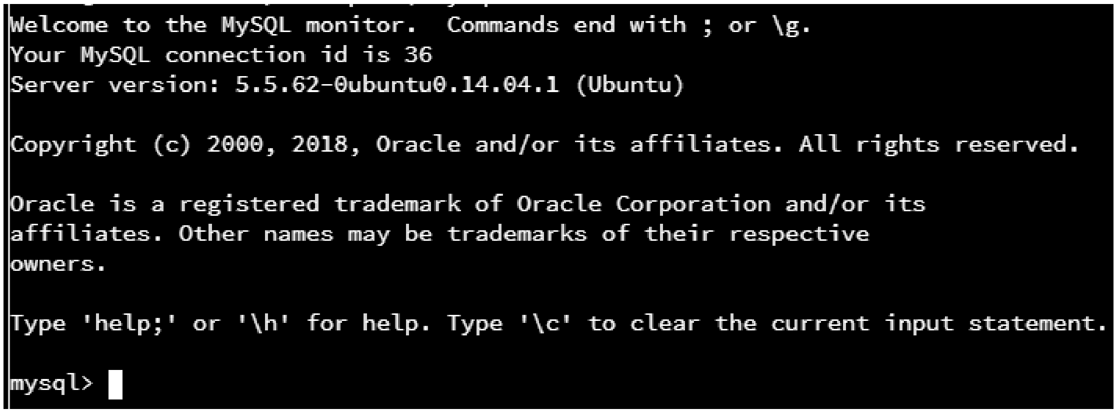
Introduction

Quantigration, a technology solutions provider, faced challenges with scattered and inconsistent order and customer records, making data analysis and reporting challenging. To address this, a relational database was designed and implemented to streamline customer interactions, order tracking, and return management.

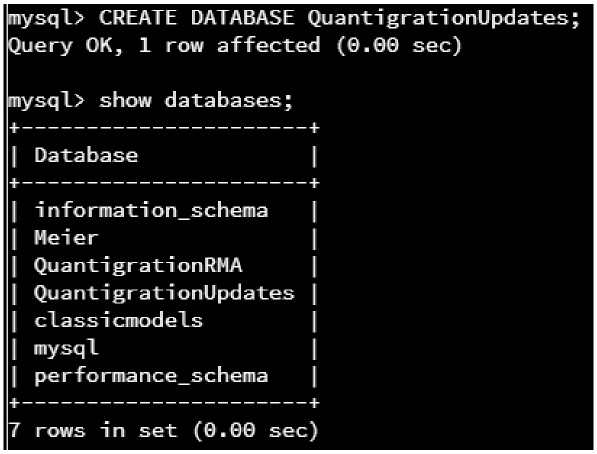
This project showcases the creation, optimization, and querying of the QuantigrationUpdates database, demonstrating core SQL competencies such as data modeling, schema updates, data import, filtering, and advanced querying. The solution enhances data integrity, ensures referential consistency, and provides actionable insights through targeted queries. The project's deliverables include a well-structured database, key performance queries, and the automation of essential business processes.

Step 1: Create a Database



```
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 36
Server version: 5.5.62-0ubuntu0.14.04.1 (Ubuntu)

Copyright (c) 2000, 2018, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> 
```

CREATE DATABASE QuantigrationUpdates;
SHOW DATABASES;

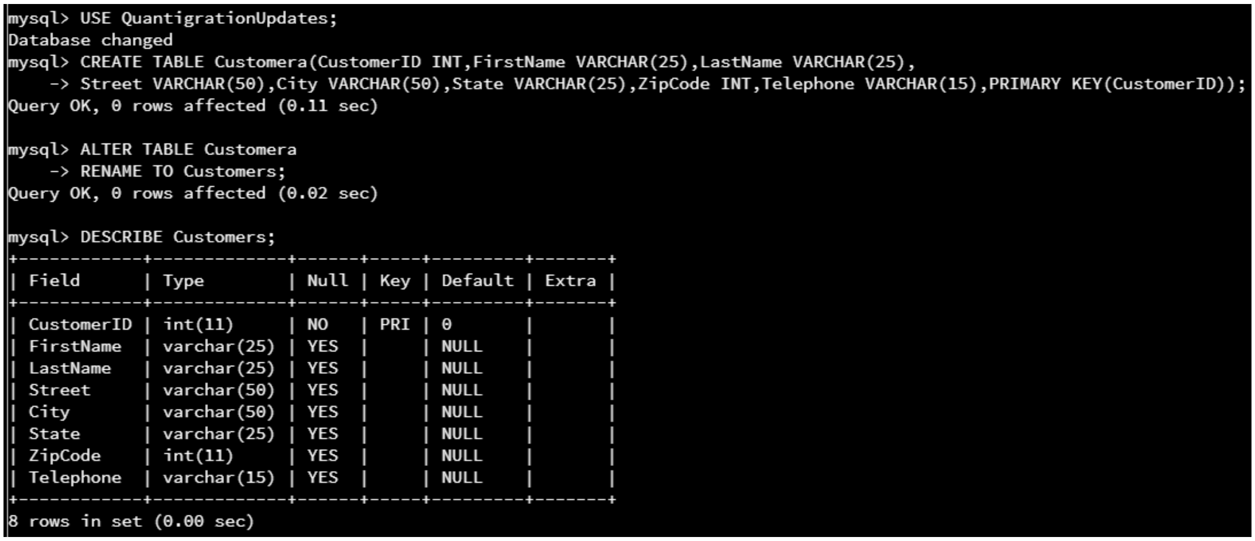


```
mysql> CREATE DATABASE QuantigrationUpdates;
Query OK, 1 row affected (0.00 sec)

mysql> show databases;
+----------------------+
| Database             |
+----------------------+
| information_schema   |
| Meier                |
| QuantigrationRMA     |
| QuantigrationUpdates |
| classicmodels        |
| mysql                |
| performance_schema   |
+----------------------+
7 rows in set (0.00 sec)
```

