**بسمه تعالی**



برنامه نویسی بانک اطلاعاتی SQL Server

**نویسنده : محمد حسین فخرآوری**

**تریگر چیست؟**

* تریگر یک رویه ذخیره شده است که زمانیکه داده ای تغییر پیدا می کند اجرا می شود.
* در اجرای اتوماتیک تریگر پارامتر و مقادیربرگشتی وجود ندارد.

**اما چه جوری با تریگر کار کنیم؟**

در طول عمر یک تریگر دو تا جدول به وجود میان به نامهای Inserted و Deleted که از اونها می شه استفاده کرد.

اما نمیشه بهشون دست زد، به این جداول می گن جداول منطقی.

تو این جدولها اطلاعات جدولی هست که قرار بوده داده هاش رو دست بزنیم و ما رو اون تریگر گذاشتیم

اگر گفته بودیم که حساس باشه به Insert کردن، اونوقت Inserted حاوی داده های ما بود که قرار بوده Insert شه یا Insert شده و Deleted خالیه. (بسته به نوع تریگر)

اگر گفته بودیم که حساس باشه به Delete کردن، اونوقت Deleted حاوی داده های ما بود که قرار بوده Delete شه یا Delete شده و Inserted خالیه

اگر گفته بودیم که حساس باشه به Update کردن، اونوقت Deleted حاوی داده های قبلی این جدول بود و Inserted حاوی اطلاعات جدید ما که قرار بوده جایگزین بشه یا جایگزین شده

با یک مثال مطلب را روشن تر می کنیم فرض کنید در فاکتور فروش می خواهید پس از فروخته شدن تعداد مشخصی از یک کالا، موجودی کالای فروخته شده کم شود در این حالت می توانید با تعریف یک تریگر به Sql Server بگوئید که پس از عملیات insert و update در جدول فاکتور فروش به طور اتوماتیک مقدار فروخته شده از جدول کالا و یا انبار کم شود.

البته برای این کار می توانید پروسیجری به طور مجزا بنویسید و شاید بگوئید نیازی به استفاده از تریگر نیست البته گفته شما درست است و اصلا می توان یک برنامه پایگاه داده را بدون استفاده از تریگرها و فقط با استفاده از پروسیجر نوشت ولی امکان تریگر در بسیاری از مواقع باعث راحتی کارشما می شود و شما را از درگیر کردن با کدهای پیچیده نجات می دهد که فهمیدن این موضوع نیازمند تجربه است.

**از موارد مورد استفاده تریگرها می‌توان به موارد زیر اشاره کرد**

* محاسبه مقدار فیلدهای مشتق شده به صورت خودکار،
* جلوگیری از [تراکنش‌های](http://fa.wikipedia.org/wiki/تراکنش) نامعتبر،
* اعمال منطق اجازه‌دهی پیچید،
* اعمال یکپارچگی داده در [پایگاه داده توزیع شده](http://fa.wikipedia.org/w/index.php?title=%D9%BE%D8%A7%DB%8C%DA%AF%D8%A7%D9%87_%D8%AF%D8%A7%D8%AF%D9%87_%D8%AA%D9%88%D8%B2%DB%8C%D8%B9_%D8%B4%D8%AF%D9%87&action=edit&redlink=1&preload=%D8%A7%D9%84%DA%AF%D9%88:%D8%A7%DB%8C%D8%AC%D8%A7%D8%AF+%D9%85%D9%82%D8%A7%D9%84%D9%87/%D8%A7%D8%B3%D8%AA%D8%AE%D9%88%D8%A7%D9%86%E2%80%8C%D8%A8%D9%86%D8%AF%DB%8C&editintro=%D8%A7%D9%84%DA%AF%D9%88:%D8%A7%DB%8C%D8%AC%D8%A7%D8%AF+%D9%85%D9%82%D8%A7%D9%84%D9%87/%D8%A7%D8%AF%DB%8C%D8%AA%E2%80%8C%D9%86%D9%88%D8%AA%DB%8C%D8%B3&summary=%D8%A7%DB%8C%D8%AC%D8%A7%D8%AF+%DB%8C%DA%A9+%D9%85%D9%82%D8%A7%D9%84%D9%87+%D9%86%D9%88+%D8%A7%D8%B2+%D8%B7%D8%B1%DB%8C%D9%82+%D8%A7%DB%8C%D8%AC%D8%A7%D8%AF%DA%AF%D8%B1&nosummary=&prefix=&minor=&create=%D8%AF%D8%B1%D8%B3%D8%AA+%DA%A9%D8%B1%D8%AF%D9%86+%D9%85%D9%82%D8%A7%D9%84%D9%87+%D8%AC%D8%AF%DB%8C%D8%AF&withJS=MediaWiki:Intro-Welcome-NewUsers.js)،
* اعمال قواعد کسب‌وکار پیچیده،
* ثبت وقایع (logging)،
* نظارت پیچیده،
* یکسان نگه داشتن انعکاس‌های (replicate) جدول،
* گردآوری آمار

