عنوان: **معرفی و استفاده از DDL Triggers در SQL Server**
نویسنده: محمد رجبی
تاریخ: ۱۳۹۳/۰۲/۰۳ ۱۱:۵۰
آدرس: www.dotnettips.info
گروه‌ها: SQL Server, Database

استفاده از DDL Trigger امکان ایجاد Trigger برای عملیات DDL(Data Definition Language) از SQL Server 2005 فراهم گردید. عملیاتی مانند ایجاد یک جدول جدید در بانک اطلاعاتی، اضافه شدن یک Login جدید و یا ایجاد یک بانک اطلاعاتی جدید را به وسیله این نوع Triggerها می‌توان کنترل نمود. در حقیقت DDL Trigger به شما اجازه می‌دهد که از تاثیر تعدادی از دستورات DDL جلوگیری کنید. بدین ترتیب که تقریباً هر دستور DDL به طور خودکار، تراکنشی (Transactional) اجرا می‌شود . می‌توان با دستور ROLLBACK TRANSACTION اجرای دستور DDL را لغو نمود. توجه شود همه دستورات DDL به صورت تراکنشی اجرا نمی‌شوند، به عنوان مثال دستور ALTER DATABASE ممکن است Database را تغییر دهد. در این صورت ساختار فایلی Database را تغییر می‌دهد، از آنجائی که سیستم عامل ویندوز به صورت تراکنشی عمل نمی‌کند بنابراین شما نمی‌توانید این عمل فایل سیستمی را لغو نمائید. به هر حال شما می‌توانید Trigger را با ALTER DATABASE فعال (fire) کنید برای عملیات Auditing، ولی نمی‌توان از انجام عمل ALTER DATABASE جلوگیری کرد.
برای نمونه می‌خواهیم از حذف و یا تغییر جداول یک بانک اطلاعاتی که به صورت عملیاتی در حال سرویس دهی است جلوگیری کنیم، برای اینکار از دستورهای زیر استفاده می‌کنیم:

```
create trigger Prevent_AlterDrop
on database
for drop_table, alter_table
as
  print 'table can not be dropped or altered'
  rollback transaction
```

از عبارت ON برای مشخص کردن محدوده Trigger در سطح SQL Instance (در این صورت ON All SERVER نوشته می‌شود) و یا در سطح Database (در این حالت ON DATABASE نوشته می‌شود) استفاده می‌شود و از عبارت FOR برای مشخص کردن رویداد یا گروه رویدادی که سبب فراخوانی Trigger می‌شود، استفاده خواهد شد.

1- معرفی تابع EVENTDATA() این تابع، یک تابع سیستمی مهم است که در DDL Trigger استفاده می‌شود. در حالیکه DDL Trigger در هر سطحی فعال (fire) شود تابع سیستمی EVENTDATA() فراخوانی (raise) می‌شود. خروجی تابع در قالب XML است . می‌توان اطلاعات را از تابع EVENTDATA دریافت کرد و آنها را در یک جدول با فیلدی از جنس XML و یا با استفاده از XPath Query ثبت کرد (Logging). عناصر کلیدی (Key Elements) تابع EVENTDATA به شرح زیر است:

- **EventType:** نوع رویدادی که باعث فراخوانی Trigger شده است.
- **PostTime:** زمانی که رویداد رخ می‌دهد.
- **SPID:** SPID کاربری که باعث ایجاد رویداد شده است.
- **ServerName:** نام SQL Instance که رویداد در آن رخ داده است.
- **LoginName:** نام Login که عمل مربوط به وقوع رویداد را اجرا می‌کند.
- **UserName:** نام User که عمل مربوط به وقوع رویداد را اجرا می‌کند.
- **DatabaseName:** نام Database که رویداد در آن رخ می‌دهد.
- **ObjectType:** نوع Object که اصلاح، حذف و یا ایجاد شده است.
- **ObjectName:** نام Object که اصلاح، حذف و یا ایجاد شده است.
- **TSQLCommand:** دستور T-SQL که اجرا شده و باعث اجرا شدن Trigger شده است.

