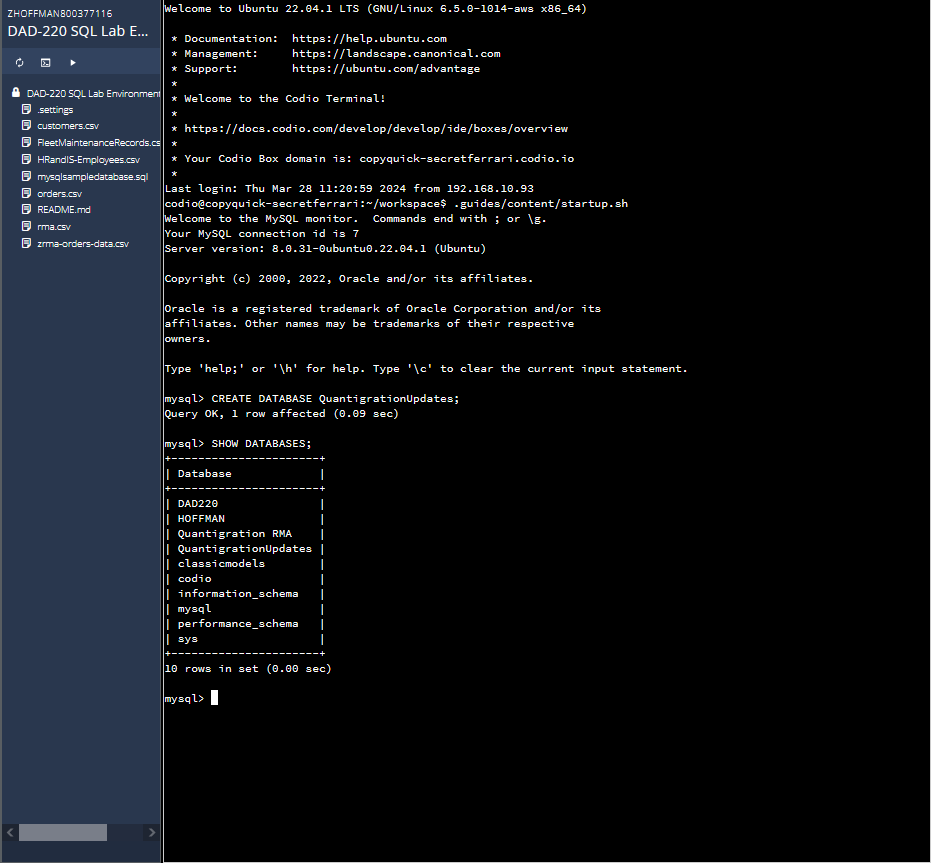
# DAD 220 Project One Template

Complete these steps as you work through the directions for Project One. Replace the bracketed text with your screenshots and brief explanations of the work they capture. Each screenshot and its explanation should be sized to approximately one-quarter of the page with the description written below the screenshot. Follow these rules for each of the prompts and questions below. Review the example document in the Supporting Materials section of the Project One Guidelines and Rubric for assistance.

## Step One: Create a Database

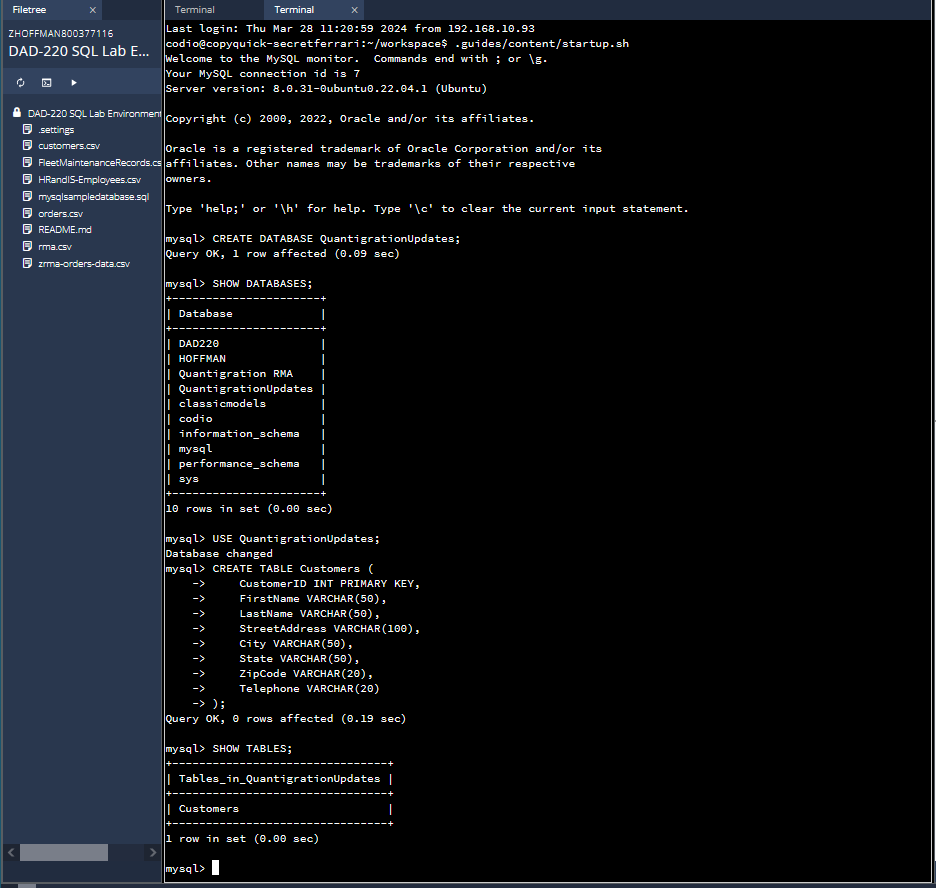
1. In your online IDE (Codio), **create a database schema** called QuantigrationUpdates that will hold tables by using SQL commands.
   1. List out the database name on the screen.
   2. Provide the SQL commands you ran against MySQL to complete this step.



Here I was asked to create a new database and call it “QuantigrationUpdates”. This is the command I used to complete this task.

CREATE DATABASE QuantigrationUpdates;

1. Connect to the QuantigrationUpdates schema. Using the ERD as a reference, **write SQL commands to create** the following **tables** with the appropriate attributes and keys to demonstrate relationships based on the ERD.
   1. A table named Customers to store customer information with a primary key of Customer ID. Provide the SQL commands you ran against MySQL to complete this step.



