



**CHANDIGARH
UNIVERSITY**

Discover. Learn. Empower.

**INSTITUTE : UIE
DEPARTMENT : APEX INSTITUTE OF
TECHNOLOGY(CSE) -AIML**

Bachelor of Engineering (Computer Science & Engineering)

Advanced Database Management System

Er. Vishwa Deepak (E12867)

DISCOVER . **LEARN** . EMPOWER

Course Objectives

| CO Number | Course Objective | Level |
|------------------|--|--------------|
| CO1 | Develop understanding the advancement in SQL | Apply |

Course Outcome

| CO Number | Course Outcome | Level |
|-----------|---|-------|
| CO2 | Create views of data and Implement transaction control using locks. | Apply |

LECTURE OUTCOMES

- ❖ Student will learn about the advances in SQL like Locks.
- ❖ Student will learn about types of locks

Revision

A single transaction, performing modifications on accounts A and B:

Modification : Transferring 50\$ balance from A to B

Data Items: bal_A, bal_B (columns/attributes in table)

E.g., transaction to transfer \$50 from account A to account B:

1. **read**(bal_A)
2. $A := A - 50$
3. **write**(bal_A)
4. **read**(bal_B)
5. $B := B + 50$
6. **write**(bal_B)

Introduction to Locks

- ❖ A lock is a mechanism to control concurrent access to a data item
- ❖ Data items can be locked in two modes :
 1. *Exclusive (X) mode*. Data item can be both read as well as written. X-lock is requested using **lock-X** instruction.
 2. *Shared (S) mode*. Data item can only be read. S-lock is requested using **lock-S** instruction.
- ❖ Lock requests are made to the concurrency-control manager by the programmer. Transaction can proceed only after request is granted.

LOCK Compatibility Matrix

❖ Lock-compatibility matrix

| | S | X |
|---|-------|-------|
| S | true | false |
| X | false | false |

- ❖ A transaction may be granted a lock on an item if the requested lock is compatible with locks already held on the item by other transactions
- ❖ Any number of transactions can hold shared locks on an item,
 - ❖ But if any transaction holds an exclusive on the item no other transaction may hold any lock on the item.
- ❖ If a lock cannot be granted, the requesting transaction is made to wait till all incompatible locks held by other transactions have been released. The lock is then granted.

Locking Protocol

T_2 : lock-S(A);
 read (A);
 unlock(A);
 lock-S(B);
 read (B);
 unlock(B);
 display($A+B$)

- ❖ Locking as above is not sufficient to guarantee serializability — if A and B get updated in-between the read of A and B , the displayed sum would be wrong.
- ❖ A **locking protocol** is a set of rules followed by all transactions while **requesting and releasing locks**. Locking protocols restrict the set of possible schedules.

Locking Protocol Rules(2PL)

- ❖ This protocol 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
- ❖ The protocol assures serializability. It can be proved that the transactions can be serialized in the order of their **lock points** (i.e., the point where a transaction acquired its final lock).

EXAMPLE

| | T ₁ | T ₂ |
|----|----------------|----------------|
| 1 | lock-S(A) | |
| 2 | | lock-S(A) |
| 3 | lock-X(B) | |
| 4 | | |
| 5 | Unlock(A) | |
| 6 | | Lock-X(C) |
| 7 | Unlock(B) | |
| 8 | | Unlock(A) |
| 9 | | Unlock(C) |
| 10 | | |

TYPE OF 2PL

Strict 2-PL

This requires that in addition to the lock being 2-Phase **all Exclusive(X) locks** held by the transaction be released until *after* the Transaction Commits.

Following Strict 2-PL ensures that our schedule is:

- Recoverable
- Cascadeless

Rigorous 2-PL –

This requires that in addition to the lock being 2-Phase **all Exclusive(X) and Shared(S) locks** held by the transaction be released until *after* the Transaction Commits. Following Rigorous 2-PL ensures that our schedule is:

- Recoverable
- Cascadeless

Conservative 2-PL –

This protocol requires the transaction to lock all the items it access before the Transaction begins execution by predeclaring its read-set and write-set. If any of the predeclared items needed cannot be locked, the transaction does not lock any of the items, instead, it waits until all the items are available for locking.

Conservative 2-PL is *Deadlock free* and but it does not ensure a Strict schedule

S U M M A R Y

Explained Locks in transaction.

HOME WORK

- What is Growing Phase?
- Explain importance of Locks.

REFERENCES

Text Book:

1. Database Systems Concepts, design and Applications, S. K. Singh

Reference book:

1. An Introduction to Database Systems, C.J. Date
2. Database System Concepts, Korth, Henry

Web References:

1. <https://www.javatpoint.com/dbms-lock-based-protocol>



THANK YOU

Mail at: Vishwa.e12867@cumail.in