Università di Catania

Master's Degree in Data Science

Data Warehouse Design Report

Star Schema for Transaction Analysis

Modeling a Data Warehouse Schema for Financial Transaction Analysis

Author: Angelo Schillaci
Student ID: 1000033449
Instructor: Prof. Giovanni Morana

Date: 16/06/2025

Contents

1 Introduction

This document presents the logical design of the Data Warehouse (DW) for transaction analysis, based on the *Star Schema* illustrated in Figure 1. The design process follows the structured methodology proposed by **Golfarelli and Rizzi**, with the primary goal of creating a data infrastructure optimized for OLAP operations within a ROLAP environment.

The project is guided by the following key objectives:

- To support fast and flexible **multidimensional analysis** of trading operations across various business axes (time, geography, symbol, transaction type).
- To ensure high performance and query efficiency through a **denormalized** dimensional structure that provides a **consistent** and **semantically** meaningful basis for data exploration and reporting.

This document explains the rationale behind the modeling choices, from the adoption of a top-down design strategy to the construction of the star schema and the implementation of key OLAP queries and interactive dashboards. The aim is to demonstrate how the Data Warehouse supports advanced analysis, facilitates insight discovery, and aligns technical design with business goals.

2 Star Schema

In this section, we will explain all the choices that influence the building of this star schema in Figure 1.

Starting from the conceptual specifications provided:

Fact

- `Fact_Transactions`

Dimensions

- `Dim_Time`
- `Dim_Geography`
- `Dim_Symbol`
- `Dim_TransactionType`

we describe how these elements have been translated into a logical star model. Surrogate keys are introduced, foreign keys mapped, and the implementation of dimensional hierarchies is illustrated in accordance with the guidelines of Golfarelli and Rizzi.

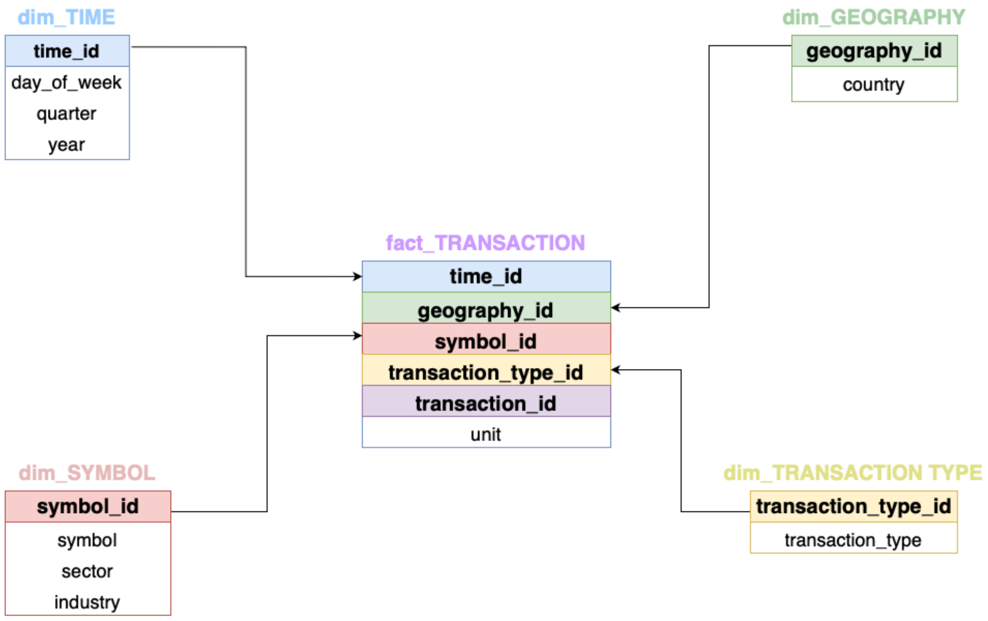

Figure 1: Star Schema

2.1 Design Strategy and Methodology

In this project, we adopted a **top-down approach**: the star schema was designed directly based on analytical requirements—that is, from the analysis of the questions that the Data Warehouse must be able to answer. This strategy guarantees that every table, dimension, and measure in the Data Warehouse is strictly aligned with the analytical objectives, **minimizing redundancy and ensuring semantic consistency**.

