

## # Домашнее задание

1. Используя операторы языка SQL, создайте таблицу "sales".  
Заполните ее данными.  
Справа располагается рисунок к первому заданию.

| id | order_date | count_product |
|----|------------|---------------|
| 1  | 2022-01-01 | 156           |
| 2  | 2022-01-02 | 180           |
| 3  | 2022-01-03 | 21            |
| 4  | 2022-01-04 | 124           |
| 5  | 2022-01-05 | 341           |

- Создаём базу данных,
- Выбираем её,
- Создаём таблицу:
- ...

```
CREATE DATABASE dz_sem2;
```

```
USE dz_sem2;
```

```
CREATE TABLE sales  
(  
    Id SERIAL PRIMARY KEY,  
    order_date DATE NOT NULL,  
    count_product INT  
);  
...
```

The screenshot displays a database management interface. On the left, the 'SCHEMAS' pane shows a tree view with 'dz\_sem2' expanded, revealing a table named 'sales'. The table's columns are listed as 'Id', 'order\_date', and 'count\_product'. The 'Information' pane at the bottom left shows 'No object selected'.

The main editor window, titled 'SQL File 6\*', contains the following SQL code:

```
15 • USE dz_sem2;  
16  
17 • CREATE TABLE sales  
18 • (  
19 •     Id SERIAL PRIMARY KEY,  
20 •     order_date DATE NOT NULL,  
21 •     count_product INT  
22 • );  
23  
24 • INSERT INTO sales(order_date, count_product)  
25 • VALUES  
26 • ('2022-01-01', '156'),  
27 • ('2022-01-02', '180'),  
28 • ('2022-01-03', '21'),  
29 • ('2022-01-04', '124'),  
30 • ('2022-01-05', '341');  
31  
32 • SELECT * FROM sales;
```

Below the SQL editor, the 'Result Grid' shows the data returned by the final query:

|   | Id   | order_date | count_product |
|---|------|------------|---------------|
| ▶ | 1    | 2022-01-01 | 156           |
|   | 2    | 2022-01-02 | 180           |
|   | 3    | 2022-01-03 | 21            |
|   | 4    | 2022-01-04 | 124           |
|   | 5    | 2022-01-05 | 341           |
| * | NULL | NULL       | NULL          |

2. Для данных таблицы "sales" укажите тип заказа в зависимости от кол-ва :

меньше 100 - Маленький заказ

от 100 до 300 - Средний заказ

больше 300 - Большой заказ

| id заказа | Тип заказа      |
|-----------|-----------------|
| 1         | Средний заказ   |
| 2         | Средний заказ   |
| 3         | Маленький заказ |
| 4         | Средний заказ   |
| 5         | Большой заказ   |

~~~

```
SELECT id,
CASE
    WHEN count_product < 100 THEN 'Маленький заказ'
    WHEN count_product >= 100 AND count_product <= 300 THEN 'Средний
заказ'
    ELSE 'Большой заказ'
END AS order_type
FROM sales;
~~~
```

The screenshot shows a SQL IDE window titled "sales" and "SQL File 6". The query editor contains the following SQL code:

```
30 ('2022-01-05', '341');
31
32 • SELECT * FROM sales;
33 /*
34 2. Для данных таблицы "sales" укажите тип заказа в зависимости от кол-ва :
35 меньше 100 - Маленький заказ
36 от 100 до 300 - Средний заказ
37 больше 300 - Большой заказ
38 */
39
40 • SELECT id,
41 CASE
42     WHEN count_product < 100 THEN 'Маленький заказ'
43     WHEN count_product >= 100 AND count_product <= 300 THEN 'Средний заказ'
44     ELSE 'Большой заказ'
45 END AS order_type
46 FROM sales;
47
```

The "Result Grid" at the bottom shows the following data:

|   | id | order_type      |
|---|----|-----------------|
| ▶ | 1  | Средний заказ   |
|   | 2  | Средний заказ   |
|   | 3  | Маленький заказ |
|   | 4  | Средний заказ   |
|   | 5  | Большой заказ   |

3.

