

# Linguagem de Banco de Dados



Cruzeiro do Sul Virtual  
Educação a Distância



# Material Teórico



Funções de Grupo

**Responsável pelo Conteúdo:**

Prof. Ms. Luis Carlos Reis

**Revisão Textual:**

Prof. Ms. Luciano Vieira Francisco



# UNIDADE

## Funções de Grupo



- Pré-Requisito para Praticar
- Conceito
- Função de Grupo Simples
- Função de Grupo Composto



### OBJETIVO DE APRENDIZADO

- Realizar consultas mais complexas e avançadas para melhor filtrar as informações.
- Entender o agrupamento de informações que possuem dados semelhantes.
- Aprender a filtrar o resultado desse agrupamento.





# Orientações de estudo

Para que o conteúdo desta Disciplina seja bem aproveitado e haja uma maior aplicabilidade na sua formação acadêmica e atuação profissional, siga algumas recomendações básicas:



## Assim:

- ✓ Organize seus estudos de maneira que passem a fazer parte da sua rotina. Por exemplo, você poderá determinar um dia e horário fixos como o seu “momento do estudo”.
- ✓ Procure se alimentar e se hidratar quando for estudar, lembre-se de que uma alimentação saudável pode proporcionar melhor aproveitamento do estudo.
- ✓ No material de cada Unidade, há leituras indicadas. Entre elas: artigos científicos, livros, vídeos e sites para aprofundar os conhecimentos adquiridos ao longo da Unidade. Além disso, você também encontrará sugestões de conteúdo extra no item **Material Complementar**, que ampliarão sua interpretação e auxiliarão no pleno entendimento dos temas abordados.
- ✓ Após o contato com o conteúdo proposto, participe dos debates mediados em fóruns de discussão, pois irão auxiliar a verificar o quanto você absorveu de conhecimento, além de propiciar o contato com seus colegas e tutores, o que se apresenta como rico espaço de troca de ideias e aprendizagem.

# Pré-Requisito para Praticar

Como utilizaremos o software *SQLDeveloper* para exemplificarmos os seguintes comandos, precisaremos da conexão *AulaHR*, que foi configurada no tutorial de instalação do software.

