## Unit-I

### What Is a Data Warehouse?

A data warehouse is a database designed to enable business intelligence activities: it exists to help users understand and enhance their organization's performance. It is designed for query and analysis rather than for transaction processing, and usually contains historical data derived from transaction data, but can include data from other sources. Data warehouses separate analysis workload from transaction workload and enable an organization to consolidate data from several sources. This helps in:

- Maintaining historical records
- Analyzing the data to gain a better understanding of the business and to improve the business

In addition to a relational database, a data warehouse environment can include an extraction, transportation, transformation, and loading (ETL) solution, statistical analysis, reporting, data mining capabilities, client analysis tools, and other applications that manage the process of gathering data, transforming it into useful, actionable information, and delivering it to business users.

To achieve the goal of enhanced business intelligence, the data warehouse works with data collected from multiple sources. The source data may come from internally developed systems, purchased applications, third-party data syndicators and other sources. It may involve transactions, production, marketing, human resources and more. In today's world of big data, the data may be many billions of individual clicks on web sites or the massive data streams from sensors built into complex machinery.

Data warehouses are distinct from online transaction processing (OLTP) systems. With a data warehouse you separate analysis workload from transaction workload. Thus data warehouses are very much read-oriented systems. They have a far higher amount of data reading versus writing and updating. This enables far better analytical performance and avoids impacting your transaction systems. A data warehouse system can be optimized to consolidate data from many sources to achieve a key goal: it becomes your organization's "single source of truth". There is great value in having a consistent source of data that all users can look to; it prevents many disputes and enhances decision-making efficiency.

A data warehouse usually stores many months or years of data to support historical analysis. The data in a data warehouse is typically loaded through an extraction, transformation, and loading (ETL) process from multiple data sources. Modern data warehouses are moving toward an extract, load, transformation (ELT) architecture in which all or most data transformation is performed on the database that hosts the data warehouse. It is important to note that defining the ETL process is a very large part of the design effort of a data warehouse. Similarly, the speed and reliability of ETL operations are the foundation of the data warehouse once it is up and running.

Users of the data warehouse perform data analyses that are often time-related. Examples include consolidation of last year's sales figures, inventory analysis, and profit by product and by customer. But time-focused or not, users want to "slice and dice" their data however they see fit and a well-designed data warehouse will be flexible enough to meet those demands. Users will sometimes need highly aggregated data, and other times they will need to drill down to details. More sophisticated analyses include trend analyses and data mining, which use existing data to forecast trends or predict futures. The data warehouse acts as the underlying engine used by middleware business intelligence environments that serve reports, dashboards and other interfaces to end users.

Although the discussion above has focused on the term "data warehouse", there are two other important terms that need to be mentioned. These are the data mart and the operation data store (ODS).

A data mart serves the same role as a data warehouse, but it is intentionally limited in scope. It may serve one particular department or line of business. The advantage of a data mart versus a data warehouse is that it can be created much faster due to its limited coverage. However, data marts also create problems with inconsistency. It takes tight discipline to keep data and calculation definitions consistent across data marts. This problem has been widely recognized, so data marts exist in two styles. Independent data marts are those which are fed directly from source data. They can turn into islands of inconsistent information. Dependent data marts are fed from an existing data warehouse. Dependent data marts can avoid the problems of inconsistency, but they require that an enterprise-level data warehouse already exist.

Operational data stores exist to support daily operations. The ODS data is cleaned and validated, but it is not historically deep: it may be just the data for the current day. Rather than support the historically rich queries that a data warehouse can handle, the ODS gives data warehouses a place to get access to the most current data, which has not yet been loaded into the data warehouse. The ODS may also be used as a source to load the data warehouse. As data warehousing loading techniques have become more advanced, data warehouses may have less need for ODS as a source for loading data. Instead, constant trickle-feed systems can load the data warehouse in near real time.

A common way of introducing data warehousing is to refer to the characteristics of a data warehouse as set forth by William Inmon:

- Subject Oriented
- Integrated
- Nonvolatile
- Time Varient

**Subject Oriented**

Data warehouses are designed to help you analyze data. For example, to learn more about your company's sales data, you can build a data warehouse that concentrates on sales. Using this data warehouse, you can answer questions such as "Who was our best customer for this item last year?" or "Who is likely to be our best customer next year?" This ability to define a data warehouse by subject matter, sales in this case, makes the data warehouse subject oriented.

**Integrated**

Integration is closely related to subject orientation. Data warehouses must put data from disparate sources into a consistent format. They must resolve such problems as naming conflicts and inconsistencies among units of measure. When they achieve this, they are said to be integrated.

**Nonvolatile**

Nonvolatile means that, once entered into the data warehouse, data should not change. This is logical because the purpose of a data warehouse is to enable you to analyze what has occurred.

**Time Varient**

A data warehouse's focus on change over time is what is meant by the term time variant. In order to discover trends and identify hidden patterns and relationships in business, analysts need large amounts of data. This is very much in contrast to online transaction processing (OLTP) systems, where performance requirements demand that historical data be moved to an archive.

**Key Characteristics of a Data Warehouse**

The key characteristics of a data warehouse are as follows:

- Data is structured for simplicity of access and high-speed query performance.
- End users are time-sensitive and desire speed-of-thought response times.
- Large amounts of historical data are used.
- Queries often retrieve large amounts of data, perhaps many thousands of rows.
- Both predefined and ad hoc queries are common.
- The data load involves multiple sources and transformations.

In general, fast query performance with high data throughput is the key to a successful data warehouse.

**Contrasting OLTP and Data Warehousing Environments**

There are important differences between an OLTP system and a data warehouse. One major difference between the types of system is that data warehouses are not exclusively in third normal form (3NF), a type of data normalization common in OLTP environments.

Data warehouses and OLTP systems have very different requirements. Here are some examples of differences between typical data warehouses and OLTP systems:

- **Workload**

Data warehouses are designed to accommodate ad hoc queries and data analysis. You might not know the workload of your data warehouse in advance, so a data warehouse should be optimized to perform well for a wide variety of possible query and analytical operations.

OLTP systems support only predefined operations. Your applications might be specifically tuned or designed to support only these operations.

- **Data modifications**

A data warehouse is updated on a regular basis by the ETL process (run nightly or weekly) using bulk data modification techniques. The end users of a data warehouse do not directly update the data warehouse except when using analytical tools, such as data mining, to make predictions with associated probabilities, assign customers to market segments, and develop customer profiles.

In OLTP systems, end users routinely issue individual data modification statements to the database. The OLTP database is always up to date, and reflects the current state of each business transaction.

- **Schema design**

Data warehouses often use partially denormalized schemas to optimize query and analytical performance.

OLTP systems often use fully normalized schemas to optimize update/insert/delete performance, and to guarantee data consistency.

- **Typical operations**

A typical data warehouse query scans thousands or millions of rows. For example, "Find the total sales for all customers last month."

A typical OLTP operation accesses only a handful of records. For example, "Retrieve the current order for this customer."

- **Historical data**

Data warehouses usually store many months or years of data. This is to support historical analysis and reporting.

OLTP systems usually store data from only a few weeks or months. The OLTP system stores only historical data as needed to successfully meet the requirements of the current transaction.

**Common Data Warehouse Tasks**

As an Oracle data warehousing administrator or designer, you can expect to be involved in the following tasks:

- Configuring an Oracle database for use as a data warehouse
- Designing data warehouses
- Performing upgrades of the database and data warehousing software to new releases
- Managing schema objects, such as tables, indexes, and materialized views
- Managing users and security
- Developing routines used for the extraction, transformation, and loading (ETL) processes
- Creating reports based on the data in the data warehouse
- Backing up the data warehouse and performing recovery when necessary
- Monitoring the data warehouse's performance and taking preventive or corrective action as required

In a small-to-midsize data warehouse environment, you might be the sole person performing these tasks. In large, enterprise environments, the job is often divided among several DBAs and designers, each with their own specialty, such as database security or database tuning.

These tasks are illustrated in the following:

- For more information regarding partitioning, see Oracle Database VLDB and Partitioning Guide.

- For more information regarding database security, see Oracle Database Security Guide.

- For more information regarding database performance, see Oracle Database Performance Tuning Guide and Oracle Database SQL Tuning Guide.

- For more information regarding backup and recovery, see Oracle Database Backup and Recovery User's Guide.

- For more information regarding ODI, see Oracle Fusion Middleware Developer's Guide for Oracle Data Integrator.

