Data Warehousing Project Report Steinway Sales Data Mart Implementation

Technical Report

Group 2 - 2DB

Francisco Silva (1230985)
Vasco Magolo (1231562)

January 2026

Instituto Superior de Engenharia do Porto
LETI-DSSMV ProjectDroid

Contents

1 Introduction

This report presents the development of a Data Mart for "Steinway", a company specializing in musical instruments. The project aims to overcome the analytical limitations of the current operational system by implementing a dimensional model capable of supporting historical analysis of sales data.

The solution follows the Kimball methodology and was implemented using SQL Server for the database and SQL Server Integration Services (SSIS) for the ETL (Extract, Transform, Load) processes.

2 Data Mart Architecture

The system architecture consists of three main layers, ensuring data integrity and decoupling the analytical environment from the operational source:

1. **Source System:** The operational database containing tables such as `Sales`, `SalesDetails`, `Customers`, and `Products`.
2. **Staging Area:** An intermediate area used to extract raw data, perform cleaning operations (DQP), and prepare data for loading. It includes specific error tables (e.g., `ProductDQP`, `SalesDQP`) to store rejected records.
3. **Data Mart:** The final destination modeled in a Star Schema, optimized for querying and reporting.

3 Dimensional Model

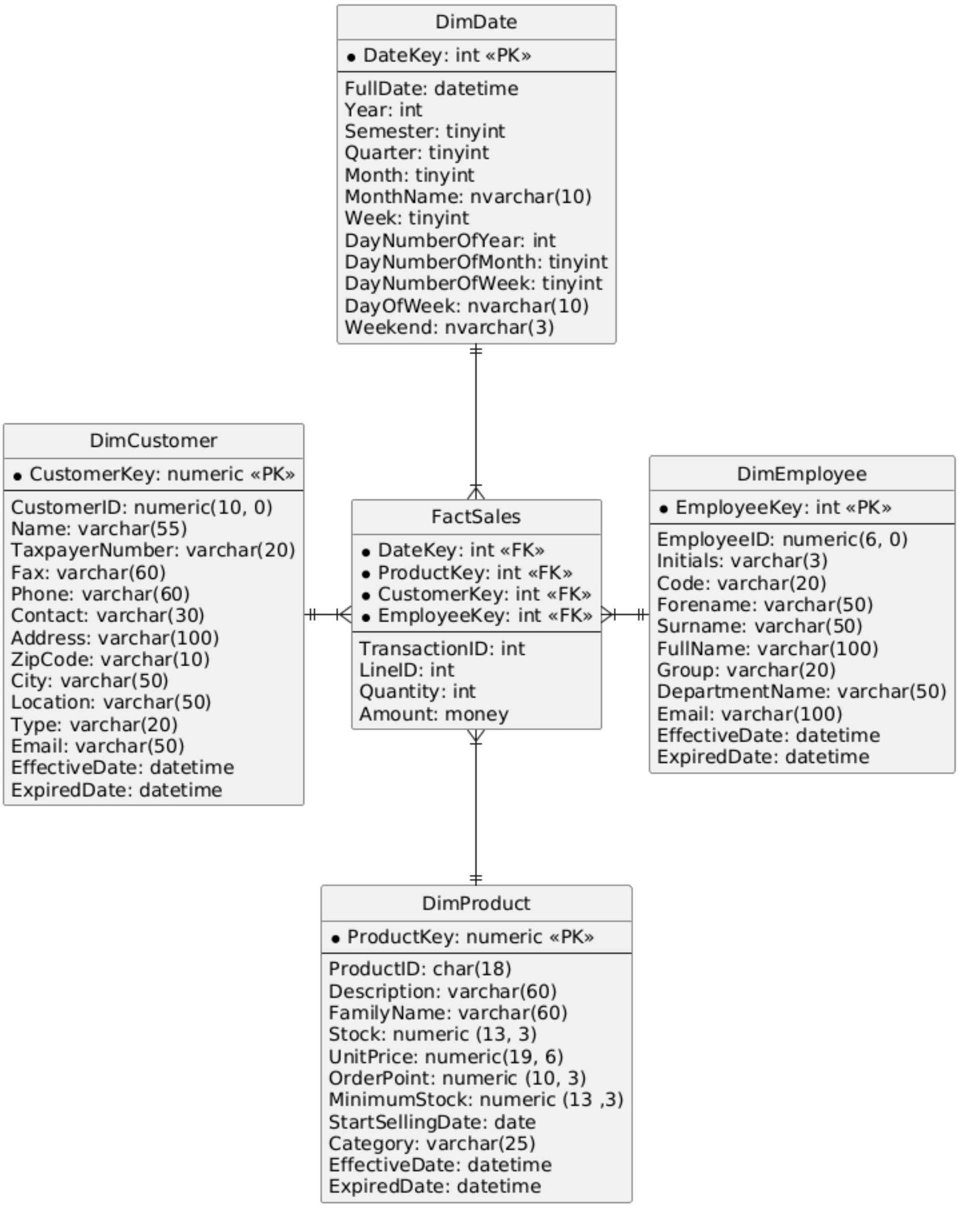
3.1 Business Process and Granularity

The chosen business process is **Sales**. The granularity was defined at the **transaction line item level**, meaning each row in the Fact table represents a specific item sold within a transaction. This allows for detailed analysis of product performance and customer behavior.

