Московский Владислав Васильевич

**Домашняя работа №4**

**Практическое задание по теме “CRUD - операции”**

**1. Повторить все действия по доработке БД vk.**

**2. Заполнить новые таблицы.**

**3. Подобрать сервис который будет служить основой для вашей курсовой работы.**

**4. (по желанию) Предложить свою реализацию лайков.**

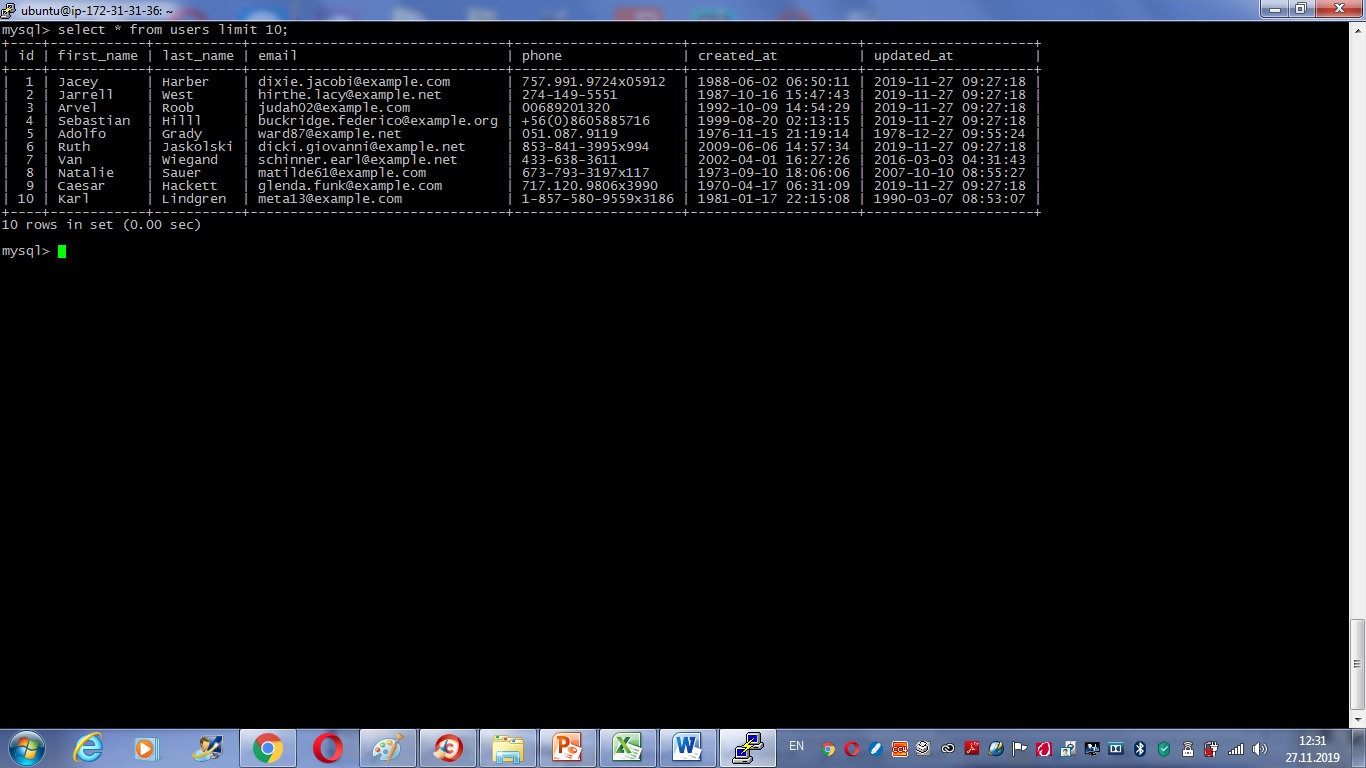
**Решение:**

1. Дорабатываю таблицу Users. Двойным запросом:

**update users set created\_at= updated\_at where created\_at>updated\_at;**

**update users set updated\_at=created\_at where created\_at>updated\_at;**

Результат ниже.



