Arial

Step 4 of 8

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.

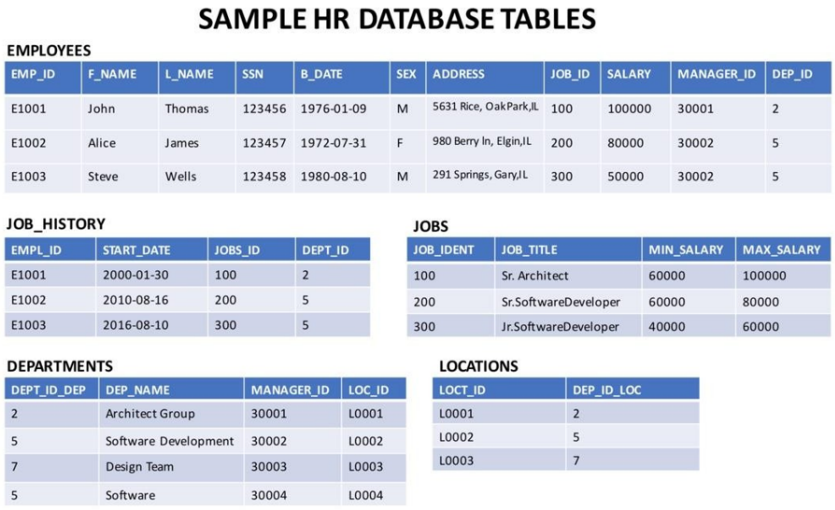
Software Used in this Lab

In this lab, you will use MySQL. 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:

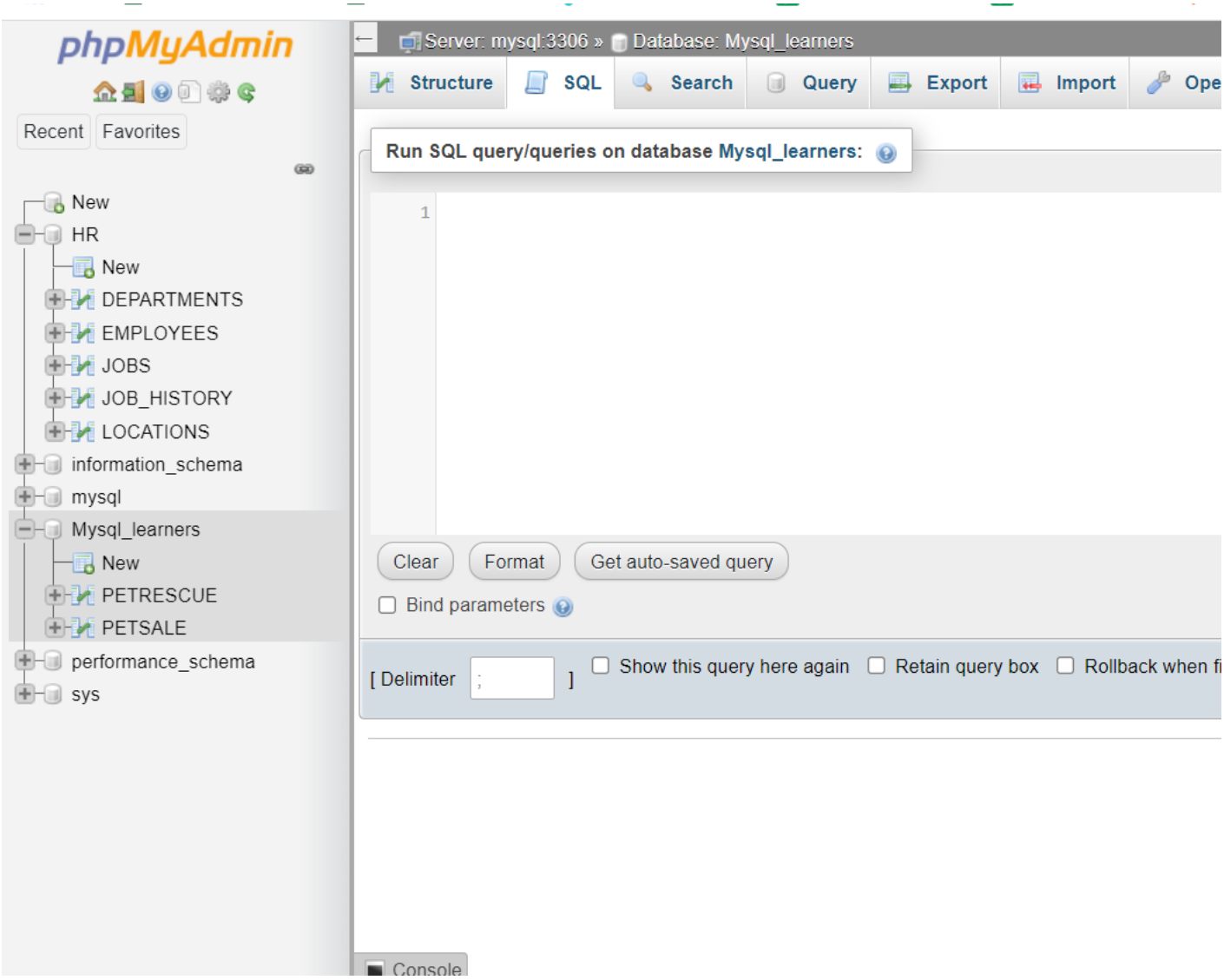


Objectives

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.

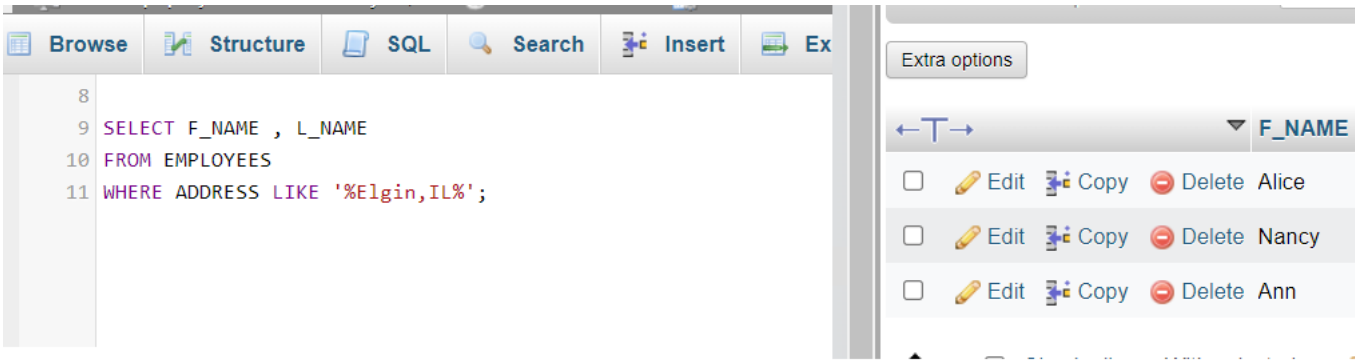
   ▼ Hint

   Use the LIKE operator to find similar strings.

   ▼ Solution

```
SELECT F_NAME , L_NAME
FROM EMPLOYEES
WHERE ADDRESS LIKE '%Elgin,IL%';
```

   ▼ Output



2. Problem:

Retrieve all employees who were born during the 1970's.

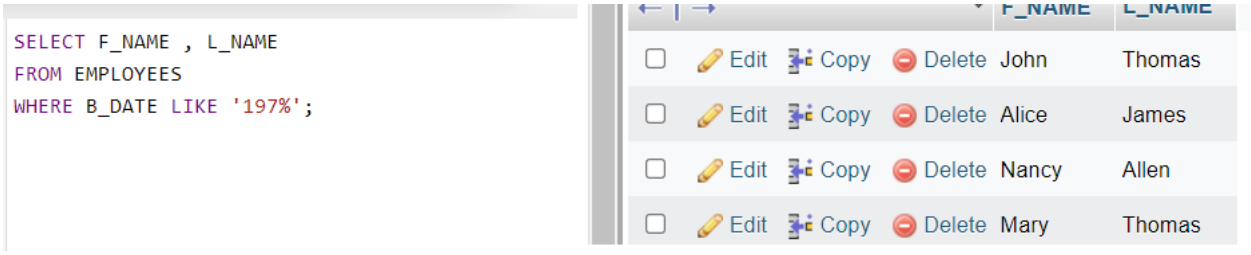
▼ Hint

Use the LIKE operator to find similar strings.

