**End of Module Assessment**

**TASK 1: Query**

*SELECT OrderID, ProductID, Quantity, Total*

*FROM Orders WHERE ProductID IN (*

*SELECT ProductID FROM Products*

*WHERE Price > 25);*

**Output:**

A screenshot of a computer

AI-generated content may be incorrect.

**TASK 2: Query**

*SELECT c.CustomerName, SUM(o.Quantity) AS TotalQuantity*

*FROM Customers c*

*JOIN Orders o*

*ON c.CustomerID = o.CustomerID*

*GROUP BY c.CustomerName*

*ORDER BY TotalQuantity DESC;*

**Output:**

A screenshot of a computer

AI-generated content may be incorrect.

**TASK 3: Query**

*SELECT p.ProductName, SUM(Total) AS TotalSales,*

*CASE*

*WHEN SUM(Total) > 300 THEN 'High Revenue'*

*ELSE 'Low Revenue'*

*END*

*AS RevenueCategory*

*FROM Orders o*

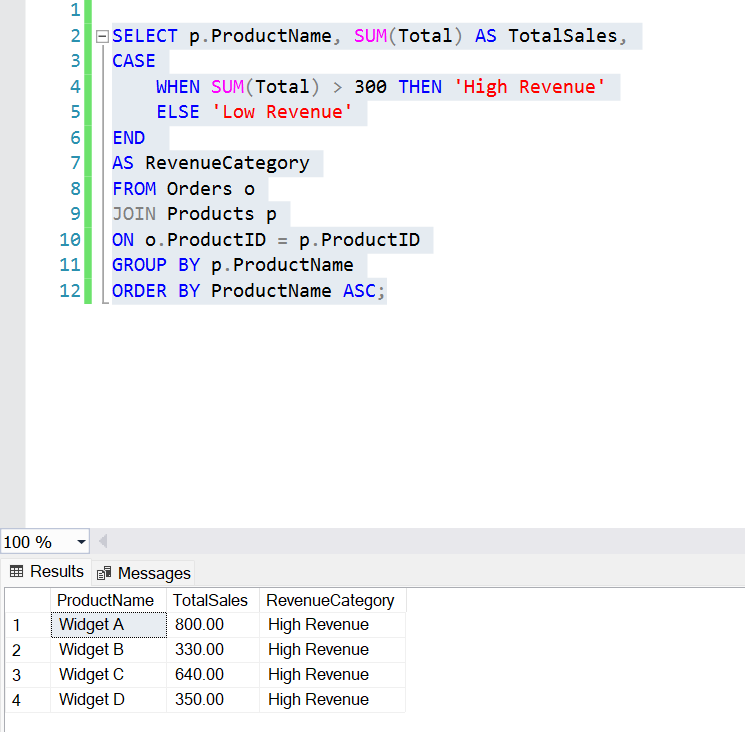
*JOIN Products p*

*ON o.ProductID = p.ProductID*

*GROUP BY p.ProductName*

*ORDER BY ProductName ASC;*

**Output:**



**TASK 4: Query**

*SELECT c.CustomerName, COUNT(\*) AS OrderCount*

*FROM Customers c*

*JOIN Orders o ON c.CustomerID = o.CustomerID*

*JOIN Products p ON o.ProductID = p.ProductID*

*WHERE*

*o.OrderDate >= '2023-12-01' AND o.OrderDate < '2024-01-01'*

*AND (o.Total) > 100*

*GROUP BY*

*c.CustomerName*

*HAVING*

*COUNT(\*) >= 2;*

**Output:**

A screenshot of a computer

AI-generated content may be incorrect.

**TASK 5: Query**

*SELECT o.OrderID, LEFT(c.CustomerName, 3) AS ShortCustomerName, p.ProductName, o.Total*

*From Orders o*

*JOIN Customers c ON c.CustomerID = o.CustomerID*

*JOIN Products p ON p.ProductID = o.ProductID*

*Order BY p.ProductName ASC;*

**Output:**

A screenshot of a computer

AI-generated content may be incorrect.

