Wide Array of Server, Storage & HPC Solutions Powered by the Latest Intel® Xeon® Processors
Call Us Today
SILICON MECHANICS SERVERS • STORAGE • HPC

Home | Weblogs | Forums | SQL Server Links
Search: Go

Active Forum Topics | Popular Articles | All Articles by Tag | SQL Server Books | About

Site Sponsored By: **TraceTune.com** - The online version of ClearTrace. Quickly identify which SQL statements use the most CPU and disk.

An Introduction to Triggers -- Part II

By Garth Wells on **4 November 2001** | 1 Comment | Tags: Triggers

4
g+1

Article Series Navigation: An Introduction to Triggers -- Part II ▼

A few months ago I wrote an article for SQLTeam called An Introduction to Triggers -- Part I. The article covered trigger fundamentals--the most important being the way the inserted and deleted virtual tables work. I also showed an example of how to use an INSERT trigger to log activity on a web site. In this article I want to show how to use an UPDATE and DELETE trigger and introduce INSTEAD OF triggers. If you are not familiar with the inserted and deleted tables please read this article before continuing.

UPDATE Trigger

An UPDATE trigger is used to perform an action after an update is made on a table. The example shown here is based on the table used in Part I of this article.

```
CREATE TRIGGER tr_Orders_UPDATE
ON Orders
AFTER UPDATE
AS

--Make sure Priority was changed
IF NOT UPDATE(Ord_Priority)
 RETURN

--Determine if Priority was changed to high
IF EXISTS (SELECT *
           FROM inserted a
           JOIN deleted b ON a.Ord_ID=b.Ord_ID
           WHERE b.Ord_Priority <> 'High' AND
           a.Ord_Priority = 'High')
 BEGIN
  DECLARE @Count tinyint
  SET @Count = (SELECT COUNT(*)
                FROM inserted a
                JOIN deleted b ON a.Ord_ID=b.Ord_ID
                WHERE b.Ord_Priority <> 'High' AND
                a.Ord_Priority = 'High')
  PRINT CAST(@Count as varchar(3))+' row(s) where changed to a priority of High'
 END
go
```

In Part I the INSERT trigger *watched* for orders with a priority of 'High.' The UPDATE trigger watches for orders whose priority are changed from something else to High.

The IF statement checks to see if the Ord_Priority value was changed. If not we can save some time by exiting the trigger immediately.

The deleted table holds the pre-UPDATE values and inserted table holds the new values. When the tables are joined it is easy to tell when the priority changes from something else to High.

DELETE Trigger

Once you understand how an UPDATE trigger works a DELETE trigger is easy to implement. In the example shown here it simply counts the number of rows in the deleted table to see how many had a priority of high.

```
CREATE TRIGGER tr_Orders_DELETE
ON Orders
AFTER DELETE
AS
```

Subscribe to SQLTeam.com

Weekly SQL Server newsletter with articles, forum posts, and blog posts via email. Subscribers receive our **white paper with performance tips for developers**.

Subscribe

SQLTeam.com Articles via RSS

SQLTeam.com Weblog via RSS

- Advertisement -

Microsoft Azure
SISENSE
Business analytics for complex data on Microsoft Azure
Learn more ▶▶▶

Resources

SQL Server Resources

Advertise on SQLTeam.com

SQL Server Books

SQLTeam.com Newsletter

Contact Us

About the Site

```
--Determine if Order with a Priority of High was deleted
IF EXISTS (SELECT * FROM deleted WHERE Ord_Priority = 'High')
 BEGIN
  DECLARE @Count tinyint
  SET @Count = (SELECT * FROM deleted WHERE Ord_Priority = 'High')
  PRINT CAST(@Count as varchar(3))+' row(s) where deleted whose priority was High'
 END
go
```

INSTEAD OF Triggers

INSTEAD OF triggers are new to SQL Server 2000. The main reason they were introduced is to facilitate updating Views. I am going to show you how one works, but I am not going to bother with updating a View. Quite frankly I think updating Views is poor application design. I was once told by one of my editors that some of my *views* on Views reflect my "insular" experience, so if you do not agree with me on this point rest assured you are not alone.

An INSTEAD OF Trigger is used to perform an action *instead of* the one that caused the trigger to be fired. This sounds like double-talk, so I will explain a little more. Let's say you have an INSTEAD OF INSERT trigger defined on a table and an INSERT is executed. A row **is not** added to the table, but the code in the trigger **is** fired. The following shows what this looks like.

```
CREATE TRIGGER tr_Orders_INSERT_InsteadOf
ON Orders
INSTEAD OF INSERT
AS
PRINT 'Updateable Views are Messy'
go
```

The following INSERT produces the message shown, but the row is not added to the table.

```
INSERT Orders (Ord_Priority) VALUES ('High')

-- Results --

Updateable Views are Messy
```

Feel free to experiment with this type of trigger, but unless you are trying to update a View that is based on multiple tables I do not think it will be of much practical value.

Need More Information on Views?

If you would like more information on Views you can read a free chapter I have listed on my web site. Go to www.SQLBook.com and click on the Sample Chapters section. After reading the chapter you will really know how I feel about this pathetic excuse for a database object (just kidding).

< Previous: An Introduction to Triggers -- Part I | Next >

Discuss this article: **1 Comment** so far. Print this Article.

g+1 4

If you like this article you can sign up for our **weekly newsletter**. There's an opt-out link at the bottom of each newsletter so it's easy to unsubscribe at any time.

Email Address: Subscribe

Delicious

Related Articles

Using DDL Triggers in SQL Server 2005 to Capture Schema Changes (13 August 2007)

Audit Triggers for SQL Server (8 May 2002)

Code to find out the statement that caused the trigger to fire! (16 April 2002)

An Introduction to Triggers -- Part I (30 April 2001)

Other Recent Forum Posts

Cost Tracker in SSRS 08 (1 Reply)

UDFs GETWORDCOUNT, GETWORDNUM (24 Replies)

SSRS need help using inscope with a matrix (2 Replies)

Replicating Triggers (14 August 2000)

only alphabat show (2 Replies)

show changes from and changes to in a history tabl (4 Replies)

Deleted Database (6 Replies)

WHERE clause statement problem (4 Replies)

Detect LDF corruption (4 Replies)