1. Display each employee’s salary and assign a rank based on how their salary compares to others within the same job category. Also, filter for employees hired after 2015, with a salary greater than 40,000, and living in Bashundhara or Bailey Road. Only include those with a job title of 'Teacher' or 'software Engineer'.

**Answer -**

SELECT p.PersonID, p.FirstName, p.LastName, j.JobTitle, p.Salary,

RANK() OVER (PARTITION BY p.JobID ORDER BY p.Salary DESC) AS SalaryRank

FROM persons p

JOIN jobs j ON p.JobID = j.JobID

WHERE p.Salary >= 40000

AND p.HireDate > '2015-01-01'

AND p.Location IN ('Bashundhara', 'Bailey Road')

AND j.JobTitle IN ('Teacher', 'software Engineer')

ORDER BY SalaryRank;

1. Find the total number of orders placed by each customer, considering the order amount per product (Quantity \* Price). Group the results by CustomerID and order the customers by total order value in descending order. Also, partition the result set by Country to show the order value for each customer within their respective country.

**Answer -**

SELECT c.CustomerID, c.CustomerName, SUM(od.Quantity \* p.Price) AS TotalOrderValue, c.Country

FROM customers c

JOIN orders o ON c.CustomerID = o.CustomerID

JOIN orderdetails od ON o.OrderID = od.OrderID

JOIN products p ON od.ProductID = p.ProductID

GROUP BY c.CustomerID, c.Country

ORDER BY TotalOrderValue DESC;

1. Retrieve all orders made in **2024** by customers who do not reside in **Bangladesh.** Display the **Customer ID, Name, Country, Gender, Order ID, Order Date, and the total number of products purchased per customer.** Sort the results by **product count in descending order.**

**Answer –**

SELECT c.CustomerID, c.CustomerName, c.Country, c.Gender,

o.OrderID, o.OrderDate, COUNT(od.ProductID) AS ProductCount

FROM orders o

JOIN customers c ON o.CustomerID = c.CustomerID

JOIN orderdetails od ON o.OrderID = od.OrderID

WHERE c.Country <> 'Bangladesh'

AND YEAR(o.OrderDate) = 2024

GROUP BY c.CustomerID, o.OrderID, o.OrderDate

ORDER BY ProductCount DESC;

1. Find the top 3 individuals from 'Bashundhara' or 'Bailey Road' who have ordered the highest quantity of items. Display their full name, location, and total quantity ordered. The results should be sorted by total quantity in descending order.

**Answer -**

SELECT p.PersonID, concat(p.FirstName, ' ', p.LastName) AS Full\_name, p.Location, SUM(od.Quantity) AS TotalQuantity

FROM persons p

JOIN orders o ON p. PersonID = o. PersonID

JOIN orderdetails od ON o.OrderID = od.OrderID

WHERE p.Location IN ('Bashundhara', 'Bailey Road')

GROUP BY p.PersonID, p.Location

ORDER BY TotalQuantity DESC

LIMIT 3;

1. Retrieve the **Order ID** with the **highest total order value**, calculated as the **sum of (Quantity × Price) for all products in that order**. However, only include orders that contain at least **two different products**. Sort the results in **descending order of total order value** and display only the top result.

**Answer -**

SELECT o.OrderID, SUM(od.Quantity \* p.Price) AS TotalOrderValue

FROM orders o

JOIN orderdetails od ON o.OrderID = od.OrderID

JOIN products p ON od.ProductID = p.ProductID

GROUP BY o.OrderID

HAVING COUNT(DISTINCT od.ProductID) >= 2

ORDER BY TotalOrderValue DESC

LIMIT 1;