1. Data warehouse Introduction

Data Warehouse is a relational database management system (RDBMS) construct to meet the requirement of transaction processing systems.

It can be loosely described as any centralized data repository which can be queried for business benefits. It is a database that stores information oriented to satisfy decision-making requests.

It is a group of decision support technologies, targets to enabling the knowledge worker (executive, manager, and analyst) to make superior and higher decisions. So, Data Warehousing support architectures and tool for business executives to systematically organize, understand and use their information to make strategic decisions.

Data Warehouse environment contains an extraction, transportation, and loading (ETL) solution, an online analytical processing (OLAP) engine, customer analysis tools, and other applications that handle the process of gathering information and delivering it to business users.

What is a Data Warehouse?

A Data Warehouse (DW) is a relational database that is designed for query and analysis rather than transaction processing. It includes historical data derived from transaction data from single and multiple sources.

A Data Warehouse provides integrated, enterprise-wide, historical data and focuses on providing support for decision-makers for data modeling and analysis.

"Data Warehouse is a subject-oriented, integrated, and time-variant store of information in support of management's decisions."

Characteristics of Data Warehouse:

-Subject-Oriented

A data warehouse target on the modeling and analysis of data for decision-makers. Therefore, data warehouses typically provide a concise and straightforward view around a particular subject, such as customer, product, or sales, instead of the global organization's ongoing operations. This is done by excluding data that are not useful concerning the subject and including all data needed by the users to understand the subject.

-Integrated

A data warehouse integrates various heterogeneous data sources like RDBMS, flat files, and online transaction records. It requires performing data cleaning and integration during data warehousing to ensure consistency in naming conventions, attributes types, etc., among different data sources.

-Time-Variant

Historical information is kept in a data warehouse. For example, one can retrieve files from 3 months, 6 months, 12 months, or even previous data from a data warehouse. These variations with a transactions system, where often only the most current file is kept.

-Non-Volatile

The data warehouse is a physically separate data storage, which is transformed from the source operational RDBMS. The operational updates of data do not occur in the data warehouse, i.e., update, insert, and delete operations are not performed. It usually requires only two procedures in data accessing: Initial loading of data and access to data. Therefore, the DW does not require transaction processing, recovery, and concurrency capabilities, which allows for substantial speedup of data retrieval. Non-Volatile defines that once entered into the warehouse, and data should not change.

History of Data Warehouse

The idea of data warehousing came to the late 1980's when IBM researchers Barry Devlin and Paul Murphy established the "Business Data Warehouse."

In essence, the data warehousing idea was planned to support an architectural model for the flow of information from the operational system to decisional support environments. The concept attempt to address the various problems associated with the flow, mainly the high costs associated with it.

In the absence of data warehousing architecture, a vast amount of space was required to support multiple decision support environments. In large corporations, it was ordinary for various decision support environments to operate independently.

Goals of Data Warehousing

- To help reporting as well as analysis
- Maintain the organization's historical information
- Be the foundation for decision making.

Need for Data Warehouse

Data Warehouse is needed for the following reasons:

1) Business User: Business users require a data warehouse to view summarized data from the past. Since these people are non-technical, the data may be presented to them in an elementary form.

2) Store historical data: Data Warehouse is required to store the time variable data from the past. This input is made to be used for various purposes.

3) Make strategic decisions: Some strategies may be depending upon the data in the data warehouse. So, data warehouse contributes to making strategic decisions.

4) For data consistency and quality: Bringing the data from different sources at a commonplace, the user can effectively undertake to bring the uniformity and consistency in data.

5) High response time: Data warehouse has to be ready for somewhat unexpected loads and types of queries, which demands a significant degree of flexibility and quick response time.

Benefits of Data Warehouse

Understand business trends and make better forecasting decisions.

Data Warehouses are designed to perform well enormous amounts of data.

The structure of data warehouses is more accessible for end-users to navigate, understand, and query.

Queries that would be complex in many normalized databases could be easier to build and maintain in data warehouses.

