**Module 1 --> Data Warehousing**

* A data warehouse is a system that stores and analyzes data from multiple sources. It's a type of data management system that's used for business intelligence (BI) activities, especially analytics.
* A **Data Warehouse**is separate from DBMS, it stores a huge amount of data, which is typically collected from multiple heterogeneous sources like files, DBMS, etc.
* The goal of datawarehouse is to produce statistical results that may help in decision-making.
* For example, a college might want to see quick different results, like how the placement of CS students has improved over the last 10 years, in terms of salaries, counts, etc.

## Goals of Data Warehousing

* To help reporting and analysis of data
* 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.

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.

## Benefits of Data Warehouse

1. Understand business trends and make better decisions.
2. Data Warehouses are designed to process enormous amounts of data.
3. The structure of data warehouses is more accessible for end-users to navigate, understand, and query.
4. Queries that would be complex in many normalized databases could be easier to build and maintain in data warehouses.
5. Data warehousing is an efficient method to manage demand for lots of information from lots of users.
6. Data warehousing provide the capabilities to analyze a large amount of historical data.

## Characteristics of Data Warehouse

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

## 1.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. 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.

## 2.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 varies from traditional systems, where often only the most current file is kept.

## 4.Non-Volatile

The data warehouse is a physically separate data storage. 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.**

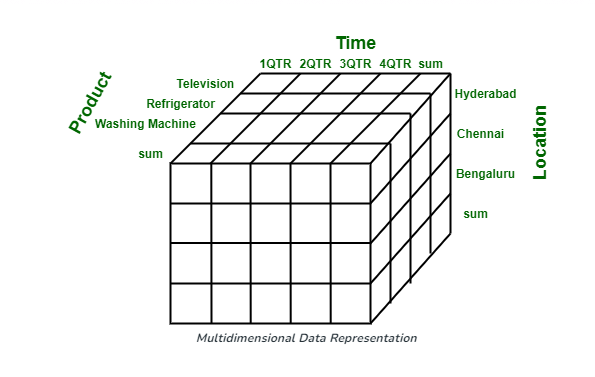
# What is Multi-Dimensional Data Model?

OLAP (online analytical processing) and data warehousing uses multi dimensional databases.

It is used to show multiple dimensions of the data to users.

It represents data in the form of data cubes. Data cubes allow to model and view the data from many dimensions and perspectives.

It is defined by dimensions and facts and is represented by a fact table. Facts are numerical measures and fact tables contain measures of the related dimensional tables or names of the facts.



### Working on a Multidimensional Data Model

The following stages should be followed for building a Multi Dimensional Data Model :

**Stage 1 : Assembling data from the client :** In first stage, a Multi Dimensional Data Model collects correct data in detail from the client.

**Stage 2 : Grouping different segments of the system :**In the second stage, the Multi Dimensional Data Model recognizes and classifies all the data to the respective section they belong to.

**Stage 3 : Noticing the different proportions :** In the third stage, it is the basis on which the design of the system is based. In this stage, the main factors are recognized according to the user’s point of view. These factors are also known as “Dimensions”.

**Stage 4 : Preparing the actual-time factors and their respective qualities :** In the fourth stage, the factors which are recognized in the previous step are used further for identifying the related qualities. These qualities are also known as **“attributes”** in the database.

**Stage 5 : Finding the actuality of factors which are listed previously and their qualities :**In the fifth stage,A Multi Dimensional Data Model separates and differentiates the actuality from the factors which are collected by it. These actually play a significant role in the arrangement of a Multi Dimensional Data Model.

**Stage 6 : Building the Schema to place the data, with respect to the information collected from the steps above :**In the sixth stage, on the basis of the data which was collected previously, a Schema is built.

### Advantages of Multi Dimensional Data Model

* A multi-dimensional data model is easy to handle.
* It is easy to maintain.
* Its performance is better than that of normal databases
* The representation of data is better than traditional databases.
* It is workable on complex systems and applications, contrary to the simple one-dimensional database systems.

