

Contents

| | | |
|----------|--|----------|
| 0.1 | Set membership (<code>IN, NOT IN</code>): | 1 |
| 0.1.1 | Sintaxis: | 1 |
| 0.2 | Set comparison (<code>ALL, SOME</code>): | 1 |
| 0.2.1 | Sintaxis: | 2 |
| 0.3 | Empty relations (<code>EXISTS, NOT EXISTS</code>): | 2 |
| 0.3.1 | Sintaxis: | 2 |
| 0.4 | Subconsulta Correlacionada: | 2 |
| 0.4.1 | Sintaxis: | 2 |
| 0.5 | Subconsulta Escalar: | 2 |
| 0.5.1 | Sintaxis: | 2 |
| 0.6 | CTE (Common Table Expression): | 3 |
| 0.6.1 | Sintaxis: | 3 |
| 0.7 | Agregaciones: | 3 |
| 0.7.1 | Sintaxis: | 3 |
| 0.8 | GROUP BY: | 3 |
| 0.8.1 | Sintaxis: | 3 |
| 0.9 | HAVING: | 3 |
| 0.9.1 | Sintaxis: | 4 |
| 1 | Tips | 4 |
| 1.1 | JOINS | 4 |
| 1.1.1 | Cuando usar LEFT o RIGHT join | 4 |
| 1.2 | SELECT | 4 |
| 1.2.1 | SELECT DISTINCT | 4 |

0.1 Set membership (`IN, NOT IN`):

Se utiliza para verificar si un valor coincide con algún valor dentro de un conjunto de resultados o una lista de valores.

0.1.1 Sintaxis:

```
SELECT column_name  
FROM table_name  
WHERE column_name IN (value1, value2, ...);
```

0.2 Set comparison (`ALL, SOME`):

`ALL` verifica si una condición es verdadera para todos los valores en una subconsulta, mientras que `SOME` verifica si es verdadera para al menos un valor.

0.2.1 Sintaxis:

```
SELECT column_name  
FROM table_name  
WHERE column_name > ALL (subquery);
```

0.3 Empty relations (EXISTS, NOT EXISTS):

EXISTS verifica si una subconsulta devuelve alguna fila, indicando una relación no vacía.
NOT EXISTS verifica si hay una relación vacía.

0.3.1 Sintaxis:

```
SELECT column_name  
FROM table_name  
WHERE EXISTS (subquery);
```

0.4 Subconsulta Correlacionada:

Este tipo de subconsulta depende de la consulta externa para sus valores y se ejecuta repetidamente para cada fila en la consulta externa.

0.4.1 Sintaxis:

```
SELECT column_name  
FROM table_name t1  
WHERE column_name = (SELECT column_name FROM table_name t2 WHERE t1.column_name = t2.colu
```

0.5 Subconsulta Escalar:

Esta subconsulta devuelve un solo valor y se puede utilizar en varias cláusulas como SELECT, WHERE y HAVING.

0.5.1 Sintaxis:

```
SELECT column_name  
FROM table_name  
WHERE column_name = (SELECT single_value FROM table_name);
```

0.6 CTE (Common Table Expression):

Un conjunto de resultados temporal nombrado definido dentro de una consulta, mejorando la legibilidad y la modularidad.

0.6.1 Sintaxis:

```
WITH cte_name AS (
    SELECT column_name
    FROM table_name
)
SELECT *
FROM cte_name;
```

0.7 Agregaciones:

Funciones que operan sobre un conjunto de valores para devolver un solo valor, como AVG, MAX, MIN, COUNT y SUM.

0.7.1 Sintaxis:

```
SELECT AVG(column_name)
FROM table_name;
```

0.8 GROUP BY:

Agrupa filas con los mismos valores en columnas especificadas, permitiendo que se apliquen funciones de agregación a cada grupo.

0.8.1 Sintaxis:

```
SELECT column_name, COUNT(*)
FROM table_name
GROUP BY column_name;
```

0.9 HAVING:

Filtra los resultados de las agregaciones de GROUP BY basándose en condiciones especificadas.

0.9.1 Sintaxis:

```
SELECT column_name, COUNT(*)
FROM table_name
GROUP BY column_name
HAVING COUNT(*) > value;
```

1 Tips

1.1 JOINS

1.1.1 Cuando usar LEFT o RIGHT join

Cuando pueden haber valores nulos.

Por ejemplo, un pais que no tiene capital.

1.2 SELECT

1.2.1 SELECT DISTINCT

se utiliza para eliminar las filas duplicadas y mostrar una lista única de valores.