What is a Data Warehouse?

- ***Formal Definition:*** " A data warehouse is a *subject-oriented*, *integrated*, *time-variant* and *non-volatile* collection of data in support of management decision making process."

WHAT????

- It means:
  - ***Subject-Oriented:*** Stored data targets specific subjects.

  Example: It may store data regarding total Sales, Number of Customers, etc. and not general data on everyday operations.
  - ***Integrated:*** Data may be distributed across heterogeneous sources which have to be integrated.

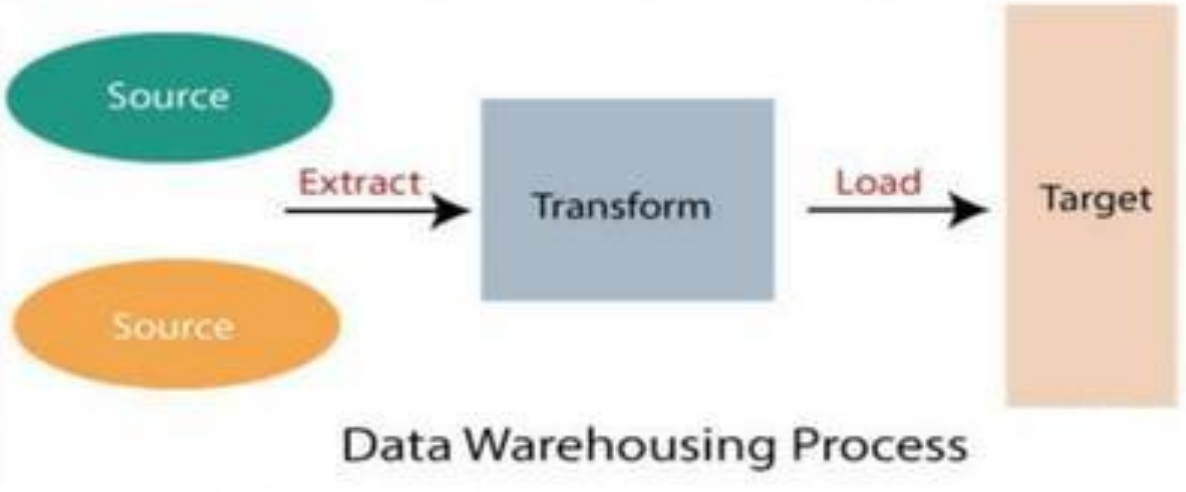
  Example: Sales data may be on RDB, Customer information on Flat files, etc.
  - ***Time Variant:*** Data stored may not be current but varies with time and data have an element of time.

  Example: Data of sales in last 5 years, etc.
  - ***Non-Volatile:*** It is separate from the Enterprise Operational Database and hence is not subject to frequent modification. It generally has only 2 operations performed on it: *Loading of data* and *Access of data*.

Data Warehouse

Data Warehouse:

- A Data Warehouse refers to a place where data can be stored for useful mining. It is like a quick computer system with exceptionally huge data storage capacity. Data from the various organization's systems are copied to the Warehouse, where it can be fetched and conformed to delete errors. Here, advanced requests can be made against the warehouse storage of data.



Data Warehousing Process

The ETL Process:

1. **Extract**:
    1. In this step, data is gathered or extracted from various **source systems**. These systems could be relational databases, flat files, APIs, or other types of data repositories.
    2. The goal of the extraction phase is to collect the raw data from different sources without making any changes to it, though sometimes it involves partial extraction, depending on the use case (e.g., extracting only certain columns or rows).

The ETL Process:

1. **Extract**:
    1. In this step, data is gathered or extracted from various **source systems**. These systems could be relational databases, flat files, APIs, or other types of data repositories.
    2. The goal of the extraction phase is to collect the raw data from different sources without making any changes to it, though sometimes it involves partial extraction, depending on the use case (e.g., extracting only certain columns or rows).

The ETL Process:

2. Transform:

After extraction, the data is **transformed** into a format suitable for analysis and consistent with the structure of the data warehouse. The transformation step may involve several operations, including:

- **Data cleaning**: Removing inconsistencies, duplicates, or errors.
- **Data enrichment**: Adding additional data (e.g., calculations or lookups).
- **Data integration**: Combining data from different sources into a unified format.
- **Data aggregation**: Summarizing data to a higher level (e.g., daily totals from hourly data).
- **Data formatting**: Converting data types or applying specific rules to match the data warehouse schema.

The ETL Process:

2. Transform:

Data Cleaning: Removing Inconsistencies, Duplicates, or Errors

Data cleaning is the process of identifying and correcting errors or inconsistencies in the data to ensure accuracy and completeness.

Example: Suppose you have a customer dataset with the following issues:

- Some records have missing values for the "Email Address" field.
- There are duplicate records for a customer with slightly different information (e.g., different spellings of the name "John Smith" as "Jon Smith" and "John Smythe").
- Some customer phone numbers are recorded in different formats, such as "(123) 456-7890" and "123-456-7890".

The ETL Process:

2. Transform:

Data Cleaning Tasks:

•**Fixing duplicates**: Identifying and merging records for "John Smith" and "Jon Smith" to avoid counting the same person multiple times.

•**Filling missing values**: Using a data imputation method (e.g., replacing missing email addresses with a placeholder or using an existing database of emails).

•**Standardizing formats**: Converting all phone numbers to a single format, such as "123-456-7890".

The ETL Process:

2. Transform:

Data Enrichment: Adding Additional Data (e.g., Calculations or Lookups)

Data enrichment involves supplementing the existing dataset with additional information, either through calculations or by looking up data from external sources.

Example: Imagine you have a sales dataset with customer names and purchase amounts. However, you also want to add **geographical data** to analyze sales by region.

Data Enrichment Tasks:

•**Lookups**: Adding a "Region" field to the sales records by performing a lookup on a customer address table, mapping customers to their respective regions.

•**Calculations**: Adding a new field like "Total Purchase Value" by multiplying the "Quantity" and "Price" fields for each sale.

For instance, if a customer is from "New York," you might enrich their record with the region name "East Coast" and add calculated fields like "Purchase Value" = "Quantity * Price."

The ETL Process:

2. Transform:

Data Enrichment: Adding Additional Data (e.g., Calculations or Lookups)

Data enrichment involves supplementing the existing dataset with additional information, either through calculations or by looking up data from external sources.

Example: Imagine you have a sales dataset with customer names and purchase amounts. However, you also want to add **geographical data** to analyze sales by region.

Data Enrichment Tasks:

•**Lookups**: Adding a "Region" field to the sales records by performing a lookup on a customer address table, mapping customers to their respective regions.

•**Calculations**: Adding a new field like "Total Purchase Value" by multiplying the "Quantity" and "Price" fields for each sale.

For instance, if a customer is from "New York," you might enrich their record with the region name "East Coast" and add calculated fields like "Purchase Value" = "Quantity * Price."

The ETL Process:

2. Transform:

Data Integration: Combining Data from Different Sources into a Unified Format

Data integration involves combining data from various source systems into a single, cohesive dataset.

Example: Suppose you have two source systems:

1.System A: Stores customer information (Name, Address, Email).

2.System B: Stores order data (Order ID, Customer ID, Product, Quantity).

You need to integrate the customer data with the order data for analysis.

Data Integration Tasks:

•Joining the two datasets by the "Customer ID" field to combine customer details (e.g., Name, Address) with the order data (e.g., Product, Quantity, Order ID).

•Ensuring that the data types match between both systems (e.g., customer IDs in both systems should be integers or strings and formatted consistently).

The ETL Process:

2. Transform:

Data Aggregation: Summarizing Data to a Higher Level

Data aggregation involves summarizing detailed data at a higher level, typically for easier analysis or reporting.

Example: Suppose you have transaction data at a minute level, but you want to analyze daily sales.

Data Aggregation Tasks:

•**Summarizing data**: Aggregating the sales data by **day**, where each record represents total sales for that day, rather than individual transactions.

- If the original data shows transactions for each hour, you can group the data by date and sum the sales for each day.

•**Aggregating metrics**: You might calculate **average daily sales**, **total quantity sold**, or **total revenue** per day, rather than by individual transactions.

The ETL Process:

2. Transform:

Data Formatting: Converting Data Types or Applying Specific Rules to Match the Data Warehouse Schema

Data formatting involves adjusting data to match the required structure, types, and rules in the data warehouse schema.

Example: Suppose the data warehouse schema requires the "Date of Birth" field to be in a specific format, such as "YYYY-MM-DD", but your source data has dates stored in various formats.

Data Formatting Tasks:

•**Converting date formats**: Converting "MM/DD/YYYY" to "YYYY-MM-DD".

•**Changing data types**: If you have a "Sales Amount" field stored as a string (e.g., "$1000"), you would convert it to a numerical format for proper analysis.

•For instance, if the "Sales Amount" column contains values like "$1000", the transformation step would strip the "$" sign and convert it to a numerical value like 1000.

The ETL Process:

3. Load:

•In the final step, the transformed data is **loaded** into the data warehouse. This can involve inserting new data, updating existing records, or even deleting outdated records. The loading process might be done in **batch** (periodic updates) or **real-time** (continuous updates).

The ETL Process:

Why ETL is Important in Data Warehousing:

•**Data Integration**: ETL allows data from various heterogeneous sources to be consolidated into a single, consistent format, enabling comprehensive analysis.

•**Data Quality**: By transforming the data, ETL helps ensure the data is clean, accurate, and usable for business intelligence purposes.

•**Efficiency**: ETL automates the process of moving data into the data warehouse, which would otherwise be time-consuming and error-prone if done manually.

•**Optimized for Analysis**: The data is structured and prepared in a way that makes it ready for reporting, querying, and analytical purposes.

