# Lab 8 – Week 11

# (MongoDB – Query)

## Objective

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

## Getting Started

Import a JSON file

In this lab, you will use products.json dataset.

Download products.json from Blackboard and store it in a folder named dataset.

products.json

Run the mongodb server:

> mongod

When MongoDB starts successfully, open another Windows command prompt and execute mongosh

> mongosh

In this lab we import data from the products.json file.

To import data in MongoDB, you need to download the MongoDB Tools.

To download the file, go to the following link. Under Tools, find the option MongoDB Database Tools and click on it.

<https://www.mongodb.com/try/download/database-tools>

Download the current version compatible to your platform.

When the file is downloaded, extract the file.

You can rename the folder to mongodb-database-tools and copy it to the C directory:

C:\mongodb-database-tools

Now, add the following path to the environment system variable path so you can run mongoimport anywhere from the command prompt.

C:\mongodb-database-tools\bin

You will import products.json to the inventory database.

Execute the following command in the command prompt:

> mongoimport --db inventory --collection products --file ..\dataset\products.json

Provide the full path to locate the products.json file.

For instance: > mongoimport --db inventory --collection products --file C:\dataset\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.pdf”.

## Tasks

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

|  |
| --- |
| inventory> db.products.find({}, { name: 1, price: 1, \_id: 0 }) [  { name: 'AC7 Phone', price: 320 },  { name: 'AC3 Series Charger', price: 19 },  { name: 'AC3 Case Green', price: 12 },  { name: 'Phone Service Basic Plan' },  { name: 'Phone Service Core Plan' },  { name: 'AC3 Case Red', price: 12 },  { name: 'Phone Service Family Plan' },  { name: 'AC3 Phone', price: 200 },  { name: 'Phone Extended Warranty', price: 38 },  { name: 'Cable TV Basic Service Package' },  { name: 'AC3 Case Black', price: 12.5 }  ]  inventory> |

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

|  |
| --- |
| inventory> db.products.find({ type: "accessory" }, { name: 1, price: 1, \_id: 0 })  [  { name: 'AC3 Series Charger', price: 19 },  { name: 'AC3 Case Green', price: 12 },  { name: 'AC3 Case Red', price: 12 },  { name: 'AC3 Case Black', price: 12.5 }  ]  inventory> |

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

|  |
| --- |
| inventory> db.products.find({ price: { $gte: 12, $lte: 20 } }, { name: 1, price: 1, \_id: 0 })  [  { name: 'AC3 Series Charger', price: 19 },  { name: 'AC3 Case Green', price: 12 },  { name: 'AC3 Case Red', price: 12 },  { name: 'AC3 Case Black', price: 12.5 }  ] |

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

|  |
| --- |
| inventory> db.products.find({ type: { $ne: "accessory" } }, { \_id: 1, name: 1, price: 1, type: 1 })  [  { \_id: 'ac7', name: 'AC7 Phone', type: 'phone', price: 320 },  {  \_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: 'ac3', name: 'AC3 Phone', type: 'phone', price: 200 },  {  \_id: ObjectId('507d95d5719dbef170f15bfb'),  name: 'Phone Extended Warranty',  type: 'warranty',  price: 38  },  {  \_id: ObjectId('507d95d5719dbef170f15c01'),  name: 'Cable TV Basic Service Package',  type: 'tv'  }  ]  inventory> |

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

|  |
| --- |
| inventory> db.products.find({ type: { $in: ["accessory", "service"] } }, { \_id: 1, 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('507d95d5719dbef170f15bfe'),  name: 'Phone Service Basic Plan',  type: 'service'  },  {  \_id: ObjectId('507d95d5719dbef170f15bff'),  name: 'Phone Service Core Plan',  type: 'service'  },  {  \_id: ObjectId('507d95d5719dbef170f15bfd'),  name: 'AC3 Case Red',  type: [ 'accessory', 'case' ],  price: 12  },  {  \_id: ObjectId('507d95d5719dbef170f15c00'),  name: 'Phone Service Family Plan',  type: 'service'  },  {  \_id: ObjectId('507d95d5719dbef170f15bfc'),  name: 'AC3 Case Black',  type: [ 'accessory', 'case' ],  price: 12.5  }  ]  db.products.find({ $or: [{ type: "accessory" }, { type: "service" }] },{ \_id: 1, name: 1, price: 1, type: 1 });  inventory> db.products.find({ $or: [{ type: "accessory" }, { type: "service" }] },{ \_id: 1, 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('507d95d5719dbef170f15bfe'),  name: 'Phone Service Basic Plan',  type: 'service'  },  {  \_id: ObjectId('507d95d5719dbef170f15bff'),  name: 'Phone Service Core Plan',  type: 'service'  },  {  \_id: ObjectId('507d95d5719dbef170f15bfd'),  name: 'AC3 Case Red',  type: [ 'accessory', 'case' ],  price: 12  },  {  \_id: ObjectId('507d95d5719dbef170f15c00'),  name: 'Phone Service Family Plan',  type: 'service'  },  {  \_id: ObjectId('507d95d5719dbef170f15bfc'),  name: 'AC3 Case Black',  type: [ 'accessory', 'case' ],  price: 12.5  }  ] |

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

|  |
| --- |
| inventory> db.products.find({ type: { $exists: true } }, { \_id: 1, name: 1, price: 1, type: 1 })  [  { \_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('507d95d5719dbef170f15bfe'),  name: 'Phone Service Basic Plan',  type: 'service'  },  {  \_id: ObjectId('507d95d5719dbef170f15bff'),  name: 'Phone Service Core Plan',  type: 'service'  },  {  \_id: ObjectId('507d95d5719dbef170f15bfd'),  name: 'AC3 Case Red',  type: [ 'accessory', 'case' ],  price: 12  },  {  \_id: ObjectId('507d95d5719dbef170f15c00'),  name: 'Phone Service Family Plan',  type: 'service'  },  { \_id: 'ac3', name: 'AC3 Phone', type: 'phone', price: 200 },  {  \_id: ObjectId('507d95d5719dbef170f15bfb'),  name: 'Phone Extended Warranty',  type: 'warranty',  price: 38  },  {  \_id: ObjectId('507d95d5719dbef170f15c01'),  name: 'Cable TV Basic Service Package',  type: 'tv'  },  {  \_id: ObjectId('507d95d5719dbef170f15bfc'),  name: 'AC3 Case Black',  type: [ 'accessory', 'case' ],  price: 12.5  } |

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

|  |
| --- |
| inventory> db.products.find({ type: { $all: ["accessory", "case"] } }, { \_id: 1, name: 1, price: 1, type: 1 })  [  {  \_id: ObjectId('507d95d5719dbef170f15bfa'),  name: 'AC3 Case Green',  type: [ 'accessory', 'case' ],  price: 12  },  {  \_id: ObjectId('507d95d5719dbef170f15bfd'),  name: 'AC3 Case Red',  type: [ 'accessory', 'case' ],  price: 12  },  {  \_id: ObjectId('507d95d5719dbef170f15bfc'),  name: 'AC3 Case Black',  type: [ 'accessory', 'case' ],  price: 12.5  }  ] |

Good luck.