Creating a Database Using MongoDB and Mongosh

NAME : KSHATRI AKHILESWARI BAI

EMAIL ID : 208X1a0537@khitguntur.ac.in

PHONE NO : 8712363393
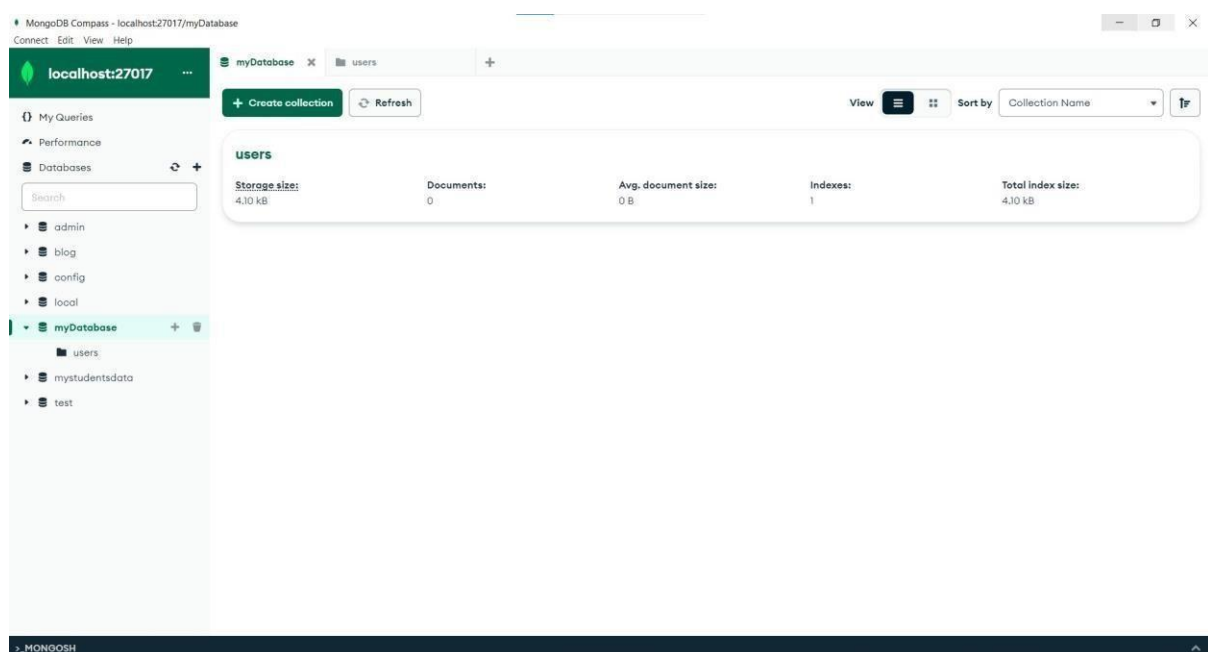
ROLL NO : 208X1A0537 (CSE)

COLLEGE : KALLAM HARANADHAREDDY INSTITUTE OF TECHNOLOGY, GUNTUR

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.

4. Queries to retrieve:

```
>_MONGOSH                                                                              0  v

myDatabase > db.users.insertMany([
            {
              name:"sure",
              email:"sure24@gmail.com",
              age:21
            },
            {
               name:"divya",
               email:"divya24@gmail.com",
               age:21
            },
            {
               name:"kimtaeyung",
               email:"btsv@gmail.com",
               age:24
            },
            {
               name:"yangyang",
               email:"yang@gmail.com",
               age:25
            },
            {
               name:"bie",
               email:"bie@gmail.com",
               age:24
            }
          ]);
```