I was then asked to create a table named “Customers” with a primary key of “CustomerID”. Here are the commands I used

CREATE TABLE Customers (

CustomerID INT PRIMARY KEY,

FirstName VARCHAR(50),

LastName VARCHAR(50),

StreetAddress VARCHAR(100),

City VARCHAR(50),

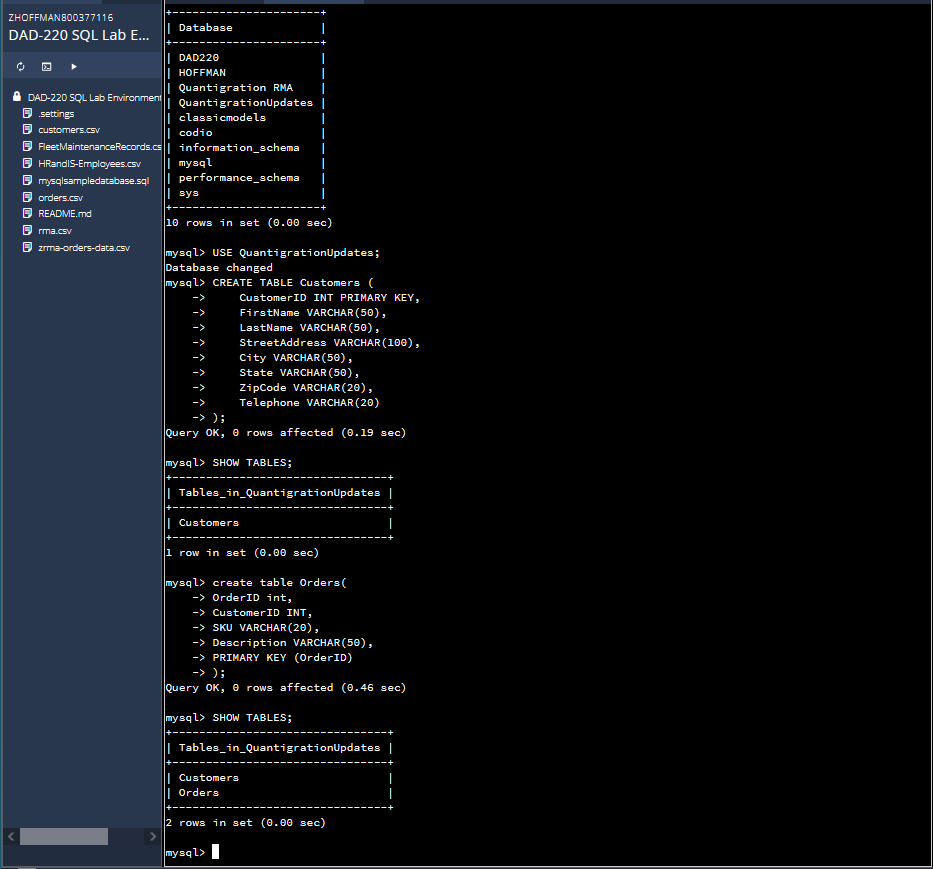
State VARCHAR(50),

ZipCode VARCHAR(20),

Telephone VARCHAR(20));

SHOW TABLES;

* 1. A table named Ordersto store order information with a primary key of Order ID and a foreign key of Customer ID. Provide the SQL commands you ran against MySQL to complete this step.



Here we were asked to create another table, but this one needs to be called “Orders”. This is the command I used

create table Orders(

OrderID int,

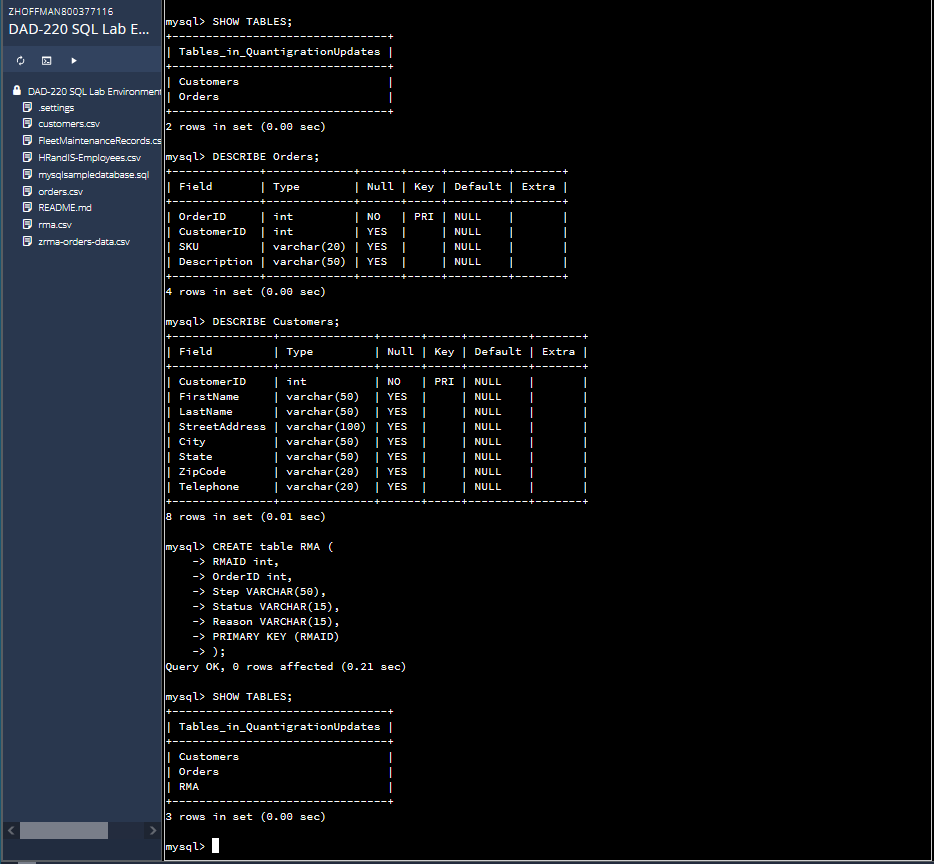
CustomerID INT,

SKU VARCHAR(20),

Description VARCHAR(50),

PRIMARY KEY (OrderID));

* 1. A table named RMA to store RMA information with a primary key of RMA ID and a foreign key of Order ID. Provide the SQL commands you ran against MySQL to complete this step.



