

Trigger

از خواننده محترم تقاضا مي شود با ارسال نظرات و پيشنهادات خود در مورد اين مقاله ويا اشكالات موجود در محتوي يا متن آن به آدرس mehdi.aliha@gmail.com يا mehdi_mohammad_aliha@yahoo.com در بهبود مقاله سهيم شود.

باتشكر
مهدی محمد علیها
اسفند 1388

منابع: <http://msdn.microsoft.com>

مزیت يك Trigger این است كه وقتی دستور مربوط به آن اجرا شود . بطور اتوماتيك توسط Sql server به اجرا در مي آید این ویژگی پایگاه داده را قوي تر مي كند.نیازی نیست شما نگران باشید كه آیا مشتریان پایگاه داده با قوانین منطقی كار و اطلاعات جزئی وابسته آشنایی دارند یا خیر Trigger ها معمولا برای پیاده سازی قوانینی كه با استفاده از قیود قابل تعریف نیستند بكار مي روند .يك Trigger يك نوع خواست از procedure های ذخیره شده است كه بطور خودكار زمانی كه يك اتفاق در سرور پایگاه داده رخ مي دهد ، اجرا مي شود .زمانی كه يك كاربر سعی مي كند تا داده را از طریق يك رخ داد یا يك زبان دستكاری داده یا DML اصلاح نماید . Trigger های DML اجرا مي شوند .رخ داد های DML شامل دستورات update،insert یا delete برروي يك جدول نما هستند . Trigger های DDL در پاسخ به يك تنوع از رخ دادهها زبان تعریف داده یا DDL اجرا مي شوند.اینها در اصل شامل عبارات Create،Alter،Drop هستند . Trigger های DDL و DML میتوانند در sql server database engine مستقیما از طریق عبارات T-sql ویا طریق متدهای اسمبلی ها اجرا شوند .كه در زبان مشترك اجرا یا CLR در Microsoft.netframework ایجاد شده وبرروي يك نمونه از sql server بار گزاری مي شوند . Sql server امکان ایجاد trigger های چندگانه مربوط به هر عبارت مشخص را فرا هم مي سازد .به این نکته توجه نمایید كه كدهای mellish's در درون triggerها مي توانند با يك امتیاز بالا اجرا شوند .به تصویر زیر توجه کنید.

Syntax

Trigger on an INSERT, UPDATE, or DELETE statement to a table or view (DML Trigger)

```
CREATE TRIGGER [ schema_name . ]trigger_name
ON { table | view }
[ WITH <dml_trigger_option> [ ,...n ] ]
{ FOR | AFTER | INSTEAD OF }
{ [ INSERT ] [ , ] [ UPDATE ] [ , ] [ DELETE ] }
[ WITH APPEND ]
[ NOT FOR REPLICATION ]
AS { sql_statement [ ; ] [ ,...n ] | EXTERNAL NAME <method specifier [ ; ] > }
```

```
<dml_trigger_option> ::=
    [ ENCRYPTION ]
    [ EXECUTE AS Clause ]
```

```
<method_specifier> ::=
    assembly_name.class_name.method_name
```

Trigger on a CREATE, ALTER, DROP, GRANT, DENY, REVOKE, or UPDATE STATISTICS statement (DDL Trigger)

```
CREATE TRIGGER trigger_name
ON { ALL SERVER | DATABASE }
[ WITH <ddl_trigger_option> [ ,...n ] ]
{ FOR | AFTER } { event_type | event_group } [ ,...n ]
AS { sql_statement [ ; ] [ ,...n ] | EXTERNAL NAME < method specifier > [ ; ] }
```

```
<ddl_trigger_option> ::=
    [ ENCRYPTION ]
    [ EXECUTE AS Clause ]
```

```
<method_specifier> ::=
    assembly_name.class_name.method_name
```

Trigger on a LOGON event (Logon Trigger)

```
CREATE TRIGGER trigger_name
ON ALL SERVER
[ WITH <logon_trigger_option> [ ,...n ] ]
{ FOR | AFTER } LOGON
AS { sql_statement [ ; ] [ ,...n ] | EXTERNAL NAME < method specifier > [ ; ] }
```

```
<logon_trigger_option> ::=
    [ ENCRYPTION ]
    [ EXECUTE AS Clause ]
```

```
<method_specifier> ::=
    assembly_name.class_name.method_name
```

Schema-name: نام الگوی است که trigger DML به آن تعلق دارد. Triggerها DML محدود به الگوی جدول یا نمایی هستند که بروی آن ایجاد می شوند. Schema-name نمی تواند برای Triggerهای DDL مشخص شود.

Trigger-name: نام Trigger می باشد. نام Trigger باید مطابق با قوانین اشاره گر ها باشد به استثنای این که این آرگومان نمی تواند با ## یا ## آغاز شود.

