DAD 220 Intro to Struct Database Env

Module Three Lab

Professor Aastha Agarwal

Matt Bramer

September 10, 2020

1. Before you begin, type the following commands prior to typing MySQL to set file permissions. This will allow you to perform the file output creation:
   1. **chmod +x change\_perm.sh** and then
   2. **./change\_perm.sh**

A picture containing drawing

Description automatically generated

* 1. Then, enter a command line session with MySQL and reconnect to the employee information you entered in the previous lab.
  2. Write a SELECT statement for the Employee table to check that you have reconnected to the right information.

A close up of text on a black background

Description automatically generated

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

A close up of text on a black background

Description automatically generated

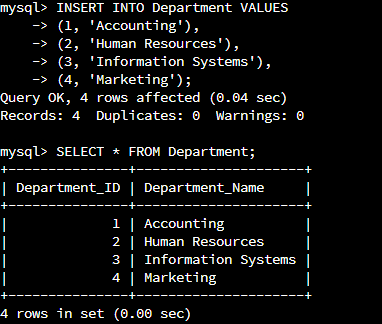
1. **Insert fields to the Department table** so that you will be able to perform joins on them.
   1. INSERT INTO Department VALUES  
      (1, ‘Accounting’),  
      (2, ‘Human Resources’),  
      (3, ‘Information Systems’),  
      (4, ‘Marketing’);
   2. Write a SELECT statement for this table to prove this step and validate that it ran correctly with a screenshot.

A screenshot of a cell phone

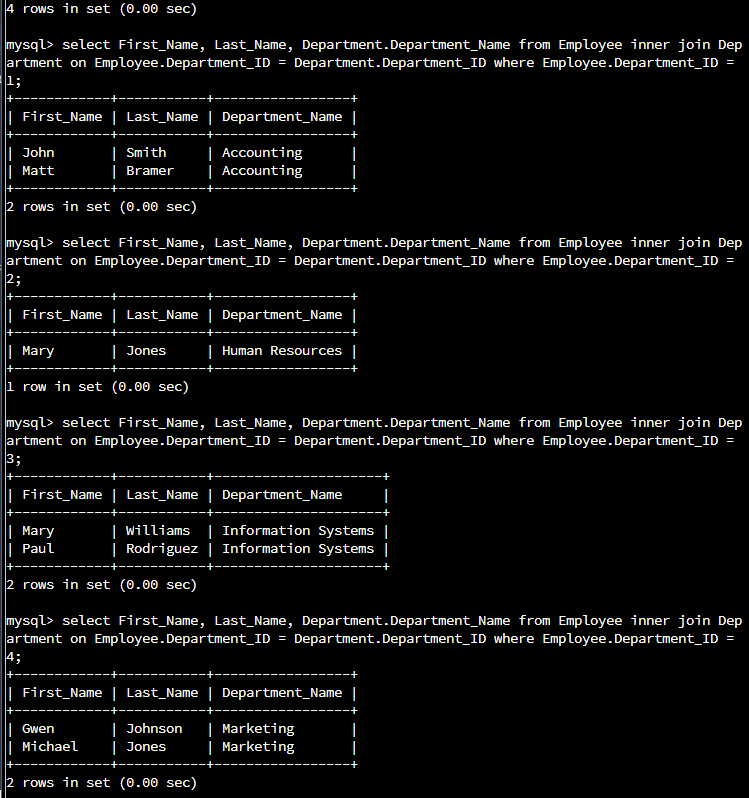
Description automatically generated

*It was here I realized I was using “Demartment\_Name” instead of DePartment\_Name. Now to find out how to change the name of a column name.*

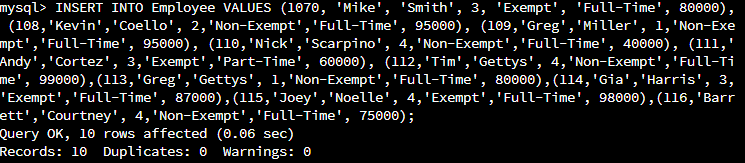
*Easiest solution: DROP TABLE Department; CREATE TABLE Department ( …*

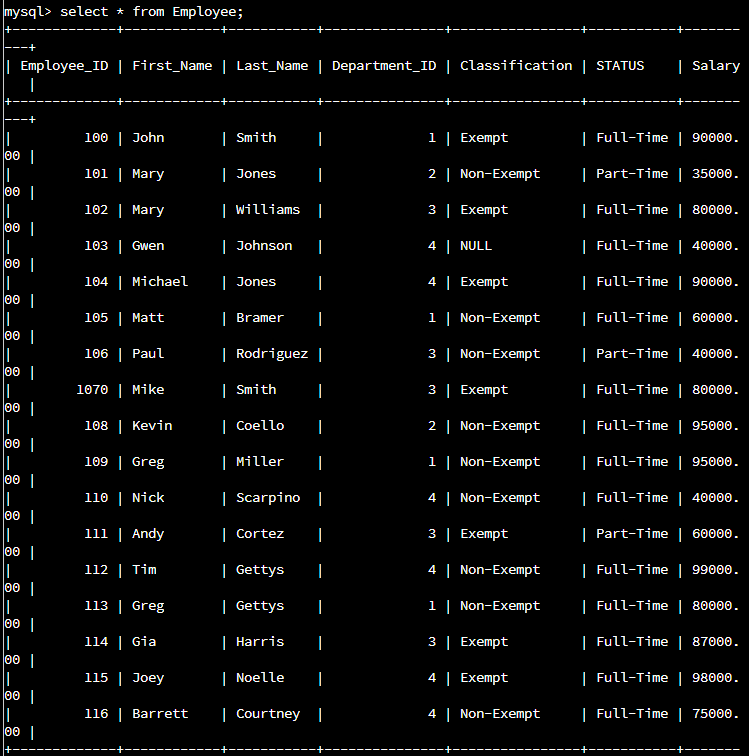
 UPDATE w/ “Department\_NAME”

1. Now, **perform joins between the Department and Employee tables and show results** for how many employees work in each one of the four departments. This 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 the one above, **perform joins to produce results** for the following tables:
      1. Department 2 = Human Resources
      2. Department 3 = Information Services
      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.

