**Отчет по дисциплине “Базы данных”**

Лабораторная №14

Цель: Применение триггеров.

Ободов Павел

2 курс 2 группа

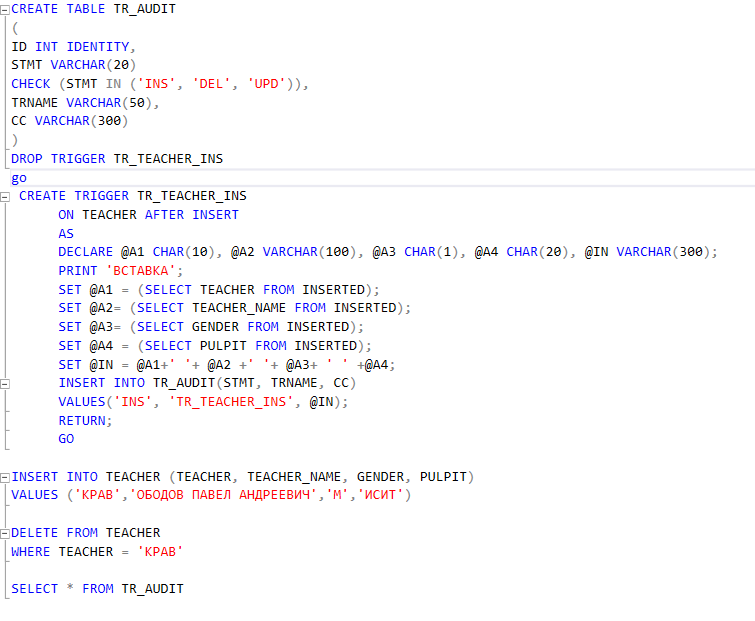


Рисунок 1

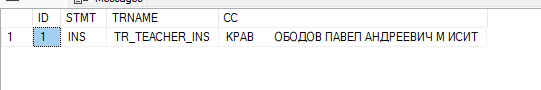


Рисунок 2

Рисунок 1 - .С помощью сценария, представленного на рисунке, создать таблицу **TR\_AUDIT**.

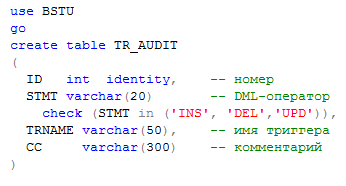
****

Таблица предназначена для добавления в нее строк триггерами. В столбец **STMT** триггер должен поместить событие, на которое он среагировал, а в столбец **TRNAME −** собственное имя. Разработать AFTER-триггер с именем **TR\_TEACHER\_INS** для таблицы **TEACHER**, реагирующий на событие **INSERT**. Триггер должен записывать строки вводимых данных в таблицу **TR\_AUDIT**. В столбец **СС** помещаются значения столбцов вводимой строки.

