Unit IV

Concurrency Control:

Definition : [Concurrency control → Concurrency में multiple transaction एक साथ execute इस बात से deal करता है कि multiple data का एक-एक execution and manipulate करने के बाद भी कोई inconsistency नहीं आए]

Concurrency control is a very important concept of DBMS which ensures the simultaneous execution or manipulation of data by several processes or users without resulting in data inconsistency. Concurrency control deals with interleaved (linked) execution of more than one transaction.

[What is Transaction?
Concurrency control transaction में use होता है]

A transaction is a collection of operation that performs a single logical function in a database application. Each transaction is a unit of both atomicity and consistency. Thus, we require that transaction do not violate any database consistency. That is, if the database was consistent when a transaction started, the database must be consistent when the transaction successfully terminates. However, during the execution of a transaction, it may be necessary temporarily to allow inconsistency, since either the debit of A or the credit of B must be done before

the other. This temporary inconsistency although necessary, may lead to difficulty if a failure occurs.

For ex:-

The transaction to transfer funds from the account of dept A to the account of dept B could be defined to be composed of two separate programs: one that debits account A, and another that credits account B. The execution of these two programs one after the other will indeed preserve consistency. However, each program by itself does not transform the database from a consistent state to a new consistent state. Thus, those programs are not transaction.

Concurrency can simply be said to be executing multiple transaction at a time. It is required to increase time efficiency. If many transactions try to access the same data, then inconsistency arises. Concurrency control required to maintain consistency data.

For example, if we take ATM machines and do not use concurrency, multiple persons cannot draw money at a time in different places. This is where we need concurrency.

Advantages

The advantages of concurrency control are as follows:

- Waiting time will be decreased
- Response time will decrease
- Resource utilization will increase
- System performance & efficiency is increased.

Concurrency Control Problems

In a database transaction, 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 transaction 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:

Problem 1: Lost Update Problems (W-W conflict)

The problem occurs when two different database transaction perform the read/write operations on the same database items in an interleaved manner that makes the values of the items incorrect hence making the database inconsistent. (different from each other)

For 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.

Time	Tx	Ty
t_1	READ(A)	
t_2	A = A - 50	
t_3		
t_4		READ(A)
t_5		A = A + 100
t_6	WRITE(A)	
t_7		WRITE(A)

LOST UPDATE PROBLEM

- At time t_1, transaction Tx reads the value of account A i.e., $300 (only reads)
- At time t_2, transaction Tx deducts $50 from account A that become $250 (only reduce and not updated/write)
- At time t_3, transaction Ty reads the value of account A that will be $300 only because Tx didn't update the value yet
- At time t_4, transaction Ty adds $100 to account A that becomes $400 (only added but not update/write).
- At time t_5, transaction Tx writes/update the value of account A that will be updated as $250 only, as Ty didn't update the value yet.

- Similarly, at time t7, transaction Ty writes the value of account A, so it will write as done at time t4 that will be $400. It mean the value written by Ty is lost, i.e. $250 is lost.

Hence data become incorrect, and database get to inconsistent.

Problem 2: Dirty Read problems (W-R conflict)

The dirty read problem occurs when one transaction updates an item of the database, and somehow the transaction fail, and before the data gets rollback, the updated database item is accessed by another transaction. There comes the Read-Write Conflict between both transaction.

~~Lost value update~~ ~~at t4~~

For example:

Consider two transactions Tx and Ty in the below diagram performing read/write operation on account A where the available in account A is $300:

Time	Tx	Ty
t1	READ(A)	
t2	A = A+50	-
t3	WRITE(A)	-
t4	-	-
t5	SERVER DOWN ROLLBACK	READ(A)

DIRTY READ PROBLEM

- At time t1, transaction Tx reads the value of account A, i.e., $300.
- At time t2, transaction Tx adds $50 to account A that becomes $350
- At time t3, transaction Tx writes the updated value in account A i.e., $350.
- Then at time t4, transaction Ty reads account A that will be read as $350.
- Then at time t5, transaction Tx rollback due to server problem, and the value changes back to $300 (as initially).
- But the value for account A remains $350 for transaction Ty as committed, which is the dirty read and therefore known as the Dirty Read problem.