Important Features of Data Warehouse

The Important features of Data Warehouse are given below:

1. Subject Oriented:

A data warehouse is subject-oriented. It provides useful data about a subject instead of the company's ongoing operations, and these subjects can be customers, suppliers, marketing, product, promotion, etc. A data warehouse usually focuses on modelling and analysis of data that helps the business organization to make data-driven decisions.

2. Time-Variant:

The different data present in the data warehouse provides information for a specific period.

3. Integrated:

A data warehouse is built by joining data from heterogeneous sources, such as social databases, level documents, etc.

4. Non- Volatile :

It means, once data entered into the warehouse cannot be change.

Advantages of Data Warehouse:

- More accurate data access
- Improved productivity and performance
- Cost-efficient
- Consistent and quality data

Characteristics of Data Warehouse

1. Subject-Oriented:

- It provides ***topic wise information*** rather than the overall processes of a business. (i.e.) sales, inventory, promotion, etc.

2. Integrated:

- DW is developed by ***integrating data*** from varied source into a consistent format.
- It is stored in data warehouse in a consistent manner in terms of ***naming, format & coding***, which facilitates data analysis.

3. Non-Volatile:

- Data once entered into a data warehouse must ***remain unchanged*** & all data is read only.

4. Time Variant:

- Data stored in a data warehouse is documented with an ***element of time***, either explicitly or implicitly.
- It is exhibited in primary key with element of time like ***day, week***, etc.

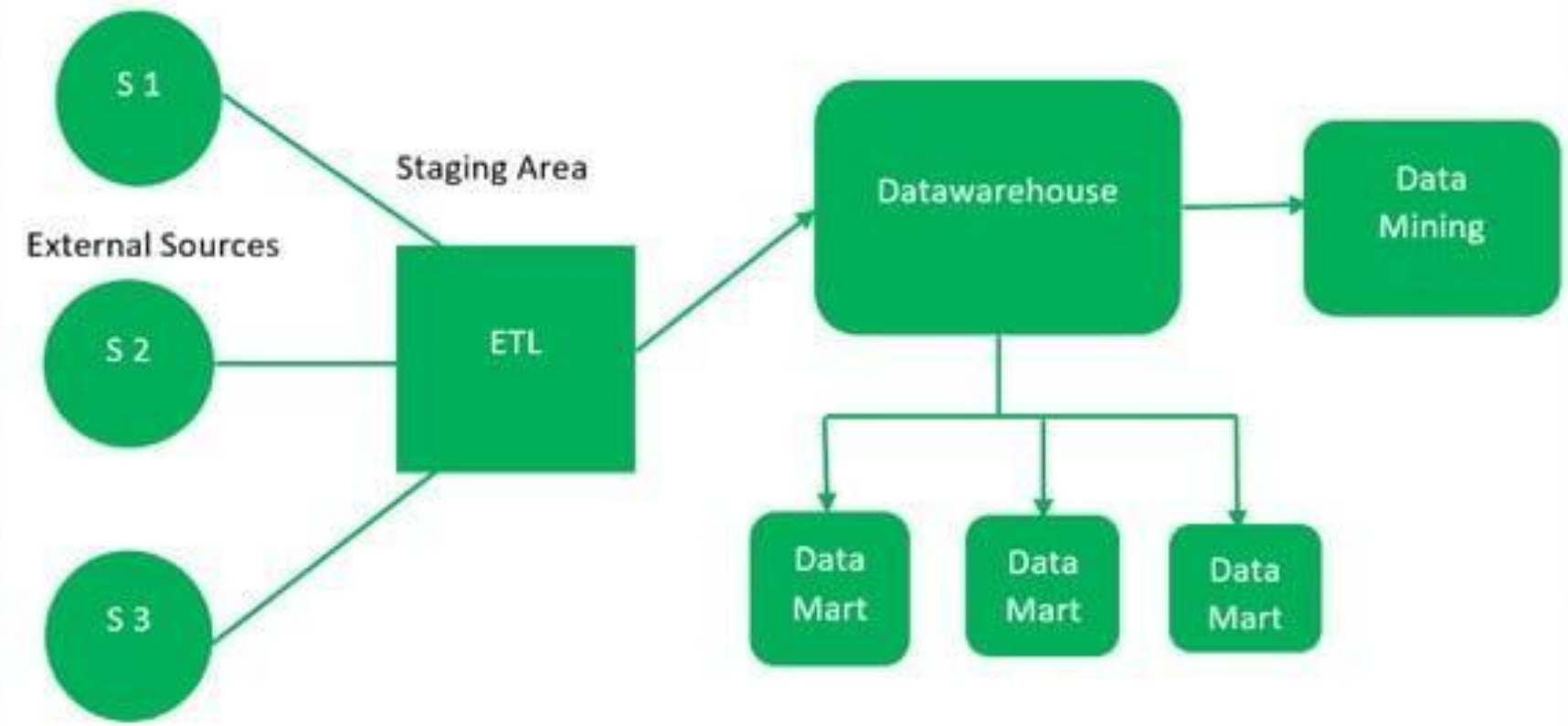
Benefits of Data Warehouse:

- Understand business trends and make better forecasting decisions.
- Data Warehouses are designed to perform well enormous amounts of data.
- The structure of data warehouses is more accessible for end-users to navigate, understand, and query.
- Queries that would be complex in many normalized databases could be easier to build and maintain in data warehouses.
- Data warehousing is an efficient method to manage demand for lots of information from lots of users.
- Data warehousing provide the capabilities to analyze a large amount of historical data.

Data Warehouse Architecture:

A **data-warehouse** is a heterogeneous collection of different data sources organised under a unified schema. There are 2 approaches for constructing data-warehouse: Top-down approach and Bottom-up approach are explained as below.

1. Top-down approach:


S 1
Staging Area
External Sources
S 2
ETL
Datawarehouse
Data Mining
S 3
Data Mart
Data Mart
Data Mart

The essential components are below:

1. **External Sources –**

   External source is a source from where data is collected irrespective of the type of data. Data can be structured, semi structured and unstructured as well.

2. **Stage Area –**

   Since the data, extracted from the external sources does not follow a particular format, so there is a need to validate this data to load into data warehouse. For this purpose, it is recommended to use **ETL** tool.

- **E(Extracted):** Data is extracted from External data source.
- **T(Transform):** Data is transformed into the standard format.
- **L(Load):** Data is loaded into data warehouse after transforming it into the standard format.

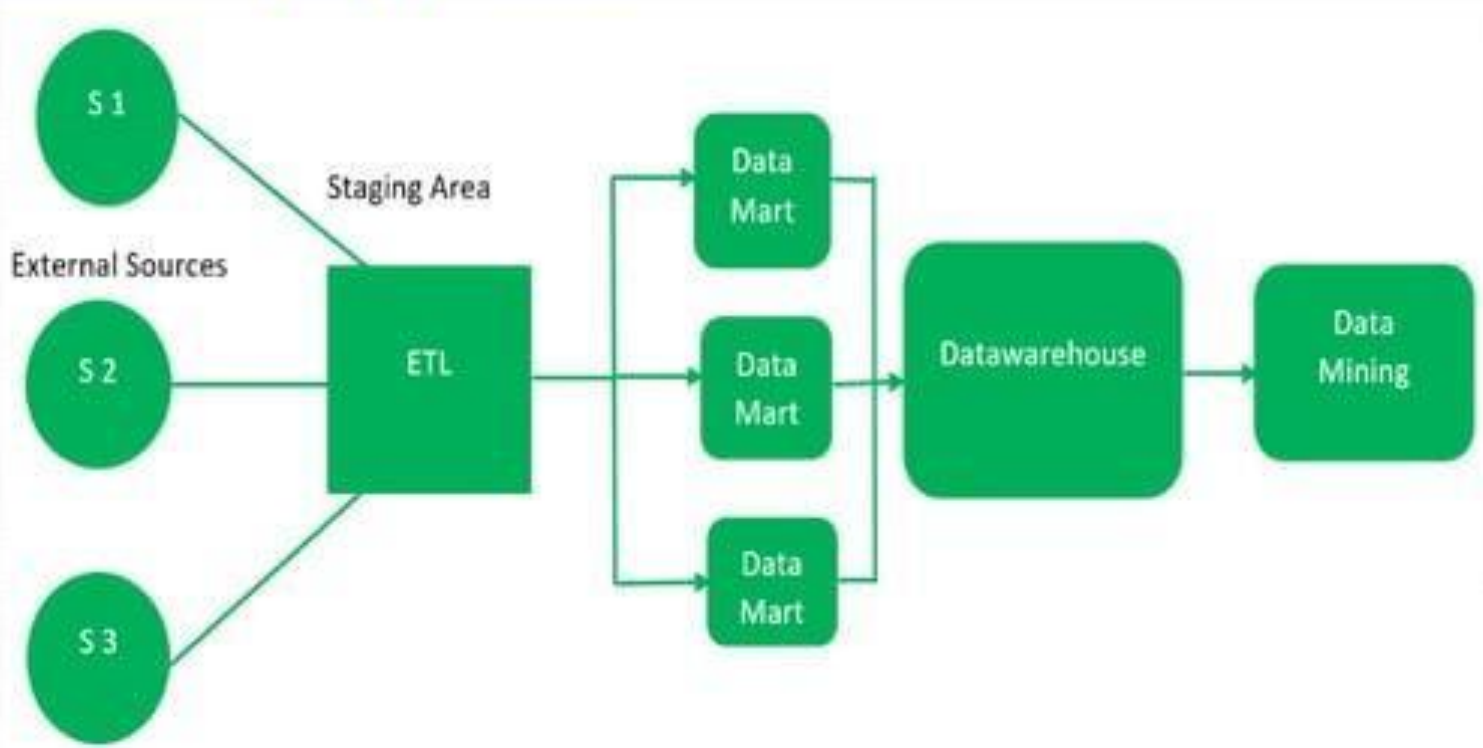
- **3.Data-warehouse-** –

  After cleansing of data, it is stored in the data warehouse as central repository. It actually stores the meta data and the actual data gets stored in the data marts. **Note** that data warehouse stores the data in its purest form in this top-down approach.

