Lab10b Mongo 1 lab

Was called Lab 7 – Week 9 (nasim) now in week 10 for 15-week semester

# MongoDB – Create/Delete Database/Collection/Documents

## Objective

In this Lab, you learn to create and remove MongoDB

* Databases
* Collections
* Documents

## Getting Started

Open your Windows command prompt and go the following directory where MongoDB is installed:

* cd C:\Program Files\MongoDB\Server\4.4\**bin**

As of March 2021, the version is 4.4. Change this to your version if necessary

To run MongoDB, execute ***mongod***

* mongod

When MongoDB starts successfully, open another Windows command prompt and go the same *bin* directory:

* cd C:\Program Files\MongoDB\Server\4.4\**bin**

and execute ***mongo***

* mongo

This opens the command line shell where you can work with MongoDB.

Or you can create a batch file: This may have been done at the end of the install document

Create a new text file named *start\_mongodb.bat* using a text editor and save the following command in it:

start call "C:\Program Files\MongoDB\Server\4.4\bin\mongod.exe"

start call "C:\Program Files\MongoDB\Server\4.4\bin\mongo.exe"

Every time you want to run MongoDB and open the client shell, execute this file.

To test your solutions before you submit, test them individually and make sure they work correctly.

**SUBMISSION:**

You submit this file with answers (in the provided spaces). Name the file **“L10b\_ID#\_LASTNAME.docx”.**

## 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**

you will see the following 🡺 switched to db 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({}) --** executing this will insert a document and the seneca will appear

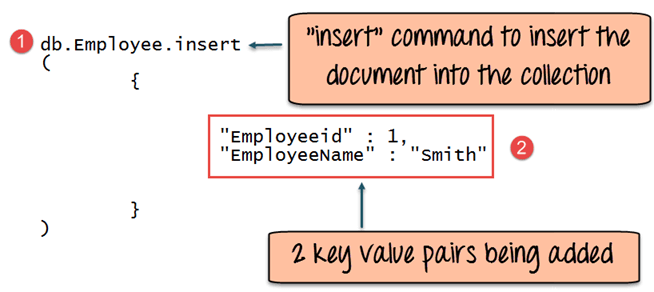
Try the following to see the error message. This will be a common error when you start out

**db.collection\_name.insertone({})** Can you see what the error is?

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

first\_name: Sara  
last\_name: Stone  
email: [stone@email.com](mailto:stone@email.com)  
city: Toronto  
status: full-time  
gpa: 2.4  
program: CPD

SAMPLE INSERT:

[](https://www.guru99.com/images/MongoDB/112115_0607_Introductio5.png)

watch the quotes and the commas. You are not entering the groups of periods.

|  |
| --- |
| Seneca> use seneca  Seneca> db.student.insertOne({first\_name: "Sara", last\_name: "Stone", e\_mail: "stone@email.com", city: "Toronto", status: "full-time", gpa: 2.4, program: "CPD"})  {  acknowledged: **true**,  insertedId: ObjectId(**'6740b332df2c309e2d0d8190'**)  } |

**FIND COMMAND**

2 Write a command to check if the document has been created successfully.

You use *find()* method to search and fetch documents.

This is an EXAMPLE of the command.

* **db.student.find()**

|  |
| --- |
| Seneca> db.student.find()  [  {  \_id: ObjectId(**'6740b521df2c309e2d0d8191'**),  first\_name: **'Sara'**,  last\_name: **'Stone'**,  email: **'stone@email.com'**,  city: **'Toronto'**,  status: **'full-time'**,  gpa: **2.4**,  program: **'CPD'**  }  ] |

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

* **db.student.find().forEach(printjson)** – note the different layout

|  |
| --- |
| Seneca> db.student.find().forEach(printjson)  {  \_id: ObjectId('6740b521df2c309e2d0d8191'),  first\_name: **'Sara'**,  last\_name: **'Stone'**,  email: **'stone@email.com'**,  city: **'Toronto'**,  status: **'full-time'**,  gpa: **2.4**,  program: **'CPD'**  } |

How many fields are in your document? \_\_\_\_\_**8**

Is there any new field added to your document? \_\_\_\_\_\_**Yes**

If yes, what is the name of the field? \_\_\_\_\_ **\_id**

1. 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 “Sara 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 “Sara Stone” from the database in the box below.)

