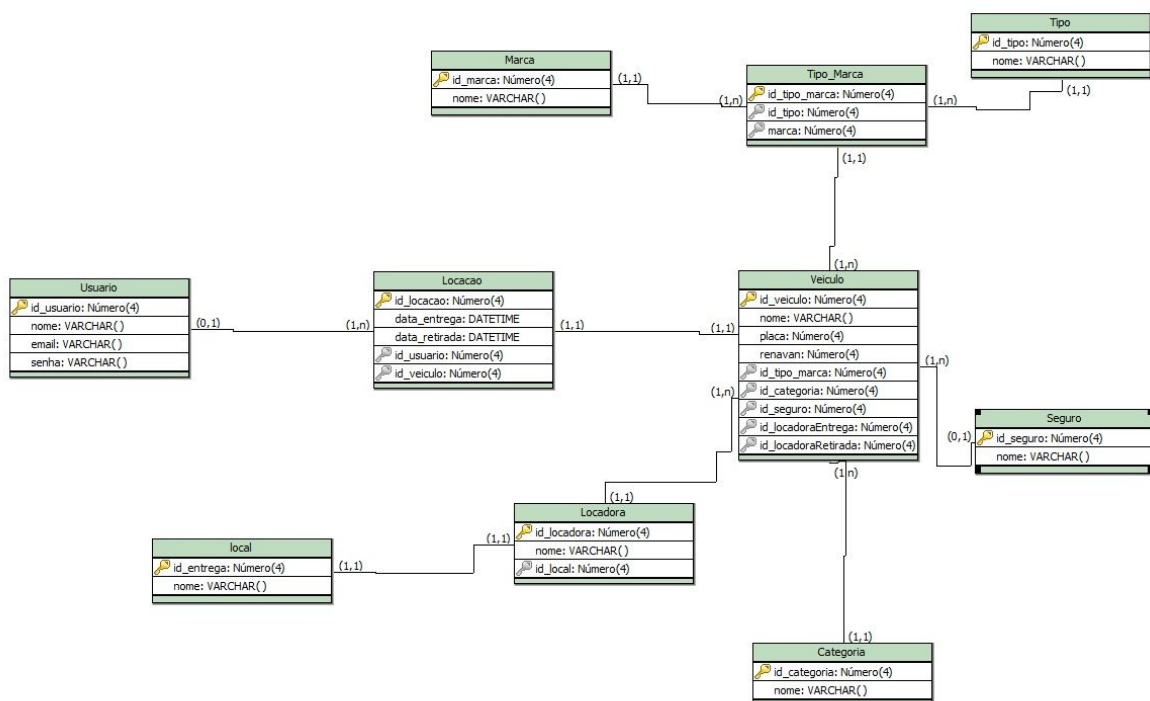


# Gatilhos e Funções

## Base de dados - Projeto Integrador

Modelo Lógico



Abaixo um arquivo com a base de dados - Tabelas (SQL)

<https://s3-us-west-2.amazonaws.com/secure.notion-static.com/8bcef152-938f-47a3-9b99-11c27ee54f3a/ScriptSQL.txt>

## Gatilhos

Uma função de gatilho pode ser criada para executar antes (BEFORE) ou após (AFTER) as consultas INSERT, UPDATE OU DELETE, uma vez para cada registro (linha) modificado ou por instrução SQL. Logo que ocorre um desses eventos do gatilho a função do gatilho é disparada automaticamente para tratar o evento.

- É criado pelo comando CREATE FUNCTION;
- Declara o procedimento como uma função sem argumentos e que retorna o tipo trigger;
- Os argumentos do gatilho são passados através de TG\_ARGV.

### Exemplo:

O gatilho deste exemplo garante que quando é deletado ou modificada uma linha na tabela, fica sempre registrado na tabela bkp\_usuario, o usuário que efetuou a exclusão ou a modificação, e a hora em que isto ocorreu.

```
CREATE TABLE bkp_usuario (  
  id_usuario serial primary key,  
  data date not null default CURRENT_DATE,  
  email varchar(255) not null,  
  nome varchar(255) not null,  
  senha varchar(255) not null,  
  data_exclusao timestamp  
);  
  
CREATE OR REPLACE FUNCTION backup_usuario()  
RETURNS TRIGGER AS  
$$  
  BEGIN  
    INSERT INTO bkp_usuario  
      (id_usuario, data, email, nome, senha, data_exclusao)  
    VALUES  
      (OLD.id_usuario, OLD.data, OLD.email,  
       OLD.nome, OLD.senha, NOW() );  
    RETURN NEW;  
  END;  
$$ LANGUAGE plpgsql;
```

```
CREATE TRIGGER trigger_usuario AFTER DELETE
    ON usuario FOR EACH ROW
EXECUTE PROCEDURE backup_usuario();

Delete from usuario where id_usuario = 1;
```

### Exemplo Prático:

<https://www.loom.com/share/934115f1836a4043a8934a7ffd6a6ec4>

## Funções

Elas são semelhantes aos Stored Procedures. Por padrão o PostgreSQL só dar suporte as funções da linguagem SQL. Dessa forma, para suportar funções de outras linguagens é preciso desenvolver e efetuar procedimentos.

- Podem ser chamadas sempre que necessárias;
- Devem possuir nome;
- Sempre retornam algum valor;
- Aceitam parâmetros;
- Podem ser utilizadas como parte de uma expressão, como por exemplo selects e joins.

### Exemplo:

Segue um exemplo simples de utilização de function para contar a quantidade de locações de um usuário passando com parâmetros o id do usuário.

```
CREATE FUNCTION n_locacoes(INTEGER)
RETURNS INT8 AS '
    SELECT COUNT(*) FROM locacao
    WHERE locacao.id_usuario = $2;
'
LANGUAGE SQL;
```

Para chamar a função é preciso utilizar o código a abaixo.

```
SELECT n_locacoes(4);
```

### Exemplo Prático:

<https://www.youtube.com/watch?v=KnVqfRRRbS4>