**Лабораторное занятие №11.**

**Триггеры.**

**Лабораторное (практическое) занятие** рассчитано на 2 часа, относится к теме «Эволюция компьютерных сетей»

**Формируемые компетенции:** ОК.1-9 ПК 2.1 - 2.4

**Цель:** Изучить используемые в реляционных СУБД операторы создания триггеров. Получить навыки работы с триггерами с помощью команд SQL и с помощью программы "SQLiteStudio".

**Методическое и техническое обеспечение:** IBM PC-совместимый компьютер, SQLiteStudio.

## Исходные данные

Студент получает индивидуальный вариант исходных данных с кратким описанием предметной области, который используется при выполнении всех лабораторных работ. При этом каждая очередная лабораторная работа является продолжением выполненной ранее и поэтому они должны обязательно выполняться последовательно. Вариант задания приведен в лабораторной работе №1.

***Теоретические сведения***

### Триггеры (Triggers)

С любым событием, вызывающим изменение содержимого таблицы, можно связать сопутствующее действие (триггер), которое СУБД должна выполнять при каждом возникновении события. Триггер – это группа операторов языка SQL, которые автоматически выполняется при вставке, модификации или удалении записи.

В СУБД можно создавать триггеры, работающие при следующих шести условиях:

* до вставки записи (BEFORE INSERT);
* после вставки записи (AFTER INSERT);
* до удаления записи (BEFORE DELETE);
* после удаления записи (AFTER DELETE);
* до модификации записи (BEFORE UPDATE);
* после модификации записи (AFTER UPDATE).

В SQLiteStudio триггер может срабатывать при возникновении одного из нескольких событий. Обычно триггеры используются для задания сложных правил контроля целостности базы данных, которые невозможно реализовать с помощью ограничений.

Недостатком триггеров является их влияние на производительность операций с базой данных.

Синтаксис оператора создания триггера:

|  |  |
| --- | --- |
|  | CREATE TRIGGER my\_u\_log BEFORE INSERT    ON users    BEGIN    INSERT INTO user\_log(id\_u, u\_date) VALUES (NEW.id, datetime('now'));    END; |

Внутри тела триггера доступны две контекстные переменные **NEW** и **OLD**, которые позволяют получить доступ к новым и старым значениям полей записи, при изменении которой был вызван триггер.

## Ход работы

Операции по созданию генераторов, исключений и триггеров с помощью выполнения сценария были изучены в ходе выполнения предыдущих лабораторных работ.

Для создания триггеров с помощью программы " SQLiteStudio " в диалоговом режиме необходимо выполнить следующие действия:

1. Подключиться к базе данных.
2. Открыть окно с таблицей, для которой будет создаваться триггер.
3. Перейти на вкладку "Триггеры" (Рис. 1).
4. Нажать в этом окне правую кнопку мыши на одном из событий, для которого будет создаваться триггер.
5. Откроется контекстное меню, в котором надо выбрать команду "Новый триггер".