ASIDE: Helpful hint:

Consider the mycol collection has the following data.

{\_id : ObjectId("507f191e810c19729de860e1"), title: "MongoDB Overview"},

{\_id : ObjectId("507f191e810c19729de860e2"), title: "NoSQL Overview"},

{\_id : ObjectId("507f191e810c19729de860e3"), title: "Tutorials Point Overview"}

Following example will remove all the documents whose title is 'MongoDB Overview'.

>db.mycol.remove({'title':'MongoDB Overview'})

WriteResult({"nRemoved" : 1})

> db.mycol.find()

{"\_id" : ObjectId("507f191e810c19729de860e2"), "title" : "NoSQL Overview" }

{"\_id" : ObjectId("507f191e810c19729de860e3"), "title" : "Tutorials Point Overview" }

Here is another example code to delete document.

db.inventory.deleteOne( { status: "D" } )

You are to delete or remove Stone from your collection. Show the command **and the output from the command**

|  |
| --- |
| Seneca> db.student.find({**"first\_name" : "Sara", "last\_name" : "Stone"**})  [  {  \_id: ObjectId(**'6740d72cdf2c309e2d0d8192'**),  first\_name: **'Sara'**,  last\_name: **'Stone'**,  email: **'stone@email.com'**,  city: **'Toronto'**,  status: **'full-time'**,  gpa: **2.4**,  program: **'CPD'**  }  ]  Seneca> db.student.deleteOne({**"first\_name" : "Sara", "last\_name" : "Stone"**}) { acknowledged: **true**, deletedCount: **1** } |

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" : "Sara", "last\_name" : "Stone"**}) |
|  |

1. 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.)

To add these students, we want to store these documents into a variable first.

Define a variable named *sarray* and add these two document to the variable.

(You are storing more than one document, so you need to define an array in box further down)

\_id: 1001

first\_name: Sara  
last\_name: Stone  
email: [stone@email.com](mailto:stone@email.com)  
city: Toronto  
status: full-time  
gpa: 2.6  
program: CPD

\_id: 1002

first\_name: Jack  
last\_name: Adam  
email: [j\_adam@email.com](mailto:j_adam@email.com)  
city: North York  
status: part-time  
gpa: 3.6  
program: CPA

|  |
| --- |
| Seneca> sarray = [{\_id: 1001, first\_name: "Sara", last\_name: "Stone", email: "stone@email.com", city: "Toronto", status: "full-time", gpa: 2.6, program: "CPD"},{\_id: 1002, first\_name: "Jack", last\_name: "Adam", email: "j\_adam@email.com", city: "North York", status: "part-time", gpa: 3.6, program: "CPA"}]  [  {  \_id: 1001,  first\_name: "Sara",  last\_name: "Stone",  email: "stone@email.com",  city: "Toronto",  status: "full-time",  gpa: 2.6,  program: "CPD"  },  {  \_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 *sarray* array to insert the documents to your collection *student*. Write your insert statement in the box bellow.

|  |
| --- |
| Seneca> db.student.insertMany(sarray) |

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

|  |
| --- |
| { acknowledged: true, insertedIds: { '0': 1001, '1': 1002 } } |

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

|  |
| --- |
| Seneca> db.student.find()  [  {  \_id: 1001,  first\_name: "Sara",  last\_name: "Stone",  email: "stone@email.com",  city: "Toronto",  status: "full-time",  gpa: 2.6,  program: "CPD"  },  {  \_id: 1002,  first\_name: "Jack",  last\_name: "Adam",  email: "j\_adam@email.com",  city: "North York",  status: "part-time",  gpa: 3.6,  program: "CPA"  }  ] |

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

|  |
| --- |
| Seneca> db.student.deleteMany({})  { acknowledged: true, deletedCount: 2 } |

1. 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> db.getName()  Seneca  Seneca> db.dropDatabase() |

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

|  |
| --- |
| { ok: 1, dropped: 'Seneca' } |