What is transaction? List and explain ACID property of transaction with example.

Transaction

- A transaction is a part of program execution that accesses and updates various data items.
- A transaction can be defined as a group of tasks in which a single task is the minimum processing unit of work, which cannot be divided further.
- A transaction is a logical unit of work that contains one or more SQL statements.
- A transaction is an atomic unit (transaction either complete 0% or 100%).

ACID property

Atomicity

- Either all operations of the transaction are properly reflected in the database or none are.
- Means either all the operations of a transaction are executed or not a single operation is executed.
- For example, consider below transaction to transfer Rs. 50 from account A to account B:
    1. **read**(*A*)
    2. *A* := *A* – 50
    3. **write**(*A*)
    4. **read**(*B*)
    5. *B* := *B* + 50
    6. **write**(*B*)
- In above transaction if Rs. 50 is deducted from account A then it must be added to account B.

Consistency

- Execution of a transaction in isolation preserves the consistency of the database.
- Means our database must remain in consistent state after execution of any transaction.
- In above example total of A and B must remain same before and after the execution of transaction.

Isolation

- Although multiple transactions may execute concurrently, each transaction must be unaware of other concurrently executing transactions.
- Intermediate transaction results must be hidden from other concurrently executed transactions.
- In above example once your transaction start from step one its result should not be access by any other transaction until last step (step 6) is completed.

Durability

- After a transaction completes successfully, the changes it has made to the database persist, even if there are system failures.
- Once your transaction completed up to step 6 its result must be stored permanently. It should not be removed if system fails.

Explain different states in transaction processing in database.
OR
Explain State Transition Diagram (Transaction State Diagram).

- Because failure of transaction may occur, transaction is broken up into states to handle various situations.
- Following are the different states in transaction processing in database
  - ✓ Active
  - ✓ Partial committed
  - ✓ Failed
  - ✓ Aborted
  - ✓ Committed

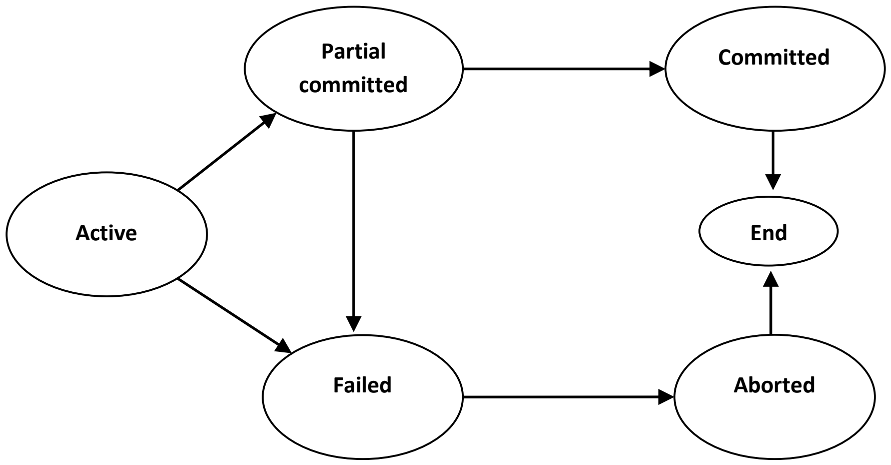


Fig. State Transition Diagram

Active
- This is the initial state. The transaction stays in this state while it is executing.

Partially Committed
- This is the state after the final statement of the transaction is executed.
- At this point failure is still possible since changes may have been only done in main memory, a hardware failure could still occur.
- The DBMS needs to write out enough information to disk so that, in case of a failure, the system could re-create the updates performed by the transaction once the system is brought back up.
- After it has written out all the necessary information, it is committed.

Failed
- After the discovery that normal execution can no longer proceed.
- Once a transaction cannot be completed, any changes that it made must be undone rolling it back.

Aborted

- The state after the transaction has been rolled back and the database has been restored to its state prior to the start of the transaction.

Committed

- The transaction enters in this state after successful completion of the transaction.
- We cannot abort or rollback a committed transaction.

What is Schedule? Explain types of schedules.