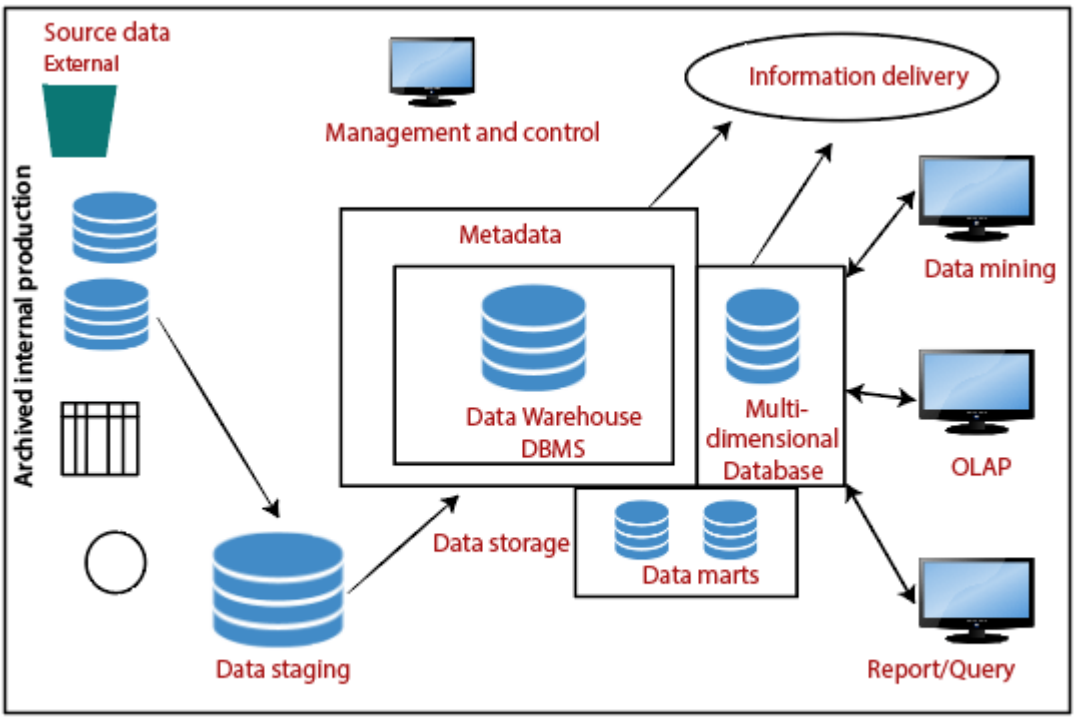
Data warehousing is an efficient method to manage demand for lots of information from lots of users.

Data warehousing provide the capabilities to analyze a large amount of historical data.

2.Data warehouse components

Components or Building Blocks of Data Warehouse:

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



Components or Building Blocks of Data Warehouse

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

Source Data Component

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

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

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

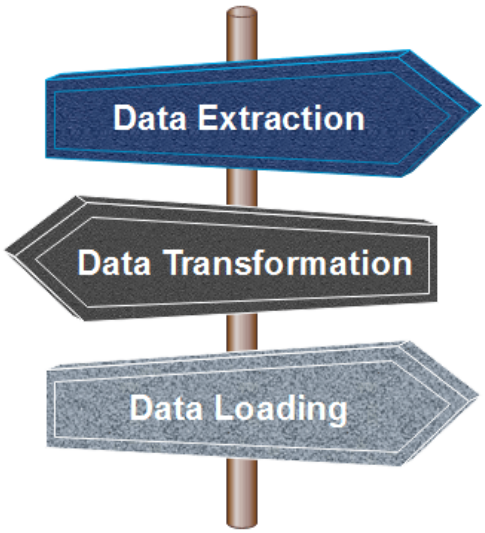
Archived Data: Operational systems are mainly intended to run the current business. In every operational system, we periodically take the old data and store it in achieved files.

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

Data Staging Component

After we have been extracted data from various operational systems and external sources, we have to prepare the files for storing in the data warehouse. The extracted data coming from several different sources need to be changed, converted, and made ready in a format that is relevant to be saved for querying and analysis.

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



1) Data Extraction: This method has to deal with numerous data sources. We have to employ the appropriate techniques for each data source.
2) Data Transformation: As we know, data for a data warehouse comes from many different sources. If data extraction for a data warehouse posture big challenges, data transformation present even significant challenges. We perform several individual tasks as part of data transformation.

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

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

   On the other hand, data transformation also contains purging source data that is not useful and separating outsource records into new combinations. Sorting and merging of data take place on a large scale in the data staging area. When the data transformation function ends, we have a collection of integrated data that is cleaned, standardized, and summarized.
3) Data Loading: Two distinct categories of tasks form data loading functions. When we complete the structure and construction of the data warehouse and go live for the first time, we do the

