

Частное учреждение образования
«Колледж бизнеса и права»

УТВЕРЖДАЮ

Ведущий методист
колледжа

_____ Е.В. Паскал

«___» _____ 2022

Специальность: «Программное обеспечение информационных технологий»	Учебная дисциплина: «Базы данных и системы управления базами данных»
--	--

ЛАБОРАТОРНАЯ РАБОТА № 23

Инструкционно-технологическая карта

Тема: Создание, модификация и удаление триггеров.

Цель работы: научиться создавать, модифицировать и удалять триггеры.

Время выполнения: 4 часа

Содержание работы

1. Теоретические сведения для выполнения работы
2. Порядок выполнения работы
3. Пример выполнения работы
4. Контрольные вопросы
5. Литература

1. ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ ДЛЯ ВЫПОЛНЕНИЯ РАБОТЫ

Триггер – это механизм, который вызывается, когда в указанной таблице происходит определенное действие. Каждый триггер имеет следующие основные составляющие:

- ✓ имя, которое может содержать максимум 128 символов;
- ✓ действие, которым может быть или инструкция DML (INSERT, DELETE или UPDATE), или инструкция DDL. Таким образом, существует два типа триггеров: триггеры DML и триггеры DDL;
- ✓ исполнение. Исполнительная часть триггера обычно состоит из хранимой процедуры или пакета.

Создание триггера DML

Триггер создается с помощью инструкции CREATE TRIGGER, которая имеет следующий синтаксис:

CREATE TRIGGER [schema_name.] trigger_name
ON {table_name| view_name}

[WITH dml_trigger_option[,...]]
 {FOR | AFTER | INSTEAD OF} {[INSERT] [,] [UPDATE] [,] [DELETE]}
 [WITH APPEND] {AS sql_statement| EXTERNAL NAME method_name}

Параметр	Значение параметра
schema_name.	имя схемы, к которой принадлежит триггер
trigger_name	имя триггера
table_name	имя таблицы, для которой создается триггер
select_statement	необходимо задать инструкцию SELECT, которая извлекает строки и столбцы из таблиц (или других представлений). См. замечание 2
FOR AFTER INSTEAD OF	указывают тип триггера: триггер AFTER (FOR является синонимом параметра AFTER) или триггер INSTEAD OF. См. замечание 1.
INSERT, UPDATE, DELETE	действие триггера. Под действием триггера имеется в виду инструкция T-SQL, которая запускает триггер. Допускается любая комбинация этих трех инструкций.
sql_statement	действие или действия

Замечание 1. Триггеры типа AFTER вызываются после выполнения действия, запускающего триггер, а триггеры типа INSTEAD OF выполняются вместо действия, запускающего триггер. Триггеры AFTER можно создавать только для таблиц, а триггеры INSTEAD — как для таблиц, так и для представлений.

Замечание 2. SQL Server позволяет создавать несколько триггеров для каждой таблицы и для каждого действия (INSERT, UPDATE и DELETE). По умолчанию определенного порядка исполнения нескольких триггеров для данного модифицирующего действия не имеется.

Изменение структуры триггера

Язык Transact-SQL также поддерживает инструкцию **ALTER TRIGGER**, которая модифицирует структуру триггера. Эта инструкция обычно применяется для изменения тела триггера. Все предложения и параметры инструкции ALTER TRIGGER имеют такое же значение, как и одноименные предложения и параметры инструкции CREATE TRIGGER.

Для удаления триггеров в текущей базе данных применяется инструкция **DROP TRIGGER**.

Использование виртуальных таблиц deleted и inserted

При создании действия триггера обычно требуется указать, ссылается ли он на значение столбца до или после его изменения действием, запускающим триггер. По этой причине, для тестирования следствия инструкции, запускающей триггер, используются две специально именованные виртуальные таблицы:

deleted — содержит копии строк, удаленных из таблицы;

inserted — содержит копии строк, вставленных в таблицу.

Структура этих таблиц эквивалентна структуре таблицы, для которой определен триггер.

Таблица deleted используется в том случае, если в инструкции CREATE TRIGGER указывается предложение DELETE или UPDATE, а если в этой инструкции указывается предложение INSERT или UPDATE, то используется таблица inserted. Это означает, что для каждой инструкции DELETE, выполненной в действии триггера, создается таблица deleted. Подобным образом для каждой инструкции INSERT, выполненной в действии триггера, создается таблица inserted.

Инструкция UPDATE рассматривается, как инструкция DELETE, за которой следует инструкция INSERT. Поэтому для каждой инструкции UPDATE, выполненной в действии триггера, создается как таблица deleted, так и таблица inserted (в указанной последовательности).

Триггеры AFTER

Триггеры AFTER вызываются после того, как выполняется действие, запускающее триггер. Триггер AFTER задается с помощью ключевого слова AFTER или FOR. Триггеры AFTER можно создавать только для базовых таблиц.

Триггеры INSTEAD OF

Триггер с предложением INSTEAD OF заменяет соответствующее действие, которое запустило его. Этот триггер выполняется после создания соответствующих таблиц inserted и deleted, но перед выполнением проверки ограничений целостности или каких-либо других действий.

