The term "Data Warehouse" was first coined by Bill Inmon in 1990. According to Inmon, a data warehouse is a subject oriented, integrated, time-variant, and non-volatile collection of data. This data helps analysts to take informed decisions in an organization.

An operational database undergoes frequent changes on a daily basis on account of the transactions that take place. Suppose a business executive wants to analyze previous feedback on any data such as a product, a supplier, or any consumer data, then the executive will have no data available to analyze because the previous data has been updated due to transactions.

A data warehouses provides us generalized and consolidated data in multidimensional view. Along with generalized and consolidated view of data, a data warehouses also provides us Online Analytical Processing (OLAP) tools. These tools help us in interactive and effective analysis of data in a multidimensional space. This analysis results in data generalization and data mining.

Data mining functions such as association, clustering, classification, prediction can be integrated with OLAP operations to enhance the interactive mining of knowledge at multiple level of abstraction. That's why data warehouse has now become an important platform for data analysis and online analytical processing.

Understanding a Data Warehouse

- A data warehouse is a database, which is kept separate from the organization's operational database.
- There is no frequent updating done in a data warehouse.
- It possesses consolidated historical data, which helps the organization to analyze its business.
- A data warehouse helps executives to organize, understand, and use their data to take strategic decisions.
- Data warehouse systems help in the integration of diversity of application systems.
- A data warehouse system helps in consolidated historical data analysis.

Why a Data Warehouse is Separated from Operational Databases

A data warehouses is kept separate from operational databases due to the following reasons −

- An operational database is constructed for well-known tasks and workloads such as searching particular records, indexing, etc. In contract, data warehouse queries are often complex and they present a general form of data.

- Operational databases support concurrent processing of multiple transactions. Concurrency control and recovery mechanisms are required for operational databases to ensure robustness and consistency of the database.
- An operational database query allows to read and modify operations, while an OLAP query needs only **read only** access of stored data.
- An operational database maintains current data. On the other hand, a data warehouse maintains historical data.

Data Warehouse Features

The key features of a data warehouse are discussed below −

- **Subject Oriented** − A data warehouse is subject oriented because it provides information around a subject rather than the organization's ongoing operations. These subjects can be product, customers, suppliers, sales, revenue, etc. A data warehouse does not focus on the ongoing operations, rather it focuses on modelling and analysis of data for decision making.
- **Integrated** − A data warehouse is constructed by integrating data from heterogeneous sources such as relational databases, flat files, etc. This integration enhances the effective analysis of data.
- **Time Variant** − The data collected in a data warehouse is identified with a particular time period. The data in a data warehouse provides information from the historical point of view.
- **Non-volatile** − Non-volatile means the previous data is not erased when new data is added to it. A data warehouse is kept separate from the operational database and therefore frequent changes in operational database is not reflected in the data warehouse.

Note − A data warehouse does not require transaction processing, recovery, and concurrency controls, because it is physically stored and separate from the operational database.

Data Warehouse Applications

As discussed before, a data warehouse helps business executives to organize, analyze, and use their data for decision making. A data warehouse serves as a sole part of a plan-execute-assess "closed-loop" feedback system for the enterprise management. Data warehouses are widely used in the following fields −

- Financial services
- Banking services

- Consumer goods
- Retail sectors
- Controlled manufacturing

Types of Data Warehouse

Information processing, analytical processing, and data mining are the three types of data warehouse applications that are discussed below −

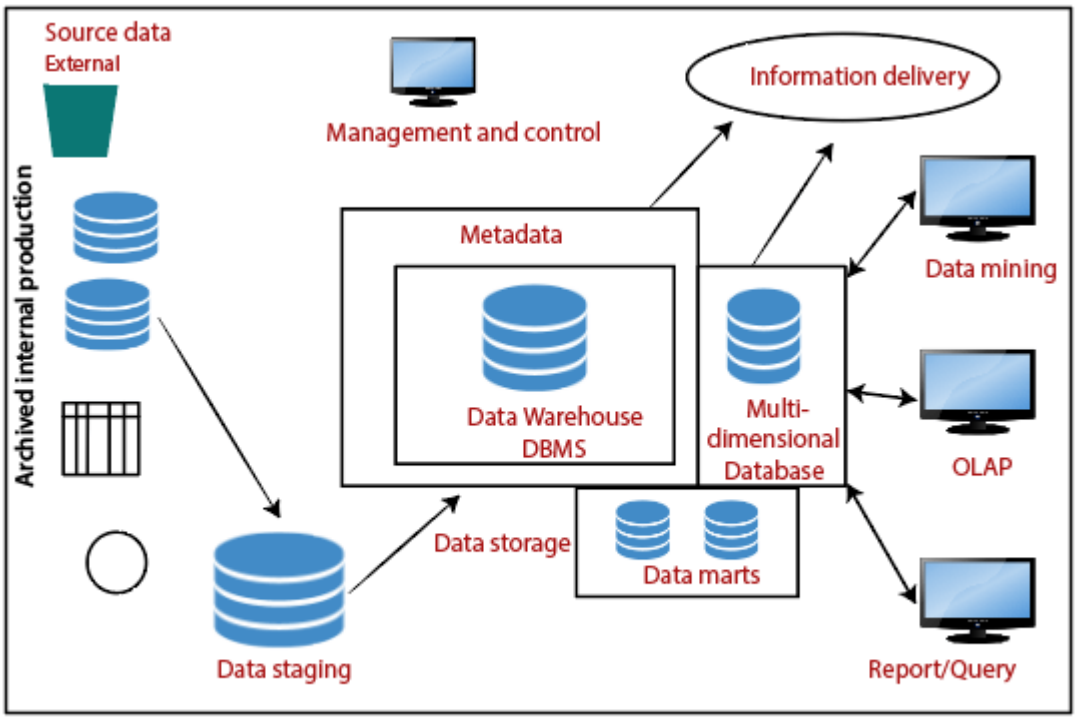
- **Information Processing** − A data warehouse allows to process the data stored in it. The data can be processed by means of querying, basic statistical analysis, reporting using crosstabs, tables, charts, or graphs.
- **Analytical Processing** − A data warehouse supports analytical processing of the information stored in it. The data can be analyzed by means of basic OLAP operations, including slice-and-dice, drill down, drill up, and pivoting.
- **Data Mining** − Data mining supports knowledge discovery by finding hidden patterns and associations, constructing analytical models, performing classification and prediction. These mining results can be presented using the visualization tools.

Sr.No.	Data Warehouse (OLAP)	Operational Database(OLTP)
1	It involves historical processing of information.	It involves day-to-day processing.
2	OLAP systems are used by knowledge workers such as executives, managers, and analysts.	OLTP systems are used by clerks, DBAs, or database professionals.
3	It is used to analyze the business.	It is used to run the business.
4	It focuses on Information out.	It focuses on Data in.
5	It is based on Star Schema, Snowflake Schema, and Fact Constellation Schema.	It is based on Entity Relationship Model.
6	It focuses on Information out.	It is application oriented.
7	It contains historical data.	It contains current data.

8	It provides summarized and consolidated data.	It provides primitive and highly detailed data.
9	It provides summarized and multidimensional view of data.	It provides detailed and flat relational view of data.
10	The number of users is in hundreds.	The number of users is in thousands.
11	The number of records accessed is in millions.	The number of records accessed is in tens.
12	The database size is from 100GB to 100 TB.	The database size is from 100 MB to 100 GB.
13	These are highly flexible.	It provides high performance.

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.

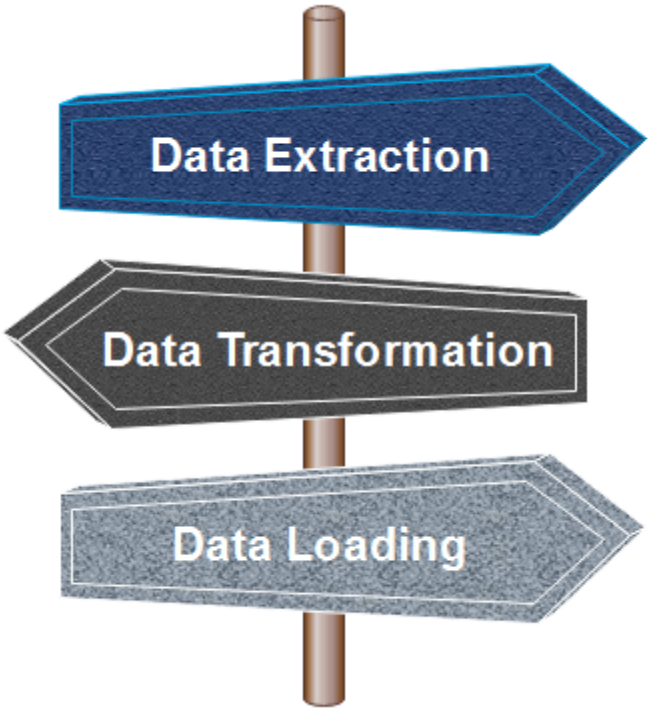
Archived Data: Operational systems are mainly intended to run the current business. In every operational system, we periodically take the old data and store it in achieved files.

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.

We will now discuss the three primary functions that take place in the staging area.



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.