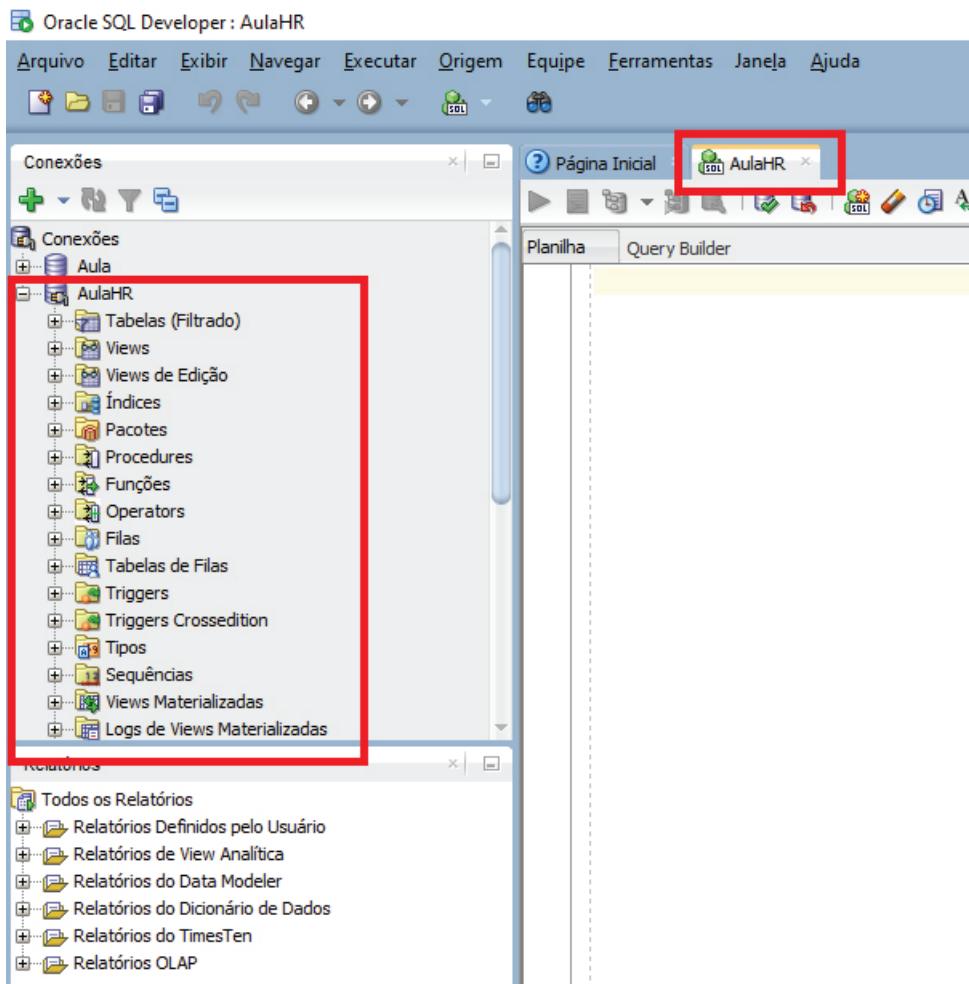


Figura 1

## Conceito

A aplicação das funções de grupo corresponde a resumir informações, permitindo ser obtida por meio de grupos de linhas com o uso de grupos ou agrupamentos de funções.

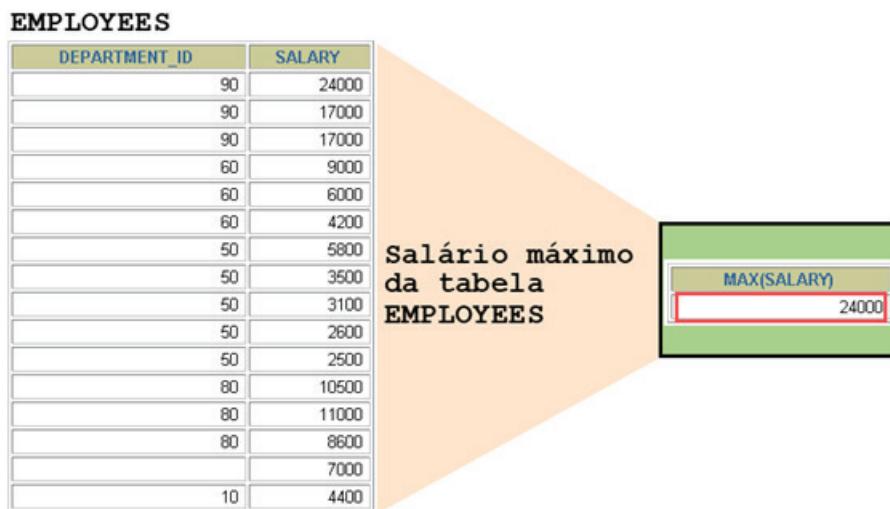


Figura 2

## Função de Grupo Simples

As funções de grupos são assim listadas:

Quadro 1

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 como sua declaração, não sendo necessária.

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

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

### AVG

Retorna a média aritmética de um grupo de registros. Por exemplo, para calcular a média salarial dos empregados, faça:

```
SELECT AVG(Salary)
```

```
FROM Employees;
```

