

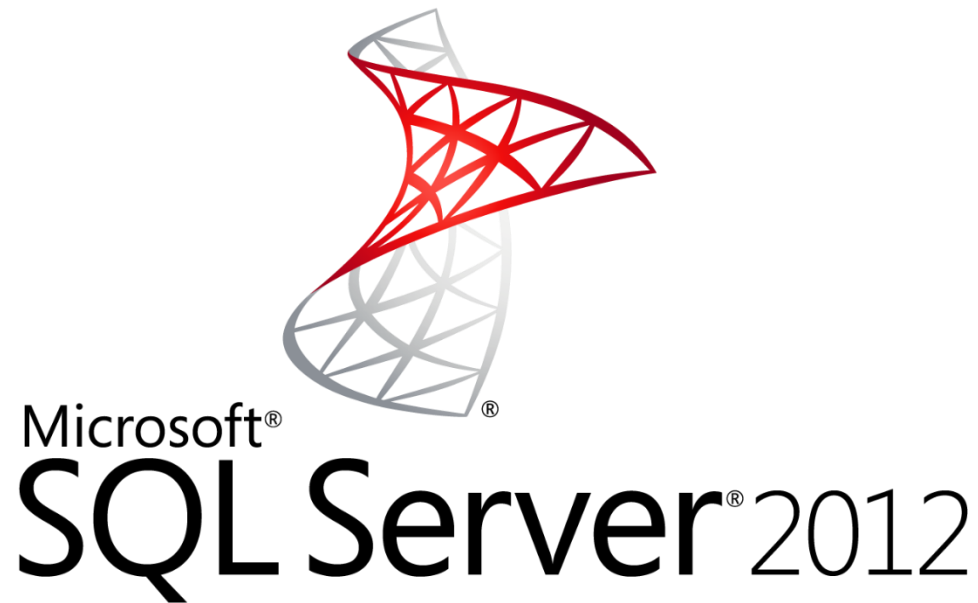


CWI SOFTWARE

módulo 2 | banco de dados

Crescer 2015-1

Feevale



3 - Comandos SQL – Funções de Grupo

André Luís Nunes

abril/2015

Tipos de comandos

→ **DDL:** Data Definition Language

→ Comandos que permitem definir a estrutura de objetos.

→ **DML:** Data Manipulation Language

→ Comandos que permitem alterar os dados de uma tabela.

→ Inserir, alterar, excluir e consultar registros.

→ **DCL:** Data Control Language

→ Comandos que permitem conceder ou revogar permissões à determinados objetos de uma base dados.

SQL -> Funções de grupo

Funções de agrupamento (aggregation functions):

FUNÇÃO	DESCRIÇÃO
COUNT	Contabiliza o total de registros (não considera valores nulos).
MIN	Busca o menor valor.
MAX	Busca o maior valor.
SUM	Soma todas as linhas.
AVG	Cálcula a média sobre todos os valores.

- » Funções que exercem cálculos sobre vários registros, agrupando-os.
- » Exemplo: selecionando o número de registros da tabela Empregado:

```
Select COUNT(IDEmpregado) From Empregado;
```

```
Select COUNT(*) From Empregado;
```

SQL -> Funções de grupo



IDEmpregado	NomeEmpregado	Cargo	IDGerente	DataAdmissao	Salario	Comissao	IDDepartamento
7369	SMITH	Atendente	7902	17-Dec-1980	800	NULL	20
7499	ALLEN	Vendedor	7698	20-Feb-1981	1600	300	30
7521	WARD	Vendedor	7698	22-Feb-1981	1250	500	30
7566	JONES	Gerente	7839	2-Apr-1981	2975	NULL	20
7654	MARTIN	Vendedor	7698	28-Sep-1981	1250	1400	30
7698	BLAKE	Gerente	7839	1-May-1981	2850	NULL	30
7782	CLARK	Gerente	7839	9-Jun-1981	2450	NULL	10
7788	SCOTT	Analista	7566	9-Dec-1982	3000	NULL	20
7839	KING	Presidente	NULL	17-Nov-1981	5000	NULL	NULL
7844	TURNER	Vendedor	7698	8-Sep-1981	1500	0	30
7876	ADAMS	Atendente	7788	12-Jan-1983	1100	NULL	20
7900	JAMES	Atendente	7698	3-Dec-1981	950	NULL	30
7902	FORD	Analista	7566	3-Dec-1981	3000	NULL	20
7934	MILLER	Atendente	7782	23-Jan-1982	1300	NULL	10
7940	ANDREW	Atendente	7782	20-Jan-1988	1150	NULL	NULL

```
Select COUNT(IDDepartamento)
From Empregado;
```

Contará 1 para cada registro com valor não-nulo.

