DATA WAREHOUSING USING SQL

PROJECT REPORT

submitted by

Aniket Tripathi & Ashok Kumar

to

The National Institute of Electronics & IT, Chennai in partial fulfilment
of the requirements for the program

PG Program in Data Engineering

Data Science Group

National Institute of Electronics and Information Technology, Chennai

(An Autonomous Scientific Society of Ministry of Electronics and Information Technology, Government of India)

No.: 25, Gandhi Mandapam Road, Chennai – 600025, Tamil Nadu

www.nielit.gov.in/chennai

September, 2025

DECLARATION

I undersigned hereby declare that the project report "**Data Warehousing Using SQL**", submitted for partial fulfilment of the requirements of the program '**PG program in Data Engineering**' at National Institute of Electronic & IT, Chennai is a bonafide work done by our team consisting of Mr. Aniket Tripathi and Mr. Ashok Kumar under supervision of **Mr. Sourav Acherjee** . This submission represents my ideas in my own words and where ideas or words of others have been included, I have adequately and accurately cited and referenced the original sources. I also declare that I have adhered to academic honesty and integrity ethics and have not misrepresented or fabricated any data, idea, fact, or source in my submission. I understand that any violation of the above will cause disciplinary action by the institute and/or the University and can also evoke penal action from the sources that have thus not been properly cited or from whom proper permission has not been obtained. This report has not previously formed the basis for awarding any degree, diploma or similar title of any other University.

Place: New Delhi   Ashok Kumar

Date: 08/09/2025   Aniket Tripathi

CERTIFICATE

This is to certify that project work entitled **Data Warehousing Using SQL** submitted by **Aniket Tripathi, Ashok Kumar** in partial fulfillment of the requirements for the PG program in Data Engineering at NIELIT Chennai is a bonafide record of the work carried out by him/her under my guidance and supervision from to 10/07/2025 to 28/07/2025.

Mr. Sourav Acherjee	Mr. Ishant Kumar Bajpai
Scientist 'B' & Course Coordinator	Scientist 'D' & Group Head
NIELIT Chennai	NIELIT Chennai

Date: 08/09/2025

ACKNOWLEDGEMENT

The completion of any project depends upon cooperation, coordination and combined efforts of several sources of knowledge. This report acknowledges a number of guidance, supervision, stimulation and lot of inspiration from numerous people. I would like to express my gratitude to all who helped me directly or indirectly in the successful completion of my project work.

I express my gratitude to **Dr. Sanjeev Kumar Jha**, Director I/C, NIELIT, Chennai for his persistent support.

I express my sincere gratitude to **Mr. Ishant Kumar Bajpai**, Scientist 'D' & Group Head for his guidance, encouragement and consistent support in completing the project successfully.

I wish to thank to **Mr. Sourav Acherjee**, Scientist 'B' & Course Coordinator, and all the staff of the **Data Science** Group for their valuable feedback and support.

ABSTRACT

This paper outlines the design and implementation of a modern data warehouse and analytics framework, designed to demonstrate end-to-end data engineering and analytical abilities. The solution adopts the Medallion Architecture- Bronze, Silver, and Gold layers, to systematically manage data quality and usability. Data is ingested from multiple source systems, including ERP and CRM datasets provided as CSV files, and consolidated into a SQL Server-based data warehouse. The Bronze level processes and maintains raw, undigested data in a traceable manner; the Silver level is focused on data cleansing, standardizing, and combining; and the Gold level provides a refined, business-ready data model designed in star schema format in order to support quick reporting and analysis.

The project implements sturdy ETL pipelines that make data ingestion, transformation, and loading automated, thereby ensuring reliability and consistency. A well-defined dimensional data model accommodates analytical query and reporting situations. SQL based analytics provide timely and precise customer behavior, product, and sales trend insights, thus offering informative metrics to the decision-makers. The project is a portfolio-ready demonstration of data warehousing industry-standard procedures, showing data engineering, modeling, and analytics expertise in addressing real-world business issues. The careful documentation and clear architectural design work well in boosting the scalability and flexibility of the solution in the enterprise environments as well.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1

INTRODUCTION

In the contemporary landscape characterized by data-driven dynamics, enterprises produce extensive quantities of data from various systems, including Enterprise Resource Planning (ERP), Customer Relationship Management (CRM), and additional operational tools. Nonetheless, unrefined data in its original state frequently suffers from deficiencies in structure, consistency, and accessibility, which are critical for informed decision-making. To mitigate this challenge, organizations are progressively depending on data warehouses-centralized repositories that are specifically engineered to store, organize, and analyze substantial volumes of data.

This project focuses on the development and deployment of a modern data warehouse and analytics architecture that complies with standard industry best practices. By embracing the Medallion Architecture that comprises Bronze, Silver, and Gold levels, this architecture details the systematic processing of raw data, its standardization through transformations, and its readiness in the production of business insights. The data warehouse assimilates information across diverse sources, consequently enabling the building of actionable dashboards and analytical reports that inform decision-makers.

The implementation uses SQL Server as the core data warehouse foundation with various other tools like Draw.io for architectural diagrams, and Notion for documentation.The approach both proves technical competency but also reflects the work patterns of modern-day enterprise data engineering teams.

1.1 Problem Statement

Businesses usually struggle when data is dispersed over various systems and is in dissimilar formats. If no well-organized data warehouse exists:

- Data analysis becomes time-consuming and error-prone.
- Business teams lack a single source of truth for decision-making
- Scaling is complicated when trying to support bigger datasets or more people.

This project aims to overcome the above challenges through the development of a centralized high-quality, as well as analytics-ready data warehouse, showcasing end-to-end data engineering capabilities.

1.2 objectives

Primary goals of this program are:

- Design and development of a modern data warehouse based on Medallion Architecture principles.
- To create ETL pipelines that extract, transform, and import data from CSV files into SQL Server.
- To build a star schema with dimension and fact tables that support efficient analytical queries.
- To insure data consistency, traceability, and quality with standard transformations.
- To offer actionable insights through SQL-based analytics and Power BI visualizations.

1.3 Delimitation of the Study

- The project is envisioned as a portfolio project, reflecting practical use of the data engineering concepts.
- The data is from simulated ERP and CRM systems through CSV files.
- The focus is entirely on the current dataset at hand; versioning with historical data is beyond the scope.

- The current implementation is done in SQL Server but can be extended to cloud infrastructure in future works.
- Reporting is based on crucial indications, including end-user behavior, product efficiency, and sales patterns.

1.4 Significance of the Project

It is the practical representation of the way raw operational data is being transformed into actionable insights. The project reflects ETL development best practices, data modeling, and business intelligence. Adhering to the systematic way, the project offers:

- A centralized and trusted source of data for analysis.
- A template for horizontally scalable data pipelines.
- A basis for the prospective amalgamation of sophisticated analytical methods and machine learning frameworks.

CHAPTER 2

LITERATURE SURVEY

S.No	Literature	Year	Technology	Insights
1	Databricks. (2022). The Medallion Architecture in Lakehouse Design. Retrieved from Databricks Blog.	2022	Medallion Architecture, Lakehouse	Introduces the Bronze, Silver, and Gold layered architecture to ensure traceability and usability.
2	Inmon, W. H. (2005). Building the Data Warehouse (4th ed.). Wiley.	2005	Data Warehousing	Establishes foundational principles of enterprise data warehouses and data integration strategies
3	Kimball, R., & Ross, M. (2013). The Data Warehouse Toolkit (3rd ed.). Wiley.	2013	Dimensional Modeling, Star Schema	Explains dimensional modeling best practices for scalable and easy-to-query data warehouse design.
4	Microsoft. (2023). SQL Server Data Warehousing Guide. Microsoft Docs.	2023	SQL Server, ETL Pipelines	Provides technical guidance for implementing ETL pipelines and analytical queries in SQL Server.

CHAPTER 3

METHODOLOGY

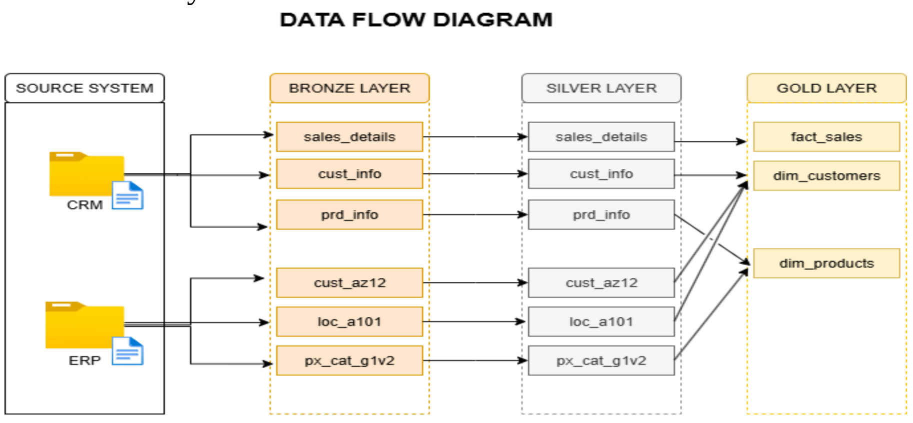
The methodology applied in this project is a systematic approach to designing and implementing a modern data warehouse and analytics platform. The goal was to build a scalable, reliable, and analytics-ready system aligned with industry-wide best practices. The following are the steps applied in the methodology:

1. **Requirement Analysis**

The project began with a study of data requirements and reporting objectives. Source data sets were reviewed in order to identity key entities, such as customers, products, and sales orders. The business requirements emphasized needing to draw insights on the basis of trends in sales, product, and customer activity, thus shaping the data modeling and pipeline development approach.

2. Data Acquisition

Data was extracted from two systems, i.e., ERP(Enterprise Resource Planning) and CRM(Customer Relationship Management). Both the data were delivered as CSV files. These files were directly loaded into SQL Server data warehouse in Bronze layer, without transformation, so that the raw data is intact and usable in case of audit and traceability.



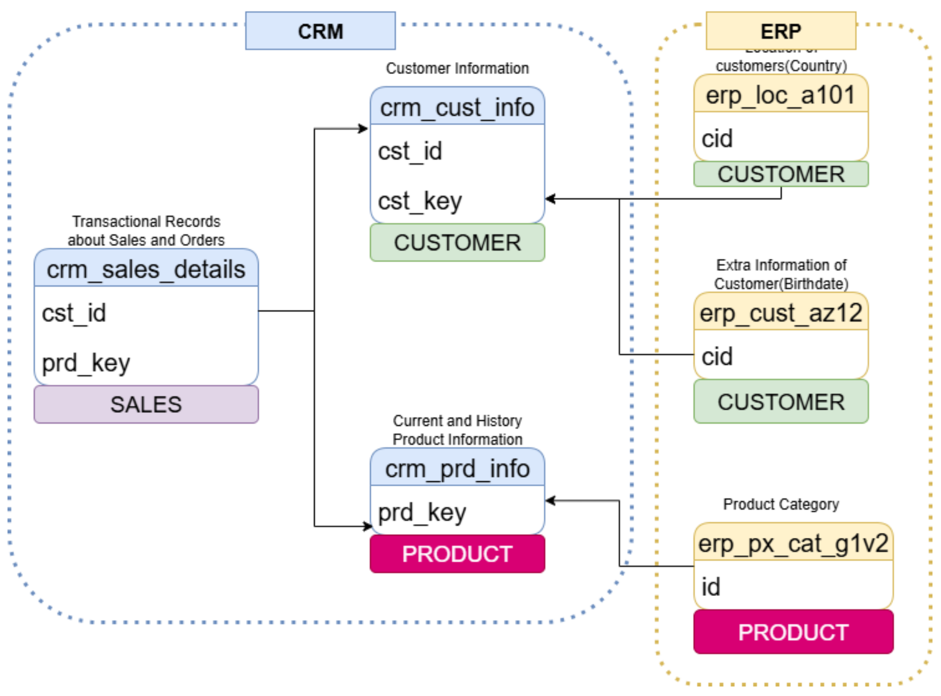
3. Data Transformation and Integration

After ingestion, data went through several transformation stages:

3.1)Data Cleansing: Duplicate removal, management of null or empty fields, and data type standardization.
3.2)Data Integration: Combining ERP and CRM data so as to make a single dataset.
3.3)Normalization: Making the format of dates, IDs, and categorical attributes consistent.

