Московский Владислав Васильевич

**Домашняя работа №3.**

1. **Создать структуру БД Вконтакте по скриптам, приложенным в файле примеров examples.sql**

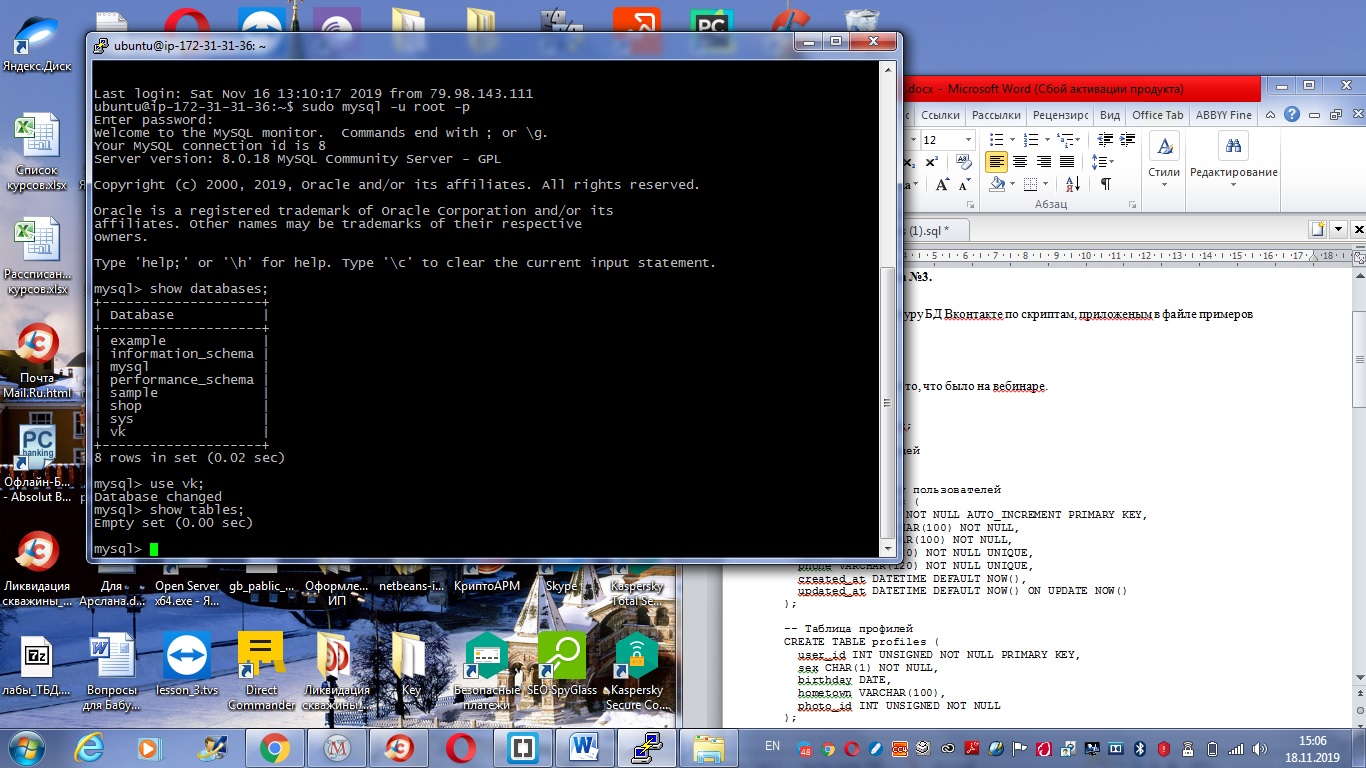
**Ответ:** *повторяем всё то, что было на вебинаре.*

Создаём БД

CREATE DATABASE vk;

-- Делаем её текущей

USE vk;



-- Создаём таблицу пользователей

CREATE TABLE users (

id INT UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY,

first\_name VARCHAR(100) NOT NULL,

last\_name VARCHAR(100) NOT NULL,

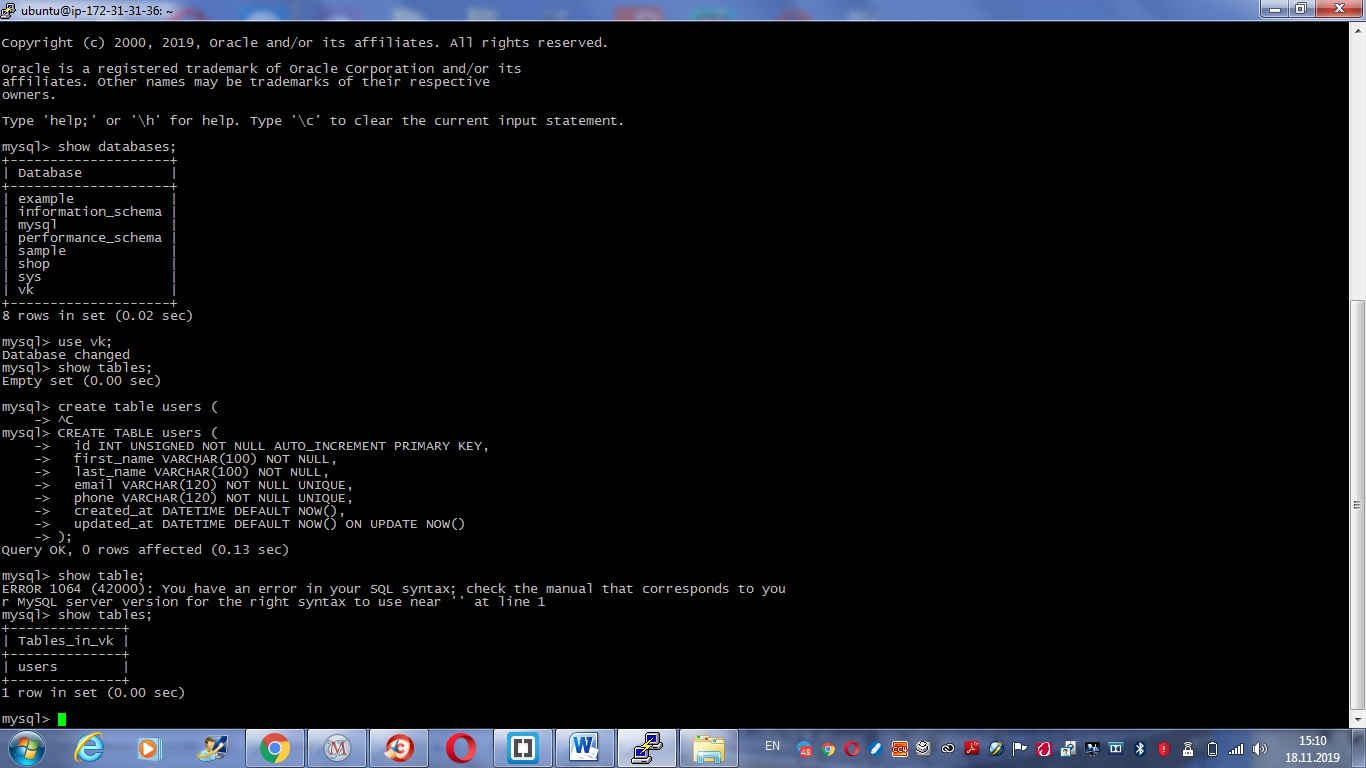
email VARCHAR(120) NOT NULL UNIQUE,

phone VARCHAR(120) NOT NULL UNIQUE,

created\_at DATETIME DEFAULT NOW(),

updated\_at DATETIME DEFAULT NOW() ON UPDATE NOW()