Schedule

- A schedule is the chronological (sequential) order in which instructions are executed in a system.
- A schedule for a set of transaction must consist of all the instruction of those transactions and must preserve the order in which the instructions appear in each individual transaction.
- Example of schedule (Schedule 1)

T1	**T2**
read(A)	
A:=A-50	
write(A)	
read(B)	
B:= B+ 50	
write(B)	
	read(A)
	temp: A * 0.1
	A: A-temp
	write (A)
	read(B)
	B:=B +temp
	write(B)

Serial schedule

- Schedule that does not interleave the actions of different transactions.
- In schedule 1 the all the instructions of T1 are grouped and run together. Then all the instructions of T2 are grouped and run together.
- Means schedule 2 will not start until all the instructions of schedule 1 are complete. This type of schedules is called serial schedule.

Non-Serial schedule (**Or** Interleaved schedule)

- Schedule that interleave the actions of different transactions.

- Means schedule 2 will start before all instructions of schedule 1 are completed. This type of schedules is called interleaved schedule.

T1	T2
read(A)	
A:=A-50	
write(A)	
	read(A)
	temp: A * 0.1
	A: A-temp
	write (A)
read(B)	
B:= B+ 50	
write(B)	
	read(B)
	B:=B +temp
	write(B)

Equivalent schedules

- Two schedules are equivalent schedule if the effect of executing the first schedule is identical (same) to the effect of executing the second schedule.
- We can also say that two schedule are equivalent schedule if the output of executing the first schedule is identical (same) to the output of executing the second schedule.

Differentiate Serial Schedule Vs Non-Serial Schedule?

Serial Schedule	Non- Serial Schedule
A serial schedule is a schedule in which no transaction starts until a running transaction has ended.	Schedule that interleaves the execution of different transactions, means second transaction is started before the first one could end.
In Serial schedule execution of transaction is very slow because on one transaction can execute at a time.	In Non-Serial schedule, parallel execution of multiple transaction is possible at a time. So, non-serial schedules are fast in execution.

In Serial schedule transactions are executed one after another.	Non-Serial schedule contains many possible orders in which the system can execute the individual operations of the transactions.
Example: If some transaction T is long, the other transaction must wait for T to complete all its operations.	Example: In this schedule the execution of other transaction goes on without waiting the completion of T.

Explain two phase commit protocol. OR Explain working of two phase commit protocol.

Two phase commit protocol

- The two phase commit protocol provides an automatic recovery mechanism in case a system or media failure occurs during execution of the transaction.
- The two phase commit protocol ensures that all participants perform the same action (either to commit or to roll back a transaction).
- The two phase commit strategy is designed to ensure that either all the databases are updated or none of them, so that the databases remain synchronized.
- In two phase commit protocol there is one node which is act as a coordinator and all other participating node are known as cohorts or participant.
- Coordinator – the component that coordinates with all the participants.
- Cohorts (Participants) – each individual node except coordinator are participant.
- As the name suggests, the two phase commit protocol involves two phases.
    1. The first phase is Commit Request phase OR phase 1
    2. The second phase is Commit phase OR phase 2

Commit Request Phase (Obtaining Decision)

- To commit the transaction, the coordinator sends a request asking for “ready for commit” to each cohort.
- The coordinator waits until it has received a reply from all cohorts to “vote” on the request.
- Each participant votes by sending a message back to the coordinator as follows:
    - ✓ It votes YES if it is prepared to commit
    - ✓ It may vote NO for any reason if it cannot prepare the transaction due to a local failure.
    - ✓ It may delay in voting because cohort was busy with other work.

Commit Phase (Performing Decision)

- If the coordinator receives YES response from all cohorts, it decides to commit. The transaction is now officially committed. Otherwise, it either receives a NO response or gives up waiting for some cohort, so it decides to abort.
- The coordinator sends its decision to all participants (i.e. COMMIT or ABORT).

- Participants acknowledge receipt of commit or about by replying DONE.

What is database recovery? OR What is system recovery?

