

Урок 6. SQL – Транзакции. Временные таблицы, управляющие конструкции, циклы

1) Написать функцию, которая удаляет всю информацию об указанном пользователе из БД vk.

Пользователь задается по id.

Удалить нужно все сообщения, лайки, медиа записи, профиль и запись из таблицы users.

Функция должна возвращать номер пользователя.

1) создаем таблицу vk2

```
DROP DATABASE IF EXISTS vk2;  
CREATE DATABASE vk2;  
USE vk2;
```

```
DROP TABLE IF EXISTS users;  
CREATE TABLE users (  
    id BIGINT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY  
KEY,  
    firstname VARCHAR(50),  
    lastname VARCHAR(50) COMMENT 'Фамилия',  
    email VARCHAR(120) UNIQUE,  
    phone BIGINT,  
    INDEX users_phone_idx(phone),  
    INDEX users_firstname_lastname_idx(firstname,  
lastname)  
) COMMENT 'юзеры';  
INSERT INTO `users` (`id`, `firstname`, `lastname`,  
`email`, `phone`) VALUES ('1', 'Reuben', 'Nienow',  
'arlo50@example.org', '9374071116');  
INSERT INTO `users` (`id`, `firstname`, `lastname`,  
`email`, `phone`) VALUES ('2', 'Reuben', 'Nienow',  
'terrence.cartwright@example.org', '9127498182');  
INSERT INTO `users` (`id`, `firstname`, `lastname`,  
`email`, `phone`) VALUES ('3', 'Reuben', 'Windler',  
'rupert55@example.org', '9921090703');  
INSERT INTO `users` (`id`, `firstname`, `lastname`,  
`email`, `phone`) VALUES ('4', 'Reuben', 'West',  
'rebekah29@example.net', '9592139196');  
INSERT INTO `users` (`id`, `firstname`, `lastname`,  
`email`, `phone`) VALUES ('5', 'Frederick', 'Effertz',  
'von.bridget@example.net', '9909791725');  
INSERT INTO `users` (`id`, `firstname`, `lastname`,  
`email`, `phone`) VALUES ('6', 'Frederick', 'Medhurst',  
'sstehr@example.net', '9456642385');
```

```

INSERT INTO `users` (`id`, `firstname`, `lastname`,
`email`, `phone`) VALUES ('7', 'Frederick', 'Braun',
'itzel.beahan@example.com', '9448906606');
INSERT INTO `users` (`id`, `firstname`, `lastname`,
`email`, `phone`) VALUES ('8', 'Frederick', 'Kilback',
'johnathan.wisozk@example.com', '9290679311');
INSERT INTO `users` (`id`, `firstname`, `lastname`,
`email`, `phone`) VALUES ('9', 'Frederick', 'Orn',
'missouri87@example.org', '9228624339');
INSERT INTO `users` (`id`, `firstname`, `lastname`,
`email`, `phone`) VALUES ('10', 'Jordyn', 'Jerde',
'edach@example.com', '9443126821');

```

```

DROP TABLE IF EXISTS messages;

```

```

CREATE TABLE messages (
    id SERIAL,
    from_user_id BIGINT UNSIGNED NOT NULL,
    to_user_id BIGINT UNSIGNED NOT NULL,
    body TEXT,
    created_at DATETIME DEFAULT NOW(),
    INDEX messages_from_user_id (from_user_id),
    INDEX messages_to_user_id (to_user_id),
    FOREIGN KEY (from_user_id) REFERENCES users(id),
    FOREIGN KEY (to_user_id) REFERENCES users(id)
);

```

```

INSERT INTO `messages` VALUES
('1','1','2','Voluptatem ut quaerat quia. Pariatur esse
amet ratione qui quia. In necessitatibus reprehenderit
et. Nam accusantium aut qui quae nesciunt non.','1995-08-
28 22:44:29'),
('2','2','1','Sint dolores et debitis est ducimus. Aut et
quia beatae minus. Ipsa rerum totam modi sunt sed.
Voluptas atque eum et odio ea molestias ipsam
architecto.','now()),
('3','3','1','Sed mollitia quo sequi nisi est tenetur at
rerum. Sed quibusdam illo ea facilis nemo sequi. Et
tempora repudiandae saepe quo.','1993-09-14 19:45:58'),
('4','1','3','Quod dicta omnis placeat id et officiis et.
Beatae enim aut aliquid neque occaecati odit. Facere eum
distinctio assumenda omnis est delectus magnam.','1985-
11-25 16:56:25'),
('5','1','5','Voluptas omnis enim quia porro debitis
facilis eaque ut. Id inventore non corrupti doloremque

