

# Hands-on Lab: String Patterns, Sorting and Grouping

Estimated time needed: 35 minutes

In this lab, you will go through some SQL practice problems that will provide hands-on experience with string patterns, sorting result sets and grouping result sets.

# Software Used in this Lab

In this lab, you will use an IBM Db2 Database. Db2 is a Relational Database Management System (RDBMS) from IBM, designed to store, analyze and retrieve 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	Œ	SEX	SEX ADDRESS		JOB_ID	SALARY		MANAGER_ID		DEP_ID
E1001	John	Thomas	12345	6 1976-	01-09	М	5631 Rice, C	akPark,IL	100	100000		30001		2
E1002	Alice	James	12345	7 1972-	07-31	F	980 Berry In	, Elgin,IL	200	80000	0	30002		5
E1003	Steve	Steve Wells		8 1980-	08-10	М	291 Springs, Gary,IL		300	50000 30002			5	
JOB_HIST	TORY					J	OBS							
EMPL_ID	START_D	TART_DATE JOBS		D DEPT_ID		JC	JOB_IDENT JOB_TIT		LE		MIN_SALARY		MAX_SALARY	
E1001	2000-01	2000-01-30 100		2		10	100 Si		Sr. Architect		60000		100000	
E1002	2010-08	2010-08-16 20		5		20	00	Sr.Softv	SoftwareDeveloper		60000		80000	
E1003	2016-08	2016-08-10 300		5		30	00 Jr.Softw		vareDeveloper		40000		60000	
DEPARTN	MENTS						LOCATIO	ONS						
DEPT_ID_D	EP DEP_NA	DEP_NAME		MANAGER_ID L			LOCT_ID		DEP	_ID_LOC				
2	Architec	Architect Group		30001 L			L0001		2	2				
5	Softwar	Software Development		30002			L0002		5	5				
7	Design 1	Design Team		30003			L0003		7	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:

- 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

# 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: 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

```
-- Query 1-----
 3 select F_NAME , L_NAME
 4 from EMPLOYEES
     where ADDRESS LIKE '%Elgin, IL%';
     --Query 2--
                 Result
 Saved scripts
Filter by status:
               Result set
                             Log
   All
Delete All
               F_NAME
∨ Al... 📋
               Alice
                                                                             James
               Nancy
                                                                             Jacob
  select ...
              Total rows: 3
  select ...
```

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

```
6 --Query 2--
     select F_NAME , L_NAME
     from EMPLOYEES
    where B_DATE LIKE '197%' ;
     ---Query3--
                  Result
 Saved scripts
 Filter by status: Result set
                              Log
   All
Delete All
                F_NAME
                                                                              L_NAME

✓ Al... 

                John
                                                                              Thomas
  select ...
                Alice
                                                                              James
                Nancy
                                                                              Allen
                Mary
                                                                              Thomas
   select ...
              Total rows: 4
```

