

Creating a Database Using MongoDB and Mongosh

Name: Poluri Jyothi

Email: polurijyothi1@gmail.com

Phone No.: 9494088812

Roll No.: 20NN1A12B2

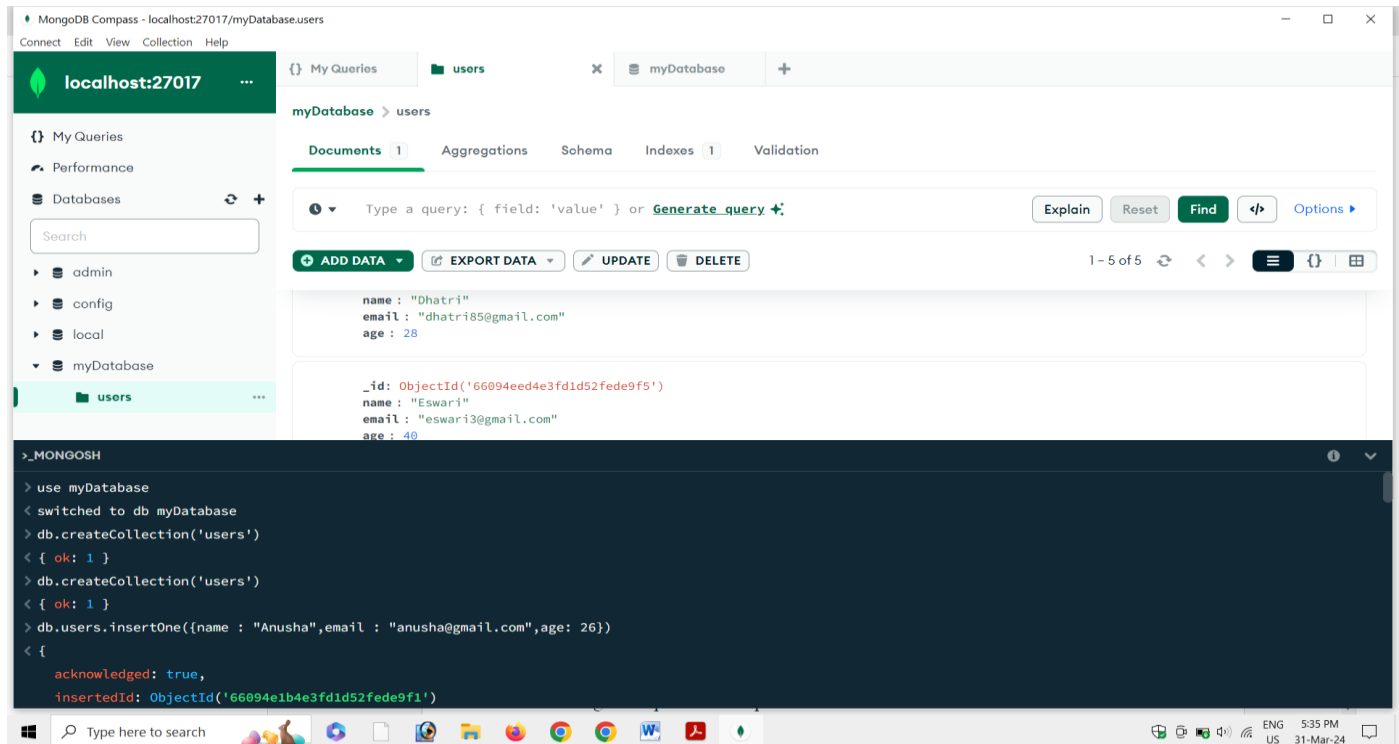
**College Name: Vignan's Nirula Institute of Technology and
Science For Women.**

Creating Database:

“Use myDatabase” command is used to create and use the database called myDatabase.

“db.createCollection(“users”)” command is used to create a collection called users in myDatabase database.