These procedures were performed in the Silver layer, generating data that was normalized and clean.

Integration Model(How Tables are related)



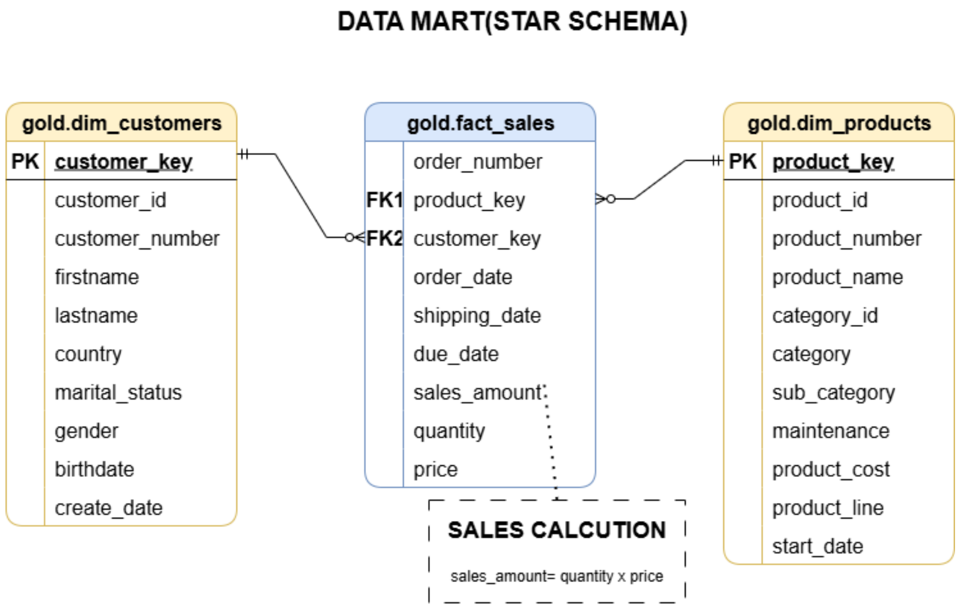
4. Dimensional Data Modeling

A star schema was developed to enhance the efficiency of analytical queries. This schema comprised:

Fact Tables: Consist of measurable business events, e.g., sales transactions.

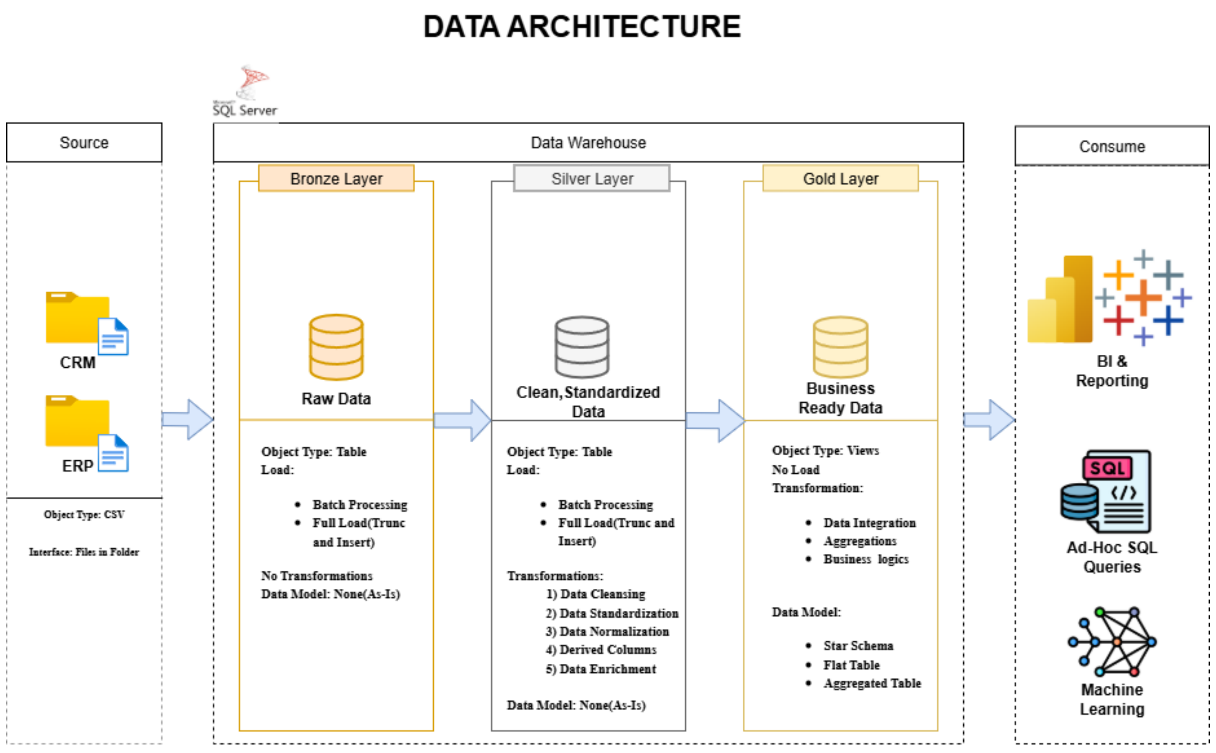
Dimension Tables: Accommodating Descriptive Attributes for Dimensions Like Customers, Products, and Dates.

This design reduced the query complexity and simplified reporting performance.



5. ETL Pipeline Construction

Automated ETL pipelines were created to extract raw data, perform transformations, and load it into the appropriate layers of the warehouse. These pipelines were developed with modularity in mind, enabling easy maintenance and extension. Incremental load logic was implemented to process only new or updated records, improving performance and efficiency.



6. Analytics and Reporting

As data in the Gold layer was populated with business-ready data, SQL analytical queries were developed. Key KPIs, such as total sales, product, and customer segmentation, were generated. These were utilized as a foundation for reporting and dashboards, enabling data-driven decisions.

This methodology reflects a practical, iterative approach to building a data warehouse solution. By following the Medallion Architecture and dimensional modeling principles, the project ensures scalability, reliability, and actionable analytics for business stakeholders.

CHAPTER 4

HARDWARE TOOLS AND COMPONENT

The successful execution of this Data Warehouse and Analytics Project relies on a stable and efficient hardware setup to support the Extract, Transform, and Load (ETL) processes, data storage, and analytical operations. The hardware environment was chosen to ensure scalability, reliability, and sufficient processing power for managing large datasets and performing analytical computations.

4.1 SYSTEM REQUIREMENTS:

S.No.	Component	Specification
1	Processor(CPU)	Intel Core i5/i7 (8th Gen or higher) or equivalent AMD processor, 2.4 GHz or above
2	Memory(RAM)	16 GB DDR4 RAM (Minimum 8 GB for small datasets)
3	Storage	512 GB SSD (Minimum 1 TB recommended for large datasets)
4	Operating System(OS)	Windows 10/11 Pro or Windows Server 2019
5	Graphics	Integrated Graphics (No dedicated GPU required)
6	Network	High-speed Internet (100 Mbps or higher)

4.2 HARDWARE JUSTIFICATION

The chosen hardware components were selected based on the following considerations:

1. Performance: Multi-core processors and adequate RAM ensure smooth ETL operations and analytical query execution.
2. Scalability: The configuration allows easy scaling to handle larger datasets and additional layers of transformation.
3. Cost Efficiency: Leveraging a personal workstation and open-source or developer editions of software ensures a cost-effective development process.
4. Reliability: SSD-based storage provides high reliability and minimizes latency during query processing.

This hardware setup provided a stable and efficient environment for designing, implementing, and testing the data warehouse solution. It also ensures that the project can be deployed to enterprise environments with minimal adjustments.

CHAPTER 5

SOFTWARE TOOLS AND LANGUAGES

This Data Warehouse and Analytics Project was developed and created using a mix of database management systems, analytics software, repository documentation software, and diagramming software. Each tool and programming language was specifically chosen with the aim of efficiency, scalability, and best practices.

5.1 SOFTWARE TOOLS USED

S.NO	Tool/Software	Purpose	Remarks
1	Microsoft SQL Server 2019	Relational Database Management System (RDBMS) used for data warehousing, schema design, and storage.	Offers enterprise-grade scalability.
2	SQL Server Management Studio	Integrated environment for database management, query writing, and debugging	Simplified database development and maintenance.
3	PowerBI(optional)	Data visualization and dashboard creation tool for interactive analytics	Adds a visual dimension to insights.
4	GitHub	Version control and repository hosting for project code and documentation.	Ensured version tracking and collaboration
5	Draw.io	Used to design architecture diagrams, data flow models, and ER diagrams.	Facilitated clear project visualization.
6	Notion	Project documentation, task planning, and note-taking	Enabled organized collaboration and knowledge management
7	Windows 10/11	Operating system hosting development and database tools	Stable and widely supported

5.2 Programming Languages

S.No.	Language	Purpose
1	SQL	Primary language for data modeling, data manipulation, and analytics queries.
2	Transact-SQL(T-SQL)	SQL Server extension used for stored procedures, indexing, and performance optimization.

The chosen combination of software tools and programming languages enabled smooth execution of the project, from data ingestion to reporting.

CHAPTER 6

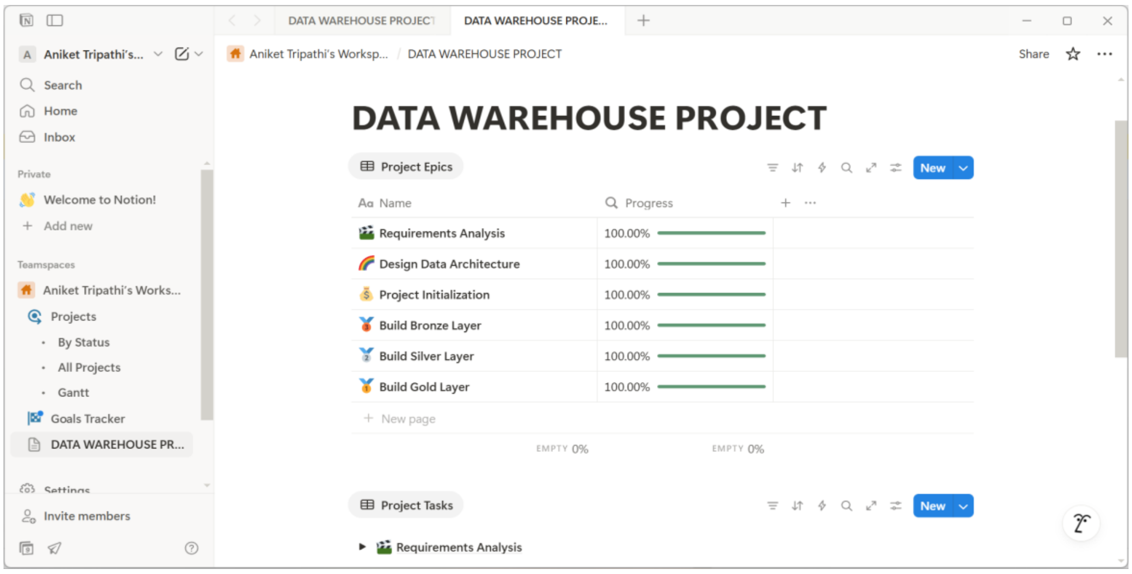
IMPLEMENTATION

This chapter outlines the end-to-end implementation of the Data Warehouse and Analytics Project, detailing the setup, data ingestion workflows, transformations, and analytics layers.

