Joshua Merren

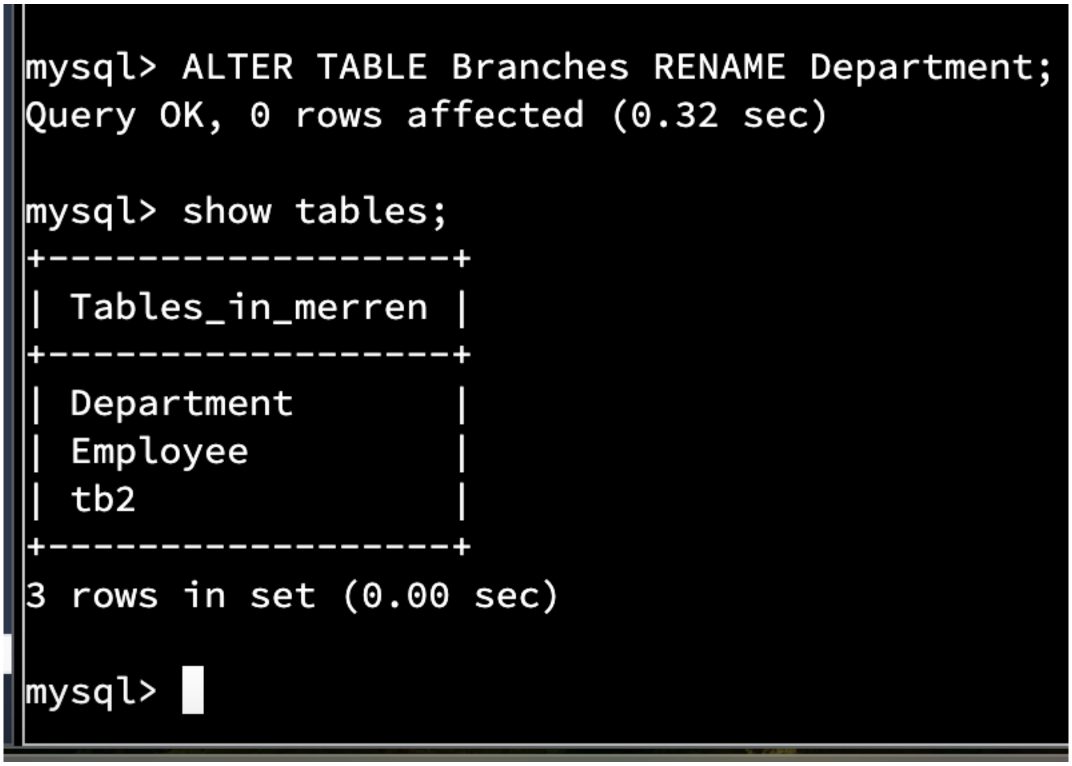
Professor Lockwood

DAD-220 Intro to Struct Database Env

24 January 2024

3-2 Lab: Table Joins

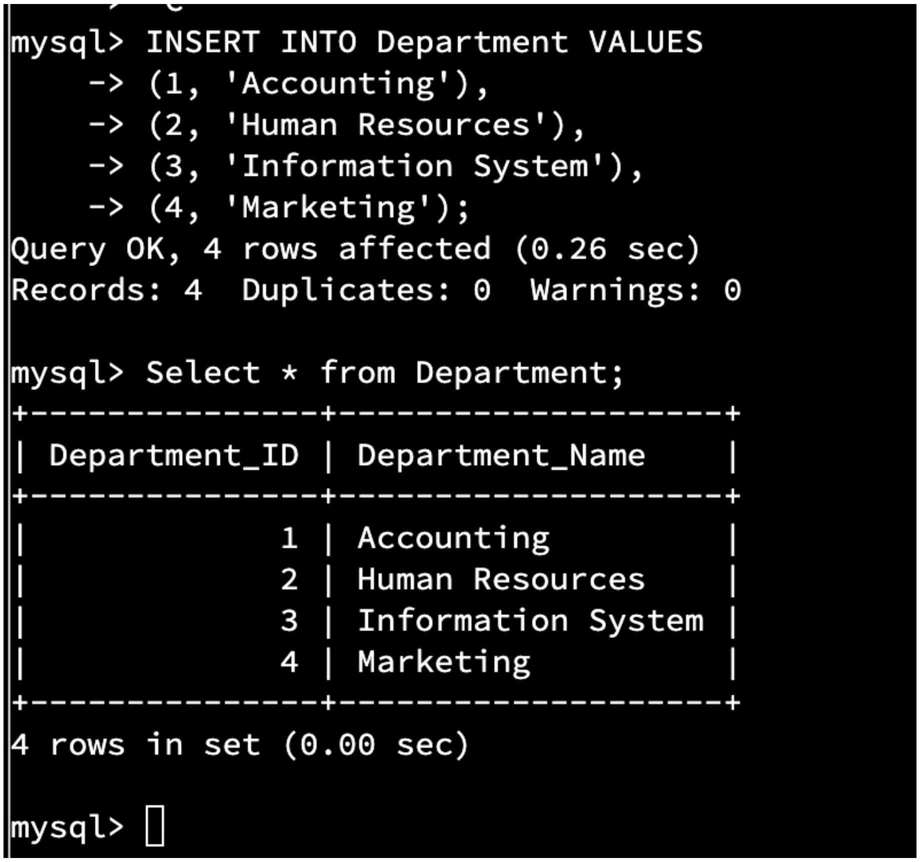
1. **Update the name of the Branches table** that you created in the previous lab to say "Department".
   a. Use an ALTER statement to successfully RENAME the "Branches" table to "Department".
   b. Capture these outputs in a screenshot to validate that you've successfully completed this step.



```
mysql> ALTER TABLE Branches RENAME Department;
Query OK, 0 rows affected (0.32 sec)

mysql> show tables;
+------------------+
| Tables_in_merren |
+------------------+
| Department       |
| Employee         |
| tb2              |
+------------------+
3 rows in set (0.00 sec)

mysql> 
```

