|  |
| --- |
| **National University of Computer and Emerging Sciences** |
| Lab Manual 9  “Transactions Manual” |
|  |
| Database Systems |
| Spring 2018 |

Department of Computer Science

FAST-NU, Lahore, Pakistan

Contents

[Transactions 2](#_Toc478624053)

[Consistent State 2](#_Toc478624054)

[Transaction Properties 2](#_Toc478624055)

[Atomicity 2](#_Toc478624056)

[Consistency 2](#_Toc478624057)

[Isolation 2](#_Toc478624058)

[Durability 2](#_Toc478624059)

[Serializability 3](#_Toc478624060)

[Types of Transactions 3](#_Toc478624061)

[DML Transactions 3](#_Toc478624062)

[Transaction Format 3](#_Toc478624063)

[Commit 3](#_Toc478624064)

[Rollback 3](#_Toc478624065)

[Transactions in Stored Procedures: 5](#_Toc478624066)

[Transactions with/without Locks: 6](#_Toc478624067)

[Note: 6](#_Toc478624068)

# Objective

The purpose of this lab is to introduce the students to the transactions in sql.

# Prerequisite

All the lab manuals till now.

# Task Distribution

|  |  |
| --- | --- |
| Total Time | 170 Minutes |
| Introduction | 10 Minutes |
| Master Page Exercise | 50 Minutes |

# Transactions

In database terms, a **transaction** is any action that reads from and/or writes to a database. A transaction may consist of a simple SELECT statement to generate a list of table contents; it may consist of a series of related UPDATE statements to change the values of attributes in various tables; it may consist of a series of INSERT statements to add rows to one or more tables; or it may consist of a combination of SELECT, UPDATE, and INSERT statements. Each successful transaction leaves the database in a consistent state

## Consistent State

A consistent state is one in which all the database integrity constraints are being fulfilled, which means that each transaction must be atomic i.e it either runs completely or does not run at all.

A transaction is a logical unit of work which must be entirely executed or entirely aborted. Atomicity is the most important principal in the transactions.

# Transaction Properties

Transactions possess the following properties that make the transactions trustworthy.

* Atomicity
* Consistency
* Isolation
* Durability
* Serializability

## Atomicity

Requires all the operations of a transactions to be completed. If an instruction/query is not completely executed it should be aborted. A transaction is treated as a single indivisible logical unit. As mentioned above that the logical unit either executes completely or does not execute at all.

## Consistency

Consistency dictates that each transaction from the database leaves the database in a consistent states i.e. no integrity constraints are violated. The transactions change the database from one consistent state to another consistent state.

## Isolation

Isolation implies that the data used in one transaction can’t be used in another transaction unless the first transaction is complete.

## Durability

Durability ensures that once the changes are made in the database they can’t be undone even in the case of system failure.

## Serializability

Serializability property belongs solely to the multiuser scenario it ensures that the schedule for the concurrent transactions yield a consistent result.

# Types of Transactions

Unlike the other types of instructions and queries that are only of two types namely DDL and DML we have an additional type of transaction so the three types of transactions are as follows:

* DDL transactions
* DML transactions
* Transactions control statements

## DML Transactions

Let us begin with the DML transactions and after that in the next section we shall move to the DDL transactions

## Transaction Format

The format of transaction is as follows:

Begin Transaction

Sql queries (DDL or DML)

Commit/Rollback

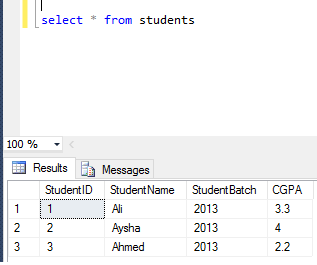
A transaction can be either committed or rolled back

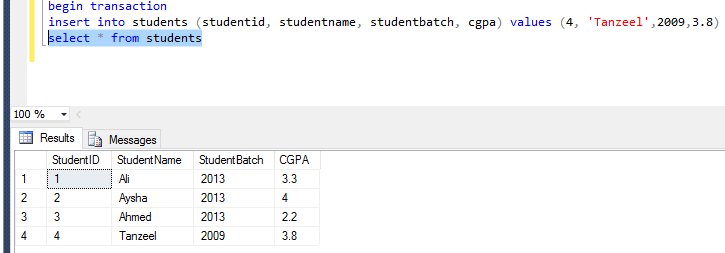
## Commit

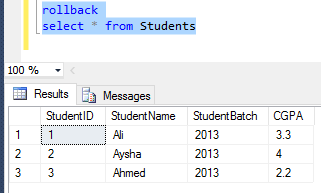
Commit command is used to save all the changes that are made in the sql queries in the transaction. The queries that follow the begin transaction command are a part of the transaction.

