Paras Singh, 165-114-232

You submit this file with answers (in the provided spaces). Name the file “L07_ID#_LASTNAME.pdf”.

Tasks

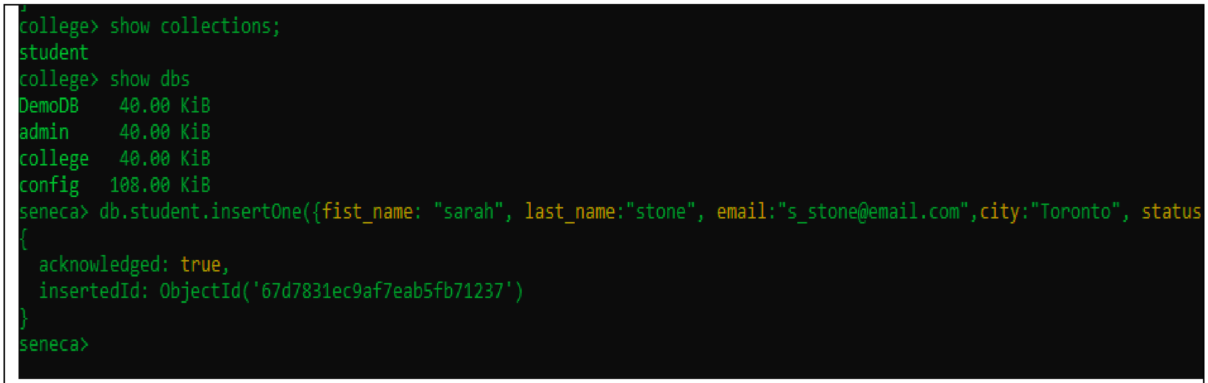
1. In this question you create a new database named *seneca* and a collection *student*. We store student data in this collection.
   - ➢ use seneca

   This command makes “seneca” your current database. However, the database will not be created until you insert a document into this database.

   db.collection_name.insertOne({})

   Insert a new document into your collection *student* with the following data:

   first_name: Sarah
   last_name: Stone
   email: s_stone@email.com
   city: Toronto
   status: full-time
   gpa: 3.4
   program: CPA

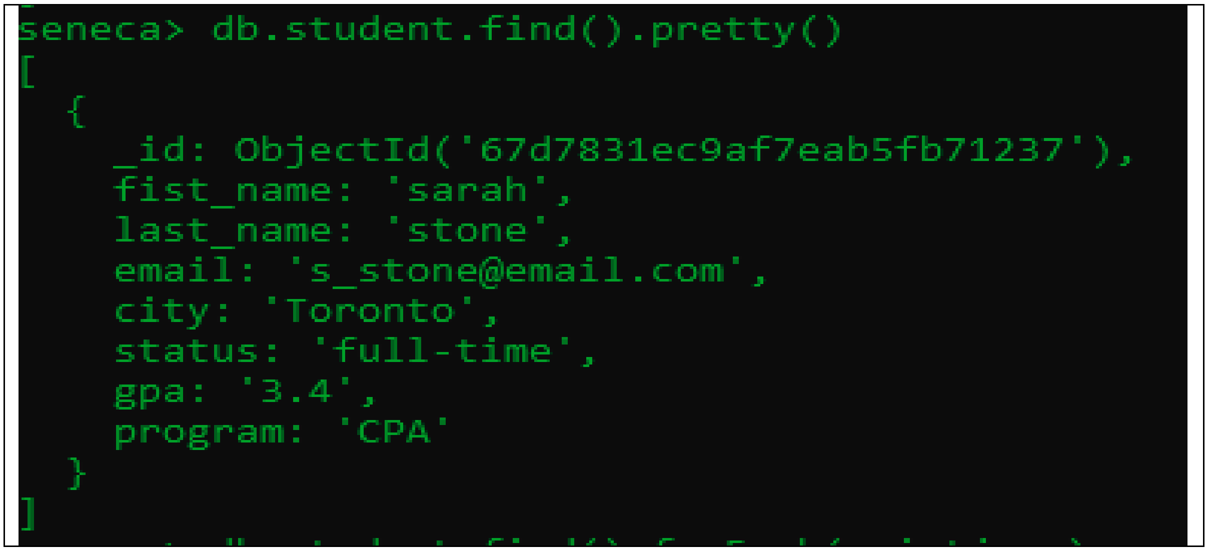


```
college> show collections;
student
college> show dbs
DemoDB    40.00 KiB
admin     40.00 KiB
college   40.00 KiB
config   108.00 KiB
seneca> db.student.insertOne({fist_name: "sarah", last_name:"stone", email:"s_stone@email.com",city:"Toronto", status
{
  acknowledged: true,
  insertedId: ObjectId('67d7831ec9af7eab5fb71237')
}
seneca>
```

2. Write a command to check if the document has been created successfully.
   You use *find()* method to search and fetch documents.

