MIS 301 Relational Database Management System

DATABASE MANAGEMENT SYSTEM

System Implementation Techniques: Query Processing & Optimization , Transaction Processing Concepts

Lecture 24 & 25

QUERY PROCESSING IN DBMS

- ❖Query Processing is the activity performed in extracting data from the database.
- ❖Query processing involves four steps namely
  1. Parsing and translation
  2. Optimization
  3. Evaluation
  4. Execution

Parsing and Translation

- ❖Before processing a query, a computer system needs to translate the query into a human-readable and understandable language.
- ❖SQL or Structured Query Language is the best for human interpretation but not for internal representation of a query.
- ❖Relational algebra is well suited for the internal representation of a query.
- ❖Parsing means examining the characters input and recognizing it as a command or statement by looking through the characters for keywords and identifiers, ignoring comments, arranging quoted portions as string constants, and matching the overall structure to the language syntax making sense of it all.
- ❖The parser translates the query to relational algebra.

Query Optimization

- ❖ There may be many different ways to execute a query.
- ❖ The goal of the **query optimizer** is to find a *reasonably efficient* strategy for executing the query
- ❖ Optimization may be *Heuristic Optimization* or *Cost Based Optimization*
- ❖ In **Heuristic Optimization**, the query execution is refined based on *heuristic rules* for reordering the individual operations.
- ❖ With **Cost Based Optimization**, the overall cost of executing the query is systematically reduced by estimating the costs of executing several different execution plans.
- ❖ Once the query optimizer has determined the execution plan (the specific ordering of access routines), the code generator writes out the actual access routines to be executed.
- ❖ The query code is interpreted and passed directly to the runtime database processor for execution.
- ❖ It is also possible to *compile* the access routines and store them for later execution.

QUERY EVALUATION

- ❖After translating the user query to the relational algebra form, the system executes a query evaluation plan.
- ❖In order to fully evaluate a query, the system needs to construct a query evaluation plan.
- ❖The query evaluation plan is also referred to as **the query execution plan**.
- ❖A **query execution engine** is responsible for generating the output of the given query. It takes the query execution plan, executes it, and finally makes the output for the user query.

QUERY EXECUTION

- ❖The runtime database processor executes the access routines against the database.
- ❖The results are returned to the application that made the query in the first place.
- ❖Any runtime errors are also returned.