This screenshot is the command “ALTER TABLE Branches RENAME Department” to Alter Table branches and RENAME the table to Department.

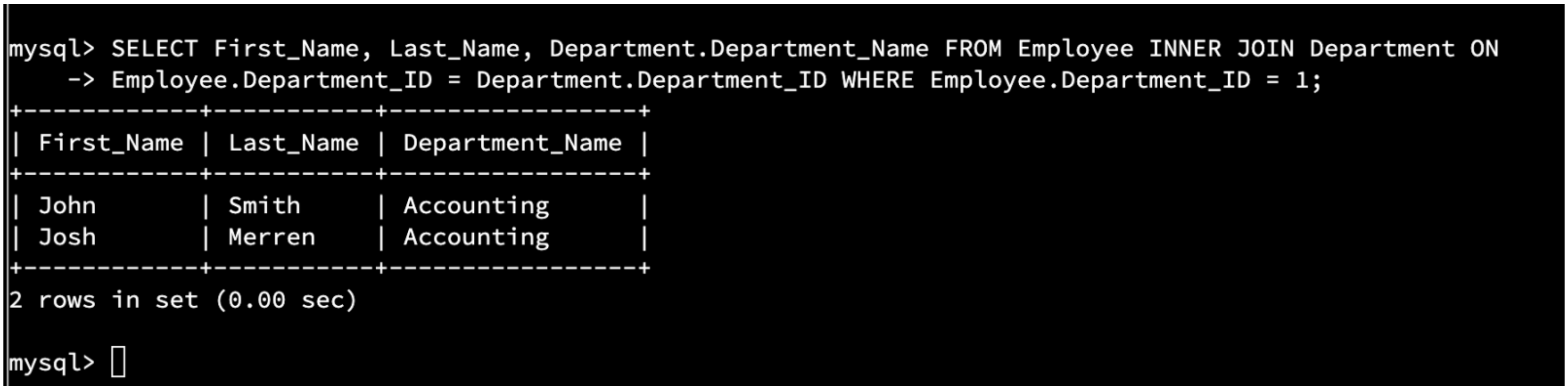
2. **Insert fields to the Department table** so that you’ll be able to perform joins on them.
    a. INSERT INTO Department VALUES
        - (1, 'Accounting'),
        - (2, 'Human Resources'),
        - (3, 'Information Systems'),
        - (4, 'Marketing');
    b. Write a SELECT statement for this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



```
mysql> INSERT INTO Department VALUES
    -> (1, 'Accounting'),
    -> (2, 'Human Resources'),
    -> (3, 'Information System'),
    -> (4, 'Marketing');
Query OK, 4 rows affected (0.26 sec)
Records: 4  Duplicates: 0  Warnings: 0

mysql> Select * from Department;
+---------------+--------------------+
| Department_ID | Department_Name    |
+---------------+--------------------+
|             1 | Accounting         |
|             2 | Human Resources    |
|             3 | Information System |
|             4 | Marketing          |
+---------------+--------------------+
4 rows in set (0.00 sec)

mysql> 
```

This is a screenshot of the commands “INSERT INTO Department VALUES” which inserted four department ID’s and four departments into the Department table. The next command is “Select * from Department” which this command selects everything in the Department table.

