*Introducción a Having*

En SQL, la cláusula GROUP BY nos permite agrupar datos. Si queremos filtrar la información obtenida utilizaremos HAVING.

HAVING se emplea para filtrar los resultados de una consulta que involucra funciones agregadas. En otras palabras, HAVING permite aplicar condiciones de filtrado a los resultados de funciones como COUNT, MAX, MIN, SUM y AVG después de que se han agrupado los datos con la cláusula GROUP BY.

Tabla

Descripción generada automáticamentePor ejemplo, si tenemos la siguiente tabla de inscripciones

Nos piden crear un reporte mostrando los meses y la cantidad de inscritos, pero solo donde hayan 2 o más inscritos.

SELECT strftime("%m", Fecha\_Inscripcion) AS mes, COUNT(Fecha\_Inscripcion) cantidad\_usuarios

FROM inscripciones

GROUP BY strftime("%m", Fecha\_Inscripcion)

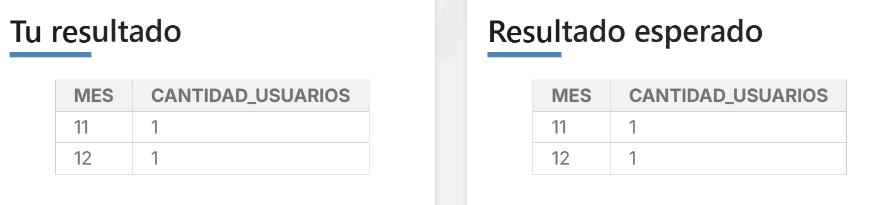
HAVING cantidad\_usuarios >= 2

En esta consulta, primero utilizamos GROUP BY para agrupar por mes. Luego, utilizamos la función de agregación COUNT(Fecha\_Inscripcion) para contar la cantidad de inscritos.Después de haber agrupado los datos y calculado el total de inscritos, aplicamos la cláusula HAVING para filtrar los resultados.

Ejercicio

Crea un reporte mostrando los meses y la cantidad de inscritos pero solo donde haya 1 inscrito. Las columnas deben llamarse mes y cantidad\_usuarios respectivamente.

SELECT strftime('%m',Fecha\_inscripcion) as mes, COUNT(Fecha\_inscripcion) as cantidad\_usuarios from inscripciones GROUP BY 1 HAVING cantidad\_usuarios=1



Primero seleccionamos o seteamos lo que se quiere seleccionar en este caso el mes junto a la columna y un alias, luego se hace el conteo y se selecciona la columna y se muestra con el nombre de otra columna ya que se hizo un alias, a continuación se selecciona la tabla para luego agrupar mediante la primera selección y por ultimo se hace la comparación para traer el resultado con el HAVING.

*Buscando duplicados*

Uno de los usos mas recurrentes de HAVING es buscar duplicados. Por ejemplo, dada una tabla de correos ver cuales están más de 1 vez.

Ejercicio

Se tiene la tabla *correos\_corporativos*

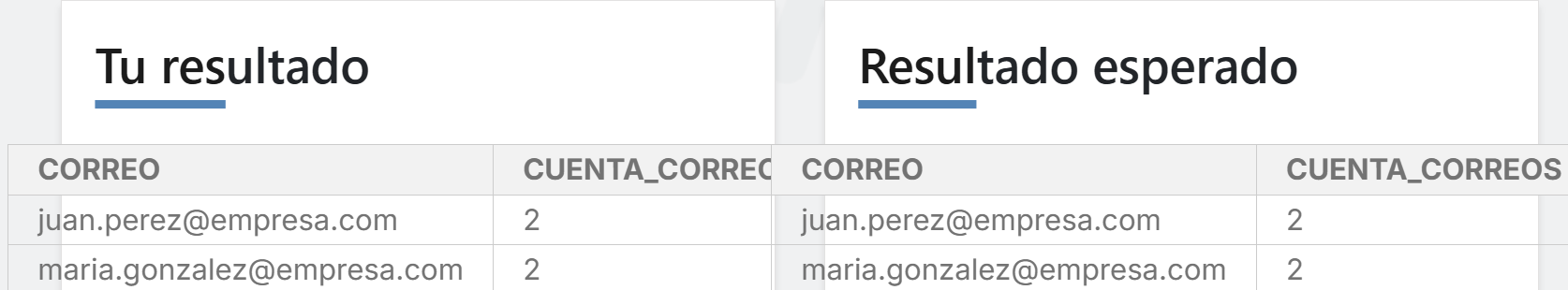
Interfaz de usuario gráfica

Descripción generada automáticamente

Muestra los correos que aparezcan en más de una ocasión. La tabla resultante debe tener dos columnas: una llamada correo, y otra llamada cuenta\_correos que muestra la cantidad de repeticiones correspondiente a cada correo.

select correo, COUNT(correo) as cuenta\_correos from correos\_corporativos

GROUP BY 1 HAVING COUNT(correo)>1



*Having y cuenta*

Ejercicio

Dada la siguiente **tabla empleados**



Crea una consulta que muestre la cantidad de usuarios y el departamento en donde haya más de un empleado. Las columnas deben llamarse cantidad\_de\_usuarios y departamento, respectivamente.