4. **DataMarts-** –
Data mart is also a part of storage component. It stores the information of a particular function of an organisation which is handled by single authority. There can be as many number of data marts in an organisation depending upon the functions. Data mart contains subset of the data stored in data warehouse.

5. **DataMining-** –
The practice of analysing the big data present in data warehouse is data mining. It is used to find the hidden patterns that are present in the database or in data warehouse with the help of algorithm of data mining. This approach is defined by **Inmon** as – data warehouse as a central repository for the complete organisation and data marts are created from it after the complete data warehouse has been created.

2. Bottom-up approach:



1. First, the data is extracted from external sources (same as happens in top-down approach).

2. Then, the data go through the staging area (as explained above) and loaded into data marts instead of data warehouse. The data marts are created first and provide reporting capability. It addresses a single business area.

3. These data marts are then integrated into data warehouse.

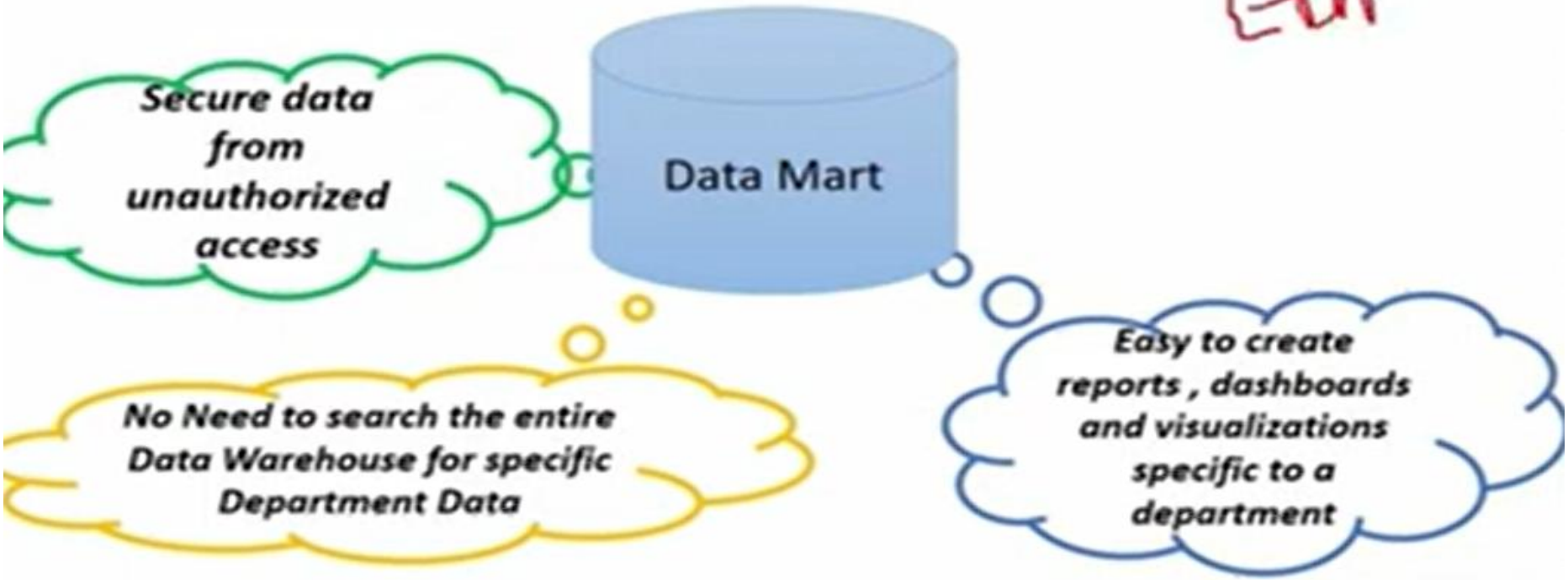
This approach is given by **Kinball** as – data marts are created first and provides a thin view for analyses and data warehouse is created after complete data marts have been created.

Top-down approach	Bottom-up Approach
• Difficult	• Easy
• Complex business requirements	• Simple
• Consistent with global business rules	• Inconsistent with global business rules
• Data warehouse to data mart	• data mart to data warehouse
• Large project	• Small project

Data Marts

Data Mart : It is a Subset of Data Warehouse focused towards specific Line of Business or built for specific group of users.

EDW



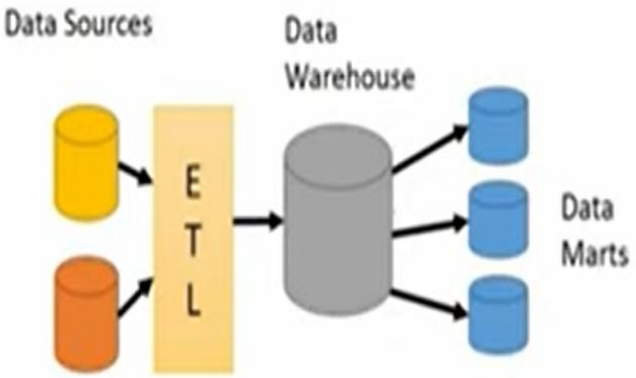
Sales Data Mart

Sales Data Mart is used for Analytics

Inventory Data Mart

Analyzing Inventory Data using Inventory Data Mart

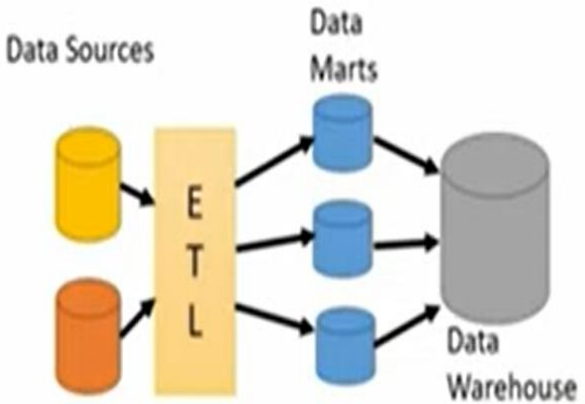
Types of Data Marts



Dependent Data Marts

Top Down Approach

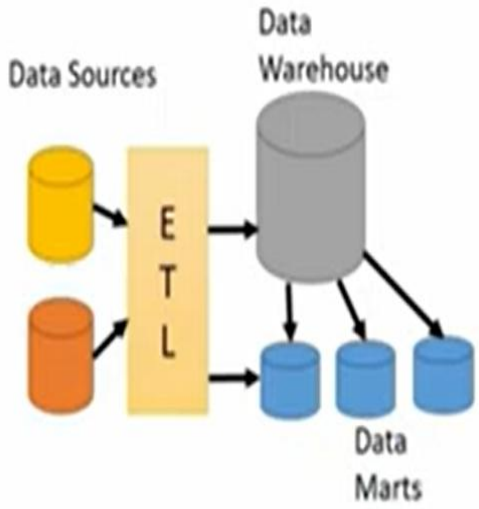
1) First Create the Data Warehouse and then design DataMart on top of it.
2) No Direct interaction with external or internal Data Sources apart from Data Warehouse
3) Good for larger companies



Independent Data Marts

Bottom Up Approach

1) Standalone Systems, can work without the Data Warehouse
2) Data is collected from external and internal data sources in a direct manner.
3) Data collected in Data Marts is used for designing the Data Warehouse down the line
4) Good for medium sized companies as entire data warehouse creation takes lot of time which could be saved if data marts are created first.



Hybrid Data Marts

Hybrid Approach

1) Data is collected from external, internal data sources along with data warehouse
2) Important while doing Ad hoc integration such as when data of new business area arises.
3) Addresses the issues of Dependent and Independent data marts.