3. 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.
    a. Department 1 = Accounting
        i. 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;

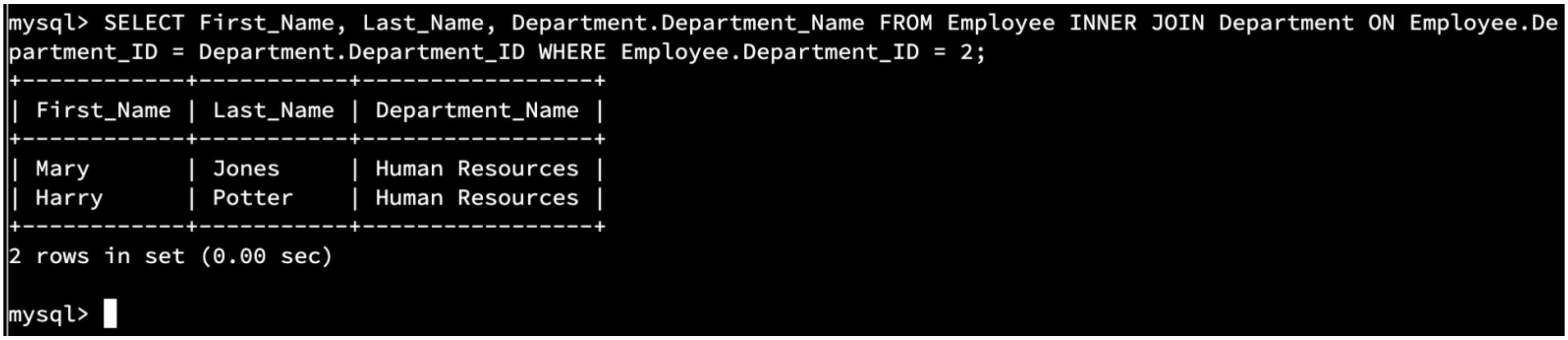


```
mysql> SELECT First_Name, Last_Name, Department.Department_Name FROM Employee INNER JOIN Department ON
    -> Employee.Department_ID = Department.Department_ID WHERE Employee.Department_ID = 1;
+------------+-----------+-----------------+
| First_Name | Last_Name | Department_Name |
+------------+-----------+-----------------+
| John       | Smith     | Accounting      |
| Josh       | Merren    | Accounting      |
+------------+-----------+-----------------+
2 rows in set (0.00 sec)

mysql> 
```

This is a screenshot of the following 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;” This command retrieves the first and last names of employees along with the name of their department, for all employees who are in the department with an ID of 1.

b. Using SELECT statements similar to the one above, **perform joins to produce results** for the following tables:

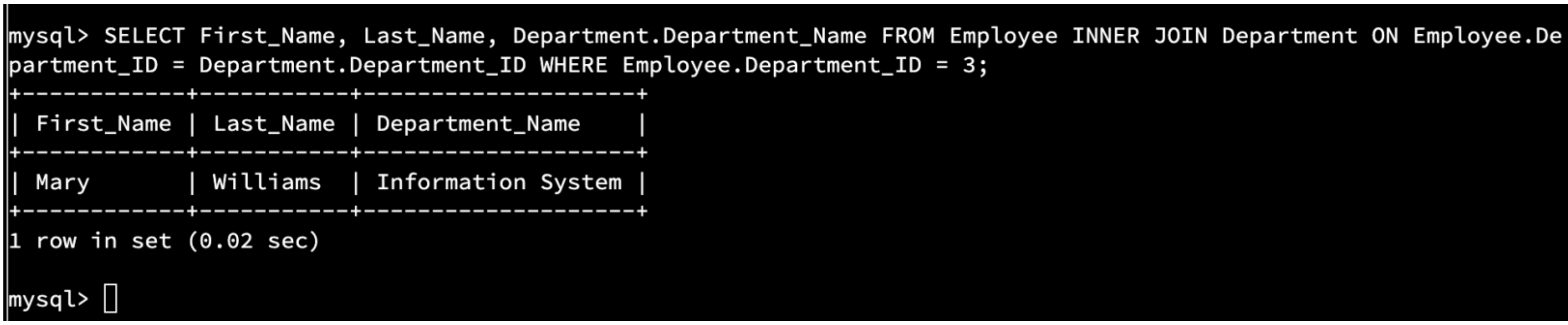
   i. Department 2 = Human Resources



```
mysql> SELECT First_Name, Last_Name, Department.Department_Name FROM Employee INNER JOIN Department ON Employee.De
partment_ID = Department.Department_ID WHERE Employee.Department_ID = 2;
+------------+-----------+-----------------+
| First_Name | Last_Name | Department_Name |
+------------+-----------+-----------------+
| Mary       | Jones     | Human Resources |
| Harry      | Potter    | Human Resources |
+------------+-----------+-----------------+
2 rows in set (0.00 sec)

mysql> 
```

This is a screen shot of the same command before but is for all employees of the department ID of 2.