## The best applications of Data Warehousing

Every organization, no matter in what industry it works in or how big or small it is, requires a data warehouse to connect its disparate sources for anticipating, analysis, reporting, business intelligence, and facilitating robust decision-making. These services are also required at a reasonable cost and optimal value. Here, we are listing down the best applications of data warehousing across different industries.

**E-commerce**: E-commerce platforms need to gather key marketing metrics (such as clicks, impressions, website visitors, etc.) from marketing tools and use that to approach their customers in a better way. This is where data warehouses help. Replicating data, tracking & visualizing KPIs such as conversion rates, churn rates, and return on ad spends, safe storage, etc. help companies perform better. In recent times, amazon redshift is the most popular warehouse being used for marketing analytics, because of its user-friendly UI and flexibility.

**Retail**: Data warehouses can be used by retailers to easily identify products with high demand and the fastest selling demand. The data can then be used to react to a rise or fall in consumer demand quickly, which can ultimately be used to gain a competitive advantage. Reverse ETL is a popular concept that leverages data from warehouses and helps target audiences better. They are the mediators between wholesalers and end customers, and that's why it is necessary for them to maintain the records of both parties. For helping them store data in an organized manner, the application of data warehousing comes into the frame.

**AI/ML**: With many companies embracing AI for their data journey, it's critical to get a reliable data warehouse now. AI enables data maturity, which is intertwined with the flexibility, scalability, and agility that a warehouse offers. On the other hand, machine learning is used on the data after the data has been replicated and transformed in the warehouse, to help newer business models emerge and advance digital disruption.

**Agritech:** Data storage is a must when it comes to the new age of farming. Data related to crop yield, weather conditions, pesticides, crop inventory and so much more demands a data warehouse. With advanced analytics, engineers and business analysts are able to figure out inefficiencies in the ecosystem, such as problems in the soil quality, unnecessary use of pesticides, etc. and iron them out.

**Sustainability and climate action**: Climate data requires a versatile data infrastructure, with cloud-first models for warehouses. To bring out sustainability insights, the data architecture must be able to integrate raw data from multiple sources and make it easy for end-users for making predictions and effective decisions related to climate change.

**Manufacturing & Supply chain**: Being one of the industries involving a higher number of intermediaries, the supply chain industry needs data warehouses to limit the number of data silos and ultimately human error. Data warehouses can help in inventory management (which items are low in count and what is the cost of each step in the life cycle), all the data related to vendors, logistics (for example: timestamp data related to product delivery), and ultimately serving the customer better.

**Healthcare**: With constant advancement in healthcare, the data captured by machines is huge. To digitally improve hospital infrastructure, reduce wait time, and make processes more efficient, data warehouses are making data work constantly in this field. Getting personalized healthcare can be possible with a single platform (such as having one place for all diagnostics, tests, prescriptions, and follow-ups). All the clinical, financial, and employee data are stored in the warehouse, and analysis is run to derive valuable insights to strategize resources in the best way possible.

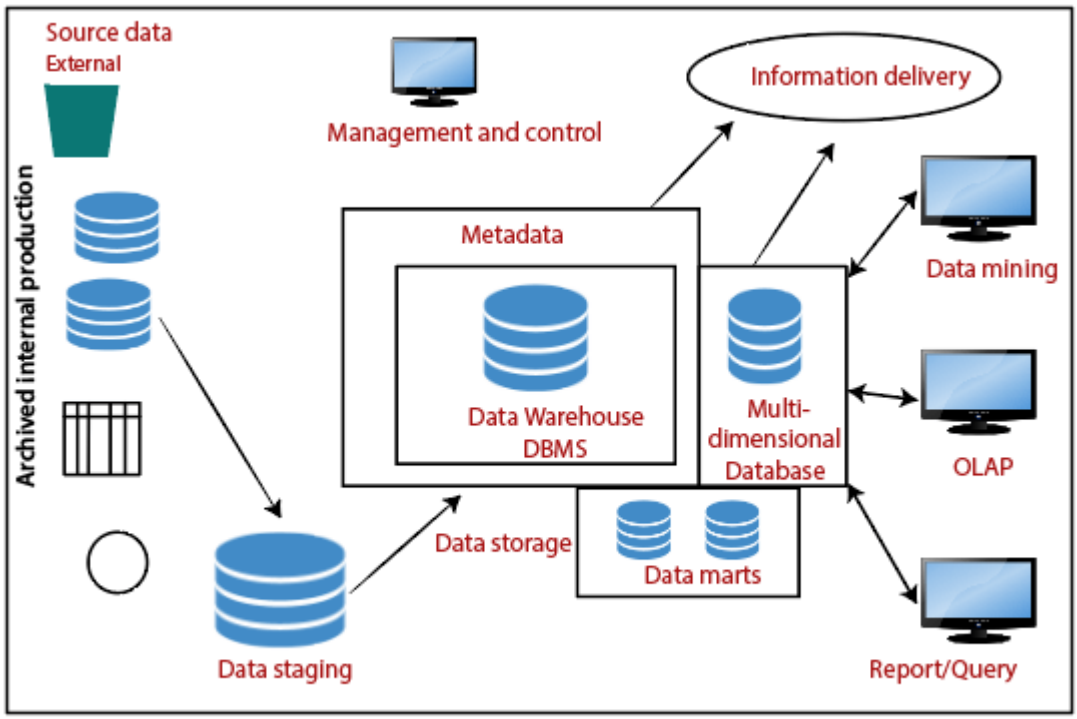
**Banking & Finance**: Data security is critical for the BFSI sector, and data warehouses solve that problem by vouching for industry-standard security compliances. The warehouses can be used to get updates about customer deposits, loans, funds, deposits, etc., and a better understanding of the performance of different branches. The right solution helps the financing industry analyze customer expenses that enable them to outline better strategies to maximize profits at both ends.

**Financial Auditing**: With access to real-time financial data, warehouses ensure decisions related to the business's current financial performance can be reached quickly. Data warehouses enable the collection of data on a daily basis and information can then be regularly used to identify any discrepancies in financial reporting & audits.

**Pharmaceuticals**: As data warehouses make data more accessible, it's now being used for making better strategic decisions and identifying & developing customer buying trends in pharmaceuticals. This results in better customer targeting, pre-call analysis as well as post-call assessments, helping the pharma industry at scale.

## Components or Building Blocks of Data Warehouse

Architecture is the proper arrangement of the elements. We build a data warehouse with software and hardware components. To suit the requirements of our organizations, we arrange these building we may want to boost up another part with extra tools and services. All of these depends on our circumstances.




**Components or Building Blocks of Data Warehouse**

The figure shows the essential elements of a typical warehouse. We see the Source Data component shows on the left. The Data staging element serves as the next building block. In the middle, we see the Data Storage component that handles the data warehouses data. This element not only stores and manages the data; it also keeps track of data using the metadata repository. The Information Delivery component shows on the right consists of all the different ways of making the information from the data warehouses available to the users.

**Source Data Component**

Source data coming into the data warehouses may be grouped into four broad categories:

**Production Data**: This type of data comes from the different operating systems of the enterprise. Based on the data requirements in the data warehouse, we choose segments of the data from the various operational modes.

**Internal Data**: In each organization, the client keeps their "private" spreadsheets, reports, customer profiles, and sometimes even department databases. This is the internal data, part of which could be useful in a data warehouse.

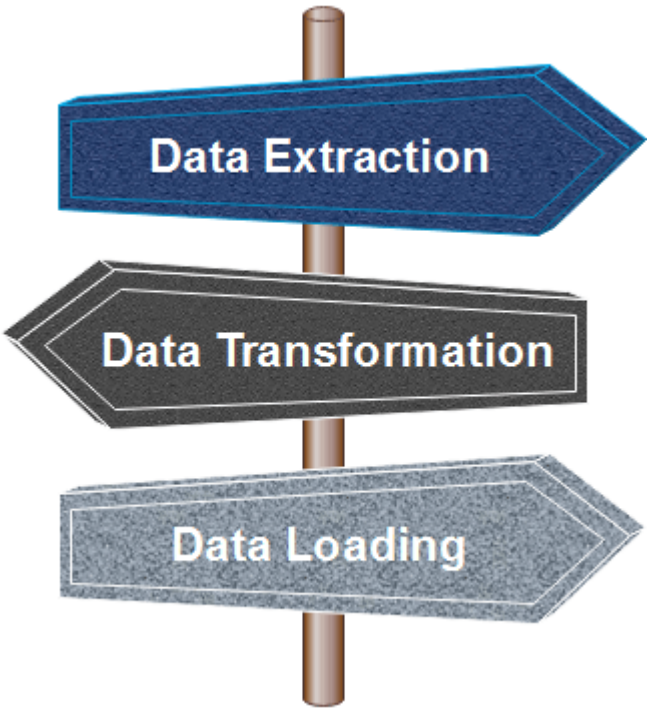
**Archived Data**: Operational systems are mainly intended to run the current business. In every operational system, we periodically take the old data and store it in achieved files.

**External Data:** Most executives depend on information from external sources for a large percentage of the information they use. They use statistics associating to their industry produced by the external department.

**Data Staging Component**

After we have been extracted data from various operational systems and external sources, we have to prepare the files for storing in the data warehouse. The extracted data coming from several different

sources need to be changed, converted, and made ready in a format that is relevant to be saved for querying and analysis.




1) **Data Extraction**: This method has to deal with numerous data sources. We have to employ the appropriate techniques for each data source.

2) **Data Transformation**: As we know, data for a data warehouse comes from many different sources. If data extraction for a data warehouse posture big challenges, data transformation present even significant challenges. We perform several individual tasks as part of data transformation.

First, we clean the data extracted from each source. Cleaning may be the correction of misspellings or may deal with providing default values for missing data elements, or elimination of duplicates when we bring in the same data from various source systems.

Standardization of data components forms a large part of data transformation. Data transformation contains many forms of combining pieces of data from different sources. We combine data from single source record or related data parts from many source records.

On the other hand, data transformation also contains purging source data that is not useful and separating outsource records into new combinations. Sorting and merging of data take place on a large scale in the data staging area. When the data transformation function ends, we have a collection of integrated data that is cleaned, standardized, and summarized.

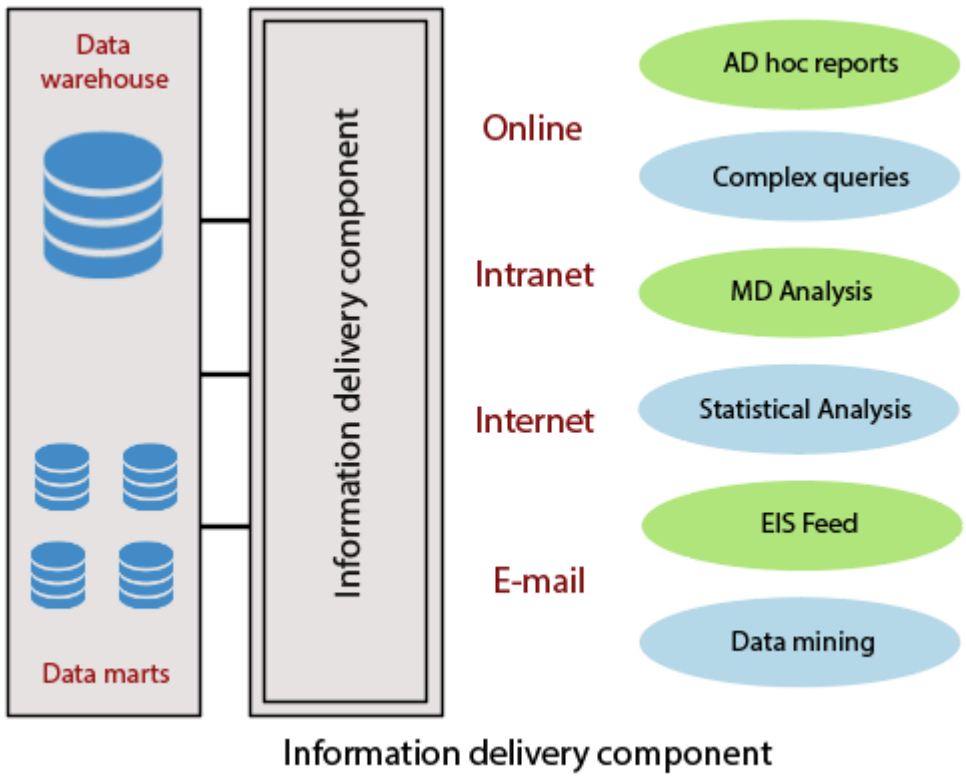
**3) Data Loading**: Two distinct categories of tasks form data loading functions. When we complete the structure and construction of the data warehouse and go live for the first time, we do the initial loading of the information into the data warehouse storage. The initial load moves high volumes of data using up a substantial amount of time.

**Data Storage Components**

Data storage for the data warehousing is a split repository. The data repositories for the operational systems generally include only the current data. Also, these data repositories include the data structured in highly normalized for fast and efficient processing.

**Information Delivery Component**

The information delivery element is used to enable the process of subscribing for data warehouse files and having it transferred to one or more destinations according to some customer-specified scheduling algorithm.



Information delivery component

**Metadata Component**

Metadata in a data warehouse is equal to the data dictionary or the data catalog in a database management system. In the data dictionary, we keep the data about the logical data structures, the data about the records and addresses, the information about the indexes, and so on.

**Data Marts**

It includes a subset of corporate-wide data that is of value to a specific group of users. The scope is confined to particular selected subjects. Data in a data warehouse should be a fairly current, but not mainly up to the minute, although development in the data warehouse industry has made standard and incremental data dumps more achievable. Data marts are lower than data warehouses and usually contain organization. The current trends in data warehousing are to developed a data warehouse with several smaller related data marts for particular kinds of queries and reports.

**Management and Control Component**

The management and control elements coordinate the services and functions within the data warehouse. These components control the data transformation and the data transfer into the data warehouse storage. On the other hand, it moderates the data delivery to the clients. Its work with the database management systems and authorizes data to be correctly saved in the repositories. It monitors the movement of

information into the staging method and from there into the data warehouses storage itself.

**Why we need a separate Data Warehouse?**

Data Warehouse queries are complex because they involve the computation of large groups of data at summarized levels.

It may require the use of distinctive data organization, access, and implementation method based on multidimensional views.

Performing OLAP queries in operational database degrade the performance of functional tasks.

Data Warehouse is used for analysis and decision making in which extensive database is required, including historical data, which operational database does not typically maintain.

The separation of an operational database from data warehouses is based on the different structures and uses of data in these systems.

Because the two systems provide different functionalities and require different kinds of data, it is necessary to maintain separate databases.

**Difference between Database and Data Warehouse**

| Database | Data Warehouse |
|---|---|
| 1. It is used for Online Transactional Processing (OLTP) but can be used for other objectives such as Data Warehousing. This records the data from the clients for history. | 1. It is used for Online Analytical Processing (OLAP). This reads the historical information for the customers for business decisions. |
| 2. The tables and joins are complicated since they are normalized for RDBMS. This is done to reduce redundant files and to save storage space. | 2. The tables and joins are accessible since they are de-normalized. This is done to minimize the response time for analytical queries. |
| 3. Data is dynamic | 3. Data is largely static |
| 4. **Entity:** Relational modeling procedures are used for RDBMS database design. | 4. **Data:** Modeling approach are used for the Data Warehouse design. |
| 5. Optimized for write operations. | 5. Optimized for read operations. |
| 6. Performance is low for analysis queries. | 6. High performance for analytical queries. |
| 7. The database is the place where the data is taken as a base and managed to get available fast and efficient access. | 7. Data Warehouse is the place where the application data is handled for analysis and reporting objectives. |

**Unit-II**

## Data Warehouse Architectures

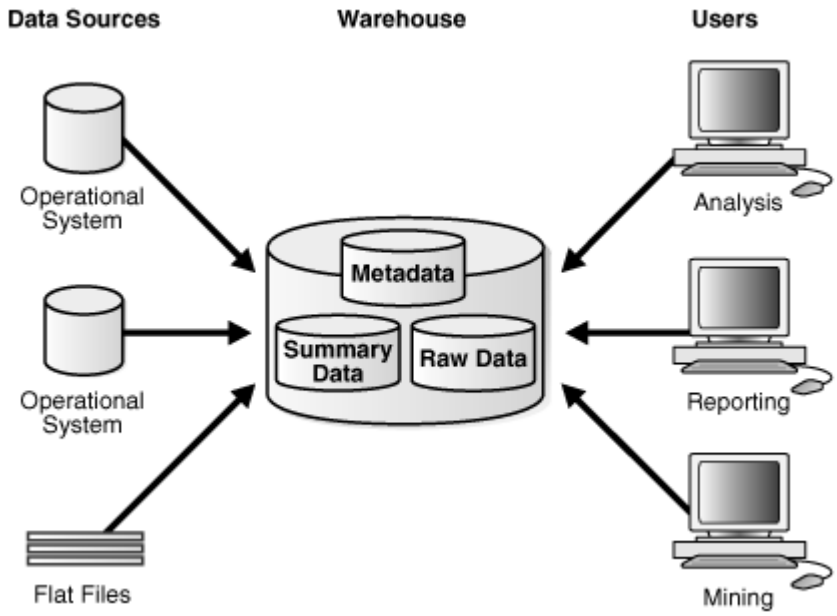
Data warehouses and their architectures vary depending upon the specifics of an organization's situation. Three common architectures are:

- Data Warehouse Architecture: Basic
- Data Warehouse Architecture: with a Staging Area
- Data Warehouse Architecture: with a Staging Area and Data Marts

## Data Warehouse Architecture: Basic

Figure 1-1 shows a simple architecture for a data warehouse. End users directly access data derived from several source systems through the data warehouse.

