



Sistemas de Informação

Prof. Me. Alexandre Barcelos
profalexandre.barcelos@fiap.com.br

Criação de Gatilhos

Objetivos

- Descrever Triggers de Banco de Dados e seu uso.
- Criar Trigger de Banco de Dados.
- Descrever Regras de Disparo de Triggers de Banco.
- Remover Triggers de Banco.

Tipos de Gatilhos

- Um gatilho:

- É um bloco PL/SQL ou um procedimento PL/SQL associado a uma tabela, visão, esquema ou banco de dados
- Executa implicitamente sempre que um evento ocorre.

-

- 1

Criando Gatilhos

```
CREATE [OR REPLACE ] TRIGGER trigger_name
{BEFORE | AFTER | INSTEAD OF }
{INSERT [OR] | UPDATE [OR] | DELETE}
[OF col_name]
ON table_name
[REFERENCING OLD AS o NEW AS n]
[FOR EACH ROW]
WHEN (condition)
DECLARE
    Declaration-statements
BEGIN
    Executable-statements
EXCEPTION
    Exception-handling-statements
END;
```

Onde temos:

- **CREATE [OR REPLACE] TRIGGER trigger_name:**
 - Cria uma nova trigger ou sobrescreve uma já existente com o nome informado, que no caso é apresentado como trigger_name.
- **{BEFORE | AFTER | INSTEAD OF }:**
 - Define quando a trigger deve ser executada. A cláusula INSTEAD OF é utilizada quando uma trigger é criada em uma view.
- **{INSERT [OR] | UPDATE [OR] | DELETE}:**
 - Especifica a operação DML.
- **[OF col_name]:**
 - Especifica a coluna que será atualizada.
- **ON table_name:**
 - Especifica o nome da tabela que será associada a trigger.
- **[REFERENCING OLD AS o NEW AS n]:**
 - Neste ponto, permitimos a referência a valores novos e valores antigos para várias declarações DML, assim como seria o caso para o insert e o delete.
- **[FOR EACH ROW]:**
 - Esta instrução especifica uma trigger no nível de linha. Neste caso, a trigger deveria ser escutada para cada linha afetada.
- **WHEN (condição):**
 - Este trecho da instrução provê a condição para que a trigger seja disparada nas

linhas. Esta cláusula só é válida para triggers em nível de linha.

+ Criando Gatilhos

- Um gatilho de aplicativo dispara uma vez para uma declaração DML.
- Um gatilho de linha dispara uma vez para cada linha afetada.

Tipos de Gatilho

— Gatilho de declaração:

- Dispara uma vez para uma declaração DML.
- Executa uma vez para o evento
- É o tipo de gatilho padrão
- Acionado uma vez, mesmo que nenhuma linha seja afetada

Tipos de Gatilho

— Gatilho de linha:

Dispara uma vez para cada linha afetada.

- Executa uma vez por cada linha afetada pelo evento desencadeante
- Não é executado se o evento desencadeante não afetar nenhuma linha
- É definido especificando a cláusula FOR EACH ROW

Trigger Timing

- Quando o gatilho pode ser disparado?

- BEFORE

- O código no corpo do trigger será executado antes do evento DML que causou o disparo.

- AFTER

- O código no corpo do trigger será executado depois do evento DML que causou o disparo.

- INSTEAD OF

- O código no corpo do trigger será executado no instante da execução da declaração do evento DML que causou o disparo. Usado para VIEWS que não são modificáveis (Resultado de joins).