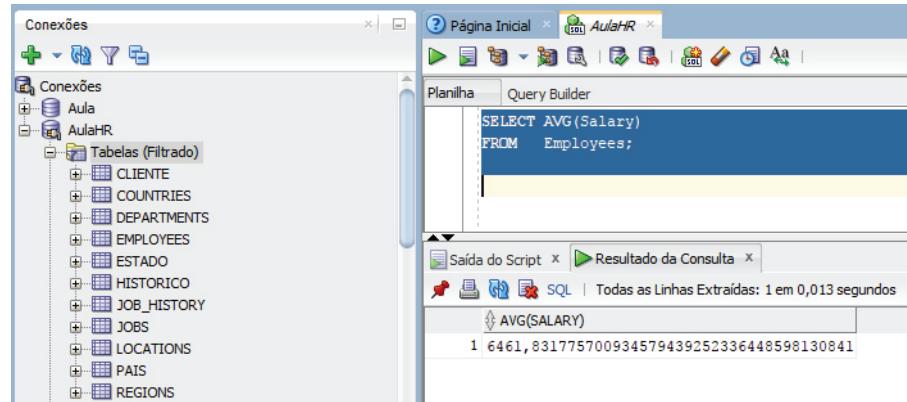


Figura 3

Note que as linhas da tabela *Employees* são trilhadas em um único grupo – em uma única linha.

- *Min – Max – Sum*

Retorna o menor, maior e a somatória dos valores de um grupo de registros.

Uma função de grupo pode ser usada para subconjunto de linhas de uma tabela por meio da cláusula *WHERE*. Assim, para encontrar o menor, maior e a soma dos salários dos funcionários do Departamento 30, faça:

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

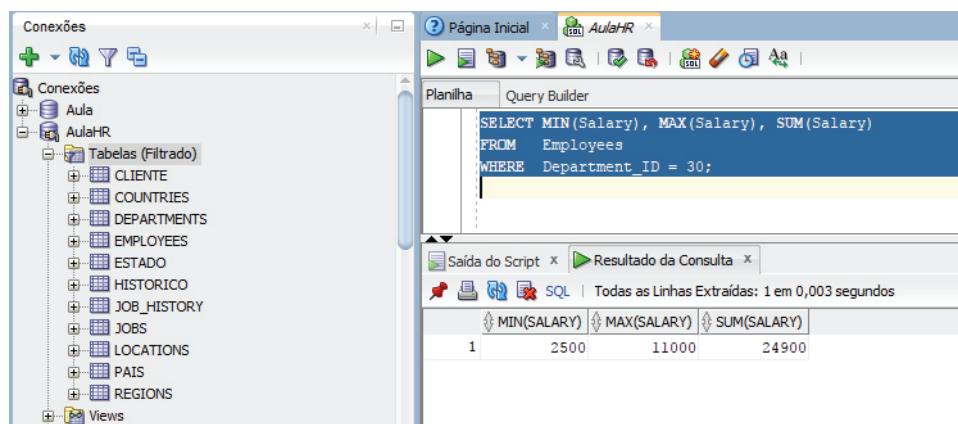


Figura 4

## Count

Retorna a quantidade de registros de um grupo de registros. Por exemplo, para encontrar o número de empregados do Departamento 30, faça:

```
SELECT Count(*)
FROM   Employees
WHERE  Department_ID = 30;
```

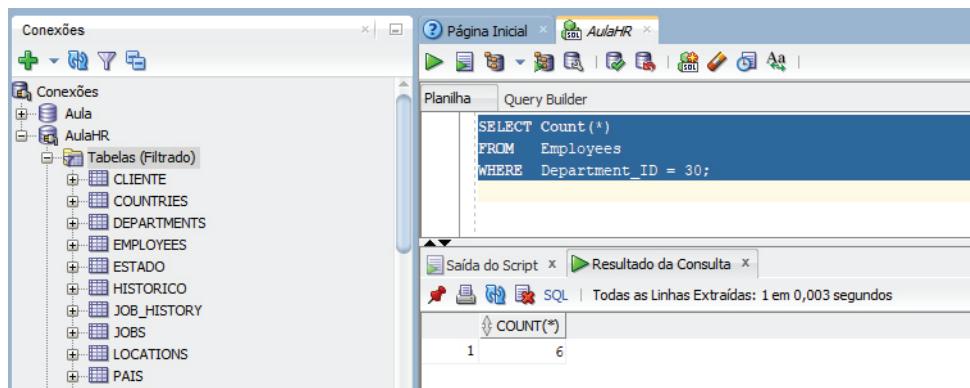


Figura 5

## Função de Grupo Composto

### Cláusula *Group By* – Criando Grupo de Dados

Com o uso do *GROUP BY* é possível agruparmos diversos registros com base em uma ou mais colunas de uma tabela.

