Carrera: Técnico Superior en Programación
Materia: Laboratorio de Computación III
Tema: Consultas de selección - Parte 4

Consultas de selección - Parte 4

Consultas anidadas

En los apuntes de consultas de selección anteriores fuimos aumentando la dificultad de las consultas de manera progresiva. Al principio comenzamos obteniendo todos los registros de una tabla, luego filtrando o agregando columnas de la tabla, después aprendimos a filtrar la cantidad de filas que se obtienen mediante condiciones.
Luego se explicó como obtener datos de más de una tabla mediante el uso de las consultas JOIN y por último cómo resumir los datos de una tabla mediante el uso de las funciones de agregado y la cláusula GROUP BY.

En la última parte del apunte de consultas de selección veremos el uso de consultas anidadas o comúnmente llamadas subconsultas. La idea principal aquí es que los datos de la consulta principal se obtienen de una o más consultas secundarias.

Éstas consultas secundarias podrían entenderse como 'tablas virtuales' en el sentido de que el listado de datos que se obtiene se elabora en el momento (suele decirse al vuelo -*on the fly*-).

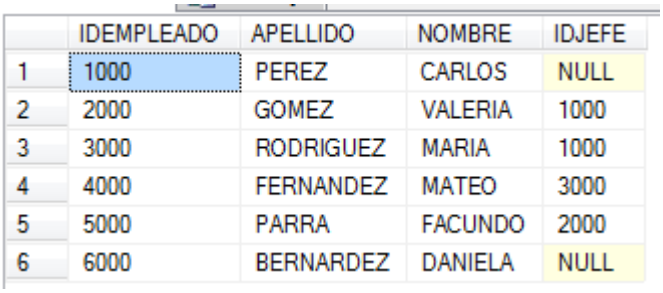
La mayoría de las veces, las subconsultas son utilizadas para evitar el uso de JOINS o intentar simplificar una consulta que de otra manera sería muy compleja. En otros casos, es la única manera de obtener un grupo de datos sin la necesidad de procesarlos con elementos de programación.

Veamos una serie de ejemplos utilizando algunos casos prácticos:

Utilizaremos la siguiente base de datos, cabe destacar que la siguiente BBDD está compuesta sólo por una tabla empleados. Las columnas son IDEMPLEADO como clave primaria, APELLIDO y NOMBRE como campos de texto y IDJEFE como clave foránea que acepta nulos. Esto quiere decir que:

- La columna IDJEFE es clave foránea que hace referencia a la misma tabla pero en el campo IDEMPLEADO.
- Al aceptar NULL pero ser FK, la columna IDJEFE deberá contener un valor que se relacione con algún registro de la misma tabla pero en la columna IDEMPLEADO. A menos que se ingrese NULL.
- Si se ingresa un valor en la columna IDJEFE se indica cuál es el jefe de dicho empleado. El jefe debe ser un empleado registrado en la tabla empleados. Sin embargo, si no se ingresa un valor relacionado en la columna IDJEFE, es decir, si se ingresa NULL. Entonces se estaría representando un empleado que no tiene ningún jefe relacionado.

Antes de comenzar con los ejemplos, veamos los valores que contiene la tabla EMPLEADOS:



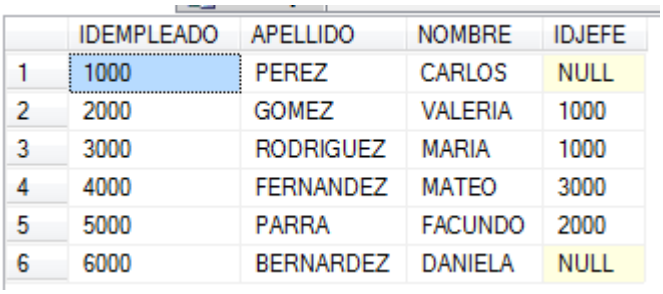
	IDEMPLEADO	APELLIDO	NOMBRE	IDJEFE
1	1000	PEREZ	CARLOS	NULL
2	2000	GOMEZ	VALERIA	1000
3	3000	RODRIGUEZ	MARIA	1000
4	4000	FERNANDEZ	MATEO	3000
5	5000	PARRA	FACUNDO	2000
6	6000	BERNARDEZ	DANIELA	NULL

Supongamos que debemos obtener los siguientes datos de la siguiente tabla:

- Obtener todos los jefes de la empresa
Antes que nada, lo primero que viene a la mente es obtener todos los registros cuyo IDJEFE sea NULL. Sin embargo, esto no significa que sean jefes sino que no tienen jefe asignado. Por lo tanto, esa solución no es correcta.
La única forma de saber si un empleado es jefe de otro es evaluando el IDJEFE. Cada empleado cuyo IDEMPLEADO figure como IDJEFE de algún otro debe ser considerado como jefe en nuestro listado.

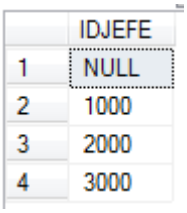
Lo primero que hay que hacer ante una situación como ésta es separar la consulta en pasos:

1. Obtener el listado completo sin filtrar
```
SELECT * FROM EMPLEADOS
```



	IDEMPLEADO	APELLIDO	NOMBRE	IDJEFE
1	1000	PEREZ	CARLOS	NULL
2	2000	GOMEZ	VALERIA	1000
3	3000	RODRIGUEZ	MARIA	1000
4	4000	FERNANDEZ	MATEO	3000
5	5000	PARRA	FACUNDO	2000
6	6000	BERNARDEZ	DANIELA	NULL

2. Obtener el listado de los códigos de empleado de aquellos empleados que sean jefes
```
SELECT DISTINCT IDJEFE FROM EMPLEADOS
```



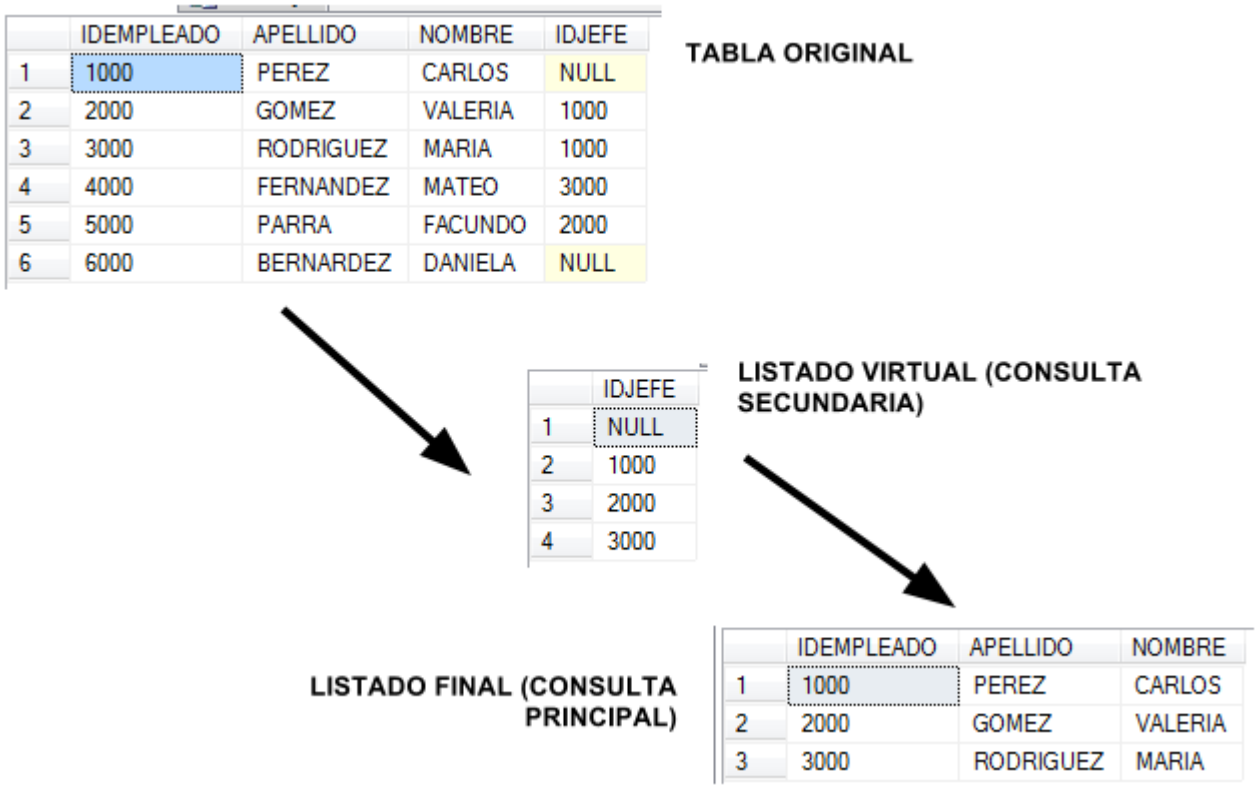
	IDJEFE
1	NULL
2	1000
3	2000
4	3000

3. Obtener, mediante el uso de una subconsulta, los datos de los empleados que sean jefes.
```
SELECT E.IDEMPLEADO, E.APELLIDO, E.NOMBRE FROM EMPLEADOS E WHERE E.IDEMPLEADO
IN (SELECT DISTINCT IDJEFE FROM EMPLEADOS)
```

