# Byron Laferriere

Professor Gupta

DAD-220

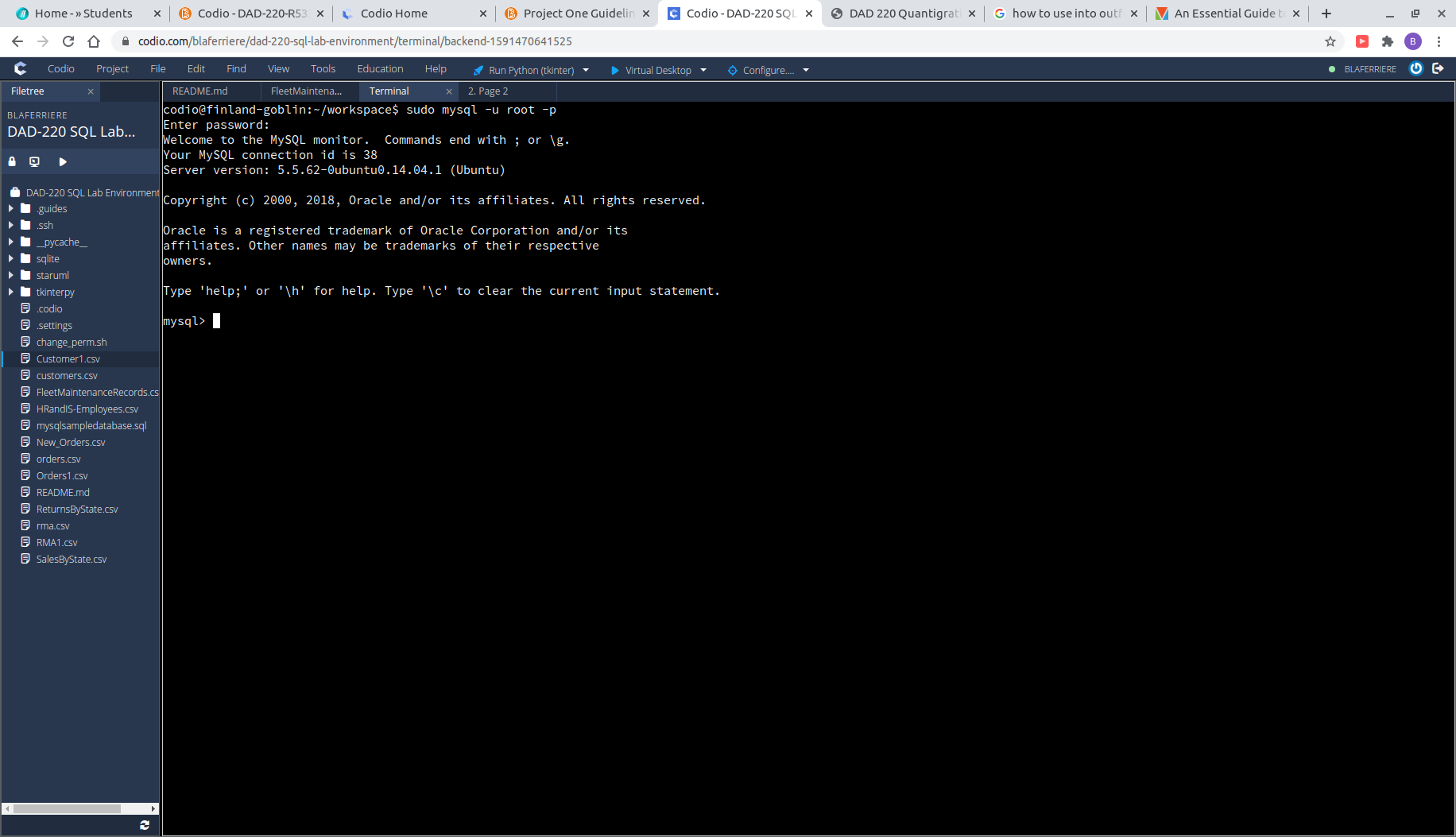
14 June 2020

# DAD 220 Database Documentation 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 located in the Project One Supporting Materials for assistance.

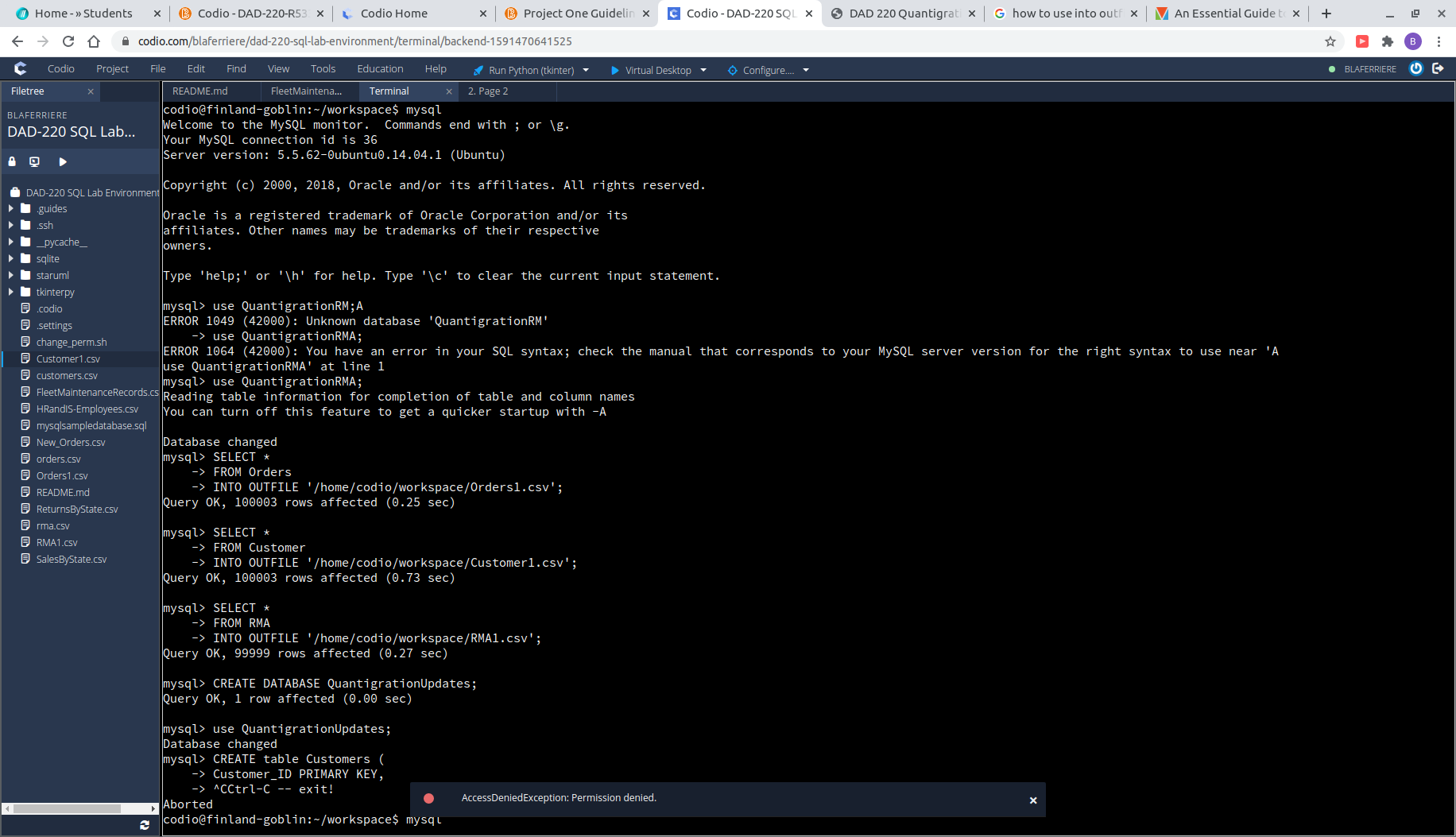
## Step One: Create a Database

1. Navigate to your online integrated development environment (IDE). Here, you will need to write the proper SQL commands in command line to create tables that demonstrate relationships based on the entity relationship diagram. List and record the SQL commands that you used to complete this step here:



Navigating to MySQL from the command line can be achieved by many different forms. On my Linux system I use ‘*sudo mysql -u root -p’* because it is password protected, but to launch MySQL on the Codio I.D.E. I just used ‘*mysql’* because it wasn’t protected by a password.