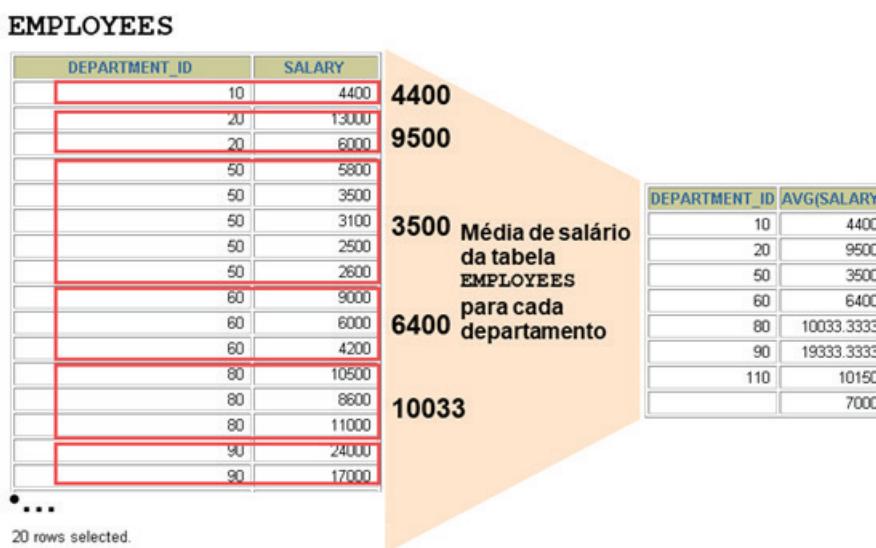


Figura 6



## Importante!

Todos 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;
```

Figura 7

Porém, os campos existentes na cláusula *GROUP BY* não precisam necessariamente aparecer no *select*.

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

Figura 8

## Exemplo 1

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;
```

Figura 9

## Exemplo 2 – Agrupando Mais de 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
60	SA_MAN	10500
60	SA REP	11000
60	SA REP	8600
...		
20	MK REP	6000
110	AC_MGR	12000
110	AC_ACCOUNT	6300

20 rows selected.

**Adiciona o salário na tabela EMPLOYEES Para cada cargo agrupado por departamento**

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

Figura 10

Limitações:

- Não use a cláusula *WHERE* para restringir grupos;
- Para restringir grupos, use 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;
  
```

Figura 11

```

WHERE      AVG(salary) > 8000
*
ERROR at line 3:
ORA-00934: group function is not allowed here
  
```

Figura 12



**Importante!**

Reiterando, não use a cláusula *WHERE* para restringir grupos.

### Exemplo 3 – Excluindo Linhas Quando Estiver Usando o *GROUP BY*

Linhos devem ser excluídos com a cláusula *WHERE* antes da divisão por grupos. Por exemplo, para mostrar a média salarial para cada cargo, excluindo os departamentos 10, 20 e 30, faça:

```
SELECT Department_ID, AVG(Salary)  
FROM Employees  
Where Department_ID Not in (10,20,30)  
Group by Department_ID  
Order by Department_ID;
```

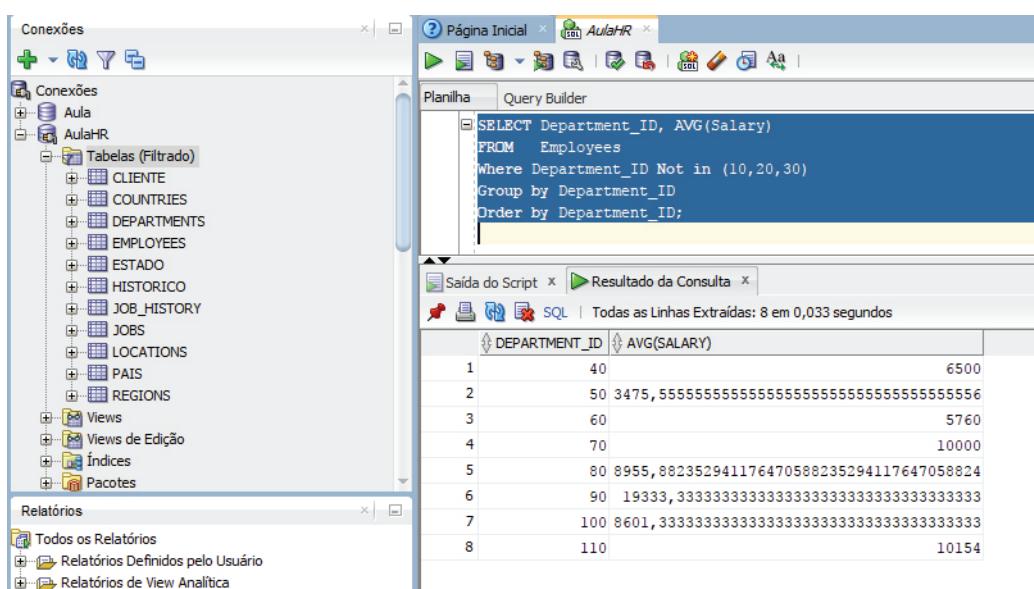


Figura 13

## Exemplo 4

Para mostrar a média 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;
```

