## Murach Chapter 15 How to Code Triggers

**1:**

Run the attached script file, "trigger tables.sql", and create a single trigger that will insert rows into the table dd.paytrack. Each time a pledge payment is added, changed, or removed to the table dd.payment the username (logon), current date, action taken (INSERT, UPDATE, or DELETED), and the ipay value for the payment record should be added to the dd.paytrack table. Use the following for your primary key in the dd.paytrack table:  
idtrack INT NOT NULL IDENTITY PRIMARY KEY

Use the following three DML statement to test your trigger:  
INSERT INTO dd.payment VALUES (1479,112,20,'01-JUN-2013','CC');

UPDATE dd.payment SET paymethod = 'YY' WHERE idpay = 1479;

DELETE FROM dd.payment WHERE idpay = 1479;

Use the following two queries to show and test your work:

select \* from dd.Payment where idpay > (SELECT MAX(idpay) - 10 FROM dd.Payment)

select \* from dd.paytrack

Trigger Here

CREATE TRIGGER payTrack\_trick

on dd.payment

after insert, update, DELETE

AS

BEGIN

declare @IDpay INT, @activity VARCHAR(20)

if exists (select \* from inserted) and exists (select \* from deleted)

BEGIN

SET @activity = 'updated'

SELECT @IDpay = idpay from inserted i;

INSERT into paytrack(ptuser, ptdate, ptaction, ptpayid) VALUES(

CURRENT\_USER, GETDATE(), @activity, @IDpay )

END

if exists (select \* from inserted) and not exists( select \* from deleted)

BEGIN

SET @activity ='insert'

SELECT @IDpay = idpay from inserted i;

Insert into paytrack(ptuser, ptdate, ptaction, ptpayid) VALUES(

CURRENT\_USER, GETDATE(), @activity, @IDpay )

END

if exists (select \* from deleted) and not exists(select \* from inserted)

BEGIN

SET @activity = 'delete'

SELECT @IDpay = idpay from deleted i;

INSERT into paytrack(ptuser, ptdate, ptaction, ptpayid) VALUES(

CURRENT\_USER, GETDATE(), @activity, @IDpay )

END

END

1 Insert

Table

Description automatically generated

Table

Description automatically generated

2 Update

Table

Description automatically generated

Table

Description automatically generated

3. Delete

Table

Description automatically generated

Table

Description automatically generated

**2:**

Create a trigger to solve the following problem. In a previous assignment you created the following view:  
create or alter view num\_3

as select s.student\_id, last\_name, first\_name, se.section\_id, description

from db2.hw.student s

join db2.hw.enrollment e on s.student\_id = e.STUDENT\_ID

join db2.hw.section se on e.section\_id=se.SECTION\_ID  
join db2.hw.course c on se.COURSE\_NO=c.COURSE\_NO;

After you created the view you run the following insert command:  
INSERT INTO db2.dbo.NUM\_3 (student\_id, last\_name, first\_name, section\_id, description) values('500', 'Professor', 'Gabor', '80', 'Intro to Programming');

You received an error message:  
Msg 4405, Level 16, State 1, Line 8

View or function 'db1.dbo.NUM\_3' is not updatable because the modification affects multiple base tables.

*Just write the trigger so the student\_id, last\_name, & first\_name are inserted correctly!*

Trigger Here

CREATE TRIGGER trackStudent on num\_3

INSTEAD OF INSERT

AS

BEGIN

insert into hw.STUDENT(STUDENT\_ID, LAST\_NAME,FIRST\_NAME)

SELECT student\_ID, last\_name, first\_name from inserted

END

Graphical user interface, text, application, email

Description automatically generated

**3:**

Create trigger that will prevent the deletion or altering of a table on the database db2. Note: DO NOT code this to eliminate the error for FKs. To test your work use the below two tables.

use db2

go

DROP TABLE dbo.BOOKCAST\_AUTHOR

use db2

go

ALTER TABLE dbo.BOOKCAST\_AUTHOR ADD DOB DATETIME

Answer:

Trigger Here

create TRIGGER PreventDelete

on database

for drop\_table, ALTER\_table

AS

BEGIN

RAISERROR('Disable trigger DDLTriggr to drop or alter tables',0,0);

ROLLBACK

END

Graphical user interface, text, application, email

Description automatically generated

Graphical user interface, text, application, email

Description automatically generated