Министерство науки и высшего образования Российской Федерации  
Пензенский государственный университет  
Кафедра вычислительная техника

**ОТЧЕТ**

по лабораторной работе №3

по дисциплине «Программно-аппаратные средства хранения и обработки данных»

на тему «Создание простых и сложных запросов на выборку. Операторы и функции MySQL.»

Выполнили:

студенты группы 22ВВП1

Демин М. С.

Беляев Д. И.

Амиров И. Р.

Приняли:

Дубинин В.Н.

Карамышева Н.С.

Пенза 2025

**Название**

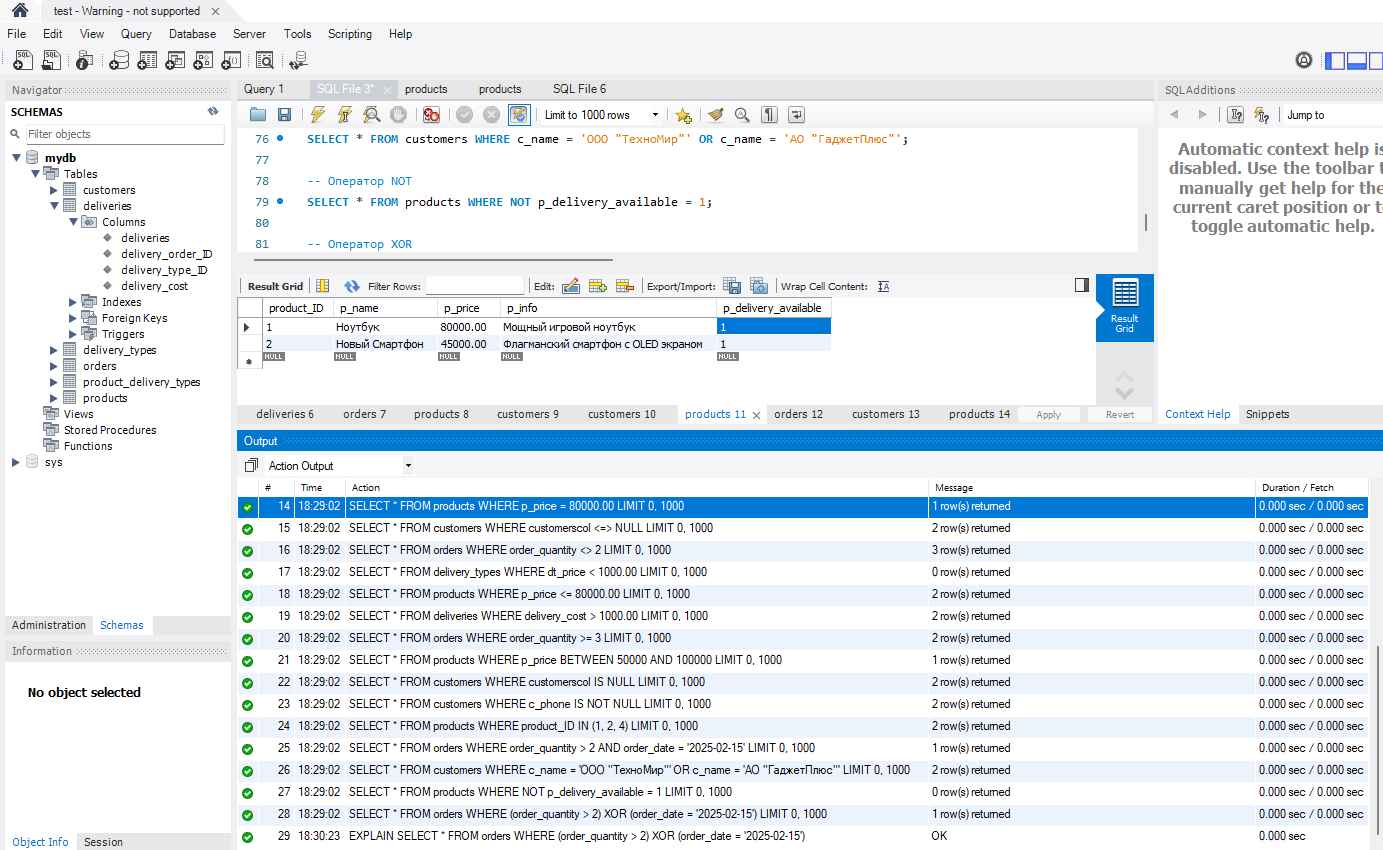
Создание простых и сложных запросов на выборку. Операторы и функции MySQL.

