



Selecionando Registros – Agrupamento / Funções de Agregação

1

Banco de Dados II



Banco de Dados II

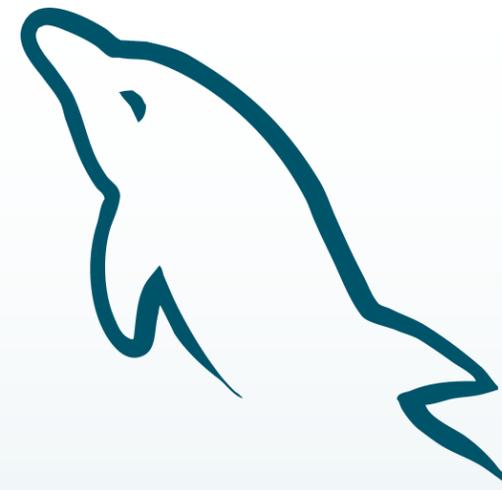
Selecionando Registros

Agrupamento / Funções de Agregação

Funções de Agregação (MAX, MIN, AVG, COUNT, SUM)

Funções de agregação são funções SQL que permitem executar uma operação aritmética nos valores de uma coluna em todos os registros de uma tabela.

Retornam um valor simples baseado em um conjunto de valores de entrada.





Banco de Dados II

Selecionando Registros

Agrupamento / Funções de Agregação

Funções de Agregação (MAX, MIN, AVG, COUNT, SUM)

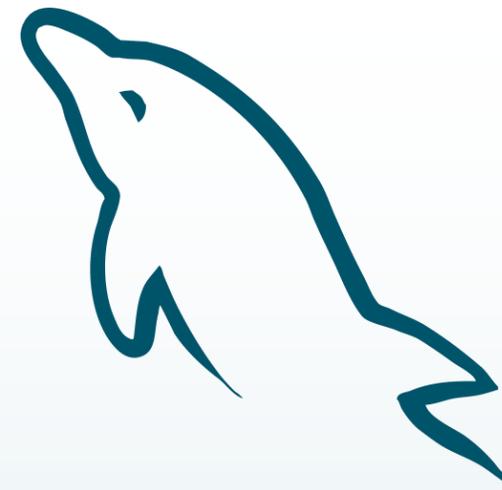
Sintaxe básica:

função(ALL | DISTINCT expressão)

ALL – avalia todos os registros ao agregar o valor da função; é o comportamento padrão.

DISTINCT – Usa apenas valores distintos (sem repetição) ao avaliar a função.

As funções de agregação desconsideram valores NULL (com exceção da função COUNT(*)).





Banco de Dados II

Selecionando Registros

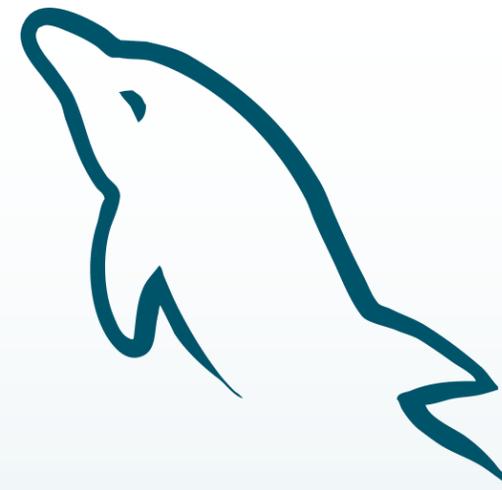
Agrupamento / Funções de Agregação

Funções de Agregação (MAX, MIN, AVG, COUNT, SUM)

As principais funções de agregação (mais comuns) em MySQL são as seguintes:

- **MIN** = Valor Mínimo de um conjunto de valores
- **MAX** = Valor Máximo de um conjunto de valores
- **AVG** = Média Aritmética de um conjunto de valores
- **SUM** = Total (Soma) de um conjunto de valores
- **COUNT** = Contar quantidade total de itens

A funções SUM e AVG somente aceitam como entrada um conjunto de números; já as demais funções podem operar também com outros tipos de dados não-numéricos, como por exemplo strings (caracteres) ou datas.





