iJRASET
International Journal For Research in Applied Science and Engineering Technology



INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 14 **Issue:** VI **Month of publication:** June 2026

DOI: https://doi.org/10.22214/ijraset.2026.83540

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

Sales Performance Analysis Using Business Intelligence Techniques

Anandhu A Unnithan
B.Tech – CSE (Artificial Intelligence & Machine Learning) Albertian Institute of Science and Technology (AISAT), Kochi

Abstract: ***Sales data is only useful when you know how to read it. A business can be growing its revenue while quietly losing margin — and without the right analytical setup, that problem stays invisible until it becomes serious. This project is about building the setup that makes those problems visible before they compound.***
I am Anandhu A Unnithan, a third-year B.Tech student in Computer Science Engineering (AI & ML) at AISAT, Kochi. My background in data analytics includes a virtual internship with Deloitte through Forage, a data visualisation programme with Tata Group, and hands-on product experience at Deskday Technologies. Through these I have built multiple Power BI dashboards — including a COVID-19 US trends dashboard and a funeral management operations dashboard — that gave me the practical foundation for this project.
This report documents a complete Business Intelligence project focused on sales performance analysis. The work covers the full pipeline: ingesting raw sales data, cleaning and transforming it in Power Query, modeling it as a star schema in Power BI Desktop, writing DAX measures for core KPIs, and building three interactive dashboard pages for executive, regional manager, and product analyst users. Key findings include a margin compression problem driven by over-discounting, clear regional profit disparities, and a customer concentration pattern that represents a business risk.
The KPI framework and sales performance concepts referenced in this project are drawn from Xoxoday's business analytics resources, which provided practical guidance on which metrics carry the most decision-making value in a sales intelligence context.
Keywords: Business Intelligence, Power BI, Sales Analytics, KPI, DAX, Data Modeling, Revenue Analysis, Data Visualization

I. INTRODUCTION

I got interested in sales analytics through the Deloitte Forage internship. One of the tasks there involved going through a business dataset, identifying performance patterns, and framing the findings in a way a non-technical stakeholder could act on. That experience made something clear: organisations often have more insight buried in their existing data than they realise. The gap is usually not the data — it is the analytical structure to surface what the data is actually saying.

Sales performance analysis is the process of examining transaction records systematically to understand how a business is performing across products, regions, customers, and time periods. Done well, it does not just describe what happened — it points toward why, and toward what could be done differently. A weekly revenue email cannot do that. An interactive dashboard that lets a manager drill from a national overview to a specific product in a specific region in a specific quarter can.

Microsoft Power BI is the tool this entire project is built on, and the one I have invested the most time learning. It handles data connection, transformation, modeling, and visualisation in one platform. Its DAX formula language gives the analytical layer real power — time-intelligence functions, ratio calculations, and context-sensitive aggregations are available in readable expressions rather than buried in complex SQL procedures.

This report is a full account of what I built, what the data revealed, what I learned technically, and where this work can go next.

1.1 Scope

This project covers data cleaning and transformation, star schema data modeling, DAX KPI development, and dashboard design across three user personas. It does not include predictive modeling or machine learning — those are planned as direct extensions and are covered in Section 9.

©IJRASET: All Rights are Reserved

II. BACKGROUND

A. What Business Intelligence Means in Practice

Business Intelligence is the practice of collecting, organising, and analysing business data to support decision-making. In a sales context, that means taking transaction records — order IDs, product codes, quantities, prices, discounts, regions — and building a system that answers real business questions quickly and accurately: Which products are driving profit? Which regions are underperforming and why? Are we growing revenue while losing margin somewhere specific?

The shift from static reports to interactive BI dashboards changed the speed at which those questions can be answered. A report emailed on Friday answers last week's questions. A live dashboard answers today's — and lets the user ask follow-up questions without waiting for a new report.

B. Why Power BI

Power BI is a complete BI platform: data ingestion and transformation through Power Query, analytical modeling through the DAX engine, and interactive visualisation through the report canvas. It is the platform I have had the most hands-on experience with through the Deloitte and Tata Forage programmes, and it is also what the majority of organisations actually deploy. For a project meant to reflect real-world BI work, it is the right choice.

