Exemplo de criação de Trigger

create table FUNC(

codigo number(4,0),

nome varchar2(30),

cargo varchar2(20),

sal number(9,2)

);

INSERT INTO func VALUES (01, 'VANESSA', 'ANALISTA', 5000);

INSERT INTO func VALUES (11, 'EMMANUELE', 'DEVELOPER', 4800);

INSERT INTO func VALUES (21, 'NUBIA', 'DBA', 6800);

INSERT INTO func VALUES (31, 'ANA', 'GERENTE', 7200);

INSERT INTO func VALUES (41, 'DAFNE', 'ESTAGIARIO', 1200);

create table Historico(

codigo number(4,0),

nome varchar2(30),

cargo varchar2(20),

sal number(9,2),

data date

);

CREATE OR REPLACE TRIGGER trg\_apagados

BEFORE DELETE

ON func

REFERENCING NEW AS NEW OLD AS OLD

FOR EACH ROW

BEGIN

insert into historico values (:old.codigo, :old.nome, :old.cargo, :old.sal, sysdate);

END;

/

select \* from historico;

delete from func where codigo = 41;

select \* from historico;

Exercícios

1. Crie uma tabela de nome LOG\_FUNC, essa tabela deve conter os dados atuais e os dados anteriores da tabela FUNC e data em que a alteração foi feita. Crie uma TRIGGER que registre todas operações de UPDATE efetuadas na tabela FUNC. As alterações devem ser registradas na tabela LOG\_FUNC. O único campo que não sofre alteração na tabela FUNC é a coluna CODIGO. A TRIGGER deve gravar o dado original e o novo dado na tabela LOG\_FUNC, bem como a data quando ocorreu a alteração. Todos os campos devem ser gravados na tabela LOG\_FUNC, independente de terem sofrido alteração ou não. Se o campo não tiver sofrido alteração, tanto o campo novo quanto o antigo terão o mesmo valor.
2. Crie uma tabela chamada CONTA\_FUNC com um único campo numérico. Esse campo conterá o número total de funcionários da tabela FUNC. Crie uma TRIGGER chamada CONTA\_FUNC. Toda vez que um funcionário for contratado (INSERT), a TRIGGER deverá aumentar em um o número total de funcionários da tabela CONTA\_FUNC. Toda vez que um funcionário sair da empresa (DELETE), a TRIGGER deverá reduzir em um o número total de funcionários da tabela CONTA\_FUNC.