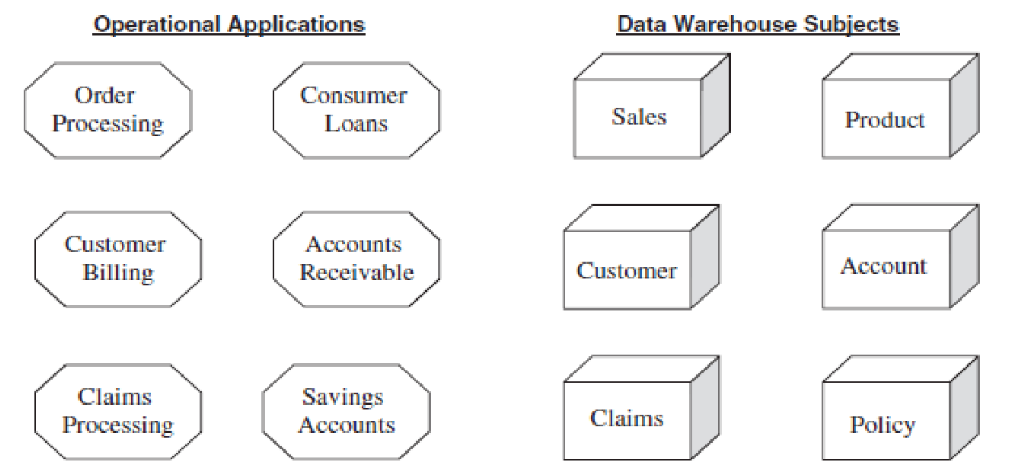
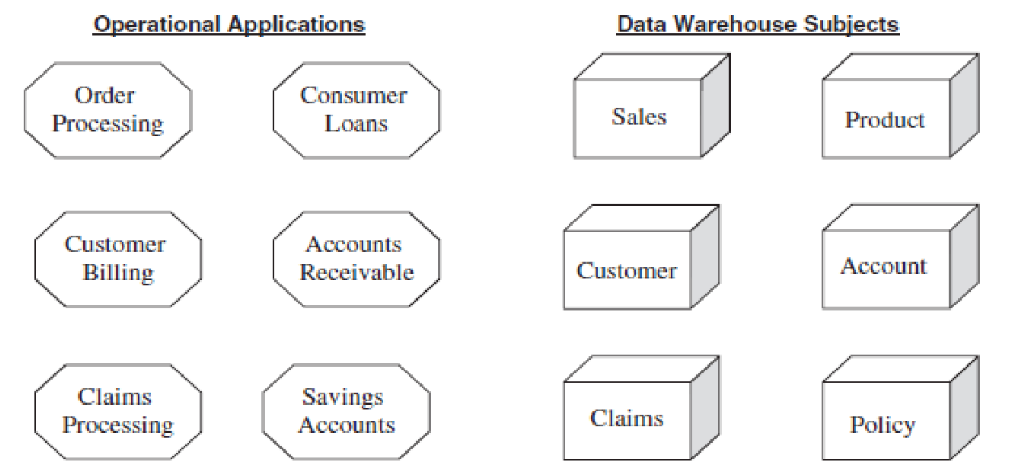
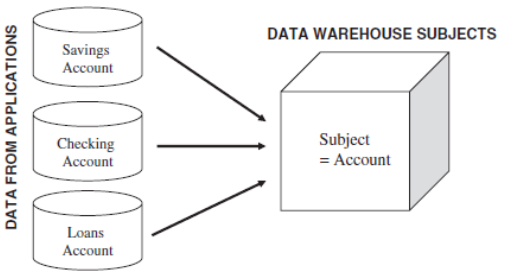
**Data Warehouse Features:-**

1. *Subject Oriented Data:*

Data is organized based on business subjects (entities important to company's operations and decisions) rather than the operational functions of the business. They can vary per organization based on their specific needs and processes.

Data in a data warehouse isn't stuck in separate apps. Instead, it's organized around key business areas like customers and sales. This setup gives a complete picture of the business.



