

Lab 08

Name : Dev Soni

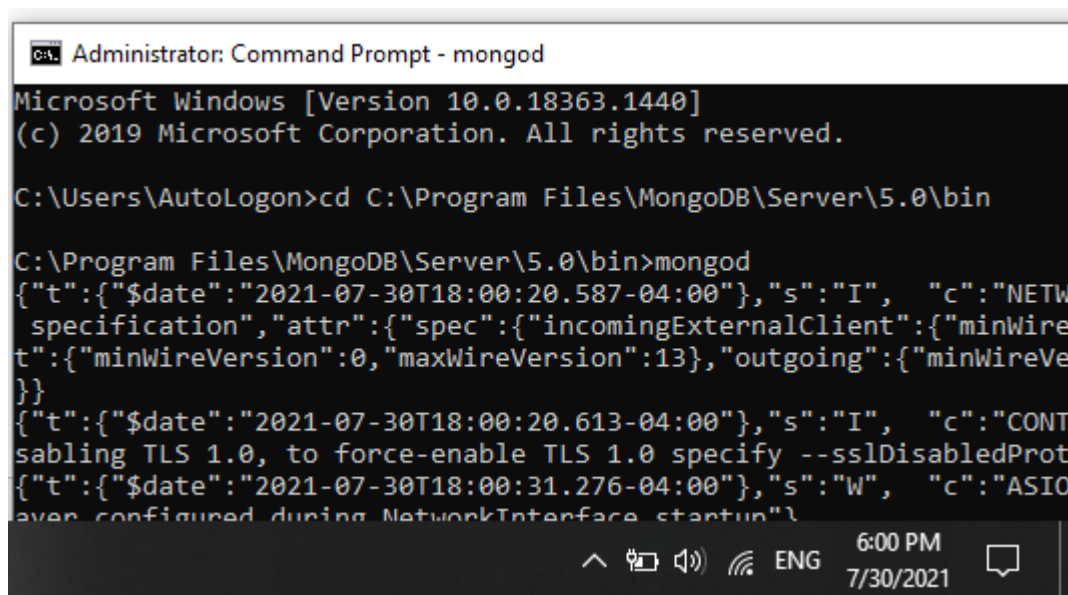
Seneca ID :130759210

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`



```
Administrator: Command Prompt - mongod
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\AutoLogon>cd C:\Program Files\MongoDB\Server\5.0\bin

C:\Program Files\MongoDB\Server\5.0\bin>mongod
{"t":{"$date":"2021-07-30T18:00:20.587-04:00"},"s":"I",  "c":"NETW
specification", "attr":{"spec":{"incomingExternalClient":{"minWire
t":{"minWireVersion":0,"maxWireVersion":13},"outgoing":{"minWireVe
}}}
{"t":{"$date":"2021-07-30T18:00:20.613-04:00"},"s":"I",  "c":"CONT
sabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProt
{"t":{"$date":"2021-07-30T18:00:31.276-04:00"},"s":"W",  "c":"ASIO
even configured during NetworkInterface startup"}
^  [network icon] [volume icon] [wifi icon] ENG 6:00 PM
7/30/2021
```

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`

```
Administrator: Command Prompt - mongo
Microsoft Windows [Version 10.0.18363.1440]
(c) 2019 Microsoft Corporation. All rights reserved.

C:\Users\AutoLogon>

C:\Users\AutoLogon>cd C:\Program Files\MongoDB\Server\5.0\bin

C:\Program Files\MongoDB\Server\5.0\bin>mongo
MongoDB shell version v5.0.0
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&g
Implicit session: session { "id" : UUID("0c5a00b-b424-432e-a700...
```

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`

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()`

```
>
>
> show dbs
Seneca      0.000GB
admin       0.000GB
config      0.000GB
inventory   0.000GB
library     0.000GB
local       0.000GB
>
>
> use inventory
switched to db inventory
> db.products.find().pretty()
{
  "_id" : 1,
  "name" : "Skewers - Bamboo",
  "price" : 14,
  "type" : [
    "Consumer Services",
    "Health Care"
  ]
}
{ "_id" : 2, "name" : "Beer - Muskoka Cream Ale", "price" : 21 }
{ "_id" : 3, "name" : "Wine - Red, Gallo, Merlot", "price" : 26 }
{
  "_id" : 4,
  "name" : "Magnotta - Bel Paese White",
  "price" : 23,
  "type" : "Consumer Services"
}
{ "_id" : 5, "name" : "Tuna - Canned, Flaked, Light", "price" : 25 }
{
  "_id" : 6,
  "name" : "Nut - Pumpkin Seeds",
  "price" : 12
}
```

