

Administrador de Banco de Dados (extra Abril) - Turma 2024A

4.8 Comandos de seleção de dados da tabela: SELECT

Ao analisar a sintaxe dos comandos SQL, considere a expressão entre colchetes como opcional e expressões com “|” (pipe ou barra vertical) como mutuamente exclusivas (uma ou outra).

Select – é utilizada para fazer uma consulta no banco de dados. Ao submeter um comando DML, o SGBD analisa a sintaxe, define a melhor estratégia de recuperação dos dados, executa o comando e retorna uma resposta (result set). Como resultado produz uma relação, que nada mais é que uma nova tabela com o resultado da consulta. A sintaxe básica do comando select é apresentada a seguir.

```
Select [ All | Distinct ] coluna1 [, coluna2]  
From tabela1  
[Where <condições de seleção> ]  
[Order by coluna1,coluna2 [asc | desc] ]
```

Lista de colunas – após o comando **select**, corresponde a quais serão os atributos mostrados como resposta. A cláusula **from** mostra a lista das tabelas envolvidas na seleção. A cláusula **where** corresponde ao predicado com as restrições. A cláusula **order by** é um recurso adicional da SQL que faz com que os dados, após serem selecionados, possam ser ordenados de acordo com a(s) coluna(s) informada(s), para em seguida serem apresentados como resultado final. O comando select possui uma enorme quantidade de opções de sintaxe que o torna extremamente poderoso e rico em termos de sua capacidade em produzir resultados. Neste texto vamos apresentar algumas das opções de sintaxe sem a pretensão de esgotar o tema.

All e distinct – são comandos opcionais, sendo que o **all** é o default. O **distinct** restringe no resultado gerado quaisquer linhas repetidas. Quando o **all** é declarado ou nada é declarado, não haverá restrições em apresentar linhas repetidas na saída.

As – é um operador de renomeação do atributo. Altera o nome do atributo gerado na saída. Cada uma das colunas separadas por vírgulas a serem retornadas podem ser renomeadas. Basta inserir após a descrição da coluna o comando as <nome_renomeado>, ou apenas <nome_renomeado>, tendo em vista que a cláusula as é opcional. Obs.: o nome utilizado na renomeação do atributo não pode conter espaços em branco. Se desejar fazê-lo, a expressão deverá ser informada entre aspas. Abaixo temos dois exemplos.

```
Selece Nro_Nf as NotaFiscal, Vlr_Pago as "Valor Pago" from NotaFiscal;
```

```
Select Nro_Nf Nota, Vlr_Pago Total from NotaFiscal;
```

Dentro da lista de colunas que podem ser retornadas no comando select, é possível, além dos atributos das tabelas envolvidas na cláusula from, definir constantes, que nada mais são que qualquer conteúdo entre aspas, resultado de operações matemáticas, etc. A lista de atributos pode ser utilizada como operando pelos operadores matemáticos e como argumentos de funções de agregação, comparação, etc. É o que veremos a seguir.

Operadores aritméticos

Os operadores aritméticos possibilitam realizar expressões aritméticas com os atributos que estão armazenados, retornando o resultado da expressão executada.

- + Adiciona dois valores numéricos.
- - Subtrai dois valores numéricos.
- * Multiplica dois valores numéricos.
- / Divide dois valores numéricos. Divisão por zero produz resultado nulo.
- % Divide dois valores numéricos. Divisão por zero produz resultado nulo.
- () Altera a precedência da execução da operação aritmética.

