

# CS504 Project 2 Screenshots

Nebojsa Hrnjez

G01337837

```
Last login: Thu Oct 28 19:00:57 on ttys000
(base) nebojsahrnjez@Nebojsas-MBP ~ % mongod
[{"t": {"$date": "2021-11-03T15:47:29.377-04:00"}, "s": "I", "c": "NETWORK", "id": 4915701, "ctx": "-", "msg": "Initialized wire specification", "attr": {"spec": {"incomingExternalClient": {"minWireVersion": 0, "maxWireVersion": 13}, "incomingInternalClient": {"minWireVersion": 0, "maxWireVersion": 13}, "isInternalClient": true}}}
[{"t": {"$date": "2021-11-03T15:47:29.379-04:00"}, "s": "I", "c": "CONTROL", "id": 23285, "ctx": "-", "msg": "Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'"}
[{"t": {"$date": "2021-11-03T15:47:29.383-04:00"}, "s": "W", "c": "ASIO", "id": 22601, "ctx": "main", "msg": "No TransportLayer configured during NetworkInterface startup"}
[{"t": {"$date": "2021-11-03T15:47:29.383-04:00"}, "s": "I", "c": "NETWORK", "id": 4648602, "ctx": "main", "msg": "Implicit TCP FastOpen in use."}
[{"t": {"$date": "2021-11-03T15:47:29.384-04:00"}, "s": "W", "c": "ASIO", "id": 22601, "ctx": "main", "msg": "No TransportLayer configured during NetworkInterface startup"}
[{"t": {"$date": "2021-11-03T15:47:29.384-04:00"}, "s": "I", "c": "REPL", "id": 5123008, "ctx": "main", "msg": "Successfully registered PrimaryOnlyService", "attr": {"service": "TenantMigrationDonorService", "ns": "config.tenantMigrationDonors"}}
[{"t": {"$date": "2021-11-03T15:47:29.384-04:00"}, "s": "I", "c": "REPL", "id": 5123008, "ctx": "main", "msg": "Successfully registered PrimaryOnlyService", "attr": {"service": "TenantMigrationRecipientService", "ns": "config.tenantMigrationRecipients"}}
[{"t": {"$date": "2021-11-03T15:47:29.385-04:00"}, "s": "I", "c": "CONTROL", "id": 4615611, "ctx": "initandlisten", "msg": "MongoDB starting", "attr": {"pid": 54779, "port": 27017, "dbPath": "/data/db", "architecture": "64-bit", "host": "Nebojsas-MBP.eduromam.gmu.edu"}}
[{"t": {"$date": "2021-11-03T15:47:29.385-04:00"}, "s": "I", "c": "CONTROL", "id": 23403, "ctx": "initandlisten", "msg": "Build Info", "attr": {"buildInfo": {"version": "5.0.3", "gitVersion": "657fea5a61a74d7a79df7aff8e4bcf0bc742b748", "modules": [], "allocator": "system", "environment": {"distarch": "x86_64", "target_arch": "x86_64"}}}}
[{"t": {"$date": "2021-11-03T15:47:29.385-04:00"}, "s": "I", "c": "CONTROL", "id": 51765, "ctx": "initandlisten", "msg": "Operating System", "attr": {"os": {"name": "Mac OS X", "version": "20.6.0"}}}
[{"t": {"$date": "2021-11-03T15:47:29.385-04:00"}, "s": "I", "c": "CONTROL", "id": 21951, "ctx": "initandlisten", "msg": "Options set by command line", "attr": {"options": {}}}
[{"t": {"$date": "2021-11-03T15:47:29.384-04:00"}, "s": "E", "c": "CONTROL", "id": 20568, "ctx": "initandlisten", "msg": "Error setting up listener", "attr": {"error": {"code": 9001, "codeName": "SocketException", "errmsg": "Address already in use"}}, "err": "Address already in use"}
[{"t": {"$date": "2021-11-03T15:47:29.418-04:00"}, "s": "I", "c": "REPL", "id": 4784900, "ctx": "initandlisten", "msg": "Stepping down the ReplicationCoordinator for shutdown", "attr": {"waitForMillis": 15000}}
[{"t": {"$date": "2021-11-03T15:47:29.419-04:00"}, "s": "I", "c": "COMMAND", "id": 4784901, "ctx": "initandlisten", "msg": "Shutting down the MirrorMaestro"}, {"t": {"$date": "2021-11-03T15:47:29.419-04:00"}, "s": "I", "c": "SHARDING", "id": 4784902, "ctx": "initandlisten", "msg": "Shutting down the WaitForMajorityService"}, {"t": {"$date": "2021-11-03T15:47:29.419-04:00"}, "s": "I", "c": "NETWORK", "id": 4784905, "ctx": "initandlisten", "msg": "Shutting down the global connection pool"}, {"t": {"$date": "2021-11-03T15:47:29.419-04:00"}, "s": "I", "c": "NETWORK", "id": 4784918, "ctx": "initandlisten", "msg": "Shutting down the ReplicaSetMonitor"}, {"t": {"$date": "2021-11-03T15:47:29.419-04:00"}, "s": "I", "c": "SHARDING", "id": 4784921, "ctx": "initandlisten", "msg": "Shutting down the MigrationUtilExecutor"}, {"t": {"$date": "2021-11-03T15:47:29.419-04:00"}, "s": "I", "c": "ASIO", "id": 22582, "ctx": "MigrationUtil-TaskExecutor", "msg": "Killing all outstanding egress activity."}, {"t": {"$date": "2021-11-03T15:47:29.419-04:00"}, "s": "I", "c": "COMMAND", "id": 4784923, "ctx": "initandlisten", "msg": "Shutting down the ServiceEntryPoint"}, {"t": {"$date": "2021-11-03T15:47:29.419-04:00"}, "s": "I", "c": "CONTROL", "id": 4784925, "ctx": "initandlisten", "msg": "Shutting down free monitoring"}, {"t": {"$date": "2021-11-03T15:47:29.419-04:00"}, "s": "I", "c": "CONTROL", "id": 4784927, "ctx": "initandlisten", "msg": "Shutting down the HealthLog"}, {"t": {"$date": "2021-11-03T15:47:29.419-04:00"}, "s": "I", "c": "CONTROL", "id": 4784928, "ctx": "initandlisten", "msg": "Shutting down the TTL monitor"}, {"t": {"$date": "2021-11-03T15:47:29.419-04:00"}, "s": "I", "c": "CONTROL", "id": 4784929, "ctx": "initandlisten", "msg": "Acquiring the global lock for shutdown"}, {"t": {"$date": "2021-11-03T15:47:29.419-04:00"}, "s": "I", "c": "-", "id": 4784931, "ctx": "initandlisten", "msg": "Dropping the scope cache for shutdown"}, {"t": {"$date": "2021-11-03T15:47:29.419-04:00"}, "s": "I", "c": "FTDC", "id": 4784926, "ctx": "initandlisten", "msg": "Shutting down full-time data capture"}, {"t": {"$date": "2021-11-03T15:47:29.419-04:00"}, "s": "I", "c": "CONTROL", "id": 20565, "ctx": "initandlisten", "msg": "Now exiting"}, {"t": {"$date": "2021-11-03T15:47:29.419-04:00"}, "s": "I", "c": "CONTROL", "id": 23138, "ctx": "initandlisten", "msg": "Shutting down", "attr": {"exitCode": 48}}
(base) nebojsahrnjez@Nebojsas-MBP ~ % 
```

