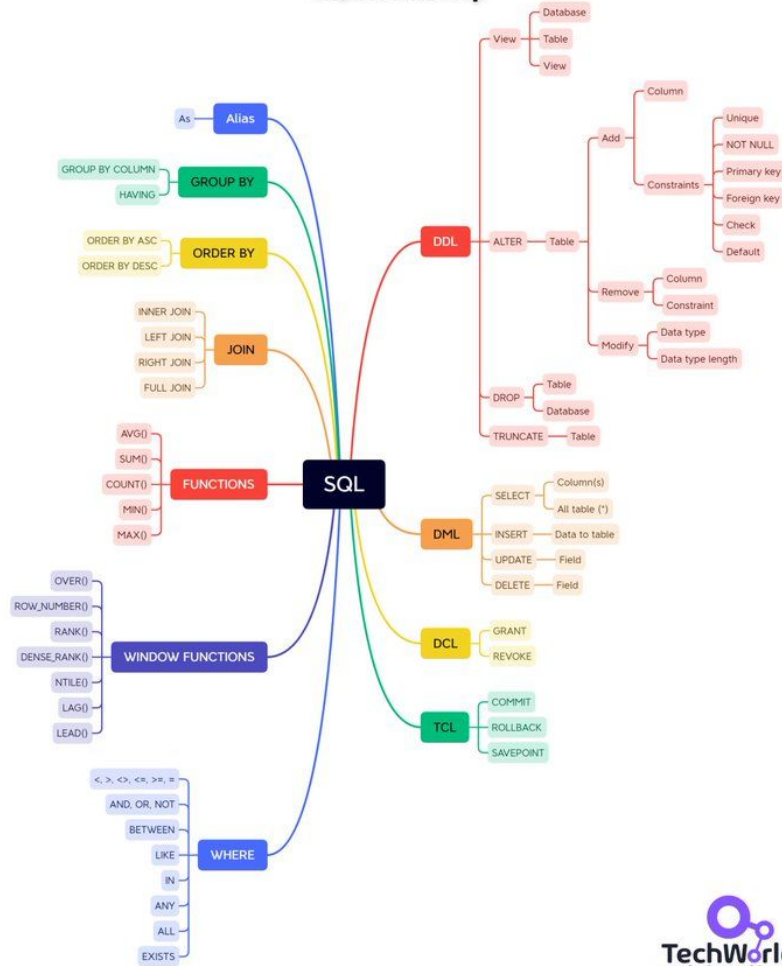


# Subconsultas, Subselects, Subquerys, Inner Querys

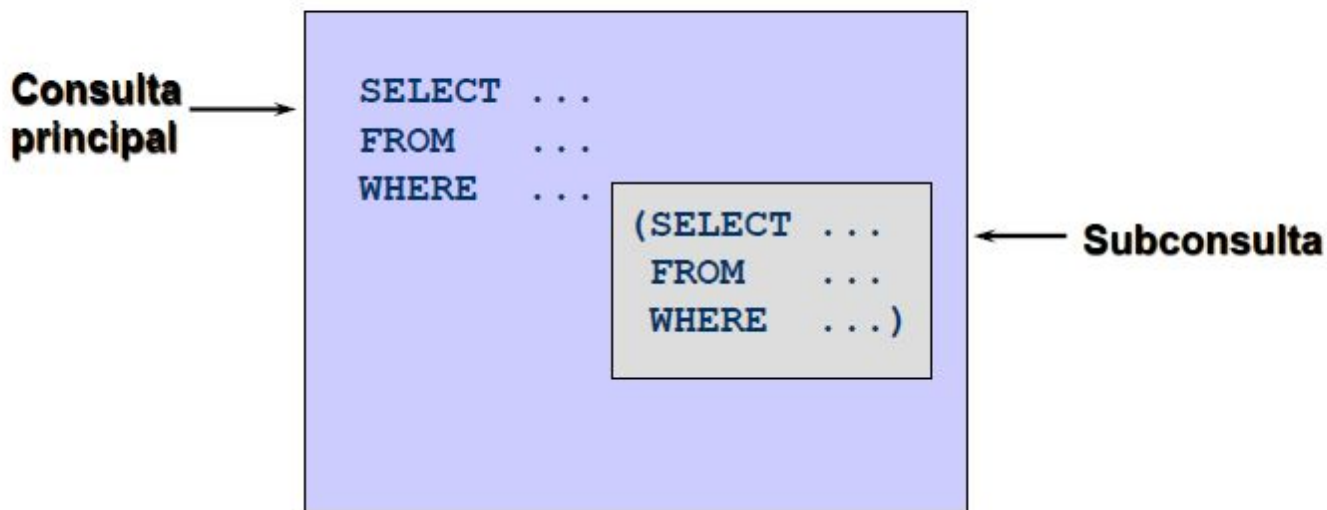
... una consulta dentro de otra consulta

