

# MySQL – GROUP BY – Agrupamento de Registros

Usamos a cláusula GROUP BY para agrupar registros em subgrupos baseados em colunas ou valores retornados por uma expressão.

Com o GROUP BY podemos agrupar os valores de uma coluna e também realizar cálculos sobre esses valores. Desta forma, ao realizarmos uma consulta, os valores encontrados nas linhas são agrupados e então uma função de agregação pode ser aplicada sobre esses grupos.

## Sintaxe básica:

```
SELECT colunas, função_agregação()  
FROM tabela  
WHERE filtro  
GROUP BY coluna
```

Criar uma tabela para testarmos o GROUP BY:

```
CREATE TABLE Vendas (  
ID Smallint Primary Key,  
Nome_Vendedor Varchar(20),  
Quantidade Int,  
Produto Varchar(20),  
Cidade Varchar(20)  
);
```

Inserir registros na tabela criada para teste de GROUP BY:

```
INSERT INTO Vendas (ID, Nome_Vendedor, Quantidade, Produto, Cidade)  
VALUES  
(10,'Jorge',1400,'Mouse','São Paulo'),  
(12,'Tatiana',1220,'Teclado','São Paulo'),  
(14,'Ana',1700,'Teclado','Rio de Janeiro'),  
(15,'Rita',2120,'Webcam','Recife'),  
(18,'Marcos',980,'Mouse','São Paulo'),  
(19,'Carla',1120,'Webcam','Recife'),  
(22,'Roberto',3145,'Mouse','São Paulo');
```

## Usando o GROUP BY:

1 – Consulta usando agregação para obter o total de vendas de Mouses (sem o GROUP BY):

```
SELECT SUM(Quantidade) As TotalMouses  
FROM Vendas  
WHERE Produto = 'Mouse';
```

TotalMouses

5525

2 – Consulta totalizando as vendas de todos os produtos por cidade:

```
SELECT Cidade, SUM(Quantidade) As Total  
FROM Vendas
```

**GROUP BY Cidade;**

Cidade	Total
Recife	3240
Rio de Janeiro	1700
São Paulo	6745

3 – Consulta contando o número de registros de vendas (quantidade de vendas) por cidade:

**SELECT Cidade, COUNT(\*) As Total  
FROM Vendas  
GROUP BY Cidade;**

Cidade	Total
Recife	2
Rio de Janeiro	1
São Paulo	4

4 – Consulta com o total de vendas realizadas por cada vendedor:

**SELECT Nome\_Vendedor, SUM(Quantidade)  
FROM Vendas  
GROUP BY Nome\_Vendedor;**

Nome_Vendedor	SUM(Quantidade)
Ana	1700
Carla	1120
Jorge	1400
Marcos	980
Rita	2120
Roberto	3145
Tatiana	1220

Também é possível utilizar a cláusula GROUP BY sem o emprego de **funções de agregação**. Neste caso, a consulta simplesmente retornará valores sem que haja repetição de dados na coluna indicada pela cláusula. Isso é equivalente a aplicar a **cláusula DISTINCT** em uma consulta usando apenas o SELECT simples.

Muitas vezes é necessário filtrar dados nos grupos retornados pelo GROUP BY. Para isso, empregamos a **cláusula HAVING**.

## MySQL – HAVING – Filtrando os resultados do Agrupamento

A cláusula HAVING é usada para especificar condições de filtragem em grupos de registros ou agregações. É frequentemente usada em conjunto com a **cláusula GROUP BY** para filtrar as colunas agrupadas.

**Sintaxe:**

```
SELECT colunas, função_agregação()
FROM tabela
WHERE filtro
GROUP BY colunas
HAVING filtro_agrupamento
```

Exemplos do uso de **HAVING** (usar tabela criada na aula anterior sobre a [cláusula GROUP BY](#)):

1. Consulta retornando total de vendas das cidades com menos de 2500 produtos vendidos:

```
SELECT Cidade, SUM(Quantidade) As Total
FROM Vendas
GROUP BY Cidade
HAVING SUM(Quantidade) < 2500;
```

2. Consulta retornando total de vendas do produto 'Teclado' das cidades com menos de 1500 teclados vendidos:

```
SELECT Cidade, SUM(Quantidade) As TotalTeclados
FROM Vendas
WHERE Produto = 'Teclado'
GROUP BY Cidade
HAVING SUM(Quantidade) < 1500;
```

## MySQL – INNER JOIN – Consultar dados em duas ou mais Tabelas relacionadas

### JOINS

A cláusula JOIN é usada para combinar dados provenientes de duas ou mais tabelas do banco de dados, baseado em um relacionamento entre colunas destas tabelas. há duas categorias principais de joins:

- **INNER JOIN:** Retorna linhas (registros) quando houver pelo menos uma correspondência em ambas as tabelas.
- **OUTER JOIN:** Retorna linhas (registros) mesmo quando não houver ao menos uma correspondência em uma das tabelas (ou ambas). O OUTER JOIN divide-se em LEFT JOIN, RIGHT JOIN e FULL JOIN.