Opened terminal and typed the command “mongod”, this runs the background processes for “mongoDB”

```
Last login: Wed Nov  3 15:47:04 on ttys000
[(base) nebojsahrnjez@Nebojsas-MBP ~ % mongo
zsh: command not found: mongodb
[(base) nebojsahrnjez@Nebojsas-MBP ~ % mongo
MongoDB shell version v5.0.3
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("68cb810c-ef17-4014-bc38-6c64584ff1f4") }
MongoDB server version: 5.0.3
=====
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in
an upcoming release.
We recommend you begin using "mongosh".
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====

The server generated these startup warnings when booting:
  2021-10-21T11:26:03.035-04:00: Access control is not enabled for the database. Read and write access to data and config
uration is unrestricted
  2021-10-21T11:26:03.036-04:00: Soft rlimits for open file descriptors too low
  2021-10-21T11:26:03.036-04:00:           currentValue: 256
  2021-10-21T11:26:03.036-04:00:           recommendedMinimum: 64000
---
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

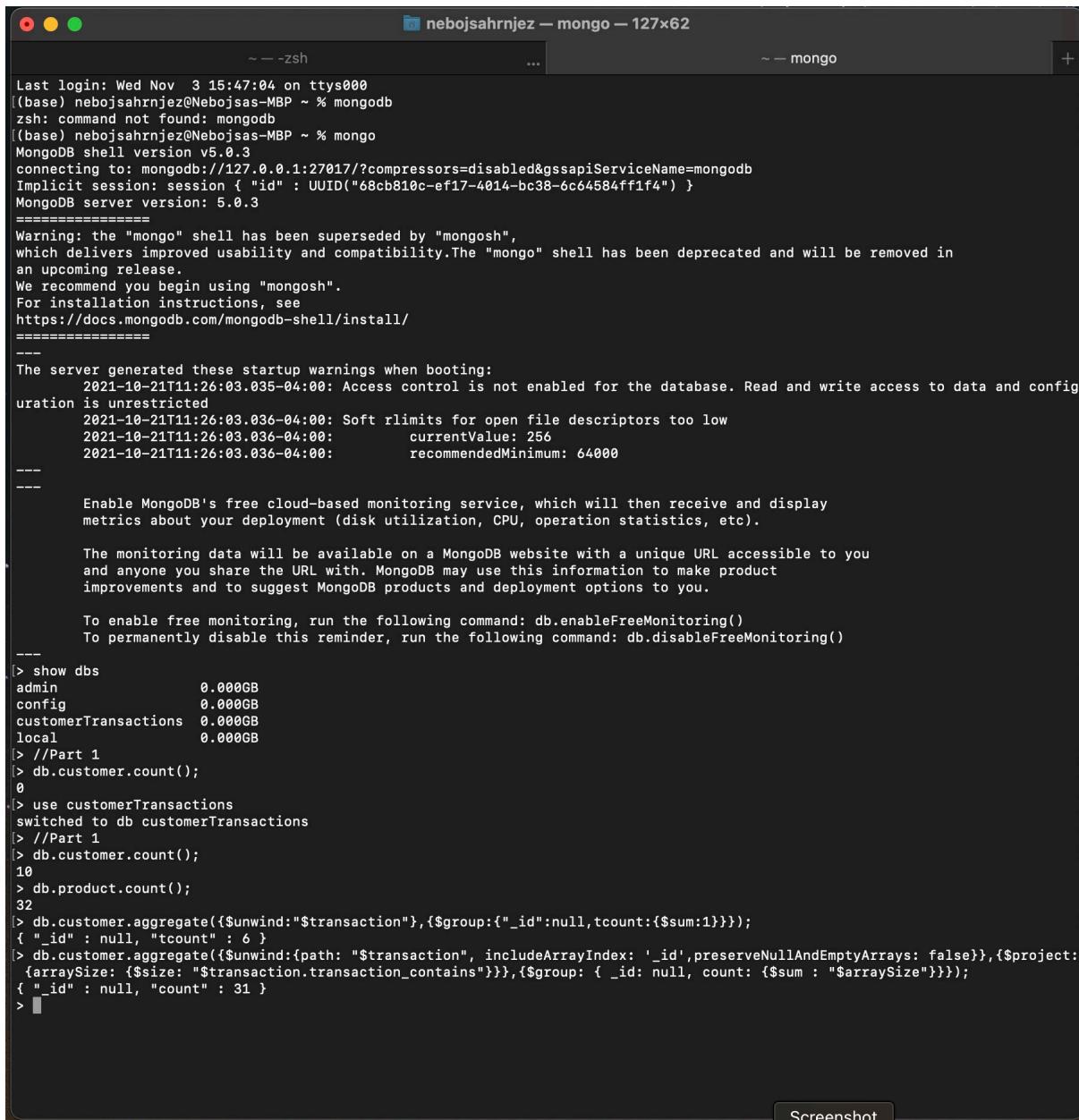
The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
> |
```

