# Lab 8

# (MongoDB – UPDATE and FIND)

## 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 option {multi:true}.

update(<filter>,<update>,<option>)

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 dataset.

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

Or you execute a batch file to start up MongoDB.

You will import students.json to the *college* database. To import data, go to the *bin* directory:

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

**Execute the following command to create “student” collection under “college” database:**

> use college

> db.student.insertOne({"\_id":0,"name":"aimee Zank","scores":[{"score":1.463179736705023,"type":"exam"},{"score":11.78273309957772,"type":"quiz"},{"score":35.8740349954354,"type":"homework"}]})

> db.student.insertOne({"\_id":1,"name":"Aurelia Menendez","scores":[{"score":60.06045071030959,"type":"exam"},{"score":52.79790691903873,"type":"quiz"},{"score":71.76133439165544,"type":"homework"}]})

> db.student.insertOne({"\_id":2,"name":"Corliss Zuk","scores":[{"score":67.03077096065002,"type":"exam"},{"score":6.301851677835235,"type":"quiz"},{"score":66.28344683278382,"type":"homework"}]})

> db.student.insertOne({"\_id":3,"name":"Bao Ziglar","scores":[{"score":71.64343899778332,"type":"exam"},{"score":24.80221293650313,"type":"quiz"},{"score":42.26147058804812,"type":"homework"}]})

> db.student.insertOne({"\_id":4,"name":"Zachary Langlais","scores":[{"score":78.68385091304332,"type":"exam"},{"score":90.2963101368042,"type":"quiz"},{"score":34.41620148042529,"type":"homework"}]})

show dbs

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

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

or

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

## Submission

You submit this file with answers (in the provided space). Name the file L09\_ID#\_LASTNAME.docx”.

## Tasks

Use the above created student database to answer the following questions:

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*.

|  |
| --- |
| db.student.updateMany( {}, { $set : {program:"CPA",item:1} } ) |

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

|  |
| --- |
| db.student.updateMany( {}, { $set :{program:"BTM"} }) |

1. Write an update statement to modify the value of the program field to “*CPA”* for the student named *Bao Ziglar*.

Before executing an update statement or a delete statement, you can use the *find()* method with the update or delete criteria, to see how many documents will be affected.

Write the update statement in the box below.

|  |
| --- |
| db.student.updateMany( {"name":{$eq:"Bao Ziglar"}}, { $set :{program:"CPA"} } ) |

How many documents are there with the value *Bao Ziglar* for the *name* field? \_1\_\_\_\_\_\_

How many documents were updated? \_1\_\_\_\_\_\_

1. Write a query to show only the *program* field for the document that the value of the field *name* is *Bao Ziglar.*

|  |
| --- |
| db.student.find({"name":"Bao Ziglar"},{\_id:0,"program":1}).pretty() |

1. Write an update statement to make the value of the *term* field by 2 for documents with *\_id* 0, 2, and 4.

|  |
| --- |
| db.student.updateMany( {\_id:{"$in":[0, 2, 4]}}, { $set : {item:2} } ) |

1. Write an update statement to remove the *term* field from documents that the value of the *term* filed is 3. Hint: to remove a field “field” {$unset: {field : 1}})

|  |
| --- |
| db.student.updateMany( {item:3}, { $unset : {item:1 }} ) |

1. Write a query to return *name* and *is* for students with score in the exam is between 12 and 30 (Values *12* and *30* are included).

|  |
| --- |
|  |

1. Write a query to return *id*, *name*, *score and test type* for students that are not of type *exam*.

|  |
| --- |
|  |