A dark blue vertical bar runs down the left side of the page. A blue arrow points to the right from this bar, containing the date.

4/8/2019

Mongo Database Report

Several thin, curved lines in dark blue and light grey originate from the bottom left and sweep upwards and to the right.

Jonathan McDonagh
20074520

Contents

Introduction	2
Installation of Mongo	2
Setting up the cloud service	5
Original Entity Relationship Diagram	7
Mongo Database	7
Create	7
Customers	7
Menu	8
Waiter	10
Order	11
Queries	14
Aggregation	14
Update	15
Delete	15
Conclusion	16
References	17

Figures

<i>Figure 1: Download for MongoDB</i>	2
<i>Figure 2: MongoDB Setup</i>	3
<i>Figure 3: Setup Finished for MongoDB</i>	3
<i>Figure 4: Mongod.exe from Command Line</i>	4
<i>Figure 5: Mongo.exe to MongoDB Shell</i>	4
<i>Figure 6: mongodump to export DB</i>	5
<i>Figure 7: mongorestore used to put database on the cloud</i>	5
<i>Figure 8: Connecting to database on Cloud and testing a find query on the Cloud</i>	6
<i>Figure 9: ER Diagram</i>	7

Introduction

This report is for my NoSQL Databases module and we will look at the mongo database that I created using MongoDB. The database is for a Restaurant Database. My database has 4 tables put into a MongoDB. “MongoDB is a document database with the scalability and flexibility that you want with the querying and indexing that you need” (Mongo, 2019)

Installation of Mongo

I downloaded MongoDB from <https://www.mongodb.com/download-center/community>, and installed it onto my laptop. I started the MongoDB server by navigating to the mongo/bin in the command line. Then I typed mongo.exe to start the server. This allowed me to start the shell process and allowed me to create my database by entering my inserts that are in this report.

Here are a few screenshots of how I setup MongoDB.

Step 1: Here is where I downloaded MongoDB:

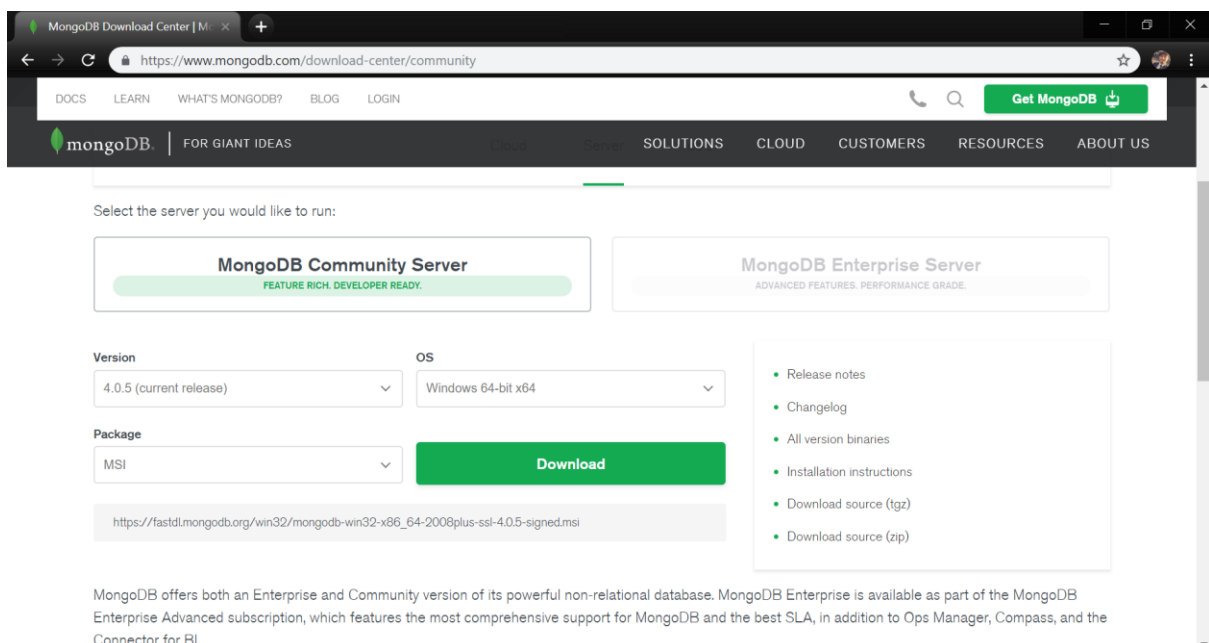


Figure 1: Download for MongoDB

Step 2: Here I started the installation process :