```

consequuntur. Molestiae molestiae deleniti exercitationem sunt qui ea accusamus deserunt.', '1999-09-19 04:35:46'), ('6', '1', '6', 'Rerum labore culpa et laboriosam eum totam. Quidem pariat sit alias. Atque doloribus ratione eum rem dolor vitae saepe.', '1973-11-09 08:12:04'), ('7', '1', '7', 'Perspiciatis temporibus doloribus debitis. Et inventore labore eos modi. Quo temporibus corporis minus. Accusamus aspernatur nihil nobis placeat molestiae et commodi eaque.', '1998-04-09 00:00:26'), ('8', '8', '8', 'Suscipit dolore voluptas et sit vero et sint. Rem ut ratione voluptatum assumenda nesciunt ea. Quas qui qui atque ut. Similique et praesentium non voluptate iure. Eum aperiam officia quia dolorem.', '2005-08-20 18:40:27'), ('9', '9', '9', 'Et quia libero aut vitae minus. Rerum a blanditiis debitis sit nam. Veniam quasi aut autem ratione dolorem. Sunt quo similique dolorem odit totam sint sed.', '2013-03-19 04:10:10'), ('10', '10', '10', 'Praesentium molestias quia aut odio. Est quis eius ut animi optio molestiae. Amet tempore sequi blanditiis in est.', '1976-05-22 14:38:15');

```
DROP TABLE IF EXISTS friend_requests;
CREATE TABLE friend_requests (
    -- id SERIAL PRIMARY KEY, -- changed to combined
primary key (initiator_user_id, target_user_id)
    initiator_user_id BIGINT UNSIGNED NOT NULL,
    target_user_id BIGINT UNSIGNED NOT NULL,
    -- `status` TINYINT UNSIGNED,
    `status` ENUM('requested', 'approved', 'unfriended',
'declined'),
    requested_at DATETIME DEFAULT NOW(),
    confirmed_at DATETIME,
    INDEX (initiator_user_id),
    INDEX (target_user_id),
    PRIMARY KEY (initiator_user_id, target_user_id),
    FOREIGN KEY (initiator_user_id) REFERENCES users(id),
    FOREIGN KEY (target_user_id) REFERENCES users(id)
);
INSERT INTO `friend_requests` (`initiator_user_id`,
`target_user_id`, `status`, `requested_at`,
`confirmed_at`) VALUES ('1', '10', 'approved', '1970-08-
21 06:40:37', '2003-12-29 23:20:55');
```

```

INSERT INTO `friend_requests` (`initiator_user_id`,
`target_user_id`, `status`, `requested_at`,
`confirmed_at`) VALUES ('1', '2', 'requested', '1987-10-
10 07:33:23', '1983-01-18 01:56:05');
INSERT INTO `friend_requests` (`initiator_user_id`,
`target_user_id`, `status`, `requested_at`,
`confirmed_at`) VALUES ('1', '3', 'approved', '2015-02-07
01:53:07', '2015-09-14 16:17:12');
INSERT INTO `friend_requests` (`initiator_user_id`,
`target_user_id`, `status`, `requested_at`,
`confirmed_at`) VALUES ('4', '1', 'approved', '1983-04-08
15:57:26', '1976-12-28 23:54:58');
INSERT INTO `friend_requests` (`initiator_user_id`,
`target_user_id`, `status`, `requested_at`,
`confirmed_at`) VALUES ('5', '2', 'unfriended', '1976-03-
03 18:22:00', '1996-12-03 11:29:04');
INSERT INTO `friend_requests` (`initiator_user_id`,
`target_user_id`, `status`, `requested_at`,
`confirmed_at`) VALUES ('6', '3', 'unfriended', '2008-12-
06 17:07:59', '2015-11-01 08:03:23');
INSERT INTO `friend_requests` (`initiator_user_id`,
`target_user_id`, `status`, `requested_at`,
`confirmed_at`) VALUES ('7', '1', 'requested', '1989-10-
26 06:20:23', '2001-08-05 16:01:03');
INSERT INTO `friend_requests` (`initiator_user_id`,
`target_user_id`, `status`, `requested_at`,
`confirmed_at`) VALUES ('8', '6', 'unfriended', '1987-12-
30 01:50:03', '2001-07-09 07:39:50');
INSERT INTO `friend_requests` (`initiator_user_id`,
`target_user_id`, `status`, `requested_at`,
`confirmed_at`) VALUES ('9', '7', 'requested', '2003-12-
27 22:52:09', '1998-04-18 04:41:46');

```

```

DROP TABLE IF EXISTS communities;

```

```

CREATE TABLE communities(
    id SERIAL,
    name VARCHAR(150),
    INDEX communities_name_idx(name)
);

```

```

/*!40000 ALTER TABLE `communities` DISABLE KEYS */;

```

```

INSERT INTO `communities` VALUES (2, 'atque'),
(1, 'beatae'), (9, 'est'), (5, 'eum'), (7, 'hic'), (6, 'nemo'),
(8, 'quis'), (4, 'rerum'), (10, 'tempora'), (3, 'voluptas');

```

```

/*!40000 ALTER TABLE `communities` ENABLE KEYS */;