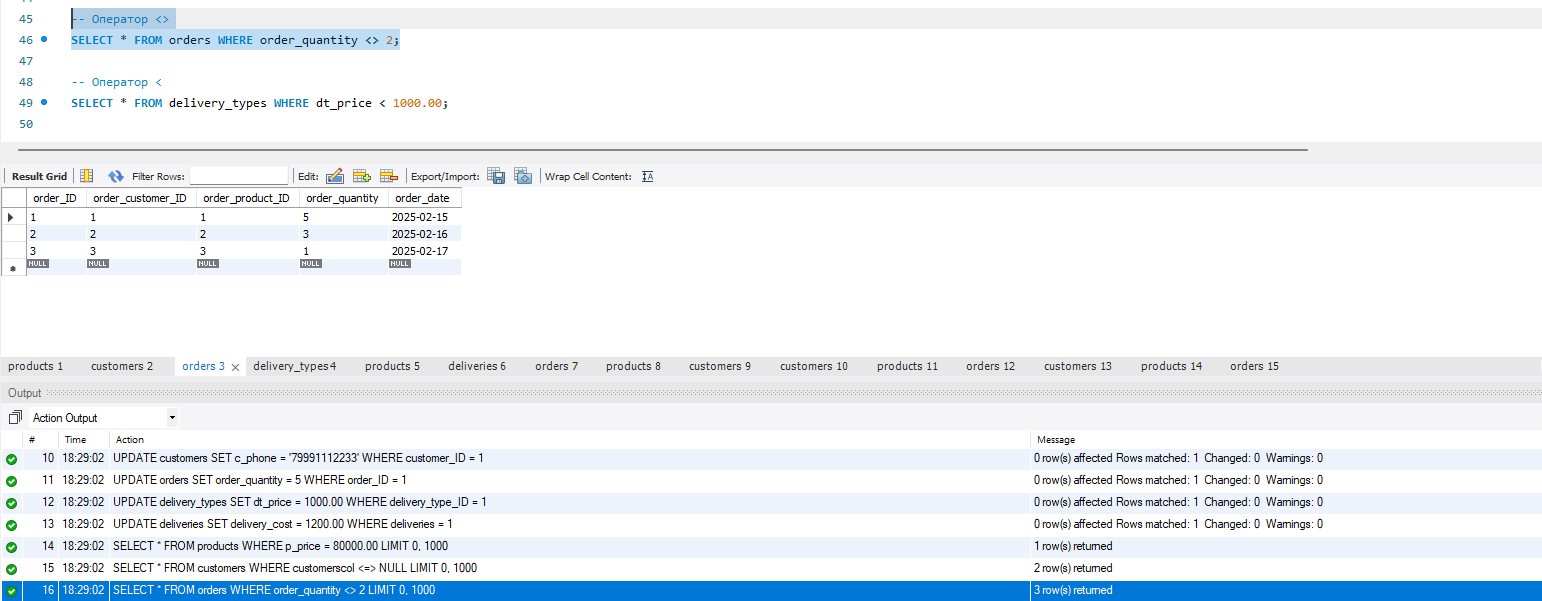
**Цель работы**

Построить не менее десяти простых запросов на выборку с использованием операторов и функций MySQL,изучить переменные SQL и временные таблицы, построить многотабличные запросы на выборку с использованием объединения, построить запросы на выборку, содержащие вложенные запросы

**Ход работы:**

1. Для заданной предметной области построить не менее десяти простых запросов на выборку с использованием операторов и функций MySQL;





Листинг:

-- Использование различных операторов

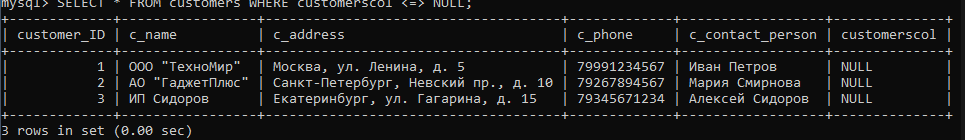
-- Оператор =

SELECT \* FROM products WHERE p\_price = 80000.00;



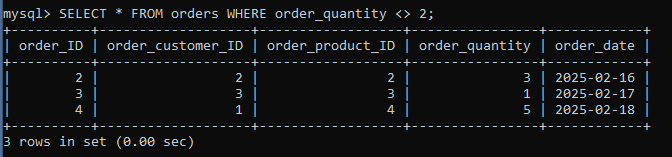
-- Оператор <=>

SELECT \* FROM customers WHERE customerscol <=> NULL;



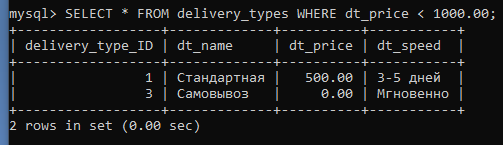
-- Оператор <>

SELECT \* FROM orders WHERE order\_quantity <> 2;



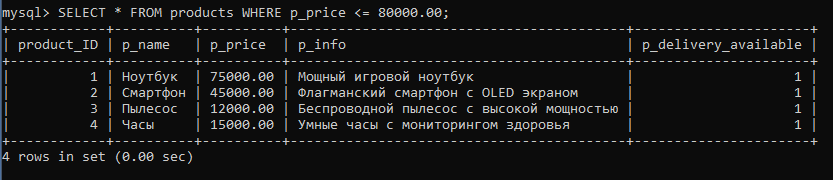
-- Оператор <

SELECT \* FROM delivery\_types WHERE dt\_price < 1000.00;



