

Практическая работа № 14

Триггеры

1. Цель работы

1. Изучить создание триггеров.
2. Изучить триггеры после событий.
3. Изучить триггеры вместо событий.
4. Изучить виртуальные таблицы в триггерах.
5. Изучить приостановление триггеров.
6. Изучить удаление триггеров.

2. Теоретическая часть

Триггер – это вид хранимой процедуры, который вызывается автоматически при определенных событиях. Часто триггеры применяются для автоматической поддержки целостности и защиты БД.

В MS SQL Server существует *три вида триггеров*, которые отличаются по функциям и по синтаксису создания и изменения:

□ *Триггеры DML* вызываются при выполнении команд INSERT, UPDATE или DELETE. Можно создать триггер, реагирующий на две или на все три команды.

□ *Триггеры DDL* реагируют на события изменения структуры БД: создание, изменение или удаление отдельных объектов БД.

□ *Триггеры входа в систему* запускаются при соединении пользователя с экземпляром сервера. Их можно применять для дополнительной проверки полномочий пользователей.

Триггеры DML можно вызвать после событий (FOR | AFTER), или вместо него (INSTEAD OF).

Триггер AFTER выполняется после успешного завершения вызвавшего его события. Можно определить несколько AFTER-триггеров для каждой операции.

Триггер INSTEAD OF вызывается вместо выполнения команд. Для каждой операции INSERT, UPDATE, DELETE можно определить только один INSTEAD OF-триггер.

Упрощенный синтаксис создания триггера имеет следующий вид:

```
CREATE TRIGGER <название триггера> ON <название таблицы>  
<FOR | AFTER | INSTEAD OF> <INSERT | UPDATE | DELETE>  
AS  
[BEGIN]  
<команды>  
[END]
```

Ключевое слово FOR или AFTER указывает, что триггер DML срабатывает только после успешного запуска всех операций в инструкции SQL, по которой срабатывает триггер.

Ключевое слово INSTEAD OF указывает, что триггер DML выполняется вместо инструкции SQL, по которой он срабатывает, то есть переопределяет действия запускающих инструкций.

В определении триггера ключевые слова INSERT | UPDATE | DELETE определяют инструкции изменения данных, при применении которых к таблице или представлению срабатывает триггер DML. Указание хотя бы одного варианта обязательно. В определении триггера разрешены любые сочетания вариантов в любом порядке.

Триггеры не вызываются рекурсивно.

Хотя инструкция TRUNCATE TABLE по сути аналогичная инструкции DELETE, она не активирует триггер.

Если триггер выполняется для события добавления данных (команды INSERT), в теле триггера доступна виртуальная таблица INSERTED, которая содержит список добавленных данных.

Если триггер выполняется для события удаления данных (команды DELETE), в теле триггера доступна виртуальная таблица DELETED, которая содержит список удаленных данных.

Если триггер выполняется для события изменения данных (команды UPDATE), в теле триггера доступны две виртуальные таблицы INSERTED и DELETED, которые содержат список новых и старых данных, соответственно.

Если при определенных обстоятельствах выполнение триггера нежелательно, то можно его отключить. Для этого используется команда DISABLE TRIGGER, его синтаксис:

DISABLE TRIGGER <название триггера> ON <название таблицы>

А когда триггер снова понадобится, его можно включить с помощью команды ENABLE TRIGGER, его синтаксис:

ENABLE TRIGGER <название триггера> ON <название таблицы>

Для удаления триггера используется команда DROP TRIGGER, его синтаксис:

DROP TRIGGER <название триггера>

3. Практическая часть

Дана таблица *Ученики*:

ИД	Фамилия	Предмет	Школа	Баллы
1	Иванова	Математика	Лицей	98,5
2	Петров	Физика	Лицей	99
3	Сидоров	Математика	Лицей	88
4	Полухина	Физика	Гимназия	78