Unrepeatable read problem = its occur when in a transaction two different values are read for the same database item.

Problem 3: Incorrect summary problem

The incorrect summary problem occurs when there is an incorrect sum of the two data. This happen when a transaction tries to sum two data using an aggregate function and the value of any one of the data get changed by another transaction.

Example: Consider two transaction Tx and Ty performing read/write operation on two data A and B in the database. The current value of A is 1000 and B is 2000:

The following table shows the read/write operation in Tx and Ty transaction.

Time	Tx	Ty
T_1	READ(A)	
T_2	add = 0	
T_3	add = add + A	
T_4		READ(B)
T_5		B = B + 500
T_6	READ(B)	
T_7	add = add + B	

In the above example, transaction 2 is calculating the sum of some record while transaction 1 is updating them. Therefore the aggregate function may calculate some values before they have been updated and others after they have been updated.

Problem 4: Unrepeatable Read problem:

Also known as Inconsistent Retrieval Problem that occurs when in a transaction, two different values are read for the same database item.

For ex: Consider two transactions, Tx and Ty, performing the read/write operation on account A, having an available balance = $300. The diagram is shown below:

Time	Tx	Ty
t_1	READ(A)	—
t_2	—	READ(A)
t_3	—	A = A + 100
t_4	—	write(A)
t_5	READ(A)	

Problem 5: Phantom Read Problem

The phantom read problem occurs when a transaction reads a variable once but when it tries to read that same variable again, an error occurs saying that the variable does not exist.

Example:

T1	T2
Read(X)	
	Read(X)
Delete(X)	
	Read(X)

In the above example, once transaction 2 reads the variable X, transaction 1 deletes the variable X without transaction 2's knowledge. Thus when transaction 2 tries to read X, it is not able to do it.

Concurrency Control Technique

Concurrency control is provided in a database to:

(i) enforce isolation among transactions.

(ii) preserve database consistency through consistency preserving execution of transactions.

(iii) resolve read-write and write-read conflicts.

Various concurrency control techniques are:

1. Two-phase locking protocol
2. Time stamp ordering protocol
3. Multi version concurrency control
4. Validation concurrency control.

1. Two-Phase locking protocol

Locking is an operation which secures: permission to read, OR permission to write a data item. Two phase locking is a process used to gain ownership of shared resources without creating the possibility of deadlock (2nd Nov). The 3 activities taking place in the two pha

but cannot acquire any.

2. Time stamp ordering protocol :

A time stamp is a tag that can be attached to any transaction or any data item, which denotes a specific time on which the transaction or the data item had been used in any way.

A timestamp can be implemented in 2 ways. One is to directly assign the current value of the clock to the transaction or data item. The other is to attach the value of a logical counter that keeps increment as new timestamp are required.

The timestamp of a data item can be of 2 types:

(i) W-timestamp(X): This means the latest time when the data item X has been written into.

(ii) R-timestamp(X): This means the latest time when the data item X has been read from.

3. Multiversion Concurrency Control:
Multiversion protocol aims to reduce the delay for read operation.
Multiversion schemes keep old versions of data item to increase concurrency. Multiversion 2 phase locking: Each successful write results in the creation of a new version of the data item written. Timestamps are used to label the versions. When a read(x) operation is issued, select an appropriate version of x based on the timestamp of the transaction

- Content - This field contains the data value of that version.
- Write-timestamp - This field contains the timestamp of the transaction that created the new version
- Read-timestamp - This field contains the timestamp of the transaction that will read the newly created value.

How does Multi-version concurrency control (MVCC) in DBMS works?

