AABD

2020/2021

G14

Meta 3

# Procedimentos

create or replace PROCEDURE chk3\_f\_transporta\_frascos(vidTransporte number, vidArmazena number, vidlocalVacina varchar , quantidade NUMBER,destino varchar, origem varchar,DATAtransporte DATE)

IS

loc number;

vac number;

armaz number;

excecao\_local exception;

Excecao\_vacina exception;

Excecao\_LocalArmazenamento exception;

BEGIN

Select count(local.LOCALVACINACAO) into loc

from local,utente,convocatoria

where local.LOCALVACINACAO=vidlocalVacina;

if loc=0 then

Raise Excecao\_Local;

end if;

Select count(armazem.id\_armazem) into armaz

from armazem,transporte

where armazem.ID\_ARMAZEM=vidArmazena;

if armaz=0 then

Raise Excecao\_LocalArmazenamento;

end if;

INSERT INTO TRANSPORTE (ID\_TRANSPORTE,ID\_ARMAZEM,LOCALVACINACAO,N\_FRASCOS,DESTINO,ORIGEM,DATA)

VALUES (vidTransporte,vidArmazena,vidlocalVacina,quantidade,destino,origem,DATAtransporte);

EXCEPTION

WHEN EXCECAO\_Local THEN

RAISE\_APPLICATION\_ERROR(-20502, 'Local de Vacinação Inexistente');

When EXCECAO\_LocalArmazenamento THEN

RAISE\_APPLICATION\_ERROR(-20508, 'Local de Armazenamento inexistente');

END;

create or replace PROCEDURE chk3\_h\_abre\_frasco( vidfrasco number)

IS

numerodoses number;

frasc number;

faberto number;

tativo number;

estadofrasco frasco.estado%TYPE;

excecao\_frasco exception;

excecao\_narmazem exception;

excecao\_faberto exception;

excecao\_tativo exception;

BEGIN

Select count(ID\_FRASCO) into frasc

from frasco

where ID\_FRASCO=vidfrasco

;

if frasc=0 then

Raise excecao\_frasco;

end if;

Select estado into faberto

from frasco

where ID\_FRASCO=vidfrasco

;

if faberto = 'Aberto' then

raise excecao\_faberto;

end if;

select count(id\_turno) into tativo

from turno

where turno.ATIVO=0;

if tativo=0 then

RAISE excecao\_tativo;

end if;

select estado into estadofrasco

from frasco

where id\_frasco = vidfrasco;

IF estadofrasco = 'Aberto' THEN

SELECT DISTINCT(seringas.quantidade)into numerodoses

FROM SERINGAS,FRASCO

WHERE SERINGAS.ID\_FRASCO = vidfrasco;

UPDATE frasco

SET n\_doses = numerodoses

WHERE id\_frasco = vidfrasco;

END IF;

EXCEPTION

WHEN EXCECAO\_frasco THEN

RAISE\_APPLICATION\_ERROR(-20511, 'Frasco Inexistente');

When EXCECAO\_narmazem THEN

RAISE\_APPLICATION\_ERROR(-20512, 'Frasco não está no local de armazenamento.');

When EXCECAO\_faberto THEN

RAISE\_APPLICATION\_ERROR(-20513, 'Frasco já foi aberto.');

When EXCECAO\_tativo THEN

RAISE\_APPLICATION\_ERROR(-20513, 'Não existe um truno de vacinação ativo.');

END;

create or replace PROCEDURE CHK3\_PROC\_A2018019746 (idenf number,numpess number)

IS

datainic date;

datafim date;

tempoenf number;

tempoturn number;

tativo number;

estado number;

enferm number;

Excecao\_enfermeiro Exception;

Excecao\_trab Exception;