<u>6.1 Project Setup</u>

1) Environment Setup:

- Installed and configured Microsoft SQL Server as the central Data Warehouse platform.
- Set up SQL Server Management Studio (SSMS) for database creation, schema design, and query execution
- Setup a GitHub Repository for version control of SQL scripts, documentation
- Created Data Architecture Design, Data flow Diagram, Data Model Structure using Draw.io
- Created a Notion workspace for documentation, progress tracking, of the whole project by segregating it into various layers.



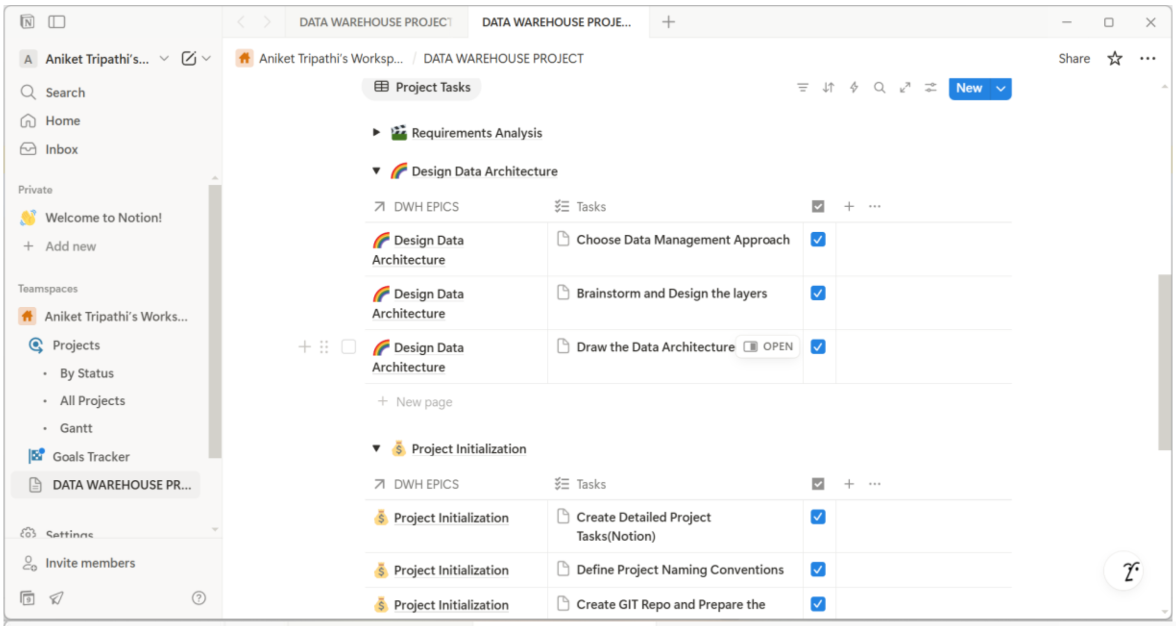
DATA WAREHOUSE PROJECT

Project Epics

Name	Progress
Requirements Analysis	100.00%
Design Data Architecture	100.00%
Project Initialization	100.00%
Build Bronze Layer	100.00%
Build Silver Layer	100.00%
Build Gold Layer	100.00%
EMPTY 0%	EMPTY 0%

Project Tasks

▸ Requirements Analysis



Requirements Analysis

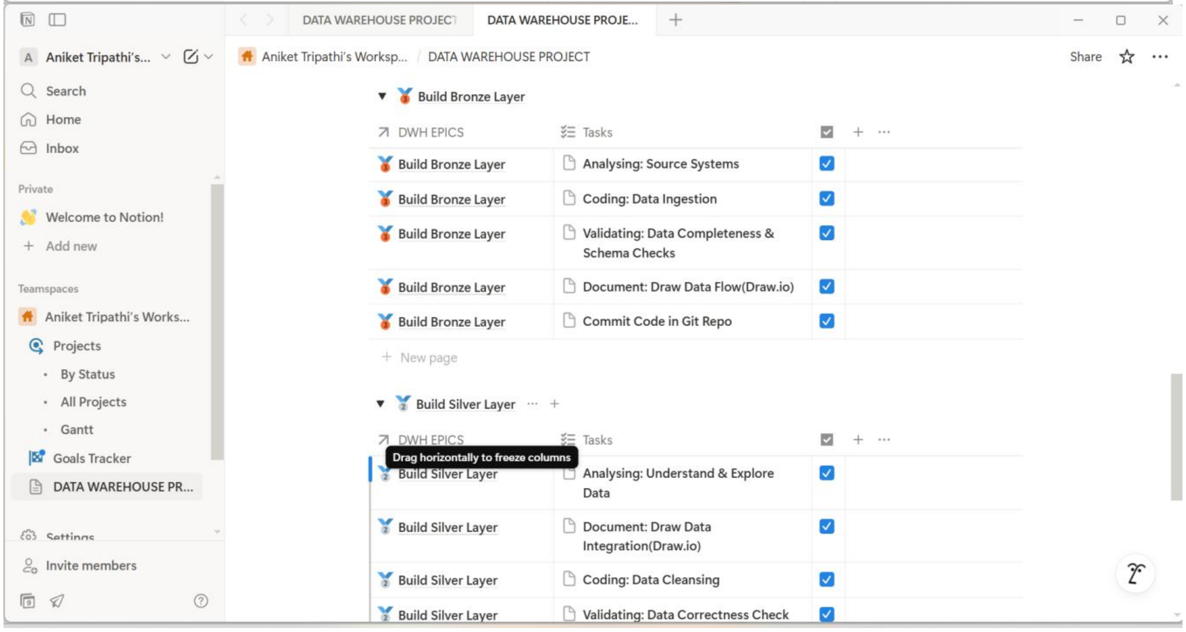
Design Data Architecture

DWH EPICS	Tasks	
Design Data Architecture	Choose Data Management Approach	✓
Design Data Architecture	Brainstorm and Design the layers	✓
Design Data Architecture	Draw the Data Architecture	✓

New page

Project Initialization

DWH EPICS	Tasks	
Project Initialization	Create Detailed Project Tasks(Notion)	✓
Project Initialization	Define Project Naming Conventions	✓
Project Initialization	Create GIT Repo and Prepare the	✓



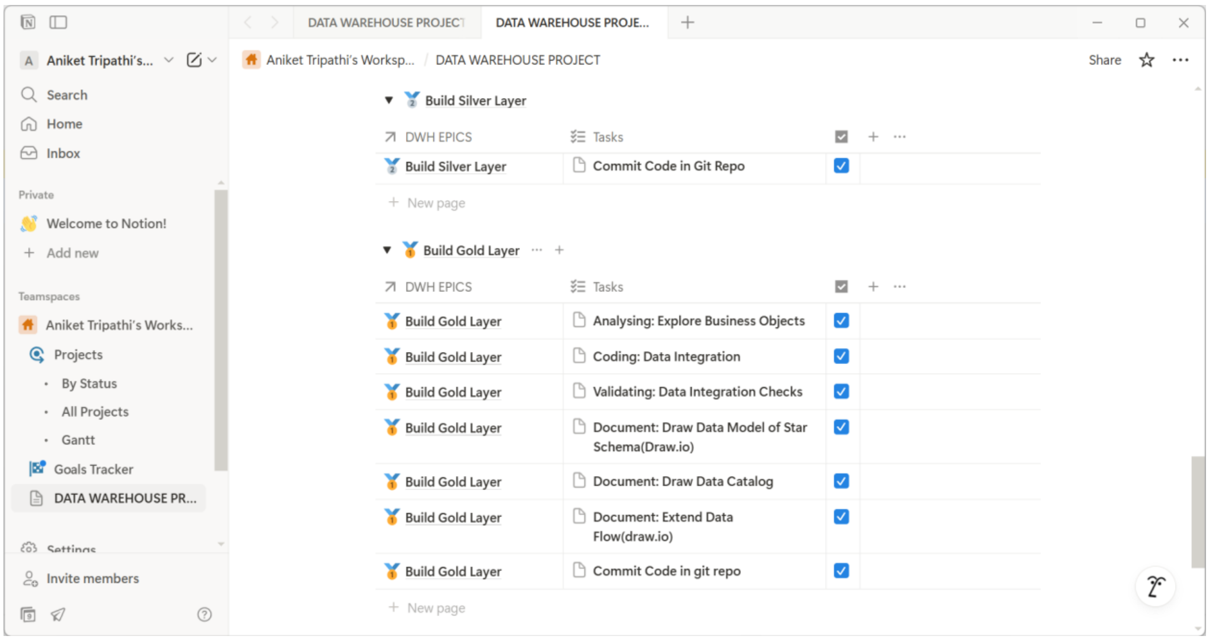
Build Bronze Layer

DWH EPICS	Tasks	
Build Bronze Layer	Analysing: Source Systems	✓
Build Bronze Layer	Coding: Data Ingestion	✓
Build Bronze Layer	Validating: Data Completeness & Schema Checks	✓
Build Bronze Layer	Document: Draw Data Flow(Draw.io)	✓
Build Bronze Layer	Commit Code in Git Repo	✓

New page

Build Silver Layer

DWH EPICS	Tasks	
Build Silver Layer	Analysing: Understand & Explore Data	✓
Build Silver Layer	Document: Draw Data Integration(Draw.io)	✓
Build Silver Layer	Coding: Data Cleansing	✓
Build Silver Layer	Validating: Data Correctness Check	✓



▼ Build Silver Layer

DWH EPICS	Tasks	☑
Build Silver Layer	Commit Code in Git Repo	✓

+ New page

▼ Build Gold Layer

DWH EPICS	Tasks	☑
Build Gold Layer	Analysing: Explore Business Objects	✓
Build Gold Layer	Coding: Data Integration	✓
Build Gold Layer	Validating: Data Integration Checks	✓
Build Gold Layer	Document: Draw Data Model of Star Schema(Draw.io)	✓
Build Gold Layer	Document: Draw Data Catalog	✓
Build Gold Layer	Document: Extend Data Flow(draw.io)	✓
Build Gold Layer	Commit Code in git repo	✓

+ New page

2) Data Preparation Step:

- Collected Datasets from 2 source systems:
  - ERP- Containing Customers, Products and Sales Data
  - CRM- Auxiliary Information about customers and products
- Source Data's format: CSV

6.2 Data Architecture Implementation

The implementation followed the Medallion Architecture to ensure a systematic and maintainable data pipeline:

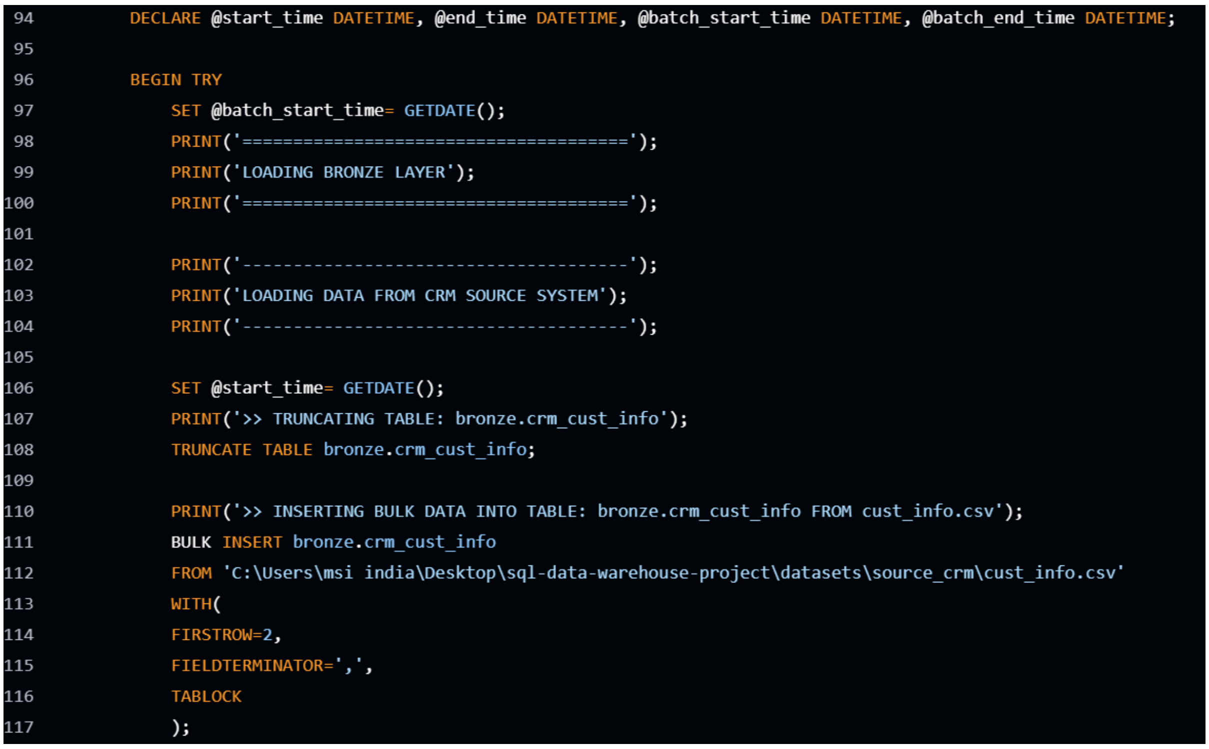
Layer	**Purpose**	**Implementation Details**
Bronze	Raw data storage for traceability.	All CSV files ingested directly into SQL Server raw tables
Silver	Data cleaning, standardization, and integration	Applied transformations to remove duplicates, handle nulls, enforce data types, and integrate ERP & CRM data
Gold	Business-ready data for analytics.	Created Star Schema with Fact and Dimension tables for reporting and analysis.

6.3 ETL Pipeline Development

The Extract, Transform, Load (ETL) process was implemented using T-SQL scripts

6.3.1 Extract Phase

- Read data from CSV source files into SQL Server raw (Bronze) tables using **BULK INSERT**

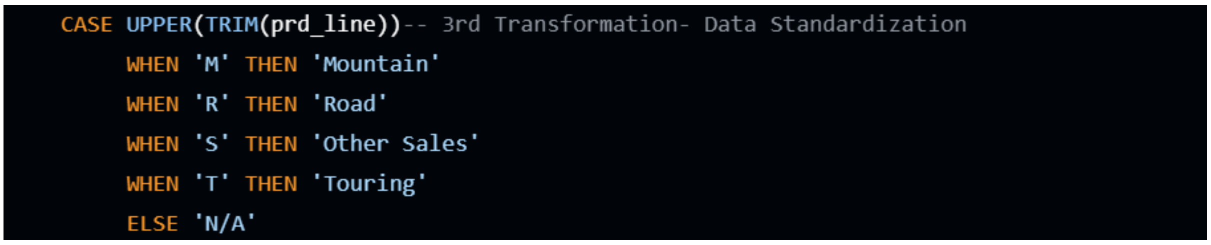


```sql
        DECLARE @start_time DATETIME, @end_time DATETIME, @batch_start_time DATETIME, @batch_end_time DATETIME;

        BEGIN TRY
            SET @batch_start_time= GETDATE();
            PRINT('======================================');
            PRINT('LOADING BRONZE LAYER');
            PRINT('======================================');

            PRINT('--------------------------------------');
            PRINT('LOADING DATA FROM CRM SOURCE SYSTEM');
            PRINT('--------------------------------------');

            SET @start_time= GETDATE();
            PRINT('>> TRUNCATING TABLE: bronze.crm_cust_info');
            TRUNCATE TABLE bronze.crm_cust_info;

            PRINT('>> INSERTING BULK DATA INTO TABLE: bronze.crm_cust_info FROM cust_info.csv');
            BULK INSERT bronze.crm_cust_info
            FROM 'C:\Users\msi india\Desktop\sql-data-warehouse-project\datasets\source_crm\cust_info.csv'
            WITH(
            FIRSTROW=2,
            FIELDTERMINATOR=',',
            TABLOCK
            );
```

- Automated file ingestion scripts ensured new files could be easily added.

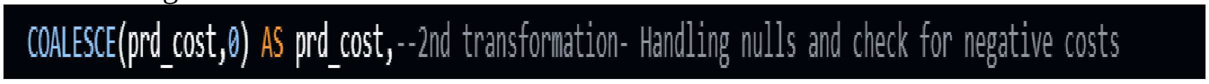
6.3.2 Transformation Phase

- Applied data cleaning rules:
    - Data Standardization



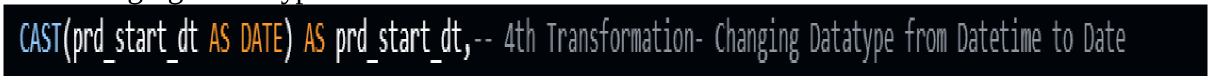
```sql
CASE UPPER(TRIM(prd_line))-- 3rd Transformation- Data Standardization
    WHEN 'M' THEN 'Mountain'
    WHEN 'R' THEN 'Road'
    WHEN 'S' THEN 'Other Sales'
    WHEN 'T' THEN 'Touring'
    ELSE 'N/A'
```

- Handling Nulls



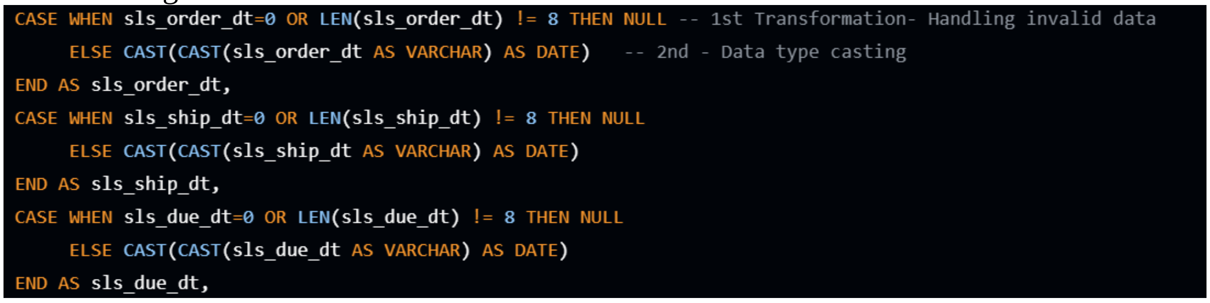
```
COALESCE(prd_cost,0) AS prd_cost,--2nd transformation- Handling nulls and check for negative costs
```

- Changing Data Types



```
CAST(prd_start_dt AS DATE) AS prd_start_dt,-- 4th Transformation- Changing Datatype from Datetime to Date
```

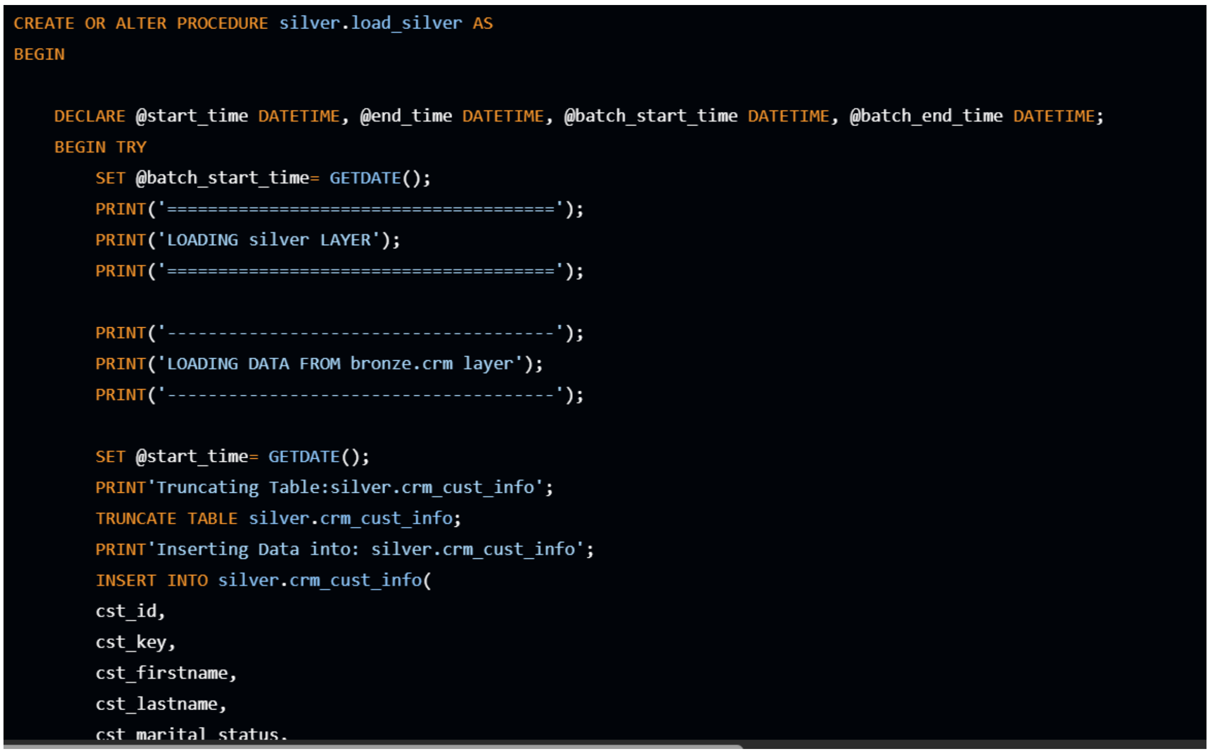
- Handling Invalid Data



```
CASE WHEN sls_order_dt=0 OR LEN(sls_order_dt) != 8 THEN NULL -- 1st Transformation- Handling invalid data
     ELSE CAST(CAST(sls_order_dt AS VARCHAR) AS DATE)   -- 2nd - Data type casting
END AS sls_order_dt,
CASE WHEN sls_ship_dt=0 OR LEN(sls_ship_dt) != 8 THEN NULL
     ELSE CAST(CAST(sls_ship_dt AS VARCHAR) AS DATE)
END AS sls_ship_dt,
CASE WHEN sls_due_dt=0 OR LEN(sls_due_dt) != 8 THEN NULL
     ELSE CAST(CAST(sls_due_dt AS VARCHAR) AS DATE)
END AS sls_due_dt,
```

