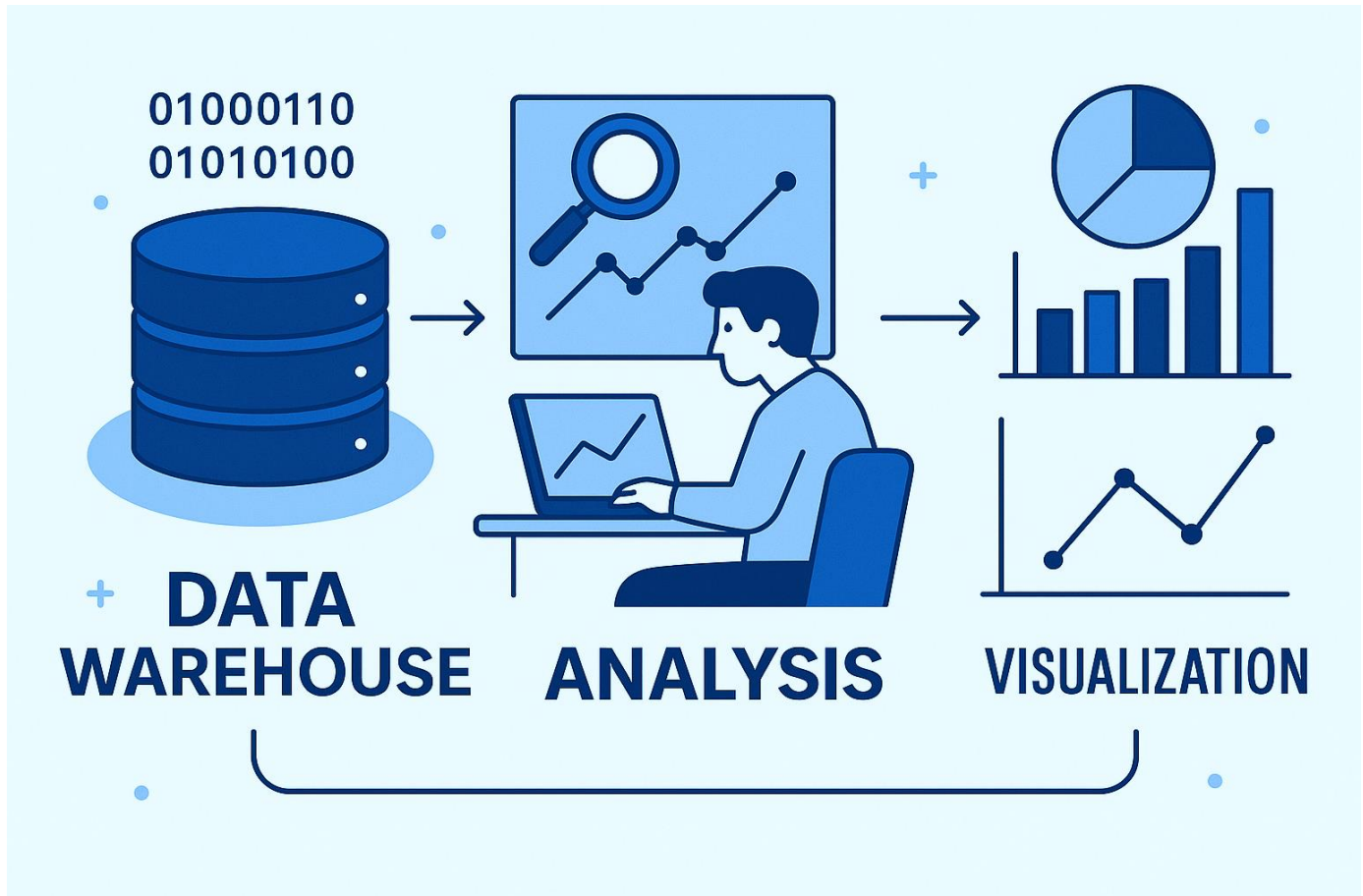


Data Warehouse Project



Made By

Mo'taz Gamal Ali

June 2025

GITHUB REPO: https://github.com/Motaz-gamal-77/SQL_Data-Warehouse_Project/tree/main

Table of Contents

1. Introduction
2. Project Overview
3. Data Architecture
4. Data Flow
5. Bronze Layer
6. Silver Layer
7. Gold Layer
8. Star Schema
9. Visualizations