ИД	Фамилия	Предмет	Школа	Баллы
5	Матвеева	Химия	Лицей	92
6	Касимов	Химия	Гимназия	68
7	Нурулин	Математика	Гимназия	81
8	Авдеев	Физика	Лицей	87
9	Никитина	Химия	Лицей	94

Пример 1: Напишите триггер на добавление записи в таблицу «Ученики». Данный триггер, в случае успешного добавления данных, выводит «Запись добавлена»:

```
CREATE TRIGGER Пример1 ON Ученики
FOR INSERT
AS
BEGIN
    PRINT 'Запись добавлена'
END
```

Пример 2: Напишите триггер на удаление записи из таблицы «Ученики». Данный триггер, в случае успешного удаления данных, выводит «Запись удалена»:

```
CREATE TRIGGER Пример2 ON Ученики
AFTER DELETE
AS
BEGIN
    PRINT 'Запись удалена'
END
```

Пример 3: Напишите триггер на добавление, изменение и удаление данных для таблицы «Ученики». Данный триггер выводит «Таблица изменена»:

```
CREATE TRIGGER Пример3 ON Ученики
FOR INSERT, UPDATE, DELETE
AS
BEGIN
    PRINT 'Таблица изменена'
END
```

Пример 4: Напишите триггер на удаление записи из таблицы «Ученики». Данный триггер, при попытке удаления данных, выводит «Нельзя удалить данные»:

```
CREATE TRIGGER Пример4 ON Ученики
INSTEAD OF DELETE
AS
BEGIN
    PRINT 'Нельзя удалить данные'
END
```

Пример 5: Создать таблицу «Ученики_Архив», которая будет содержать все данные об удаленных учениках и даты их удаления. Написать триггер, который будет фиксировать в таблице «Ученики_Архив» данные ученика, удаленного из таблицы «Ученики»:

```
CREATE TABLE Ученики_Архив
(
    ID INT NOT NULL,
    Фамилия VARCHAR(50) NULL,
    Предмет VARCHAR(50) NULL,
    Школа VARCHAR(50) NULL,
    Баллы FLOAT NULL,
    Удалено DATETIME NOT NULL
)

CREATE TRIGGER Пример5 ON Ученики
FOR DELETE
AS
BEGIN
    INSERT
        Ученики_Архив
    SELECT
        ID,
        Фамилия,
        Предмет,
        Школа,
        Баллы,
        GETDATE() AS Удалено
    FROM
        DELETED
END
```

Пример 6: Напишите команды для приостановления и запуска триггера из примера 5:

```
DISABLE TRIGGER Пример5 ON Ученики
```

```
ENABLE TRIGGER Пример5 ON Ученики
```

Пример 7: Напишите команды для удаления триггера из примера 5:

```
DROP TRIGGER Пример5
```

4. Задание

1. Напишите триггер на изменение записи в таблице «Ученики». Данный триггер, в случае изменения данных, должен вывести «Запись изменена».
2. Напишите триггер на добавление и удаление записи из таблицы «Ученики». Данный триггер, в случае успешного добавления или удаления данных, должен вывести «Количество строк изменено».
3. Напишите триггер на добавление, изменение и удаление данных в таблице «Ученики». Данный триггер должен вывести «{Текущий пользователь} изменил таблицу. Время: {текущее время}».
4. Напишите триггер на изменение записи в таблице «Ученики». Данный триггер, при попытке изменения данных, должен вывести «Нельзя редактировать данные».
5. Создать таблицу «Ученики_{Ваша_фамилия}», которая будет содержать фамилии удаленных учеников и даты их удаления. Написать триггер, который будет фиксировать в таблице «Ученики_{Ваша_фамилия}» данные учеников при удалении из таблицы «Ученики», в том случае, если у них остались однофамильцы в таблице «Ученики».
6. Напишите команды для приостановления и запуска триггера из предыдущей задачи.
7. Напишите команды для удаления всех созданных вами триггеров.

