**Базы данных и SQL. Обучение в записи**

**Урок 6. Семинар: SQL – выборка данных, сортировка, агрегатные функции**

**ЧАСТЬ I.**

***Используя операторы языка SQL, создать таблицы SALESPEOPLE, CUSTOMERS, ORDERS. Заполнить их данными.***

*DROP DATABASE IF EXISTS seminar\_3;*

*CREATE DATABASE IF NOT EXISTS seminar\_3;*

*USE seminar\_3;*

*DROP TABLE IF EXISTS salespeople;*

*CREATE TABLE salespeople*

*(*

*snum INT,*

*sname VARCHAR(45),*

*city VARCHAR(45),*

*comm INT*

*);*

*DROP TABLE IF EXISTS customers;*

*CREATE TABLE customers*

*(*

*cnum INT,*

*cname VARCHAR(45),*

*city VARCHAR(45),*

*rating INT,*

*snum INT*

*);*

*DROP TABLE IF EXISTS orders;*

*CREATE TABLE orders*

*(*

*onum INT,*

*amt DECIMAL(10,2),*

*odate DATE,*

*cnum INT,*

*snum INT*

*);*

*INSERT salespeople (snum, sname, city, comm)*

*VALUES*

*(1001, 'Peel', 'London', 12),*

*(1002, 'Serres', 'San Jose', 13),*

*(1004, 'Motika', 'London', 11),*

*(1007, 'Rifkin', 'Barcelona', 15),*

*(1003, 'Axelrod', 'New York', 10);*

*INSERT customers (cnum, cname, city, rating, snum)*

