Invoice-Based Dimensional Modelling Report (Stage 8)

Submitted by: Mildred Okonye

Slack ID: Mildred O

Date: 27th March 2025

Project: Invoice-Based Dimensional Modeling

Table of Contents

1. Introduction

Dimensional modeling is an essential method used for structuring data in a way that supports high-performance analytics. In the context of this assignment, a **dimensional model** has been designed based on a given invoice dataset to facilitate reporting and analysis for a retail business. This model aims to enhance business intelligence capabilities and optimize querying and reporting performance. This report provides a detailed explanation of the steps and strategies adopted, as well as the rationale behind the design decisions made throughout the process.

2. Objective

The primary objective of this assignment is to create a robust dimensional model that can support detailed analytical reporting, provide drill-down capabilities, maintain historical tracking, and scale effectively for future data growth. The report will focus on how the dimensional model addresses various business objectives and ensure that it is optimized for query performance and data consistency.

3. Data Profiling and Inference

The invoice dataset provided simulates a typical point-of-sale transactional system where each record represents a line item within an invoice. Each line contains key attributes related to a transaction, including details about the customer, product, time, and store. Profiling this data is essential for identifying which fields should be modeled as dimensions and which as facts.

1. Invoice_ID, Invoice_Date,
2. Customer_ID, Customer_Name,
3. Product_ID, Product_Name, Quantity, Unit_Price, Line_Total,
4. Store_ID.

From this schema, we can profile and infer the business logic behind each data field, identifying the foundational entities for analytical modeling.

3.1 Identification of Dimensions

-**Customer Dimension**:

This dimension tracks information about each unique customer.

Attributes include Customer_ID and Customer_Name.

Future enhancements could involve additional attributes such as email, phone number, and customer segments, which will allow for more detailed analysis of customer behavior.

-**Product Dimension**:

It contains Product_ID and Product_Name.

Attributes like category, brand, and product description can be inferred and included to enable effective grouping and hierarchical structuring of products.

These enhancements support deeper analysis and classification of products.

- **Time Dimension**:

The Time Dimension is derived from the Invoice_Date field.

It includes various time aggregations such as Hour, Day, Week, Month, Quarter, and Year.

This dimension enables powerful time-series analysis and supports drill-down capabilities across different time levels.

- **Store Dimension**:

Captured using Store_ID, this dimension represents store-level data.

Store_Name, Region, and Manager are possible additional attributes.

By enriching the Store Dimension with these attributes, deeper regional and managerial analysis can be facilitated.

- **Degenerate Dimension**:

Invoice_ID is a degenerate dimension, meaning it exists in the fact table but lacks additional descriptive attributes.

This field is primarily used for transaction tracking and reporting at the invoice level.

3.2 Fact Table Grain

The atomic unit of analysis is at the invoice line level. Each record corresponds to the sale of a single product in each invoice making it the granularity for the fact table. This granularity supports the following:

- Accurate transaction reporting: The invoice line level enables tracking of each item sold, making it possible to generate detailed transaction reports.

- Aggregated reporting: Aggregated reports can be generated by customer, product, store, or time, facilitating insights into sales performance at different levels.

- Drill-down capability: The granularity allows for in-depth analysis, such as drilling down from overall sales to specific customer or product-level insights.

4. Dimensional Modeling Strategy

Dimensional modeling focuses on structuring data in a way that makes querying efficient and intuitive for business users. In this case, the star schema is chosen for its simplicity, ease of use, and optimal performance in analytical querying.