**مثالی برای لاگ سیستم**

|  |  |
| --- | --- |
| CREATE TABLE [dbo].[Manegment](  [ID] [int] IDENTITY(1,1) PRIMARY KEY,  [UserName] [nvarchar](50) NULL,  [Reshte] [nvarchar](50) NULL,  [Tell] [nchar](10) NULL) | CREATE TABLE [dbo].[Log\_Manegment]  (  [IdRow] NUMERIC(18,0) IDENTITY(1,1) PRIMARY KEY,  [ID] [int] ,  [UserName] [nvarchar](50) NULL,  [Reshte] [nvarchar](50) NULL,  [Tell] [nchar](10) NULL,  [OperationTime] DATETIME,  [Operation] VARCHAR(5),  [UserNameLogin] NVARCHAR(100)  ) |

**توضیحات مسئله**

2 جدول به نام های مدیریت و لاگ مدیریت درست کرده ایم.

تریگر هایی که قصد مثال زدن داریم بر روی Event های Insert – Update – Delete کار می کند.

بعد از اجرای هر یک از Event ها یک رکورد به عنوان لاگ در Log\_Manegment ثبت می شود.

**جداول منطقی INSERTED و DELETED**

* لازم به ذکر است که جدول منطقی UPDATED وجود ندارد.
* COLUMNS\_UPDATED جهت بررسی تغیرات ستون ها مورد استفاده قرار می گیرد.

**تریگر Insert**

|  |
| --- |
| CREATE TRIGGER [dbo].[Insert\_Manegment]  ON [dbo].[Manegment]  AFTER Insert  AS  SET NOCOUNT ON  INSERT INTO dbo.Log\_Manegment (ID,UserName,Reshte,Tell,OperationTime,Operation,UserNameLogin)  SELECT x.ID,x.UserName,x.Reshte,x.Tell,SYSDATETIME() ,'Ins',SYSTEM\_USER FROM Inserted x |

**تریگر Update**

|  |
| --- |
| CREATE TRIGGER [dbo].[Update\_Manegment]  ON [dbo].[Manegment]  AFTER Update  AS  SET NOCOUNT ON  if (COLUMNS\_UPDATED()> 0)  BEGIN  INSERT INTO dbo.Log\_Manegment (ID,UserName,Reshte,Tell,OperationTime,Operation,UserNameLogin)  SELECT b.ID,b.UserName,b.Reshte,b.Tell,b.OperationTime,b.Operation,b.UserNameLogin FROM  (  SELECT x.ID,x.UserName,x.Reshte,x.Tell,SYSDATETIME() AS OperationTime ,'New' AS Operation,SYSTEM\_USER AS UserNameLogin FROM Inserted x  UNION  SELECT x.ID,x.UserName,x.Reshte,x.Tell,SYSDATETIME() AS OperationTime,'Old' AS Operation,SYSTEM\_USER AS UserNameLogin FROM Deleted x  )b  END |

* COLUMNS\_UPDATED جهت بررسی تغیر دیتای سطر جدول بکار می رود.

|  |
| --- |
| CREATE TRIGGER [Update\_OnlyUserName] ON [dbo].[Manegment]  FOR UPDATE  AS  SET NOCOUNT ON  IF (UPDATE(UserName))  BEGIN  INSERT INTO dbo.Log\_Manegment (ID,UserName,Reshte,Tell,OperationTime,Operation,UserNameLogin)  SELECT b.ID,b.UserName,b.Reshte,b.Tell,b.OperationTime,b.Operation,b.UserNameLogin FROM  (  SELECT x.ID,x.UserName,x.Reshte,x.Tell,SYSDATETIME() AS OperationTime ,'New' AS Operation,SYSTEM\_USER AS UserNameLogin FROM Inserted x  UNION  SELECT x.ID,x.UserName,x.Reshte,x.Tell,SYSDATETIME() AS OperationTime,'Old' AS Operation,SYSTEM\_USER AS UserNameLogin FROM Deleted x  )b  END |

* IF ( UPDATE(Column1) OR UPDATE(Column2))

