Создание триггеров

Задание:
1. Изучить принцип работы триггера на примере.
2. Создать триггеры для любой своей таблицы по своей предметной области.
3. Сделать скрины и поместить их в отчет.
4. Показать триггеры и отчет преподавателю.

Создадим триггеры для таблицы **"Студенты"**. Триггеры создаются отдельно для каждой таблицы и располагаются в обозревателе объектов в папке **"Triggers"**. В нашем случае, папка **"Triggers"** входит в состав таблицы **"Студенты"** (рис. 1).

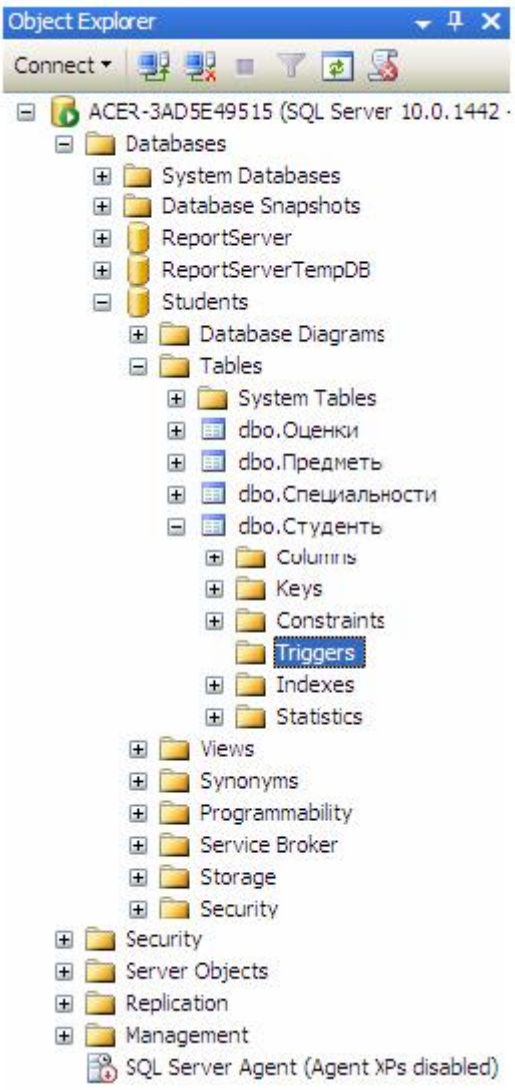

Рисунок 1

Для начала создадим триггер, выводящий сообщение **"Запись добавлена"** при добавлении записи в таблицу **"Студенты"**.

Создадим новый триггер, щелкнув **ПКМ** по папке "**Triggers**" в таблице **"Студенты"** и в появившемся меню выбрав пункт "**New Trigger**". Появится следующее окно с новым триггером (рис. 2):

