

MSDS 7330

File Organization and Database Management Final Exam

Due 23 April 2019

Name: Javier Saldana

Final Exam: Directions

This is a final exam for MSDS 7330, File Organization and Database Management. This document contains the questions for the exam. Follow the question directions and submit the requested information by 23 April 2019.

Final Exam: Questions

Use MongoDB as a database to store the eCommerce Database that we used in class. Use Python to connect to the MongoDB database and insert, update, and query data from MongoDB. Submit the code and screenshots of your execution.

Run the following SQL to view the individual tables. Once the queries are run, the data is then exported into CSV files. Each table is exported into its own respective CSV file in order to facilitate the data import to MongoDB.

The screenshot shows the MySQL Workbench interface with the following components:

- Navigator:** Displays the database schema 'ecommercedb' with tables: address, customertable, item, itemprice, orderheader, orderline, and orderpayment. It also shows 'Views', 'Stored Procedures', and 'Functions'.
- SQL Editor:** Contains a query labeled 'Query 1' with the following SQL statements:


```

1  -- Each line represents its respective table
2  • SELECT * FROM address;
3  • SELECT * FROM customertable;
4  • SELECT * FROM item;
5  • SELECT * FROM itemprice;
6  • SELECT * FROM orderheader;
7  • SELECT * FROM orderline;
8  • SELECT * FROM orderpayment;
9  -- Once data is queried, it is exported using built-in export tool of MySQL
10 -- Files are exported into CSV files on the local machine
      
```
- Result Grid:** Displays the results of the query. A red circle highlights the 'Export/Imports' button in the toolbar above the grid. The grid shows a table with columns: PaymentId, PaymentType, CardNumber, AmountCharged, and OrderHeader_OrderNumber. The data consists of 21 rows of credit card transactions.
- Output:** Shows the execution log with two entries:

#	Time	Action	Message	Duration / Fetch
6	20:55:10	SELECT * FROM orderline LIMIT 0, 1000	1000 row(s) returned	0.000 sec / 0.000 sec
7	20:55:10	SELECT * FROM orderpayment LIMIT 0, 1000	1000 row(s) returned	0.000 sec / 0.000 sec

Initiate shell and create database in MongoDB

```
Microsoft Windows [Version 10.0.17134.706]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\javie>"C:\Program Files\MongoDB\Server\4.0\bin\mongo.exe"
MongoDB shell version v4.0.6
connecting to: mongodb://127.0.0.1:27017/?gssapiServiceName=mongodb
Implicit session: session { "id" : UUID("4b9c6baf-09e8-4505-ac00-01b55cd4ad6a") }
MongoDB server version: 4.0.6
Server has startup warnings:
2019-04-11T20:18:43.652-0500 I CONTROL [initandlisten]
2019-04-11T20:18:43.652-0500 I CONTROL [initandlisten] ** WARNING: Access control is not enabled
for the database.
2019-04-11T20:18:43.652-0500 I CONTROL [initandlisten] **           Read and write access to data
and configuration is unrestricted.
2019-04-11T20:18:43.652-0500 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()
---

> use ecommerce
switched to db ecommerce
```

Create the collections for each table export from MySQL

```
> db.createCollection("Address")
{ "ok" : 1 }
>
> db.createCollection("CustomerTable")
{ "ok" : 1 }
>
> db.createCollection("Item")
{ "ok" : 1 }
>
> db.createCollection("ItemPrice")
{ "ok" : 1 }
>
> db.createCollection("OrderHeader")
{ "ok" : 1 }
>
> db.createCollection("OrderLine")
{ "ok" : 1 }
>
> db.createCollection("OrderPayment")
{ "ok" : 1 }
```

Import tables into collections

```
C:\Users\javie>"C:\Program Files\MongoDB\Server\4.0\bin\mongoimport.exe" --db ecommerce --type
csv --collection Address --drop --headerline --file "C:\Users\javie\OneDrive\Documents\Academics\
SMU\MSDS 7330 - File Organization and Database Management\Final\Address.csv"
2019-04-15T21:36:43.113-0500    connected to: localhost
2019-04-15T21:36:43.120-0500    dropping: ecommerce.Address
2019-04-15T21:36:44.206-0500    imported 100 documents

C:\Users\javie>"C:\Program Files\MongoDB\Server\4.0\bin\mongoimport.exe" --db ecommerce --type
csv --collection CustomerTable --drop --headerline --file "C:\Users\javie\OneDrive\Documents\Acad
emics\SMU\MSDS 7330 - File Organization and Database Management\Final\CustomerTable.csv"
2019-04-15T21:58:04.731-0500    connected to: localhost
2019-04-15T21:58:04.743-0500    dropping: ecommerce.CustomerTable
2019-04-15T21:58:05.810-0500    imported 100 documents

C:\Users\javie>"C:\Program Files\MongoDB\Server\4.0\bin\mongoimport.exe" --db ecommerce --type
csv --collection Item --drop --headerline --file "C:\Users\javie\OneDrive\Documents\Academics\SMU
\MSDS 7330 - File Organization and Database Management\Final\Item.csv"
2019-04-15T21:58:47.629-0500    connected to: localhost
2019-04-15T21:58:47.633-0500    dropping: ecommerce.Item
2019-04-15T21:58:48.711-0500    imported 100 documents

C:\Users\javie>"C:\Program Files\MongoDB\Server\4.0\bin\mongoimport.exe" --db ecommerce --type
csv --collection ItemPrice --drop --headerline --file "C:\Users\javie\OneDrive\Documents\Academic
s\SMU\MSDS 7330 - File Organization and Database Management\Final\ItemPrice.csv"
2019-04-15T21:59:15.056-0500    connected to: localhost
2019-04-15T21:59:15.059-0500    dropping: ecommerce.ItemPrice
2019-04-15T21:59:16.132-0500    imported 100 documents

C:\Users\javie>"C:\Program Files\MongoDB\Server\4.0\bin\mongoimport.exe" --db ecommerce --type
csv --collection OrderHeader --drop --headerline --file "C:\Users\javie\OneDrive\Documents\Academ
ics\SMU\MSDS 7330 - File Organization and Database Management\Final\OrderHeader.csv"
2019-04-15T21:59:49.227-0500    connected to: localhost
2019-04-15T21:59:49.230-0500    dropping: ecommerce.OrderHeader
2019-04-15T21:59:50.320-0500    imported 1000 documents

C:\Users\javie>"C:\Program Files\MongoDB\Server\4.0\bin\mongoimport.exe" --db ecommerce --type
csv --collection OrderLine --drop --headerline --file "C:\Users\javie\OneDrive\Documents\Academic
s\SMU\MSDS 7330 - File Organization and Database Management\Final\OrderLine.csv"
2019-04-15T22:00:17.225-0500    connected to: localhost
2019-04-15T22:00:17.229-0500    dropping: ecommerce.OrderLine
2019-04-15T22:00:18.336-0500    imported 1000 documents

C:\Users\javie>"C:\Program Files\MongoDB\Server\4.0\bin\mongoimport.exe" --db ecommerce --type
csv --collection OrderPayment --drop --headerline --file "C:\Users\javie\OneDrive\Documents\Acade
mics\SMU\MSDS 7330 - File Organization and Database Management\Final\OrderPayment.csv"
2019-04-15T22:01:33.371-0500    connected to: localhost
2019-04-15T22:01:33.374-0500    dropping: ecommerce.OrderPayment
2019-04-15T22:01:34.483-0500    imported 1000 documents
```


