

1. Посчитай, сколько пользователей зарегистрировано в системе. Это таблица user\_model. В результате выведи только количество пользователей.

```
SELECT COUNT(*)  
FROM user_model;
```

2. Добавь три новых разных продукта в таблицу product\_model.

```
INSERT INTO  
product_model  
(name, price, weight, units, "categoryId")  
VALUES  
("Чипсы "Chitos", 300, 50, 'г', 3), ("Сок "Dobry", 120, 500, 'мл', 1), ("Хлеб "Bovarcky", 120, 500, 'г', 2);
```

3. Посчитай количество продуктов в каждой категории и вывести id только тех категорий, в которых количество продуктов больше пяти. Это таблица product\_model. Результат отсортируй в порядке возрастания количества продуктов.

```
SELECT "categoryId", COUNT("categoryId")  
FROM product_model  
GROUP BY "categoryId"  
HAVING COUNT("categoryId") > 5  
ORDER BY COUNT("categoryId");
```

4. В приложение хотят добавить фичу — возможность вносить правки в заказы. Сработает только с теми заказами, где:

- стоимость доставки (deliveryPrice) больше 500,
- стоит статус «заказ формируется» или «заказ в доставке».

Напиши запрос, который будет выводить в системе id всех заказов и возможность внести правки. Назови эту колонку update\_order. Если статус заказа позволяет вносить изменения, то в колонку update\_order нужно вывести yes. Если правки внести нельзя — вывести no.

Подсказка: в запросе используй таблицу order\_model и оператор CASE-WHEN-THEN.

```
SELECT id,  
CASE  
WHEN "deliveryPrice" > 500 AND  
  (status = 0 OR status = 1) THEN 'yes'  
ELSE 'no'  
END update_order  
FROM order_model;
```

```
deviona -- ssh c5e70f63-e0fc-4c53-8e54-53165fb800a3@serverhub.praktikum-services.ru -p -- c5e70f63-e0fc-4c53-8e54-53165fb800a3@serverhub.prakt...

postgres=> select * from order_model;
postgres=> select * from order_model;
 id | userId | deliveryPrice | deliveryPriceOur | deliveryTime | wareHouse | courierService | status | productsList
-----+-----+-----+-----+-----+-----+-----+-----+-----
  1 |      |      250      |                  |              |           |                |    0   |
  2 |      |      250      |                  |              |           |                |    1   |
  3 |      |      250      |                  |              |           |                |    2   |
  4 |      |      333      |                  |              |           |                |    0   |
  5 |      |      333      |                  |              |           |                |    1   |
  6 |      |      333      |                  |              |           |                |    2   |
  7 |      |      450      |                  |              |           |                |    0   |
  8 |      |      450      |                  |              |           |                |    1   |
  9 |      |      450      |                  |              |           |                |    2   |
 10 |      |      500      |                  |              |           |                |    0   |
 11 |      |      500      |                  |              |           |                |    1   |
 12 |      |      500      |                  |              |           |                |    2   |
 13 |      |      555      |                  |              |           |                |    0   |
 14 |      |      555      |                  |              |           |                |    1   |
 15 |      |      555      |                  |              |           |                |    2   |
(15 rows)

postgres=> SELECT id,
postgres=> CASE
postgres-> WHEN "deliveryPrice" > 500 AND
postgres-> (status = 0 OR status = 1) THEN 'yes'
postgres-> ELSE 'no'
postgres-> END update_order
postgres-> FROM order_model;
 id | update_order
-----+-----
  1 | no
  2 | no
  3 | no
  4 | no
  5 | no
  6 | no
  7 | no
  8 | no
  9 | no
 10 | no
 11 | no
 12 | no
 13 | yes
 14 | yes
 15 | no
(15 rows)

postgres=>
```

5. Выведи информацию о продуктах, цена которых находится в диапазоне от 200 до 500. Информация по каждому продукту включает: название продукта, цену, название категории, к которой он относится. Подсказка: в запросе используй объединение двух таблиц: product\_model и category\_model.

```
SELECT pm.name, pm.price, cm.name AS category_name
FROM product_model AS pm
LEFT JOIN category_model AS cm ON pm."categoryId" = cm.id
WHERE pm.price BETWEEN 200 AND 500;
```

6. Для каждой карточки выведи ее название и количество продуктов (productsCount) для этой карточки. Результат отсортируй по названию карточки. Подсказка: в запросе используй оператор группировки и объединение двух таблиц, card\_model и kit\_model.

```
SELECT c.name, SUM(k."productsCount") AS productscount
FROM kit_model AS k
LEFT JOIN card_model AS c ON k."cardId" = c.id
GROUP BY c.name
ORDER BY c.name;
```