

1. Create a new database and update the name to your last name, then connect to it.

```
mysql> create database Benoit;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| Benoit      |
| mysql      |
| performance_schema |
+-----+
4 rows in set (0.00 sec)

mysql> use Benoit;
Database changed
```

- a. In this screenshot I created a database with my last name, had the databases available display on the screen, and set the database in use to the one I created.

2. Create a table called tb2.

```
mysql> create table tb2(user_id VARCHAR(50));
Query OK, 0 rows affected (0.07 sec)

mysql> show tables;
+-----+
| Tables_in_Benoit |
+-----+
| tb2                |
+-----+
1 row in set (0.00 sec)

mysql> describe tb2;
+-----+-----+-----+-----+-----+-----+
| Field | Type      | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| user_id | varchar(50) | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

- a. In this screenshot I created a table and named it tb2. While creating the table I added data to it. Once the table was created with the data added to it I used the command describe to display the information held within the table.
3. Add a second field into the table and describe.

```
mysql> alter table tb2 add newfield VARCHAR(25);
Query OK, 0 rows affected (0.22 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> describe tb2;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| user_id    | varchar(50)   | YES  |     | NULL    |       |
| newfield   | varchar(25)   | YES  |     | NULL    |       |
+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> 
```

- a. In this screenshot I added a new field to the table. After the new field was added I used the command desc describe to display the information held in the table.

## Create a Database

1. In your integrated development environment (IDE), **create a database schema** called QuantigrationRMA. List out the database name. Provide the SQL commands you ran to successfully complete this in your answer, then connect to it:

```
mysql> Create Database QuantigrationRMA;
Query OK, 1 row affected (0.00 sec)

mysql> use QuantigrationRMA;
Database changed
```

- a. In this screenshot I have created a new database called QuantigrationRMA and have changed to use this database. Both queries were successful the commands utilized where the create command to create the database and the use command to change the database to the one I just created.
2. Using the entity relationship diagram (ERD) as a reference, **create** the following **tables** with the appropriate attributes and keys:

- a. A table named **customers** in the QuantigrationRMA database as defined on the project ERD. Provide the SQL commands you ran against MySQL to complete this successfully in your answer:

```
mysql> Create table Customers(CustomerID int not null, FirstName varchar(25), LastName varchar(25), Street varchar(50), City varchar(50), State varchar(25), ZipCode int, Telephone varchar(15), primary key (CustomerID));
Query OK, 0 rows affected (0.04 sec)
```

- i. In this screenshot I have created the first table for this database. The table I created was the Customers table. When creating this table I took care to look at the diagram provided to follow the order that is necessary to have the table in the required layout. Also at the end of the screenshot it is visible that I labeled the CustomerID as the primary key for the table.

There is no foreign key identified in this table.

- b. A table named **orders** in the QuantigrationRMA database as defined on the project ERD. Provide the SQL commands you ran against MySQL to complete this successfully in your answer:

```
mysql> Create table Orders(OrderID int, CustomerID int, SKU varchar(20), Description varchar(50), primary key(OrderID), foreign key(customerID) references Customers(CustomerID));
Query OK, 0 rows affected (0.04 sec)
```

- a. In this screenshot I have created the second table for this database. The table that I created was the Orders table. In this table I took care to follow the order and specifications given in the diagram in regards to type of data and length of characters allowed in each attribute. At the end of this screen shot it is visible that I labeled the OrderID as primary key and CustomerID as the foreign key.
- c. A table named **rma** in the QuantigrationRMA database as defined on the project ERD. Provide the SQL commands you ran against MySQL to complete this successfully in your answer:

```
[107] at line 1
mysql> create table RMA(RMAID int, OrderID int, Step varchar(50), Status varchar(15), Reason varchar(15), primary key(RMAID), foreign key(OrderID) references Orders(OrderID));
Query OK, 0 rows affected (0.05 sec)
```

- i. In this screenshot I created the last of the three tables laid out in the diagram provided. This table is the RMA table. I took care to follow the order provided in the diagram to make sure the data is entered in the order the client wishes. I also labeled the RMAID as the primary key and the OrderID as the foreign key.
1. The importance of a foreign key is that it is a piece of data that connects different tables and this is useful when trying to perform a join operation or any other operation or task that requires comparing different tables.
3. Manually **add 10 records** into the **Customers table**. The data can be made up for now, as you you'll populate all three tables later from the provided CSV files.