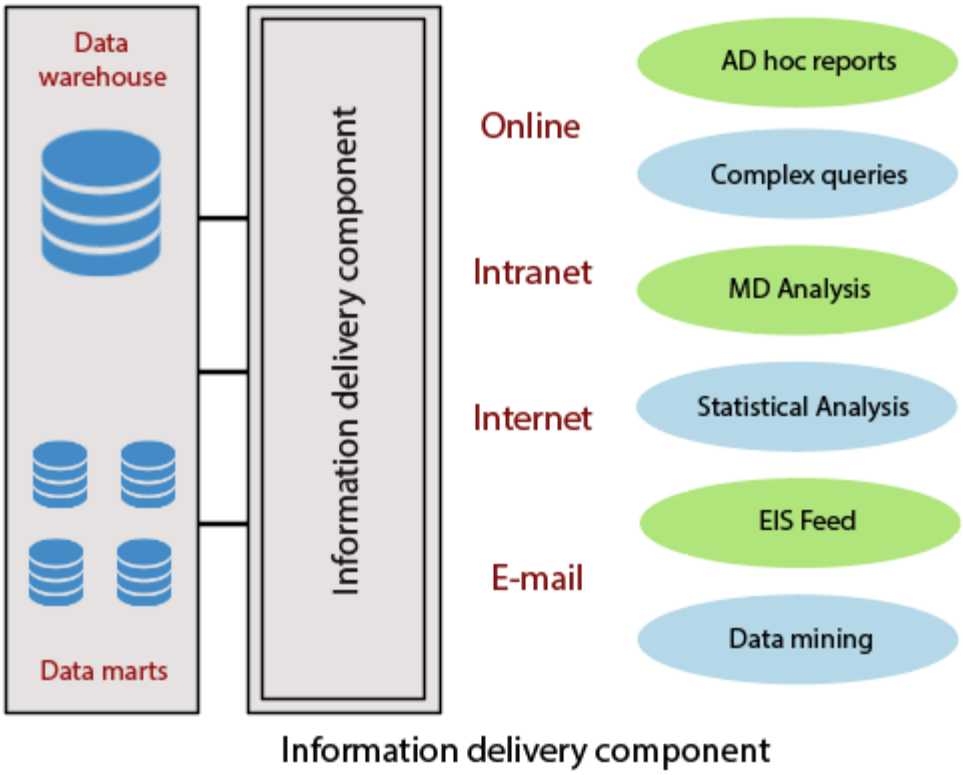
3) Data Loading: Two distinct categories of tasks form data loading functions. When we complete the structure and construction of the data warehouse and go live for the first time, we do the initial loading of the information into the data warehouse storage. The initial load moves high volumes of data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

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.

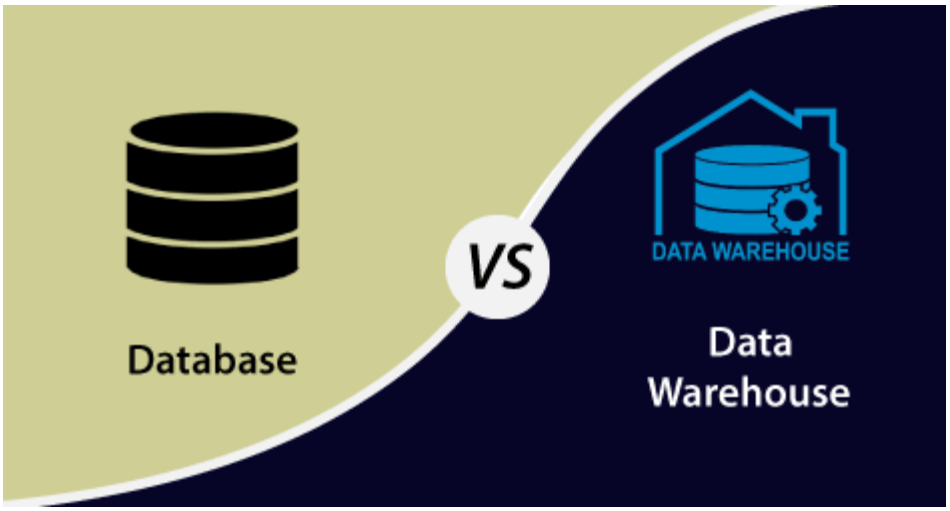
Performing OLAP queries in operational database degrade the performance of functional tasks.

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	1. It is used for Online Analytical Processing (OLAP). This reads the

as Data Warehousing. This records the data from the clients for history.	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.

The Operational Database is the source of information for the data warehouse. It includes detailed information used to run the day to day operations of the business. The data frequently changes as updates are made and reflect the current value of the last transactions.

Operational Database Management Systems also called as OLTP (Online Transactions Processing Databases), are used to manage dynamic data in real-time.

Data Warehouse Systems serve users or knowledge workers in the purpose of data analysis and decision-making. Such systems can organize and present information in specific formats to accommodate the diverse needs of various users. These systems are called as Online-Analytical Processing (OLAP) Systems.

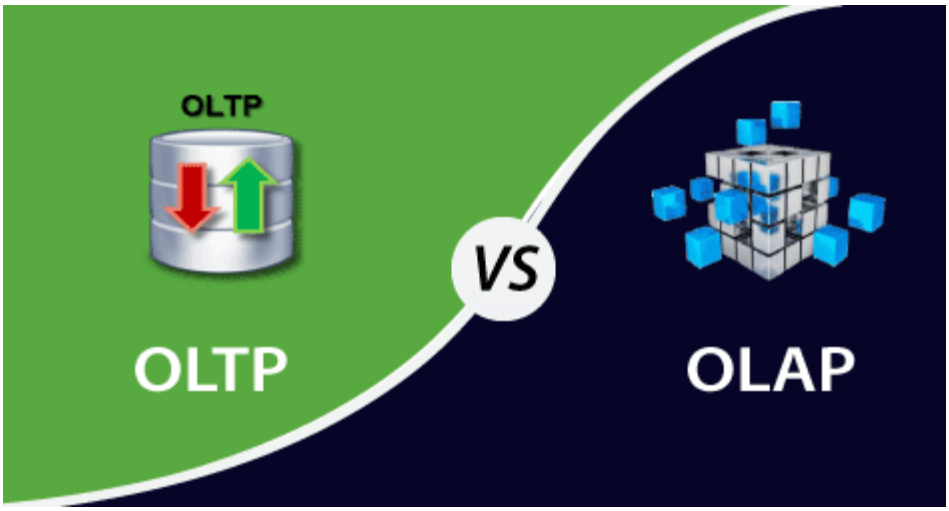
Data Warehouse and the OLTP database are both relational databases. However, the goals of both these databases are different.

Operational Database	**Data Warehouse**

Operational systems are designed to support high-volume transaction processing.	Data warehousing systems are typically designed to support high-volume analytical processing (i.e., OLAP).
Operational systems are usually concerned with current data.	Data warehousing systems are usually concerned with historical data.
Data within operational systems are mainly updated regularly according to need.	Non-volatile, new data may be added regularly. Once Added rarely changed.
It is designed for real-time business dealing and processes.	It is designed for analysis of business measures by subject area, categories, and attributes.
It is optimized for a simple set of transactions, generally adding or retrieving a single row at a time per table.	It is optimized for extent loads and high, complex, unpredictable queries that access many rows per table.
It is optimized for validation of incoming information during transactions, uses validation data tables.	Loaded with consistent, valid information, requires no real-time validation.
It supports thousands of concurrent clients.	It supports a few concurrent clients relative to OLTP.
Operational systems are widely process-oriented.	Data warehousing systems are widely subject-oriented
Operational systems are usually optimized to perform fast inserts and updates of associatively small volumes of data.	Data warehousing systems are usually optimized to perform fast retrievals of relatively high volumes of data.
Data In	Data Out
Less Number of data accessed.	Large Number of data accessed.

Relational databases are created for on-line transactional Processing (OLTP)	Data Warehouse designed for on-line Analytical Processing (OLAP)

Difference between OLTP and OLAP



OLTP System

OLTP System handle with operational data. Operational data are those data contained in the operation of a particular system. Example, ATM transactions and Bank transactions, etc.

OLAP System

OLAP handle with Historical Data or Archival Data. Historical data are those data that are achieved over a long period. For example, if we collect the last 10 years information about flight reservation, the data can give us much meaningful data such as the trends in the reservation. This may provide useful information like peak time of travel, what kind of people are traveling in various classes (Economy/Business) etc.

The major difference between an OLTP and OLAP system is the amount of data analyzed in a single transaction. Whereas an OLTP manage many concurrent customers and queries touching only an individual record or limited groups of files at a time. An OLAP system must have the capability to operate on millions of files to answer a single query.

Feature	OLTP	OLAP
Characteristic	It is a system which is used to manage operational Data.	It is a system which is used to manage informational Data.

Users	Clerks, clients, and information technology professionals.	Knowledge workers, including managers, executives, and analysts.
System orientation	OLTP system is a customer-oriented, transaction, and query processing are done by clerks, clients, and information technology professionals.	OLAP system is market-oriented, knowledge workers including managers, do data analysts executive and analysts.
Data contents	OLTP system manages current data that too detailed and are used for decision making.	OLAP system manages a large amount of historical data, provides facilitates for summarization and aggregation, and stores and manages data at different levels of granularity. This information makes the data more comfortable to use in informed decision making.
Database Size	100 MB-GB	100 GB-TB
Database design	OLTP system usually uses an entity-relationship (ER) data model and application-oriented database design.	OLAP system typically uses either a star or snowflake model and subject-oriented database design.
View	OLTP system focuses primarily on the current data within an enterprise or department, without referring to historical information or data in different organizations.	OLAP system often spans multiple versions of a database schema, due to the evolutionary process of an organization. OLAP systems also deal with data that originates from various organizations, integrating information from many data stores.
Volume of data	Not very large	Because of their large volume, OLAP data are stored on multiple storage media.

Access patterns	The access patterns of an OLTP system subsist mainly of short, atomic transactions. Such a system requires concurrency control and recovery techniques.	Accesses to OLAP systems are mostly read-only methods because of these data warehouses stores historical data.
Access mode	Read/write	Mostly write
Insert and Updates	Short and fast inserts and updates proposed by end-users.	Periodic long-running batch jobs refresh the data.
Number of records accessed	Tens	Millions
Normalization	Fully Normalized	Partially Normalized
Processing Speed	Very Fast	It depends on the amount of files contained, batch data refresh, and complex query may take many hours, and query speed can be upgraded by creating indexes.

Data Warehouse Architecture

A data warehouse architecture is a method of defining the overall architecture of data communication processing and presentation that exist for end-clients computing within the enterprise. Each data warehouse is different, but all are characterized by standard vital components.

Production applications such as payroll accounts payable product purchasing and inventory control are designed for online transaction processing **(OLTP)**. Such applications gather detailed data from day to day operations.

