

Assignment - 4

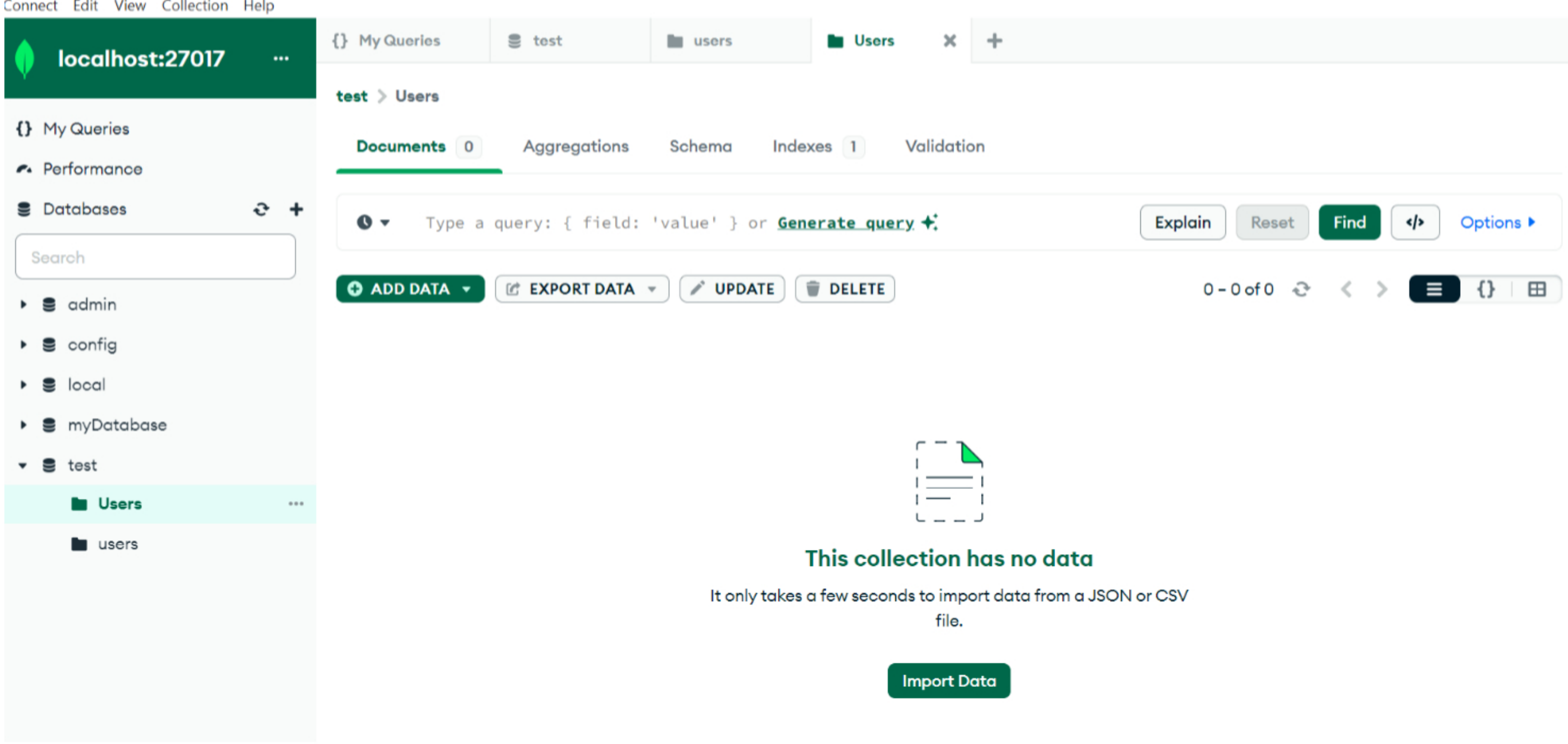
Creating a Database Using MongoDB and Mongosh

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

```
>_MONGOSH

> db.Users.insertMany([
  {Name:"Bakshu",
   Age:21,
   Email:"Bakshu@gmail.com"
 },
  {Name:"javid",
   Age:22,
   Email:"javid@gmail.com"
 },
  {
   Name:"Asif",
   Age:21,
   Email:"asif@gmail.com"
 }
])
< {
  acknowledged: true,
  insertedIds: {
    '0': ObjectId('65fdae1c74ffc01a2acc2f82'),
    '1': ObjectId('65fdae1c74ffc01a2acc2f83'),
    '2': ObjectId('65fdae1c74ffc01a2acc2f84')
  }
}
```

4. Queries to retrieve:

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

```
>_MONGOSH

> db.users.find({})
< {
  _id: ObjectId('65fda62df4d90c6c798df3ea'),
  Name: 'Bakshu',
  Age: 24,
  Email: 'Bakshu@gmail.com'
}
{
  _id: ObjectId('65fda62df4d90c6c798df3eb'),
  Name: 'Javid',
  Age: 22,
  Email: 'javid@gmail.com'
}
{
  _id: ObjectId('65fda62df4d90c6c798df3ec'),
  Name: 'Asif',
  Age: 21,
  Email: 'Asifa@gmail.com'
}
```

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

```
>_MONGOSH

> db.users.find({Age: { $gt: 21}});
< {
  _id: ObjectId('65fda62df4d90c6c798df3ea'),
  Name: 'Bakshu',
  Age: 22,
  Email: 'Bakshu@gmail.com'
}
{
  _id: ObjectId('65fda62df4d90c6c798df3eb'),
  Name: 'Javid',
  Age: 22,
  Email: 'javid@gmail.com'
}
{
  _id: ObjectId('65fda62df4d90c6c798df3ec'),
  Name: 'Asif',
  Age: 21,
  Email: 'Asifa@gmail.com'
}
```

5. Update Operation:

- To update the age of a user with a specific email address in MongoDB, use the updateOne() method.
- Example: o Db.users.updateOne({ Name: "Bakshu" }, { \$set: { Age: 24 } });

```
>_MONGOSH

> db.users.updateOne({Name:"Bakshu"},{$set: {Age:24}});
< {
  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:"asif@gmail.com"})
< {
  acknowledged: true,
  deletedCount: 1
}
```

db.users.deleteOne({Email: asif@gmail.com});

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 });

```
>_MONGOSH

> 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' }
]
test>
```

Final Outcome:

