```
// create new database
use new_users;

// Show all dbs
show dbs;

// Show current db
Db

// Show collections
show collections;

// Creating collection in new_users
db.createCollection('users');

// Showing all data from user's collection , here is empty
db.users.find();

// Inserting mupltiple data as given in task // box bracket
db.users.insertMany([{
name : "Ganesh",
age : 25,
email : "ganesh@hotmail.com",
currency : "NPR",
bankDetails : [{
bankName : "Global IME",
balance : 1000
},{
bankName : "NIC ASIA",
balance : 2000
},{
bankName : "Himalaya Bank",
balance : 5263
}]
},{
name : "Amir",
age : 26,
email : "Amir@gmail.com",
currency : "INR",
bankDetails : [{
bankName : "State Bank of India",
balance : 500
},{
bankName : "Kashmir Bank",
balance : 600
},{
bankName : "Modi Finance Bank",
balance : 668
}]
},{
name : "Helen",
age : 27,
email : "helen@outlook.com",
currency : "USD",
bankDetails : [{
```
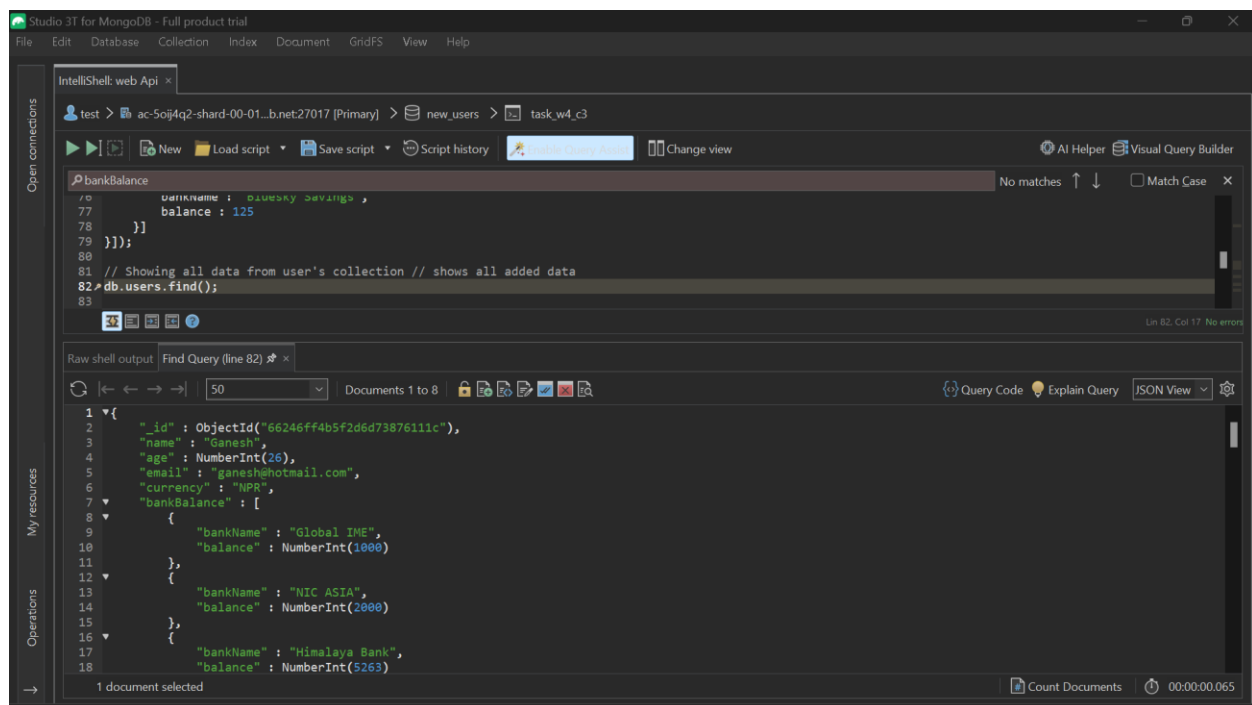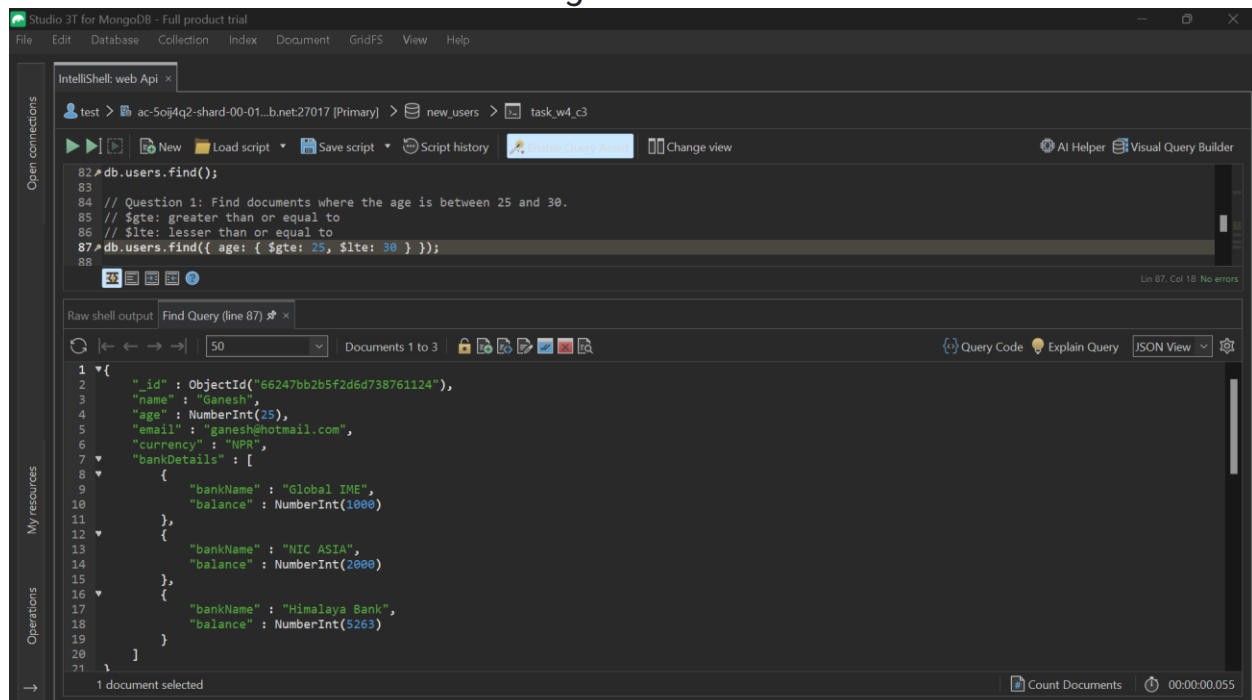
```
bankName : "Bank of America",
balance : 566
},{
bankName : "California Bank",
balance : 555
},]
},{
name : "Miller",
age : 44,
email : "millerx@proton.com",
currency : "CAD",
bankDetails : [{
bankName : "Toronto Bikas Bank",
balance : 600
},{
bankName : "Flywire Bank",
balance : 988
},{
bankName : "Bluesky Savings",
balance : 125
}]
}]);
// Showing all data from user's collection // shows all added data
db.users.find();
```

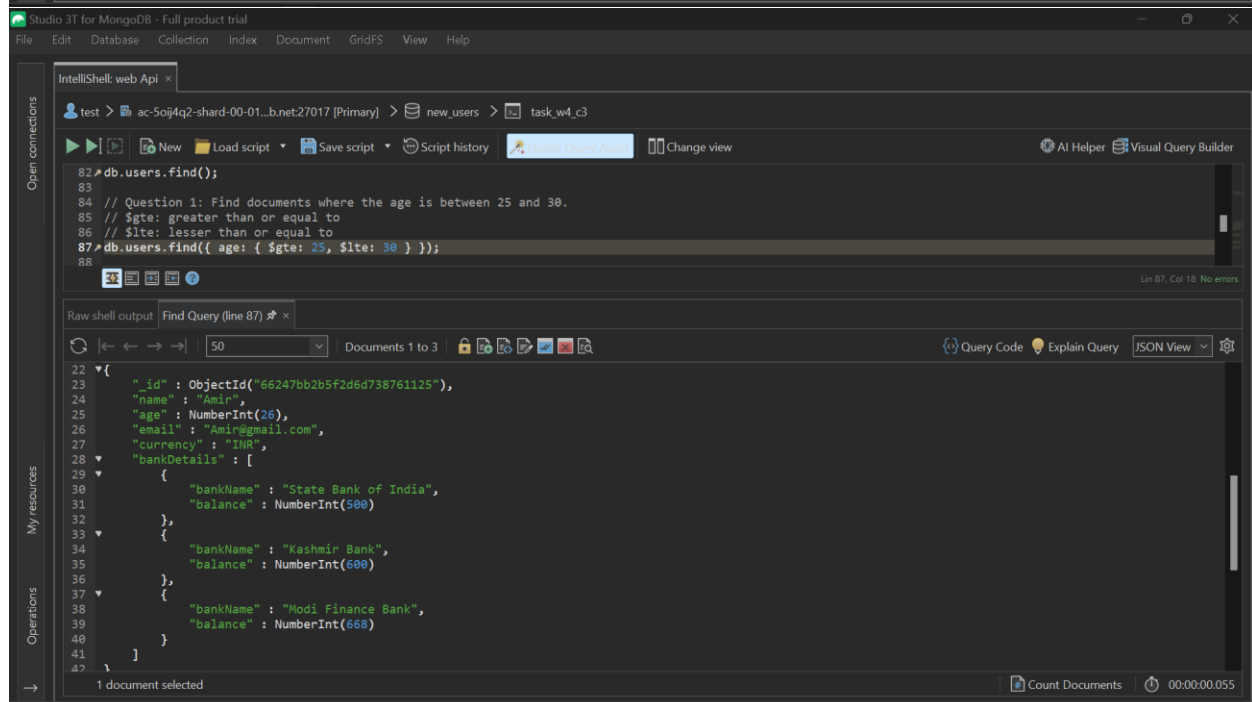# 1. Find documents where the age is between 25 and 30:
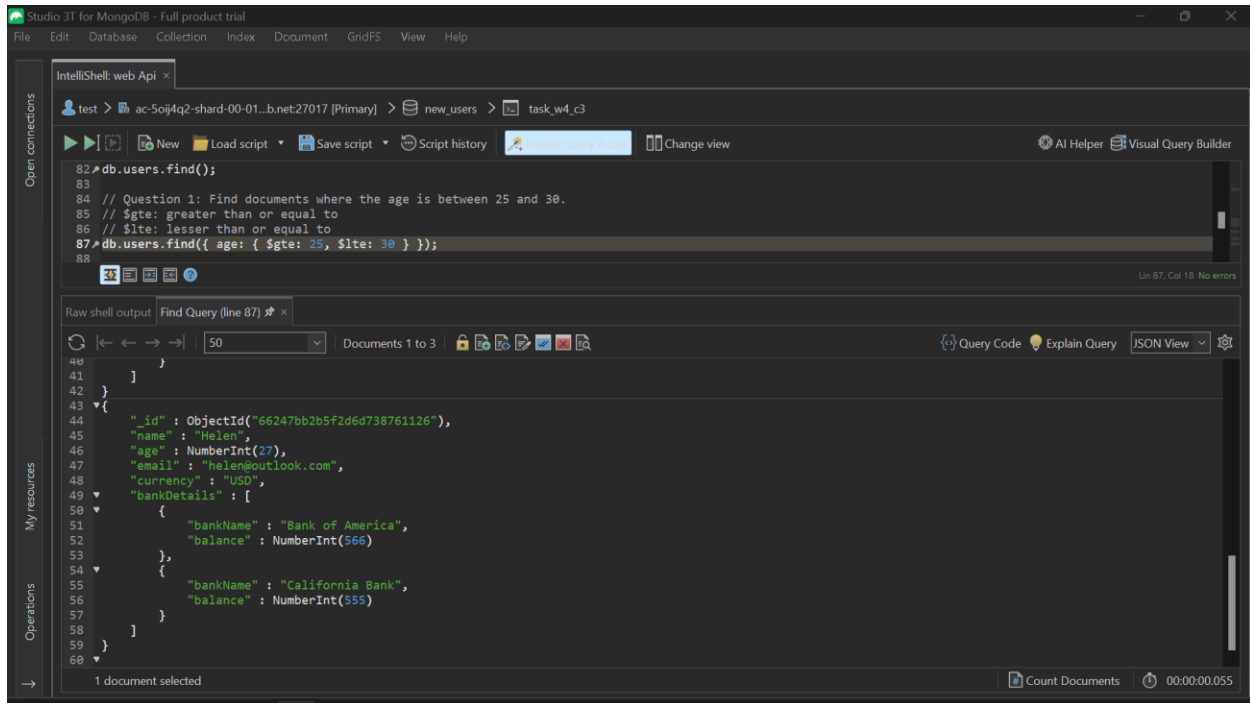
2. Find all users with a bank balance > 500 in any bank.
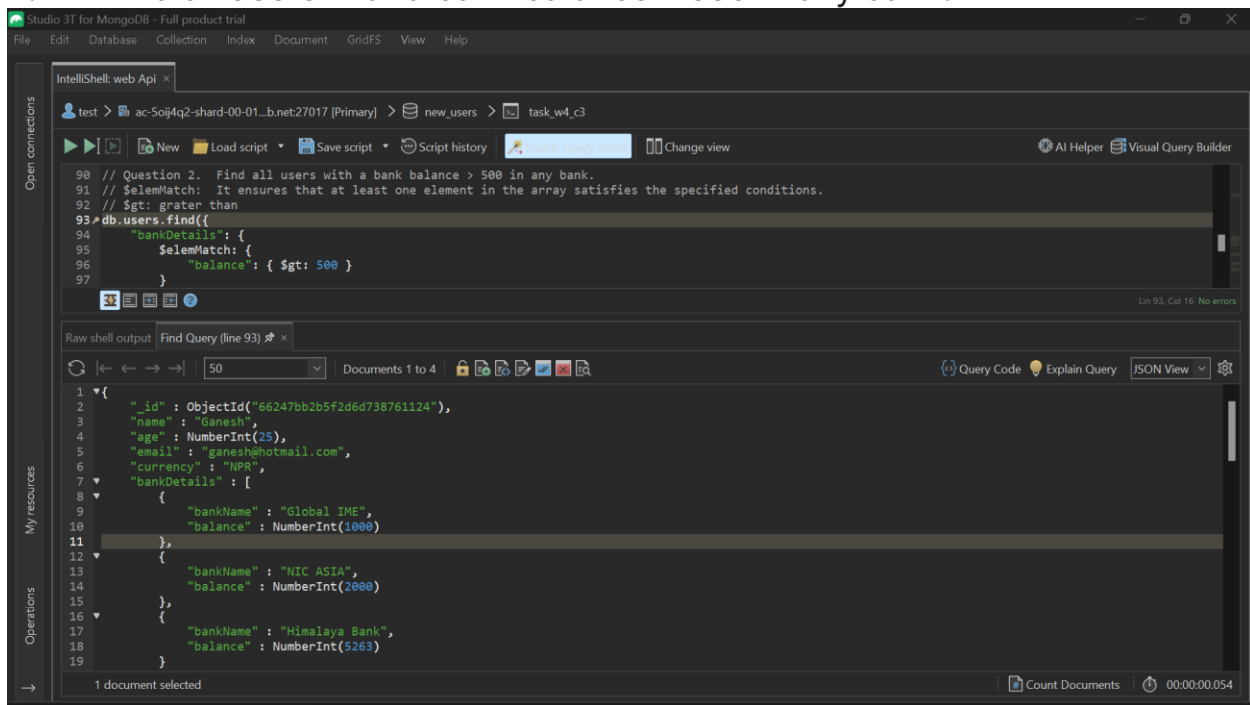
File   Edit   Database   Collection   Index   Document   GridFS   View   Help

IntelliShell: web Api

test > ac-5oij4q2-shard-00-01...b.net:27017 [Primary] > new_users > task_w4_c3

New   Load script   Save script   Script history   Enable Query Assist   Change view   AI Helper   Visual Query Builder

```
90  // Question 2.  Find all users with a bank balance > 500 in any bank.
91  // $elemMatch:  It ensures that at least one element in the array satisfies the specified conditions.
92  // $gt: grater than
93  db.users.find({
94      "bankDetails": {
95          $elemMatch: {
96              "balance": { $gt: 500 }
97          }
```

Lin 93. Col 16 No errors

Raw shell output   Find Query (line 93)

50   Documents 1 to 4   Query Code   Explain Query   JSON View

```
22 ▾{
23      "_id" : ObjectId("66247bb2b5f2d6d738761125"),
24      "name" : "Amir",
25      "age" : NumberInt(26),
26      "email" : "Amir@gmail.com",
27      "currency" : "INR",
28 ▾    "bankDetails" : [
29 ▾        {
30              "bankName" : "State Bank of India",
31              "balance" : NumberInt(500)
32          },
33 ▾        {
34              "bankName" : "Kashmir Bank",
35              "balance" : NumberInt(600)
36          },
37 ▾        {
38              "bankName" : "Modi Finance Bank",
39              "balance" : NumberInt(668)
40          }
```

1 document selected   Count Documents   00:00:00.054

```
90  // Question 2.  Find all users with a bank balance > 500 in any bank.
91  // $elemMatch:  It ensures that at least one element in the array satisfies the specified conditions.
92  // $gt: grater than
93  db.users.find({
94      "bankDetails": {
95          $elemMatch: {
96              "balance": { $gt: 500 }
97          }
```

Lin 93. Col 16 No errors

Raw shell output   Find Query (line 93)
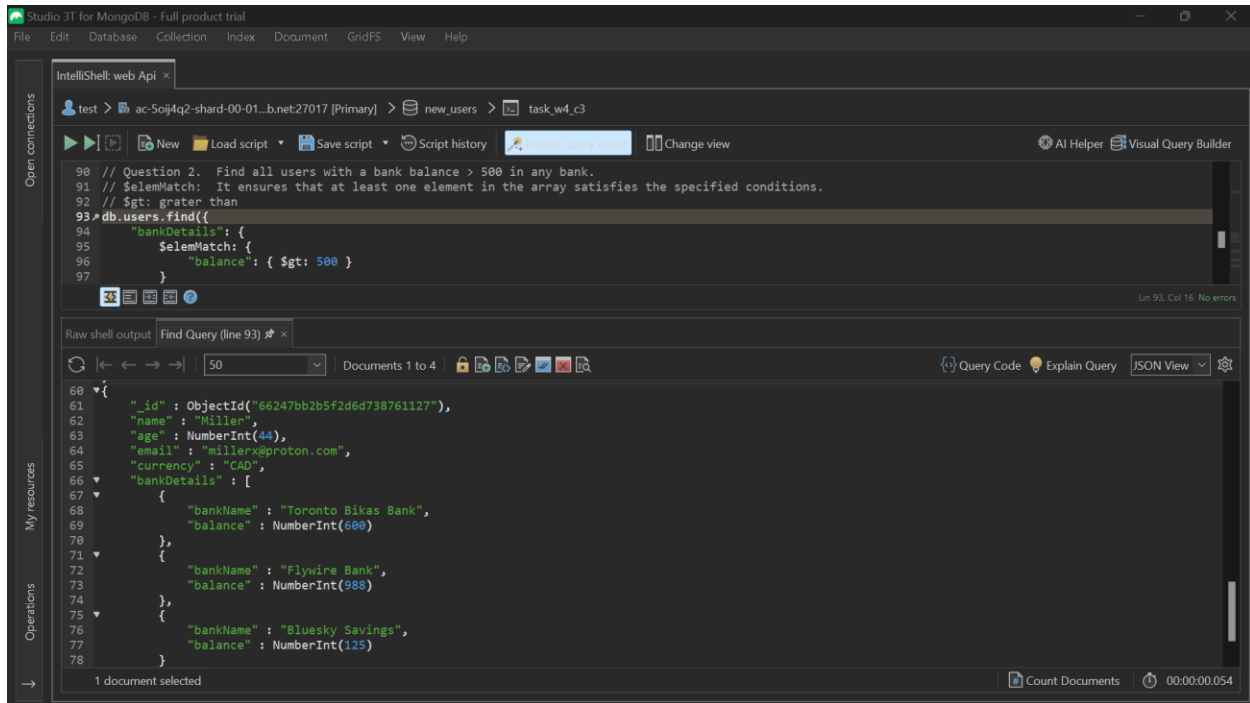
50   Documents 1 to 4   Query Code   Explain Query   JSON View

```
43 ▾{
44      "_id" : ObjectId("66247bb2b5f2d6d738761126"),
45      "name" : "Helen",
46      "age" : NumberInt(27),
47      "email" : "helen@outlook.com",
48      "currency" : "USD",
49 ▾    "bankDetails" : [
50 ▾        {
51              "bankName" : "Bank of America",
52              "balance" : NumberInt(566)
53          },
54 ▾        {
55              "bankName" : "California Bank",
56              "balance" : NumberInt(555)
57          }
58      ]
59  }
60 ▾{
61      "_id" : ObjectId("66247bb2b5f2d6d738761127"),
```

1 document selected   Count Documents   00:00:00.054

```
90  // Question 2.  Find all users with a bank balance > 500 in any bank.
91  // $elemMatch:  It ensures that at least one element in the array satisfies the specified conditions.
92  // $gt: grater than
93  db.users.find({
94      "bankDetails": {
95          $elemMatch: {
96              "balance": { $gt: 500 }
97          }
```
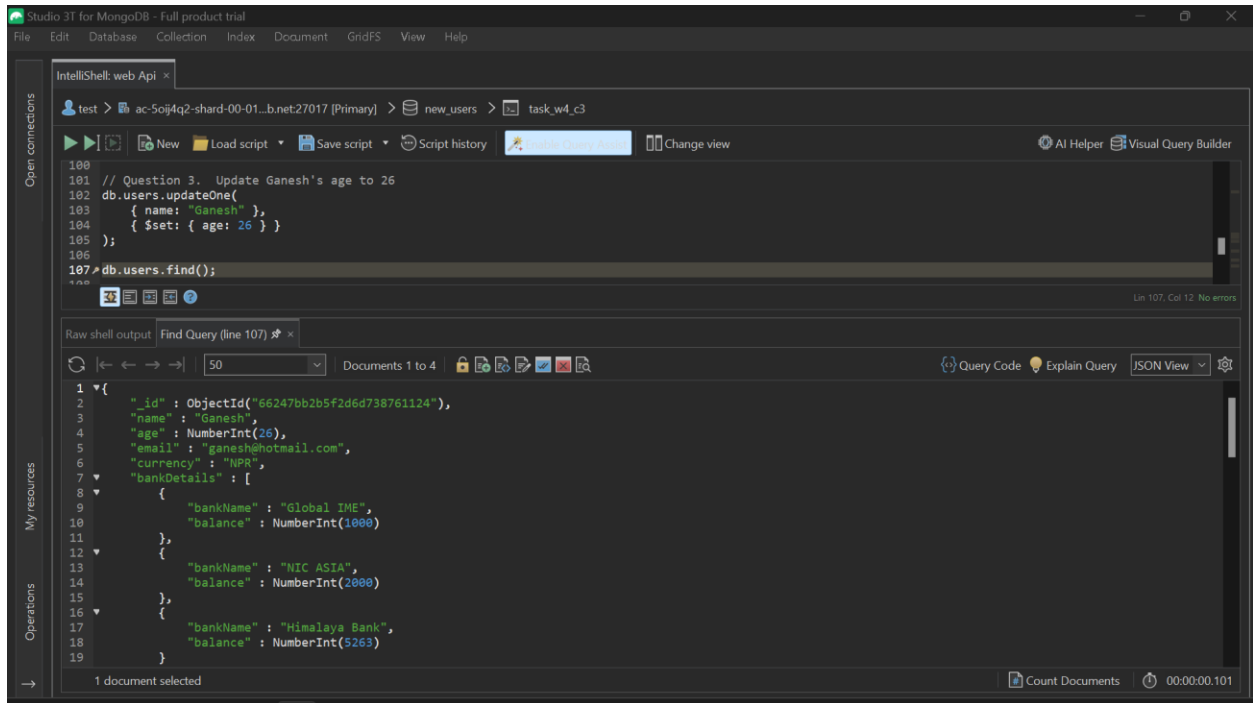
```
60  {
61      "_id" : ObjectId("66247bb2b5f2d6d738761127"),
62      "name" : "Miller",
63      "age" : NumberInt(44),
64      "email" : "millerx@proton.com",
65      "currency" : "CAD",
66      "bankDetails" : [
67          {
68              "bankName" : "Toronto Bikas Bank",
69              "balance" : NumberInt(600)
70          },
71          {
72              "bankName" : "Flywire Bank",
73              "balance" : NumberInt(988)
74          },
75          {
76              "bankName" : "Bluesky Savings",
77              "balance" : NumberInt(125)
78          }
```

3.   Update Ganesh's age to 26


```
100
101  // Question 3.  Update Ganesh's age to 26
102  db.users.updateOne(
103      { name: "Ganesh" },
104      { $set: { age: 26 } }
105  );
106
107  db.users.find();
```

```
1  {
2      "acknowledged" : true,
3      "insertedId" : null,
4      "matchedCount" : 1.0,
5      "modifiedCount" : 1.0,
6      "upsertedCount" : 0.0
7  }
8
```
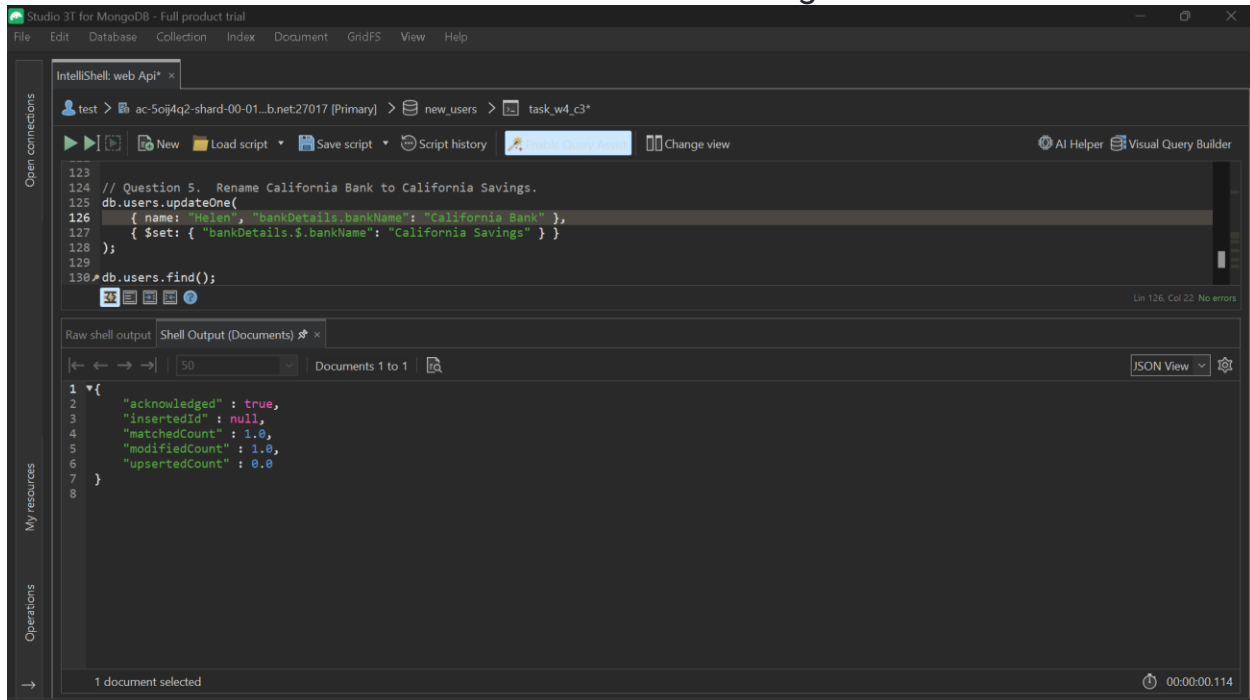
## 4.	Add new bank for Helen (California State Bank, 9K)

5. Rename California Bank to California Savings.


```
123
124  // Question 5.  Rename California Bank to California Savings.
125  db.users.updateOne(
126      { name: "Helen", "bankDetails.bankName": "California Bank" },
127      { $set: { "bankDetails.$.bankName": "California Savings" } }
128  );
129
130  db.users.find();
```

```
1 ▼{
2      "acknowledged" : true,
3      "insertedId" : null,
4      "matchedCount" : 1.0,
5      "modifiedCount" : 1.0,
6      "upsertedCount" : 0.0
7  }
8
```
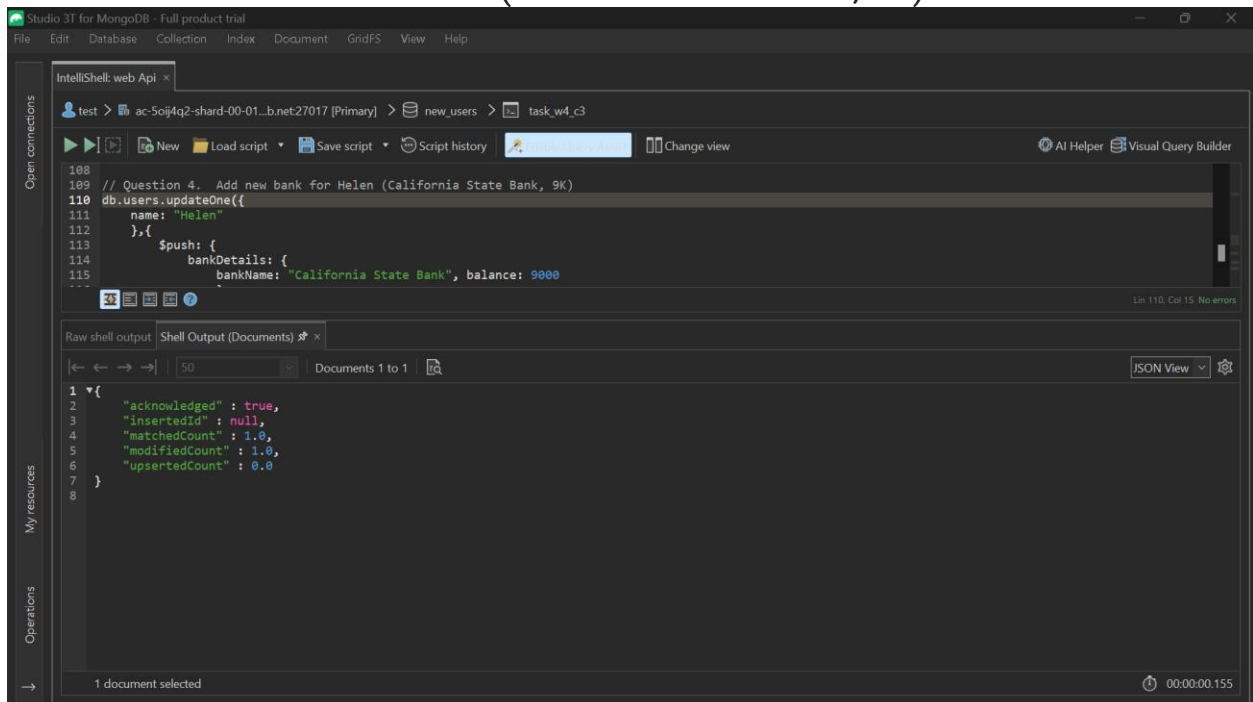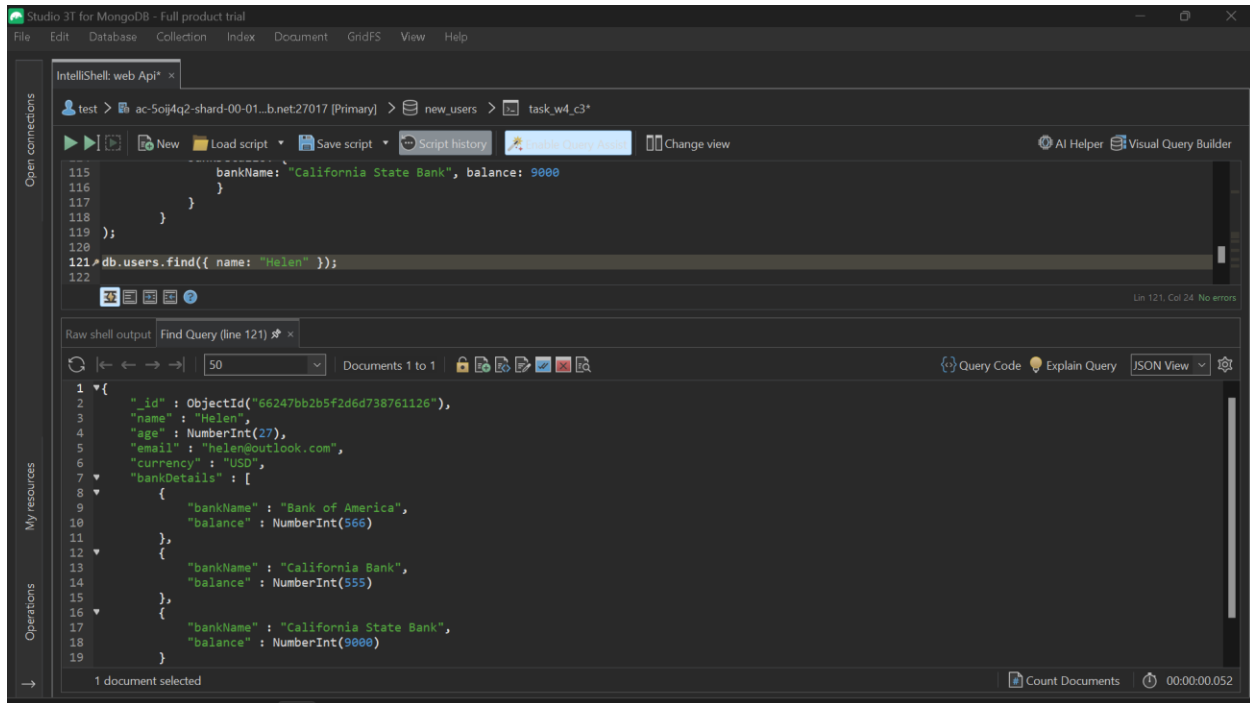