



Banco de Dados

Utilizando funções de grupo

Prof. Claudiney Sanches

Agenda

- Funções de Grupo
 - Revisão sobre funções de Grupo
 - Cláusula Group By
 - Cláusula Having
- Exercícios
- Bibliografia

A funcionalidade das funções de grupo é de resumir informações, permitindo ser obtida através de grupos de linhas com o uso de grupos ou agrupamento

Funções de grupo operam sobre um conjunto de linhas para dar um único resultado por grupo.

EMPLOYEES

DEPARTMENT_ID	SALARY
90	24000
90	17000
90	17000
60	9000
60	6000
60	4200
50	5800
50	3500
50	3100
50	2600
50	2500
80	10500
80	11000
80	8600
	7000
10	4400

**Maximum salary in
EMPLOYEES table**

MAX(SALARY)
24000

...
20 rows selected.

As funções de grupos são listadas abaixo:

Função	Descrição
SUM	Retorna a soma de N
AVG	Retorna a média aritmética de N
COUNT	Retorna o número de linhas da consulta.
MAX	Retorna o valor máximo de N
MIN	Retorna o valor mínimo de N

Todas estas funções operam sobre um número de linhas (por exemplo, uma tabela inteira) e são portanto funções de GRUPO.

DISTINCT faz uma função de grupo considerar valores não duplicados;

ALL considera todos os valores sua declaração e não é necessária.

Os tipos de dados dos argumentos devem ser alfanuméricos, numéricos ou data onde a expressão é listada.

Todas as funções de grupo exceto o COUNT(*) ignoram os valores nulos.

AVG

Retorna a média aritmética de um grupo de registros.

Para calcular a média salarial dos empregados, faça:

```
SELECT AVG(Salary)
FROM Employees;
```

AVG(SALARY)
6461.68224299065420560747663551401869159

Note que as linhas da tabela Employees são trilhadas num único grupo(uma única linha).

MIN – MAX - SUM

Retorna o menor valor de um grupo de registros.

Uma função de grupo pode ser usada para subconjunto de linhas de uma tabela usando a cláusula WHERE.

Para encontrar o menor salário, maior salário, e a soma dos salários dos funcionários do departamento 30, faça:

```
SELECT MIN(Salary), MAX(Salary), SUM(Salary)
FROM Employees
WHERE Department_ID = 30;
```

MIN(SALARY)	MAX(SALARY)	SUM(SALARY)
2500	11000	24900

COUNT

Retorna a quantidade de registros de um grupo de registros.

Para encontrar o número de empregados do departamento 30, faça:

```
SELECT Count(*)
```

```
FROM Employees
```

```
WHERE Department_ID = 30;
```

COUNT(*)
6

A CLÁUSULA GROUP BY

EMPLOYEES

DEPARTMENT_ID	SALARY
10	4400
20	13000
20	6000
50	5800
50	3500
50	3100
50	2500
50	2600
60	9000
60	6000
60	4200
80	10500
80	8600
80	11000
90	24000
90	17000

4400

9500

3500

6400

10033

Average
salary in
EMPLOYEES
table for each
department

DEPARTMENT_ID	AVG(SALARY)
10	4400
20	9500
50	3500
60	6400
80	10033.3333
90	19333.3333
110	10150
	7000

...

20 rows selected.

OBS: todas os campos no select que não possuírem função de grupo **DEVEM APARECER NA CLÁUSULA GROUP BY.**

```
SELECT department_id, AVG(salary)
FROM employees
GROUP BY department_id ;
```

Porém os campos existentes na cláusula GROUP BY não precisam aparecer no select

```
SELECT AVG(salary)
FROM employees
GROUP BY department_id ;
```

Para calcular a média salarial de cada grupo de cargo, faça:

```
SELECT Department_ID, AVG(Salary)
FROM Employees
Group by Department_ID
Order by Department_ID;
```

[illegible]

AGRUPANDO MAIS QUE UMA COLUNA

EMPLOYEES

DEPARTMENT_ID	JOB_ID	SALARY
90	AD_PRES	24000
90	AD_VP	17000
90	AD_VP	17000
60	IT_PROG	9000
60	IT_PROG	6000
60	IT_PROG	4200
50	ST_MAN	5800
50	ST_CLERK	3500
50	ST_CLERK	3100
50	ST_CLERK	2600
50	ST_CLERK	2500
80	SA_MAN	10500
80	SA_REP	11000
80	SA_REP	8600
...		
20	MK_REP	6000
110	AC_MGR	12000
110	AC_ACCOUNT	8300

20 rows selected.

