**Fatec Sorocaba**

**Curso de ADS - Disciplina de Laboratório de Banco de Dados**

**Profa Maria Angélica Calixto de Andrade Cardieri**

**Introdução a Programação Procedural no ORACLE**

**Procedimentos Armazenados – Parte 02**

**Março/2021**

1. **Resolução do exercício da aula anterior – Lista01 de Procedures:**
2. Criar o campo pco\_unit number(6,2) na tabela de item pedido.

alter table tb\_item\_pedido add pco\_unit number(6,2);

Escreva uma stored procedure para inserir um item de pedido. Esta deve receber como parametro o número do pedido, codproduto, e a quantidade.

O preço unitario deve ser obtido atraves da tabela produto.

Tratar no-data-found gravando mensagem na tabela tab\_erro

select \* from tb\_item\_pedido order by numpedido;

select \* from tb\_produto;

Solução:

Create or replace procedure Inserir\_item(PnumPed number,Pcodprod number,Pquant number)

as

v\_preco tb\_produto.valor\_unit%type;

begin

select valor\_unit into V\_preco

from Tb\_produto where codProduto = Pcodprod;

insert into tb\_item\_pedido values (Pnumped,Pcodprod,Pquant, v\_preco);

exception

when no\_data\_found then

insert into tab\_erro values (sysdate,' Produto inexistente: ' || Pcodprod);

end;

Realizando testes sobre o procedimento criado:

select \* from tb\_pedido

select \* from tb\_produto

--- Inserindo dados para preparar para os testes

insert into tb\_pedido values (15,'14/06/2020',31,25);

--- testando com produto que existe

exec Inserir\_item(15, 13, 2);

--- testando com produto que não existe – deve executar o

--- exceptions no\_data\_found

exec Inserir\_item(15, 99, 2);

select \* from tb\_item\_pedido

select \* from tab\_erro

1. **Como verificar as procedures criadas** **?**

As procedures criadas podem ser visualizadas usando-se a tabela USER\_OBJECTS’

select object\_name from user\_objects where object\_type = 'PROCEDURE';

OU

Select \* from user\_source;

-- exibir só o texto da procedure

select text from user\_source where name ='INSERIR\_ITEM'

sugestão criar um visão ver\_procedures.

create view ver\_procedures as

select object\_name from user\_objects

where object\_type = 'PROCEDURE';

select \* from ver\_procedures

1. **Uso da variável SQL%Rowcount**

SQL%rowcount => variável usada pelo oracle para indicar o número de linhas do último comando SQL. Útil para delete e update pois este comandos não fazem o fluxo de execução desviar para o “no\_data\_found” qdo devolvem zero linhas.

**Exemplo 1: Usando SQL%rowcount em um bloco anônimo**

set serveroutput ON;

DECLARE

i NUMBER;

BEGIN

UPDATE paciente

SET desconto = 'N'

WHERE datanasc > '01-01-1990';

i := SQL%rowcount;

--note que a atribuição tem que preceder o COMMIT

COMMIT;

dbms\_output.Put\_line(i);

END;

**Exemplo 2: Abaixo temos um procedimento para excluir um cliente cujo código foi passado como parâmetro**.



create or replace procedure excluir\_cliente (pcodcli number)

as

Begin

insert into tab\_erro values (sysdate,'1-Pedido de exclusão do cliente '|| pcodcli);

delete TB\_cliente

where codcliente = pcodcli;

if sql%rowcount = 0 then

insert into tab\_erro

values (sysdate,'2-Cliente a ser excluído não existe' || pcodcli);

Else

insert into tab\_erro values (sysdate,'3- Cliente excluído com sucesso '|| pcodcli);

end if;

commit;

end;

/

**-- testar com cliente que não existe**

exec excluir\_cliente(99)

select \* from tab\_erro;