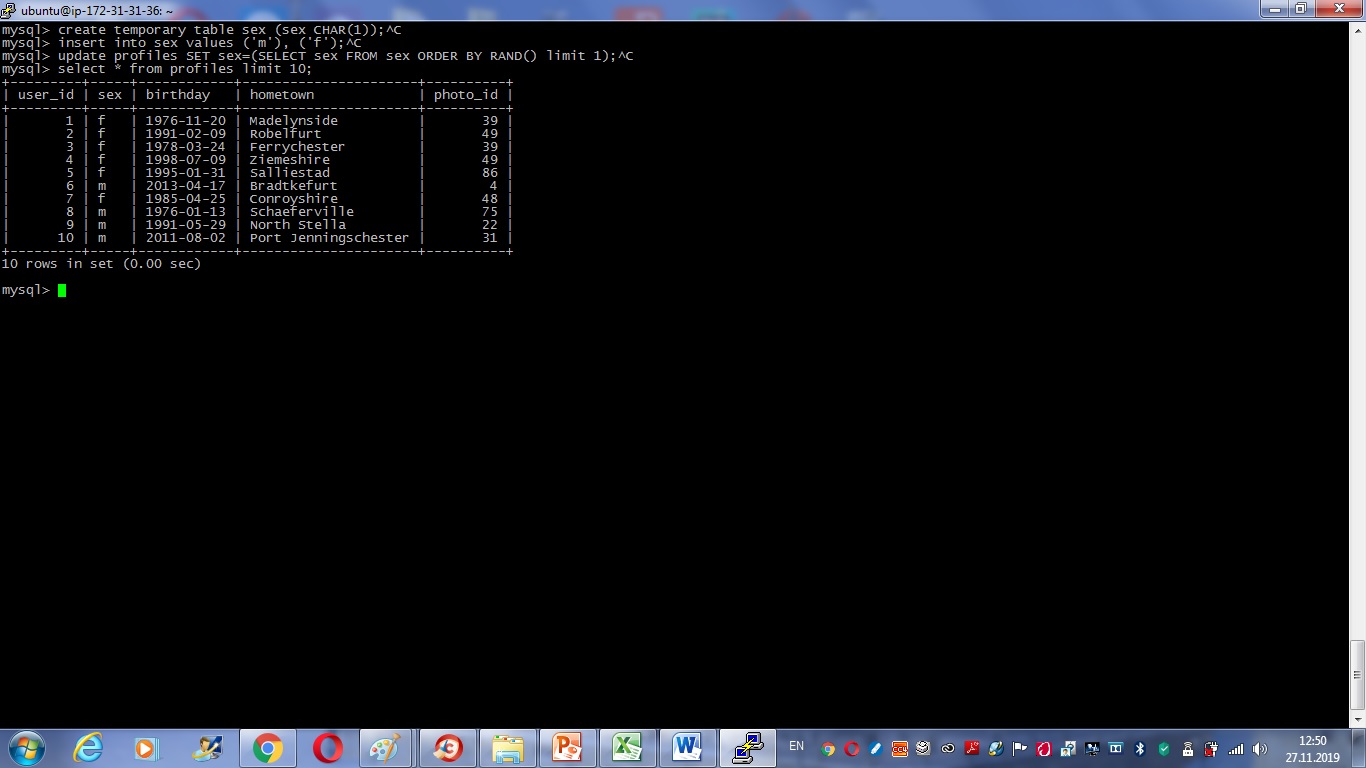
Модернизируем таблицу profiles. В данной таблице необходимо поменять значения колонки ПОЛ на”f” или “m” в случайном порядке. Для этого создаем временную таблицу SEX командой:

**create temporary table sex (sex CHAR(1));**

Добавляем туда значения ‘f’ и ‘m’ командой**: insert into sex values ('m'), ('f');**

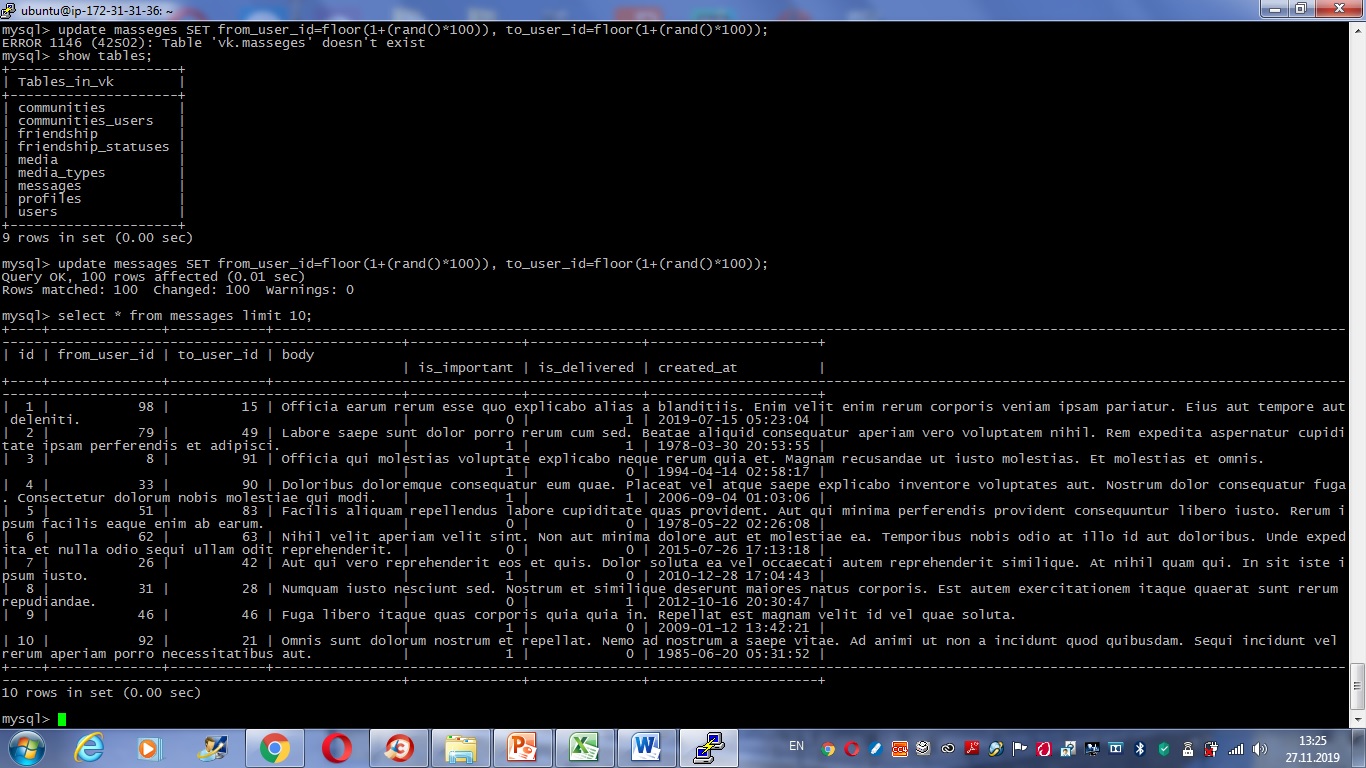
Модернизируем таблицу profiles командой:

**update profiles SET sex=(SELECT sex FROM sex ORDER BY RAND() limit 1);**

****

Модернизируем таблицу messages командой:

**update messages SET from\_user\_id=floor(1+(rand()\*100)), to\_user\_id=floor(1+(rand()\*100));**

****

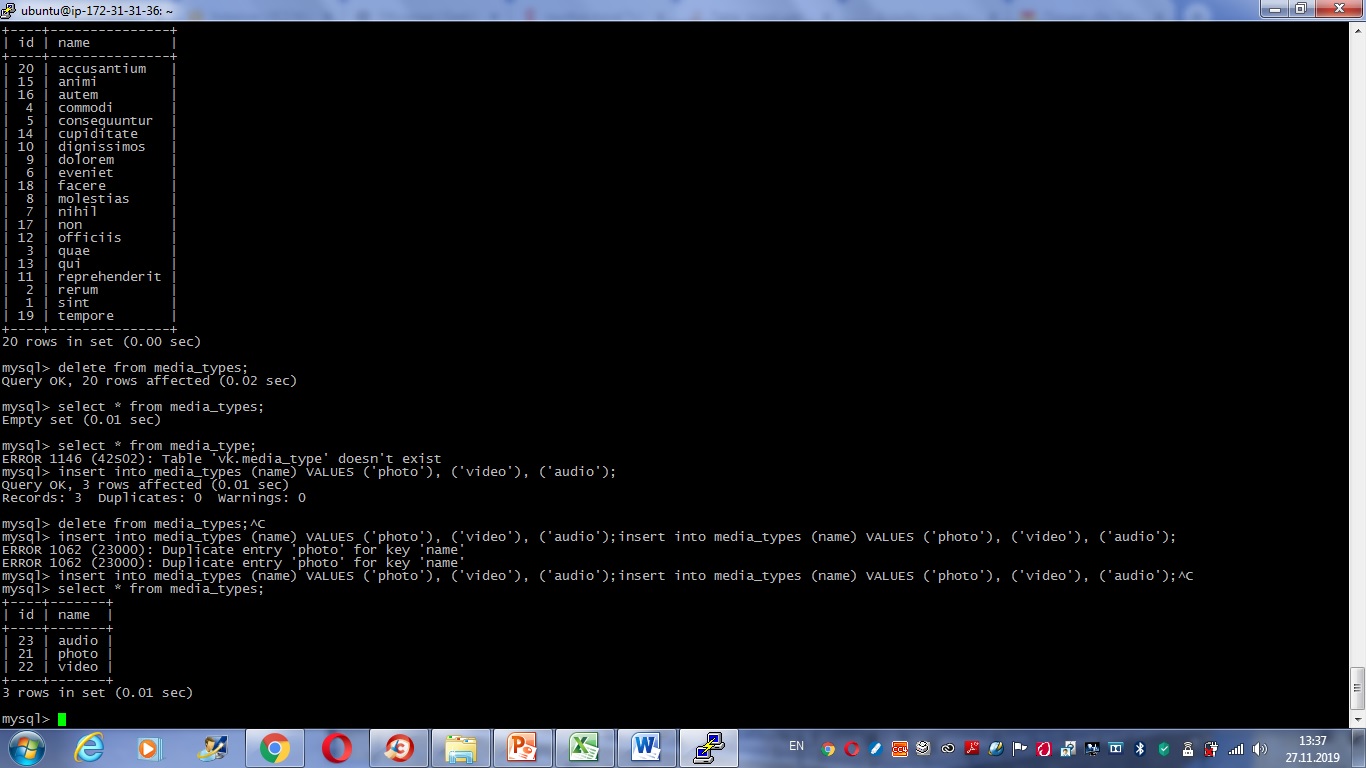
Очищаем таблицу media\_types командой**:**

**truncate from media\_types;**

Вставляем нужные значения командой:

**insert into media\_types (name) VALUES**

**('photo'), ('video'), ('audio');**

****

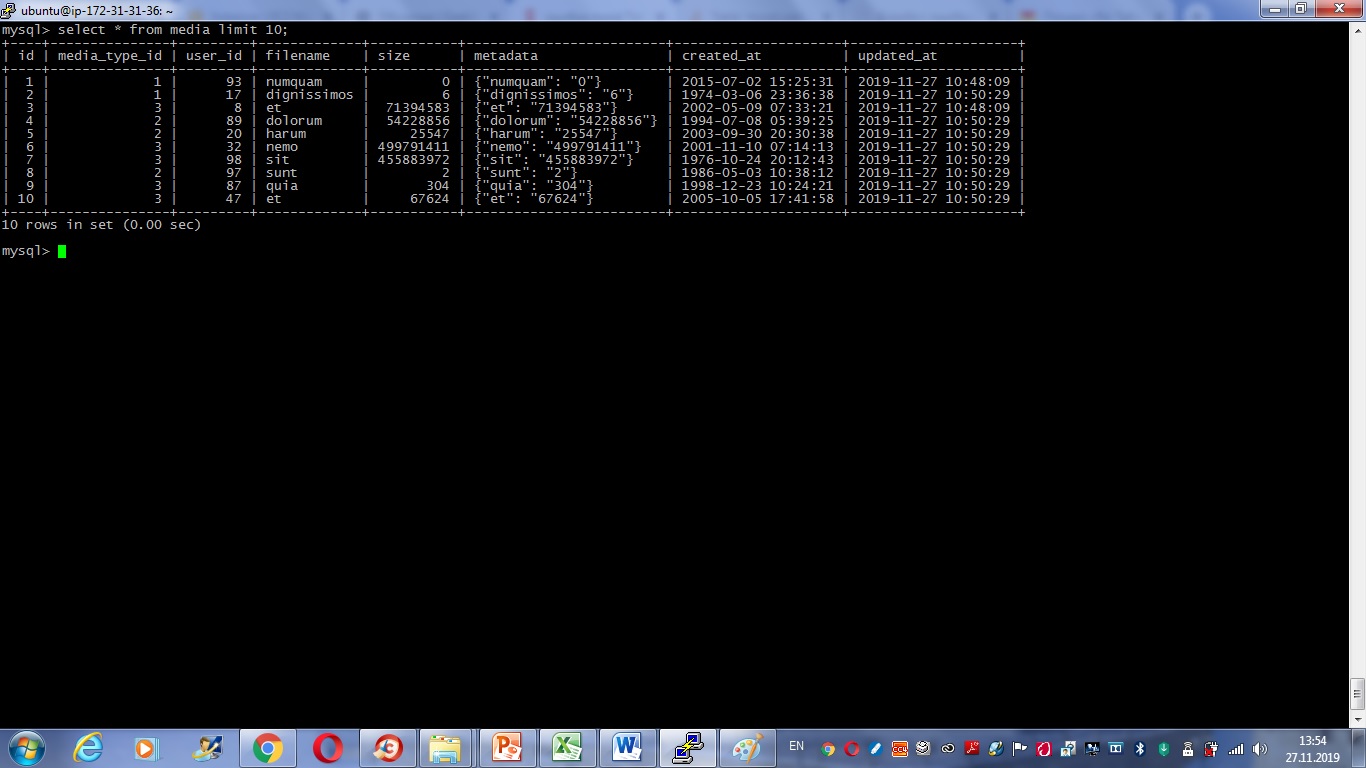
Корректируем таблицу media.

SELECT \* FROM media LIMIT 10;

UPDATE media SET user\_id = FLOOR(1 + (RAND() \* 100));

UPDATE media SET metadata = CONCAT('{"', filename, '":"', size, '"}');

ALTER TABLE media MODIFY COLUMN metadata JSON;