The design process began by identifying and formalizing the key business questions:

- Analyzing all quarters of 2024 by the total number of transactions (`BUY` + `SELL`).
- Identifying the top 5 countries by number of units traded in Financial Services companies.
- Finding the top 3 sectors by total number of units sold on Mondays across 2024.
- Performing time-based analysis through a line chart showing the number of transactions over time, and bar charts showing the most traded symbols, sectors, and industries.

Based on these defined analytical goals, this phase was crucial to ensure that every piece of data modeled within the Data Warehouse had a clear and justified analytical purpose. Each measure, attribute, and hierarchy was carefully selected to support the required queries in a way that is both semantically meaningful and computationally efficient.

Once analytical needs were clearly defined, the conceptual model (DFM) was translated into a logical *Star Schema*, introducing surrogate keys and mapping foreign key relationships. This structure ensures high performance for OLAP queries and a clear dimensional organization.

At the end, only after the logical model was consolidated did we proceed with the identification of the source systems and the design of the necessary ETL processes to populate the Data Warehouse.

2.2 Model overview

The model consists of one fact table, `fact_TRANSACTION`, and four dimension tables: `dim_TIME`, `dim_GEOGRAPHY`, `dim_SYMBOL`, and `dim_TRANSACTION_TYPE`.

As previously stated, this is the final star schema designed based on the analytical requirements, with the goal of enhancing the overall **consistency** and alignment between data structure and reporting needs.

2.2.1 Fact table

The fact table is `fact_TRANSACTION`, it is composed by the set of foreign keys referencing dimension table, but the **concatenation of these is not enough to guarantee the uniqueness of each row**; for this reason we also include the primary key `transaction_id`. Furthermore, the fact table contain the measure unit with **SUM** as **aggregation operator**.

Looking all table, we have:

- `time_id` : **foreign key** of the time dimension.
- `geography_id` : **foreign key** of the geography dimension.
- `symbol_id` : **foreign key**y of the symbol dimension.
- `transaction_type_id` : **foreign key** of the transaction-type dimension.
- `transaction_id` : **primary key** inherited from the source system; kept in the fact table as a *degenerate dimension* to support drill-through and reconciliation.
- `unit` : **measure** that represents the number of units traded; it is fully additive and its default OLAP roll-up operator is **`SUM`**. A complementary implicit measure **`COUNT(*)`** returns the number of transactions.

2.2.2 Dimensions

The `dim_TIME` provides temporal context for each transaction and is linked to the fact table via `time_id`, which acts as the ***surrogate key***. Unlike a full calendar dimension that typically includes day, month, and full date attributes, this implementation focuses only on the attributes most relevant to the analytical goals of this data warehouse.

Attributes:

- `day_of_week` : weekday label (Mon–Sun), useful for identifying weekly trading patterns.
- `quarter` : fiscal quarter (Q1–Q4), used for seasonal comparisons.
- `year` : calendar year, the top level of the time hierarchy.

Hierarchy

The analytical hierarchy follows this drill path:

`Day of Week` → `Quarter` → `Year`.

Although this path does not reflect a conventional calendar structure, it was intentionally designed to match the granularity and reporting requirements identified during the data analysis phase.

The `dim_GEOGRAPHY` provides the geographical context for each transaction. It is linked to the fact table via the ***surrogate key*** `geography_id` and enables regional filtering and aggregation.

Attributes:

- `country` : the country where the issuing company is headquartered or registered; this serves as the sole level of the geographical hierarchy used for aggregation.

Hierarchy

This is a flat, single-level hierarchy with no sub-regions: each transaction is directly associated with a country, since analysis at broader or more granular geographic levels (e.g., region, sub-region) is not required for this project.

The `dim_SYMBOL` captures the business identity of the traded asset and its classification within the financial market. It is connected to the fact table through the ***surrogate key*** `symbol_id` and allows aggregation by product or business category.

Attributes:

- `symbol` : the ticker or identifier of the traded asset (e.g., stock symbol).
- `company_name` : the complete name of each symbol.
- `sector` : the economic sector to which the asset belongs.
- `industry` : the specific industry classification, nested within the sector.

