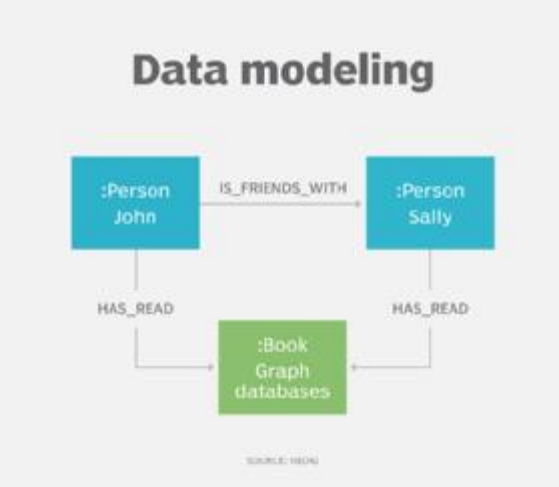
Data Modeling Summary

Rio Anggara Sufilin

What is Data Modeling?

Data Modeling is the process of creating a data model for the data to be stored in a database.

This **data model** is a conceptual representation of Data objects, the associations between different data objects, and the rules.

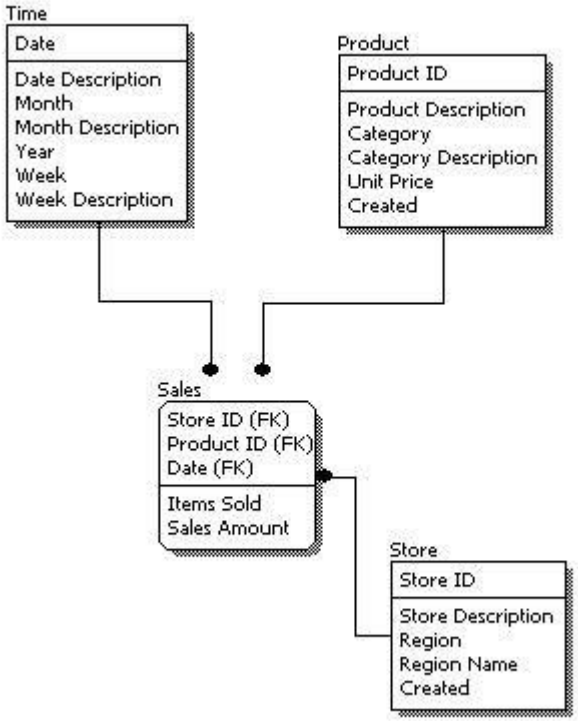


Why Use Data Model?

The primary goal of using data model are:

- Ensures that all data objects required by the database are accurately represented.
- A data model helps design the database at the conceptual, physical and logical levels.
- Data Model structure helps to define the relational tables, primary and foreign keys and stored procedures.
- It provides a clear picture of the base data and can be used by database developers to create a physical database.
- It is also helpful to identify missing and redundant data.
- Though the initial creation of data model is labor and time consuming, in the long run, it makes your IT infrastructure upgrade and maintenance cheaper and faster.

Data Model Example



Data Modeling Terminologies

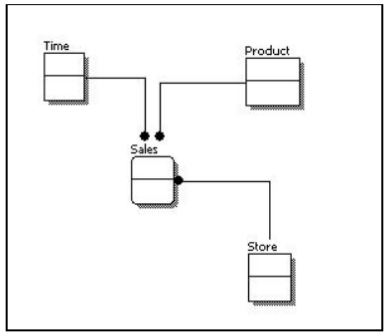
The 3 main things in Data Modeling are

- **Entity**: A real-world thing (table)
- **Attribute**: Characteristics or properties of an entity (column)
- **Relationship**: Dependency or association between two entities

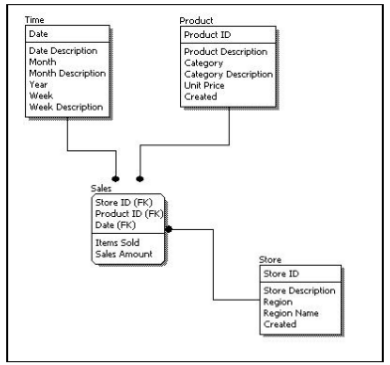
Types of Data Models

There are mainly **three** different **types** of **data models**: **conceptual data models**, **logical data models**, and **physical data models**, and each one has a specific purpose. The data models are used to represent the data and how it is stored in the database and to set the relationship between data items.

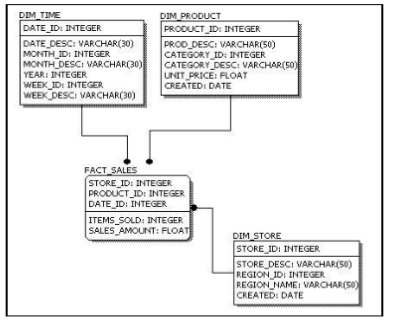
Conceptual Model Design



Logical Model Design



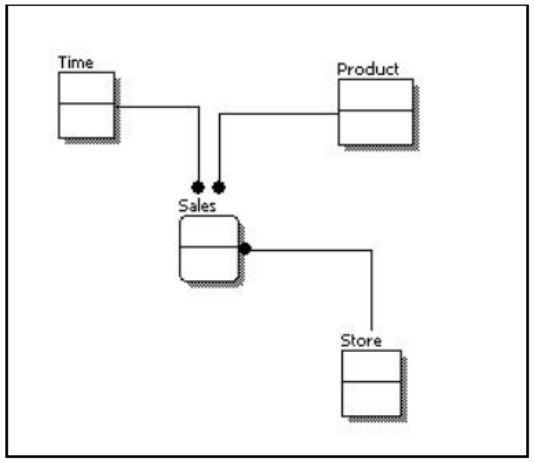
Physical Model Design



Conceptual Data Model

This Data Model defines **WHAT** the system contains. This model is typically created by Business stakeholders and Data Architects. The purpose is to organize, scope and define business concepts and rules.