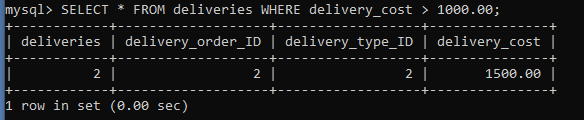
-- Оператор <=

SELECT \* FROM products WHERE p\_price <= 80000.00;



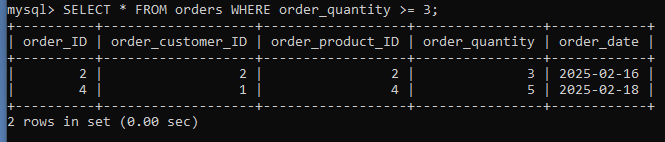
-- Оператор >

SELECT \* FROM deliveries WHERE delivery\_cost > 1000.00;



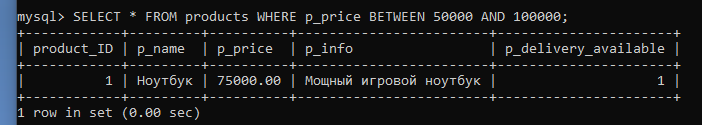
-- Оператор >=

SELECT \* FROM orders WHERE order\_quantity >= 3;



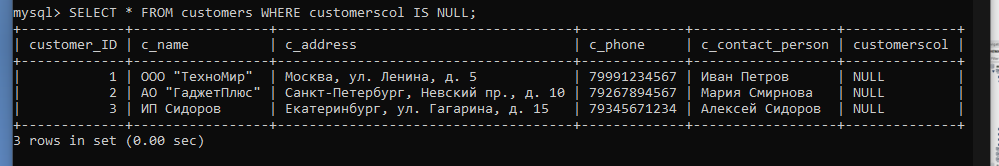
-- Оператор BETWEEN

SELECT \* FROM products WHERE p\_price BETWEEN 50000 AND 100000;



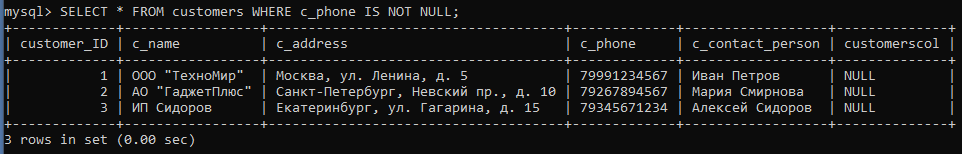
-- Оператор IS NULL

SELECT \* FROM customers WHERE customerscol IS NULL;



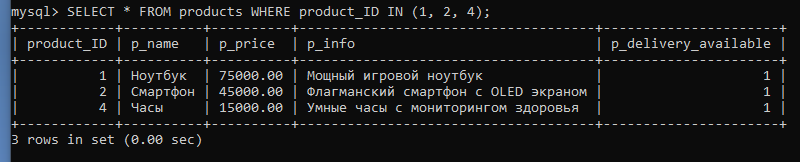
-- Оператор IS NOT NULL

SELECT \* FROM customers WHERE c\_phone IS NOT NULL;



-- Оператор IN

SELECT \* FROM products WHERE product\_ID IN (1, 2, 4);



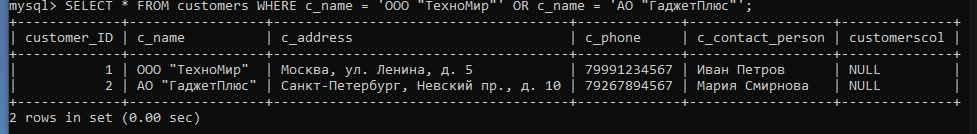
-- Оператор AND

SELECT \* FROM orders WHERE order\_quantity > 2 AND order\_date = '2025-02-15';



-- Оператор OR

SELECT \* FROM customers WHERE c\_name = 'ООО "ТехноМир"' OR c\_name = 'АО "ГаджетПлюс"';



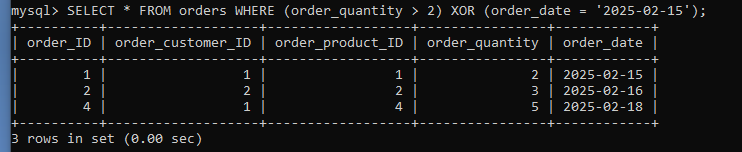
-- Оператор NOT

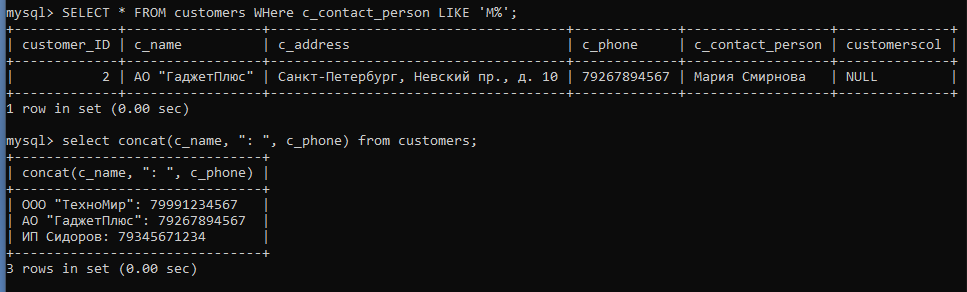
SELECT \* FROM products WHERE NOT p\_delivery\_available = 1;



