**Data Warehouse**

A subject-oriented, integrated, time-variant and non-volatile collection of data to support decision making process. It is a central repository that stores data from multiple sources and enables organizations to organize large volumes of historical data for efficient querying, analysis, and reporting.

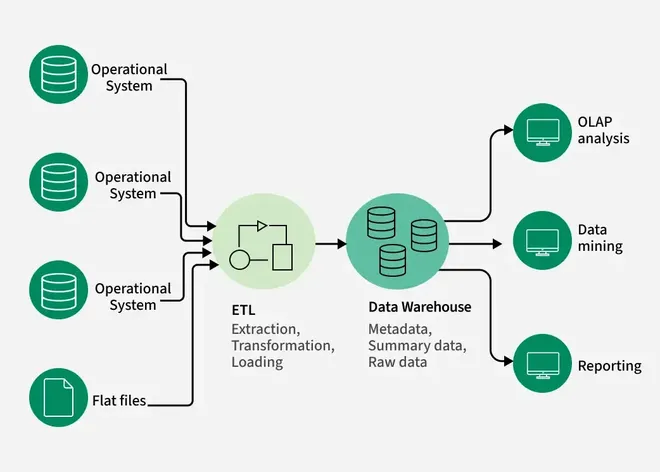
The main goal of data warehousing is to support decision-making by providing clean, consistent, and timely access to data. It ensures fast data retrieval even when working with massive datasets.

**Subject-oriented:** Focused on a specific business area (sales. Customers, finance, etc.)

**Integrated**: Gets the data from multiple sources

**Time-variant:** Ability to store historical data

**Non-volatile:** Once the data is stored in a data warehouse, it isn’t deleted or modified.



Data sources -> ETL -> Warehouse -> Application

**ETL**

* Which data to load from the source
* **Extract** : Identify the subset of data required from the source and pull it out without making any changes
* **Transform** : Perform modifications (correct format, transformations, standardization, cleaning, etc.) on the extracted data to make it suitable for analysis.
* **Load** : Insert the transformed data into its final destination, the target.

A diagram of a data architecture

AI-generated content may be incorrect.

A typical data warehouse doesn’t contain of only 1 source and target, there are multiple layers of the pipeline, where different functions are performed based on the requirements.

A diagram of etl method

AI-generated content may be incorrect.

**Extraction methods:**

* **Pull extraction**: Getting the data from the source by pulling the data
* **Push extraction**: Source system is pushing the data into the data warehouse

**Extraction types:**

* **Full extraction:** All the data from source is loaded to the warehouse in every single iteration.
* **Incremental extraction:** Identify the new, changeddata and load it to the warehouse in corresponding iterations.

**Extraction techniques:**

* **Manual Data Extraction**
* **Database Querying (**SQL query runs within a database to fetch data from source**)**
* **File Parsing**
* **API Calls**
* **CDC**
* **Event based streaming (**like in Kafka**)**
* **Web Scraping**

**Transformations:**

* Data enrichment
* Data integration
* Derived columns
* Data normalization & Standardization
* Business Rules & Logic
* Data Aggregation
* Data Cleansing
  + Remove duplicates
  + Data Filtering
  + Handle missing values
  + Handle invalid values
  + Handle unwanted spaces
  + Data type casting
  + Outlier detection

**Load**

**Processing types:**

* **Batch processing:** Load data in batches, one time load to refresh the data stored in data warehouse, scheduled action
* **Streaming processing:** Process changes to warehouse from source system` in near real time or as soon as possible

**Load methods:**

* Full load (Truncate and insert, Upsert, Drop Create Insert)
* Incremental load (Upsert, Append, Merge)

**ABOUT THE PROJECT:**

Extraction

* Pull extraction, full load, file parsing

Transformation

* All types of transformations

Load

* Batch processing, Full load (Truncate, insert), SCD1 (overwrite)

