

Aluno: Guilherme Bandeira Rondon

Matrícula: 116031052

Questão 1

Uma transação é um conjunto de comandos a serem executados em um SGBD que alteram o estado do Banco de Dados. Uma transação deve ser atômica, consistente, isolada e durável, além disso, deve fornecer unidades de trabalho confiáveis que mantenham a consistência do banco mesmo em casos de falha do sistema.

Questão 2

Uma transação deve ser:

- Atômica: Todas as operações que compõem a unidade de trabalho devem ser executadas com sucesso para o resultado ser refletido no banco(commit). Caso durante a execução alguma etapa falhar, a transação deve ser desfeita(rollback).
- Consistente: Todas as regras e restrições definidas no banco de dados devem ser obedecidas, como relacionamentos por FK, validação de campos e etc. Para que uma transação seja completa.
- Isolada: Cada transação não pode depender de outra para ser concluída, ou seja, enquanto ela ainda não for concluída, o estado do banco deve ser mantido para evitar que outras transações alterem os dados que ainda estão em processamento.
- Durável: Uma transação após ser concluída tem seus dados permanente, e isso só pode ser desfeito com a execução de outra transação.

Questão 3

Um plano de execução é a ordem cronológica das operações executadas(read, write, commit, abort) de N transações. O isolamento é fundamental para garantir a consistência do BD. Operações críticas devem ser gerenciadas por protocolos baseados em bloqueio, impedindo que múltiplas transações acessem itens concorrentes.

Questão 4

T1	T2
read_lock(Y)	
read_item(Y)	
write_lock(X)	
	read_lock(X)
unlock(Y)	
read_item(X)	
write_item(X)	
unlock(X)	
	read_item(X)

	write_lock(Y)
	unlock(X)
	read_item(Y)
	write_item(Y)
	unlock(Y)

Questão 5

A técnica de ordenação por rótulo de tempo consiste em utilizar a ordem de chegada das transações para gerenciar as operações no schedule. Para cada transação executada, é associado um timestamp ao mesmo, ou seja, no início da transação o sistema fornecerá um rótulo de tempo exclusivo para identificar a transação. Assim, transações que acessem algum campo no BD têm seu rótulo de tempo comparado ao rótulo marcado no campo destacado.

Questão 6

```
CREATE OR replace FUNCTION validacao_concerto() returns TRIGGER
AS
$$
DECLARE
BEGIN
    perform artista, arena, inicio, fim FROM concerto WHERE (
        NEW.artista = artista
        OR
        NEW.arena = arena
    )
    AND
    (
        NEW.inicio BETWEEN inicio AND fim
        OR
        NEW.fim BETWEEN inicio AND fim
    )
    ;
    IF FOUND THEN
        RAISE EXCEPTION 'Local ou Artista indisponível na data
requisitada';
    ELSE
        RETURN NEW;
    END IF;
END;
$$ LANGUAGE plpgsql;
CREATE TRIGGER valida_concerto BEFORE
INSERT
```

```

OR
UPDATE
ON concerto FOR EACH ROW EXECUTE FUNCTION validacao_concerto();

CREATE OR replace FUNCTION validacao_atividade() returns TRIGGER
AS
$$
DECLARE
    atividade_artista INTEGER;
BEGIN
    SELECT      count(nome)
    INTO        atividade_artista
    FROM        artista
    inner join  atividade
    ON          artista.atividade = atividade.id
    WHERE       atividade.id = OLD.atividade;

    IF atividade_artista > 1 THEN
        IF (tg_op = 'DELETE') THEN
            RETURN OLD;
        ELSE
            RETURN NEW;
        END IF;
    ELSE
        RAISE EXCEPTION 'Removendo último artista da atividade %',
OLD.atividade;
    END IF;
END;
$$ LANGUAGE plpgsql;
CREATE TRIGGER valida_artista BEFORE
DELETE
OR
UPDATE
ON artista FOR EACH ROW EXECUTE FUNCTION validacao_atividade();

```