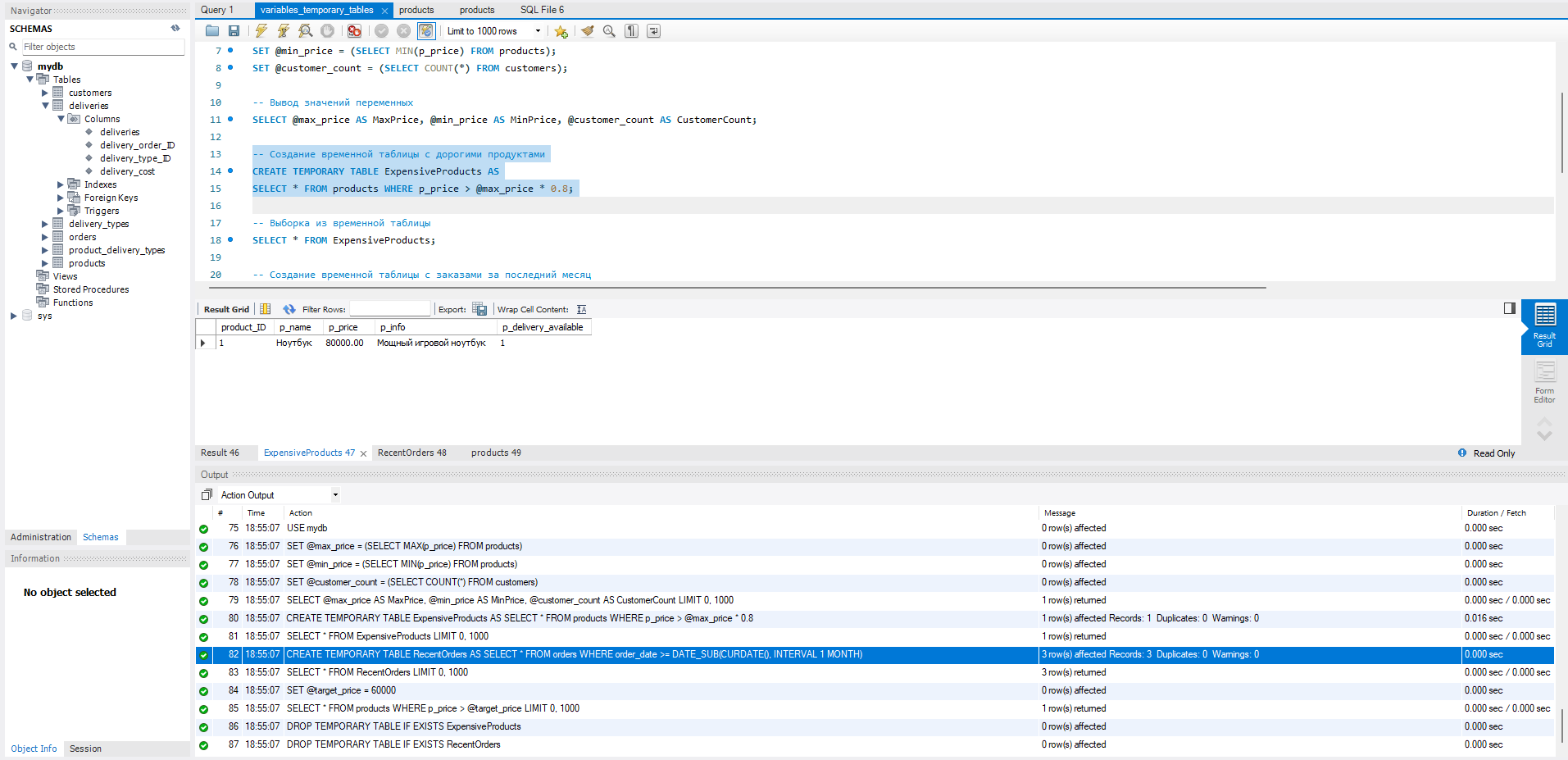
-- Оператор XOR

SELECT \* FROM orders WHERE (order\_quantity > 2) XOR (order\_date = '2025-02-15');





1. Создание переменных и временные таблицы.