2- بررسی یک سناریو نمونه
برای نمونه در دستورات زیر جدولی با نام ddl_log

```
CREATE TABLE ddl_log
  (
     EventType nvarchar(100),
 PostTime datetime,
```

```
 SPID nvarchar(100),
 ServerName nvarchar(100),
 LoginName nvarchar(100),
 UserName  nvarchar(100),
 DatabaseName  nvarchar(100),
 ObjectName  nvarchar(100),
 ObjectType  nvarchar(100),
 DefaultSchema nvarchar(100),
 [SID]  nvarchar(100),
 TSQLCommand nvarchar(2000));
```

و یک Trigger با نام log برای رویدادهایی که در سطح Database رخ می‌دهد، ایجاد می‌کنیم.

```
CREATE TRIGGER [Log] ON DATABASE
 FOR DDL_DATABASE_LEVEL_EVENTS
 AS
 DECLARE @data XML
 SET @data = EVENTDATA()
 INSERT INTO ddl_log
 VALUES (
   @data.value('(/EVENT_INSTANCE/EventType)[1]',   'nvarchar(100)'),
   @data.value('(/EVENT_INSTANCE/PostTime)[1]',   'datetime'),
   @data.value('(/EVENT_INSTANCE/SPID)[1]',   'nvarchar(100)'),
   @data.value('(/EVENT_INSTANCE/ServerName)[1]',   'nvarchar(100)'),
   @data.value('(/EVENT_INSTANCE/LoginName)[1]',   'nvarchar(100)'),
   @data.value('(/EVENT_INSTANCE/UserName)[1]',   'nvarchar(100)'),
   @data.value('(/EVENT_INSTANCE/DatabaseName)[1]',   'nvarchar(100)'),
   @data.value('(/EVENT_INSTANCE/ObjectName)[1]',   'nvarchar(100)'),
   @data.value('(/EVENT_INSTANCE/ObjectType)[1]',   'nvarchar(100)'),
   @data.value('(/EVENT_INSTANCE/DefaultSchema)[1]',   'nvarchar(100)'),
   @data.value('(/EVENT_INSTANCE/SID)[1]',   'nvarchar(100)'),
   @data.value('(/EVENT_INSTANCE/TSQLCommand)[1]',   'nvarchar(2000)'));
```

نمونه ای از مقادیر ذخیره شده در جدول ddl_log به شکل زیر خواهد بود:

EventType	PostTime	SPID	ServerName	LoginName	UserName	DatabaseName	ObjectName
CREATE_TABLE	11/7/2010 10:36:26 AM	55	1936-603102405\SQL2008	sa	dbo	Test	Tmp_TestTable
ALTER_TABLE	11/7/2010 10:36:26 AM	55	1936-603102405\SQL2008	sa	dbo	Test	Tmp_TestTable
DROP_TABLE	11/7/2010 10:36:26 AM	55	1936-603102405\SQL2008	sa	dbo	Test	TestTable
RENAME	11/7/2010 10:36:26 AM	55	1936-603102405\SQL2008	sa	dbo	Test	Tmp_TestTable
DROP_TABLE	11/7/2010 10:38:26 AM	52	1936-603102405\SQL2008	sa	dbo	Test	TestTable

DefaultSchema	SID	TSQLCommand
NULL	NULL	CREATE TABLE dbo.Tmp_TestTable (a nvarchar(50) NULL) ON [PRIMARY]
NULL	NULL	ALTER TABLE dbo.Tmp_TestTable SET (LOCK_ESCALATION = TABLE)
NULL	NULL	DROP TABLE dbo.TestTable
NULL	NULL	EXECUTE sp_rename N'dbo.Tmp_TestTable', N'TestTable', 'OBJECT'
NULL	NULL	DROP TABLE TestTable ;

3- ملاحظات

در صورت فعال شدن Trigger می‌توان برخی موارد مانند محدودیت زمانی، کاربر اجرا کننده و ... را اضافه نمود. برای مثال در دستور زیر اجازه تغییرات در این زمان (بین 7:00 A.M تا 8:00 P.M) امکان پذیر نیست و در صورت اقدام پیغام خطا دریافت می‌کنید و دستورات Create لغو خواهند شد و اگر خارج از زمان فوق دستورات DDL را اجرا کنید دستورات به طور موفقیت آمیز اجرا می‌شود و البته تغییرات نیز Log می‌شوند.

```
IF DATEPART(hh,GETDATE()) > 7 AND DATEPART(hh,GETDATE()) < 20
BEGIN
    RAISERROR ('You can only perform this change between 8PM and 7AM. Please tr
    this change again or contact Production support for an override.', 16, 1)
ROLLBACK
END
```

این Trigger تاثیرات کمی بر روی کارایی دارد به این دلیل که معمولاً رویدادهای DDL به ندرت رخ می‌دهد. می‌توانید هنگامی که قصد دارید دستورات DDL را اجرا کنید موقتاً Trigger را با دستورات زیر غیر فعال نمائید:

```
DISABLE TRIGGER ALL ON DATABASE
DISABLE TRIGGER ALL ON ALL SERVER
```

پس از Overrdie کردن می‌توانید مجدداً Trigger را فعال کنید:

```
ENABLE TRIGGER ALL ON DATABASE
ENABLE TRIGGER ALL ON ALL SERVER
```

4- معرفی DDL Event Groups:

برای مشاهده جزئیات بیشتر می‌توانید به این لینک مراجعه کنید.

