



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

TAREA 14

Lilian De la Peña Osorio - Grupo 1

SUBCONSULTAS

1. Definición

Las subconsultas son consultas SQL que se encuentran dentro de otras consultas, y a menudo se les llama "consultas anidadas" o "consultas internas". Permiten usar el resultado de una consulta como entrada para otra, lo que ayuda a realizar operaciones complejas al filtrar registros, agregar datos o cruzar información entre tablas sin necesidad de JOIN.

Una subconsulta es una expresión select encerrada entre paréntesis como un bloque de consulta anidado en una sentencia de columna.

Puede utilizar estos bloques de consulta anidados en cualquiera de las siguientes sentencias SQL:

*SELECT
INSERT
DELETE
UPDATE
CREATE TABLE AS
INSERT INTO
SELECT INTO*

Puede anidar subconsultas a cualquier profundidad arbitraria.

La consulta padre que contiene la subconsulta se denomina a menudo super consulta o consulta externa. Las subconsultas del mismo parentesco se utilizan para derivar conjuntos de resultados que se pueden evaluar junto con la consulta padre.

2. ¿Dónde puedo usarlas y bajo qué condiciones?

Las subconsultas en SQL se utilizan para realizar operaciones que requieren múltiples pasos o lógica compleja, anidando una consulta (interna) dentro de otra



(externa). Son herramientas versátiles que ayudan a filtrar registros, agregar datos y realizar cálculos dinámicamente.

Dónde usarlas

Las subconsultas se pueden emplear en varias cláusulas de sentencias SQL:

- **SELECT:** Para devolver un solo valor (subconsulta escalar) en la lista de campos seleccionados, por ejemplo, para mostrar el promedio global junto a un valor individual.
- **WHERE / HAVING:** Para filtrar registros basándose en el resultado de otra consulta. Se usan con operadores de comparación (=, <, >, etc.) si devuelven un solo valor, o con IN, ANY, ALL, EXISTS si devuelven múltiples filas.
- **FROM / JOIN:** El resultado de una subconsulta puede tratarse como una tabla temporal, lo que permite unirla con otras tablas o usarla para operaciones adicionales.
- **INSERT / UPDATE / DELETE:** Para especificar los datos a insertar, los registros a actualizar o los registros a eliminar, basándose en criterios dinámicos obtenidos de otras tablas.

Bajo qué condiciones se usan

Las condiciones y reglas clave para su uso son:

- **Claridad y Simplicidad Relativa:** Son útiles para consultas simples y puntuales donde la lógica se puede integrar fácilmente. A menudo son más fáciles de entender para los principiantes que un JOIN para ciertas tareas.
- **Resultados Específicos:** Se usan cuando se necesita que una consulta interna devuelva un resultado (valor, fila o conjunto de filas) que la consulta externa pueda utilizar para su procesamiento.
- **Restricciones de Devolución:**
 - Si se usan en la lista de SELECT o con operadores de comparación simple (=, >, etc.) en WHERE/HAVING, deben devolver un solo valor (escalar).
 - Si se usan con IN, ANY, ALL, deben devolver una sola columna, pero pueden ser múltiples filas.

Limitaciones Estructurales:

- Generalmente, una subconsulta no puede contener una cláusula ORDER BY (salvo en algunos SGBD y casos específicos, pero no es estándar).
- No pueden contener cláusulas BETWEEN o LIKE directamente en su definición para filtrar la subconsulta en sí misma.

3. Subconsultas correlacionadas

Una subconsulta correlacionada es una consulta anidada que depende de la consulta externa, ejecutándose una vez por cada fila de la consulta externa. Esto significa que la subconsulta utiliza valores de la fila actual de la consulta principal para ejecutarse, y es útil para filtrar datos complejos como buscar empleados con salario superior al promedio de su departamento o autores que venden por encima del promedio de su categoría. A diferencia de las subconsultas simples, que se ejecutan una sola vez, las correlacionadas pueden ser más lentas debido a esta ejecución repetida.

¿Cómo funcionan?

- Dependencia: La subconsulta hace referencia a una columna de la consulta externa.
- Ejecución: La consulta externa se ejecuta primero, y para cada fila que devuelve, la subconsulta se ejecuta utilizando el valor de esa fila.
- Ejemplo:
- Consulta externa: `SELECT nombre, salario FROM empleados`
- Consulta interna: Para cada empleado, la subconsulta calcula el salario promedio de ese empleado's departamento: `SELECT AVG(salario) FROM empleados WHERE departamento_id = empleados.departamento_id;`
- Correlación: El `empleados.departamento_id` en la subconsulta hace referencia a la columna de la tabla en la consulta externa.

