

Distributed Database System (ECS-242/602)

Mr. Abdul Rehman

email id: abdul@northern.edu.pk Whatsapp# 03087792217

Objectives: Learning objectives of this lecture are

- **Transaction**
- **Phases of Transaction**
- **Execution of Transaction**

Overview:

In this lecture, we will learn what are transaction in database? Examples of transactions. What is the need of transaction? Properties of transaction.

Transaction:

An action, or series of actions, carried out by a single user or application program, which reads or updates the content of the database.

A transaction is a logical unit of work on database. It may be an entire program, a part of program, or a single command (for example, the SQL command INSERT or UPDATE), and it may involve any number of operations on the database.

Major property of Transaction is that it executes as a unit. By unit we mean all operations in a transaction must remain in sequence and must execute successfully. If any of operation in a transaction does not execute successfully then it will be just some operations not a transaction. A Transaction has possible two outcomes. Successful or Failed. In case of Successful transaction, all operations in a transaction must be executed successfully and in a sequence. In case of Failed transaction, all operations which are executed must revert back so that no operation will have effect on database.

Example:

Let us say, a user wants to withdraw amount from bank using an ATM machine. Following steps will be carried out complete transaction.

- 1- Insert correct PIN**
- 2- Input Valid amount to be withdrawn**
- 3- Deduction of Amount from user's account (by bank)**
- 4- Release of Money by ATM machine.**

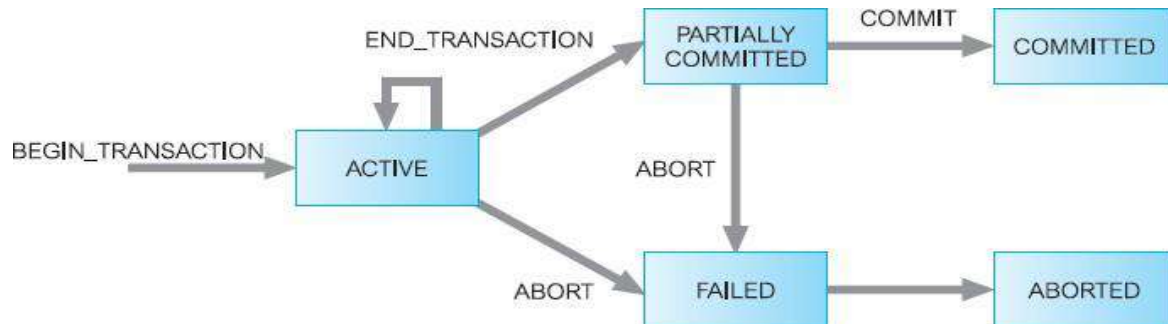
Now, here transaction's importance is highlighted. All above operations must be in sequence to complete the process of Money withdrawal. If any of step is not completed, then user will not get his/her amount. Now let us say, first three steps of above process are completed successfully. Which means

- 1- Insert correct PIN**
- 2- Input Valid amount to be withdrawn**
- 3- Deduction of Amount from user's account (by bank)**
- 4- Release of Money by ATM machine.**

Now in above situation, if we do not do anything. Amount from user's account will be deducted but amount will not be received as last step has failed. And user will be pretty upset about it. Here comes the transaction importance, if above process is executed as transaction, it will revert all above processes on failure of step 4. Which means it will revert user's amount back to his/her account.

Phases of Transaction:

- 1- Active
- 2- Partially committed
- 3- Committed
- 4- Failed
- 5- Aborted



1- Active

When operations of a transaction are being executed then it is said to exist in “Active” phase.

2- Partially Committed

When all operations are successfully executed in a transaction, it moves to “Partially Committed” phase. At this phase, DBMS check whether any integrity constraint or any rule of DBMS is violated or not.

- i- If no rule of database is violated then system moves toward “Committed” phase.
- ii- If a rule of database is violated during execution of transaction’s operations, then it is moved toward “Failed” phase.

3- Committed

Committed states means all operations in transaction are successful and no database rule is violated. At this stage data is written on database permanently, and transaction is termed as successful transaction. After commit stage, transaction cannot roll back or revert back.