Creating Tables

Customers Table

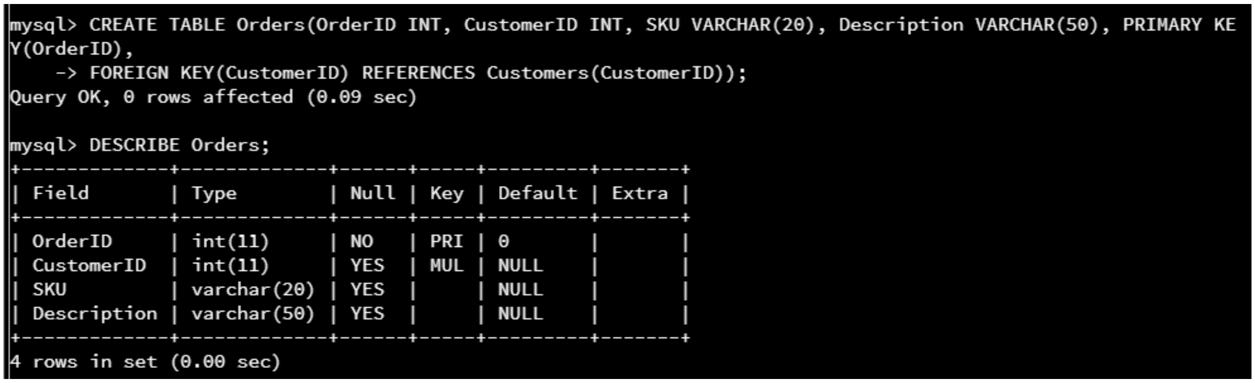
```
CREATE TABLE Customers (
   CustomerID INT PRIMARY KEY AUTO_INCREMENT,
   FirstName VARCHAR(50) NOT NULL,
   LastName VARCHAR(50) NOT NULL,
   StreetAddress VARCHAR(255) NOT NULL,
   City VARCHAR(100) NOT NULL,
   State CHAR(2) NOT NULL,
   ZipCode CHAR(5) NOT NULL,
   Telephone VARCHAR(15) UNIQUE NOT NULL,
   CreatedAt TIMESTAMP DEFAULT CURRENT_TIMESTAMP
);
```



```
mysql> USE QuantigrationUpdates;
Database changed
mysql> CREATE TABLE Customera(CustomerID INT,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 (0.11 sec)

mysql> ALTER TABLE Customera
    -> RENAME TO Customers;
Query OK, 0 rows affected (0.02 sec)

mysql> DESCRIBE Customers;
+------------+-------------+------+-----+---------+-------+
| Field      | Type        | Null | Key | Default | Extra |
+------------+-------------+------+-----+---------+-------+
| CustomerID | int(11)     | NO   | PRI | 0       |       |
| 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)
```

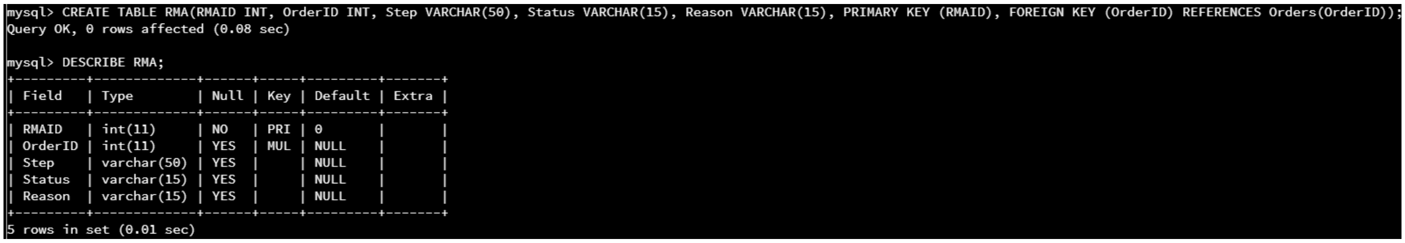
Orders Table

```
CREATE TABLE Orders (
   OrderID INT PRIMARY KEY AUTO_INCREMENT,
   CustomerID INT NOT NULL,
   SKU VARCHAR(20) NOT NULL,
   Description TEXT NOT NULL,
   OrderDate TIMESTAMP DEFAULT CURRENT_TIMESTAMP,
   FOREIGN KEY (CustomerID) REFERENCES Customers(CustomerID) ON DELETE CASCADE
);
CREATE INDEX idx_customer_id ON Orders(CustomerID);
```



```
mysql> CREATE TABLE Orders(OrderID INT, CustomerID INT, SKU VARCHAR(20), Description VARCHAR(50), PRIMARY KE
Y(OrderID),
    -> FOREIGN KEY(CustomerID) REFERENCES Customers(CustomerID));
Query OK, 0 rows affected (0.09 sec)

mysql> DESCRIBE Orders;
+-------------+-------------+------+-----+---------+-------+
| Field       | Type        | Null | Key | Default | Extra |
+-------------+-------------+------+-----+---------+-------+
| OrderID     | int(11)     | NO   | PRI | 0       |       |
| CustomerID  | int(11)     | YES  | MUL | NULL    |       |
| SKU         | varchar(20) | YES  |     | NULL    |       |
| Description | varchar(50) | YES  |     | NULL    |       |
+-------------+-------------+------+-----+---------+-------+
4 rows in set (0.00 sec)
```

I created the Orders table. I assigned the primary key to OrderID and using the foreign key command, made a connection between the CustomerID primary key in the Customers table and CustomerID in this table.



```
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 (0.08 sec)

mysql> DESCRIBE RMA;
+---------+-------------+------+-----+---------+-------+
| Field   | Type        | Null | Key | Default | Extra |
+---------+-------------+------+-----+---------+-------+
| RMAID   | int(11)     | NO   | PRI | 0       |       |
| OrderID | int(11)     | YES  | MUL | NULL    |       |
| Step    | varchar(50) | YES  |     | NULL    |       |
| Status  | varchar(15) | YES  |     | NULL    |       |
| Reason  | varchar(15) | YES  |     | NULL    |       |
+---------+-------------+------+-----+---------+-------+
5 rows in set (0.01 sec)
```

The primary key in the RMA table is RMAID and OrderID is given a relationship to OrderID on the Orders table, making it a foreign key.

Updating Schema Based on New Requirements

Renaming "Customers" to "Collaborators"

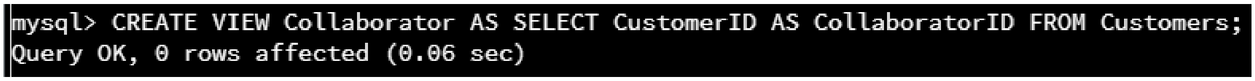
Update your existing tables from “Customer” to “Collaborator” using SQL based on this change in requirements. Provide the SQL commands you ran against MySQL to complete this successfully in your answer:

a. Rename all instances of “Customer” to “Collaborator.”