4.1 Star Schema Design

Fact Table:

- **Fact_Sales**
    - Fact_ID (Surrogate Key)
    - Invoice_ID (Degenerate Dimension)
    - Customer_Key (Foreign Key)
    - Product_Key (Foreign Key)
    - Store_Key (Foreign Key)
    - Date_Key (Foreign Key)
    - Quantity (Numeric Fact)
    - Unit_Price (Numeric Fact)
    - Line_Total (Derived Fact)

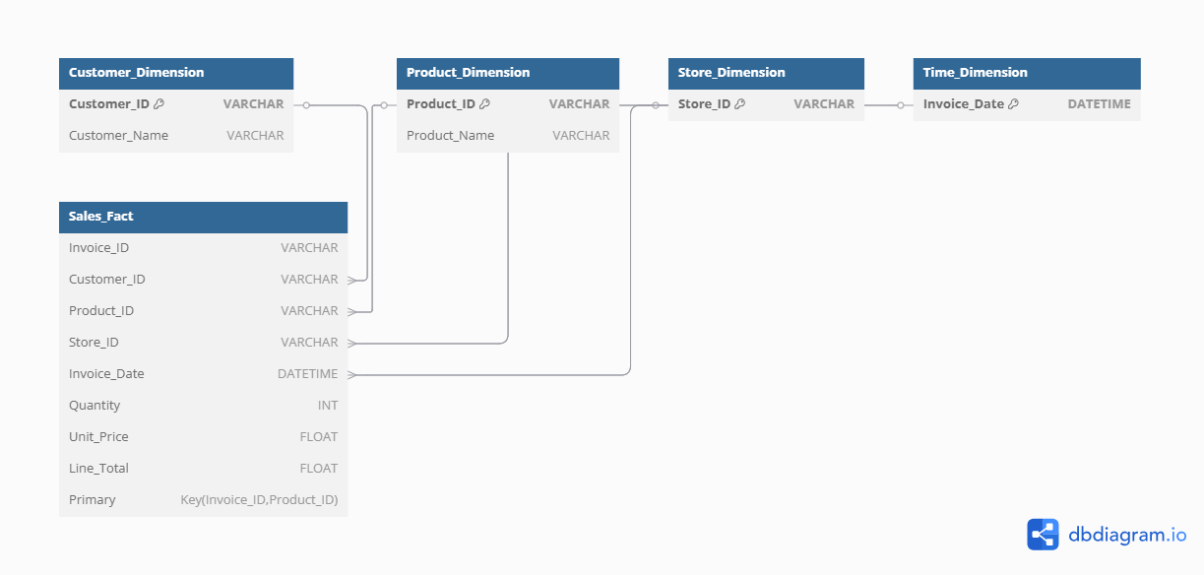
Dimension Tables:

- **Dim_Customer**
    - Customer_Key (Primary Key)
    - Customer_ID
    - Customer_Name
- **Dim_Product**
    - Product_Key (Primary Key)
    - Product_ID
    - Product_Name
    - Category
    - Brand
- **Dim_Store**
    - Store_Key (Primary Key)

    - Store_ID
    - Store_Name
    - Region
    - Manager
- **Dim_Date**
    - Date_Key (Primary Key)
    - Full_Date
    - Day
    - Month
    - Quarter
    - Year
    - Weekday

This star schema facilitates easy aggregation of sales data and provides a clear path for users to perform cross-dimensional analysis such as revenue by product, store performance over time, or customer purchasing behavior.

```
Table Customer_Dimension {
    Customer_ID VARCHAR [primary key]
    Customer_Name VARCHAR
}

Table Product_Dimension {
    Product_ID VARCHAR [primary key]
    Product_Name VARCHAR
    Unit_Price FLOAT
}

Table Store_Dimension {
    Store_ID VARCHAR [primary key]
    Store_Location VARCHAR
}

Table Time_Dimension {
    Time_ID INT [primary key]
    Invoice_Date DATETIME
}

Table Sales_Fact {
    Invoice_ID VARCHAR
    Customer_ID VARCHAR [ref: > Customer_Dimension.Customer_ID]
    Product_ID VARCHAR [ref: > Product_Dimension.Product_ID]
    Store_ID VARCHAR [ref: > Store_Dimension.Store_ID]
    Time_ID INT [ref: > Time_Dimension.Time_ID]
    Quantity INT
    Unit_Price FLOAT
    Line_Total FLOAT

    Primary Key (Invoice_ID, Product_ID)
}
```



4.2 **Degenerate Dimensions**

Invoice_ID qualifies as a degenerate dimension. Since it is used for transaction-level tracking but does not have associated descriptive attributes, it remains in the fact table. Its primary role is to track individual sales transactions for reporting purposes.

