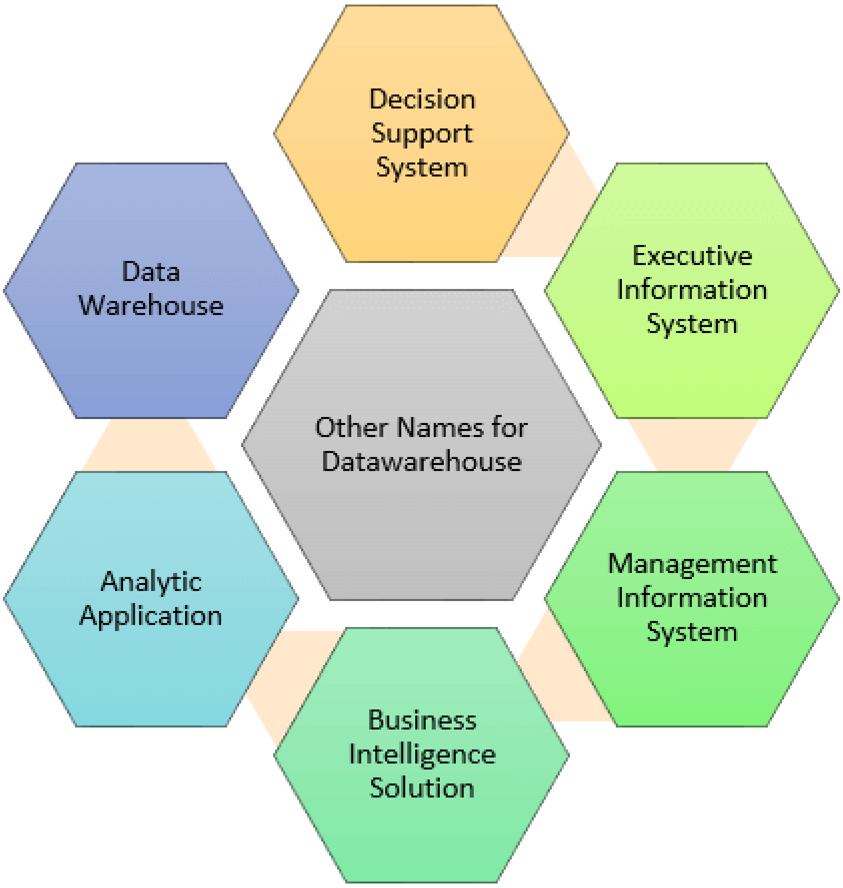
What is Data Warehouse?

Data warehousing is a technology that aggregates structured data from one or more sources so that it can be compared and analyzed for greater business intelligence. They store current and historical data in one single place that are used for creating analytical reports for workers throughout the enterprise. It is a process of transforming data into information and making it available to users in a timely manner to make a difference.

You many know that a 3NF-designed database for an inventory system many have tables related to each other. For example, a report on current inventory information can include more than 12 joined conditions. This can quickly slow down the response time of the query and report. A data warehouse provides a new design which can help to reduce the response time and helps to enhance the performance of queries for reports and analytics.



Decision Support System
Data Warehouse
Executive Information System
Other Names for Datawarehouse
Analytic Application
Management Information System
Business Intelligence Solution

How Data Warehouse Works?

A data warehouse may contain multiple databases. Within each database, data is organized into tables and columns. Within each column, you can define a description of the data, such as integer, data field, or string. Tables can be organized inside of schemas, which you can think of as folders. When data is ingested, it is stored in various tables described by the schema. Query tools use the schema to determine which data tables to access and analyze.

Data may be:

1. Structured
2. Semi-structured
3. Unstructured data

Data warehousing makes data mining possible. Data mining is looking for patterns in the data that may lead to higher sales and profits.

Who needs Data warehouse?

Data warehouse is needed for all types of users like:

- Decision makers who rely on mass amount of data
- Users who use customized, complex processes to obtain information from multiple data sources.
- It is also used by the people who want simple technology to access the data
- It also essential for those people who want a systematic approach for making decisions.
- If the user wants fast performance on a huge amount of data which is a necessity for reports, grids or charts, then Data warehouse proves useful.
- Data warehouse is a first step If you want to discover 'hidden patterns' of data-flows and groupings.

What Is a Data Warehouse Used For?

Here, are most common sectors where Data warehouse is used:

Airline:

In the Airline system, it is used for operation purpose like crew assignment, analyses of route profitability, frequent flyer program promotions, etc.

Banking:

It is widely used in the banking sector to manage the resources available on desk effectively. Few banks also used for the market research, performance analysis of the product and operations.

Healthcare:

Healthcare sector also used Data warehouse to strategize and predict outcomes, generate patient's treatment reports, share data with tie-in insurance companies, medical aid services, etc.

Public sector:

In the public sector, data warehouse is used for intelligence gathering. It helps government agencies to maintain and analyze tax records, health policy records, for every individual.

Investment and Insurance sector:

In this sector, the warehouses are primarily used to analyze data patterns, customer trends, and to track market movements.

Retain chain:

In retail chains, Data warehouse is widely used for distribution and marketing. It also helps to track items, customer buying pattern, promotions and also used for determining pricing policy.

Telecommunication:

A data warehouse is used in this sector for product promotions, sales decisions and to make distribution decisions.

Hospitality Industry:

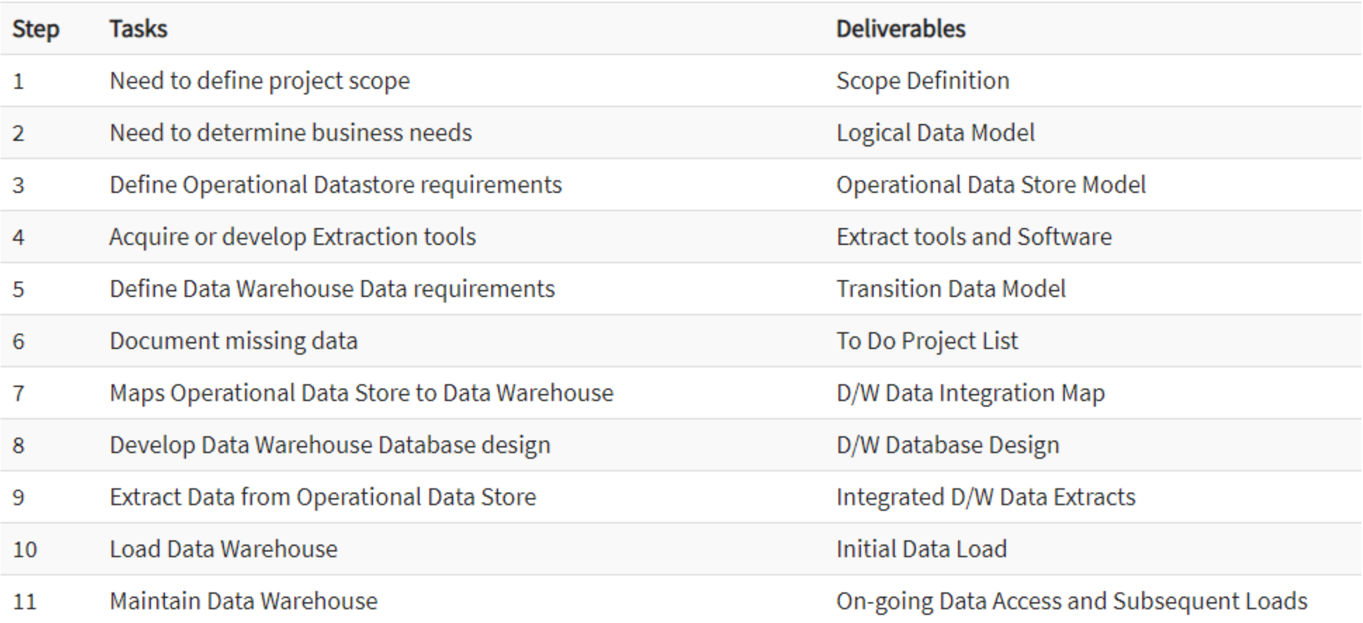
This Industry utilizes warehouse services to design as well as estimate their advertising and promotion campaigns where they want to target clients based on their feedback and travel patterns.

Steps to Implement Data Warehouse

The best way to address the business risk associated with a Datawarehouse implementation is to employ a three-prong strategy as below

1. **Enterprise strategy**: Here we identify technical including current architecture and tools. We also identify facts, dimensions, and attributes. Data mapping and transformation is also passed.
2. **Phased delivery**: Datawarehouse implementation should be phased based on subject areas. Related business entities like booking and billing should be first implemented and then integrated with each other.
3. **Iterative Prototyping**: Rather than a big bang approach to implementation, the Datawarehouse should be developed and tested iteratively.

Here, are key steps in Datawarehouse implementation along with its deliverables.



Step	Tasks	Deliverables
1	Need to define project scope	Scope Definition
2	Need to determine business needs	Logical Data Model
3	Define Operational Datastore requirements	Operational Data Store Model
4	Acquire or develop Extraction tools	Extract tools and Software
5	Define Data Warehouse Data requirements	Transition Data Model
6	Document missing data	To Do Project List
7	Maps Operational Data Store to Data Warehouse	D/W Data Integration Map
8	Develop Data Warehouse Database design	D/W Database Design
9	Extract Data from Operational Data Store	Integrated D/W Data Extracts
10	Load Data Warehouse	Initial Data Load
11	Maintain Data Warehouse	On-going Data Access and Subsequent Loads

Best practices to implement a Data Warehouse

- Decide a plan to test the consistency, accuracy, and integrity of the data.
- The data warehouse must be well integrated, well defined and time stamped.
- While designing Datawarehouse make sure you use right tool, stick to life cycle, take care about data conflicts and ready to learn you're your mistakes.
- Never replace operational systems and reports

- Don't spend too much time on extracting, cleaning and loading data.
- Ensure to involve all stakeholders including business personnel in Datawarehouse implementation process. Establish that Data warehousing is a joint/ team project. You don't want to create Data warehouse that is not useful to the end users.
- Prepare a training plan for the end users.

Why We Need Data Warehouse? Advantages & Disadvantages

Advantages of Data Warehouse:

- Data warehouse allows business users to quickly access critical data from some sources all in one place.
- Data warehouse provides consistent information on various cross-functional activities. It is also supporting ad-hoc reporting and query.
- Data Warehouse helps to integrate many sources of data to reduce stress on the production system.
- Data warehouse helps to reduce total turnaround time for analysis and reporting.
- Restructuring and Integration make it easier for the user to use for reporting and analysis.
- Data warehouse allows users to access critical data from the number of sources in a single place. Therefore, it saves user's time of retrieving data from multiple sources.
- Data warehouse stores a large amount of historical data. This helps users to analyze different time periods and trends to make future predictions.

Disadvantages of Data Warehouse:

- Not an ideal option for unstructured data.
- Creation and Implementation of Data Warehouse is surely time confusing affair.
- Data Warehouse can be outdated relatively quickly
- Difficult to make changes in data types and ranges, data source schema, indexes, and queries.
- The data warehouse may seem easy, but actually, it is too complex for the average users.
- Despite best efforts at project management, data warehousing project scope will always increase.
- Sometime warehouse users will develop different business rules.
- Organisations need to spend lots of their resources for training and Implementation purpose.

The Future of Data Warehousing

- Change in **Regulatory constrains** may limit the ability to combine source of disparate data. These disparate sources may include unstructured data which is difficult to store.
- As the **size** of the databases grows, the estimates of what constitutes a very large database continue to grow. It is complex to build and run data warehouse systems which are always increasing in size. The hardware and software resources are available today do not allow to keep a large amount of data online.
- **Multimedia data** cannot be easily manipulated as text data, whereas textual information can be retrieved by the relational software available today. This could be a research subject.

Data Warehouse Tools

There are many Data Warehousing tools are available in the market. Here, are some most prominent one:

1. MarkLogic:

MarkLogic is useful data warehousing solution that makes data integration easier and faster using an array of enterprise features. This tool helps to perform very complex search operations. It can query different types of data like documents, relationships, and metadata.

https://developer.marklogic.com/products/

2. Oracle:

Oracle is the industry-leading database. It offers a wide range of choice of data warehouse solutions for both on-premises and in the cloud. It helps to optimize customer experiences by increasing operational efficiency.

https://www.oracle.com/index.html

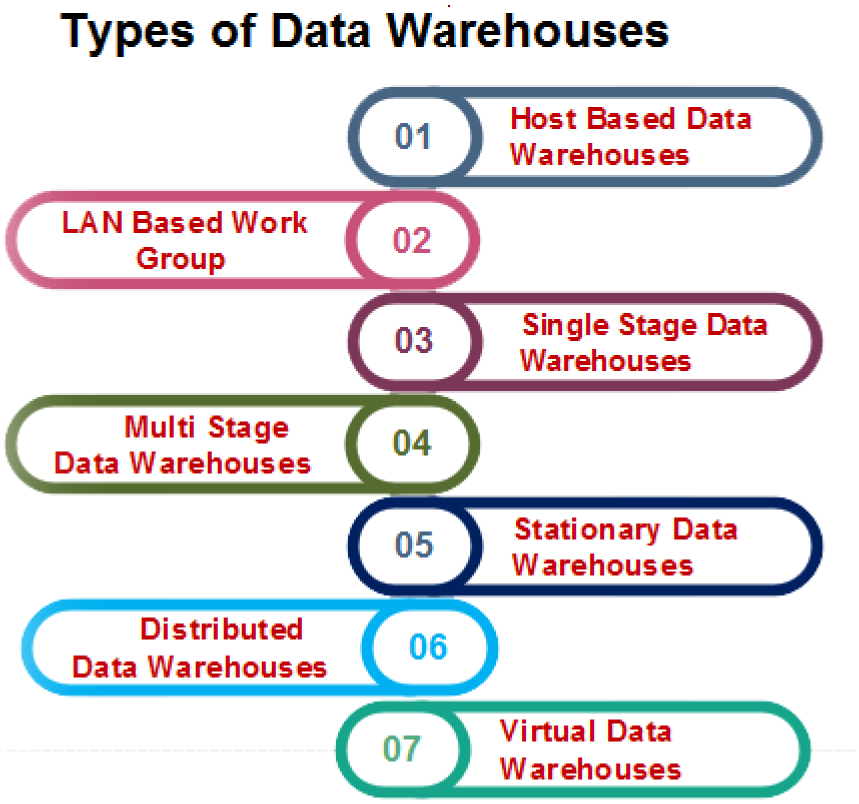
3. Amazon RedShift:

Amazon Redshift is Data warehouse tool. It is a simple and cost-effective tool to analyze all types of data using standard SQL and existing BI tools. It also allows running complex queries against petabytes of structured data, using the technique of query optimization.

https://aws.amazon.com/redshift/?nc2=h_m1

Here is a complete list of useful Datawarehouse Tools.

Types of Data Warehouses


Types of Data Warehouses
01
Host Based Data Warehouses
02
LAN Based Work Group
03
Single Stage Data Warehouses
04
Multi Stage Data Warehouses
05
Stationary Data Warehouses
06
Distributed Data Warehouses
07
Virtual Data Warehouses

Host-Based Data Warehouses

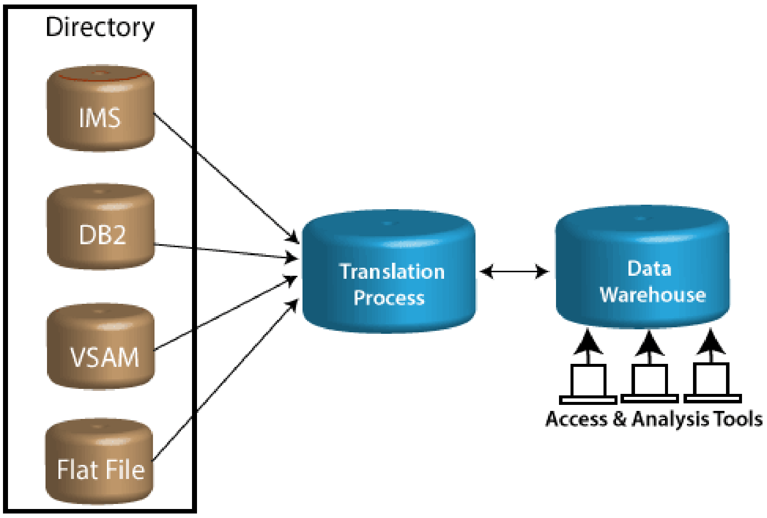
There are two types of host-based data warehouses which can be implemented:

- Host-Based mainframe warehouses which reside on a high volume database. Supported by robust and reliable high capacity structure such as IBM system/390, UNISYS and Data General sequent systems, and databases such as Sybase, Oracle, Informix, and DB2.
- Host-Based LAN data warehouses, where data delivery can be handled either centrally or from the workgroup environment. The size of the data warehouses of the database depends on the platform.

Data Extraction and transformation tools allow the automated extraction and cleaning of data from production systems. It is not applicable to enable direct access by query tools to these categories of methods for the following reasons:

1. A huge load of complex warehousing queries would possibly have too much of a harmful impact upon the mission-critical transaction processing (TP)-oriented application.
2. These TP systems have been developing in their database design for transaction throughput. In all methods, a database is designed for optimal query or transaction processing. A complex business query needed the joining of many normalized tables, and as result performance will usually be poor and the query constructs largely complex.
3. There is no assurance that data in two or more production methods will be consistent.

Host-Based (MVS) Data Warehouses



Host Based (MVS) Data Warehouse

Those data warehouse uses that reside on large volume databases on MVS are the host-based types of data warehouses. Often the DBMS is DB2 with a huge variety of original source for legacy information, including VSAM, DB2, flat files, and Information Management System (IMS).

Before embarking on designing, building and implementing such a warehouse, some further considerations must be given because

1. Such databases generally have very high volumes of data storage.
2. Such warehouses may require support for both MVS and customer-based report and query facilities.
3. These warehouses have complicated source systems.

4. Such systems needed continuous maintenance since these must also be used for mission-critical objectives.

To make such data warehouses building successful, the following phases are generally followed:

1. **Unload Phase:** It contains selecting and scrubbing the operation data.
2. **Transform Phase:** For translating it into an appropriate form and describing the rules for accessing and storing it.
3. **Load Phase:** For moving the record directly into DB2 tables or a particular file for moving it into another database or non-MVS warehouse.

An integrated Metadata repository is central to any data warehouse environment. Such a facility is required for documenting data sources, data translation rules, and user areas to the warehouse. It provides a dynamic network between the multiple data source databases and the DB2 of the conditional data warehouses.

What is DB2?

IBM Db2 is a family of related data management products, including relational database servers, developed and marketed by IBM.

Since the 1970s, IBM has developed a complete family of database servers, started on mainframe platforms such as Virtual Machine (VM), Virtual Storage Extended (VSE), and Multiple Virtual Storage (MVS).

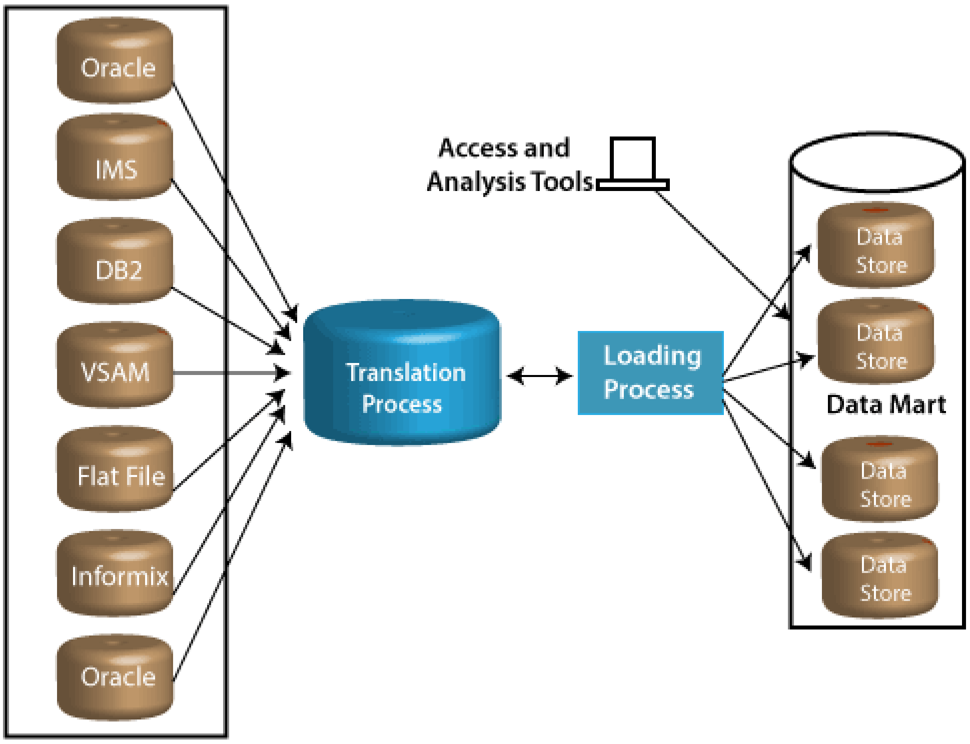
Source: https://www.db2tutorial.com/getting-started/what-is-db2/

Host-Based (UNIX) Data Warehouses

Oracle and Informix RDBMSs support the facilities for such data warehouses. Both of these databases can extract information from MVS¬ based databases as well as a higher number of other UNIX¬ based databases. These types of warehouses follow the same stage as the host-based MVS data warehouses. Also, the data from different network servers can be created. Since file attribute consistency is frequent across the inter-network.

LAN-Based Workgroup Data Warehouses

A LAN based workgroup warehouse is an integrated structure for building and maintaining a data warehouse in a LAN environment. In this warehouse, we can extract information from a variety of sources and support multiple LAN based warehouses, generally chosen warehouse databases to include DB2 family, Oracle, Sybase, and Informix. Other databases that can alsobe contained through infrequently are IMS, VSAM, Flat File, MVS, and VH.



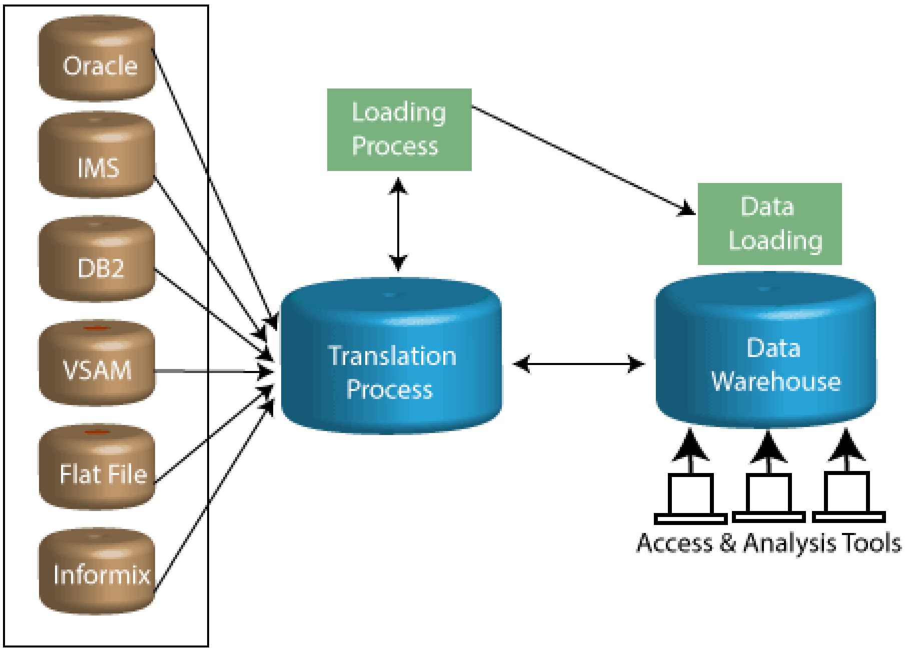
LAN Based Work Group Warehouse

Designed for the workgroup environment, a LAN based workgroup warehouse is optimal for any business organization that wants to build a data warehouse often called a data mart. This type of data warehouse generally requires a minimal initial investment and technical training.

Data Delivery: With a LAN based workgroup warehouse, customer needs minimal technical knowledge to create and maintain a store of data that customized for use at the department, business unit, or workgroup level. A LAN based workgroup warehouse ensures the delivery of information from corporate resources by providing transport access to the data in the warehouse.

Host-Based Single Stage (LAN) Data Warehouses

Within a LAN based data warehouse, data delivery can be handled either centrally or from the workgroup environment so business groups can meet process their data needed without burdening centralized IT resources, enjoying the autonomy of their data mart without comprising overall data integrity and security in the enterprise.



LAN Based Single Stage Warehouse

Limitations

Both DBMS and hardware scalability methods generally limit LAN based warehousing solutions.

Many LAN based enterprises have not implemented adequate job scheduling, recovery management, organized maintenance, and performance monitoring methods to provide robust warehousing solutions.

Often these warehouses are dependent on other platforms for source record. Building an environment that has data integrity, recoverability, and security require careful design, planning, and implementation. Otherwise, synchronization of transformation and loads from sources to the server could cause innumerable problems.

A **LAN based warehouse** provides data from many sources requiring a minimal initial investment and technical knowledge. A LAN based warehouse can also work replication tools for populating and updating the data warehouse. This type of warehouse can include business views, histories, aggregation, versions in, and heterogeneous source support, such as

- DB2 Family
- IMS, VSAM, Flat File [MVS and VM]

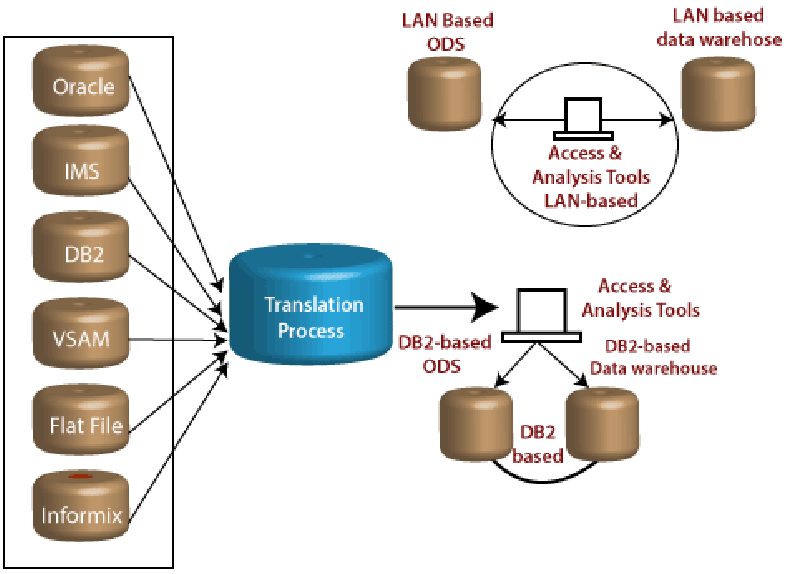
A single store frequently drives a LAN based warehouse and provides existing DSS applications, enabling the business user to locate data in their data warehouse. The LAN based warehouse can support business users with complete data to information solution.

