



ORACLE

Academy



Database Programming with SQL

10-4

Subconsultas Correlacionadas

ORACLE
Academy



Objetivos

- En esta lección se abordan los siguientes objetivos:
 - Identificar cuándo se necesitan subconsultas correlacionadas
 - Crear y ejecutar subconsultas correlacionadas
 - Crear una consulta con los operadores EXISTS y NOT EXISTS para probar filas devueltas de la subconsulta
 - Crear y ejecutar subconsultas con nombre mediante la cláusula WITH



Objetivo

- En ocasiones, puede que tenga que responder a más de una pregunta en una frase
- Su amigo puede preguntarle si tiene suficiente dinero para una entrada de cine, palomitas y una bebida
- Antes de que pueda responder a su amigo, tiene que saber los precios de las entradas, las palomitas y la bebida
- También tiene que ver cuánto dinero tiene en el bolsillo
- Por lo tanto, en realidad, lo que parecía una pregunta fácil, se convierte en cuatro preguntas para las que necesita respuestas antes de poder decir "Sí" o "No"

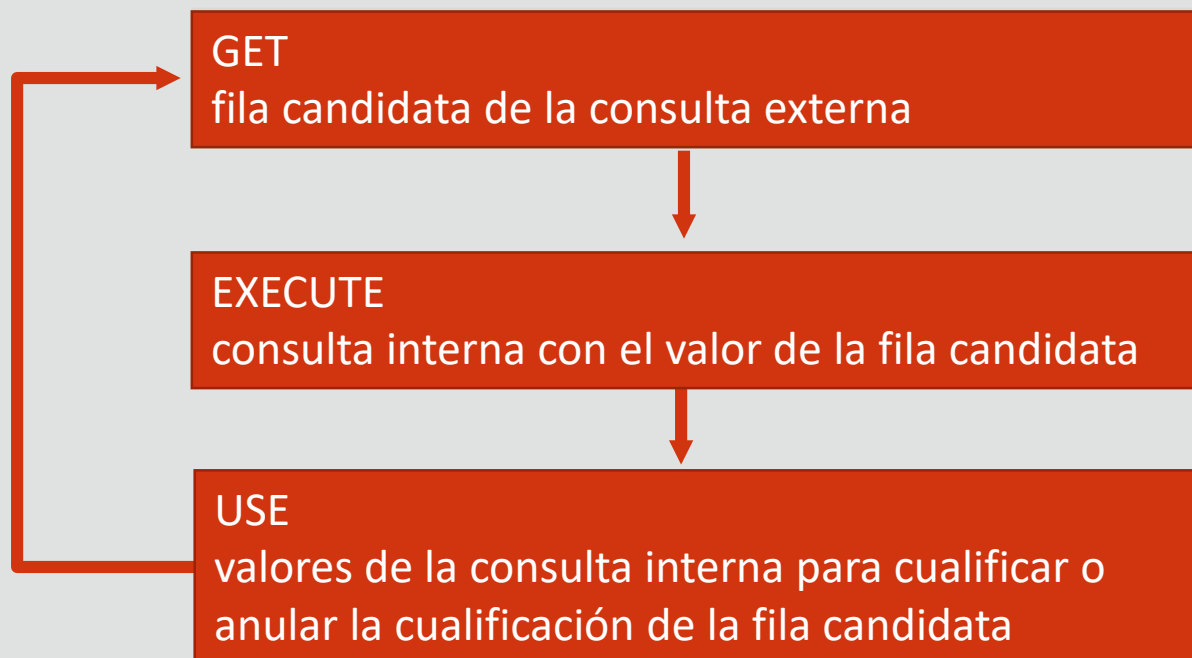
Objetivo

- En un negocio, puede que le pidan elaborar un informe de todos los empleados que ganan más que el salario medio de sus departamentos
- Por lo tanto, aquí primero tiene que calcular el salario medio por departamento y, a continuación, comparar el salario de cada empleado con el salario medio del departamento de ese empleado