BEGIN

Select count(id\_enfermeiro) into enferm

from ENFERMEIRO

where ID\_ENFERMEIRO=idenf;

if enferm=0 then

Raise Excecao\_enfermeiro;

end if;

select ENFERMEIRO.ESTADO\_ENF into estado

from enfermeiro

where ID\_ENFERMEIRO=idenf

;

if estado != 'Trabalhar' then

raise excecao\_trab;

end if;

select TURNO.DATAINICIOTURNO into datainic

from turno

where turno.ID\_ENFERMEIRO=idenf

and ativo=1;

select turno.DATAFIMTURNO into datafim

from turno

where turno.ID\_ENFERMEIRO=idenf

and ativo=1;

tempoturn:=to\_number(datafim-datainic);

tempoenf:=CHK3\_FUNC\_A2018019746(idenf,numpess);

select ativo into tativo

from turno

where turno.ID\_ENFERMEIRO=idenf

and ativo=1;

if (tempoenf >= tempoturn and tativo=1)then

UPDATE turno

SET ativo = 0

where turno.ID\_ENFERMEIRO=idenf;

end if;

EXCEPTION

When EXCECAO\_enfermeiro THEN

RAISE\_APPLICATION\_ERROR(-20504, 'Enfermeiro Inexistente');

When EXCECAO\_trab THEN

RAISE\_APPLICATION\_ERROR(-20520, 'Enfermeiro não está a trabalhar');

END;

create or replace PROCEDURE CHK3\_PROC\_A2019123778 ( vidutente number)

IS

ut number;

numerodoses number;

EXCECAO\_utente exception;

BEGIN

Select count(utente.id\_utente) into ut

from utente

where utente.ID\_UTENTE=vIdUtente;

if ut=0 then

Raise Excecao\_utente;

end if;

select distinct(COUNT(toma.ID\_TOMA)) into numerodoses

from TOMA,UTENTE

where toma.id\_utente = vidutente;

UPDATE utente

SET NDOSESTOMADAS = numerodoses

where utente.id\_utente = vidutente;

EXCEPTION

When EXCECAO\_utente THEN

RAISE\_APPLICATION\_ERROR(-20501, 'Utente Inexistente');

END;

create or replace PROCEDURE ck3\_j\_cria\_turno\_vacinacao(vidturno number ,vidlocal varchar, videnfermeiro NUMBER, datainicio DATE, datafim DATE, tempo\_por\_vacina NUMBER)

IS

numerovacinasdisponiveis number;

numeroutentes number;

estadoenfermeitov enfermeiro.ESTADO\_ENF%type;

BEGIN

select distinct(enfermeiro.estado\_enf) into estadoenfermeitov

from enfermeiro

where enfermeiro.id\_enfermeiro = videnfermeiro;

select distinct(local.NUMVACINASDISPONIVEIS) into numerovacinasdisponiveis

from local,turno

where turno.localvacinacao= local.localvacinacao and turno.localvacinacao = vidlocal;

select distinct(count(utente.id\_utente)) into numeroutentes

from local,concelho,utente,turno

where local.id\_concelho= concelho.id\_concelho and concelho.id\_concelho = utente.id\_concelho and turno.localvacinacao= local.localvacinacao and turno.localvacinacao = vidlocal;

if (numerovacinasdisponiveis >numeroutentes and estadoenfermeitov = 'Nao trabalhar' ) then

INSERT INTO turno (ID\_TURNO,ID\_enfermeiro,LOCALVACINACAO,DATAINICIOTURNO,DATAFIMTURNO,ATIVO)

VALUES (vidturno,videnfermeiro,vidlocal,datainicio,datafim,'1');

end if;

END;

# Funções

create or replace FUNCTION CHK3\_FUNC\_A2018019746(codenf number,numpac number)

RETURN number

IS

estado enfermeiro.estado\_enf%type;

tempovacinacao number;

ritmoenf number;

enferm number;

Excecao\_enfermeiro Exception;

Excecao\_trab Exception;

BEGIN

Select count(id\_enfermeiro) into enferm

from ENFERMEIRO

where ID\_ENFERMEIRO=codenf;

if enferm=0 then

Raise Excecao\_enfermeiro;

end if;

select ENFERMEIRO.ESTADO\_ENF into estado

from enfermeiro

where ID\_ENFERMEIRO=codenf

;

if estado != 'Trabalhar' then

raise excecao\_trab;

end if;

SELECT enfermeiro.RITMO\_VACINACAO INTO ritmoenf

FROM ENFERMEIRO

where ID\_ENFERMEIRO=codenf;

tempovacinacao:=ritmoenf\*numpac;