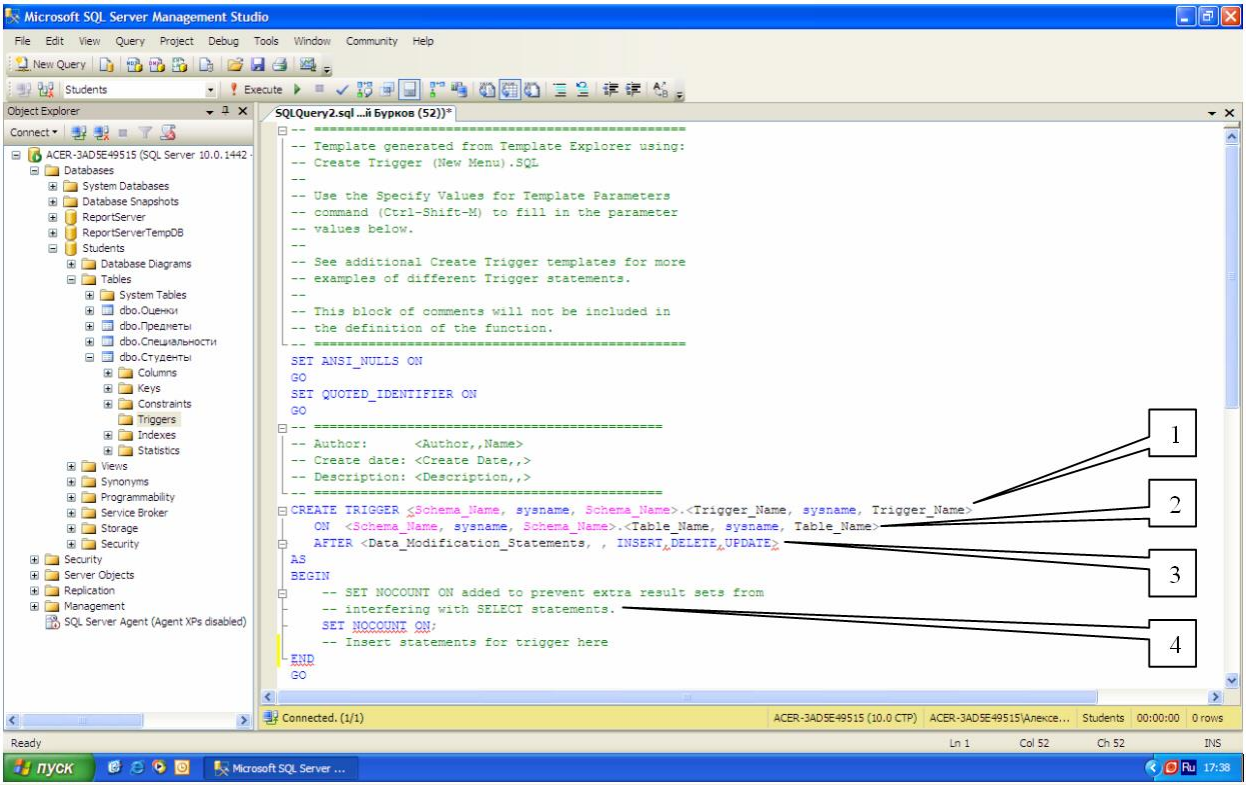

Рисунок 2

Рассмотрим структуру триггеров:

1. Область определения имени функции (Trigger_Name);

2. Область, показывающая для какой таблицы создается триггер (Table_Name);

3. Область, показывающая когда выполнять триггер (INSERT - при создании записи в таблице, DELETE - при удалении и UPDATE - при изменении) и как его выполнять (AFTER - после выполнения операции, INSTEAD OF - вместо выполнения операции);

4. Тело триггера, содержит команды языка программирования запросов T-SQL.

В окне нового триггера наберите код как показано на рис.3.

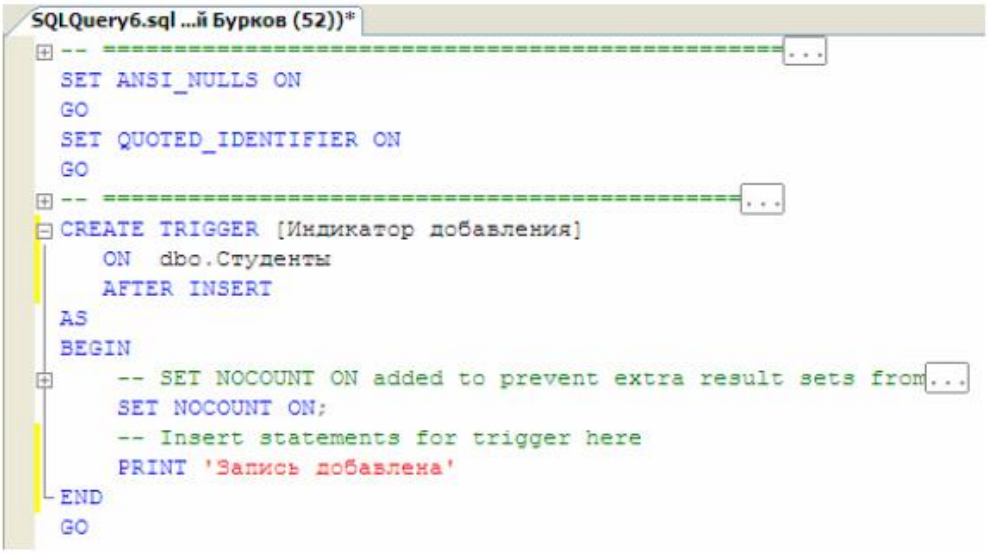

Рисунок 3

Из рис. 3 видно, что создаваемый триггер **"Индикатор добавления"** выполняется после добавления записи (**AFTER INSERT**) в таблицу "Студенты" (**ON dbo.Студенты**). После добавления записи триггер выведет на экран сообщение "Запись добавлена" (**PRINT 'Запись добавлена'**).

Выполните набранный код, нажав кнопку Execute на панели инструментов. В нижней части окна с кодом появится сообщение "Command(s) completed successfully.".

Проверим, как работает новый триггер. Создайте новый пустой запрос и в нем наберите следующую команду для добавления новой записи в таблицу **"Студенты"** (рис. 4):