Opened a new terminal window and called the “mongo” command.

```
|---> show dbs
admin          0.000GB
config         0.000GB
customerTransactions 0.000GB
local          0.000GB
> |
```

Used the “show dbs” command to see what databases were present



```
Last login: Wed Nov  3 15:47:04 on ttys000
(base) nebojsahrnjez@Nebojsas-MBP ~ % mongod
zsh: command not found: mongod
(base) nebojsahrnjez@Nebojsas-MBP ~ % mongo
MongoDB shell version v5.0.3
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("68cb810c-ef17-4014-bc38-6c64584ff1f4") }
MongoDB server version: 5.0.3
=====
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in
an upcoming release.
We recommend you begin using "mongosh".
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====

The server generated these startup warnings when booting:
2021-10-21T11:26:03.035-04:00: Access control is not enabled for the database. Read and write access to data and config
uration is unrestricted
2021-10-21T11:26:03.036-04:00: Soft rlimits for open file descriptors too low
2021-10-21T11:26:03.036-04:00:           currentValue: 256
2021-10-21T11:26:03.036-04:00:           recommendedMinimum: 64000
---

Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()

> show dbs
admin          0.000GB
config         0.000GB
customerTransactions 0.000GB
local          0.000GB
> //Part 1
> db.customer.count();
0
.> use customerTransactions
switched to db customerTransactions
> //Part 1
> db.customer.count();
10
> db.product.count();
32
> db.customer.aggregate({$unwind:"$transaction"},{$group:{"_id":null,tcount:{$sum:1}}});
{ "_id" : null, "tcount" : 6 }
> db.customer.aggregate({$unwind:{path: "$transaction", includeArrayIndex: '_id',preserveNullAndEmptyArrays: false}},{$project:
{arraySize: {$size: "$transaction.transaction_contains"}},{$group: { _id: null, count: {$sum : "$arraySize"}}}});
{ "_id" : null, "count" : 31 }
```

Screenshot

Typed the command “use customerTransactions” to select the project 2 database.

Proceeded to complete part 1 of project 2:

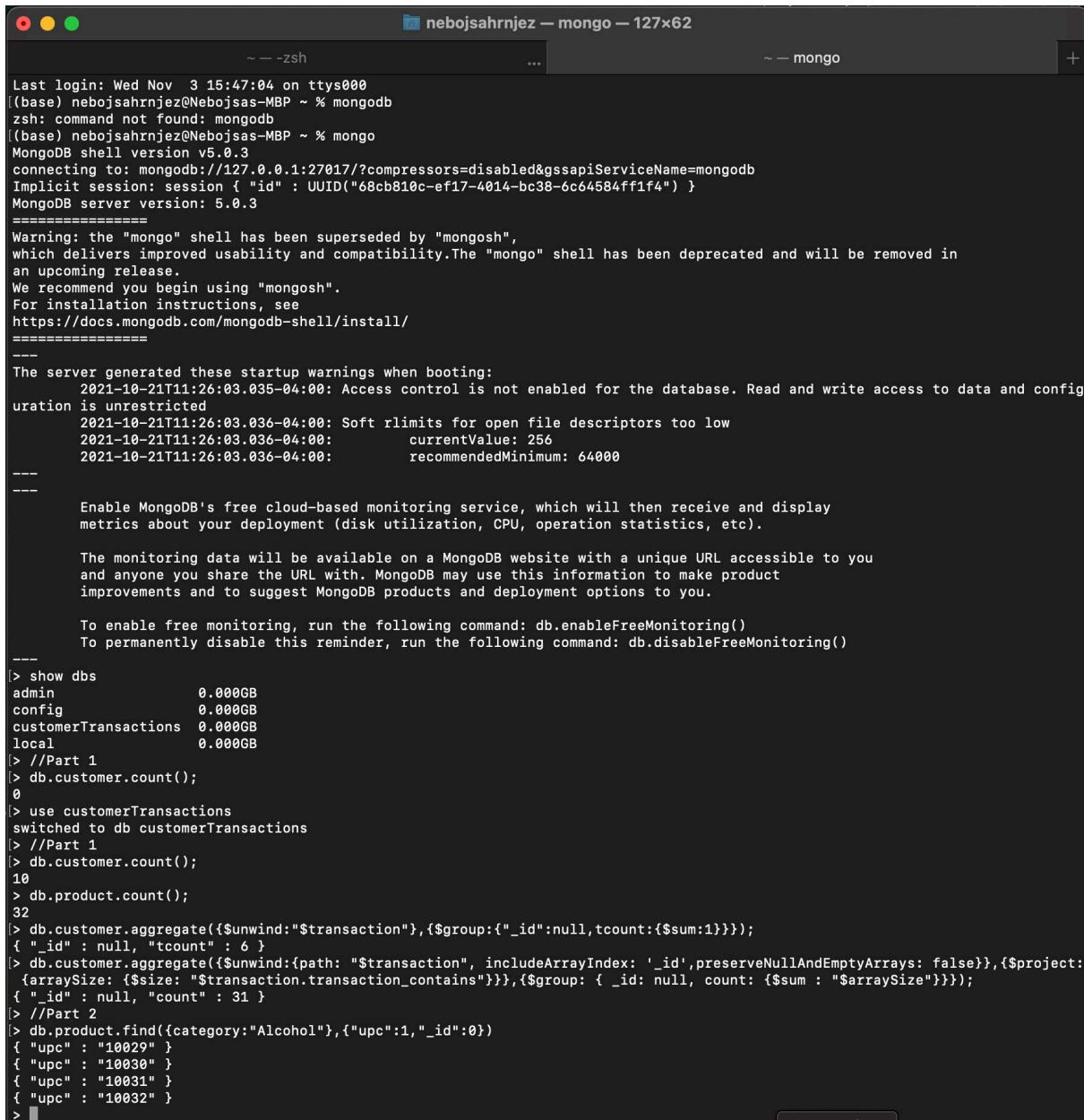
Typed the command “db.customer.count();” , this returns the count of documents in the customer collection