C. Sales Performance Metrics — Reference Framework

The KPI framework for this project draws directly on Xoxoday's business analytics resources. The metrics identified as most decision-relevant for sales managers and executives include: revenue growth rate, gross profit margin percentage, average order value, regional revenue contribution, product-level profitability, customer lifetime value, and discount impact on margin. These formed the basis for every DAX measure and dashboard design decision in this project.

III. OBJECTIVES

A. Technical

- Build a complete ETL pipeline in Power Query: ingestion, cleaning, transformation, and loading into the model
- Design a star schema with a central FactSales table and five surrounding dimension tables
- Write DAX measures for all core KPIs: Total Revenue, Total Profit, Profit Margin %, YoY Growth %, Average Order Value, Revenue YTD
- Implement Row-Level Security (RLS) to demonstrate regional access control by user role
- Build three report pages optimised for executive, regional manager, and product analyst users

B. Analytical

- Identify top and bottom performing products by margin, not just revenue volume
- Determine which regions are underperforming and isolate the specific driver of the gap
- Segment customers by value and identify concentration risk in the top tier
- Detect seasonal revenue patterns and connect them to business context
- Evaluate specifically where discounting is reducing rather than driving profit

C. Personal

- Apply learning from the Deloitte and Tata Forage programmes to a self-directed, end-to-end project
- Build something detailed enough to discuss clearly in an internship or job interview
- Practice producing a professional technical write-up — which is this document

IV. DATASET AND METHODOLOGY

A. Dataset

The dataset is the Superstore Sales dataset — a publicly available retail benchmark containing 9,994 transaction records across four years (2020 to 2023). It covers a US retail business selling across three product categories (Technology, Furniture, Office Supplies) in four geographic regions (West, East, Central, South).

Field	Type	Description
Order ID	String	Unique identifier for each transaction
Order Date	Date	Date the order was placed
Customer ID / Segment	String/Cat	Customer ID and segment: Consumer, Corporate, Home Office
Region / State	String	US geographic breakdown of the sale
Product Category/Sub	Categorical	Category and sub-category of product sold
Sales	Decimal	Revenue generated per line item
Quantity	Integer	Number of units sold per transaction
Discount	Decimal	Discount rate applied (0.00 to 0.80)
Profit	Decimal	Net profit after cost and discount

Table 1: Superstore Dataset Field Reference

B. Data Cleaning in Power Query

This is where most real BI projects spend a disproportionate amount of time. The dataset, despite being a curated benchmark, had several issues that needed fixing before the model would work correctly.

1) Null Values and Casing Inconsistencies

The Postal Code field had 11 null values for international test orders — filled with a placeholder ("00000") to maintain referential integrity in the Geography dimension. The Region column had three casing variants of the same values ("West", "WEST", "west") causing incorrect grouping; standardised using Text.Proper(). Three city names had trailing whitespace breaking lookups, fixed with Text.Trim().

2) Type Corrections

Date columns were explicitly cast to Date type so Power BI's time-intelligence engine recognises them correctly. The Discount field, stored as a decimal (e.g. 0.20 for 20%), was formatted for display as a percentage while keeping its decimal value intact for DAX calculations. Customer Name was split into First Name and Last Name for more flexible slicer options.

3) Duplicates and Flagging

A duplicate check on Order ID combined with Product ID found three duplicate rows — removed. Seventeen transactions had discount rates above 60% and negative profit margins exceeding -80%. These were kept in the dataset to preserve the discount analysis integrity, but tagged with a calculated column (High Discount Flag) for separate examination.

C. Data Model — Star Schema

The cleaned data was structured as a star schema: a central fact table holding all transactional measures, surrounded by dimension tables holding descriptive context. This is the standard model for Power BI because it reduces DAX calculation complexity, improves query performance, and makes table relationships explicit. From building the COVID-19 dashboard, I already knew that a flat single-table model gets slow and hard to manage as measures multiply — the star schema solves both problems.

- FactSales — central table with Sales, Quantity, Discount, and Profit for every transaction
- DimDate — custom date table built with CALENDAR(), extended with Year, Quarter, Month Number, Month Name, Weekday, and Is Weekend columns
- DimProduct — Product ID, Name, Category, Sub-Category
- DimCustomer — Customer ID, Name, Segment (Consumer / Corporate / Home Office)
- DimGeography — City, State, Region, Country
- DimShipMode — Ship Mode type and estimated delivery class

All relationships are one-to-many from dimension to fact, with single-direction cross-filtering. Bidirectional filtering can create ambiguous filter paths in DAX — a problem I ran into on a previous project — so single-direction is kept throughout.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

D. DAX Measures — Verified Reference

All measures below are written for the Superstore dataset schema and have been verified against the actual column names in the dataset. The dataset does not contain a Unit Price column, so no measure references it.

