

Banco de Dados

José Antônio da Cunha

CEFET – RN

Funções de Agregação

Para totalizar dados, ou seja, somar, contar, calcular média, obter o maior e o menor valores existentes, utilizamos as seguintes funções oferecidas pelo SQL:

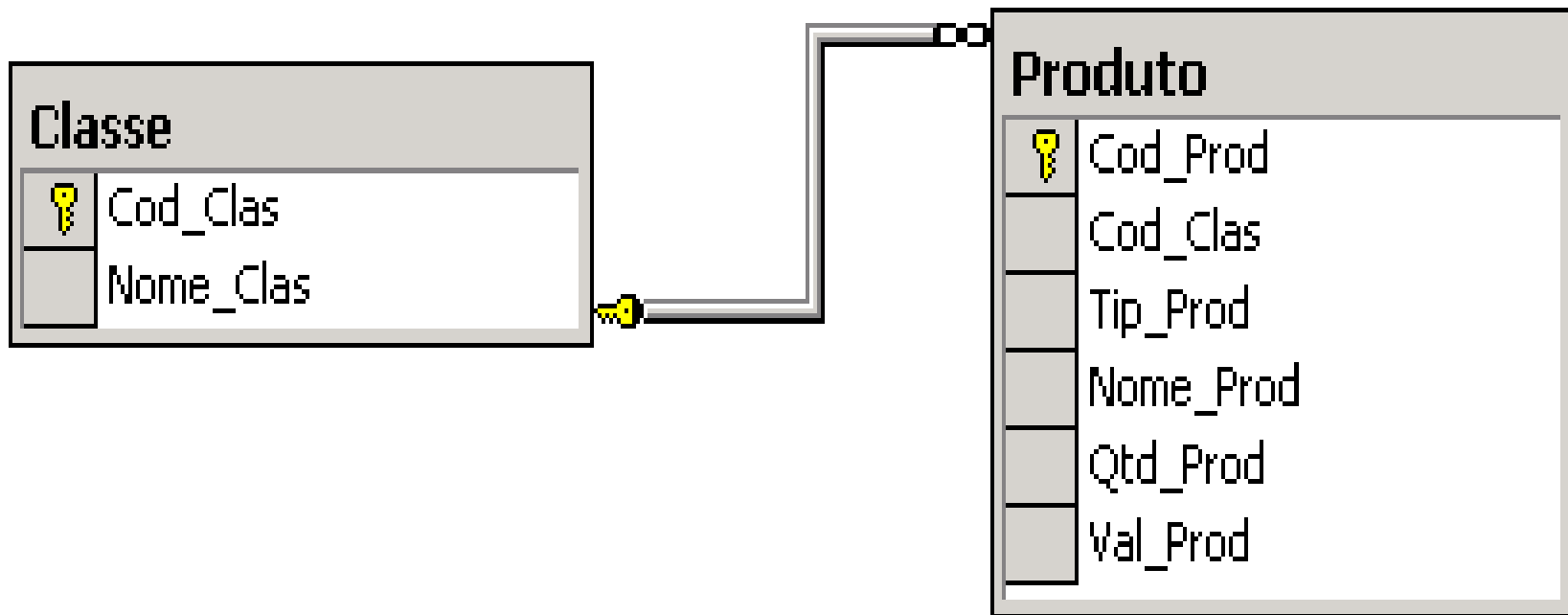
Função	Sintaxe	O que faz
AVG()	AVG([ALL DISTINCT] exp)	Calcula média.
COUNT()	COUNT (ALL DISTINCT)exp)	Conta número de linhas
MAX()	MAX(ALL DISTINCT] exp)	Obtém o maior valor
MIN()	MIN(ALL DISTINCT] exp)	Obtém o menor valor
SUM()	SUM(ALL DISTINCT] exp)	Soma valores
GROUNPING()	GROUNPING(nome-da_coluna)	Mostra o número 1 quando o valor é resultado de uma totalização e zero caso contrário.