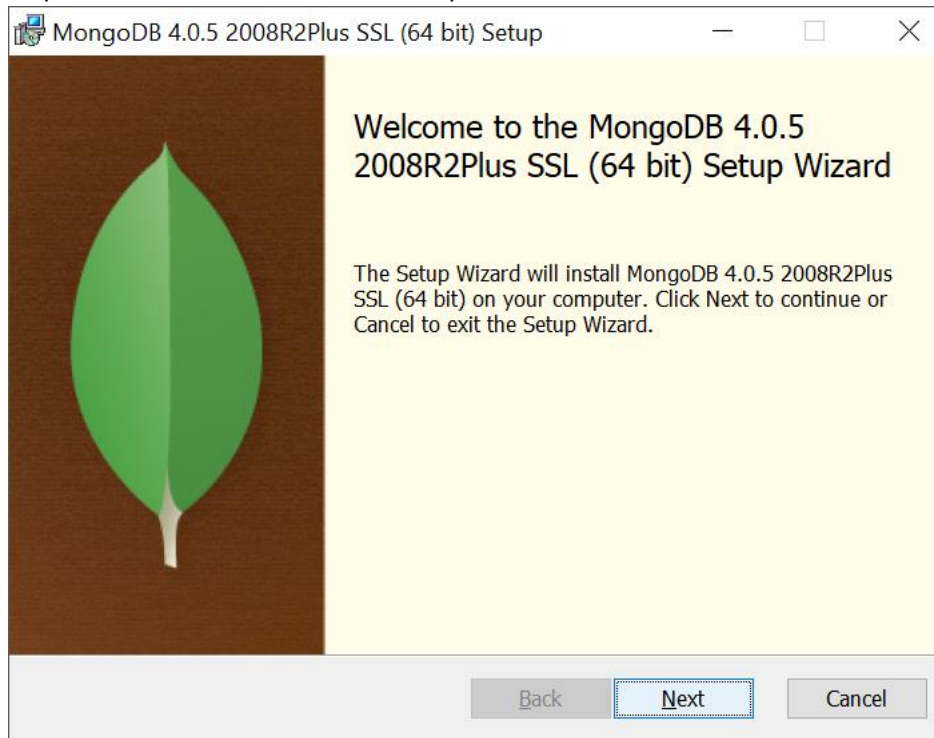


Figure 2: MongoDB Setup

Step 3: Here I completed the installation:

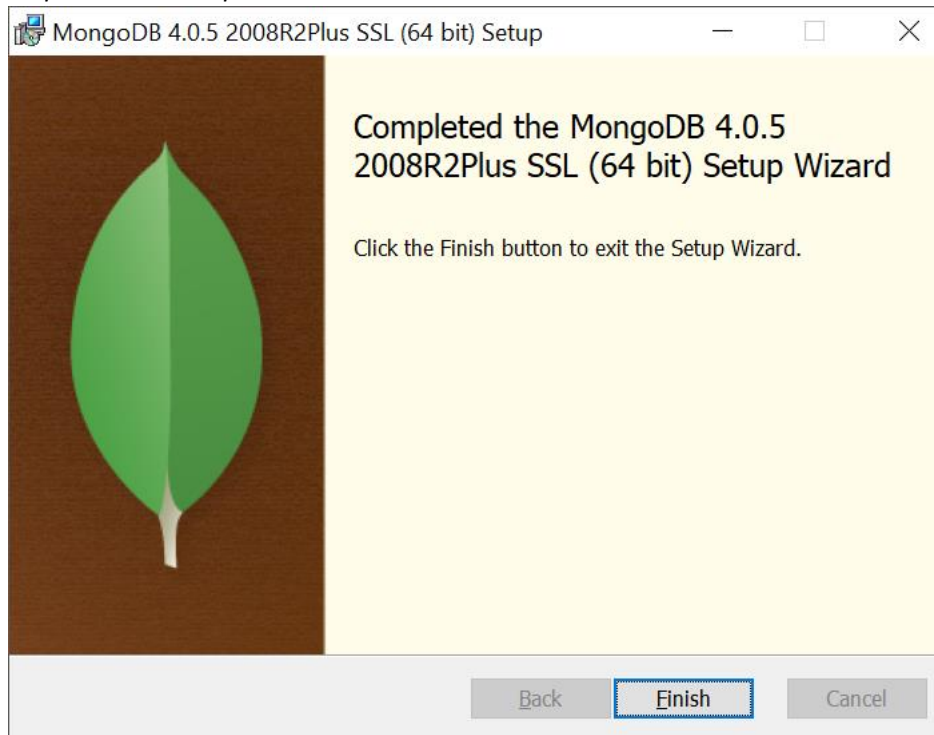


Figure 3: Setup Finished for MongoDB

Step 4: Here I executed mongod.exe from the MongoDB\4.0\bin\ directory

```

C:\Program Files\MongoDB\Server\4.0\bin>dir
03/04/2018  18:58                2,462,720 libeay32.dll
19/12/2018  19:14                18,417,664 mongo.exe
15/01/2019  17:37                 616 mongod.cfg
19/12/2018  19:20                32,523,264 mongod.exe
19/12/2018  19:20            357,355,520 mongod.pdb
19/12/2018  18:56            15,931,795 mongodump.exe
19/12/2018  18:53            13,962,639 mongorestore.exe
19/12/2018  18:52            13,847,511 mongofiles.exe
19/12/2018  18:54            14,122,933 mongoimport.exe
19/12/2018  18:55            17,042,858 mongorestore.exe
19/12/2018  19:09            16,837,120 mongos.exe
19/12/2018  19:09            187,379,712 mongos.pdb
19/12/2018  18:51            14,152,541 mongostat.exe
19/12/2018  18:57            13,807,887 mongotop.exe
03/04/2018  18:58                357,888 ssleay32.dll
17 File(s)      731,812,020 bytes
2 Dir(s)        148,213,768,192 bytes free

C:\Program Files\MongoDB\Server\4.0\bin>mongod.exe
2019-01-15T09:45:57.942-0800 I CONTROL [main] Automatically disabling TLS 1.0, to force-enable TLS 1.0 specify --sslDisabledProtocols 'none'
2019-01-15T09:45:57.945-0800 I CONTROL [initandlisten] MongoDB starting : pid=4596 port=27017 dbpath=C:\data\db\ 64-bit host=DESKTOP-SLALSD2
2019-01-15T09:45:57.945-0800 I CONTROL [initandlisten] targetMinOS: Windows 7/Windows Server 2008 R2
2019-01-15T09:45:57.946-0800 I CONTROL [initandlisten] db version v4.0.5
2019-01-15T09:45:57.946-0800 I CONTROL [initandlisten] git version: 3739429dd92b92d1b0ab120911a23d50bf03c412
2019-01-15T09:45:57.946-0800 I CONTROL [initandlisten] allocator: tomalloc
2019-01-15T09:45:57.947-0800 I CONTROL [initandlisten] modules: none
2019-01-15T09:45:57.947-0800 I CONTROL [initandlisten] build environment:
2019-01-15T09:45:57.948-0800 I CONTROL [initandlisten] distmod: 2008plus-ssl
2019-01-15T09:45:57.948-0800 I CONTROL [initandlisten] distarch: x86_64
2019-01-15T09:45:57.948-0800 I CONTROL [initandlisten] target_arch: x86_64
2019-01-15T09:45:57.949-0800 I CONTROL [initandlisten] options: {}
2019-01-15T09:45:57.950-0800 I STORAGE [initandlisten] exception in initAndListen: NonExistentPath: Data directory C:\data\db\ not found., terminating
2019-01-15T09:45:57.950-0800 I NETWORK [initandlisten] shutdown: going to close listening sockets...
2019-01-15T09:45:57.950-0800 I CONTROL [initandlisten] now exiting
2019-01-15T09:45:57.950-0800 I CONTROL [initandlisten] shutting down with code:100

C:\Program Files\MongoDB\Server\4.0\bin>

