

Триггеры

Триггер (trigger) -

объект базы данных, представляющий собой специализированную хранимую процедуру, которая вызывается автоматически при модификации записей (DML), при модификации объектов БД (DDL) или при входе в систему

Триггеры :

типы триггеров

- триггер DML;
- триггер DDL;
- триггер входа.

Триггер :

фазы работы триггера:

- ❑ **FOR** (срабатывает до изменения данных);
- ❑ **AFTER** (срабатывает после изменения данных);
- ❑ **INSTEAD OF ...** (срабатывает в место указанного события, только для **DML**).

Триггеры DML :

привязан к таблице и срабатывает на одно или несколько событий

События триггера :

INSERT;

DELETE;

UPDATE.

Триггеры DML :

форма создания триггера DML

```
CREATE TRIGGER [<схема>] .<имя триггера>  
ON <таблица> | <представление>  
[WITH ENCRYPTION]  
FOR | AFTER | INSTEAD OF  
INSERT | UPDATE | DELETE  
AS  
    BEGIN  
        --тело триггера  
    END  
GO
```

Триггеры DML :

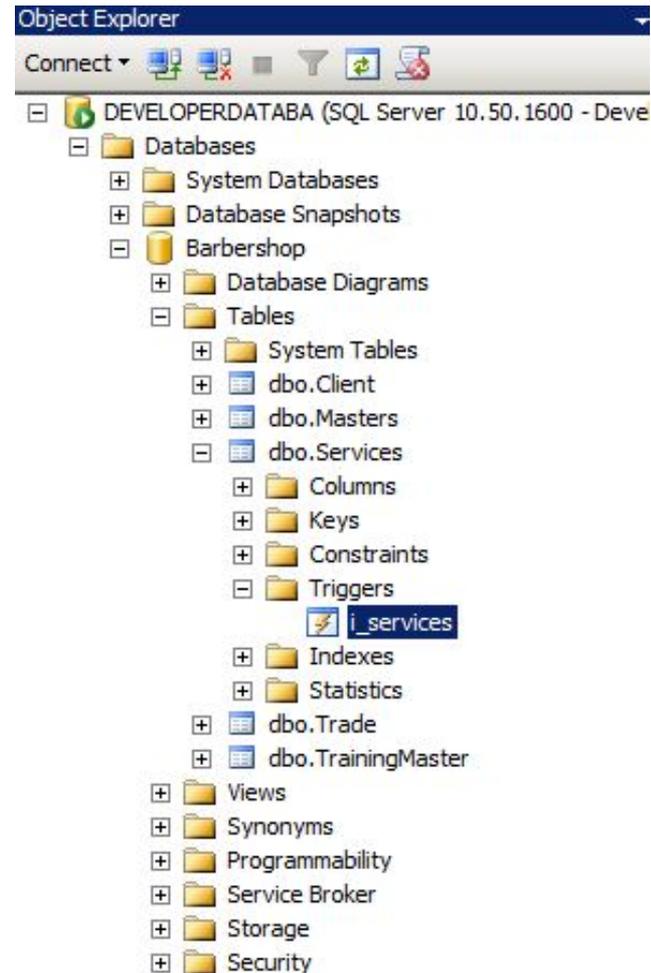
Упражнение 1

В базе данных Barbershop создать триггер `i_service`, не позволяющий добавлять данные в таблицу `Services`

```
CREATE TRIGGER i_services
ON Services
FOR INSERT
AS
BEGIN
    ROLLBACK TRANSACTION
    PRINT 'Данные не добавлены в таблицу Services'
END
GO
```

Триггеры DML :

Созданный триггер привязывается к таблице чьи события отлавливает



Триггеры DML:

При выполнении команды **INSERT**, данные вначале добавляются в логическую таблицу **inserted**

Триггеры DML :

Упражнение 2

В базе данных Barbershop создать триггер `i_tradeTime_Trade`, запрещающий добавление пустых значений в поле `tradeTime` таблицы `Trade`

```
CREATE TRIGGER i_tradeTime_Trade
ON Trade
FOR INSERT
AS
IF EXISTS (SELECT *
           FROM inserted
           WHERE tradeTime is NULL)
BEGIN
    RAISERROR('Нужно обязательно указать время', 10,1)
    ROLLBACK TRANSACTION
END
GO
```

RAISERROR('сообщение', степень серьезности, состояние)

Триггеры DML :

При выполнении команды **DELETE**, данные перед удалением помещаются в таблицу **deleted**

Триггеры DML :

Упражнение 3

В базе данных Barbershop создать триггер d_Client, запрещающий удаление клиента, если он имеет связь с таблицей Trade

```
CREATE TRIGGER u_Client
ON Client
FOR DELETE
AS
DECLARE @clientID INT, @temp int
IF ((SELECT COUNT(*)
      FROM Trade, deleted
      WHERE Trade.clientID = deleted.clientID) > 0)
BEGIN
    PRINT 'Запись не удалена'
    ROLLBACK TRANSACTION
END
GO
```

Триггеры DML :

При выполнении команды UPDATE, запись вначале удаляется (таблица `deleted`), а затем вставляются (таблица `inserted`)

Триггеры DML :

Упражнение 4

В базе данных Barbershop создать триггер `u_tradeDate_Trade`, запрещающий редактировать дату записи на выполнение услуги

```
CREATE TRIGGER u_tradeDate_Trade
ON Trade
FOR UPDATE
AS
    BEGIN
        IF UPDATE (tradeDate)
            BEGIN
                PRINT 'Нельзя редактировать дату записи!'
                ROLLBACK TRANSACTION
            END
        END
    END
GO
```

Триггеры DDL :

привязан к базе данных или к текущему экземпляру,
срабатывает на одно или несколько событий

События триггера :

CREATE;

ALTER;

DROP.

Триггеры DDL :

форма создания триггера DDL

```
CREATE TRIGGER [<схема>] .<имя триггера>  
ON <ALL SERVER> | <DATABASE>  
[WITH ENCRYPTION]  
FOR | AFTER  
CREATE | ALTER | DROP  
AS  
    BEGIN  
        --тело триггера  
    END  
GO
```

Триггеры входа :

привязан к текущему экземпляру, срабатывает при соединении пользователя с экземпляром сервера

Триггеры входа :

форма создания триггера входа

```
CREATE TRIGGER [<схема>] .<имя триггера>  
ON ALL SERVER  
[WITH ENCRYPTION]  
FOR | AFTER  
LOGON  
AS  
    BEGIN  
        --тело триггера  
    END  
GO
```