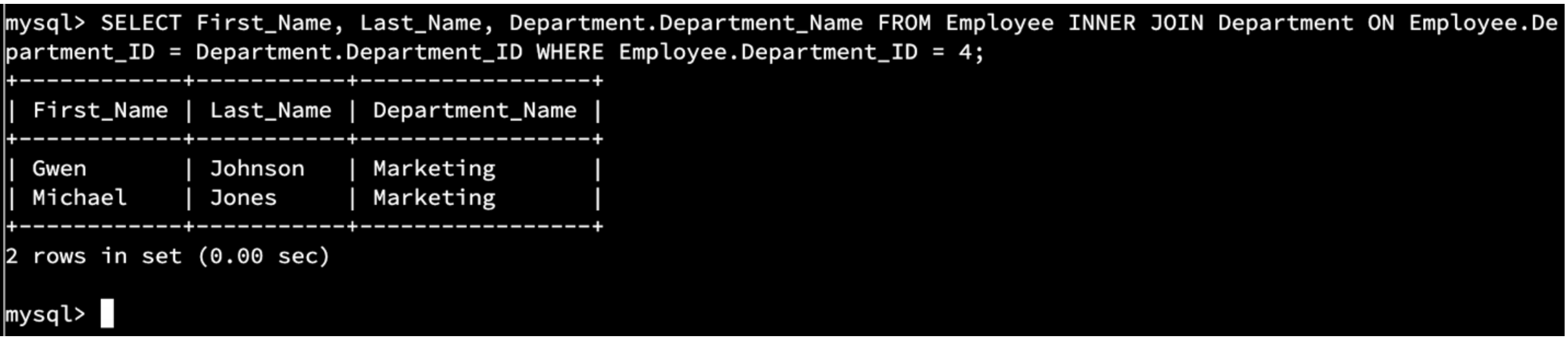
   ii. Department 3 = Information Systems



```
mysql> SELECT First_Name, Last_Name, Department.Department_Name FROM Employee INNER JOIN Department ON Employee.De
partment_ID = Department.Department_ID WHERE Employee.Department_ID = 3;
+------------+-----------+--------------------+
| First_Name | Last_Name | Department_Name    |
+------------+-----------+--------------------+
| Mary       | Williams  | Information System |
+------------+-----------+--------------------+
1 row in set (0.02 sec)

mysql> 
```

This is a screen shot of the same command before but is for all employees of the department ID of 3.

   iii. Department 4 = Marketing



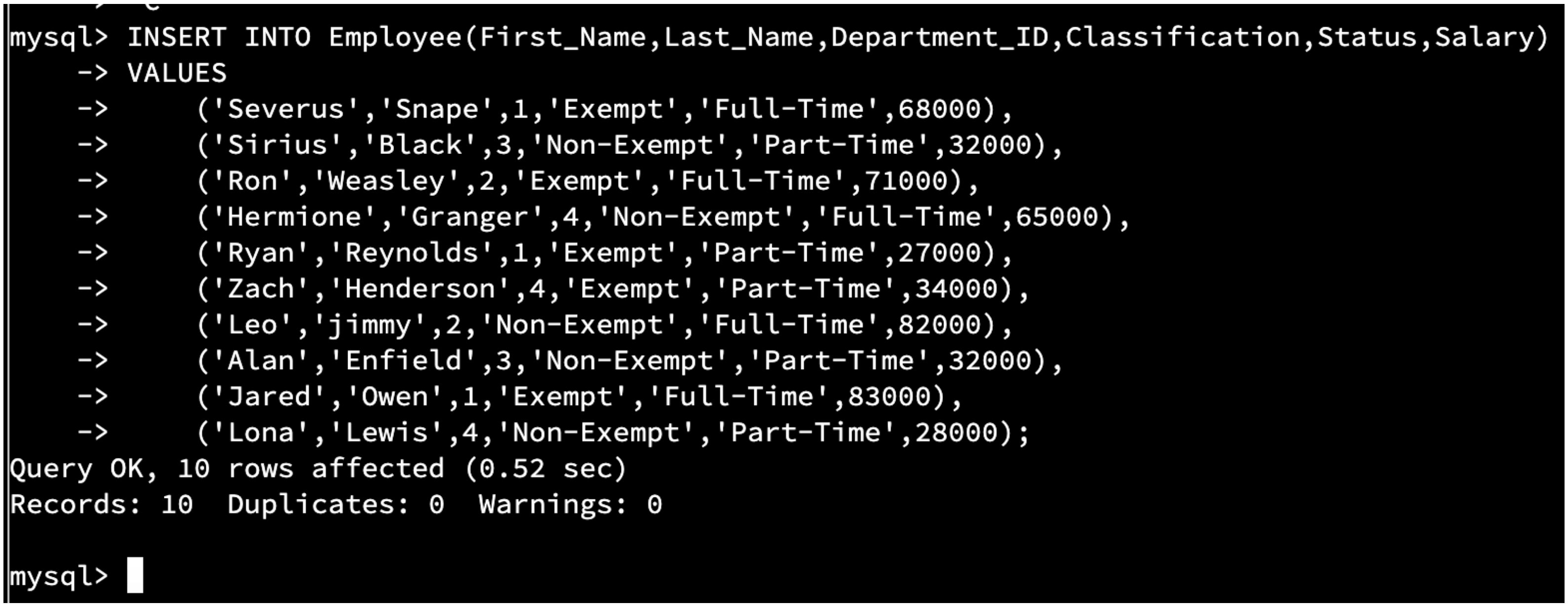
```
mysql> SELECT First_Name, Last_Name, Department.Department_Name FROM Employee INNER JOIN Department ON Employee.De
partment_ID = Department.Department_ID WHERE Employee.Department_ID = 4;
+------------+-----------+-----------------+
| First_Name | Last_Name | Department_Name |
+------------+-----------+-----------------+
| Gwen       | Johnson   | Marketing       |
| Michael    | Jones     | Marketing       |
+------------+-----------+-----------------+
2 rows in set (0.00 sec)

mysql> 
```

This is a screen shot of the same command before but is for all employees of the department ID of 4.

c. 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.