```

Figure 4: Mongod.exe from Command Line

Step 5: I then changed the directory to the bin folder of the MongoDB I then ran mongo.exe which brought me into the MongoDB shell

```

C:\Users\Jonat>cd "C:\Program Files\MongoDB\Server\4.0\bin"
C:\Program Files\MongoDB\Server\4.0\bin>mongo.exe
MongoDB shell version v4.0.5
connecting to: mongodb://127.0.0.1:27017/?sslapiServiceName=mongodb
Implicit session: session { "id" : UUID("76d5f94a-980a-4301-baad-b5fffa00bbd2") }
MongoDB server version: 4.0.5
Welcome to the MongoDB shell.
For interactive help, type "help".
For more comprehensive documentation, see
  http://docs.mongodb.org/
Questions? Try the support group
  http://groups.google.com/group/mongodb-user
Server has startup warnings:
2019-01-15T09:37:49.715-0800 I CONTROL [initandlisten]
2019-01-15T09:37:49.715-0800 I CONTROL [initandlisten] ** WARNING: Access control is not enabled for the database.
2019-01-15T09:37:49.715-0800 I CONTROL [initandlisten] **          Read and write access to data and configuration is unrestricted.
2019-01-15T09:37:49.715-0800 I CONTROL [initandlisten]
---
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()
---
```

Figure 5: Mongo.exe to MongoDB Shell

Setting up the cloud service

To setup the cloud for my database I used Clever-Cloud. I started of by using mongodump in the command line to export the database. Once exported I used the mongorestore which I got of Clever-Cloud which put the database onto the cloud. Once I completed those two steps I used the mongo – host command which I got of Clever-Cloud. I could then run queries on the cloud from the command line.

Below is the mongodump command I used to export the database

```
C:\Program Files\MongoDB\Server\4.0\bin>mongodump --db Restaurant_DB --gzip --archive > dump_Restaurant_DB.gz
2019-04-02T19:21:40.383+0100   writing Restaurant_DB.Waiter to archive on stdout
2019-04-02T19:21:40.393+0100   writing Restaurant_DB.customer to archive on stdout
2019-04-02T19:21:40.393+0100   writing Restaurant_DB.menu to archive on stdout
2019-04-02T19:21:40.396+0100   writing Restaurant_DB.Order to archive on stdout
2019-04-02T19:21:40.423+0100   done dumping Restaurant_DB.Waiter (5 documents)
2019-04-02T19:21:41.427+0100   done dumping Restaurant_DB.menu (5 documents)
2019-04-02T19:21:41.428+0100   done dumping Restaurant_DB.customer (5 documents)
2019-04-02T19:21:41.429+0100   done dumping Restaurant_DB.Order (4 documents)
```

Figure 6: mongodump to export DB

Below is the mongorestore command I used to put the database on the cloud

```
C:\Program Files\MongoDB\Server\4.0\bin>mongorestore --host=bqqaneknatuqube-mongodb.services.clever-cloud.com --port=27017 --username=u7xmcvmt51qlfmbjkc --password=ndwSdQYqF5mrUj79HiZh --nsFrom="Restaurant_DB.*" --nsTo="bqqaneknatuqube.*" --authenticationDatabase="bqqaneknatuqube" --archive=C:\data\dump_Restaurant_DB.gz --gzip
2019-04-02T19:24:24.685+0100   preparing collections to restore from
2019-04-02T19:24:26.377+0100   reading metadata for bqqaneknatuqube.Waiter from archive 'C:\data\dump_Restaurant_DB.gz'
2019-04-02T19:24:27.500+0100   restoring bqqaneknatuqube.Waiter from archive 'C:\data\dump_Restaurant_DB.gz'
2019-04-02T19:24:29.309+0100   no indexes to restore
2019-04-02T19:24:29.310+0100   finished restoring bqqaneknatuqube.Waiter (5 documents)
2019-04-02T19:24:29.860+0100   reading metadata for bqqaneknatuqube.menu from archive 'C:\data\dump_Restaurant_DB.gz'
2019-04-02T19:24:31.191+0100   restoring bqqaneknatuqube.menu from archive 'C:\data\dump_Restaurant_DB.gz'
2019-04-02T19:24:31.193+0100   reading metadata for bqqaneknatuqube.customer from archive 'C:\data\dump_Restaurant_DB.gz'
2019-04-02T19:24:32.727+0100   no indexes to restore
2019-04-02T19:24:32.728+0100   finished restoring bqqaneknatuqube.menu (5 documents)
2019-04-02T19:24:32.731+0100   restoring bqqaneknatuqube.customer from archive 'C:\data\dump_Restaurant_DB.gz'
2019-04-02T19:24:32.738+0100   reading metadata for bqqaneknatuqube.Order from archive 'C:\data\dump_Restaurant_DB.gz'
2019-04-02T19:24:33.391+0100   bqqaneknatuqube.customer 541B
2019-04-02T19:24:33.548+0100   restoring bqqaneknatuqube.Order from archive 'C:\data\dump_Restaurant_DB.gz'
2019-04-02T19:24:33.549+0100   bqqaneknatuqube.customer 541B
2019-04-02T19:24:33.555+0100   no indexes to restore
2019-04-02T19:24:33.556+0100   finished restoring bqqaneknatuqube.customer (5 documents)
2019-04-02T19:24:34.162+0100   no indexes to restore
2019-04-02T19:24:34.162+0100   finished restoring bqqaneknatuqube.Order (4 documents)
2019-04-02T19:24:34.166+0100   done
```