*VALUES*

*(2001, 'Hoffman', 'London', 100, 1001),*

*(2002, 'Giovanni', 'Rome', 200, 1003),*

*(2003, 'Liu', 'SanJose', 200, 1002),*

*(2004, 'Grass', 'Berlin', 300, 1002),*

*(2006, 'Clemens', 'London', 100, 1001),*

*(2008, 'Cisneros', 'SanJose', 300, 1007),*

*(2007, 'Pereira', 'Rome', 100, 1004);*

*INSERT orders (onum, amt, odate, cnum, snum)*

*VALUES*

*(3001, 18.69, '1990-03-10', 2008, 1007),*

*(3003, 767.19, '1990-03-10', 2001, 1001),*

*(3002, 1900.10, '1990-03-10', 2007, 1004),*

*(3005, 5160.45, '1990-03-10', 2003, 1002),*

*(3006, 1098.16, '1990-03-10', 2008, 1007),*

*(3009, 1713.23, '1990-04-10', 2002, 1003),*

*(3007, 75.75, '1990-04-10', 2004, 1002),*

*(3008, 4723.00, '1990-05-10', 2006, 1001),*

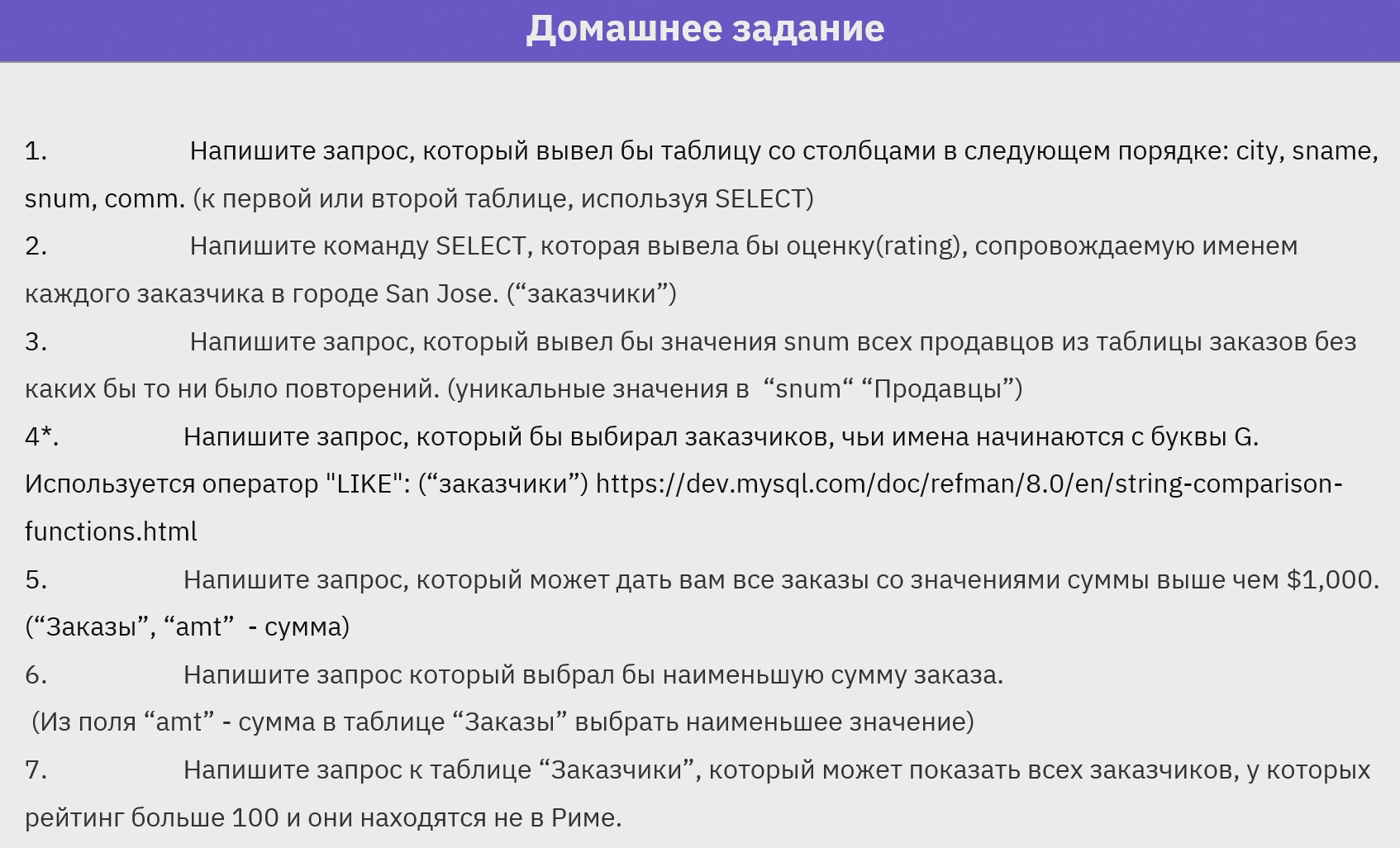
*(3010, 1309.95, '1990-06-10', 2004, 1002),*

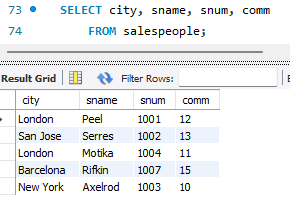
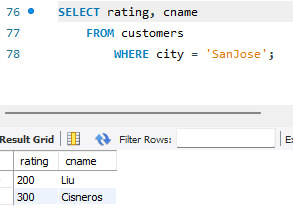
*(3011, 9891.88, '1990-04-10', 2006, 1001);*

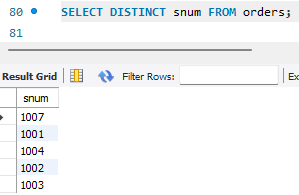
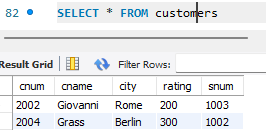
*SELECT \* FROM customers;*

