Министерство образования и науки РФ

Пермский национальный исследовательский политехнический университет

Электротехнический факультет

Кафедра Информационные технологии и автоматизированные системы

Базы данных

Лабораторная работа № 5

Тема: «Создание и использование триггеров в СУБД MySQL»

Выполнил: студентка группы

РИС-23-2б

Жилина А.А.

Проверил: доцент кафедры ИТАС

Петренко А.А.

г. Пермь – 2024

**Оглавление**

Цель работы 3

Задачи работы 4

Этапы выполнения 5

Вывод 15

Заключение 16

Список используемой литературы 17

# Цель работы

Создать и использовать триггеры в СУБД *MySQL* в веб-приложении *phpMyAdmin*.

# Задачи работы

Для достижения поставленной цели необходимо выполнить следующие задачи:

1. Изучить понятие триггер;
2. Создать базу данных и таблицы;
3. Создать триггеры (на добавление, изменение и удаление строк в таблице);
4. Заполнить таблицу *table\_students\_zhilina* и посмотреть результаты в таблице *LOG*.

# Этапы выполнения

1. Понятие триггер в *MySQL*, и его создание:

Триггер в *MySQL* — это определяемая пользователем *SQL*-команда, которая автоматически вызывается во время операций ***INSERT***, ***DELETE*** или ***UPDATE***. Код триггера связан с таблицей и уничтожается после удаления таблицы.

Существует возможность определения времени действия триггера и указания его нужно активации – до или после определенного события базы данных.

Для создания триггера применяется выражение ***CREATE TRIGGER***, после которого идет имя триггера. Как правило, имя триггера отражает тип операций и имя таблицы, над которой производится операция.

Каждый триггер ассоциируется с определенной таблицей или представлением, имя которых указывается после слова ***ON***.

Затем устанавливается тип триггера. Мы можем использовать один из двух типов:

* ***AFTER***: выполняется после выполнения действия. Определяется только для таблиц.
* ***BEFORE***: выполняется до выполнения действия. Определяется для таблиц и представлений

После типа триггера идет указание операции, для которой определяется триггер: ***INSERT***, ***UPDATE*** или ***DELETE***.

Для триггера ***AFTER*** можно применять сразу для нескольких действий, например, ***UPDATE*** и ***INSERT***. В этом случае операции указываются через запятую. Для триггера ***BEFORE*** можно определить только одно действие.

И затем после слова ***AS*** идет набор выражений *SQL* (*SQL* запрос), которые составляют тело триггера.

1. Создание таблиц (*table\_students\_zhilina* и *LOG*).

Заходим в *phpMyAdmin* и открываем созданную ранее базу данных [*Zhilina*](http://127.0.0.1/openserver/phpmyadmin/index.php?route=/database/structure&db=Zhilina)и добавляем таблицу *table\_students\_zhilina* с следующими столбцами:

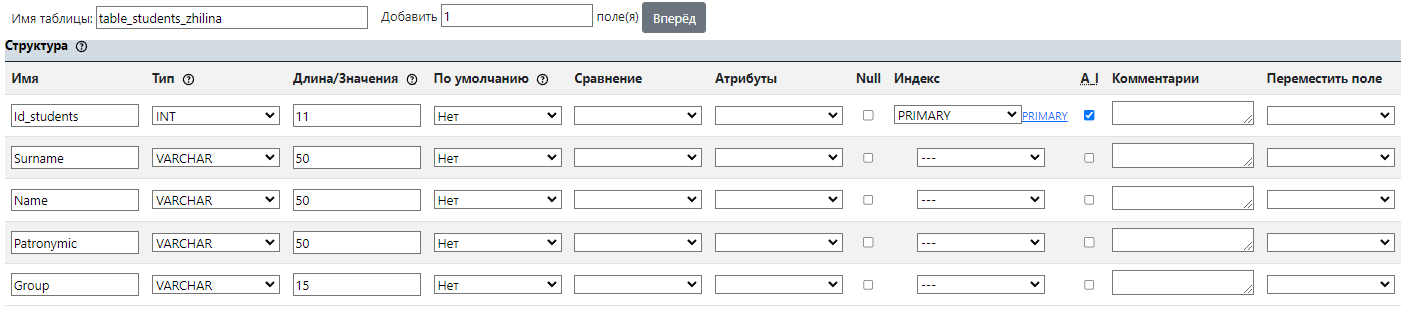
*Id\_students* – идентификатор, тип *INT*, с *PRIMARY* ключом, дополнительно выбираем *AUTO*\_*INCREMENT*.

