**DATA WAREHOUSING**

**CHAPTER 2**

**DEFINE**:

Data warehouse is the large and centralized system use to store , manage , organize data collected from multiple sources . it is designed to support business analysis , decision making and reporting by providing clean , consistent and historical data in one place .

**NEED OF DATA WAREHOUSING**

1. **Centralized Data Storage System**  
   Data warehousing consolidates data from multiple, often disparate sources into a single, unified repository. This centralization simplifies data management and ensures consistency across departments and business units.
2. **Enhanced Data Visibility (Enhanced Vision)**  
   A data warehouse provides a transparency between the organization and the stake holders by showing the organization’s operations. This visibility helps stakeholders understand the bigger picture, enabling better strategic decision-making.
3. **Improved Analytical Capabilities**  
   By organizing historical and current data in a structured format, data warehouses enable complex querying, reporting, and analysis. This supports deeper insights into performance, customer behavior, and operational efficiency.
4. **Trend Analysis**  
   With historical data stored over time, businesses can perform trend analysis to identify patterns, predict future outcomes, and make data-driven forecasts.
5. **Support for Business Intelligence (BI)**  
   Data warehousing serves as the foundation for Business Intelligence tools. It facilitates dashboards, reports, and visualizations that support executive decision-making, performance tracking, and strategic planning.

**Functions of Data warehouse**

It serves as a collection of organized data, managed by different groups to support data retrieval. It tracks high-transaction tables and helps define key data warehousing techniques and functions.

* Data Consolidation: Combines data from multiple sources into a single, consistent repository.
* Data Cleaning: Removes errors, duplicates, and irrelevant information to ensure data quality.
* Data Integration: Merges data from various sources into a unified format for accurate analysis.
* Data Storage: Stores large volumes of historical data for easy and quick access.
* Data Transformation: Converts and standardizes data to ensure consistency and usability.
* Data Analysis: Enables deep data exploration and insight generation.
* Data Reporting: Supports dashboards and reports for stakeholders and departments.
* Data Mining: Identifies patterns and trends to aid in strategic decisions.
* Performance Optimization: Ensures fast querying and efficient data access.

**Components data warehouse**

* *Data Sources*
* [*ETL*](https://www.geeksforgeeks.org/dbms/etl-process-in-data-warehouse/) *(Extract, Transform, Load) Process*
* *Data Warehouse Storage*
* *Metadata*
* [*OLAP*](https://www.geeksforgeeks.org/dbms/olap-full-form/) *(Online Analytical Processing) Tools*
* *Reporting & Visualization*

.

**Components of Data Warehouse antiitecture**.

1. **External Sources:**External sources are where data originates. These sources provide a variety of data types, such as structured data (databases, spreadsheets); semi-structured data (XML, JSON) and unstructured data (emails, images).
2. **Staging Area:**The staging area is a temporary space where raw data from external sources is validated and prepared before entering the data warehouse. This process ensures that the data is consistent and usable. To handle this preparation effectively, ETL (Extract, Transform, Load) tools are used

