Project Requirements

**Building the Data Warehouse (Data Engineering)**

Objective:

Develop a modern data warehouse using SQL server to consolidate sales data, enabling analytical reporting and informed decision-making.

Specifications

-Data Sources: Import data from two source systems (ERP and CRM) provided as CSV files.

-Data Quality: Cleanse and resolve data quality issues prior to analysis.

-Integration: Combine both sources into a single, user-friendly data model designed for analytical queries

-Scope: Focus on the latest dataset only; historization of data is not required.

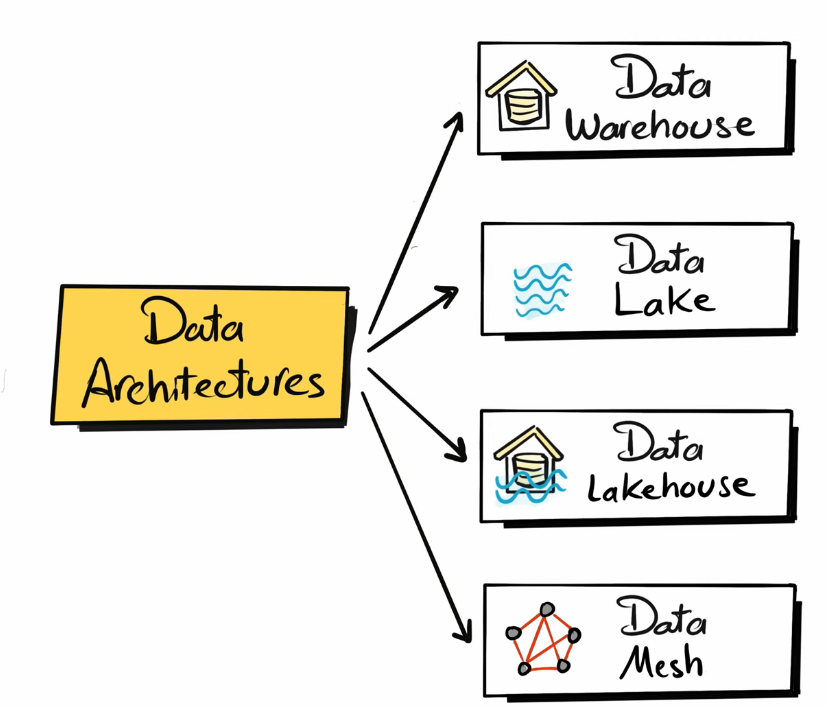
-Documentation: Provide clear documentation of the data model to support both business stakeholders and analytics teams.

**BI: Analytics & Reporting (Data Analysis)**

Objective:

Design The Data Architecture

**Choose the Right Approach**



**1. Data Warehouse**

* **Definition:** Centralized, structured repository optimized for reporting and analytics (historical data, BI).
* **When used:** For business intelligence (dashboards, KPIs, reporting).
* **Pros:** High performance for queries, clean structured data, trusted single source of truth.
* **Cons:** Expensive at scale, rigid schema, not ideal for unstructured data.

**2. Data Lake**

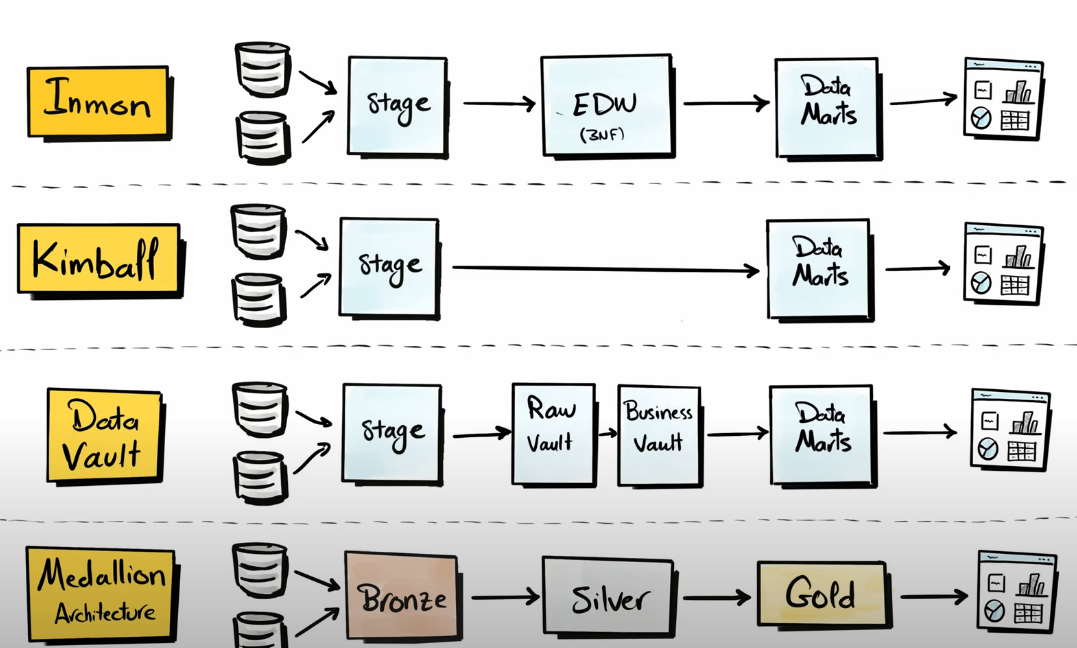
* **Definition:** Storage system that holds raw, structured, semi-structured, and unstructured data.
* **When used:** For big data analytics, ML/AI, storing diverse data at scale.
* **Pros:** Low-cost storage, handles all data types, flexible schema-on-read.
* **Cons:** Can turn into a “data swamp” if unmanaged, slower queries, less governance.

