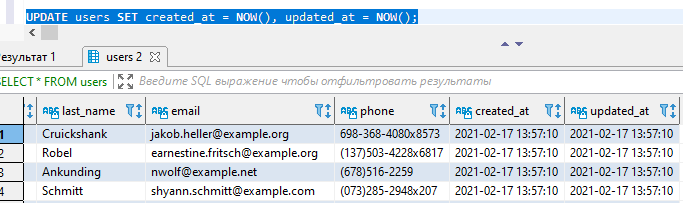
**Практическое задание по теме «Операторы, фильтрация, сортировка и ограничение»**

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

**UPDATE** users **SET** created\_at = **NOW**(), updated\_at = **NOW**();



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.

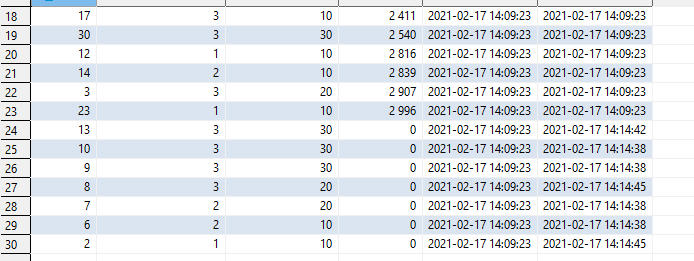
**ALTER** **TABLE** users **MODIFY** created\_at **datetime** **DEFAULT** **CURRENT\_TIMESTAMP**;

**ALTER** **TABLE** users **MODIFY** updated\_at **datetime** **DEFAULT** **CURRENT\_TIMESTAMP** **ON** **UPDATE** **CURRENT\_TIMESTAMP**;

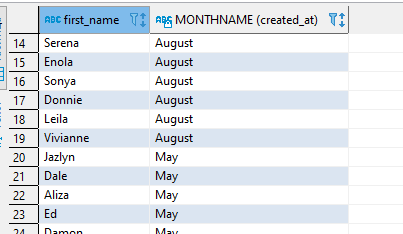


1. В таблице складских запасов storehouses\_products в поле value могут встречаться самые разные цифры: 0, если товар закончился и выше нуля, если на складе имеются запасы. Необходимо отсортировать записи таким образом, чтобы они выводились в порядке увеличения значения value. Однако нулевые запасы должны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

**SELECT** \* **FROM** storehouses\_products **ORDER** **BY** value = 0, value;



1. (по желанию) Из таблицы users необходимо извлечь пользователей, родившихся в августе и мае. Месяцы заданы в виде списка английских названий (may, august)
2. **SELECT** first\_name,
3. **MONTHNAME** (created\_at)
4. **FROM** users
5. **WHERE** **MONTHNAME** (created\_at) = 'may' **or** **MONTHNAME** (created\_at) = 'august'
6. **ORDER** **BY** **MONTHNAME** (created\_at);



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.

**SELECT** \* **FROM** catalogs **WHERE** id **IN** (5, 1, 2) **ORDER** **BY** **FIELD**(id, 5, 1, 2) ;

**Практическое задание теме «Агрегация данных»**

1. Подсчитайте средний возраст пользователей в таблице users.

**SELECT** **AVG**( **FLOOR**((**TO\_DAYS**(**NOW**()) - **TO\_DAYS**(birthday\_at)) / 365.25)) **AS** avg\_age **FROM** users;

1. 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**(**CONCAT**(**YEAR**(**NOW**()), '-', **SUBSTRING**(birthday\_at, 6, 10))) **AS** week\_day, **COUNT**(\*) **AS** birthday

**FROM** users

**GROUP** **BY** week\_day

**ORDER** **BY** birthday;

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