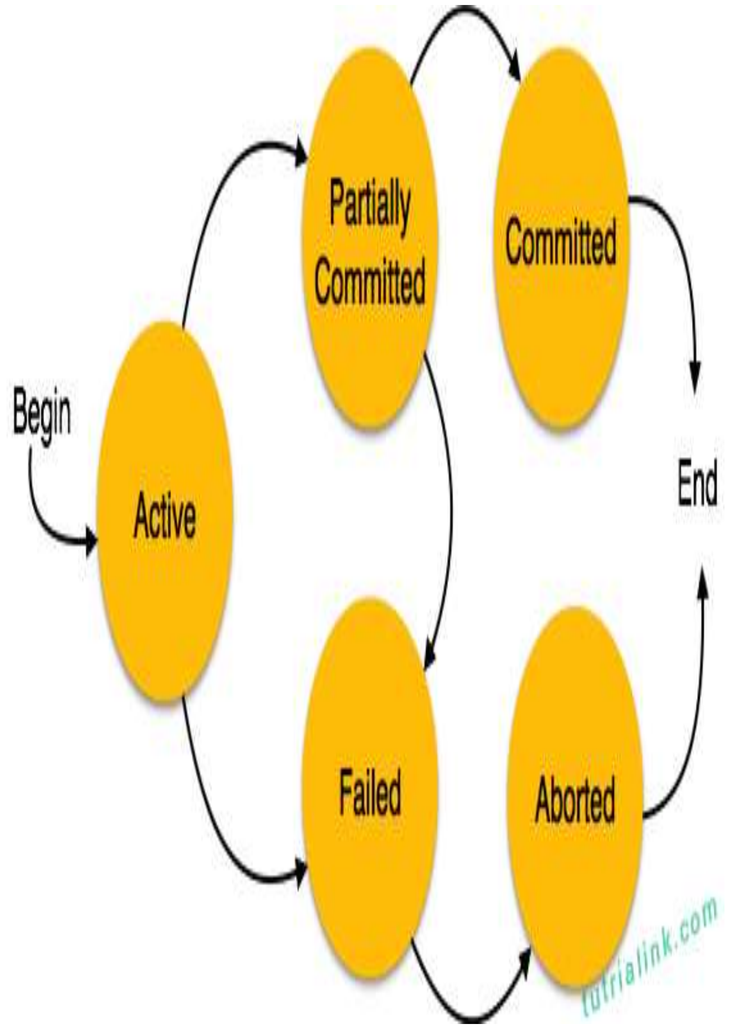
TRANSACTION

- ❖A transaction is a program including a collection of database operations, executed as a logical unit of data processing.
- ❖It is an atomic process that is either performed to completion entirely or is not performed at all.
- ❖Each high level operation can be divided into a number of low level tasks or operations.
- ❖Transaction Operations include low level operations like **begin_transaction(** A marker that specifies start of transaction execution), **read_item or write_item(** Database operations that may be interleaved with main memory operations as a part of transaction), **end_transaction(** A marker that specifies end of transaction), **commit(** A signal to specify that the transaction has been successfully completed in its entirety and will not be undone) and **rollback(** A signal to specify that the transaction has been unsuccessful and so all temporary changes in the database are undone).

TRANSACTION STATES

❖A transaction may go through a subset of five states:-

- **Active** − The initial state where the transaction enters is the active state. The transaction remains in this state while it is executing (read, write or other operations).
- **Partially Committed** − The transaction enters this state after the last statement of the transaction has been executed.
- **Committed** − The transaction enters this state after successful completion of the transaction and system checks have issued commit signal.
- **Failed** − The transaction goes from partially committed state or active state to failed state when it is discovered that normal execution can no longer proceed or system checks fail.
- **Aborted** − This is the state after the transaction has been rolled back after failure and the database has been restored to its state that was before the transaction began.



ACID PROPERTIES OF A TRANSACTION

- **Atomicity** − This property states that a transaction is an atomic unit of processing, that is, either it is performed in its entirety or not performed at all. No partial update should exist.
- **Consistency** − A transaction should take the database from one consistent state to another consistent state. It should not adversely affect any data item in the database.
- **Isolation** − A transaction should be executed as if it is the only one in the system. There should not be any interference from the other concurrent transactions that are simultaneously running.
- **Durability** − If a committed transaction brings about a change, that change should be durable in the database and not lost in case of any failure.

TRANSACTION LOG

- ❖To support transaction processing, DBMSs maintain a transaction record of every change made to the database into a log (also called journal).
- ❖Log is a record of all transactions and the corresponding changes to the database.
- ❖The log is written before any updates are made to the database. This is called **write-ahead log strategy.**
- ❖In case of a system failure, the DBMS examines the transaction log for all uncommitted or incomplete transactions and restores (ROLLBACK) the database to its previous state based on the information in the transaction log.

SCHEDULES AND CONFLICTS

- ❖In a system with a number of simultaneous transactions, a **schedule** is the total order of execution of operations.
- ❖There are two types of schedules-serial and parallel.
- ❖In a serial schedule, at any point of time, only one transaction is active, i.e. there are no overlapping of transactions.
- ❖In parallel schedules, more than one transactions are active simultaneously, i.e. the transactions contain operations that overlap at time.
- ❖In a schedule comprising of multiple transactions, a **conflict** occurs when two active transactions perform non-compatible operations.
- ❖Two operations are said to be in conflict, when all of the following three conditions exists simultaneously −
  - The two operations are parts of different transactions.
  - Both the operations access the same data item.
  - At least one of the operations is a write operation, i.e. it tries to modify the data item.

- TILL WE MEET AGAIN IN THE NEXT CLASS……….