**Практическое задание к Уроку 05**

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

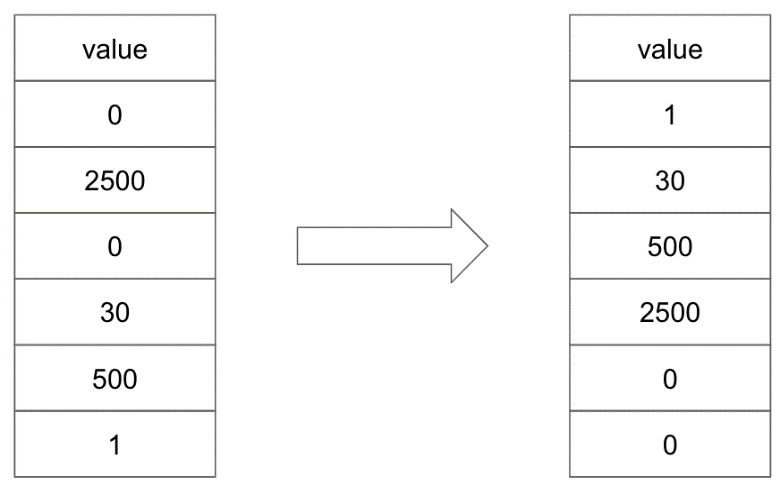
1. Пусть в таблице users поля created\_at и updated\_at оказались незаполненными. Заполните их текущими датой и временем.

|  |
| --- |
| DROP TABLE IF EXISTS users; |
| CREATE TABLE users( |
| id serial primary key, |
| created\_at datetime, |
| updated\_at datetime); |
|  |
| insert |
| into users (created\_at, updated\_at) |
| values |
| (null, null), |
| (null, null), |
| (null, null); |
|  |
| -- меняем NULL на текущее время |
| 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 20.08.2017 20:15. Необходимо преобразовать поля к типу DATETIME, сохранив введённые ранее значения.

|  |
| --- |
| DROP TABLE IF EXISTS users; |
| CREATE TABLE users( |
| id serial primary key, |
| created\_at VARCHAR(255), |
| updated\_at VARCHAR(255) ); |
|  |
| insert |
| into users (created\_at, updated\_at) |
| values |
| ('20.08.2019 20:15', '20. 08.2020 20:15'), |
| ('20. 08.1944 20:15', '20. 08.2011 20:15'), |
| ('20. 08.1955 20:15', '20. 08.2000 20:15'); |
|  |
| -- меняем формат |
| update users |
| set created\_at = str\_to\_date(created\_at, '%d.%m.%Y %k:%i'), |
| updated\_at = str\_to\_date(updated\_at, '%d.%m.%Y %k:%i'); |
|  |
| -- меняем тип данных |
| ALTER TABLE users |
| CHANGE created\_at created\_at DATETIME DEFAULT NOW(); |
|  |
| ALTER TABLE users |
| CHANGE updated\_at updated\_at DATETIME DEFAULT NOW() ON UPDATE NOW(); |

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.



|  |  |
| --- | --- |
|  | DROP TABLE IF EXISTS storehouses\_products; |
|  | CREATE TABLE storehouses\_products( |
|  | id serial PRIMARY KEY, |
|  | value INT UNSIGNED); |
|  |  |
|  | insert |
|  | into storehouses\_products (value) |
|  | values |
|  | (0), |
|  | (123), |
|  | (332), |
|  | (3), |
|  | (12); |
|  | -- сортирую |
|  | select \* from storehouses\_products order by if(value > 0, 0, 1), 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. (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.

**По теме «Агрегация данных»**

1. Подсчитайте средний возраст пользователей в таблице users.

|  |
| --- |
| DROP TABLE IF EXISTS users; |
| CREATE TABLE users( |
| id serial PRIMARY KEY, |
| birthday\_at DATE); |
|  |
| insert |
| into users (birthday\_at) |
| values |
| ('1988-4-4'), |
| ('1899-9-3'), |
| ('2011-11-14'), |
| ('1978-7-28'), |
| ('2005-12-3'), |
| ('2011-7-1'), |
| ('1899-3-20'), |
| ('1999-6-2'), |
| ('1966-6-11'); |
| -- считаем средний возраст |
| select avg(timestampdiff(year, birthday\_at, now())) as mid\_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 date\_format(date(concat\_ws('-', year(now()), month(birthday\_at), day(birthday\_at))), '%W') AS what\_day, |
| count(\*) as total from users |
| group by what\_day; |