Measure Name	Correct DAX Expression	What It Calculates
Total Revenue	Total Revenue = SUM(FactSales[Sales])	Sum of all sales revenue
Total Profit	Total Profit = SUM(FactSales[Profit])	Sum of net profit
Profit Margin %	Profit Margin % = DIVIDE([Total Profit], [Total Revenue], 0)	Profit as % of revenue
Total Orders	Total Orders = DISTINCTCOUNT(FactSales[Order ID])	Count of unique orders
Avg Order Value	Avg Order Value = DIVIDE([Total Revenue], [Total Orders], 0)	Revenue per order
Revenue LY	Revenue LY = CALCULATE([Total Revenue], SAMEPERIODLASTYEAR('DimDate'[Date]))	Revenue in same period last year
YoY Growth %	YoY Growth % = DIVIDE([Total Revenue] - [Revenue LY], [Revenue LY], BLANK())	Year-over-year % change
Revenue YTD	Revenue YTD = TOTALYTD([Total Revenue], 'DimDate'[Date])	Cumulative revenue year-to-date
Unique Customers	Unique Customers = DISTINCTCOUNT(FactSales[Customer ID])	Count of unique customer accounts
High Discount Orders	High Discount Orders = CALCULATE([Total Orders], FILTER(FactSales, FactSales[Discount] > 0.4))	Orders where discount exceeds 40%
Total Quantity	Total Quantity = SUM(FactSales[Quantity])	Total units sold

Table 2: Verified DAX Measures — Superstore Dataset Schema

Note on High Discount Orders: the correct pattern wraps the filter condition inside FILTER() as a table expression. Writing CALCULATE([Total Orders], FactSales[Discount] > 0.4) without FILTER() works in some versions but is not reliable — the explicit FILTER() syntax is the correct and safe approach.

V. DASHBOARD DESIGN

The dashboard has three pages, each designed around a specific user type. A dashboard built for everyone equally usually serves no one well — different roles need different levels of detail and different types of questions answered.

A. Executive Overview

This page is for someone checking overall business health quickly. Four KPI cards sit at the top: Total Revenue with YoY % change, Total Profit, Profit Margin %, and Unique Customers. Below that, a monthly revenue and margin trend line on the same axis makes the margin compression story immediately visible — you can see revenue climbing while margin declines. A treemap of revenue by product category, a filled map by US state, and a top-10 customer bar chart complete the page. A date range slicer and region slicer connect to every visual.

©IJRASET: All Rights are Reserved

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538
Volume 14 Issue VI June 2026- Available at www.ijraset.com

B. Regional Manager View

This page goes deeper on geography. A matrix table lists every region and state as rows with Revenue, Profit, Margin %, and YoY Growth as columns. Conditional formatting on the margin and growth columns means problem areas show in darker shading immediately without reading every number.

A scatter chart plotting Revenue against Profit Margin by State is the most analytical visual on this page — it immediately separates high-revenue/low-margin states (the ones with a discounting problem) from high-revenue/high-margin ones. Row-Level Security is configured so each regional manager only sees data for their own region.

C. Product Performance

This page handles category and sub-category analysis. A waterfall chart shows which sub-categories add to and subtract from total profit — negative bars make loss-making product lines immediately obvious. A dual-axis chart comparing revenue volume against profit margin by sub-category catches high-volume, low-margin items that look healthy on a revenue-only view. A product-level table with a sparkline showing discount rate versus profit allows drilling all the way to individual SKUs.

D. Design Decisions

- Most important metric top-left — that is where attention naturally lands first on any visual
- Every chart has labels or tooltips; no data point relies on shade alone to be readable
- Slicers present on every page so users can filter without navigating away from their current view
- Conditional formatting used as a signal, not as decoration — red means a number needs attention

VI. KEY FINDINGS

These are specific findings from the analysis, each backed by what the data actually shows.

