



## UNIT IV - Transaction

Transaction Concepts – ACID Properties – Schedules – Serializability – Concurrency Control – Need for Concurrency – Locking Protocols – Two Phase Locking – Deadlock – Transaction Recovery – Save Points – Isolation Levels – SQL Facilities for Concurrency and Recovery.

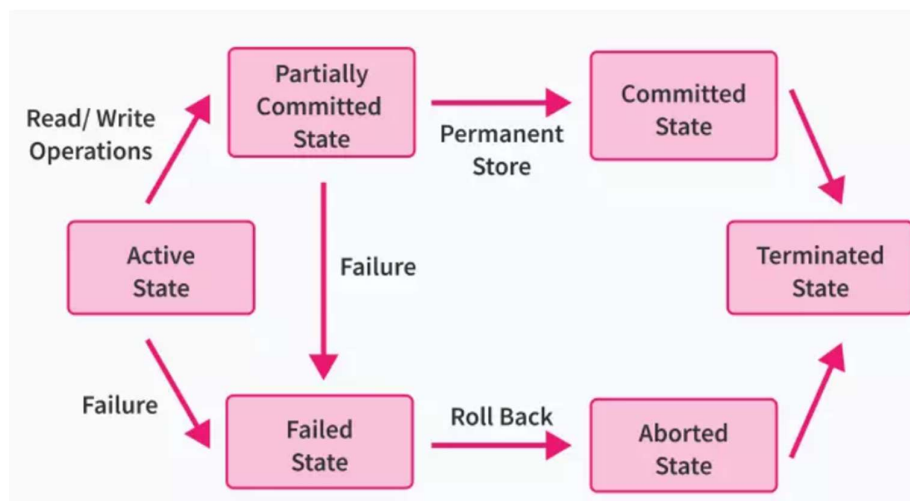
### Transaction Concepts

A transaction is a single, logical unit of work that consists of one or more related tasks. A transaction is treated as a single, indivisible operation, which means that either all the tasks within the transaction are executed successfully, or none are.

#### Transaction States in DBMS

During the lifetime of a transaction, there are a lot of states to go through. These states update the operating system about the current state of the transaction and also tell the user about how to plan further processing of the transaction. These states decide the regulations which decide the fate of a transaction whether it will commit or abort.

The ROLLBACK statement undo the changes made by the current transaction. A transaction cannot undo changes after COMMIT execution.



### **Following are the different types of transaction States :**

**Active State:** When the operations of a transaction are running then the transaction is said to be in active state. If all the read and write operations are performed without any error then it progresses to the partially committed state, if somehow any operation fails, then it goes to a state known as failed state.

**Partially Committed:** After all the read and write operations are completed, the changes which were previously made in the main memory are now made permanent in the database, after which the state will progress to committed state but in case of a failure it will go to the failed state.

**Failed State:** If any operation during the transaction fails due to some software or hardware issues, then it goes to the failed state . The occurrence of a failure during a transaction makes a permanent change to data in the database. The changes made into the local memory data are rolled back to the previous consistent state.

**Aborted State:** If the transaction fails during its execution, it goes from failed state to aborted state and because in the previous states all the changes were only made in the main memory, these uncommitted changes are either deleted or rolled back. The transaction at this point can restart and start afresh from the active state.

**Committed State:** If the transaction completes all sets of operations successfully, all the changes made during the partially committed state are permanently stored and the transaction is stated to be completed, thus the transaction can progress to finally get terminated in the terminated state.

**Terminated State:** If the transaction gets aborted after roll-back or the transaction comes from the committed state, then the database comes to a consistent state and is ready for further new transactions since the previous transaction is now terminated.

### **Properties of Transaction in DBMS**

There are four major properties that are vital for a transaction to be successful. These are used to maintain state consistency in the database, both before and after the transaction. These are called ACID properties.

**Atomicity:** This property means that either the transaction takes place completely at once or doesn't happen at all. There is no middle option, i.e., transactions do not occur partially. Each transaction is considered as one single step which either runs completely or is not executed at all.

Example) Transactions taking place in a bank (Credit or Debit of Money)

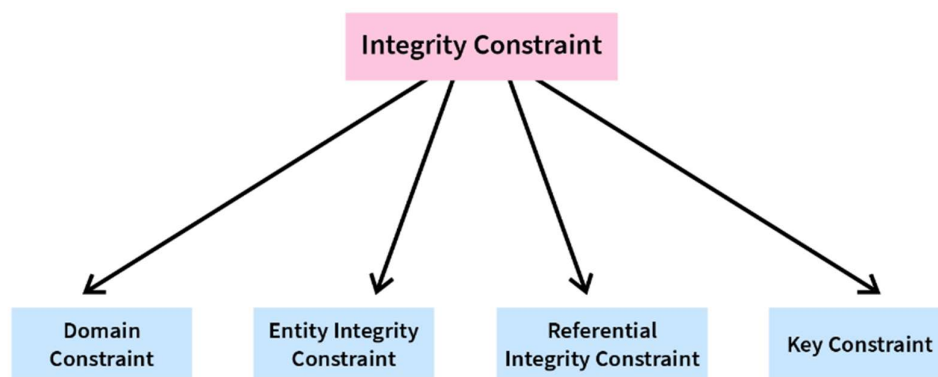
If Marino has an account namely A with \$50 in it and wants to send \$20 to Amanda who has an account namely B. An amount of \$200 is already existing in account B. When \$20 is deposited to account B, the total becomes \$220.

