

CREATE DATABASE AND COLLECTION IN MONGODB

>show dbs

MyDatabase1 72.00 KiB

admin 40.00 KiB

config 108.00 KiB

local 72.00 KiB

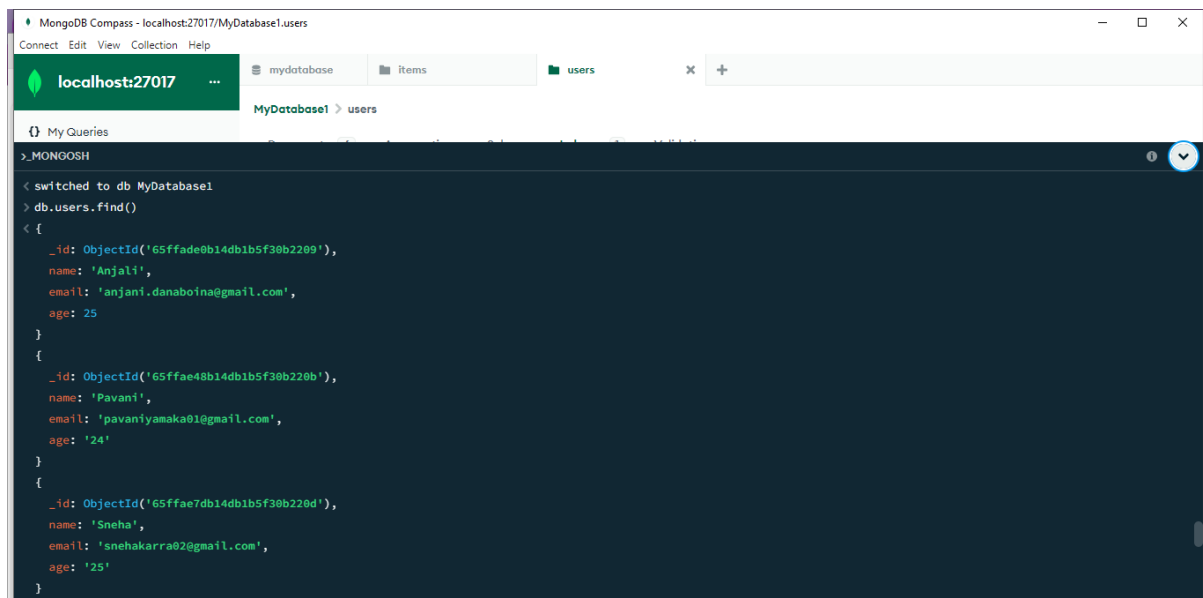
>use MyDatabase1

switched to db MyDatabase1

TO DISPLAY DATA

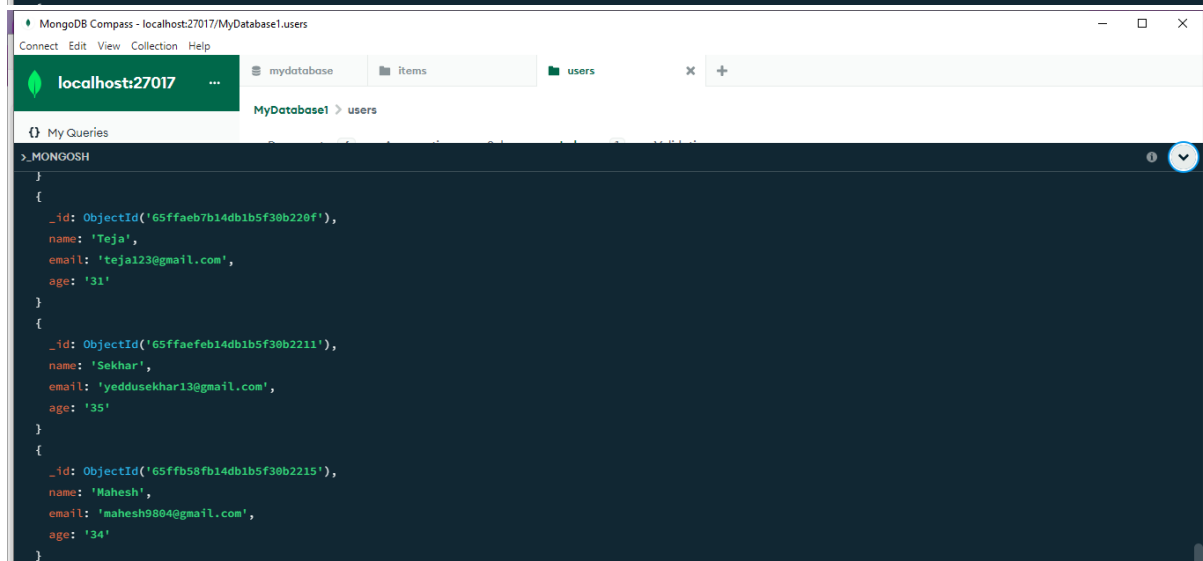
Syntax:

MyDatabase1> db.users.find()



The screenshot shows the MongoDB Compass interface. The top bar indicates the connection to 'localhost:27017/MyDatabase1.users'. The left sidebar shows the database structure with 'MyDatabase1' selected and 'users' as the current collection. The main panel displays the results of the query 'db.users.find()' in a JSON array format. The results show three documents, each with an '_id', 'name', 'email', and 'age' field.

```
>MONGOSH
< switched to db MyDatabase1
> db.users.find()
< [
  {
    _id: ObjectId('65ffade9b14db1b5f30b2209'),
    name: 'Anjali',
    email: 'anjani.danaboina@gmail.com',
    age: 25
  },
  {
    _id: ObjectId('65ffae48b14db1b5f30b220b'),
    name: 'Pavani',
    email: 'pavaniamaka01@gmail.com',
    age: '24'
  },
  {
    _id: ObjectId('65ffae7db14db1b5f30b220d'),
    name: 'Sneha',
    email: 'snehakarra02@gmail.com',
    age: '25'
  }
]
```



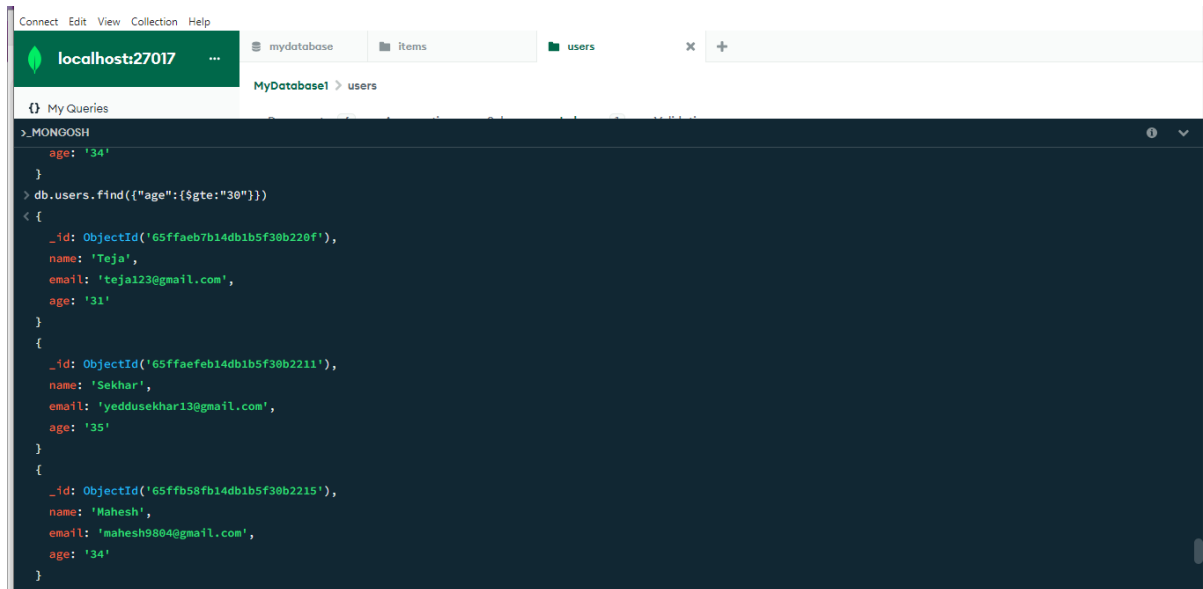
The screenshot shows the MongoDB Compass interface. The top bar indicates the connection to 'localhost:27017/MyDatabase1.users'. The left sidebar shows the database structure with 'MyDatabase1' selected and 'users' as the current collection. The main panel displays the results of the query 'db.users.find()' in a JSON array format. The results show three documents, each with an '_id', 'name', 'email', and 'age' field.

```
>MONGOSH
}
{
  _id: ObjectId('65ffae7b14db1b5f30b220f'),
  name: 'Teja',
  email: 'teja123@gmail.com',
  age: '31'
}
{
  _id: ObjectId('65ffae7b14db1b5f30b2211'),
  name: 'Sekhar',
  email: 'yeddusekhar13@gmail.com',
  age: '35'
}
{
  _id: ObjectId('65ff58fb14db1b5f30b2215'),
  name: 'Mahesh',
  email: 'mahesh9804@gmail.com',
  age: '34'
}
}
```

TO DISPLAY AGE GREATER THAN EQUAL TO 30

Syntax:

```
MyDatabase1> db.users.find({"age":{$gte:"30"}})
```

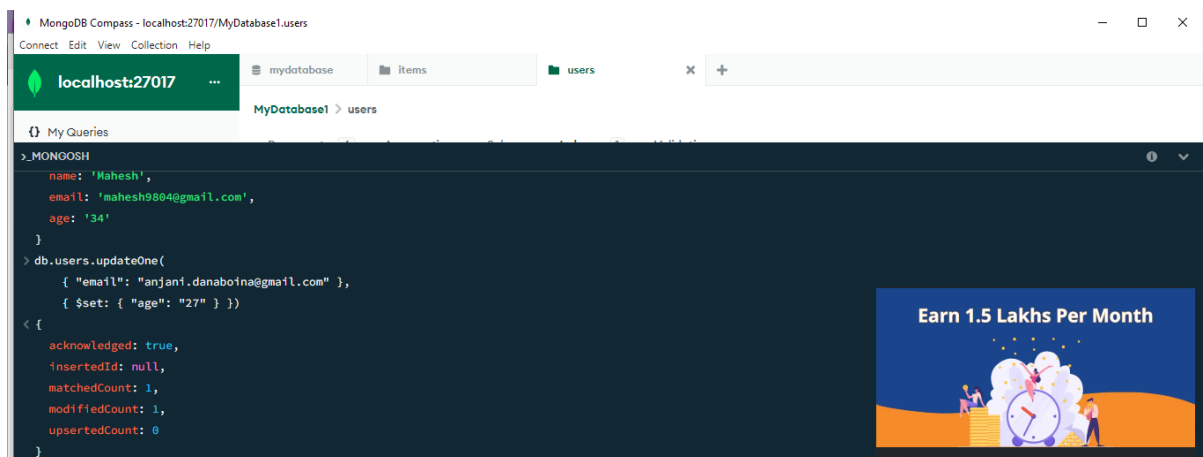


The screenshot shows the MongoDB Compass interface. The top bar indicates the connection to 'localhost:27017' and the current database 'MyDatabase1' with the 'users' collection selected. The 'My Queries' tab is active, displaying a MongoDB query: `> db.users.find({"age":{$gte:"30"}})`. The result shows three documents from the 'users' collection, all with ages greater than or equal to 30. The documents are: 1. User 'Teja' with age 31, email 'teja123@gmail.com'. 2. User 'Sekhar' with age 35, email 'yeddusekhar13@gmail.com'. 3. User 'Mahesh' with age 34, email 'mahesh9884@gmail.com'.

Update the age of a user with a specific email Address

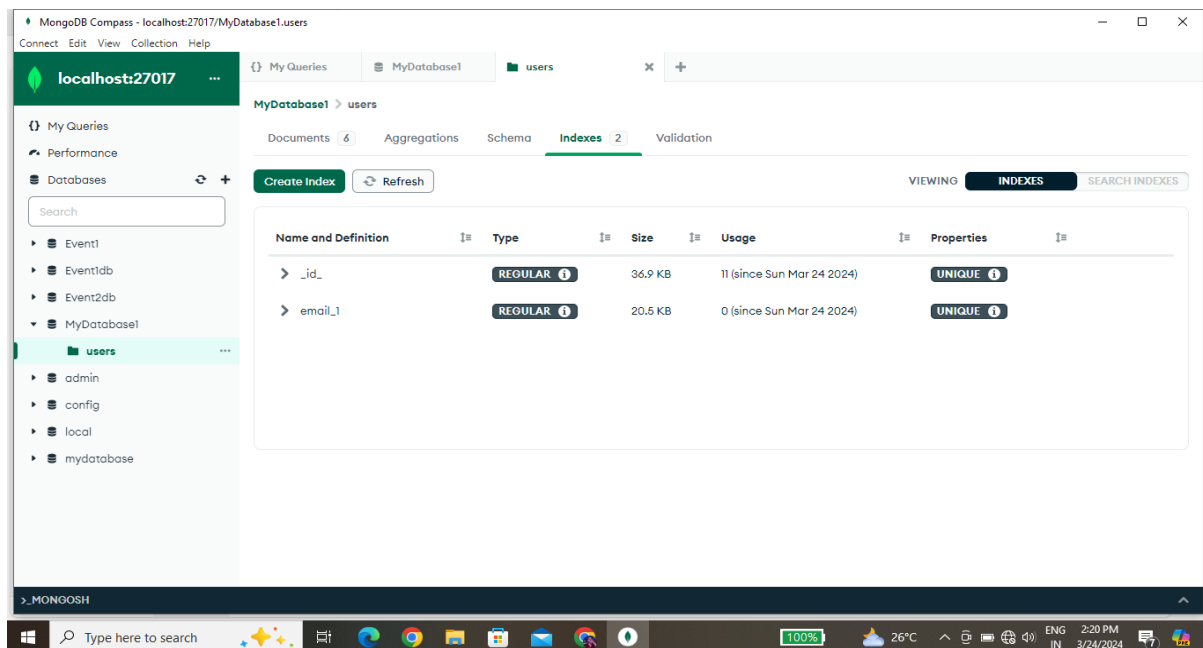
Syntax:

```
MyDatabase1> db.users.updateOne(  
  { "email": "anjani.danaboina@gmail.com" },  
  { $set: { "age": "27" } })
```



The screenshot shows the MongoDB Compass interface after executing an update query. The top bar shows the connection to 'localhost:27017' and the current database 'MyDatabase1' with the 'users' collection selected. The 'My Queries' tab is active, displaying the query: `> db.users.updateOne({ "email": "anjani.danaboina@gmail.com" }, { $set: { "age": "27" } })`. The result shows the updated document for user 'Mahesh' with age 34 and email 'mahesh9884@gmail.com'. The result also includes the following fields: `acknowledged: true`, `insertedId: null`, `matchedCount: 1`, `modifiedCount: 1`, and `upsertedCount: 0`. A promotional banner for 'Earn 1.5 Lakhs Per Month' is visible in the bottom right corner.

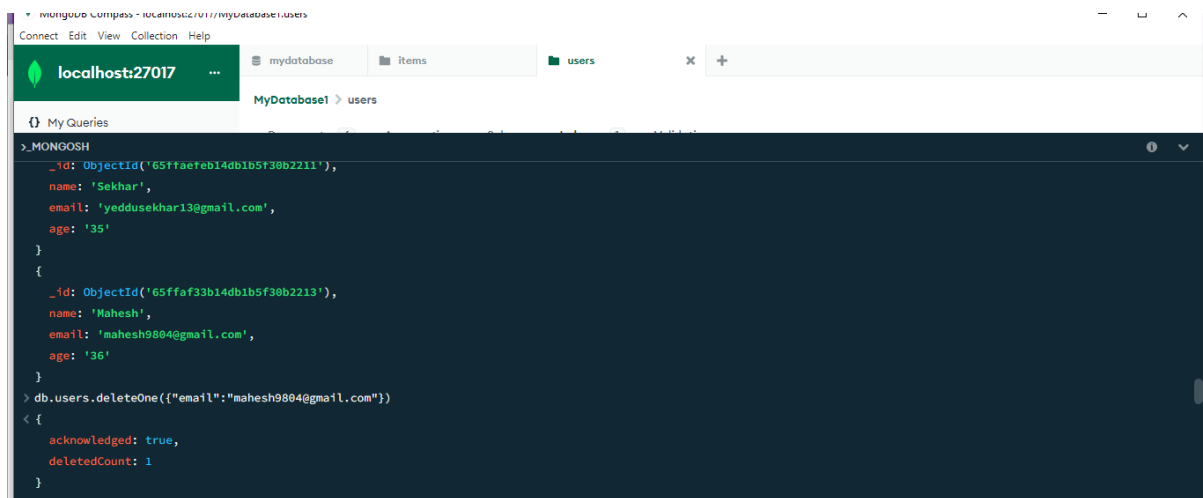
Create an index on the email field of the users collection.



Delete a user document based on a specific email Address.

Syntax:

MyDatabase1> [db.users.deleteOne\({"email":"mahesh9804@gmail.com"}\)](#)



Selection deleted