|  |
| --- |
| **National University of Computer and Emerging Sciences** |
| Lab Manual 7  “Triggers” |
|  |
| Database Systems |
| Spring 2018 |

Department of Computer Science

FAST-NU, Lahore, Pakistan

**Table of Contents**

[1. Objective 2](#_Toc475651203)

[2. Prerequisites 2](#_Toc475651204)

[3. Task Distribution 2](#_Toc475651205)

[4. Triggers 3](#_Toc475651206)

[5. DML triggers 3](#_Toc475651207)

[AFTER and INSTEAD OF Triggers: 4](#_Toc475651208)

[Syntax to create trigger: 6](#_Toc475651209)

[How Get The Effect Rows inside DML In Trigger: 6](#_Toc475651210)

[ALTER/DROP/ENABLE/DISABLE TRIGGER 10](#_Toc475651211)

[6. DDL Triggers 11](#_Toc475651212)

[SYNTAX OF DDL 12](#_Toc475651213)

# Objective

The purpose of this document is to give a practical implementation of Trigger. What are triggers? Why are they used? What type of triggers are after instead for triggers.

# Prerequisites

* All the previous lab manuals
* Chapter 5 lesson 3, from MCTS 70-433 SQLServer 2008 Database Development.
* Trigger from Elmasri

# Task Distribution

|  |  |
| --- | --- |
| Total Time | 170 Minutes |
| Introduction + Examples | 70 Minutes |
| Exercise | 80 Minutes |
| Evaluation | 20 Minutes |

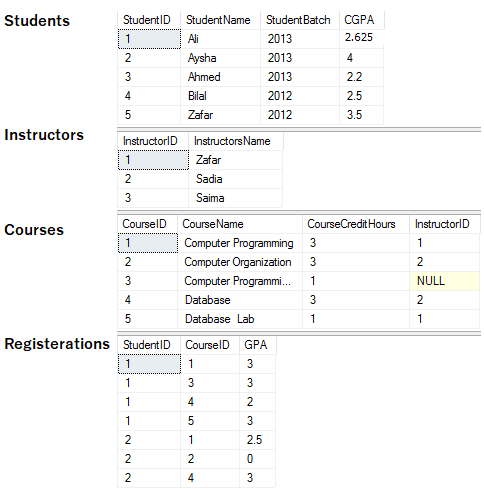
# Triggers

Triggers are special kind of stored procedures that automatically execute when a DML or DDL statement associated with the trigger is executed. Each trigger will be associated with one DML or DDL statement. Unlike stored procedure triggers cannot be executed directly by application/user, they will ONLY be executed by DBMS in reaction to DML or DDL statement with which the trigger was associated.

Triggers can be divided in two categories depending on the type of statement they are associated with as follow:

* DML triggers
* DDL triggers

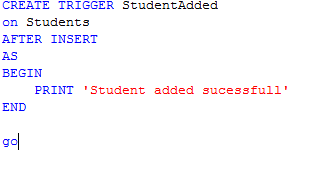
We will be using the student schema of previous lab for examples:



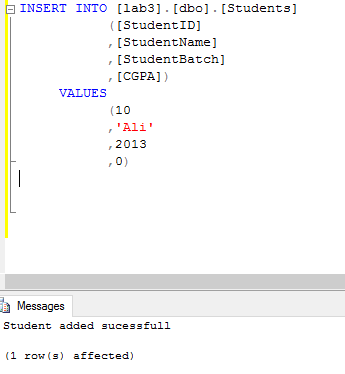
# DML triggers

DML is data modification language and it includes INSERT, UPDATE, and DELETE operation on table or View. DML triggers are against these DML statements on specific table.

*For example if you want to print a message “Student added successfully” whenever a student is added in student table, you can create a trigger, that will run after every INSERT in STUDENTS table as follow:*



This trigger is now associated with INSERT into STUDENT statement, after every insertion in Student table this trigger will be fired.



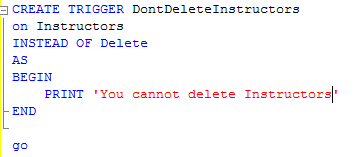
NOTE: The trigger and the statement that fires it are treated as a single transaction.

