MAI172: Advance Database Technologies

Register Number: 2448513

Name: Deshmukh Pratik Bhushanrao

Experiment Number and Name: 7: Demonstration of CRUD Operations on MongoDB.

Date: 29/08/2024 Time: 9.45 to 11.45

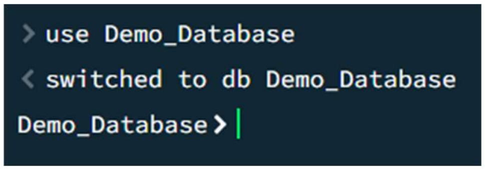
---

1. Create a Database "Demo_Database"

Query:

use Demo_Database

Output:



```
> use Demo_Database
< switched to db Demo_Database
Demo_Database >
```

Inference:

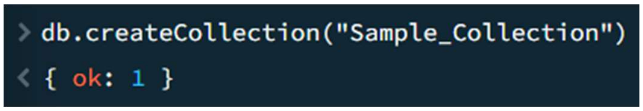
Created a database for our task.

2. Create a Collection "Sample_Collection" in the Demo_Database

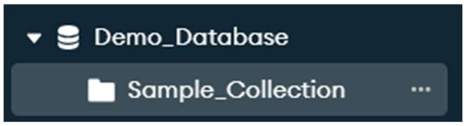
Query:

db.createCollection("Sample_Collection")

Output:



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



Inference:

Created a collection for our task.

3. Perform Insert, Read, Update and Remove operations using Sample_Collection

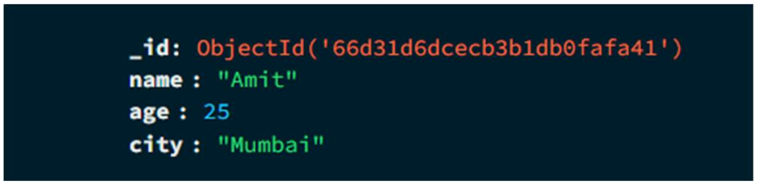
Query:

Insert queries:

1. Query:

db.collection.insertOne({ name: "Alice", age: 25, city: "New York" })

Output:



```
_id: ObjectId('66d31d6dcecb3b1db0fafa41')
name : "Amit"
age : 25
city : "Mumbai"
```

Inference: Adds a new document with the specified fields

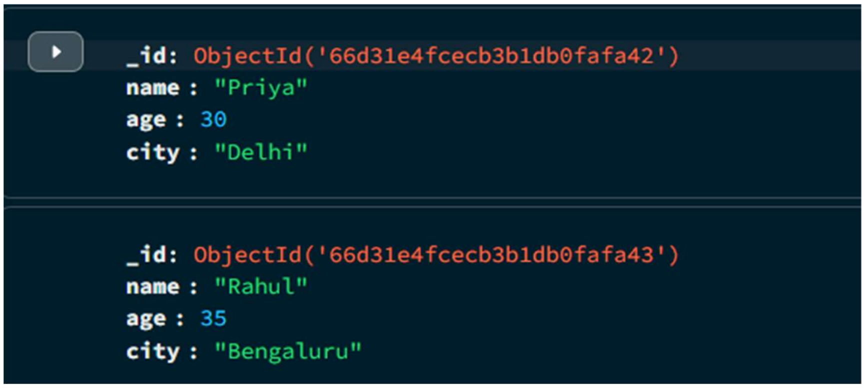
2. Query:

db.Sample_Collection.insertMany([

{ name: "Priya", age: 30, city: "Delhi" },

{ name: "Rahul", age: 35, city: "Bengaluru" }]);

Output:



```
_id: ObjectId('66d31e4fcecb3b1db0fafa42')
name : "Priya"
age : 30
city : "Delhi"

_id: ObjectId('66d31e4fcecb3b1db0fafa43')
name : "Rahul"
age : 35
city : "Bengaluru"
```