1. Introduction

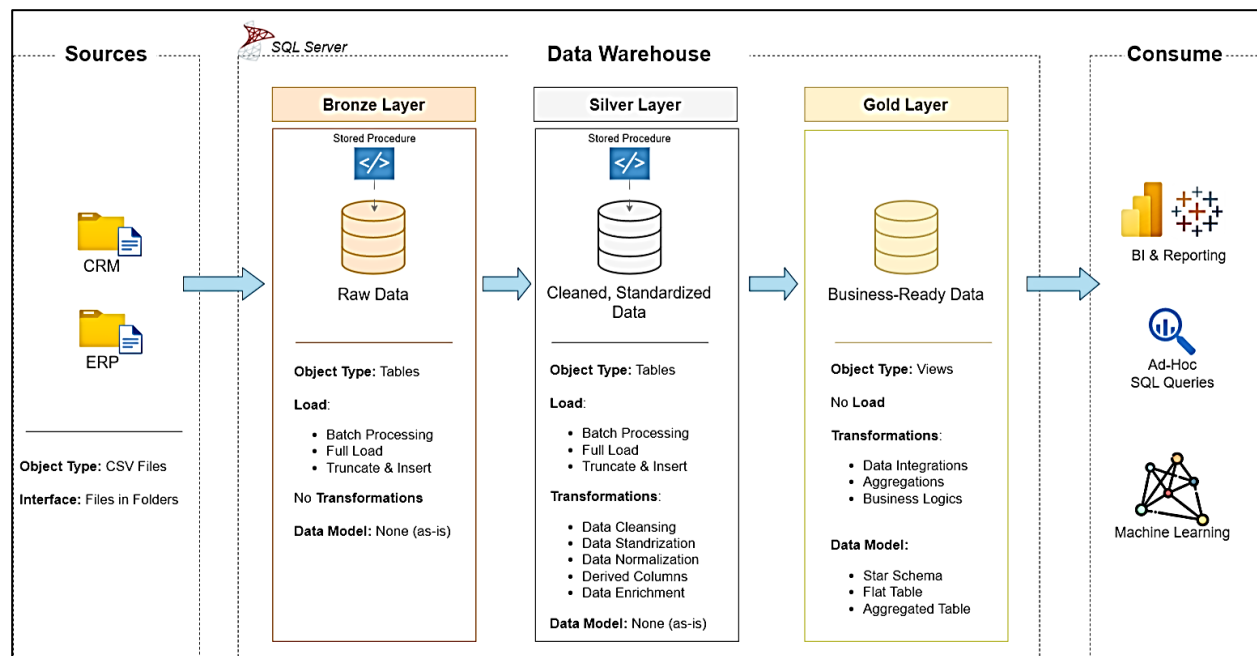
This project demonstrates a comprehensive data warehousing and analytics solution, from building a data warehouse to generating actionable insights. It highlights industry best practices in data engineering and analytics. The data was sourced from a structured SQL Server database and analyzed using Python (Pandas, Seaborn, Matplotlib) within a Jupyter Notebook environment.

2. Project Overview

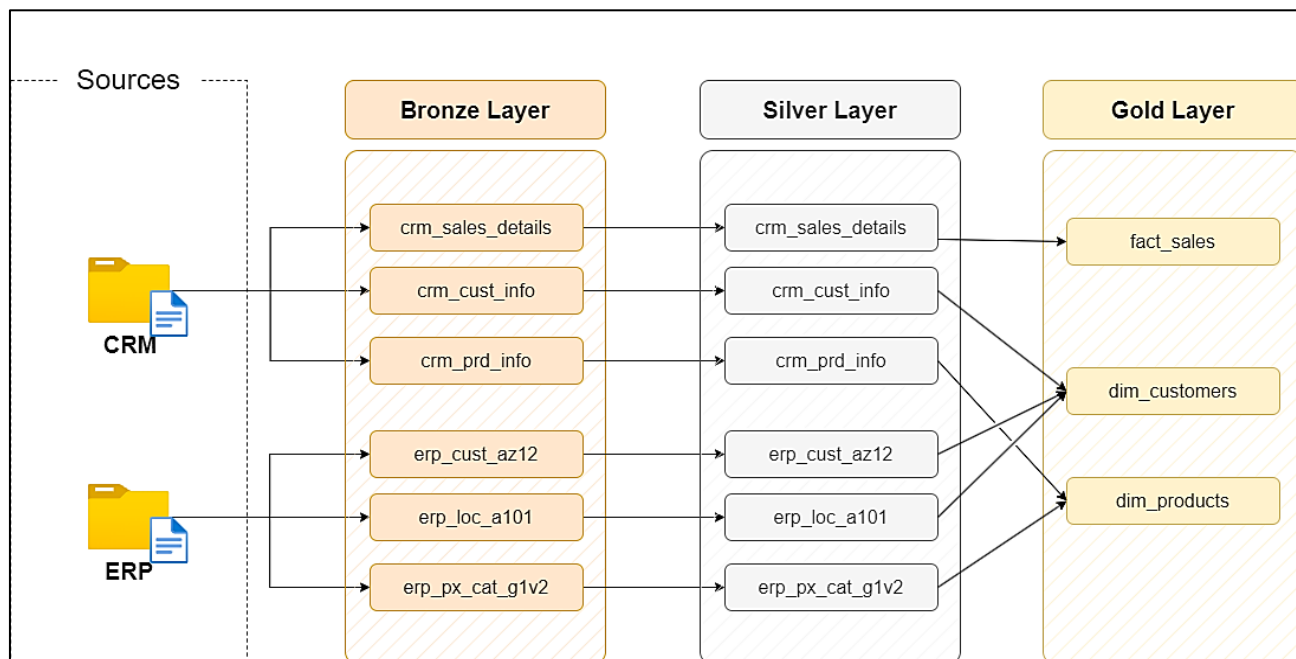
This project involves:

1. Data Architecture: Designing a Modern Data Warehouse Using Medallion Architecture Bronze, Silver, and Gold layers.
2. ETL Pipelines: Extracting, transforming, and loading data from source systems into the warehouse.
3. Data Modeling: Developing fact and dimension tables optimized for analytical queries.
4. Analytics & Reporting: Creating SQL-based visualizations and charts for actionable insights.