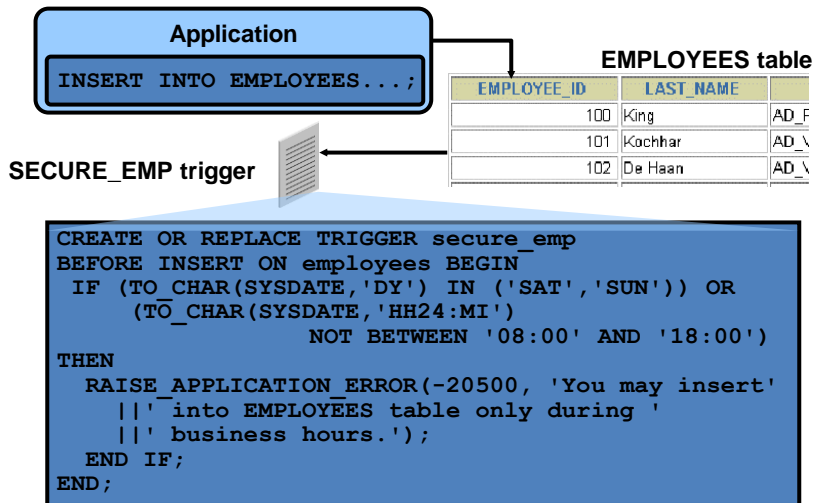
Evento de gatilho

Um evento de gatilho determina qual declaração DML faz com que o gatilho seja executado

— Os tipos são:

- INSERT
- UPDATE [OF column]
- DELETE

Criando um Gatilho



+ Testando o gatilho

SECURE_EMP

```
INSERT INTO employees
VALUES (300, 'Smith', 'Rob', 'RSMITH', null,
SYSDATE, 'IT_PROG', 4500, null, null, 60);
```

```
INSERT INTO employees (employee_id, last_name, first_name, email,
*)
```

ERROR at line 1:

ORA-20500: You may insert into EMPLOYEES table only during business hours.

ORA-06512: at "PLSQL_SECURE_EMP", line 4

ORA-04088: error during execution of trigger 'PLSQL_SECURE_EMP'

Usando Predicados Condicionais

```
CREATE OR REPLACE TRIGGER secure_emp BEFORE
INSERT OR UPDATE OR DELETE ON employees BEGIN
  IF (TO_CHAR(SYSDATE,'DY') IN ('SAT','SUN')) OR
     (TO_CHAR(SYSDATE,'HH24')
      NOT BETWEEN '08' AND '18') THEN
    IF DELETING THEN RAISE_APPLICATION_ERROR(
      -20502,'You may delete from EMPLOYEES table'||
      'only during business hours.');
```

```
    ELSIF INSERTING THEN RAISE_APPLICATION_ERROR(
      -20500,'You may insert into EMPLOYEES table'||
      'only during business hours.');
```

```
    ELSIF UPDATING('SALARY') THEN
      RAISE_APPLICATION_ERROR(-20503, 'You may '||
      'update SALARY only during business hours.');
```

```
    ELSE RAISE_APPLICATION_ERROR(-20504,'You may'||
      ' update EMPLOYEES table only during'||
      ' normal hours.');
```

```
  END IF;
END IF;
END;
```


Criando um gatilho de linha

```
CREATE OR REPLACE TRIGGER restrict_salary
BEFORE INSERT OR UPDATE OF salary ON employees
FOR EACH ROW
BEGIN
    IF NOT (:NEW.job_id IN ('AD_PRES', 'AD_VP'))
        AND :NEW.salary > 15000 THEN
        RAISE_APPLICATION_ERROR (-20202,
            'Employee cannot earn more than $15,000.');
```

```
END IF;
END;
/
```

Qualificadores OLD e NEW

Com o Trigger de linha, referenciar uma coluna antes da alteração com o prefixo de qualificação OLD e referenciar o valor depois de uma alteração com prefixo NEW.

OPERAÇÃO DE DADO	VALOR OLD	VALOR NEW
INSERT	NULL	VALOR INSERIDO
UPDATE	VALOR ANTES OPERAÇÃO	VALOR MODIFICADO
DELETE	VALOR ANTES DELEÇÃO	NULL

- Use esse qualificador com Trigger de linha BEFORE e AFTER.
- Não use esses qualificadores com Trigger de declarações.
- Prefixar esses qualificadores com dois pontos (:) em declarações SQL e PL/SQL.

