Leonard Wassenaar

25948083

ITRW 321

SU 1: Summary

**Transaction Management and Concurrency control**

Table of Contents

[Transactions: 3](#_Toc520307908)

[Properties 3](#_Toc520307909)

[Concurrency Control: 4](#_Toc520307910)

[Concurrency Control Problems**:** 4](#_Toc520307911)

[Concurrency Control with Locking methods: 5](#_Toc520307912)

[Lock granularity: 5](#_Toc520307913)

[Two-Phase Locking: 5](#_Toc520307914)

[Concurrency Control with Time Stamping Methods: 6](#_Toc520307915)

[Wait/die: 6](#_Toc520307916)

[Wound/wait: 6](#_Toc520307917)

[Concurrency Control with Optimistic Methods: 6](#_Toc520307918)

[Database Recovery Management: 7](#_Toc520307919)

[Database Recovery 7](#_Toc520307920)

[Critical events 7](#_Toc520307921)

[Transactional Recovery 7](#_Toc520307922)

**Transaction Management and Concurrency control**

# Transactions:

A transaction involves accessing a database with an arrangement of database requests.

Properties:

Atomicity: Consistency: Isolation: Durability:

Every part of a Database must be Data of a When changes

transaction is in a consistent state. Transaction is made and committed,

treated as can’t be used it can’t be undone or go

single logical until the first missing.

unit one is finished.

# 

# Concurrency Control:

A feature of the database management system coordinating the simultaneous transaction execution in multiuser database systems.

## Concurrency Control Problems**:**

Lost Update Uncomitted Data Inconsistent Retrievals

Transaction accesing A transaction accessing Is a problem where a

data before and after data that is uncomitted transaction accesses

other transactions from a different transaction. data before and after

are finshed working other transactions are

with the data. done working with the

data.

# 

# Concurrency Control with Locking methods:

Lock:

A device that provides exclusive use of data to a current transaction.

## Lock granularity:

|  |  |
| --- | --- |
| Database-level lock | Database access is restricted to the owner of the lock and allows only one user at a time. |
| Table-level lock | Allows one transaction at a time to access the table. |
| Page-level lock | The DBMS lock the entire diskpage. |
| Row-level lock | Enables concurrent transactions to access separate rows of the same table. |
| Field-level lock | Access the same row as long as they have different fields. |

## Two-Phase Locking:

Principles that rules how a transaction acquires and relinquish locks.

Phases:

|  |  |
| --- | --- |
| Growing phase: | Transaction gets locked without unlocking data |
| Shrinking phase: | Releases locks and cannot aquire a new lock |
| Rules:   * Transactions cannot have locks that have conflicts. * Unlock operations cannot precede a lock operation with similar transactions. * Locks must be obtained first before data will be effected. | | |

Deadlock:

More than one transaction wait for others to release the lock from a previously locked item.

# Concurrency Control with Time Stamping Methods:

Methods in time stamping is used to control concurrent transaction execution.

## Wait/die:

Transactions that are older must wait for the newer transaction to finish, then release locks before it can request the lock.

## Wound/wait:

Older transactions can request the lock, then preempt the newer transaction and reschedule it.

# Concurrency Control with Optimistic Methods:

Optimistic approach:

Assuming that most database operations do not battle.

Transaction moves though 2 or 3 Phases:

Read phase

Write phase

Validation phase

|  |  |
| --- | --- |
| Read phase | Database is read, needed computations is executed and updates are made. |
| Validation phase | Validate transactions to see if the changes effects consistency and integrity. |
| Write phase | Permanent changes are applied to the database |

# Database Recovery Management:

Database Recovery:

Involves restoring the database to a consistent previous state.

Critical events that can cause the database to be less functional and compromise data integrity:

* Hardware failures
* Software failures
* Human- caused incidents
* Natural disasters

Transactional Recovery:

4 concepts that influences the recovery process;

* Write-ahead-log protocol
* Redundant transaction logs
* Buffers
* Database checkpoints