```
mysql> Insert into Customers Values(001,'Shiloh','Benoit','123 First Street','New York','New York',10009,'212-524-1234'),
-> (002,'John','Winchester','2350 S Pleasant Valley Road','Winchester','Virginia',22601,'540-667-9111'),
-> (003,'Dean','Winchester','2536 Airline Drive','Bossier City','Louisiana',71111,'318-747-0173'),
-> (004,'Sam','Winchester','201 W Marcy Drive','Big Spring','Texas',79720,'432-267-3363'),
-> (005,'Samwise','Gamjee','537 Johansen Expressway','Fairbanks','Alaska',99701,'907-451-9900'),
-> (006,'Frodo','Baggins','1205 W Ferndon Street','Litchfield','Illinois',62056,'217-324-6195'),
-> (007,'Bilbo','Baggins','258 Post Road','Westerly','Rhode Island',02891,'401-322-0790'),
-> (008,'Teresa','Mendosa','210 Greenville Blvd SW','Greenville','North Carolina',27834,'252-355-2441'),
-> (009,'Pote','N/A','10675 SR-5 S','Centerville','Alabama',35034,'205-926-4878'),
-> (010,'King','George','8580 Rock Ridge Drive','Mountain Iron','Minnesota',55768,'218-305-3314');
Query OK, 10 rows affected (0.02 sec)
Records: 10 Duplicates: 0 Warnings: 0

mysql> select* from Customers
-> ;
```

CustomerID	FirstName	LastName	Street	City	State	ZipCode	Telephone
1	Shiloh	Benoit	123 First Street	New York	New York	10009	212-524-1234
2	John	Winchester	2350 S Pleasant Valley Road	Winchester	Virginia	22601	540-667-9111
3	Dean	Winchester	2536 Airline Drive	Bossier City	Louisiana	71111	318-747-0173
4	Sam	Winchester	201 W Marcy Drive	Big Spring	Texas	79720	432-267-3363
5	Samwise	Gamjee	537 Johansen Expressway	Fairbanks	Alaska	99701	907-451-9900
6	Frodo	Baggins	1205 W Ferndon Street	Litchfield	Illinois	62056	217-324-6195
7	Bilbo	Baggins	258 Post Road	Westerly	Rhode Island	2891	401-322-0790
8	Teresa	Mendosa	210 Greenville Blvd SW	Greenville	North Carolina	27834	252-355-2441
9	Pote	N/A	10675 SR-5 S	Centerville	Alabama	35034	205-926-4878
10	King	George	8580 Rock Ridge Drive	Mountain Iron	Minnesota	55768	218-305-3314

```
10 rows in set (0.00 sec)
```

- a. While adding the ten records into the customer table I made sure that I entered the information in the correct order. The screenshot above shows the code I used to enter the records with each one being separated by hitting enter. This makes the commands/code easier to read and is a good way to check that the order of record is correct. Once I entered the ten records with no error messages I utilized the select\* command to print to screen the information within the table. This verifies that the information took and is in the correct order. Something I did notice was that since I started at CustomerID 001 the leading 0's were not included in the table. I figure that this is due to a formating setting that can be changed prior to creating the table. Which, with the order numbers is not the most important things but one of the zip codes has a leading 0 which could cause issues with billing or shipping down the line.

4. Create a view from the **existing Customers table** by using the SQL command provided below to say "Collaborators." The view should show all instances of "Customer" renamed as "Collaborator."

```
mysql> Create view Collaborator as Select CustomerID As CollaboratorID, FirstName, LastName, Street, City, State, ZipCode, Telephone from Customers;
Query OK, 0 rows affected (0.02 sec)

mysql> describe Collaborator;
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| CollaboratorID | int(11) | NO | | NULL | |
| FirstName | varchar(25) | YES | | NULL | |
| LastName | varchar(25) | YES | | NULL | |
| Street | varchar(50) | YES | | NULL | |
| City | varchar(50) | YES | | NULL | |
| State | varchar(25) | YES | | NULL | |
| ZipCode | int(11) | YES | | NULL | |
| Telephone | varchar(15) | YES | | NULL | |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)