);



-- Таблица профилей

CREATE TABLE profiles (

user\_id INT UNSIGNED NOT NULL PRIMARY KEY,

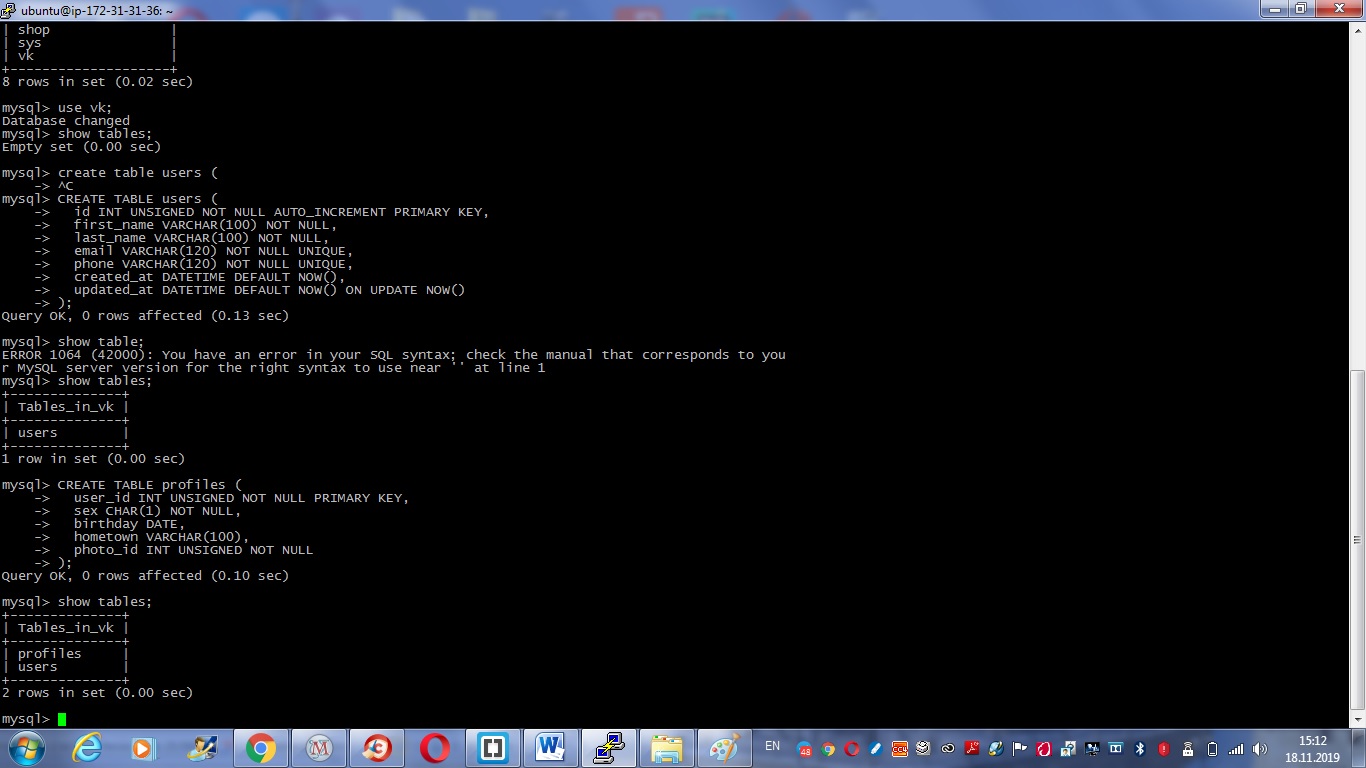
sex CHAR(1) NOT NULL,

birthday DATE,

hometown VARCHAR(100),

photo\_id INT UNSIGNED NOT NULL

);



