

# Design of the Datawarehouse

In the realm of data management, design a Datawarehouse system that meets the requirements defined in the previous step (*1\_requirements\_analysis*), is a multifaceted task. It involves a meticulous process that unfolds in three distinct phases: **conceptual**, **logical** and **physical design**. These levels of design are crucial in creating a DWH that not only captures the essence of the data but also ensures its integrity, efficiency and security.

## Conceptual Design

The conceptual design is the highest level of abstraction in the Datawarehouse design process. At this stage, designers focus on understanding the problem domain and defining the overall structure of the database without getting into technical implementation details. The primary goal is to create a clear and comprehensive representation of the data and its hierarchies.

For this purpose, we decided to use the so called “*DFM, Dimensional Fact Model*” which represents a graphical formalism specifically devised to support the conceptual modelling phase in a data warehouse project. DFM is extremely intuitive because it can be used by analysts, technical users, non-technical users and customers. It is used to realize a clear and exhaustive representation of multidimensional concepts (e.g., attributes, measures and hierarchies).

In order to understand the schema, we must have a clear understanding of what is exactly a fact, a measure and a dimension:

- A *fact* is a concept relevant to decision-making processes. It typically models a set of events taking place within a company. An example of facts in the Foreign Exchange is the *F\_FX\_SAMPLE\_RATE* table or the *F\_FX\_CURRENCY\_RATE* table.
- A *measure* is a numerical property of a fact that describes a quantitative attribute, relevant for the analysis. For example, in the *F\_FX\_CURRENCY\_RATE*, example of measures are *CurrentRate*, and *YestardayNYTimeChange*.
- A *dimension* is a property that describes an analysis coordinate of the fact. A fact generally has multiple dimensions that define its minimum representation granularity. Typical dimensions for the context are dates and currency.