Banco de Dados II

Selecionando Registros

Agrupamento / Funções de Agregação

Funções de Agregação (MAX, MIN, AVG, COUNT, SUM)

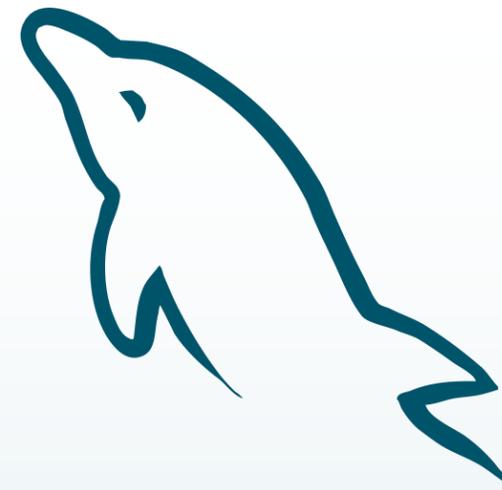
Exemplos:

Retornar o número total de autores cadastrados na tabela de autores:

```
SELECT COUNT (*)  
FROM tbl_autores;
```

Contar o número de autores que possuem livros cadastrados na tabela de autores, sem repetições

```
SELECT COUNT(DISTINCT id_autor)  
FROM tbl_Livro;
```





Banco de Dados II

Selecionando Registros

Agrupamento / Funções de Agregação

Funções de Agregação (MAX, MIN, AVG, COUNT, SUM)

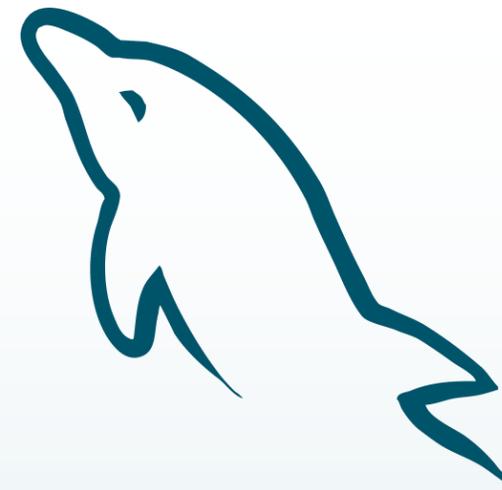
Exemplos:

Descobrir o preço mais alto dos livros:

```
SELECT MAX(Preco_Livro)  
FROM tbl_Livro;
```

Descobrir a data de publicação do livro mais antigo:

```
SELECT MIN(Data_Pub)  
FROM tbl_Livro;
```





Banco de Dados II

Selecionando Registros

Agrupamento / Funções de Agregação

Funções de Agregação (MAX, MIN, AVG, COUNT, SUM)

Exemplos:

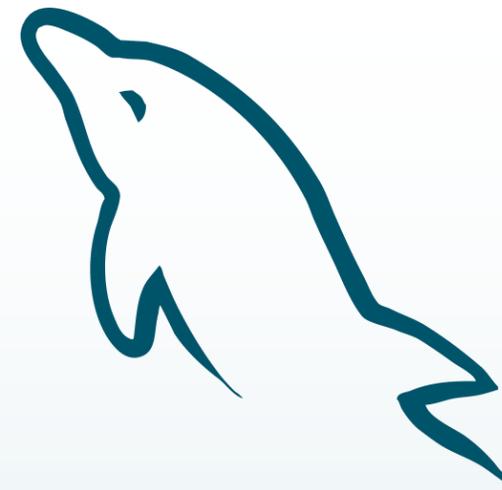
Retornar o preço médio dos livros cadastrados no banco:

```
SELECT AVG(Preco_Livro)  
FROM tbl_Livro;
```

Descobrir o valor total dos livros presentes na tabela de livros:

```
SELECT SUM(Preco_Livro)  
FROM tbl_Livro;
```

É muito comum usar as funções agregadas em conjunto com a cláusula **GROUP BY**, aplicando-as a grupos de dados específicos.