Funções de Agregação

Função	Sintaxe	O que faz
STDEV()	STDEV(exp)	Calcula o desvio padrão de todos os valores (estatística).
STDEVP()	STDEVP(exp)	Calcula o desvio padrão de uma população-estatística.
VAR()	VAR(exp)	Calcula a variância – Estatística.
VARP()	VARP(exp)	Calcula a variância de uma população de valores – estatística.

Funções de Agregação

Vamos utilizar as tabelas a seguir para explicarmos estes assuntos.



Funções de Agregação

AVG() – esta função calcula a média de valores de uma determinada coluna de uma tabela.

Exemplo: calcule a média de valores de todos os produtos da tabela produto:

```
SELECT AVG(Val_Prod) FROM Produto
```

Exemplo: calcule a média de quantidade de produtos em estoque dos produtos da classe 1:

```
SELECT AVG(ALL Qtd_Prod) FROM Produto Where Cod_Clas = 1
```

```
SELECT AVG(DISTINCT Qtd_Prod) FROM Produto Where Cod_Clas = 1
```

Funções de Agregação

COUNT() – esta função conta as linhas de uma tabela.

Exemplo: Qual o total de linhas que existe na tabela produto:

```
SELECT COUNT( * ) FROM Produto
```

Exemplo: Quantas linhas existem na tabela produto cuja classe seja 1:

```
SELECT COUNT( * ) FROM Produto Where Cod_Clas = 1
```

```
SELECT COUNT( Cod_Clas ) FROM Produto -- conta o número de linhas  
onde Cod_Clas é diferente de null.
```

Funções de Agregação

MAX() – esta função obtém o maior valor existente em uma coluna de uma determinada coluna de uma tabela.

Exemplo: Mostre qual é o maior valor de produtos da tabela produto:

```
SELECT MAX(Val_Prod) FROM Produto
```

MIN() – esta função obtém o menor valor existente em uma coluna de uma determinada coluna de uma tabela.

Exemplo: Mostre qual é o menor valor de produtos da tabela produto:

```
SELECT MIN(Val_Prod) FROM Produto
```

Funções de Agregação

SUM() – esta função realiza a soma de valores.

Exemplo: Mostre quantidade de produtos existente na tabela produto:

```
SELECT SUM(Qtd_Prod) FROM Produto
```

```
SELECT SUM(DISTINCT Qtd_Prod) FROM Produto
```


Funções de Agregação

GROUP BY – agrupa dados para depois totalizá-los.

Exemplo: Mostre-me o valor total em estoque dos produtos separados por tipo. Neste caso, você teria que utilizar a função de totalização SUM(), fazer uma multiplicação da quantidade de produtos em estoque pelo valor unitário de cada produto e agrupar os produtos por tipo. Veja:

```
SELECT Tipo_Prod,  
       SUM(Qtd_Prod * Val_Prod) AS Val_Estoq  
FROM Produto  
GROUP BY Tipo_Prod
```

Funções de Agregação

GROUP BY – agrupa dados para depois totalizá-los.

Exemplo: Você poderia também querer saber o valor total em estoque dos produtos separados por classe. Veja:

```
SELECT Cod_Clas,  
       SUM(Qtd_Prod * Val_Prod) AS Val_Estoq  
FROM Produto  
GROUP BY Cod_Clas
```

Funções de Agregação

GROUP BY/HAVING – agrupa dados para depois totalizá-los.

Exemplo: Suponha que você quisesse obter a resposta no mesmo formato anterior, mas quisesse ver o total em estoque apenas das linhas em que a totalização dos valores produz um valor maior do que 3 mil reais. Veja:

```
SELECT Cod_Clas,  
       SUM(Qtd_Prod * Val_Prod) AS Val_Estoq  
FROM Produto  
GROUP BY Cod_Clas  
HAVING SUM(Qtd_Prod * Val_Prod) > 3000.00
```

Nota: a cláusula HAVING só pode ser usada com a cláusula Group By

Funções de Agregação

Exibindo e agrupando dados de mais de uma coluna

Exemplo: Você poderia desejar ver o total em estoque dos produtos agrupados por tipo, mas poderia querer ver também o nome dos produtos no resultado exibido. Neste caso, você deve especificar a coluna Nome_Prod na lista de colunas do comando select, e deve também agrupar os valores por essa coluna. Veja:

```
SELECT Tip_Prod, Nome_Prod,  
       SUM(Qtd_Prod * Val_Prod) AS Estoque  
FROM Produto  
GROUP BY Tip_Prod, Nome_Prod  
Order By Tip_Prod
```

Funções de Agregação

Exibindo e agrupando dados de mais de uma coluna

Nota: como regra geral, assuma que se você quiser exibir uma coluna com dados agrupados, na cláusula group by você deve especificar essa coluna. Se você quiser mostrar duas colunas, na cláusula group by deve especificar essas colunas. E assim por diante.

