**Reporting Project: Adventurework Company**

|  |  |
| --- | --- |
| Student Name | *Sayana Thavanut* |
| Course Name | SQL SERVER Reporting |
| Instructor | SOEUNG SENGHONG |
| Submission Date | *31/03/2025* |

# SQL QUERY REPORTING

## Report #1

SQL SERVER QUERY CODE

1. USE [AdventureWorks2016];

2.

3. SELECT TOP 10

4. p.Name AS [Product Name],

5. SUM(soh.LineTotal) AS [Total Sales Revenue],

6. SUM(soh.OrderQty) AS [Total Quantity Sold]

7. FROM Sales.SalesOrderDetail AS soh

8. INNER JOIN Production.Product AS p

9. ON soh.ProductID = p.ProductID

10. GROUP BY p.Name

11. ORDER BY [Total Sales Revenue] DESC;

12.

OUTPUT RESULT

**

## Report #2

SQL SERVER QUERY CODE

1. USE [AdventureWorks2016];

2.

3. WITH ProductRankByCategory AS (

4. SELECT

5. pc.Name AS [Product Category],

6. p.Name AS [Product Name],

7. SUM(sod.LineTotal) AS [Total Sales Revenue],

8. SUM(sod.OrderQty) AS [Total Quantity Sold],

9. ROW\_NUMBER() OVER (

PARTITION BY pc.Name ORDER BY SUM(sod.LineTotal) DESC) AS [Product Rank]

10. FROM Sales.SalesOrderDetail AS sod

11. INNER JOIN Production.Product AS p

12. ON sod.ProductID = P.ProductID

13. INNER JOIN Production.ProductSubcategory AS ps

14. ON p.ProductSubcategoryID = ps.ProductSubcategoryID

15. INNER JOIN Production.ProductCategory AS pc

16. ON ps.ProductCategoryID = pc.ProductCategoryID

17. GROUP BY pc.Name, p.Name

18. )

19. SELECT

20. [Product Category],

21. [Product Name],

22. [Total Sales Revenue],

23. [Total Quantity Sold]

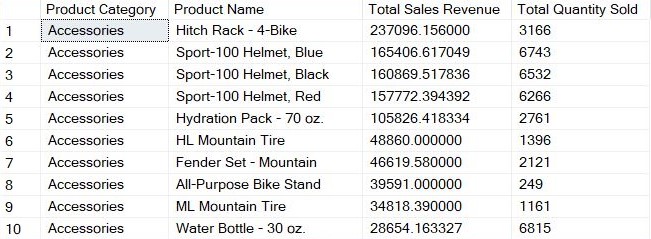
24. FROM ProductRankByCategory

25. WHERE [Product Rank] <= 10

26. ORDER BY [Product Category], [Total Sales Revenue] DESC;

27.

OUTPUT RESULT



## Report #3

SQL SERVER QUERY CODE

**1. USE [AdventureWorks2016];**

**2.**

**3. SELECT**

**4. YEAR(OrderDate) AS [Year],**

**5. MONTH(OrderDate) AS [Month],**

**6. SUM(TotalDue) AS [Total Sales Revenue]**

**7. FROM Sales.SalesOrderHeader**

**8. GROUP BY YEAR(OrderDate), MONTH(OrderDate)**

**9. ORDER BY [Year], [Month];**

**10.**

OUTPUT RESULT



## Report #4

SQL SERVER QUERY CODE

1. USE [AdventureWorks2016];

2.

3. SELECT

4. CONCAT(p.FirstName, ' ', p.LastName) AS [Customer Name],

5. SUM(soh.TotalDue) AS [Total Lifetime Spend]

6. FROM Sales.Customer AS c

7. LEFT JOIN Person.Person AS p

8. ON c.PersonID = p.BusinessEntityID

9. INNER JOIN Sales.SalesOrderHeader AS soh

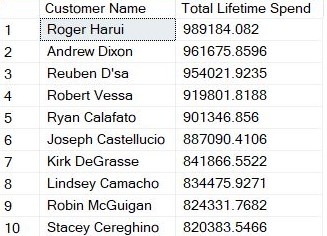
10. ON c.CustomerID = soh.CustomerID

11. GROUP BY c.CustomerID, p.FirstName, p.LastName

12. ORDER BY [Total Lifetime Spend] DESC;

13.

OUTPUT RESULT

**

## Report #5

SQL SERVER QUERY CODE

1. USE [AdventureWorks2016];

2.

