**1 - Операторы соединения и пересечения таблиц:**

**Внутреннее соединение:**

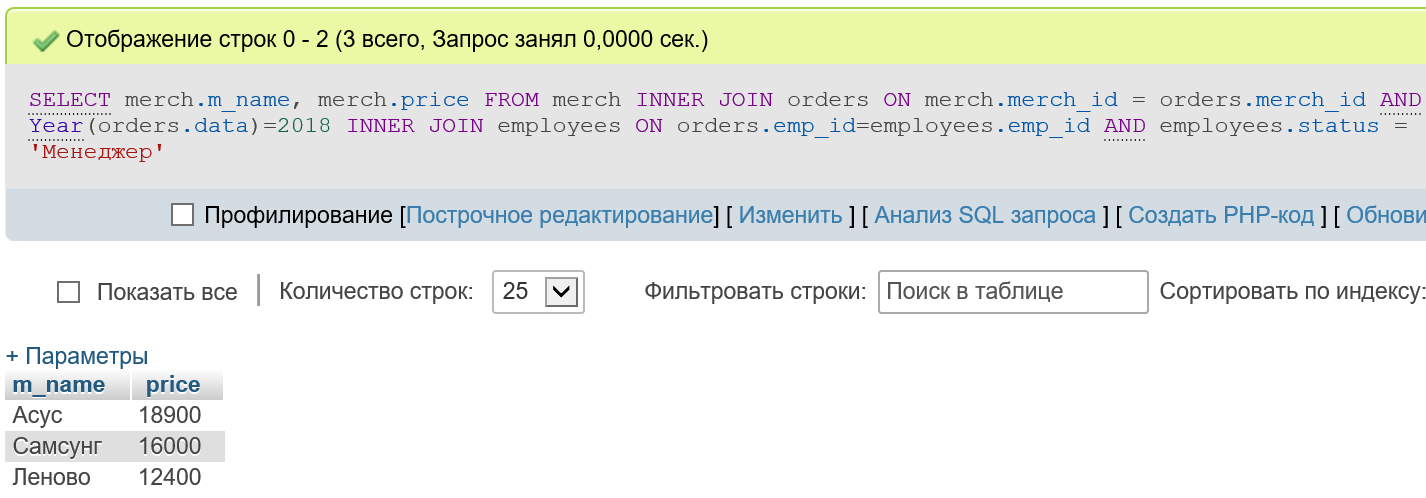
1.1) Вывести товары и их цену, которые заказаны в 2018 году любым менеджером.

SELECT merch.m\_name, merch.price

FROM merch

INNER JOIN orders ON merch.merch\_id = orders.merch\_id AND Year(orders.data)=2018

INNER JOIN employees ON orders.emp\_id=employees.emp\_id AND employees.status = 'Менеджер'



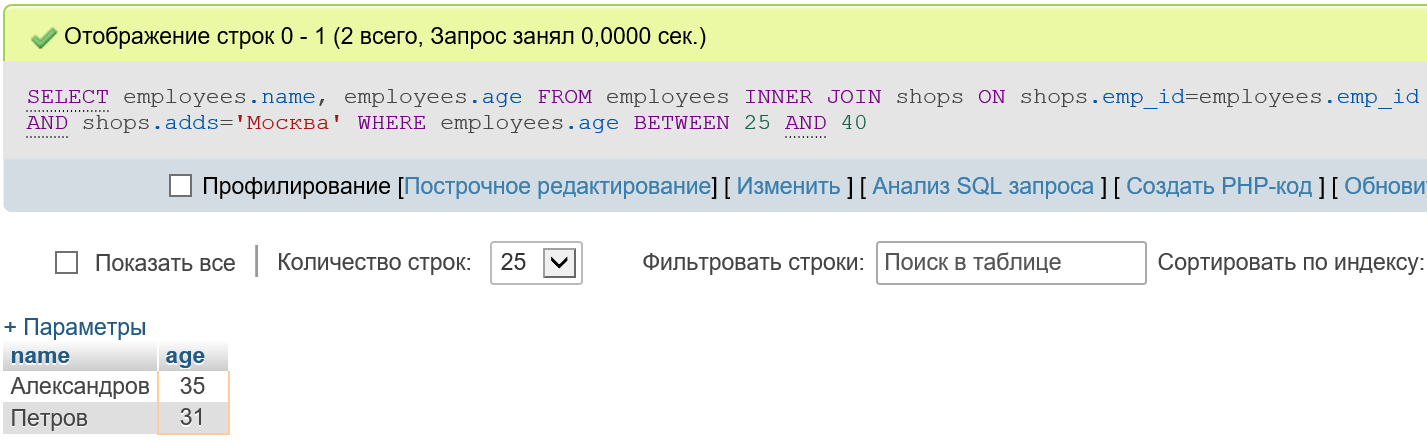
1.2) Вывести список сотрудников, возраст которых находится в диапазоне (25; 40), которые работаю в Московском магазине

SELECT employees.name, employees.age

FROM employees

INNER JOIN shops ON shops.emp\_id=employees.emp\_id AND shops.adds='Москва'

WHERE employees.age BETWEEN 25 AND 40



1.3) Вывести общую сумму заказов (в рублях) для каждого города.

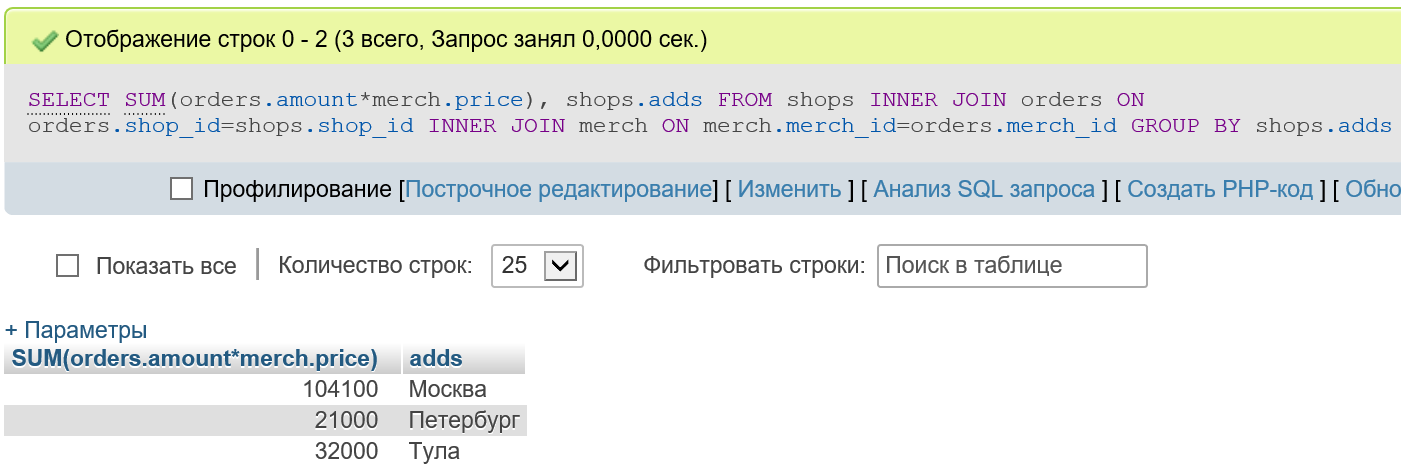
SELECT SUM(orders.amount\*merch.price), shops.adds

FROM shops

INNER JOIN orders ON orders.shop\_id=shops.shop\_id

INNER JOIN merch ON merch.merch\_id=orders.merch\_id

GROUP BY shops.adds



1.4) Вывести название самого прибыльного магазина (то есть магазина, у которого наибольшая сумма за все продажи).

SELECT SUM(orders.amount\*merch.price) AS S, shops.name

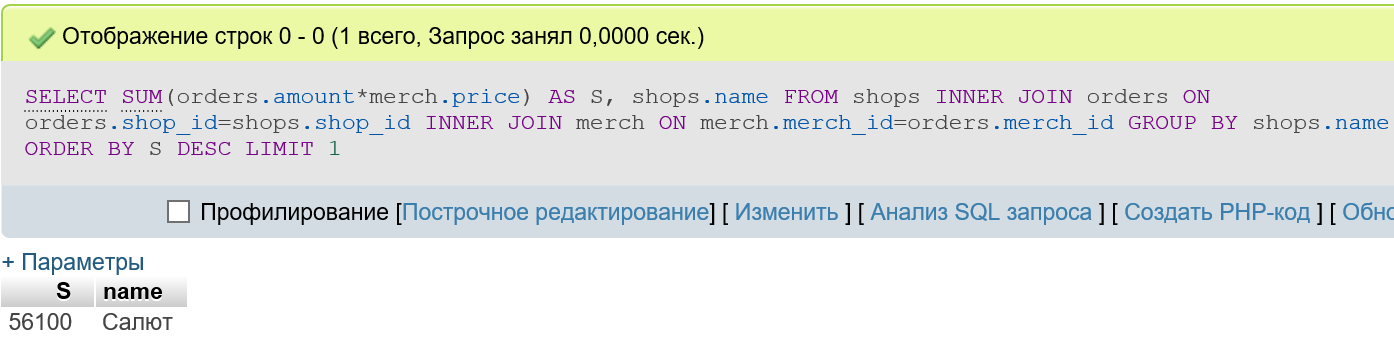
FROM shops

INNER JOIN orders ON orders.shop\_id=shops.shop\_id

INNER JOIN merch ON merch.merch\_id=orders.merch\_id

GROUP BY shops.name

ORDER BY S DESC LIMIT 1



1.5) Вывести имя самого прибыльного сотрудника.

SELECT SUM(orders.amount\*merch.price) AS S, employees.name

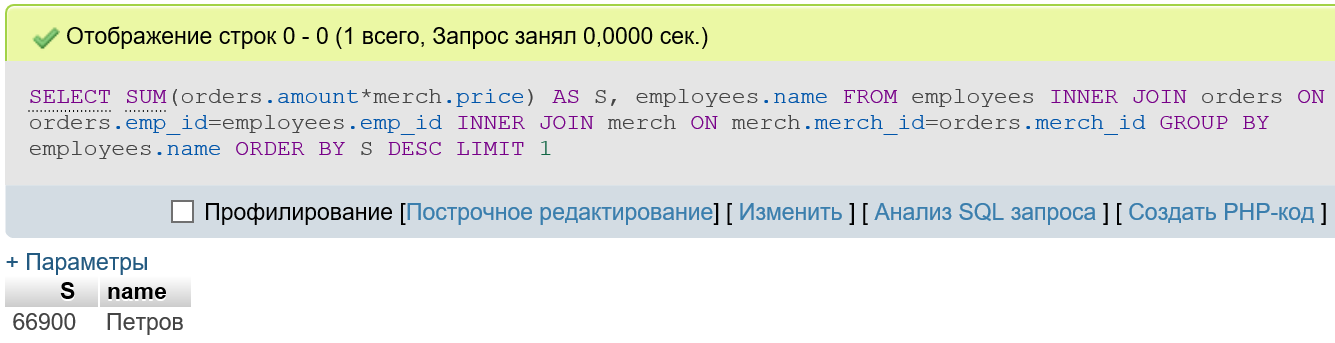
FROM employees

INNER JOIN orders ON orders.emp\_id=employees.emp\_id

INNER JOIN merch ON merch.merch\_id=orders.merch\_id

GROUP BY employees.name

ORDER BY S DESC LIMIT 1



1.6) Вывести директоров, которые продали товар марки  "Самсунг".

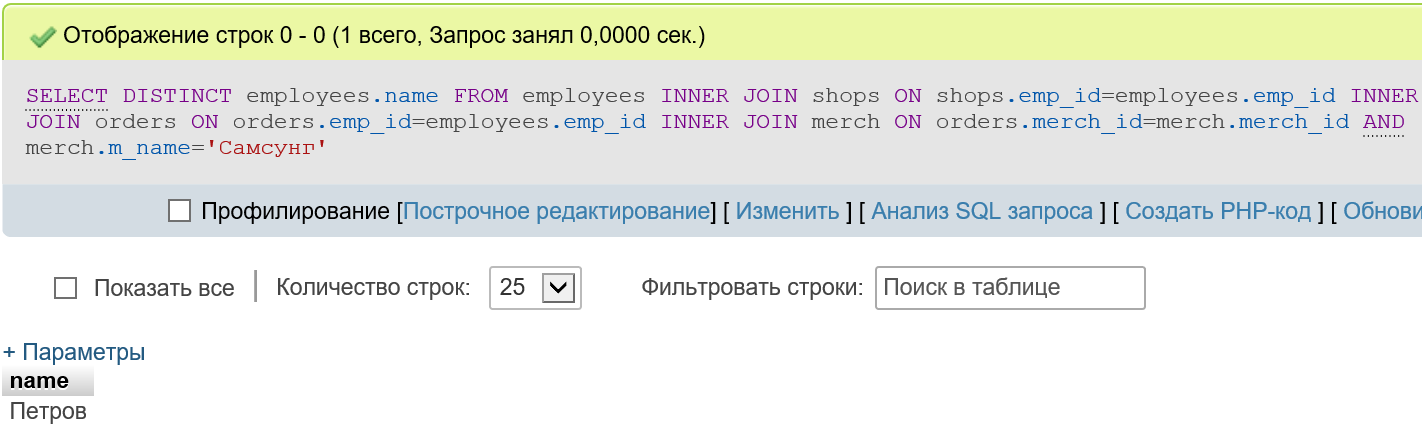
SELECT DISTINCT employees.name

FROM employees

INNER JOIN shops ON shops.emp\_id=employees.emp\_id

INNER JOIN orders ON orders.emp\_id=employees.emp\_id

INNER JOIN merch ON orders.merch\_id=merch.merch\_id AND merch.m\_name='Самсунг'



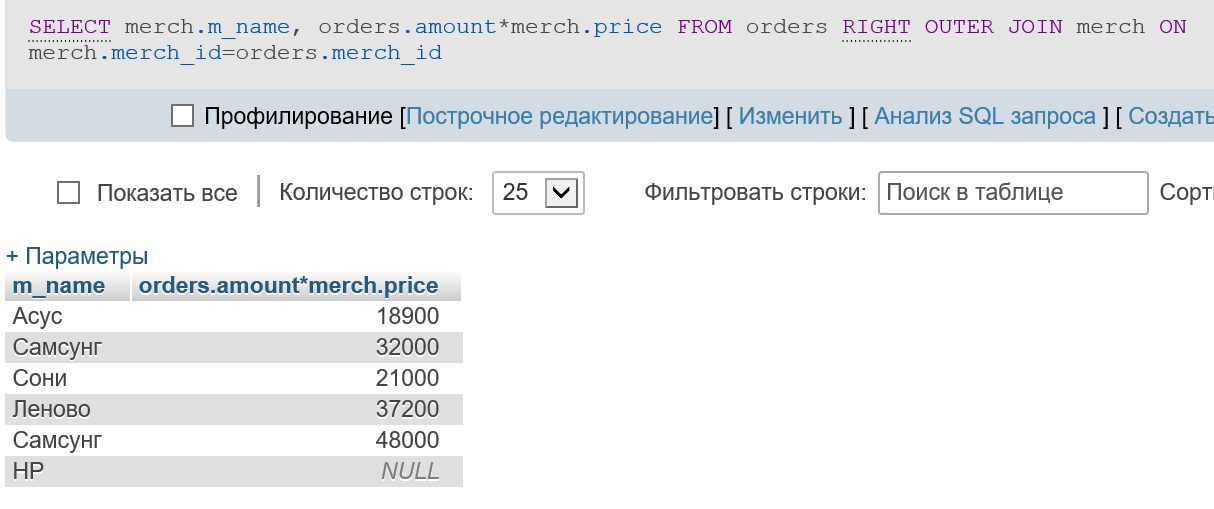
**Внешнее соединение :**

1.7) Вывести для каждого товара в БД его заказы: наименование товара, сумму заказа (в рублях), даже если для какого-то товара нет заказов.

SELECT merch.m\_name, orders.amount\*merch.price

FROM orders

RIGHT OUTER JOIN merch ON merch.merch\_id=orders.merch\_id

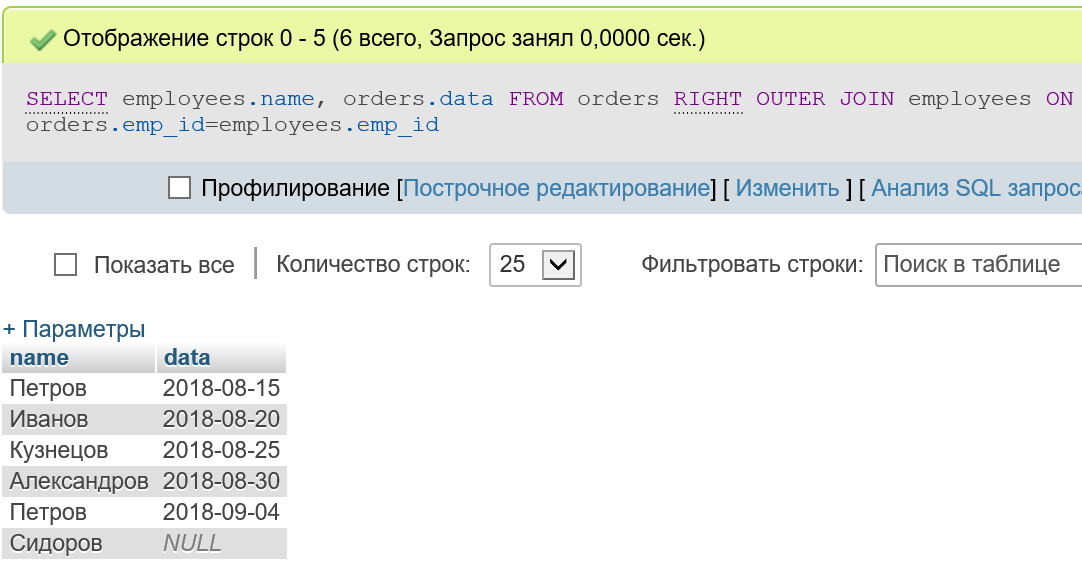


1.8) Вывести список всех сотрудников и дату заказов, оформленных этими сотрудниками

SELECT employees.name, orders.data

FROM orders

RIGHT OUTER JOIN employees ON orders.emp\_id=employees.emp\_id



**2 - Объединение запросов**

2.1) Создайте объединение из двух запросов, которое показало бы товар, количество их в заказах и описание. Те из них, которые имеют  количество больше среднего, должны иметь описание "Выше среднего", а те которые имеют количество меньше среднего описание "Ниже среднего."

SELECT merch.m\_name, orders.amount, 'high'

FROM merch, orders

WHERE orders.merch\_id=merch.merch\_id AND orders.amount>(SELECT AVG(orders.amount)

FROM orders)

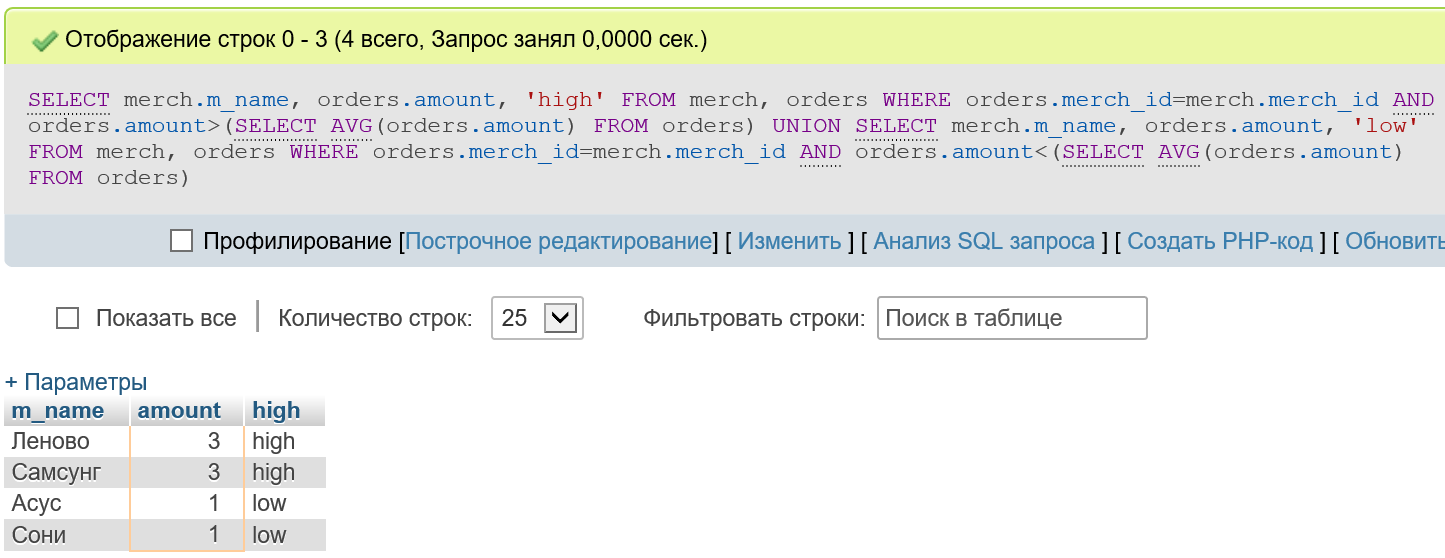
UNION

SELECT merch.m\_name, orders.amount, 'low'

FROM merch, orders

WHERE orders.merch\_id=merch.merch\_id AND orders.amount<(SELECT AVG(orders.amount)

FROM orders)



2.2) Создайте объединение из двух запросов, которое показало бы имена сотрудников и количество их заказов. Те из них, которые имеют   максимальное количество, должны, кроме того, иметь слова "Наивысший", а те которые имеют  минимальное количество "Низший". Результат отсортируйте по алфавиту

SELECT tab.N, tab.S, 'Highest'

FROM(SELECT employees.name as N, COUNT(orders.ord\_id) as S

FROM orders

INNER JOIN employees ON orders.emp\_id=employees.emp\_id

GROUP BY employees.name)tab

WHERE tab.S=(SELECT MAX(q.S)

FROM(SELECT employees.name as N, COUNT(orders.ord\_id) as S

FROM orders

INNER JOIN employees ON orders.emp\_id=employees.emp\_id

GROUP BY employees.name)q)

UNION

SELECT tab.N, tab.S, 'Lowest'

FROM(SELECT employees.name as N, COUNT(orders.ord\_id) as S

FROM orders

INNER JOIN employees ON orders.emp\_id=employees.emp\_id

GROUP BY employees.name)tab

WHERE tab.S=(SELECT MIN(q.S)

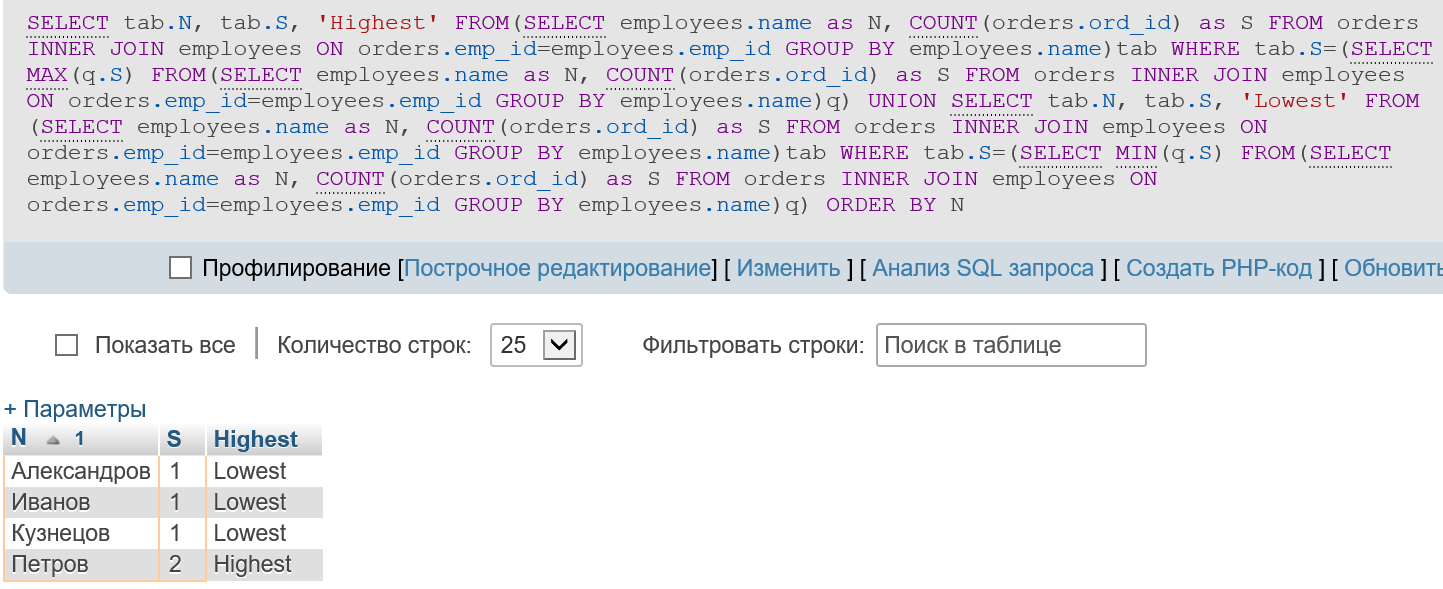
FROM(SELECT employees.name as N, COUNT(orders.ord\_id) as S

