

# Banco de Dados I

Prof. Msc. Aparecido Vilela Junior  
[aparecido.vilela@unicesumar.edu.br](mailto:aparecido.vilela@unicesumar.edu.br)

# **Agregando Dados Usando Funções de Grupo**

# O Que São Funções de Grupo?

As funções de grupo operam em conjuntos de linhas para fornecer um resultado por grupo.

EMP

DEPTNO	SAL
10	2450
10	5000
10	1300
20	800
20	1100
20	3000
20	3000
20	2975
30	1600
30	2850
30	1250
30	950
30	1500
30	1250

"salário  
máximo na  
tabela EMP"

MAX (SAL)

5000

# Tipos de Funções de Grupo

- **AVG**
- **COUNT**
- **MAX**
- **MIN**
- **STDDEV**
- **SUM**
- **VARIANCE**

# Tipos de Funções de Grupo

- Cada uma das funções aceita um argumento. A tabela a seguir identifica as opções que podem ser usadas na sintaxe.

Função	Descrição
AVG([DISTINCT ALL] <i>n</i> )	Valor médio de <i>n</i> , ignorando valores nulos
COUNT({* [DISTINCT ALL] <i>expr</i> })	Número de linhas, onde <i>expr</i> avalia para algo diferente de nulo (Conte todas as linhas selecionadas usando *, inclusive duplicadas e linhas com nulos.)
MAX([DISTINCT ALL] <i>expr</i> )	Valor máximo de <i>expr</i> , ignorando valores nulos
MIN([DISTINCT ALL] <i>expr</i> )	Valor mínimo de <i>expr</i> , ignorando valores nulos
STDDEV([DISTINCT ALL] <i>x</i> )	Desvio padrão de <i>n</i> , ignorando valores nulos
SUM([DISTINCT ALL] <i>n</i> )	Valores somados de <i>n</i> , ignorando valores nulos
VARIANCE([DISTINCT ALL] <i>x</i> )	Variação de <i>n</i> , ignorando valores nulos

# Usando Funções de Grupo

```
SELECT      [coluna,] group_function(coluna)  
FROM        tabela  
[WHERE      condição]  
[GROUP BY   coluna]  
[ORDER BY   coluna];
```

# Diretrizes para o Uso de Funções de Grupo

- DISTINCT faz com que a função considere somente valores não-duplicados; ALL faz com que ela considere cada valor, inclusive duplicados. O default é ALL e, portanto, não precisa ser especificado.
- Os tipos de dados para os argumentos podem ser CHAR, VARCHAR2, NUMBER ou DATE, onde *expr* está listado.
- Todas as funções de grupo, exceto COUNT(\*), ignoram valores nulos. Para substituir um valor por valores nulos, use a função NVL.
- Implicitamente é realizada a definição do resultado em ordem crescente quando usa uma cláusula GROUP BY. Para sobrepor essa ordenação default, DESC pode ser usado em uma cláusula ORDER BY.

# Usando Funções AVG e SUM

Você pode usar AVG e SUM para dados numéricos.

```
SQL> SELECT  AVG(sal), MAX(sal),  
2           MIN(sal), SUM(sal)  
3 FROM      emp  
4 WHERE     job LIKE 'SALES%';
```

AVG (SAL)	MAX (SAL)	MIN (SAL)	SUM (SAL)
1400	1600	1250	5600

- Você pode usar as funções AVG, SUM, MIN e MAX com colunas que possam armazenar dados numéricos. O exemplo no slide exibe os salários maior, médio, menor e a soma dos salários mensais de todos os vendedores.



