Unidad Didáctica 5: Consultas Multitabla en MySQL

1. Introducción a Consultas Multitabla

Las bases de datos relacionales en MySQL suelen organizar la información en múltiples tablas interconectadas mediante claves primarias y foráneas. Las consultas multitabla permiten extraer y combinar datos de varias tablas simultáneamente, lo que facilita un análisis más completo y detallado de la información almacenada.

Las consultas multitabla son fundamentales para obtener relaciones entre datos, como por ejemplo, obtener una lista de clientes con sus pedidos, o mostrar productos con sus categorías.

---

2. Conceptos Fundamentales

Relaciones Entre Tablas

En MySQL, las relaciones entre tablas se basan en claves:

- **Clave Primaria (`PRIMARY KEY`)**: Identificador único para cada fila en una tabla.
- **Clave Foránea (`FOREIGN KEY`)**: Referencia a una clave primaria de otra tabla.

Ejemplo de Relación:

- Tabla `clientes`: Almacena datos de clientes (clave primaria: `id`).
- Tabla `pedidos`: Almacena los pedidos realizados (clave primaria: `id`, clave foránea: `id_cliente`).

---

3. Tipos de Consultas Multitabla

3.1. `JOIN` – Unión de Tablas

La instrucción `JOIN` permite combinar registros de dos o más tablas en función de columnas relacionadas.

Tipos de `JOIN`:

- `INNER JOIN` – Devuelve solo las filas donde hay coincidencias en ambas tablas.
- `LEFT JOIN` **(o** `LEFT OUTER JOIN`**)** – Devuelve todas las filas de la tabla izquierda y las coincidencias de la tabla derecha. Si no hay coincidencias, se devuelven `NULL`.
- `RIGHT JOIN` **(o** `RIGHT OUTER JOIN`**)** – Devuelve todas las filas de la tabla derecha y las coincidencias de la tabla izquierda. Si no hay coincidencias, se devuelven `NULL`.
- `FULL JOIN` **(no soportado directamente por MySQL)** – Devuelve todas las filas cuando hay coincidencias en alguna de las tablas.

---

4. Sintaxis y Ejemplos de `JOIN`

4.1. `INNER JOIN` (Unión Interna)

Selecciona registros que tienen coincidencias en ambas tablas.
Sintaxis:

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

Ejemplo:

```
SELECT clientes.nombre, pedidos.fecha
FROM clientes
INNER JOIN pedidos ON clientes.id = pedidos.id_cliente;
```

Esto devuelve una lista de clientes con sus pedidos correspondientes.

4.2. `LEFT JOIN` (Unión Externa Izquierda)

Devuelve todos los registros de la tabla izquierda, incluso si no hay coincidencias en la tabla derecha.
Sintaxis:

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

Ejemplo:

```
SELECT clientes.nombre, pedidos.fecha
FROM clientes
LEFT JOIN pedidos ON clientes.id = pedidos.id_cliente;
```

Esto muestra todos los clientes, incluidos aquellos que no han realizado pedidos (con valores `NULL` en `pedidos`).

4.3. `RIGHT JOIN` (Unión Externa Derecha)

Devuelve todos los registros de la tabla derecha, incluso si no hay coincidencias en la tabla izquierda.
Sintaxis:

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

Ejemplo:

```
SELECT pedidos.id, clientes.nombre
FROM clientes
RIGHT JOIN pedidos ON clientes.id = pedidos.id_cliente;
```

Esto muestra todos los pedidos, incluidos aquellos que no tienen cliente asignado.

---

5. `CROSS JOIN` (Producto Cartesiano)

El `CROSS JOIN` devuelve el producto cartesiano de dos tablas, combinando todas las filas de la primera tabla con todas las filas de la segunda.

Sintaxis:

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

Ejemplo:

```
SELECT productos.nombre, categorias.nombre
FROM productos
CROSS JOIN categorias;
```

Esto genera todas las combinaciones posibles de productos y categorías.

---

6. `SELF JOIN` (Auto-Unión de Tablas)

`SELF JOIN` permite unir una tabla consigo misma. Es útil para comparar filas dentro de la misma tabla.

Ejemplo – Empleados y sus Gerentes:

```
SELECT e.nombre AS empleado, g.nombre AS gerente
FROM empleados e
INNER JOIN empleados g ON e.id_gerente = g.id;
```

7. Uso de Alias en Consultas Multitabla

El uso de alias (`AS`) simplifica la escritura y lectura de consultas multitabla, especialmente cuando se manejan múltiples uniones.

Ejemplo con Alias:

```
SELECT c.nombre AS cliente, p.fecha AS fecha_pedido
FROM clientes AS c
INNER JOIN pedidos AS p ON c.id = p.id_cliente;
```

8. Combinación de `JOIN` con Funciones de Agregación

`JOIN` se puede utilizar junto con `GROUP BY` y funciones de agregación (`SUM()`, `COUNT()`, `AVG()`).

Ejemplo – Total de pedidos por cliente:

```
SELECT clientes.nombre, COUNT(pedidos.id) AS total_pedidos
FROM clientes
LEFT JOIN pedidos ON clientes.id = pedidos.id_cliente
GROUP BY clientes.id;
```

9. `UNION` – Combinar Resultados de Varias Consultas

`UNION` permite combinar resultados de dos o más consultas `SELECT`.

- **`UNION`** – Elimina duplicados.
- **`UNION ALL`** – Mantiene duplicados.

Ejemplo – Clientes de dos ciudades diferentes:

```
SELECT nombre FROM clientes WHERE ciudad = 'Madrid'
UNION
SELECT nombre FROM clientes WHERE ciudad = 'Barcelona';
```

10. Buenas Prácticas en Consultas Multitabla

- **Optimizar el uso de `JOIN`**: Evitar uniones innecesarias y asegurarse de tener índices en las columnas utilizadas en las condiciones de unión.

- **Filtrado eficiente**: Aplicar condiciones `WHERE` antes de `JOIN` para reducir la cantidad de registros que se unirán.
- **Alias descriptivos**: Usar alias claros y significativos para mejorar la legibilidad.
- **Documentación**: Comentar consultas complejas para facilitar el mantenimiento futuro.