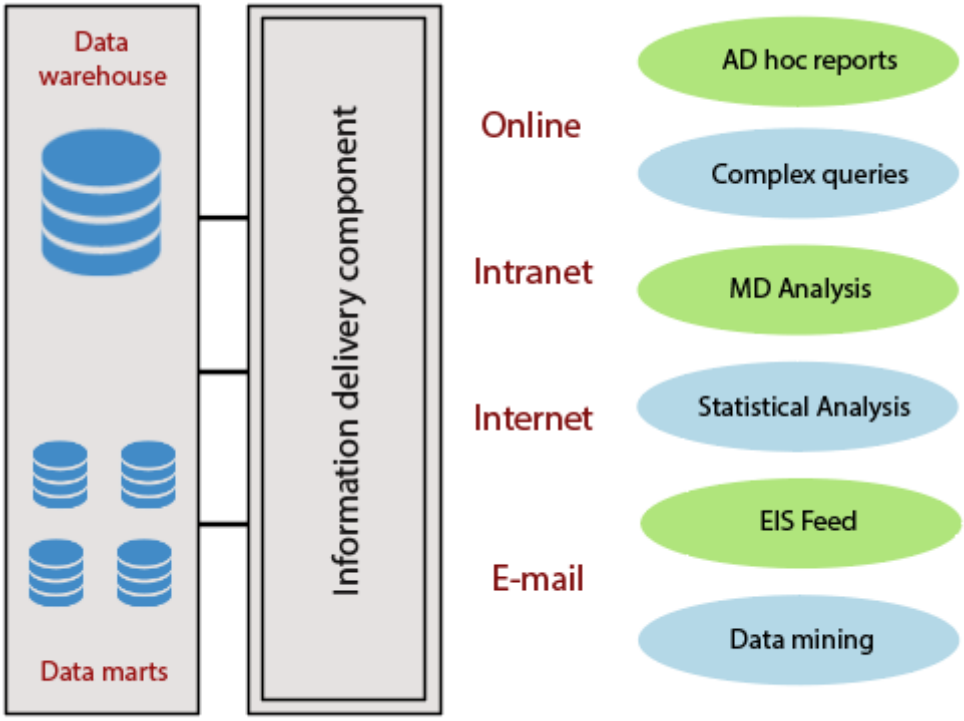
initial loading of the information into the data warehouse storage. The initial load moves high volumes of data using up a substantial amount of time.

Data Storage Components

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

Information Delivery Component

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



Information delivery component

Metadata Component

Metadata in a data warehouse is equal to the data dictionary or the data catalog in a database management system. In the data dictionary, we keep the data about the logical data structures,

the data about the records and addresses, the information about the indexes, and so on.

Data Marts

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

Management and Control Component

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

3.Operational Database vs Data Warehouse

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

Difference between OLTP and OLAP

OLTP System

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

OLAP System

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

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

Feature	OLTP	OLAP
Characteristic	It is a system which is used to manage operational Data.	It is a system which is used to manage informational Data.
Users	Clerks, clients, and information technology professionals.	Knowledge workers, including managers, executives, and analysts.
System orientation	OLTP system is a customer-oriented, transaction, and query processing are done by clerks, clients, and information technology professionals.	OLAP system is market-oriented, knowledge workers including managers, do data analysts executive and analysts.
Data contents	OLTP system manages current data that too detailed and are used for decision making.	OLAP system manages a large amount of historical data, provides facilitates for summarization and aggregation, and stores and manages data at different levels of granularity.

		This information makes the data more comfortable to use in informed decision making.
Database Size	100 MB-GB	100 GB-TB
Database design	OLTP system usually uses an entity-relationship (ER) data model and application-oriented database design.	OLAP system typically uses either a star or snowflake model and subject-oriented database design.
View	OLTP system focuses primarily on the current data within an enterprise or department, without referring to historical information or data in different organizations.	OLAP system often spans multiple versions of a database schema, due to the evolutionary process of an organization. OLAP systems also deal with data that originates from various organizations, integrating information from many data stores.
Volume of data	Not very large	Because of their large volume, OLAP data are stored on multiple storage media.
Access patterns	The access patterns of an OLTP system subsist mainly of short, atomic transactions. Such a system requires concurrency control and recovery techniques.	Accesses to OLAP systems are mostly read-only methods because of these data warehouses stores historical data.
Access mode	Read/write	Mostly write
Insert and Updates	Short and fast inserts and updates proposed by end-users.	Periodic long-running batch jobs refresh the data.
Number of records accessed	Tens	Millions
Normalization	Fully Normalized	Partially Normalized
Processing Speed	Very Fast	It depends on the amount of files contained, batch data refresh, and complex query may take many hours, and query speed can be upgraded by creating indexes.