Figure 7: mongorestore used to put database on the cloud

Below I connected to the cloud server and ran a simple find query for the database on the cloud

```
C:\Program Files\MongoDB\Server\4.0\bin>mongo --host bqaneknatuqube-mongodb.services.clever-cloud.com --port 27017 --username=u7xmcvmt51qlfmbikc --password=ndwSdQYqF5mrU
j79HiZh bqaneknatuqube
MongoDB shell version v4.0.5
connecting to: mongodb://bqaneknatuqube-mongodb.services.clever-cloud.com:27017/bqaneknatuqube?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("7a938242-8db5-4a61-906f-63653c80d06b") }
MongoDB server version: 4.0.3
rs0:PRIMARY> db.menu.find().pretty();
{
  "_id" : ObjectId("5ca3a62a62afb83fad9e3733"),
  "items_ID" : 1,
  "Items" : "Any Burger and Fries",
  "Price" : 12
}
{
  "_id" : ObjectId("5ca3a63362afb83fad9e3734"),
  "items_ID" : 2,
  "Items" : "Pizza and Fries",
  "Price" : 15
}
{
  "_id" : ObjectId("5ca3a63a62afb83fad9e3735"),
  "item_ID" : 3,
  "Items" : "Sunday Roast and a pint",
  "Price" : 14
}
{
  "_id" : ObjectId("5ca3a63f62afb83fad9e3736"),
  "item_ID" : 4,
  "Items" : "Any desert",
  "Price" : 6
}
{
  "_id" : ObjectId("5ca3a65362afb83fad9e3738"),
  "item_ID" : 6,
  "Items" : "Pints",
  "Price" : 5
}
rs0:PRIMARY>
```

Figure 8: Connecting to database on Cloud and testing a find query on the Cloud

Original Entity Relationship Diagram

Below are the four tables that I focused on for my MongoDB Report.

