

BANCO DE DADOS

Profa. Andréa Sakai



FUNÇÕES AGREGADAS

- Estas funções processam conjunto de valores armazenados em um única coluna da tabela e retorna um único valor como resultado.

NOME_DA_FUNCAO(COLUNA)

→ No comando SELECT é possível usar uma ou mais funções de agregação.

FUNÇÕES AGREGADAS

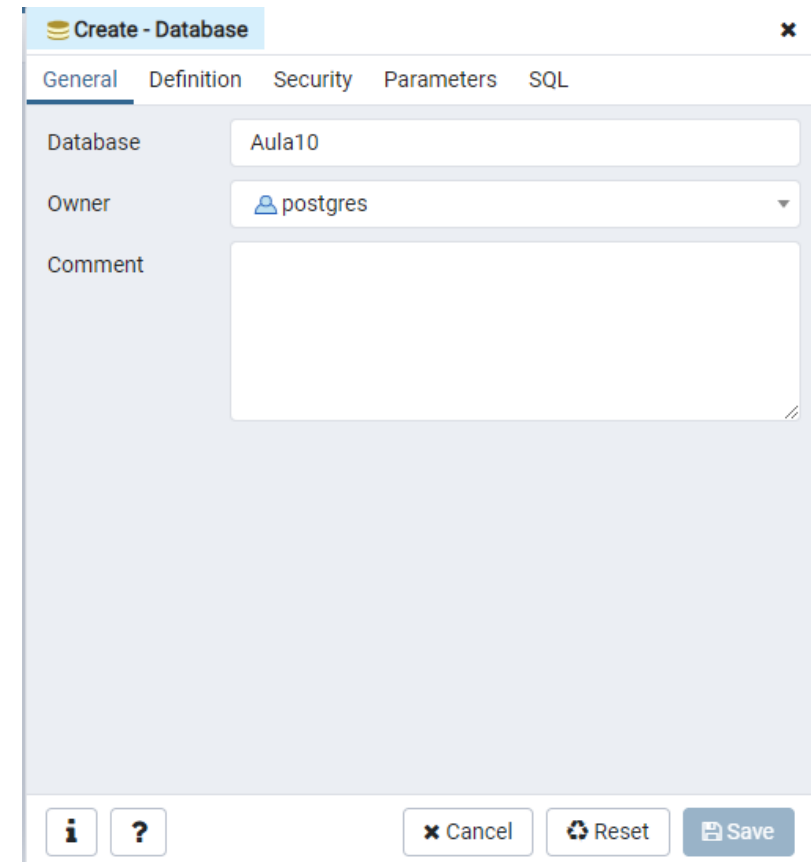
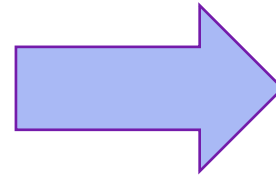
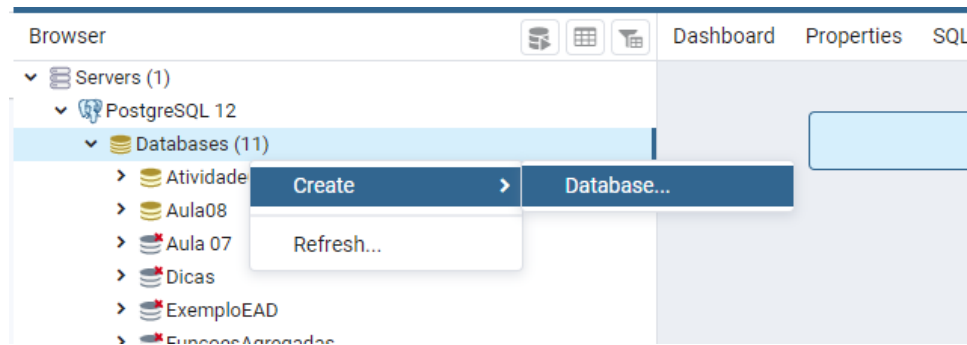
- COUNT()
- SUM()
- MAX()
- MIN()
- AVG()

GROUP BY – O uso da cláusula GROUP BY faz com que os registros sejam retornados na ordem crescente do(s) campo(s) usado(s) para agrupar o resultado da função agregada. Mais um detalhe, **todos os campos na cláusula SELECT** devem aparecer **após a cláusula GROUP BY** separados por vírgula, ou você terá um erro de sintaxe na consulta.