Рисунок 2 – выполнение триггера

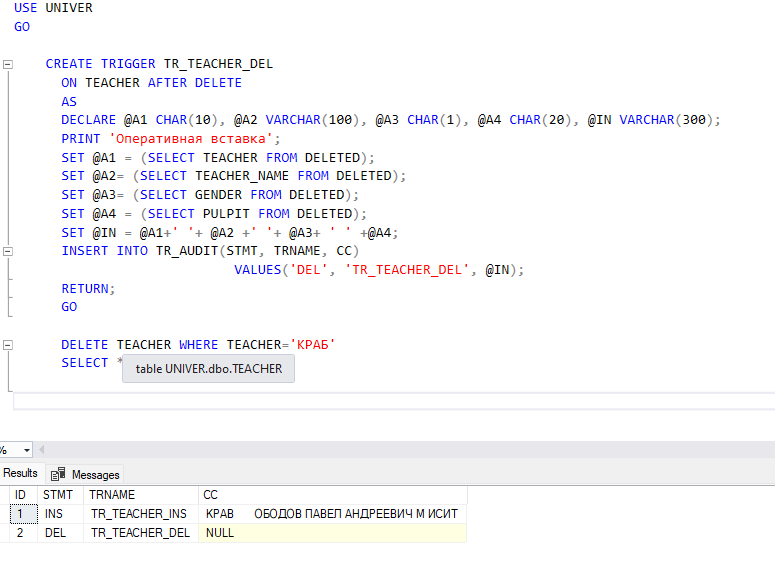


Рисунок 3

Рисунок 3 - Создать AFTER-триггер с именем **TR\_TEACHER\_DEL** для таблицы **TEA-CHER**, реагирующий на событие **DELETE**. Триггер должен записывать строку данных в таблицу **TR\_AUDIT** для каждой удаляемой строки. В столбец **СС** помещаются значения столбца **TEACHER** удаляемой строки.

А также вывод триггера

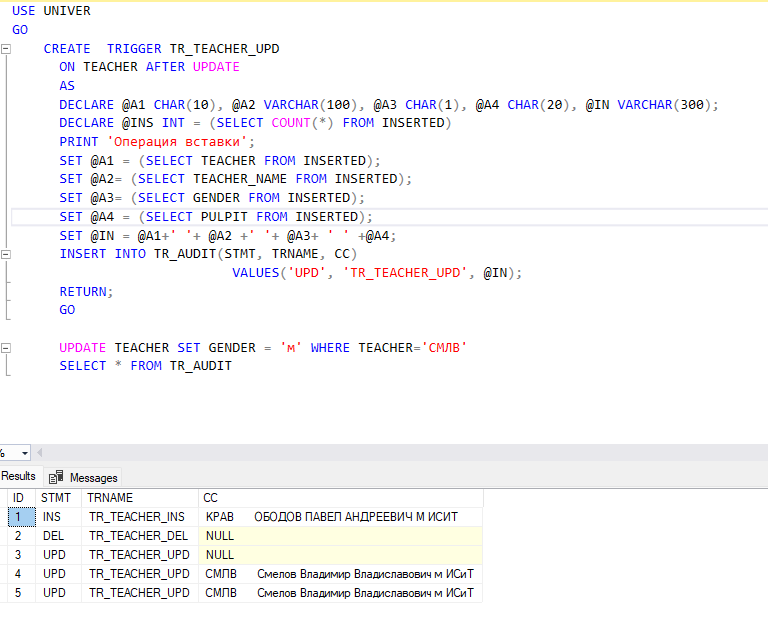


Рисунок 4

Рисунок 4 - Создать AFTER-триггер с именем **TR\_TEACHER\_UPD** для таблицы **TEA-CHER**, реагирующий на событие **UPDATE**. Триггер должен записывать строку данных в таблицу **TR\_AUDIT** для каждой изменяемой строки. В столбец **СС** помещаются значения столбцов изменяемой строки до и после изменения.

А также выполнение триггера.

