**Hands-on Lab: Working with Joins 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**

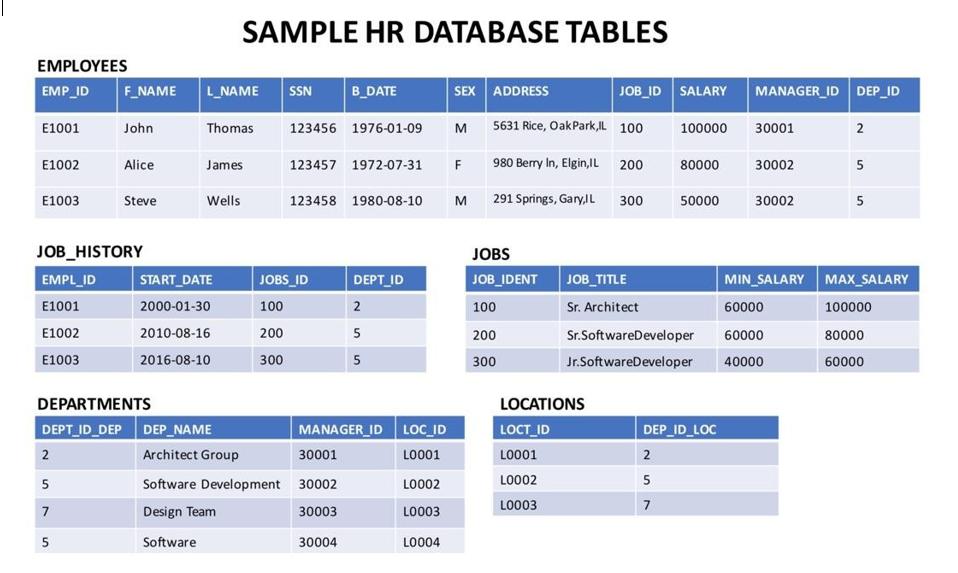
In this lab, you will use [MySQL](https://www.mysql.com/?utm_medium=Exinfluencer&utm_source=Exinfluencer&utm_content=000026UJ&utm_term=10006555&utm_id=NA-SkillsNetwork-Channel-SkillsNetworkCoursesIBMDB0110ENSkillsNetwork24601058-2021-01-01). MySQL is a Relational Database Management System (RDBMS) designed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



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 5 of these tables of the HR database populated with sample data on MySQL. If you don't have the tables above populated with sample data on MySQL, please go through the lab below first:

[Hands-on Lab: Create and Load Tables using SQL Scripts](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork/labs/MySQL/week2/Create_and%20_Load.md.html)

**How does a CROSS JOIN (also known as Cartesian Join) statement syntax look?**

1. 1
2. 2
3. 3
4. SELECT column\_name(s)
5. FROM table1
6. CROSS JOIN table2;

Copied!

**How does an INNER JOIN statement syntax look?**

1. 1
2. 2
3. 3
4. 4
5. 5
6. SELECT column\_name(s)
7. FROM table1
8. INNER JOIN table2
9. ON table1.column\_name = table2.column\_name;
10. WHERE condition;

Copied!

**How does a LEFT OUTER JOIN statement syntax look?**

1. 1
2. 2
3. 3
4. 4
5. 5
6. SELECT column\_name(s)
7. FROM table1
8. LEFT OUTER JOIN table2
9. ON table1.column\_name = table2.column\_name
10. WHERE condition;

Copied!

**How does a RIGHT OUTER JOIN statement syntax look?**

1. 1
2. 2
3. 3
4. 4
5. 5
6. SELECT column\_name(s)
7. FROM table1
8. RIGHT OUTER JOIN table2
9. ON table1.column\_name = table2.column\_name
10. WHERE condition;

Copied!

**How does a FULL OUTER JOIN statement syntax look?**

1. 1
2. 2
3. 3
4. 4
5. 5
6. 6
7. 7
8. 8
9. 9
10. 10
11. 11
12. 12
13. 13
14. SELECT column\_name(s)
15. FROM table1
16. LEFT OUTER JOIN table2
17. ON table1.column\_name = table2.column\_name
18. WHERE condition
19. UNION
20. SELECT column\_name(s)
21. FROM table1
22. RIGHT OUTER JOIN table2
23. ON table1.column\_name = table2.column\_name
24. WHERE condition

Copied!