Query Data from ecommercedb Database in MongoDB for each document

```
> db.Address.find({})
({ "_id" : ObjectId("5cb53fcd781fc84082e5b37"), "AddressID" : 1, "HouseNumber" : 584, "StreetName" : "Royal Lane", "City" : "Dallas", "State" : "TX", "Zip" : 75266 },
{ "_id" : ObjectId("5cb53fcd781fc84082e5b38"), "AddressID" : 2, "HouseNumber" : 810, "StreetName" : "Fallcreek", "City" : "Austin", "State" : "TX", "Zip" : 75277 },
{ "_id" : ObjectId("5cb53fcd781fc84082e5b39"), "AddressID" : 3, "HouseNumber" : 364, "StreetName" : "Airport blvd", "City" : "Houston", "State" : "TX", "Zip" : 75699 },
{ "_id" : ObjectId("5cb53fcd781fc84082e5b3a"), "AddressID" : 5, "HouseNumber" : 198, "StreetName" : "JFK Dr", "City" : "Houston", "State" : "TX", "Zip" : 75876 },
{ "_id" : ObjectId("5cb53fcd781fc84082e5b3b"), "AddressID" : 4, "HouseNumber" : 798, "StreetName" : "Lovers Lane", "City" : "Austin", "State" : "TX", "Zip" : 75419 },
{ "_id" : ObjectId("5cb53fcd781fc84082e5b3c"), "AddressID" : 6, "HouseNumber" : 658, "StreetName" : "Fallcreek", "City" : "Dallas", "State" : "TX", "Zip" : 75402 },
{ "_id" : ObjectId("5cb53fcd781fc84082e5b3d"), "AddressID" : 8, "HouseNumber" : 117, "StreetName" : "Lovers Lane", "City" : "Austin", "State" : "TX", "Zip" : 75688 },
{ "_id" : ObjectId("5cb53fcd781fc84082e5b3e"), "AddressID" : 7, "HouseNumber" : 494, "StreetName" : "Royal Lane", "City" : "Houston", "State" : "TX", "Zip" : 75841 },
{ "_id" : ObjectId("5cb53fcd781fc84082e5b3f"), "AddressID" : 9, "HouseNumber" : 363, "StreetName" : "Royal Lane", "City" : "Dallas", "State" : "TX", "Zip" : 75595 },
{ "_id" : ObjectId("5cb53fcd781fc84082e5b40"), "AddressID" : 10, "HouseNumber" : 192, "StreetName" : "Lovers Lane", "City" : "Austin", "State" : "TX", "Zip" : 75623 },
{ "_id" : ObjectId("5cb53fcd781fc84082e5b41"), "AddressID" : 11, "HouseNumber" : 223, "StreetName" : "Royal Lane", "City" : "Dallas", "State" : "TX", "Zip" : 75832 },
{ "_id" : ObjectId("5cb53fcd781fc84082e5b42"), "AddressID" : 14, "HouseNumber" : 413, "StreetName" : "Fallcreek", "City" : "Dallas", "State" : "TX", "Zip" : 75103 },
{ "_id" : ObjectId("5cb53fcd781fc84082e5b43"), "AddressID" : 20, "HouseNumber" : 559, "StreetName" : "JFK Dr", "City" : "Dallas", "State" : "TX", "Zip" : 75530 },
{ "_id" : ObjectId("5cb53fcd781fc84082e5b44"), "AddressID" : 21, "HouseNumber" : 918, "StreetName" : "Royal Lane", "City" : "Houston", "State" : "TX", "Zip" : 75801 },
{ "_id" : ObjectId("5cb53fcd781fc84082e5b45"), "AddressID" : 22, "HouseNumber" : 953, "StreetName" : "Royal Lane", "City" : "Austin", "State" : "TX", "Zip" : 75535 },
{ "_id" : ObjectId("5cb53fcd781fc84082e5b46"), "AddressID" : 23, "HouseNumber" : 301, "StreetName" : "Lovers Lane", "City" : "Dallas", "State" : "TX", "Zip" : 75403 },
{ "_id" : ObjectId("5cb53fcd781fc84082e5b47"), "AddressID" : 16, "HouseNumber" : 611, "StreetName" : "Fallcreek", "City" : "Austin", "State" : "TX", "Zip" : 75609 },
{ "_id" : ObjectId("5cb53fcd781fc84082e5b48"), "AddressID" : 24, "HouseNumber" : 188, "StreetName" : "Royal Lane", "City" : "Houston", "State" : "TX", "Zip" : 75804 },
{ "_id" : ObjectId("5cb53fcd781fc84082e5b49"), "AddressID" : 25, "HouseNumber" : 910, "StreetName" : "Fallcreek", "City" : "Dallas", "State" : "TX", "Zip" : 75259 },
{ "_id" : ObjectId("5cb53fcd781fc84082e5b4a"), "AddressID" : 27, "HouseNumber" : 719, "StreetName" : "Royal Lane", "City" : "Houston", "State" : "TX", "Zip" : 75604 },
type "it" for more
>
> db.Item.find({})
({ "_id" : ObjectId("5cb544e8d781fc84082e5c0e"), "ItemNumber" : 2, "ItemDescription" : "Description for item 2", "ItemWeight" : "987 lbs", "CountryOfOrigin" : "US", "CostPrice" : 79, "ItemCategory" : "Appliances" },
{ "_id" : ObjectId("5cb544e8d781fc84082e5c0f"), "ItemNumber" : 1, "ItemDescription" : "Description for item 1", "ItemWeight" : "286 lbs", "CountryOfOrigin" : "US", "CostPrice" : 67, "ItemCategory" : "Grocery" },
{ "_id" : ObjectId("5cb544e8d781fc84082e5c10"), "ItemNumber" : 3, "ItemDescription" : "Description for item 3", "ItemWeight" : "754 lbs", "CountryOfOrigin" : "US", "CostPrice" : 66, "ItemCategory" : "Appliances" },
{ "_id" : ObjectId("5cb544e8d781fc84082e5c11"), "ItemNumber" : 4, "ItemDescription" : "Description for item 4", "ItemWeight" : "546 lbs", "CountryOfOrigin" : "US", "CostPrice" : 87, "ItemCategory" : "Grocery" },
{ "_id" : ObjectId("5cb544e8d781fc84082e5c12"), "ItemNumber" : 5, "ItemDescription" : "Description for item 5", "ItemWeight" : "905 lbs", "CountryOfOrigin" : "US", "CostPrice" : 35, "ItemCategory" : "Appliances" },
{ "_id" : ObjectId("5cb544e8d781fc84082e5c13"), "ItemNumber" : 6, "ItemDescription" : "Description for item 6", "ItemWeight" : "286 lbs", "CountryOfOrigin" : "US", "CostPrice" : 96, "ItemCategory" : "Electronics" },
{ "_id" : ObjectId("5cb544e8d781fc84082e5c14"), "ItemNumber" : 7, "ItemDescription" : "Description for item 7", "ItemWeight" : "319 lbs", "CountryOfOrigin" : "US", "CostPrice" : 75, "ItemCategory" : "Electronics" },
{ "_id" : ObjectId("5cb544e8d781fc84082e5c15"), "ItemNumber" : 8, "ItemDescription" : "Description for item 8", "ItemWeight" : "348 lbs", "CountryOfOrigin" : "US", "CostPrice" : 75, "ItemCategory" : "Grocery" },
