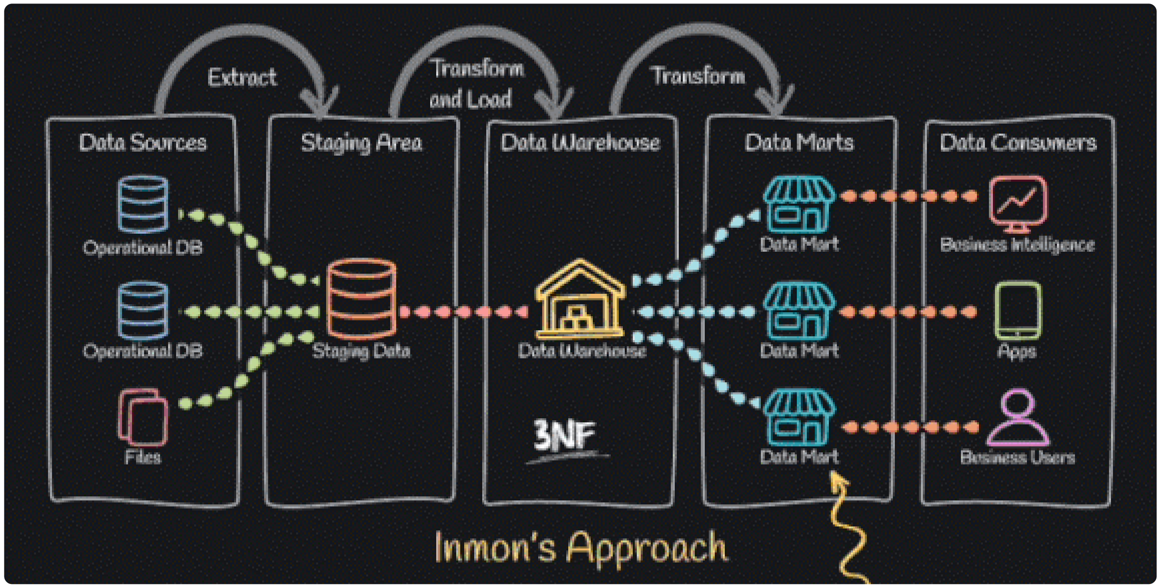
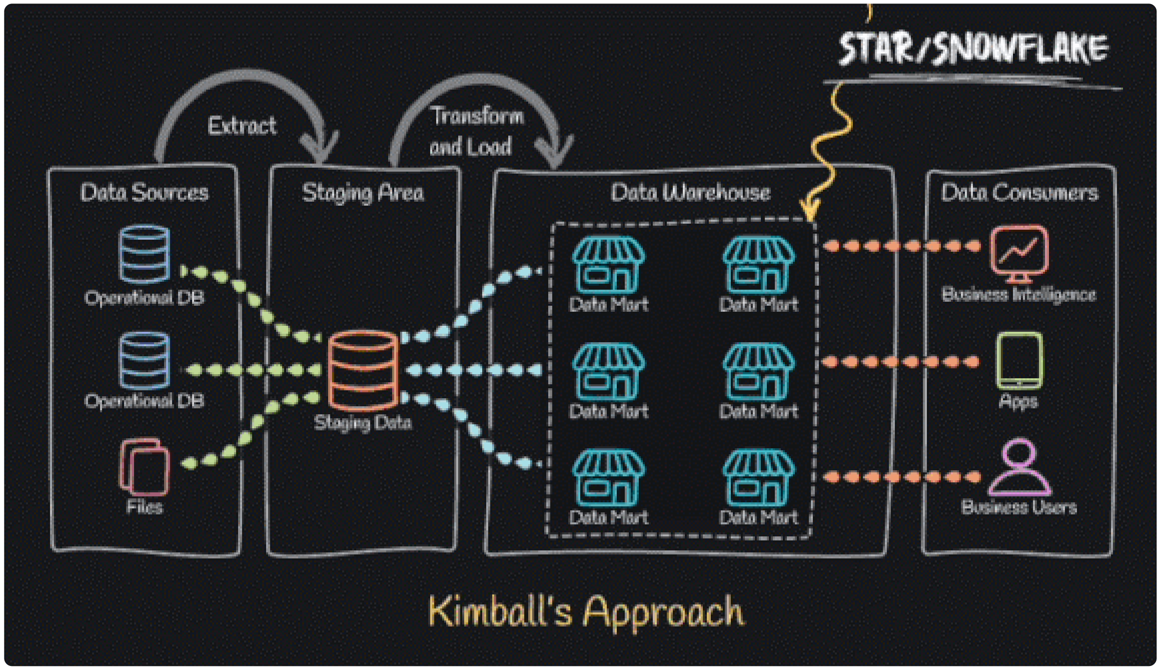
Kimball's architecture VS Inmon's architecture