**Union operator**

The UNION operator is used to combine the result-set of two or more SELECT statements.

Every SELECT statement within UNION must have the same number of columns  
The columns must also have similar data types  
The columns in every SELECT statement must also be in the same order

1. 1
2. 2
3. 3
4. SELECT column\_name(s) FROM table1
5. UNION
6. SELECT column\_name(s) FROM table2;

Copied!

**How does a SELF JOIN statement syntax look?**

1. 1
2. 2
3. 3
4. SELECT column\_name(s)
5. FROM table1 T1, table1 T2
6. WHERE condition;

Copied!

**Exercise**

1. Problem:

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

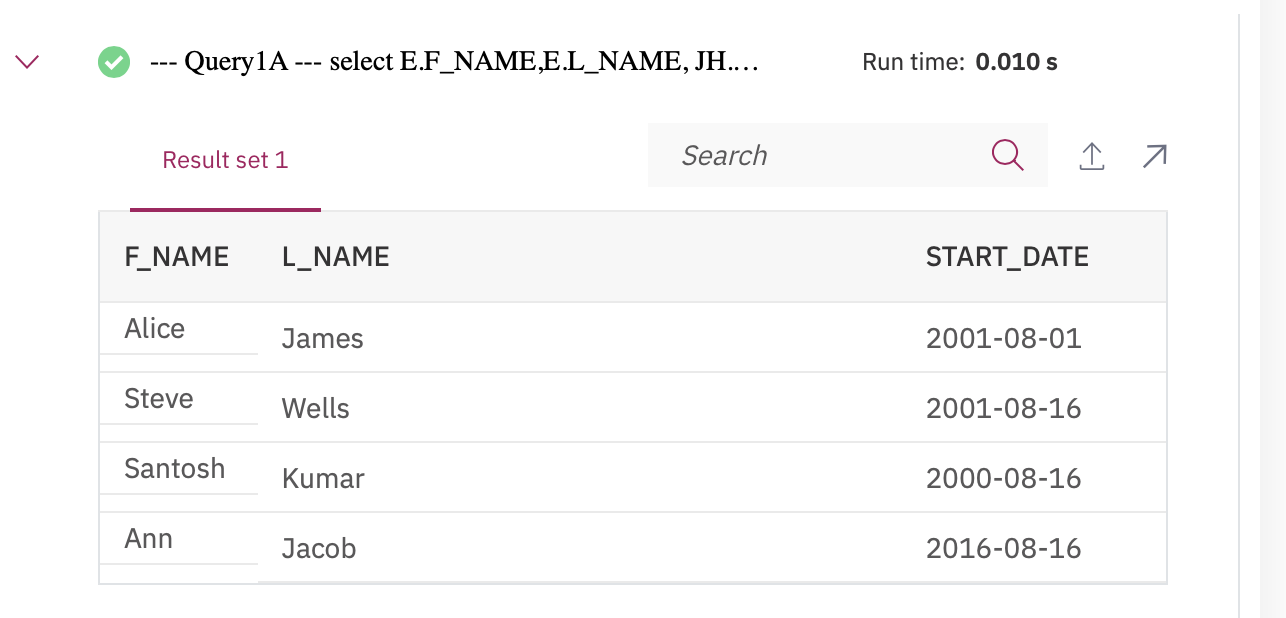
Hint

*Use the Inner join operation with the EMPLOYEES table as the left table and the JOB\_HISTORY table as the right table.*

Solution

* 1. 1
  2. 2
  3. 3
  4. 4
  5. select E.F\_NAME,E.L\_NAME, JH.START\_DATE
  6. from EMPLOYEES as E
  7. INNER JOIN JOB\_HISTORY as JH on E.EMP\_ID=JH.EMPL\_ID
  8. where E.DEP\_ID ='5';

Copied!

