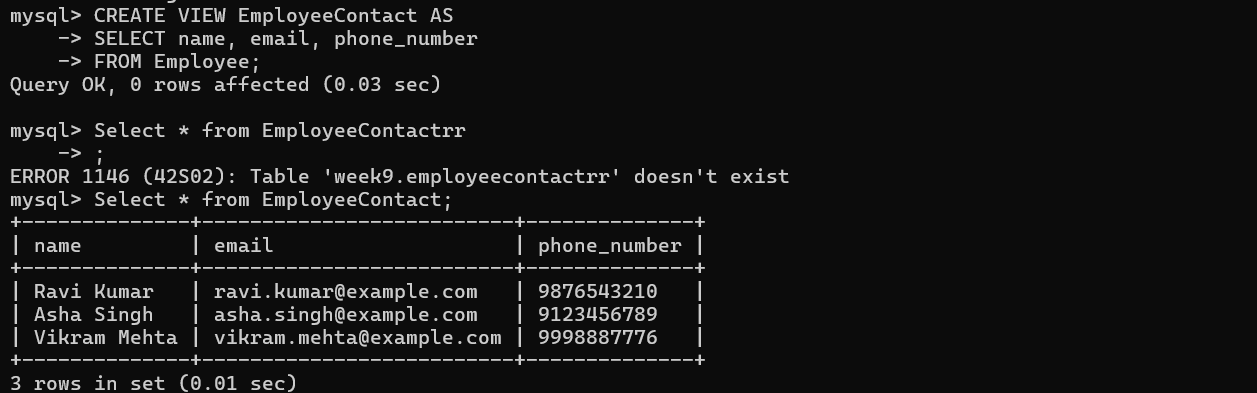
**Objective:**

Practice creating and dropping database views on complex data sets that span across multiple related tables. These views should help simplify data retrieval from multiple joined tables, apply filters, and perform aggregations for meaningful insights. The focus is on understanding how to create, modify, and delete views effectively

**Simple views:**

**Problem statement**: The HR department needs access to employee contact information, such as name, email, and phone number, but should not see sensitive information like salary or personal identification numbers.

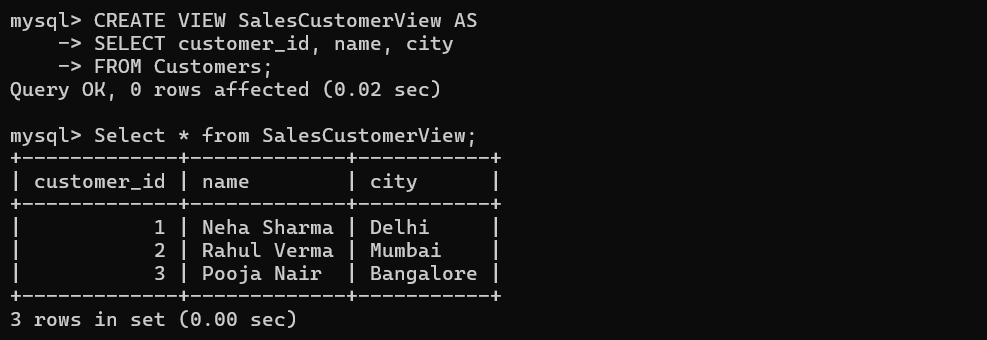
**Query:** Create a simple view that includes only the fields HR staff needs, hiding sensitive columns.



**scenario 2: Creating a Custom View for a Specific Department**

**Problem statement**: The Sales department only needs access to information about customers, specifically their IDs, names, and cities. The rest of the data (like customer credit scores or billing addresses) is irrelevant for their daily tasks.

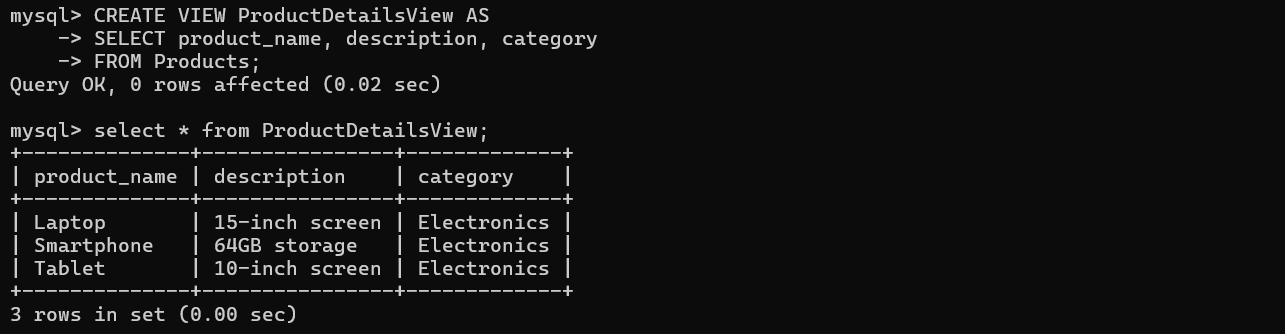
Problem statement: Create a view with only the necessary columns, focusing on the data that the Sales team needs.