**TASK 6: Query**

*SELECT LOWER(p.ProductName) AS ProductNameLower, SUM(o.Quantity) AS TotalQuantityOrdered*

*FROM Products p*

*JOIN Orders o*

*ON p.ProductID = o.ProductID*

*GROUP BY LOWER(p.ProductName)*

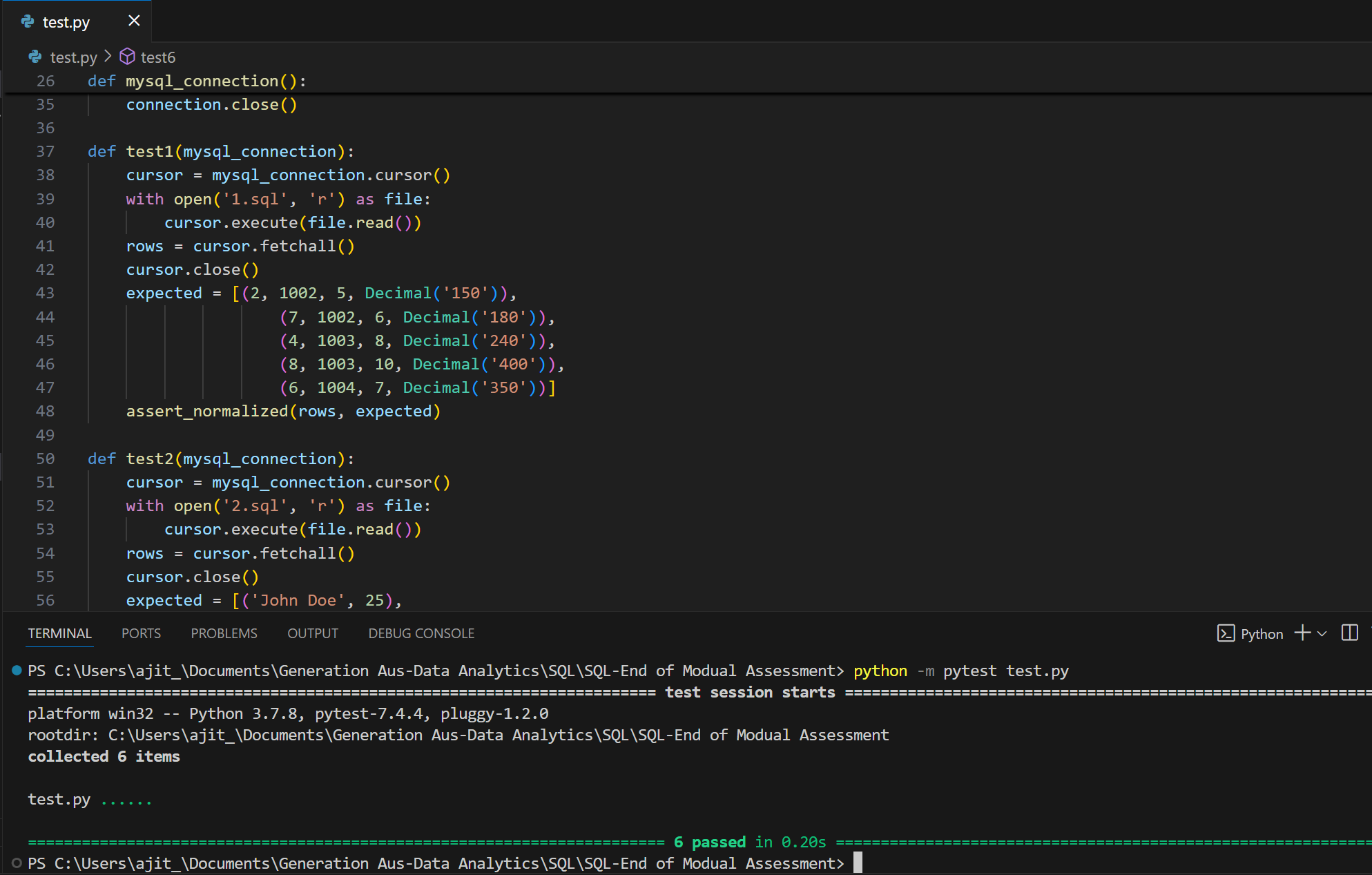
*ORDER BY ProductNameLower;*

**Output:**

A screenshot of a computer

AI-generated content may be incorrect.

**test.py Result:**

****