3. Data Architecture



4. Data Flow



5. Bronze Layer

Stores raw data as-is from the source systems. Data is ingested from CSV Files into SQL Server Database.

- Creating bronze layer tables

```
CREATE TABLE bronze.crm_cust_info(  
    cst_id INT,  
    cst_key NVARCHAR(50),  
    cst_firstname NVARCHAR(50),  
    cst_lastname NVARCHAR(50),  
    cst_marital_status NVARCHAR(50),  
    cst_gndr NVARCHAR(50),  
    cst_create_date DATE  
);  
GO  
  
CREATE TABLE bronze.crm_prd_info (  
    prd_id INT,  
    prd_key NVARCHAR(50),  
    prd_nm NVARCHAR(50),  
    prd_cost INT,  
    prd_line NVARCHAR(50),  
    prd_start_dt DATETIME,  
    prd_end_dt DATETIME  
);  
GO  
  
CREATE TABLE bronze.crm_sales_details (  
    sls_ord_num NVARCHAR(50),  
    sls_prd_key NVARCHAR(50),  
    sls_cust_id INT,  
    sls_order_dt INT,  
    sls_ship_dt INT,  
    sls_due_dt INT,  
    sls_sales INT,  
    sls_quantity INT,  
    sls_price INT  
);
```

- Loading data into bronze layer

```
CREATE OR ALTER PROC bronze.load_bronze AS  
BEGIN  
  
    TRUNCATE TABLE bronze.crm_cust_info  
    BULK INSERT bronze.crm_cust_info  
    FROM 'C:\Users\motaz\Desktop\DW Project\sql-data-warehouse-project\datasets\source_crm\cust_info.csv'  
    WITH (  
        FIRSTROW = 2,  
        FIELDTERMINATOR = ',',  
        TABLOCK);  
  
    TRUNCATE TABLE bronze.crm_prd_info  
    BULK INSERT bronze.crm_prd_info  
    FROM 'C:\Users\motaz\Desktop\DW Project\sql-data-warehouse-project\datasets\source_crm\prd_info.csv'  
    WITH (  
        FIRSTROW = 2,  
        FIELDTERMINATOR = ',',  
        TABLOCK);  
  
    TRUNCATE TABLE bronze.crm_sales_details  
    BULK INSERT bronze.crm_sales_details  
    FROM 'C:\Users\motaz\Desktop\DW Project\sql-data-warehouse-project\datasets\source_crm\sales_details.csv'  
    WITH (  
        FIRSTROW = 2,  
        FIELDTERMINATOR = ',',  
        TABLOCK);
```

6. Silver Layer

This layer includes data cleansing, standardization, and normalization processes to prepare data for analysis.

- Remove unwanted spaces

```
SELECT
cst_id,
cst_key,
TRIM(cst_firstname) AS cst_firstname,
TRIM(cst_lastname) AS cst_lastname,
```

- Data Normalization & handling missing values

```
CASE
WHEN UPPER(TRIM(cst_marital_status)) = 'S' THEN 'Single'
WHEN UPPER(TRIM(cst_marital_status)) = 'M' THEN 'Married'
ELSE 'n/a'
END AS cst_marital_status,
CASE
WHEN UPPER(TRIM(cst_gndr)) = 'F' THEN 'Female'
WHEN UPPER(TRIM(cst_gndr)) = 'M' THEN 'Male'
ELSE 'n/a'
END AS cst_gndr,
cst_create_date
```

- Remove duplicates & filtering

```
FROM (
SELECT *,
ROW_NUMBER() OVER (PARTITION BY cst_id ORDER BY cst_create_date DESC) AS flag_last
FROM bronze.crm_cust_info
WHERE cst_id IS NOT NULL
) t
WHERE flag_last = 1;
```

- Data Enrichment & Casting

```
CAST(prd_start_dt AS DATE) AS prd_start_dt,
CAST(LEAD(prd_start_dt) OVER (PARTITION BY prd_key ORDER BY prd_start_dt) -1 AS DATE) AS prd_end_dt
FROM bronze.crm_prd_info
```

- Derived Columns

```
SELECT
prd_id,
REPLACE(SUBSTRING(prd_key, 1, 5), '-', '_') AS cat_id,
(SUBSTRING(prd_key, 7, LEN(prd_key))) AS prd_id,
prd_nm,
```

7. Gold Layer

Houses business-ready data modeled into a star schema required for reporting and analytics.