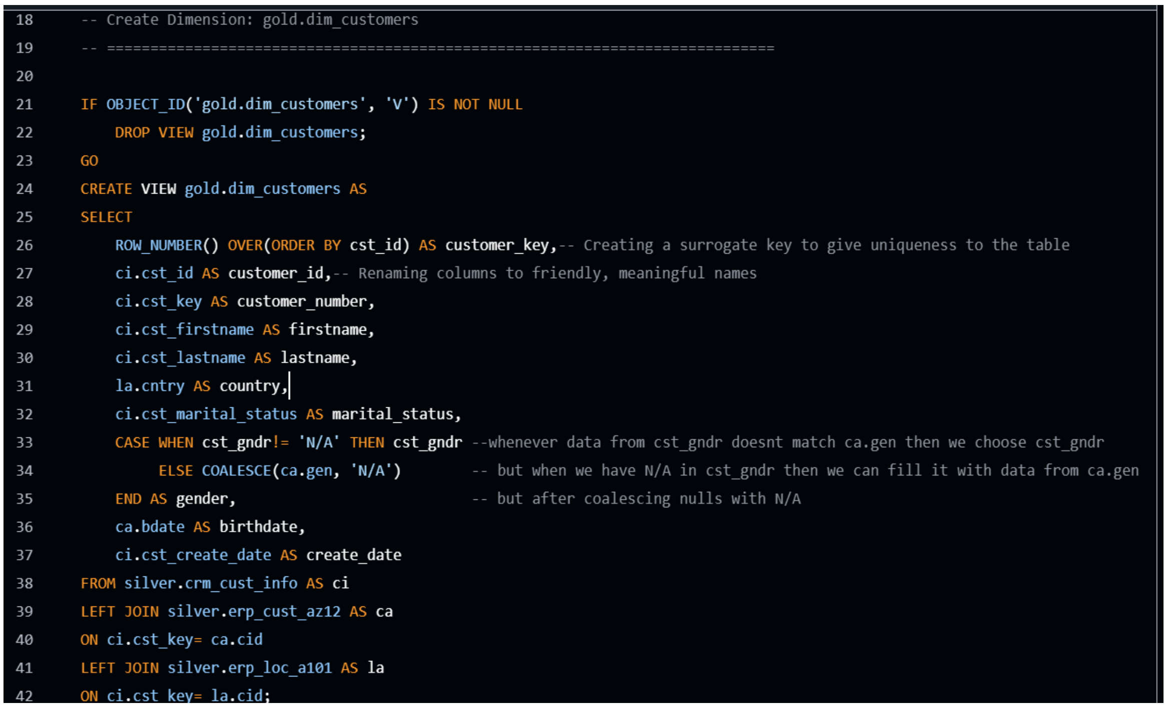
6.3.3 Load Phase

- Loaded cleansed and integrated data into Silver tables

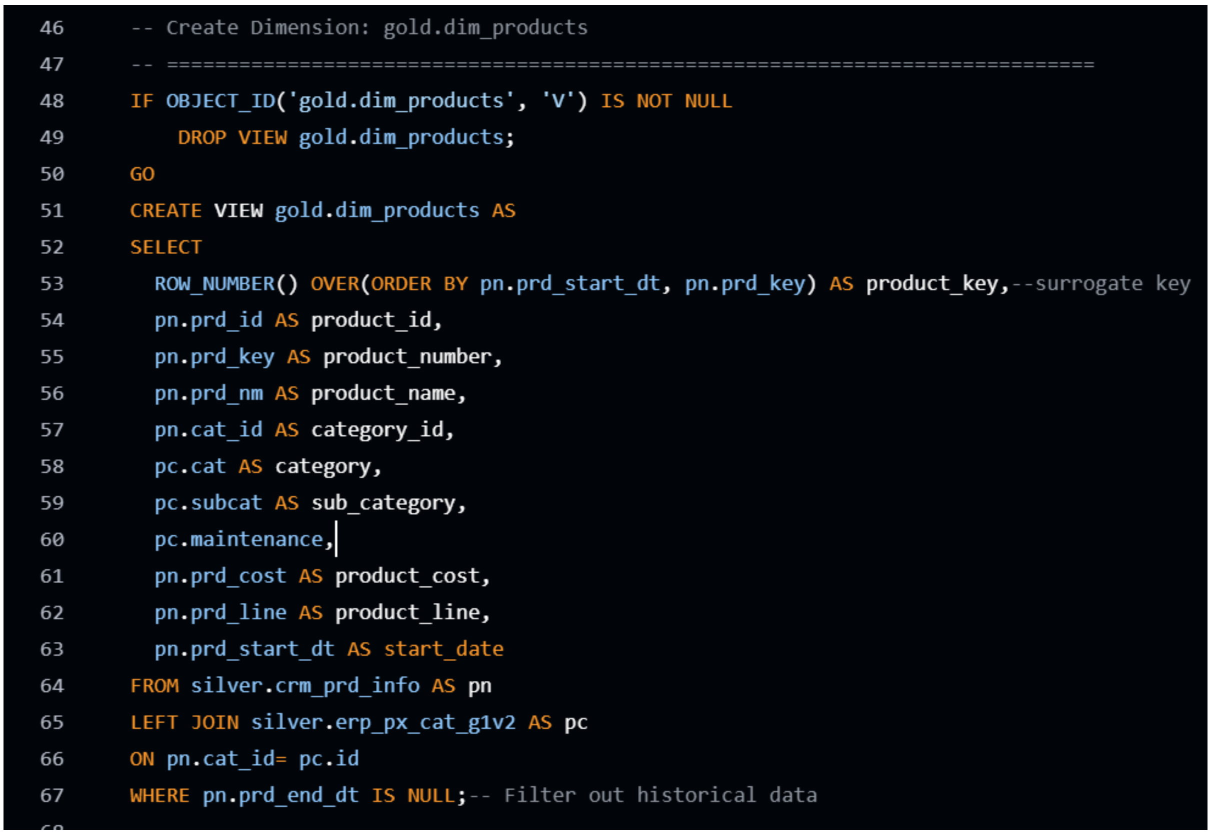


```
CREATE OR ALTER PROCEDURE silver.load_silver AS
BEGIN

    DECLARE @start_time DATETIME, @end_time DATETIME, @batch_start_time DATETIME, @batch_end_time DATETIME;
    BEGIN TRY
        SET @batch_start_time= GETDATE();
        PRINT('=====================================');
        PRINT('LOADING silver LAYER');
        PRINT('=====================================');

        PRINT('-------------------------------------');
        PRINT('LOADING DATA FROM bronze.crm layer');
        PRINT('-------------------------------------');

        SET @start_time= GETDATE();
        PRINT'Truncating Table:silver.crm_cust_info';
        TRUNCATE TABLE silver.crm_cust_info;
        PRINT'Inserting Data into: silver.crm_cust_info';
        INSERT INTO silver.crm_cust_info(
        cst_id,
        cst_key,
        cst_firstname,
        cst_lastname,
        cst_marital_status,
```

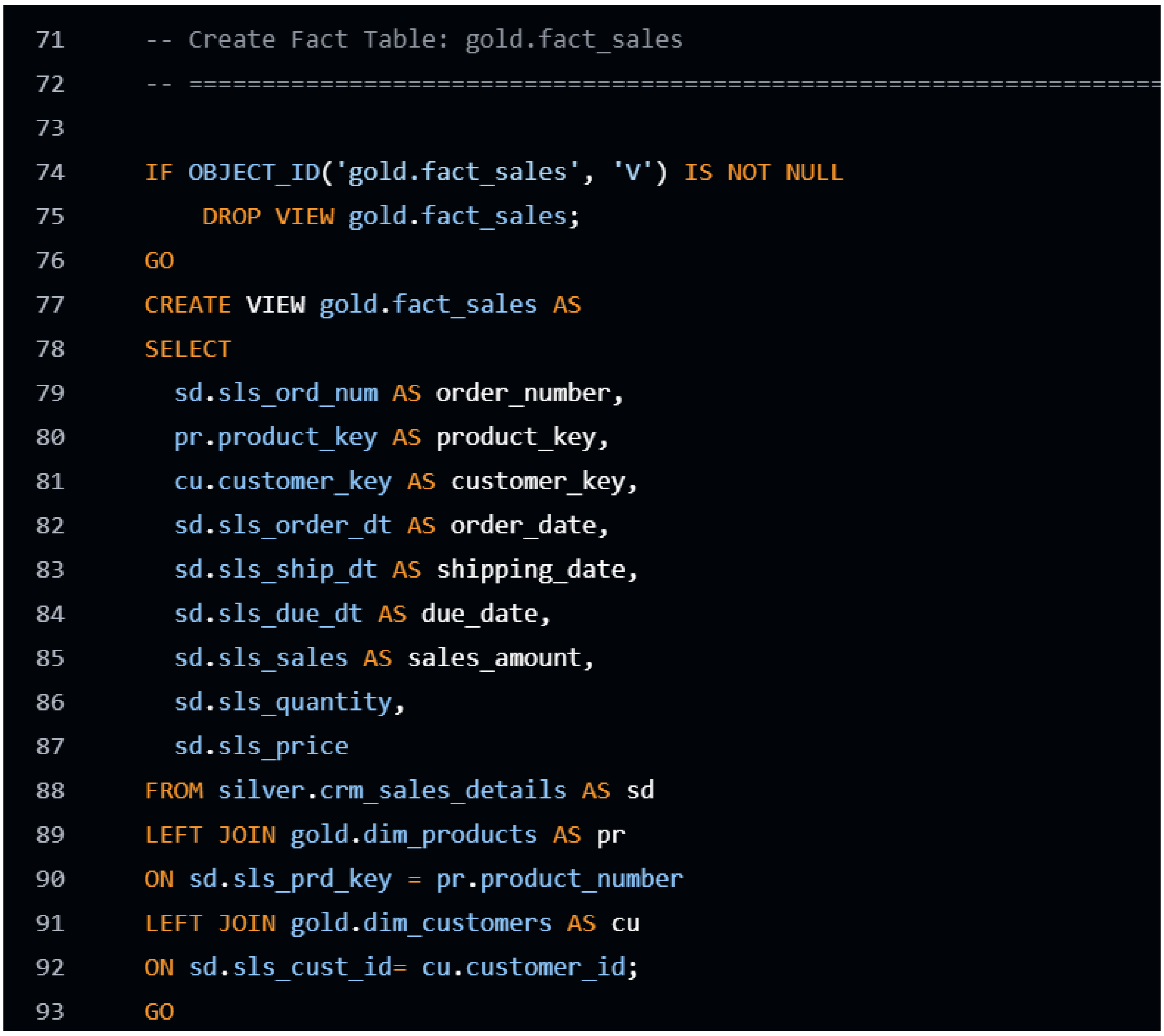
- Created Fact and Dimension tables in the Gold Layer for optimized query performance.



```sql
-- Create Dimension: gold.dim_customers
-- ===========================================================================

IF OBJECT_ID('gold.dim_customers', 'V') IS NOT NULL
    DROP VIEW gold.dim_customers;
GO
CREATE VIEW gold.dim_customers AS
SELECT
    ROW_NUMBER() OVER(ORDER BY cst_id) AS customer_key,-- Creating a surrogate key to give uniqueness to the table
    ci.cst_id AS customer_id,-- Renaming columns to friendly, meaningful names
    ci.cst_key AS customer_number,
    ci.cst_firstname AS firstname,
    ci.cst_lastname AS lastname,
    la.cntry AS country,
    ci.cst_marital_status AS marital_status,
    CASE WHEN cst_gndr!= 'N/A' THEN cst_gndr --whenever data from cst_gndr doesnt match ca.gen then we choose cst_gndr
         ELSE COALESCE(ca.gen, 'N/A')        -- but when we have N/A in cst_gndr then we can fill it with data from ca.gen
    END AS gender,                           -- but after coalescing nulls with N/A
    ca.bdate AS birthdate,
    ci.cst_create_date AS create_date
FROM silver.crm_cust_info AS ci
LEFT JOIN silver.erp_cust_az12 AS ca
ON ci.cst_key= ca.cid
LEFT JOIN silver.erp_loc_a101 AS la
ON ci.cst_key= la.cid;
```



```sql
-- Create Dimension: gold.dim_products
-- ===========================================================================
IF OBJECT_ID('gold.dim_products', 'V') IS NOT NULL
    DROP VIEW gold.dim_products;
GO
CREATE VIEW gold.dim_products AS
SELECT
  ROW_NUMBER() OVER(ORDER BY pn.prd_start_dt, pn.prd_key) AS product_key,--surrogate key
  pn.prd_id AS product_id,
  pn.prd_key AS product_number,
  pn.prd_nm AS product_name,
  pn.cat_id AS category_id,
  pc.cat AS category,
  pc.subcat AS sub_category,
  pc.maintenance,
  pn.prd_cost AS product_cost,
  pn.prd_line AS product_line,
  pn.prd_start_dt AS start_date
FROM silver.crm_prd_info AS pn
LEFT JOIN silver.erp_px_cat_g1v2 AS pc
ON pn.cat_id= pc.id
WHERE pn.prd_end_dt IS NULL;-- Filter out historical data
```



```sql
-- Create Fact Table: gold.fact_sales
-- ==================================================================

IF OBJECT_ID('gold.fact_sales', 'V') IS NOT NULL
    DROP VIEW gold.fact_sales;
GO
CREATE VIEW gold.fact_sales AS
SELECT
  sd.sls_ord_num AS order_number,
  pr.product_key AS product_key,
  cu.customer_key AS customer_key,
  sd.sls_order_dt AS order_date,
  sd.sls_ship_dt AS shipping_date,
  sd.sls_due_dt AS due_date,
  sd.sls_sales AS sales_amount,
  sd.sls_quantity,
  sd.sls_price
FROM silver.crm_sales_details AS sd
LEFT JOIN gold.dim_products AS pr
ON sd.sls_prd_key = pr.product_number
LEFT JOIN gold.dim_customers AS cu
ON sd.sls_cust_id= cu.customer_id;
GO
```

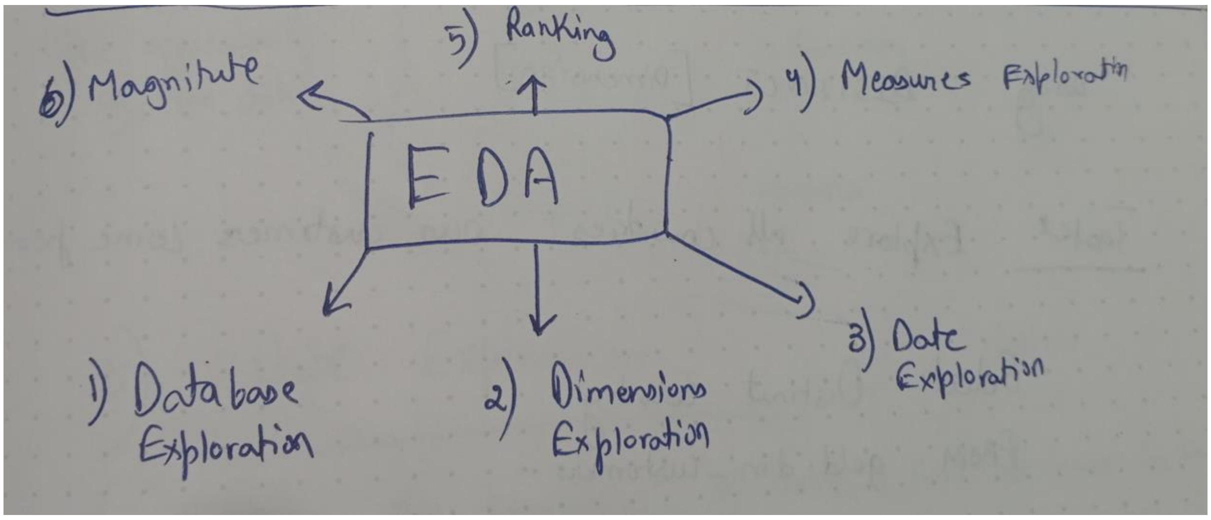
6.4 Data Modeling

A Star Schema was designed for the analytics layer.

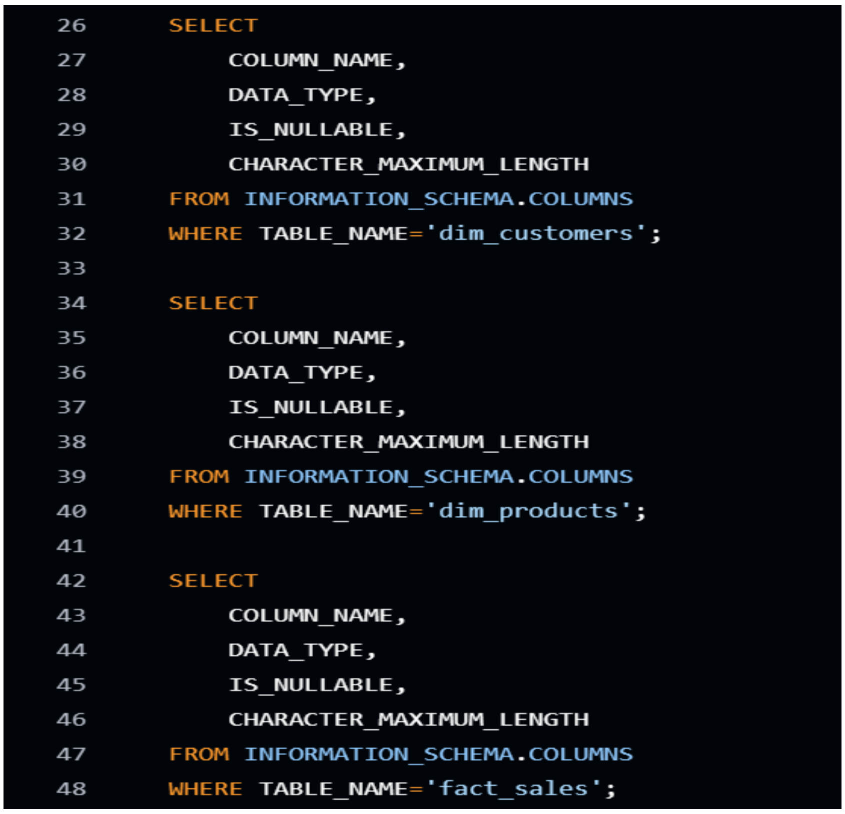
- Fact Table(gold.fact_sales): Stores transactional data like sales_amount, quantity,price
- Dimesion Tables(gold.dim_products, gold.dim_customers): Contains descriptive data like product_name, first_name,last_name
- The schema design simplifies reporting queries and ensures fast aggregations.

6.5 Advance and Exploratory Data Analysis(EDA)

6.5.1 EDA

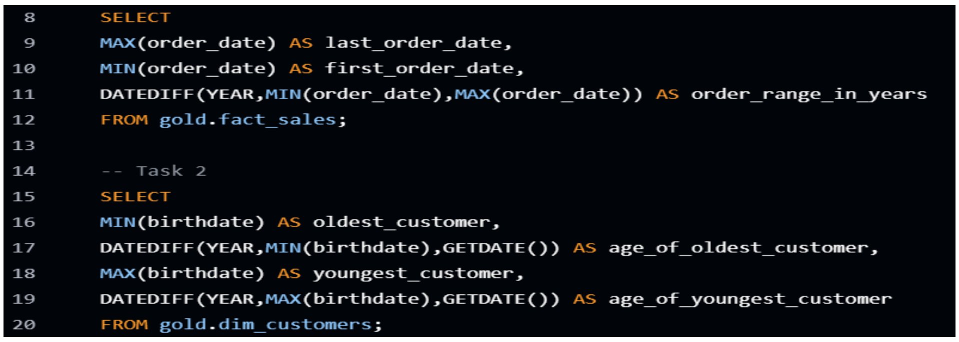


- Database Exploration

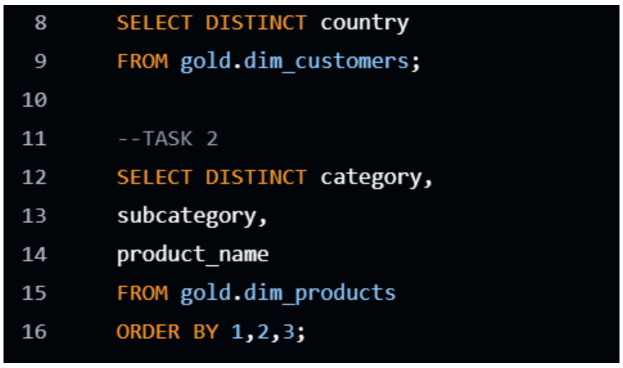


```sql
SELECT
    COLUMN_NAME,
    DATA_TYPE,
    IS_NULLABLE,
    CHARACTER_MAXIMUM_LENGTH
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME='dim_customers';

SELECT
    COLUMN_NAME,
    DATA_TYPE,
    IS_NULLABLE,
    CHARACTER_MAXIMUM_LENGTH
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME='dim_products';

SELECT
    COLUMN_NAME,
    DATA_TYPE,
    IS_NULLABLE,
    CHARACTER_MAXIMUM_LENGTH
FROM INFORMATION_SCHEMA.COLUMNS
WHERE TABLE_NAME='fact_sales';
```

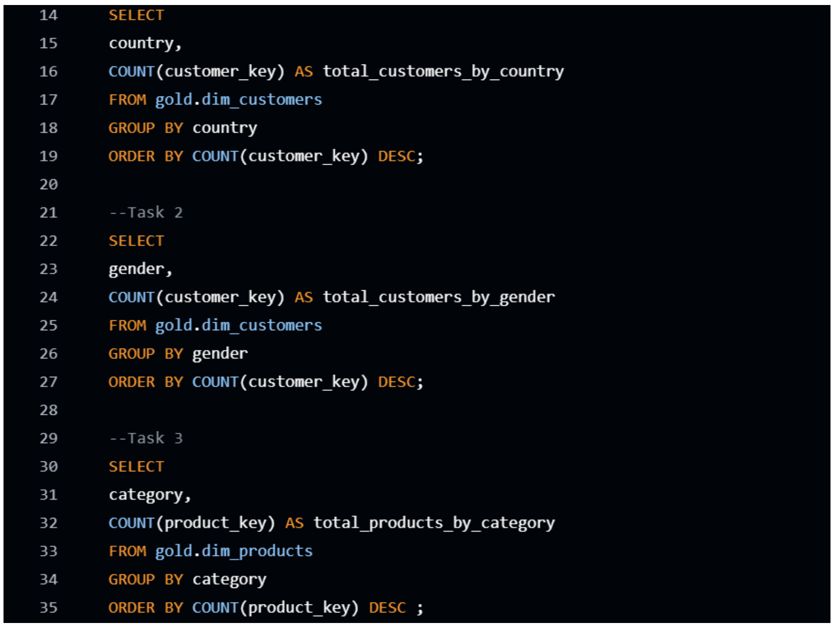
- Similarly performed the date exploration, dimension exploration,magnitude analysis, measures exploration, ranking analysis



