TLBD II

Continuação (03)

Funções do SQL

O que são funções?

São pedaços de código, definidos pelo utillizador ou pré-definidos pela linguagem, utilizados para manipular dados. Aceitam um ou mais argumentos, devolvendo um valor. O argumento é uma constante, variável ou o nome de uma coluna. O valor devolvido serve para classificar a função: será numérica se o valor devolvido é numérico, char se devolve um char.

As funções são utilizadas para manipular dados, tornando mais potentes as consultas. Dividem-se em três grandes grupos: manipulação de **linhas**, manipulação de **grupos** de linhas e funções analíticas. Dentro do primeiro grupo há funções para manipular **caracteres**, **números**, **datas** e funções que permitem **converter** dados de um tipo para outro. As **funções de grupo** permitem obter um valor que depende do grupo de linhas, por exemplo uma média, variância ou um máximo. As funções analíticas misturam os dois tipos anteriores: devolvem um valor por cada linha, mas esse valor depende do grupo.

Atuam sobre cada uma das linhas resultantes da consulta. Para cada linha produzem um valor que depende dos argumentos recebidos. Podem ser encadeadas com outras funções.

Funções de linha

Funções de manipulação de caracter

Estas funções aceitam como argumento(s) caracteres e devolvem caracteres ou valores numéricos:

- LOWER( )
- UPPER( )
- CONCAT( )
- LPAD( )
- RPAD( )
- SUBSTR( )
- INSTR( )
- LTRIM( )
- RTRIM( )
- LENGTH( )
- REPLACE( )
- DECODE( ) - Funções de conversão

LOWER() e UPPER()

LOWER(string) converte 'string' para letras minúsculas.

Exemplo:
select LOWER(dnome), LOWER('CURSO') from dept;

UPPER(string) converte 'string' para letras maiúsculas.

Exemplo:
select UPPER('tlbd'), UPPER(dnome) from dept;

CONCAT()

CONCAT(string1, string2) devolve a concatenação de string1 com string2.

Exemplo:

```
SELECT CONCAT(enome, emprego),
CONCAT(empno,enome) from emp;
```

LPAD() e RPAD()

LPAD(string1, n, string2)
São reservados n caracteres para output. A string1 é encostada à direita, sendo o espaço à esquerda preenchido com string2.

Exemplo:
SELECT LPAD(enome, 10, "*"), from emp;

RPAD(string1, n, string2)
São reservados n caracteres para output. A string1 é encostada à esquerda, sendo o espaço à direita preenchido com string2.

Exemplo:
SELECT RPAD(enome, 10, "*"), from emp;

SUBSTR()

SUBSTR(string1,pos,n) pressupõe que os caracteres numa cadeia são numerados da esquerda para a direita, começando em 1. Devolve a substring de 'string1' que começa na posição 'pos' e tem comprimento n. Se o parametro 'n' for omitido, devolve todos os caracteres desde a posição 'pos' até ao fim.

Exemplo:

```
SELECT SUBSTR('MYSQL',2,3),
SUBSTR(dnome,2), substr(dnome,3,5)
FROM dept;
```

INSTR()

INSTR(string1,string2) Devolve a posição da primeira ocorrência de string2 dentro de string1.

Exemplo:

```
SELECT INSTR(dnome,'A'), dnome
FROM dept;
```

LTRIM() e RTRIM()

LTRIM(string)
Retira todas as ocorrências de espaço dentro de string que estejam encostadas à **esquerda**.

RTRIM(string)
Retira todas as ocorrências de espaço dentro de string que estejam encostadas à **direita**.

Exemplo:

```
SELECT LTRIM(dnome), RTRIM(dnome), dnome
FROM emp;
```

OBS: O RTRIM é particularmente útil para retirar espaços indesejados à direita de uma string. Ao introduzir dados para uma coluna (por exemplo ENOME da tabela EMP) podem ser introduzidos espaços em branco à direita, que além de ocupar espaço, podem dificultar as pesquisas. Com RTRIM é possível retirar esses espaços, por exemplo: UPDATE emp SET enome=RTRIM(enome);

LENGTH()

LENGTH(string1) Devolve o comprimento de string1.

