

## GROUP BY

Agrupamentos e Funções de Agrupamento

## Preparando base de dados de atividades

- Importe o banco de dados de exemplo completo com os relacionamentos aplicando no workbench o script denominado "empresa\_crpkfk.sql"

### Dados nas tabelas

EMPREGADO_PROJETO						HISTORICO_SALARIO			
CODIGO	RG_EMPREGADO	NUMERO_PROJECTO	HORAS	RG	DAT_INI_SAL	DAT_FIM_SAL	SALARIO		
1	2020	5	10	1010	01/01/2010	30/11/2010	2000.00		
2	3030	10	25	1010	01/12/2010	30/04/2011	4000.00		
3	3030	5	35	2020	01/05/2007	31/12/2007	2500.00		
4	4040	20	50	2020	01/01/2008	30/11/2010	4000.00		
5	5050	20	35	4040	01/01/2009	31/08/2009	1500.00		
6	8080	5	70	4040	01/10/2010	30/09/2010	3000.00		
7	8080			4040	01/10/2010	31/10/2013	3500.00		
				7070	01/01/2008	31/07/2008	1000.00		

  

DEPENDENTE						DEPARTAMENTO			
CODIGO	RG_RESPONSABEL	NOME_DEPENDENTE	NASCIMENTO	RELACIONAMENTO	SEXO	NUMERO	NOME	RG_GERENTE	
1	1010	Jorge	1986-12-27	Filho	M	1	Contabilidade	1010	
2	1010	Luiz	1979-11-18	Filho	M	2	Engenharia Civil	3030	
3	2020	Fernanda Carla	1969-02-14	Cônjugue	F	3	Engenharia Mecânica	2020	
4	4040	Antônio	1980-01-01	Pai	M	4	Industrial		
5	3030	André	1990-05-01	Filho	F				
6	8080	Ana Maria	1980-06-30	Cônjugue	F				
7	8080	Karla Cristina	1999-08-25	Filha	F				

  

EMPREGADO						PROJETO			
RG	NAME	CPF	DEPTO	RG_SUPERVISOR	SALARIO	DAT_INI_SAL	NUMERO	NAME	LOCALIZACAO
1010	João Luiz	11111	1	1010	6000.00	2011-05-01	1	Finsmeiro	São Paulo
2020	Fernanda	22222	1	1010	5500.00	2008-12-01	2	Mesor	Rio Claro
3030	Ricardo	33333	2	2020	2300.00	2009-05-01	3	Predio Central	Campinas
4040	José	44444	2	3030	3000.00	2012-01-01	5	Águas Limpas	Vitória
5050	Renata	55555	2	3030	1300.00	2012-02-01	10		
6060	Luiz Renato	66666	3	2020	3000.00	2012-05-01	20		
7070	Luiz Fernando	77777	3	6060	2000.00	2008-08-01	25		
8080	João Antonio	88888	1	2020	3950.00	2010-07-01			

### DQL (Data Query Language)

- Select: Group By (múltiplas linhas)

```
SELECT select_list
[FROM <table_name1>
Group By <column_name>,...
```

Execução de código:

```
SELECT RG_responsavel FROM dependente
Group by RG_responsavel
```

### DQL (Data Query Language)

- Select: Group By (múltiplas linhas)

```
select RG_responsavel from dependente
Group by RG_responsavel

(Observe que neste caso temos vários dependentes associados a um RG de responsável)
```

Select RG\_responsavel  
from dependente

Select RG\_responsavel  
from dependente  
group by RG\_responsavel

rg_responsavel	integer
1010	
1010	
2020	
2020	
3030	
3030	
8080	
8080	

rg_responsavel	integer
2020	
3030	
1010	
8080	

### DQL (Data Query Language)

- Select: Funções de agrupamento com e sem Group By

```
SELECT select_list [COUNT, AVG, SUM, MAX, MIN, STDDEV]
[FROM <table_name1>
Group By <column_name>
```

COUNT () - Contagem	MAX() - Valor máximo
AVG() - Média	MIN() - Valor mínimo
SUM() - Soma	STDDEV() - Desvio padrão

## DQL (Data Query Language)

### • Select: Group By (múltiplas linhas)

```
select RG_responsavel from dependente
Group by RG_responsavel
```

(Observe que neste caso temos vários dependentes associados a um RG de responsável)

Select RG_responsavel from dependente	Select RG_responsavel from dependente group by RG_responsavel	Select RG_responsavel, count(*) from dependente group by RG_responsavel
rg_responsavel integer 1010 1010 2020 2020 3030 3030 8080 8080	rg_responsavel integer 2020 3030 1010 8080	rg_responsavel count integer integer 2020 2 3030 1 1010 2 8080 2

### Count(\*) vs Count(atributo) e valores nulos

É importante observar que ao utilizar count (\*) estamos incluindo todos os valores de um determinado campo, inclusive os nulos.