- In the database, every tuple has a version number. The tuple with the greatest version number can have a read operation done on it simultaneously.
- Only a copy of the record may be used for writing operations.
- While the copy is being updated concurrently, the user may still view the previous version.
- The version number is increased upon successful completion of the writing process.
- The upgraded version is now used for every new record operation and every time there is an update, this cycle is repeated.

Advantages

- The reduced read-and-write necessity for database locks.
- Increased concurrency
- Minimize read operation delays.
- Accuracy and consistency.

Disadvantages

- Garbage collecting
- Increase the size of the database
- Complexity
- Overhead.

4) Validation Based protocol :-

Validation Based protocol is also called Optimistic Concurrency control Technique. This protocol is used in DBMS for avoiding concurrency in transactions. It is called optimistic because of the assumption it makes, i.e. very less interference occur, therefore there is no need for checking while the transaction is executed.

Optimistic Concurrency control is a three-phase protocol. The three phases for validation based protocol:

1. Read phase:

Values of committed data items from the database can be read by a transaction. Updates are only applied to local data versions.

2. Validation Phase:

Checking is performed to make sure that there is no violation of serializability when the transaction updates are applied to the database.

3. Write phase:

On the success of the validation phase, the transaction updates are applied to the database, otherwise, the updates are discarded and the transaction is slowed down.

In the following phase there are different time stamp

1. Start (T_i): It is the time when T_i started its execution.

2. Validation (T_i): It is the time when T_i just finished its read phase and begin its validation phase.

3. Finish (T_i): the time when T_i and it's all writing operation in the database under write-phase.

Two more terms that we need to know are:

1. Write-set: of a transaction contains all the write operation that T_i performs.

2. Read-set: of a transaction contains all the read operation that T_i performs.

Two-phase locking

2PL

Example: In the below example, if lock conversions is allowed then the following phase can happen:

1. Upgarding of lock (from S(a) to X(a)) is allowed in growing phase.
2. Downgarding of lock (from X(a) to S(a)) is must be done in shrinking phase.

	T_1	T_2
0	LOCK-S(A)	
1		LOCK-S(A)
2	LOCK-X(B)	
3	—	—
4	UNLOCK(A)	
5		LOCK-X(C)
6	UNLOCK(B)	
7		UNLOCK(A)
8		UNLOCK(C)
9	—	—

The following way shows how unlocking and locking work with 2PL.

Transaction T1:

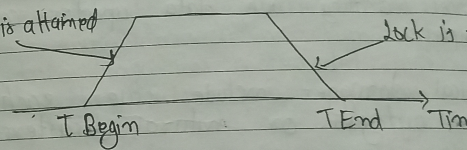
- Growing phase: from step 1-3
- Shrinking phase: from step 5-7
- Lock point: at 3

Transaction T2:

from step 2-6
from step 8-9
at 6

Define:

- The two-phase locking protocol divides the execution phase of the transaction into three parts.
- In the first part, when the execution of the transaction starts, it seeks permission for the lock it requires.
- In the second part, the transaction acquires all the locks. The third phase is started as soon as the transaction releases its first lock.
- In the third phase, the transaction cannot demand any new locks. It only release the acquired locks.



There are two phases of 2PL:

1) Growing phase: In this phase a transaction can only acquire locks but cannot release any lock. The point when a transaction acquires all the locks it needs is called the lock point.

2) Shrinking phase: In this phase a transaction can only release locks but cannot acquire any.

Recovery concepts

Data may be monitored stored, and changed rapidly and effectively using a DBMS. A database possesses atomicity, consistency, isolation, and durability qualities. The ability of a system to preserve data and changes made to data defines its durability.

A database could fail for any of the following reasons:

- System breakdowns occur as a result of hardware or software issues in the system.
- Transaction failures arise when a certain process dealing with data updates cannot be completed.
- Physical damages include issues such as power outages or natural disasters.

The recovery procedures in DBMS ensure the database's atomicity and durability. If a system crashes in the middle of a transaction and all of its data is lost, it is not regarded as durable. If just a portion of the data is updated during the transaction, it is not considered atomic. Data recovery procedures in DBMS make sure that the property data is always recoverable to protect the durability property and that its state is retained to protect the atomic property.