Data Warehouse applications are designed to support the user ad-hoc data requirements, an activity recently dubbed online analytical processing (OLAP). These include applications such as forecasting, profiling, summary reporting, and trend analysis.

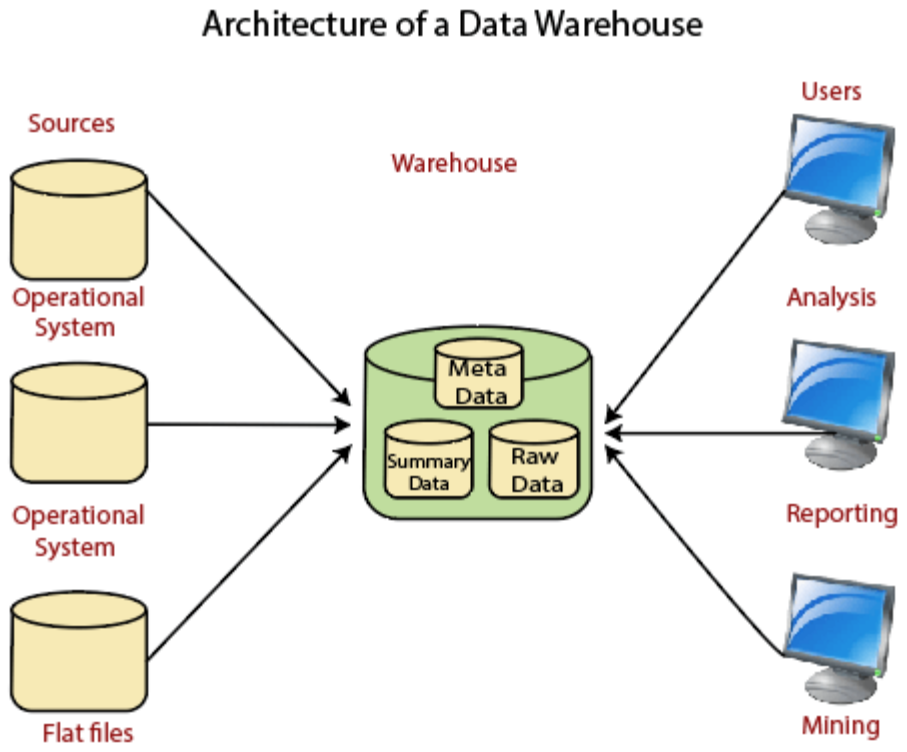
Production databases are updated continuously by either by hand or via OLTP applications. In contrast, a warehouse database is updated from operational systems periodically, usually during off-hours. As OLTP data accumulates in production databases, it is regularly extracted, filtered, and then loaded into a dedicated warehouse server that is accessible to users. As the warehouse is populated, it must be restructured tables de-normalized, data cleansed of errors and redundancies and new fields and keys added to reflect the needs to the user for sorting, combining, and summarizing data.

Data warehouses and their architectures very depending upon the elements of an organization's situation.

Three common architectures are:

- Data Warehouse Architecture: Basic
- Data Warehouse Architecture: With Staging Area
- Data Warehouse Architecture: With Staging Area and Data Marts

Data Warehouse Architecture: Basic



Operational System

An **operational system** is a method used in data warehousing to refer to a **system** that is used to process the day-to-day transactions of an organization.

Flat Files

A **Flat file** system is a system of files in which transactional data is stored, and every file in the system must have a different name.

Meta Data

A set of data that defines and gives information about other data.

Meta Data used in Data Warehouse for a variety of purpose, including:

Meta Data summarizes necessary information about data, which can make finding and work with particular instances of data more accessible. For example, author, data build, and data changed, and file size are examples of very basic document metadata.

Metadata is used to direct a query to the most appropriate data source.

Lightly and highly summarized data

The area of the data warehouse saves all the predefined lightly and highly summarized (aggregated) data generated by the warehouse manager.

The goals of the summarized information are to speed up query performance. The summarized record is updated continuously as new information is loaded into the warehouse.

End-User access Tools

The principal purpose of a data warehouse is to provide information to the business managers for strategic decision-making. These customers interact with the warehouse using end-client access tools.

The examples of some of the end-user access tools can be:

- Reporting and Query Tools
- Application Development Tools
- Executive Information Systems Tools
- Online Analytical Processing Tools
- Data Mining Tools

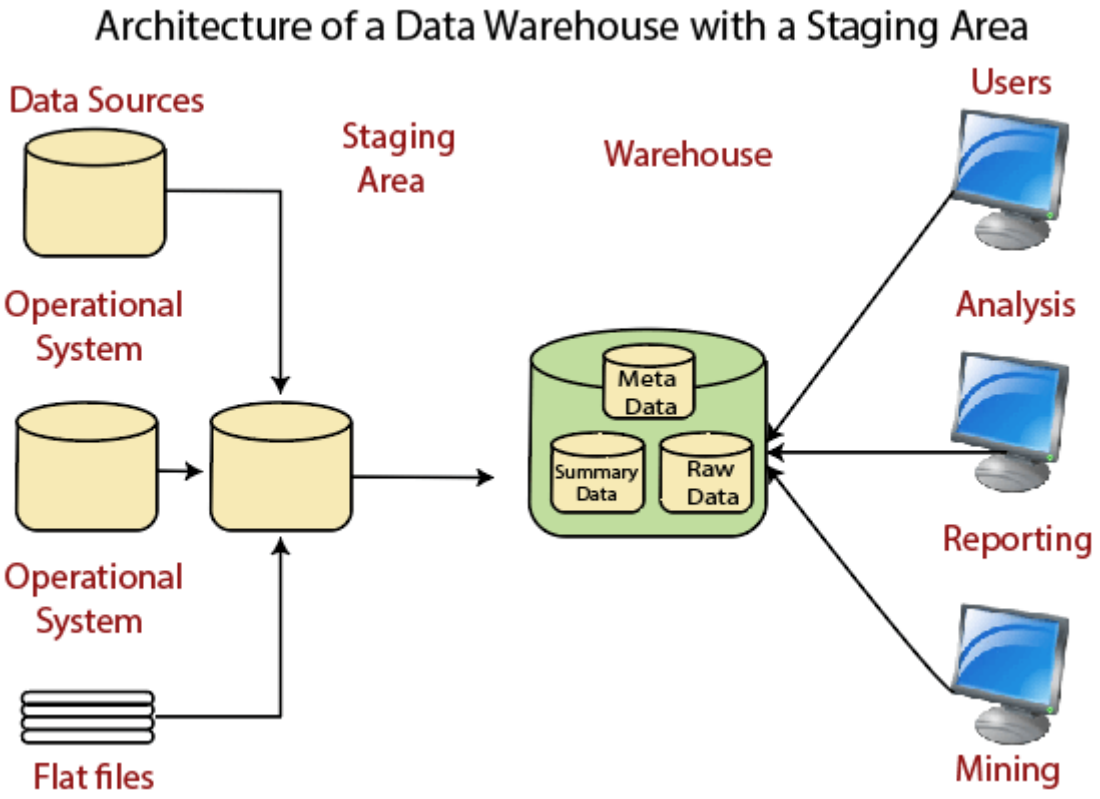
Data Warehouse Architecture: With Staging Area

We must clean and process your operational information before put it into the warehouse.

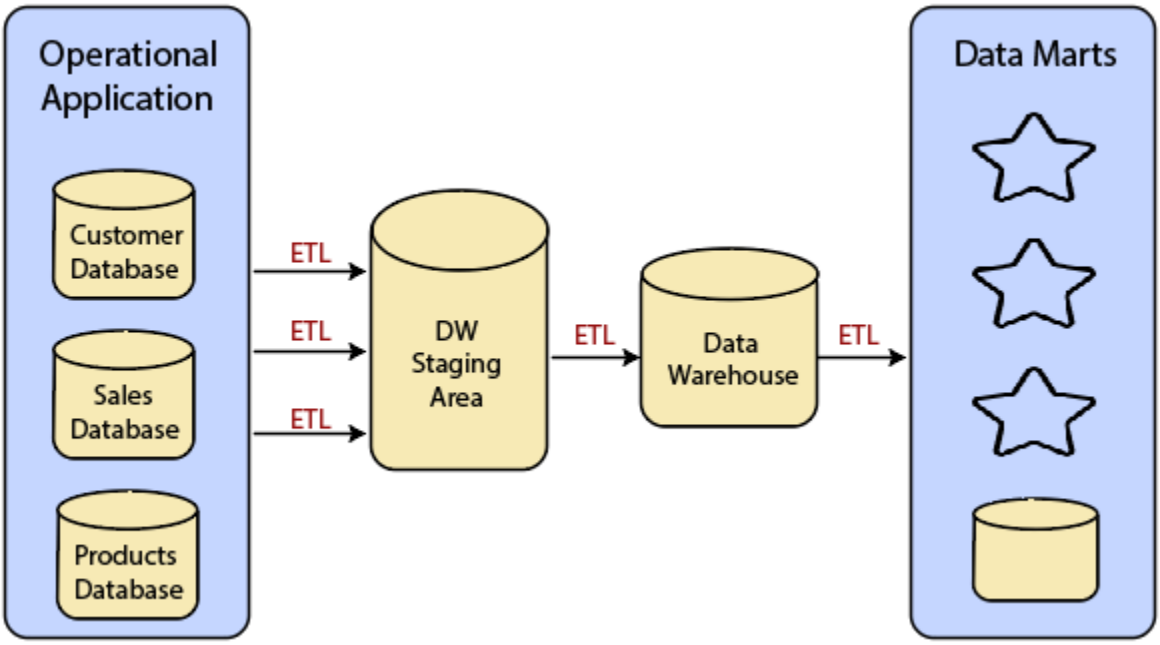
W

e can do this programmatically, although data warehouses uses a **staging area** (A place where data is processed before entering the warehouse).

A staging area simplifies data cleansing and consolidation for operational method coming from multiple source systems, especially for enterprise data warehouses where all relevant data of an enterprise is consolidated.



