

5. Subconsultas

5.1. Objetivos

Los objetivos de esta actividad son:

- Realizar consultas que contienen a otras consultas como parte de la sentencia SELECT.

5.2. Actividad

5.2.1. Subconsultas

Una subconsulta es una sentencia SELECT contenida en otra sentencia del lenguaje de manipulación de datos. Los resultados de la subconsulta afectan en el resultado de la sentencia que la contiene. Se puede decir que una subconsulta está anidada en la consulta externa o consulta principal y es posible anidar subconsultas dentro de otras subconsultas hasta una profundidad considerable.

En esta actividad se utilizarán las subconsultas como parte de una sentencia SELECT para resolver consultas complejas.

Un ejemplo:

```
SELECT * FROM t1 WHERE column1 = (SELECT column1 FROM t2);
```

- La sentencia SELECT column1 FROM t2 sería la subconsulta.
- La sentencia SELECT * FROM t1 WHERE column1 = sería la consulta principal.

Algunas características de las subconsultas:

- Aparecen cerradas entre paréntesis, dentro de la consulta principal.
- Pueden retornar un escalar (un valor único), una tabla formada por una o varias columnas y una o varias filas o el valor NULL (si no existe ninguna fila que cumpla las condiciones de la subconsulta).
- Los nombres de columna que aparecen en una subconsulta pueden hacer referencia a tablas de la consulta principal. A estas referencias se llama 'referencias externas' en la subconsulta.
- Se pueden utilizar en cualquier parte de la sentencia SELECT, aunque lo más habitual es que formen parte de una condición en la cláusula WHERE o en la cláusula HAVING.

Ventajas del uso de subconsultas:

- Permite escribir consultas estructuradas de forma que es posible aislar cada parte de la consulta, y descomponer un problema grande en varios pequeños.
- Proporcionan una manera alternativa de resolver consultas que de otro modo necesitarían combinaciones de tablas (JOIN) o uniones complejas.
- Hacen que las consultas sean más fáciles de leer.

5.2.2. Subconsultas como parte de una condición

La forma más habitual de utilización de subconsultas es formando parte de una condición dentro de las cláusulas WHERE o HAVING. La forma de utilizar las subconsultas y los operadores que se pueden utilizar en la sentencia principal para hacer comparaciones con el resultado del de la subconsulta, depende de los valores que devuelve la subconsulta, que pueden ser valores escalares o tablas.

5.2.2.1. Utilizando operadores relacionales

Subconsultas que devuelven un escalar

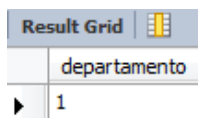
En su forma más simple, una subconsulta devuelve un solo valor y puede ser utilizada como operando en cualquier expresión (como si fuera un literal o una columna) o como parámetro de una función. Normalmente se compara el valor que devuelve la subconsulta con una expresión mediante un operador relacional.

Por ejemplo: mostrar *dni*, *apellidos* y *nombre* de los empleados del mismo departamento que el empleado que tiene el DNI '12852654'.

- Paso 1: conocer el departamento al que pertenece ese empleado. Eso se puede saber ejecutando la consulta:

```
select cod_departamento
from empleado
where dni = '12852654';
```

Resultado de la ejecución:

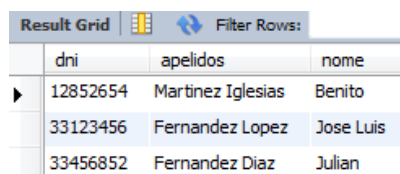


departamento
1

- Paso 2: la consulta anterior devuelve un valor escalar y puede ser utilizada como subconsulta en la sentencia SELECT que permite mostrar los datos solicitados de los empleados que trabajan en ese departamento:

```
select dni, apellidos, nombre
from empleado
where cod_departamento = (select cod_departamento
                           from empleado
                           where dni = '12852654');
```

Resultado de la ejecución:



dni	apellidos	nombre
12852654	Martinez Iglesias	Benito
33123456	Fernandez Lopez	Jose Luis
33456852	Fernandez Diaz	Julian

Subconsultas que devuelven una tabla

Las subconsultas pueden devolver una tabla con una fila al menos. Para la comparación con la subconsulta hay que utilizar los operadores relacionales combinados con las palabras ALL o ANY.

