

```
DROP TABLE IF EXISTS likes;
CREATE TABLE likes (
  id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  user_id INT UNSIGNED NOT NULL,
  target_id INT UNSIGNED NOT NULL,
  target_type ENUM('messages', 'users', 'posts', 'media') NOT
NULL,
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP
);
```

```
-- Временная таблица типов лайков
DROP TABLE IF EXISTS target_types;
CREATE TEMPORARY TABLE target_types (
  name VARCHAR(100) NOT NULL UNIQUE
);
```

```
INSERT INTO target_types (name) VALUES
('messages'),
('users'),
('media'),
('posts');
```

```
-- Заполняем лайки
INSERT INTO likes
SELECT
  id,
  FLOOR(1 + (RAND() * 100)),
  FLOOR(1 + (RAND() * 100)),
  (SELECT name FROM target_types ORDER BY RAND() LIMIT 1),
  CURRENT_TIMESTAMP
FROM messages;
```

```
-- Проверим
SELECT * FROM likes LIMIT 10;
```

```
-- Создадим таблицу постов
DROP TABLE IF EXISTS posts;
CREATE TABLE posts (
  id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY,
  user_id INT UNSIGNED NOT NULL,
  community_id INT UNSIGNED,
  head VARCHAR(255),
  body TEXT NOT NULL,
  media_id INT UNSIGNED,
  is_public BOOLEAN DEFAULT TRUE,
  is_deleted BOOLEAN DEFAULT FALSE,
  created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
  updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE
CURRENT_TIMESTAMP
);
```

```
SELECT * FROM posts LIMIT 10;
```

```
SELECT * FROM communities;  
SELECT * FROM media;
```

```
-- Добавляем внешние ключи в БД vk  
-- Для таблицы профилей
```

```
-- Смотрим структуру таблицы
```

```
DESC profiles;  
SELECT * FROM profiles;  
DESC users;  
DESC communities_users;  
DESC friendship;  
SELECT * FROM friendship;  
DESC cities;  
DESC media;  
DESC likes;  
SELECT * FROM likes;  
DESC posts;
```

```
-- Добавляем внешние ключи
```

```
ALTER TABLE profiles  
  ADD CONSTRAINT profiles_user_id_fk  
    FOREIGN KEY (user_id) REFERENCES users(id)  
    ON DELETE CASCADE;
```

```
ALTER TABLE profiles  
  ADD CONSTRAINT profiles_city_id_fk  
    FOREIGN KEY (city_id) REFERENCES cities(id)  
    ON DELETE SET NULL;
```

```
ALTER TABLE communities_users  
  ADD CONSTRAINT communities_users_user_id_fk  
    FOREIGN KEY (user_id) REFERENCES users(id)  
    ON DELETE CASCADE;
```

```
ALTER TABLE communities_users  
  ADD CONSTRAINT communities_users_community_id_fk  
    FOREIGN KEY (community_id) REFERENCES communities(id);
```

```
ALTER TABLE friendship  
  ADD CONSTRAINT friendship_user_id_fk  
    FOREIGN KEY (user_id) REFERENCES users(id)  
    ON DELETE CASCADE;
```

```
ALTER TABLE friendship  
  ADD CONSTRAINT friendship_friend_id_fk  
    FOREIGN KEY (friend_id) REFERENCES users(id)  
    ON DELETE CASCADE;
```

```

ALTER TABLE friendship
  ADD CONSTRAINT friendship_friendship_status_id_fk
    FOREIGN KEY (friendship_status_id) REFERENCES
friendship_statuses(id);

ALTER TABLE cities
  ADD CONSTRAINT cities_country_id_fk
    FOREIGN KEY (country_id) REFERENCES countries(id);

ALTER TABLE media
  ADD CONSTRAINT media_user_id_fk
    FOREIGN KEY (user_id) REFERENCES users(id)
    ON DELETE CASCADE;

ALTER TABLE media
  ADD CONSTRAINT media_media_type_id_fk
    FOREIGN KEY (media_type_id) REFERENCES media_types(id);

ALTER TABLE likes
  ADD CONSTRAINT likes_user_id_fk
    FOREIGN KEY (user_id) REFERENCES users(id)
    ON DELETE CASCADE;

ALTER TABLE posts
  ADD CONSTRAINT posts_user_id_fk
    FOREIGN KEY (user_id) REFERENCES users(id)
    ON DELETE CASCADE,
  ADD CONSTRAINT posts_community_id_fk
    FOREIGN KEY (community_id) REFERENCES communities(id),
  ADD CONSTRAINT posts_media_id_fk
    FOREIGN KEY (media_id) REFERENCES media(id);

-- Изменяем тип столбца при необходимости
ALTER TABLE profiles DROP FOREIGN KEY profiles_user_id_fk;
ALTER TABLE profiles MODIFY COLUMN photo_id INT(10) UNSIGNED;

-- Для таблицы сообщений
-- Смотрим структурв таблицы
DESC messages;

-- Добавляем внешние ключи
ALTER TABLE messages
  ADD CONSTRAINT messages_from_user_id_fk
    FOREIGN KEY (from_user_id) REFERENCES users(id),
  ADD CONSTRAINT messages_to_user_id_fk
    FOREIGN KEY (to_user_id) REFERENCES users(id),
  ADD CONSTRAINT messages_media_id_fk
    FOREIGN KEY (media_id) REFERENCES media(id);

