**Дополнительные sql-запросы.**

**1.** Составить sql запрос, который выведет все категории, даже с

количеством товаров 0, но с указанием количества товара для каждой

категории, как и в оригинальном задании.

SELECT c.category\_id,

c.name,

count(pc.category\_id) AS count\_category,

c.description

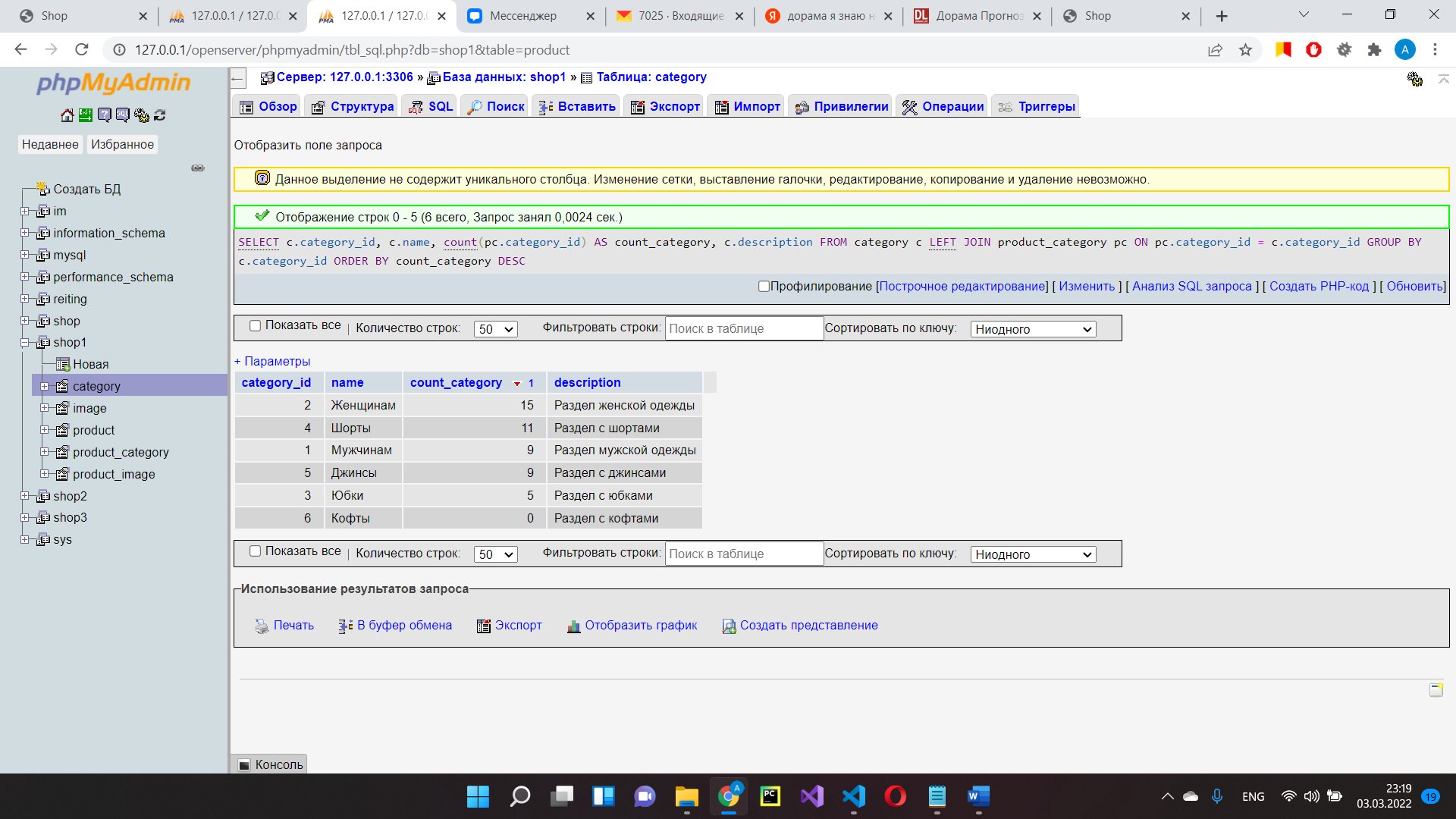
FROM category c

LEFT JOIN product\_category pc

ON pc.category\_id = c.category\_id

GROUP BY c.category\_id

ORDER BY count\_category DESC



**2.** Составить sql запрос, который выведет все категории товаров с

указанием количество товаров, имеющих количество товаров больше

либо равно 2.

