SESI SENAI

PELO FUTURO DO TRABALHO

06- Seleção de dados com funções de agrupamento

Professor Hermano Roepke

O que são funções de agrupamentos de dados?

As funções de grupo operam em conjuntos de registros (linhas) para fornecer um resultado por grupo. Os conjuntos de registros podem ser tabelas inteira ou uma tabela dividida em grupos menores. Considere o cenário:

EMPREGADO

ID_DEPARTAMENTO	SALARIO
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 empregado"

EMPREGADO

MAX(SALARIO)
5000

Como usar funções de agrupamento de dados

Observe a utilização da função de agrupamento de dados na cláusula SELECT. Dessa forma, temos:

```
SELECT função_de_grupo(coluna)
FROM tabela
[WHERE condição]
[ORDER BY coluna];
```

Importante destacar que as cláusulas WHERE e ORDER BY são utilizadas conforme a necessidade e não são fundamentais para a utilização das funções de agrupamento.

Tipos de Funções de grupo

São 5 os tipos de função de grupo:
AVG (média);
COUNT (contar número de linhas/ocorrências);
MAX (maior valor);
MIN (menor valor);
SUM (soma dos valores);

Exemplos de funções AVG e SUM

Você deve utilizar funções de grupo AVG e SUM somente para dados numéricos, pois o retorno delas sempre será um valor numérico. Veja o exemplo onde é recuperado a média salarial e a soma de todos os salários entre os empregados que são vendedores:

```
SELECT AVG(salario), SUM(salario)
FROM empregado
WHERE funcao = 'VENDEDOR';
```

O retorno será:

AVG(salario)	SUM(salario)
1400.000000	5600.00

Utilizando as funções MIN e MAX

As funções MIN e MAX podem ser utilizadas para qualquer tipo de dado. O exemplo a seguir ilustra a sentença SQL para recuperar a data em que o primeiro empregado foi contratado e a data de contratação do empregado mais recente:

```
SELECT MIN(data_contratacao), MAX(data_contratacao)
FROM empregado;
```

O retorno será:

MIN(data_contratacao)	MAX(data_contratacao)
1980-12-17	1983-01-12

Utilizando as funções MIN e MAX

O próximo exemplo ilustra a sentença SQL para recuperar o menor e o maior salário dentre os empregados que são vendedores:

```
SELECT MIN(salario), MAX(salario)
FROM empregado
WHERE funcao = 'VENDEDOR';
```

O retorno será:

MIN(salario)	MAX(salario)
1250.00	1600.00

Utilizando a função COUNT

A função COUNT(*) retorna o número de linhas (registros) em uma tabela ou conjunto de dados restringidos pela condição da cláusula WHERE. O exemplo abaixo exibe o número de empregados vinculados ao departamento 30:

```
SELECT COUNT(*)
FROM empregado
WHERE id_departamento = 30;
```

O retorno será:

COUNT(*)
6

Utilizando a função COUNT

Já no exemplo abaixo temos a função COUNT recebendo como parâmetro uma coluna. Dessa forma, apenas as linhas que apresentarem um valor para a coluna COMISSAO serão consideradas. Veja:

```
SELECT COUNT(comissao)
FROM empregado
WHERE id_departamento = 30;
```

O retorno será:

COUNT(comissao)
4

Funções de grupo e valores nulos

As funções de grupo ignoram valores nulos na coluna. Por isso, você deve ficar muito atento e, quando necessário, utilizar uma função para converter o valor da coluna em tempo real, a fim de evitar distorções no resultado apurado. Vejamos um exemplo para que você compreenda. A sentença SQL abaixo tem por objetivo exibir a comissão média recebida pelos empregados:

```
SELECT AVG(comissao)
FROM empregado;
```

O retorno será:

AVG(comissao)
550.000000

Funções de grupo e valores nulos

Se você realizar um teste de mesa, vai perceber que a comissão média recebida pelos empregados não é um valor exibido. Isso ocorre porque a função de grupo desprezou as linhas (registros) que não apresentam um valor válido para a coluna COMISSAO. No contexto, a soma das comissões foi de 2200, e esse valor foi dividido por 4, pois esse é o número de empregados que apresentam um valor válido para a coluna que foi somada.