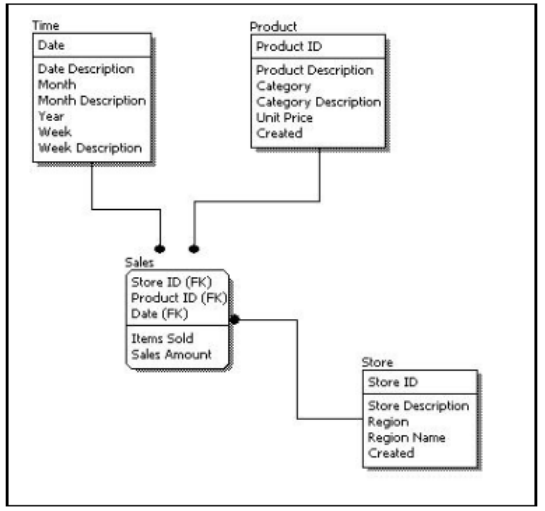
Conceptual Model Design



Logical Data Model

Defines **HOW** the system should be implemented regardless of the DBMS. This model is typically created by Data Architects and Business Analysts. The purpose is to developed technical map of rules and data structures

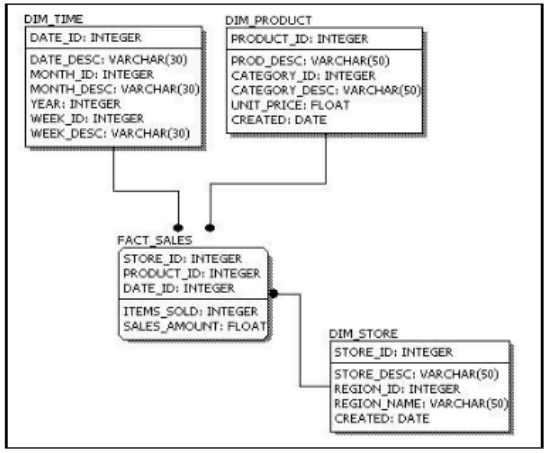
Logical Model Design



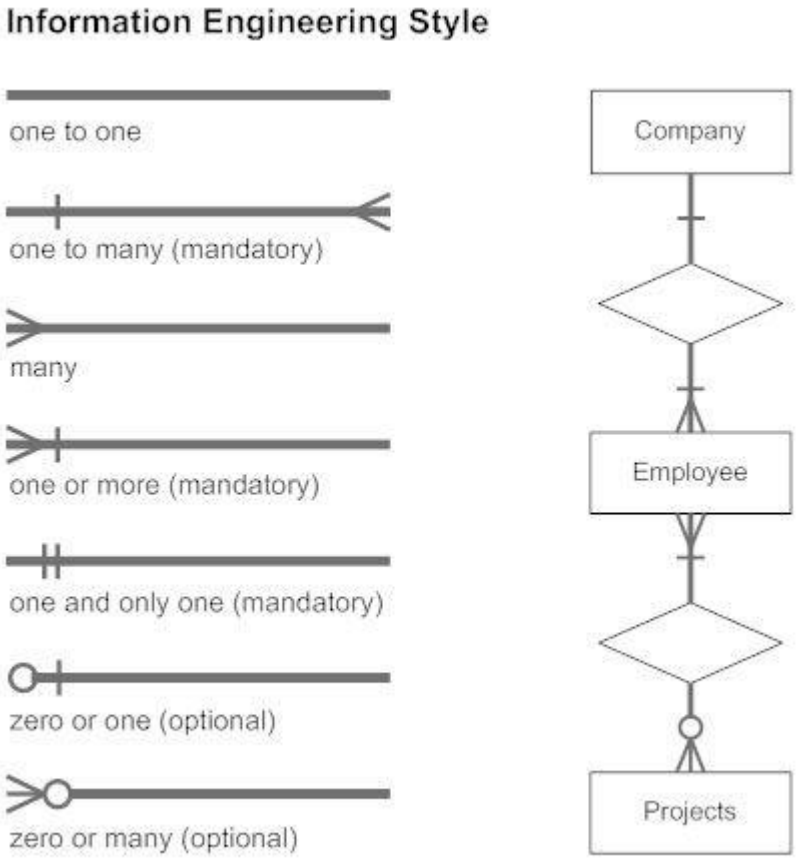
Physical Data Model

This Data Model describes **HOW** the system will be implemented using a specific DBMS system. This model is typically created by DBA and developers. The purpose is actual implementation of the database.

