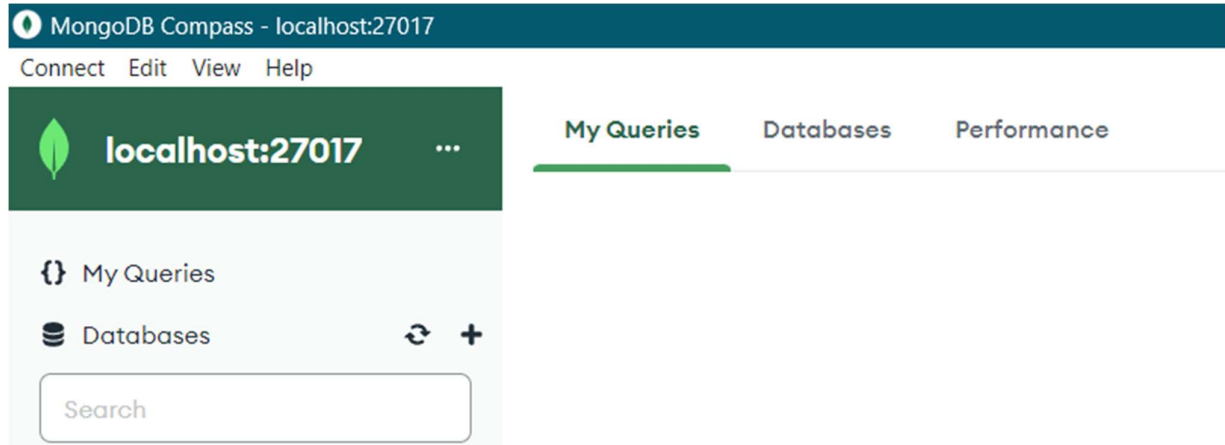
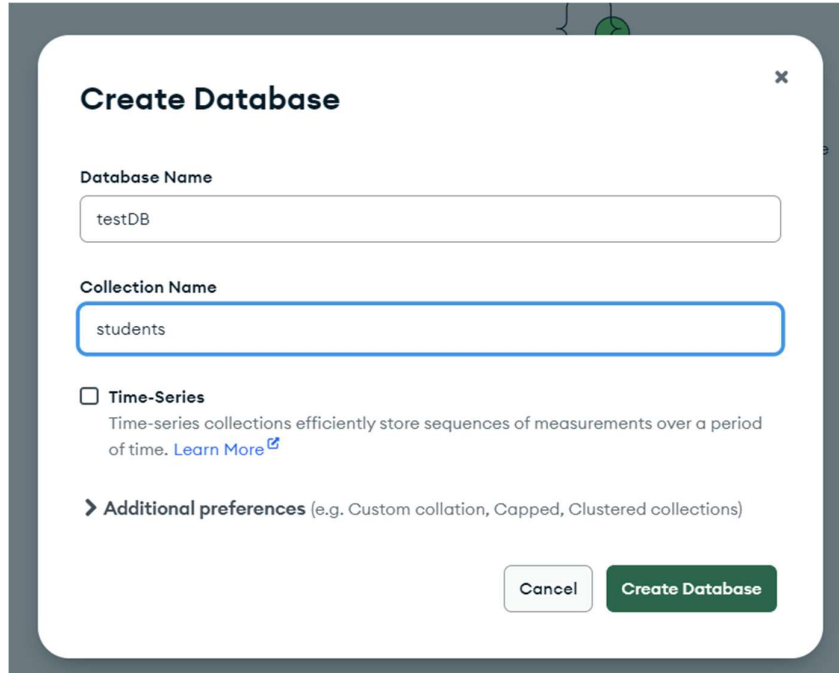


Web Technology Assignment Lab-10

1: Connect to a MongoDB server using MongoDB Compass.



2 and 3: Create a new database named "testdb" in MongoDB Compass.
And Create a new collection named "students" in the "testdb" database.