Листинг

USE mydb;

-- Использование переменных и временных таблиц

-- Объявление переменных

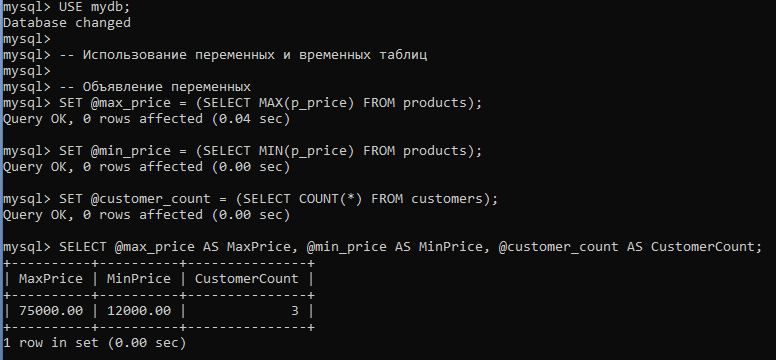
SET @max\_price = (SELECT MAX(p\_price) FROM products);

SET @min\_price = (SELECT MIN(p\_price) FROM products);

SET @customer\_count = (SELECT COUNT(\*) FROM customers);

-- Вывод значений переменных

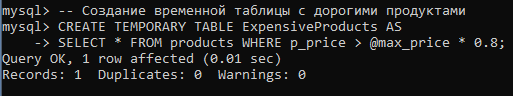
SELECT @max\_price AS MaxPrice, @min\_price AS MinPrice, @customer\_count AS CustomerCount;



-- Создание временной таблицы с дорогими продуктами

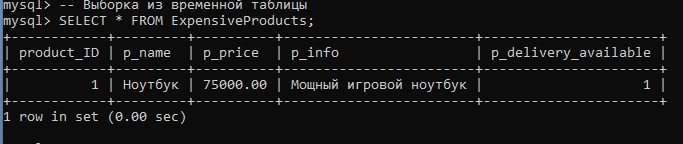
CREATE TEMPORARY TABLE ExpensiveProducts AS

SELECT \* FROM products WHERE p\_price > @max\_price \* 0.8;



-- Выборка из временной таблицы

SELECT \* FROM ExpensiveProducts;



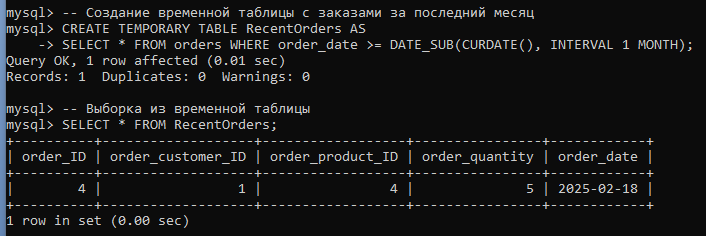
-- Создание временной таблицы с заказами за последний месяц

CREATE TEMPORARY TABLE RecentOrders AS

SELECT \* FROM orders WHERE order\_date >= DATE\_SUB(CURDATE(), INTERVAL 1 MONTH);

-- Выборка из временной таблицы

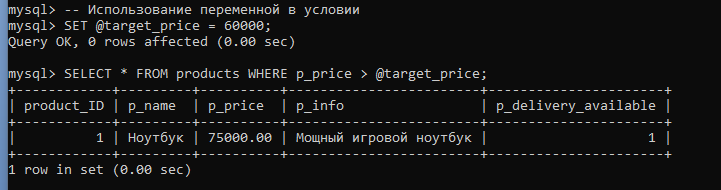
SELECT \* FROM RecentOrders;



-- Использование переменной в условии

SET @target\_price = 60000;

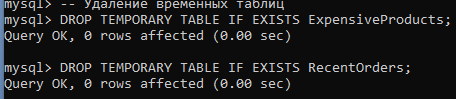
SELECT \* FROM products WHERE p\_price > @target\_price;