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

```
> _MONGOSH                                                                    ℹ ∨
> db.users.find({});
< {
    _id: ObjectId('65fdcaa41e25bd20f7b29ab5'),
    name: 'sure',
    email: 'sure24@gmail.com',
    age: 21
  }
  {
    _id: ObjectId('65fdcaa41e25bd20f7b29ab6'),
    name: 'divya',
    email: 'divya24@gmail.com',
    age: 21
  }
  {
    _id: ObjectId('65fdcaa41e25bd20f7b29ab7'),
    name: 'kimtaeyung',
    email: 'btsv@gmail.com',
    age: 24
  }
  {
    _id: ObjectId('65fdcaa41e25bd20f7b29ab8'),
    name: 'yangyang',
    email: 'yang@gmail.com',
    age: 25
  }
  {
    _id: ObjectId('65fdcaa41e25bd20f7b29ab9'),
    name: 'bie',
    email: 'bie@gmail.com',
    age: 24
  }
```

- To retrieve the specific users with an age greater than or equal to 22
  - db.users.find({ age: { $gte: 22 } });

```
> db.users.find({age:{$gte:22}});
< {
    _id: ObjectId('65fdcaa41e25bd20f7b29ab7'),
    name: 'kimtaeyung',
    email: 'btsv@gmail.com',
    age: 24
  }
  {
    _id: ObjectId('65fdcaa41e25bd20f7b29ab8'),
    name: 'yangyang',
    email: 'yang@gmail.com',
    age: 25
  }
  {
    _id: ObjectId('65fdcaa41e25bd20f7b29ab9'),
    name: 'bie',
    email: 'bie@gmail.com',
    age: 24
  }
```

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(

              { email: "sure24@gmail.com" },

              { $set: { age: 22 } }

        );

```
> db.users.updateOne(
  { email:"sure24@gmail.com"},
  { $set: { age: 22}}
);
< {
  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.

      o db.users.deleteOne({email:yangyang@gmail.com});

```
> db.users.deleteOne({ email:"yangyang@gmail.com"});
< {
   acknowledged: true,
   deletedCount: 0
}
```

7. Index Creation:

    To create an index on the email field of the users collection in MongoDB, use the db.users.createIndex() method. o 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' }
]
```

Final Outcome:

localhost:27017   ...

{} My Queries   **users**   ✕   myDatabase   +

myDatabase > users

{} My Queries
Performance
Databases   ⟳ +
Search

Documents 0   Aggregations   Schema   Indexes 1   Validation

🕐 ▾   Type a query: { field: 'value' } or **Generate query** +:   Explain   Reset   **Find**   </>   Options ▸

● ADD DATA ▾   ☐ EXPORT DATA ▾   ✎ UPDATE   🗑 DELETE   1 - 5 of 5   ⟳   < >   ▤   {}   ⊞

▸ admin
▸ config
▸ local
▾ myDatabase
   users   ...

```
_id: ObjectId('65fdcaa41e25bd20f7b29ab5')
name : "sure"
email : "sure24@gmail.com"
age : 22

_id: ObjectId('65fdcaa41e25bd20f7b29ab6')
name : "divya"
email : "divya24@gmail.com"
age : 21

_id: ObjectId('65fdcaa41e25bd20f7b29ab7')
name : "kimtaeyung"
email : "btsv@gmail.com"
age : 24

_id: ObjectId('65fdcaa41e25bd20f7b29ab8')
name : "yangyang"
email : "yang@gmail.com"
age : 25

_id: ObjectId('65fdcaa41e25bd20f7b29ab9')
name : "bie"
email : "bie@gmail.com"
age : 24
```

>_MONGOSH   ^

---

localhost:27017   ...

myDatabase   **users**   ✕   +

myDatabase.users

0   1
DOCUMENTS   INDEXES

{} My Queries
Performance
Databases   ⟳ +
Search

Documents   Aggregations   Schema   **Indexes**   Validation

Create Index   ⟳ Refresh   VIEWING   **INDEXES**   SEARCH INDEXES

▸ admin
▸ blog
▸ config
▸ local
▾ myDatabase
   users   ...
▸ mystudentsdata
▸ test

Name and Definition	⬍	Type	⬍	Size	⬍	Usage	⬍	Properties	⬍
⌄ _id_		REGULAR ⓘ		36.9 KB		4 (since Fri Mar 22 2024)		UNIQUE ⓘ	
_id ↑									
⌄ email_1		REGULAR ⓘ		20.5 KB		0 (since Fri Mar 22 2024)			
email ↑									

>_MONGOSH   ^