Server Level

DDL_SERVER_LEVEL_EVENTS

DDL_LINKED_SERVER_EVENTS

DDL_LINKED_SERVER_LOGIN_EVENTS

DDL_REMOTE_SERVER_EVENTS

DDL_EXTENDED_PROCEDURE_EVENTS

DDL_MESSAGE_EVENTS

DDL_ENDPOINT_EVENTS

DDL_SERVER_SECURITY_EVENTS

DDL_LOGIN_EVENTS

DDL_GDR_SERVER_EVENTS

DDL_AUTHORIZATION_SERVER_EVENT Database Level

Database Level

```
DDL_DATABASE_LEVEL_EVENTS
```

DDL_TABLE_VIEW_EVENTS

DDL_TABLE-EVENTS

DDL_VIEW_EVENTS

DDL_INDEX_EVENTS

DDL_STATISTICS_EVENTS

DDL_DATABASE_SECURITY_EVENTS

DDL_CERTIFICATE_EVENTS

DDL_USER_EVENTS

DDL_ROLE_EVENTS

DDL_APPLICATION_ROLE_EVENTS

DDL_SCHEMA_EVENTS

DDL_GDR_DATABASE_EVENTS

DDL_AUTHORIZATION_DATABASE_EVENTS

DDL_FUNCTION_EVENTS

DDL_PROCEDUER_EVENTS

DDL_TRIGGER_EVENTS

DDL_PARTITION_EVENTS

DDL_PARTITION_FUNCTION_EVENTS

DDL_PARTITION_SCHEME_EVENTS

DDL_SSB_EVENTS

DDL_MESSAGE_TYPE_EVENTS

DDL_CONTRACT_EVENTS

DDL_QUEUE_EVENTS

DDL_SERVER_EVENTS

DDL_ROUTE_EVENTS

DDL_REMOTE_SERVICE_BINDING_EVENTS

DDL_XML_SCHEMA_COLEECTION_EVENTS

DDL_FULLTEXT_CATALOG_EVENTS

DDL_DEFAULT_EVENTS

DDL_EXTENDED_PROPERTY_EVENTS

DDL_PLAN_GUIDE_EVENTS

DDL_RULE_EVENTS

DDL_SYNONYM_EVENTS

DDL_EVENT_NOTIFICATION_EVENTS

DDL_ASSEMBLY_EVENTS

DDL_TYPE_EVENTS

نظرات خوانندگان

نویسنده: محمد
تاریخ: ۱۳۹۳/۰۶/۲۵ ۲۰:۴۱

با سلام
اول از همه تشکر میکنم از شما دوست عزیز بابت زحماتی که کشیدید.
سئوال بنده اینه اگر بخواهیم خودمون یک نوع TRIGGER جدید به Database اضافه کنیم به چه صورت امکان پذیره؟
به طور مثال من می‌خوام یک تریگر فقط برای کلیه فعالیت‌ها در حوزه‌ی بکاپ و ریستور بنویسم.
مثل تریگر Alter table یا Drop_table که ماهیتشون از قبل تعریف شده است.

با تشکر

نویسنده: محمد رجبی
تاریخ: ۱۳۹۳/۰۷/۰۲ ۱۰:۳۴

با سلام و احترام؛ همانطور که در متن به عرض رسانده شده:
" از عبارت ON برای مشخص کردن محدوده Trigger در سطح SQL Instance (در این صورت ON All SERVER نوشته می‌شود) و یا در سطح Database (در این حالت ON DATABASE نوشته می‌شود) استفاده می‌شود و از عبارت FOR برای مشخص کردن رویداد یا گروه رویدادی که سبب فراخوانی Trigger می‌شود، استفاده خواهد شد. "
در خصوص مثالی که اشاره کردید، به نظرم می‌رسد از Trigger برای این منظور استفاده نمی‌شود (در حوزه‌ی بکاپ و ریستور)، شاید اگر قصدتان به منظور ثبت log و ... بایست از Auditing استفاده کنید. به این منظور در Auditing با توجه به جدول زیر می‌توان اقدام به ثبت موارد نمود:

Server-Level Audit Action Groups

Action group name	Event Class	Description
BACKUP_RESTORE_GROUP	Audit Backup/Restore	یک دستور Backup یا Restore صادر شود

به طور مختصر Auditing به شرح زیر است:

بررسی SQL Server Audit

بازبینی (Auditing) شامل پیگیری و ثبت رویدادهایی است که در سطح SQL Instance و یا Databaseهای روی یک سیستم اتفاق می‌افتد. چندین سطح برای Auditing در SQL Server وجود دارد که به صلاحدید و نیازمندی‌های نصب شما وابسته است. شما می‌توانید گروه اقدامات بازبینی سرویس دهنده (server audit action groups) را به ازای هر SQL Instance و گروه اقدامات بازبینی بانک اطلاعاتی (database audit action groups) را به ازای هر بانک اطلاعاتی ثبت کنید. رویداد Audit هر زمان عملی که مورد رسیدگی قرار گرفته اتفاق افتد، رخ می‌دهد.
تا پیش از SQL SERVER 2008، شما باید از خصیصه‌های متعددی برای انجام یک مجموعه کامل بازبینی (Auditing) برای نمونه DDL Trigger، DML Trigger و SQL Trace، بر روی یک SQL Instance استفاده می‌کردید.
SQL SERVER 2008، همه قابلیت‌های Auditing را روی یک audit specification ترکیب می‌کند. Audit Specification با تعریف یک شی بازبینی (audit object) در سطح سرویس دهنده برای ثبت (logging) یک دنباله بازبینی (audit trial) آغاز می‌شود.
توجه شود که بایست یک شیء بازبینی ایجاد کنید پیش از اینکه یک Server Audit Specification و یا Database Audit Specification ایجاد کنید.
Server Audit Specification، گروه اقدامات در سطح سرویس دهنده را جمع آوری می‌کند که با رویدادهای وسیعی فعال می‌شوند، این گروه اقدامات تحت عنوان Server-Level Audit Action Groups تشریح شده اند. شما می‌توانید یک Server Audit Specification را به ازای هر Audit ایجاد کنید چرا که هر دو در محدوده یک SQL Instance ایجاد می‌شوند.
Database Audit Specification، گروه اقدامات در سطح بانک اطلاعاتی را جمع آوری می‌کند که با رویدادهای وسیعی فعال می‌شود. این گروه اقدامات تحت عنوان‌های Database-Level Audit Action Groups و Database-Level Audit Actions تشریح

شده اند.می توانید یک Database Audit Specification را به ازای هر Audit در بانک اطلاعاتی SQL Server ایجاد کنید.
همچنین می‌توانید هر گروه اقدامات بازبینی(audit action groups) یا رویدادهای بازبینی(audit events) را به یک Database Audit Specification اضافه کنید. گروه اقدامات بازبینی، گروه اقدامات از پیش تعریف شده ای هستند و رویدادهای بازبینی اقدامات تجزیه ناپذیری هستند که توسط موتور بانک اطلاعاتی مورد رسیدگی قرار می‌گیرند، هر دو در محدوده بانک اطلاعاتی (Database) هستند. این اقدامات برای Audit فرستاده می‌شوند تا در Target (که می‌تواند یک فایل، Windows Security Log و یا Windows Application Log باشد) ذخیره شوند. برای نوشتن در Windows Security Log لازم است که Service Account سرویس دهنده شما به Policy، Generate security audits اضافه شده باشد، به صورت پیش فرض Local System، Local Service و Network Service بخشی از این Policy می‌باشند. پس از اینکه Audit را ایجاد و فعال کردید، Target ورودی‌ها را دریافت خواهد کرد.

Server-Level Audit Action Groups (گروه اقدامات بازبینی در سطح سرویس دهنده)

این گروه اقدامات به گروه رویداد Security Audit شبیه هستند. به طور خلاصه این گروه اقدامات، اقداماتی را که در یک SQL Instance شامل می‌شوند، در بر می‌گیرد. برای مثال اگر گروه اقدام مناسب با Server Audit Specification اضافه شده باشد هر شیء در هر Schema که مورد دستیابی قرار می‌گیرد، ثبت می‌شود. اقدامات در سطح سرویس دهنده به شما اجازه نمی‌دهد که جزئیات اقدامات در سطح بانک اطلاعاتی را فیلتر کنید. یک بازبینی در سطح بانک اطلاعاتی برای انجام به جزئیات دقیق فیلتر کردن نیاز دارد، برای مثال اجرای دستور Select روی جدول Customers برای login هایی که در گروه Employee هستند.

Database-Level Audit Action Groups (گروه اقدامات بازبینی در سطح بانک اطلاعاتی)

این گروه اعمال به کلاس‌های رویداد Security Audit شبیه هستند.

