2b119cba058c7a6b8f'),
  product: 'Washing Meashin',
  product_ID: '29WID456',
  price: 40000
}
```

5.Find the all document from the User collection who age grater than 25.

Query:

```
db.UserDetails.find({User_Age:{$gt:25}});
```

Output:

```
> db.userDetails.find({User_Age:{$gt:25}});
< {
  _id: ObjectId('685ae4cfec4a9c5b95d561b6'),
  Username: 'Lokesh',
  User_Age: 28,
  User_ID: '23BME0378',
  Location: 'chennai',
  Email: 'sss.com',
  Create_Date: '7-oct-2006',
  active: 'false'
}
{
  _id: ObjectId('685ae4cfec4a9c5b95d561b7'),
  Username: 'Vishwa',
  User_Age: 30,
  User_ID: '23B03MI78',
  Location: 'Bangalore',
  Email: 'zzz.com',
  Create_Date: '9-nov-2006',
  active: 'true'
}
```

6.Find first five document from the user collection using the limit().

Query:

```
db.userDetails.find().limit(5);
```

Output:

```
> db.userDetails.find().limit(5);
< {
  _id: ObjectId('685ae4cfec4a9c5b95d561b4'),
  Username: 'Chandru',
  User_Age: 20,
  User_ID: '23BIT0378',
  Location: 'Bangalore',
  Email: 'xxx.com',
  Create_Date: '2-aug-2006',
  active: 'false'
}
```

```
{
  _id: ObjectId('685ae4cfec4a9c5b95d561b5'),
  Username: 'Akash',
  User_Age: 19,
  User_ID: '23BEC0378',
  Location: 'Bangalore',
  Email: 'yyy.com',
  Create_Date: '5-sep-2006',
  active: 'true'
}
{
  _id: ObjectId('685ae4cfec4a9c5b95d561b6'),
  Username: 'Lokesh',
  User_Age: 28,
  User_ID: '23BME0378',
  Location: 'chennai',
  Email: 'sss.com',
  Create_Date: '7-oct-2006',
  active: 'false'
}
```

```
{
  _id: ObjectId('685ae4cfec4a9c5b95d561b7'),
  Username: 'Vishwa',
  User_Age: 30,
  User_ID: '23B03MI78',
  Location: 'Bangalore',
  Email: 'zzz.com',
  Create_Date: '9-nov-2006',
  active: 'true'
}
{
  _id: ObjectId('685ae70cec4a9c5b95d561b8'),
  Username: 'naresh',
  User_Age: 30,
  User_ID: '23BMIC78',
  Location: 'karnadaka',
  Email: 'aaa.com',
  Create_Date: '9-nov-2006',
  active: 'true'
}
```

7.sort the user documents assending order based on createdate.

Query:

```
db.userDetails.find({}, {Username:1, Create_Date:1, _id:0}).sort({Create_Date:-1});
```

Output:

```
< {  
  Username: 'Vishwa',  
  Create_Date: '9-nov-2006'  
}  
{  
  Username: 'naresh',  
  Create_Date: '9-nov-2006'  
}  
{  
  Username: 'Lokesh',  
  Create_Date: '7-oct-2006'  
}
```

```
{  
  Username: 'Akash',  
  Create_Date: '5-sep-2006'  
}  
{  
  Username: 'ashvanth',  
  Create_Date: '21-nov-2006'  
}  
{  
  Username: 'Chandru',  
  Create_Date: '2-aug-2006'  
}
```

```
{  
  Username: 'ashik',  
  Create_Date: '12-feb-2006'  
}  
{  
  Username: 'raja',  
  Create_Date: '10-jan-2006'  
}
```


8.Retrive all documents from user where location is Bangalore or usagerage grathethanorequalt 30.

Query:

```
db.userDetails.find({$or:[{Location:"Bangalore"},{User_Age:{$gte:30}}]},{Username:1,User_Age:1,_id:0});
```

Output:

```
> db.userDetails.find({$or:[{Location:"Bangalore"},{User_Age:{$gte:30}}]},{Username:1,User_Age:1,_id:0});
< {
  Username: 'Chandru',
  User_Age: 20
}
{
  Username: 'Akash',
  User_Age: 19
}
{
  Username: 'Vishwa',
  User_Age: 30
}
{
  Username: 'naresh',
  User_Age: 30
}
```

9.delete all documents from user collection where active:false.

Query:

```
db.userDetails.deleteMany({active:"false"});
```

Output:

```
> db.userDetails.deleteMany({active:"false"});
< {
  acknowledged: true,
  deletedCount: 4
}
```

10. Retrive name and email fields from user collertions.

Query:

```
db.userDetails.find({}, {Username:1, Email:1});
```

Output:

```
> db.userDetails.find({}, {Username:1, Email:1});
< {
  _id: ObjectId('685ae4cfec4a9c5b95d561b5'),
  Username: 'Akash',
  Email: 'yyy.com'
}
{
  _id: ObjectId('685ae4cfec4a9c5b95d561b7'),
  Username: 'Vishwa',
  Email: 'zzz.com'
}
{
  _id: ObjectId('685ae70cec4a9c5b95d561b8'),
  Username: 'naresh',
  Email: 'aaa.com'
}
{
  _id: ObjectId('685ae70cec4a9c5b95d561b9'),
  Username: 'ashvanth',
  Email: 'bbb.com'
}
```

11. Find the all documents from the collection order with field either status “pending” ,”shipped”.

Query :

