

Consulta en múltiples tablas

Objetivos

- ¿Por qué debemos hacer consultas en múltiples tablas?
- INNER JOIN
- FULL JOIN
- CROSS JOIN

En SQL, las consultas que involucran múltiples tablas son comunes y necesarias para obtener información más completa y específica de una base de datos. Cuando una base de datos contiene varias tablas, a menudo es necesario combinar la información de estas tablas para obtener resultados significativos. Esto se logra mediante el uso de instrucciones SQL llamadas "JOIN" que permiten unir filas de una tabla con filas de otra basándose en una condición específica.

Las instrucciones que veremos en esta unidad serán las siguientes

- INNER JOIN
- LEFT JOIN
- FULL JOIN
- CROSS JOIN

INNER JOIN

El INNER JOIN es una operación fundamental en SQL que permite combinar filas de dos tablas en función de una condición de coincidencia específica. La esencia de un INNER JOIN radica en su capacidad para seleccionar solo las filas que tienen correspondencias en ambas tablas, eliminando aquellas que no cumplen con la condición establecida.

Sintaxis Básica:

```
SELECT columnas FROM tabla1 INNER JOIN tabla2 ON tabla1.columna = tabla2.columna;
```

Explicación de la consulta anterior

- SELECT columnas: Indica las columnas que deseas recuperar en el resultado de la consulta.
- FROM tabla1: Especifica la primera tabla de la cual deseas seleccionar datos.

- INNER JOIN tabla2: Indica la segunda tabla que deseas combinar con la primera.
- ON tabla1.columna = tabla2.columna: Define la condición de coincidencia entre las dos tablas.

¿Cómo funciona?

Supongamos que tenemos dos tablas, A y B. Cuando aplicamos un INNER JOIN entre estas tablas, se comparan las filas de ambas tablas en base a una columna específica (definida por la condición ON). Solo las filas que cumplen con esta condición se incluyen en el resultado final. Las filas que no tienen coincidencias en ambas tablas son descartadas.

Caso de uso

Imaginemos una base de datos que contiene información sobre clientes en una tabla y detalles de los pedidos en otra. Al utilizar un INNER JOIN entre estas tablas en función del identificador del cliente, podemos obtener información detallada sobre los clientes y los productos que han comprado.

```
SELECT clientes.nombre, clientes.apellido, pedidos.producto FROM clientes INNER JOIN pedidos ON clientes.cliente_id = pedidos.cliente_id;
```

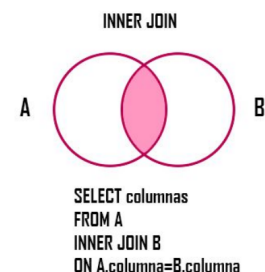
Este ejemplo nos proporciona una lista de clientes junto con los productos que han comprado, excluyendo aquellos clientes que no han realizado ningún pedido.

Un ejemplo mas visual

INNER JOIN selecciona las filas que hacen match en ambas columnas.

id	title	content
1	Artículo 1	LOREM IPSUM ET ...
2	Artículo 2	LOREM IPSUM ET ...

content	post_id	id	title	content
comentario 1	1	1	Artículo 1	LOREM IPSUM ET ...
comentario 2		1	Artículo 1	LOREM IPSUM ET ...
comentario 1		2	Artículo 2	LOREM IPSUM ET ...



¿Qué es un LEFT JOIN?

El LEFT JOIN, también conocido como LEFT OUTER JOIN, es una operación que combina filas de dos tablas en base a una condición específica, priorizando todas las filas de la tabla izquierda y trayendo las coincidencias de la tabla derecha. Si no hay coincidencias en la tabla derecha, se devuelven valores NULL.

Sintaxis Básica:

```
SELECT columnas FROM tabla1 LEFT JOIN tabla2 ON tabla1.columna = tabla2.columna;
```

Explicación de la consulta anterior

- **SELECT columnas:** Especifica las columnas que deseas recuperar en el resultado de la consulta.
- **FROM tabla1:** Indica la primera tabla de la cual deseas seleccionar datos.
- **LEFT JOIN tabla2:** Especifica la segunda tabla que deseas combinar con la primera.
- **ON tabla1.columna = tabla2.columna:** Define la condición de coincidencia entre las dos tablas.