# SQL Mindmap



# ¿Qué Es una Subconsulta?

Una subconsulta es una sentencia `SELECT` que está embebida en una cláusula de otra sentencia `SQL`.



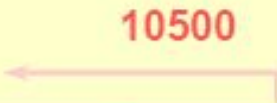
## Subconsultas

```
SELECT select_list
FROM   table
WHERE  expr operator (SELECT select_list
                        FROM   table);
```

- La subconsulta (consulta interna) se ejecuta una vez antes de la consulta principal.
- El resultado de la subconsulta lo utiliza la consulta principal (consulta externa).

## Uso de una Subconsulta

```
SELECT last_name  
FROM employees  
WHERE salary >  
      (SELECT salary  
        FROM employees  
        WHERE employee_id = 149) ;
```



The diagram illustrates the execution of the SQL query. A red arrow points from the value '10500' to the comparison operator '>' in the WHERE clause of the main query. The subquery is enclosed in a red rectangular box.

## Subconsultas de Varias Columnas

### Consulta principal

WHERE (MANAGER\_ID, DEPARTMENT\_ID) IN

### Subconsulta

100	90
102	60
124	50



**Cada fila de la consulta principal se compara con los valores de una subconsulta de varias filas y de varias columnas.**

## Comparaciones de Columnas

Las comparaciones de columnas en una subconsulta de varias columnas pueden ser:

- Comparaciones entre pares
- Comparaciones no entre pares

## Subconsulta de Comparación entre Pares

Visualice los detalles de los empleados dirigidos por el mismo director y que trabajen en el mismo departamento que los empleados cuyo `EMPLOYEE_ID` sea 178 o 174.

```
SELECT employee_id, manager_id, department_id
FROM   employees
WHERE  (manager_id, department_id) IN
      (SELECT manager_id, department_id
       FROM   employees
       WHERE  employee_id IN (178,174))
AND    employee_id NOT IN (178,174);
```



## Subconsulta de Comparación No entre Pares

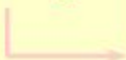
Visualice los detalles de los empleados dirigidos por el mismo director que los empleados con `EMPLOYEE_ID` 174 o 141 y que trabajen en el mismo departamento que los empleados con `EMPLOYEE_ID` 174 o 141.

```
SELECT  employee_id, manager_id, department_id
FROM    employees
WHERE   manager_id IN
        (SELECT  manager_id
         FROM    employees
         WHERE   employee_id IN (174,141))
AND     department_id IN
        (SELECT  department_id
         FROM    employees
         WHERE   employee_id IN (174,141))
AND     employee_id NOT IN(174,141);
```

## Uso de Subconsultas Correlacionadas

Busque todos los empleados que ganan más del salario medio en su departamento.

```
SELECT last_name, salary, department_id
FROM   employees outer
WHERE  salary >
      (SELECT AVG(salary)
       FROM   employees
       WHERE  department_id =
             outer.department_id) ;
```



Cada vez que se procesa una fila de la consulta externa, se evalúa la consulta interna.

## Uso de una Subconsulta en la Cláusula FROM

```
SELECT  a.last_name, a.salary,  
        a.department_id, b.salavg  
FROM    employees a, (SELECT  department_id,  
                        AVG(salary) salavg  
                     FROM    employees  
                     GROUP BY department_id) b  
WHERE   a.department_id = b.department_id  
AND     a.salary > b.salavg;
```

