

Utilizando **GROUP BY** e **HAVING** em SQL

Introdução

Hoje vamos aprender a trabalhar com as cláusulas **GROUP BY** e **HAVING** em SQL. Elas são muito úteis quando queremos agrupar informações e aplicar funções de agregação, como somar, contar ou calcular a média. Além disso, o **HAVING** nos permite filtrar os resultados após a agregação, similar ao **WHERE**, mas aplicável em conjuntos agrupados.

Cenário

Vamos usar um banco de dados simples de uma empresa com três tabelas principais:

- **Clientes:** Contém informações sobre os clientes.
- **Pedidos:** Armazena os pedidos feitos pelos clientes.
- **ItensPedido:** Registra os itens de cada pedido.

Você já conhece a estrutura dessas tabelas com exemplos reais de dados, então vamos focar diretamente em como o **GROUP BY** e o **HAVING** funcionam.

GROUP BY: Agrupando Resultados

Imagine que queremos saber **quantos pedidos cada cliente fez**. Podemos usar o **GROUP BY** para agrupar os pedidos por cliente e contar quantos pedidos há para cada um. Vamos fazer isso!

```
SELECT c.nome, COUNT(p.pedido_id) AS total_pedidos
FROM Clientes c
JOIN Pedidos p ON c.cliente_id = p.cliente_id
GROUP BY c.nome;
```

Explicação:

- Estamos selecionando o nome do cliente da tabela **Clientes** e contando quantos **pedido_id** existem na tabela **Pedidos**.
- A cláusula **GROUP BY** agrupa os resultados pelo nome do cliente. Assim, para cada cliente, teremos o número total de pedidos feitos.

Resultado esperado:

nome	total_pedidos
João Silva	4
Maria Souza	3
Carlos Lima	3

Funções de Agregação

Além de contar, também podemos somar, calcular médias e muito mais.

Por exemplo, se quisermos saber o **valor total dos pedidos por cliente**, podemos usar o `SUM()`:

```
SELECT c.nome, SUM(p.valor_total) AS total_gasto
FROM Clientes c
JOIN Pedidos p ON c.cliente_id = p.cliente_id
GROUP BY c.nome;
```

Aqui, estamos somando o valor total de todos os pedidos de cada cliente.

HAVING: Filtrando após o agrupamento

O `HAVING` é usado quando queremos **filtrar resultados após o `GROUP BY`**. Imagine que queremos ver apenas os clientes que fizeram **mais de 3 pedidos**. O `WHERE` não funcionaria nesse caso porque estamos lidando com um valor agregado, então usamos `HAVING`:

```
SELECT c.nome, COUNT(p.pedido_id) AS total_pedidos
FROM Clientes c
JOIN Pedidos p ON c.cliente_id = p.cliente_id
GROUP BY c.nome
HAVING COUNT(p.pedido_id) > 3;
```

Explicação:

- A função `COUNT()` nos dá o número total de pedidos por cliente.
- `HAVING COUNT(p.pedido_id) > 3` vai exibir apenas os clientes que têm mais de 3 pedidos.

Resultado esperado:

nome	total_pedidos	
João Silva	4	

Exemplo Combinado: Pedido Médio por Cliente

Vamos calcular o **valor médio de pedidos por cliente**, mas queremos exibir apenas os clientes que gastaram mais de R\$ 1000 em média.

```
SELECT c.nome, AVG(p.valor_total) AS media_gasto
FROM Clientes c
JOIN Pedidos p ON c.cliente_id = p.cliente_id
GROUP BY c.nome
HAVING AVG(p.valor_total) > 1000;
```

Aqui, usamos **AVG()** para calcular a média de cada cliente e **HAVING** para filtrar aqueles cuja média de gastos é maior que R\$ 1000.

Resultado esperado:

nome	media_gasto	
Carlos Lima	1300.00	
João Silva	825.00	

Conclusão

As cláusulas **GROUP BY** e **HAVING** são extremamente úteis quando estamos trabalhando com dados que precisam ser agrupados e agregados. Enquanto o **GROUP BY** organiza os dados, o **HAVING** permite filtrar resultados após a agregação.