Two procedures are now scheduled to take place. One is that the \$20 that Marino wishes to send will be deducted from his account A and would be credited to account B, i.e., into Amanda's account. What happens now is that the initial debit operation succeeds, but the crediting operation fails.

As a result, the value in Marino's account A becomes 30\$, while the value in Amanda's account remains \$200 as it was earlier.

**Consistency:** This property means that the integrity constraints of a database are maintained so that the database is consistent before and after the transaction. It refers to the correctness of a database.

In Database Management Systems, integrity constraints are pre-defined set of rules that are applied on the table fields(columns) or relations to ensure that the overall validity, integrity, and consistency of the data present in the database table is maintained. Evaluation of all the conditions or rules mentioned in the integrity constraint is done every time a table insert, update, delete, or alter operation is performed. The data can be inserted, updated, deleted, or altered only if the result of the constraint comes out to be True. Thus, integrity constraints are useful in preventing any accidental damage to the database by an authorized user.



### Isolation in DBMS

Isolation is defined as a state of separation. Isolation is an ACID Property in DBMS where no data from one database should impact the other and where many transactions can take place at the same time. In other words, when the operation on the first state of the database is finished, the process on the second state of the database should begin. It indicates that if two actions are conducted on two different databases, the value of one database may not be affected by the value of the other. When two or more transactions occur at the same time in the case of transactions, consistency should be maintained. Any modifications made in one transaction will not be visible to other transactions until the change is committed to the memory.

**Example** Suppose two people try to book the same seat simultaneously. Transactions are serialized to maintain data consistency. The first person's transaction succeeds, and they receive a ticket. The second person's transaction fails as the seat is already booked. They receive an error message indicating no available seats.

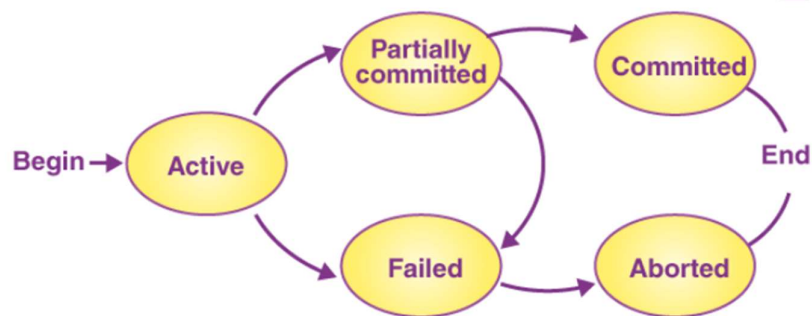
### **Durability in DBMS**

The ACID Property durability in DBMS refers to the fact that if an operation is completed successfully, the database remains permanent in the disk. The database's durability should be such that even if the system fails or crashes, the database will survive. However, if the database is lost, the recovery manager is responsible for guaranteeing the database's long-term viability. Every time we make a change, we must use the COMMIT command to commit the values.

**Example** Suppose that there is a system failure in the railway management system resulted in the loss of all booked train details. Millions of users who had paid for their seats are now unable to board the train, causing significant financial losses and eroding trust in the company. The situation is particularly critical as these trains are needed for important reasons, causing widespread panic and inconvenience.

### **States of Transactions**

In a database, a transaction can be in one of these states given below –



**Active** – This is the state in which a transaction is being executed. Thus, it is like the initial state of any given transaction.

**Partially Committed** – A transaction is in its partially committed state whenever it executes the final operation.

**Failed** – In case any check made by a database recovery system fails, then that transaction is in a failed state. Remember that a failed transaction can not proceed further.

**Aborted** – In case any check fails, leading the transaction to a failed state, the recovery manager then rolls all its write operations back on the database so that it can bring the DB (database) back to the original state (the state where it actually was prior to the transaction execution). The transactions in this state are known to be aborted. A DB recovery module can actually select

one of these two operations after the abortion of a transaction –

**Re-start**

Kill the transaction

**Committed** – We can say that a transaction is committed in case it actually executes all of its operations successfully. In such a case, all of its effects are now established permanently on the DB system.