# Group B

---

## Assignment No 10

### Title of the Assignment: MongoDB Queries

Design and Develop MongoDB Queries using CRUD operations. (Use CRUD operations, SAVE method, logical operators etc.).

**Objective of the Assignment:** To understand the concept of CURD operations and logical operators.

**Outcome:** Students will be able to learn and understand MySQL/Oracle database Connectivity.

### Theory:

**MongoDB Information:**

MongoDB is a cross-platform, document-oriented database that provides, high performance, high availability, andeasy scalability. MongoDB works on concept of collection and document.

**Database:**

Database is a physical container for collections. Each database gets its own set of files on the file system. A singleMongoDB server typically has multiple databases.

**Collection:**

Collection is a group of MongoDB documents. It is the equivalent of an RDBMS table. A collection exists within asingle database. Collections do not enforce a schema. Documents within a collection can have different fields.
Typically, all documents in a collection are of similar or related purpose.

**Document:**

A document is a set of key-value pairs. Documents have dynamic schema. Dynamic schema means that documentsin the same collection do not need to have the same set of fields or structure, and common fields in a collection's documents may hold different types of data.

**The following table shows the relationship of RDBMS terminology with MongoDB.**

| RDBMS | MongoDB |
|---|---|
| Database | Database |
| Table | Collection |
| Tuple/Row | Document |
| column | Field |

## ❖ Features of MongoDB:

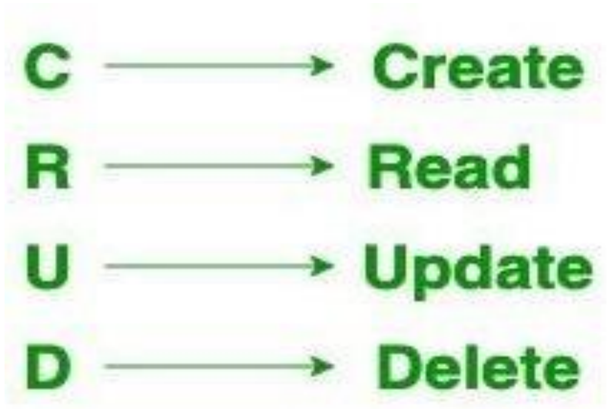
ι. Multiple Servers: It can run over multiple servers.

ιι. Schema-less Database: It is a schema-less database.

ιιι. Indexing: Any field in the document can be indexed.

ιϖ. Rich Object Model: It supports a rich object model.

## MongoDB CRUD operations



## ❖ Create Operations –

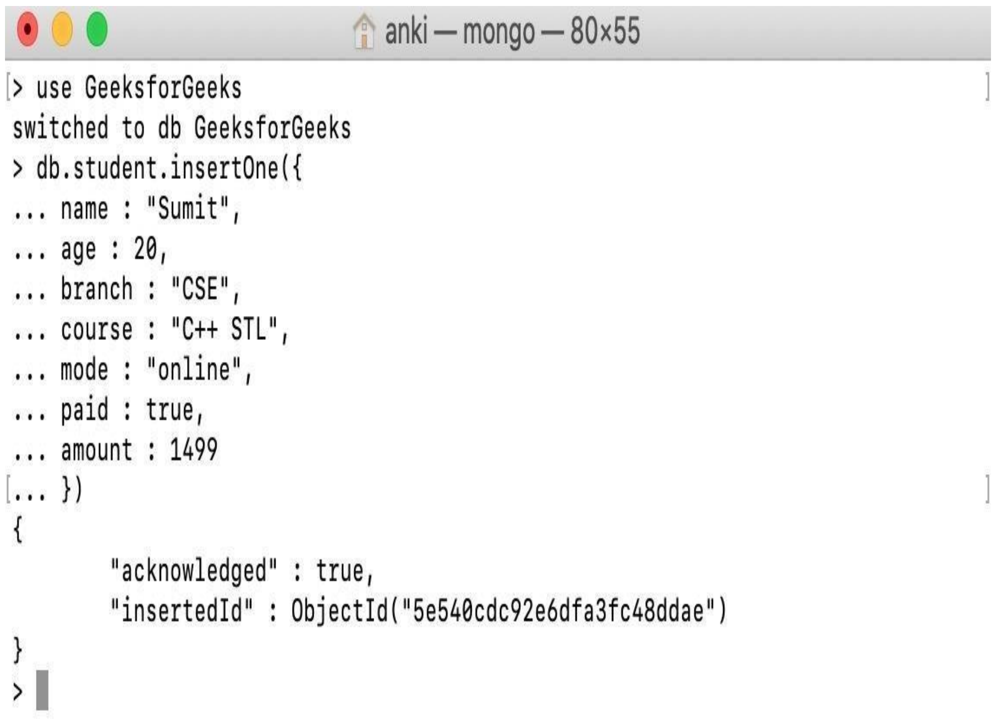
The create or insert operations are used to insert or add new documents in the collection.If a collection does not exist, then it will create a new collection in the database.

You can perform, create operations using the following methods provided by the MongoDB:

| Method | Description |
|---|---|
| db.collection.insertOne() | It is used to insert single document in the collection. |
| db.collection.insertMany() | It is used to insert multiple documents in the collection. |
| db.createCollection() | It is used to create an empty collection. |