****

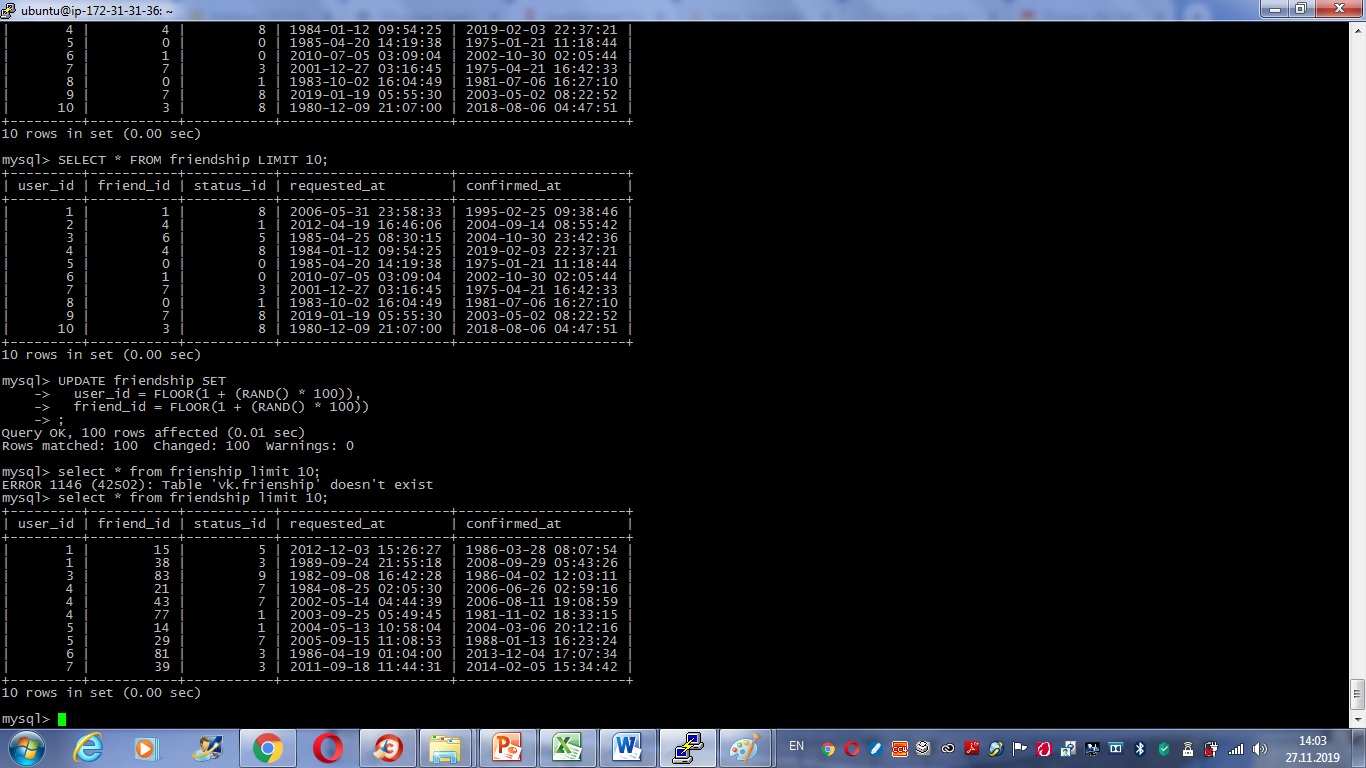
**Обновляем таблицу friendship:**

SELECT \* FROM friendship LIMIT 10;

UPDATE friendship SET

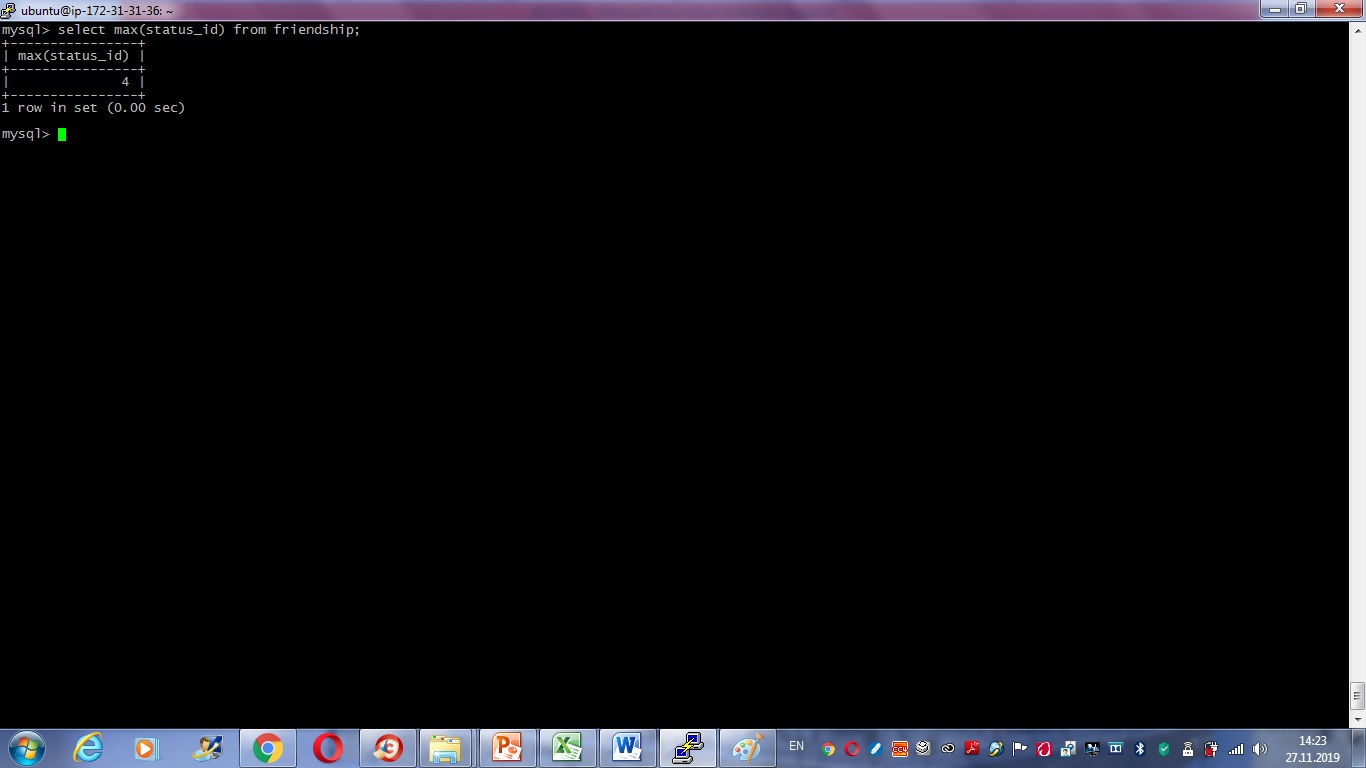
user\_id = FLOOR(1 + (RAND() \* 100)),

friend\_id = FLOOR(1 + (RAND() \* 100));