#### 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.

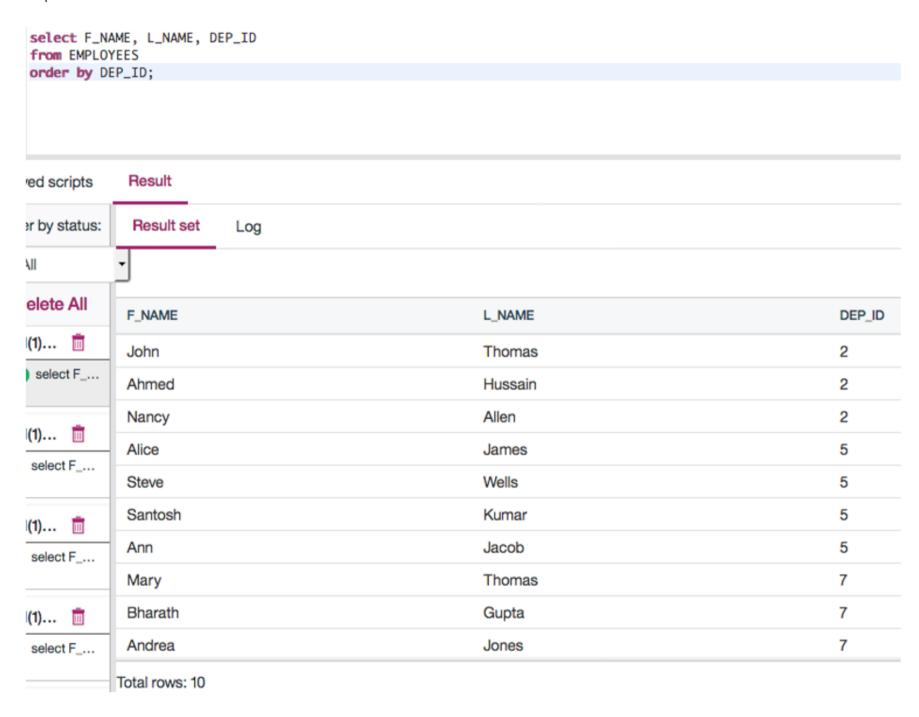
#### ▼ Hint

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



### 2. Problem:

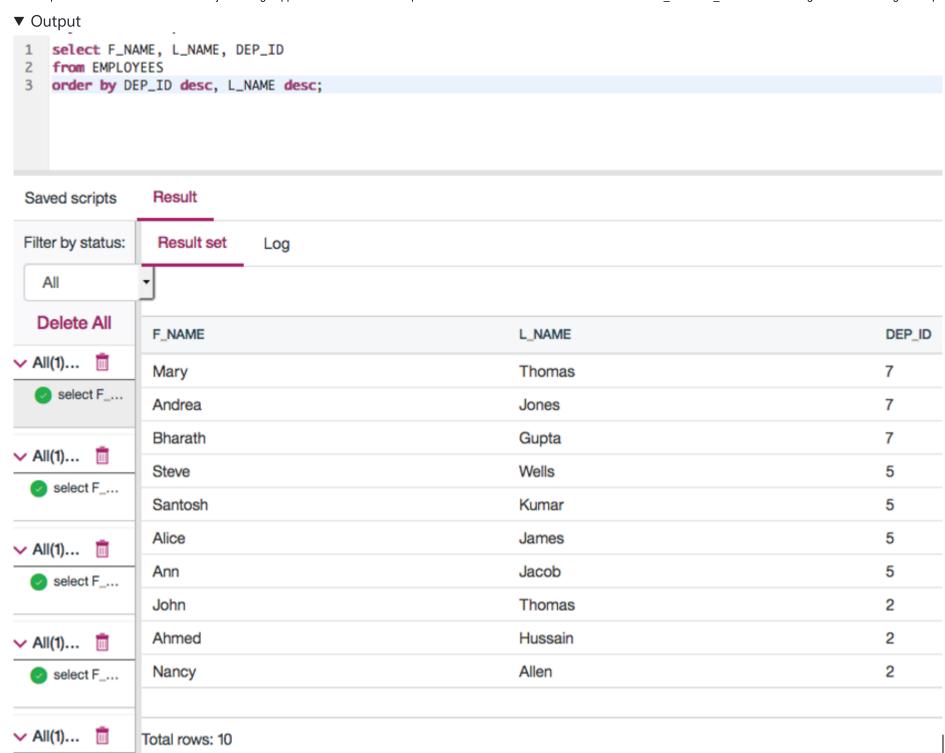
Retrieve a list of employees ordered in descending order by department ID and within each department ordered alphabetically in descending order by last name.

