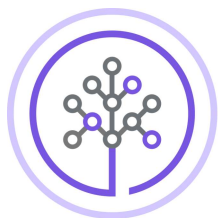


# Laboratorio práctico: Subconsultas y selecciones anidadas



**Skills**  
Network

**Tiempo estimado necesario:** 20 minutos

## Objetivos

Después de completar este laboratorio, podrás:

- Escribir consultas SQL que demuestren la necesidad de usar subconsultas
- Componer subconsultas en la cláusula where
- Construir expresiones de columna (por ejemplo, subconsulta en lugar de una columna)
- Escribir expresiones de tabla (por ejemplo, subconsulta en lugar de una tabla)

## Software utilizado en este laboratorio

En este laboratorio, utilizarás [MySQL](#). MySQL es un Sistema de Gestión de Bases de Datos Relacional (RDBMS) diseñado para almacenar, manipular y recuperar datos de manera eficiente.



Para completar este laboratorio, utilizarás el servicio de base de datos relacional MySQL disponible como parte de IBM Skills Network Labs (SN Labs) en la nube. SN Labs es un entorno de laboratorio virtual utilizado en este curso.

## Base de datos utilizada en este laboratorio

La base de datos utilizada en este laboratorio es interna. Estarás trabajando en una base de datos de recursos humanos de muestra. Este esquema de base de datos de recursos humanos consta de 5 tablas: **EMPLOYEES**, **JOB\_HISTORY**, **JOBS**, **DEPARTMENTS**, y **LOCATIONS**. Cada tabla tiene unas pocas filas de datos de muestra. El siguiente diagrama muestra las tablas de la base de datos de recursos humanos:

## 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

LOCATIONS

LOCT_ID	DEP_ID_LOC
L0001	2
L0002	5
L0003	7

## Cargar la base de datos

Usando las habilidades adquiridas en los módulos anteriores, primero debes crear la base de datos en MySQL. Sigue los pasos a continuación.

1. Abre la interfaz de phpMyAdmin desde el Skills Network Toolbox en Cloud IDE.
2. Crea una base de datos en blanco llamada 'HR'. Utiliza el script compartido en el enlace a continuación para crear las tablas necesarias.  
[Script\\_Create\\_Tables.sql](#)
3. Descarga los archivos en los enlaces a continuación a tu dispositivo local (si no lo has hecho ya en laboratorios anteriores)  
[Departments.csv](#)  
[Jobs.csv](#)  
[JobsHistory.csv](#)  
[Locations.csv](#)  
[Employees.csv](#)
4. Utiliza estos archivos en la interfaz de phpMyAdmin como los datos para las respectivas tablas en la base de datos 'HR'.

## Subconsultas y Selecciones Anidadas

Supongamos que se te pide recuperar todos los registros de empleados cuyo salario es inferior al salario promedio. Podrías utilizar la siguiente consulta para hacerlo.

```
SELECT *
FROM EMPLOYEES
```

```
WHERE salary < AVG(salary);
```

Sin embargo, esta consulta generará un error que indica: “Uso ilegal de la función de grupo”. Aquí, la función de grupo es AVG y no se puede usar directamente en la condición ya que no se ha recuperado de los datos. Por lo tanto, la condición utilizará una subconsulta para recuperar la información del salario promedio para comparar con el salario existente. La consulta modificada se convertiría en:

```
SELECT *  
FROM EMPLOYEES  
WHERE SALARY < (SELECT AVG(SALARY) FROM EMPLOYEES);
```

Ahora, considere ejecutar una consulta que recupere todos los registros de empleados con EMP\_ID, SALARY y el salario máximo como MAX\_SALARY en cada fila. Para esto, el salario máximo debe ser consultado y utilizado como una de las columnas. Esto se puede hacer utilizando la consulta a continuación.

```
SELECT EMP_ID, SALARY, (SELECT MAX(SALARY) FROM EMPLOYEES) AS MAX_SALARY  
FROM EMPLOYEES;
```

Ahora, considera que deseas extraer los nombres y apellidos del empleado más antiguo. Dado que el empleado más antiguo será el que tenga la fecha de nacimiento más pequeña, la consulta se puede escribir como:

```
SELECT F_NAME, L_NAME  
FROM EMPLOYEES  
WHERE B_DATE = (SELECT MIN(B_DATE) FROM EMPLOYEES);
```

También puedes usar subconsultas para crear tablas derivadas, que luego se pueden usar para consultar información específica. Supongamos que quieres conocer el salario promedio de los 5 principales ganadores de la empresa. Primero tendrás que extraer una tabla con los cinco salarios más altos. A partir de esa tabla, puedes consultar el valor promedio del salario. La consulta se puede escribir de la siguiente manera.

```
SELECT AVG(SALARY)
FROM (SELECT SALARY
      FROM EMPLOYEES
      ORDER BY SALARY DESC
      LIMIT 5) AS SALARY_TABLE;
```

Tenga en cuenta que es necesario dar un alias a cualquier tabla derivada.

## Problemas de Práctica

1. Escribe una consulta para encontrar el salario promedio de los cinco empleados con menor salario.

▼ Haz clic aquí para una pista

Necesitas ordenar los datos en orden ascendente de salario y limitarlo a las cinco entradas principales, tratando esto como una tabla derivada. Toma el promedio de estas entradas.

▼ Haz clic aquí para la solución

```
SELECT AVG(SALARY)
FROM (SELECT SALARY
      FROM EMPLOYEES
      ORDER BY SALARY
      LIMIT 5) AS SALARY_TABLE;
```

2. Escribe una consulta para encontrar los registros de empleados mayores que la edad promedio de todos los empleados.

▼ Haz clic aquí para una pista

La edad en años se puede calcular como el componente del año en la diferencia entre la fecha de nacimiento y la fecha actual. Necesitas comparar la edad en años con la edad promedio en años. La edad promedio en años se evaluará como una subconsulta.

▼ Haz clic aquí para la solución

```
SELECT *  
FROM EMPLOYEES  
WHERE YEAR(FROM_DAYS(DATEDIFF(CURRENT_DATE, B_DATE))) >  
      (SELECT AVG(YEAR(FROM_DAYS(DATEDIFF(CURRENT_DATE, B_DATE))))  
       FROM EMPLOYEES);
```

3. Desde la tabla Job\_History, muestra la lista de IDs de empleados, años de servicio y promedio de años de servicio para todas las entradas.

▼ Haz clic aquí para obtener una pista

Para esto, necesitas calcular los años de servicio como la diferencia entre la fecha de ingreso y la fecha actual. El promedio de años de servicio debe consultarse por separado para ser mostrado.

▼ Haz clic aquí para la solución

```
SELECT EMPL_ID, YEAR(FROM_DAYS(DATEDIFF(CURRENT_DATE, START_DATE))),  
      (SELECT AVG(YEAR(FROM_DAYS(DATEDIFF(CURRENT_DATE, START_DATE))))  
       FROM JOB_HISTORY)  
FROM JOB_HISTORY;
```

## Conclusión

¡Felicidades! Has completado este laboratorio y estás listo para el siguiente tema.

Ahora deberías ser capaz de:

- Escribir consultas SQL que demuestren la necesidad de utilizar subconsultas
- Componer subconsultas en la cláusula WHERE
- Construir Expresiones de Columna (por ejemplo, subconsulta en lugar de una columna)
- Escribir Expresiones de Tabla (por ejemplo, subconsulta en lugar de una tabla)

## Author(s)

[Abhishek Gagneja](#)

[Lakshmi Holla](#)

[Malika Singla](#)

**© IBM Corporation 2023. Todos los derechos reservados.**