

Триггеры

Лекция №9

Бутенко И.В. 2017 год

Определение

Триггеры – это методы, с помощью которых разработчик БД может обеспечить ее целостность. Триггером называют хранимую процедуру особого типа, которая автоматически выполняется при возникновении языкового события.

SQL Server поддерживает два основных типа триггеров:

- DML-триггеры
- DDL-триггеры.

DML-триггеры

DML-триггеры выполняются при возникновении событий языка манипулирования данными (INSERT, UPDATE, DELETE) в базе данных

Триггеры выполняются после применения правил и других проверок целостности ссылок, поэтому, если операция не проходит эти предварительные проверки, триггеры не выполняются.

Применение DML-триггеров

- Для обеспечения целостности
- Для создания бизнес – правил
- Как дополнительное средство безопасности

DDL-триггеры

Они срабатывают в ответ на некоторые инструкции языка определения данных (CREATE, ALTER, DROP, GRANT, DENY, REVOKE, UPDATE STATISTICS).

Триггеры DDL срабатывают только после выполнения соответствующих инструкций DDL.

Применение DDL-триггеров

- Предотвратить внесение определенных изменений в схему базы данных.
- Выполнить в базе данных некоторые действия в ответ на изменения в схеме базы данных.
- Записывать изменения или события схемы базы данных.

Создание DML-триггера

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

Создание DDL-триггера

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

Удаление и отключение триггера

- Удаление:

```
DROP TRIGGER trigger_name [ ,...n ]  
ON { DATABASE | ALL SERVER }
```

- Отключение:

```
DISABLE TRIGGER { [ schema . ] trigger_name [ ,...n ] |  
  ALL }  
ON { object_name | DATABASE | ALL SERVER } [ ; ]
```

- Включение:

```
ENABLE TRIGGER { [ schema_name . ] trigger_name [ ,...n ] | ALL }  
ON { object_name | DATABASE | ALL SERVER } [ ; ]
```

Пример 1

```
drop TRIGGER tr1
go
CREATE TRIGGER tr1
  ON students
  AFTER INSERT,DELETE,UPDATE
AS
BEGIN
  print 'Работа триггера'
END
go
```

```
DISABLE TRIGGER tr1 on students
ENABLE TRIGGER tr1 on students
```

```
update students
set name = 'Махмут'
where id = 5
```

Программирование DML-триггеров 1

Для отслеживания изменений в таблицах сервер автоматически создает при вызове триггера две специальные таблицы – `inserted` и `deleted`.

Таблицы `inserted` и `deleted` не существуют физически, а представляют собой логические структуры, создаваемые сервером индивидуально для каждого триггера. Эти таблицы доступны в режиме `read only`.

Программирование DML-триггеров 2

Оператор	Содержимое inserted	Содержимое deleted
Insert	Добавленные строки	Пусто
Update	Новые строки	Старые строки
Delete	Пусто	Удаленные строки

Пример 2

```
drop trigger tr2
go
create trigger tr2
  on students
  after update
as
begin
  select * from inserted
  select * from deleted
end
go
```

```
update students
set name = 'Иван'
where name = 'Иван'
```

Пример 3

```
drop trigger tr3
go
create trigger tr3
  on subjects
  instead of delete
as
begin

  if not exists (select * from deleted where name like 'Б%')
    delete from subjects
    where id in (select id from deleted)

end
go
```

Пример 4

```
CREATE TABLE ddl_log (PostTime datetime, DB_User nvarchar(100), Event nvarchar(100), TSQL
    nvarchar(2000));
GO

drop TRIGGER tr4 ON database
go
CREATE TRIGGER tr4
    ON database FOR DROP_TABLE, ALTER_TABLE
AS
BEGIN
DECLARE @data XML
SET @data = EVENTDATA()
INSERT ddl_log
    (PostTime, DB_User, Event, TSQL)
    VALUES
    (GETDATE(),
    CONVERT(nvarchar(100), CURRENT_USER),
    @data.value('/EVENT_INSTANCE/EventType)[1]', 'nvarchar(100)'),
    @data.value('/EVENT_INSTANCE/TSQLCommand)[1]', 'nvarchar(2000)) );
END
GO
```