

GROUP BY



INSTITUTO FEDERAL
Rio Grande do Sul
Campus Erechim



Definição

- Com o uso do **GROUP BY**, é possível agruparmos diversos registros com base em uma ou mais colunas de uma tabela.
- Por exemplo, os produtos de uma loja podem ser agrupados pelo **preço** (maior valor, menor valor), pelo **tipo** (livros, filmes, séries), etc.



Definição

- **As funções de agrupamento (ou agregadores)** operam em conjuntos de linhas:

COUNT (coluna)	Retorna a quantidade de ocorrências
SUM (coluna)	Retorna o somatório dos valores de um conjunto
AVG (coluna)	Retorna a media obtida entre os valores de um conjunto
MIN (coluna)	Retorna o menor valor de um conjunto
MAX (coluna)	Retorna o maior valor de um conjunto



Sintaxe

```
SELECT [coluna,] funcao_de_agg(coluna)
FROM tabela
[WHERE condição]
[GROUP BY coluna]
[HAVING condição]
[ORDER BY coluna]
```



Caso do COUNT

COUNT (*)	Retorna o número de linhas em uma tabela, inclusive linhas duplicadas e linhas contendo nulos em alguma coluna
COUNT (coluna)	Retorna o número de linhas não nulas da coluna, identificadas por coluna



Exemplo

EMPLOYEE_ID	FIRST_NAME	LAST_NAME	JOINING_DATE	JOB_ID	SALARY	DEPARTMENT_ID
100	Gerald	Cambrault	34675	AD_PRES	5500	10
101	Renske	Ladwig	34837	AD_VP	15000	20
102	Janette	King	35230	AD_VP	7000	20
103	Sarath	Sewall	35477	IT_PROG	12000	30
104	William	Gietz	35627	IT_PROG	5100	30
105	Jennifer	Whalen	35662	IT_PROG	4900	30
106	Britney	Everett	35733	IT_PROG	5800	30
107	Anthony	Cabrio	35788	IT_PROG	5600	30
108	Alexis	Bull	35861	FI_MGR	7500	40
109	Adam	Fripp	36033	FI_ACCOUNT	8000	40
110	James	Marlow	36066	FI_ACCOUNT	9000	50
111	James	Landry	36174	FI_ACCOUNT	8500	50
112	Payam	Kaufling	36260	FI_ACCOUNT	9500	50
113	Shelley	Higgins	36480	FI_ACCOUNT	8500	50
114	Shanta	Vollman	36501	PU_MAN	10500	50
115	Irene	Mikkilineni	36506	PU_CLERK	10000	50
116	Mozhe	Atkinson	36593	PU_CLERK	9500	50

Acima temos a base de dados



NECESSIDADE: Soma dos salários dos empregados agrupando por departamento

INSTITUTO FEDERAL
Rio Grande do Sul
Campus Erechim

Soma dos salários dos empregados agrupando por departamento

Employees

DEPARTMENT_ID	SALARY
10	5500
20	15000
20	7000
30	12000
30	5100
30	4900
30	5800
30	5600
40	7500
40	8000
50	9000
50	8500
50	9500
50	8500
50	10500
50	10000
50	9500

5500

22000

33400

15500

65550

*Sum of
Salary in
Employees
table for
each
department*

SUM

DEPARTMENT_ID	SUM(SALARY)
10	5500
20	22000
30	33400
40	15500
50	65550



SQL

```
SELECT cod_departamento, sum(salario)
FROM empregado
GROUP BY cod_departamento;
```



GROUP BY

- Número de empregados do departamento 1, por sexo

Sexo	Total de Empregados
M	3
F	1

```
SELECT e.sexo 'Sexo', count(e.nome)
      'Total de Empregados'
FROM DEPARTAMENTO d, EMPREGADO e
WHERE d.codDepto = e.codDepto
AND d.codDepto = 1
GROUP BY e.sexo
ORDER BY count(e.nome) DESC;
```



GROUP BY

- Filtrar o resultado do GROUP BY
 - Usar a cláusula HAVING

```
SELECT d.nome, e.sexo, count(e.nome)
FROM DEPARTAMENTO d, EMPREGADO e
WHERE d.codDepto = e.codDepto
GROUP BY d.codDepto, e.sexo
      HAVING count(e.nome) > 1
ORDER BY count(e.nome) DESC;
```

ALERTA

Todas as colunas e expressões no SELECT, que não tenham agregador, devem aparecer na cláusula GROUP BY

GROUP BY

Quantidade e o valor total das faturas de cada conveniada

```
SELECT
    cod_conveniada,
    count(num_fatura),
    sum(vlr_fatura)
FROM fatura
GROUP BY cod_conveniada
HAVING SUM(vlr_fatura) > 30000;
```

HAVING é usado
para filtrar o
conjunto que foi
agrupado



GROUP BY x DISTINCT

- Quando usamos o **GROUP BY sem agregador**, podemos obter o mesmo resultado do DISTINCT

```
SELECT DISTINCT coluna1  
FROM tabela;
```

```
SELECT coluna1  
FROM tabela  
GROUP BY coluna1;
```