SQL -> Funções de grupo

Exemplos de funções de grupo:

» Selecionando o menor e o maior salário:

```
Select MIN(salario) as MenorSalario,  
        MAX(salario) as MaiorSalario  
From Empregado;
```

» Selecionando o número de empregados e o número de cargos distintos:

```
Select COUNT(1) as TotalEmpregados,  
        COUNT(Distinct Cargo) as TotalCargos  
From Empregado;
```

SQL -> Funções de grupo (GROUP BY)

Funções de grupo com GROUP BY

Incluindo “quebras” por colunas juntamente com funções de grupo.

» Sintaxe do GROUP BY:

```
Select <ColumnName>,  
        aggregate_function(<ColumnName>)  
  From <TableName>  
 Group By <ColumnName>;
```

Agrupando por colunas: quando se deseja exibir o total por alguma quebra.
Por exemplo, o total de empregados de cada departamento.

SQL -> Funções de grupo (GROUP BY)

Exemplos:

» Retornando o número de empregados por cargo:

```
Select Cargo,  
        COUNT(1) as Total  
  From Empregado  
 Group By Cargo;
```

» Retornando o número de empregados por cargo e gerente:

```
Select Cargo,  
        IDGerente,  
        COUNT(1) as Total  
  From Empregado  
 Group By Cargo,  
        IDGerente;
```


SQL -> Funções de grupo (GROUP BY)

Compreendendo o mecanismo de agrupamento

» Retornando o número de empregados por cargo:

IDEmpregado	NomeEmpregado	Cargo	IDGerente	DataAdmissao	Salario	Comissao	IDDepartamento	COUNT(1)
7788	SCOTT	Analista	7566	9-Dec-1982	3000	NULL	20	2
7902	FORD	Analista	7566	3-Dec-1981	3000	NULL	20	
7934	MILLER	Atendente					10	5
7940	ANDREW	Atendente				NULL	20	
7876	ADAMS	Atendente					30	
7900	JAMES	Atendente					20	
7369	SMITH	Atendente					20	
7566	JONES	Gerente	7839	2-Apr-1981	2975	NULL	20	3
7698	BLAKE	Gerente	7839	1-May-1981	2850	NULL	30	
7782	CLARK	Gerente	7839	9-Jun-1981	2450	NULL	10	
7839	KING	Presidente	NULL	17-Nov-1981	5000	NULL	NULL	1
7844	TURNER	Vendedor	7698	8-Sep-1981	1500	0	30	4
7654	MARTIN	Vendedor	7698	28-Sep-1981	1250	1400	30	
7499	ALLEN	Vendedor	7698	20-Feb-1981	1600	300	30	
7521	WARD	Vendedor	7698	22-Feb-1981	1250	500	30	

```
Select Cargo,  
COUNT(1) as Total  
From Empregado  
Group By Cargo;
```

SQL -> Funções de grupo (HAVING)

Restringindo resultados das funções de grupo.

» Sintaxe do HAVING:

```
Select <ColumnName>,  
        aggregate_function(<ColumnName>  
    From <TableName>  
    Group By <ColumnName>  
    Having aggregate_function(<ColumnName>) <Operator> <value>;
```

Diferenças:

Where: impor condições sobre os registros que serão selecionados, não permite que seja adicionado uma função de grupo (SUM, MAX, MIN, COUNT, AVG) nesta cláusula.

Having: impor condições sobre o resultado de uma função de agrupamento. Evite utilizar condições que podem ser utilizadas no where.

SQL -> Funções de grupo (HAVING)

» Exemplo: retornando o número de empregados por cargo, restringindo os cargos com mais de um empregado:

```
Select Cargo,  
        COUNT(1) as Total  
  From Empregado  
 Group By Cargo  
 Having COUNT(1) > 1;
```

» Exemplo que não funciona:

```
Select Cargo,  
        COUNT(1) as Total  
  From Empregado  
 Where Count(1) > 1  
 Group By Cargo;
```

» Exemplo que funciona, mas **não deve** ser utilizado:

```
Select Cargo,  
        COUNT(1) as Total  
  From Empregado  
 Group By Cargo  
 Having Cargo LIKE 'V%';
```

SQL -> Funções SQL

Funções (SQL Server) SQL: são comuns em várias linguagens, podem ser utilizadas diretamente na SQL e também na T-SQL (Transaction SQL).