FROM orders

INNER JOIN employees ON orders.emp\_id=employees.emp\_id

GROUP BY employees.name)q)

ORDER BY N



**3 - Представления**

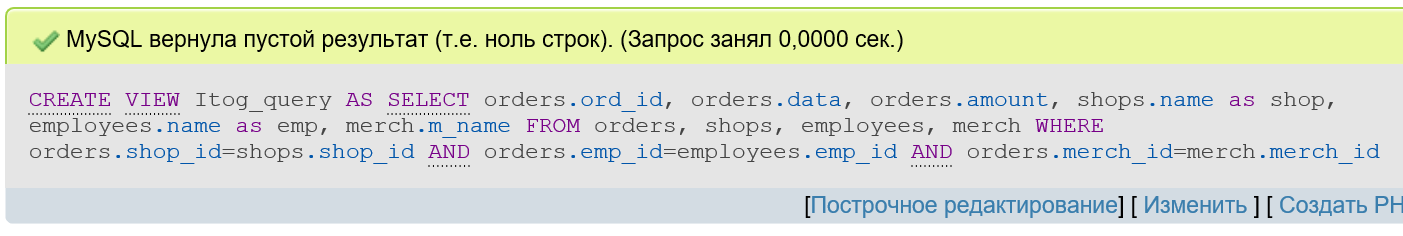
3.1) Создайте представление "Itog\_query" в котором все внешние id из таблицы заказов поменяются на наименование товара/магазина/сотрудника

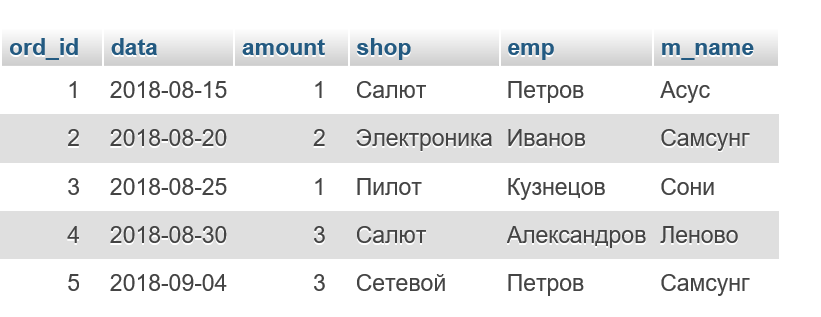
CREATE VIEW Itog\_query

AS SELECT orders.ord\_id, orders.data, orders.amount, shops.name as shop, employees.name as emp, merch.m\_name

FROM orders, shops, employees, merch

WHERE orders.shop\_id=shops.shop\_id AND orders.emp\_id=employees.emp\_id AND orders.merch\_id=merch.merch\_id



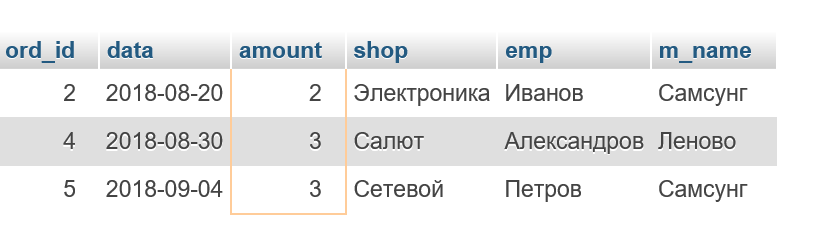


3.1.1) Вывести сотрудников, которые оформили заказ, имеющий более 1 единицы товара в своем составе из созданного представления.

SELECT \*

FROM itog\_query

WHERE itog\_query.amount>1



3.2) Создайте представление "Avg\_Sum", которое бы показывало усредненные суммы заказов для каждого продавца после его имени.

CREATE VIEW Avg\_Sum

AS SELECT AVG(orders.amount\*merch.price), employees.name

FROM employees

INNER JOIN orders ON orders.emp\_id=employees.emp\_id

INNER JOIN merch ON merch.merch\_id=orders.merch\_id

GROUP BY employees.name