mysql> select* from Collaborator LIMIT 5;
+-----+-----+-----+-----+-----+-----+-----+
| CollaboratorID | FirstName | LastName | Street | City | State | ZipCode | Telephone |
+-----+-----+-----+-----+-----+-----+-----+
| 1 | Shiloh | Benoit | 123 First Street | New York | New York | 10009 | 212-524-1234 |
| 2 | John | Winchester | 2250 S Pleasant Valley Road | Winchester | Virginia | 22601 | 540-667-9111 |
| 3 | Dean | Winchester | 2536 Airline Drive | Bossier City | Louisiana | 71111 | 318-747-0173 |
| 4 | Sam | Winchester | 201 W Marcy Drive | Big Spring | Texas | 79720 | 432-267-3363 |
| 5 | Samwise | Gamjee | 537 Johansen Expressway | Fairbanks | Alaska | 99701 | 907-451-9900 |
+-----+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)
```

- a. In this screen shot I utilized the command create view. In this instance I kept all of the information within the Customer table but changed the table name to Collaborator and set CustomerID to CollaboratorID. Other than that I set the view up to take all of the other information from Customers and in the same order. I then used the select\* function to print to screen the collaborator view with a limit of 5. The limit of 5 limits the print to screen to the first 5 in the table.

1. **Import the data from each file into tables.**

```
Database changed
mysql> Load Data INFILE '/home/codio/workspace/customers.csv' INTO TABLE Customers FIELDS TERMINATED BY ',' LINES TERMINATED BY '\r\n';
Query OK, 37994 rows affected (0.33 sec)
Records: 37994 Deleted: 0 Skipped: 0 Warnings: 0

mysql> Load Data INFILE '/home/codio/workspace/orders.csv' INTO TABLE Orders FIELDS TERMINATED BY ',' LINES TERMINATED BY '\r\n';
Query OK, 37994 rows affected (0.34 sec)
Records: 37994 Deleted: 0 Skipped: 0 Warnings: 0

mysql> show tables;
+-----+
| Tables_in_QuantigrationRMA |
+-----+
| Collaborator                |
| Customers                   |
| Orders                      |
| RNA                         |
+-----+
4 rows in set (0.00 sec)

mysql> Load Data INFILE '/home/codio/workspace/rma.csv' INTO TABLE RNA FIELDS TERMINATED BY ',' LINES TERMINATED BY '\r\n';
Query OK, 38162 rows affected (0.37 sec)
Records: 38162 Deleted: 0 Skipped: 0 Warnings: 0
```

- A. In the above screenshot shows that that I utilized the load data infile command to load the files into the appropriate tables. I made sure I specified the correct tables. I also stated how the lines and fields were terminated even though it was a CSV file as the system still needs to be told the format of the file. I also used the command show tables to make sure that I identified the tables properly. It is not shown in this screenshot but prior to this step I changed the database to QuantigrationRMA.

2. **Write basic queries** against imported tables to organize and analyze targeted data.

```
mysql> Select count(Orders.OrderID) FROM Orders INNER JOIN Customers ON Orders.orderID = Customers.CustomerID WHERE Customers.City = 'Framingham' AND Customers.State = 'Massachusetts';
+-----+
| count(Orders.OrderID) |
+-----+
| 199 |
+-----+
1 row in set (0.04 sec)
```

- a. In the above screenshot I utilized a select count command and an inner join to count the number of customers from Framingham, Massachusetts. At the end of this command I had to identify the city and state separately and state that the state and city was being drawn from the customer table.

```
mysql> Select count(Customers.CustomerID) From Customers where Customers.City = 'Woonsocket';
+-----+
| count(Customers.CustomerID) |
+-----+
| 7 |
+-----+
1 row in set (0.01 sec)
```

```
mysql> select status, step from RMA WHERE OrderID = 5175;
+-----+-----+
| status | step |
+-----+-----+
| Pending | Awaiting customer Documentation |
+-----+-----+
1 row in set (0.00 sec)

mysql> update RMA set status="Complete",step="Credit Customer Account" where OrderID = 5175;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1 Changed: 1 Warnings: 0
```

```
mysql> DELETE FROM RMA WHERE reason='Rejected';
Query OK, 596 rows affected (0.08 sec)
```

A.

B. Write an SQL query to select all of the customers located in the state of Massachusetts.

```
mysql> SELECT count(Customers.CustomerID) FROM Customers WHERE State='Massachusetts';
+-----+
| count(Customers.CustomerID) |
+-----+
| 982 |
+-----+
1 row in set (0.00 sec)
```

- a. In the above screenshot I chose not to perform an inner join operation as I felt it was unneeded to obtain the correct answer. I used a select count command, Identified the specific attribute and which table to get it from. I then set the attribute of state to Massachusetts. There were a total of 982 records returned.

