

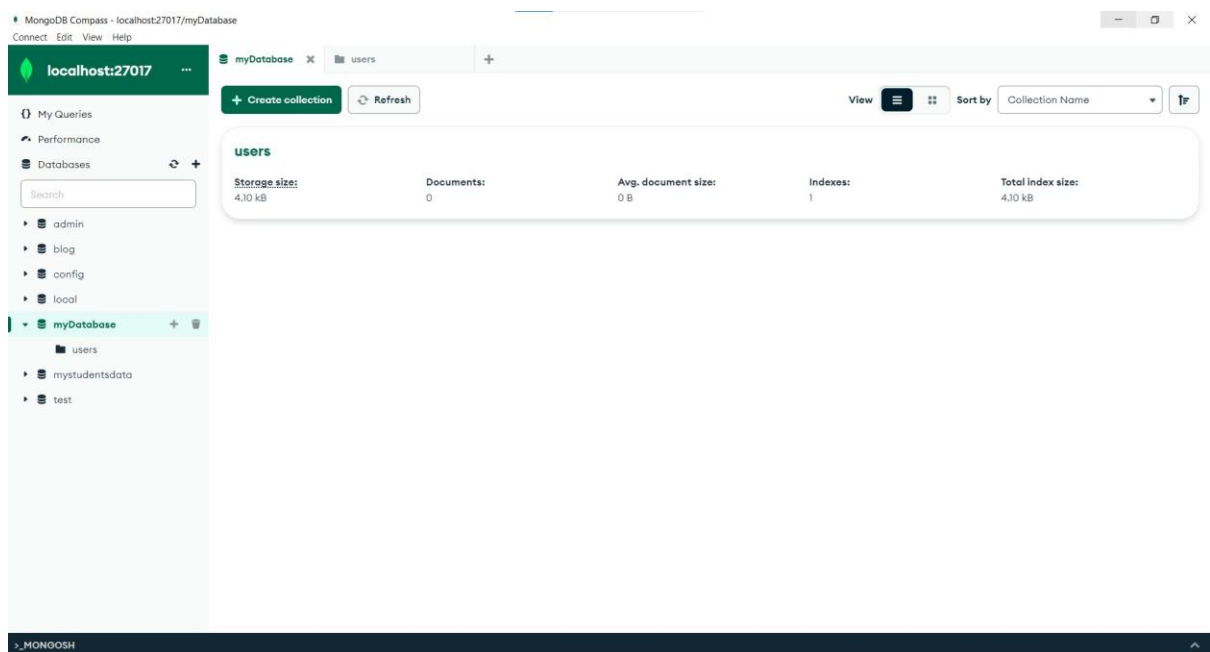
Assignment-4: Creating a Database Using MongoDB and Mongosh