Output

1. Problem:

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

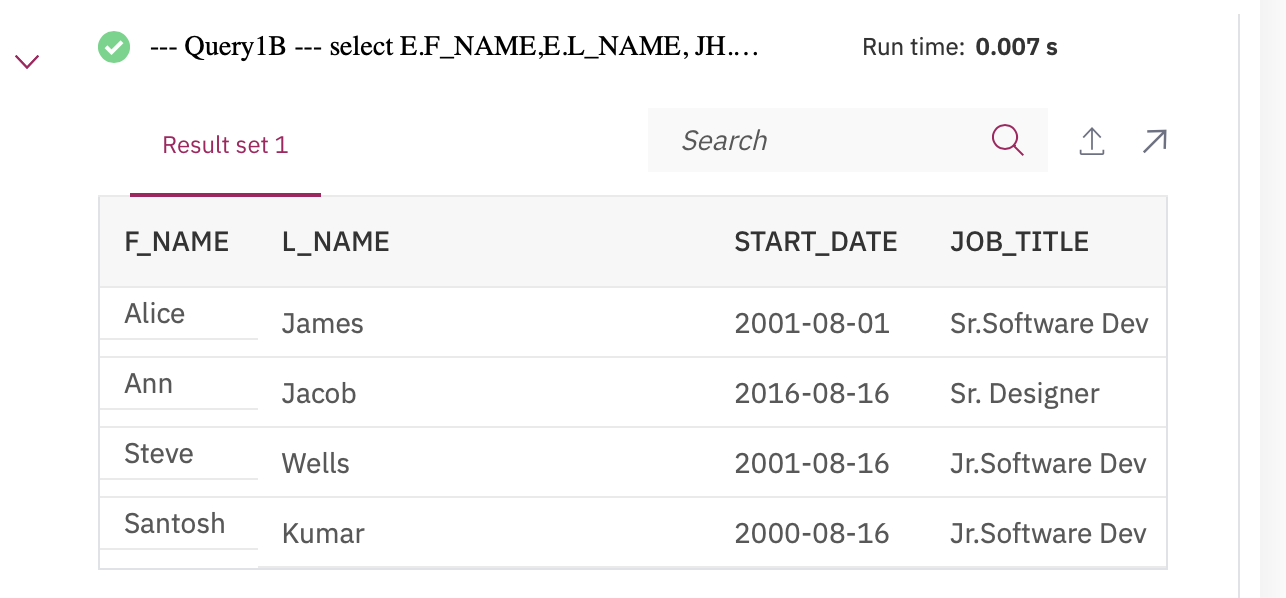
Hint

*Perform an INNER JOIN with 3 tables ⬜ EMPLOYEES, JOB\_HISTORY, JOBS.*

Solution

* 1. 1
  2. 2
  3. 3
  4. 4
  5. 5
  6. select E.F\_NAME,E.L\_NAME, JH.START\_DATE, J.JOB\_TITLE
  7. from EMPLOYEES as E
  8. INNER JOIN JOB\_HISTORY as JH on E.EMP\_ID=JH.EMPL\_ID
  9. INNER JOIN JOBS as J on E.JOB\_ID=J.JOB\_IDENT
  10. where E.DEP\_ID ='5';

Copied!

Output

1. Problem:

*Perform a Left Outer Join on the EMPLOYEES and DEPARTMENT tables and select employee id, last name, department id and department name for  
all employees.*