Pode-se usar a cláusula as para atribuir um nome ao resultado de uma expressão:

```
Select Nro_NF, Nro_Item, (Qtd_Comprada * Vlr_Pago) as Subtotal  
from ItemNotaFiscal;
```

Funções de agregação

As funções agregadas podem ser utilizadas de forma exclusiva na lista de atributos. Nesse caso retornam apenas um única linha como resultado final. Ou, usadas combinadas com atributos comuns originados das tabelas existentes no banco de dados:

- **AVG([distinct]expr)** Retorna a média da expr. Considera apenas valores não nulos. O uso do distinct faz com que sejam considerados para cálculo da média apenas os valores distintos.
- **COUNT(expr)** Conta o número de linhas, valores não nulos, retornados pelo comando select.
- **COUNT(*)** Conta o número de valores não nulos retornados pelo comando select.
- **COUNT(distinct expr)** Conta o número de valores não nulos das linhas retornadas que são distintos.
- **MAX(expr)** Retorna o máximo valor não nulo encontrado na expr.
- **MIN(expr)** Retorna o mínimo valor não nulo encontrado na expr.

No caso de a agregação ser adicionada com outros atributos na lista de seleção do comando SQL, o resultado retornado será uma linha para cada conjunto distinto dos atributos da lista, acrescida do resultado retornado pela função agregada aplicada a esse mesmo subconjunto de dados, o qual deverá estar definido de forma explícita na cláusula group by do comando select. Abaixo temos um exemplo:

```
Select Nro_Nf, sum(Qtd_Comprada * Vlr_Pago) as Total
from ItemNotaFiscal
group by Nro_Nf;
```

Obs.: Alguns SGBDs aceitam uma sintaxe em que a lista de atributos do group by, pode ser informada a partir de números que representam a posição que o(s) campo(s) que está(ão) sendo agrupado(s) se encontra(m) da lista dos atributos que estão sendo selecionados. Veja o exemplo:

```
Select Nro_Nf, sum(Qtd_Comprada * Vlr_Pago) as Total
from ItemNotaFiscal
group by 1
order by 2 desc;
```

Nesse caso o resultado será ordenado por valor total de cada nota de forma decrescente.

Funções numéricas

As funções numéricas podem utilizar os atributos como argumentos, retornando o resultado dessas operações. A Figura 5.8 apresenta um lista com algumas das funções numéricas.

- **ABS(x)** Retorna o valor absoluto de x.
- **POW(x, y)** Retorna o resultado de x elevado a y.
- **RAND([n])** Retorna um valor aleatório de ponto flutuante no intervalo de 0.0 a 1.0. O n é o valor da semente para geração do número. Obs.: o resultado retornado sempre será o mesmo para um dado valor de n.
- **ROUND(x, d)** Retorna o valor de x, arredondado para d casas decimais. Sendo para d = 0, o valor retornado é um número inteiro.
- **SQRT(x)** Retorna a raiz quadrada de x.
- **TRUNCATE(x, d)** Retorna o valor de x truncado para d casas decimais.

A seguir é apresentado um exemplo do uso da função Round():

```
Select Nro_NF, Nro_Item, Round(Qtd_Comprada * Vlr_Pago),0) Subtotal  
  
from ItemNotaFiscal;
```

Funções de string

As funções de string operam nos atributos do tipo string, possibilitando realizar nesses atributos algumas alterações visando adequar o resultado que se deseja obter. Abaixo temos um lista com algumas das funções numéricas:

- **ASCII(string)** Retorna o código ASCII do caracter mais à esquerda da string.
- **CHAR(n1, n2, ...)** Retorna os caracteres correspondentes ao código ASCII.
- **CHARSET** Retorna o conjunto de caracteres da string fornecida.
- **COALESCE(expr1,expr2, ...)** Retorna o primeiro elemento não nulo da lista.
- **LCASE (string)** Retorna a string com os caracteres convertidos para maiúsculas.
- **LENGTH(string)** Retorna o número de caracteres da string.
- **LTRIM(string)** Retorna a string sem os caracteres brancos iniciais.
- **RTRIM(string)** Retorna a string sem os caracteres brancos finais.
- **SUBSTRING(string, pos,tam)** Retorna uma parte da string, a partir da posição pos com um tamanho tam.
- **UCASE(string)** Retorna a string com os caracteres convertidos para maiúsculas.