Output:

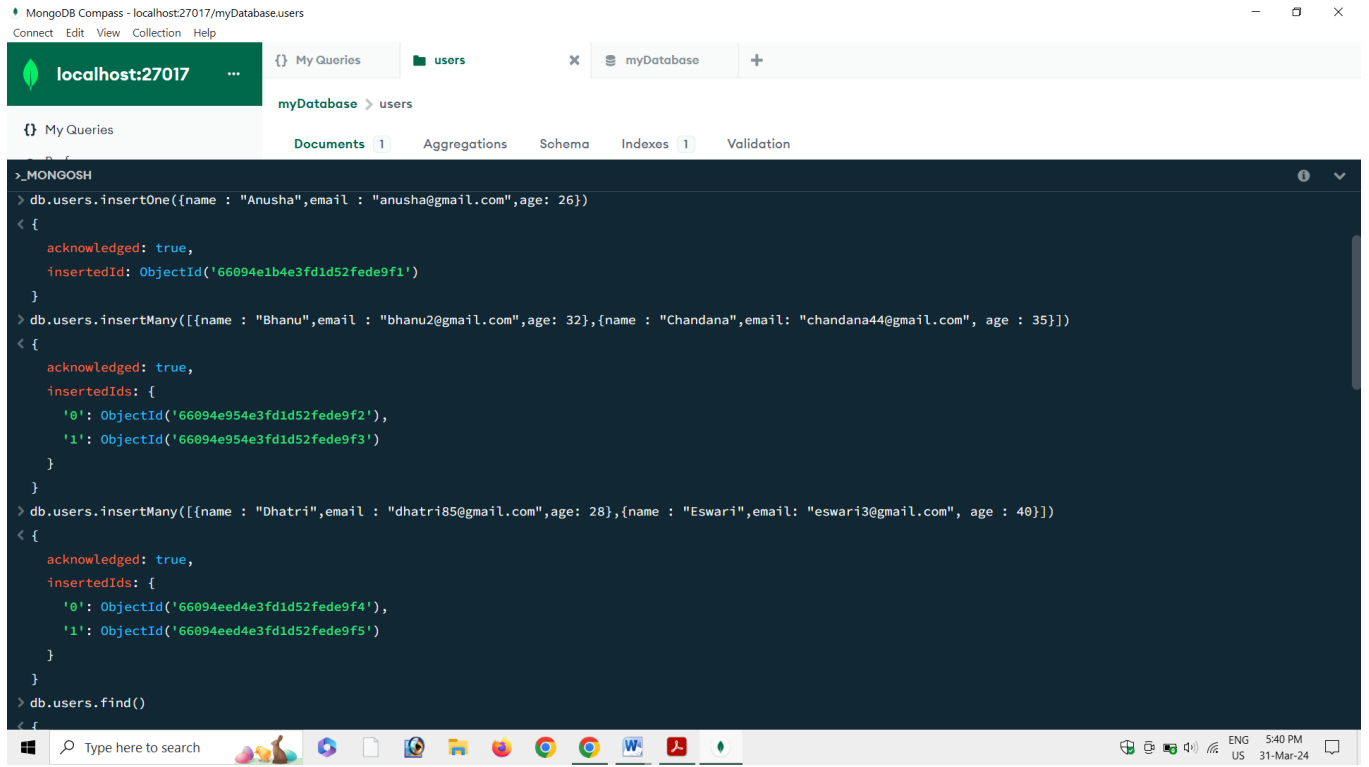


Inserting documents in to users collection:

“db.users.insertOne({ })”- This command is used to insert a single document in to the collection.

“db.users.insertMany([{ }])” – This command is used to insert multiple documents at a time in to the collection.

Output:



MongoDB Compass - localhost:27017/myDatabase.users

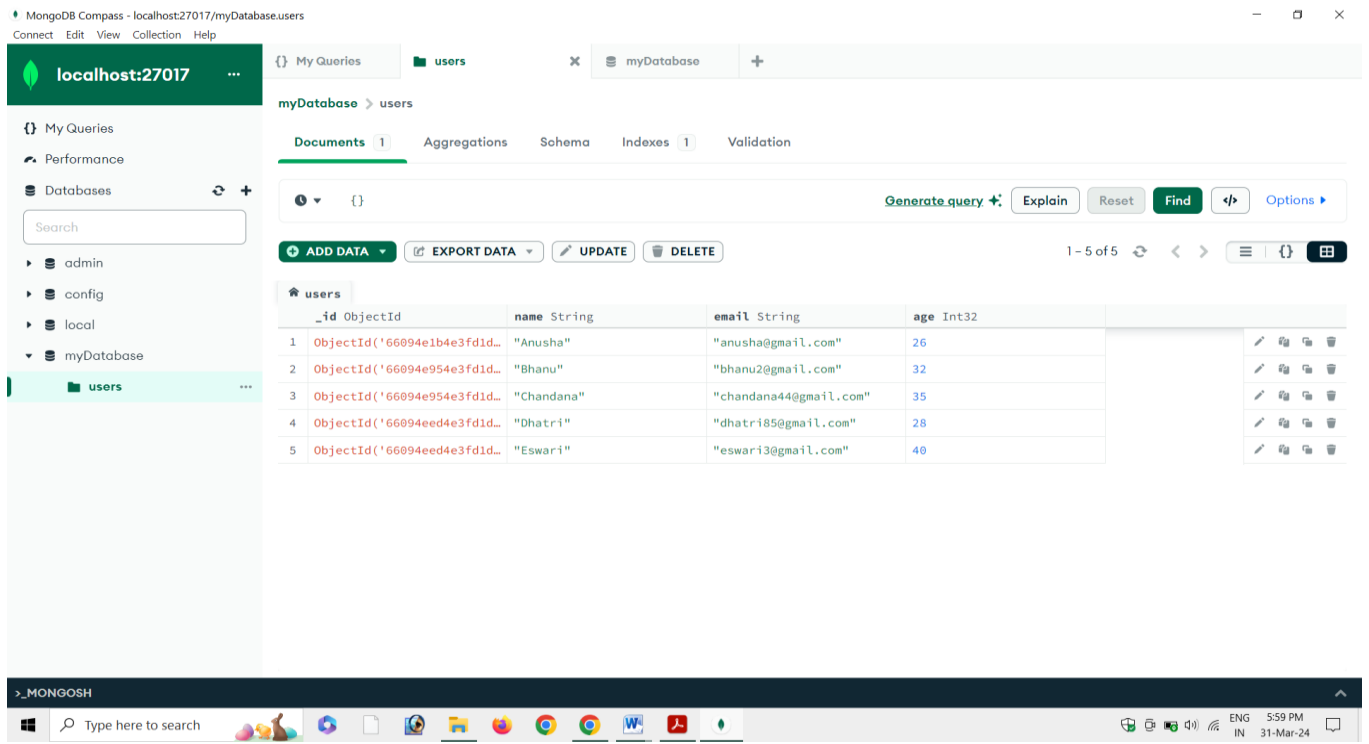
Connect Edit View Collection Help

localhost:27017 ... {} My Queries users x myDatabase +

myDatabase > users

Documents 1 Aggregations Schema Indexes 1 Validation

```
>_MONGOSH
> db.users.insertOne({name : "Anusha",email : "anusha@gmail.com",age: 26})
< {
  acknowledged: true,
  insertedId: ObjectId('66094e1b4e3fd1d52fede9f1')
}
> db.users.insertMany([{name : "Bhanu",email : "bhanu2@gmail.com",age: 32},{name : "Chandana",email: "chandana44@gmail.com", age : 35}])
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('66094e954e3fd1d52fede9f2'),
    '1': ObjectId('66094e954e3fd1d52fede9f3')
  }
}
> db.users.insertMany([{name : "Dhatri",email : "dhatri85@gmail.com",age: 28},{name : "Eswari",email: "eswari3@gmail.com", age : 40}])
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('66094eed4e3fd1d52fede9f4'),
    '1': ObjectId('66094eed4e3fd1d52fede9f5')
  }
}
> db.users.find()
< {
```