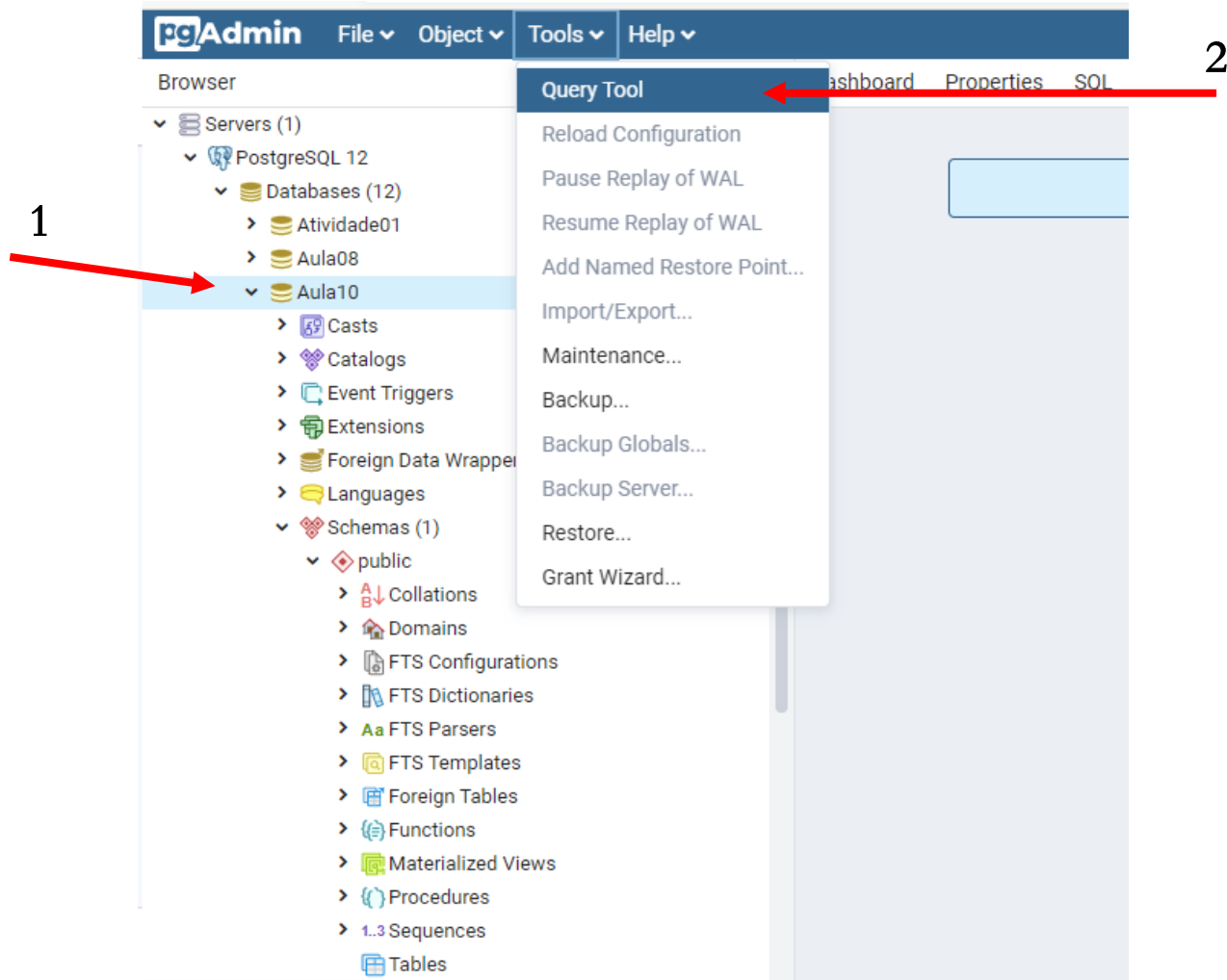
HAVING – A cláusula HAVING vem a complementar a cláusula GROUP BY. Quando usamos GROUP BY, os registros retornados serão todos os que satisfizerem ao critério informado após a palavra WHERE. Porém, podemos querer fazer uma segunda filtragem após termos os resultados dos cálculos das funções agregadas. Neste caso, usamos a cláusula HAVING para fazer esta filtragem posterior.