**Scenario 3: Standardized View of Product Information**

**Problem statement**: The Product Management team needs a consistent view of product details to analyse offerings. They only need the product name, description, and category, but not inventory or supplier details.

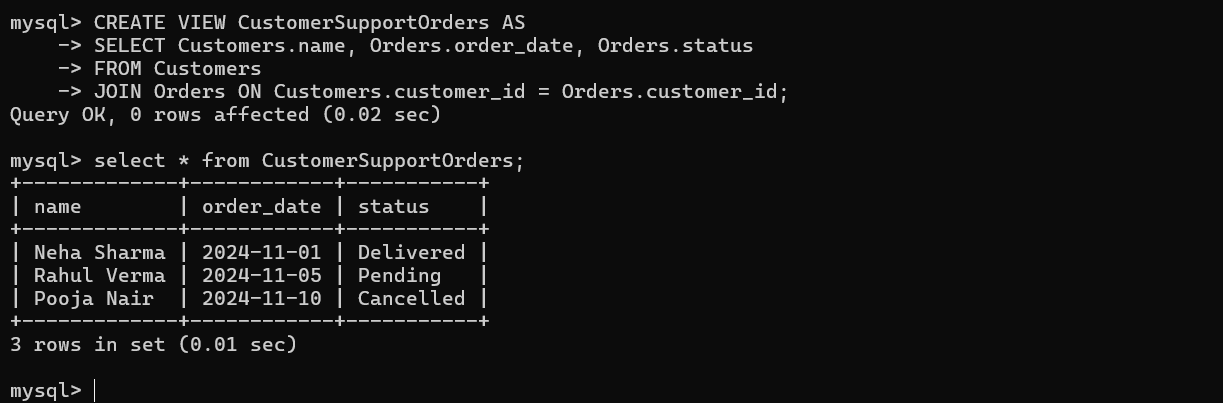
Query: Create a simple view that presents product details in a standardized format.



**Scenario 4: Presenting Simplified Customer Orders for Customer Support**

**Problem statement**: Customer Support agents need to see the latest orders placed by customers to assist with inquiries. However, they should only have access to customer names, order dates, and status—not financial details like order amounts or payment info.

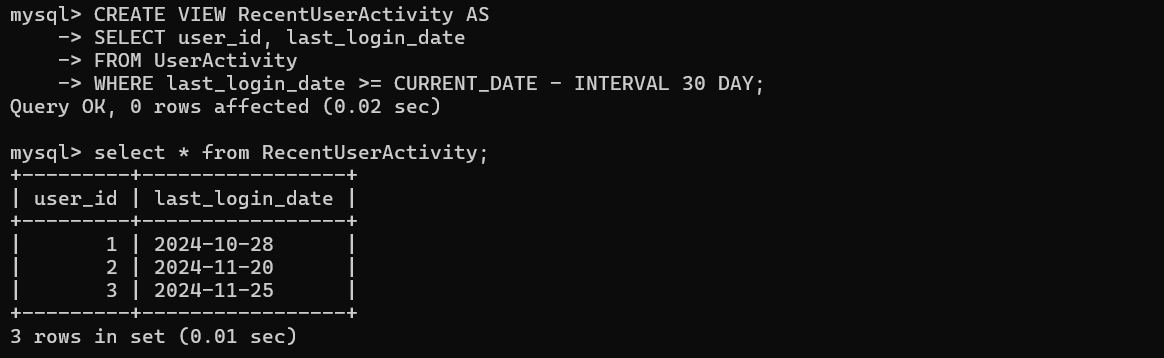
Query: Create a view that shows only the essential fields for customer support tasks.



