

1 task

Создание таблицы sales

Создать таблицу sales с тремя столбцами: id, order_date (дата заказа) и count_product (количество продуктов в заказе).

Затем заполнить эту таблицу данными, включая информацию о дате заказа и количестве продуктов в каждом заказе.

Заполните ее данными.

'2022-01-01', 156

'2022-01-02', 180

'2022-01-03', 21

'2022-01-04', 124

'2022-01-05', 341

Важно: Чтобы проверка прошла успешно, перед нажатием кнопки Проверить студент должен написать запрос и нажать кнопку Выполнить.

-- При написании запросов указывайте не только имя таблицы, но и схему.

-- Название вашей схемы - itresume8663785

-- Например, itresume8663785.tablename

-- Вы работаете с PostgreSQL

-- Введите свой код ниже

```
CREATE TABLE itresume8663785.sales (  
    id INT AUTO_INCREMENT PRIMARY KEY,  
    order_date DATE,  
    count_product INT  
);
```

```
INSERT INTO itresume8663785.sales (order_date, count_product)  
VALUES  
    ('2022-01-01', 156),  
    ('2022-01-02', 180),  
    ('2022-01-03', 21),  
    ('2022-01-04', 124),  
    ('2022-01-05', 341);
```

CREATE TABLE IF NOT EXISTS

itresume8663785.sales

(
id SERIAL PRIMARY KEY,

order_date DATE,

count_product INT

);

INSERT INTO itresume8663785.sales

(order_date,count_product)

VALUES

('2022-01-01', 156),

('2022-01-02', 180),

('2022-01-03', 21),

('2022-01-04', 124),

('2022-01-05', 341);

SELECT * FROM itresume8663785.sales;

The screenshot shows a web application interface for creating a PostgreSQL table and inserting data. The interface is divided into two main sections: a task description on the left and a code editor on the right.

Task Description (Left Panel):

- Задача (Task):** Создание таблицы sales (Creating the sales table).
- Решение (Solution):** Инструкция по использованию платформы (Platform usage instructions).
- Instructions:** Создать таблицу sales с тремя столбцами: id, order_date (дата заказа) и count_product (количество продуктов в заказе). Затем заполнить эту таблицу данными, включая информацию о дате заказа и количестве продуктов в каждом заказе. Названия столбцов: order_date, count_product. Заполните ее данными.
- Data Table:** A table with two columns: order_date and count_product. The data rows are: ('2022-01-01', 156), ('2022-01-02', 180), ('2022-01-03', 21), ('2022-01-04', 124), and ('2022-01-05', 341).
- Important Note:** Важно: чтобы проверка прошла успешно, перед нажатием кнопки Проверить студент должен написать запрос и нажать кнопку Выполнить.

Code Editor (Right Panel):

- Database:** PostgreSQL 13.10.
- Code:**

```
-- При написании запросов указывайте не только имя таблицы, но и
-- схему,
-- Название вашей схемы - itresume8663785
-- Например, itresume8663785.table_name

1 -- Вы работаете с PostgreSQL
2 -- Введите свой код ниже
3 CREATE TABLE IF NOT EXISTS
4 itresume8663785.sales
5 (
6 id SERIAL PRIMARY KEY,
7 order_date DATE,
8 count_product INT
9 );
10
11 INSERT INTO itresume8663785.sales
12 (order_date,count_product)
13 VALUES
14 ('2022-01-01', 156),
15 ('2022-01-02', 180),
16 ('2022-01-03', 21),
17 ('2022-01-04', 124),
18 ('2022-01-05', 341);
19
20
21 SELECT * FROM itresume8663785.sales;
22
```
- Buttons:** Показать терминал (Show terminal), Ассистент (Assistant), Выполнить (Execute), Проверить (Check).

2 task

Тип заказа в sales

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.

