



**Instituto Federal**  
Campus Goiânia

**Bacharelado em Sistemas de Informação**

# Banco de Dados II



**Prof. Dory Gonzaga Rodrigues**





## Agenda

- Objetos Avançados
  - TRIGGERS





## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Objetos Avançados: **VARIÁVEIS DE AMBIENTE EM UMA TRIGGER**

Quando trabalhamos com funções de gatilho, diversas variáveis de ambiente são criadas automaticamente e podem ser manipuladas dentro da função.

**NEW**

**OLD**

**TG\_WHEN**

**TG\_LEVEL**

**TG\_OP**

**TG\_RELID**

**TG\_NAME**

**TG\_TABLE\_SCHEMA**

**TG\_TABLE\_NAME**

**TG\_NARGS**

**TG\_ARGV []**





## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Objetos Avançados: **VARIÁVEIS DE AMBIENTE EM UMA TRIGGER**

Quando trabalhamos com funções de gatilho, diversas variáveis de ambiente são criadas automaticamente e podem ser manipuladas dentro da função.

#### **NEW**

Esta variável armazena:

- A nova linha de banco de dados para operações de **INSERT** ou **UPDATE**;
- Utilizada em funções de gatilho do tipo Nível de Linha.

Esta variável irá conter **NULL**:

- Em operações **EXCLUIR**;
- Utilizada em funções de gatilho do tipo Nível de instrução.





## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Objetos Avançados: **VARIÁVEIS DE AMBIENTE EM UMA TRIGGER**

Quando trabalhamos com funções de gatilho, diversas variáveis de ambiente são criadas automaticamente e podem ser manipuladas dentro da função.

#### **OLD**

Esta variável armazena:

- A antiga linha de banco de dados para operações de **UPDATE** ou **DELETE**;
- Utilizada em funções de gatilho do tipo Nível de Linha.

Esta variável irá conter **NULL**:

- Em operações **INSERT**;
- Nas funções de gatilho do tipo Nível de instrução.





## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Objetos Avançados: **NEW / OLD** utilizado no **RETURN** da **FUNÇÃO**

- Os gatilhos no nível de linha disparados **BEFORE**:

Para operações de **INSERT** ou **UPDATE**, o valor de retorno deve ser **NEW**

Para operações de **DELETE**, o valor de retorno deve ser **OLD**.

- Os gatilhos no nível de linha disparados **AFTER** ou nível de instrução:

Para operações **INSERT**, **UPDATE** ou **DELETE**, o valor de retorno é sempre ignorado e pode ser especificado como **NULL**;





## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Objetos Avançados: **VARIÁVEIS DE AMBIENTE EM UMA TRIGGER**

Quando trabalhamos com funções de gatilho, diversas variáveis de ambiente são criadas automaticamente e podem ser manipuladas dentro da função.

#### **TG\_WHEN**

Esta variável armazena:

- Os valores **BEFORE** ou **AFTER**, de acordo com o que foi definido na criação do gatilho.





## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Objetos Avançados: **VARIÁVEIS DE AMBIENTE EM UMA TRIGGER**

Quando trabalhamos com funções de gatilho, diversas variáveis de ambiente são criadas automaticamente e podem ser manipuladas dentro da função.

#### **TG\_LEVEL**

Esta variável armazena:

- Os valores **ROW** ou **STATEMENT**, de acordo com o que foi definido na criação do gatilho.







## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Objetos Avançados: **VARIÁVEIS DE AMBIENTE EM UMA TRIGGER**

Quando trabalhamos com funções de gatilho, diversas variáveis de ambiente são criadas automaticamente e podem ser manipuladas dentro da função.

#### **TG\_OP**

Esta variável armazena:

- Os valores **INSERT** ou **UPDATE** ou **DELETE**, informando para qual operação o gatilho foi disparado.





## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Atividade

1) Crie uma única função de gatilho que registre as ações de auditoria, na tabela Gravadora\_Auditoria, para as operações de INSERT, UPDATE e DELETE ocorridas na tabela Gravadora. Deve ser registradas as seguintes informações na tabela de auditoria:

- Os valores de todos os campos da tabela Gravadora;
- A data e hora da operação;
- O usuário que realizou a operação;
- O tipo da operação;





## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Objetos Avançados: **VARIÁVEIS DE AMBIENTE EM UMA TRIGGER**

Quando trabalhamos com funções de gatilho, diversas variáveis de ambiente são criadas automaticamente e podem ser manipuladas dentro da função.

#### **TG\_RELID**

Esta variável armazena:

- O valor **ID do objeto da tabela**, informando qual tabela disparou o gatilho.





## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Objetos Avançados: **VARIÁVEIS DE AMBIENTE EM UMA TRIGGER**

Quando trabalhamos com funções de gatilho, diversas variáveis de ambiente são criadas automaticamente e podem ser manipuladas dentro da função.

#### **TG\_NAME**

Esta variável armazena:

- O **Nome do Gatilho**, informando qual foi efetivamente o gatilho disparado.





## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Objetos Avançados: **VARIÁVEIS DE AMBIENTE EM UMA TRIGGER**

Quando trabalhamos com funções de gatilho, diversas variáveis de ambiente são criadas automaticamente e podem ser manipuladas dentro da função.

#### **TG\_TABLE\_SCHEMA**

Esta variável armazena:

- O **Nome do Schema**, informando o schema que contém a tabela que causou o disparo do gatilho.





## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Objetos Avançados: **VARIÁVEIS DE AMBIENTE EM UMA TRIGGER**

Quando trabalhamos com funções de gatilho, diversas variáveis de ambiente são criadas automaticamente e podem ser manipuladas dentro da função.

#### **TG\_TABLE\_NAME**

Esta variável armazena:

- O **Nome da Tabela**, informando o nome da tabela que causou o disparo do gatilho.





## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### - Objetos Avançados: **VARIÁVEIS DE AMBIENTE EM UMA TRIGGER**

Quando trabalhamos com funções de gatilho, diversas variáveis de ambiente são criadas automaticamente e podem ser manipuladas dentro da função.

#### **TG\_NARGS**

Esta variável armazena:

- O **número de argumentos**, informando o número de argumentos fornecidos à função de gatilho na instrução **CREATE TRIGGER**.

#### **TG\_ARGV[]**

Esta variável armazena:

- Uma **matriz de texto**, informando os argumentos da instrução CREATE TRIGGER.





## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### 2) Atividade (Valor 1.0)

2a) Criar a tabela de **ENTRADA\_PRODUTO** de CDs como segue:

```
CREATE TABLE entrada_produto (  
  
    id          SERIAL,  
    idcd        INTEGER NOT NULL,  
    qtde        INTEGER NOT NULL,  
    v_unitario  NUMERIC(14,2) NULL DEFAULT '0.00',  
    dt_entrada  DATE NULL DEFAULT NULL,  
);
```







## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### 2) Atividade (Valor 1.0)

2b) Criar a tabela de **SAIDA\_PRODUTO** de CDs como segue:

```
CREATE TABLE saida_produto (  
  
    id          SERIAL,  
    idcd        INTEGER NOT NULL,  
    qtde        INTEGER NOT NULL,  
    v_unitario  NUMERIC(14,2) NULL DEFAULT '0.00',  
    dt_entrada  DATE NULL DEFAULT NULL,  
);
```





## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### 2) Atividade (Valor 1.0)

2c) Criar a tabela de **ESTOQUE** de CDs como segue:

```
CREATE TABLE estoque (  
    id    SERIAL,  
    idcd  INTEGER NOT NULL,  
    qtde  INTEGER NOT NULL,  
    v_venda NUMERIC(14,2) NULL DEFAULT '0.00'  
);
```





## OBJETOS AVANÇADOS

### SQL – AVANÇADA

#### 2) Atividade (Valor 1.0)

2d) Criar um procedimento que atualize o quantitativo do produto em estoque e o valor unitário do estoque quando ocorrer uma operação de inserção de produtos na tabela entrada ou na tabela saída de produto. Observe que deve ser tratada as situações onde podemos ter a inserção de um novo produto no estoque ou apenas a atualização do quantitativo e do valor unitário em estoque para produtos já cadastrados.