VAMOS ENTENDER MELHOR COMO USAR ESTAS FUNÇÕES ATRAVÉS DE UMA EXEMPLO.

Passo 1: Vamos criar uma base de dados chamada Aula10



Passo 2: Agora vamos criar uma tabela chamada produtos



1. Crie a
tabela

The screenshot shows a PostgreSQL Query Editor window titled 'Aula10/postgres@PostgreSQL 12'. The 'Query Editor' tab is active, displaying a SQL command to create a table named 'produtos'. The command is as follows:

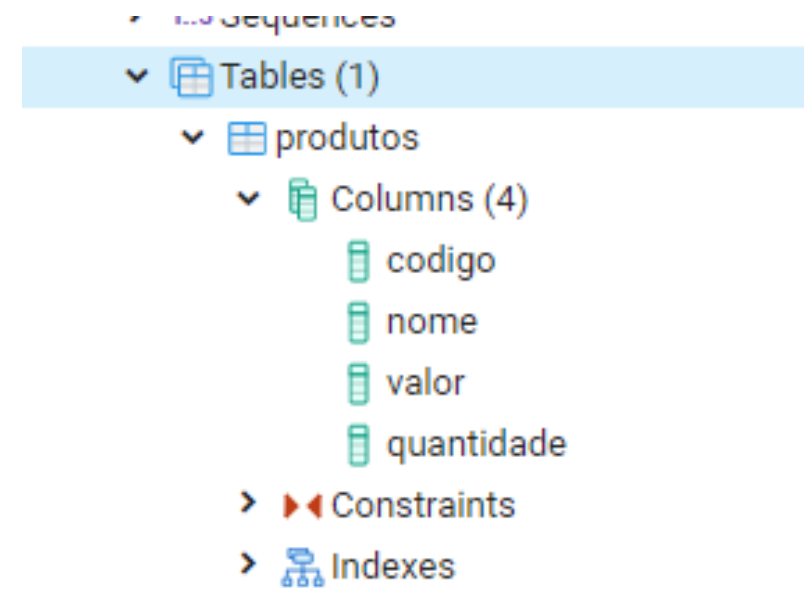
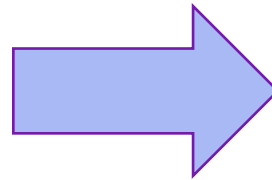
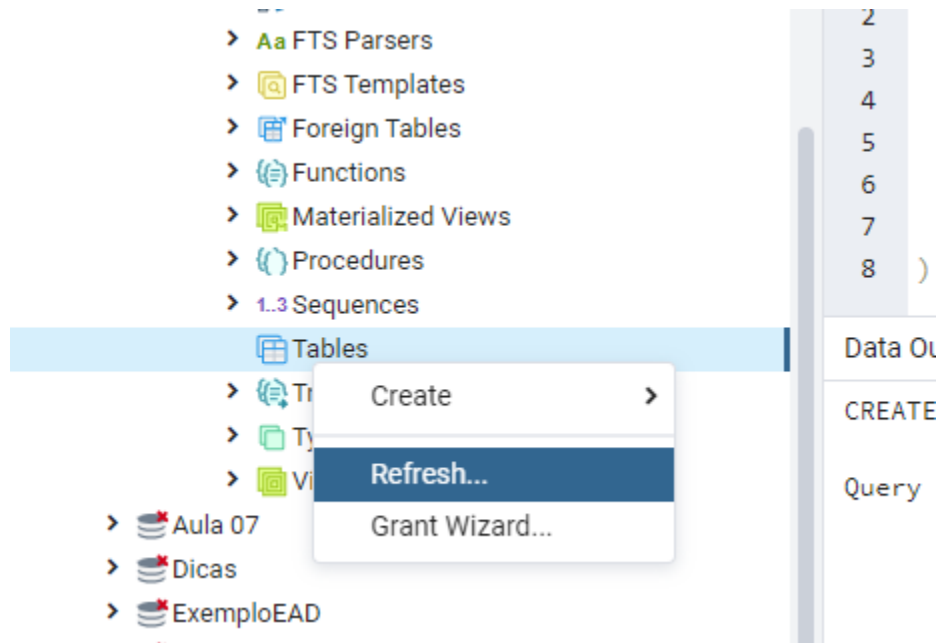
```
1 CREATE TABLE produtos(  
2     codigo      SERIAL,  
3     nome        VARCHAR(50) NOT NULL,  
4     valor       NUMERIC(10,2) NOT NULL,  
5     quantidade  NUMERIC(10,2) NOT NULL,  
6  
7     PRIMARY KEY(codigo)  
8 );
```