4. Data warehouse Architecture

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

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

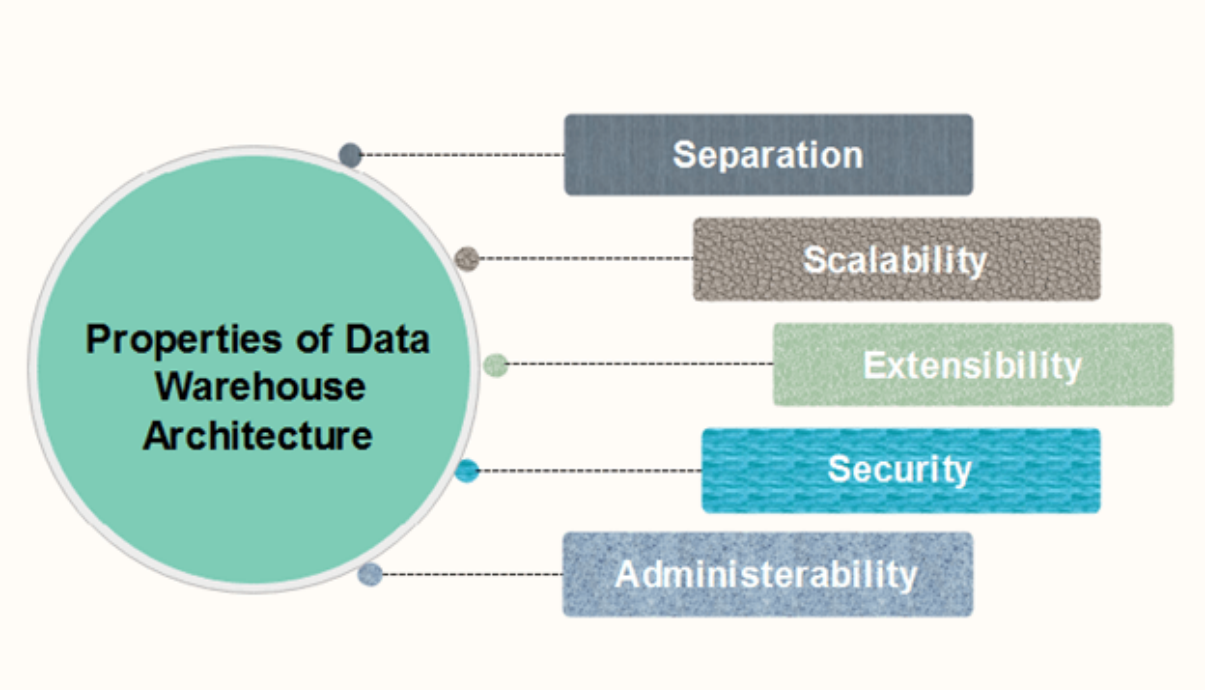
Data Warehouse applications are designed to support the user ad-hoc data requirements, an activity recently dubbed online analytical processing (OLAP). These include applications such as forecasting, profiling, summary reporting, and trend analysis.

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

Three common architectures are:

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

Properties of Data Warehouse Architectures:



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

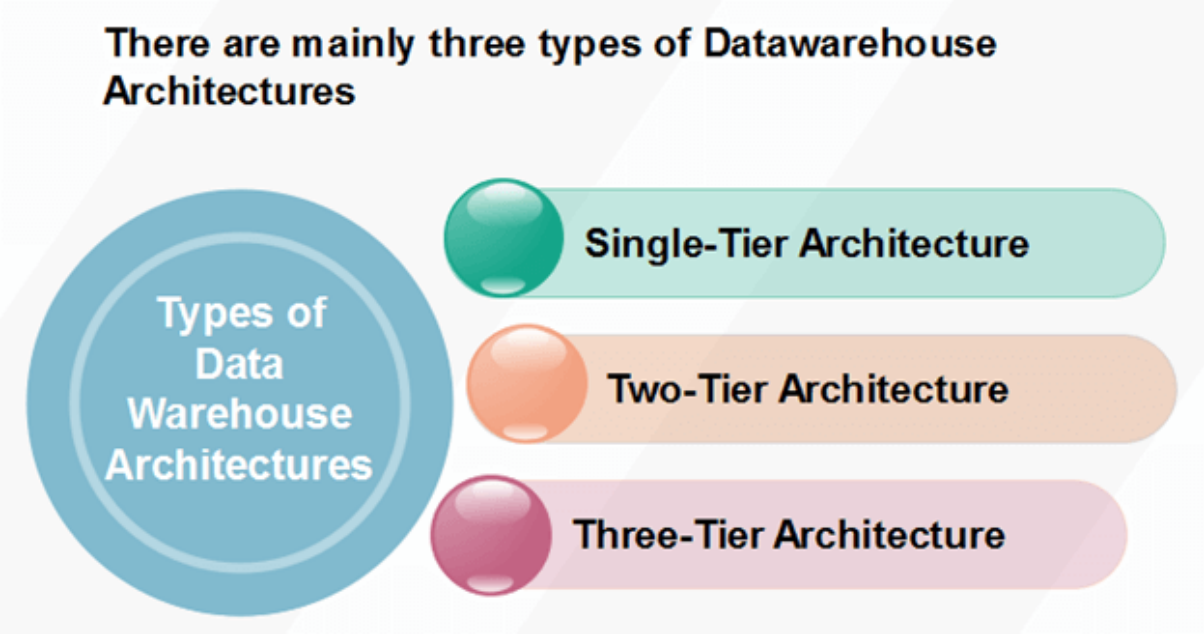
2. Scalability: Hardware and software architectures should be simple to upgrade the data volume, which has to be managed and processed, and the number of user's requirements, which have to be met, progressively increase.

