



Geek University

Evolua seu lado geek!

www.geekuniversity.com.br

Funções de Agregação

Funções de Agregação

Contextualizando...

Ao desenvolver sistemas é muito comum que uma aplicação necessite de informações resumidas, como por exemplo obter a menor ou maior venda do dia, o maior salário de um funcionário, o mês que teve mais vendas e etc.

A linguagem **SQL** contém diversas funções nativas para esse fim, que podem ser usadas para agregar um conjunto de valores em um único resultado através de uma consulta.

Nesta aula, iremos conhecer as principais **funções de agregação** da linguagem **SQL**.

Funções de Agregação

Sintaxe...

Uma **função de agregação** processa um conjunto de valores contidos em uma única coluna de uma tabela e retorna um único valor como resultado.

Sua sintaxe é semelhante aquela utilizada em funções encontradas nas linguagens de programação, contudo o parâmetro de entrada é sempre a coluna cujos valores desejamos processar.

Exemplo:

nome-da-funcao(coluna)

Podemos informar no comando **SELECT** uma ou mais **funções de agregação**, de acordo com a necessidade.

Funções de Agregação

Base de Dados de Exemplo...

```
1 CREATE DATABASE agregacao;
2
3 USE agregacao;
4
5 CREATE TABLE categorias(
6     id INT NOT NULL AUTO_INCREMENT,
7     nome VARCHAR(60) NOT NULL,
8     PRIMARY KEY (id)
9 );
10
11 CREATE TABLE produtos(
12     id INT NOT NULL AUTO_INCREMENT,
13     descricao VARCHAR(60) NOT NULL,
14     preco_venda DECIMAL(8,2) NOT NULL,
15     preco_custo DECIMAL(8,2) NOT NULL,
16     id_categoria int NOT NULL,
17     PRIMARY KEY (id),
18     FOREIGN KEY (id_categoria) REFERENCES categorias(id)
19 );
20
21
22 INSERT INTO categorias (nome) VALUES ('Material Escolar');
23 INSERT INTO categorias (nome) VALUES ('Acessório Informática');
24 INSERT INTO categorias (nome) VALUES ('Material Escritório');
25 INSERT INTO categorias (nome) VALUES ('Game');
26
27 INSERT INTO produtos (descricao, preco_venda, preco_custo, id_categoria) VALUES ('Caderno', '5.45', '2.30', 1);
28 INSERT INTO produtos (descricao, preco_venda, preco_custo, id_categoria) VALUES ('Caneta', '1.20', '0.45', 1);
29 INSERT INTO produtos (descricao, preco_venda, preco_custo, id_categoria) VALUES ('Pendrive 32GB', '120.54', '32.55', 2);
30 INSERT INTO produtos (descricao, preco_venda, preco_custo, id_categoria) VALUES ('Mouse', '17.00', '4.30', 2);
31
32
```

Funções de Agregação

Base de Dados de Exemplo...

Comando DDL para criar o banco de dados.

```
1 CREATE DATABASE agregacao;
2
3 USE agregacao;
4
5 CREATE TABLE categorias(
6     id INT NOT NULL AUTO_INCREMENT,
7     nome VARCHAR(60) NOT NULL,
8     PRIMARY KEY (id)
9 );
10
11 CREATE TABLE produtos(
12     id INT NOT NULL AUTO_INCREMENT,
13     descricao VARCHAR(60) NOT NULL,
14     preco_venda DECIMAL(8,2) NOT NULL,
15     preco_custo DECIMAL(8,2) NOT NULL,
16     id_categoria int NOT NULL,
17     PRIMARY KEY (id),
18     FOREIGN KEY (id_categoria) REFERENCES categorias(id)
19 );
20
21
22 INSERT INTO categorias (nome) VALUES ('Material Escolar');
23 INSERT INTO categorias (nome) VALUES ('Acessório Informática');
24 INSERT INTO categorias (nome) VALUES ('Material Escritório');
25 INSERT INTO categorias (nome) VALUES ('Game');
26
27 INSERT INTO produtos (descricao, preco_venda, preco_custo, id_categoria) VALUES ('Caderno', '5.45', '2.30', 1);
28 INSERT INTO produtos (descricao, preco_venda, preco_custo, id_categoria) VALUES ('Caneta', '1.20', '0.45', 1);
29 INSERT INTO produtos (descricao, preco_venda, preco_custo, id_categoria) VALUES ('Pendrive 32GB', '120.54', '32.55', 2);
30 INSERT INTO produtos (descricao, preco_venda, preco_custo, id_categoria) VALUES ('Mouse', '17.00', '4.30', 2);
31
32
```