**3. Data Lakehouse**

* **Definition:** Hybrid of data lake and data warehouse—stores raw data but supports structured querying.
* **When used:** When you need both data science (unstructured/ML) and BI analytics in one place.
* **Pros:** Unified platform, cost-efficient, supports structured + unstructured, avoids data duplication.
* **Cons:** Newer technology, evolving tools, can be complex to implement.

**4. Data Mesh**

* **Definition:** Decentralized data architecture where domain teams own and share their data as products.
* **When used:** Large organizations with many domains, needing scalability and autonomy.
* **Pros:** Scales well across teams, fosters ownership, avoids central bottlenecks.
* **Cons:** Requires cultural shift, governance complexity, risk of inconsistency across domains.

****

**1. Inmon (Top-Down)**

* **Definition:** Enterprise Data Warehouse (EDW) in 3NF → Data Marts.
* **When used:** Large enterprises needing centralized, integrated data.
* **Pros:** Strong governance, consistent data.
* **Cons:** Complex, slow to deliver business insights.

**2. Kimball (Bottom-Up)**

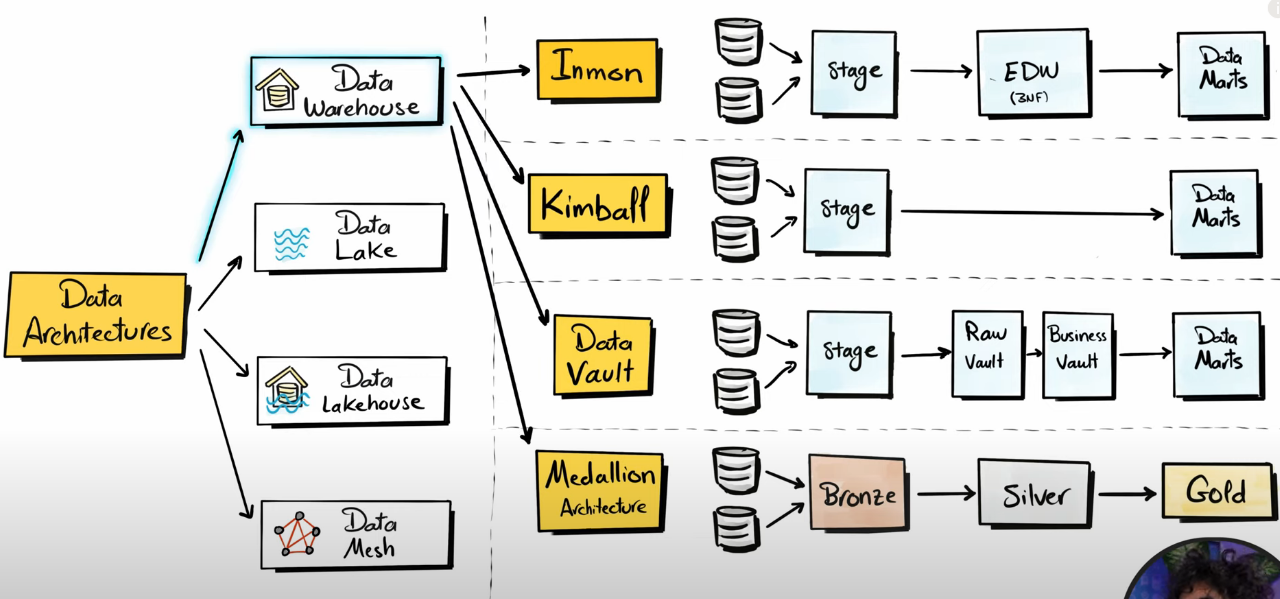
* **Definition:** Build Data Marts first → integrated via dimensional models.
* **When used:** Faster BI and analytics delivery.
* **Pros:** Quick results, business-friendly.
* **Cons:** Risk of data silos, harder long-term integration.

