

## 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 = CURRENT_TIMESTAMP,  
              updated_at = CURRENT_TIMESTAMP;
```

2. 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 введенные ранее значения.

```
DROP TEMPORARY TABLE IF EXISTS temp_users;  
CREATE TEMPORARY TABLE temp_users(  
  id INT UNSIGNED NOT NULL AUTO_INCREMENT PRIMARY KEY COMMENT "Идентификатор строки",  
  created_at DATETIME COMMENT "Для преобразования VARCHAR to DATETIME"  
);  
  
INSERT INTO temp_users VALUES ( (SELECT id FROM users ORDER BY id LIMIT 1 ), CURRENT_TIMESTAMP );  
  
SELECT * FROM temp_users LIMIT 10;  
  
UPDATE temp_users SET created_at = STR_TO_DATE(  
  (SELECT created_at_2_change FROM users ORDER BY id LIMIT 1 ), '%d.%m.%Y %r' );  
  
UPDATE users SET created_at_aftrter_change = (SELECT created_at FROM temp_users ORDER BY id LIMIT 1 );  
  
ALTER TABLE users RENAME COLUMN created_at_aftrter_change TO created_at_1;
```

3. 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.

value	value
0	1
2500	30
0	500
30	2500
500	0
1	0

```
DROP TEMPORARY TABLE IF EXISTS store_prices;  
  
CREATE TEMPORARY TABLE store_prices(  
  id INT UNSIGNED AUTO_INCREMENT NOT NULL,  
  value INT UNSIGNED NOT NULL,  
  catalog_id BOOL AS (value = 0),  
  PRIMARY KEY (id)  
);
```

```
INSERT INTO store_prices(value) VALUES
(0),(2500),(0),(30),(500),(1);
```

```
SELECT value FROM store_prices ORDER BY catalog_id, value;
```

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

```
CREATE TEMPORARY TABLE months(
  id INT UNSIGNED AUTO_INCREMENT NOT NULL,
  name VARCHAR(20) NOT NULL UNIQUE,
  PRIMARY KEY (id)
);

INSERT INTO months(name) VALUES
('январь'),('февраль'),('март'),('апрель'),('май'),('июнь'),
('июль'),('август'),('сентябрь'),('октябрь'),('ноябрь'),('декабрь');

-- РАБОТАЕТ --
SELECT id FROM months WHERE name RLIKE 'февраль|май' ORDER BY id;

-- РАБОТАЕТ --
SELECT * FROM users WHERE created_at RLIKE CONCAT( '^[[[:digit:]]{4}-[0, ',
  (SELECT id FROM months WHERE name RLIKE 'февраль',
  ']{2}-[[[:digit:]]{2}');

-- НЕ РАБОТАЕТ -- SQL Error [1242] [21000]: Subquery returns more than 1 row
SELECT * FROM users WHERE created_at RLIKE CONCAT( '^[[[:digit:]]{4}-[0, ',
  (SELECT id FROM months WHERE name RLIKE 'февраль|май',
  ']{2}-[[[:digit:]]{2}');
```

```
SELECT * FROM users WHERE DATE_FORMAT(created_at, '%M') IN ('may', 'june');
```

5. (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 users WHERE id IN (5, 1, 2) ORDER BY FIELD(id, 5, 1, 2);
```

## Практическое задание теме “Агрегация данных”

- Подсчитайте средний возраст пользователей в таблице users  
`SELECT AVG ( YEAR(NOW())-YEAR(birthday_at) ) FROM users;`
- Подсчитайте количество дней рождения, которые приходятся на каждый из дней недели. Следует учесть, что необходимы дни недели текущего года, а не года рождения.

Вычислить число дней рождений для конкретного дня недели (суббота), можно так:

```
SELECT COUNT( DAYOFWEEK( DATE_FORMAT(created_at, '2020-%m-%d') ) ) FROM users WHERE
DAYOFWEEK( DATE_FORMAT(created_at, '2020-%m-%d') ) = 7;
```

после просмотра стало понятно

```
SELECT DATE_FORMAT(created_at, '2020-%m-%d') FROM users;

SELECT DATE_FORMAT(created_at, '2020-%m-%d') AS birth_date FROM users ;

SELECT DAYOFWEEK(DATE_FORMAT(created_at, '2020-%m-%d')) AS dy FROM users GROUP BY dy;

SELECT DAYOFWEEK(DATE_FORMAT(created_at, '2020-%m-%d')) AS dy, COUNT(*) total FROM users GROUP BY dy;

SELECT DAYNAME(DATE_FORMAT(created_at, '2020-%m-%d')) AS dy, COUNT(*) total FROM users GROUP BY dy;
```

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

value
1
2
3
4
5



120