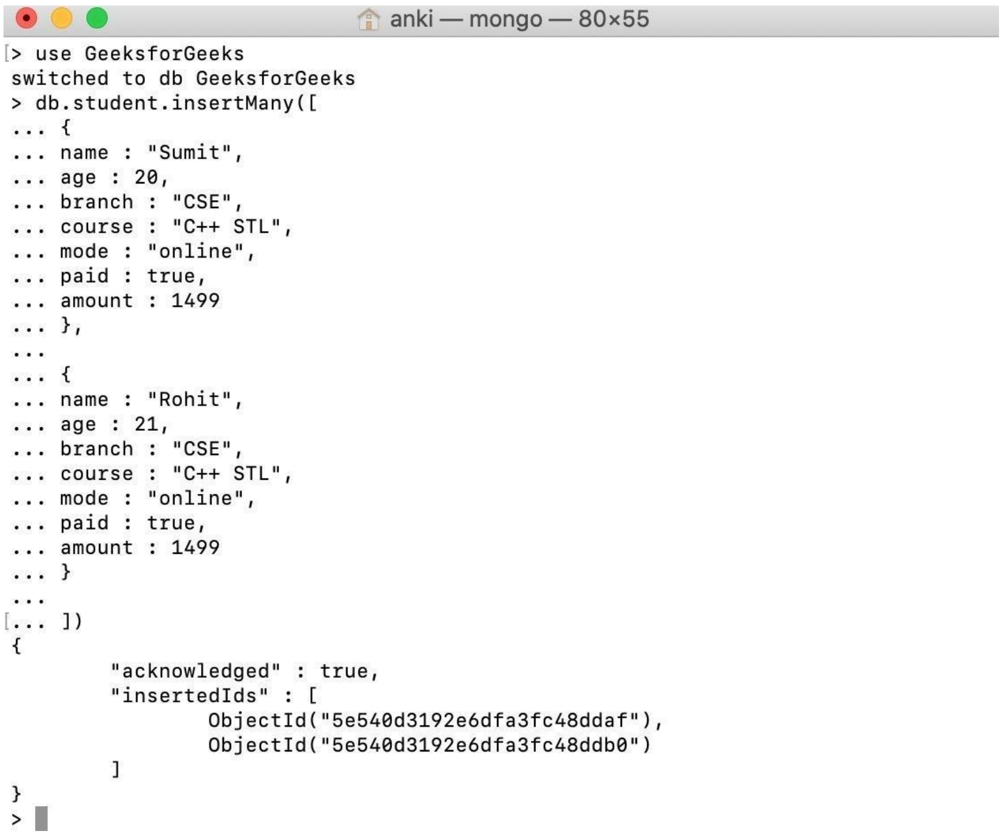
**Example 1:**

In this example, we are inserting details of a single student in the form of document in the student collection using **db.collection.insertOne()** method.



```
> use GeeksforGeeks
switched to db GeeksforGeeks
> db.student.insertOne({
... name : "Sumit",
... age : 20,
... branch : "CSE",
... course : "C++ STL",
... mode : "online",
... paid : true,
... amount : 1499
... })
{
        "acknowledged" : true,
        "insertedId" : ObjectId("5e540cdc92e6dfa3fc48ddae")
}
>
```

**Example 2:**
In this example, we are inserting details of the multiple students in the form of documents in the student collection using **db.collection.insertMany()** method.


anki — mongo — 80×55

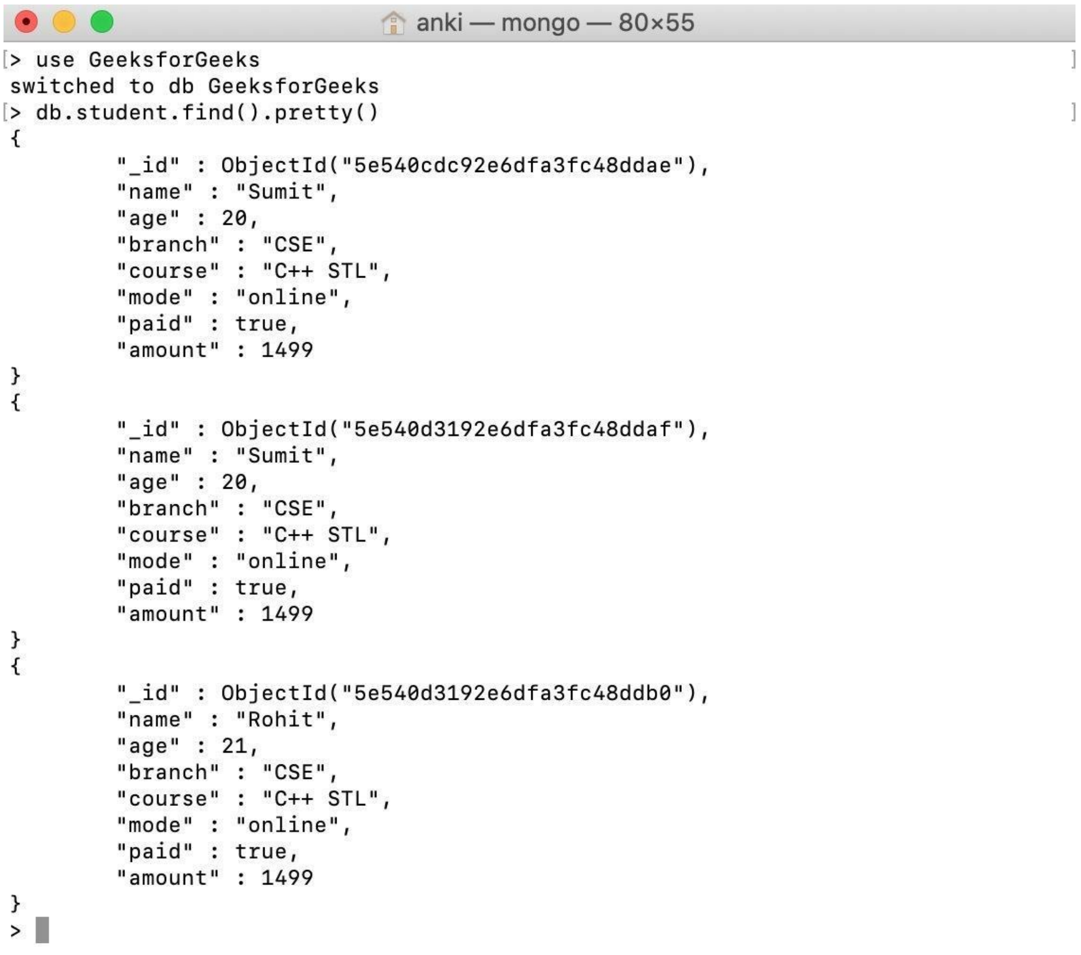
```
[> use GeeksforGeeks
switched to db GeeksforGeeks
> db.student.insertMany([
... {
... name : "Sumit",
... age : 20,
... branch : "CSE",
... course : "C++ STL",
... mode : "online",
... paid : true,
... amount : 1499
... },
...
... {
... name : "Rohit",
... age : 21,
... branch : "CSE",
... course : "C++ STL",
... mode : "online",
... paid : true,
... amount : 1499
... }
...
[... ])
{
        "acknowledged" : true,
        "insertedIds" : [
                ObjectId("5e540d3192e6dfa3fc48ddaf"),
                ObjectId("5e540d3192e6dfa3fc48ddb0")
        ]
}
>
```

❖ **Read Operations –**
The Read operations are used to retrieve documents from the collection, or in other words, read operations are usedto query a collection for a document.