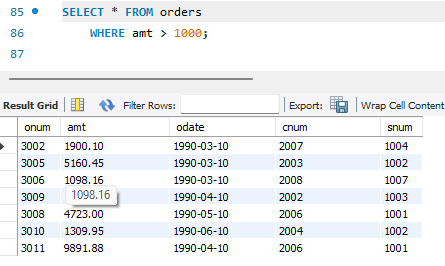
*SELECT \* FROM orders;*

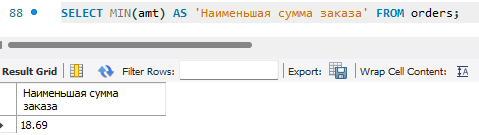
*SELECT \* FROM salespeople;*

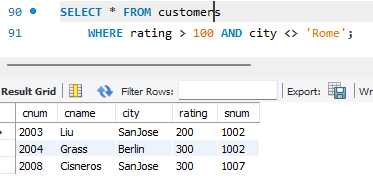
****

**1.  2. **

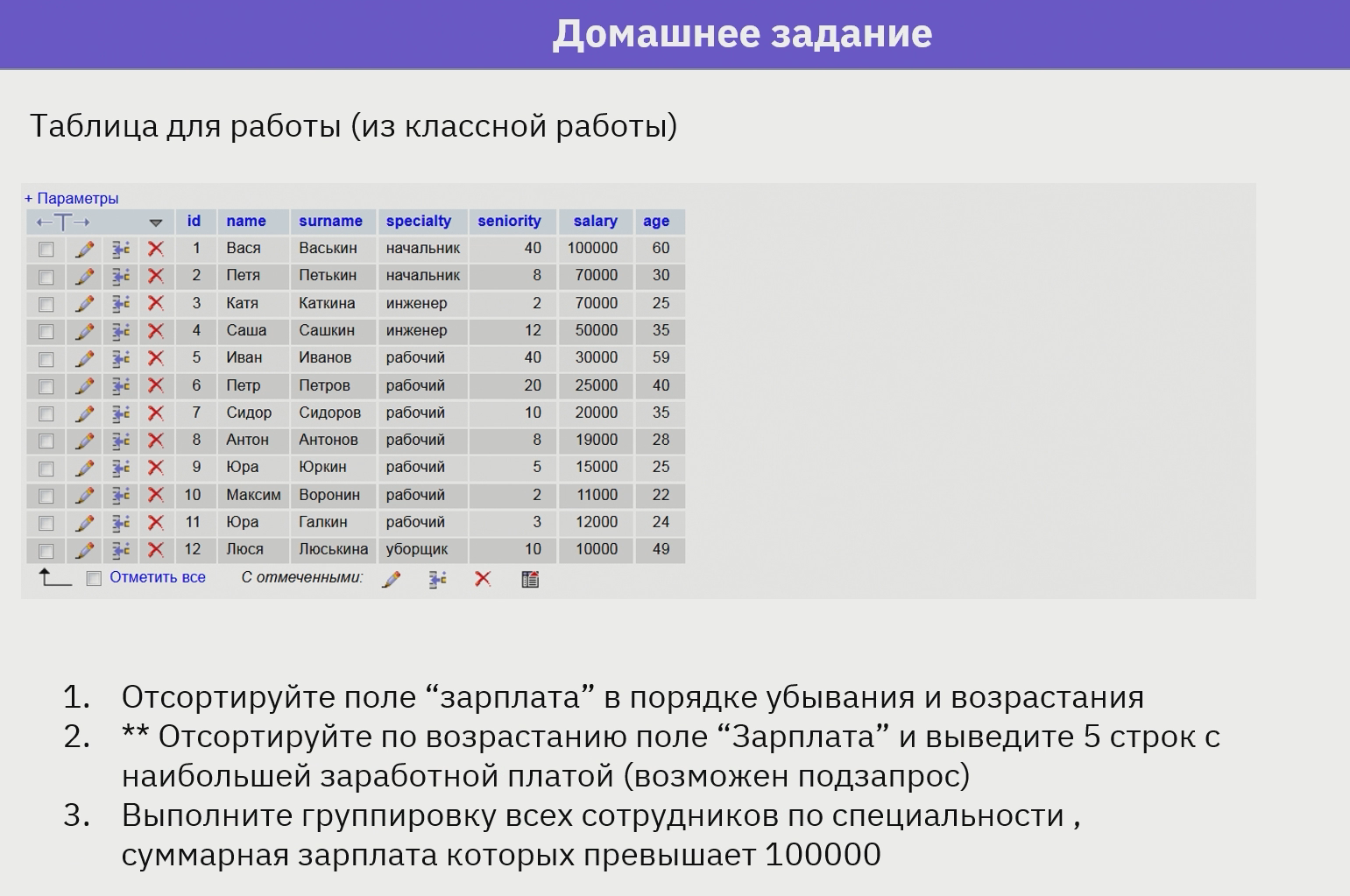
**3.  4. **

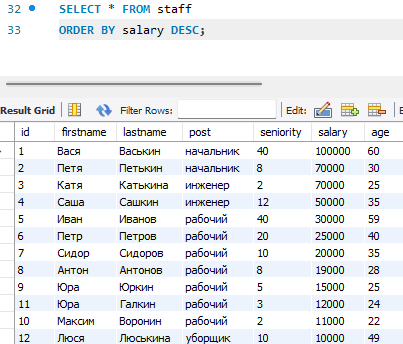
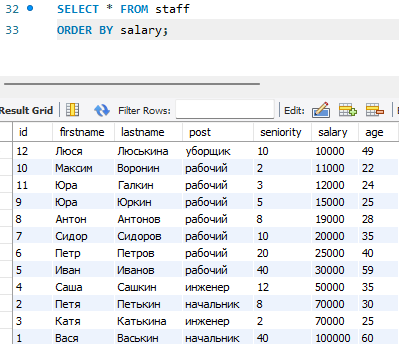
**5. **

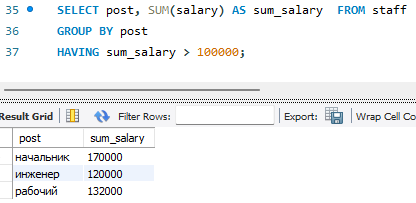
**6. **

**7. **

**ЧАСТЬ II.**

****

**1.  2. **

**3. **

**ЧАСТЬ III.**

1. **Ход выполнения задания 0.**

*Запрос на создание начальной таблицы, выполнять задания на основе этих данных.*

*Вы можете воспользоваться заготовкой fiddle* [*https://dbfiddle.uk/S8qGS2s4*](https://dbfiddle.uk/S8qGS2s4)