#### exemplos:

Com base na tabela de

empregados temos que:

EMPREGADO	RG	NOME	CPF	DEPTO	RG_SUPERVISOR	SALARIO	DAT_INI_SAL
1010	João Luiz	11111	1	1010	6000.00	2011-05-01	
2020	Fernanda	22222	2	2020	5500.00	2008-12-01	
3030		33333	3	3030	3000.00	2009-01-01	
4040	Jorge	44444	2	3030	3200.00	2010-10-01	
5050	Renata	55555	2	3030	1300.00	2012-02-01	
6060	Luiz Renato	66666	3	2020	3000.00	2012-05-01	
7070	Lúcio Fernando	77777	3	6060	2000.00	2008-08-01	
8080	João Antônio	88888	1	2020	3900.00	2010-07-01	

select count(rg\_supervisor),sum(rg\_supervisor) from empregado

retornará:

Observe portanto, que o valor nulo não foi contabilizado

select count(\*),sum(rg\_supervisor) from empregado

retornará:

Observe portanto, que o valor nulo foi contabilizado

## Group by com erro

- Em agrupamentos não podemos utilizar colunas na seleção sem incluir a coluna no group by (caso aconteça gerará erro!)

```
10 select rg_responsavel, nome_dependente from dependente
group by rg_responsavel
LINE 1: select rg_responsavel, nome_dependente from dependente
          ^
***** Error *****

ERROR: column "dependente.nome_dependente" must appear in the GROUP BY clause or be used in an aggregate function

```

## DQL (Data Query Language)

### • Select: Group By

– (agrupando múltiplas linhas e colunas)

Execução de código:

```
select rg_responsavel,relacao
from dependente
```

Data Output	Explain	Messages	History
rg_responsavel integer 1010 1010 2020 2020 3030 3030 8080 8080	relacao character varying(10) Filho Filho Cônjugue Filho Filho Cônjugue Filha Filha		
1	1010 Filho		
2	1010 Filho		
3	2020 Cônjugue		
4	2020 Filho		
5	3030 Filho		
6	8080 Cônjugue		
7	8080 Filha		

Execução de código:

```
select
rg_responsavel,relacao,count(*)
from dependente
group by rg_responsavel,relacao
```

rg_responsavel	relacao	count
2020	Filho	1
1010	Filho	2
2020	Cônjugue	1
8080	Filha	1
8080	Cônjugue	1
3030	Filho	1

## DQL (Data Query Language)

- Select: Funções de agrupamento com e sem Group By

```
SELECT select_list [COUNT, AVG, SUM, MAX, MIN, STDDEV]
[FROM <table_name>]
Group By <column_name>
```

Execução de código:

select avg(salario) as med_salario from empregado	med_salario numeric 3406,2500 (SEM GROUP BY)
select depto,avg(salario) as med_salario from empregado group by DEPTO	depto med_salario integer numeric 1 5150,0000 3 2500,0000 2 2266,6666 (COM GROUP BY)
Select COUNT(sexo) as qtd_sexo from dependente group by sexo	qtd_sexo integer character(1) 3 F 4 M

## DQL (Data Query Language)

### • Select: Group By (múltiplas linhas)

Execução de código:

```
select * from dependente
group by RG_responsavel
```

CODIGO	RG_RESPONSAVEL	NOME_DEPENDENTE	NASCIMENTO	RELACAO	SEXO
2	1010	Luiz	1979-11-18	Filho	M
3	2020	Fernanda Carla	1969-02-14	Cônjugue	F
5	3030	André	1990-05-01	Filho	F
6	8080	Ana Maria	1980-06-30	Cônjugue	F

