

Практическое задание по теме "CRUD - операции"
1. Повторить все действия по доработке структуры БД vk и данных.

Таблица users

столбцы created_at & updated_at

-- проверил даты на наличие несоответствий:

```
SELECT * FROM users WHERE updated_at < created_at;
```

-- исправил даты при помощи UPDATE:

```
UPDATE users SET updated_at = NOW() WHERE updated_at < created_at;
```

-- проверил, что исправлено:

```
SELECT * FROM users WHERE updated_at < created_at;
```

столбец phone

формат номеров телефона в некоторых строках не понравился, но трогать не стал

Таблица profiles

столбцы created_at & updated_at

аналогично таблице users

столбец countries_id

-- сделал вразброс нумерацию:

```
UPDATE profiles SET city_id = FLOOR(1 + RAND() * 100);
```

столбец birthday

-- исправлял даты рождения, которые раньше даты created_at:

```
UPDATE profiles SET birthday = MAKEDATE(2001, 92) WHERE user_id = 30;
```

```
UPDATE profiles SET birthday = MAKEDATE(2002, 16) WHERE user_id = 71;
```

```
UPDATE profiles SET birthday = MAKEDATE(2001, 116) WHERE user_id = 100;
```

столбец gender

не трогал, т. к. там F & M уже вразброс

Таблица cities

столбец countries_id

-- сделал вразброс нумерацию:

```
UPDATE cities SET country_id = FLOOR(1 + RAND() * 100);
```

Таблица messages

столбцы from_user_id и to_user_id

-- сделал вразброс нумерацию:

```
UPDATE messages SET from_user_id = FLOOR(1 + RAND() * 100);
```

```
UPDATE messages SET to_user_id = FLOOR(1 + RAND() * 100);
```

столбец is_important

-- сделал вразброс атрибуты „usual“ и „important“:

```
DELETE FROM messages WHERE is_important;
```

```
CREATE TEMPORARY TABLE importants (name VARCHAR(50));
INSERT INTO importants VALUES
  ('usual'),
  ('important');
SELECT * FROM importants;
ALTER TABLE messages MODIFY COLUMN is_important ENUM('usual', 'important');
UPDATE messages SET is_important = (SELECT name FROM importants ORDER BY RAND()
LIMIT 1);
```

столбец is_delivered

аналогично is_important поменял атрибуты „delivered“ и „undelivered“

Таблица communities

столбцы created at & updated at

-- исправил даты аналогично таблицы users:

-- обрезал количество communities до 30:

```
DELETE FROM communities WHERE id > 30;
```

столбец name

-- вручную изменил наименование сообществ, первые 10 строк, на остальное хватило терпения, наверняка потом будем изучать как это все делается.

```
UPDATE communities SET name = 'РИА Новости' WHERE id = 1;
```

...

```
UPDATE communities SET name = 'Музыка ВКонтакте' WHERE id = 10;
```

Таблица communities users

столбец users id

-- не трогал, т. к. там уже вразброс

столбец communities id

-- сделал вразброс нумерацию до 30, как количество сообществ:

```
UPDATE communities_users SET community_id = FLOOR(1 + RAND() * 30);
```

странный разброс получается, не совсем рандомный, не понятно почему.

| | community_id | user_id | created_at |
|----|--------------|---------|---------------------|
| 1 | 1 | 48 | 2020-08-05 14:40:23 |
| 2 | 1 | 90 | 2020-05-19 12:13:38 |
| 3 | 2 | 35 | 2020-11-22 08:02:48 |
| 4 | 2 | 90 | 2020-06-04 18:24:52 |
| 5 | 3 | 6 | 2021-04-07 02:26:25 |
| 6 | 3 | 34 | 2020-05-03 00:09:45 |
| 7 | 3 | 64 | 2021-01-08 23:45:53 |
| 8 | 3 | 83 | 2020-10-17 22:00:30 |
| 9 | 4 | 6 | 2020-05-31 10:50:25 |
| 10 | 4 | 13 | 2020-06-28 10:59:09 |
| 11 | 4 | 20 | 2021-03-05 13:34:05 |
| 12 | 4 | 46 | 2021-03-27 22:02:54 |
| 13 | 4 | 84 | 2021-03-08 23:58:34 |
| 14 | 5 | 26 | 2020-06-05 18:26:08 |
| 15 | 5 | 34 | 2020-12-19 05:44:48 |
| 16 | 6 | 17 | 2020-05-13 04:51:30 |
| 17 | 6 | 46 | 2020-06-25 17:01:15 |
| 18 | 6 | 57 | 2021-04-03 03:49:52 |
| 19 | 6 | 76 | 2020-12-30 13:26:11 |
| 20 | 7 | 89 | 2020-09-06 07:09:45 |
| 21 | 7 | 92 | 2020-11-23 13:14:26 |

Таблица friendship_statuses

```
-- скорректировал таблицу:  
TRUNCATE friendship_statuses;  
  
INSERT INTO friendship_statuses (name) VALUES  
  ('Requested'),  
  ('Confirmed'),  
  ('Rejected');
```