Banco de Dados II

Selecionando Registros

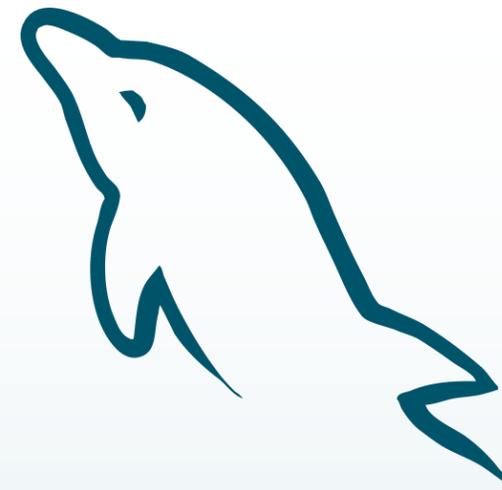
Agrupamento / Funções de Agregação

Consultas com AGRUPAMENTO – cláusula GROUP BY

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.

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





Banco de Dados II

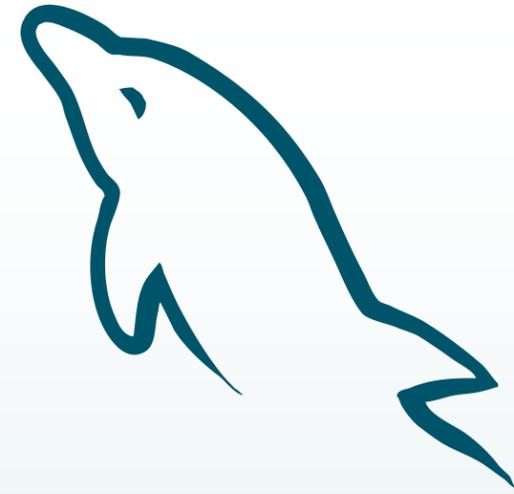
Selecionando Registros

Agrupamento / Funções de Agregação

Consultas com AGRUPAMENTO – cláusula GROUP BY

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)  
);
```





Banco de Dados II

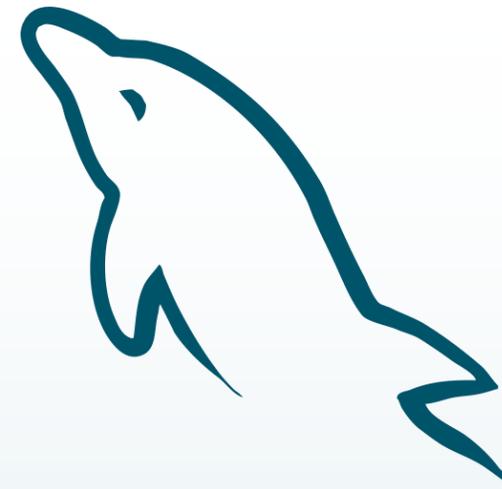
Selecionando Registros

Agrupamento / Funções de Agregação

Consultas com AGRUPAMENTO – cláusula GROUP BY

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');
```





Banco de Dados II

Selecionando Registros

Agrupamento / Funções de Agregação

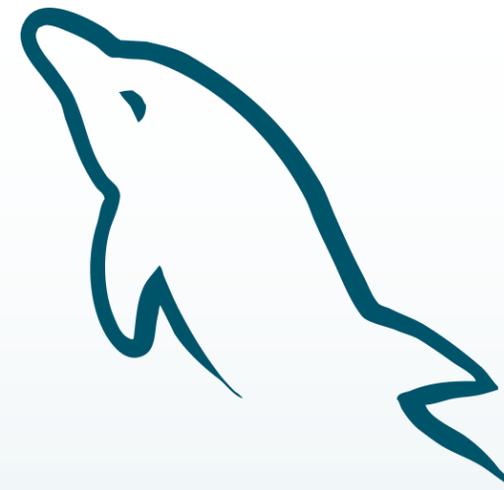
Consultas com AGRUPAMENTO – cláusula GROUP BY