5. Slowly Changing Dimensions (SCD)

SCDs are critical for handling historical changes in dimension attributes over time. For this model, different types of SCDs are applied depending on the dimension and business requirements.

5.1 SCD Types Considered

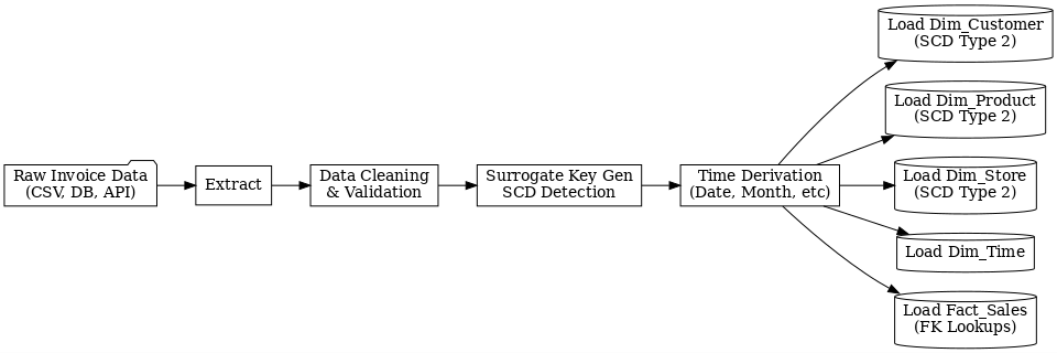
- **Type 1**: Overwrites old data. This type is suitable for cases where historical accuracy is not needed (e.g., correcting typographical errors).
- **Type 2**: Maintains historical data by versioning rows. This is crucial for tracking changes in customer information, product details, or store management, ensuring that historical data is preserved.
- **Type 3**: Stores both current and previous values in the same row. This type can be used for limited tracking of changes, such as capturing a previous price along with the current one.

5.2 Application of SCDs

- **Dim_Customer**: Use **Type 2** to track changes in customer data (e.g., changes in customer name or segment).
- **Dim_Product**: Use **Type 2** to capture changes in product information, such as category or pricing updates.
- **Dim_Store**: Use **Type 2** for changes in store data, including location or management.
- **Dim_Date**: This dimension does not require an SCD as time-related attributes are static and do not change over time.

6.Extract Transform Load/Extract Load Transform Strategy

The ETL/ELT process plays a critical role in ensuring that data is extracted, transformed, and loaded efficiently and accurately.



6.1 Batch and Incremental Load

- **Initial Full Load (Batch)**: This step involves extracting all source data, cleaning it, standardizing it, and loading it into the appropriate dimension and fact tables.
- **Incremental Load (Daily/Hourly ELT)**: In this step, only new or changed records are identified and processed. Techniques such as change data capture (CDC) or timestamp-based watermarking are used to detect and process changes efficiently.

6.2 Data Quality and Validation

Data quality is paramount in ensuring that the dimensional model is consistent and reliable. The following checks are implemented:

- Null checks for mandatory fields
- Data type validation (e.g., ensuring numeric fields are not text)
- Duplicate detection, especially in the combination of Invoice_ID and Product_ID
- Referential integrity between fact and dimension tables
- Logging of errors and ETL failures for auditing and debugging purposes

7. Advanced Analysis and Scalability

7.1 Hierarchical Relationships

Hierarchies in the dimensions allow for drill-down and roll-up analysis:

- **Time Hierarchy**: Hour → Day → Month → Quarter → Year
- **Product Hierarchy**: Product → Category → Brand
- **Store Hierarchy**: Store → Region

These hierarchies facilitate detailed analysis, such as drilling down from monthly sales to daily performance or aggregating store performance by region.