-- Таблица сообщений

CREATE TABLE messages (

id INT UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY,

from\_user\_id INT UNSIGNED NOT NULL,

to\_user\_id INT UNSIGNED NOT NULL,

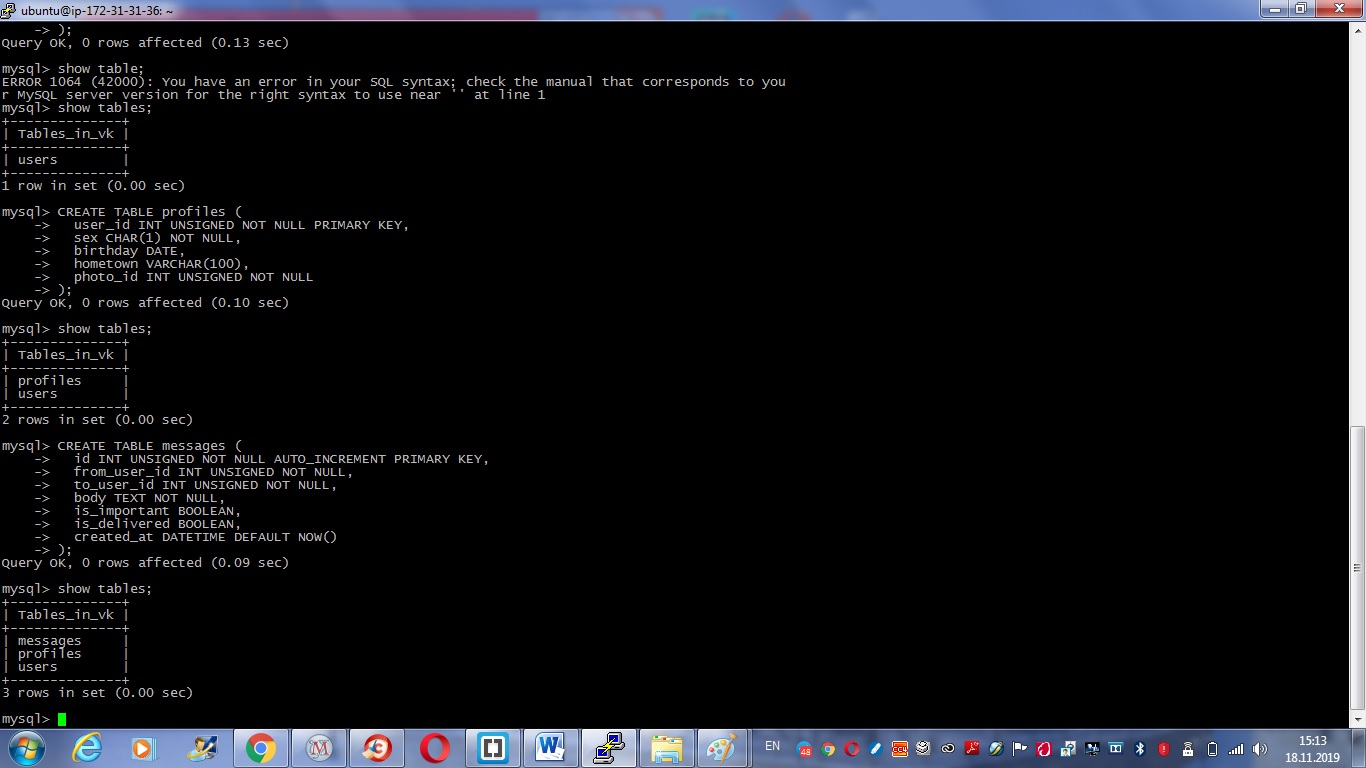
body TEXT NOT NULL,

is\_important BOOLEAN,

is\_delivered BOOLEAN,

created\_at DATETIME DEFAULT NOW()

);



-- Таблица дружбы

CREATE TABLE friendship (

user\_id INT UNSIGNED NOT NULL,

friend\_id INT UNSIGNED NOT NULL,

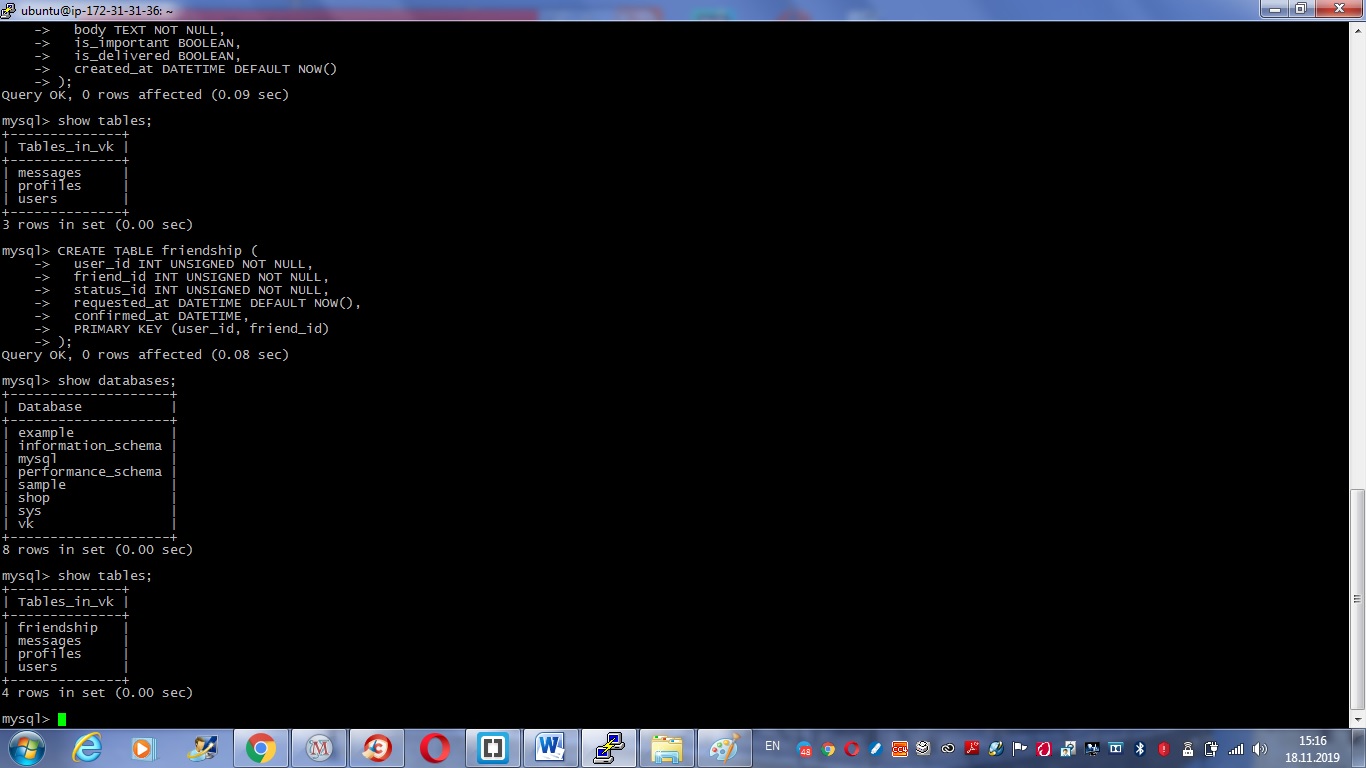
status\_id INT UNSIGNED NOT NULL,

requested\_at DATETIME DEFAULT NOW(),

confirmed\_at DATETIME,

PRIMARY KEY (user\_id, friend\_id)

);



