# Lab9 Mongo query questions

Clear screenshots of successful run of query statements and output is required in a single file.

Zero will be assigned otherwise. You may use mongosh from compass for the screenshots.

Create inventory database and upload the products data from products.json file.

Answer the following in command line and take screenshots.

## 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 server, 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 (if mongo.exe is missing run mongosh to connect)

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 inventory --collection products --file ..\dataset\products.json

(if mongoimport is missing, do some research to find a way around. You may use the compass GUI to upload. Or you may do bulk insert, the skills learnt in the previous lab)

To import the *json* file, provide the full path to the products.json. After executing the command, the data is imported to the *inventory* 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 inventory added to the list of your databases. To see the documents inside the database:

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

or

* db.products.find().pretty()

Extra questions

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

|  |
| --- |
| db.products.find({}, { name: 1, price: 1, \_id: 0 }) |

1. Write a query to return *name* and *price* for products of type “Health Care” in the *inventory* database.

|  |
| --- |
| db.products.find({ type: "Health Care" },{ name: 1, price: 1, \_id: 0 } ) |

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 } }, { name: 1, price: 1, \_id: 0 }) |

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

|  |
| --- |
| db.products.find( { type: { $ne: "Health Care" } }, { name: 1, price: 1, type: 1 }) |

1. Write a query to return *id*, *name*, *price*, and type for products with type “Health Care” or “Consumer Services.”

|  |
| --- |
| db.products.find( { type: { $in: ["Health Care", "Consumer Services"] } }, { name: 1, price: 1, type: 1 }) |

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

|  |
| --- |
| db.products.find( { type: { $exists: true } }, { name: 1, price: 1, type: 1 }) |

1. Write a query to return *id*, *name*, *price*, and *type* for products that their type is both Health Care and Consumer Services.

|  |
| --- |
| db.products.find( { type: { $all: ["Health Care", "Consumer Services"] } }, { name: 1, price: 1, type: 1 }) |