# Lab 8 – Week 10

# (MongoDB – Query)

## Objective

In this Lab, you learn to query a database in MongoDB.

## Getting Started

In this lab, you will use products.json dataset. Download products.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 products.json to the *inventory* database. To import data, go to the *bin* directory:

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

Execute the following command:

* mongoimport --db college --collection students --file ..\dataset\students.json

For the *json* file, provide the full path to the 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 documents inside the database:

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

## Submission

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

## Tasks

1. Write a query to return *name* and *price* of each product in the *inventory* database.

|  |
| --- |
| db.inventory.find({},{“name”:1,”price”:1}) |

1. Write a query to return *name* and *price* for products of type *accessory* in the *inventory* database.

|  |
| --- |
| db.inventory.aggregate({$match: {“type”:”accessory”}}) |

1. Write a query to return *name* and *price* for products with price between $12 and $20 (Values *12* and *20* are included).

|  |
| --- |
| db.products.find({“price”:{$gte:12, $lte:20}}) |

1. Write a query to return *id*, *name*, *price*, and *type* for products that are not of type *accessory*.

|  |
| --- |
| db.products.find({“id”:1, “name”:1, “price”:1, {$not:{“type”: “accessory”}}}) |

1. Write a query to return *id*, *name*, *price*, and type for products with type *accessory* or *service*.

|  |
| --- |
| db.products.aggregate({“\_id”:1, “name”:1, “price”:1},{$or: [{“type”: “accessory”},{“type: “service”}]}) |

1. Write a query to return *id*, *name*, *price*, and *type* for products that do have the *type* key.

|  |
| --- |
| db.products.find({“\_id”: 1, “name”:1, “price” :1, “type”: 1}) |

1. Write a query to return *id*, *name*, *price*, and *type* for products that their type is both *accessory* and *case*.

|  |
| --- |
| db.products.aggregate({“\_id”:1, “name”:1, “price”:1},{$and: [{“type”: “accessory”},{“type: “case”}]}) |