Funções de grupo e valores nulos

Para contornar o problema ilustrado é necessário utilizar uma função que transforme, em tempo real, o valor nulo da coluna que foi passada como parâmetro em um valor válido que não interfira no cálculo (neste caso, zero).

```
SELECT AVG(COALESCE(COMISSAO,0))
FROM empregado;
```

O retorno será:

AVG(COALESCE(COMISSAO,0))
157.142857

Criando grupos de dados

EMPREGADO

ID_DEPARTAMENTO	SALARIO
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 médio na tabela empregado para cada departamento"

ID_DEPARTAMENTO	AVG(SALARIO)
10	2916,667
20	2175
30	1566,6667

A cláusula GROUP BY

Observe a sintaxe:

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

É importante destacar que a coluna que acompanha a função de grupo (colunaX) é opcional, mas, se acrescentada necessariamente deverá ser utilizada a cláusula necessidade e não são fundamentais para a utilização das função de agrupamento.

Utilizando a cláusula GROUP BY

Todas as colunas na lista SELECT que não estejam em funções de grupo devem estar na cláusula GROUP BY. Veja o exemplo abaixo, que corresponde ao comando ilustrado no cenário recente onde se deseja exibir o valor médio dos empregados por departamento de lotação:

```
SELECT id_departamento, AVG(salario)
FROM empregado
GROUP BY id_departamento;
```

O retorno será:

id_departamento	AVG(salario)
10	2916.666667
20	2175.000000
30	1566.666667

Utilizando a cláusula GROUP BY

A coluna GROUP BY não precisa estar na lista SELECT. Mas se estiver na cláusula SELECT junto a uma função de gripo, esta, por sua vez, deve estar na cláusula GROUP BY. Veja um exemplo onde ocorre o agrupamento, mas a coluna que foi utilizada para agrupar não está presente na cláusula SELECT:

```
SELECT AVG(salario)
FROM empregado
GROUP BY id_departamento;
```

O retorno será:

AVG(salario)
2916.666667
2175.000000
1566.666667

Sem informações dos departamentos, os resultados não parecem significativos. Casos como esses são raros, mas podem ser necessários.

Agrupando por mais de uma coluna

EMPREGADO

ID_DEPARTAMENTO	FUNCAO	SALARIO
10	GERENTE	2450
10	PRESIDENTE	5000
10	ESCRITURÁRIO	1300
20	ESCRITURÁRIO	800
20	ESCRITURÁRIO	1100
20	ANALISTA	3000
20	ANALISTA	3000
20	GERENTE	2975
30	VENDEDOR	1600
30	GERENTE	2850
30	VENDEDOR	1250
30	ESCRITURÁRIO	950
30	VENDEDOR	1500
30	VENDEDOR	1250

"soma dos salários na tabela empregado para cada função, agrupados por departamento"

ID_DEPARTAMENTO	FUNCAO	SUM(SALARIO)
10	ESCRITURÁRIO	1300
10	GERENTE	2450
10	PRESIDENTE	5000
20	ANALISTA	6000
20	ESCRITURÁRIO	1900
20	GERENTE	2975
30	ESCRITURÁRIO	950
30	GERENTE	2850
30	VENDEDOR	5600

Utilizando a cláusula GROUP BY em várias colunas

O exemplo a seguir corresponde ao cenário apresentado no slide anterior. Observe a presença das duas colunas na cláusula SELECT e também na cláusula GROUP BY:

```
SELECT id_departamento, funcao, SUM(salario)
FROM empregado
GROUP BY id_departamento, funcao;
```

