[Начало работы](#_s4d3vqgz8m2n)

[Практическое задание по теме «Операторы, фильтрация, сортировка и ограничение»](#_3l11jkq62r)

[Практическое задание теме «Агрегация данных»](#_p6juxxo4yszb)

# Начало работы

1. *Настоятельно рекомендуем сдавать практическое задание в виде ссылки на личный репозиторий на Github.*
2. *Рекомендуемый способ организации данных в репозитории: создать отдельные папки по темам и помещать в них отдельные файлы для каждой задачи с правильным расширением. Например*: ***topic3/do\_something.sql***.

*Ссылка на инструкцию по работе с git и сдачу практики:*

*<https://docs.google.com/document/d/1RAT_ukE39iOfbz1xa39QXae2hBUEZ4U6Fko_wFDdrsM/edit>*

Ссылка на видеокурс по Git:

<https://geekbrains.ru/courses/66>

*Если остались сложности, то обратитесь к преподавателю или наставнику.*

1. *Мы для вас сделали* [*образ ОС Ubuntu*](https://drive.google.com/open?id=1Xk6VG9P9I6GAzZujt4Dee6578CtgnIdR) *и записали скринкаст, как установить виртуальную машину + мы добавили базовые методички по линуксу (можно и без них) для ознакомления с системой. Это позволит повторять в командной строке за преподавателем в видеоуроках.*

# Практическое задание по теме «Операторы, фильтрация, сортировка и ограничение»

1. Пусть в таблице users поля created\_at и updated\_at оказались незаполненными. Заполните их текущими датой и временем.

**Создал новые столбцы updated\_at\_copy1 и deleted\_at\_copy1.**

**ЗАПРОСЫ:**

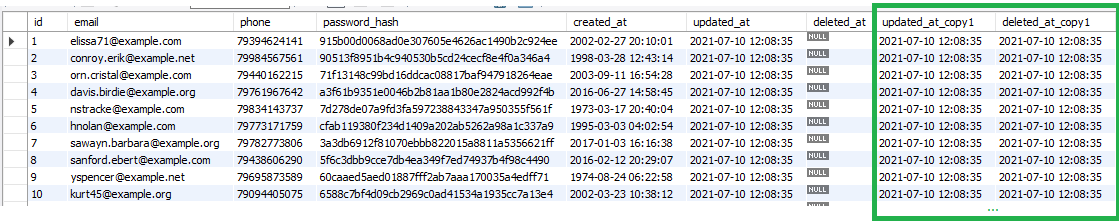
UPDATE `user` SET

updated\_at\_copy1 = NOW(),

deleted\_at\_copy1 = NOW();

SELECT \* FROM vk.user;

**РЕЗУЛЬТАТ:**



1. Таблица users была неудачно спроектирована. Записи created\_at и updated\_at были заданы типом VARCHAR и в них долгое время помещались значения в формате 20.10.2017 8:10. Необходимо преобразовать поля к типу DATETIME, сохранив введённые ранее значения.

**Создал новые столбцы created\_at\_copy1 (с значением по умолчанию - 20.10.2017 8:10) и updated\_at\_copy2 (с значением по умолчанию - 20.10.2017 18:10).**

**ЗАПРОСЫ - РЕЗУЛЬТАТ:**

UPDATE `user` SET

created\_at\_copy1 = '20.10.2017 8:10',

updated\_at\_copy2 = '20.10.2017 18:10';

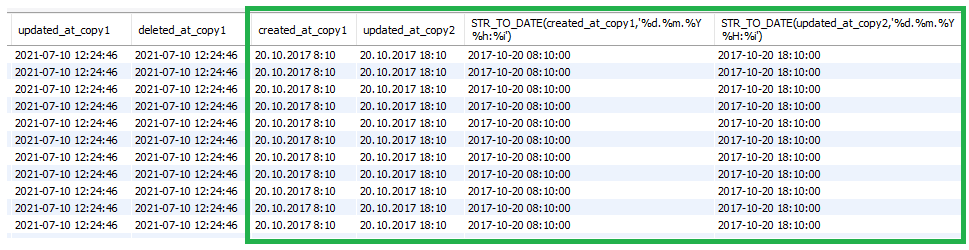
SELECT

\*,

STR\_TO\_DATE(created\_at\_copy1,'%d.%m.%Y %h:%i'),

STR\_TO\_DATE(updated\_at\_copy2,'%d.%m.%Y %H:%i')

FROM vk.user;