Триггеры INSTEAD OF можно создавать как для таблиц, так и для представлений. Когда инструкция Transact-SQL ссылается на представление, для которого определен триггер INSTEAD OF, система баз данных выполняет этот триггер вместо выполнения любых действий с любой таблицей. Данный тип триггера всегда использует информацию в таблицах inserted и deleted, созданных для представления, чтобы создать любые инструкции, требуемые для создания запрошенного события.

Значения столбцов, предоставляемые триггером INSTEAD OF, должны удовлетворять определенным требованиям:

- ✓ значения не могут задаваться для вычисляемых столбцов;
- ✓ значения не могут задаваться для столбцов с типом данных TIMESTAMP;
- ✓ значения не могут задаваться для столбцов со свойством IDENTITY, если только параметру IDENTITY_INSERT не присвоено значение ON.

Эти требования действительны только для инструкций INSERT и UPDATE, которые ссылаются на базовые таблицы.

Триггеры DDL и области их применения SQL Server позволяет определять триггеры для инструкций DDL, таких как CREATE DATABASE, DROP TABLE и ALTER TABLE.

Триггеры для инструкций DDL имеют следующий синтаксис:

```
CREATE TRIGGER [schema_name.]trigger_name
ON {ALL SERVER | DATABASE}
[WITH {ENCRYPTION | EXECUTE AS clause_name}]
{FOR | AFTER} {event_group | event_type | LOGON}
```

AS {batch | EXTERNAL NAME method_name}

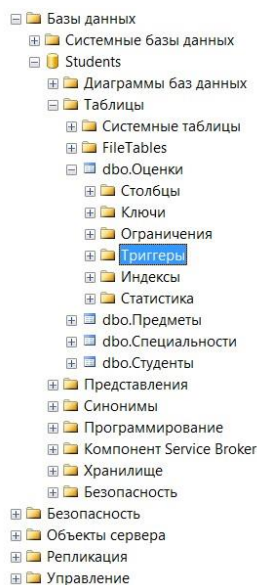
Триггеры DDL создаются таким же способом, как и триггеры DML. А для изменения и удаления этих триггеров используются те же инструкции **ALTER TRIGGER** и **DROP TRIGGER**, что и для триггеров DML.

Первым делом при определении триггера DDL нужно указать его область действия. Предложение **DATABASE** указывает в качестве области действия триггера DDL текущую базу данных, а предложение **ALL SERVER** — текущий сервер.

После указания области действия триггера DDL нужно в ответ на выполнение одной или нескольких инструкций DDL указать способ запуска триггера. В параметре **event_type** указывается инструкция DDL, выполнение которой запускает триггер, а в альтернативном параметре **event_group** указывается группа событий языка Transact-SQL. Триггер DDL запускается после выполнения любого события языка Transact-SQL, указанного в параметре **event_group**.

Кроме сходства триггеров DML и DDL, между ними также есть несколько различий. Основным различием между этими двумя видами триггеров является то, что для триггера DDL можно задать в качестве его области действия всю базу данных или даже весь сервер, а не всего лишь отдельный объект. Кроме этого, триггеры DDL не поддерживают триггеров **INSTEAD OF**. Кроме того, для триггеров DDL не требуются таблицы **inserted** и **deleted**, поскольку эти триггеры не изменяют содержимого таблиц.

Работа с триггерами в среде SQL Server Management Studio Триггеры создаются отдельно для каждой таблицы и располагаются в обозревателе объектов в папке «Триггеры» соответствующей таблицы, которая представлена на рисунке ниже.



2. ПОРЯДОК ВЫПОЛНЕНИЯ РАБОТЫ

1. Изучить теоретическую часть настоящей инструкционно-технологической карты.

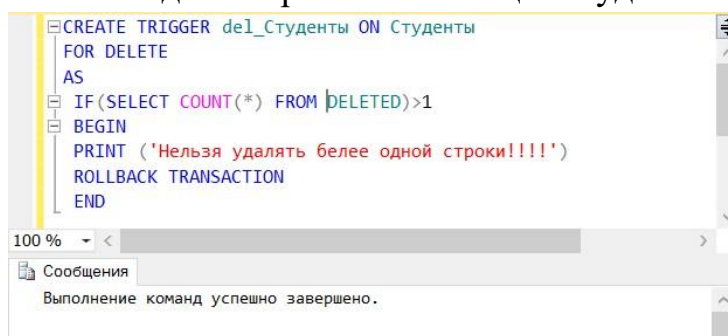
2. Рассмотреть работу триггеров, описанную в разделе «Примеры выполнения работы» настоящей инструкционно-технологической карты.

3. Получить у преподавателя индивидуальное задание и выполнить лабораторную работу в соответствии с вариантом задания согласно описанной в разделе «Пример выполнения работы» методике настоящей инструкционно-технологической карты.

4. Ответить на контрольные вопросы.