Table|view: جدول یا نمایی است که Trigger DML بروی آن اجرا می شود و برخی اوقات به همان جدول Trigger یا نمایی trigger اشاره می شود. مشخص کردن نام مناسب برای جدول یا نما اختیاری است. یک نما می تواند تنها به وسیله یک Trigger in state of مراجعه شود.

Data base: محدوده یک Trigger DML را به پایگاه داده جاری اعمال می کند. در صورت تعیین در هر زمان Event Group یا Event Type در هر محل پایگاه داده جاری رخ می دهد.

All server: محدوده یک Trigger DDL را به پایگاه داده جاری اعمال می کند. در صورت تعیین در هر زمان Event Group یا Event Type در هر محلی Server جاری رخ می دهد.

With Encryption: متن مربوط به عبارت Create Trigger را رمز دار می کند. استفاده از With Encryption مانع از انتشار راه انداز به صورت قسمتی از پاسخ sql server می شود. With Encryption نمی تواند برای اندازه CLR مشخص شود.

Execute AS: امنیت را برای راه اندازی که اجرا می شود تعیین می نماید. نما را قادر می سازد تا کنترل کنید کدام حساب کاربری از sql server استفاده می کند. تا مجوز هایی را بروی هر شی از پایگاه داده که به وسیله راه انداز اجرا می شود ایجاد نماید.

After: تعیین می کند که تنها زمانی که عملیات های مشخص شده در راه انداز ی عبارت Sql با موفقیت اجرا شوند. Trigger DML بکار می افتد. به خاطر داشته باشید که تمام اعمال پشت سرهم و چک های اجباری نیز باید قبل از اینکه Trigger به کار بیفتد موفق شوند. هنگامی که for تنها کلمه کلیدی باشد. After کلمه پیش فرض می باشد. Trigger های After نمی توانند بروی نما تعریف شوند.

Instate of: تعیین می کند که Trigger DML بجای عبارت sql Trigger اجرا می شود. بنا براین اعمال مربوط به عبارات Trigger دارای برتری هستند. یک Trigger in state of می تواند توسط عبارات delete, insert یا update برای یک

نما یا جدول تعریف شود. بنابراین شما می توانید جای که هر نما راه انداز in state of خود را دارد نما ها را بروی نما تعریف کنید. راه انداز های in state of بر روی نمایی قابل بهنگام رسانی که از With check option استفاده می کنند مجاز نمی باشند. هنگامی که یک Trigger in state of به یک نمایی قابل به هنگام رسانی With check اضافه می شود. Sql Server یک خطا را اعلام می کند کاربر باید قبل از تعریف Trigger in state of آن گزینه را با استفاده از Alter View حذف نماید.

Delete, update, insert: مشخص می نماید هنگامی که عبارات اصلاح داده در مقابل این جدول یا نما آزمایش شود این عبارات Trigger DML را فعال می کند. در نهایت یک گزینه باید مشخص شود. هر ترکیبی از این گزینه ها به تعریف Trigger مجاز می باشد. در مورد Triggerهای Instate of گزینه Delete بر روی جداولی که یک ارتباط ارجاعی دارند که یک عمل undelet پشت سر هم را تعیین می کند مجاز نمی باشد.

Event-type: نام رخ داد T-sql پس از اجرا می باشد. که یک Trigger DDL را وادار به اجرا می کند رخ دادهای معتبر برای استفاده در Trigger های DDL در DDL Event for use with ddl trigger فهرست می شوند.

Event-group: نام گروه بندی از پیش تعریف شده مربوط به رخ دادهای زبان T-sql می باشد. Trigger DDL پس از اجرای هر رخ داد زبان T-sql که متعلق به Event group می باشد اجرا می گردد. گروهی رخ دادی که برای استفاده در Trigger های از نوع DDL معتبر هستند Event groupes for use with ddl trigger فهرست می شوند. پس از اینکه اجرای Create Trigger به پایان رسید. Event group نیز با افزودن انواع رخ داده های تحت آن به نمایی فهرستی sys.triggerEvent مانند یک ماکرو عمل می کند.

With Append: تعیین می نماید که یک Trigger اضافی مربوط به یک نوع موجود باید افزودن شود. استفاده از این دستور اختیاری تنها زمانی لازم می شود که سطح تطابق ۶۵ یا پایین تر باشد. در صورتی که سطح تطابق ۷۰ یا بیشتر باشد دستور with Append برای افزودن یک Trigger اضافی از نوع موجود لازم نمی شود. این یک رفتار پیش فرض از Create Trigger با تنظیم سطح تطابق بر روی ۷۰ یا بالاتر می باشد. with Append نمی تواند به همراه Trigger های in state of استفاده شود. و همین طور در صورتی که Trigger After دقیقاً تعیین شود. نمی تواند مورد استفاده قرار گیرد. تنها زمانی که for in state of یا After و به دلایل تطابق مشخص شود. with Append می تواند مورد استفاده قرار گیرد. در صورتی که external Name تعیین شده باشد یعنی در صورتی که Trigger CLR یک Trigger باشد نمی تواند مشخص شود. مهم است بدانید که with Append در نسخه های بعد sql server حذف خواهد شد.