### **▼** 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;
```



#### 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

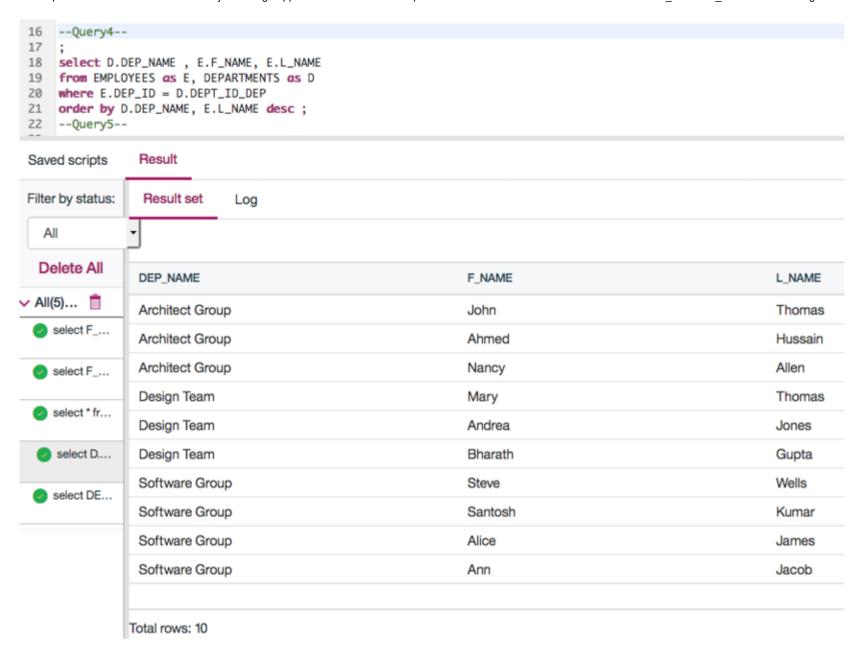
Department name is in the DEPARTMENTS table. So your query will need to retrieve data from more than one table. Don't worry if you are not able to figure this SQL problem out. We'll cover working with multiple tables in the lecture Working with Multiple Tables.

## ▼ 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



# **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.

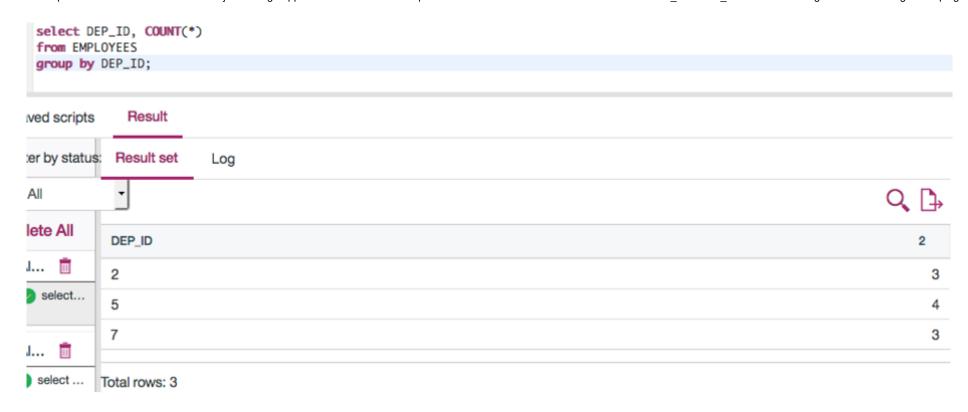
▼ Hint

Use COUNT(\*) to retrieve the total count of a column, and then GROUP BY.

**▼** Solution

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

**▼** Output



#### 2. Problem:

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

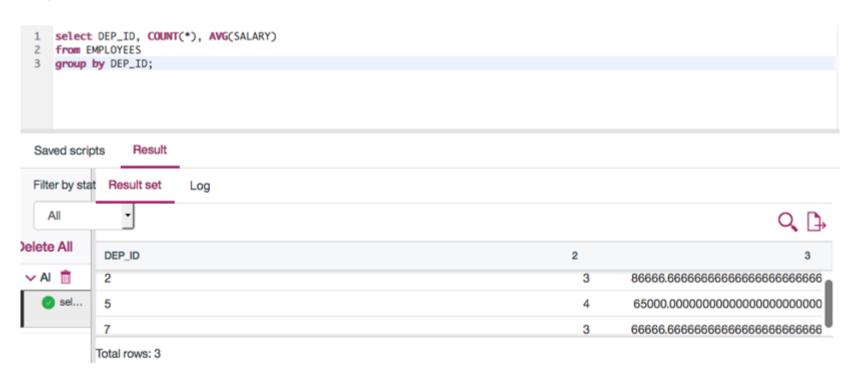
#### ▼ Hint

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 NUM\_EMPLOYEES and AVG\_SALARY.

#### **▼** 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..

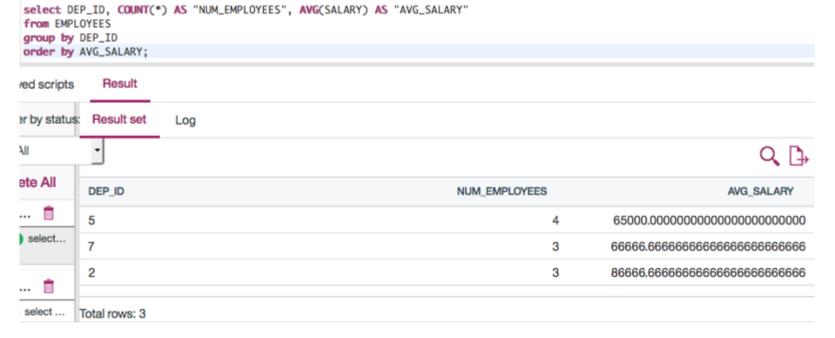
#### ▼ Hint

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.

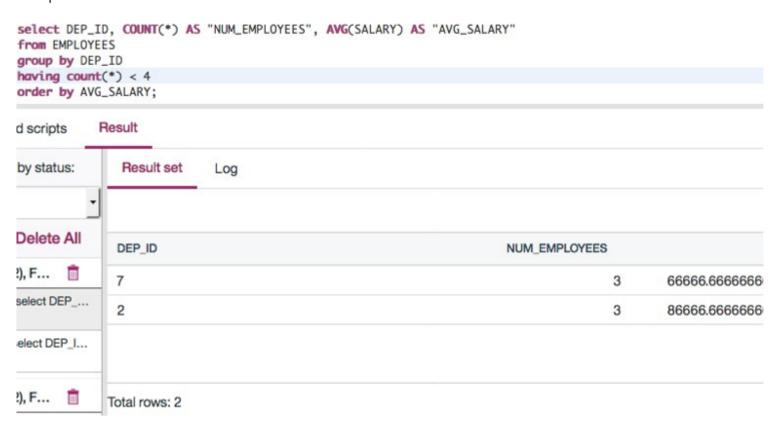
▼ Hint

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

• <u>StringPattern-Sorting-Grouping 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
2020-12-24	2.1	Steve Ryan	ID Reviewed
2020-12-08	2.0	Sandip Saha Joy	Created revised version from DB0201EN
2020	1.0	Rav Ahuja	Created initial version

© IBM Corporation 2020. All rights reserved.