

## Consultas con subconsultas

## Índice

---

<b>1.</b>	<b>Consultas con subconsultas.....</b>	<b>3</b>
1.1	Subconsultas .....	3
1.2	Subconsultas como parte dunha condición.....	3
1.2.1	Utilizando operadores relacionais .....	4
	Subconsultas que devolven un escalar .....	4
	Subconsultas que devolven unha táboa .....	4
1.2.2	Utilizando o predicado BETWEEN .....	6
1.2.3	Utilizando o predicado IN .....	7
1.2.4	Utilizando o predicado EXISTS .....	8
1.3	Subconsultas que non forman parte dunha condición .....	9

# 1. Consultas con subconsultas

---

## 1.1 Subconsultas

Unha subconsulta é unha sentenza `SELECT` contida noutra sentenza da linguaxe de manipulación de datos. Os resultados da subconsulta afectan ao resultado da sentenza que a contén. Pódese dicir que unha subconsulta está aniñada na consulta externa ou consulta principal e é posible aniñar subconsultas dentro doutras subconsultas ata unha profundidade considerable.

Neste caso utilizaremos as subconsultas como parte dunha sentenza `SELECT` para resolver consultas complexas; tamén reciben o nome de consultas subordinadas.

Exemplo:

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

- A sentenza `SELECT column1 FROM t2` sería a subconsulta ou consulta subordinada.
- A sentenza `SELECT * FROM t1 WHERE column1 =` sería a consulta externa ou principal.

Algunhas características das subconsultas:

- Aparecen pechadas entre parénteses, dentro da consulta principal.
- Poden retornar un escalar (un valor único), unha táboa formada por unha ou varias columnas e unha ou varias filas ou o valor `NULL` (se non existe ningunha fila que cumpra as condicións da subconsulta).
- Os nomes de columna que aparecen nunha subconsulta poden facer referencia a táboas da consulta principal. A estas referencias chámaseles 'referencias externas' na subconsulta.
- Pódense utilizar en calquera parte da sentenza `SELECT`, aínda que o máis habitual é que formen parte dunha condición na cláusula `WHERE` ou na cláusula `HAVING`.

Avantaxes do uso de subconsultas:

- Permite escribir consultas estruturadas de forma que é posible illar cada parte da consulta, e descompoñer un problema grande en varios pequenos.
- Proporcionan unha maneira alternativa de resolver consultas que doutro modo necesitarían combinacións de táboas (`JOIN`) ou unións complexas.
- Fan consultas máis fáciles de ler.

## 1.2 Subconsultas como parte dunha condición

A forma máis habitual de utilización das subconsultas é formando parte dunha condición dentro das cláusulas `WHERE` ou `HAVING`. A forma de utilizar as subconsultas e os operadores que se poden utilizar na sentenza principal para facer comparacións co resultado da subconsulta, está en función dos valores que devolve a subconsulta, que poden ser valores escalares ou táboas.

## 1.2.1 Utilizando operadores relacionais

### Subconsultas que devolven un escalar

Na súa forma máis simple, unha subconsulta devolve un só valor, e pode ser utilizada como operando en calquera expresión (coma se fose un literal ou unha columna) ou como parámetro dunha función. Normalmente compárase o valor que devolve a subconsulta cunha expresión mediante un operador relacional. Sintaxe:

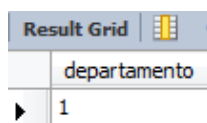
```
expresión operador_relacional (sentenza SELECT escalar)
```

Exemplo: mostrar *dni*, *apelidos* e *nome* dos empregados do mesmo departamento que o empregado que ten o DNI '12852654'.

- Paso 1: coñecer o departamento ao que pertence ese empregado. Iso pódese saber executando a consulta:

```
select departamento
from empregado
where dni = '12852654';
```

Resultado da execución:

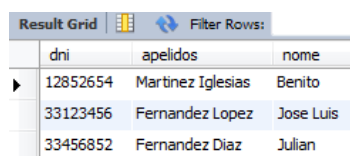


departamento
1

- Paso 2: a consulta anterior devolve un valor escalar e pode ser utilizada como subconsulta na sentenza SELECT que permite mostrar os datos solicitados dos empregados que traballan nese departamento:

```
select dni,apelidos,nome
from empregado
where departamento = (select departamento
                        from empregado
                        where dni = '12852654');
```

Resultado da execución:



dni	apelidos	nome
12852654	Martinez Iglesias	Benito
33123456	Fernandez Lopez	Jose Luis
33456852	Fernandez Diaz	Julian

### Subconsultas que devolven unha táboa

As subconsultas poden devolver unha táboa cunha fila polo menos. Para a comparación coa subconsulta hai que utilizar os operadores relacionais combinados coas palabras ALL ou ANY. Sintaxe:

```
expresión operador_relacional {ALL | ANY} (sentencia SELECT)
```

- Cando se utiliza a palabra ALL diante do operador relacional, a condición será verdadeira se é verdadeira para todos os valores que devolve a subconsulta.
- Cando se utiliza a palabra ANY diante do operador relacional, a condición é verdadeira se é verdadeira para polo menos un dos valores que devolve a subconsulta.

Exemplo con ALL: mostrar *dni*, *apelidos*, *nome* e *salario\_bruto* dos empregados que teñan un salario bruto superior ao de todos os empregados que traballan no departamento 4.

- Paso 1: seleccionar os salarios de todos os empregados do departamento 4:

```
select salario_bruto
from empregado
where departamento = 4
```

Resultado da execución:

Result Grid	
	salario_bruto
▶	35520.00
	25000.00
	58500.00

- Paso 2: a consulta anterior devolve unha táboa que ten varias filas, e unha columna cos salarios de todos os empregados do departamento 4. A sentenza SELECT que hai que construír ten que comparar o salario de cada empregado con todos os valores que devolve a subconsulta, e seleccionar só os que teñen un salario bruto maior que todos eses valores.

```
select dni,apelidos,nome,salario_bruto
from empregado
where salario_bruto > ALL (select salario_bruto
                           from empregado
                           where departamento = 4);
```

Resultado da execución:

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

Unha solución alternativa para resolver esta consulta consiste en calcular o salario bruto máis alto dos empregados do departamento 4, que sería un valor escalar, e despois comparar o salario bruto de cada empregado con ese valor escalar utilizando un operador relacional.

- Paso 1: calcular o salario máis alto dos empregados do departamento 4.

```
select max(salario_bruto)
from empregado
where departamento = 4
```

Resultado da execución:

Result Grid	
	max(salario_bruto)
▶	58500.00

- Paso 2: o resultado da consulta é un valor escalar e pódese comparar co operador relacional 'maior que' (>) .

```
select dni,apelidos, nome, salario_bruto
from empregado
where salario_bruto > (select max(salario_bruto)
                       from empregado
                       where departamento = 4);
```

Exemplo con ANY: mostrar *dni*, *apelidos*, *nome*, *salario\_bruto* e *departamento* dos empregados que traballan nalgún departamento da provincia de Lugo (*id\_provincia* = 27).

- Paso 1: Seleccionamos os códigos dos departamentos que están situados na provincia de Lugo.

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

Resultado da execución:

Result Grid	
	codigo
▶	1
	2
	6
	8

- Paso 2: na sentenza principal hai que comparar os departamentos de cada empregado dos valores que devolve a subconsulta, e seleccionar só aqueles que teñan un departamento igual a algún dos valores que devolve a subconsulta.

```
select dni,apelidos, nome, salario_bruto, departamento
from empregado
where departamento = ANY (select codigo
                           from departamento
                           where id_provincia = '27');
```

Resultado da execución:

Result Grid					
	dni	apelidos	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
	NULL	NULL	NULL	NULL	NULL

\* empregado 33 x

Output

Action Output

	Time	Action	Message	Duration / Fetch
✓	1	12:42:13	select dni,apelidos, nome, salario_bruto, departame... 7 row(s) returned	0.010 sec / 0.000 sec

## 1.2.2 Utilizando o predicado BETWEEN

Pódese utilizar o predicado BETWEEN con subconsultas que devolven un escalar. Serve para comparar unha expresión cun intervalo de valores que se obteñen dunha ou dúas subconsultas.

Exemplo: mostrar *dni*, *apelidos*, *nome*, *departamento* e *salario\_bruto* dos empregados que teñan un salario bruto entre o salario máis alto do departamento 2 e o salario máis alto do departamento 8.

