**Hands-on Lab: String Patterns, Sorting and Grouping in MySQL using phpMyAdmin**

**Estimated time needed:** 20 minutes

In this lab, you will learn how to create tables and load data in the MySQL database service using the phpMyAdmin graphical user interface (GUI) tool.

**Software Used in this Lab**

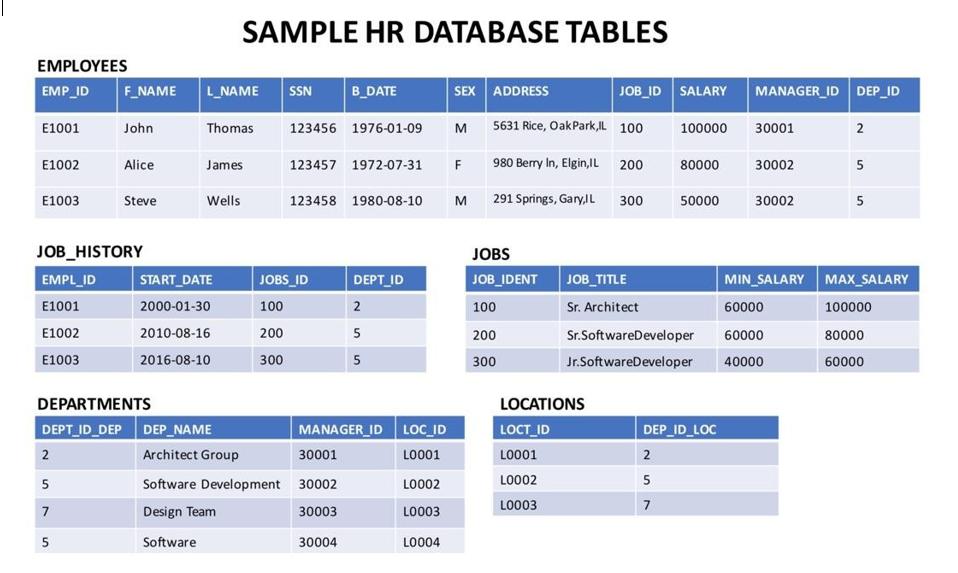
In this lab, you will use [MySQL](https://www.mysql.com/?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDB0110ENSkillsNetwork24601058-2021-01-01). MySQL is a Relational Database Management System (RDBMS) designed to efficiently store, manipulate, and retrieve data.



To complete this lab you will utilize MySQL relational database service available as part of IBM Skills Network Labs (SN Labs) Cloud IDE. SN Labs is a virtual lab environment used in this course.

**Database Used in this Lab**

The database used in this lab is an internal database. You will be working on a sample HR database. This HR database schema consists of 5 tables called **EMPLOYEES**, **JOB\_HISTORY**, **JOBS**, **DEPARTMENTS** and **LOCATIONS**. Each table has a few rows of sample data. The following diagram shows the tables for the HR database:

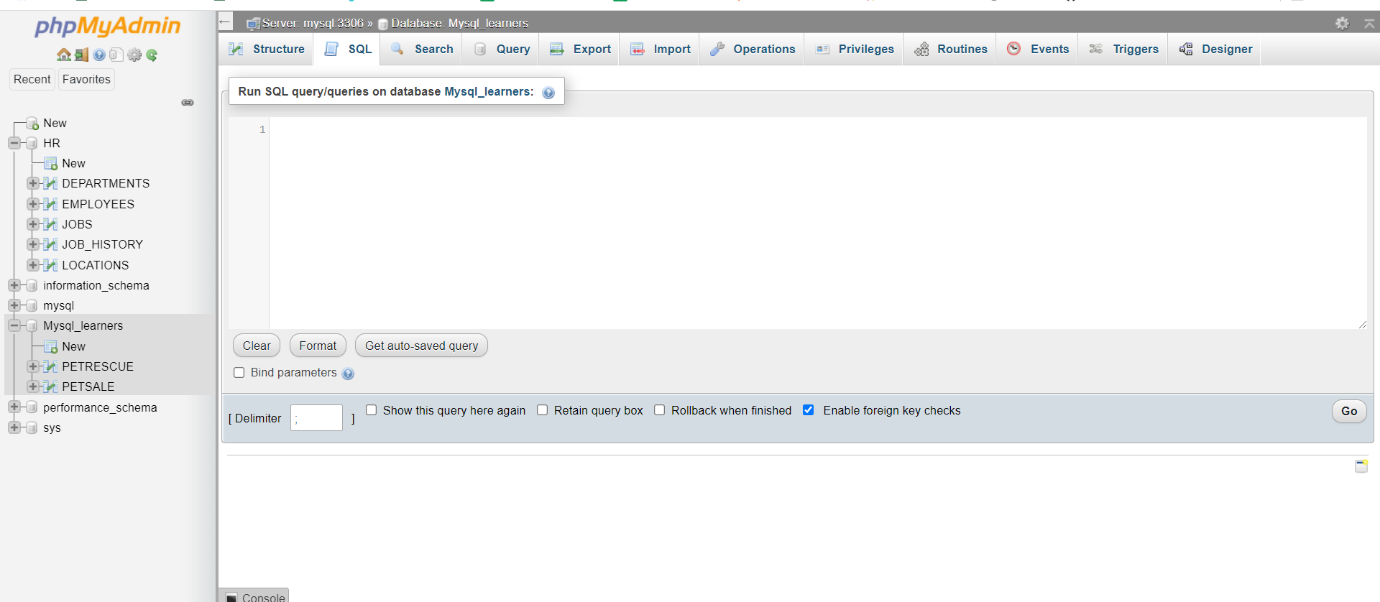


# Objectives

After completing this lab, you will be able to:

* Simplify a SELECT statement by using string patterns, ranges, or sets of values
* Sort the result set in either ascending or descending order and identify which column to use for the sorting order
* Eliminate duplicates from a result set and further restrict a result set

Once the tables are loaded open the sql editor to start executing the functions.



**Exercise 1: String Patterns**

In this exercise, you will go through some SQL problems on String Patterns.

1. Problem:

*Retrieve all employees whose address is in Elgin,IL.*

Hint

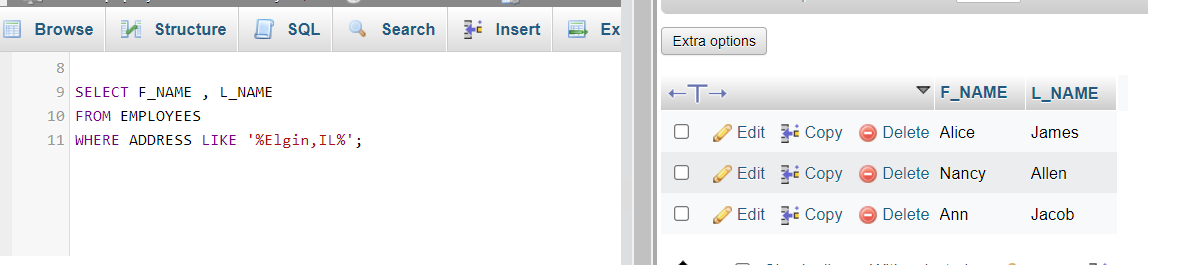
*Use the LIKE operator to find similar strings.*

Solution

* 1. 1
  2. 2
  3. 3
  4. SELECT F\_NAME , L\_NAME
  5. FROM EMPLOYEES
  6. WHERE ADDRESS LIKE '%Elgin,IL%';

Copied!

Output



1. Problem:

*Retrieve all employees who were born during the 1970’s.*

Hint

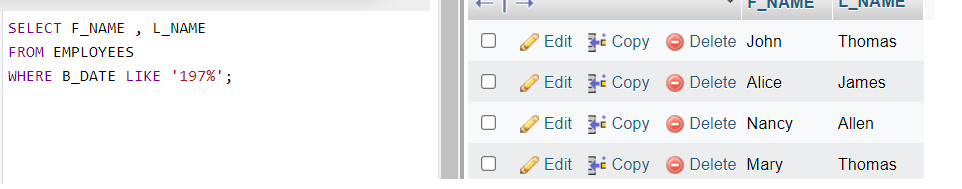
*Use the LIKE operator to find similar strings.*

Solution

* 1. 1
  2. 2
  3. 3
  4. SELECT F\_NAME , L\_NAME
  5. FROM EMPLOYEES
  6. WHERE B\_DATE LIKE '197%';

Copied!

