



INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE SÃO PAULO

Câmpus Birigui - Código INEP: 150588

Rua Pedro Cavalo, 709, CEP 16201-407, Birigui (SP)

CNPJ: 10.882.594/0014-80 - Telefone: (18) 3643-1160

ATIVIDADE ASSÍNCRONA

1 - IDENTIFICAÇÃO

Curso:

BRI.BAC.CPT.2016 - BACHARELADO EM ENGENHARIA DA COMPUTAÇÃO

Data da Criação

03/10/2021

Diário/Componente Curricular:

SUP.01029 (BDAE6) - BANCO DE DADOS II - Graduação [31.70 h/38 Aulas]

Nome do Aluno

Vinicius de Souza Santos

Prontuário:

BI3008061

Professores:

Helen de Freitas Santos

2 – ATIVIDADE PROPOSTA

Exercícios para Fixação

Elaborar uma função para retornar o nome completo do cidadão, uma vez que o nome do cidadão é formado pelos atributos primeiroNome, nomeMeio e ultimoNome;

Elaborar uma função para retornar a quantidade de resíduos entregue por um cidadão, para um tipo de resíduo, em um determinado mês;

Foi identificada a necessidade de controlar estoque dos resíduos. Assim, houve uma modificação na modelagem, conforme pode ser observado no arquivo "Modelagem de Dados" contido na pasta "Aula 2021-09-28". Desta forma, é necessário elaborar um procedimento para atualizar o estoque

3 - DESENVOLVIMENTO

```
create or replace FUNCTION "FN_NOMECIDADAO" (ID IN CIDADAO.ID%TYPE) RETURN VARCHAR2 IS
P_Nome CIDADAO.PRIMEIRONOME%TYPE;
M_Nome CIDADAO.NOMEMEIO%TYPE;
U_Nome CIDADAO.ULTIMONOME%TYPE;

BEGIN
    SELECT CIDADAO.PRIMEIRONOME, CIDADAO.NOMEMEIO, CIDADAO.ULTIMONOME INTO P_Nome, M_Nome, U_Nome F
    ROM CIDADAO WHERE CIDADAO.ID = ID;

    RETURN(' Seu nome eh: ' || P_Nome || ' ' || M_Nome || ' ' || U_Nome);
END;

select FN_NOMECIDADAO (1) from dual

create or replace FUNCTION "FN_QTTRESIDUO" (IDCIDADAO IN CIDADAO.ID%TYPE, IDRESIDUO IN RESIDUO.ID%
TYPE) RETURN VARCHAR2 IS
```

```

P_Nome CIDADA0.PRIMEIRONOME%TYPE;
Quantidade_residuos RESIDUO.QTT%TYPE;

BEGIN
    SELECT CIDADA0.PRIMEIRONOME, RESIDUO.QTT INTO P_Nome, Quantidade_residuos FROM CIDADA0,RESIDUO
WHERE CIDADA0.ID = IDCIDADA0 and RESIDUO.ID = IDRESIDUO;

    RETURN('nome: ' || P_Nome || ' Entregou ' || Quantidade_residuos || ' De Residuos');
END;

select FN_QTTRESIDUO(1,1) from dual

```

```

CREATE table "ESTOQUE" (
    "ID"          NUMBER,
    "MES"         NUMBER,
    "ANO"         NUMBER,
    "QTT"         NUMBER,
    "IDTIPORESIDUO" NUMBER,
    constraint "ESTOQUE_PK" primary key ("ID")
)
/

CREATE sequence "ESTOQUE_SEQ"
/

CREATE trigger "BI_ESTOQUE"
before insert on "ESTOQUE"
for each row
begin
    if :NEW."ID" is null then
        select "ESTOQUE_SEQ".nextval into :NEW."ID" from sys.dual;
    end if;
end;
/

ALTER TABLE "ESTOQUE" ADD CONSTRAINT "ESTOQUE_FK_IDTIPORESIDUO"
FOREIGN KEY ("IDTIPORESIDUO")
REFERENCES "TIPORESIDUO" ("ID")

/

alter table "ESTOQUE" add
constraint "ESTOQUE_ID"
check (ID IS NOT NULL)
/

alter table "ESTOQUE" add
constraint "ESTOQUE_MES"
check (MES IS NOT NULL)
/

alter table "ESTOQUE" add
constraint "ESTOQUE_ANO"
check (ANO IS NOT NULL)
/

alter table "ESTOQUE" add
constraint "ESTOQUE_QTT"
check (QTT IS NOT NULL)

```

```

/
alter table "ESTOQUE" add
constraint "ESTOQUE_TIPORESIDUO"
check (IDTIPORESIDUO IS NOT NULL)
/