SELECT COUNT(departamento) AS cantidad\_de\_usuarios, departamento FROM empleados

GROUP BY departamento HAVING COUNT(departamento)>1

Interfaz de usuario gráfica, Aplicación, Word

Descripción generada automáticamentePrimero se selecciona y se hace el conteo seleccionando la columna y agregándole un alias, también se selecciona otra columna(departamento) y se selecciona la tabla donde están los datos, luego se agrupa filtrando solamente los que tengan mas de una coincidencia.

*Having y promedio*

Ejercicio

Se tiene la siguiente tabla notas:



Crea una consulta para determinar cuales son los estudiantes que aprobaron. El criterio de aprobación es promedio de notas >= 50.

Las columnas a mostrar deben ser email y promedio\_notas.

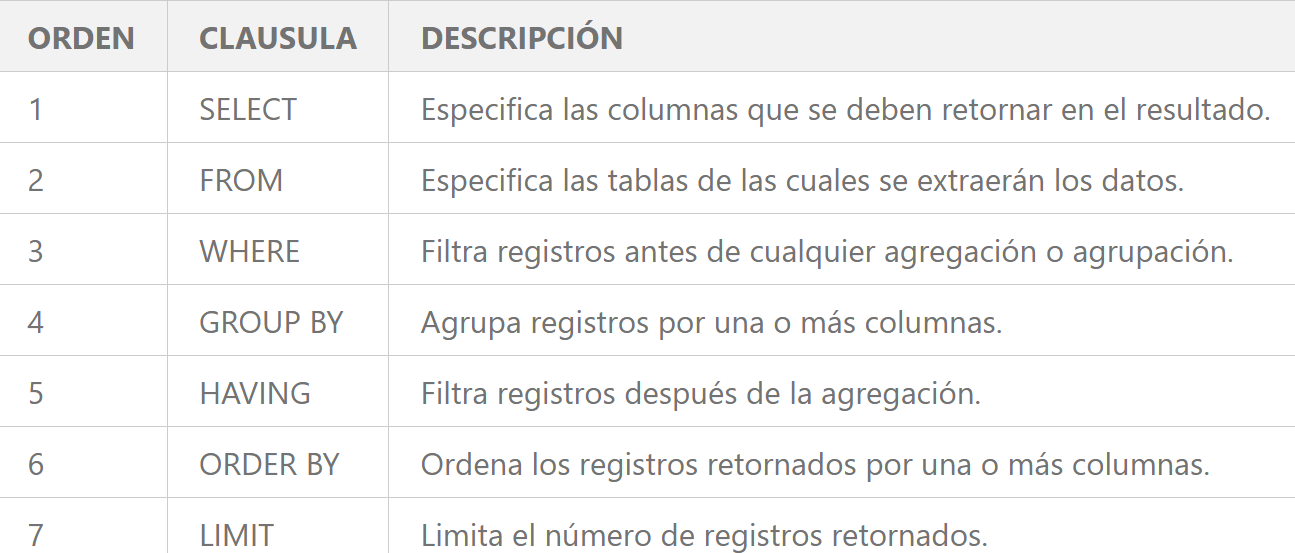
SELECT email, AVG(notas) as promedio\_notas from notas

GROUP BY email HAVING(promedio\_notas)>50;

Se selecciona las columnas email y notas y esta ultima se usa para sacar el promedio y se le asigna un alias y luego se indica desde la tabla que se quiere rescatar la columna y luego se agrupan para seleccionar solo los criterios de promedio en este caso sobre 50.

*Having y order*

Una vez que hemos agrupado datos utilizando la cláusula GROUP BY, es común que necesitemos ordenar esos grupos según algún criterio específico. Por lo general, queremos ordenar los grupos en función de alguna métrica agregada, como la suma, el conteo, el promedio, etc. Para hacer esto, usamos la cláusula ORDER BY junto con las funciones de agregación.