Exemplo:

```
SELECT LENGTH(dnome), dnome
FROM dept;
```

REPLACE()

REPLACE(string1,string2,string3)
Trabalhando sobre a string1, converte todas as ocorrências de string2 para string3.

Exemplo:
SELECT enome, REPLACE(enome,'A','O')
FROM emp

Funções encadeadas

As funções de linha podem ser progressivamente encadeadas, umas dentro das outras. A avaliação é feita das funções interiores para as funções exteriores.

Como faria para saber quantas vezes aparece um determinado caracter dentro de uma string? Uma solução é encadear as funções LENGTH e REPLACE. No exemplo abaixo o caracter S é substituído por nada (é retirado). A diferença de comprimentos dá o número de S's!

Exemplo:

```
SELECT dnome, LENGTH(dnome),
LENGTH(dnome)-LENGTH(REPLACE(dnome,'S',''))
FROM dept;
```

Funções numéricas

- ROUND( )
- CEIL( )
- FLOOR( )
- POWER( )
- EXP( )
- SQRT( )
- SIGN( )
- ABS( )
- MOD( )
- LOG( )
- SIN( )
- TAN( )
- COS( )

ROUND()

ROUND(valor, [n]) Arredonda o valor. Se n for omitido arredonda para as unidades. Se n for positivo arredonda na posição n à direita da virgula. Se n for negativo arredonda na posição n à esquerda da virgula.

Exemplo:

SELECT ROUND(45.923,1), ROUND(45.923), ROUND(45.323,1), ROUND(254.323,-1)

CEIL() e FLOOR()

CEIL(valor) Determina o menor inteiro **maior ou igual** que o valor introduzido como argumento.

Exemplo:
SELECT CEIL(99.1), CEIL(101.76),
CEIL(-11.1)

FLOOR(valor) Determina o maior inteiro **menor ou igual** que o valor introduzido como parâmetro.

Exemplo:
SELECT FLOOR(99.1), FLOOR(101.76),
FLOOR(-11.1)

POWER() e EXP()

POWER(valor, n) Eleva à potência de n o valor. Pode ser negativo.

Exemplo:
SELECT POWER(2,10), POWER(sal,2)
FROM emp;

EXP (n) Devolve o valor 'e' elevado a n. e=2.7182818

Exemplo:
SELECT EXP(1), EXP(4);

SQRT()

SQRT(valor) Devolve a raiz quadrada de valor.

Exemplo:

```
SELECT SQRT(144), SQRT(9), SQRT(sal)
FROM emp WHERE comissao > 0;
```

SIGN()

SIGN(valor) Devolve -1 se for negativo, 0 se for nulo (zero), 1 se for positivo.

Exemplos:

```
SELECT SIGN(20), SIGN(-20), SIGN(0)

SELECT sal, comissao, sal-comissao,
SIGN(sal-comissao)
FROM emp WHERE deptno = 30;
```

ABS()

ABS(valor) Devolve o valor absoluto de valor.

Exemplo:

```
SELECT sal, comissao, comissao-sal,
ABS(comissao-sal)
FROM emp WHERE deptno = 30;
```

MOD()

MOD(valor1,valor2) Devolve o resto da divisão de valor1 por valor2.

Exemplo:

```
SELECT sal, comissao,
MOD(sal,comissao), MOD(100,40)
FROM emp WHERE deptno = 30
ORDER BY comissao;
```

LOG(), SIN(), TAN(), COS()

- **LOG(m,n)** - logarítmo de base m de n;
- **SIN(n)** - seno de n (em radiano);
- **TAN(n)** - tangente de n (em radiano).
- **COS(n)** - coseno de n (em radiano);

Funções de manipulação de datas

Veremos algumas:

CURDATE();
NOW();
DATE_FORMAT();
EXTRACT();
DAY(), MONTH() e YEAR();
DATE_ADD();
DATEDIFF();
PERIOD_DIFF();
DAYOFYEAR();

CURDATE() e NOW()

Instrução SQL para retornar data atual do servidor.