Output



1. Problem:

*Retrieve all employees in department 5 whose salary is between 60000 and 70000.*

Hint

*Use the keyword BETWEEN for this SQL problem.*

Solution

* 1. 1
  2. 2
  3. 3
  4. SELECT \*
  5. FROM EMPLOYEES
  6. WHERE (SALARY BETWEEN 60000 AND 70000) AND DEP\_ID = 5;

Copied!

Output



**Exercise 2: Sorting**

In this exercise, you will go through some SQL problems on Sorting.

1. Problem:

*Retrieve a list of employees ordered by department ID.*

Hint

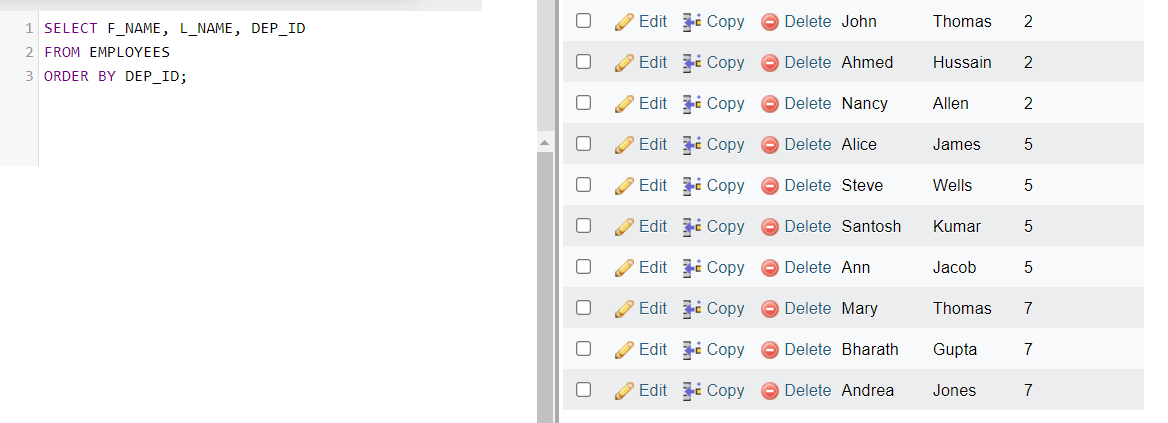
*Use the ORDER BY clause for this SQL problem. By default, the ORDER BY clause sorts the records in ascending order.*

Solution

* 1. 1
  2. 2
  3. 3
  4. SELECT F\_NAME, L\_NAME, DEP\_ID
  5. FROM EMPLOYEES
  6. ORDER BY DEP\_ID;

Copied!

Output



1. Problem:

*Retrieve a list of employees ordered in descending order by department ID and within each department ordered alphabetically in descending order by last name.*

