

# Objetivos da Aula:

## - Triggers

Professora: Juliana

**19/06/2020**

# Triggers

- São Unidades de Programa (Bloco PLSQL/Procedimentos) que são executadas, automaticamente, **antes** ou **depois** da **ocorrência de um evento**.

- **Ocorrências de evento**  **Comando disparador**

**DML (update, insert ou delete)**

# Trigger

## Exemplos de Utilização:

- Atualização do estoque.
- Implantação de regras de *negócios* (Ex. Envio de e-mail ao fornecedor para compra de produtos quando atingir a quant. *Mínima em estoque*)
- Criação de valores. (Ex. Ao cadastrar um novo funcionário realiza a soma do salários dos funcionários e atualiza na tabela departamento).
- Manutenção de um registro histórico das alterações.
- Implantação de segurança (auditoria).

# Trigger - Sintaxe

**CREATE** [OR REPLACE] **TRIGGER** [esquema.] nome\_trigger

[BEFORE ou AFTER]

[INSERT OR DELETE OR UPDATE]

[OF coluna]

ON [esquema.]tabela\_nome

[FOR EACH ROW]

**WHEN** (condição)

DECLARE

variáveis

BEGIN

corpo\_trigger

END;



O código contido no trigger será executado para cada linha .

BLOCO PL/SQL

# Trigger – Exemplo 1

5

Um gatilho para uma tabela de funcionários

```
CREATE OR REPLACE TRIGGER Imp_Altera_Salarial
BEFORE DELETE OR INSERT OR UPDATE
                                ON Emp_tab
FOR EACH ROW WHEN ( new.EmpID > 0 )
DECLARE    sal_diff    number ;
BEGIN
    sal_diff := :new.sal - :old.sal ;
    dbms_output.put_line ( 'Old salary: ' || :old.sal ) ;
    dbms_output.put_line ( ' New salary: ' || :new.sal ) ;
    dbms_output.put_line ( ' Difference ' || sal_diff ) ;
END;
```

# Criar tabela emp\_tab

---

```
create table emp_tab  
( empid number(10) primary key,  
  emp_nome varchar2(30),  
  sal number(10,2),  
  depid number(10));
```

# Executando o Trigger Imp\_altera\_Salarial.

;

The screenshot displays the Oracle SQL Developer interface. The left-hand pane shows the 'Conexões' (Connections) tree with the 'Triggers' folder expanded, highlighting the 'IMP\_ALTERA\_SALARIAL' trigger. The main 'Query Builder' window contains the following SQL script:

```
SET SERVEROUTPUT ON SIZE UNLIMITED;  
  
insert into emp_tab values (1,'Gabriel',6000.00,1);
```

Below the query editor, the 'Saída do Script' (Script Output) window shows the execution results:

```
Old salary:  
New salary:6000  
Difference:  
  
1 linha inserido.
```

The status bar at the bottom of the output window indicates 'Tarefa concluída em 0,085 segundos' (Task completed in 0.085 seconds).

# Executando o Trigger Imp\_altera\_Salarial.

The screenshot displays the Oracle SQL Developer interface. The left sidebar shows the 'Conexões' (Connections) tree with the 'Triggers' folder expanded, highlighting the 'IMP\_ALTERA\_SALARIAL' trigger. The main window shows the 'Query Builder' tab with the following SQL statement:

```
update emp_tab
set sal=10000
where empid=1;
```

Below the query, the 'Saída do Script' (Script Output) window shows the execution results:

```
Old salary:6000
New salary:10000
Difference:4000

1 linha atualizado.
```

The status bar at the bottom indicates the current position is 'Linha 1 Coluna 1' and the window is titled 'Windows: CF'.



# Trigger – Exemplo 1 (COMENTÁRIOS)

9

- Comentários sobre o exemplo anterior
  - A ativação do **gatilho ocorre** quando **comandos DML** são executados sobre a tabela de funcionários
  - Uso de **BEFORE** possibilita o acesso a valores antigos e novos
  - Uso de **FOR EACH ROW** gera várias execuções do gatilho
- Por exemplo, o comando DML abaixo:

```
UPDATE emp_tab  
SET sal= (sal*0.1) + sal  
Where depid=1;
```



Ativa o gatilho  
uma vez para  
*cada tupla*  
atualizada

- Pode ser omitida se deseja-se apenas registrar a ocorrência do evento.