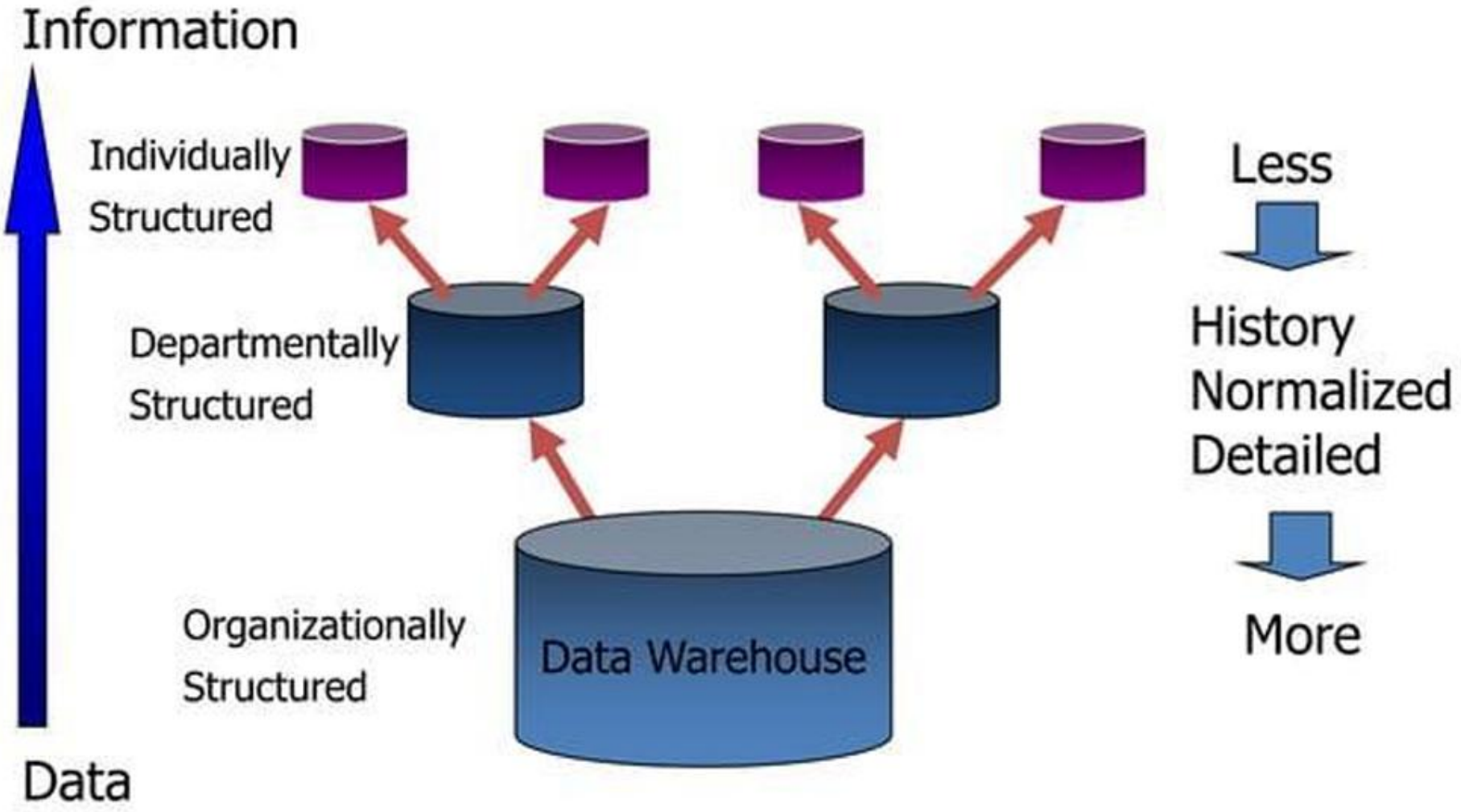
Data Marts

- A data mart is a scaled down version of a data warehouse that focuses on a particular subject area.
- A **data mart** is a subset of an organizational data store, usually oriented to a specific purpose or major data subject, that may be distributed to support business needs.
- Data marts are analytical data stores designed to focus on specific business functions for a specific community within an organization.
- Usually designed to support the unique business requirements of a specified department or business process
- Implemented as the first step in proving the usefulness of the technologies to solve business problems

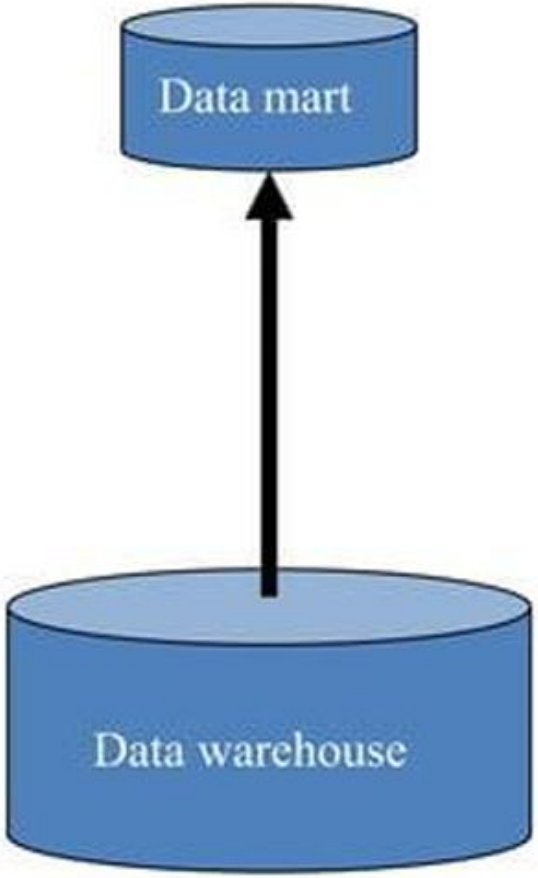
Reasons for creating a data mart

- Easy access to frequently needed data
- Creates collective view by a group of users
- Improves end-user response time
- Ease of creation in less time
- Lower cost than implementing a full Data warehouse
- Potential users are more clearly defined than in a full Data warehouse

From the Data Warehouse to Data Marts



Characteristics of the Departmental Data Mart



- Small
- Flexible
- Customized by Department
- OLAP
- Source is departmentally structured data warehouse

TABLE 9-2 Data Warehouse Versus Data Mart

Data Warehouse	Data Mart
Scope	**Scope**
• Application independent • Centralized, possibly enterprise-wide • Planned	• Specific DSS application • Decentralized by user area • Organic, possibly not planned
Data	**Data**
• Historical, detailed, and summarized • Lightly denormalized	• Some history, detailed, and summarized • Highly denormalized
Subjects	**Subjects**
• Multiple subjects	• One central subject of concern to users
Sources	**Sources**
• Many internal and external sources	• Few internal and external sources
Other Characteristics	**Other Characteristics**
• Flexible • Data oriented • Long life • Large • Single complex structure	• Restrictive • Project oriented • Short life • Start small, becomes large • Multi, semi-complex structures, together complex

What are Operational DBMS?

- They consist of Tables having attributes and are populated by tuples.
- They generally use the E-R data model.
- It is used to store transactional data.
- The information content is generally recent.
- These are thus called as OLTP systems.
- Their goals are data accuracy & consistency , Concurrency , Recoverability, Reliability (ACID Properties).

Contd...

Features of a Warehouse:

- It is separate from Operational Database.
- Integrates data from heterogeneous systems.
- Stores HUGE amount of data, more historical than current data.
- Does not require data to be highly accurate.
- Queries are generally complex.
- Goal is to execute statistical queries and provide results which can influence decision making in favor of the Enterprise.
- These systems are thus called Online Analytical Processing Systems (OLAP).

Operational DBS vs. Warehouses

- **Data Contents:**

Operational DB Systems: Current and detailed data and is subject to modifications.

Data Warehouse: Historical data, course granularity, generally not modified.

- **Users:**

Operational DB Systems: *Customer – Oriented, thus used by customers/clerks/IT professionals.*

Data Warehouse: Market – Oriented, thus used by Managers/Executives/Analysts.

- **Database Design:**

Operational DB Systems: *Usually E-R model.*

Data Warehouse: Usually Multidimensional model. (Star, Snowflake…)

- **Nature of Queries:**

Operational DB Systems: *Short, atomic queries desiring high performance (less latency) and accuracy.*

Data Warehouse: Mostly read only queries, operate on HUGE volumes of data, queries are quite complex.

OLTP vs. OLAP

	OLTP	OLAP
users	clerk, IT professional	knowledge worker
function	day to day operations	decision support
DB design	application-oriented	subject-oriented
data	current, up-to-date detailed, flat relational isolated	historical, summarized, multidimensional integrated, consolidated
usage	repetitive	ad-hoc
access	read/write index/hash on prim. key	lots of scans
unit of work	short, simple transaction	complex query
# records accessed	tens	millions
#users	thousands	hundreds
DB size	100MB-GB	100GB-TB
metric	transaction throughput	query throughput, response

Table 2.1: summarize the difference between OLTP and OLAP

Parameters	OLTP	OLAP
Process	It is an Online Transactional System. It manages database modification.	OLAP is an Online Analysis and data retrieving process.
What the Data	Reveals a snapshot of ongoing business processes.	Multi-dimensional views of various kinds of business activities.
Purpose of Data	Used to run business	Used to analyze business
Source of Data	Operational Data; OLTPs are the original source of the data.	Consolidation data; OLAP data comes from the various OLTP databases.
Query	Insert, Update and Delete information from the database.	Based on Select commands to aggregate data for reporting.
Processing Speed	Typically, very fast	Depending on the amount of data involved batch data refreshes and complex queries may take many hours.
Space Requirements	Can be relatively small if historical data is archived	Larger due to the existence of aggregation structures and history data; requires more indexes than OLTP.

Response	Its response time is in milliseconds.	Response time in seconds to minutes.
Performance Metric	Transaction throughput is the performance metric.	Query throughput is the performance metric.
Database Design	ER modeling	Schema-Star Schema and Snowflake Schema
Orientation	Application-oriented	Subject Oriented
Data Integrity	OLTP database must maintain data integrity constraints.	OLAP database does not get frequently modified. Hence, data integrity is not an issue.
Target Audience/Spectators	It is a market-oriented process.	It is a customer-oriented process.
Productivity	It helps to increase users' self-service and productivity.	Help to increase the productivity of the business analysts.

Transaction Throughput:

1. Transaction Throughput:

Definition: Transaction throughput refers to the rate at which the data warehouse can process **write operations**, such as **inserts**, **updates**, and **deletes**. This includes the addition of new data, updating existing records, and removing data from the system.

Key Focus: It focuses on how quickly the system can handle **data ingestion** or **transactional activities**.

Performance Impact: High transaction throughput indicates that the system can handle a large volume of data being loaded or modified in a given time period. This is crucial during ETL (Extract, Transform, Load) processes or when real-time data is being added to the data warehouse.

Example: In a retail data warehouse, transaction throughput could refer to how quickly new sales records are added to the warehouse, such as each sale being processed and written into the system.