Extra questions

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

```
db.products.find({},{"_id":0,"name":1,"price":1})
```

```
> db.products.find({}, {"_id":0,"name":1,"price":1})
{ "name" : "Skewers - Bamboo", "price" : 14 }
{ "name" : "Beer - Muskoka Cream Ale", "price" : 21 }
{ "name" : "Wine - Red, Gallo, Merlot", "price" : 26 }
{ "name" : "Magnotta - Bel Paese White", "price" : 23 }
{ "name" : "Tuna - Canned, Flaked, Light", "price" : 25 }
{ "name" : "Nut - Pumpkin Seeds", "price" : 27 }
{ "name" : "Red Currants", "price" : 5 }
{ "name" : "Sprouts - Peppercreess", "price" : 2 }
{ "name" : "Lemonade - Strawberry, 591 Ml", "price" : 2 }
{ "name" : "Bread - Dark Rye, Loaf", "price" : 15 }
{ "name" : "Pancetta", "price" : 22 }
{ "name" : "Bagel - Plain", "price" : 13 }
{ "name" : "Nori Sea Weed - Gold Label", "price" : 25 }
{ "name" : "Lime Cordial - Roses", "price" : 12 }
{ "name" : "Cookie - Oatmeal", "price" : 21 }
{ "name" : "Wine - Prosecco Valdobienne", "price" : 4 }
{ "name" : "Dr. Pepper - 355ml", "price" : 19 }
{ "name" : "Compound - Mocha", "price" : 25 }
{ "name" : "Beans - Wax", "price" : 6 }
{ "name" : "Soup - French Onion", "price" : 12 }
Type "it" for more
> _
```

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

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

```
db.products.find({"type":"Health Care"}, {"_id":0,"name":1,"price":1})
{ "name" : "Skewers - Bamboo", "price" : 14 }
{ "name" : "Bread - Dark Rye, Loaf", "price" : 15 }
{ "name" : "Wine - Prosecco Valdobienne", "price" : 4 }
{ "name" : "Soup - French Onion", "price" : 12 }
{ "name" : "Beer - Moosehead", "price" : 5 }
{ "name" : "Squeeze Bottle", "price" : 5 }
```

- 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}},{"_id":0,"name":2,"price":2})
```

```
>  
> db.products.find({"price": {"$gte":12,"$lte":20}},{"_id":0,"name":2,"price":2})  
{ "name" : "Skewers - Bamboo", "price" : 14 }  
{ "name" : "Bread - Dark Rye, Loaf", "price" : 15 }  
{ "name" : "Bagel - Plain", "price" : 13 }  
{ "name" : "Lime Cordial - Roses", "price" : 12 }  
{ "name" : "Dr. Pepper - 355ml", "price" : 19 }  
{ "name" : "Soup - French Onion", "price" : 12 }  
{ "name" : "Ketchup - Tomato", "price" : 17 }  
{ "name" : "Chocolate - Milk", "price" : 15 }  
{ "name" : "Wine - Alsace Riesling Reserve", "price" : 18 }
```

4. 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"}},{"_id":1,"name":1,"price":1,"type":1})
```

```

> db.products.find({"type": {"$ne": "Health Care"}}, {"_id":1,"name":1,"price":1,"type":1})
{ "_id" : 2, "name" : "Beer - Muskoka Cream Ale", "price" : 21 }
{ "_id" : 3, "name" : "Wine - Red, Gallo, Merlot", "price" : 26 }
{ "_id" : 4, "name" : "Magnotta - Bel Paese White", "price" : 23, "type" : "Consumer Serv
{ "_id" : 5, "name" : "Tuna - Canned, Flaked, Light", "price" : 25 }
{ "_id" : 6, "name" : "Nut - Pumpkin Seeds", "price" : 27, "type" : "Consumer Services" }
{ "_id" : 7, "name" : "Red Currants", "price" : 5, "type" : "Public Utilities" }
{ "_id" : 8, "name" : "Sprouts - Peppercreess", "price" : 2, "type" : "Technology" }
{ "_id" : 9, "name" : "Lemonade - Strawberry, 591 Ml", "price" : 2, "type" : "Finance" }
{ "_id" : 11, "name" : "Pancetta", "price" : 22, "type" : "Finance" }
{ "_id" : 12, "name" : "Bagel - Plain", "price" : 13 }
{ "_id" : 13, "name" : "Nori Sea Weed - Gold Label", "price" : 25 }
{ "_id" : 14, "name" : "Lime Cordial - Roses", "price" : 12, "type" : "Capital Goods" }
{ "_id" : 15, "name" : "Cookie - Oatmeal", "price" : 21, "type" : "Finance" }
{ "_id" : 17, "name" : "Dr. Pepper - 355ml", "price" : 19 }
{ "_id" : 18, "name" : "Compound - Mocha", "price" : 25 }
{ "_id" : 19, "name" : "Beans - Wax", "price" : 6, "type" : "Capital Goods" }
{ "_id" : 21, "name" : "Island Oasis - Strawberry", "price" : 5 }
{ "_id" : 22, "name" : "Wine - Chianti Classico Riserva", "price" : 28, "type" : "Consume
{ "_id" : 23, "name" : "Sprouts - Brussel", "price" : 3 }
{ "_id" : 24, "name" : "Hersey Shakes", "price" : 23, "type" : "n/a" }
Type "it" for more
>

```

- 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"]}}, {"_id":1,"name":1,"price":1,"type":1})
```