   See the following example:

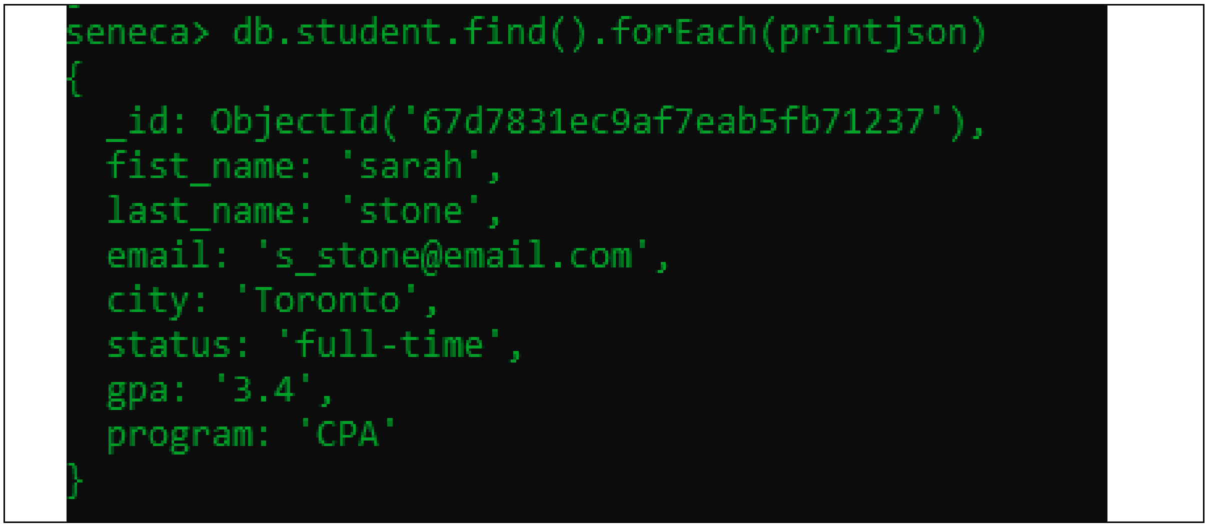
   ➢ db.student.find()



```
seneca> db.student.find().pretty()
[
  {
    _id: ObjectId('67d7831ec9af7eab5fb71237'),
    fist_name: 'sarah',
    last_name: 'stone',
    email: 's_stone@email.com',
    city: 'Toronto',
    status: 'full-time',
    gpa: '3.4',
    program: 'CPA'
  }
]
```

To see the result in *JSON* format, you can run the following statement:

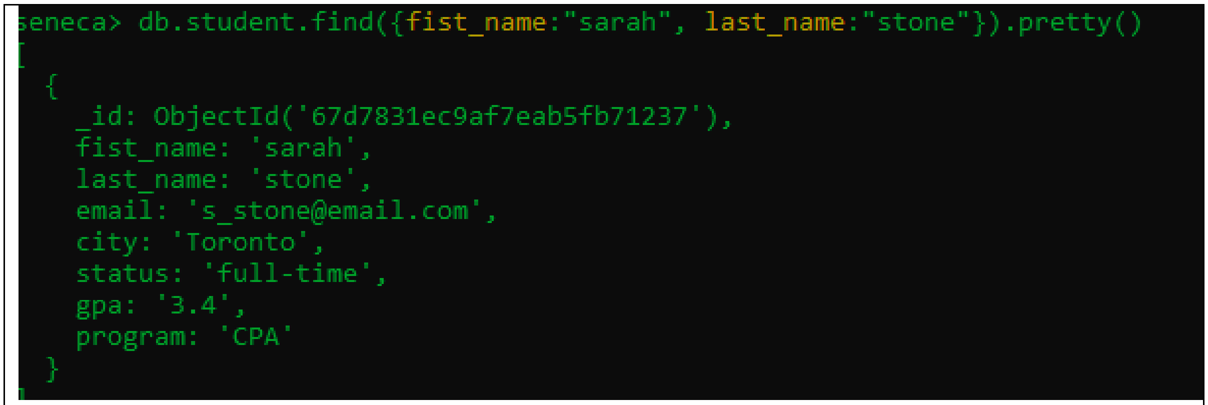
➢ db.student.find().forEach(printjson)



```
seneca> db.student.find().forEach(printjson)
{
  _id: ObjectId('67d7831ec9af7eab5fb71237'),
  fist_name: 'sarah',
  last_name: 'stone',
  email: 's_stone@email.com',
  city: 'Toronto',
  status: 'full-time',
  gpa: '3.4',
  program: 'CPA'
}
```