Таблица friendship

```
-- Обновил ссылки на статус  
UPDATE friendship SET friendship_status_id = FLOOR(1 + RAND() * 3);  
  
столбец user_id  
-- сделал вразброс нумерацию:  
UPDATE friendship SET user_id = FLOOR(RAND() * 100) + 1;  
  
столбцы user_id и friend_id  
-- проверил наличие совпадений user_id = friend_id:  
SELECT * FROM friendship f WHERE user_id = friend_id ;  
  
-- исправил совпадения:  
UPDATE friendship SET friend_id = friend_id + 2 WHERE user_id = friend_id;  
* friend_id + 1 – выдало ошибку, написал friend_id + 2 – прошло без ошибки.  
Совпадений нет.  
  
столбцы confirmed_at created_at updated_at  
-- не определив, что должно быть раньше created_at или confirmed_at, прировнял  
их даты:  
UPDATE friendship SET confirmed_at = created_at ;  
  
-- проверил наличие несоответствий подтвержденные updated_at < created_at:  
SELECT * FROM friendship WHERE updated_at < created_at ;  
несоответствий нет, дата updated_at уже стоит текущая, хотя я не правил.
```

Таблица media_types

```
-- обнулil таблицу:  
TRUNCATE media_types ;  
  
-- заполнил таблицу:  
INSERT INTO media_types (name) VALUES  
  ('photo'),  
  ('video'),  
  ('audio');
```

Таблица media

```
столбец user_id  
-- сделал вразброс нумерацию:  
UPDATE media SET user_id = FLOOR(1 + RAND() * 100);  
  
столбец id  
-- сделал новую нумерацию, которая начинается с «1»:  
UPDATE media SET id = id-100;
```

TRUNCATE не стал использовать, т. к. пришлось бы много заполнять строк.

столбец filename

```
-- скопировал колонку из таблицы media_types, чтоб были рандомные названия
файлов, т. к. на сайте «filldb» я заполнил этот столбец только тремя названиями:
UPDATE media SET filename = (SELECT name FROM media_types ORDER BY RAND() LIMIT
1);
```

```
-- создал наименование файла состоящее из ссылки и расширения файла:
```

```
CREATE TEMPORARY TABLE extensions (name VARCHAR(10));
INSERT INTO extensions VALUES ('jpeg'), ('mp4'), ('mp3'), ('avi'), ('png');
```

```
UPDATE media SET filename = CONCAT(
'http://dropbox.net/vk/',
filename,
'.',
(SELECT name FROM extensions ORDER BY RAND() LIMIT 1));
```

столбец size

```
-- скорректировал столбец с размерами файлов:
```

```
UPDATE media SET `size` = FLOOR(10000 + (RAND() * 1000000)) WHERE `size` <
10000;
```

столбец METADATA

```
-- создал собственника медиафайла:
```

```
UPDATE media SET metadata = CONCAT('{ "owner":',
(SELECT CONCAT(first_name, ' ', last_name) FROM users WHERE id = user_id),
'}');
```

```
-- проверил, имя фамилия свопали
```

```
SELECT first_name, last_name FROM users u WHERE id = 76;
```

столбец media_type_id

```
-- сделал вразброс нумерацию трех типов медиа:
```

```
UPDATE media SET media_type_id = FLOOR(1 + RAND() * 3);
```

столбы created at updated at

```
-- Даты соответствуют
```

2. Подобрать сервис который будет служить основой для вашей курсовой работы.

OZON.ru

3. (по желанию) Предложить свою реализацию лайков и постов.

```
-- Таблица пользователей
```

```
CREATE TABLE users (
id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY COMMENT "Идентификатор
строки",
first_name VARCHAR(100) NOT NULL COMMENT "Имя пользователя",
last_name VARCHAR(100) NOT NULL COMMENT "Фамилия пользователя",
created_at DATETIME DEFAULT CURRENT_TIMESTAMP COMMENT "Время создания строки",
updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
COMMENT "Время обновления строки"
);
```

```

-- Таблица с постами
CREATE TABLE IF NOT EXISTS posts (
  id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY COMMENT "Идентификатор строки",
  user_id INT UNSIGNED NOT NULL COMMENT "Ссылка на пользователя",
  post TEXT NOT NULL COMMENT "Текст поста",
  likes_id INT UNSIGNED NOT NULL COMMENT "Последний номер id таблицы лайков",
  dislikes_id INT UNSIGNED NOT NULL COMMENT "Последний номер id таблицы дизлайков",
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP COMMENT "Время создания строки",
  updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT "Время обновления строки"
);

-- Таблица лайков
CREATE TABLE IF NOT EXISTS likes (
  id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY COMMENT "Идентификатор строки",
  user_id INT UNSIGNED NOT NULL COMMENT "Ссылка на пользователя",
  post_id INT UNSIGNED NOT NULL COMMENT "Ссылка на пост",
  comment TEXT NOT NULL COMMENT "Текст коммента",
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP COMMENT "Время создания строки",
  updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT "Время обновления строки"
); COMMENT "Думаю, что можно еще создать уникальную связь user_id+comment_id для ответных комментов на коммент",

-- Таблица дизлайков
CREATE TABLE IF NOT EXISTS dislikes (
  id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY COMMENT "Идентификатор строки",
  user_id INT UNSIGNED NOT NULL COMMENT "Ссылка на пользователя",
  post_id INT UNSIGNED NOT NULL COMMENT "Ссылка на пост",
  comment TEXT NOT NULL COMMENT "Текст коммента",
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP COMMENT "Время создания строки",
  updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP COMMENT "Время обновления строки"
); COMMENT "Думаю, что еще можно создать уникальную связь user_id+comment_id для ответных комментов на коммент",

```