```
mysql> SELECT concat(FirstName, ' ', LastName) FROM Customers WHERE State='Massachusetts';
+-----+
| concat(FirstName, ' ', LastName) |
+-----+
| Donna Hanson |
| Micheal Webster |
| Harvey Cisneros |
| Irma Kemp |
| Todd Bishop |
| Roberta Roman |
+-----+
```

- a. In the above screenshot I utilized the command concat and identified I wanted the first and last name separated by a space. I then stated the table and attribute I wanted to be identified for the list. The list is comprised of all the first and last names of every customer



with Massachusetts as the state in their address. This query resulted in the 982 records printed in the table.

C. Write an SQL query to insert four new records into the Orders and Customers tables using the following data:

i. Customers Table

CustomerID	FirstName	LastName	StreetAddress	City	State	ZipCode	Telephone
100004	Luke	Skywalker	17 Maiden Lane	New York	NY	10222	212-555-1234
100005	Winston	Smith	128 Sycamore Street	Greensboro	NC	27401	919-555-6623
100006	MaryAnne	Jenkins	2 Coconut Way	Jupiter	FL	33458	321-555-8907
100007	Janet	Williams	58 Redondo Beach Blvd	Torrence	CA	90501	310-555-5678

[Insert screenshot and brief explanation here.]

ii. Orders Table

OrderID	CustomerID	SKU	Description
1204305	100004	ADV-24-10C	Advanced Switch 10GigE Copper 24 port
1204306	100005	ADV-48-10F	Advanced Switch 10 GigE Copper/Fiber 44 port copper 4 port fiber
1204307	100006	ENT-24-10F	Enterprise Switch 10GigE SFP+ 24 Port
1204308	100007	ENT-48-10F	Enterprise Switch 10GigE SFP+ 48 port

```
mysql> INSERT INTO Customers VALUES(100004,'Luke','Skywalker','17 Maiden Lane','New York','New York',10222,'212-555-1234'),
-> (100005,'Winston','Smith','128 Sycamore Street','Greensboro','North Carolina',27401,'919-555-6623'),
-> (100006,'MaryAnne','Jenkins','2 Coconut Way','Jupiter','Florida',33456,'321-555-8907'),
-> (100007,'Janet','Williams','58 Redondo Beach Blvd','Torrance','CA',90501,'310-555-5678'));
ERROR 1064 (42000): You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near '
(100007,'Janet','Williams','58 Redondo Beach Blvd','Torrance','CA',90501,'310-' at line 3
mysql> INSERT INTO Customers VALUES(100004,'Luke','Skywalker','17 Maiden Lane','New York','New York',10222,'212-555-1234'), (100005,'Winston','Smith','128 Sycamore Street',
'Greensboro','North Carolina',27401,'919-555-6623'), (100006,'MaryAnne','Jenkins','2 Coconut Way','Jupiter','Florida',33456,'321-555-8907'), (100007,'Janet','Williams','58
Redondo Beach Blvd','Torrance','CA',90501,'310-555-5678'));
Query OK, 4 rows affected (0.02 sec)
Records: 4 Duplicates: 0 Warnings: 0

mysql> INSERT INTO Orders VALUES(1204305,100004,'ADV-24-10C','Advanced Switch 10GigE Copper 24 port'),
-> (1204306,100005,'ADV-48-10F','Advanced 10 GigE Copper/Fiber 48 port copper 4 port fiber'),
-> (1204307,100006,'ENT-24-10F','Enterprise Switch 10GigE SFP+ 48 port'),
-> (1204308,100007,'ENT-48-10F','Enterprise Switch 10GigE SFP+ 48 port');
Query OK, 4 rows affected, 1 warning (0.02 sec)
Records: 4 Duplicates: 0 Warnings: 1
```

- A. In the above screenshot I entered the two tables of data requested. When I went to enter the data into the customer table I made a syntax error. When I was placing a comma and hitting enter after every line so it was neater I accidentally hit a period which threw the syntax error. I was able to correct the error and the query was then successful. I then utilized the same command of insert into values and wrote out the requested information in order separated by comas. I also separated each entry with parentheses, comas and new lines.

- D. In the Customers table, perform a query to count all records where the city is Woonsocket, Rhode Island.

- i. How many records are in the customers table where the field “city” equals “Woonsocket”?

```
mysql> Select count(Customers.CustomerID) From Customers where Customers.City = 'Woonsocket';
+-----+
| count(Customers.CustomerID) |
+-----+
|                               7 |
+-----+
1 row in set (0.01 sec)
```

- A. In the above screenshot I used a select count command and identified the attribute I wish to be counted as customers city and set it to Woonsocket. I did not feel as it was necessary to identify the state as a native Rhode Islander I do not think there is another city called Woonsocket.

- E. In the RMA database, update a customer’s records.

- i. Write an SQL statement to select the current fields of **status** and **step** for the record in the **rma** table with an **orderid** value of “5175.”

1. What are the current status and step?