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





Banco de Dados II

Selecionando Registros

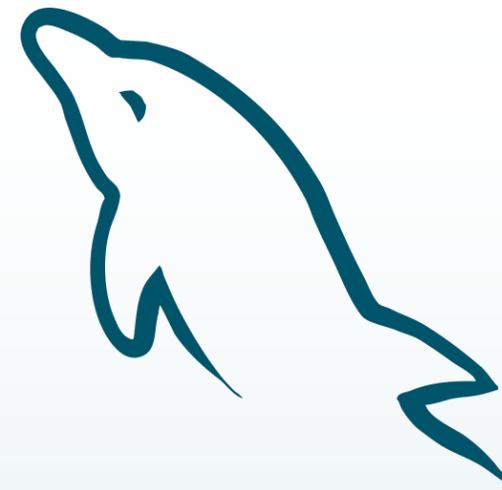
Agrupamento / Funções de Agregação

Consultas com AGRUPAMENTO – cláusula GROUP BY

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





Banco de Dados II

Selecionando Registros

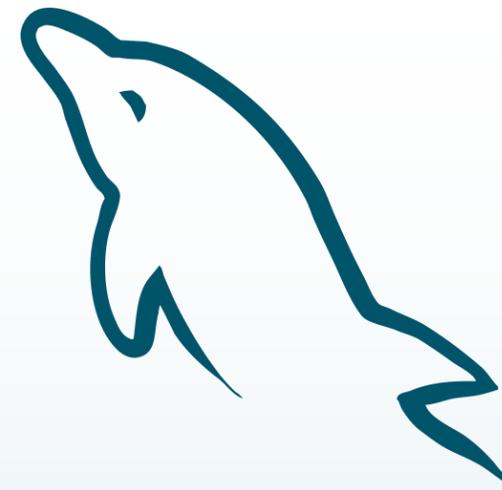
Agrupamento / Funções de Agregação

Consultas com AGRUPAMENTO – cláusula GROUP BY

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





Banco de Dados II

Selecionando Registros

Agrupamento / Funções de Agregação

Consultas com AGRUPAMENTO – cláusula GROUP BY

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



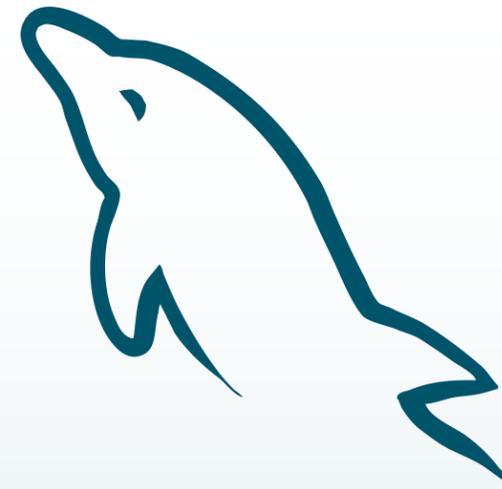


Banco de Dados II

Selecionando Registros

Agrupamento / Funções de Agregação

Consultas com AGRUPAMENTO – cláusula GROUP BY



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 em grupos retornados pelo GROUP BY. Para isso, empregamos a cláusula **HAVING**.



Banco de Dados II

Selecionando Registros

Agrupamento / Funções de Agregação

Consultas com AGRUPAMENTO

Cláusula HAVING - Filtrando os resultados de um 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.

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





Banco de Dados II

Selecionando Registros

Agrupamento / Funções de Agregação

Consultas com AGRUPAMENTO

Cláusula HAVING - Filtrando os resultados de um Agrupamento

Exemplos do uso de HAVING

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;
```





Banco de Dados II

Selecionando Registros

Agrupamento / Funções de Agregação

Consultas com AGRUPAMENTO

Cláusula HAVING - Filtrando os resultados de um Agrupamento

Exemplos do uso de HAVING

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;
```





Referências

- **Boson Treinamentos**

www.bosontreinamentos.com.br/mysql/mysql-order-by-consultas-com-ordenacao-13/