RETURN tempovacinacao;

EXCEPTION

When EXCECAO\_enfermeiro THEN

RAISE\_APPLICATION\_ERROR(-20504, 'Enfermeiro Inexistente');

When EXCECAO\_trab THEN

RAISE\_APPLICATION\_ERROR(-20520, 'Enfermeiro não está a trabalhar');

END;

create or replace FUNCTION CHK3\_FUNC\_A2019123778 (vIdFrasco number)

RETURN frasco.data\_aberto%TYPE

IS

frasc number;

Datafrasco frasco.data\_aberto%TYPE;

estadofrasco frasco.estado%TYPE;

excecao\_frasco exception;

BEGIN

Select count(id\_frasco) into frasc

from frasco

where ID\_frasco=vidfrasco;

if frasc=0 then

Raise Excecao\_frasco;

end if;

select estado into estadofrasco

from frasco

where id\_frasco = vidfrasco;

IF estadofrasco = 'Aberto' THEN

SELECT frasco.data\_aberto INTO Datafrasco

FROM frasco

where id\_frasco = vidfrasco;

END IF;

RETURN Datafrasco;

EXCEPTION

When EXCECAO\_frasco THEN

RAISE\_APPLICATION\_ERROR(-20521, 'Frasco Inexistente');

END;

create or replace FUNCTION ck3\_a\_fase\_vacinacao(codutente VARCHAR)

RETURN fase.datainicioturno%TYPE

IS

datafunction fase.datainicioturno%TYPE;

n number;

vac number;

Excecao\_utente Exception;

EXCECAO\_VACINATOMADA Exception;

BEGIN

Select count(utente.id\_utente) into n

from utente

where ID\_UTENTE=codutente;

if n=0 then

Raise Excecao\_utente;

end if;

Select count(toma.id\_toma) into vac

from toma

where toma.ID\_UTENTE=codutente;

if vac>0 then

Raise Excecao\_vacinatomada;

end if;

SELECT DISTINCT(fase.datainicioturno) into datafunction

FROM FASE, UTENTE, PROFISSAO

WHERE fase.id\_profissao = profissao.id\_profissao AND profissao.id\_profissao = utente.id\_profissao AND utente.ID\_UTENTE = codutente;

RETURN datafunction;

EXCEPTION

WHEN EXCECAO\_utente THEN --LANÇA EXCECOES

RAISE\_APPLICATION\_ERROR(-20501, 'Utente Inexistente');

When EXCECAO\_VACINATOMADA THEN

RAISE\_APPLICATION\_ERROR(-20505, 'Vacina já tomada');

END;

create or replace FUNCTION ck3\_c\_vacinas\_dadas (nomeLocal varchar, vIdVacina number)

RETURN seringas.quantidade%TYPE

IS

vacinasdadas seringas.quantidade%TYPE;

loc number;

vac number;

Excecao\_Local Exception;

EXCECAO\_VACINA Exception;

BEGIN

Select count(LOCALVACINACAO) into loc

from local

where LOCALVACINACAO=nomeLocal;

if loc=0 then

Raise Excecao\_Local;

end if;

Select count(vacina.id\_vacina) into vac

from vacina

where vacina.ID\_vacina=vIdVacina;

if vac=0 then

Raise Excecao\_vacina;

end if;

SELECT SUM(SERINGAS.QUANTIDADE) into vacinasdadas

FROM TOMA,UTENTE,CONCELHO,LOCAL,SERINGAS

WHERE local.id\_concelho = concelho.id\_concelho and utente.id\_concelho= concelho.id\_concelho and toma.id\_utente = utente.id\_utente AND TOMA.ID\_TOMA = SERINGAS.ID\_TOMA AND LOCALVACINACAO = nomeLocal;

RETURN vacinasdadas;

EXCEPTION

WHEN EXCECAO\_Local THEN --LANÇA EXCECOES

RAISE\_APPLICATION\_ERROR(-20502, 'Local Inexistente');

When EXCECAO\_VACINA THEN

RAISE\_APPLICATION\_ERROR(-20503, 'Vacina Inexistente');