- Database recovery is the process of restoring a database to the correct state in the event of a failure.
- Database recovery is a service that is provided by the DBMS to ensure that the database is reliable and remain in consistent state in case of a failure.
- Restoring a physical backup means reconstructing it and making it available to the database server.
- To recover a restored backup, data is updated using redo command after the backup was taken.
- Database server such as SQL server or ORACLE server performs cash recovery and instance recovery automatically after an instance failure.
- In case of media failure, a database administrator (DBA) must initiate a recovery operation.
- Recovering a backup involves two distinct operations: rolling the backup forward to a more recent time by applying redo data and rolling back all changes made in uncommitted transactions to their original state.
- In general, recovery refers to the various operations involved in restoring, rolling forward and rolling back a backup.
- Backup and recovery refer to the various strategies and operations involved in protecting the database against data loss and reconstructing the database.

Explain Log based recovery method.

Log based recovery

- The most widely used structure for recording database modification is the log.
- The log is a sequence of log records, recording all the update activities in the database.
- In short Transaction log is a journal or simply a data file, which contains history of all transaction performed and maintained on stable storage.
- Since the log contains a complete record of all database activity, the volume of data stored in the log may become unreasonable large.
- For log records to be useful for recovery from system and disk failures, the log must reside on stable storage.
- Log contains
    1. Start of transaction
    2. Transaction-id
    3. Record-id

4. Type of operation (insert, update, delete)
5. Old value, new value
6. End of transaction that is committed or aborted.

- All such files are maintained by DBMS itself. Normally these are sequential files.
- Recovery has two factors Rollback (Undo) and Roll forward (Redo).
- When transaction T_i starts, it registers itself by writing a <T_i start>log record
- Before T_i executes write(X), a log record <T_i, X, V_1, V_2> is written, where V_1 is the value of X before the write, and V_2 is the value to be written to X.
  - ✓ Log record notes that T_i has performed a write on data item X_j
  - ✓ X_j had value V_1 before the write, and will have value V_2 after the write.
- When T_i finishes it last statement, the log record <T_i commit> is written.

Log based Recovery Methods

- Once a failure occurs, DBMS retrieves the database using the back-up of database and transaction log. Various log based recovery techniques used by DBMS are as per below:
  1. **Deferred Database Modification**
  2. **Immediate Database Modification**
- Both of the techniques use transaction logs.

Immediate Database Modification

Updates (changes) to the database are applied immediately as they occur without waiting to reach to the commit point.

Deferred Database Modification

Updates (changes) to the database are deferred (postponed) until the transaction commits.

Immediate Vs Deferred database modification

Immediate Database Modification	Deferred Database Modification
Updates (changes) to the database are applied immediately as they occur without waiting to reach to the commit point.	Updates (changes) to the database are deferred (postponed) until the transaction commits.
If transaction is not committed, then we need to do undo operation and restart the transaction again.	If transaction is not committed, then no need to do any undo operations. Just restart the transaction.
If transaction is committed, then no need to do redo the updates of the transaction.	If transaction is committed, then we need to do redo the updates of the transaction.
Undo and Redo both operations are performed.	Only Redo operation is performed.

What is concurrency? What are the three problems occurring due to concurrency? How the problems can be avoided?

Concurrency

- Concurrency is the ability of a database to allow multiple (more than one) users to access data at the same time.

Three problems due to concurrency

1. **The lost update problem**: This problem indicates that if two transactions T1 and T2 both read the same data and update it then effect of first update will be overwritten by the second update.

T1	Time	T2
---	T0	---
Read X	T1	---
---	T2	Read X
Update X	T3	---
---	T4	Update X
---	T5	---

   How to avoid: In above example a transaction T2 must not update the data item (X) until the transaction T1 can commit data item (X).

2. **The dirty read problem**: The dirty read arises when one transaction update some item and then fails due to some reason. This updated item is retrieved by another transaction before it is changed back to the original value.

T1	Time	T2
---	T0	---
---	T1	Update X
Read X	T2	---
---	T3	Rollback
---	T5	---

   How to avoid: In above example a transaction T1 must not read the data item (X) until the transaction T2 can commit data item (X).

3. **The incorrect retrieval problem**: The inconsistent retrieval problem arises when one transaction retrieves data to use in some operation but before it can use this data another transaction updates that data and commits. Through this change will be hidden from first transaction and it will continue to use previous retrieved data. This problem is also known as inconsistent analysis problem.