Execução de código:

```
select rg_responsavel,count(*) as qtd_dependentes
from dependente
group by rg_responsavel
```

rg_responsavel	qtd_dependentes
1010	1
2020	2
3030	1
8080	2

DQL (Data Query Language)

### • Select: Group By - Filtrando com Having

Execução de código:

```
select rg_responsavel,count(*) as qtd_dependentes  
from dependente  
group by rg_responsavel
```

rg_responsavel	qtd_dependentes
2020	2
3030	1
1010	2
8080	2

Execução de código:

```
select rg_responsavel,count(*) as qtd_dependentes  
from dependente  
group by rg_responsavel  
having rg_responsavel>2020
```

rg_responsavel	qtd_dependentes
8080	2
3030	1

## Exercícios

1. Agrupe os dependentes por tipo de relação
2. Agrupe os dependentes por sexo
3. Mostre a quantidade de dependentes por sexo
4. Mostre o valor do maior salário dos empregados
5. Mostre o valor do menor salário dos empregados
6. Mostre a soma total de todos os salários dos empregados
7. Mostre a soma total dos salários dos empregados por departamento mostrando o código do departamento e os valores para cada um.
8. Mostre a média dos salários, a soma dos salários, o maior salário, o menor salário, e o desvio padrão dos salários.



### REMOÇÃO DE LINHAS DUPLICADAS COM DISTINCT

## Eliminando repetições com distinct

### Exemplo de join com tres tabelas sem distinct

```
Select distinct emp.nome, Rg, RG_EMPREGADO from  
empregado_projeto empproj  
inner join  
empregado emp  
on (emp.rg = empproj.rg_empregado)  
inner join  
projeto proj  
on (proj.NUMERO = empproj.numero_projeto)
```

nome	rg	rg_empregado
Fernanda	2020	2020
Jorge	4040	4040
Ricardo	3030	3030
Renata	5050	5050
João Antonio	8080	8080

## Eliminando repetições com distinct

### Exemplo de join com tres tabelas sem distinct

```
Select emp.nome, Rg, RG_EMPREGADO from  
empregado_projeto empproj  
inner join  
empregado emp  
on (emp.rg = empproj.rg_empregado)  
inner join  
projeto proj  
on (proj.NUMERO = empproj.numero_projeto)
```

nome	rg	rg_empregado
Fernanda	2020	2020
Fernanda	2020	2020
Ricardo	3030	3030
Jorge	4040	4040
Renata	5050	5050
João Antonio	8080	8080

## Em alguns casos podemos obter os mesmos resultados com distinct e group by

```
Select distinct emp.nome, Rg, RG_EMPREGADO from  
empregado_projeto empproj  
inner join  
empregado emp  
on (emp.rg = empproj.rg_empregado)  
inner join  
projeto proj  
on (proj.NUMERO = empproj.numero_projeto)
```

```
Select emp.nome, Rg, RG_EMPREGADO from  
empregado_projeto empproj  
inner join  
empregado emp  
on (emp.rg = empproj.rg_empregado)  
inner join  
projeto proj  
on (proj.NUMERO = empproj.numero_projeto)  
group by emp.nome,rg,rg_empregado
```

nome	rg	rg_empregado
Fernanda	2020	2020
Jorge	4040	4040
Ricardo	3030	3030
Renata	5050	5050
João Antonio	8080	8080

nome	rg	rg_empregado
Fernanda	2020	2020
Jorge	4040	4040
Ricardo	3030	3030
Renata	5050	5050
João Antonio	8080	8080

## Exercícios – Group by com Join

1. Mostre a quantidade de dependentes que cada funcionário possui.
2. Mostre a quantidade de projetos em que cada funcionário está relacionado.
3. Mostre a soma da quantidade de horas envolvidas com projetos para cada funcionário, em ordem decrescente.
4. Mostre a quantidade de funcionários alocados a cada departamento.
5. Mostre a quantidade de funcionários alocados a cada departamento, somente nos departamento onde a quantidade de funcionários é maior ou igual a 2.
6. Mostre a soma total dos salários dos empregados por departamento mostrando o nome do departamento somente para departamentos onde a soma é maior que 6000.

## Atividades no Trabalho em andamento

Implementar no mínimo 5 consultas com Group By nas tabelas que possuem relacionamentos (1:N) ou (M:N).