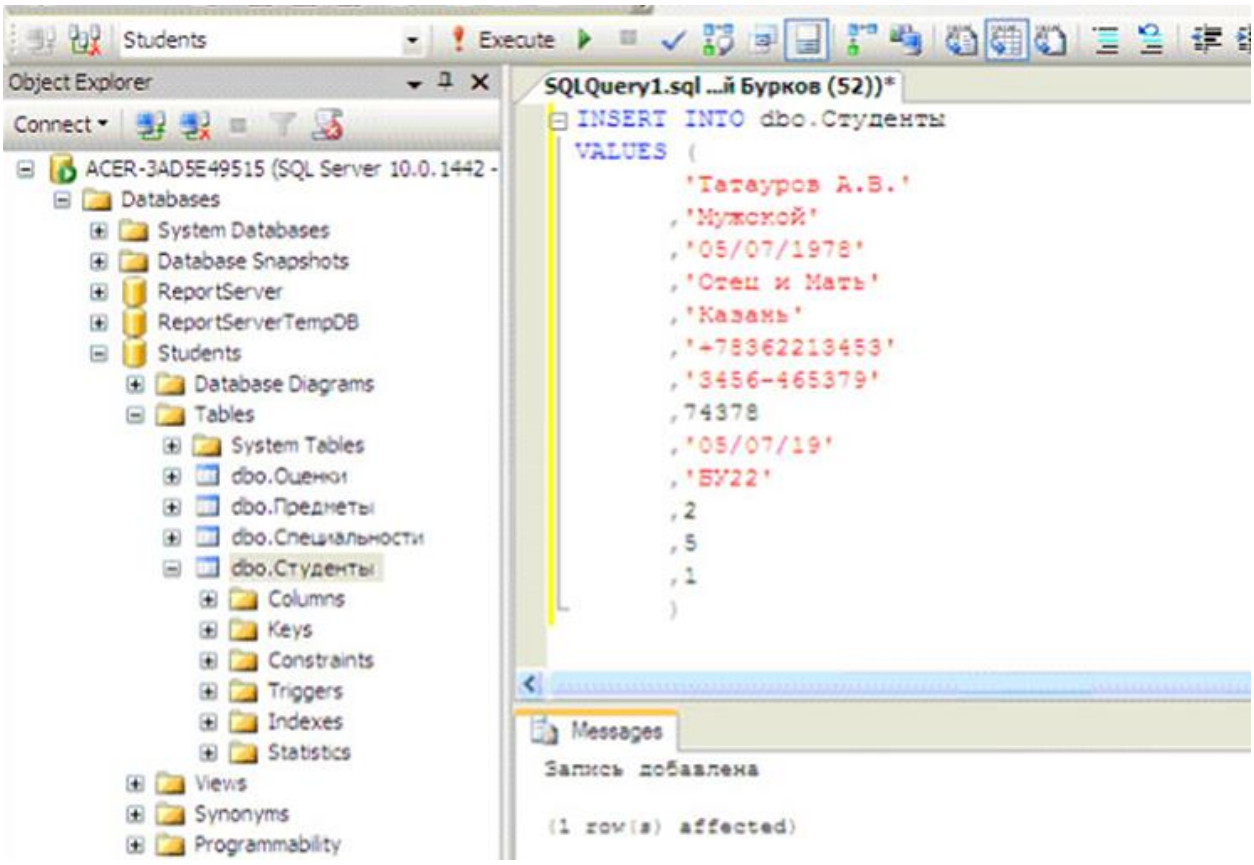

Рисунок 4

Выполните набранную команду, нажав кнопку 

на панели инструментов. В таблицу будет добавлена новая запись, и триггер выведет сообщение "Запись добавлена" (рис. 4).

Теперь создадим триггер отображающий сообщение **"Запись изменена"**. Создайте новый триггер, как в предыдущем случае. В окне нового триггера наберите следующий код (рис. 5):

