DATA WAREHOUSE

Introduction:

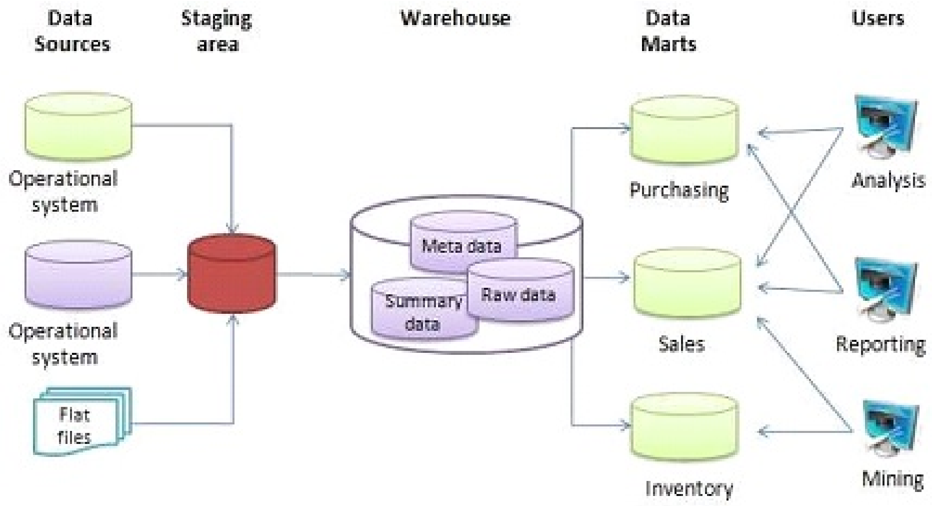
In computing, a data warehouse (DW or DWH), also known as an enterprise data warehouse (EDW), is a system used for reporting and data analysis and is considered a core component of business intelligence.Data warehouses are central repositories of integrated data from one or more disparate sources. They store current and historical data in one single place that are used for creating analytical reports for workers throughout the enterprise. This is beneficial for companies as it enables them to interrogate and draw insights from their data and make decisions.

Data warehouse:

Definition: A data warehouse is a type of data management system that is designed to enable and support business intelligence (BI) activities, especially analytics. Data warehouses are solely intended to perform queries and analysis and often contain large amounts of historical data

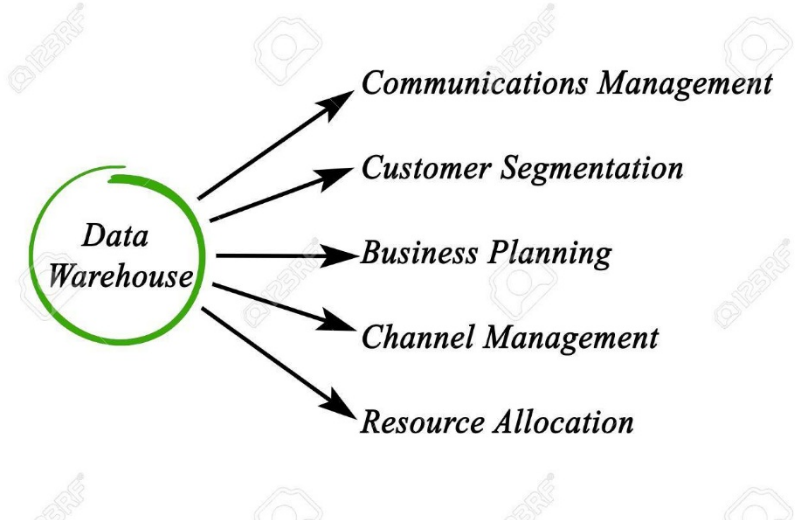
Overview:

A data warehouse is an enterprise system used for the analysis and reporting of structured and semi-structured data from multiple sources, such as point-of-sale transactions, marketing automation, customer relationship management, and more. A data warehouse is suited for ad hoc analysis as well custom reporting.



