

Funções De Agregação, Agrupamento e Ordenação

Utilizarei o banco MySQL, atualize a sintaxe do SQL para a versão do seu banco, caso necessário

Funções de agregação

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

Tabela Produtos

```
CREATE TABLE produtos(  
    id BIGINT NOT NULL AUTO_INCREMENT,  
    descricao VARCHAR(100) NOT NULL,  
    preco_venda DECIMAL(10,2) NOT NULL,  
    preco_custo DECIMAL(10,2) NOT NULL,  
    quantidade_em_estoque BIGINT NOT NULL,  
    PRIMARY KEY(id)  
);  
  
INSERT INTO produtos(id, descricao, preco_venda, preco_custo, quantidade_em_estoque) VALUES (1,'CADERNO', 5.45,2.30,10);  
  
INSERT INTO produtos(id, descricao, preco_venda, preco_custo, quantidade_em_estoque) VALUES (2,'CANETA', 1.20,0.45,50);  
  
INSERT INTO produtos(id, descricao, preco_venda, preco_custo, quantidade_em_estoque) VALUES (3,'PENDRIVE',  
49.90,20.90,25);  
  
INSERT INTO produtos(id, descricao, preco_venda, preco_custo, quantidade_em_estoque) VALUES (4,'MOUSE', 17.90,5.90,30);
```

SELECT * FROM na tabela Produtos

#	id	descricao	preco_venda	preco_custo	quantidade_em_estoque
1	1	CADERNO	5.45	2.30	10
2	2	CANETA	1.20	0.45	50
3	3	PENDRIVE	49.90	20.90	25
4	4	MOUSE	17.90	5.90	30
*	NULL	NULL	NULL	NULL	NULL

Função De Agregação - MAX

A função **MAX** analisa um conjunto de valores de uma determinada coluna, de uma determinada tabela, e retorna o **MAIOR** valor entre as instâncias.

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

#	MAX(preco_venda)
1	49.90

Portugol

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

Selecione o **MAIOR** valor da coluna **preco_venda** entre as instâncias da tabela **produtos**.

#	MAX(preco_venda)
1	49.90

Função De Agregação - MIN

A função **MIN** analisa um conjunto de valores de uma determinada coluna, de uma determinada tabela, e retorna o **MENOR** valor entre as instâncias.

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

#	MIN(preco_venda)
1	1.20

Portugol

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

Selecione o **MAIOR** valor da coluna **preco_venda** entre as instâncias da tabela **produtos**.

#	MIN(preco_venda)
1	1.20

Função De Agregação - SUM

A função **SUM** realiza um somatório de um conjunto de valores de uma determinada coluna, de uma determinada tabela, e retorna o **SOMATÓRIO** dos valores presentes nas instâncias.

```
SELECT SUM(quantidade_em_estoque) FROM produtos;
```

#	SUM(quantidade_em_estoque)
1	115

Portugol

```
SELECT SUM(quantidade_em_estoque) FROM produtos;
```

Selecione o **SOMATÓRIO** dos valores da coluna **quantidade_em_estoque** entre as instâncias da tabela **produtos**.

#	SUM(quantidade_em_estoque)
1	115

Função De Agregação - AVG

A função **AVG** calcula a média aritmética de um conjunto de valores de uma determinada coluna, de uma determinada tabela, e retorna a **MÉDIA** dos valores presentes nas instâncias.

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

#	AVG(preco_venda) ▼
1	18.612500

Portugol

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

Selecione a MÉDIA ARITMÉTICA dos valores da coluna `preco_venda` entre as instâncias da tabela `produtos`.

#	AVG(preco_venda) ▼
1	18.612500

Função De Agregação - ROUND

A função **ROUND** arredonda valores e desta forma especifica quantas casas decimais queremos apresentar o valor de um determinado valor a partir de uma determinada coluna, de uma determinada tabela.

```
SELECT ROUND(AVG(preco_venda), 3) FROM produtos;
```

#	ROUND(AVG(preco_venda), 3)
1	18.613

Portugol

```
SELECT ROUND(AVG(preco_venda), 3) FROM produtos;
```

Selecione a MÉDIA ARITMÉTICA e ARREDONDE EM 3 CASAS DECIMAIS os valores da coluna `preco_venda` entre as instâncias da tabela `produtos`.

#	ROUND(AVG(preco_venda), 3)
1	18.613

Função De Agregação - COUNT

A função **COUNT** retorna o total de linhas selecionadas a partir de uma determinada coluna, de uma determinada tabela. Podemos passar por parâmetro o nome da coluna ou um asterisco.

```
SELECT COUNT(preco_venda) AS Quantidade_De_Instancias FROM produtos;
```

```
SELECT COUNT(*) AS Quantidade_De_Instancias FROM produtos;
```

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

#	Quantidade_De_Instancias
1	4

Portugol

```
SELECT COUNT(preco_venda) AS Quantidade_De_Instancias FROM produtos;
```

Selecione a QUANTIDADE DE ELEMENTOS da coluna preco_venda como Quantidade_De_Instancias entre as instâncias da tabela produtos.

