

Name – Vikas Srivastava

Topic – SQL Assignment : 13

Batch - DATACOM+5G Dev

Batch Id - 25SUB4505

User Id – 55984

Email Id – vikas200027@gmail.com

Assignment 13 : 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.

Solution : Transaction logs are critical components in database management systems (DBMS) for ensuring **data consistency**, **reliability**, and **recoverability**. A transaction log, often referred to as a **write-ahead log (WAL)**, records every transaction that occurs within the database. These logs keep a sequential record of all changes made to the database, such as insertions, updates, deletions, and schema changes. In case of an unexpected shutdown, power failure, or system crash, transaction logs play a crucial role in recovering data to its last consistent state.

Importance of Transaction Logs :

- **Data Recovery:**
Transaction logs allow for the **recovery of data** after a system failure by replaying or rolling back transactions.
- **Atomicity and Durability:**
The **ACID properties** (Atomicity, Consistency, Isolation, Durability) of a transaction rely on the transaction log to maintain **durability**.
- **Point-in-Time Recovery:**
Transaction logs also support **point-in-time recovery**. This allows administrators to restore the database to a specific moment before an issue occurred, such as a human error, corruption, or data loss.
- **Backup and Restoration:**
While database backups are essential, transaction logs enable more granular recovery. By applying transaction logs to a backup, DBAs (Database Administrators) can restore the database to the exact state it was in just before the shutdown, even if the last full backup was taken hours or days ago.

Hypothetical Scenario: Using Transaction Logs for Data Recovery After an Unexpected Shutdown

Imagine a scenario where an e-commerce application is running on a database that stores orders, customer information, and product inventory. The database is regularly backed up, and transaction logs are enabled to track all changes made to the data. It is a busy weekend, and several customers are making purchases, causing numerous order records to be inserted and inventory records to be updated.

At 2:00 AM, the system experiences an unexpected **power failure**, causing an abrupt shutdown of the server. When the system comes back online at 2:15 AM, the database appears corrupted and inconsistent, and some recent order transactions are missing.

Recovery Steps Using Transaction Logs:

1. **Database Recovery Process:**
 - **Automatic Recovery:** The DBMS detects the unexpected shutdown and automatically checks the transaction logs to identify any uncommitted transactions.
2. **Rollback of Incomplete Transactions:**
 - The DBMS finds several transactions in progress that were **not committed** before the shutdown, such as an order placed by a customer at 1:59 AM. These uncommitted changes are **rolled back**, ensuring that the database is not left in an inconsistent state.
3. **Reapply Committed Transactions:**

- The system then identifies the **committed transactions** (such as customer orders processed before the power failure) by reviewing the transaction logs.
 - The DBMS re-applies the changes that were **committed but not written to the disk** before the shutdown. For example, an order for a high-value product placed at 1:45 AM will be restored.
4. **Point-in-Time Recovery:**
- Using transaction logs, the system can restore the database to **2:00 AM**, the exact moment just before the power failure, even if the last backup was taken hours earlier.
 - The DBA can apply the **transaction logs** starting from the most recent backup to restore the database to its last known good state.
5. **Successful Recovery:**
- After recovery, the database is in a consistent state, with all customer orders and inventory updates correctly reflected, and no data loss or corruption has occurred.