SELECT \* FROM (SELECT c.category\_id,

c.name,

count(\*) AS count\_category,

c.description

FROM category c

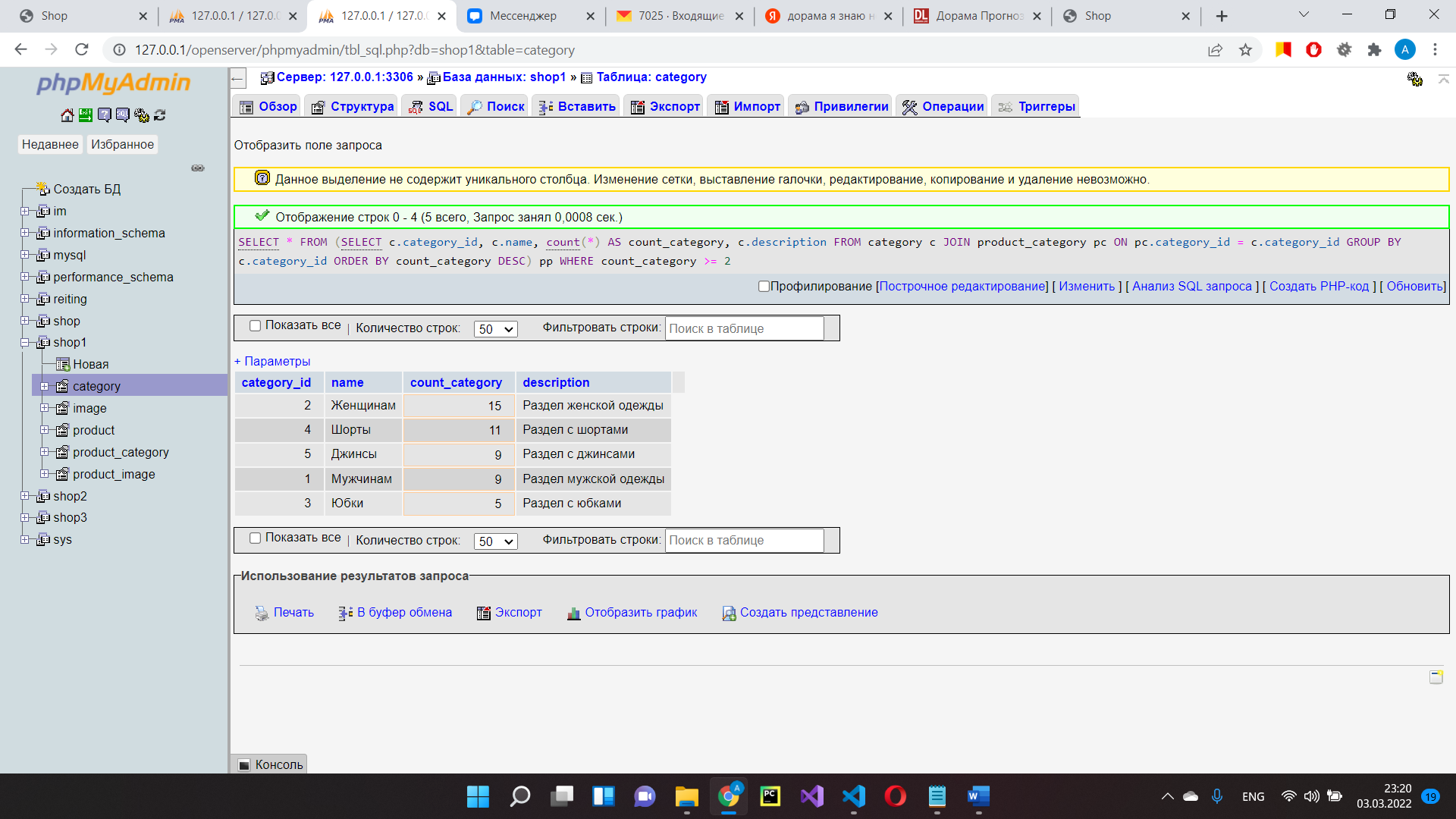
JOIN product\_category pc

ON pc.category\_id = c.category\_id

GROUP BY c.category\_id

ORDER BY count\_category DESC) pp

WHERE count\_category >= 2



**Безопасность сайта.**

Уязвимым местом для XSS и SQL-инъекций является адресная строка, в которой передаются GET-параметры, такие как category\_id и product\_id.

Так как валидными GET-параметрами являются только положительные целые числа, то при обработке данных, которые может ввести пользователь в адресной строке, была использована функция int:

$category\_id = (int)$\_GET['category\_id'];

Таким образом при вводе строки с нежелательными тегами или же SQL-инъекций, в результате будет получен 0.