» Exemplos do uso das funções maiúsculo e minúsculo:

```
Select UPPER(Nome) as NomeCidade_Upper,  
        LOWER(Nome) as NomeCidade_Lower,  
        UF  
  From CIDADE  
 Order by NomeCidade_Upper;
```

» Exemplos do uso da função SUBSTRING:

```
Select SUBSTRING(Nome, 1, 3) as NomeCidade,  
        UF  
  From CIDADE  
 Order by NomeCidade;
```

SQL -> Funções SQL - exemplos

Alguns exemplos do uso de funções em comandos SQL:

```
Select NomeEmpregado,  
       (Salario + Comissao) as Total_sal_a,  
       (Salario + (ISNULL(Comissao,0))) as Total_sal_b,  
       Comissao  
From Empregado;
```

» Observem o resultado desta consulta, como a comissão é opcional, e alguns registros não possuem valor no momento do cálculo pode ocorrer “distorções”, no caso, não exibir valor algum.

SQL -> Funções SQL - exemplos

Calculando datas, pode ser utilizada qualquer parte da data:

```
Select NomeEmpregado,  
       DataAdmissao,  
       DateDiff(Month, DataAdmissao, getdate()) MesesTrabalho  
From Empregado;
```

» É permitido utilizar uma função sobre o resultado de outra função.

```
Select Nome,  
       REPLACE(REPLACE(Nome, 'ção', 'cion'), 'são', 'san') NomeCidade,  
       UF  
From Cidade;
```

SQL -> Funções SQL - CASE

Avalia uma lista de condições e retorna uma das várias expressões possíveis.

```
select IDAssociado,  
       Nome,  
       Case when sexo = 'F' then 'Feminino'  
            when sexo = 'M' then 'Masculino'  
            else 'sei lá'  
       End Genero,  
       decode( sexo, 'F', 'Feminino',  
              'M', 'Masculino',  
              'sei lá') Genero2  
from associado;
```

SQL - Exercícios



- 1) Faça uma consulta (query) que retorne apenas o primeiro nome do Associado (tabela associado).
- 2) Faça uma consulta que retorne o nome dos associados e a idade de cada um (em anos, exibir um número inteiro).
- 3) Faça uma consulta que liste os empregados admitidos entre 01/05/1980 e 20/01/1982. Exibir também o total de meses de trabalho até a data de 31/12/2000.
- 4) Qual o cargo (tabela empregado) possuir mais empregados?
- 5) Qual é o associado de maior nome (número de caracteres) ?

SQL - Exercícios



- 6) Faça uma consulta que retorne o nome do associado e a data de quando cada completará (ou completou) 50 anos, liste também o dia da semana.
- 7) Liste a quantidade de cidades agrupando por UF.
- 8) Liste as cidades que possuem o mesmo nome e UF.
- 9) Identifique qual deve ser o próximo ID para a criação de um novo registro na tabela Associado (maior + 1).
- 10) Limpe a tabela CidadeAux, e insira somente as cidades com nomes e UF's distintos, considere somente o menor código ID das cidades duplicadas.

SQL - Exercícios



CursoSQL



exercícios



- 11) Altere todas cidades duplicadas (nome e uf iguais), acrescente no início do nome um asterisco (*).
- 12) Faça uma consulta que liste o nome do Associado e a descrição da coluna Sexo, informando: Masculino ou Feminino.
- 13) Faça uma consulta que mostre o nome do empregado, o Salario e percentual a ser descontado do Imposto de Renda, considerando a tabela abaixo:

Até R\$ 1.164,00 = 0%	De R\$ 1.164,00 a R\$ 2.326,00 = 15%
Acima de R\$ 2.326,00 = 27,5%.	
- 14) Elimine as cidades duplicadas (mantendo 1 registro para cada).
- 15) Adicione uma regra que impeça exista mais de uma cidade com o mesmo nome em um estado.

André Nunes

Núcleo de Tecnologia

andrenunes@cwí.com.br

(51) 3081.3622