

# Transaction management in DBMS

Krishna Agrawal, 19111031

January 21, 2022

## 1 Introduction to Transactions

Transactions are a set of operations used to perform a logical set of work. A transaction usually means that the data in the database has changed. One of the major uses of DBMS is to protect the user's data from system failures. It is done by ensuring that all the data is restored to a consistent state when the computer is restarted after a crash. The transaction is any one execution of the user program in a DBMS. Executing the same program multiple times will generate multiple transactions.

## 2 Properties of Transaction

There are properties that all transactions should follow and possess. The four basic are in combination termed as ACID properties. ACID properties and its concepts of a transaction are put forwarded by Haerder and Reuter in the year 1983. The ACID has a full form and is as follows:

- Atomicity: The 'all or nothing' property. A transaction is an indivisible entity that is either performed in its entirety or will not get performed at all. This is the responsibility or duty of the recovery subsystem of the DBMS to ensure atomicity.
- Consistency: A transaction must alter the database from one steady-state to another steady state. This is the responsibility of both the DBMS and the application developers to make certain consistency. The DBMS can ensure consistency by putting into effect all the constraints that have been mainly on the database schema such as integrity and enterprise constraints.
- Isolation: Transactions that are executing independently of one another is the primary concept followed by isolation. In other words, the frictional effects of incomplete transactions should not be visible or come into notice to other transactions going on simultaneously. It is the responsibility of the concurrency control sub-system to ensure adapting the isolation.
- Durability: The effects of an accomplished transaction are permanently recorded in the database and must not get lost or vanished due to subsequent failure. So this becomes the responsibility of the recovery sub-system to ensure durability.

### 3 DBMS States of Transaction

A database, the transaction can be in one of the following states -

- Active state

The active state is the first state of every transaction. In this state, the transaction is being executed. For example: Insertion or deletion or updating a record is done here. But all the records are still not saved to the database.

- Partially committed

In the partially committed state, a transaction executes its final operation, but the data is still not saved to the database. In the total mark calculation example, a final display of the total marks step is executed in this state.

- Committed

A transaction is said to be in a committed state if it executes all its operations successfully. In this state, all the effects are now permanently saved on the database system.

- Failed state

If any of the checks made by the database recovery system fails, then the transaction is said to be in the failed state. In the example of total mark calculation, if the database is not able to fire a query to fetch the marks, then the transaction will fail to execute.

- Aborted

If any of the checks fail and the transaction has reached a failed state then the database recovery system will make sure that the database is in its previous consistent state. If not then it will abort or roll back the transaction to bring the database into a consistent state. If the transaction fails in the middle of the transaction then before executing the transaction, all the executed transactions are rolled back to its consistent state. After aborting the transaction, the database recovery module will select one of the two operations:

- Re-start the transaction
- Kill the transaction