Physical Model Design

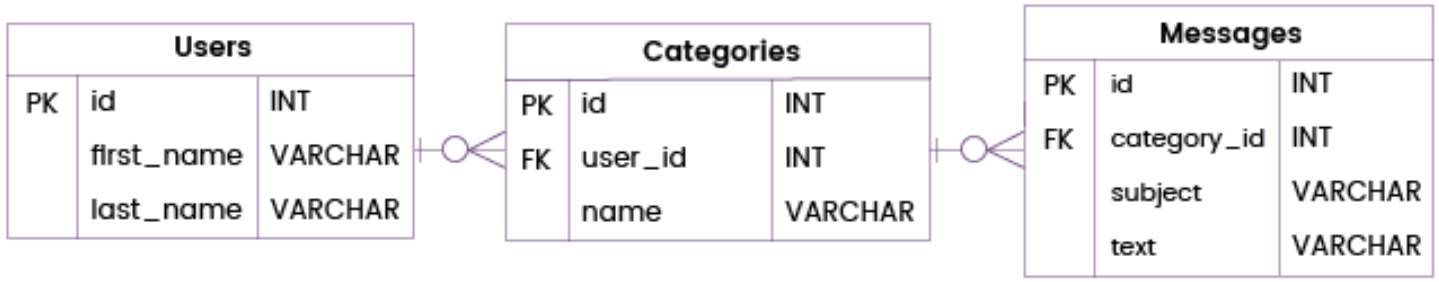


Cardinality

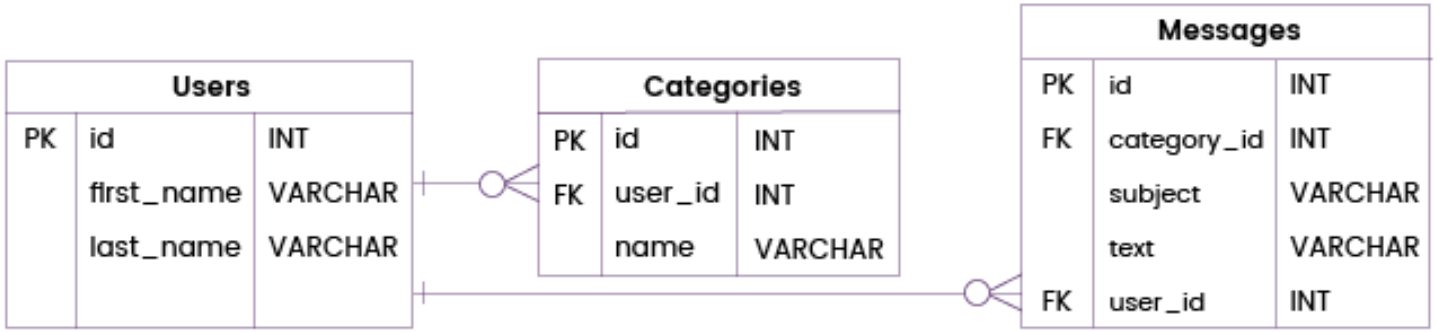


Normalization vs Denormalization

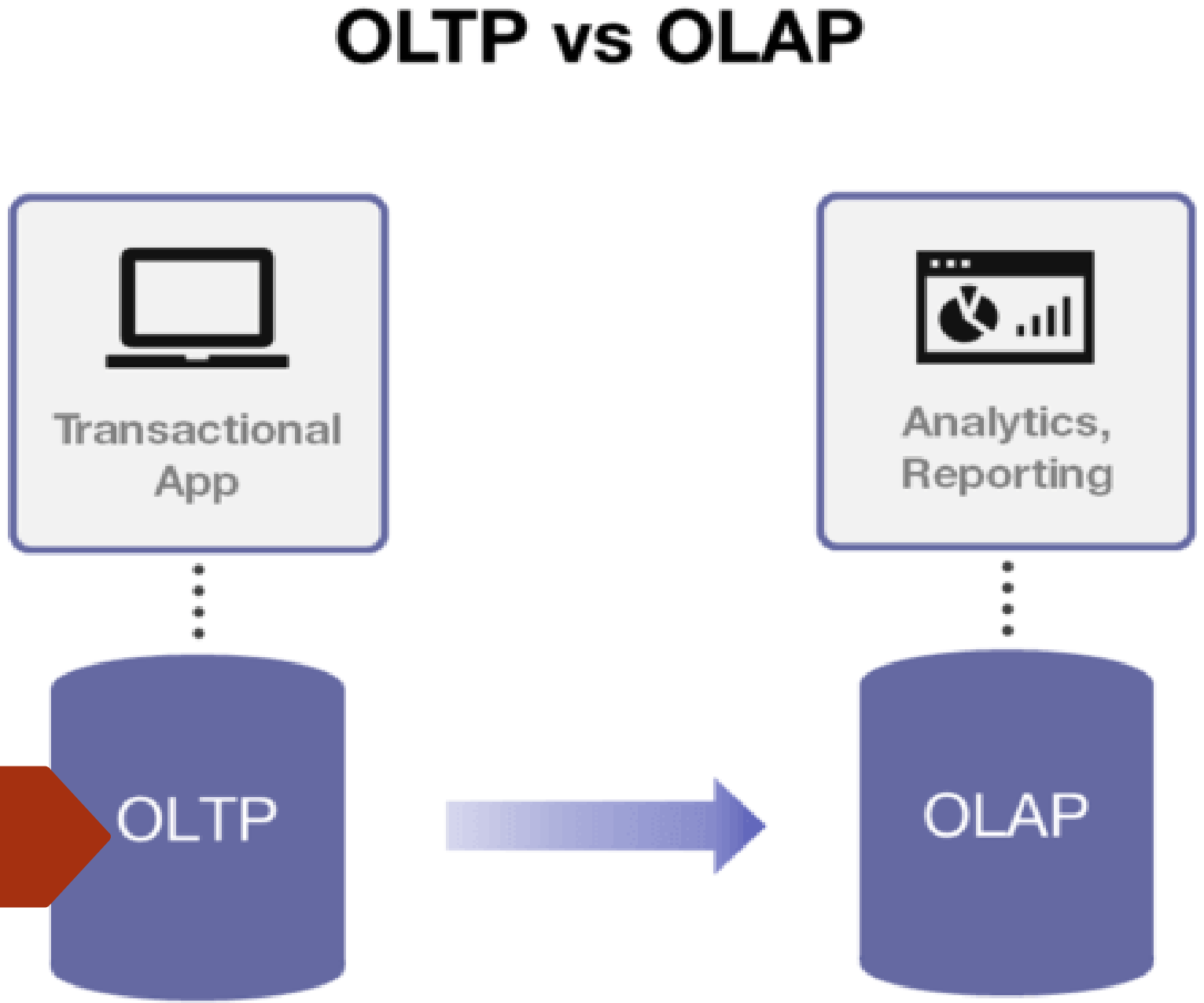
Normalized database



Denormalized database



OLTP & OLAP

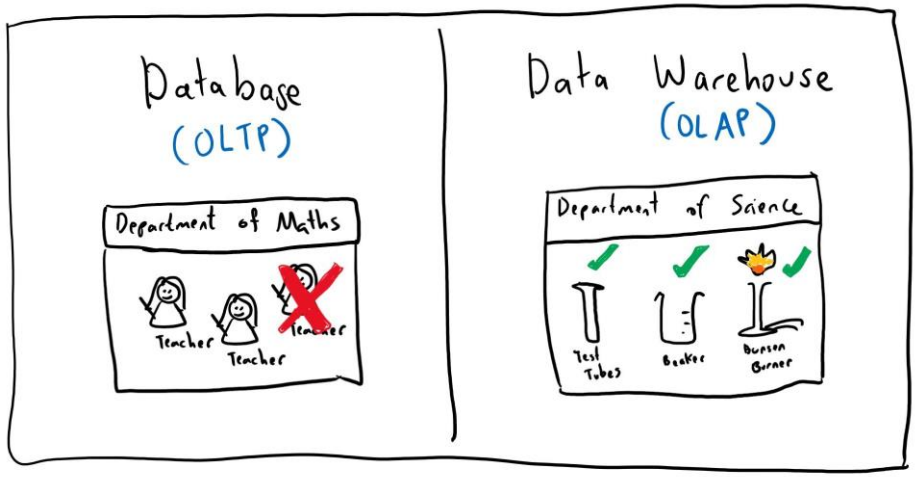

OLTP vs OLAP
Transactional App
Analytics, Reporting
OLTP
OLAP

OLTP vs OLAP

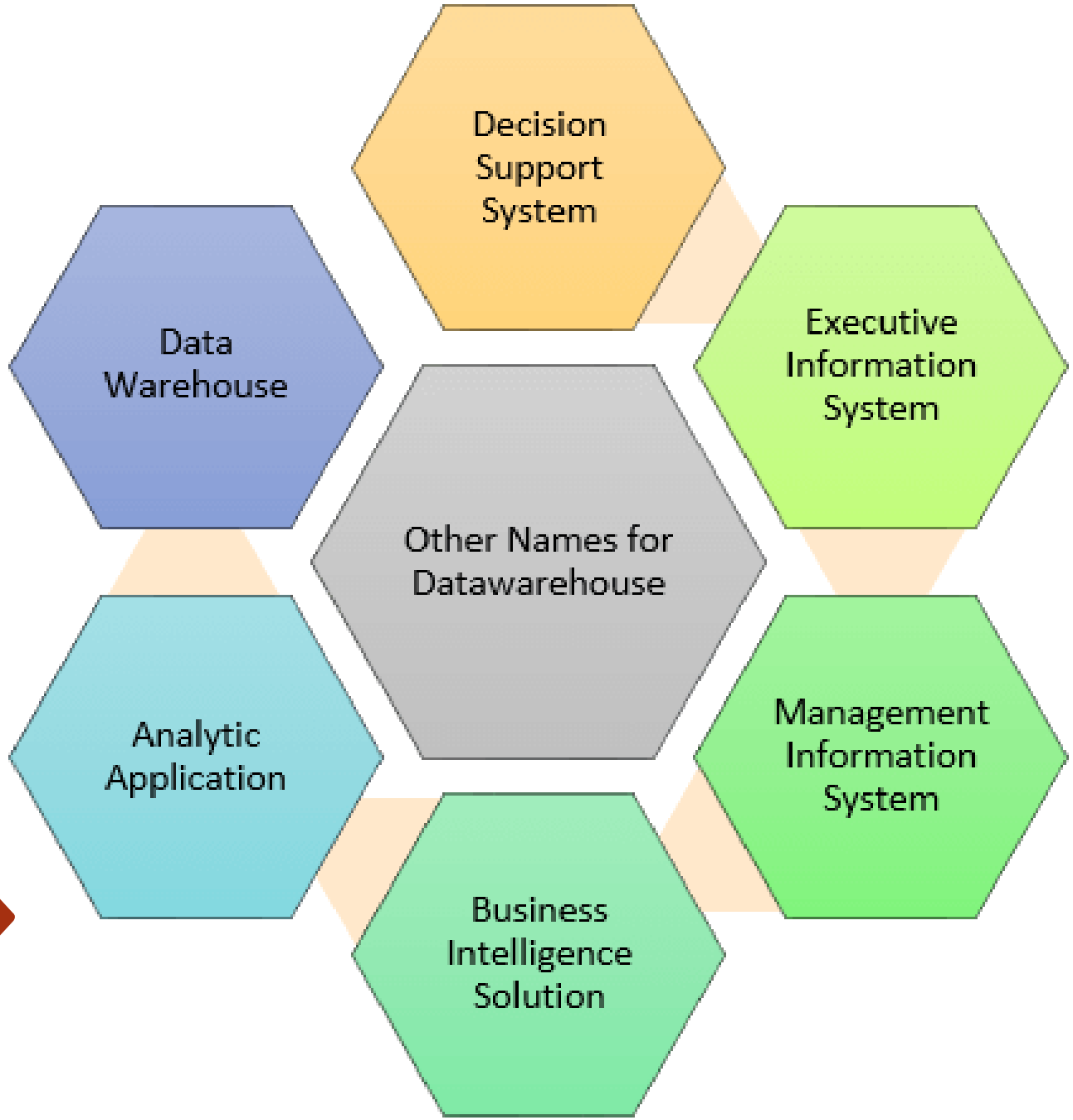
OLTP: *Online transaction processing information systems facilitates and manages transaction-oriented applications.*

OLAP: *Online analytical processing is an approach to answer multi-dimensional analytical queries swiftly.*