Funções de Agregação

GROUP BY/WITH ROLLUP – exibir um subtotal e um total geral no final.

Exemplo: No exemplo anterior, seria interessante, se fosse exibido um subtotal após cada tipo de produto e no final um total geral. Veja:

```
SELECT Tip_Prod, Nome_Prod,  
       SUM(Qtd_Prod * Val_Prod) AS Estoque  
FROM Produto  
GROUP BY Tip_Prod, Nome_Prod  
WITH ROLLUP
```

Funções de Agregação

GROUP BY/WITH CUBE – Para cada coluna na cláusula group by, exibir um novo subtotal.

Exemplo: Veja a saída quando você utiliza a cláusula WITH CUBE.

```
SELECT Tip_Prod, Nome_Prod,  
       SUM(Qtd_Prod * Val_Prod) AS Estoque  
FROM Produto  
GROUP BY Tip_Prod, Nome_Prod  
WITH CUBE
```

Funções de Agregação

GROUP BY e a Função GROUPING – a função GROUPING é utilizada para facilitar a visualização dos resultados apresentados pelo comando Select que tiver as cláusulas Group By With Rollup ou With Cube. Essa função mostra o número 1 quando o valor da totalização apresentado trata-se de um total ou subtotal produzidos pelas cláusulas With Cube ou With Rollup.

Exemplo:

```
SELECT Tip_Prod,  
       GROUPING(Tip_Prod) as [É Total Agrupado ?],  
       SUM(Qtd_Prod ) AS Val_Estoque  
FROM Produto  
GROUP BY Tip_Prod  
WITH ROLLUP
```


Funções de Agregação

Resultados Relacionais – os valores obtidos com a totalização utilizando a cláusula group by são chamados de valores relacionais, porque é possível carregá-los em variáveis de memória e trabalhar com eles.

Funções de Agregação

Resultados não Relacionais – os valores obtidos com as totalização utilizando a cláusula **Compute** e **Compute By** são chamados de valores não relacionais, porque não é possível carregá-los em variáveis de memória e trabalhar com eles. Em fim, os dados podem apenas ser lidos.

```
SELECT Tip_Prod, Nome_Prod, Qtd_Prod, Total = Qtd_Prod * Val_Prod
FROM Produto
ORDER BY Tip_Prod, Nome_Prod
COMPUTE SUM(Qtd_Prod * Val_Prod) BY Tip_Prod
COMPUTE SUM(Qtd_Prod)
COMPUTE SUM(Qtd_Prod * Val_Prod)
```

Funções de Agregação

A cláusula “Compute By” gera subtotais e a cláusula “Compute” apenas gera total geral. Você pode utilizar as cláusulas coumpute e compute by juntas ou separadamente.

```
SELECT Tip_Prod,  
       Nome_Prod,  
       Qtd_Prod,  
       Total = Qtd_Prod * Val_Prod  
FROM Produto  
ORDER BY Tip_Prod, Nome_Prod  
COMPUTE SUM(Qtd_Prod * Val_Prod) BY Tip_Prod
```

Funções de Agregação

A cláusula “Compute By” gera subtotais e a cláusula “Compute” apenas gera total geral. Você pode utilizar as cláusulas coumpute e compute by juntas ou separadamente.

```
SELECT Tip_Prod,  
       Nome_Prod,  
       Qtd_Prod,  
       Total = Qtd_Prod * Val_Prod  
FROM Produto  
ORDER BY Tip_Prod, Nome_Prod  
COMPUTE SUM(Qtd_Prod * Val_Prod)
```

Fim