7.2 Performance Optimization

To ensure that the model scales with data growth and supports efficient querying, the following techniques are implemented:

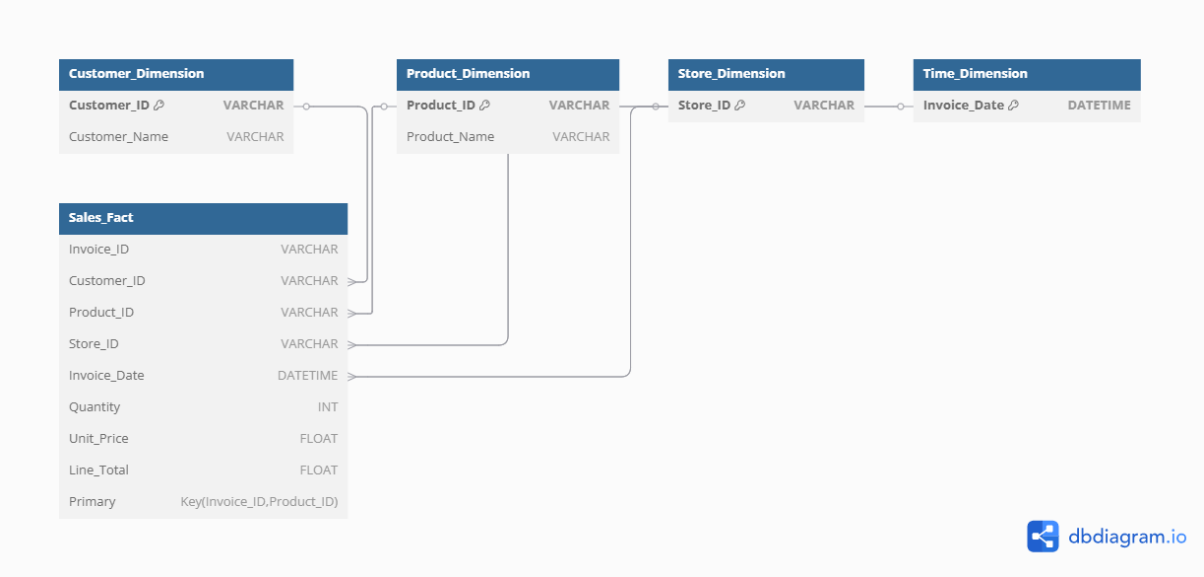
- **Partitioning**: The fact table is partitioned by time (e.g., monthly or yearly) to improve query performance on large datasets.
- **Indexing**: Foreign keys in the fact table and the time-related fields in the Date dimension are indexed for faster joins.
- **Materialized Views**: Pre-aggregated summaries of sales data (e.g., total revenue by category) are stored in materialized views for quicker access.
- **Surrogate Keys**: Surrogate keys (integer-based) are used in dimension tables to optimize join performance and avoid potential issues with natural keys.
- **Parallel Processing**: ETL jobs can be executed in parallel across multiple cores to speed up data processing.

8. Business Objectives Fulfillment

Objective	Model Feature	Benefit
Analytical Reporting	Star Schema, Facts & Dimensions	Flexible queries and reporting
Drill-down Capability	Time, Product, Store Hierarchies	Deep analysis by any dimension
Historical Tracking	SCD Type 2	Accurate historical analysis
Scalability & Performance	Partitioning, Indexing, Surrogate Keys	Efficient querying for large datasets

9. Star Schema Diagram

Below is the visual representation of the Star Schema used in this dimensional model:



10. Assumptions and Trade-Offs

The design and implementation of the invoice-based dimensional model rely on several key assumptions and strategic trade-offs to balance complexity, performance, scalability, and usability. These are outlined below:

10.1 Assumptions

1. **Grain of the Fact Table is Invoice Line Level**
   It is assumed that each record in the dataset corresponds to a single product sold within an invoice. This atomic level of granularity supports maximum analytical flexibility and enables aggregations across dimensions such as product, customer, and time.