# Triggers

10

- Ativação de Triggers
  - Ativando Gatilhos **UMA ou VÁRIAS** vezes
    - ❑ A opção **FOR EACH ROW**:
      - ✓ Determina se o gatilho é do tipo **row trigger** ou **statement trigger**.
      - ✓ Se especificada, o gatilho é executado UMA vez para CADA tupla afetada pelo evento.
      - ✓ Se omitida, o gatilho é executado UMA ÚNICA vez para cada ocorrência de evento

# Ativando Gatilho várias vezes Imp\_altera\_Salarial.

The screenshot shows the Oracle SQL Developer interface. The left pane displays the database schema, including triggers like DEF\$\_AQCALL, DEF\$\_AQERROR, DEF\$\_PROPAGATOR\_TRIG, IMP\_ALTERA\_SALARIAL, and REPCATLOGTRIG. The main window shows a SQL script with two insert statements:

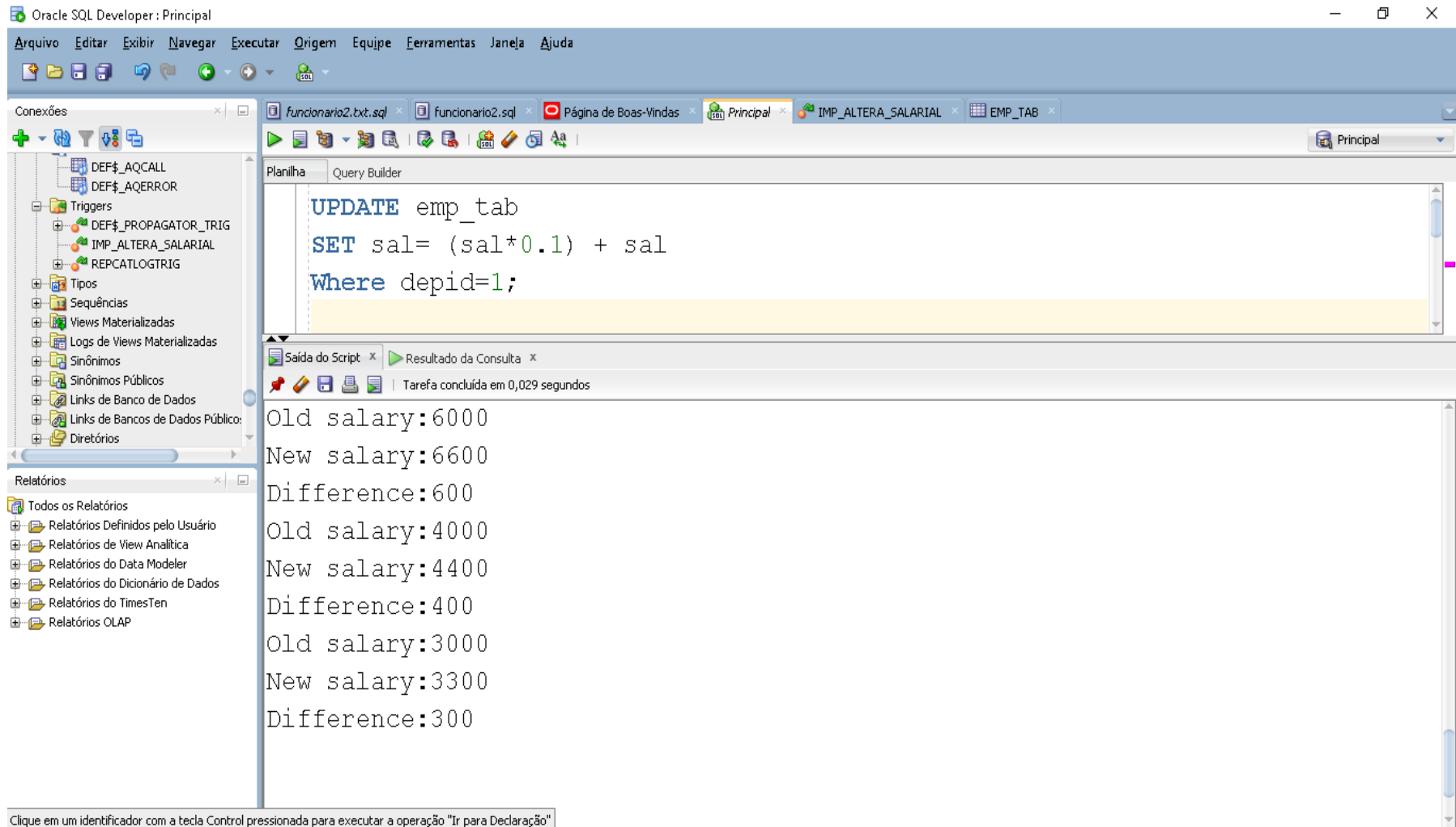
```
insert into emp_tab values (2, 'Pedro', 4000, 1);  
insert into emp_tab values (3, 'João', 3000, 1);
```