*Surname* – фамилия студента, тип *VARCHAR*, максимальное значение символов 50.

*Name* – имя студента, тип *VARCHAR*, максимальное значение символов 50.

*Patronymic* – отчество студента, тип *VARCHAR*, максимальное значение символов 50.

*Group* – название группы, тип *VARCHAR*, максимальное значение символов 15. (См. Рисунок 1)

**

*Рисунок 1 – создание столбцов таблицы table\_students\_zhilina*

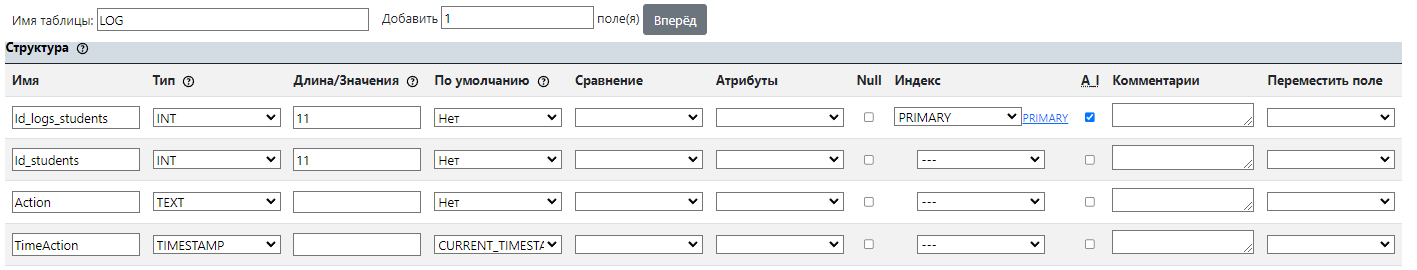
Создаем ещё одну таблицу с названием *LOG*, которая будет хранить все изменения, произошедшие в таблице *table\_students\_zhilina*. Для этого добавим следующие столбцы:

*Id\_logs\_students* – идентификатор, тип *INT*, с *PRIMARY* ключом, в дополнительно выбираем *AUTO*\_*INCREMENT*.

*Id\_students* – идентификатор продукта, тип *INT.*

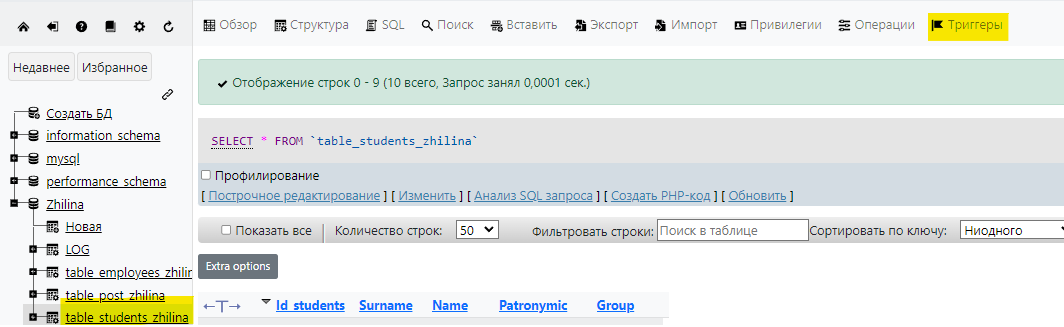
*Action* – операция, произошедшая в таблице *table\_students\_zhilina*, тип *TEXT*.