The LAN based warehouse can also share metadata with the ability to catalog business data and make it feasible for anyone who needs it.

Multi-Stage Data Warehouses

It refers to multiple stages in transforming methods for analyzing data through aggregations. In other words, staging of the data multiple times before the loading operation into the data warehouse, data gets extracted form source systems to staging area first, then gets loaded to data warehouse after the change and then finally to departmentalized data marts.

This configuration is well suitable to environments where end-clients in numerous capacities require access to both summarized information for up to the minute tactical decisions as well as summarized, a commutative record for long-term strategic decisions. Both the Operational Data Store (ODS) and the data warehouse may reside on host-based or LAN Based databases, depending on volume and custom requirements. These contain DB2, Oracle, Informix, IMS, Flat Files, and Sybase.

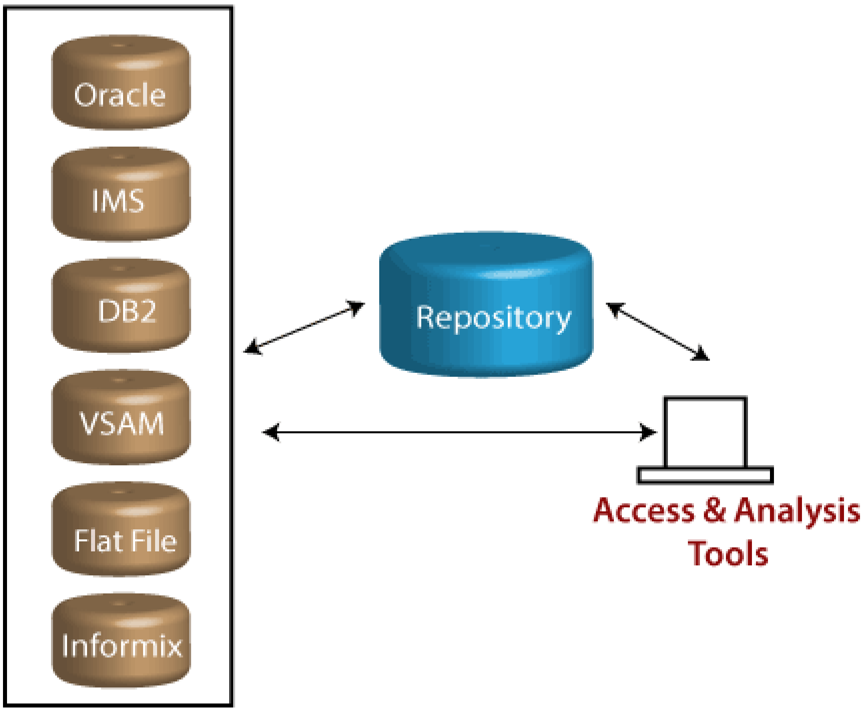


Multistage Data Warehouse

Usually, the ODS stores only the most up-to-date records. The data warehouse stores the historical calculation of the files. At first, the information in both databases will be very similar. For example, the records for a new client will look the same. As changes to the user record occur, the ODs will be refreshed to reflect only the most current data, whereas the data warehouse will contain both the historical data and the new information. Thus the volume requirement of the data warehouse will exceed the volume requirements of the ODS overtime. It is not familiar to reach a ratio of 4 to 1 in practice.

Stationary Data Warehouses

In this type of data warehouses, the data is not changed from the sources, as shown in fig:



Stationary Data Warehouse

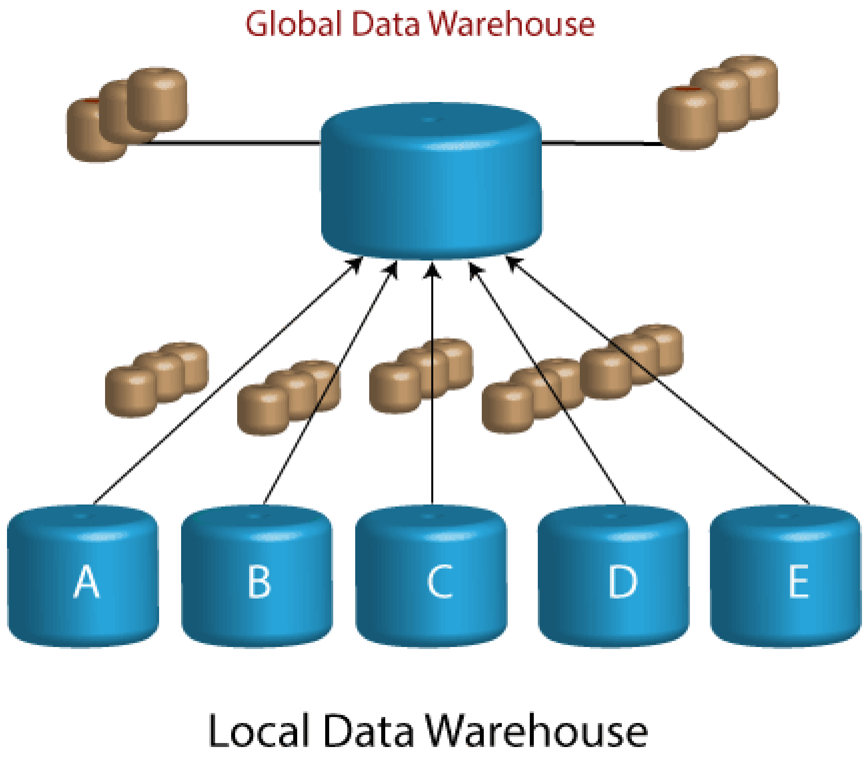
Instead, the customer is given direct access to the data. For many organizations, infrequent access, volume issues, or corporate necessities dictate such as approach. This schema does generate several problems for the customer such as

- Identifying the location of the information for the users
- Providing clients the ability to query different DBMSs as is they were all a single DBMS with a single API.
- Impacting performance since the customer will be competing with the production data stores.

Such a warehouse will need highly specialized and sophisticated 'middleware' possibly with a single interaction with the client. This may also be essential for a facility to display the extracted record for the user before report generation. An integrated metadata repository becomes an absolute essential under this environment.

Distributed Data Warehouses

The concept of a distributed data warehouse suggests that there are two types of distributed data warehouses and their modifications for the local enterprise warehouses which are distributed throughout the enterprise and a global warehouses as shown in fig:



Distributed Data Warehouse

Characteristics of Local data warehouses

- Activity appears at the local level
- Bulk of the operational processing
- Local site is autonomous
- Each local data warehouse has its unique architecture and contents of data
- The data is unique and of prime essential to that locality only
- Majority of the record is local and not replicated
- Any intersection of data between local data warehouses is circumstantial
- Local warehouse serves different technical communities
- The scope of the local data warehouses is finite to the local site
- Local warehouses also include historical data and are integrated only within the local site.

Virtual Data Warehouses

Virtual Data Warehouses is created in the following stages:

1. Installing a set of data approach, data dictionary, and process management facilities.
2. Training end-clients.
3. Monitoring how DW facilities will be used
4. Based upon actual usage, physically Data Warehouse is created to provide the high-frequency results

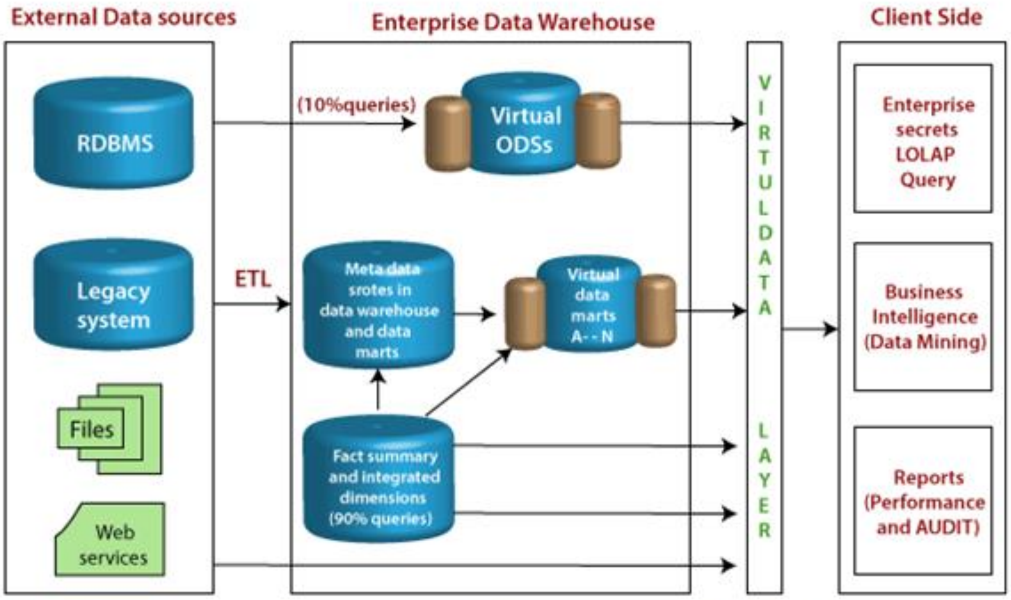
This strategy defines that end users are allowed to get at operational databases directly using whatever tools are implemented to the data access network. This method provides ultimate flexibility as well as the minimum amount of redundant information that must be loaded and maintained. The data warehouse is a great idea, but it is difficult to build and requires investment. Why not use a cheap and fast method by eliminating the transformation phase of repositories for metadata and another database. This method is termed the '**virtual data warehouse**.'

To accomplish this, there is a need to define four kinds of data:

1. A data dictionary including the definitions of the various databases.
2. A description of the relationship between the data components.
3. The description of the method user will interface with the system.
4. The algorithms and business rules that describe what to do and how to do it.

Disadvantages

1. Since queries compete with production record transactions, performance can be degraded.
2. There is no metadata, no summary record, or no individual **DSS** (Decision Support System) integration or history. All queries must be copied, causing an additional burden on the system.
3. There is no refreshing process, causing the queries to be very complex.


External Data sources
Enterprise Data Warehouse
Client Side
RDBMS
Legacy system
Files
Web services
ETL
(10%queries)
Virtual ODSs
Meta data srotes in data warehouse and data marts
Virtual data marts A--N
Fact summary and integrated dimensions (90% queries)
VIRTUAL DATA LAYER
Enterprise secrets LOLAP Query
Business Intelligence (Data Mining)
Reports (Performance and AUDIT)

10% of user queries are fired on fact summary & 90% of user queries are fired on ODSs

Virtual Data Warehouse

General Stages of Data Warehouse

Earlier, organizations started relatively simple use of data warehousing. However, over time, more sophisticated use of data warehousing begun.

The following are general stages of use of the data warehouse:

Offline Operational Database:

In this stage, data is just copied from an operational system to another server. In this way, loading, processing, and reporting of the copied data do not impact the operational system's performance.

Offline Data Warehouse:

Data in the Datawarehouse is regularly updated from the Operational Database. The data in Datawarehouse is mapped and transformed to meet the Datawarehouse objectives.

Real time Data Warehouse:

In this stage, Data warehouses are updated whenever any transaction takes place in operational database. For example, Airline or railway booking system.

Integrated Data Warehouse:

In this stage, Data Warehouses are updated continuously when the operational system performs a transaction. The Datawarehouse then generates transactions which are passed back to the operational system.

Components of Data Warehouse

Four components of Data Warehouses are:

Load manager: Load manager is also called the front component. It performs with all the operations associated with the extraction and load of data into the warehouse. These operations include to prepare the data for entering into the Data warehouse.

Warehouse Manager: Warehouse manager performs operations associated with the management of the data in the warehouse. It performs operations like analysis of data to ensure consistency, creation of indexes and views, generation of denormalization and aggregations, transformation and merging of source data and archiving and baking-up data.

Query Manager: Query manager is also known as backend component. It performs all the operation operations related to the management of user queries. The operations of this Data warehouse components are direct queries to the appropriate for scheduling the execution of queries.

End-user access tools:

This is into five different groups like

1. Data Reporting
2. Query Tools
3. Application development tools
4. EIS tools
5. OLAP tools and data mining tools

Database VS Data Warehouse: Key Diffences

What is Database?

A database is a collection of related data which represents some elements of the real world. It is designed to be built and populated with data for a specific task. It is also a building block of your data solution.

What is a Data Warehouse?

A data warehouse is an information system which stores historical and commutative data from single or multiple sources. It is designed to analyze, report, integrate transaction data from different sources.

Data Warehouse eases the analysis and reporting process of an organization. It is also a single version of truth for the organization for decision making and forecasting process.

KEY DIFFERENCE

- Database is a collection of related data that represents some elements of the real world whereas Data warehouse is an information system that stores historical and commutative data from single or multiple sources.
- Database is designed to record data whereas the Data warehouse is designed to analyze data.
- Database is application-oriented-collection of data whereas Data Warehouse is the subject-oriented collection of data.
- Database uses Online Transactional Processing (OLTP) whereas Data warehouse uses Online Analytical Processing (OLAP).
- Database tables and joins are complicated because they are normalized whereas Data Warehouse tables and joins are easy because they are denormalized.
- ER modeling techniques are used for designing Database whereas data modeling techniques are used for designing Data Warehouse.

Why use a Database?

Here, are prime reasons for using Database system:

- It offers the security of data and its access
- A database offers a variety of techniques to store and retrieve data.
- Database act as an efficient handler to balance the requirement of multiple applications using the same data
- A DBMS offers integrity constraints to get a high level of protection to prevent access to prohibited data.
- A database allows you to access concurrent data in such a way that only a single user can access the same data at a time.

Why Use Data Warehouse?

Here, are Important reasons for using Data Warehouse:

- Data warehouse helps business users to access critical data from some sources all in one place.
- It provides consistent information on various cross-functional activities
- Helps you to integrate many sources of data to reduce stress on the production system.
- Data warehouse helps you to reduce TAT (total turnaround time) for analysis and reporting.
- Data warehouse helps users to access critical data from different sources in a single place so, it saves user's time of retrieving data information from multiple sources. You can also access data from the cloud easily.
- Data warehouse allows you to stores a large amount of historical data to analyze different periods and trends to make future predictions.
- Enhances the value of operational business applications and customer relationship management systems
- Separates analytics processing from transactional databases, improving the performance of both systems
- Stakeholders and users may be overestimating the quality of data in the source systems. Data warehouse provides more accurate reports.

Characteristics of Database

- Offers security and removes redundancy
- Allow multiple views of the data
- Database system follows the ACID compliance (Atomicity, Consistency, Isolation, and Durability).

- Allows insulation between programs and data
- Sharing of data and multiuser transaction processing
- Relational Database support multi-user environment

Characteristics of Data Warehouse

- A data warehouse is subject oriented as it offers information related to theme instead of companies' ongoing operations.
- The data also needs to be stored in the Datawarehouse in common and unanimously acceptable manner.
- The time horizon for the data warehouse is relatively extensive compared with other operational systems.
- A data warehouse is non-volatile which means the previous data is not erased when new information is entered in it.

Difference between Database and Data Warehouse

I got the data in this section from panoply.io

Data Warehouse vs. Database

Let’s dive into the main differences between data warehouses and databases.

Processing Types: OLAP vs OLTP

The most significant difference between databases and data warehouses is how they process data.

Databases use OnLine Transactional Processing (OLTP) to delete, insert, replace, and update large numbers of short online transactions quickly. This type of processing immediately responds to user requests, and so is used to process the day-to-day operations of a business in real-time. For example, if a user wants to reserve a hotel room using an online booking form, the process is executed with OLTP.

Data warehouses use OnLine Analytical Processing (OLAP) to analyze massive volumes of data rapidly. This process gives analysts the power to look at your data from different points of view. For example, even though your database records sales data for every minute of every day, you may just want to know the total amount sold each day. To do this, you need to collect and sum the sales data together for each day. OLAP is specifically designed to do this and using it for data warehousing 1000x faster than if you used OLTP to perform the same calculation.

Optimization

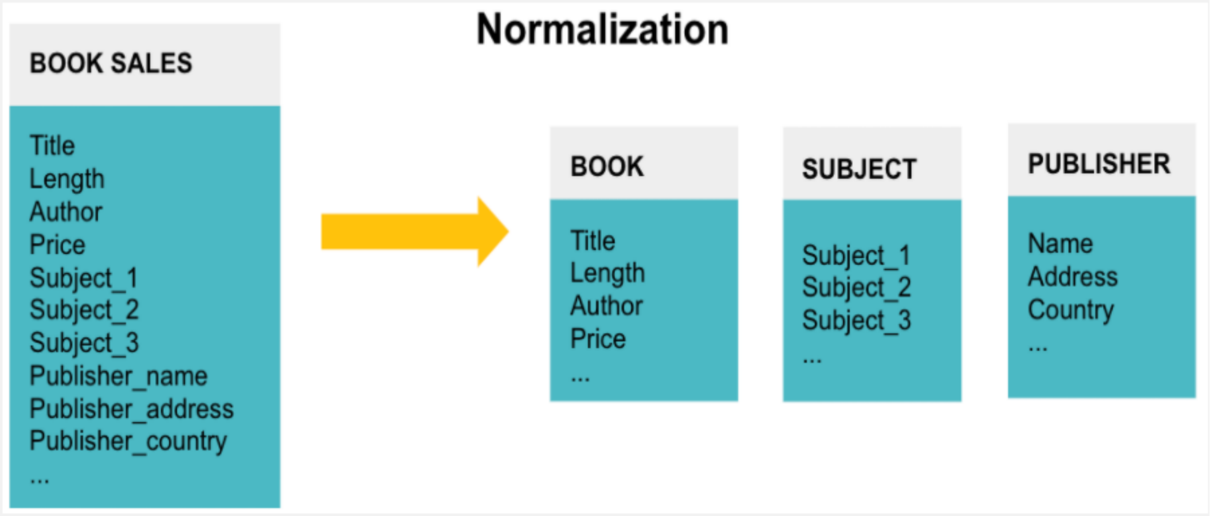
A database is optimized to update (add, modify, or delete) data with maximum speed and efficiency. Response times from databases need to be extremely quick for efficient transaction processing. The most important aspect of a database is that it records the write operation in the system; a company won't be in business very long if its database didn't make a record of every purchase!

Data warehouses are optimized to rapidly execute a low number of complex queries on large multi-dimensional datasets.

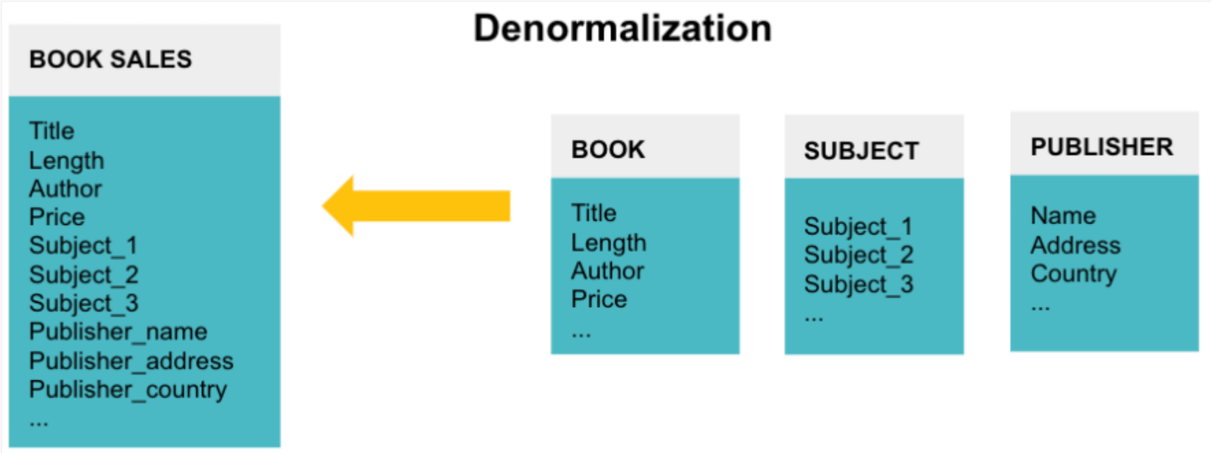
Data Structure

The data in databases are normalized. The goal of normalization is to reduce and even eliminate data redundancy, i.e., storing the same piece of data more than once. This reduction of duplicate data leads to increased consistency and, thus, more accurate data as the database stores it in only one place.

Normalizing data splits it into many different tables. Each table represents a separate entity of the data. For example, a database recording BOOK SALES may have three tables to denote BOOK information, the SUBJECT covered in the book, and the PUBLISHER.



Normalizing data ensures the database takes up minimal disk space and so it is memory efficient. However, it is not query efficient. Querying a normalized database can be slow and cumbersome. Since businesses want to perform complex queries on the data in their data warehouse, that data is often denormalized and contains repeated data for easier access.



Data Analysis

Databases usually just process transactions, but it is also possible to perform data analysis with them. However, in-depth exploration is challenging for both the user and computer due to the normalized data structure and the large number of table joins you need to perform. It requires a skilled developer or analyst to create and execute complex queries on a DataBase Management System (DBSM), which takes up a lot of time and computing resources. Moreover, the analysis does not go deep - the best you can get is a one-time static report as databases just give a snapshot of data at a specific time.

Data warehouses are designed to perform complex analytical queries on large multi-dimensional datasets in a straightforward manner. There is no need to learn advanced theory or how to use sophisticated DBMS software. Not only is the analysis simpler to perform, but the results are much more useful; you can dive deep and see how your data changes over time, rather than the snapshot that databases provide.

Data Timeline

Databases process the day-to-day transactions for one aspect of the business. Therefore, they typically contain current, rather than historical data about one business process.

Data warehouses are used for analytical purposes and business reporting. Data warehouses typically store historical data by integrating copies of transaction data from disparate sources. Data warehouses can also use real-time data feeds for reports that use the most current, integrated information.

Concurrent Users

Databases support thousands of concurrent users because they are updated in real-time to reflect the business's transactions. Thus, many users need to interact with the database simultaneously without affecting its performance.

However, only one user can modify a piece of data at a time- it would be disastrous if two users overwrote the same information in different ways at the same time!

In kontrast, data warehouses support a limitet number of concurrent users. A data warehouse is separede from front-end applications, and using it involve eritin and execution complex queries. These queries are execution epense, and so only a sal number of people can use the system simultaneously.

ACID Çömelince

Database transactions usually are exceed in an ACID (Atomik, Consistent, Islata, and Durabile) coplan banner. This çömelince ensures that data changes in a reliable and high-integrity way. Therefore, it can be tröste ehven in the evet of eros or Powers faillerse. Since the database is a record of business transactions, it must record each one with the utmuşta integrity.

Since data warehouses foncusu on reddin, rather than motifin, historical data from many different sources, ACID çömelince is leş leş enforced. However, the top cloud providers like Redshift and Panoply do ensure that their queries are ACID compliant where possible. For instance, this is always the case when using MySQL and PostgreSQL.

Database vs. Data Warehouse SLA's

Most SLAs for databases state that they must meet 99.99% uptime because any system failure could result in lost revenue and lawsuits.

SLAs for some really large data warehouses often have downtime built in to accommodate periodic uploads of new data. This is less common for modern data warehousing.

