

استفاده از DDL Trigger امکان ایجاد Trigger برای عملیات DDL(Data Definition Language) از SQL Server 2005 فراهم کردید. 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 نمایید. 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 دستورهایی 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 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 dd1_log

```
CREATE TABLE dd1_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	Temp_TestTable
ALTER_TABLE	11/7/2010 10:36:26 AM	55	1936-603102405\SQL2008	sa	dbo	Test	Temp_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	Temp_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.Temp_TestTable (a nvarchar(50) NULL) ON [PRIMARY]					
NULL	NULL	ALTER TABLE dbo.Temp_TestTable SET (LOCK_ESCALATION = TABLE)					
NULL	NULL	DROP TABLE dbo.TestTable					
NULL	NULL	EXECUTE sp_rename N'dbo.Temp_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 تأثیرات 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_COLLECTION_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:

۲۰:۴۱ ۱۳۹۳/۰۶/۲۵

با سلام

اول از همه تشکر میکنم از شما دوست عزیز بابت زحماتی که کشیدید.
سؤال بنده اینه اگر بخواهیم خودمون یک نوع TRIGGER جدید به Database اضافه کنیم به چه صورت امکان پذیره؟
به طور مثال من می‌خواهم 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