3. ПРИМЕР ВЫПОЛНЕНИЯ РАБОТЫ

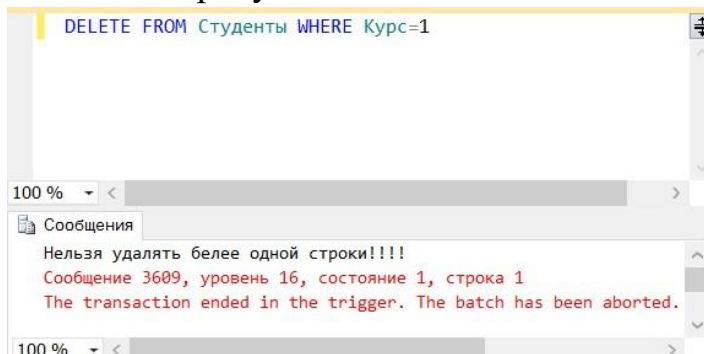
В следующем примере показано создание триггера del_Студенты, который запрещает удалять более одной строки из таблицы Студенты.



```
CREATE TRIGGER del_Студенты ON Студенты
FOR DELETE
AS
IF (SELECT COUNT(*) FROM DELETED) > 1
BEGIN
PRINT ('Нельзя удалять более одной строки!!!!')
ROLLBACK TRANSACTION
END
```

Сообщения
Выполнение команд успешно завершено.

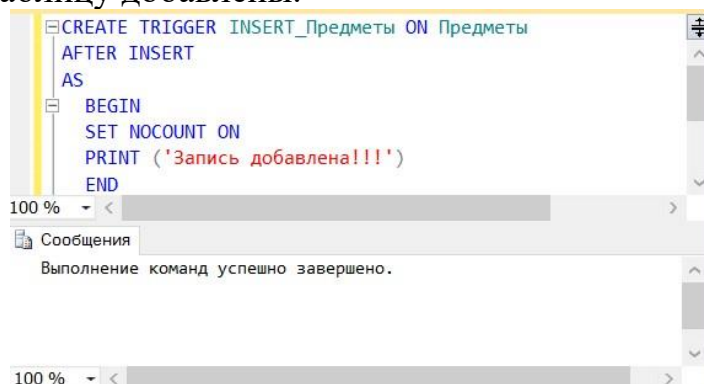
Для того, чтобы проверить, как работает новый триггер необходимо создать новый пустой запрос и в нём выполним удаление данных из таблицы «Студенты», как представлено на рисунке ниже.



```
DELETE FROM Студенты WHERE Курс=1
```

Сообщения
Нельзя удалять более одной строки!!!!
Сообщение 3609, уровень 16, состояние 1, строка 1
The transaction ended in the trigger. The batch has been aborted.

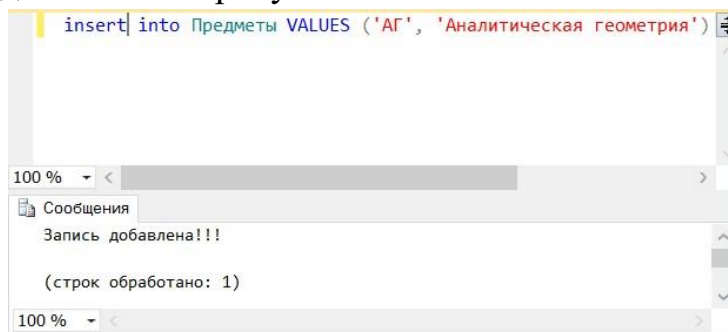
В следующем примере показано создание триггера INSERT_Предметы, который срабатывает после добавления записей в таблицу Предметы и выводит запись, что записи в таблицу добавлены.



```
CREATE TRIGGER INSERT_Предметы ON Предметы
AFTER INSERT
AS
BEGIN
SET NOCOUNT ON
PRINT ('Запись добавлена!!!!')
END
```

Сообщения
Выполнение команд успешно завершено.

Для того, чтобы проверить, как работает новый триггер необходимо создать новый пустой запрос и в нём выполним добавление данных в таблицу «Предметы», как представлено на рисунке ниже.



4. КОНТРОЛЬНЫЕ ВОПРОСЫ

1. Дайте определение понятию триггер.
 2. Назовите два типа триггеров по типу действия, которое запускает триггер.
- Поясните.
3. Назовите инструкции создания, модификации и удаления триггера.
 4. Поясните суть виртуальных таблиц `deleted` и `inserted`.
 5. Назовите основные отличия триггеров `AFTER` и триггеров `INSTEAD OF`.
 6. Назовите основные отличия триггеров `DML` и `DDL`. Дайте определение понятия «Индекс».

5. ЛИТЕРАТУРА

1. Петкович, Д. Microsoft SQL Server 2012. Руководство для начинающих: пер. с английского / Д. Петкович. – СПб.: БХВ-Петербург, 2013. – 816 с.: ил.
2. Сеть разработчиков Microsoft [Электронный ресурс]. – Режим доступа: <https://msdn.microsoft.com/ru-ru/library>

Преподаватель

В.Ю.Купцова

Рассмотрено на заседании цикловой
комиссии программного обеспечения
информационных технологий №10
Протокол № __ от «__» _____ 2022
Председатель ЦК В.Ю.Михалевич