1. **Inmon Model: EDW Approach (Top-Down)**



- **Description**: Inmon's **Enterprise Data Warehouse (EDW)** approach focuses on building a centralized, integrated data warehouse first. Data from various sources are cleaned, integrated, and stored in a normalized structure (usually 3NF). Once the centralized warehouse is in place, data marts can be built on top of it.
- **Advantages**:
  - **Single source of truth** for the entire organization.
  - **Centralized data** ensures consistency and integrity across departments.
- **Disadvantages**:
  - Slower to implement because the entire warehouse must be developed before any reports or analysis can be done.
  - More complex to manage and maintain, especially as the volume of data grows.

2. **Kimball Model: Data Mart Approach (Bottom-Up)**



- **Description**: Kimball's approach focuses on building **data marts** first, which are smaller, department-specific data repositories. These marts use **star** or **snowflake schemas** and are built in a denormalized structure to optimize for reporting and query performance. Over time, the data marts are integrated to form a larger data warehouse.
- **Advantages**:
  - **Faster to implement** since data marts can be developed and deployed independently.
  - **Optimized for business intelligence and reporting**, as data marts are designed for specific business needs.
- **Disadvantages**:
  - Can lead to **data redundancy** if integration across marts is not managed properly.
  - Less control over data consistency, as each data mart might have its own version of the data.

Which Model is Best?

- **There is no one-size-fits-all strategy** for a data warehouse, as the best approach depends on the organization’s size, complexity, and needs.

- **Inmon’s model** is typically preferred in large enterprises where data consistency and a centralized, integrated view are priorities.
- **Kimball’s model** is often better suited for organizations that need faster reporting, more flexibility, and departmental autonomy.

Data Warehouse vs Data Lake

Data Warehouse:

- **Data Structure**: Stores **structured data** that has been processed, cleaned, and transformed. It uses predefined schemas (like star or snowflake) and is optimized for reporting and querying.
- **Data Processing**: Data undergoes the **ETL process** (Extract, Transform, Load), where it is cleaned, validated, and structured before being loaded.
- **Use Cases**: Best for **business intelligence (BI)**, **reporting**, **analytics**, and **historical data analysis**. It supports decision-making processes by providing clean, accurate data.
- **Performance**: Optimized for **high performance** in querying and reporting, providing fast insights on structured data.
- **Storage Cost**: Generally higher because it uses **high-performance storage** systems designed for optimized queries.

Data Lake:

- **Data Structure**: Stores **raw, unstructured, semi-structured, and structured data**. Data is stored in its **native format**, which can include text files, images, logs, social media posts, sensor data, etc.
- **Data Processing**: Data is stored with minimal processing, allowing for **schema-on-read**, meaning the schema is applied only when the data is read for analysis.
- **Use Cases**: Ideal for **big data**, **machine learning**, **data science**, and handling **large volumes of diverse data**. It's used to store data for future analysis, often without knowing the exact use cases upfront.

- **Performance**: Typically slower for querying, as the data is unprocessed and may require complex transformations before analysis.
- **Storage Cost**: More cost-effective because it utilizes **cheap, scalable storage** solutions, often in the cloud, that are designed to handle large amounts of raw data.