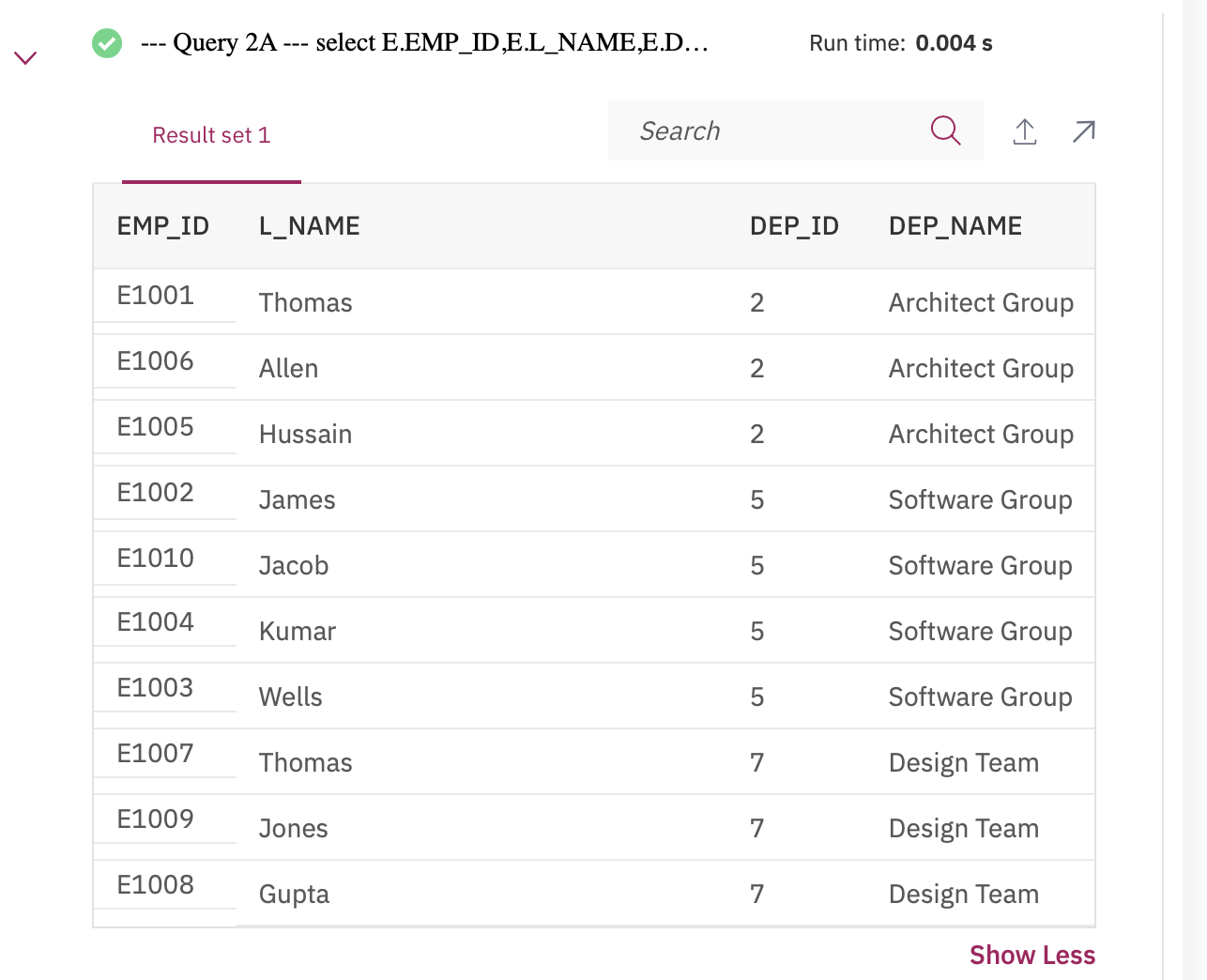
Hint

*Use the Left Outer Join operation with the EMPLOYEES table as the left table and the DEPARTMENTS table as the right table.*

Solution

* 1. 1
  2. 2
  3. 3
  4. select E.EMP\_ID,E.L\_NAME,E.DEP\_ID,D.DEP\_NAME
  5. from EMPLOYEES AS E
  6. LEFT OUTER JOIN DEPARTMENTS AS D ON E.DEP\_ID=D.DEPT\_ID\_DEP;

Copied!

Output

1. Problem:

*Re-write the previous query but limit the result set to include only the rows for employees born before 1980.*