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

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

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

Types of Data Warehouse Architectures



5. Three-tier Data Warehouse Architecture

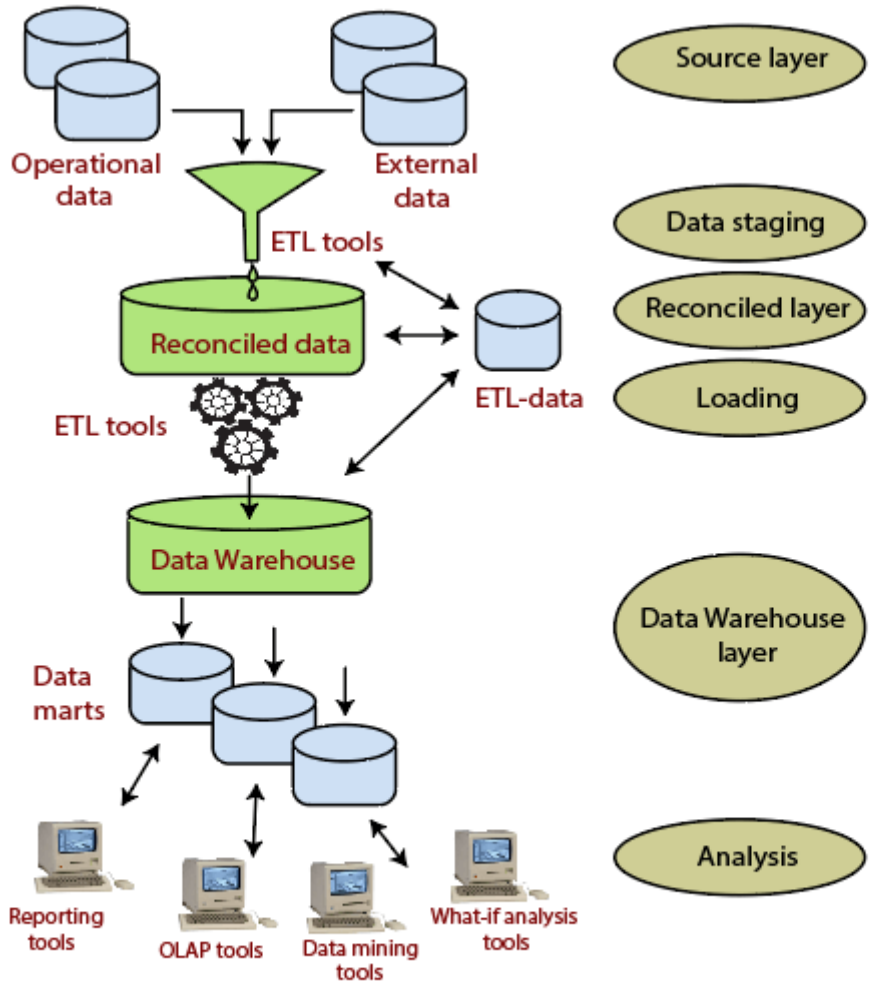
The three-tier architecture consists of the source layer (containing multiple source system), the reconciled layer and the data warehouse layer (containing both data warehouses and data marts).

The reconciled layer sits between the source data and data warehouse.

The main advantage of the reconciled layer is that it creates a standard reference data model for a whole enterprise.

