

Agenda:

- TRIGGERS

O que é uma trigger /gatilho em PL/SQL?

- A trigger é um objeto de banco de dados que dispara automaticamente em resposta a eventos específicos quando realizamos **INSERT, UPDATE, DELETE**.
- Também podemos utilizar trigger para automatizar ações, aplicar regras de negócios, manter integridade da base de dados e até mesmo para auditar mudanças em uma determinada tabela.

Tipos de Triggers

- **BEFORE :**

São disparadas/acionadas antes que uma ação seja executada . Podemos utilizar para modificar valor em uma coluna antes que um **insert** ou **update** seja realizado.

- **AFTER :**

São disparadas ou acionadas depois que uma ação seja executada. Podemos utilizar para executar funções adicionais , como inserir valor de auditoria em uma tabela após um **update** ou **delete**.

- **INSTEAD OF:**

Utilizada para interceptar comandos, **DML(INSERTUPDATE, DELETE)** e executar o bloco **PL/SQL** que esta na trigger, esse comando é útil quando estamos utilizando **DML** em uma **view**. Exe.: quando vamos realizar insert em uma view que não tem a coluna da tabela origem que não permite **not null**.

Sintaxe da Trigger

Create or replace trigger **nome_da_trigger**

[BEFORE|AFTER|INSTEAD OF] definição de quando a trigger será executada.

Especifica a instrução **DML**

Especificamos a coluna a ser atualizada **[Of nome_coluna]**

Especificamos a tabela associada a trigger **[ON nome_tabela]**

[REFERENCING OLD AS o NEW AS n]: neste ponto pegamos os valores antigos ou novos para varias instruções **DML**.

[FOR EACH ROW]: se quisermos que a trigger seja executada em nível de linha.

When (Condicional): Caso utilizamos triggers em nível de linha, podemos utilizar condicional quando a trigger for disparada.

Vamos praticar

```
CREATE TABLE PEDIDO_NOVOS AS SELECT * FROM PEDIDO;
```

```
ALTER TABLE PEDIDO_NOVOS ADD STATUS VARCHAR2(30);
```

```
SELECT * FROM PEDIDO_NOVOS;
```

```
CREATE OR REPLACE TRIGGER trg_pedido
BEFORE INSERT ON PEDIDO_NOVOS
FOR EACH ROW
BEGIN
    -- Atualiza o status do pedido para "Novo" após a inserção
    IF :NEW.STATUS IS NULL THEN
        :NEW.STATUS := 'Novo Pedido';
    END IF;
END;
```

```
create table TB_AUDITORIA  
(  
  id      NUMBER generated always as identity,  
  tabela  VARCHAR2(30),  
  operacao VARCHAR2(30),  
  data    DATE,  
  usuario VARCHAR2(30)  
)
```

```
CREATE OR REPLACE TRIGGER trg_auditoria
  AFTER INSERT OR UPDATE OR DELETE ON pedido_novos
  FOR EACH ROW
DECLARE
  operacao      VARCHAR2(30);
  nome_usuario  VARCHAR2(100);

BEGIN
  -- Determina a operação realizada (INSERT, UPDATE ou DELETE)
  IF INSERTING THEN
    operacao := 'INSERT';
  ELSIF UPDATING THEN
    operacao := 'UPDATE';
  ELSIF DELETING THEN
    operacao := 'DELETE';
  END IF;

  -- Obtém o nome de usuário da sessão atual
  nome_usuario := SYS_CONTEXT('USERENV', 'SESSION_USER');

  -- Registra a auditoria na tabela de auditoria
  INSERT INTO TB_AUDITORIA
    (tabela, operacao, DATA, USUARIO)
  VALUES
    ('PEDIDO_NOVOS', operacao, sysdate, nome_usuario);
END;
```


Exercícios

1. Crie uma trigger para registrar as alterações de DATA DE ENTREGA na tabela PEDIDO.
2. crie uma trigger que consulta a quantidade de itens na tabela ITEM_PEDIDO e some o total de produto do mesmo pedido , caso esse total seja maior que 20 itens aplique um desconto automático que 20% .
3. crie uma trigger em na tabela de clientes para validar se CPF/CNPJ contenha somente valores numéricos.