Hint

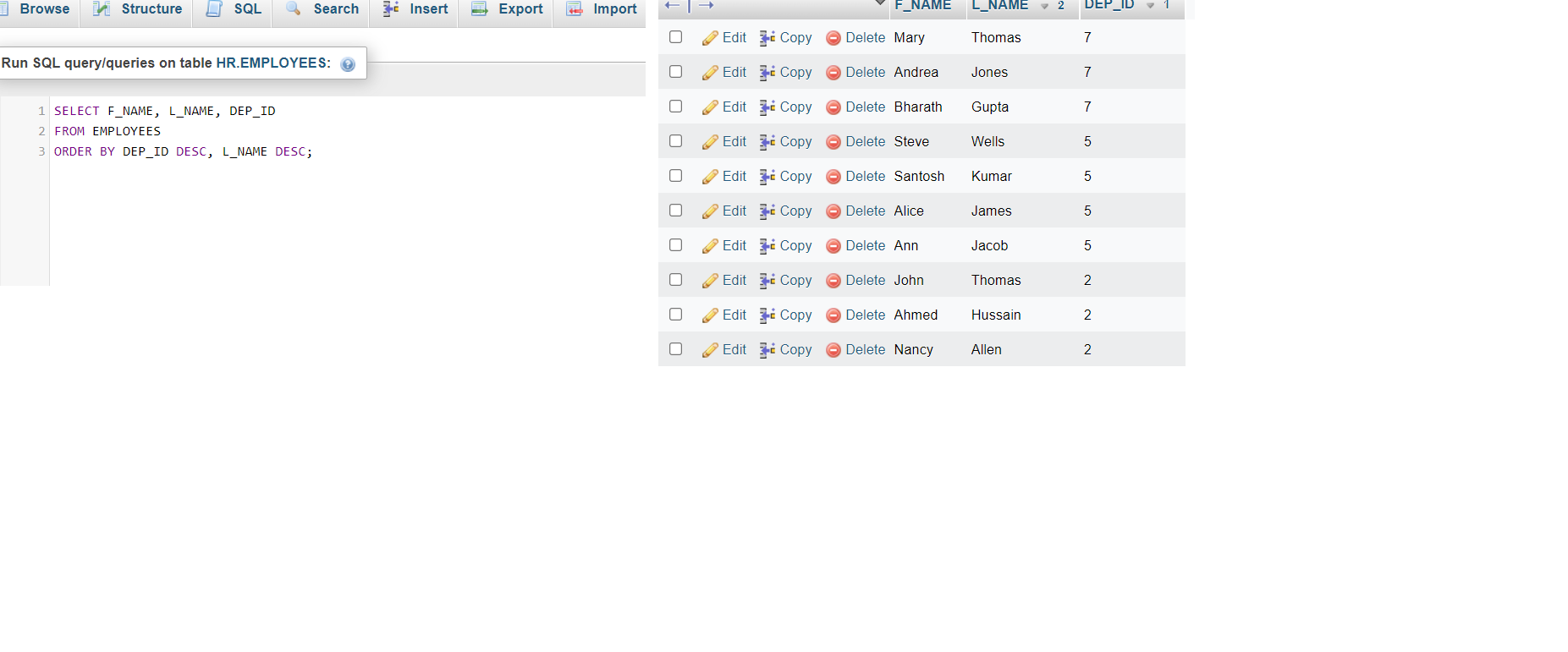
*Use the ORDER BY clause with DESC for this SQL problem.*

Solution

* 1. 1
  2. 2
  3. 3
  4. SELECT F\_NAME, L\_NAME, DEP\_ID
  5. FROM EMPLOYEES
  6. ORDER BY DEP\_ID DESC, L\_NAME DESC;

Copied!

Output



1. (Optional) Problem:

*In SQL problem 2 (Exercise 2 Problem 2), use department name instead of department ID. Retrieve a list of employees ordered by department name, and within each department ordered alphabetically in descending order by last name.*

Hint

*Department name is in the DEPARTMENTS table. So your query will need to retrieve data from more than one table. Donâ€™t worry if you are not able to figure this SQL problem out. Weâ€™ll cover working with multiple tables in the lecture****Working with Multiple Tables****.*

