Hands-on Lab : Sub-queries and Nested SELECTs

Estimated time needed: 20 minutes

In this lab, you will run through some SQL practice problems that will provide hands-on experience with nested SQL SELECT statements (also known as Sub-queries).

How does a typical Nested SELECT statement syntax look?

```
SELECT column_name [, column_name ]
FROM table1 [, table2 ]
WHERE column_name OPERATOR
   (SELECT column_name [, column_name ]
   FROM table1 [, table2 ]
   WHERE condition);
```

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

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
5	Software	30004	L0004

LOCATIONS

LOCT_ID	DEP_ID_LOC
L0001	2
L0002	5
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 demonstrate the necessity of using sub-queries
- Compose sub-queries in the where clause
- Build Column Expressions (i.e. sub-query in place of a column)
- Write Table Expressions (i.e. sub-query in place of a table)

NOTE : Make sure that you are using the CSV file and datasets from the same instruction file.

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

Exercise:

1. Problem:

> *Execute a failing query (i.e. one which gives an error) to retrieve all employees records whose salary is lower than the average salary.*

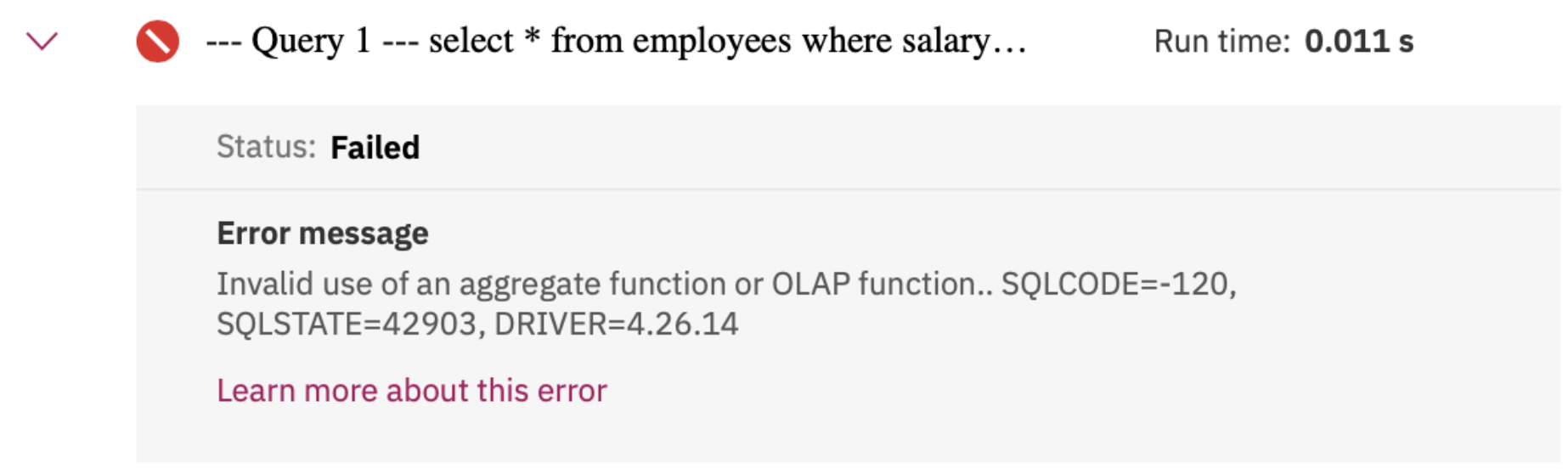
▼ Hint

> Use the AVG aggregate function.

▼ Solution

```
select *
from employees
where salary < AVG(salary);
```

▼ Output



2. Problem:

> *Execute a working query using a sub-select to retrieve all employees records whose salary is lower than the average salary.*

▼ Hint

> Put AVG(SALARY) of the inner SELECT in comparison with SALARY of the outer SELECT.

▼ Solution

```
select EMP_ID, F_NAME, L_NAME, SALARY
from employees
where SALARY < (select AVG(SALARY)
                from employees);
```

▼ Output

--- Query 2--- select EMP_ID, F_NAME, L_NAME… Run time: **0.001 s**

Result set 1 Search

EMP_ID	F_NAME	L_NAME	SALARY
E1003	Steve	Wells	50000.00
E1004	Santosh	Kumar	60000.00
E1005	Ahmed	Hussain	70000.00
E1007	Mary	Thomas	65000.00
E1008	Bharath	Gupta	65000.00
E1009	Andrea	Jones	70000.00
E1010	Ann	Jacob	70000.00

Show Less

3. Problem:

> *Execute a failing query (i.e. one which gives an error) to retrieve all employees records with EMP_ID, SALARY and maximum salary as MAX_SALARY in every row.*