### Disadvantages of Multi Dimensional Data Model

The following are the disadvantages of a Multi Dimensional Data Model :

* The multi-dimensional Data Model is slightly complicated in nature and it requires professionals to recognize and examine the data in the database.
* It is complicated in nature due to which the databases are generally dynamic in design.
* The path to achieving the end product is complicated most of the time.
* As the Multi Dimensional Data Model has complicated systems, databases have a large number of databases due to which the system is very insecure when there is a security break.

|  |  |
| --- | --- |
| **Database** | **Data Warehouse** |
| 1. It is used for Online Transactional Processing (OLTP) but can be used for other objectives such as Data Warehousing. This records the data from the clients for history. | 1. It is used for Online Analytical Processing (OLAP). This reads the historical information for the customers for business decisions. |
| 2. The tables and joins are complicated since they are normalized for RDBMS. This is done to reduce redundant files and to save storage space. | 2. The tables and joins are accessible since they are de-normalized. This is done to minimize the response time for analytical queries. |
| 3. Data is dynamic | 3. Data is largely static |
| 4. ****Entity:**** Relational modeling procedures are used for RDBMS database design. | 4. ****Data:**** Modeling approach are used for the Data Warehouse design. |
| 5. Optimized for write operations. | 5. Optimized for read operations. |
| 6. Performance is low for analysis queries. | 6. High performance for analytical queries. |
| 7. The database is the place where the data is taken as a base and managed to get available fast and efficient access. | 7. Data Warehouse is the place where the application data is handled for analysis and reporting objectives. |

# **Data Warehouse - Schemas**

A schema is defined as a logical description of database where fact and dimension tables are joined in a logical manner.

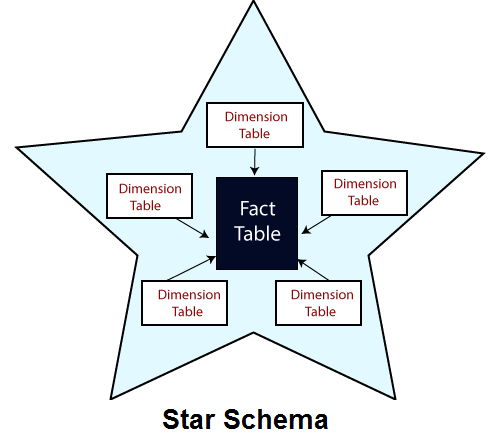
Data Warehouse is maintained in the form of

1. Star schema
2. Snow flake schema
3. Fact Constellation schema.

# Star Schema in Data Warehouse modeling

A star schema is a type of data modeling technique used in data warehousing to represent data in a structured and intuitive way.

In a star schema, data is organized into a central fact table that contains  ****facts**** and ****dimensions****.

