



# Triggers

**Curso de Introdução a Oracle 11g:  
SQL/Avançado**  
**Prof.: Marlon Mendes Minussi**  
**[marlonminussi@gmail.com](mailto:marlonminussi@gmail.com)**

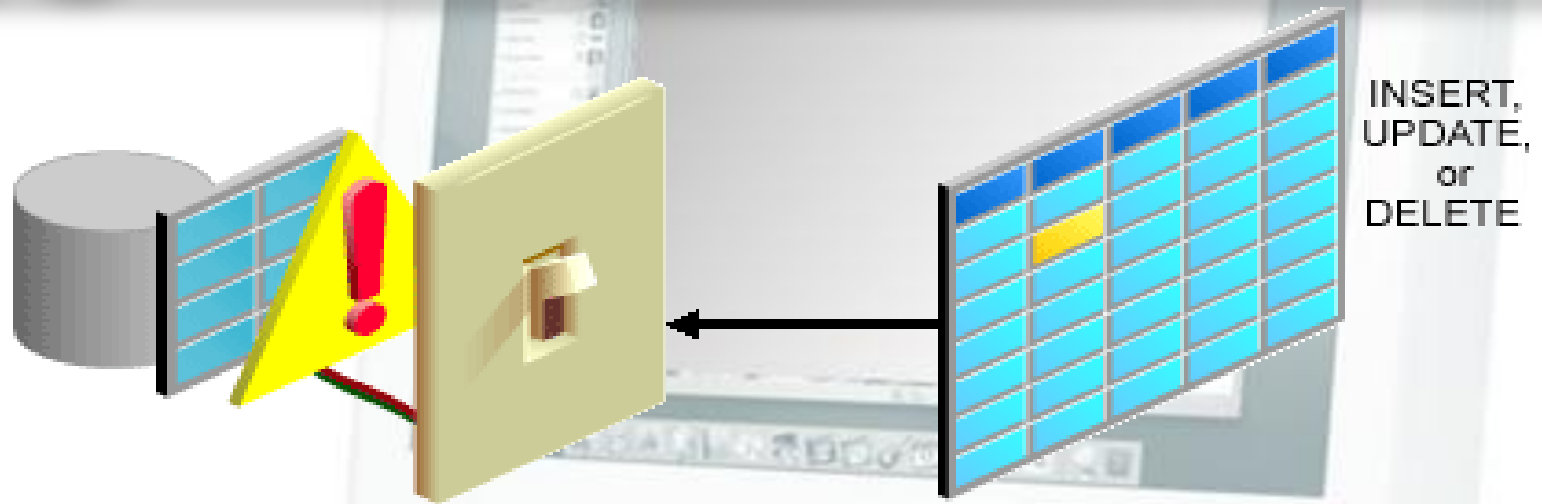


# Triggers

- Blocos PL/SQL armazenados, mas que são disparados automaticamente sempre que ocorrer um evento associado a uma tabela ou visão.
- Não recebe, explicitamente, parâmetros de entrada e tampouco de saída
- Benefícios:
  - Automatiza processos e regras;
  - Melhora a segurança no que diz respeito ao acesso aos dados;
  - Preserva a integridade dos dados através da implementação de condições complexas de integridade referencial.



# Triggers



- As triggers ficam armazenadas no banco de dados e são executadas em uma operação INSERT, UPDATE OU DELETE de uma tabela do banco de dados.



# Triggers

```
CREATE OR REPLACE TRIGGER nome-trigger
  TEMPO EVENTO1 [ OR EVENTO2 ] [ OR EVENTO3 ]
  ON nome-tabela
  [ FOR EACH ROW ]
  [ WHEN condição ]
  DECLARE
    var1 tipo(tamanho);
    varN tipo(tamanho);
  BEGIN
    Comandos;
  END nome-trigger;
```

TEMPO = BEFORE ou AFTER

EVENTO = INSERT, UPDATE  
ou DELETE



# Triggers

- Trigger de Comando: executada uma única vez, independentemente do número de linhas afetadas pelo comando que causou o seu disparo → sem FOR EACH ROW(para cada linha)
- Trigger de Linha: executada para cada linha afetada pelo comando que causou o seu disparo → com FOR EACH ROW
- Para o controle das alterações de linha são utilizados dois registros:
  - NEW: armazena os dados da linha inserida (INSERT) ou os novos dados da linha alterada (UPDATE)
  - OLD: armazena os dados da linha excluída (DELETE) ou os antigos dados da linha alterada (UPDATE)



# Triggers

- Segundo Goya(2006a), as partes que compõem uma trigger são as seguintes:
  - Momento: BEFORE (tabela), AFTER (tabela) e INSTEAD (view);
  - Evento: INSERT, UPDATE ou DELETE;
  - Tipo: de linha ou instrução;
  - Corpo: bloco de código PLSQL



# Momento

- O momento define quando uma triggers será executada, que pode ser:
- 1º BEFORE (tabela): indica que os comandos PL/SQL, do corpo da trigger serão executados ANTES de os dados da tabela serem alterados. Quando por exemplo salvar o valor de uma coluna antes de alterar o valor dela.