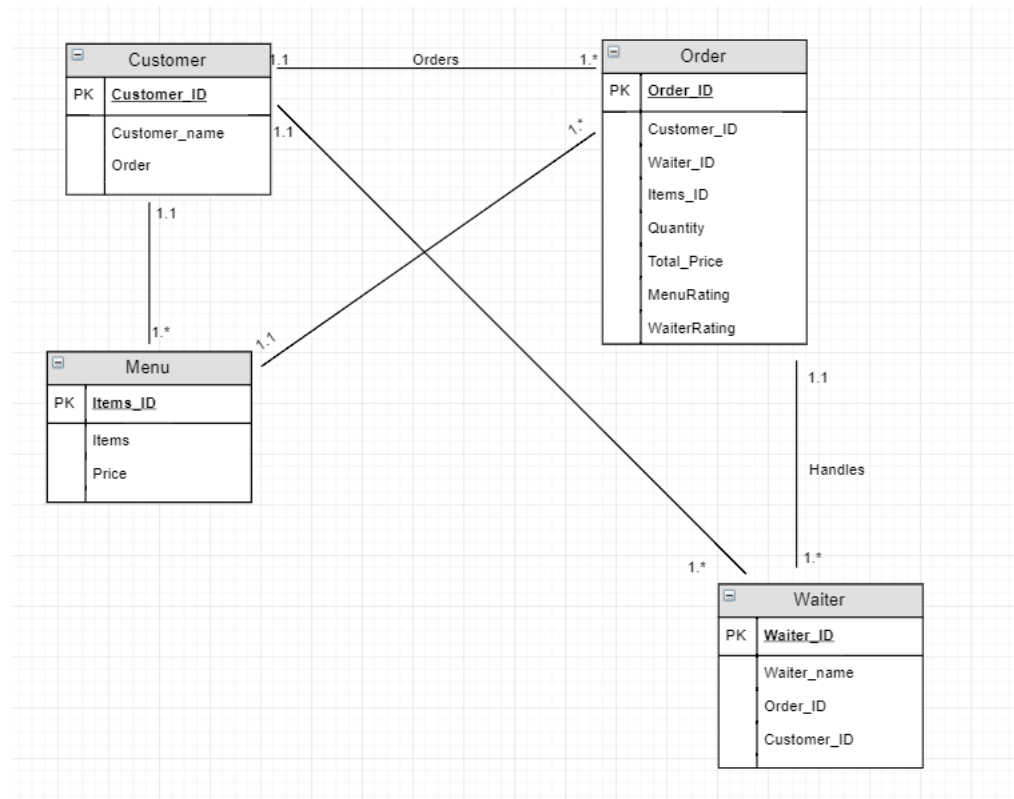


Figure 9: ER Diagram

Mongo Database

I created a database for a series of collections, Customer, Menu, Waiter and Order and I inserted multiple documents into each one. Below I will display all the code I used to create the CRUD (Create, Read, Update and Delete). The customer can have multiple items from the menu assigned to them using an array. I also used included some queries, here is the code below.

Create

Customers

```

db.customer.insert({
    customer_ID:1,
    customer_name:'Jonathan',
    order: 'Pizza meal and desert and drink'
});
  
```



```
db.customer.insert({  
  customer_ID:2,  
  customer_name:'James',  
  order: 'Burger meal and desert'  
});
```

```
db.customer.insert({  
  customer_ID:3,  
  customer_name:'Lauren',  
  order: 'Pizza meal and desert and drink'  
});
```

```
db.customer.insert({  
  customer_ID:4,  
  customer_name:'Richie',  
  order: 'Burger meal and desert'  
});
```

```
db.customer.insert({  
  customer_ID:5,  
  customer_name:'Nick',  
  order: 'Burger meal and desert'  
});
```

Menu

```
db.menu.insert({  
  items_ID:1,  
  Items:'Any Burger and Fries',  
  Price:'12'  
});
```

```
db.menu.insert({  
  items_ID:2,  
  Items:'Pizza and Fries',  
  Price:'15'  
});
```

```
db.menu.insert({  
  item_ID:3,  
  Items:'Sunday Roast and a pint',  
  Price:'14'  
});
```

```
db.menu.insert({  
  item_ID:4,  
  Items:'Any desert',  
  Price:'6'  
});
```

```
db.menu.insert({  
  item_ID:5,  
  Items:'Curry with rice and chips',  
  Price:'12'  
});
```

```
db.menu.insert({  
  item_ID:6,  
  Items:'Pints',  
  Price:'5'  
});
```

Waiter

```
db.Waiter.insert({  
    Waiter_ID:1,  
    Waiter_name: 'Andy',  
    Order_ID: 1,  
    customer_ID: 1  
});
```

```
db.Waiter.insert({  
    Waiter_ID:2,  
    Waiter_name: 'Jessica',  
    Order_ID: 2,  
    customer_ID: 2  
});
```

```
db.Waiter.insert({  
    Waiter_ID:3,  
    Waiter_name: 'Mick',  
    Order_ID: 3,  
    customer_ID: 3  
});
```

```
db.Waiter.insert({  
    Waiter_ID:4,  
    Waiter_name: 'Jake',  
    Order_ID: 2,  
    customer_ID: 4,  
});
```

```
db.Waiter.insert({  
  Waiter_ID:5,  
  Waiter_name: 'Lisa',  
  Order_ID: 1,  
  customer_ID: 5  
});
```

