

Практическое задание №5

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.

```
mysql> SELECT * FROM users limit 2;
+----+-----+-----+-----+-----+
| id | name   | birthday_at | created_at | updated_at |
+----+-----+-----+-----+-----+
| 1  | Геннадий | 1990-10-05 | NULL      | NULL      |
| 2  | Наталья | 1984-11-12 | NULL      | NULL      |
+----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> UPDATE users SET created_at = now(), updated_at = now();
Query OK, 6 rows affected (0.01 sec)
Rows matched: 6  Changed: 6  Warnings: 0

mysql> SELECT * FROM users limit 2;
+----+-----+-----+-----+-----+
| id | name   | birthday_at | created_at          | updated_at          |
+----+-----+-----+-----+-----+
| 1  | Геннадий | 1990-10-05 | 2021-06-16 19:35:07 | 2021-06-16 19:35:07 |
| 2  | Наталья | 1984-11-12 | 2021-06-16 19:35:07 | 2021-06-16 19:35:07 |
+----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

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.

```
mysql> DESCRIBE users;
+-----+-----+-----+-----+-----+-----+
| Field      | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id         | bigint unsigned | NO   | PRI | NULL    | auto_increment |
| name       | varchar(255)   | YES  |     | NULL    |                |
| birthday_at | date          | YES  |     | NULL    |                |
| created_at | varchar(255)   | YES  |     | NULL    |                |
| updated_at | varchar(255)   | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
5 rows in set (0.00 sec)

mysql> INSERT INTO users (name, birthday_at, created_at, updated_at) VALUES ('Геннадий', '1990-10-05', '20.10.2017 8:10', '20.10.2017 8:10'), ('Наталья', '1984-11-12', '20.10.2018 8:10', '20.10.2018 8:10');
Query OK, 2 rows affected (0.01 sec)
Records: 2  Duplicates: 0  Warnings: 0

mysql> SELECT * FROM users;
+----+-----+-----+-----+-----+
| id | name   | birthday_at | created_at          | updated_at          |
+----+-----+-----+-----+-----+
| 1  | Геннадий | 1990-10-05 | 20.10.2017 8:10    | 20.10.2017 8:10    |
| 2  | Наталья | 1984-11-12 | 20.10.2018 8:10    | 20.10.2018 8:10    |
+----+-----+-----+-----+-----+
2 rows in set (0.00 sec)

mysql> ALTER TABLE users ADD created_at_new DATETIME;
Query OK, 0 rows affected (0.02 sec)
Records: 0  Duplicates: 0  Warnings: 0
```

```
mysql> ALTER TABLE users ADD updated_at_new DATETIME;
Query OK, 0 rows affected (0.02 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> DESCRIBE users;
+-----+-----+-----+-----+-----+-----+
| Field          | Type          | Null | Key | Default | Extra          |
+-----+-----+-----+-----+-----+-----+
| id             | bigint unsigned | NO   | PRI | NULL    | auto_increment |
| name          | varchar(255)   | YES  |     | NULL    |                |
| birthday_at   | date           | YES  |     | NULL    |                |
| created_at     | varchar(255)   | YES  |     | NULL    |                |
| updated_at     | varchar(255)   | YES  |     | NULL    |                |
| created_at_new | datetime       | YES  |     | NULL    |                |
| updated_at_new | datetime       | YES  |     | NULL    |                |
+-----+-----+-----+-----+-----+-----+
7 rows in set (0.00 sec)

mysql> UPDATE users SET created_at_new = STR_TO_DATE(created_at, '%d.%m.%Y %h:%i'), updated_at_new = STR_TO_DATE(updated_at, '%d.%m.%Y %h:%i');
Query OK, 2 rows affected (0.01 sec)
Rows matched: 2 Changed: 2 Warnings: 0

mysql> ALTER TABLE users DROP created_at, DROP updated_at, RENAME COLUMN created_at_new TO created_at, RENAME COLUMN updated_at_new TO updated_at;
Query OK, 0 rows affected (0.04 sec)
Records: 0 Duplicates: 0 Warnings: 0

mysql> SELECT * FROM users;
+-----+-----+-----+-----+-----+
| id | name      | birthday_at | created_at          | updated_at          |
+-----+-----+-----+-----+-----+
| 1  | Геннадий | 1990-10-05  | 2017-10-20 08:10:00 | 2017-10-20 08:10:00 |
| 2  | Наталья  | 1984-11-12  | 2018-10-20 08:10:00 | 2018-10-20 08:10:00 |
+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

