INSTITUTO FEDERAL
Paraíba
Campus Guarabira

Curso:	**Tecnologia em Sistemas Para Internet**
Disciplina:	**Banco de Dados II**
Professor:	**José Barros**
Data:	**05/06/2025**
Nome:	

Atividade – Transações

1. O que é uma transação?

2. Qual é o significado da sigla ACID?

3. Explique o funcionamento dos seguintes comandos: begin, savepoint, commit e rollback.

4. Quais são os problemas de isolamento que podem acontecer quando existem transações concorrentes?

5. Quais são os níveis de isolamento que podem ser utilizados em uma transação? Explique cada um deles.

6. Como configurar o nível de isolamento de uma transação? Dê exemplo.

Para as questões a seguir, pesquise um modelo relacional no Google, crie no mínimo três tabelas do modelo selecionado e insira ao menos três registros em cada uma das tabelas.

7. Neste exercício, são necessárias duas conexões na base de dados, chamadas aqui de SESSÃO 1 e SESSÃO 2.

Execute as instruções de i. a x. uma vez usando isolamento READ COMMITTED, e outra vez usando isolamento REPEATABLE READ. Para cada caso, analise e explique o que acontece em cada caso respondendo às perguntas propostas nos itens vi, viii e x.

i. Abra uma conexão para SESSÃO 1 (janela1);

ii. Abra outra conexão para SESSÃO 2 (janela2);

iii. Na SESSÃO 1, inicie uma transação com um dos níveis de isolamento (**OBS:** inicie a transação executando o comando **BEGIN** seguido de um comando **SET TRANSACTION ISOLATION LEVEL** – execute os comandos nesta ordem, do contrário o PostgreSQL irá se comportar de maneira imprevisível);

iv. Na SESSÃO 1, faça uma consulta, vamos chamar esta consulta de *consulta1*

(**OBS:** cuidado para não executar novamente o comando SET TRANSACTION – execute apenas o comando de consulta);

v. Na SESSÃO 2, inicie uma transação com um dos níveis de isolamento e faça o **update** de uma linha retornada na *consulta1*, alterando assim os dados da resposta da *consulta1*;

vi. Repita o passo iv – o que aconteceu e por quê?

vii. Na SESSÃO 2, execute **commit** para efetivar a transação;

viii. Repita o passo iv – o que aconteceu e por quê?

ix. Na SESSÃO 1, execute **commit** para efetivar a transação;

x. Repita o passo iv – o que aconteceu e por quê?

8. Para este exercício:

a) Usando duas sessões (de maneira semelhante ao exercício 1) e duas transações com nível de isolamento READ COMMITTED, use uma tabela para exemplificar a anomalia *Phantom Read*. Realize uma consulta na sessão 1; na sessão 2, insira/remova/altere tuplas na tabela selecionada e confirme a transação; na sessão 1, repita a mesma consulta para demonstrar o *Phantom Read*;

b) Refaça o experimento usando o nível de isolamento adequado para que não ocorra a anomalia *Phantom Read.*

9. Usando duas sessões cada uma com sua transação:

a)

i. Inicie as duas transações com nível de isolamento READ COMMITTED;
ii. Defina uma tupla para ser atualizada com o comando UPDATE e execute o comando em ambas as transações;
iii. O que acontece na transação 2? Porquê?
iv. Faça commit na transação 1; agora o que aconteceu na transação 2?

b) Repita o exercício a) usando nível de isolamento SERIALIZABLE.

i. Inicie as duas transações com nível de isolamento SERIALIZABLE;
ii. Defina uma tupla para ser atualizada com o comando UPDATE e execute o comando em ambas as transações;
iii. O que acontece na transação 2?
iv. Faça commit na transação 1; agora o que aconteceu na transação 2?

c) Repita o exercício b) usando nível de isolamento SERIALIZABLE, mas executando rollback no passo iv.

i. Inicie as duas transações com nível de isolamento SERIALIZABLE;
ii. Defina uma tupla para ser atualizada com o comando UPDATE e execute o comando em ambas as transações;
iii. O que acontece na transação 2?
iv. Faça rollback na transação 1; agora o que aconteceu na transação 2?