

Subconsultas Avanzadas

¿Quién tiene un salario mayor que el de

Consulta Principal:



¿Qué empleados tienen salarios mayores que el de Abel?

Subconsulta



¿Cuál es el salario de Abel?

¿Qué Es una Subconsulta?

Una subconsulta es una sentencia `SELECT` que está embebida en una cláusula de otra sentencia SQL.

Consulta principal →

```
SELECT ...  
FROM ...  
WHERE ...
```

```
(SELECT ...  
FROM ...  
WHERE ...)
```

← **Subconsulta**

Sintaxis de Subconsulta

```
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).

Tipos de Subconsultas

- Subconsulta de una sola fila



- Subconsulta de varias filas




Subconsultas de una Sola Fila

- 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

Uso de una Subconsulta

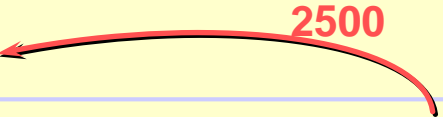
```
SELECT last_name
FROM employees
WHERE salary > 10500
               (SELECT salary
                  FROM employees
                  WHERE employee_id = 149) ;
```

A blue arrow points from the value 10500 to the comparison operator > in the WHERE clause of the main query.

LAST_NAME
King
Kochhar
De Haan
Abel
Hartstein
Higgins

6 rows selected.

Uso de Funciones de Grupo en una Subconsulta

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

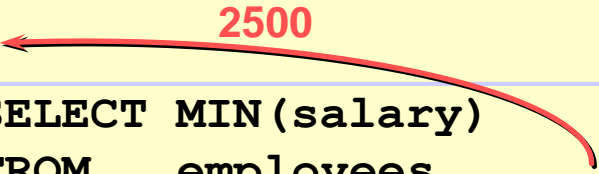
LAST_NAME	JOB_ID	SALARY
Vargas	ST_CLERK	2500

La Cláusula HAVING con Subconsultas

- 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);
```



2500

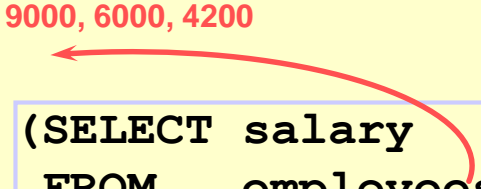
Subconsultas de Varias Filas

- Devuelven más de una fila
- Utilizan operadores de comparación de varias filas

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 en Subconsultas de Varias Filas

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



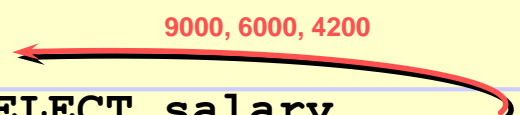
The diagram shows a red arrow pointing from the subquery result set (9000, 6000, 4200) to the 'ANY' operator in the main query's WHERE clause, indicating that the salary must be less than any of these values.

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.