## DFMs – Dimensional Fact Models

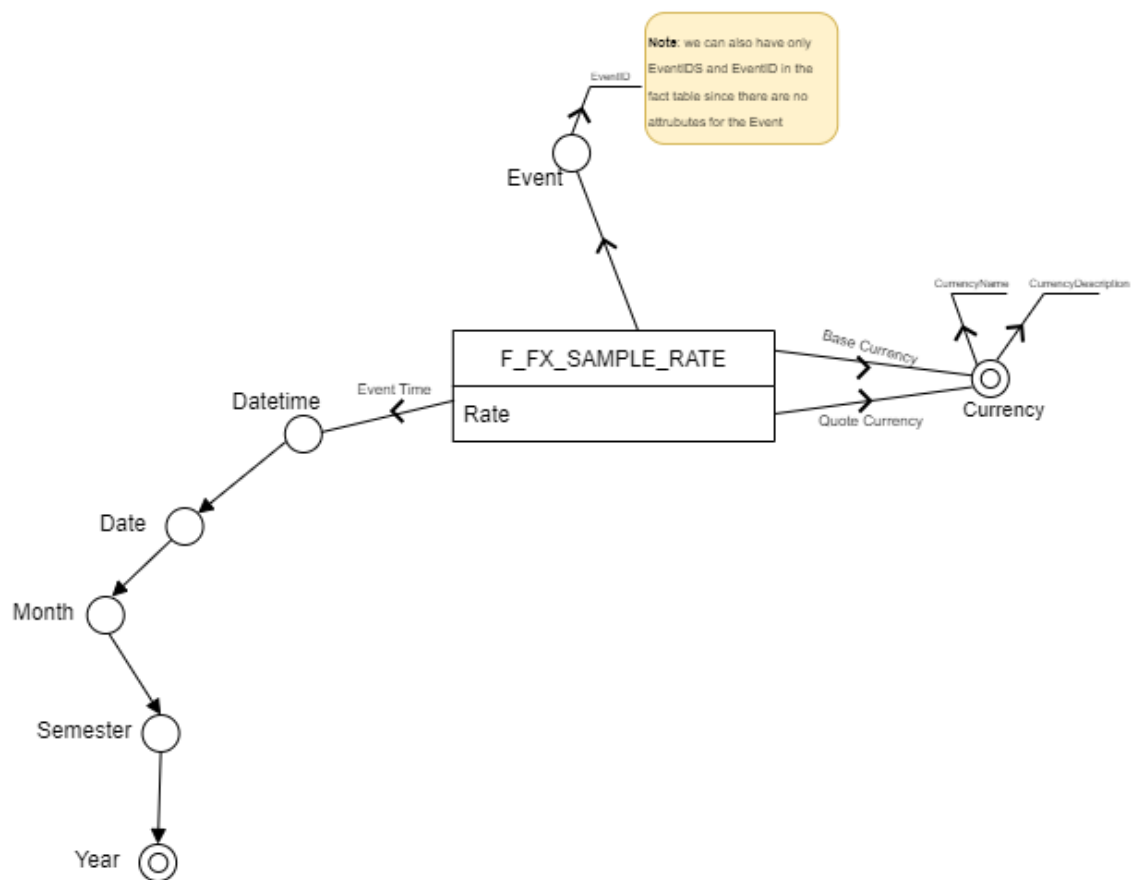


Figure 1 - Dimensional Fact Model, F\_FX\_SAMPLE\_RATE

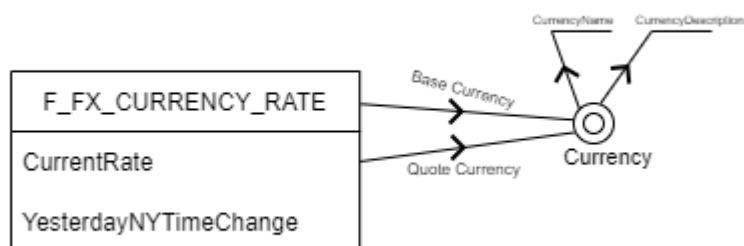


Figure 2 - Dimensional Fact Model, F\_FX\_CURRENCY\_RATE

### Business Logic for measures:

- The “Rate” in *F\_FX\_SAMPLE\_RATE* fact is just what we receive in input for a particular eventId. The input comes from the CSV file.
- The “CurrentRate” in *F\_FX\_CURRENCY\_RATE* fact, according to the requirements, is computed as follows: we want to consider only **active** rates. A rate is considered active if and only if it’s the last one received for a given currency couple AND it’s not older than 30 seconds.
- The “YesterdayNYTimeChange” in *F\_FX\_CURRENCY\_RATE* fact instead, is computed by dividing the *CurrentRate* with the yesterday’s rate at 5PM New York time. The latter is computed by checking the most recent rate starting from yesterday at 5PM NY Time back to the past. If there is no correspondence, we consider the less recent row starting from yesterday at 5PM NY Time going to the future.

## Logical Design

The logical design bridges the gap between the conceptual and physical levels. In this phase we want to translate the conceptual model into a more detailed representation, in the Datawarehouse design, this coincides with the definition of the star schema or snowflake schema. In the star schema the fact tables and dimension tables are contained. In this schema fewer foreign-key join is used. This schema forms a star with fact table and dimension tables. Snowflake schema is also a type of multidimensional model which is used for data warehouses. This schema forms a snowflake with fact tables, dimension tables as well as sub-dimension tables.

According to the conceptual model, it’s better in this phase to translate the DFM to a star schema since there aren’t lot of sub dimensions.