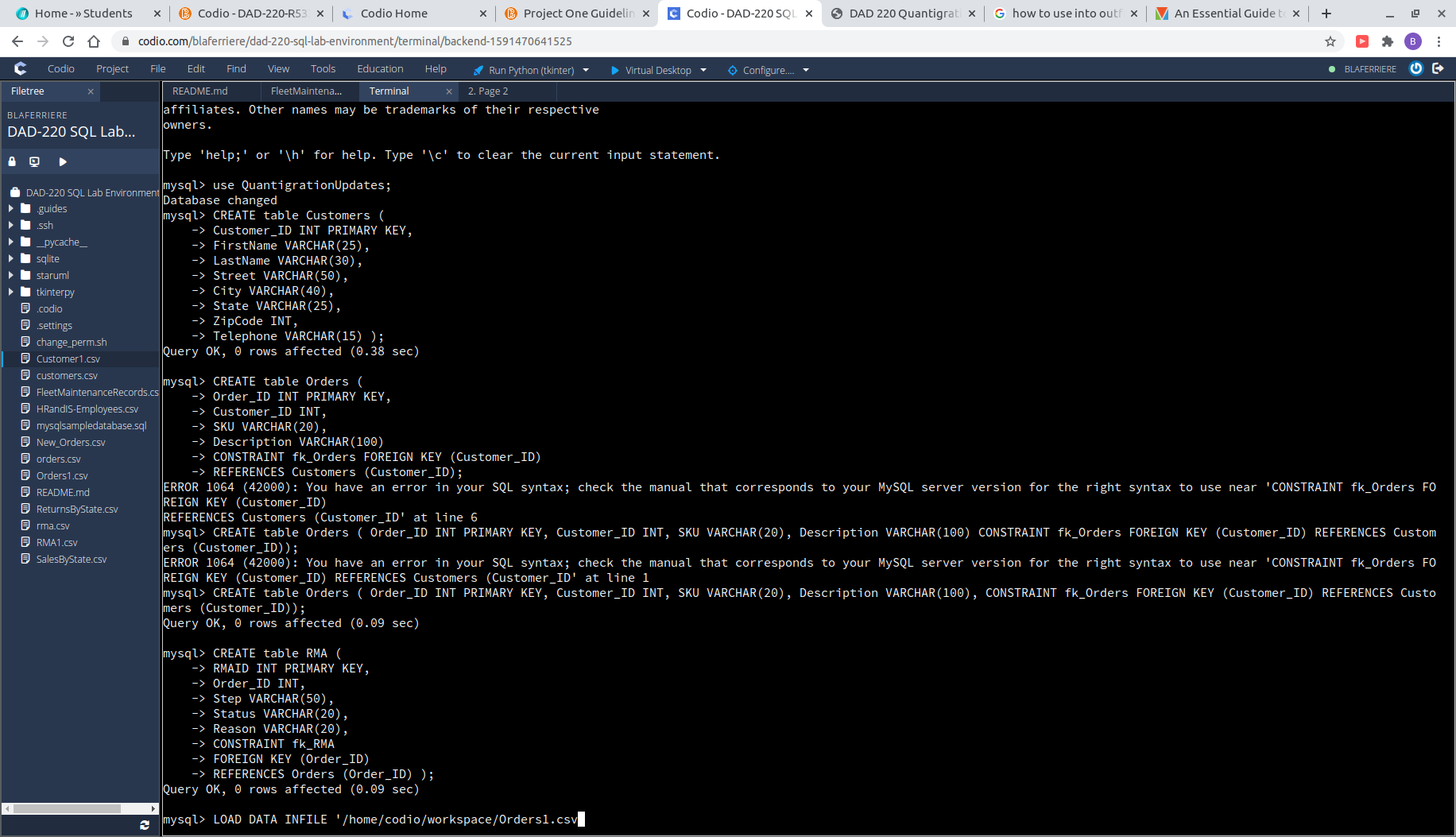
1. Create a database schema called *QuantigrationUpdates*. List out the database name. Provide the SQL commands you ran against MySQL to successfully complete this in your answer:



To create a new database once in MySQL, all you have to do is type ‘*CREATE DATABASE QuantigrationUpdates;’*  Typing ‘*SHOW databases;’* after will display the current schemas available to verify it has been created. Then enter ‘*USE QuantigrationUpdates;’*  to enter the new database.

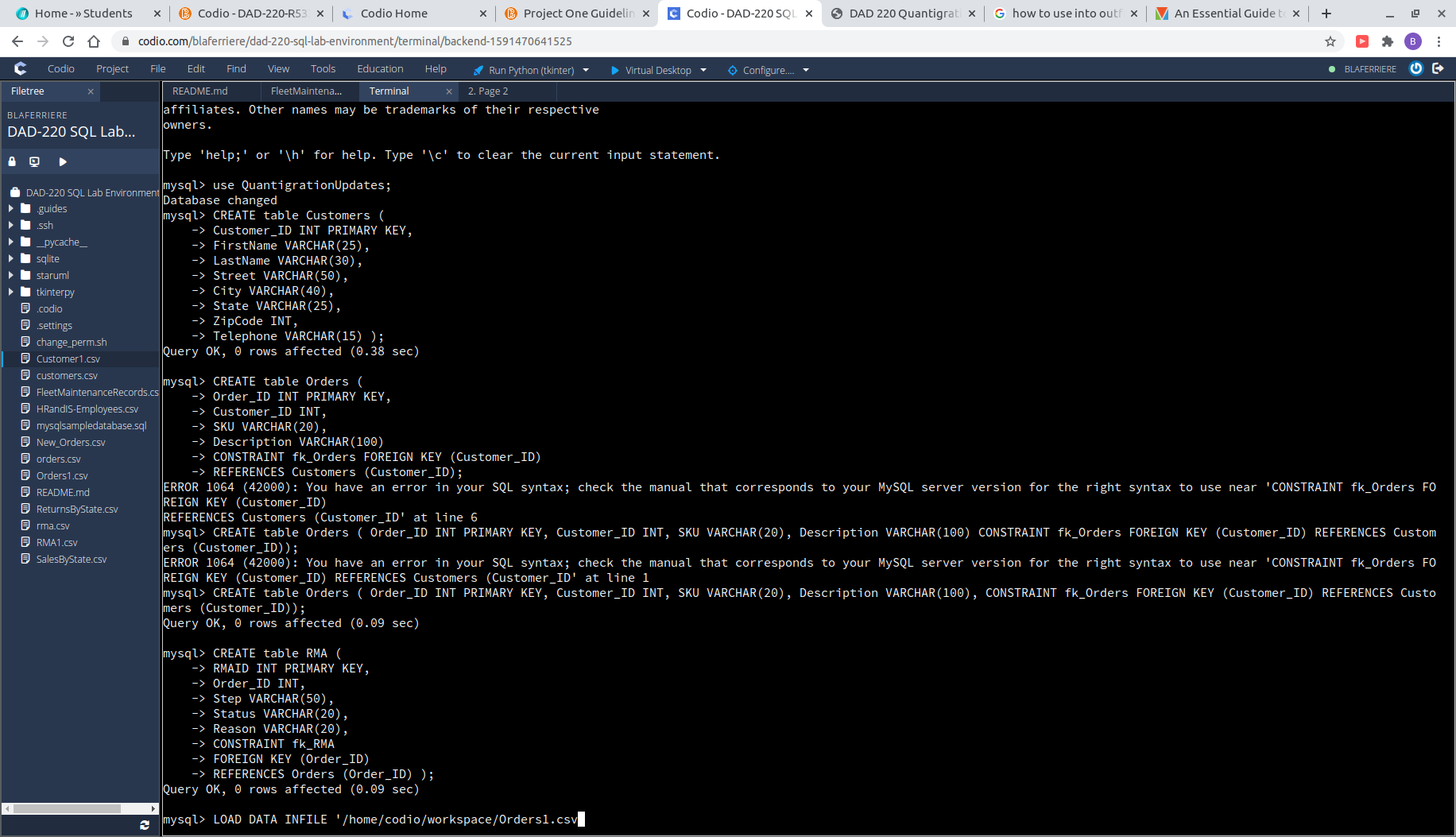
Using the ERD as a reference, **create the following tables with the appropriate attributes and keys**:

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



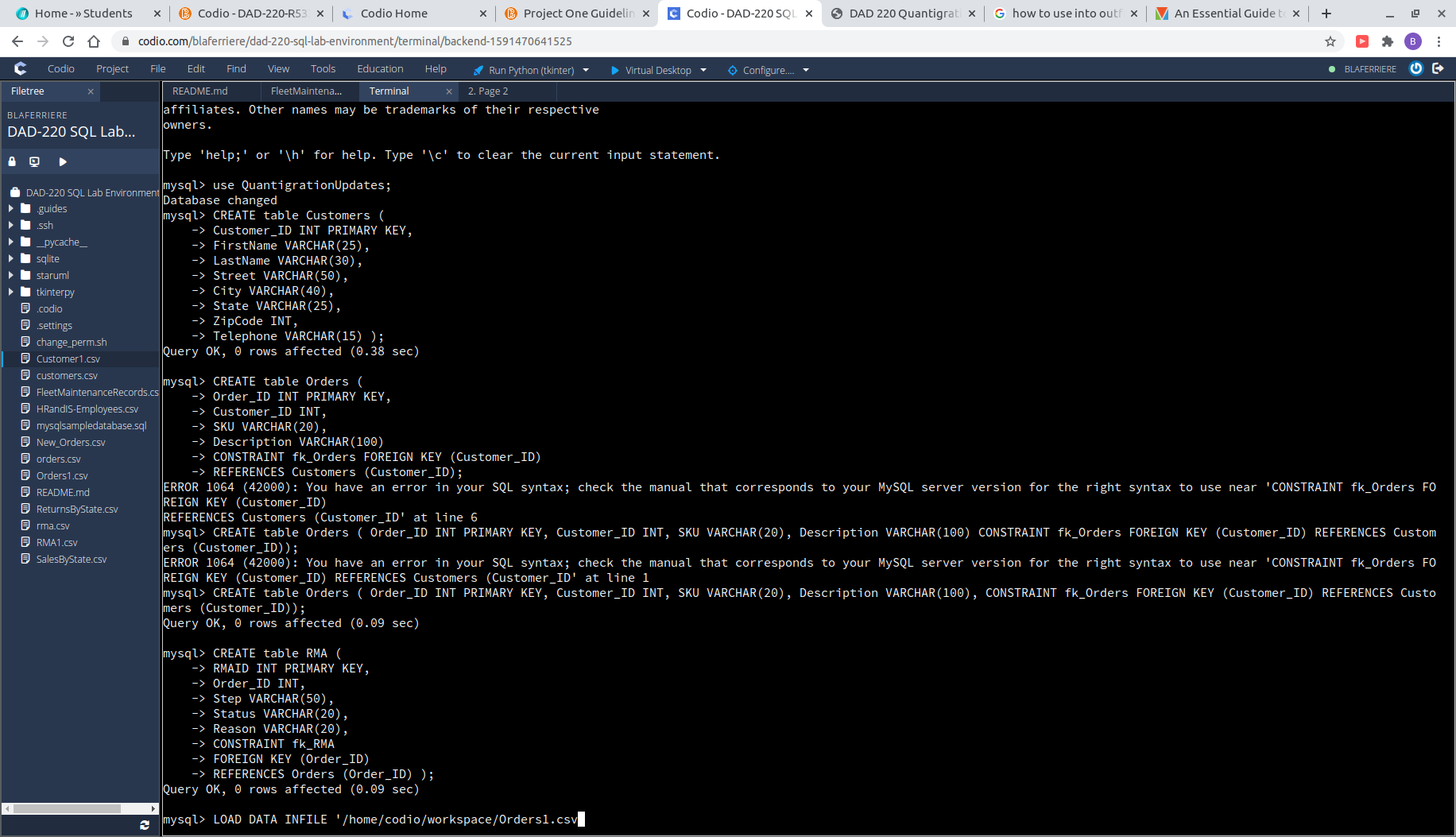
Following the ERD, we see that Customer\_ID needs to be the PRIMARY KEY of this first table. Using ‘*CREATE table Customers ( ‘*  to define all of our columns, allows us to create our first table.

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



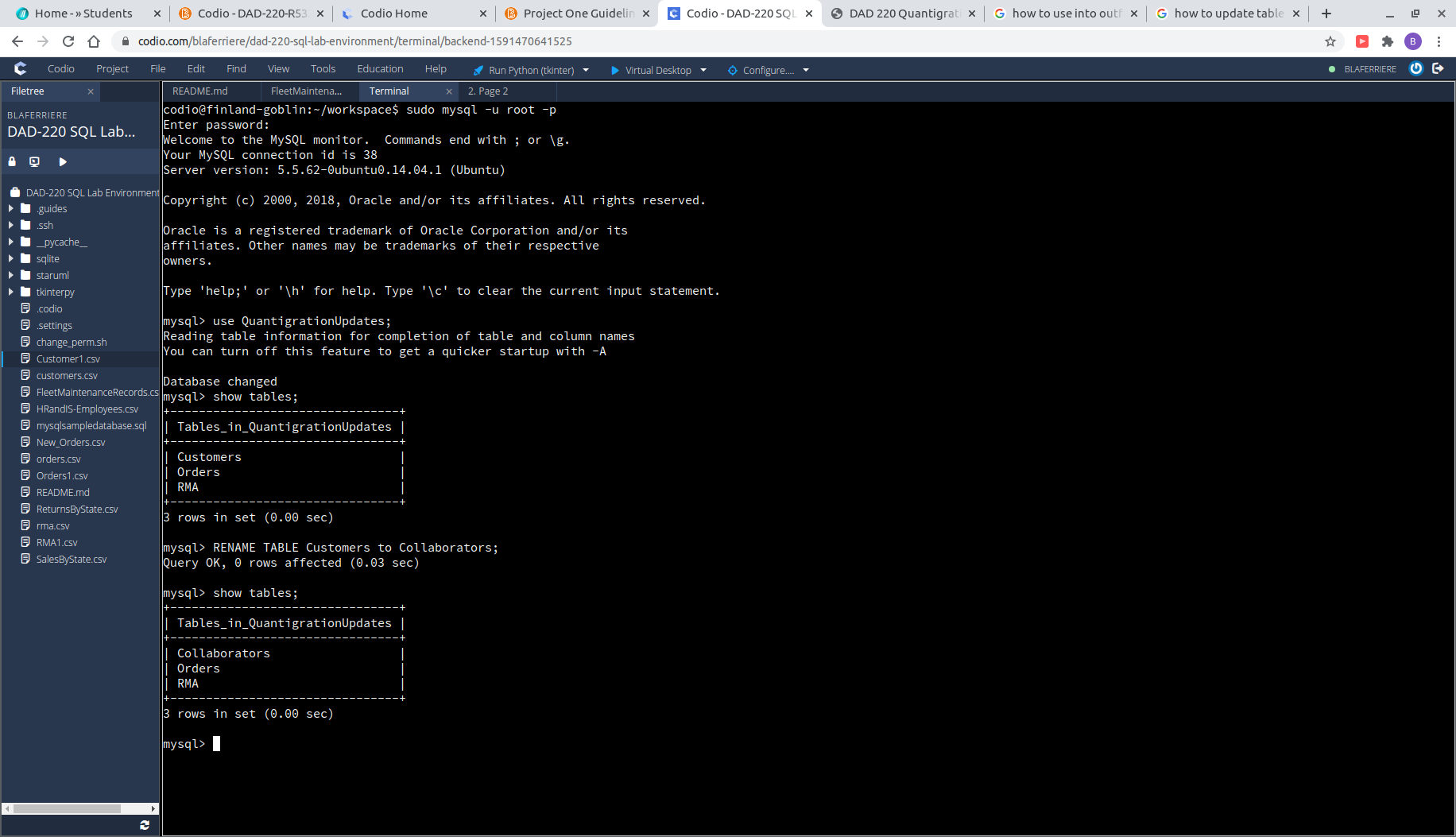
As progression is made through the ERD, each table is connected to the table before it. This means the Orders table will have a PRIMARY KEY and a FOREIGN KEY connected to the Customers table. PRIMARY can be assigned when defining the original column, but FOREIGN requires some extra terminology. My preferred method is to name ‘*CONSTRAINT fk\_(tbl\_name) FOREIGN KEY (columnID) REFERENCES parent\_tbl (columnID);’*