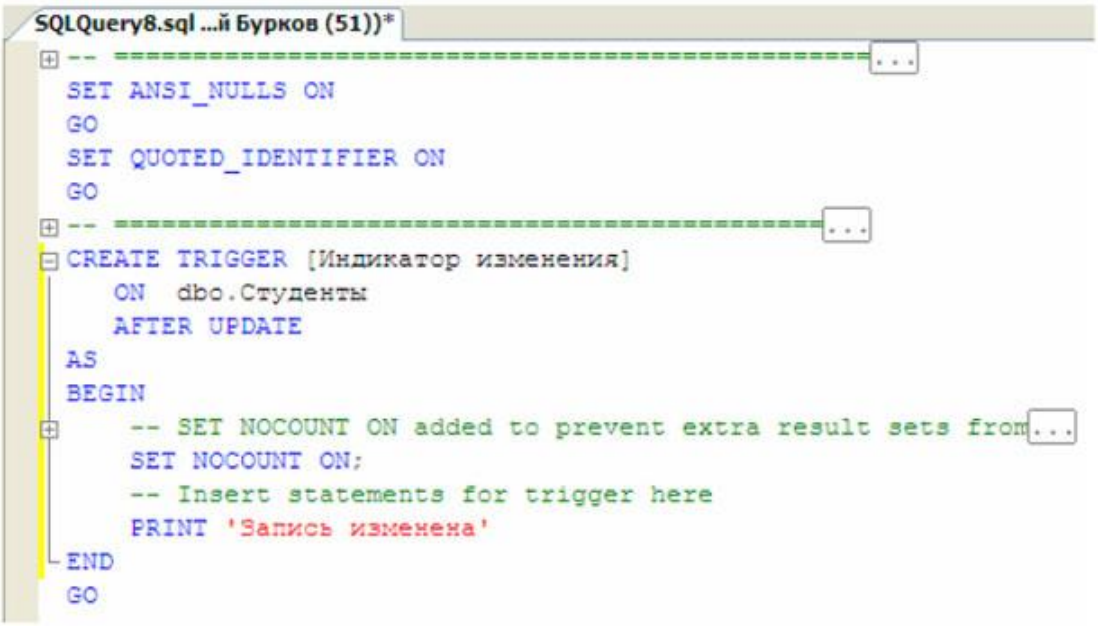

Рисунок 5

Из рис.5 видно, что новый триггер **"Индикатор изменения"** выполняется после изменения записи (**AFTER UPDATE**) в таблице "Студенты" (**ON dbo.Студенты**). После изменения записи триггер выведет на экран сообщение "Запись изменена" (**PRINT 'Запись изменена'**). Выполните набранный код. В нижней части окна с кодом появится сообщение **"Command(s) completed successfully.**".

Проверим работоспособность созданного триггера. Создайте новый запрос и в нем наберите команду, представленную на рис. 6.

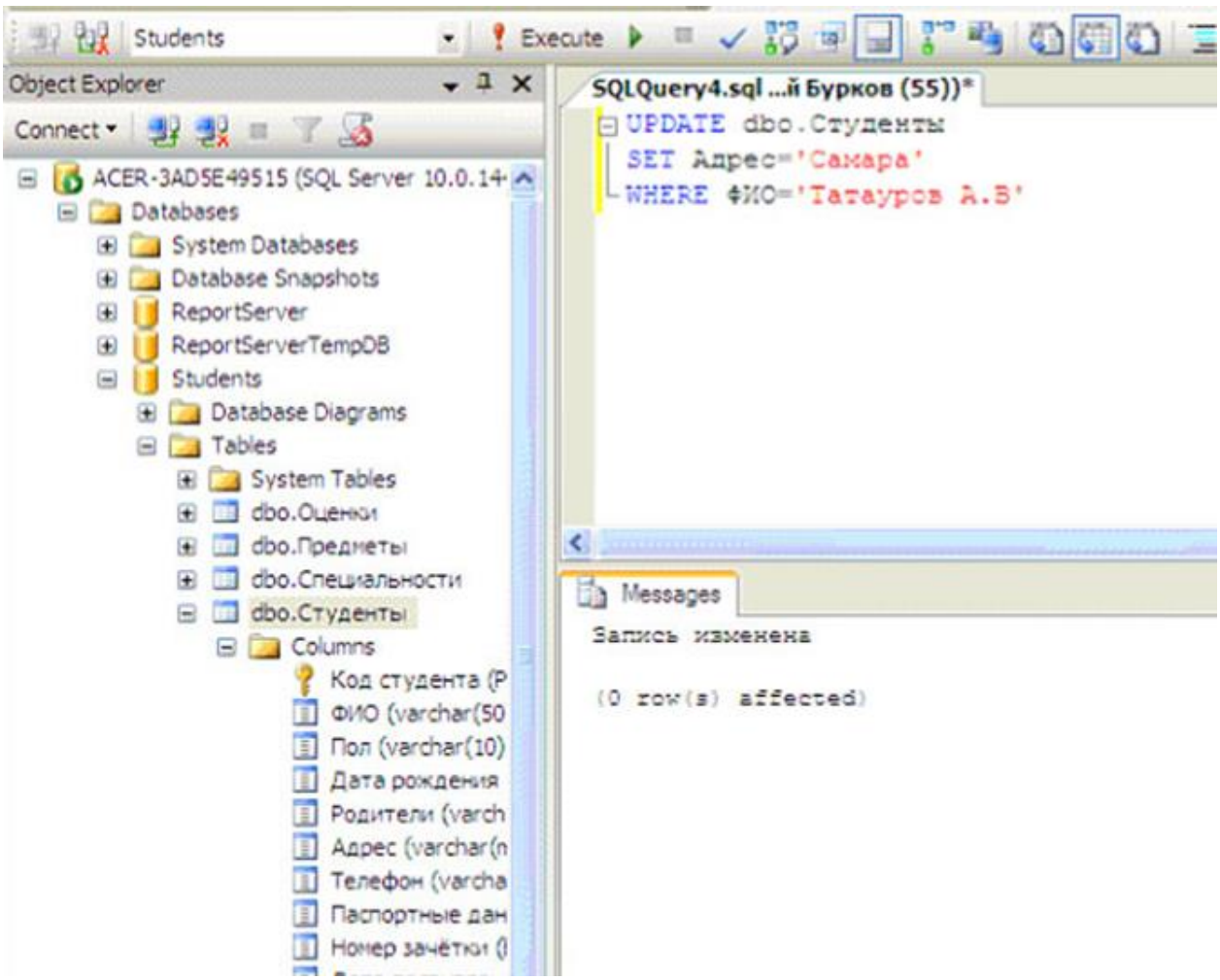

Рисунок 6

Выполните набранную команду, нажав кнопку 

на панели инструментов. В таблицу будет добавлена новая запись, и триггер выведет сообщение **"Запись изменена"** (рис. 6).