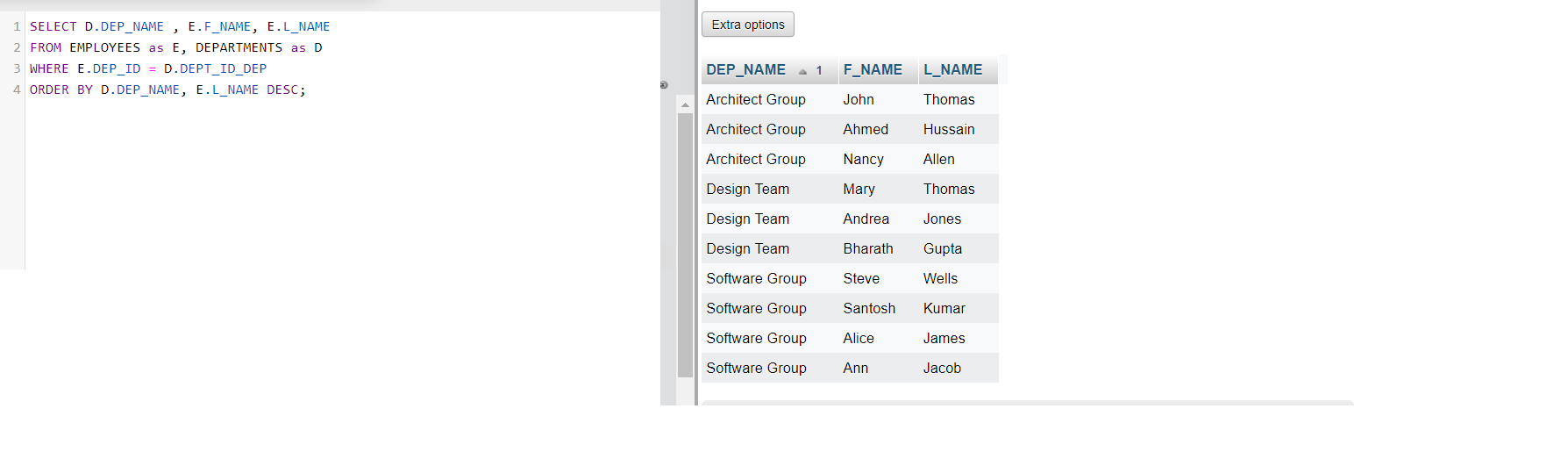
Solution

* 1. 1
  2. 2
  3. 3
  4. 4
  5. SELECT D.DEP\_NAME , E.F\_NAME, E.L\_NAME
  6. FROM EMPLOYEES as E, DEPARTMENTS as D
  7. WHERE E.DEP\_ID = D.DEPT\_ID\_DEP
  8. ORDER BY D.DEP\_NAME, E.L\_NAME DESC;

Copied!

*In the SQL Query above, D and E are aliases for the table names. Once you define an alias like D in your query, you can simply write D.COLUMN\_NAME rather than the full form DEPARTMENTS.COLUMN\_NAME.*

Output



**Exercise 3: Grouping**

In this exercise, you will go through some SQL problems on Grouping.

**NOTE:** The SQL problems in this exercise involve usage of SQL Aggregate functions AVG and COUNT. COUNT has been covered earlier. AVG is a function that can be used to calculate the Average or Mean of all values of a specified column in the result set. For example, to retrieve the average salary for all employees in the EMPLOYEES table, issue the query: SELECT AVG(SALARY) FROM EMPLOYEES;. You will learn more about AVG and other aggregate functions later in the lecture **Built-in Database Functions**.

1. Problem:

*For each department ID retrieve the number of employees in the department.*

Hint

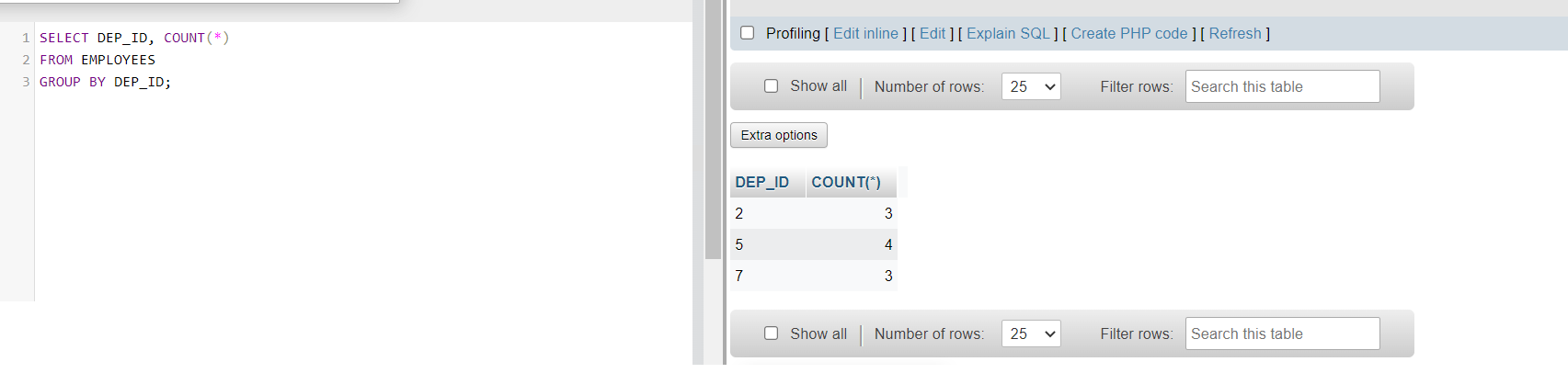
*Use COUNT(\*) to retrieve the total count of a column, and then GROUP BY.*

Solution

* 1. 1
  2. 2
  3. 3
  4. SELECT DEP\_ID, COUNT(\*)
  5. FROM EMPLOYEES
  6. GROUP BY DEP\_ID;

Copied!