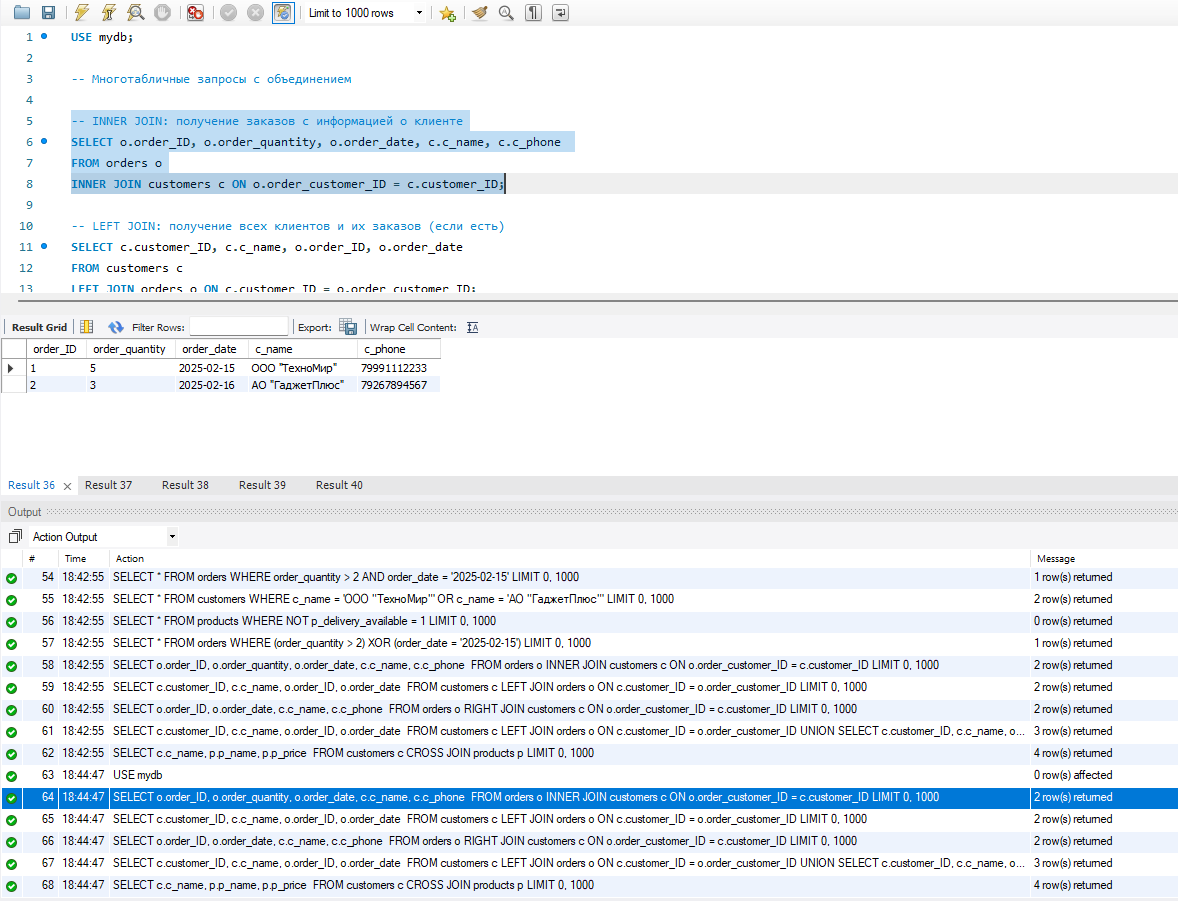
-- Удаление временных таблиц

DROP TEMPORARY TABLE IF EXISTS ExpensiveProducts;

DROP TEMPORARY TABLE IF EXISTS RecentOrders;



1. Для заданной предметной области построить многотабличные запросы на выборку с использованием объединения



**Листинг**

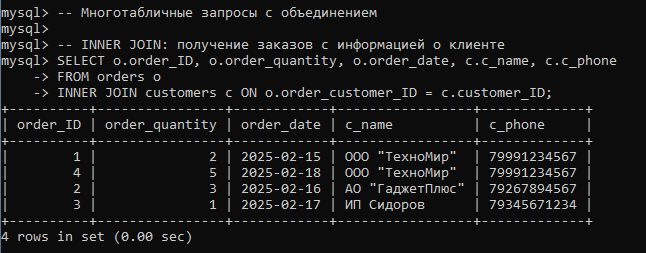
-- Многотабличные запросы с объединением

-- INNER JOIN: получение заказов с информацией о клиенте

SELECT o.order\_ID, o.order\_quantity, o.order\_date, c.c\_name, c.c\_phone

FROM orders o

INNER JOIN customers c ON o.order\_customer\_ID = c.customer\_ID;

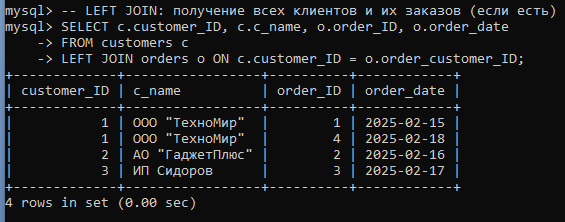


-- LEFT JOIN: получение всех клиентов и их заказов (если есть)

SELECT c.customer\_ID, c.c\_name, o.order\_ID, o.order\_date

FROM customers c

LEFT JOIN orders o ON c.customer\_ID = o.order\_customer\_ID;

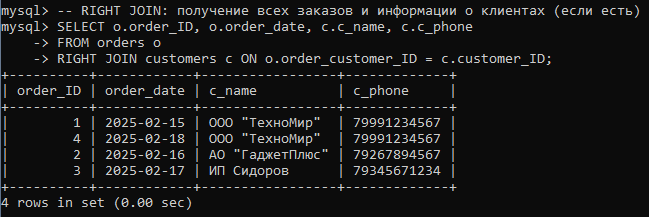


-- RIGHT JOIN: получение всех заказов и информации о клиентах (если есть)