```

-- 3. Определить кто больше поставил лайков (всего) – мужчины или женщины?

```
SELECT
    gender,
    COUNT(gender) AS 'sum'
FROM
    likes
INNER JOIN profiles
    ON likes.user_id = profiles.user_id
GROUP BY gender;
```

```
SELECT IF ((SELECT p.gender FROM profiles AS p WHERE p.user_id =
l.user_id) = 'М', 'Мужчины', 'Женщины') gender_like, COUNT(*)
like_count FROM likes AS l
GROUP BY gender_like
ORDER BY like_count DESC
LIMIT 1
```

-- 4. Вывести для каждого пользователя количество созданных сообщений, постов, загруженных медиафайлов и поставленных лайков.

```
SELECT
    id,
    CONCAT(first_name, ' ', last_name),
    (SELECT COUNT(m.from_user_id) FROM messages AS m WHERE
m.from_user_id = u.id GROUP BY m.from_user_id) AS 'Sum Message',
    (SELECT COUNT(p.user_id) FROM posts AS p WHERE p.user_id = u.id
GROUP BY p.user_id) AS 'Sum Posts',
    (SELECT COUNT(d.user_id) FROM media AS d WHERE d.user_id = u.id
GROUP BY d.user_id) AS 'Sum Media',
    (SELECT COUNT(l.user_id) FROM likes AS l WHERE l.user_id = u.id
GROUP BY l.user_id) AS 'Sum Likes'
FROM users AS u;
```

```
SELECT CONCAT(u.last_name, ' ', u.first_name) AS user_name,
(SELECT COUNT(*) FROM messages AS m WHERE m.from_user_id = u.id)
AS messages,
(SELECT COUNT(*) FROM posts AS p WHERE p.user_id = u.id) AS posts,
(SELECT COUNT(*) FROM media AS m WHERE m.user_id = u.id) AS
mediafiles,
(SELECT COUNT(*) FROM likes AS l WHERE l.user_id = u.id) AS likes
FROM users AS u;
```

-- 5.(по желанию) Подсчитать количество лайков которые получили 10 самых последних сообщений.

```
SELECT SUM(target_id) FROM likes WHERE (SELECT target_type =
'messages' FROM likes ORDER BY created_at DESC LIMIT 10);
```

```
SELECT target_id FROM likes WHERE target_type = 'messages' ORDER
BY created_at DESC LIMIT 10,
SUM(target_id) FROM likes;
```

```
SELECT SUM(target_id) FROM likes WHERE target_type = 'messages'
ORDER BY created_at DESC LIMIT 10;
```

```
SELECT
  (SELECT target_id FROM likes WHERE target_type = 'messages'
ORDER BY created_at DESC LIMIT 10) AS 't',
  SUM(target_id) AS total
FROM likes;
```

не могу вывести ответ 246, ругается на строку

```
SELECT
  user_id,
  (SELECT CONCAT(first_name, ' ', last_name) FROM users WHERE
users.id = media.user_id) AS user,
  SUM(size) AS total
FROM media
  GROUP BY user_id
  HAVING total > 100000000;
```

```
SELECT * FROM likes;
```