```

```

DROP TABLE IF EXISTS users_communities;
CREATE TABLE users_communities(
    user_id BIGINT UNSIGNED NOT NULL,
    community_id BIGINT UNSIGNED NOT NULL,
    PRIMARY KEY (user_id, community_id),
    FOREIGN KEY (user_id) REFERENCES users(id),
    FOREIGN KEY (community_id) REFERENCES communities(id)
);
INSERT INTO `users_communities` (`user_id`,
`community_id`) VALUES ('1', '1');
INSERT INTO `users_communities` (`user_id`,
`community_id`) VALUES ('1', '2');
INSERT INTO `users_communities` (`user_id`,
`community_id`) VALUES ('1', '3');
INSERT INTO `users_communities` (`user_id`,
`community_id`) VALUES ('2', '4');
INSERT INTO `users_communities` (`user_id`,
`community_id`) VALUES ('2', '5');
INSERT INTO `users_communities` (`user_id`,
`community_id`) VALUES ('6', '6');
INSERT INTO `users_communities` (`user_id`,
`community_id`) VALUES ('7', '7');
INSERT INTO `users_communities` (`user_id`,
`community_id`) VALUES ('8', '8');
INSERT INTO `users_communities` (`user_id`,
`community_id`) VALUES ('9', '9');
INSERT INTO `users_communities` (`user_id`,
`community_id`) VALUES ('10', '10');

DROP TABLE IF EXISTS media_types;
CREATE TABLE media_types(
    id SERIAL,
    name VARCHAR(255),
    created_at DATETIME DEFAULT NOW()
);
/*!40000 ALTER TABLE `media_types` DISABLE KEYS */;
INSERT INTO `media_types` VALUES
    (1, 'Photo', '2003-07-09 10:08:05'),
    (2, 'Music', '2009-06-19 20:08:09'),
    (3, 'Video', '1984-04-18 01:55:09'),
    (4, 'Post', '2001-04-17 06:47:52');
/*!40000 ALTER TABLE `media_types` ENABLE KEYS */;

```

```

DROP TABLE IF EXISTS media;
CREATE TABLE media(
    id SERIAL,
    media_type_id BIGINT UNSIGNED NOT NULL,
    user_id BIGINT UNSIGNED NOT NULL,
    body text,
    filename VARCHAR(255),
    size INT,
    metadata JSON,
    created_at DATETIME DEFAULT NOW(),
    updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON
UPDATE CURRENT_TIMESTAMP,
    INDEX (user_id),
    FOREIGN KEY (user_id) REFERENCES users(id),
    FOREIGN KEY (media_type_id) REFERENCES
media_types(id)
);
INSERT INTO media
(media_type_id,user_id,body,filename,`size`,metadata,crea
ted_at,updated_at) VALUES
(1,1,'Est aut aliquid est. Soluta eveniet doloribus ipsam
ut id assumenda quam. Consequuntur et velit aperiam error
corporis rem corporis. Est similique quisquam quia maxime
eaeque possimus facere officiis. Odio magni atque dicta
voluptatibus hic iste
delectus.','user1_avatar.jpg',5,NULL,'1991-11-16
08:14:38.000','2019-06-19 18:11:58.000')
,(2,2,'Laborum dolorem magni unde vero sit hic. Pariatur
quia voluptas magnam. Blanditiis dolorem et molestias
veniam facere
provident.','asperiores.mp4',4510,NULL,'1986-12-10
19:34:26.000','2019-06-19 18:11:58.000')
,(1,3,'A quam ea ullam reiciendis vel et reiciendis.
Delectus iure mollitia est. Eum dolores dolores sed
officia vitae qui. Sed repudiandae voluptatem
voluptatibus libero culpa
quo.','sapiente.avi',763116,NULL,'1978-09-18
17:07:49.000','2019-06-19 18:11:58.000')
,(4,4,'Velit commodi in veniam occaecati. Tenetur
temporibus ullam enim officiis aut illum. In maiores
velit nobis soluta molestias maiores tempora. Quisquam et
ut et sequi.','rerum.mp4',740,NULL,'1984-10-21
04:41:15.000','2019-06-19 18:11:58.000')
,(1,5,'Est eaeque est qui et omnis rerum. Expedita porro
nesciunt inventore consequatur quos. Inventore aut rerum