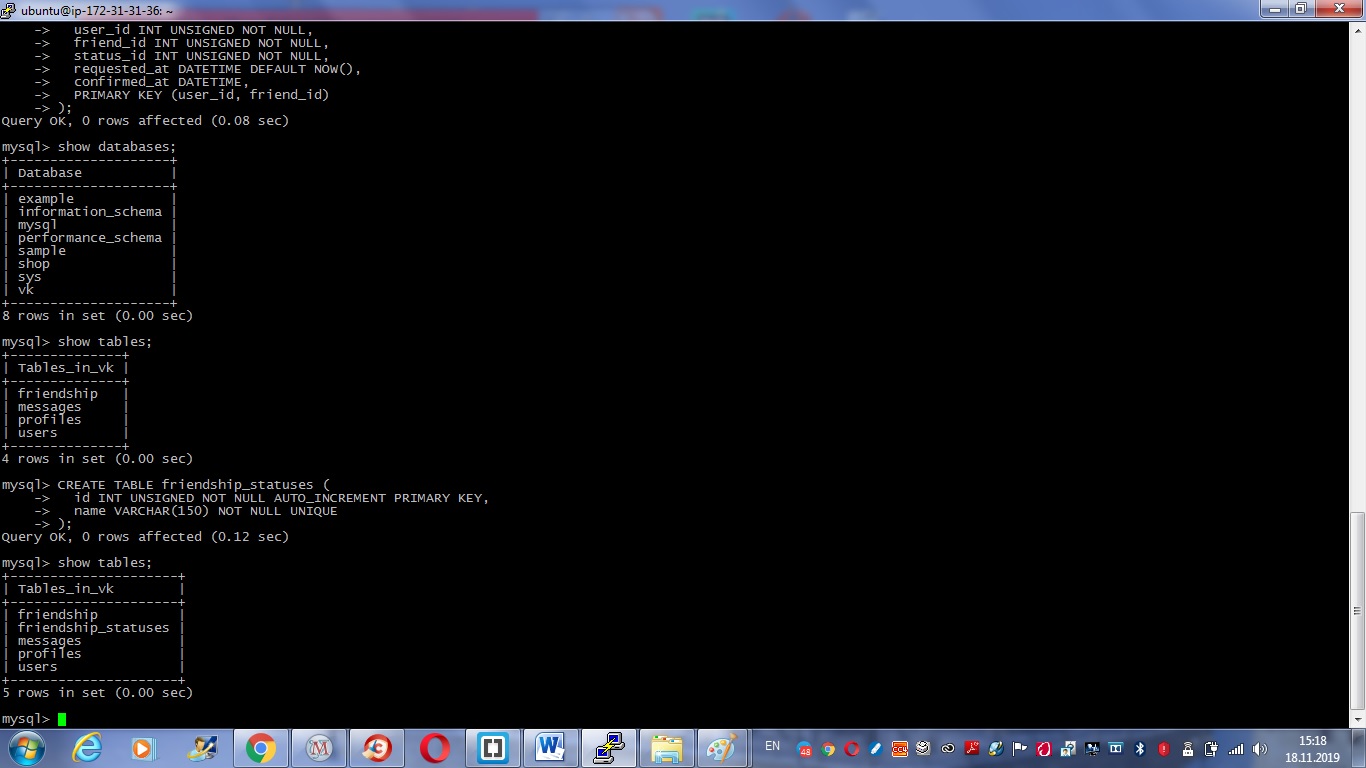
-- Таблица статусов дружеских отношений

CREATE TABLE friendship\_statuses (

id INT UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(150) NOT NULL UNIQUE

);



-- Таблица групп

CREATE TABLE communities (

id INT UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(150) NOT NULL UNIQUE

);

-- Таблица связи пользователей и групп

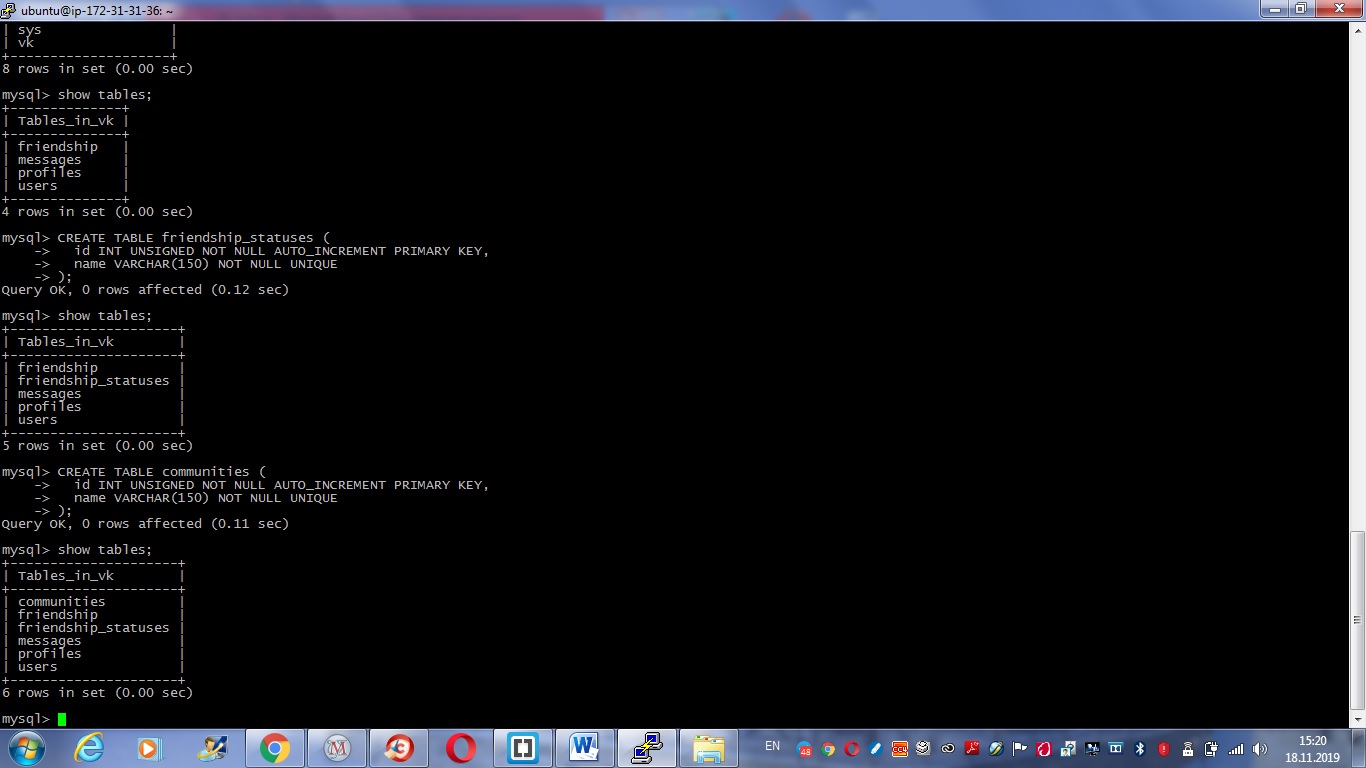
CREATE TABLE communities\_users (

community\_id INT UNSIGNED NOT NULL,

user\_id INT UNSIGNED NOT NULL,

PRIMARY KEY (community\_id, user\_id)

);