Data Warehouse Staging Area is a temporary location where a record from source systems is copied.

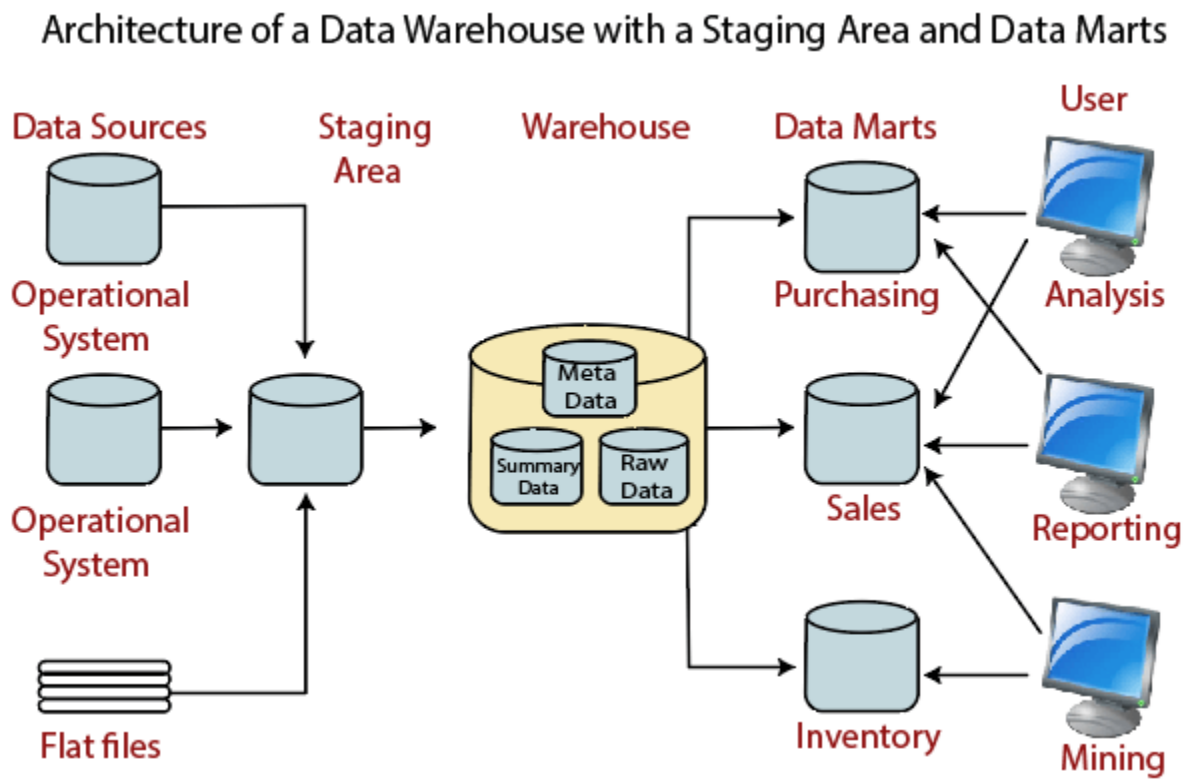


Data Warehouse Architecture: With Staging Area and Data Marts

We may want to customize our warehouse's architecture for multiple groups within our organization.

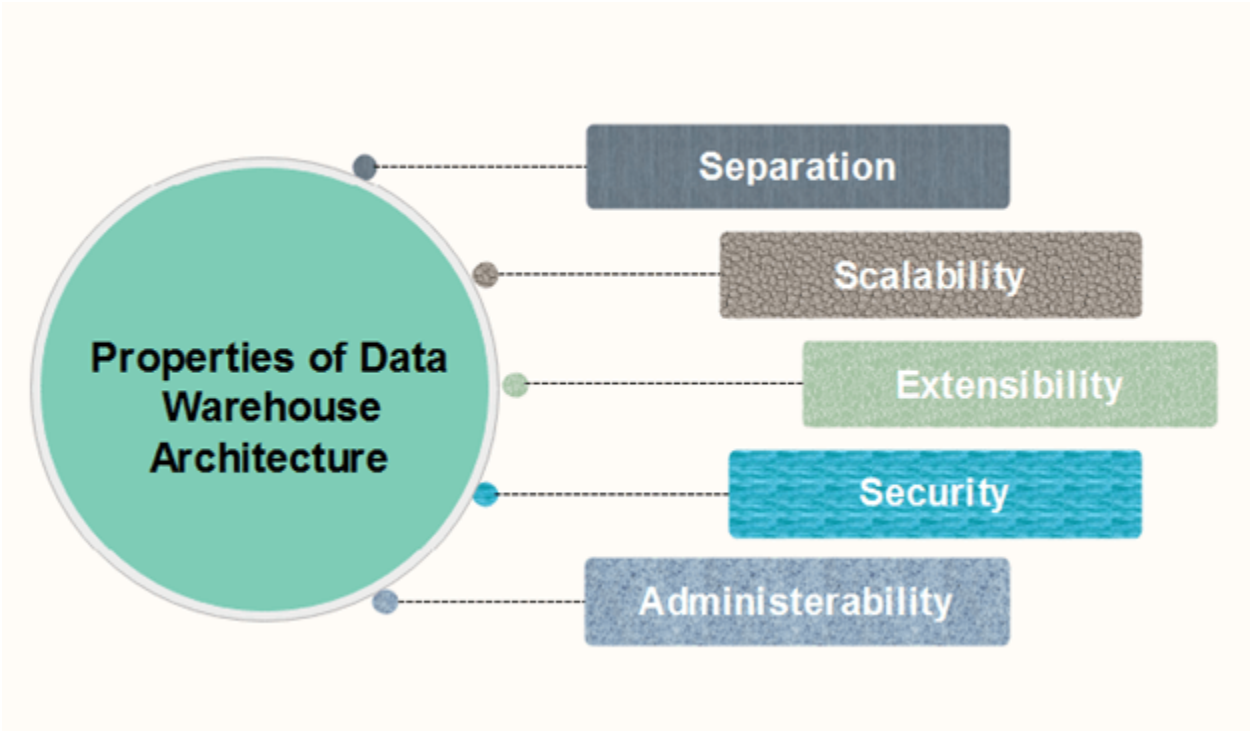
We can do this by adding **data marts**. A data mart is a segment of a data warehouses that can provided information for reporting and analysis on a section, unit, department or operation in the company, e.g., sales, payroll, production, etc.

The figure illustrates an example where purchasing, sales, and stocks are separated. In this example, a financial analyst wants to analyze historical data for purchases and sales or mine historical information to make predictions about customer behavior.



Properties of Data Warehouse Architectures

The following architecture properties are necessary for a data warehouse system:



1. Separation: Analytical and transactional processing should be keep apart as much as possible.

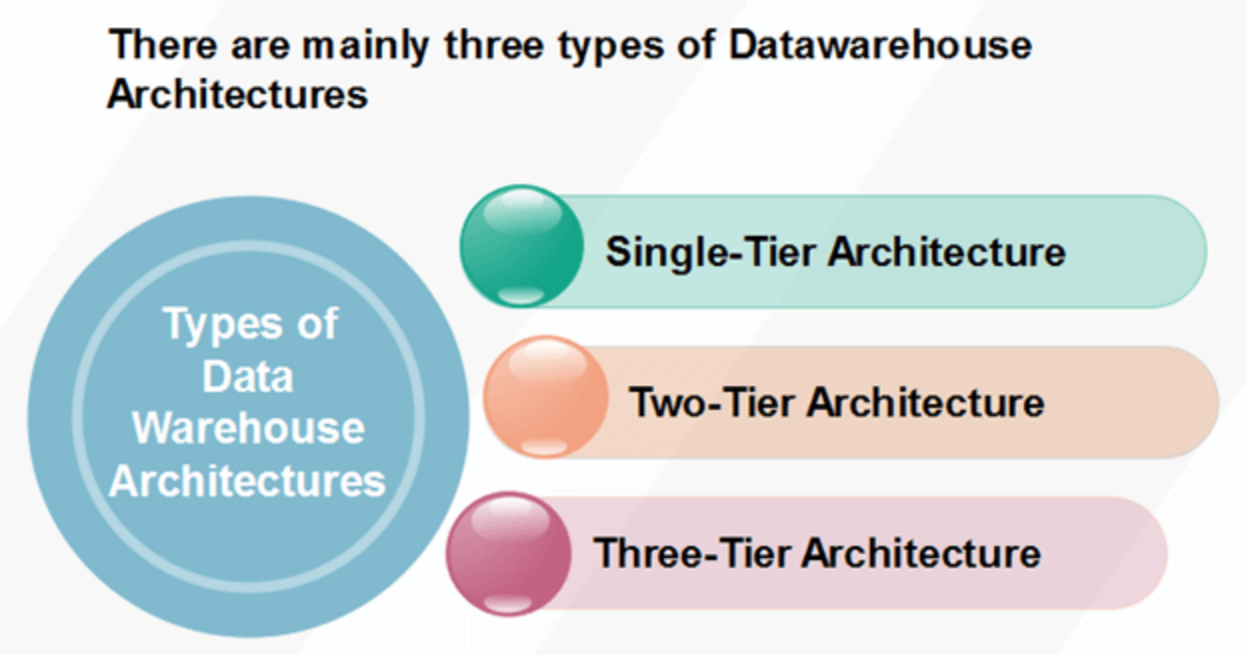
2. Scalability: Hardware and software architectures should be simple to upgrade the data volume, which has to be managed and processed, and the number of user's requirements, which have to be met, progressively increase.

3. Extensibility: The architecture should be able to perform new operations and technologies without redesigning the whole system.

