

## UNIDAD 6: SUBCONSULTAS

Este capítulo, trata sobre funciones más avanzadas de la sentencia SELECT. Se escribirán subconsultas en la cláusula WHERE de otra sentencia SQL, para obtener valores basados en un valor condicional desconocido. Las subconsultas pueden ser de una sola fila y de varias filas.

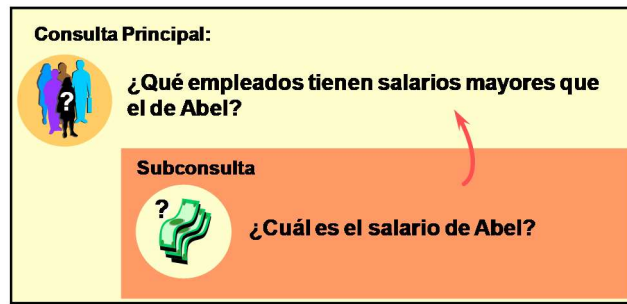
### USO DE UNA SUBCONSULTA PARA RESOLVER UN PROBLEMA

Suponga que desea escribir una consulta para averiguar quién gana un salario mayor que el de Abel. Para resolver este problema, necesita *dos* consultas: una para buscar lo que gana Abel y una segunda para buscar quién gana más de esta cantidad.

Puede resolver este problema combinando las dos consultas, colocando una consulta *dentro* de la otra.

La consulta interna o *subconsulta* devuelve un valor que utiliza la consulta externa o principal. La utilización de una subconsulta es equivalente a realizar dos consultas secuenciales y usar el resultado de la primera como valor de búsqueda en la segunda.

#### ¿Quién tiene un salario mayor que el de Abel?



### SINTAXIS DE LA SUBCONSULTA

Las subconsultas son sentencias SELECT embebidas en cláusulas de otras sentencias SELECT. Con ellas puede crear sentencias potentes a partir de sentencias simples. Esto puede resultar muy útil si necesita seleccionar filas de una tabla con una condición que depende de los datos de la propia tabla.

Puede colocar la subconsulta en diversas cláusulas SQL como, por ejemplo:

- La cláusula WHERE
- La cláusula HAVING
- La cláusula FROM

En la sintaxis:

*operator* incluye una condición de comparación como >, = o IN

**Nota:** Las condiciones de comparación son de dos clases: operadores de una sola fila (>, =, >=, <, <>, <=) y operadores de varias filas (IN, ANY, ALL).

Con frecuencia se hace referencia a la subconsulta como sentencia SELECT anidada, sub-SELECT o SELECT interna. La subconsulta se suele ejecutar primero y su resultado se utiliza para completar la condición de consulta de la consulta principal o externa.

```
SELECT select_list
FROM table
WHERE expr operator
      (SELECT select_list
       FROM table);
```

- La subconsulta (consulta interna) se ejecuta una vez antes de la consulta principal.
- El resultado de la subconsulta lo utiliza la consulta principal (consulta externa).

## USO DE UNA SUBCONSULTA

En la siguiente figura, la consulta interna determina el salario del empleado Abel. La consulta externa toma el resultado de la consulta interna y lo utiliza para mostrar todos los empleados que ganan más de esa cantidad.

```
SELECT last_name
FROM employees 11000
WHERE salary >
(SELECT salary
FROM employees
WHERE last_name = 'Abel');
```

LAST_NAME
King
Kochhar
De Haan
Hartstein
Higgins

## INSTRUCCIONES PARA UTILIZAR SUBCONSULTAS

- La subconsulta se debe escribir entre paréntesis.
- Coloque la subconsulta a la derecha de la condición de comparación para una mejor legibilidad.
- Antes de la versión Oracle8i, las subconsultas no podían contener una cláusula ORDER BY. Sólo se puede utilizar una cláusula ORDER BY para una sentencia SELECT y, si se especifica, debe ser la última cláusula de la sentencia SELECT principal. A partir de la versión Oracle8i, se puede utilizar una cláusula ORDER BY y es necesaria en la subconsulta para realizar un análisis N principal.
- En las subconsultas se utilizan dos clases de condiciones de comparación: operadores de una sola fila y operadores de varias filas.

## TIPOS DE SUBCONSULTAS

- Subconsultas de una sola fila: consultas que devuelven una sola fila a partir de la sentencia SELECT interna.
- Subconsultas de varias filas: consultas que devuelven más de una fila a partir de la sentencia SELECT interna.
- Subconsultas de varias columnas: consultas que devuelven más de una columna a partir de la sentencia SELECT interna.

- **Subconsulta de una sola fila**



- **Subconsulta de varias filas**



## SUBCONSULTAS DE UNA SOLA FILA

Una subconsulta de una sola fila es aquella que devuelve una fila a partir de la sentencia SELECT interna. Este tipo de subconsulta utiliza un operador de una sola fila. La transparencia proporciona una lista de operadores de una sola fila.