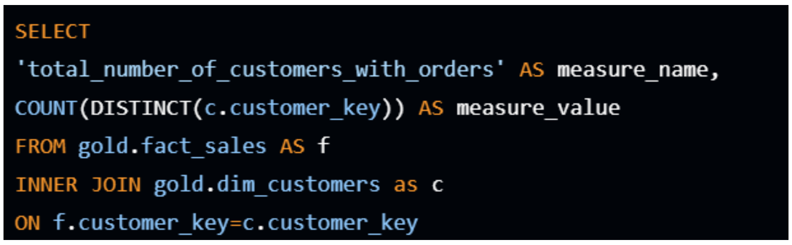
```sql
SELECT
MAX(order_date) AS last_order_date,
MIN(order_date) AS first_order_date,
DATEDIFF(YEAR,MIN(order_date),MAX(order_date)) AS order_range_in_years
FROM gold.fact_sales;

-- Task 2
SELECT
MIN(birthdate) AS oldest_customer,
DATEDIFF(YEAR,MIN(birthdate),GETDATE()) AS age_of_oldest_customer,
MAX(birthdate) AS youngest_customer,
DATEDIFF(YEAR,MAX(birthdate),GETDATE()) AS age_of_youngest_customer
FROM gold.dim_customers;
```



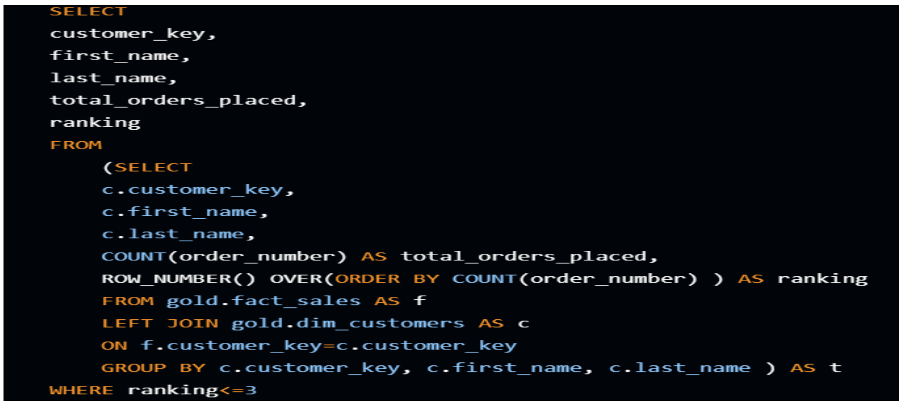
```sql
SELECT DISTINCT country
FROM gold.dim_customers;

--TASK 2
SELECT DISTINCT category,
subcategory,
product_name
FROM gold.dim_products
ORDER BY 1,2,3;
```



```sql
SELECT
country,
COUNT(customer_key) AS total_customers_by_country
FROM gold.dim_customers
GROUP BY country
ORDER BY COUNT(customer_key) DESC;

--Task 2
SELECT
gender,
COUNT(customer_key) AS total_customers_by_gender
FROM gold.dim_customers
GROUP BY gender
ORDER BY COUNT(customer_key) DESC;

--Task 3
SELECT
category,
COUNT(product_key) AS total_products_by_category
FROM gold.dim_products
GROUP BY category
ORDER BY COUNT(product_key) DESC ;
```

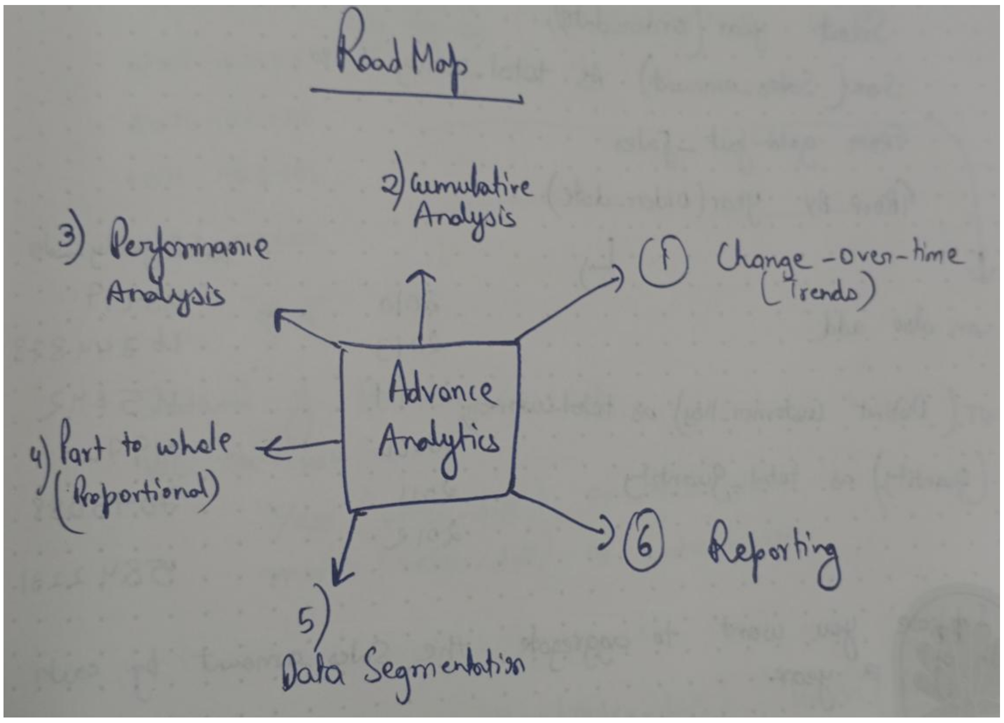


```
SELECT
'total_number_of_customers_with_orders' AS measure_name,
COUNT(DISTINCT(c.customer_key)) AS measure_value
FROM gold.fact_sales AS f
INNER JOIN gold.dim_customers as c
ON f.customer_key=c.customer_key
```

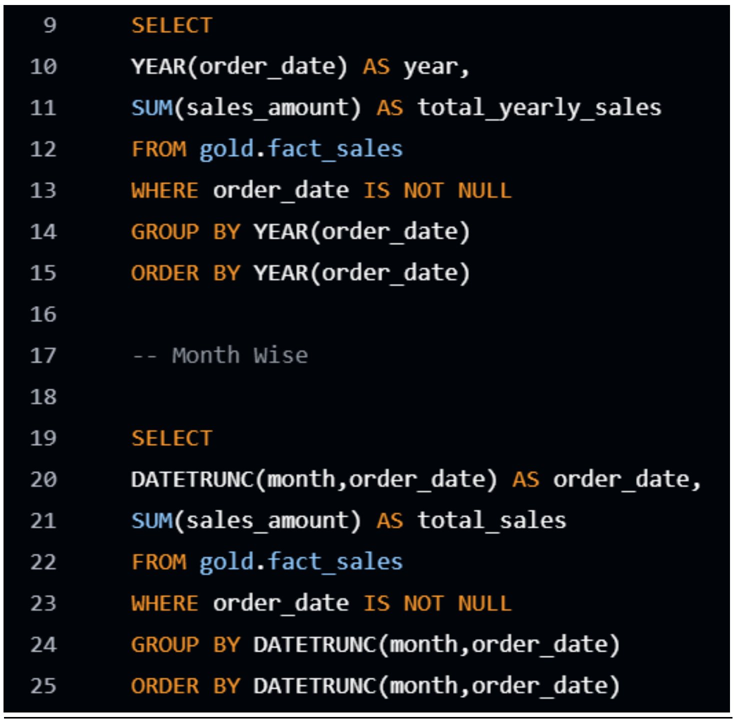


```
SELECT
customer_key,
first_name,
last_name,
total_orders_placed,
ranking
FROM
    (SELECT
    c.customer_key,
    c.first_name,
    c.last_name,
    COUNT(order_number) AS total_orders_placed,
    ROW_NUMBER() OVER(ORDER BY COUNT(order_number) ) AS ranking
    FROM gold.fact_sales AS f
    LEFT JOIN gold.dim_customers AS c
    ON f.customer_key=c.customer_key
    GROUP BY c.customer_key, c.first_name, c.last_name ) AS t
WHERE ranking<=3
```

6.5.2 Advance Analytics

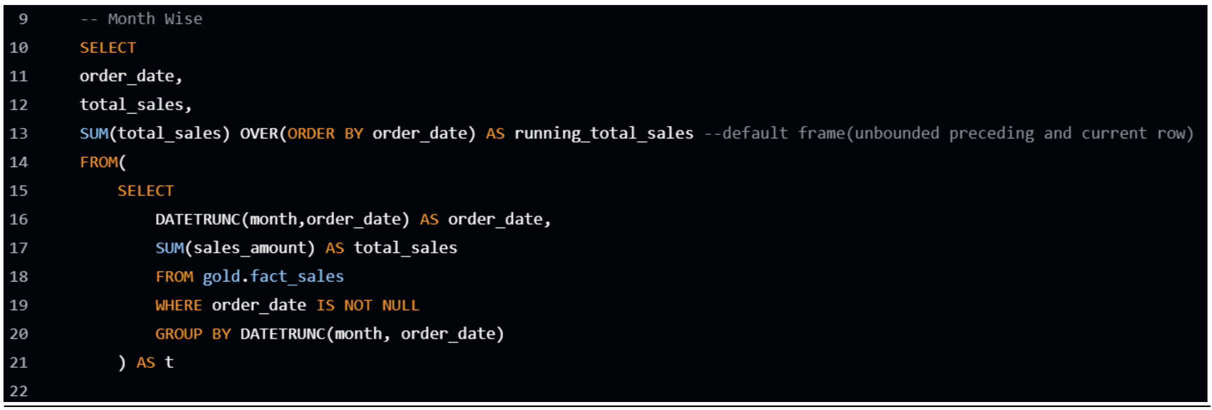


- Change over time Analysis



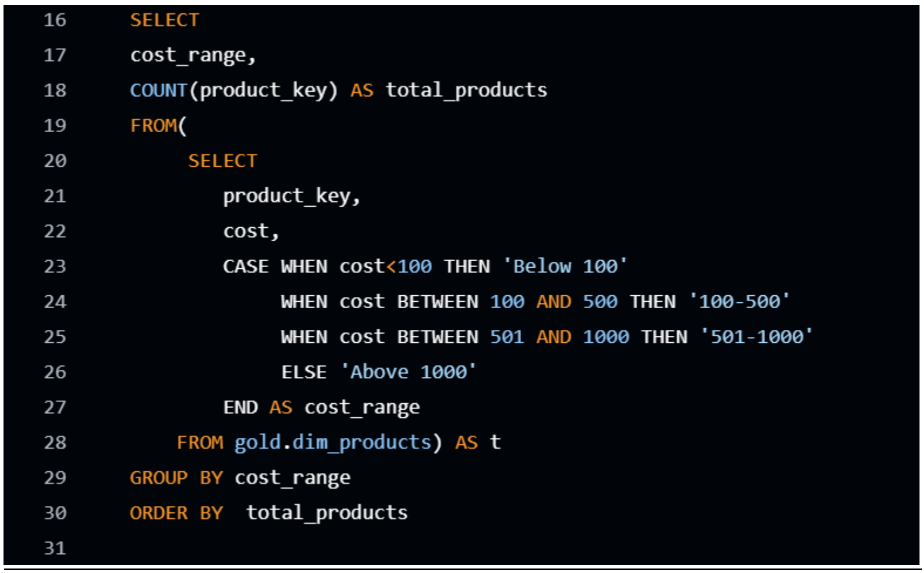
```sql
SELECT
YEAR(order_date) AS year,
SUM(sales_amount) AS total_yearly_sales
FROM gold.fact_sales
WHERE order_date IS NOT NULL
GROUP BY YEAR(order_date)
ORDER BY YEAR(order_date)

-- Month Wise

SELECT
DATETRUNC(month,order_date) AS order_date,
SUM(sales_amount) AS total_sales
FROM gold.fact_sales
WHERE order_date IS NOT NULL
GROUP BY DATETRUNC(month,order_date)
ORDER BY DATETRUNC(month,order_date)
```