{ "_id" : ObjectId("5cb544e8d781fc84082e5c16"), "ItemNumber" : 9, "ItemDescription" : "Description for item 9", "ItemWeight" : "325 lbs", "CountryOfOrigin" : "US", "CostPrice" : 60, "ItemCategory" : "Appliances" },
{ "_id" : ObjectId("5cb544e8d781fc84082e5c17"), "ItemNumber" : 12, "ItemDescription" : "Description for item 12", "ItemWeight" : "594 lbs", "CountryOfOrigin" : "US", "CostPrice" : 18, "ItemCategory" : "Grocery" },
{ "_id" : ObjectId("5cb544e8d781fc84082e5c18"), "ItemNumber" : 10, "ItemDescription" : "Description for item 10", "ItemWeight" : "108 lbs", "CountryOfOrigin" : "US", "CostPrice" : 72, "ItemCategory" : "Grocery" },
{ "_id" : ObjectId("5cb544e8d781fc84082e5c19"), "ItemNumber" : 11, "ItemDescription" : "Description for item 11", "ItemWeight" : "257 lbs", "CountryOfOrigin" : "US", "CostPrice" : 25, "ItemCategory" : "Appliances" },
{ "_id" : ObjectId("5cb544e8d781fc84082e5c1a"), "ItemNumber" : 13, "ItemDescription" : "Description for item 13", "ItemWeight" : "703 lbs", "CountryOfOrigin" : "US", "CostPrice" : 19, "ItemCategory" : "Electronics" },
{ "_id" : ObjectId("5cb544e8d781fc84082e5c1b"), "ItemNumber" : 14, "ItemDescription" : "Description for item 14", "ItemWeight" : "253 lbs", "CountryOfOrigin" : "US", "CostPrice" : 82, "ItemCategory" : "Appliances" },
{ "_id" : ObjectId("5cb544e8d781fc84082e5c1c"), "ItemNumber" : 15, "ItemDescription" : "Description for item 15", "ItemWeight" : "865 lbs", "CountryOfOrigin" : "US", "CostPrice" : 80, "ItemCategory" : "Appliances" },
{ "_id" : ObjectId("5cb544e8d781fc84082e5c1d"), "ItemNumber" : 16, "ItemDescription" : "Description for item 16", "ItemWeight" : "954 lbs", "CountryOfOrigin" : "US", "CostPrice" : 74, "ItemCategory" : "Grocery" },
{ "_id" : ObjectId("5cb544e8d781fc84082e5c1e"), "ItemNumber" : 17, "ItemDescription" : "Description for item 17", "ItemWeight" : "865 lbs", "CountryOfOrigin" : "US", "CostPrice" : 11, "ItemCategory" : "Grocery" },
{ "_id" : ObjectId("5cb544e8d781fc84082e5c1f"), "ItemNumber" : 18, "ItemDescription" : "Description for item 18", "ItemWeight" : "846 lbs", "CountryOfOrigin" : "US", "CostPrice" : 59, "ItemCategory" : "Grocery" },
{ "_id" : ObjectId("5cb544e8d781fc84082e5c20"), "ItemNumber" : 22, "ItemDescription" : "Description for item 22", "ItemWeight" : "461 lbs", "CountryOfOrigin" : "US", "CostPrice" : 94, "ItemCategory" : "Grocery" },
type "it" for more
>
> db.CustomerTable.find({})
({ "_id" : ObjectId("5cb544bdd781fc84082e5ba4"), "CustomerID" : 1, "CustomerName" : "Rob Ranger", "Address_AddressID" : 4 },
{ "_id" : ObjectId("5cb544bdd781fc84082e5ba5"), "CustomerID" : 5, "CustomerName" : "Rob Laws", "Address_AddressID" : 97 },
{ "_id" : ObjectId("5cb544bdd781fc84082e5ba6"), "CustomerID" : 3, "CustomerName" : "John Doe", "Address_AddressID" : 29 },
{ "_id" : ObjectId("5cb544bdd781fc84082e5ba7"), "CustomerID" : 6, "CustomerName" : "Ed Ranger", "Address_AddressID" : 48 },
{ "_id" : ObjectId("5cb544bdd781fc84082e5ba8"), "CustomerID" : 2, "CustomerName" : "Bill Francis", "Address_AddressID" : 9 },
{ "_id" : ObjectId("5cb544bdd781fc84082e5ba9"), "CustomerID" : 7, "CustomerName" : "John Doe", "Address_AddressID" : 12 },
{ "_id" : ObjectId("5cb544bdd781fc84082e5baa"), "CustomerID" : 8, "CustomerName" : "Adam Ranger", "Address_AddressID" : 55 },
{ "_id" : ObjectId("5cb544bdd781fc84082e5bab"), "CustomerID" : 9, "CustomerName" : "John Sullivan", "Address_AddressID" : 79 },
{ "_id" : ObjectId("5cb544bdd781fc84082e5bac"), "CustomerID" : 4, "CustomerName" : "John Doe", "Address_AddressID" : 47 },
{ "_id" : ObjectId("5cb544bdd781fc84082e5bad"), "CustomerID" : 16, "CustomerName" : "Ed Laws", "Address_AddressID" : 5 },
{ "_id" : ObjectId("5cb544bdd781fc84082e5bae"), "CustomerID" : 18, "CustomerName" : "Rob Sullivan", "Address_AddressID" : 51 },
{ "_id" : ObjectId("5cb544bdd781fc84082e5baf"), "CustomerID" : 19, "CustomerName" : "Rob Ranger", "Address_AddressID" : 25 },
{ "_id" : ObjectId("5cb544bdd781fc84082e5bb0"), "CustomerID" : 17, "CustomerName" : "Adam Doe", "Address_AddressID" : 7 },
{ "_id" : ObjectId("5cb544bdd781fc84082e5bb1"), "CustomerID" : 20, "CustomerName" : "John Ranger", "Address_AddressID" : 83 },
{ "_id" : ObjectId("5cb544bdd781fc84082e5bb2"), "CustomerID" : 21, "CustomerName" : "Adam Ranger", "Address_AddressID" : 64 },
{ "_id" : ObjectId("5cb544bdd781fc84082e5bb3"), "CustomerID" : 22, "CustomerName" : "Adam Sullivan", "Address_AddressID" : 40 },
{ "_id" : ObjectId("5cb544bdd781fc84082e5bb4"), "CustomerID" : 23, "CustomerName" : "Bill Doe", "Address_AddressID" : 83 },
{ "_id" : ObjectId("5cb544bdd781fc84082e5bb5"), "CustomerID" : 24, "CustomerName" : "Ed Francis", "Address_AddressID" : 82 },
{ "_id" : ObjectId("5cb544bdd781fc84082e5bb6"), "CustomerID" : 25, "CustomerName" : "John Francis", "Address_AddressID" : 40 },
{ "_id" : ObjectId("5cb544bdd781fc84082e5bb7"), "CustomerID" : 12, "CustomerName" : "Bill Ranger", "Address_AddressID" : 20 },
type "it" for more
>
```

```
> db.OrderHeader.find({})
{ "_id" : ObjectId("5cb54526d781fc84082e5ce0"), "OrderNumber" : 1, "OrderDate" : "2017-12-22 00:00:00", "CustomerTable_CustomerID" : 96, "OrderType" : "Sales", "Address_AddressID" : 48 }
{ "_id" : ObjectId("5cb54526d781fc84082e5ce1"), "OrderNumber" : 2, "OrderDate" : "2017-02-04 00:00:00", "CustomerTable_CustomerID" : 44, "OrderType" : "Sales", "Address_AddressID" : 79 }