```
db.OrderDetails.find({$or:[{Status:"pendding"},{Status:"shipped"}]});
```

Output :

```
> db.OrderDetails.find({$or:[{Status:"pendding"},{Status:"shipped"}]});
< {
  _id: ObjectId('685aa9d3b26ed23464abf988'),
  Order_Site: 'FlipCart',
  Status: 'shipped',
  Delivery_Date: '3-jun-2006'
}
{
  _id: ObjectId('685aa9d3b26ed23464abf98a'),
  Order_Site: 'Insta',
  Status: 'shipped',
  Delivery_Date: null
}
```

12.Delete document from the student collection who's Roll No: is 23BIT0378.

Query;

```
db.StudentInfo.deleteMany({Roll_No:"23BIT0378"});
```

Output:

```
> db.StudentInfo.deleteMany({Roll_No:"23BIT0378"});
< {
  acknowledged: true,
  deletedCount: 1
}
```

13.Retrive all documents from the collection Students who's marks between 60 and 90.

Query:

```
db.StudentInfo.find({mark:{$gte:60,$lte:90}});
```

Output:

```
> db.StudentInfo.find({mark:{$gte:60,$lte:90}});
< {
  _id: ObjectId('685a959e119cba058c7a6b84'),
  name: 'chandru',
  age: 18,
  course: 'Maths',
  mark: 70,
  Roll_No: '23BIT038'
}
{
  _id: ObjectId('685ad77ee1d9b7d2d8a80a99'),
  name: 'Lokesh',
  salary: 20000,
  dept: 'Sales',
  mark: 80,
  Roll_No: '23MIT343'
}
```

```
{
  _id: ObjectId('685ad77ee1d9b7d2d8a80a9a'),
  name: 'Vishwa',
  salary: 30000,
  dept: 'IT',
  mark: 80,
  Roll_No: '23MIT352'
}
```

14.Sort the documents based on price in product collection.

Query:

```
db.ProductDetails.find().sort({price:1});
```

Output:

```
> db.ProductDetails.find().sort({price:1});
< {
  _id: ObjectId('685a9b2b119cba058c7a6b8c'),
  product: 'Computer',
  product_ID: '23CID456',
  price: 10000
}
{
  _id: ObjectId('685a9b2b119cba058c7a6b8d'),
  product: 'Fan',
  product_ID: '28FID456',
  price: 20000
}

{
  _id: ObjectId('685a9b2b119cba058c7a6b8e'),
  product: 'Car',
  product_ID: '21CID456',
  price: 30000
}
{
  _id: ObjectId('685a9b2b119cba058c7a6b8f'),
  product: 'Washing Meashin',
  product_ID: '29WID456',
  price: 40000
}
```

15.Update the order collection status equalt to completed where deliveryDate is not null.

Query:

```
db.OrderDetails.updateMany({Delivery_Date:{$ne:null}},{$set:{Status:"Completed"}});
```

Output:

```
> db.OrderDetails.updateMany({Delivery_Date:{$ne:null}},{$set:{Status:"Completed"}});
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 3,
  modifiedCount: 3,
  upsertedCount: 0
}
```

16.Retrive all documents from Sales collection where amount less that 500 or grater than 5000.

Query:

```
db.SalseDetails.find({$or:[{Amount:{$lte:500}},{Amount:{$gte:5000}}]})
```

Output:

```
> db.SalseDetails.find({$or:[{Amount:{$lte:500}},{Amount:{$gte:5000}}]})
< {
  _id: ObjectId('685aee8f114816546d3b6717'),
  Sales_ID: '34API90',
  Amount: 400
}
{
  _id: ObjectId('685aee8f114816546d3b6719'),
  Sales_ID: '94ASI90',
  Amount: 7000
}
{
  _id: ObjectId('685aee8f114816546d3b671a'),
  Sales_ID: '64NDI90',
  Amount: 500
}
```

17.Create collection called as logsand then skip first 10 documents and retrive next 5 documents form the collection.

Query:

```
db.LogsDetails.find({}, {_id:0}).skip(10).limit(5);
```

Output:

```
> db.LogsDetails.find({}, {_id:0}).skip(10).limit(5);
< {
  user: 'Chandru',
  user_ID: '1MIP2'
}
{
  user: 'Chandru',
  user_ID: '1SEQ2'
}
{
  user: 'Chandru',
  user_ID: '1DV2'
}
{
  user: 'Chandru',
  user_ID: '1BFH2'
}
{
  user: 'Chandru',
  user_ID: '1REG2'
}
```

18.count the documents where Location is Delhi.

Query:

```
db.CustomerDetails.find({Location:"Delhi"}).count();
```

Output:

```
db.CustomerDetails.find({Location:"Delhi"}).count();
3
```

19.Retrieve the documents where Authorname is Bharathiyar and PublishYear is 1901.

Query:

```
db.BookDetails.find({Author:"Bharathiyar",Publish_Year:1901});
```

Output:

```
> db.BookDetails.find({Author:"Bharathiyar",Publish_Year:1901});  
< {  
  _id: ObjectId('685af7e19e8d71a430badec9'),  
  Author: 'Bharathiyar',  
  Publish_Year: 1901  
}
```

-----X-----X-----X-----X-----X-----