

Re: Help in writing a trigger

Source:

<http://www.tech-archive.net/Archive/SQL-Server/microsoft.public.sqlserver.programming/2009-05/msg01446.html>

- *From:* SqlBeginner <SqlBeginner@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx>
 - *Date:* Sun, 24 May 2009 07:27:01 -0700
-

Thanks for the script John. Instead of TEMP I made use of TABLE variable.
Hope this is what you meant.

```
ALTER TRIGGER sampleTrigger ON dbo.tSource
FOR INSERT
AS
BEGIN TRY
```

```
--Table Variable to stored the content within Inserted Table
```

```
Declare @tSource_Inserted TABLE
```

```
(
[SID] int not null,
pcatid char(5) null,
pname varchar(50) null,
cost varchar(50) null,
dtCreateDate datetime null
)
```

```
Insert into @tSource_Inserted Select * from INSERTED
```

```
--Check for the existance of master record
```

```
INSERT INTO dbo.tblPK ( cPID, intPID, strPName )
SELECT LEFT(I.pcatid,1),CAST(SUBSTRING(I.pcatid,2,4) AS smallint),
MAX(I.pname) FROM @tSource_Inserted I
WHERE NOT EXISTS (
SELECT * FROM dbo.tblPK P
WHERE P.cPID = LEFT(I.pcatid,1)
AND P.intPID = CAST(SUBSTRING(I.pcatid,2,4) AS smallint)
)
GROUP BY LEFT(I.pcatid,1), CAST(SUBSTRING(I.pcatid,2,4) AS smallint)
```

```
-- Update Existing
```

```
UPDATE F SET cost=I.cost FROM dbo.tblFromTrigger F JOIN @tSource_Inserted I
ON I.[SID] = F.[SID]
```

```
--Check whether SID already present. If not present insert
```

```
INSERT INTO dbo.tblFromTrigger ( [SID], cPID, intPID, cost,
dtCreateTimeStamp )
```

Re: Help in writing a trigger

```
SELECT I.[SID], LEFT(I.pcatid,1), CAST(SUBSTRING(I.pcatid,2,4) AS smallint),  
I.cost, I.dtCreateDate FROM @tSource_Inserted I  
WHERE NOT EXISTS ( SELECT * FROM dbo.tblFromTrigger F WHERE F.[SID] =  
I.[SID] )  
  
END TRY  
BEGIN CATCH  
Print ('Do error handling or logging here.')END CATCH  
GO
```

Regards
Pradeep

"John Bell" wrote:

"SqlBeginner" <SqlBeginner@xxxxxxxxxxxxxxxxxxxxxxxxxxxx> wrote in message
news:786D12FE-AE29-4B18-8602-619C236B9C8A@xxxxxxxxxxxxxxxxxxxx

This is the script I came up after going through some trigger examples in
internet. I am not too sure whether I have written it the optimized way.

Any
suggestions?

```
Alter trigger sampleTrigger On dbo.tSource  
For INSERT  
As  
Begin Try  
--As per your inserting something and it throws an error  
Declare @SID int  
Declare @pcatid char(5)  
Declare @pname varchar(50)  
Declare @cost varchar(50)  
Declare @dtCreateDate datetime  
  
Select @SID = SID, @pcatID=pcatid, @pname=pname, @cost=cost,  
@dtCreateDate=dtCreateDate FROM INSERTED I  
  
--Check for the existance of master record  
if not exists (select cPID, intPID from dbo.tblPK where cPID  
=left(@pcatid,1) and intPID=substring(@pcatid,2,4))  
Insert into dbo.tblPK values  
(left(@pcatid,1),substring(@pcatid,2,4),@pname)  
  
--Check whether @SID already present. If present update the record else  
Insert it  
if not exists (select SID from dbo.tblFromTrigger where SID =@SID)  
Insert into dbo.tblFromTrigger values (@SID, left(@pcatid,1),  
Substring(@pcatid,2,4), @cost, @dtcreatedate)
```

Re: Help in writing a trigger

```
else
Update dbo.tblFromTrigger Set cost=@cost where SID=@SID
End Try
Begin Catch
Print ('Do error handling or logging here.')
End Catch
Go
```

Regards
Pradeep

"Erland Sommarskog" wrote:

SqlBeginner (SqlBeginner@xxxxxxxxxxxxxxxxxxxxxxxxxxxxxx)
writes:

Thanks for the response Erland. Is it possible to help me with the script for this. I have not written a trigger in past :(More over we are talking abt 1 million transactions per day.

