

Cálculos e Funções em SQL – resultados numéricos e alfanuméricos

Banco de Dados II

Profa. Simone Carboni Garcia



Cálculos

- Podem ser feitos cálculos quando são realizadas buscas em um banco de dados, aplicando operadores aritméticos a colunas.
- Por exemplo: é possível obter, mesmo não tendo armazenado, o preço de um item caso este sofra um aumento de 5%.

```
SELECT cod_produto, nm_produto, preco_venda * 1,05  
FROM CD;
```

Cálculos

- Operadores utilizados na SQL:

Operador	Ação
+	Soma
-	Subtração
*	Multiplicação
/	Divisão

- A precedência dos operadores ocorre como na matemática: multiplicação e divisão precedem soma e subtração.
- O uso de parênteses é utilizado para alterar as prioridades.

$8 / 2 * 2$ é diferente de: $8 / (2 * 2)$

Resultados numéricos

Função Position

- Retorna a posição do caractere de busca em uma cadeia de caracteres.

```
SELECT position ('Silv' IN Marcos Silveira) tamanho;
```

tamanho
<hr/>
8

```
SELECT position ('Silv' IN nome) tamanho FROM clientes;
```

Função Length

- Retorno o número de caracteres contidos em uma cadeia de caracteres.

```
SELECT length ('Maria Flores') tamanho;
```

tamanho
<hr/>
12

```
SELECT length (nome) tamanho FROM clientes;
```

Função de agregação COUNT

- Retorna o total de ocorrências com base em um filtro.

- Sintaxe:

```
SELECT count (atributo)  
FROM <nome da tabela>  
WHERE [condição];
```

- Exemplo 1: Total de linhas da tabela Cliente.

```
SELECT count(*)  
FROM cliente;
```

- Exemplo 2: Quantas compras foram realizadas no dia 09/11/2006?

```
SELECT count (dt_compra)  
FROM Venda  
WHERE dt_compra = '2006/11/09';
```

Função de agregação SUM

- Retorna o somatório de um conjunto valores.

- Sintaxe:

```
SELECT sum (atributo)  
FROM <nome da tabela>  
WHERE [condição];
```

- Exemplo 1: Soma da quantidade de produtos vendidos para o cliente de código igual a 1.

```
SELECT sum (qt_produto)  
FROM venda  
WHERE cd_cliente = 1;
```


Função de agregação AVG

- Retorna a média de valores.

- Sintaxe:

```
SELECT avg (atributo)  
FROM <nome da tabela>  
WHERE [condição];
```

- Exemplo 1: Obter média de vendas do produto arroz no dia “09/11/2006” .

```
SELECT avg (qt_produto)  
FROM venda  
WHERE cd_produto=1 AND dt_compra='2006/11/09'
```

Função de agregação MAX

- Retorna o maior valor de um conjunto de valores.
- Sintaxe:
SELECT **max** (atributo)
FROM <nome da tabela>
WHERE [condição];
 - Exemplo 1: Buscar a maior quantidade de vendas de um produto.
SELECT **max** (qt_produto)
FROM Venda;

Função de agregação MIN

- Retorna o menor valor de um conjunto de valores.
- Sintaxe:
SELECT **min**(atributo)
FROM <nome da tabela>
WHERE [condição];
 - Exemplo 1: Buscar a menor quantidade de vendas de um produto.
SELECT **min**(qt_produto)
FROM Venda;

Resultados alfanuméricos

Concatenação de um conjunto de caracteres

- Para concatenar um conjunto de caracteres, utiliza-se dois pipes ||
- Exemplo: Para listar o nome de um cliente seguido de seu endereço:
SELECT nm_cliente || ' - ' || endereco
FROM clientes;

Maria Silveira - Tiradentes

José da Silva - Av. Bento Gonçalves

Roberto Pereira - Dom Pedro II

Função UPPER

- Quando realizamos buscas em um banco de dados o conteúdo do campo é comparado, literalmente, com a cadeia de caracteres inserida no filtro da busca.
- Exemplo:

```
SELECT * FROM clientes  
WHERE nm_cliente LIKE 'JOÃO%';
```



```
SELECT * FROM clientes  
WHERE upper (nome) LIKE 'JOÃO%';
```
- **UPPER** converte todos os caracteres para maiúsculos e depois compara com a condição (no caso LIKE 'JOÃO%').

Função LOWER

- **LOWER** converte todos os caracteres para minúsculos e depois compara com a condição, como no exemplo:

```
SELECT * FROM clientes
```

```
WHERE lower (nome) LIKE 'joão%';
```

Função SUBSTRING

- Retorna uma parte de uma cadeia de caracteres.

substring (cadeia_caracteres FROM posição_início
FOR número_de_caracteres)

```
SELECT substring (nome FROM 1 FOR 5) resultado  
FROM clientes WHERE nome LIKE 'M%';
```

```
SELECT substr (nome,1,1) resultado  
FROM clientes WHERE nome LIKE 'M%';
```

```
resultado  
Maria  
Marco
```


Manipulação de datas e horas

CURRENT_DATE

- Retorna a data atual (ano-mês-dia)

```
SELECT current_date;
```

CURRENT_TIME

- Retorna a hora atual

```
SELECT current_time;
```

Função EXTRACT

- Extrai e retorna um valor de um campo do tipo data ou hora. É possível extrair o dia, o mês, o ano, a hora, o minuto ou o segundo.

```
SELECT extract (month FROM current_date);
```

```
SELECT extract (year FROM current_date);
```

```
SELECT extract (day FROM current_date);
```

```
SELECT extract (hour FROM current_time);
```

```
SELECT extract (minute FROM current_time);
```

```
SELECT extract (second FROM current_time);
```

```
SELECT extract (milliseconds FROM current_time);
```

Função DATE_PART

- Retorna parte de uma data ou hora.

```
SELECT date_part('day', current_date);
```

```
SELECT date_part('month', current_date);
```

```
SELECT date_part('year', current_date);
```

```
SELECT date_part('hour', current_time);
```

```
SELECT date_part('minute', current_time);
```

```
SELECT date_part('second', current_time);
```

```
SELECT date_part('milliseconds', current_time);
```

Função AGE

- Retorna o tempo em anos, meses e dias.

```
SELECT age (timestamp '2000-10-12');
```

```
SELECT age (timestamp '2000-10-12',  
            timestamp '2001-10-12');
```

Função TO_CHAR

- Converte vários tipos dados em uma cadeia de caracteres.

```
SELECT to_char(current_date, 'DD/MM/YYYY');
```

TMmonth

- Retorna um inteiro que representa o mês da data especificada.
- TM = texto na língua portuguesa

```
SELECT to_char(current_date, 'TMmonth');
```


Interval

- Pode ser utilizado para realizar cálculos.

```
SELECT date_current + interval '7' day;
```

```
SELECT data_lancamento + interval '2' month FROM CD;
```

Obtenção de número de caracteres e número de bits

Função CHAR_LENGTH

- Retorna o número de caracteres de uma cadeia de caracteres.

```
SELECT char_length('Ana');
```

```
SELECT char_length(nome) from clientes;
```

Função BIT_LENGTH

- Retorna o número de bits que uma cadeia de caracteres ocupa em memória.

```
SELECT bit_length('Paulo');
```

```
SELECT bit_length(nome), nome from clientes;
```