END;

create or replace FUNCTION ck3\_d\_vacinas\_disponiveis (nomeLocal varchar, vIdVacina number)

RETURN number

IS

vacinasdisponiveis number;

loc number;

vac number;

Excecao\_Local Exception;

EXCECAO\_VACINA Exception;

BEGIN

Select count(LOCALVACINACAO) into loc

from local

where LOCALVACINACAO=nomeLocal;

if loc=0 then

Raise Excecao\_Local;

end if;

Select count(vacina.id\_vacina) into vac

from vacina

where vacina.ID\_vacina=vIdVacina;

if vac=0 then

Raise Excecao\_vacina;

end if;

SELECT local.numvacinasdisponiveis into vacinasdisponiveis

FROM LOCAL

WHERE LOCALVACINACAO = nomeLocal;

RETURN vacinasdisponiveis;

EXCEPTION

WHEN EXCECAO\_Local THEN --LANÇA EXCECOES

RAISE\_APPLICATION\_ERROR(-20502, 'Local Inexistente');

When EXCECAO\_VACINA THEN

RAISE\_APPLICATION\_ERROR(-20503, 'Vacina Inexistente');

END;

create or replace FUNCTION ck3\_e\_data\_ultima\_convocatoria(vIdUtente number)

RETURN fase.datainicioturno%TYPE

IS

DATAX convocatoria.data\_convocatoria%TYPE;

ut number;

loc number;

vac number;

armaz number;

Excecao\_utente Exception;

Excecao\_Local Exception;

EXCECAO\_VACINA Exception;

Excecao\_LocalArmazenamento EXCEPTION;

BEGIN

Select count(utente.id\_utente) into ut

from utente

where utente.ID\_UTENTE=vIdUtente;

if ut=0 then

Raise Excecao\_utente;

end if;

Select count(local.LOCALVACINACAO) into loc

from local,utente,convocatoria

where utente.ID\_UTENTE=vIdUtente

and utente.ID\_UTENTE=convocatoria.ID\_UTENTE

and convocatoria.LOCALVACINACAO=local.LOCALVACINACAO;

if loc=0 then

Raise Excecao\_Local;

end if;

Select count(vacina.id\_vacina) into vac

from utente,toma,seringas,frasco,lotes,vacina

where utente.Id\_Utente=vIdUtente

and utente.ID\_UTENTE=toma.ID\_UTENTE

and toma.ID\_TOMA=seringas.ID\_TOMA

and seringas.ID\_FRASCO=frasco.ID\_FRASCO

and frasco.ID\_LOTE=lotes.ID\_LOTE

and lotes.ID\_VACINA=vacina.ID\_VACINA

;

if vac=0 then

Raise Excecao\_vacina;

end if;

Select count(armazem.id\_armazem) into armaz

from utente,toma,seringas,frasco,lotes,armazem

where utente.Id\_Utente=vIdUtente

and utente.ID\_UTENTE=toma.ID\_UTENTE

and toma.ID\_TOMA=seringas.ID\_TOMA

and seringas.ID\_FRASCO=frasco.ID\_FRASCO

and frasco.ID\_LOTE=lotes.ID\_LOTE

and lotes.ID\_ARMAZEM=armazem.ID\_ARMAZEM;

if armaz=0 then

Raise Excecao\_LocalArmazenamento;

end if;

SELECT data\_convocatoria INTO DATAX

FROM UTENTE,CONVOCATORIA

WHERE UTENTE.ID\_UTENTE = CONVOCATORIA.ID\_UTENTE AND CONVOCATORIA.ID\_UTENTE = vIdUtente ;

RETURN DATAX;

EXCEPTION

When EXCECAO\_utente THEN

RAISE\_APPLICATION\_ERROR(-20501, 'Utente Inexistente');

WHEN EXCECAO\_Local THEN

RAISE\_APPLICATION\_ERROR(-20502, 'Local Inexistente');

When EXCECAO\_VACINA THEN

RAISE\_APPLICATION\_ERROR(-20503, 'Vacina Inexistente');

When EXCECAO\_LocalArmazenamento THEN

RAISE\_APPLICATION\_ERROR(-20508, 'Local de Armazenamento inexistente');