Next we are to create a table named “RMA” to store RMA information with a primary key of RMA ID and a foreign key of Order ID. This is the command I used

CREATE table RMA (

RMAID int,

OrderID int,

Step VARCHAR(50),

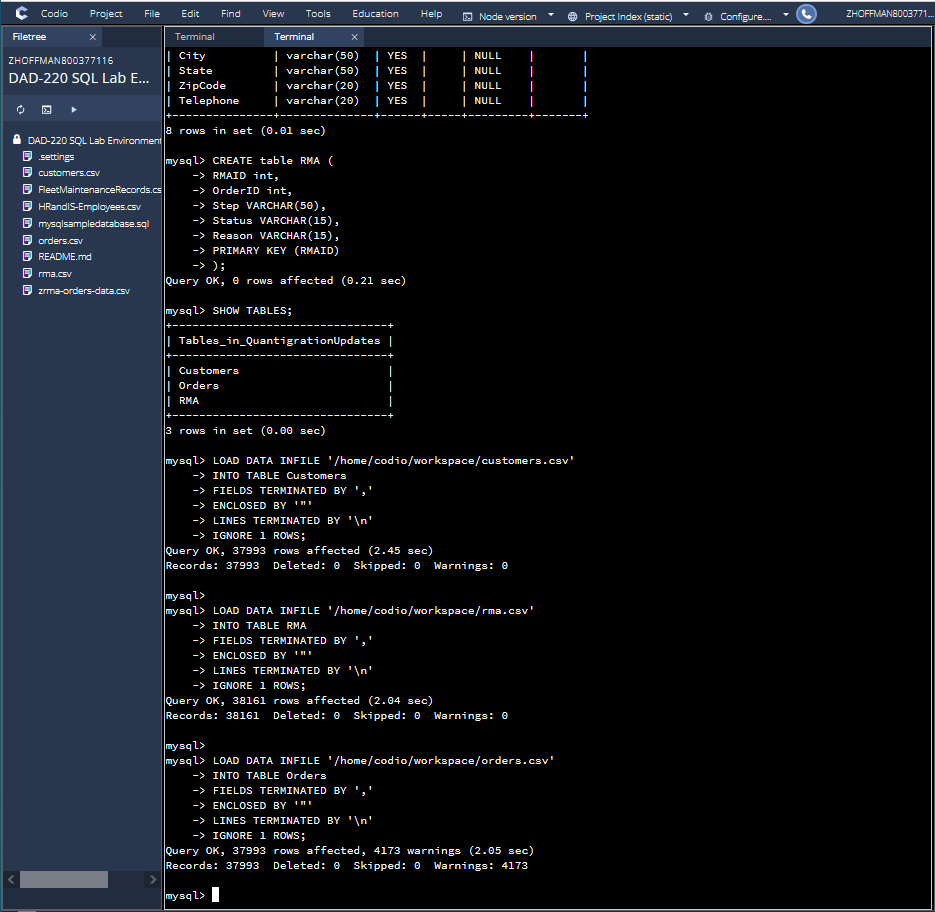
Status VARCHAR(15),

Reason VARCHAR(15),

PRIMARY KEY (RMAID));

## Step Two: Load and Query the Data

1. **Import** the **data** from each file **into tables.** 
   1. Use the QuantigrationUpdates database, the three tables you created, and the three CSV files preloaded into Codio.
   2. Use the import utility of your database program to load the data from each file into the table of the same name. Perform this step three times, once for each table.
   3. Provide the SQL commands you ran against MySQL to complete this step.



Now we have been asked to load the data from each pre-loaded file into the table of the same name and perform this step three times, once for each table. Here are the commands I used.

LOAD DATA INFILE '/home/codio/workspace/customers.csv'

INTO TABLE Customers

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 ROWS;

LOAD DATA INFILE '/home/codio/workspace/rma.csv'

INTO TABLE RMA

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 ROWS;

LOAD DATA INFILE '/home/codio/workspace/orders.csv'

INTO TABLE Orders

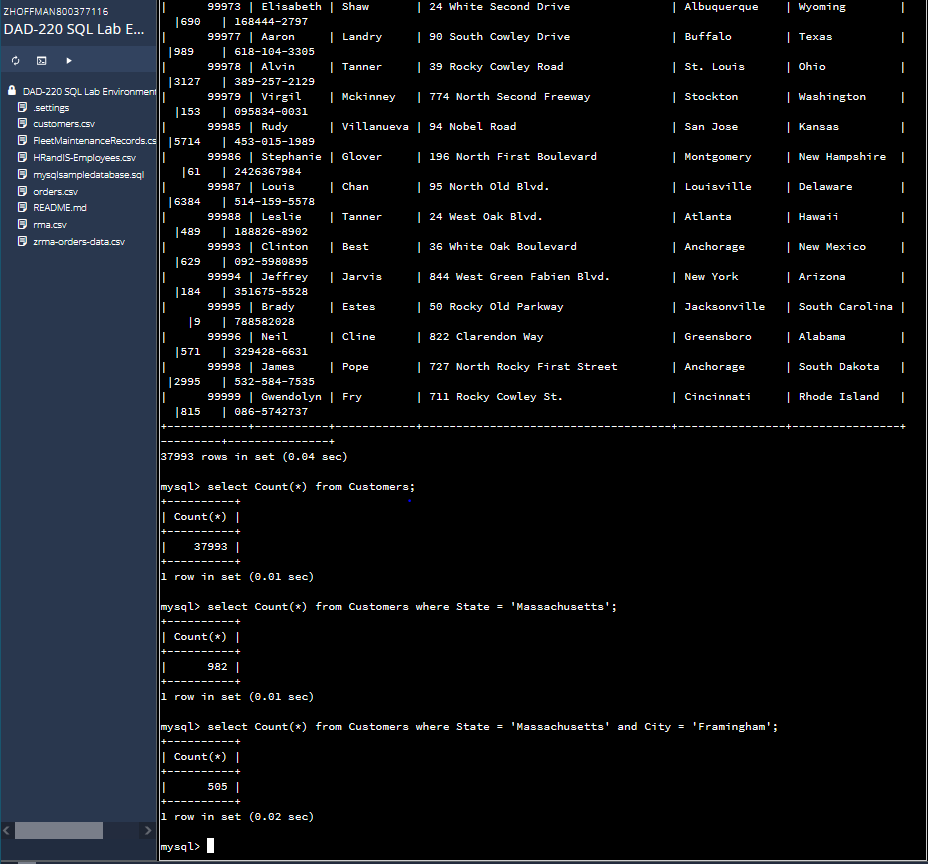
FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n'

IGNORE 1 ROWS;

1. **Write basic queries** against the imported tables to organize and analyze the targeted data**.** For each query, replace the bracketed text with a screenshot of the query and its output. Also, include a one- to three-sentence description of the output.
   1. Write a SQL query that returns the count of orders for customers located only in Framingham, Massachusetts.
      1. This query will use a table join between the Customers and Orders tables. The query will also use a WHERE clause.
      2. How many records were returned?