Writing a trigger is not that different from writing a stored procedure, but you need to take some care.

In a trigger you have access to two virtual tables, "inserted" and "deleted". "inserted" holds the inserted rows for an INSERT statement and the after-image of the updated rows in an UPDATE statement. "deleted" holds the deleted rows in a DELETE, and the before-image for an UPDATE.

These tables have the same schema as the base table. You can only access them from the trigger, not from stored procedures you call or from dynamic SQL.

It's important to understand that triggers fires once per statement. It's a common error to read variables from inserted/deleted into variables.

Re: Help in writing a trigger

In a trigger, XACT_ABORT is ON by default, meaning that any error terminates the batch, and if you use TRY-CATCH, the transaction will be doomed. And there is always a transaction. If there is no user-defined transaction, there is the transaction defined by the statement that fired the trigger.

As for performance, plans with "inserted"/"deleted" do not always behave that well, although it may be better in SQL 2005, when triggers are implemented with help of the row-versioning technology. Sometimes it may be better to copy the rows into a table variable or temp table.

In your case the trigger is on the subscriber side of a replication scheme. What implications that has, I don't know, as I have worked very little with replication.

—
Erland Sommarskog, SQL Server MVP,
esquel@xxxxxxxxxxxxx

Links for SQL Server Books Online:

SQL 2008:

<http://msdn.microsoft.com/en-us/sqlserver/cc514207.aspx>

SQL 2005:

<http://msdn.microsoft.com/en-us/sqlserver/bb895970.aspx>

SQL 2000:

<http://www.microsoft.com/sql/prodinfo/previousversions/books.msp>

Your code assumes that only one row will be in the inserted table, this may not be the case as a INSERT statement can be used to insert many rows (even if the application doesn't).

This is untested and without moving data from INSERTED into another temp table, you should test this method and one that uses a temp table if it proves to be slow:

```
ALTER TRIGGER sampleTrigger ON dbo.tSource
FOR INSERT
AS
BEGIN TRY
```

Re: Help in writing a trigger

--Check for the existance of master record

```
INSERT INTO dbo.tblPK ( cPID, intPID, strPName )
SELECT LEFT(I.pcatid,1),CAST(SUBSTRING(I.pcatid,2,4) AS smallint),
MAX(I.pname)
FROM INSERTED I
WHERE NOT EXISTS ( SELECT * FROM dbo.tblPK P
WHERE P.cPID = LEFT(I.pcatid,1)
AND P.intPID = CAST(SUBSTRING(I.pcatid,2,4) AS smallint) )
GROUP BY LEFT(I.pcatid,1), CAST(SUBSTRING(I.pcatid,2,4) AS smallint)
```

-- Update Existing

```
UPDATE F
SET cost=I.cost
FROM dbo.tblFromTrigger F
JOIN INSERTED I ON I.[SID] = F.[SID]
```

--Check whether SID already present. If not present insert

```
INSERT INTO dbo.tblFromTrigger ( [SID], cPID, intPID, cost,
dtCreateTimeStamp )
SELECT I.[SID], LEFT(I.pcatid,1), CAST(SUBSTRING(I.pcatid,2,4) AS smallint),
I.cost, I.dtCreateDate
FROM INSERTED I
WHERE NOT EXISTS ( SELECT * FROM dbo.tblFromTrigger F WHERE F.[SID] =
I.[SID] )
```

```
END TRY
BEGIN CATCH
Print ('Do error handling or logging here.')
END CATCH
GO
```

John