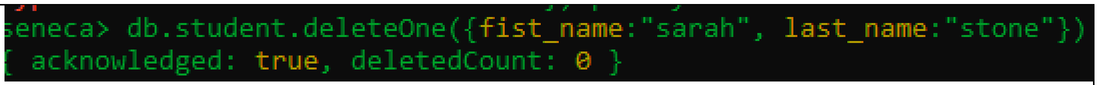
How many fields are in your document? __8_(_id, first_name, last_name, email, city, status, gpa, program)__
Is there any new field added to your document? ___yes___
If yes, what is the name of the field? _ **_id** (automatically added by MongoDB)____

3. Write a command to remove the document that you created in Question 1. (We have only one document at this time, but when removing documents make sure you will not remove some other documents if not needed. So, make sure your command will remove "Sarah Stone" from your collection. For now, we assume that we do not have duplicate names in our database.)
Note: To avoid making mistakes, you can first write a find command with the proper criteria to see if the required document is fetched. Then, you can use the same criteria in your delete statement. (Write the statement to remove "Sarah Stone" from the database in the box below.)



```
seneca> db.student.find({fist_name:"sarah", last_name:"stone"}).pretty()
[
  {
    _id: ObjectId('67d7831ec9af7eab5fb71237'),
    fist_name: 'sarah',
    last_name: 'stone',
    email: 's_stone@email.com',
    city: 'Toronto',
    status: 'full-time',
    gpa: '3.4',
    program: 'CPA'
  }
]
```

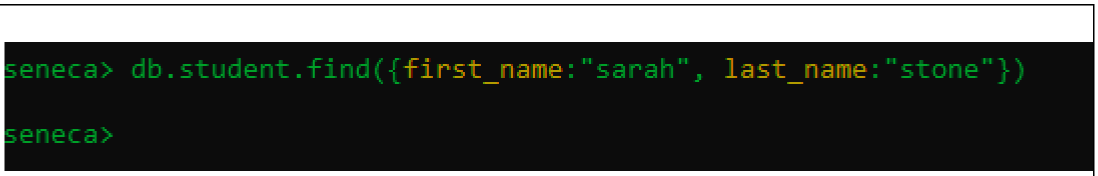
What is the message as a result of your delete statement? Copy the message in the following box:



```
seneca> db.student.deleteOne({fist_name:"sarah", last_name:"stone"})
{ acknowledged: true, deletedCount: 0 }
```

I made a typo that's why it is showing deletedCount= 0
however I ran the commands again, and corrected first_name previously it was fist_name so you can see in the below screenshot nothing is returned so, it is successfully executed.

To see if the document is removed successfully, write a search statement to see if the document exists. (We look for one document not all).



```
seneca> db.student.find({first_name:"sarah", last_name:"stone"})

seneca>
```

4. We want to add some students to our collection, but this time, we define the value for the *_id* field. (If the _id is not defined in your document, it will be added automatically.)

   _id: 1001
   first_name: Sarah
   last_name: Stone
   email: s_stone@email.com
   city: Toronto
   status: full-time
   gpa: 3.4
   program: CPA

   _id: 1002
   first_name: Jack
   last_name: Adam
   email: j_adam@email.com
   city: North York
   status: part-time
   gpa: 3.6
   program: CPA

   To add these students, we want to store these documents into a variable first.
   Define a variable named *starray* and add these two document to the variable. (You are storing more than one document so you need to define an array.