The task for this part is to figure out how many records there are when you join the Customers and Orders table together and filter for just Framingham, Massachusetts. Here are the commands I used

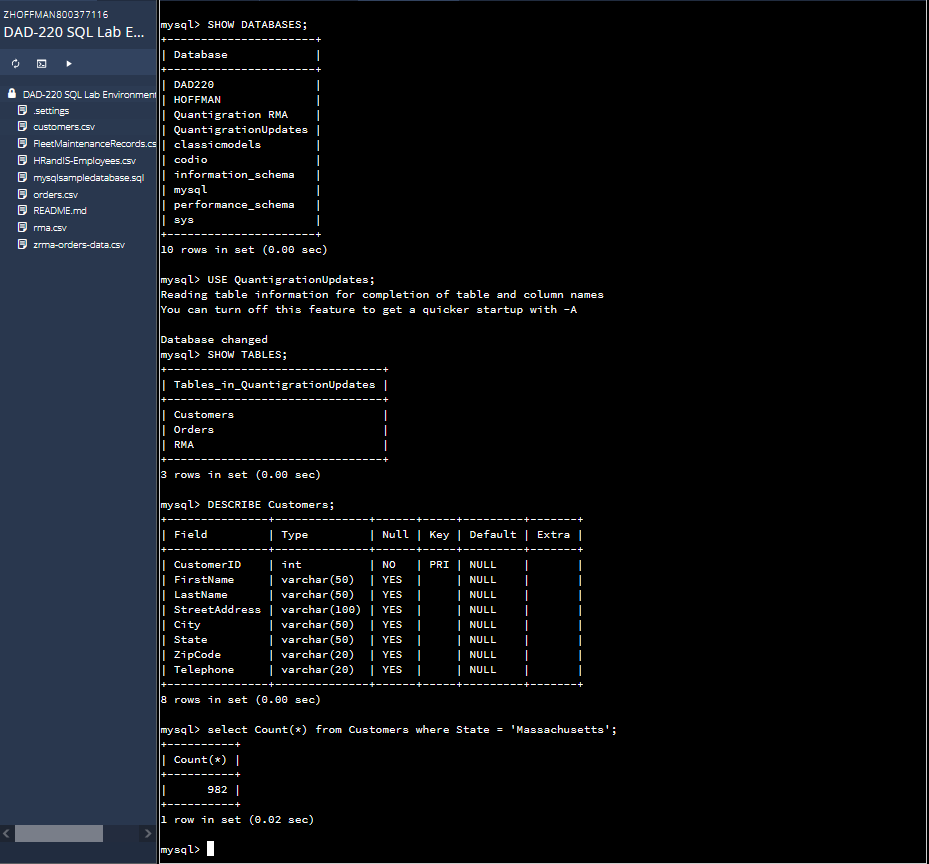
select Count(\*) from Customers;

select Count(\*) from Customers where State = 'Massachusetts';

select Count(\*) from Customers where State = 'Massachusetts' and City = 'Framingham';

Output - 505 records were returned

* 1. Write a SQL query to select all of the customers located in Massachusetts.
     1. Use a WHERE clause to limit the number of records in the Customers table to only those who are located in Massachusetts.
     2. How many records were returned?



Here we were just asked to use a WHERE clause to limit the number of records in the Customers table to only those who are located in Massachusetts. This is the command I used and the output I received

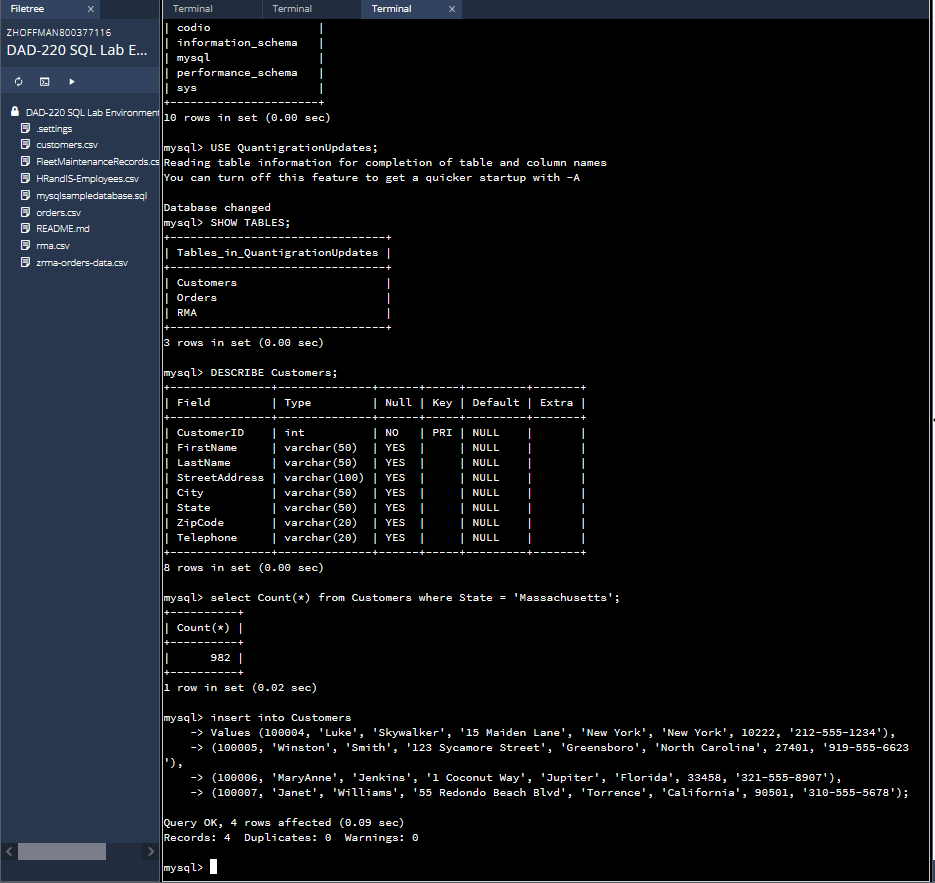
select Count(\*) from Customers where State = 'Massachusetts';

Output - 982 records returned

* 1. Write a SQL query to insert four new records into the Orders and Customers tables using the data below:

**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 |



Now we have to enter 4 new records into our Customers table. Here is the command I used.