- Customer Dimension

```
CREATE OR ALTER VIEW gold.dim_customer AS
SELECT
    ROW_NUMBER() OVER (ORDER BY ci.cst_id) AS customer_key,
    ci.cst_id AS customer_id,
    ci.cst_key AS customer_number,
    ci.cst_firstname AS first_name,
    ci.cst_lastname AS last_name,
    la.cntry AS country,
    ci.cst_marital_status AS marital_status,
    CASE WHEN ci.cst_gndr != 'n/a' THEN ci.cst_gndr
         ELSE COALESCE (ca.gen , 'n/a')
    END AS gender,
    ca.bdate AS birthdate,
    ci.cst_create_date AS create_date
FROM silver.crm_cust_info ci
LEFT JOIN silver.erp_cust_az12 ca
ON ci.cst_key = ca.cid
LEFT JOIN silver.erp_loc_a101 la
ON ci.cst_key = la.cid
GO
```

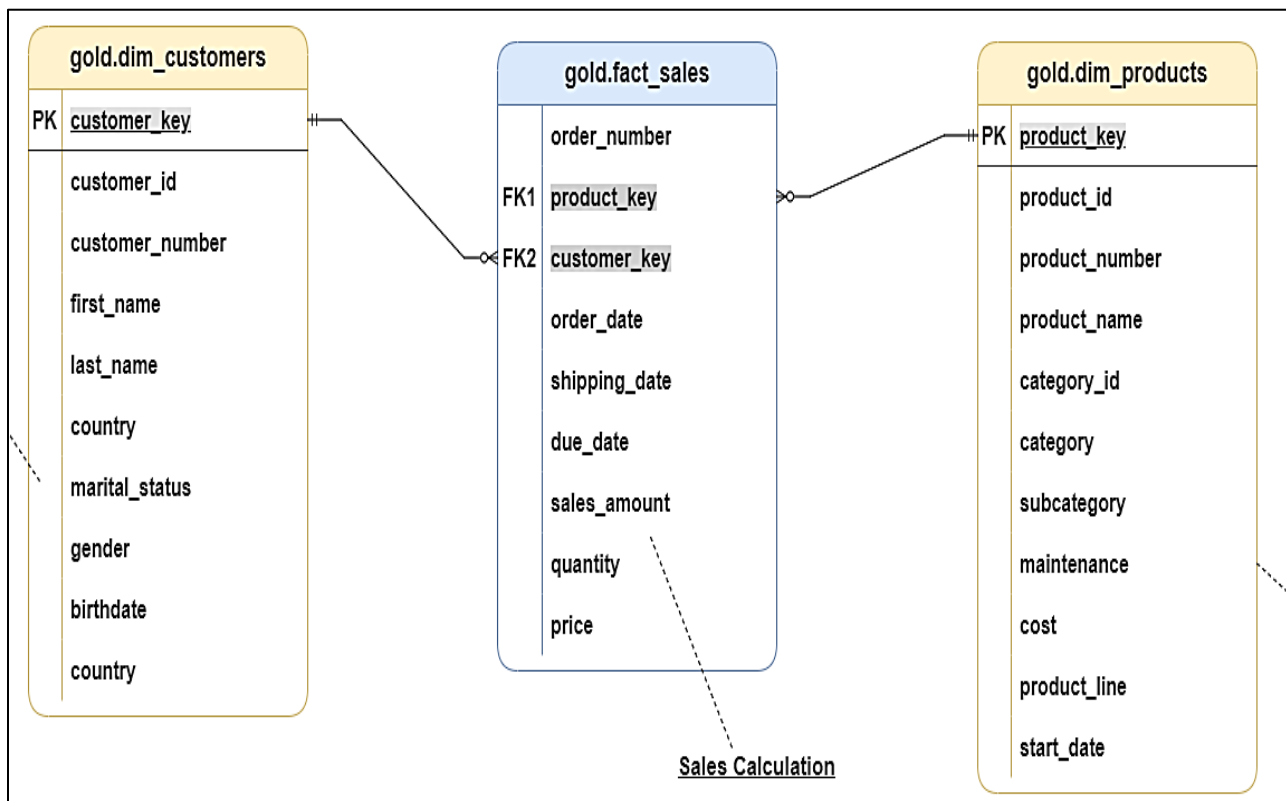
- Products Dimension

```
CREATE OR ALTER VIEW gold.dim_products AS
SELECT
    ROW_NUMBER() OVER (ORDER BY pc.prđ_start_dt , pc.prđ_key) AS product_key,
    pc.prđ_id AS product_id,
    pc.prđ_key AS product_number,
    pc.prđ_nm AS product_name,
    pc.cat_id AS category_id,
    pe.cat AS category,
    pe.subcat AS subcategory,
    pe.maintenance,
    pc.prđ_cost AS cost,
    pc.prđ_line AS Product_line,
    pc.prđ_start_dt AS start_date
FROM silver.crm_prđ_info pc
LEFT JOIN silver.erp_px_cat_g1v2 pe
ON pc.cat_id = pe.id
WHERE pc.prđ_end_dt IS NULL
```


- Sales Fact

```
CREATE OR ALTER VIEW gold.fact_sales AS
SELECT
    sls_ord_num AS order_number,
    pr.product_key AS product_key,
    c.customer_key AS customer_key,
    sls_order_dt AS order_date,
    sls_ship_dt AS ship_date,
    sls_due_dt AS due_date,
    sls_sales AS sales_amount,
    sls_quantity AS quantity,
    sls_price AS price
FROM silver.crm_sales_details sc
LEFT JOIN gold.dim_products pr
ON sc.sls_prd_key = pr.product_number
LEFT JOIN gold.dim_customer c
ON sc.sls_cust_id = c.customer_id
```