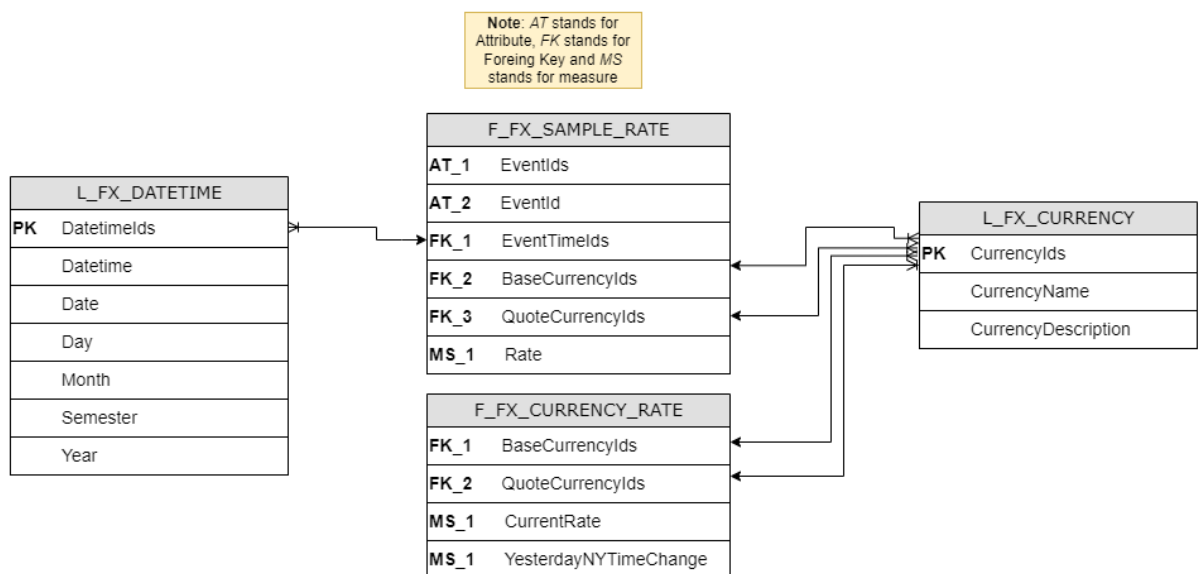


Figure 3 - Star schema

## Physical Design

Physical Datawarehouse design is the most detailed and technical level of the design process. At this stage, we make decisions about how the logical design will be implemented on a specific infrastructure. Considerations include indexing, storage, performance optimization and security measures. For this purpose, we create the following indices:

- Indices on non-key attributes in the dimension tables in order to accelerate the selection operations.
- Indices on external keys in the fact table in order to accelerate the execution of join operations.

Considering this, the following indices will be create:

Index	Table	Column
IDX_001	F_FX_SAMPLE_RATE	EventTimelds
IDX_002	F_FX_SAMPLE_RATE	BaseCurrencyIds
IDX_003	F_FX_SAMPLE_RATE	QuoteCurrencyIds
IDX_004	F_FX_CURRENCY_RATE	BaseCurrencyIds
IDX_005	F_FX_CURRENCY_RATE	QuoteCurrencyIds
IDX_005	L_FX_CURRENCY	CurrencyName
IDX_006	L_FX_CURRENCY	CurrencyDescription
IDX_007	L_FX_DATETIME	Datetime
IDX_008	L_FX_DATETIME	Date
IDX_009	L_FX_DATETIME	Day
IDX_010	L_FX_DATETIME	Month
IDX_011	L_FX_DATETIME	Semester
IDX_012	L_FX_DATETIME	Year

## Datawarehouse structure

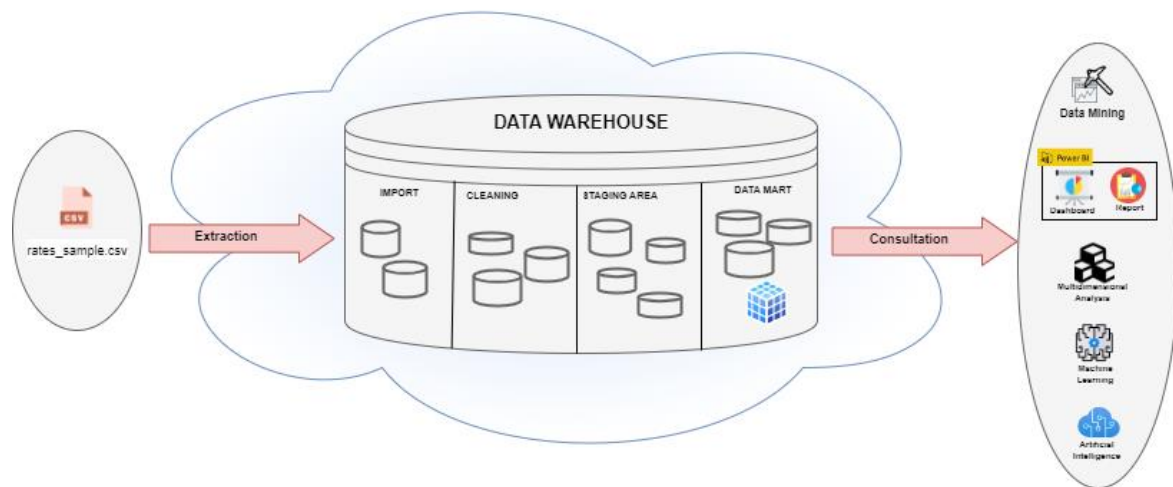


Figure 4 - DWH Structure