

TEMA 4.4.-SUBCONSULTAS

A veces, para realizar alguna operación de consulta, necesitamos los datos devueltos por otra consulta; así, si queremos obtener los datos de los empleados que tengan el mismo oficio que 'PEPE', hemos de averiguar el oficio de 'PEPE' (primera consulta). Una vez conocido este dato, podemos averiguar qué empleados tienen el mismo oficio que 'PEPE' (segunda consulta). Este problema se puede resolver usando una subconsulta, que no es más que una sentencia SELECT dentro de otra SELECT. Las subconsultas son aquellas sentencias SELECT que forman parte de una cláusula WHERE de una sentencia SELECT anterior. Una subconsulta consistirá en incluir una declaración SELECT como parte de una cláusula WHERE. El formato de una subconsulta es similar a este:

```
SELECT ...  
FROM ...  
WHERE columna operador_comparativo (SELECT...  
                                     FROM...  
                                     WHERE...);
```

La subconsulta (el comando SELECT entre paréntesis) se ejecutará primero y, posteriormente, el valor extraído es “introducido” en la consulta principal.

Ejercicio 10 relación complementaria

A. Condiciones de búsqueda en subconsultas

Las subconsultas suelen aparecer como parte de la condición de búsqueda de una cláusula WHERE o HAVING. Dichas condiciones pueden ser:

- Test de comparación en búsquedas (>, <, <>, <=, >=, =). Compara valor de una expresión con un valor único producido por una subconsulta. Ejemplo: apellidos de empleados cuyo oficio es igual a 'GIL':

```
SELECT apellido FROM emple WHERE oficio = (SELECT oficio FROM emple  
WHERE apellido='GIL');
```

- Test de pertenencia a un conjunto devuelto por una subconsulta (IN): Comprueba si el valor de una expresión coincide con uno del conjunto de

valores producidos en la subconsulta. Ejemplo: obtener aquellos apellidos de empleados cuyo oficio sea alguno de los oficios que hay en el departamento 20:

```
SELECT apellido FROM emple WHERE oficio IN (SELECT oficio FROM emple WHERE dept_no=20);
```

- Test de existencia (EXISTS, NOT EXISTS): examina si una subconsulta produce algunas filas de resultados. El test es TRUE si devuelve filas, si no es FALSE. Ejemplo: listar los departamentos que tengan empleados:

```
SELECT dnombre, dept_no FROM depart WHERE EXISTS (SELECT * FROM emple WHERE emple.dept_no = depart.dept_no)
```

Para calcular los que no tengan empleados se usa NOT EXISTS

- Test de comparación cuantificada (ANY y ALL): se usa en conjunción con los operadores (>, <, <>, <=, >=,=)

ANY: compara el valor de una expresión con cada uno del conjunto de valores producidos por una subconsulta, si alguna de las comparaciones individuales da como resultado TRUE, ANY devuelve TRUE, si la subconsulta no devuelve nada devolverá FALSE. Ejemplo: obtener los datos de los empleados cuyo salario sea igual a algún salario de los empleados del departamento 30:

```
SELECT * FROM emple WHERE salario = ANY (SELECT salario FROM emple WHERE dept_no = 30);
```

ALL: Compara el valor de una expresión con cada uno del conjunto de valores producidos por una subconsulta, si todas las comparaciones individuales da como resultado TRUE, ALL devuelve TRUE, en caso contrario devuelve FALSE. Ejemplo: Obtener los datos de los empleados cuyo salario sea menor a cualquier salario de los empleados del departamento 30:

```
SELECT * FROM emple WHERE salario < ANY (SELECT salario FROM emple WHERE dept_no = 30);
```

B. Subconsultas que generan valores simples

Se trata de aquellas subconsultas que devuelven una fila o un valor simple; en éstas se utilizan los operadores de comparación (>, <, <>, <=, >=,=). Si la subconsulta obtiene más de una fila, se produce un mensaje de error. Por ejemplo, con la siguiente

consulta se pretende obtener los apellidos de los empleados cuyo oficio coincida con algún oficio del departamento 20.

```
SELECT apellido FROM emple WHERE oficio = (SELECT oficio FROM emple WHERE dept_no=20);
```

En el departamento 20 hay varios oficios por tanto la subconsulta devuelve varias filas. Por ello, cuando una subconsulta devuelve más de una fila no se puede recurrir a los operadores de comparación.

C. Subconsultas que generan listas de valores

Son aquellas subconsultas que devuelven más de una fila o más de un valor. Cuando una subconsulta devuelva más de un valor, usaremos el operador IN en la cláusula WHERE. La solución al ejemplo anterior sería:

```
SELECT apellido FROM emple WHERE oficio IN (SELECT oficio FROM emple WHERE dept_no=20);
```

El tipo de dato de la expresión situada después de WHERE debe coincidir con el tipo de dato devuelto por la subconsulta.

Ejercicio 11 relación complementaria

D. Subconsultas correlacionadas

Una subconsulta correlacionada es aquella que hace referencia a una columna o varias columnas más externas. A veces la subconsulta hace uso de columnas que tienen el mismo nombre que las columnas de las tablas usadas en la consulta más externa. Si la subconsulta necesita acceder a esas columnas deberá definirse un alias en la tablas más externa. Ejemplo: deseamos obtener los datos de los empleados cuyo salario sea el máximo salario de su departamento:

```
SELECT * FROM emple e WHERE salario = (SELECT MAX(salario) FROM emple WHERE dept_no = e.dept_no);
```

La subconsulta devuelve para cada fila que se recupere de la consulta más externa el máximo salario del departamento que se está recuperando en la consulta

externa; para referenciar a dicho departamento se necesita el alias E usado en la tabla de la consulta externa.

COMBINACIÓN DE TABLAS

En las consultas realizadas solo se ha utilizado una tabla, indicada a la derecha de la palabra FROM; pero hay veces que una consulta necesita de varias tablas. En este caso, las tablas se expresarán a la derecha de la palabra FROM.

Sintaxis:

```
SELECT    columnas de las tablas citadas en la cláusula "from"
FROM      tabla1, tabla2, ...
WHERE     tabla1.columna = tabla2.columna;
```

Cuando combinamos varias tablas, hemos de tener en cuenta una serie de reglas:

- Es posible unir tantas tablas como deseemos
- En la cláusula SELECT se pueden citar columnas de todas las tablas
- Si hay columnas con el mismo nombre en las distintas tablas de la cláusula FROM, se deben identificar, especificando "NombreTabla.NombreColumna"
- Si el nombre de una columna existe sólo en una tabla, no será necesario especificar como "NombreTabla.NombreColumna". Sin embargo, hacerlo mejoraría la legibilidad de la sentencia SELECT
- El criterio que se siga para combinar las tablas ha de especificar en la cláusula WHERE. Si se omite esta cláusula, que especifica la condición de combinación, el resultado será un PRODUCTO CARTERIANO, que emparejará todas las filas de una tabla con cada fila de la otra.

Ejercicio 12 relación complementaria