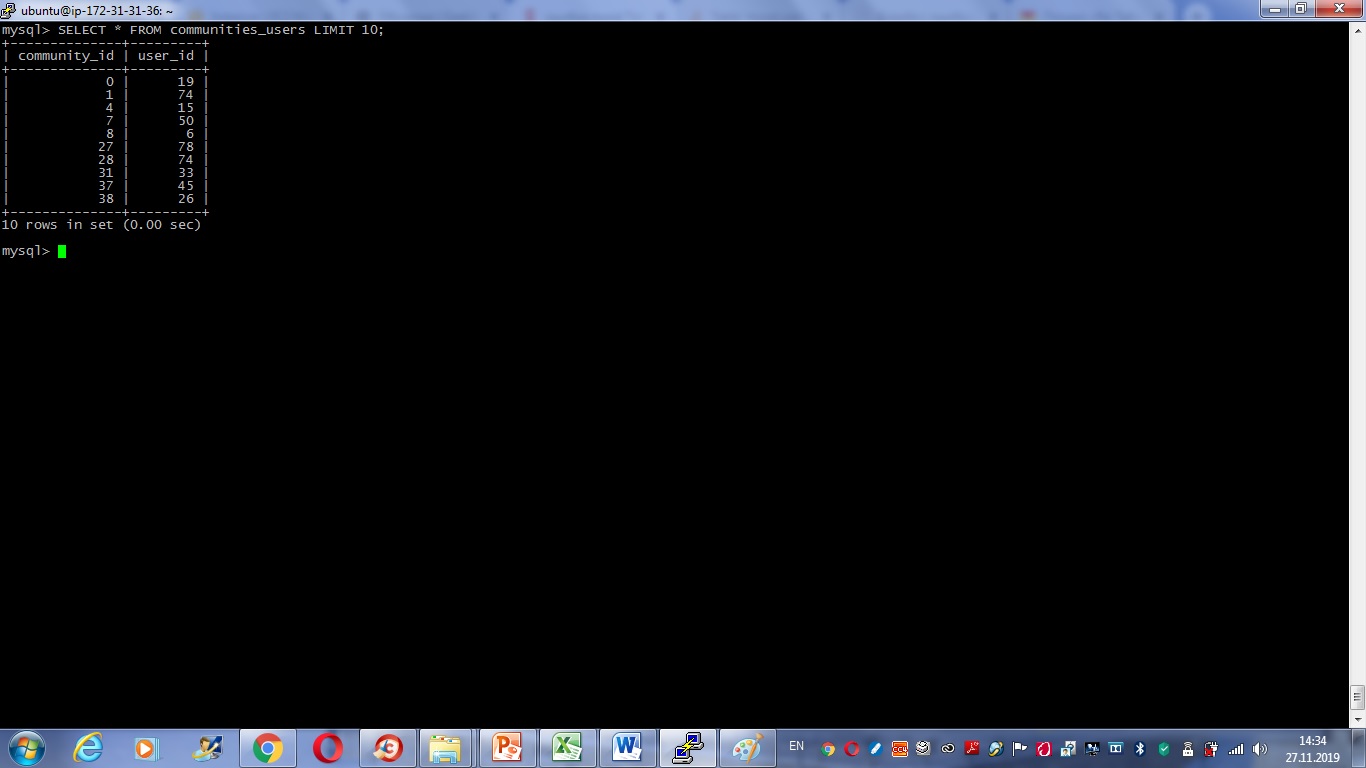
Проверяем максимальное значение status\_id в таблице friendship. Максимальное значение равно 9. Поскольку статусов всего 4, необходимо внести корректировки в таблицу friendship командой:

**update friendship SET status\_id=floor(1+(rand()\*4));**

****

Корректируем таблицу communities\_users:

**UPDATE communities\_users SET user\_id = FLOOR(1 + (RAND() \* 100));**

****

1. **Дорабатываем базу данных. Вносим новые таблицы:**

**CREATE TABLE user\_privacy (**

**id INT UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY,**

**user\_id INT UNSIGNED NOT NULL,**

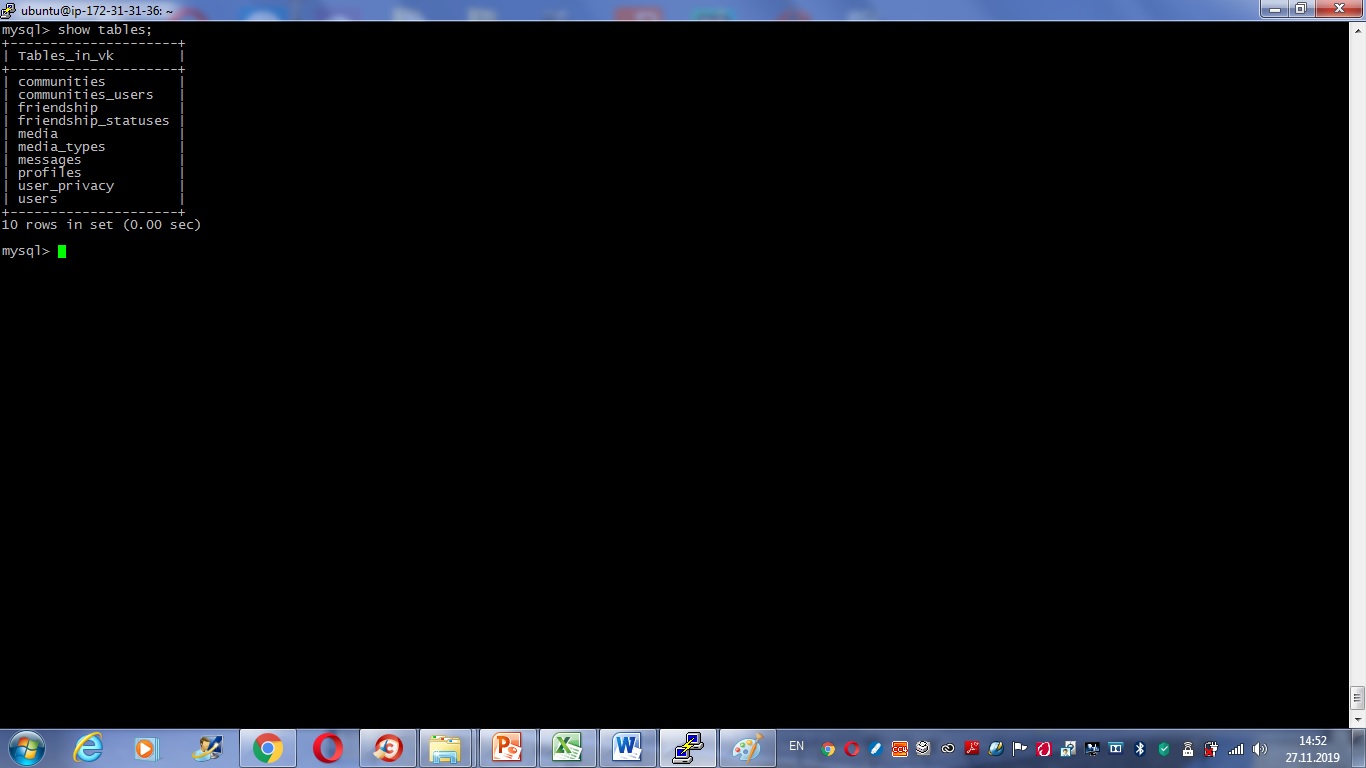
**section\_id INT UNSIGNED NOT NULL,**

**privacy\_id INT UNSIGNED NOT NULL,**

**created\_at DATETIME DEFAULT NOW(),**

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

**);**

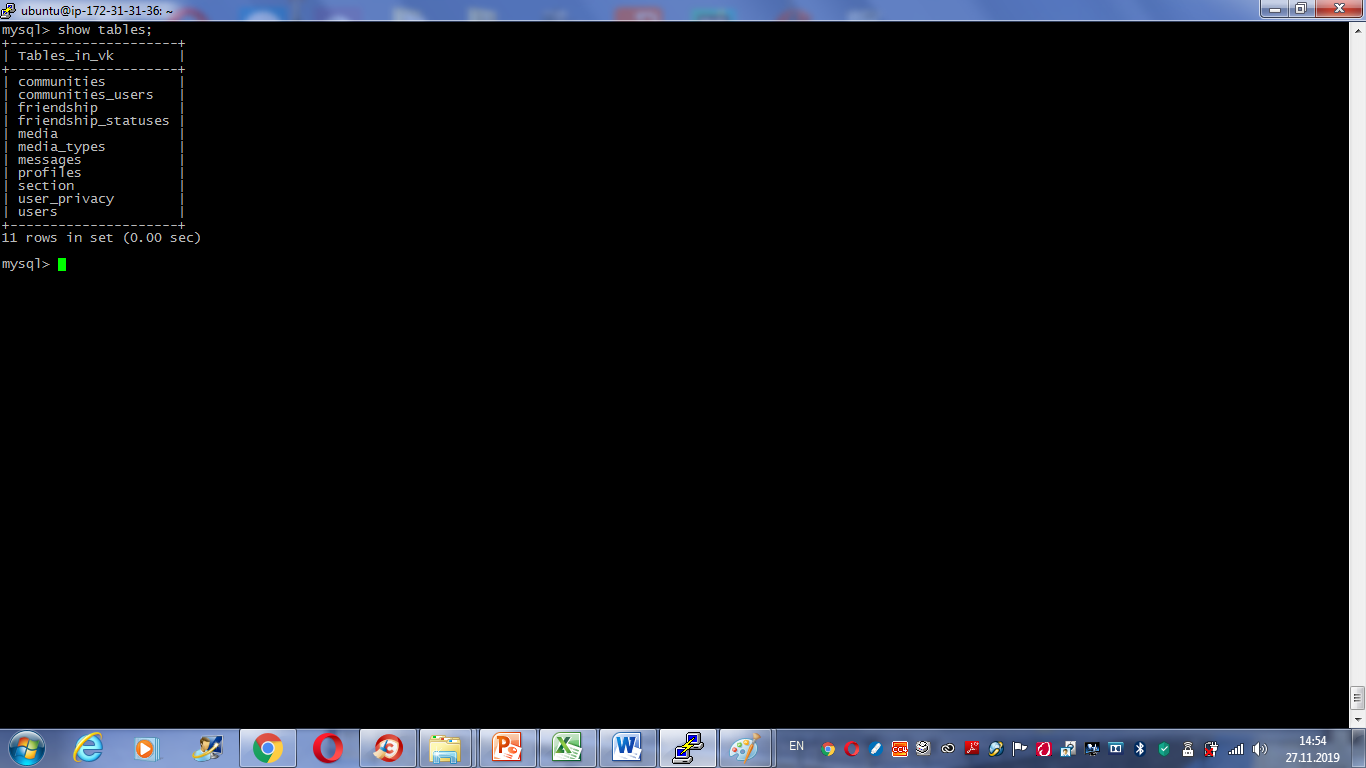
****

CREATE TABLE section (

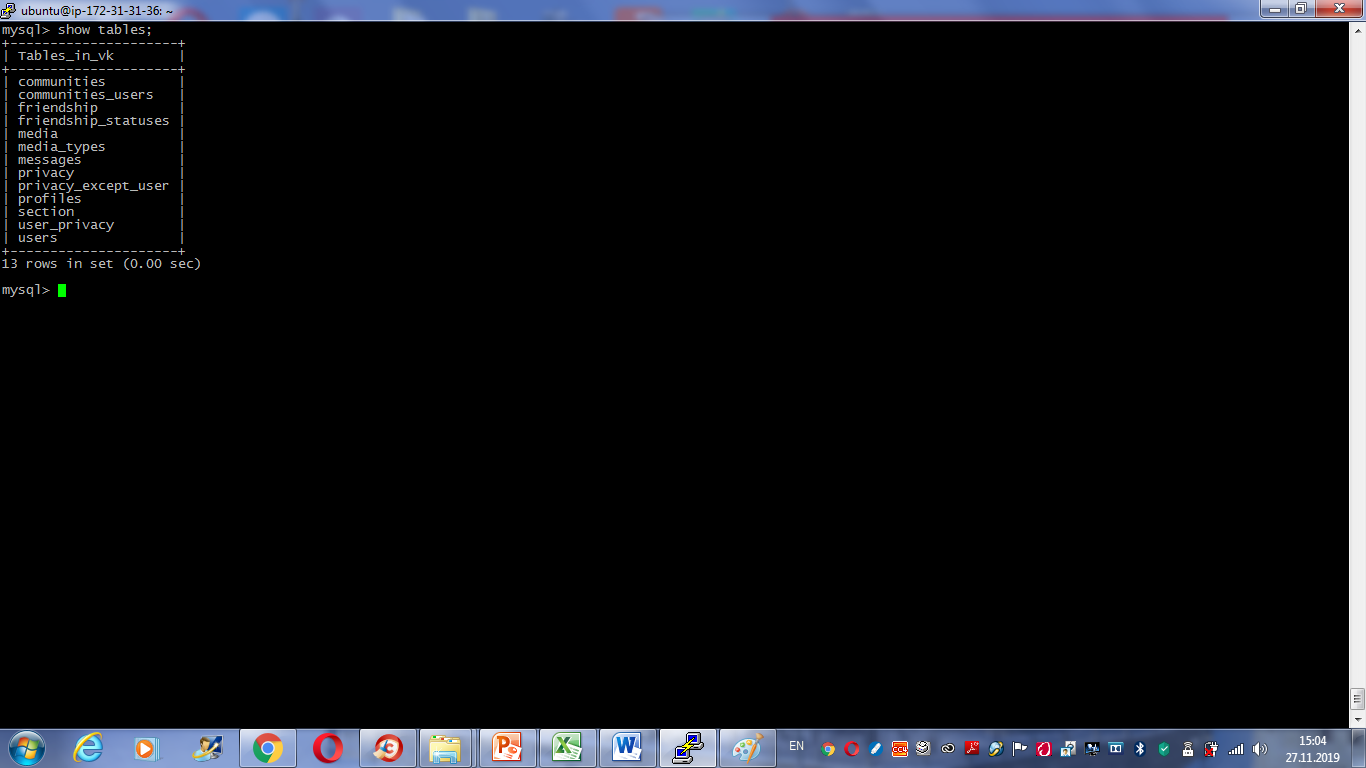
id INT UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY,

name VARCHAR(255) NOT NULL

);



Добавляем и заполняем новые таблицы командами из файла example.sql. Все таблицы созданы.



1. Подобрать сервис, который будет служить основой для вашей курсовой работы.

Ответ: я буду пытаться реализовать базу данных delimobil.ru

1. Предложить свою реализацию лайков.

Решение:

В каждую таблицу, которую мы планируем “лайкать” добавляем 2 колонки «like», «unlike» одна будет отвечать за положительные лайки, вторая за отрицательные. Обе колонки должны иметь целочисленные параметры INT. При нажатии кнопки “like”, в колонке будет увеличиваться счетчик на 1.

Реализация на примере таблицы MEDIA:

ALTER TABLE media ADD COLUMN likes INT after updated\_at;

ALTER TABLE media ADD COLUMN unlikes INT after updated\_at;

Также пытался сделать “лайки” на основе создания таблицы с о статусами “лайков”. Создал только таблицу статусов. Как её применить не придумал.

CREATE TABLE likes ( likes\_id INT UNSIGNED NOT NULL AUTO\_INCREMENT PRIMARY KEY, status\_like VARCHAR(255) NOT NULL );

INSERT INTO likes (likes\_id, status\_like) VALUES (1, 'Нравится');

INSERT INTO likes (likes\_id, status\_like) VALUES (2, 'Не нравится');