Создайте таблицу "orders", заполните ее значениями

| id | employee_id | amount | order_status |
|----|-------------|--------|--------------|
| 1  | e03         | 15.00  | OPEN         |
| 2  | e01         | 25.50  | OPEN         |
| 3  | e05         | 100.70 | CLOSED       |
| 4  | e02         | 22.18  | OPEN         |
| 5  | e04         | 9.50   | CANCELLED    |

~~~

```
USE dz_sem2;
```

```
CREATE TABLE orders (
  id SERIAL PRIMARY KEY,
  employee_id VARCHAR(3) NOT NULL,
  amount DECIMAL(8,2) NOT NULL,
  order_status VARCHAR(10) NOT NULL
);
```

~~~

The screenshot shows the MySQL Workbench interface. On the left, the 'Navigator' pane displays the database schema, including the 'orders' table with columns: id, employee\_id, amount, and order\_status. The 'Information' pane at the bottom shows the table structure: id (bigint UN AI PK), employee\_id (varchar(3)), amount (decimal(8,2)), and order\_status (varchar(10)). The main editor shows a SQL script with the following content:

```

37  больше 300 - Большой заказ
38  */
39  /*
40  SELECT id,
41  CASE
42    WHEN count_product < 100 THEN 'Маленький заказ'
43    WHEN count_product >= 100 AND count_product <= 300 THEN 'Средний заказ'
44    ELSE 'Большой заказ'
45  END AS order_type
46  FROM sales;
47
48  -- Создайте таблицу "orders", заполните ее значениями
49  /*
50  Выберите все заказы.
51  В зависимости от поля order_status выведите столбец full_order_status:
52  OPEN - «Order is in open state»; CLOSED - «Order is closed»; CANCELLED - «Order is cancelled»
53  */
54
55  • USE dz_sem2;
56
57  • CREATE TABLE orders (
58    id SERIAL PRIMARY KEY,
59    employee_id VARCHAR(3) NOT NULL,
60    amount DECIMAL(8,2) NOT NULL,
61    order_status VARCHAR(10) NOT NULL
62  );
63

```

~~~

```
INSERT INTO orders (employee_id, amount, order_status)
VALUES
```

```

('e03', 15.00, 'OPEN'),
('e01', 25.50, 'OPEN'),
('e05', 100.70, 'CLOSED'),
('e02', 22.18, 'OPEN'),
('e04', 9.50, 'CANCELLED');
```

```
SELECT * FROM orders;
```

.....

```

54
55 • USE dz_sem2;
56
57 • CREATE TABLE orders (
58     id SERIAL PRIMARY KEY,
59     employee_id VARCHAR(3) NOT NULL,
60     amount DECIMAL(8,2) NOT NULL,
61     order_status VARCHAR(10) NOT NULL
62 );
63 • INSERT INTO orders (employee_id, amount, order_status)
64 VALUES
65     ('e03', 15.00, 'OPEN'),
66     ('e01', 25.50, 'OPEN'),
67     ('e05', 100.70, 'CLOSED'),
68     ('e02', 22.18, 'OPEN'),
69     ('e04', 9.50, 'CANCELLED');
70 • SELECT * FROM orders;

```

Result Grid

|   | id   | employee_id | amount | order_status |
|---|------|-------------|--------|--------------|
| ▶ | 1    | e03         | 15.00  | OPEN         |
|   | 2    | e01         | 25.50  | OPEN         |
|   | 3    | e05         | 100.70 | CLOSED       |
|   | 4    | e02         | 22.18  | OPEN         |
|   | 5    | e04         | 9.50   | CANCELLED    |
| * | NULL | NULL        | NULL   | NULL         |

4. Чем 0 отличается от NULL?

Напишите ответ в комментарии к домашнему заданию на платформе

~~~

0 - это конкретное число (целое или дробное), которое может быть использовано в качестве значения в столбцах числовых типов данных, таких как INTEGER, FLOAT и тп.

NULL обозначает отсутствие значения или неопределенное значение. Это может быть использовано в столбцах, которые могут содержать значения или не содержать их, таких как столбцы с датами и временем, столбцы, которые могут содержать текстовые данные, но не обязательно должны содержать их, и т.д.

Важно отметить, что операции сравнения с NULL, такие как равенство или неравенство, дают результат "неизвестно" (UNKNOWN). Таким образом, при использовании NULL в запросах SELECT, WHERE, JOIN и т.д., необходимо использовать операторы IS NULL или IS NOT NULL для корректного сравнения значений.

~~~