1. задание

SHOW TABLES;

DESC profiles;

ALTER TABLE profiles MODIFY COLUMN photo\_id INT(10) UNSIGNED;

ALTER TABLE profiles

ADD CONSTRAINT profiles\_user\_id\_fk

FOREIGN KEY (user\_id) REFERENCES users(id)

ON DELETE CASCADE,

ADD CONSTRAINT profiles\_photo\_id\_fk

FOREIGN KEY (photo\_id) REFERENCES media(id)

ON DELETE SET NULL;

DESC users;

DESC media;

DESC media\_types;

ALTER TABLE media MODIFY COLUMN user\_id INT(10) UNSIGNED;

ALTER TABLE media MODIFY COLUMN media\_type\_id INT(10) UNSIGNED;

ALTER TABLE media

ADD CONSTRAINT media\_user\_id\_fk

FOREIGN KEY (user\_id) REFERENCES users(id)

ON DELETE SET NULL,

ADD CONSTRAINT media\_media\_type\_id\_fk

FOREIGN KEY (media\_type\_id) REFERENCES media\_types(id)

ON DELETE SET NULL;

DESC communities;

DESC communities\_users;

ALTER TABLE communities\_users

ADD CONSTRAINT communities\_users\_user\_id\_fk

FOREIGN KEY (user\_id) REFERENCES users(id)

ON DELETE CASCADE,

ADD CONSTRAINT communities\_users\_community\_id\_fk

FOREIGN KEY (community\_id) REFERENCES communities(id)

ON DELETE CASCADE;

DESC friendship;

DESC friendship\_statuses;

ALTER TABLE friendship MODIFY COLUMN status\_id INT(10) UNSIGNED;

ALTER TABLE friendship

ADD CONSTRAINT friendship\_user\_id\_fk

FOREIGN KEY (user\_id) REFERENCES users(id)

ON DELETE CASCADE,

ADD CONSTRAINT friendship\_friend\_id\_fk

FOREIGN KEY (friend\_id) REFERENCES users(id)

ON DELETE CASCADE,

ADD CONSTRAINT friendship\_status\_id\_fk

FOREIGN KEY (status\_id) REFERENCES friendship\_statuses(id)

ON DELETE SET NULL;

DESC likes;

DESC target\_types;

ALTER TABLE likes MODIFY COLUMN user\_id INT(10) UNSIGNED;

ALTER TABLE likes MODIFY COLUMN target\_type\_id INT(10) UNSIGNED;

ALTER TABLE likes

ADD CONSTRAINT likes\_user\_id\_fk

FOREIGN KEY (user\_id) REFERENCES users(id)

ON DELETE SET NULL,

ADD CONSTRAINT likes\_target\_id\_fk

FOREIGN KEY (target\_id) REFERENCES users(id)

ON DELETE CASCADE,

ADD CONSTRAINT likes\_target\_type\_id\_fk

FOREIGN KEY (target\_type\_id) REFERENCES target\_types(id)

ON DELETE SET NULL;

DESC meetings;

DESC meetings\_users;

ALTER TABLE meetings\_users

ADD CONSTRAINT meetings\_users\_user\_id\_fk

FOREIGN KEY (user\_id) REFERENCES users(id);

ALTER TABLE meetings

ADD CONSTRAINT meetings\_user\_id\_fk

FOREIGN KEY (user\_id) REFERENCES users(id);

DESC messages;

DESC posts;

ALTER TABLE messages MODIFY COLUMN to\_user\_id INT(10) UNSIGNED;

ALTER TABLE messages MODIFY COLUMN from\_user\_id INT(10) UNSIGNED;

ALTER TABLE messages

ADD CONSTRAINT messages\_from\_user\_id\_fk

FOREIGN KEY (from\_user\_id) REFERENCES users(id)

ON DELETE SET NULL,

ADD CONSTRAINT messages\_to\_user\_id\_fk

FOREIGN KEY (to\_user\_id) REFERENCES users(id)

ON DELETE SET NULL;

ALTER TABLE posts MODIFY COLUMN user\_id INT(10) UNSIGNED;

ALTER TABLE posts MODIFY COLUMN media\_id INT(10) UNSIGNED;