{ "_id" : ObjectId("5cb54526d781fc84082e5ce2"), "OrderNumber" : 3, "OrderDate" : "2017-08-12 00:00:00", "CustomerTable_CustomerID" : 29, "OrderType" : "Sales", "Address_AddressID" : 76 }
{ "_id" : ObjectId("5cb54526d781fc84082e5ce3"), "OrderNumber" : 5, "OrderDate" : "2017-03-18 00:00:00", "CustomerTable_CustomerID" : 56, "OrderType" : "Sales", "Address_AddressID" : 79 }
{ "_id" : ObjectId("5cb54526d781fc84082e5ce4"), "OrderNumber" : 6, "OrderDate" : "2017-03-09 00:00:00", "CustomerTable_CustomerID" : 69, "OrderType" : "Sales", "Address_AddressID" : 99 }
{ "_id" : ObjectId("5cb54526d781fc84082e5ce5"), "OrderNumber" : 7, "OrderDate" : "2017-03-12 00:00:00", "CustomerTable_CustomerID" : 83, "OrderType" : "Sales", "Address_AddressID" : 78 }
{ "_id" : ObjectId("5cb54526d781fc84082e5ce6"), "OrderNumber" : 4, "OrderDate" : "2017-03-08 00:00:00", "CustomerTable_CustomerID" : 20, "OrderType" : "Sales", "Address_AddressID" : 79 }
{ "_id" : ObjectId("5cb54526d781fc84082e5ce7"), "OrderNumber" : 8, "OrderDate" : "2017-08-07 00:00:00", "CustomerTable_CustomerID" : 8, "OrderType" : "Sales", "Address_AddressID" : 38 }
{ "_id" : ObjectId("5cb54526d781fc84082e5ce8"), "OrderNumber" : 9, "OrderDate" : "2017-05-16 00:00:00", "CustomerTable_CustomerID" : 50, "OrderType" : "Sales", "Address_AddressID" : 34 }
{ "_id" : ObjectId("5cb54526d781fc84082e5ce9"), "OrderNumber" : 10, "OrderDate" : "2017-07-05 00:00:00", "CustomerTable_CustomerID" : 20, "OrderType" : "Sales", "Address_AddressID" : 79 }
{ "_id" : ObjectId("5cb54526d781fc84082e5cea"), "OrderNumber" : 12, "OrderDate" : "2017-01-16 00:00:00", "CustomerTable_CustomerID" : 38, "OrderType" : "Sales", "Address_AddressID" : 100 }
{ "_id" : ObjectId("5cb54526d781fc84082e5ceb"), "OrderNumber" : 11, "OrderDate" : "2017-12-15 00:00:00", "CustomerTable_CustomerID" : 33, "OrderType" : "Sales", "Address_AddressID" : 12 }
{ "_id" : ObjectId("5cb54526d781fc84082e5cec"), "OrderNumber" : 15, "OrderDate" : "2017-06-03 00:00:00", "CustomerTable_CustomerID" : 13, "OrderType" : "Sales", "Address_AddressID" : 84 }
{ "_id" : ObjectId("5cb54526d781fc84082e5ced"), "OrderNumber" : 13, "OrderDate" : "2017-01-22 00:00:00", "CustomerTable_CustomerID" : 78, "OrderType" : "Sales", "Address_AddressID" : 50 }
{ "_id" : ObjectId("5cb54526d781fc84082e5cee"), "OrderNumber" : 14, "OrderDate" : "2017-05-09 00:00:00", "CustomerTable_CustomerID" : 83, "OrderType" : "Sales", "Address_AddressID" : 80 }
{ "_id" : ObjectId("5cb54526d781fc84082e5cef"), "OrderNumber" : 23, "OrderDate" : "2017-05-26 00:00:00", "CustomerTable_CustomerID" : 84, "OrderType" : "Sales", "Address_AddressID" : 87 }
{ "_id" : ObjectId("5cb54526d781fc84082e5cf0"), "OrderNumber" : 10, "OrderDate" : "2017-04-10 00:00:00", "CustomerTable_CustomerID" : 92, "OrderType" : "Sales", "Address_AddressID" : 33 }
{ "_id" : ObjectId("5cb54526d781fc84082e5cf1"), "OrderNumber" : 25, "OrderDate" : "2017-09-11 00:00:00", "CustomerTable_CustomerID" : 27, "OrderType" : "Sales", "Address_AddressID" : 13 }
{ "_id" : ObjectId("5cb54526d781fc84082e5cf2"), "OrderNumber" : 26, "OrderDate" : "2017-10-02 00:00:00", "CustomerTable_CustomerID" : 29, "OrderType" : "Sales", "Address_AddressID" : 67 }
{ "_id" : ObjectId("5cb54526d781fc84082e5cf3"), "OrderNumber" : 27, "OrderDate" : "2017-02-14 00:00:00", "CustomerTable_CustomerID" : 53, "OrderType" : "Sales", "Address_AddressID" : 28 }
type "it" for more
> db.OrderPayment.find({})
{ "_id" : ObjectId("5cb5458ed781fc84082e64ba"), "PaymentId" : 8, "PaymentType" : "CreditCard", "CardNumber" : NumberLong("736294552086728"), "AmountCharged" : 2758, "OrderHeader_OrderNumber" : 8 }
{ "_id" : ObjectId("5cb5458ed781fc84082e64bb"), "PaymentId" : 9, "PaymentType" : "CreditCard", "CardNumber" : NumberLong("952402038989636"), "AmountCharged" : 5834, "OrderHeader_OrderNumber" : 9 }
{ "_id" : ObjectId("5cb5458ed781fc84082e64bc"), "PaymentId" : 10, "PaymentType" : "CreditCard", "CardNumber" : NumberLong("6450126901355931"), "AmountCharged" : 10314, "OrderHeader_OrderNumber" : 10 }
{ "_id" : ObjectId("5cb5458ed781fc84082e64bd"), "PaymentId" : 11, "PaymentType" : "CreditCard", "CardNumber" : NumberLong("9687459372803632"), "AmountCharged" : 8004, "OrderHeader_OrderNumber" : 11 }
{ "_id" : ObjectId("5cb5458ed781fc84082e64be"), "PaymentId" : 12, "PaymentType" : "CreditCard", "CardNumber" : NumberLong("650808617775619"), "AmountCharged" : 7270, "OrderHeader_OrderNumber" : 12 }
{ "_id" : ObjectId("5cb5458ed781fc84082e64bf"), "PaymentId" : 13, "PaymentType" : "CreditCard", "CardNumber" : NumberLong("6443948286397203"), "AmountCharged" : 7950, "OrderHeader_OrderNumber" : 13 }
{ "_id" : ObjectId("5cb5458ed781fc84082e64c0"), "PaymentId" : 14, "PaymentType" : "CreditCard", "CardNumber" : NumberLong("6121739292121567"), "AmountCharged" : 3146, "OrderHeader_OrderNumber" : 14 }
{ "_id" : ObjectId("5cb5458ed781fc84082e64c1"), "PaymentId" : 15, "PaymentType" : "CreditCard", "CardNumber" : NumberLong("9286738143604700"), "AmountCharged" : 12072, "OrderHeader_OrderNumber" : 15 }
{ "_id" : ObjectId("5cb5458ed781fc84082e64c2"), "PaymentId" : 16, "PaymentType" : "CreditCard", "CardNumber" : NumberLong("8397272855081098"), "AmountCharged" : 11014, "OrderHeader_OrderNumber" : 16 }