¿Cómo funciona?

Al aplicar un LEFT JOIN, todas las filas de la tabla izquierda se incluirán en el resultado final, independientemente de si tienen coincidencias en la tabla derecha o no. Aquellas filas que no tienen coincidencias en la tabla derecha tendrán valores NULL en las columnas correspondientes.

Caso de uso

Supongamos que tenemos una tabla de empleados y otra de proyectos. Queremos obtener una lista de todos los empleados y los proyectos en los que están trabajando, incluso si algunos empleados no están asignados a ningún proyecto.

```
SELECT empleados.nombre, empleados.apellido, proyectos.nombre AS proyecto  
FROM empleados  
LEFT JOIN asignaciones ON empleados.empleado_id = asignaciones.empleado_id  
LEFT JOIN proyectos ON asignaciones.proyecto_id = proyectos.proyecto_id;
```

En este ejemplo, obtenemos una lista de empleados junto con los nombres de los proyectos en los que están trabajando. Si un empleado no tiene asignado ningún proyecto, aún así se incluirá en el resultado con el nombre del proyecto como NULL.

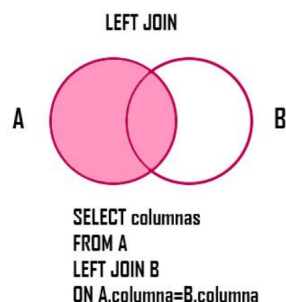
Un ejemplo más visual

LEFT JOIN selecciona todas las filas de la primera tabla y se unifica con la segunda siempre y cuando haya match

id	title	content
1	Artículo 1	LOREM IPSUM ET ...
2	Artículo 2	LOREM IPSUM ET ...

content	post_id	id	title	content
comentario 1	1	1	Artículo 1	LOREM IPSUM ET ...
comentario 2		1	Artículo 1	LOREM IPSUM ET ...
comentario 1		2	Artículo 2	LOREM IPSUM ET ...

LEFT JOIN



FULL JOIN

El FULL JOIN combina filas de dos tablas basándose en una condición de coincidencia, incluyendo todas las filas de ambas tablas en el resultado final. Si no hay coincidencias en una de las tablas, se llenarán las columnas correspondientes con valores NULL.

Sintaxis Básica:

```
SELECT columnas FROM tabla1 FULL JOIN tabla2 ON tabla1.columna = tabla2.columna;
```

Explicación de la consulta anterior

- **SELECT columnas:** Especifica las columnas que deseas recuperar en el resultado de la consulta.
- **FROM tabla1:** Indica la primera tabla de la cual deseas seleccionar datos.
- **FULL JOIN tabla2:** Especifica la segunda tabla que deseas combinar con la primera.
- **ON tabla1.columna = tabla2.columna:** Define la condición de coincidencia entre las dos tablas.

¿Cómo funciona?

El FULL JOIN devuelve todas las filas de ambas tablas, combinando las filas que cumplen con la condición de coincidencia y llenando con valores NULL en las columnas donde no hay coincidencias.

Caso practico

Supongamos que tenemos una tabla de estudiantes y otra de cursos. Queremos obtener una lista de todos los estudiantes y los cursos que están cursando, mostrando NULL en las columnas de curso cuando un estudiante no esté matriculado en ningún curso.

```
SELECT estudiantes.nombre, estudiantes.apellido, cursos.nombre AS curso
FROM estudiantes
FULL JOIN matriculas ON estudiantes.estudiante_id = matriculas.estudiante_id
FULL JOIN cursos ON matriculas.curso_id = cursos.curso_id;
```

En este ejemplo, obtenemos una lista de todos los estudiantes junto con los cursos en los que están matriculados. Si un estudiante no está matriculado en ningún curso o si hay un curso sin estudiantes matriculados, aún así se incluirán en el resultado con los valores correspondientes como NULL.