Below the query editor, the 'Messages' tab is selected, showing the execution result: 'CREATE TABLE' and 'Query returned successfully in 416 msec.'

2. Execute

3. Verifique o resultado da execução

4. Verifique a criação da tabela fazendo um refresh



Passo 3: Depois da tabela criada insira 6 linhas de registros

```
9  INSERT INTO produtos(nome, valor, quantidade)
10 VALUES('borracha', 4.5, 2),
11 ('caneta', 7.9, 3),
12 ('caderno', 12.3, 4),
13 ('sulfite', 24, 2),
14 ('cola', 8.75, 1),
15 ('régua', 6.0, 1);
```

Data Output	<u>Messages</u>	Query History	Explain	Notifications
-------------	-----------------	---------------	---------	---------------

INSERT 0 6

Query returned successfully in 177 msec.

Passo 4: Faça uma consulta para verificar todos os produtos cadastrados.

16

17 **SELECT * FROM produtos;**

Data Output Messages Query History Explain Notifications

	codigo [PK] integer	nome character varying (50)	valor numeric (10,2)	quantidade numeric (10,2)
1	1	borracha	4.50	2.00
2	2	caneta	7.90	3.00
3	3	caderno	12.30	4.00
4	4	sulfite	24.00	2.00
5	5	cola	8.75	1.00
6	6	régua	6.00	1.00

COUNT() – RETORNA O NÚMERO DE LINHAS.

- Passo 5: Faça uma consulta para verificar quantos produtos estão cadastrados.

Todas as linhas

```
19 SELECT COUNT(*) FROM produtos;
```

	Data Output	Messages	Query History	Explain
	count bigint			
1	6			

Quantidade de linhas

Todas as linhas da coluna nome

18	
19	<code>SELECT COUNT(nome) FROM produtos;</code>
<div>Data OutputMessagesQuery HistoryExplain</div>	
	<div><div>count bigint</div><div></div></div>
1	6

- Passo 6: Vamos fazer um teste.
- Altere a configuração da coluna quantidade retirando o NOT NULL;
- Insira uma nova linha de produto;
- Consulte todos os produtos.

```
21 ALTER TABLE produtos ALTER COLUMN quantidade DROP NOT NULL;
```

