# Lab 10 – Week 12 due by Saturday, April 17th by 7pm

# (MongoDB – Update) Change documents

# Subject of your mail must be like 311-Lab10 by Smith, John

## Objective

In this lab, students learn how to update documents in a MongoDB database.

**update():** This method updates one document by default. If you want to update all documents that match the criteria using this method, you need the **updateMany()**

update(<filter>,<update>)

The *filter* parameter specifies the criteria. For instance:

{“\_id”= 0}

{} for updating all documents

The *update* parameter specifies the changes that will be applied to a document.

**updateOne():** This method updates only the first document that matches the criteria.

updateOne(<filter>,<update>)

**updateMany():** This method updates all documents that match the criteria.

updateMany(<filter>,<update>)

## Getting Started

In this lab, you will use students.json dataset. Download students.json from Blackboard and store it in a folder named C:\data\DB

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

* cd C:\Program Files\MongoDB\Server\4.2\**bin**

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\4.2\**bin**

and execute ***mongo***

* mongo

You will now open a Third Command Window and go to the folder where **mongimport** utility is stored and run it like shown below:

* cd C:\Program Files\MongoDB\Tools\100\bin

Execute the following command:

* mongoimport --db college --collection students C:\data\DB\students.json

After executing the command, the data is imported to the *college* database. To make sure data is imported successfully, go to the MongoDB shell and execute the following command to see the imported documents:

* show dbs

You should see the database *college* added to the list of your databases. To see the all documents inside the database:

* use college
* db.students.find().forEach(printjson)

or

* db.students.find().pretty()

## Submission

You submit this file as an e-mail attachment with answers (in the provided space). Name the file L10\_LASTNAME.docx”.

## Tasks

1. Write an update statement to add new fields *program* and *term* to all documents in the *students* collection and set them to values “*CPA*” and *1*. How many modifications were made?

|  |
| --- |
| Code:  db.students.updateMany({},{"$set":{program:"CPA", term:1}})  Output:  For my case the modified count was 27. |

1. Write an update statement to modify the value of the *program* field to “*BTM”* for all documents in the *students* collection.

|  |
| --- |
| db.students.updateMany({},{"$set":{"program":"BTM"}})  Output:  { "acknowledged" : true, "matchedCount" : 27, "modifiedCount" : 27 } |

1. Write a query to show only the *program* field for the documents having field *name* equal to *Jonie Raby*.

How many documents are there with the value *Jonie Raby* for the *name* field?

There is only one field with the name Jonie Raby with the program name BTM.

|  |
| --- |
| db.students.find({"name":"Jonie Raby"},{"program":1 , "\_id":0}) |

1. Write an update statement to modify the value of the program field to “*CPA”* for the student named *Jonie Raby*. Verify your modification by repeating Query 3.

|  |
| --- |
| Updating:  db.students.update({"name":"Jonie Raby"},{"$set":{"program":"CPA"}})  finding:  db.students.find({"name":"Jonie Raby"},{"program":1 , "\_id":0})  Output of the find command:  { "program" : "CPA" } |

How many documents were updated? 1 documents was updated

1. Write a query to show only the *term* field for documents with *\_id* value 20 or 22 or 24. Do not exclude *\_id* from the output.

|  |
| --- |
| Code:  db.students.find({"\_id": {"$in":[20,22,24]}}, {"term":1,"\_id":1})  Output:  { "\_id" : 20, "term" : "1" }  { "\_id" : 22, "term" : "1" }  { "\_id" : 24, "term" : "1" } |

1. Write an update statement to increase the value of the *term* field by 2 for documents with *\_id* value 20,22 and 24. Verify your modification by repeating Query 5.

|  |
| --- |
| db.students.updateMany({"\_id": {"$in":[20,22,24]}}, {"$inc":{"term":2}})  find command : Query 5  db.students.find({"\_id": {"$in":[20,22,24]}}, {"term":1,"\_id":1})  OUTPUT:  { "\_id" : 20, "term" : 3 }  { "\_id" : 22, "term" : 3 }  { "\_id" : 24, "term" : 3 } |

1. Write an update statement to remove the *term* field from documents with value of the *term* field equal to 3.

|  |
| --- |
| db.students.updateMany({"term":3}, {"$unset":{"term":""}})  output:  { "acknowledged" : true, "matchedCount" : 3, "modifiedCount" : 3 } |