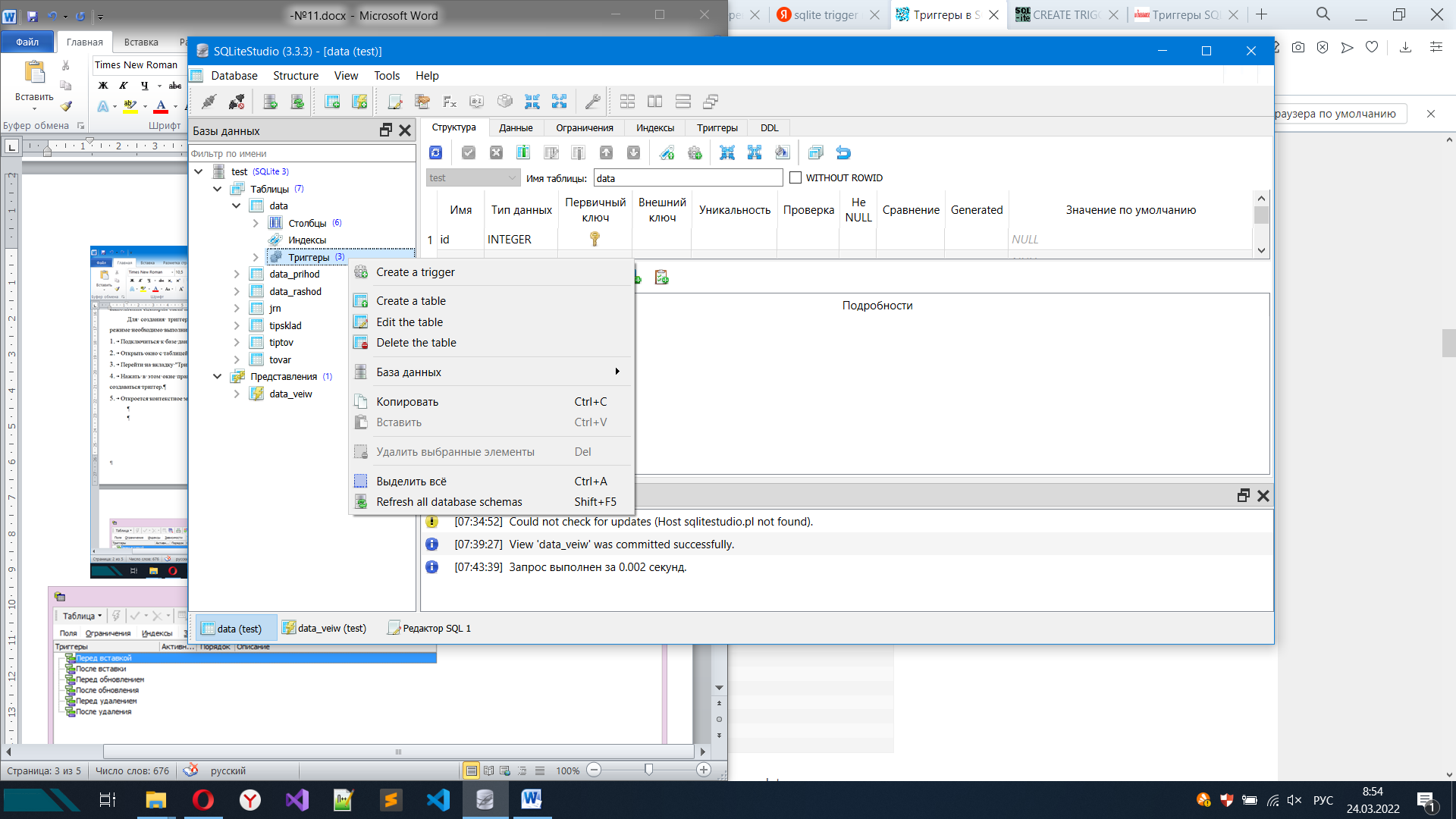


Рис. 1: Вкладка "Triggers" окно просмотра таблицы

1. В результате откроется окно создания триггера (рис. 2), в котором необходимо ввести «Имя триггера»; по какому событию должен он вызываться; когда должно срабатывать событие. Далее пишем тело триггера, а затем нажать кнопку выполнения [**ОК**].
2. Для просмотра полного кода триггера перейдите на вкладку «DDL» (рис 3).