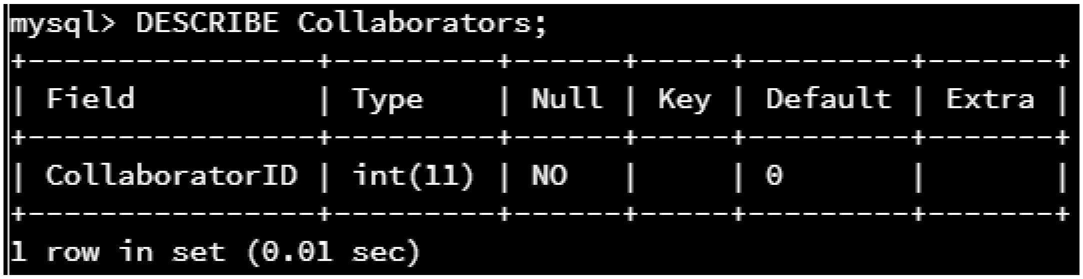
ALTER TABLE Customers RENAME TO Collaborators;
ALTER TABLE Orders CHANGE CustomerID CollaboratorID INT;

Creating a View for Unified Data Representation

CREATE VIEW CollaboratorsView AS
SELECT * FROM Collaborators;



```
mysql> CREATE VIEW Collaborator AS SELECT CustomerID AS CollaboratorID FROM Customers;
Query OK, 0 rows affected (0.06 sec)
```

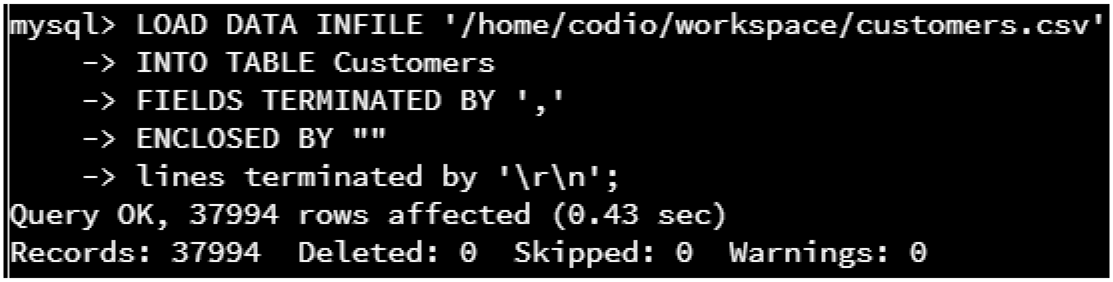


```
mysql> DESCRIBE Collaborators;
+----------------+---------+------+-----+---------+-------+
| Field          | Type    | Null | Key | Default | Extra |
+----------------+---------+------+-----+---------+-------+
| CollaboratorID | int(11) | NO   |     | 0       |       |
+----------------+---------+------+-----+---------+-------+
1 row in set (0.01 sec)
```

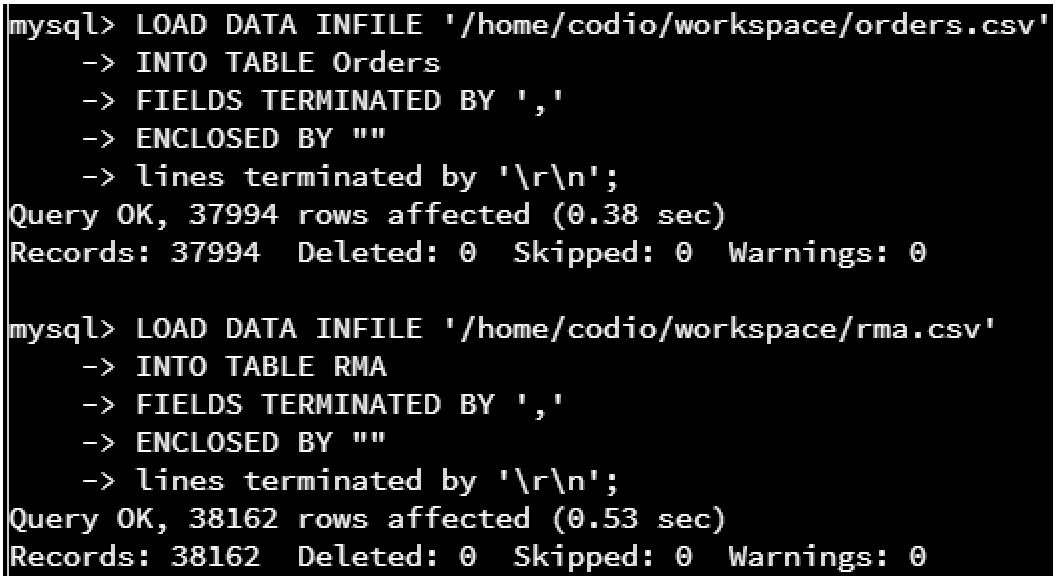
Used create view command to create a virtual table named “Collaborators.” This allows for all data to be in one table rather than across several tables.

Step 2: Data Import and Querying

1. **Import the data from each file into tables.**
    - Use the *QuantigrationUpdates* database, the three tables you created, and the three CSV files preloaded into Codio.
    - Use the import utility of your database program to load the data from each file into the table of the same name. You will perform this step three times, once for each table.