ALTER TABLE posts

ADD CONSTRAINT posts\_user\_id\_fk

FOREIGN KEY (user\_id) REFERENCES users(id)

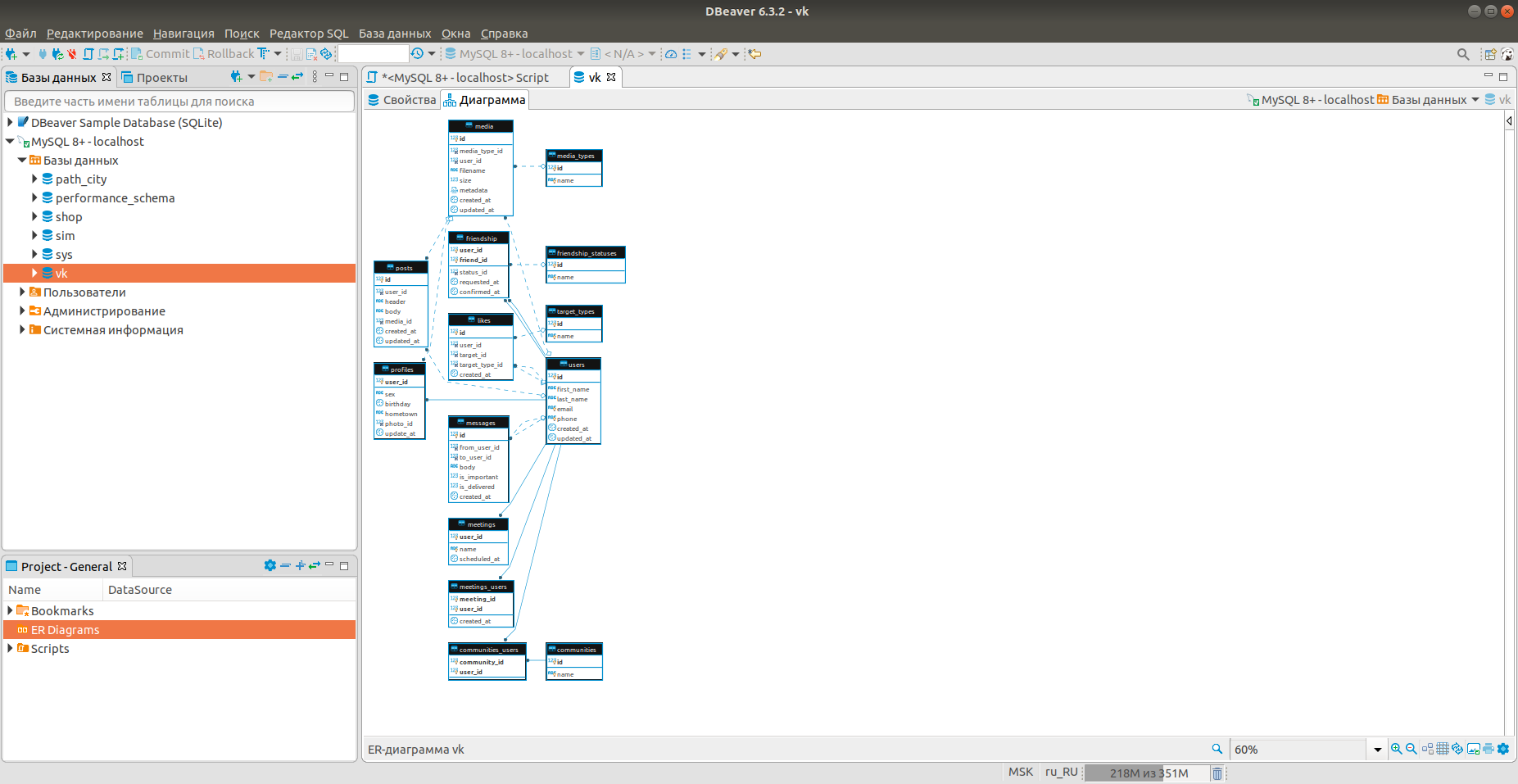
ON DELETE SET NULL,

ADD CONSTRAINT posts\_media\_id\_fk

FOREIGN KEY (media\_id) REFERENCES media(id)

ON DELETE SET NULL;

1. задание



1. 3.1. Задание.