Typed the command “db.product.count();” , this returns the count of documents in the product collection

Typed the command “db.customer.aggregate({\$unwind：“\$transaction”},{“\$group: {“\_id”:null,tcount:{\$sum:1}}});” , this returns the count of “transactions” embedded in the documents in the customer collection

Typed the command “db.customer.aggregate({\$unwind:{path: “\$transaction”, includeArrayIndex: ‘\_id’,preserveNullAndEmptyArrays: false}},{\$project: {arraySize: {\$size: “\$transaction.transaction\_contains”}},{\$group: { \_id: null, count: {\$sum : “\$arraySize”}}}}) , this

returns the count of “transaction\_contains” that is embedded in “transactions” that is embedded in the documents in the customer collection



```
Last login: Wed Nov  3 15:47:04 on ttys000
((base) nebojsahrnjez@Nebojsas-MBP ~ % mongodb
zsh: command not found: mongodb
((base) nebojsahrnjez@Nebojsas-MBP ~ % mongo
MongoDB shell version v5.0.3
connecting to: mongodb://127.0.0.1:27017/?compressors=disabled&gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("68cb810c-ef17-4014-bc38-6c64584ff1f4") }
MongoDB server version: 5.0.3
=====
Warning: the "mongo" shell has been superseded by "mongosh",
which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in
an upcoming release.
We recommend you begin using "mongosh".
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====
The server generated these startup warnings when booting:
  2021-10-21T11:26:03.035-04:00: Access control is not enabled for the database. Read and write access to data and config
uration is unrestricted
  2021-10-21T11:26:03.036-04:00: Soft rlimits for open file descriptors too low
  2021-10-21T11:26:03.036-04:00:           currentValue: 256
  2021-10-21T11:26:03.036-04:00:           recommendedMinimum: 64000
---
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---
|> show dbs
admin          0.000GB
config         0.000GB
customerTransactions 0.000GB
local          0.000GB
|> //Part 1
|> db.customer.count();
0
|> use customerTransactions
switched to db customerTransactions
|> //Part 1
|> db.customer.count();
10
|> db.product.count();
32
|> db.customer.aggregate({$unwind:"$transaction"},{$group:{"_id":null,tcount:{$sum:1}}});
{ "_id" : null, "tcount" : 6 }
|> db.customer.aggregate({$unwind:{path: "$transaction", includeArrayIndex: '_id',preserveNullAndEmptyArrays: false}},{$project:
  {arraySize: {$size: "$transaction.transaction_contains"}},{$group: { _id: null, count: {$sum : "$arraySize"}}}};
{ "_id" : null, "count" : 31 }
|> //Part 2
|> db.product.find({category:"Alcohol"}, {"upc":1, "_id":0})
{ "upc" : "10029" }
{ "upc" : "10030" }
{ "upc" : "10031" }
{ "upc" : "10032" }
|>
```

With part 1 complete, I proceeded on part 2 of project 2:

Typed the command “db.product.find({category:”Alcohol”},{“upc”:1, “\_id”:0})” , This finds all the documents in the product collection with the “alcohol” category, I then restricted the return to only show the “upc”

MongoDB server version: 5.0.3  
=====Warning: the "mongo" shell has been superseded by "mongosh", which delivers improved usability and compatibility. The "mongo" shell has been deprecated and will be removed in an upcoming release.  
We recommend you begin using "mongosh".  
For installation instructions, see  
<https://docs.mongodb.com/mongodb-shell/install/>  
=====The server generated these startup warnings when booting:  
2021-10-21T11:26:03.035-04:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted  
2021-10-21T11:26:03.036-04:00: Soft rlimits for open file descriptors too low  
2021-10-21T11:26:03.036-04:00: currentValue: 256  
2021-10-21T11:26:03.036-04:00: recommendedMinimum: 64000  
---  
---  
Enable MongoDB's free cloud-based monitoring service, which will then receive and display metrics about your deployment (disk utilization, CPU, operation statistics, etc).  
The monitoring data will be available on a MongoDB website with a unique URL accessible to you and anyone you share the URL with. MongoDB may use this information to make product improvements and to suggest MongoDB products and deployment options to you.  
To enable free monitoring, run the following command: db.enableFreeMonitoring()  
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()  
---  
> show dbs  
admin 0.000GB  
config 0.000GB  
customerTransactions 0.000GB  
local 0.000GB  
> //Part 1  
> db.customer.count();  
0  
> use customerTransactions  
switched to db customerTransactions  
> //Part 1  
> db.customer.count();  
10  
> db.product.count();  
32  
> db.customer.aggregate({\$unwind:"\$transaction"},{\$group:{"\_id":null,tcount:{\$sum:1}}});  
{ "\_id" : null, "tcount" : 6 }  
> db.customer.aggregate({\$unwind:{path: "\$transaction", includeArrayIndex: '\_id',preserveNullAndEmptyArrays: false}},{\$project:{arraySize: {\$size: "\$transaction.transaction\_contains"}},{\$group: { \_id: null, count: {\$sum: "\$arraySize"}}}};  
{ "\_id" : null, "count" : 31 }  
> //Part 2  
> db.product.find({category:"Alcohol"}, {"upc":1,"\_id":0})  
{ "upc" : "10029" }  
{ "upc" : "10030" }  
{ "upc" : "10031" }  
{ "upc" : "10032" }  
> db.customer.find({zip\_code:"22030"}, {"first\_name":1,"last\_name":1,"\_id":0})  
{ "first\_name" : "David", "last\_name" : "Smith" }  
{ "first\_name" : "Candice", "last\_name" : "Bolton" }  
{ "first\_name" : "David", "last\_name" : "Goldenberg" }  
{ "first\_name" : "Greg", "last\_name" : "Jennings" }  
{ "first\_name" : "John", "last\_name" : "Wayne" }  
{ "first\_name" : "Jennifer", "last\_name" : "Johnson" }  
> ]

Typed the command “db.customer.find({zip\_code:”22030”}, {“first\_name”:1,”last\_name”:1,”\_id”:0})” , This returns all the documents in the customer collection with the zip code “22030” (the zip code for fairfax), I then restricted the return so it only shows the first and last name.