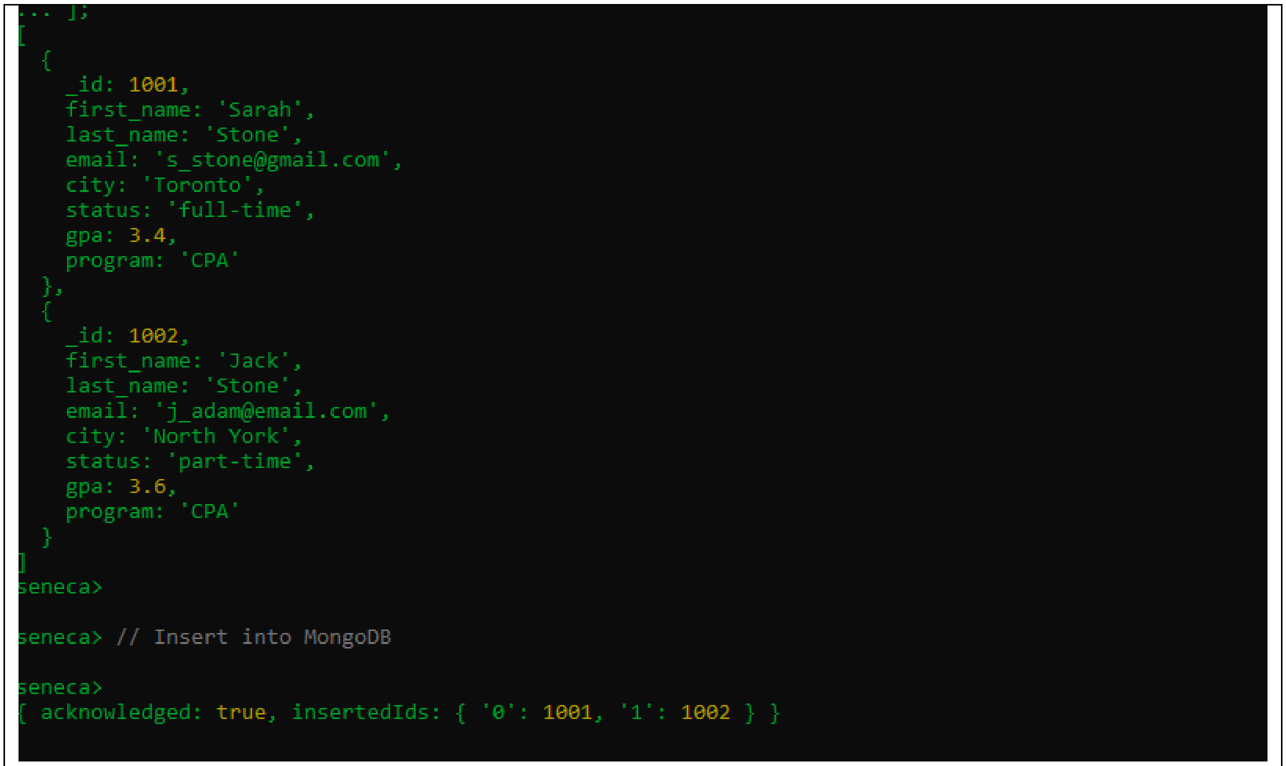
```
starray = [
   {
      _id: 1001,
      first_name: "Sarah",
      last_name: "Stone",
      email: "s_stone@email.com",
      city: "Toronto",
      status: "full-time",
      gpa: 3.4,
      program: "CPA"
   },
   {
      _id: 1002,
      first_name: "Jack",
      last_name: "Adam",
      email: "j_adam@email.com",
      city: "North York",
      status: "part-time",
      gpa: 3.6,
      program: "CPA"
   }

]
```

Now, use the *starray* array to insert the documents to your collection *student*. Write your insert statement in the box bellow.

```
db.student.insertMany(starray)
```

What message is displayed after you execute the insert statement. Copy the message in the following box:



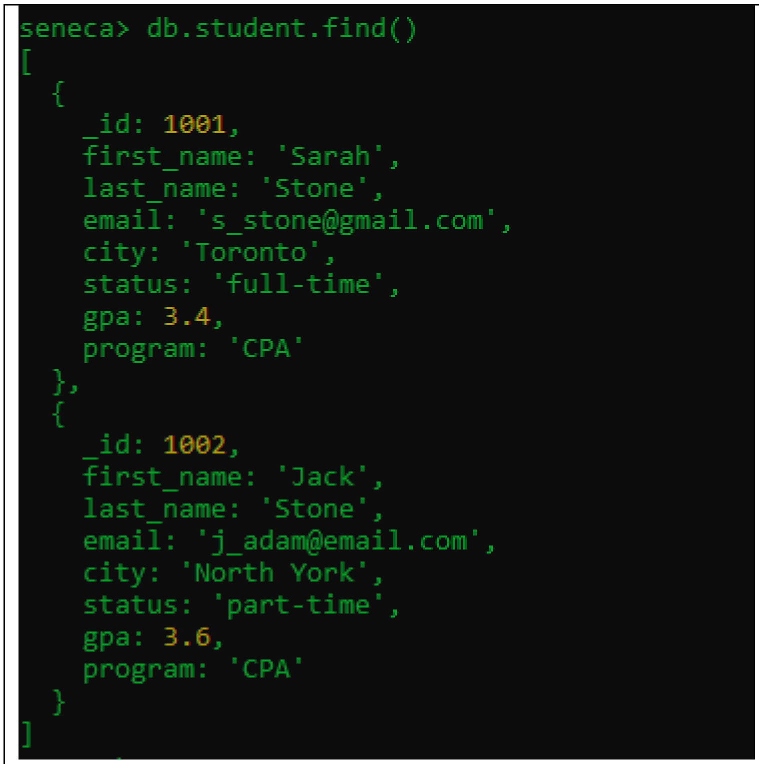
```
... ];
[
  {
    _id: 1001,
    first_name: 'Sarah',
    last_name: 'Stone',
    email: 's_stone@gmail.com',
    city: 'Toronto',
    status: 'full-time',
    gpa: 3.4,
    program: 'CPA'
  },
  {
    _id: 1002,
    first_name: 'Jack',
    last_name: 'Stone',
    email: 'j_adam@email.com',
    city: 'North York',
    status: 'part-time',
    gpa: 3.6,
    program: 'CPA'
  }
]
seneca>

seneca> // Insert into MongoDB

seneca>
{ acknowledged: true, insertedIds: { '0': 1001, '1': 1002 } }
```

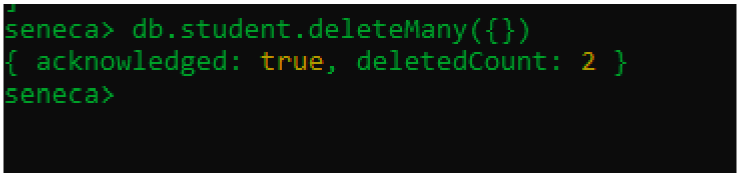
Write a statement that shows all documents inserted in your collection *student*:

```
db.student.find()
```



```
seneca> db.student.find()
[
  {
    _id: 1001,
    first_name: 'Sarah',
    last_name: 'Stone',
    email: 's_stone@gmail.com',
    city: 'Toronto',
    status: 'full-time',
    gpa: 3.4,
    program: 'CPA'
  },
  {
    _id: 1002,
    first_name: 'Jack',
    last_name: 'Stone',
    email: 'j_adam@email.com',
    city: 'North York',
    status: 'part-time',
    gpa: 3.6,
    program: 'CPA'
  }
]
```

5. Write a statement to remove all documents in the *student* collection.



```
seneca> db.student.deleteMany({})
{ acknowledged: true, deletedCount: 2 }
seneca>
```

6. Write a statement to drop the database *seneca*.

Before dropping a database, make sure your current database is the one you want to delete. For this question, we want to delete (drop) the *seneca* database.

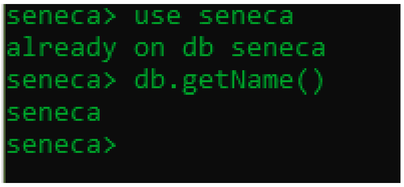
You can write the *use* statement before removing the database to make sure your current database is *seneca*.

- use seneca

Or, you can write the *db* or db.getName() statement to see the name of your current database:

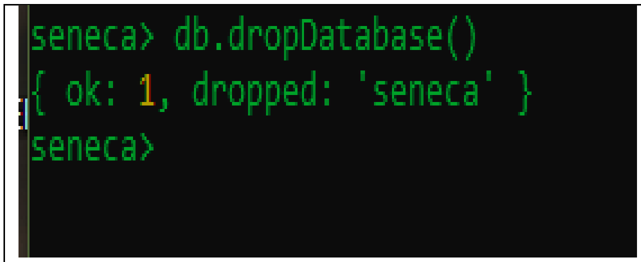
- db
- db.getName()

If your current database is not *seneca*, write the use statement to switch to *seneca* and then delete the database.



```
seneca> use seneca
already on db seneca
seneca> db.getName()
seneca
seneca>
```

What message is displayed after you execute the drop statement? Copy the message in the box below:



```
seneca> db.dropDatabase()
{ ok: 1, dropped: 'seneca' }
seneca>
```