## Uso de Subconsultas Correlacionadas

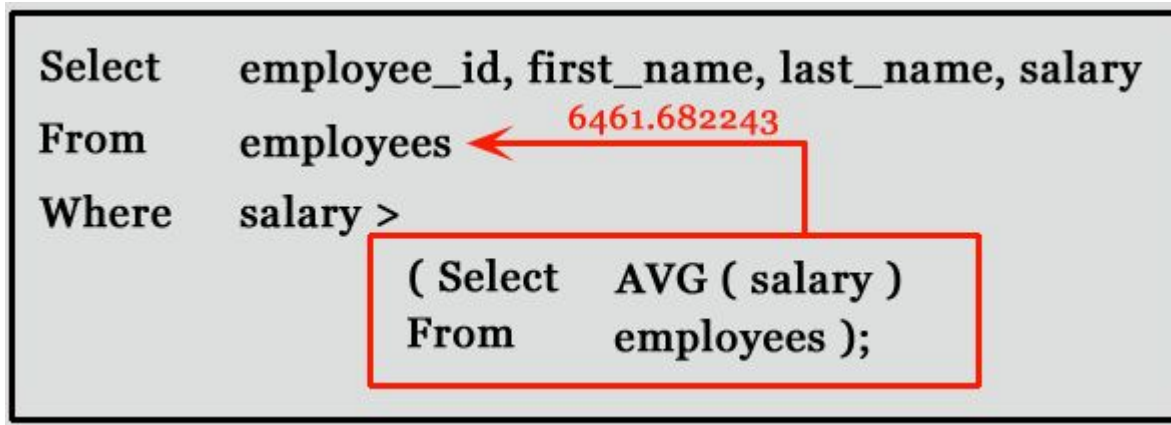
Visualice detalles de los empleados que han cambiado de cargo al menos dos veces.

```
SELECT e.employee_id, last_name, e.job_id
FROM   employees e
WHERE  2 <= (SELECT COUNT(*)
              FROM   job_history
              WHERE  employee_id = e.employee_id);
```

# Subselect ESCALARES. En la cláusula

!!!Devuelve un único valor en un único campo!!!

```
Select  employee_id, first_name, last_name, salary
From    employees
Where   salary >
        ( Select  AVG ( salary )
          From    employees );
```

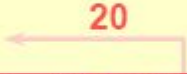


Van en el Where o en el Order By (nunca en Group By)

# Subselect ESCALARES. En el Select

!!!Devuelve un único valor en un único campo!!!

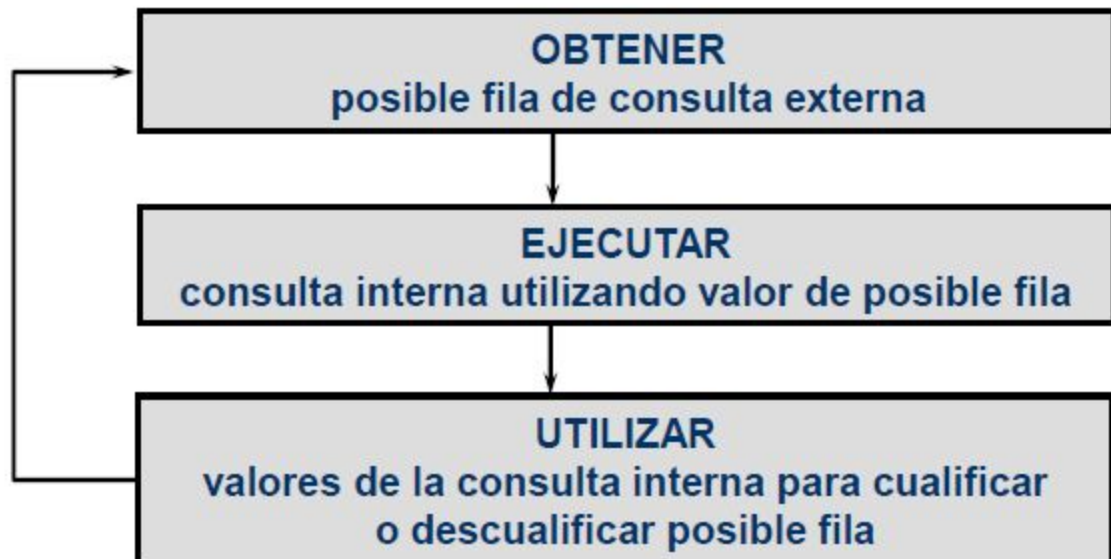
## Subconsultas Escalares en Expresiones

```
SELECT employee_id, last_name,  
       (CASE  
         WHEN department_id = 20  
         (SELECT department_id FROM departments  
          WHERE location id = 1800)  
         THEN 'Canada' ELSE 'USA' END) location  
FROM   employees;
```



## Subconsultas Correlacionadas

Las subconsultas correlacionadas se utilizan para el procesamiento fila a fila. Cada subconsulta se ejecuta una vez para cada fila de la consulta externa.