Figure 1-1 Architecture of a Data Warehouse




Description of "Figure 1-1 Architecture of a Data Warehouse"

In Figure 1-1, the metadata and raw data of a traditional OLTP system is present, as is an additional type of data, summary data. Summaries are a mechanism to pre-compute common expensive, long-running operations for sub-second data retrieval. For example, a typical data warehouse query is to retrieve something such as August sales. A summary in an Oracle database is called a materialized view.

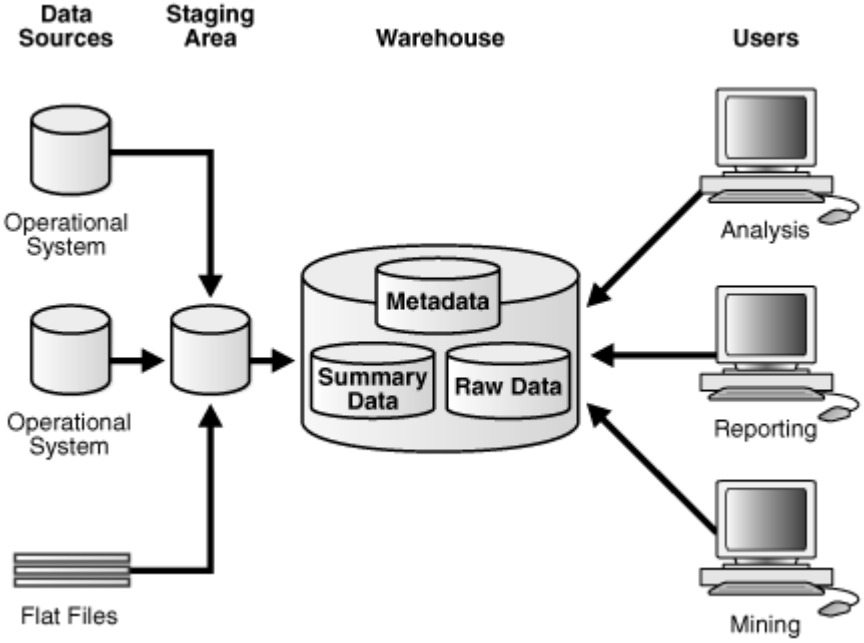
The consolidated storage of the raw data as the center of your data warehousing architecture is often referred to as an Enterprise Data Warehouse (EDW). An EDW provides a 360-degree view into the business of an organization by holding all relevant business information in the most detailed format.

### Data Warehouse Architecture: with a Staging Area

You must clean and process your operational data before putting it into the warehouse, as shown in Figure 1-2. You can do this programmatically, although most data warehouses use a staging area instead. A staging area simplifies data cleansing and consolidation for operational data coming from

multiple source systems, especially for enterprise data warehouses where all relevant information of an enterprise is consolidated. Figure 1-2 illustrates this typical architecture.

Figure 1-2 Architecture of a Data Warehouse with a Staging Area

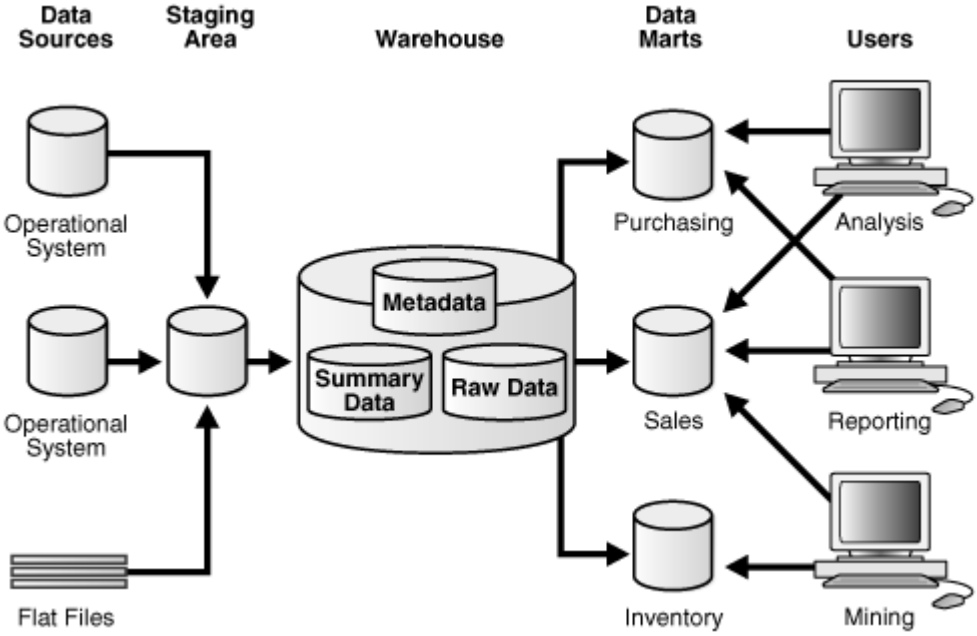



Description of "Figure 1-2 Architecture of a Data Warehouse with a Staging Area"

**Data Warehouse Architecture: with a Staging Area and Data Marts**

Although the architecture in Figure 1-2 is quite common, you may want to customize your warehouse's architecture for different groups within your organization. You can do this by adding data marts, which are systems designed for a particular line of business. Figure 1-3 illustrates an example where purchasing, sales, and inventories are separated. In this example, a financial analyst might want to analyze historical data for purchases and sales or mine historical data to make predictions about customer behavior.

Figure 1-3 Architecture of a Data Warehouse with a Staging Area and Data Marts




Description of "Figure 1-3 Architecture of a Data Warehouse with a Staging Area and Data Marts"

# GATHERING BUSINESS REQUIREMENTS FOR DATA WAREHOUSE

## 1. Platform Functions

These features establish a baseline for the system to operate around. Interactivity refers to the communication process between human users and the software and how easy the system is to use. Customizations and white labeling allow users to remake the software to their preferences and needs. This has the double benefit of a seamless experience with other software systems you might use and the assurance that your employees will actually use it.

- Interactive Visualization
- User-Friendly
- Platform Customization
- White Labeling

## 2. Scalability

Scalability is one of the most vital differentiators for a data warehouse solution. A robust solution scales rapidly to terabytes or even petabytes of data and concurrent users without downtimes or disruptions.

Elasticity refers to scaling up and down instantly to meet demands. Scale up rapidly to handle unexpected workloads, scale down just as quickly to reduce resources and expenses.

- Massive Storage Capacity
- Rapid Scalability
- Elasticity
- Scalable Concurrency
- Independently Sized

## 3. Performance Requirements

At the end of the day, your data warehouse should be able to handle huge workloads efficiently, utilize finite resources to deliver the best performance, parallelly process multiple queries, users and processes – enhancing analytics and business decisions.

An ideal solution lets you stream data in real time while sustaining ACID properties for transactions. Workload separation is essential for parallel processing, it refers to the proper balancing and prioritization of processes and users. Increasing data load throughput enables faster ETL processing while a lower latency leads to faster querying.

- Workload Separation
- Maximum Data Loading Throughput
- ACID Transactional Consistency
- Reduced Latency
- Maximum Concurrency

## 4. Data Visualization

Once data is organized in a data warehouse, it is ready to be visualized. This involves the system discovering trends and patterns in data sets and generating graphs, charts, scattergrams and other visual depictions.