Types of recovery

There are two types of recovery

1) Log Based recovery
2) Shadow paging

1) Log Based Recovery :- This type of recovery is based on log in of records. There records are also called log Records. When even a transaction starts, the essential login is recorded. Under this, start flag have in a transaction identifier. When any work of storaging or updating data is done in the database, the log record stores the following :-

1) Transaction identification number
2) Data being processed
3) Old value of data
4) New value of that data

5) Type of log record

A log record in some database has a pointer to another log record that point to different log records. Records are stored permanently when even comment or rollback is given. Log records are stored in a file called log file. The current backup is kept in these log files. This work is done automatically by the database. The user is not required to take this type of backup.

To completely restore the database, DSA first restored all the offline backup and then restore all online backup, due to which the log file compare to their previous state, Database stored log file and data. Backup of transaction details are kept in log files and the actual data related to it is kept in data files.

Under this mainly two methods are adopted ~~first~~

1) Deferred update

2) Immediate update

2) Shadow paging

Shadow paging is one of the techniques that is used to recover from failure. We all know that recovery means to get back the information, which is lost. It helps to maintain database consistency in case of failure.

Concept of shadow paging

Now let see the concept of shadow paging step by step -

- Step 1 - Page is a segment of memory. Page table is an index of pages. Each table entry points to a page on the disk.
- Step 2 - Two page tables are used during the life of a transaction. The current page table and the shadow page table. Shadow page table is a copy of the current page table.
- Step 3 - When a transaction starts, both the tables look identical, the current table is updated for each write operation.
- Step 4 - The shadow page is never changed during the life of the transaction.

Step 5 - When the current transaction is comitted, the shadow page entry become a copy of the current page table entry and the disk block with the old data is released.

Step 6 - The shadow page table is stored in non-volatile memory. If the system crash occurs, then the shadow page table ~~entry~~ is copied to the current page table.

Database Security / Elementary concept of database security

Database Security means keeping sensitive information safe and prevent the loss of data. Security of database is controlled by Database administrator (DBA).

The following are the main control measures are used to provide security of data in database :

1. Authentication
2. Access control
3. Inference control
4. Flow control
5. Encryption

1. Authentication :

Authentication is the procers of confirmation that whether the user login ~~in~~ only according to the rights provided to him to perform. A particular user can login only up to his privilege but he can't access the other sensitive data.
[Authentication का मतलब यह है कि एक user सिर्फ अपनी id को ही login कर सकता है अगर वो दूसरे की id login करेगा तो वो unauthent होगा और data security दिख जाएगी]

2. Access Control;

Access Control ensure that what data user can access and what data he cannot access.

The accers control are of two types :-

1. Physical
2. Logical

3. Inference Control:

Inference control is a method of database security which is used to obtain hidden information from the normal users, hence, inference is also called mining technique.

Inference database is quite complex, which is good because the more complex the database is the more secure it is.

4. Flow control:

In this method used to monitor the flow of data between sender and receiver. With its help, it can be ascertained whether the database is happening properly or not.

- The flow of data is also monitored so that the speed of data flow can be increased so that the receiver receive the data of the faster speed.

5. Encryption:

Encryption is a technology used to ~~prevent~~ provide security to digital data. Encryption user password and keys to secure digital data.

To provide even greater security to the data encryption converts the data into secret code using algorithm, making the data completely secure. It is impossible to read these secret code.

System Failure

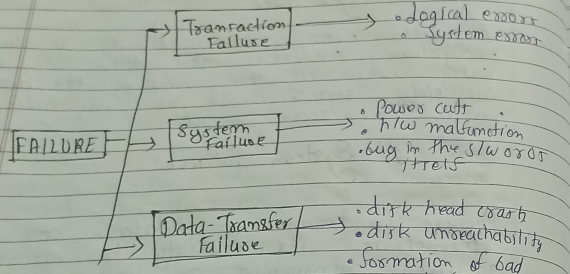
Failure in terms of database can be defined as its inability to execute the specified transaction or loss of data from the database.

