

## Bases de Dados

## SQL- Agregação de Dados Usando Funções de Grupo



## O que são Funções de Grupo?

- ▶ As funções de grupo operam em conjuntos de linhas para dar um resultado por grupo.

**EMP**

NDEP	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



## Sintaxe das Funções de Grupo

```
SELECT      [column,] group_function(column)
FROM        table
[WHERE      condition]
[GROUP BY   column]
[ORDER BY   column];
```



## Usar as Funções AVG e SUM

- Pode usar-se AVG e SUM para dados numéricos.

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

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



## Usar as Funções MIN e MAX

- Pode usar-se MIN e MAX para qualquer tipo de dados.

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

MIN(DATA_ENTRADA)	MAX(DATA_ENTRADA)
17-DEC-80	12-JAN-83



## Usar a Função COUNT

- COUNT(\*) devolve o número de linhas de uma tabela.

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

COUNT(*)
6



## Usar a Função COUNT

- ▶ COUNT(expr) devolve o número de linhas não nulas.

```
SQL> SELECT COUNT(premios)
2 FROM emp
3 WHERE ndep = 30;
```

```
COUNT (PREMIOS)
-----
4
```



## Funções de Grupo e Valores Nulos

- ▶ As funções de grupo ignoram os valores nulos na coluna.

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

```
AVG (PREMIOS)
-----
550
```



## Usar a função NVL com funções de Grupo

- ▶ A função NVL força as funções de grupo a incluir os valores nulos.

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

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



## Criar Grupos de Dados

### EMP

NDEP	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

2916.6667

“salário  
médio  
de cada  
departamento  
na tabela  
EMP”

1566.6667

2175

NDEP	AVG(SAL)
10	2916.6667
20	2175
30	1566.6667



## Criar Grupos de Dados: Clausula GROUP BY

- ▶ Divide as linhas de uma tabela em pequenos grupos usando a clausula GROUP BY

```
SELECT      column, group_function(column)
FROM        table
[WHERE      condition]
[GROUP BY   group_by_expression]
[ORDER BY   column];
```



## Usar a Clausula GROUP BY

- ▶ Todas as colunas na lista do SELECT que não estão nas funções de grupo têm de estar na clausula GROUP BY.

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

NDEP	AVG(SAL)
10	2916.6667
20	2175
30	1566.6667



## Usar a clausula GROUP BY

- ▶ A coluna GROUP BY não tem de estar na lista do SELECT.

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

AVG(SAL)

2916.6667
2175
1566.6667



## Group By por Mais de uma Coluna

EMP

NDEP	FUNCAO	SAL
-----		
10	GESTOR	2450
10	PRESIDENTE	5000
10	CONTADOR	1300
20	CONTADOR	800
20	CONTADOR	1100
20	ANALISTA	3000
20	ANALISTA	3000
20	GESTOR	2975
30	VENDEDOR	1600
30	GESTOR	2850
30	VENDEDOR	1250
30	CONTADOR	950
30	VENDEDOR	1500
30	VENDEDOR	1250

“O somatório dos  
salários na tabela  
EMP para cada  
função, agrupado  
por departamento”

NDEP	FUNCAO	SUM(SAL)
-----		
10	CONTADOR	1300
10	GESTOR	2450
10	PRESIDENTE	5000
20	ANALISTA	6000
20	CONTADOR	1900
20	GESTOR	2975
30	CONTADOR	950
30	GESTOR	2850
30	VENDEDOR	5600





## Usar a clausula GROUP BY em várias colunas

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

NDEP	FUNCAO	SUM(SAL)
10	CONTADOR	1300
10	GESTOR	2450
10	PRESIDENTE	5000
20	ANALISTA	6000
20	CONTADOR	1900
...		

9 rows selected.

► 17



## Queries Ilegais usando Funções de Grupo

- Qualquer coluna ou expressão na lista do SELECT que não seja uma função de agregado tem de estar na clausula GROUP BY.

```
SQL> SELECT  ndep, COUNT(nome)
2  FROM      emp;
```

**Falta da coluna na cláusula GROUP BY**

```
SELECT ndep, COUNT(nome)
      *
ERROR at line 1:
ORA-00937: not a single-group group function
```

► 18



## Queries Ilegais usando Funções de Grupo

- ▶ Não pode usar-se a cláusula WHERE para restringir grupos.
- ▶ Usa-se a cláusula HAVING para restringir grupos.

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

```
WHERE AVG(sal) > 2000
*
ERROR at line 3:
ORA-00934: group function is not allowed here
```

Não pode usar-se a cláusula WHERE para restringir grupos

## Excluir Resultados de Grupo

EMP

NDEP	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

5000

3000

2850

"salário máximo  
por departamento  
maior do que  
\$2900"

NDEP	MAX (SAL)
10	5000
20	3000

## Excluir Resultados de Grupo: Cláusula HAVING

- ▶ Usar a cláusula HAVING para restringir grupos
  - ▶ As linhas são agrupadas.
  - ▶ A função de grupo é aplicada.
  - ▶ Os grupos que satisfazem a cláusula HAVING são mostrados.

```
SELECT      column, group_function
FROM        table
[WHERE      condition]
[GROUP BY   group_by_expression]
[HAVING     group_condition]
[ORDER BY   column];
```



## Usar a cláusula HAVING

```
SQL> SELECT  ndep, max(sal)
2  FROM      emp
3  GROUP BY  ndep
4  HAVING    max(sal)>2900;
```

NDEP	MAX(SAL)
10	5000
20	3000



## Usar a Cláusula HAVING

```
SQL> SELECT      funcao, SUM(sal) VENC
  2  FROM          emp
  3  WHERE         funcao NOT LIKE 'VEND%'
  4  GROUP BY     funcao
  5  HAVING        SUM(sal)>5000
  6  ORDER BY     SUM(sal);
```

FUNCAO	VENC
ANALISTA	6000
GESTOR	8275



## Aninhar Funções de Grupo

- Mostrar o salário médio máximo.

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

MAX(AVG(SAL))
2916.6667



## Resumo

```
SELECT      column, group_function(column)
FROM        table
[WHERE      condition]
[GROUP BY   group_by_expression]
[HAVING     group_condition]
[ORDER BY   column];
```

- ▶ Ordem de cálculo das cláusulas:
  - ▶ Clausula WHERE
  - ▶ Clausula GROUP BY
  - ▶ Clausula HAVING

