



SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with ‘A’ Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING-IOT Including CS&BCT

COURSE NAME : 19SB504 DATABASE MANAGEMENT SYSTEMS

III YEAR / V SEMESTER

Unit V- **CONCURRENCY CONTROL AND
RECOVERY SYSTEM**

Topic : **CONCURRENCY CONTROL, LOCKING
PROTOCOLS**



CONCURRENCY CONTROL



CONCURRENCY CONTROL IN DBMS

CONCURRENCY CONTROL AND RECOVERY
SYSTEM/ 19SB504/DATABASE
MANAGEMENT
SYSTEMS/Mr.R.Kamalakkannan/CSE-



CONCURRENCY CONTROL

Concurrent Execution in DBMS

In a multi-user system, multiple users can access and use the same database at one time, which is known as the concurrent execution of the database.

It means that the same database is executed simultaneously on a multi-user system by different users.



CONCURRENCY CONTROL

Problems with Concurrent Execution

The two main operations are **READ** and **WRITE** operations.

So, there is a need to manage these two operations in the concurrent execution of the transactions as if these operations are not performed in an interleaved manner, and the data may become inconsistent.

So, the following problems occur with the Concurrent Execution of the operations:



CONCURRENCY CONTROL



1. Lost Update Problems (W - W Conflict)
2. Dirty Read Problems (W-R Conflict)
3. Unrepeatable Read Problem (W-R Conflict)

Example

Consider the below diagram where two transactions TX and TY, are performed on the same account A where the balance of account A is \$300.



CONCURRENCY CONTROL

Time	T_x	T_y
t_1	READ (A)	—
t_2	$A = A - 50$	—
t_3	—	READ (A)
t_4	—	$A = A + 100$
t_5	—	—
t_6	WRITE (A)	—
t_7	—	WRITE (A)

LOST UPDATE PROBLEM



CONCURRENCY CONTROL

- ✓ At time t1, transaction TX reads the value of account A, i.e., \$300 (only read).
- ✓ At time t2, transaction TX deducts \$50 from account A that becomes \$250 (only deducted and not updated/write).
- ✓ Alternately, at time t3, transaction TY reads the value of account A that will be \$300 only because TX didn't update the value yet.
- ✓ At time t4, transaction TY adds \$100 to account A that becomes \$400 (only added but not updated/write).



CONCURRENCY CONTROL

- ✓ At time t_6 , transaction TX writes the value of account A that will be updated as \$250 only, as TY didn't update the value yet.
- ✓ Similarly, at time t_7 , transaction TY writes the values of account A, so it will write as done at time t_4 that will be \$400. It means the value written by TX is lost, i.e., \$250 is lost.
- ✓ Hence data becomes incorrect, and database sets to inconsistent.



CONCURRENCY CONTROL

Concurrently control

- ✓ Concurrently control is a very important concept of DBMS which **ensures the simultaneous execution** or manipulation of data by several processes or user without resulting in data inconsistency.
- ✓ Concurrency Control **deals with interleaved execution of more than one transaction.**
- ✓ **maintaining the concurrency of the database, we have the concurrency control protocols.**



CONCURRENCY CONTROL

Concurrency Control Protocols

The concurrency control protocols ensure the atomicity, consistency, isolation, durability and serializability of the concurrent execution of the database transactions. Therefore, these protocols are categorized as:

- ✓ **Lock Based Concurrency Control Protocol**
- ✓ **Time Stamp Concurrency Control Protocol**
- ✓ **Validation Based Concurrency Control Protocol**



Lock Based Concurrency Control Protocol

Lock-based concurrency control protocols are commonly used in database management systems (DBMS) to **manage concurrent access to data**.

These protocols use locks to **ensure that transactions can access and modify data** in a **controlled and consistent** manner.

There are two primary types of locks:

- 1. read locks** (shared locks)
- 2. write locks** (exclusive locks).



Read Lock (Shared Lock): When a transaction acquires a read lock on a data item, it indicates that it intends to read the data item. Multiple transactions can acquire read locks on the same data item simultaneously, **allowing for concurrent reading but preventing concurrent writing.**

Example: Library Book Reading

Write Lock (Exclusive Lock): When a transaction acquires a write lock on a data item, it indicates that it intends to modify (write) the data item. Only one transaction can have a write lock on a data item at a time. Write locks are exclusive and block other transactions from acquiring read or write locks on the same data item.

Example: Conference Room Reservation System



Thank you