

## Lab 8 – Week 10 (Nasim)

### (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 inventory --collection products --file ../dataset/products.json`

For 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)

## Submission

You submit this file with answers (in the provided space). Name the file L08\_ID#\_LASTNAME.docx”.

## Tasks

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

```
db.products.find({}, {name:1, price:1});

{ "_id" : "ac3", "name" : "AC3 Phone", "price" : 200 }
{ "_id" : "ac7", "name" : "AC7 Phone", "price" : 320 }
{ "_id" : ObjectId("507d95d5719dbef170f15bf9"), "name" : "AC3 Series Charger", "price" : 19 }
{ "_id" : ObjectId("507d95d5719dbef170f15bfa"), "name" : "AC3 Case Green", "price" : 12 }
{ "_id" : ObjectId("507d95d5719dbef170f15bfb"), "name" : "Phone Extended Warranty", "price" : 38 }
{ "_id" : ObjectId("507d95d5719dbef170f15bfc"), "name" : "AC3 Case Black", "price" : 12.5 }
{ "_id" : ObjectId("507d95d5719dbef170f15bfd"), "name" : "AC3 Case Red", "price" : 12 }
{ "_id" : ObjectId("507d95d5719dbef170f15bfe"), "name" : "Phone Service Basic Plan" }
{ "_id" : ObjectId("507d95d5719dbef170f15bff"), "name" : "Phone Service Core Plan" }
{ "_id" : ObjectId("507d95d5719dbef170f15c00"), "name" : "Phone Service Family Plan" }
{ "_id" : ObjectId("507d95d5719dbef170f15c01"), "name" : "Cable TV Basic Service Package" }
```

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

```
db.products.find({type: "accessory"}, {name:1, price:1});

{ "_id" : ObjectId("507d95d5719dbef170f15bf9"), "name" : "AC3 Series Charger", "price" : 19 }
{ "_id" : ObjectId("507d95d5719dbef170f15bfa"), "name" : "AC3 Case Green", "price" : 12 }
{ "_id" : ObjectId("507d95d5719dbef170f15bfc"), "name" : "AC3 Case Black", "price" : 12.5 }
{ "_id" : ObjectId("507d95d5719dbef170f15bfd"), "name" : "AC3 Case Red", "price" : 12 }
```

3. 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" : ObjectId("507d95d5719dbef170f15bf9"), "name" : "AC3 Series Charger",
  "price" : 19 }
{ "_id" : ObjectId("507d95d5719dbef170f15bfa"), "name" : "AC3 Case Green", "price" :
  12 }
{ "_id" : ObjectId("507d95d5719dbef170f15bfc"), "name" : "AC3 Case Black", "price" :
  12.5 }
{ "_id" : ObjectId("507d95d5719dbef170f15bfd"), "name" : "AC3 Case Red", "price" :
  12 }
```

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

```
db.products.find({type: {$ne: "accessory"}}, {name:1, price:1, type:1});

{ "_id" : "ac3", "name" : "AC3 Phone", "type" : "phone", "price" : 200 }
{ "_id" : "ac7", "name" : "AC7 Phone", "type" : "phone", "price" : 320 }
{ "_id" : ObjectId("507d95d5719dbef170f15bfb"), "name" : "Phone Extended Warranty",
  "type" : "warranty", "price" : 38 }
{ "_id" : ObjectId("507d95d5719dbef170f15bfe"), "name" : "Phone Service Basic Plan",
  "type" : "service" }
{ "_id" : ObjectId("507d95d5719dbef170f15bff"), "name" : "Phone Service Core Plan",
  "type" : "service" }
{ "_id" : ObjectId("507d95d5719dbef170f15c00"), "name" : "Phone Service Family
Plan", "type" : "service" }
{ "_id" : ObjectId("507d95d5719dbef170f15c01"), "name" : "Cable TV Basic Service
Package", "type" : "tv" }
```

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

```
db.products.find({type: {$in: ["accessory", "service"]}}, {name:1, price:1,
type:1});

{ "_id" : ObjectId("507d95d5719dbef170f15bf9"), "name" : "AC3 Series Charger",
  "type" : [ "accessory", "charger" ], "price" : 19 }
{ "_id" : ObjectId("507d95d5719dbef170f15bfa"), "name" : "AC3 Case Green", "type" :
  [ "accessory", "case" ], "price" : 12 }
```

```
{ "_id" : ObjectId("507d95d5719dbef170f15bfc"), "name" : "AC3 Case Black", "type" :
[ "accessory", "case" ], "price" : 12.5 }
{ "_id" : ObjectId("507d95d5719dbef170f15bfd"), "name" : "AC3 Case Red", "type" :
[ "accessory", "case" ], "price" : 12 }
{ "_id" : ObjectId("507d95d5719dbef170f15bfe"), "name" : "Phone Service Basic Plan",
"type" : "service" }
{ "_id" : ObjectId("507d95d5719dbef170f15bff"), "name" : "Phone Service Core Plan",
"type" : "service" }
{ "_id" : ObjectId("507d95d5719dbef170f15c00"), "name" : "Phone Service Family
Plan", "type" : "service" }
```

6. 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});

{ "_id" : "ac3", "name" : "AC3 Phone", "type" : "phone", "price" : 200 }
{ "_id" : "ac7", "name" : "AC7 Phone", "type" : "phone", "price" : 320 }
{ "_id" : ObjectId("507d95d5719dbef170f15bf9"), "name" : "AC3 Series Charger",
"type" : [ "accessory", "charger" ], "price" : 19 }
{ "_id" : ObjectId("507d95d5719dbef170f15bfa"), "name" : "AC3 Case Green", "type" :
[ "accessory", "case" ], "price" : 12 }
{ "_id" : ObjectId("507d95d5719dbef170f15bfb"), "name" : "Phone Extended Warranty",
"type" : "warranty", "price" : 38 }
{ "_id" : ObjectId("507d95d5719dbef170f15bfc"), "name" : "AC3 Case Black", "type" :
[ "accessory", "case" ], "price" : 12.5 }
{ "_id" : ObjectId("507d95d5719dbef170f15bfd"), "name" : "AC3 Case Red", "type" :
[ "accessory", "case" ], "price" : 12 }
{ "_id" : ObjectId("507d95d5719dbef170f15bfe"), "name" : "Phone Service Basic Plan",
"type" : "service" }
{ "_id" : ObjectId("507d95d5719dbef170f15bff"), "name" : "Phone Service Core Plan",
"type" : "service" }
{ "_id" : ObjectId("507d95d5719dbef170f15c00"), "name" : "Phone Service Family
Plan", "type" : "service" }
{ "_id" : ObjectId("507d95d5719dbef170f15c01"), "name" : "Cable TV Basic Service
Package", "type" : "tv" }
```

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

```
db.products.find({type: {$all: ["accessory", "case"]}}, {name:1, price:1, type:1});

{ "_id" : ObjectId("507d95d5719dbef170f15bfa"), "name" : "AC3 Case Green", "type" :
[ "accessory", "case" ], "price" : 12 }
{ "_id" : ObjectId("507d95d5719dbef170f15bfc"), "name" : "AC3 Case Black", "type" :
```

```
[ "accessory", "case" ], "price" : 12.5 }  
{ "_id" : ObjectId("507d95d5719dbef170f15bfd"), "name" : "AC3 Case Red", "type" :  
[ "accessory", "case" ], "price" : 12 }
```

This study resource was  
shared via CourseHero.com