



доц. д-р Цветанка Георгиева-Трифенова

DDL ТРИГЕРИ



СЪДЪРЖАНИЕ

- ✗ DDL тригери – същност
- ✗ Създаване на DDL тригери
- ✗ Функцията EVENTDATA ()
- ✗ DDL тригери – приложение
- ✗ DDL тригери – примери

DDL ТРИГЕРИ

- ✗ DDL – *Data Definition Language*;
- ✗ Задействат се при изпълняване на команди за дефиниране на данни:
 - + CREATE DATABASE, DROP DATABASE
 - + CREATE TABLE, ALTER TABLE, DROP TABLE
 - + ...

СЪЗДАВАНЕ DDL ТРИГЕРИ

```
CREATE TRIGGER trigger_name
  ON { ALL SERVER | DATABASE }
  { FOR | AFTER }
  { event_type | event_group } [ , ... n ]
AS
  sql_statements
```

- ✗ Дефинират се на ниво сървър или на ниво база от данни;
- ✗ *event_type* – при кои действия за дефиниране на данни да се стартира тригерът;
- ✗ *event_group* – групи от действия, чрез които тригерът се стартира за всяко действие, включено в групата.

СЪЗДАВАНЕ DDL ТРИГЕРИ (2)

	Server Scope	Database Scope
DDL_SERVER_LEVEL_EVENTS (CREATE DATABASE, ALTER DATABASE, DROP DATABASE)	X	
DDL_ENDPOINT_EVENTS (CREATE ENDPOINT, ALTER ENDPOINT, DROP ENDPOINT)	X	
DDL_SERVER_SECURITY_EVENTS	X	
DDL_LOGIN_EVENTS (CREATE LOGIN, ALTER LOGIN, DROP LOGIN)	X	
DDL_GDR_SERVER_EVENTS (GRANT SERVER, DENY SERVER, REVOKE SERVER)	X	
DDL_AUTHORIZATION_SERVER_EVENTS (ALTER AUTHORIZATION SERVER)	X	
DDL_DATABASE_LEVEL_EVENTS		X
DDL_TABLE_VIEW_EVENTS		X
DDL_TABLE_EVENTS (CREATE TABLE, ALTER TABLE, DROP TABLE)		X
DDL_VIEW_EVENTS (CREATE VIEW, ALTER VIEW, DROP VIEW)		X
DDL_INDEX_EVENTS (CREATE INDEX, ALTER INDEX, DROP INDEX, CREATE XML INDEX)		X
DDL_STATISTICS_EVENTS (CREATE STATISTICS, UPDATE STATISTICS, DROP STATISTICS)		X

Някои групи от действия за дефиниране на данни

ФУНКЦИЯТА EVENTDATA ()

- ✗ Предоставя информация за действието, стартирало DDL тригера;
- ✗ Връща стойност от тип *xml*;
- ✗ XML схемата включва информация за:
 - + времето, когато действието се е изпълнило `PostTime`;
 - + идентификатора на системния процес `SPID` на конекцията, от която тригерът се изпълнява;
 - + типа на действието `EventType`, стартирало тригера;
 - + други.

ФУНКЦИЯТА EVENTDATA () (2)

<EVENT_INSTANCE>

<EventType>event</EventType>

<PostTime>date-time</PostTime>

<SPID>spid</SPID>

<ServerName>name</ServerName>

<LoginName>login </LoginName>

<UserName>name</UserName>

<DatabaseName>name</DatabaseName>

<SchemaName>name</SchemaName>

<ObjectName>name</ObjectName>

<ObjectType>type</ObjectType>

<TSQLCommand>command</TSQLCommand>

</EVENT_INSTANCE>

ФУНКЦИЯТА EVENTDATA () – ПРИМЕРЕН РЕЗУЛТАТ

```
- <EVENT_INSTANCE>
  <EventType>CREATE_TABLE</EventType>
  <PostTime>2011-02-17T13:25:21.690</PostTime>
  <SPID>52</SPID>
  <ServerName>C_GEORGIEVA-PC\SQL_SERVER_2005</ServerName>
  <LoginName>sa</LoginName>
  <UserName>dbo</UserName>
  <DatabaseName>MyDatabase</DatabaseName>
  <SchemaName>dbo</SchemaName>
  <ObjectName>NewTable</ObjectName>
  <ObjectType>TABLE</ObjectType>
- <TSQLCommand>
  <SetOptions ANSI_NULLS="ON" ANSI_NULL_DEFAULT="ON" ANSI_PADDING="ON"
    QUOTED_IDENTIFIER="ON" ENCRYPTED="FALSE" />
  <CommandText>CREATE TABLE NewTable ( ID int, Descrption varchar(50) )
  </CommandText>
</TSQLCommand>
</EVENT_INSTANCE>
```


DDL ТРИГЕРИ – ПРИЛОЖЕНИЕ

- ✗ Проследяване на изпълняваните действия за дефиниране на данни;
 - + DDL тригерът може да записва в `log` таблица информация за изпълнените действия за дефиниране на данни, които проследява.
- ✗ Контролиране на изпълняваните действия за дефиниране на данни.
 - + Възможно е например, да се дефинира тригер, който да не допуска създаването на таблица, чието наименование не започва с `tbl`.