8. Star Schema



9. Visualizations

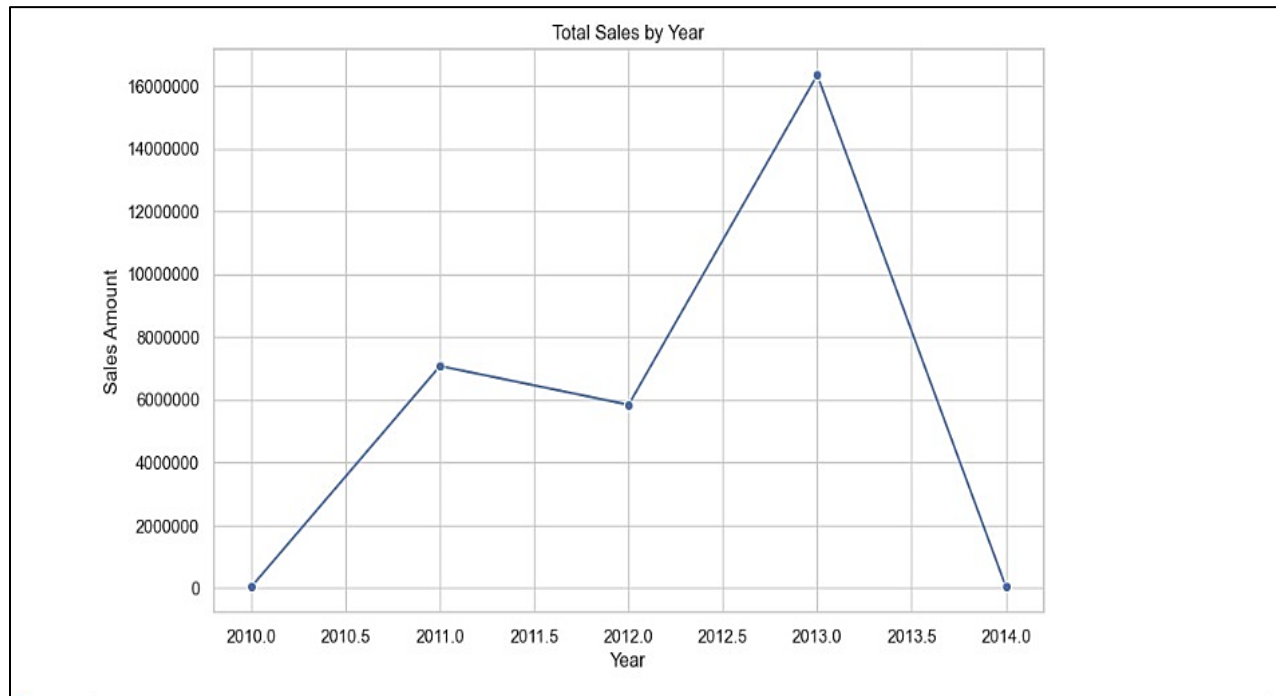


Figure 1: Total Sales Per Year

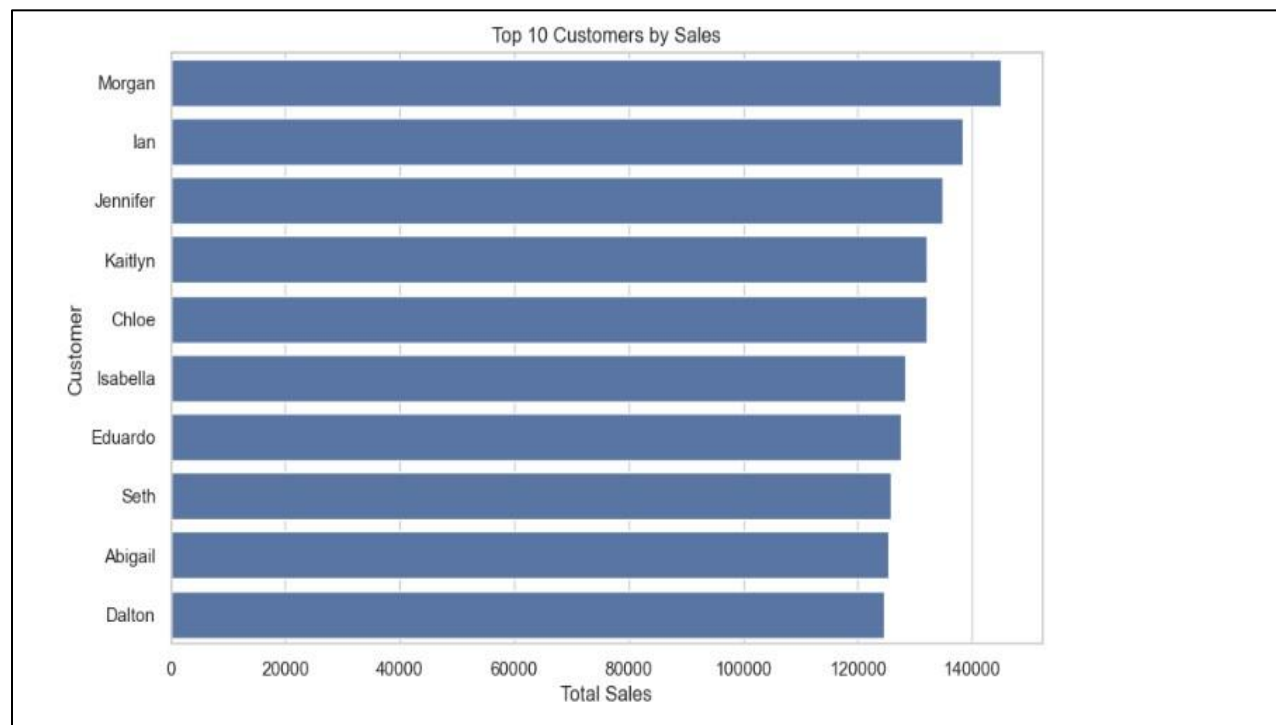


