**TDE: Transação - Exercícios**

Alunos:

* Beatriz Perotto Muniz
* Daniel de Almeida Santos Bina
* Eduardo Ferreira de Melo
* Iago Rhudá Ramos

(Grupo 7)

Faça uma **pesquisa**sobre **Transação**para responder às perguntas a seguir - até 0,3 pontos EXTRAS na NOTA FINAL.

1) No contexto de Banco de Dados Relacional, **o que é uma Transação** e por que **razão que ela deve ser utilizada**?

Uma **transação** é um conjunto de operações que são executadas como uma única unidade lógica de trabalho de um SGBD Relacional. Ela pode incluir operações como **inserção, atualização, exclusão e leitura de dados**. O principal objetivo de uma transação é garantir a consistência e a integridade dos dados, mesmo em casos de falhas, garantindo que todas as operações envolvidas sejam **completamente bem-sucedidas ou totalmente revertidas** se algo der errado.

As transações devem ser utilizadas porque garantem a **integridade e consistência dos dados**, permitindo que operações sejam concluídas com sucesso ou revertidas em caso de falha. Elas também asseguram **isolamento** entre transações simultâneas e, uma vez concluídas, garantem a **durabilidade** dos dados, tornando-as essenciais para a confiabilidade de operações críticas e para a **recuperação de falhas** no sistema.

2) Explique cada uma das **propriedades de uma Transação**, que têm a sigla **ACID**.

- **Atomicidade** (Atomicity): A transação é **indivisível**; tudo é realizado ou nada é. Se houver uma falha, todas as alterações feitas pela transação são revertidas (rollback).

- **Consistência** (Consistency): Mantém as regras de integridade, garantindo que os dados permaneçam válidos.

- **Isolamento** (Isolation): Transações simultâneas não interferem entre si, evitando conflitos.

- **Durabilidade** (Durability): Uma vez que a transação é concluída (commit), as alterações são permanentes no banco de dados, mesmo que ocorra uma falha no sistema logo em seguida.

3) Explique como os comando **COMMIT TRANSACTION** e **ROLLBACK**realizam o controle da Transação.

Os comandos **COMMIT TRANSACTION** e **ROLLBACK**são usados para controlar a execução de transações em um banco de dados

- **COMMIT TRANSACTION:** marca o término de uma transação bem-sucedida; o BD está novamente em um estado consistente e as atualizações feitas se tornam permanentes.

- **ROLLBACK**: marca o término de uma transação mal-sucedida; deve ser executado se o BD estiver em um estado inconsistente. Logo, quaisquer operações da transação sevem ser desfeitas.

4) Apresente um **script SQL**para criar uma Transação. Indique:

DROP TABLE IF EXISTS Tab\_Teste;

CREATE TABLE Tab\_Teste (

col1 INT NOT NULL PRIMARY KEY,

col2 INT NOT NULL);

SELECT \* FROM Tab\_Teste;

DROP PROCEDURE IF EXISTS tratErroTransact;

DELIMITER $$

CREATE PROCEDURE tratErroTransact()

BEGIN

DECLARE EXIT HANDLER FOR SQLEXCEPTION

BEGIN

ROLLBACK;

RESIGNAL;

END;

START TRANSACTION;

INSERT Tab\_Teste VALUES (1,111) ;

INSERT Tab\_Teste VALUES (2,222) ;

INSERT Tab\_Teste VALUES (3,'um') ;

INSERT Tab\_Teste VALUES (3,333) ;

COMMIT; -- esse commando executa?

END $$

DELIMITER ;

CALL tratErroTransact();

SELECT \* FROM Tab\_Teste;

- O que o código faz.

O código cria uma tabela Tab\_Teste, adicionando duas colunas (col1 e col2), aceitando valores INT NOT NULL, sendo col1 uma PK. O código trata erros na transação, como podemos ver no terceiro INSERT, é adicionado uma STRING ao invés de um INT:

**“INSERT Tab\_Teste VALUES (3,'um');”**

Com isso, o código não executa e não chega no COMMIT.

- Qual o resultado esperado no caso de execução **BEM SUCEDIDA**?

Se o terceiro INSERT com a string 'um' fosse removido, o restante dos INSERT funcionaria corretamente, e o comando COMMIT seria executado. O resultado final seria a inserção das linhas:

Tabela

Descrição gerada automaticamente

- Qual o resultado esperado no caso de execução com **FALHA**?

A inserção da string ('um') em uma coluna que espera um inteiro gera um erro de execução. Com isso, o handler de erro é ativado, executando um ROLLBACK que reverte todas as operações da transação. Assim, nenhuma das inserções anteriores é aplicada, resultando em uma tabela Tab\_Teste vazia.

Portanto, a execução gera uma mensagem informando que houve uma falha na transação.

