Concurrency control

By: R. K MAHTO

RAJENDRA KUMAR MAHTO
DEPARTMENT OF INFORMATION TECHNOLOGY

- **Concurrency control :** Process which is used to access a particular resource of the database is isolated manner.

- **Locks :** (A lock is a mechanism to control concurrent access to a data item)
  - Shared locks
  - Exclusive locks

Issues/Problems related to concurrent transaction

- Dirty read : *(transaction T_1 is updating the table and another transaction T_2 tries to retrieve data from the same table then the transaction may not get the correct date)*

- Non – Repeatable read : *(transaction T_1 is not able to retrieve the same data twice due to the activity performed by transaction T_2)*

- lost update : *(the updates made by various transactions are lost due to unawareness of the activities)*

Two-Phase Locking Protocol

- This is a protocol which ensures conflict-serializable schedules
  - Phase 1: Growing Phase
    - transaction may obtain locks
    - transaction may not release locks
  - Phase 2: Shrinking Phase
    - transaction may release locks
    - transaction may not obtain locks

Timestamp-based Protocols

- This protocol uses either system time or logical counter as a timestamp.
- Every transaction has a timestamp associated with it, and the ordering is determined by the age of the transaction.
- every data item is given the latest read and write-timestamp. This lets the system know when the last ‘read and write’ operation was performed on the data item.

Deadlock

- Multiple transaction working concurrently on the same resource and are not able to finish there task in absence of the right and permission held by the other.

Deadlock Prevention

- Apply lock such that no cyclic wait can occur by ordering or requesting the lock.

- Deadlock recover mechanism *(the roll-back of transaction is perform instead of waiting for the lock)*
  - Wait/die *(transaction is non-primitive nature)*

Recovery from deadlock

- **Selection of victim** (determines which transaction to rollback to break the deadlock)
- **Roll back** (how for transaction should be rolled back)
- **Starvation** (some transaction never completes its designated task after infinite number of times, thus there is starvation)

Recovery after system crash

- **Redo phase :** The system perform the update again and again by scanning the log file to forward direction from previous checkpoint.

- **Undo phase :** The system performs roll back of transaction by scanning the log file in backward direction from the end.