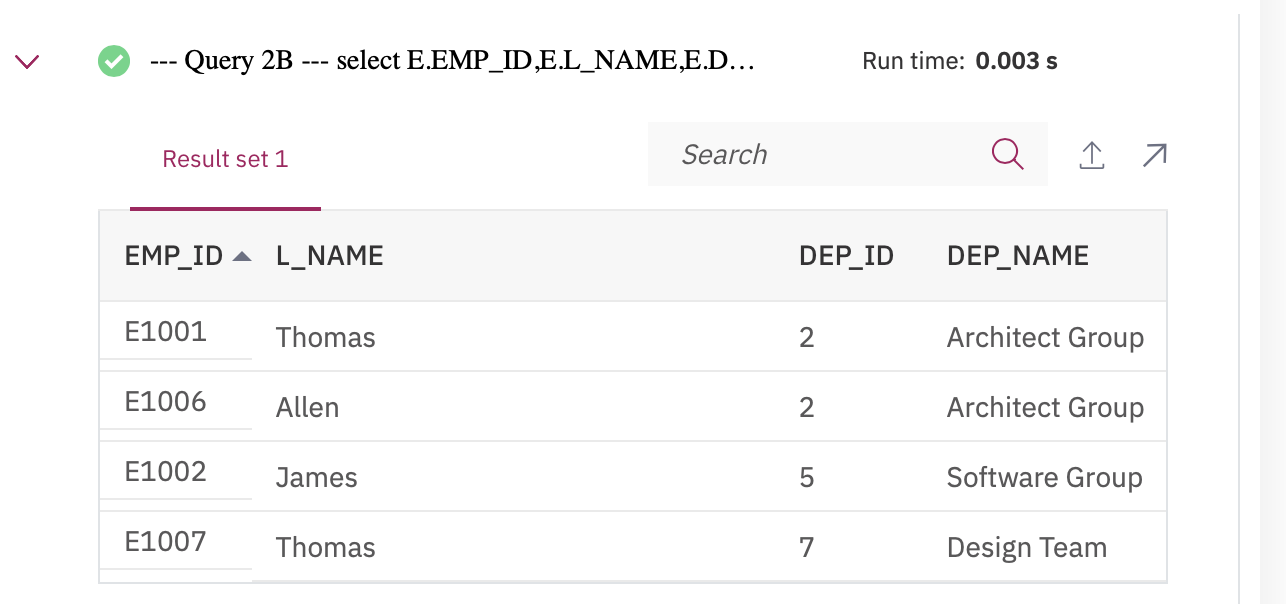
Hint

*Use a WHERE clause and Left Outer Join operation. Alternatively, you could also use an INNER JOIN.*

Solution

* 1. 1
  2. 2
  3. 3
  4. 4
  5. select E.EMP\_ID,E.L\_NAME,E.DEP\_ID,D.DEP\_NAME
  6. from EMPLOYEES AS E
  7. LEFT OUTER JOIN DEPARTMENTS AS D ON E.DEP\_ID=D.DEPT\_ID\_DEP
  8. where YEAR(E.B\_DATE) < 1980;

Copied!

Output

1. Problem:

*Re-write the previous query but have the result set include all the employees but department names for only the employees who were born before 1980.*

