# DAD 220 Module Three Lab Template

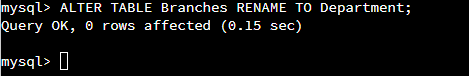
## Overview

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

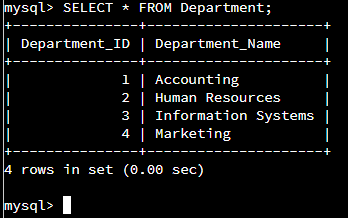
1. **Update the name of the Branches table** that you created in the previous lab to say "Department".
   1. Use an ALTER statement to RENAME the Branches table "Department".
   2. Capture these outputs in a screenshot to validate that you successfully completed this step.



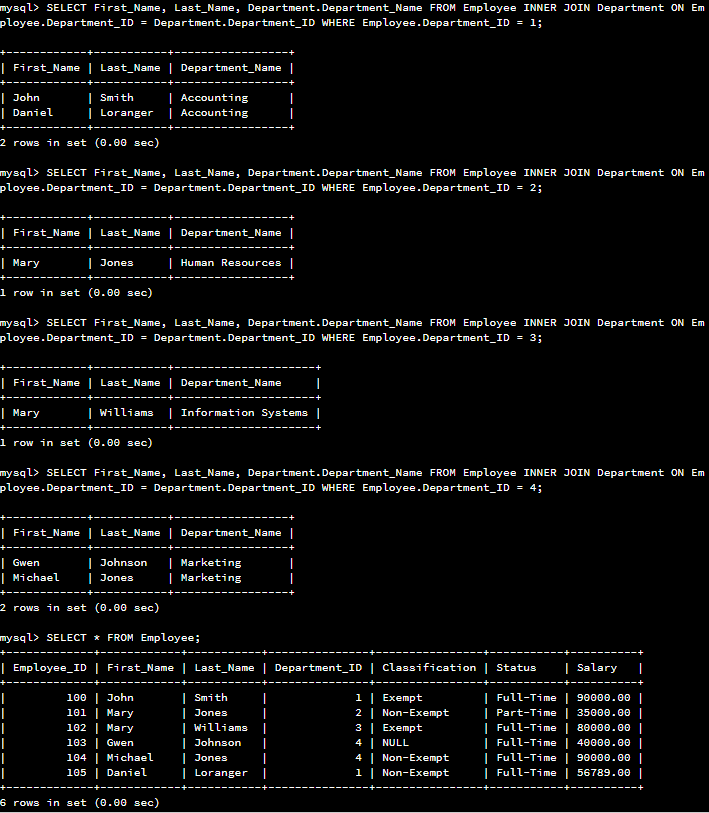
1. **Insert fields to the Department table** so that joins can be performed on tables.
   1. INSERT INTO Department VALUES

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

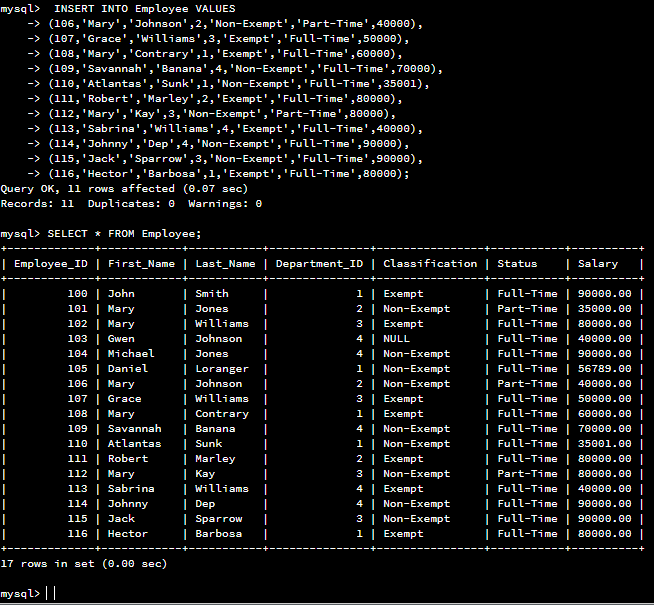
* 1. Write a SELECT statement for this table to prove this step and validate that it ran correctly with a screenshot.