MongoDB Compass - localhost:27017/myDatabase.users

Connect Edit View Collection Help

localhost:27017 ... {} My Queries users x myDatabase +

myDatabase > users

Documents 1 Aggregations Schema Indexes 1 Validation

Generate query + Explain Reset Find </> Options

ADD DATA EXPORT DATA UPDATE DELETE 1 - 5 of 5

#	_id	name	email	age	
1	ObjectId('66094e1b4e3fd1d52fede9f1')	Anusha	anusha@gmail.com	26	
2	ObjectId('66094e954e3fd1d52fede9f2')	Bhanu	bhanu2@gmail.com	32	
3	ObjectId('66094e954e3fd1d52fede9f3')	Chandana	chandana44@gmail.com	35	
4	ObjectId('66094eed4e3fd1d52fede9f4')	Dhatri	dhatri85@gmail.com	28	
5	ObjectId('66094eed4e3fd1d52fede9f5')	Eswari	eswari3@gmail.com	40	

>_MONGOSH

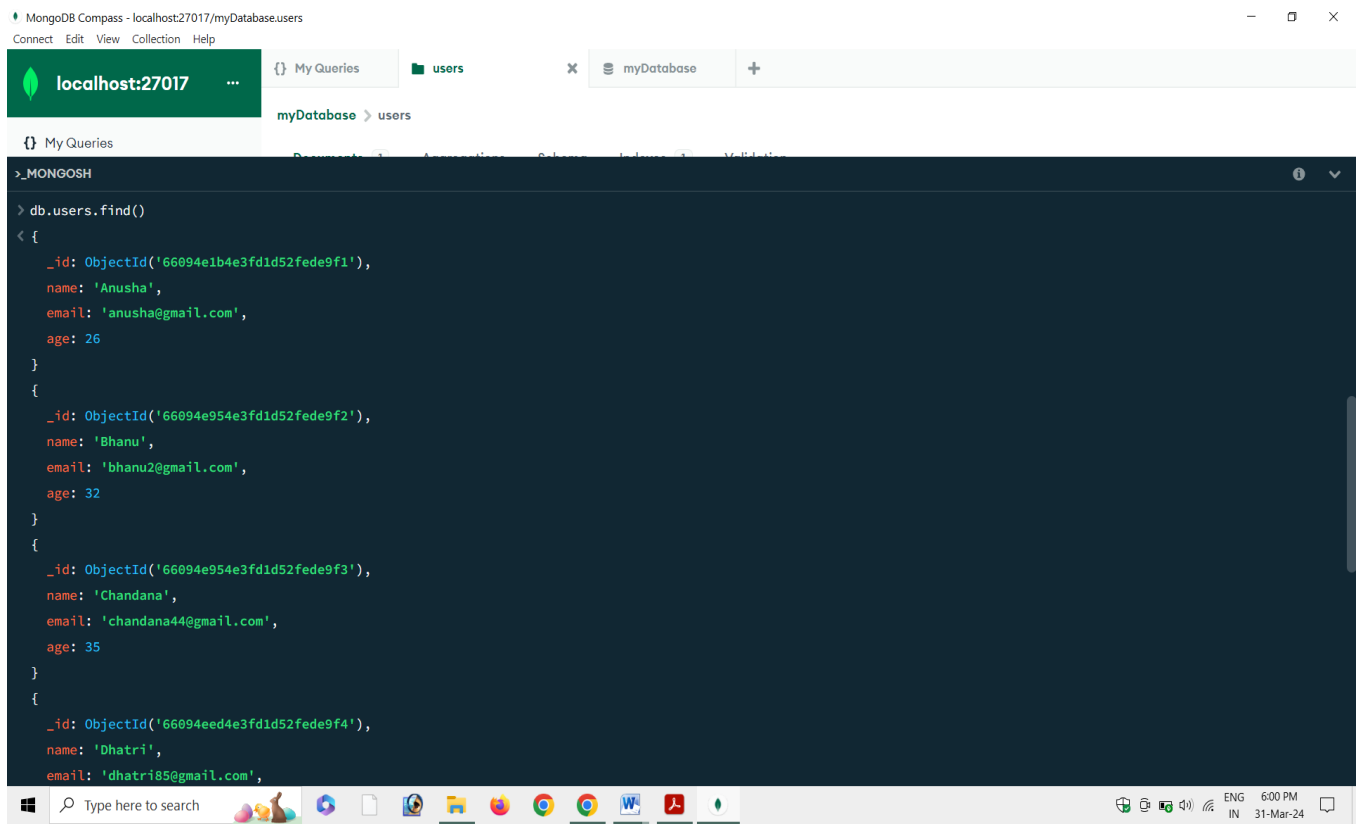
Retrieving the documents in collections:

“db.users.find()” – This command is used to retrieve all the documents in the users collections.