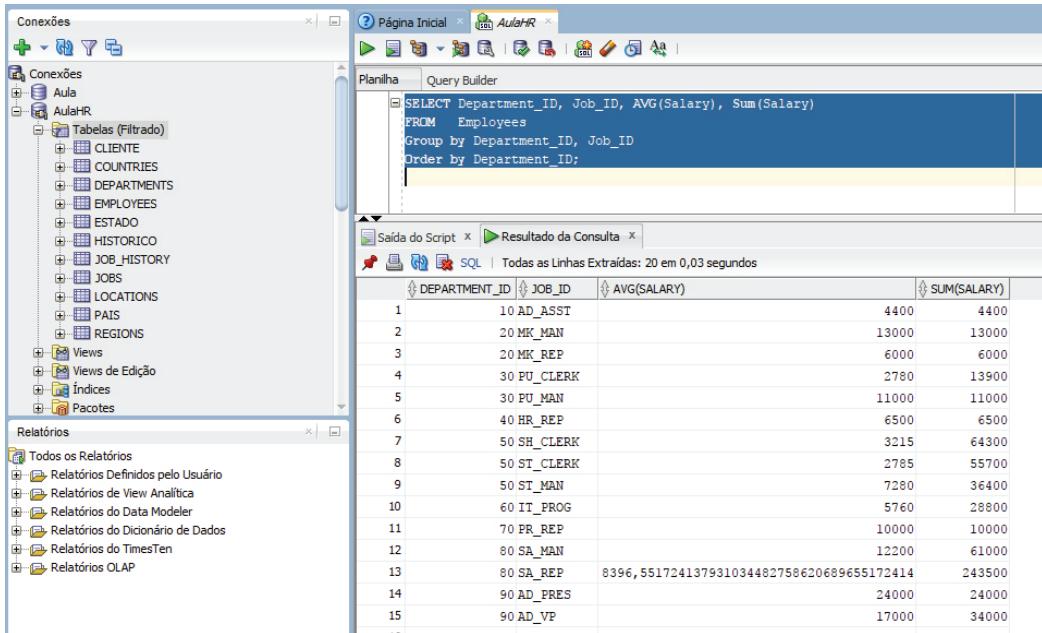


Figura 14

## Cláusula *HAVING* – Restringindo o Resultado por Grupo

Use a cláusula *HAVING* se quiser especificar o grupo a ser mostrado.

