Data Warehouse Project Report

Project Title: SQL DataWarehouse

Author: N Lokesh Reddy

Date: 1-11-2025

Abstract

This report documents the design, implementation, and initial results of a small-scale data warehouse project. The repository contains sample source datasets (CRM and ERP), SQL scripts for reporting, and example views built in a layered architecture (bronze/silver/gold). The goal of the project is to consolidate sales and product/customer data from multiple sources into a single analytical store and provide reusable reporting views.

---

1. Introduction

Modern analytics require integrated, clean, and historically-aware data. This project demonstrates a basic Data Warehouse implementation combining CRM and ERP sales data into consolidated reporting tables/views.

2. Objectives

- Ingest source data from CRM and ERP CSV exports.
- Clean and standardize product and customer records.
- Create dimensional structures (products, customers, date) and a sales fact table.
- Provide analytical views for business reporting (e.g., customer-level KPIs, product performance).

3. Project Scope

This project focuses on a small demonstrative dataset located in datasets/ and contains SQL scripts in scripts/. It is **not** a production-grade pipeline but illustrates standard DW patterns: staging, dimensional modelling, and report creation.

4. Data Sources

The extracted repository contains the following source files (sample list):

- datasets/source_crm/cust_info.csv
- datasets/source_crm/prd_info.csv
- datasets/source_crm/sales_details.csv
- datasets/source_erp/cust_info.csv