* 1. A table named **rma** in the *QuantigrationUpdates* database as defined on the project ERD. Provide the SQL commands you ran against MySQL to complete this successfully in your answer:



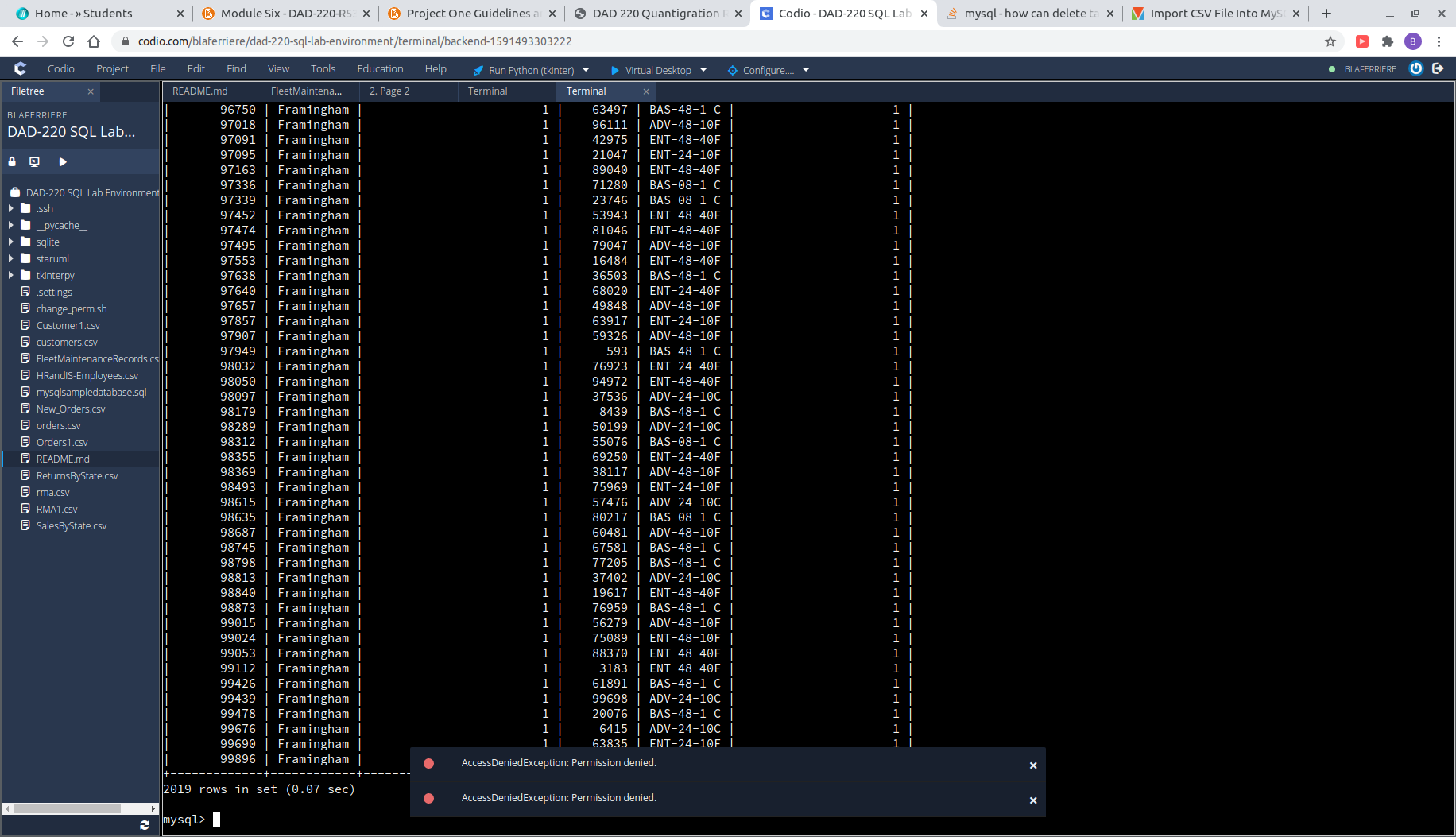
The third and final table in our schema also has a PRIMARY and FOREIGN key assignment. The same method was used to create this table that was used in the creation of Orders.

1. **Update your existing table** 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:

****

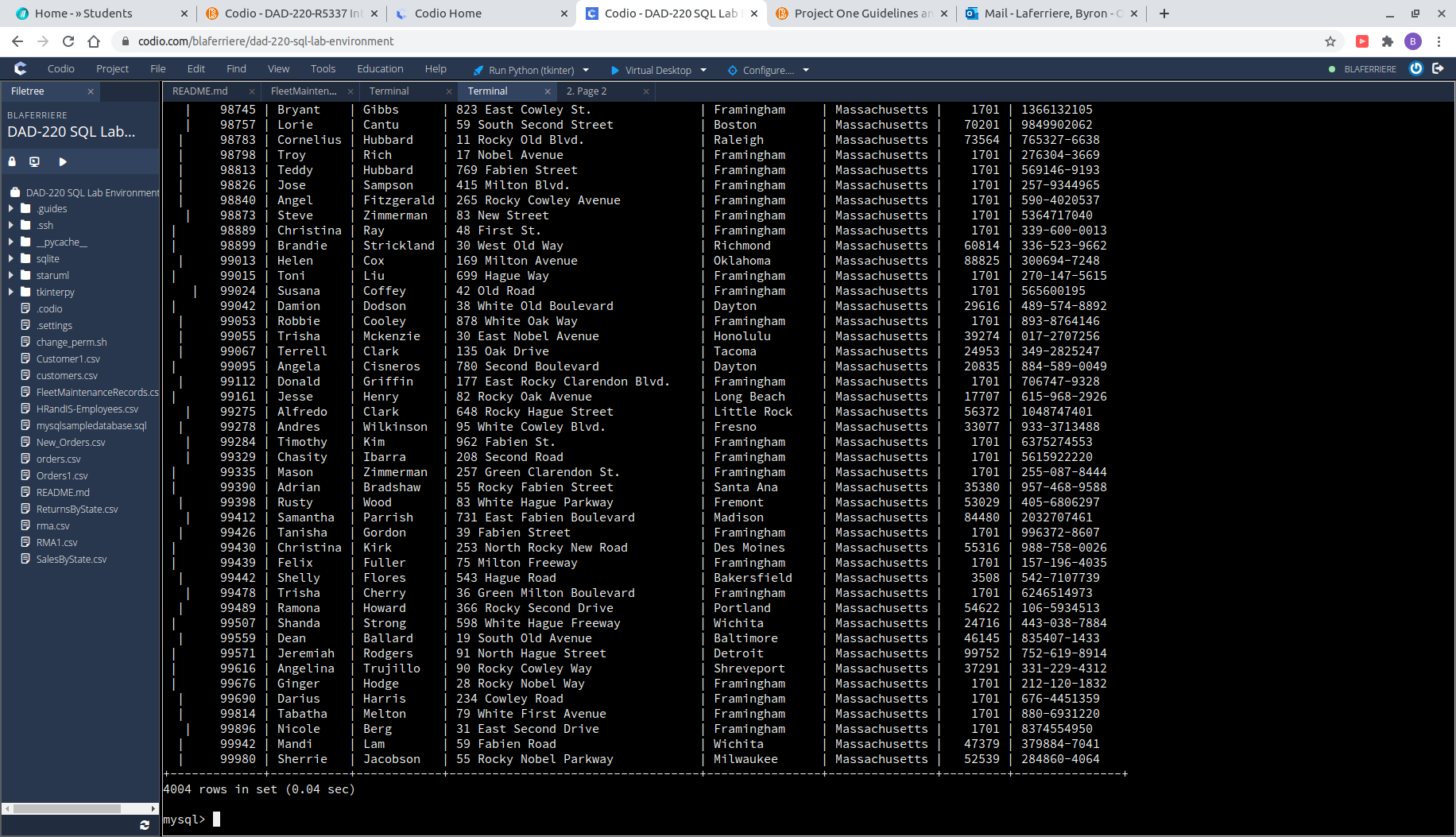
Updating the name of the table is an easy change in MySQL. The syntax needed to be used is : ‘*RENAME TABLE tbl1 to tbl2;’.*  Using ‘*show tables;’* will verify the changes have been made.