You can perform read operation using the following method provided by the MongoDB:

| Method | Description |
|---|---|
| db.collection.find() | It is used to retrieve documents from the collection. |

In this example, we are retrieving the details of students from the student collection using **db.collection.find()** method.



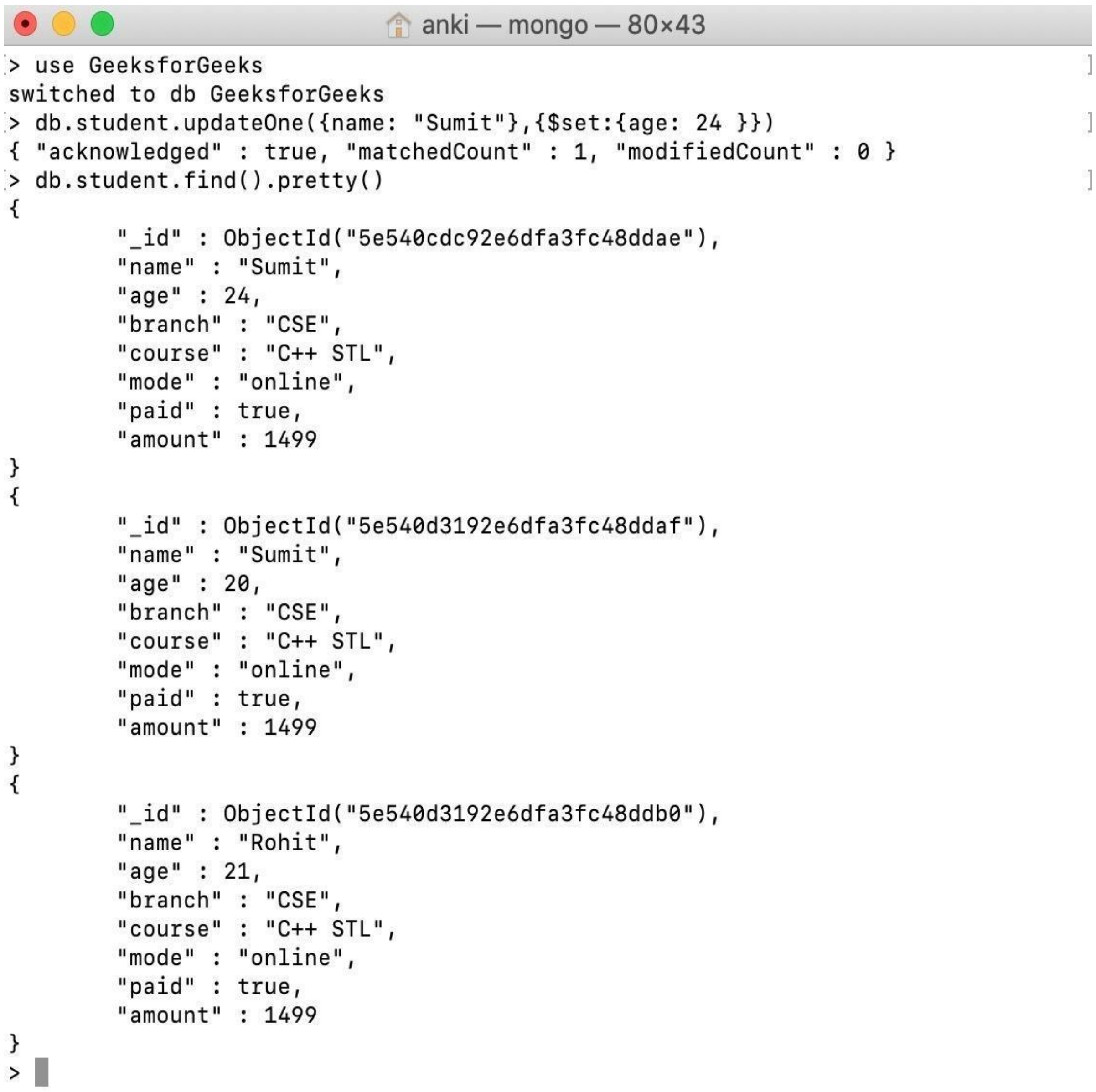
```
> use GeeksforGeeks
switched to db GeeksforGeeks
> db.student.find().pretty()
{
        "_id" : ObjectId("5e540cdc92e6dfa3fc48ddae"),
        "name" : "Sumit",
        "age" : 20,
        "branch" : "CSE",
        "course" : "C++ STL",
        "mode" : "online",
        "paid" : true,
        "amount" : 1499
}
{
        "_id" : ObjectId("5e540d3192e6dfa3fc48ddaf"),
        "name" : "Sumit",
        "age" : 20,
        "branch" : "CSE",
        "course" : "C++ STL",
        "mode" : "online",
        "paid" : true,
        "amount" : 1499
}
{
        "_id" : ObjectId("5e540d3192e6dfa3fc48ddb0"),
        "name" : "Rohit",
        "age" : 21,
        "branch" : "CSE",
        "course" : "C++ STL",
        "mode" : "online",
        "paid" : true,
        "amount" : 1499
}
>
```

## ❖ Update Operations –

The update operations are used to update or modify the existing document in the collection. You can perform update operations using the following methods provided by the MongoDB:

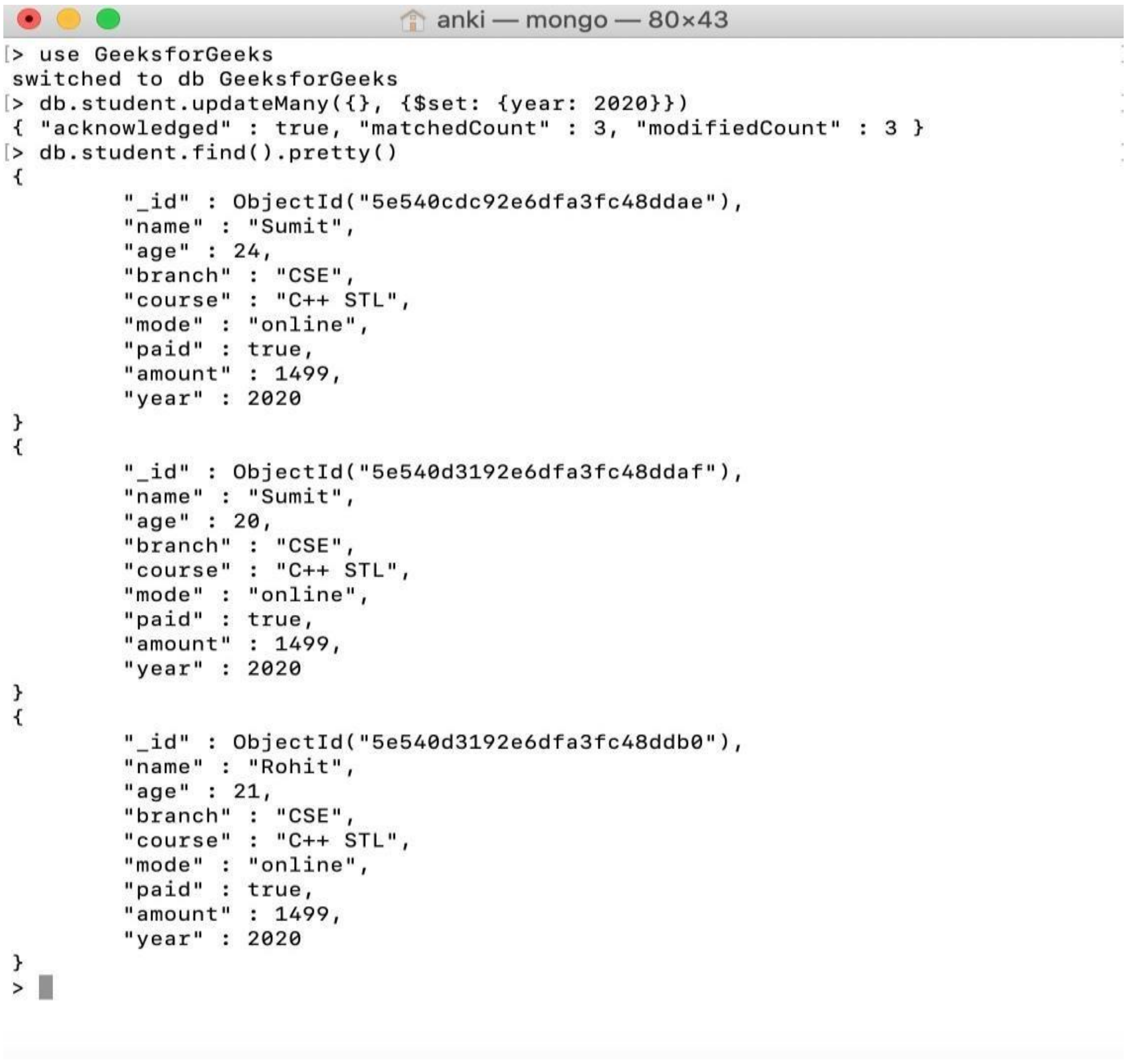
| Method | Description |
|---|---|
| db.collection.updateOne() | It is used to update a single document in the collection that satisfy the given criteria. |
| db.collection.updateMany() | It is used to update multiple documents in the collection that satisfy the given criteria. |
| db.collection.replaceOne() | It is used to replace single document in the collection that satisfy the given criteria. |

**Example 1:**
In this example, we are updating the age of Sumit in the student collection using **db.collection.updateOne()** method.



```
> use GeeksforGeeks
switched to db GeeksforGeeks
> db.student.updateOne({name: "Sumit"},{$set:{age: 24 }})
{ "acknowledged" : true, "matchedCount" : 1, "modifiedCount" : 0 }
> db.student.find().pretty()
{
        "_id" : ObjectId("5e540cdc92e6dfa3fc48ddae"),
        "name" : "Sumit",
        "age" : 24,
        "branch" : "CSE",
        "course" : "C++ STL",
        "mode" : "online",
        "paid" : true,
        "amount" : 1499
}
{
        "_id" : ObjectId("5e540d3192e6dfa3fc48ddaf"),
        "name" : "Sumit",
        "age" : 20,
        "branch" : "CSE",
        "course" : "C++ STL",
        "mode" : "online",
        "paid" : true,
        "amount" : 1499
}
{
        "_id" : ObjectId("5e540d3192e6dfa3fc48ddb0"),
        "name" : "Rohit",
        "age" : 21,
        "branch" : "CSE",
        "course" : "C++ STL",
        "mode" : "online",
        "paid" : true,
        "amount" : 1499
}
> 
```

**Example 2:**
In this example, we are updating the year of course in all the documents in the student collection using **db.collection.updateMany()** method.