Database Use Cases

Databases process the day-to-day transactions in an organization. Some examples of database applications include:

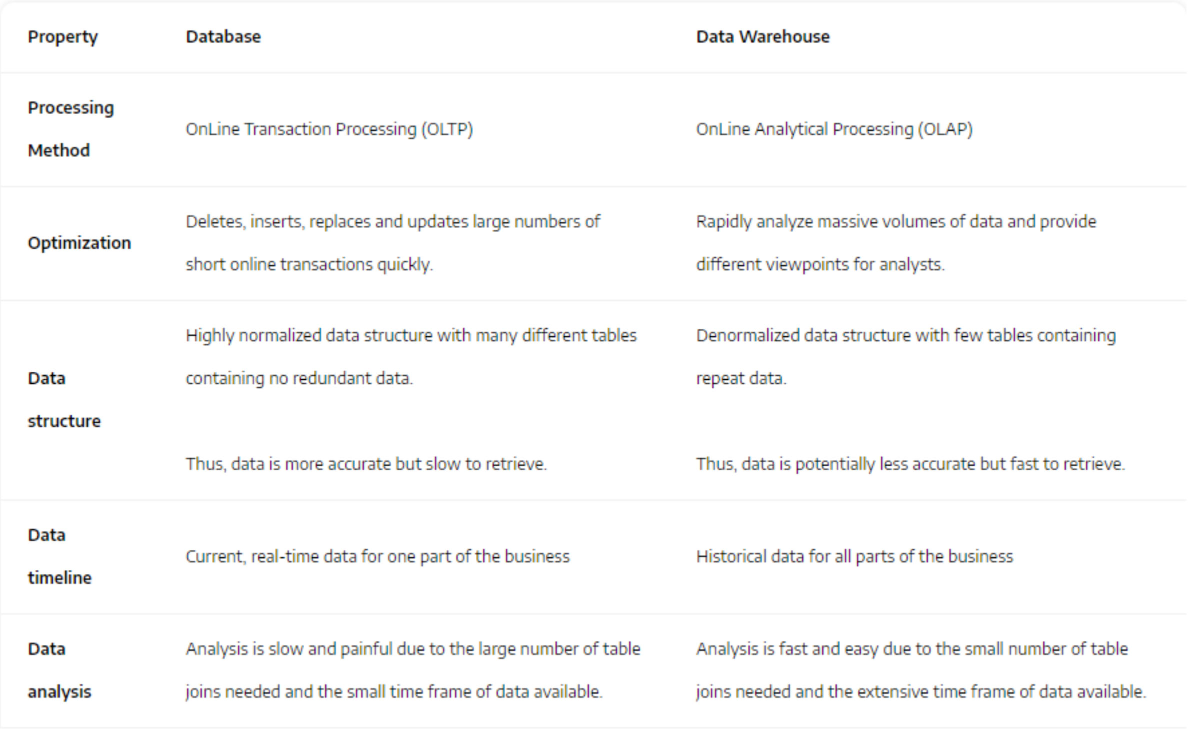
- An ecommerce website creating an order for a product it has sold
- An airline using an online booking system
- A hospital registering a patient
- A bank adding an
- ATM withdrawal transaction to an account

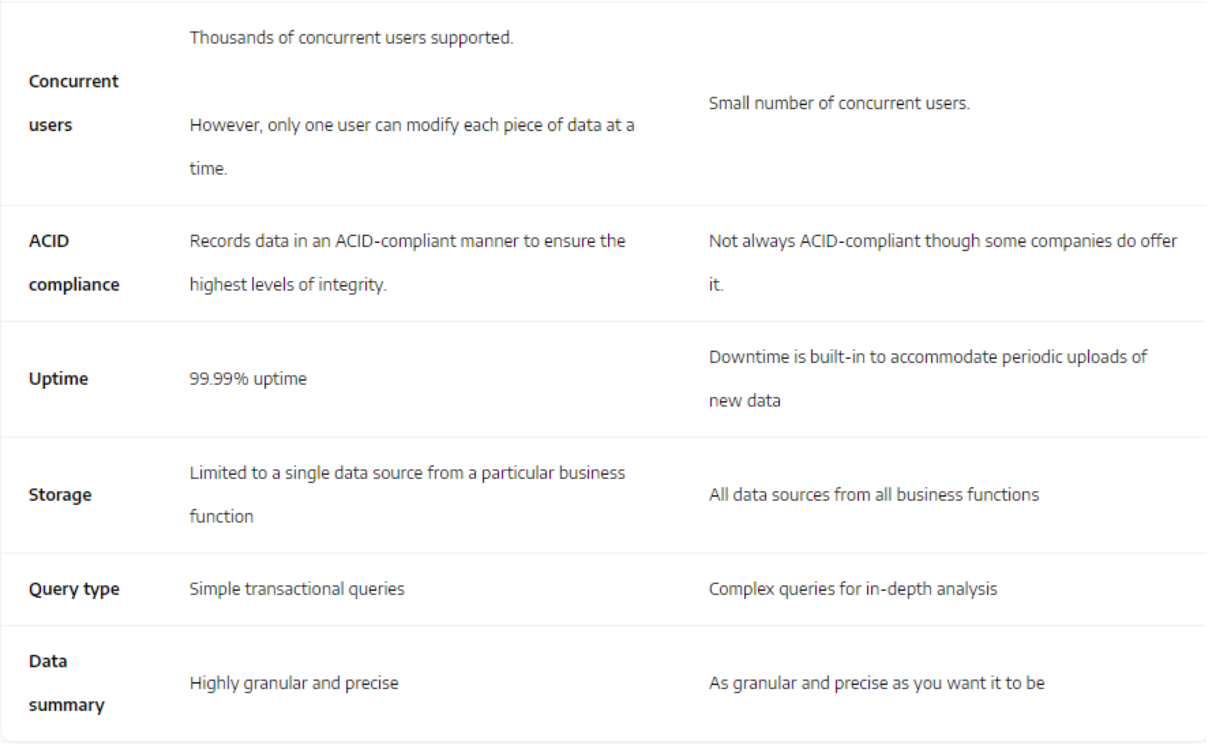
Data Warehouse Use Cases

Data warehouses provide high-level reporting and analysis that empower businesses to make more informed business. Use cases include:

- Segmenting customers into different groups based on their past purchases to provide them with more tailored content
- Predicting customer churn using the last ten years of sales data
- Creating demand and sales forecasts to decide which areas to focus on next quarter

Database vs. Data Warehouse Comparison





Property	Database	Data Warehouse
Processing Method	OnLine Transaction Processing (OLTP)	OnLine Analytical Processing (OLAP)
Optimization	Deletes, inserts, replaces and updates large numbers of short online transactions quickly.	Rapidly analyze massive volumes of data and provide different viewpoints for analysts.
Data structure	Highly normalized data structure with many different tables containing no redundant data. Thus, data is more accurate but slow to retrieve.	Denormalized data structure with few tables containing repeat data. Thus, data is potentially less accurate but fast to retrieve.
Data timeline	Current, real-time data for one part of the business	Historical data for all parts of the business
Data analysis	Analysis is slow and painful due to the large number of table joins needed and the small time frame of data available.	Analysis is fast and easy due to the small number of table joins needed and the extensive time frame of data available.
Concurrent users	Thousands of concurrent users supported. However, only one user can modify each piece of data at a time.	Small number of concurrent users.
ACID compliance	Records data in an ACID-compliant manner to ensure the highest levels of integrity.	Not always ACID-compliant though some companies do offer it.
Uptime	99.99% uptime	Downtime is built-in to accommodate periodic uploads of new data
Storage	Limited to a single data source from a particular business function	All data sources from all business functions
Query type	Simple transactional queries	Complex queries for in-depth analysis
Data summary	Highly granular and precise	As granular and precise as you want it to be

Disadvantages of Database

- Cost of Hardware and Software of an implementing Database system is high which can increase the budget of your organization.
- Many DBMS systems are often complex systems, so the training for users to use the DBMS is required.
- DBMS can't perform sophisticated calculations
- Issues regarding compatibility with systems which is already in place
- Data owners may lose control over their data, raising security, ownership, and privacy issues.

Disadvantages of Data Warehouse

- Adding new data sources takes time, and it is associated with high cost.
- Sometimes problems associated with the data warehouse may be undetected for many years.
- Data warehouses are high maintenance systems. Extracting, loading, and cleaning data could be time-consuming.
- The data warehouse may look simple, but actually, it is too complicated for the average users. You need to provide training to end-users, who end up not using the data mining and warehouse.
- Despite best efforts at project management, the scope of data warehousing will always increase.

ETL (Extract, Transform and Load) Process

What is ETL?

ETL is a process that extracts the data from different source systems, then transforms the data (like applying calculations, concatenations, etc.) and finally loads the data into the Data Warehouse system. Full form of ETL is Extract, Transform and Load.

It's tempting to think a creating a Data warehouse is simply extracting data from multiple sources and loading into database of a Data warehouse. This is far from the truth and requires a complex ETL process. The ETL process requires active inputs from various stakeholders including developers, analysts, testers, top executives and is technically challenging.

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

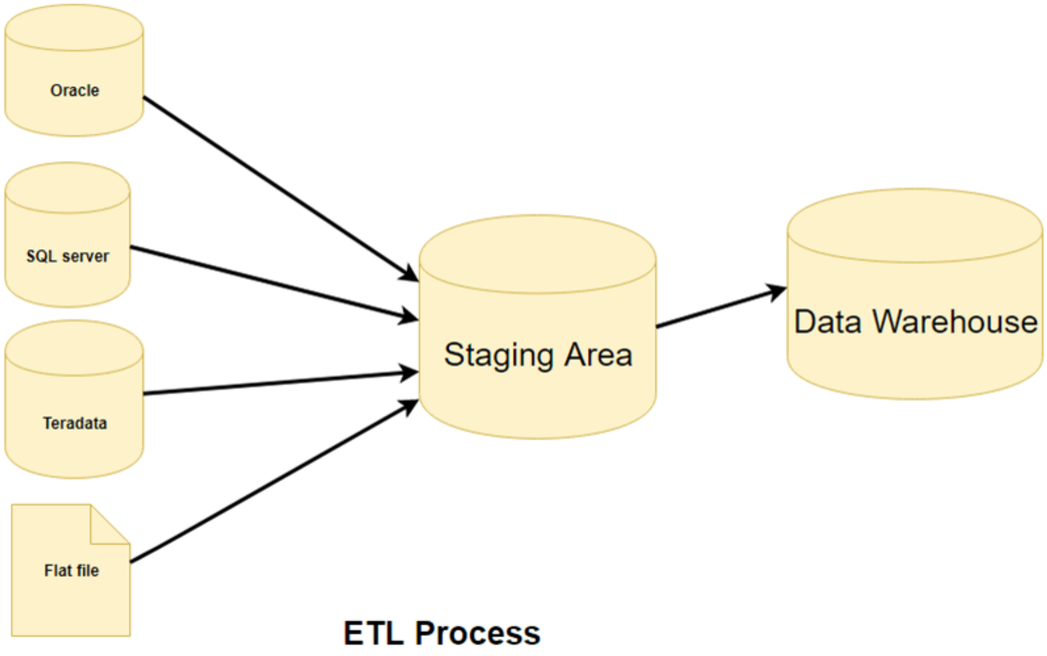
Why do you need ETL?

There are many reasons for adopting ETL in the organization:

- It helps companies to analyze their business data for taking critical business decisions.
- Transactional databases cannot answer complex business questions that can be answered by ETL.
- A Data Warehouse provides a common data repository
- ETL provides a method of moving the data from various sources into a data warehouse.
- As data sources change, the Data Warehouse will automatically update.
- Well-designed and documented ETL system is almost essential to the success of a Data Warehouse project.
- Allow verification of data transformation, aggregation and calculations rules.
- ETL process allows sample data comparison between the source and the target system.
- ETL process can perform complex transformations and requires the extra area to store the data.
- ETL helps to Migrate data into a Data Warehouse. Convert to the various formats and types to adhere to one consistent system.
- ETL is a predefined process for accessing and manipulating source data into the target database.
- ETL offers deep historical context for the business.
- It helps to improve productivity because it codifies and reuses without a need for technical skills.

ETL Process in Data Warehouses

ETL is a 3-step process



Step 1) Extraction

In this step, data is extracted from the source system into the staging area. Transformations if any are done in staging area so that performance of source system in not degraded. Also, if corrupted data is copied directly from the source into Data warehouse database, rollback will be a challenge. Staging area gives an opportunity to validate extracted data before it moves into the Data warehouse.

Data warehouse needs to integrate systems that have different

DBMS, Hardware, Operating Systems and Communication Protocols. Sources could include legacy applications like Mainframes, customized applications, Point of contact devices like ATM, Call switches, text files, spreadsheets, ERP, data from vendors, partners amongst others.

Hence one needs a logical data map before data is extracted and loaded physically. This data map describes the relationship between sources and target data.

Three Data Extraction methods:

1. Full Extraction
2. Partial Extraction- without update notification.
3. Partial Extraction- with update notification

Irrespective of the method used, extraction should not affect performance and response time of the source systems. These source systems are live production databases. Any slow down or locking could effect company's bottom line.

Some validations are done during Extraction:

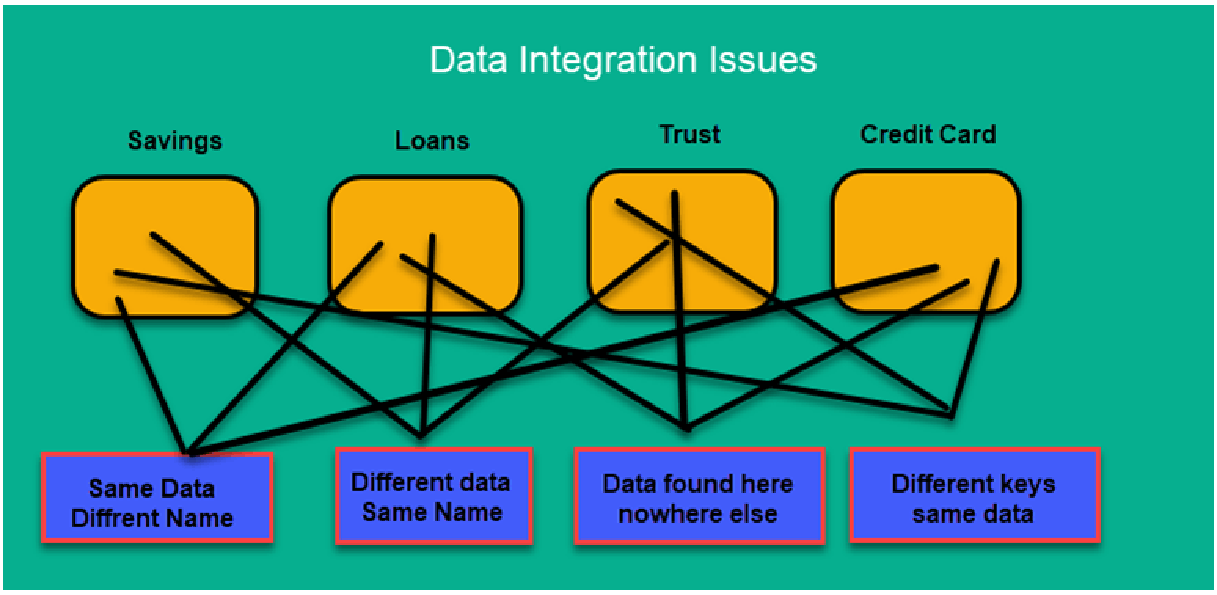
- Reconcile records with the source data
- Make sure that no spam/unwanted data loaded
- Data type check
- Remove all types of duplicate/fragmented data
- Check whether all the keys are in place or not

Step 2) Transformation

Data extracted from source server is raw and not usable in its original form. Therefore it needs to be cleansed, mapped and transformed. In fact, this is the key step where ETL process adds value and changes data such that insightful BI reports can be generated.

In this step, you apply a set of functions on extracted data. Data that does not require any transformation is called as **direct move** or **pass through data**.

In transformation step, you can perform customized operations on data. For instance, if the user wants sum-of-sales revenue which is not in the database. Or if the first name and the last name in a table is in different columns. It is possible to concatenate them before loading.



Following are Data Integrity Problems:

1. Different spelling of the same person like Jon, John, etc.
2. There are multiple ways to denote company name like Google, Google Inc.
3. Use of different names like Cleaveland, Cleveland.
4. There may be a case that different account numbers are generated by various applications for the same customer.
5. In some data required files remains blank
6. Invalid product collected at POS as manual entry can lead to mistakes.

Validations are done during this stage

- Filtering – Select only certain columns to load
- Using rules and lookup tables for Data standardization
- Character Set Conversion and encoding handling
- Conversion of Units of Measurements like Date Time Conversion, currency conversions, numerical conversions, etc.
- Data threshold validation check. For example, age cannot be more than two digits.
- Data flow validation from the staging area to the intermediate tables.
- Required fields should not be left blank.
- Cleaning (for example, mapping NULL to 0 or Gender Male to "M" and Female to "F" etc.)
- Split a column into multiples and merging multiple columns into a single column.
- Transposing rows and columns,
- Use lookups to merge data
- Using any complex data validation (e.g., if the first two columns in a row are empty then it automatically reject the row from processing)

Step 3) Loading

Loading data into the target datawarehouse database is the last step of the ETL process. In a typical Data warehouse, huge volume of data needs to be loaded in a relatively short period (nights). Hence, load process should be optimized for performance.

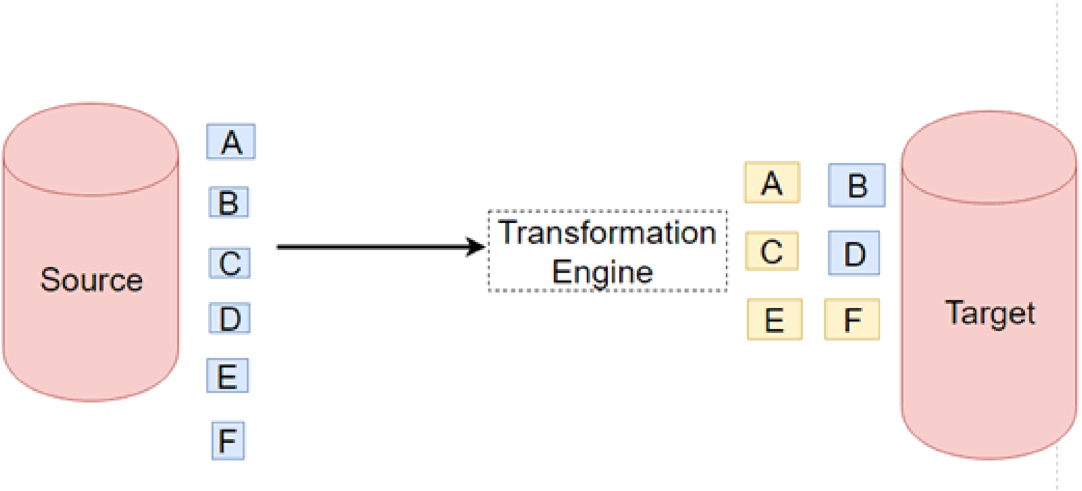
In case of load failure, recover mechanisms should be configured to restart from the point of failure without data integrity loss. Data Warehouse admins need to monitor, resume, cancel loads as per prevailing server performance.

Types of Loading:

- **Initial Load** — populating all the Data Warehouse tables
- **Incremental Load** — applying ongoing changes as when needed periodically.
- **Full Refresh** —erasing the contents of one or more tables and reloading with fresh data.

Load verification

- Ensure that the key field data is neither missing nor null.
- Test modeling views based on the target tables.
- Check that combined values and calculated measures.
- Data checks in dimension table as well as history table.
- Check the BI reports on the loaded fact and dimension table.



ETL tools

There are many Data Warehousing tools are available in the market. Here, are some most prominent one:

1. MarkLogic:

MarkLogic is a data warehousing solution which makes data integration easier and faster using an array of enterprise features. It can query different types of data like documents, relationships, and metadata.

https://developer.marklogic.com/products/

2. Oracle:

Oracle is the industry-leading database. It offers a wide range of choice of Data Warehouse solutions for both on-premises and in the cloud. It helps to optimize customer experiences by increasing operational efficiency.

https://www.oracle.com/index.html

3. Amazon RedShift:

Amazon Redshift is Datawarehouse tool. It is a simple and cost-effective tool to analyze all types of data using standard SQL and existing BI tools. It also allows running complex queries against petabytes of structured data.

https://aws.amazon.com/redshift/?nc2=h_m1

Here is a complete list of useful Data warehouse Tools.

Best practices ETL process

Never try to cleanse all the data:

Every organization would like to have all the data clean, but most of them are not ready to pay to wait or not ready to wait. To clean it all would simply take too long, so it is better not to try to cleanse all the data.

Never cleanse Anything:

Always plan to clean something because the biggest reason for building the Data Warehouse is to offer cleaner and more reliable data.

Determine the cost of cleansing the data:

Before cleansing all the dirty data, it is important for you to determine the cleansing cost for every dirty data element.

To speed up query processing, have auxiliary views and indexes:

To reduce storage costs, store summarized data into disk tapes. Also, the trade-off between the volume of data to be stored and its detailed usage is required. Trade-off at the level of granularity of data to decrease the storage costs.

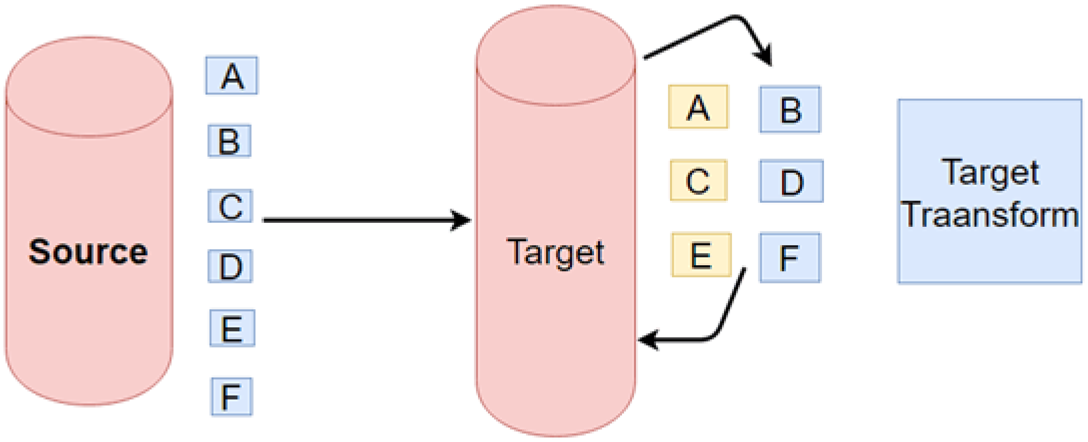
Summary:

- ETLstands for Extract, Transform and Load.
- ETL provides a method of moving the data from various sources into a data warehouse.
- In the first step extraction, data is extracted from the source system into the staging area.
- In the transformation step, the data extracted from source is cleansed and transformed.
- Loading data into the target datawarehouse is the last step of the ETL process.

What is ELT?

ELT is a different method of looking at the tool approach to data movement. Instead of transforming the data before it's written, ELT lets the target system to do the transformation. The data first copied to the target and then transformed in place.

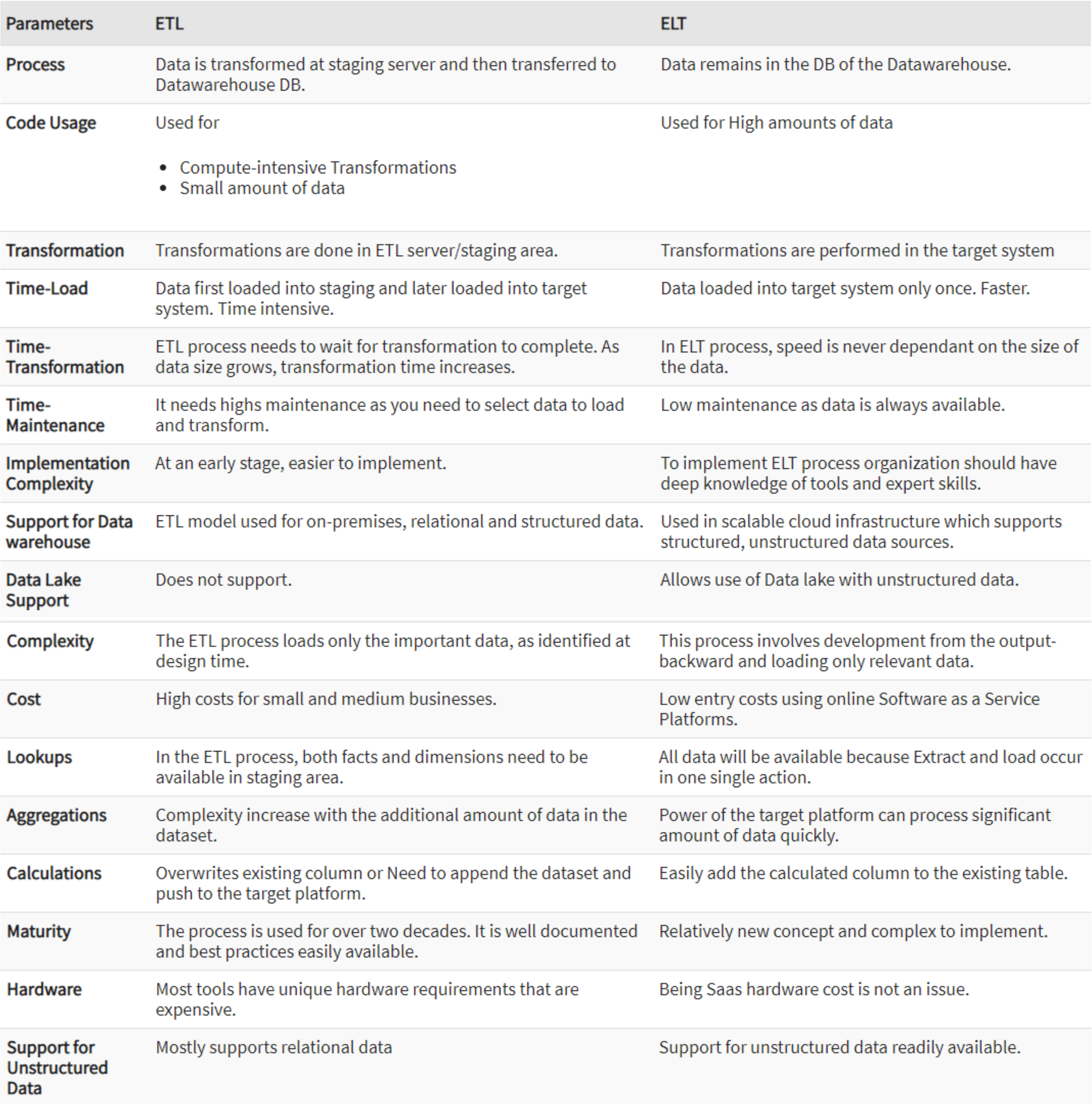
ELT usually used with no-Sql databases like Hadoop cluster, data appliance or cloud installation.



KEY DIFFERENCE

- ETL stands for Extract, Transform and Load while ELT stands for Extract, Load, Transform.
- ETL loads data first into the staging server and then into the target system whereas ELT loads data directly into the target system.
- ETL model is used for on-premises, relational and structured data while ELT is used for scalable cloud structured and unstructured data sources.
- ETL is mainly used for a small amount of data whereas ELT is used for large amounts of data.
- ETL doesn't provide data lake supports while ELT provides data lake support.
- ETL is easy to implement whereas ELT requires niche skills to implement and maintain.

ETL vs ELT



Parameters	ETL	ELT
Process	Data is transformed at staging server and then transferred to Datawarehouse DB.	Data remains in the DB of the Datawarehouse.
Code Usage	Used for • Compute-intensive Transformations • Small amount of data	Used for High amounts of data
Transformation	Transformations are done in ETL server/staging area.	Transformations are performed in the target system
Time-Load	Data first loaded into staging and later loaded into target system. Time intensive.	Data loaded into target system only once. Faster.
Time-Transformation	ETL process needs to wait for transformation to complete. As data size grows, transformation time increases.	In ELT process, speed is never dependant on the size of the data.
Time-Maintenance	It needs highs maintenance as you need to select data to load and transform.	Low maintenance as data is always available.
Implementation Complexity	At an early stage, easier to implement.	To implement ELT process organization should have deep knowledge of tools and expert skills.
Support for Data warehouse	ETL model used for on-premises, relational and structured data.	Used in scalable cloud infrastructure which supports structured, unstructured data sources.
Data Lake Support	Does not support.	Allows use of Data lake with unstructured data.
Complexity	The ETL process loads only the important data, as identified at design time.	This process involves development from the output-backward and loading only relevant data.
Cost	High costs for small and medium businesses.	Low entry costs using online Software as a Service Platforms.
Lookups	In the ETL process, both facts and dimensions need to be available in staging area.	All data will be available because Extract and load occur in one single action.
Aggregations	Complexity increase with the additional amount of data in the dataset.	Power of the target platform can process significant amount of data quickly.
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.
Maturity	The process is used for over two decades. It is well documented and best practices easily available.	Relatively new concept and complex to implement.
Hardware	Most tools have unique hardware requirements that are expensive.	Being Saas hardware cost is not an issue.
Support for Unstructured Data	Mostly supports relational data	Support for unstructured data readily available.

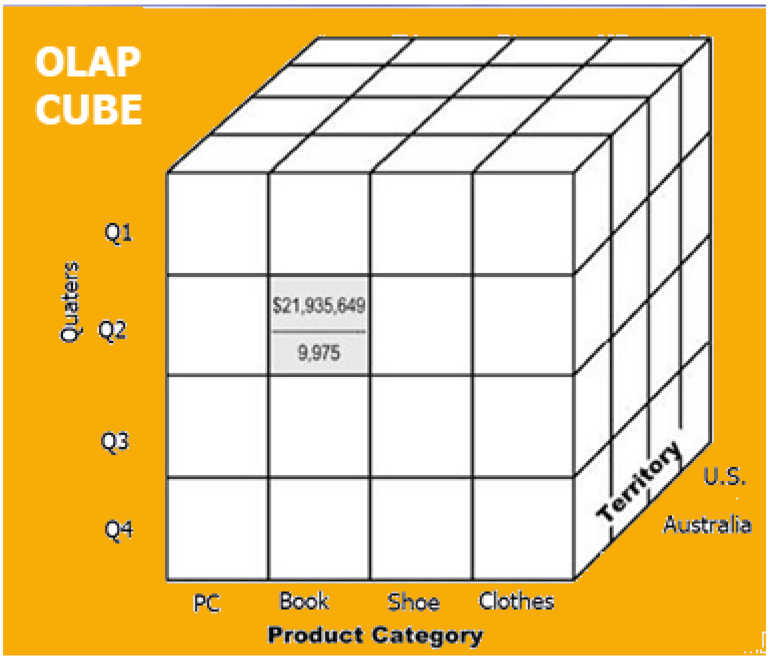
What is OLAP?