# Subconsultas Correlacionadas

```
SELECT column1, column2, ...  
FROM   table1 outer  
WHERE  column1 operator  
        (SELECT column1, column2  
         FROM   table2  
         WHERE  expr1 =  
                outer.expr2) ;
```

La subconsulta hace referencia a una columna de una tabla en la consulta principal.



## Uso del Operador EXISTS

- El operador EXISTS comprueba la existencia de filas en el juego de resultados de la subconsulta.
- Si se encuentra un valor de fila de la subconsulta:
  - La búsqueda no continúa en la consulta interna.
  - Se señala a la condición como TRUE.
- Si no se encuentra un valor de fila de la subconsulta:
  - Se señala a la condición como FALSE.
  - La búsqueda continúa en la consulta interna.

## Uso del Operador EXISTS

Busque los empleados que tengan al menos una persona que les informe.

```
SELECT employee_id, last_name, job_id, department_id
FROM   employees outer
WHERE  EXISTS ( SELECT 'X'
                  FROM   employees
                  WHERE  manager_id =
                        outer.employee_id) ;
```

## Uso del Operador NOT EXISTS

Busque todos los departamentos que no tengan empleados.

```
SELECT department_id, department_name
FROM departments d
WHERE NOT EXISTS (SELECT 'X'
                  FROM employees
                  WHERE department_id
                     = d.department_id);
```

# La Cláusula WITH

- Con la cláusula WITH, puede utilizar el mismo bloque de consulta en una sentencia SELECT cuando se produce más de una vez dentro de una consulta compleja.
- La cláusula WITH recupera el resultado de un bloque de consulta y lo almacena en el tablespace temporal del usuario.
- La cláusula WITH mejora el rendimiento.

```
WITH  
  cte1 AS (SELECT a, b FROM table1),  
  cte2 AS (SELECT c, d FROM table2)  
SELECT b, d FROM cte1 JOIN cte2  
WHERE cte1.a = cte2.c;
```

## Cláusula WITH: Ejemplo

**WITH**

```
dept_costs AS (  
    SELECT d.department_name, SUM(e.salary) AS dept_total  
    FROM   employees e, departments d  
    WHERE  e.department_id = d.department_id  
    GROUP BY d.department_name),  
avg_cost AS (  
    SELECT SUM(dept_total)/COUNT(*) AS dept_avg  
    FROM   dept_costs)  
SELECT *  
FROM   dept_costs  
WHERE  dept_total >  
        (SELECT dept_avg  
         FROM avg_cost)  
ORDER BY department_name;
```

# Consejos y recomendaciones

- Una subconsulta debe estar siempre entre **paréntesis**.
- Se debe colocar una subconsulta en el **lado derecho** del operador de comparación.
- Las subconsultas no pueden manipular sus resultados internamente, por lo tanto, la cláusula **ORDER BY no se puede agregar a una subconsulta**. Puede usar una cláusula ORDER BY en la instrucción SELECT principal (consulta externa), que será la última cláusula.
- Utilice operadores de una sola fila con subconsultas de una sola fila.
- Si una subconsulta (consulta interna) devuelve un valor nulo a la consulta externa, la consulta externa no devolverá ninguna fila al usar ciertos operadores de comparación en una cláusula WHERE.

# Consejos y recomendaciones

- Mecanismo muy útil junto con INSERT
  - Inserta en la Tabla A los registros que la Tabla B que cumplan una condición más unos datos fijos (hacer un INSERT - SELECT)
- Idem con UPDATE y DELETE.
  - Aunque es recomendable operar con extrema cautela este tipo de operaciones
- El uso de subselects no está enfrentado al uso de Joins, pero es cierto que este último suele ser más eficiente y rápido pues las columnas PK y FK suelen estar indexadas. Más info en <https://www.navicat.com/en/company/aboutus/blog/1704-joins-versus-subqueries-which-is-faster>

