Unit 1

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.

A Data Warehouse is a group of data specific to the entire organization, not only to a particular group of users.

It is not used for daily operations and transaction processing but used for making decisions.

A Data Warehouse can be viewed as a data system with the following attributes:

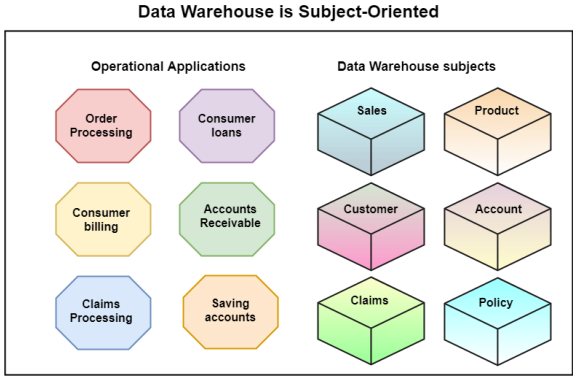
- It is a database designed for investigative tasks, using data from various applications.
- It supports a relatively small number of clients with relatively long interactions.
- It includes current and historical data to provide a historical perspective of information.
- Its usage is read-intensive.
- It contains a few large tables.

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

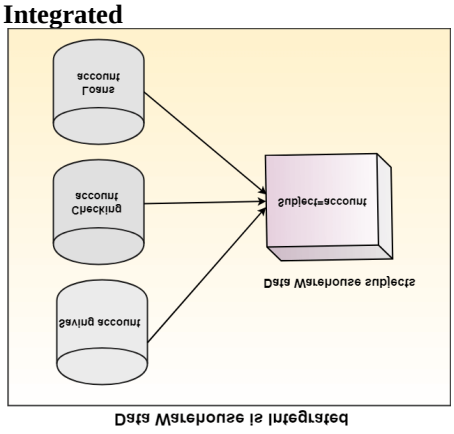
Characteristics of Data Warehouse

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.

The Advantages of Data Warehouse

Data warehouses will help you make better, more informed decisions for *many* reasons:

Improved business intelligence: When you integrate multiple sources, you make decisions based on ALL of your data.

- **Timely access to data:** Quickly access critical data in one centralized location.
- **Enhanced data quality and consistency**: Data throughout the organization is standardized and stored in the same format so all departments are making decisions based on uniform data.
- **Historical intelligence:** Because a data warehouse stores large amounts of historical data, you can identify trends through year-over-year and month-over-month analysis.
- **Quick query response times:** Most data warehouses are modeled, built, and optimized for read access, and that means fast report generation.
- **Data mining**: Explore “Big Data” to predict future trends.
- **Security**: A data warehouse makes provision easy by giving access to specific data to qualified end users, while excluding others.

- **Auditing**: Data stored appropriately in a data warehouse provides a complete audit trail of exactly when data was loaded and from which source(s).

- **Analytical tool support**: Analytical tools that offer drill-down ability work best when extracting data from a data warehouse.

- **Government regulation requirements**: With a data warehouse, it is easier to comply with Sarbanes-Oxley and other related regulations than with some transactional systems.

- **Metadata creation**: Descriptions of the data can be stored in the data warehouse so that users understand the data in the warehouse, making report creation much simpler.

- **Scalability**: If you have volumes of historical data that need consolidation, a data warehouse makes for easy access in a common place, with the ability to scale in the future.

- **Real-time performance**: A data warehouse can merge disparate data source with capabilities to preserve history as soon as the data is available.

Need of 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.

Goals of a Data Warehouse

- **Make an organization's information easily accessible**
- The contents of the data warehouse must be understandable and be intuitive and obvious to the business user. The contents of the data warehouse need to be labeled meaningfully. The tools that access the data warehouse must be simple and easy to use. They also must return query results to the user with minimal wait times.
- **Present the organization's information consistently**
- Consistent information means high-quality information. It means that all the data is accounted for and complete. Consistency also implies that common definitions for the contents of the data warehouse are available for users.
- **Be adaptive and resilient to change**
- We simply can't avoid change. User needs, business conditions, data, and technology are all subject to the shifting sands of time. The data warehouse must be designed to handle this inevitable change.
- **Be a secure bastion that protects our information assets**
- The data warehouse must effectively control access to the organization's confidential information.
- **serve as the foundation for improved decision making**
- The data warehouse must have the right data in it to support decision making.