-- Таблица медиафайлов

CREATE TABLE media (

id INT UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY,

media\_type\_id INT UNSIGNED NOT NULL,

user\_id INT UNSIGNED NOT NULL,

filename VARCHAR(255) NOT NULL,

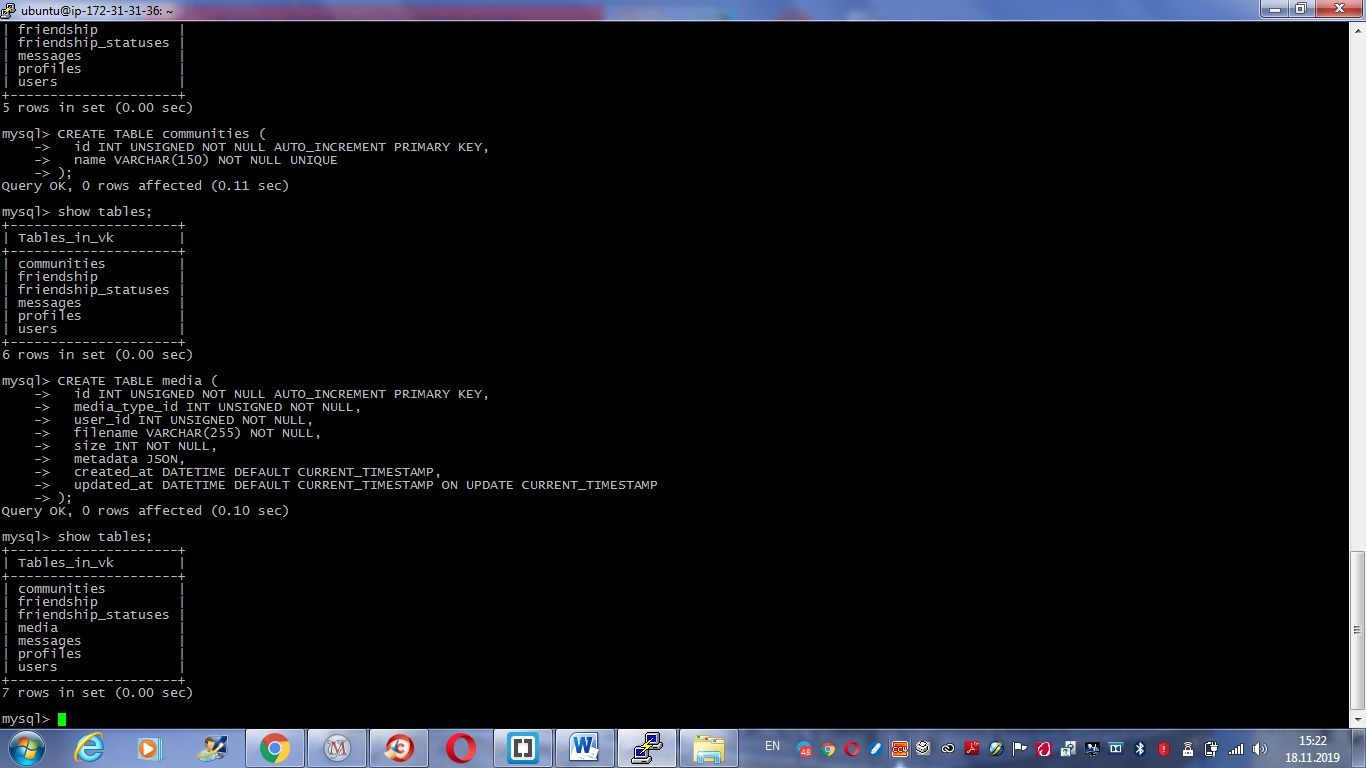
size INT NOT NULL,

metadata JSON,

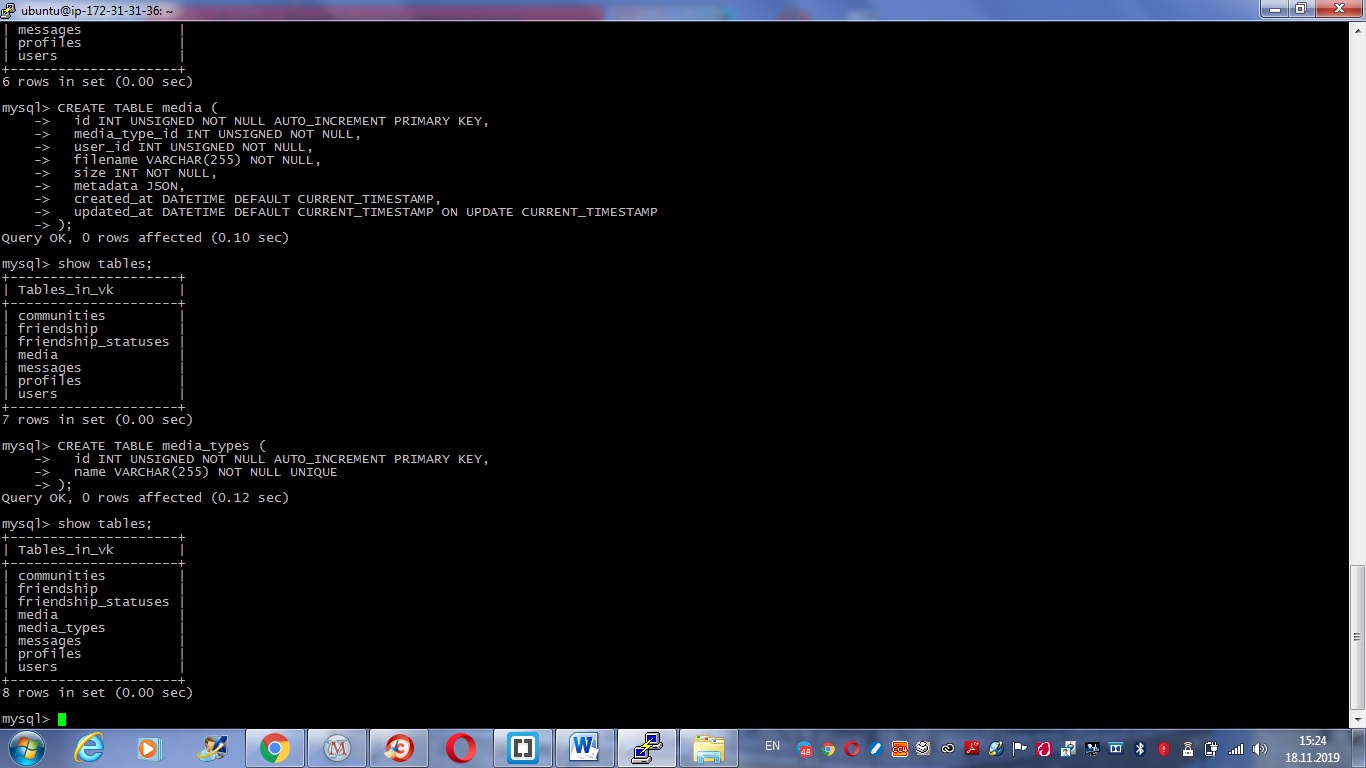
created\_at DATETIME DEFAULT CURRENT\_TIMESTAMP,

updated\_at DATETIME DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP

);