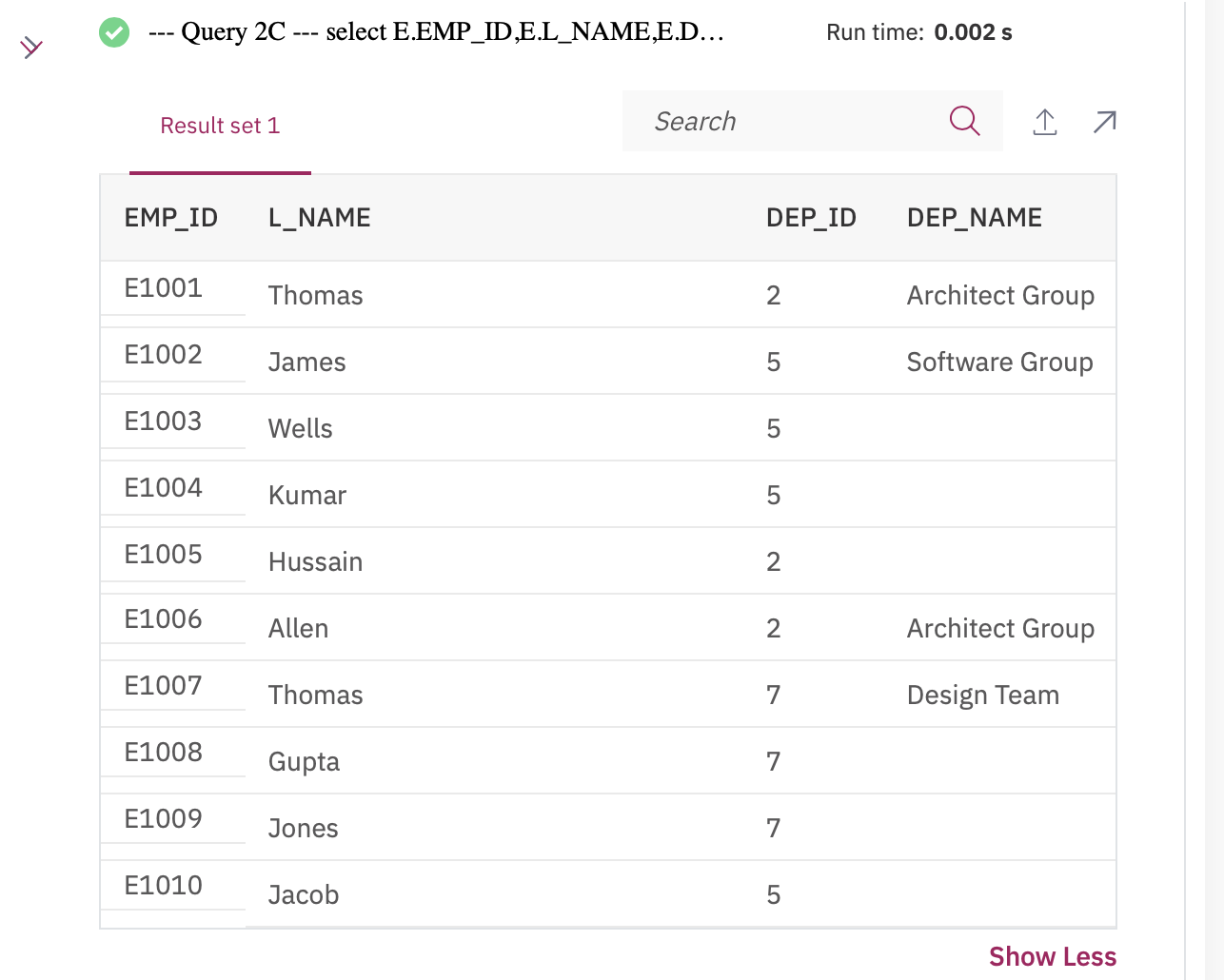
Hint

*Use an AND in the LEFT OUTER JOIN clause.*

Solution

* 1. 1
  2. 2
  3. 3
  4. 4
  5. select E.EMP\_ID,E.L\_NAME,E.DEP\_ID,D.DEP\_NAME
  6. from EMPLOYEES AS E
  7. LEFT OUTER JOIN DEPARTMENTS AS D ON E.DEP\_ID=D.DEPT\_ID\_DEP
  8. AND YEAR(E.B\_DATE) < 1980;

Copied!

Output

1. Problem:

*Perform a Full Join on the EMPLOYEES and DEPARTMENT tables and select the First name, Last name and Department name of all employees.*

