

Group By, Funções Agregadoras e Having

Entendendo Conceitos

→ O que é o GROUP BY?

O **GROUP BY** é um comando SQL usado para **agrupar linhas** da tabela que possuem o **mesmo valor** em uma ou mais colunas.

- Ele **organiza os dados em grupos**.
- Cada grupo pode ser **resumido** usando **funções agregadoras**.

Pense assim: você quer **juntar** todas as vendas do mesmo vendedor, do mesmo produto, da mesma cidade, etc.

→ O que são Funções Agregadoras?

As funções agregadoras **pegam os dados de cada grupo** e **fazem cálculos** em cima deles.

Principais funções:

Função	O que faz	Exemplo
SUM()	Soma todos os valores	Total vendido
AVG()	Calcula a média dos valores	Média de vendas
COUNT()	Conta quantas linhas tem	Quantas vendas
MAX()	Mostra o maior valor	Venda mais cara
MIN()	Mostra o menor valor	Venda mais barata

→ O que é o HAVING?

- O **HAVING** é usado para **filtrar os grupos** depois que o **GROUP BY** e as funções agregadoras foram aplicadas.
- É **parecido com o WHERE**, mas o **WHERE** filtra linhas **antes** do agrupamento, e o **HAVING** filtra **depois**.

Parte 2: Estrutura dos Comandos

→ Estrutura geral de um SQL com GROUP BY:

```
SELECT
    Coluna(s), Função_Agregadora(Coluna) AS NomeResultado
FROM
    NomeTabela
[WHERE condição]      -- (opcional) → Filtra linhas antes do agrupamento
GROUP BY Coluna(s)
[HAVING condição]     -- (opcional) → Filtra grupos depois do agrupamento
```

Parte 3: Exemplos Práticos com Tabelas

Vamos trabalhar com esta tabela de exemplo chamada **Vendas**:

VendaID	Vendedor	Valor
1	Ana	100
2	João	200
3	Ana	150
4	Maria	300

Exemplo 1: Somar quanto cada vendedor vendeu

Consulta:

```
SELECT
    Vendedor, SUM(Valor) AS TotalVendas
FROM
    Vendas
GROUP BY
    Vendedor;
```

Resultado:

Vendedor	TotalVendas
Ana	250
João	200
Maria	300

✅ Aqui o **GROUP BY** agrupou os vendedores e o **SUM()** somou as vendas de cada um.

Exemplo 2: Contar quantas vendas cada vendedor fez

Consulta:

```
SELECT
    Vendedor, COUNT(*) AS QuantidadeVendas
FROM
    Vendas
GROUP BY
    Vendedor;
```

Resultado:

Vendedor	QuantidadeVendas
Ana	2
João	1
Maria	1

✓ Usamos `COUNT(*)` para contar quantas vendas cada vendedor realizou.

Exemplo 3: Mostrar vendedores que venderam acima de 200

Consulta:

```
SELECT
    Vendedor, SUM(Valor) AS TotalVendas
FROM
    Vendas
GROUP
    BY Vendedor
HAVING
    SUM(Valor) > 200;
```

Resultado:

Vendedor	TotalVendas
Ana	250
Maria	300

✓ O João ficou de fora, porque vendeu só 200.

Parte 4: Entendendo melhor o funcionamento

Fluxo de execução do SQL com GROUP BY:

1. **FROM:** Escolhe a tabela.
2. **WHERE:** Filtra registros antes de agrupar (se existir).
3. **GROUP BY:** Agrupa os registros.
4. **Funções Agregadoras:** Calculam resultados por grupo.
5. **HAVING:** Filtra os grupos (se existir).
6. **SELECT:** Escolhe as colunas e mostra o que vai aparecer.

Parte 5: Relação de Dependência entre GROUP BY e Funções Agregadoras

→ Explicação:

Quando você usa **GROUP BY**, o SQL **espera** que você:

- Use **funções agregadoras** nas colunas que **não estão** no **GROUP BY**.
- Ou, **obrigatoriamente**, coloque **todas as colunas** no **GROUP BY**.

Isso acontece porque, **depois do agrupamento**, cada grupo é tratado como **uma única linha**.

Sem função agregadora, o SQL **não saberia o que mostrar** para os dados agrupados.

→ Exemplo errado (não usar função agregadora):

```
SELECT
    Vendedor, Valor
FROM
    Vendas
GROUP
    BY Vendedor;
```

❌ **Erro:** O SQL não sabe qual **Valor** mostrar para cada vendedor, pois há vários valores diferentes.

→ Exemplo certo (usar função agregadora):

```
SELECT
    Vendedor, SUM(Valor) AS TotalVendas
FROM
    Vendas
GROUP BY
    Vendedor;
```

✅ Agora sim! Usamos `SUM(Valor)` para resumir os valores de cada grupo.

Dica de Ouro:

Sempre que usar GROUP BY, no SELECT:

- As colunas que **não são agregadas DEVEM** estar no `GROUP BY`.
- As colunas **que não estão no GROUP BY DEVEM** ser usadas em uma função agregadora (SUM, AVG, COUNT, etc.).