**SQL TASK**

|  |
| --- |
| **Query 1: List of Users with Total Spending. Retrieve users and their total spending on orders.** |

|  |  |
| --- | --- |
| **SELECT**  **U.UserID,**  **U.Username,**  **SUM(O.TotalAmount) AS TotalSpending**  **FROM**  **Users U**  **INNER JOIN**  **Orders O ON U.UserID = O.UserID**  **GROUP BY**  **U.UserID, U.Username;** |  |

|  |
| --- |
| **Query 2: Products Never Ordered. Find products that have never been ordered.** |

|  |  |
| --- | --- |
| **SELECT**  **P.ProductID,**  **P.ProductName**  **FROM**  **Products P**  **LEFT JOIN**  **OrderDetails OD ON P.ProductID = OD.ProductID**  **WHERE**  **OD.OrderID IS NULL;** |  |

|  |
| --- |
| **Query 3: Top Selling Products. List the top 5 selling products by quantity.** |

|  |  |
| --- | --- |
| **SELECT**  **TOP 5**  **P.ProductID,**  **P.ProductName,**  **(SELECT SUM(Quantity) FROM OrderDetails WHERE ProductID = P.ProductID) AS TotalQuantitySold**  **FROM**  **Products AS P**  **ORDER BY**  **TotalQuantitySold DESC;** |  |

|  |
| --- |
| **Query 4: Users with No Orders. Retrieve a list of users who have never placed an order.** |

|  |  |
| --- | --- |
| **SELECT**  **UserID,**  **Username**  **FROM**  **Users**  **WHERE**  **UserID NOT IN (SELECT DISTINCT UserID FROM Orders);** |  |

|  |
| --- |
| **Query 5: Month-over-Month Growth in Sales. Calculate the month-over-month percentage growth in sales.** |

|  |  |
| --- | --- |
| **WITH sales\_per\_month AS (**  **SELECT**  **YEAR(OrderDate) AS OrderYear,**  **MONTH(OrderDate) AS OrderMonth,**  **SUM(TotalAmount) AS TotalSales,**  **LAG(SUM(TotalAmount)) OVER (ORDER BY YEAR(OrderDate), MONTH(OrderDate)) AS PreviousMonthSales**  **FROM**  **Orders**  **GROUP BY**  **YEAR(OrderDate),**  **MONTH(OrderDate)**  **)**    **SELECT**  **OrderYear,**  **OrderMonth,**  **TotalSales,**  **PreviousMonthSales,**  **(TotalSales - PreviousMonthSales) \* 100.0 / PreviousMonthSales AS MoM\_Growth**  **FROM**  **sales\_per\_month**  **ORDER BY**  **OrderYear, OrderMonth;** |  |

|  |
| --- |
| **Query 6: Most Popular Product Category. Identify the most popular product category based on quantity sold.** |

|  |  |
| --- | --- |
| **SELECT**  **TOP 1**  **PC.CategoryName,**  **SUM(OD.Quantity) AS TotalQuantitySold**  **FROM**  **ProductCategories PC**  **INNER JOIN**  **Products P ON PC.CategoryID = P.CategoryID**  **INNER JOIN**  **OrderDetails OD ON P.ProductID = OD.ProductID**  **GROUP BY**  **PC.CategoryName**  **ORDER BY**  **TotalQuantitySold DESC;** |  |

|  |
| --- |
| **Query 7: Users with Orders Above Average Value. Find users who have placed orders above the average order value.** |

|  |  |
| --- | --- |
| **SELECT**  **U.UserID,**  **U.Username**  **FROM**  **Users U**  **INNER JOIN**  **Orders O ON U.UserID = O.UserID**  **GROUP BY**  **U.UserID, U.Username**  **HAVING**  **AVG(O.TotalAmount) > (SELECT AVG(TotalAmount) FROM Orders);** |  |

|  |
| --- |
| **Query 8: Product Stock Status. List products along with a status indicating if their stock is Low, Medium, or High based on quantity.** |

|  |  |
| --- | --- |
| **SELECT**  **ProductID,**  **ProductName,**  **StockQuantity,**  **CASE**  **WHEN StockQuantity <= 10 THEN 'Low'**  **WHEN StockQuantity <= 50 THEN 'Medium'**  **ELSE 'High'**  **END AS StockStatus**  **FROM**  **Products;** |  |

|  |
| --- |
| **Query 9: Orders with Multiple Products. Retrieve orders that contain more than one type of product.** |

|  |  |
| --- | --- |
| **SELECT**  **OrderID**  **FROM**  **OrderDetails**  **GROUP BY**  **OrderID**  **HAVING**  **COUNT(DISTINCT ProductID) > 1;** |  |