Visualization makes complex statistical relations easy to interpret for users. Did you know that when we sit down to read a website, we only read an average of 28 percent of the words on the page? We skim, make assumptions and extrapolate based on the words we do read to glean information. That's one reason visual depictions are so much more effective at delivering information to our brains. Data visualization helps bridge that gap and offer information that sticks.

Storyboarding functions like a flowchart — it maps out the flow of data and insights in a linear narrative to make it easily digestible. The drag and drop feature lets users customize their dashboard at the click of a button and create personalized templates to meet their specific needs.

- Storyboarding
- Geospatial Integration
- Animations
- Barcodes
- Tables
- Charts and Graphs
- Infographics
- Filters
- Widgets
- Drag and Drop Creation
- Customization
- Templates
- Freehand SQL Command
- Layouts
- Themes

**5. Analytics**

The analytics portion of BI offers insights into your business processes by evaluating trends in data and applying predictions to them. Benchmarking compares business practices and performance to industry metrics in order to create action plans to improve your business.

Predictive analytics offers suggestions based on forecasts for future performance or data events. Social media analytics is pretty simply just what it sounds like — it tracks engagement, followers, traffic and other social media metrics to generate reports on your organization's social presence. Web analytics is similar but tracks metrics for your website.

Geolocation analysis measures the location of customers, traffic or other location-based metrics. These can be used to glean an understanding of customer demographics, improve services, optimize sales territories and more. Ad-hoc analysis is a report generated on a specific query for a single question or KPI; it can be custom-made or generated from a template.

Data mining is a subcategory of BI like data warehousing. It is the process of collecting the data from the database or warehouse in order to analyze it. In-memory analytics performs complex queries that would otherwise be done on physical disks within the RAM of the machine, increasing the speed of analysis.

Machine learning automates the model building process. It is a form of AI that allows systems to learn from previous data in order to identify patterns and reach conclusions without human interference.

- Benchmarking
- Predictive Analytics
- Social Media Analytics
- Web Analytics
- Geolocation Analysis
- Ad-Hoc Analysis
- Trend Indicators
- Profit Analysis
- In-Memory Analysis
- Statistic Analytics
- Data Mining
- Machine Learning

**6. OLAP**

Online analytical processing (or OLAP) is a process that performs multi-dimensional analysis on large, layered datasets. It drills down and explores data to offer users both detailed information on their daily operations and overviews of business trends. With this data, users can extrapolate predictions by changing variables and uncovering relations between them within the data.

- Multi-Dimensional Analysis
- Drill-Down
- Data Exploration
- Time-Series Auto Generation

**7. Document Management**

Reporting is another key tenet of BI, and what happens to those reports after they're generated all takes place in document management. Users can export reports and visualizations in a range of document formats to send to team members, investors and more with ease. Versioning and version control ensure that individual instances of a software solution (for example, the iOS on your iPhone when you bought it versus the most recent update) employ different versions of the product. This lets software programmers track changes and revert back to previous versions if a serious bug occurs.

- Export to Microsoft Excel
- Export to Microsoft Workbook
- Export to PDF
- Export to HTML
- Versioning

**8. Decision Services**

This module focuses on how users take the insights they derive from data and turn it into action. Financial management features offer forecasting and budgeting to help you achieve financial success. Regulatory compliance and threat/fraud detection capabilities ensure data security, alert you to suspicious activity and protect you during audits.

- Financial Management
- Regulatory Compliance
- Monitoring
- Threat/Fraud Detection
- Consulting Services

**9. Integrations**

While some BI tools restrict their users to proprietary architecture, more and more are offering a range of integrations with other kinds of software systems and data sources. For example, service-centered organizations need to be able to draw data directly from their CRM to generate reports and visualizations on that information.

Extract, transform, load (ETL) is also a crucial integration. ETL combines three database functions into a single tool in order to transfer data from one database to another.

Big data integration is also important — it enables large data set incorporation from sources like Hadoop, Hive, etc.

- ERP Integration
- ETL Integration
- Portal Integration
- CRM Integration
- MS Office Applications
- Big Data Connectors
- Hadoop
- Hive
- Hbase
- Cassandra
- MapReduce

## Data Warehouse Design

A data warehouse is a single data repository where a record from multiple data sources is integrated for online business analytical processing (OLAP). This implies a data warehouse needs to meet the requirements from all the business stages within the entire organization. Thus, data warehouse design is a hugely complex, lengthy, and hence error-prone process. Furthermore, business analytical functions change over time, which results in changes in the requirements for the systems. Therefore, data warehouse and OLAP systems are dynamic, and the design process is continuous.

Data warehouse design takes a method different from view materialization in the industries. It sees data warehouses as database systems with particular needs such as answering management related queries. The target of the design becomes how the record from multiple data sources should be extracted, transformed, and loaded (ETL) to be organized in a database as the data warehouse.

There are two approaches

1. "top-down" approach

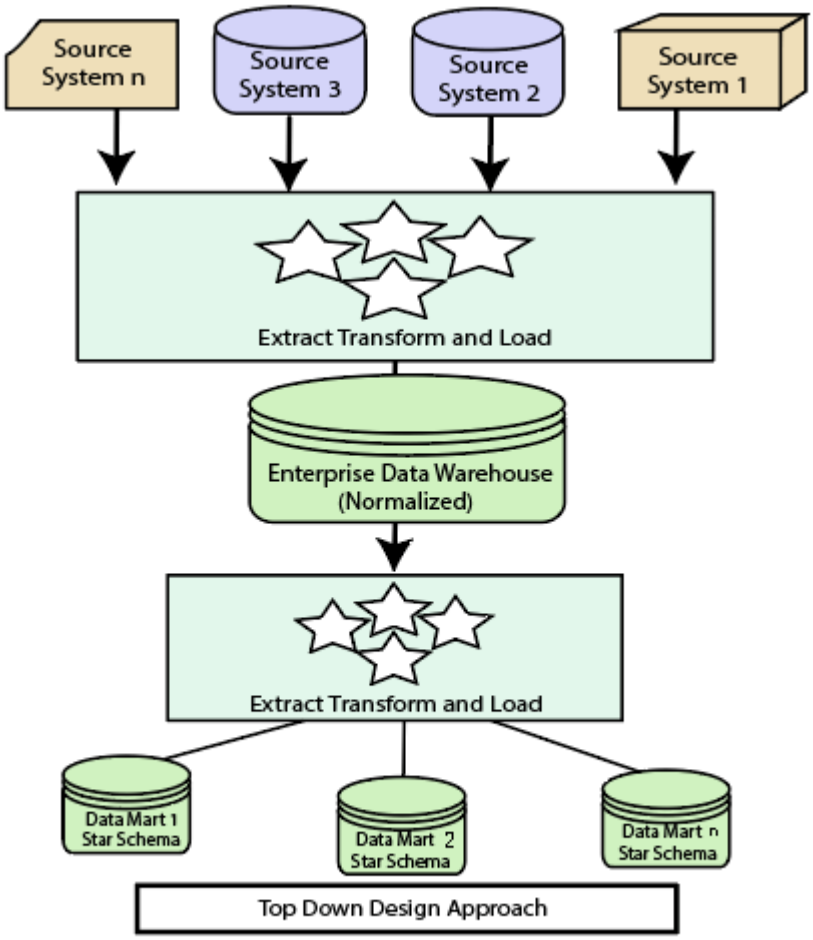
2. "bottom-up" approach

**Top-down Design Approach**

In the "Top-Down" design approach, a data warehouse is described as a subject-oriented, time-variant, non-volatile and integrated data repository for the entire enterprise data from different sources are validated, reformatted and saved in a normalized (up to 3NF) database as the data warehouse. The data warehouse stores "atomic" information, the data at the lowest level of granularity, from where dimensional data marts can be built by selecting the data required for specific business subjects or particular departments. An approach is a data-driven approach as the information is gathered and integrated first and then business requirements by subjects for building data marts are formulated. The advantage of this method is which it supports a single integrated data source. Thus data marts built from it will have consistency when they overlap.

**Advantages of top-down design**

- Data Marts are loaded from the data warehouses.
- Developing new data mart from the data warehouse is very easy.
- Disadvantages of top-down design
- This technique is inflexible to changing departmental needs.
- The cost of implementing the project is high.