**Scenario 5: Creating a View for Recent Activity**

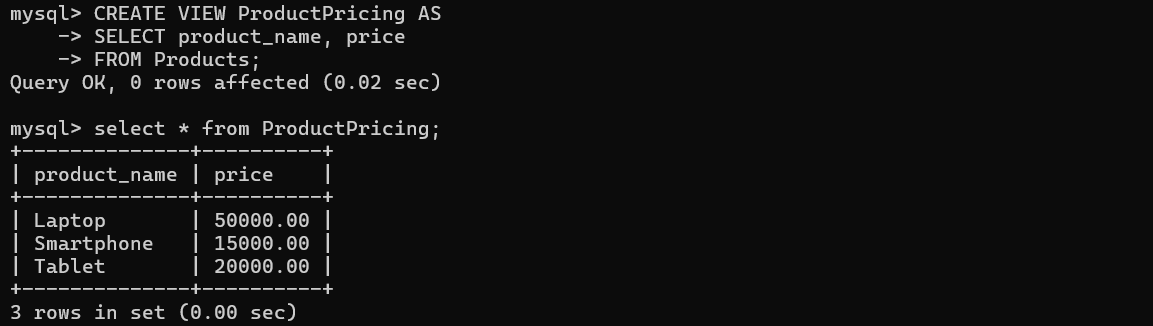
**Problem statement**: A team wants a view of recent user activity from a `UserActivity` table, showing only the user ID and last login date for users who logged in within the last 30 days.

Query: Create a view that filters data to show only recent activity.



**Scenario 6: View for Product Pricing Information**

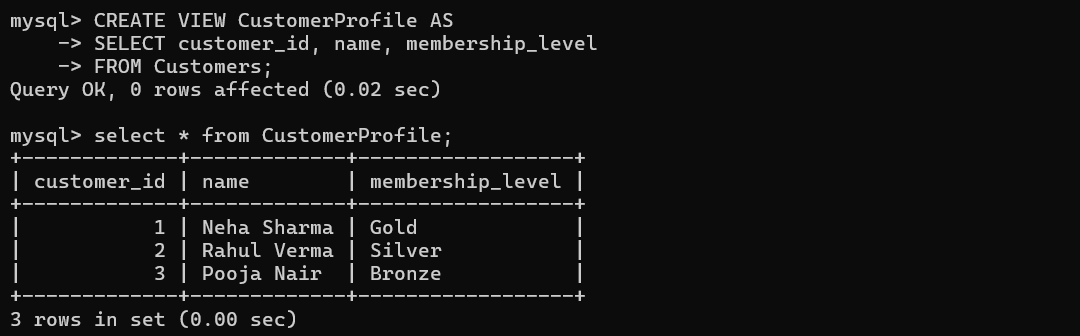
**Problem statement**: A marketing analyst needs access to product pricing information, specifically the product name and price, but should not see inventory levels or internal cost details.



**Scenario 7: Simplified Customer Profile View**

**Problem statement**: The Customer Relations team needs quick access to customer profiles, focusing only on fields like `customer\_id`, `name`, and `membership\_level` to manage loyalty programs.

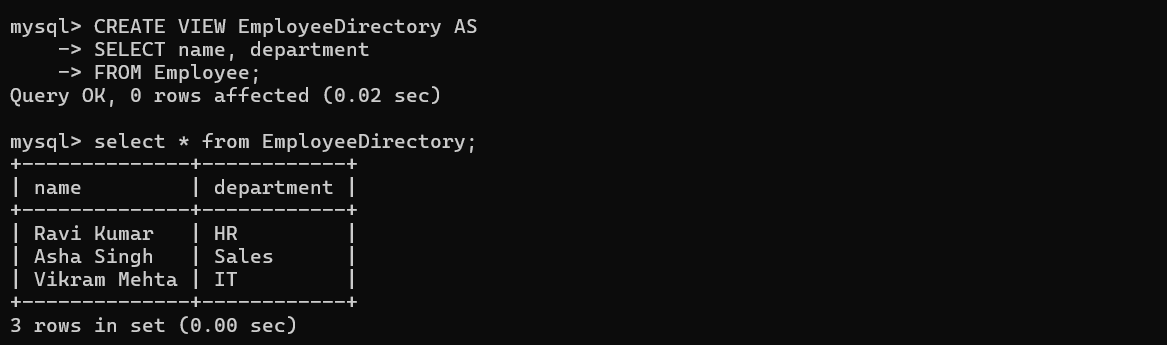
Query Approach: Create a view to show only essential profile fields.



