**ST. XAVIER’S COLLEGE**

**(Affiliated to Tribhuvan University)**

**Maitighar, Kathmandu**

****

**DATABASE MANAGEMENT SYSTEM**

**THEORY ASSIGNMENT**

**SUBMITTED BY:**

Sneha Prasai

013BSCCSIT040

**SUBMITTED TO:**

Er. Sanjay K. Yadav

Lecturer

Department of Computer Science

Database Concurrency Control

1. Purpose of Concurrency Control

Process of managing simultaneous execution of transactions in a shared database, to ensure the serializability of transactions, is known as concurrency control. Concurrency Control Protocols are mechanisms to control concurrency of transactions to ensure isolation of transactions.

1-Lost Updates:

This problem occurs when two transactions that access the same database items have their operations interleaved in a way that makes the value of some database item incorrect. Successfully completed update is overridden by another user.

2-The dirty read problem:

Transactions read a value written by a transaction that has been later aborted. This value disappears from the database upon abort, and should not have been read by any transaction ("dirty read"). The reading transactions end with incorrect results.

3-The incorrect summary problem:

While one transaction takes a summary over the values of all the instances of a repeated data-item, a second transaction updates some instances of that data-item. The resulting summary does not reflect a correct result for any (usually needed for correctness) precedence order between the two transactions.

1. Two phase locking

In databases and transaction processing, two-phase locking (2PL) is a concurrency control method that guarantees serializability. It is also the name of the resulting set of database transaction schedules (histories). The protocol utilizes locks, applied by a transaction to data, which may block (interpreted as signals to stop) other transactions from accessing the same data during the transaction's life. By the 2PL protocol locks are applied and removed in two phases:

Expanding phase: locks are acquired and no locks are released.

Shrinking phase: locks are released and no locks are acquired.

Two types of locks are utilized by the basic protocol: Shared and Exclusive locks. Refinements of the basic protocol may utilize more lock types. Using locks that block processes, 2PL may be subject to deadlocks that result from the mutual blocking of two or more transactions.

1. Limitations of CCMs.

Concurrency control is a type of management style where employers or supervisors constantly monitor how employees are working while the work is still in progress. This kind of management makes employees feel like slaves and lowers their morale to work, which lowers production. It also creates a sense of mistrust between the employer and the employees.

1. Time-Stamp-Based Protocols

The most commonly used concurrency protocol is the timestamp based protocol. This protocol uses either system time or logical counter as a timestamp.

Lock-based protocols manage the order between the conflicting pairs among transactions at the time of execution, whereas timestamp-based protocols start working as soon as a transaction is created.

Every transaction has a timestamp associated with it, and the ordering is determined by the age of the transaction. A transaction created at 0002 clock time would be older than all other transactions that come after it. For example, any transaction 'y' entering the system at 0004 is two seconds younger and the priority would be given to the older one.

In addition, 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.

1. Commit Protocols
2. Index Locking
3. Lock Granularity
4. Time Stamp Ordering multi-version concurrency control
5. Deadlock handling- detection and resolution