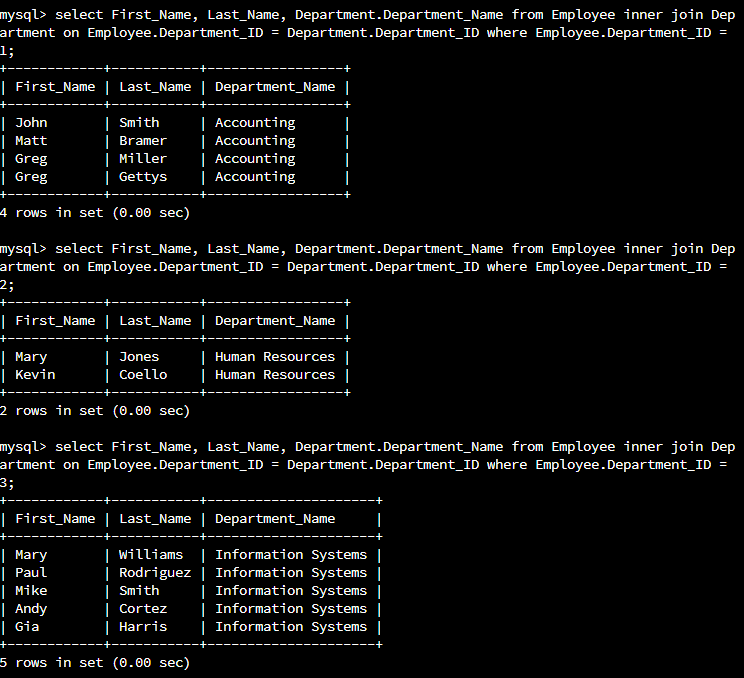


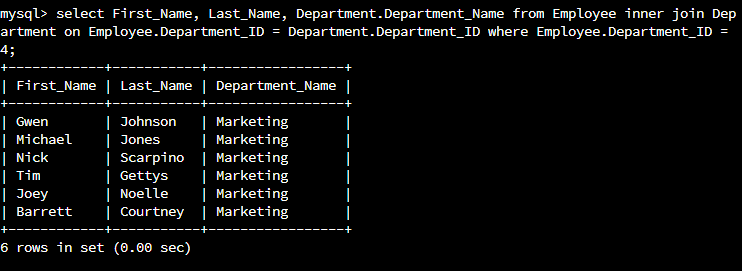
1. **Populate the Employee table with** information for ten **new employees**. Give them unique names and include attributes for *all* necessary fields. (Note: Please reference attributes from the lab in Module Two. Department ID values must be between 1 and 4.)





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.
2. Take a screenshot to capture the updated results that the Employee and Department joins show to validate that they have run correctly. You should have the same number of records as you do employees.

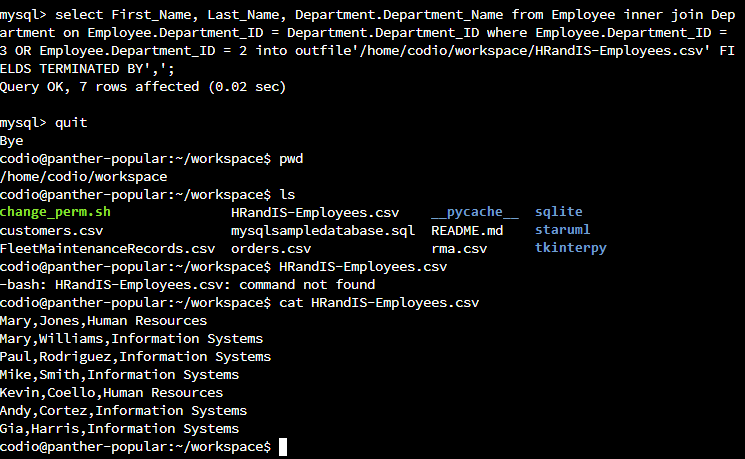




1. **Identify the resultant outputs** of the commands that you have written:
   1. How many records are returned for employees in each department?

**Three records (First\_Name, Last\_Name, Department\_Name)**

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.
2. 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',';
3. Print the file output to the screen.
   1. You’ll need to type the word quit after your MySQL prompt and then press Enter to exit to the Linux shell. Do not exit the virtual lab environment itself.
   2. Next, print the output of your file to the screen by following these steps:
      * 1. Type **pwd** and press **Enter**, then type **ls** and press **Enter** again. This will list your files.
        2. Now, type **cat HRandIS-Employees.csv** and press **Enter**.
        3. Capture these outputs in a screenshot to validate that you have successfully completed this step.



1. Reflections: **Provide detailed insight on the prompts below** by explaining your process along with how and why it ultimately worked.
2. **Process**
   1. **Explain how the joins** you used in this assignment worked.

Joins were used with INNER JOIN and they connected First\_Name, Last\_Name, and Department.Department\_Name from Employee to Department.

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

Because the INNER JOIN connected Employee to Department, between the two tables.

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

Only 7 records, because there were only 5 Employees in Information Systems, and 2 Employees in Human Resources.

* 1. **Provide a detailed explanation** of how the process of extracting data to a flat file works.

Extracting data is basically copying data from one database/table to another.