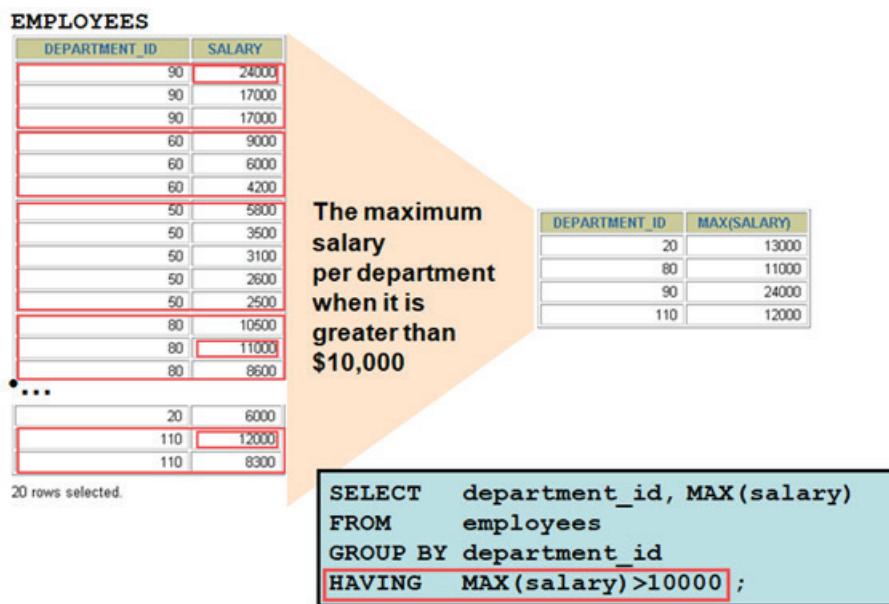


Figura 15

## Exemplo 5

Para mostrar a média salarial de todos os departamentos que tiverem a média maior que 6.000, faça:

*SELECT Department\_ID, AVG(Salary)*

*FROM Employees*

*Group by Department\_ID*

*Having AVG(Salary) > 6000*

*Order by Department\_ID;*

Figura 16

## A cláusula:

- *WHERE* é condição para o *SELECT*;
  - *HAVING* é condição para um *GROUP BY*.

## Exemplo 6

Para mostrar a quantidade de empregados com salário acima de 10.000 por departamento, faça:

*SELECT Department\_ID, Count(\*)*

*FROM Employees*

*Group by Department\_ID*

*Having MAX(Salary) > 10000*

*Order by Department\_ID;*

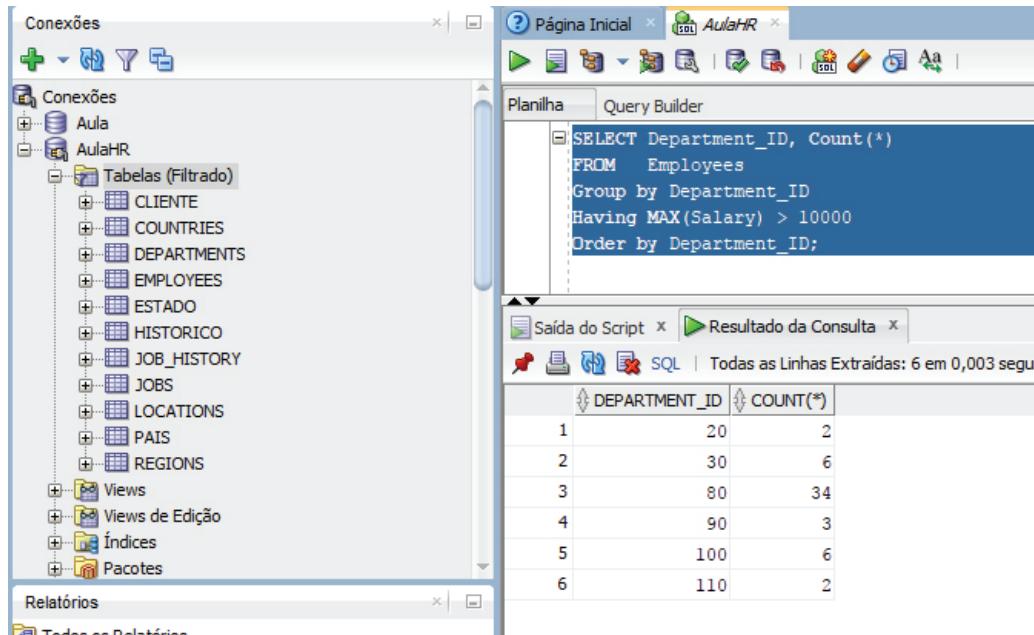


Figura 17

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

Ademais, a cláusula *WHERE* não pode ser utilizada para restringir itens de grupo, de modo que a seguinte declaração da cláusula *WHERE* é incorreta:

```
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

Figura 18

Pode-se usar a cláusula *WHERE* para restringir linhas individuais. Assim, para restringir colunas de grupos, utilize a cláusula *HAVING*.

Você pode limitar a sua seleção incluindo apenas os departamentos 10, 20, 30 e 40, usando a cláusula *WHERE* quando estiver agrupando por departamento, vejamos:

```
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;
```

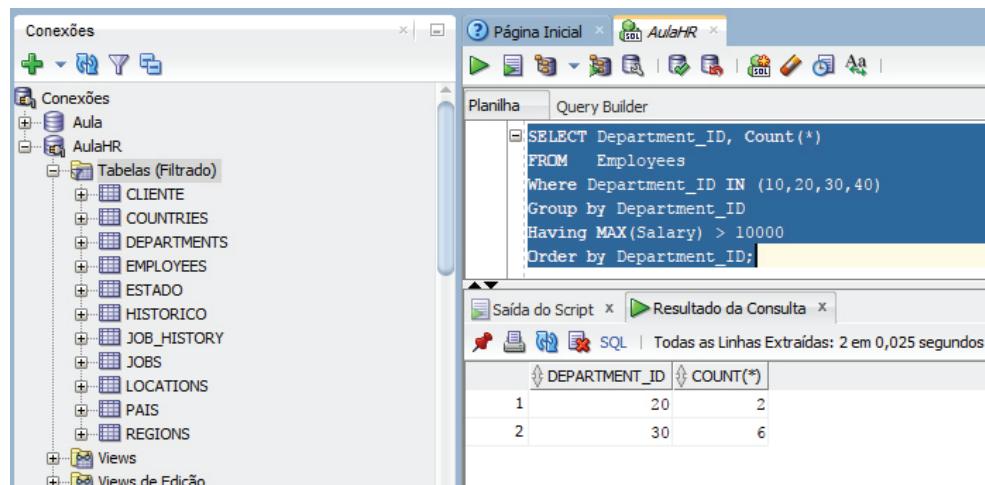


Figura 19

Por fim, a ordem das cláusulas na declaração *SELECT* é:

1. *SELECT* coluna(s).
2. *FROM* tabela(s).
3. *WHERE* condição linha.
4. *GROUP BY* coluna(s).
5. *HAVING* condição de grupo de linhas.
6. *ORDER BY* coluna(s).

# Material Complementar

## Indicações para saber mais sobre os assuntos abordados nesta Unidade:



Livros

### Oracle Database 11g Sql

Como complemento desta unidade, sugiro a leitura dos capítulos IV e VII do livro: PRICE, J. **Oracle Database 11g Sql**. Porto Alegre: Bookman, 2009.

### SQL: Guia Prático

Ler capítulo IV do livro: Costa, Rogério Luis de C. **SQL: guia prático**. 2. ed. Rio de Janeiro : Brasport, 2006.



Vídeos

### SQL with Oracle 10g XE - Using the MIN, MAX, and AVG Functions

<https://youtu.be/jKQECYhtEl4>

### SQL: Max/Min Functions

<https://youtu.be/k8V08eG1gp4>

# Referências

FANDERUFF, D. **Dominando o Oracle 9i**: modelagem e desenvolvimento. São Paulo: Pearson Education do Brasil, 2003.

MORELLI, E. M. T. **Oracle 9i fundamental**: Sql, Pl/SQL e administração. São Paulo: Érica, 2002.

PRICE, J. **Oracle Database 11g Sql**. Porto Alegre, RS: Bookman, 2009.



**Cruzeiro do Sul Virtual**  
Educação a Distância

www.cruzeirodosulvirtual.com.br  
Campus Liberdade  
Rua Galvão Bueno, 868  
CEP 01506-000  
São Paulo - SP - Brasil  
Tel: (55 11) 3385-3000



**Cruzeiro do Sul**  
Educacional