```
mysql> select status, step from RMA WHERE OrderID = 5175;
+-----+-----+
| status | step                               |
+-----+-----+
| Pending | Awaiting customer Documentation |
+-----+-----+
1 row in set (0.00 sec)

mysql> update RMA set status="Complete",step="Credit Customer Account" where OrderID = 5175;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

A. In the above screenshot I utilized a select command to look up the status and step within the RMA table for OrderID 5175. The current status of the order was pending and the step was awaiting customer documentation.

- ii. Write an SQL statement to update the **status** and **step** for the **OrderID**, 5175 to **status** = "Complete" and **step** = "Credit Customer Account."

1. What are the updated **status** and **step** values for this record? Provide a screenshot of your work.

```
mysql> select status, step from RMA WHERE OrderID = 5175;
+-----+-----+
| status | step                               |
+-----+-----+
| Pending | Awaiting customer Documentation |
+-----+-----+
1 row in set (0.00 sec)

mysql> update RMA set status="Complete",step="Credit Customer Account" where OrderID = 5175;
Query OK, 1 row affected (0.02 sec)
Rows matched: 1  Changed: 1  Warnings: 0
```

A. At the above screenshot at the bottom of the screenshot I utilized the update command and identified the RMA table as the table I wanted to use. I then stated to set the status to complete and the step to credit customer account. At the end of the screenshot I then stated where the OrderID was 5175 as that was the record in need of change. The response was one row was matched and changed.

F. Delete RMA records.

- i. Write an SQL statement to delete all records with a reason of "Rejected."

1. How many records were deleted? Provide a screenshot of your work.

```
mysql> DELETE FROM RMA WHERE reason='Rejected';
Query OK, 596 rows affected (0.08 sec)
```

- A. In the above screenshot I utilized the delete from command and identified the RMA table as the table I want to delete from. I then identified the reason on the records as rejected. When the query ran it resulted as 596 rows affected.

3. **Create an output file** of the required query results.

- A. Write an SQL statement to list the contents of the orders table and send the output to a file with a .csv extension.

```
mysql> select* from Orders into OUTFILE'Orders2.csv' FIELDS TERMINATED BY ',' LINES TERMINATED BY '\r\n';
Query OK, 37998 rows affected (0.04 sec)
```

- A. In the above screenshot I utilized a select\* command and identified the table Orders as the table I wish to select from. I then utilized the command OUTFILE and named the file to be created as Orders2.csv. After naming the file I stated that fields are terminated by comas and lines are terminated by new lines. The query was successful with the entire table being sent to an outfile.

- B. **Analyze the data you've been provided with to identify themes:**

- a. Which parts are being replaced most?

```
mysql> SELECT repair as Repair, COUNT(*) AS NumberOfRepairs FROM PartsMaintenance GROUP BY Repair ORDER BY count(*)DESC;
+-----+-----+
| Repair                | NumberOfRepairs |
+-----+-----+
| Fuel tank              | 95              |
| Tire repair            | 74              |
| Tire replacement       | 66              |
| Windshield replacement | 63              |
| Battery replacement     | 56              |
| Wheel Arch             | 55              |
| Fender replacement     | 54              |
| Rocker Panel           | 53              |
| Brake line replacement | 52              |
| Struts                 | 51              |
| Cab corner panel       | 49              |
| Shocks                 | 47              |
| Dent Repair Left Fender | 37              |
| Transmission           | 28              |
| Dent Repair Rear       | 25              |
+-----+-----+
15 rows in set (0.01 sec)
```

The above screenshot shows all of the repairs performed by this company in descending order. This shows that the most repaired part is the fuel tank. The fuel tank has been repaired 95 times.

- b. Is there a region of the country that experiences more part failures and replacements than others?
- i. Identify region:

Mid West

```
mysql> SELECT Repair, COUNT(*) as NumberofRepairs FROM PartsMaintenance
Where State IN('ND','SD','KS','NE','MN','WI','IA','MO','MI','IN','IL',
', 'OH')Group By Repair Order By count(*)DESC;
```

Repair	NumberofRepairs
Fule tank	48
Rocker Panel	32
Wheel Arch	21
Fender replacement	17
Struts	17
Tire replacement	17
Windshield replacement	15
Battery replacment	15
Shocks	14
Cab corner panel	13
Tire repair	13
Dent Repair Left Fender	11
Brake line replacement	11
Dent Repair Rear	10
Transmission	6

15 rows in set (0.00 sec)