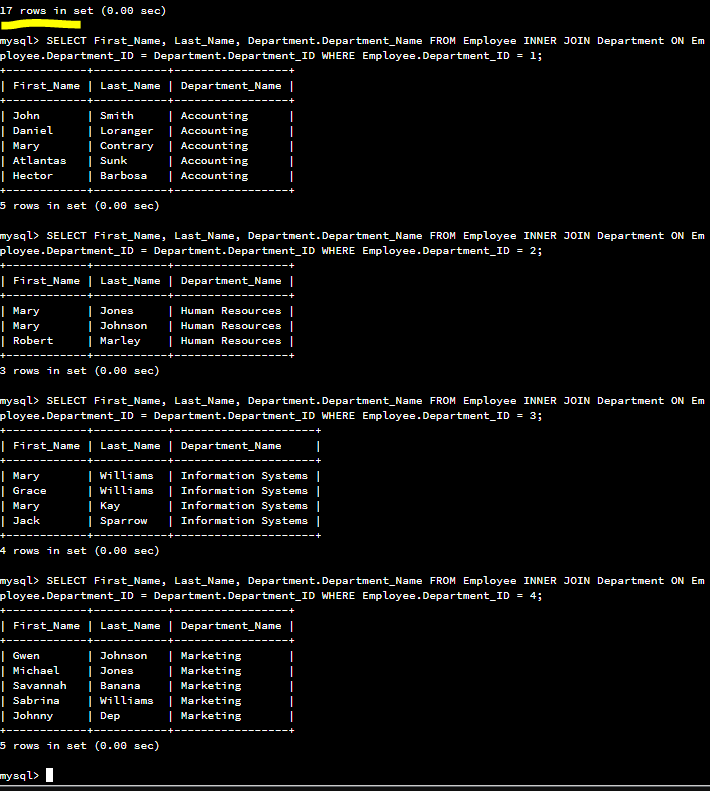
1. **Perform joins between the Department and Employee tables** **and show results** for how many employees work in each of the four departments. This action will only provide information on the records that are already there.
   1. Department 1 = Accounting
      1. Command: SELECT First\_Name, Last\_Name, 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.



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.



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.



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?

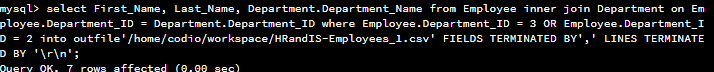
Department 1: 5

Department 2: 3

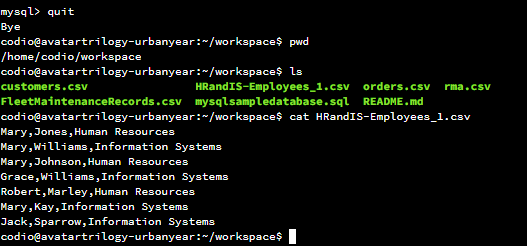
Department 3: 4

Department 4: 5

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';



* 1. 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.



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.

“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;”

The Join first gathers the appropriate rows based on ‘WHERE Employee.Department\_ID = 1’.

Next the columns ‘First\_Name’, ‘Last\_Name’ are gathered along with the ‘Employee.Department\_ID’ column, but having joined the tables based on the ‘Department\_ID’ columns, the querry can then retrieve the appropriate text field for the rows department name respectively.

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

The join commands cause the tables to merge in predictable ways, while still maintaining the column names from original table names for selectivity. The columns where uniquely named are self identified and do not require the table name prefix, but where duplicate column names are available, the original table needs to be called out as a prefix value to make the selection non-ambiguous.

* 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.

As can be visualized in the previous step evidence, the query for HR and IS yielded 7 records from the available 17 records in all departments.

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

Having completed the join and select sequence as described in subsection A above, the ‘into outfile <path>’ tells the processor to itemize each record into respective fields with a delimitator value of a comma “,” declared with the processor directive “FIELDS TERMINATED BY’,’ “ and further directive for how to handle each row ending by including the windows line ending formatting of “ ’\r\n’ ”.

Having all these directives, the processor will select the first record, break it into fields of data (presumably in an array or list), then print to file each fields contents followed by the comma separator. On the last entry in the list, rather than using a separator it uses the line terminator.

This process then repeats for each record.