ST.XAVIER’S COLLEGE

**Maitighar, Kathmandu**



**DBMS LAB ASSIGNMENT**

**Submitted By:**

**Narayan Panthi**

013BSCCSIT024

**Submitted To:**

**Er. Sanjay Kumar Yadav** ………..…………….

**Lecturer**

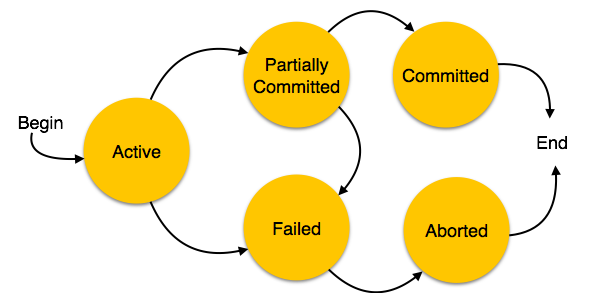
**Department of Computer Science**

**Date of Submission:- *2015 -*** *\_\_****\_-*** *\_\_\_*

## Transaction

* A transaction is the basic logical unit of execution in an information system. A transaction is a sequence of operations that must be executed as a whole, taking a consistent (& correct) database state into another consistent (& correct) database state;
* A collection of actions that make consistent transformations of system states while preserving system consistency
* An indivisible unit of processing
* Atomicity: a transaction is an atomic unit of processing and it is either performed entirely or not at all
* Consistency Preservation: a transaction's correct execution must take the database from one correct state to another
* Isolation/Independence: the updates of a transaction must not be made visible to other transactions until it is committed (solves the temporary update problem)
* Durability (or Permanency): if a transaction changes the database and is committed, the changes must never be lost because of subsequent failure
  + *Serialisability*: transactions are considered serialisable if the effect of running them in an interleaved fashion is equivalent to running them serially in some order

Transaction state:

A transaction in a database can be in one of the following states:

* **Active:**

Active, the initial state; the transaction stays in this state until while it is still executing. A transition is terminated only if it has either been committed or aborted.

* **Partially committed:**

When a transaction executes its final operation, it is said to be in a partially committed state.

* **Failed:**

A transaction is said to be in a failed state if any of the checks made by the database recovery system fails. A failed transaction can no longer proceed further.

* **Aborted:**

If any of the checks fails and the transaction has reached a failed state, then the recovery manager rolls back all its write operations on the database to bring the database back to its original state where it was prior to the execution of the transaction. Transactions in this state are called aborted.

* **Committed:**

If a transaction executes all its operations successfully, it is said to be committed. All its effects are now permanently established on the database system.

## Concurrency Control

* + Most DBMS are multi-user systems.
  + The concurrent execution of many different transactions submitted by various users must be organised such that each transaction does not interfere with another transaction with one another in a way that produces incorrect results.
  + The concurrent execution of transactions must be such that each transaction appears to execute in isolation.
  + Recovery
  + System failures, either hardware or software, must not result in an inconsistent database
* If an error or hardware/software crash occurs between the begin and end, the database will be inconsistent
  + Computer Failure (system crash)
  + A transaction or system error
  + Local errors or exception conditions detected by the transaction
  + Concurrency control enforcement
  + Disk failure
  + Physical problems and catastrophes
* The database is restored to some state from the past so that a correct state—close to the time of failure—can be reconstructed from the past state.
* A DBMS ensures that if a transaction executes some updates and then a failure occurs before the transaction reaches normal termination, then those updates are undone.
* The statements COMMIT and ROLLBACK (or their equivalent) ensure Transaction Atomicity

## System Recovery

System failure can be caused by bugs in the data base, operating system, or hardware. In each case, the Transaction processing is terminated without control of the application. Data in the memory is lost; however, disk storage remains stable. The system must recover in the amount of time it takes to complete all interrupted transactions. At one transaction per second, the system should recover in a few seconds. System failures may occur as often as several times a week.

## Media Recovery

Disk crashes or controller failures can occur because of disk-write bugs in the operating system release, hardware errors in the channel or controller, head crashes, or media degradation. These failures are rare but costly. By identifying the type of DBMS failure, an organization can define the state of activity to return to after recovery. To design the data base recovery procedures, the potential failures must be identified and the reliability of the hardware and software must be determined. the following is a summary of four such recovery actions:

1. **TRANSACTION UNDO:** A transaction that aborts itself or must be aborted by the system during routine execution.
2. **GLOBAL REDO:** When recovering from a system failure, the effects of all incomplete transaction must be rolled back.
3. **PARTIAL UNDO:** While a system is recovering from failure, the results of completed transactions may not yet be reflected in the data base because execution has been Previous erminated in an uncontrolled manner. Therefore, they must be repeated, if necessary, by the recovery component.
4. **GLOBAL UNDO:** If the data base is totally destroyed, a copy of the entire data base must be reloaded from a backup source. A supplemental copy of the transaction is necessary to roll up the state of the data base to the present.

## Recovery

## Mirroring

* + keep two copies of the database and maintain them simultaneously
  + Backup
  + periodically dump the complete state of the database to some form of tertiary storage

## System Logging

* + the log keeps track of all transaction operations affecting the values of database items. The log is kept on disk so that it is not affected by failures except for disk and catastrophic failures.

## Recovery Manager

* Processes Commit, Abort and Restart
* Commit(T)
  + Write T’s updated pages to stable storage atomically*,* even if the system crashes.
* Abort(T)
  + Undo the effects of T’s writes
* Restart = recover from system failure
  + Abort all transactions that were not committed at the time of the previous failure
  + Fix stable storage so it includes all committed writes and no uncommitted ones (so it can be read by new txns)

## Two phase commit

A special object, known as a coordinator, is required in a distributed transaction. As its name implies, the coordinator arranges activities and synchronization between distributed servers. The two-phase commit is implemented as follows:

**Phase 1** - Each server that needs to commit data writes its data records to the log. If a server is unsuccessful, it responds with a failure message. If successful, the server replies with an OK message.

**Phase 2** - This phase begins after all participants respond OK. Then, the coordinator sends a signal to each server with commit instructions. After committing, each writes the commit as part of its log record for reference and sends the coordinator a message that its commit has been successfully implemented. If a server fails, the coordinator sends instructions to all servers to roll back the transaction. After the servers roll back, each sends feedback that this has been completed.

* Two phase commit (2PC) is the standard protocol for making commit and abort atomic
* Coordinator - the component that coordinates commitment at home(T)
* Participant - a resource manager accessed by T
* A participant P is ready to commit T if all of T’s after-images at P are in stable storage
* The coordinator must not commit T until all participants are ready
  + If P isn’t ready, T commits, and P fails, then P can’t commit when it recovers.

### 

### SQL BASIC FACILITIES:

## Table Facility

First and foremost, SQL provides a table facility that enables a prompted, intuitive interface for the

following functions:

* Defining databases
* Populating databases with rows
* Manipulating databases.

## Table Editor

SQL also provides a table editor that makes it easy for you to perform the following functions against

rows in table data that is structured in row and column format:.

* Access
* Insert
* Update
* Delete

## Query Facility:

With the Query facility, SQL permits you to interactively define queries and have results displayed in

a variety of report formats including the following:

* Tabular
* Matrix
* Free format

A relational language can be based on either the relational algebra or the calculus.

Regrettably, SQL is partly both and partly neither. For instance: ORDER BY

statement not based on relational algebra and calculus.

Some SQL constructs are ’algebra-like’. For instance, operations UNION, INTERSECT.

Some SQL constructs are ’calculus-like’.

For instance: SELECT