## North East

```
mysql> SELECT Repair, COUNT(*) as NumberofRepairs FROM PartsMaintenance
Where State IN('PA','NJ','NY','CT','RI','MA','VT','NH','OH')Group By
Repair Order By count(*)DESC;
```

Repair	NumberofRepairs
Wheel Arch	34
Cab corner panel	32
Fule tank	28
Struts	24
Brake line replacement	21
Rocker Panel	15
Shocks	14
Fender replacement	13
Tire repair	6
Tire replacement	5
Battery replacment	5
Windshield replacement	5
Dent Repair Left Fender	4
Transmission	4
Dent Repair Rear	3

15 rows in set (0.00 sec)

## South East

```
mysql> SELECT Repair, COUNT(*) as NumberofRepairs FROM PartsMaintenance
e Where State IN('AR','LA','MS','AL','GA','FL','KY','TN','SC','NC','VA
','WV','DE','MD')Group By Repair Order By count(*)DESC;
```

Repair	NumberofRepairs
Battery replacment	22
Tire repair	21
Tire replacement	21
Shocks	18
Fule tank	17
Fender replacement	13
Transmission	13
Windshield replacement	13
Brake line replacement	10
Dent Repair Left Fender	9
Struts	9
Wheel Arch	6
Dent Repair Rear	5
Rocker Panel	5
Cab corner panel	4

15 rows in set (0.00 sec)

#### South West

```
mysql> SELECT Repair, COUNT(*) as NumberofRepairs FROM PartsMaintenance
Where State IN('AZ','NM','TX','OK')Group By Repair Order By count(*)
DESC;
```

Repair	NumberofRepairs
Tire repair	21
Tire replacement	12
Windshield replacement	9
Transmission	6
Battery replacment	6
Dent Repair Rear	4
Dent Repair Left Fender	4
Wheel Arch	1

8 rows in set (0.00 sec)

#### West

```
mysql> SELECT Repair, COUNT(*) as NumberofRepairs FROM PartsMaintenance
Where State IN('WA','ID','MT','OR','WY','CO','UT','NV','CA')Group By
Repair Order By count(*)DESC;
```

Repair	NumberofRepairs
Windshield replacement	21
Tire repair	13
Tire replacement	10
Dent Repair Left Fender	9
Battery replacment	8
Dent Repair Rear	3
Fender replacement	1
Wheel Arch	1

8 rows in set (0.00 sec)

Above are screenshots of all of the repairs in a decending order for each region as identified in the provided documentation. This breaks down repairs from each region. By reviewing these tables the regions with the most repairs are the Mid West and North



East. The Mid West has the most repairs by total numbers of repairs with the most common repair being the fuel tank.

ii. How might the fleet maintenance team use the information to update its maintenance schedule?

1. The way that a fleet maintenance team could utilize the above tables is by reviewing them and making sure the most common repairs per region have the parts on stand by. Also they can provide greater resources in manpower and monetary concerns to the Mid West and North East as they are the busiest regions for this company.

c. Which parts are being replaced most due to corrosion or rust?

```
mysql> Select repair as Repair, Count(*) as NumberofRepairs From Parts
Maintenance Where Reason IN('Rust','Corrosion') Group By Repair Order
by count(*)DESC;
```

Repair	NumberofRepairs
Wheel Arch	55
Fender replacement	54
Rocker Panel	53
Brake line replacement	52
Struts	51
Cab corner panel	49
Shocks	47
Fule tank	46

8 rows in set (0.00 sec)

The above screenshot shows the list of repairs performed with the reason listed as rust or corrosion. The query resulted in the Wheel Arch being repaired the most for these reasons.

- d. Which parts are being replaced most because of mechanical failure or accident, like a flat tire or rock through the windshield?

```
mysql> select repair as Repair, Count(*) as NumberofRepairs from PartsMaintenance where Reason IN('Hole','Collision','Crack','not repairable') group by Repair order by count(*)DESC;
```

Repair	NumberofRepairs
windshield replacement	63
Fule tank	49

2 rows in set (0.01 sec)

In the above screenshot I ran a query that matched all reasons for the reasons closest to accident or mechanical failure as those terms were not used in the reasons section of the table. It resulted in windshield being the most common part being replaced for these reasons.