Balance (A=200 B=250 C=150)		
T1	**Time**	**T2**
---	T0	---
Read (A) Sum ← 200	T1	---
Read (B) Sum ← Sum + 250 = 450	T2	---
---	T3	Read (C)
---	T4	Update (C) 150 → 150 – 50 = 100
---	T5	Read (A)
---	T6	Update (A) 200 → 200 + 50 = 250
---	T7	COMMIT
Read (C) Sum ←Sum + 100 = 550	T8	---

How to avoid: In above example a transaction T2 must not read or update data item (X) until the transaction T1 can commit data item (X).

Define lock and locking. Explain lock based protocol.

Lock

- A lock is a variable associated with data item to control concurrent access to that data item.
- Lock requests are made to concurrency-control manager.
- Transaction can proceed only after request is granted.

Locking

- One major problem in databases is concurrency.
- Concurrency problems arise when multiple users try to update or insert data into a database table at the same time. Such concurrent updates can cause data to become corrupt or inconsistent.
- Locking is a strategy that is used to prevent such concurrent updates to data.

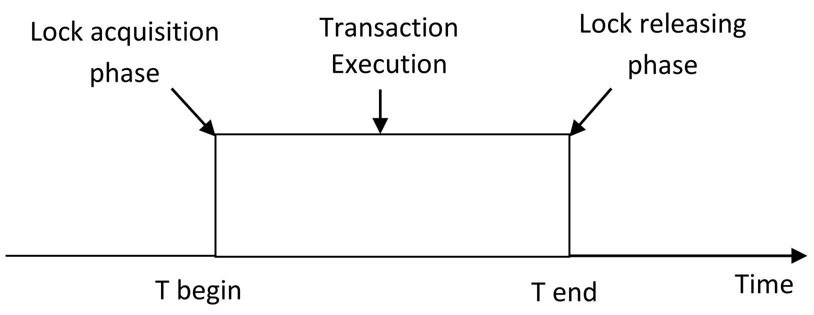
Lock based protocol

- 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 concurrency-control manager.
- Transaction can proceed only after request is granted.
- Lock-compatibility matrix

	Shared Lock	**Exclusive Lock**
Shared Lock	TRUE	FALSE
Exclusive Lock	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.
- This locking protocol divides transaction execution phase into three parts.
  1. In the first part, when transaction starts executing, transaction seeks grant for locks it needs as it executes.
  2. Second part is where the transaction acquires all locks and no other lock is required. Transaction keeps executing its operation.
  3. As soon as the transaction releases its first lock, the third phase starts. In this phase a transaction cannot demand for any lock but only releases the acquired locks.



Explain two phase locking protocol. What are its advantages and disadvantages? OR Explain two phase locking. Explain its advantage and disadvantage.

Two-Phase Locking Protocol

- The use of locks has helped us to create neat and clean concurrent schedule.
- The Two Phase Locking Protocol defines the rules of how to acquire the locks on a data item and how to release the locks.
- Two phase locking (2PL) is a concurrency control method that guarantees serializability.

- The protocol utilizes locks, applied by a transaction on data, which may block (stop) other transactions from accessing the same data during the transaction's life.
- The Two Phase Locking Protocol assumes that a transaction can only be in one of two phases.

Phase 1 - Growing Phase

- ✓ In this phase the transaction can only acquire locks, but cannot release any lock.
- ✓ The transaction enters the growing phase as soon as it acquires the first lock it wants.
- ✓ From now on it has no option but to keep acquiring all the locks it would need.
- ✓ It cannot release any lock at this phase even if it has finished working with a locked data item.
- ✓ Ultimately the transaction reaches a point where all the lock it may need has been acquired. This point is called **Lock Point**.

Phase 2 - Shrinking Phase

- ✓ After Lock Point has been reached, the transaction enters the shrinking phase.
- ✓ In this phase the transaction can only release locks, but cannot acquire any new lock.
- ✓ The transaction enters the shrinking phase as soon as it releases the first lock after crossing the Lock Point.
- ✓ From now on it has no option but to keep releasing all the acquired locks.

- Initially the transaction is in growing phase, that is the transaction acquires locks as needed.
- Once the transaction releases lock, it enters the shrinking phase and no more lock request may be issued.
- Upgrading of lock is not possible in shrinking phase, but it is possible in growing phase.