

Transações, Concorrência e Recuperação em SGBDs

1. Escreva um exemplo SQL de transação que transfira R\$ 200 da conta A para a conta B. Mostre como ficariam os comandos BEGIN, COMMIT e ROLLBACK.
2. Dado o cenário de transferência entre contas, explique o que aconteceria se a transação falhar após o UPDATE da conta A, mas antes do UPDATE da conta B.
3. Simule em SQL duas transações concorrentes que tentam atualizar o estoque de um mesmo produto. Mostre por que o requisito de isolamento é importante.
4. Explique como garantir a atomicidade no caso de falha de energia durante uma transação.
5. Classifique a seguinte situação: 'Um valor é debitado de uma conta, mas o crédito correspondente não acontece'. Isso é falha de atomicidade ou consistência?
6. Monte um schedule serial e um schedule concorrente equivalente para duas transações: T1: sacar R\$ 100 da conta A. T2: depositar R\$ 50 na conta B.
7. Dado o seguinte schedule, determine se é serializável por conflito: T1: read(A), write(A) | T2: read(A), write(A).
8. Explique por que o problema da leitura suja (dirty read) pode gerar inconsistências. Dê um exemplo SQL.
9. Simule em SQL uma situação de leitura não-repetível, usando duas transações.
10. Explique como ocorre o fenômeno phantom (tupla fantasma) e construa um exemplo usando INSERT em uma tabela de funcionários.
11. Crie um exemplo em SQL que mostre a necessidade de usar um bloqueio compartilhado (S) e um bloqueio exclusivo (X).
12. Explique como o protocolo 2PL (Two-Phase Locking) funciona e simule uma execução curta com duas transações que usam locks.
13. Mostre uma situação em que ocorre deadlock entre duas transações e como o SGBD poderia resolver com rollback.
14. Crie um exemplo que ilustre o problema da inanição (starvation) em bloqueios.
15. Mostre como aplicar a cláusula SQL SET TRANSACTION ISOLATION LEVEL SERIALIZABLE em um exemplo de atualização de dados.
16. Explique como os registros de log UNDO e REDO ajudam a restaurar o banco de dados após uma falha.
17. Simule em SQL uma transação com UPDATE e COMMIT, e mostre o que o sistema deveria fazer se houver queda logo após o UPDATE mas antes do COMMIT.
18. Explique a diferença prática entre os esquemas de modificação adiada e modificação imediata em logs de recuperação.

19. Mostre como funcionaria a recuperação via shadow page em um exemplo de inserção de registro.

20. Explique como o algoritmo ARIES melhora o tempo de recuperação em comparação a métodos mais antigos.