Министерство образования и науки РФ

Пермский национальный исследовательский политехнический университет

Электротехнический факультет

Кафедра Информационные технологии и автоматизированные системы

Базы данных

Лабораторная работа № 5

Тема: «Триггеры»

Выполнил:

студент группы РИС-23- 2б

Колосов Д. Е.

Проверил:

доцент кафедры ИТАС

Петренко А.А.

г. Пермь – 2024

Оглавление

Цель работы 3

Задачи работы 4

Этапы выполнения 5

Создание таблиц 5

Создание триггеров 6

Заключение 9

Список используемой литературы 10

# Цель работы

Создать таблицу логов, заполняемую с помощью триггеров команд добавления, изменения и удаления атрибутов и записей, привязанных к основной таблице

# Задачи работы

* Создать две таблицы – основную и содержащую логи к основной
* Для заполнения таблицы логов создать триггеры, привязанные к действиям над основной таблицей

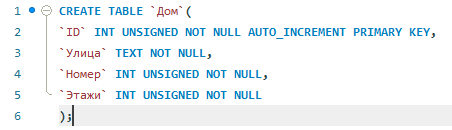
# Этапы выполнения

## Создание таблиц

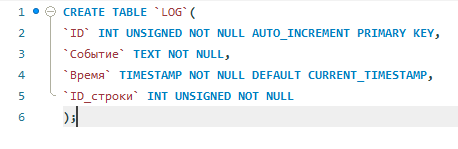
Таблицы создаются в локальной базе данных «kolosovLab5» в СУБД «MySQL workbench».

Создадим две таблицы: первая таблица имеет имя «» и будет содержать информацию о мебели, вторая таблица - «log», она содержит информацию, которую ей отправят триггеры после их вызова.

Код создания таблиц имеет следующий вид, создание таблицы Дом на Рис 1 и таблицы LOG на Рис 2:



Рис

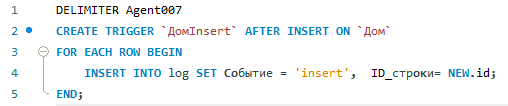


Рис

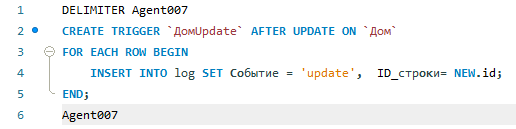
В итоге таблица «Дом» получила три столбца: id (уникальный идентификатор), Улица (название улицы) и Этажи (количество этажей); Таблица «log» получила четыре столбца: id (уникальный идентификатор), Событие (информация о событии, на которое был вызван триггер), Время (момент времени вызова триггера), ID\_строки (уникальный идентификатор дома из таблицы «Дом»).

## Создание триггеров

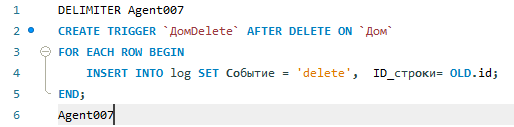
Для каждого действия INSERT, UPDATE, DELETE прописываем алгоритм работы каждого триггера**. Код триггеров** INSERT, UPDATE, DELETE показан на рисунках Рис 3, Рис 4 и Рис 5 соответственно

****

Рис

****

Рис

****

Рис

Каждый триггер привязан к определенному действию над таблицей «Дом». Триггер вызывается при совершении действия с строкой таблицы, на которую был навешен триггер. При вызове триггера для каждой строки таблицы выполняется указанная SQL команда: для триггера команды insert - устанавливается запись под атрибутом Событие как «insert», а ID\_строки как ID новой строки таблицы «Дом». Тесть **при добавлении записи в таблицу** «Дом» **в таблице log появится запись с сообщением «insert»**, а в поле ID\_строки будет сохранён идентификатор вставленной строки

**Стоит обратить внимание на ключевые слова OLD, NEW и SET.**

**OLD** — это состояние строки перед запросом. Для команды INSERT эта переменная содержит NULL, а для команд UPDATE и DELETE — старую строку базы данных.

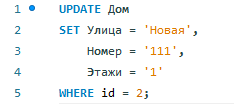
**NEW** — это текущее состояние строки. Переменная содержит новую строку базы данных для команд INSERT и UPDATE

SET — используется для того, чтобы явно задать значения столбцов (Событие и ID\_строки)

Выполним несколько команд для иллюстрации работы триггеров, они показаны на Рис 6, Рис 7 и Рис 8



Рис

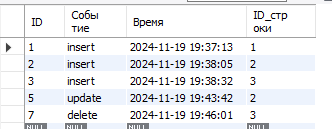


Рис



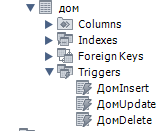
Рис

Проверим таблицу «log», Рис 9, и увидим, что результаты работы триггеров отображаются в ее строках.



Рис

На Рис 10 представлена структура таблицы Дом, где можно увидеть привязанные к ней триггеры



Рис

# Заключение

В рамках лабораторной работы были созданы триггеры, а основе таблицы Дом, добавляющие записи об изменениях в этой таблице, что позволяет отслеживать действия, проводимые с данной таблицей, а также в перспективе организовать процесс возвращения к старым версиям таблицы.

# Список используемой литературы

1. <https://dev.mysql.com/doc/workbench/en/wb-table-editor-triggers-tab.html>
2. <https://habr.com/ru/articles/37693/>
3. [https://doka.guide/js/promise-then/#:~:text=Метод%20then()%20используют%2C%20чтобы,в%20состояние%20«успех»%20fulfilled%20.20](https://doka.guide/js/promise-then/#:~:text=Метод%20then()%20используют%2C%20чтобы,в%20состояние%20)