```

```

voluptas. Dicta vel ut quibusdam cupiditate
tempore.', 'user5-profile-image.mp4', 40813, NULL, '2014-06-
20 03:34:11.000', '2019-06-19 18:11:58.000')
, (2, 6, 'Atque et nam nulla et aliquid. Vel repellendus aut
natus odit quibusdam quis atque et. Laudantium minima
velit dolores eos voluptate
nostrum.', 'quasi.jpg', 119422, NULL, '1991-11-02
07:50:45.000', '2019-06-19 18:11:58.000')
, (3, 1, 'Nihil ut et reprehenderit et. Numquam veniam quis
impedit. Voluptas saepe rerum illo vel omnis. Aliquam
illum ut quo saepe ea unde
est.', 'non.mp4', 704558, NULL, '1988-09-17
20:56:38.000', '2019-06-19 18:11:58.000')
, (3, 1, 'Sint quod fugit molestiae dolore repellendus est.
Quis corporis necessitatibus commodi placeat temporibus
doloremque voluptatem. Nemo maxime sit dolorem
preferendis repellat. Et impedit deleniti eaque pariat
ut quia voluptatibus.', 'earum.mp4', 65108, NULL, '1995-10-07
10:19:53.000', '2019-06-19 18:11:58.000')
, (3, 1, 'Optio sed aperiam veniam eum. Rerum placeat soluta
iusto perspiciatis quibusdam dolorem eos. Ea pariat
optio est ut.', 'ullam.avi', 83717449, NULL, '2003-08-05
05:55:25.000', '2019-06-19 18:11:58.000')
, (3, 10, 'Facere nostrum facilis aperiam quisquam dolor.
Minima omnis est nam.', 'maiores.avi', 75, NULL, '1984-05-17
07:37:30.000', '2019-06-19 18:11:58.000')
;
INSERT INTO media
(media_type_id, user_id, body, filename, `size`, metadata, crea
ted_at, updated_at) VALUES
(3, 1, 'A omnis ratione non. Et velit sed incidunt corporis
ut rerum nemo. Ut pariat tempora ea incidunt
praesentium ut. Eligendi et et rerum sequi
ut.', 'non.mp4', 0, NULL, '2010-05-07 19:13:09.000', '2019-06-
19 18:11:58.000')
, (4, 1, 'Quo minus harum debitis debitis quis sunt. Dolores
suscipit placeat dolorum non voluptate et. Non eos odio
esse tempore. Illum minus quidem temporibus
porro.', 'earum.jpg', 1826766, NULL, '2013-08-25
22:49:45.000', '2019-06-19 18:11:58.000')
, (1, 1, 'Et aliquid sit natus vitae fuga enim repudiandae.
Quis ab et debitis est accusantium labore. Exercitationem
delectus eaque necessitatibus.', 'hic.jpg', 979, NULL, '2015-
11-02 00:11:31.000', '2019-06-19 18:11:58.000')