**Scenario 8: Summarized Employee Directory**

Scenario: A general employee directory is needed to display employee names and departments to all users without showing sensitive personal details.

Query: Create a view with only the name and department fields from the employee data.

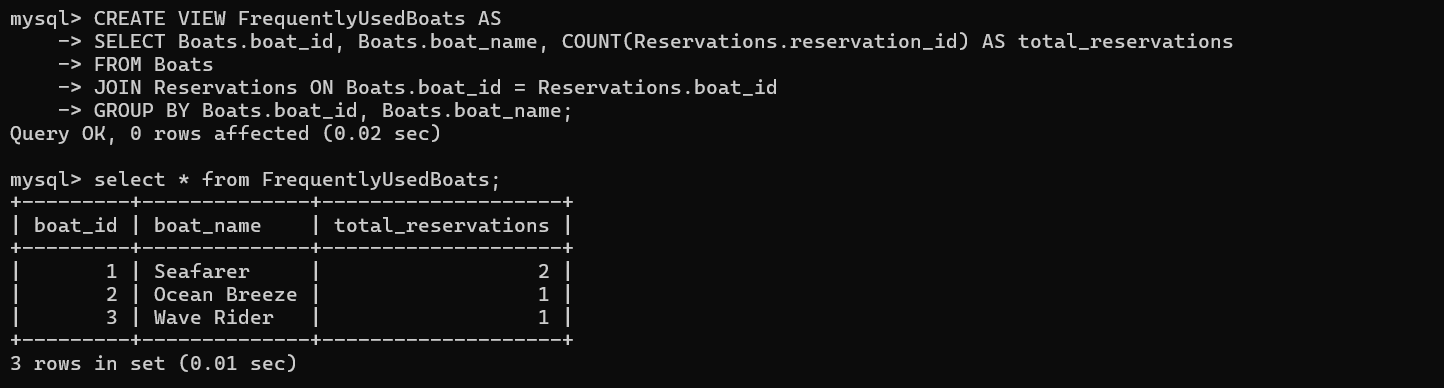


**Complex views:**

**Scenario 1: Frequently Used Boats**

Scenario: The marina management wants to see which boats are most frequently used, including the total number of reservations per boat**, the Frequently Used Boats contains the tables: Sailors**, **Boats**,**Reservations**:

**Query: Create a view that joins Boats and Reservations, counting reservations grouped by boat\_id.**



**Scenario: E-commerce Database**

You are a database administrator for an e-commerce platform. The database consists of the following tables