1. Database Setup:

- Open the mongoDB compass
- Create a new MongoDB database: myDatabase

2. Collection Creation:

- Create a collection within database: users

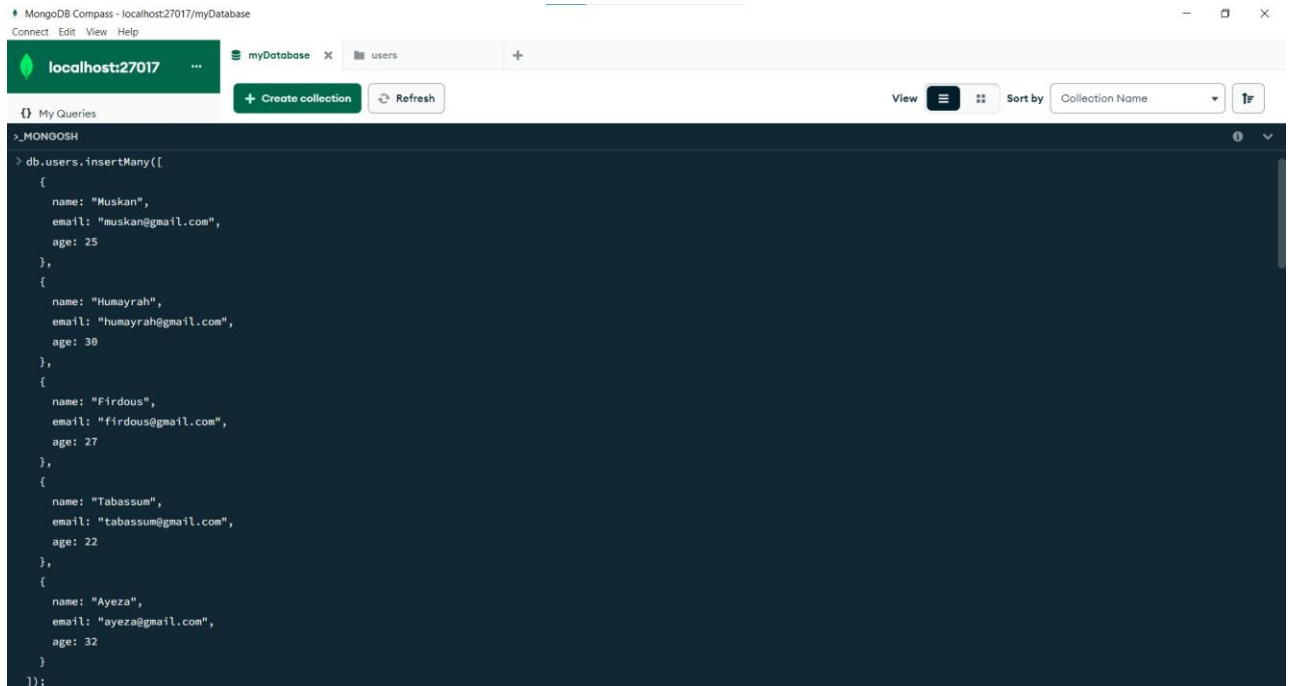


3. Document Insertion:

- Insert five documents into the users collection, each representing a user with fields such as name, email, and age.
- Before inserting into collection “use db” command to switch the current database context within MongoDB.

```
>_MONGOSH
> use myDatabase
< switched to db myDatabase
```

- `db.users.insertMany()` method is used to insert the documents into users collection as shown below.

