Unit - IV (Transaction)

Introduction

- A transaction refers to a sequence of one or more operation Such as (read, write, update or delete) performed on the database as a single logical unit of work.

- Transaction are designed to maintain the integrity, consis-tency, and reliability of database

- The transaction is executed as a single unit.

- DBMS transaction must be atomic, consistent, isolated and durable.

Operation of transaction

1) Read

A read operation is used to read the value of particular database element and stores in the main memory.

2) Write

A write operation is used to write the value of database from the buffer in the main memory.

3) Commit

This operation is used to maintain integrity in database.

4) Rollback

Rollback operation undergoes all changes made by transaction.

Properties of transaction (ACID)

- Transaction follows 4 properties namely, Atomicity, Consistency, isolation and durability these are refered to as ACID properties of transaction in DBMS

A - Atomicity "All or nothing transaction"

C - Consistency "Valid state maintenance"

I - Isolation "Transaction do not effect each other"

D - Durability "Permanent changes after commitment"

1) Atomicity

- This property ensure that either all operation of transaction are executed or it is aborted.

- Atomicity is achieved through commit and rollback operation.

2) Consistency

- This property of transaction keep the database consistent before and after transaction is completed.

- Consistency means the changes made in database are result of logical operation.

3) Isolation

- This property ensure that two transaction must not interfere with each other.

- It ensure that Integrity of database is maintained.

4) Durability

- These property ensure that changes made to the database after transaction is completely executed are durable

- It indicates that permanent changes are made by successful execution of transaction

Concurrency Control

- Concurrency control is a concept in DBMS that ensure multiple transaction can simultaneously access or modify data without causing errors or inconsistencies.

Advantages

- Concurrency leads to less waiting time
- Concurrency leads to less response time
- Concurrency leads to more resource utilization
- Concurrency leads to more efficiency.

Disadvantages
- Overhead
- Deadlock
- Complexity
- Inconsistencies

Techniques

- Concurrency control technique in DBMS help manage simultaneous transaction to ensure data integrity and consistencies.

1) Lock - based protocol
These use locks to regulate access to data
Types -
- Binary lock
- Shared lock
- Two- phase locking

2) Time stamp - Based protocol
Every transaction is assigned a timestamp, older transaction get more priority than newer one.

3) Optimistic concurrency control
Transaction execute without restriction and are checked for conflicts before committing.

4) Validation - based protocol
Transaction go through stages read, validation and write. If validation fails the transaction is aborted.

SQL transaction

- A transaction in SQL is a sequence of one or more SQL Statement executed as a single unit of work.

- SQL transaction are essential for ensuring data integrity and consistency in relational database.

Key properties : ACID
→ Atomicity
→ Consistency
→ Isolation
→ Durability

Types of SQL transaction

→ Read transaction
Used to only read the data, with SELECT queries

→ Write transaction
Involves modifying data in database with INSERT, UPDATE or DELETE operation

→ Distributed transaction
Spans multiple database and ensure consistency across them

→ Implicit transaction
Automatically started by SQL server

Locking techniques

Locking techniques in DBMS are used to ensure data integrity and prevent concurrency issue when multiple transaction access the same data.

Some locking techniques are :-

1> Binary lock
Each data item is either locked (1) or unlocked (0)

2> Shared and exclusive lock

Shared (Read lock) - Multiple transaction can read the data but cannot modify it

Exclusive (write lock) - Only one transaction can read and modify the data

3> Two - phase locking

Growing Phase - Transaction acquire locks but do not release any

Shrinking Phase - Transaction release lock but do not acquire any ones

4> Strict two phase locking
Transaction hold all locks untill they commit or abort

Database Recovery techniques

Database recovery techniques are essential for maintaining data integrity and ensuring that database can recover from failure.

1> Log - Based Recovery

Transaction are recorded in a log, allowing the database to replay operation in case of failure.

- Undo logging - If transaction fails, changes are rolled back using log records.
- Redo logging - Lost updates are restored.

2> Shadow paging

Shadow paging ensure the reliability and consistency of data. It plays a crucial role in maintaining atomicity and durability.

These technique ensure database can recove from failure without losing data integrity.

* Recovery algorithm is same as recovery technique and locking technique

Database Security

Database security in DBMS refers to protecting the integrity, confidentiallity and availability of data stored in database.

→ Authentication and authorization
User must login with valid credentials

→ Backup and Recovery
Regular backup ensure data is not lost due to cyberattacks or system failures.

→ Network security & firewalls
firewall filter malicious traffic

→ Secure database configuration
Strong password prevent unauthorized access

Imp.

Deadlock

A deadlock in DBMS is a situation where two or more transaction are waiting for each other to relase resource creating a cycle where none of them can proceed.

- It happens when multiple transaction hold locks on different resource and request locks on resource held by others

Condition for deadlock

A deadlock occurs when following four condition hold simultaneously :-

1) Mutual exclusion

At least one resource is held in a non-shareable mode

2) Hold and wait

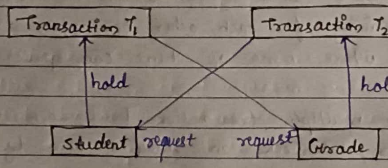
A process is holding atleast one resource and waiting for another resource.

3) No preemption

A transaction cannot forcefully take a resource from another transaction

4) Circular wait

A set of transaction from a cycle, each waiting for a resource held by another transaction in a cycle



Deadlock Avoidance

- when a database is stuck in a deadlock, it is always better to avoid the deadlock rather than restarting or aborting the database.
- The deadlock avoidance method is suitable for smaller database whereas deadlock prevention is suitable for larger database.

Technique

Banker's algorithm

This algorithm ensure that resource allocation keeps the system in a safe state, preventing deadlock.

Timeout mechanism

Transaction are forced to release lock if they exceed a predefined waiting time.

Deadlock Detection

when a transaction waits indefinitely to obtain a lock. the DBMS should detect whether the transaction is involved in a deadlock or not.

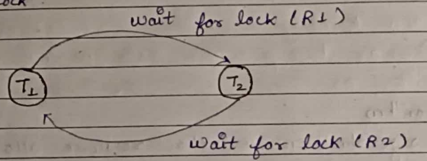
Technique :-

Wait - for - graph

Wait - for - graph is one of the method for detecting the deadlock situation this method is suitable for smaller database

- In this method a graph is drawn based on the trasaction, if the graph created is closed loop or cycle then there is deadlock

Ex:-



Deadlock Prevention or Recovery

- Deadlock prevention method is suitable for larger database. deadlock can be prevented if the resource are allocated in such way that deadlock never occurs.

Deadlock prevention mechanism proposes two Scheme :-

Wait - Die scheme

In this scheme, if transaction request a resource that is locked by another transaction, then DBMS simply checks the timestamp of both transaction and allow the older transaction to wait until the resource is available for execution.

Wound - wait Scheme

In this scheme, if an older transaction request for a resource held by a younger transaction, then an older transaction forces a younger transaction to kill the transaction and release the resource.

Difference b/w wait - die and wound wait

Wait - Die	Wound - Die
• It is based on non - preemptive technique	• It is based on preemptive technique
• older transaction must wait for younger one to release	• older transaction never wait for younger transactio
• No. of abborts and rollback is higher	• No. of aborts and rollback is lesser.

Advantages of Deadlock

→ Mutual exclusion
→ Hold and wait
→ No preemption
→ Circular wait
→ Inconsistent data
→ Indefinite blocking

Disadvantages of deadlock

→ System downtime
→ Resource wait
→ Reduce concurren
→ Complex resolution
→ Increased Systen overhead.