4. **Populate the Employee table with** information for ten **new employees.**

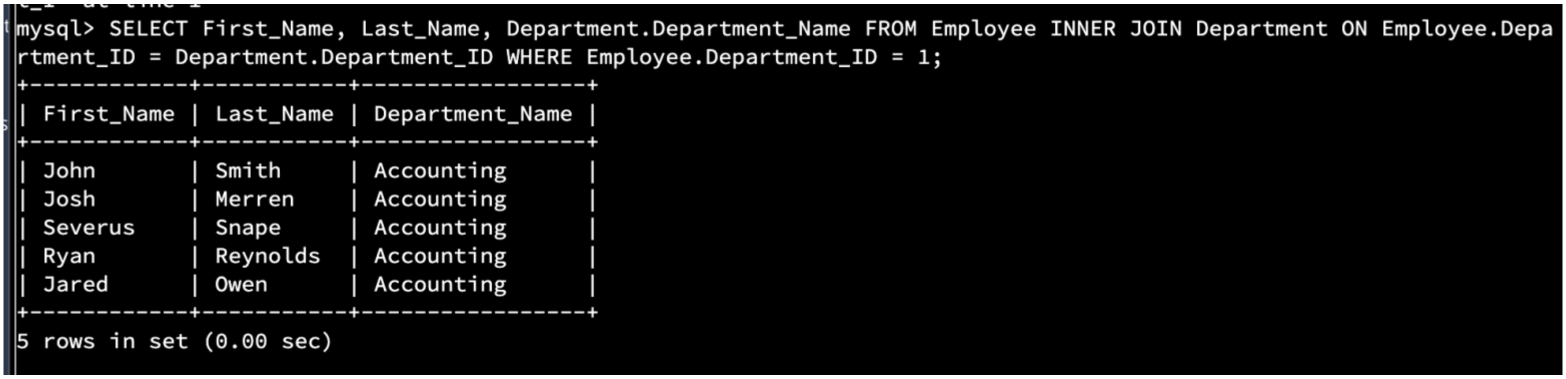
a. Give them unique names and include attributes for **all** necessary fields. (Note: Please reference attributes from the lab in Module Two. Department ID values must be between 1 and 4.)



```
mysql> INSERT INTO Employee(First_Name,Last_Name,Department_ID,Classification,Status,Salary)
    -> VALUES
    ->     ('Severus','Snape',1,'Exempt','Full-Time',68000),
    ->     ('Sirius','Black',3,'Non-Exempt','Part-Time',32000),
    ->     ('Ron','Weasley',2,'Exempt','Full-Time',71000),
    ->     ('Hermione','Granger',4,'Non-Exempt','Full-Time',65000),
    ->     ('Ryan','Reynolds',1,'Exempt','Part-Time',27000),
    ->     ('Zach','Henderson',4,'Exempt','Part-Time',34000),
    ->     ('Leo','jimmy',2,'Non-Exempt','Full-Time',82000),
    ->     ('Alan','Enfield',3,'Non-Exempt','Part-Time',32000),
    ->     ('Jared','Owen',1,'Exempt','Full-Time',83000),
    ->     ('Lona','Lewis',4,'Non-Exempt','Part-Time',28000);
Query OK, 10 rows affected (0.52 sec)
Records: 10  Duplicates: 0  Warnings: 0

mysql> 
```

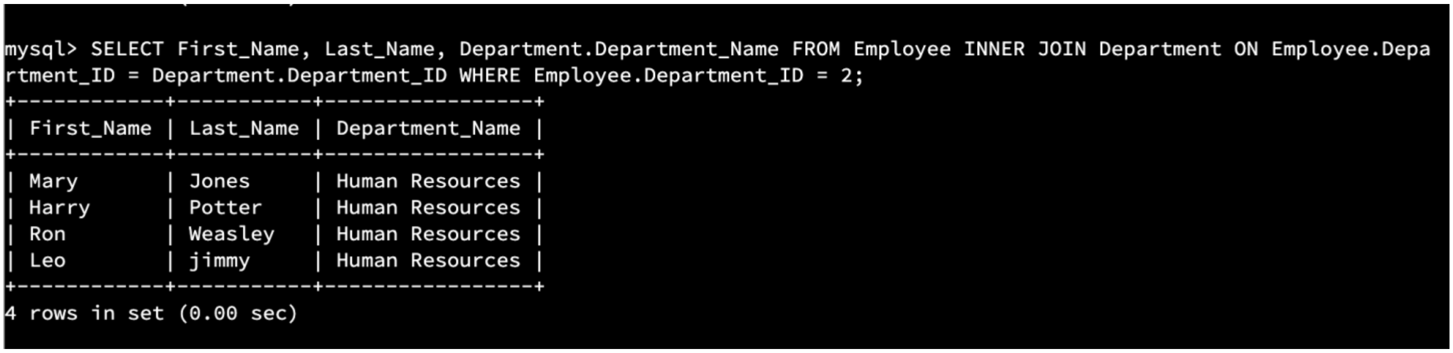
This is a screenshot of my command “INSERT INTO Employee (First_Name,Department_ID,Classification,Status,Salary) and the values to be entered into the Employee table.