#	Quantidade_De_Instancias
1	4

Função De Agrupamento - GROUP BY

Utilizamos a cláusula GROUP BY para agrupar elementos de uma determinada tabela.

```
SELECT p.descricao AS 'Descrição',  
SUM(p.quantidade_em_estoque) AS 'Quantidade em Estoque'  
FROM produtos AS p  
GROUP BY p.descricao;
```

#	Descrição	Quantidade em Estoque
1	CADERNO	10
2	CANETA	50
3	MOUSE	30
4	PENDRIVE	25

Portugol

```
SELECT p.descricao AS 'Descrição',
```

```
SUM(p.quantidade_em_estoque) AS 'Quantidade em Estoque'
```

```
FROM produtos AS p
```

```
GROUP BY p.descricao;
```

#	Descrição	Quantidade em Estoque
1	CADERNO	10
2	CANETA	50
3	MOUSE	30
4	PENDRIVE	25

SELECIONE a **DESCRICAO DE PRODUTOS COMO** 'Descrição', fazendo o **SOMATÓRIO** dos valores da coluna **quantidade_em_estoque COMO** 'Quantidade em Estoque' entre as instâncias da tabela **produtos COMO p AGRUPANDO PELA COLUNA p.descricao**.

Função De Agrupamento - HAVING

Utilizamos a cláusula HAVING para substituir a cláusula WHERE que não pode ser usada com funções de agrupamento.

```
SELECT p.descricao AS 'Descrição',  
SUM(p.quantidade_em_estoque) AS 'Quantidade em Estoque'  
FROM produtos AS p  
GROUP BY p.descricao  
HAVING SUM(p.quantidade_em_estoque)>25;
```

#	Descrição	Quantidade em Estoque
1	CANETA	50
2	MOUSE	30

Portugol

SELECT p.descricao AS 'Descrição',

SUM(p.quantidade_em_estoque) AS 'Quantidade em Estoque'

FROM produtos AS p

GROUP BY p.descricao,

HAVING SUM(p.quantidade_em_estoque)>25;

#	Descrição	Quantidade em Estoque
1	CANETA	50
2	MOUSE	30

SELECIONE a **DESCRICAO DE PRODUTOS COMO** 'Descrição', fazendo o **SOMATÓRIO** dos valores da coluna **quantidade_em_estoque** **COMO** 'Quantidade em Estoque' entre as instâncias da tabela **produtos** **COMO** p **AGRUPANDO PELA COLUNA** p.descricao quando o **SOMATÓRIO** dos valores da coluna **quantidade_em_estoque** for maior que 25

Função De Ordenação - ORDER BY

A cláusula ORDER BY organiza os dados em ordem alfabética ou numérica. A ordenação pode ser ASC(Ascendente) ou DESC(Descendente) com base em uma determinada coluna, de uma determinada tabela. Por padrão, a ordenação é ascendente, ou seja do menor para o maior.

Função De Ordenação - ORDER BY ASC

```
SELECT id, descricao, preco_venda, preco_custo, quantidade_em_estoque  
FROM produtos ORDER BY quantidade_em_estoque ASC;
```

#	id	descricao	preco_venda	preco_custo	quantidade_em_estoque
1	1	CADERNO	5.45	2.30	10
2	3	PENDRIVE	49.90	20.90	25
3	4	MOUSE	17.90	5.90	30
4	2	CANETA	1.20	0.45	50
*	NULL	NULL	NULL	NULL	NULL

Portugol

```
SELECT id, descricao, preco_venda, preco_custo, quantidade_em_estoque  
FROM produtos ORDER BY quantidade_em_estoque ASC;
```

SELECIONE o id, descricao, preco_venda, preco_custo,
quantidade_em_estoque DE PRODUTOS ORDENANDO POR
quantidade_em_estoque DE FORMA CRESCENTE;

#	id	descricao	preco_venda	preco_custo	quantidade_em_estoque
1	1	CADERNO	5.45	2.30	10
2	3	PENDRIVE	49.90	20.90	25
3	4	MOUSE	17.90	5.90	30
4	2	CANETA	1.20	0.45	50
*	NULL	NULL	NULL	NULL	NULL

Função De Ordenação - ORDER BY DESC

```
SELECT id, descricao, preco_venda, preco_custo, quantidade_em_estoque  
FROM produtos ORDER BY quantidade_em_estoque DESC;
```

#	id	descricao	preco_venda	preco_custo	quantidade_em_estoque
1	2	CANETA	1.20	0.45	50
2	4	MOUSE	17.90	5.90	30
3	3	PENDRIVE	49.90	20.90	25
4	1	CADERNO	5.45	2.30	10
*	NULL	NULL	NULL	NULL	NULL

Portugol

```
SELECT id, descricao, preco_venda, preco_custo, quantidade_em_estoque  
FROM produtos ORDER BY quantidade_em_estoque DESC;
```

SELECIONE o id, descricao, preco_venda, preco_custo,
quantidade_em_estoque DE PRODUTOS ORDENANDO POR
quantidade_em_estoque DE FORMA DECRESCENTE;

#	id	descricao	preco_venda	preco_custo	quantidade_em_estoque
1	2	CANETA	1.20	0.45	50
2	4	MOUSE	17.90	5.90	30
3	3	PENDRIVE	49.90	20.90	25
4	1	CADERNO	5.45	2.30	10
*	NULL	NULL	NULL	NULL	NULL