4. Security: Monitoring accesses are necessary because of the strategic data stored in the data warehouses.

5. Administerability: Data Warehouse management should not be complicated.

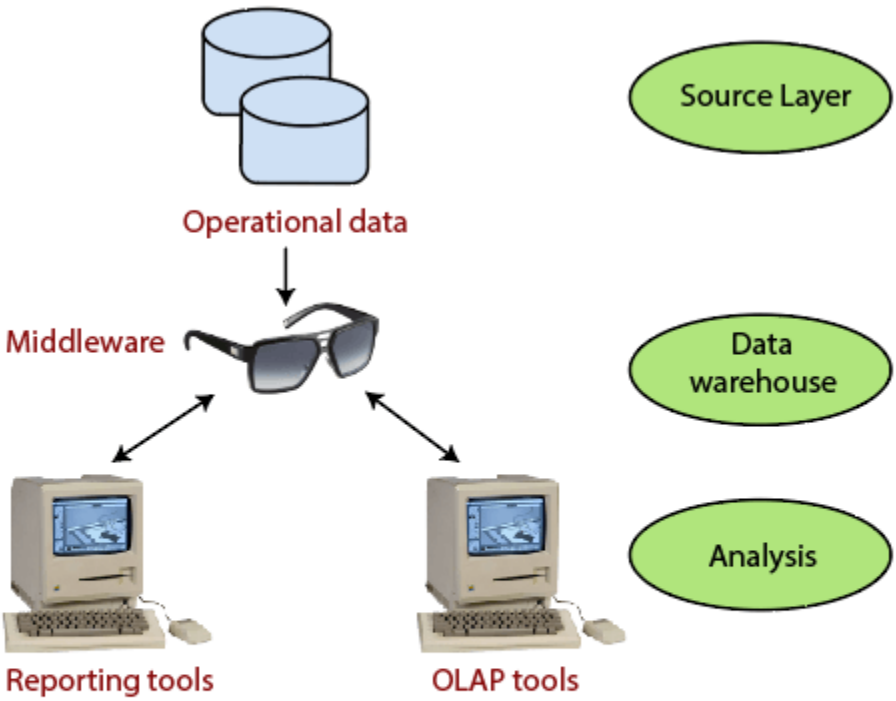
Types of Data Warehouse Architectures



Single-Tier Architecture

Single-Tier architecture is not periodically used in practice. Its purpose is to minimize the amount of data stored to reach this goal; it removes data redundancies.

The figure shows the only layer physically available is the source layer. In this method, data warehouses are virtual. This means that the data warehouse is implemented as a multidimensional view of operational data created by specific middleware, or an intermediate processing layer.

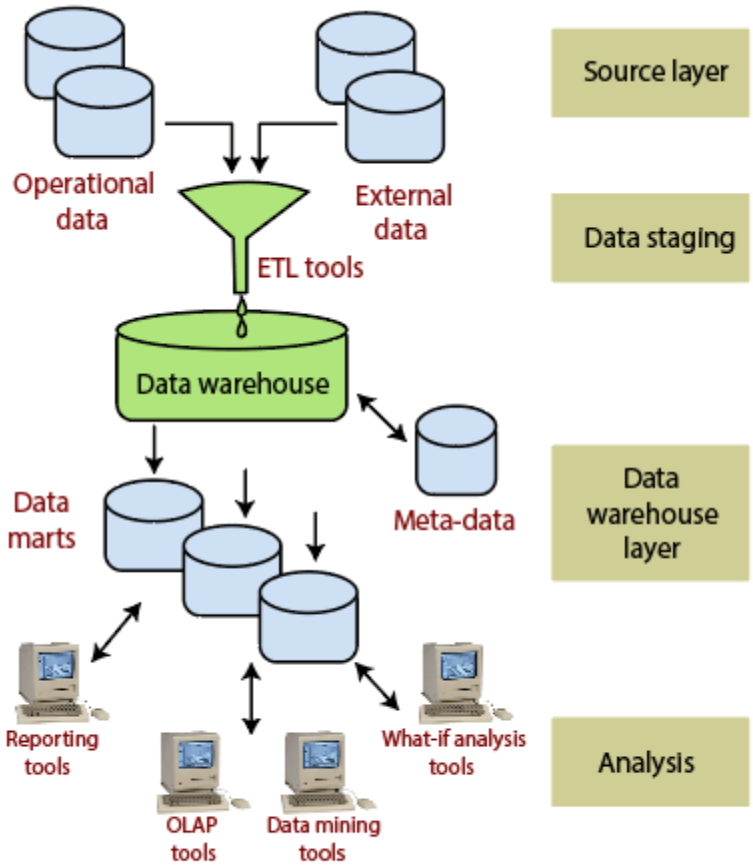


Single-Tier Data Warehouse Architecture

The vulnerability of this architecture lies in its failure to meet the requirement for separation between analytical and transactional processing. Analysis queries are agreed to operational data after the middleware interprets them. In this way, queries affect transactional workloads.

Two-Tier Architecture

The requirement for separation plays an essential role in defining the two-tier architecture for a data warehouse system, as shown in fig:



Two-Tier Data Warehouse Architecture

Although it is typically called two-layer architecture to highlight a separation between physically available sources and data warehouses, in fact, consists of four subsequent data flow stages:

1. **Source layer:** A data warehouse system uses a heterogeneous source of data. That data is stored initially to corporate relational databases or legacy databases, or it may come from an information system outside the corporate walls.
2. **Data Staging:** The data stored to the source should be extracted, cleansed to remove inconsistencies and fill gaps, and integrated to merge heterogeneous sources into one standard schema. The so-named **Extraction**, **Transformation**, and **Loading Tools (ETL)** can combine heterogeneous schemata, extract, transform, cleanse, validate, filter, and load source data into a data warehouse.
3. **Data Warehouse layer:** Information is saved to one logically centralized individual repository: a data warehouse. The data warehouses can be directly accessed, but it can also be used as a source for creating data marts, which partially replicate data warehouse contents and are designed for specific enterprise departments. Meta-data

repositories store information on sources, access procedures, data staging, users, data mart schema, and so on.

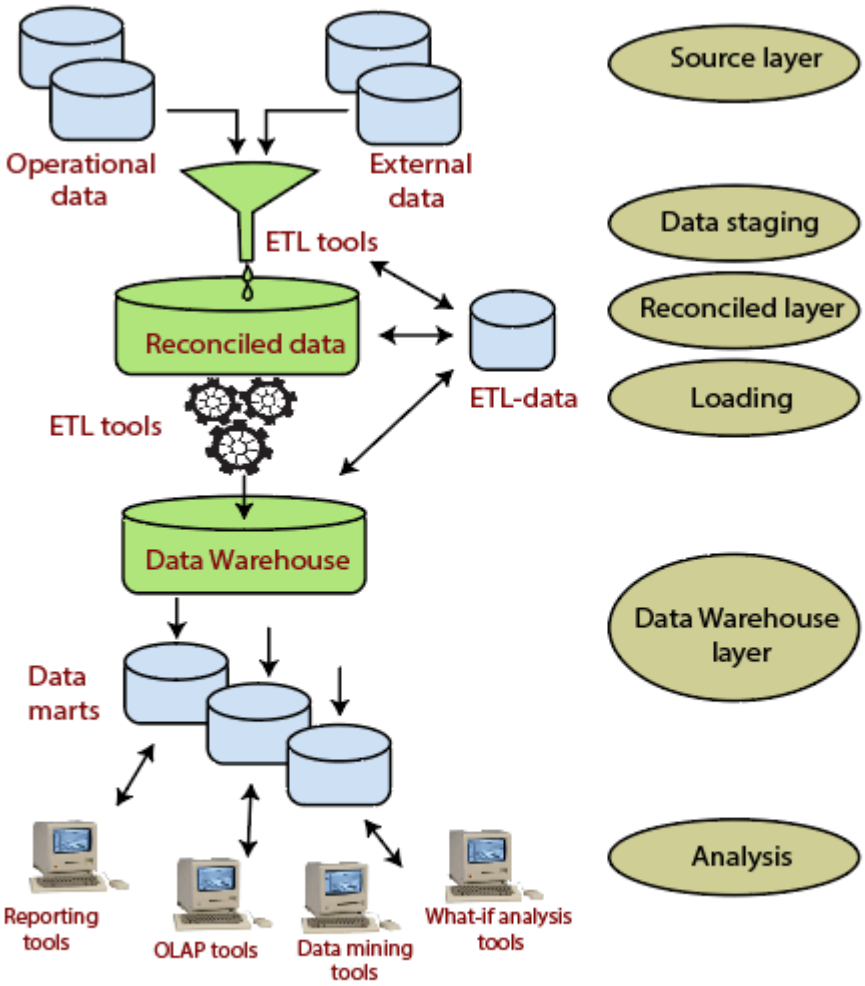
4. **Analysis:** In this layer, integrated data is efficiently, and flexible accessed to issue reports, dynamically analyze information, and simulate hypothetical business scenarios. It should feature aggregate information navigators, complex query optimizers, and customer-friendly GUIs.

Three-Tier Architecture

The three-tier architecture consists of the source layer (containing multiple source system), the reconciled layer and the data warehouse layer (containing both data warehouses and data marts). The reconciled layer sits between the source data and data warehouse.

The main advantage of the **reconciled layer** is that it creates a standard reference data model for a whole enterprise. At the same time, it separates the problems of source data extraction and integration from those of data warehouse population. In some cases, the **reconciled layer** is also directly used to accomplish better some operational tasks, such as producing daily reports that cannot be satisfactorily prepared using the corporate applications or generating data flows to feed external processes periodically to benefit from cleaning and integration.

This architecture is especially useful for the extensive, enterprise-wide systems. A disadvantage of this structure is the extra file storage space used through the extra redundant reconciled layer. It also makes the analytical tools a little further away from being real-time.