Query Throughput:

Query Throughput:

Definition: Query throughput refers to the rate at which the data warehouse can handle **read operations**, such as executing **queries** to retrieve data for analysis, reporting, or business intelligence purposes.

Key Focus: It focuses on how quickly the system can process **queries** to retrieve the requested data for users or applications.

Performance Impact: High query throughput indicates that the system can execute a large number of queries efficiently and return results quickly. This is essential for business analysts, data scientists, or reporting systems that rely on fast data retrieval.

Example: In the same retail data warehouse, query throughput could refer to how fast the system can respond to queries like "What is the total sales for the last quarter in Region X?" or "How many units of product Y were sold in the past week?"

Aspect	Transaction Throughput	Query Throughput
Focus	Write operations (inserts, updates, deletes)	Read operations (select, retrieve data)
Performance Concern	How fast the system can handle data loading or modification	How fast the system can execute queries and return results
Use Case	Relevant during ETL processes, data loading, or real-time data updates	Relevant when querying the data warehouse for reporting, analytics, or business insights
Impact on System	High transaction throughput ensures that data can be quickly ingested and updated	High query throughput ensures that users can quickly retrieve data for analysis

Granularity

- In a **data warehouse**, **granularity** refers to the level of detail or summarization at which data is stored. It defines how finely or coarsely data is represented in the warehouse and determines the scope of information captured for analysis and reporting.
- Granularity is an important concept because it directly impacts the volume of data, the types of queries that can be efficiently performed, and the overall performance of the data warehouse. The granularity can range from highly detailed, transaction-level data to highly summarized, aggregated data.

Granularity

- In the context of a data warehouse, **course granularity** refers to the level of detail or summarization of the data stored. Essentially, it defines how detailed or aggregated the data is in the warehouse.
- **Fine Granularity**: This represents a highly detailed level of data, capturing every individual transaction or event.
- **Coarse Granularity**: This represents a more summarized level of data, where details are aggregated or rolled up to provide a broader view.

Granularity

Example of Granularity in a Data Warehouse:

Imagine a sales data warehouse where the objective is to store and analyze sales transactions. The granularity can vary based on the requirements of the business:

Fine Granularity (Detailed Data):

- Each individual sales transaction is recorded with attributes such as:
  - Date of sale
  - Product sold
  - Quantity
  - Price
  - Customer ID
  - Store ID
  - Salesperson ID
- This level allows for detailed analysis such as tracking individual customer purchases or identifying sales patterns at a very granular level.

Granularity

Coarse Granularity (Summarized Data):

Data might be aggregated by day or by product category, e.g.:

- Total sales per day
- Total sales per product category
- Average sales per store

This level provides higher-level insights, which might be more useful for reporting or making strategic decisions.

Fine granularity provides detailed, transaction-level data.

Coarse granularity provides summarized, high-level data.

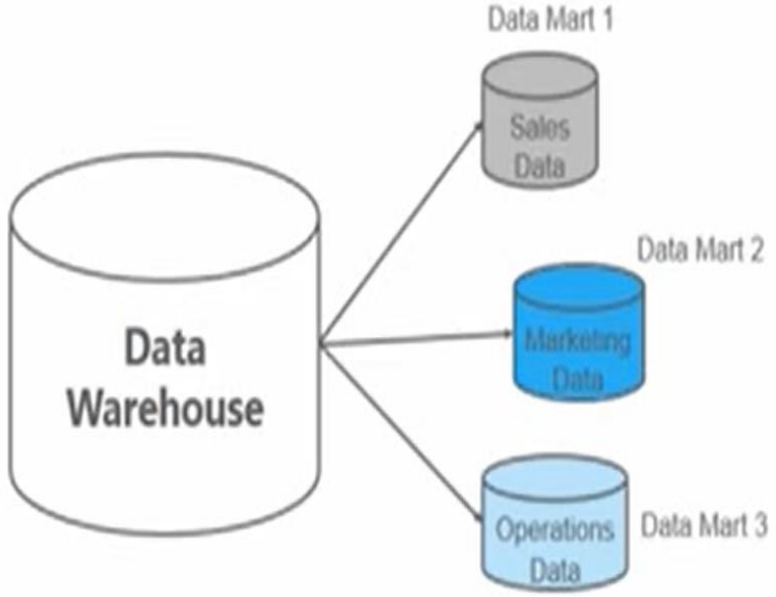
- OLTP Systems are used to *"run"* a business

- The Data Warehouse helps to *"optimize"* the business

Data Mart

- Data mart is a smaller version of the Data Warehouse which deals with a single subject
- Data marts are focused on one area. Hence, they draw data from a limited number of sources
- Time taken to build Data Marts is very less compared to the time taken to build a Data Warehouse

Data Warehouse	Data Marts
Enterprise wide data	Department wide data
Multiple subject areas	Single subject area
Multiple data sources	Limited data sources
Occupies large memory	Occupies limited memory
Longer time to implement	Shorter time to implement

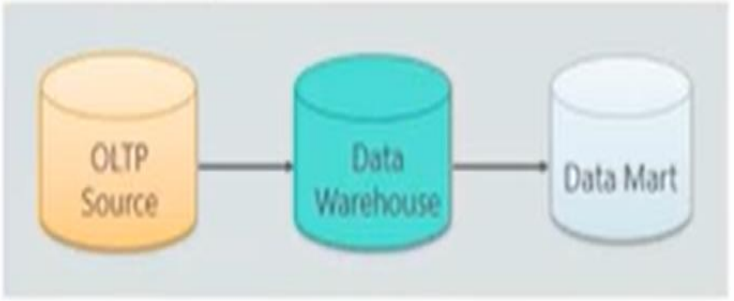


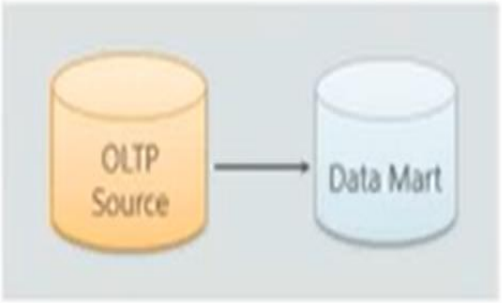
Types Of Data Mart

1. **Dependent Data Mart**
   - The data is first extracted from the OLTP systems and then populated in the central DWH
   - From the DWH, the data travels to the Data Mart

2. **Independent Data Mart**
   - The data is directly received from the source system
   - This is suitable for small organizations or smaller groups within an organization

3. **Hybrid Data Mart**
   - The data is fed both from OLTP systems as well as the Data Warehouse





Metadata

- In a data warehouse, **metadata** is **information about the data**.
- It describes what the data is, where it comes from, how it's organized, and how to use it.
- Think of metadata as a **"label"** or **"instruction manual"** for the data stored in the warehouse.

Simple Example:

Imagine you have a **library** full of books. The books are the actual data. But to find a book, you need a catalog that tells you things like:

The **title** of the book

The **author**

The **year it was published**

The **category** (e.g., fiction, science, history)

This catalog is like **metadata**. It helps you understand the books (data) and find what you're looking for.

Metadata

Types of Metadata with Examples:

1. **Structural Metadata**
   - **What it is**: Describes how the data is organized and where to find it.
   - **Example**: In a data warehouse with customer information, structural metadata tells you that there's a **Customer** table, and inside that table, there are columns for **name**, **address**, **phone number**, and **email**.
   - **In simple terms**: It’s like a map showing where everything is in the data warehouse.
2. **Descriptive Metadata**
   - **What it is**: Describes what the data means or represents.
   - **Example**: If a column is called **customer_id**, descriptive metadata might explain that this column holds a **unique number** that identifies each customer.
   - **In simple terms**: It’s like a description that explains the meaning of the data

Metadata

Types of Metadata with Examples:

3. Administrative Metadata

What it is: Tells how the data is managed, who can use it, and when it was updated.

Example: It might say the customer data was last updated on **January 1, 2025**, and only certain employees have permission to change the data.

In simple terms: It's like a record of who is in charge of the data and when it was last touched.

4. Process Metadata

What it is: Describes how the data is processed or changed.

Example: If the customer data comes from a website and then gets cleaned (like fixing spelling mistakes) before being added to the warehouse, process metadata describes these steps.

In simple terms: It's like a list of instructions or steps for how the data is prepared

Metadata

Types of Metadata with Examples:

Why is Metadata Important?

Without metadata, it would be hard to understand what data is in the warehouse, how to use it, or even where to find it. Metadata helps people make sense of the data quickly and accurately.

In Summary:

Metadata is like a guide or label that helps you understand the data in the warehouse.

It explains how the data is organized, what it means, who manages it, and how it's processed.

Need for DW in business intelligence

- A **data warehouse** is a tool that helps businesses store and manage all their data in one place.
- It is very important for **Business Intelligence (BI)** because it makes it easier to understand and use data to make smart decisions.
- Here's why businesses need it:

Need for DW in business intelligence

1. Keeps All Data in One Place

A data warehouse collects data from many sources (like sales, marketing, or customer support) and puts it together.

This gives a complete view of the business.

2. Makes Data Clean and Organized

It fixes errors, removes duplicates, and organizes data so it’s clear and easy to understand.

Everyone in the business gets the same, reliable data.

Need for DW in business intelligence

3. Finds Answers Quickly

Data warehouses are built to answer questions fast, even if the questions are complex.

This saves time when making reports or analyzing data.

4. Keeps Old Data for Trends

Businesses can save years of data in the warehouse.

This helps them see patterns, compare results, and plan for the future.

Need for DW in business intelligence

5. Works with BI Tools

- Data warehouses connect easily to BI tools that create reports, charts, and dashboards.
- These tools help people see and understand the data better.