Figure 2: Top 10 Customer By Total Sales

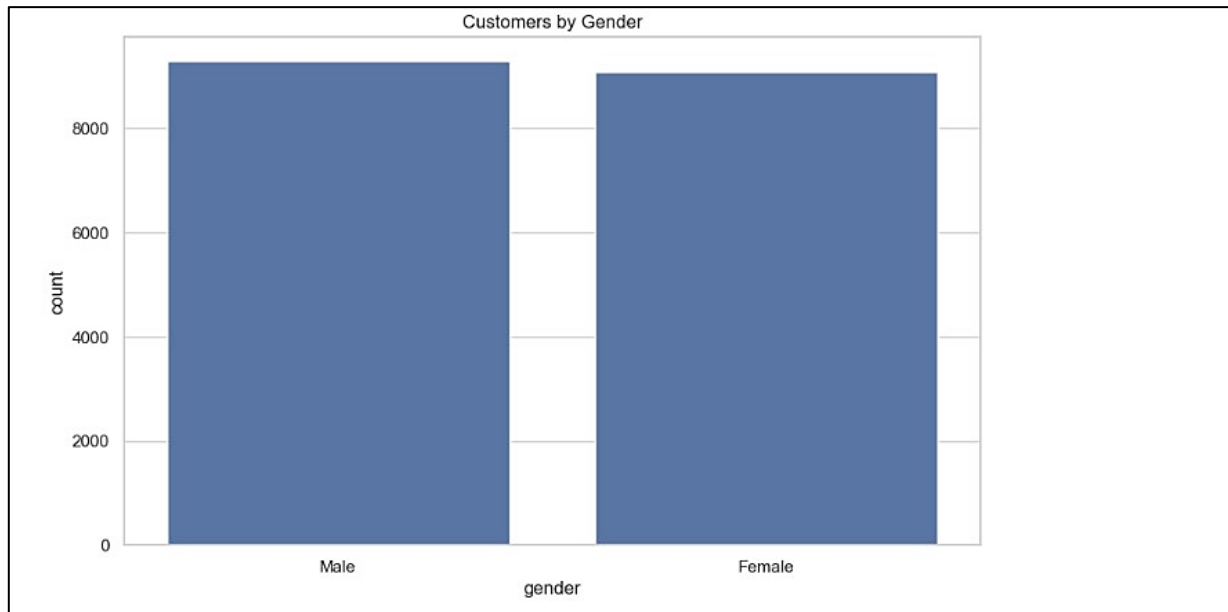


Figure 3: Customers Gender Distribution

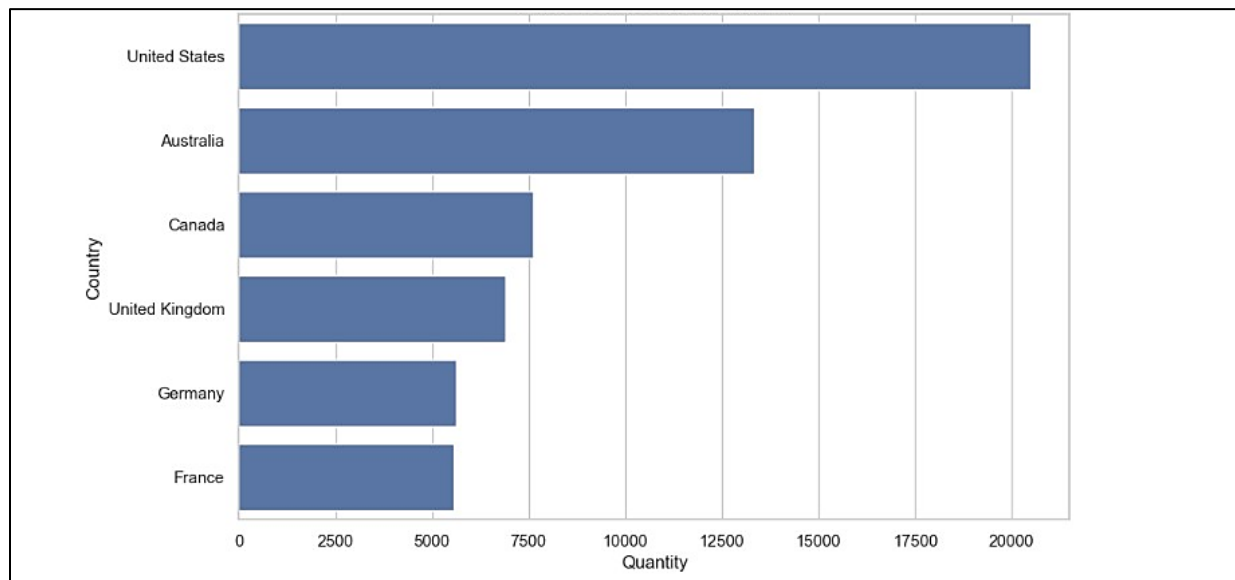