Three-Tier Architecture for a data warehouse system

Three-Tier Data Warehouse Architecture

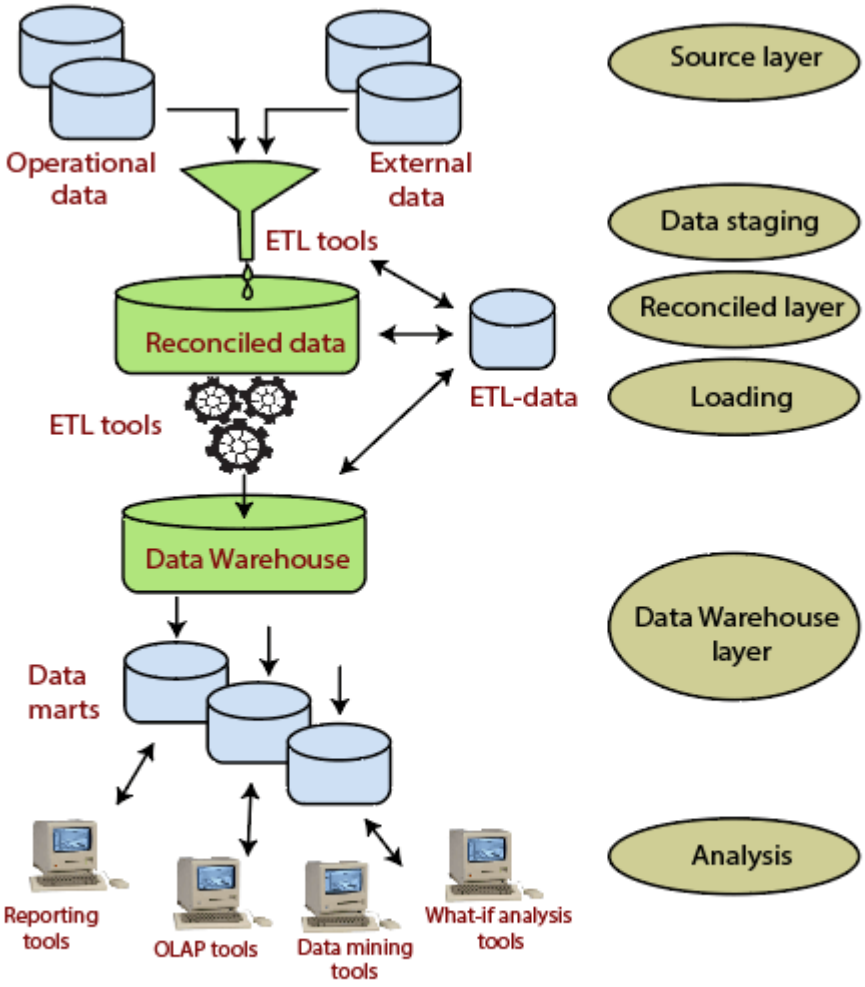
Data Warehouses usually have a three-level (tier) architecture that includes:

1. Bottom Tier (Data Warehouse Server)
2. Middle Tier (OLAP Server)
3. Top Tier (Front end Tools).

A **bottom-tier** that consists of the **Data Warehouse server**, which is almost always an RDBMS. It may include several specialized data marts and a metadata repository.

Data from operational databases and external sources (such as user profile data provided by external consultants) are extracted using application program interfaces called a gateway. A gateway is provided by the underlying DBMS and allows customer programs to generate SQL code to be executed at a server.

Examples of gateways contain **ODBC** (Open Database Connection) and **OLE-DB** (Open-Linking and Embedding for Databases), by **Microsoft**, and **JDBC** (Java Database Connection).



Three-Tier Architecture for a data warehouse system

A **middle-tier** which consists of an **OLAP server** for fast querying of the data warehouse.

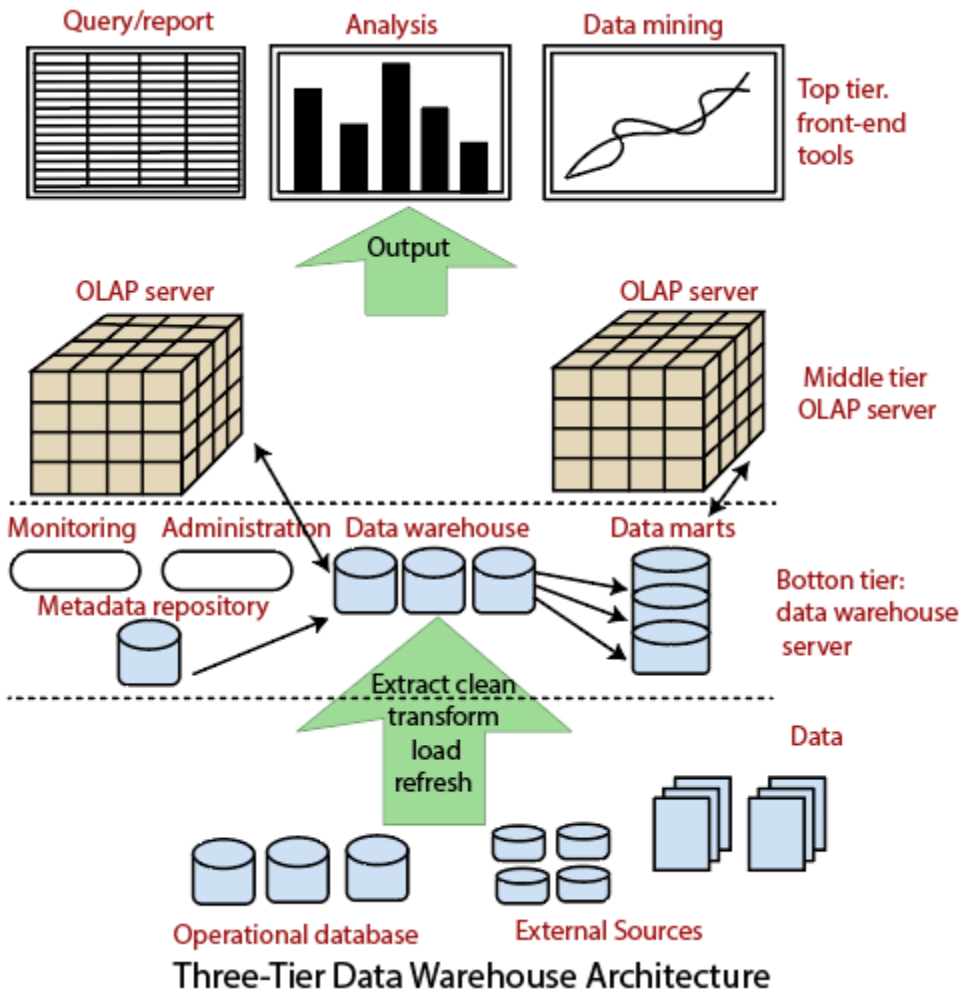
The OLAP server is implemented using either

(1) A **Relational OLAP (ROLAP) model**, i.e., an extended relational DBMS that maps functions on multidimensional data to standard relational operations.

(2) A **Multidimensional OLAP (MOLAP) model**, i.e., a particular purpose server that directly implements multidimensional information and operations.

A **top-tier** that contains **front-end tools** for displaying results provided by OLAP, as well as additional tools for data mining of the OLAP-generated data.

The overall Data Warehouse Architecture is shown in fig:

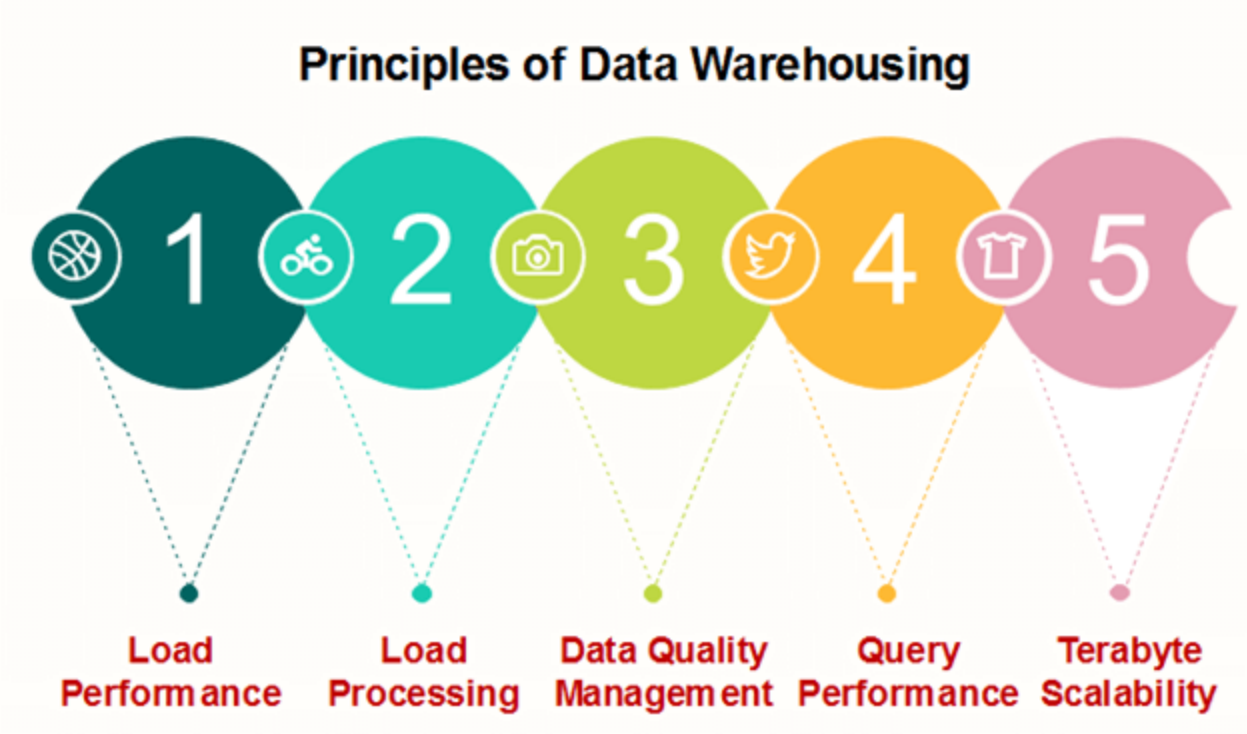


Three-Tier Data Warehouse Architecture

The **metadata repository** stores information that defines DW objects. It includes the following parameters and information for the middle and the top-tier applications:

1. A description of the DW structure, including the warehouse schema, dimension, hierarchies, data mart locations, and contents, etc.
2. Operational metadata, which usually describes the currency level of the stored data, i.e., active, archived or purged, and warehouse monitoring information, i.e., usage statistics, error reports, audit, etc.
3. System performance data, which includes indices, used to improve data access and retrieval performance.
4. Information about the mapping from operational databases, which provides source **RDBMSs** and their contents, cleaning and transformation rules, etc.
5. Summarization algorithms, predefined queries, and reports business data, which include business terms and definitions, ownership information, etc.