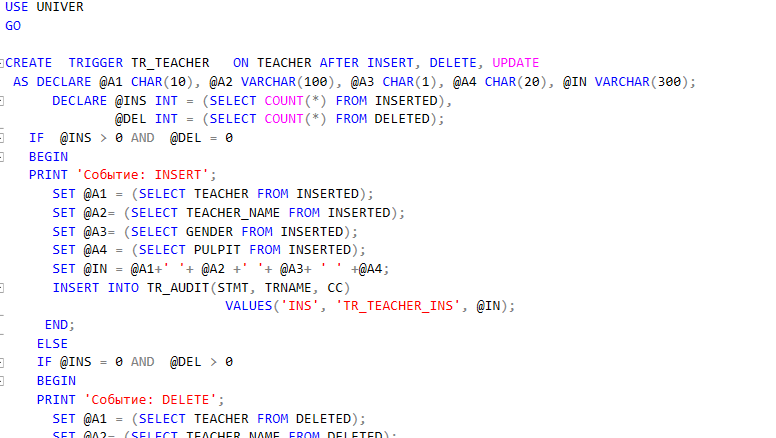


Рисунок 5

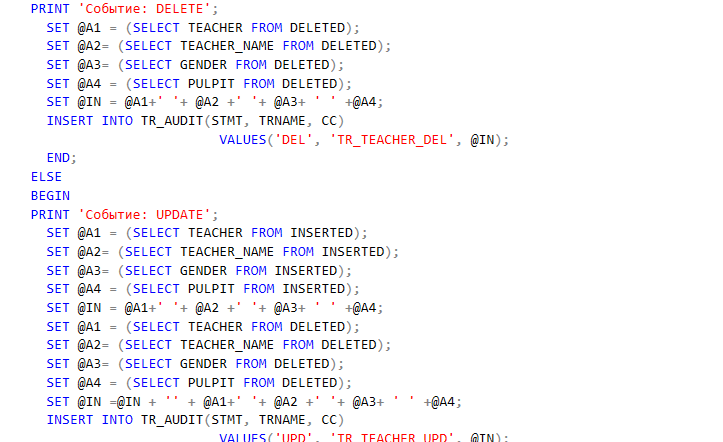


Рисунок 6

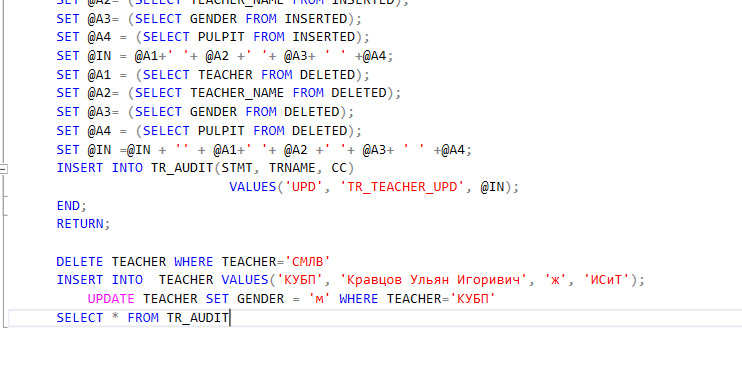


Рисунок 7

Рисунок 5,6,7 - Создать AFTER-триггер с именем **TR\_TEACHER** для таблицы **TEACHER**, реагирующий на события **INSERT**, **DELETE**, **UPDATE**. Триггер должен записывать строку данных в таблицу **TR\_AUDIT** для каждой изменяемой строки. В коде триггера определить событие, активизировавшее триггер и поместить в столбец **СС** соответствующую событию информацию. Разработать сценарий, демонстрирующий работоспособность триггера.