Not for Replication: تعیین می نماید که زمانی که یک نمونه تکراری جدولی را که در Trigger قرار دارد اصلاح می کند Trigger نباید اجرا شود.

Sql_statement: شرایط و اعمال Trigger می باشد. شرایط Trigger ضوابط افزودن آن را تعیین می کند که مبنی بر این می باشند که آیا عبارات DML یا DDL باعث اجرای اعمال Trigger می شوند یا خیر اعمال Trigger که در عبارات T-sql مشخص شده اند زمانی موثر واقع می شوند که عملیات DML یا DDL آزمایش شوند. Trigger ها می توانند شامل هر تعداد و نوع از دستورات T-sql باشند. البته استثنائاتی هم وجود دارد. یک Trigger که برای کنترل یا تغییر داده مبتنی به اصلاح داده یا عبارات تعریفی طراحی می شوند نباید داده را به کاربر برگرداند. عبارات T-sql در یک Trigger مکرراً شامل زبان Control to flow می باشند. Trigger DML از جداول منطقی حذف شده و وارد شده استفاده می کنند. آنها از نظر ساختاری مشابه با جدولی هستند که trigger بر روی آن تعریف می شود. که این جدول همان جدولی است که عمل کاربر بر روی آن آزمایش می شود. جداول حذف و وارد شده مقادیر قدیمی یا جدید داده های ستونهای که ممکن است بوسیله کاربر تغییر داده شوند را در خود نگهداری می کند. برای مثال برای بازیابی تمام مقادیر در جدول حذف شده از کد زیر استفاده کنید.

```
SELECT *  
FROM deleted
```

Trigger های DDL اطلاعات مربوط به رخ داد راه اندازی را با استفاده از تابع Event data نگهداری می کنند. در صورتی که سطح تطابق بر روی ۷۰ تنظیم شده باشد. sql server در یک Trigger، Delete، insert یا update به ستون text یا Ntext یا image اجازه نمی دهد تا به جداول وارد و حذف شده رجوع نمایند. مقادیر image، Ntext، Text در جدول وارد و حذف شده قابل دست یابی نیستند. برای بازیابی مقدار جدید در Trigger، insert، update جدول وارد شده را با جدول بروز رسانی اصلی مرتبط نمایید. هنگامی که سطح تطابق برابر ۶۵ یا کمتر شد رشته های خالی برای ستونهای وارد یا حذف شده image، Ntext، Text برگردانده می شوند که مقادیر پوچ یا Null مجاز هستند. رشته های با طول 0 در صورتی که ستونها قادر به دریافت مقادیر پوچ نباشند برگردانده می شوند. در صورتی که سطح تطابق برابر با ۸۰ یا بیشتر باشد. sql server اجازه می دهد تا ستونهای image، Text، Ntext از طریق Trigger In state of بر روی جداول یا نماها بهنگام شوند. مهم است بدانید که از انواع داده image، text، Ntext در نسخه های بعد sql server حذف خواهند شد. در عوض می توانید از varchar(max)، nvarchar(max)، varbinary(max) استفاده نمایید. هر دو Trigger های After و in state of این داده را جداول اضافه شده پشتیبانی می کنند.

Method-specifier: برای یک Trigger CLR این متد یک اسمبلی را برای اتصال به یک Trigger مشخص می کند. متد نباید هیچ آرگومانی را دریافت نماید و باید مقدار تهی یا پوچ برگرداند. Class Name. باید یک اشاره گر پوچ از sql Server باشد و باید بصورت یک کلاس در اسمبلی به همراه قابلیت دید اسمبلی وجود داشته باشد. در صورتی که کلاس یک نام name space qualify داشته باشد که از کاراکتر نقطه برای جدا سازی قسمت های فضای نام استفاده می کند نام کلاس باید با استفاده از محدود کننده های [] یا " تعیین محدوده شود. کلاس نمی تواند تو در تو باشد. توجه کنید که بصورت پیش فرض قابلیت sql server برای اجرای کد CLR خاموش می باشد. شما می توانید اشیاء پایگاه داده که به ماژولهای کد مدیریت شد مرجوع می شوند را ایجاد، اصلاح و یا رها نمایید ولی این مراجع در یک نمونه از sql server اجرا نمی شوند. مگر اینکه گزینه CLR Enable option با استفاده از Sp-Configuration فعال شود.