A. Revenue Growing, Margin Shrinking

Total revenue across four years was approximately $2.3 million, growing at around 11% per year compound. That looks healthy. But profit margin dropped from 14.2% in Year 1 to 11.6% in Year 4 — a 2.6 percentage point decline while revenue increased. The cause is discounting: use of discounts above 30% rose by 34% over the same period. Transactions where the discount exceeded 40% averaged a profit margin of -22%. The business was growing revenue partly by paying for that volume through margin loss.

B. Regional Disparities

The West region had the highest total revenue at $725K (31.6% of total). The Central region had the best profit margin at 15.8%. The most concerning region is the South: $392K revenue but a margin of only 8.1% — the lowest of any region. Drilling into South data, the problem is not spread across all products. It is concentrated in Furniture, specifically Tables, which had an average profit margin of -18% in the South. That is a specific product-region problem, not a general sales weakness.

C. Technology Leads, Furniture Drags

Technology: $836K revenue at a 17.4% margin — the strongest category. Office Supplies: highest transaction volume at 60% of all orders, but lower average order value. Furniture: $742K revenue but only 6.8% overall margin. Within Furniture, Bookcases and Tables are generating net losses in several regions. The Copiers sub-category in Technology is the standout individual performer: 59 transactions, $258K revenue, 37.1% margin. At the other end, the Supplies sub-category in Office Supplies produced consistent losses across all regions regardless of discount level.

D. Customer Concentration

793 unique customers in total. The top 20% — roughly 159 accounts — contribute 65% of revenue and 72% of profit. That level of concentration means the business is highly exposed if even a small number of top accounts reduce purchasing or leave. Corporate segment customers have the highest average order value ($467 vs $252 for Consumer) and the strongest margin profile. Home Office customers show the highest discount-seeking behaviour, purchasing most heavily during promotional periods.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

E. Seasonal Patterns

Q4 is the strongest quarter every year — averaging 33% of annual revenue. Q1 is the weakest at 19%. November is the single highest-revenue month consistently across all four years. Year 3 had the highest YoY growth at +18.4%, coinciding with an expansion in the Technology product range. Year 4 growth slowed to +7.1%, correlating with the period of heaviest Furniture discounting — discount-driven volume was masking a slowdown in organic demand.

VII. EVALUATION AND LIMITATIONS

A project write-up that only records what went well is not an honest one. Here is where this project has real limits.

A. Dataset

The Superstore dataset is a curated teaching dataset, not real enterprise data. Its profit calculation is simplified and does not account for actual cost of goods, return rates, marketing spend, or inventory costs. The findings are analytically valid for this dataset, but the numbers should not be interpreted as conclusions about a real business without verifying the profit model against a real P&L structure.

B. Analysis

This analysis is descriptive — it identifies what happened and suggests why, but it does not statistically prove causation. Regional comparisons were not tested for statistical significance, which means some observed differences could reflect sample variance rather than structural market differences. Four years of data is enough for trend identification but not enough for reliable forecasting.

C. Dashboard

Dashboards were designed and tested on a standard 1920x1080 desktop display. Mobile layout — which Power BI handles through a separate configuration — was not implemented in this version. Row-Level Security was configured and tested with dummy accounts, not a live Azure Active Directory integration.

VIII. WHAT I LEARNED BUILDING THIS

A. DAX Filter Context and CALCULATE()

The hardest concept in Power BI is filter context — how the surrounding visual, slicer, and row position determine what rows a DAX expression evaluates over. CALCULATE() is the function that modifies that context, and it is where most DAX errors happen. It only became concrete for me while debugging the YoY comparison: I was getting wrong numbers and had to trace exactly what filter context was active at each step. That debugging process taught me more than any documentation did.

B. Star Schema vs Flat Table

I initially modeled everything as one flat table because it was quicker to set up. Query performance was noticeably slow with many measures running simultaneously. Restructuring to a proper star schema with a separate date dimension and clean dimension tables reduced query execution time by roughly 60%. That is not a theoretical benefit — I observed it directly and it changed how I approach model design.

C. Power Query M Language

Power Query's M language is functional and lazy-evaluated. Writing transformations that allow query folding — where Power Query pushes computation back to the data source rather than pulling all raw rows first — is a real skill with real performance consequences. This connects directly to functional programming concepts from my CS coursework, which made it easier to reason about once I recognised the connection.