### Ejemplo

Mostrar los empleados cuyos identificadores de cargo sean los mismos que el del empleado 141.

```
SELECT last_name, job_id
FROM employees
WHERE job_id =
  (SELECT job_id
   FROM employees
   WHERE employee_id = 141);
```

- Devuelven una sola fila
- Utilizan operadores de comparación de una sola fila

Operador	Significado
=	Igual que
>	Mayor que
>=	Mayor o igual que
<	Menor que
<=	Menor o igual que
<>	No igual a

### EJECUCION DE SUBCONSULTAS DE UNA SOLA FILA

Se puede considerar una consulta SELECT como un bloque de consultas. El ejemplo de la transparencia muestra empleados cuyos identificadores de cargo son los mismos que el del empleado 141 y cuyos salarios son mayores que el del empleado 143.

El ejemplo consta de tres bloques de consultas: la consulta externa y dos consultas internas. Los bloques de la consulta interna se ejecutan en primer lugar y producen los resultados de consulta ST\_CLERK y 2600, respectivamente. A continuación, se procesa el bloque de la consulta externa y éste utiliza los valores devueltos por las consultas internas para completar sus condiciones de búsqueda.

Las dos consultas internas devuelven valores únicos (ST\_CLERK y 2600, respectivamente), por lo que esta sentencia SQL se llama subconsulta de una sola fila.

**Nota:** Las consultas externas e internas pueden obtener datos de tablas diferentes.

```
SELECT last_name, job_id, salary
FROM employees
WHERE job_id =
  (SELECT job_id
   FROM employees
   WHERE employee_id = 141)
AND salary >
  (SELECT salary
   FROM employees
   WHERE employee_id = 143);
```

LAST_NAME	JOB_ID	SALARY
Rajs	ST_CLERK	3500
Davies	ST_CLERK	3100

### USO DE FUNCIONES DE GRUPO EN UNA SUBCONSULTA

Puede mostrar datos de una consulta principal utilizando una función de grupo en una subconsulta para devolver una sola fila. La subconsulta está entre paréntesis y colocada detrás de la condición de comparación.

El ejemplo de la figura muestra el apellido del empleado, el identificador de cargo y el salario de todos los empleados cuyos salarios sean iguales al mínimo. La función de grupo MIN devuelve un valor único (2500) a la consulta externa.

```
SELECT last_name, job_id, salary
FROM employees
WHERE salary =
    (SELECT MIN(salary)
     FROM employees);
```

LAST_NAME	JOB_ID	SALARY
Vargas	ST_CLERK	2500

## LA CLAUSULA HAVING CON SUBCONSULTAS

Puede utilizar subconsultas no sólo en la cláusula WHERE, sino también en la cláusula HAVING. Oracle Server ejecuta la subconsulta y los resultados se devuelven a la cláusula HAVING de la consulta principal.

La sentencia SQL de la figura muestra todos los departamentos que tienen un salario mínimo mayor que el del departamento 50.

- Oracle Server ejecuta en primer lugar las subconsultas.
- Oracle Server devuelve resultados a la cláusula HAVING de la consulta principal.

```
SELECT department_id, MIN(salary)
FROM employees
GROUP BY department_id
HAVING MIN(salary) >
    (SELECT MIN(salary)
     FROM employees
     WHERE department_id = 50);
```

### Ejemplo

Busque el cargo con el salario medio más bajo.

```
SELECT job_id, AVG(salary)
FROM employees
GROUP BY job_id
HAVING AVG(salary) = (SELECT MIN(AVG(salary))
                     FROM employees
                     GROUP BY job_id);
```

## QUE ES INCORRECTO EN ESTA SENTENCIA

Un error común de las subconsultas es que se devuelve más de una fila para una subconsulta de una sola fila.

En la sentencia SQL de la transparencia, la subconsulta contiene una cláusula GROUP BY, que implica que devolverá varias filas, una por cada grupo que busque. En este caso, el resultado de la subconsulta será 4400, 6000, 2500, 4200, 7000, 17000 y 8300.

La consulta externa toma los resultados de la subconsulta (4400, 6000, 2500, 4200, 7000, 17000, 8300) y los utiliza en su cláusula WHERE. Esta cláusula contiene un operador igual (=), un operador de comparación de una sola fila que espera un solo valor. El operador = no puede aceptar más de un valor de la subconsulta y, por lo tanto, genera el error.

Para corregir este error, cambie el operador = por IN.

### ¿Qué Es Incorrecto en esta Sentencia?

```
SELECT employee_id, last_name
FROM employees
WHERE salary =
    (SELECT MIN(salary)
     FROM employees
     GROUP BY department_id);
```

```
ERROR at line 4:
ORA-01427: single-row subquery returns more than
one row
```

Operador de una sola fila con subconsulta de varias filas



## DEVUELVE FILAS ESTA SENTENCIA

Un problema común de las subconsultas, es que la consulta interna no devuelva ninguna fila. En la sentencia SQL de la figura, la subconsulta contiene una cláusula WHERE. Probablemente, la intención es buscar el empleado de nombre Haas. La sentencia es correcta, pero no selecciona ninguna fila cuando se ejecuta.

