

INSTITUTO FEDERAL DE SÃO PAULO (IFSP)

CAMPUS SÃO MIGUEL PAULISTA (SMP)

DISCIPLINA: BDS2

Profª Vanessa e Prof. Akira

LISTA SOBRE FUNCTIONS

1. Crie uma função chamada Saudacao que não recebe nenhum parâmetro e retorna a string "Olá, estudante! Seja bem-vindo ao MySQL".

SQL

```
SELECT Saudação();
```

2. Crie uma função chamada CalcularDobro que recebe um número inteiro (numero) como parâmetro e retorna o dobro desse número.

SQL

```
SELECT CalcularDobro(15); -- Deve retornar 30
```

3. Crie uma função chamada NomeCompleto que recebe dois parâmetros, primeiro_nome e sobrenome, ambos do tipo VARCHAR, e retorna a string completa (ex: "Fulano de Tal").

SQL

```
SELECT NomeCompleto('Maria', 'Silva'); -- Deve retornar "Maria Silva"
```

4. Crie uma função chamada CalcularTempo que recebe a data de validade de um produto (data_validade do tipo DATE). A função deve calcular e retornar o tempo de validade do produto em meses.

SQL

```
SELECT nome, CalcularTempo(data_validade) FROM Produto;
```

5. Crie uma função chamada EhMaiorIdade que receba como parâmetro a data de nascimento de uma pessoa (tipo DATE) e retorne TRUE ou FALSE (ou 1/0) indicando se a pessoa tem 18 anos ou mais.

6. Crie uma função Categoricalidade que receba a idade de uma pessoa (tipo INT) e retorne um texto com a categoria:

"Criança" (0 a 12)

"Adolescente" (13 a 17)

"Adulto" (18 a 59)

"Idoso" (60+)

Use estrutura IF ... THEN ... ELSEIF ...

7. Crie uma função chamada AplicarDesconto que recebe uma forma de pagamento (forma_pagamento do tipo VARCHAR) como parâmetro. A função deve retornar a string "Desconto de 5%" se a forma_pagamento for "Dinheiro", e "Sem desconto" caso contrário.

SQL

```
SELECT AplicarDesconto("Dinheiro"); -- Deve retornar "Desconto de 5%"
```

```
SELECT AplicarDesconto("Crédito"); -- Deve retornar "Sem desconto"
```

8. Crie uma função chamada `CalcularDesconto` que receba como parâmetros o valor original de um produto (do tipo `DECIMAL`) e o percentual de desconto (do tipo `INT`) e retorne o valor final com o desconto aplicado.
9. Crie uma função chamada `SomaCompra` que receba o id de um cliente (`id_cliente`). A função deve calcular a soma de todas as compras desse cliente e retornar o resultado.
SQL
`SELECT SomaCompra(21);` -- Assume que o cliente com `id_cliente` 21 existe
10. Crie uma função chamada `PrimeiraLetraMaiuscula` que receba uma string (palavra). A função deve retornar a string com a primeira letra em maiúscula e o restante em minúscula.
SQL
`SELECT PrimeiraLetraMaiuscula('maria');` -- Deve retornar "Maria"
`SELECT PrimeiraLetraMaiuscula('JOAO');` -- Deve retornar "Joao"
Dica: Pesquise sobre as funções `UPPER()`, `LOWER()` e `CONCAT()` no MySQL.
11. Crie uma função `FormatarDataBR` que receba uma data no formato `DATE` e retorne uma `VARCHAR` com a data no formato "dd/mm/aaaa".
Use `DATE_FORMAT(data, '%d/%m/%Y')`
`DETERMINISTIC`
12. Crie uma função chamada `FuncionarioEstrela` que não receba nenhum parâmetro. A função deve identificar o funcionario que fez mais vendas finalizadas (na tabela `Compra`) e retornar o nome dele.
SQL
`SELECT FuncionarioEstrela();`