**📊 Advanced Sales Report with ROLLUP**

Final Internship Project by Abhay Kumar | July 2025  
Department of Computer Science & Engineering, Chandigarh University  
Internship at Celebal Technologies (June–August 2025)

**✅ 1. Executive Summary**

This report presents the design and implementation of an advanced sales reporting system built on Microsoft SQL Server.  
The project leverages GROUP BY ROLLUP, Common Table Expressions (CTEs), and window functions to produce detailed sales data, category-level subtotals, overall totals, and key analytics (average, minimum, and maximum sales).  
Dynamic filters by date and category make the report flexible, enabling managers to explore historical data over any period or for specific product lines.

The goal is to reduce manual reporting effort and improve data-driven decision making for the business.

**📌 2. Project Objectives**

* Build a dynamic sales report that includes:

Product-level sales data

Subtotals by product category

Grand total for the entire dataset

Add business analytics:

Average sale per category

Minimum and maximum sale amounts per category

Ranking of product categories by total sales

* Use SQL Server tools to ensure the report is:

Easy to maintain and extend

Performant for larger datasets

* Enable parameterized filtering so managers can view:

Sales in a specific date range

Sales for selected product categories

**⚙️ 3. Methodology & Tools**

| Step | Description |
| --- | --- |
| Database | Microsoft SQL Server 2019 |
| Data Modeling | Created a normalized table Sales with columns: ProductCategory, ProductName, SaleAmount, and SaleDate. |
| ETL (Sample Data) | Inserted realistic sample data covering multiple categories and dates |
| Reporting Logic | Used GROUP BY ROLLUP for hierarchical aggregation |
| Analytics | Used window functions (RANK()) and aggregation functions (AVG, MIN, MAX) |
| Flexibility | Added parameters for dynamic date and category filters |
| Testing & Validation | Executed sample queries, compared results with manual calculations |

**🧩 4. Detailed Design**

4.1 Table Design

| Column | Data Type | Description |
| --- | --- | --- |
| Id | INT (Identity) | Primary key |
| ProductCategory | VARCHAR(50) | Product category (e.g., Electronics) |
| ProductName | VARCHAR(50) | Product name (e.g., Laptop) |
| SaleAmount | DECIMAL(10,2) | Sale value |
| SaleDate | DATE | Date of sale |

**4.2 Query Logic**

CTE (Common Table Expression): Filters data by date and optional category.

ROLLUP: Produces:

Product-level totals

Category-level subtotals

Grand total

Analytics section:

AVG(SaleAmount), MIN(SaleAmount), MAX(SaleAmount) by category

Ranking: Uses RANK() to order categories by total sales.

**📈 5. Sample Data**

| ProductCategory | ProductName | SaleAmount | SaleDate |
| --- | --- | --- | --- |
| Electronics | Laptop | 1000.00 | 2025-07-01 |
| Electronics | Phone | 800.00 | 2025-07-02 |
| Electronics | Tablet | 500.00 | 2025-07-03 |
| Clothing | Shirt | 300.00 | 2025-07-01 |
| Clothing | Pants | 400.00 | 2025-07-02 |
| Furniture | Sofa | 1200.00 | 2025-07-03 |
| Furniture | Bed | 900.00 | 2025-07-04 |

**🛠 6. Key SQL Features Used**

| Feature | Purpose |
| --- | --- |
| GROUP BY ROLLUP | Creates subtotal and grand total rows automatically |
| ISNULL & CASE | Format subtotal and total rows clearly |
| CTE | Keeps query organized and improves readability |
| Window functions (RANK()) | Rank product categories based on total sales |
| Parameters | Make report dynamic (date range & category filter) |

**🔍 7. Results & Output**

7.1 Sales Report Output

| ProductCategory | ProductName | TotalSales |
| --- | --- | --- |
| Electronics | Laptop | 1000.00 |
| Electronics | Phone | 800.00 |
| Electronics | Tablet | 500.00 |
| Electronics | Total | 2300.00 |
| Clothing | Pants | 400.00 |
| Clothing | Shirt | 300.00 |
| Clothing | Total | 700.00 |
| Furniture | Bed | 900.00 |
| Furniture | Sofa | 1200.00 |
| Furniture | Total | 2100.00 |
| Total | Total | 5100.00 |

**7.2 Analytics Output**

| ProductCategory | NumberOfProducts | TotalSales | AverageSale | MinSale | MaxSale |
| --- | --- | --- | --- | --- | --- |
| Electronics | 3 | 2300.00 | 766.67 | 500.00 | 1000.00 |
| Furniture | 2 | 2100.00 | 1050.00 | 900.00 | 1200.00 |
| Clothing | 2 | 700.00 | 350.00 | 300.00 | 400.00 |

**7.3 Ranking Output**

| ProductCategory | TotalSales | CategoryRank |
| --- | --- | --- |
| Electronics | 2300.00 | 1 |
| Furniture | 2100.00 | 2 |
| Clothing | 700.00 | 3 |

**✅ 8. Business Impact**

Time savings: Automatic subtotals reduce manual Excel work.

