**Введение**

**СУБД MySQL** является превосходным сервером БД. Основная причина её популярности – это быстрота (производительность) и стабильность. Система MySQL хорошо проверена и надёжна. MySQL является прекрасным инструментом изучения удаленных баз данных, благодаря простоте использования, а также исключительно скромными требованиями к дисковому пространству и памяти. В MySQL используется двойная схема лицензирования: бесплатная и коммерческая.

Пользовательский интерфейс представляет собой «интерфейс командной строки», который называется «монитор MySQL». При установке MySQL по умолчанию устанавливается и Монитор MySQL.

**Операторы SQL**

Язык **SQL** (structured query language – язык структурированных запросов) разработан фирмой IBM в 1974 году и был принят Американским Национальным институтом стандартов (ANSI) в качестве национального стандарта США в 1992 году. Операторы SQL поддерживаются большинством СУБД. Язык SQL ориентирован на текст. Т.к. он был разработан задолго до появления графических интерфейсов пользователя, то для работы с ним требуется лишь текстовый редактор.

Этот язык представляет собой настолько удачную программную разработку для манипулирования данными реляционной модели, что в настоящее время он является наиболее распространенным программным продуктом такого рода.

**Правила**:

1. Ключевые слова SQL не зависят от регистра
2. каждая команда и каждый оператор MySql заканчивается символом «точка с запятой»
3. **[ ]** не обязательный элемент

**Триггеры**

Поддержка триггеров появилась в MySQL начиная с версии 5.0.2.

**Триггер** – именованный объект БД, который ассоциирован с таблицей и активируемый при наступлении определенного события, связанного с этой таблицей.

**Триггер** – это процедура, которая автоматически начинает работает при наступлении определённого события: добавления новой записи в таблицу базы данных, редактирования записи или удаления записи из таблицы БД.

В MySQL для каждой таблицы базы данных можно одновременно хранить на сервере максимум *шесть* триггеров. Для каждого события для одной таблицы БД можно сохранить два триггера с учётом времени их срабатывания: «**до** наступления события» и «**после** наступления события».

Общий вид оператора создания триггера

**CREATE** **TRIGGER** имя\_триггера время\_триггера событие\_срабатывания\_триггера **ON** имя\_таблицы **FOR EACH ROW** выражение\_выполняемое\_при\_срабатывании\_триггера ;

ВРЕМЯ\_ТРИГГЕРА

Определяет время свершения действия триггера. **BEFORE** означает, что триггер выполнится до завершения события срабатывания триггера, а **AFTER** означает, что после.

СОБЫТИЕ\_СРАБАТЫВАНИЯ\_ТРИГГЕРА

INSERT при операциях вставки ( INSERT, LOAD DATA )

UPDATE при изменении записи

DELETE при удалении записи

В теле триггера при обращении к полю таблицы БД перед именем поля необходимо указывать **New** или **Old**. Если данные для таблицы новые, то New, если данные уже были в таблице, то Old.

Пример 1:

CREATE TRiGGER primer1 BEFORE iNSERT ON SPORT FOR EACH ROW

SET New. VES\_Tochno = New.VES – 1 ;

В таблице SPORT имеются поля VES и VES\_Tochno.

Во время добавления новой записи к таблице SPORT, пользователь в поле VES вводит данные, а значение для поля VES\_Tochno вводится NULL.

Но при этом поле VES\_Tochno заполняется автоматически с учётом алгоритма записанного в триггере.

Пример 2:

CREATE TRiGGER primer2 AFTER iNSERT ON SPORT FOR EACH ROW

INSERT INTO ARHiV VALUES (NULL, New.Nom, New.FiO) ;

В базе данных существует таблица ARHiV c тремя полями: автоинкрементное ключевое, для номера спортсмена и для ФИО спортсмена.

Во время добавления новой записи к таблице SPORT, автоматически добавляется новая запись к таблице ARHiV, и значения полей Nom и Fio записываются не только в таблицу SPORT, но и в таблицу ARHiV.

Пример 3:

В базе данных существует таблица ARHiV\_1 c тремя полями: автоинкрементное ключевое, для номера спортсмена и для ФИО спортсмена.

Создадим двумя способами триггер, который во время удаления записи из таблицы SPORT, автоматически будет добавлять эту запись к таблице ARHiV\_1 и значения полей Nom и Fio удаляемой из таблицы SPORT записи перенесутся в таблицу ARHiV\_1.

Способ первый

Mysql> DELIMITER #

Mysql> CREATE TRiGGER primer3 BEFORE DELETE ON SPORT FOR EACH ROW

BEGIN

SET @ A = OLD.Fio ;

SET @ B = OLD.Nom ;

INSERT INTO ARHiV\_1 VALUES (NULL, @ B, @ A) ;

END #

Mysql> DELIMITER ;

Способ второй

CREATE TRiGGER primer3 BEFORE DELETE ON SPORT FOR EACH ROW

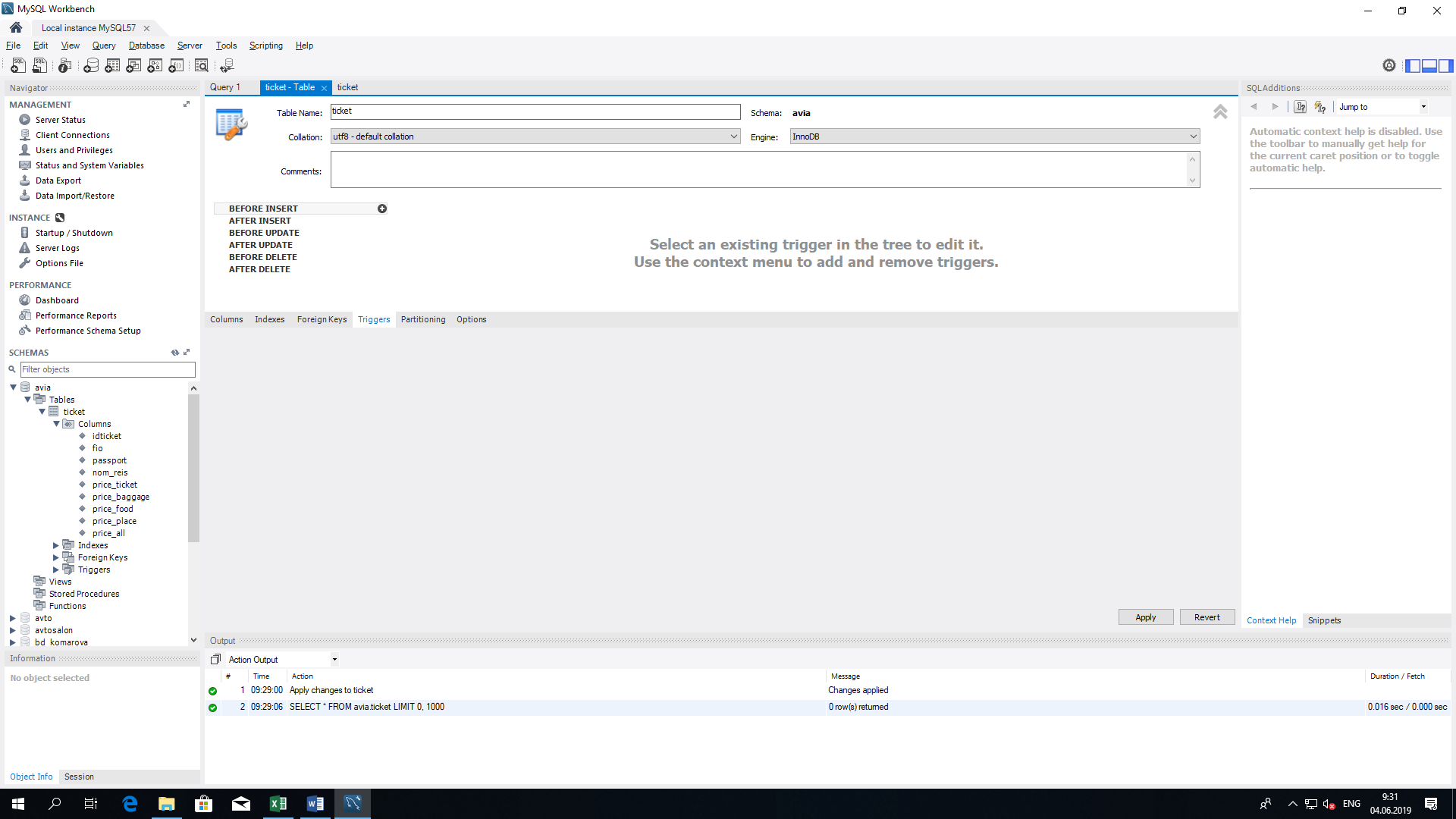
INSERT INTO ARHiV\_1 VALUES (NULL, OLD.Nom, OLD.Fio) ;