DDL ТРИГЕРИ – ПРИМЕР

- ✗ Тригер, който не позволява създаване на таблици, чиито имена нямат префикс `tbl`.

```
CREATE TRIGGER ddltrg_CheckTableName
ON DATABASE
FOR CREATE_TABLE
AS
DECLARE @xmlEventData xml, @tableName varchar(50)

SET @xmlEventData = EVENTDATA()
SET @tableName = CONVERT(varchar(25),
    @xmlEventData.query('data(/EVENT_INSTANCE/Object
Name)'))

IF LEFT(@tableName, 3) <> 'tbl'
    BEGIN
        RAISERROR ( 'Не е допустимо създаване на
таблица, ако името не започва с tbl.', 16, 1 )
        ROLLBACK TRANSACTION
    END
GO
```


ЗАДЕЙСТВАНЕ НА ТРИГЕРА DDLTRG_CHECKTABLENAME

```
CREATE TABLE MyTable  
( ID int,  
  Description varchar(50) )
```

предизвиква извеждане на следното съобщение:

```
Msg 50000, Level 16, State 1, Procedure  
  dltrg_CheckTableName, Line 11
```

Не е допустимо създаване на таблица, ако името
не започва с tbl.

```
Msg 3609, Level 16, State 2, Line 2
```

The transaction ended in the trigger. The
batch has been aborted.

DDL ТРИГЕРИ – ПРИМЕР (2)

- ✗ Тригер, чрез който се записва информация за всички изпълнени действия за създаване на таблици в дадена база от данни.

DDL ТРИГЕРИ – ПРИМЕР (3)

- ✗ Първо се създава *log* таблица, в която ще се записва необходимата информация за изпълнените действия за създаване на таблици:

```
CREATE TABLE tblDDLEventLog
( ID int IDENTITY NOT NULL,
  EventTime datetime NULL,
  EventType varchar(15) NULL,
  ServerName varchar(50) NULL,
  DatabaseName varchar(25) NULL,
  ObjectType varchar(25) NULL,
  ObjectName varchar(25) NULL,
  UserName varchar(15) NULL,
  CommandText varchar(max) NULL )
```


DDL ТРИГЕРИ – ПРИМЕР (4)

```
CREATE TRIGGER ddltrg_CREATE_TABLE_LOG
    ON DATABASE
FOR CREATE_TABLE
AS
DECLARE @xmlEventData xml

SET @xmlEventData = EVENTDATA()

INSERT INTO tblDDLEventLog
( EventTime, EventType, ServerName, DatabaseName,
  ObjectType, ObjectName, UserName, CommandText )

SELECT CONVERT(varchar(50),
@xmlEventData.query('data(/EVENT_INSTANCE/PostTime)')),
        CONVERT(varchar(15),
@xmlEventData.query('data(/EVENT_INSTANCE/EventType)')),
```

```
CONVERT (varchar (50) ,  
@xmlEventData.query ('data (/EVENT_INSTANCE/ServerName) ')) ,  
        CONVERT (varchar (25) ,  
@xmlEventData.query ('data (/EVENT_INSTANCE/DatabaseName) ')) ,  
        CONVERT (varchar (25) ,  
@xmlEventData.query ('data (/EVENT_INSTANCE/Object Type) ')) ,  
        CONVERT (varchar (25) ,  
@xmlEventData.query ('data (/EVENT_INSTANCE/Object Name) ')) ,  
        CONVERT (varchar (15) ,  
@xmlEventData.query ('data (/EVENT_INSTANCE/UserName) ')) ,  
        CONVERT (varchar (max) ,  
@xmlEventData.query ('data (/EVENT_INSTANCE/TSQLCommand/CommandText) '))
```

GO

DDL ТРИГЕРИ – ПРИМЕР (5)

- ✗ Примерно съдържание на таблицата `tblDDLEventLog`

ID	EventTime	EventType	ServerName	DatabaseName	ObjectType	ObjectName	UserName	CommandText
1	2011-02-17 13:01:22.530	CREATE_TABLE	C_GEORGIEVA-PC\SQL_SERVER_2005	MyDatabase	TABLE	tblDDLEventLog	dbo	CREATE TABLE tblDDLEve
2	2011-02-17 13:01:51.027	CREATE_TABLE	C_GEORGIEVA-PC\SQL_SERVER_2005	MyDatabase	TABLE	NewTable	dbo	CREATE TABLE NewTable

ЗАДАЧИ

- ✖ **Задача 1.** Да се създаде DDL тригер, който проследява промените в структурите на таблиците. Да се записва в log таблица времето, името на потребителя, типа и текста на Transact SQL командата на всяко действие за промяна на таблица.
- ✖ **Задача 2.** Да се създаде DDL тригер, който не позволява създаване на изгледи, чиито имена нямат префикс `vw`.
- ✖ **Задача 3.** Да се създаде DDL тригер, който извежда съобщение при изпълняване на някоя от командите за създаване или изтриване на база от данни (`CREATE DATABASE` или `DROP DATABASE`) и извежда текста на изпълнената Transact-SQL командата.



Цветанка Георгиева-Трифорова, 2017

Някои права запазени.

Презентацията е достъпна под лиценз Creative Commons,

Признание-Некомерсиално-Без производни,

<https://creativecommons.org/licenses/by-nc-nd/4.0/legalcode>