Pedro H. Potiguara C. 09/0128559

Pedro da Cunha Tomioka 09/0128206

Prática 6

create or replace PROCEDURE SP\_INSERE\_VALORES(

P\_TABELA IN VARCHAR2,

P\_NOME\_ATRIBUTO IN VARCHAR2,

P\_VALOR\_ATRIBUTO\_NOVO IN VARCHAR2,

P\_VALOR\_ATRIBUTO\_CHAVE IN VARCHAR2)

AS

v\_pk\_column\_id VARCHAR2(35);

v\_flag NUMBER DEFAULT 0;

E\_PK\_NAO\_SIMPLES EXCEPTION;

E\_TABELA\_NAO\_EXISTE EXCEPTION;

sql\_stmt VARCHAR2(2000);

BEGIN

--Verifica a existencia da tabela

SELECT COUNT(1)

INTO v\_flag

FROM USER\_IND\_COLUMNS

WHERE upper(TABLE\_NAME) = upper(P\_TABELA);

IF v\_flag = 0 THEN

RAISE E\_TABELA\_NAO\_EXISTE;

END IF;

-- Verifica se a chave da tabela nao eh uma unique

SELECT COUNT(1)

INTO v\_flag

FROM USER\_CONSTRAINTS US

WHERE CONSTRAINT\_TYPE = 'U' AND US.TABLE\_NAME = upper(P\_TABELA);

IF v\_flag <> 0 THEN

RAISE E\_PK\_NAO\_SIMPLES;

end if;

SELECT column\_name

INTO v\_pk\_column\_id

FROM USER\_IND\_COLUMNS

WHERE table\_name = P\_TABELA AND column\_position = 1;

sql\_stmt := 'update '|| p\_tabela || ' set ' || p\_nome\_atributo ||

' = '''|| P\_VALOR\_ATRIBUTO\_NOVO ||''' where ' || p\_tabela || '.'

|| v\_pk\_column\_id|| ' = ' || P\_VALOR\_ATRIBUTO\_CHAVE;

--Executa o comando SQL

EXECUTE immediate SQL\_STMT;

dbms\_output.put\_line('Stmt: ' || SQL\_STMT);

EXCEPTION

WHEN E\_PK\_NAO\_SIMPLES THEN

dbms\_output.put\_line('E\_PK\_NAO\_SIMPLES');

WHEN E\_TABELA\_NAO\_EXISTE THEN

dbms\_output.put\_line('E\_TABELA\_NAO\_EXISTE');

WHEN OTHERS THEN

dbms\_output.put\_line(SQLERRM || ' ERRO grave ' || SQL\_STMT );

END;

/\*

--Criando sequence para urna

CREATE SEQUENCE SQ\_NSERIAL START WITH 37 NOCACHE ORDER ;

CREATE TRIGGER BI\_NSERIAL BEFORE

INSERT ON URNA FOR EACH ROW WHEN (NEW.NSERIAL IS NULL) BEGIN :NEW.NSERIAL := SQ\_NSERIAL.NEXTVAL;

END;

/

\*/

CREATE OR replace PROCEDURE Substitui\_urna(p\_urna\_nserial NUMBER)

AS

v\_nrozona NUMBER;

v\_nserial NUMBER;

v\_flag NUMBER DEFAULT 0;

e\_nao\_possui\_urna\_reserva EXCEPTION;

BEGIN

SELECT nrozona

INTO v\_nrozona

FROM urna

WHERE nserial = p\_urna\_nserial;

SELECT nrodeurnasreservas

INTO v\_flag

FROM zona

WHERE v\_nrozona = nrozona;

-- checa se tem urna reserva

IF v\_flag = 0 THEN

RAISE e\_nao\_possui\_urna\_reserva;

END IF;

UPDATE zona

SET nrodeurnasreservas = nrodeurnasreservas - 1

WHERE nrozona = v\_nrozona;

INSERT INTO l05\_urna

(estado,

nrozona)

VALUES ( 'funcional',

v\_nrozona );

UPDATE urna

SET estado = 'manutencao'

WHERE nserial = v\_nserial;

dbms\_output.Put\_line('Urna trocada com sucesso');

EXCEPTION

WHEN e\_nao\_possui\_urna\_reserva THEN

dbms\_output.Put\_line('Nao possui urna reserva nessa sessão'); WHEN

OTHERS THEN

dbms\_output.Put\_line('Erro '

|| SQLERRM);

END;