**Лабораторная работа № 14. Применение триггеров**

**Вариант 12**

Цель: изучить триггеры и их назначение; научиться взаимодействовать с триггерами.

Триггер – это особый вид хранимой процедуры, предназначенной для обработки событий в БД. Поддерживается два типа триггеров: DDL-триггеры и DML-триггеры. Для каждого типа определено свое семейство событий, обработку которых триггер этого типа может выполнять.

DML-триггеры бывают двух типов: AFTER-триггеры и INSTEAD OF-триггеры. Триггеры типа AFTER исполняются после выполнения оператора, вызвавшего соответствующее событие. При этом создаются автоматически две псевдотаблицы INSERTED и DELETED.

Триггер типа INSTEAD OF выполняется вместо оператора, вызвавшего соответствующее событие. Выполнение INSTEAD OF триггера предшествует проверке установленных для таблицы ограничений целостности.

Событие INSERT при выполнении AFTER-триггера приводит к тому, что в псевдотаблицу INSERTED помещаются строки, добавленные оператором INSERT, вызвавшим это событие. Псевдотаблица DELETED остается пустой.

При возникновении события DELETE в таблицу DELETED копируются удаленные строки, а таблица INSERTED остается пустой.

При изменении строк таблицы с помощью оператора UPDATE заполняются обе псевдотаблицы, при этом таблица INSERTED содержит обновленные версии строк, а таблица DELETED − версию строк до их изменения.

Первое задание­ создания таблицы для триггеров и разработка триггера вставки – рисунок 1.1.

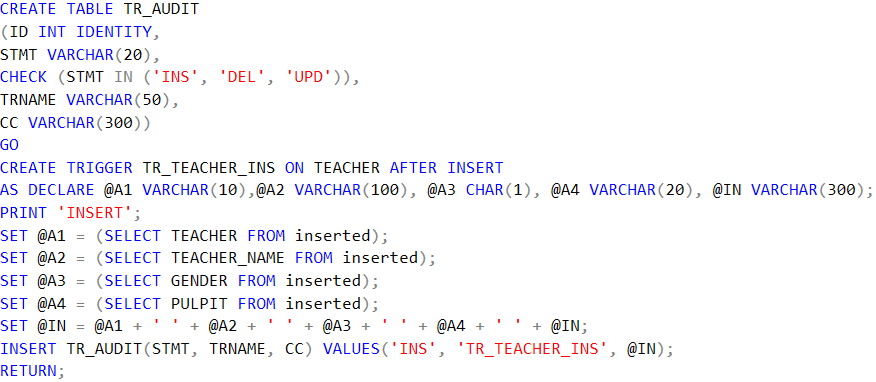


Рисунок 1.1 ­– Задание 1

Второе задание – разработка триггера удаления – рисунок 1.2.

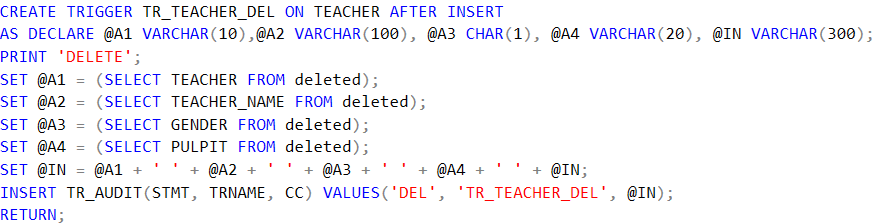


Рисунок 1.2 ­– Триггер удаления

Третье задание: разработка триггера обновления – рисунок 1.3.

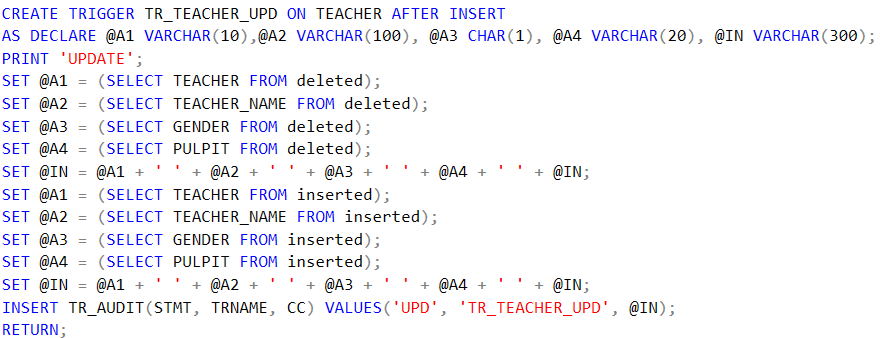


Рисунок 1.3 ­– триггер обновления

Четвертое задание ­ разработка триггера комбинирующего задания 1-3. Сценарий представлен на рисунке 1.4.