**Extract (E):**Pulls raw data from external sources

**Transform (T):**Converts raw data into a standard, uniform format.

**Load (L):**Loads the transformed data into the data warehouse for further processing.

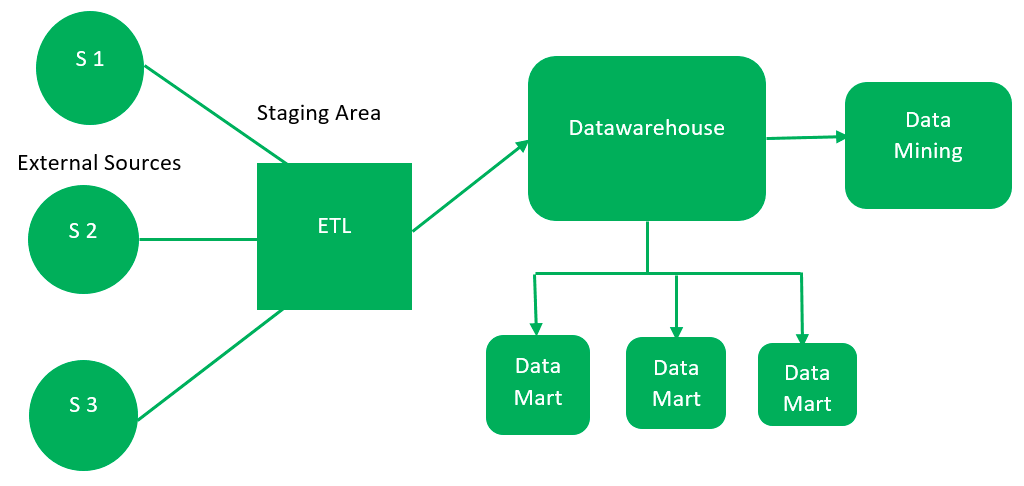
1. **Data Warehouse:**The data warehouse acts as the central repository for storing cleansed and organized data. It contains metadata and raw data. The data warehouse serves as the foundation for advanced analysis, reporting, and decision-making.
2. **Data Marts:**A data mart is a subset of a data warehouse that stores data for a specific team or purpose, like sales or marketing. It helps users quickly access the information they need for their work.
3. **Data Mining:**Data mining is the process of analyzing large datasets stored in the data warehouse to uncover meaningful patterns, trends, and insights. The insights gained can support decision-making, identify hidden opportunities, and improve operational efficiency.

**Top-Down Approach**

The Top-Down Approach, introduced by Bill Inmon, is a method for designing data warehouses that starts by building a centralized, company-wide data warehouse. This central repository acts as the single source of truth for managing and analyzing data across the organization. It ensures data consistency and provides a strong foundation for decision-making.

**Working of Top-Down Approach**

* **Central Data Warehouse:**The process begins with creating a comprehensive data warehouse where data from various sources is collected, integrated, and stored. This involves the ETL (Extract, Transform, Load) process to clean and transform the data.
* **Specialized Data Marts:**Once the central warehouse is established, smaller, department-specific data marts (e.g., for finance or marketing) are built. These data marts pull information from the main data warehouse, ensuring consistency across departments.



**Advantages of Top-Down Approach**

**1. Consistent Dimensional View:** Data marts are created directly from the central data warehouse, ensuring a consistent dimensional view across all departments. This minimizes discrepancies and aligns data reporting with a unified structure.

**2. Improved Data Consistency:**By sourcing all data marts from a single data warehouse, the approach promotes standardization. This reduces the risk of errors and inconsistencies in reporting, leading to more reliable business insights.

**3. Easier Maintenance:**Centralizing data management simplifies maintenance. Updates or changes made in the data warehouse automatically propagate to all connected data marts, reducing the effort and time required for upkeep.

**4. Better Scalability:**The approach is highly scalable, allowing organizations to add new data marts seamlessly as their needs grow or evolve. This is particularly beneficial for businesses experiencing rapid expansion or shifting demands.

**5. Enhanced Governance:**Centralized control of data ensures better governance. Organizations can manage data access, security, and quality from a single point, ensuring compliance with standards and regulations.

**6. Reduced Data Duplication:**Storing data only once in the central warehouse minimizes duplication, saving storage space and reducing inconsistencies caused by redundant data.

**7. Improved Reporting:**A consistent view of data across all data marts enables more accurate and timely reporting. This enhances decision-making and helps drive better business outcomes.

**8. Better Data Integration:**With all data marts being sourced from a single warehouse, integrating data from multiple sources becomes easier. This provides a more comprehensive view of organizational data and improves overall analytics capabilities.

**Disadvantages of Top-Down Approach**

**1. High Cost and Time-Consuming:**The Top-Down Approach requires significant investment in terms of cost, time, and resources. Designing, implementing, and maintaining a central data warehouse and its associated data marts can be a lengthy and expensive process, making it challenging for smaller organizations.

**2. Complexity:**Implementing and managing the Top-Down Approach can be complex, especially for large organizations with diverse and intricate data needs. The design and integration of a centralized system demand a high level of expertise and careful planning.

**3. Lack of Flexibility:**Since the data warehouse and data marts are designed in advance, adapting to new or changing business requirements can be difficult. This lack of flexibility may not suit organizations that require dynamic and agile data reporting capabilities.