```
db.products.find({"$or": [{"type" : "Health Care"}, {"type" : "Consumer Services"}]}, {"_id":1,"name":1,"price":1,"type":1})
```

```
>
> db.products.find({"type": {"$in": ["Health Care", "Consumer Services"]}}, {"_id":1,
{ "_id" : 1, "name" : "Skewers - Bamboo", "price" : 14, "type" : [ "Consumer Service
{ "_id" : 4, "name" : "Magnotta - Bel Paese White", "price" : 23, "type" : "Consumer
{ "_id" : 6, "name" : "Nut - Pumpkin Seeds", "price" : 27, "type" : "Consumer Servic
{ "_id" : 10, "name" : "Bread - Dark Rye, Loaf", "price" : 15, "type" : "Health Care
{ "_id" : 16, "name" : "Wine - Prosecco Valdobienne", "price" : 4, "type" : "Health
{ "_id" : 20, "name" : "Soup - French Onion", "price" : 12, "type" : "Health Care" }
{ "_id" : 27, "name" : "Beer - Moosehead", "price" : 5, "type" : "Health Care" }
{ "_id" : 29, "name" : "Squeeze Bottle", "price" : 5, "type" : "Health Care" }
>
```



```
>
> db.products.find({"$or": [{"type" : "Health Care"}, {"type" : "Consumer Services"}]
{ "_id" : 1, "name" : "Skewers - Bamboo", "price" : 14, "type" : [ "Consumer Service
{ "_id" : 4, "name" : "Magnotta - Bel Paese White", "price" : 23, "type" : "Consumer
{ "_id" : 6, "name" : "Nut - Pumpkin Seeds", "price" : 27, "type" : "Consumer Servic
{ "_id" : 10, "name" : "Bread - Dark Rye, Loaf", "price" : 15, "type" : "Health Care
{ "_id" : 16, "name" : "Wine - Prosecco Valdobienne", "price" : 4, "type" : "Health
{ "_id" : 20, "name" : "Soup - French Onion", "price" : 12, "type" : "Health Care" }
{ "_id" : 27, "name" : "Beer - Moosehead", "price" : 5, "type" : "Health Care" }
{ "_id" : 29, "name" : "Squeeze Bottle", "price" : 5, "type" : "Health Care" }
>
```



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}}, {"_id":1,"name":1,"price":1,"type":1})
```

```

>
> db.products.find({"type": {"$exists": true}}, {"_id":1,"name":1,"price":1,"type":1})
{ "_id" : 1, "name" : "Skewers - Bamboo", "price" : 14, "type" : [ "Consumer Services" ] }
{ "_id" : 4, "name" : "Magnotta - Bel Paese White", "price" : 23, "type" : "Consumer Services" }
{ "_id" : 6, "name" : "Nut - Pumpkin Seeds", "price" : 27, "type" : "Consumer Services" }
{ "_id" : 7, "name" : "Red Currants", "price" : 5, "type" : "Public Utilities" }
{ "_id" : 8, "name" : "Sprouts - Peppercreess", "price" : 2, "type" : "Technology" }
{ "_id" : 9, "name" : "Lemonade - Strawberry, 591 Ml", "price" : 2, "type" : "Finance" }
{ "_id" : 10, "name" : "Bread - Dark Rye, Loaf", "price" : 15, "type" : "Health Care" }
{ "_id" : 11, "name" : "Pancetta", "price" : 22, "type" : "Finance" }
{ "_id" : 14, "name" : "Lime Cordial - Roses", "price" : 12, "type" : "Capital Goods" }
{ "_id" : 15, "name" : "Cookie - Oatmeal", "price" : 21, "type" : "Finance" }
{ "_id" : 16, "name" : "Wine - Prosecco Valdobienne", "price" : 4, "type" : "Health Care" }
{ "_id" : 19, "name" : "Beans - Wax", "price" : 6, "type" : "Capital Goods" }
{ "_id" : 20, "name" : "Soup - French Onion", "price" : 12, "type" : "Health Care" }
{ "_id" : 22, "name" : "Wine - Chianti Classico Riserva", "price" : 28, "type" : "Capital Goods" }
{ "_id" : 24, "name" : "Hersey Shakes", "price" : 23, "type" : "n/a" }
{ "_id" : 25, "name" : "Ketchup - Tomato", "price" : 17, "type" : "Basic Industries" }
{ "_id" : 26, "name" : "Chocolate - Milk", "price" : 15, "type" : "n/a" }
{ "_id" : 27, "name" : "Beer - Moosehead", "price" : 5, "type" : "Health Care" }
{ "_id" : 29, "name" : "Squeeze Bottle", "price" : 5, "type" : "Health Care" }
>

```

7. 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"]}}, {"_id":1,"name":1,"price":1,"type":1})
```

```

>
> db.products.find({"type": {"$all": ["Health Care", "Consumer Services"]}}, {"_id":1,"name":1,"price":1,"type":1})
{ "_id" : 1, "name" : "Skewers - Bamboo", "price" : 14, "type" : [ "Consumer Services" ] }
>

```