**UNIT V**

**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.

**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.

Key Characteristics of Data Warehouse

The main characteristics of a data warehouse are as follows:

Subject-Oriented

A data warehouse is subject-oriented since it provides topic-wise information rather than the overall processes of a business. Such subjects may be sales, promotion, inventory, etc. For example, if you want to analyze your company’s sales data, you need to build a data warehouse that concentrates on sales. Such a warehouse would provide valuable information like ‘who was your best customer last year?’ or ‘who is likely to be your best customer in the coming year?’

Integrated

A data warehouse is developed by integrating data from varied sources into a consistent format. The data must be stored in the warehouse in a consistent and universally acceptable manner in terms of naming, format, and coding. This facilitates effective [data analysis](https://www.simplilearn.com/data-analysis-methods-process-types-article).

Non-Volatile

Data once entered into a data warehouse must remain unchanged. All data is read-only. Previous data is not erased when current data is entered. This helps you to analyze what has happened and when.

Time-Variant

The data stored in a data warehouse is documented with an element of time, either explicitly or implicitly. An example of time variance in Data Warehouse is exhibited in the Primary Key, which must have an element of time like the day, week, or month.

**Difference between operational database and data warehouse**

| **Sr. No.** | **Key** | **Data warehouse** | **Operational Database** |
| --- | --- | --- | --- |
| 1 | Basic | A data warehouse is a repository for structured, filtered data that has already been processed for a specific purpose | Operational Database are those databases where data changes frequently |
| 2 | Data Structure | Data warehouse has denormalized schema | It has normalized schema |
| 3 | Performance | It is fast for analysis queries | It is slow for analytics queries |
| 4. | Type of Data | It focuses on historical data | It focuses on current transactional data |
| 5. | Uses Case | It is used for OLAP | It is used for OLTP |

Why do we need a separate Data Warehouse?

Data Warehousing is a technique that is mainly used to collect and manage data from various sources to give the business a meaningful business insight. A data warehouse is specifically designed to support management decisions.

In simple terms, a data warehouse refers to a database that is maintained separately from an organization’s operational databases. Data warehouse systems enable for integration of several application systems. They provide data processing by supporting a solid platform of consolidated, historical information for analysis.

Data Warehouse queries are complicated because they contain the computation of huge groups of information at summarized levels. It can require the use of distinctive data organization, access, and implementation technique depends on multidimensional views.

A major reason for such a separation is to help boost the high implementation of both systems. An operational database is created and tuned from known functions and workloads, including indexing and hashing using primary keys, searching for specific records, and optimizing “canned” queries.

**A Multidimensional Data Model**

A multidimensional model is a technique, structure for data warehousing tools. Ralph Kimball firstly introduce the three-dimension model. A multidimensional model display data in the form of a data-cube. This data cube able data to be designed and display in multiple dimensions. This cube is defined through dimensions and facts.

The dimensions are the parts or entities regards which an organization saves data or records. The Dimensional model is basically developed to read, analyze numeric information like values, counts, weights, balances, summarize the data, etc. in a data warehouse. This model has a unique way of stored data for different advantages.

This model used in the relational model to minimize redundancy and also used to normalize ER models. The data is stored in three dimensions in an easy way, it is easily retrieved and generate reports.

A shop keeper may build a sales data warehouse in which he wants to keep a record of the stores’ sales for the dimension product, time, and location. These three dimensions keep the record of these things. For example, it keeps records of how many sales are made in one month. Each dimension allows keeping the records.

Each dimension has a table, this table is called dimension tablets table also specifies the further dimensions. The dimension table for products may contain attributes like product name, brand, price, and type.

The whole model is built around a specific theme. For example, we create it for the sales data warehouse. The theme is represented by the fact table.

The fact table consists of numerical measures. This may contain the name of fact and all measurements of related dimensions.

**Elements of the dimensional data model**

1. Dimension
2. Facts
3. Attribute
4. Dimension table
5. Fact table
6. Dimension

**Dimension**

Dimension provides information about a business process event. In other words, they specify who, what, whereof a fact. For example In the Sales dimension or business process, for the fact quarterly sales number, dimensions may contain

* Who –  Names of customer
* Where – Location/place
* What – Product Name

**Fact**

Facts are the numerical measurements or facts from your business process. For a Sales business process, a measurement may be sales number

**Attribute**

The characteristics of dimension are called attributes. For the location dimension, the attributes would be

1. Country
2. City
3. Zipcode
4. State

**Dimension table**

A dimension table consists of dimensions of a fact. The dimension table is used to join the fact table with a foreign key.
Dimension tables are de-normalized table .In a dimension table the Dimension Attributes as different columns.
It describes the characteristics of the facts with the help of their attributes

**Fact table**

It is known as the primary table in a dimensional model. This table consists of

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

**Data Warehouse Architecture**

The single tier Data Warehouse architecture is composed of a single hardware layer. This hardware layer is composed of a single hardware layer. There are three approaches to creating a data warehouse layer: Single tier, two-tier, and three-tier.