Для полноты картины создадим триггер, выводящий сообщение при удалении записи из таблицы **"Студенты"**. Создайте новый триггер и в нем наберите код, показанный на рис. 7.

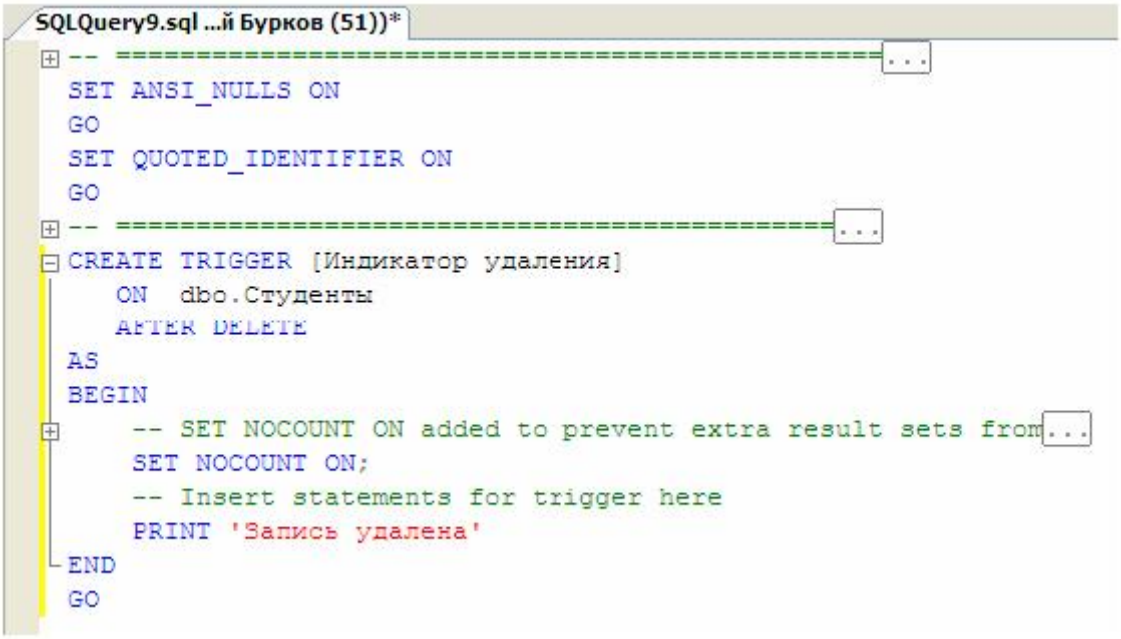

Рисунок 7

Создаваемый триггер **"Индикатор удаления"** выполняется после удаления записи (**AFTER DELETE**) из таблицы студенты (**ON dbo.Студенты**). После удаления записи триггер выводит сообщение "Запись удалена" (**PRINT 'Запись удалена'**).

Выполните код, представленный рис. 7. В нижней части окна с кодом появиться сообщение **"Command(s) completed successfully.**".

Проверим работу триггера **"Индикатор удаления"** удалив созданную ранее запись из таблицы **"Студенты"**. Для этого создайте новый запрос и в нем наберите следующую команду (рис. 8):

