



Trabajo con datos

Consultas combinadas DML



Consultas combinadas DML

El lenguaje DML nos permite realizar la manipulación de los datos que se encuentran en las tablas de acuerdo con instrucciones específicas y con un propósito definido; pero, en este caso, abordaremos algunas de ellas que son mucho más precisas y permiten llegar a un nivel de profundidad y comprensión mucho más complejo.

1 inner join

La instrucción INNER JOIN en cualquier cláusula FROM.

Este es el tipo de combinación más común.

Las combinaciones internas combinan los registros de dos tablas si hay valores coincidentes en un campo común de las dos tablas.

Se puede usar INNER JOIN con las tablas Departamentos y Empleados para seleccionar todos los empleados de cada departamento. En cambio, para seleccionar todos los departamentos (aunque algunos no tengan ningún empleado asignado) o todos los empleados (aunque algunos no estén asignados a un departamento).

Puede combinar dos campos numéricos de tipos similares. Por ejemplo, puede combinar en campos Autonumeración y Largo, porque son tipos similares. Pero no puede combinar los tipos de campo Sencillo y Doble.

2 left outer join

La cláusula Left Join en SQL permite consultar datos de varias tablas. Devuelve todas las filas de la tabla de la izquierda y las filas coincidentes de la tabla de la derecha.

Si no se encuentran filas que coincidan en la tabla de la derecha, se utiliza NULL.

Otras cláusulas que se pueden utilizar para hacer combinaciones entre tablas.

A continuación, se muestra cómo unir dos tablas T1 y T2, mediante la cláusula Left Join en SQL:

```
SELECT
    select_list
FROM
    T1
LEFT JOIN T2 ON
    join_predicate;
```

En esta sintaxis, T1 y T2 son las tablas izquierda y derecha, respectivamente. Para cada fila de la tabla T1 la consulta las compara con todas las filas de la tabla T2.

Si un par de filas hacen que el predicado de combinación se evalúa como VERDADERO, los valores de columna de estas filas se combinarán para formar una nueva fila, que luego se incluirá en el conjunto de resultados.

Si una fila de la tabla de la izquierda (T1), no tiene ninguna fila coincidente de la tabla T2, la consulta combina los valores de columna de la fila de la tabla de la izquierda con NULL para los valores de cada columna de la tabla de la derecha.

En resumen, la cláusula Left Join en SQL, devuelve todas las filas de la tabla izquierda (T1) y las filas coincidentes o valores NULL de la tabla derecha (T2).

3 right outer join

Si está uniendo dos tablas y desea que el conjunto de resultados incluya filas no coincidentes de una sola tabla, utilice una cláusula LEFT OUTER JOIN o una cláusula RIGHT OUTER JOIN. La coincidencia se basa en la condición de unión.

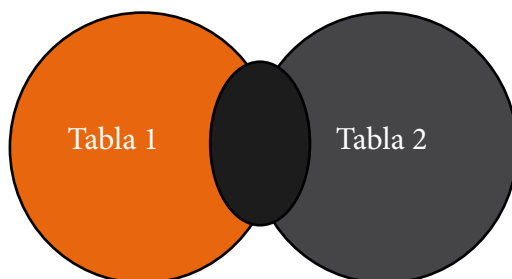
La cláusula RIGHT OUTER JOIN, incluye filas de la tabla, que se especifican después de RIGHT OUTER JOIN que no tienen valores coincidentes en la tabla que se especifica antes de RIGHT OUTER JOIN. Como en una combinación interna, la condición de combinación puede ser cualquier condición de búsqueda simple o compuesta que no contenga una referencia de subconsulta.

4 full outer join

La palabra clave FULL OUTER JOIN devuelve todos los registros cuando hay una coincidencia en los registros de la tabla izquierda (tabla1) o derecha (tabla2); es decir, todos los registros que se encuentren en coincidencia serán traídos y mostrados al usuario; a continuación, veremos una ilustración al respecto.

```
SELECT column_name(s)
FROM table1
FULL OUTER JOIN table2
ON table1.column_name = table2.column_name
WHERE condition;
```

Como se puede observar en el código de ejemplo, la tabla 1 es la que se referencia dentro de la cláusula FROM porque es donde se encuentran los datos que luego se desean comparar; luego se utiliza la cláusula FULL OUTER JOIN para comparar los campos que tienen esta coincidencia dentro de la tabla 2; si lo observáramos de manera visual, tendríamos lo siguiente:



La cantidad de combinaciones que esto puede generar son bastantes, debido a que se buscarían muchas coincidencias encontradas en la tabla 1 que tengan valores compartidos en la tabla 2; es por eso que cuando se utiliza este tipo de operaciones es cuando se desea realizar recorridos que generen más de una coincidencia, con respecto a los datos almacenados; es decir, se combinan demasiadas veces los datos en los que no se puede, en ocasiones, conocer la magnitud de los resultados, pero es muy eficiente si deseamos que la combinación de estos valores puedan arrojar resultados mucho más precisos, pero teniendo en cuenta que también podrían retornar valores vacíos si no encuentra la coincidencia exacta a evaluar.

5 cross join

Esta instrucción retorna todos los registros de todas las tablas implicadas en la unión, devuelve el producto cartesiano. No es muy utilizado.

Un pequeño restaurante tiene almacenados los nombres y los precios de sus comidas en una tabla llamada “comidas” y en una tabla denominada “postres”, los mismos datos de sus postres.

A continuación, veremos un ejemplo teniendo en cuenta la siguiente estructura:

	codigo	nombre	rubro	precio
▶	1	1	milanesa y fritas	3,4
	2	2	arroz primavera	2,5
	3	3	pollo	2,8
*	NULL	NULL	NULL	NULL

Entidad comidas

	codigo	nombre	precio
▶	1	flan	1
	2	porcion de torta	2,1
	3	gelatina	0,9
*	NULL	NULL	NULL

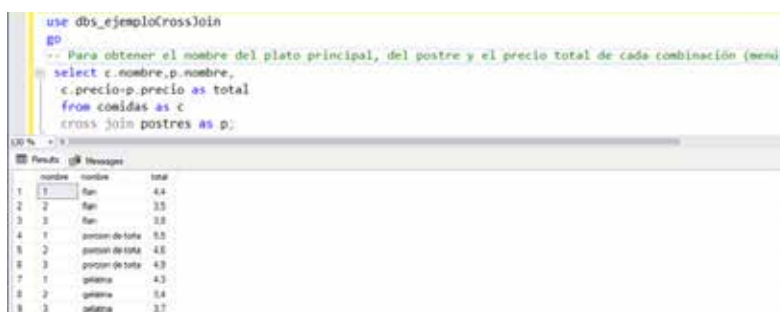
Entidad postres

A continuación, aplicaremos algunas combinaciones con el comando CROSS JOIN para ver los resultados aplicados.

CROSS JOIN para obtener el producto cartesiano de ambas tablas:

```
select c.*,p.*
from comidas as c
cross join postres as p;
```

Resultado arrojado



	nombre	nombre	total
1	1	flan	4,4
2	2	flan	3,5
3	3	flan	3,8
4	1	porcion de torta	5,5
5	2	porcion de torta	4,6
6	3	porcion de torta	4,9
7	1	gelatina	4,3
8	2	gelatina	3,4
9	3	gelatina	3,7

Como se puede observar en las imágenes, las combinaciones que este comando genera son mucho más detalladas que las que entregaría un JOIN normal o una consulta normal, porque este tipo de comandos llega a realizar, de manera mucho más profunda, las combinaciones que se requieren en los datos almacenados