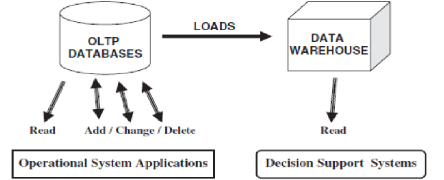
1. *Integrated Data:*

Data from diff places (db systems, platforms, external sources, etc.) might look diff varying in how it's labeled, formatted, and presented. It undergoes transformation, consolidation, and integration to ensure consistency and standardization (to understand easily) before being stored in DW.

Standardization: making sure names, codes, and measurements are all the same across the board for smooth analysis and reporting.

1. *Time-Variant Data:*

It includes historical data rather than just current values, which is crucial for decision-making as it allows businesses to analyze past trends, performance, and behaviors to make informed decisions. Data in a DW is stored as snapshots over different periods and is time-stamped to indicate when it was captured. This enables users to analyze past data, relate it to current trends, and make forecasts for the future based on historical patterns.

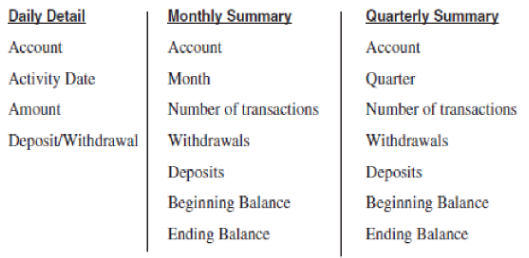


1. *Nonvolatile Data:*

Data in a DW doesn't change as often as in operational databases. It's transferred from operational systems to the DW based on business needs. The frequency of this transfer varies, depending on how important the data is. Unlike operational dbs, where data is frequently updated or deleted, data in a DW usually stays same once it's stored. This allows for more focus on analyzing & reporting rather than constantly updating transactions.

**Data granularity** refers to how detailed data is. The finer the granularity, the more detailed the data. When analyzing data, you start at a higher level and can then break it down into more detailed summaries as needed. The level of granularity is determined based on the types of data and the expected performance for queries. Lower granularity means more data stored in DW. In a banking DW, there are typically three levels of data granularity:

1. *Daily Detail:* Includes data on individual transactions made each day, such as account number, activity date, amount, and type of transaction (deposit/withdrawal).
2. *Monthly Summary:* Summarizes data monthly, including account number, month, number of transactions, withdrawals, deposits, beginning balance, & ending balance.
3. *Quarterly Summary:* Provides data for each quarter-account no, quarter, no. of transactions, withdrawals, deposits, beginning balance, & ending balance.



These different levels of granularity allow analysts to view data at varying levels of detail, depending on the specific needs of their analysis.

**Top-Down Approach (Bill Inmon):**

Building a centralized DW containing the lowest level of granular data. Data marts, which are subsets of the DW, are then created to serve specific business departments or functions.

*Advantages:*

* Offers a complete view of company data.
* Provides a well-organized structure.
* Centralizes data storage and control.
* Can show results quickly with step-by-step implementation.

*Disadvantages:*

* Takes time to build due to the step-by-step process.
* High risk if not done right.
* Needs expertise across different areas.
* Requires a big investment without prior testing.

**Bottom-Up Approach (Ralph Kimball):**

Creation of individual data marts, each containing detailed and summary data. These data marts are designed based on a dimensional data model. Over time, these data marts are integrated into a corporate DW by conforming (comply with rules) dimensions.