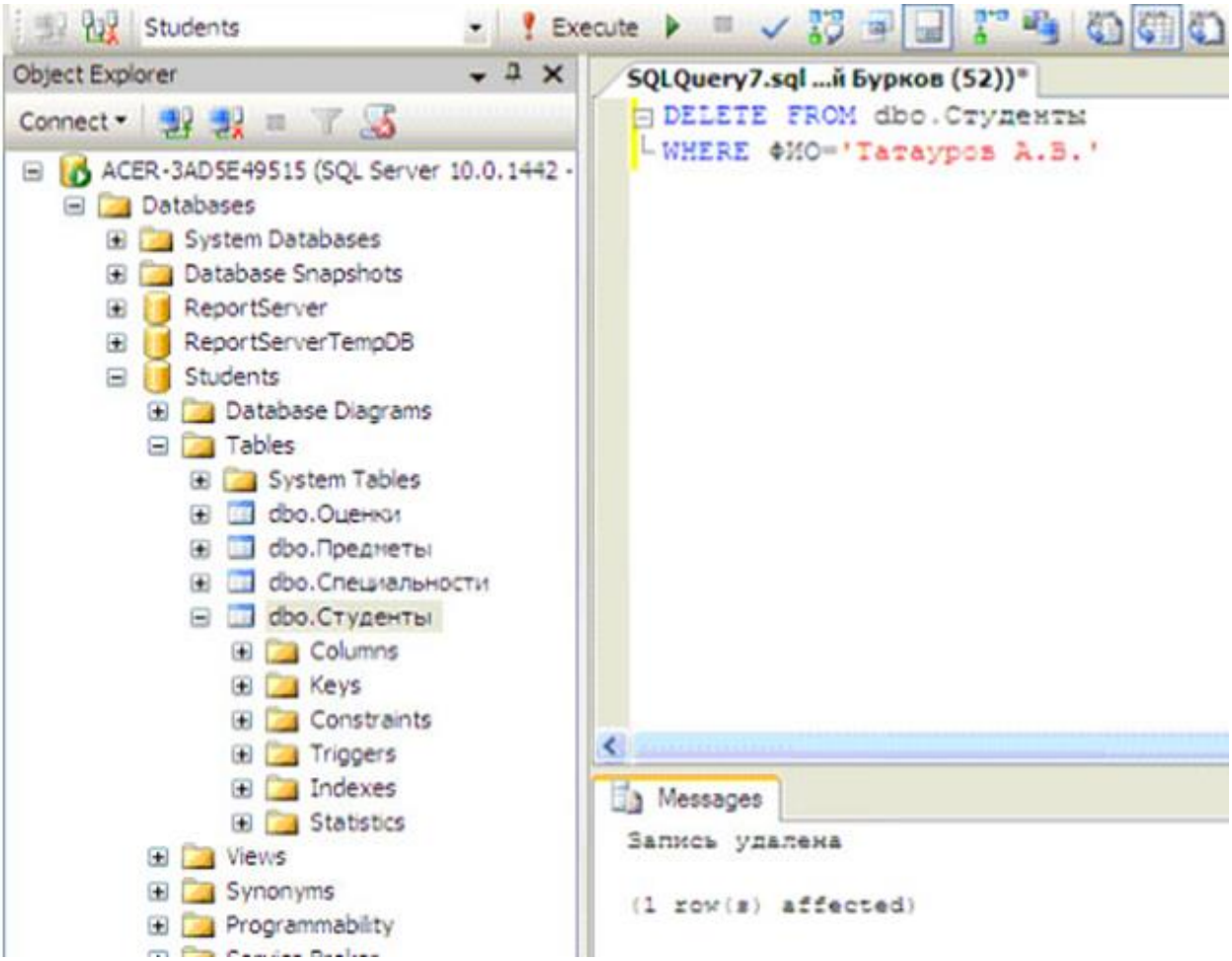

Рисунок 8

Выполните вышеприведенную команду. После удаления записи триггер **"Индикатор удаления"** отобразит сообщение **"Запись удалена"** (рис. 8).

В заключение рассмотрим пример применения триггеров для обеспечения целостности данных. Создадим триггер **"Удаление студента"**, который при удалении записи из таблицы студенты сначала удаляет все связанные с ней записи из таблицы **"Оценки"**, а затем удаляет саму запись из таблицы **"Студенты"**, тем самым обеспечивается целостность данных.

Создайте новый триггер и в нем наберите следующий код (рис. 9):