## AFTER and INSTEAD OF Triggers:

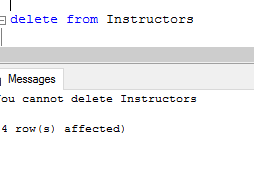
The above trigger was fired only AFTER the INSERT into STUDENT statement was successfully executed. These type of triggers are called AFTER trigger.

Another type of trigger is INSTEAD OF trigger, this type of DML trigger is executed instead of the executing that SQL statement associated with it, therefore, overriding the actions of the triggering statements

*Example: You have to create a trigger that stops everyone from deleting Instructors such that f someone tries to delete Instructors it should print “you cannot delete Instructors” instead of deleting an Instructors.*



Now try deleting an Instructor



Is any row deleted from Instructors table?

## Syntax to create trigger:

General syntax to create trigger is

CREATE TRIGGER *trigger\_name*

ON < *table* | *view* >

< AFTER | INSTEAD OF >

< [ INSERT ] [ **,** ] [ UPDATE ] [ **,** ] [ DELETE ] >

AS

BEGIN

*<sql\_statement*s>

END

NOTE: See MSSQL dynamic help for more details

## How Get The Effect Rows inside DML In Trigger:

The triggers we have seen above are simple one, what if you want the value of effect rows from DML and use them in triggers.

*Example: Whenever an Instructor is inserted in, it should automatically convert that name of that instructor in Upper Case.*

*For example: Instructor with Name “ali ahmed” should be inserted as “ALI AHMED”*

For that we use special table “DELETED” and “INSERTED” designed for DML triggers.

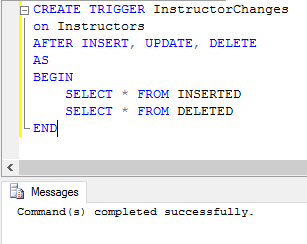
DML triggers use the **deleted** and **inserted** logical (conceptual) tables. They are structurally similar to the table on which the trigger is defined, that is, the table on which the user action is tried. The **deleted** and **inserted** tables hold the old values or new values of the rows that may be changed by the DML action.

More details

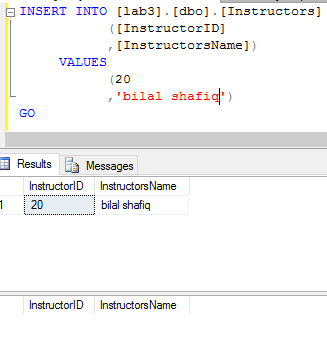
<http://technet.microsoft.com/en-us/library/ms191300.aspx>

NOTE: These tables are only accessible in triggers

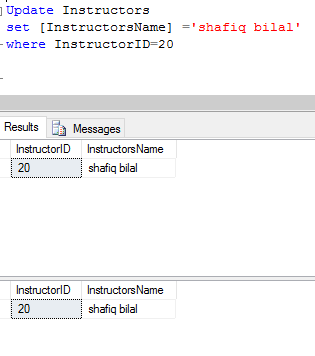
TRY THIS: Create a triggers of Instructor table that selects inserted and deleted table, whenever insert, delete or update is performed.



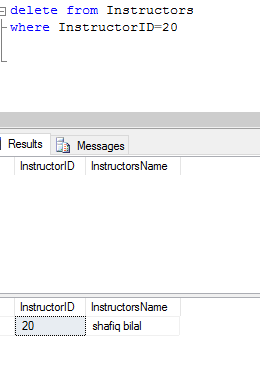
Now insert a row in instructor:



Now update a row from table



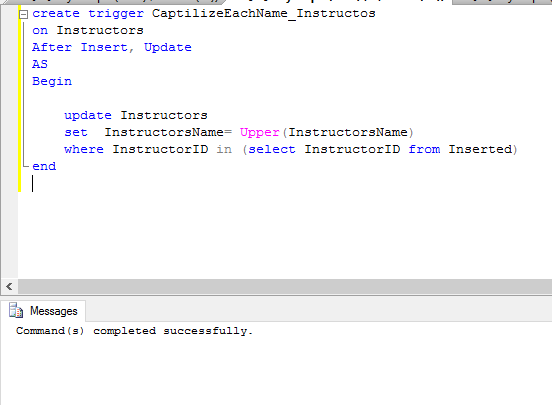
Now delete a row from instructor table



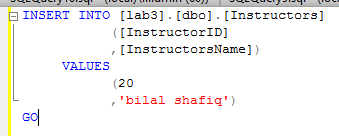
Now you can use the inserted and deleted table to solve the following problem