**EPIC/Stages/Phases of the Project –** A large task that needs a lot of effort and tracking to solve it

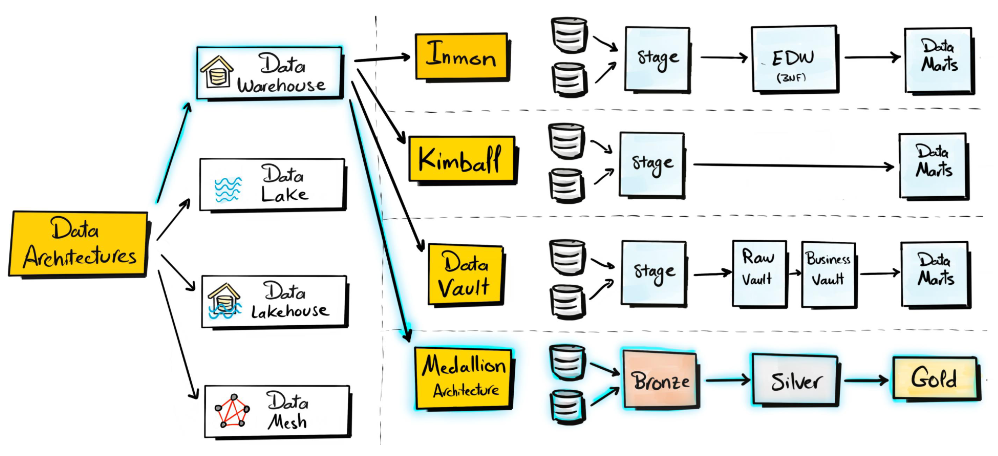
**Design Data Architecture**

**Choose the Correct Data Architecture**

* Data Warehouse
* Data Lake
* Data Lakehouse
* Data Mesh – Decentralized data management system

For this project, we are using Data Warehouse.

**Data Warehouse Design Approaches**

****

1. **Inmon Approach**

* Source
* Layer 1 (Staging)
* Layer 2 (Enterprise Data Warehouse, uses 3NF)
* Layer 3 (Data Marts, small subsets of DW designed to be readily consumed for reporting, topic-specific)
* BI tool (PowerBI, Tableau)

1. **Kimball Approach**

* Source
* Layer 1 (Staging)
* Layer 2 (Data Marts, small subsets of DW designed to be readily consumed for reporting, topic-specific)
* BI tool (PowerBI, Tableau)

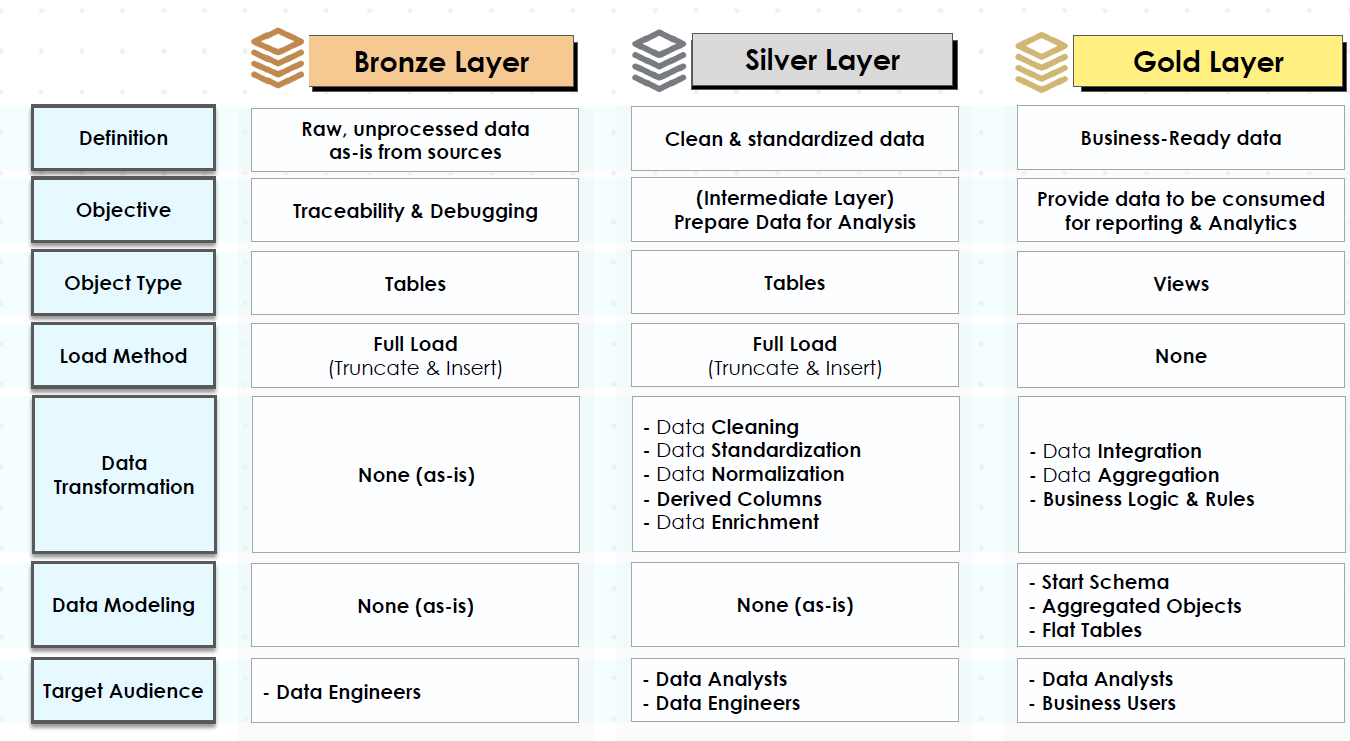
**(Tradeoff between speed and consistent data warehouse)**