Basically, OLTP are the type of queries or operations executed on a database and OLAP are the type of queries executed in a data warehouse.



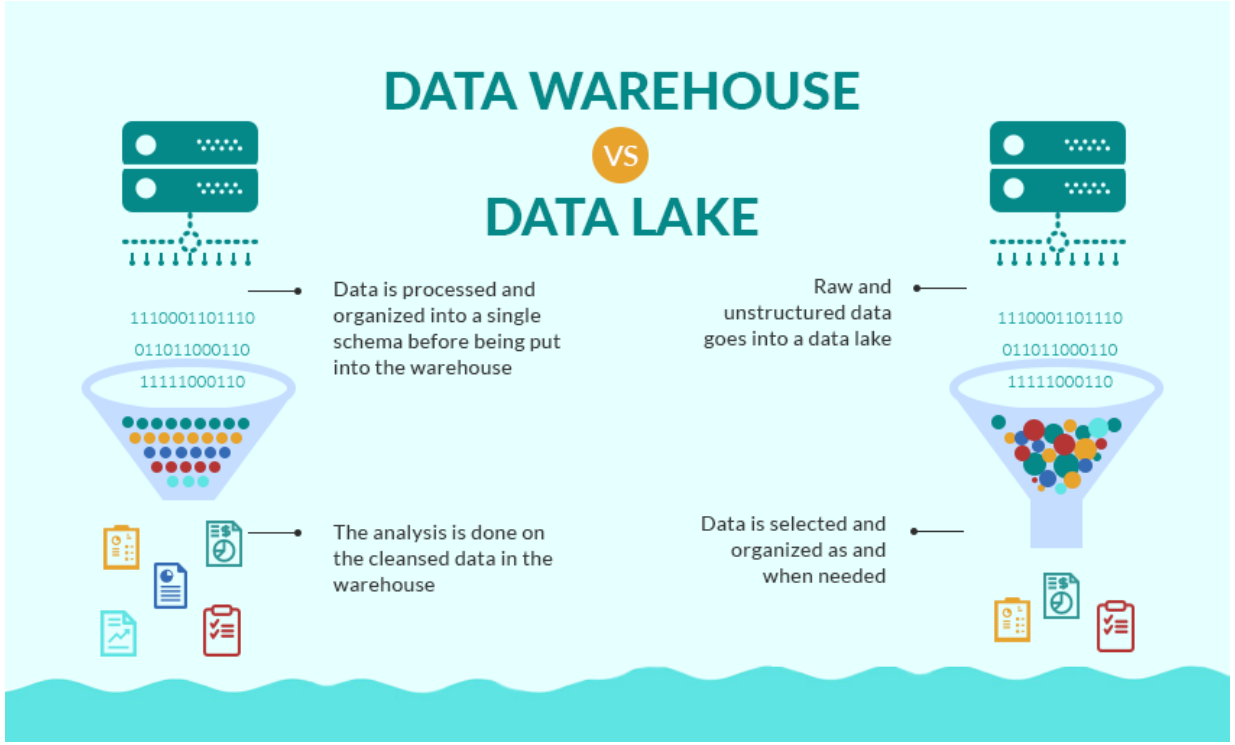
Data Warehouse



Data Lake

A **Data Lake** is a storage repository that can store a large amount of structured, semi-structured, and unstructured data.

It is a place to store every type of data in its native format with no fixed limits on account size or file.

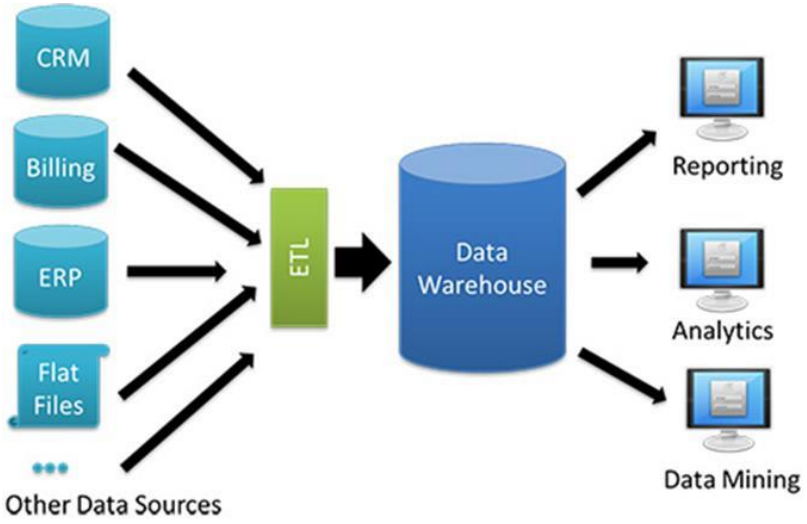


Data Warehouse

A **Data Warehousing** (DW) is process for collecting and managing data from varied sources to provide meaningful business insights.