The screenshot shows a terminal window titled "nebojsahrnjez — mongo — 127x62". The window has two tabs: "-zsh" and "mongo". The "mongo" tab displays the MongoDB shell interface. It starts with startup messages about access control and file descriptors, followed by a reminder to enable free monitoring. Then, it shows a command history starting with "show dbs", switching to the "customerTransactions" database, and performing various aggregate and find operations on the "customer" collection. The output includes document counts and specific customer details like first and last names.

```
For installation instructions, see
https://docs.mongodb.com/mongodb-shell/install/
=====
The server generated these startup warnings when booting:
2021-10-21T11:26:03.035-04:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2021-10-21T11:26:03.036-04:00: Soft rlimits for open file descriptors too low
2021-10-21T11:26:03.036-04:00:           currentValue: 256
2021-10-21T11:26:03.036-04:00:           recommendedMinimum: 64000
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you and anyone you share the URL with. MongoDB may use this information to make product improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
> show dbs
admin          0.000GB
config         0.000GB
customerTransactions 0.000GB
local          0.000GB
> //Part 1
> db.customer.count();
0
> use customerTransactions
switched to db customerTransactions
> //Part 1
> db.customer.count();
10
> db.product.count();
32
> db.customer.aggregate({$unwind:"$transaction"},{$group:{"_id":null,tcount:{$sum:1}}});
[{"_id" : null, "tcount" : 6 }
> db.customer.aggregate({$unwind:{$path: "$transaction", includeArrayIndex: '_id',preserveNullAndEmptyArrays: false}},{$project:{$arraySize: {$size: "$transaction.transaction_contains"}},{$group: { _id: null, count: {$sum : "$arraySize"}}}};
{"_id" : null, "count" : 31 }
> //Part 2
> db.product.find({category:"Alcohol"}, {"upc":1, "_id":0})
[{"upc" : "10029" }
{"upc" : "10030" }
{"upc" : "10031" }
{"upc" : "10032" }
> db.customer.find({zip_code:"22030"}, {"first_name":1, "last_name":1, "_id":0})
[{"first_name" : "David", "last_name" : "Smith" }
{"first_name" : "Candice", "last_name" : "Bolton" }
{"first_name" : "David", "last_name" : "Goldenberg" }
{"first_name" : "Greg", "last_name" : "Jennings" }
{"first_name" : "John", "last_name" : "Wayne" }
{"first_name" : "Jennifer", "last_name" : "Johnson" }
> db.customer.find({age: {$gt:30}}, {"customer_ID":1, "_id":0})
[{"customer_ID" : "3" }
{"customer_ID" : "5" }
{"customer_ID" : "7" }
{"customer_ID" : "8" }
{"customer_ID" : "9" }
>
```

Typed the command “db.customer.find({age: {\$gt:30}}, {"customer\_ID":1, "\_id":0})” , this returns all the documents in the customer collections with an age greater than 30. I then restricted theft to only show “customer\_ID”

```

nebojsahrnjez — mongo — 127x62
~ -- -zsh ... ~ -- mongo +
=====
The server generated these startup warnings when booting:
    2021-10-21T11:26:03.035-04:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
    2021-10-21T11:26:03.036-04:00: Soft rlimits for open file descriptors too low
    2021-10-21T11:26:03.036-04:00:          currentValue: 256
    2021-10-21T11:26:03.036-04:00:          recommendedMinimum: 64000
---

Enable MongoDB's free cloud-based monitoring service, which will then receive and display metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you and anyone you share the URL with. MongoDB may use this information to make product improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()
---

[> show dbs
admin          0.000GB
config         0.000GB
customerTransactions 0.000GB
local          0.000GB
[> //Part 1
[> db.customer.count();
0
[> use customerTransactions
switched to db customerTransactions
[> //Part 1
[> db.customer.count();
10
[> db.product.count();
32
[> db.customer.aggregate({$unwind:"$transaction"},{$group:{"_id":null,tcount:{$sum:1}}});
{ "_id" : null, "tcount" : 6 }
[> db.customer.aggregate({$unwind:{path: "$transaction", includeArrayIndex: '_id',preserveNullAndEmptyArrays: false}},{$project:
  {arraySize: {$size: "$transaction.transaction_contains"}},{$group: { _id: null, count: {$sum : "$arraySize"}}});
{ "_id" : null, "count" : 31 }
[> //Part 2
[> db.product.find({category:"Alcohol"}, {"upc":1, "_id":0})
{ "upc" : "10029" }
{ "upc" : "10030" }
{ "upc" : "10031" }
{ "upc" : "10032" }
[> db.customer.find({zip_code:"22030"}, {"first_name":1, "last_name":1, "_id":0})
{ "first_name" : "David", "last_name" : "Smith" }
{ "first_name" : "Candice", "last_name" : "Bolton" }
{ "first_name" : "David", "last_name" : "Goldenberg" }
{ "first_name" : "Greg", "last_name" : "Jennings" }
{ "first_name" : "John", "last_name" : "Wayne" }
{ "first_name" : "Jennifer", "last_name" : "Johnson" }
[> db.customer.find({age: {$gt:30}}, {"customer_ID":1, "_id":0})
{ "customer_ID" : "3" }
{ "customer_ID" : "5" }
{ "customer_ID" : "7" }
{ "customer_ID" : "8" }
{ "customer_ID" : "9" }
[> db.product.find({product_Description: {$exists : false}}, {"product_name":1, "_id":0})
{ "product_name" : "Rice-A-Roni Cajun" }
>