**تریگر Delete**

|  |
| --- |
| CREATE TRIGGER [dbo].[Delete\_Manegment]  ON [dbo].[Manegment]  AFTER DELETE  AS  INSERT INTO dbo.Log\_Manegment (ID,UserName,Reshte,Tell,OperationTime,Operation,UserNameLogin)  SELECT x.ID,x.UserName,x.Reshte,x.Tell,SYSDATETIME() AS OperationTime,'Del' AS Operation,SYSTEM\_USER AS UserNameLogin FROM Deleted x |

* **بعد از ساخت توابع فوق بر روی جدول Manegment دستورات I-U-D را اعمال کنید ، سپس جدول Log\_Manegment را ببنید.**

**جهت On-Off کردن تریگر**

|  |
| --- |
| ENABLE TRIGGER [Update\_Manegment] ON Manegment; |
| DISABLE TRIGGER [Update\_Manegment] ON Manegment; |

**حذف تریگر**

|  |
| --- |
| DROP TRIGGER [Update\_Manegment] |
| sp\_msforeachtable 'ALTER TABLE ? DISABLE TRIGGER all' |
| sp\_msforeachtable 'ALTER TABLE ? ENABLE TRIGGER all' |

**نمایش لیست تریگر های دیتابیس و متن آنها**

|  |
| --- |
| SELECT \* FROM sys.triggers |
| SELECT  sysobjects.name AS trigger\_name,  USER\_NAME(sysobjects.uid) AS trigger\_owner, s.name AS table\_schema,  OBJECT\_NAME(parent\_obj) AS table\_name,  OBJECTPROPERTY(id, 'ExecIsUpdateTrigger') AS isupdate,  OBJECTPROPERTY(id, 'ExecIsDeleteTrigger') AS isdelete,  OBJECTPROPERTY(id, 'ExecIsInsertTrigger') AS isinsert,  OBJECTPROPERTY(id, 'ExecIsAfterTrigger') AS isafter,  OBJECTPROPERTY(id, 'ExecIsInsteadOfTrigger') AS isinsteadof,  OBJECTPROPERTY(id, 'ExecIsTriggerDisabled') AS [disabled]  FROM sysobjects  JOIN sysusers ON sysobjects.uid=sysusers.uid  JOIN sys.tables t ON sysobjects.parent\_obj=t.object\_id  JOIN sys.schemas s ON t.schema\_id=s.schema\_id  WHERE sysobjects.type='TR' |
| SELECT Tables.name TableName,Triggers.name TriggerName,Triggers.crdate TriggerCreatedDate,Comments.text TriggerText  FROM sysobjects Triggers  JOIN sysobjects Tables ON Triggers.parent\_obj = Tables.id  JOIN syscomments Comments ON Triggers.id = Comments.id  WHERE Triggers.xtype = 'TR' AND Tables.xtype = 'U'  ORDER BY Tables.name,Triggers.name |

**قفل کردن تریگر با WITH ENCRYPTION**

|  |
| --- |
| ALTER TRIGGER [dbo].[Delete\_Manegment]  ON [dbo].[Manegment]  WITH ENCRYPTION |

**مثال جهت جلوگیری از درج یا ویرایش وحذف داده ها با تریگر**

* تریگری که قبل از ثبت ظرفیت لیست را بررسی می کند اگر کم بود اجازه ثبت می دهد.

|  |
| --- |
| CREATE TRIGGER [dbo].[ChekAdd]  ON Manegment  FOR insert  AS  declare @Row int = (SELECT COUNT(\*) FROM Manegment)  IF (@Row > 5) BEGIN  RAISERROR ('Full List',0,0); ROLLBACK; END  ELSE BEGIN  INSERT INTO dbo.Log\_Manegment (ID,UserName,Reshte,Tell,OperationTime,Operation,UserNameLogin)  SELECT x.ID,x.UserName,x.Reshte,x.Tell,SYSDATETIME() ,'Ins',SYSTEM\_USER FROM Inserted x  END |

**نکته مهم:**

تریگر ها روی جداول فیزیکی گذاشته می شوند نه جداول موقتی مثل # یا ## و تریگر ها پارامتر ندارند.

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; |

**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:

EventTypeنوع رویدادی که باعث فراخوانی Trigger شده است.

PostTimeزمانی که رویداد رخ می‌دهد.

SPIDکاربری که باعث ایجاد رویداد شده است.

ServerNameنام SQL Instance که رویداد در آن رخ داده است.

LoginNameنام Login که عمل مربوط به وقوع رویداد را اجرا می‌کند.

UserNameنام User که عمل مربوط به وقوع رویداد را اجرا می‌کند.