A **Data warehouse** is typically used to connect and analyze business data from heterogeneous sources.

The data warehouse is the **core of the BI system** which is built for data analysis and reporting.



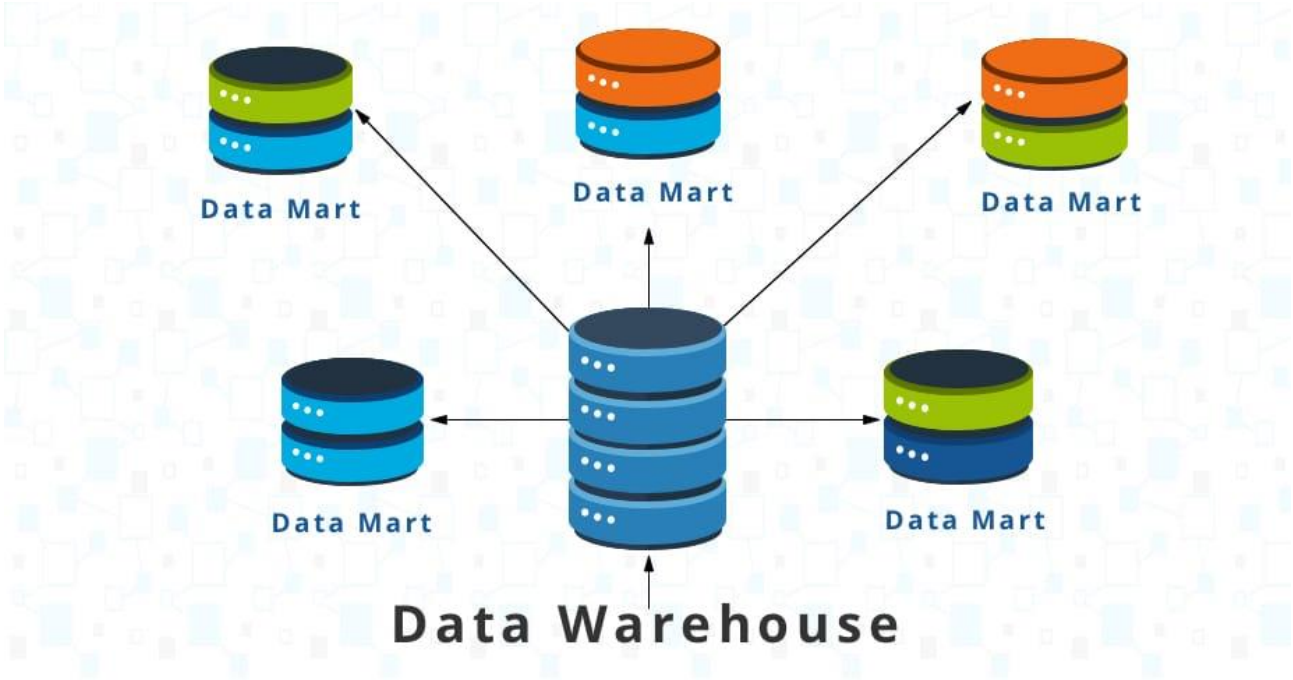
Data Lake vs Data Warehouse

- Data Lake stores all data irrespective of the source and its structure
- Data Lake is a storage repository that stores huge structured, semi-structured and unstructured data
- Data Lake defines the schema after data is stored
- Data Lake uses the ELT(Extract Load Transform) process

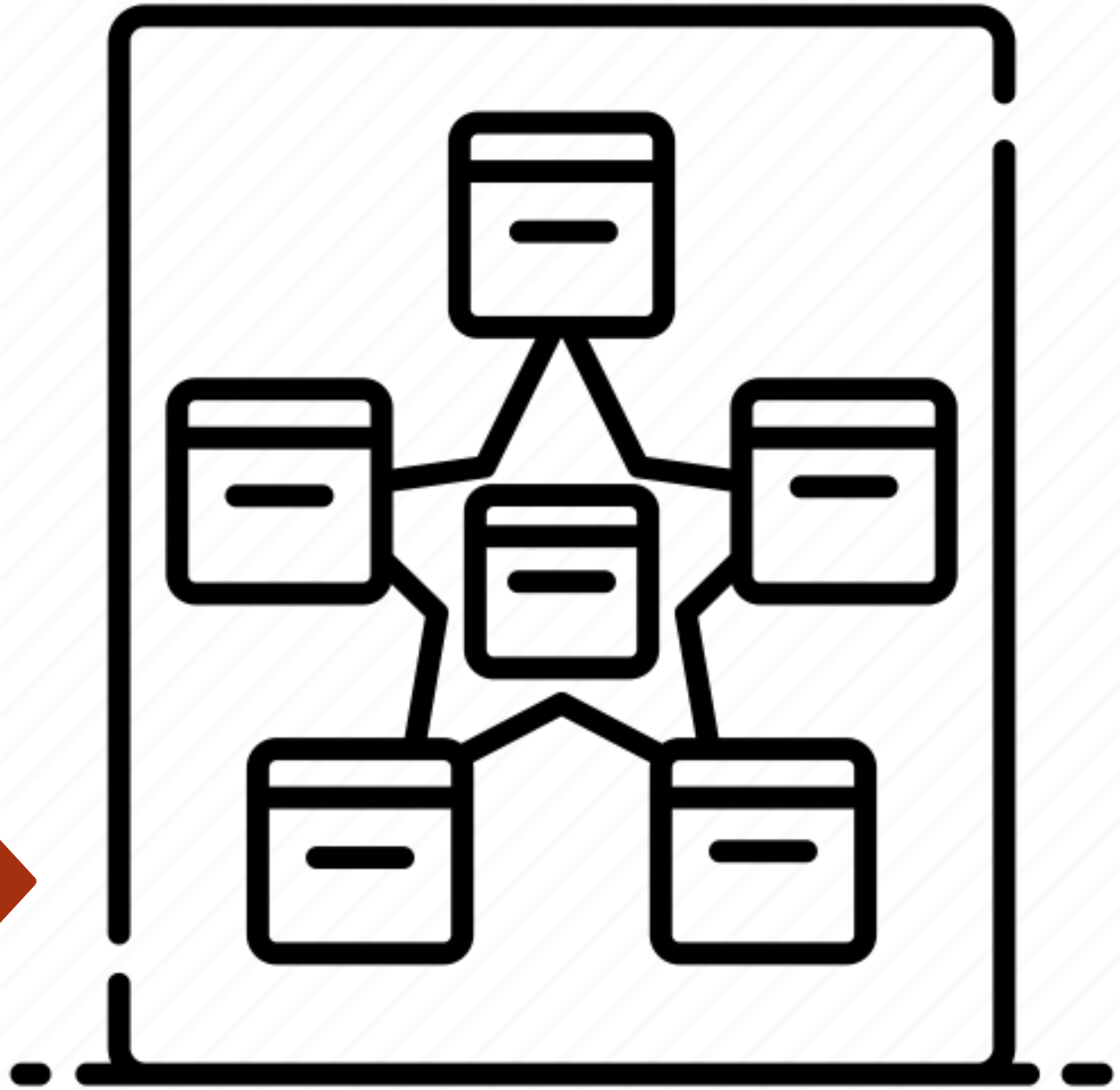
- Data Warehouse stores data in quantitative metrics with their attributes
- Data Warehouse is blending of technologies and component which allows the strategic use of data
- Data Warehouse defines the schema before data is store
- Data Warehouse uses ETL(Extract Transform Load) process