5. **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.
    a. 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.



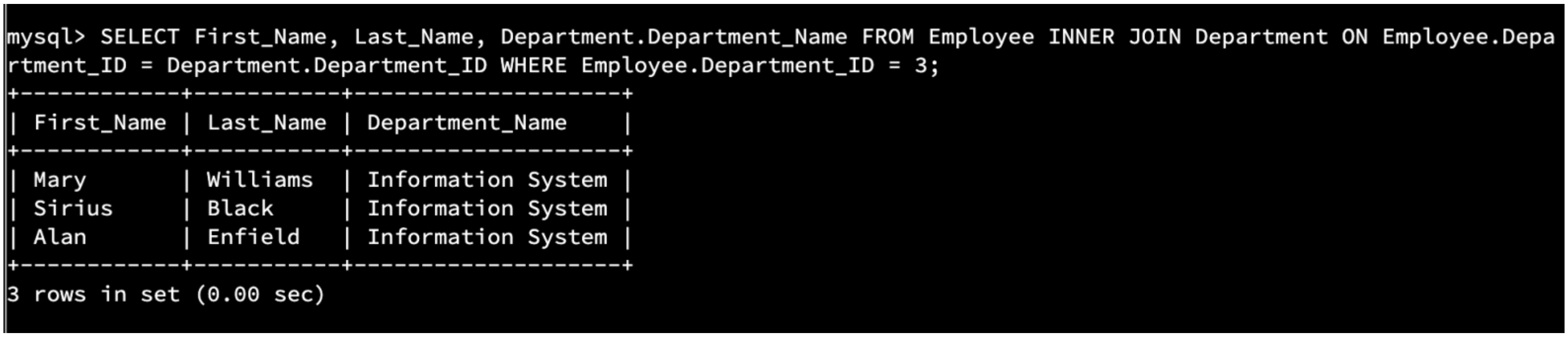
```
mysql> SELECT First_Name, Last_Name, Department.Department_Name FROM Employee INNER JOIN Department ON Employee.Depa
rtment_ID = Department.Department_ID WHERE Employee.Department_ID = 1;
+------------+-----------+-----------------+
| First_Name | Last_Name | Department_Name |
+------------+-----------+-----------------+
| John       | Smith     | Accounting      |
| Josh       | Merren    | Accounting      |
| Severus    | Snape     | Accounting      |
| Ryan       | Reynolds  | Accounting      |
| Jared      | Owen      | Accounting      |
+------------+-----------+-----------------+
5 rows in set (0.00 sec)
```

This screenshot is the 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;. This command retrieves the first and last names of employees along with the name of their department, for all employees who are in the department with an ID of 1.



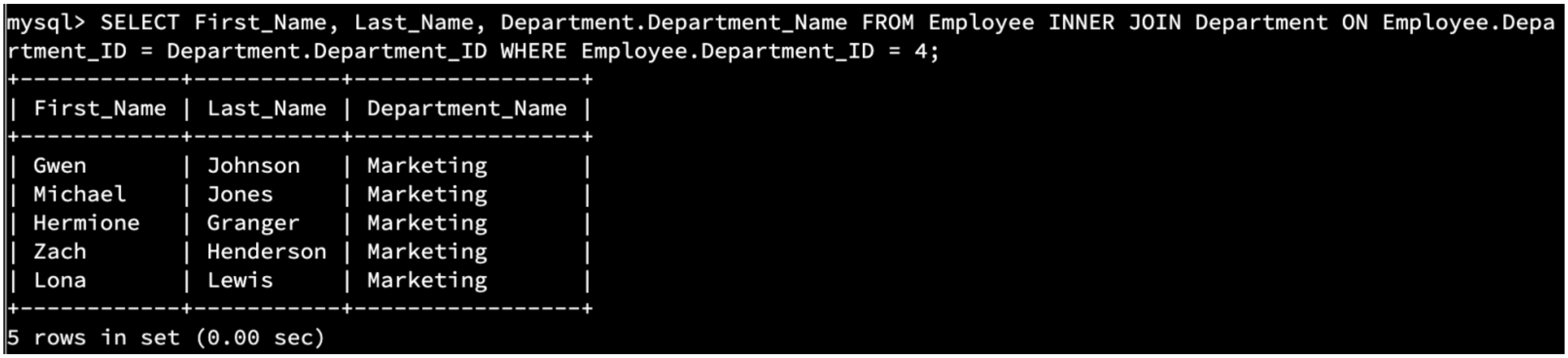
```
mysql> SELECT First_Name, Last_Name, Department.Department_Name FROM Employee INNER JOIN Department ON Employee.Depa
rtment_ID = Department.Department_ID WHERE Employee.Department_ID = 2;
+------------+-----------+-----------------+
| First_Name | Last_Name | Department_Name |
+------------+-----------+-----------------+
| Mary       | Jones     | Human Resources |
| Harry      | Potter    | Human Resources |
| Ron        | Weasley   | Human Resources |
| Leo        | jimmy     | Human Resources |
+------------+-----------+-----------------+
4 rows in set (0.00 sec)
```