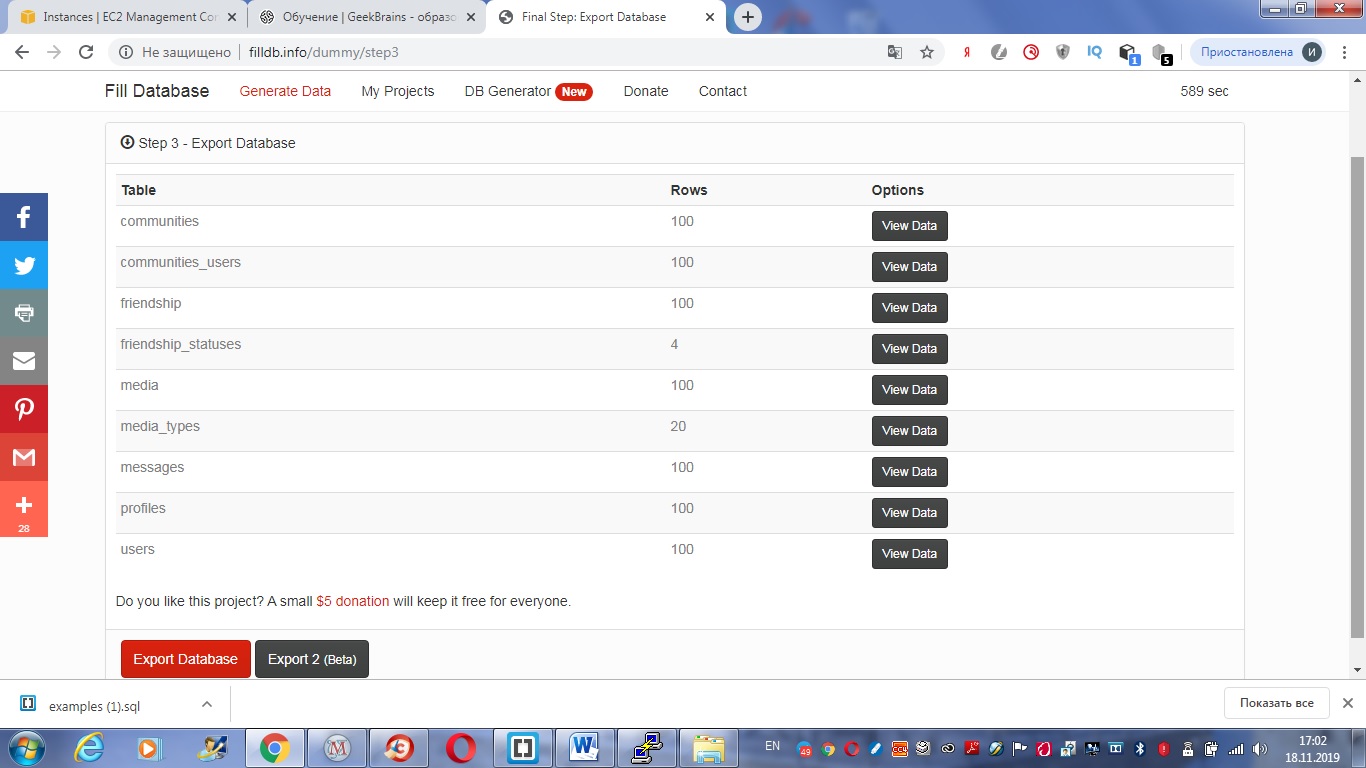


-- Таблица типов медиафайлов



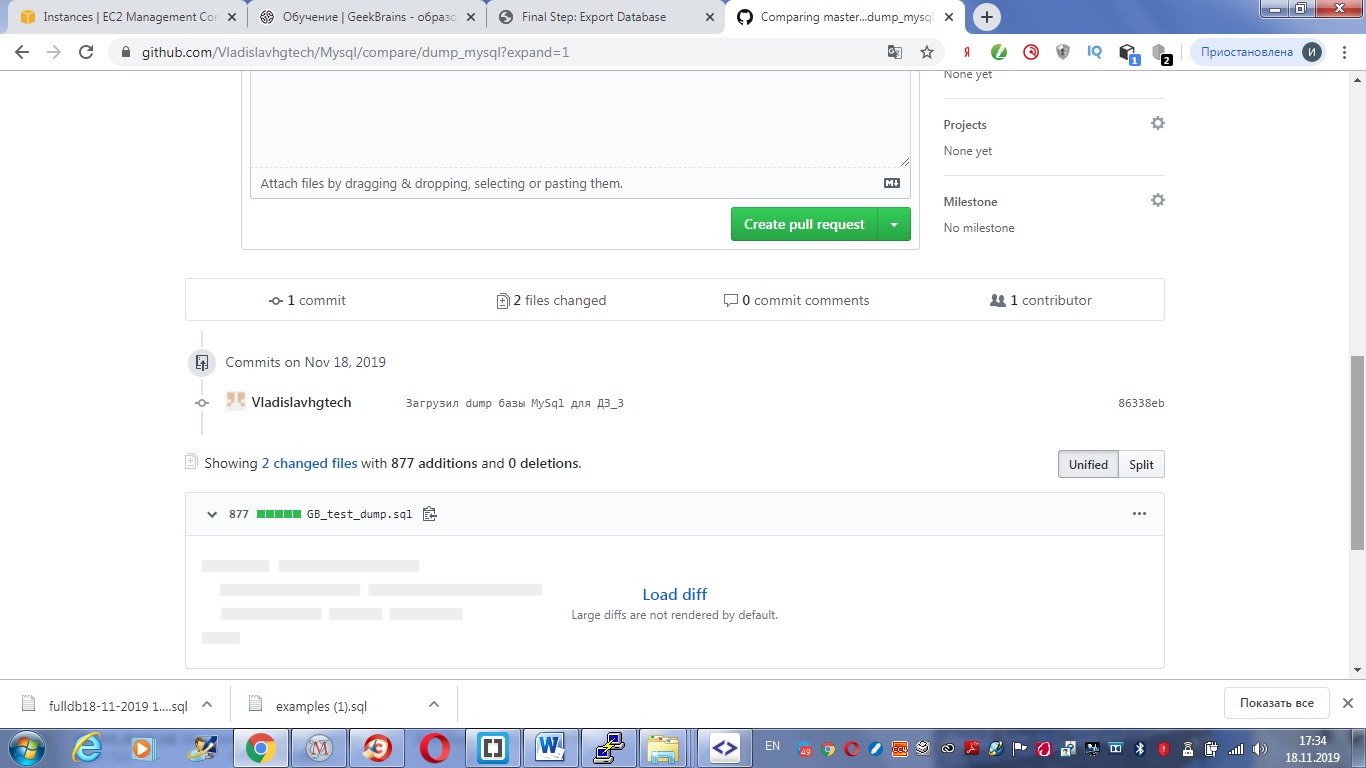
1. **Используя сервис http://filldb.info или другой по вашему желанию, сгенерировать тестовые данные для всех таблиц, учитывая логику связей. Для всех таблиц, где это имеет смысл, создать не менее 100 строк. Загрузить тестовые данные.**

Ответ: загрузка тестовых данных производилась с помощью <http://filldb.info>.