**1. Products**

**2.Categories**

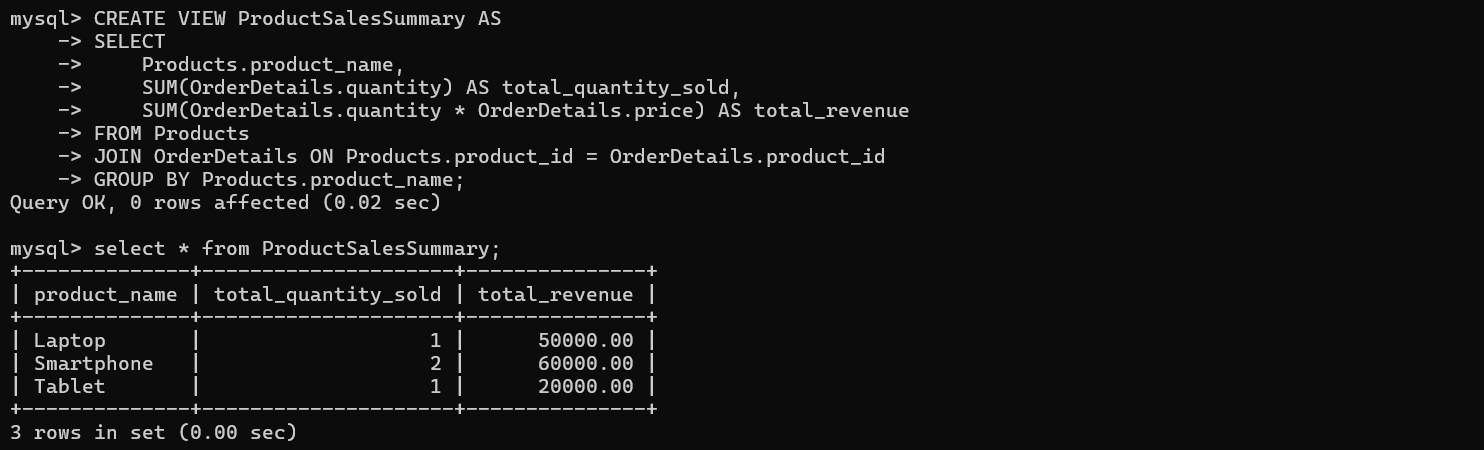
**3. Orders**

**4. Order Details**

**5. Customers**

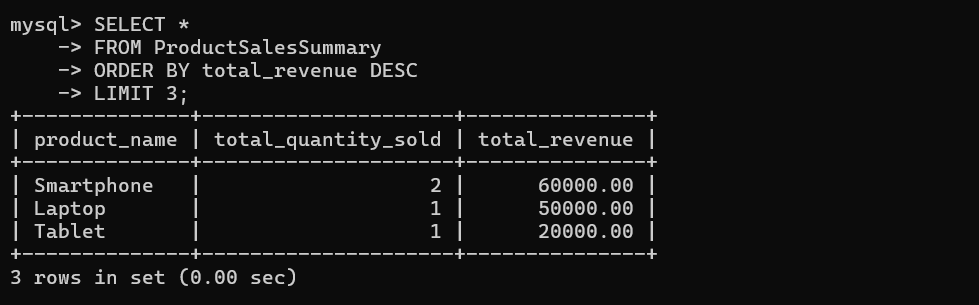
**Question 1:**

**Create a view `ProductSalesSummary` to show the total sales revenue for each product, including product name, total quantity sold, and total revenue.**



