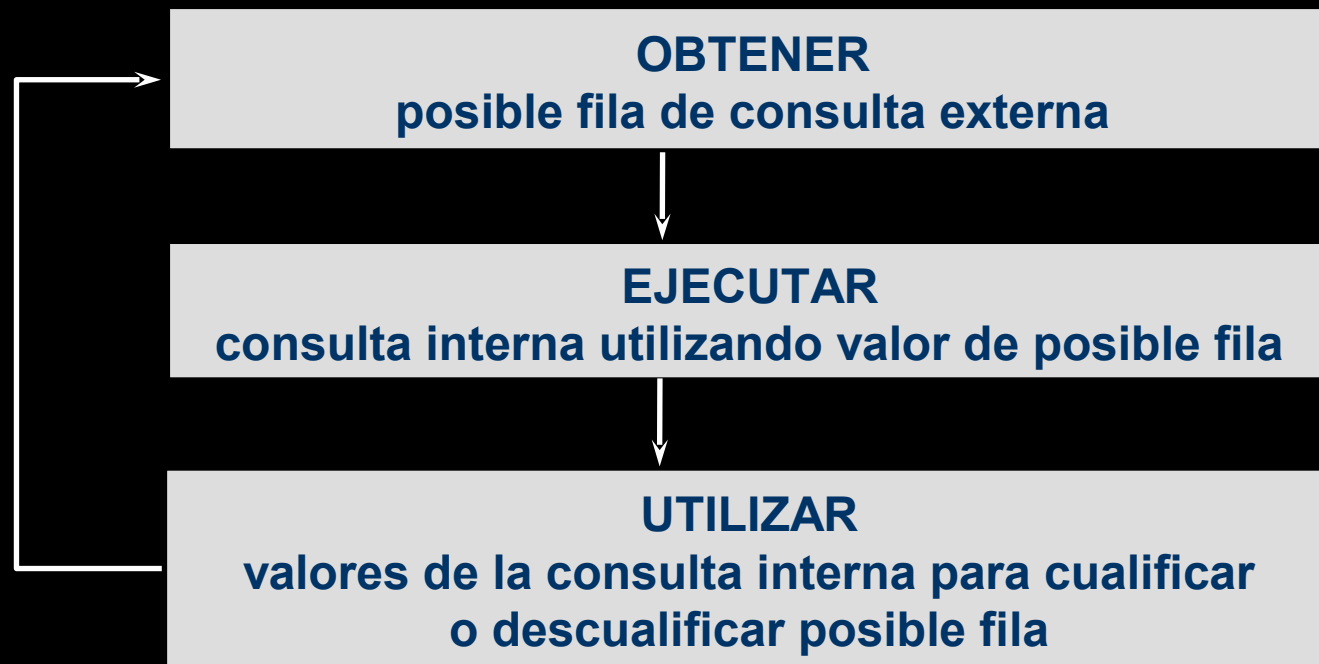


# Subconsultas Correlacionadas

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, column2
                 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);
```