This is a screen shot of the same command as before but is for all employees of the department ID of 2.



```
mysql> SELECT First_Name, Last_Name, Department.Department_Name FROM Employee INNER JOIN Department ON Employee.Depa
rtment_ID = Department.Department_ID WHERE Employee.Department_ID = 3;
+------------+-----------+--------------------+
| First_Name | Last_Name | Department_Name    |
+------------+-----------+--------------------+
| Mary       | Williams  | Information System |
| Sirius     | Black     | Information System |
| Alan       | Enfield   | Information System |
+------------+-----------+--------------------+
3 rows in set (0.00 sec)
```

This is a screen shot of the same command as before but is for all employees of the department ID of 3.



```
mysql> SELECT First_Name, Last_Name, Department.Department_Name FROM Employee INNER JOIN Department ON Employee.Depa
rtment_ID = Department.Department_ID WHERE Employee.Department_ID = 4;
+------------+-----------+-----------------+
| First_Name | Last_Name | Department_Name |
+------------+-----------+-----------------+
| Gwen       | Johnson   | Marketing       |
| Michael    | Jones     | Marketing       |
| Hermione   | Granger   | Marketing       |
| Zach       | Henderson | Marketing       |
| Lona       | Lewis     | Marketing       |
+------------+-----------+-----------------+
5 rows in set (0.00 sec)
```

This is a screen shot of the same command as before but is for all employees of the department ID of 4.

6. **Identify the resultant outputs** of the commands that you've written:

    a. How many records are returned for employees in each department?

i. Accounting – 5

ii. Human Resources – 4

iii. Information Systems – 3

iv. Marketing - 5

7. **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.

    a. Enter the command listed below.

        i. 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';

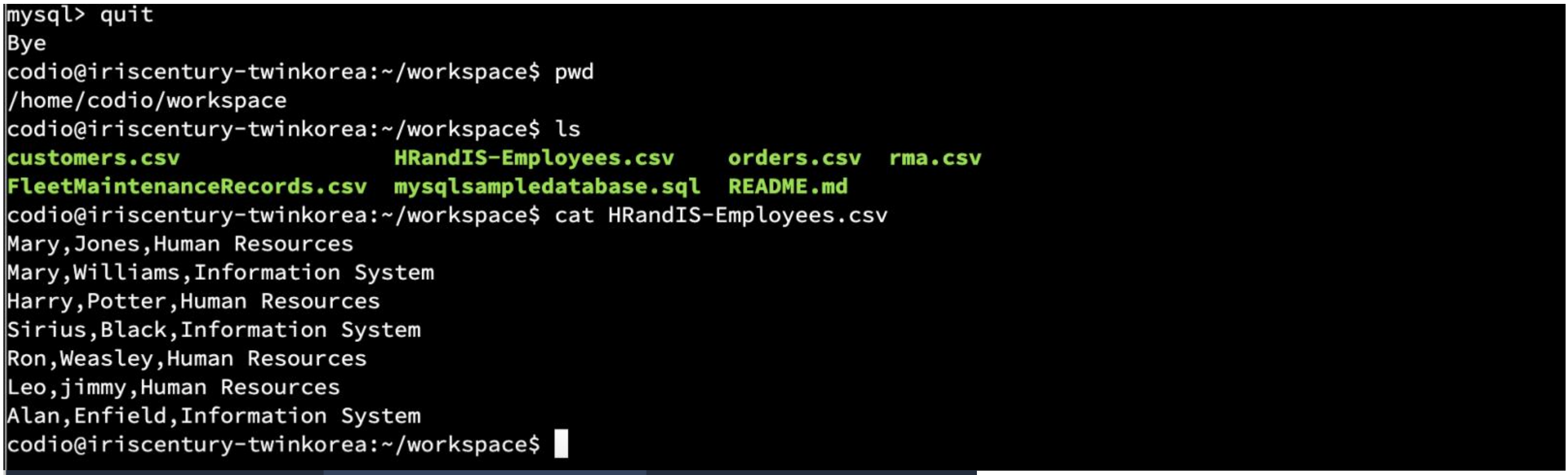
    b. Print the file output to the screen.

        i. In order to print your screen, start by refreshing your browser.