- datasets/source_erp/prd_info.csv
- datasets/source_erp/sales_details.csv

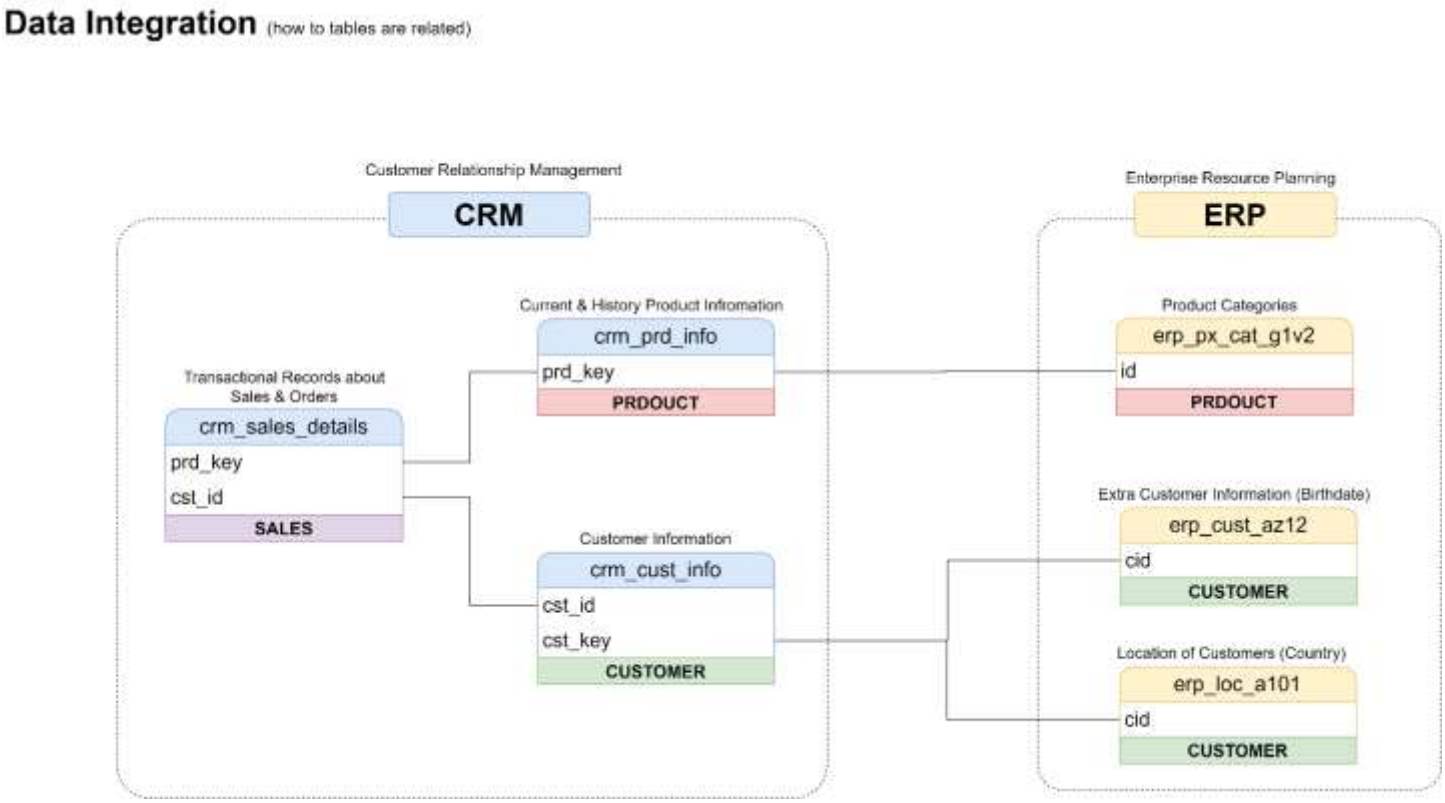

Figure 1 — Data integration map: how CRM tables and ERP tables relate to final dimensions.

These CSV files represent customer master, product master, and transactional sales data from two systems (CRM and ERP).

5. Architecture and Design

The repository follows a layered approach often described as bronze/silver/gold or staging/dim/fact:

- **Staging/Bronze:** Raw CSV loads into staging tables without heavy transformation.
- **Silver (Cleansed):** Standardization and deduplication of master data, surrogate key assignment.
- **Gold (Analytics):** Aggregated views and reporting tables used by analysts.

This is consistent with the available SQL views located under scripts/Reports/ which create gold.report_customers and gold.report_products views.

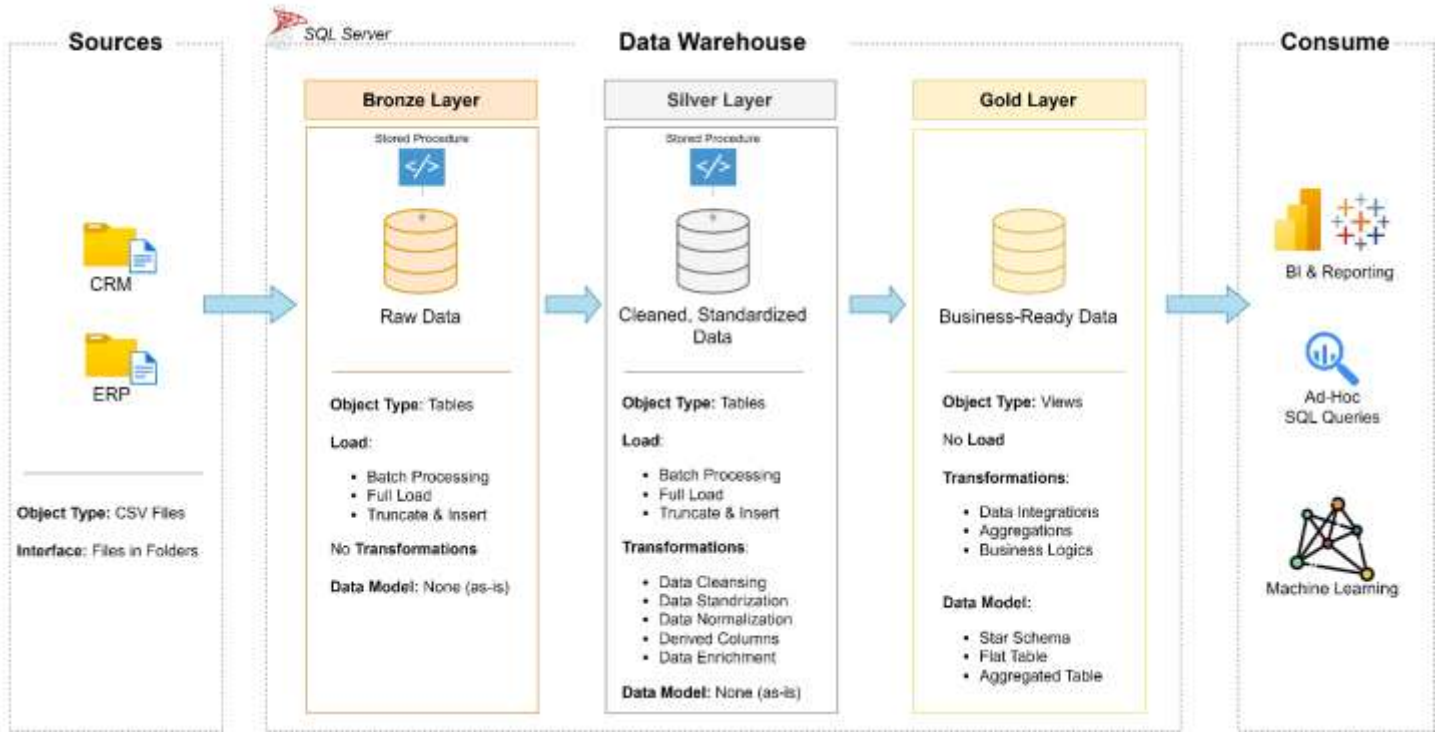

Figure 2 for the end-to-end architecture. The bronze layer preserves raw data for debugging, the silver layer performs cleanses and joins, and the gold layer exposes reporting views (for example, gold.dim_customers, gold.dim_products, gold.fact_sales).

6. ETL / Data Processing

Extraction

Data is exported as CSV from source systems and stored under datasets/.

Transformation

Typical transformations applied (to be adapted with real code where missing):

- Trim and normalize text fields (names, product codes).
- Convert date strings to proper DATE types.
- Join/merge CRM and ERP customer/product records using match rules.
- Compute surrogate keys for dimension tables.
- Handle slowly changing dimensions (SCD Type 2) if historical tracking is needed.

	Bronze Layer	Silver Layer	Gold Layer
Definition	Raw, unprocessed data as-is from sources	Clean & standardized data	Business-Ready data
Objective	Traceability & Debugging	(Intermediate Layer) Prepare Data for Analysis	Provide data to be consumed for reporting & Analytics
Object Type	Tables	Tables	Views
Load Method	Full Load (Truncate & Insert)	Full Load (Truncate & Insert)	None
Data Transformation	None (as-is)	- Data Cleaning - Data Standardization - Data Normalization - Derived Columns - Data Enrichment	- Data Integration - Data Aggregation - Business Logic & Rules
Data Modeling	None (as-is)	None (as-is)	- Start Schema - Aggregated Objects - Flat Tables
Target Audience	- Data Engineers	- Data Analysts - Data Engineers	- Data Analysts - Business Users

Figure 3 — Bronze / Silver / Gold comparison table.

7. Data Warehouse Schema

A common star schema for this project would include:

- **dim_product (product_key PK, product_id, product_name, product_line, cost, start_dt, end_dt, current_flag)**
- **dim_customer (customer_key PK, customer_id, name, region, segment, start_dt, end_dt, current_flag)**
- **dim_date (date_key PK, date, year, quarter, month, day)**
- **fact_sales (sales_key PK, order_number, date_key, product_key, customer_key, quantity, unit_price, total_amount)**

