Hands-on Lab: Working with Joins in MySQL using phpMyAdmin

Estimated time needed: 20 minutes

SQL JOIN is a clause that combines rows from two or more tables based on a related column between them. The table's relationship is established by comparing the values in the columns. The purpose of using JOINs is to retrieve data from multiple tables in a single query. There are four types of JOINs in SQL: INNER JOIN, LEFT JOIN, RIGHT JOIN, and FULL OUTER JOIN.

- INNER JOIN: Returns only the rows with matching values in both tables.
- LEFT JOIN: Returns all the rows from the left table and matching rows from the right table.
- RIGHT JOIN: Returns all the rows from the right table and matching rows from the left table.
- FULL OUTER JOIN: Returns all the rows when a match in the left or right table.

Objectives

By the end of this lab, you'll be able to:

- Write SQL queries on multiple tables using INNER JOINS
- Write SQL queries on multiple tables using OUTER JOINS

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 internal. You will be working on a sample HR database. This HR database schema consists of five tables: **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:

SAMPLE HR DATABASE TABLES

EMPLOYEES

EMP_ID	F_NAME	L_NAME	SSN	B_DATE	SEX	ADDRESS	JOB_ID	SALARY	MANAGER_ID	DEP_ID
E1001	John	Thomas	123456	1976-01-09	M	5631 Rice, OakPark,IL	100	100000	30001	2
E1002	Alice	James	123457	1972-07-31	F	980 Berry ln, Elgin,IL	200	80000	30002	5
E1003	Steve	Wells	123458	1980-08-10	M	291 Springs, Gary,IL	300	50000	30002	5

JOB_HISTORY

EMPL_ID	START_DATE	JOBS_ID	DEPT_ID
E1001	2000-01-30	100	2
E1002	2010-08-16	200	5
E1003	2016-08-10	300	5

JOBS

JOB_IDENT	JOB_TITLE	MIN_SALARY	MAX_SALARY
100	Sr. Architect	60000	100000
200	Sr.SoftwareDeveloper	60000	80000
300	Jr.SoftwareDeveloper	40000	60000

DEPARTMENTS

DEPT_ID_DEP	DEP_NAME	MANAGER_ID	LOC_ID
2	Architect Group	30001	L0001
5	Software Development	30002	L0002
7	Design Team	30003	L0003

LOCATIONS

LOCT_ID	DEP_ID_LOC
L0001	2
L0002	5
L0003	7

In this lab, you will run through some SQL practice problems that will provide hands-on experience with the different kinds of join operations.

NOTE: This lab requires you to have all five of these tables of the HR database populated with sample data on MySQL.

Load the database

Using the skills acquired in the previous modules, you should first create the database in MySQL. Follow the steps below:

1. Open the phpMyAdmin interface from the Skills Network Toolbox in Cloud IDE.
2. Create a blank database named 'HR'. Use the script shared in the link below to create the required tables.
   Script_Create_Tables.sql
3. Download the files in the links below to your local machine (if not already done in previous labs).
   Departments.csv
   Jobs.csv
   JobsHistory.csv
   Locations.csv
   Employees.csv
4. Use these files to the interface as data for respective tables in the 'HR' database.

JOINS

Let us see some examples of JOINS being used to query the data.

1. Retrieve the names and job start dates of all employees who work for department number 5.

We need to use the Inner join operation with the EMPLOYEES table as the left table and the JOB_HISTORY table as the right table. The join will be made over employee ID, and the query response will be filtered for the Department ID value 5.
The query for this question will be as shown below.

```
SELECT E.F_NAME,E.L_NAME, JH.START_DATE
FROM EMPLOYEES as E
INNER JOIN JOB_HISTORY as JH
ON E.EMP_ID=JH.EMPL_ID
```

```
WHERE E.DEP_ID ='5';
```

2. Retrieve employee ID, last name, department ID, and department name for all employees.

For this, you must use the Left Outer Join operation with the EMPLOYEES table as the left table and the DEPARTMENTS table as the right table. The join will happen on the Department ID.

Left join query retrieves all employees, including their department details if available. If an employee does not belong to any department, the department fields will be NULL.

The query will be written as follows:

```
SELECT E.EMP_ID, E.L_NAME, E.DEP_ID, D.DEP_NAME
FROM EMPLOYEES AS E
LEFT OUTER JOIN DEPARTMENTS AS D
ON E.DEP_ID=D.DEPT_ID_DEP;
```

3. Retrieve the First name, Last name, and Department name of all employees.

For this, you will use the Full Outer Join operation with the EMPLOYEES table as the left table and the DEPARTMENTS table as the right table. A full outer join in MySQL is implemented as a UNION of left and right outer joins.

Full Outer Join query retrieves all employees and departments, showing all combinations. If an employee is not associated with a department, or a department has no employees, the missing fields will be NULL.

The query will be written as shown below.

```
SELECT E.F_NAME, E.L_NAME, D.DEP_NAME
FROM EMPLOYEES AS E
LEFT OUTER JOIN DEPARTMENTS AS D
ON E.DEP_ID = D.DEPT_ID_DEP
UNION
SELECT E.F_NAME, E.L_NAME, D.DEP_NAME
FROM EMPLOYEES AS E
RIGHT OUTER JOIN DEPARTMENTS AS D
ON E.DEP_ID=D.DEPT_ID_DEP
```

Practice Problems

1. Retrieve the names, job start dates, and job titles of all employees who work for department number 5.

   ▶ Hint
   ▶ Solution

2. Retrieve employee ID, last name, and department ID for all employees but department names for only those born before 1980.

   ▶ Hint
   ▶ Solution

3. Retrieve the first name and last name of all employees but department ID and department names only for male employees.

   ▶ Hint
   ▶ Solution

Conclusion

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

You now can:

- Query multiple tables using INNER JOINS
- Query multiple tables using LEFT/RIGHT OUTER JOINS
- Query multiple tables using FULL OUTER JOINS

Author(s)

Lakshmi Holla

Malika Singla

Abhishek Gagneja