

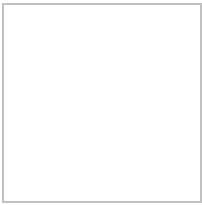
Hands-on Lab: Working with Multiple Tables in MySQL using phpMyAdmin

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:

- Write SQL queries that access more than one table
- Compose queries that access multiple tables using a nested statement in the WHERE clause
- Build queries with multiple tables in the FROM clause
- Write Implicit Join queries with join criteria specified in the WHERE clause
- Specify aliases for table names and qualify column names with table aliases

In this lab, you will through some SQL practice problems that will provide hands-on experience with SQL queries that access multiple tables. You will be:

- Accessing Multiple Tables with Sub-Queries
- Accessing Multiple Tables with Implicit Joins

How does an Implicit version of CROSS JOIN (also known as Cartesian Join) statement syntax look?

1. 1
2. 2

1. SELECT column_name(s)
2. FROM table1, table2;

Copied!

How does an Implicit version of INNER JOIN statement syntax look?

1. 1
2. 2
3. 3

1. SELECT column_name(s)
2. FROM table1, table2
3. WHERE table1.column_name = table2.column_name;

Copied!

Exercise 1: Accessing Multiple Tables with Sub-Queries

1. Problem:

Retrieve only the EMPLOYEES records that correspond to jobs in the JOBS table.

- ▶ Solution
- ▶ Output

2. Problem:

Retrieve only the list of employees whose JOB_TITLE is Jr. Designer.

- ▶ Solution
- ▶ Output

3. Problem:

Retrieve JOB information and who earn more than \$70,000.

- ▶ Solution
- ▶ Output

4. Problem:

Retrieve JOB information and list of employees whose birth year is after 1976.

- ▶ Solution
- ▶ Output

5. Problem:

Retrieve JOB information and list of female employees whose birth year is after 1976.

- ▶ Solution
- ▶ Output

Exercise 2: Accessing Multiple Tables with Implicit Joins

1. Problem:

Perform an implicit cartesian/cross join between EMPLOYEES and JOBS tables.

- ▶ Solution
- ▶ Output

2. Problem:

Retrieve only the EMPLOYEES records that correspond to jobs in the JOBS table.

- ▶ Solution
- ▶ Output

3. Problem:

Redo the previous query, using shorter aliases for table names.

- ▶ Solution
- ▶ Output

4. Problem:

Redo the previous query, but retrieve only the Employee ID, Employee Name and Job Title.

- ▶ Solution
- ▶ Output

5. Problem:

Redo the previous query, but specify the fully qualified column names with aliases in the SELECT clause.

- ▶ Solution
- ▶ 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 mysql phpadmin interface and run. Follow [Hands-on Lab : Create tables using SQL scripts and Load data into tables](#) on how to import a script to MYsql phpadmin interface and run it.

- [MultipleTables_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
2023-05-10	0.3	Eric Hao & Vladislav Boyko	Updated Page Frames
2023-05-04	0.2	Rahul Jaideep	Updated Markdown file
2021-11-01	0.1	Lakshmi Holla, Malika Singla	Initial Version

© IBM Corporation 2023. All rights reserved.