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

1 задание. Заполнил в таблице users поля created_at и updated_at, сделал их varchar.

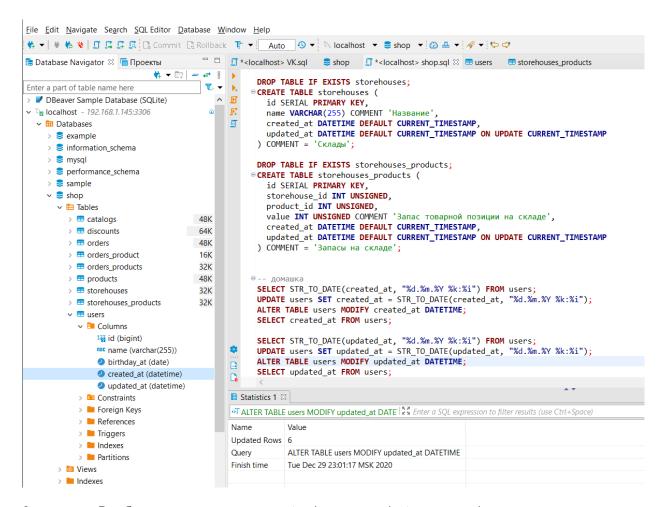
2 задание. Таблица users была неудачно спроектирована. Записи created_at и updated_at были заданы типом VARCHAR и в них долгое время помещались значения в формате 20.10.2017 8:10. Необходимо преобразовать поля к типу DATETIME, сохранив введённые ранее значения.

```
Beaver 7.3.1 - <localhost> shop.sql
<u>File Edit Navigate Search SQL Editor Database Window Help</u>
 ♥ ▼ ♥ ♥ ♥ □ 耳耳耳尾 □ Commit □ Rollback 〒 ▼ Auto ⊙ ▼ □ localhost ▼ ● shop ▼ ② 品 ▼ Ø ▼ ♥ ♥
                                     🥦 Database Navigator 🛭 🫅 Проекты
                                                 DROP TABLE IF EXISTS storehouses;
                                 ₹, ▼
Enter a part of table name here
                                               ○ CREATE TABLE storehouses (
                                       ^ F
 DBeaver Sample Database (SQLite)
                                                   id SERIAL PRIMARY KEY,

√ In localhost - 192.168.1.145:3306

                                                   name VARCHAR(255) COMMENT 'Название
                                                   created_at DATETIME DEFAULT CURRENT_TIMESTAMP,
  v 🛅 Databases
                                                   updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
     > = example
                                                 ) COMMENT = 'Склады';
     > 🛢 information_schema
     > 🛢 mysql
                                                 DROP TABLE IF EXISTS storehouses_products;
     > sperformance_schema
                                               ○ CREATE TABLE storehouses_products (
     > 🛢 sample
                                                   id SERIAL PRIMARY KEY
     v 🛢 shop
                                                   storehouse id INT UNSIGNED,
                                                   product_id INT UNSIGNED,
      v 🗎 Tables
                                                   value INT UNSIGNED COMMENT 'Запас товарной позиции на складе',
         > == catalogs
                                    48K
                                                   created at DATETIME DEFAULT CURRENT_TIMESTAMP,
         > == discounts
                                  64K
                                                   updated_at DATETIME DEFAULT CURRENT_TIMESTAMP ON UPDATE CURRENT_TIMESTAMP
         > == orders
                                   48K
                                                 ) СОММЕНТ = 'Запасы на складе';
         > == orders_product
                                    16K
         > == orders_products
                                    32K
                                   48K
         > == products
                                                 SELECT STR_TO_DATE(created_at, "%d.%m.%Y %k:%i") FROM users;
UPDATE users SET created_at = STR_TO_DATE(created_at, "%d.%m.%Y %k:%i");
                                   32K
         > == storehouses
         storehouses_products
                                    32K
                                                 ALTER TABLE users MODIFY created_at DATETIME;
         🗸 🖽 users
                                                 SELECT created_at FROM users:
           v 🗎 Columns
               12 id (bigint)
                                                 SELECT STR_TO_DATE(updated_at, "%d.%m.%Y %k:%i") FROM users;
UPDATE users SET updated_at = STR_TO_DATE(updated_at, "%d.%m.%Y %k:%i");
ALTER TABLE users MODIFY updated_at DATETIME;
               name (varchar(255))
               birthday_at (date)
                                                 SELECT updated_at FROM users;
               created_at (varchar(255))
               nec updated_at (varchar(255))
                                            ■ Statistics 1 🖂
            > Constraints
            > Foreign Keys
                                          oT INSERT INTO users (name, birthday at, createc | ♣ → Enter a SQL expression to filter results (use Ctrl+Space)
SELECT STR_TO_DATE(created_at, "%d.%m.%Y %k:%i") FROM users;
UPDATE users SET created_at = STR_TO_DATE(created_at, "%d.%m.%Y %k:%i");
ALTER TABLE users MODIFY created_at DATETIME;
SELECT STR_TO_DATE(updated_at, "%d.%m.%Y %k:%i") FROM users;
UPDATE users SET updated_at = STR_TO_DATE(updated_at, "%d.%m.%Y %k:%i");
```

