

# Hands-on Lab: Working with Multiple Tables

Estimated time needed: 30 minutes

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?

```
SELECT column_name(s)
FROM table1, table2;
```

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

```
SELECT column_name(s)
FROM table1, table2
WHERE table1.column_name = table2.column_name;
```

# Software Used in this Lab

In this lab, you will use <u>IBM Db2 Database</u>. Db2 is a Relational Database Management System (RDBMS) from IBM, designed to store, analyze and retrieve the data efficiently.

To complete this lab you will utilize a Db2 database service on IBM Cloud. If you did not already complete this lab task earlier in this module, you will not yet have access to Db2 on IBM Cloud, and you will need to follow the lab below first:

• Hands-on Lab: Sign up for IBM Cloud, Create Db2 service instance and Get started with the Db2 console

# 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:

# SAMPLE HR DATABASE TABLES

EMPLOYE	ES													
EMP_ID	F_NAME	L_NAME	SSN	B_DAT	E	SEX	ADDRESS		JOB_ID	SALAI	RY	MANAGE	R_ID	DEP_ID
E1001	John	Thomas	12345	6 1976-	01-09	М	5631 Rice, 0	DakPark,IL	100	10000	00	30001		2
E1002	Alice	James	12345	7 1972-	07-31	F	980 Berry In	, Elgin,IL	200	80000	0 :	30002		5
E1003	Steve	teve Wells		8 1980-	08-10	М	291 Springs	, Gary,IL	300	50000		30002		5
JOB_HIST	ORY					J	OBS							
EMPL_ID	START_D	START_DATE JOBS		DEPT_ID		JO	JOB_IDENT JOB_TIT		LE		MIN_	MIN_SALARY		X_SALARY
E1001	2000-01	2000-01-30 100		2		1	00	Sr. Arch	Sr. Architect		60000		100000	
E1002	2010-08	2010-08-16 20		5		2	200 Sr.Soft		wareDeveloper		60000		80000	
E1003	2016-08	2016-08-10 300		5		3	300 Jr.Softv		vareDeveloper		40000		60000	
DEPARTM	ENTS						LOCATI	ONS						
DEPT_ID_DE	P DEP_NA	DEP_NAME		MANAGER_ID			LOCT_ID		DEP	DEP_ID_LOC				
2	Architec	Architect Group		30001			L0001		2					
5	Software	Software Development		30002			L0002		5					
7	Design T	Design Team		30003			L0003		7					
_														

**NOTE:** This lab requires you to have all 5 of these tables of the HR database populated with sample data on Db2. If you didn't complete the earlier lab in this module, you won't have the tables above populated with sample data on Db2, so you will need to go through the lab below first:

• Hands-on Lab: Create tables using SQL scripts and Load data into tables

# **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 gueries with join criteria specified in the WHERE clause
- Specify aliases for table names and qualify column names with table aliases

# Instructions

When you approach the exercises in this lab, follow the instructions to run the queries on Db2:

- Go to the Resource List of IBM Cloud by logging in where you can find the Db2 service instance that you created in a previous lab under **Services** section. Click on the **Db2-xx service**. Next, open the Db2 Console by clicking on **Open Console** button. Click on the 3-bar menu icon in the top left corner and go to the **Run SQL** page. The Run SQL tool enables you to run SQL statements
  - o If needed, follow Hands-on Lab: Sign up for IBM Cloud, Create Db2 service instance and Get started with the Db2 console

# **Exercise 1: Accessing Multiple Tables with Sub-Queries**

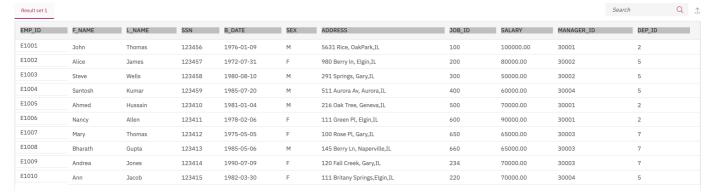
1. Problem:

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

▼ Solution

select \* from employees where JOB\_ID IN (select JOB\_IDENT from jobs);

### **▼** Output



### 2. Problem:

Retrieve only the list of employees whose JOB\_TITLE is Jr. Designer.