Data Output Messages Query History Explain Notifications

ALTER TABLE

Query returned successfully in 74 msec.



```
22
23 INSERT INTO produtos(nome, valor)
24 VALUES('lápiz', 5.3);
```

Data Output Messages Query History Explain Notifications

INSERT 0 1

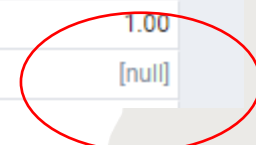
Query returned successfully in 40 msec.



```
25
26 SELECT * FROM produtos;
27
```

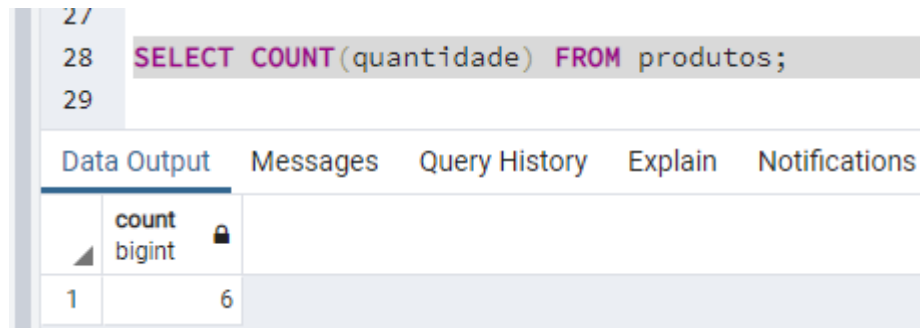
Data Output Messages Query History Explain Notifications

	codigo [PK] integer	nome character varying (50)	valor numeric (10,2)	quantidade numeric (10,2)
1	1	borracha	4.50	2.00
2	2	caneta	7.90	3.00
3	3	caderno	12.30	4.00
4	4	sulfite	24.00	2.00
5	5	cola	8.75	1.00
6	6	régua	6.00	1.00
7	7	lápiz	5.30	[null]



- Passo 6: Agora consulte novamente todos os produtos para saber a quantidade de produtos cadastrados, usando a coluna quantidade.

Como a consulta é na coluna quantidade e nela possui um valor NULL, este registro não entra na contagem.



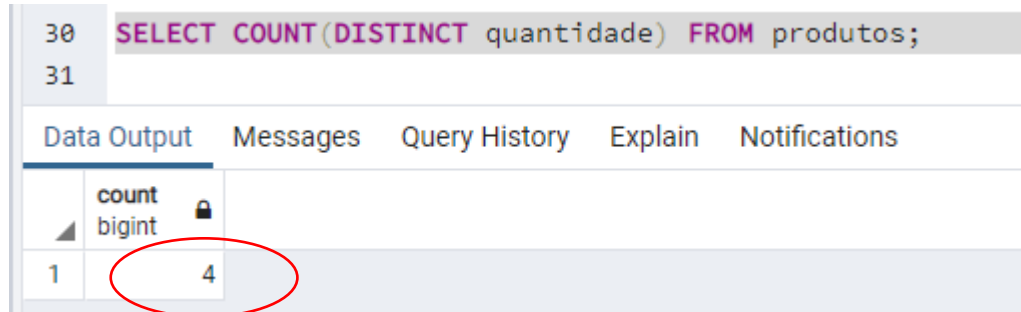
The screenshot shows a SQL query editor with the following text:

```
27  
28 SELECT COUNT(quantidade) FROM produtos;  
29
```

Below the query, there are tabs for "Data Output", "Messages", "Query History", "Explain", and "Notifications". The "Data Output" tab is selected, showing a table with the following structure:

	count bigint
1	6

- Passo 7: Teste novamente o select count, usando o distinct.