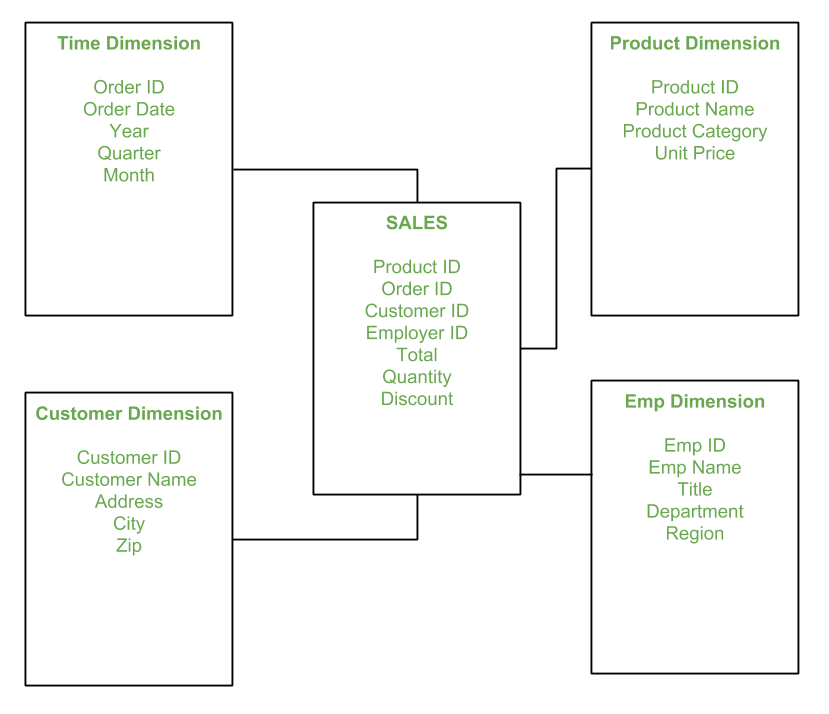


The fact table in a star schema contains the measures or metrics that are of interest to the user or organization. For example, in a sales data warehouse, the fact table might contain sales revenue, units sold, and profit margins. Each record in the fact table represents a specific event or transaction, such as a sale or order.

The dimension tables in a star schema contain the descriptive attributes of the measures in the fact table. These attributes are used to slice and dice the data in the fact table, allowing users to analyze the data from different perspectives. For example, in a sales data warehouse, the dimension tables might include product, customer, time, and location.

In a star schema, each dimension table is joined to the fact table through a foreign key relationship. This allows users to query the data in the fact table using attributes from the dimension tables. For example, a user might want to see sales revenue by product category, or by region and time period.

It is said to be star as its physical model resembles to the star shape having a fact table at its center and the dimension tables at its peripheral representing the star’s points. Below is an example to demonstrate the Star Schema:



In the above demonstration, SALES is a fact table having attributes i.e. (Product ID, Order ID, Customer ID, Employer ID, Total, Quantity, Discount) which references to the dimension tables. **Employee dimension table** contains the attributes: Emp ID, Emp Name, Title, Department and Region. *Product dimension table* contains the attributes: Product ID, Product Name, Product Category, Unit Price. *Customer dimension table* contains the attributes: Customer ID, Customer Name, Address, City, Zip. *Time dimension table* contains the attributes: Order ID, Order Date, Year, Quarter, Month.

****Advantages of Star Schema :****

1. It is easy to understand and query in a star schema.
2. The simple structure of the star schema allows for fast query response times and efficient use of database resources.
3. It can be easily extended by adding new dimension tables or measures to the fact table, making it a scalable and flexible solution for data warehousing.

**Disadvantages of Star Schema –**

1. Data integrity is not enforced well since in a highly de-normalized schema state.
2. Not flexible in terms if analytical needs as a normalized data model.
3. Star schemas don’t reinforce many-to-many relationships within business entities.

### Features of Star Schema:

1. **Central fact table:**The star schema revolves around a central fact table that contains the numerical data being analyzed. This table contains foreign keys to link to dimension tables.
2. **Dimension tables:** Dimension tables are tables that contain descriptive attributes about the data being analyzed. Each dimension table is linked to the fact table through a foreign key.
3. **Denormalized structure:**A star schema is denormalized, which means that redundancy is allowed in the schema design to improve query performance. This is because it is easier and faster to join a small number of tables than a large number of tables.
4. **Simple queries:**Star schema is designed to make queries simple and fast. Queries can be written in a straightforward manner by joining the fact table with the appropriate dimension tables.
5. **Aggregated data:**The numerical data in the fact table is usually aggregated at different levels of granularity, such as daily, weekly, or monthly. This allows for analysis at different levels of detail.
6. **Fast performance:** Star schema is designed for fast query performance. This is because the schema is denormalized and data is pre-aggregated, making queries faster and more efficient.
7. **Easy to understand:**The star schema is easy to understand and interpret, even for non-technical users. This is because the schema is designed to provide context to the numerical data through the use of dimension tables.

| **S.NO** | **Star Schema** | **Snowflake Schema** |
| --- | --- | --- |
| 1. | In [star schema](https://www.geeksforgeeks.org/star-schema-in-data-warehouse-modeling/), The fact tables and the dimension tables are contained. | While in [snowflake schema](https://www.geeksforgeeks.org/snowflake-schema-in-data-warehouse-model/), The fact tables, dimension tables as well as sub dimension tables are contained. |
| 2. | Star schema is a top-down model. | While it is a bottom-up model. |
| 3. | Star schema uses more space. | While it uses less space. |
| 4. | It takes less time for the execution of queries. | While it takes more time than star schema for the execution of queries. |
| 5. | In star schema, Normalization is not used. | While in this, Both normalization and denormalization are used. |
| 6. | It’s design is very simple. | While it’s design is complex. |
| 7. | The query complexity of star schema is low. | While the query complexity of snowflake schema is higher than star schema. |
| 8. | It’s understanding is very simple. | While it’s understanding is difficult. |
| 9. | It has less number of foreign keys. | While it has more number of foreign keys. |
| 10. | It has high data redundancy. | While it has low data redundancy. |