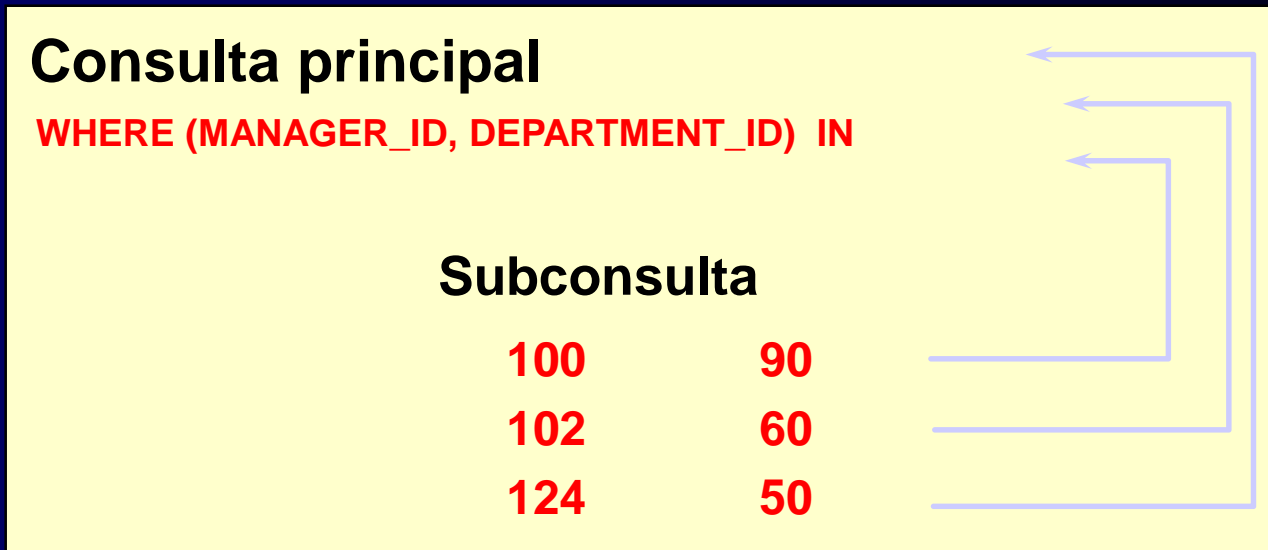
Uso del Operador ALL en Subconsultas de Varias Filas

```
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

Subconsultas de Varias Columnas



Cada fila de la consulta principal se compara con los valores de una subconsulta de varias filas y de varias columnas.

Comparaciones de Columnas

- Las comparaciones de columnas en una subconsulta de varias columnas pueden ser:
 - Comparaciones entre pares
 - Comparaciones no entre pares

Subconsulta de Comparación entre Pares

Visualice los detalles de los empleados dirigidos por el mismo director y que trabajen en el mismo departamento que los empleados cuyo EMPLOYEE_ID sea 178 o 174.

```
SELECT employee_id, manager_id, department_id
FROM   employees
WHERE  (manager_id, department_id) IN
      (SELECT manager_id, department_id
       FROM   employees
       WHERE  employee_id IN (178,174))
AND employee_id NOT IN (178,174);
```

Subconsulta de Comparación

No entre Pares

Visualice los detalles de los empleados dirigidos por el mismo director que los empleados con EMPLOYEE_ID 174 o 141 y que trabajen en el mismo departamento que los empleados con EMPLOYEE_ID 174 o 141.

```
SELECT  employee_id, manager_id, department_id
FROM    employees
WHERE   manager_id IN
        (SELECT  manager_id
         FROM    employees
         WHERE   employee_id IN (174,141))
AND     department_id IN
        (SELECT  department_id
         FROM    employees
         WHERE   employee_id IN (174,141))

AND employee_id NOT IN(174,141);
```


Uso de una Subconsulta en la Cláusula FROM

