What is Data Warehousing?

A data warehousing is defined as a technique for collecting and managing data from varied sources to provide meaningful business insights. It is a blend of technologies and components which aids the strategic use of data.

It is electronic storage of a large amount of information by a business which is designed for query and analysis instead of transaction processing. It is a process of transforming data into information and making it available to users in a timely manner to make a difference.

The decision support database (Data Warehouse) is maintained separately from the organization's operational database. However, the data warehouse is not a product but an environment. It is an architectural construct of an information system which provides users with current and historical decision support information which is difficult to access or present in the traditional operational data store.

The data warehouse is the core of the BI system which is built for data analysis and reporting.

Data warehouse system is also known by the following name:

- Decision Support System (DSS)
- Executive Information System
- Management Information System
- Business Intelligence Solution
- Analytic Application
- Data Warehouse

How Datawarehouse works?

A Data Warehouse works as a central repository where information arrives from one or more data sources. Data flows into a data warehouse from the transactional system and other relational databases.

Data may be:

1. Structured

2. Semi-structured

3. Unstructured data

The data is processed, transformed, and ingested so that users can access the processed data in the Data Warehouse through Business Intelligence tools, SQL clients, and spreadsheets. A data warehouse merges information coming from different sources into one comprehensive database. By merging all of this information in one place, an organization can analyze its customers more holistically. This helps to ensure that it has considered all the information available. Data warehousing makes data mining possible. Data mining is looking for patterns in the data that may lead to higher sales and profits.

Features of Data Warehouse

- Subject Oriented
- Integrated
- Time Variant
- Non-Volatile

Types of Data Warehouse

Three main types of Data Warehouses are:

1. **Enterprise Data Warehouse:** Enterprise Data Warehouse is a centralized warehouse. It provides decision support service across the enterprise. It offers a unified approach for organizing and representing data. It also provide the ability to classify data according to the subject and give access according to those divisions.

2**. Operational Data Store:** Operational Data Store, which is also called ODS, are nothing but data store required when neither Data warehouse nor OLTP systems support organizations reporting needs. In ODS, Data warehouse is refreshed in real time. Hence, it is widely preferred for routine activities like storing records of the Employees.

3. **Data Mart:** A data mart is a subset of the data warehouse. It specially designed for a particular line of business, such as sales, finance, sales or finance. In an independent data mart, data can collect directly from sources.

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 transformations 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 tables for scheduling the execution of queries.

End-user access tools: This is categorized 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

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 we want to discover 'hidden patterns' of data-flows and groupings.

Applications of Data warehouse:

- Information processing
  - supports querying, basic statistical analysis, and reporting using crosstabs, tables, charts and graphs.
- Analytical processing
  - multidimensional analysis of data warehouse data.
  - supports basic OLAP operations, slice-dice, drilling, pivoting.
- Data mining
  - knowledge discovery from hidden patterns .
  - supports associations, constructing analytical models, performing classification and prediction, and presenting the mining results using visualization tools.

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.

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.

DSS architectural styles

OLTP (Online Transaction Processing)

-used by traditional operational systems (RDBMS).

OLAP (Online Analytical Processing)

-used by Data Warehouse.

The Operational Database is one which is accessed by an Operational System to carry out regular operations of an organization.

Operational Databases usually use an OLTP architecture which is optimized for faster transaction processing.

OLTP Databases access the data in the form of operations like- Inserting, Deleting, and Updating data.

OLTP

- OLTP is a methodology to provide end users with access to large amounts of data
- It works in an intuitive and rapid manner to assist with deductions based on investigative reasoning.
- OLTP refers to a class of systems that facilitate and manage transaction-oriented applications, typically for data entry and retrieval transaction processing.

Advantages of OLTP

- Simplicity & Efficiency: Reduced paper trails and the faster & more accurate forecasts for revenues and expenses are both examples of how OLTP makes things simpler for businesses.
- OLTP systems maintain data integrity and they also provide fast query processing in environments having multiple access.

Disadvantage of OLTP

- OLTP requires instant update.
- The data what we get from OLTP is not suitable for data analysis.

- To perform one simple transaction even with the normalized structure, we need to query multiple tables by using joins.

OLAP Servers:

OLAP Server receives the data from data warehouse by which it represents the data in a user understandable format which actually supply analytical functionality for the DSS system.

OLAP Server generally performs data analysis in two forms:

- ROLAP (Relational OLAP)
- MOLAP (Multi-dimensional OLAP)

ROLAP

- It is a form of OLAP that performs dynamic multi- dimensional analysis of data stored in a relational database rather than in a multi-dimensional database (which is usually considered the OLAP standard).
- Data processing may take place within the database system, a mid-tier server, or the client.
- In two-tier architecture, the user submits a Structured Query Language (SQL) query to the database and receives back the requested data.

MOLAP (Multi-dimensional OLAP)

- It is a form of OLAP that helps the user to “slice and dice” information, providing multi-dimensional analysis of data by putting data in a cube structure.
- Most MOLAP products use a multi-cube approach in which a series of small, dense, pre-calculated cubes make a hypercube.

OLTP vs OLAP

OLAP (Online Analytical Processing) and OLTP (Online Transaction Processing) are two different types of database systems designed to serve distinct purposes in an organization. Here are the key differences between OLAP and OLTP:

OLAP (Online Analytical Processing):

Purpose:

Analytical: OLAP is designed for complex queries and analysis of historical and aggregated data. It supports decision-making and business intelligence activities.

Data Type:

Multidimensional Data: OLAP systems deal with multidimensional data models, enabling users to analyze data from different perspectives.

Database Structure:

Star or Snowflake Schema: OLAP databases typically use star or snowflake schema structures to store aggregated and summarized data in a way that facilitates fast query performance.