SELECT o.order\_ID, o.order\_date, c.c\_name, c.c\_phone

FROM orders o

RIGHT JOIN customers c ON o.order\_customer\_ID = c.customer\_ID;



-- FULL OUTER JOIN (имитация через UNION): объединение клиентов и заказов

SELECT c.customer\_ID, c.c\_name, o.order\_ID, o.order\_date

FROM customers c

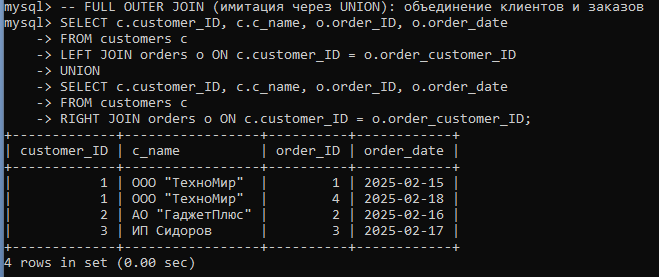
LEFT JOIN orders o ON c.customer\_ID = o.order\_customer\_ID

UNION

SELECT c.customer\_ID, c.c\_name, o.order\_ID, o.order\_date

FROM customers c

RIGHT JOIN orders o ON c.customer\_ID = o.order\_customer\_ID;