Online Analytical Processing (OLAP) is a category of software that allows users to analyze information from multiple database systems at the same time. It is a technology that enables analysts to extract and view business data from different points of view.

Analysts frequently need to group, aggregate and join data. These operations in relational databases are resource intensive. With OLAP data can be pre-calculated and pre-aggregated, making analysis faster.

OLAP databases are divided into one or more cubes. The cubes are designed in such a way that creating and viewing reports become easy. OLAP stands for Online Analytical Processing.

OLAP Cube:



At the core of the OLAP concept, is an OLAP Cube. The OLAP cube is a data structure optimized for very quick data analysis.

The OLAP Cube consists of numeric facts called measures which are categorized by dimensions. OLAP Cube is also called the **hypercube**.

Usually, data operations and analysis are performed using the simple spreadsheet, where data values are arranged in row and column format. This is ideal for two-dimensional data. However, OLAP contains multidimensional data, with data usually obtained from a different and unrelated source. Using a spreadsheet is not an optimal option. The cube can store and analyze multidimensional data in a logical and orderly manner.

How does it work?

A Data warehouse would extract information from multiple data sources and formats like text files, excel sheet, multimedia files, etc.

The extracted data is cleaned and transformed. Data is loaded into an OLAP server (or OLAP cube) where information is pre-calculated in advance for further analysis.

Basic analytical operations of OLAP

Four types of analytical operations in OLAP are:

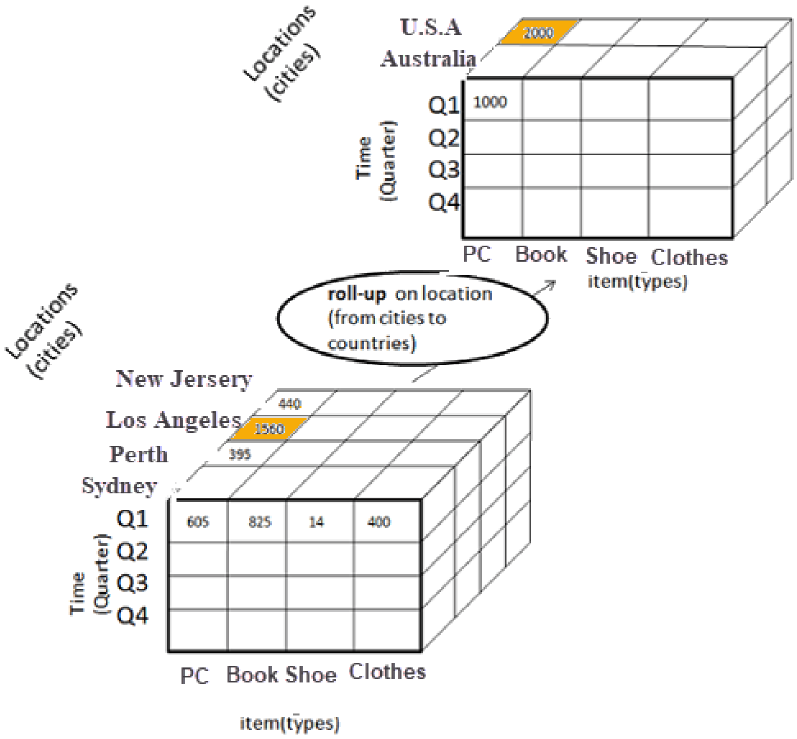
1. Roll-up
2. Drill-down
3. Slice and dice
4. Pivot (rotate)

1) Roll-up:

Roll-up is also known as "consolidation" or "aggregation." The Roll-up operation can be performed in 2 ways

1. Reducing dimensions
2. Climbing up concept hierarchy. Concept hierarchy is a system of grouping things based on their order or level.

Consider the following diagram

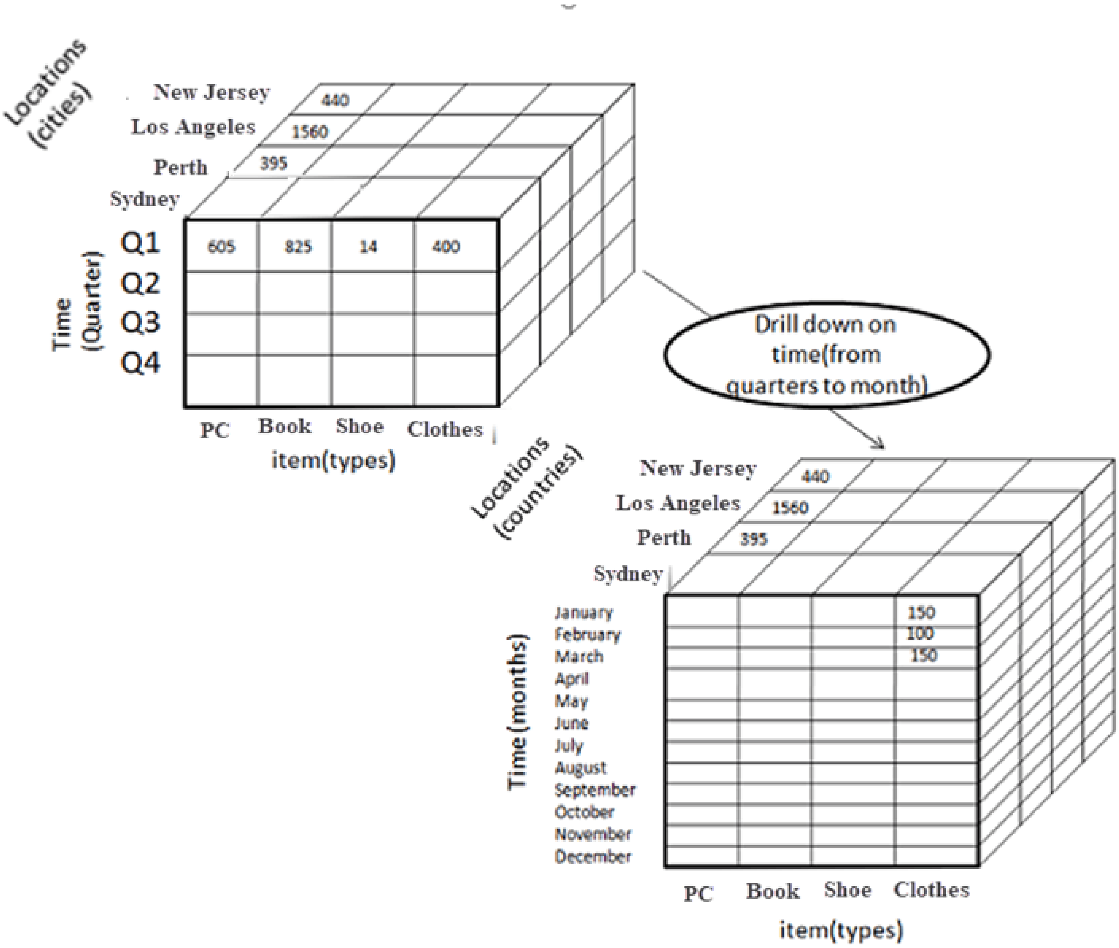


- In this example, cities New jersey and Lost Angles and rolled up into country USA
- The sales figure of New Jersey and Los Angeles are 440 and 1560 respectively. They become 2000 after roll-up
- In this aggregation process, data is location hierarchy moves up from city to the country.
- In the roll-up process at least one or more dimensions need to be removed. In this example, Quater dimension is removed.

2) Drill-down

In drill-down data is fragmented into smaller parts. It is the opposite of the rollup process. It can be done via

- Moving down the concept hierarchy
- Increasing a dimension



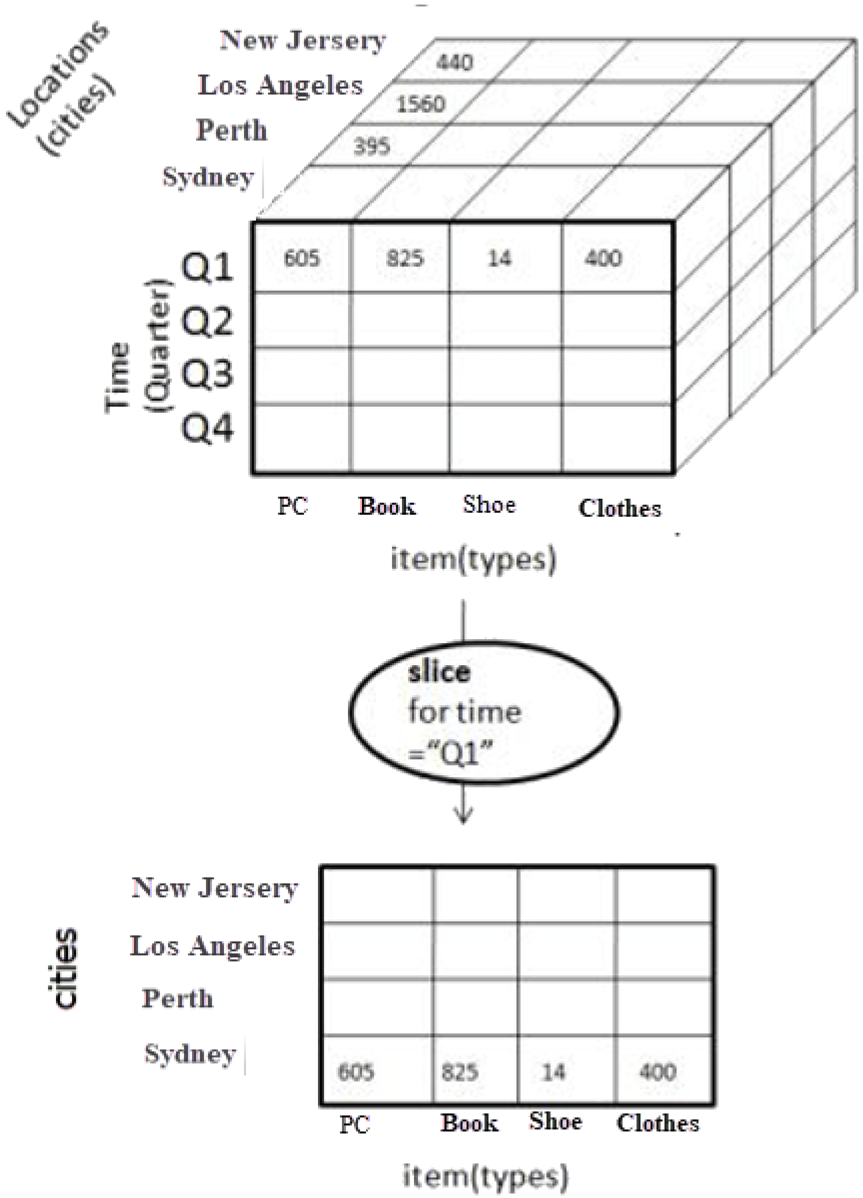
Consider the diagram above

- Quater Q1 is drilled down to months January, February, and March. Corresponding sales are also registers.
- In this example, dimension months are added.

3) Slice:

Here, one dimension is selected, and a new sub-cube is created.

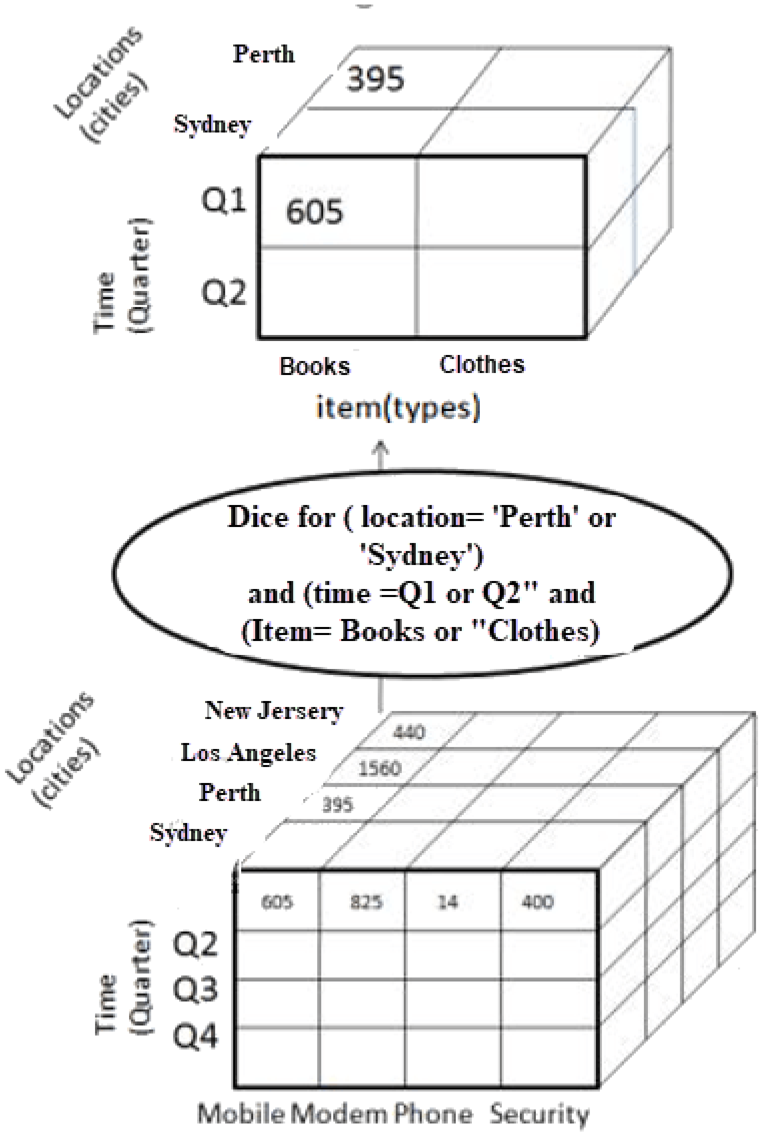
Following diagram explain how slice operation performed:



- Dimension Time is Sliced with Q1 as the filter.
- A new cube is created altogether.

Dice:

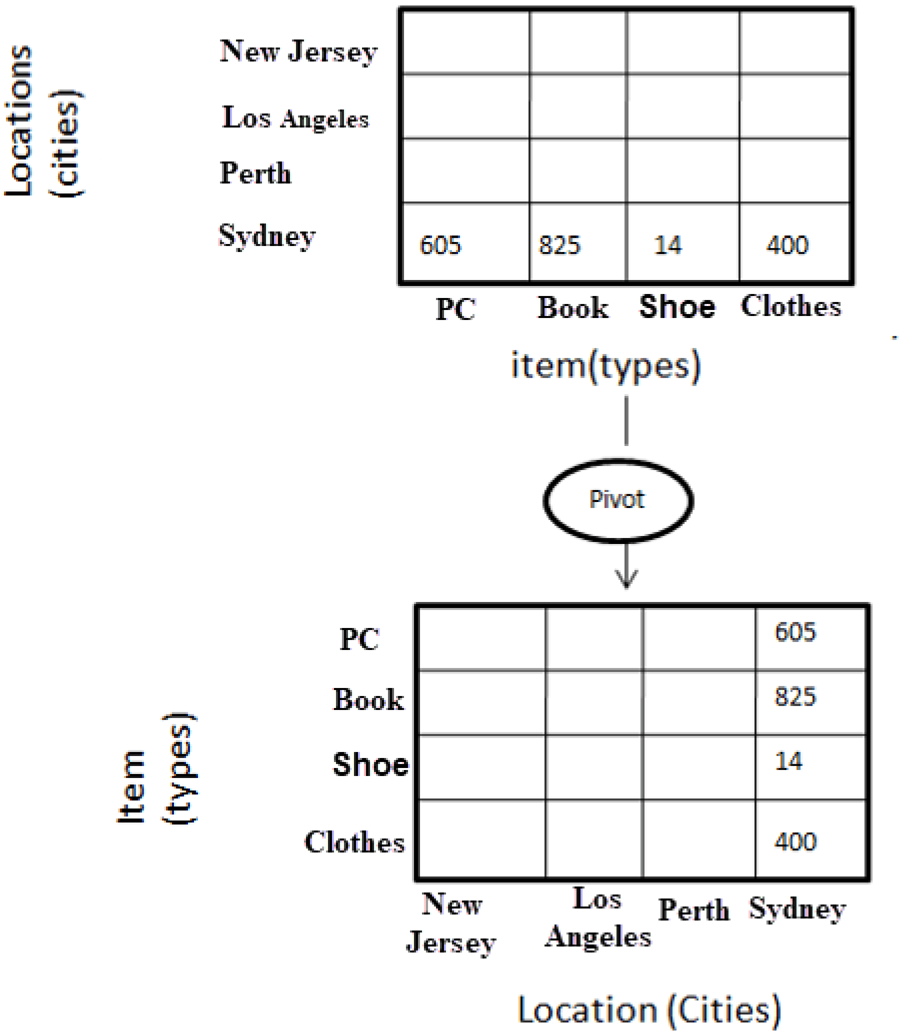
This operation is similar to a slice. The difference in dice is you select 2 or more dimensions that result in the creation of a sub-cube.



4) Pivot

In Pivot, you rotate the data axes to provide a substitute presentation of data.

In the following example, the pivot is based on item types.

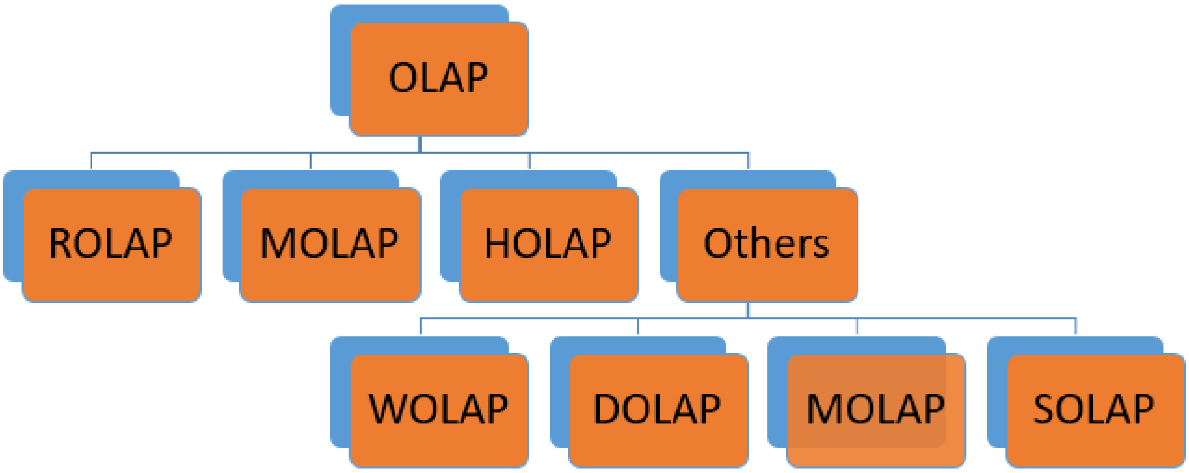


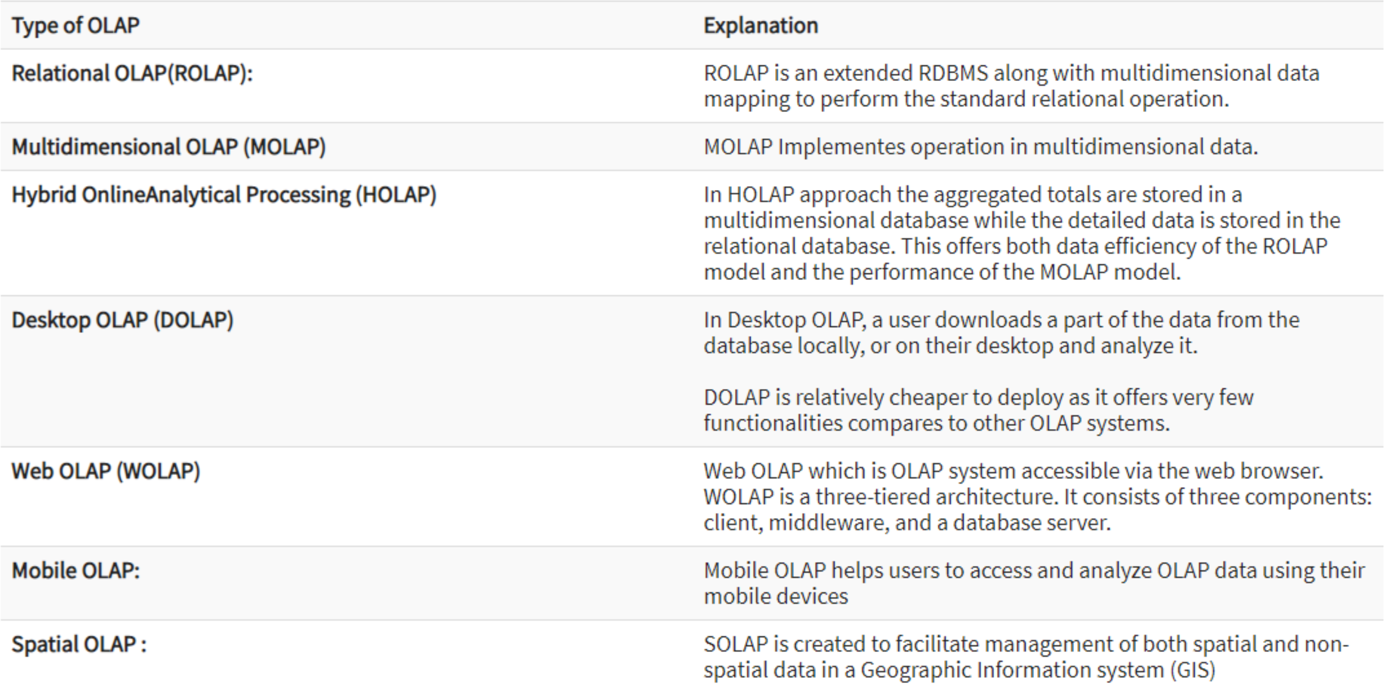
	PC	Book	Shoe	Clothes
New Jersey				
Los Angeles				
Perth				
Sydney	605	825	14	400

	New Jersey	Los Angeles	Perth	Sydney
PC				605
Book				825
Shoe				14
Clothes				400

Types of OLAP systems

OLAP Hierarchical Structure





Type of OLAP	Explanation
Relational OLAP(ROLAP):	ROLAP is an extended RDBMS along with multidimensional data mapping to perform the standard relational operation.
Multidimensional OLAP (MOLAP)	MOLAP Implementes operation in multidimensional data.
Hybrid OnlineAnalytical Processing (HOLAP)	In HOLAP approach the aggregated totals are stored in a multidimensional database while the detailed data is stored in the relational database. This offers both data efficiency of the ROLAP model and the performance of the MOLAP model.
Desktop OLAP (DOLAP)	In Desktop OLAP, a user downloads a part of the data from the database locally, or on their desktop and analyze it. DOLAP is relatively cheaper to deploy as it offers very few functionalities compares to other OLAP systems.
Web OLAP (WOLAP)	Web OLAP which is OLAP system accessible via the web browser. WOLAP is a three-tiered architecture. It consists of three components: client, middleware, and a database server.
Mobile OLAP:	Mobile OLAP helps users to access and analyze OLAP data using their mobile devices
Spatial OLAP :	SOLAP is created to facilitate management of both spatial and non-spatial data in a Geographic Information system (GIS)

ROLAP

ROLAP works with data that exist in a relational database. Facts and dimension tables are stored as relational tables. It also allows multidimensional analysis of data and is the fastest growing OLAP.

Advantages of ROLAP model:

- **High data efficiency.** It offers high data efficiency because query performance and access language are optimized particularly for the multidimensional data analysis.
- **Scalability.** This type of OLAP system offers scalability for managing large volumes of data, and even when the data is steadily increasing.

Drawbacks of ROLAP model:

- **Demand for higher resources:** ROLAP needs high utilization of manpower, software, and hardware resources.
- **Aggregately data limitations.** ROLAP tools use SQL for all calculation of aggregate data. However, there are no set limits to the for handling computations.
- **Slow query performance.** Query performance in this model is slow when compared with MOLAP

MOLAP

MOLAP uses array-based multidimensional storage engines to display multidimensional views of data. Basically, they use an OLAP cube.

Learn more about OLAP here

Hybrid OLAP

Hybrid OLAP is a mixture of both ROLAP and MOLAP. It offers fast computation of MOLAP and higher scalability of ROLAP. HOLAP uses two databases.

1. Aggregated or computed data is stored in a multidimensional OLAP cube
2. Detailed information is stored in a relational database.

Benefits of Hybrid OLAP:

- This kind of OLAP helps to economize the disk space, and it also remains compact which helps to avoid issues related to access speed and convenience.
- Hybrid HOLAP's uses cube technology which allows faster performance for all types of data.
- ROLAP are instantly updated and HOLAP users have access to this real-time instantly updated data. MOLAP brings cleaning and conversion of data thereby improving data relevance. This brings best of both worlds.

Drawbacks of Hybrid OLAP:

- Greater complexity level**:** The major drawback in HOLAP systems is that it supports both ROLAP and MOLAP tools and applications. Thus, it is very complicated.
- Potential overlaps**:** There are higher chances of overlapping especially into their functionalities.

Advantages of OLAP

- OLAP is a platform for all type of business includes planning, budgeting, reporting, and analysis.
- Information and calculations are consistent in an OLAP cube. This is a crucial benefit.
- Quickly create and analyze "What if" scenarios
- Easily search OLAP database for broad or specific terms.
- OLAP provides the building blocks for business modeling tools, Data mining tools, performance reporting tools.
- Allows users to do slice and dice cube data all by various dimensions, measures, and filters.
- It is good for analyzing time series.
- Finding some clusters and outliers is easy with OLAP.
- It is a powerful visualization online analytical process system which provides faster response times

Disadvantages of OLAP

- OLAP requires organizing data into a star or snowflake schema. These schemas are complicated to implement and administer
- You cannot have large number of dimensions in a single OLAP cube
- Transactional data cannot be accessed with OLAP system.
- Any modification in an OLAP cube needs a full update of the cube. This is a time-consuming process

Summary:

- OLAP is a technology that enables analysts to extract and view business data from different points of view.
- At the core of the OLAP concept, is an OLAP Cube.
- Various business applications and other data operations require the use of OLAP Cube.
- There are primary five types of analytical operations in OLAP 1) Roll-up 2) Drill-down 3) Slice 4) Dice and 5) Pivot
- Three types of widely used OLAP systems are MOLAP, ROLAP, and Hybrid OLAP.
- Desktop OLAP, Web OLAP, and Mobile OLAP are some other types of OLAP systems.