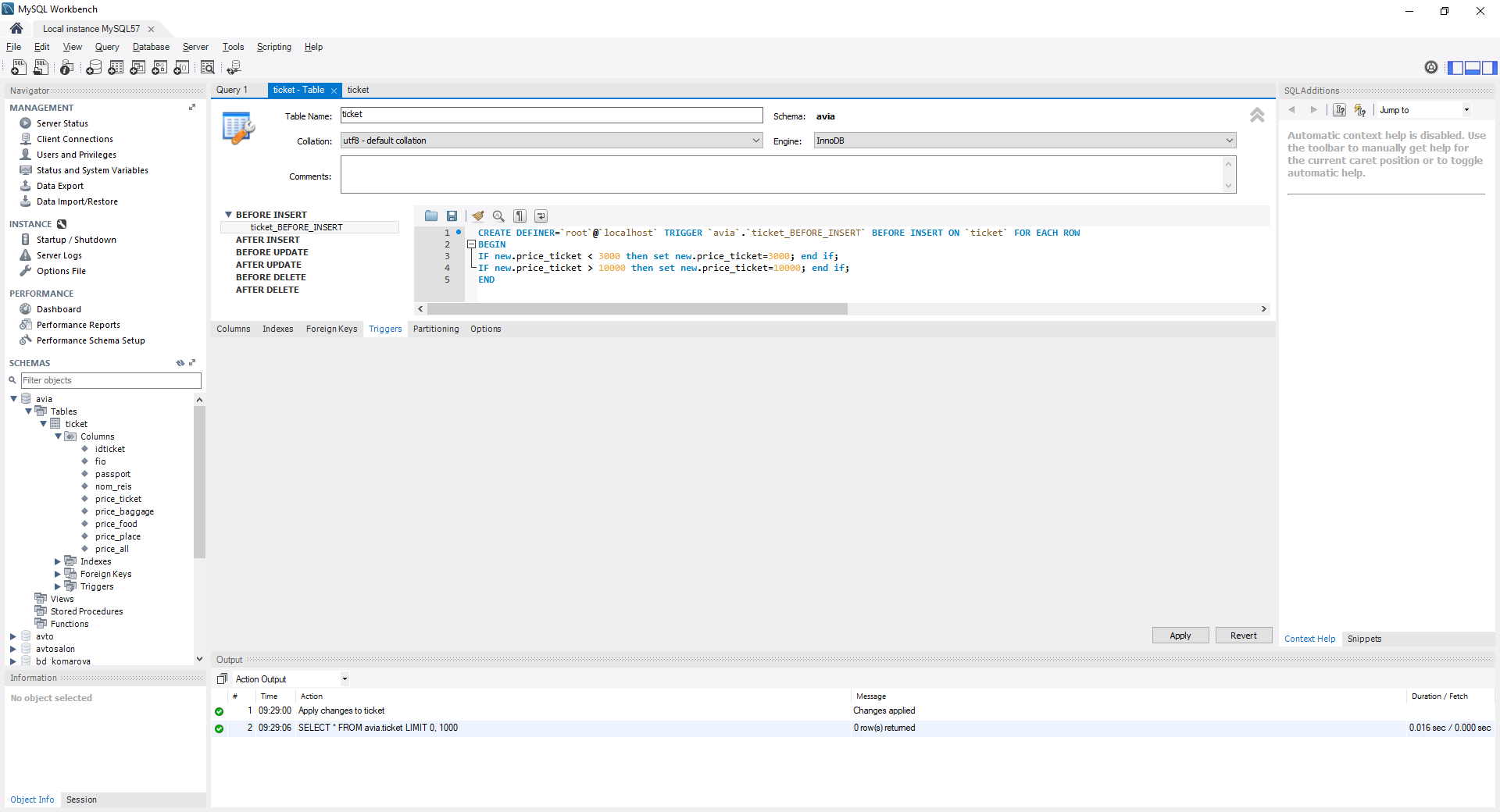
Удаление триггера

**DROP** **TRIGGER** имя\_триггера ;

Триггеры можно использовать для проверки корректности ввода данных.

В MySql Workbench при создании структуры таблицы базы данных, на закладке «Triggers» надо выбрать одно из шести возможных событий и в теле процедуры-триггера прописать необходимые действия.





Лабораторная работа

***Тема*: Создание триггеров в MySQL**

***Цель*:** Обрести практические навыки создания триггеров в СУБД MySQL

**Задания лабораторной работы:**

1. Установите связь с сервером
2. Создайте на сервере свою базу данных с именем, соответствующим Вашей фамилии (например, BDKomarova)
3. Создайте таблицу Student\_ВАША ФАМИЛИЯ (например, Student\_Komarova)содержащую поля: N\_stud (автоинкрементное), FIO(символьное), D\_R (дата), Gruppa(символьное), Stipendiya (вещественное) , Adres(символьное)
4. Просмотрите структуру таблицы
5. Заполните таблицу Student\_ВАША ФАМИЛИЯ следующими записями (10 студентов)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| N\_Stud | FIO | D\_R | Gruppa | Stipendiya | Kod\_spec |
| 1 | Сидоров | 1992-07-23 | 4п1 | 880 | 3322 |
| 2 | Петров | 1993-11-07 | 4п3 | NULL | 4466 |
| 3 | Григорьев | 1992-04-19 | 4п2 | 920 | 8811 |
| 4 | Данилова | 1992-06-11 | 4п2 | 1200 | 8811 |
| 5 | Андреев | 1993-11-06 | 4п1 | NULL | 3322 |
| 6 | Серова | 1991-02-25 | 4п1 | NULL | 3322 |
| 7 | Коробкова | 1992-11-20 | 4п2 | 880 | 8811 |
| 8 | Смирнов | 1991-07-16 | 4п2 | 920 | 8811 |
| 9 | Подгорнова | 1991-09-25 | 4п3 | 1200 | 4466 |
| 10 | Лебедев | 1991-07-26 | 4п3 | 920 | 4466 |