END;

create or replace FUNCTION ck3\_i\_data\_proxima\_dose (vIdUtente number)

RETURN fase.datainicioturno%TYPE

IS

DATAX toma.data\_toma%TYPE;

DATAFINAL toma.data\_toma%TYPE;

tempo number;

n number;

dosestomadas number;

excecao\_utente exception;

excecao\_ndose exception;

BEGIN

Select count(utente.id\_utente) into n

from utente

where ID\_UTENTE=vIdUtente;

if n=0 then

Raise excecao\_utente;

end if;

Select utente.ndosestomadas into dosestomadas

from utente

where ID\_UTENTE=vIdUtente;

if dosestomadas=0 then

Raise excecao\_ndose;

end if;

SELECT DATA\_TOMA INTO DATAX

FROM UTENTE,TOMA

WHERE UTENTE.ID\_UTENTE = TOMA.ID\_UTENTE AND TOMA.ID\_UTENTE = vIdUtente ;

SELECT TEMPO\_INTERVALO INTO tempo

FROM VACINA,TOMA,REACAO

WHERE VACINA.ID\_VACINA = REACAO.id\_vacina AND TOMA.ID\_TOMA = REACAO.ID\_REACAO and TOMA.ID\_UTENTE = vIdUtente;

datafinal:=datax+tempo;

RETURN DATAFINAL;

EXCEPTION

When EXCECAO\_utente THEN

RAISE\_APPLICATION\_ERROR(-20501, 'Utente Inexistente');

WHEN EXCECAO\_ndose THEN

RAISE\_APPLICATION\_ERROR(-20515, 'Utente ainda não tomou nenhuma dose');

END;

# Triggers

create or replace TRIGGER chk3\_k\_tg\_apos\_toma

AFTER INSERT ON toma

FOR EACH ROW

DECLARE

rea number;

LOCALA LOCAL.LOCALVACINACAO%TYPE;

SERINGAA SERINGAS.ID\_SERINGA%TYPE;

BEGIN

select count(id\_reacao) into rea

from reacao,toma

where reacao.ID\_TOMA=:New.id\_toma;

if rea > 0 then

UPDATE frasco --ATUALIZA O ESTADO Dos frascos

SET ESTADO ='inutilzavel';

end if;

END;

create or replace TRIGGER chk3\_l\_tg\_convoca\_proximo

after UPDATE OF REJEITOU ON UTENTE

FOR EACH ROW

DECLARE

IDNCONV UTENTE.ID\_UTENTE%TYPE;

convoc CONVOCATORIA.ID\_CONVOCATORIA%type;

BEGIN

SELECT UTENTE.ID\_UTENTE into IDNCONV

FROM UTENTE

WHERE CONVOCADO=0

and rownum = 1

;

select CONVOCATORIA.ID\_CONVOCATORIA into convoc

from convocatoria

where convocatoria.id\_utente=IDNCONV;

IF :new.REJEITOU=1 THEN

UPDATE CONVOCATORIA

SET id\_utente = IDNCONV

where id\_convocatoria=convoc;

end if;

END;

create or replace TRIGGER CHK3\_Q\_TRIG\_A2018019746

after update of estado\_enf ON enfermeiro

FOR EACH ROW

DECLARE

BEGIN

if :new.estado\_enf = 'não trabalhar' then

UPDATE turno --ATUALIZA O ESTADO DA SERINGA UTILIZADA

SET ativo = 0

WHERE turno.id\_enfermeiro = :new.id\_enfermeiro;

end if;

END;

create or replace TRIGGER CHK3\_Q\_TRIG\_A2019123778

AFTER DELETE ON CONCELHO

FOR EACH ROW

DECLARE

LOCALREMOVIDO local.localvacinacao%TYPE;

BEGIN

SELECT local.localvacinacao INTO LOCALREMOVIDO

FROM CONCELHO,LOCAL

WHERE concelho.id\_concelho = local.id\_concelho AND concelho.id\_concelho=:OLD.ID\_CONCELHO;

DELETE FROM local

WHERE local.localvacinacao = LOCALREMOVIDO;

END;