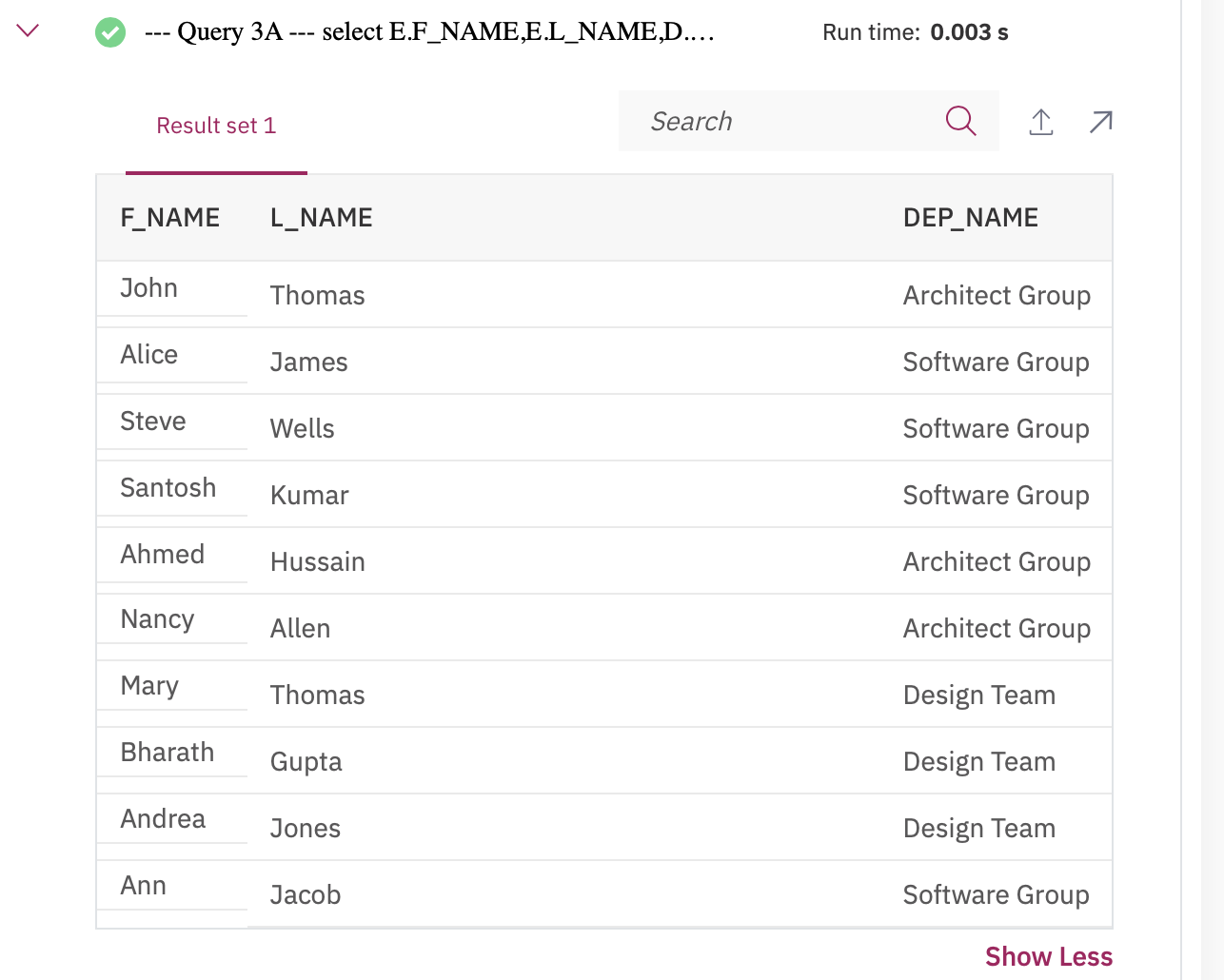
Hint

*Use the Full Outer Join operation with the EMPLOYEES table as the left table and the DEPARTMENTS table as the right table.*

Solution

* 1. 1
  2. 2
  3. 3
  4. 4
  5. 5
  6. 6
  7. 7
  8. 8
  9. 9
  10. select E.F\_NAME,E.L\_NAME,D.DEP\_NAME
  11. from EMPLOYEES AS E
  12. LEFT OUTER JOIN DEPARTMENTS AS D ON E.DEP\_ID=D.DEPT\_ID\_DEP
  13. UNION
  14. select E.F\_NAME,E.L\_NAME,D.DEP\_NAME
  15. from EMPLOYEES AS E
  16. RIGHT OUTER JOIN DEPARTMENTS AS D ON E.DEP\_ID=D.DEPT\_ID\_DEP

Copied!

Output

1. Problem:

*Re-write the previous query but have the result set include all employee names but department id and department names only for male employees.*