*TimeAction* – точное время изменения, тип *TIMESTAMP*, значение по умолчанию *CURRENT*\_*TIMESTAMP*. (См. Рисунок 2)

**

*Рисунок 2 – создание столбцов таблицы LOG*

1. Создание триггеров (на добавление, изменение и удаление строк в таблице):
   1. Открываем таблицу *table\_students\_zhilina,* нажимаем на кнопку Триггеры (См. Рисунок 3).

**

*Рисунок 3 – кнопка триггеры*

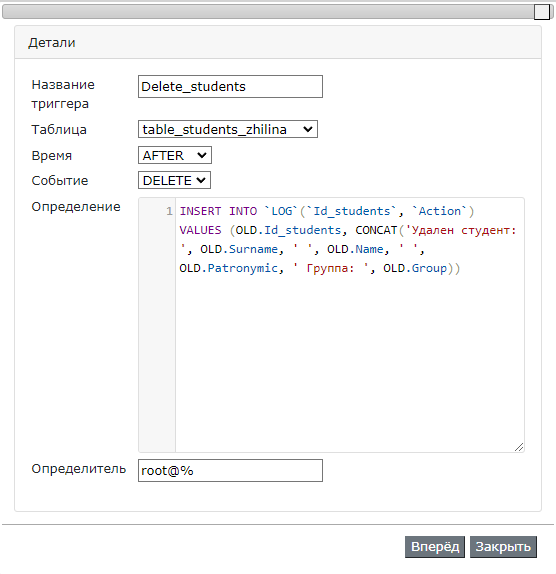
2.2. Добавим триггер, который будет срабатывать при удалении строки. Заполняем следующие поля: название триггера, таблица в которой будет срабатывать триггер, время срабатывания триггера, событие при котором срабатывает, определение тела триггера (*SQL* запрос), определитель оставляем по умолчанию. (См. Рисунок 4)

В поле «Определение» запишем следующий *SQL* запрос:

INSERT INTO `LOG` (`Id\_students`, `Action`) VALUES (OLD.Id\_students, CONCAT('Удален студент: ', OLD.Surname, ' ', OLD.Name, ' ', OLD.Patronymic, ' Группа: ', OLD.Group))

**Ключевые слова:**

* INSERT INTO `LOG` (`Id\_students`, `Action`) – Вставить в таблицу *LOG*  в поля Id\_students, Action.
* VALUES – значения.
* OLD – предыдущее значение элемента таблицы, вызывается в обязательном порядке, если триггер имеет тип *DELETE*.
* CONCAT – встроенная функция, которая объединяет переданные строки.

**

*Рисунок 4 – создание триггера Delete\_students*

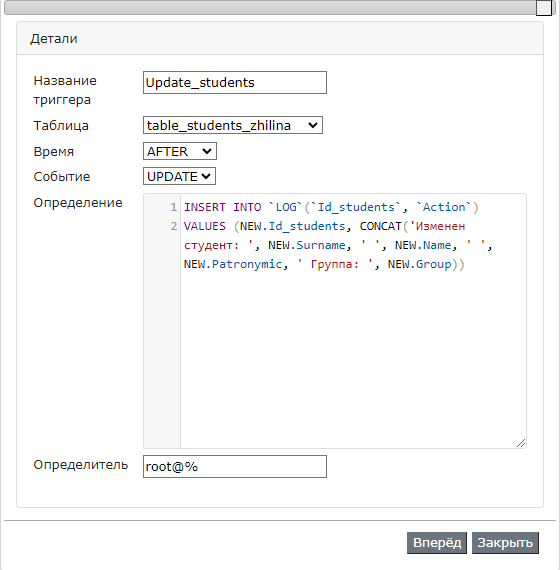
2.3. Добавим триггер, который будет срабатывать при изменении строки. (См. Рисунок 5)

В поле «Определение» запишем следующий *SQL* запрос:

INTO `LOG` (`Id\_students`, `Action`) VALUES (NEW.Id\_students, CONCAT('Изменен студент: ', NEW.Surname, ' ', NEW.Name, ' ', NEW.Patronymic, ' Группа: ', NEW.Group))

**Ключевые слова:**

* NEW – новые значение элемента таблицы, необязателен.

