# SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

**An Autonomous Institution**

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

COURSE NAME : 23CST207 - DATABASE MANAGEMENT SYSTEMS

II YEAR / IV SEMESTER

Unit 4- Transactions

Topic 3 : Transaction Recovery

# Transaction Recovery

- DBMS is a highly complex system with hundreds of transactions being executed every second.
- The durability and robustness of a DBMS depends on its complex architecture and its underlying hardware and system software.
- If it fails or crashes amid transactions, it is expected that the system would follow some sort of algorithm or techniques to recover lost data.
- (or)
- Database recovery means recovering the data when it get deleted, hacked or damaged accidentally.
- Atomicity is must whether is transaction is over or not it should reflect in the database permanently or it should not effect the database at all.

# Cont..

- What is recovery?
- Recovery is the process of restoring a database to the correct state in the event of a failure.
- It ensures that the database is reliable and remains in consistent state in case of a failure.
- **Database recovery can be classified into two parts**

**1. Rolling Forward** -Applies redo records to the corresponding data blocks.

**2. Rolling Back** - Applies rollback segments to the datafiles. It is stored in transaction tables.

# Cont..

- Failure Classification
  - **Logical errors**: transaction cannot complete due to some internal error condition
  - **System errors**: the database system must terminate an active transaction due to an error condition (e.g., deadlock)
- **System crash**: a power failure or other hardware or software failure causes the system to crash.
  - **Fail-stop assumption**: non-volatile storage contents are assumed to not be corrupted by system crash
    - Database systems have numerous integrity checks to prevent corruption of disk data
- **Disk failure**: a head crash or similar disk failure destroys all or part of disk storage
  - Destruction is assumed to be detectable: disk drives use checksums to detect failures

# Storage Structure

- **Volatile storage**:
  - does not survive system crashes
  - examples: main memory, cache memory
- **Nonvolatile storage**:
  - survives system crashes
  - examples: disk, tape, flash memory,
    non-volatile (battery backed up) RAM
  - but may still fail, losing data
- **Stable storage**:
  - a mythical form of storage that survives all failures
  - approximated by maintaining multiple copies on distinct nonvolatile media
  - See book for more details on how to implement stable storage

# Recovery and Atomicity

- When a DBMS recovers from a crash, it should maintain the following

✓ It should check the states of all the transactions, which were being executed.

✓ A transaction may be in the middle of some operation; the DBMS must ensure the atomicity of the transaction in this case.

✓ It should check whether the transaction can be completed now or it needs to be rolled back.

✓ No transactions would be allowed to leave the DBMS in an inconsistent state.

# Cont..

- There are two types of techniques, which can help a DBMS in recovering as well as maintaining the atomicity of a transaction

✓ Maintaining the logs of each transaction, and writing them onto some stable storage before actually modifying the database.

✓ Maintaining shadow paging, where the changes are done on a volatile memory, and later, the actual database is updated.

## Log-based recovery Or Manual Recovery

- **Log-based recovery Or Manual Recovery**
- Log could be a sequence of records, which maintains the records of actions performed by dealing.
- It’s necessary that the logs area unit written before the particular modification and hold on a stable storage media, that is failsafe.
- Log-based recovery works as follows
- The log file is unbroken on a stable storage media.
- When a transaction enters the system and starts execution, it writes a log regarding it.

# Cont..

- **Example:**
  Assume, a transaction to modify the address of an employee. The following logs are written for this transaction,
- **Log 1:** Transaction is initiated, writes 'START' log.
  **Log: <$T_n$ START>**
  **Log 2:** Transaction modifies the address from 'Pune' to 'Mumbai'.
  **Log: <$T_n$ Address, 'Pune', 'Mumbai'>**
  **Log 3:** Transaction is completed. The log indicates the end of the transaction.
  **Log: <$T_n$ COMMIT>**

# Cont..

- **There are two methods of creating the log files and updating the database,**

  1. Deferred Database Modification
  2. Immediate Database Modification

- **In Deferred Database Modification,** all the logs for the transaction are created and stored into stable storage system

- **In Immediate Database Modification,** after creating each log record, the database is modified for each step of log entry immediately.

# Activity

# THANK YOU