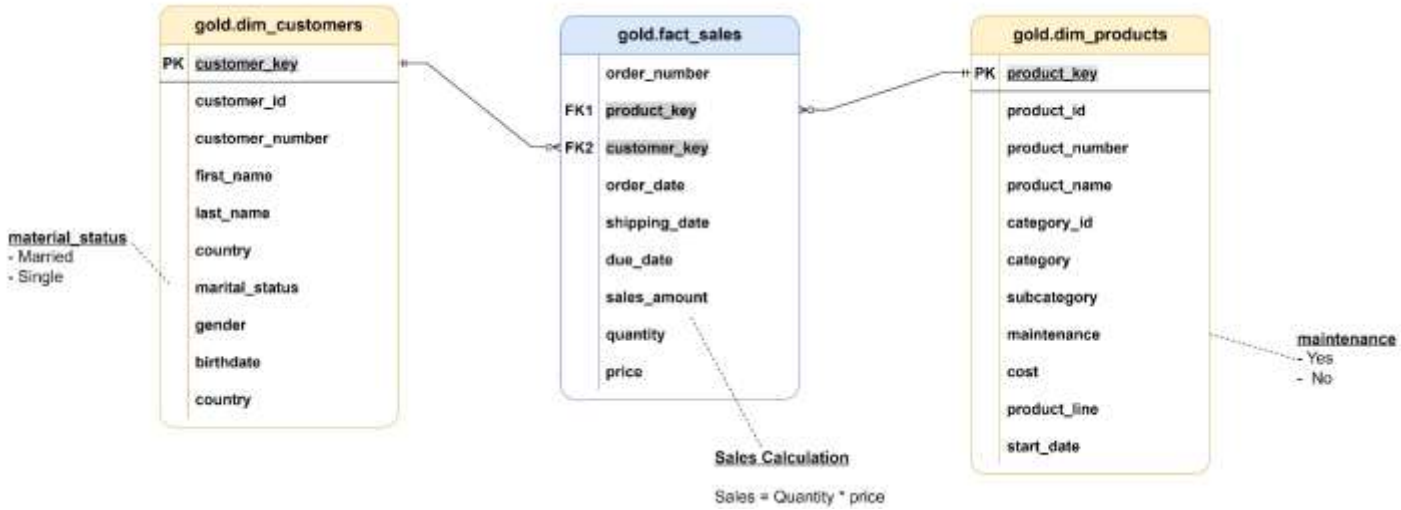

Figure 4: Logical Data Model (ER diagram) after this schema listing.

The repository's reporting SQL references fields like order_number, product_key, and aggregated metrics (total_sales, total_orders, avg_order_revenue), which aligns with the star schema above.

8. Reporting and Views

The repository includes reporting SQL under scripts/Reports/Report.sql. Two main analytics views are defined:

- gold.report_customers — customer-level KPI aggregations (total sales, total orders, last order date, average order revenue, months active).
- gold.report_products — product-level aggregations and lifecycle metrics.

An excerpt from the report SQL (for illustration):

```
CREATE VIEW gold.report_customers AS

WITH base_query AS (

  SELECT f.order_number, f.product_key, ...

)

-- further aggregations to compute total_sales, total_orders, avg_order_revenue--
```

(Full SQL lives in scripts/Reports/Report.sql.)

Data Flow (data Lineage)

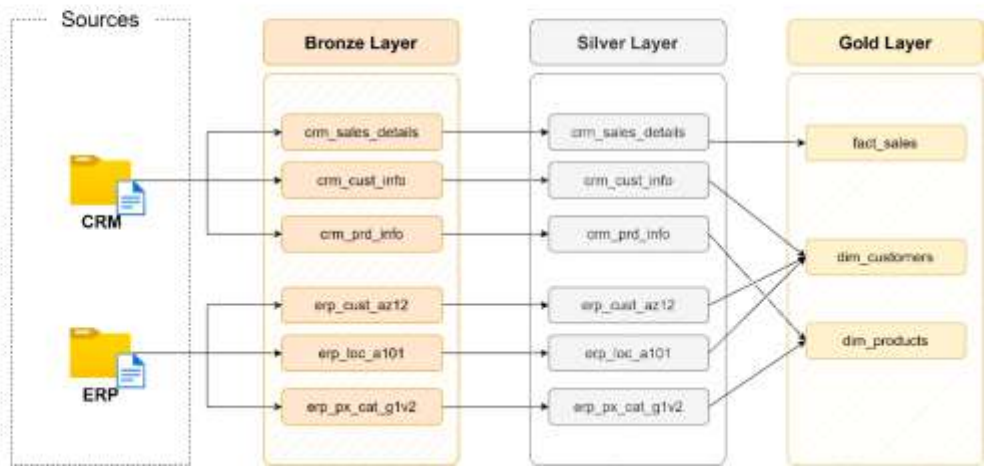

Figure 5 illustrates how crm_sales_details and the ERP tables are loaded into bronze tables and progressively transformed and merged into gold objects like gold.fact_sales, gold.dim_customers and gold.dim_products.

9. Results and Examples

At present, the repository provides the report views as the output artifacts. To demonstrate results we can:

- Populate the staging tables from the CSVs and run the transformations and view creation.
- Run sample analytical queries (top customers, product trends, monthly revenue pivot).
- Produce charts (monthly revenue, product sales distribution) and include them in the final report.

If you want, I can run quick data sampling and produce example charts and tables — tell me which outputs you prefer (tables, charts, PowerPoint slides, PDF report).

10.GITHUB REPOSITORY

https://github.com/lokeshreddynakkala/SQL-DataWarehouse/tree/main