Challenges of **Implementing Data Warehouses**:

1. Implementing a data warehouse is generally a massive effort that must be planned and executed according to established methods.
2. Construction, administration, and quality control are the significant operational issues which arises with data warehousing.

3. Some of the important and challenging consideration while implementing data warehouse are: the design, construction and implementation of the warehouse.
4. The building of an enterprise-wide warehouse in a large organization is a major undertaking.
5. Manual Data Processing can risk the correctness of the data being entered.
6. An intensive enterprise is the administration of a data warehouse, which is proportional to the complexity and size of the warehouse.
7. The complex nature of the administration should be understood by an organization that attempts to administer a data warehouse.
8. There must be a flexibility to accept and integrate analytics to streamline the business intelligence process.
9. To handle the evolutions, acquisition component and the warehouse's schema should be updated.
10. A significant issue in data warehousing is the quality control of data. The major concerns are: quality and consistency of data.

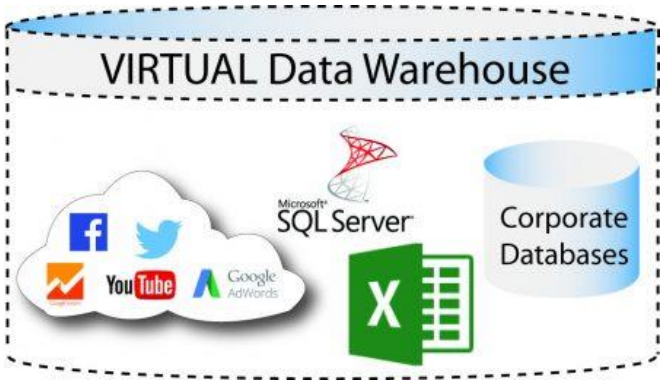
Types of Data Warehouse

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. Virtual data warehouse refers to a layer that sits on top of existing data bases and enables the user to query all of them as if they were one entity (although they are logically and physically separated).

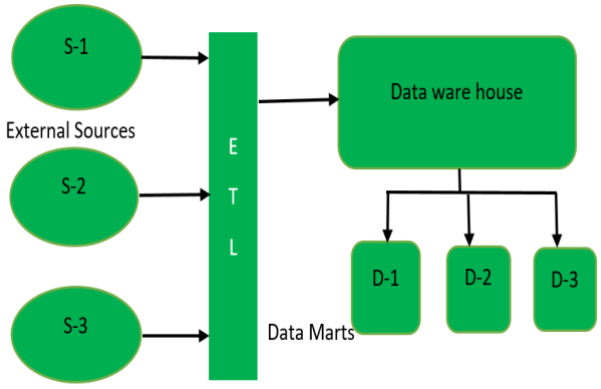


Data Virtualization makes all data, regardless of where it's located and regardless of what format it's in, look as if it is one place and in a consistent format. It provides access to data directly from one or more disparate data sources, without physically moving the data and provides it in such a manner that the technical aspects of location, structure, and access language are transparent to the analyst.

3. **Data mart** is such a storage component which is concerned on a specific department of an organisation. It is a subset of the data stored in the datawarehouse. Data mart is focused only on particular function of an organisation and it is maintained by single authority only, e.g.m finance, Marketing. Data Marts are small in size and are flexible.