3.2 Star Schema Design

The model comprises one fact table and four dimension tables:

- **FactSale:** Centralizes quantitative metrics such as `Quantity`, `UnitPrice`, `LineID`, and calculated line totals. It links to dimensions via surrogate keys.
- **DimCustomer:** Consolidates *Customers* and *CustomerTypes*.
- **DimProduct:** Consolidates *Products* and *Families*.
- **DimEmployee:** Consolidates *Employees* and *Departments* data.
- **DimDate:** A calendar dimension generated to support temporal aggregation (Year, Quarter, Month).



DimDate
- DateKey: int «PK»

FullDate: datetime
Year: int
Semester: tinyint
Quarter: tinyint
Month: tinyint
MonthName: nvarchar(10)
Week: tinyint
DayNumberOfYear: int
DayNumberOfMonth: tinyint
DayNumberOfWeek: tinyint
DayOfWeek: nvarchar(10)
Weekend: nvarchar(3)

DimCustomer
- CustomerKey: numeric «PK»

CustomerID: numeric(10, 0)
Name: varchar(55)
TaxpayerNumber: varchar(20)
Fax: varchar(60)
Phone: varchar(60)
Contact: varchar(30)
Address: varchar(100)
ZipCode: varchar(10)
City: varchar(50)
Location: varchar(50)
Type: varchar(20)
Email: varchar(50)
EffectiveDate: datetime
ExpiredDate: datetime

FactSales
- DateKey: int «FK»
- ProductKey: int «FK»
- CustomerKey: int «FK»
- EmployeeKey: int «FK»

TransactionID: int
LineID: int
Quantity: int
Amount: money

DimEmployee
- EmployeeKey: int «PK»

EmployeeID: numeric(6, 0)
Initials: varchar(3)
Code: varchar(20)
Forename: varchar(50)
Surname: varchar(50)
FullName: varchar(100)
Group: varchar(20)
DepartmentName: varchar(50)
Email: varchar(100)
EffectiveDate: datetime
ExpiredDate: datetime

DimProduct
- ProductKey: numeric «PK»

ProductID: char(18)
Description: varchar(60)
FamilyName: varchar(60)
Stock: numeric (13, 3)
UnitPrice: numeric(19, 6)
OrderPoint: numeric (10, 3)
MinimumStock: numeric (13 ,3)
StartSellingDate: date
Category: varchar(25)
EffectiveDate: datetime
ExpiredDate: datetime

4 ETL Process and Data Cleaning

The ETL process is the core of this project, ensuring that only high-quality data reaches the Data Mart.

4.1 Extraction and Staging

Data is extracted from the OLE DB Source and loaded into Staging tables. During this phase, data types are standardized (e.g., converting Unicode to Non-Unicode where necessary) to prevent compatibility issues.

4.2 Transformation and Data Quality (DQP)

A key requirement was to filter records that violate domain rules or contain missing mandatory values. To achieve this efficiently, we implemented a **Sequential Derived Column** pattern in SSIS:

4.2.1 The "Accumulator" Strategy

Instead of splitting the data flow for every single error (which would result in complex branching), we utilized a variable column named `DQP`. This column passes through a chain of Derived Column transformations, accumulating error messages:

```
-- Step 1: Validate Columns
DQP = (InvalidDepartmentDescription ? "DepartmentDescription; " : "") + (InvalidSurname ? "Invalid Surname; " : "") + (InvalidForename ? "Invalid Forename; " : "") + (InvalidEmail ? "Invalid Email; " : "") + (InvalidInitials ? "Invalid Initials; " : "")

-- Step 2: Validate DQP Accumulator
LEN(DQP) >= 1
```

Listing 1: Example of SSIS Expression for Error Accumulation

This approach ensures that a single rejected record in the Staging Area contains a complete diagnosis of all its issues (e.g., "Invalid Address; Invalid Fax").

4.3 Loading and Error Handling

A **Conditional Split** component directs rows based on the `DQP`:

- **Clean Rows:** Proceed to the Data Mart dimensions or fact table.
- **Dirty Rows:** Redirected to DQP tables (e.g., `ProductDQP`) for auditing purposes, fulfilling the project's quality requirements.

5 Justification of Options

1. **Denormalized Dimensions:** We chose to combine Product and Families into a single `DimProduct` table to simplify queries for the end-user and improve read performance, characteristic of the Kimball approach.

2. **Sequential Error Handling:** Using a chain of Derived Columns is more efficient than multiple Conditional Splits because it avoids row duplication and multiple passes over the data, processing the entire validation logic in a single pipeline stream.

3. **Surrogate Keys:** All dimensions use `IDENTITY` integer keys (e.g., `CustomerKey`) instead of operational codes to isolate the Data Mart from changes in the source system production keys.

6 Conclusion and Future Improvements

This project resulted in the deployment of a fully functional ETL pipeline that transforms raw transactional data into an optimized analytical Data Mart. The solution features automated Staging Area creation and handles complex requirements, such as the historical tracking of product categories. Despite challenges related to source schema naming errors and strict SSIS data typing, the implementation of dynamic expressions ensured a resilient process. Ultimately, the system provides a stable foundation for querying sales metrics, supported by rigorous data quality checks that guarantee the integrity of the information.