Qualificadores OLD e NEW

```
CREATE OR REPLACE TRIGGER audit_emp_values
AFTER DELETE OR INSERT OR UPDATE ON employees
FOR EACH ROW
BEGIN
    INSERT INTO audit_emp(user_name, time_stamp, id,
        old_last_name, new_last_name, old_title,
        new_title, old_salary, new salary)
VALUES (USER, SYSDATE, :OLD.employee_id,
    :OLD.last_name, :NEW.last_name, :OLD.job_id,
    :NEW.job_id, :OLD.salary, :NEW.salary);
END;
/
```

Data Operation	Old Value	New Value
INSERT	NULL	Inserted value
UPDATE	Value before update	Value after update
DELETE	Value before delete	NULL

Qualificadores OLD e NEW

```
INSERT INTO employees
(employee_id, last_name, job_id, salary, ...)
VALUES (999, 'Temp emp', 'SA_REP', 6000,...);
```

```
UPDATE employees
SET salary = 7000, last_name = 'Smith'
WHERE employee_id = 999;
```

```
SELECT user_name, timestamp, ...
FROM audit_emp;
```

USER_NAME	TIMESTAMP	ID	OLD_LAST_N	NEW_LAST_N	OLD_TITLE	NEW_TITLE	OLD_SALARY	NEW_SALARY
PLSQL	28-SEP-01			Temp emp		SA_REP		1000
PLSQL	28-SEP-01	999	Temp emp	Smith	SA_REP	SA_REP	1000	2000

Qualificadores OLD e NEW

```
CREATE OR REPLACE TRIGGER restrict_salary
BEFORE INSERT OR UPDATE OF salary ON employees
FOR EACH ROW
BEGIN
    IF NOT (:NEW.job_id IN ('AD_PRES', 'AD_VP'))
        AND :NEW.salary > 15000 THEN
        RAISE_APPLICATION_ERROR (-20202,
            'Employee cannot earn more than $15,000.');
```

```
END IF;
END;
/
```

Criando um gatilho de visão

```
CREATE OR REPLACE VIEW REGION_COUNTRY
(REGION_ID, REGION_NAME, COUNTRY_ID, COUNTRY_NAME) AS
select
r.region_id,r.region_name,c.country_id,c.country_name
from regions r inner join countries c
on (r.region_id=c.region_id);
```

Criando um gatilho de visão

```
CREATE OR REPLACE EDITIONABLE TRIGGER
INS_REGIONS_COUNTRIES
instead of insert on region_country
declare
    e_instrg_rc exception;
    pragma exception_init(e_instrg_rc,-1);
begin
    insert into regions
    values (:NEW.region_id,:NEW.region_name);
    insert into countries
    values
    (:NEW.country_id,:NEW.country_name,:NEW.region_id);
exception
    when e_instrg_rc then
        raise_application_error(-20107,'Duplicate ID
value');
end;
/
```

Gerenciando Gatilhos

- Habilitar ou Desabilitar um gatilho de banco de dados:

```
ALTER TRIGGER trigger_name DISABLE | ENABLE
```

- Habilitar ou Desabilitar todos os gatilhos de uma tabela

```
ALTER TABLE table_name DISABLE | ENABLE  
ALL TRIGGERS
```

- Compilar um Gatilho

```
ALTER TRIGGER trigger_name COMPILE
```


Removendo Gatilhos

- Utilize o comando DROP:

```
DROP TRIGGER trigger_name;
```

- Exemplo:

```
DROP TRIGGER secure_emp;
```

OBRIGADO



profalexandre.barcelos@fiap.com.br



<https://www.linkedin.com/in/alexandrebarcelos>

FIAP

Copyright © 2023 | Professor Me. Alexandre Barcelos
Todos os direitos reservados. Reprodução ou divulgação total ou parcial deste documento, é expressamente
proibido sem consentimento formal, por escrito, do professor/autor.