**4. Limited User Involvement:**The Top-Down Approach is often led by IT departments, which can result in limited involvement from business users. This may lead to data marts that fail to address the specific needs of end-users, reducing their overall effectiveness.

**5. Data Latency:**When data is sourced from multiple systems, the Top-Down Approach may introduce delays in data processing and availability. This latency can affect the timeliness and accuracy of reporting and analysis.

**6. Data Ownership Challenges:**Centralizing data in the data warehouse can create ambiguity around data ownership and responsibilities. It may be unclear who is accountable for maintaining and updating the data, leading to potential governance issues.

**7. Integration Challenges:**Integrating data from diverse sources with different formats or structures can be difficult in the Top-Down Approach. These challenges may result in inconsistencies and inaccuracies in the data warehouse.

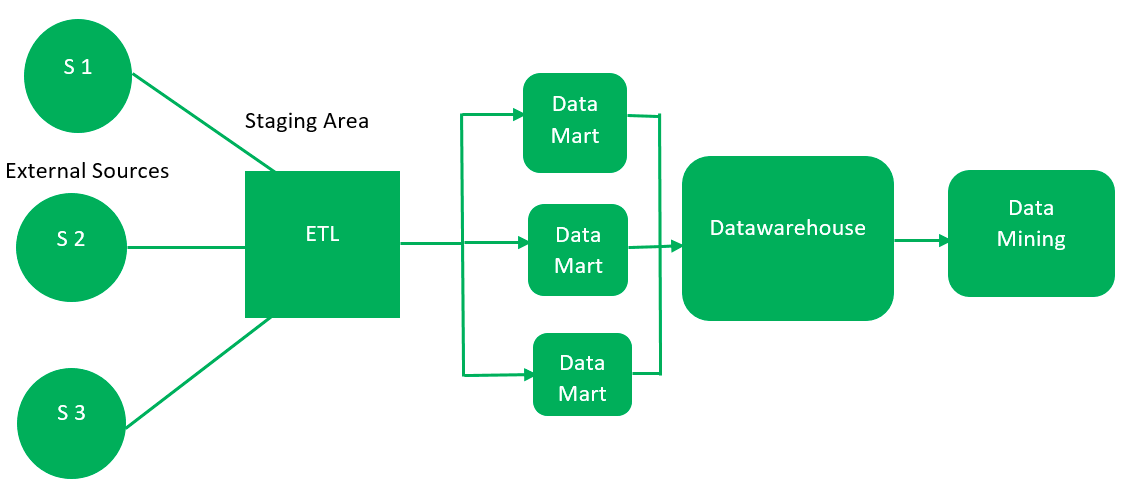
**8. Not Ideal for Smaller Organizations:**Due to its high cost and resource requirements, the Top-Down Approach is less suitable for smaller organizations or those with limited budgets and simpler data needs.

**Bottom-Up Approach**

The Bottom-Up Approach, popularized by Ralph Kimball, takes a more flexible and incremental path to designing data warehouses. Instead of starting with a central data warehouse, it begins by building small, department-specific data marts that cater to the immediate needs of individual teams, such as sales or finance. These data marts are later integrated to form a larger, unified data warehouse.

**Working of Bottom-Up Approach**

* **Department-Specific Data Marts:** The process starts with creating data marts for individual departments or specific business functions. These data marts are designed to meet immediate data analysis and reporting needs, allowing departments to gain quick insights.
* **Integration into a Data Warehouse:**Over time, these data marts are connected and consolidated to create a unified data warehouse. The integration ensures consistency and provides a comprehensive view of the organization’s data.



**Advantages of Bottom-Up Approach**

**1. Faster Report Generation:**Since data marts are created first, reports can be generated quickly, providing immediate value to the organization. This enables faster insights and decision-making.

**2. Incremental Development:**This approach supports incremental development by allowing the creation of data marts one at a time. Organizations can achieve quick wins and gradually improve data reporting and analysis over time.

**3. User Involvement:**The Bottom-Up Approach encourages active involvement from business users during the design and implementation process. Users can provide feedback on data marts and reports, ensuring the solution meets their specific needs.

**4. Flexibility:**This approach is highly flexible, as data marts are designed based on the unique requirements of specific business functions. It is particularly beneficial for organizations that require dynamic and customizable reporting and analysis.