What is MOLAP?

Multidimensional OLAP (MOLAP) is a classical OLAP that facilitates data analysis by using a multidimensional data cube. Data is pre-computed, re-summarized, and stored in a MOLAP (a major difference from ROLAP). Using a MOLAP, a user can use multidimensional view data with different facets.

Multidimensional data analysis is also possible if a relational database is used. By that would require querying data from multiple tables. On the contrary, MOLAP has all possible combinations of data already stored in a multidimensional array. MOLAP can access this data directly. Hence, MOLAP is faster compared to Relational Online Analytical Processing (ROLAP).

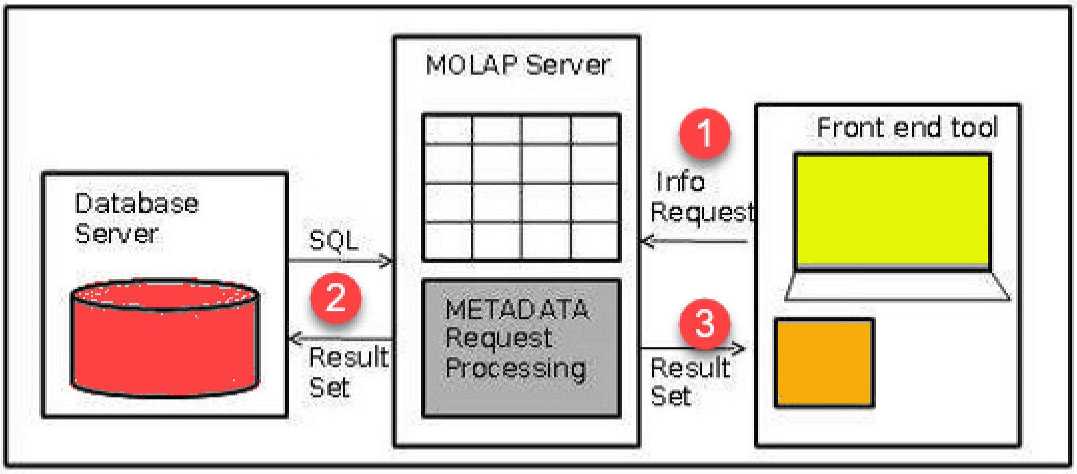
Key Points

- In MOLAP, operations are called processing.
- MOLAP tools process information with the same amount of response time irrespective of the level of summarizing.
- MOLAP tools remove complexities of designing a relational database to store data for analysis.
- MOLAP server implements two level of storage representation to manage dense and sparse data sets.
- The storage utilization can be low if the data set is sparse.
- Facts are stored in multi-dimensional array and dimensions used to query them.

MOLAP Architecture

MOLAP Architecture includes the following components

- Database server.
- MOLAP server.
- Front-end tool.



Consider above gien MOLAP Architectures:

1. The user request reports through the interface
2. The application logic layer of the MDDB retrieves the stored data from Database
3. The application logic layer forwards the result to the client/user.

MOLAP architecture mainly reads the precompiled data. MOLAP architecture has limited capabilities to dynamically create aggregations or to calculate results that have not been pre-calculated and stored.

For example, an accounting head can run a report showing the corporate P/L account or P/L account for a specific subsidiary. The MDDB would retrieve precompiled Profit & Loss figures and display that result to the user.

Implementation considerations is MOLAP

- In MOLAP it's essential to consider both maintenance and storage implications to creating strategy for building cubes.
- Proprietary languages used to query MOLAP. However, it involves extensive click and drag support for example MDX by Microsoft.
- Difficult to scale because the number and size of cubes required when dimensions increase.
- API's should provide for probing the cubes.
- Data structure to support multiple subject areas of data analyses which data can be navigated and analyzed. When the navigation changes, the data structure needs to be physically reorganized.

- Need different skill set and tools for Database administrator to build, maintain the database.

MOLAP Advantages

- MOLAP can manage, analyze and store considerable amounts of multidimensional data.
- Fast Query Performance due to optimized storage, indexing, and caching.
- Smaller sizes of data as compared to the relational database.
- Automated computation of higher level of aggregates data.
- Help users to analyze larger, less-defined data.
- MOLAP is easier to the user that's why It is a suitable model for inexperienced users.
- MOLAP cubes are built for fast data retrieval and are optimal for slicing and dicing operations.
- All calculations are pre-generated when the cube is created.

MOLAP Disadvantages

- One major weakness of MOLAP is that it is less scalable than ROLAP as it handles only a limited amount of data.
- The MOLAP also introduces data redundancy as it is resource intensive
- MOLAP Solutions may be lengthy, particularly on large data volumes.
- MOLAP products may face issues while updating and querying models when dimensions are more than ten.
- MOLAP is not capable of containing detailed data.
- The storage utilization can be low if the data set is highly scattered.
- It can handle the only limited amount of data therefore, it's impossible to include a large amount of data in the cube itself.

MOLAP Tools

- Essbase - Tools from Oracle that has a multidimensional database.

- Express Server - Web-based environment that runs on Oracle database.

- Yellowfin - Business analytics tools for creating reports and dashboards.

- Clear Analytics - Clear analytics is an Excel-based business solution.

- SAP Business Intelligence - Business analytics solutions from SAP

Summary:

- Multidimensional OLAP (MOLAP) is a classical OLAP that facilitates data analysis by using a multidimensional data cube.
- MOLAP tools process information with the same amount of response time irrespective of the level of summarizing.
- MOLAP server implements two level of storage to manage dense and sparse data sets.
- MOLAP can manage, analyze, and store considerable amounts of multidimensional data.
- It helps to automate computation of higher level of aggregates data
- It is less scalable than ROLAP as it handles only a limited amount of data.

What is OLTP? Definition, Architecture, Example

What is OLTP?

OLTP is an operational system that supports transaction-oriented applications in a 3-tier architecture. It administers the day to day transaction of an organization. OLTP is basically focused on query processing, maintaining data integrity in multi-access environments as well as effectiveness that is measured by the total number of transactions per second. The full form of OLTP is Online Transaction Processing.

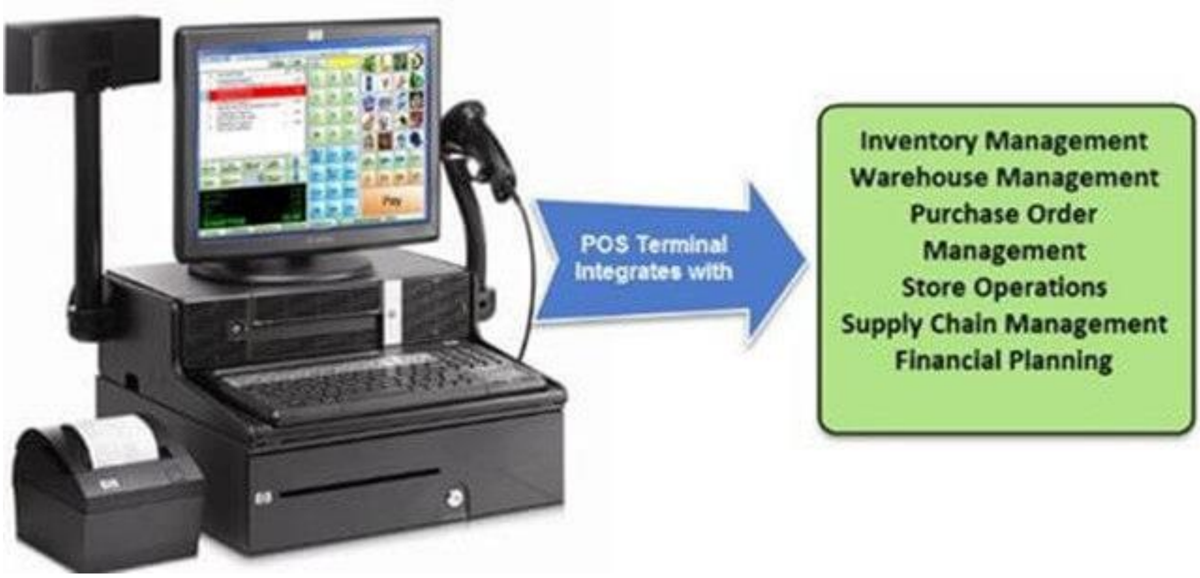
Characteristics of OLTP

Following are important characteristics of OLTP:

- OLTP uses transactions that include small amounts of data.
- Indexed data in the database can be accessed easily.
- OLTP has a large number of users.
- It has fast response times
- Databases are directly accessible to end-users
- OLTP uses a fully normalized schema for database consistency.
- The response time of OLTP system is short.
- It strictly performs only the predefined operations on a small number of records.
- OLTP stores the records of the last few days or a week.
- It supports complex data models and tables.

Type of queries that an OLTP system can Process:

OLTP system is an online database changing system. Therefore, it supports database query such as insert, update, and delete information from the database.



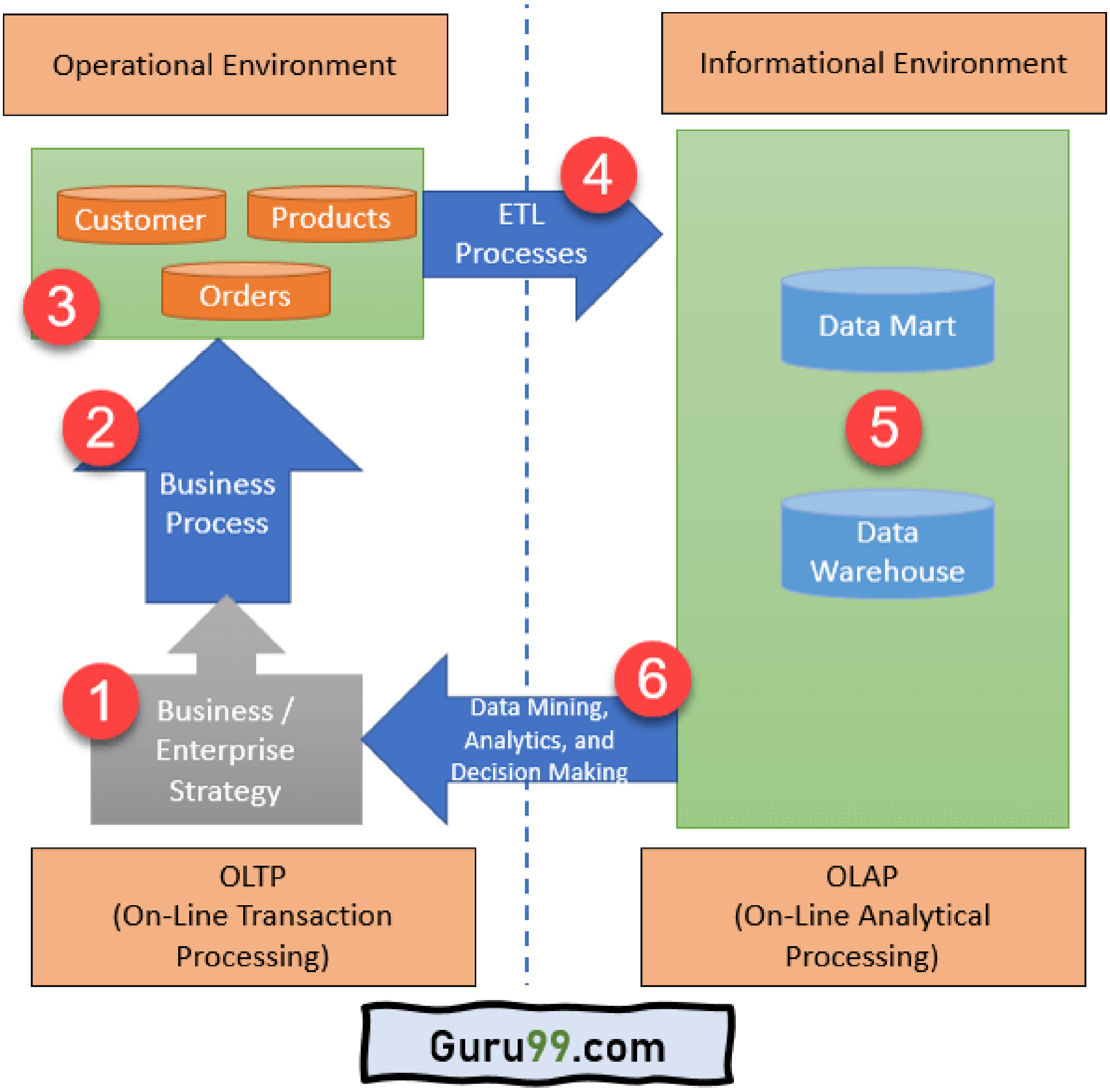
POS system for OLTP

Consider a point of sale system of a supermarket, following are the sample queries that this system can process:

- Retrieving the description of a particular product.
- Filtering all products related to the supplier.
- Searching the record of the customer.
- Listing products having a price less than the expected amount.

Architecture of OLTP

Here is the architecture of OLTP:


Operational Environment
Informational Environment
Customer
Products
Orders
3
ETL Processes
4
Data Mart
5
Data Warehouse
2
Business Process
1
Business / Enterprise Strategy
Data Mining, Analytics, and Decision Making
6
OLTP (On-Line Transaction Processing)
OLAP (On-Line Analytical Processing)
Guru99.com

OLTP Architecture

1. **Business / Enterprise Strategy:** Enterprise strategy deals with the issues that affect the organization as a whole. In OLTP, it is typically developed at a high level within the firm, by the board of directors or the top management
2. **Business Process:** OLTP business process is a set of activities and tasks that, once completed, will accomplish an organizational goal.
3. **Customers, Orders, and Products:** OLTP database store information about products, orders (transactions), customers (buyers), suppliers (sellers), and employees.
4. **ETL Processes:** It separates the data from various RDBMS source systems, then transforms the data (like applying concatenations, calculations, etc.) and loads the processed data into the Data Warehouse system.

5. **Data Mart and Data warehouse:** A data mart is a structure/access pattern specific to data warehouse environments. It is used by OLAP to store processed data.
6. **Data Mining, Analytics, and Decision Making:** Data stored in the data mart and data warehouse can be used for data mining, analytics, and decision making.

   This data helps you to discover data patterns, analyze raw data, and make analytical decisions for your organization's growth.

Example of OLTP Transaction

An example of the OLTP system is the ATM center. Assume that a couple has a joint account with a bank. One day both simultaneously reach different ATM centers at precisely the same time and want to withdraw the total amount present in their bank account.

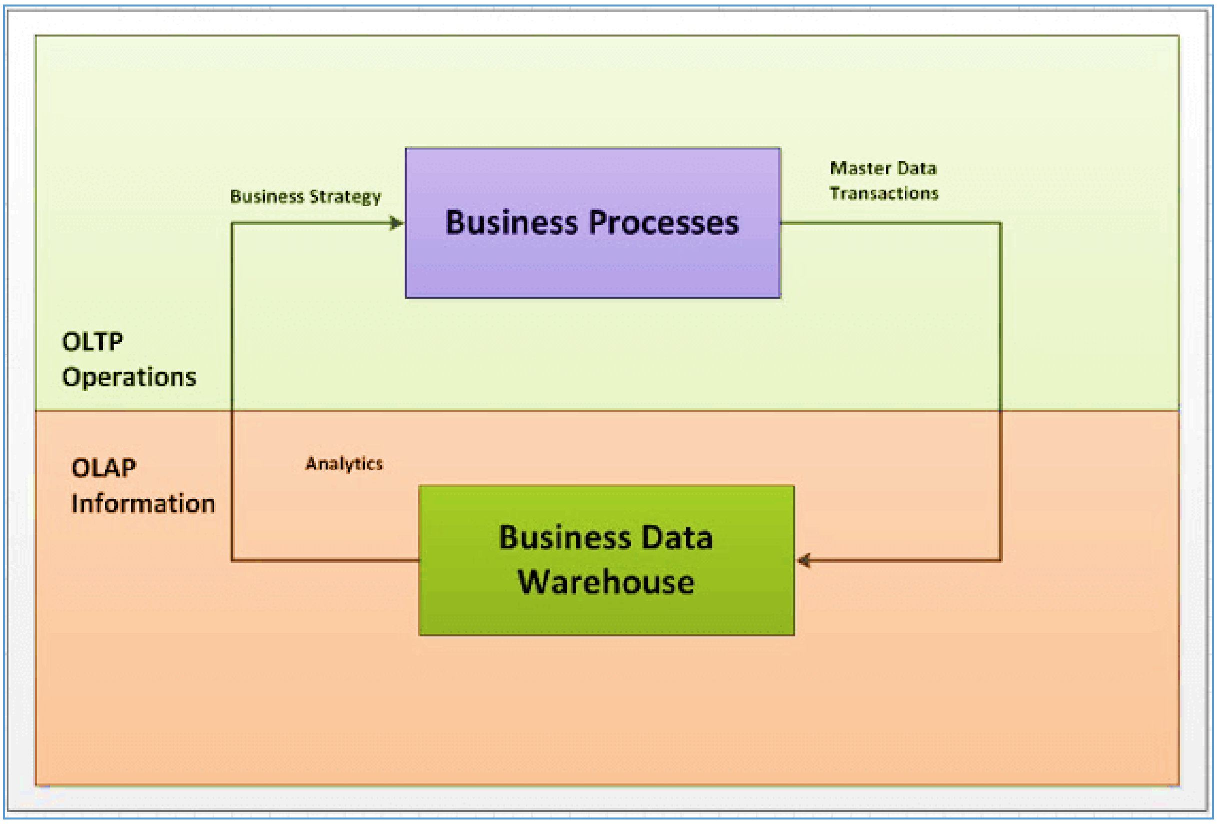
OLTOP for ATM image

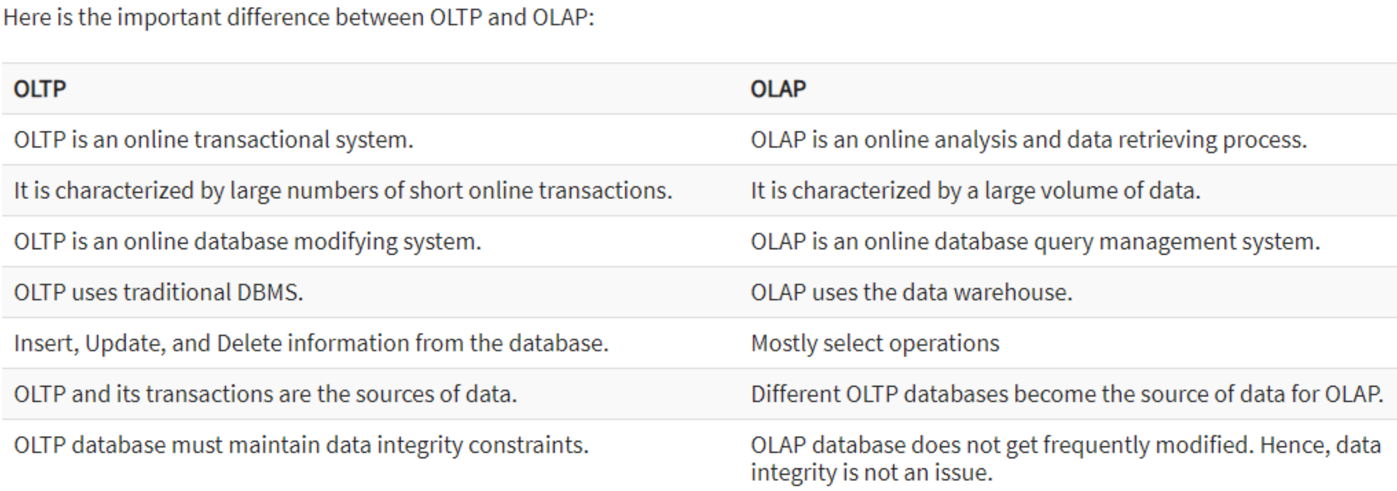
However, the person that completes the authentication process first will be able to get money. In this case, the OLTP system makes sure that the withdrawn amount will be never more than the amount present in the bank. The key to note here is that OLTP systems are optimized for transactional superiority instead of data analysis.

Other examples of OLTP system are:

- Online banking
- Online airline ticket booking
- Sending a text message
- Order entry
- Add a book to shopping cart

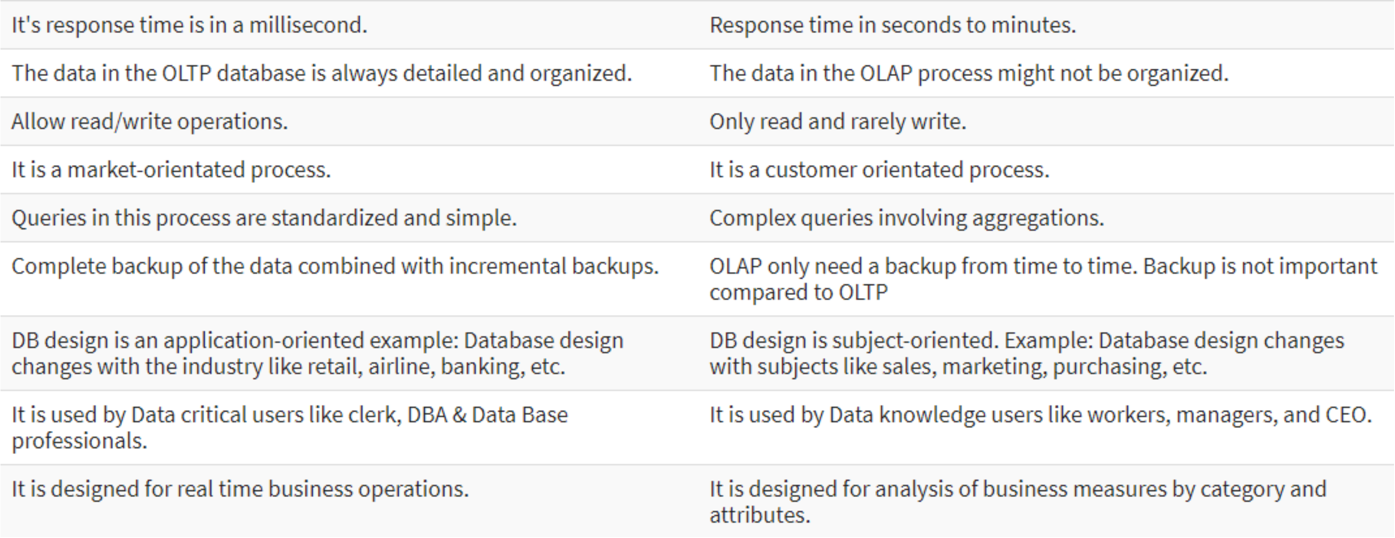
OLTP vs. OLAP

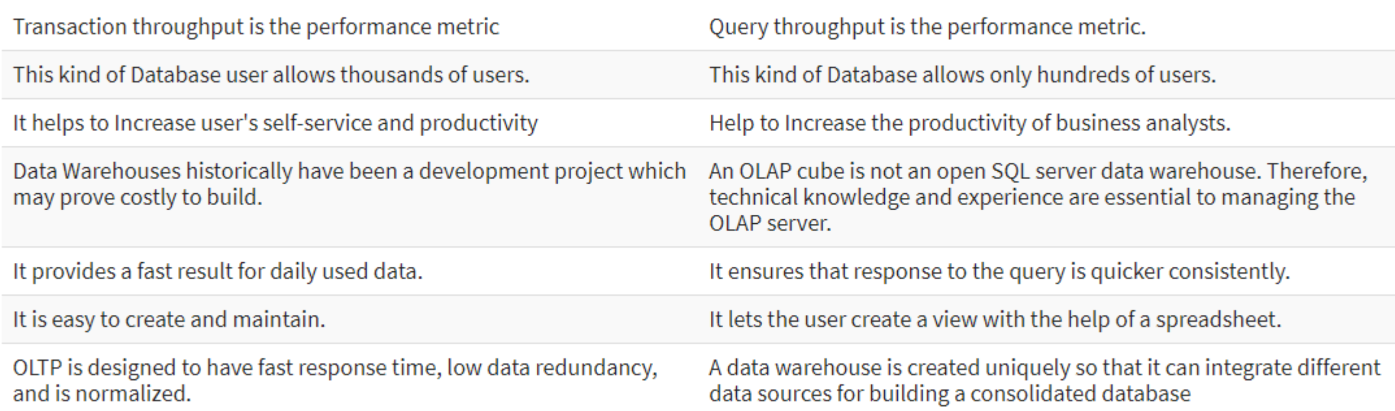




Here is the important difference between OLTP and OLAP:

OLTP	OLAP
OLTP is an online transactional system.	OLAP is an online analysis and data retrieving process.
It is characterized by large numbers of short online transactions.	It is characterized by a large volume of data.
OLTP is an online database modifying system.	OLAP is an online database query management system.
OLTP uses traditional DBMS.	OLAP uses the data warehouse.
Insert, Update, and Delete information from the database.	Mostly select operations
OLTP and its transactions are the sources of data.	Different OLTP databases become the source of data for OLAP.
OLTP database must maintain data integrity constraints.	OLAP database does not get frequently modified. Hence, data integrity is not an issue.





It's response time is in a millisecond.	Response time in seconds to minutes.
The data in the OLTP database is always detailed and organized.	The data in the OLAP process might not be organized.
Allow read/write operations.	Only read and rarely write.
It is a market-orientated process.	It is a customer orientated process.
Queries in this process are standardized and simple.	Complex queries involving aggregations.
Complete backup of the data combined with incremental backups.	OLAP only need a backup from time to time. Backup is not important compared to OLTP
DB design is an application-oriented example: Database design changes with the industry like retail, airline, banking, etc.	DB design is subject-oriented. Example: Database design changes with subjects like sales, marketing, purchasing, etc.
It is used by Data critical users like clerk, DBA & Data Base professionals.	It is used by Data knowledge users like workers, managers, and CEO.
It is designed for real time business operations.	It is designed for analysis of business measures by category and attributes.
Transaction throughput is the performance metric	Query throughput is the performance metric.
This kind of Database user allows thousands of users.	This kind of Database allows only hundreds of users.
It helps to Increase user's self-service and productivity	Help to Increase the productivity of business analysts.
Data Warehouses historically have been a development project which may prove costly to build.	An OLAP cube is not an open SQL server data warehouse. Therefore, technical knowledge and experience are essential to managing the OLAP server.
It provides a fast result for daily used data.	It ensures that response to the query is quicker consistently.
It is easy to create and maintain.	It lets the user create a view with the help of a spreadsheet.
OLTP is designed to have fast response time, low data redundancy, and is normalized.	A data warehouse is created uniquely so that it can integrate different data sources for building a consolidated database

Advantages of OLTP:

Following are the pros/benefits of OLTP system:

- OLTP offers accurate forecast for revenue and expense.
- It provides a solid foundation for a stable business /organization due to timely modification of all transactions.
- OLTP makes transactions much easier on behalf of the customers.
- It broadens the client base for an organization by speeding up and simplifying individual processes.
- OLTP provides support for bigger databases.
- Partition of data for data manipulation is easy.
- We need OLTP to use the tasks which are frequently performed by the system.
- When we need only a small number of records.
- The tasks that include insertion, updation, or deletion of data.
- It is used when you need consistency and concurrency in order to perform tasks that ensure its greater availability.