No hay ningún empleado llamado Haas, por lo que la subconsulta no devuelve ninguna fila. La consulta externa toma los resultados de la subconsulta (valor nulo) y los utiliza en su cláusula WHERE. La consulta externa no encuentra ningún empleado con el identificador de cargo igual a valor nulo y no devuelve ninguna fila. Si existiera un cargo con el valor nulo, la fila no se devolvería porque la comparación de dos valores nulos origina un valor nulo, por lo tanto la condición WHERE no es verdadera.

```
SELECT last_name, job_id
FROM employees
WHERE job_id =
    (SELECT job_id
     FROM employees
     WHERE last_name = 'Haas');
```

```
no rows selected
```

**La subconsulta no devuelve ningún valor**

## SUBCONSULTAS DE VARIAS FILAS

Las subconsultas que devuelven más de una fila se denominan subconsultas de varias filas. Se utiliza un operador de varias filas, en lugar de uno de una sola fila. El operador de varias filas espera un valor o varios.

```
SELECT last_name, salary, department_id
FROM employees
WHERE salary IN (SELECT MIN(salary)
                FROM employees
                GROUP BY department_id);
```

### Ejemplo

Busque los empleados que ganan el mismo salario que el mínimo de cada departamento. La consulta interna se ejecuta en primer lugar y proporciona un resultado de consulta. A continuación, se procesa el bloque de consulta principal y utiliza los valores devueltos por la consulta interna para completar su condición de búsqueda. En realidad, la consulta principal aparecería en Oracle Server como se indica a continuación:

```
SELECT last_name, salary, department_id
FROM employees
WHERE salary IN (2500, 4200, 4400, 6000, 7000, 8300, 8600, 17000);
```

Operador	Significado
IN	Igual a cualquier miembro de la lista
ANY	Compara el valor con cada valor devuelto por la subconsulta
ALL	Compara el valor con todos los valores devueltos por la subconsulta

## USO DEL OPERADOR ANY

El operador ANY (y su sinónimo, el operador SOME) compara un valor con *cada* valor devuelto por una subconsulta. El ejemplo de la figura muestra los empleados que no son programadores de IT y cuyos salarios son menores que los de cualquier programador de IT. El salario máximo que gana un programador es de \$9.000.

=ANY es equivalente a IN.

< ANY significa menos que el máximo. > ANY significa más que el mínimo.  
> ALL significa más que el máximo. < ALL significa menos que el mínimo.

```
SELECT employee_id, last_name, job_id, salary
FROM employees 9000,6000,4200
WHERE salary < ANY
                (SELECT salary
                 FROM employees
                 WHERE job_id = 'IT_PROG')
AND job_id <> 'IT_PROG';
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
124	Mourgos	ST_MAN	5800
141	Rajs	ST_CLERK	3500
142	Davies	ST_CLERK	3100
143	Matos	ST_CLERK	2600
144	Vargas	ST_CLERK	2500

10 rows selected.

### USO DEL OPERADOR ALL

El operador ALL compara un valor con *todos* los valores devueltos por una subconsulta. El ejemplo de la transparencia muestra los empleados cuyos salarios son menores que el salario de todos los empleados con el identificador de cargo IT\_PROG y cuyos cargos no son IT\_PROG.

El operador NOT se puede utilizar con los operadores IN, ANY y ALL.

```
SELECT employee_id, last_name, job_id, salary
FROM employees
WHERE salary < ALL
                (SELECT salary
                 FROM employees
                 WHERE job_id = 'IT_PROG')
AND job_id <> 'IT_PROG';
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
141	Rajs	ST_CLERK	3500
142	Davies	ST_CLERK	3100
143	Matos	ST_CLERK	2600
144	Vargas	ST_CLERK	2500

### VALORES NULOS EN UNA SUBCONSULTA

La sentencia SQL de la figura intenta mostrar todos los empleados que no tienen subordinados. Lógicamente, esta sentencia SQL debería haber devuelto 12 filas. Sin embargo, no devuelve ninguna. Uno de los valores devueltos por la consulta interna es un valor nulo, por lo que la consulta entera no devuelve ninguna fila. El motivo es que todas las condiciones que comparen un valor nulo producen un valor nulo. Por ello, si es posible que en el juego de resultados de una subconsulta haya valores nulos, no utilice el operador NOT IN. Este operador es equivalente a <> ALL.

```
SELECT emp.last_name
FROM employees emp
WHERE emp.employee_id NOT IN
                (SELECT mgr.manager_id
                 FROM employees mgr);

no rows selected
```

Observe que el valor nulo como parte del juego de resultados de una subconsulta no supone un problema si utiliza el operador IN. Este operador es equivalente a =ANY. Por ejemplo, para mostrar los empleados que tienen subordinados, utilice la siguiente sentencia SQL:



```
SELECT emp.last_name
FROM employees emp
WHERE emp.employee_id IN
      (SELECT mgr.manager_id
       FROM employees mgr);
```

Igualmente, se puede incluir una cláusula WHERE en la subconsulta para mostrar todos los empleados que no tienen subordinados:

```
SELECT last_name FROM employees
WHERE employee_id NOT IN
      (SELECT manager_id
       FROM employees
       WHERE manager_id IS NOT NULL);
```