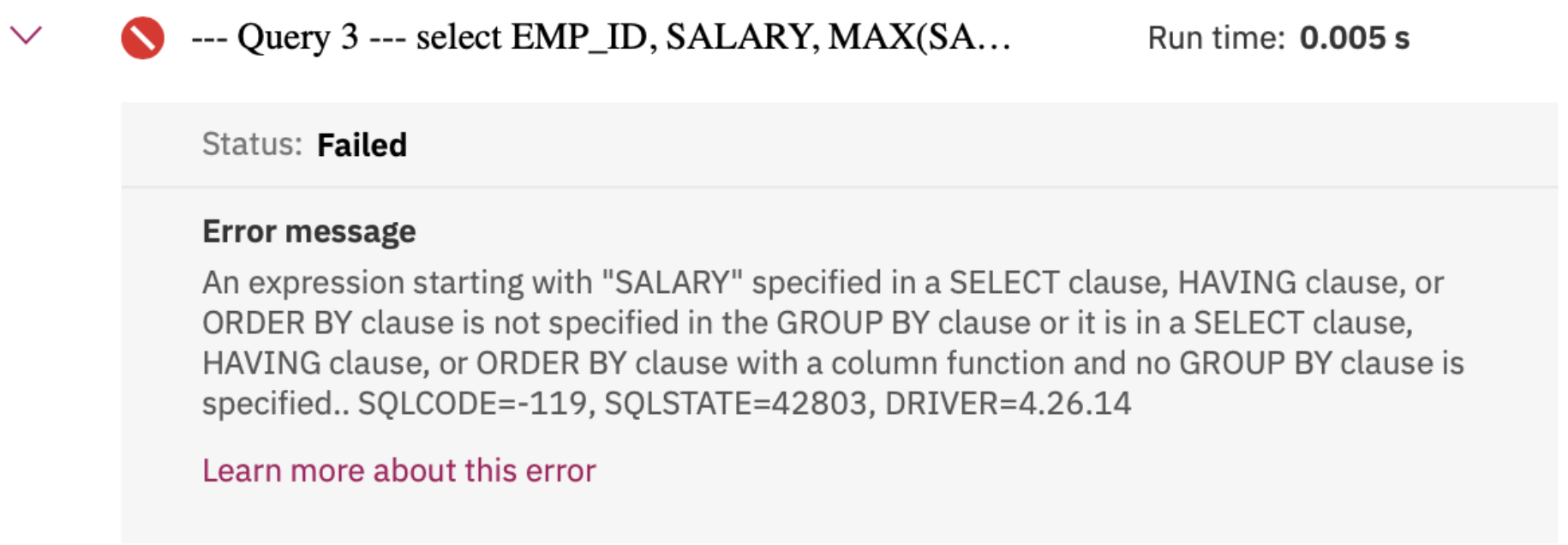
▼ Hint

> Use the MAX aggregate function.

▼ Solution

```
select EMP_ID, SALARY, MAX(SALARY) AS MAX_SALARY
from employees;
```

▼ Output



4. Problem:

> *Execute a Column Expression that retrieves all employees records with EMP_ID, SALARY and maximum salary as MAX_SALARY in every row.*

▼ Hint

> Use the SELECT (which retrieves MAX(SALARY)) as a column of the other SELECT.

▼ Solution

```
select EMP_ID, SALARY, ( select MAX(SALARY) from employees ) AS MAX_SALARY
from employees;
```

▼ Output

--- Query 4 --- select EMP_ID, SALARY, (select M… Run time: **0.001 s**

Result set 1 Search

EMP_ID	SALARY	MAX_SALARY
E1001	100000.00	100000.00
E1002	80000.00	100000.00
E1003	50000.00	100000.00
E1004	60000.00	100000.00
E1005	70000.00	100000.00
E1006	90000.00	100000.00
E1007	65000.00	100000.00
E1008	65000.00	100000.00
E1009	70000.00	100000.00
E1010	70000.00	100000.00

Show Less

5. Problem:

> *Execute a Table Expression for the EMPLOYEES table that excludes columns with sensitive employee data (i.e. does not include columns: SSN, B_DATE, SEX, ADDRESS, SALARY).*

▼ Hint

> Use a SELECT (which retrieves non-sensitive employee data) after FROM of the other SELECT.

▼ Solution

```
select * from ( select EMP_ID, F_NAME, L_NAME, DEP_ID from employees) AS EMP4ALL;
```

▼ Output

--- Query 5 --- select * from (select EMP_ID, F_N… Run time: **0.001 s**

Result set 1

Search

EMP_ID	F_NAME	L_NAME	DEP_ID
E1001	John	Thomas	2
E1002	Alice	James	5
E1003	Steve	Wells	5
E1004	Santosh	Kumar	5
E1005	Ahmed	Hussain	2
E1006	Nancy	Allen	2
E1007	Mary	Thomas	7
E1008	Bharath	Gupta	7
E1009	Andrea	Jones	7
E1010	Ann	Jacob	5

Show Less

Solution Script

If you would like to run all the solution queries of the SQL problems in this lab with a script, download the script below. Upload the script to the Db2 console and run it. 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.

- SubQueries_Solution_Script.sql

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