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

Insert Document

To collection testDB.students

VIEW { } ≡

1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

[

{

"name": "John Doe",

"age": 20,

"email": "john.doe@example.com"

},

{

"name": "Alice Smith",

"age": 22,

"email": "alice.smith@example.com"

},

{

"name": "Bob Johnson",

"age": 21,

"email": "bob.johnson@example.com"

},

{

"name": "Emily Davis",

"age": 23,

"email": "emily.davis@example.com"

},

{

"name": "Michael Wilson",

"age": 19,

"email": "michael.wilson@example.com"

},

{

"name": "Sophia Martinez",

Cancel Insert

5: View the contents of the "students" collection.

testDB.students

101DOCUMENTSINDEXES

DocumentsAggregationsSchemaIndexesValidation

FilterType a query: { field: 'value' } or [generate query](#)

ExplainResetFindOptions

ADD DATAEXPORT DATA

1 - 10 of 10

<pre>{ "_id": ObjectId("660be48c82bc785bb7bbf5cb"), "name": "John Doe", "age": 20, "email": "john.doe@example.com" }</pre>
<pre>{ "_id": ObjectId("660be48c82bc785bb7bbf5cc"), "name": "Alice Smith", "age": 22, "email": "alice.smith@example.com" }</pre>
<pre>{ "_id": ObjectId("660be48c82bc785bb7bbf5cd"), "name": "Bob Johnson", "age": 21, "email": "bob.johnson@example.com" }</pre>
<pre>{ "_id": ObjectId("660be48c82bc785bb7bbf5ce"), "name": "Emily Davis", "age": 23, "email": "emily.davis@example.com" }</pre>
<pre>{ "_id": ObjectId("660be48c82bc785bb7bbf5cf"), "name": "Michael Wilson" }</pre>

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

testDB.students 10 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Indexes Validation

Filter { "name": "John Doe" } Generate query Explain Reset Find Options

ADD DATA EXPORT DATA 1 - 1 of 1

1	_id: ObjectId('660be40c82bc705bb7bbf5cb')	ObjectId
2	name: "John Doe"	String
3	age: 20	Int32
4	email: "john.doe@example.com"	String

CANCEL UPDATE

Filter { "name": "John Doe" } Generate query Explain Reset Find Options

ADD DATA EXPORT DATA 1 - 1 of 1

_id: ObjectId('660be40c82bc705bb7bbf5cb')				
name: "John Doe"				
age: 35				
email: "john.doe@example.com"				

Updated age of john doe from 20 to 35

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

testDB.students 10 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Indexes Validation

Filter { "name": "Bob Johnson" } Generate query Explain Reset Find Options

ADD DATA EXPORT DATA 1 - 1 of 1

_id: ObjectId('660be40c82bc705bb7bbf5cd')
name: "Bob Johnson"
age: 21
email: "bob.johnson@example.com"

Document flagged for deletion. CANCEL DELETE

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

testDB.students

9 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Indexes Validation

Pipeline ▼ Sgroup

average

Generate

SAVE CREATE NEW EXPORT TO LANGUAGE

PREVIEW STAGES TEXT

Stage 1 Sgroup

```
1 {
2   _id: null,
3   average: {
4     $avg: "$age",
5   },
6 }
```

Output after Sgroup stage (Sample of 1 document)

```
{
  _id: null
  average: 22.8888888888889
}
```

+ Add Stage

[Learn more about aggregation pipeline stages](#)

Here average is : 22.88

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

testDB.students

9 DOCUMENTS 1 INDEXES

Documents Aggregations Schema Indexes Validation

Create Index Refresh

VIEWING INDEXES SEARCH INDEXES

Name and Definition	Type	Size	Usage	Properties
> _id_	REGULAR	36.9 KB	8 (since Tue Apr 02 2024)	UNIQUE
> name_text	TEXT	20.5 KB	0 (since Tue Apr 02 2024)	

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

Export

Collection testDB.students

Export File Type

JSON CSV

> Advanced JSON Format

Cancel Export...

