

Aula 06 – Linguagem SQL DDL / FUNCTION

1. INTRODUÇÃO

Uma *function* (função) em *Structured Query Language* (SQL), são blocos de código em SQL que executam cálculos e retornam um único valor. As funções podem ser usadas dentro dos comandos SQL DML, procedimentos armazenados (*Stored Procedures*) e gatilhos (*triggers*).

2. TIPOS DE FUNÇÃO

- Funções embutidas (*Built-in Functions*)
- Funções definidas pelo usuário (*UDF - User Defined Functions*)

3. FUNÇÕES EMBITUDAS

- **Funções Matemáticas**

- `SELECT ABS(-10);`
- `SELECT ROUND(3.1415, 2);`
- `SELECT CEIL(4.3);`
- `SELECT FLOOR(4.9);`

- **Funções de Texto**

- `SELECT UPPER('sql');`
- `SELECT LOWER('SQL');`
- `SELECT LENGTH('Cristiano Martins Nunes');`
- `SELECT CONCAT('Olá', 'Seu Nome ', 'Seja bem-vindo ao Mundo de DevSQL');`

- **Funções de Data e Hora**

- `SELECT NOW();`
- `SELECT CURDATE();`
- `SELECT YEAR('2025-04-04');`
- `SELECT DATEDIFF(NOW(), '2025-01-01');`

- **Funções de Agregação**

- `SELECT COUNT (*) FROM CLIENTE;`
- `SELECT AVG(CODIGO) FROM COLABORADOR.`
- `SELECT SUM(NOTA) FROM ESTUDANTE;`
- `SELECT MIN(IDADE) FROM CLIENTE;`
- `SELECT MAX(IDADE) FROM CLIENTE;`

4. FUNÇÕES DEFINIDAS PELO USUÁRIO

As *FUNCTIONS* (funções), como os procedimentos armazenados (*STORED PROCEDURE*), também são blocos de códigos SQL armazenados no servidor de banco de dados (SGBD). Apesar de serem comparadas as *PROCEDURES*, conceitualmente seus objetivos divergem bastante.

As funções não têm como objetivo realizar transações rotineiras complexas, mas sim, pequenas operações, normalmente auxiliares, que possam ser solicitadas em um processo de transação; já as funções definidas pelo usuário (*UDF - User Defined Functions*), tornam-se responsáveis por tratamento de textos, variáveis, regras do negócio e realização de cálculos específicos, tendo sempre como entrada um ou mais parâmetros e retornando

apenas um. Todas as funções podem trabalhar em conjunto c/os procedimentos armazenados, através da troca de parâmetros, execução, realização cálculos e retorno de valores.

- **Sintaxe 01 - UDF**

DELIMITER \$\$

CREATE FUNCTION Nome da função (...)

RETURNS Tipo de dado

DETERMINISTIC | NOT DETERMINISTIC

BEGIN

Bloco de Comandos SQL;

END;

DELIMITER ;

- **Componentes da Sintaxe**

- **DELIMITER \$\$**

- Altera o delimitador temporariamente para evitar conflitos com ; dentro da função

- **CREATE FUNCTION nome_da_funcao(...)**

- Define o nome e os parâmetros da função.

- **RETURNS tipo_de_retorno**

- Indica o tipo de dado retornado (*INT, DECIMAL, VARCHAR...*)

- **DETERMINISTIC | NOT DETERMINISTIC**

- Indica que a função sempre retorna o mesmo resultado para os mesmos valores de entrada.

- Indica que a função não retorna o mesmo resultado para os mesmos valores de entrada.

- **BEGIN ... END**

- Bloco de comandos SQL onde a lógica (programação) da função é definida.

- **RETURN**

- Define o valor (variável) que será retornado pela função.

- **DELIMITER ;**

- Restaura o delimitador padrão (;).

- **Sintaxe 02 – UDF**

DROP FUNCTION IF EXISTS Nome da função;

5. PRÁTICA 01

A) *CREATE DATABASE UDF;*

B) *USE UDF;*

C) *CREATE TABLE CLIENTE*
(CODIGO INT AUTO_INCREMENT,
NOME VARCHAR (50),
DATA CAD DATETIME,
PRIMARY KEY (CODIGO));

D) *SELECT * FROM CLIENTE;*

E) *INSERT INTO CLIENTE (NOME, DATA CAD)*
VALUES ('CRISTIANO MARTINS NUNES', NOW()), ('Faculdade Anhanguera', NOW()),
('Betim Minas Gerais', NOW());

6. PRÁTICA 02

- SGBD MYSQL
- FUNÇÕES UDF
- FUNÇÃO BÁSICA
- FUNÇÃO DECLARAÇÃO DE VARIÁVEL
- FUNÇÃO ESTRUTURA CONDICIONAL
- FUNÇÃO ESTRUTURA DE REPETIÇÃO CONDICIONAL

7. BIBLIOGRAFIA:

- <https://www.mysql.com/>
- DATE, C. J.. Introdução a Sistemas de Banco de Dados. Campus, 2000.
- RAMEZ, Elsmari; NAVATHE, S. B. Sistemas de Banco de Dados. 4. ed. São Paulo: Addison Wesley, 2005.
- https://www.w3schools.com/mysql/mysql_ref_functions.asp
- <https://www.mysqltutorial.org/>
- <https://roadmap.sh/sql>