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

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



Three-Tier Architecture for a data warehouse system

Three-Tier Data Warehouse Architecture

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

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

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

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

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

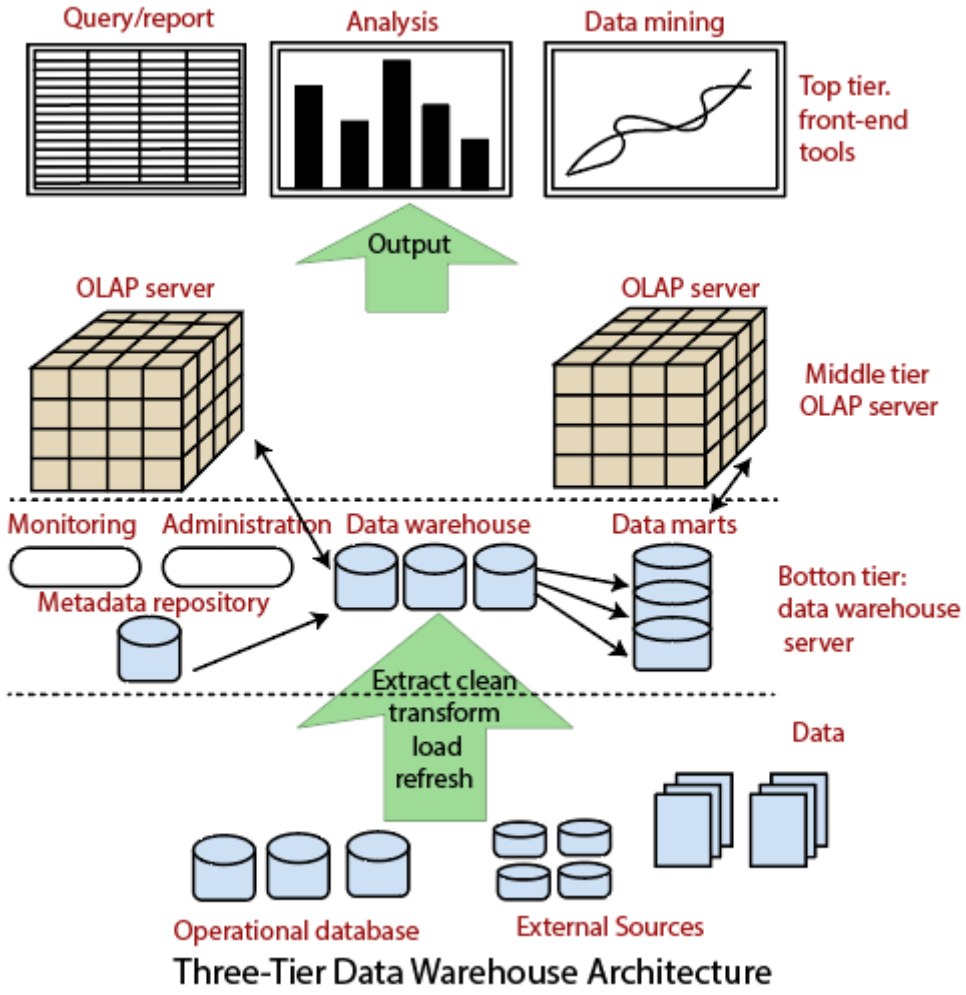
A middle-tier which consists of an OLAP server for fast querying of the data warehouse.

The OLAP server is implemented using either

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

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

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



Three-Tier Data Warehouse Architecture

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

A description of the DW structure, including the warehouse schema, dimension, hierarchies, data mart locations, and contents, etc.

Operational metadata, which usually describes the currency level of the stored data, i.e., active, archived or purged, and warehouse monitoring information, i.e., usage statistics, error reports, audit, etc.

System performance data, which includes indices, used to improve data access and retrieval performance.

Information about the mapping from operational databases, which provides source RDBMSs and their contents, cleaning and transformation rules, etc.

Summarization algorithms, predefined queries, and reports business data, which include business terms and definitions, ownership information, etc.

6. Autonomous Data Warehouse

Oracle Autonomous Data Warehouse is the world's first and only autonomous database optimized for analytic workloads, including data marts, data warehouses, data lakes, and data lakehouses.

With Autonomous Data Warehouse, data scientists, business analysts, and nonexperts can rapidly, easily, and cost-effectively discover business insights using data of any size and type.

Built for the cloud and optimized using Oracle Exadata, Autonomous Data Warehouse benefits from faster performance and, according to an IDC report (PDF), lowers operational costs by an average of 63%.

Why ADW?

It is built on Oracle Database, that has automatic Datawarehouse procedures.

It is easy to use as all the management tasks are automated, all configuration and tuning tasks are fully automated. All data is automatically compressed and encrypted.