6. Helps in Better Decisions

- With clear and accurate data, leaders can make smarter and quicker decisions.
- They can react faster to changes in the market.

Need for DW in business intelligence

7. Grows with the Business

- As a business grows and collects more data, the data warehouse can grow too.
- New data sources can be added easily.

8. Keeps Data Safe

- Data warehouses have strong security to protect sensitive information.
- They also help meet legal rules for storing data.

Need for DW in business intelligence

9. Saves Money Over Time

- Setting up a data warehouse may cost money at first, but it saves money later by making data work easier and faster.
- It reduces manual effort and avoids mistakes.

Difference between DW and DM

S. No	Domain	Data Warehouse	Data Mining
1.	Definition	A place to store and organize large amounts of data. **Example**: A company stores data about all sales, customers, and products in a single database.	The process of finding patterns or useful information in data. **Example**: Discovering that customers aged 20-30 buy more electronics.
2.	Purpose	Collects and stores data in one central location. **Example**: A retail chain saves sales data from all its stores in one central place.	Analyzes the data to find hidden insights. **Example**: Analyzing sales data to find which products sell best during holidays.
3.	Focus	Focuses on storing and organizing data. **Example**: Organizing customer data into categories like age, location, and purchases.	Focuses on analyzing and understanding the data. **Example**: Predicting which customers are likely to buy a new product.

Difference between DW and DM

S. No	Domain	Data Warehouse	Data Mining
4.	Input	Needs raw data from many sources (e.g., sales, marketing, etc. **Example**: Collecting data from marketing tools, sales systems, and websites.	Uses organized data, often from a data warehouse. **Example**: Using clean, structured sales data for analysis.
5.	Output	Provides clean and structured data. **Example**: Monthly sales figures for different regions.	Produces reports, patterns, and predictions. **Example**: Identifying that sales in the north region are higher on weekends.
6.	Process	Stores data through ETL (Extract, Transform, Load) processes. **Example**: Data from an e-commerce site is cleaned and stored in a database.	Uses techniques like algorithms, machine learning, or statistics. **Example**: Using clustering to group customers by spending habits

Difference between DW and DM

S. No	Domain	Data Warehouse	Data Mining
7.	Time	Handles past and current data. **Example**: Storing 5 years of financial data for analysis.	Looks for future trends or patterns in data. **Example**: Predicting which products will sell more next year.
8.	**Tools**	Uses tools like SQL, ETL software, and databases. **Example**: Tools like Amazon Redshift or Snowflake	Uses tools like machine learning models, statistical software, or AI tools. **Example**: Using Python to find which marketing campaigns increase sales.
9.	Complexity	Mainly involves organizing and storing data. **Example**: Structuring sales data into tables.	Involves deeper analysis and advanced methods. **Example**: Using a decision tree to identify customer behavior patterns.

Difference between DW and DM

S. No	Domain	Data Warehouse	Data Mining
10.	User	Mostly used by IT teams and data engineers. **Example**: An IT team manages the company's central data storage system.	Used by data analysts, scientists, and business decision-makers. **Example**: A marketing team uses insights from mining to create targeted ads.
11.	Data Type	Works with structured data (tables, rows, columns). **Example**: Sales records organized by date, store, and product.	Can work with both structured and unstructured data (text, images, etc.). **Example**: Analyzing text reviews or social media posts for customer feedback.

Difference between DW and DM

S. No	Domain	Data Warehouse	Data Mining
12.	Goal	Provides a central data repository for easy access. **Example**: Storing all customer data for creating reports.	Helps make better decisions by finding patterns or trends. **Example**: Discovering which customers are likely to churn.
13.	Tools	Uses tools like SQL, ETL software, and databases. **Example**: Saving yearly sales data for 10 years.	Uses tools like machine learning models, statistical software, or AI tools **Example**: Finding which products are trending this month.
14.	Time Frame	Focuses on long-term data storage.	Focuses on short-term insights from existing data.

Difference between DW and DM

S. No	Domain	Data Warehouse	Data Mining
15.	Example	Storing all sales data of a company in one place.	Analyzing sales data to find which products sell best in winter.
16.	Dependency	Can exist without data mining. **Example**: A company can use it for basic reporting.	Often depends on a data warehouse for clean and structured data. **Example**: Analysts use stored data to find sales trends.
17		A **data warehouse** is like a storage room where all the data is kept safe and organized.	**Data mining** is like finding treasure in that storage room by discovering patterns and useful insights. Both work together to help businesses make smarter decisions.

Data Mining

- **Data Mining** is the process of extracting information from the company's various databases and re-organizing it for purposes other than what the databases were originally intended for.
- It provides a means of extracting previously unknown, predictive information from the base of accessible data in data warehouses.
- Data mining process is different for different organizations depending upon the nature of the data and organization.
- Data mining tools use sophisticated, automated algorithms to discover hidden patterns, correlations, and relationships among organizational data.
- Data mining tools are used to predict future trends and behaviors, allowing businesses to make proactive, knowledge driven decisions.
- For ex: for targeted marketing, data mining can use data on past promotional mailings to identify the targets most likely to maximize the return on the company's investment in future mailings.

Functions

- Classification: It infers the defining characteristics of a certain group
- Clustering: identifies group of items that share a particular characteristic
- Association: identifies relationships between events that occur at one time
- Sequencing: similar to association, except that the relationship exists over a period of time
- Forecasting: estimates future values based on patterns within large sets of data

Common data mining applications

APPLICATION	DESCRIPTION
Market segmentation	Identifies the common characteristics of customers who buys the same products from the company
Customer churn	Predicts which customers are likely to leave your company and go to a competitor
Fraud detection	Identifies which transactions are most likely to be fraudulent
Direct marketing	Identifies which prospects should be included in a mailing list to obtain the highest response rate
Market based analysis	Understands what products or services are commonly purchased together
Trend analysis	Reveals the difference between a typical customer this month versus last month
Science	Simulates nuclear explosions; visualizes quantum physics

Entertainment	Models customer flows in theme parks; analyzes safety of amusement parks rides
Insurance and health care	Predicts which customers will buy new policies; identifies behavior patterns that increase insurance risk; spots fraudulent claims
Manufacturing	Optimizes product design, balancing manufacturability and safety; improves shop-floor scheduling and machine utilization
Medicine	Ranks successful therapies for different illnesses; predicts drug efficacy; discovers new drugs and treatments
Oil and gas	Analyzes seismic data for signs of underground deposits ; prioritizes drilling locations; simulates underground flows to improve recovery
Retailing	Discerns buying-behavior patterns; predicts how customers will respond to marketing campaigns

Data Mining works with Data Warehouse

- *Data Warehousing provides the Enterprise with a memory*

- *Data Mining provides the Enterprise with intelligence*

Data mining for decision support

Two capabilities are provided new business opportunities

- Automated prediction of trends and behavior: for ex, targeted marketing.

- Automated discovery of previously unknown patterns: for ex, detecting fraudulent credit card transactions and identifying anomalous data representing data entry-keying errors.

Datawarehouse Components

There are mainly five components of Data Warehouse:

1. Data Warehouse Database
2. Extract-Transform-Load (ETL) Tools
3. Metadata
4. Data Warehouse Access Tools
5. Data Mart

What is a Data Warehouse Database?

A **data warehouse database** is a centralized repository designed specifically for **storing, organizing, and analyzing large volumes of data** from multiple sources. It is optimized for **query performance, historical analysis, and decision-making** rather than transaction processing.

Characteristics:

Subject-oriented: Focused on specific business areas like sales, finance, or customer data.

Integrated: Combines data from various sources into a unified format.

Non-volatile: Data is stable and doesn't change frequently; historical data is preserved.

Time-variant: Tracks changes over time to enable trend analysis.

Types of Data Warehouse Databases

1.Enterprise Data Warehouse (EDW)

Description: A centralized database that stores all the data for an organization, providing a unified source for enterprise-wide analytics.

Use Case: Large organizations needing a comprehensive view of all business operations.

2.Operational Data Store (ODS)

Description: A type of database designed for operational reporting and is often used as an intermediate stage before data moves to the main warehouse.

Use Case: Real-time or near-real-time reporting, such as current inventory or customer orders.

3. Data Mart

Description: A smaller, specialized subset of a data warehouse designed for a specific department or business unit.

Use Case: When specific teams (e.g., marketing, finance) need tailored data for analysis.

Extract-Transform-Load (ETL) Tools

Often considered the backbone of data warehousing, you will need an ETL tool to extract data from disparate source systems across the enterprise, transform this data to convert it in a format suited for your data warehouse, and load it into your data warehouse.

Metadata

- Metadata is data about data which defines the data warehouse. Metadata refers to data that defines the data warehouse and provide context to data.
- A record in your customer database may look like:

  Ram 76 13,000 94,923.00

This data is not comprehensible unless you review associated metadata:

Customer Name: Ram

Purchase ID: 76

Purchase Quantity: 13,000

Order Amount: 94,923.00

Functions of Metadata

1. **Data Integration and ETL (Extract, Transform, Load)**

   Metadata defines how data is extracted from source systems, transformed, and loaded into the warehouse.

   Example:

   Metadata specifies that "OrderDate" from the sales database should be transformed into the "YYYY-MM-DD" format before loading into the warehouse.

Functions of Metadata

2.Data Mapping

Provides a clear map of the relationship between source systems and data warehouse tables.

Example:

Maps "CustomerID" in the CRM system to "CustID" in the data warehouse.

Functions of Metadata

3. Data Governance and Quality

Metadata ensures compliance with data governance policies and monitors data quality metrics such as accuracy, completeness, and consistency.

Example:

Metadata defines rules like "OrderID must be unique" or "ProductCode cannot be null."