3. SELECT

4. CONCAT(p.FirstName, ' ', p.LastName) AS [Customer Name],

5. COUNT(soh.SalesOrderID) AS [Number of Orders]

6. FROM Sales.Customer AS c

7. LEFT JOIN Person.Person AS p

8. ON c.PersonID = p.BusinessEntityID

9. INNER JOIN Sales.SalesOrderHeader AS soh

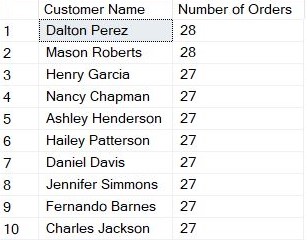
10. ON c.CustomerID = soh.CustomerID

11. GROUP BY c.CustomerID, p.FirstName, p.LastName

12. ORDER BY [Number of Orders] DESC;

13.

OUTPUT RESULT

**

## Report #6

SQL SERVER QUERY CODE

1. USE [AdventureWorks2016];

2.

3. SELECT

4. CustomerID AS [Customer ID],

5. MIN(OrderDate) [AS First Purchase Date],

6. MAX(OrderDate) AS [Most Recent Purchase Date],

7. COUNT(SalesOrderID) AS [Number of Orders]

8. FROM Sales.SalesOrderHeader

9. GROUP BY CustomerID

10. ORDER BY [Number of Orders] DESC;

11.

OUTPUT RESULT

**

## Report #7

SQL SERVER QUERY CODE

1. USE [AdventureWorks2016];

2.

3. SELECT

4. st.Name AS [Sales Region],

5. SUM(soh.TotalDue) AS [Total Sales Revenue]

6. FROM Sales.SalesOrderHeader AS soh

7. INNER JOIN Sales.SalesTerritory AS st

8. ON soh.TerritoryID = st.TerritoryID

9. GROUP BY st.TerritoryID, st.Name

10. ORDER BY [Total Sales Revenue] DESC;

11.

OUTPUT RESUL

**

## Report #8

SQL SERVER QUERY CODE

1. USE [AdventureWorks2016];

2.

3. SELECT

4. CONCAT(p.FirstName, ' ', p.LastName) AS [Employee Name],

5. e.HireDate AS [Hire Date],

6. DATEDIFF(YEAR, e.HireDate, GETDATE()) -

7. CASE

8. WHEN DATEADD(YEAR, DATEDIFF(YEAR, e.HireDate, GETDATE()), e.HireDate) >= GETDATE()

9. THEN 1

10. ELSE 0

11. END AS [Years of Service]

12. FROM HumanResources.Employee AS e

13. INNER JOIN Person.Person AS P

14. ON e.BusinessEntityID = p.BusinessEntityID

15. ORDER BY [Years of Service] DESC;

16.

OUTPUT RESULT

**

## Report #9

SQL SERVER QUERY CODE

1. USE [AdventureWorks2016];

2.

3. WITH MostValuableCustomers AS (

4. SELECT

5. CONCAT(p.FirstName, ' ', p.LastName) AS [Customer Name],

6. COUNT(soh.SalesOrderID) AS [Total Orders],

7. SUM(soh.TotalDue) AS [Total Amount Spent]

8. FROM Sales.Customer AS c

9. LEFT JOIN Person.Person AS p

10. ON c.PersonID = p.BusinessEntityID

11. INNER JOIN Sales.SalesOrderHeader AS soh

12. ON c.CustomerID = soh.CustomerID

13. GROUP BY c.CustomerID, p.FirstName, p.LastName

14. )

15. SELECT

16. [Customer Name],

17. [Total Orders],

18. [Total Amount Spent]

19. FROM MostValuableCustomers

20. ORDER BY [Total Amount Spent] DESC;

21.

OUTPUT RESULT

**

## Report #10

SQL SERVER QUERY CODE

1. USE [AdventureWorks2016];

2.

3. WITH SalesPerformance AS (

4. SELECT

5. CONCAT(p.FirstName, ' ', p.LastName) AS [Employee Name],

6. sp.SalesYTD AS [Total Sales]

7. FROM Sales.SalesPerson AS sp

8. INNER JOIN Person.Person AS P

9. ON sp.BusinessEntityID = P.BusinessEntityID

10. )

11. SELECT

12. [Employee Name],

13. [Total Sales],

14. RANK() OVER (ORDER BY [Total Sales] DESC) AS [Rank]

15. FROM SalesPerformance;

16.

OUTPUT RESULT

**

## Report #11

SQL SERVER QUERY CODE

1. USE [AdventureWorks2016];

2.

3. WITH MonthlySales AS (

4. SELECT

5. DATEPART(YEAR, OrderDate) AS [Year],

6. DATEPART(MONTH, OrderDate) AS [MonthNumber],

7. SUM(TotalDue) AS [Sales]

8. FROM Sales.SalesOrderHeader

9. GROUP BY

10. DATEPART(YEAR, OrderDate),

11. DATEPART(MONTH, OrderDate)

12. ),

13. SalesGrowth AS (

14. SELECT

15. [Year],

16. [MonthNumber],

17. [Sales],

18. LAG([Sales]) OVER (ORDER BY [Year], [MonthNumber]) AS [PreviousSales]

19. FROM MonthlySales

20. )

21. SELECT

22. FORMAT(DATEFROMPARTS([Year], [MonthNumber], 1), 'MMMM yyyy') AS [Month],

23. [Sales],

24. ROUND(

25. CASE

26. WHEN [PreviousSales] IS NULL THEN 0

27. ELSE (([Sales] - [PreviousSales]) / [PreviousSales]) \* 100

28. END, 2) AS [Growth %]

29. FROM SalesGrowth

30. ORDER BY [Year] ASC, [MonthNumber] ASC;

31.

OUTPUT RESULT



## Report #12

SQL SERVER QUERY CODE

1. USE [AdventureWorks2016];

2.

3. WITH LastOrders AS (

4. SELECT

5. CONCAT(p.FirstName, ' ', p.LastName) AS [Customer Name],

6. MAX(OrderDate) AS [Last Order Date]

7. FROM [Sales].[Customer] AS c

8. LEFT JOIN Person.Person AS p

9. ON c.PersonID = p.BusinessEntityID

10. INNER JOIN Sales.SalesOrderHeader AS soh

11. ON c.CustomerID = soh.CustomerID

12. GROUP BY c.CustomerID, p.FirstName, p.LastName

13. )

14. SELECT

15. [Customer Name],

16. l.[Last Order Date]

17. FROM LastOrders AS l

18. WHERE l.[Last Order Date] < (

19. SELECT DATEADD(YEAR, -1, MAX(OrderDate))

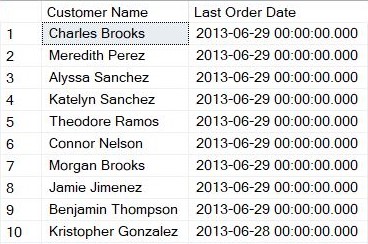
20. FROM Sales.SalesOrderHeader

21. )

22. ORDER BY l.[Last Order Date] DESC;

23.

OUTPUT RESULT



## Report #13

SQL SERVER QUERY CODE

1. USE AdventureWorks2016;

2.

3. WITH HighValueCustomers AS (

4. SELECT

5. CONCAT(p.FirstName, ' ', p.LastName) AS [Customer Name],

6. DATEDIFF(DAY, MAX(soh.OrderDate),

(SELECT MAX(OrderDate)

FROM Sales.SalesOrderHeader)) AS [Recency (Days since last order)],

7. COUNT(soh.SalesOrderID) AS [Frequency (Total Orders)],

8. SUM(soh.TotalDue) AS [Monetary (Total Spend)]

9. FROM Sales.Customer AS c

10. LEFT JOIN Person.Person AS p

11. ON c.PersonID = p.BusinessEntityID

12. INNER JOIN Sales.SalesOrderHeader AS soh

13. ON c.CustomerID = soh.CustomerID

14. GROUP BY c.CustomerID, p.FirstName, p.LastName

15. )

16. SELECT

17. [Customer Name],

18. [Recency (Days since last order)],

19. [Frequency (Total Orders)],

20. [Monetary (Total Spend)]

21. FROM HighValueCustomers

22. ORDER BY

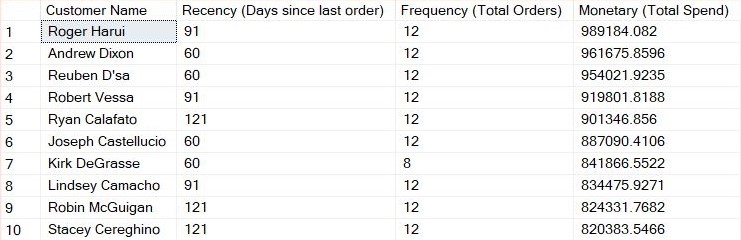
23. [Monetary (Total Spend)] DESC,

24. [Frequency (Total Orders)] DESC,

25. [Recency (Days since last order)] ASC;

26.

OUTPUT RESULT

**

## Report #14

SQL SERVER QUERY CODE

1. USE [AdventureWorks2016];

2.

3. WITH RevenueByMonth AS (

4. SELECT

5. DATEPART(YEAR, OrderDate) AS [Year],

6. DATEPART(MONTH, OrderDate) AS [Month],

7. SUM(TotalDue) AS [Total Revenue]

8. FROM Sales.SalesOrderHeader

9. GROUP BY

10. DATEPART(YEAR, OrderDate), DATEPART(MONTH, OrderDate)

12. ),

13. FindNewCustomer AS (

14. SELECT

15. DATEPART(YEAR, FirstOrderDate) AS [Year],

16. DATEPART(MONTH, FirstOrderDate) AS [Month],

17. COUNT(CustomerID) AS [NumNewCustomers]

18. FROM (

19. SELECT

20. CustomerID,

21. MIN(OrderDate) AS [FirstOrderDate]

22. FROM Sales.SalesOrderHeader

23. GROUP BY CustomerID

24. ) FirstOrder

25. GROUP BY DATEPART(YEAR, FirstOrderDate), DATEPART(MONTH, FirstOrderDate)

26. ),

27. CombineBoth AS (

28. SELECT

29. rm.[Year],

30. rm.[Month],

31. rm.[Total Revenue],

32. COALESCE(fnc.[NumNewCustomers], 0) AS [NewCustomers],

33. LAG(rm.[Total Revenue]) OVER (ORDER BY rm.[Year], rm.[Month]) AS [PreviousRevenue],

34. LAG(fnc.[NumNewCustomers]) OVER (ORDER BY rm.[Year], rm.[Month]) AS [PreviousCustomers]

35. FROM RevenueByMonth AS rm

36. LEFT JOIN FindNewCustomer AS fnc

37. ON rm.[Year] = fnc.[Year] AND rm.[Month] = fnc.[Month]

38. )

39. SELECT

40. FORMAT(DATEFROMPARTS(Year, Month, 1), 'MMMM yyyy') AS [Month/Year],

41. [Total Revenue],

42. [NewCustomers] AS [New Customers Acquired],

43. ROUND(

44. CASE

45. WHEN [PreviousRevenue] IS NULL THEN 0

46. ELSE (([Total Revenue] - [PreviousRevenue]) / [PreviousRevenue]) \* 100

47. END, 2) AS [Percentage Change in Revenue],

48. ROUND(

49. CASE

50. WHEN [PreviousCustomers] IS NULL OR [PreviousCustomers] = 0 THEN 0

51. ELSE (([NewCustomers] - [PreviousCustomers]) / CAST([PreviousCustomers] AS FLOAT)) \* 100

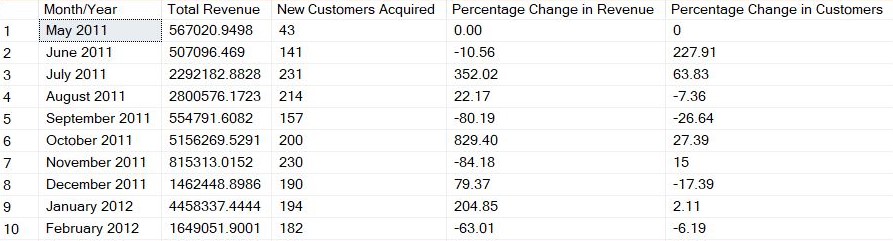
52. END, 2) AS [Percentage Change in Customers]

53. FROM CombineBoth

54. ORDER BY [Year], [Month];

55.

OUTPUT RESULT



## Report #15

SQL SERVER QUERY CODE

1. WITH ProductRevenueByTerritory AS (

2. SELECT

3. st.Name AS [Sales Territory Region Name],

4. p.Name AS [Product Name],

5. SUM(sod.LineTotal) AS [Revenue]

6. FROM Sales.SalesOrderDetail AS sod

7. INNER JOIN Sales.SalesOrderHeader AS soh

8. ON sod.SalesOrderID = soh.SalesOrderID

9. INNER JOIN Sales.SalesTerritory AS st

10. ON soh.TerritoryID = st.TerritoryID

11. INNER JOIN Production.Product AS p

12. ON sod.ProductID = p.ProductID

13. GROUP BY

14. st.Name, p.Name

15. ),

16. RankProducts AS (

17. SELECT

18. [Sales Territory Region Name],

19. [Product Name],

20. [Revenue],

21. ROW\_NUMBER() OVER (

22. PARTITION BY [Sales Territory Region Name]

23. ORDER BY [Revenue] DESC

24. ) AS [Rank (Top 3) within Territory]

25. FROM ProductRevenueByTerritory

26. )

27. SELECT

28. [Sales Territory Region Name],

29. [Product Name],

30. [Revenue],

31. [Rank (Top 3) within Territory]

32. FROM RankProducts

33. WHERE [Rank (Top 3) within Territory] <= 3

34. ORDER BY

35. [Sales Territory Region Name], [Rank (Top 3) within Territory];

36.

OUTPUT RESULT