Output



1. Problem:

*For each department retrieve the number of employees in the department, and the average employee salary in the department..*

Hint

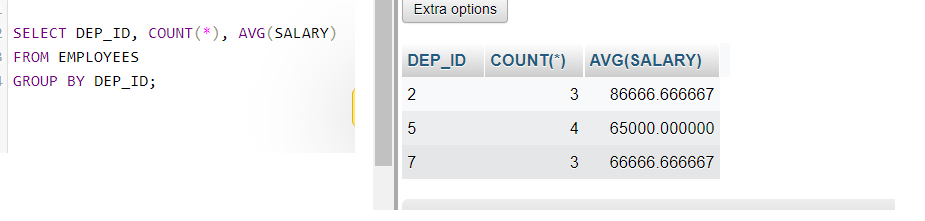
*Use COUNT(\*) to retrieve the total count of a column, and AVG() function to compute average salaries, and then GROUP BY.*

Solution

* 1. 1
  2. 2
  3. 3
  4. SELECT DEP\_ID, COUNT(\*), AVG(SALARY)
  5. FROM EMPLOYEES
  6. GROUP BY DEP\_ID;

Copied!

Output



1. Problem:

*Label the computed columns in the result set of SQL problem 2 (Exercise 3 Problem 2) as NUM\_EMPLOYEES and AVG\_SALARY.*

Hint

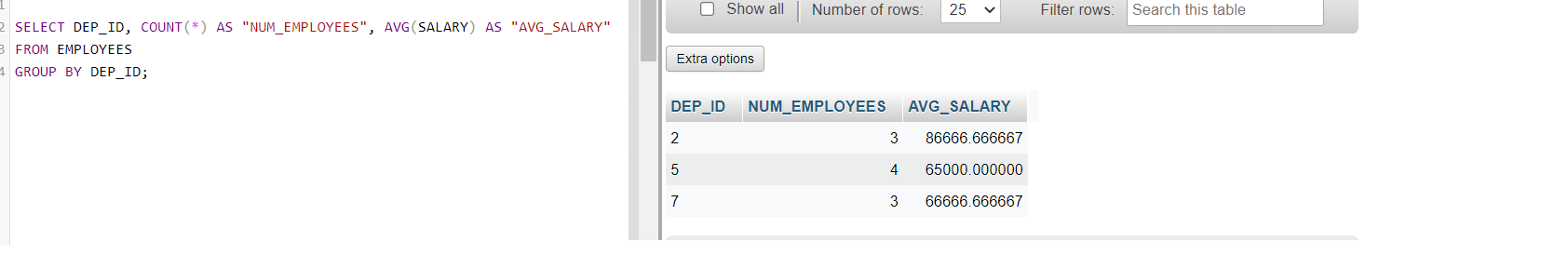
*Use SQL Aliases: column\_name AS alias\_name. For example, AVG(SALARY) AS “AVG\_SALARY”.*

Solution

* 1. 1
  2. 2
  3. 3
  4. SELECT DEP\_ID, COUNT(\*) AS "NUM\_EMPLOYEES", AVG(SALARY) AS "AVG\_SALARY"
  5. FROM EMPLOYEES
  6. GROUP BY DEP\_ID;

Copied!

