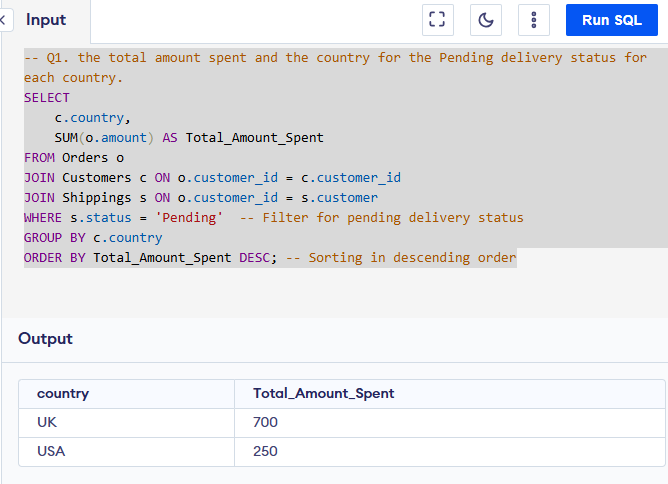
-- Q1. the total amount spent and the country for the Pending delivery status for each country.

Ans:

|  |
| --- |
| SELECT  c.country,  SUM(o.amount) AS Total\_Amount\_Spent  FROM Orders o  JOIN Customers c ON o.customer\_id = c.customer\_id  JOIN Shippings s ON o.customer\_id = s.customer  WHERE s.status = 'Pending' -- Filter for pending delivery status  GROUP BY c.country  ORDER BY Total\_Amount\_Spent DESC; -- Sorting in descending order |

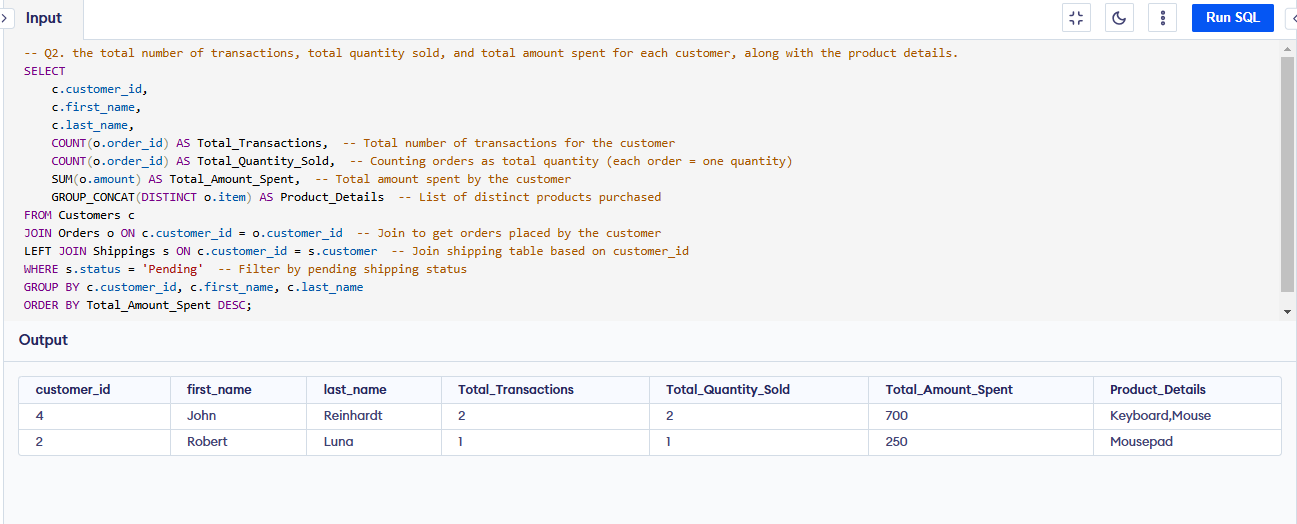


-- Q2. the total number of transactions, total quantity sold, and total amount spent for each customer, along with the product details.

Ans:

|  |
| --- |
| SELECT  c.customer\_id,  c.first\_name,  c.last\_name,  COUNT(o.order\_id) AS Total\_Transactions, -- Total number of transactions for the customer  COUNT(o.order\_id) AS Total\_Quantity\_Sold, -- Counting orders as total quantity (each order = one quantity)  SUM(o.amount) AS Total\_Amount\_Spent, -- Total amount spent by the customer  GROUP\_CONCAT(DISTINCT o.item) AS Product\_Details -- List of distinct products purchased  FROM Customers c  JOIN Orders o ON c.customer\_id = o.customer\_id -- Join to get orders placed by the customer  LEFT JOIN Shippings s ON c.customer\_id = s.customer -- Join shipping table based on customer\_id  WHERE s.status = 'Pending' -- Filter by pending shipping status  GROUP BY c.customer\_id, c.first\_name, c.last\_name  ORDER BY Total\_Amount\_Spent DESC; |

**OutPut:**



-- Q3. the maximum product purchased for each country.