Top Down Design Approach

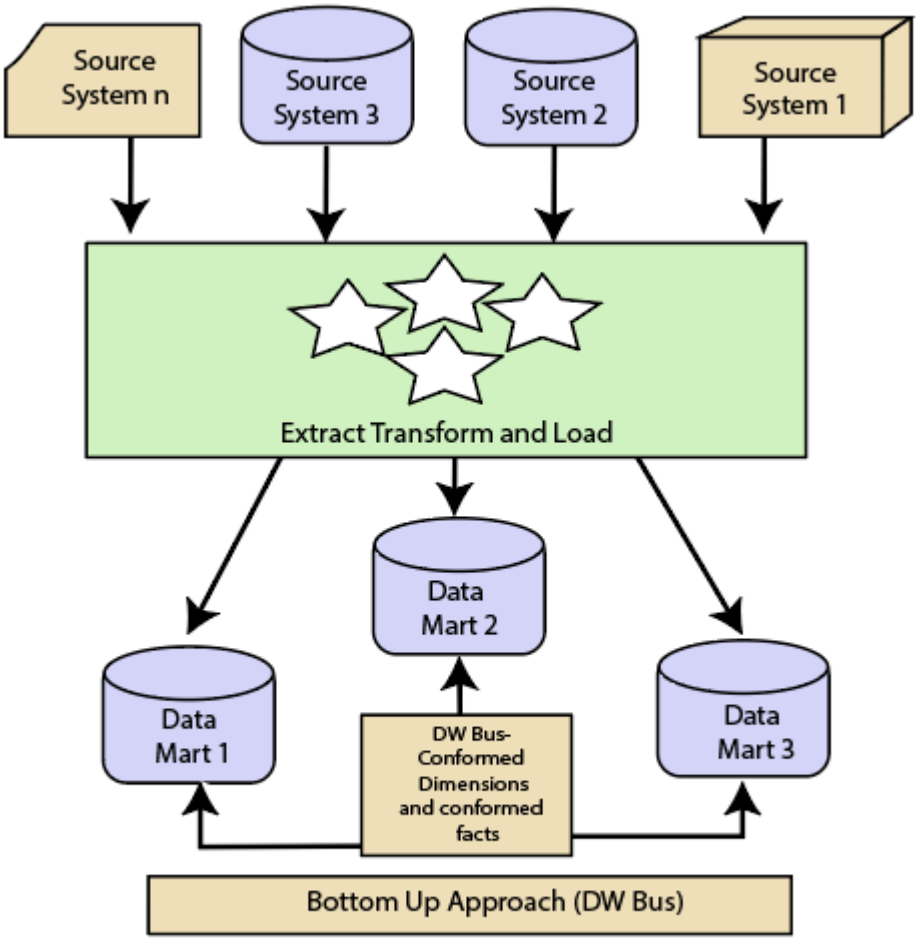
**Bottom-Up Design Approach**

In the "Bottom-Up" approach, a data warehouse is described as "a copy of transaction data specifical architecture for query and analysis," term the star schema. In this approach, a data mart is created first

to necessary reporting and analytical capabilities for particular business processes (or subjects). Thus it is needed to be a business-driven approach in contrast to Inmon's data-driven approach.

Data marts include the lowest grain data and, if needed, aggregated data too. Instead of a normalized database for the data warehouse, a denormalized dimensional database is adapted to meet the data delivery requirements of data warehouses. Using this method, to use the set of data marts as the enterprise data warehouse, data marts should be built with conformed dimensions in mind, defining that ordinary objects are represented the same in different data marts. The conformed dimensions connected the data marts to form a data warehouse, which is generally called a virtual data warehouse.

The advantage of the "bottom-up" design approach is that it has quick ROI, as developing a data mart, a data warehouse for a single subject, takes far less time and effort than developing an enterprise-wide data warehouse. Also, the risk of failure is even less. This method is inherently incremental. This method allows the project team to learn and grow.




Bottom Up Design Approach

**Advantages of bottom-up design**

Documents can be generated quickly.

The data warehouse can be extended to accommodate new business units.

It is just developing new data marts and then integrating with other data marts.

**Disadvantages of bottom-up design**

the locations of the data warehouse and the data marts are reversed in the bottom-up approach design.

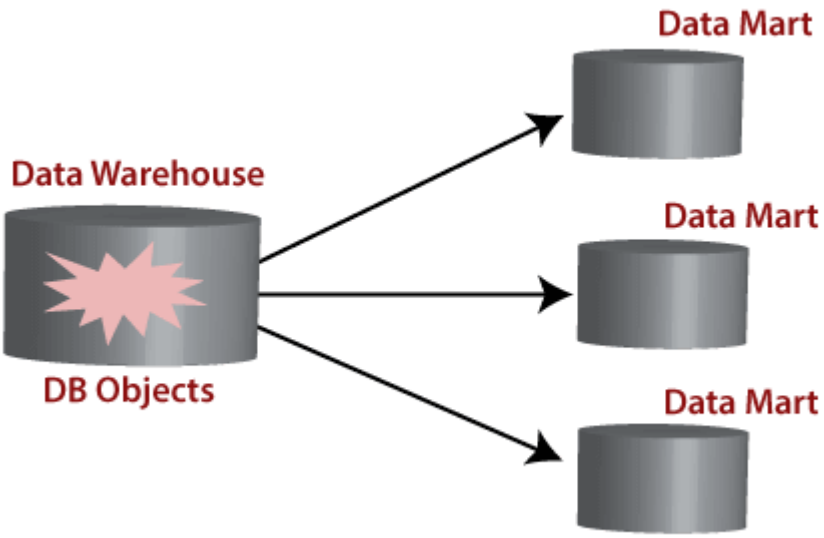
Differentiate between Top-Down Design Approach and Bottom-Up Design Approach

| Top-Down Design Approach | Bottom-Up Design Approach |
|---|---|
| Breaks the vast problem into smaller subproblems. | Solves the essential low-level problem and integrates them into a higher one. |
| Inherently architected- not a union of several data marts. | Inherently incremental; can schedule essential data marts first. |
| Single, central storage of information about the content. | Departmental information stored. |
| Centralized rules and control. | Departmental rules and control. |
| It includes redundant information. | Redundancy can be removed. |
| It may see quick results if implemented with repetitions. | Less risk of failure, favourable return on investment, and proof of techniques. |

**What is Data Mart?**

A **Data Mart** is a subset of a directorial information store, generally oriented to a specific purpose or primary data subject which may be distributed to provide business needs. Data Marts are analytical record stores designed to focus on particular business functions for a specific community within an organization. Data marts are derived from subsets of data in a data warehouse, though in the bottom-up data warehouse design methodology, the data warehouse is created from the union of organizational data marts.

The fundamental use of a data mart is **Business Intelligence (BI)** applications. **BI** is used to gather, store, access, and analyze record. It can be used by smaller businesses to utilize the data they have accumulated since it is less expensive than implementing a data warehouse.




**Reasons for creating a data mart**

- Creates collective data by a group of users
- Easy access to frequently needed data
- Ease of creation
- Improves end-user response time
- Lower cost than implementing a complete data warehouses
- Potential clients are more clearly defined than in a comprehensive data warehouse
- It contains only essential business data and is less cluttered.