# parâmetros físicos

tabela frasco

tabela convocatoria

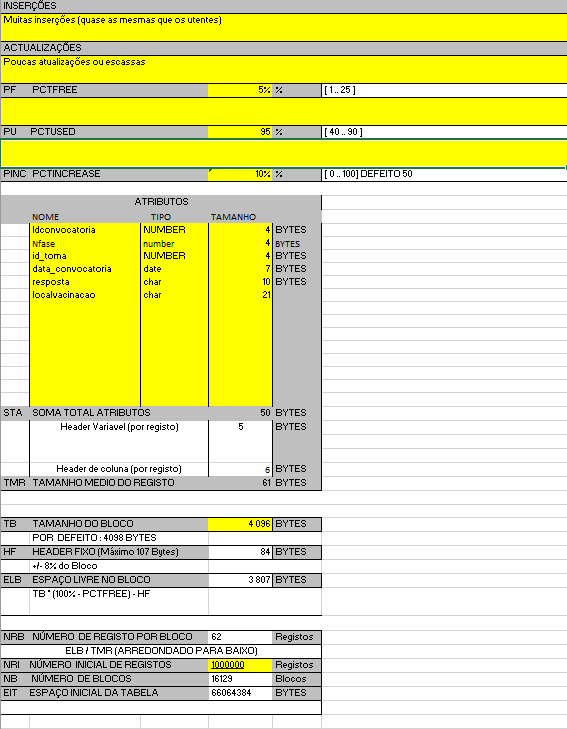


Tabela utente

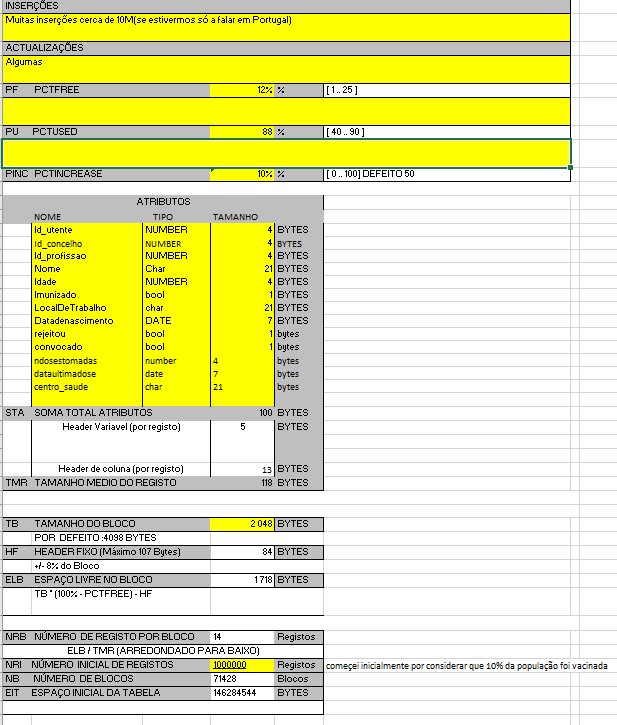


Tabela toma

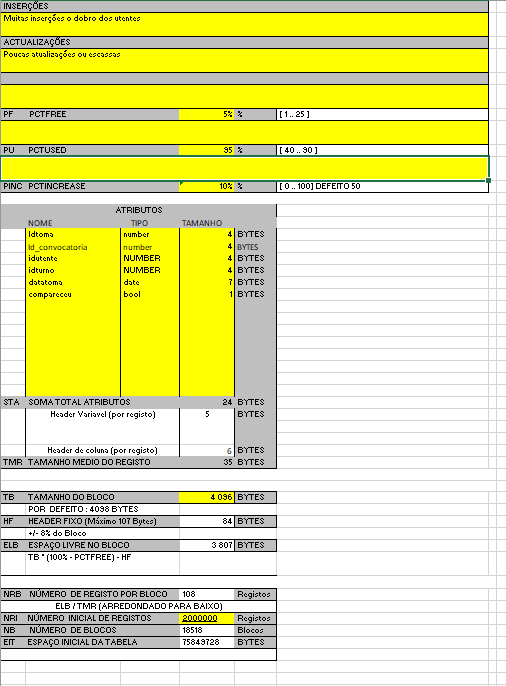


Tabela seringas