Function:

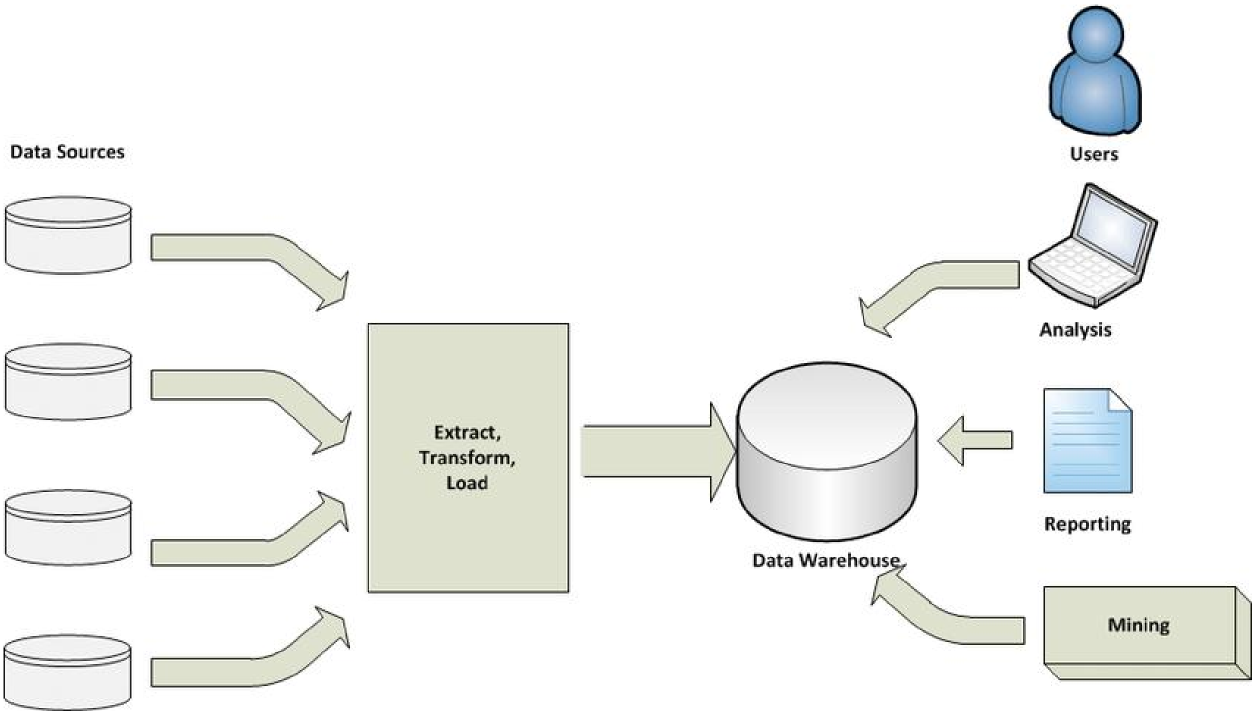
Data warehouses are solely intended to perform queries and analysis and often contain large amounts of historical data. The data within a data warehouse is usually derived from a wide range of sources such as application log files and transaction applications.