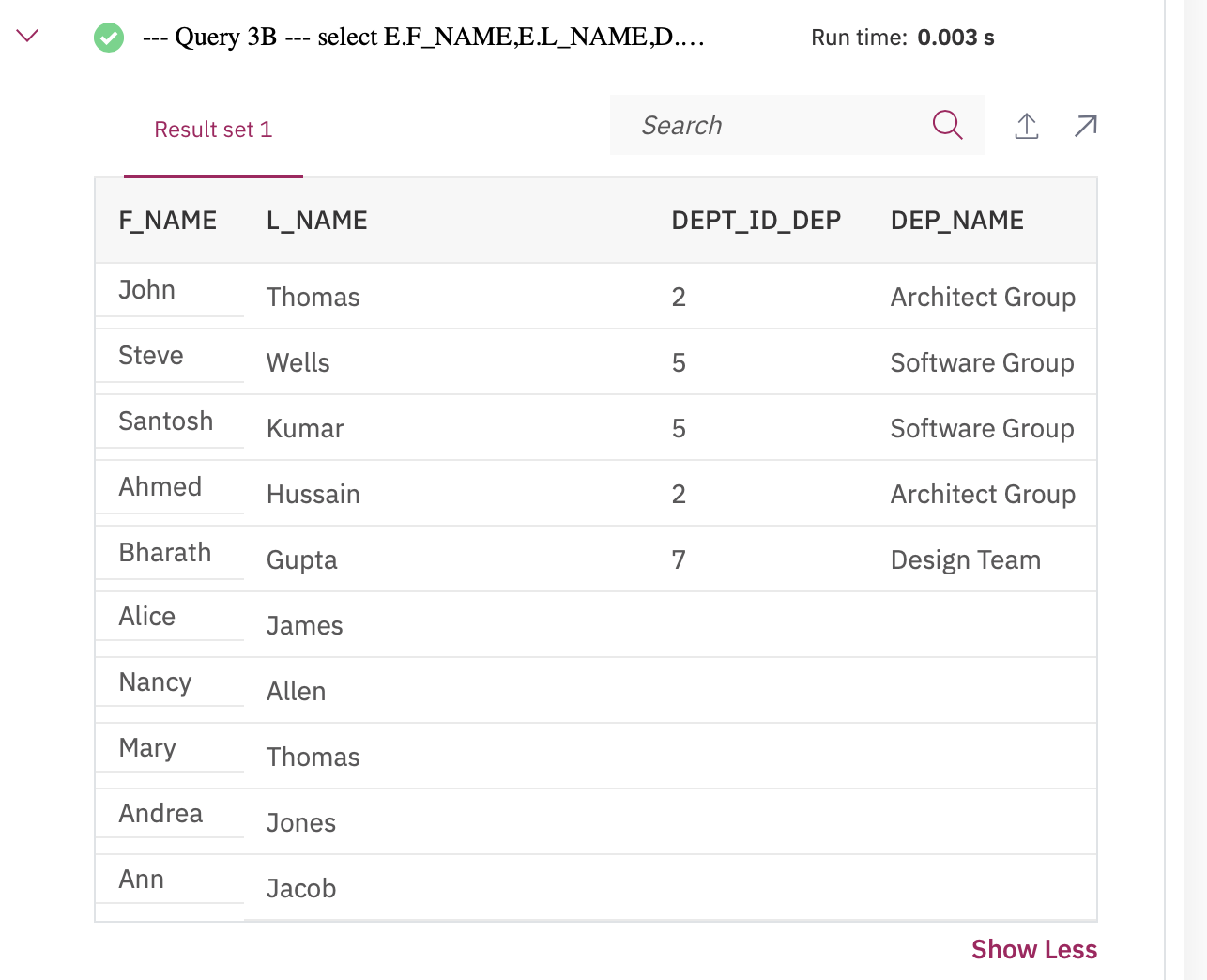
Hint

*Add an AND in Query 3A to filter on male employees in the ON clause. Alternatively, you can also use Left Outer Join.*

Solution

* 1. 1
  2. 2
  3. 3
  4. 4
  5. 5
  6. 6
  7. 7
  8. 8
  9. 9
  10. select E.F\_NAME,E.L\_NAME,D.DEPT\_ID\_DEP, D.DEP\_NAME
  11. from EMPLOYEES AS E
  12. LEFT OUTER JOIN DEPARTMENTS AS D ON E.DEP\_ID=D.DEPT\_ID\_DEP AND E.SEX = 'M'
  13. UNION
  14. select E.F\_NAME,E.L\_NAME,D.DEPT\_ID\_DEP, D.DEP\_NAME
  15. from EMPLOYEES AS E
  16. RIGHT OUTER JOIN DEPARTMENTS AS D ON E.DEP\_ID=D.DEPT\_ID\_DEP AND E.SEX = 'M';

Copied!

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. 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 import a script to mysql phpadmin interface.

* [JOIN\_Solution\_Script.sql](https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBMDeveloperSkillsNetwork-DB0201EN-SkillsNetwork/labs/MySQL/week6/JOIN_Solution_Script.sql)

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

# Author(s)

[Lakshmi Holla](https://www.linkedin.com/in/lakshmi-holla-b39062149/)

[Malika Singla](https://www.linkedin.com/in/malika-goyal-04798622/)

# Changelog

| **Date** | **Version** | **Changed by** | **Change Description** |
| --- | --- | --- | --- |
| 2023-05-05 | 0.4 | Rahul Jaideep | Updated Markdown file |
| 2022-10-28 | 0.3 | Appalabhaktula Hema | Updated image links |
| 2021-08-09 | 0.2 | Sathya Priya | Updated SQL link |
| 2021-11-01 | 0.1 | Lakshmi Holla, Malika Singla | Initial Version |

### IBM Corporation 2023. All rights reserved.