{ "_id" : ObjectId("5cb5458ed781fc84082e64c3"), "PaymentId" : 17, "PaymentType" : "CreditCard", "CardNumber" : NumberLong("6790127286606767"), "AmountCharged" : 8600, "OrderHeader_OrderNumber" : 17 }
{ "_id" : ObjectId("5cb5458ed781fc84082e64c4"), "PaymentId" : 4, "PaymentType" : "CreditCard", "CardNumber" : NumberLong("5360488591662001"), "AmountCharged" : 2440, "OrderHeader_OrderNumber" : 4 }
{ "_id" : ObjectId("5cb5458ed781fc84082e64c5"), "PaymentId" : 18, "PaymentType" : "CreditCard", "CardNumber" : NumberLong("8960393662908723"), "AmountCharged" : 12975, "OrderHeader_OrderNumber" : 18 }
{ "_id" : ObjectId("5cb5458ed781fc84082e64c6"), "PaymentId" : 19, "PaymentType" : "CreditCard", "CardNumber" : NumberLong("7807438357367336"), "AmountCharged" : 9919, "OrderHeader_OrderNumber" : 19 }
{ "_id" : ObjectId("5cb5458ed781fc84082e64c7"), "PaymentId" : 20, "PaymentType" : "CreditCard", "CardNumber" : NumberLong("7302230338556467"), "AmountCharged" : 2160, "OrderHeader_OrderNumber" : 20 }
{ "_id" : ObjectId("5cb5458ed781fc84082e64c8"), "PaymentId" : 22, "PaymentType" : "CreditCard", "CardNumber" : NumberLong("8350343860914202"), "AmountCharged" : 2574, "OrderHeader_OrderNumber" : 22 }
{ "_id" : ObjectId("5cb5458ed781fc84082e64c9"), "PaymentId" : 21, "PaymentType" : "CreditCard", "CardNumber" : NumberLong("5661670417397937"), "AmountCharged" : 12375, "OrderHeader_OrderNumber" : 21 }
{ "_id" : ObjectId("5cb5458ed781fc84082e64ca"), "PaymentId" : 24, "PaymentType" : "CreditCard", "CardNumber" : NumberLong("8853925379632267"), "AmountCharged" : 12594, "OrderHeader_OrderNumber" : 24 }
{ "_id" : ObjectId("5cb5458ed781fc84082e64cb"), "PaymentId" : 23, "PaymentType" : "CreditCard", "CardNumber" : NumberLong("9964758940487483"), "AmountCharged" : 2882, "OrderHeader_OrderNumber" : 23 }
{ "_id" : ObjectId("5cb5458ed781fc84082e64cc"), "PaymentId" : 25, "PaymentType" : "CreditCard", "CardNumber" : NumberLong("5797463328299096"), "AmountCharged" : 6958, "OrderHeader_OrderNumber" : 25 }
```

```
> db.OrderLine.find({})
{ "_id" : ObjectId("5cb54542d781fc84082e60cd"), "OrderLineNumber" : 1, "OrderHeader_OrderNumber" : 1, "Quantity" : 2, "PricePerQty" : 223, "Item_ItemNumber" : 8 }
{ "_id" : ObjectId("5cb54542d781fc84082e60ce"), "OrderLineNumber" : 2, "OrderHeader_OrderNumber" : 1, "Quantity" : 3, "PricePerQty" : 455, "Item_ItemNumber" : 81 }
{ "_id" : ObjectId("5cb54542d781fc84082e60cf"), "OrderLineNumber" : 3, "OrderHeader_OrderNumber" : 2, "Quantity" : 5, "PricePerQty" : 239, "Item_ItemNumber" : 3 }
{ "_id" : ObjectId("5cb54542d781fc84082e60d0"), "OrderLineNumber" : 4, "OrderHeader_OrderNumber" : 2, "Quantity" : 10, "PricePerQty" : 315, "Item_ItemNumber" : 68 }
{ "_id" : ObjectId("5cb54542d781fc84082e60d1"), "OrderLineNumber" : 5, "OrderHeader_OrderNumber" : 2, "Quantity" : 5, "PricePerQty" : 310, "Item_ItemNumber" : 3 }
{ "_id" : ObjectId("5cb54542d781fc84082e60d2"), "OrderLineNumber" : 6, "OrderHeader_OrderNumber" : 2, "Quantity" : 6, "PricePerQty" : 337, "Item_ItemNumber" : 99 }
{ "_id" : ObjectId("5cb54542d781fc84082e60d3"), "OrderLineNumber" : 7, "OrderHeader_OrderNumber" : 2, "Quantity" : 5, "PricePerQty" : 337, "Item_ItemNumber" : 54 }
{ "_id" : ObjectId("5cb54542d781fc84082e60d4"), "OrderLineNumber" : 8, "OrderHeader_OrderNumber" : 3, "Quantity" : 7, "PricePerQty" : 234, "Item_ItemNumber" : 81 }
{ "_id" : ObjectId("5cb54542d781fc84082e60d5"), "OrderLineNumber" : 9, "OrderHeader_OrderNumber" : 4, "Quantity" : 10, "PricePerQty" : 244, "Item_ItemNumber" : 21 }
{ "_id" : ObjectId("5cb54542d781fc84082e60d6"), "OrderLineNumber" : 13, "OrderHeader_OrderNumber" : 5, "Quantity" : 1, "PricePerQty" : 271, "Item_ItemNumber" : 56 }
{ "_id" : ObjectId("5cb54542d781fc84082e60d7"), "OrderLineNumber" : 18, "OrderHeader_OrderNumber" : 7, "Quantity" : 1, "PricePerQty" : 273, "Item_ItemNumber" : 9 }
{ "_id" : ObjectId("5cb54542d781fc84082e60d8"), "OrderLineNumber" : 19, "OrderHeader_OrderNumber" : 8, "Quantity" : 7, "PricePerQty" : 394, "Item_ItemNumber" : 62 }
{ "_id" : ObjectId("5cb54542d781fc84082e60d9"), "OrderLineNumber" : 20, "OrderHeader_OrderNumber" : 9, "Quantity" : 4, "PricePerQty" : 410, "Item_ItemNumber" : 10 }
{ "_id" : ObjectId("5cb54542d781fc84082e60da"), "OrderLineNumber" : 21, "OrderHeader_OrderNumber" : 9, "Quantity" : 9, "PricePerQty" : 466, "Item_ItemNumber" : 12 }
{ "_id" : ObjectId("5cb54542d781fc84082e60db"), "OrderLineNumber" : 22, "OrderHeader_OrderNumber" : 10, "Quantity" : 8, "PricePerQty" : 204, "Item_ItemNumber" : 99 }
{ "_id" : ObjectId("5cb54542d781fc84082e60dc"), "OrderLineNumber" : 23, "OrderHeader_OrderNumber" : 10, "Quantity" : 6, "PricePerQty" : 457, "Item_ItemNumber" : 7 }
{ "_id" : ObjectId("5cb54542d781fc84082e60dd"), "OrderLineNumber" : 24, "OrderHeader_OrderNumber" : 10, "Quantity" : 5, "PricePerQty" : 375, "Item_ItemNumber" : 1 }
{ "_id" : ObjectId("5cb54542d781fc84082e60de"), "OrderLineNumber" : 25, "OrderHeader_OrderNumber" : 10, "Quantity" : 5, "PricePerQty" : 339, "Item_ItemNumber" : 22 }
{ "_id" : ObjectId("5cb54542d781fc84082e60df"), "OrderLineNumber" : 26, "OrderHeader_OrderNumber" : 10, "Quantity" : 5, "PricePerQty" : 474, "Item_ItemNumber" : 61 }
{ "_id" : ObjectId("5cb54542d781fc84082e60e0"), "OrderLineNumber" : 27, "OrderHeader_OrderNumber" : 11, "Quantity" : 9, "PricePerQty" : 381, "Item_ItemNumber" : 49 }
```