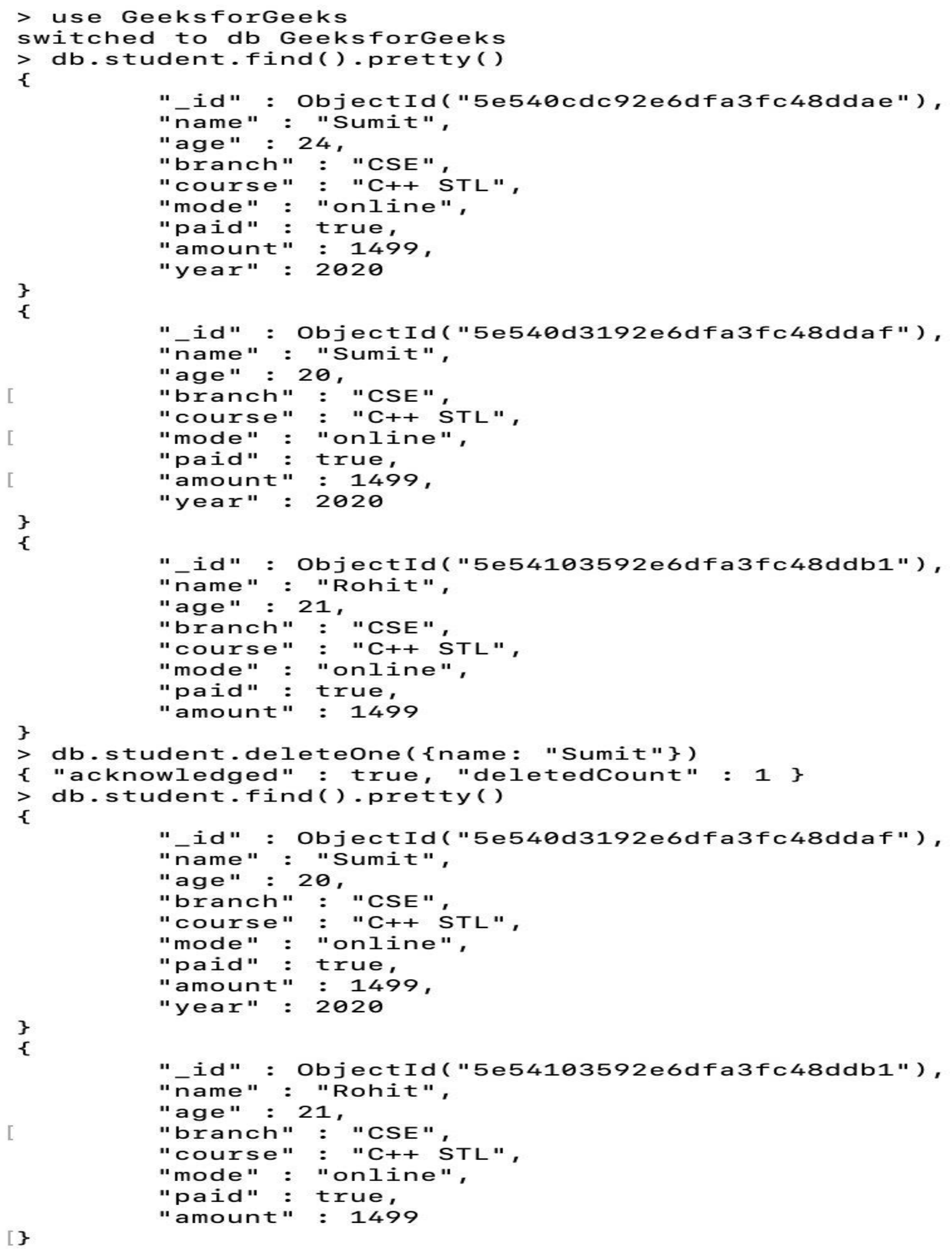
```
> use GeeksforGeeks
switched to db GeeksforGeeks
> db.student.updateMany({}, {$set: {year: 2020}})
{ "acknowledged" : true, "matchedCount" : 3, "modifiedCount" : 3 }
> db.student.find().pretty()
{
        "_id" : ObjectId("5e540cdc92e6dfa3fc48ddae"),
        "name" : "Sumit",
        "age" : 24,
        "branch" : "CSE",
        "course" : "C++ STL",
        "mode" : "online",
        "paid" : true,
        "amount" : 1499,
        "year" : 2020
}
{
        "_id" : ObjectId("5e540d3192e6dfa3fc48ddaf"),
        "name" : "Sumit",
        "age" : 20,
        "branch" : "CSE",
        "course" : "C++ STL",
        "mode" : "online",
        "paid" : true,
        "amount" : 1499,
        "year" : 2020
}
{
        "_id" : ObjectId("5e540d3192e6dfa3fc48ddb0"),
        "name" : "Rohit",
        "age" : 21,
        "branch" : "CSE",
        "course" : "C++ STL",
        "mode" : "online",
        "paid" : true,
        "amount" : 1499,
        "year" : 2020
}
> 
```

## ❖ Delete Operations –

The delete operation are used to delete or remove the documents from a collection. You can perform delete operations using the following methods provided by the MongoDB:

| Method | Description: |
|---|---|
| db.collection.deleteOne() | It is used to delete a single document from the collection that satisfy the given criteria. |
| db.collection.deleteMany() | It is used to delete multiple documents from the collection that satisfy the given criteria. |

**Example 1:**
In this example, we are deleting a document from the student collection using **db.collection.deleteOne**() method.