- Ex:

```
CREATE OR REPLACE TRIGGER novo_pac  
  BEFORE...
```

```
  .
```

```
  .
```

```
  .
```

```
END;
```

```
/
```



# Momento

- 2º AFTER (tabela): indica que os comandos PL/SQL do corpo da trigger serão executados após os dados da tabela serem alterados.
- Ex:

```
CREATE OR REPLACE TRIGGER novo_pac  
  AFTER...
```

```
  .
```

```
  .
```

```
  .
```

```
END;
```

```
/
```





# Momento

- **INSTEAD (view):** indica que a trigger irá ser executada no lugar da instrução que disparou a trigger. Literalmente, a instrução é substituída pela trigger. Essa técnica permite que façamos, por exemplo, alterações em uma tabela através de uma view.
- Ex: 

```
CREATE OR REPLACE TRIGGER novo_pac
    INSTEAD OF INSERT ON vpac
    FOR EACH ROW
    WHEN ...
    :
END; /
```



# Trigger

- Predicativos:

```
CREATE OR REPLACE TRIGGER tg-exemplo
BEFORE INSERT OR DELETE OR UPDATE ON tabela
BEGIN
    IF Inserting
    THEN ...
    ELSIF Deleting
    THEN ...
    ELSIF Updating('coluna')
    THEN ...
    END IF;
END TRIGGER;
```



# Trigger

Ex.:

```
//CRIANDO TRIGGER PARA CONTROLAR A SEQUENCE COD_ATENDIMENTO  
CREATE TRIGGER ATENDIMENTO_TG BEFORE INSERT  
ON ATENDIMENTO  
FOR EACH ROW  
DECLARE  
    vSeq NUMBER;  
BEGIN  
    SELECT SEQ_COD_ATENDIMENTO.NEXTVAL  
        INTO vSeq  
    FROM DUAL;  
    :NEW.COD_ATENDIMENTO := vSeq;  
END;  
/
```



# Exemplo Controle de Acesso

```
CREATE TABLE controle_acesso(  
    descricao varchar2(50),  
    data date  
);
```

```
CREATE OR REPLACE TRIGGER controle_logon AFTER LOGON ON DATABASE  
BEGIN  
    INSERT INTO controle_acesso values('Acessado por ' || USER, SYSDATE);  
    COMMIT;  
END;
```

```
SELECT SUBSTR(descricao,1,20),  
       TO_CHAR(data,'DD/MM/YYYY HH24:MI:SS') DATA  
FROM controle_acesso  
ORDER BY data;
```



# Exemplos

- Ex1:

```
CREATE OR REPLACE TRIGGER tg_Controle  
BEFORE INSERT OR UPDATE ON exame  
FOR EACH ROW  
WHEN (TO_NUMBER(TO_CHAR(Sysdate,'hh24')) NOT BETWEEN 8  
AND 18)  
BEGIN  
    Raise_Application_Error(-20000,'Não é permitido incluir ou  
    alterar funcionário nesse horário');  
END tg_Controle;  
/
```



# Exemplos

- Ex2:

```
CREATE OR REPLACE TRIGGER tg_aumento
BEFORE UPDATE OF salario ON medico
FOR EACH ROW
BEGIN
    IF (:NEW.salario - :OLD.salario) < :OLD.salario * 0.25
    THEN
        RAISE_APPLICATION_ERROR (-20512, 'Favor corrigir
aumento');
    END IF;
END;
```



# Exemplos

```
Ex 3:
CREATE OR REPLACE TRIGGER tg_valida_salario
BEFORE INSERT ON medico
FOR EACH ROW
BEGIN
    IF (:NEW.SALARIO > 10000)
    THEN
        RAISE_APPLICATION_ERROR (-20500,'Não e permitido cadastrar salário maior que 10.000');
    ELSE
        IF (TO_CHAR(SYSDATE,'DY') IN ( 'SAB','DOM')) OR (TO_CHAR(SYSDATE, 'HH24:MI') NOT
        BETWEEN '08:30' AND '18:30')
        THEN
            RAISE_APPLICATION_ERROR(-20500, 'Só e permitido cadastrar das 7:30 as 18:30 de
            seg. a sexta.');
```

Para testar:

```
INSERT INTO medico VALUES(4,'Dr. Barone', 45745128 ,'Clínico Geral','(54)9999666',11000)
/
```



# Trigger noBrowser

<http://st-curriculum.oracle.com/tutorial/DBXETutorial/html/module8/cretrig.htm>



Criando uma trigger para auditoria.

<http://st-curriculum.oracle.com/tutorial/DBXETutorial/html/module8/audittrig.htm>





# Exemplos

- Exclusão de trigger:

```
DROP TRIGGER tg_Nome_trigger;
```

- Desabilitação/Habilitação de trigger:

```
ALTER TRIGGER tg_Nome_trigger DISABLE |  
ENABLE;
```

```
ALTER TABLE nome_tabela DISABLE | ENABLE ALL  
TRIGGERS;
```



# Exercícios:

1. Criar uma Trigger para a tabela paciente do Schema Clinica, com as mesmas restrições que o exemplo utilizado para a tabela Employeers.
2. Criar uma trigger que não permita inserir um novo atendimento com valor de desconto igual a ZERO, usando when.



# Bibliografia

- RAMALHO, José Antonio. *Oracle 10g. São Paulo : Pioneira Thomson Learning, 2005.*
- <http://st-curriculum.oracle.com/tutorial/DBXETutorial/index.htm>
- <http://www.linhadecodigo.com.br/Artigo.aspx?id=322&pag=1>