Database-Level Audit Actions

اقدامات در سطح بانک اطلاعاتی، اقدامات بازبینی خاصی را به طور مستقیم روی Database، Schema و اشیاء Schema (از قبیل جداول، View ها، رویه‌های ذخیره شده، توابع و ...) فراهم می‌کند. این اقدامات برای فیلدها (Columns) صدق نمی‌کنند.

Audit-Level Audit Action Groups

شما می‌توانید اقداماتی را که در فرآیند Auditing هستند، بازبینی کنید که می‌تواند در محدوده سرویس دهنده یا بانک اطلاعاتی باشد. در محدوده بانک اطلاعاتی تنها برای database audit specification رخ می‌دهد.
جهت بررسی بیشتر به این لینک مراجعه شود.

نویسنده: MF
تاریخ: ۱۳۹۳/۰۷/۰۲ ۱۷:۷

DDL Triggerها , حالت Befor ندارند(انجام یک سری کارها قبل از عملیات درج,حذف و ...).برای حل این مساله باید از INSTEAD OF در تریگر استفاده نمود ؟

نویسنده: محمد رجبی
تاریخ: ۱۳۹۳/۰۷/۰۳ ۱۲:۱

مواردی که اشاره کردید مربوط به DML Triggers می‌باشند.
برای اطلاعات بیشتر به این لینک Understanding DDL Triggers vs. DML Triggers مراجعه شود.

نویسنده: مجید فاضلی
تاریخ: ۱۳۹۳/۰۷/۰۶ ۱۲:۴۶

باید از از حالت INSTEAD OF استفاده کنیم در **DML** Trigger ای که قراره نوشته بشه.
می‌توانیم در یک جدول از دیتابیس مان بر اساس یک شرط خاص, عملیات Insert,Delete,Update را مدیریت کنیم.
بعنوان مثال در قطعه کد زیر ما قبل از عملیات Insert در جدول tblTest چک میکنیم که اگر مقدار ستون FirstName برابر با null بود عملیات Insert آن رکورد در دیتابیس لغو شود.

```
ALTER TRIGGER [dbo].[Prevent_Befor_Insert_Null]
   ON  [dbTest].[dbo].[tblTest]
   INSTEAD OF INSERT
AS
BEGIN
```

```
SET NOCOUNT ON
IF OBJECT_ID(N'dbTest.dbo.tblTest.FirstName') is null
BEGIN
DECLARE @Id int
SET @Id = (select Id from inserted)
RAISERROR ('16,1,'مقدار فیلد نام نباید خالی باشد)
ROLLBACK
END
END
```

از دو طریق می‌توان به مقادیر فیلدهای رکورد جاری دسترس داشت:
-1 استفاده از OBJECT_ID و ذکر نام فیلد مورد نظر
-2 گرفتن فیلد مورد نظر از جدول INSERTED یا DELETED
DML Triggerها دارای دو جدول خاص بنام‌های INSERTED و DELETED هستند که توسط خود SQL Server مدیریت می‌شوند.در حقیقت در پشت صحنه, ما با این دو جدول در هنگام تغییر مقادیر داده‌های جداول دیتابیس کار می‌کنیم و نمی‌توانیم بصورت مستقیم داده‌های جداول موجود در دیتا بیس مان را تغییر دهیم.
جدول INSERTED و DELETED حاوی رکورد جاری است که تحت تاثیر عمل درج, ویرایش و حذف در دیتابیس قرار گرفته است.
اطلاعات بیشتر در اینجا و اینجا

نویسنده: محمد
تاریخ: ۱۳۹۳/۰۷/۲۸ ۰:۱۵

سلام؛ تشکر از توضیحات شما. اجازه بدید من طور دیگری سئوالم رو مطرح کنم. به طور مثال ما برای کار با تاریخ شمسی در SQl چندین روش پیش رو داریم که وارد جزئیات آن نمیشوم ولی یکی از این روش‌ها که به خوبی جواب میدهد استفاده از CLR است که ما با توسط این قابلیت می‌توانیم یک نوع دیتا تایپ جدید ، با ماهیت جدید در اس کیو ال اضافه کنیم. حالا منظور بنده این است که آیا برای تریگرها هم می‌شود این کار را انجام داد یا خیر؟ مثلا توسط CLR یا هر روش دیگری که وجود دارد ، ما بیاییم و یک نوع تریگر کاملا جدید و Customize شده برای خودمان درست کنیم. به طور مثال : زمانی که کاربر از دیتابیس بخواهد بکاپ تهیه کند یا آن را ریستور کند ، یکسری فعالیتها به آن فعالیت اضافه شود یا در راستای آن انجام شود. با تشکر