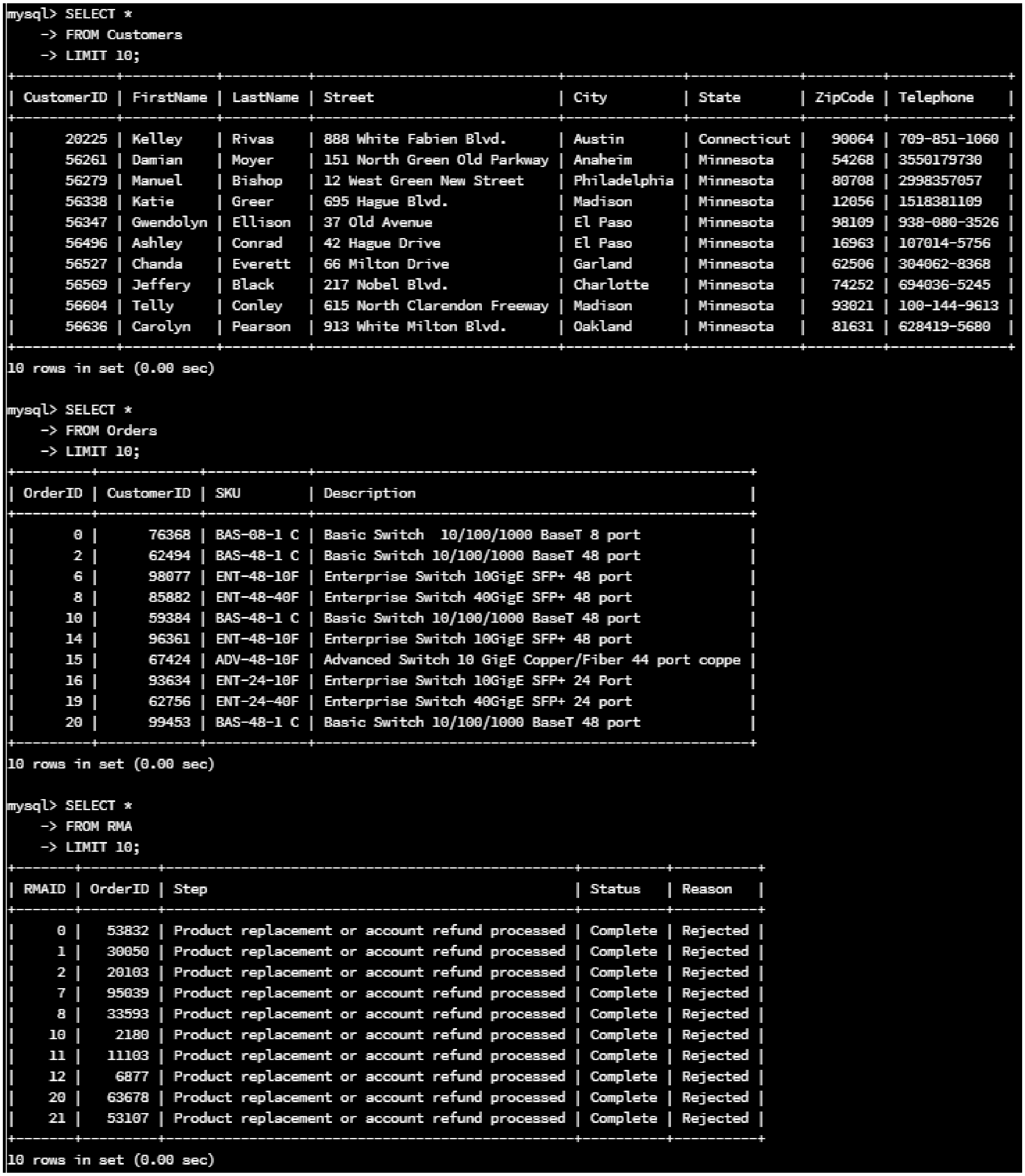


```
mysql> LOAD DATA INFILE '/home/codio/workspace/customers.csv'
    -> INTO TABLE Customers
    -> FIELDS TERMINATED BY ','
    -> ENCLOSED BY ""
    -> lines terminated by '\r\n';
Query OK, 37994 rows affected (0.43 sec)
Records: 37994  Deleted: 0  Skipped: 0  Warnings: 0
```



```
mysql> LOAD DATA INFILE '/home/codio/workspace/orders.csv'
    -> INTO TABLE Orders
    -> FIELDS TERMINATED BY ','
    -> ENCLOSED BY ""
    -> lines terminated by '\r\n';
Query OK, 37994 rows affected (0.38 sec)
Records: 37994  Deleted: 0  Skipped: 0  Warnings: 0

mysql> LOAD DATA INFILE '/home/codio/workspace/rma.csv'
    -> INTO TABLE RMA
    -> FIELDS TERMINATED BY ','
    -> ENCLOSED BY ""
    -> lines terminated by '\r\n';
Query OK, 38162 rows affected (0.53 sec)
Records: 38162  Deleted: 0  Skipped: 0  Warnings: 0
```

The above commands imported the customers, orders and rma csv files into the customers, orders and rma tables respectively.

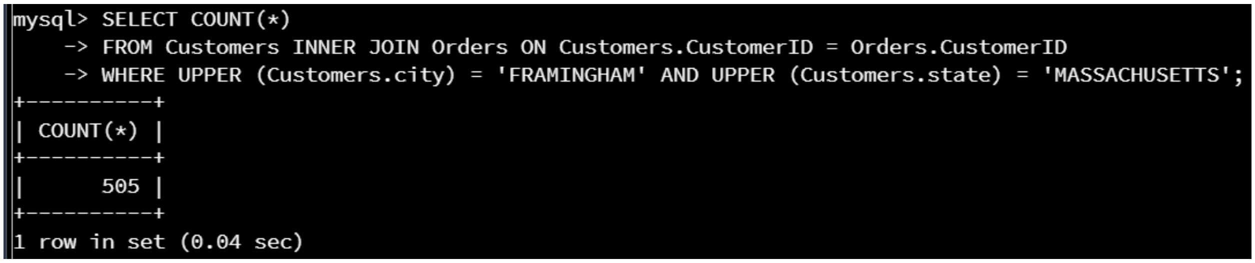


```
mysql> SELECT *
    -> FROM Customers
    -> LIMIT 10;
+------------+-----------+----------+-----------------------------+--------------+-------------+---------+--------------+
| CustomerID | FirstName | LastName | Street                      | City         | State       | ZipCode | Telephone    |
+------------+-----------+----------+-----------------------------+--------------+-------------+---------+--------------+
|      20225 | Kelley    | Rivas    | 888 White Fabien Blvd.      | Austin       | Connecticut |   90064 | 709-851-1060 |
|      56261 | Damian    | Moyer    | 151 North Green Old Parkway | Anaheim      | Minnesota   |   54268 | 3550179730   |
|      56279 | Manuel    | Bishop   | 12 West Green New Street    | Philadelphia | Minnesota   |   80708 | 2998357057   |
|      56338 | Katie     | Greer    | 695 Hague Blvd.             | Madison      | Minnesota   |   12056 | 1518381109   |
|      56347 | Gwendolyn | Ellison  | 37 Old Avenue               | El Paso      | Minnesota   |   98109 | 938-080-3526 |
|      56496 | Ashley    | Conrad   | 42 Hague Drive              | El Paso      | Minnesota   |   16963 | 107014-5756  |
|      56527 | Chanda    | Everett  | 66 Milton Drive             | Garland      | Minnesota   |   62506 | 304062-8368  |
|      56569 | Jeffery   | Black    | 217 Nobel Blvd.             | Charlotte    | Minnesota   |   74252 | 694036-5245  |
|      56604 | Telly     | Conley   | 615 North Clarendon Freeway | Madison      | Minnesota   |   93021 | 100-144-9613 |
|      56636 | Carolyn   | Pearson  | 913 White Milton Blvd.      | Oakland      | Minnesota   |   81631 | 628419-5680  |
+------------+-----------+----------+-----------------------------+--------------+-------------+---------+--------------+
10 rows in set (0.00 sec)

mysql> SELECT *
    -> FROM Orders
    -> LIMIT 10;
+---------+------------+------------+----------------------------------------------------+
| OrderID | CustomerID | SKU        | Description                                        |
+---------+------------+------------+----------------------------------------------------+
|       0 |      76368 | BAS-08-1 C | Basic Switch  10/100/1000 BaseT 8 port             |
|       2 |      62494 | BAS-48-1 C | Basic Switch 10/100/1000 BaseT 48 port             |
|       6 |      98077 | ENT-48-10F | Enterprise Switch 10GigE SFP+ 48 port              |
|       8 |      85882 | ENT-48-40F | Enterprise Switch 40GigE SFP+ 48 port              |
|      10 |      59384 | BAS-48-1 C | Basic Switch 10/100/1000 BaseT 48 port             |
|      14 |      96361 | ENT-48-10F | Enterprise Switch 10GigE SFP+ 48 port              |
|      15 |      67424 | ADV-48-10F | Advanced Switch 10 GigE Copper/Fiber 44 port coppe |
|      16 |      93634 | ENT-24-10F | Enterprise Switch 10GigE SFP+ 24 Port              |
|      19 |      62756 | ENT-24-40F | Enterprise Switch 40GigE SFP+ 24 port              |
|      20 |      99453 | BAS-48-1 C | Basic Switch 10/100/1000 BaseT 48 port             |
+---------+------------+------------+----------------------------------------------------+
10 rows in set (0.00 sec)

mysql> SELECT *
    -> FROM RMA
    -> LIMIT 10;
+-------+---------+-------------------------------------------------+----------+----------+
| RMAID | OrderID | Step                                            | Status   | Reason   |
+-------+---------+-------------------------------------------------+----------+----------+
|     0 |   53832 | Product replacement or account refund processed | Complete | Rejected |
|     1 |   30050 | Product replacement or account refund processed | Complete | Rejected |
|     2 |   20103 | Product replacement or account refund processed | Complete | Rejected |
|     7 |   95039 | Product replacement or account refund processed | Complete | Rejected |
|     8 |   33593 | Product replacement or account refund processed | Complete | Rejected |
|    10 |    2180 | Product replacement or account refund processed | Complete | Rejected |
|    11 |   11103 | Product replacement or account refund processed | Complete | Rejected |
|    12 |    6877 | Product replacement or account refund processed | Complete | Rejected |
|    20 |   63678 | Product replacement or account refund processed | Complete | Rejected |
|    21 |   53107 | Product replacement or account refund processed | Complete | Rejected |
+-------+---------+-------------------------------------------------+----------+----------+
10 rows in set (0.00 sec)
```