A seguir é apresentado um exemplo do uso da função lcase:

```
Select lcase(Nom_Cliente) as Nomes  
from Cliente;
```

Funções de data

As funções de data operam nos atributos do tipo date, datetime, time, possibilitando realizar algumas alterações nesses atributos visando adequar o resultado de acordo com a necessidade. Abaixo temos um lista com algumas das funções de data:

- **DAY(data)** Retorna o dia do mês de 1 até 31 de uma data.
- **DAYOFWEEK(data)** Retorna um valor numérico de 1 até 7, correspondendo ao dia da semana de domingo até sábado, respectivamente.
- **DAYOFYEAR(data)** Retorna o número de dias corridos no ano de uma data. Pode variar de 1 a 366.
- **MONTH(data)** Retorna o mês de 1 até 12 de uma data.
- **NOW()** Retorna data e hora correntes em uma string no formato “AAAA-MM-DD HH:MM:SS”.
- **WEEKDAY(data)** Retorna um número 0 até 6 correspondendo ao dia da semana de segunda até domingo respectivamente.
- **WEEKOFYEAR(data)** Retorna quantas semanas houve, de acordo com o calendário, desde o início do ano até a data passada como argumento. O número de semanas retornado pode variar de 1 até 53.
- **YEAR(data)** Retorna o ano de uma data com quatro algarismos.

A seguir é apresentado um exemplo do uso da função DayOfWeek:

```
Select DayOfWeek(Dat_Emissao) as Dia_da_Semana,  
count(*) as Total_Notas_Emitidas  
from NotaFiscal  
group by 1
```

Funções de comparação

As funções de comparação operam nos atributos possibilitando realizar algumas comparações entre eles antes de retornar o resultado. Abaixo temos uma lista com algumas dessas funções:

- **GREATEST(expr1,expr2)** Retorna a maior das duas expressões.
- **IF(expr1,expr2,expr3)** Se expr1 for verdadeira retorna expr2. Se expr1 for falsa ou nula retorna expr3.
- **IFNULL(expr1, expr2)** Retorna expr2 se expr1 for nula, senão retorna a expr1.
- **ISNULL(expr)** Retorna 1 (verdadeiro) se a expr for nula, senão retorna 0 (falso). Isto é necessário, pois comparação de valores nulos usando o símbolo = retorna falso.
- **LEAST(expr1,expr2)** Retorna a menor das duas expressões.

Abaixo temos um exemplo do uso de função de comparação.

```
Select Cod_Produto,  
Least(Vlr_Venda, Vlr_Promocional) as Valor_Venda  
from Produto
```

Comando case

A comando case é utilizado quando queremos testar o conteúdo de um atributo armazenado no banco de dados e retornar como resposta um outro valor. Veja um exemplo da sintaxe utilizada:

```
CASE expr  
WHEN valor1 THEN Resultado  
[WHEN valor2 THEN Resultado]  
[ELSE Resultado]  
END
```

Compara expr com um conjunto de possibilidades de valores. Onde a comparação for verdadeira, retorna o resultado.

- **WHEN** – Especifica o valor a ser comparado com expr.
- **THEN** – Especifica o resultado a ser retornado.
- **ELSE** – É selecionado como opção do resultado, caso todas as comparações anteriores forem falsas.

Vejamos um exemplo que retorna a quantidade de Notas Fiscais emitidas a cada dia da semana, apresentado a seguir:

```
Select Case DayOfWeek(Dat_Emissao)
when 0 then "Domingo"
when 1 then "Segunda"
when 2 then "Terca"
when 3 then "Quarta"
when 4 then "Quinta"
when 5 then "Sexta"
when 0 then "Sabado"
End as Dia_da_Semana,
count(*) as Total_Notas_Emitidas
from NotaFiscal
group by 1
```

Este material foi baseado em:

da SILVA, Edson Marchetti. **Banco de Dados I**. Minas Gerais: Centro Federal de Educação Tecnológica de Minas Gerais/Rede e-Tec, 2012.

Última atualização: sexta, 7 jul 2023, 10:41

◀ 4.7 Comandos de manipulação de dados da tabela

Seguir para...

4.9 Comandos de seleção de dados da tabela: FROM ►