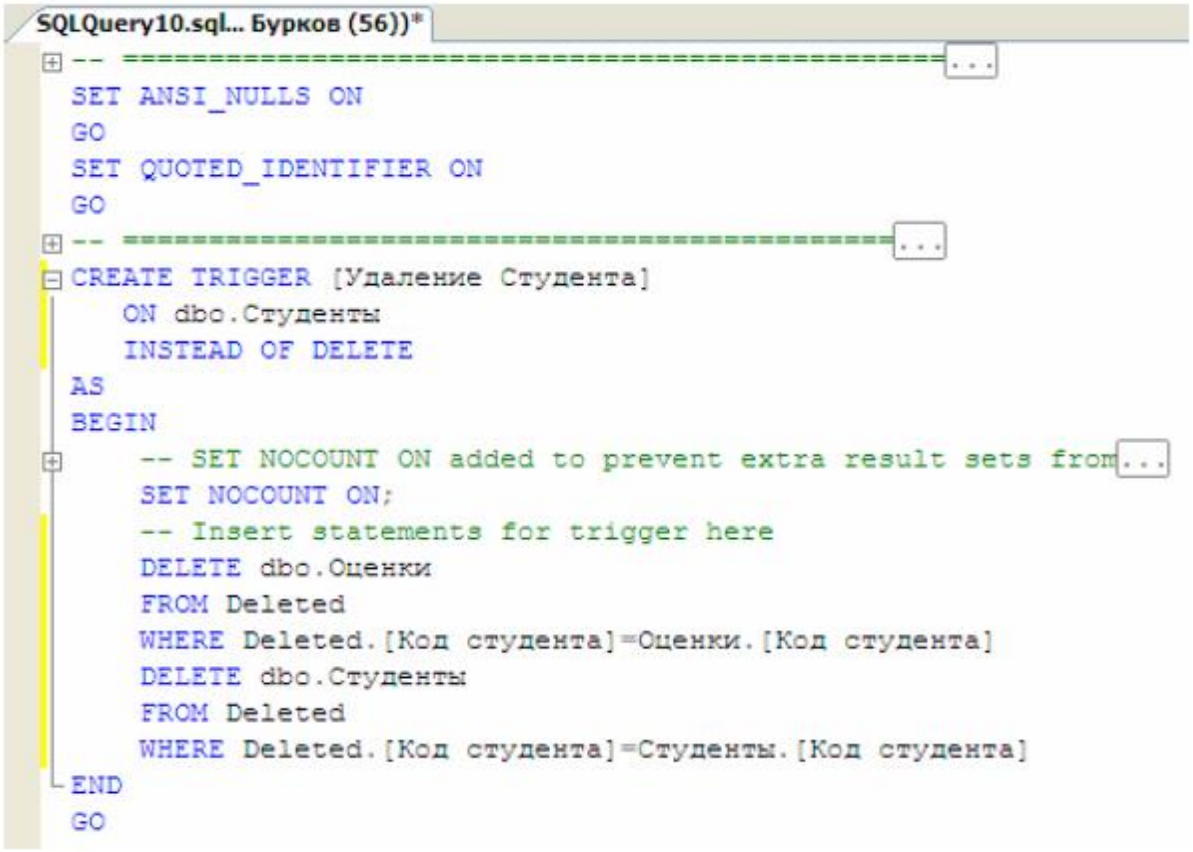

Рисунок 9

Создаваемый триггер **"Удаление студента"** выполняется вместо удаления записи (**INSTEAD OF DELETE**) из таблицы "Студенты" (**ON dbo.Студенты**).

Замечание: При срабатывании триггера вместо удаления записи создается временная константа **Deleted**, содержащая имя таблицы из которой должно было быть произведено удаление.

После срабатывания триггера из таблицы **"Оценки"** удаляется запись, у которой значение поля **"Код студента"** равно значению такого же поля у удаляемой записи из таблицы **"Студенты"**. Эту операцию выполняют следующие команды:

```
DELETE FROM dbo.Оценки
WHERE Deleted.[Код студента] = Оценки.[Код студента]
```

Затем удаляется запись из таблицы **"Студенты"**, которую удаляли до срабатывания триггера. Удаление выполняется следующими командами:

```
DELETE dbo.Студенты
FROM Deleted
WHERE Deleted.[Код студента] = Студенты.[Код студента]
```

Выполните код, представленный на рис. 3.17. В нижней части окна с кодом появиться сообщение **"Command(s) completed successfully."**.

Проверим, как работает триггер **"Удаление студента"**. Для этого создайте новый запрос и в нем наберите следующий код (рис. 10):