```

Typed the command “db.product.find({product\_Description: {\$exists : false}}, {"product\_name":1, "\_id":0})” , this returns all the documents from the product collection where “product\_Description” exists evaluates to false. I then restricted the output to only showing the “product\_name”

```

2021-10-21T11:26:03.035-04:00: Access control is not enabled for the database. Read and write access to data and configuration is unrestricted
2021-10-21T11:26:03.036-04:00: Soft rlimits for open file descriptors too low
2021-10-21T11:26:03.036-04:00:           currentValue: 256
2021-10-21T11:26:03.036-04:00:           recommendedMinimum: 64000
---
---
Enable MongoDB's free cloud-based monitoring service, which will then receive and display metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you and anyone you share the URL with. MongoDB may use this information to make product improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()

|> show dbs
admin          0.000GB
config         0.000GB
customerTransactions 0.000GB
local          0.000GB
|> //Part 1
|> db.customer.count();
0
|> use customerTransactions
switched to db customerTransactions
|> //Part 1
|> db.customer.count();
10
|> db.product.count();
32
|> db.customer.aggregate({$unwind:"$transaction"},{$group:{"_id":null,tcount:{$sum:1}}});
{ "_id" : null, "tcount" : 6 }
|> db.customer.aggregate({$unwind:{path: "$transaction", includeArrayIndex: '_id',preserveNullAndEmptyArrays: false}},{$project:[{arraySize: $size: "$transaction.transaction_contains"}]},{$group: { _id: null, count: {$sum : "$arraySize"} }});
{ "_id" : null, "count" : 31 }
|> //Part 2
|> db.product.find({category:"Alcohol"}, {"upc":1, "_id":0})
{ "upc" : "10029" }
{ "upc" : "10030" }
{ "upc" : "10031" }
{ "upc" : "10032" }
|> db.customer.find({zip_code:"22030"}, {"first_name":1, "last_name":1, "_id":0})
{ "first_name" : "David", "last_name" : "Smith" }
{ "first_name" : "Candice", "last_name" : "Bolton" }
{ "first_name" : "David", "last_name" : "Goldenberg" }
{ "first_name" : "Greg", "last_name" : "Jennings" }
{ "first_name" : "John", "last_name" : "Wayne" }
{ "first_name" : "Jennifer", "last_name" : "Johnson" }
|> db.customer.find({age: {$gt:30}}, {"customer_ID":1, "_id":0})
{ "customer_ID" : "3" }
{ "customer_ID" : "5" }
{ "customer_ID" : "7" }
{ "customer_ID" : "8" }
{ "customer_ID" : "9" }
|> db.product.find({product_Description: {$exists : false}}, {"product_name":1, "_id":0})
{ "product_name" : "Rice-A-Roni Cajun" }
|> db.product.aggregate({$group:{_id:"$brand", count : {$sum : 1}}},{$match:{"_id":{$in:["Coca-Cola","Pepsi"]}}})
{ "_id" : "Pepsi", "count" : 2 }
{ "_id" : "Coca-Cola", "count" : 1 }
|>

```

Typed the command “> db.product.aggregate({\$group:{\_id：“brand”, count : {\$sum : 1}}}, {\$match:{“\_id”:{“\$in:[“Coca-Cola”, “Pepsi”]}})”, this command looks at the documents in the product collection, groups them by “brand” and pulls the count for each of these “brands”, it then matches the “\_id” to “Coca-Cola” or “Pepsi”. It then returns the two brand names desired and the count of each product with that brand name

```

nebojsahrnjez -- mongo -- 127x62
~ -- zsh
... -- mongo
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()

--> show dbs
admin          0.000GB
config         0.000GB
customerTransactions 0.000GB
local          0.000GB
--> //Part 1
--> db.customer.count();
0
--> use customerTransactions
switched to db customerTransactions
--> //Part 1
--> db.customer.count();
10
--> db.product.count();
32
--> db.customer.aggregate({$unwind:"$transaction"},{$group:{"_id":null,tcount:{$sum:1}}});
{ "_id" : null, "tcount" : 6 }
--> db.customer.aggregate({$unwind:{path: "$transaction", includeArrayIndex: '_id',preserveNullAndEmptyArrays: false}},{$project:
{arraySize: {$size: "$transaction.transaction_contains"}},{$group: { _id: null, count: {$sum : "$arraySize"}}}});
{ "_id" : null, "count" : 31 }
--> //Part 2
--> db.product.find({category:"Alcohol"}, {"upc":1, "_id":0})
{ "upc" : "10029" }
{ "upc" : "10030" }
{ "upc" : "10031" }
{ "upc" : "10032" }
--> db.customer.find({zip_code:"22030"}, {"first_name":1, "last_name":1, "_id":0})
{ "first_name" : "David", "last_name" : "Smith" }
{ "first_name" : "Candice", "last_name" : "Bolton" }
{ "first_name" : "David", "last_name" : "Goldenberg" }
{ "first_name" : "Greg", "last_name" : "Jennings" }
{ "first_name" : "John", "last_name" : "Wayne" }
{ "first_name" : "Jennifer", "last_name" : "Johnson" }
--> db.customer.find({age: {$gt:30}}, {"customer_ID":1, "_id":0})
{ "customer_ID" : "3" }
{ "customer_ID" : "5" }
{ "customer_ID" : "7" }
{ "customer_ID" : "8" }
{ "customer_ID" : "9" }
--> db.product.find({product_Description: {$exists : false}}, {"product_name":1, "_id":0})
{ "product_name" : "Rice-A-Roni Cajun" }
--> db.product.aggregate({$group:{_id:"$brand", count : {$sum : 1}}},{$match:{_id:{$in:["Coca-Cola","Pepsi"]}}})
{ "_id" : "Pepsi", "count" : 2 }
{ "_id" : "Coca-Cola", "count" : 1 }
--> db.product.aggregate({$match:{ "brand": {$in : ["Pepsi","Coca-Cola"]} },{$group:{ _id: null, count: { $sum: 1 } }}}
{ "_id" : null, "count" : 3 }
--> db.product.find({quantity:{$gte:50},marked_price:{$lt:10}}, {"product_name":1, "_id":0})
{ "product_name" : "Coca-Cola 2L" }
{ "product_name" : "Pepsi 2L" }
{ "product_name" : "Honed Glazed Ham" }
{ "product_name" : "Hot Dog Buns" }