*Или использовать код ниже для создания таблиц*

*USE seminar\_3;*

*-- Создание таблицы Customers*

*DROP TABLE IF EXISTS Customers;*

*CREATE TABLE Customers (*

*customer\_id INT PRIMARY KEY,*

*customer\_name VARCHAR(255)*

*);*

*-- Создание таблицы Orders*

*DROP TABLE IF EXISTS Orders;*

*CREATE TABLE Orders (*

*order\_id INT PRIMARY KEY,*

*customer\_id INT,*

*order\_date DATE,*

*total\_amount DECIMAL(10, 2),*

*shipper\_id INT,*

*FOREIGN KEY (customer\_id) REFERENCES Customers(customer\_id)*

*);*

*-- Создание таблицы Shippers*

*DROP TABLE IF EXISTS Shippers;*

*CREATE TABLE Shippers (*

*shipper\_id INT PRIMARY KEY,*

*shipper\_name VARCHAR(255)*

*);*

*-- Создание таблицы Products*

*DROP TABLE IF EXISTS Products;*

*CREATE TABLE Products (*

*product\_id INT PRIMARY KEY,*

*product\_name VARCHAR(255),*

*category\_id INT,*

*author VARCHAR(255),*

*price DECIMAL(10, 2)*

*);*

*-- Создание таблицы OrderDetails*

*DROP TABLE IF EXISTS OrderDetails;*

*CREATE TABLE OrderDetails (*

*order\_detail\_id INT PRIMARY KEY,*

*order\_id INT,*

*product\_id INT,*

*quantity INT,*

*FOREIGN KEY (order\_id) REFERENCES Orders(order\_id),*

*FOREIGN KEY (product\_id) REFERENCES Products(product\_id)*

*);*

*-- Создание таблицы Categories*

*DROP TABLE IF EXISTS Categories;*

*CREATE TABLE Categories (*

*category\_id INT PRIMARY KEY,*

*category\_name VARCHAR(255)*

*);*

*-- Наполнение таблиц данными*

*INSERT INTO Customers (customer\_id, customer\_name)*

*VALUES*

*(1, 'Иван Иванов'), (2, 'Мария Смирнова'), (3, 'Алексей Попов'), (4,*

*'Наталья Кузнецова'), (5, 'Дмитрий Васильев'),*

*(6, 'Ольга Петрова'), (7, 'Андрей Сидоров'), (8, 'Елена Алексеева'),*

*(9, 'Сергей Морозов'), (10, 'Ирина Фёдорова'),*

*(11, 'Андрей Иванов'), (12, 'Екатерина Мартынова');*

*INSERT INTO Shippers (shipper\_id, shipper\_name)*

*VALUES*

*(1, 'СДЕК'), (2, 'Почта России'), (3, 'ПЭК');*

*INSERT INTO Categories (category\_id, category\_name)*

*VALUES*

*(1, 'Художественная литература'), (2, 'Наука'), (3, 'Мистика');*

*INSERT INTO Products (product\_id, product\_name, category\_id, author, price)*

*VALUES*

*(1, '1984', 1, 'Джордж Оруэлл', 250),*

*(2, 'Убить пересмешника', 1, 'Харпер Ли', 300),*

*(3, 'Великий Гэтсби', 1, 'Фрэнсис Скотт Фицджеральд', 200),*

*(4, 'Краткая история времени', 2, 'Стивен Хокинг', 320),*

*(5, 'Собака Баскервилей', 3, 'Артур Конан Дойл', 350),*

*(6, 'Моби Дик', 1, 'Герман Мелвилл', 400),*

*(7, 'Скотный двор', 1, 'Джордж Оруэлл', 220),*

*(8, 'Похвала Каталонии', 1, 'Джордж Оруэлл', 180),*

*(9, 'Дневник Анны Франк', 1, 'Анна Франк', 300),*

*(10, 'Краткая история времени', 2, 'Стивен Хокинг', 320);*

*-- Добавление заказов и деталей заказов с реалистичным распределением*

*INSERT INTO Orders (order\_id, customer\_id, order\_date, total\_amount, shipper\_id)*