Data Mart

A **data mart** is a subset of the data warehouse. It specially designed for a particular line of business, such as sales, finance, sales or finance. In an independent data mart, data can collect directly from sources.



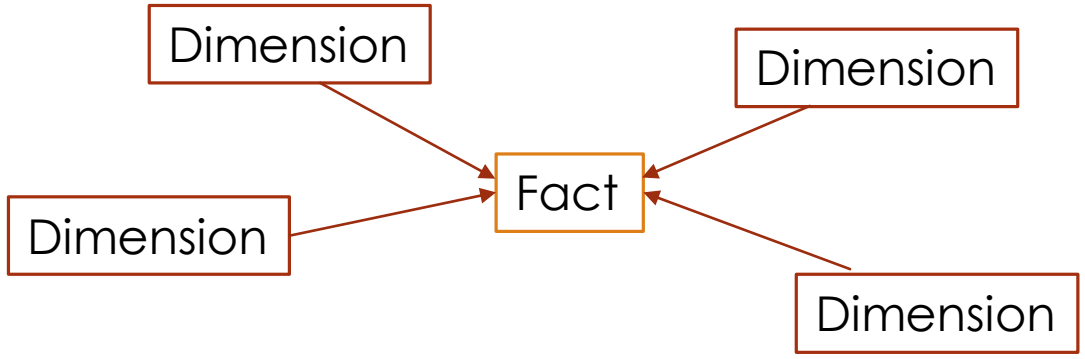
Dimensional Modeling



Dimensional Modeling

Dimensional Modeling (DM) is a data structure technique optimized for data storage in a Data warehouse. The purpose of dimensional modeling is to optimize the database for faster retrieval of data.

The concept of Dimensional Modelling was developed by Ralph Kimball and consists of "fact" and "dimension" tables.



Fact and Dimension

- **Fact**

Facts are the measurements/metrics or facts from your business process. For a Sales business process, a measurement would be quarterly sales number

- **Dimension**

Dimension provides the context surrounding a business process event. In simple terms, they give who, what, where of a fact. In the Sales business process, for the fact quarterly sales number, dimensions would be

- Who – Customer Names
- Where – Location
- What – Product Name

In other words, a dimension is a window to view information in the facts.

Fact and Dimension Table

Fact Table

A fact table is a primary table in dimension modelling.

A Fact Table contains

- Measurements/facts
- Foreign key to dimension table

Dimension Table

- A dimension table contains dimensions of a fact.
- They are joined to fact table via a foreign key.
- Dimension tables are de-normalized tables.
- The Dimension Attributes are the various columns in a dimension table
- Dimensions offers descriptive characteristics of the facts with the help of their attributes
- No set limit set for given for number of dimensions
- The dimension can also contain one or more hierarchical relationships

Dimensional Modeling Steps

1. **Identify the Business Process**
2. **Identify the Grain**

The CEO at an MNC wants to find the sales for specific products in different locations on a daily basis.

So, the grain is "product sale information by location by the day."

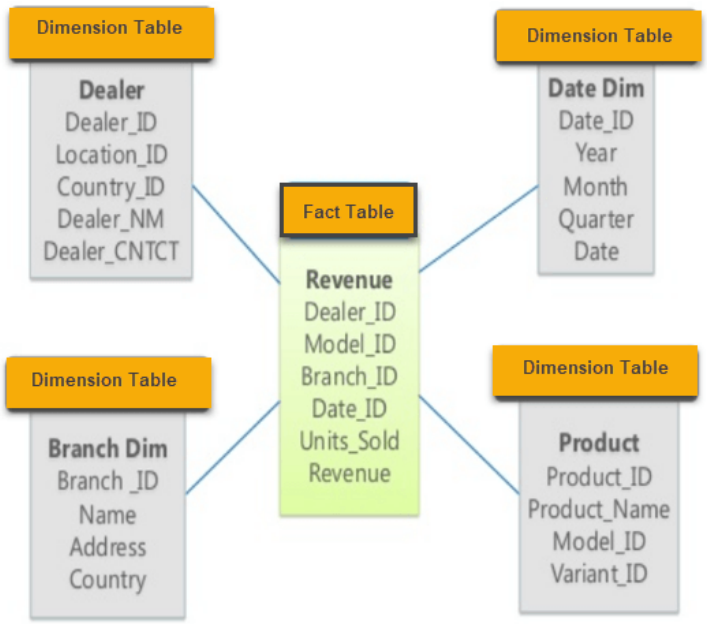
3. **Identify the Dimensions**
4. **Identify the Fact**
5. **Build Schema**

Dimensional Schema

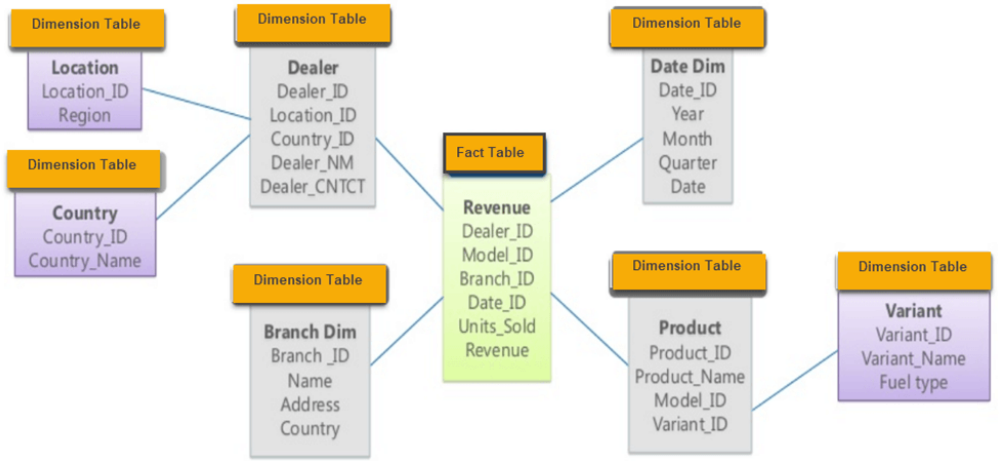
- **Star Schema** in data warehouse, in which the center of the star can have one fact table and a number of associated dimension tables. It is known as star schema as its structure resembles a star. The Star Schema data model is the simplest type of Data Warehouse schema. It is also known as Star Join Schema and is optimized for querying large data sets.
- **Snowflake Schema** in data warehouse is a logical arrangement of tables in a multidimensional database such that the ER Diagram resembles a snowflake shape. A Snowflake Schema is an extension of a Star Schema, and it adds additional dimensions. The dimension tables are normalized which splits data into additional tables.