Hierarchy

The dimension follows a hierarchical structure:

`Symbol` → `Sector` → `Industry`.

This allows analysts to perform drill-downs and roll-ups from specific assets to broader business categories, enabling both high-level overviews and detailed investigations.

The `company_name` attribute was **intentionally included**, as the presence of full names improves the **readability, interpretability, and usability** of visualizations and aggregated reports — especially in chart labels and dashboards. This modeling choice aligns with the needs of the **target audience** of the data warehouse, which may include both domain experts familiar with standard symbols and non-expert users who benefit from more descriptive labels.

The `dim_TRANSACTION_TYPE` describes the nature of the transaction event, distinguishing whether it represents a buy, sell, or divident. It is linked to the fact table via the ***surrogate key*** `transaction_type_id` and allows basic behavioral segmentation of trading activity.

Attribute:

- `transaction_type` : a categorical label that defines the type of transaction (e.g., `Buy`, `Sell`).

Hierarchy

This is a non-hierarchical dimension. The attribute is flat and self-descriptive, with no subcategories or roll-up levels required.

2.3 ETL Process

Once the logical model was finalized, the ETL (Extract, Transform, Load) process was designed to populate the Data Warehouse with consistent and high-quality data.

2.3.1 Extraction

The source data were provided in CSV format and consist of the following files:

1. `symbols.csv` — Contains information about listed financial instruments, including company name, sector, industry, and country of registration.
2. `account-statement-1-1-2024-12-31-2024.csv` — Contains the transactional log, recording each trade with date, transaction type, symbol, and quantity.
3. `country.csv` — Contains metadata about countries, such as ISO codes, region, and sub-region.

However, since the analytical objectives of the project do not require geographic granularity beyond the country level, the `country.csv` file was **excluded** from the ETL process. Only the first two data sources were used to construct the fact and dimension tables in the star schema.

2.3.2 Transformation

The transformation phase included the following steps:

- **Column cleaning**: removal of the unnamed column and rows with all-null values.
- **Renaming**: all columns were renamed to match the naming conventions adopted in the DW.
- **Date parsing and enrichment**: the `date` field was parsed into a proper datetime object; from it, the following derived attributes were extracted:
  - `day_of_week`
  - `year`
  - `quarter`

  After the original `date` column was removed.

- **Join**: a `LEFT JOIN` was performed between the transactions and the symbol dataset (the first two data sources mentioned above), using the `symbol` key as join attribute.

  This join type was chosen because the fact table is based on transactions, and it is essential to retain all of them—even if some symbols have no corresponding metadata—in order to preserve analytical completeness.

- **Null handling**: after the join, missing values in the fields `company_name`, `sector`, `industry`, and `country`—corresponding to transactions that had no matching record in `symbols.csv`—were replaced with the default label `"unknown"`. This affected approximately 9.29% of the records and was necessary to ensure schema consistency and prevent null propagation across the dimensional model. During the analysis phase, these **unknown** values will be explicitly **highlighted** to ensure transparency and allow users to assess their impact.

2.3.3 Loading

The loading step was handled programmatically within the **Streamlit** application itself. No external database or file-based storage was used for populating the star schema tables; instead, the transformed data were directly structured and used in memory during execution.

3 Data Analysis

This section illustrates how the star schema enables a set of multidimensional queries by applying OLAP techniques to extract meaningful insights. The design of the Data Warehouse supports interactive analysis through operations such as **roll-up**, **drill-down**, **slice**, and **dice**.

OLAP Query Examples and Interpretation

1. **Analyzing all quarters of 2024 by the total number of transactions (BUY + SELL)**
   This operation combines **dice** and **roll-up**:

   - A **dice** operation filters facts based on two dimensions:
     - `dim_TIME`, selecting only rows where `year = 2024`.
     - `dim_TRANSACTION_TYPE`, restricting the analysis to `BUY` and `SELL`.
   - A **roll-up** operation aggregates data by `quarter`, using the corresponding attribute in `dim_TIME`, with `COUNT(*)` as the aggregation function.

   The resulting visualization presents a grouped bar chart where each quarter shows the number of **`BUY` and `SELL` transactions side by side**, allowing immediate comparison of trading activity across types and periods.