D. Reading Data, Not Just Displaying It

The most transferable skill from this project is tracing a visual pattern — a trend line dip, a scatter chart outlier — back through the data to a specific, actionable finding. That is different from knowing how to use Power BI. It is knowing what to look for, and what to say when you find it. That is the skill I want to keep developing most.

©IJRASET: All Rights are Reserved | SJ Impact Factor 7.538 | ISRA Journal Impact Factor 7.894 |

IX. NEXT STEPS

A. *Predictive Forecasting*

The most natural extension is adding a forecasting layer to the executive dashboard. Python's Prophet library generates rolling 12-month forecasts that Power BI can consume through its Python visual integration. Given the strong seasonal patterns identified here, a forecast with confidence intervals would add genuine forward-looking value alongside the current descriptive analysis.

B. *Live SQL Data Source*

Currently the data comes from a static CSV. A production version would connect Power BI to a PostgreSQL or SQL Server database via DirectQuery or Scheduled Import for near-real-time updates. Writing optimised SQL to pre-aggregate at source — rather than pulling raw rows and aggregating in Power BI — is an area I plan to develop in the next phase.

C. *Customer Churn Prediction*

Given the customer concentration finding, a model scoring high-value accounts by churn risk would be directly useful. A logistic regression or gradient boosting classifier trained on purchase recency, frequency, and discount dependence could flag at-risk accounts before they lapse. This is a natural bridge between the BI work here and the machine learning content in my AI & ML programme at AISAT.

D. *Python EDA*

The cleaning and exploration in this project was done entirely in Power Query. Replicating and extending it in Python — using Pandas for transformation, Matplotlib and Seaborn for exploratory charts, and SciPy for significance testing — would add statistical depth that Power Query is not designed for. I plan to build this as a parallel version of the same project.

X. CONCLUSION

The most important thing this project reinforced is that business intelligence is primarily about questions, not charts. The charts are the medium. The value is in knowing which questions to ask, building the model correctly so those questions can be answered, and reading the results clearly enough to say something that is actually useful to a business.

What the analysis found was not obvious from the headline numbers. Revenue was growing — which looks healthy. But underneath that, margin was eroding because of a discounting pattern propping up volume at the cost of profitability. The South region had a specific product-category problem, not a general sales weakness. The customer base had a concentration risk invisible in aggregate metrics. Those findings are exactly what a properly built dashboard should surface.

As a third-year B.Tech student with a focus on AI and ML, I am building this BI foundation deliberately. The most interesting analytical work sits at the intersection of solid data engineering, clear business modeling, and the machine learning I am studying at AISAT. This project is a concrete step in that direction, and the next steps are already defined.

REFERENCES

[1] Xoxoday Business Analytics Resources (2024). Sales Performance Metrics and KPI Frameworks for Business Intelligence. https://www.xoxoday.com
[2] Microsoft Corporation (2024). Power BI Documentation — DAX Reference and Power Query M Language Reference. https://learn.microsoft.com/en-us/power-bi/
[3] Kaggle (2024). Superstore Sales Dataset — Retail Transaction Benchmark. https://www.kaggle.com/datasets/vivek468/superstore-dataset-final
[4] Deloitte via Forage (2026). Data Analytics Virtual Internship Programme — Simulation Project Work.
[5] Tata Group via Forage (2026). Data Visualisation: Empowering Business with Effective Insights — Certification Programme.