1. **Data Vault Approach**

* Source
* Layer 1 (Staging)
* Layer 2 (Raw Vault, original data)
* Layer 3 (Business Vault, data that is governed by business rules and transformations)
* Layer 4 (Data Marts, small subsets of DW designed to be readily consumed for reporting, topic-specific)
* BI tool (PowerBI, Tableau)

1. **Medallion Architecture**

* Source
* Layer 1 (Bronze, original data)
* Layer 2 (Silver, perform data cleansing and transformations, business rules not applied)
* Layer 3 (Gold, similar to Data marts, build different types of business ready objects for reporting, ML, and AI served as data products)
* BI tool (PowerBI, Tableau). Reporting, ML and AI



**ARCHITECTING THE BRONZE LAYER**

**Analyzing source systems**

1. **Business Context and Ownership**

* Who **owns** the **data**?
* What **business process** it supports?
* System and Data **documentation**
* **Data Model** and **Data Catalog**

1. **Architecture and Tech Stack**

* How is Data **Stored**? (Om-prem or Cloud)
* What are the **integration capabilities**? (API, Kafka, File extraction, Direct DB Connection, etc.)

1. **Extract and Load**

* **Increment** vs **Full** Load
* Data **Scope** and **Historical** Needs
* Expected **size** of the **extracts**
* Data **volume limitations**
* How to **avoid impacting** the source system’s **performance**?
* **Authentication** and **Authorization** (tokens, SSH keys, VPN, IP whitelisting)

Data Profiling – Explore the data to identify column names and data types.

**Data Modeling**

The process of taking the raw data and organizing and structuring into a clean, meaningful and friendly way that is easy to understand.

**Stages:**

**Conceptual data model** (big picture) – focuses on entity

**Logical data model** (blueprint) – describes the attributes of an entity and relationships between the entities

**Physical data model** (implementation) – Schema and data types, how data is going to be stored

**Star vs Snowflake schema**

* **Star schema**

Central fact table (contains transactions, events, etc.) surrounded by supporting dimensional tables (contains descriptive information)

Relationships between the fact and dimensional tables form a star, hence the name.

Simpler and easy, easy to query

Big dimensions, can have duplicates and grow more and more overtime.

* **Snowflake schema**

Central fact table (contains transactions, events, etc.) surrounded by supporting dimensional tables (contains descriptive information).

The dimension tables are divided further into sub-dimensional tables, which resemble the shape of a snowflake.

More complex, harder to query, normalizations (redundancies broken down further into smaller tables to optimize storage). Useful for large datasets.

A diagram of different types of snowflakes

AI-generated content may be incorrect.

**Dimension vs Fact Tables**

* **Dimension**

Descriptive information that gives context to your data (Who? What? Where?)

* **Fact**

Quantitative information that represents events

Contains multiple products, date of transactions, measures and number (How much? How many?)

**Surrogate key –** System generated unique identifier assigned to each record in a table (substitute for primary key from source systems)