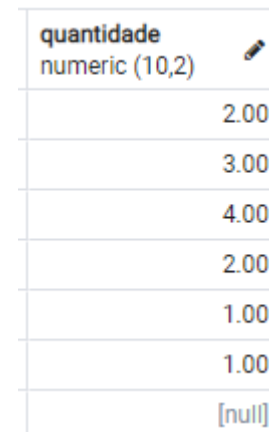
The screenshot shows a SQL query editor with the following text:

```
30 SELECT COUNT(DISTINCT quantidade) FROM produtos;  
31
```

Below the query, there are tabs for "Data Output", "Messages", "Query History", "Explain", and "Notifications". The "Data Output" tab is selected, showing a table with the following structure:

	count bigint
1	4

The value "4" in the result table is circled in red.



The screenshot shows a table with the following structure:

quantidade numeric (10,2)
2.00
3.00
4.00
2.00
1.00
1.00
[null]

Os valores distintos das linhas da coluna quantidade é 1, 2, 3,4, totalizando 4 valores de quantidade.

SUM() – DEVOLVE A SOMA

- Passo 7: Qual o valor total em produtos cadastrados?

```
33  
34 SELECT SUM(valor) FROM produtos;
```

Data Output		Messages	Query History	Explain
	sum numeric			
1	68.75			

- Vamos criar um nome para coluna da somatória.

```
33  
34 SELECT SUM(valor) as total FROM produtos;
```

Data Output		Messages	Query History	Explain	Notificati
	total numeric				
1	68.75				

MAX () – RETORNA O MAIOR VALOR

- Passo 8: Qual o produto de maior valor?

```
38 SELECT MAX(valor) as valor_maximo FROM produtos;
```

```
39
```

Data Output Messages Query History Explain Notifications

	valor_maximo numeric	
1	24.00	

MIN() – RETORNA O MENOR VALOR

- Passo 9: Qual o produto de menor valor?

```
40 SELECT MIN(valor) as valor_minimo FROM produtos;
```

```
41
```

Data Output Messages Query History Explain Notifications

	valor_minimo numeric	
1	4.50	

AVG () – RETORNA O VALOR MÉDIO

- Passo 10: Qual o valor médio dos produtos cadastrados?

41

```
42 SELECT AVG(valor) as media FROM produtos;
```

43

Data Output Messages Query History Explain Notifications



media
numeric



1

9.8214285714285714

- Para fazer o arredondamento de um valor decimal usamos a função ROUND()

O resultado da função AVG
será um número decimal

Quantidade de
casas decimais

```
44 SELECT ROUND(AVG(valor),2) as media FROM produtos;
```

```
45
```

Data Output

Messages

Query History

Explain

Notifications

media
numeric



1

9.82

- Agora vamos criar uma outra tabela chamada funcionários.

```
46 CREATE TABLE funcionarios(  
47     id_func      SERIAL,  
48     nome         VARCHAR(50),  
49     cargo        VARCHAR(50),  
50     salario      NUMERIC(10,2),  
51     dt_inicio    DATE,  
52     PRIMARY KEY(id_func)  
53 );  
54
```

Data Output Messages Query History Explain Notifications

CREATE TABLE

Query returned successfully in 164 msec.

- Vamos inserir dados nesta tabela.

```
54
55 INSERT INTO funcionarios(nome, cargo, salario, dt_inicio)
56 VALUES('Ana','vendedor', 1200,'2021-01-02'),
57 ('Bruna','vendedor', 1500,'2021-04-12'),
58 ('Claudia','caixa',3200,'2021-02-10'),
59 ('Fernanda','vendedor', 2000,'2021-01-07'),
60 ('Carlos','vendedor', 1200,'2021-01-02'),
61 ('Mateus','gerente', 4500,'2020-09-10'),
62 ('Daniel','vendedor', 1200,'2021-03-14'),
63 ('Julia','estoquista', 1350,'2021-05-20'),
64 ('Maria','vendedor', 1300,'2021-02-02'),
65 ('Caio','estoquista', 1300,'2021-09-09'),
66 ('Luis','limpeza', 1100,'2021-01-24');
```







Data Output Messages Query History Explain Notifications

INSERT 0 11

Query returned successfully in 41 msec.