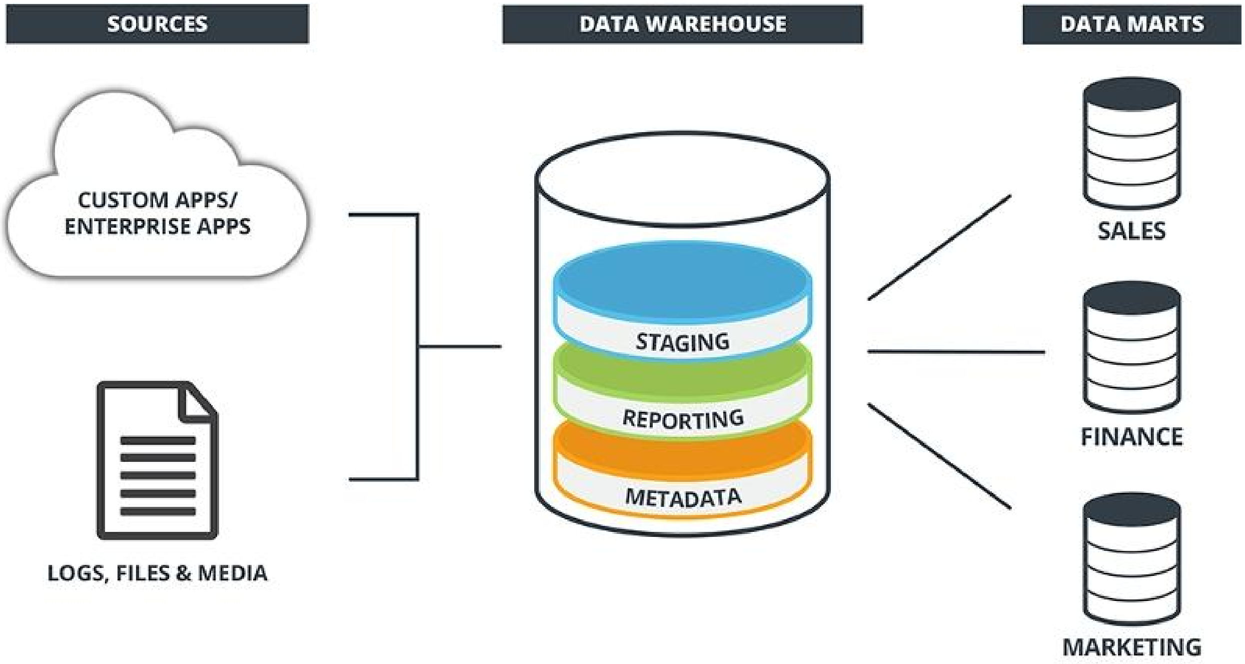
Architecture of Data warehouse:

The data stored in the warehouse is uploaded from the operational systems (such as marketing or sales). The data may pass through an operational data store and may require data cleansing[2] for additional operations to ensure data quality before it is used in the data warehouse for reporting.

Extract, transform, load (ETL) and extract, load, transform (ELT) are the two main approaches used to build a data warehouse



Architecture Layer:

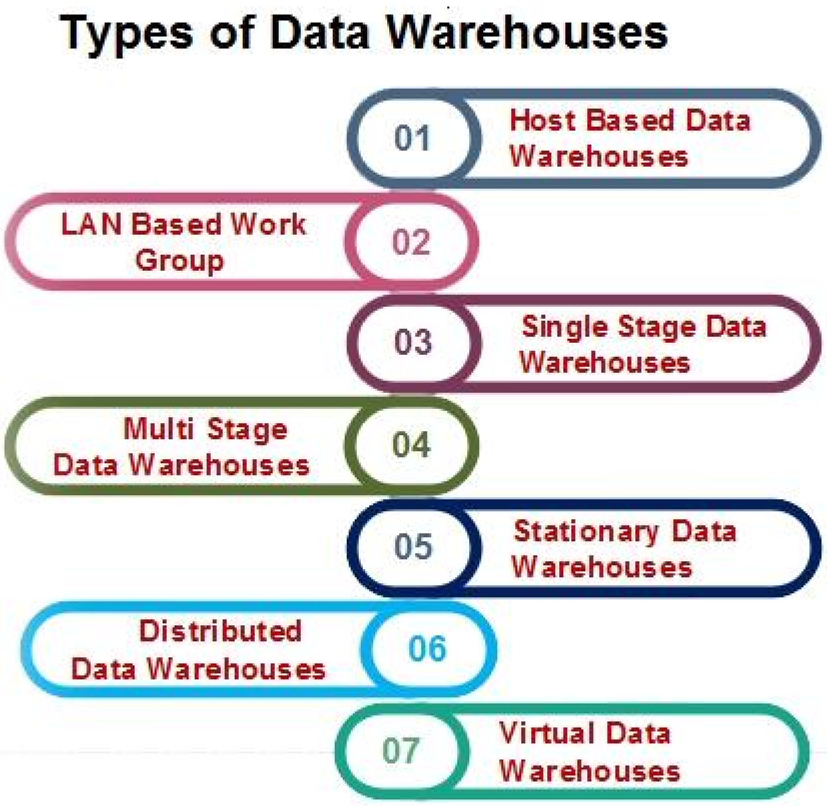


Three-tiered architecture: This architecture has three layers: the source, reconciled, and data warehouse layer. The reconciled layer in this architecture sits between the source and data warehouse layer and acts as a standard reference for an enterprise data model.