It is fast since its built on Exadata and Oracle database. It also offers instant elasticity on computing and storage dimensions. When it comes to elasticity, the user can choose the exact amount of storage and CPU as needed. Later when more CPU's are required, one can Scale Up or Scale down.

Machine Learning enables continuous optimization.

ML in ADW delivers an excellent query performance. Since its built-on oracle database, every business intelligence and Data Integration Services that are compatible with Oracle database

supports this service out of the box. For development purpose, existing tools or a newer version of SQL developer (which supports ADW) can be used.

ADW Patches all software online at all levels (security, OS, network, database) while the system is running.

ADW Features

Features of Oracle Autonomous Data Warehouse:

- **Self-Driving**: Automated database tuning and optimization for better performance and reduced manual tasks.
- **Self-Securing:** Automated security measures to protect data and prevent unauthorized access.
- **Self-Repairing:** Automatic error detection and resolution to ensure high availability.
- **Scalability:** ADW can scale compute and storage resources independently to match workload demands.
- **In-Memory Processing:** Utilizes in-memory columnar processing for faster query performance.
- **Parallel Execution:** Queries are processed in parallel across multiple nodes for faster results.
- Integration with Oracle Ecosystem: Seamless integration with other Oracle Cloud services and tools.
- **Data Encryption:** Provides data encryption both at rest and in transit for data security.
- **Easy Data Loading:** Supports data loading from various sources, including Oracle Data Pump, SQL Developer, and SQL*Loader.
- **Pay-as-You-Go Pricing:** Based on consumption, offering cost-effective pricing.

7. Autonomous Data Warehouse Vs Snowflake

Introduction

In today's data-driven world, businesses need robust and scalable data warehousing solutions to stay ahead of the competition. Two key players in this domain are Oracle Autonomous Data Warehouse (ADW) and Snowflake Data Cloud. Both platforms offer unique features and

capabilities for businesses looking to leverage the power of their data.

What Is Snowflake?

Snowflake is a Data Warehouse built for the cloud. It centralizes data from multiple sources, enabling you to run in-depth business insights that power your teams.

At its core, Snowflake is designed to handle structured and semi-structured data from various sources, allowing organizations to integrate and analyze data from diverse systems seamlessly. Its unique architecture separates compute and storage, enabling users to scale each independently based on their specific needs. This elasticity ensures optimal resource allocation and cost-efficiency, as users only pay for the actual compute and storage utilized.

Snowflake uses a SQL-based query language, making it accessible to data analysts and SQL developers. Its intuitive interface and user-friendly features allow for efficient data exploration, transformation, and analysis. Additionally, Snowflake provides robust security and compliance features, ensuring data privacy and protection.

One of Snowflake's notable strengths is its ability to handle large-scale, concurrent workloads without performance degradation. Its auto-scaling capabilities automatically adjust resources based on the workload demands, eliminating the need for manual tuning and optimization.

Another key advantage of Snowflake is its native integration with popular data processing and analytics

tools, such as Apache Spark, Python, and R. This compatibility enables seamless data integration, data engineering, and advanced analytics workflows.

Snowflake vs. Oracle: Which Is Best?

When comparing Snowflake and Oracle, two prominent players in the data warehousing landscape, several factors come into play. Let's delve into the comparison to help you determine which platform might be the best fit for your needs.

Scalability and Performance:

Snowflake: Snowflake's cloud-native architecture provides unparalleled scalability, allowing you to effortlessly scale compute and storage resources independently. Its multi-cluster architecture ensures optimal performance even with large-scale, concurrent workloads.

Oracle: Oracle offers robust scalability options, particularly with its Exadata and Autonomous Data Warehouse offerings. These solutions are engineered for high-performance data warehousing, enabling organizations to handle massive data volumes effectively.

Flexibility and Agility:

Snowflake: Snowflake's separation of compute and storage, along with its cloud-based nature, grants users the flexibility to scale resources on-demand and pay only for what is utilized. It also supports semi-structured data natively, allowing for easy integration and analysis of diverse data types.

Oracle: Oracle provides a comprehensive suite of data management tools and technologies that enable agility and flexibility. With its extensive ecosystem, organizations can leverage various Oracle products and

services for seamless integration and advanced analytics capabilities.

Ease of Use and User Experience:

Snowflake: Snowflake boasts a user-friendly interface and intuitive SQL-based query language, making it accessible to data analysts and SQL developers. Its self-tuning capabilities and auto-scaling features simplify administration and optimize performance.

Oracle: Oracle has a long-standing reputation for its user-friendly interfaces and robust tools. Oracle Database, combined with its analytics and business intelligence solutions, offers a familiar environment for users already experienced with Oracle technologies.