Project

{ field: 0 }

Sort

{"age": -1}

MaxTimeMS 60000

Collation

{ locale: 'simple' }

Skip 0

Limit 0

ADD DATA

EXPORT DATA

1 - 9 of 9

name: "John Doe"

age: 35

email: "john.doe@example.com"

_id: ObjectId('660be40c82bc705bb7bbf5d3')

name: "James Brown"

age: 24

email: "james.brown@example.com"

_id: ObjectId('660be40c82bc705bb7bbf5ce')

name: "Emily Davis"

age: 23

email: "emily.davis@example.com"

_id: ObjectId('660be40c82bc705bb7bbf5cc')

name: "Alice Smith"

age: 22

email: "alice.smith@example.com"

Create Index

testDB.students

Index fields

location

2dsphere

+

> Options

Cancel

Create Index

testDB.students

9

DOCUMENTS

1

INDEXES

Documents

Aggregations

Schema

Indexes

Validation

Create Index

Refresh

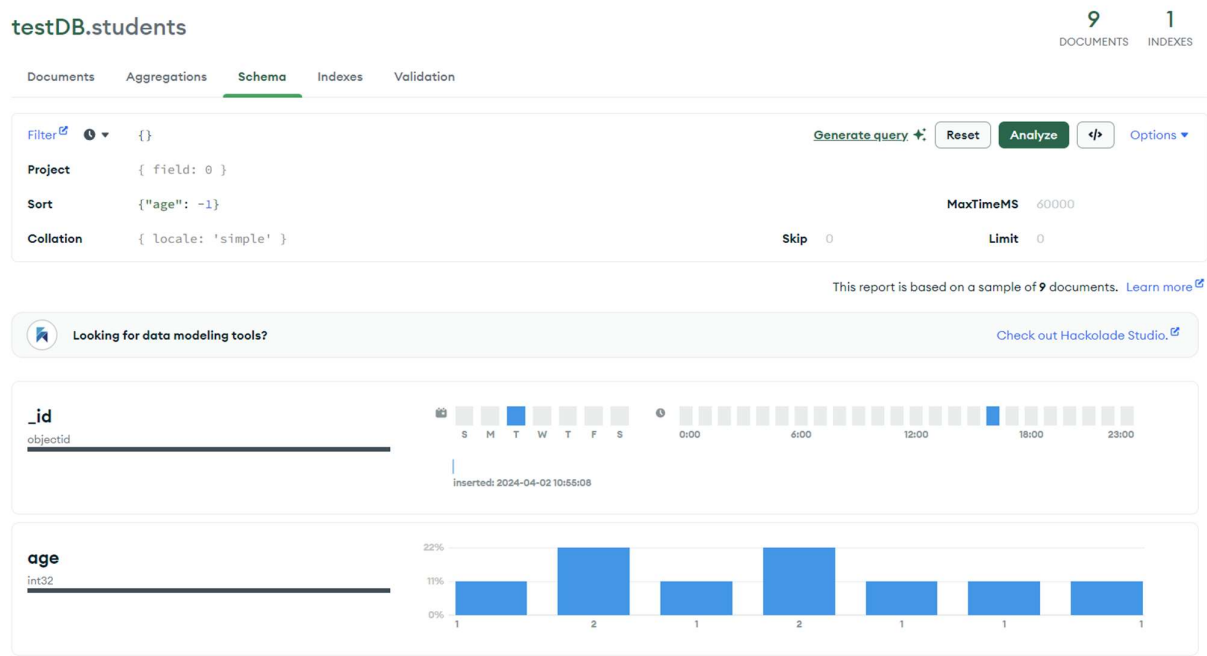
VIEWING

INDEXES

SEARCH INDEXES

Name and Definition	Type	Size	Usage	Properties
> _id_	REGULAR ⓘ	36.9 KB	10 (since Tue Apr 02 2024)	UNIQUE ⓘ
> location_2dsphere	GEOSPATIAL ⓘ	8.2 KB	0 (since Tue Apr 02 2024)	
> name_text	TEXT ⓘ	20.5 KB	0 (since Tue Apr 02 2024)	

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



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

Validation Action Error Validation Level Strict

```
1 {
2   required: [
3     'name'
4   ],
5   properties: {
6     name: {
7       bsonType: 'string',
8       description: 'Name must be a present.'
9     }
10  }
11 }
```

Sample documents

Passed validation

No Preview Documents

Failed validation

```
_id: ObjectId('660be46c82bc705bb7bbf5d0')
name: "Sophia Martinez"
age: 20
email: "sophia.martinez@example.com"
```