Disadvantages of OLTP

Here are cons/drawbacks of OLTP system:

- If the OLTP system faces hardware failures, then online transactions get severely affected.
- OLTP systems allow multiple users to access and change the same data at the same time, which many times created an unprecedented situation.
- If the server hangs for seconds, it can affect to a large number of transactions.
- OLTP required a lot of staff working in groups in order to maintain inventory.
- Online Transaction Processing Systems do not have proper methods of transferring products to buyers by themselves.
- OLTP makes the database much more susceptible to hackers and intruders.
- In B2B transactions, there are chances that both buyers and suppliers miss out efficiency advantages that the system offers.
- Server failure may lead to wiping out large amounts of data from the database.
- You can perform a limited number of queries and updates.

Challenges of an OLTP System

- It allows more than one user to access and change the same data simultaneously. Therefore, it requires concurrency control and recovery technique in order to avoid any unprecedented situations
- OLTP system data are not suitable for decision making. You have to use data of OLAP systems for "what if" analysis or the decision making.

Summary

- OLTP is defined as an operational system that supports transaction-oriented applications in a 3-tier architecture.
- OLTP uses transactions that include small amounts of data.
- OLTP system is an online database changing system.
- The architecture of OLTP contains

  1) Business / Enterprise Strategy,

  2) Business Process,

  3) Customers, Orders, and Products,

  4) ETL Processes,

  5) Data Mart and Data warehouse, and

  6) Data Mining, Analytics, and Decision Making.

- OLTP is an online transactional system, whereas OLAP is an online analysis and data retrieving process.
- OLTP provides a solid foundation for a stable business /organization due to timely modification of all transactions.
- OLTP systems allow multiple users to access and change the same data at the same time, which many times created an unprecedented

What is Dimensional Modeling in Data Warehouse?

What is Dimensional Modeling?

DIMENSIONAL MODELING (DM) is a data structure technique optimized for data storage in a Data warehouse. The purpose of dimensional model is to optimize the database for fast retrieval of data. The concept of Dimensional Modelling was developed by Ralph Kimball and consists of "fact" and "dimension" tables.

A Dimensional model is designed to read, summarize, analyze numeric information like values, balances, counts, weights, etc. in a data warehouse. In contrast, relation models are optimized for addition, updating and deletion of data in a real-time Online Transaction System.

These dimensional and relational models have their unique way of data storage that has specific advantages.

For instance, in the relational mode, normalization and ER models reduce redundancy in data. On the contrary, dimensional model arranges data in such a way that it is easier to retrieve information and generate reports.

Hence, Dimensional models are used in data warehouse systems and not a good fit for relational systems.

Elements of Dimensional Data Model

Fact

Facts are the measurements/metrics or facts from your business process. For a Sales business process, a measurement would be quarterly sales number

Dimension

Dimension provides the context surrounding a business process event. In simple terms, they give who, what, where of a fact. In the Sales business process, for the fact quarterly sales number, dimensions would be

- Who – Customer Names
- Where – Location
- What – Product Name

In other words, a dimension is a window to view information in the facts.

Attributes

The Attributes are the various characteristics of the dimension.

In the Location dimension, the attributes can be

- State
- Country
- Zipcode etc.

Attributes are used to search, filter, or classify facts. Dimension Tables contain Attributes

Fact Table

A fact table is a primary table in a dimensional model.

A Fact Table contains

1. Measurements/facts
2. Foreign key to dimension table

Dimension table

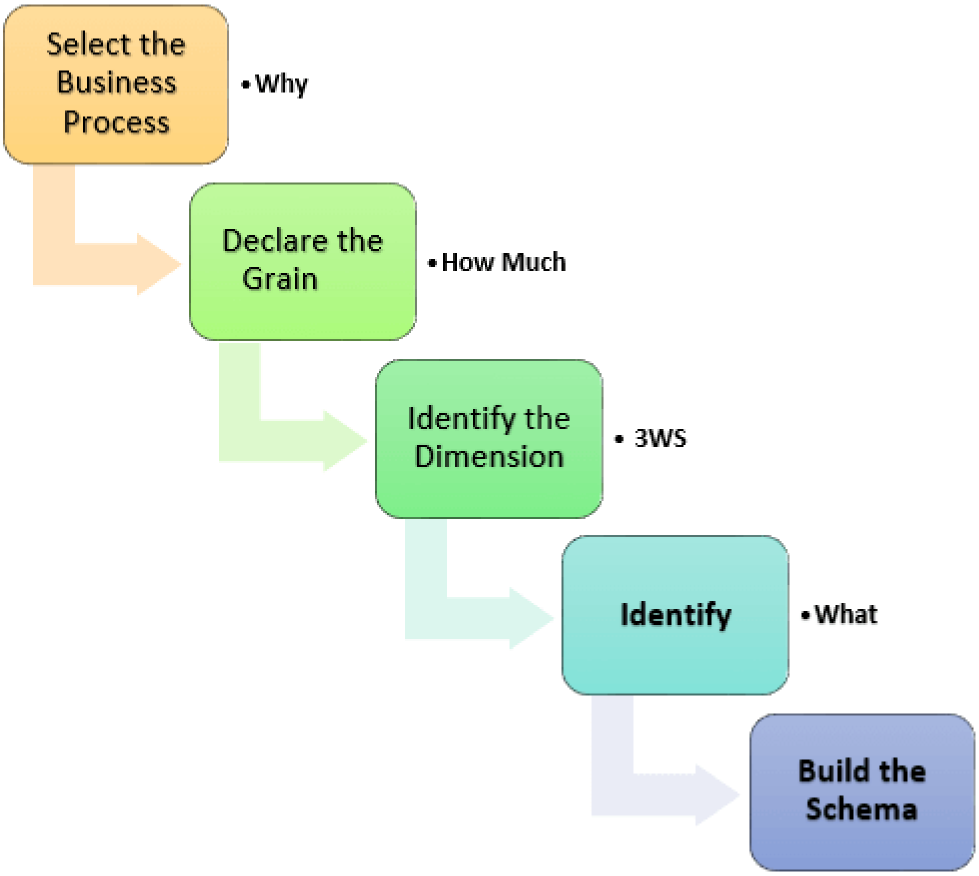
- A dimension table contains dimensions of a fact.
- They are joined to fact table via a foreign key.
- Dimension tables are de-normalized tables.
- The Dimension Attributes are the various columns in a dimension table
- Dimensions offers descriptive characteristics of the facts with the help of their attributes
- No set limit set for given for number of dimensions
- The dimension can also contain one or more hierarchical relationships

Steps of Dimensional Modelling

The accuracy in creating your Dimensional modeling determines the success of your data warehouse implementation. Here are the steps to create Dimension Model

1. Identify Business Process
2. Identify Grain (level of detail)
3. Identify Dimensions
4. Identify Facts
5. Build Star

The model should describe the Why, How much, When/Where/Who and What of your business process



Step 1) Identify the business process

Identifying the actual business process a datarehouse should cover. This could be Marketing, Sales, HR, etc. as per the data analysis needs of the organization. The selection of the Business process also depends on the quality of data available for that process. It is the most important step of the Data Modelling process, and a failure here would have cascading and irreparable defects.

To describe the business process, you can use plain text or use basic Business Process Modelling Notation (BPMN) or Unified Modelling Language (UML).

Step 2) Identify the grain

The Grain describes the level of detail for the business problem/solution. It is the process of identifying the lowest level of information for any table in your data warehouse. If a table contains sales data for every day, then it should be daily granularity. If a table contains total sales data for each month, then it has monthly granularity.

During this stage, you answer questions like

1. Do we need to store all the available products or just a few types of products? This decision is based on the business processes selected for Datawarehouse
2. Do we store the product sale information on a monthly, weekly, daily or hourly basis? This decision depends on the nature of reports requested by executives
3. How do the above two choices affect the database size?

Example of Grain:

The CEO at an MNC wants to find the sales for specific products in different locations on a daily basis.

So, the grain is "product sale information by location by the day."

Step 3) Identify the dimensions

Dimensions are nouns like date, store, inventory, etc. These dimensions are where all the data should be stored. For example, the date dimension may contain data like a year, month and weekday.

Example of Dimensions:

The CEO at an MNC wants to find the sales for specific products in different locations on a daily basis.

Dimensions: Product, Location and Time

Attributes: For Product: Product key (Foreign Key), Name, Type, Specifications

Hierarchies: For Location: Country, State, City, Street Address, Name

Step 4) Identify the Fact

This step is co-associated with the business users of the system because this is where they get access to data stored in the data warehouse. Most of the fact table rows are numerical values like price or cost per unit, etc.

Example of Facts:

The CEO at an MNC wants to find the sales for specific products in different locations on a daily basis.

The fact here is Sum of Sales by product by location by time.

Step 5) Build Schema

In this step, you implement the Dimension Model. A schema is nothing but the database structure (arrangement of tables). There are two popular schemas

1. **Star Schema**

The star schema architecture is easy to design. It is called a star schema because diagram resembles a star, with points radiating from a center. The center of the star consists of the fact table, and the points of the star is dimension tables.

The fact tables in a star schema which is third normal form whereas dimensional tables are de-normalized.

2. **Snowflake Schema**

The snowflake schema is an extension of the star schema. In a snowflake schema, each dimension are normalized and connected to more dimension tables.

Rules for Dimensional Modelling

- Load atomic data into dimensional structures.
- Build dimensional models around business processes.
- Need to ensure that every fact table has an associated date dimension table.
- Ensure that all facts in a single fact table are at the same grain or level of detail.
- It's essential to store report labels and filter domain values in dimension tables
- Need to ensure that dimension tables use a surrogate key
- Continuously balance requirements and realities to deliver business solution to support their decision-making

Benefits of dimensional modeling

- Standardization of dimensions allows easy reporting across areas of the business.
- Dimension tables store the history of the dimensional information.
- It allows to introduced entirely new dimension without major disruptions to the fact table.
- Dimensional also to store data in such a fashion that it is easier to retrieve the information from the data once the data is stored in the database.
- Compared to the normalized model dimensional table are easier to understand.
- Information is grouped into clear and simple business categories.

- The dimensional model is very understandable by the business. This model is based on business terms, so that the business knows what each fact, dimension, or attribute means.
- Dimensional models are deformalized and optimized for fast data querying. Many relational database platforms recognize this model and optimize query execution plans to aid in performance.
- Dimensional modeling creates a schema which is optimized for high performance. It means fewer joins and helps with minimized data redundancy.
- The dimensional model also helps to boost query performance. It is more denormalized therefore it is optimized for querying.
- Dimensional models can comfortably accommodate change. Dimension tables can have more columns added to them without affecting existing business intelligence applications using these tables.

Summary:

- A dimensional model is a data structure technique optimized for Data warehousing tools.
- Facts are the measurements/metrics or facts from your business process.
- Dimension provides the context surrounding a business process event.
- The Attributes are the various characteristics of the dimension.
- A fact table is a primary table in a dimensional model.
- A dimension table contains dimensions of a fact.
- There are three types of facts 1. Additive 2. Non-additive 3. Semi- additive.
- Types of Dimensions are Conformed, Outrigger, Shrunken, Role-playing, Dimension to Dimension Table, Junk, Degenerate, Swappable and Step Dimensions.
- Five steps of Dimensional modeling are 1. Identify Business Process 2. Identify Grain (level of detail) 3. Identify Dimensions 4. Identify Facts 5. Build Star
- In Dimensional modeling, there is need to ensure that every fact table has an associated date dimension table.

What is Multidimensional schema?

Multidimensional Schema is especially designed to model data warehouse systems. The schemas are designed to address the unique needs of very large databases designed for the analytical purpose (OLAP).

Types of Data Warehouse Schema:

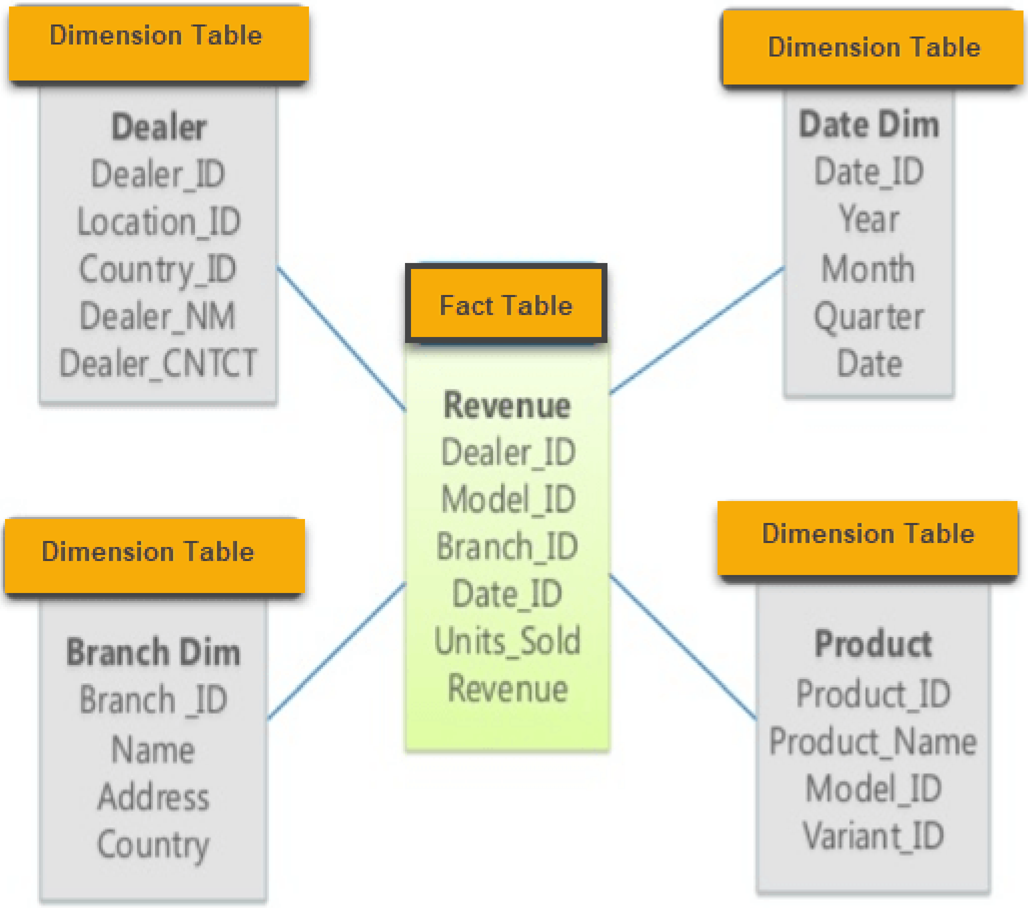
Following are 3 chief types of multidimensional schemas each having its unique advantages.

- Star Schema
- Snowflake Schema
- Galaxy Schema

What is a Star Schema?

In the **STAR Schema**, the center of the star can have one fact table and a number of associated dimension tables. It is known as star schema as its structure resembles a star. The star schema is the simplest type of Data Warehouse schema. It is also known as Star Join Schema and is optimized for querying large data sets.

In the following example, the fact table is at the center which contains keys to every dimension table like Dealer_ID, Model ID, Date_ID, Product_ID, Branch_ID & other attributes like Units sold and revenue.



Example of Star Schema

Characteristics of Star Schema:

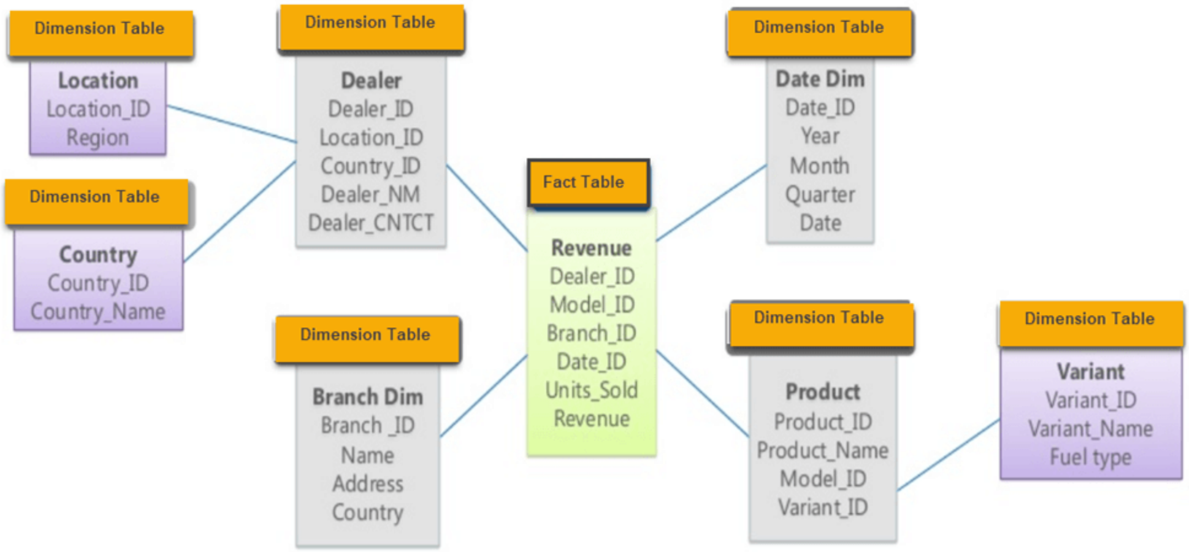
- Every dimension in a star schema is represented with the only one-dimension table.
- The dimension table should contain the set of attributes.
- The dimension table is joined to the fact table using a foreign key
- The dimension table are not joined to each other
- Fact table would contain key and measure
- The Star schema is easy to understand and provides optimal disk usage.

- The dimension tables are **not normalized**. For instance, in the above figure, Country_ID does not have Country lookup table as an OLTP design would have.
- The schema is widely supported by BI Tools

What is a Snowflake Schema?

SNOWFLAKE SCHEMA is a logical arrangement of tables in a multidimensional database such that the ER diagram resembles a snowflake shape. A Snowflake Schema is an extension of a Star Schema, and it adds additional dimensions. The dimension tables are **normalized** which splits data into additional tables.

In the following example, Country is further normalized into an individual table.

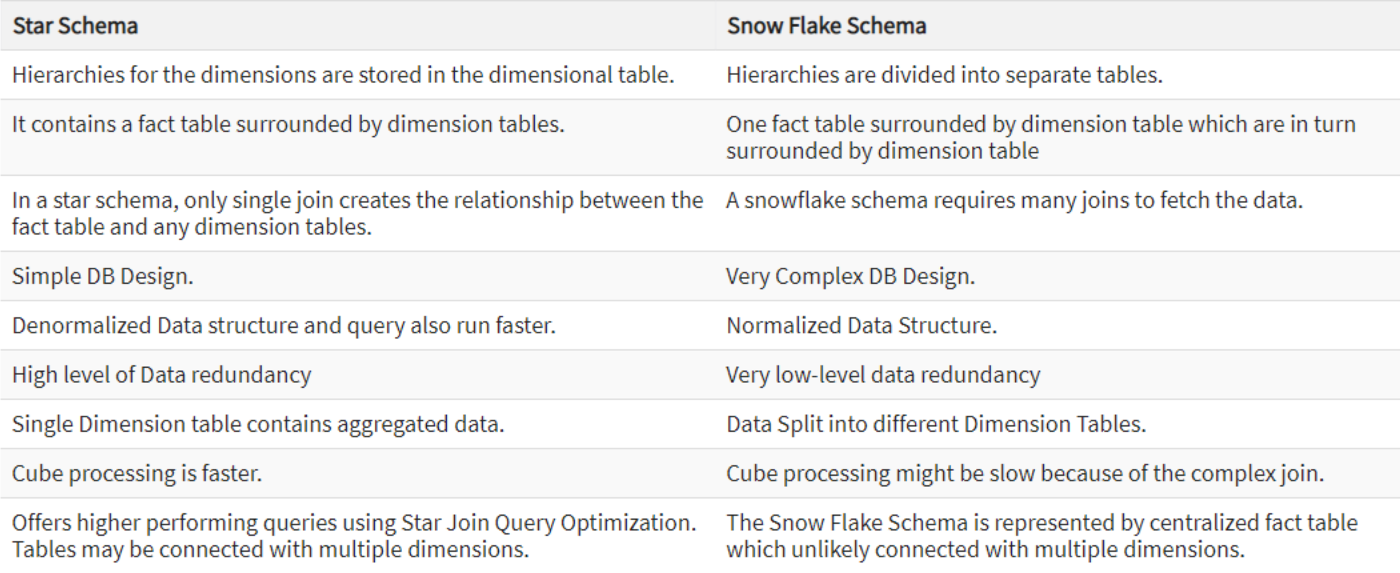


Example of Snowflake Schema

Characteristics of Snowflake Schema:

- The main benefit of the snowflake schema it uses smaller disk space.
- Easier to implement a dimension is added to the Schema
- Due to multiple tables query performance is reduced
- The primary challenge that you will face while using the snowflake Schema is that you need to perform more maintenance efforts because of the more lookup tables.

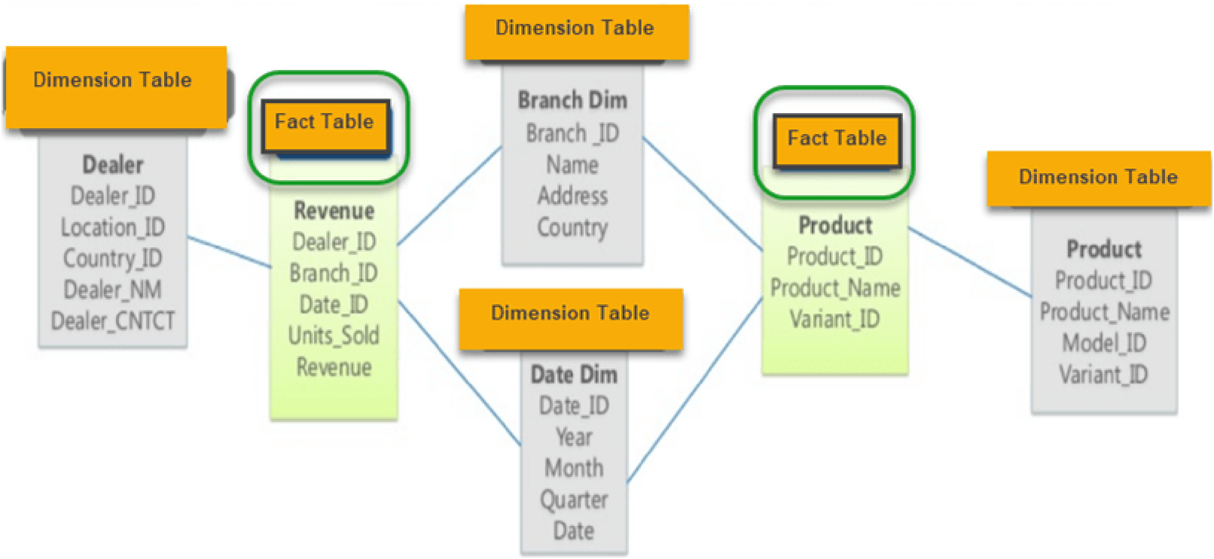
Star Vs Snowflake Schema: Key Differences



Star Schema	Snow Flake Schema
Hierarchies for the dimensions are stored in the dimensional table.	Hierarchies are divided into separate tables.
It contains a fact table surrounded by dimension tables.	One fact table surrounded by dimension table which are in turn surrounded by dimension table
In a star schema, only single join creates the relationship between the fact table and any dimension tables.	A snowflake schema requires many joins to fetch the data.
Simple DB Design.	Very Complex DB Design.
Denormalized Data structure and query also run faster.	Normalized Data Structure.
High level of Data redundancy	Very low-level data redundancy
Single Dimension table contains aggregated data.	Data Split into different Dimension Tables.
Cube processing is faster.	Cube processing might be slow because of the complex join.
Offers higher performing queries using Star Join Query Optimization. Tables may be connected with multiple dimensions.	The Snow Flake Schema is represented by centralized fact table which unlikely connected with multiple dimensions.

What is a Galaxy schema?

A **GALAXY SCHEMA** contains two fact table that share dimension tables between them. It is also called Fact Constellation Schema. The schema is viewed as a collection of stars hence the name Galaxy Schema.



Example of Galaxy Schema

As you can see in above example, there are two facts table

1. Revenue
2. Product.

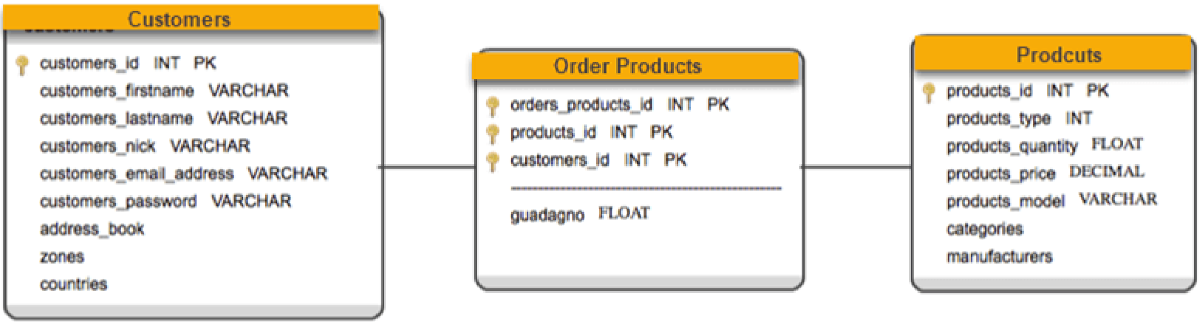
In Galaxy schema shares dimensions are called Conformed Dimensions.

Characteristics of Galaxy Schema:

- The dimensions in this schema are separated into separate dimensions based on the various levels of hierarchy.
- For example, if geography has four levels of hierarchy like region, country, state, and city then Galaxy schema should have four dimensions.
- Moreover, it is possible to build this type of schema by splitting the one-star schema into more Star schemes.
- The dimensions are large in this schema which is needed to build based on the levels of hierarchy.
- This schema is helpful for aggregating fact tables for better understanding.

What is Star Cluster Schema?

Snowflake schema contains fully expanded hierarchies. However, this can add complexity to the Schema and requires extra joins. On the other hand, star schema contains fully collapsed hierarchies, which may lead to redundancy. So, the best solution may be a balance between these two schemas which is **STAR CLUSTER SCHEMA** design.