“db.users.find({ field: { \$gt: value}})” – This command is used to retrieve the specific field with specific value.

“db.users.find({ age : { \$gt: 30 }})” – This command is used to retrieve the documents of the persons whose age is greater than 30.

Output:



The screenshot shows the MongoDB Compass interface. The top bar indicates the connection to 'localhost:27017' and the database 'myDatabase'. The 'users' collection is selected. The 'My Queries' tab is active, showing a query: `> db.users.find()`. The output is displayed in a dark-themed console, showing four documents as JSON objects. Each document contains an '_id' (ObjectId), 'name', 'email', and 'age'.

```
> db.users.find()
< {
  _id: ObjectId('66094e1b4e3fd1d52fede9f1'),
  name: 'Anusha',
  email: 'anusha@gmail.com',
  age: 26
}
{
  _id: ObjectId('66094e954e3fd1d52fede9f2'),
  name: 'Bhanu',
  email: 'bhanu2@gmail.com',
  age: 32
}
{
  _id: ObjectId('66094e954e3fd1d52fede9f3'),
  name: 'Chandana',
  email: 'chandana44@gmail.com',
  age: 35
}
{
  _id: ObjectId('66094eed4e3fd1d52fede9f4'),
  name: 'Dhatri',
  email: 'dhatri85@gmail.com',
  age: 30
}
```

MongoDB Compass - localhost:27017/myDatabase.users

Connect Edit View Collection Help

localhost:27017

My Queries users myDatabase +

myDatabase > users

```
>_MONGOSH
{
  "_id": ObjectId("66094eed4e3fd1d52fede9f5"),
  name: 'Dhatrî',
  email: 'dhatrî85@gmail.com',
  age: 28
}
{
  "_id": ObjectId("66094eed4e3fd1d52fede9f5"),
  name: 'Eswari',
  email: 'eswari3@gmail.com',
  age: 40
}
}
> db.users.find({ age : { $gt: 30}})
< {
  "_id": ObjectId("66094e954e3fd1d52fede9f2"),
  name: 'Bhanu',
  email: 'bhanu2@gmail.com',
  age: 32
}
{
  "_id": ObjectId("66094e954e3fd1d52fede9f3"),
  name: 'Chandana',
  email: 'chandana44@gmail.com',
  age: 35
}
}
```

Retrieving the documents whose age is greater than 30:

MongoDB Compass - localhost:27017/myDatabase.users

Connect Edit View Collection Help

localhost:27017

My Queries users myDatabase +

myDatabase > users

Documents 1 Aggregations Schema Indexes 1 Validation

Generate query + Explain Reset Find Options

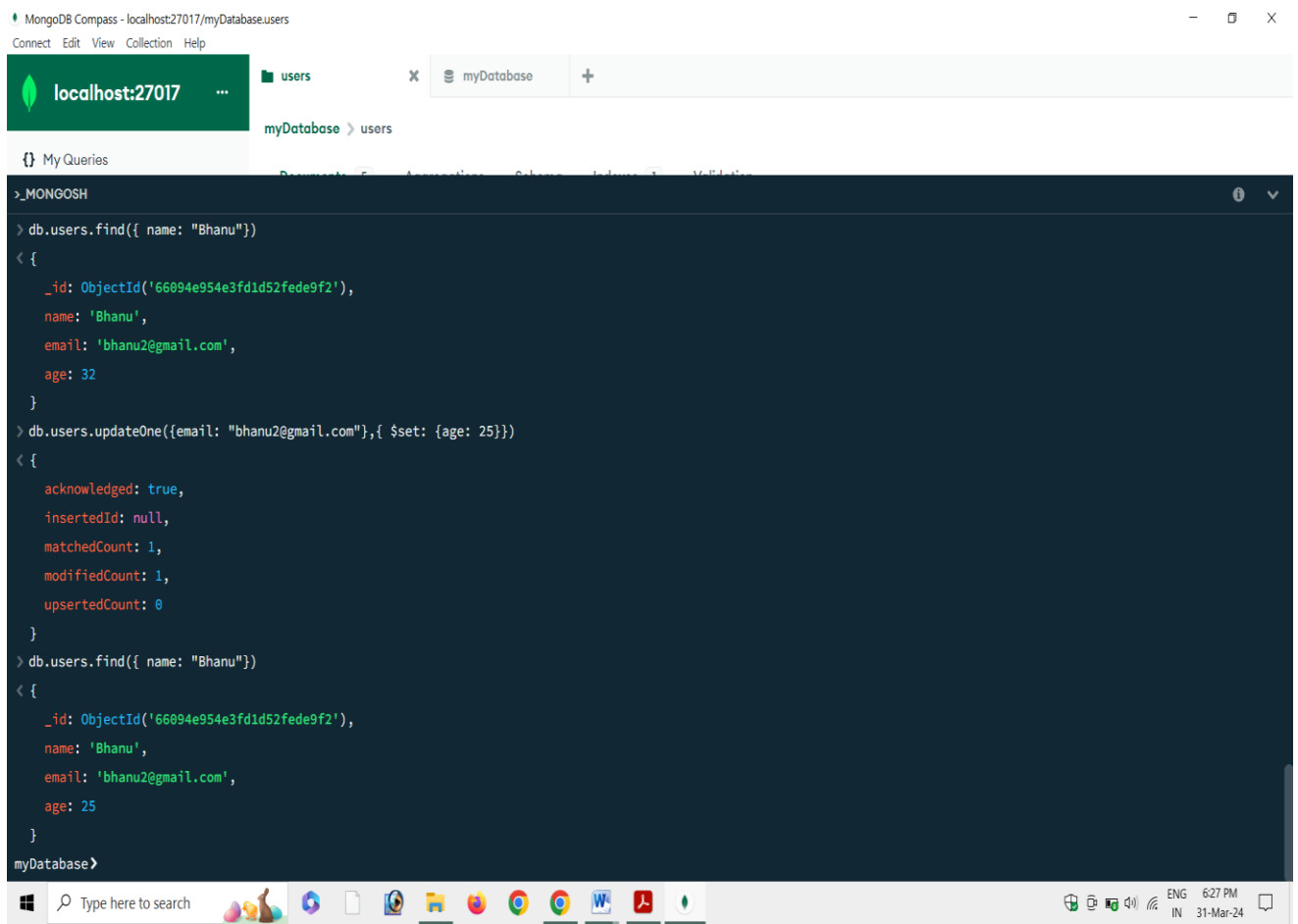
```
>_MONGOSH
> db.users.find({ age : { $gt: 30}})
< {
  "_id": ObjectId("66094e954e3fd1d52fede9f2"),
  name: 'Bhanu',
  email: 'bhanu2@gmail.com',
  age: 32
}
{
  "_id": ObjectId("66094e954e3fd1d52fede9f3"),
  name: 'Chandana',
  email: 'chandana44@gmail.com',
  age: 35
}
}
{
  "_id": ObjectId("66094eed4e3fd1d52fede9f5"),
  name: 'Eswari',
  email: 'eswari3@gmail.com',
  age: 40
}
}
myDatabase>
```

Update Operation:

“db.users.updateOne({field: value}, { \$set: {field: value}})” – This command is used to update the document of a specific user.

“db.users.updateOne({email: bhanu2@gmail.com},{ \$set: {age: 25}})”

Output:



The screenshot shows the MongoDB Compass interface. The top bar indicates the connection to 'localhost:27017/myDatabase.users'. The left sidebar shows the 'users' collection. The main area displays the MONGODB shell output for the update operation. The output shows the document being updated and the result of the operation, including the 'acknowledged' status and counts.

```
> MONGODB
> db.users.find({ name: "Bhanu"})
< {
  _id: ObjectId('66094e954e3fd1d52fede9f2'),
  name: 'Bhanu',
  email: 'bhanu2@gmail.com',
  age: 32
}
> db.users.updateOne({email: "bhanu2@gmail.com"},{ $set: {age: 25}})
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
> db.users.find({ name: "Bhanu"})
< {
  _id: ObjectId('66094e954e3fd1d52fede9f2'),
  name: 'Bhanu',
  email: 'bhanu2@gmail.com',
  age: 25
}
myDatabase>
```

Delete Operation:

“db.users.deleteOne()” –This command is used delete a document from the collection.