## Step Two: Load and Query the Data

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.
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.
     1. How many records were returned?

This was a tougher challenge for me in Module four, but this time around I have figured out the best syntax to use for JOIN’s. We can clearly see all 2,019 orders and locations of those orders displayed by the SQL query used. The lesson I learned was to eliminate ambiguity in calling your columns.

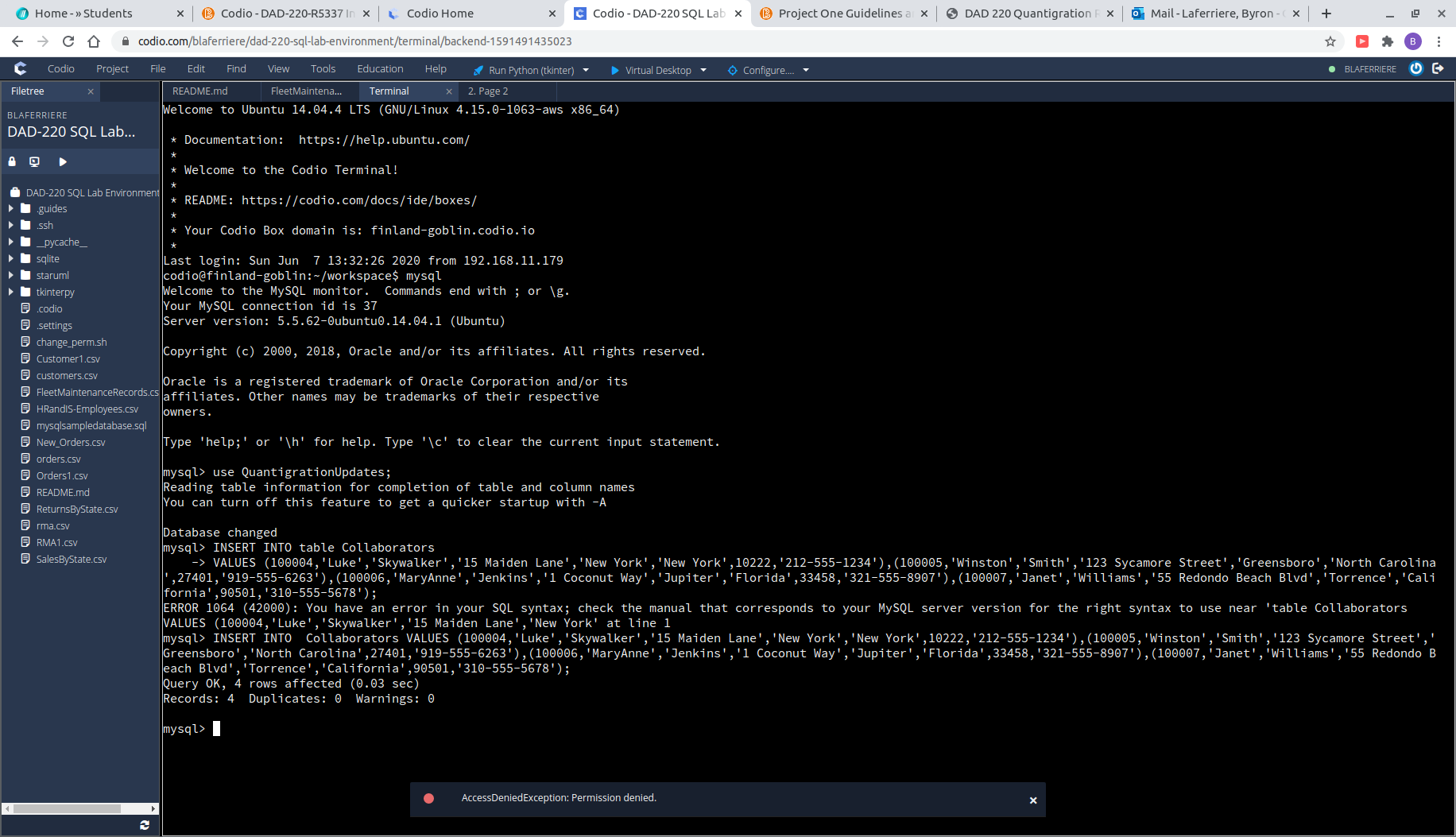
* + Write an SQL query to select all of the customers located in the state of Massachusetts.
    1. Use a WHERE clause to limit the number of records in the customers table to only those who are located in Massachusetts.
    2. Record an answer to the following question: How many records were returned?



The state of Massachusetts has 4,004 customers in the state. The WHERE clause easily eliminates all customers that don’t match the search in the output.

* + Write a SQL query to insert four new records into the orders and customers tables using the following data:
    1. Customers Table