**C. Write a brief summary of your analysis that takes the information from Step 1 and presents it in a way that nontechnical stakeholders can understand.**

- a. When reviewing all of this data it can seem like a lot. When we are able to break all of it down into tables and organize the numbers it gets much clearer. The busiest areas that would need the most resources are the Mid West and North East. It also shows that the most common issues that are ran into in those areas are rust or corrosion damage. Also the table that displays the list of repairs due to rust show that the lowest parts of the vehicle that are exposed; wheel arch, fender, rocker panel, etc. are the parts that rust targets the most. In the areas such as the "Rust Belt" in the Mid West should be stocked with these parts especially during the winter when the roads are salted which enhance the risk of and damage from rust.

**D. Outline the approach that you took to conduct the analysis.**

- a. What queries did you use to identify trends or themes in the data?
- i. The queries that I utilized to complete this analysis are all visible at the tops of each of the screenshots. I utilized count queries with what queries to create custom tables that answer the questions posed. I wrote queries that listed all of

the items in a descending order by utilizing DESC to make sure the results showed the highest number to answer the question. I then did not limit them to show the amount of repairs for each query for all applicable answers.

- b. What are the benefits of using these queries to retrieve the information in a way that allows you to provide valuable information to your stakeholders?
  - i. The benefit of utilizing queries to retrieve information is that you can take a large file with a lot of information and create custom tables or pull out a specific answer from the table. It takes the analysis or sorting out of the users hands and makes the system do it so that it is faster. It also gives a user the ability to sort the data into tables of importance for different purposes.

**E. Explain how the functions in the analysis tool allowed you to organize the data and retrieve records quickly.**

- a. The quick analysis tool allows you to quickly and easily format your data into either a chart, table, or other organizational format.

**A. Sales by Region.**

- i.

```
mysql> Select State, COUNT(*) From Customers GROUP BY State ORDER BY count(*) DESC LIMIT 3;
+-----+-----+
| State      | COUNT(*) |
+-----+-----+
| Massachusetts | 982 |
| Arkansas    | 854 |
| West Virginia | 843 |
+-----+-----+
3 rows in set (0.02 sec)
```

The above screenshot shows the result of a query to find the state with the most customers. The results of the query was Massachusetts has the most customers with 982 individual customers.

ii.

```
mysql> Select SKU, Description From Orders o Join Customers c ON o.CustomerID = c.CustomerID GROUP BY o.SKU ORDER BY count(*) DESC LIMIT 3;
```

SKU	Description
BAS-48-1 C	Basic Switch 10/100/1000 BaseT 48 port
ENT-48-48F	Enterprise Switch 40GigE SFP+ 48 port
ENT-48-18F	Enterprise Switch 10GigE SFP+ 48 port

3 rows in set (0.03 sec)

The above screenshot shows the result of a query to find what is the top three items sold. The query created a custom table with the results. The most common sold item is BAS-48-1 C Basic Switch 10/100/1000 BaseT 48 Port.

iii.

```
mysql> Select SKU, Description FROM Orders o JOIN Customers c ON o.CustomerID = c.CustomerID WHERE c.State IN('Virginia','North Carolina','South Carolina','Georgia') GROUP BY o.SKU ORDER BY count(*) DESC LIMIT 3;
```

SKU	Description
BAS-48-1 C	Basic Switch 10/100/1000 BaseT 48 port
ENT-48-48F	Enterprise Switch 40GigE SFP+ 48 port
BAS-08-1 C	Basic Switch 10/100/1000 BaseT 8 port

3 rows in set (0.08 sec)

The above screenshot shows the result of a query to find the top three items sold in the south east. To perform this I had to identify the states in the query as in the table the customer table has the address for the individual customers. So, singling out the states to identify the region of the south east was the easiest and fastest way to perform this query.

## B. Returns by region.

```
mysql> Select SKU, Description FROM RMA r JOIN Orders o ON r.OrderID = o.OrderID JOIN Customers c ON o.CustomerID = c.CustomerID GROUP BY o.SKU ORDER BY count(*) DESC LIMIT 3;
```

SKU	Description
BAS-48-1 C	Basic Switch 10/100/1000 BaseT 48 port
ENT-48-48F	Enterprise Switch 40GigE SFP+ 48 port
ENT-48-18F	Enterprise Switch 10GigE SFP+ 48 port

3 rows in set (0.05 sec)

i.