“db.users.deleteOne({email: chandana44@gmail.com})”

Output:

MongoDB Compass - localhost:27017/myDatabase.users

Connect Edit View Collection Help

localhost:27017

users myDatabase

myDatabase > users

Documents 5 Aggregations Schema Indexes 1 Validation

Generate query Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE

1 - 5 of 5

```
{
  "_id": ObjectId('66094e954e3fd1d52fede9f2'),
  "name": "Bhanu",
  "email": "bhanu2@gmail.com",
  "age": 25
}
```

> MONGOSH

```
> db.users.find({email: "chandana44@gmail.com"})
< {
  "_id": ObjectId('66095f824e3fd1d52fede9f6'),
  "name": 'Chandana',
  "email": 'chandana44@gmail.com',
  "age": 35
}
> db.users.deleteOne({email: "chandana44@gmail.com"})
< {
  "acknowledged": true,
  "deletedCount": 1
}
> db.users.find({email: "chandana44@gmail.com"})
<
myDatabase>
```

MongoDB Compass - localhost:27017/myDatabase.users

Connect Edit View Collection Help

localhost:27017

users myDatabase

myDatabase > users

Documents 5 Aggregations Schema Indexes 1 Validation

Generate query Explain Reset Find Options

ADD DATA EXPORT DATA UPDATE DELETE

1 - 4 of 4

	_id ObjectId	name String	email String	age Int32	
1	ObjectId('66094e1b4e3fd1d...	"Anusha"	"anusha@gmail.com"	26	
2	ObjectId('66094e954e3fd1d...	"Bhanu"	"bhanu2@gmail.com"	25	
3	ObjectId('66094eed4e3fd1d...	"Dhatri"	"dhatri85@gmail.com"	28	
4	ObjectId('66094eed4e3fd1d...	"Eswari"	"eswar13@gmail.com"	40	

> MONGOSH

Index Creation:

“db.users.createIndex({email:”text”})” –This command is used to create index for email field.

Output:

The screenshot shows the MongoDB Compass interface for a local instance at localhost:27017. The left sidebar displays the database structure with 'myDatabase' expanded to show the 'users' collection. The main panel is set to the 'Indexes' tab for the 'users' collection. A table lists the existing index: '_id_' with a 'REGULAR' type, a size of 36.9 KB, and usage of 20 documents since Sun Mar 31 2024. Below the table, a terminal window shows the execution of the command `db.users.createIndex({email:"text"})` and the output of `db.users.getIndexes()`, which lists the new 'email_text' index with a 'TEXT' type, a size of 20.5 KB, and 0 documents indexed.

Name and Definition	Type	Size	Usage	Properties
> _id_	REGULAR	36.9 KB	20 (since Sun Mar 31 2024)	UNIQUE

```
> MONGODB
> db.users.createIndex({email:"text"})
< email_text
> db.users.getIndexes()
< [
  { v: 2, key: { '_id': 1 }, name: '_id_' },
  {
    v: 2,
    key: { '_fts': 'text', '_ftsx': 1 },
    name: 'email_text',
    weights: { email: 1 },
    default_language: 'english',
    language_override: 'language',
    textIndexVersion: 3
  }
]
```

This screenshot shows the MongoDB Compass interface after the index creation. The 'Indexes' tab for the 'users' collection now displays two indexes: '_id_' (REGULAR) and 'email_text' (TEXT). The 'email_text' index has a size of 20.5 KB and 0 documents indexed since Sun Mar 31 2024.

Name and Definition	Type	Size	Usage	Properties
> _id_	REGULAR	36.9 KB	20 (since Sun Mar 31 2024)	UNIQUE
> email_text	TEXT	20.5 KB	0 (since Sun Mar 31 2024)	