2. **Unit Price May Vary Across Transactions**
   Unit price is treated as a transaction-level attribute stored in the fact table, not in the product dimension. This assumes that pricing can vary based on sales, promotions, or customer-specific discounts.

3. **Use of Surrogate Keys Across All Dimensions**
   Each dimension table uses a surrogate key (auto-generated integer ID) instead of relying on business/natural keys like Customer_ID or Product_ID. This improves join performance and allows for SCD implementation.

4. **Type 2 SCDs for Dim_Customer, Dim_Product, and Dim_Store**
   Historical tracking is important for these dimensions, especially for time-based analysis. Therefore, changes in attributes like customer name, store region, or product category are versioned using SCD Type 2.

5. **Static Time Dimension**
   The time dimension is derived from invoice dates and is assumed to be static. No changes to time attributes (e.g., calendar structure) are expected.

6. **Single Currency Context**
   It is assumed that all transactions are recorded in the same currency. If multiple currencies were involved, an additional currency conversion dimension would be necessary.

7. **No Junk Dimensions Identified**
   All attributes in the dataset map clearly to specific business entities, and there is no need to group miscellaneous flags or indicators into a junk dimension.

8. **Data Volumes Are Expected to Grow Linearly**
   The model is designed under the assumption that data volumes will grow steadily over time. As such, performance optimizations like partitioning and indexing are in place to ensure query efficiency.

10.2 Trade-Offs

1. **Type 2 SCD Complexity vs Historical Accuracy**
   Implementing SCD Type 2 adds complexity in ETL processing and increases storage requirements due to multiple rows per entity over time. However, this is justified by the need for accurate historical analysis and auditability.

2. **Star Schema vs Snowflake Schema**
   A star schema was chosen for its simplicity and performance in querying, even though a snowflake schema might reduce redundancy. This trade-off favors user accessibility and reporting speed over strict normalization.

3. **Storing Derived Facts in the Fact Table**
   Line_Total is stored in the fact table, even though it can be computed from Quantity × Unit_Price. This design decision trades off some normalization to improve query performance and simplify reporting.

4. **Batch + Incremental ELT Over Streaming**
   The ETL/ELT strategy is based on scheduled batch and micro-batch (incremental) loads, assuming that real-time analytics is not a strict requirement. This reduces infrastructure complexity but limits immediate availability of the most recent data.

5. **Predefined Time Hierarchies**
   Time hierarchies (Hour > Day > Month, etc.) are hardcoded based on the Gregorian calendar. Any need for fiscal calendars or custom business periods would require changes to the time dimension.

6. **Limited Enrichment of Dimensions in Initial Design**
   To meet project scope and deadlines, dimensions were designed with essential attributes. Further enrichment (e.g., customer segments, product subcategories) is planned but not implemented in the initial version.

7. **ETL Failure Logging vs Automatic Recovery**
   While failures are logged for audit and investigation, automatic recovery from ETL interruptions is not part of the initial implementation, prioritizing transparency over resilience.

11. Conclusion

This dimensional model has been designed to ensure that the business can extract valuable insights from transaction-level data. By implementing best practices in dimensional modeling, such as the **star schema, surrogate keys, SCDs, and scalable ETL processes**, this model provides a solid foundation for advanced analytics, forecasting, and business intelligence. With the potential for expansion and adaptation to additional attributes or dimensions, it is well-suited for handling the growth of the retail business and addressing emerging data analysis needs.

12. Reference

Stage 8 Task B: https://dorian-twister-cea.notion.site/Stage-8-Task-B-1c1241d25b9c80dcb3f5dd77ac372161