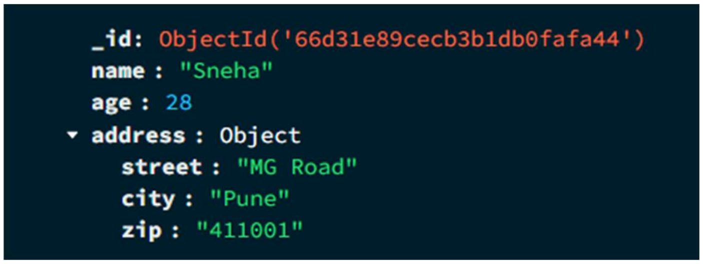
Inference: Adds multiple new documents to the Sample_Collection in a single operation

3. Query:

db.Sample_Collection.insertOne({ name: "Sneha", age: 28,

address: { street: "MG Road", city: "Pune", zip: "411001" }});

Output:



```
  _id: ObjectId('66d31e89cecb3b1db0fafa44')
  name : "Sneha"
  age : 28
▾ address : Object
    street : "MG Road"
    city : "Pune"
    zip : "411001"
```

Inference:

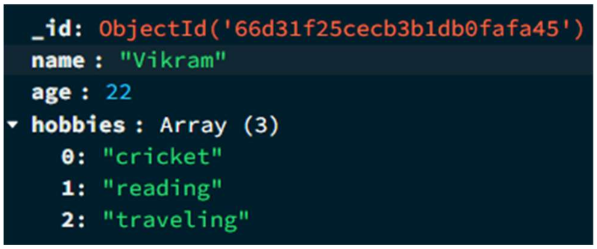
Adds a new document with nested fields (e.g., an address object) to the Sample_Collection.

4. Query:

db.Sample_Collection.insertOne({ name: "Vikram", age: 22,

hobbies: ["cricket", "reading", "traveling"]});

Output:



```
  _id: ObjectId('66d31f25cecb3b1db0fafa45')
  name : "Vikram"
  age : 22
▾ hobbies : Array (3)
    0: "cricket"
    1: "reading"
    2: "traveling"
```

Inference:

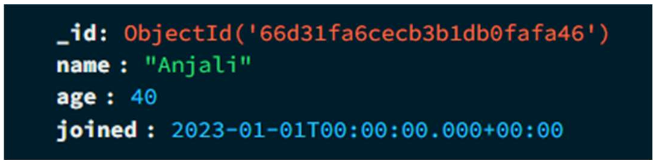
Adds a new document with an array field (e.g., hobbies) to the Sample_Collection.

5. Query:

db.Sample_Collection.insertOne({ name: "Anjali", age: 40,

joined: new Date("2023-01-01")});

Output:



```
_id: ObjectId('66d31fa6cecb3b1db0fafa46')
name : "Anjali"
age : 40
joined : 2023-01-01T00:00:00.000+00:00
```

Inference:

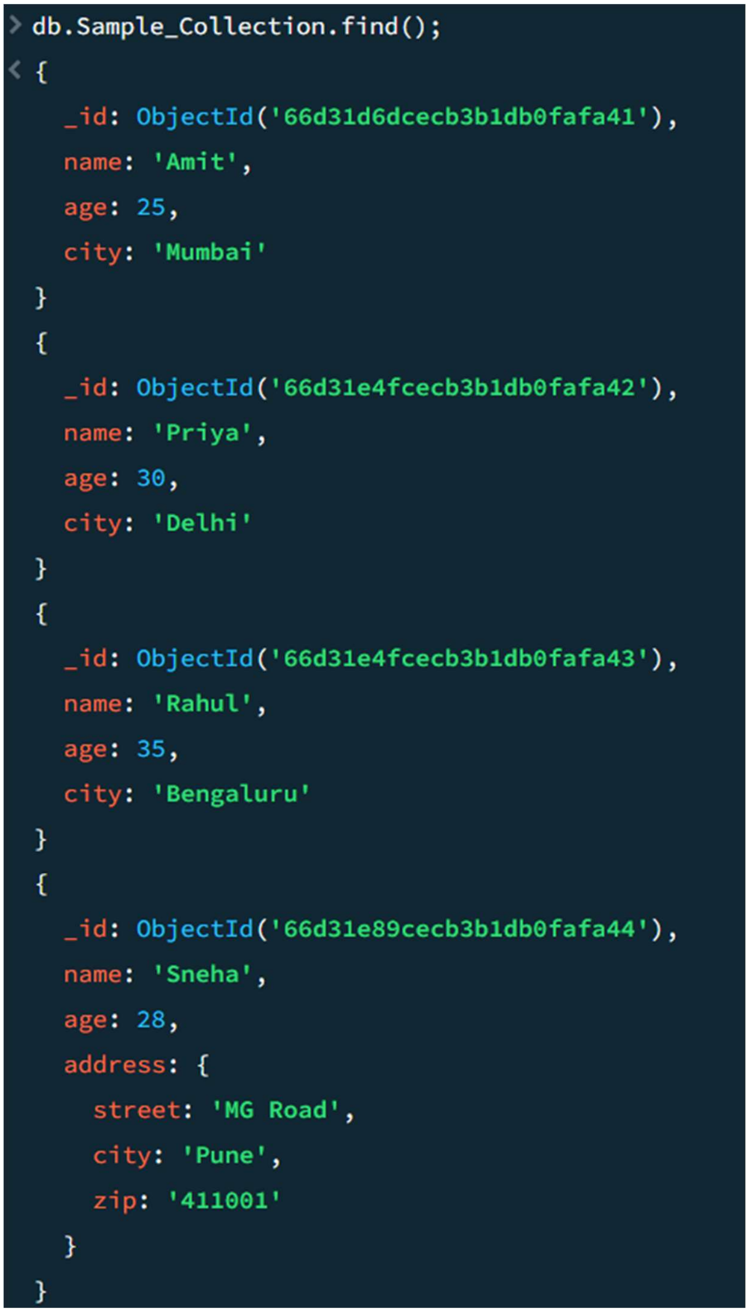
Adds a new document with a date field to the Sample_Collection.

Read queries

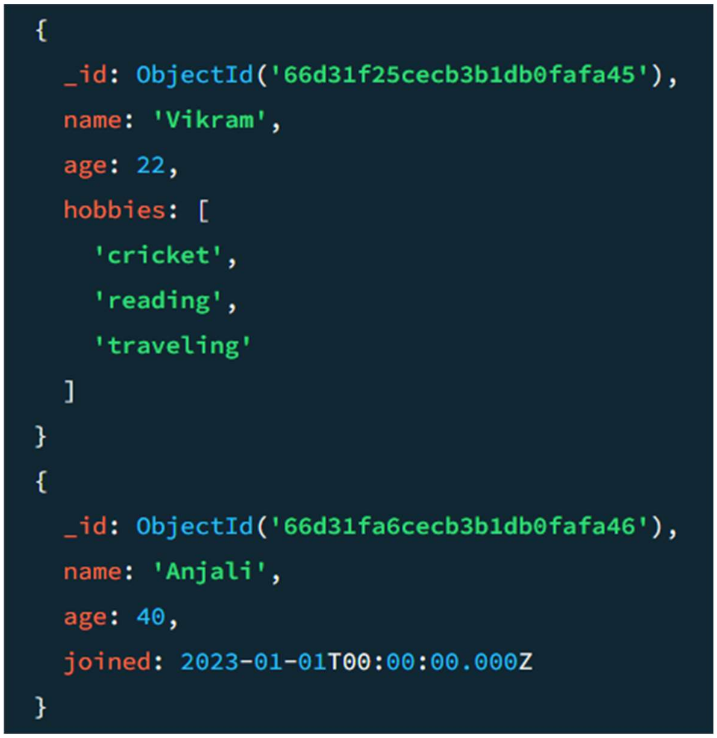
1. Query:

db.Sample_Collection.find();