Exemplo:

```
SELECT CURDATE();
ou
SELECT CURRENT_DATE();
```

A Função **NOW();** retorna além da data a hora atual.

Exemplo:

```
SELECT NOW();
```

DATE_FORMAT()

DATE_FORMAT(data, formato) altera o formato da data.

Exemplo:
SELECT DATE_FORMAT(CURDATE() , '%d/%m/%Y');

OBS: O parâmetro "%" é obrigatório antes de informar os caracteres de formato. Abaixo seguem alguns formatos aceitos:

Especificação	Descrição
%d	Dia do mês numérico(00..31)
%D	Dia do mês com sufixo (em Inglês)
%m	Mês, numérico(00..12)
%M	Nome do Mês(em Inglês)
%y	Ano, numérico (dois dígitos)
%Y	Ano, quatro dígitos numéricos

EXTRACT()

EXTRACT(tipo FROM data) extrai o tipo de informação sobre a data passada como parâmetro.

Exemplo:

```
SELECT
EXTRACT(DAY FROM CURDATE()) AS DIA,
EXTRACT(MONTH FROM CURDATE()) AS MES,
EXTRACT(YEAR FROM CURDATE()) AS ANO;
```

DAY() MONTH() e YEAR()

Uma maneira mais simples de fazer a mesma coisa:

Exemplo:

```
SELECT
DAY(CURDATE()) AS DIA,
MONTH(CURDATE()) AS MES,
YEAR(CURDATE()) AS ANO;
```

DATE_ADD()

DATE_ADD(data, INTERVAL qtd DAY) recebe uma **data** e acrescenta a quantidade (**qtd**) de dias, meses ou anos nela.

Exemplo:

```
SELECT DATE_ADD(CURDATE(), INTERVAL 30 DAY));
```

OBS: Pode-se usar DAY, MONTH ou YEAR.

DATEDIFF()

DATEDIFF(data1,data2) retorna a diferença em dias entre a data1 e data2.

Exemplo:

```
SELECT DATEDIFF('2012-08-21', '2012-08-05');
```

PERIOD_DIFF()

PERIOD_DIFF(data1,data2) usamos pra retornar a diferença entre data1 e data2 em meses. Para isso basta informar como parâmetro o período no formato de “YYMM” ou “YYYYMM”.

Exemplo:

```
SELECT PERIOD_DIFF('201212', '201208');
```

Contas entre datas

Também é possível realizar contas entre datas de forma simples usando operadores aritméticos:

Exemplos:

```
SELECT MONTH('2013-12-01')-MONTH('2013-05-25');
```

Resulta: 7 meses.

```
SELECT YEAR('2012-05-02')-YEAR('1983-07-03');
```

Resulta: 29 anos.

DAYOFYEAR()

DAYOFYEAR(data) retorna o dia do ano (1-366).

Exemplo:

```
SELECT DAYOFYEAR('2012-08-21');
```

GREATEST(v1,v2,v3) Devolve o maior dos valores v1, v2 ou v3. A função pode receber mais que 3 argumentos.

Exemplo:
SELECT GREATEST(1,2,3);

Algumas funções extras

CASE WHEN campo (= > < >= <=) valor1 THEN valor2 ELSE valor3 END AS apelido comparar o valor de um campo com valor1, se positivo retorna valor2, se negativo retorna valor3.

Exemplo:

```
SELECT enome, funcao,
CASE   WHEN funcao = 'BALCONIST' THEN 'Peão'
       WHEN funcao = 'VENDEDOR' THEN 'Peão que vende'
       WHEN funcao = 'GERENTE' THEN 'Chefe'
       WHEN funcao = 'ANALISTA' THEN 'Amigo do chefe'
      WHEN funcao = 'PRESIDENTE' THEN 'Manda-chuva'
      ELSE 'DESCONHECIDO'
      END AS "Visto como"
FROM EMP
```

Algumas funções extras

Exercícios02