Principles of Data Warehousing



Load Performance

Data warehouses require increase loading of new data periodically basis within narrow time windows; performance on the load process should be measured in hundreds of millions of rows and gigabytes per hour and must not artificially constrain the volume of data business.

Load Processing

Many phases must be taken to load new or update data into the data warehouse, including data conversion, filtering, reformatting, indexing, and metadata update.

Data Quality Management

Fact-based management demands the highest data quality. The warehouse ensures local consistency, global consistency, and referential integrity despite "dirty" sources and massive database size.

Query Performance

Fact-based management must not be slowed by the performance of the data warehouse RDBMS; large, complex queries must be complete in seconds, not days.

Terabyte Scalability

Data warehouse sizes are growing at astonishing rates. Today these size from a few to hundreds of gigabytes and terabyte-sized data warehouses.

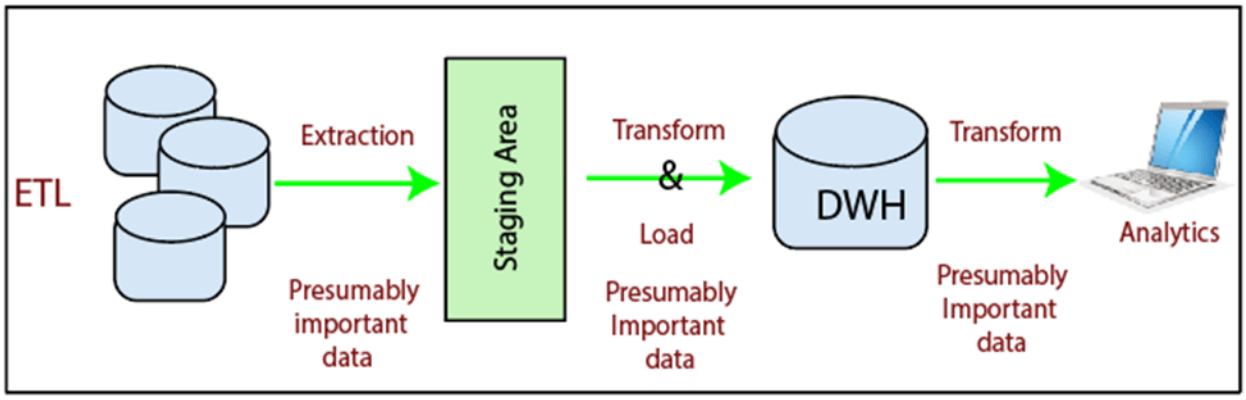
ETL (Extract, Transform, and Load) Process

What is ETL?

The mechanism of extracting information from source systems and bringing it into the data warehouse is commonly called **ETL**, which stands for **Extraction, Transformation and Loading**.

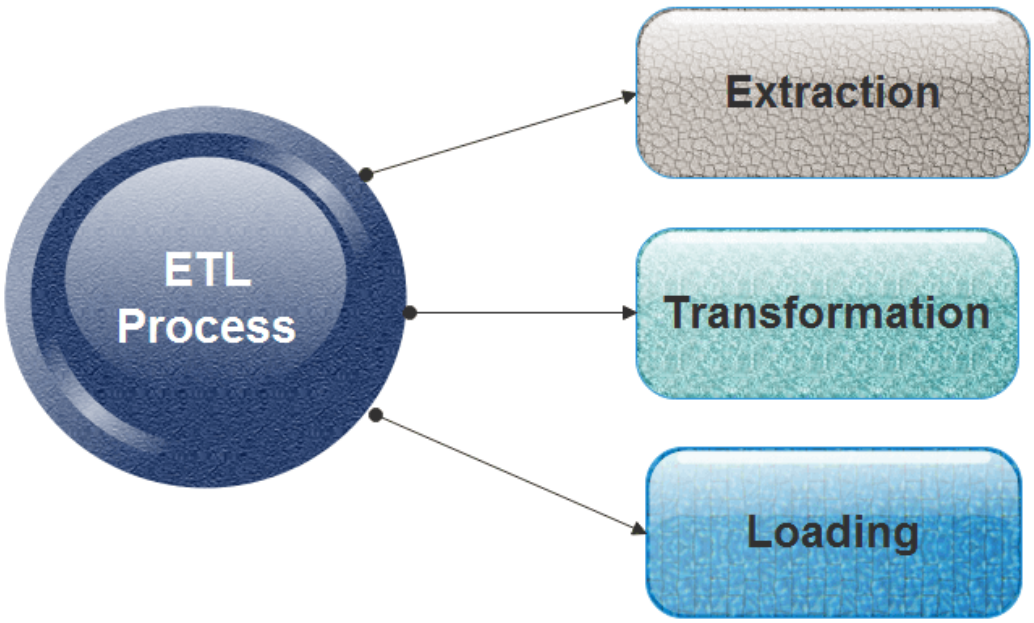
The ETL process requires active inputs from various stakeholders, including developers, analysts, testers, top executives and is technically challenging.

To maintain its value as a tool for decision-makers, Data warehouse technique needs to change with business changes. ETL is a recurring method (daily, weekly, monthly) of a Data warehouse system and needs to be agile, automated, and well documented.



How ETL Works?

ETL consists of three separate phases:



Extraction

- Extraction is the operation of extracting information from a source system for further use in a data warehouse environment. This is the first stage of the ETL process.
- Extraction process is often one of the most time-consuming tasks in the ETL.
- The source systems might be complicated and poorly documented, and thus determining which data needs to be extracted can be difficult.
- The data has to be extracted several times in a periodic manner to supply all changed data to the warehouse and keep it up-to-date.

Cleansing

The cleansing stage is crucial in a data warehouse technique because it is supposed to improve data quality. The primary data cleansing features found in ETL tools are rectification and homogenization. They use specific dictionaries to rectify typing mistakes and to recognize synonyms, as well as rule-based cleansing to enforce domain-specific rules and defines appropriate associations between values.

The following examples show the essential of data cleaning:

If an enterprise wishes to contact its users or its suppliers, a complete, accurate and up-to-date list of contact addresses, email addresses and telephone numbers must be available.

If a client or supplier calls, the staff responding should be quickly able to find the person in the enterprise database, but this need that the caller's name or his/her company name is listed in the database.

If a user appears in the databases with two or more slightly different names or different account numbers, it becomes difficult to update the customer's information.

Transformation

Transformation is the core of the reconciliation phase. It converts records from its operational source format into a particular data warehouse format. If we implement a three-layer architecture, this phase outputs our reconciled data layer.

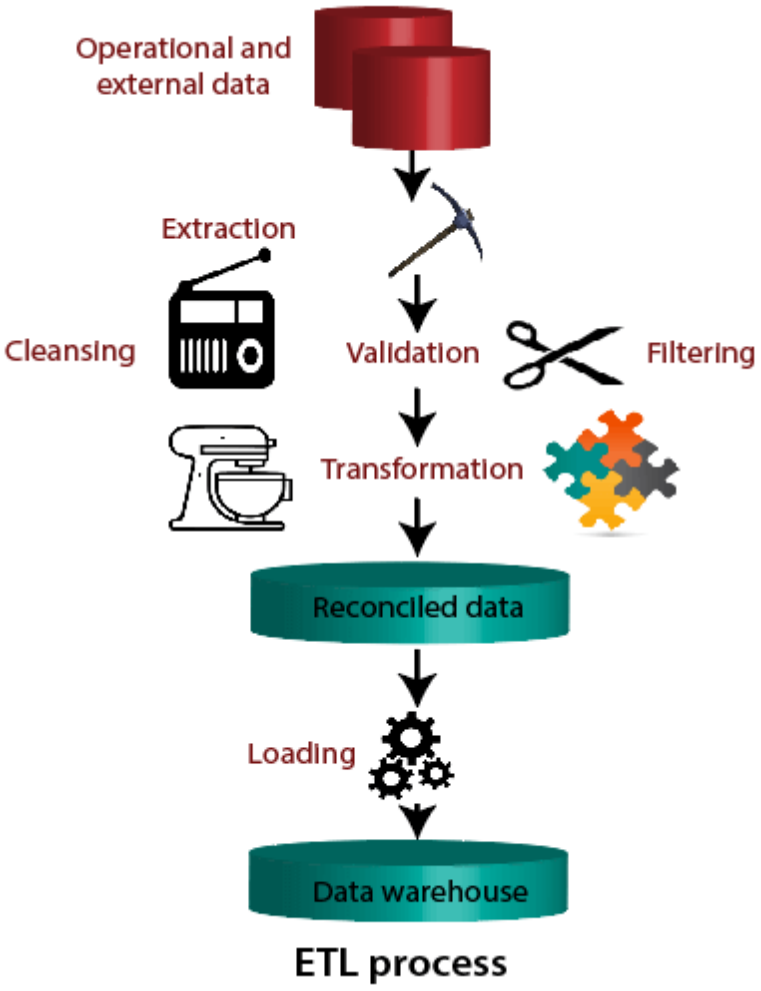
The following points must be rectified in this phase:

- Loose texts may hide valuable information. For example, XYZ PVT Ltd does not explicitly show that this is a Limited Partnership company.
- Different formats can be used for individual data. For example, data can be saved as a string or as three integers.

Following are the main transformation processes aimed at populating the reconciled data layer:

- Conversion and normalization that operate on both storage formats and units of measure to make data uniform.
- Matching that associates equivalent fields in different sources.
- Selection that reduces the number of source fields and records.