I validated the output using select * from each table, and with a limit of 10 records per table.

2. **Write basic queries against imported tables to organize and analyze targeted data.** For each query, replace the bracketed text with a screenshot of the query and its output. You should also include a brief, 1- to 3-sentence description of the output.

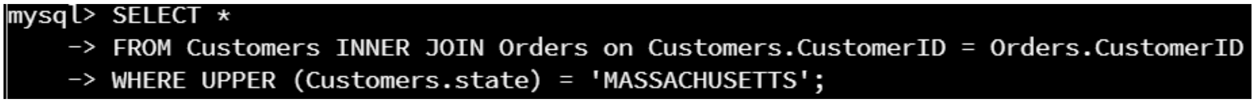
- Write an SQL query that returns the count of orders for customers located only in the city of Framingham, Massachusetts.
    i. How many records were returned?



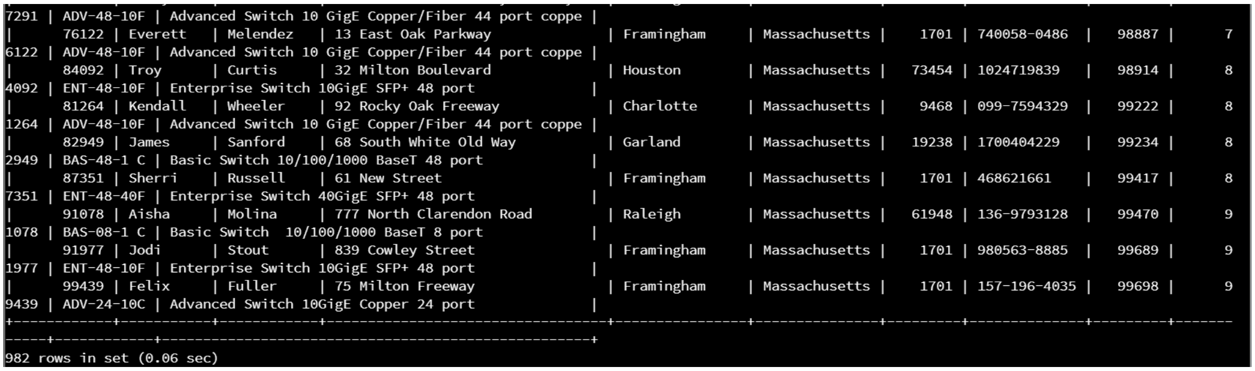
```
mysql> SELECT COUNT(*)
    -> FROM Customers INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID
    -> WHERE UPPER (Customers.city) = 'FRAMINGHAM' AND UPPER (Customers.state) = 'MASSACHUSETTS';
+----------+
| COUNT(*) |
+----------+
|      505 |
+----------+
1 row in set (0.04 sec)
```

There are 505 records of customers located in Framingham, MA. The above query only returns data related to Framingham, Massachusetts between the Orders and Customers tables using an inner join command.

- Write an SQL query to select all of the customers located in the state of Massachusetts.
    i. Use a WHERE clause to limit the number of records in the customers table to only those who are located in Massachusetts.
    ii. Record an answer to the following question: How many records were returned?



```
mysql> SELECT *
    -> FROM Customers INNER JOIN Orders on Customers.CustomerID = Orders.CustomerID
    -> WHERE UPPER (Customers.state) = 'MASSACHUSETTS';
```