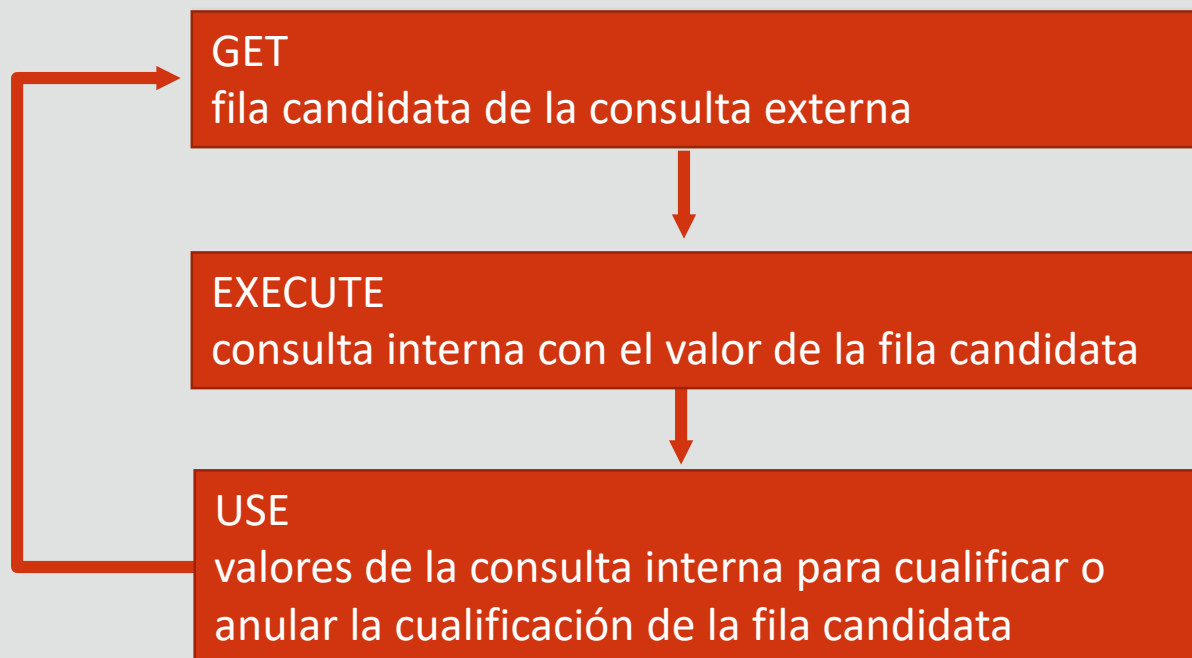
Subconsultas Correlacionadas

- El servidor de Oracle realiza una subconsulta correlacionada cuando la subconsulta hace referencia a una columna de una tabla a la que se hace referencia en la sentencia principal