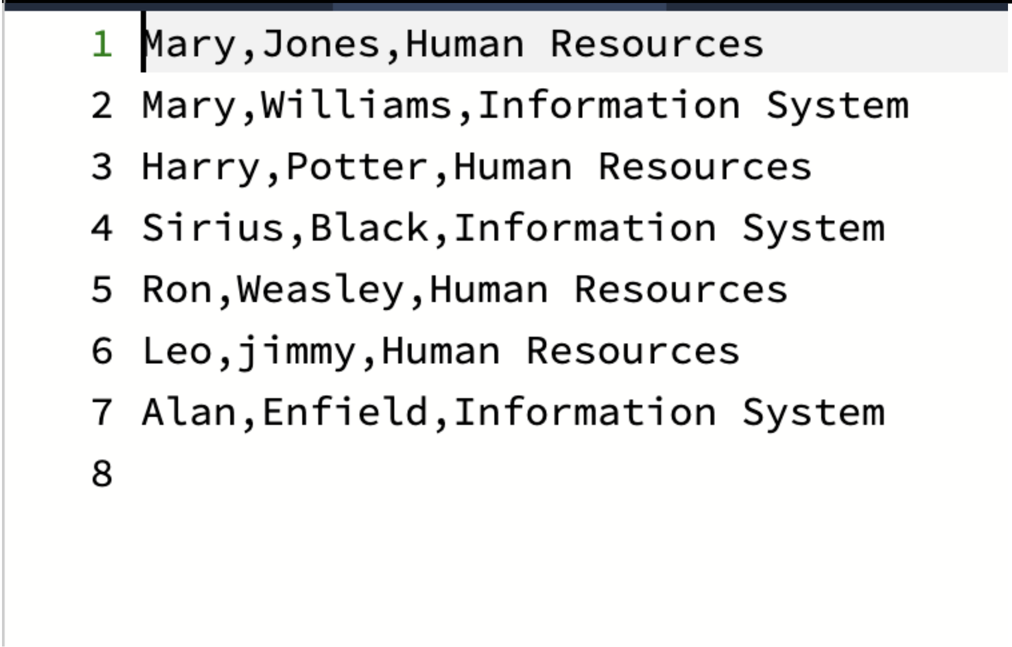
        ii. You'll need to type the word **quit** after your MySQL prompt and then press **Enter** to exit to the Linux shell. Do not exit the virtual lab environment itself.

        iii. 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've successfully completed this step.



```
mysql> quit
Bye
codio@iriscentury-twinkorea:~/workspace$ pwd
/home/codio/workspace
codio@iriscentury-twinkorea:~/workspace$ ls
customers.csv                HRandIS-Employees.csv    orders.csv  rma.csv
FleetMaintenanceRecords.csv  mysqlsampledatabase.sql  README.md
codio@iriscentury-twinkorea:~/workspace$ cat HRandIS-Employees.csv
Mary,Jones,Human Resources
Mary,Williams,Information System
Harry,Potter,Human Resources
Sirius,Black,Information System
Ron,Weasley,Human Resources
Leo,jimmy,Human Resources
Alan,Enfield,Information System
codio@iriscentury-twinkorea:~/workspace$
```



```
Mary,Jones,Human Resources
Mary,Williams,Information System
Harry,Potter,Human Resources
Sirius,Black,Information System
Ron,Weasley,Human Resources
Leo,jimmy,Human Resources
Alan,Enfield,Information System

```

These are screenshots of a CSV file that only contains records from the Employees in Human Resources and Information Systems departments.

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

The joins in this assignment were key for combining data from two separate tables, namely the Department and Employee tables. By using joins, particularly INNER JOIN, I could merge these tables based on a common field (Department_ID), allowing me to display employees within their respective departments. This was essential for organizing and presenting the data coherently and department-wise segregated.

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

The commands I used efficiently retrieved data from the Department table when selecting by the Department name. This was possible because the Employee and Department tables contained a common field, Department_ID. Utilizing the INNER JOIN operation on this shared field allowed me to combine relevant data from both tables, ensuring that the output displayed the department names and employee details.

b. File creation and extraction

i. **Identify** how many **records** are in the file when you write the records of your query to a CSV file.

The CSV file created contained seven records. This number corresponds to the total count of employees in the specified departments (Human Resources and Information Systems) that were extracted.

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

The process of exporting data to a flat file (CSV format) began with a SELECT statement to choose the desired fields from the Employee table, such as first name, last name, and department details.

- An INNER JOIN clause was then used to merge this data with the Department table based on the shared Department_ID field.

- The final step was to direct this combined data into a CSV file using the OUTFILE command. This command specified the file path, the data separation format (commas for fields), and line termination.

- The outcome was a well-structured CSV file named 'HRandIS-Employees.csv,' which made the data more accessible and easier for further analysis and readability for anyone who cannot understand what they are looking at in MySQL.