- Faça uma consulta para visualizar todos os dados dos funcionários.

```
68 SELECT * FROM funcionarios;  
69
```

Data Output		Messages	Query History	Explain	Notifications
	id_func [PK] integer 	nome character varying (50) 	cargo character varying (50) 	salario numeric (10,2) 	dt_inicio date 
1	1	Ana	vendedor	1200.00	2021-01-02
2	2	Bruna	vendedor	1500.00	2021-04-12
3	3	Claudia	caixa	3200.00	2021-02-10
4	4	Fernanda	vendedor	2000.00	2021-01-07
5	5	Carlos	vendedor	1200.00	2021-01-02
6	6	Mateus	gerente	4500.00	2020-09-10
7	7	Daniel	vendedor	1200.00	2021-03-14
8	8	Julia	estoquista	1350.00	2021-05-20
9	9	Maria	vendedor	1300.00	2021-02-02
10	10	Caio	estoquista	1300.00	2021-09-09
11	11	Luis	limpeza	1100.00	2021-01-24


- Qual o total de salários pagos a todos os funcionários?

69

```
70 SELECT SUM(salario) as total_salario FROM funcionarios;
```

71

Data Output Messages Query History Explain Notifications

	total_salario numeric	
1	19850.00	

- Qual o total de salários pagos para cada tipo de cargo?
- Antes de responder esta pergunta vamos fazer uma consulta apresentando os cargos e salários de todos os funcionários.

71

72 **SELECT** cargo, salario **FROM** funcionarios;

73

Data Output

Messages

Query History

Explain

Notifications

	cargo character varying (50)	salario numeric (10,2)
1	vendedor	1200.00
2	vendedor	1500.00
3	caixa	3200.00
4	vendedor	2000.00
5	vendedor	1200.00
6	gerente	4500.00
7	vendedor	1200.00
8	estoquista	1350.00
9	vendedor	1300.00
10	estoquista	1300.00
11	limpeza	1100.00

- Para melhorar o nosso entendimento vamos colocar a apresentação dos dados por ordem crescente de cargo.

```
73  
74 SELECT cargo, salario FROM funcionarios ORDER BY cargo;  
75
```

	Data Output	Messages	Query History	Explain	Notifications
	cargo character varying (50)		salario numeric (10,2)		
1	caixa		3200.00		
2	estoquista		1300.00		
3	estoquista		1350.00		
4	gerente		4500.00		
5	limpeza		1100.00		
6	vendedor		1200.00		
7	vendedor		1200.00		
8	vendedor		2000.00		
9	vendedor		1300.00		
10	vendedor		1500.00		
11	vendedor		1200.00		

Agora responda olhando a tabela. Quanto a empresa irá pagar no total para os estoquistas?

2650

Qual o total de salários pagos para cada tipo de cargo?

- A ideia para responder a pergunta acima é agrupar os cargos e somar os seus salários.

```
73  
74 SELECT cargo, salario FROM funcionarios ORDER BY cargo;  
75
```

	cargo character varying (50)	salario numeric (10,2)
1	caixa	3200.00
2	estoquista	1300.00
3	estoquista	1350.00
4	gerente	4500.00
5	limpeza	1100.00
6	vendedor	1200.00
7	vendedor	1200.00
8	vendedor	2000.00
9	vendedor	1300.00
10	vendedor	1500.00
11	vendedor	1200.00

Teremos 5 grupos de cargos.

Qual o total de salários pagos para cada tipo de cargo?