DatabaseNameنام Database که رویداد در آن رخ می‌دهد.

ObjectTypeنوع Object که اصلاح، حذف و یا ایجاد شده است.

ObjectNameنام Object که اصلاح، حذف و یا ایجاد شده است.

TSQLCommandدستور T-SQL که اجرا شده و باعث اجرا شدن Trigger شده است.

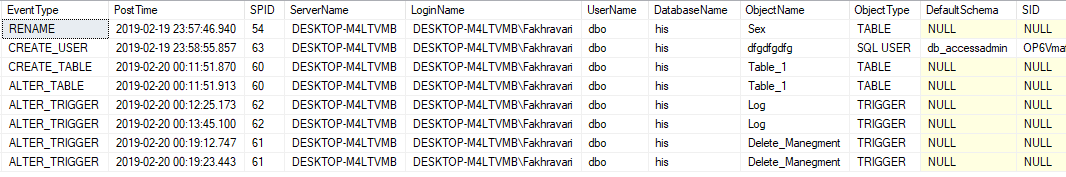
**برای نمونه در دستورات زیر جدولی با نام ddl\_log**

|  |
| --- |
| 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(max)')); |

نمونه ای از مقادیر ذخیره شده در جدول  ddl\_log به شکل زیر خواهد بود:



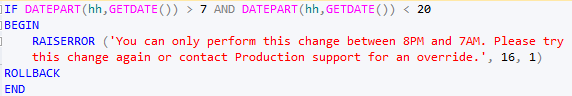
**معرفی DDL Event Groups**

**وقایع زیر DDL\_DATABASE\_LEVEL\_EVENTS در سطح پایگاه داده اجرا می شوند.**

|  |
| --- |
| SELECT \* FROM sys.trigger\_events |
| WITH DirectReports (name, parent\_type, type, level, sort)  AS  (  SELECT CONVERT(VARCHAR(255), type\_name),parent\_type,type,1,CONVERT(VARCHAR(255), type\_name)  FROM sys.trigger\_event\_types  WHERE parent\_type IS NULL  UNION ALL  SELECT CONVERT(VARCHAR(255), REPLICATE('| ', level) + e.type\_name),  e.parent\_type,e.type,level + 1, CONVERT(VARCHAR(255), RTRIM(sort) + '| ' + e.type\_name)  FROM sys.trigger\_event\_types AS e JOIN DirectReports AS d ON e.parent\_type = d.type  )  SELECT parent\_type,type,name FROM DirectReports  ORDER BY sort; |