Functions of Metadata

4. Query Optimization

Metadata provides information about data structures, indexes, and partitions, enabling efficient query execution.

Example:

Metadata indicates that the "Sales" table is partitioned by "Region" to speed up regional sales analysis queries.

Partition

- Partitioning in a data warehouse involves dividing a large table into smaller, more manageable pieces, called partitions, based on specific criteria. Partitioning improves query performance, manageability, and maintenance by allowing operations to focus only on relevant portions of the table.

Types of Table Partitioning in a Data Warehouse

1.Horizontal Partitioning

- Divides data into partitions based on rows.
- Each partition contains a subset of the rows from the table, often based on a key column such as date, region, or department.
- Example:
  - A sales table partitioned by sales_year will have separate partitions for 2020, 2021, 2022, etc.

Types of Table Partitioning in a Data Warehouse

2. Vertical Partitioning

- Divides data into partitions based on columns.
- Frequently used to separate less-accessed columns from frequently queried ones.
- Example:
  - Splitting a customer table into one table with frequently queried columns (e.g., customer_id, name, email) and another with less-used columns (e.g., address, preferences).

Types of Table Partitioning in a Data Warehouse

3. List Partitioning

- Divides data based on specific predefined values in a column.
- Example:
  - Partition a table based on region:
    - Partition 1: Data for North.
    - Partition 2: Data for South.
    - Partition 3: Data for East, West.

Types of Table Partitioning in a Data Warehouse

4. **Range Partitioning**
   - Divides data based on a range of values in a column.
   - Example:
     - A table partitioned by order_date:
       - Partition 1: Orders from 2022-01-01 to 2022-06-30.
       - Partition 2: Orders from 2022-07-01 to 2022-12-31.

Types of Table Partitioning in a Data Warehouse

5. Hash Partitioning

- Distributes data across partitions based on the result of a hash function applied to a column.
- Ensures even distribution, which is helpful for parallel processing.
- **Example:**
  - Partition a table by hashing customer_id to evenly distribute customer data.

. .

Types of Table Partitioning in a Data Warehouse

5. Composite Partitioning

- Combines two or more types of partitioning (e.g., range + hash).
- Example:
  - Range partitioning on order_date combined with hash partitioning on customer_id.

Benefits of Table Partitioning

1.Improved Query Performance

- Queries scan only the relevant partitions, reducing I/O and execution time.

2.Simplified Data Management

- Easier to archive, backup, and delete data at the partition level.

3.Parallel Processing

- Queries and operations can be distributed across partitions for faster execution.

4.Efficient Indexing

- Indexes can be created for individual partitions, making lookups more efficient.

5.Enhanced Data Availability

- Some partitioning schemes allow unaffected partitions to remain accessible if one partition is under maintenance.

When to Use Partitioning

- Tables with a large number of rows (millions or billions).
- Queries that frequently filter data by a specific column (e.g., date, region).
- Scenarios requiring regular archiving or deletion of old data.

Functions of Metadata

5. Business User Accessibility

Metadata acts as a dictionary that provides business users with understandable definitions of technical terms.

Example:

Metadata defines "Customer Lifetime Value (CLV)" as "Total revenue generated by a customer over their lifetime."

Functions of Metadata

6. Data Warehouse Administration

Helps administrators monitor and manage the data warehouse by providing details about table sizes, data refresh schedules, and user access logs.

Example:

Metadata shows that the "Sales" table is refreshed daily at 2 AM.

Types of Metadata in a Data Warehouse

1. Technical Metadata:

Describes the technical details of data, such as table structures, data types, relationships, and ETL workflows.

Example:

1. Table schema: "Orders(OrderID: INT, CustomerID: INT, OrderDate: DATE)."

Types of Metadata in a Data Warehouse

2. Business Metadata:

Provides context and meaning to the data from a business perspective.

Example:

"OrderID" represents a unique identifier for customer orders.

3. Operational Metadata:

Tracks information about data operations, such as data load times, refresh schedules, and error logs.

Example:

Metadata logs that the "Product" table was last refreshed on "2025-01-20" at 3:00 AM.

Data Warehouse Access Tools

At the backend, the data warehouse is built on top of a database or collection of databases. Business users cannot be expected to interact with databases, so tools designed to facilitate access to it are needed. The type of access tool you choose determines the level of access.

Data Mart

- Data mart can be defined as the subset of data warehouse of an organization which is limited to a specific business unit or group of users.
- A data mart is focused on a single functional area of an organization and contains a subset of data stored in a Data Warehouse.
- A data mart is a condensed version of Data Warehouse and is designed for use by a specific department, unit or set of users in an organization. E.g., Marketing, Sales, HR or finance. It is often controlled by a single department in an organization.
- Data Mart usually draws data from only a few sources compared to a Data warehouse. Data marts are small in size and are more flexible compared to a Datawarehouse.

What is a Data Warehouse Model?

- A **data warehouse model** defines how data is organized, stored, and managed in the data warehouse.
- It is a blueprint for structuring data to facilitate efficient querying, analysis, and reporting.
- Data warehouse models are designed to support analytical processing (OLAP) and are optimized for reading large volumes of historical data, aggregations, and multidimensional analysis.

Data Warehouse Models

Data warehouse models:

1. Enterprise Warehouse Models:

"An enterprise warehouse models collects ***all of the information*** about subjects spanning the entire organization".

- ✓ It provides ***corporate-wide data integration***, usually one or more operational systems or external information providers and is ***cross-functional*** scope.
- ✓ It consists of ***summarized data*** and its range in size from gigabytes, terabytes or beyond.
- ✓ It is implemented on ***traditional mainframes***, computer super servers, parallel architecture platforms.
- ✓ It takes years to ***design and build***.

Data Warehouse Models

2. Data Mart:

"A data mart contains a ***subset of corporate-wide data*** that is of value to a specific group of users".

(E.g.) ***Marketing data mart*** contains details about customer, item and sales.

- ✓ It is implemented on ***low – cost*** departmental servers (i.e.) Unix / Linux.
- ✓ It involve ***complex integration*** in long run.

It is categorized into two source of data –

i) Independent data marts – data captured from ***one or more operational systems*** or external information or geographic area.

ii) Dependent data marts – it is sourced directly from ***enterprise data warehouses.***

Virtual Data Warehouse (VDW)

- A **Virtual Data Warehouse (VDW)** is a type of data warehouse architecture where the data remains in its original source systems and is not physically stored in a centralized repository. Instead, it provides a unified and integrated view of the data by querying the underlying data sources in real-time or near-real-time.
- This model acts as a logical data warehouse layer, enabling users to access and analyze data without the need for time-consuming Extract, Transform, Load (ETL) processes or physical data duplication.

Virtual Data Warehouse (VDW)

Key Characteristics of a Virtual Data Warehouse

1. **No Centralized Storage**:
   - Data is not copied or moved to a central repository.
   - Queries are executed on the original data sources.
2. **Real-Time Data Access**:
   - Provides up-to-date information by directly querying the data sources.

3. Logical Integration:

- Uses metadata to map and integrate data from multiple sources into a single view.

4.Flexibility:

- Easy to adapt to changes in data sources or business requirements.

5.Simplified ETL:

- Reduces or eliminates the need for complex ETL processes.

VDW Vs TDW

Aspect	Virtual Data Warehouse	Traditional Data Warehouse
Storage	No physical storage; data remains in sources	Centralized physical repository
Performance	May suffer due to real-time integration	Optimized for query performance
Real-Time Access	Provides up-to-date data	Data may be outdated due to ETL schedules
Cost	Lower cost due to no storage requirements	Higher due to storage and ETL processes
Implementation Time	Faster	Slower due to ETL and schema design

Types of Data Warehouse

4. Cloud Data Warehouse

Description:

A modern data warehouse hosted on the cloud, offering scalability, flexibility, and cost efficiency.

Examples:

Snowflake, Amazon Redshift, Google BigQuery, and Azure Synapse Analytics.

Types of Data Warehouse

5. Federated Data Warehouse

Description:

A virtual data warehouse that integrates data from multiple, distributed systems without physically storing all the data in one place.

