

Consultas Avanzadas en SQL

Principales técnicas y conceptos

1. Subconsultas (Subqueries)

Una subconsulta es una consulta dentro de otra:

- Subconsulta en SELECT, WHERE o FROM.
- Subconsulta correlacionada: Se refiere a columnas de la consulta principal.

Ejemplo:

```
SELECT nombre, salario
FROM empleados
WHERE salario > (SELECT AVG(salario) FROM empleados);
```

2. Consultas con Joins complejos

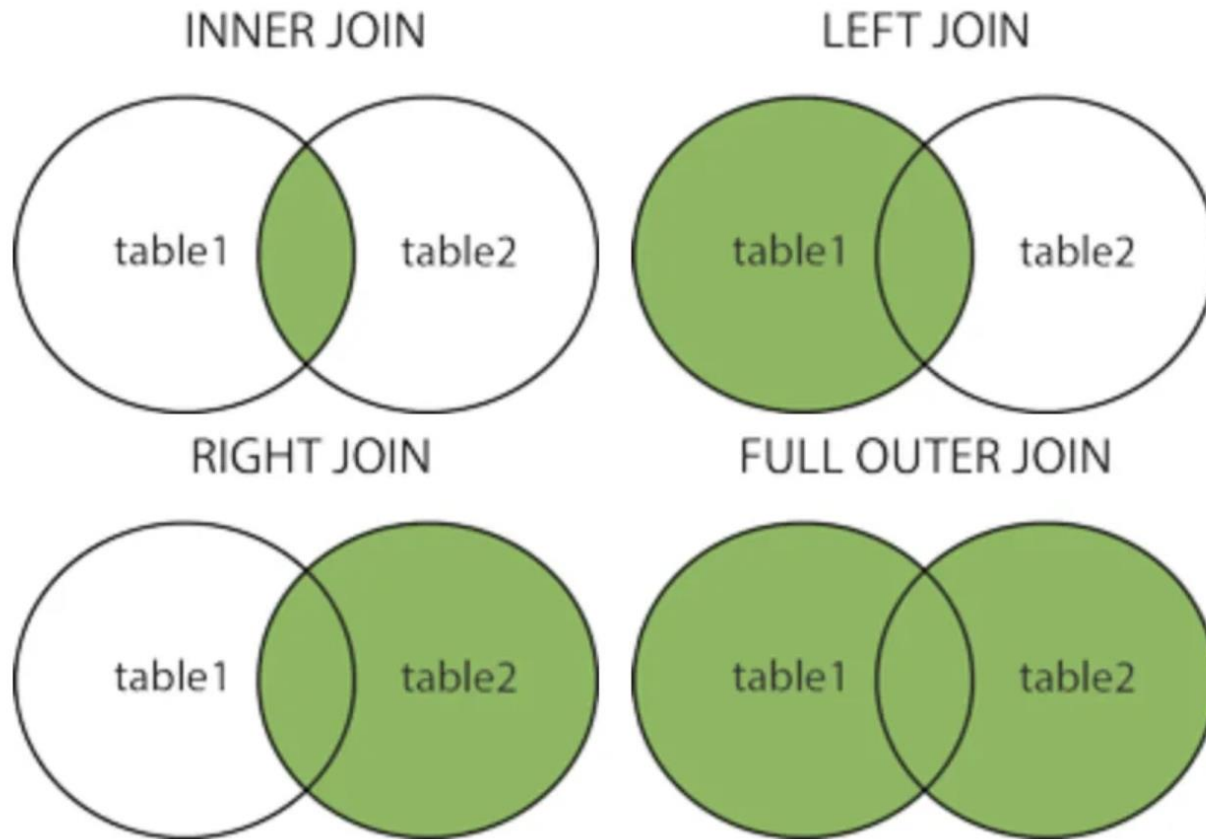
Permiten combinar filas de dos o más tablas:

- INNER JOIN: Coincidencias en ambas tablas.
- LEFT JOIN: Devuelve todas las filas de la tabla de la izquierda.
- RIGHT JOIN: Todas las filas de la tabla derecha.

Ejemplo:

```
SELECT e.nombre, d.nombre AS departamento
FROM empleados e
LEFT JOIN departamentos d ON e.departamento_id = d.id;
```

2. Consultas con Joins complejos



3. Funciones de Ventana (Window Functions)

Permiten realizar cálculos en un conjunto de filas relacionadas:

- ROW_NUMBER(): Asigna un número de fila.
- RANK(): Asigna un rango basado en valores equivalentes.

Ejemplo:

```
SELECT nombre, salario, RANK() OVER (ORDER BY salario DESC) AS salario_rango  
FROM empleados;
```

4. Consultas Recursivas con CTE

Permiten ejecutar consultas jerárquicas como relaciones de empleados y gerentes:

- Se usa la cláusula WITH RECURSIVE.

Ejemplo:

```
WITH RECURSIVE EmpleadoJerarquia AS (  
    SELECT id, nombre, gerente_id  
    FROM empleados  
    WHERE gerente_id IS NULL  
    UNION ALL  
    SELECT e.id, e.nombre, e.gerente_id  
    FROM empleados e  
    INNER JOIN EmpleadoJerarquia ej ON e.gerente_id = ej.id  
)  
SELECT * FROM EmpleadoJerarquia;
```

5. Operadores de Conjunto

- **UNION**: Combina resultados eliminando duplicados.
- **UNION ALL**: Combina todos los resultados, incluidas duplicadas.
- **INTERSECT**: Devuelve filas comunes en ambas consultas.

```
SELECT nombre FROM empleados_2023  
UNION ALL  
SELECT nombre FROM empleados_2024;
```

6. Cláusulas avanzadas de agrupamiento

- **GROUP BY CUBE:** Todas las combinaciones de grupos.
- **GROUP BY ROLLUP:** Subtotales y totales generales.

Ejemplo:

```
SELECT departamento, año, SUM(ventas)
FROM ventas
GROUP BY ROLLUP(departamento, año);
```


7. Particionamiento con PARTITION BY

Permite aplicar funciones agregadas dentro de particiones.

Ejemplo:

```
SELECT nombre, departamento, salario,  
       AVG(salario) OVER (PARTITION BY departamento) AS salario_promedio_departam  
FROM empleados;
```

8. Consultas con índices y optimización

- Uso de índices para acelerar consultas.
- Evitar `SELECT *` para reducir el volumen de datos.