Output:



```
> db.Sample_Collection.find();
< {
    _id: ObjectId('66d31d6dcecb3b1db0fafa41'),
    name: 'Amit',
    age: 25,
    city: 'Mumbai'
  }
  {
    _id: ObjectId('66d31e4fcecb3b1db0fafa42'),
    name: 'Priya',
    age: 30,
    city: 'Delhi'
  }
  {
    _id: ObjectId('66d31e4fcecb3b1db0fafa43'),
    name: 'Rahul',
    age: 35,
    city: 'Bengaluru'
  }
  {
    _id: ObjectId('66d31e89cecb3b1db0fafa44'),
    name: 'Sneha',
    age: 28,
    address: {
      street: 'MG Road',
      city: 'Pune',
      zip: '411001'
    }
  }
```



```
{
  _id: ObjectId('66d31f25cecb3b1db0fafa45'),
  name: 'Vikram',
  age: 22,
  hobbies: [
    'cricket',
    'reading',
    'traveling'
  ]
}
{
  _id: ObjectId('66d31fa6cecb3b1db0fafa46'),
  name: 'Anjali',
  age: 40,
  joined: 2023-01-01T00:00:00.000Z
}
```

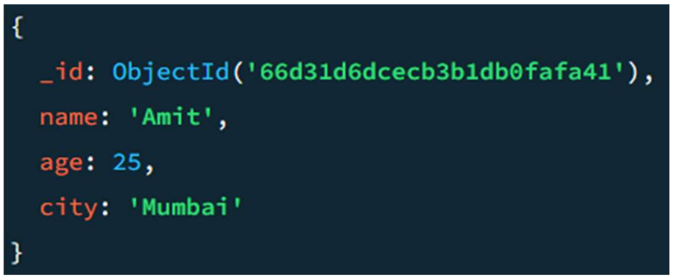
Inference:

Retrieves all documents from the Sample_Collection.

2. Query:

db.Sample_Collection.find({ city: "Mumbai" });

Output:



```
{
  _id: ObjectId('66d31d6dcecb3b1db0fafa41'),
  name: 'Amit',
  age: 25,
  city: 'Mumbai'
}
```

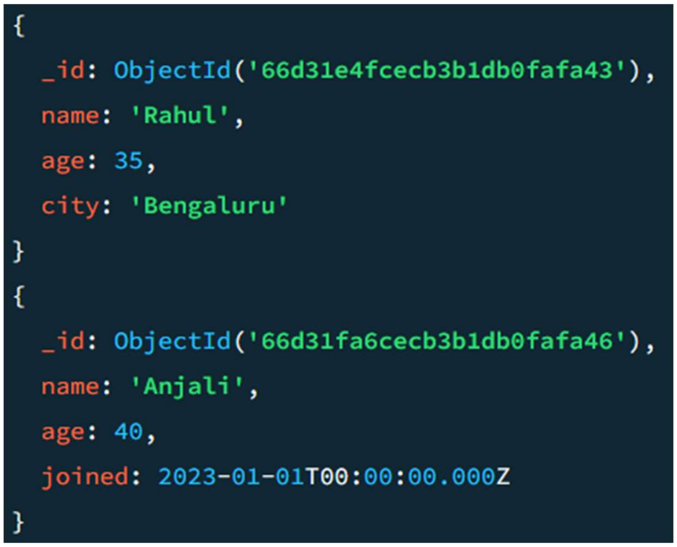
Inference:

Retrieves documents from the Sample_Collection where the specified field matches a given value.

3. Query:

db.Sample_Collection.find({ age: { $gt: 30 } });

Output:



```
{
  _id: ObjectId('66d31e4fcecb3b1db0fafa43'),
  name: 'Rahul',
  age: 35,
  city: 'Bengaluru'
}
{
  _id: ObjectId('66d31fa6cecb3b1db0fafa46'),
  name: 'Anjali',
  age: 40,
  joined: 2023-01-01T00:00:00.000Z
}
```