▼ Solution

```
SELECT F_NAME , L_NAME
FROM EMPLOYEES
WHERE B_DATE LIKE '197%';
```

▼ Output



3. Problem:

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

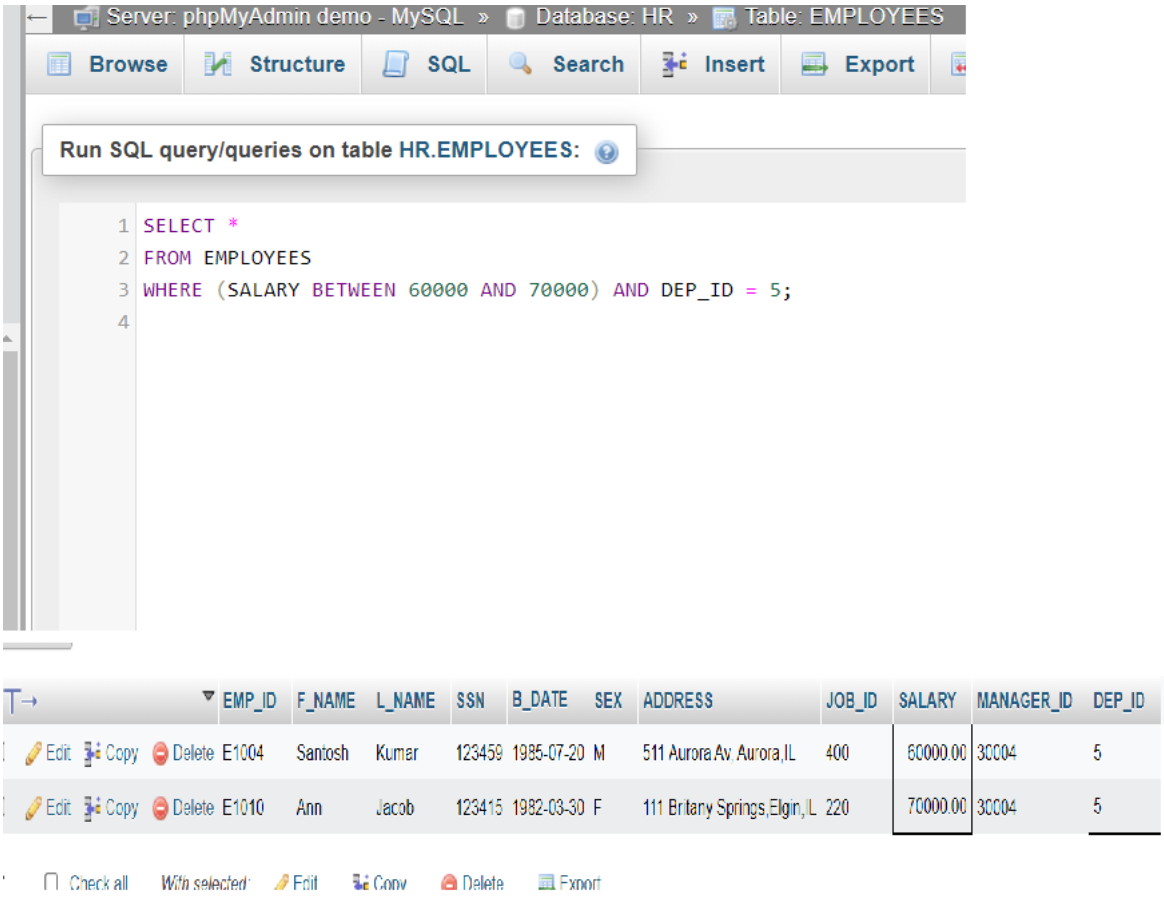
▼ Hint

Use the keyword BETWEEN for this SQL problem.