**5. Faster Time to Value:**With quicker implementation compared to the Top-Down Approach, the Bottom-Up Approach delivers faster time to value. This is especially useful for smaller organizations with limited resources or businesses looking for immediate results.

**6. Reduced Risk:**By creating and refining individual data marts before integrating them into a larger data warehouse, this approach reduces the risk of failure. It also helps identify and resolve data quality issues early in the process.

**7. Scalability:**The Bottom-Up Approach is scalable, allowing organizations to add new data marts as needed. This makes it an ideal choice for businesses experiencing growth or undergoing significant change.

**8. Clarified Data Ownership:**Each data mart is typically owned and managed by a specific business unit, which helps clarify data ownership and accountability. This ensures data accuracy, consistency, and proper usage across the organization.

**9. Lower Cost and Time Investment:**Compared to the Top-Down Approach, the Bottom-Up Approach requires less upfront cost and time to design and implement. This makes it an attractive option for organizations with budgetary or time constraints.

**Disadvantage of Bottom-Up Approach**

**1. Inconsistent Dimensional View:**Unlike the Top-Down Approach, the Bottom-Up Approach may not provide a consistent dimensional view of data marts. This inconsistency can lead to variations in reporting and analysis across departments.

**2. Data Silos:**This approach can result in the creation of data silos, where different business units develop their own data marts independently. This lack of coordination may cause redundancies, data inconsistencies, and difficulties in integrating data across the organization.

**3. Integration Challenges:**Integrating multiple data marts into a unified data warehouse can be challenging. Differences in data structures, formats, and granularity may lead to issues with data quality, accuracy, and consistency.

**4. Duplication of Effort:**In a Bottom-Up Approach, different business units may inadvertently duplicate efforts by creating data marts with overlapping or similar data. This can result in inefficiencies and increased costs in data management.

**5. Lack of Enterprise-Wide View:**Since data marts are typically designed to meet the needs of specific departments, this approach may not provide a comprehensive, enterprise-wide view of data. This limitation can hinder strategic decision-making and limit an organization’s ability to analyze data holistically.

**6. Complexity in Management:**Managing and maintaining multiple data marts with varying complexities and granularities can be more challenging compared to a centralized data warehouse. This can lead to higher maintenance efforts and potential difficulties in ensuring long-term scalability.

**7. Risk of Inconsistency:**The decentralized nature of the Bottom-Up Approach increases the risk of data inconsistency. Differences in data structures and definitions across data marts can make it difficult to compare or combine data, reducing the reliability of reports and analyses.

**8. Limited Standardization:**Without a central repository to enforce standardization, the Bottom-Up Approach may lack uniformity in data formats and definitions. This can complicate collaboration and integration across departments.

**Approaches to Building a Data Warehouse**

A good approach to the structure of a data warehouse allows it to operate more efficiently. There are two main approaches to building a data warehouse: **Inmon’s approach** and **Kimball’s approach**.

1. **Inmon’s Approach**

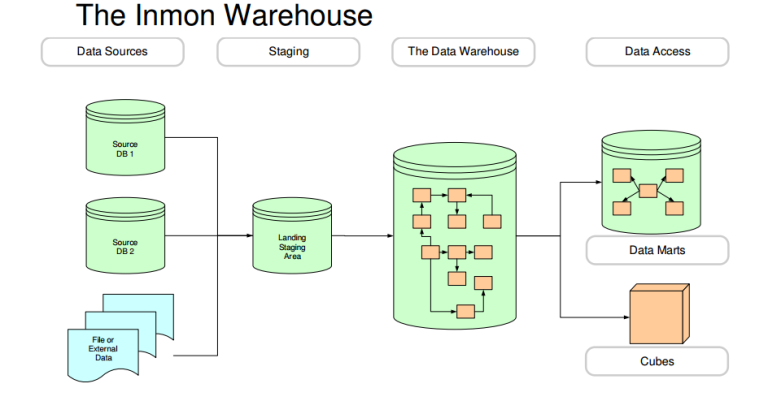
The father of data warehousing, Bill Inmon, developed the model of a normalised data warehouse that’s still widely used today. The aptly titled “Inmon approach” starts by **identifying the primary entities of the business**—departments, customers, products, etc.—and then creates a detailed **logical model of each entity**.