**Single-tier architecture:** A single-layer structure aimed at keeping data space minimal. This structure is rarely used in real life.

**Two-tier architecture:** Data warehouse is the aggregation of data in a format that is easy to transform and load into a database. Data warehouses can be implemented in a number of different ways, and it is important to pick the right one for your business needs. The most important thing to consider is scalability. If you want to store large amounts of data in a small amount of space, then you should consider using a data warehouse.

**Three-Tier Data Warehouse Architecture:** The Top, Middle, and Bottom Tiers of this Architecture of Data Warehouse are collectively referred to as the Top Tier.

1. The bottom tier of the Datawarehouse is a relational database system. This database system typically contains a relational database system. Back-end tools clean, transform, and load data into this layer.
2. A middle tier OLAP server is either ROLAP or MOLAP-based. It abstracts OLAP from the end user by serving as a middle tier OLAP server. Data warehouses that facilitate end-user interaction with the database and middle tier OLAP servers that abstract OLAP from the end user are known as middle tier OLAP servers.
3. The front-end client layer of the top-tier is important because it is the first point of interaction with the data. It is where data is presented to the end user, and decisions are made with the data. The front-end client layer of top-tier must work with real-time data and must be able to process data quickly. It is also important to work with data that is in a format that top-tier can understand and use. Typically, top-tier data is in a relational database format, but it could be a file or a stream. Top-tier data must be well-structured, must be validated, and must be structured in a way that allows for easier data profiling and analytics.



**Data Warehouse Architecture Properties**

A data warehouse system must meet the following architecture features:



* We sometimes wish to keep analytical and transactional processing as far away as possible.
* The scalability of the solution should be demonstrated by the ability to process a huge volume of data and stream it to different destinations, at high speed, in various formats. The data stream should be processed and presented in the required format, at the right time and location, with the minimum impact to the existing infrastructure. The data stream must be protected and managed with the highest level of confidentiality and integrity. The size of the data stream and the rate at which the data is being generated must be determined by the business requirements, and the available hardware and software resources must be utilized to the fullest extent possible.
* The architecture should be extensible; new functionality can be implemented in an existing service by extending the service’s APIs. For example, an insurance company could extend their customer service platform to provide a new feature that allows customers to obtain a personalized quote based on their preferences. Newer technologies, such as artificial intelligence, can be implemented in an existing service by extending the service’s APIs. For example, an insurance company could extend their customer service platform to provide a new feature that allows customers to obtain a personalized quote based on their preferences. Newer technologies, such as artificial intelligence, should be implemented in the core services; the core services can be extended for new business functions, such as customer relationship management.
* Data security is a critical aspect of the data governance strategy. Data security controls at the source include establishing data access controls and data encryption. Data security controls at the perimeter include data security policies and monitoring access to the data.
* It should be simple and straightforward, and users should be able to work with the data in an efficient and effective manner. Data Warehouse management should be easy to understand and implement. Data Warehouse management should not be complicated and difficult for beginners should not find their way into data warehouse management. It should be simple to use and easy to understand.

**Data Warehouse Back-End Tools and Utilities**

* **Data extraction:**
	+ get data from multiple, heterogeneous, and external sources
* **Data cleaning:**
	+ detect errors in the data and rectify them when possible
* **Data transformation:**
	+ convert data from legacy or host format to warehouse format
* **Load:**
	+ sort, summarize, consolidate, compute views, check integrity, and build indexes and partitions
* **Refresh**
	+ propagate the updates from the data sources to the warehouse

Metadata Repository

Metadata repository is an integral part of a data warehouse system. It has the following metadata −

* **Definition of data warehouse** − It includes the description of structure of data warehouse. The description is defined by schema, view, hierarchies, derived data definitions, and data mart locations and contents.
* **Business metadata** − It contains has the data ownership information, business definition, and changing policies.
* **Operational Metadata** − It includes currency of data and data lineage. Currency of data means whether the data is active, archived, or purged. Lineage of data means the history of data migrated and transformation applied on it.
* **Data for mapping from operational environment to data warehouse** − It includes the source databases and their contents, data extraction, data partition cleaning, transformation rules, data refresh and purging rules.
* **Algorithms for summarization** − It includes dimension algorithms, data on granularity, aggregation, summarizing, etc.