Order

```
db.Order.insert({  
  Order_ID: 1,  
  customer_ID: 1,  
  Waiter_ID: 1,  
  item_ID: [  
    2, 4, 6,  
  ],  
  Quantity: 1,  
  Total_Price: 26,  
  MenuRating: '5 out of 5',  
  WaiterRating: '5 out of 5',  
});
```

```
db.Order.insert({  
  Order_ID: 2,  
  customer_ID: 2,  
  Waiter_ID: 2,  
  item_ID: [  
    1, 4,  
  ],  
  Quantity: 1,  
  Total_Price: 18,  
  MenuRating: '5 out of 5',  
  WaiterRating: '5 out of 5'  
});
```

```
db.Order.insert({  
  Order_ID: 3,  
  customer_ID: 3,  
  Waiter_ID: 3,  
  item_ID: [  
    2, 4, 6,  
  ],  
  Quantity: 1,  
  Total_Price: 26,  
  MenuRating: '4 out of 5',  
  WaiterRating: '5 out of 5'  
});
```

```
db.Order.insert({  
  Order_ID: 4,  
  customer_ID: 4,  
  Waiter_ID: 4,  
  item_ID: [  
    1, 4,  
  ],  
  Quantity: 1,  
  Total_Price: 18,  
  MenuRating: '5 out of 5',  
  WaiterRating: '5 out of 5'  
});
```

```
db.Order.insert({  
  Order_ID: 5,  
  customer_ID: 5,  
  Waiter_ID: 5,  
  item_ID: [  
    1, 4,  
  ],  
  Quantity: 1,  
  Total_Price: 18,  
  MenuRating: '3 out of 5',  
  WaiterRating: '4 out of 5'  
});
```

Queries

This query displays all customers who had item 2 from the menu (Pizza and Fries):

```
db.Order.find({item_ID:2,}).pretty();
```

These four find queries below just displays all the records in each table:

```
db.customer.find().pretty();
```

```
db.Waiter.find().pretty();
```

```
db.menu.find().pretty();
```

```
db.Order.find().pretty();
```

Here I have a query to count all the customers and waiters:

```
db.customer.find().count();
```

```
db.Waiter.find().count();
```

Here is a query that I used to find any items on the menu that are lower than 10 euro:

```
db.menu.find({"Price":{$lt:10}}).pretty();
```

and here is the query to find any items that are greater than 10 euro:

```
db.menu.find({"Price":{$gt:10}}).pretty();
```

Aggregation

This query \$unwind deconstructs an array field from the input documents to output a document for each element. Each output document is the input document with the value of the array field replaced by the element.:

```
db.Order.aggregate([{$unwind : "$item_ID" }]).pretty();
```

This query \$project passes along the documents with the requested fields to the next stage in the pipeline. The specified fields can be existing fields from the input documents or newly computed fields. The code below displays all the customer names and their order:

```
db.customer.aggregate([{$project : { customer_name: 1, order: 1 } }]).pretty();
```

Update

Here I update the order for customer 5 (Nick), the update changes the order from 'Burger meal and desert' to 'Burger meal and desert and pint'.

```
db.customer.update({
    customer_ID:5},
    {$set: { order: 'Burger meal and desert and pint'
    }
});
```

Here I update the item_ID for customer 5 (Nick), I updated the item_ID from '1, 4' to '1, 4, 6' so it includes the Burger meal (1) the desert (4) and the pint (6).

```
db.Order.update({
    Order_ID:5},
    {
    $set: {Order_ID: [1,4,6,], Total_Price: 23,}
});
```

Delete

I created a simple remove statement to remove the only item on the menu that was not ordered the Chicken Curry. Its item_ID is 5.

```
db.menu.remove({
    item_ID:5,
});
```


Conclusion

After completing my assignment I feel that I learned a lot of new skills regarding NoSQL and MongoDB. I found the initial design of my Mongo database a bit difficult but I quickly learned. I found the insertion of data to sometimes be a bit tedious. But saying that, I did enjoy learning the syntax and performing some queries.

I found Mongo to be extremely user friendly. For example I liked the way Exceptions are not thrown if a query contains an unknown field. I found MongoDB interesting due to its flexibility and its scalability. This definitely is a major advantage over other traditional databases.

I used Clever-Cloud to store my data on the cloud. Which I found extremely easy to setup.

Overall I really enjoyed this assignment and I'm happy that I was able to develop my skills in this area.

References

MongoDB. 2019. MongoDB. [Online] Available at: <https://www.mongodb.com/what-is-mongodb> [Accessed 05 March 2019].