Types of Data warehouse:

There are three main types of data warehouse.

- Enterprise Data Warehouse (EDW) This type of warehouse serves as a key or central database that facilitates decision-support services throughout the enterprise. ...
- Operational Data Store (ODS) This type of data warehouse refreshes in real-time. ...
- Data Mart


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

Why data warehouse needed?

- Data warehousing improves the speed and efficiency of accessing different data sets and makes it easier for corporate decision-makers to derive insights that will guide the business and marketing strategies that set them apart from their competitors. Improve their bottom line.

Data warehouse in ETL?

Extract, transform, and load (ETL) is the process of combining data from multiple sources into a large, central repository called a data warehouse. ETL uses a set of business rules to clean and organize raw data and prepare it for storage, data analytics, and machine learning (ML).

Who uses data warehouse?

A warehouse is a building for storing goods. Warehouses are used by manufacturers, importers, exporters, wholesalers, transport businesses, customs, etc. They are usually large plain buildings in industrial parks on the outskirts of cities, towns, or villages

Where data warehouse used?

Data warehouses are relational environments that are used for data analysis, particularly of historical data. Organizations use data warehouses to discover patterns and relationships in their data that develop over time.

Design the Dimensional Model

- We need to design Dimensional Model to suit requirements of users which must address business needs and contains information which can be easily accessible. Design of model should be easily extensible according to future needs. This model design must supports OLAP cubes to provide "instantaneous" query results for analysts.

- Let us take a quick look at a few new terms and then we will identify/derive it for our requirement.

Dimension:

- The dimension is a master table composed of individual, non-overlapping data elements. The primary functions of dimensions are to provide filtering, grouping and labeling on your data. Dimension tables contain textual descriptions about the subjects of the business.

- Let me give you a glimpse on different types of dimensions available like confirmed dimension, Role Playing dimension, Degenerated dimension, Junk Dimension.

- Slowly changing dimension (SCD) specifies the way using which you are storing values of your dimension which is changing over a time and preserver the history. Different methods / types are available to store history of this change E.g. SCD1, SCD2, and SCD3 you can use as per your requirement.

- Let us identify dimensions related to the above case study.

Product, Customer, Store, Date, Time, Sales person

Measure

- A measure represents a column that contains quantifiable data, usually numeric, that can be aggregated. A measure is generally mapped to a column in a fact table. For your information, various types of measures are there. E.g. Additive, semi additive and Non additive.

- Let us define what will be the Measures in our case.

- Actual Cost, Total Sales, Quantity, Fact table record count

Fact Table

- Data in fact table are called measures (or dependent attributes), Fact table provides statistics for sales broken down by customer, salesperson, product, period and store dimensions. Fact table usually contains historical transactional entries of your live system, it is mainly made up of Foreign key column which references to various dimension and numeric measure values on which aggregation will be performed. Fact tables are of different types, E.g. Transactional, Cumulative and Snapshot.

- Let us identify what attributes should be there in our Fact Sales Table.

Foreign Key Column

- Sales Date key, Sales Time key, Invoice Number, Sales Person ID, Store ID, Customer ID

Measures

- Actual Cost, Total Sales, Quantity, Fact table record count

Create data warehouse in MYSQL:

Create database:

```
   Createdatabase Sales_DW
Go

Use Sales_DW
Go
```

Create dimension table:

```
Create table DimCustomer
(
CustomerID int primary key identity,
CustomerAltID varchar(10) not null,
CustomerName varchar(50),
Gender varchar(20)
)
Go
```