| **parent\_type** | **type** | **name** |
| --- | --- | --- |
| NULL | 296 | ALTER\_SERVER\_CONFIGURATION |
| NULL | 10001 | DDL\_EVENTS |
| 10001 | 10016 | DDL\_DATABASE\_LEVEL\_EVENTS |
| 10016 | 10027 | DDL\_ASSEMBLY\_EVENTS |
| 10027 | 102 | ALTER\_ASSEMBLY |
| 10027 | 101 | CREATE\_ASSEMBLY |
| 10027 | 103 | DROP\_ASSEMBLY |
| 10016 | 10029 | DDL\_DATABASE\_SECURITY\_EVENTS |
| 10029 | 10033 | DDL\_APPLICATION\_ROLE\_EVENTS |
| 10033 | 138 | ALTER\_APPLICATION\_ROLE |
| 10033 | 137 | CREATE\_APPLICATION\_ROLE |
| 10033 | 139 | DROP\_APPLICATION\_ROLE |
| 10029 | 10038 | DDL\_ASYMMETRIC\_KEY\_EVENTS |
| 10038 | 248 | ALTER\_ASYMMETRIC\_KEY |
| 10038 | 247 | CREATE\_ASYMMETRIC\_KEY |
| 10038 | 249 | DROP\_ASYMMETRIC\_KEY |
| 10029 | 10036 | DDL\_AUTHORIZATION\_DATABASE\_EVENTS |
| 10036 | 205 | ALTER\_AUTHORIZATION\_DATABASE |
| 10029 | 10030 | DDL\_CERTIFICATE\_EVENTS |
| 10030 | 198 | ALTER\_CERTIFICATE |
| 10030 | 197 | CREATE\_CERTIFICATE |
| 10030 | 199 | DROP\_CERTIFICATE |
| 10029 | 10039 | DDL\_CRYPTO\_SIGNATURE\_EVENTS |
| 10039 | 257 | ADD\_SIGNATURE |
| 10039 | 255 | ADD\_SIGNATURE\_SCHEMA\_OBJECT |
| 10039 | 258 | DROP\_SIGNATURE |
| 10039 | 256 | DROP\_SIGNATURE\_SCHEMA\_OBJECT |
| 10029 | 10066 | DDL\_DATABASE\_AUDIT\_SPECIFICATION\_EVENTS |
| 10066 | 291 | ALTER\_DATABASE\_AUDIT\_SPECIFICATION |
| 10066 | 290 | CREATE\_DATABASE\_AUDIT\_SPECIFICATION |
| 10066 | 292 | DROP\_DATABASE\_AUDIT\_SPECIFICATION |
| 10029 | 10062 | DDL\_DATABASE\_ENCRYPTION\_KEY\_EVENTS |
| 10062 | 279 | ALTER\_DATABASE\_ENCRYPTION\_KEY |
| 10062 | 278 | CREATE\_DATABASE\_ENCRYPTION\_KEY |
| 10062 | 280 | DROP\_DATABASE\_ENCRYPTION\_KEY |
| 10029 | 10035 | DDL\_GDR\_DATABASE\_EVENTS |
| 10035 | 171 | DENY\_DATABASE |
| 10035 | 170 | GRANT\_DATABASE |
| 10035 | 172 | REVOKE\_DATABASE |
| 10029 | 10040 | DDL\_MASTER\_KEY\_EVENTS |
| 10040 | 253 | ALTER\_MASTER\_KEY |
| 10040 | 252 | CREATE\_MASTER\_KEY |
| 10040 | 254 | DROP\_MASTER\_KEY |
| 10029 | 10032 | DDL\_ROLE\_EVENTS |
| 10032 | 207 | ADD\_ROLE\_MEMBER |
| 10032 | 135 | ALTER\_ROLE |
| 10032 | 134 | CREATE\_ROLE |
| 10032 | 136 | DROP\_ROLE |
| 10032 | 208 | DROP\_ROLE\_MEMBER |
| 10029 | 10034 | DDL\_SCHEMA\_EVENTS |
| 10034 | 142 | ALTER\_SCHEMA |
| 10034 | 141 | CREATE\_SCHEMA |
| 10034 | 143 | DROP\_SCHEMA |
| 10029 | 10037 | DDL\_SYMMETRIC\_KEY\_EVENTS |
| 10037 | 245 | ALTER\_SYMMETRIC\_KEY |
| 10037 | 244 | CREATE\_SYMMETRIC\_KEY |
| 10037 | 246 | DROP\_SYMMETRIC\_KEY |
| 10029 | 10031 | DDL\_USER\_EVENTS |
| 10031 | 132 | ALTER\_USER |
| 10031 | 131 | CREATE\_USER |
| 10031 | 133 | DROP\_USER |
| 10016 | 10052 | DDL\_DEFAULT\_EVENTS |
| 10052 | 218 | BIND\_DEFAULT |
| 10052 | 220 | CREATE\_DEFAULT |
| 10052 | 231 | DROP\_DEFAULT |
| 10052 | 242 | UNBIND\_DEFAULT |
| 10016 | 10026 | DDL\_EVENT\_NOTIFICATION\_EVENTS |
