# 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). List and record the SQL commands that you used to complete this step here:

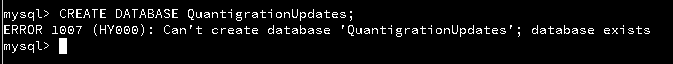
A screenshot of a computer

Description automatically generated

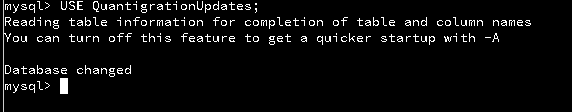
Mysql

* I use the command mysql when I start up Codio to enter the correct IDE to start using sql to create or alter databases.

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:



CREATE DATABASE QuantigrationUpdates;



USE QuantigrationUpdates;

* Once I start mysql and im in the right environment I start my creating the required database called QuantigrationUpdates by using the CREATE command in front of the databases name, once the database is created I call USE on the database so I can navigate to and start creating tables within my specific database.

1. Using the entity relationship diagram (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:

A screenshot of a computer program

Description automatically generated

CREATE TABLE Customers(CustomerID INT, FirstName VARCHAR(25), LastName VARCHAR(25), StreetAddress VARCHAR(50), City VARCHAR(50), State VARCHAR(25), ZipCode VARCHAR(15), Telephone VARCHAR(15), PRIMARY KEY(CustomerID));

SHOW TABLES;

A screenshot of a computer screen

Description automatically generated

DESCRIBE Customers;

* 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:

A screenshot of a computer program

Description automatically generated

CREATE TABLE Orders ( OrderID INT, CustomerID INT, SKU VARCHAR(20), Description VARCHAR(50), PRIMARY KEY(OrderID), FOREIGN KEY(CustomerID) REFERENCES Customers(CustomerID));

SHOW TABLES;

A black screen with white text

Description automatically generated

DESCRIBE Orders;

* 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:

A screenshot of a computer program

Description automatically generated

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));

SHOW TABLES;

A black screen with white text

Description automatically generated

DESCRIBE RMA;

* The three steps above above were using to create three tables, Customer,Orders, and RMA using specific values given by the RMA Diagram and then I listed each table with there descriptions using the DESCRIBE command.

## 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.

A computer screen shot of a black screen

Description automatically generated

LOAD DATA INFILE '/home/codio/workspace/customers.csv' INTO TABLE Customers FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n';

A computer screen with white text

Description automatically generated

LOAD DATA INFILE '/home/codio/workspace/orders.csv' INTO TABLE Orders FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n';

A screenshot of a computer error

Description automatically generated

LOAD DATA INFILE '/home/codio/workspace/rma.csv' INTO TABLE RMA FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n';

* The three commands above are loading data and storing that data in our already created tables, so in other words we are populating our tables with data from an external source.

1. **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 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?
        1. There were 505 records returned for the number of customer orders in Framingham, Massachusetts.

A black screen with white text

Description automatically generated

SELECT COUNT(\*) FROM Customers INNER JOIN Orders ON Customers.CustomerID = Orders.CustomerID WHERE UPPER(Customers.city) = 'FRAMINGHAM' AND UPPER(Customers.state) = 'MASSACHUSETTS';

* + The SQL query provided counts the number of records where customers from Framingham, Massachusetts, have corresponding orders in the "Orders" table. The condition is specified using an inner join on the "CustomerID" column between the "Customers" and "Orders" tables, filtering for customers with uppercase "city" and "state" values matching 'FRAMINGHAM' and 'MASSACHUSETTS', respectively. The output of this query will be a single value representing the count of such customer-order pairs meeting the specified criteria.
  + 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?
       - According the data in the table below there are 982 people that are from Massachusetts.

A computer screen with white text

Description automatically generated

SELECT COUNT(\*) FROM Customers WHERE UPPER(Customers.state) = 'MASSACHUSETTS';

* + - * The output of this SQL query will be a numerical count representing the total number of customer records in the "Customers" table where the state is specified as 'MASSACHUSETTS'. In other words, it provides the quantity of customers associated with the state of Massachusetts based on the given uppercase comparison.
  + Write a SQL query to insert four new records into the Orders and Customers tables using the following data:

**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 |

A screen shot of a computer

Description automatically generated

INSERT INTO Customers (CustomerID, FirstName, LastName, StreetAddress, City, State, ZipCode, Telephone) VALUES (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', 'Torrance', 'CA', '90501', '310-555-5678');

* + - * I use the INSERT INTO command to add any values into a specific table followed by the VALUES command and the values that will be entered into the table as long as the values line up with the parameters that were originally given to the table. With this specific table I am adding 4 values that all have CustomerID, FirstName, LastName, StreetAddress, City, State, ZipCode, Telephone.