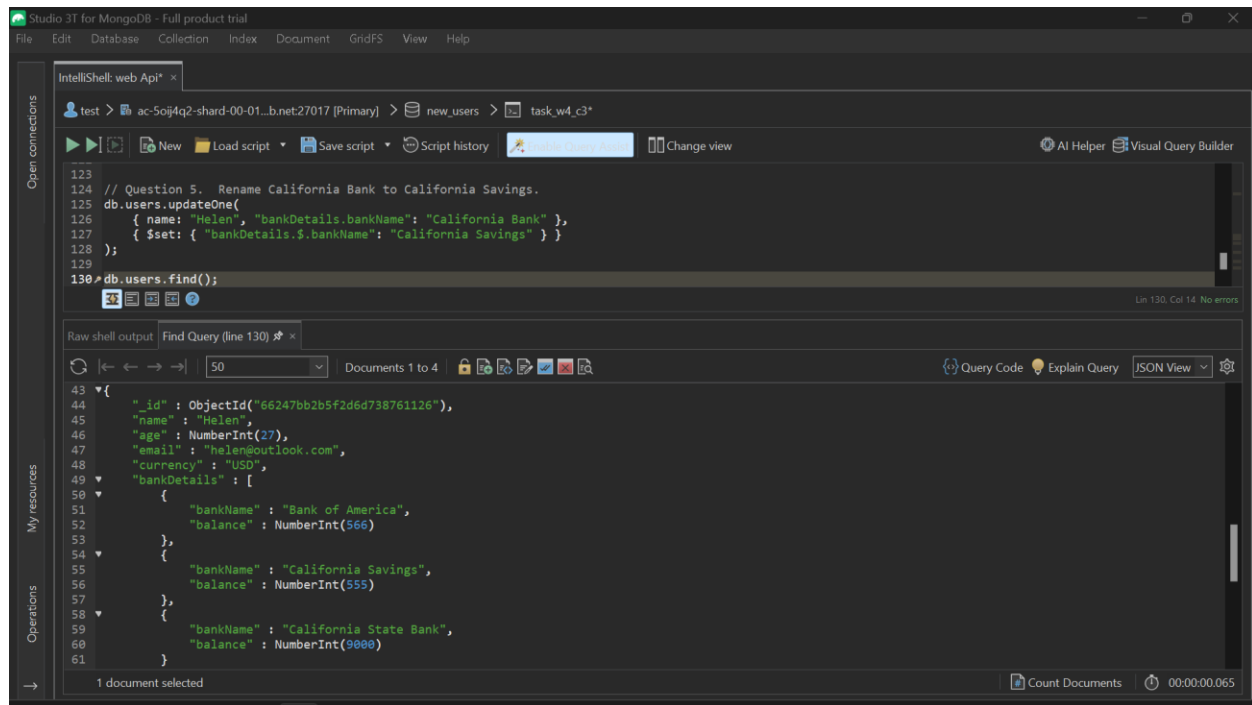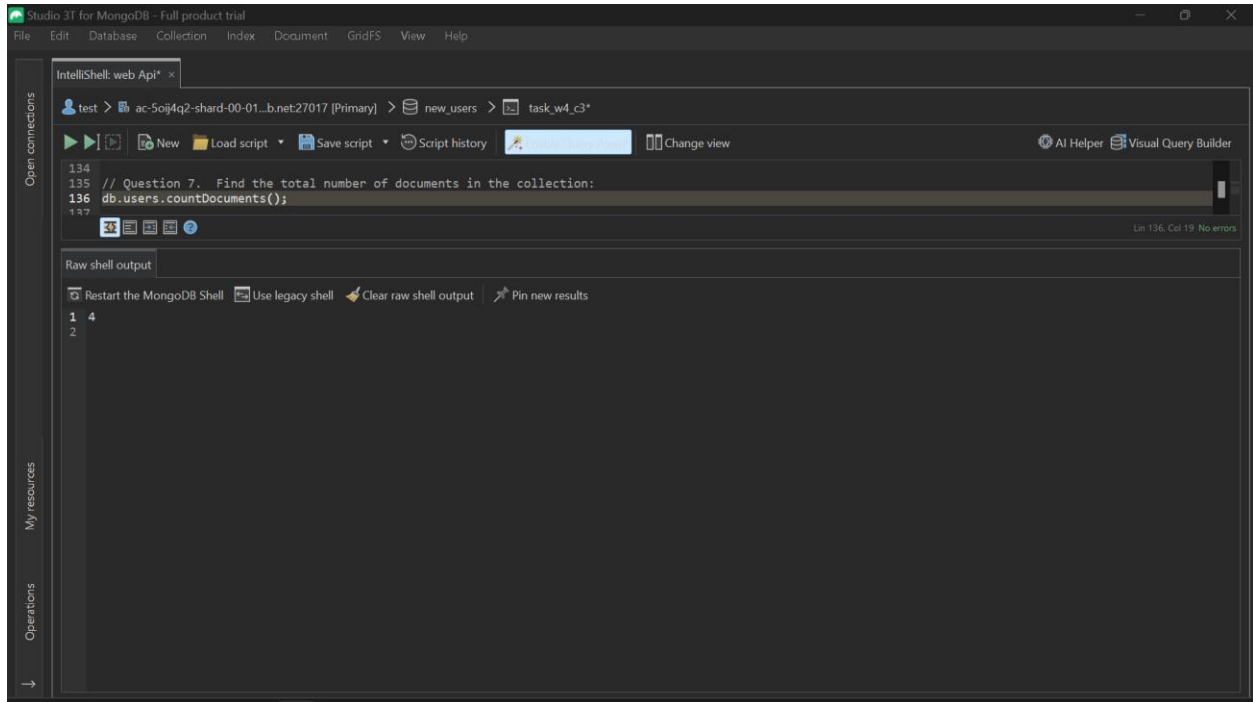
6. Find documents with only the name and email fields included in the result:
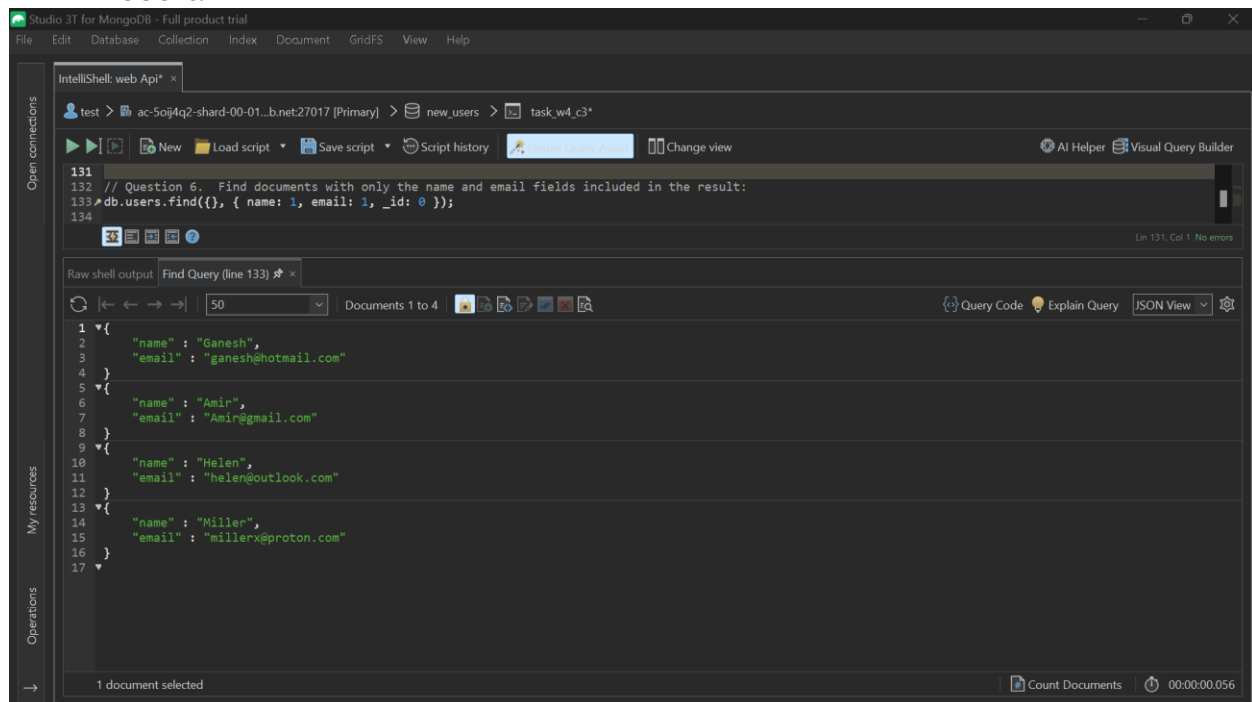
7. Find the total number of documents in the collection:

8. Find the user with the highest bank balance: