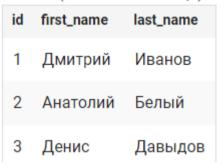
Создайте таблицу users с полем id типа INT и двумя текстовыми полями, которые будут хранить имя (first\_name) и фамилию (last\_name). Длина имени и фамилии не превышает 50 символов.

Добавьте в таблицу трех пользователей: Дмитрия Иванова, Анатолия Белого и Дениса Давыдова.

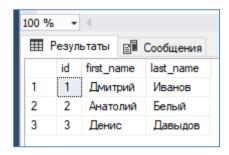
users (итоговая таблица)



```
CREATE table users
(id INT, first_name nvarchar (50), last_name nvarchar (50))

INSERT INTO users (id, first_name, last_name)

VALUES (1,'Дмитрий', 'Иванов'),
(2, 'Анатолий', 'Белый'),
(3, 'Денис', 'Давыдов');
```



Создайте таблицу orders с полем id типа INT, полем state для хранения статуса заказа и полем amount для хранения суммы заказа. Статус заказа умещается в строку в 10 символов, а сумма заказа имеет целочисленный тип данных.

Заполните таблицу тремя заказами: со статусом new на сумму 10000 рублей; со статусом new на сумму 3400 рублей; со статусом delivery на сумму 7300 рублей. оrders (итоговая

таблица)

| id | state    | amount |
|----|----------|--------|
| 1  | new      | 10000  |
| 2  | new      | 3400   |
| 3  | delivery | 7300   |

```
CREATE table orders (id INT, state varchar (10), amount INT);
INSERT INTO orders (id , state, amount)
VALUES
(1, 'new', 10000),
(2, 'new', 3400),
(3, 'delivery', 7300);
```

| Number of Records: 3 |          |        |  |  |
|----------------------|----------|--------|--|--|
| id                   | state    | amount |  |  |
| 1                    | new      | 10000  |  |  |
| 2                    | new      | 3400   |  |  |
| 3                    | delivery | 7300   |  |  |
|                      |          |        |  |  |

Создайте таблицу users с полем id типа INT и двумя строковыми полями, которые будут хранить имя (first\_name) и фамилию (last\_name). Длина имени не превышает 20 символов, а фамилии 50 символов. Также добавьте в таблицу поле birthday типа DATE.

Заполните таблицу тремя пользователями:

Дмитрий Иванов, 12 августа 1995

Светлана Демчук, 8 июля 1993

Денис Антонов, 23 декабря 1996

users (итоговая таблица)

| id | first_name | last_name | birthday   |
|----|------------|-----------|------------|
| 1  | Дмитрий    | Иванов    | 1995-08-12 |
| 2  | Светлана   | Демчук    | 1993-07-08 |
| 3  | Денис      | Антонов   | 1996-12-23 |

```
CREATE table users (id INT, first_name nvarchar (20), last_name nvarchar (50))
ALTER TABLE users ADD birthday date;
INSERT INTO users (id, first_name, last_name, birthday)
VALUES
(1,'Дмитрий', 'Иванов','1995-08-12'),
(2, Светлана', 'Демчук', '1993-07-08'),
(3, 'Денис', 'Антонов', '1996-12-23');
```

| Number of Records: 3 |            |           |            |  |
|----------------------|------------|-----------|------------|--|
| id                   | first_name | last_name | birthday   |  |
| 1                    | Дмитрий    | Иванов    | 1995-08-12 |  |
| 2                    | Светлана   | Демчук    | 1993-07-08 |  |
| 3                    | Денис      | Антонов   | 1996-12-23 |  |
|                      |            |           |            |  |

Создайте таблицу messages с со следующими полями:

id типа INT;

subject типа VARCHAR длиной 100 символов

message тип TEXT;

add\_date типа DATETIME;

is public типа INT.

Добавьте в таблицу сообщение с темой «Первое сообщение» и текстом «Это мое первое сообщение!». Дату установите 12 декабря 2016 года 14 часов, 16 минут. Сообщение должно быть публичным.

messages (итоговая таблица)

| i<br>d | subject             | message                      | add_date               | is_publi<br>c |
|--------|---------------------|------------------------------|------------------------|---------------|
| 1      | Первое<br>сообщение | Это мое первое<br>сообщение! | 2016-12-12<br>14:16:00 | 1             |

#### Решение.

CREATE table messages (id INT, subject varchar(100), message text, add date datetime, is public INT);

```
INSERT INTO messages (id, subject, message, add_date, is_public)
VALUES
```

(1, 'Первое сообщение', 'Это моё первое сообщение!', '2016-12-12 14:16:00', 1)

| id s | ubject             | message                         | add_date                   | is_public |
|------|--------------------|---------------------------------|----------------------------|-----------|
|      | Іервое<br>ообщение | Это моё<br>первое<br>сообщение! | 2016-12-<br>12<br>14:16:00 | 1         |

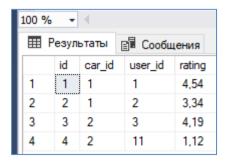
Создайте таблицу rating id типа INT; car\_id типа INT; user\_id типа INT; rating типа FLOAT.

Добавьте в неё 4 записи так, чтобы получалась таблица ниже:

rating

| id | car_id | user_id | rating |
|----|--------|---------|--------|
| 1  | 1      | 1       | 4.54   |
| 2  | 1      | 2       | 3.34   |
| 3  | 2      | 3       | 4.19   |
| 4  | 2      | 11      | 1.12   |

```
CREATE table rating (id INT, car_id INT, user_id INT, rating FLOAT);
INSERT INTO rating (id, car_id, user_id, rating)
VALUES (1, 1, 1, 4.54),
(2, 1, 2, 3.34),
(3, 2, 3, 4.19),
(4, 2, 11, 1.12);
```



В конце каждого номера нужно вывести получившуюся таблицу на экран и заскринить.

### Номер 1

В таблицу products внесли данные с ошибкой, вместо iMac в наименовании написали IMAC. Исправьте ошибку.

products (исходная таблица)

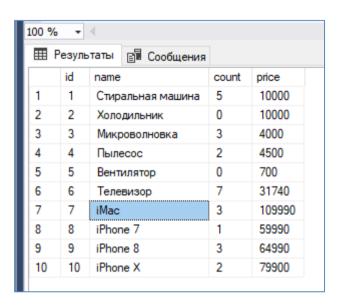
| id | name              | count | price  |
|----|-------------------|-------|--------|
| 1  | Стиральная машина | 5     | 10000  |
| 2  | Холодильник       | 0     | 10000  |
| 3  | Микроволновка     | 3     | 4000   |
| 4  | Пылесос           | 2     | 4500   |
| 5  | Вентилятор        | 0     | 700    |
| 6  | Телевизор         | 7     | 31740  |
| 7  | IMAC              | 3     | 109990 |
| 8  | iPhone 7          | 1     | 59990  |
| 9  | iPhone 8          | 3     | 64990  |
| 10 | iPhone X          | 2     | 79900  |

```
CREATE TABLE products (
   id INT NOT NULL PRIMARY KEY ,
   name NVARCHAR(255) NULL,
   count INTEGER NULL,
   price INTEGER NULL
);
INSERT INTO products (id, name, count, price
VALUES
```

- (1, 'Стиральная машина', 5, 10000),
- (2, 'Холодильник', 0, 10000)
- (3, 'Микроволновка', 3, 4000)
- (4, 'Пылесос', 2, 4500),
- (5, 'Вентилятор', 0, 700)
- (6, 'Телевизор', 7, 31740)
- (7, 'IMAC', 3, 109990),

```
(8, 'iPhone 7', 1, 59990),
(9, 'iPhone 8', 3, 64990),
(10, 'iPhone X', 2, 79900);
```

```
UPDATE products
SET name = 'iMac'
WHERE name = 'Imac'
```



Увеличьте в таблице users сотрудникам, у которых зарплата менее 20 000 рублей, зарплату (salary) на 10%.

users (исходная таблица)

| id | first_name | last_name    | birthday   | salary |
|----|------------|--------------|------------|--------|
| 1  | Дмитрий    | Петров       | 2000-03-14 | 17000  |
| 2  | Ольга      | Антонова     | 1999-12-01 | 20000  |
| 3  | Сергей     | Васильев     | 2002-02-20 | 15000  |
| 4  | Константин | Степаниденко | None       | 35000  |
| 5  | Алена      | Шикова       | 1999-08-17 | 40000  |
| 6  | Василина   | Парамонова   | 2000-10-10 | 18500  |
| 7  | Александр  | Пузаков      | 2002-02-20 | 21000  |
| 8  | Алина      | Антонова     | 2002-01-01 | 55000  |
| 9  | Антон      | Пепеляев     | 1992-07-12 | 60000  |

```
CREATE TABLE users (
    id INT NOT NULL PRIMARY KEY ,
    first_name NVARCHAR(50) NULL,
    last_name NVARCHAR(50) NULL,
    birthday DATE NULL,
    salary INTEGER NULL
);
INSERT INTO users (id, first_name, last_name, birthday, salary)
VALUES
    (1, 'Дмитрий', 'Петров', '2000-03-14', 17000),
    (2, 'Ольга', 'Антонова', '1999-12-01', 20000),
    (3, 'Сергей', 'Васильев', '2002-02-20', 15000),
    (4, 'Константин', 'Степаниденко', NULL, 35000),
    (5, 'Алена', 'Шикова', '1999-08-17', 40000),
    (6, 'Василина', 'Парамонова', '2000-10-10', 18500),
    (7, 'Александр', 'Пузаков', '2002-02-20', 21000),
```

```
UPDATE users
SET salary = salary + (salary/100*10)
WHERE salary < 20000</pre>
```

| Number of Records: 9 |            |              |            |        |
|----------------------|------------|--------------|------------|--------|
| id                   | first_name | last_name    | birthday   | salary |
| 1                    | Дмитрий    | Петров       | 2000-03-14 | 18700  |
| 2                    | Ольга      | Антонова     | 1999-12-01 | 20000  |
| 3                    | Сергей     | Васильев     | 2002-02-20 | 16500  |
| 4                    | Константин | Степаниденко | null       | 35000  |
| 5                    | Алена      | Шикова       | 1999-08-17 | 40000  |
| 6                    | Василина   | Парамонова   | 2000-10-10 | 20350  |
| 7                    | Александр  | Пузаков      | 2002-02-20 | 21000  |
| 8                    | Алина      | Антонова     | 2002-01-01 | 55000  |
| 9                    | Антон      | Пепеляев     | 1992-07-12 | 60000  |
|                      |            |              |            |        |

Проставьте всем заказам без статуса (status paвен NULL) статус "new". NULL – это особое слово в MySQL и в отличии от "cancelled" или "new", его нужно писать без кавычек.

А чтобы сравнить значение в поле с NULL, нужно использовать не символы равенства (=) и неравенства (<>), а специальное выражение IS NULL или IS NOT NULL.

orders (исходная таблица)

| id | user_id | products_count | sum   | status      |
|----|---------|----------------|-------|-------------|
| 1  | 1       | 2              | 1300  | new         |
| 2  | 18      | 1              | 10000 | cancelled   |
| 3  | 11      | 1              | 2140  | in_progress |
| 4  | 145     | 5              | 6800  | NULL        |
| 5  | 23      | 1              | 999   | new         |
| 6  | 1       | 2              | 7690  | cancelled   |
| 7  | 17      | 1              | 1600  | NULL        |
| 8  | 5       | 4              | 400   | delivery    |
| 9  | 2355    | 1              | 1450  | NULL        |
| 10 | 13      | 7              | 13000 | cancelled   |

```
CREATE TABLE orders (
   id INT NOT NULL PRIMARY KEY ,
   user_id INTEGER NULL,
   products_count INTEGER NULL,
   sum INTEGER NULL,
   status NVARCHAR(20) NULL
);
INSERT INTO orders (id, user_id, products_count, sum, status)
VALUES
   (1, 1, 2, 1300, 'new'),
   (2, 18, 1, 10000, 'cancelled'),
   (3, 11, 1, 2140, 'in_progress'),
   (4, 145, 5, 6800, NULL),
   (5, 23, 1, 999, 'new'),
   (6, 1, 2, 7690, 'cancelled'),
```

(7, 17, 1, 1600, NULL), (8, 5, 4, 400, 'delivery'), (9, 2355, 1, 1450, NULL), (10, 13, 7, 13000, 'cancelled');

# Решение.

UPDATE orders
SET status = 'new'
WHERE status IS NULL

| III F |    |         |                |       |             |
|-------|----|---------|----------------|-------|-------------|
|       | id | user_id | products_count | sum   | status      |
| 1     | 1  | 1       | 2              | 1300  | new         |
| 2     | 2  | 18      | 1              | 10000 | cancelled   |
| 3     | 3  | 11      | 1              | 2140  | in_progress |
| 4     | 4  | 145     | 5              | 6800  | new         |
| 5     | 5  | 23      | 1              | 999   | new         |
| 6     | 6  | 1       | 2              | 7690  | cancelled   |
| 7     | 7  | 17      | 1              | 1600  | new         |
| 8     | 8  | 5       | 4              | 400   | delivery    |
| 9     | 9  | 2355    | 1              | 1450  | new         |
| 10    | 10 | 13      | 7              | 13000 | cancelled   |

В поле amount в таблице orders должно стоять число, которое равно произведению цены (sum) на количество (products\_count). Но из-за сбоя в системе некоторые значения суммы получили 0 или NULL. Обновите таблицу, чтобы в поле amount были правильные значения.

orders (исходная таблица)

| id | user_id | products_count | sum   | status      | amount |
|----|---------|----------------|-------|-------------|--------|
| 1  | 1       | 2              | 1300  | new         | 2600   |
| 2  | 18      | 1              | 10000 | cancelled   | 0      |
| 3  | 11      | 1              | 2140  | in_progress | NULL   |
| 4  | 145     | 5              | 6800  | new         | 34000  |
| 5  | 23      | 1              | 999   | new         | 999    |
| 6  | 1       | 2              | 7690  | cancelled   | NULL   |
| 7  | 17      | 1              | 1600  | new         | 0      |
| 8  | 5       | 4              | 400   | delivery    | NULL   |
| 9  | 2355    | 1              | 1450  | new         | 1450   |
| 10 | 13      | 7              | 13000 | cancelled   | 91000  |

```
CREATE TABLE orders (
   id INT NOT NULL PRIMARY KEY ,
   user_id INTEGER NULL,
   products_count INTEGER NULL,
   sum INTEGER NULL,
   status NVARCHAR(20) NULL,
   amount INTEGER NULL
);
INSERT INTO orders (id, user_id, products_count, sum, status, amount)
VALUES
   (1, 1, 2, 1300, 'new', 2600),
   (2, 18, 1, 10000, 'cancelled', 0),
   (3, 11, 1, 2140, 'in_progress', NULL),
   (4, 145, 5, 6800, 'new', 34000),
   (5, 23, 1, 999, 'new', 999),
```

```
(6, 1, 2, 7690, 'cancelled', NULL),
(7, 17, 1, 1600, 'new', 0),
(8, 5, 4, 400, 'delivery', NULL),
(9, 2355, 1, 1450, 'new', 1450),
(10, 13, 7, 13000, 'cancelled', 91000);
```

```
UPDATE orders
SET amount = sum * products_count
WHERE amount IS NULL
OR amount = 0
```

| umb | er of Recor | ds: 10         |       |             |        |
|-----|-------------|----------------|-------|-------------|--------|
| id  | user_id     | products_count | sum   | status      | amount |
| 1   | 1           | 2              | 1300  | new         | 2600   |
| 2   | 18          | 1              | 10000 | cancelled   | 10000  |
| 3   | 11          | 1              | 2140  | in_progress | 2140   |
| 4   | 145         | 5              | 6800  | new         | 34000  |
| 5   | 23          | 1              | 999   | new         | 999    |
| 6   | 1           | 2              | 7690  | cancelled   | 15380  |
| 7   | 17          | 1              | 1600  | new         | 1600   |
| 8   | 5           | 4              | 400   | delivery    | 1600   |
| 9   | 2355        | 1              | 1450  | new         | 1450   |
| 10  | 13          | 7              | 13000 | cancelled   | 91000  |

Номер 5

Измените статус (status) заказа под номером (id) 5 с delivery на success.

#### orders (исходная таблица)

| id | user_id | products_count | sum   | status      | amount |
|----|---------|----------------|-------|-------------|--------|
| 1  | 1       | 2              | 1300  | new         | 2600   |
| 2  | 18      | 1              | 10000 | cancelled   | 10000  |
| 3  | 11      | 1              | 2140  | in_progress | 2140   |
| 4  | 145     | 5              | 6800  | new         | 34000  |
| 5  | 23      | 1              | 999   | delivery    | 999    |
| 6  | 1       | 2              | 7690  | cancelled   | 15380  |
| 7  | 17      | 1              | 1600  | new         | 1600   |
| 8  | 5       | 4              | 400   | delivery    | 1600   |
| 9  | 2355    | 1              | 1450  | new         | 1450   |
| 10 | 13      | 7              | 13000 | cancelled   | 91000  |

```
CREATE TABLE orders (
  id INT NOT NULL,
  user_id INT NOT NULL,
  products_count INT,
  sum INT,
  status NVARCHAR(20),
  amount INT NULL
INSERT INTO orders (id, user_id, products_count, sum, status, amount)
VALUES
(1, 1, 2, 1300, 'new', 2600),
(2, 18, 1, 10000, 'cancelled', 10000),
(3, 11, 1, 2140, 'in_progress', 2140),
(4, 145, 5, 6800, 'new', 34000),
(5, 23, 1, 999, 'delivery', 999),
(6, 1, 2, 7690, 'cancelled', 15380),
(7, 17, 1, 1600, 'new', 1600),
(8, 5, 4, 400, 'delivery', 1600),
```

```
(9, 2355, 1, 1450, 'new', 1450),
(10, 13, 7, 13000, 'cancelled', 91000);
```

```
UPDATE orders
SET status = 'success'
WHERE id = 5;
```

| III I | резуль | таты 🖺  | Сообщения      |       |             |        |
|-------|--------|---------|----------------|-------|-------------|--------|
|       | id     | user_id | products_count | sum   | status      | amount |
| 1     | 1      | 1       | 2              | 1300  | new         | 2600   |
| 2     | 2      | 18      | 1              | 10000 | cancelled   | 10000  |
| 3     | 3      | 11      | 1              | 2140  | in_progress | 2140   |
| 4     | 4      | 145     | 5              | 6800  | new         | 34000  |
| 5     | 5      | 23      | 1              | 999   | success     | 999    |
| 6     | 6      | 1       | 2              | 7690  | cancelled   | 15380  |
| 7     | 7      | 17      | 1              | 1600  | new         | 1600   |
| 8     | 8      | 5       | 4              | 400   | delivery    | 1600   |
| 9     | 9      | 2355    | 1              | 1450  | new         | 1450   |
| 10    | 10     | 13      | 7              | 13000 | cancelled   | 91000  |

Увеличьте цену 5 самых дешевых товаров на 5%. products (исходная таблица)

| id | name              | count | price  |
|----|-------------------|-------|--------|
| 1  | Стиральная машина | 5     | 10000  |
| 2  | Микроволновка     | 3     | 4000   |
| 3  | Пылесос           | 2     | 4500   |
| 4  | Вентилятор        | 0     | 700    |
| 5  | Телевизор         | 7     | 31740  |
| 6  | iMac              | 3     | 109990 |
| 7  | iPhone 7          | 1     | 59990  |
| 8  | iPhone 8          | 3     | 64990  |
| 9  | iPhone X          | 2     | 79900  |
| 10 | Холодильник       | 0     | 10000  |

```
CREATE TABLE products (
  id INT NOT NULL PRIMARY KEY ,
  name NVARCHAR(255) NULL,
  count INTEGER NULL,
  price INTEGER NULL
);
INSERT INTO products (id, name, count, price)
VALUES
```

- (1, 'Стиральная машина', 5, 10000),
- (2, 'Микроволновка', 3, 4000),
- (3, 'Пылесос', 2, 4500),
- (4, 'Вентилятор', 0, 700),
- (5, 'Телевизор', 7, 31740),
- (6, 'iMac', 3, 109990),
- (7, 'iPhone 7', 1, 59990),
- (8, 'iPhone 8', 3, 64990),
- (9, 'iPhone X', 2, 79900),
- (10, 'Холодильник', 0, 10000);

**Решение.** Тут я решил довольно примитивным способом, зато именно то условие, которое указано в задании, а не просто пять первых товаров.

```
SELECT *FROM products

ORDER BY price ASC;

UPDATE products

SET price = price + (price/100*5)

WHERE id IN (4, 2, 3, 1, 10);
```

| Numbe | er of Records: 10 |       |        |
|-------|-------------------|-------|--------|
| id    | name              | count | price  |
| 1     | Стиральная машина | 5     | 10500  |
| 2     | Микроволновка     | 3     | 4200   |
| 3     | Пылесос           | 2     | 4725   |
| 4     | Вентилятор        | 0     | 735    |
| 5     | Телевизор         | 7     | 31740  |
| 6     | iMac              | 3     | 109990 |
| 7     | iPhone 7          | 1     | 59990  |
| 8     | iPhone 8          | 3     | 64990  |
| 9     | iPhone X          | 2     | 79900  |
| 10    | Холодильник       | 0     | 10500  |
|       |                   |       |        |

## Уменьшите цену 5 самых дорогих товаров на 5000 рублей.

### products (исходная таблица)

| id | name              | count | price  |
|----|-------------------|-------|--------|
| 1  | Стиральная машина | 5     | 10000  |
| 2  | Микроволновка     | 3     | 4000   |
| 3  | Пылесос           | 2     | 4500   |
| 4  | Вентилятор        | 0     | 700    |
| 5  | Телевизор         | 7     | 31740  |
| 6  | iMac              | 3     | 109990 |
| 7  | iPhone 7          | 1     | 59990  |
| 8  | iPhone 8          | 3     | 64990  |
| 9  | iPhone X          | 2     | 79900  |
| 10 | Холодильник       | 0     | 10000  |

```
CREATE TABLE products (
   id INT NOT NULL PRIMARY KEY ,
   name NVARCHAR(255) NULL,
   count INTEGER NULL,
   price INTEGER NULL
);
INSERT INTO products (id, name, count, price)
```

(1, 'Стиральная машина', 5, 10000),

- (2, 'Микроволновка', 3, 4000),
- (3, 'Пылесос', 2, 4500),

**VALUES** 

- (4, 'Вентилятор', 0, 700),
- (5, 'Телевизор', 7, 31740),
- (6, 'iMac', 3, 109990),
- (7, 'iPhone 7', 1, 59990),
- (8, 'iPhone 8', 3, 64990),
- (9, 'iPhone X', 2, 79900),
- (10, 'Холодильник', 0, 10000);

```
SELECT * FROM [products]
ORDER BY price DESC;

UPDATE products
SET price = price - 5000
WHERE id IN (6, 9, 8, 7, 5);
```

| d  | name              | count | price  |
|----|-------------------|-------|--------|
| 6  | iMac              | 3     | 104990 |
| 9  | iPhone X          | 2     | 74900  |
| 8  | iPhone 8          | 3     | 59990  |
| 7  | iPhone 7          | 1     | 54990  |
| 5  | Телевизор         | 7     | 26740  |
| 1  | Стиральная машина | 5     | 10000  |
| 10 | Холодильник       | 0     | 10000  |
| 3  | Пылесос           | 2     | 4500   |
| 2  | Микроволновка     | 3     | 4000   |
| 4  | Вентилятор        | 0     | 700    |

Ниже находится таблица с товарами в магазине. В поле count содержится текущее количество товаров на полках и на складе.

В магазин привезли 2 упаковки Сникерса и 2 упаковки Марса. В каждой упаковке по 20 шоколадок. Обновите данные так, чтобы они отражали количество шоколадок с учетом нового привоза.

# products (исходная таблица)

| id | name        | count |
|----|-------------|-------|
| 1  | Баунти      | 50    |
| 2  | Твикс       | 33    |
| 3  | Сникерс     | 4     |
| 4  | Пикник      | 25    |
| 5  | Марс        | 2     |
| 6  | Милка       | 18    |
| 7  | Альпен Голд | 13    |
| 8  | Дав         | 31    |
| 9  | Кит-кат     | 38    |
| 10 | Зебра       | 11    |

```
CREATE TABLE products (
id INT NOT NULL PRIMARY KEY,
name NVARCHAR(255) NULL,
count INTEGER NULL
);
INSERT INTO products (id, name, count)
VALUES
(1, 'Баунти', 50),
(2, 'Твикс', 33),
(3, 'Сникерс', 4),
(4, 'Пикник', 25),
(5, 'Марс', 2),
```

```
(6, 'Милка', 18),
(7, 'Альпен Голд', 13),
(8, 'Дав', 31),
(9, 'Кит-кат', 38),
(10, 'Зебра', 11);
```

```
UPDATE products
SET count =count + 40
WHERE name IN ('Сникерс', 'Марс')
```

Number of Records: 10

| id | name        | count |
|----|-------------|-------|
| 1  | Баунти      | 50    |
| 2  | Твикс       | 33    |
| 3  | Сникерс     | 44    |
| 4  | Пикник      | 25    |
| 5  | Марс        | 42    |
| 6  | Милка       | 18    |
| 7  | Альпен Голд | 13    |
| 8  | Дав         | 31    |
| 9  | Кит-кат     | 38    |
| 10 | Зебра       | 11    |

В конце каждого номера нужно вывести получившуюся таблицу на экран.

# Номер 1

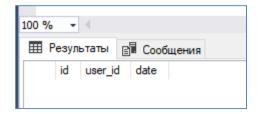
Удалите из таблицы visits все данные с помощью конструкции DELETE. visits (исходная таблица)

| id | user_id | date                |
|----|---------|---------------------|
| 1  | 1       | 2017-01-02 12:23:03 |
| 2  | 18      | 2017-01-02 14:23:24 |
| 3  | 11      | 2017-01-03 12:17:50 |
| 4  | 145     | 2017-01-06 18:37:01 |
| 5  | 1       | 2017-01-07 12:12:08 |
| 6  | 4       | 2017-01-07 12:12:14 |

```
CREATE TABLE visits (
    id INT NOT NULL PRIMARY KEY,
    user_id INTEGER NULL,
    date DATETIME NULL
);
INSERT INTO visits (id, user_id, date)
VALUES
    (1, 1, '2017-01-02 12:23:03'),
    (2, 18, '2017-01-02 14:23:24'),
    (3, 11, '2017-01-03 12:17:50'),
    (4, 145, '2017-01-06 18:37:01'),
    (5, 1, '2017-01-07 12:12:08'),
    (6, 4, '2017-01-07 12:12:14');
```

#### Решение.

#### DELETE FROM visits



### Удалите из таблицы products все товары, которых нет на складе

# products (исходная таблица)

| id | name        | count |
|----|-------------|-------|
| 1  | Баунти      | 50    |
| 2  | Твикс       | 33    |
| 3  | Сникерс     | 0     |
| 4  | Пикник      | 25    |
| 5  | Марс        | 0     |
| 6  | Милка       | 18    |
| 7  | Альпен Голд | 13    |
| 8  | Дав         | 0     |
| 9  | Кит-кат     | 38    |
| 10 | Зебра       | 11    |

```
CREATE TABLE products (
   id INT NOT NULL PRIMARY KEY,
   name NVARCHAR(255) NULL,
   count INTEGER NULL
);
INSERT INTO products (id, name, count)
VALUES
   (1, 'Баунти', 50),
   (2, 'Твикс', 33),
   (3, 'Сникерс', 0),
   (4, 'Пикник', 25),
   (5, 'Марс', 0),
   (6, 'Милка', 18),
   (7, 'Альпен Голд', 13),
   (8, 'Дав', 0),
   (9, 'Кит-кат', 38),
```

```
(10, 'Зебра', 11);
```

DELETE FROM products
WHERE count = 0

| Jumber of Records: 7 |             |       |  |
|----------------------|-------------|-------|--|
| id                   | name        | count |  |
| 1                    | Баунти      | 50    |  |
| 2                    | Твикс       | 33    |  |
| 4                    | Пикник      | 25    |  |
| 6                    | Милка       | 18    |  |
| 7                    | Альпен Голд | 13    |  |
| 9                    | Кит-кат     | 38    |  |
| 10                   | Зебра       | 11    |  |
|                      |             |       |  |

Удалите из таблицы cars все автомобили начиная с 2010 года и старше cars

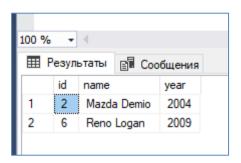
| id | name            | year |
|----|-----------------|------|
| 1  | Toyota Camry    | 2012 |
| 2  | Mazda Demio     | 2004 |
| 3  | BA3 2110        | 2010 |
| 4  | Nissan Almera   | 2016 |
| 5  | Nissan Juke     | 2016 |
| 6  | Reno Logan      | 2009 |
| 7  | Lada Priora     | 2017 |
| 8  | Reno Duster     | 2011 |
| 9  | Hyundai Solaris | 2010 |
| 10 | Nissan Patrol   | 2014 |

CREATE TABLE cars (
id INT NOT NULL PRIMARY KEY,
name NVARCHAR(255) NULL,
year INTEGER NULL

INSERT INTO cars (id, name, year)
VALUES

- (1, 'Toyota Camry', 2012),
- (2, 'Mazda Demio', 2004),
- (3, 'BA3 2110', 2010),
- (4, 'Nissan Almera', 2016),
- (5, 'Nissan Juke', 2016),
- (6, 'Reno Logan', 2009),
- (7, 'Lada Priora', 2017),
- (8, 'Reno Duster', 2011),
- (9, 'Hyundai Solaris', 2010),
- (10, 'Nissan Patrol', 2014);

DELETE FROM cars WHERE year >= 2010



Удалите из таблицы cars все корейские (country = "KR") автомобили, а также все автомобили мощностью (power) меньше 80 лс. Используйте один SQL-запрос.

cars

| id | name            | year | country | power |
|----|-----------------|------|---------|-------|
| 1  | Toyota Camry    | 2012 | JP      | 139   |
| 2  | Mazda Demio     | 2004 | JP      | 113   |
| 3  | BA3 2110        | 2010 | RU      | 79    |
| 4  | Nissan Almera   | 2016 | JP      | 130   |
| 5  | Nissan Juke     | 2016 | JP      | 120   |
| 6  | Reno Logan      | 2009 | FR      | 87    |
| 7  | Lada Priora     | 2017 | RU      | 78    |
| 8  | Reno Duster     | 2011 | FR      | 143   |
| 9  | Hyundai Solaris | 2010 | KR      | 98    |
| 10 | Nissan Patrol   | 2014 | JP      | 200   |

```
CREATE TABLE cars (
id INT NOT NULL PRIMARY KEY,
name NVARCHAR(255) NULL,
year INTEGER NULL,
country NVARCHAR(2) NULL,
power INTEGER NULL
);
INSERT INTO cars (id, name, year, country, power)
VALUES
(1, 'Toyota Camry', 2012, 'JP', 139),
(2, 'Mazda Demio', 2004, 'JP', 113),
(3, 'BA3 2110', 2010, 'RU', 79),
(4, 'Nissan Almera', 2016, 'JP', 130),
(5, 'Nissan Juke', 2016, 'JP', 120),
```

(6, 'Reno Logan', 2009, 'FR', 87), (7, 'Lada Priora', 2017, 'RU', 78), (8, 'Reno Duster', 2011, 'FR', 143), (9, 'Hyundai Solaris', 2010, 'KR', 98), (10, 'Nissan Patrol', 2014, 'JP', 200);

#### Решение.

DELETE FROM cars
WHERE country = 'KR'
OR power < 80

#### Number of Records: 7

| id | name          | year | country | power |
|----|---------------|------|---------|-------|
| 1  | Toyota Camry  | 2012 | JP      | 139   |
| 2  | Mazda Demio   | 2004 | JP      | 113   |
| 4  | Nissan Almera | 2016 | JP      | 130   |
| 5  | Nissan Juke   | 2016 | JP      | 120   |
| 6  | Reno Logan    | 2009 | FR      | 87    |
| 8  | Reno Duster   | 2011 | FR      | 143   |
| 10 | Nissan Patrol | 2014 | JP      | 200   |
|    |               |      |         |       |

Удалите из таблицы cars все японские автомобили мощностью менее 80 и более 130 лс. (включая крайние значения).

cars (исходная таблица)

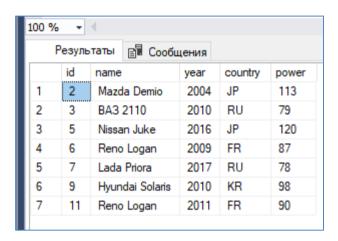
| id | name            | year | country | power |
|----|-----------------|------|---------|-------|
| 1  | Toyota Camry    | 2012 | JP      | 139   |
| 2  | Mazda Demio     | 2004 | JP      | 113   |
| 3  | BA3 2110        | 2010 | RU      | 79    |
| 4  | Nissan Almera   | 2016 | JP      | 130   |
| 5  | Nissan Juke     | 2016 | JP      | 120   |
| 6  | Reno Logan      | 2009 | FR      | 87    |
| 7  | Lada Priora     | 2017 | RU      | 78    |
| 8  | Reno Duster     | 2011 | FR      | 143   |
| 9  | Hyundai Solaris | 2010 | KR      | 98    |
| 10 | Nissan Patrol   | 2014 | JP      | 200   |
| 11 | Reno Logan      | 2011 | FR      | 90    |
| 12 | Nissan Sunny    | 1990 | JP      | 67    |

```
CREATE TABLE cars (
  id INT NOT NULL PRIMARY KEY,
  name NVARCHAR(255) NULL,
  year INTEGER NULL,
 country NVARCHAR(2) NULL,
  power INTEGER NULL
);
INSERT INTO cars (id, name, year, country, power)
```

- (1, 'Toyota Camry', 2012, 'JP', 139),
- (2, 'Mazda Demio', 2004, 'JP', 113),
- (3, 'BA3 2110', 2010, 'RU', 79),
- (4, 'Nissan Almera', 2016, 'JP', 130),

- (5, 'Nissan Juke', 2016, 'JP', 120),
- (6, 'Reno Logan', 2009, 'FR', 87),
- (7, 'Lada Priora', 2017, 'RU', 78),
- (8, 'Reno Duster', 2011, 'FR', 143),
- (9, 'Hyundai Solaris', 2010, 'KR', 98),
- (10, 'Nissan Patrol', 2014, 'JP', 200),
- (11, 'Reno Logan', 2011, 'FR', 90),
- (12, 'Nissan Sunny', 1990, 'JP', 67);

```
DELETE FROM cars
WHERE country = 'JP' AND
power <= 80 OR power >= 130
```



Удалите из таблицы cars все данные с помощью инструкции TRUNCATE (придется погуглить).

cars (исходная таблица)

| id | name            | year | country | power |
|----|-----------------|------|---------|-------|
| 1  | Toyota Camry    | 2012 | JP      | 139   |
| 2  | Mazda Demio     | 2004 | JP      | 113   |
| 3  | BA3 2110        | 2010 | RU      | 79    |
| 4  | Nissan Almera   | 2016 | JP      | 130   |
| 5  | Nissan Juke     | 2016 | JP      | 120   |
| 6  | Reno Logan      | 2009 | FR      | 87    |
| 7  | Lada Priora     | 2017 | RU      | 78    |
| 8  | Reno Duster     | 2011 | FR      | 143   |
| 9  | Hyundai Solaris | 2010 | KR      | 98    |
| 10 | Nissan Patrol   | 2014 | JP      | 200   |
| 11 | Reno Logan      | 2011 | FR      | 90    |
| 12 | Nissan Sunny    | 1990 | JP      | 67    |

```
CREATE TABLE cars (
id INT NOT NULL PRIMARY KEY,
name NVARCHAR(255) NULL,
year INTEGER NULL,
country NVARCHAR(2) NULL,
power INTEGER NULL
);
INSERT INTO cars (id, name, year, country, power)
VALUES
(1, 'Toyota Camry', 2012, 'JP', 139),
(2, 'Mazda Demio', 2004, 'JP', 113),
(3, 'BA3 2110', 2010, 'RU', 79),
```

(4, 'Nissan Almera', 2016, 'JP', 130),

- (5, 'Nissan Juke', 2016, 'JP', 120),
- (6, 'Reno Logan', 2009, 'FR', 87),
- (7, 'Lada Priora', 2017, 'RU', 78),
- (8, 'Reno Duster', 2011, 'FR', 143),
- (9, 'Hyundai Solaris', 2010, 'KR', 98),
- (10, 'Nissan Patrol', 2014, 'JP', 200),
- (11, 'Reno Logan', 2011, 'FR', 90),
- (12, 'Nissan Sunny', 1990, 'JP', 67);

#### TRUNCATE TABLE cars;