Add the salaries in the EMPLOYEES table for each job, grouped by department

DEPARTMENT_ID	JOB_ID	SUM(SALARY)
10	AD_ASST	4400
20	MK_MAN	13000
20	MK_REP	6000
50	ST_CLERK	11700
50	ST_MAN	5800
60	IT_PROG	19200
80	SA_MAN	10500
80	SA_REP	19600
90	AD_PRES	24000
90	AD_VP	34000
110	AC_ACCOUNT	8300
110	AC_MGR	12000
	SA_REP	7000

13 rows selected.

```
SELECT    department_id, job_id, SUM(salary)
FROM      employees
GROUP BY  department_id, job_id ;
```

Restrições

- Não use a cláusula WHERE para restringir grupos
- Para restringir grupos usamos a cláusula HAVING
- Você não pode usar funções de grupo na cláusula WHERE

```
SELECT    department_id, AVG(salary)
FROM      employees
WHERE     AVG(salary) > 8000
GROUP BY department_id;
```

```
WHERE     AVG(salary) > 8000
          *
```

ERROR at line 3:

ORA-00934: group function is not allowed here

Não use a cláusula WHERE para restringir grupos

[illegible]

Para mostrar a media salarial e a soma dos salários por cada departamento e cargo, faça:

```
SELECT Department_ID, Job_ID, AVG(Salary), Sum(Salary)
FROM Employees
Group by Department_ID, Job_ID
Order by Department_ID;
```

DEPARTMENT_ID	JOB_ID	AVG(SALARY)	SUM(SALARY)
10	AD_ASST	4400	4400
20	MK_MAN	13000	13000
20	MK_REP	6000	6000
30	PU_CLERK	2780	13900
30	PU_MAN	11000	11000
40	HR_REP	6500	6500
50	SH_CLERK	3215	64300
50	ST_CLERK	2785	55700
50	ST_MAN	7280	36400
60	IT_PROG	5760	28800
More than 10 rows available. Increase rows selector to view more rows.			

Restringindo o Resultado por Grupo

EMPLOYEES

DEPARTMENT_ID	SALARY
90	24000
90	17000
90	17000
60	9000
60	6000
60	4200
50	5800
50	3500
50	3100
50	2600
50	2500
80	10500
80	11000
80	8600
...	...
20	6000
110	12000
110	8300

20 rows selected.

The maximum salary per department when it is greater than \$10,000

DEPARTMENT_ID	MAX(SALARY)
20	13000
80	11000
90	24000
110	12000

```
SELECT    department_id, MAX(salary)
FROM      employees
GROUP BY  department_id
HAVING    MAX(salary)>10000 ;
```

Use a cláusula HAVING se você quiser especificar o qual grupo será mostrado.

Para mostrar a média salarial de todos os departamentos que tiverem a média maior que 6000, faça:

- Relembrando...
- A cláusula WHERE é condição para o SELECT
- A cláusula HAVING é condição para um GROUP BY

```
SELECT Department_ID, AVG(Salary)
FROM Employees
Group by Department_ID
Having AVG(Salary) > 6000
Order by Department ID;
```

DEPARTMENT_ID	AVG(SALARY)
20	9500
40	6500
70	10000
80	8955.88235294117647058823529411764705882
90	19333.3333333333333333333333333333333333
100	8600
110	10150
-	7000

A cláusula HAVING

Para mostrar a quantidade de empregados que tenham o salário acima de 10000 por departamento, faça:

```
SELECT Department_ID, Count(*),  
FROM Employees  
Group by Department_ID  
Having MAX(Salary) > 10000  
Order by Department_ID;
```

DEPARTMENT_ID	COUNT(*)
20	2
30	6
80	34
90	3
100	6
110	2

A cláusula HAVING

Nota:

A cláusula HAVING deve ser colocada depois da cláusula GROUP BY e é utilizada para estabelecer condições dentro das funções de grupo.

A cláusula WHERE não pode ser usada para restringir itens de grupo.

A seguinte declaração da cláusula WHERE é errada.

```
SELECT Department_ID, Count(*)  
FROM Employees  
Where MAX(Salary) > 10000  
Group by Department_ID  
Order by Department_ID;
```



ORA-00934: group function is not allowed here

A cláusula HAVING

Você pode unicamente usar WHERE para restringir linhas individuais. Para restringir colunas de grupos usa-se a cláusula HAVING.