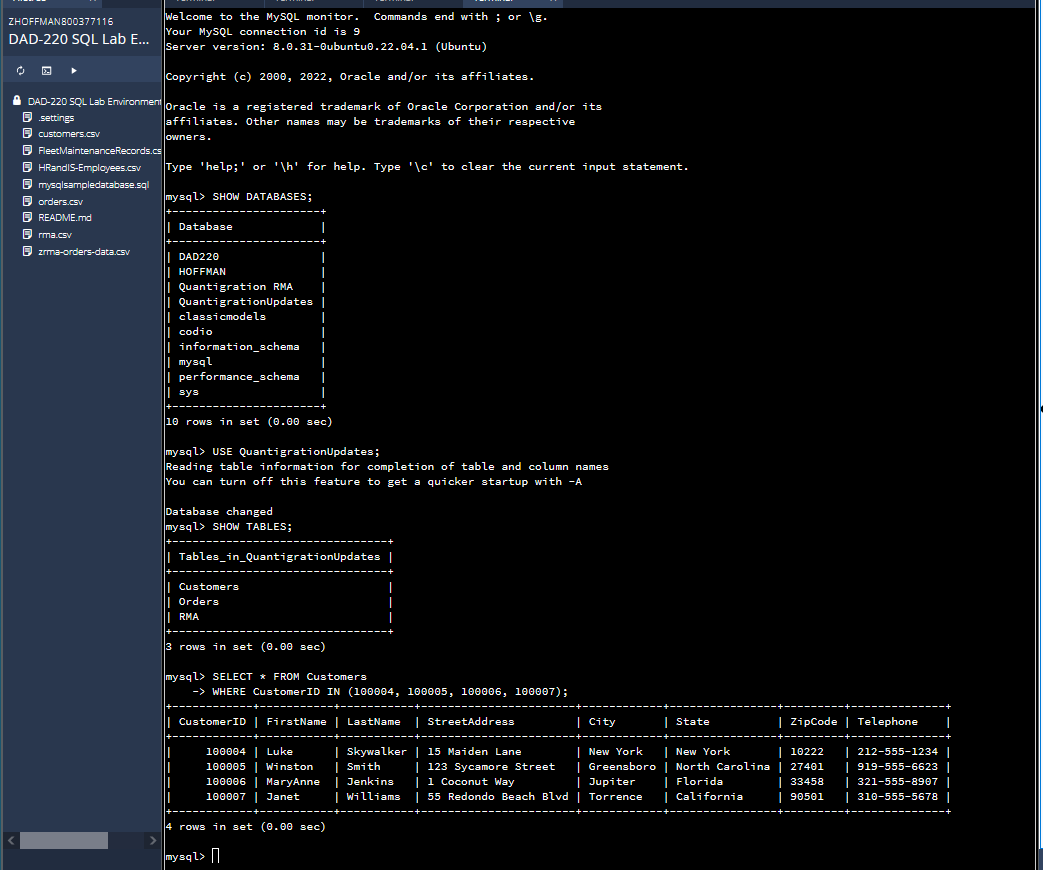
insert into Customers

Values (100004, 'Luke', 'Skywalker', '15 Maiden Lane', 'New York', 'New York', 10222, '212-555-1234'),

(100005, 'Winston', 'Smith', '123 Sycamore Street', 'Greensboro', 'North Carolina', 27401, '919-555-6623'),

(100006, 'MaryAnne', 'Jenkins', '1 Coconut Way', 'Jupiter', 'Florida', 33458, '321-555-8907'),

(100007, 'Janet', 'Williams', '55 Redondo Beach Blvd', 'Torrence', 'California', 90501, '310-555-5678');



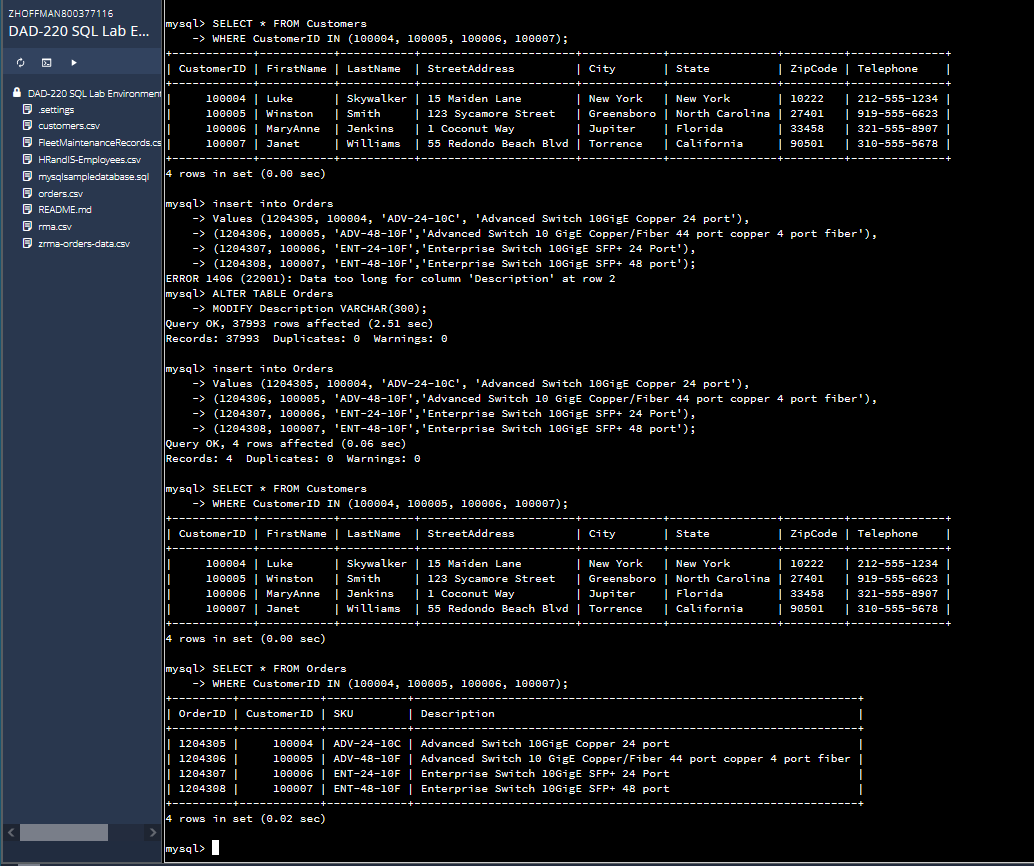
I also provided a screenshot of a specific part of the customers table that shows only the newly added records. I used CustomerID to pull the records I wanted to display. The command I used to complete this task was as follows

SELECT \* FROM Customers

WHERE CustomerID IN (100004, 100005, 100006, 100007);

**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 |



In this section we were also asked to add 4 new records to the Orders table. I kept getting this error that line 2 in the command which is the Description column of the table was too long. So I had to run a command to expand the limit of what can be entered in the table. Then I used a command to view the 4 new records in the Orders table that we just created.