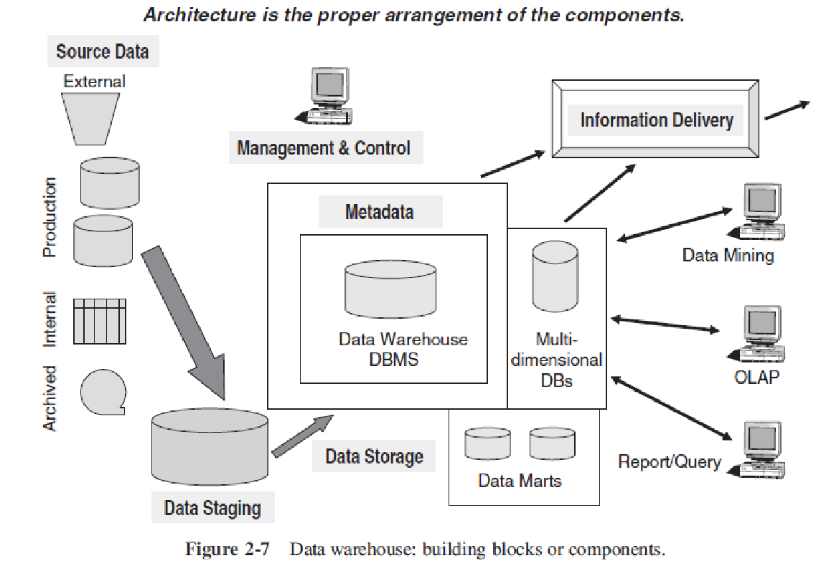
*Advantages:*

* Easier and quicker to set up.
* Usually brings good returns and proves its value early.
* Less risky because it's built step by step.
* Teams can learn and enhance their skills over time.

*Disadvantages:*

* Provides a narrower view of data compared to the top-down approach.
* Can create duplicate data across different data marts.
* Risks having data that doesn't match up if not managed well.
* Handling connections between data marts can get complex as they grow.

**Data Warehouse Architecture**



1. *Source Data Component:*

Production Data: Comes from operational systems within the organization.

Internal Data: Information generated internally within the organization.

Archived Data: Historical data that's stored for reference.

External Data: Information sourced from external sources related to the organization.

1. *Data Staging Component:*

Data Extraction: Gathering data from various sources.

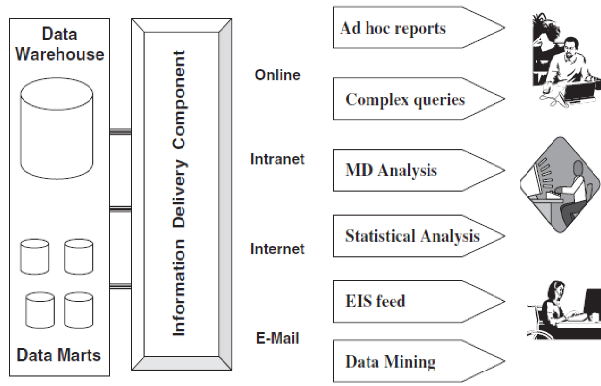
Data Transformation: Changing the format or structure of the data.

Data Loading: Putting the transformed data into the warehouse for storage.

1. *Data Storage Component:*

Data Repositories: Stores the data, typically in read-only format.

Multi-Dimensional Databases: Optional; used for specialized analysis.

1. *Information Delivery Component:*

DW: Main storage for data retrieval.

Ad Hoc Reports: Customized reports generated as needed.

Online Queries: Real-time access to data for immediate insights.

Complex Queries: Adv analysis of data.

Intranet/Internet: Access to data through internal or external networks.

Statistical Analysis: Using data for statistical purposes.

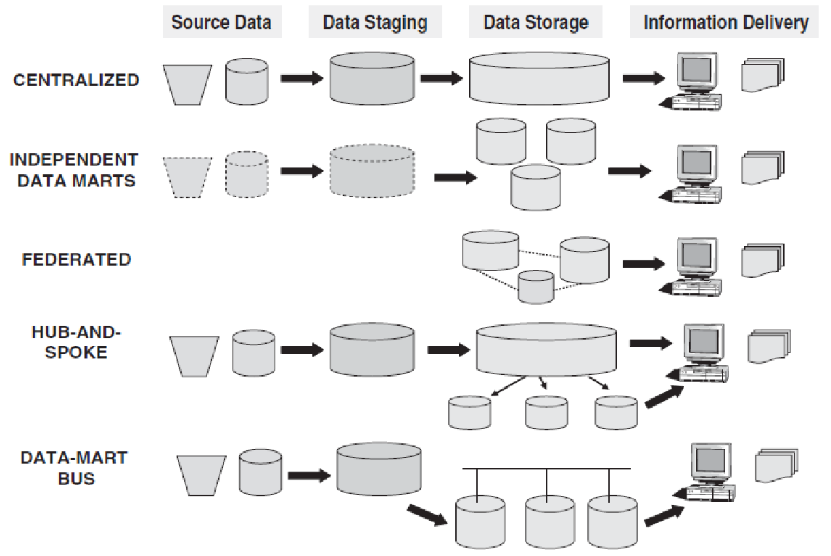
Executive Information Systems (EIS) feed: Data tailored for senior executives.