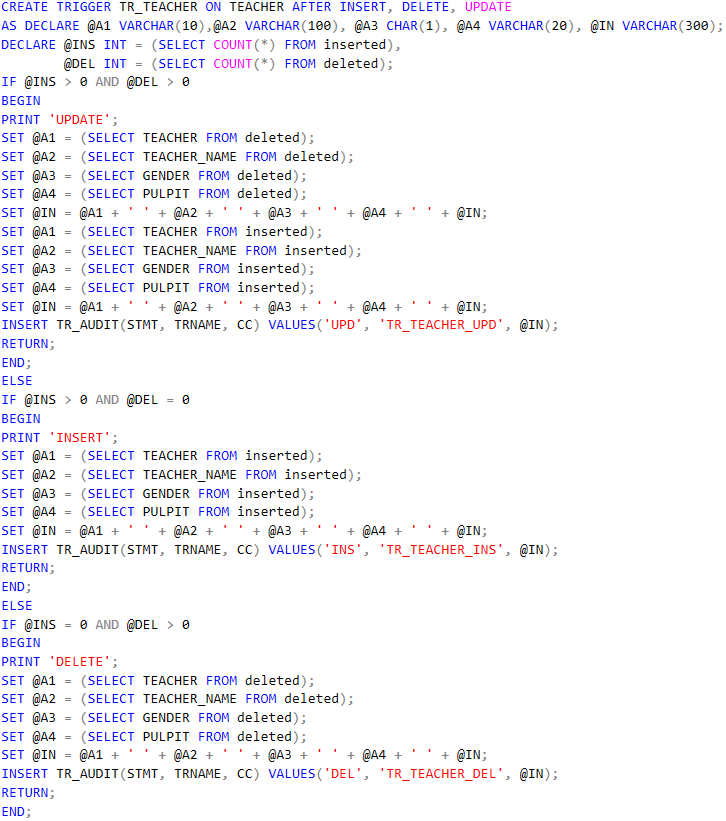


Рисунок 1.4 ­– Итоговый сценарий

Важной особенностью AFTER-триггера является то, что он вызывается после выполнения активизирующего его оператора. Поэтому, если оператор нарушает ограничение целостности, то возникшая ошибка не допускает выполнения этого оператора и соответствующих триггеров.

Разработать сценарий, который демонстрирует на примере базы данных X\_UNIVER, что проверка ограничения целостности выполняется до срабатывания AFTER-триггера – рисунок 1.5.



Рисунок 1.5 ­– Задание 5

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 – рисунок 1.6.

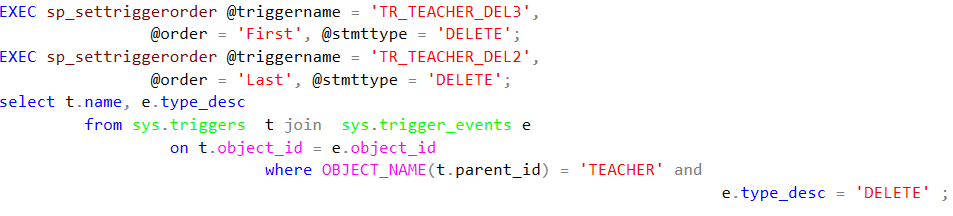


Рисунок 1.6 ­– Задание 6

Разработать сценарий, демонстрирующий на примере базы данных X\_UNIVER утверждение: AFTER-триггер является частью транзакции, в рамках которого выполняется оператор, активизировавший триггер – рисунок 1.7.

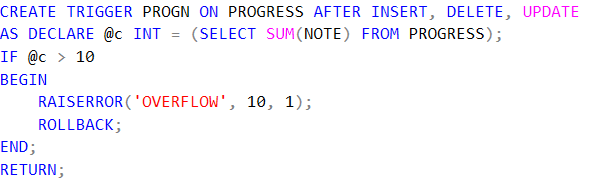


Рисунок 1.7 ­– Задание 7

Для таблицы FACULTY создать INSTEAD OF-триггер, запрещающий удаление строк в таблице – рисунок 1.8.

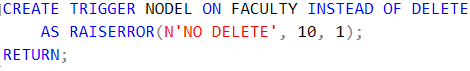


Рисунок 1.8 ­– Задание 8

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 – рисунок 1.9.

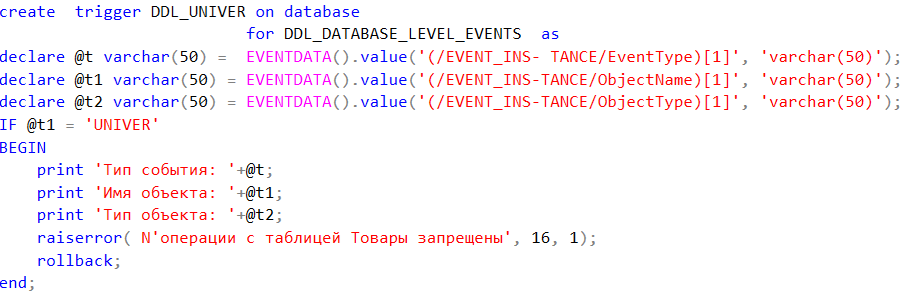


Рисунок 1.9 ­– Задание 9

**Вывод**

В ходе лабораторной работы были изучены триггеры, а также способы их применения; были получены навыки по созданию, использованию и взаимодействию с триггерами.