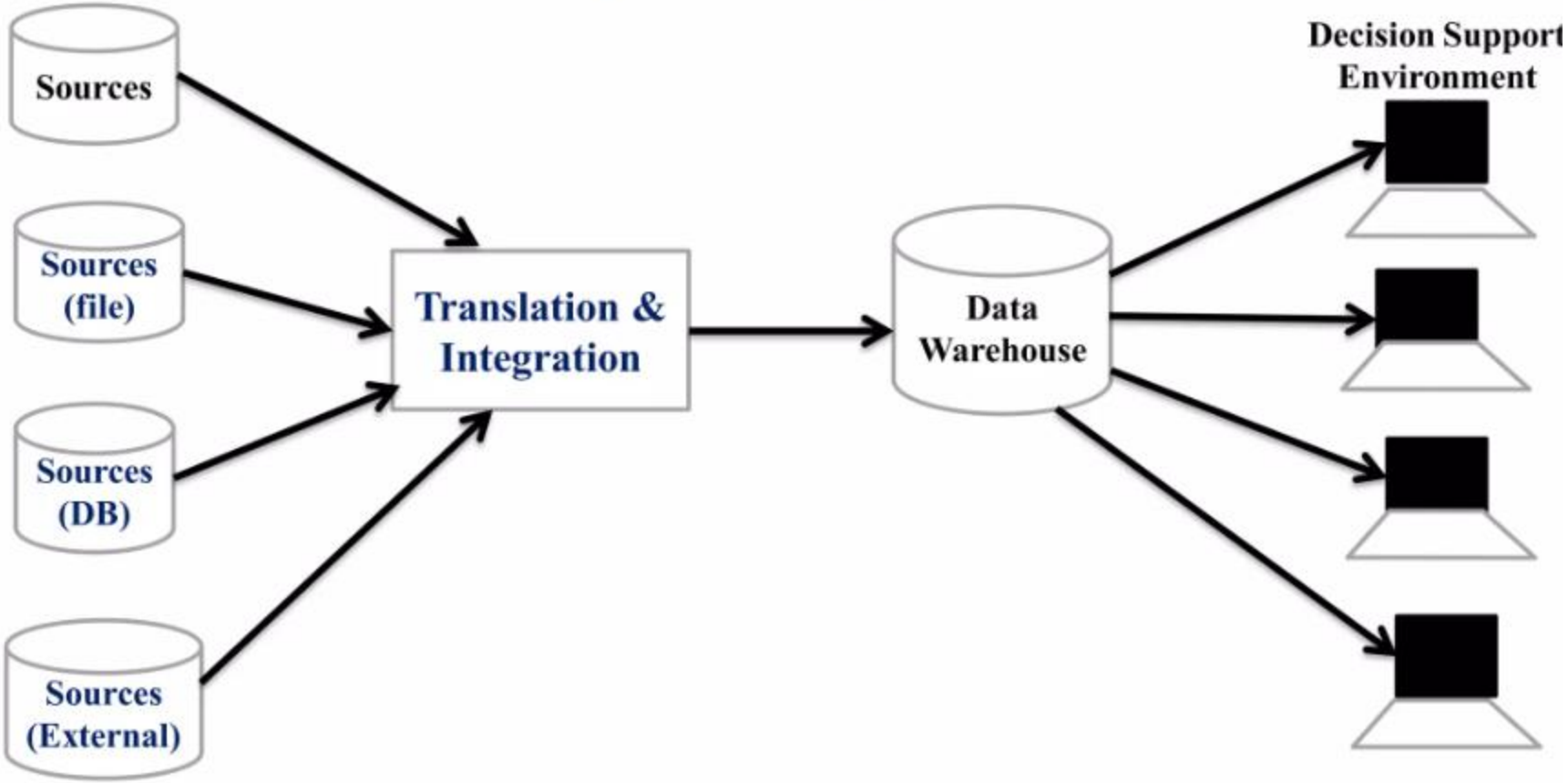
Use Case:

Organizations that want to avoid duplicating data but still require a unified view.

Data Warehouse system has another name

Two Tier Architecture

2. Generic Two Level (Two Tier):

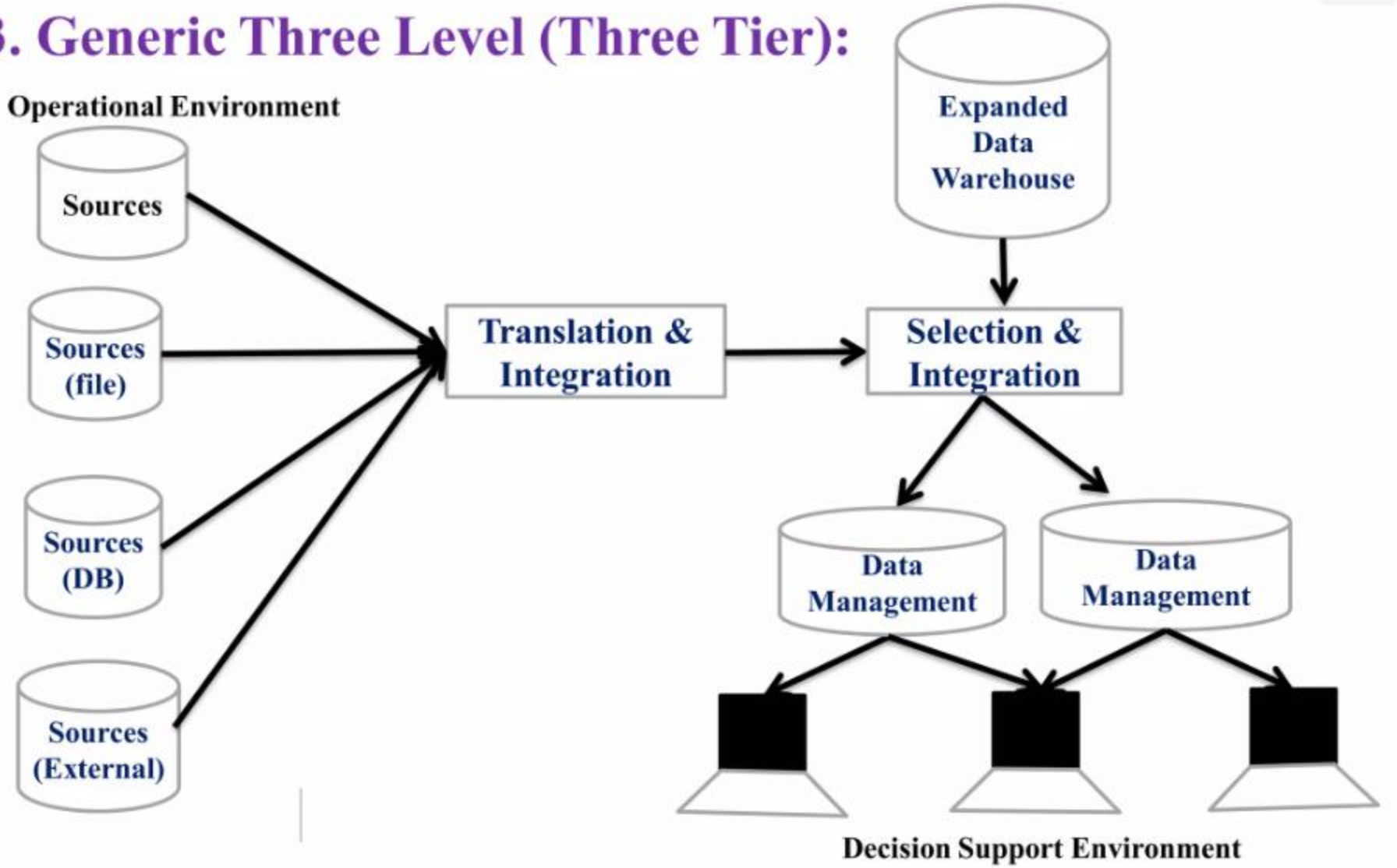


Two Tier (Level) Architecture:

- Two-layer architecture is one of the Data Warehouse layers which separates physically available ***sources and data warehouse***. This architecture is ***not expandable*** and also not supporting a large number of end-users. It also has connectivity problems because of network limitations.
- A two-tier architecture includes a ***staging area*** for all data sources, before the data warehouse layer. By adding a staging area between the sources and the storage repository, you ensure all data loaded into the warehouse is ***cleansed*** and in the appropriate format.

Three Tier Architecture

3. Generic Three Level (Three Tier):



Three Tier (Level) Architecture:

- This is the most widely used Architecture of Data Warehouse.

It consists of the Top, Middle and Bottom Tier.

a) Bottom Tier:

- It is the ***database*** of the Data warehouse servers.
- It is usually a ***relational database*** system.
- Data is ***cleansed, transformed***, and loaded into this layer using back-end tools.

It includes –

i) Data Extraction – get data from multiple, heterogeneous, and external sources.

ii) Data cleaning - detect errors in the data and rectify them when possible.

iii) Data transformation - convert data from legacy or host format to warehouse format.

iv) Load - sort, summarize, consolidate, compute views, check integrity, and build indices and partitions.

v) Refresh - propagate the updates from the data sources to the warehouse.

Load

The **load** phase involves **inserting or updating data into the data warehouse** from one or more data sources. This is a part of the broader ETL (Extract, Transform, Load) or ELT (Extract, Load, Transform) process.

Types of Load Operations:

Initial Load: The first-time loading of historical or baseline data into the warehouse.

Incremental Load: Periodically adding only new or updated data to the warehouse to keep it current.

Full Reload: Replacing all the data in the data warehouse with the latest data (less common and typically used when the dataset is small).

Refresh

The **refresh** phase ensures that the data in the warehouse stays **up-to-date and synchronized with the source systems**. It often refers to updating specific datasets or parts of the data warehouse rather than a full load.

Refresh Strategies:

Batch Refresh: Periodic updates based on a pre-set schedule (e.g., nightly or hourly batch jobs).

Real-time Refresh: Continuous updates as data changes in the source system, often using streaming data pipelines.

On-Demand Refresh: Triggered manually or automatically in response to specific needs or events.

Difference Between Load and Refresh

Aspect	Load	Refresh
Purpose	Moves data into the warehouse for the first time.	Keeps existing data up-to-date.
Scope	Involves larger-scale data insertion or updates.	Typically involves smaller, more targeted updates.
Frequency	Initial load or periodic incremental updates.	Can be batch-based, real-time, or event-driven.
Complexity	May involve significant transformations.	Primarily focuses on synchronization.

b) Middle Tier:

- It is the ***application layer*** giving an ***abstracted view*** of the database.
- The middle tier in Data warehouse is an OLAP server which is implemented using either ***ROLAP or MOLAP*** model.
- This layer also acts as a mediator between the end-user and the database.
- It arranges the data to make it more ***suitable for analysis***.

c) Top-Tier:

- It represent the ***front-end client layer***.
- It is where the user ***accesses and interacts*** with the data.
- It could be ***Query tools, reporting tools***, managed query tools, ***Analysis tools*** and Data mining tools.

Some ***popular data warehouse tools*** are -

- ✓ Xplenty.
- ✓ Amazon Redshift.
- ✓ Teradata.
- ✓ Oracle 12c.
- ✓ Informatica.
- ✓ IBM Infosphere.
- ✓ Cloudera.
- ✓ Panoply.