=======================================================

**FUNCTION**

=======================================================

FUNCTION é uma função armazenada onde é possível passar parâmetros para ela e então retornar um valor armazenados dentro do servidor MySQL.

----------------------------------------------------------------------------------------------

<https://www.youtube.com/watch?v=WTTGml9Co90>

----------------------------------------------------------------------------------------------

**PARA CRIAR UMA FUNÇÃO SE USA:**

DELIMITER //

CREATE FUNCTION <Nome Da Função> (Parâmetro 1 Tipo de dados, Parâmetro 2 Tipo de Dados , ...)

RETURNS <Tipo de Dado da Função>

BEGIN

<Declaração de Variáveis>,

<Definição de Instruções SQL>,

RETURN <Informação que a Função irá Retornar>;

END; //

=======================================================

**TRIGGER**

=======================================================

Um trigger (Gatilho) é um objeto associado a uma tabela, que é utilizado para responder a um evento específico quando for disparado.

Os triggers podem ser disparados para responderem antes ou depois de um evento.

Por exemplo você tem um estoque de produtos com a relação de quantidade, onde foi vendido 10 de 1000 produtos em estoque, a função TRIGGER serve para indicar que houve alteração de valor (1000 - 10) e sua mudança (990).

----------------------------------------------------------------------------------------------

<https://www.youtube.com/watch?v=OfzSftfAQEg>

----------------------------------------------------------------------------------------------

**PARA CRIAR UM TRIGGER SE USA:**

DELIMITER //

CREATE TRIGGER trigger\_name trigger\_time trigger\_event

ON (local)

FOR EACH ROW (Vai executar para todas as linhas)

BEGIN

<Declaração de Variáveis>,

<Definição de Instruções SQL>,

END; //

* trigger\_name: define o nome do procedimento, por exemplo, trg\_test;
* trigger\_time: define se o TRIGGER será ativado antes (BEFORE) ou depois (AFTER) do comando que o disparou;
* trigger\_event: aqui se define qual será o evento, INSERT, REPLACE, DELETE ou UPDATE;

=======================================================

**VIEW**

=======================================================

Tem como função de mecanismo de busca, onde você insere o dado e ela devolve a informação para você, podendo ser filtrado.

Uma view é um objeto que é formado por declarações SELECTs, que retornam uma visualização de dados específica de uma ou mais tabelas de um banco de dados. Esses objetos também são conhecidos como virtual tables (tabelas virtuais), justamente por não fazerem parte do esquema físico da base.

Uma view pode ser utilizada com um conjunto de tabelas que podem ser agregadas a outros conjuntos através do comando UNION. Também pode aumentar a segurança da base por definir políticas de acesso em nível de tabela e coluna.

----------------------------------------------------------------------------------------------

<https://www.youtube.com/watch?v=twqE0a7ckYg>

----------------------------------------------------------------------------------------------

**PARA CRIAR UM VIEW SE USA:**

CREATE VIEW <Nome Da VIEW> AS

SELECT <Código SQL>;

=======================================================

**STORED PROCEDURE**

=======================================================

É uma sub-rotina que fica armazenada no banco de dados. Uma PROCEDURE tem um nome, uma lista de parâmetros e declarações de comandos SQL, podem executar funções complexas, retornando ou não as informações para o usuário.

----------------------------------------------------------------------------------------------

<https://www.youtube.com/watch?v=tgzlIYhXlkQ>

----------------------------------------------------------------------------------------------

**PARA CRIAR UMA PROCEDURE SE USA:**

DELIMITER //

CREATE PROCEDURE <Nome Do Procedure> (<param1>[tipo], <param2>[tipo],...)

BEGIN

<Declaração de Variáveis>,

<Definição de Instruções SQL>,

RETURN <Informação que a Função irá Retornar>;

END; //

CALL <Nome do Procedure> ;