Output



1. Problem:

*In SQL problem 3 (Exercise 3 Problem 3), order the result set by Average Salary..*

Hint

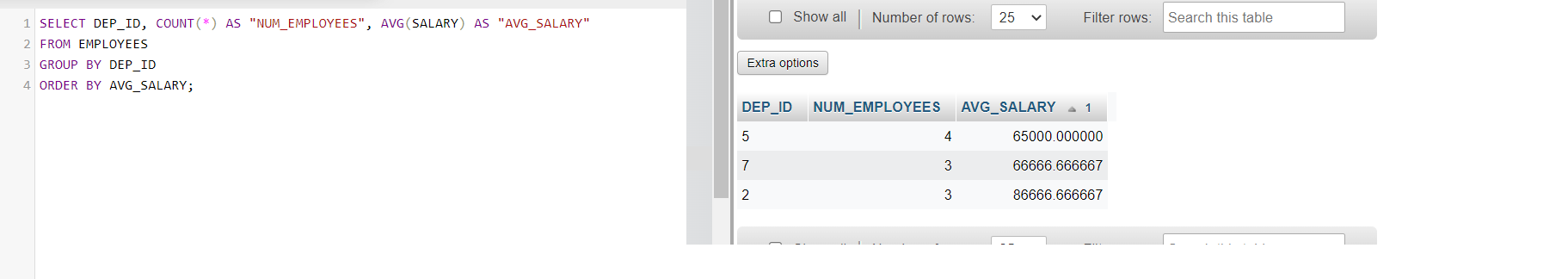
*Use ORDER BY after the GROUP BY.*

Solution

* 1. 1
  2. 2
  3. 3
  4. 4
  5. SELECT DEP\_ID, COUNT(\*) AS "NUM\_EMPLOYEES", AVG(SALARY) AS "AVG\_SALARY"
  6. FROM EMPLOYEES
  7. GROUP BY DEP\_ID
  8. ORDER BY AVG\_SALARY;

Copied!

Output



1. Problem:

*In SQL problem 4 (Exercise 3 Problem 4), limit the result to departments with fewer than 4 employees.*

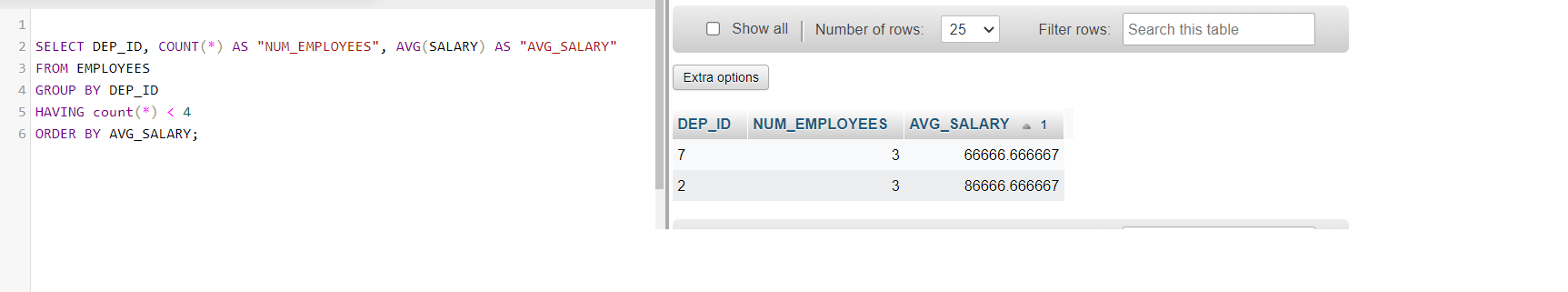
Hint

*Use HAVING after the GROUP BY, and use the count() function in the HAVING clause instead of the column label.*

Solution

* 1. SELECT DEP\_ID, COUNT(\*) AS "NUM\_EMPLOYEES", AVG(SALARY) AS "AVG\_SALARY"
  2. FROM EMPLOYEES
  3. GROUP BY DEP\_ID
  4. HAVING count(\*) < 4
  5. ORDER BY AVG\_SALARY;

Output



# Solution Script

If you would like to run all the solution queries of the SQL problems of this lab with a script, download the script below.Import the script to phpadmin mysql interface and run. Follow [Hands-on Lab : Create tables using SQL scripts and Load data into tables](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork/labs/MySQL/week2/Create_and%20_Load.md.html) on how to upload a script to phpmyadmin console and run it.

* [StringPattern-Sorting-Grouping\_Solution\_Script.sql](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork/labs/MySQL/week3/StringPattern-Sorting-Grouping_Solution_Script.sql)

### Congratulations! You have completed this lab, and you are ready for the next topic.