In 1-2 pages, compare and contrast the suitability of MySQL and MongoDB for the eCommerce database. Be sure to identify both advantages and disadvantages of each database.

The eCommerce data is extremely structured and perfectly suited for a relational database (MySQL). The advantages of MySQL are that the relationships are preserved and clearly defined. It is easy to see which table has a relationship to which table. The relationships thus make for querying the data much simpler. In addition to the clear relationships, MySQL allows for line item views across multiple tables. This allows the user to query the data and perform operations for analytical/mathematical purposes. Furthermore, the joins across the data also allow for easier queries which don't require sophisticated SQL.

However, MySQL does have its limitations. Based on the current design, there are some serious constraints in scalability. Currently, the database doesn't allow for differentiation between physical, billing, mailing, and shipping address among other critical aspects. While this is acceptable based on its current state (assuming that is how the business owner wants it), it will pose problems as the business expands. Should the company decide to allow for products to have reviews attached to the products and have those reviews linked to user/customer profiles will add a complexity to the database it is not currently designed to handle. Surely, tables may be added and new relations may be created but as the business grows the complexities of the relationship begin to grow much more complex and sophisticated, which have the ability to make the database extremely messy. The problem with having many relationships across the database mean that a query would have to read in every single table in order to get to the end/designed table/tuple. As a result, MySQL meets the current needs of the database, but it will outgrow its current design eventually as typical ecommerce businesses mature into its desired level.