1. Проверьте заполнение таблицы. Для этого составьте запрос для вывода на экран всей информации о студентах
2. Измените структуру таблицы Student, добавьте в неё поле Stip\_New (новая стипендия)
3. Создайте триггер TR1 на добавление новой записи в таблицу Student, который рассчитывает новую стипендию студента, увеличивая базовый размер стипендии в 2 раза.
4. Проверьте работу триггера. Для этого добавьте в таблицу Student две записи
5. Составьте запрос для вывода на экран всей информации о студентах
6. Измените структуру таблицы Student, добавьте в неё поле PREMiYA
7. Назначьте каждому студенту премию в размере 500 рублей
8. Создайте триггер TR2 на изменение записи в таблице Student, который рассчитывает премию студента, как сумму размера базовой стипендии и 1000 рублей.
9. Проверьте работу триггера. Для этого измените размер стипендии студента АНДРЕЕВ на 1200
10. Измените размер стипендии студентки СЕРОВА на 900
11. Cоставьте запрос для вывода на экран всей информации о студентах
12. Создайте новую таблицу базы данных ARHiV\_1, которая содержит следующие поля: номер записи (целочисленное, ключевое, автоинкрементное), номер студента, фамилия студента, сумма стипендии и премии
13. Создайте триггер TR3 на удаление записи из таблицы Student, который при удалении записи из таблицы Student записывает в таблицу ARHiV\_1 информацию: номер студента, фамилия студента, сумма полей Stipendiya и Premiya
14. Проверьте работу триггера. Для этого удалите из таблицы Student информацию о студентах АНДРЕЕВ, СЕРОВА
15. Cоставьте запрос для вывода на экран всей информации о студентах
16. Cоставьте запрос для вывода на экран всей информации из таблицы ARHiV\_1
17. Просмотрите все имеющиеся в базе данных триггеры
18. Удалите все созданные триггеры TR1 .. TR3
19. Создайте новую таблицу базы данных ARHiV\_2, которая содержит следующие поля: номер записи (целочисленное, ключевое, автоинкрементное), фамилия студента, группа и размер стипендии
20. Создайте триггер TR4 на добавление новой записи в таблицу Student, который записывает фамилию студента, его группу и увеличенную стипендию студента в таблицу ARHiV\_2.
21. Проверьте работу триггера. Для этого добавьте в таблицу Student две записи о студентах Вашей учебной группы
22. Cоставьте запрос для вывода на экран всей информации о студентах
23. Cоставьте запрос для вывода на экран всей информации из таблицы ARHiV\_2 и убедитесь в правильной работе триггера.
24. Измените структуру таблицы Student, добавив в неё поле YYY (год рождения)
25. Измените структуру таблицы Student, добавив в неё поле MMM (месяц рождения)
26. Измените структуру таблицы Student, добавив в неё поле DDD (день рождения)
27. Создайте триггер TR5 на добавление новой записи в таблицу Student, который выделяет из даты рождения студента отдельно год, месяц, число и записывает их в соответствующие поля таблицы.
28. Проверьте работу триггера. Для этого добавьте в таблицу Student две записи о студентах Вашей учебной группы
29. Cоставьте запрос для вывода на экран всей информации о студентах и убедитесь в правильной работе триггера
30. Создайте триггер TR6 на редактирование записи в таблице Student, который переводит студента в группу заочного отделения 2ЗАО1.
31. Проверьте работу триггера. Для этого измените двум последним студентам размер премии на Ваше усмотрение
32. Cоставьте запрос для вывода на экран всей информации о студентах и убедитесь в правильной работе триггера.
33. Создайте триггер TR7 на своё усмотрение и продемонстрируйте его работу.
34. Просмотрите все имеющиеся в базе данных триггеры.
35. Удалите триггеры.
36. Оформите отчёт по лабораторной работе

Отчёт по лабораторной работе должен представлять собой файл **Отчёт.doc**, дополненный рисунками экранных форм с результатами выполнения каждого задания.

Например,