A DBMS is unsafed (Vulnerable) to several kinds of failure and each of these failures needs to be managed differently.

There are many reasons that can cause database failures such as network failure, system crash, natural disasters, carelessness, software errors, etc.

Failure classification in DBMS

A failure in DBMS can be classified as:



A. Transaction Failure

If a transaction is not able to execute or it comes to a point from where the transaction becomes incapable of executing further then it is termed as a failure in a transaction.

Reason for a transaction failure in DBMS:

1) Logical error: A logical error occurs if a transaction is unable to execute because of some mistake in the code or due to the presence of some internal faults.

2) System error: When the transaction is executing but due to a fault in system, the transaction fails abruptly.

For ex:- Deadlock condition in transaction can result in System error.

B. System Crash

- A system ~~crash~~ failure can occur due to power failure or other hardware or software failure
  Ex:- Operating system errors.
- A system crash usually occurs when there is some sort of hardware or software breakdown. Some other problems which are external to the system and cause the system to abruptly stop or eventually crash include failure of the transaction, operating system errors, power cut, main memory crash, etc.

C. Data - Transfer Failure

- It occurs where hard-disk drives or storage drives used to fail frequently. It was a common problem in the early days of technology evolution.

- Disk failure occurs due to the formation of bad sectors, disk head crash, and unreachability to the disk or any other failure, which destroy all part of disk storage.

Authorization and authentication

Authorization and authentication are the two words used in the security world. They might sound similar but are completely different from each other. Authentication is used to authenticate (verify) someone's identity, whereas authorization is a way to provide permission to someone to access a particular resource. These are the two basic security terms and hence need to be understood.

★ What is Authentication? (Who are you)

- Authentication is the process of identifying someone's identity by assuring that the person is the same as what he is claiming for.

- It is used by both server and client. The server use authentication when someone wants to access the information, and the server needs to know who is accessing the info. The client uses it when he wants to know that it is the same server that it claims to be.

- The authentication by the server is done mostly by using the username and password. Others ways of authenication by the server can also be done using cards, retina scans, voice recognition, and fingerprints.

- Authentication does not ensure what tasks under a process one person can do, what files he can view, read, or update. It mostly identifies who the person or system is actually

=> Authentication factors

As per the security levels and the type of application, there are different types of authentication factors.

- Single-factor Authentication

Single-factor authentication is the simplest way of authentication. It just needs a username and password to allows a user to access a system.

- Two-Factor Authentication

As per the name, it is two level security; hence it needs two-step verification to authenticate a user. It does not require only a username and password but also needs the unique information that only the particular user knows, such as first school name, a favourite destination. Apart from this, it can also verify the user by sending the OTP or a unique link on the user's registered number or email address.

- Multi-Factor Authentication

This is the most secure and advanced level of authorization authentication. It requires two or more than two levels of security from different and independent categories. This type of authentication is usually used in financeral organizations, banks etc.

* What is authorization (what can you do)

- Authorization is the process of granting someone to do something. It means it a way to check if the user has permission to use a resource or not.
- It defines that what data and information one user can access. It
- The authorization usually works with authentication so that the system could know who is accessing the information.
- Authorization is not always necessary to access information available over the internet. Some data available over the internet can be accessed without any authorization.

→ Authorization Techniques

- Role-based access control

RBAC or Role-based access control technique is given to users as per their role or profile in the organization. It can be implemented for system-system or user-to ~~user~~ system.

- JSON web token

JSON web token of JWT is an open standard used to securely transmit the data between the parties in the form of the JSON object. The users are verified and authorized using the private/ public key pair.

- OpenID authorization

It helps the clients to verify the identity of end-users on the basis of authentication.

- OAuth

OAuth is an authorization protocol, which enables the API to authenticate and allows the requested resource.