Figure 4: Top Countries by Quantity Sold

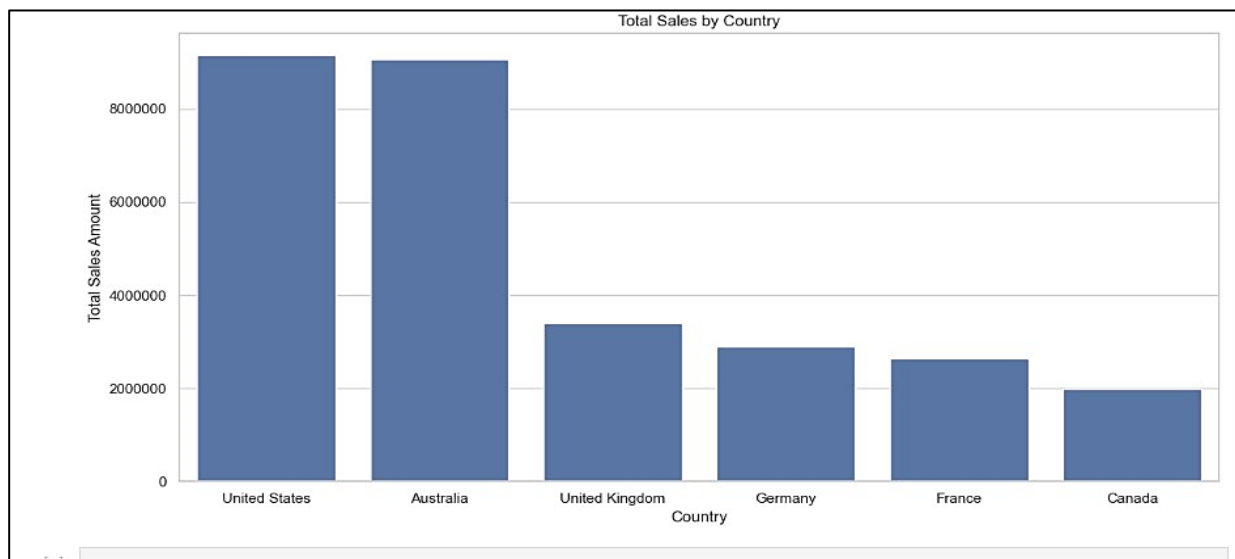


Figure 5: Top Countries by Sales Amount

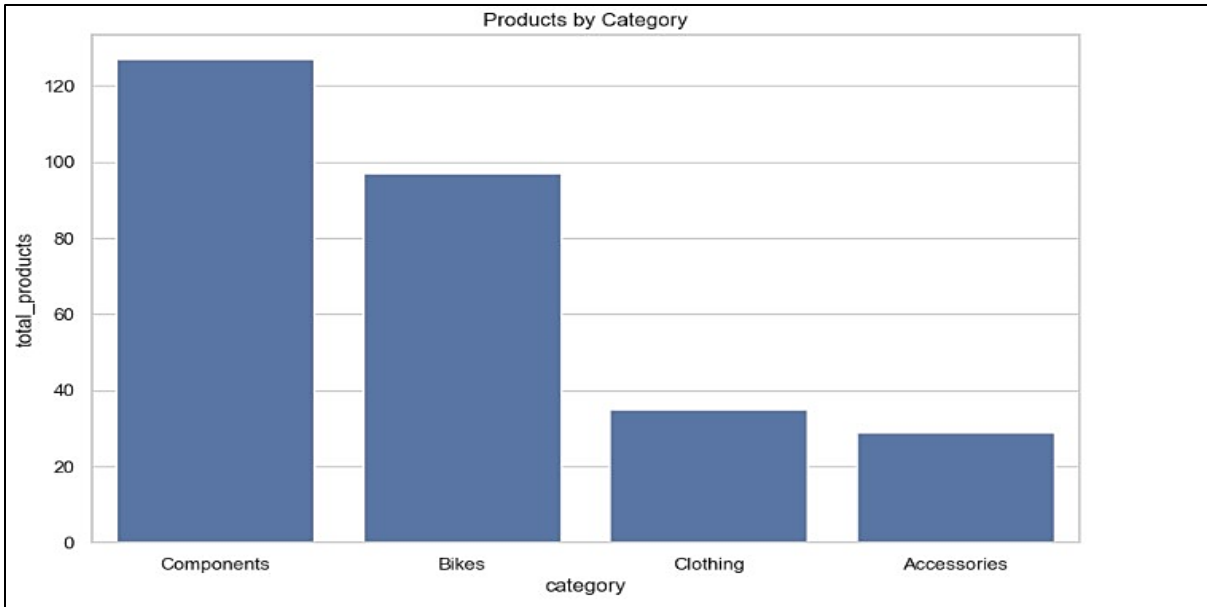


Figure 7: Total Products by Category