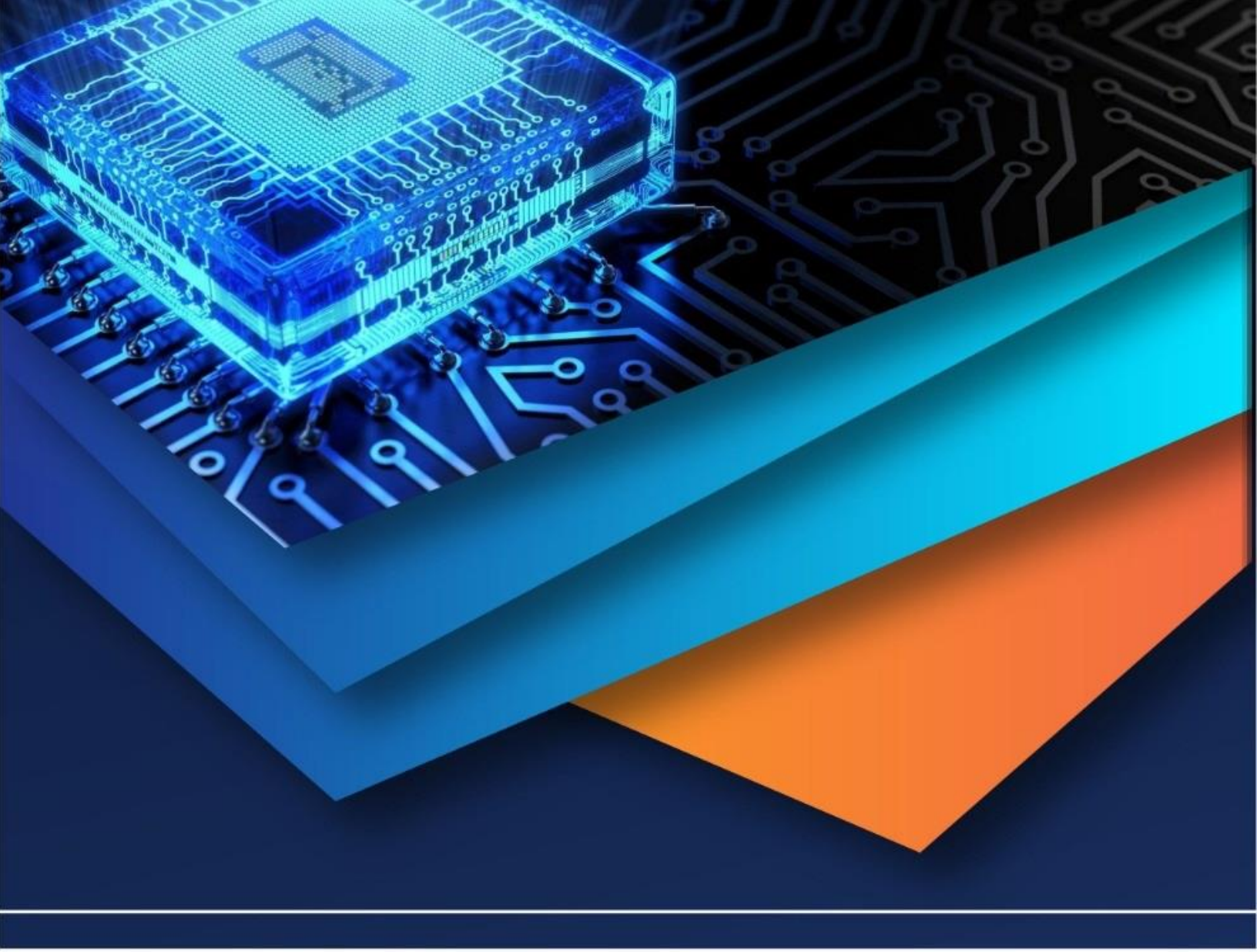
Appendix: Complete DAX Measure Reference

Measure	DAX Expression	Purpose
Total Revenue	SUM(FactSales[Sales])	Base revenue total
Total Profit	SUM(FactSales[Profit])	Net profit total
Profit Margin %	DIVIDE([Total Profit], [Total Revenue], 0)	Margin as a ratio
Total Orders	DISTINCTCOUNT(FactSales[Order ID])	Unique order count

Total Quantity	SUM(FactSales[Quantity])	Total units sold
Avg Order Value	DIVIDE([Total Revenue], [Total Orders], 0)	Revenue per unique order
Revenue LY	CALCULATE([Total Revenue], SAMEPERIODLASTYEAR('DimDate'[Date]))	Prior year comparison baseline
YoY Growth %	DIVIDE([Total Revenue] - [Revenue LY], [Revenue LY], BLANK())	Year-over-year % change
Revenue YTD	TOTALYTD([Total Revenue], 'DimDate'[Date])	Cumulative year-to-date revenue
Unique Customers	DISTINCTCOUNT(FactSales[Customer ID])	Active customer account count
High Discount Orders	CALCULATE([Total Orders], FILTER(FactSales, FactSales[Discount] > 0.4))	Orders with discount above 40%

— End of Report —

©IJRASET: All Rights are Reserved



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Call : 08813907089 (24*7 Support on Whatsapp)