ALTER TABLE users MODIFY updated_at DATETIME;



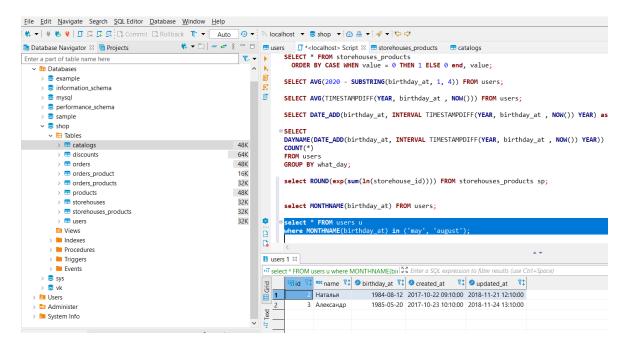
3 задание. В таблице складских запасов storehouses_products в поле value могут встречаться самые разные цифры: 0, если товар закончился и выше нуля, если на складе имеются запасы. Необходимо отсортировать записи таким образом, чтобы они выводились в порядке увеличения значения value. Однако нулевые запасы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

```
Сделал вот так:

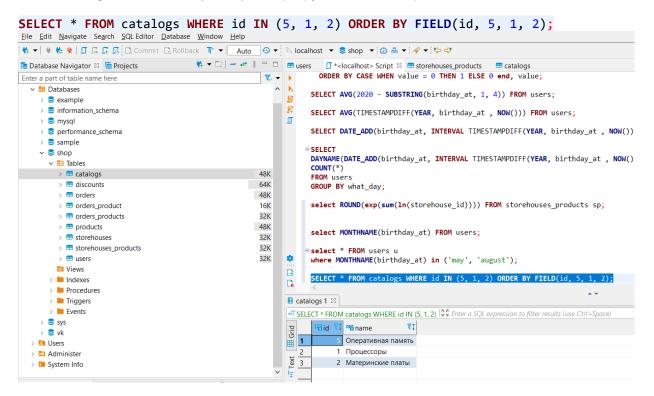
SELECT * FROM storehouses_products

ORDER BY CASE WHEN value = 0 THEN 1 ELSE 0 end, value;
```

4 задание. (по желанию) Из таблицы users необходимо извлечь пользователей, родившихся в августе и мае. Месяцы заданы в виде списка английских названий (may, august) select * FROM users where MONTHNAME(birthday_at) in ('may', 'august');

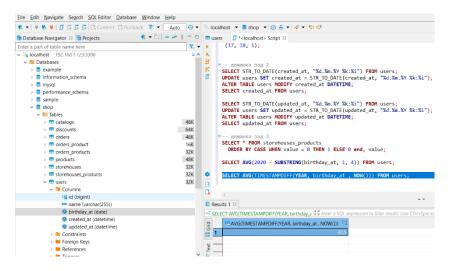


5 задание. (по желанию) Из таблицы catalogs извлекаются записи при помощи запроса. SELECT * FROM catalogs WHERE id IN (5, 1, 2); Отсортируйте записи в порядке, заданном в списке IN



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

1 Задание. Средний возраст пользователей в таблице users посчитал командой: SELECT AVG(TIMESTAMPDIFF(YEAR, birthday_at , NOW())) FROM users;

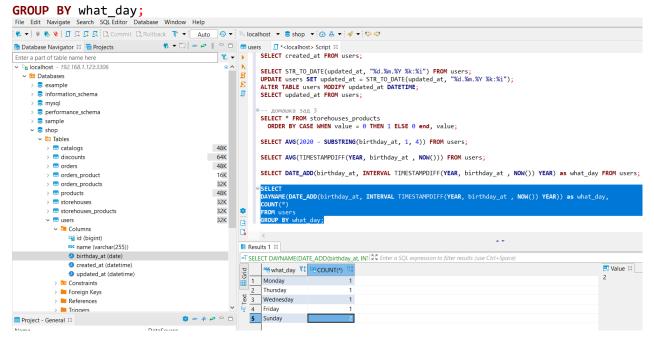


2 задание. Подсчитайте количество дней рождения, которые приходятся на каждый из дней недели. Следует учесть, что необходимы дни недели текущего года, а не года рождения. **SELECT**

DAYNAME(DATE_ADD(birthday_at, INTERVAL TIMESTAMPDIFF(YEAR, birthday_at , NOW())
YEAR)) as what_day,

COUNT(*)

FROM users



3 задание. (по желанию) Подсчитайте произведение чисел в столбце таблицы. Нашел только вот такое решение. Например, посчитал произведение столбца storehouse_id таблицы storehouses_products:

select ROUND(exp(sum(ln(storehouse_id)))) FROM storehouses_products