MODIFY Description VARCHAR(300);

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'),

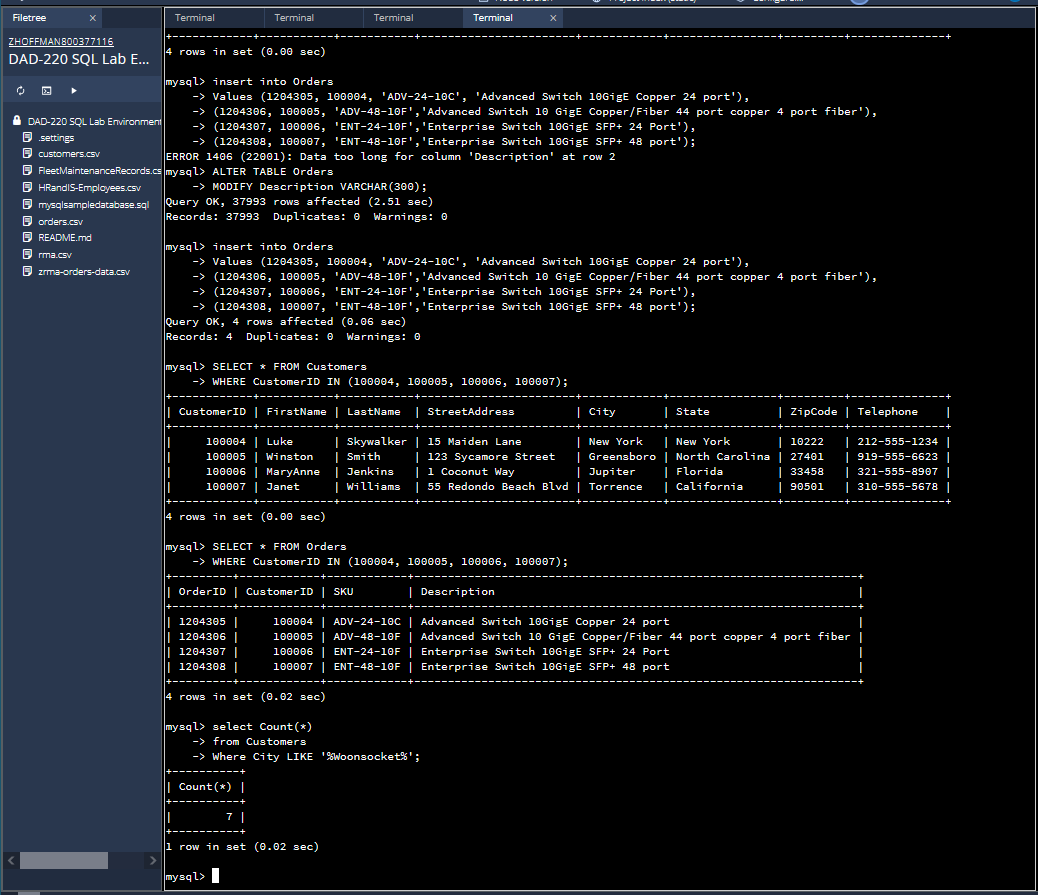
(1204307, 100006, 'ENT-24-10F', 'Enterprise Switch 10GigE SFP+ 24 Port'),

(1204308, 100007, 'ENT-48-10F', 'Enterprise Switch 10GigE SFP+ 48 port');

SELECT \* FROM Orders

WHERE CustomerID IN (100004, 100005, 100006, 100007);

* 1. In the Customers table, perform a query to count all records where the city is Woonsocket and the state is Rhode Island.
     1. How many records are in the Customers table where the field "city" equals "Woonsocket"?



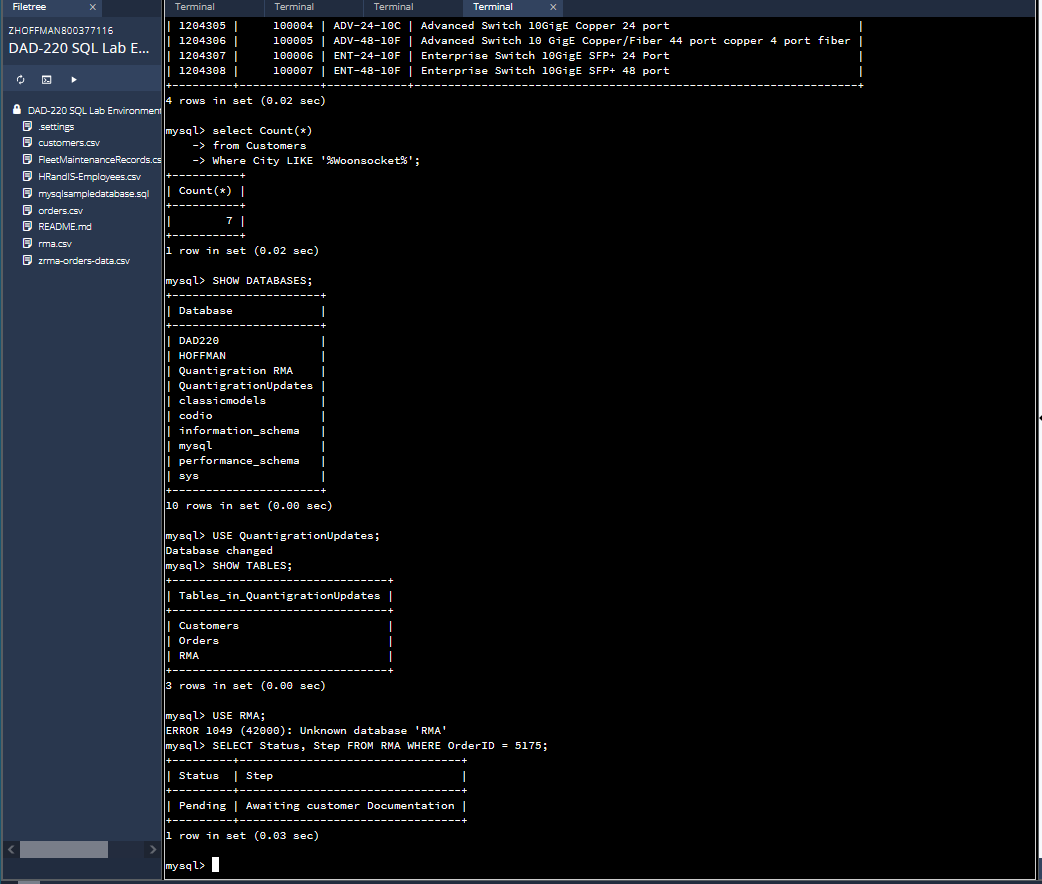
Next we want to figure out how many records are in the Customers table where the field city equals Woonsocket. This is the command I used to find the answer

select Count(\*)

from Customers

Where City LIKE '%Woonsocket%';

* 1. In the RMA database, update a customer's records.
     1. Write a 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?