The script is executed, and the output pane shows the results of the trigger execution:

```
Old salary:  
New salary:4000  
Difference:  
  
1 linha inserido.  
  
Old salary:  
New salary:3000  
Difference:
```

The status bar at the bottom indicates the current position in the script: "Linha 1 Coluna 1 | Inserir | Modificado | Windows: CF".

# Ativando Gatilho várias vezes Imp\_altera\_Salarial.



Oracle SQL Developer : Principal

Arquivo Editar Exibir Navegar Executar Origem Equipe Ferramentas Janela Ajuda

Conexões

- DEF\$\_AQCALL
- DEF\$\_AQERROR
- Triggers
  - DEF\$\_PROPAGATOR\_TRIG
  - IMP\_ALTERA\_SALARIAL
  - REPCATLOGTRIG
- Tipos
- Sequências
- Views Materializadas
- Logs de Views Materializadas
- Sinônimos
- Sinônimos Públicos
- Links de Banco de Dados
- Links de Bancos de Dados Públicos
- Diretórios

Relatórios

- Todos os Relatórios
- Relatórios Definidos pelo Usuário
- Relatórios de View Analítica
- Relatórios do Data Modeler
- Relatórios do Dicionário de Dados
- Relatórios do TimesTen
- Relatórios OLAP

funcionario2.txt.sql funcionario2.sql Página de Boas-Vindas Principal IMP\_ALTERA\_SALARIAL EMP\_TAB

Planilha Query Builder

```
UPDATE emp_tab
SET sal= (sal*0.1) + sal
Where depid=1;
```

Saída do Script x Resultado da Consulta x

Tarefa concluída em 0,029 segundos

Old salary:6000  
New salary:6600  
Difference:600  
Old salary:4000  
New salary:4400  
Difference:400  
Old salary:3000  
New salary:3300  
Difference:300

Clique em um identificador com a tecla Control pressionada para executar a operação "Ir para Declaração"

# Ativando Gatilho várias vezes

13

## ➤ Ativação de Triggers

### ■ Ativando Gatilhos UMA ou VÁRIAS vezes

```
create or replace
trigger aumenta_salario
after update on emp_tab
for each row when (new.sal>1000)
begin
    insert into emp_log (emp_id,data_emp,novo_sal,msg)
values (:new.empid,sysdate,:new.sal,'Novo salário
Superavaliado');
end;
```

```
UPDATE emp_tab
SET sal= (sal*0.1) + sal
Where depid=1;
```

Se existem 5 funcionários no departamento 20, o gatilho é executado 5 vezes quando o UPDATE é processado.

## Ativando Gatilho UMA vez

14

- Ativação de Triggers
  - Ativando Gatilho UMA vez

```
CREATE OR REPLACE TRIGGER aumento_salarial  
AFTER UPDATE ON Emp_tab  
BEGIN  
    INSERT INTO Emp_log (Data, Msg)  
    VALUES (SYSDATE, 'Mudança em Emp_tab ');  
END;
```

```
UPDATE Emp_tab SET Sal = Sal + 1000.0  
WHERE Deptno = 20;
```

Por outro lado, o gatilho acima é executado apenas UMA vez para cada solicitação de UPDATE.

```
create or replace TRIGGER TR_EMP
BEFORE INSERT OR UPDATE ON EMP_tab
FOR EACH ROW
BEGIN
  IF INSERTING THEN
    IF :NEW.SAL > 5000 THEN
      DBMS_OUTPUT.PUT_LINE ('Inserido' || :NEW.EMP_ID || 'com salário maior que R$5000,00');
    END IF;
  END IF;
  IF UPDATING THEN
    IF :NEW.SAL > 5000 THEN
      DBMS_OUTPUT.PUT_LINE ('Atualizado' || :NEW.EMP_ID || 'com salário maior que R$5000,00');
    END IF;
  END IF;
END TR_EMP;
```