Выведите таблицу с названиями столбцов Номер заказа, Количество продукта, Тип

вариант1:

```
ALTER TABLE itresume8663785.sales  
ADD order_type VARCHAR(20);
```

```
UPDATE itresume8663785.sales  
SET order_type = CASE  
    WHEN count_product < 100 THEN 'Маленький заказ'  
    WHEN count_product >= 100 AND count_product <= 300 THEN 'Средний  
заказ'  
    WHEN count_product > 300 THEN 'Большой заказ'  
    ELSE 'Неизвестно'  
END;
```

```
SELECT id , count_product , order_type  
FROM itresume8663785.sales;
```

```
ALTER TABLE itresume8663785.sales  
ADD order_type VARCHAR(20);
```

```
UPDATE itresume8663785.sales  
SET order_type = CASE  
    WHEN count_product < 100 THEN 'Маленький заказ'  
    WHEN count_product >= 100 AND count_product <= 300 THEN 'Средний  
заказ'  
    WHEN count_product > 300 THEN 'Большой заказ'  
    ELSE 'Неизвестно'  
END;
```

```
SELECT id , count_product , order_type  
FROM itresume8663785.sales;
```

```
SELECT order_id AS id, employee_id, amount, order_status,  
    CASE  
        WHEN order_status = 'OPEN' THEN 'Order is in open state'  
        WHEN order_status = 'CLOSED' THEN 'Order is closed'  
        WHEN order_status = 'CANCELLED' THEN 'Order is cancelled'  
        ELSE 'Not mentioned'  
    END AS full_order_status  
FROM itresume8663785.orders;
```

```
SELECT id AS 'Номер заказа', count_product AS 'Количество продукта',  
order_type AS 'Тип заказа'  
FROM itresume8663785.sales;
```

решение:

```
SELECT id AS 'Номер заказа', count_product AS 'Количество продукта',  
CASE  
    WHEN count_product < 100 THEN 'Маленький заказ'  
    WHEN count_product >= 100 AND count_product <= 300 THEN  
'Средний заказ'  
    WHEN count_product > 300 THEN 'Большой заказ'  
    ELSE 'Неизвестно'  
END AS full_order_status  
FROM itresume8663785.sales;
```

The screenshot shows a web application interface for a SQL problem. On the left, the problem description is in Russian, asking for a query to categorize orders based on the number of products. The categories are: less than 100 (Small order), 100 to 300 (Medium order), and more than 300 (Large order). The output should be a table with columns: Order Number, Product Quantity, and Type. On the right, the solution is provided in a MySQL 8.1.0 editor. The SQL code is as follows:

```
1 -- Вы работаете с MySQL  
2 -- Введите свой код ниже  
3 SELECT id AS 'Номер заказа', count_product AS 'Количество  
4   продукта',  
5   CASE  
6     WHEN count_product < 100 THEN 'Маленький заказ'  
7     WHEN count_product >= 100 AND count_product <= 300 THEN  
8     'Средний заказ'  
9     WHEN count_product > 300 THEN 'Большой заказ'  
10    ELSE 'Неизвестно'  
11  END AS full_order_status  
12 FROM sales;
```

At the bottom of the interface, there are buttons for 'Показать терминал' (Show terminal), 'Ассистент' (Assistant), 'Выполнить' (Execute), and 'Проверить' (Check). The status bar at the very bottom indicates 'Сброс' (Reset), '100%', and '23:58'.

3 task

Создание таблицы orders

Используя операторы языка SQL, создайте таблицу orders, заполните ее значениями.

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

Выберите все заказы. В зависимости от поля order_status выведите столбец full_order_status:

OPEN – «Order is in open state» ;
CLOSED - «Order is closed»;
CANCELLED - «Order is cancelled».

Важно: Чтобы проверка прошла успешно, перед нажатием кнопки Проверить студент должен написать запрос и нажать кнопку Выполнить.

-- При написании запросов указывайте не только имя таблицы, но и схему.
-- Название вашей схемы - itresume8663785
-- Например, itresume8663785.tablename

-- Вы работаете с PostgreSQL
-- Введите свой код ниже

```
CREATE TABLE IF NOT EXISTS itresume8663785.orders
```

```
(  
  order_id VARCHAR(10) PRIMARY KEY,  
  total_price DECIMAL(10, 2),  
  order_status VARCHAR(10)  
);
```

```
INSERT INTO itresume8663785.orders  
(order_id, total_price, order_status)  
VALUES
```

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

```
SELECT *,  
CASE
```

```
  WHEN order_status = 'OPEN' THEN 'Order is in open state'
```

```
  WHEN order_status = 'CLOSED' THEN 'Order is closed'
```

```
  WHEN order_status = 'CANCELLED' THEN 'Order is cancelled'
```

```
  ELSE 'Unknown'
```

```
END AS full_order_status
```

```
FROM itresume8663785.orders;
```

The screenshot shows a web application interface for creating a PostgreSQL table and inserting data. The interface is divided into two main sections: a task description on the left and a code editor on the right.

Task Description (Left Panel):

- Задача:** Создание таблицы orders
- Решение:** Инструкция по использованию платформы
- Используя операторы языка SQL, создайте таблицу orders, заполните ее значениями.**
- Названия столбцов:** employee_id, amount, order_status.
- Вывод данных:** Выберите все заказы. В зависимости от поля order_status выведите столбец full_order_status:
 - OPEN - «Order is in open state»;
 - CLOSED - «Order is closed»;
 - CANCELLED - «Order is cancelled».
- Важно:** Чтобы проверка прошла успешно, перед нажатием кнопки Проверить студент должен написать запрос и нажать кнопку Выполнить.

Code Editor (Right Panel):

```
1 -- Вы работаете с PostgreSQL  
2 -- Введите свой код ниже  
3 CREATE TABLE IF NOT EXISTS itresume8663785.orders  
4 (  
5   order_id VARCHAR(10) PRIMARY KEY,  
6   total_price DECIMAL(10, 2),  
7   order_status VARCHAR(10)  
8 );  
9  
10 INSERT INTO itresume8663785.orders  
11 (order_id, total_price, order_status)  
12 VALUES  
13 ('e03', 15.00, 'OPEN'),  
14 ('e01', 25.50, 'OPEN'),  
15 ('e05', 100.70, 'CLOSED'),  
16 ('e02', 22.18, 'OPEN'),  
17 ('e04', 9.50, 'CANCELLED');  
18  
19  
20 SELECT *,  
21 CASE  
22   WHEN order_status = 'OPEN' THEN 'Order is in open state'  
23   WHEN order_status = 'CLOSED' THEN 'Order is closed'  
24   WHEN order_status = 'CANCELLED' THEN 'Order is cancelled'  
25   ELSE 'Unknown'  
26 END AS full_order_status  
27 FROM itresume8663785.orders;  
28
```

Buttons at the bottom: Показать терминал, Ассистент, Выполнить, Проверить.

4 task

Статус заказа

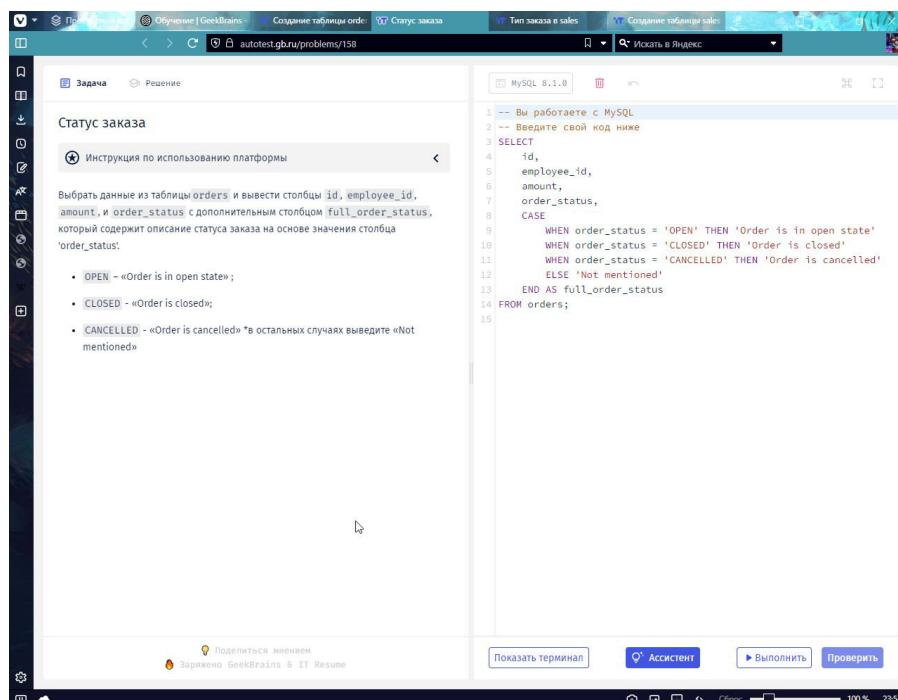
Выбрать данные из таблицы orders и вывести столбцы id, employee_id, amount, и order_status с дополнительным столбцом full_order_status, который содержит описание статуса заказа на основе значения столбца 'order_status'.

OPEN – «Order is in open state» ;

CLOSED - «Order is closed»;

CANCELLED - «Order is cancelled» *в остальных случаях выведите «Not mentioned»

```
SELECT id,employee_id,amount,order_status,  
CASE  
    WHEN order_status = 'OPEN' THEN 'Order is in open state'  
    WHEN order_status = 'CLOSED' THEN 'Order is closed'  
    WHEN order_status = 'CANCELLED' THEN 'Order is cancelled'  
    ELSE 'Not mentioned'  
END AS full_order_status  
FROM itresume8663785.orders;
```




```
SELECT order_id AS id, employee_id, amount, order_status,  
CASE  
    WHEN order_status = 'OPEN' THEN 'Order is in open state'  
    WHEN order_status = 'CLOSED' THEN 'Order is closed'  
    WHEN order_status = 'CANCELLED' THEN 'Order is cancelled'  
    ELSE 'Not mentioned'  
END AS full_order_status  
FROM itresume8663785.orders;
```