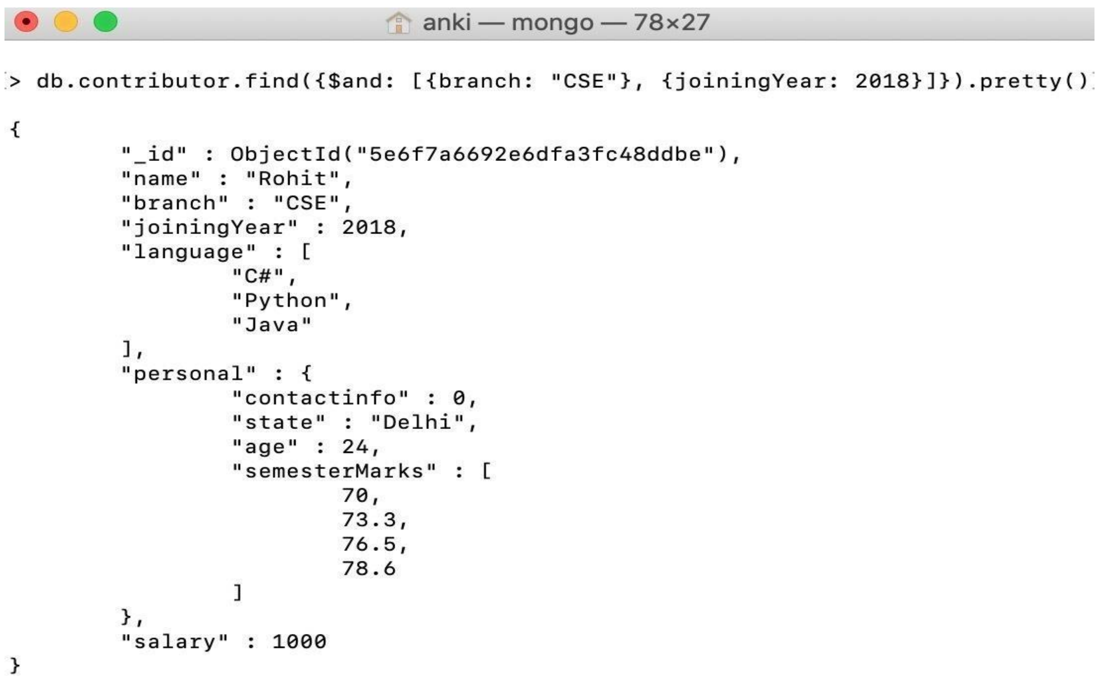
```
> use GeeksforGeeks
switched to db GeeksforGeeks
> db.student.find().pretty()
{
        "_id" : ObjectId("5e540cdc92e6dfa3fc48ddae"),
        "name" : "Sumit",
        "age" : 24,
        "branch" : "CSE",
        "course" : "C++ STL",
        "mode" : "online",
        "paid" : true,
        "amount" : 1499,
        "year" : 2020
}
{
        "_id" : ObjectId("5e540d3192e6dfa3fc48ddaf"),
        "name" : "Sumit",
        "age" : 20,
[       "branch" : "CSE",
        "course" : "C++ STL",
[       "mode" : "online",
        "paid" : true,
[       "amount" : 1499,
        "year" : 2020
}
{
        "_id" : ObjectId("5e54103592e6dfa3fc48ddb1"),
        "name" : "Rohit",
        "age" : 21,
        "branch" : "CSE",
        "course" : "C++ STL",
        "mode" : "online",
        "paid" : true,
        "amount" : 1499
}
> db.student.deleteOne({name: "Sumit"})
{ "acknowledged" : true, "deletedCount" : 1 }
> db.student.find().pretty()
{
        "_id" : ObjectId("5e540d3192e6dfa3fc48ddaf"),
        "name" : "Sumit",
        "age" : 20,
        "branch" : "CSE",
        "course" : "C++ STL",
        "mode" : "online",
        "paid" : true,
        "amount" : 1499,
        "year" : 2020
}
{
        "_id" : ObjectId("5e54103592e6dfa3fc48ddb1"),
        "name" : "Rohit",
        "age" : 21,
[       "branch" : "CSE",
        "course" : "C++ STL",
        "mode" : "online",
        "paid" : true,
        "amount" : 1499
[}
```

## MongoDB – Logical Query Operators

- MongoDB supports logical query operators.
- These operators are used for filtering the data and getting precise results based on the given conditions.
- The following table contains the comparison query operators:

1. $and
   It is used to join query clauses with a logical AND and return all documents that match the given conditions ofboth clauses.

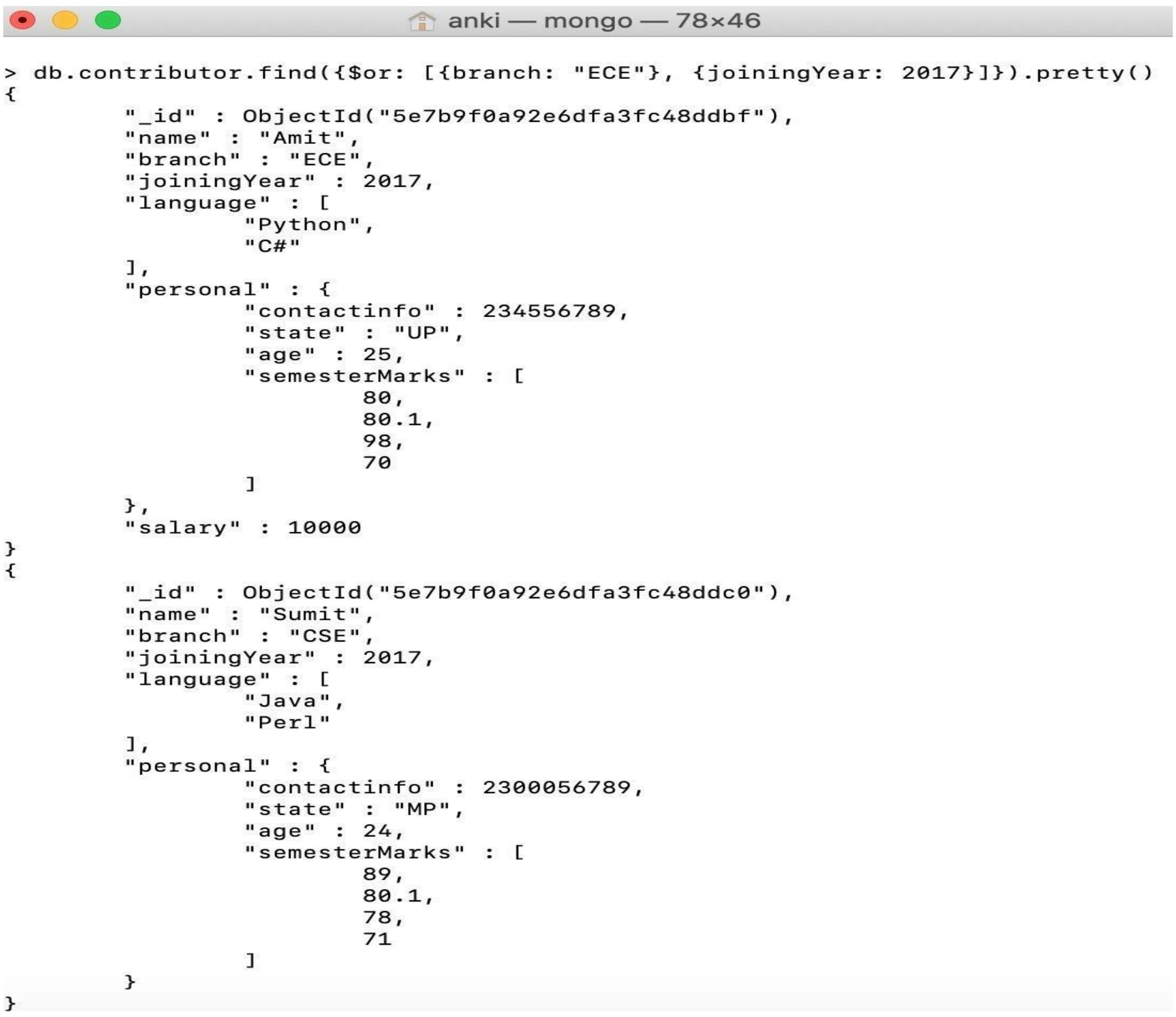
**Example:**
Query:
db.contributor.find({$and: [{branch: "CSE"}, {joiningYear: 2018}]}).pretty()



```
> db.contributor.find({$and: [{branch: "CSE"}, {joiningYear: 2018}]}).pretty()
{
        "_id" : ObjectId("5e6f7a6692e6dfa3fc48ddbe"),
        "name" : "Rohit",
        "branch" : "CSE",
        "joiningYear" : 2018,
        "language" : [
                "C#",
                "Python",
                "Java"
        ],
        "personal" : {
                "contactinfo" : 0,
                "state" : "Delhi",
                "age" : 24,
                "semesterMarks" : [
                        70,
                        73.3,
                        76.5,
                        78.6
                ]
        },
        "salary" : 1000
}
```

