Assignment-6

Name: Boyapati Durgamallikarjuna
RPS UserID: 21951

Assignment 6: Draft a brief report on the use of transaction logs for data recovery and create a hypothetical scenario where a transaction log is instrumental in data recovery after an unexpected shutdown.

Report on the Use of Transaction Logs for Data Recovery:

1. Overview of Transaction Logs

Transaction logs are a crucial component of modern database systems, used to ensure data integrity and facilitate recovery after unexpected events such as system crashes or power failures. A transaction log is a file that records all changes made to a database, including INSERT, UPDATE, DELETE, and transaction-related operations like BEGIN, COMMIT, and ROLLBACK.

2. Purpose of Transaction Logs

- **Data Recovery**: In the event of a failure, transaction logs can be replayed to restore the database to its last consistent state.
- **Point-in-Time Recovery**: They enable recovery of the database to a specific moment by applying committed transactions and ignoring incomplete ones.
- **Concurrency Control**: Logs track ongoing transactions, ensuring ACID (Atomicity, Consistency, Isolation, Durability) compliance.
- **Audit Trail**: Logs serve as a record of database changes for auditing and debugging purposes.

3. Contents of a Transaction Log

- **Transaction ID**: A unique identifier for each transaction.
- **Timestamp**: The time of each operation.
- **Operation Type**: Actions such as INSERT, UPDATE, or DELETE.
- **Pre-Image and Post-Image**: Data states before and after the change (optional, depending on configuration).
- **Commit/Rollback Markers**: Indicate whether a transaction was successfully committed or rolled back.

Hypothetical Scenario: Transaction Logs for Data Recovery

Scenario:
A retail company operates an e-commerce website, storing orders in a database. During a peak sale, the database server suffers an unexpected power outage, leading to data corruption. Customers complain about missing orders.

Steps for Data Recovery Using Transaction Logs:

1. **Identifying the Last Consistent State**:
    - Inspect the transaction log to find the point at which the database was consistent before the crash.
2. **Rolling Forward Committed Transactions**:
    - Apply all changes from transactions marked as committed in the log. For example:
        - INSERT INTO orders VALUES (101, '2025-01-25', 100.00);
        - UPDATE orders SET order_amount = 120.00 WHERE order_id = 101;
3. **Ignoring Incomplete Transactions**:
    - Discard transactions that were not marked as committed. For example:
        - A partially completed transaction:
            - BEGIN;
            - INSERT INTO orders VALUES (102, '2025-01-26', 200.00);
            - (No COMMIT recorded, so it is skipped.)
4. **Restoring Database Consistency**:
    - After applying committed transactions, the database is restored to its last consistent state.

Outcome:
The database is successfully restored with minimal data loss. Customers' orders placed before the outage are recovered, and the business resumes normal operations.