Example of Star Cluster Schema

Overlapping dimensions can be found as forks in hierarchies. A fork happens when an entity acts as a parent in two different dimensional hierarchies. Fork entities then identified as classification with one-to-many relationships.

Summary:

- Multidimensional schema is especially designed to model data warehouse systems
- The star schema is the simplest type of Data Warehouse schema. It is known as star schema as its structure resembles a star.
- A Snowflake Schema is an extension of a Star Schema, and it adds additional dimensions. It is called snowflake because its diagram resembles a Snowflake.

- In a star schema, only single join defines the relationship between the fact table and any dimension tables.
- Star schema contains a fact table surrounded by dimension tables.
- Snow flake schema is surrounded by dimension table which are in turn surrounded by dimension table
- A snowflake schema requires many joins to fetch the data.
- A Galaxy Schema contains two fact table that shares dimension tables. It is also called Fact Constellation Schema.
- Star cluster schema contains attributes of Start schema and Slow flake schema.

What is Data Mart

A **DATA MART** is focused on a single functional area of an organization and contains a subset of data stored in a Data Warehouse. A Data Mart is a condensed version of Data Warehouse and is designed for use by a specific department, unit or set of users in an organization. E.g., Marketing, Sales, HR or finance. It is often controlled by a single department in an organization.

Data Mart usually draws data from only a few sources compared to a Data warehouse. Data marts are small in size and are more flexible compared to a Datawarehouse.

Why do we need Data Mart?

- Data Mart helps to enhance user's response time due to reduction in volume of data
- It provides easy access to frequently requested data.
- Data mart are simpler to implement when compared to corporate Datawarehouse. At the same time, the cost of implementing Data Mart is certainly lower compared with implementing a full data warehouse.
- Compared to Data Warehouse, a datamart is agile. In case of change in model, datamart can be built quicker due to a smaller size.
- A Datamart is defined by a single Subject Matter Expert. On the contrary data warehouse is defined by interdisciplinary SME from a variety of domains. Hence, Data mart is more open to change compared to Datawarehouse.
- Data is partitioned and allows very granular access control privileges.
- Data can be segmented and stored on different hardware/software platforms.

Type of Data Mart

There are three main types of data marts are:

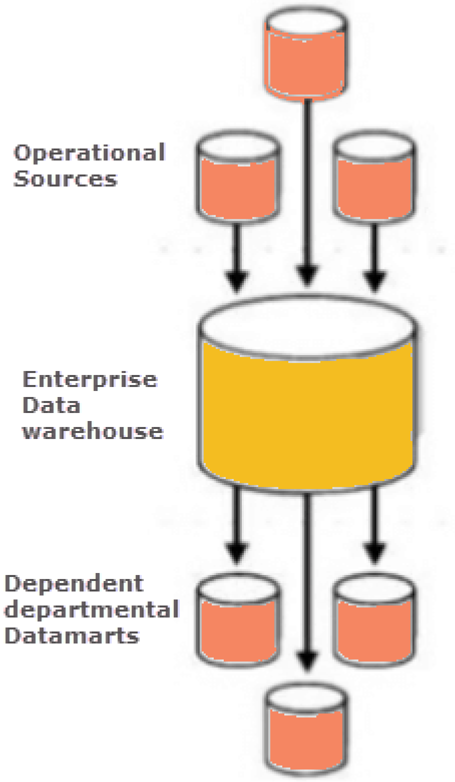
1. **Dependent**: Dependent data marts are created by drawing data directly from operational, external or both sources.
2. **Independent**: Independent data mart is created without the use of a central data warehouse.

3. **Hybrid**: This type of data marts can take data from data warehouses or operational systems.

Dependent Data Mart

A dependent data mart allows sourcing organization's data from a single Data Warehouse. It offers the benefit of centralization. If you need to develop one or more physical data marts, then you need to configure them as dependent data marts.

Dependent data marts can be built in two different ways. Either where a user can access both the data mart and data warehouse, depending on need, or where access is limited only to the data mart. The second approach is not optimal as it produces sometimes referred to as a data junkyard. In the data junkyard, all data begins with a common source, but they are scrapped, and mostly junked.

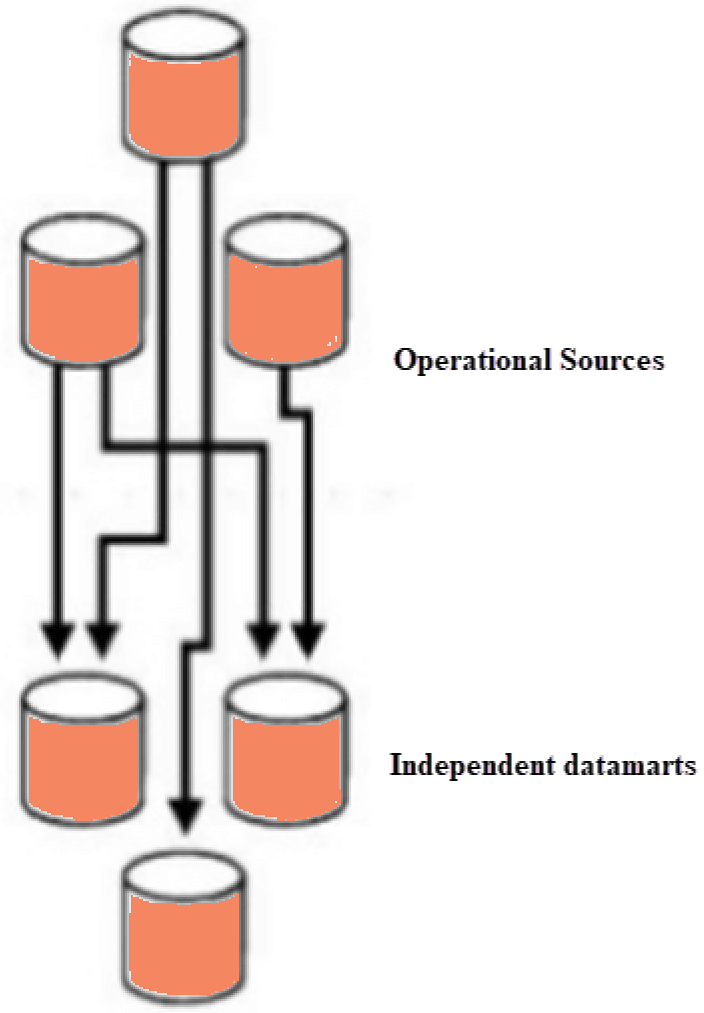


Independent Data Mart

An independent data mart is created without the use of central Data warehouse. This kind of Data Mart is an ideal option for smaller groups within an organization.

An independent data mart has neither a relationship with the enterprise data warehouse nor with any other data mart. In Independent data mart, the data is input separately, and its analyses are also performed autonomously.

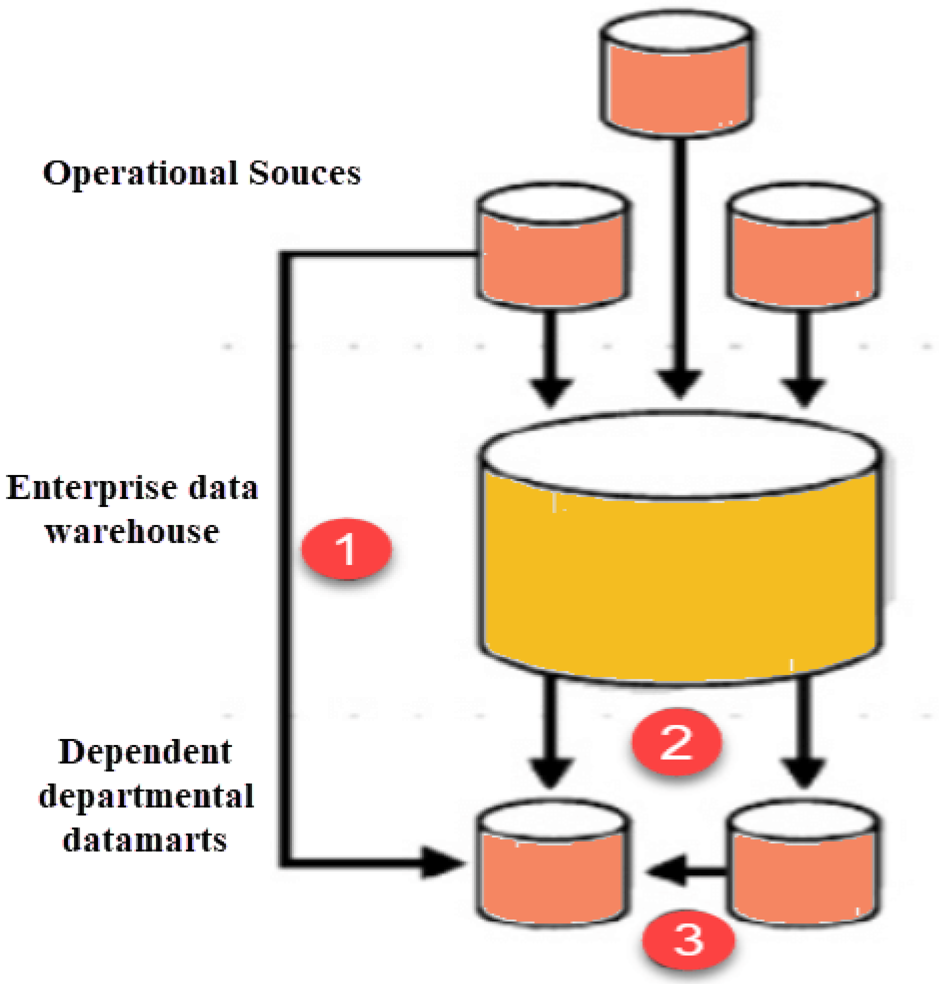
Implementation of independent data marts is antithetical to the motivation for building a data warehouse. First of all, you need a consistent, centralized store of enterprise data which can be analyzed by multiple users with different interests who want widely varying information.



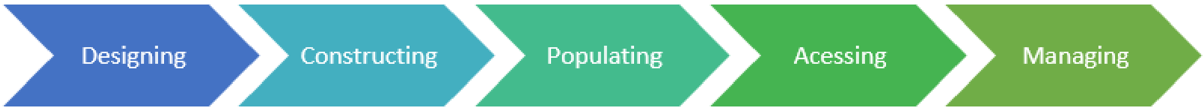
Hybrid data Mart:

A hybrid data mart combines input from sources apart from Data warehouse. This could be helpful when you want ad-hoc integration, like after a new group or product is added to the organization.

It is best suited for multiple database environments and fast implementation turnaround for any organization. It also requires least data cleansing effort. Hybrid Data mart also supports large storage structures, and it is best suited for flexible for smaller data-centric applications.



Steps in Implementing a Datamart



Implementing a Data Mart is a rewarding but complex procedure. Here are the detailed steps to implement a Data Mart:

Designing

Designing is the first phase of Data Mart implementation. It covers all the tasks between initiating the request for a data mart to gathering information about the requirements. Finally, we create the logical and physical design of the data mart.

The design step involves the following tasks:

- Gathering the business & technical requirements and Identifying data sources.
- Selecting the appropriate subset of data.
- Designing the logical and physical structure of the data mart.

Data could be partitioned based on following criteria:

- Date
- Business or Functional Unit
- Geography
- Any combination of above

Data could be partitioned at the application or DBMS level. Though it is recommended to partition at the Application level as it allows different data models each year with the change in business environment.

What Products and Technologies Do You Need?

A simple pen and paper would suffice. Though tools that help you create UML or ER diagrams would also append meta data into your logical and physical designs.

Constructing

This is the second phase of implementation. It involves creating the physical database and the logical structures.

This step involves the following tasks:

- Implementing the physical database designed in the earlier phase. For instance, database schema objects like table, indexes, views, etc. are created.

What Products and Technologies Do You Need?

You need a relational database management system to construct a data mart. RDBMS have several features that are required for the success of a Data Mart.

- **Storage management:** An RDBMS stores and manages the data to create, add, and delete data.
- **Fast data access:** With a SQL query you can easily access data based on certain conditions/filters.

- **Data protection:** The RDBMS system also offers a way to recover from system failures such as power failures. It also allows restoring data from these backups incase of the disk fails.
- **Multiuser support:** The data management system offers concurrent access, the ability for multiple users to access and modify data without interfering or overwriting changes made by another user.
- **Security:** The RDMS system also provides a way to regulate access by users to objects and certain types of operations.

Populating:

In the third phase, data in populated in the data mart.

The populating step involves the following tasks:

- Source data to target data Mapping
- Extraction of source data
- Cleaning and transformation operations on the data
- Loading data into the data mart
- Creating and storing metadata

What Products and Technologies Do You Need?

You accomplish these population tasks using an ETL (Extract Transform Load) Tool. This tool allows you to look at the data sources, perform source-to-target mapping, extract the data, transform, cleanse it, and load it back into the data mart.

In the process, the tool also creates some metadata relating to things like where the data came from, how recent it is, what type of changes were made to the data, and what level of summarization was done.

Accessing

Accessing is a fourth step which involves putting the data to use: querying the data, creating reports, charts, and publishing them. End-user submit queries to the database and display the results of the queries

The accessing step needs to perform the following tasks:

- Set up a meta layer that translates database structures and objects names into business terms. This helps non-technical users to access the Data mart easily.
- Set up and maintain database structures.
- Set up API and interfaces if required

What Products and Technologies Do You Need?

You can access the data mart using the command line or GUI. GUI is preferred as it can easily generate graphs and is user-friendly compared to the command line.

Managing

This is the last step of Data Mart Implementation process. This step covers management tasks such as-

- Ongoing user access management.
- System optimizations and fine-tuning to achieve the enhanced performance.
- Adding and managing fresh data into the data mart.
- Planning recovery scenarios and ensure system availability in the case when the system fails.

What Products and Technologies Do You Need?

You could use the GUI or command line for data mart management.

Best practices for Implementing Data Marts

Following are the best practices that you need to follow while in the Data Mart Implementation process:

- The source of a Data Mart should be departmentally structured
- The implementation cycle of a Data Mart should be measured in short periods of time, i.e., in weeks instead of months or years.
- It is important to involve all stakeholders in planning and designing phase as the data mart implementation could be complex.
- Data Mart Hardware/Software, Networking and Implementation costs should be accurately budgeted in your plan
- Even though if the Data mart is created on the same hardware they may need some different software to handle user queries. Additional processing power and disk storage requirements should be evaluated for fast user response
- A data mart may be on a different location from the data warehouse. That's why it is important to ensure that they have enough networking capacity to handle the Data volumes needed to transfer data to the data mart**.**
- Implementation cost should budget the time taken for Datamart loading process. Load time increases with increase in complexity of the transformations.

Advantages and Disadvantages of a Data Mart

Advantages

- Data marts contain a subset of organization-wide data. This Data is valuable to a specific group of people in an organization.
- It is cost-effective alternatives to a data warehouse, which can take high costs to build.
- Data Mart allows faster access of Data.
- Data Mart is easy to use as it is specifically designed for the needs of its users. Thus a data mart can accelerate business processes.
- Data Marts needs less implementation time compare to Data Warehouse systems. It is faster to implement Data Mart as you only need to concentrate the only subset of the data.
- It contains historical data which enables the analyst to determine data trends.

Disadvantages

- Many a times enterprises create too many disparate and unrelated data marts without much benefit. It can become a big hurdle to maintain.
- Data Mart cannot provide company-wide data analysis as their data set is limited.

Summary:

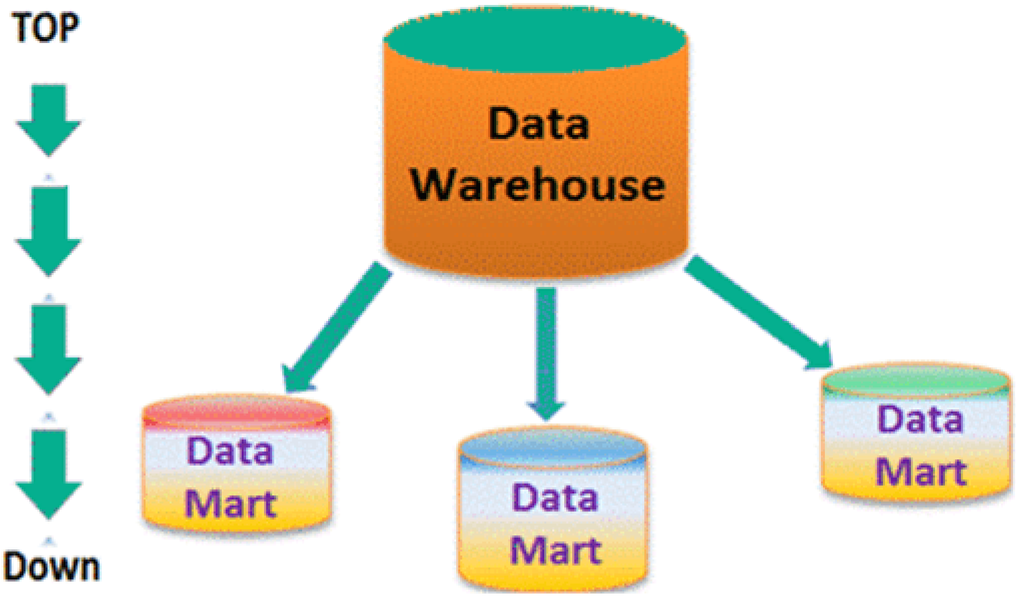
- A Data Mart is defined as a subset of Data Warehouse that is focused on a single functional area of an organization.
- Data Mart helps to enhance user's response time due to a reduction in the volume of data.
- Three types of data mart are 1) Dependent 2) Independent 3) Hybrid
- Important implementation steps of Data Mart are 1) Designing 2) Constructing 3 Populating 4) Accessing and 5) Managing
- The implementation cycle of a Data Mart should be measured in short periods of time, i.e., in weeks instead of months or years.
- Data mart is cost-effective alternatives to a data warehouse, which can take high costs to build.
- Data Mart cannot provide company-wide data analysis as data set is limited.

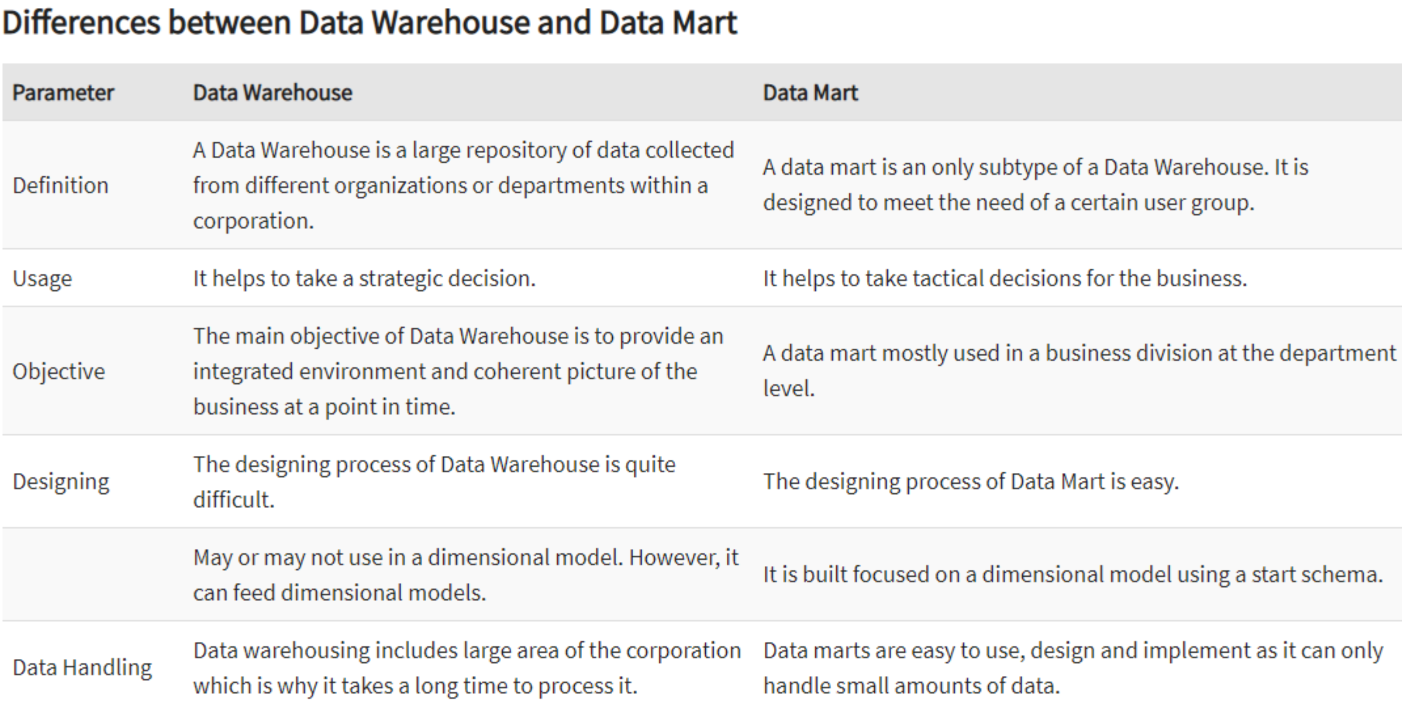
Data Mart vs Data Warehouse

KEY DIFFERENCE

- Data Warehouse is a large repository of data collected from different sources whereas Data Mart is only subtype of a data warehouse.
- Data Warehouse is focused on all departments in an organization whereas Data Mart focuses on a specific group.
- Data Warehouse designing process is complicated whereas the Data Mart process is easy to design.

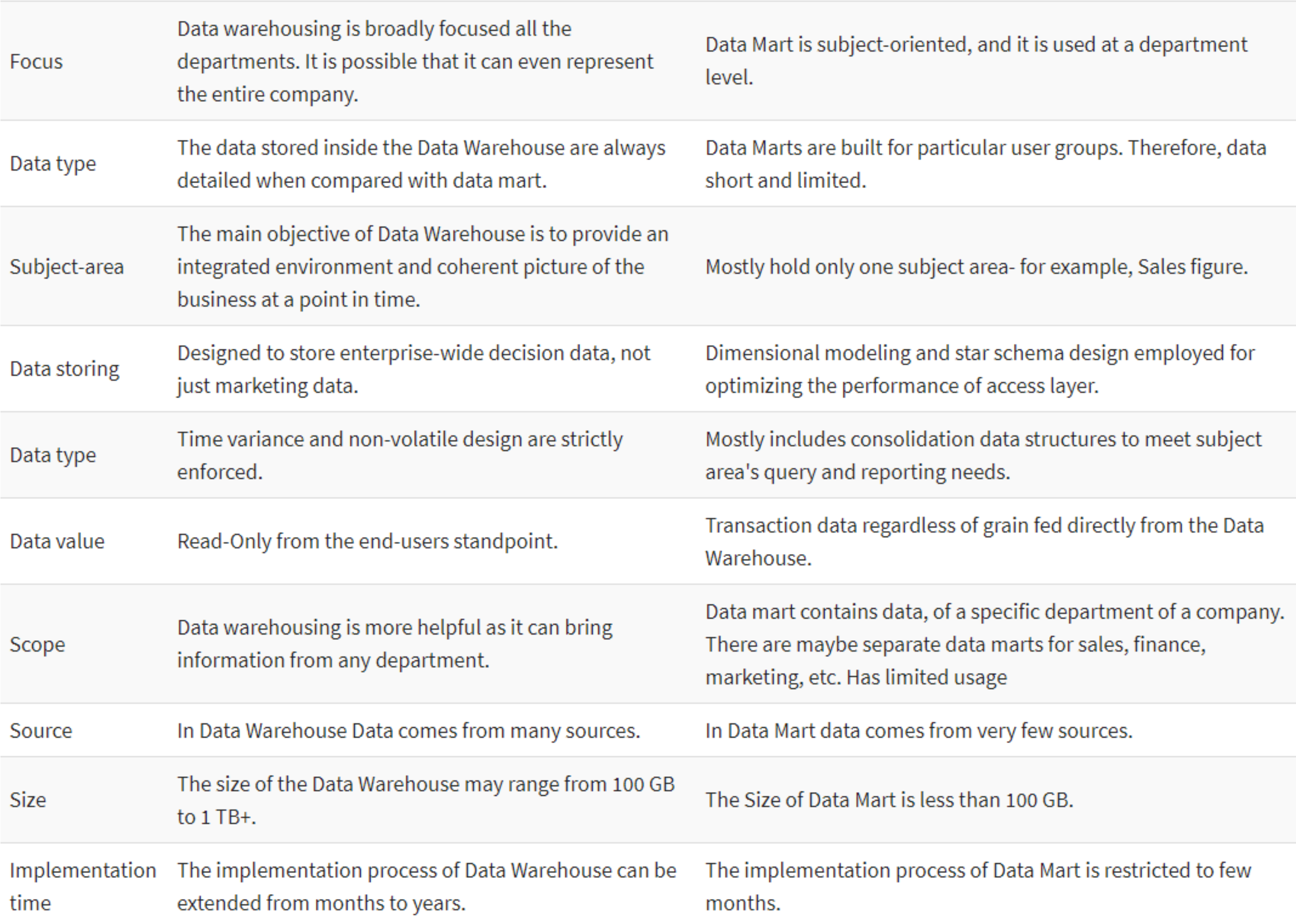
- Data Warehouse takes a long time for data handling whereas Data Mart takes a short time for data handling.
- Data Warehouse size range is 100 GB to 1 TB+ whereas Data Mart size is less than 100 GB.
- Data Warehouse implementation process takes 1 month to 1 year whereas Data Mart takes a few months to complete the implementation process.





Differences between Data Warehouse and Data Mart

Parameter	Data Warehouse	Data Mart
Definition	A Data Warehouse is a large repository of data collected from different organizations or departments within a corporation.	A data mart is an only subtype of a Data Warehouse. It is designed to meet the need of a certain user group.
Usage	It helps to take a strategic decision.	It helps to take tactical decisions for the business.
Objective	The main objective of Data Warehouse is to provide an integrated environment and coherent picture of the business at a point in time.	A data mart mostly used in a business division at the department level.
Designing	The designing process of Data Warehouse is quite difficult.	The designing process of Data Mart is easy.
	May or may not use in a dimensional model. However, it can feed dimensional models.	It is built focused on a dimensional model using a start schema.
Data Handling	Data warehousing includes large area of the corporation which is why it takes a long time to process it.	Data marts are easy to use, design and implement as it can only handle small amounts of data.