### **▼** Solution

select \* from employees where JOB\_ID IN (select JOB\_IDENT from jobs where JOB\_TITLE= 'Jr. Designer');

### ▼ Output



## 3. Problem:

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

## ▼ Solution

select JOB\_TITLE, MIN\_SALARY,MAX\_SALARY,JOB\_IDENT from jobs where JOB\_IDENT IN (select JOB\_ID from employees
where SALARY > 70000 );

## ▼ Output

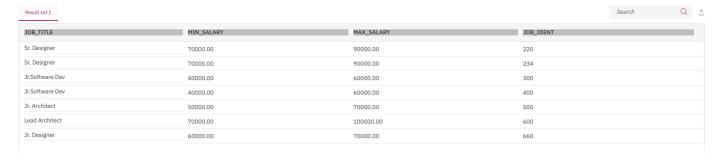


# 4. Problem:

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

## ▼ Solution

select JOB\_TITLE, MIN\_SALARY,MAX\_SALARY,JOB\_IDENT from jobs where JOB\_IDENT IN (select JOB\_ID from employees
where YEAR(B\_DATE)>1976 );



### 5. Problem:

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

# ▼ Solution

select JOB\_TITLE, MIN\_SALARY,MAX\_SALARY,JOB\_IDENT from jobs where JOB\_IDENT IN (select JOB\_ID from employees
where YEAR(B\_DATE)>1976 and SEX='F' );

# ▼ Output



# Exercise 2: Accessing Multiple Tables with Implicit Joins

1. Problem:

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

## ▼ Solution

select \* from employees, jobs;



#### 2. Problem:

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

### ▼ Solution

select \* from employees, jobs where employees.JOB\_ID = jobs.JOB\_IDENT;

## ▼ Output



### 3. Problem:

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

## ▼ Solution

select \* from employees E, jobs J where E.JOB\_ID = J.JOB\_IDENT;



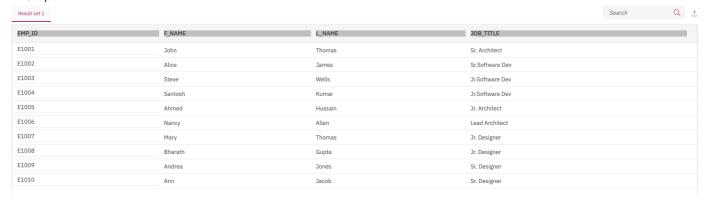
## 4. Problem:

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

### ▼ Solution

select EMP\_ID,F\_NAME,L\_NAME, JOB\_TITLE from employees E, jobs J where E.JOB\_ID = J.JOB\_IDENT;

## ▼ Output

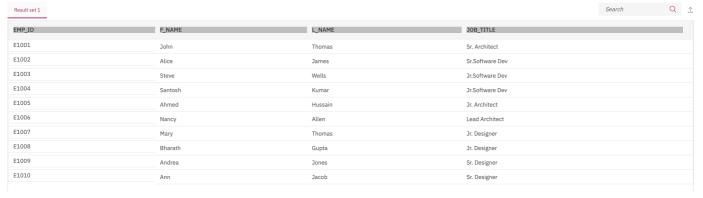


### 5. Problem:

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

# ▼ Solution

select E.EMP\_ID, E.F\_NAME, E.L\_NAME, J.JOB\_TITLE from employees E, jobs J where E.JOB\_ID = J.JOB\_IDENT;



# **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. Upload the script to the Db2 console and run. Follow <a href="Hands-on Lab">Hands-on Lab</a>: Create tables using SQL scripts and Load data into tables on how to upload a script to Db2 console and run it.

• <u>MultipleTables\_Solution\_Script.sql</u>

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

# Author(s)

- Rav Ahuja
- Sandip Saha Joy

# Other Contributor(s)

•

# Changelog

Date	Version	Changed by	Change Description
2020-12-25	2.1	Steve Ryan	ID Reviewed
2020-12-10	2.0	Sandip Saha Joy	Created revised version from DB0201EN
2020	1.0	Rav Ahuja	Created initial version

© IBM Corporation 2020. All rights reserved.