Accuracy: Centralized database ensures consistent results.

Insights: Managers can see best/worst-performing categories and trends.

Flexibility: Filters let users analyze specific months or product groups.

**🧭 9. Future Enhancements**

Add month-over-month and year-over-year growth calculations.

Build a web dashboard (using Power BI / React) for interactive reports.

Extend to multi-currency and region-wise sales.

Automate daily/weekly email reports to stakeholders.

👨‍💻 Appendix: Key SQL Script Highlights

-- ===========================================================================================

-- Project: Advanced Sales Report with ROLLUP and Analytics

-- Author: Abhay Kumar

-- Final Internship Project | July 2025

-- ===========================================================================================

-- Drop table if it already exists

IF OBJECT\_ID('dbo.Sales') IS NOT NULL DROP TABLE dbo.Sales;

-- Create Sales table

CREATE TABLE dbo.Sales (

Id INT IDENTITY(1,1) PRIMARY KEY,

ProductCategory VARCHAR(50),

ProductName VARCHAR(50),

SaleAmount DECIMAL(10,2),

SaleDate DATE

);

-- Insert sample data

INSERT INTO dbo.Sales (ProductCategory, ProductName, SaleAmount, SaleDate) VALUES

('Electronics', 'Laptop', 1000, '2025-07-01'),

('Electronics', 'Phone', 800, '2025-07-02'),

('Electronics', 'Tablet', 500, '2025-07-03'),

('Clothing', 'Shirt', 300, '2025-07-01'),

('Clothing', 'Pants', 400, '2025-07-02'),

('Furniture', 'Sofa', 1200, '2025-07-03'),

('Furniture', 'Bed', 900, '2025-07-04');

-- Declare parameters for dynamic filtering

DECLARE @StartDate DATE = '2025-07-01';

DECLARE @EndDate DATE = '2025-07-31';

DECLARE @CategoryFilter VARCHAR(50) = NULL; -- Example: set to 'Electronics' if you want to filter

-- Option 1: Use CTE followed immediately by SELECT

WITH FilteredSales AS (

SELECT \*

FROM dbo.Sales

WHERE SaleDate BETWEEN @StartDate AND @EndDate

AND (@CategoryFilter IS NULL OR ProductCategory = @CategoryFilter)

)

-- 1️⃣ Main sales report: details + subtotals + grand total

SELECT

ISNULL(ProductCategory, 'Total') AS ProductCategory,

CASE

WHEN ProductCategory IS NULL THEN 'Total'

WHEN ProductName IS NULL THEN 'Total'

ELSE ProductName

END AS ProductName,

SUM(SaleAmount) AS TotalSales

FROM FilteredSales

GROUP BY ROLLUP(ProductCategory, ProductName)

ORDER BY

GROUPING(ProductCategory),

ProductCategory,

GROUPING(ProductName),

ProductName;

-- Option 2: reuse data → insert into temp table

SELECT \*

INTO #FilteredSales

FROM dbo.Sales

WHERE SaleDate BETWEEN @StartDate AND @EndDate

AND (@CategoryFilter IS NULL OR ProductCategory = @CategoryFilter);

-- 2️⃣ Analytics per category

SELECT

ProductCategory,

COUNT(\*) AS NumberOfProducts,

SUM(SaleAmount) AS TotalSales,

AVG(SaleAmount) AS AverageSale,

MIN(SaleAmount) AS MinSale,

MAX(SaleAmount) AS MaxSale

FROM #FilteredSales

GROUP BY ProductCategory

ORDER BY TotalSales DESC;

-- 3️⃣ Ranking of categories by total sales

;WITH CategoryTotals AS (

SELECT ProductCategory, SUM(SaleAmount) AS TotalSales

FROM #FilteredSales

GROUP BY ProductCategory

)

SELECT

ProductCategory,

TotalSales,

RANK() OVER (ORDER BY TotalSales DESC) AS CategoryRank

FROM CategoryTotals;

-- Drop temp table at the end

DROP TABLE #FilteredSales;

-- ===========================================================================================

-- End of script

-- ===========================================================================================

**🙌 Conclusion**

This project shows how SQL Server’s advanced grouping and analytics can transform raw sales data into actionable insights.  
By combining ROLLUP, window functions, and flexible filters, the solution delivers both detail and summary views that help managers make data-driven decisions faster.

Final Internship Project by Abhay Kumar | Celebal Technologies | July 2025