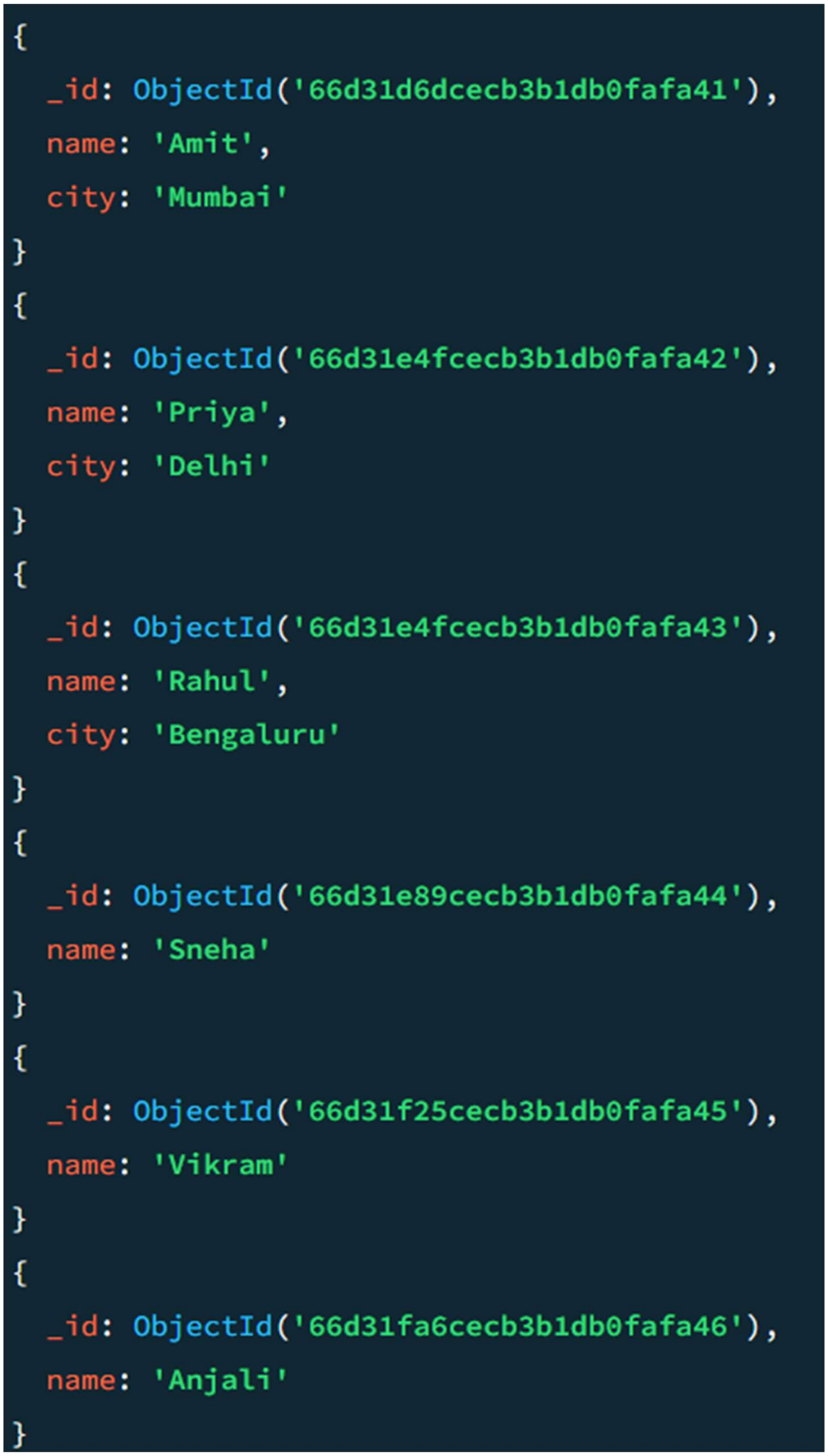
Inference:

Retrieves documents from the Sample_Collection that meet a specified condition (e.g., age greater than 30).

4. Query:

```
db.Sample_Collection.find({}, { name: 1, city: 1 });
```

Output:



```
{
  _id: ObjectId('66d31d6dcecb3b1db0fafa41'),
  name: 'Amit',
  city: 'Mumbai'
}
{
  _id: ObjectId('66d31e4fcecb3b1db0fafa42'),
  name: 'Priya',
  city: 'Delhi'
}
{
  _id: ObjectId('66d31e4fcecb3b1db0fafa43'),
  name: 'Rahul',
  city: 'Bengaluru'
}
{
  _id: ObjectId('66d31e89cecb3b1db0fafa44'),
  name: 'Sneha'
}
{
  _id: ObjectId('66d31f25cecb3b1db0fafa45'),
  name: 'Vikram'
}
{
  _id: ObjectId('66d31fa6cecb3b1db0fafa46'),
  name: 'Anjali'
}
```