O retorno será:

id_departamento	funcao	SUM(salario)
20	ESCRITURÁRIO	1900.00
30	VENDEDOR	5600.00
20	GERENTE	2975.00
30	GERENTE	2850.00
10	GERENTE	2450.00
20	ANALISTA	6000.00
10	PRESIDENTE	5000.00
30	ESCRITURÁRIO	950.00
10	ESCRITURÁRIO	1300.00

Consultas ilegais utilizando funções de grupo

Qualquer coluna ou expressão na lista SELECT que não seja uma função de grupo deve estar na cláusula GROUP BY. Veja um exemplo que apresenta uma mensagem de erro:

```
SELECT id_departamento, COUNT(nm_empregado)
FROM empregado;
```

O retorno será:

```
Error Code: 1140. In aggregated query without GROUP BY, expression #1 of SELECT list contains
nonaggregated column 'hermanoempregadoresa.empregado.ID_DEPARTAMENTO'; this is incompatible
with sql_mode=only_full_group_by
```

Consultas ilegais utilizando funções de grupo

Não é possível usar a cláusula WHERE para restringir grupos. Neste caso, você deve utilizar a cláusula HAVING. Mas, antes, veja um exemplo de uma tentativa de utilizar a função de grupo na cláusula WHERE e o respectivo erro:

```
SELECT id_departamento, AVG(salario)
FROM empregado
WHERE AVG(salario) > 2000
GROUP BY id_departamento;
```

O retorno será:

```
Error Code: 1111. Invalid use of group function
```

Não é possível usar a cláusula WHERE para restringir.

Excluindo resultados do grupo

EMPREGADO

ID_DEPARTAMENTO	SALARIO
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 por departamento maior que R$ 2900"

ID_DEPARTAMENTO	MAX(SALARIO)
10	5000
20	3000

A cláusula HAVING

Observe a sintaxe:

```
SELECT [colunaX, ... ] função_de_grupo(coluna)
FROM tabela
[WHERE condição]
[GROUP BY colunaX]
HAVING condição_de_grupo]
[ORDER BY coluna];
```

Use a cláusula HAVING para restringir grupos. Neste caso:

- as linhas são agrupadas;
- a função de grupo é aplicada;
- os grupos que correspondem à cláusula HAVING são exibidos.

Utilizando a cláusula HAVING

O exemplo a seguir corresponde ao cenário apresentado anteriormente. Observe a presença da cláusula HAVING para restringir o grupo a ser exibido:

```
SELECT id_departamento, MAX(salario)
FROM empregado
GROUP BY id_departamento
HAVING MAX(salario) > 2900;
```

O retorno será:

id_departamento	MAX(salario)
10	5000.00
20	3000.00

Exemplo SELECT com todas as cláusulas

Com a cláusula HAVING temos o comando SELECT completo.

```
SELECT funcao, SUM(salario) TOTAL
FROM empregado
WHERE funcao not like "VENDEDOR%"
GROUP BY funcao
HAVING SUM(salario) > 5000
ORDER BY SUM(salario);
```

O retorno será:

funcao	total
ANALISTA	6000.00
GERENTE	8275.00

O exemplo exibe a função e o salário mensal total para cada função, com uma folha de pagamento total excedendo R$ 5.000. o exemplo exclui vendedores e classifica a lista pelo salário mensal total.

Resumo

Funções de agrupamento de dados:

- AVG, COUNT, MAX, MIN e SUM.
- Criar grupos de dados e restringi-los. Com isso, o comando SELECT ganhou mais duas cláusulas, e agora está completo:

```
SELECT          [coluna,...]        função_de_grupo(coluna)
FROM                                tabela
[WHERE          condição]
[GROUP BY       coluxaX]
[HAVING         condição_de_grupo]
[ORDER BY       coluna];
```

SESI SENAI

PELO FUTURO DO TRABALHO

sesisc.org.br sc.senai.br

0800 048 1212

Rodovia Admar Gonzaga, 2765 - Itacorubi - 88034-001 - Florianópolis, SC