Fill the Customer dimension with sample Values:

```
SQL
Insert into DimCustomer(CustomerAltID,CustomerName,Gender)values
(‘IMI-001’,’Henry Ford’,’M’),
(‘IMI-002’,’Bill Gates’,’M’),
(‘IMI-003’,’Muskan Shaikh’,’F’),
(‘IMI-004’,’Richard Thrubin’,’M’),
(‘IMI-005’,’Emma Wattson’,’F’);
Go
```

```
 SQL
Insert into DimCustomer(CustomerAltID,CustomerName,Gender)values
(‘IMI-001’,’Henry Ford’,’M’),
(‘IMI-002’,’Bill Gates’,’M’),
(‘IMI-003’,’Muskan Shaikh’,’F’),
(‘IMI-004’,’Richard Thrubin’,’M’),
(‘IMI-005’,’Emma Wattson’,’F’);
Go
```

```
Create table DimProduct
(
ProductKey int primary key identity,
ProductAltKey varchar(10)not null,
ProductName varchar(100),
ProductActualCost money,
ProductSalesCost money

)
Go
```

Fact table:

```
    Create Table FactProductSales
(
TransactionId bigint primary key identity,
SalesInvoiceNumber int not null,
SalesDateKey int,
SalesTimeKey int,
SalesTimeAltKey int,
StoreID int not null,
CustomerID int not null,
ProductID int not null,
SalesPersonID int not null,
Quantity float,
SalesTotalCost money,
ProductActualCost money,
Deviation float
)
Go
```

Add relation between dimension table:

```
AlTER TABLE FactProductSales ADD CONSTRAINT _
FK_StoreID FOREIGN KEY (StoreID)REFERENCES DimStores(StoreID);
AlTER TABLE FactProductSales ADD CONSTRAINT _
FK_CustomerID FOREIGN KEY (CustomerID)REFERENCES Dimcustomer(CustomerID);
AlTER TABLE FactProductSales ADD CONSTRAINT _
FK_ProductKey FOREIGN KEY (ProductID)REFERENCES Dimproduct(ProductKey);
AlTER TABLE FactProductSales ADD CONSTRAINT _
FK_SalesPersonID FOREIGN KEY (SalesPersonID)REFERENCES Dimsalesperson(SalesPersonID);
Go
AlTER TABLE FactProductSales ADD CONSTRAINT _
FK_SalesDateKey FOREIGN KEY (SalesDateKey)REFERENCES DimDate(DateKey);
Go
AlTER TABLE FactProductSales ADD CONSTRAINT _
FK_SalesTimeKey FOREIGN KEY (SalesTimeKey)REFERENCES DimDate(TimeKey);
```

```
Go
```