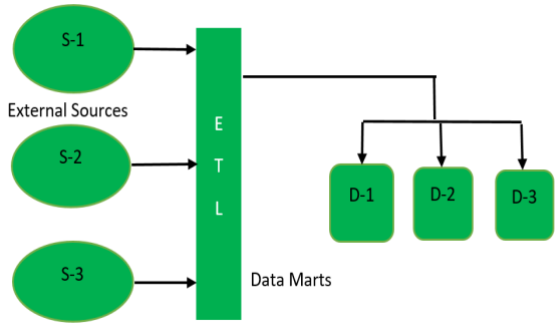
Types of Data Mart:
There are three types of data marts:

1. **Dependent Data Mart –**



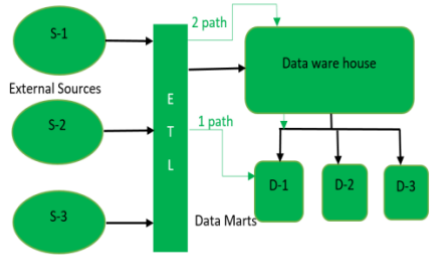
Dependent Data Mart is created by extracting the data from central repository, Datawarehouse. First data warehouse is created by extracting data (through ETL tool) from external sources and then data mart is created from data warehouse. Dependent data mart is created in top-down approach of datawarehouse architecture. This model of data mart is used by big organisations.

1. **Independent Data Mart –**



Independent Data Mart is created directly from external sources instead of data warehouse. First data mart is created by extracting data from external sources and then datawarehouse is created from the data present in data mart. Independent data mart is designed in bottom-up approach of datawarehouse architecture. This model of data mart is used by small organisations and is cost effective comparatively.

2. **Hybrid Data Mart –**



This type of Data Mart is created by extracting data from operational source or from data warehouse. 1Path reflects accessing data directly from external sources and 2Path reflects dependent data model of data mart.

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	Mostly hold only one subject area- for example, Sales figure.

	provide an integrated environment and coherent picture of the business at a point in time.	
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.

Facct data and dimension data

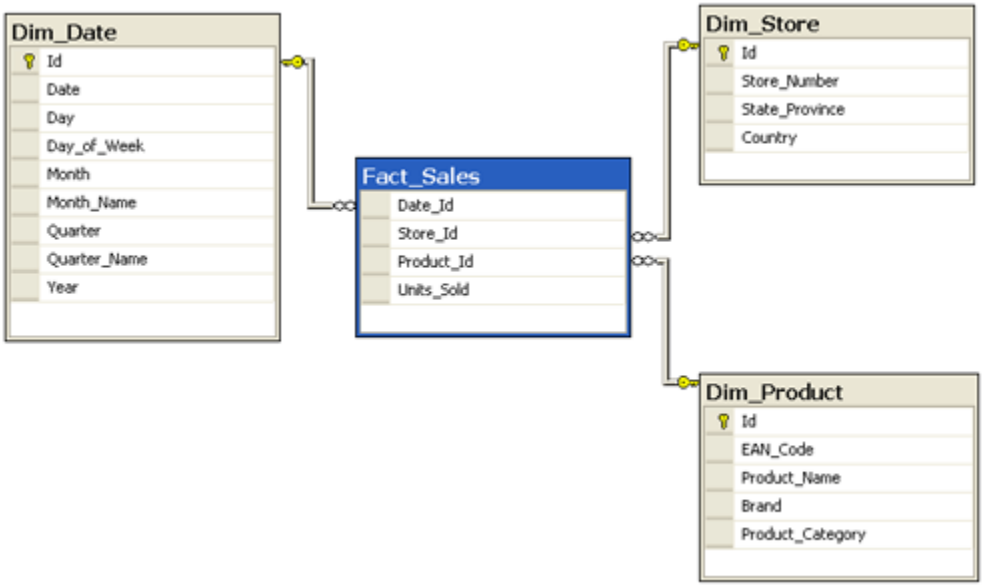
- Facts are business measurements. Facts are normally but not always numeric values that could be aggregated. e.g, a number of products sold per quarter.
- Dimensions are called contexts. Dimensions are business descriptors that specify the facts, for example, product name, brand, quarter, etc

A fact table works with dimension tables. A fact table holds the data to be analyzed, and a **dimension table** stores data about the ways in which the data in the fact table can be analyzed. Thus, the fact table consists of two types of columns. The foreign keys column allows joins with dimension tables, and the measures columns contain the data that is being analyzed.