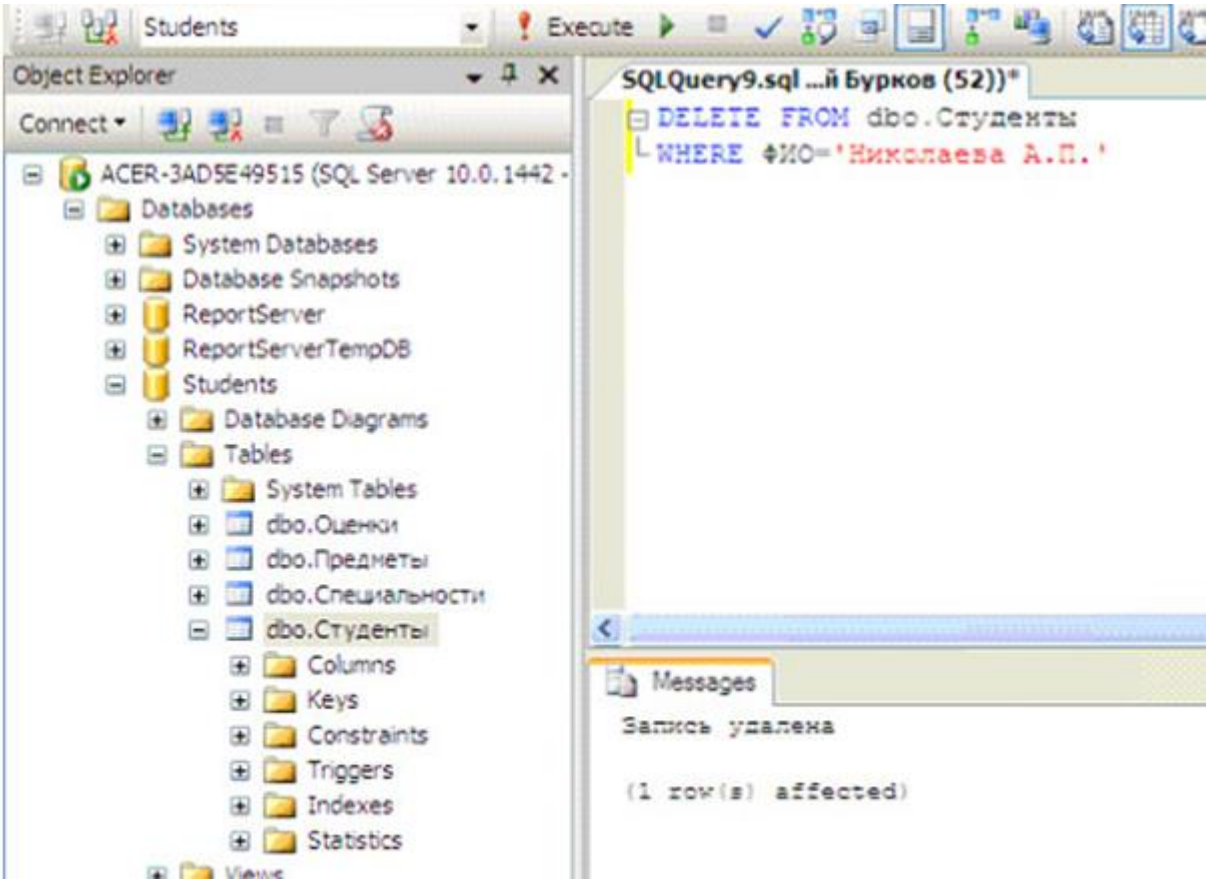

Рисунок 10

При срабатывании триггера сначала из таблицы **"Оценки"** удалятся все связанные с удаляемой записью записи, а затем удаляется сама удаляемая запись из таблицы **"Студенты"**, при этом сохраняется целостность данных.

Замечание: Хотелось бы заметить, что без использования триггера **"Удаление студента"** нам бы не удалось удалить запись из таблицы **"Студенты"**. Команда удаления была бы заблокирована диаграммой **"Диаграмма БД Студенты"** во избежание нарушения целостности данных.

На этом мы завершаем работу с диаграммами и триггерами. После выполнения всех вышеописанных действий обозреватель объектов будет иметь следующий вид (рис. 11):

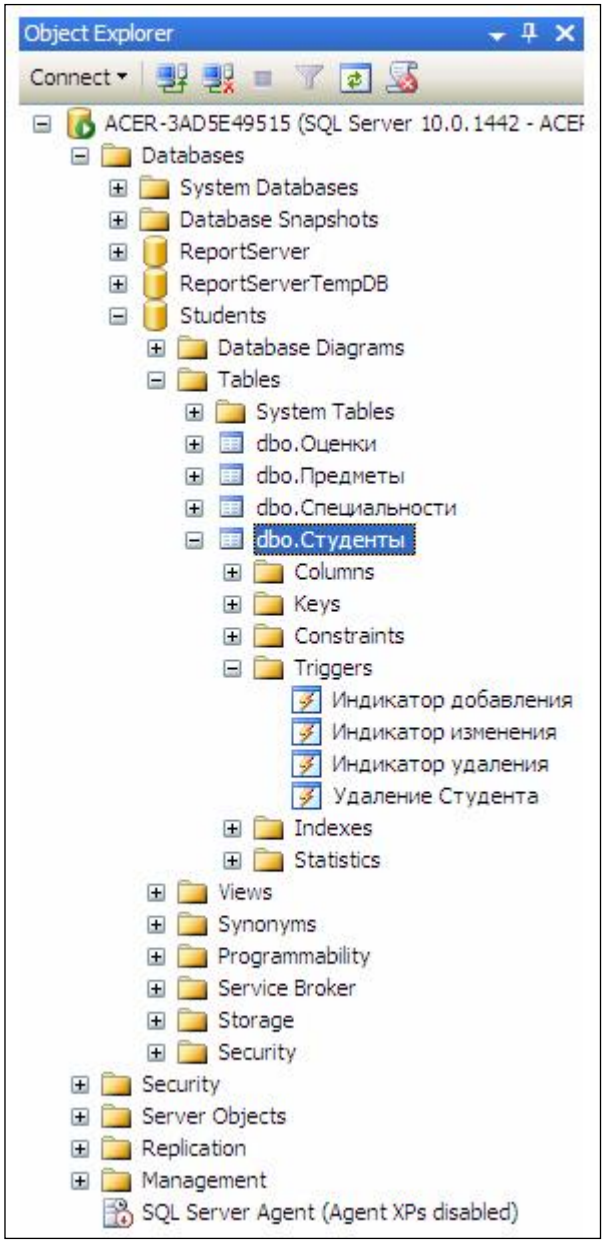
Object Explorer
Connect
ACER-3AD5E49515 (SQL Server 10.0.1442 - ACEI
Databases
System Databases
Database Snapshots
ReportServer
ReportServerTempDB
Students
Database Diagrams
Tables
System Tables
dbo.Оценки
dbo.Предметы
dbo.Специальности
dbo.Студенты
Columns
Keys
Constraints
Triggers
Индикатор добавления
Индикатор изменения
Индикатор удаления
Удаление Студента
Indexes
Statistics
Views
Synonyms
Programmability
Service Broker
Storage
Security
Security
Server Objects
Replication
Management
SQL Server Agent (Agent XPs disabled)

Рисунок 11