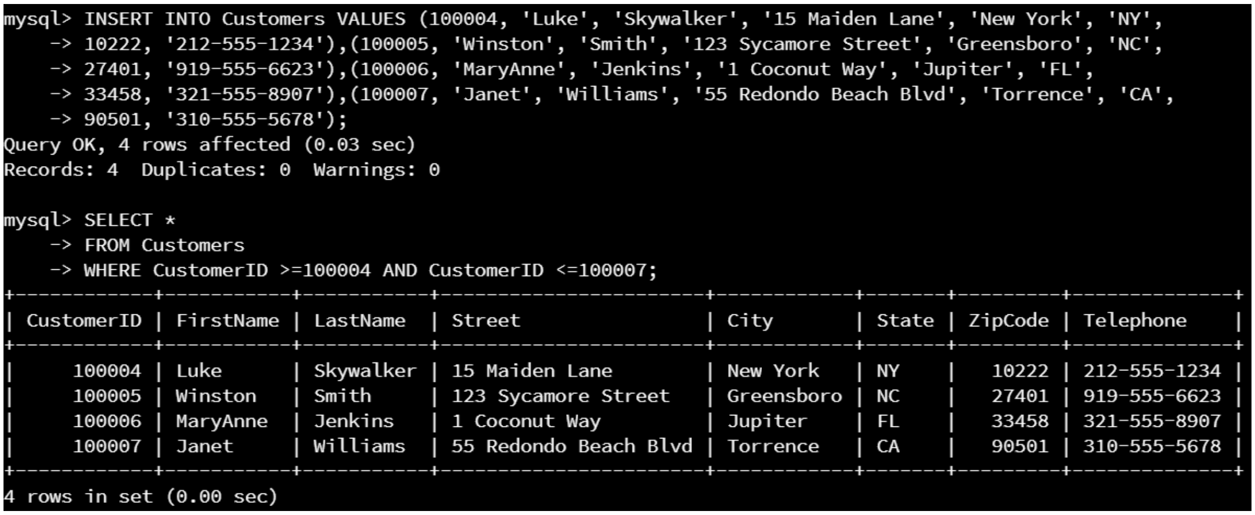
```
7291 | ADV-48-10F | Advanced Switch 10 GigE Copper/Fiber 44 port coppe |
|      76122 | Everett   | Melendez   | 13 East Oak Parkway              | Framingham     | Massachusetts |    1701 | 740058-0486  |   98887 |       7
6122 | ADV-48-10F | Advanced Switch 10 GigE Copper/Fiber 44 port coppe |
|      84092 | Troy      | Curtis     | 32 Milton Boulevard              | Houston        | Massachusetts |   73454 | 1024719839   |   98914 |       8
4092 | ENT-48-10F | Enterprise Switch 10GigE SFP+ 48 port              |
|      81264 | Kendall   | Wheeler    | 92 Rocky Oak Freeway             | Charlotte      | Massachusetts |    9468 | 099-7594329  |   99222 |       8
1264 | ADV-48-10F | Advanced Switch 10 GigE Copper/Fiber 44 port coppe |
|      82949 | James     | Sanford    | 68 South White Old Way           | Garland        | Massachusetts |   19238 | 1700404229   |   99234 |       8
2949 | BAS-48-1 C | Basic Switch 10/100/1000 BaseT 48 port             |
|      87351 | Sherri    | Russell    | 61 New Street                    | Framingham     | Massachusetts |    1701 | 468621661    |   99417 |       8
7351 | ENT-48-40F | Enterprise Switch 40GigE SFP+ 48 port              |
|      91078 | Aisha     | Molina     | 777 North Clarendon Road         | Raleigh        | Massachusetts |   61948 | 136-9793128  |   99470 |       9
1078 | BAS-08-1 C | Basic Switch  10/100/1000 BaseT 8 port             |
|      91977 | Jodi      | Stout      | 839 Cowley Street                | Framingham     | Massachusetts |    1701 | 980563-8885  |   99689 |       9
1977 | ENT-48-10F | Enterprise Switch 10GigE SFP+ 48 port              |
|      99439 | Felix     | Fuller     | 75 Milton Freeway                | Framingham     | Massachusetts |    1701 | 157-196-4035 |   99698 |       9
9439 | ADV-24-10C | Advanced Switch 10GigE Copper 24 port              |
+------------+-----------+------------+----------------------------------+----------------+---------------+---------+--------------+---------+-------
-----+------------+----------------------------------------------------+
982 rows in set (0.06 sec)
```

There are 982 records returned. These are the last 9 records.

- Write a 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	15 Maiden Lane	New York	NY	10222	212-555-1234
100005	Winston	Smith	123 Sycamore Street	Greensboro	NC	27401	919-555-6623
100006	MaryAnne	Jenkins	1 Coconut Way	Jupiter	FL	33458	321-555-8907
100007	Janet	Williams	55 Redondo Beach Blvd	Torrence	CA	90501	310-555-5678

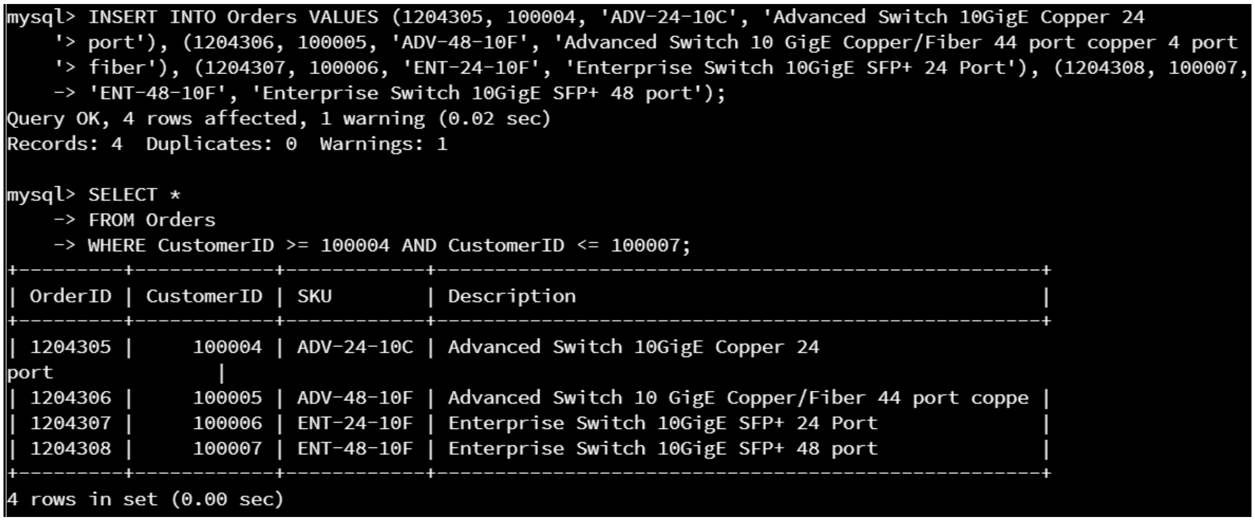


```
mysql> INSERT INTO Customers VALUES (100004, 'Luke', 'Skywalker', '15 Maiden Lane', 'New York', 'NY',
    -> 10222, '212-555-1234'),(100005, 'Winston', 'Smith', '123 Sycamore Street', 'Greensboro', 'NC',
    -> 27401, '919-555-6623'),(100006, 'MaryAnne', 'Jenkins', '1 Coconut Way', 'Jupiter', 'FL',
    -> 33458, '321-555-8907'),(100007, 'Janet', 'Williams', '55 Redondo Beach Blvd', 'Torrence', 'CA',
    -> 90501, '310-555-5678');
Query OK, 4 rows affected (0.03 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> SELECT *
    -> FROM Customers
    -> WHERE CustomerID >=100004 AND CustomerID <=100007;
+------------+-----------+-----------+-----------------------+------------+-------+---------+--------------+
| CustomerID | FirstName | LastName  | Street                | City       | State | ZipCode | Telephone    |
+------------+-----------+-----------+-----------------------+------------+-------+---------+--------------+
|     100004 | Luke      | Skywalker | 15 Maiden Lane        | New York   | NY    |   10222 | 212-555-1234 |
|     100005 | Winston   | Smith     | 123 Sycamore Street   | Greensboro | NC    |   27401 | 919-555-6623 |
|     100006 | MaryAnne  | Jenkins   | 1 Coconut Way         | Jupiter    | FL    |   33458 | 321-555-8907 |
|     100007 | Janet     | Williams  | 55 Redondo Beach Blvd | Torrence   | CA    |   90501 | 310-555-5678 |
+------------+-----------+-----------+-----------------------+------------+-------+---------+--------------+
4 rows in set (0.00 sec)
```