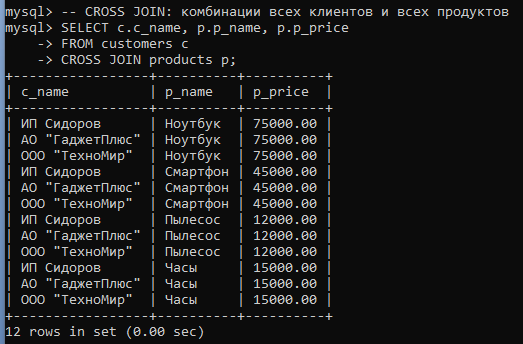


-- CROSS JOIN: комбинации всех клиентов и всех продуктов

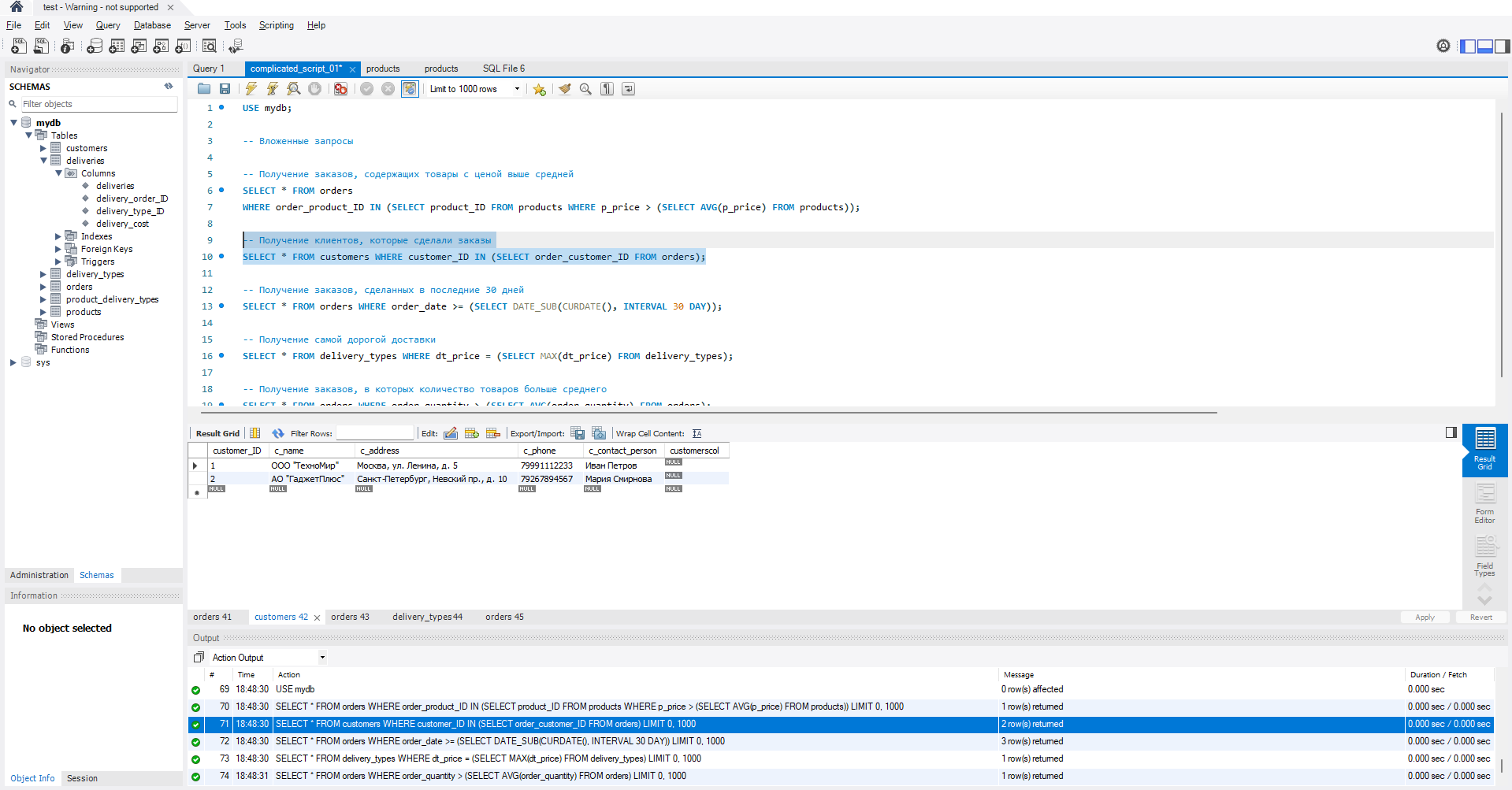
SELECT c.c\_name, p.p\_name, p.p\_price

FROM customers c

CROSS JOIN products p;



1. Для заданной предметной области построить запросы на выборку, содержащие вложенные запросы;



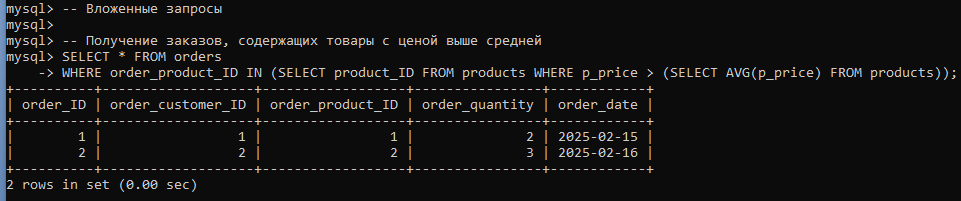
Листинг

-- Вложенные запросы

-- Получение заказов, содержащих товары с ценой выше средней

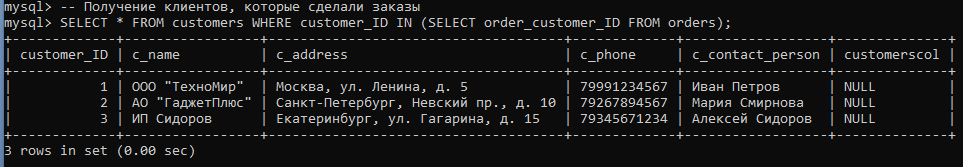
SELECT \* FROM orders

WHERE order\_product\_ID IN (SELECT product\_ID FROM products WHERE p\_price > (SELECT AVG(p\_price) FROM products));



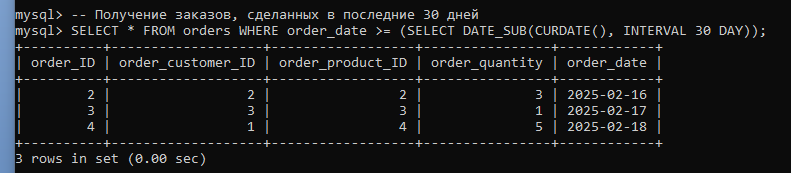
-- Получение клиентов, которые сделали заказы

SELECT \* FROM customers WHERE customer\_ID IN (SELECT order\_customer\_ID FROM orders);



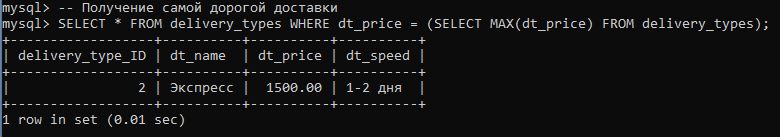
-- Получение заказов, сделанных в последние 30 дней

SELECT \* FROM orders WHERE order\_date >= (SELECT DATE\_SUB(CURDATE(), INTERVAL 30 DAY));



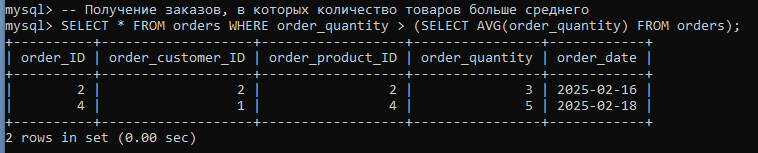
-- Получение самой дорогой доставки

SELECT \* FROM delivery\_types WHERE dt\_price = (SELECT MAX(dt\_price) FROM delivery\_types);



-- Получение заказов, в которых количество товаров больше среднего

SELECT \* FROM orders WHERE order\_quantity > (SELECT AVG(order\_quantity) FROM orders);



**Вывод:**

Построили не менее десяти простых запросов на выборку с использованием операторов и функций MySQL,изучили переменные SQL и временные таблицы, построили многотабличные запросы на выборку с использованием объединения, построили запросы на выборку, содержащие вложенные запросы