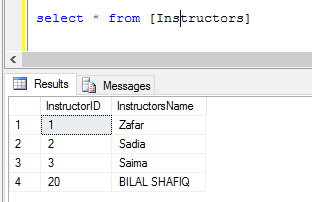
*Whenever an Instructor is inserted in, it should automatically convert that name of that instructor in Upper Case.*

*For example: Instructor with Name “ali ahmed” should be inserted as “ALI AHMED”*

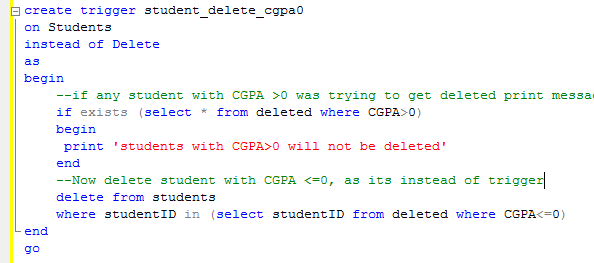


Try inserting a new instructor and see the results.





TRY THIS: *Create a trigger that if a student with CGPA > 0 is deleted it should now allow to delete it, instead it should print, ‘Deleting a student with CGPA>0 is not allowed’.*



Now see the effect of this

delete from students

NOTE: YOU CAN ONLY CREATE ONE INSTEAD OF AND ONLY ON AFTER TRIGGER ON INSERT/DELETE/UPDATE ON ONE TABLE

## ALTER/DROP/ENABLE/DISABLE TRIGGER

Use alter to change the body or type of trigger. Disable trigger will stop the execution of that trigger and enable trigger will enable execution of disabled trigger. Drop will delete trigger

ALTER <TriggerName>

On <view/table>

After/Instead of <insert/update/delete>

As

begin

<Body>

end

Drop trigger <TriggerName>

Enable trigger <TriggerName> on <ObjectName>

Disable trigger <TriggerName> on <ObjectName>

\*\*NOTE: OBJECT NAME IS THE OBJECT ON WHICH YOU CREATED THE TRIGGER

Try these for triggers you created.

# DDL Triggers

DDL triggers, fire in response to a DDL statement to which they are associated. DDL event primarily correspond to SQL statements that start with the keywords CREATE, ALTER, and DROP. These triggers are current databases.

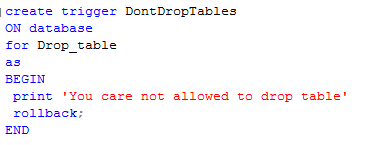
There triggers are also of two types, FOR and AFTER, first one executes instead of the DDL statement it is associated with and second one executes after the DDL statement, it is associate with is successfully executed.

(For in DML FOR is same as Instead of in DDL)

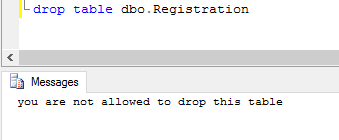
Use DDL triggers when you want to do the following:

* You want to prevent certain changes to your database schema.
* You want something to occur in the database in response to a change in your database schema.
* You want to record changes or events in the database schema.

Example: Create a trigger on your database that instead of dropping a table prints ‘you are not allowed to drop this table’



This trigger is create on ‘lab3’ database and will execute for every DROP table DDL statement



## Syntax Of DDL

CREATE TRIGGER *trigger\_name*

ON DATABASE

{ FOR | AFTER } { *event\_type*} [ **,**...*n* ]

AS

Begin

<Body>

End

Some examples of event\_type are

Drop\_table

Create\_table

Alter\_table

More information on: <https://msdn.microsoft.com/en-us/library/bb522542.aspx>