2. $or
   It is used to join query clauses with a logical OR and return all documents that match the given conditions ofeither clause.

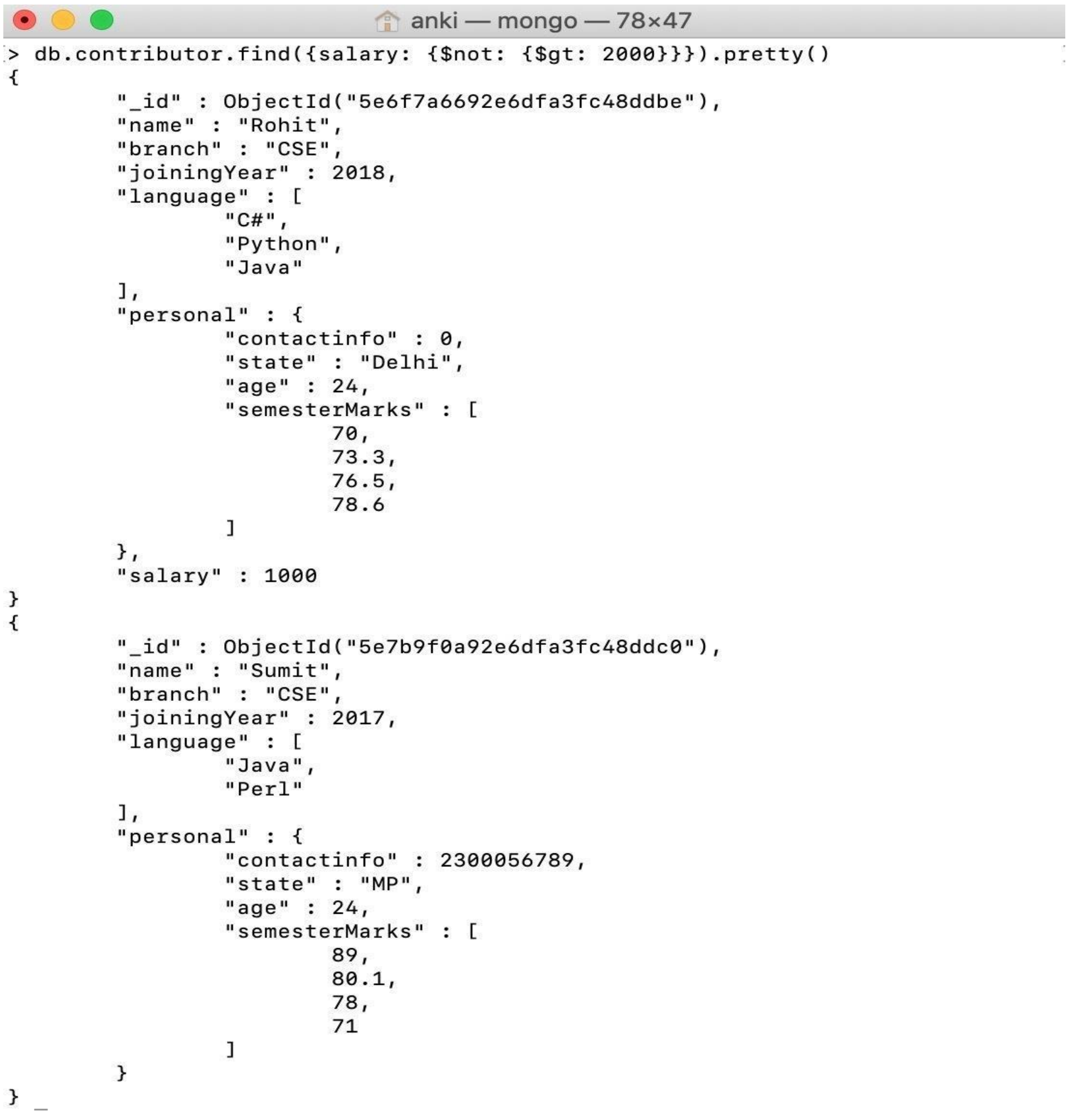
   **Example:**
   Query:
   db.contributor.find({$or: [{branch: "ECE"}, {joiningYear: 2017}]}).pretty()



```
> db.contributor.find({$or: [{branch: "ECE"}, {joiningYear: 2017}]}).pretty()
{
        "_id" : ObjectId("5e7b9f0a92e6dfa3fc48ddbf"),
        "name" : "Amit",
        "branch" : "ECE",
        "joiningYear" : 2017,
        "language" : [
                "Python",
                "C#"
        ],
        "personal" : {
                "contactinfo" : 234556789,
                "state" : "UP",
                "age" : 25,
                "semesterMarks" : [
                        80,
                        80.1,
                        98,
                        70
                ]
        },
        "salary" : 10000
}
{
        "_id" : ObjectId("5e7b9f0a92e6dfa3fc48ddc0"),
        "name" : "Sumit",
        "branch" : "CSE",
        "joiningYear" : 2017,
        "language" : [
                "Java",
                "Perl"
        ],
        "personal" : {
                "contactinfo" : 2300056789,
                "state" : "MP",
                "age" : 24,
                "semesterMarks" : [
                        89,
                        80.1,
                        78,
                        71
                ]
        }
}
```