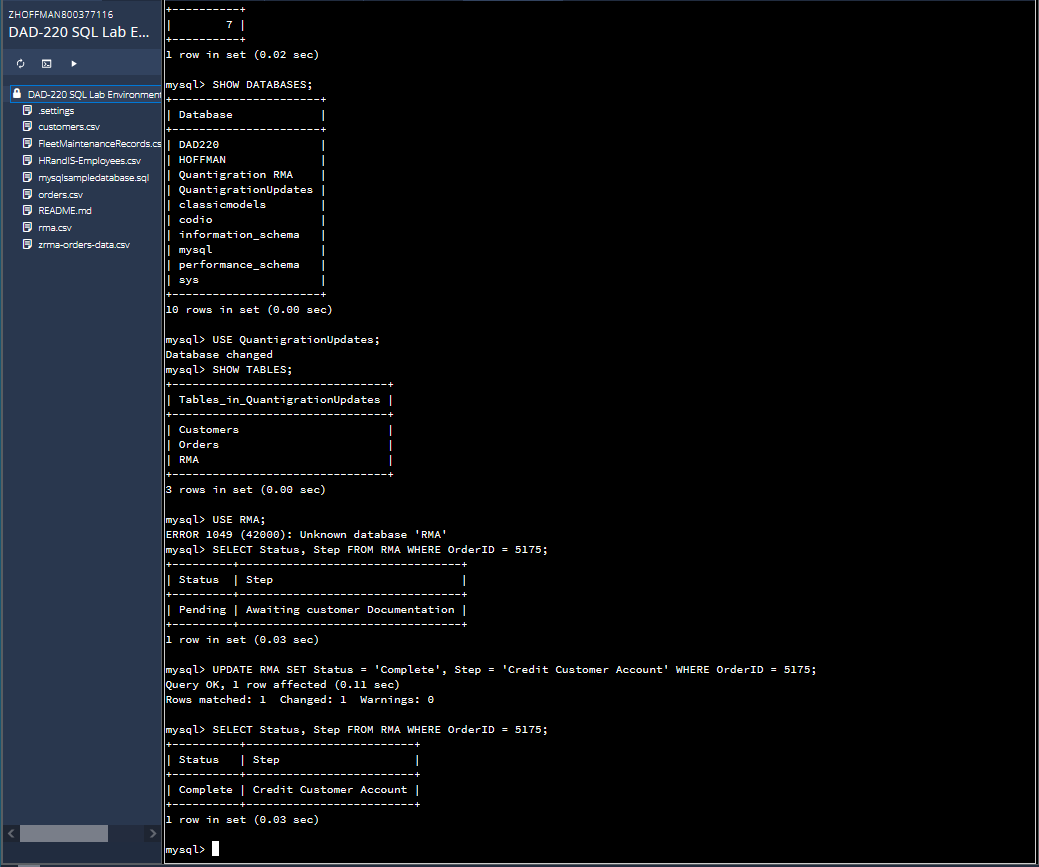
Here we were asked to find out the status and step for the record with an OderID value of 5175 located in the RMA table. Here is the command I used.

SELECT Status, Step FROM RMA WHERE OrderID = 5175;

Status = Pending

Step = Awaiting customer Documentation

* + 1. Write a 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?



Next task is to update the status and step for record 5175 to complete and credit customer account. I also used a select command to show the completed work. Here are the commands I used

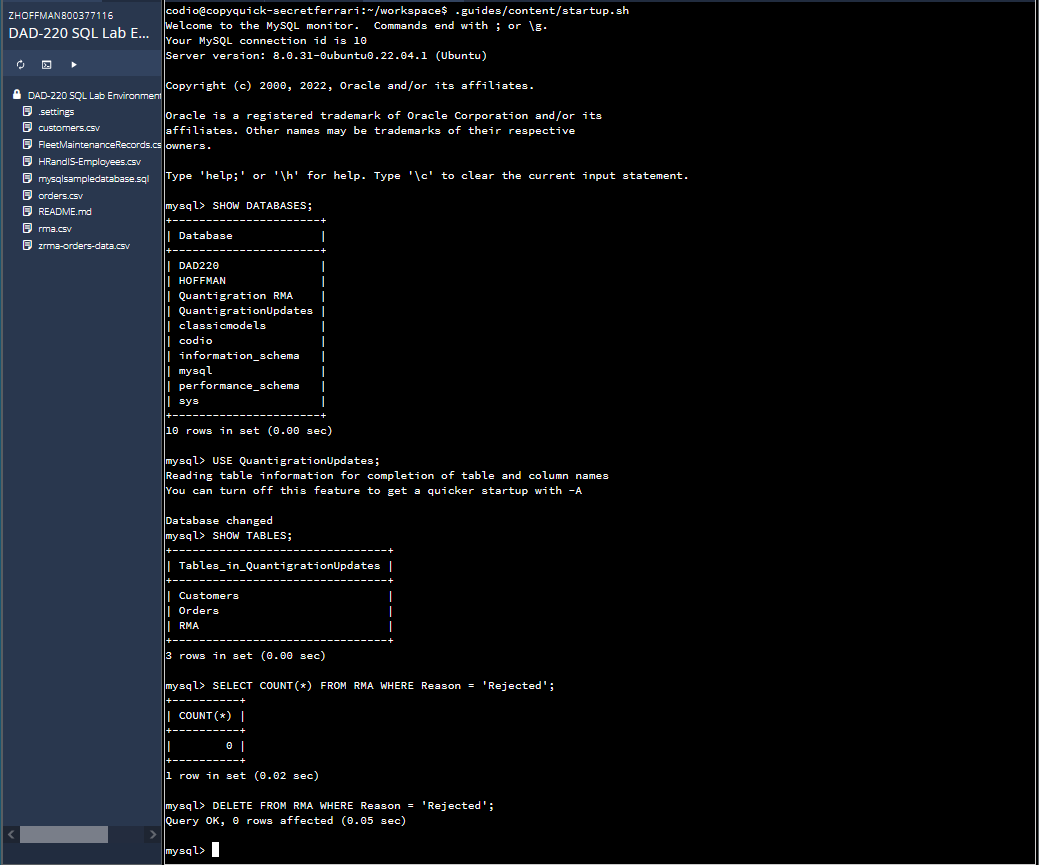
UPDATE RMA SET Status = 'Complete', Step = 'Credit Customer Account' WHERE OrderID = 5175;

SELECT Status, Step FROM RMA WHERE OrderID = 5175;

Status = Complete

Step = Credit Customer Account

* 1. Delete RMA records.
     1. Write a SQL statement to delete all records with a reason of "Rejected".
        1. How many records were deleted?