Example of fact table

In the schema below, we have a fact table `FACT_SALES` that has a grain which gives us a number of units sold by date, by store and by product.

All other tables such as `DIM_DATE`, `DIM_STORE` and `DIM_PRODUCT` are dimensions tables. This schema is known as the star schema.



A fact table works with dimension tables. A fact table holds the data to be analyzed, and a dimension table stores data about the ways in which the data in the fact table can be analyzed. Thus, the fact table consists of two types of columns. The foreign keys column allows joins with dimension tables, and the measures columns contain the data that is being analyzed.

Suppose that a company sells products to customers. Every sale is a fact that happens, and the fact table is used to record these facts. For example:

Time ID	Product ID	Customer ID	Unit Sold
4	17	2	1
8	21	3	2
8	4	1	1

Now we can add a dimension table about customers:

Customer ID	Name	Gender	Income	Education	Region
1	Brian Edge	M	2	3	4
2	Fred Smith	M	3	5	1
3	Sally Jones	F	1	7	3

In this example, the customer ID column in the fact table is the foreign key that joins with the dimension table. By following the links, you can see that row 2 of the fact table records the fact that customer 3, Sally Jones, bought two items on day 8. The company would also have a product table and a time table to determine what Sally bought and exactly when

Measure types

Fact table can store different types of measures such as additive, non-additive, semi-additive.

- **Additive** – As its name implied, additive measures are measures which can be added to all dimensions.
- **Non-additive** – different from additive measures, non-additive measures are measures that cannot be added to all dimensions.
- **Semi-additive** – semi-additive measures are the measure that can be added to only some dimensions and not across other.

Types of fact tables

All fact tables are categorized by three most basic measurement events:

- **Transactional** – Transactional fact table is the most basic one that each grain associated with it indicated as “one row per line in a transaction”, e.g., every line item appears on an invoice. Transaction fact table stores data of the most detailed level, therefore, it has a high number of dimensions associated with.

- **Periodic snapshots** – Periodic snapshots fact table stores the data that is a snapshot in a period of time. The source data of periodic snapshots fact table is data from a transaction fact table where you choose a period to get the output.

- **Accumulating snapshots** – The accumulating snapshots fact table describes the activity of a business process that has clear beginning and end. This type of fact table, therefore, has multiple date columns to represent milestones in the process. A good example of accumulating snapshots fact table is processing of a material. As steps towards handling the material are finished, the corresponding record in the accumulating snapshots fact table gets updated.

Measure types

Fact table can store different types of measures such as additive, non-additive, semi-additive.

- **Additive** – As its name implied, additive measures are measures which can be added to all dimensions.
- **Non-additive** – different from additive measures, non-additive measures are measures that cannot be added to all dimensions.
- **Semi-additive** – semi-additive measures are the measure that can be added to only some dimensions and not across other.

Designing fact table steps

Here is overview of four steps to designing a fact table:

1. **Choosing business process to model** – The first step is to decide what business process to model by gathering and understanding business needs and available data
2. **Declare the grain** – by declaring a grain means describing exactly what a fact table record represents
3. **Choose the dimensions** – once grain of fact table is stated clearly, it is time to determine dimensions for the fact table.
4. **Identify facts** – identify carefully which facts will appear in the fact table.

Difference between Fact Table and Dimension Table:

S.NO	FACT TABLE	DIMENSION TABLE
1.	Fact table contains the measuring on the attributes of a dimension table.	Dimension table contains the attributes on that truth table calculates the metric.
2.	In fact table, There is less attributes than dimension table.	While in dimension table, There is more attributes than fact table.

3.	In fact table, There is more records than dimension table.	While in dimension table, There is less records than dimension table.
4.	Fact table forms a vertical table.	While dimension table forms a horizontal table.
5.	The attribute format of fact table is in numerical format and text format.	While the attribute format of dimension table is in text format.
6.	It comes after dimension table.	While it comes before fact table.
7.	The number of fact table is less than dimension table in a schema.	While the number of dimension is more than fact table in a schema.