Importantly, these logical models are **normalised** (the process of structuring databases according to specific norms), and the **physical data warehouse is built** to reflect the normalised structure of the warehouse.

In this approach, **data marts** are constructed separately for each division of the company (e.g., marketing, sales, etc.). They exist independently of the data warehouse, but they draw data from a communal, centralised database.

The advantage of the Inmon approach is that it preserves the integrity, or the “**single point of truth**” of the data. Since there’s only one definitive source of data, there is a low opportunity for redundancy or contradiction.

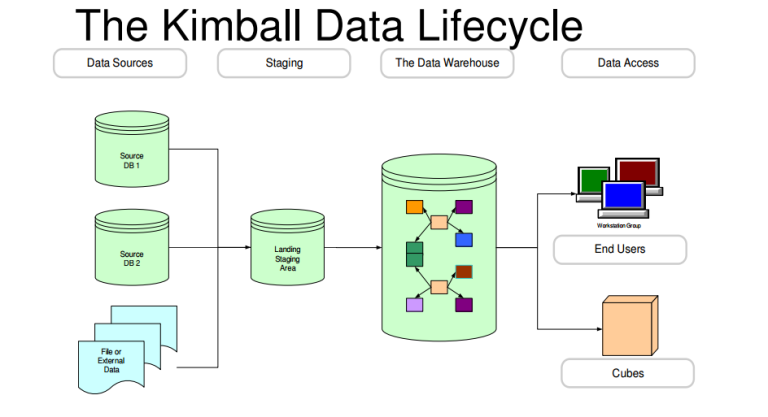
The downside, however, is that this structure requires a high level of investment in the modelling of each entity, and as a business grows, their data warehouse can become exponentially more complicated to maintain.



2. **Kimball’s Approach**

The other side of the coin is Kimball’s approach, named after Ralph Kimball. He proposed a “bottom-up” approach that starts by **identifying processes** and building the warehouse around the kinds of questions data analysts and decision-makers need to be answered.

In contrast to Inmon’s approach, the data warehouse contains a web of data marts. Data is organised into “**star schemas**” where data items are used consistently across different data marts.

The strength of Kimball’s approach is that it’s easier to implement, as the time-consuming process of constructing logical models for each entity is avoided altogether. However, unlike the Inmon approach, there is no centralised “**true**” data repository, so incorrect or redundant data can easily propagate as records are updated. Data Warehouse Design Approaches

As the Inmon and Kimball approaches illustrate, there’s more than one way to build a data warehouse. Similarly, there are different ways to **design** a data warehouse.

While the **top-down** and **bottom-up** design approaches ultimately work toward the same goal (storing and processing data), there are key differences that Inmon and Kimball recognized that make the approach to designing a data warehouse more nuanced than it may seem.

1. Top-Down Design Approach

Inmon’s approach is an example of top-down design. The top-down design describes a data warehouse where **information flows from external sources** (business operational systems) through a so-called staging area **into data marts**.

The staging area is a temporary **landing zone** for data that’s not ready to be implemented into the model.

From the data warehouse, data can be **mined, analysed, and distributed** to data marts, where subsections of the data can be used by various entities.

The top-down approach is the most common way for large organisations to build data centres. Since it’s a strong, high-integrity model and data marts can be easily created from the warehouse, it’s a natural fit for organisations with large data needs. However, the expense of upkeep and maintaining an increasingly complex system is prohibitive for some organisations.

2. Bottom-Up Design Approach

Bottom-up design is more reminiscent of Kimball’s approach. Unlike top-down design, where data flows from the data warehouse into data marts, in a bottom-up approach, **data flows out from decentralised data marts into the central repository** of the data warehouse.

The bottom-up approach includes the same process of data being collected from external sources and flowing through a staging area. But in this case, data is portioned out from the staging area into individual data marts. Then, these data marts form a constellation of data and are consolidated in the data warehouse.

Although both approaches utilise a staging area, it’s technically an optional step. The reason most data warehouses are built with a **staging area** in mind is that time-sensitive documents should be moved into the database at **coordinated times** and not simply as soon as the data becomes available.