Ans:

|  |
| --- |
| -- This Query will return only one product per country (even if multiple products have the same count  SELECT country, product AS most\_purchased\_product, total\_transactions  FROM (  SELECT  c.country,  o.item AS product,  COUNT(o.order\_id) AS total\_transactions,  ROW\_NUMBER() OVER (PARTITION BY c.country ORDER BY COUNT(o.order\_id) DESC) AS rank  FROM Orders o  JOIN Customers c ON o.customer\_id = c.customer\_id  GROUP BY c.country, o.item  ) ranked  WHERE rank = 1; |

**OutPut:**

A screenshot of a computer

AI-generated content may be incorrect.

|  |
| --- |
| --Keeps multiple products if they have the same max transaction count in a country.  SELECT country, product AS most\_purchased\_product, total\_transactions  FROM (  SELECT  c.country,  o.item AS product,  COUNT(o.order\_id) AS total\_transactions,  DENSE\_RANK() OVER (PARTITION BY c.country ORDER BY COUNT(o.order\_id) DESC) AS rank  FROM Orders o  JOIN Customers c ON o.customer\_id = c.customer\_id  GROUP BY c.country, o.item  ) ranked  WHERE rank = 1; |

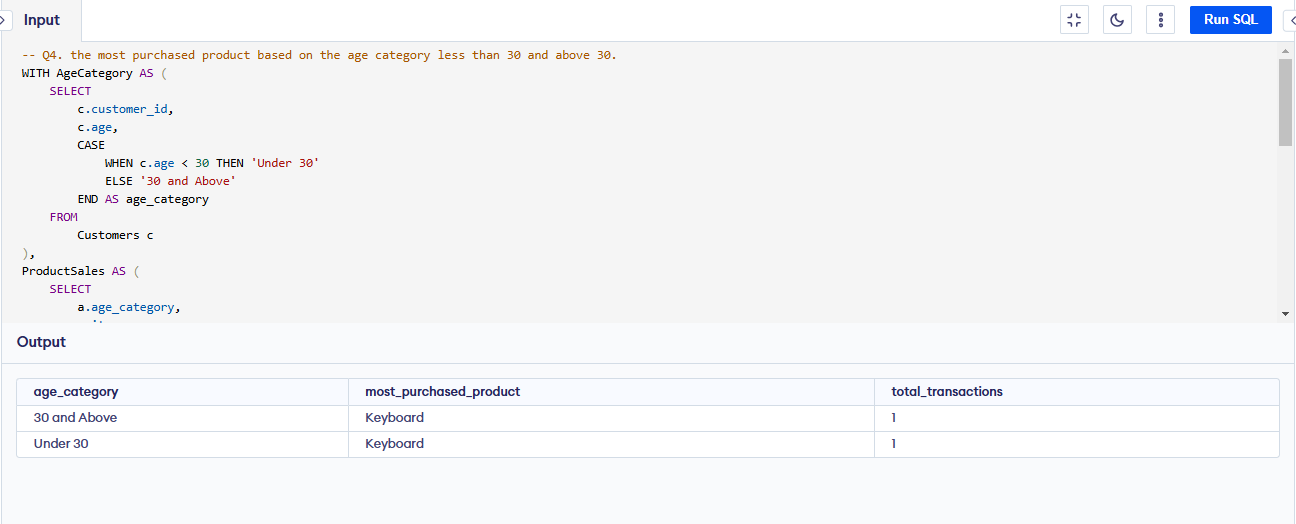
A screenshot of a computer

AI-generated content may be incorrect.

-- Q4. the most purchased product based on the age category less than 30 and above 30.

|  |
| --- |
| WITH AgeCategory AS ( -- categorizes customers into two groups based on their age: "Under 30" and "30 and Above"  SELECT  c.customer\_id, c.age,  CASE  WHEN c.age < 30 THEN 'Under 30'  ELSE '30 and Above'  END AS age\_category  FROM  Customers c  ),  -- calculate the total number of transactions (or total sales) for each product within each age category.  ProductSales AS (  SELECT  a.age\_category,  o.item,  COUNT(o.order\_id) AS total\_transactions  FROM  Orders o  JOIN  AgeCategory a ON o.customer\_id = a.customer\_id  GROUP BY  a.age\_category, o.item  ),  -- rank the products within each age category based on the total number of transactions in descending order  RankedProducts AS (  SELECT  age\_category, item AS product, total\_transactions,  ROW\_NUMBER() OVER (PARTITION BY age\_category ORDER BY total\_transactions DESC) AS rn  FROM  ProductSales  )  SELECT  age\_category,  product AS most\_purchased\_product,  total\_transactions  FROM  RankedProducts  WHERE  rn = 1; -- filter out the top-ranked product for each age category |

**OutPut:**



-- Q5. the country that had minimum transactions and sales amount.

|  |
| --- |
| WITH CountrySales AS (  SELECT  c.country, COUNT(o.order\_id) AS total\_transactions,  SUM(o.amount) AS total\_sales  FROM  Orders o  JOIN  Customers c ON o.customer\_id = c.customer\_id  GROUP BY  c.country  )  SELECT  country, total\_transactions, total\_sales  FROM  CountrySales  ORDER BY  total\_transactions ASC, total\_sales ASC  LIMIT 2; -- One can change the limit value ans see number of countries with minimum transaction |

**OutPut:**