Funções de Agregação

Base de Dados de Exemplo...

```
1 CREATE DATABASE agregacao;
2
3 USE agregacao;
4
5 CREATE TABLE categorias(
6     id INT NOT NULL AUTO_INCREMENT,
7     nome VARCHAR(60) NOT NULL,
8     PRIMARY KEY (id)
9 );
10
11 CREATE TABLE produtos(
12     id INT NOT NULL AUTO_INCREMENT,
13     descricao VARCHAR(60) NOT NULL,
14     preco_venda DECIMAL(8,2) NOT NULL,
15     preco_custo DECIMAL(8,2) NOT NULL,
16     id_categoria int NOT NULL,
17     PRIMARY KEY (id),
18     FOREIGN KEY (id_categoria) REFERENCES categorias(id)
19 );
20
21
22 INSERT INTO categorias (nome) VALUES ('Material Escolar');
23 INSERT INTO categorias (nome) VALUES ('Acessório Informática');
24 INSERT INTO categorias (nome) VALUES ('Material Escritório');
25 INSERT INTO categorias (nome) VALUES ('Game');
26
27 INSERT INTO produtos (descricao, preco_venda, preco_custo, id_categoria) VALUES ('Caderno', '5.45', '2.30', 1);
28 INSERT INTO produtos (descricao, preco_venda, preco_custo, id_categoria) VALUES ('Caneta', '1.20', '0.45', 1);
29 INSERT INTO produtos (descricao, preco_venda, preco_custo, id_categoria) VALUES ('Pendrive 32GB', '120.54', '32.55', 2);
30 INSERT INTO produtos (descricao, preco_venda, preco_custo, id_categoria) VALUES ('Mouse', '17.00', '4.30', 2);
31
32
```

Comando DML para utilizar o banco de dados.

Funções de Agregação

Base de Dados de Exemplo...

```
1 CREATE DATABASE agregacao;
2
3 USE agregacao;
4
5 CREATE TABLE categorias(
6     id INT NOT NULL AUTO_INCREMENT,
7     nome VARCHAR(60) NOT NULL,
8     PRIMARY KEY (id)
9 );
10
11 CREATE TABLE produtos(
12     id INT NOT NULL AUTO_INCREMENT,
13     descricao VARCHAR(60) NOT NULL,
14     preco_venda DECIMAL(8,2) NOT NULL,
15     preco_custo DECIMAL(8,2) NOT NULL,
16     id_categoria int NOT NULL,
17     PRIMARY KEY (id),
18     FOREIGN KEY (id_categoria) REFERENCES categorias(id)
19 );
20
21
22 INSERT INTO categorias (nome) VALUES ('Material Escolar');
23 INSERT INTO categorias (nome) VALUES ('Acessório Informática');
24 INSERT INTO categorias (nome) VALUES ('Material Escritório');
25 INSERT INTO categorias (nome) VALUES ('Game');
26
27 INSERT INTO produtos (descricao, preco_venda, preco_custo, id_categoria) VALUES ('Caderno', '5.45', '2.30', 1);
28 INSERT INTO produtos (descricao, preco_venda, preco_custo, id_categoria) VALUES ('Caneta', '1.20', '0.45', 1);
29 INSERT INTO produtos (descricao, preco_venda, preco_custo, id_categoria) VALUES ('Pendrive 32GB', '120.54', '32.55', 2);
30 INSERT INTO produtos (descricao, preco_venda, preco_custo, id_categoria) VALUES ('Mouse', '17.00', '4.30', 2);
31
32
```

Comando DDL para criar tabela.

Criamos aqui duas tabelas, 'categorias' e 'produtos'.