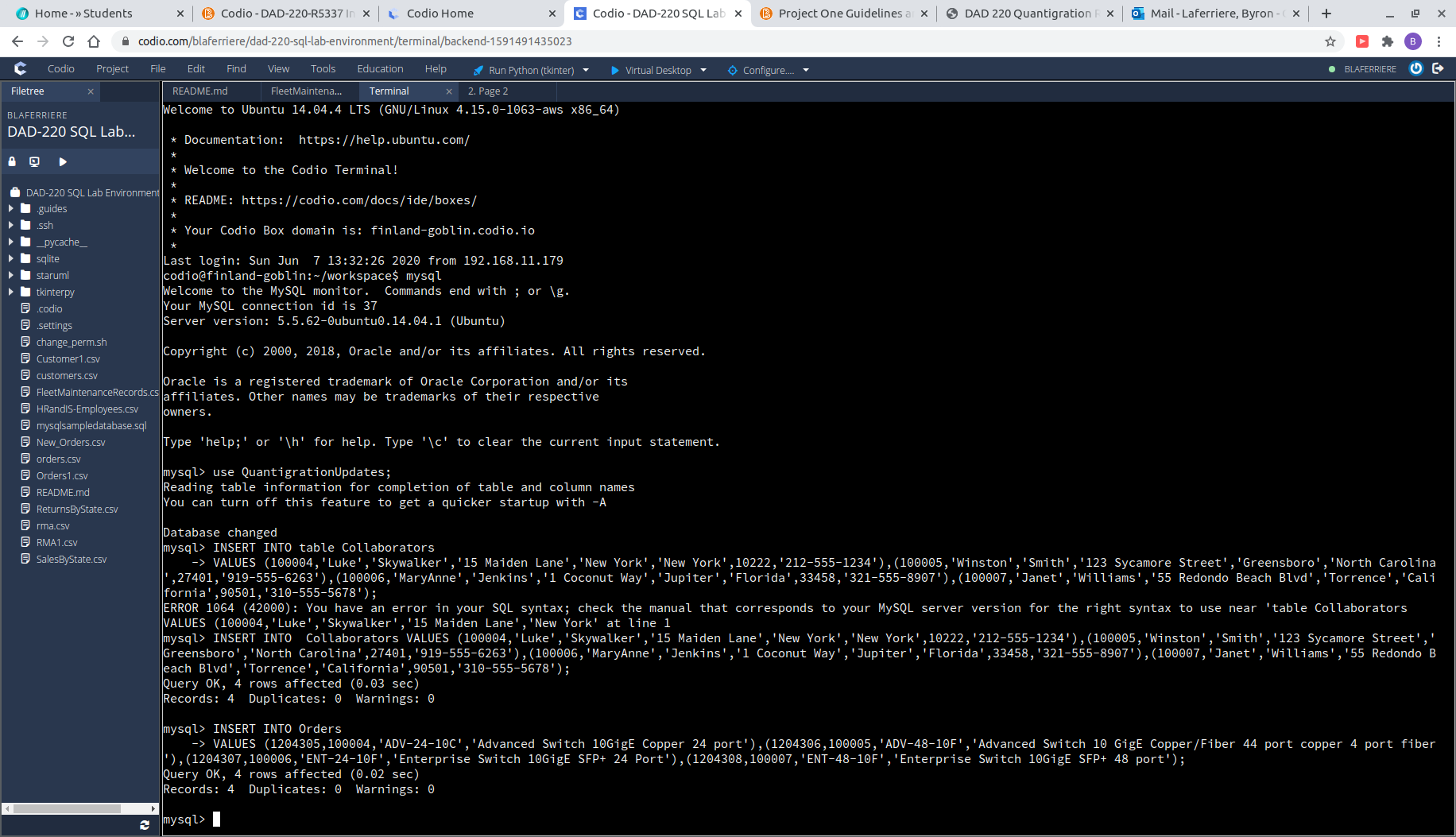
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **CustomerID** | **FirstName** | **LastName** | **StreetAddress** | **City** | **State** | **ZipCode** | **Telephone** |
| 100004 | Luke | Skywalker | 15 Maiden Lane | New York | NY | 10222 | 212-555-1234 |
| 100005 | Winston | 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 |



Using INSERT INTO statements have become second-nature for me during this class. In module four, I did this step one by one to ensure I didnt mess up my syntax. This time I typed it all out at once, feeling more comfortable.

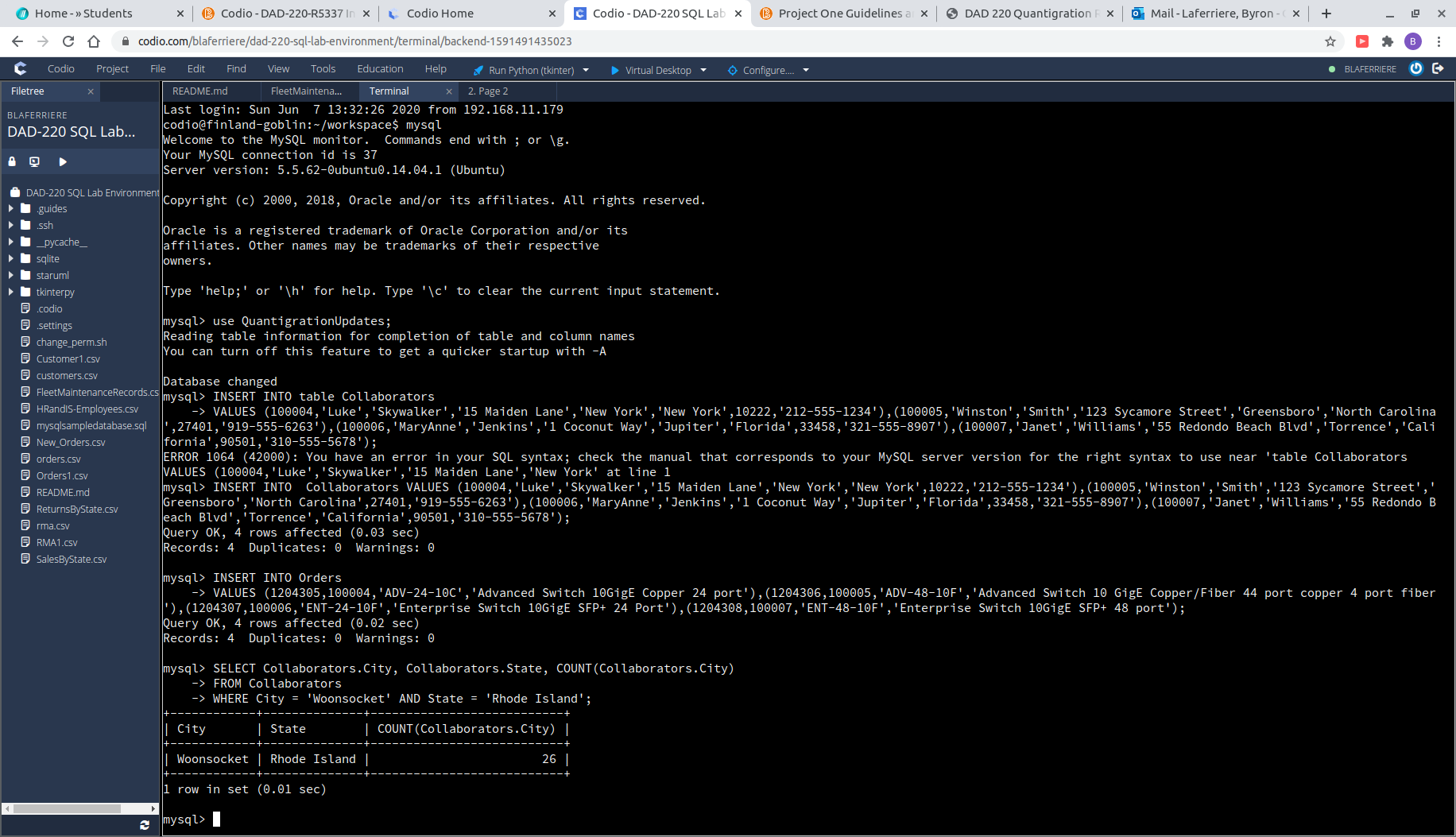
* + 1. Orders Table

|  |  |  |  |
| --- | --- | --- | --- |
| **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 |