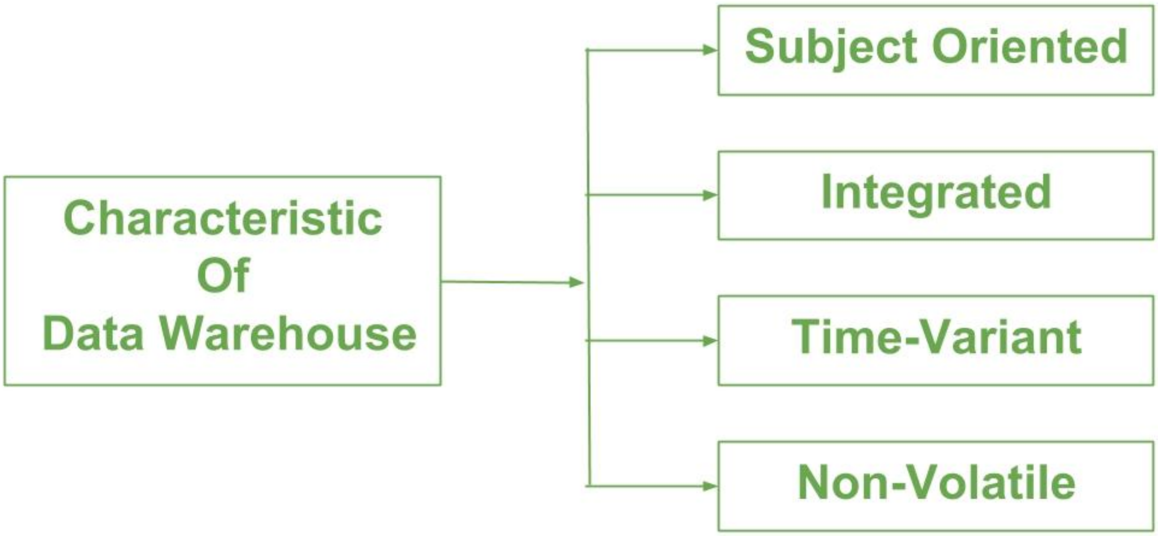
Populate your Fact table with historical transaction values of sales for previous day, with proper values of dimension key values:

```
      Insert into FactProductSales(SalesInvoiceNumber,SalesDateKey,_
SalesTimeKey,SalesTimeAltKey,StoreID,CustomerID,ProductID ,_
SalesPersonID,Quantity,ProductActualCost,SalesTotalCost,Deviation)values
--1-jan-2013
--SalesInvoiceNumber,SalesDateKey,SalesTimeKey,SalesTimeAltKey,_
StoreID,CustomerID,ProductID ,SalesPersonID,Quantity,_
ProductActualCost,SalesTotalCost,Deviation)
(1,20130101,44347,121907,1,1,1,1,2,11,13,2),
(1,20130101,44347,121907,1,1,2,1,1,22.50,24,1.5),
(1,20130101,44347,121907,1,1,3,1,1,42,43.5,1.5),

(2,20130101,44519,122159,1,2,3,1,1,42,43.5,1.5),
(2,20130101,44519,122159,1,2,4,1,3,54,60,6),

(3,20130101,52415,143335,1,3,2,2,2,11,13,2),
(3,20130101,52415,143335,1,3,3,2,1,42,43.5,1.5),
(3,20130101,52415,143335,1,3,4,2,3,54,60,6),
(3,20130101,52415,143335,1,3,5,2,1,135,139,4),
--2-jan-2013
--SalesInvoiceNumber,SalesDateKey,SalesTimeKey,SalesTimeAltKey,_
StoreID,CustomerID,ProductID
,SalesPersonID,Quantity,ProductActualCost,SalesTotalCost,Deviation)
(4,20130102,44347,121907,1,1,1,1,2,11,13,2),
(4,20130102,44347,121907,1,1,2,1,1,22.50,24,1.5),

(5,20130102,44519,122159,1,2,3,1,1,42,43.5,1.5),
(5,20130102,44519,122159,1,2,4,1,3,54,60,6),

(6,20130102,52415,143335,1,3,2,2,2,11,13,2),
(6,20130102,52415,143335,1,3,5,2,1,135,139,4),

(7,20130102,44347,121907,2,1,4,3,3,54,60,6),
(7,20130102,44347,121907,2,1,5,3,1,135,139,4),

--3-jan-2013
--SalesInvoiceNumber,SalesDateKey,SalesTimeKey,SalesTimeAltKey,StoreID,_
CustomerID,ProductID ,SalesPersonID,Quantity,ProductActualCost,SalesTotalCost,Deviation)
(8,20130103,59326,162846,1,1,3,1,2,84,87,3),
(8,20130103,59326,162846,1,1,4,1,3,54,60,3),
(9,20130103,59349,162909,1,2,1,1,1,5.5,6.5,1),
(9,20130103,59349,162909,1,2,2,1,1,22.50,24,1.5),

(10,20130103,67390,184310,1,3,1,2,2,11,13,2),
(10,20130103,67390,184310,1,3,4,2,3,54,60,6),

(11,20130103,74877,204757,2,1,2,3,1,5.5,6.5,1),
(11,20130103,74877,204757,2,1,3,3,1,42,43.5,1.5)
Go
```

Purpose of data warehouse:

- The primary purpose of a data warehouse is to enable companies to access and analyze all of their data to derive the most accurate business insights and forecasting models

Characteristics of data warehouse:



Unlike the operational systems, the data in the data warehouse revolves around the subjects of the enterprise. Subject orientation is not database normalization. Subject orientation can be really useful for decision-making. Gathering the required objects is called subject-oriented.