What is Multidimensional schemas?

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).

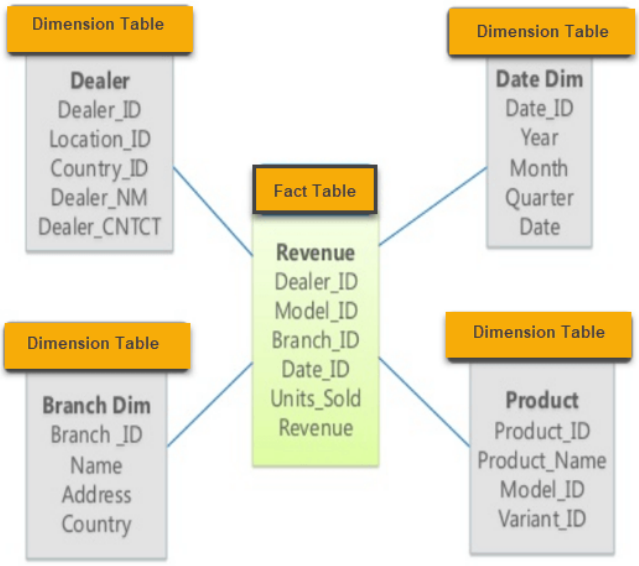
Types of Data Warehouse Schema:

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.



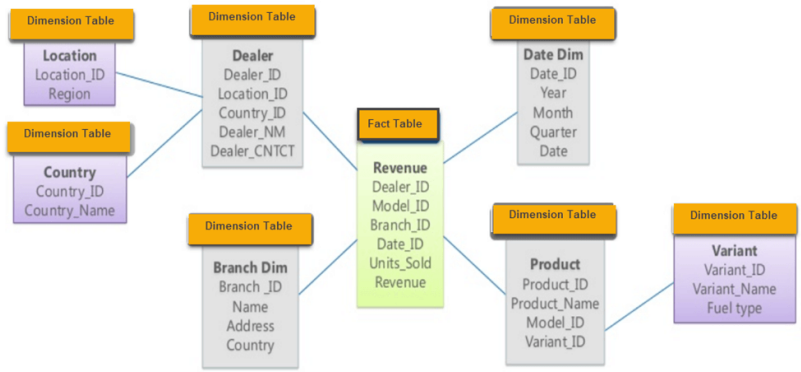
For example, as you can see in the above-given image that fact table is at the center which contains keys to every dimension table like Deal_ID, Model ID, Date_ID, Product_ID, Branch_ID & other attributes like Units sold and revenue.

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.

What is a Snowflake Schema?

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.The dimension tables are **normalized** which splits data into additional tables. In the following example, Country is further normalized into an individual table.



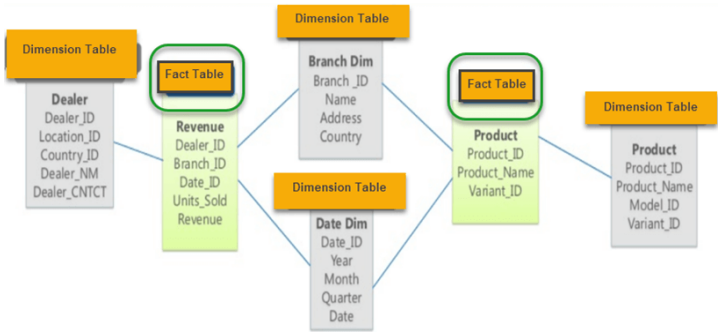
Characteristics of Snowflake Schema:

- 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.

What is a Galaxy schema?