| 10026 | 74 | CREATE\_EVENT\_NOTIFICATION |
| 10026 | 76 | DROP\_EVENT\_NOTIFICATION |
| 10016 | 10053 | DDL\_EXTENDED\_PROPERTY\_EVENTS |
| 10053 | 211 | ALTER\_EXTENDED\_PROPERTY |
| 10053 | 222 | CREATE\_EXTENDED\_PROPERTY |
| 10053 | 233 | DROP\_EXTENDED\_PROPERTY |
| 10016 | 10054 | DDL\_FULLTEXT\_CATALOG\_EVENTS |
| 10054 | 212 | ALTER\_FULLTEXT\_CATALOG |
| 10054 | 223 | CREATE\_FULLTEXT\_CATALOG |
| 10054 | 234 | DROP\_FULLTEXT\_CATALOG |
| 10016 | 10067 | DDL\_FULLTEXT\_STOPLIST\_EVENTS |
| 10067 | 294 | ALTER\_FULLTEXT\_STOPLIST |
| 10067 | 293 | CREATE\_FULLTEXT\_STOPLIST |
| 10067 | 295 | DROP\_FULLTEXT\_STOPLIST |
| 10016 | 10023 | DDL\_FUNCTION\_EVENTS |
| 10023 | 62 | ALTER\_FUNCTION |
| 10023 | 61 | CREATE\_FUNCTION |
| 10023 | 63 | DROP\_FUNCTION |
| 10016 | 10049 | DDL\_PARTITION\_EVENTS |
| 10049 | 10050 | DDL\_PARTITION\_FUNCTION\_EVENTS |
| 10050 | 192 | ALTER\_PARTITION\_FUNCTION |
| 10050 | 191 | CREATE\_PARTITION\_FUNCTION |
| 10050 | 193 | DROP\_PARTITION\_FUNCTION |
| 10049 | 10051 | DDL\_PARTITION\_SCHEME\_EVENTS |
| 10051 | 195 | ALTER\_PARTITION\_SCHEME |
| 10051 | 194 | CREATE\_PARTITION\_SCHEME |
| 10051 | 196 | DROP\_PARTITION\_SCHEME |
| 10016 | 10055 | DDL\_PLAN\_GUIDE\_EVENTS |
| 10055 | 216 | ALTER\_PLAN\_GUIDE |
| 10055 | 228 | CREATE\_PLAN\_GUIDE |
| 10055 | 238 | DROP\_PLAN\_GUIDE |
| 10016 | 10024 | DDL\_PROCEDURE\_EVENTS |
| 10024 | 52 | ALTER\_PROCEDURE |
| 10024 | 51 | CREATE\_PROCEDURE |
| 10024 | 53 | DROP\_PROCEDURE |
| 10016 | 10056 | DDL\_RULE\_EVENTS |
| 10056 | 219 | BIND\_RULE |
| 10056 | 229 | CREATE\_RULE |
| 10056 | 239 | DROP\_RULE |
| 10056 | 243 | UNBIND\_RULE |
| 10016 | 10069 | DDL\_SEARCH\_PROPERTY\_LIST\_EVENTS |
| 10069 | 298 | ALTER\_SEARCH\_PROPERTY\_LIST |
| 10069 | 297 | CREATE\_SEARCH\_PROPERTY\_LIST |
| 10069 | 299 | DROP\_SEARCH\_PROPERTY\_LIST |
| 10016 | 10070 | DDL\_SEQUENCE\_EVENTS |
| 10070 | 304 | ALTER\_SEQUENCE |
| 10070 | 303 | CREATE\_SEQUENCE |
| 10070 | 305 | DROP\_SEQUENCE |
| 10016 | 10041 | DDL\_SSB\_EVENTS |
| 10041 | 10063 | DDL\_BROKER\_PRIORITY\_EVENTS |
| 10063 | 282 | ALTER\_BROKER\_PRIORITY |
| 10063 | 281 | CREATE\_BROKER\_PRIORITY |
| 10063 | 283 | DROP\_BROKER\_PRIORITY |
| 10041 | 10043 | DDL\_CONTRACT\_EVENTS |
| 10043 | 154 | CREATE\_CONTRACT |
| 10043 | 156 | DROP\_CONTRACT |
| 10041 | 10042 | DDL\_MESSAGE\_TYPE\_EVENTS |
| 10042 | 152 | ALTER\_MESSAGE\_TYPE |
| 10042 | 151 | CREATE\_MESSAGE\_TYPE |
| 10042 | 153 | DROP\_MESSAGE\_TYPE |
| 10041 | 10044 | DDL\_QUEUE\_EVENTS |
| 10044 | 158 | ALTER\_QUEUE |
| 10044 | 157 | CREATE\_QUEUE |
| 10044 | 159 | DROP\_QUEUE |
| 10041 | 10047 | DDL\_REMOTE\_SERVICE\_BINDING\_EVENTS |
| 10047 | 175 | ALTER\_REMOTE\_SERVICE\_BINDING |
| 10047 | 174 | CREATE\_REMOTE\_SERVICE\_BINDING |
| 10047 | 176 | DROP\_REMOTE\_SERVICE\_BINDING |