Funções de Agregação

Base de Dados de Exemplo...

```
1 CREATE DATABASE agregacao;
2
3 USE agregacao;
4
5 CREATE TABLE categorias(
6     id INT NOT NULL AUTO_INCREMENT,
7     nome VARCHAR(60) NOT NULL,
8     PRIMARY KEY (id)
9 );
10
11 CREATE TABLE produtos(
12     id INT NOT NULL AUTO_INCREMENT,
13     descricao VARCHAR(60) NOT NULL,
14     preco_venda DECIMAL(8,2) NOT NULL,
15     preco_custo DECIMAL(8,2) NOT NULL,
16     id_categoria int NOT NULL,
17     PRIMARY KEY (id),
18     FOREIGN KEY (id_categoria) REFERENCES categorias(id)
19 );
20
21 INSERT INTO categorias (nome) VALUES ('Material Escolar');
22 INSERT INTO categorias (nome) VALUES ('Acessório Informática');
23 INSERT INTO categorias (nome) VALUES ('Material Escritório');
24 INSERT INTO categorias (nome) VALUES ('Game');
25
26 INSERT INTO produtos (descricao, preco_venda, preco_custo, id_categoria) VALUES ('Caderno', '5.45', '2.30', 1);
27 INSERT INTO produtos (descricao, preco_venda, preco_custo, id_categoria) VALUES ('Caneta', '1.20', '0.45', 1);
28 INSERT INTO produtos (descricao, preco_venda, preco_custo, id_categoria) VALUES ('Pendrive 32GB', '120.54', '32.55', 2);
29 INSERT INTO produtos (descricao, preco_venda, preco_custo, id_categoria) VALUES ('Mouse', '17.00', '4.30', 2);
30
31
32
```

Comando DML para
inserir dados em tabelas.

Estamos inserindo 4
registros em cada tabela.

Funções de Agregação

Max...

Funções de Agregação

Max...

A função max analisa um conjunto de valores e retorna o maior entre eles.

Funções de Agregação

Max...Exemplo 1

```
SELECT MAX(preco_venda) FROM produtos;
```

No exemplo acima a função max() retorna o maior preço de venda da tabela produtos.

Funções de Agregação

Max...Exemplo 1

SELECT MAX(preco_venda) FROM produtos;

No exemplo acima a função max() retorna o maior preço de venda da tabela produtos.

#	max(preco_venda)
1	120.54

Funções de Agregação

Max...Exemplo 2

```
SELECT id_categoria, MAX(preco_venda) FROM produtos GROUP BY id_categoria;
```

No exemplo acima estamos buscando o maior preço de venda em todas as categorias de produtos que temos. Ao final estamos agrupando pelo id da categoria.

OBS: Iremos estudar sobre agrupamento e ordenação não nesta aula mas ainda nesta seção.

OBS: Quando utilizamos uma função de agregação em uma consulta juntamente com outros campos devemos sempre utilizar o agrupamento (**GROUP BY**) por algum elemento da consulta.

Funções de Agregação

Max...Exemplo 2

SELECT id_categoria, MAX(preco_venda) FROM produtos GROUP BY id_categoria;

No exemplo acima estamos buscando o maior preço de venda em cada uma das categorias de produtos que temos. Ao final estamos agrupando pelo id da categoria.

OBS: Iremos estudar sobre agrupamento e ordenação não nesta aula mas ainda nesta seção.

OBS: Quando utilizamos uma função de agregação em uma consulta juntamente com outros campos devemos sempre utilizar o agrupamento (**GROUP BY**) por algum elemento da consulta.

#	id_categoria	max(preco_venda)
1	1	5.45
2	2	120.54

Funções de Agregação

Max...Exemplo 3

```
SELECT id_categoria, MAX(preco_venda) FROM produtos GROUP BY id_categoria  
HAVING MAX(preco_venda) > 10;
```

No exemplo acima estamos buscando o maior preço de venda em cada uma das categorias de produtos que temos onde o preço de venda seja maior que 10,00. Ao final estamos agrupando pelo id da categoria.

OBS: A cláusula **HAVING** funciona quase como um **WHERE**, mas é geralmente utilizada em conjunto com funções de agregação enquanto o **WHERE** é utilizado com o **SELECT**.

Funções de Agregação

Max...Exemplo 3

```
SELECT id_categoria, MAX(preco_venda) FROM produtos GROUP BY id_categoria  
HAVING MAX(preco_venda) > 10;
```

No exemplo acima estamos buscando o maior preço de venda em cada uma das categorias de produtos que temos onde o preço de venda seja maior que 10,00. Ao final estamos agrupando pelo id da categoria.

OBS: A cláusula **HAVING** funciona quase como um **WHERE**, mas é geralmente utilizada em conjunto com funções de agregação enquanto o **WHERE** é utilizado com o **SELECT**.