# Usando Funções MIN e MAX

Você pode usar MIN e MAX para qualquer tipo de dados.

```
SQL> SELECT MIN(hiredate), MAX(hiredate)
2 FROM emp;
```

MIN(HIRED	MAX(HIRED
-----	-----
17-DEC-80	12-JAN-83

# Usando a Função COUNT

**COUNT(\*)** retorna o número de linhas em uma tabela.

```
SQL> SELECT COUNT(*)  
2 FROM emp  
3 WHERE deptno = 30;
```

```
COUNT(*)
```

```
-----  
6
```

# Usando a Função COUNT

- A Função COUNT tem dois formatos:
- COUNT(\*)
- COUNT(*expr*)
  - COUNT(\*) retorna o número de linhas em uma tabela, inclusive linhas duplicadas e linhas contendo valores nulos em qualquer uma das colunas. Se uma cláusula WHERE estiver incluída na instrução SELECT, COUNT(\*) retornará o número de linhas que satisfizer a condição na cláusula WHERE.
  - Entretanto, COUNT(*expr*) retorna o número de linhas não nulas na coluna identificada por *expr*.

# Usando a Função COUNT

**COUNT(*expr*)** retorna o número de linhas não nulas.

```
SQL> SELECT COUNT(comm)
      2 FROM emp
      3 WHERE deptno = 30;
```

```
COUNT (COMM)
```

```
-----
4
```

As funções de grupo ignoram valores nulos na coluna.

```
SQL> SELECT AVG(comm)
      2 FROM emp;
```

```
AVG (COMM)
```

```
-----
      550
```

- Todas as funções de grupo, com a exceção de COUNT (\*), ignoram valores nulos na coluna.

# Usando a Função NVL com Funções de Grupo

A função NVL força as funções de grupo a incluírem valores nulos.

```
SQL> SELECT AVG (NVL (comm, 0))  
2 FROM emp;
```

```
AVG (NVL (COMM, 0))  
-----  
157.14286
```

# Criando Grupos de Dados

EMP

DEPTNO	SAL			DEPTNO	AVG (SAL)
10	2450			10	2916.6667
10	5000	2916.6667		20	2175
10	1300			30	1566.6667
20	800				
20	1100				
20	3000	2175	"salário médio na tabela EMP para cada departamento"		
20	3000				
20	2975				
30	1600				
30	2850				
30	1250	1566.6667			
30	950				
30	1500				
30	1250				

# Criando Grupos de Dados:

## Cláusula GROUP BY

```
SELECT      coluna, group_function(coluna)
FROM        tabela
[WHERE      condição]
[GROUP BY   group_by_expression]
[ORDER BY   coluna];
```

Divida linhas de uma tabela em grupos menores usando a cláusula GROUP BY.



# Diretrizes

- Se você incluir uma função de grupo em uma cláusula SELECT, não poderá selecionar resultados individuais, *a menos que* a coluna individual apareça na cláusula GROUP BY.
- Se você não conseguir incluir a lista de colunas, uma mensagem de erro será exibida.
- Ao usar uma cláusula WHERE, você pode excluir linhas com antecedência antes de dividi-las em grupos.
- Você deve incluir as *colunas* na cláusula GROUP BY.
- Não é possível usar o apelido de coluna na cláusula GROUP BY.
- Por default, as linhas são classificadas por ordem crescente das colunas incluídas na lista GROUP BY. Isso pode ser sobreposto usando a cláusula ORDER BY.

Todas as colunas na lista SELECT que não estejam em funções de grupo devem estar na cláusula GROUP BY.

```
SQL> SELECT deptno, AVG(sal)
2 FROM emp
3 GROUP BY deptno;
```

DEPTNO	AVG(SAL)
10	2916.6667
20	2175
30	1566.6667

# Usando a Cláusula GROUP BY em Várias Colunas

```
SQL> SELECT deptno, job, sum(sal)
2 FROM emp
3 GROUP BY deptno, job;
```

DEPTNO	JOB	SUM(SAL)
10	CLERK	1300
10	MANAGER	2450
10	PRESIDENT	5000
20	ANALYST	6000
20	CLERK	1900
...		

9 rows selected.