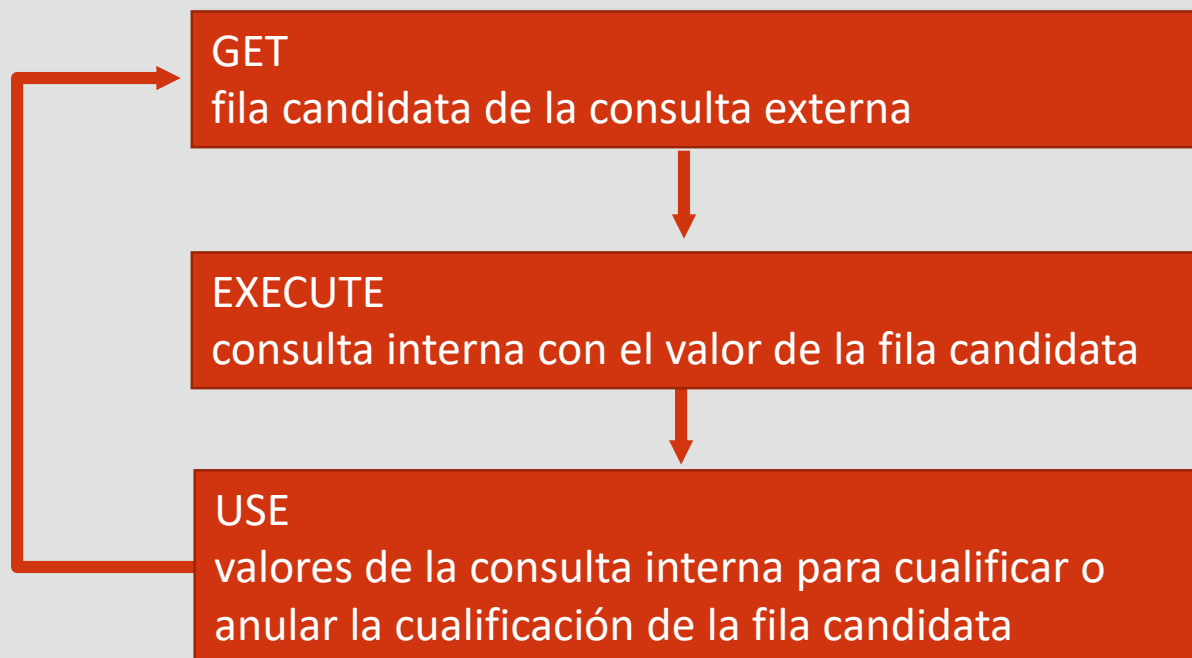
Subconsultas Correlacionadas

- Una subconsulta correlacionada se evalúa una vez para cada fila procesada por la sentencia principal



Subconsultas Correlacionadas

- La sentencia principal puede ser una sentencia SELECT, UPDATE o DELETE



Ejemplo de Subconsulta Correlacionada

- ¿De quién es el salario que es mayor que el salario medio de su departamento?
- Para responder a esta pregunta, tenemos que escribir una subconsulta correlacionada
- Las subconsultas correlacionadas se utilizan para el procesamiento fila por fila

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

FIRST_NAME	LAST_NAME	SALARY
Steven	King	24000
Shelley	Higgins	12000
Eleni	Zlotkey	10500
Ellen	Abel	11000
Kevin	Mourgos	5800
Alexander	Hunold	9000
Michael	Hartstein	13000

Ejemplo de Subconsulta Correlacionada

- Cada subconsulta se ejecuta una vez para cada fila de la consultar externa
- Con una subconsulta normal, la consulta SELECT interna se ejecuta en primer lugar y una vez, devolviendo valores que utilizará la consulta externa

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

FIRST_NAME	LAST_NAME	SALARY
Steven	King	24000
Shelley	Higgins	12000
Eleni	Zlotkey	10500
Ellen	Abel	11000
Kevin	Mourgos	5800
Alexander	Hunold	9000
Michael	Hartstein	13000

Ejemplo de Subconsulta Correlacionada

- Una subconsulta correlacionada, sin embargo, se ejecuta una vez para cada fila considerada por la consulta externa
- Es decir, la consulta interna está controlada por la consulta externa
- La subconsulta correlacionada en este ejemplo está marcada en rojo

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

FIRST_NAME	LAST_NAME	SALARY
Steven	King	24000
Shelley	Higgins	12000
Eleni	Zlotkey	10500
Ellen	Abel	11000
Kevin	Mourgos	5800
Alexander	Hunold	9000
Michael	Hartstein	13000

EXISTS Y NOT EXISTS en Subconsultas

- EXISTS y su opuesto NOT EXISTS son dos cláusulas que se pueden utilizar al comprobar las coincidencias de subconsultas
- EXISTS comprueba un resultado TRUE o coincidente en la subconsulta
- Para responder a la pregunta: "¿qué empleados no son jefes?"
 - Primero tiene que preguntar "¿quiénes son los jefes?"
 - Y, a continuación, preguntar "¿quién NO EXISTE en la lista de jefes?"

EXISTS Y NOT EXISTS en Subconsultas

- En este ejemplo, la subconsulta selecciona los empleados que son jefes
- A continuación, la consulta externa devuelve las filas de la tabla de empleados que NO EXISTEN en la subconsulta

```
SELECT last_name AS "Not a Manager"
FROM   employees emp
WHERE  NOT EXISTS
      (SELECT *
       FROM employees mgr
        WHERE mgr.manager_id = emp.employee_id);
```

Not a Manager
Whalen
Gietz
Abel
Taylor
Grant
Rajs
Davies
Matos
Vargas
Ernst
...