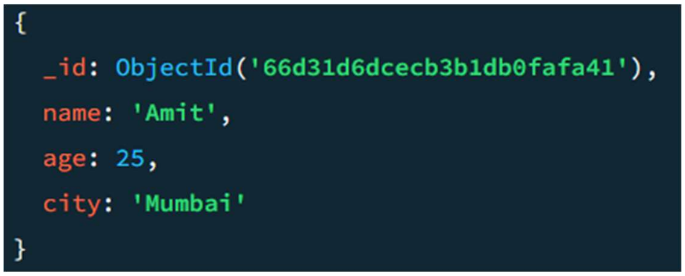
Inference:

Retrieves documents from the Sample_Collection but only includes specified fields in the output.

5. Query:

db.Sample_Collection.findOne({ name: "Amit" });

Output:



```
{
  _id: ObjectId('66d31d6dcecb3b1db0fafa41'),
  name: 'Amit',
  age: 25,
  city: 'Mumbai'
}
```

Inference:

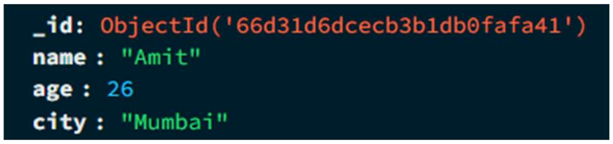
Retrieves a single document from the Sample_Collection that matches the specified criteria.

Update query:

1. **Query:**

db.Sample_Collection.updateOne({ name: "Amit" }, { $set: { age: 26 } });

Output:



```
_id: ObjectId('66d31d6dcecb3b1db0fafa41')
name : "Amit"
age : 26
city : "Mumbai"
```

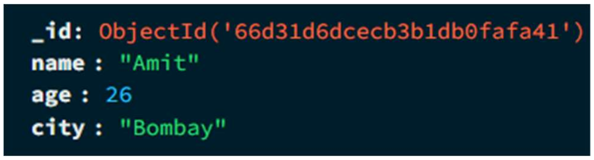
Inference:

Updates the first document that matches the specified criteria.

2. **Query:**

db.Sample_Collection.updateMany({ city: "Mumbai" }, { $set: { city: "Bombay" } });

Output:



```
_id: ObjectId('66d31d6dcecb3b1db0fafa41')
name : "Amit"
age : 26
city : "Bombay"
```

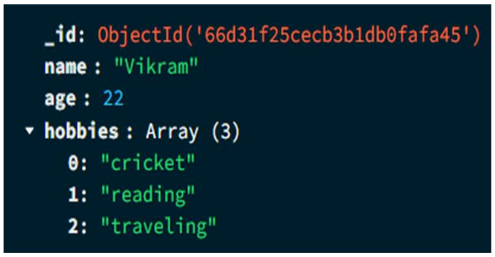
Inference:

Updates all documents that match the specified criteria

3. **Query:**

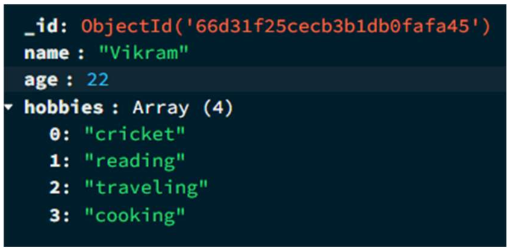
db.Sample_Collection.updateOne({{ name: "Vikram" }, { $push: { hobbies: "cooking" }})

Output:



```
_id: ObjectId('66d31f25cecb3b1db0fafa45')
name : "Vikram"
age : 22
▾ hobbies : Array (3)
    0: "cricket"
    1: "reading"
    2: "traveling"
```

→



```
_id: ObjectId('66d31f25cecb3b1db0fafa45')
name : "Vikram"
age : 22
▾ hobbies : Array (4)
    0: "cricket"
    1: "reading"
    2: "traveling"
    3: "cooking"
```

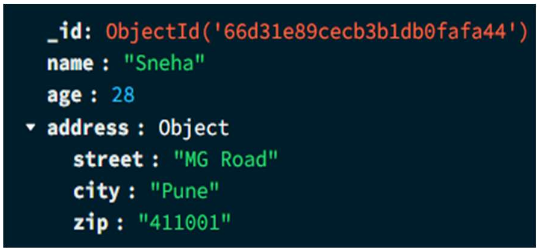
Inference:

Updates a document by adding a new element to an array field.