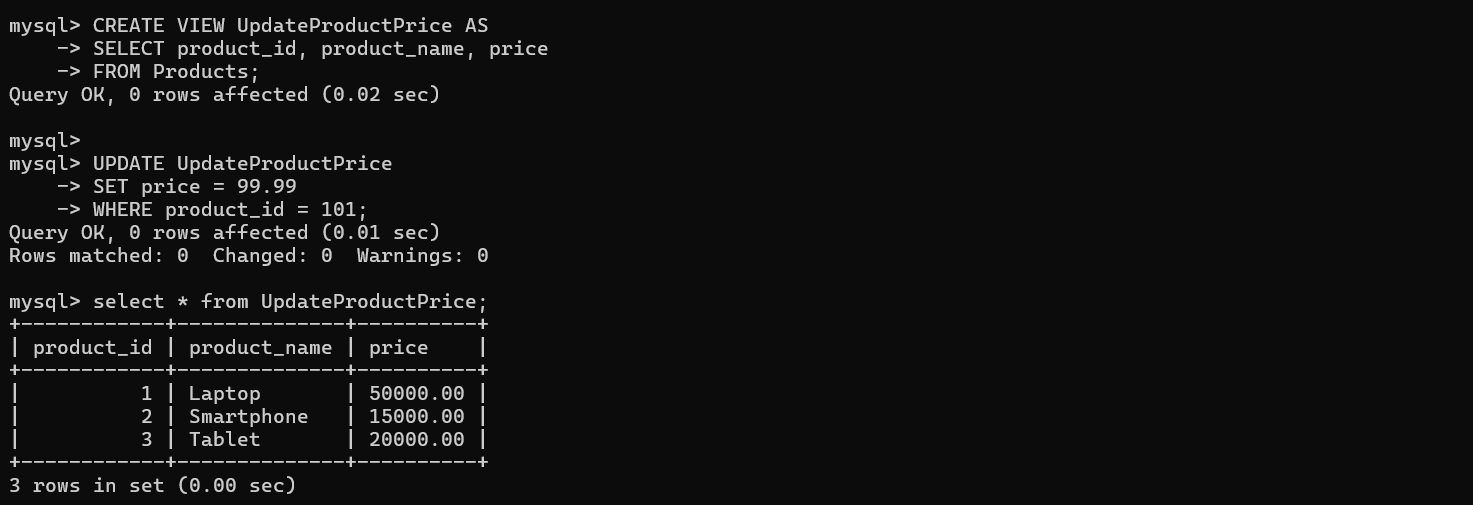
**Question 2:**

**Write a query to fetch the top 3 products with the highest sales revenue from the `ProductSalesSummary` view.**



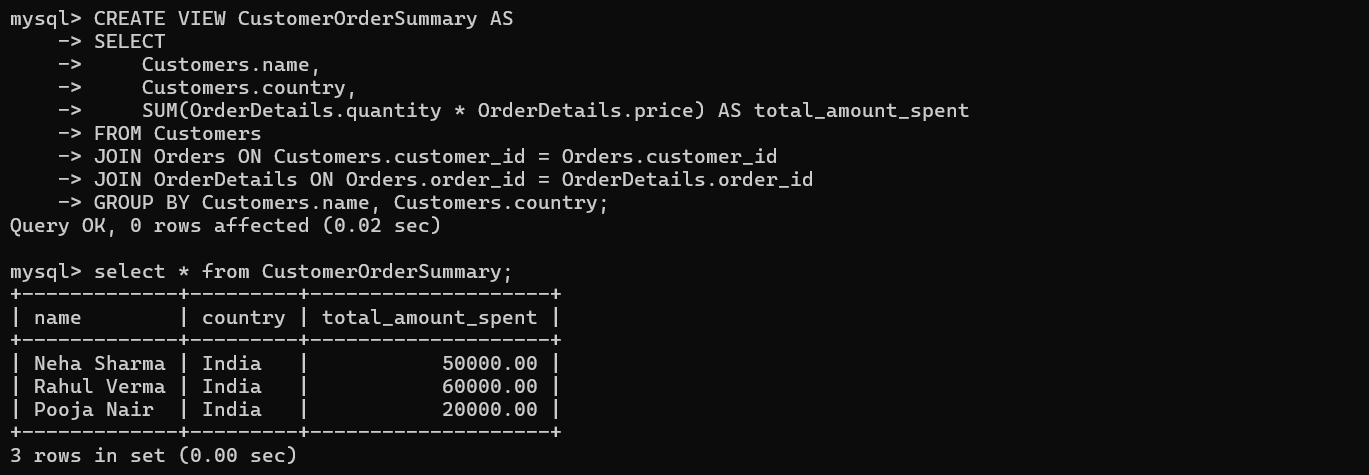
**Question 3:**

**You notice that the `Price` of a specific product (`ProductID = 101`) is incorrect. Update the price in the `Products` table via a view.**



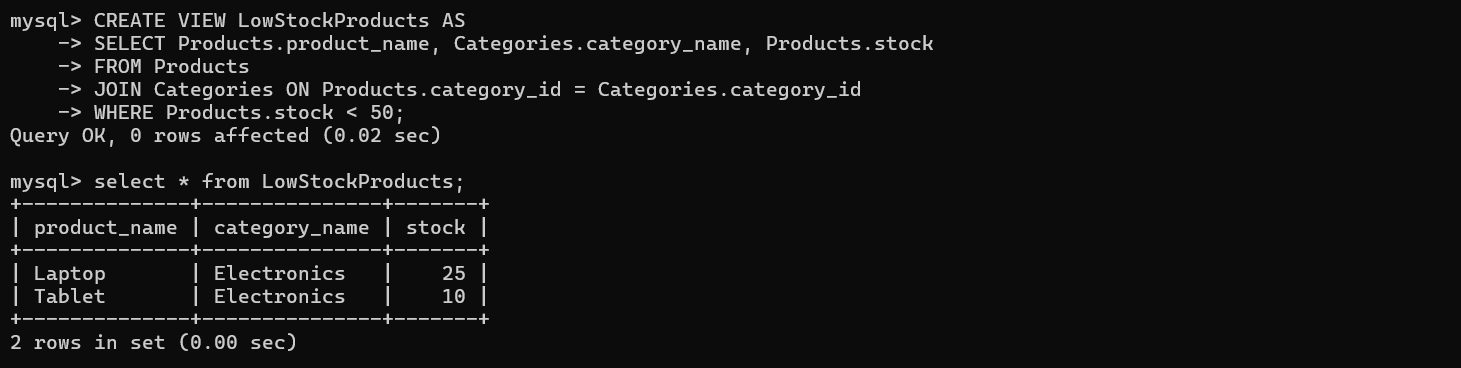
**Question 4:**

**Create a view `CustomerOrderSummary` to display each customer's name, their country, and the total amount spent across all orders.**



**Question 5:**

**Create a view `LowStockProducts` to display products with stock below 50, including product name, category, and stock level.**



**Question 6:**

**Dealing with Non-Updatable Views**

**Why might the view `CustomerOrderSummary` not be updatable? Provide a solution to handle updates.?**