```

Additionally, if you just want the total count for all products with the brand “Coca-Cola” or “Pepsi” the command is “db.product.aggregate({\$match:{ “brand”: {\$in : [“Pepsi”, “Coca-Cola”]} }},{\$group:{ \_id: null, count: { \$sum: 1 } }})”

```

2021-10-21T11:26:03.036-04:00:      currentValue: 256
2021-10-21T11:26:03.036-04:00:      recommendedMinimum: 64000
---
--- 
Enable MongoDB's free cloud-based monitoring service, which will then receive and display
metrics about your deployment (disk utilization, CPU, operation statistics, etc).

The monitoring data will be available on a MongoDB website with a unique URL accessible to you
and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()

[> show dbs
admin          0.000GB
config         0.000GB
customerTransactions 0.000GB
local          0.000GB
[> //Part 1
[> db.customer.count();
0
[> use customerTransactions
switched to db customerTransactions
[> //Part 1
[> db.customer.count();
10
[> db.product.count();
32
[> db.customer.aggregate({$unwind:"$transaction"},{$group:{_id:null,tcount:{$sum:1}}})
{ _id : null, "tcount" : 6 }
[> db.customer.aggregate({$unwind:{path: "$transaction", includeArrayIndex: '_id',preserveNullAndEmptyArrays: false}},{$project:
{arraySize: {$size: "$transaction.transaction_contains"}},{$group: { _id: null, count: {$sum : "$arraySize"}}})
{ _id : null, "count" : 31 }
[> //Part 2
[> db.product.find({category:"Alcohol"},{_id:0,"upc":1})
{ "upc" : "10029" }
{ "upc" : "10030" }
{ "upc" : "10031" }
{ "upc" : "10032" }
[> db.customer.find({zip_code:"22030"},{_id:0,"first_name":1,"last_name":1})
{ "first_name" : "David", "last_name" : "Smith" }
{ "first_name" : "Candice", "last_name" : "Bolton" }
{ "first_name" : "David", "last_name" : "Goldenberg" }
{ "first_name" : "Greg", "last_name" : "Jennings" }
{ "first_name" : "John", "last_name" : "Wayne" }
{ "first_name" : "Jennifer", "last_name" : "Johnson" }
[> db.customer.find({age: {$gt:30}},{_id:0,"customer_ID":1})
{ "customer_ID" : "3" }
{ "customer_ID" : "5" }
{ "customer_ID" : "7" }
{ "customer_ID" : "8" }
{ "customer_ID" : "0" }
[> db.product.find({product_Description: {$exists : false}},{_id:0,"product_name":1})
{ "product_name" : "Rice-A-Roni Cajun" }
[> db.product.aggregate({$group:{_id:"$brand", count : {$sum : 1}}},{$match:{_id:{$in:["Coca-Cola", "Pepsi"]}}})
{ "_id" : "Pepsi", "count" : 2 }
{ "_id" : "Coca-Cola", "count" : 1 }
[> db.product.aggregate({$match:{brand: {$in : ["Pepsi", "Coca-Cola"]}}},{$group:{_id: null, count: {$sum: 1}}})
{ "_id" : null, "count" : 3 }
[>

```

Typed the command “db.product.find({quantity:{\$gte:50},marked\_price:{\$lt:10}}, {"product\_name":1,”\_id”:0})” , This finds documents in the product collections with a quantity greater than 50 and a price less than 10, I then restricted the output to only show “product\_name”

```

and anyone you share the URL with. MongoDB may use this information to make product
improvements and to suggest MongoDB products and deployment options to you.

To enable free monitoring, run the following command: db.enableFreeMonitoring()
To permanently disable this reminder, run the following command: db.disableFreeMonitoring()

--> show dbs
admin          0.000GB
config         0.000GB
customerTransactions 0.000GB
local          0.000GB
--> //Part 1
--> db.customer.count();
0
--> use customerTransactions
switched to db customerTransactions
--> //Part 1
--> db.customer.count();
10
--> db.product.count();
32
--> db.customer.aggregate({$unwind:"$transaction"},{$group:{"_id":null,tcount:{$sum:1}}});
{ "_id" : null, "tcount" : 6 }
--> db.customer.aggregate({$unwind:{path: "$transaction", includeArrayIndex: '_id',preserveNullAndEmptyArrays: false}},{$project:
  {arraySize: {$size: "$transaction.transaction_contains"}},{$group: { _id: null, count: {$sum : "$arraySize"}}}};
{ "_id" : null, "count" : 31 }
--> //Part 2
--> db.product.find({category:"Alcohol"}, {"upc":1, "_id":0})
{ "upc" : "10029" }
{ "upc" : "10030" }
{ "upc" : "10031" }
{ "upc" : "10032" }
--> db.customer.find({zip_code:"22030"}, {"first_name":1, "last_name":1, "_id":0})
{ "first_name" : "David", "last_name" : "Smith" }
{ "first_name" : "Candice", "last_name" : "Bolton" }
{ "first_name" : "David", "last_name" : "Goldenberg" }
{ "first_name" : "Greg", "last_name" : "Jennings" }
{ "first_name" : "John", "last_name" : "Wayne" }
{ "first_name" : "Jennifer", "last_name" : "Johnson" }
--> db.customer.find({age: {$gt:30}}, {"customer_ID":1, "_id":0})
{ "customer_ID" : "3" }
{ "customer_ID" : "5" }
{ "customer_ID" : "7" }
{ "customer_ID" : "8" }
{ "customer_ID" : "9" }
--> db.product.find({product_Description: {$exists : false}}, {"product_name":1, "_id":0})
{ "product_name" : "Rice-A-Roni Cajun" }
--> db.product.aggregate({$group:{_id:"$brand", count : {$sum : 1}}},{$match:{_id:{$in:["Coca-Cola","Pepsi"]}}})
{ "_id" : "Pepsi", "count" : 2 }
{ "_id" : "Coca-Cola", "count" : 1 }
--> db.product.aggregate({$match:{ "brand": {$in : ["Pepsi","Coca-Cola"]} }},{$group:{ _id: null, count: { $sum: 1 }}})
{ "_id" : null, "count" : 3 }
--> db.product.find({quantity:{$gte:50},marked_price:{$lt:10}}, {"product_name":1, "_id":0})
{ "product_name" : "Coca-Cola 2L" }
{ "product_name" : "Pepsi 2L" }
{ "product_name" : "Honed Glazed Ham" }
{ "product_name" : "Hot Dog Buns" }
--> db.customer.aggregate({$unwind:{path:"$transaction",includeArrayIndex:"_id",preserveNullAndEmptyArrays:true}},{$match:{transaction:null}},{$group:{_id:"no transaction",count: {$sum:1}}})
{ "_id" : "no transaction", "count" : 5 }

