# DAD 220 Module Four Lab Template

## Overview

Begin by doing the following steps in the IDE (Codio):

1. Load the classicmodels data set.
2. Start a new terminal session and run this command: mysqlsampledatabase.sql
3. Write commands to use the classicmodels database and show its tables to verify that you are in the right place.

Then perform the steps below to complete the lab. Manually enter any commands you are asked to write. At the end of each step, replace the bracketed text in this template with your screenshot, response, or both, as indicated. Submit your completed template for grading and feedback. Screenshots should be sized to about one-quarter of a page. Written responses should be in complete sentences. Rename this document by adding your last name to the file name before you submit it.

## Identify Cardinality and Table Relationships

1. **Retrieve employee tuples and identify the number of employees** in San Francisco and New York.
   1. Command for San Francisco: select firstName, lastName, jobTitle, offices.city from employees inner join offices on employees.officeCode = offices.officeCode where state = 'CA'.
   2. Write and run a command to return records from New York on your own.
   3. Validate the completion of this step with a screenshot of these two tables.

A screenshot of a computer

Description automatically generated

1. **Retrieve order details** for orderNumber 10330, 10338, and 10194 and **identify** what **type of cardinality** this represents in the entity relationship model.
   1. Retrieve the order details by running SELECT queries with WHERE clauses against the Orders table.

I’m including two screen shots just in case one is wanted, and the other is to much because based on the question the request above.

A screen shot of a computer

Description automatically generated

A black screen with many small squares

Description automatically generated with medium confidence

The second screen shot has 326 rows and is a union of the tables on the orderNumber using the same where clause.

* 1. Now, identify what type of cardinality this represents in the entity relationship model.
     1. Reference the Module Four Lab ERD diagram linked in the guidelines and rubric to help identify relationships.

A diagram of a number of orders

Description automatically generated with medium confidence

The cardinality this represents in the entity relationship model based on the ERD diagram above is one from the orders table to zero or many when connecting to the orderdetails table. As each orderNumber can have a lot of order details associated with it as can be seen in the screen shots above.

1. **Delete records** from the payments table where the customer number equals 103.
   1. Run a DESCRIBE statement to identify fields in the Payments table first.
   2. Select the records from the Payments table for customer number 103 before deleting them.

A screenshot of a computer screen

Description automatically generated

* 1. Delete the records from the Payments table for customer number 103.
  2. Run a SELECT statement against the table to show that customer number 103 is no longer there.

A black screen with white text

Description automatically generated

1. **Retrieve customer records** for sales representative Barry Jones and **identify** if the **relationships** are one-to-one or one-to-many**.**
   1. Remember: SELECT, FROM, INNER JOIN, and WHERE.
   2. Use Barry’s employeeNumber, 1504, and perform a join between the customer salesRepEmployeeNumber to retrieve these records.
      1. Identify whether these entities demonstrate one-to-one or many-to-many relationships.

A screen shot of a computer

Description automatically generated

The relationship for these entities in this statement is many to one. While the entities themselves are one to many, this is a case where one sales rep is associated with many different customers in the query.

1. **Retrieve records** for customers who reside in Massachusetts and **identify** **their sales rep and the relationship of entities**. Identify if these entities demonstrate one-to-one or many-to-many relationships.
   1. Remember: SELECT, FROM, INNER JOIN, and WHERE.
   2. Use employee.firstName and employee.lastName in your command.
   3. Identify whether these entities demonstrate one-to-one or many-to-many relationships.

A screen shot of a computer screen

Description automatically generated

In this particular statement we are looking at is a many to many relationships because we have more than one employee being compared to many different customers. Even if in general for the database a one to many is more accurate as each salesperson can have multiple customers while being unique employees themselves.

1. **Add one customer record** with your last name using an INSERT statement. You may use the name of a celebrity or fictional character if you don’t use your own name.
   1. You may use the name of a celebrity or fictional character if you don’t use your own name. Think of this as your signature.
   2. Complete these actions to get to the right place to enter this information: (1) Show databases, (2) use classicmodels, (3) show tables, (4) describe customers;
      1. You should now see all of the fields that you’ll need to fill in to complete this step.
      2. Reference your Module Two lab or resources on how to populate these fields if you need to.
      3. Populate the following fields:

customerNumber

customerName

contactLastName

contactFirstName

phone

addressLine1

addressLine2

city

state

postalCode

country

salesRepEmployeeNumber

creditLimit

* + 1. Run a SELECT statement on the Customers table and capture it in a screenshot.

A black background with many small colored lines

Description automatically generated with medium confidence

1. **Reflection:** Use the lab environment or the screenshots you’ve worked with for this step. Address the prompts below in your reflection. Write a paragraph in response to each prompt:
   1. **Define how cardinality is applied** to the databases you’ve been working with. Explain why different numbers of records were returned from the different offices.

The cardinality on the database I was working is described in the document that we had pictured below.

A diagram of a company

Description automatically generated

We can see that there is one product line associated to zero or many products. And then the products are associated as one to zero or many order details. From there zero or many order details may be associated with one particular order from the orders table. Which can flow into as zero to many into a particular customer in the customers table. A customer in the customers table can have a record in the payments table that can be zero to one payment account. From the customers table there can be zero to many customers associated with zero to one employee that handles their sales. The employees themselves might have zero to many associated with another employee as a manager. And the employees have a zero to many relationship with one office.

* 1. **Compare and contrast** the different **queries** you ran and how cardinality applies to them.

Each query corresponds to the creation of a new table which might have the same cardinality of the table or tables the query came from might also have different. For instance in one query we looked at all customers in “MA” and the sales reps associated with them. In this case there were many sales reps associated with many different customers. We had another query where we selected the records associated with the customerNumber 103 which resulted in one to many in the form of one customerNumber being associated with many records.

* 1. **Describe two** of the crucial **benefits** **of cardinality** in this type of database.

The benefits of cardinality in this type of Database becomes clear when you realize just how much data could end up being stored here. I see this kind of diagram as being an overview of the logic of a database and how it is organized. With this information as a Data Scientist it makes it much easier to track what relationships exists between tables and thus gain the data you need from the different tables and how to retireve it. That way Data can be pulled into a CSV with and through these relationships saving time in certain aspects or getting a better value in the data samples that are pulled.