# Lab 7 – 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\4.2\**bin**

and execute ***mongo***

> mongo

## Submission

You will be submitting your answers through a quiz created for you in Blackboard. The quiz will be open for a series of days and you will have multiple attempts if you want to correct things later. Do the work on your own, save your code to a text file so you can submit later.

*MAKE SURE you format your code in blackboard for an easily read format. It is recommended you change the font to “Courier New” when submitting coded answers.*

## Tasks

1. In this question you create a new database named *senecaLab07* and a collection *student*. We store student data in this collection.

> use senecaLab07

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](mailto:s_stone@email.com)  
city: Markham  
status: full-time  
gpa: 3.2  
program: CPA

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

> db.student.find()

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

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

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? Name of the field is \_id

Write a command to remove the “Sarah Stone” document that you created in Question 1. So, make sure your command will remove “Sarah Stone” from your collection. For now, we can assume that we do not have duplicate full names in our database.

***Note:*** *To avoid making mistakes, you can first write a find command with the proper criteria to see if only 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.)*

Include both the command and the output!

db.Student.deleteOne({

    \_id: ObjectId("6376dd1fd726bdd98f1cff2c")

    });

{

  "acknowledged": true,

  "deletedCount": 1

}

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). Include both the statement and the output.
2. db.Student.find({
3. \_id: ObjectId("6376dd1fd726bdd98f1cff2c")
4. });

[]

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.)

\_id: 1001

first\_name: Sarah  
last\_name: Stone  
email: [s\_stone@email.com](mailto: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](mailto: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 documents to the variable. (You are storing more than one document, so you need to define an array.

In your answer include:

* the array definition

var *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"

  }

];

* the statement to insert the array into the collection

db.Students.insertMany(*starray*);

* the output message after executing the above statements

{

  "acknowledged": true,

  "insertedIds": {

    "0": "1001",

    "1": "1002"

  }

}

1. Write a statement that shows all documents inserted in your collection *student*: Include both the statement and the output.

db.Students.find();

[

  {

    "\_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"

  }

]

1. Write a statement to remove all documents in the *student* collection. Include both the statement and the output.

db.Students.deleteMany({});

{

  "acknowledged": true,

  "deletedCount": 2

}

1. Write a statement to drop the database *senecaLab07*. 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 *senecaLab07* database.   
   You can write the *use* statement before removing the database to make sure your current database is *senecaLab07*.

> 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 *senecaLab07*, write the use statement to switch to *senecaLab07* and then delete the database.

Include both commands used as will as the output received.

db.dropDatabase();

{

  "ok": 1,

  "dropped": "test"

}