

Web Technology

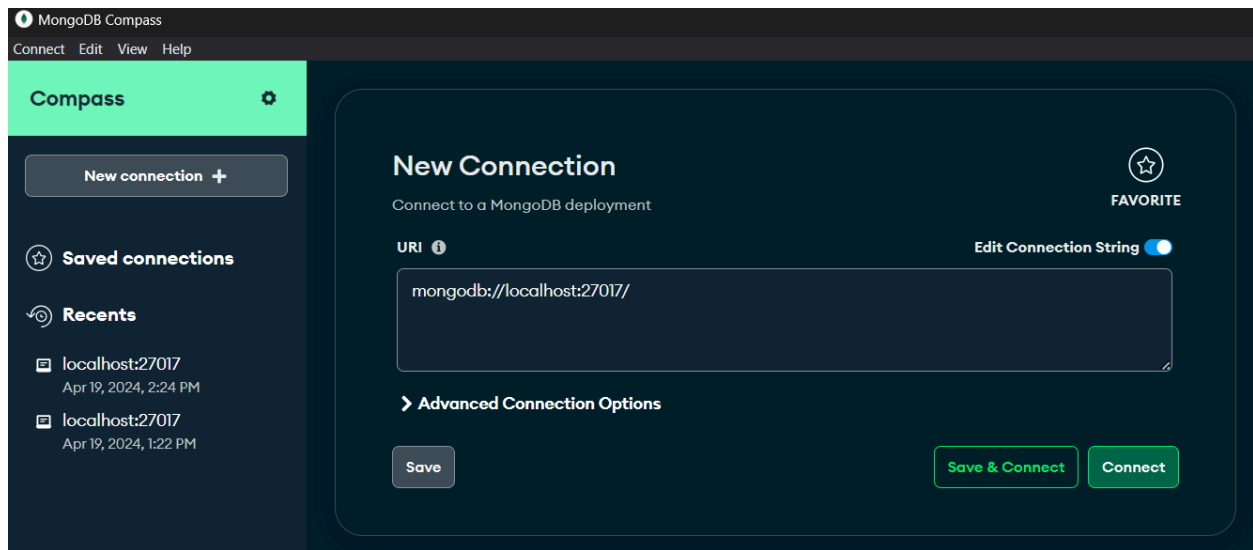
Lab Assignment 9

Name-Tanish Maheshwari

Rollno-22MC3035

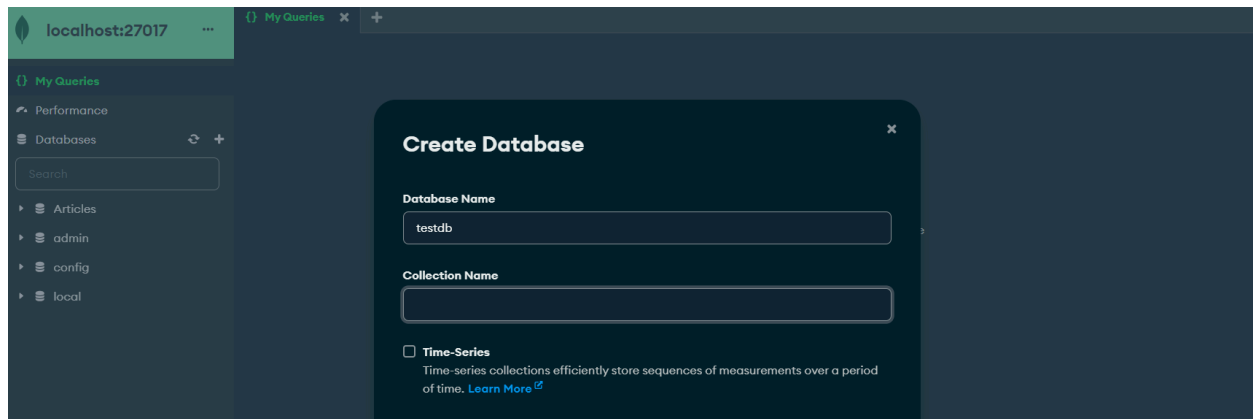
Q1 Connect to a MongoDB server using MongoDB Compass.

Ans

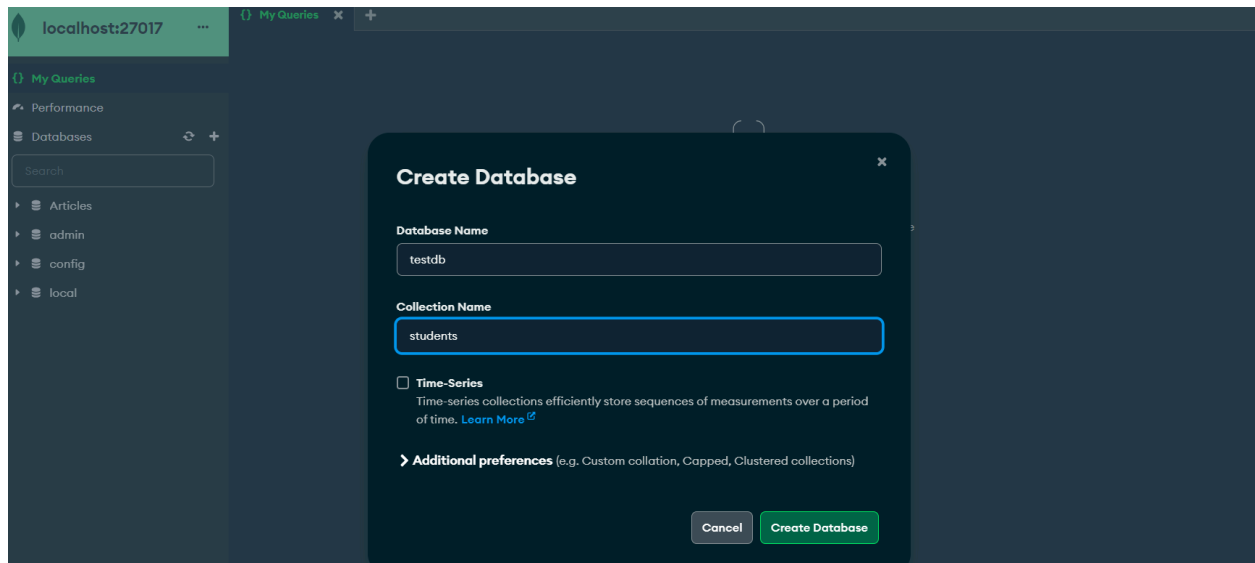


Q2. Create a new database named "testdb" in MongoDB Compass.

Ans



Q3. Create a new collection named "students" in the "testdb" database.
Ans



Q4. Insert ten documents into the "students" collection with the following fields:
name, age, and email.

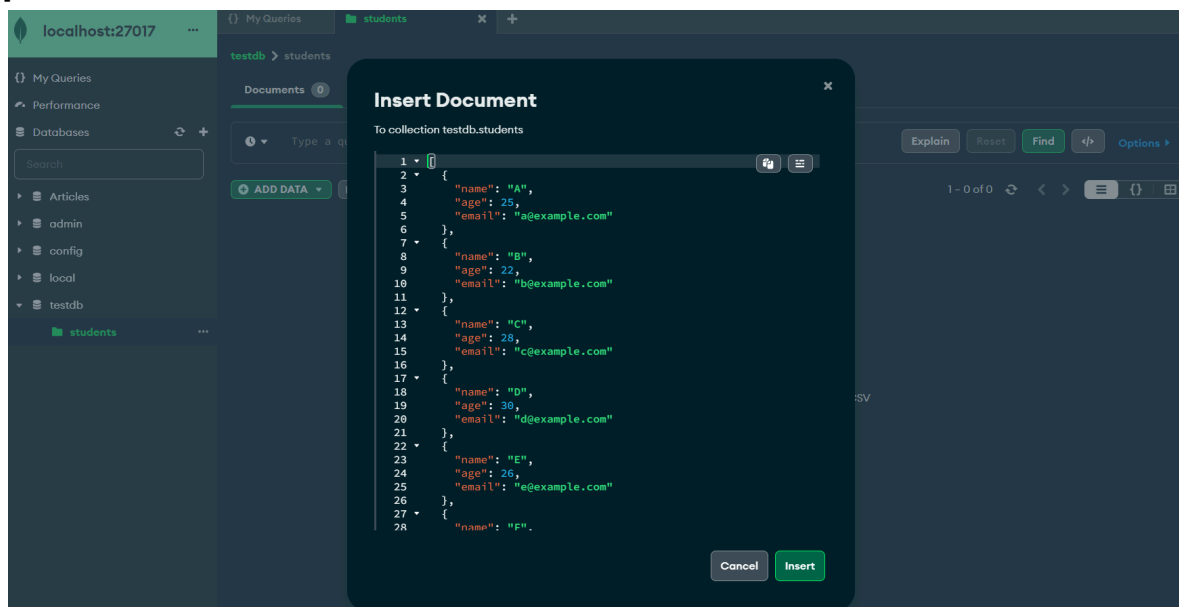
Ans

```
[
  {
    "name": "A",
    "age": 25,
    "email": "a@example.com"
  },
  {
    "name": "B",
    "age": 22,
    "email": "b@example.com"
  },
  {
    "name": "C",
    "age": 28,
    "email": "c@example.com"
  },
  {
    "name": "D",
    "age": 30,
    "email": "d@example.com"
  },
  {
```

```

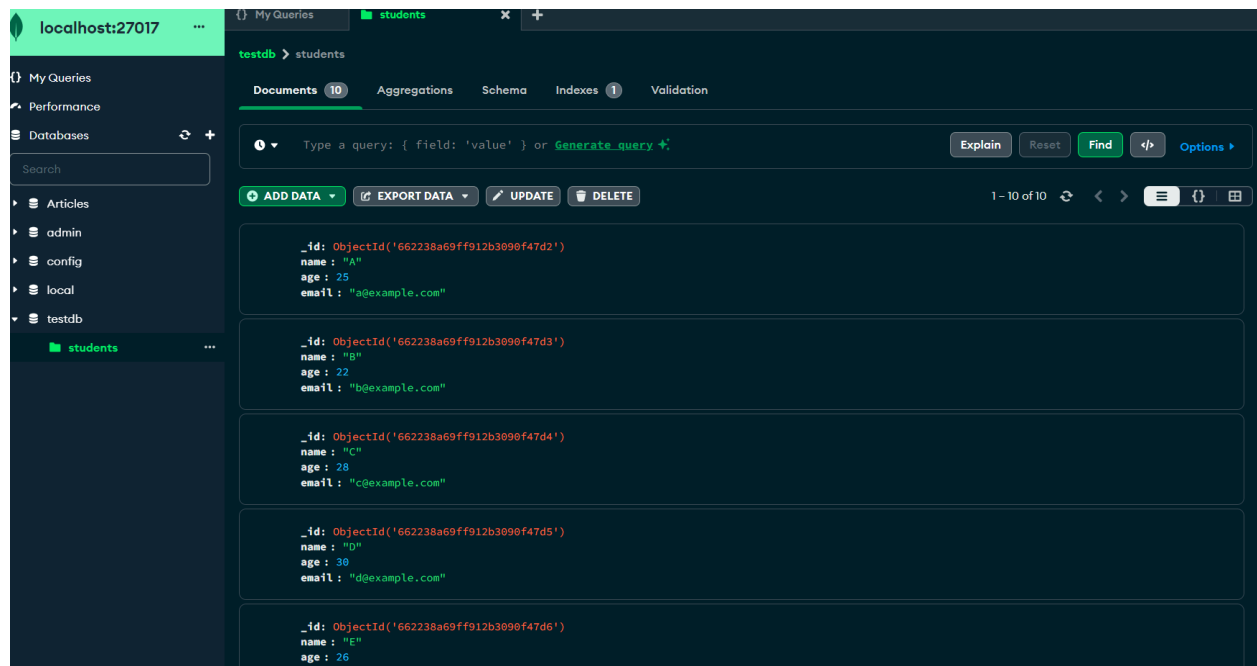
    "name": "E",
    "age": 26,
    "email": "e@example.com"
  },
  {
    "name": "F",
    "age": 24,
    "email": "f@example.com"
  },
  {
    "name": "G",
    "age": 27,
    "email": "g@example.com"
  },
  {
    "name": "H",
    "age": 29,
    "email": "h@example.com"
  },
  {
    "name": "I",
    "age": 23,
    "email": "i@example.com"
  },
  {
    "name": "J",
    "age": 31,
    "email": "j@example.com"
  }
]

```



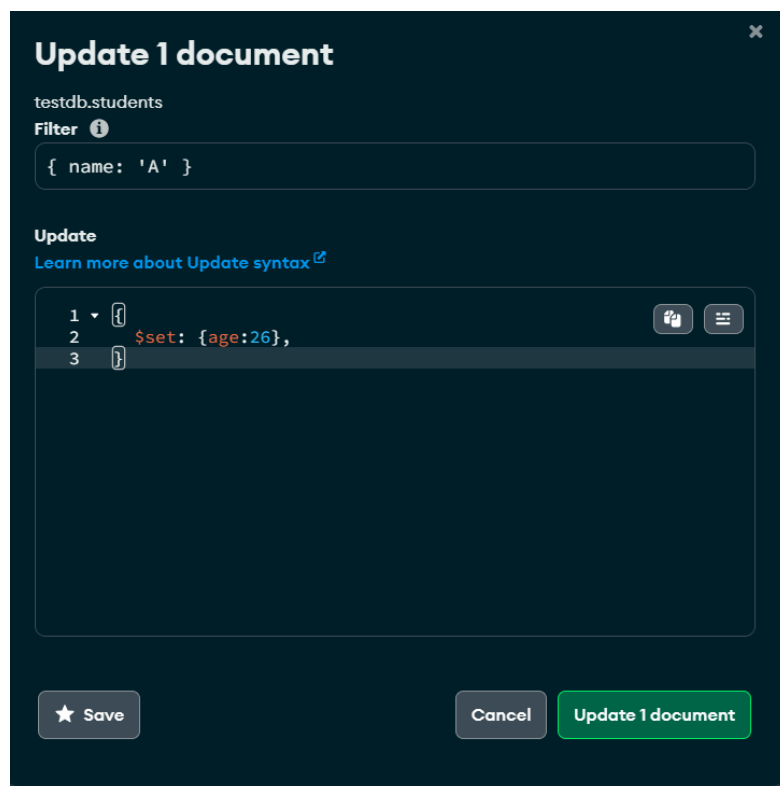
Q5. View the contents of the "students" collection.

Ans



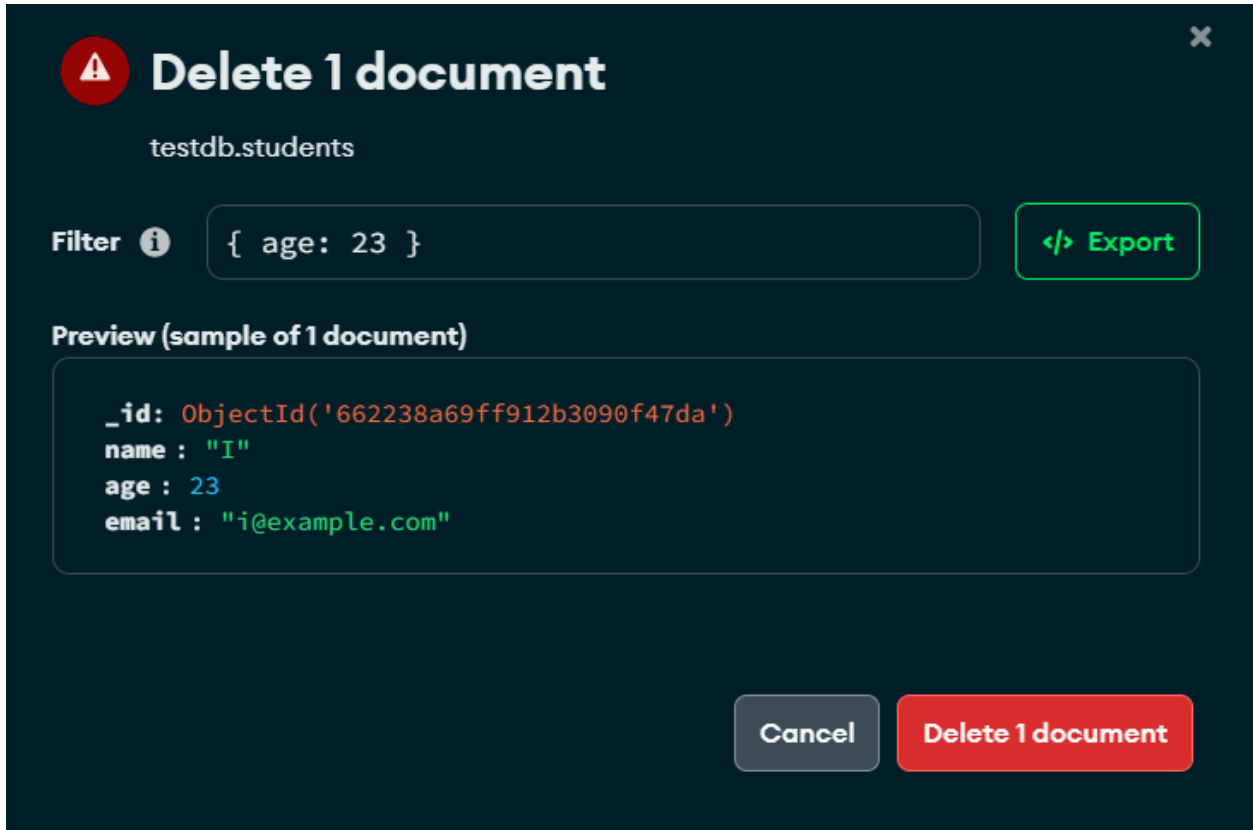
Q6. Update the age of a specific student in the "students" collection.

Ans




Q7. Delete a document from the "students" collection based on a specific condition.