Query Performance:

Read-Optimized: OLAP databases are optimized for read-intensive operations, allowing for fast query performance on large volumes of data.

Concurrency:

Lower Concurrency Requirements: OLAP systems generally have lower concurrency requirements compared to OLTP systems because they are focused on analytical processing rather than transactional processing.

Examples:

Business Intelligence Tools: OLAP databases are commonly used in conjunction with business intelligence tools for data analysis and reporting.

OLTP (Online Transaction Processing):

Purpose:

Transactional: OLTP is designed for handling day-to-day transactional operations, such as inserting, updating, and deleting data. It supports real-time transactional processing.

Data Type:

Transactional Data: OLTP systems deal with transactional data, which is typically normalized to reduce redundancy and maintain data integrity.

Database Structure:

Normalized Schema: OLTP databases typically use normalized schema structures to ensure data integrity and minimize data redundancy.

Query Performance:

Write-Optimized: OLTP databases are optimized for write-intensive operations to handle a large number of transactions in real-time.

Concurrency:

High Concurrency Requirements: OLTP systems often have high concurrency requirements because they need to support multiple users simultaneously making transactions.

Examples:

E-commerce Systems: OLTP databases are commonly used in e-commerce systems, banking applications, and other environments where real-time transaction processing is critical.

RDBMS

RDBMS stands for Relational Database Management System. It is a type of database management system that organizes data into tables, which consist of rows and columns. The relational model, introduced by E.F. Codd in 1970, forms the basis for RDBMS. Here are the key components and characteristics of RDBMS:

Tables:

Structure: Data is organized into tables, where each table consists of rows and columns. Rows represent individual records, while columns represent attributes or fields of the records.

Table Relationships: Tables can be related to each other using common attributes, forming relationships that enable the linking of information across tables.

Data Integrity:

Entity Integrity: Each row in a table must have a unique identifier, known as a primary key, to ensure entity integrity.

Referential Integrity: Relationships between tables are maintained through foreign keys, ensuring that references between tables are valid.

Normalization:

Normalization Process: RDBMS supports the normalization process, which involves organizing data to reduce redundancy and improve data integrity. Normalization helps in avoiding data anomalies and ensures efficient data storage.

SQL (Structured Query Language):

Standard Query Language: SQL is the standard language used to interact with RDBMS. It provides a set of commands for creating, modifying, and querying relational databases.

CRUD Operations: SQL supports CRUD operations (Create, Read, Update, Delete), allowing users to manage data in the database.

ACID Properties:

Transaction Integrity: RDBMS adheres to ACID properties (Atomicity, Consistency, Isolation, Durability) to ensure the integrity of transactions. This means that database transactions are reliable even in the event of system failures.

Concurrent Access:

Concurrency Control: RDBMS includes mechanisms for controlling concurrent access to the database, allowing multiple users to work with the data simultaneously without compromising data consistency.

Popular RDBMS Systems:

Examples: Some popular RDBMS systems include MySQL, PostgreSQL, Oracle Database, Microsoft SQL Server, and SQLite.

SQL

SQL, which stands for Structured Query Language, is a programming language used for managing and manipulating relational databases. It provides a standardized way to interact with relational database management systems (RDBMS) and perform various operations on data.

Here are the key features of SQL:

Data Query Language (DQL):

SELECT Statement: SQL includes a powerful query language that allows users to retrieve data from one or more tables using the SELECT statement. This is part of the Data Query Language (DQL) aspect of SQL.

Data Definition Language (DDL):

CREATE, ALTER, DROP Statements: SQL includes statements for defining and managing the structure of a database. The Data Definition Language (DDL) allows users to create tables (CREATE), modify table structure (ALTER), and delete tables or other database objects (DROP).

Data Manipulation Language (DML):

INSERT, UPDATE, DELETE Statements: SQL provides statements for manipulating data within tables. The Data Manipulation Language (DML) includes commands for inserting new records into tables (INSERT), updating existing records (UPDATE), and deleting records (DELETE).

Data Control Language (DCL):

GRANT, REVOKE Statements: SQL includes statements for managing access and permissions to the database. The Data Control Language (DCL) allows users to grant or revoke privileges to other users or roles.

Transaction Control Language (TCL):

COMMIT, ROLLBACK, SAVEPOINT Statements: SQL provides statements to control transactions, ensuring the integrity and consistency of data. The Transaction Control Language (TCL) includes commands for committing transactions (COMMIT), rolling back changes (ROLLBACK), and setting savepoints within transactions.

Constraints:

UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK Constraints: SQL supports the definition of constraints to ensure data integrity. Constraints include uniqueness (UNIQUE), primary key (PRIMARY KEY), referential integrity through foreign keys (FOREIGN KEY), and conditions to check values (CHECK).

Joins:

JOIN Operations: SQL enables the combination of data from multiple tables through various types of joins, such as INNER JOIN, LEFT JOIN, and RIGHT JOIN. Joins allow users to retrieve and analyze related data from different tables.

Aggregation Functions:

SUM, AVG, COUNT, MAX, MIN: SQL includes aggregate functions for performing calculations on sets of values. Common aggregate functions include SUM, AVG (average), COUNT (count of rows), MAX (maximum value), and MIN (minimum value).

Grouping and Sorting:

GROUP BY, ORDER BY: SQL allows users to group data based on specific criteria using the GROUP BY clause. The ORDER BY clause is used to sort query results based on one or more columns.

Views:

CREATE VIEW Statement: SQL supports the creation of views, which are virtual tables derived from the result of a SELECT query. Views provide a way to simplify complex queries and present data in a more organized manner.