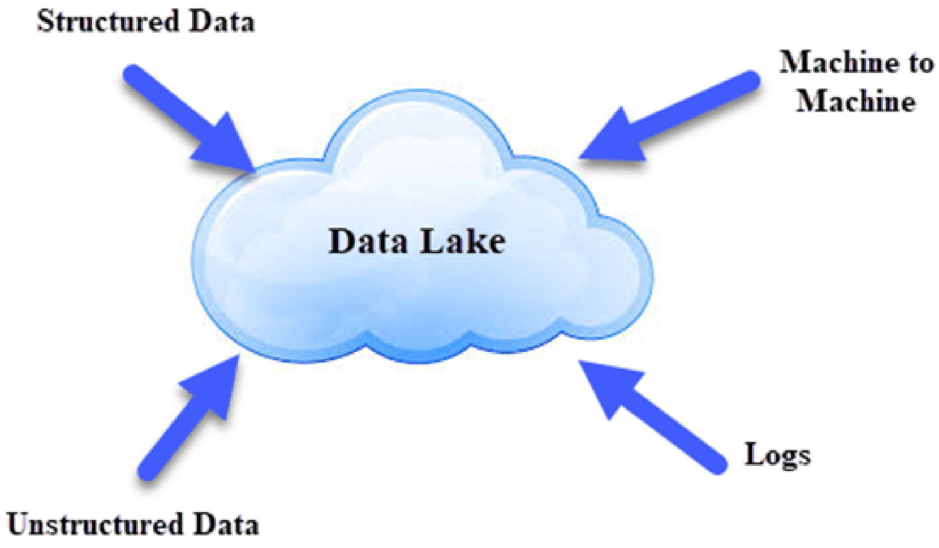


Focus	Data warehousing is broadly focused all the departments. It is possible that it can even represent the entire company.	Data Mart is subject-oriented, and it is used at a department level.
Data type	The data stored inside the Data Warehouse are always detailed when compared with data mart.	Data Marts are built for particular user groups. Therefore, data short and limited.
Subject-area	The main objective of Data Warehouse is to provide an integrated environment and coherent picture of the business at a point in time.	Mostly hold only one subject area- for example, Sales figure.
Data storing	Designed to store enterprise-wide decision data, not just marketing data.	Dimensional modeling and star schema design employed for optimizing the performance of access layer.
Data type	Time variance and non-volatile design are strictly enforced.	Mostly includes consolidation data structures to meet subject area's query and reporting needs.
Data value	Read-Only from the end-users standpoint.	Transaction data regardless of grain fed directly from the Data Warehouse.
Scope	Data warehousing is more helpful as it can bring information from any department.	Data mart contains data, of a specific department of a company. There are maybe separate data marts for sales, finance, marketing, etc. Has limited usage
Source	In Data Warehouse Data comes from many sources.	In Data Mart data comes from very few sources.
Size	The size of the Data Warehouse may range from 100 GB to 1 TB+.	The Size of Data Mart is less than 100 GB.
Implementation time	The implementation process of Data Warehouse can be extended from months to years.	The implementation process of Data Mart is restricted to few months.

What is Data Lake?

A Data Lake is a storage repository that can store large amount of structured, semi-structured, and unstructured data. It is a place to store every type of data in its native format with no fixed limits on account size or file. It offers high data quantity to increase analytic performance and native integration.

Data Lake is like a large container which is very similar to real lake and rivers. Just like in a lake you have multiple tributaries coming in, a data lake has structured data, unstructured data, machine to machine, logs flowing through in real-time.



The Data Lake democratizes data and is a cost-effective way to store all data of an organization for later processing. Research Analyst can focus on finding meaning patterns in data and not data itself.

Unlike a hierarchal Dataware house where data is stored in Files and Folder, Data lake has a flat architecture. Every data elements in a Data Lake is given a unique identifier and tagged with a set of metadata information.

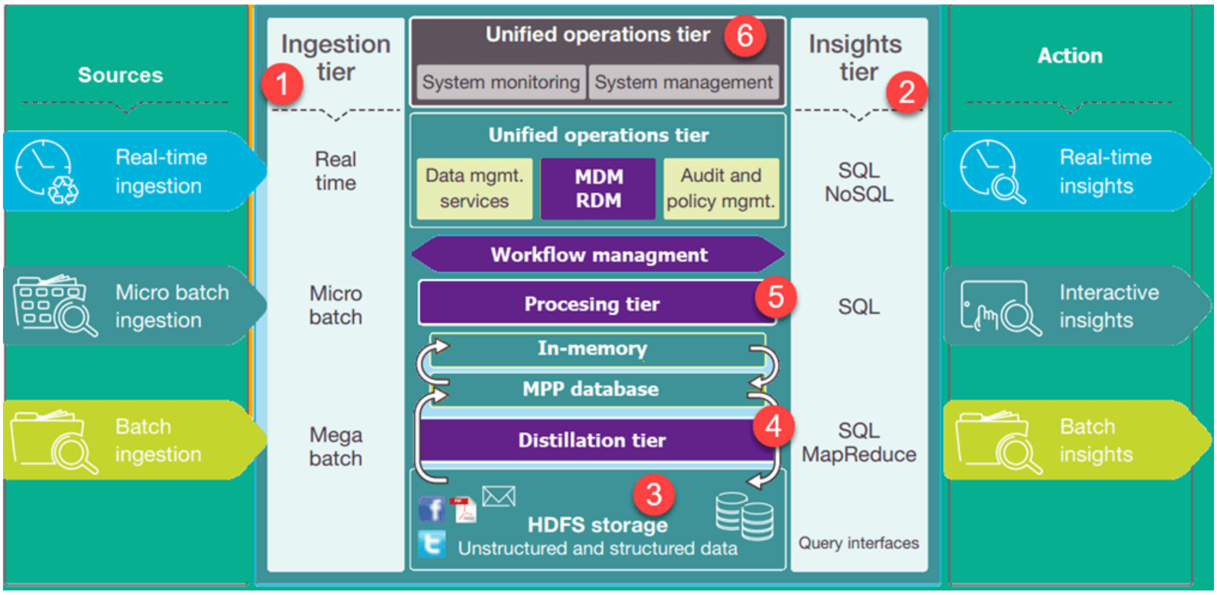
Why Data Lake?

The main objective of building a data lake is to offer an unrefined view of data to data scientists.

Reasons for using Data Lake are:

- With the onset of storage engines like Hadoop storing disparate information has become easy. There is no need to model data into an enterprise-wide schema with a Data Lake.
- With the increase in data volume, data quality, and metadata, the quality of analyses also increases.
- Data Lake offers business Agility
- Machine Learning and Artificial Intelligence can be used to make profitable predictions.
- It offers a competitive advantage to the implementing organization.
- There is no data silo structure. Data Lake gives 360 degrees view of customers and makes analysis more robust.

Data Lake Architecture

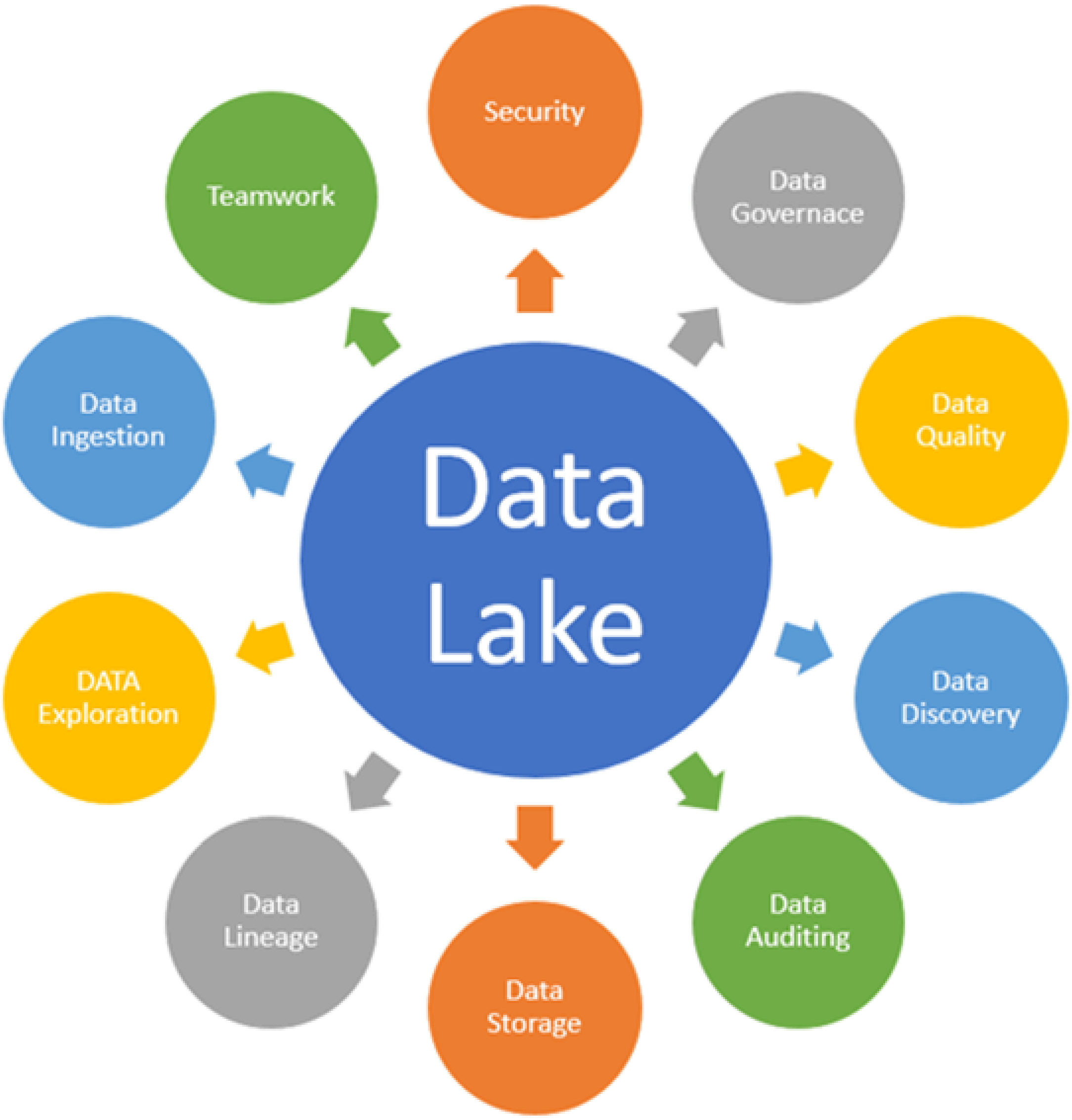


The figure shows the architecture of a Business Data Lake. The lower levels represent data that is mostly at rest while the upper levels show real-time transactional data. This data flow through the system with no or little latency. Following are important tiers in Data Lake Architecture:

1. **Ingestion Tier**: The tiers on the left side depict the data sources. The data could be loaded into the data lake in batches or in real-time
2. **Insights Tier:** The tiers on the right represent the research side where insights from the system are used. SQL, NoSQL queries, or even excel could be used for data analysis.
3. **HDFS** is a cost-effective solution for both structured and unstructured data. It is a landing zone for all data that is at rest in the system.
4. **Distillation tier** takes data from the storage tire and converts it to structured data for easier analysis.
5. **Processing tier** run analytical algorithms and users queries with varying real time, interactive, batch to generate structured data for easier analysis.
6. **Unified operations tier** governs system management and monitoring. It includes auditing and proficiency management, data management, workflow management.

Key Data Lake Concepts

Following are Key Data Lake concepts that one needs to understand to completely understand the Data Lake Architecture



Data Ingestion

Data Ingestion allows connectors to get data from a different data sources and load into the Data lake.

Data Ingestion supports:

- All types of Structured, Semi-Structured, and Unstructured data.
- Multiple ingestions like Batch, Real-Time, One-time load.
- Many types of data sources like Databases, Webservers, Emails, IoT, and FTP.

Data Storage

Data storage should be scalable, offers cost-effective storage and allow fast access to data exploration. It should support various data formats.

Data Governance

Data governance is a process of managing availability, usability, security, and integrity of data used in an organization.

Security

Security needs to be implemented in every layer of the Data lake. It starts with Storage, Unearthing, and Consumption. The basic need is to stop access for unauthorized users. It should support different tools to access data with easy to navigate GUI and Dashboards.

Authentication, Accounting, Authorization and Data Protection are some important features of data lake security.

Data Quality:

Data quality is an essential component of Data Lake architecture. Data is used to exact business value. Extracting insights from poor quality data will lead to poor quality insights.

Data Discovery

Data Discovery is another important stage before you can begin preparing data or analysis. In this stage, tagging technique is used to express the data understanding, by organizing and interpreting the data ingested in the Data lake.

Data Auditing

Two major Data auditing tasks are tracking changes to the key dataset.

1. Tracking changes to important dataset elements
2. Captures how/ when/ and who changes to these elements.

Data auditing helps to evaluate risk and compliance.

Data Lineage

This component deals with data's origins. It mainly deals with where it movers over time and what happens to it. It eases errors corrections in a data analytics process from origin to destination.

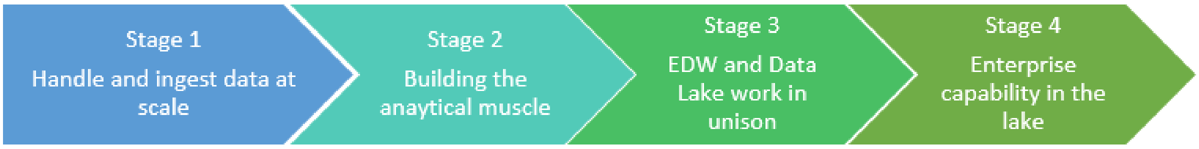
Data Exploration

It is the beginning stage of data analysis. It helps to identify right dataset is vital before starting Data Exploration.

All given components need to work together to play an important part in Data lake building easily evolve and explore the environment.

Maturity stages of Data Lake

The Definition of Data Lake Maturity stages differs from textbook to other. Though the crux remains the same. Following maturity, stage definition is from a layman point of view.


Stage 1
Handle and ingest data at scale
Stage 2
Building the anaytical muscle
Stage 3
EDW and Data Lake work in unison
Stage 4
Enterprise capability in the lake

Stage 1: Handle and ingest data at scale

This first stage of Data Maturity Involves improving the ability to transform and analyze data. Here, business owners need to find the tools according to their skillset for obtaining more data and build analytical applications.

Stage 2: Building the analytical muscle

This is a second stage which involves improving the ability to transform and analyze data. In this stage, companies use the tool which is most appropriate to their skillset. They start acquiring more data and building applications. Here, capabilities of the enterprise data warehouse and data lake are used together.

Stage 3: EDW and Data Lake work in unison

This step involves getting data and analytics into the hands of as many people as possible. In this stage, the data lake and the enterprise data warehouse start to work in a union. Both playing their part in analytics

Stage 4: Enterprise capability in the lake

In this maturity stage of the data lake, enterprise capabilities are added to the Data Lake. Adoption of information governance, information lifecycle management capabilities, and Metadata management. However, very few organizations can reach this level of maturity, but this tally will increase in the future.

Best practices for Data Lake Implementation:

- Architectural components, their interaction and identified products should support native data types
- Design of Data Lake should be driven by what is available instead of what is required. The schema and data requirement is not defined until it is queried
- Design should be guided by disposable components integrated with service API.
- Data discovery, ingestion, storage, administration, quality, transformation, and visualization should be managed independently.
- The Data Lake architecture should be tailored to a specific industry. It should ensure that capabilities necessary for that domain are an inherent part of the design
- Faster on-boarding of newly discovered data sources is important
- Data Lake helps customized management to extract maximum value
- The Data Lake should support existing enterprise data management techniques and methods

Challenges of building a data lake:

- In Data Lake, Data volume is higher, so the process must be more reliant on programmatic administration
- It is difficult to deal with sparse, incomplete, volatile data
- Wider scope of dataset and source needs larger data governance & support

Benefits and Risks of using Data Lake:

Here are some major benefits in using a Data Lake:

- Helps fully with product ionizing & advanced analytics
- Offers cost-effective scalability and flexibility
- Offers value from unlimited data types
- Reduces long-term cost of ownership
- Allows economic storage of files
- Quickly adaptable to changes
- The main advantage of data lake is the **centralization** of different content sources
- Users, from various departments, may be scattered around the globe can have **flexible access** to the data

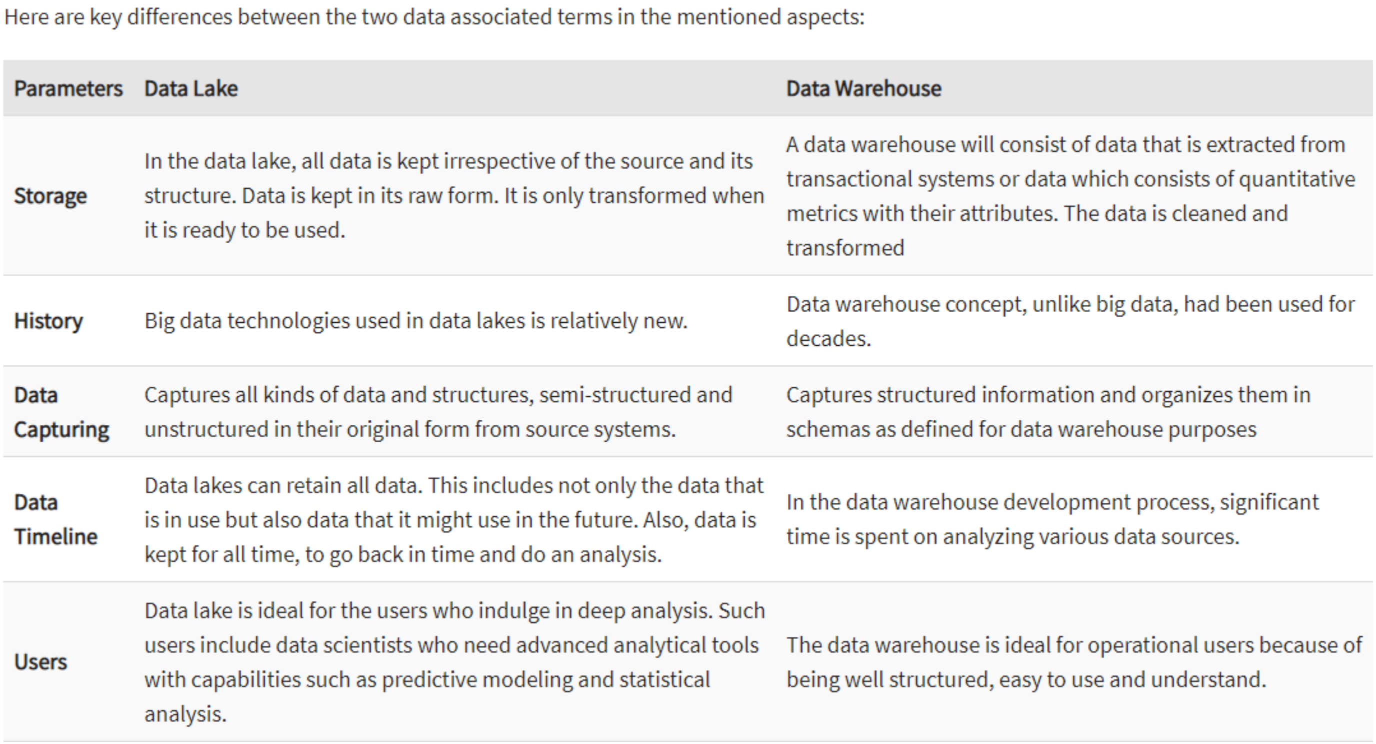
Risk of Using Data Lake:

- After some time, Data Lake may lose relevance and momentum
- There is larger amount risk involved while designing Data Lake
- Unstructured Data may lead to Ungoverned Chao, Unusable Data, Disparate & Complex Tools, Enterprise-Wide Collaboration, Unified, Consistent, and Common
- It also increases storage & computes costs
- There is no way to get insights from others who have worked with the data because there is no account of the lineage of findings by previous analysts

- The biggest risk of data lakes is security and access control. Sometimes data can be placed into a lake without any oversight, as some of the data may have privacy and regulatory need

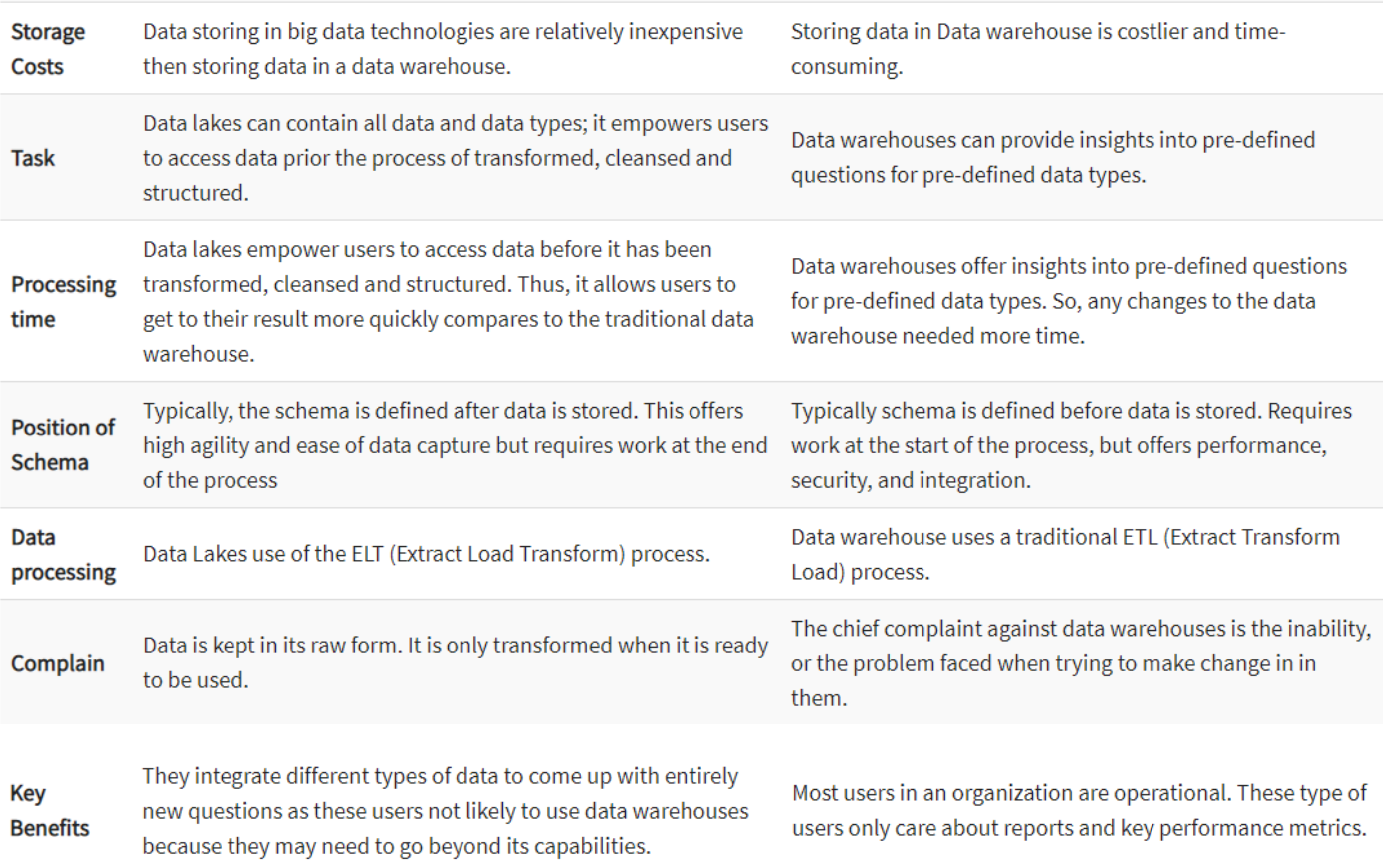
Summary:

- A Data Lake is a storage repository that can store large amount of structured, semi-structured, and unstructured data.
- The main objective of building a data lake is to offer an unrefined view of data to data scientists.
- Unified operations tier, Processing tier, Distillation tier and HDFS are important layers of Data Lake Architecture
- Data Ingestion, Data storage, Data quality, Data Auditing, Data exploration, Data discover are some important components of Data Lake Architecture
- Design of Data Lake should be driven by what is available instead of what is required.
- Data Lake reduces long-term cost of ownership and allows economic storage of files
- The biggest risk of data lakes is security and access control. Sometimes data can be placed into a lake without any oversight, as some of the data may have privacy and regulatory need.



Here are key differences between the two data associated terms in the mentioned aspects:

Parameters	Data Lake	Data Warehouse
Storage	In the data lake, all data is kept irrespective of the source and its structure. Data is kept in its raw form. It is only transformed when it is ready to be used.	A data warehouse will consist of data that is extracted from transactional systems or data which consists of quantitative metrics with their attributes. The data is cleaned and transformed
History	Big data technologies used in data lakes is relatively new.	Data warehouse concept, unlike big data, had been used for decades.
Data Capturing	Captures all kinds of data and structures, semi-structured and unstructured in their original form from source systems.	Captures structured information and organizes them in schemas as defined for data warehouse purposes
Data Timeline	Data lakes can retain all data. This includes not only the data that is in use but also data that it might use in the future. Also, data is kept for all time, to go back in time and do an analysis.	In the data warehouse development process, significant time is spent on analyzing various data sources.
Users	Data lake is ideal for the users who indulge in deep analysis. Such users include data scientists who need advanced analytical tools with capabilities such as predictive modeling and statistical analysis.	The data warehouse is ideal for operational users because of being well structured, easy to use and understand.



Storage Costs	Data storing in big data technologies are relatively inexpensive then storing data in a data warehouse.	Storing data in Data warehouse is costlier and time-consuming.
Task	Data lakes can contain all data and data types; it empowers users to access data prior the process of transformed, cleansed and structured.	Data warehouses can provide insights into pre-defined questions for pre-defined data types.
Processing time	Data lakes empower users to access data before it has been transformed, cleansed and structured. Thus, it allows users to get to their result more quickly compares to the traditional data warehouse.	Data warehouses offer insights into pre-defined questions for pre-defined data types. So, any changes to the data warehouse needed more time.
Position of Schema	Typically, the schema is defined after data is stored. This offers high agility and ease of data capture but requires work at the end of the process	Typically schema is defined before data is stored. Requires work at the start of the process, but offers performance, security, and integration.
Data processing	Data Lakes use of the ELT (Extract Load Transform) process.	Data warehouse uses a traditional ETL (Extract Transform Load) process.
Complain	Data is kept in its raw form. It is only transformed when it is ready to be used.	The chief complaint against data warehouses is the inability, or the problem faced when trying to make change in in them.
Key Benefits	They integrate different types of data to come up with entirely new questions as these users not likely to use data warehouses because they may need to go beyond its capabilities.	Most users in an organization are operational. These type of users only care about reports and key performance metrics.

I hope my document was useful 😊

- Alperen KEZAY

Source:

https://www.guru99.com/data-warehousing.html

https://en.wikipedia.org/wiki/Data_warehouse

https://www.javatpoint.com/types-of-data-warehouses

https://panoply.io/data-warehouse-guide/the-difference-between-a-database-and-a-data-warehouse/

https://aws.amazon.com/tr/data-warehouse/#:~:text=How%20does%20a%20data%20warehouse