Scalability is a strong point for MongoDB. Unlike MySQL, MongoDB uses documents to store the tables without preserving the relational constraints. This allows the owner to scale the database by simply adding new "documents" to the database in MongoDB. This enhances the database's versatility but also allows the business to grow horizontally or vertically without having database constraints. The flexibility of MongoDB helps the database also perform at a much more efficient level than MySQL. Whereas traditional MySQL queries read in every join they perform, documents allow the user to bypass the unnecessary data by simply joining data from the desired documents. However, the eCommerce database is considerably small performance isn't at issue at its current state. There are only a handful of tables which may require joins, which essentially translate into nanoseconds in performance difference.

As efficient as MongoDB is and its promise for the future scale, there are downsides to the holding eCommerce in MongoDB is its complexity to perform joins. Considering the current data was imported into different collections using the tables as collections, data redundancy was minimized. However, this unstructured design doesn't allow for standard joins in the data. While manual joins are possible in MongoDB between collections, they must be manually coded into the query and could become complex when attempting to join multiple documents from various collections. As the code becomes sophisticated, the efficiency of MongoDB is sacrificed. Furthermore, as the database continues to grows, we would essentially end up seeing a lot of redundancy in order to preserve the ability to join in the future. While redundancy isn't

necessarily a deal-breaker, the data consumption as the database grows is something that must be considered. Storage space is valuable and expensive and excessive unstructured data means a company must purchase/lease physical space in order to store the data. This is echoed by the fact that a document is limited to 16MB in size. While under its present state the size limit doesn't pose a problem, as the database grows (especially unstructured data) then there is a greater chance of documents getting maximized much quicker. This would only add to the complexity when attempting to join collections/documents into the query.

Ultimately, as the eCommerce database currently stands it is better suited under MySQL. Relationships are preserved and queries are just as effective as they would be in a MongoDB database. The limitations of MySQL don't hinder the current state of affairs at this point, which mean that MySQL would suffice for the eCommerce database. However, if the business begins to grow and collect additional complex data then it would worth considering a migration over to MongoDB. It is important for the owner to recognize the direction in which they want to take the database and prepare for how its going suit its business needs.