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

Locks

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 Shared and Exclusive 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

- 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).

Locks

- Two-Phase Locking Techniques
  - Locking is an operation which secures
    - (a) permission to Read
    - (b) permission to Write a data item for a transaction.
  - Example:
    - Lock (X). Data item X is locked in behalf of the requesting transaction.
  - Unlocking is an operation which removes these permissions from the data item.
  - Example:
    - Unlock (X): Data item X is made available to all other transactions.
  - Lock and Unlock are Atomic operations.

Two-Phase Locking Techniques: Essential components

- **Two locks modes:**
  (a) shared (read) (b) exclusive (write).

- **Shared mode: shared lock (X)**

  - More than one transaction can apply share lock on X for reading its value but no write lock can be applied on X by any other transaction.

- **Exclusive mode: Write lock (X)**
  - Only one write lock on X can exist at any time and no shared lock can be applied by any other transaction on X.

Two-Phase Locking Techniques: Essential components

- **Lock Manager:**
  - Managing locks on data items.
- **Lock table:**
  - Lock manager uses it to store the identify of transaction locking a data item, the data item, lock mode and pointer to the next data item locked. One simple way to implement a lock table is through linked list.

Summary

Explained Locks in transaction.

Home Work

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

References

- **Reference book:** **1.** Database Systems concepts, Korth
  2. Sql&Pl/Sql by Bayross,Ivan
- **Web References:**

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

THANK YOU

Mail at: Vishwa.e12867@cumail.in