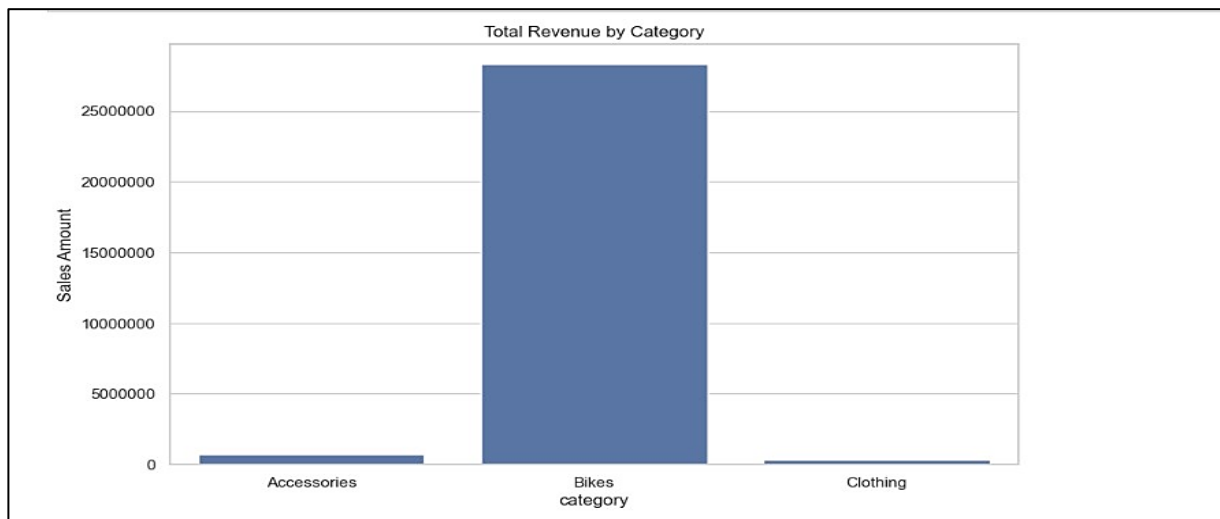


Figure 6: Total Sales by Category

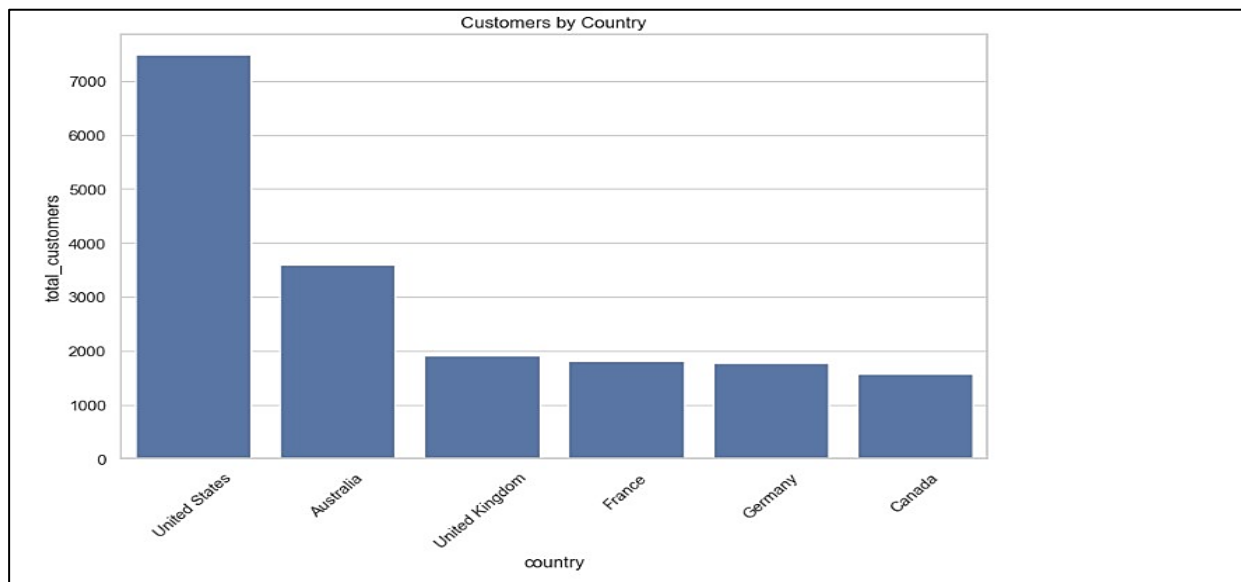


Figure 8: Total Customers by Country