I added four new customer records into Customers. The query only retrieves the newly added using the less than greater than operators against their customer id numbers.

ii. Orders Table

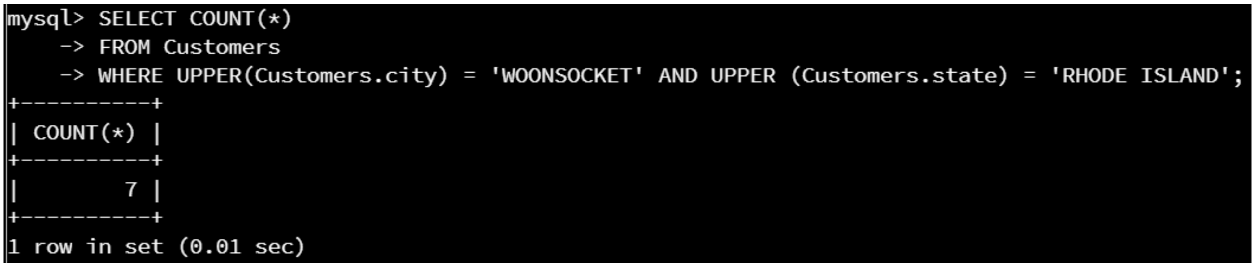
OrderID	CustomerID	SKU	Description
1204305	100004	ADV-24-10C	Advanced Switch 10GigE Copper 24 port
1204306	100005	ADV-48-10F	Advanced Switch 10 GigE Copper/Fiber 44 port copper 4 port fiber
1204307	100006	ENT-24-10F	Enterprise Switch 10GigE SFP+ 24 Port
1204308	100007	ENT-48-10F	Enterprise Switch 10GigE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



```
mysql> INSERT INTO Orders VALUES (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');
Query OK, 4 rows affected, 1 warning (0.02 sec)
Records: 4  Duplicates: 0  Warnings: 1

mysql> SELECT *
    -> FROM Orders
    -> WHERE CustomerID >= 100004 AND CustomerID <= 100007;
+---------+------------+------------+-----------------------------------------------------+
| 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 coppe |
| 1204307 |     100006 | ENT-24-10F | Enterprise Switch 10GigE SFP+ 24 Port               |
| 1204308 |     100007 | ENT-48-10F | Enterprise Switch 10GigE SFP+ 48 port               |
+---------+------------+------------+-----------------------------------------------------+
4 rows in set (0.00 sec)
```

As above, I added four new orders into Orders. This query only retrieves the newly added orders using the less than greater than operators against customer id numbers.

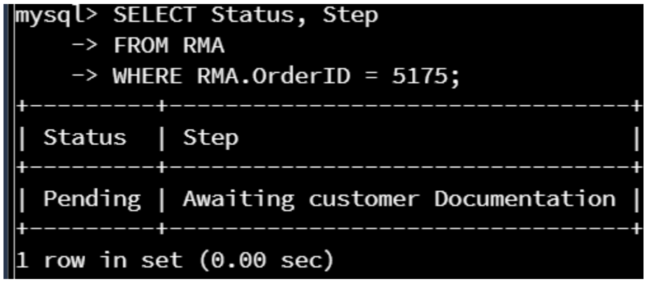
- In the customers table, perform a query to count all records where the city is Woonsocket, Rhode Island.
    i. How many records are in the customers table where the field “city” equals “Woonsocket”?



```
mysql> SELECT COUNT(*)
    -> FROM Customers
    -> WHERE UPPER(Customers.city) = 'WOONSOCKET' AND UPPER (Customers.state) = 'RHODE ISLAND';
+----------+
| COUNT(*) |
+----------+
|        7 |
+----------+
1 row in set (0.01 sec)
```

The above query returned that there are 7 records where the city is Woonsocket, Rhode Island.

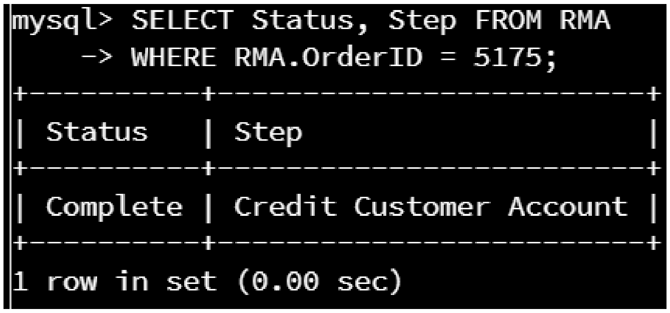
- In the rma database, update a customer’s records.
    i. Write an SQL statement to select the current fields of *status* and *step* for the record in the *rma* table with an *orderid* value of “5175.”
        1. What are the current status and step?



```
mysql> SELECT Status, Step
    -> FROM RMA
    -> WHERE RMA.OrderID = 5175;
+---------+-------------------------------+
| Status  | Step                          |
+---------+-------------------------------+
| Pending | Awaiting customer Documentation |
+---------+-------------------------------+
1 row in set (0.00 sec)
```

The current status is pending and it’s in the ‘Awaiting customer documentation’ step of the process.