	IDEMPLEADO	APELLIDO	NOMBRE
1	1000	PEREZ	CARLOS
2	2000	GOMEZ	VALERIA
3	3000	RODRIGUEZ	MARIA

Como vimos en el paso a paso, lo primero que observamos en el listado completo es que debemos obtener (sin repeticiones) todos los valores almacenados en las celdas IDJEFE. Luego, si hacemos un SELECT sólo de los IDJEFE, tenemos el IDEMPLEADO de todos aquellos empleados que sean jefes, pero a esto le falta incorporar los datos de nombre y apellido.
Ahí es donde se necesita aplicar el uso de consultas anidadas. Obtener todos los datos de la tabla empleados siempre que el IDEMPLEADO se encuentre dentro del listado de IDEMPLEADOS de aquellos que sean jefes. Se hace uso del operador IN en el WHERE ya que el resultado de la subconsulta es una lista de datos. Nos debemos asegurar que dicha subconsulta sólo tenga una columna y que sea compatible con el dato que queremos comparar. Entiendase que los resultados de esa lista se traducirían como si se escribiera `WHERE IDEMPLEADO IN (NULL, 1000, 2000, 3000)`.

Trabalenguas aparte, podemos observar como el listado de códigos de jefe es el resultado de una subconsulta o consulta secundaria, la misma se usa de manera temporal para aportar información de la consulta principal.



- Obtener para cada empleado, el apellido y el nombre de su jefe

```
SELECT E.IDEMPLEADO, E.APELLIDO +', '+ E.NOMBRE AS 'APENOM', (SELECT
AUX.APELLIDO + ', ' + AUX.NOMBRE FROM EMPLEADOS AS AUX WHERE AUX.IDEMPLEADO =
E.IDJEFE) AS 'JEFE APENOM' FROM EMPLEADOS AS E
```

	IDEMPLEADO	APENOM	JEFE APENOM
1	1000	PEREZ, CARLOS	NULL
2	2000	GOMEZ, VALERIA	PEREZ, CARLOS
3	3000	RODRIGUEZ, MARIA	PEREZ, CARLOS
4	4000	FERNANDEZ, MATEO	RODRIGUEZ, MARIA
5	5000	PARRA, FACUNDO	GOMEZ, VALERIA
6	6000	BERNARDEZ, DANIELA	NULL

El ejemplo anterior demuestra como se puede utilizar una subconsulta no sólo como sección del WHERE sino también como una columna del listado principal. Para ello debemos asegurarnos que el resultado de la consulta sea exactamente una fila y una columna. Comúnmente conocido como un dato escalar.
En este caso, para cada registro de la tabla empleados del listado principal (llamada con el alias E) debemos obtener el apellido y nombre de la misma tabla (llamada AUX) pero sólo del registro cuyo IDEMPLEADO sea igual al valor que tenga el registro de la tabla E en el campo IDJEFE.

- Obtener el apellido y nombre del empleado que tenga mayor cantidad de empleados a cargo

```
SELECT TOP 1 AUX.APENOM FROM   AS AUX ORDER BY AUX.CANT DESC
```

	APENOM
1	PEREZ, CARLOS

En la consulta anterior, podemos ver como tenemos dos niveles de subconsultas que hacen a la consulta final. En principio notamos como debemos de contar la cantidad de veces que un empleado es jefe. Luego a dicho cálculo lo asociamos a su apellido y nombre. Se puede observar como el apellido y nombre surge de la tabla con el alias E mientras que el cálculo de la cantidad de veces que dicho empleado es jefe se obtiene a partir de la tabla con alias E2.
Luego la combinación de esas subconsultas forman un listado donde por cada empleado se obtiene el apellido y nombre y la cantidad de personas que tiene a cargo.
Dicho listado es utilizado como fuente de datos (FROM) para realizar la última consulta de selección en la que se obtiene Apellido y nombre del primer registro ordenado de manera descendiente por la columna CANT que representa la cantidad de personas que tiene a cargo.
De esta manera obtenemos una consulta que obtiene sus datos a partir del resultado de otras consultas secundarias o tablas virtuales.