EXISTS Y NOT EXISTS en Subconsultas

- Si se ejecuta la misma consulta con NOT IN en lugar de NOT EXISTS, el resultado es muy diferente
- El resultado de esta consulta sugiere que no hay ningún empleado que también sea jefe, por lo que todos los empleados son jefes, lo que ya sabemos que no es cierto

```
SELECT last_name AS "Not a Manager"  
FROM   employees emp  
WHERE  emp.employee_id NOT IN  
       (SELECT mgr.manager_id  
        FROM employees mgr);
```

no se han encontrado datos

EXISTS Y NOT EXISTS en Subconsultas

- La causa del extraño resultado es el valor NULL devuelto por la subconsulta
- Una de las filas de la tabla employees no tiene un jefe y esto hace que todo el resultado sea incorrecto
- Las subconsultas pueden devolver tres valores: TRUE, FALSE y UNKNOWN
- Un valor NULL en el juego de resultados de la subconsulta devolverá un valor UNKNOWN, que Oracle no puede evaluar, por lo que no lo hace

```
SELECT last_name AS "Not a Manager"  
FROM   employees emp  
WHERE  emp.employee_id NOT IN  
       (SELECT mgr.manager_id  
        FROM employees mgr);
```

no se han encontrado datos

EXISTS Y NOT EXISTS en Subconsultas

- TENGA CUIDADO con los valores NULL en las subconsultas al utilizar IN o NOT IN
- Si no está seguro de si una subconsulta incluirá un valor nulo, elimine el valor nulo mediante IS NOT NULL en una cláusula WHERE
- Por ejemplo : WHERE emp.manager_id IS NOT NULL o utilice NOT EXISTS para estar seguro

Cláusula WITH

- Si tiene que escribir una consulta muy compleja con uniones y agregaciones utilizadas varias veces, puede escribir las distintas partes de la sentencia como bloques de consulta y, a continuación, utilizar esos mismos bloques de consulta en una sentencia SELECT
- Oracle le permite escribir subconsultas con nombre en una sola sentencia, siempre que empiece su sentencia con la palabra clave WITH
- La cláusula WITH recupera los resultados de uno o más bloques de consulta y almacena esos resultados para el usuario que ejecuta la consulta

Cláusula WITH

- La cláusula WITH mejora el rendimiento
- La cláusula WITH hace que la consulta se pueda leer fácilmente
- La sintaxis de la cláusula WITH es la siguiente:

```
WITH    subquery-name AS (subquery),  
        subquery-name AS (subquery)  
SELECT  column-list  
FROM    {table | subquery-name | view}  
WHERE   condition is true;
```

Cláusula WITH

- Escriba la consulta para el siguiente requisito:
 - Visualice una lista de apellidos de los empleados que no son jefes
- Para crear esta consulta, primero tendrá que obtener una lista de valores de `manager_id` de la tabla de empleados y, a continuación, devolver los nombres de los empleados cuyo ID de empleado no esté en la lista de jefes
- Podemos crear una subconsulta con nombre mediante la cláusula `WITH` para recuperar los valores de `manager_id` de la tabla `employees` y, a continuación, la consulta externa devolverá los empleados que no aparecen en esa lista

Cláusula WITH

```
WITH managers AS
  (SELECT DISTINCT manager_id
   FROM employees
   WHERE manager_id IS NOT NULL)

SELECT last_name AS "Not a manager"
FROM employees
WHERE employee_id NOT IN
  (SELECT *
   FROM managers) ;
```

Not a manager
Whalen
Gietz
Abel
Taylor
Grant
Rajs
Davies
Vargas
Ernst
...

Resumen

- En esta lección, debe haber aprendido lo siguiente:
 - Identificar cuándo se necesitan subconsultas correlacionadas
 - Crear y ejecutar subconsultas correlacionadas
 - Crear una consulta con los operadores EXISTS y NOT EXISTS para probar filas devueltas de la subconsulta
 - Crear y ejecutar subconsultas con nombre mediante la cláusula WITH





ORACLE

Academy