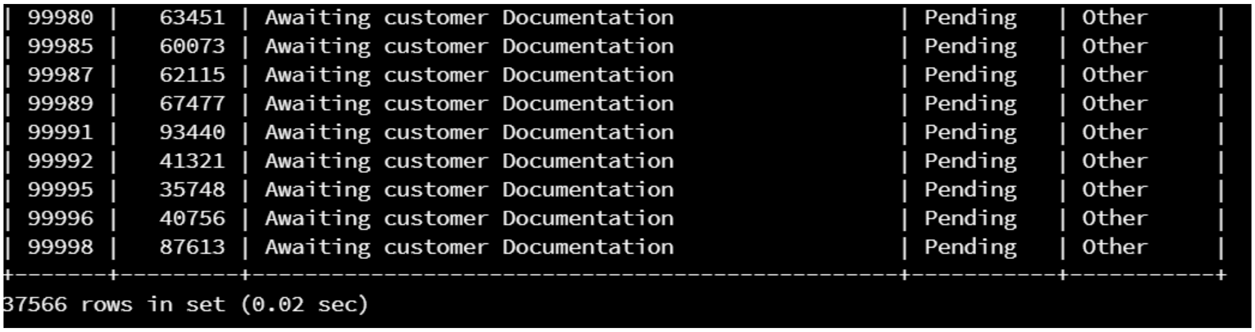
    ii. Write an SQL statement to update the *status* and *step* for the *orderid*, 5175 to *status* = “Complete” and *step* = “Credit Customer Account.”
        1. What are the updated *status* and *step* values for this record?



```
mysql> SELECT Status, Step FROM RMA
    -> WHERE RMA.OrderID = 5175;
+----------+-------------------------+
| Status   | Step                    |
+----------+-------------------------+
| Complete | Credit Customer Account |
+----------+-------------------------+
1 row in set (0.00 sec)
```

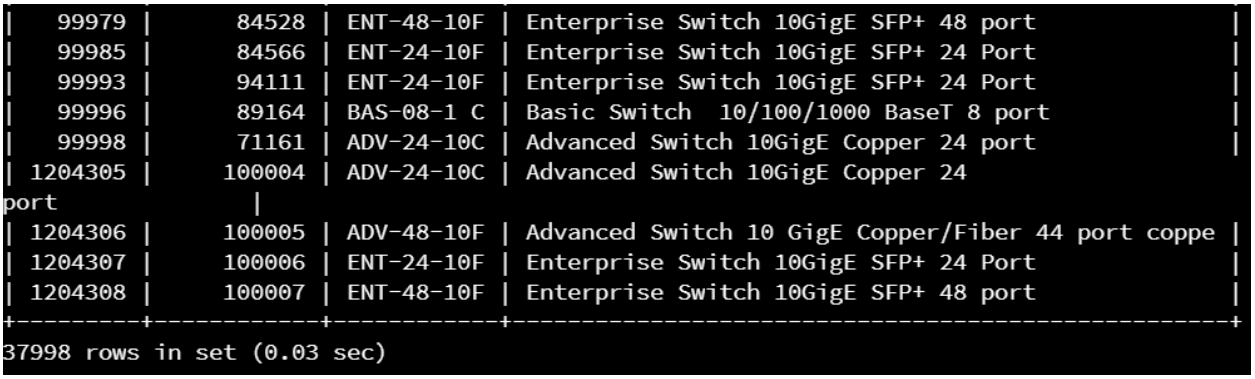
Status is now complete, and the step is “Credit Customer Account.”

- Delete rma records.
    i. Write an SQL statement to delete all records with a reason of “Rejected.”
    DELETE FROM RMA WHERE Reason = 'Rejected';
        1. How many records were deleted? 596.



```
| 99980 |   63451 | Awaiting customer Documentation             | Pending   | Other     |
| 99985 |   60073 | Awaiting customer Documentation             | Pending   | Other     |
| 99987 |   62115 | Awaiting customer Documentation             | Pending   | Other     |
| 99989 |   67477 | Awaiting customer Documentation             | Pending   | Other     |
| 99991 |   93440 | Awaiting customer Documentation             | Pending   | Other     |
| 99992 |   41321 | Awaiting customer Documentation             | Pending   | Other     |
| 99995 |   35748 | Awaiting customer Documentation             | Pending   | Other     |
| 99996 |   40756 | Awaiting customer Documentation             | Pending   | Other     |
| 99998 |   87613 | Awaiting customer Documentation             | Pending   | Other     |
+-------+---------+---------------------------------------------+-----------+-----------+
37566 rows in set (0.02 sec)
```

Create an output file of the required query results. Write an SQL statement to list the contents of the *orders* table and send the output to a file that has a .csv extension.



```
|   99979 |      84528 | ENT-48-10F | Enterprise Switch 10GigE SFP+ 48 port              |
|   99985 |      84566 | ENT-24-10F | Enterprise Switch 10GigE SFP+ 24 Port              |
|   99993 |      94111 | ENT-24-10F | Enterprise Switch 10GigE SFP+ 24 Port              |
|   99996 |      89164 | BAS-08-1 C | Basic Switch  10/100/1000 BaseT 8 port             |
|   99998 |      71161 | ADV-24-10C | Advanced Switch 10GigE Copper 24 port              |
| 1204305 |     100004 | ADV-24-10C | Advanced Switch 10GigE Copper 24
port             |
| 1204306 |     100005 | ADV-48-10F | Advanced Switch 10 GigE Copper/Fiber 44 port coppe |
| 1204307 |     100006 | ENT-24-10F | Enterprise Switch 10GigE SFP+ 24 Port              |
| 1204308 |     100007 | ENT-48-10F | Enterprise Switch 10GigE SFP+ 48 port              |
+---------+------------+------------+----------------------------------------------------+
37998 rows in set (0.03 sec)
```

The four new records are returned to the table.

Query used:

```
SELECT * FROM Orders
INTO OUTFILE '/home/codio/workspace/OrdersUpdates.csv'
FIELDS TERMINATED BY ','
ENCLOSED BY '"'
LINES TERMINATED BY '\n';
```

This SQL project demonstrates proficiency in database design, data manipulation, and querying skills relevant to a data analyst role. The project includes:

- Database and table creation
- Schema modification to meet evolving requirements
- Efficient data import and validation
- SQL queries for analytical insights
- Data updates and deletions
- Exporting results for reporting