4. Query:

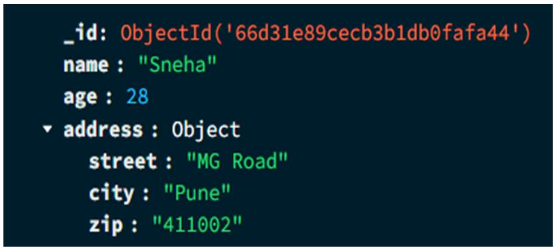
db.Sample_Collection.updateOne({ name: "Sneha" }, { $set: { "address.zip": "411002" } });

Output:



```
  _id: ObjectId('66d31e89cecb3b1db0fafa44')
  name : "Sneha"
  age : 28
▾ address : Object
    street : "MG Road"
    city : "Pune"
    zip : "411001"
```

→



```
  _id: ObjectId('66d31e89cecb3b1db0fafa44')
  name : "Sneha"
  age : 28
▾ address : Object
    street : "MG Road"
    city : "Pune"
    zip : "411002"
```

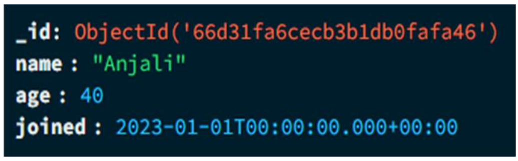
Inference:

Updates a nested field within a document.

5. Query:

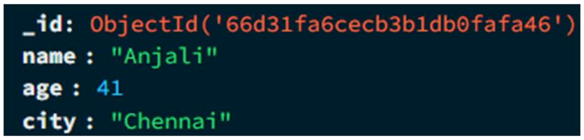
db.Sample_Collection.replaceOne({ name: "Anjali" }, { name: "Anjali", age: 41, city: "Chennai" });

Output:



```
_id: ObjectId('66d31fa6cecb3b1db0fafa46')
name : "Anjali"
age : 40
joined : 2023-01-01T00:00:00.000+00:00
```

→



```
_id: ObjectId('66d31fa6cecb3b1db0fafa46')
name : "Anjali"
age : 41
city : "Chennai"
```

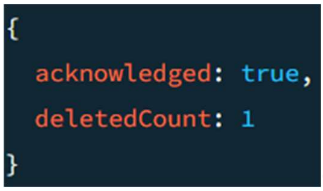
Inference:

Replaces an entire document with a new document.

Delete queries:

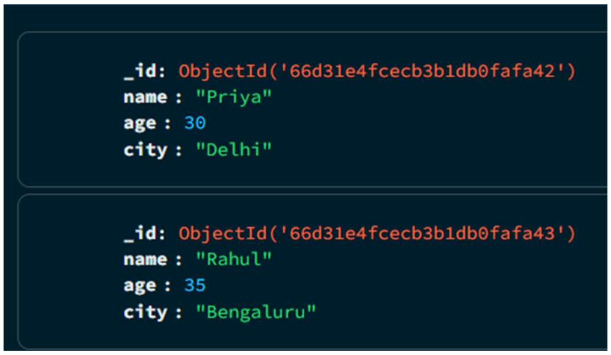
1. Query: db.Sample_Collection.deleteOne({ name: "Amit" });

Output:



```
{
  acknowledged: true,
  deletedCount: 1
}
```

→



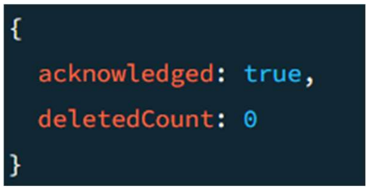
```
_id: ObjectId('66d31e4fcecb3b1db0fafa42')
name : "Priya"
age : 30
city : "Delhi"

_id: ObjectId('66d31e4fcecb3b1db0fafa43')
name : "Rahul"
age : 35
city : "Bengaluru"
```

Inference:

Deletes the first document that matches the specified criteria from the Sample_Collection.

2. Query: db.Sample_Collection.deleteMany({ city: "Bombay" });

Output:



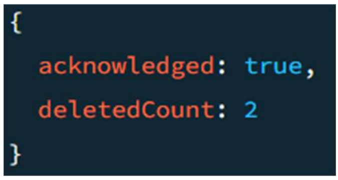
```
{
  acknowledged: true,
  deletedCount: 0
}
```

Inference:

Deletes all documents that match the specified criteria from the Sample_Collection.

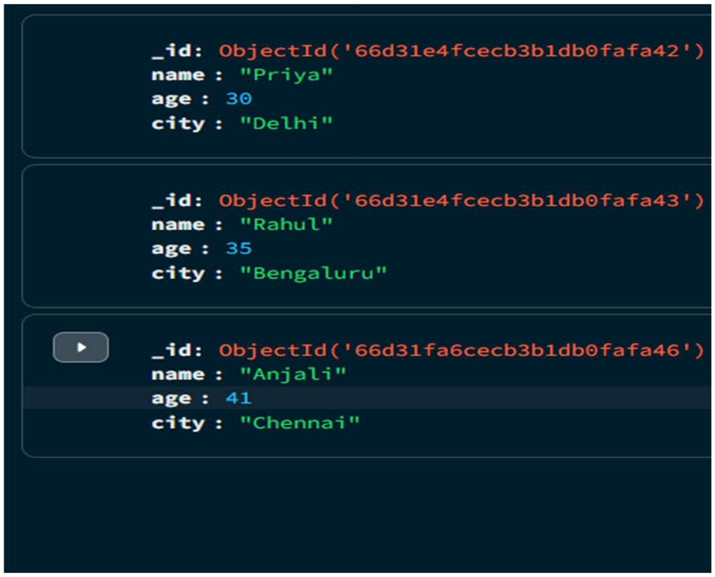
3. Query: db.Sample_Collection.deleteMany({ age: { $lt: 30 } });

Output:



```
{
  acknowledged: true,
  deletedCount: 2
}
```

→



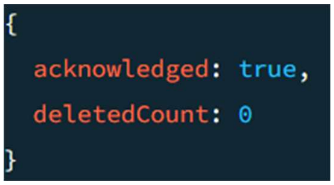
```
_id: ObjectId('66d31e4fcecb3b1db0fafa42')
name : "Priya"
age : 30
city : "Delhi"

_id: ObjectId('66d31e4fcecb3b1db0fafa43')
name : "Rahul"
age : 35
city : "Bengaluru"

_id: ObjectId('66d31fa6cecb3b1db0fafa46')
name : "Anjali"
age : 41
city : "Chennai"
```

Inference:

Deletes documents that meet a specified condition from the Sample_Collection.

4. Query: db.Sample_Collection.deleteOne({ "address.zip": "411002" });

Output:



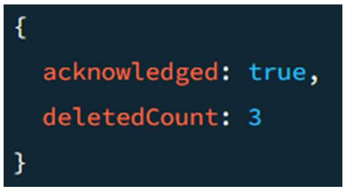
```
{
  acknowledged: true,
  deletedCount: 0
}
```

Inference:

Deletes a document that contains a nested field matching the specified criteria from the Sample_Collection.

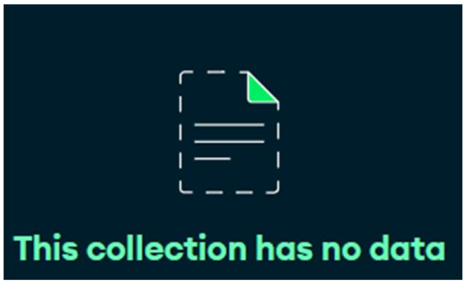
5. Query: db.Sample_Collection.deleteMany({});

Output:



```
{
  acknowledged: true,
  deletedCount: 3
}
```

→



This collection has no data

Inference:

Deletes all documents from the Sample_Collection.