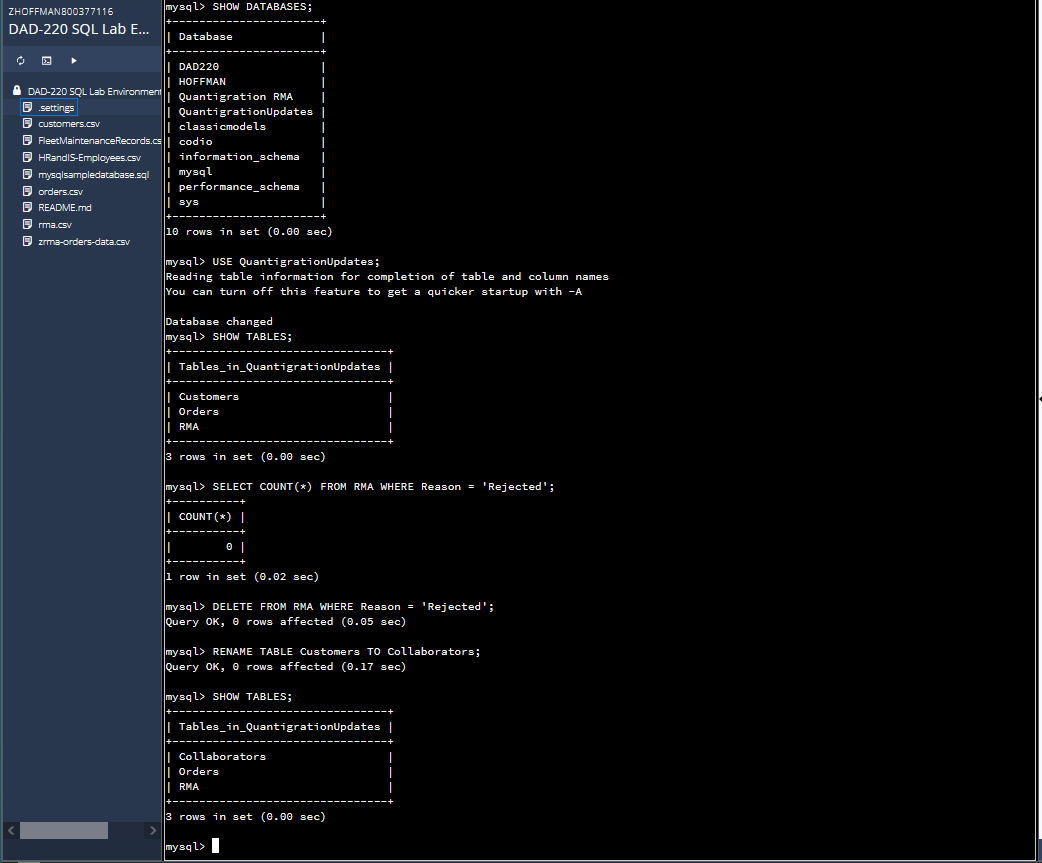
Here are the commands I used to delete records from the RMA table.

SELECT COUNT(\*) FROM RMA WHERE Reason = 'Rejected';

DELETE FROM RMA WHERE Reason = 'Rejected';

Output = 0 rows affected??

1. **Update your existing tables** from "Customer" to "Collaborator" using SQL based on this change in requirements. Copy and paste the SQL you write to do the following action:
   1. Rename all instances of "Customer" to "Collaborator".

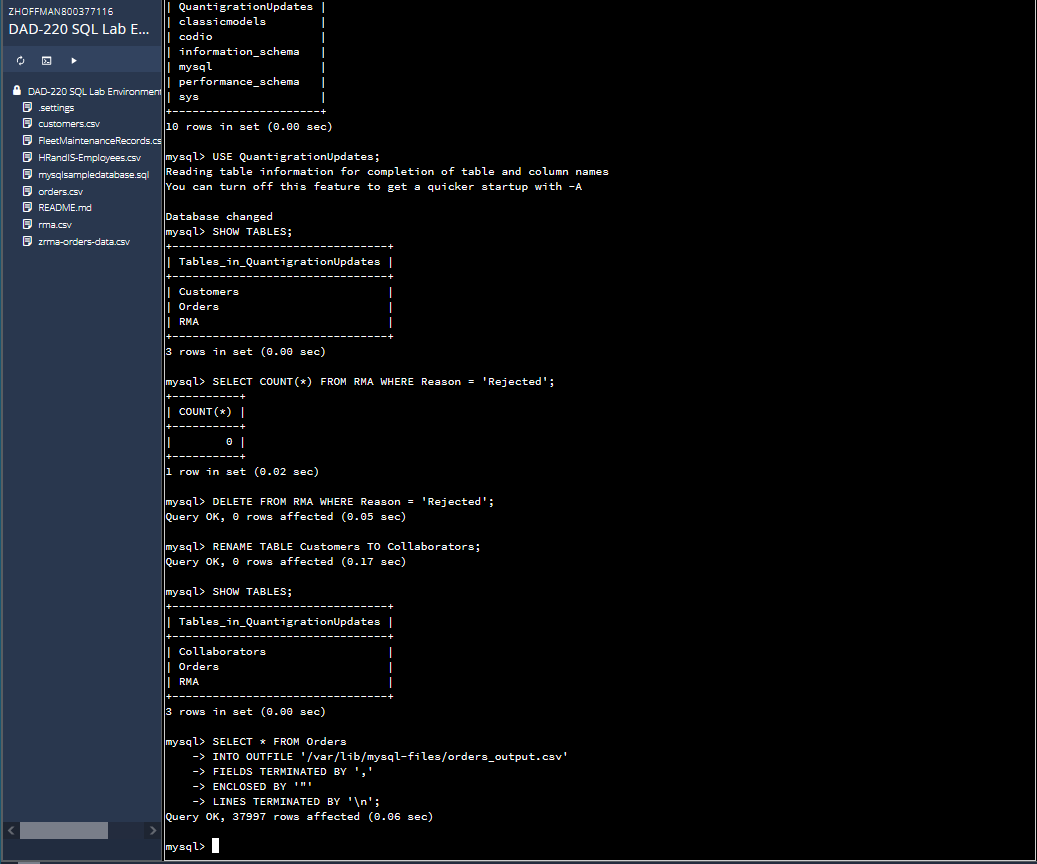


Here we were just asked to change the name of the Customers table to Collaborator. I also used a simple select command to show the completed work. These are the commands I used to complete this task

RENAME TABLE Customers TO Collaborators;

SHOW TABLES;

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



Last step is to create an output file of the orders table and send it to a file that has a CSV extension. Here is the command I used.

SELECT \* FROM Orders

INTO OUTFILE '/var/lib/mysql-files/orders\_output.csv'

FIELDS TERMINATED BY ','

ENCLOSED BY '"'

LINES TERMINATED BY '\n';