Star Schema vs Snowflake Schema

Star Schema



Snowflake Schema



Star Schema vs Snowflake Schema

Star Schema	Snowflake Schema
It contains a fact table surrounded by dimension tables.	One fact table surrounded by dimension table which are in turn surrounded by dimension table
In a star schema, only single join creates the relationship between the fact table and any dimension tables.	A snowflake schema requires many joins to fetch the data.
Simple DB Design.	Very Complex DB Design.
Denormalized Data structure and query also run faster.	Normalized Data Structure.
Single Dimension table contains aggregated data.	Data Split into different Dimension Tables.

How to manage historical data?

Dimensions in data management and data warehouses contain relatively static data, but data from dimensions can change slowly over time and at unpredictable intervals.

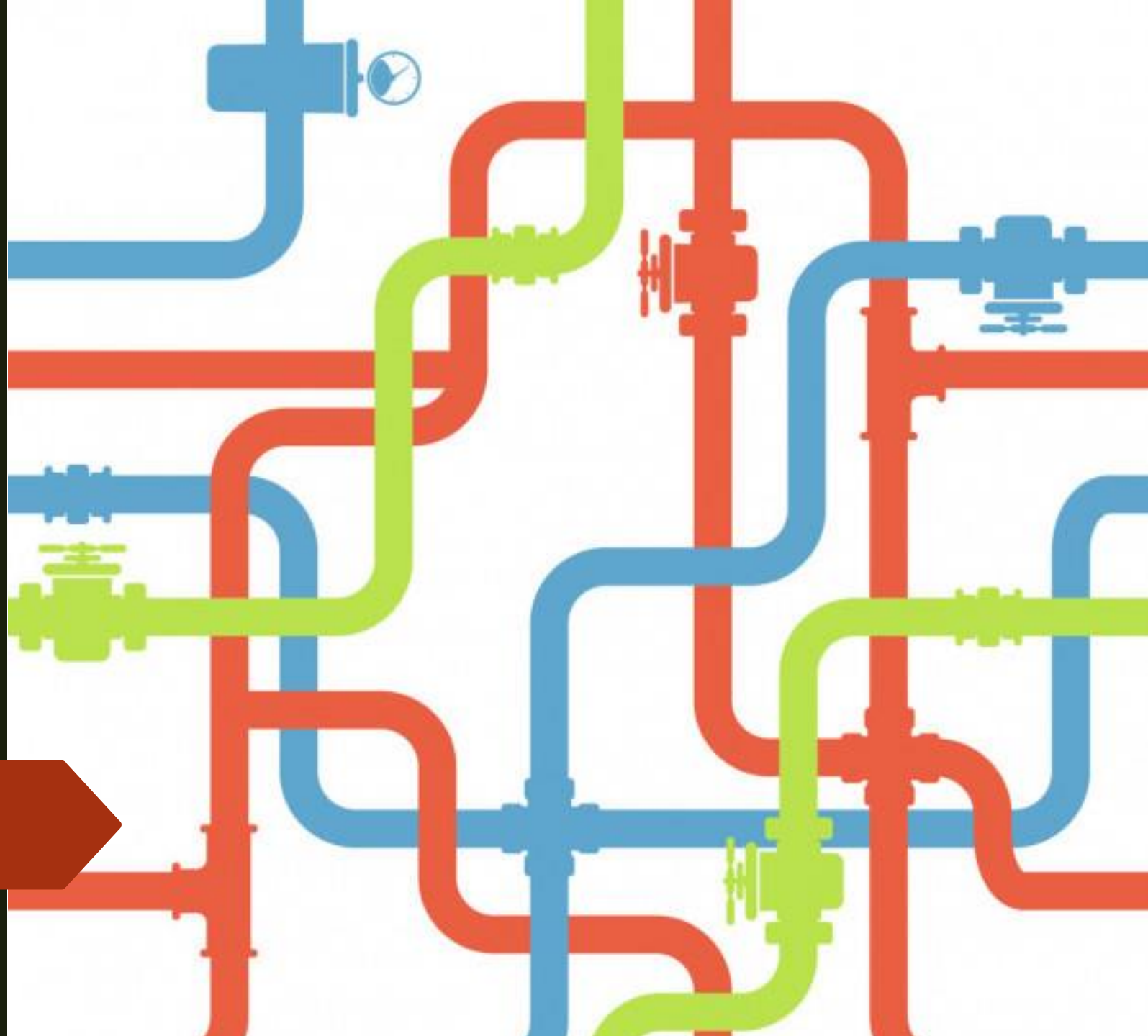
The type of data from dimensions can be called *Slowly Changing Dimension*.

The old value can be saved as a "history of change in value" of the attribute of a changing dimension. This old value store can be done by creating a column specifically for storing the old value.