```

Typed the command “db.customer.aggregate({\$unwind:  
{path:”\$transaction”,includeArrayIndex:”\_id”,preserveNullAndEmptyArrays:true}},{\$match:  
{transaction:null}},{\$group:{\_id:”no transaction”,count: {\$sum:1}}})”, this command goes into  
the transactions embedded within the documents in the customer collection, it finds all the  
transactions that contain “null” or nothing and then counts them

```

config          0.000GB
customerTransactions 0.000GB
local          0.000GB
> //Part 1
> db.customer.count();
0
> use customerTransactions
switched to db customerTransactions
> //Part 1
> db.customer.count();
10
> db.product.count();
32
> db.customer.aggregate({$unwind:"$transaction"},{$group:{_id:null,tcount:{$sum:1}}});
{ "_id" : null, "tcount" : 6 }
> db.customer.aggregate({$unwind:{path: "$transaction", includeArrayIndex: '_id',preserveNullAndEmptyArrays: false}},{$project:{$arraySize: {$size: "$transaction.transaction_contains"}},{$group: { _id: null, count: {$sum : "$arraySize"}}}};
{ "_id" : null, "count" : 31 }
> //Part 2
> db.product.find({category:"Alcohol"}, {"upc":1, "_id":0})
{ "upc" : "10029" }
{ "upc" : "10030" }
{ "upc" : "10031" }
{ "upc" : "10032" }
> db.customer.find({zip_code:"22030"}, {"first_name":1, "last_name":1, "_id":0})
{ "first_name" : "David", "last_name" : "Smith" }
{ "first_name" : "Candice", "last_name" : "Bolton" }
{ "first_name" : "David", "last_name" : "Goldenberg" }
{ "first_name" : "Greg", "last_name" : "Jennings" }
{ "first_name" : "John", "last_name" : "Wayne" }
{ "first_name" : "Jennifer", "last_name" : "Johnson" }
> db.customer.find({age: {$gt:30}}, {"customer_ID":1, "_id":0})
{ "customer_ID" : "3" }
{ "customer_ID" : "5" }
{ "customer_ID" : "7" }
{ "customer_ID" : "8" }
{ "customer_ID" : "9" }
> db.product.find({product_Description: {$exists : false}}, {"product_name":1, "_id":0})
{ "product_name" : "Rice-A-Roni Cajun" }
> db.product.aggregate({$group:{_id:"$brand", count : {$sum : 1}}},{$match:{_id:{$in:["Coca-Cola", "Pepsi"]}}}
{ "_id" : "Pepsi", "count" : 2 }
{ "_id" : "Coca-Cola", "count" : 1 }
> db.product.aggregate({$match:{ "brand": {$in : ["Pepsi", "Coca-Cola"]} }},{$group:{ _id: null, count: { $sum: 1 } }})
{ "_id" : null, "count" : 3 }
> db.product.find({quantity:{$gte:50},marked_price:{$lt:10}}, {"product_name":1, "_id":0})
{ "product_name" : "Coca-Cola 2L" }
{ "product_name" : "Pepsi 2L" }
{ "product_name" : "Honed Glazed Ham" }
{ "product_name" : "Hot Dog Buns" }
> db.customer.aggregate({$unwind:{path:"$transaction",includeArrayIndex:"_id",preserveNullAndEmptyArrays:true}},{$match:{transaction:null}},{$group:{_id:"no transaction",count: {$sum:1}}})
{ "_id" : "no transaction", "count" : 5 }
> //Extra Credit
> db.customer.aggregate({$unwind:{path:"$transaction",preserveNullAndEmptyArrays:false}},{$unwind:{path:"$transaction.transaction_contains"},{$lookup:
{from:"product",localField:"upc",foreignField:"transaction.transaction_contains.upc",as:"item"}},{$unwind:{path:"$item"}},{$match:{item.category:"Alcohol"}},{$match:{$expr: { "$eq": ["$transaction.transaction_contains.upc", "$item.upc"]}}},{$group:{_id:"$transaction.transaction_ID",Transaction_ID:{$first:"$transaction.transaction_ID"}},{$project:{"Transaction_ID":1, "_id":0}}}
{ "Transaction_ID" : "104" }
{ "Transaction_ID" : "103" }

```

Typed the command “db.customer.aggregate({\$unwind: {path:”\$transaction”, preserveNullAndEmptyArrays:false}},{\$unwind: {path:”\$transaction.transaction\_contains”}},{\$lookup: {from:”product”,localField:”upc”,foreignField:”transaction.transaction\_contains.upc”,as:”item”}},{\$unwind:{path:”\$item”}},{\$match:{item.category:”Alcohol”}},{\$match:{\$expr: { ”\$eq”: [”\$transaction.transaction\_contains.upc”, ”\$item.upc”]}},{\$group: {\_id:”\$transaction.transaction\_ID”,Transaction\_ID:{\$first:”\$transaction.transaction\_ID”}}},{\$project:{”Transaction\_ID”:1, ”\_id”:0}}” , this command creates a new field from the

documents in product called “upc”, this is then used to compare to the “upc” field in the embedded transaction contains.