```

```
, (2,1,'Nobis est sed blanditiis assumenda incidunt
explicabo. Facere rem assumenda odio explicabo ad enim
repellat quia. Dolor mollitia fugiat officia nihil
animi.','iste.jpg',18,NULL,'2006-05-11
12:11:06.000','2019-06-19 18:11:58.000')
, (3,1,'Tempora ad et aspernatur laborum ut dolor et.
Exercitationem quaerat corporis placeat
et.','quia.jpg',7,NULL,'2000-03-29 04:01:57.000','2019-
06-19 18:11:58.000')
, (4,1,'Accusantium est ea fuga omnis mollitia. Dolores
officia et consequatur iste est quo. Ullam laborum qui ut
quo delectus temporibus.','omnis.jpg',2058670,NULL,'1971-
10-08 06:43:32.000','2019-06-19 18:11:58.000')
, (1,1,'Quasi itaque atque ut aliquam debitis. Qui
consequuntur maiores sit ad perspiciatis quaerat
assumenda repudiandae. Necessitatibus placeat quaerat aut
voluptatem ratione voluptate expedita. Nulla doloremque
aut molestiae nam.','aperiam.jpg',3489253,NULL,'1973-12-
19 09:20:43.000','2019-06-19 18:11:58.000')
, (2,1,'Deserunt voluptatem quia voluptatem sit. Qui omnis
distinctio optio voluptatem veniam atque dolorem.
Repellat laboriosam illo consequatur nesciunt voluptas
magni rerum. Saepe quos qui
doloremque.','commodi.jpg',8,NULL,'1989-03-01
17:19:42.000','2019-06-19 18:11:58.000')
, (3,5,'Fugiat consequuntur voluptatem odit omnis. Quia
aut voluptate officia rerum. Cumque voluptatem eaque
dolorum voluptas sed. Et autem atque est minus veritatis
explicabo eos.','id.avi',1,NULL,'1970-11-04
03:06:33.000','2019-06-19 18:11:58.000')
, (4,2,'Autem dolore beatae aut corporis fugit ratione. Ex
beatae qui at. Est deleniti asperiores temporibus
preferendis est.','fugiat.jpg',82,NULL,'2017-06-08
03:23:04.000','2019-06-19 18:11:58.000')
;
```

```
DROP TABLE IF EXISTS `profiles`;
CREATE TABLE `profiles` (
  user_id BIGINT UNSIGNED NOT NULL UNIQUE,
  gender CHAR(1),
  birthday DATE,
  photo_id BIGINT UNSIGNED NULL,
  created_at DATETIME DEFAULT NOW(),
  hometown VARCHAR(100),
```



```

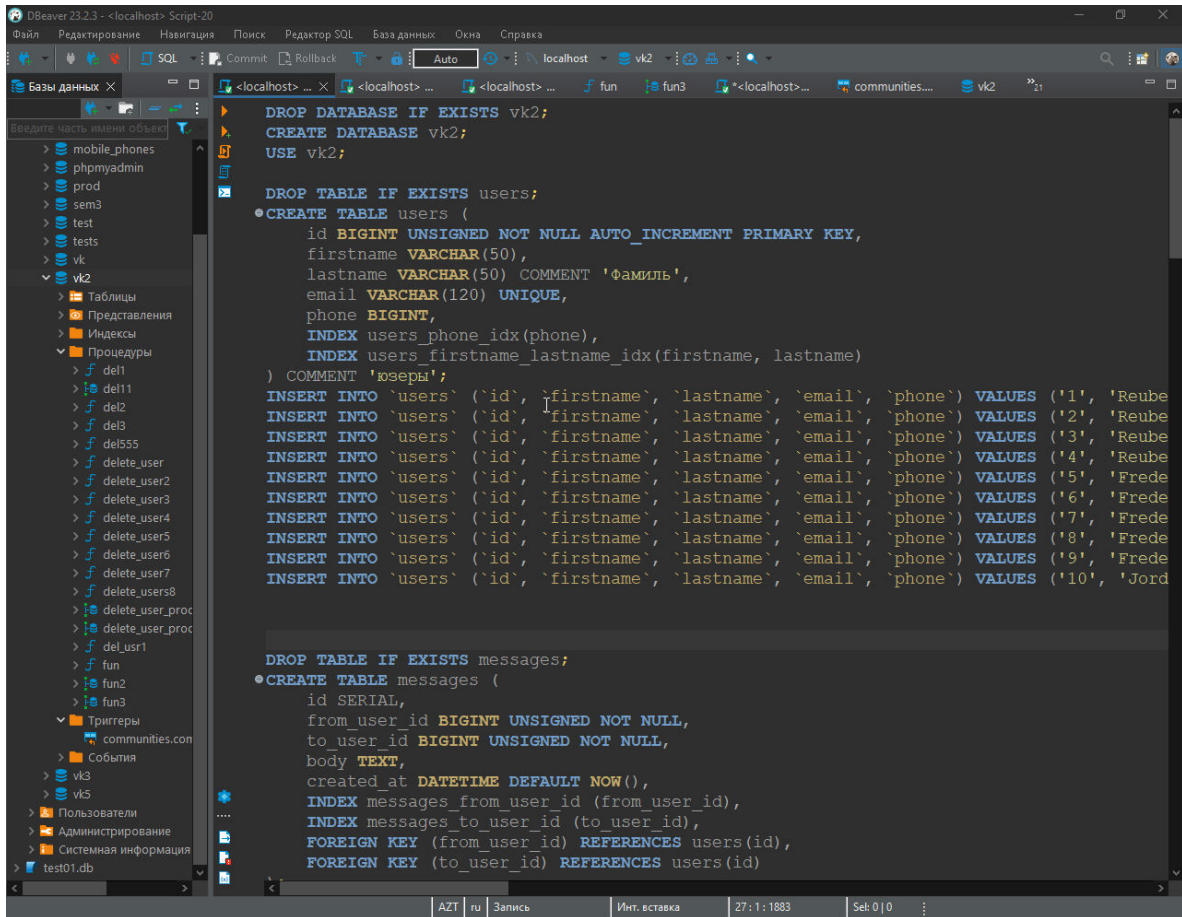
    FOREIGN KEY (user_id) REFERENCES users(id) ON UPDATE
    CASCADE ON DELETE restrict,
    FOREIGN KEY (photo_id) REFERENCES media(id)
);
INSERT INTO `profiles` VALUES
('1','f','1976-11-08','1','1970-01-18
03:54:01','Adriannaport'),
('2','f','2011-12-20','2','1994-11-06 23:56:10','North
Nettieton'),
('3','m','1994-06-15','3','1975-11-27
02:27:11','Padbergtown'),
('4','f','1979-11-07','4','1994-04-12 04:31:49','Port
Rupertville'),
('5','f','1976-04-19','5','1976-07-05
02:25:30','Spencerfort'),
('6','f','1983-09-07','6','1981-06-20 15:54:43','South
Woodrowmouth'),
('7','m','2014-07-31','7','1997-06-21 07:52:05','South
Jeffereyshire'),
('8','f','1975-03-26','8','2008-08-18
18:23:25','Howeside'),
('9','f','1982-11-01','9','2014-09-29
01:22:26','Loweborough'),
('10','m','1977-05-14','10','1980-03-17 18:17:45','New
Nellaside');