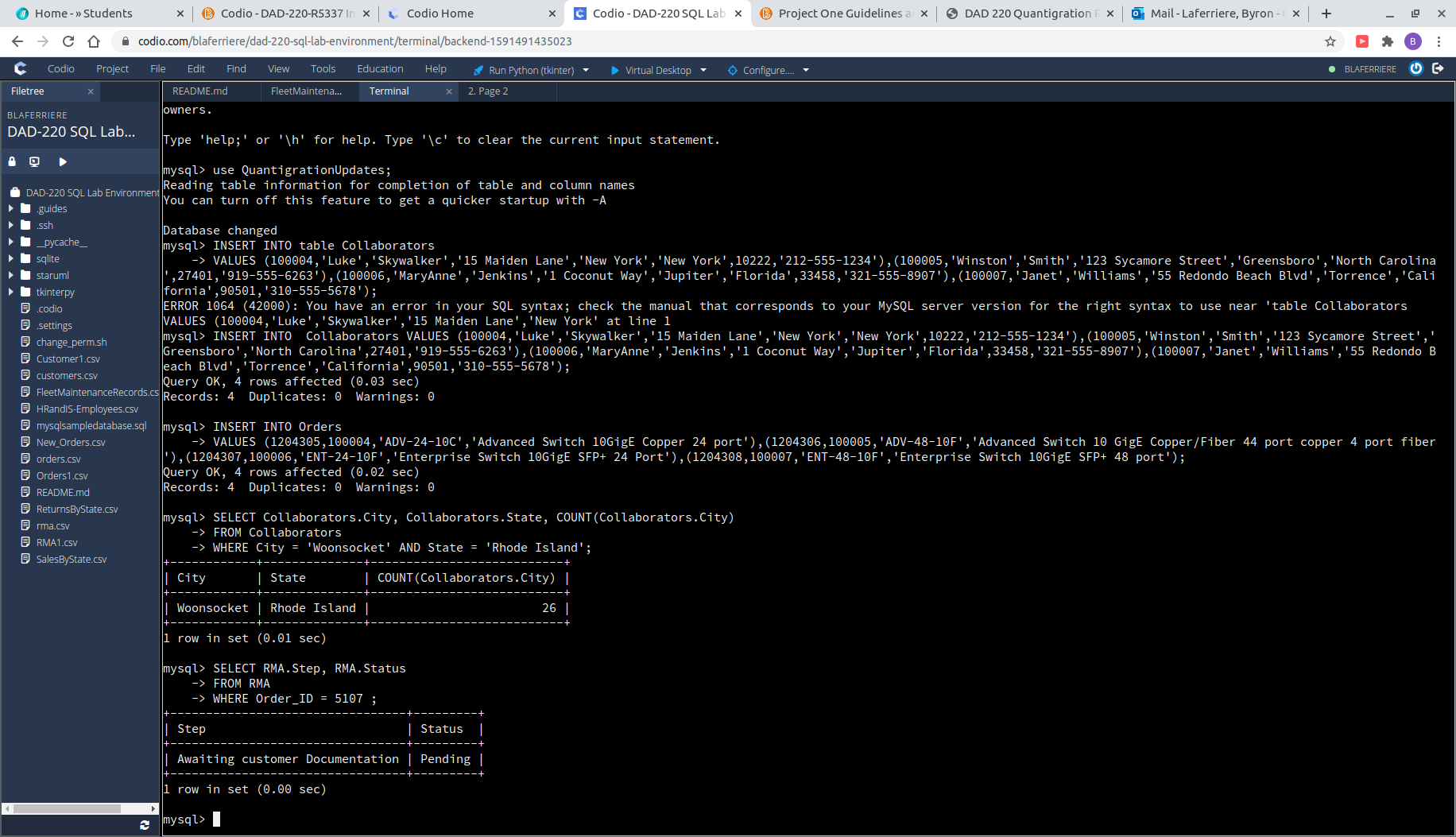
This step was also similar to the previous step of entering customer records. The most important thing to remember is ensuring that the column data matches up.

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



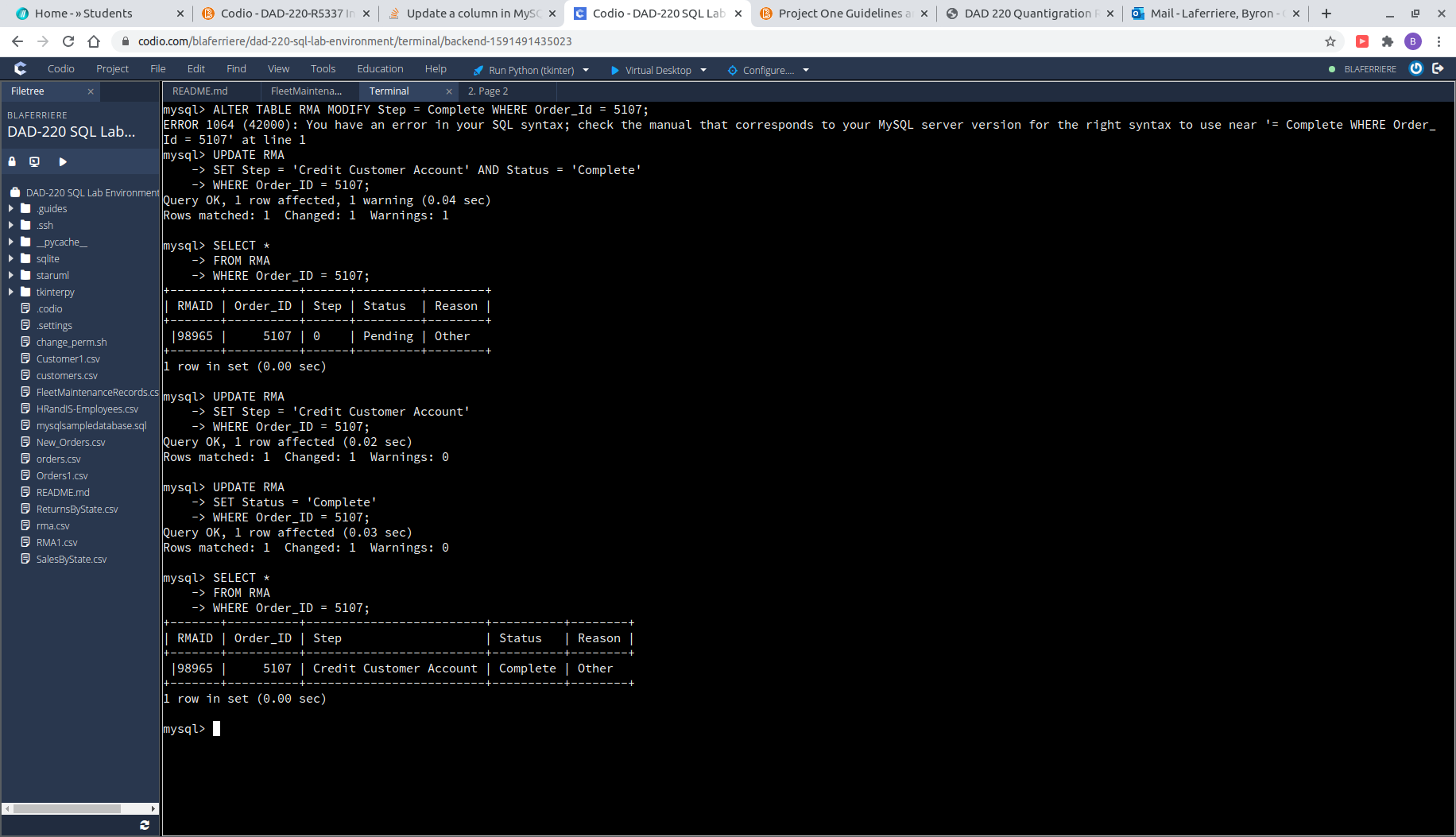
The customer table holds 26 records for customers from the city of Woonsocket, Rhode Island. Although not necessary, I used a WHERE .. AND .. to limit the query to the city of Woonsocket and the state of Rhode Island.

* + In the rma database, update a customer’s records.
    1. 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?



