

SQL, o Lenguaje de Consulta Estructurado, es un lenguaje de programación utilizado para interactuar con bases de datos relacionales. Nos va a permitir definir la estructura de los datos, manipular y consultar información almacenada en las bases de datos.

Consultas básicas	
SELECT	Se utiliza para seleccionar columnas específicas de una o varias tablas.
FROM	Indica la tabla o tablas de donde se seleccionarán los datos.
WHERE	Permite filtrar los registros según ciertos criterios.
AS	Se utiliza para renombrar columnas o tablas en los resultados de la consulta.
<pre>SELECT col_a AS alias_a FROM tabla_a WHERE condición</pre>	

Operadores de comparación				
== (Igual que)	≠ (Distinto de)	> (Mayor que)	< (Menor que)	>= Mayor o igual
<= (Menor o igual)	IN (Coincidencia valores)	NOT IN (No coincidencia valaores)	BETWEEN (Rango de valores)	AND (Varias condiciones ocurren)
OR (Ocurre una condición o la otra)				

Consultas de agregación	
MIN	Devuelve el valor mínimo de una columna numérica
MAX	Devuelve el valor máximo de una columna numérica
SUM	Calcula la suma de una columna numérica
COUNT	Cuenta el número de filas de una tabla o el de no nulos en una columna.
AVG	Calcula el promedio de una columna numérica
STDDEV	Calcula la desviación estándar de una columna numérica
VARIANCE	Calcula la varianza de una columna numérica
CONCAT	Concatena dos o más cadenas de texto en una columna

Consultas de ordenación	
ORDER BY	Se utiliza para ordenar los resultados según un criterio específico.
LIMY	Permite especificar el número máximo de filas que queremos devolver.
OFFSET	Se utiliza con LIMIT para especificar el número de filas que se deben omitir.

GROUP BY

La cláusula **GROUP BY** se utiliza para agrupar filas en base a un criterio específico y realizar operaciones de agregación en cada grupo. Esta cláusula nos permite resumir y obtener información agregada de un conjunto de datos.

Color	Valor
rojo	1
verde	9
rojo	11
rojo	3
verde	2
verde	6
rojo	8
verde	4
verde	4
rojo	11
verde	11
rojo	2
verde	2
verde	4
verde	3
verde	2
rojo	12
rojo	8

groupby ("Color")

1
3
2
6
4
2
4
9
11
8
11
12
6
4
2
3
2
12
8

agregación en Valor
Ej: mean()

3,14
9,83
2,75

ordenamos
Ej: desc

9,83
3,14
2,75

La cláusula **HAVING** se utiliza en combinación con la cláusula **GROUP BY** para filtrar los resultados de una consulta agregada. Permite aplicar condiciones a los grupos resultantes de la cláusula **GROUP BY**.

Mientras que la cláusula **WHERE** se utiliza para filtrar filas individuales antes de que se realice la agregación, la cláusula **HAVING** se utiliza para filtrar grupos después de la agregación.

Orden de ejecución de las queries en PostgreSQL	
1. FROM	Se seleccionan las tablas de donde se obtendrán los datos.
2. WHERE	Se aplican las condiciones de filtrado a las filas individuales.
3. GROUP BY	Se agrupan los datos en base a las columnas especificadas.
4. HAVING	Se aplican las condiciones de filtrado a los grupos resultantes.
5. SELECT	Se seleccionan las columnas a mostrar en el resultado.
6. ORDER BY	Se ordenan los resultados según el criterio especificado.
7. LIMIT	Se limita la cantidad de filas del resultado a mostrar.
8. OFFSET	Se omiten las filas iniciales del resultado.

Orden de ejecución de las queries en PostgreSQL	
INNER JOIN	Devuelve las filas que tienen coincidencias en ambas tablas en función de una condición de unión.
LEFT JOIN	Devuelve todas las filas de la tabla izquierda y las filas coincidentes de la tabla derecha en función de una condición de unión.
RIGHT JOIN	Devuelve todas las filas de la tabla derecha y las filas coincidentes de la tabla izquierda en función de una condición de unión.
FULL JOIN	Devuelve todas las filas de ambas tablas, incluidas las filas que no tienen coincidencias en la otra tabla.
CROSS JOIN	Devuelve el producto cartesiano de dos tablas. El resultado final contendrá todas las combinaciones posibles de filas de ambas tablas
SELF JOIN	Combina filas de una misma tabla. Se hace referencia a la misma tabla en la cláusula FROM y se utilizan alias para distinguir las tablas

GROUP BY	
SELECT	t_a.columna_a, t_b.columna_b
FROM	tabla_a AS t_a
JOIN	tabla_b AS t_b
ON	col_union_a = col_union_b
WHERE	condición
GROUP BY	col_agrupación
HAVING	condición_de_agrupación
ORDER BY	col_ordenación
LIMIT	registros_a_mostrar
OFFSET	registros_a_saltar