Você pode limitar sua seleção incluindo somente os departamentos 10, 20, 30 e 40, usando a cláusula WHERE quando estiver agrupando por departamento.

```
SELECT Department_ID, Count(*)  
FROM Employees  
Where Department_ID IN (10,20,30,40)  
Group by Department_ID  
Having MAX(Salary) > 10000  
Order by Department_ID;
```

DEPARTMENT_ID	COUNT(*)
20	2
30	6

A Ordem das cláusulas na declaração SELECT.

SELECT	coluna(s)
FROM	tabela(s)
WHERE	condição linha
GROUP BY	coluna(s)
HAVING	condição de grupo de linhas
ORDER BY	coluna(s);

1. **WHERE** para estabelecer condições em linhas (SELECT).
2. **GROUP BY** para configurar grupos
3. **HAVING** para estabelecer condições dentro dos grupos (GROUP BY).

1) Encontre o maior, menor, soma e média salarial da tabela de empregados. Dê alias para as colunas: Maior Salário, Menor Salário, Soma de Todos os Salários e Média Salarial.

Maior Salário	Menor Salário	Soma De Todos Os Salários	Média Salarial
24000	2100	681400	6428,30188679245283018867924528301886793

1 linhas retornadas em 0,00 segundos

[Exportação para CSV](#)

2) Altere o exemplo 1 para agrupar por cargo e acrescente a quantidade de funcionários por função. Crie um alias para a quantidade de funcionários conforme abaixo.

Quantidade De Funcionários	JOB_ID	Maior Salário	Menor Salário	Soma De Todos Os Salários	Média Salarial
5	IT_PROG	9000	4200	28800	5760
1	AC_MGR	12000	12000	12000	12000
1	AC_ACCOUNT	8300	8300	8300	8300
5	ST_MAN	8200	5800	36400	7280
1	PU_MAN	11000	11000	11000	11000
1	AD_ASST	4400	4400	4400	4400
2	AD_VP	17000	17000	34000	17000

3) Determine o número de gerentes sem listá-los. Dê o nome do alias da coluna de "Número de Gerentes"