**3. Data Vault**

* **Definition:** Raw vault (raw data) + business vault (business rules) → Data Marts.
* **When used:** Complex, evolving environments with changing rules.
* **Pros:** Scalable, flexible, audit-friendly.
* **Cons:** More layers, complex queries, higher storage.

**4. Medallion Architecture**

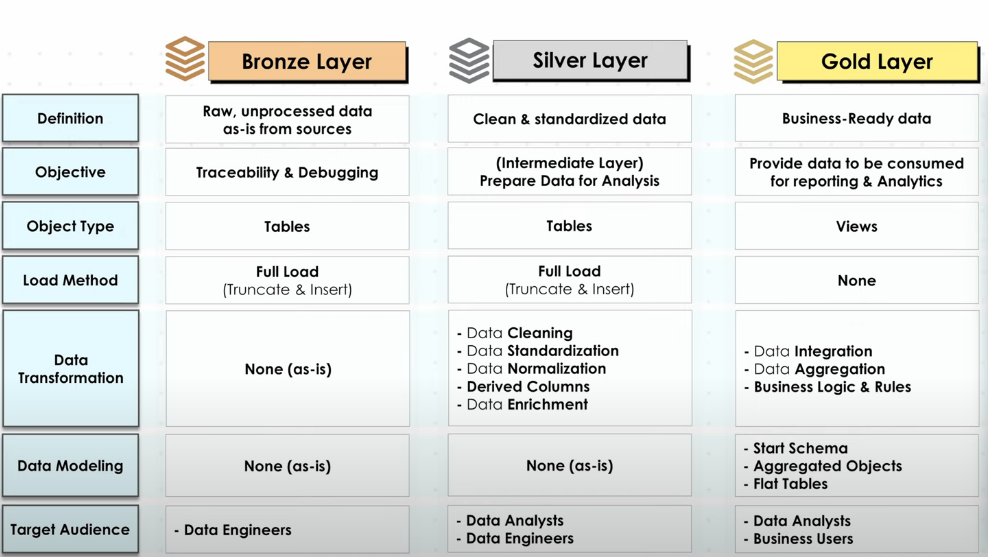
* **Definition:** Bronze (raw) → Silver (cleaned) → Gold (curated).
* **When used:** Cloud & big data pipelines (e.g., Databricks).
* **Pros:** Simple, modern, supports streaming + batch.
* **Cons:** Still evolving, less standardized than warehouse methods.



This diagram shows how different **data architectures** connect to their **modeling approaches and processing flows**:

* **Data Warehouse** typically uses **Inmon, Kimball, or Data Vault** approaches.
  + *Inmon:* Data → Stage → Enterprise Data Warehouse (3NF) → Data Marts.
  + *Kimball:* Data → Stage → Data Marts directly (dimensional model).
  + *Data Vault:* Data → Stage → Raw Vault → Business Vault → Data Marts.
* **Data Lakehouse** often applies the **Medallion Architecture:** Data → Bronze (raw) → Silver (clean/structured) → Gold (curated for analytics).
* **Data Lake** and **Data Mesh** are other architectures, but here the focus is on how Warehouse and Lakehouse get modeled for analytics delivery.

👉 In short: raw data flows through different structured layers depending on the chosen architecture, ending in **Data Marts/Gold Layer** for business reporting and analytics.



Separation of Concern (SoC)

These two diagrams explain **Separation of Concerns (SoC)** in data systems:

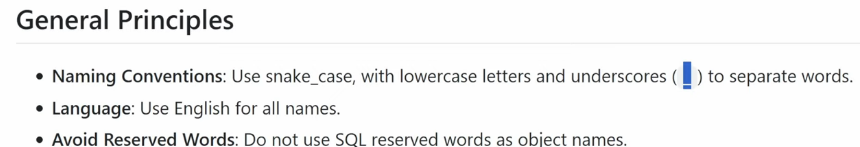
* **Without SoC:** everything is tangled together, making it hard to manage, scale, or debug.
* **With SoC:** data is broken into independent **modules (A, B, C)**, each owning its logic, making the system cleaner and easier to maintain.

When combined with the **Medallion Architecture (Bronze → Silver → Gold):**

* **Bronze:** raw ingested data.
* **Silver:** cleaned and standardized data.
* **Gold:** curated, business-ready data.

👉 Together, SoC + Medallion = modular, layered data pipelines that are easier to scale and govern.

Define Naming Conventions: Set of Rules or Guidelines for naming anything in the project



Different styles of naming: camelCase, UpperCamel(aka PascalCase), or snake\_case