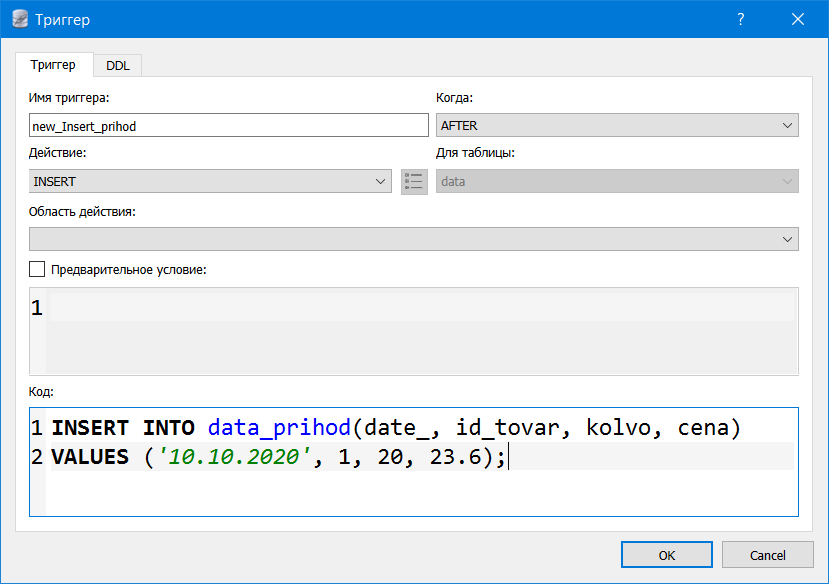


Рис. 2: Окно создания триггера

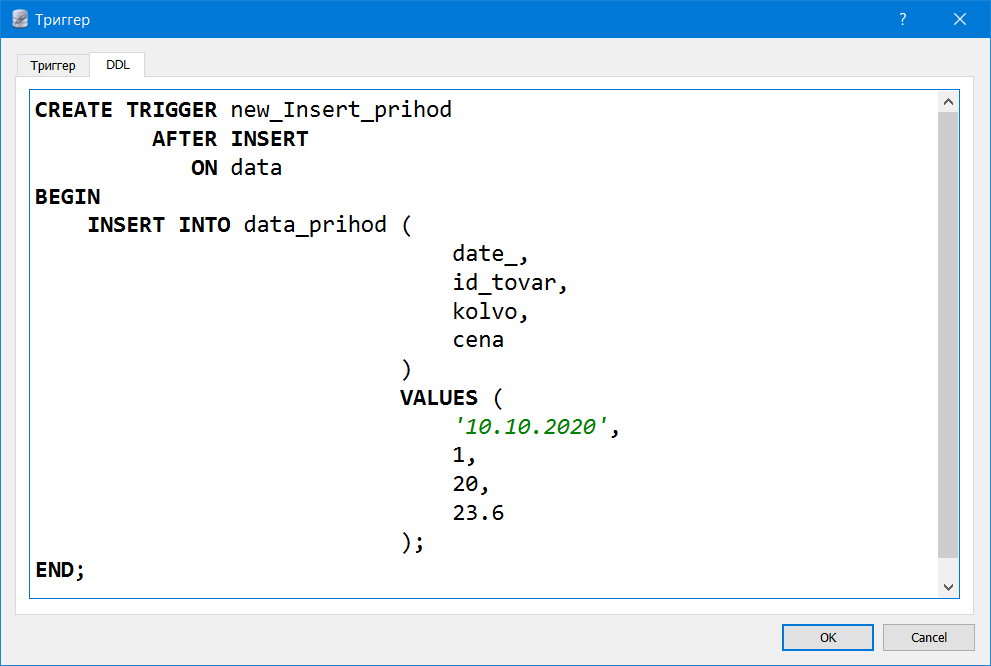
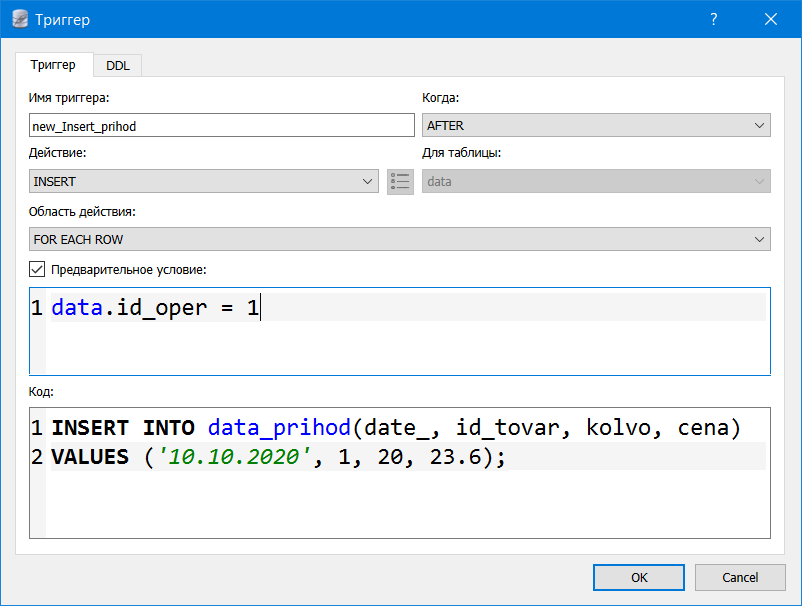


Рис. 3: Окно создания триггера в режиме кода

**Условия в запросе**

Если требуется добавить дополнительное условие срабатывания, то необходимо активировать соответствующий блок и задать условие.