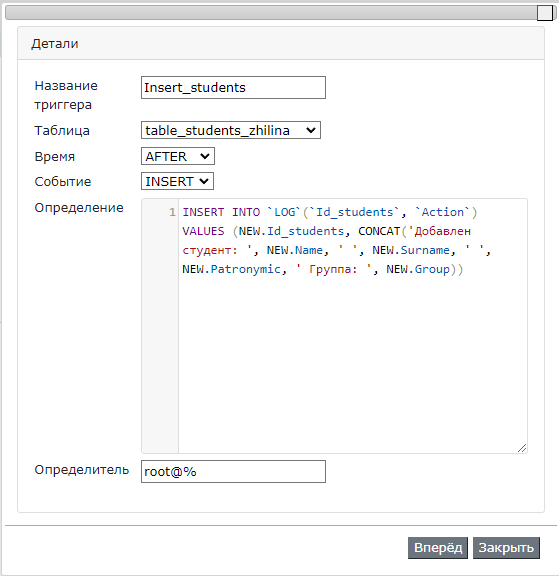
**

*Рисунок 5 – создание триггера Update\_students*

* 1. Добавим триггер, который будет срабатывать при добавлении строки. (См. Рисунок 6)

В поле «Определение» запишем следующий *SQL* запрос:

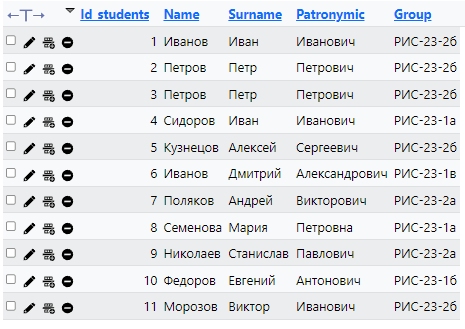
INSERT INTO `LOG` (`Id\_students`, `Action`) VALUES (NEW.Id\_students, CONCAT('Добавлен студент: ', NEW.Surname, ' ', NEW.Name, ' ', NEW.Patronymic, ' Группа: ', NEW.Group))

**

*Рисунок 6 – создание триггера Insert\_students*

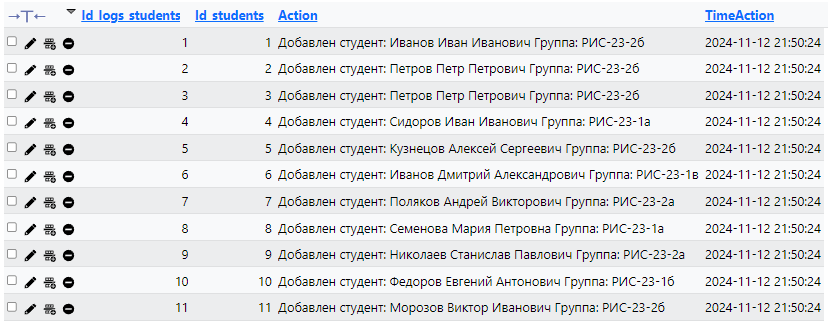
3. Заполнение таблицы *table\_students\_zhilina* проверка результатов в таблице *LOG*.