Рисунок 8

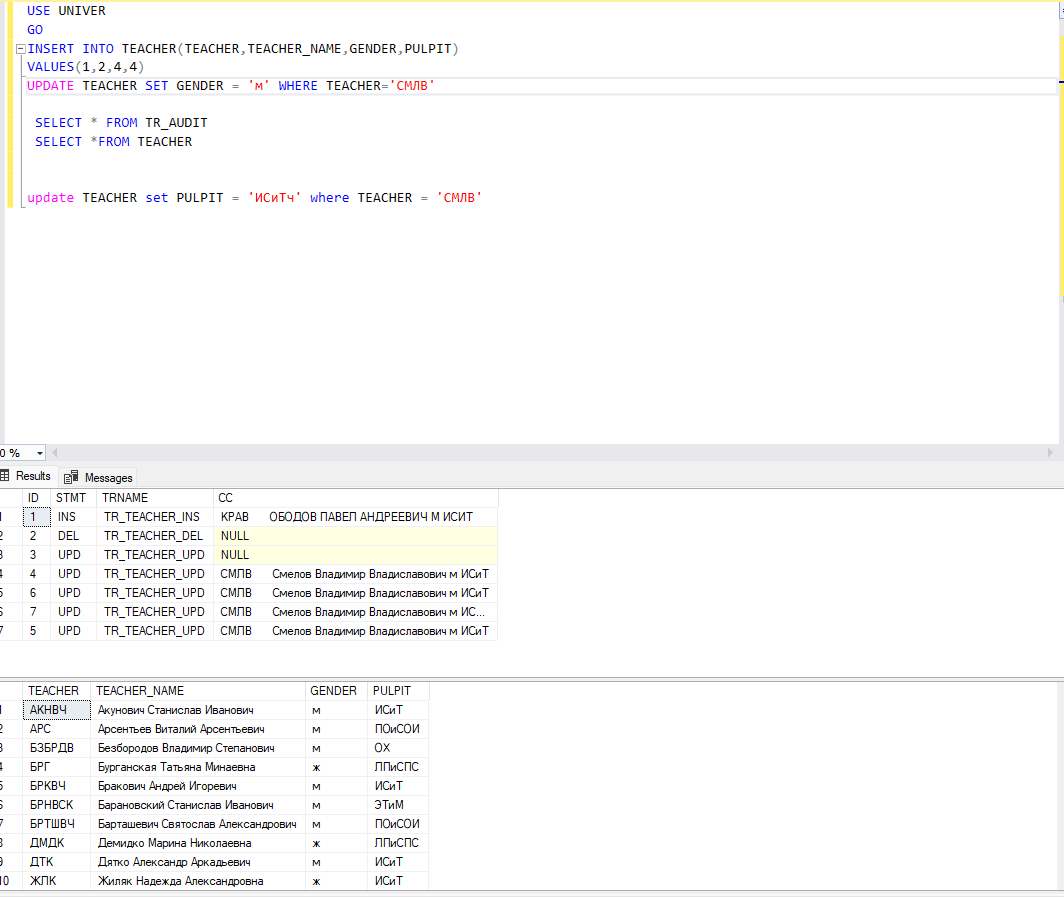


Рисунок 8 - Разработать сценарий, который демонстрирует на примере базы данных X\_UNIVER, что проверка ограничения целостности выполняется до срабатывания AFTER-триггера.

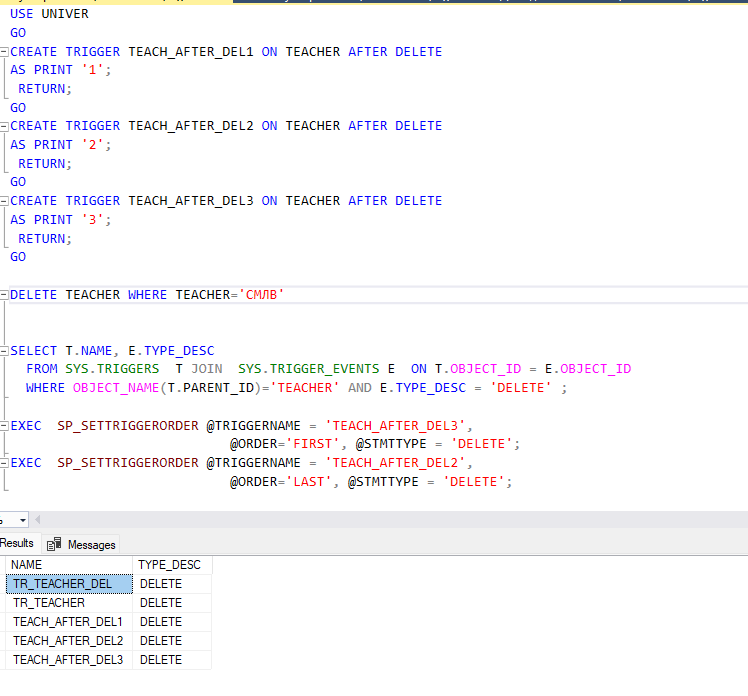


Рисунок 9

Рисунок 9 - Создать для таблицы **TEACHER** три AFTER-триггера с именами: **TR\_TEACHER\_ DEL1**, **TR\_TEACHER\_DEL2** и **TR\_TEA-CHER\_ DEL3**. Триггеры должны реагировать на событие DELETE и формировать соответствующие строки в таблицу **TR\_AUDIT**. Получить список триггеров таблицы **TEACHER**. Упорядочить выполнение триггеров для таблицы **TEACHER**, реагирующих на событие **DELETE** следующим образом: первым должен выполняться триггер с именем **TR\_TEA-CHER\_DEL3**, последним – триггер **TR\_TEACHER\_DEL2**. Примечание: использовать системные представления **SYS.TRIGGERS** и **SYS.TRIG-GERS\_ EVENTS**, а также системную процедуру **SP\_SETTRIGGERORDERS**.

