Retail Sales Analytics – SQL Data Layer & Governance Framework

Project Objective

This project extends the Retail Sales Power BI dashboard by implementing a structured SQL data layer with controlled transformations, referential integrity, and certified revenue logic before exposing data to reporting tools.

The goal was to simulate enterprise-level architecture:

Raw Data → Clean Star Schema → Certified Metric Layer → Power BI Reporting

Architecture Overview

1. Staging Layer (stg_)

    - Direct import of raw CSV data
    - No transformations applied
    - Used for audit and validation checks

    Tables:

    - stg_Customers
    - stg_Products
    - stg_Stores
    - stg_Transactions

2. Clean Layer (Star Schema)

Implemented structured dimensional modeling:

Dimension Tables

- dim_Customers
- dim_Products
- dim_Stores

Fact Table

- fact_Transactions

Key Design Decisions

- Business keys stored as NVARCHAR(50) (e.g., C001, P002)
- Primary Keys enforced on all dimensions
- Foreign Keys enforced in fact table
- CHECK constraints applied on:
    - Quantity (> 0)
    - Discount (0–1 range)
    - UnitPrice and CostPrice (≥ 0)

This ensures referential integrity and data quality control at the database level.

Data Quality Validation

Before loading into clean tables, the following checks were performed:

- NULL checks on primary identifiers
- Duplicate key validation
- Negative quantity detection
- Discount range validation
- Unit price vs cost price validation

Example validation query:

```
SELECT CustomerID, COUNT(*)
FROM stg_Customers
GROUP BY CustomerID
HAVING COUNT(*) > 1;
```

All staging data passed integrity checks prior to transformation.

Certified Revenue Definition

To prevent metric drift, revenue logic was centralized in a SQL view:

vw_Certified_Revenue

Revenue Formula

Net Revenue = Quantity × UnitPrice × (1 - Discount)

Gross Profit Formula

Gross Profit = Net Revenue - (Quantity × CostPrice)

SQL Implementation:

```
CREATE OR ALTER VIEW vw_Certified_Revenue AS
SELECT
    t.TransactionID,
    t.Date,
    t.CustomerID,
    t.ProductID,
    t.StoreID,
    t.Quantity,
    p.UnitPrice,
    p.CostPrice,
    t.Discount,
    (t.Quantity * p.UnitPrice * (1 - t.Discount)) AS CertifiedRevenue,
    ((t.Quantity * p.UnitPrice * (1 - t.Discount))
        - (t.Quantity * p.CostPrice)) AS GrossProfit
FROM fact_Transactions t
JOIN dim_Products p
    ON t.ProductID = p.ProductID;
```

This ensures:

- Single source of truth for revenue
- Consistency across reports
- Prevention of alternate revenue calculations

Analytical SQL Demonstrations

To validate business logic prior to visualization, the following analytical queries were implemented:

Year-wise Revenue

```
SELECT
    YEAR(Date) AS Year,
    SUM(CertifiedRevenue) AS TotalRevenue,
    SUM(GrossProfit) AS TotalProfit
FROM vw_Certified_Revenue
GROUP BY YEAR(Date)
ORDER BY Year;
```

Top Products by Revenue (Ranking)

```
SELECT TOP 5
    p.ProductName,
    SUM(v.CertifiedRevenue) AS TotalRevenue,
    RANK() OVER (ORDER BY SUM(v.CertifiedRevenue) DESC) AS RevenueRank
FROM vw_Certified_Revenue v
JOIN dim_Products p ON v.ProductID = p.ProductID
GROUP BY p.ProductName
ORDER BY TotalRevenue DESC;
```

Month-over-Month Revenue (CTE + Window Function)

```
WITH MonthlyRevenue AS (
    SELECT
        YEAR(Date) AS YearNum,
        MONTH(Date) AS MonthNum,
        SUM(CertifiedRevenue) AS TotalRevenue
    FROM vw_Certified_Revenue
    GROUP BY YEAR(Date), MONTH(Date)
)
SELECT
    YearNum,
    MonthNum,
    TotalRevenue,
    LAG(TotalRevenue) OVER (ORDER BY YearNum, MonthNum) AS PreviousMonthRevenue,
    TotalRevenue
        - LAG(TotalRevenue) OVER (ORDER BY YearNum, MonthNum) AS RevenueChange
FROM MonthlyRevenue
ORDER BY YearNum, MonthNum;
```

Validation Against Power BI

All SQL aggregates were validated against Power BI DAX measures to ensure full consistency between:

- SQL data layer
- Semantic layer
- Dashboard outputs

No discrepancies were found.

Key Outcomes

- Implemented layered SQL architecture (Raw → Clean → Certified)
- Enforced referential integrity via PK/FK constraints
- Centralized revenue governance logic
- Demonstrated CTEs and window functions
- Validated analytical outputs prior to visualization

This project demonstrates full ownership of the data lifecycle — from ingestion and validation to governed metric design and analytical reporting.