1. *Metadata Component:*

Similar to a data dictionary or catalog in traditional databases.

Contains information about the data stored in the warehouse, like its structure and meaning.

**Data Warehouse Architectural Types**



1. *Centralized Data Warehouse:*

Serves enterprise-wide needs (whole company).

Stores data in third normal form (neatly but w/o separate sections).

Lacks separate data marts; all queries access centralized data.

1. *Independent Data Marts:*

Each unit creates its own data marts.

They're separate and might not match or may have inconsistent data.

Analyzing data across them is challenging. (It's tough to compare data between them.)

1. *Federated:*

Integrates existing structures like operational systems. (Combines old systems together)

Uses shared fields, global metadata, rules, etc.

No single data warehouse; integration occurs logically or physically.

1. *Hub-and-Spoke:*

Follows Inmon's CIF approach (starts big).

Central DW holds atomic data (main data).

Branches (data marts) come from it, focusing on different needs.

1. *Data-Mart Bus:*

Follows Kimball's approach (starts small).

Starts with specific business subjects (each subject has its gets4 area).

Builds interconnected data marts with shared dimensions for an enterprise view.

**Types of Metadata**

1. *Operational Metadata:*

Contains details about the original operational data sources.

Includes information about how data is manipulated before it reaches the data warehouse.

When users access warehouse data, they can trace it back to its original source.

1. *Extraction and Transformation Metadata:*

Holds information about how data is extracted from source systems.

Covers extraction frequencies, methods, and any business rules applied during extraction.

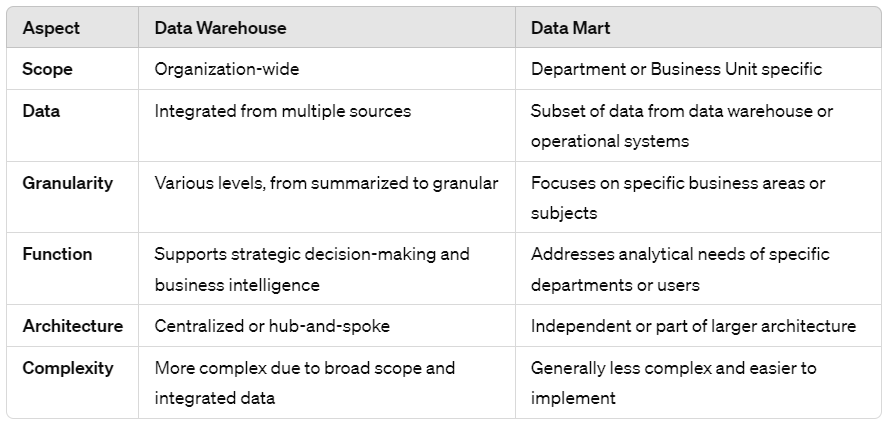
Tracks all the changes made to the data before it's stored in the warehouse.

1. *End-User Metadata:*

Acts like a map for users navigating the data warehouse.

Helps users locate specific information.

Allows users to search using familiar business terms and concepts.



**DW design strategies:**

*Dimensional Modeling:* Organizes data into dimensions and measures for efficient querying.

*ETL Processes:* Extracts, transforms, and loads data into the warehouse, ensuring consistency and accuracy.

*Loading:* Updates warehouse with changes since last load to save time & resources.

*Partitioning:* Large→smaller tables based on criteria like date ranges for faster querying.

*Indexing:* Creates indexes on frequently queried columns to speed up data retrieval.

*Aggregation:* Pre-calculates summary statistics to improve query performance.

*Data Compression:* Reduces storage requirements by compressing data.

*Data Governance:* Establishes policies for managing data quality, security, and compliance.

*Scalability and Performance:* Designs the warehouse to scale and optimizes performance through hardware upgrades and query tuning.

