**ST.XAVIER,S COLLEGE**

**Maitighar, Kathmandu**



DataBase Management System Assignment #6

**.**

Yub Raj Basnet

013BScCSIT048 (4th Semester)

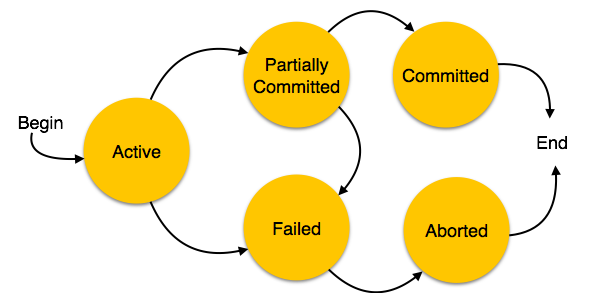
**Submitted to**

|  |  |
| --- | --- |
| Er. Sanjya Kumar Yadav  (Lecturer, St.Xavier’s College ) |  |

## 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.

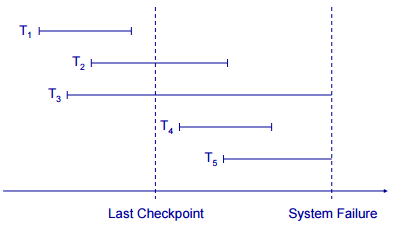
* **Committed:**

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

1. **SYSTEM RECOVERY**

****

* Any transaction that was running at the time of failure needs to be undone and restarted .
* Any transactions that committed since the last checkpoint need to be redone .
* Transactions of type T1 need no recovery .
* Transactions of type T3 or T5 need to be undone and restarted Transactions of type T2 or T4 need to be redone

**MEDIA RECOVERY**

* Disk failure can corrupt the persistent database
* The database must be restored from backup
* The transaction logs can be used to roll forward from the backup point, to recover as much of the recent transaction history as possible
* Restore the database from the last backup.
* Use the transaction log to redo any changes made since the last backup
* Store the log on a separate physical device to the database
* The risk of losing both is then reduced

**TWO-PHASE COMMIT**

Required for distributed or heterogeneous environments, so that correctness is maintained in case of failure during a multi-part COMMIT

Prepare phase has all local resource managers force logs to a persistent log, local managers reply ok or not

Commit phase – if all replies are ok, the coordinator commits, and orders the local managers to complete the process; otherwise all are ordered to ROLLBACK

**SQL FACILITIES**

SQL is rich in the type of ease of use capabilitiesthat are necessary to support relational databases from the simple to the complex.

Table Facility

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

For those readers who have a System i5 background, you will notice that SQL brings with it its own naming scheme that is significantly different from corresponding native objects.