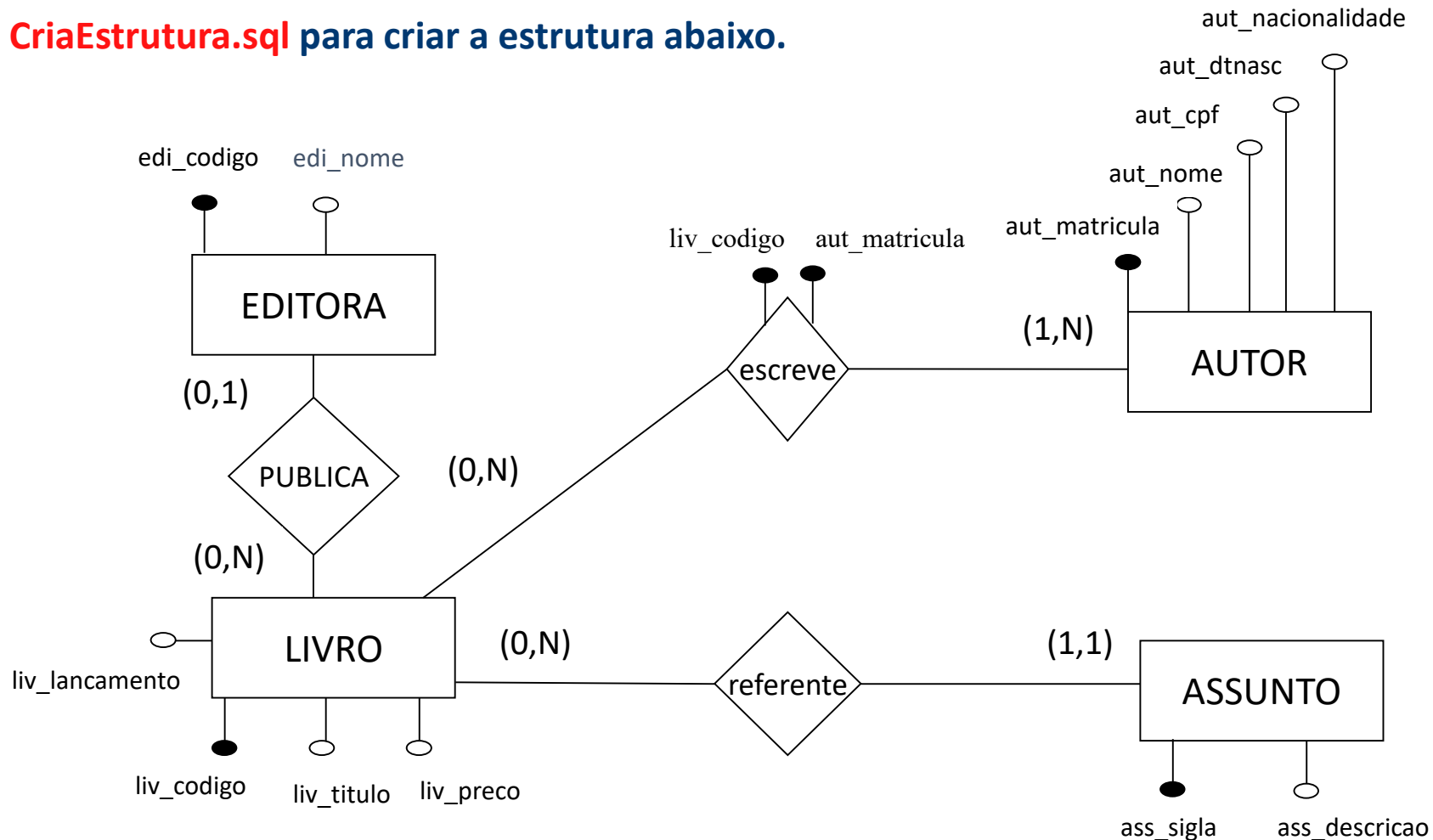
Número De Gerentes
18

4) Encontre a diferença entre o maior e o menor salário, de o alias para a coluna de diferença.

DIFFERENCE
21900

Salve os comandos no arquivo **Trabalho13.sql** e envie

Com base no der abaixo, e os dados que estão no próximo slide, resolva os exercícios propostos. Caso não tenha criado essas tabelas, utilize o Script **CriaEstrutura.sql** para criar a estrutura abaixo.



LIVRO					
liv_codigo	liv_titulo	liv_preco	liv_lancamento	edi_codigo	ass_sigla
1	banco de dados para web	32,20	10/01/1999	1	BAN
2	programando em linguagem c	30,00	01/10/1997	1	PRO
3	progrmando em linguagem c++	115,50	01/11/1998	3	PRO
4	banco de dados na bioinformática	48,00		2	BAN
5	redes de computadores	42,00	01/09/1996	2	RED

ASSUNTO	
ass_sigla	ass_descricao
BAN	Banco de Dados
PRO	Programação
RED	Redes
SIS	Sistema Operacional

EDITORIA	
edi_codigo	edi_nome
1	Mirandela
2	Editora Via Norte
3	Editora Ilhas Tijucas
4	Maria José

ESCREVE	
liv_codigo	aut_matricula
1	1
2	1
3	2
4	3
5	4

AUTOR				
aut_matricula	aut_nome	aut_cpf	aut_dtnasc	aut_nacionalidade
1	Luiz	11122233344	11/11/1980	Brasileira
2	Hugo	22233344455	11/11/1981	Brasileira
3	Joaquim	33344455566	12/01/1980	Portuguesa
4	Regina	44455566677	19/02/1979	Brasileira

5. Exibir a quantidade de livros que existem por sigla.
6. Exibir o maior, o menor e a média dos valores dos livros.
7. Exibir os livros (Título) que o autor “Luiz” escreveu.
8. Exibir os livros (Título), valor, Lançamento, cujo assunto seja “Banco de Dados”.
9. Exibir a quantidade de livros publicados por sigla, exibindo a sigla e a quantidade.
10. Quais livros foram publicados pela editora “Mirandela”.

- [1] Fanderuff, Damaris. Dominando o Oracle 9i: Modelagem e desenvolvimento. São Paulo: Pearson Education do Brasil, 2003.
- [2] Costa, Rogério Luis de C., SQL : guia prático. 2. ed. Rio de Janeiro : Brasport, 2006.
- [3] SILBERSCHATZ, A. Sistema de bancos de dados. São Paulo: Pearson Education do Brasil, 2004.
- [4] Morelli, Eduardo M. Terra, 1996. Oracle 9i Fundamental: Sql, Pl/SQL e Administração. São Paulo: Érica, 2002.