- Cuando se utiliza la palabra ALL delante del operador relacional, la condición será verdadera si es verdadera para todos los valores que devuelve la subconsulta.
- Cuando se utiliza la palabra ANY ante el operador relacional, la condición es verdadera si es verdadera para al menos uno de los valores que devuelve la subconsulta.

Ejemplo con ALL: mostrar el *dni*, *apellidos*, *nombre* y *salario_bruto* de los empleados que tengan un salario bruto superior al de todos los empleados que trabajan en el departamento 4.

- Paso 1: seleccionar los salarios de todos los empleados del Departamento 4:

```
select salario_bruto
from empleado
where cod_departamento = 4
```


Resultado de la ejecución:

Result Grid
salario_bruto
35520.00
25000.00
58500.00

- Paso 2: la consulta anterior devuelve una tabla que tiene varias filas, y una columna con los salarios de todos los empleados del departamento 4. La sentencia SELECT a construir tiene que comparar el salario de cada empleado con todos los valores que devuelve la subconsulta, y seleccionar solo los que tienen un salario bruto mayor que todos esos valores.

```
select dni, apellidos, nombre, salario_bruto
from empleado
where salario_bruto > ALL (select salario_bruto
                           from empleado
                           where cod_departamento = 4);
```

Resultado de la ejecución:

Result Grid		 Filter Rows:	Edit:	
	dni	apellidos	nome	salario_bruto
▶	33123456	Fernandez Lopez	Jose Luis	160000.00
	33456852	Fernandez Diaz	Julian	85400.00
	33963258	Bernardez Macia	Luisa	65200.00
	58745124	Cendan Villa	Lorenzo	65350.00

Una solución alternativa para resolver esta consulta consiste en calcular el salario bruto más alto de los empleados del Departamento 4, que sería un valor escalar, y después comparar el salario bruto de cada empleado con ese valor escalar utilizando un operador relacional.

- Paso 1: calcular el salario más alto de los empleados del Departamento 4.

```
select max(salario_bruto)
from empleado
where cod_departamento = 4
```

Resultado de la ejecución:

Result Grid	
	max(salario_bruto)
▶	58500.00

- Paso 2: el resultado de la consulta es un valor escalar y se puede comparar con el operador relacional 'mayor que' (>).

```
select dni, apellidos, nombre, salario_bruto
from empleado
where salario_bruto > (select max(salario_bruto)
                        from empleado
                        where cod_departamento = 4);
```

Ejemplo con ANY: mostrar el *dni*, *apellidos*, *nombre*, *salario_bruto* y *departamento* de los empleados que trabajan en un departamento de la provincia de Lugo (*id_provincia* 27).

- Paso 1: Seleccionamos los códigos de los departamentos que están situados en la provincia de Lugo.

```
select codigo
from departamento
where id_provincia = '27';
```

Resultado de la ejecución:

Result Grid	
	codigo
▶	1
	2
	6
	8

- Paso 2: en la sentencia principal hay que comparar los departamentos de cada empleado de los valores que devuelve la subconsulta, y seleccionar sólo aquellos que tengan un departamento igual a alguno de los valores que devuelve la subconsulta.

```

select dni, apellidos, nombre, salario_bruto, cod_departamento
from empleado
where cod_departamento = ANY (select codigo
                                from departamento
                                where id_provincia = '27');

```

Resultado de la ejecución:

The screenshot shows a database application interface. At the top, there's a 'Result Grid' with columns: dni, apellidos, nome, salario_bruto, departamento. It lists 7 rows of employee data. Below the grid, there's a status bar indicating 'empleado 33' and '7 row(s) returned'. At the bottom, there's an 'Output' section with a table showing the execution time (12:42:13) and the action (select dni, apellidos, nome, salario_bruto, departame...). The output table has columns: Time, Action, Message, and Duration / Fetch.