CROSS JOIN

El CROSS JOIN es una operación que combina cada fila de la primera tabla con cada fila de la segunda tabla, generando todas las combinaciones posibles entre ambas. A diferencia de otras operaciones de JOIN, no requiere una condición de coincidencia.

Sintaxis:

```
SELECT columnas FROM tabla1 CROSS JOIN tabla2;
```

¿Cómo funciona?

El CROSS JOIN crea todas las combinaciones posibles entre las filas de ambas tablas, multiplicando el número de filas de una tabla por el número de filas de la otra.

Caso de uso

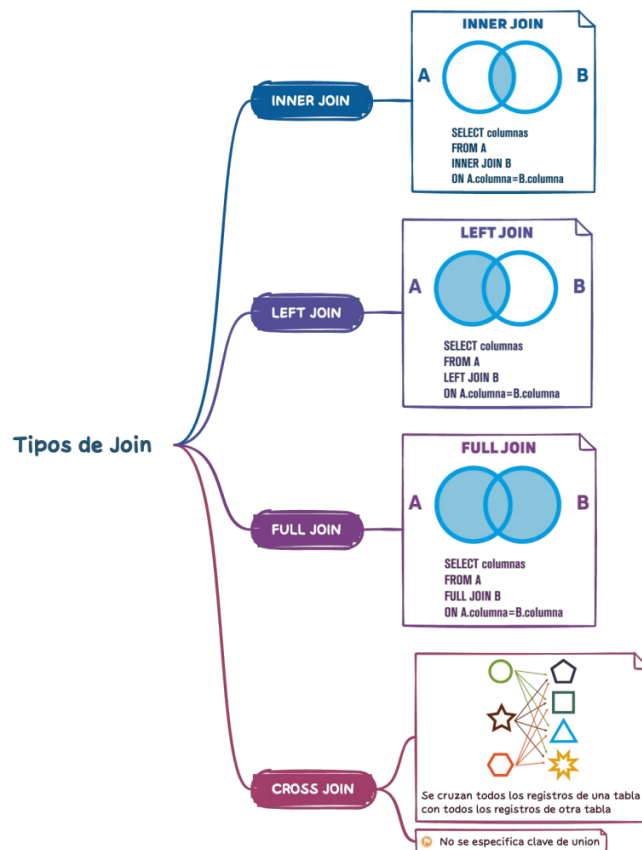
Supongamos que tenemos una tabla de empleados y otra de proyectos. Queremos obtener una lista que contenga todas las combinaciones posibles entre empleados y proyectos, mostrando cada empleado emparejado con cada proyecto.

```
SELECT empleados.nombre, empleados.apellido, proyectos.nombre AS proyecto
FROM empleados
CROSS JOIN proyectos;
```

En este ejemplo, obtenemos una lista que incluye a cada empleado emparejado con cada proyecto, generando todas las combinaciones posibles.

Importante: Ten en cuenta que el CROSS JOIN puede generar un número significativo de filas en el resultado, especialmente si las tablas involucradas son grandes. Es esencial utilizar esta operación con precaución y solo en situaciones donde se requieren todas las combinaciones posibles entre las filas de ambas tablas.

Diagrama de JOINS



Resumen

INNER JOIN:

Cuándo usarlo: Utiliza un INNER JOIN cuando solo estés interesado en las filas que tienen coincidencias en ambas tablas.

LEFT JOIN (o LEFT OUTER JOIN):

Cuándo usarlo: Emplea un LEFT JOIN cuando quieras todas las filas de la tabla izquierda y las filas coincidentes de la tabla derecha. Si algunas filas en la tabla derecha no tienen coincidencias, se mostrarán con valores NULL.

FULL JOIN (o FULL OUTER JOIN):

Cuándo usarlo: Emplea un FULL JOIN cuando quieras todas las filas de ambas tablas, mostrando coincidencias donde las haya y llenando con valores NULL donde no haya coincidencias.

CROSS JOIN:

Cuándo usarlo: Utiliza un CROSS JOIN cuando desees generar todas las combinaciones posibles entre las filas de dos tablas, sin ninguna condición de coincidencia específica.