```

```

DROP TABLE IF EXISTS likes;
CREATE TABLE likes(
    id SERIAL,
    user_id BIGINT UNSIGNED NOT NULL,
    media_id BIGINT UNSIGNED NOT NULL,
    created_at DATETIME DEFAULT NOW()
    , FOREIGN KEY (user_id) REFERENCES users(id) ON
UPDATE CASCADE ON DELETE restrict
    , FOREIGN KEY (media_id) REFERENCES media(id)
);
INSERT INTO `likes` VALUES
('4','4','1','1991-05-12 20:32:08'),
('5','5','2','1978-09-10 14:36:01'),
('6','6','2','1992-04-15 01:27:31'),
('7','7','2','2003-02-03 04:56:27'),
('8','8','8','2017-04-24 09:30:19'),
('9','9','9','1974-02-07 20:53:55'),
('10','10','10','1973-05-11 03:21:40');

```



2) создаем функцию

```
-- DROP FUNCTION IF EXISTS vk2.fun;
```

```
-- DELIMITER $$
```

```
-- $$
```

```
CREATE FUNCTION vk2.fun(user_id_to_delete BIGINT UNSIGNED)
RETURNS BIGINT UNSIGNED MODIFIES SQL DATA
BEGIN
```

```
-- Удаление из таблицы friend_requests
```

```
DELETE FROM friend_requests WHERE initiator_user_id = user_id_to_delete OR
target_user_id = user_id_to_delete;
```

```
-- Удаление из таблицы messages
```

```
DELETE FROM messages WHERE from_user_id = user_id_to_delete OR to_user_id =
user_id_to_delete;
```

```
-- Удаление из таблицы likes
```

```
DELETE FROM likes WHERE media_id IN (SELECT id FROM media WHERE user_id =
user_id_to_delete);
```

-- Удаление из таблицы media

DELETE FROM media WHERE user_id = user_id_to_delete;

-- Удаление из таблицы profiles

DELETE FROM profiles WHERE user_id = user_id_to_delete;

-- Удаление из таблицы users_communities

DELETE FROM users_communities WHERE user_id = user_id_to_delete;

-- Удаление из таблицы users

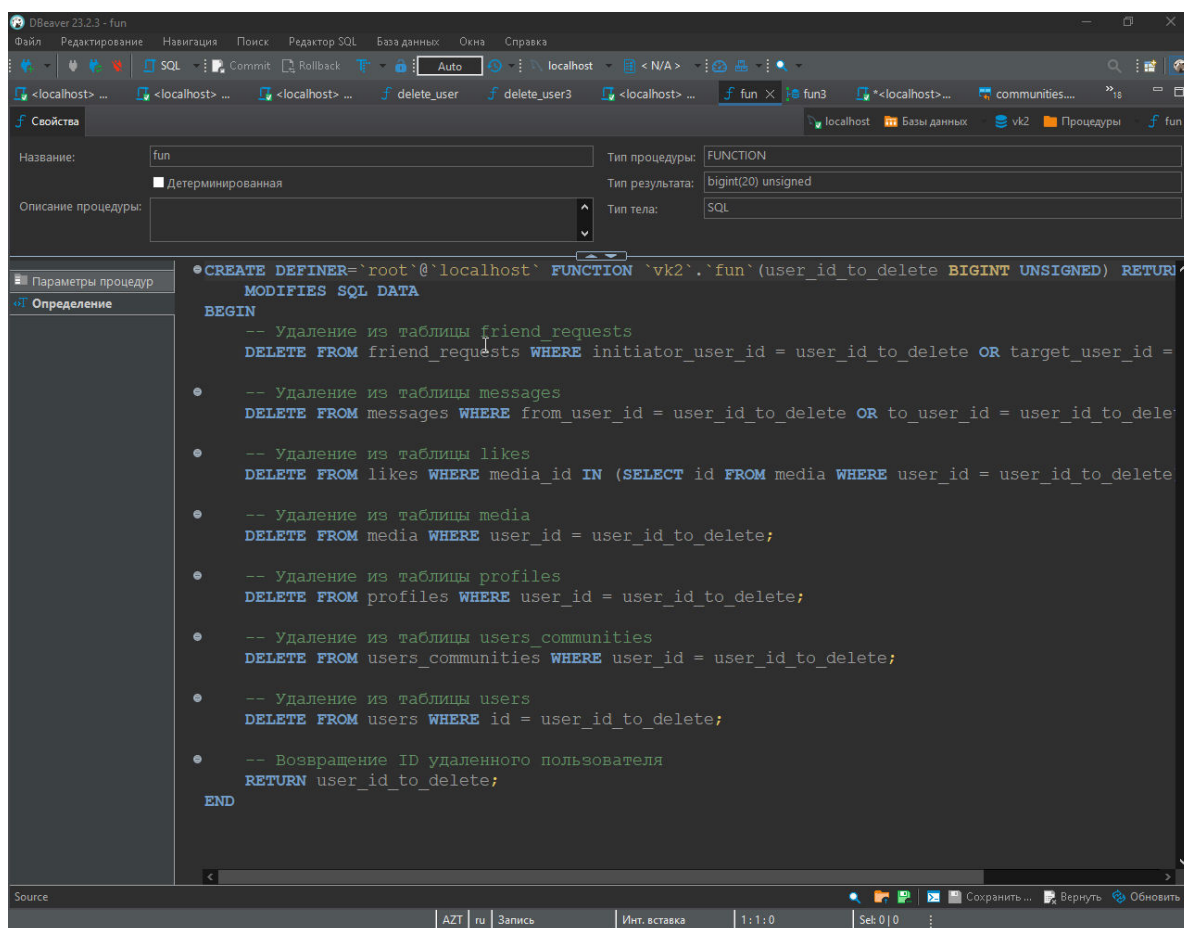
DELETE FROM users WHERE id = user_id_to_delete;