The above screen shot shows the result of the a query to find the three top returned items throughout the United States. The result of the query shows that the most returned item is the BAS-48-1 C Basic Switch 10/100/1000 BaseT 48 port.

ii.

```
mysql> select SKU, Description FROM RMA r JOIN Orders o ON r.OrderID = o.OrderID JOIN Customers c ON o.CustomerID = c.CustomerID WHERE c.State IN('Washington','Oregon','Idaho','Montana') GROUP BY o.SKU ORDER BY c.Count(*) DESC LIMIT 3;
```

SKU	Description
BAS-48-1 C	Basic Switch 10/100/1000 BaseT 48 port
EXT-48-48P	Enterprise Switch 400igs SFP+ 48 port
BAS-48-1 C	Basic Switch 10/100/1000 BaseT 48 port

3 rows in set (0.47 sec)

The above screenshot shows the results of a query that identifies the top three items returned in the north west. To identify the region I had to identify the individual states to make up the region as the customer table only has state listed not region. I had to perform a join operation to connect the RMA, Orders, and customer tables to properly perform this query.

3) Write a report.

The point of all of the queries in part A is to show the most sold items by the client. The point of utilizing SQL to do this is to take the large files and tons of data and make it into manageable pieces. The results of these queries showed us that the state with the most customers is Massachusetts. This can tell you a few things. You can look at this from a marketing standpoint and figure out what the marketing team over Massachusetts is doing that is working and implement that over the other states to help grow the business. It can also tell you that the biggest need for a new warehouse to ship items faster to large portion of your customer base should be placed somewhere in or near Massachusetts. Also, the queries in part A shows us that the most purchased item is the BAS-48-1 C Basic Switch 10/100/1000 BaseT

48 Port. This tells us that it would be a good idea to keep that item on hand or atleast a larger stock of the items as it is being purchased often. This last query that was run in part A was to check the top sellers in the south eastern states this company operates in. It tells us that the top seller the BAS-48-1 C Basic Switch 10/100/1000 BaseT 48 Port is the same but the second and third top sellers are different. This tells us that the BAS-48-1 C Basic Switch 10/100/1000 BaseT 48 Port is a very popular item that is sold in may places. It also tells us that each region might have the need for different items.

In part B we ran queries to analyze the returns by this company. To perform these queries I had to utilize a join operation between all three of the tables to help identify the most returned items. This information is important as it can help identify possible problems with items. The most returned item in the entirety of the US is the BAS-48-1 C Basic Switch 10/100/1000 BaseT 48 Port. This is interesting since it is the most bought item. This could lead to possible over ordering or an issue with the supplier of the item. If the item that is being shipped to customers is defective this could be an issue that could cause a company to lose customers. Also the second section of part B was to identify the most returned item in the north west. To perform this query I had to identify the states in the north west I wanted to pull data from as there is no attribute for region in any of the tables for me to organize the data from. The result of this query showed us that the most returned item in the north west was the BAS-48-1 C Basic Switch 10/100/1000 BaseT 48 Port. Again, this can point to an issue with a supplier. As this is the most purchased item in the US this is a very important item to make sure is functional and the

proper item. With the information gained from these queries I would evaluate the shipping and the manufacturers of the BAS-48-1 C Basic Switch 10/100/1000 BaseT 48 Port to make sure that a functional item is being shipped and to make sure that the merchandise is not being destroyed during transport. I would also look into the customers that are ordering them. This could be as simple as over ordering and people returning the left over items they did not use. This could be fixed by changing the size of package when customers purchase this item. For example, if the switch comes in a ten pack and people are returning 5 of them each time I would recommend changing the pack size to five to limit the returns.

**4. Navigate to your online integrated development environment (IDE). List and record the SQL commands that you used to complete this step here:**

- A. To begin this assignment I accessed the Codio Lab through the Codio tab in the module menu. After I accessed Codio I used the mysql command in the terminal. I do not have a screenshot of this step.

**5. Create a database schema called QuantigrationUpdates. List out the database name. Provide the SQL commands you ran against MySQL to successfully complete this in your answer:**

```
mysql> Create Database Quantigrati
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| Benoit |
| QuantigratiRMA |
| QuantigratiUpdates |
| classicmodels |
| mysql |
| performance_schema |
+-----+
7 rows in set (0.00 sec)

mysql> use QuantigratiUpdates;
Database changed
mysql> 
```

A.

In the above screenshot I utilized the Create database function to create a database named QuantigratiUpdates. I then used the show database command to print to screen all of the available databases. Finally I used the use command to change the database in use to QuantigratiUpdates.

**Formatted:** List 2, Left, Indent: Left: 1", Line spacing: Double

## Security and Tool Discussion

Due to the requirements of the client which were the need to only access the database when at work and not on the go due to the type of business they are in moving the database to a non-SQL database does not make sense. The security risk of putting a database online through a cloud service or database like MongoDB does not make sense. I was able to condense the artifact and show emphasis on the tools that are being used by the client to make their database work for them.