# DAD 220 Module Three Lab Template

## Overview

To complete this lab, go to your Codio virtual lab environment and start a new terminal session. Once there, **connect to the employee information you entered in the Module Two lab**. Then perform the steps below to complete the activity. Manually enter any commands you are asked to write.

At the end of each step in the activity, replace bracketed text with a screenshot, brief explanation, or both, as indicated. Size each screenshot and its explanation to fit about one-quarter of the page with the description written below the screenshot. Review the Template Screenshot Example linked in the guidelines and rubric for this assignment to see an example of how screenshots for your assignment should look.

## Create Joins Between Tables

1. **Update the name of the Branches table** that you created in the previous lab to say "Department".
   1. Use an ALTER statement to RENAME the Branches table "Department".
   2. Capture these outputs in a screenshot to validate that you successfully completed this step.

A screen shot of a computer program

Description automatically generated

1. **Insert fields to the Department table** so that joins can be performed on tables.
   1. INSERT INTO Department VALUES

(1, 'Accounting'),   
(2, 'Human Resources'),   
(3, 'Information Systems'),   
(4, 'Marketing');

* 1. Write a SELECT statement for this table to prove this step and validate that it ran correctly with a screenshot.

A screenshot of a computer program

Description automatically generated

1. **Perform joins between the Department and Employee tables** **and show results** for how many employees work in each of the four departments. This action will only provide information on the records that are already there.
   1. Department 1 = Accounting
      1. Command: SELECT First\_Name, Last\_Name, Department.Department\_Name FROM Employee INNER JOIN Department ON Employee.Department\_ID = Department.Department\_ID WHERE Employee.Department\_ID = 1;
   2. Using SELECT statements similar to that above, **perform joins to produce results** for the following tables:
      1. Department 2 = Human Resources
      2. Department 3 = Information Systems
      3. Department 4 = Marketing
   3. Capture the results of these joins and validate your work by providing a screenshot. You should have the same number of records as you do employees.

A computer screen shot of a black screen

Description automatically generated

A screen shot of a computer program

Description automatically generated

1. **Populate the Employee table with**information for 10 **new employees**.
   1. Give the employees unique names and include attributes for all necessary fields. Note: Reference attributes from the lab in Module Two. Department ID values must be between 1 and 4.

For this part I went ahead and used a few names I could think of along with some names from several video games I used. One thing that was super important was making sure to get the Employee ID right cause that needed to be unique to the employee. Personally to write this out I used note pad FIRST before copy pasting into codio that way I could check how things looked and make sure that commas and such were in the correct spots.

A screen shot of a computer program

Description automatically generated

1. **Perform a join across the Employee and Department tables** for each of the four departments. New and existing records should be displayed in the results.
   1. Take a screenshot to capture the updated results that the Employee and Department joins show and validate that they have run correctly. You should have the same number of records as you do employees.

A screenshot of a computer program

Description automatically generated

1. **Identify the resultant outputs** of the commands you wrote and answer the following question:
   1. How many records are returned for employees in each department?

For the departments there are 6 records for the Accounting Department, 3 for Human Resources, 3 for Information Systems, and 5 for Marketing.

1. **Create a CSV file** that contains only the records of employees in Human Resources and Information Systems. If you run this query multiple times, be sure to use a different file name each time. MySQL will not overwrite an existing file.
   1. Enter the command listed below.
      1. Command: select First\_Name, Last\_Name, Department.Department\_Name from Employee inner join Department on Employee.Department\_ID = Department.Department\_ID where Employee.Department\_ID = 3 OR Employee.Department\_ID = 2 into outfile'/home/codio/workspace/HRandIS-Employees.csv' FIELDS TERMINATED BY',' LINES TERMINATED BY '\r\n';
   2. Print the file output to the screen.
      1. In order to print your screen, start by refreshing your browser.
      2. Type the word "quit" after your MySQL prompt. Then press **Enter** to exit to the Linux shell. Do not exit the virtual lab environment.
      3. Print the output of your file to the screen using these steps:
         1. Type "pwd" and press **Enter**. Then type "ls" and press **Enter** again to list your files.
         2. Next, type "cat HRandIS-Employees.csv" and press **Enter**.
         3. Capture these outputs in a screenshot to validate that you successfully completed this step.

A screenshot of a computer

Description automatically generated

1. **Reflection:** Provide detailed insight on the prompts below. Explain your process and how and why your process worked. Write your responses to the questions below in paragraph form.

* 1. Process
     1. **Explain** how **the joins** you used in this assignment worked.

The joins for this assignment worked by looking at each rows Department ID in the Employee Table and then matching it to the ID in Department Table. If match was found. It would create a new joined table with the First Name, Last Name and Department Name were the attributes(columns) selected from the tables joining together.

* + 1. **Describe** why the **commands** you used were able to retrieve the Department table when you selected the Department name.

The commands I used were able to retrieve the Department table due to when you use join it is calling up the second table for comparison using the join type and any where statements as directions on what to look for matches. Jome join types selecting all attributes(columns) of a table regardless of what matches there are besides what is used as a limiter in the where statement.

* 1. File creation and extraction
     1. **Identify** how many **records** are in the file when you write the records of your query to a CSV file.

6 records for the CSV file.

* + 1. **Explain**, in detail, the process of **extracting data** to a flat file.

Databases while set up in a table format actually can not be straight red by other programs without the the capability built into it to do so. For this case let’s use Excl which needs the csv file..or make it easier to access for R or Python. It requires the data to be transformed from its database form to a flat file form in this case being csv which looks at the table and goes these are the attrtubutes so we will put them in commas. Then the next line matching each value to its attribute using commas and so on to create the comma separated values needed to read in other programs.