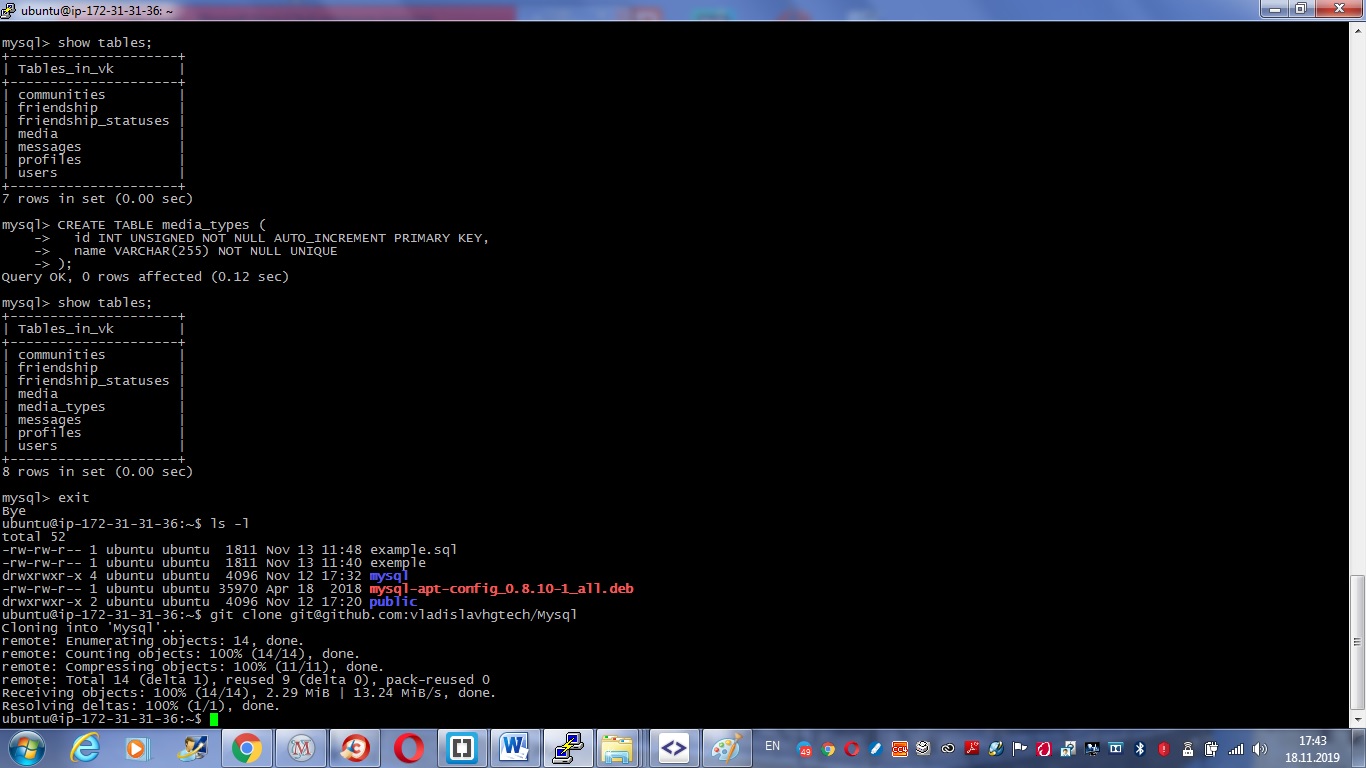
Поскольку моя СУБД размещена на instance AWS, необходимо передать туда dump. Для этого файл с базой данных отправим на GitHub. Копируем файл с dump в папку GitHub. В SmartGit добавляем Branch (add Branch). Нажимаем Commit и Push

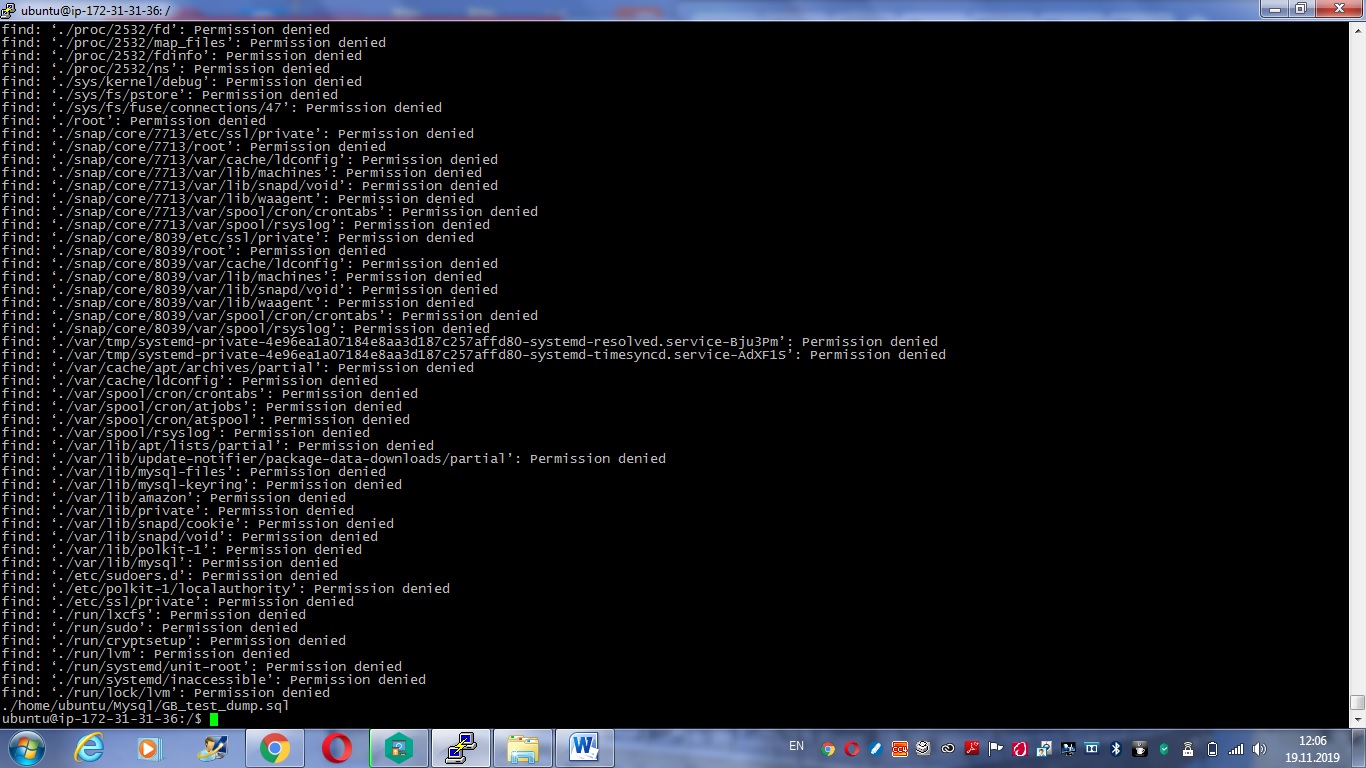
Проверяем наличие файла на GitHub



Файл GB\_test\_dump.sql появился.