3. $not
   It is used to invert the effect of the query expressions and return documents that does not match the query expression

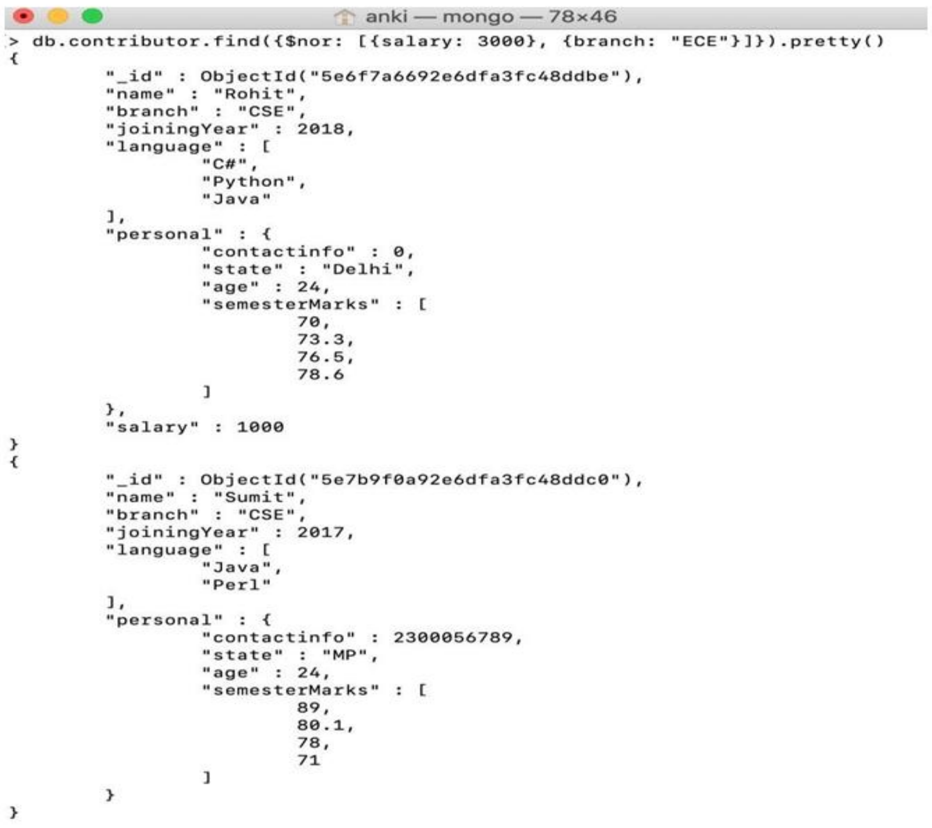
   **Example:**
   Query:
   db.contributor.find({salary: {$not: {$gt: 2000}}}).pretty()



```
> db.contributor.find({salary: {$not: {$gt: 2000}}}).pretty()
{
        "_id" : ObjectId("5e6f7a6692e6dfa3fc48ddbe"),
        "name" : "Rohit",
        "branch" : "CSE",
        "joiningYear" : 2018,
        "language" : [
                "C#",
                "Python",
                "Java"
        ],
        "personal" : {
                "contactinfo" : 0,
                "state" : "Delhi",
                "age" : 24,
                "semesterMarks" : [
                        70,
                        73.3,
                        76.5,
                        78.6
                ]
        },
        "salary" : 1000
}
{
        "_id" : ObjectId("5e7b9f0a92e6dfa3fc48ddc0"),
        "name" : "Sumit",
        "branch" : "CSE",
        "joiningYear" : 2017,
        "language" : [
                "Java",
                "Perl"
        ],
        "personal" : {
                "contactinfo" : 2300056789,
                "state" : "MP",
                "age" : 24,
                "semesterMarks" : [
                        89,
                        80.1,
                        78,
                        71
                ]
        }
} _
```

4. $nor
   It is used to join query clauses with a logical NOR and return all documents that fail to match both clauses.

   **Example:**
   Query:
   db.contributor.find({$nor: [{salary: 3000}, {branch: "ECE"}]}).pretty()



```
> db.contributor.find({$nor: [{salary: 3000}, {branch: "ECE"}]}).pretty()
{
        "_id" : ObjectId("5e6f7a6692e6dfa3fc48ddbe"),
        "name" : "Rohit",
        "branch" : "CSE",
        "joiningYear" : 2018,
        "language" : [
                "C#",
                "Python",
                "Java"
        ],
        "personal" : {
                "contactinfo" : 0,
                "state" : "Delhi",
                "age" : 24,
                "semesterMarks" : [
                        70,
                        73.3,
                        76.5,
                        78.6
                ]
        },
        "salary" : 1000
}
{
        "_id" : ObjectId("5e7b9f0a92e6dfa3fc48ddc0"),
        "name" : "Sumit",
        "branch" : "CSE",
        "joiningYear" : 2017,
        "language" : [
                "Java",
                "Perl"
        ],
        "personal" : {
                "contactinfo" : 2300056789,
                "state" : "MP",
                "age" : 24,
                "semesterMarks" : [
                        89,
                        80.1,
                        78,
                        71
                ]
        }
}
```

**Conclusion:** Performed and implement the CURD operations and logical operators.

## Viva Question:

- What is MongoDB?
- What are some of the advantages of MongoDB??
- What is a Document and collection in MongoDB?
- What are the CURD operations?
- Write the all logical operators with examples?

| | |
|---|---|
| **Date:** | |
| **Marks obtained:** | |
| **Sign of course coordinator:** | |
| **Name of course Coordinator:** | |