   In designing the chart, particular attention was given to accessibility: instead of using red and green—which can be problematic for **color-blind users**—more distinguishable hues were adopted (e.g., blue and orange) to ensure inclusiveness and clarity in visual interpretation.

   This allows analysts to compare transaction volumes over time and by type within 2024.

2. **Top 5 countries by number of units traded in Financial Services companies**
   This query applies **slice** and **roll-up**:

   - A **slice** operation filters only companies in the `Financial Services` sector using `dim_SECTOR`.
   - A **roll-up** operation aggregates the data by `country`, using the attribute in `dim_GEOGRAPHY`, and `SUM(units)` as the aggregation function *(where* `units` *is a measure stored in the fact table).*

   The countries identified are shown both on a world map and in a horizontal bar chart, allowing users to ***simultaneously observe spatial distribution and quantitative ranking***. This dual visualization enhances interpretability by connecting geographic location with trading intensity in an intuitive and visually accessible way.

   The result highlights the geographic distribution of trading activity in this specific sector.

3. **Top 3 sectors by total number of units sold on Mondays across 2024**
   This query uses **slice** and **roll-up**:

   - A **slice** operation filters transactions occurring in `year = 2024` and on `day = Monday`, both attributes belonging to `dim_TIME`.
   - A **roll-up** operation aggregates the data by sector, using the attribute in `dim_SECTOR`, with `SUM(units)` as the aggregation function *(where* `units` *is a measure stored in the fact table).*

   This query reveals which sectors dominate Monday trading activity over the course of 2024.

These OLAP operations demonstrate the effectiveness of the schema in supporting multidimensional exploration, pattern discovery, and aggregation at various levels of granularity. They also confirm the validity of the top-down modeling strategy adopted, in which analytical goals directly shaped the dimensional structure.

4 Time Analysis

This section focuses on the temporal dimension of transactions, enabling users to explore patterns over time and identify the most active categories in different periods.

An interactive dashboard allows the user to define a custom time range by selecting a **start and end quarter/year**. Once the period is set, transactions are filtered accordingly, excluding dividend operations to maintain focus on active trading (`BUY` and `SELL` types only).

The choice to include `quarter` and `year` as explicit attributes in the `dim_TIME` dimension—rather than relying solely on exact dates (e.g., `01/01/2024`)—facilitates a more business-oriented analysis. This enables intuitive filtering, high-level temporal aggregation, and easier comparison across periods.

The filtered data is then analyzed and visualized through four main components:

1. **Temporal Trend**
   A line chart displays the **number of transactions per quarter** (`COUNT(*)`), showing how trading activity evolved over time. The x-axis represents periods in `YYYY-QN` format, allowing clear trend detection and comparisons.

2. **Symbol Analysis**
   Users can explore the **most traded financial instruments (symbols)** within the selected time range. A bar chart shows the top-N symbols, optionally ***colored by sector***, supporting **cross-dimensional comparison** (symbol vs. sector).

3. **Sector Analysis**
   A bar chart displays the top-N **sectors** by transaction volume. This view highlights the most active macro areas of the market in the selected period.

4. **Industry Analysis**
   Another bar chart shows the top-N **industries**, offering a more granular perspective. An optional ***color mapping by sector*** allows users to relate each industry to its corresponding macro-sector, supporting **cross-dimensional comparison** (industry vs. sector).

5 Conclusion

This project demonstrates the design and implementation of a Data Warehouse tailored to transaction analysis, following a top-down modeling strategy based on analytical requirements. By adopting a star schema structure, the model ensures a high degree of consistency, simplicity, and efficiency in supporting OLAP queries.

The dimensional model, centered on the fact table `fact_TRANSACTION`, enables flexible and expressive multidimensional analysis across time, geography, product, and transaction type. The Streamlit-based visualizations confirm that the schema supports both technical needs and user-facing analytical goals.

The star schema designed:

- follows the Golfarelli–Rizzi principles in translating from the DFM;
- optimises OLAP query performance through denormalised dimensions and surrogate key usage;
- provides consistent semantics aligned with analytical objectives.

This structure provides a solid, scalable, and governable foundation for multidimensional analysis on transaction data.