| 10041 | 10046 | DDL\_ROUTE\_EVENTS |
| 10046 | 165 | ALTER\_ROUTE |
| 10046 | 164 | CREATE\_ROUTE |
| 10046 | 166 | DROP\_ROUTE |
| 10041 | 10045 | DDL\_SERVICE\_EVENTS |
| 10045 | 162 | ALTER\_SERVICE |
| 10045 | 161 | CREATE\_SERVICE |
| 10045 | 163 | DROP\_SERVICE |
| 10016 | 10022 | DDL\_SYNONYM\_EVENTS |
| 10022 | 34 | CREATE\_SYNONYM |
| 10022 | 36 | DROP\_SYNONYM |
| 10016 | 10017 | DDL\_TABLE\_VIEW\_EVENTS |
| 10017 | 10020 | DDL\_INDEX\_EVENTS |
| 10020 | 213 | ALTER\_FULLTEXT\_INDEX |
| 10020 | 25 | ALTER\_INDEX |
| 10020 | 224 | CREATE\_FULLTEXT\_INDEX |
| 10020 | 24 | CREATE\_INDEX |
| 10020 | 274 | CREATE\_SPATIAL\_INDEX |
| 10020 | 206 | CREATE\_XML\_INDEX |
| 10020 | 235 | DROP\_FULLTEXT\_INDEX |
| 10020 | 26 | DROP\_INDEX |
| 10017 | 10021 | DDL\_STATISTICS\_EVENTS |
| 10021 | 27 | CREATE\_STATISTICS |
| 10021 | 29 | DROP\_STATISTICS |
| 10021 | 28 | UPDATE\_STATISTICS |
| 10017 | 10018 | DDL\_TABLE\_EVENTS |
| 10018 | 22 | ALTER\_TABLE |
| 10018 | 21 | CREATE\_TABLE |
| 10018 | 23 | DROP\_TABLE |
| 10017 | 10019 | DDL\_VIEW\_EVENTS |
| 10019 | 42 | ALTER\_VIEW |
| 10019 | 41 | CREATE\_VIEW |
| 10019 | 43 | DROP\_VIEW |
| 10016 | 10025 | DDL\_TRIGGER\_EVENTS |
| 10025 | 72 | ALTER\_TRIGGER |
| 10025 | 71 | CREATE\_TRIGGER |
| 10025 | 73 | DROP\_TRIGGER |
| 10016 | 10028 | DDL\_TYPE\_EVENTS |
| 10028 | 91 | CREATE\_TYPE |
| 10028 | 93 | DROP\_TYPE |
| 10016 | 10048 | DDL\_XML\_SCHEMA\_COLLECTION\_EVENTS |
| 10048 | 178 | ALTER\_XML\_SCHEMA\_COLLECTION |
| 10048 | 177 | CREATE\_XML\_SCHEMA\_COLLECTION |
| 10048 | 179 | DROP\_XML\_SCHEMA\_COLLECTION |
| 10016 | 241 | RENAME |
| 10001 | 10002 | DDL\_SERVER\_LEVEL\_EVENTS |
| 10002 | 214 | ALTER\_INSTANCE |
| 10002 | 10071 | DDL\_AVAILABILITY\_GROUP\_EVENTS |
| 10071 | 307 | ALTER\_AVAILABILITY\_GROUP |
| 10071 | 306 | CREATE\_AVAILABILITY\_GROUP |
| 10071 | 308 | DROP\_AVAILABILITY\_GROUP |
| 10002 | 10004 | DDL\_DATABASE\_EVENTS |
| 10004 | 202 | ALTER\_DATABASE |
| 10004 | 201 | CREATE\_DATABASE |
| 10004 | 203 | DROP\_DATABASE |
| 10002 | 10003 | DDL\_ENDPOINT\_EVENTS |
| 10003 | 182 | ALTER\_ENDPOINT |
| 10003 | 181 | CREATE\_ENDPOINT |
| 10003 | 183 | DROP\_ENDPOINT |
| 10002 | 10057 | DDL\_EVENT\_SESSION\_EVENTS |
| 10057 | 265 | ALTER\_EVENT\_SESSION |
| 10057 | 264 | CREATE\_EVENT\_SESSION |
| 10057 | 266 | DROP\_EVENT\_SESSION |
| 10002 | 10011 | DDL\_EXTENDED\_PROCEDURE\_EVENTS |
| 10011 | 221 | CREATE\_EXTENDED\_PROCEDURE |
| 10011 | 232 | DROP\_EXTENDED\_PROCEDURE |
| 10002 | 10012 | DDL\_LINKED\_SERVER\_EVENTS |
| 10012 | 263 | ALTER\_LINKED\_SERVER |
| 10012 | 225 | CREATE\_LINKED\_SERVER |
| 10012 | 10013 | DDL\_LINKED\_SERVER\_LOGIN\_EVENTS |
| 10013 | 226 | CREATE\_LINKED\_SERVER\_LOGIN |
| 10013 | 236 | DROP\_LINKED\_SERVER\_LOGIN |
| 10012 | 262 | DROP\_LINKED\_SERVER |
| 10002 | 10014 | DDL\_MESSAGE\_EVENTS |