dni	apellidos	nome	salario_bruto	departamento
12852654	Martinez Iglesias	Benito	25000.00	1
33123456	Fernandez Lopez	Jose Luis	160000.00	1
33456852	Fernandez Diaz	Julian	85400.00	1
15245258	Nuñez Bernardes	Antonia	42000.00	2
33251256	Martinez Diaz	Carlos	31500.00	2
33558994	Abelleira Carrion	Dorinda	19500.00	6
33963258	Bernardes Macia	Luisa	65200.00	8

5.2.2.2. Utilizando el predicado BETWEEN

Se puede utilizar el predicado BETWEEN con subconsultas que devuelven un escalar. Sirve para comparar una expresión con un intervalo de valores que se obtienen de una o dos subconsultas.

Por ejemplo: mostrar el *dni*, *apellidos*, *nombre*, *departamento* y *salario_bruto* de los empleados que tengan un salario bruto entre el salario más alto del departamento 2 y el salario más alto del departamento 8.

- Paso 1: calcular el salario más alto de los empleados del Departamento 2.

```

select max(salario_bruto)
from empleado
where cod_departamento = 2;

```

The screenshot shows a 'Result Grid' with one column: max(salario_bruto). It contains one row with the value 42000.00.

max(salario_bruto)
42000.00

- Paso 2: calcular el salario más alto de los empleados del departamento 8.

```

select max(salario_bruto)
from empleado
where cod_departamento = 8;

```

The screenshot shows a 'Result Grid' with one column: max(salario_bruto). It contains one row with the value 65200.00.

max(salario_bruto)
65200.00

- Paso 3: subconsulta final utilizando las dos anteriores.

```

select dni, apellidos, nombre, cod_departamento, salario_bruto
from empleado
where salario_bruto between (select max(salario_bruto)
                             from empleado
                             where cod_departamento = 2)
and (select max(salario_bruto)
     from empleado
     where cod_departamento = 8);

```

Result Grid					
Filter Rows:					
	dni	apellidos	nombre	departamento	salario_bruto
▶	15245258	Núñez Bernardez	Antonia	2	42000.00
	33254916	Iglesias Dominguez	Adolfo	NULL	52500.00
	33963258	Bernardez Macia	Luisa	8	65200.00
	33987456	Porto Novo	Begoña	9	52000.00
	54528788	Canedo Tellez	Angeles	4	58500.00
*	NULL	NULL	NULL	NULL	NULL

5.2.2.3. Utilizando el predicado IN

El predicado IN se utiliza cuando la subconsulta devuelve una tabla con una o más filas y permite comparar el valor que corresponde a una expresión con los valores que devuelve la subconsulta. Si de antemano se sabe que la subconsulta va a devolver una tabla de una fila, no es recomendable utilizar este predicado y deberíamos utilizar operadores relacionales.

- La condición es verdadera cuando el valor que representa la expresión es igual a alguno de los valores que devuelve la subconsulta.
- Cuando se utiliza el operador NOT delante, la condición es verdadera cuando el valor que representa la expresión es distinto de todos los valores que devuelve la subconsulta. En el caso que en la lista de valores esté incluido el valor NULL hay que tener en cuenta que la comparación de cualquier valor con el valor NULL devuelve NULL y no *true* o *false*, lo que puede provocar problemas y obtener resultados que no son correctos.
- Cuando comparamos el valor de una expresión con los valores que devuelve una subconsulta que produce como resultado una tabla de más de una fila:
 - IN equivale a una condición = ANY
 - NOT IN equivale a una condición != ALL

Por ejemplo: mostrar el *dni*, *apellidos*, *nombre*, *salario_bruto* y *departamento* de los empleados que trabajan en algún departamento que no esté ubicado en la provincia de Lugo.