# Consultas Ilegais

Qualquer coluna ou expressão na lista **SELECT** que não seja uma função agregada deve estar na cláusula **GROUP BY**.

```
SQL> SELECT deptno, COUNT(ename)
      2 FROM emp;
```

*Coluna ausente na cláusula GROUP BY*

```
SELECT deptno, COUNT(ename)
      *
```

```
ERROR at line 1:
```

```
ORA-00937: Nenhuma função de grupo de grupo único
(Not a single-group group function)
```

# Consultas Ilegais

- Não é possível usar a cláusula WHERE para restringir grupos.
- Use a cláusula HAVING para restringir grupos.

```
SQL> SELECT      deptno, AVG(sal)
  2  FROM          emp
  3  WHERE          AVG(sal) > 2000
  4  GROUP BY      deptno;
```

```
WHERE AVG(sal) > 2000
```

```
*
```

```
ERROR at line 3:
```

```
ORA-00934: A função de grupo não é permitida aqui
(Group function is not allowed here)
```

# Cláusula HAVING

Use a cláusula HAVING para restringir grupos

- As linhas são agrupadas.
- A função de grupo é aplicada.
- Os grupos que correspondem à cláusula HAVING são exibidos.

```
SELECT      coluna, group_function
FROM        tabela
[WHERE      condição]
[GROUP BY   group_by_expression]
[HAVING     group_condition]
[ORDER BY   coluna];
```

# Aninhando Funções de Grupo

Exiba o salário médio máximo.

```
SQL> SELECT max(avg(sal))  
2 FROM emp  
3 GROUP BY deptno;
```

```
MAX (AVG (SAL))  
-----  
2916.6667
```

# Sumário

```
SELECT      coluna, group_function(coluna)
FROM        tabela
[WHERE      condição]
[GROUP BY   group_by_expression]
[HAVING     group_condition]
[ORDER BY   coluna];
```

Ordem de avaliação das cláusulas:

- cláusula WHERE
- cláusula GROUP BY
- cláusula HAVING



# Exercícios

- 1) Exiba os salários maior, médio, menor e a soma de todos os salários de todos os funcionários.
  - Coloque um label nas colunas Maximo, Minimo, Soma e Media, respectivamente. Arredonde os resultados para o número inteiro mais próximo.
- 2) Modifique o exercício anterior para exibir o salário maior, médio, menor e a soma de todos os salários para cada tipo de cargo.
- 3) Crie uma consulta para exibir o número de pessoas com o mesmo cargo.

# Exercícios

- 4) Determine o número de gerentes sem listá-los. Coloque um label na coluna Numero de Gerentes.
- 5) Crie uma consulta para exibir a diferença entre os maiores e menores salários. Coloque um label na coluna Diferenca.
- 6) Exiba o número do gerente e o salário do funcionário com menor pagamento sob a supervisão desse gerente. Exclua todos cujo gerente não seja conhecido. Exclua todos os grupos em que o salário mínimo seja menor do que US\$ 1.000. Classifique a saída em ordem decrescente de salário.

# Exercícios

- 7) Crie uma consulta para exibir o nome do departamento, o nome do local, o número de funcionários e o salário médio de todos os funcionários nesse departamento.
  - Coloque um label nas colunas Departamento, Local, Numero de Pessoas e Salario, respectivamente. Arredonde o salário médio para duas casas decimais.
- 8) Crie uma consulta que exiba o número total de funcionários e, desse total, o número total de funcionários contratados em 1980, 1981, 1982 e 1983. Coloque os cabeçalhos apropriados nas colunas.

# Exercícios

- 9) Crie uma consulta matriz para exibir o cargo, o salário desse cargo baseado no número do departamento e o salário total desse cargo para todos os departamentos, colocando em cada coluna um cabeçalho apropriado.

Ex

Job	Dept 10	Dept 20	Dept 30	Total
ANALYST		6000		6000
CLERK	1300	1900	950	4150
MANAGER	2450	2975	2850	8275
PRESIDENT	5000			5000
SALESMAN			5600	5600