- Paso 1: calcular o salario máis alto dos empregados do departamento 2.

```
select max(salario_bruto)
from empregado
where departamento = 2;
```

Result Grid	
	max(salario_bruto)
▶	42000.00

- Paso 2: calcular o salario máis alto dos empregados do departamento 8.

```
select max(salario_bruto)
from empleado
where departamento = 8;
```

Result Grid	
	max(salario_bruto)
▶	65200.00

- Paso 3: subconsulta final utilizando as dúas anteriores.

```
select dni,apellidos,nome,departamento,salario_bruto
from empleado
where salario_bruto between      (select max(salario_bruto)
                                from empleado
                                where departamento = 2)
and (select max(salario_bruto)
     from empleado
     where departamento = 8);
```

Result Grid		Filter Rows:			Edit:
	dni	apellidos	nome	departamento	salario_bruto
▶	15245258	Nuñez 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

### 1.2.3 Utilizando o predicado IN

O predicado IN utilízase cando a subconsulta devolve unha táboa cunha ou máis filas e permite comparar o valor que corresponde a unha expresión cos valores que devolve a subconsulta. Se de antemán se sabe que a subconsulta vai devolver unha táboa dunha fila, non é recomendable utilizar este predicado e deberíanse de utilizar operadores relacionais.

Sintaxe:

```
expresión [ NOT ] IN (sentenza SELECT)
```

- A condición é verdadeira cando o valor que representa a expresión é igual a algún dos valores que devolve a subconsulta.
- Cando se utiliza o operador NOT diante, a condición é verdadeira cando o valor que representa a expresión é distinto de todos os valores que devolve a subconsulta. No caso que na lista de valores estea incluído o valor NULL hai que ter en conta que a comparación de calquera valor co valor NULL devolve NULL e non *true* ou *false*, o que pode provocar problemas e obter resultados que non son correctos.

- Cando comparamos o valor dunha expresión cos valores que devolve unha subconsulta que produce como resultado unha táboa de máis dunha fila:
  - IN equivale a unha condición = ANY
  - NOT IN equivale a unha condición != ALL

Exemplo: mostrar *dni*, *apelidos*, *nome*, *salario\_bruto* e *departamento* dos empregados que traballan nalgún departamento que non estea situado na provincia de Lugo.

- Paso 1: seleccionar os códigos dos departamentos que están situados na provincia de Lugo.

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

Resultado da execución:

Result Grid	
	codigo
▶	1
	2
	6
	8

- Paso 2: na sentenza principal hai que comparar o departamento de cada empregado cos valores que devolve a subconsulta, e seleccionar só aqueles que teñan un departamento distinto a algún dos valores que devolve a subconsulta.

```
select dni,apelidos, nome, salario_bruto, departamento
from empregado
where departamento NOT IN (select codigo
                           from departamento
                           where id_provincia = '27');
```

Result Grid					
	dni	apelidos	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	Aguilar Lopez	Luis	35000.00	9
	33987456	Porto Novo	Begoña	52000.00	9
	54528788	Canedo Tellez	Angeles	58500.00	4

## 1.2.4 Utilizando o predicado EXISTS

Permite comprobar se a subconsulta devolve algunha fila ou non. Os valores que devolve a subconsulta non se comparan con ningún outro valor.

Sintaxe:

```
[ NOT ] EXISTS (sentencia SELECT)
```

- A condición é verdadeira se a subconsulta devolve unha o máis filas, e é falsa se a subconsulta non devolve ningunha fila.



Para que este tipo de condición teña utilidade, a subconsulta debe conter algunha condición que devolva resultados diferentes para cada fila da táboa da sentenza principal. Isto supón que nalgunha condición da subconsulta ten que intervir algunha columna das táboas da consulta principal ('referencias externas'); posto que se isto non fose así, a condición sería verdadeira ou falsa para todas as filas da consulta principal.

Exemplo: mostrar *dni*, *apelidos*, *nome*, *salario\_bruto* e *departamento* dos compañeiros de departamento do empregado con DNI 33456345, que teñan un salario maior có seu.

```
select dni,apelidos,nome,salario_bruto,departamento
from empregado as empl
where exists ( select emp2.departamento
               from empregado as emp2
               where emp2.dni = '33845215'
               and empl.departamento = emp2.departamento
               and empl.salario_bruto > emp2.salario_bruto);
```

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

## 1.3 Subconsultas que non forman parte dunha condición

As subconsultas non só se utilizan como parte dunha condición nas cláusulas WHERE ou HAVING, se non que poden aparecer noutras partes dunha instrución SELECT.

Exemplo: mostrar os salarios brutos de todos os empregados e a diferenza entre o seu salario e o salario medio de todos os empregados que teñan como xefe ao empregado con DNI 33252141.

```
select dni, apelidos, nome, salario_bruto,
       round(salario_bruto-(select avg(salario_bruto) from empregado),2) as diferenza
from (select dni, apelidos, nome, salario_bruto
      from empregado
      where dni_xefe = '33456852') as xefe_33456852
order by diferenza;
```

Result Grid					
Filter Rows:					
	dni	apelidos	nome	salario_bruto	diferencia_media
▶	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