Ans



Delete 1 document

testdb.students

Filter  { age: 23 }

 Export

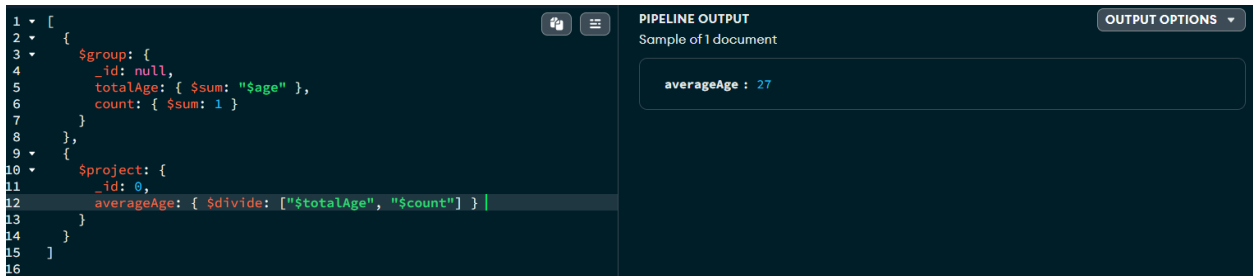
Preview (sample of 1 document)

```
_id: ObjectId('662238a69ff912b3090f47da')
name : "I"
age : 23
email : "i@example.com"
```

Cancel Delete 1 document

Q8. Use the aggregation pipeline to calculate the average age of all students in the "students" collection.

Ans



```
1 [
2   {
3     $group: {
4       _id: null,
5       totalAge: { $sum: "$age" },
6       count: { $sum: 1 }
7     },
8   },
9   {
10    $project: {
11      _id: 0,
12      averageAge: { $divide: [ "$totalAge", "$count" ] }
13    }
14  }
15 ]
16
```

PIPELINE OUTPUT

Sample of 1 document

averageAge : 27

OUTPUT OPTIONS

Q9. Create an index on the "name" field in the "students" collection.

Ans

Create Index

testdb.students

Index fields

name

text

+

> Options

Cancel

Create Index

Q10. Export the contents of the "students" collection to a JSON file.

Ans

Export

Collection testdb.students

Export File Type

JSON

CSV

> Advanced JSON Format

Cancel

Export...

Q11. Perform a complex aggregation operation to find the top 5 oldest students in the "students" collection.

Ans

The screenshot shows the MongoDB Compass interface. On the left, a pipeline is defined with two stages: a \$sort stage with { age: -1 } and a \$limit stage with 5. The \$project stage is highlighted. On the right, the 'PIPELINE OUTPUT' tab shows a sample of 5 documents. Each document contains fields for name, age, and email.

```
1 [
2   {
3     $sort: { age: -1 }
4   },
5   {
6     $limit: 5
7   },
8   {
9     $project: {
10      _id: 0,
11      name: 1,
12      age: 1,
13      email: 1
14    }
15  }
16 ]
```

PIPELINE OUTPUT
Sample of 5 documents

- name: "J", age: 31, email: "j@example.com"
- name: "D", age: 30, email: "d@example.com"
- name: "H", age: 29, email: "h@example.com"
- name: "C", age: 28, email: "c@example.com"
- name: "G", age: 27, email: "g@example.com"

Q12. Create a geospatial index on a field representing the location of students.

Ans

The screenshot shows the 'INDEXES' tab in MongoDB Compass. It displays a table of indexes for the 'students' collection. The table has columns for Name and Definition, Type, Size, Usage, and Properties. Three indexes are listed: _id_ (REGULAR, UNIQUE), name_text (TEXT), and students_2dsphere (GEOSPATIAL).

Name and Definition	Type	Size	Usage	Properties
> _id_	REGULAR <i>i</i>	36.9 KB	3 (since Fri Apr 19 2024)	UNIQUE <i>i</i>
> name_text	TEXT <i>i</i>	20.5 KB	0 (since Fri Apr 19 2024)	
> students_2dsphere	GEOSPATIAL <i>i</i>	8.2 KB	0 (since Fri Apr 19 2024)	

Q13. Use MongoDB Compass to visualize the data distribution in the "students" collection.

Ans



Q14. Set up a data validation rule to ensure that documents in the "students" collection must have a non-empty name field.

Ans

The screenshot displays the MongoDB Compass interface for a collection, specifically the **Validation** tab. The **Validation Action** is set to **Error** and the **Validation Level** is set to **Strict**. The validation rule is defined as follows:

```
1 {
2   "bsonType": "object",
3   "required": ["name"],
4   "properties": {
5     "name": {
6       "bsonType": "string",
7       "minLength": 1
8     }
9   }
10 }
```

Validation modified

Cancel Update