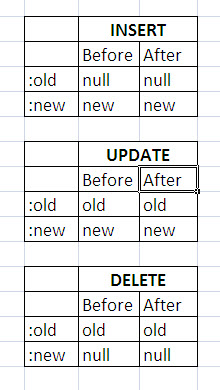
## Рис. 4

**Переменные New и Old**

Переменная **NEW** содержит кортеж данных, которые будут записаны в таблицу, для которой создается триггер.

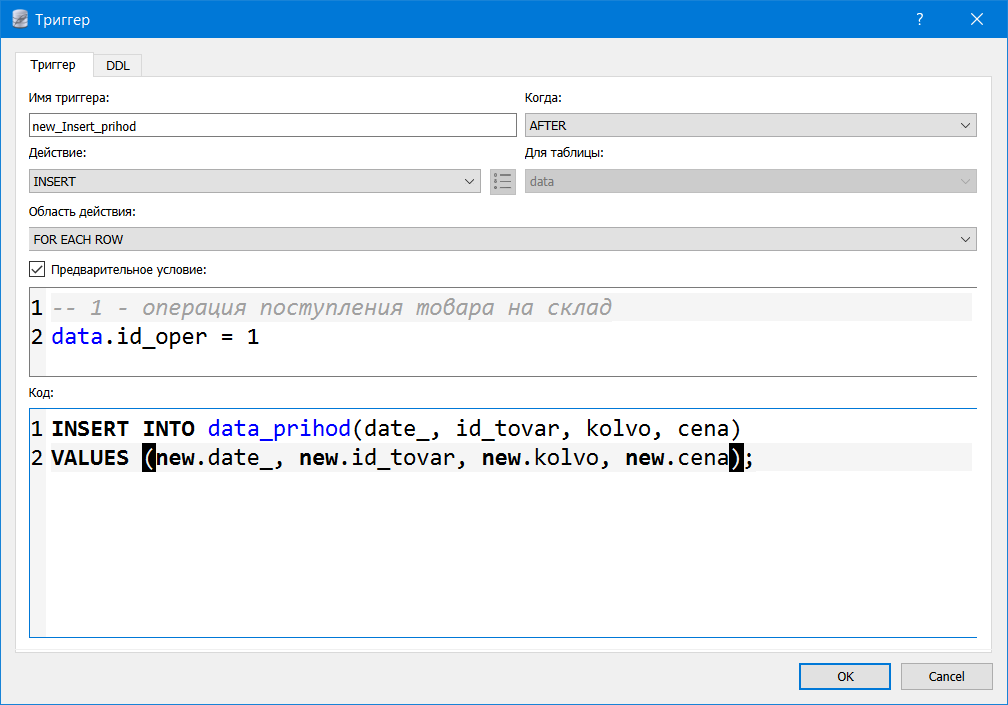
Переменная **OLD** содержит кортеж, который уже содержится в таблице, и над которым будет производиться изменения.

Данные переменные могут вызываться только при определенных условиях. На рисунке 5 представлены таблицы соответствия вызовов данных переменных.



## Рис. 5

Пример использования переменной **new**



## Задание

Работу следует выполнять в следующем порядке:

1. Открыть в приложении " SQLiteStudio " этот сценарий.
2. Добавить в сценарий операторы создания триггеров.
3. Создать триггеры, реализующие ссылочную целостность, заменив одно ограничение внешнего ключа. Команды создания этого внешнего ключа из сценария не удалять, а добавить команду удаления этого ограничения перед созданием триггеров. Также добавить оператор создания индекса по полю бывшего внешнего ключа. Триггеры должны обеспечивать каскадное обновление и удаление зависимых данных, а также запрет ссылки на отсутствующую запись.
4. Создать не менее **пяти триггеров**, реализующих бизнес-правила в соответствии с выданным индивидуальным заданием. Для каждого триггера должны присутствовать комментарии, поясняющие выполняемые операции.

## Отчет о выполнении работы

Отчет о выполнении необходимо оформить на листах формата A4. Отчет должен содержать описание и результаты работы, представляемые в следующей последовательности:

1. Описание бизнес-правил, реализованных с помощью триггеров.
2. Распечатка сценария создания базы данных с комментариями, которые должны содержать сведения об авторе, дате создания, всех выполняемых действиях и пояснения к генераторам, исключениям и триггерам.