

Triggers

for EFP0700 Database Servers



Alexey Tukalo,
EFA12SF,
Information Technology,
Savonia University of Applied Sciences

November 16, 2014

1 Introduction

At the first step I have added the new columns to the tables thought GUI and I also set the default value for user columns via Script below.

```
1 ALTER TABLE Project
2 ADD DEFAULT(CURRENT_USER) FOR [User]
```

To complete all the instructions I decided to write three triggers for every of the three tables.

2 The triggers

2.1 deleteEmp

I have started from deleteEmp trigger. The trigger have to prevent any tries to delete row from the table and put the time of the tries into the Removed column on the table. The trigger uses the INSTEAD OF command to fire itself instead of delete command.

```
1 USE [test_db]
2 GO
3 /***** Object: Trigger [dbo].[deleteEmp]    Script Date: 16.11.2014 16:27:15 *****/
4 SET ANSI_NULLS ON
5 GO
6 SET QUOTED_IDENTIFIER ON
7 GO
8
9 ALTER TRIGGER [dbo].[deleteEmp] ON [dbo].[employee]
10 INSTEAD OF DELETE
11 AS
12 BEGIN
13     UPDATE employee SET Removed=GETDATE() WHERE employee.emp_no IN (SELECT emp_no FROM deleted)
14 END
```

We also use deleted table to get the reference to the rows which are desired to be deleted.

2.2 insertEmp

The trigger has to fill the User and Timestamp columns by current user's name and by current time when new row is created. The trigger is very similar with previous one, but it uses the inserted table, which contain the information about recently inserted row.

```
1 USE [test_db]
2 GO
3 /***** Object: Trigger [dbo].[insertEmp]    Script Date: 16.11.2014 16:27:17 *****/
4 SET ANSI_NULLS ON
5 GO
6 SET QUOTED_IDENTIFIER ON
7 GO
8
9 ALTER TRIGGER [dbo].[insertEmp] ON [dbo].[employee]
10 AFTER INSERT
11 AS
12 BEGIN
13     UPDATE Employee SET [User]= CURRENT_USER, Timestamp=GETDATE() WHERE employee.emp_no=(SELECT emp_no FROM inserted)
14 END
```

The trigger should be fired after insert and it is possible to make via command AFTER.

2.3 updateEmp

The updateEmp trigger has very same code with insertEmp. It also uses inserted table, because Update actually works like Insert faired after Delete that's why actually in most cases you can use any of the virtual tables.

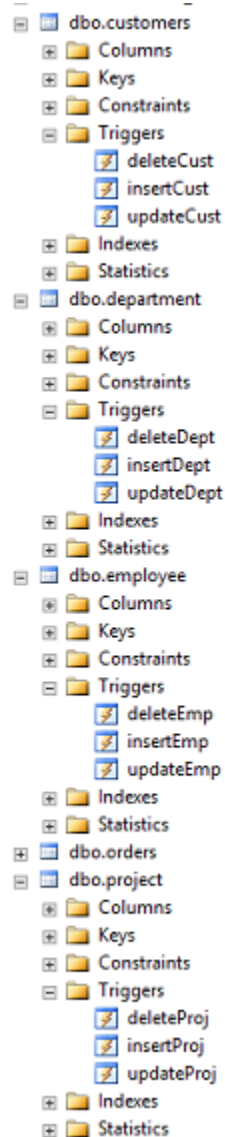
```

1 USE [test_db]
2 GO
3 /***** Object: Trigger [dbo].[updateEmp]    Script Date: 16.11.2014 16:27:20 *****/
4 SET ANSI_NULLS ON
5 GO
6 SET QUOTED_IDENTIFIER ON
7 GO
8
9 ALTER TRIGGER [dbo].[updateEmp] ON [dbo].[employee]
10 AFTER UPDATE
11 AS
12 BEGIN
13 UPDATE Employee SET [User]= CURRENT_USER, Timestamp=GETDATE() WHERE employee.emp_no IN (SELECT emp_no FROM inserted)
14 END

```

2.4 Other triggers

All other triggers can be written via replacing of the triggers names, table names, and primary keys, all the stuff are marked by yellow on the source code of the first three functions.



3 Results

You can see the results of the work of the triggers on the customers and employee tables below.

customerid	companyname	contactname	address	town	phone	fax	Timestamp	Removed	User
1	2	3 ...	5	2	1	2	2014-11-16 14:2...	2014-11-16 14:2...	dbo
NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

emp_no	emp_fname	emp_lname	dept_no	User	Timestamp	Removed
1	Daria	Doncova	1	NULL	NULL	NULL
2	Sergey	Lukanov	2	NULL	NULL	NULL
3	Artur	Doile	3	dbo	2014-11-16 14:0...	2014-11-16 14:0...
4	Leo	Zig	4	dbo	2014-11-16 14:0...	NULL
5	Josev	Stalin	5	NULL	NULL	NULL
6	Max	Pain	6	dbo	2014-11-15 18:1...	NULL
9	Mark	Kolin	9	dbo	2014-11-16 14:0...	2014-11-16 14:0...
NULL	NULL	NULL	NULL	NULL	NULL	NULL