



DATABASE PROGRAMMING

Instructions

- Complete all tasks below.
- **TOPIC:** TRANSACTIONS AND CONCURRENCY CONTROL

QUESTION

1. Basic Transaction

- Write a SQL statement that demonstrates a simple transaction to transfer funds between two bank accounts.

2. Start and Commit a Transaction

- Write a SQL block that begins a transaction, updates a record, and commits the transaction.

3. Rollback a Transaction

- Write a SQL statement to demonstrate how to rollback a transaction if an error occurs during an update operation.

4. Transaction Isolation Levels

- Explain the different transaction isolation levels (e.g., READ UNCOMMITTED, READ COMMITTED, REPEATABLE READ, SERIALIZABLE) and write an example of SQL setting a transaction's isolation level.

5. Using Savepoints

- Write a SQL statement that creates a savepoint during a transaction, allowing the ability to roll back to that point.

6. Handling Deadlocks

- Explain what a deadlock is and describe how to prevent and resolve deadlocks in SQL. Provide an example query scenario that may lead to a deadlock.

7. Concurrency Control Mechanisms

- Discuss optimistic vs. pessimistic concurrency control. Provide SQL scenarios for both approaches.

8. Using Stored Procedures in Transactions

- Write a stored procedure that performs a series of updates within a transaction, ensuring all updates are completed successfully or rolled back if any part fails.

9. Transaction Log Management

- Explain the role of the transaction log in a database system and write a SQL query to view transaction log information.

10. Database Locking

- Describe the different types of locks (e.g., shared, exclusive) and how they affect database transactions. Provide a SQL example of a situation where locks may be applied.

11. ACID Properties

- Explain the ACID properties of transactions (Atomicity, Consistency, Isolation, Durability) and give an SQL example demonstrating each property.

12. Performance Considerations

- Discuss how using transactions may impact performance and what strategies can be employed to minimize this impact.

13. Test for Transaction Success

- Write a SQL query that checks whether a transaction was successful by checking the resulting data state after a commit.

14. Isolation Level Effects

- Write two queries that demonstrate how different isolation levels affect the visibility of data between transactions.

15. Concurrent Transactions Scenario

- Describe a scenario involving multiple concurrent transactions that could lead to data inconsistency and how you would design the database to handle it.

END