*VALUES*

*(1, 1, '2023-01-10', 750, 1),*

*(2, 3, '2023-01-12', 820, 2),*

*(3, 2, '2023-01-15', 600, 3),*

*(4, 4, '2023-02-01', 670, 1),*

*(5, 6, '2023-02-05', 550, 2),*

*(6, 3, '2023-02-10', 400, 3),*

*(7, 7, '2023-03-01', 320, 1),*

*(8, 8, '2023-03-05', 500, 2),*

*(9, 1, '2023-03-10', 270, 3),*

*(10, 5, '2023-03-15', 350, 1),*

*(11, 4, '2023-03-20', 420, 2),*

*(12, 9, '2023-04-01', 300, 3),*

*(13, 5, '2023-04-05', 220, 1),*

*(14, 6, '2023-04-10', 500, 2),*

*(15, 8, '2023-04-15', 450, 3),*

*(16, 10, '2023-05-01', 350, 1),*

*(17, 4, '2023-05-05', 250, 2),*

*(18, 9, '2023-05-10', 300, 3),*

*(19, 7, '2023-05-15', 400, 1);*

*INSERT INTO OrderDetails (order\_detail\_id, order\_id, product\_id, quantity)*

*VALUES*

*(1, 1, 1, 1), (2, 1, 2, 1), (3, 1, 4, 1),*

*(4, 2, 3, 1), (5, 2, 5, 1), (6, 2, 7, 1),*

*(7, 3, 6, 1), (8, 3, 9, 1),*

*(9, 4, 1, 1), (10, 4, 8, 1),*

*(11, 5, 2, 1), (12, 5, 5, 1),*

*(13, 6, 3, 1),*

*(14, 7, 4, 1), (15, 7, 9, 1),*

*(16, 8, 10, 1),*

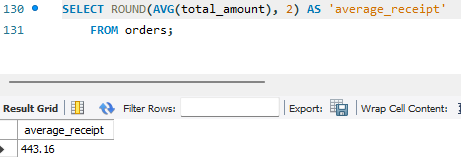
*(17, 9, 5, 1), (18, 9, 6, 1),*

*(19, 10, 7, 1);*

1. **Ход выполнения задания 1.**

***Средний чек по заказам***

*Выведите средний чек (average\_receipt) для заказов.*

**

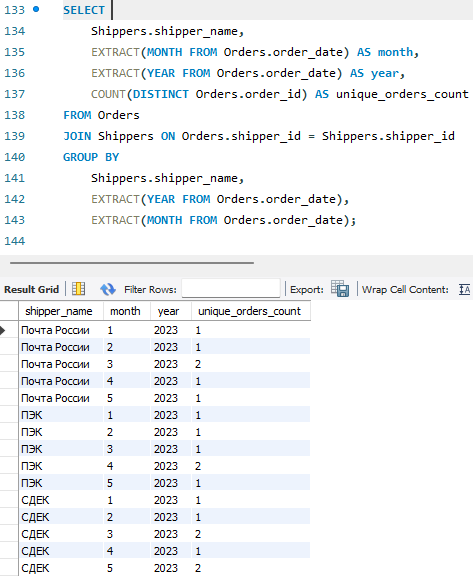
1. **Ход выполнения задания 2.**

***Количество заказов по перевозчикам***

*Выведите имя перевозчика, месяц и год заказа, а также количество уникальных заказов, доставленных каждым перевозчиком в каждый месяц и год.*

*Подсказка:*

*● Используйте EXTRACT, GROUP BY и COUNT.*

**

1. **Ход выполнения задания 3.**

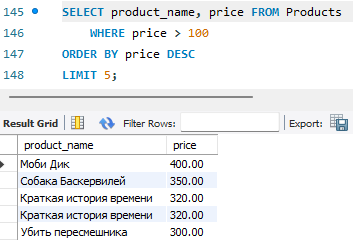
***Продукты по цене***

*Выведите название и цену книг, которые стоят более 100 единиц. Отсортируйте*

*результат по цене в порядке убывания и ограничьте вывод 5 результатами.*

*Подсказка:*

*● Используйте WHERE, ORDER BY и LIMIT.*

**

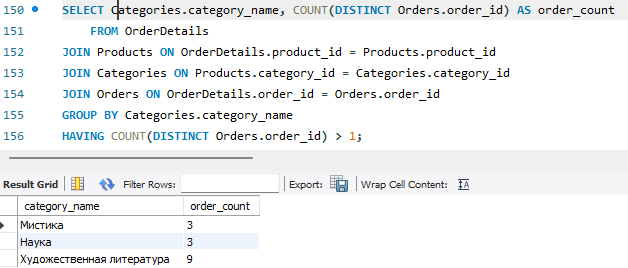
1. **Ход выполнения задания 4.**

***Количество заказов по категориям книг***

*Выведите категорию книг и количество заказов, которые содержат книги этой категории. Показать только те категории, которые имеют больше одного заказа.*

*Подсказка:*

*● Используйте GROUP BY, HAVING и агрегатную функцию COUNT.*

**

1. **Ход выполнения задания 5.**

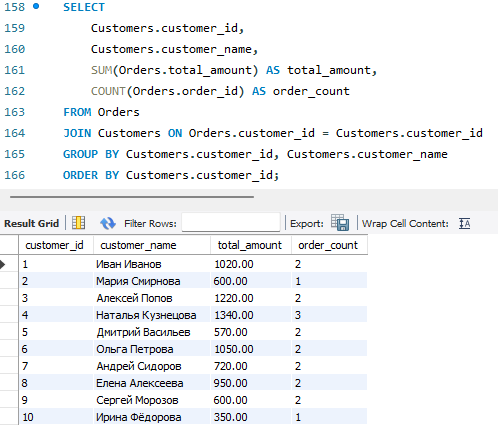
***Сумма и количество заказов по клиентам***

*Выведите идентификатор клиента, имя клиента, сумму и количество его заказов.*

*Отсортируйте результат по идентификатору клиента.*

*Подсказка:*

*● Используйте GROUP BY и агрегатные функции SUM и COUNT.*

**