-- testar com cliente que existe mas não tem pedido

select \* from Tb\_cliente

select \* from tb\_pedido

-- preparando a base de dados para os testes – incluindo com cliente sem pedidos

insert into tb\_cliente values (40,'Maria','Rua x','Sorocaba', '12222-1','SP');

exec excluir\_cliente(40);

select \* from tab\_erro;

-- testar com cliente que exista e tenha pedidos

exec excluir\_cliente(31)

**Observação: A exclusão de um cliente que tenha pedidos não poderá ser realizada. A Integridade Referencial (integridade de chave estrangeira-FK) será violada, pois não é possível excluir um cliente que tenha um pedido apontando para ele. O SGBD irá devolver uma mensagem de erro na tela.**

Para que, em caso de deleção de clientes que tenham pedidos, o erro não se propague na tela, vamos alterar a Procedure anterior, como no exemplo 3.

**Exemplo 3: Procedure excluir\_cliente alterada para tratar a integridade referencial. Excluir um cliente mas antes testar se não existem pedidos para ele.**

- Foram incluindos alguns inserts na tab\_erro apenas para rastrear o fluxo de execução realizado pelo SGBD, para efeito didático. A lógica não exigia isso.

alter table tab\_erro modify mensagem varchar2(100);



Create or replace procedure excluir\_cliente\_FK (pcodcli number)

as

vtotal number;



vcod tb\_cliente.codcliente%type;

Begin

insert into tab\_erro values (sysdate,'1-Pedido de exclusão do cliente '|| pcodcli);



select codcliente into vcod from tb\_cliente where codcliente = pcodcli;

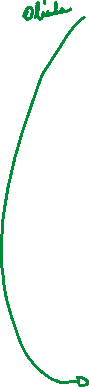


select count(\*) into vtotal from Tb\_pedido where codcliente = pcodcli;



If vtotal > 0 then

insert into tab\_erro values (sysdate,'4- cliente tem pedido não pode ser excluido '|| pcodcli);



else

delete TB\_cliente



where codcliente = pcodcli;



insert into tab\_erro values (sysdate,'3- Cliente excluído com sucesso '|| pcodcli);

end if;

commit;

Exception

when no\_data\_found then

insert into tab\_erro values (sysdate,'2-Cliente a ser excluído não existe' || pcodcli);

end;

-- Testando

exec excluir\_cliente\_FK(31) -- testar com cliente que existe e tem pedidos e verificar na tabela de erros se o resultado é o esperado.(msg = 4)

exec excluir\_cliente\_FK(99) -- testar com cliente que não existe

(msg=2)

select \* from tab\_erro

insert into tb\_cliente values (40,'Maria','Rua x','Sorocaba', '12222-1','SP');

exec excluir\_cliente (40); -- testar com cliente que existe e não tem pedidos (msg =3)

1. **Procedimento raise\_application\_error**

O procedimento RAISE\_APPLICATION\_ERROR é um utilitário que cria números e mensagens de erros que vão de (-20000 a -20999).



-- cancelando a execução da procedure usando o raise



Create procedure .....



Update ..........



delete cliente

where cod\_cliente = 40;

if slq%rowcount(\*) = 0 then

raise\_application\_error (-20001,'Cliente a ser excluido não existe');

end if;



....

.....

=========================================================

Observação: Dependendo da forma de construção do select o SGBD pode ou não considerar no\_data\_found

Exemplo 04:

select count(\*) from tb\_pedido

where codcliente =94;

Como o cliente de código = 94 não existe o rsultado do count será igual a zero. Isto não causará um no\_data\_found.

COUNT(\*)

----------

0

Neste exemplo acima o banco não irá considerar no\_data\_found

**Exemplo 04:**

Já o comando abaixo causará um no-data\_found

select count(\*) from tb\_pedido

O SGBD não retorna linha alguma

where codcliente =4

group by codcliente;

não há linhas selecionadas

**Tarefa: Exercício 1 e 2 da Lista 04B - Procedures**