Министерство образования и науки РФ

Пермский национальный исследовательский политехнический университет

Электротехнический факультет

Кафедра Информационные технологии и автоматизированные системы

Базы данных

Лабораторная работа № 7

Тема: «MySQL»

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

Изучение синтаксиса хранимых процедур в MySQL 5

Реализация хранимых процедур 6

Заключение 9

Список использованных источников 10

# Цель работы

Реализовать программу, которая демонстрирует работу хранимых процедур.

# Задачи работы

Изучить синтаксис и принцип работы хранимых процедур в MySQL

Создать процедуры на основе таблицы

# Этапы выполнения

## **Изучение синтаксиса хранимых процедур в MySQL**

Синтаксис создания представления имеет следующую форму:

DELIMITER //

CREATE PROCEDURE [IF NOT EXISTS] имя\_процедуры

([IN|OUT|INOUT] имя\_параметра тип\_параметра)

BEGIN

-- выполняемые инструкции SQL

END //

DELIMITER;

Для создания процедуры используется инструкция CREATE PROCEDURE. Операторы SQL, которые необходимо выполнить, помещаются между ключевыми словами BEGIN и END

**IF NOT EXISTS** (не обязательно) предотвращает возникновение ошибки, если существует процедура с таким же именем.

**Параметр IN** (по умолчанию) передаёт значение в процедуру.

**Параметр OUT** передаёт значение от процедуры обратно вызывающему.

**Параметр INOUT** передаёт значение в процедуру, и любые изменения, сделанные процедурой, передаются обратно вызывающему.

DELIMITER – изначально знаком разделителем была, но так как в блоке команд SQL она используется, то чтобы не прерывать тело процедуры разделитель заменяется на любой не командное сочетание символов, дальше когда тело процедуры закончится необходимо поставить символы, отвечающие за новый дециметр – это значит что одна операция, создание таблицы, выполнена. Дальше, хоть и не обязательно, можно снова вернуть разделителю его первоначальный вид – “;”

Для выполнения хранимой процедуры используется инструкция CALL:

CALL имя\_процедуры (параметры);

В инструмент для визуального проектирования баз данных — MySQL Workbench, можно просмотреть созданные процедуры, а также их код в соответствующем разделе базы данных (рис. 1).

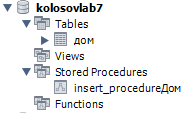


Рисунок 1 – расположение хранимых процедур в MySQL Workbench

Таким образом, с помощью рассмотренного синтаксиса можно создавать хранимые процедуры в MySQL, при этом их расположения в среде MySQL Workbench проиллюстрировано на рис. 1.

## **Реализация хранимых процедур**

Таблица, по которой будут создаваться хранимые процедуры выглядела следующим образом, (рис. 2):

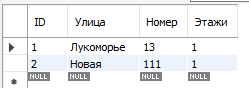


Рисунок 2 - исходная таблица

Для демонстрации работы, создадим процедуры update\_procedureДом и insert\_procedureДом. Insert\_procedureДом принимает в качестве параметров название улицы, номер дома и количество этажей в нем. Код создания процедуры приведен на (рис. 3)

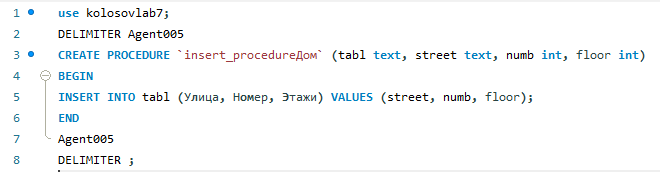


Рисунок 3 - код создания процедуры insert\_procedureДом

После вызова процедуры (рис. 4) получили таблицу с добавленной записью: id = 3, Улица = Лукоморье, Номер = 17, Этажи = 9, (рис. 5).



Рисунок 4 - вызов процедуры insert\_procedureДом

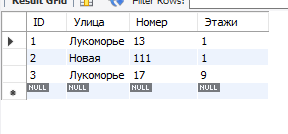


Рисунок 5 - таблица после добавления в нее записи с помощью процедуры

Процедура update\_procedureДом принимает в качестве параметров id строки, которую необходимо обновить, название улицы, номер дома и количество этажей в нем. Код создания процедуры приведен на (рис. 6)



Рисунок 6 - - код создания процедуры Update\_procedureДом

После вызова процедуры (рис. 7) получили таблицу с обновленной записью: id = 2, Улица = Лукоморье, Номер = 15, Этажи = 9, (рис. 8).



Рисунок 7 – вызов процедуры Update\_procedureДом

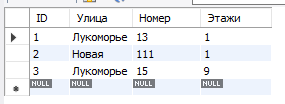


Рисунок 8 - таблица после обновления записи с помощью процедуры

Процедуры также можно использовать и для вывода данных из таблицы.

Сформируем с помощью хранимой процедуры, на основе таблицы дом, значение, соответствующее сумме всех этажей на определенной улице (рис. 9).

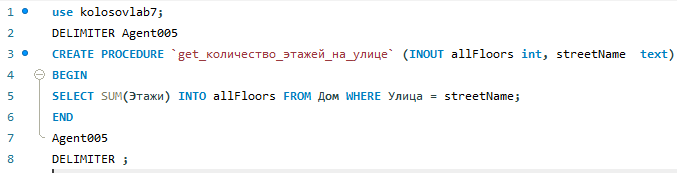


Рисунок 9 - получение количества этажей всех домов на одной улице

Для выполнения этой процедуры необходимо использовать локальную переменную. Символ @ в MySQL используется для обозначения **локальных переменных в запросе**.

С помощью командного слова SET присвоим начальное значение сумме всех этажей на определенной улице равным 0.

Дальше вызовем процедуру, передав в качестве параметра INOUT обозначенную раньше переменную и название улицы.

С помощью запроса SELECT (рис. 10) выведем полученное значение переименовав его с помощью псевдонима AS в "Количество этажей на улице Лукоморье" (рис. 11)

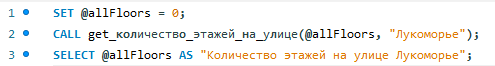


Рисунок 10 - получение и вывод значения с помощью процедуры



Рисунок 11 - вывод результата работы процедуры подсчета количества этажей

Таким образом можно использовать хранимые процедуры MySQL для уменьшения объема запросов и ускорения их написания.

# Заключение

В ходе выполнения лабораторной работы был разобран синтаксис создания хранимых процедур в MySQl

Реализованы хранимые процедуры на основе готовой таблицы, позволяющие упростить сложные операции, повысить производительность и усилить безопасность. Они обеспечивают группировку нескольких инструкций SQL в единый блок, который затем можно выполнить с помощью одной команды

# Список использованных источников

1. <https://www.mysqltutorial.org/mysql-stored-procedure/getting-started-with-mysql-stored-procedures/>
2. <https://dev.mysql.com/doc/workbench/en/wb-table-editor-triggers-tab.html>
3. <https://habr.com/ru/articles/37693/>