▼ Solution

```
SELECT *
FROM EMPLOYEES
WHERE (SALARY BETWEEN 60000 AND 70000) AND DEP_ID = 5;
```

▼ 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

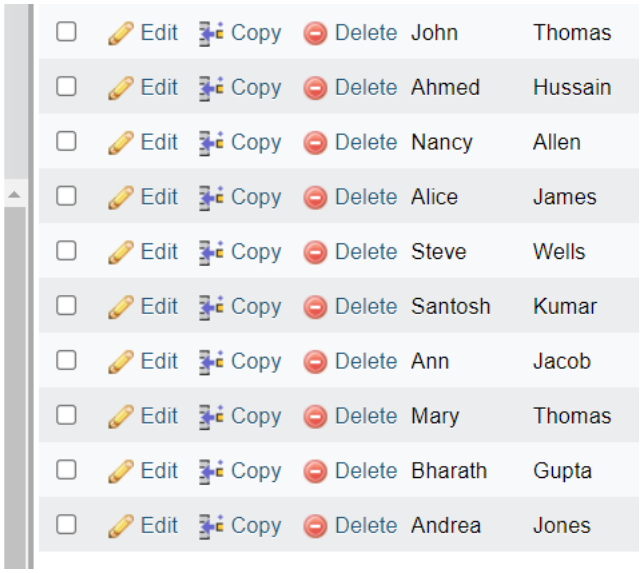
Use the ORDER BY clause for this SQL problem. By default, the ORDER BY clause sorts the records in ascending order.

▼ Solution

```
SELECT F_NAME, L_NAME, DEP_ID
FROM EMPLOYEES
ORDER BY DEP_ID;
```

▼ Output

```
1 SELECT F_NAME, L_NAME, DEP_ID
2 FROM EMPLOYEES
3 ORDER BY DEP_ID;
```



2. 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.

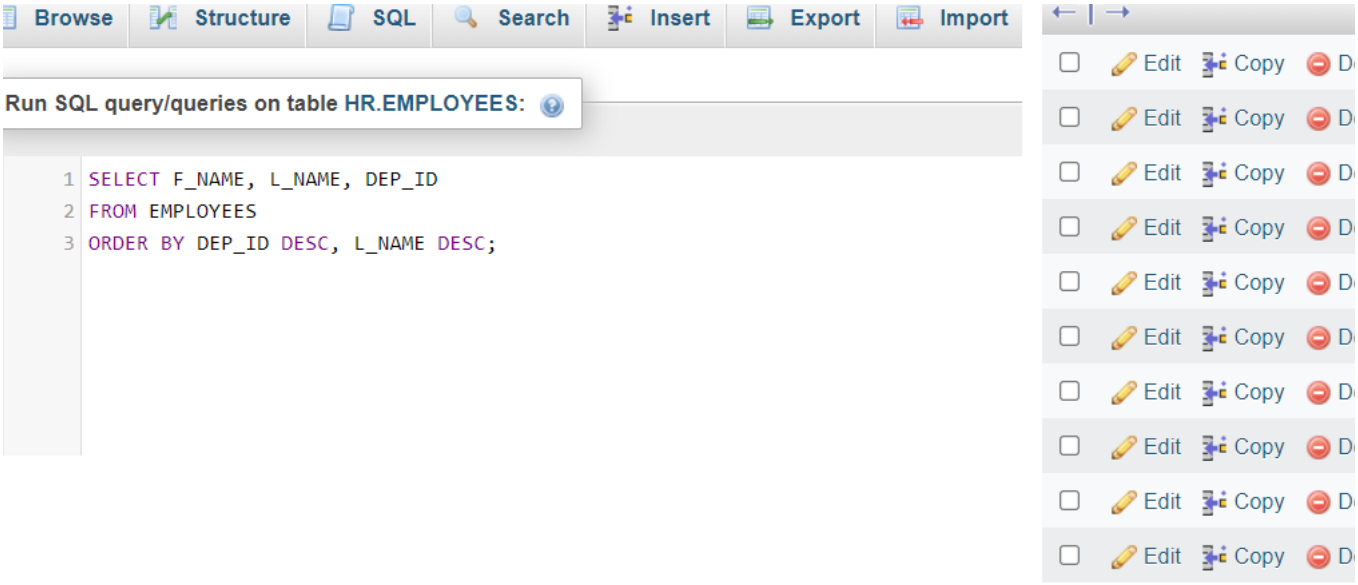
▼ Hint

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

▼ Solution

```
SELECT F_NAME, L_NAME, DEP_ID
FROM EMPLOYEES
ORDER BY DEP_ID DESC, L_NAME DESC;
```

▼ Output



3. (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

```
SELECT D.DEP_NAME , E.F_NAME, E.L_NAME
FROM EMPLOYEES as E, DEPARTMENTS as D
WHERE E.DEP_ID = D.DEPT_ID_DEP
ORDER BY D.DEP_NAME, E.L_NAME DESC;
```