#	id_categoria	max(preco_venda)
1	2	120.54

Funções de Agregação

Min...

Funções de Agregação

Min...

A função **min** analisa um conjunto de valores e retorna o menor entre eles.

Funções de Agregação

Min...Exemplo

```
SELECT MIN(preco_venda) FROM produtos;
```

No exemplo acima a função min() retorna o menor preço de venda da tabela produtos.

Funções de Agregação

Min...Exemplo

```
SELECT MIN(preco_venda) FROM produtos;
```

No exemplo acima a função min() retorna o menor preço de venda da tabela produtos.

#	min(preco_venda)
1	1.20

Funções de Agregação

Sum...

A função **sum()** realiza a soma dos valores em uma única coluna e retorna esse resultado.

Funções de Agregação

Sum...Exemplo

```
SELECT SUM(preco_venda) FROM produtos WHERE id_categoria = 1;
```

No exemplo acima somamos todos os preços de venda dos produtos onde a categoria seja igual a 1.

Funções de Agregação

Sum...Exemplo

SELECT SUM(preco_venda) FROM produtos WHERE id_categoria = 1;

No exemplo acima somamos todos os preços de venda dos produtos onde a categoria seja igual a 1.

#	sum(preco_venda)
1	6.65

Funções de Agregação

AVG...

Funções de Agregação

AVG...

Com a função **avg()** podemos calcular a média aritmética dos valores em uma única coluna.

Funções de Agregação

AVG...Exemplo

```
SELECT AVG(preco_venda) FROM produtos;
```

No exemplo acima estamos calculando a média aritmética dos preços de venda da tabela produtos.

Funções de Agregação

AVG...Exemplo

SELECT AVG(preco_venda) FROM produtos;

No exemplo acima estamos calculando a média aritmética dos preços de venda da tabela produtos.

#	avg(preco_venda)
1	36.047500

Funções de Agregação

AVG...Exemplo

SELECT AVG(preco_venda) FROM produtos;

No exemplo acima estamos calculando a média aritmética dos preços de venda da tabela produtos.

#	avg(preco_venda)
1	36.047500

Você deveria estar se perguntando agora: Como fazer para mostrar apenas 2 casas decimais?

Funções de Agregação

AVG...Exemplo

```
SELECT AVG(preco_venda) FROM produtos;
```

No exemplo acima estamos calculando a média aritmética dos preços de venda da tabela produtos.

#	avg(preco_venda)
1	36.047500

Você deveria estar se perguntando agora: Como fazer para mostrar apenas 2 casas decimais?

Sorte sua que está estudando na Geek University ;)

Funções de Agregação

ROUND...

Utilizamos a função `round()` para arredondar valores e desta forma especificar quantas casas decimais queremos apresentar o valor.

Funções de Agregação

ROUND...Exemplo

```
SELECT ROUND(AVG(preco_venda), 2) FROM produtos;
```

No exemplo acima, estamos calculando a média aritmética de todos os preços de venda da tabela produtos e apresentando o valor com duas casas decimais.

Funções de Agregação

ROUND...Exemplo

SELECT ROUND(AVG(preco_venda), 2) FROM produtos;

No exemplo acima, estamos calculando a média aritmética de todos os preços de venda da tabela produtos e apresentando o valor com duas casas decimais.

#	ROUND(AVG(preco_venda), 2)
1	36.05

Funções de Agregação

COUNT...

Funções de Agregação

COUNT...

A função count() retorna o total de linhas selecionadas.

Podemos passar por parâmetro o nome da coluna ou um asterisco.

Por padrão, quando informado o nome de uma coluna, valores do tipo NULL são ignorados, mas quando informado um asterisco (*) todas as linhas serão contabilizadas.

Funções de Agregação

COUNT...Exemplo

SELECT COUNT(preco_venda) AS Quantidade FROM produtos WHERE id_categoria = 1;

No exemplo acima estamos contando quantos produtos da categoria 1 temos cadastrados.

Funções de Agregação

COUNT...Exemplo

SELECT COUNT(preco_venda) AS Quantidade FROM produtos WHERE id_categoria = 1;

No exemplo acima estamos contando quantos produtos da categoria 1 temos cadastrados.

#	Quantidade
1	2



Geek University

Evolua seu lado geek!

www.geekuniversity.com.br