Пусть задан некоторый пользователь.

Из всех друзей этого пользователя найдите человека, который больше всех общался с нашим пользователем.

-- Выберем id друзей

SELECT \* FROM friendship WHERE user\_id = 3 OR friend\_id = 3;

-- В один столбец

SELECT friend\_id AS id FROM friendship WHERE user\_id = 3

UNION

SELECT user\_id AS id FROM friendship WHERE friend\_id = 3;

-- Выбираем id отправителей сообщений

SELECT from\_user\_id FROM messages

WHERE to\_user\_id = 3

AND from\_user\_id IN (

SELECT friend\_id AS id FROM friendship WHERE user\_id = 3

UNION

SELECT user\_id AS id FROM friendship WHERE friend\_id = 3

);

-- Добавляем имя

SELECT (SELECT CONCAT(first\_name, ' ', last\_name) FROM users WHERE id = from\_user\_id) AS friend

FROM messages

WHERE to\_user\_id = 3

AND from\_user\_id IN (

SELECT friend\_id AS id FROM friendship WHERE user\_id = to\_user\_id

UNION

SELECT user\_id AS id FROM friendship WHERE friend\_id = to\_user\_id

);

-- Добавляем подсчёт и сортировку

SELECT (SELECT CONCAT(first\_name, ' ', last\_name) FROM users WHERE id = from\_user\_id) AS friend,

COUNT(\*) AS total\_messages

FROM messages

WHERE to\_user\_id = 3

AND from\_user\_id IN (

SELECT friend\_id AS id FROM friendship WHERE user\_id = to\_user\_id

UNION

SELECT user\_id AS id FROM friendship WHERE friend\_id = to\_user\_id

)

GROUP BY messages.from\_user\_id

ORDER BY total\_messages DESC

LIMIT 1;

* 1. Задание.

Подсчитать общее количество лайков, которые получили 10 самых молодых пользователей.

-- Смотрим типы для лайков

SELECT \* FROM target\_types;

-- Выбираем профили с сортировкой по дате рождения

SELECT \* FROM profiles ORDER BY birthday DESC LIMIT 10;

-- Выбираем лайки по типу пользователь

SELECT \* FROM likes WHERE target\_type\_id = 2;

-- Объединяем, но так не работает

SELECT \* FROM likes WHERE target\_type\_id = 2

AND target\_id IN (

SELECT user\_id FROM profiles ORDER BY birthday DESC LIMIT 10

);

-- Идём обходным путём

SELECT target\_id, COUNT(\*) FROM likes

WHERE target\_type\_id = 2

AND target\_id IN (SELECT \* FROM (

SELECT user\_id FROM profiles ORDER BY birthday DESC LIMIT 10

) AS sorted\_profiles )

GROUP BY target\_id;

-- Суммируем для всех пользователей

SELECT SUM(likes\_per\_user) AS likes\_total FROM (

SELECT COUNT(\*) AS likes\_per\_user

FROM likes

WHERE target\_type\_id = 2

AND target\_id IN (

SELECT \* FROM (

SELECT user\_id FROM profiles ORDER BY birthday DESC LIMIT 10

) AS sorted\_profiles

)

GROUP BY target\_id

) AS counted\_likes;

* 1. Задание.

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

SELECT CASE(sex)

WHEN 'm' THEN 'man'

WHEN 'f' THEN 'woman'

END AS sex,

COUNT(\*) as likes\_count

FROM (

SELECT

user\_id as user,

(SELECT sex FROM profiles WHERE user\_id = user) as sex

FROM likes) dummy\_table

GROUP BY sex

ORDER BY likes\_count DESC

LIMIT 1;

* 1. Задание.

Найти 10 пользователей, которые проявляют наименьшую активность в использовании социальной сети.

SELECT CONCAT(first\_name, ' ', last\_name) AS user,

(SELECT COUNT(\*) FROM likes WHERE likes.user\_id = users.id) +

(SELECT COUNT(\*) FROM media WHERE media.user\_id = users.id) +

(SELECT COUNT(\*) FROM messages WHERE messages.from\_user\_id = users.id)

AS overall\_activity

FROM users

ORDER BY overall\_activity

LIMIT 10;