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



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

1. Revenue
2. Product.

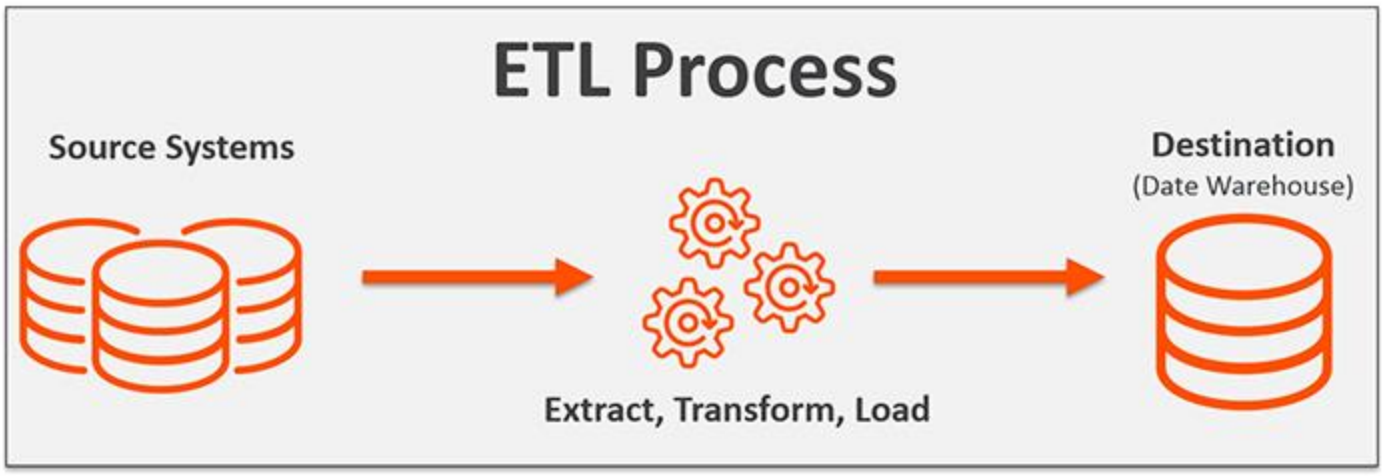
In Galaxy schema shares dimensions are called Conformed Dimensions.

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.

Pre-requisite phases: ETL

The following tasks are the main actions that happen in the ETL process:



Extraction of Data

The first step in ETL is extraction. During extraction, data is specifically identified and then taken from many different locations, referred to as the Source. The Source can be a variety of things, such as files, spreadsheets, database tables, a pipe, etc. It is not typically possible to pinpoint the exact subset of interest, so more data than necessary is extracted to ensure it covers everything needed. The volume of data extracted greatly varies and depends on business needs and requirements. Some extractions consist of hundreds of kilobytes all the way up to gigabytes. This is also the case for the timespan between two extractions; some may vary between days or hours to almost real-time.

Data extraction most typically occurs in one of three ways:

1. Update notification – the system notifies you when a record has been changed. This is typically referred to as the easiest method of extraction.
2. Incremental extraction – some systems cannot provide notifications for updates, so they identify when records have been modified and provide an extract on those specific records
3. Full extraction – some systems aren't able to identify when data has been changed at all, so the only way to get it out of the system is to reload it all. This is usually only recommended for small amounts of data as a last resort

Transformation of Data

The next step in the ETL process is transformation. After data is extracted, it must be physically transported to the target destination and converted into the appropriate format. This data transformation may include operations such as cleaning, joining, and validating data or generating calculated data based on existing values.

Whether the transformation takes place in the data warehouse or beforehand, there are both common and advanced transformation types that prepare data for analysis. Some of these include:

- Basic transformations:
  - Cleaning
  - Format revision
  - Restructuring
  - Deduplication
- Advanced transformations:
  - Filtering
  - Joining
  - Splitting
  - Derivation
  - Summarization
  - Integration

Loading Data

The final step in the ETL process involves loading the transformed data into the destination target. This target may be a database or a data warehouse. There are two primary methods for loading data into a warehouse: full load and incremental load. The full load method involves an entire data dump that occurs the first time the source is loaded into the warehouse. The incremental load, on the other hand, takes place at regular intervals. These intervals can be streaming increments (better for smaller data volumes) or batch increments (better for larger data volumes).