Types of OLAP Servers

We have four types of OLAP servers −

* Relational OLAP (ROLAP)
* Multidimensional OLAP (MOLAP)
* Hybrid OLAP (HOLAP)
* Specialized SQL Servers

Relational OLAP

ROLAP servers are placed between relational back-end server and client front-end tools. To store and manage warehouse data, ROLAP uses relational or extended-relational DBMS.

ROLAP includes the following −

* Implementation of aggregation navigation logic.
* Optimization for each DBMS back end.
* Additional tools and services.

Multidimensional OLAP

MOLAP uses array-based multidimensional storage engines for multidimensional views of data. With multidimensional data stores, the storage utilization may be low if the data set is sparse. Therefore, many MOLAP server use two levels of data storage representation to handle dense and sparse data sets.

Hybrid OLAP

Hybrid OLAP is a combination of both ROLAP and MOLAP. It offers higher scalability of ROLAP and faster computation of MOLAP. HOLAP servers allows to store the large data volumes of detailed information. The aggregations are stored separately in MOLAP store.

Data Warehouse Implementation

Data warehouses contain huge volumes of data. OLAP servers demand that decision support queries be acknowledged in the order of seconds. Thus, it is essential for data warehouse systems to provide highly effective cube computation techniques, access techniques, and query processing techniques.

Efficient Computation of Data Cubes

At the core of multidimensional data analysis is the efficient computation of aggregations across many sets of dimensions. In SQL terms, these aggregations are referred to as group-by’s. Each group-by can be represented by a cuboid, where the set of group-by’s forms a lattice of cuboids defining a data cube.

There are three choices for data cube materialization given a base cuboid −

* **No materialization** − It does not precompute any of the “nonbase” cuboids. This leads to computing expensive multidimensional aggregates on the fly, which can be extremely slow.
* **Full materialization** − It can Pre-compute all of the cuboids. The resulting lattice of computed cuboids is defined as the full cube. This choice typically requires huge amounts of memory space to store all of the precomputed cuboids.
* **Partial materialization** − It can selectively calculate a proper subset of the whole set of possible cuboids. Alternatively, it can calculate a subset of the cube, which includes only those cells that satisfy some user-specified criterion, including where the tuple count of each cell is following some threshold.

Indexing OLAP Data

It can support efficient data accessing, some data warehouse systems provide index structures and materialized views (using cuboids). The bitmap indexing approaches is famous in OLAP products because it enables fast searching in data cubes. The bitmap index is an alternative representation of the record ID (RID) list.

In the bitmap index for a given attribute, there is a distinct bit vector, Bv, for each value v in the domain of the attribute. If the domain of a given attribute includes n values, then n bits are required for each entry in the bitmap index (i.e., there are n bit vectors). If the attribute has the value v for a given row in the data table, then the bit defining that value is set to 1 in the corresponding row of the bitmap index. All other bits for that row are set to 0.

Efficient Processing of OLAP Queries

The goals of materializing cuboids and constructing OLAP index structures is to speed up query processing in data cubes.

* **Determine which operations should be performed on the available cuboids** − This contains transforming some selection, projection, roll-up (group-by), and drill-down operations represented in the query into the corresponding SQL and/or OLAP operations. For instance, slicing and dicing a data cube can correspond to selection and projection operations on a materialized cuboid.
* **Determine to which materialized cuboid(s) the relevant operations should be applied** − This contains identifying some materialized cuboids that can potentially be used to answer the query, pruning the following set using knowledge of “dominance” relationships between the cuboids, estimating the values of using the remaining materialized cuboids and choosing the cuboid with the minimum cost.

From online Analytical Processing to online analytical mining

 OLAM performs several data mining tasks, such as concept description, association, classification, forecasting, clustering, and time series analysis. It usually consists of several integrated data mining modules, so in this way OLAM is more complex than the OLAP mechanism.

 OLAP and OLAM data cubes are similar. However, OLAM analysis may require more powerful tools for building and accessing data cubes. That is why OLAM includes more dimensions with finer granularity, or includes a study based on the detection of multifunctional aggregations in a data cube, which requires more than OLAP analysis.