## Rollback

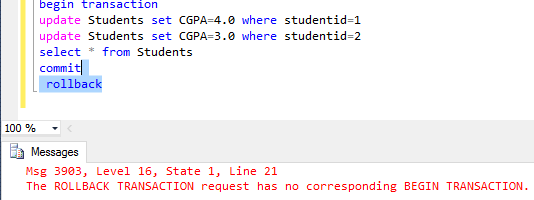
Rollback takes the data back to the last save point or to the beginning of the transaction depending on the scenario, which means that all the changes since the last save-point are undone.





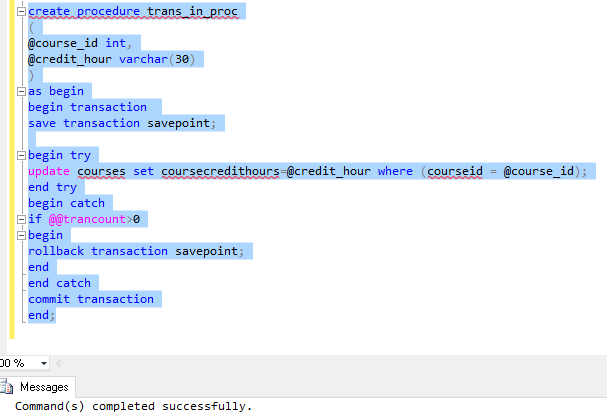


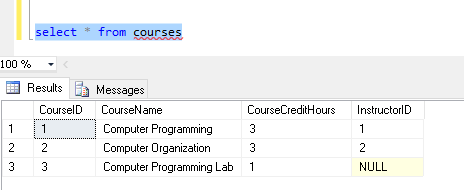
Similarly we can apply this to any of the DML queries. Before moving forward, let us see that every rollback must have a corresponding begin transaction. Because once a transaction has been committed it cannot be rolled back it is a permanent change in the database.

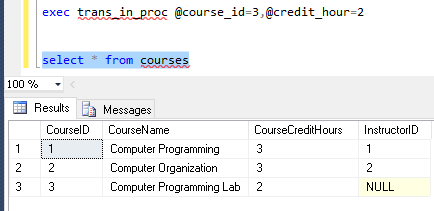


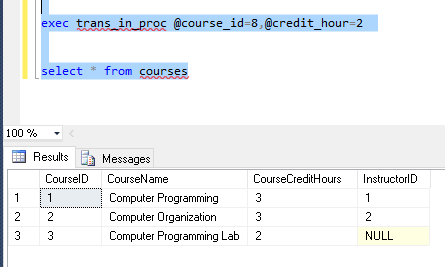
# Transactions in Stored Procedures:

In the following code you will see “save transaction” in the code this is used to create a save point in the transaction it is a point in the transaction where in case of roll back the transaction should roll back to i.e. the changes will be undone till this point in the transaction.









# Transactions with/without Locks:

Transaction with no locks don ‘t allow multiple command s to be run on the same data set however in case of transactions with no locks allow you to run multiple commands on the same data set.

Note:

In case of nested transactions, the rollback takes back to the outer most transactions.

# Practice Exercise

Create a procedure for student registration in some course the procedure must check that the CGPA of the student if the CGPA is less than 2.5 the transaction must rollback and the record must not be stored and a message must be displayed that he can only enroll in subjects that he can improve however if the student’s CGPA is greater than 2.5 he can enroll in any subject he wants in this case the transaction should be committed. Assume that the enrollment can only be done through stored procedure.