Integrated

The data found within the data warehouse is integrated. Since it comes from several operational systems, all inconsistencies must be removed. Consistencies include naming conventions, measurement of variables, encoding structures, physical attributes of data, and so forth.

Time-variant

While operational systems reflect current values as they support day-to-day operations, data warehouse data represents a long time horizon (up to 10 years) which means it stores mostly historical data. It is mainly meant for data mining and forecasting. (E.g. if a user is searching for a buying pattern of a specific customer, the user needs to look at data on the current and past purchases.

Nonvolatile

The data in the data warehouse is read-only, which means it cannot be updated, created, or deleted (unless there is a regulatory or statutory obligation to do so

Application:

A data warehouse is a type of data management system that is designed to enable and support business intelligence (BI) activities, especially analytics. Data warehouses are solely intended to perform queries and analysis and often contain large amounts of historical data.

Design method:

Bottom up :

data marts are first created to provide reporting and analytical capabilities for specific business processes. These data marts can then be integrated to create a comprehensive data warehouse. The data warehouse bus architecture is primarily an implementation of “the bus”, a collection of conformed dimensions and conformed facts, which are dimensions that are shared (in a specific way) between facts in two or more data marts.

Top-down design

The top-down approach is designed using a normalized enterprise data model. “Atomic” data, that is, data at the greatest level of detail, are stored in the data warehouse. Dimensional data marts containing data needed for specific business processes or specific departments are created from the data warehouse.

Hybrid design

Data warehouses often resemble the hub and spokes architecture. Legacy systems feeding the warehouse often include customer relationship management and enterprise resource planning, generating large amounts of data. To consolidate these various data models, and facilitate the extract transform load process, data warehouses often make use of an operational data store, the information from which is parsed into the actual data warehouse. To reduce data redundancy, larger systems often store the data in a normalized way. Data marts for specific reports can then be built on top of the data warehouse.

Example of data warehouse:

Here are some of the most well-known data warehouse platforms that companies choose to store and analyze their data: Google BigQuery. Snowflake. Amazon Redshift

OLAP:

Online analytical processing (OLAP) is characterized by a relatively low volume of transactions. Queries are often very complex and involve aggregations. For OLAP systems, response time is an effective measure. OLAP applications are widely used by Data Mining techniques. OLAP databases store aggregated, historical data in multi-dimensional schemas (usually star schemas). OLAP systems typically have a data latency of a few hours, as opposed to data marts, where latency is expected to be closer to one day. The OLAP approach is used to analyze multidimensional data from multiple sources and perspectives. The three basic operations in OLAP are Roll-up (Consolidation), Drill-down, and Slicing & Dicing.

OLTP:

Online transaction processing (OLTP) is characterized by a large number of short on-line transactions (INSERT, UPDATE, DELETE). OLTP systems emphasize very fast query processing and maintaining data integrity in multi-access environments. For OLTP systems, effectiveness is measured by the number of transactions per second. OLTP databases contain detailed and current data. The schema used to store transactional databases is the entity model (usually 3NF).[10] Normalization is the norm for data modeling techniques in this system.

Benefits of data warehouse:

- Provide a single common data model for all data of interest regardless of the data's source.
- Restructure the data so that it makes sense to the business users.
- Restructure the data so that it delivers excellent query performance, even for complex analytic queries, without impacting the operational systems.
- Add value to operational business applications, notably customer relationship management (CRM) systems.
- Make decision–support queries easier to write.
- Organize and disambiguate repetitive data.

Advantage:

- Storing large volumes of historical data from databases within a data warehouse allows for easy investigation of different time phases and trends, which can be highly impactful for your company. Thus, you can make superior corporate decisions concerning your business strategies with the right and real-time data.

Disadvantage:

Underestimation of data loading resources. Often, we fail to estimate the time needed to retrieve, clean, and upload the data to the warehouse. ...

Hidden problems in source systems. ...

Data homogenization.

Conclusion:

Conclusion. Data collection is an essential part of the research process, whether you’re conducting scientific experiments, market research, or surveys. The methods and tools used for data collection will vary depending on the research type, the sample size required, and the resources available.