```
SELECT  a.last_name, a.salary,  
        a.department_id, b.salavg  
FROM    employees a, (SELECT  department_id,  
                        AVG(salary) salavg  
                        FROM    employees  
                        GROUP BY department_id) b  
WHERE   a.department_id = b.department_id  
AND     a.salary > b.salavg;
```

LAST_NAME	SALARY	DEPARTMENT_ID	SALAVG
Hartstein	13000	20	9500
Mourgos	5800	50	3500
Hunold	9000	60	6400
Zlotkey	10500	80	10033.3333
Abel	11000	80	10033.3333
King	24000	90	19333.3333
Higgins	12000	110	10150

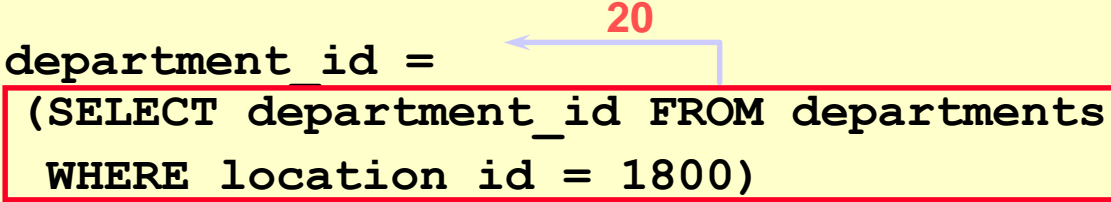
7 rows selected.

Expresiones de Subconsulta Escalar

- Una expresión de subconsulta escalar es una subconsulta que devuelve exactamente un valor de columna de una fila.
- Las subconsultas escalares se pueden utilizar en :
 - Sentencia SELECT (cláusulas FROM y WHERE)
 - Lista VALUES de una sentencia INSERT
 - Condición y expresión parte de DECODE y CASE
 - Todas las cláusulas de SELECT excepto GROUP BY

Subconsultas Escalares: Ejemplos

Subconsultas Escalares en Expresiones

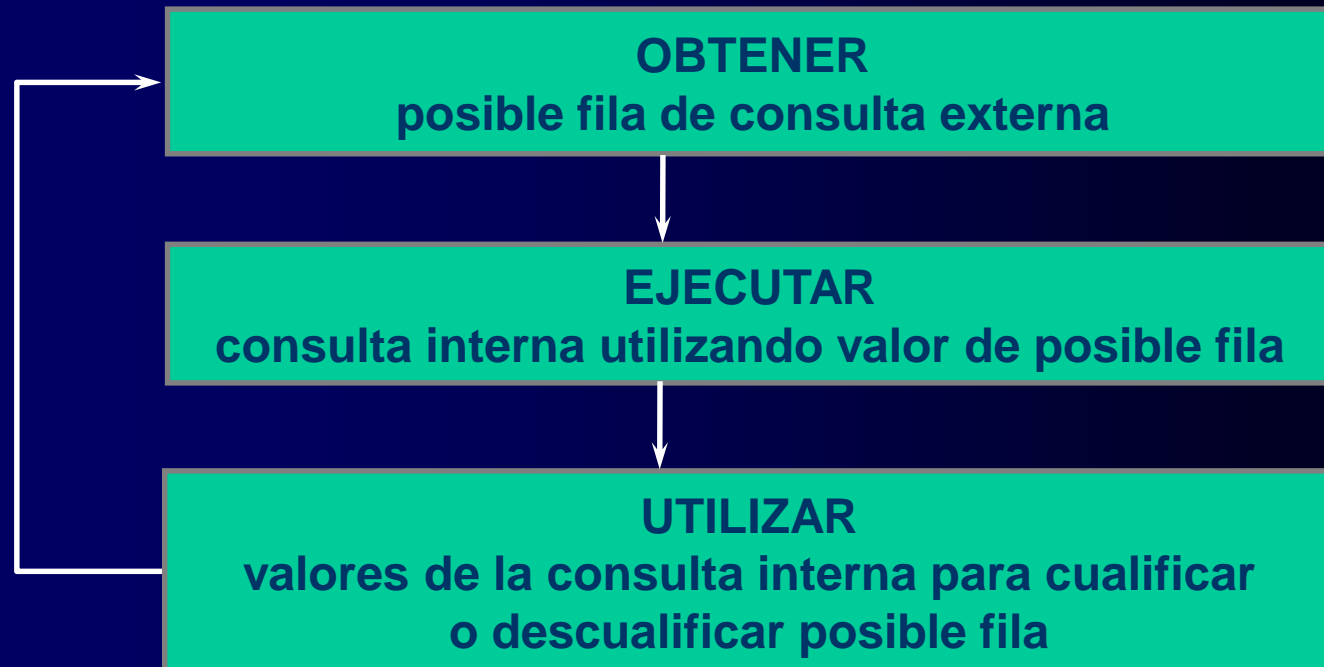
```
SELECT employee_id, last_name,  
       (CASE  
         WHEN department_id =   
           (SELECT department_id FROM departments  
            WHERE location_id = 1800)  
         THEN 'Canada' ELSE 'USA' END) location  
FROM   employees;
```

Subconsultas Escalares en la Cláusula Order By

```
SELECT   employee_id, last_name  
FROM     employees e  
ORDER BY (SELECT department_name  
          FROM departments d  
          WHERE e.department_id = d.department_id);
```

Subconsultas Correlacionadas

Las subconsultas correlacionadas se utilizan para el procesamiento fila a fila. Cada subconsulta se ejecuta una vez para cada fila de la consulta externa.



Subconsultas Correlacionadas


```
SELECT column1, column2, ...  
FROM   table1 outer  
WHERE  column1 operator  
        (SELECT column1  
              FROM   table2  
              WHERE  expr1 =  
                    outer .expr2) ;
```

La subconsulta hace referencia a una columna de una tabla en la consulta principal.

Uso de Subconsultas Correlacionadas

Busque todos los empleados que ganan más del salario medio en su departamento.