El orden de las clausulas en una consulta debe ser el siguiente:

Ejercicio

Dada la siguiente tabla **ventas**, escribe una consulta SQL para obtener los productos que se han vendido en una cantidad total mayor a 1000, ordenados en orden descendente de cantidad vendida.

Tabla

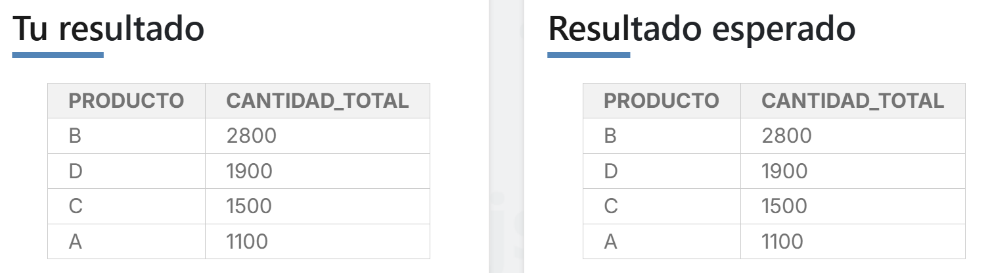
Descripción generada automáticamente

La tabla resultante debe tener dos columnas: 'producto' y 'cantidad\_total'.

select producto, SUM(cantidad) as cantidad\_total from ventas

group by producto having (cantidad\_total)>1000 order by cantidad\_total desc

se seleccionan las columas product y cantidad esta ultima se le agrega la función SUM para hacer toda la suma de los productos y se le asigna un alias y se selecciona la tabla, después se agrupan los resultados y se hace una verificación de que la cantidad total sea mayor a 1000 y se ordenan de forma descendiente seleccionando la columna a ordenar.



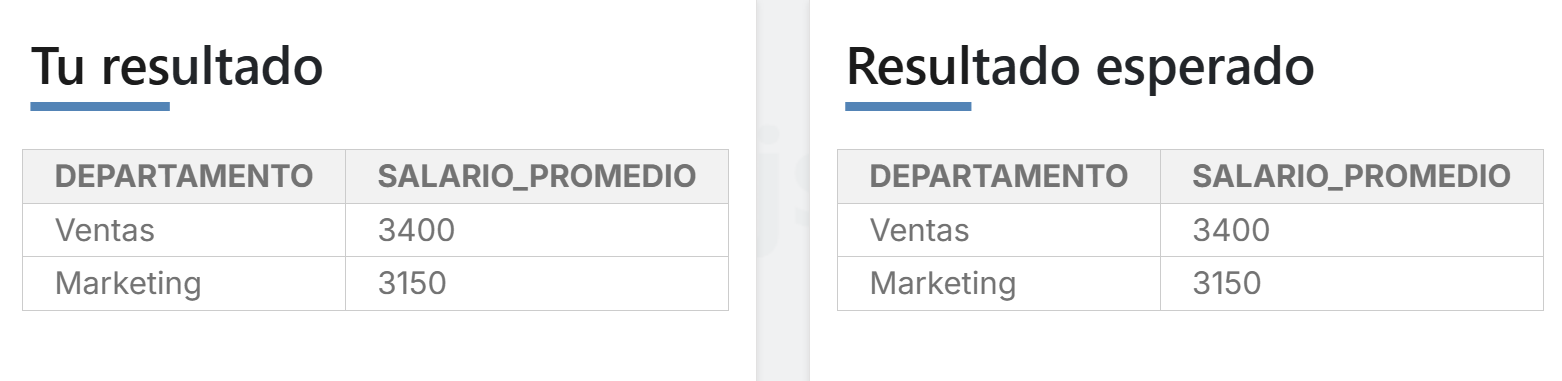
*Having y order 2*

Ejercicio

Supongamos que tienes una tabla de empleados con los siguientes datos:

Tu tarea es escribir una consulta SQL que devuelva los departamentos cuyo salario promedio es mayor a 3000, ordenados de mayor a menor salario promedio. Los resultados deben mostrar el nombre del departamento y el salario promedio, con los nombres de las columnas como Departamento y Salario\_Promedio respectivamente.

SELECT departamento, AVG(salario) as Salario\_promedio FROM empleados

GROUP BY departamento HAVING(Salario\_promedio) > 3000 ORDER BY Salario\_promedio desc

Se selecciona las columnas departamento y salario esta ultima se hace calculo para sacar promedio y luego se le asigna un alias para mostrar el resultado y se selecciona la tabla, luego se agrupa y se hace una validación para mostrar los resultados de forma de que se muestre el promedio mas alto y descendiendo los demás.