A screenshot of a computer screen

Description automatically generated

SELECT \* FROM Customers WHERE CustomerID IN (100004, 100005, 100006, 100007);

* + - * The table above shows us that we have successfully added the 4 values correctly.

**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 |

A black screen with white text

Description automatically generated

INSERT INTO Orders ( OrderID, CustomerID, SKU, Description) VALUES (1204305, 100004, 'ADV-24-10C', 'Advanced Switch 10GigE Copper 24 port'), (1204306, 100005, 'ADV-48-10F', 'Advanced Switch 10GigE 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');

* + - * I use the INSERT INTO command to add any values into a specific table followed by the VALUES command and the values that will be entered into the table as long as the values line up with the parameters that were originally given to the table. With this specific table I am adding 4 values that all have OrderID, CustomerID, SKU, and Description.

A screenshot of a computer program

Description automatically generated

SELECT \* FROM Orders WHERE OrderID IN (1204305, 1204306, 1204307, 1204308);

* + - * The table above shows us that we have successfully added the 4 values correctly.
  + 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”?
       - There are 7 values that are returned for customers in Woonsocket.

A screenshot of a computer program

Description automatically generated

SELECT COUNT(\*) FROM Customers WHERE UPPER(Customers.city) = 'WOONSOCKET';

* + 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?
       - The current step and status of the value with the orderID of 5175 is Status = Pending and Step = Awaiting Customer Documentation.

A computer screen with white text

Description automatically generated

SELECT \* FROM RMA WHERE OrderID = 5175;

* + 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?
       - The updated status is Complete and the updated step is Credit Customer Account.

A black screen with white text

Description automatically generated

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

A screen shot of a computer screen

Description automatically generated

SELECT \* FROM RMA WHERE OrderID = 5175;

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

A black screen with white text

Description automatically generated

DELETE FROM RMA WHERE UPPER(Reason) LIKE '%REJ%';

1. **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 black background with white text

Description automatically generated

RENAME TABLE Customers TO Collaborators;

A screenshot of a computer program

Description automatically generated

SHOW TABLES;

* 1. Rename all instances of “Customer” to “Collaborator.”

A screen shot of a black screen

Description automatically generated

ALTER TABLE Orders DROP FOREIGN KEY Orders\_ibfk\_1;

A black screen with white text

Description automatically generated

ALTER TABLE Collaborators DROP PRIMARY KEY;

A black screen with white text

Description automatically generated

ALTER TABLE Collaborators CHANGE CustomerID CollaboratorID INT;

A black screen with white text

Description automatically generated

ALTER TABLE Collaborators ADD PRIMARY KEY(CollaboratorID);

A black screen with white text

Description automatically generated

ALTER TABLE Orders CHANGE CustomerID CollaboratorID INT;



ALTER TABLE ORDERS ADD FOREIGN KEY(CollaboratorID) REFERENCES Collaborators (CollaboratorID);

A screenshot of a computer screen

Description automatically generated

DESCRIBE Collaborators;

* + - * First, we used some SQL code to do a few things with our database. We removed a special link between tables called a foreign key from the Orders table, and also took away the main way we identified each row in the Collaborators table, which could make it tricky to tell them apart. Then, we changed the name of a column in the Orders table from "CustomerID" to "CollaboratorID" and made it a number. We did the same thing in the Collaborators table. After that, we added a new way to uniquely identify each row in the Collaborators table using a column called "CollaboratorID." Then, we created a connection between the Orders and Collaborators tables, saying that the "CollaboratorID" in Orders should match the one in Collaborators. When we checked what the Orders table looked like with "Describe Orders," we saw that "CustomerID" had become "CollaboratorID." Checking the Collaborators table showed that there was no column named "CustomerID" anymore. Lastly, when we tried to see information about a table called "Customers," it told us that the table doesn't exist. This confirms that we successfully changed all mentions of "Customer" to "Collaborator."

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.

A screen shot of a computer code

Description automatically generated

SELECT \* FROM Orders INTO OUTFILE '/home/codio/workspace/orders-data.csv' FIELDS TERMINATED BY ',' LINES TERMINATED BY '\n';

* + - * This SQL code is an example of exporting the contents of the Orders table to a CSV file named orders-data.csv. The data will be formatted with fields separated by commas and lines terminated by newline characters. The MySQL query executes successfully and affects 37998 rows, which suggests that there are 37998 records in the Orders table. The resulting CSV file likely contains a snapshot of the Orders table's data, where each row represents an order and columns represent various attributes associated with each order.

A screenshot of a computer

Description automatically generated

* + - * Here is the proof that the file was successfully outputted into a separate csv file called orders-data.csv.