```

```

create or replace procedure "PR_ATUALIZARESTOQUE" (pv_opcao IN VARCHAR2,
                                                    pn_idresiduo IN NUMBER,
                                                    pd_dataentrega IN DATE,
                                                    pn_quantidade IN NUMBER,
                                                    pn_idcidade IN NUMBER,
                                                    pn_idpontocoleta IN NUMBER,
                                                    pn_idtiporesiduo IN NUMBER) is

vEXCEPTION EXCEPTION;
vn_dia NUMBER;
vn_mes NUMBER;
vn_ano NUMBER;
vn_idEstoque NUMBER;
vn_tipoResiduo NUMBER;
vn_quantidade NUMBER;
vn_antigo NUMBER;
vn_cont NUMBER;
BEGIN
    IF(UPPER(pv_opcao) = 'INSERIR') THEN
        INSERT INTO RESIDUO(DATAENTREGA, QTT, IDCIDADA0, IDPONTODECOLETA, IDTIPORESIDUO)
        VALUES (pd_dataentrega, pn_quantidade, pn_idcidade, pn_idpontocoleta, pn_idtiporesiduo);

        SELECT COUNT(*) INTO vn_mes FROM ESTOQUE WHERE EXTRACT(MONTH FROM pd_dataentrega) = ESTOQUE.MES AND EXTRACT(YEAR FROM pd_dataentrega) = ESTOQUE.ANO;
        IF(vn_mes = 0) THEN
            INSERT INTO ESTOQUE(MES, ANO, QTT, IDTIPORESIDUO) VALUES (EXTRACT(MONTH FROM pd_dataentrega), EXTRACT(YEAR FROM pd_dataentrega), pn_quantidade, pn_idtiporesiduo);
        ELSE
            SELECT ESTOQUE.QTT INTO vn_quantidade FROM ESTOQUE WHERE ESTOQUE.MES = EXTRACT(MONTH FROM pd_dataentrega) AND EXTRACT(YEAR FROM pd_dataentrega) = ESTOQUE.ANO;
            UPDATE ESTOQUE SET ESTOQUE.QTT = vn_quantidade + pn_quantidade WHERE EXTRACT(MONTH FROM pd_dataentrega) = ESTOQUE.MES AND EXTRACT(YEAR FROM pd_dataentrega) = ESTOQUE.ANO;
        END IF;
    ELSE IF(UPPER(pv_opcao) = 'ALTERAR') THEN
        SELECT COUNT(*) INTO vn_mes FROM ESTOQUE WHERE EXTRACT(MONTH FROM pd_dataentrega) = ESTOQUE.MES AND EXTRACT(YEAR FROM pd_dataentrega) = ESTOQUE.ANO;
        IF(vn_mes = 0) THEN
            INSERT INTO ESTOQUE(MES, ANO, QTT, IDTIPORESIDUO) VALUES (EXTRACT(MONTH FROM pd_dataentrega), EXTRACT(YEAR FROM pd_dataentrega), pn_quantidade, pn_idtiporesiduo);
        ELSE
            SELECT RESIDUO.QTT INTO vn_antigo FROM RESIDUO WHERE RESIDUO.ID = pn_idresiduo;
            SELECT ESTOQUE.QTT INTO vn_quantidade FROM ESTOQUE WHERE ESTOQUE.MES = EXTRACT(MONTH FROM pd_dataentrega) AND EXTRACT(YEAR FROM pd_dataentrega) = ESTOQUE.ANO;
            UPDATE ESTOQUE SET ESTOQUE.QTT = vn_quantidade -
            vn_antigo + pn_quantidade WHERE EXTRACT(MONTH FROM pd_dataentrega) = ESTOQUE.MES AND EXTRACT(YEAR FROM pd_dataentrega) = ESTOQUE.ANO;
        END IF;

        UPDATE RESIDUO SET DATAENTREGA = pd_dataentrega, QTT = pn_quantidade, IDCIDADA0 = pn_idcidade, IDTIPORESIDUO = pn_idtiporesiduo

```

```

WHERE RESIDUO.ID = pn_idresiduo;
ELSE IF(UPPER(pv_opcao) = 'DELETAR') THEN
    SELECT EXTRACT(MONTH FROM RESIDUO.DATAENTREGA) INTO vn_mes FROM RESIDUO WHERE RESIDUO.ID
= pn_idresiduo;
    SELECT EXTRACT(YEAR FROM RESIDUO.DATAENTREGA) INTO vn_ano FROM RESIDUO WHERE RESIDUO.ID =
pn_idresiduo;
    SELECT COUNT(*) INTO vn_cont FROM ESTOQUE WHERE ESTOQUE.MES = vn_mes AND ESTOQUE.ANO = vn
_ano;

    IF(vn_cont = 0) THEN
        SELECT RESIDUO.IDTIPORESIDUO INTO vn_tipoResiduo FROM RESIDUO WHERE RESIDUO.ID = pn_i
dresiduo;
        INSERT INTO ESTOQUE(MES, ANO, QTT, IDTIPORESIDUO) VALUES (vn_mes, vn_ano, 0, vn_tipoR
esiduo);
    ELSE
        SELECT ESTOQUE.ID INTO vn_idEstoque FROM ESTOQUE WHERE vn_mes = ESTOQUE.MES AND vn_an
o = ESTOQUE.ANO;
        SELECT RESIDUO.QTT INTO vn_antigo FROM RESIDUO WHERE RESIDUO.ID = pn_idresiduo;
        SELECT ESTOQUE.QTT INTO vn_quantidade FROM ESTOQUE WHERE ESTOQUE.ID = vn_idEstoque;
        UPDATE ESTOQUE SET ESTOQUE.QTT = vn_quantidade -
vn_antigo WHERE vn_mes = ESTOQUE.MES AND vn_ano = ESTOQUE.ANO;
    END IF;

    DELETE FROM RESIDUO WHERE RESIDUO.ID = pn_idresiduo;
ELSE
    RAISE vEXCEPTION;
END IF;
END IF;
END IF;
EXCEPTION
    WHEN vEXCEPTION THEN
        RAISE_APPLICATION_ERROR(-
20999,'ATENÇÃO! Operação diferente de Inserir, Deletar ou Alterar.', FALSE);
END PR_ATUALIZARESTOQUE;

```