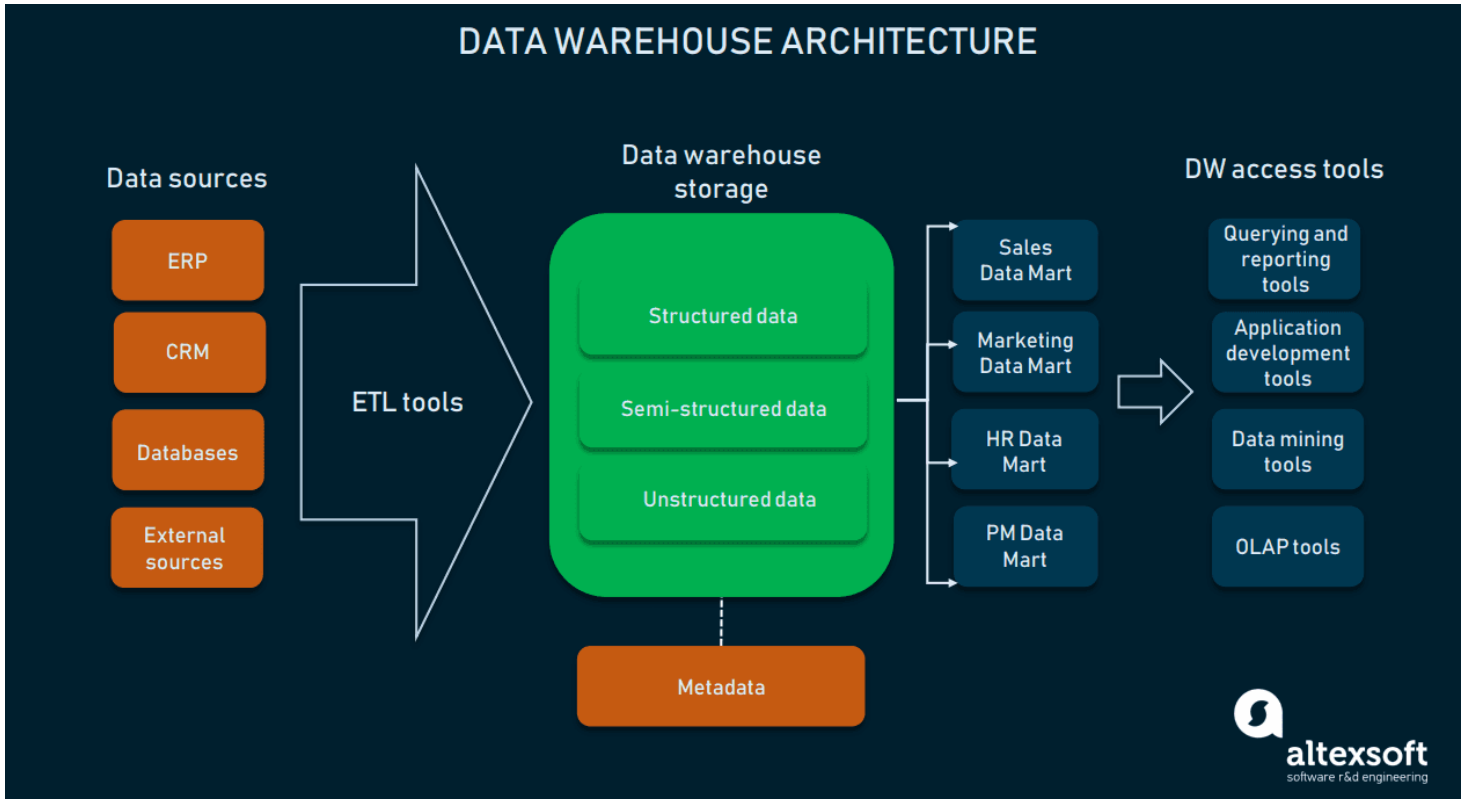
There are many approaches to the SCD types, but the most popular approaches to SCD are:

- Type 0: Passive
- Type 1: Overwrite (change value in the same row)
- Type 2: Create a new row for every changes

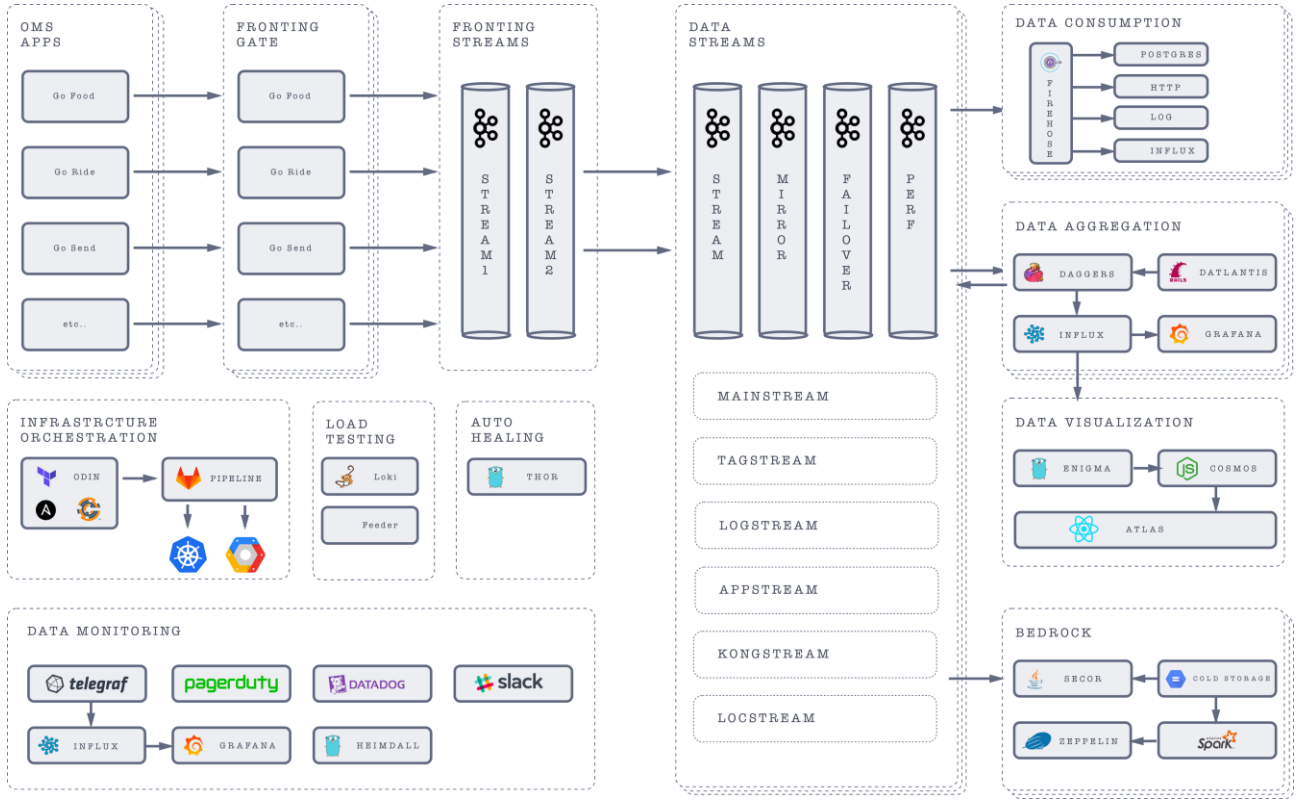
Data Pipeline



Data Pipeline



Data Pipeline



References

- https://www.guru99.com/data-modelling-conceptual-logical.html
- https://www.guru99.com/data-warehousing.html
- https://www.guru99.com/data-lake-vs-data-warehouse.html
- https://www.guru99.com/dimensional-model-data-warehouse.html
- https://www.guru99.com/star-snowflake-data-warehousing.html
- https://medium.com/@michelle.xie/explain-by-example-oltp-vs-olap-d5603ac2038b
- https://medium.com/@atriadplt/slowly-changing-dimension-whats-that-8ebf7cfef113