- Cumulative Analysis



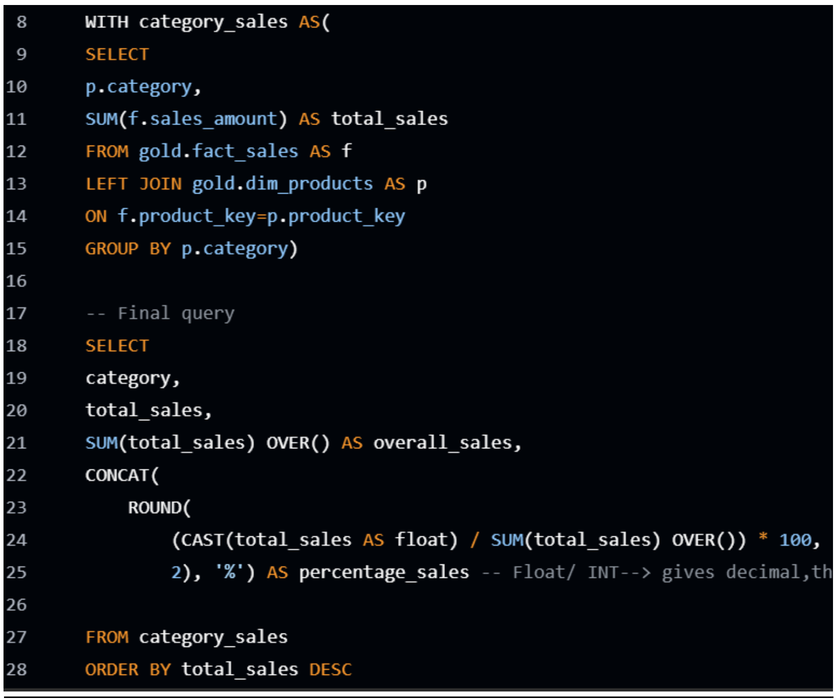
```sql
-- Month Wise
SELECT
order_date,
total_sales,
SUM(total_sales) OVER(ORDER BY order_date) AS running_total_sales --default frame(unbounded preceding and current row)
FROM(
    SELECT
        DATETRUNC(month,order_date) AS order_date,
        SUM(sales_amount) AS total_sales
        FROM gold.fact_sales
        WHERE order_date IS NOT NULL
        GROUP BY DATETRUNC(month, order_date)
    ) AS t

```

- Data Segmentation



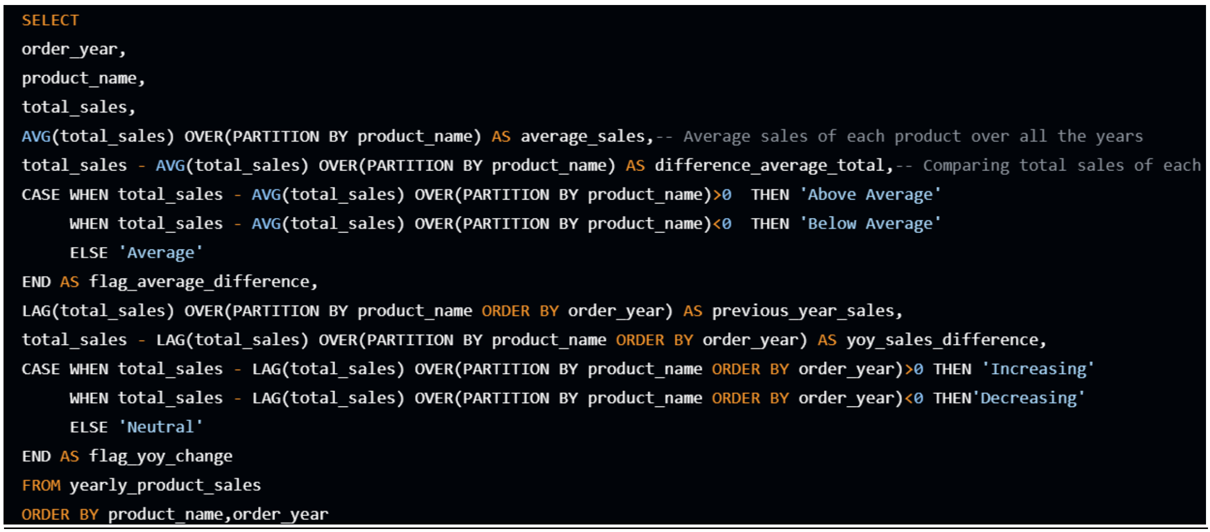
```
SELECT
cost_range,
COUNT(product_key) AS total_products
FROM(
     SELECT
        product_key,
        cost,
        CASE WHEN cost<100 THEN 'Below 100'
             WHEN cost BETWEEN 100 AND 500 THEN '100-500'
             WHEN cost BETWEEN 501 AND 1000 THEN '501-1000'
             ELSE 'Above 1000'
        END AS cost_range
    FROM gold.dim_products) AS t
GROUP BY cost_range
ORDER BY  total_products
```

- <u>Part to Whole Analysis</u>



```
WITH category_sales AS(
SELECT
p.category,
SUM(f.sales_amount) AS total_sales
FROM gold.fact_sales AS f
LEFT JOIN gold.dim_products AS p
ON f.product_key=p.product_key
GROUP BY p.category)

-- Final query
SELECT
category,
total_sales,
SUM(total_sales) OVER() AS overall_sales,
CONCAT(
    ROUND(
        (CAST(total_sales AS float) / SUM(total_sales) OVER()) * 100,
        2), '%') AS percentage_sales -- Float/ INT--> gives decimal,th

FROM category_sales
ORDER BY total_sales DESC
```

- Performance Analysis



```
SELECT
order_year,
product_name,
total_sales,
AVG(total_sales) OVER(PARTITION BY product_name) AS average_sales,-- Average sales of each product over all the years
total_sales - AVG(total_sales) OVER(PARTITION BY product_name) AS difference_average_total,-- Comparing total sales of each
CASE WHEN total_sales - AVG(total_sales) OVER(PARTITION BY product_name)>0  THEN 'Above Average'
     WHEN total_sales - AVG(total_sales) OVER(PARTITION BY product_name)<0  THEN 'Below Average'
     ELSE 'Average'
END AS flag_average_difference,
LAG(total_sales) OVER(PARTITION BY product_name ORDER BY order_year) AS previous_year_sales,
total_sales - LAG(total_sales) OVER(PARTITION BY product_name ORDER BY order_year) AS yoy_sales_difference,
CASE WHEN total_sales - LAG(total_sales) OVER(PARTITION BY product_name ORDER BY order_year)>0 THEN 'Increasing'
     WHEN total_sales - LAG(total_sales) OVER(PARTITION BY product_name ORDER BY order_year)<0 THEN'Decreasing'
     ELSE 'Neutral'
END AS flag_yoy_change
FROM yearly_product_sales
ORDER BY product_name,order_year
```

CHAPTER 7

RESULTS AND DISCUSSIONS

This chapter highlights the key outcomes of the Data Warehouse and Analytics Project and provides a detailed discussion of its effectiveness

7.1 Results Achieved

1) Successful Implementation of Medallion Architecture

2) Successful deployment of End-to-End ETL Workflow

- Developed a fully automated ETL pipeline that extracted, transformed, and loaded data without manual intervention
- Ensured referential integrity and consistency using surrogate keys in all dimension tables(customer_key, product_key)

3) Data Quality Improvements

- Cleaned and standardized customer and product data to remove duplicates and inconsistencies.
- Introduced audit fields such as "modified_date" for better traceability of data changes.

4) Performed Analytics and Extracted Insights about:

- Customer Analysis
- Products Insights
- Sales Trends

Overall, this project serves as a strong example of how modern data warehousing approaches can solve real-world business problems, improve decision-making, and deliver measurable value.

CHAPTER 8

CONCLUSION AND FUTURE SCOPE

8.1 Conclusion

This project is an effective demonstration of the design and implementation of a modern data warehousing and analytics system based on the Medallion Architecture, which includes Bronze, Silver, and Gold levels. By bringing together data from many source systems, in this case, ERP and CRM systems, in an optimized SQL Server data warehouse, the project created a scalable and sustainable data pipeline.

The ETL process was carefully designed keeping in view the data quality, consistency, and reliability. The automated data loading reduced the processing time and optimized the work flow. The Star Schema data architecture reduced the analytical queries and optimized the reporting performance, and the analysis done on top of gold layer presented concise, actionable insights of the customer behavior, product's performance, as well as the sales trends to the stakeholders.

The project highlights the importance of well-structured documentation and visualization tools, such as Notion for tracking workflows and Draw.io in building out the schema diagrams, in order to increase maintainability and aid stakeholder understanding. Overall, the solution proposed is a strong foundation in data engineering, data modelling, and analytics, that closely aligns with standard industry best approaches.

8.2 Future Scope

While this project provides a complete end-to-end solution, there are several areas where the system can be expanded and improved to meet enterprise-level requirements:

- Real-Time Data Streaming using Apache Kafka instead of batch uploads from CSV files
- Advanced Analytics using Machine Learning - Use predictive analytics or machine learning models to forecast sales trends, predict customer churn, or recommend products.

- Data Quality Automation- Implement automated data quality checks and anomaly detection to further improve reliability.
- Cloud Migration- Move the data warehouse to a cloud platform like Azure Synapse, Snowflake, or Databricks for better scalability, security, and integration with modern BI tools.

Upon implementation of these future improvements, the project could transform into a fully-fledged data platform, to support massive-scale real-time analytics and thus further supporting operative-level strategic decision-making.

REFERENCES

1. Kimball, R., & Ross, M. (2013). *The Data Warehouse Toolkit: The Definitive Guide to Dimensional Modeling* (3rd ed.). Wiley.
2. Inmon, W. H. (2005). *Building the Data Warehouse* (4th ed.). Wiley Publishing.
3. Microsoft. (2025). *SQL Server Documentation*. Retrieved from https://learn.microsoft.com/en-us/sql
4. Databricks. (2025). *Medallion Architecture Documentation*. Retrieved from https://docs.databricks.com
5. Draw.io. (2025). *Diagramming and Visualization Platform*. Retrieved from https://www.drawio.com
6. Notion Labs Inc. (2025). *Notion Documentation*. Retrieved from https://www.notion.so/help
7. Kimball Group. (2016). *Dimensional Modeling Techniques*. Retrieved from https://kimballgroup.com
8. Hernandez, M. J. (2013). *Database Design for Mere Mortals: A Hands-On Guide to Relational Database Design* (3rd ed.). Addison-Wesley.
9. Singh, A. (2024). *Best Practices for Building ETL Pipelines and Incremental Loading in SQL Server*. Journal of Data Engineering Practices, 18(4), 210–220.
10. Salkini, B. K. (2024). Data with Baraa: Modern Data Warehouse Design and Implementation in SQL Server. Retrieved from https://datawithbaraa.com