| 10014 | 215 | ALTER\_MESSAGE |
| 10014 | 227 | CREATE\_MESSAGE |
| 10014 | 237 | DROP\_MESSAGE |
| 10002 | 10015 | DDL\_REMOTE\_SERVER\_EVENTS |
| 10015 | 217 | ALTER\_REMOTE\_SERVER |
| 10015 | 230 | CREATE\_REMOTE\_SERVER |
| 10015 | 240 | DROP\_REMOTE\_SERVER |
| 10002 | 10058 | DDL\_RESOURCE\_GOVERNOR\_EVENTS |
| 10058 | 273 | ALTER\_RESOURCE\_GOVERNOR\_CONFIG |
| 10058 | 10059 | DDL\_RESOURCE\_POOL |
| 10059 | 268 | ALTER\_RESOURCE\_POOL |
| 10059 | 267 | CREATE\_RESOURCE\_POOL |
| 10059 | 269 | DROP\_RESOURCE\_POOL |
| 10058 | 10060 | DDL\_WORKLOAD\_GROUP |
| 10060 | 271 | ALTER\_WORKLOAD\_GROUP |
| 10060 | 270 | CREATE\_WORKLOAD\_GROUP |
| 10060 | 272 | DROP\_WORKLOAD\_GROUP |
| 10002 | 10005 | DDL\_SERVER\_SECURITY\_EVENTS |
| 10005 | 209 | ADD\_SERVER\_ROLE\_MEMBER |
| 10005 | 301 | ALTER\_SERVER\_ROLE |
| 10005 | 300 | CREATE\_SERVER\_ROLE |
| 10005 | 10008 | DDL\_AUTHORIZATION\_SERVER\_EVENTS |
| 10008 | 204 | ALTER\_AUTHORIZATION\_SERVER |
| 10005 | 10009 | DDL\_CREDENTIAL\_EVENTS |
| 10009 | 260 | ALTER\_CREDENTIAL |
| 10009 | 259 | CREATE\_CREDENTIAL |
| 10009 | 261 | DROP\_CREDENTIAL |
| 10005 | 10061 | DDL\_CRYPTOGRAPHIC\_PROVIDER\_EVENTS |
| 10061 | 276 | ALTER\_CRYPTOGRAPHIC\_PROVIDER |
| 10061 | 275 | CREATE\_CRYPTOGRAPHIC\_PROVIDER |
| 10061 | 277 | DROP\_CRYPTOGRAPHIC\_PROVIDER |
| 10005 | 10007 | DDL\_GDR\_SERVER\_EVENTS |
| 10007 | 168 | DENY\_SERVER |
| 10007 | 167 | GRANT\_SERVER |
| 10007 | 169 | REVOKE\_SERVER |
| 10005 | 10006 | DDL\_LOGIN\_EVENTS |
| 10006 | 145 | ALTER\_LOGIN |
| 10006 | 144 | CREATE\_LOGIN |
| 10006 | 146 | DROP\_LOGIN |
| 10005 | 10064 | DDL\_SERVER\_AUDIT\_EVENTS |
| 10064 | 285 | ALTER\_SERVER\_AUDIT |
| 10064 | 284 | CREATE\_SERVER\_AUDIT |
| 10064 | 286 | DROP\_SERVER\_AUDIT |
| 10005 | 10065 | DDL\_SERVER\_AUDIT\_SPECIFICATION\_EVENTS |
| 10065 | 288 | ALTER\_SERVER\_AUDIT\_SPECIFICATION |
| 10065 | 287 | CREATE\_SERVER\_AUDIT\_SPECIFICATION |
| 10065 | 289 | DROP\_SERVER\_AUDIT\_SPECIFICATION |
| 10005 | 10010 | DDL\_SERVICE\_MASTER\_KEY\_EVENTS |
| 10010 | 251 | ALTER\_SERVICE\_MASTER\_KEY |
| 10005 | 302 | DROP\_SERVER\_ROLE |
| 10005 | 210 | DROP\_SERVER\_ROLE\_MEMBER |

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

[](https://www.dotnettips.info/file/image?name=04-144e4f422aae4661a9e3255b00ceaa7e.png)

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:

[https://www.dotnettips.info/file/image?name=05-d8fe129937514a23b34019c1825e4958.png](https://www.dotnettips.info/file/image?name=05-d8fe129937514a23b34019c1825e4958.png)

پس از Overrdie کردن می‌توانید مجدداً Trigger را فعال کنید:

[https://www.dotnettips.info/file/image?name=06-b50e23137905408c961527cf3365dbd4.png](https://www.dotnettips.info/file/image?name=06-b50e23137905408c961527cf3365dbd4.png)

* DISABLE TRIGGER ALL ON DATABASE = **DDL triggers**