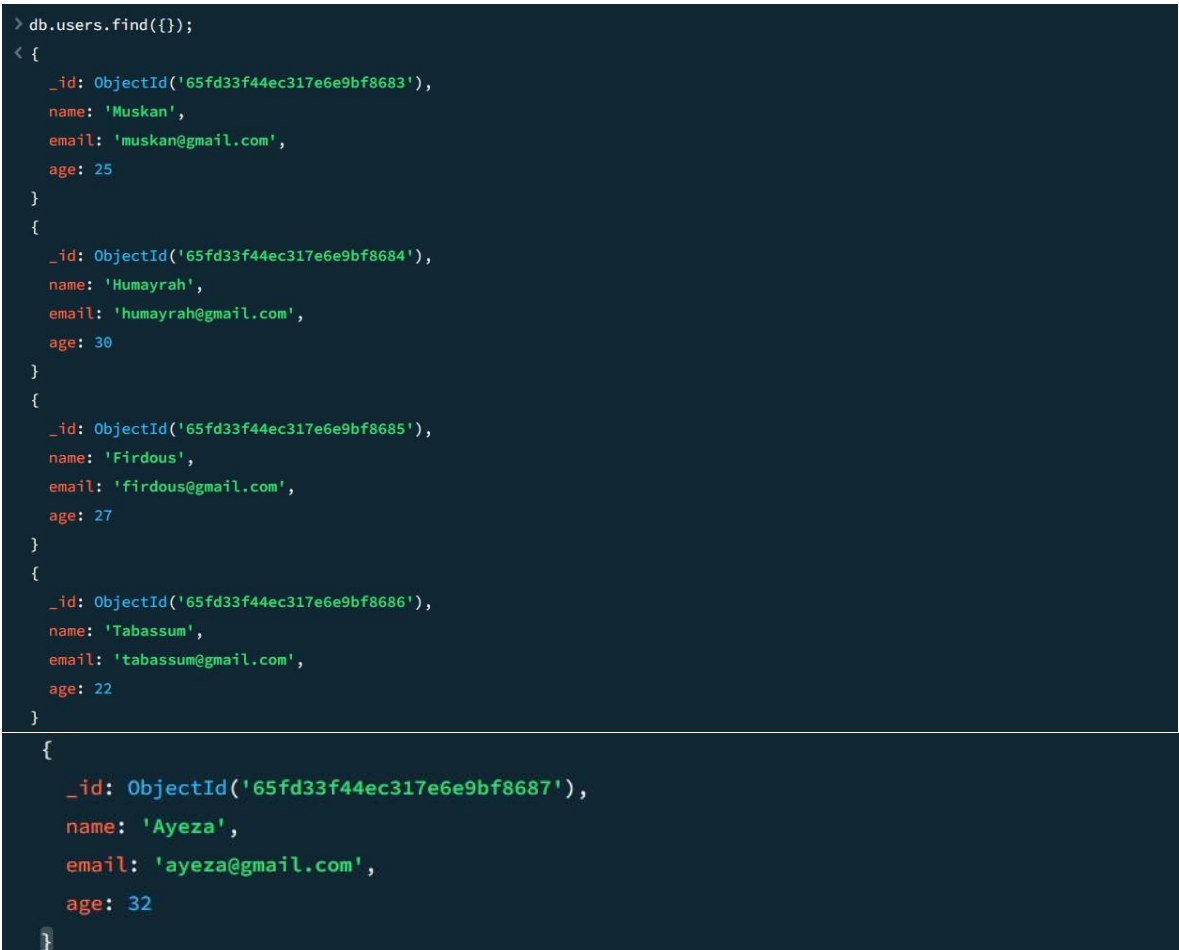


The screenshot shows the MongoDB Compass application. The top bar indicates the connection to 'localhost:27017/myDatabase'. The left sidebar shows the 'users' collection. The main area displays the MongoDB Shell with the following command and result:

```
> db.users.insertMany([
  {
    name: "Muskan",
    email: "muskan@gmail.com",
    age: 25
  },
  {
    name: "Humayrah",
    email: "humayrah@gmail.com",
    age: 30
  },
  {
    name: "Firdous",
    email: "firdous@gmail.com",
    age: 27
  },
  {
    name: "Tabassum",
    email: "tabassum@gmail.com",
    age: 22
  },
  {
    name: "Ayeza",
    email: "ayeza@gmail.com",
    age: 32
  }
]);
```

4. Queries to retrieve:

- To retrieve all the users from the users collection.
 - `db.users.find({ });`



The screenshot shows the MongoDB Shell output for the query `db.users.find({});`. The result is a JSON array of five user documents:

```
> db.users.find({});
< {
  _id: ObjectId('65fd33f44ec317e6e9bf8683'),
  name: 'Muskan',
  email: 'muskan@gmail.com',
  age: 25
}
{
  _id: ObjectId('65fd33f44ec317e6e9bf8684'),
  name: 'Humayrah',
  email: 'humayrah@gmail.com',
  age: 30
}
{
  _id: ObjectId('65fd33f44ec317e6e9bf8685'),
  name: 'Firdous',
  email: 'firdous@gmail.com',
  age: 27
}
{
  _id: ObjectId('65fd33f44ec317e6e9bf8686'),
  name: 'Tabassum',
  email: 'tabassum@gmail.com',
  age: 22
}
{
  _id: ObjectId('65fd33f44ec317e6e9bf8687'),
  name: 'Ayeza',
  email: 'ayeza@gmail.com',
  age: 32
}
```

- To retrieve the specific users with an age greater than or equal to 30
 - `db.users.find({ age: { $gte: 30 } });`

```
> db.users.find({ age: { $gte: 30 } });
< {
  _id: ObjectId('65fd33f44ec317e6e9bf8684'),
  name: 'Humayrah',
  email: 'humayrah@gmail.com',
  age: 30
}
{
  _id: ObjectId('65fd33f44ec317e6e9bf8687'),
  name: 'Ayeza',
  email: 'ayeza@gmail.com',
  age: 32
}
```

5. Update Operation:

- To update the age of a user with a specific email address in MongoDB, use the `updateOne()` method.
- Example:
 - `Db.users.updateOne(
 { email: "muskan@gmail.com" },
 { $set: { age: 20 } }
);`

```
> db.users.updateOne(
  { email: "muskan@gmail.com" },
  { $set: { age: 20 } }
);
< {
  acknowledged: true,
  insertedId: null,
  matchedCount: 1,
  modifiedCount: 1,
  upsertedCount: 0
}
```

6. Deletion Operation:

- To delete the user document based on a specific email address in MongoDB, you can use the `deleteOne()` method.
 - `db.users.deleteOne({email: firdous@gmail.com});`

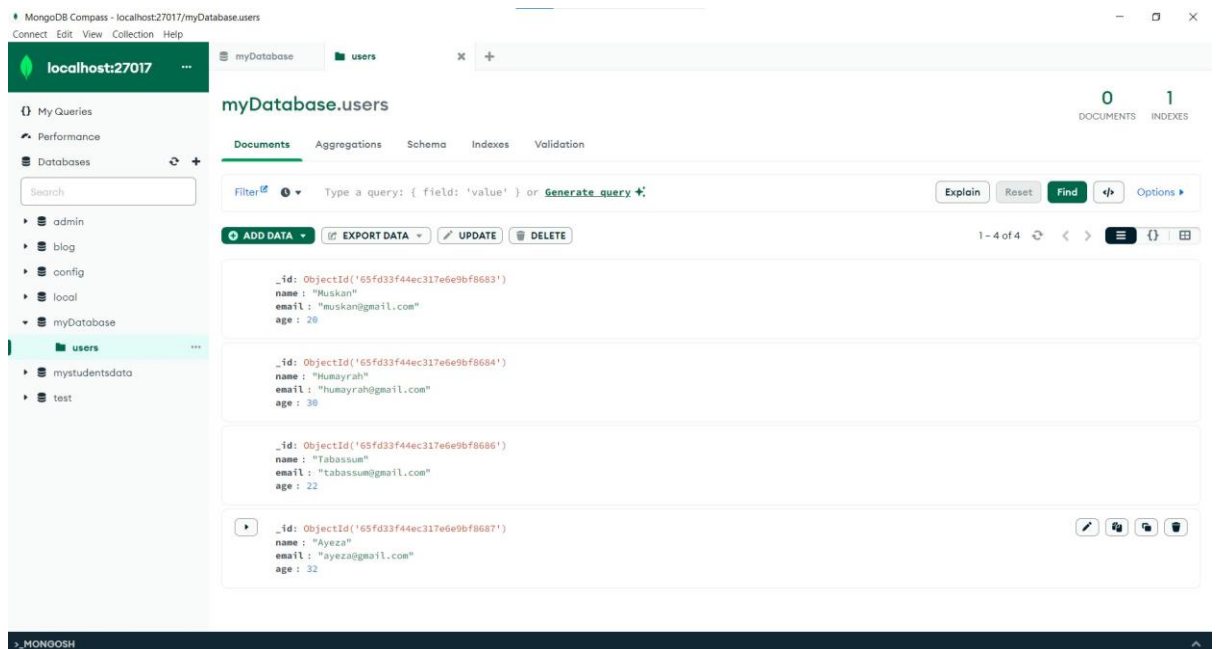
```
> db.users.deleteOne({ email: "firdous@gmail.com" });
< {
  acknowledged: true,
  deletedCount: 1
}
```

7. Index Creation:

- To create an index on the email field of the users collection in MongoDB, use the `db.users.createIndex()` method.
 - `db.users.createIndex({ email: 1 });`

```
> db.users.createIndex({ email: 1 });
< email_1
> db.users.getIndexes();
< [
  { v: 2, key: { _id: 1 }, name: '_id_' },
  { v: 2, key: { email: 1 }, name: 'email_1' }
]
myDatabase>
```

Final Outcome:



MongoDB Compass - localhost:27017/myDatabase.users

Connect Edit View Collection Help

localhost:27017

myDatabase users

My Queries

Performance

Databases

admin

blog

config

local

myDatabase

users

mystudentsdata

test

myDatabase.users

0 DOCUMENTS1 INDEXES

DocumentsAggregationsSchemaIndexesValidation

Create IndexRefresh

VIEWINGINDEXESSearch Indexes

Name and Definition	Type	Size	Usage	Properties
<div><div>id</div><div>id</div></div>	REGULAR	36.9 KB	4 (since Fri Mar 22 2024)	UNIQUE
<div><div>email_1</div><div>email</div></div>	REGULAR	20.5 KB	0 (since Fri Mar 22 2024)	

MONGODB