Клонируем репозиторий на удаленный Instance командой **git clone git@github.com:vladislav hgtech/Mysql**

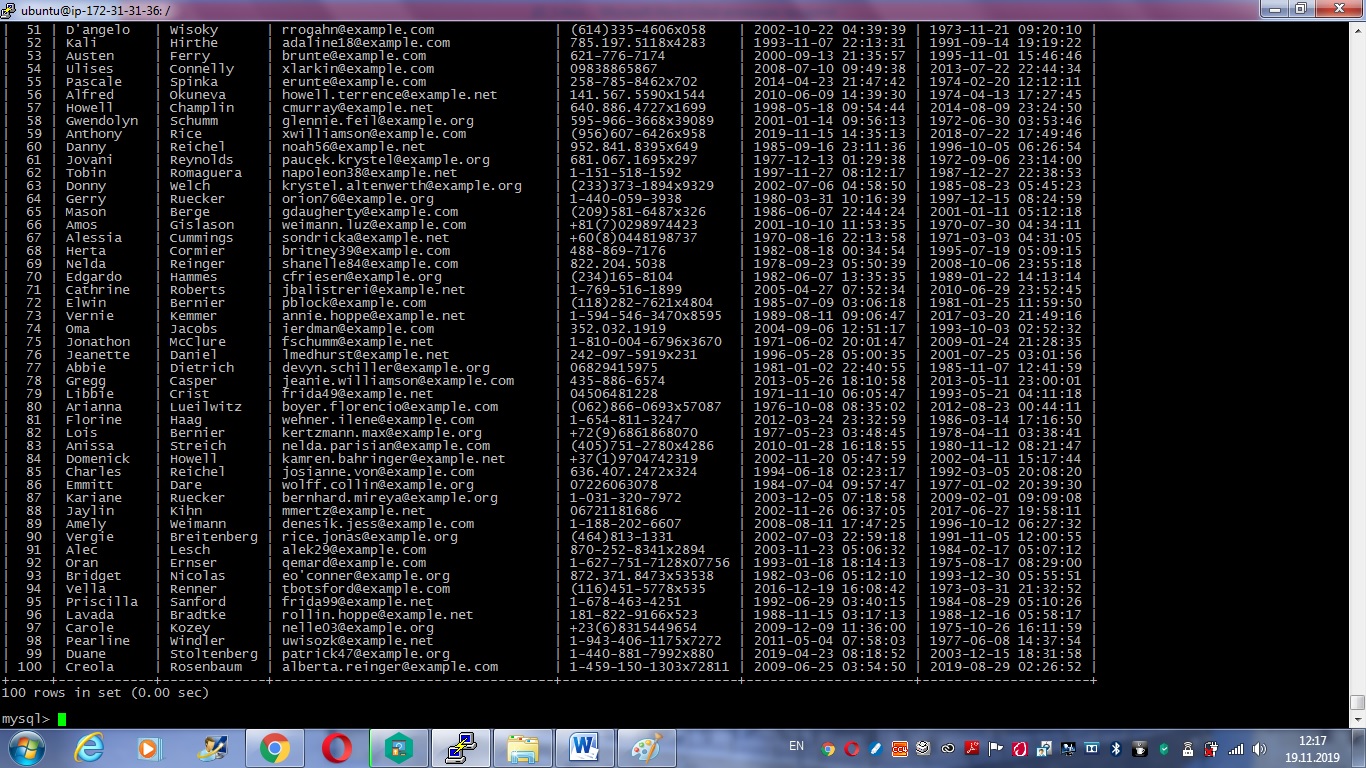
****

Переходим в корневой каталог и запускаем поиск dump по имени командой **find –name GB\_test\_dump.sq**

Разворачиваем dump командой **sudo mysql -u root -p vk < ./home/ubuntu/Mysql/GB\_test\_dump.sql**

Выбираем базу данных **use vk;**

Проверяем наличие данных в таблице USERS командой **select \* from users;**

****

Все данные перенеслись.

1. (по желанию) Проанализировать структуру БД vk, которую мы создали на занятии, и внести предложения по усовершенствованию (если такие идеи есть). Напишите пожалуйста, всё-ли понятно по структуре.

**Ответ:** не выполнено, потому что ничего не понял.

*При выполнении ДЗ\_3 у меня возникло несколько вопросов:*

1. *Я переносил файл dump через GitHub. Насколько я понимаю это не самый лучший вариант обмениваться данными с удаленным сервером. Какие наиболее эффективные и универсальные способы обмена файлами с instans?*
2. *Третье задание я не сделал, поскольку не до конца понял теоретическую часть. А именно вообще плохо понял:*

* *Как таблицы взаимодействую друг с другом?*
* *Что такое ключи и почему их может быть много?*
* *Насколько я понял, с точки зрения математики база данных позволяет найти какой-то элемент исходя из заданных условий. Грубо говоря, если выполняется одно условие и второе условие (например совпадают ключи), то выдается список элементов, для которых это выполняется.*
* *Получается следующее: база данных – это множество каких-то элементов (всё, что угодно машины, двери, продукты, слова, числа и т.д.). Каждый из этих элементов обладает какими-то свойствами – цвет, стоимость, мощность и т.д. Над этим множеством определены какие-то действия (сложение, умножение, объединение, вычитание и т.д). Количество действий ограничено. Вопрос: имея множество, свойства элементов и действия над ними что мы хотим получить? Какую задачу мы решаем? Т.е. мы находим свойства объекта или наоборот по свойствам мы ищем сам объект? Или и то и другое? Не понятно.*

1. *Проектирование баз данных вообще не понятно. Не хватает практических задач для самостоятельного решения. 2-3 небольшие задачи исправили бы такое положение.*

*По проектированию непонятно следующее. Насколько я понял основная цель проектирования баз данных – это придание какому-то бесструктурному множеству какую-то упорядоченную структуру. Откуда взялись эти нормальные формы?*

*На самом деле понятно, что эта модель представляет собой набор взаимодействующих таблиц. Не понятно как эти таблицы между собой взаимодействуют. Нужны рисунки со стрелочками, принципиальная схема. И как в простейшей модели базы данных обрабатываются запросы. Хорошо было бы показать как проходит запрос информации. Тогда стало бы ясно что-такое ключи внешние, внутренние и зачем они нужны*.