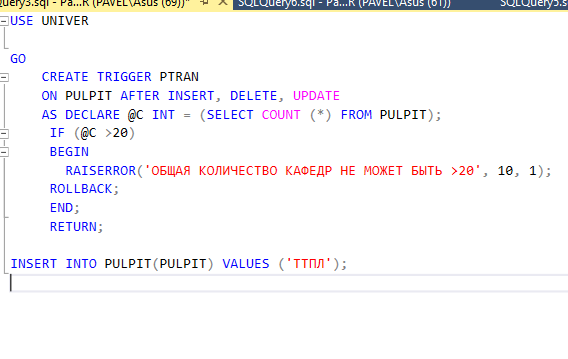


Рисунок 10

Рисунок 10 - Разработать сценарий, демонстрирующий на примере базы данных X\_UNIVER утверждение: AFTER-триггер является частью транзакции, в рамках которого выполняется оператор, активизировавший триггер.

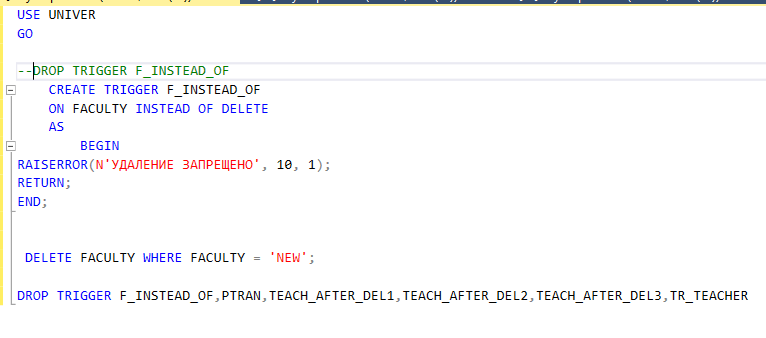


Рисунок 11

Рисунок 11 - Для таблицы **FACULTY** создать **INSTEAD OF**-триггер, запрещающий удаление строк в таблице.

Разработать сценарий, который демонстрирует на примере базы данных X\_UNIVER, что проверка ограничения целостности выполнена, если есть INSTEAD OF-триггер.

С помощью оператора DROP удалить все DML-триггеры, созданные в этой лабораторной работе.

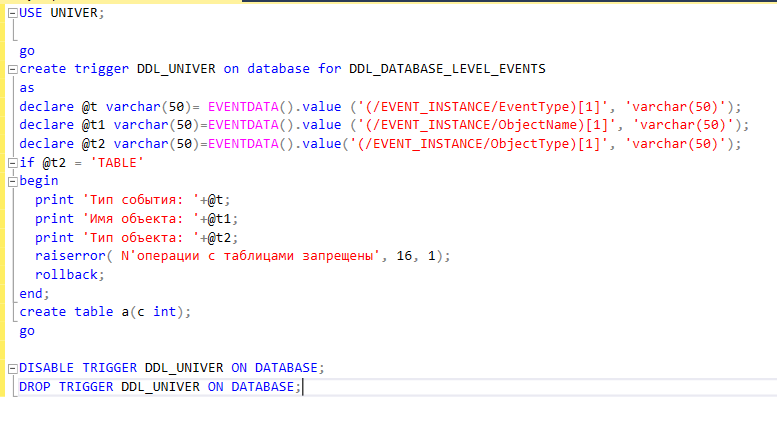


Рисунок 12

Рисунок 12 - Создать DDL-триггер, реагирующий на все DDL-события в БД **UNIVER**. Триггер должен запрещать создавать новые таблицы и удалять существующие. Свое выполнение триггер должен сопровождать сообщением, которое содержит: тип события, имя и тип объекта, а также пояснительный текст, в случае запрещения выполнения оператора. Разработать сценарий, демонстрирующий работу триггера.

**Вывод:** Изучил применение триггеров. Какие они бывают, что делают.