-- Возвращение ID удаленного пользователя

RETURN user_id_to_delete;

END -- \$\$

-- DELIMITER ;



2) Предыдущую задачу решить с помощью процедуры и обернуть используемые команды в транзакцию внутри процедуры.

```
-- DROP PROCEDURE IF EXISTS vk2.fun;

-- DELIMITER $$
-- $$

CREATE PROCEDURE vk2.fun3(user_id_to_delete BIGINT UNSIGNED)
BEGIN
    DECLARE user_exists INT;

    -- Проверка наличия user_id_to_delete в таблице users
    SELECT COUNT(*) INTO user_exists FROM users WHERE id = user_id_to_delete;

    IF user_exists > 0 THEN
        -- Если user_id_to_delete найден, выполняем удаление
        START TRANSACTION;

        -- Удаление из таблицы friend_requests
        DELETE FROM friend_requests WHERE initiator_user_id = user_id_to_delete OR
target_user_id = user_id_to_delete;

        -- Удаление из таблицы messages
        DELETE FROM messages WHERE from_user_id = user_id_to_delete OR to_user_id =
user_id_to_delete;

        -- Удаление из таблицы likes
        DELETE FROM likes WHERE media_id IN (SELECT id FROM media WHERE user_id =
user_id_to_delete);

        -- Удаление из таблицы media
        DELETE FROM media WHERE user_id = user_id_to_delete;

        -- Удаление из таблицы profiles
        DELETE FROM profiles WHERE user_id = user_id_to_delete;

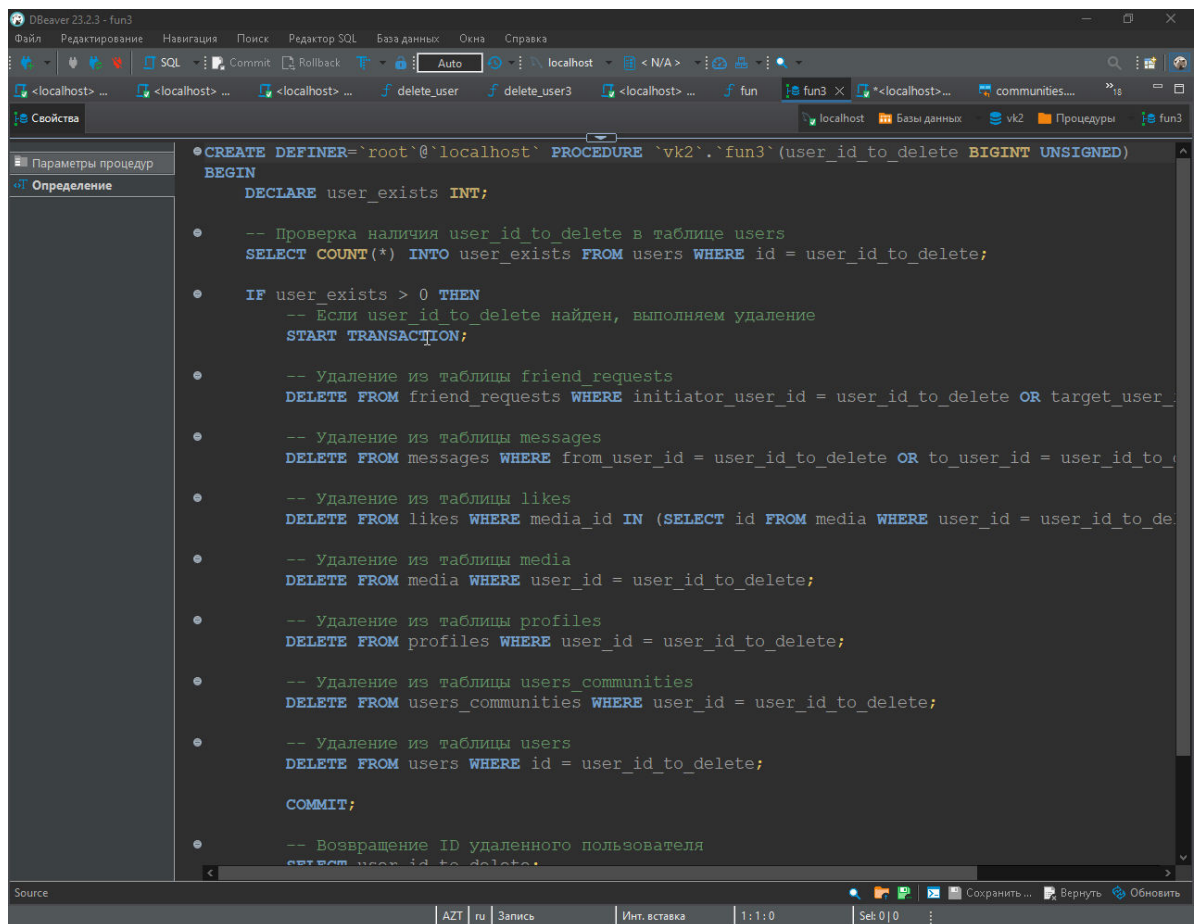
        -- Удаление из таблицы users_communities
        DELETE FROM users_communities WHERE user_id = user_id_to_delete;

        -- Удаление из таблицы users
        DELETE FROM users WHERE id = user_id_to_delete;

        COMMIT;

        -- Возвращение ID удаленного пользователя
        SELECT user_id_to_delete;
    ELSE
        -- Если user_id_to_delete не найден, выдаем сообщение
        SELECT 'ТАКОГО ID НЕТ';
    END IF;
END;
```

```
END IF;  
END -- $$  
-- DELIMITER ;
```