For example, if the accounting department records all invoices first thing in the morning but receives payments in the afternoon, then a decision-maker who accesses the data in between could be misled into thinking accounts receivable has a higher balance than it really does. And since the main purpose of data is to inform, asynchronous data may then misinform its users.

**Data warehouse architecture**

A data warehouse is a centralized repository that consolidates data from various sources, enabling comprehensive analysis and reporting. Its architecture is meticulously crafted to optimize data storage, enable high performance and support scalable analytical workloads. The primary architectural models include:

**1. Single-tier architecture**

* **Overview:**Aims to minimize data redundancy by consolidating data into a unified layer
* **Advantages:** Simplifies design and can offer rapid query responses
* **Limitations:** May face scalability challenges as data volumes grow

**2. Two-tier architecture**

* **Overview:** Separates data sources from analytical processes
* **Advantages:** Offers a clear distinction between operational and analytical data processing
* **Limitations:** Direct connections between data sources and end-user tools can hinder scalability and performance

**3. Three-tier architecture (most common)**

* **Bottom tier:**The database server where raw data is stored
* **Middle tier:**An online analytical processing (OLAP) server that facilitates complex analyses
* **Top tier:**The front-end layer comprising visualization, reporting and business intelligence tools

This three-tier structure is prevalent due to its ability to balance performance, scalability and user accessibility

**Difference between Operational Database and Data Warehouse**

| **Operational Database** | **Data Warehouse** |
| --- | --- |
| Operational frameworks are outlined to back high-volume exchange preparing. | Data warehousing frameworks are regularly outlined to back high-volume analytical processing (i.e., OLAP). |
| operational frameworks are more often than not concerned with current data. | Data warehousing frameworks are ordinarily concerned with verifiable information. |
| Data inside operational frameworks are basically overhauled frequently agreeing to need. | Non-volatile, unused information may be included routinely. Once Included once in a while changed. |
| It is planned for real-time commerce managing and processes. | It is outlined for investigation of commerce measures by subject range, categories, and qualities. |
| Relational databases are made for on-line value-based Preparing (OLTP) | Data Warehouse planned for on-line Analytical Processing (OLAP) |
| Operational frameworks are ordinarily optimized to perform quick embeds and overhauls of cooperatively little volumes of data. | Data warehousing frameworks are more often than not optimized to perform quick recoveries of moderately tall volumes of information. |
| Data In | Data out |
| Operational database systems are generally application-oriented. | While data warehouses are generally subject-oriented. |

**Characteristics of Data warehouse**

**1. Subject-Oriented**

A data warehouse is **subject-oriented**, meaning it focuses on specific themes like **sales, healthcare**, **marketing**, or **distribution**, rather than day-to-day operations. It is designed to collect and organize data related to a particular topic to support **analysis and decision-making**. Unnecessary data is removed, making it easier to get clear and relevant insights for that subject.

Subject-oriented

**2. Integrated**

**Integration** in a data warehouse means combining data from different sources like mainframes and relational databases into a consistent and reliable format. This involves using **standard naming conventions, formats, and codes** so that data can be easily understood and analyzed. Integration ensures that all related data is unified, allowing for more accurate and efficient decision-making across different subject areas.

Data warehouse is integrated

**3. Time-Variant**

**Time-variance** means that data in a data warehouse is stored over different time periods—such as weekly, monthly, or yearly. Unlike operational systems, it keeps **historical data** for long-term analysis. Once data is entered, it is **not changed or updated**, preserving the state of data at a specific point in time. This allows users to analyze trends and changes **over time**.

Time-Variant

**4. Non-Volatile**

**Non-volatility** means that once data is stored in a data warehouse, it is **not**deleted**or**update**d**. Instead, new data is added over **time**, keeping the historical records intact. The data is read-only and refreshed at specific intervals, making it ideal for analyzing trends and long-term performance.

Unlike operational systems, a data warehouse does not require transaction processing, recovery, or concurrency control. Operations like insert**,**update, and delete used in day-to-day applications are generally not performed here.

There are mainly **two types of**data operations in a data warehouse:

1. **Data Loading:** inserting bulk data from various sources.
2. **Data Access:** reading and analyzing the stored data.