3) Задание.

```
mysql> INSERT INTO storehouses_products (value) VALUES (0), (2500), (0), (30), (500), (1);
Query OK, 6 rows affected (0.00 sec)
Records: 6 Duplicates: 0 Warnings: 0

mysql> SELECT value FROM storehouses_products;
+-----+
| value |
+-----+
| 0      |
| 2500   |
| 0      |
| 30     |
| 500    |
| 1      |
+-----+
6 rows in set (0.00 sec)

mysql> SELECT value FROM storehouses_products ORDER BY CASE WHEN value = 0 then 1 else 0 end, value;
+-----+
| value |
+-----+
| 1      |
| 30     |
| 500    |
| 2500   |
| 0      |
| 0      |
+-----+
6 rows in set (0.00 sec)
```

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

```
mysql> SELECT name, birthday_at FROM users;
+-----+-----+
| name   | birthday_at |
+-----+-----+
| Геннадий | 1990-08-05 |
| Наталья | 1984-05-05 |
| Сергей  | 1988-02-14 |
| Иван    | 1998-01-12 |
| Мария   | 1992-08-29 |
+-----+-----+
5 rows in set (0.00 sec)

mysql> SELECT name, birthday_at FROM users WHERE MONTHNAME(birthday_at) IN ('may', 'august');
+-----+-----+
| name   | birthday_at |
+-----+-----+
| Геннадий | 1990-08-05 |
| Наталья | 1984-05-05 |
| Мария   | 1992-08-29 |
+-----+-----+
3 rows in set (0.00 sec)
```

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.

```
mysql> SELECT * FROM catalogs WHERE id IN (5, 1, 2);
+----+-----+
| id | name           |
+----+-----+
| 1  | Процессоры    |
| 2  | Материнские платы |
| 5  | Оперативная память |
+----+-----+
3 rows in set (0.00 sec)

mysql> SELECT * FROM catalogs WHERE id IN (5, 1, 2) ORDER BY CASE WHEN id = 5 THEN 0 WHEN id = 1
THEN 1 WHEN id = 2 THEN 2 END;
+----+-----+
| id | name           |
+----+-----+
| 5  | Оперативная память |
| 1  | Процессоры    |
| 2  | Материнские платы |
+----+-----+
3 rows in set (0.00 sec)
```

Практическое задание теме «Агрегация данных»

1) Подсчитайте средний возраст пользователей в таблице users.

```
mysql> SELECT name, timestampdiff(YEAR, birthday_at, NOW()) AS age FROM users;
+-----+-----+
| name   | age |
+-----+-----+
| Геннадий | 30 |
| Наталья | 37 |
| Сергей  | 33 |
| Иван    | 23 |
| Мария   | 28 |
+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> SELECT round(avg(timestampdiff(YEAR, birthday_at, NOW()))), 0) AS average_age FROM users;
+-----+
| average_age |
+-----+
|          30 |
+-----+
1 row in set (0.00 sec)
```

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

```
mysql> SELECT name, birthday_at FROM users;
+-----+-----+
| name   | birthday_at |
+-----+-----+
| Геннадий | 1990-08-05 |
| Наталья | 1984-05-05 |
| Сергей  | 1988-02-14 |
| Иван    | 1998-01-12 |
| Мария   | 1992-08-29 |
+-----+-----+
5 rows in set (0.00 sec)
```

```
mysql> SELECT DATE_FORMAT(DATE(CONCAT_WS('-', YEAR(NOW()), MONTH(birthday_at), DAY(birthday_at))), '%W') AS day, COUNT(*) AS total FROM users GROUP BY DAY ORDER BY total DESC;
+-----+-----+
| day       | total |
+-----+-----+
| Sunday    | 2 |
| Thursday  | 1 |
| Wednesday | 1 |
| Tuesday   | 1 |
+-----+-----+
4 rows in set (0.00 sec)
```

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