- Vamos usar o SUM() para somar os salários e GROUP BY para agrupar os tipos de cargos.

```
76 SELECT cargo, SUM(salario) FROM funcionarios GROUP BY cargo;  
77
```

Data Output Messages Query History Explain Notifications

	cargo character varying (50)	sum numeric	
1	limpeza	1100.00	
2	vendedor	8400.00	
3	caixa	3200.00	
4	gerente	4500.00	
5	estoquista	2650.00	

Qual o total de salários pagos para cada tipo de cargo ordenados pelo cargo?

```
13  
76 SELECT cargo, SUM(salario) FROM funcionarios GROUP BY cargo ORDER BY cargo;  
77
```

Data Output Messages Query History Explain Notifications

	cargo character varying (50)	sum numeric
1	caixa	3200.00
2	estoquista	2650.00
3	gerente	4500.00
4	limpeza	1100.00
5	vendedor	8400.00

Qual o total de salários pagos para os tipos de cargo que ganham um total acima de 3000 ordenados pelo cargo?

```
78 SELECT cargo, SUM(salario)
79 FROM funcionarios
80 GROUP BY cargo
81 HAVING SUM(salario) > 3000
82 ORDER BY cargo;
83 |
```

Data Output Messages Query History Explain Notifications

	cargo character varying (50)	sum numeric	
1	caixa	3200.00	
2	gerente	4500.00	
3	vendedor	8400.00	

Qual o total de salários pagos para os tipos de cargo que possuem data de admissão de março a junho ordenados pelo cargo?

```
85 SELECT cargo, SUM(salario)
86 FROM funcionarios
87 WHERE dt_inicio BETWEEN '2021-03-01' and '2021-06-30'
88 GROUP BY cargo
89 ORDER BY cargo;
```

Data Output	Messages	Query History	Explain	Notifications
cargo	sum			
estoquista	1350.00			
vendedor	2700.00			

```
90
91 SELECT cargo, salario, dt_inicio
92 FROM funcionarios
93 WHERE dt_inicio BETWEEN '2021-03-01' and '2021-06-30'
94 ORDER BY cargo;
```

Data Output	Messages	Query History	Explain	Notifications
cargo	salario	dt_inicio		
estoquista	1350.00	2021-05-20		
vendedor	1500.00	2021-04-12		
vendedor	1200.00	2021-03-14		

A clausula HAVING cria uma condição para função agregada.

Qual a quantidade de funcionários para cada tipo de cargo?

```
96 SELECT cargo, COUNT(*)
97 FROM funcionarios
98 GROUP BY cargo;
```

Data Output		Messages	Query History	Explai
	cargo		count	
	character varying (50)	🔒	bigint	🔒
1	limpeza		1	
2	vendedor		6	
3	caixa		1	
4	gerente		1	
5	estoquista		2	

Qual para cada tipo de cargo apresente aqueles que possui mais de 1 funcionário?

```
100 SELECT cargo, COUNT(*)
101     FROM funcionarios
102     GROUP BY cargo
103     HAVING COUNT(*) > 1;
```

Data Output		Messages	Query History	Explain
	cargo character varying (50)		count bigint	
1	vendedor		6	
2	estoquista		2	

Exiba os tipos de cargos com o total de salários pagos pela empresa e o número de funcionários.

115	SELECT	cargo,	MAX(salario),	COUNT(*)	
116		FROM	funcionarios		
117		GROUP BY	cargo;		

	Data Output	Messages	Query History	Explain	No
	<div><div>cargo</div><div>character varying (50)</div></div>	<div><div></div></div>	<div><div>max</div><div>numeric</div></div>	<div><div>count</div><div>bigint</div></div>	
1	limpeza		1100.00	1	
2	vendedor		2000.00	6	
3	caixa		3200.00	1	
4	gerente		4500.00	1	
5	estoquista		1350.00	2	

Exiba os tipos de cargos com o total de salários pagos acima de 2000 pela empresa e o número de funcionários.

```
120 SELECT cargo,MAX(salario), COUNT(*)
121     FROM funcionarios
122     GROUP BY cargo
123     HAVING MAX(salario) > 2000;
```

Data Output		Messages	Query History	Explain	Notifications
	cargo character varying (50)		max numeric	count bigint	
1	caixa		3200.00	1	
2	gerente		4500.00	1	