Cleansing and **Transformation** processes are often closely linked in ETL tools.



Loading

The **Load** is the process of writing the data into the target database. During the load step, it is necessary to ensure that the load is performed correctly and with as little resources as possible.

Loading can be carried in two ways:

1. **Refresh:** Data Warehouse data is completely rewritten. This means that older file is replaced. Refresh is usually used in combination with static extraction to populate a data warehouse initially.

2. **Update:** Only those changes applied to source information are added to the Data Warehouse. An update is typically carried out without deleting or modifying preexisting data. This method is used in combination with incremental extraction to update data warehouses regularly.

Selecting an ETL Tool

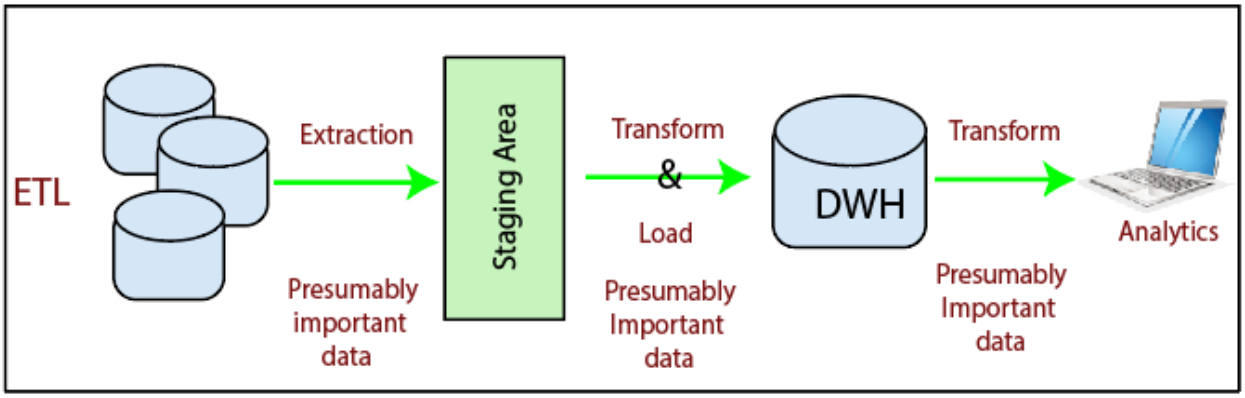
Selection of an appropriate ETL Tools is an important decision that has to be made in choosing the importance of an ODS or data warehousing application. The ETL tools are required to provide coordinated access to multiple data sources so that relevant data may be extracted from them. An ETL tool would generally contains tools for data cleansing, re-organization, transformations, aggregation, calculation and automatic loading of information into the object database.

An ETL tool should provide a simple user interface that allows data cleansing and data transformation rules to be specified using a point-and-click approach. When all mappings and transformations have been defined, the ETL tool should automatically generate the data extract/transformation/load programs, which typically run in batch mode.

Difference between ETL and ELT

ETL (Extract, Transform, and Load)

Extract, Transform and Load is the technique of extracting the record from sources (which is present outside or on-premises, etc.) to a staging area, then transforming or reformatting with business manipulation performed on it in order to fit the operational needs or data analysis, and later loading into the goal or destination databases or data warehouse.



Strengths

Development Time: Designing from the output backwards provide that only information applicable to the solution is extracted and processed, potentially decreasing development, delete, and processing overhead.

Targeted data: Due to the targeted feature of the load process, the warehouse contains only information relevant to the presentation. Reduced warehouse content simplify the security regime enforce and hence the administration overheads.

Tools Availability: The number of tools available that implement ETL provides the flexibility of approach and the opportunity to identify the most appropriate tool. The proliferation of tools has to lead to a competitive functionality war, which often results in loss of maintainability.

Weaknesses

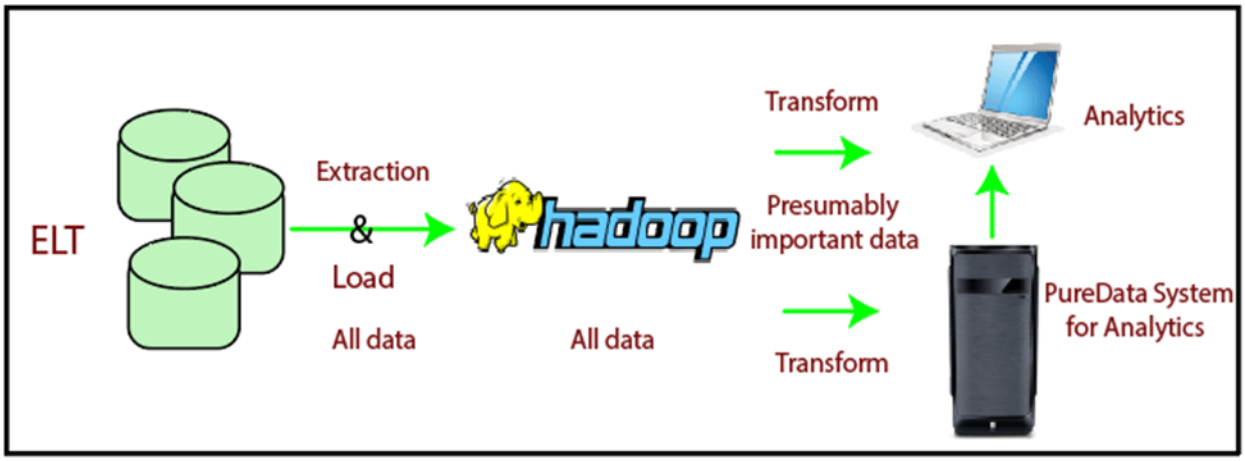
Flexibility: Targeting only relevant information for output means that any future requirements that may need data that was not included in the original design will need to be added to the ETL routines. Due to the nature of tight dependency between the methods developed, this often leads to a need for fundamental redesign and development. As a result, this increase the time and cost involved.

Hardware: Most third-party tools utilize their engine to implement the ETL phase. Regardless of the estimate of the solution, this can necessitate the investment in additional hardware to implement the tool's ETL engine. The use of third-party tools to achieve the ETL process compels the information of new scripting languages and processes.

Learning Curve: Implementing a third-party tools that uses foreign processes and languages results in the learning curve that is implicit in all technologies new to an organization and can often lead to consecutive blind alleys in their use due to shortage of experience.

ELT (Extract, Load and Transform)

ELT stands for Extract, Load and Transform is the various sight while looking at data migration or movement. ELT involves the extraction of aggregate information from the source system and loading to the target method instead of transformation between the extraction and loading phase. Once the data is copied or loaded into the target method, then change takes place.



The **extract** and **load** step can be isolated from the transformation process. Isolating the load phase from the transformation process delete an inherent dependency between these phases. In addition to containing the data necessary for the transformations, the extract and load process can include components of data that may be essential in the future. The load phase could take the entire source and loaded it into the warehouses.

Separating the phases enables the project to be damaged down into smaller chunks, thus making it more specific and manageable.

Performing the data integrity analysis in the staging method enables a further phase in the process to be isolated and dealt with at the most appropriate point in the process. This method also helps to ensure that only cleaned and checked information is loaded into the warehouse for transformation.

Isolating the transformations from the load steps helps to encourage a more staged way to the warehouse design and implementation.

Strengths

Project Management: Being able to divide the warehouse method into specific and isolated functions, enables a project to be designed on a smaller function basis, therefore the project can be broken down into feasible chunks.

Flexible & Future Proof: In general, in an ELT implementation, all record from the sources are loaded into the data warehouse as part of the extract and loading process. This, linked with the isolation of the transformation phase, means that future requirements can easily be incorporated into the data warehouse architecture.

Risk minimization: Deleting the close interdependencies between each technique of the warehouse build system enables the development method to be isolated, and the individual process design can thus also be separated. This provides a good platform for change, maintenance and management.

Utilize Existing Hardware: In implementing ELT as a warehouse build process, the essential tools provided with the database engine can be used.

Utilize Existing Skill sets: By using the functionality support by the database engine, the existing investment in database functions are re-used to develop the warehouse. No new skills need to be learned, and the full weight of the experience in developing the engine?s technology is utilized, further reducing the cost and risk in the development process.

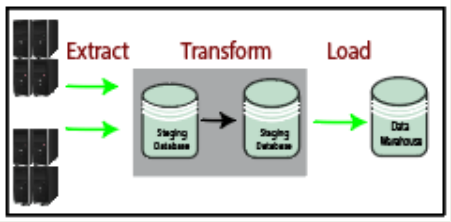
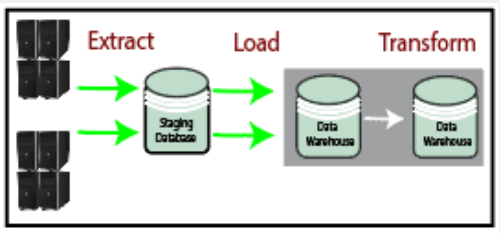
Weaknesses

Against the Norm: ELT is a new method to data warehouse design and development. While it has proven itself many times over through its abundant use in implementations throughout the world, it does require a change in mentality and design approach against traditional methods.

Tools Availability: Being an emergent technology approach, ELT suffers from the limited availability of tools.

Difference between ETL vs. ELT

Basics	ETL	ELT
Process	Data is transferred to the ETL server and moved back to DB. High network bandwidth required.	Data remains in the DB except for cross Database loads (e.g. source to object).
Transformation	Transformations are performed in ETL Server.	Transformations are performed (in the source or) in the target.
Code Usage	Typically used for o Source to target transfer o Compute-intensive Transformations o Small amount of data	Typically used for o High amounts of data
Time-Maintenance	It needs highs maintenance as you need to select data to load and transform.	Low maintenance as data is always available.
Calculations	Overwrites existing column or Need to append the dataset and push to the target platform.	Easily add the calculated column to the existing table.

Analysis	 Extract Transform Load Staging Database Staging Database Data Warehouse	 Extract Load Transform Staging Database Data Warehouse Data Warehouse