- Paso 1: seleccionar los códigos de los departamentos que están situados en la provincia de Lugo.

```

select codigo
from departamento
where id_provincia = '27';

```

Resultado de la ejecución:

Result Grid	
	codigo
▶	1
	2
	6
	8

- Paso 2: en la sentencia principal hay que comparar el departamento de cada empleado con los valores que devuelve la subconsulta, y seleccionar sólo aquellos que tengan un departamento distinto a alguno de los valores que devuelve la subconsulta.

```
select dni, apellidos, nombre, salario_bruto, cod_departamento
from empleado
where cod_departamento NOT IN (select codigo
                                from departamento
                                where id_provincia = '27');
```

dni	apellidos	nome	salario_bruto	departamento
12549563	Case Rodriguez	Fernanda	35520.00	4
33147258	Ruiz Macias	Dario	32000.00	10
33219853	Hernandez Valin	Valentina	26000.00	3
33257964	Vila Bernal	Rosario	25000.00	4
33322541	Villar Bernal	Teolindo	17500.00	5
33365846	Quiroga Juarez	Francisco	36500.00	3
33625566	Garcia Perez	Adrian	21500.00	5
33845215	Sanchez Rodriguez	Maria	35400.00	3
33891564	Aguar Lopez	Luis	35000.00	9
33987456	Porto Novo	Begoña	52000.00	9
54528788	Canedo Tellez	Angeles	58500.00	4

5.2.2.4. Utilizando el predicado EXISTS

Permite comprobar si la subconsulta devuelve alguna fila o no. Los valores que devuelve la subconsulta no se comparan con ningún otro valor.

- La condición es verdadera si la subconsulta devuelve una o más filas, y es falsa si la subconsulta no devuelve ninguna fila.

Para que este tipo de condición tenga utilidad, la subconsulta debe contener alguna condición que devuelva resultados diferentes para cada fila de la tabla de la sentencia principal. Esto supone que en alguna condición de la subconsulta tiene que intervenir alguna columna de las tablas de la consulta principal ('referencias externas'). Si esto no fuese así, la condición sería verdadera o falsa para todas las filas de la consulta principal.

Por ejemplo: mostrar *dni*, *apellidos*, *nombre*, *salario_bruto* y *departamento* de los compañeros de departamento de los empleados con DNI 33456345, que tengan un salario mayor que el suyo.

```
select dni, apellidos, nombre, salario_bruto, cod_departamento
from empleado as emp1
where exists (select emp2.cod_departamento
              from empleado as emp2
              where emp2.dni = '33845215'
              and emp1.cod_departamento = emp2.cod_departamento
              and emp1.salario_bruto > emp2.salario_bruto);
```

Result Grid		Filter Rows:		Edit:	
dni	apellidos	nome	salario_bruto	departamento	
33365846	Quiroga Juarez	Francisco	36500.00	3	
*	NULL	NULL	NULL	NULL	

5.2.3.Subconsultas que no forman parte de una condición

Las subconsultas no sólo se utilizan como parte de una condición en las cláusulas WHERE o HAVING, si no que pueden aparecer en otras partes de una instrucción SELECT.

Por ejemplo: mostrar los salarios brutos de todos los empleados y la diferencia entre su salario y el salario medio de todos los empleados que tengan como jefe al empleado con DNI 33252141.

```
select dni, apellidos, nombre, salario_bruto,
       round(salario_bruto-(select avg(salario_bruto) from empleado), 2) as diferencia
from (select dni, apellidos, nombre, salario_bruto
      from empleado
      where dni_jefe = '33456852') as jefe_33456852
order by diferencia;
```

Result Grid		Filter Rows:		Export:		Wrap Cell
dni	apellidos	nome	salario_bruto	diferencia_medio		
12852654	Martinez Iglesias	Benito	25000.00	-21068.50		
33147258	Ruiz Macias	Dario	32000.00	-14068.50		
33987456	Porto Novo	Begoña	52000.00	5931.50		
54528788	Canedo Tellez	Angeles	58500.00	12431.50		
33963258	Bernardez Macia	Luisa	65200.00	19131.50		
58745124	Cendan Villa	Lorenzo	65350.00	19281.50		