Заполняем строки в таблице *table\_students\_zhilina* (См. Рисунок 7)

**

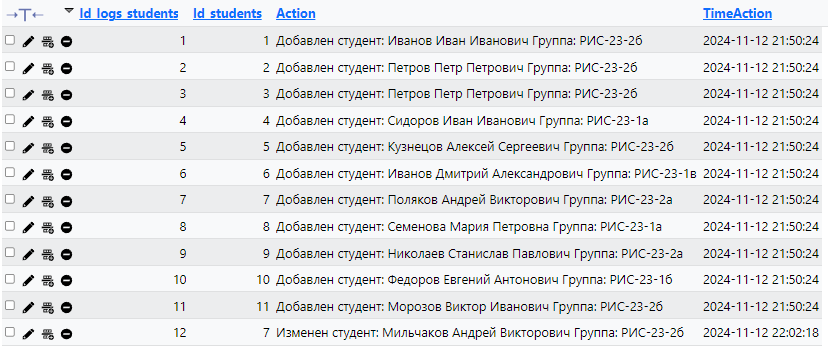
*Рисунок 7 – заполнение строк в таблице table\_students\_zhilina*

В таблице *LOG* с помощью триггера заполнились строки на добавление (См. Рисунок 8)

**

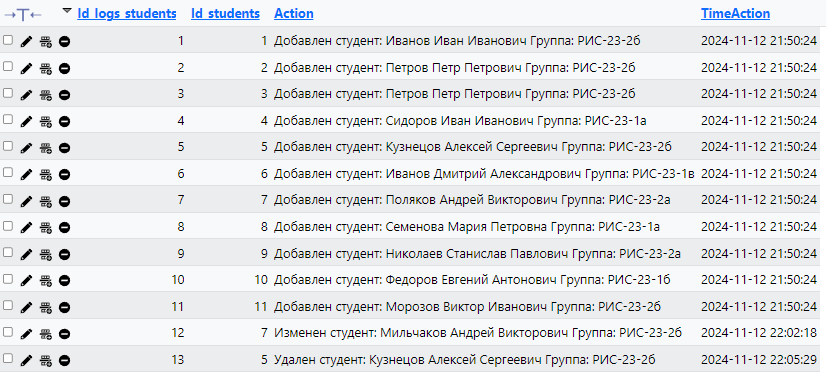
*Рисунок 8 – таблица LOG после добавления строк*

В таблице *table\_students\_zhilina* изменим седьмую строку, в столбце *Surname* «*Поляков*» поменяем на «*Мильчаков*» и в столбце *Group* «*РИС-23-2а*» поменяем на «*РИС-23-2б*», с помощью триггера на изменение строки в таблицу *LOG* добавится строка 12. (См. Рисунок 9)

**

*Рисунок 9 – таблица LOG после изменения строки*

В таблице *table\_students\_zhilina* удалим пятую строку, с помощью триггера на удаление строки в таблицу *LOG* добавится строка 13. (См. Рисунок 10)

**

*Рисунок 10 – таблица LOG после удаления строки*

# Вывод

В результате работы были получены навыки работы с триггерами, их создание и подключение к таблицам, а также навыки работы с СУБД *MySQL* и веб-приложением *phpMyAdmin*.

# Заключение

Результатом проведенной работы, являются:

1. Разработанная база данных.
2. Таблицы, созданные с применением СУБД MySQL.
3. Подключенные к таблице триггеры (на добавление, изменение и удаление строк в таблице).

# Список используемой литературы

1. Mysql Insert Trigger Using PhpMyAdmin https://www.youtube.com/watch?time\_continue=110&v=r9jR4FR9INw&embeds\_referring\_euri=https%3A%2F%2Fyastatic.net%2Fvideo-player%2F0x4add4491140%2Fpages-common%2Fyoutube%2Fyoutube.html&source\_ve\_path=Mjg2NjY (дата обращения 12.11.2024);
2. Копейкин М.В., Спиридонов В.В., Шумова Е.О. Базы данных. Основы SQL реляционных баз данных: Учебное пособие. - СПб.: СЗТУ, 2005. - 160 с;
3. Сравнение языков программирования https://www.cyberforum.ru/mysql/thread1259738.html (дата обращения 12.11.2024);
4. Э.В. Сысоев, Е.В. Бурцева. Базы данных: лекции к курсу. - Тамбов : Изд-во Тамб. гос. техн. ун-та, 2007. - 48 с.