### INNER JOIN

Como dito, um INNER JOIN (ou simplesmente JOIN) permite obter registros com dados provenientes de duas ou mais tabelas relacionadas do banco de dados no MySQL. A sintaxe básica de um INNER JOIN em uma consulta é:

```
SELECT colunas
FROM tabela1
INNER JOIN tabela2
ON tabela1.coluna=tabela2.coluna;
```

Onde **tabela1.coluna** é o nome da primeira tabela concatenado com um ponto e com o nome da coluna chave primária ou estrangeira da tabela, e **tabela2.coluna** é o nome da segunda tabela concatenado com um ponto e com a chave estrangeira ou primária dessa tabela que se relaciona com a chave da primeira tabela.

Vejamos um exemplo. Vamos consultar as tabelas de livros e autores (tbl\_livro e tbl\_autores) e retornar os dados relativos aos livros e aos autores ao mesmo tempo:

```
SELECT * FROM tbl_Livro  
INNER JOIN tbl_autores  
ON tbl_Livro.ID_Autor = tbl_autores.ID_Autor;
```

ID_Livro	Nome_Livro	ISBN	ID_Autor	Data_Pub	Preco_Livro	ID_editora	ID_Autor	Nome_Autor	Sobrenome_Autor
1	Linux Command Line and Shell Scripting	143856969	5	2009-12-21	68.35	4	5	Richard	Blum
2	SSH, the Secure Shell	127658789	1	2009-12-21	58.30	2	1	Daniel	Barret
3	Using Samba	123856789	2	2000-12-21	61.45	2	2	Gerald	Carter
4	Fedora and Red Hat Linux	123346789	3	2010-11-01	62.24	1	3	Mark	Sobell
5	Windows Server 2012 Inside Out	123356789	4	2004-05-17	66.80	3	4	William	Stanek
6	Microsoft Exchange Server 2010	123366789	4	2000-12-21	45.30	3	4	William	Stanek
7	Enciclopédia de Componentes Eletrônicos vol. 03	153642397	13	2016-05-05	63.39	5	13	Charles	Platt

Vejamos outro exemplo. Faremos a consulta nas tabelas de livros e autores, porém retornando apenas os nomes dos livros, seus ISBNs e os nomes dos autores desses livros. Note que as duas primeiras informações estão na tabela de livros, e o nome do autor está somente na tabela de autores. O que as liga é o relacionamento entre as colunas ID\_Autor existentes nas tabelas – é chave primária na tabela de autores e estrangeira na de livros:

```
SELECT tbl_Livro.Nome_Livro, tbl_Livro.ISBN, tbl_autores.Nome_Autor  
FROM tbl_Livro  
INNER JOIN tbl_autores  
ON tbl_Livro.ID_Autor = tbl_autores.ID_Autor;
```

Nome_Livro	ISBN	Nome_Autor
Linux Command Line and Shell Scripting	143856969	Richard
SSH, the Secure Shell	127658789	Daniel
Using Samba	123856789	Gerald
Fedora and Red Hat Linux	123346789	Mark
Windows Server 2012 Inside Out	123356789	William
Microsoft Exchange Server 2010	123366789	William
Enciclopédia de Componentes Eletrônicos vol. 03	153642397	Charles

Mais um exemplo, agora com filtros. Vamos retornar os nomes dos livros e nomes das editoras, mas somente das editoras cujo nome se inicia com a letra M. Note o uso de aliases nestas declarações, a fim de simplificar o código:

```
SELECT L.Nome_Livro AS Livros, E.Nome_editora AS Editoras  
FROM tbl_Livro AS L  
INNER JOIN tbl_editoras AS E  
ON L.ID_editora = E.ID_editora
```

```
WHERE E.Nome_Editora LIKE 'M%';
```

Livros	Editoras
Windows Server 2012 Inside Out	Microsoft Press
Microsoft Exchange Server 2010	Microsoft Press
Enciclopédia de Componentes Eletrônicos vol. 03	Makermedia

Mais um exemplo para terminar. Agora vamos fazer um INNER JOIN com as três tabelas do banco de dados simultaneamente. Queremos os nomes e preços dos livros, nomes de seus autores e editoras, mas somente das editoras cujo nome se inicia com a letra O, tudo isso ordenado em ordem decrescente de preço dos livros:

```
SELECT L.Nome_Livro AS Livro,  
A.Nome_autor AS Autor,  
E.Nome_Editora AS Editora,  
L.Preco_Livro AS 'Preço do Livro'  
FROM tbl_Livro AS L  
INNER JOIN tbl_autores AS A  
ON L.ID_autor = A.ID_autor  
INNER JOIN tbl_editoras AS E  
ON L.ID_editora = E.ID_editora  
WHERE E.Nome_Editora LIKE 'O%'  
ORDER BY L.Preco_Livro DESC;
```

Livro	Autor	Editora	Preço do Livro
Using Samba	Gerald	O'Reilly	61.45
SSH, the Secure Shell	Daniel	O'Reilly	58.30

Neste exemplo fazemos o join entre a tabela de livros e a de autores, e logo em seguida entra a tabela de livros e a de editoras, pois os relacionamentos se dão entre essas tabelas. Também ordenamos os resultados por ordem decrescente dos preços dos livros.