## **Лабораторная работа № 5 Создание триггеров**

**Теоретические сведения**

Рассмотрим следующие вопросы:

* понятие триггера;
* создание триггеров с помощью оператора *CREATE trigger*;
* удаление триггеров с помощью оператора *DROP trigger*.

Триггер – эта та же хранимая процедура, но привязанная к событию изменения со­держимого конкретной таблицы.

Возможны три события, связанных с изменением содержимого таблицы, к которым можно привязать триггер:

* *insert* – вставка новых данных в таблицу;
* *delete* – удаление данных из таблицы;
* *Update* – обновление данных в таблице.

Например, при оформлении нового заказа, т. е. при добавлении новой записи в таблицу *orders*, можно создать триггер, автоматически вычитающий число заказанных товарных позиций в таблице *books*.

**Создание триггеров**

Создать новый триггер позволяет оператор:

*CREATE TRIGGER trigger\_name trigger\_time trigger\_event*

*ON tbl\_name FOR EACH ROW trigger\_stmt ;*

Оператор создает триггер с именем *trigger\_name*, привязанный к таблице *tbl\_name*. Не допускается привязка триггера к временной таблице или представлению. Конструкция *trigger\_time* указывает момент выполнения триггера и может прини­мать два значения:

* *before* – действия триггера производятся до выполнения операции изменения таблицы;
* *after* – действия триггера производятся после выполнения операции изменения таблицы.

Конструкция *trigger\_event* показывает, на какое событие должен реагировать триггер, и может принимать три значения:

* *insert* – триггер привязан к событию вставки новой записи в таблицу;
* *update* – триггер привязан к событию обновления записи таблицы;
* *delete* – триггер привязан к событию удаления записей таблицы.

Для таблицы *tbl\_name* может быть создан только один триггер для каждого из со­бытий *trigger\_event* и момента *trigger\_time*. Таким образом, для каждой из таблиц мо­жет быть создано всего шесть триггеров.

Конструкция *trigger\_stmt* представляет тело триггера – оператор, который необ­ходимо выполнить при возникновении события *trigger\_event* в таблице *tbl\_name*.

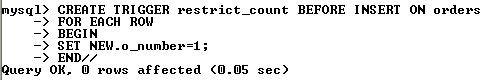
Если требуется выполнить несколько операторов, то необходимо использовать составной оператор *begin ... end*. Синтаксис и допустимые операторы такие же, как и у хранимых процедур. Внутри составного оператора *begin ... end* допускаются все специфичные для хранимых процедур операторы и конструкции:

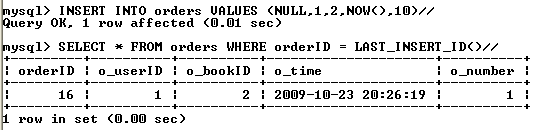
* другие составные операторы *begin ... end*;
* операторы управления потоком *(if, case, while, loop, repeat, leave, iterate)*;
* объявления локальных переменных при помощи оператора *declare* и назначение им значений при помощи оператора *set*;
* именованные условия и обработчики ошибок.

В MySQL триггеры нельзя привязать к каскадному обновлению или удалению записей из таблицы типы *InnoDB* по связи первичный ключ/внешний ключ.

Триггеры сложно использовать, не имея доступа к новым записям, которые вставляются в таблицу, или старым записям, которые обновляются или удаляются. Для доступа к новым и старым записям используются префиксы *new* и *old* соответст­венно. Если в таблице обновляется поле *total*, то получить доступ к старому значению можно по имени *old.total*, а к новому – *new.total*.

Пример простейшего триггера для учебной БД *book* см. в пункте «Пример выполнения работы» (пример 1). Он демонстрирует работу триггеров после добавления запи­си в таблицу без вмешательства в запрос. Рассмотрим триггер, который будет включаться до вставки новых записей в таблицу *orders* и ограничивает число заказываемых товаров до 1:





Часто при обновлении полей таблицы производится попытка добавления некорректных значений. Пример триггера, который при добавлении нового покупателя преобразует полные имена и отчества в инициалы, см. в пункте «Пример выполнения работы» (пример 2). Он привязан к событию *INSERT*. Чтобы имя и отчество не могло быть отредактировано при помощи оператора *update*, можно создать триггер, привязанный к событию *update*.

**Удаление триггеров**. Удалить существующий триггер позволяет оператор

*DROP TRIGGER trigger\_name;*

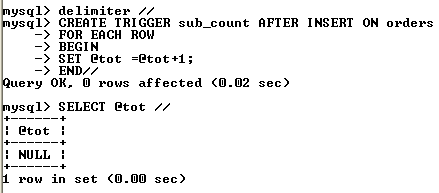
**Практическая работа**

При выполнении лабораторной работы необходимо:

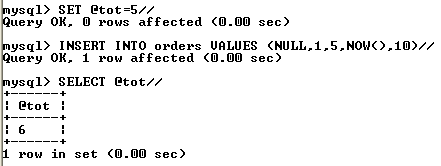
* для заданной предметной области написать два триггера для разных таблиц базы данных;
* составить отчет по лабораторной работе.

**Пример выполнения работы**

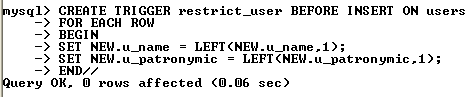
1. Создадим триггер, который при оформлении нового заказа (при добавлении новой записи в таблицу *orders*) будет увеличивать на 1 значение пользовательской пе­ременной @*tot*.



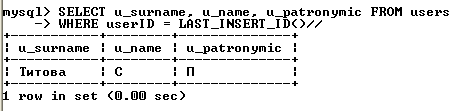
Для корректной работы триггера необходимо, чтобы пользовательская переменная @*tot* имела значение, отличное от *null*, т. к. операция сложения с *null* также приводит к *NULL*.



2. Создадим триггер, который при добавлении новых покупателей преобразует имена и отчества покупателей в инициалы.

****

****

****