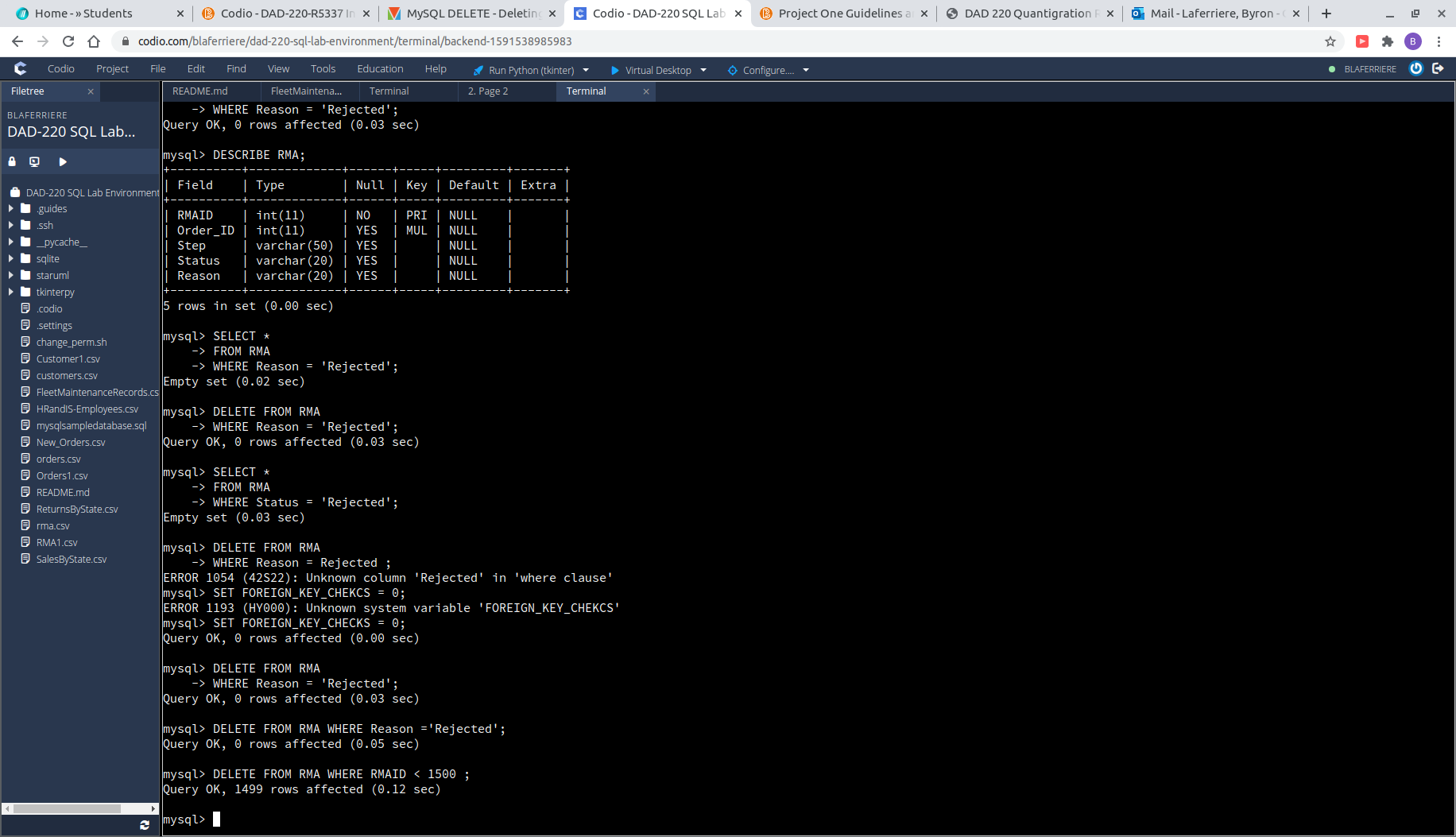
This is a simple two-column query in the RMA table. Combining that with WHERE and the value requested, limited the results to the exact order.

* + 1. 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?



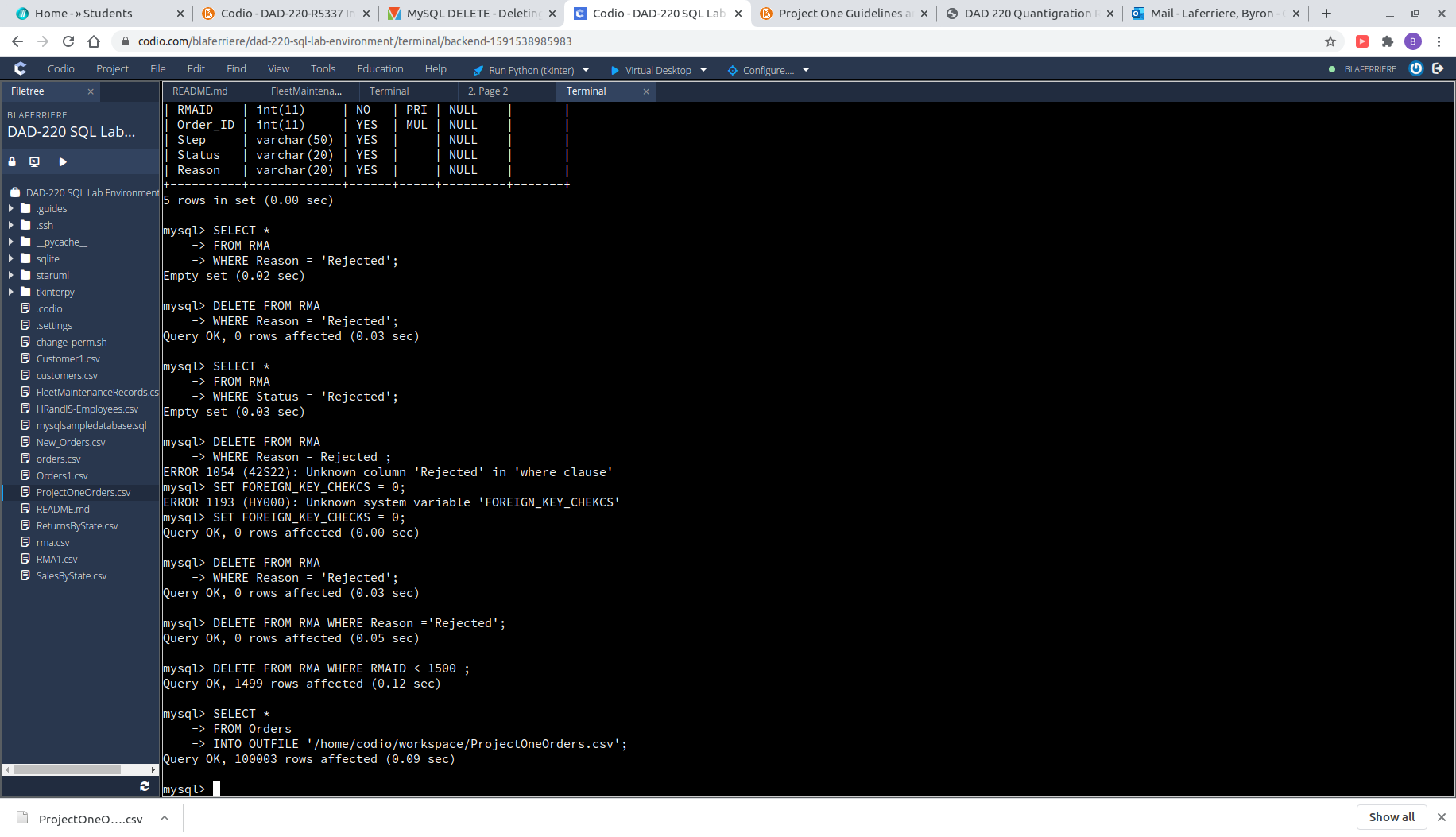
Now that the step and status of Order\_ID 5107 have been discovered, we can update the columns to the desired values. Using ‘UPDATE tbl\_name SET column = ‘’ WHERE OrderID = ‘ allowed for both columns to easily be replaced. The WHERE condition allows us to specify columns to be selected.

* + Delete rma records.
    1. Write an SQL statement to delete all records with a reason of “Rejected.”
       1. How many records were deleted?



I wish I would have caught this one the first time around, but due to primary and foreign keys attached to this table I never deleted the rejected files. By deleting by the RMAID PRIMARY KEY and finding the files associated with “rejected” that way, I was able to DELETE 1499 files.

1. **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.

****

I have been practicing my INTO OUTFILE syntax all week, learning how to avoid printing out large output caused by JOIN. This ending command was the easiest step of all. *‘SELECT \* FROM Orders INTO OUTFILE ‘/home/codio/workspace/file-name-chosen.csv’’;*