Integration and Ecosystem:

Snowflake: Snowflake provides native integration with popular data processing and analytics tools, facilitating seamless data integration and workflows. It has a growing ecosystem of partners and connectors, expanding its compatibility with various third-party systems.

Oracle: Oracle's extensive ecosystem offers a wide range of tools, applications, and industry-specific solutions. With its strong integration capabilities and partnerships, Oracle enables organizations to connect and consolidate their data across multiple sources effectively.

Security and Compliance:

Snowflake: Snowflake places a strong emphasis on security and compliance. It provides robust security features, including encryption, access controls, and compliance certifications, ensuring data protection and regulatory compliance.

Oracle: Oracle has a long history of prioritizing security and compliance. Its data management solutions offer advanced security features, auditing capabilities, and data governance controls to safeguard sensitive information.

8. Modern Data Warehouse

Modern data warehouses use cloud technologies to deliver more flexible data processing and analytics from various data sources.

The key difference is that modern data warehouses are cloud-based data warehouses. All of the other differences (architecture flexibility, data sources, etc.) generally come from this difference.

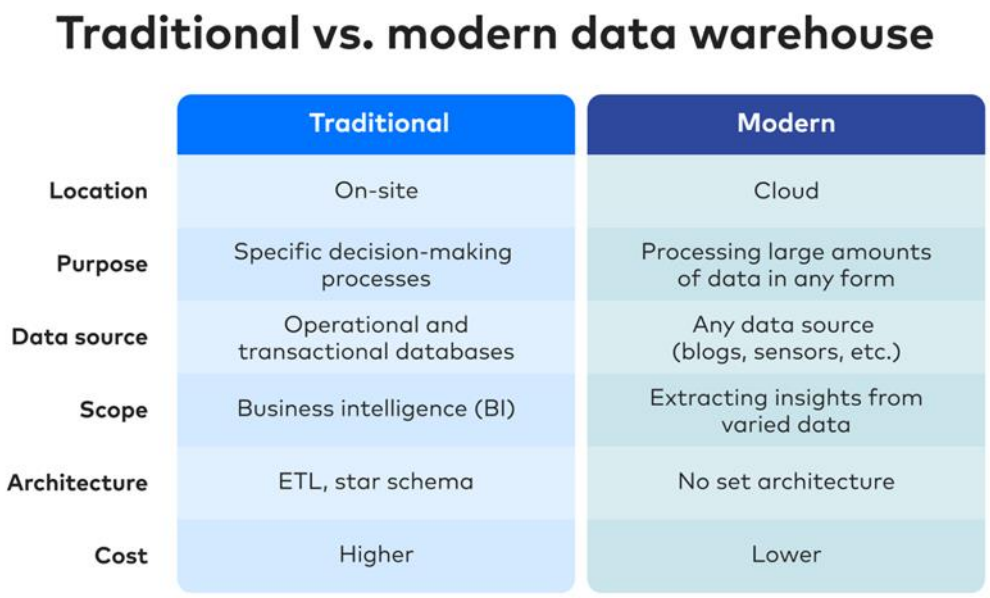
Modern data warehouse architecture

Modern data warehouses are much less limited. Not only can they use star schema, but they can also use specialized architectures such as:

- **Hybrid architectures:** Utilize a combination of on-premises and cloud infrastructure, usually with on-premises resources only serving to augment the cloud where necessary.
- **Massively Parallel Processing (MPP) architectures:** Data processing is distributed across multiple nodes or servers.
- **Lambda architectures:** Process vast amounts of data using a combination of layers (batch, speed and service). Here, data is simultaneously fed to the batch and speed layers, with the batch layer supporting raw data processing and the speed layer supporting low-latency data not already delivered to the batch layer. Meanwhile,

the service layer supports queries in real time. This architecture is common in big data applications.

Traditional vs. modern data warehouses



Traditional vs. modern data warehouse

	Traditional	Modern
Location	On-site	Cloud
Purpose	Specific decision-making processes	Processing large amounts of data in any form
Data source	Operational and transactional databases	Any data source (blogs, sensors, etc.)
Scope	Business intelligence (BI)	Extracting insights from varied data
Architecture	ETL, star schema	No set architecture
Cost	Higher	Lower

Benefits of a modern data warehouse

Important topics/questions:

1. Data warehouse introduction
2. Data warehouse components
3. Operational database vs Data warehouse
4. Data warehouse Architecture-Three tier Data warehouse Architecture
5. Autonomous Data warehouse
6. Autonomous Data warehouse vs Snowflake
7. Modern Data warehouse