

PUC-Rio

Data: 27/05/2019

Alunos: João Garcia (1912657)

Wellington Bezerra(1413383)

LABORATÓIO 6

1) Em SA e SB verifique na visão USER_TRIGGERS a existência de gatilhos na tabela de CLIENTE (comando: select trigger_name from user_triggers).

a- drop trigger T_EXCLUICLIENTE. Automaticamente o trigger também foi excluído na outra conexão do banco.

b- CREATE TRIGGER t_excluicliente

BEFORE DELETE ON cliente

FOR EACH ROW

BEGIN

INSERT INTO hcli VALUES (:old.id, :old.nome);

END;

O gatilho foi criado no SB mas dessa vez tive que dar refresh para ele aparecer.

c- drop trigger T_EXCLUICLIENTE

Agora o trigger foi eliminado em SA porém continua em SB

d- Após o rollback SB fica igual à SA. Ao tentar criar o trigger em AS, ele não é criado em SB.

Termine as transações e retorne ao nível de isolamento Read Committed (comando "SET TRANSACTION ISOLATION LEVEL READ COMMITTED;").

2)

(a) insert into cliente values (12,'ANDRÉ',1000,'RUA A N. 1', 'M',TO_DATE('01/01/2001','dd/mm/yyyy'));

Foi inserido em AS, mas não em SB

(b) insert into cliente values (13,'CADU',1000,'RUA A N. 1', 'M',TO_DATE('01/01/2001','dd/mm/yyyy'));

Foi inserido em SB, mas não foi inserido em AS

(c) commit

O commit foi concluído

(d) O André foi inserido na tabela SB após o commit

(e) exec SP_EXCLUICLIENTE(12);

O Cliente foi excluído em AS , mas em SB apenas após o commit

f-

AS Com read commit e SB serializavel

(a) **Com SA em read commit e SB serializavel, após o comando no SA:**

“insert into cliente values (12,'ANDRÉ',1000,'RUA A N. 1', 'M',TO_DATE('01/01/2001',
'dd/mm/yyyy'))”

O usuário foi criado apenas em SA, mas não em SB.

(b) **E após o comando em SB:**

“insert into cliente values (13,'CADU',1000,'RUA A N. 1', 'M',TO_DATE('01/01/2001',
'dd/mm/yyyy'))” Ele foi criado em SB mas não em SA

(c) **Após o SA realizar commit André apareceu em SA e SB não foi atualizado, logo não encontra André na tabela cliente.**

(d) **E a Isadora não aparece na tabela SA, apenas na SB.**

(e) **A exclusão do André foi realizada da tabela SA, mas SB se manteve igual
Se o commit for realizado em SB Isadora aparece em SA.**

SB com read commit e SA serializavel

(a) “insert into cliente values (12,'ANDRÉ',1000,'RUA A N. 1', 'M',TO_DATE('01/01/2001',
'dd/mm/yyyy'))”

André Adicionado. Foi feito o insert no SA, mas não no B

(b) “insert into cliente values (15,'Tomer',1000,'RUA A N. 1', 'M',TO_DATE('01/01/2001',
'dd/mm/yyyy'))”

Tomer foi adicionado em SB, mas não em AS

(c) **Commit bem sucedido**

(d) **Após o commit André apareceu em SB**

(e) exec SP_EXCLUICLIENTE(12);

André removido de AS mas não de SB

Após o commit André sai de SB

3)

(a) **EM SB:**

```
CREATE TABLE poupanca (  
  conta integer,  
  saldo decimal (10,2),  
  id_cliente integer REFERENCES CLIENTE(ID) NOT NULL,  
  CONSTRAINT conta_PK PRIMARY KEY ( conta )  
  ENABLE  
);
```

A tabela foi criada com sucesso

insert into POUPANCA values (3, 3000, 3);

insert into POUPANCA values (3, 3000, 3);

insert into POUPANCA values (3, 3000, 3);

Nada ocorreu os valores não foram adicionados

(b) **Executado em SB:**

EXEC SP_EXCLUICLIENTE(3);

Erro retornado:

BEGIN SP_EXCLUICLIENTE(3); END; Relatório de erros - ORA-02292: restrição de integridade (C##1310966.SYS_C0018361) violada - registro filho localizado ORA-06512: em "C##1310966.SP_EXCLUICLIENTE", line 10 ORA-06512: em line 1 02292. 00000 - "integrity constraint (%s.%s) violated - child record found" *Cause: attempted to delete a parent key value that had a foreign dependency. *Action: delete dependencies first then parent or disable constraint.

(c) **Commit concluído tabela poupança do AS foi atualizada**

4)

Com autocommit ligado

- (a) Foi criado tanto em SA quanto em SB
- (b) Foi tanto em AS quanto em SB
- (c) Não é necessário dar o commit
- (d) As novas entradas já haviam aparecido em ambas as tabelas
- (e) As modificações ocorreram em ambas as tabelas sem a necessidade do comando commit

5)

O read committed permite que o usuário veja alterações em tabelas durante uma única transação, enquanto que o serializavel apenas permite visualizar as atualizações após a transação atual ser terminada. O serializavel executa todos os seus comandos de uma só vez, "em serie" enquanto que o read committed vai alterando aos poucos. Com auto commit não é necessário utilizar o comando commit.