4- Failed

At this stage, any operation which has made changes in database are reverted.

5- Aborted

At this stage, transaction is termed as failed transaction. Aborted means transaction has not affected database because all its operations have been reverted. Which means database is at same position where it was when transaction was started.

Distributed Database System (ECS-242/602)

Mr. Abdul Rehman

email id: abdul@northern.edu.pk Whatsapp# 03087792217

How We will Represent Transaction Onwards:

T1
Read (x)
$x = x + 20$
Read (y)
$x = x + y$
Write(x)
Commit

- T1 is the name of transaction, it can be anything.
- Read operations reads data from permanent storage (DB) and fetch its value into memory (RAM) and stores in a local variable of RAM.
- Write operations stores changed value of variable into the permanent storage. (which means data is stored in database at WRITE operation)
- X / Y can be any value in database for example (age of employee, CGPA of student).

Let us execute above transaction in detailed way.

$x = 50, y = 20$.

We will execute step by step

T1	Local Variable Value (RAM)	Database Value (HardDrive)
Read (x)	$x=50$	$x=50$
$x = x + 20$	$x=70$	$x=50$
Read (y)	$y=20$	$y=20$
$x = x + y$	$x=90$	$x=50$
Write(x)	$x=90$	$x=90$
Commit		

Here we can see that local variable values are updated by database value is only updated whenever Write operation is issued. Until that, database value will not be changed.

Now let us do above process for multiple transactions executing at the same time.

Distributed Database System (ECS-242/602)

Mr. Abdul Rehman

email id: abdul@northern.edu.pk Whatsapp# 03087792217

x=100, y=50 (Database Values)

Time	T1	T2
t0	Read (x)	
t1	x = x + 20	Read(x)
t2	Read (y)	x = x + 50
t3	x = x + y	Write(x)
t4	Write(x)	Commit
t5	Commit	

Here in above example, there are two transactions running at the same time. The execution pattern will be left to right. First operation $t0 \rightarrow T1$ will be executed. As there is no operation $t0 \rightarrow T2$. So, system will move towards the next time which is t1. So, execution result of above transactions will look like as given below.

x=100, y=50 (Database Values)

Time	T1	T1- Local Variable Values (x, y)	Database Values (x, y)	T2	T2- Local Variable Values (x, y)
t0	Read (x)	x=100, y=50	x=100, y=50		
t1	x = x + 20	x=120, y=50	x=100, y=50	Read(x)	x=100, y=50
t2	Read (y)	x=120, y=50	x=100, y=50	x = x + 50	x=100, y=50
t3	x = x + y	x=170, y=50	x=150, y=50	Write(x)	x=150, y=50
t4	Write(x)	x=170, y=50	x=170, y=50	Commit	x=170, y=50
t5	Commit				

Explanation:

Here we can see that each transaction has its own local variable which does not interfere in other transaction's variable. But database variables are same because both transactions are accessing same database.

- Both will read from database values
- Operations will be performed on local variables
- Write statement will update database variables from values of local variables.

Distributed Database System (ECS-242/602)

Mr. Abdul Rehman

email id: abdul@northern.edu.pk Whatsapp# 03087792217

-
- Here important thing is why T2 has read $x=100$, it is because transaction T1 has not changed value in database, it only updated its local variable which is not accessible to other transactions.

Assignment #5

Question#1: Give any three examples of transaction where next operations is dependent upon previous and focus on importance of transaction.

Question#2: Execute the given transaction left to right and fill “local variable” and “Database variable” columns.

Initial Database Values:

$x=30, y=50, z=80$

Time	T1	T1- Local Variable Values (x, y)	Database Values (x, y)	T2	T2- Local Variable Values (x, y)
t0	Read (x)				
t1	$x = x * 2$			Read(y)	
t2	Read (z)			$y = y + 50$	
t3	$x = x + z$			Write(y)	
t4	Write(x)			Read(x)	
t5	Read (y)			$x=x + 50$	
t6	$z = z + y$			Read(z)	
t7	Write (z)			$x = x + z$	
t8	Commit			Write(x)	
t9				Commit	