    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

```sql
SELECT D.DEP_NAME , E.F_NAME, E.L_NAME
FROM EMPLOYEES as E, DEPARTMENTS as D
WHERE E.DEP_ID = D.DEPT_ID_DEP
ORDER BY D.DEP_NAME, E.L_NAME DESC;
```

Extra options

DEP_NAME ▲ 1	F_NAM
Architect Group	John
Architect Group	Ahmed
Architect Group	Nancy
Design Team	Mary
Design Team	Andrea
Design Team	Bharatl
Software Group	Steve
Software Group	Santos
Software Group	Alice
Software Group	Ann

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

```sql
SELECT DEP_ID, COUNT(*)
FROM EMPLOYEES
GROUP BY DEP_ID;
```

▼ Output

```sql
SELECT DEP_ID, COUNT(*)
FROM EMPLOYEES
GROUP BY DEP_ID;
```

Profiling [Edit inline] [E

Show all | Number

Extra options

DEP_ID	COUNT(*)
2	3
5	4
7	3

Show all | Number

2. Problem:

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

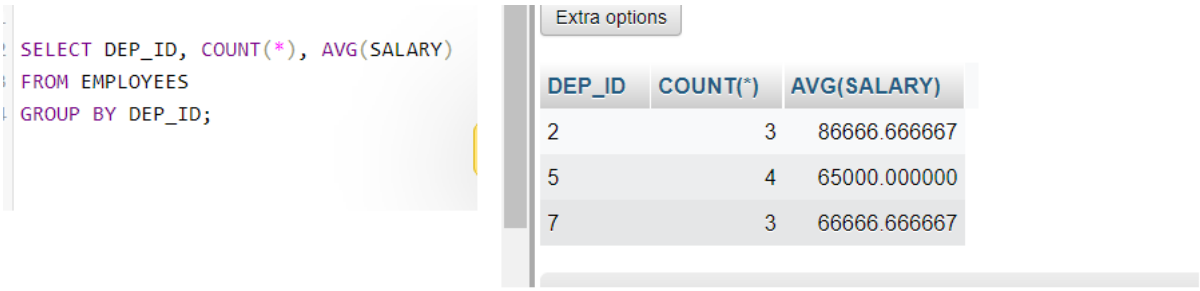
▼ Hint

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

▼ Solution

```
SELECT DEP_ID, COUNT(*), AVG(SALARY)
FROM EMPLOYEES
GROUP BY DEP_ID;
```

▼ Output



3. Problem:

   Label the computed columns in the result set of SQL problem 2 (Exercise 3 Problem 2) as NUMEMPLOYEES and AVGSALARY.

   ▼ Hint

   Use SQL Aliases: `column_name AS alias_name`. For example, AVG(SALARY) AS "AVG_SALARY".

   ▼ Solution

   ```
   SELECT DEP_ID, COUNT(*) AS "NUM_EMPLOYEES", AVG(SALARY) AS "AVG_SALARY"
   FROM EMPLOYEES
   GROUP BY DEP_ID;
   ```

   ▼ Output

4. 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

   ```
   SELECT DEP_ID, COUNT(*) AS "NUM_EMPLOYEES", AVG(SALARY) AS "AVG_SALARY"
   FROM EMPLOYEES
   GROUP BY DEP_ID
   ORDER BY AVG_SALARY;
   ```

   ▼ Output

5. 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

   ```
   SELECT DEP_ID, COUNT(*) AS "NUM_EMPLOYEES", AVG(SALARY) AS "AVG_SALARY"
   FROM EMPLOYEES
   GROUP BY DEP_ID
   HAVING count(*) < 4
   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 on how to upload a script to phpmyadmin console and run it.

- StringPattern-Sorting-Grouping*Solution*Script.sql

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

Author(s)

Lakshmi Holla

Malika Singla

Changelog

Date	Version	Changed by	Change Description
2021-11-01	0.1	Lakshmi Holla, Malika Singla	Initial Version