The screenshot shows the DBeaver 23.2.3 SQL editor interface. The main window displays the definition of a stored procedure named 'fun3' in the 'vk2' database. The procedure is created by 'root' at 'localhost' and takes a parameter 'user_id_to_delete' of type 'BIGINT UNSIGNED'. The procedure logic includes a check for the user's existence in the 'users' table, followed by a transaction that deletes records from several tables: 'friend_requests', 'messages', 'likes', 'media', 'profiles', 'users_communities', and 'users'. Finally, it returns the ID of the deleted user.

```
CREATE DEFINER='root'@'localhost' PROCEDURE `vk2`.`fun3`(user_id_to_delete BIGINT UNSIGNED)  
BEGIN  
    DECLARE user_exists INT;  
  
    -- Проверка наличия user id to delete в таблице users  
    SELECT COUNT(*) INTO user_exists FROM users WHERE id = user_id_to_delete;  
  
    IF user_exists > 0 THEN  
        -- Если user_id_to_delete найден, выполняем удаление  
        START TRANSACTION;  
  
        -- Удаление из таблицы friend_requests  
        DELETE FROM friend_requests WHERE initiator_user_id = user_id_to_delete OR target_user_id = user_id_to_delete;  
  
        -- Удаление из таблицы messages  
        DELETE FROM messages WHERE from_user_id = user_id_to_delete OR to_user_id = user_id_to_delete;  
  
        -- Удаление из таблицы likes  
        DELETE FROM likes WHERE media_id IN (SELECT id FROM media WHERE user_id = user_id_to_delete);  
  
        -- Удаление из таблицы media  
        DELETE FROM media WHERE user_id = user_id_to_delete;  
  
        -- Удаление из таблицы profiles  
        DELETE FROM profiles WHERE user_id = user_id_to_delete;  
  
        -- Удаление из таблицы users_communities  
        DELETE FROM users_communities WHERE user_id = user_id_to_delete;  
  
        -- Удаление из таблицы users  
        DELETE FROM users WHERE id = user_id_to_delete;  
  
        COMMIT;  
  
        -- Возвращение ID удаленного пользователя  
        RETURN user_id_to_delete;  
    END IF;  
END
```

The interface includes a top menu bar with options like 'Файл', 'Редактирование', 'Навигация', 'Поиск', 'Редактор SQL', 'База данных', 'Окна', and 'Справка'. Below the menu is a toolbar with icons for various actions. The left sidebar shows a tree view with 'Свойства' and 'Определение'. The bottom status bar displays 'Source', 'AZT ru', 'Запись', 'Инт. вставка', '1:1:0', 'Set 0/0', and buttons for 'Сохранить...', 'Вернуть', and 'Обновить'.

3) * Написать триггер, который проверяет новое появляющееся сообщество. Длина названия сообщества (поле name) должна быть не менее 5 символов. Если требование не выполнено, то выбрасывать исключение с пояснением.

```
use vk2;
```

```
DELIMITER //
```

```
CREATE TRIGGER community_name_length_trigger
BEFORE INSERT ON communities FOR EACH ROW
BEGIN
    IF CHAR_LENGTH(NEW.name) < 5 THEN
        SIGNAL SQLSTATE '45000' SET MESSAGE_TEXT = 'Длина названия сообщества
должна быть не менее 5 символов';
    END IF;
END //
```

```
DELIMITER ;
```