```
SELECT last_name, salary, department_id
FROM   employees outer
WHERE  salary >
      (SELECT AVG(salary)
       FROM   employees
       WHERE  department_id =
             outer.department_id) ;
```



Cada vez que se procesa una fila de la consulta externa, se evalúa la consulta interna.

Uso de Subconsultas Correlacionadas

Visualice detalles de los empleados que han cambiado de cargo al menos dos veces.

```
SELECT e.employee_id, last_name, e.job_id
FROM   employees e
WHERE  2 <= (SELECT COUNT(*)
              FROM   job_history
              WHERE  employee_id = e.employee_id);
```

EMPLOYEE_ID	LAST_NAME	JOB_ID
101	Kochhar	AD_VP
176	Taylor	SA_REP
200	Whalen	AD_ASST

Uso del Operador EXISTS

- El operador EXISTS comprueba la existencia de filas en el juego de resultados de la subconsulta.
- Si se encuentra un valor de fila de la subconsulta:
 - La búsqueda no continúa en la consulta interna.
 - Se señala a la condición como TRUE.
- Si no se encuentra un valor de fila de la subconsulta:
 - Se señala a la condición como FALSE.
 - La búsqueda continúa en la consulta interna.

Uso del Operador EXISTS

Busque los empleados que tengan al menos una persona que les informe.

```
SELECT employee_id, last_name, job_id, department_id
FROM   employees outer
WHERE  EXISTS ( SELECT 'X'
                FROM   employees
                WHERE  manager_id =
                      outer.employee_id) ;
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90
103	Hunold	IT_PROG	60
124	Mourgos	ST_MAN	50
149	Zlotkey	SA_MAN	80
201	Hartstein	MK_MAN	20
205	Higgins	AC_MGR	110

8 rows selected.

Uso del Operador NOT EXISTS

Busque todos los departamentos que no tengan empleados.

```
SELECT department_id, department_name
FROM departments d
WHERE NOT EXISTS (SELECT 'X'
                   FROM employees
                   WHERE department_id
                     = d.department_id);
```

DEPARTMENT_ID	DEPARTMENT_NAME
190	Contracting

UPDATE Correlacionado

```
UPDATE table1 alias1
SET    column = (SELECT expression
                      FROM   table2 alias2
                      WHERE  alias1.column =
                           alias2.column);
```

Utilice una subconsulta correlacionada para actualizar las filas de una tabla basadas en las filas de otra tabla.

UPDATE Correlacionado

- Desnormalice la tabla EMPLOYEES agregando una columna que almacene el nombre de departamento.
- Rellene la tabla mediante una actualización correlacionada.

```
ALTER TABLE employees  
ADD (department_name VARCHAR2(14));
```

```
UPDATE employees e  
SET    department_name =  
        (SELECT department_name  
         FROM    departments d  
         WHERE   e.department_id = d.department_id);
```

DELETE Correlacionado

```
DELETE FROM table1 alias1
WHERE  column operator
        (SELECT expression
         FROM   table2 alias2
         WHERE  alias1.column = alias2.column);
```

Utilice una subconsulta correlacionada para suprimir las filas de una tabla basadas en filas de otra tabla.

DELETE Correlacionado

Utilice una subconsulta correlacionada para suprimir solamente las filas de la tabla EMPLOYEES que también existen en la tabla EMP_HISTORY.

```
DELETE FROM employees E
WHERE employee_id =
      (SELECT employee_id
       FROM   emp_history
       WHERE  employee_id = E.employee_id) ;
```