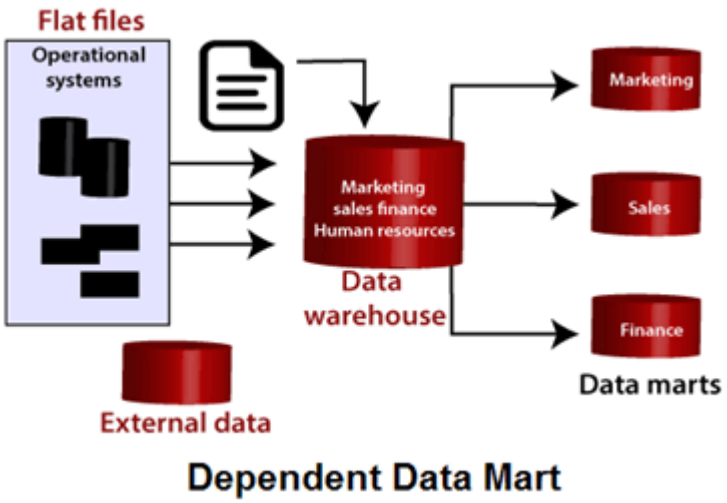
## Types of Data Marts

There are mainly two approaches to designing data marts. These approaches are

- Dependent Data Marts
- Independent Data Marts

## Dependent Data Marts

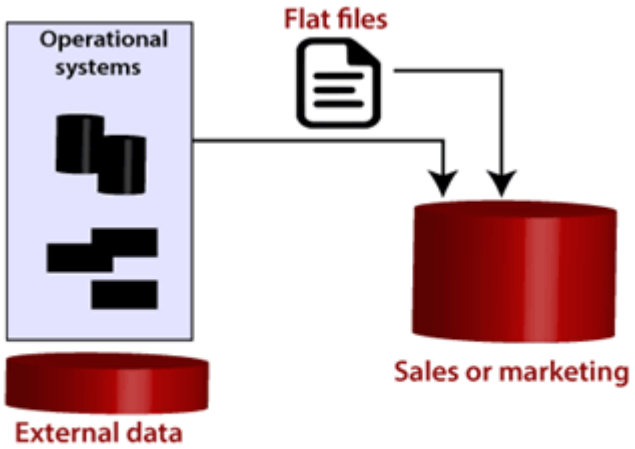
A dependent data marts is a logical subset of a physical subset of a higher data warehouse. According to this technique, the data marts are treated as the subsets of a data warehouse. In this technique, firstly a data warehouse is created from which further various data marts can be created. These data mart are dependent on the data warehouse and extract the essential record from it. In this technique, as the data warehouse creates the data mart; therefore, there is no need for data mart integration. It is also known as a **top-down approach**.




Dependent Data Mart

## Independent Data Marts

The second approach is Independent data marts (IDM) Here, firstly independent data marts are created, and then a data warehouse is designed using these independent multiple data marts. In this approach, as all the data marts are designed independently; therefore, the integration of data marts is required. It is also termed as a **bottom-up approach** as the data marts are integrated to develop a data warehouse.




Independent Data Mart

Other than these two categories, one more type exists that is called "**Hybrid Data Marts**."

## Hybrid Data Marts

It allows us to combine input from sources other than a data warehouse. This could be helpful for many situations; especially when Adhoc integrations are needed, such as after a new group or product is added to the organizations.

**Steps in Implementing a Data Mart**

The significant steps in implementing a data mart are to design the schema, construct the physical storage, populate the data mart with data from source systems, access it to make informed decisions and manage it over time. So, the steps are:

**Designing**

The design step is the first in the data mart process. This phase covers all of the functions from initiating the request for a data mart through gathering data about the requirements and developing the logical and physical design of the data mart.

It involves the following tasks:

1. Gathering the business and technical requirements
2. Identifying data sources
3. Selecting the appropriate subset of data
4. Designing the logical and physical architecture of the data mart.

**Constructing**

This step contains creating the physical database and logical structures associated with the data mart to provide fast and efficient access to the data.

It involves the following tasks:

1. Creating the physical database and logical structures such as tablespaces associated with the data mart.
2. creating the schema objects such as tables and indexes describe in the design step.
3. Determining how best to set up the tables and access structures.

**Populating**

This step includes all of the tasks related to the getting data from the source, cleaning it up, modifying it to the right format and level of detail, and moving it into the data mart.

It involves the following tasks:

1. Mapping data sources to target data sources
2. Extracting data
3. Cleansing and transforming the information.
4. Loading data into the data mart
5. Creating and storing metadata

**Accessing**

This step involves putting the data to use: querying the data, analyzing it, creating reports, charts and graphs and publishing them.

It involves the following tasks:

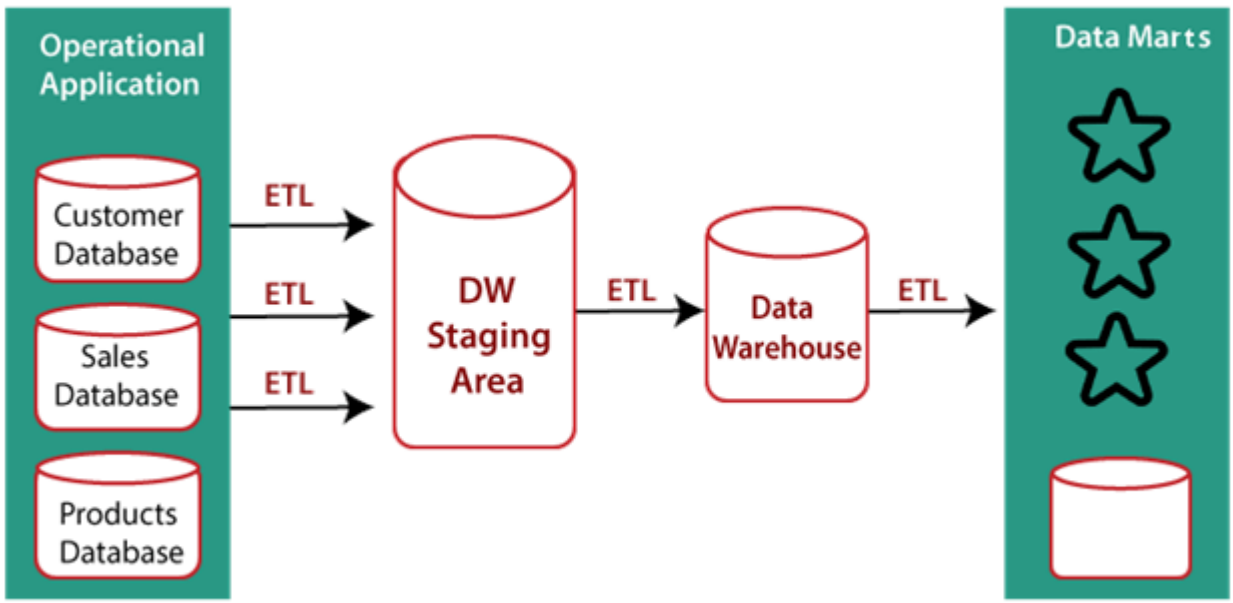
1. Set up and intermediate layer (Meta Layer) for the front-end tool to use. This layer translates database operations and objects names into business conditions so that the end-clients can interact with the data mart using words which relates to the business functions.
2. Set up and manage database architectures like summarized tables which help queries agree through the front-end tools execute rapidly and efficiently.

**Managing**

This step contains managing the data mart over its lifetime. In this step, management functions are performed as:

1. Providing secure access to the data.
2. Managing the growth of the data.
3. Optimizing the system for better performance.
4. Ensuring the availability of data event with system failures.

**Difference between Data Warehouse and Data Mart**




| Data Warehouse | Data Mart |
| --- | --- |

| | |
|---|---|
| A Data Warehouse is a vast repository of information collected from various organizations or departments within a corporation. | A data mart is an only subtype of a Data Warehouses. It is architecture to meet the requirement of a specific user group. |
| It may hold multiple subject areas. | It holds only one subject area. For example, Finance or Sales. |
| It holds very detailed information. | It may hold more summarized data. |
| Works to integrate all data sources | It concentrates on integrating data from a given subject area or set of source systems. |
| In data warehousing, Fact constellation is used. | In Data Mart, Star Schema and Snowflake Schema are used. |
| It is a Centralized System. It is a Decentralized System. | |
| Data Warehousing is the data-oriented. | Data Marts is a project-oriented. |

## What is Operational Data Stores?

An **ODS** has been described by **Inmon** and **Imhoff** (1996) as a subject-oriented, integrated, volatile, current valued data store, containing only detailed corporate data. A data warehouse is a documenting database that includes associatively recent as well as historical information and may also include aggregate data.

The **ODS** is a **subject-oriented**. It is organized around the significant information subject of an enterprise. In a university, the subjects may be students, lecturers and courses while in the company the subjects might be users, salespersons and products.

The **ODS** is an **integrated**. That is, it is a group of subject-oriented record from a variety of systems to provides an enterprise-wide view of the information.

The **ODS** is a **current-valued**. That is, an **ODS** is up-to-date and follow the current status of the data. An ODS does not contain historical information. Since the OLTP system data is changing all the time, data from underlying sources refresh the ODS as generally and frequently as possible.