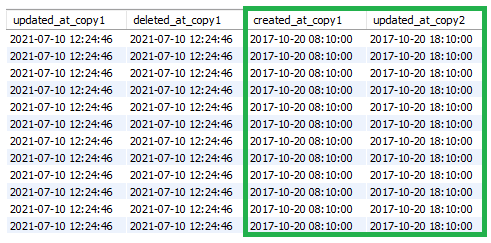


UPDATE `user` SET

created\_at\_copy1 = STR\_TO\_DATE(created\_at\_copy1,'%d.%m.%Y %h:%i'),

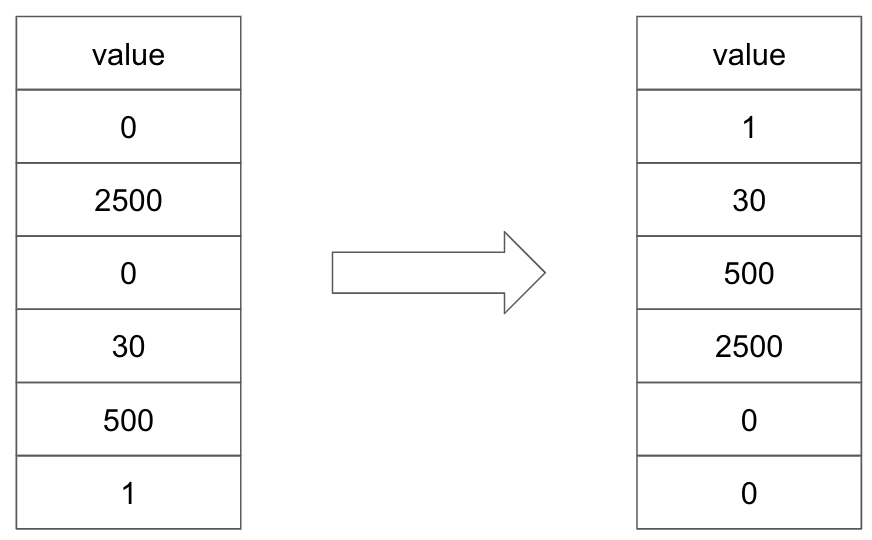
updated\_at\_copy2 = STR\_TO\_DATE(updated\_at\_copy2,'%d.%m.%Y %H:%i');

SELECT \* FROM vk.user;



Далее поменял тип данных у created\_at\_copy1 и updated\_at\_copy2 на DATETIME.

1. В таблице складских запасов storehouses\_products в поле value могут встречаться самые разные цифры: 0, если товар закончился и выше нуля, если на складе имеются запасы. Необходимо отсортировать записи таким образом, чтобы они выводились в порядке увеличения значения value. Однако нулевые запасы должны выводиться в конце, после всех записей.



**Создал новую таблицу - storehouses\_products, со столбцами: id, value.**

**ЗАПРОСЫ – РЕЗУЛЬТАТ:**

INSERT `storehouses\_products` VALUES

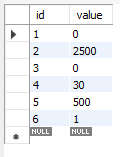
(3, 0),

(4, 30),

(5, 500),

(6, 1);

SELECT \* FROM vk.storehouses\_products;



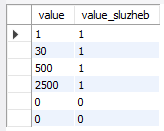
SELECT

value,

IF ( value = 0, 0, 1) AS value\_sluzheb

FROM vk.storehouses\_products

ORDER BY value\_sluzheb DESC, value



1. (по желанию) Из таблицы users необходимо извлечь пользователей, родившихся в августе и мае. Месяцы заданы в виде списка английских названий (may, august)

**ЗАПРОС - РЕЗУЛЬТАТ:**

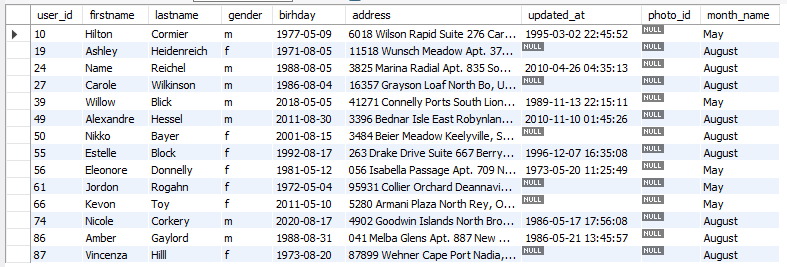
SELECT \*,

MONTHNAME(CONCAT(YEAR(NOW()),RIGHT (birhday, 6))) AS month\_name

FROM vk.profile

HAVING

month\_name IN ('May', 'August');



1. (по желанию) Из таблицы catalogs извлекаются записи при помощи запроса. SELECT \* FROM catalogs WHERE id IN (5, 1, 2); Отсортируйте записи в порядке, заданном в списке IN.

# Практическое задание теме «Агрегация данных»

1. Подсчитайте средний возраст пользователей в таблице users.

**ЗАПРОС - РЕЗУЛЬТАТ:**

SELECT

ROUND(AVG(TIMESTAMPDIFF (YEAR, birhday, NOW())),0) AS avg\_age

FROM vk.profile;



1. Подсчитайте количество дней рождения, которые приходятся на каждый из дней недели. Следует учесть, что необходимы дни недели текущего года, а не года рождения.

**ЗАПРОС - РЕЗУЛЬТАТ:**

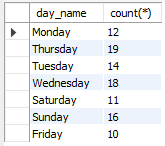
SELECT

DAYNAME(CONCAT(YEAR(NOW()),RIGHT (birhday, 6))) AS day\_name,

COUNT(\*)

FROM vk.profile

GROUP BY (day\_name)



1. (по желанию) Подсчитайте произведение чисел в столбце таблицы.