4. Ejemplos

Tablas usadas:

Hoja de Trabajo	Generador de Consultas
	<pre> create table trabajadores(id number(5) not null, nombre varchar(20), apellido varchar(20)); insert into trabajadores values(10311,'Adriana','Abreu'); insert into trabajadores values(10312,'Eduardo','Galicia'); insert into trabajadores values(10313,'Izquierdo','Alvarado'); insert into trabajadores values(10314,'Jaqueline','Piña'); insert into trabajadores values(10315,'Dafne','Santos'); create table alumnos(id number(5) not null, nombre varchar(20), apellido varchar(20)); insert into alumnos values(10311,'Adriana','Abreu'); insert into alumnos values(10316,'Juan','Perez'); insert into alumnos values(10313,'Izquierdo','Alvarado'); insert into alumnos values(10315,'Escorpion','Dorado'); </pre>



```
select * from trabajadores;
```

Salida de Script | Resultado de la Consulta | SQL | Todas las Filas Recuperadas:

ID	NOMBRE	APELLIDO
1	10311 Adriana	Abreu
2	10312 Eduardo	Galicia
3	10313 Izquierdo	Alvarado
4	10314 Jaqueline	Piña
5	10315 Dafne	Santos

```
select * from alumnos;
```

Salida de Script | Resultado de la Consulta | SQL | Todas las Filas Recuperadas:

ID	NOMBRE	APELLIDO
1	10311 Adriana	Abreu
2	10316 Juan	Perez
3	10313 Izquierdo	Alvarado
4	10315 Escorpion	Dorado

- SUBCONSULTAS

1)

```
--Alumnos que también son trabajadores
SELECT nombre, apellido
FROM alumnos
WHERE id IN (SELECT id FROM trabajadores);
```

Salida de Script | Resultado de la Consulta | SQL | Todas las Filas Recuperadas: 3 en 0.017 segundos

NOMBRE	APELLIDO
1 Adriana	Abreu
2 Izquierdo	Alvarado
3 Escorpion	Dorado

La subconsulta obtiene los id de todos los trabajadores.

La consulta principal muestra los alumnos cuyo id aparece también en la lista.

2)

```
--Trabajadores que NO son alumnos
SELECT nombre, apellido
FROM trabajadores
WHERE id NOT IN (SELECT id FROM alumnos);
```

Salida de Script | Resultado de la Consulta | SQL | Todas las Filas Recuperadas: 2 en 0.012 segundos

NOMBRE	APELLIDO
1 Jaqueline	Piña
2 Eduardo	Galicia

Busca los trabajadores cuyo id no está en la tabla de alumnos.

3)

```
--Mostrar el alumno cuyo ID es el mayor de todos los trabajadores
SELECT nombre, apellido
FROM alumnos
WHERE id = (SELECT MAX(id) FROM trabajadores);
```

Salida de Script | Resultado de la Consulta | Resultado de la Consulta 1 | Resultado de la Consulta 2

SQL | Todas las Filas Recuperadas: 1 en 0.019 segundos

	NOMBRE	APELLIDO
1	Escorpion	Dorado

La subconsulta obtiene el mayor ID de trabajadores (10315).

La principal busca el alumno con ese mismo ID.

SUBCONSULTAS CORRELACIONADAS

1)

```
--SUBCONSULTAS CORRELACIONADAS
--Alumnos que tienen un homónimo (mismo nombre y apellido) entre los trabajadores
SELECT a.nombre, a.apellido
FROM alumnos a
WHERE EXISTS (
    SELECT 1
    FROM trabajadores t
    WHERE t.nombre = a.nombre
    AND t.apellido = a.apellido
);
```

Salida de Script | Resultado de la Consulta | Resultado de la Consulta 1 | Resultado de la Consulta 2

SQL | Todas las Filas Recuperadas: 2 en 0.058 segundos

	NOMBRE	APELLIDO
1	Adriana	Abreu
2	Izquierdo	Alvarado

Por cada alumno, Oracle busca en trabajadores si hay alguien con el mismo nombre y apellido.

Si lo encuentra, se incluye en el resultado.

2)

```
--Trabajadores que también están registrados como alumnos
SELECT t.nombre, t.apellido
FROM trabajadores t
WHERE EXISTS (
    SELECT 1
    FROM alumnos a
    WHERE a.id = t.id
);
```

Salida de Script | Resultado de la Consulta | Resultado de la Consulta 1 | Resultado de la Consulta 2

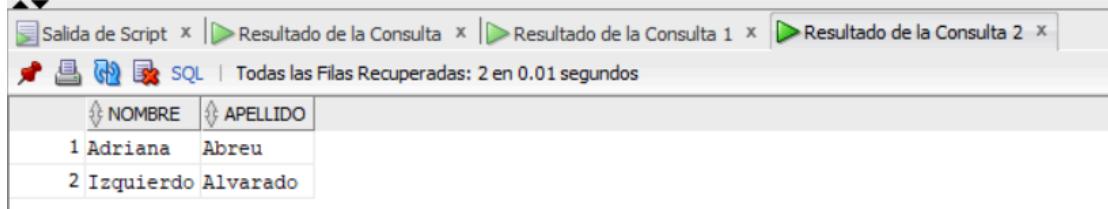
SQL | Todas las Filas Recuperadas: 3 en 0.009 segundos

	NOMBRE	APELLIDO
1	Adriana	Abreu
2	Izquierdo	Alvarado
3	Dafne	Santos

Por cada trabajador, Oracle revisa si existe un alumno con el mismo id.
Si sí, se muestra.

3)

```
--Alumnos cuyo apellido también aparece en la tabla de trabajadores
SELECT a.nombre, a.apellido
FROM alumnos a
WHERE EXISTS (
    SELECT 1
    FROM trabajadores t
    WHERE t.apellido = a.apellido
);
```



Salida de Script | Resultado de la Consulta | Resultado de la Consulta 1 | Resultado de la Consulta 2

SQL | Todas las Filas Recuperadas: 2 en 0.01 segundos

	NOMBRE	APELLIDO
1	Adriana	Abreu
2	Izquierdo	Alvarado

La subconsulta revisa si el apellido del alumno existe en la tabla de trabajadores.
Si sí, se muestra el alumno.