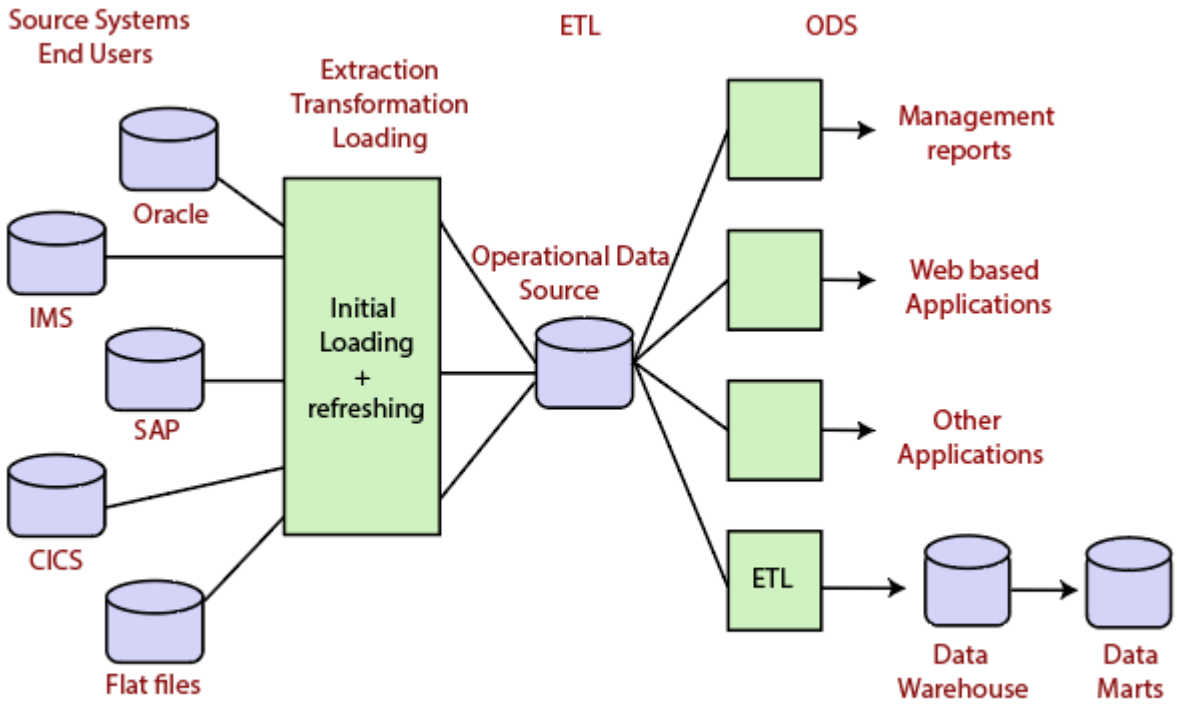
The **ODS** is **volatile**. That is, the data in the ODS frequently changes as new data refreshes the ODS.

The ODS is a **detailed**. That is, ODS is detailed enough to serve the need of the operational management staff in the enterprise. The granularity of the information in the ODS does not have to be precisely the same as in the source **OLTP** system.

### ODS Design and Implementation

The extraction of data from source databases needs to be efficient, and the quality of records needs to be maintained. Since the data is refreshed generally and frequently, suitable checks are required to ensure the quality of data after each refresh. An **ODS** is a read-only database other than regular refreshing by the OLTP systems. Customer should not be allowed to update ODS information.

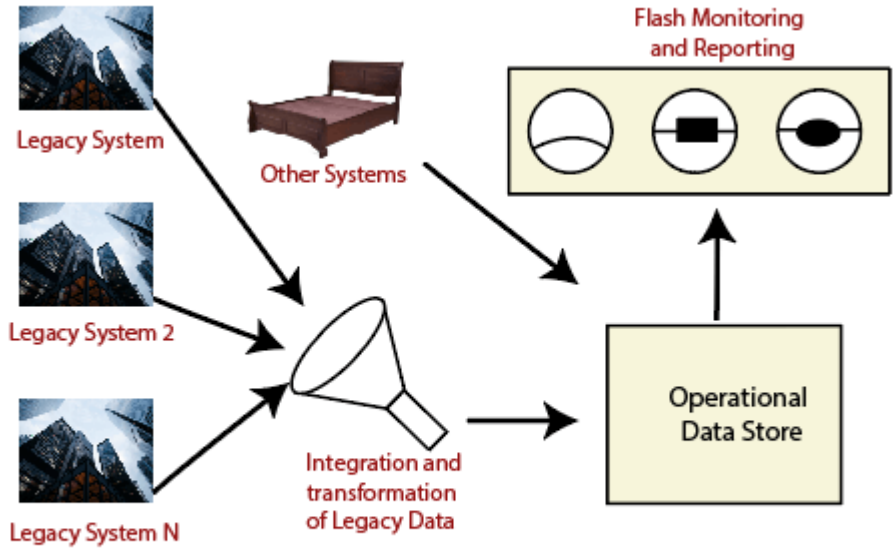
Populating an ODS contains an acquisition phase of extracting, transforming and loading information from OLTP source systems. This procedure is **ETL**. Completing populating the database, analyze for anomalies and testing for performance are essential before an ODS system can go online.




Operational Data Store Structure

**Flash Monitoring and Reporting Tools**

Flash monitoring and the reporting tools are like a dashboard that support meaningful online data on the operational status of the enterprise. This method is achieved by the use of ODS data as inputs to the flash monitoring and reporting tools, to provide business users with a refreshed continuously, enterprise-wide view of operations without creating unwanted interruptions or additional load on transactions-processing systems.




Operational Monitoring

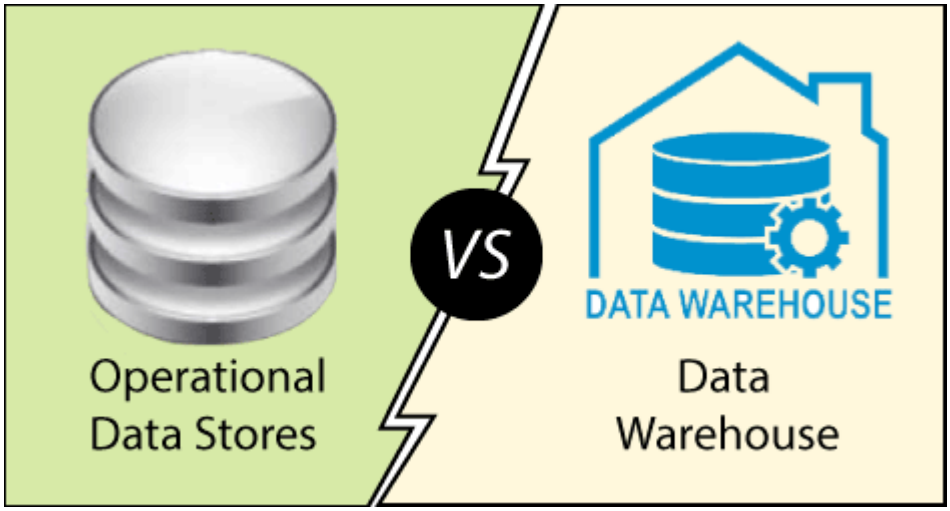
**Zero Latency Enterprise (ZLE)**

The Gantner Group has used a method Zero Latency Enterprise (ZLE) for near real-time integration of operational information so that there is no necessary delay in getting data from one part or one system of an enterprise to another system that needs the data.

A ZLE data store is like an ODS that is integrated and up-to-date. The objective of a ZLE data store is to allow management a single view of enterprise information by bringing together relevant information in real-time and providing management with a "360-degree" aspect of the user.

A ZLE generally has the following features. It has a consolidated view of the enterprise operational information. It has a massive level of availability, and it contains online refreshing of data. ZLE requires data that is as current as possible. Since a ZLE needs to provide a large number of concurrent users, for example, call centre users, the fast turnaround time for transactions and 24/7 availability are required.

**Difference between Operational Data Stores and Data Warehouse**




| Operational Data Stores | Data Warehouse |
|---|---|
| ODS means for **operational** reporting and supports current or near real-time reporting requirements. | A **data warehouse** is intended for historical and trend analysis, usually reporting on a large volume of data. |
| An **ODS** consist of only a short window of **data**. | A **data warehouse** includes the entire history of **data**. |
| It is typically detailed data only. | It contains summarized and detailed data. |
| It is used for detailed decision making and operational reporting. | It is used for long term decision making and management reporting. |
| It is used at the operational level. | It is used at the managerial level. |
| It serves as conduct for data between operational and analytics system. | It serves as a repository for cleansed and consolidated data sets. |
| It is updated often as the transactions system generates new data. | It is usually updated in batch processing mode on a set schedule. |