# Snowflake Schema

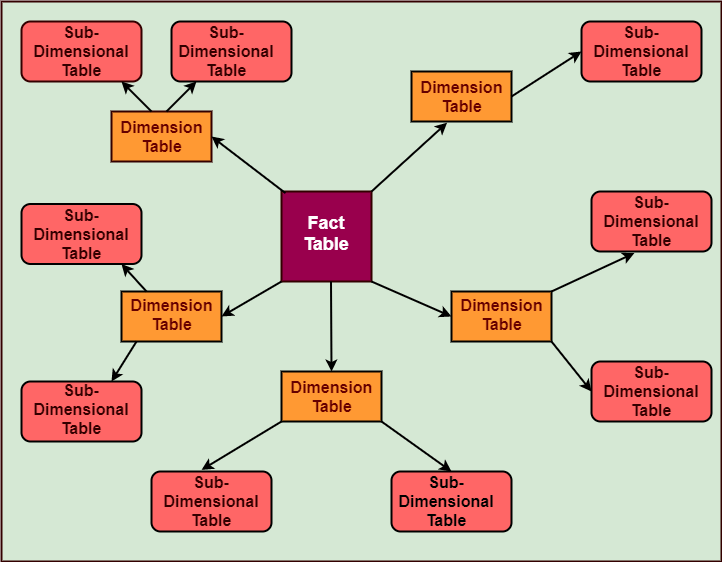
The snowflake schema is an expansion of the star schema where each point of the star explodes into more points. It is called snowflake schema because the diagram of snowflake schema resembles a snowflake.

****Snowflaking**** is a method of normalizing the dimension tables in a STAR schemas. When we normalize all the dimension tables entirely, the resultant structure resembles a snowflake with the fact table in the middle.

Snowflaking is used to develop the performance of specific queries. The schema is diagramed with each fact surrounded by its associated dimensions, and those dimensions are related to other dimensions, branching out into a snowflake pattern.

The snowflake schema consists of one fact table which is linked to many dimension tables, which can be linked to other dimension tables through a many-to-one relationship. Tables in a snowflake schema are generally normalized to the third normal form. Each dimension table performs exactly one level in a hierarchy.

A snowflake schemas can have any number of dimension, and each dimension can have any number of levels.

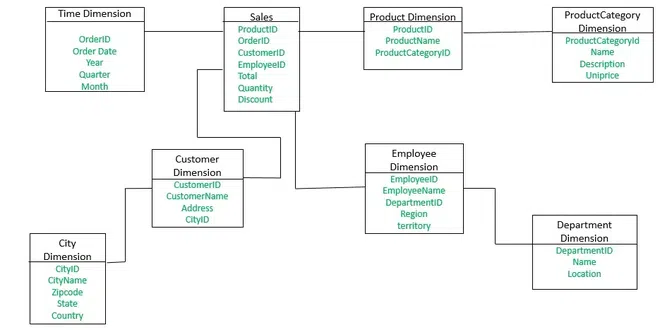


In snowflake, schema tables are normalized to delete redundancy. In snowflake dimension tables are damaged into multiple dimension tables.

A snowflake schema is designed for flexible querying across more complex dimensions and relationship. It is suitable for many to many and one to many relationships between dimension levels.

For Example, in a sales data warehouse, the product dimension table might be normalized into multiple related tables, such as product category, product subcategory, and product details. Each of these tables would be related to the product dimension table through a[foreign key](https://www.geeksforgeeks.org/postgresql-foreign-key/) relationship.

**Example:**



The **Employee**dimension table now contains the attributes: EmployeeID, EmployeeName, DepartmentID, Region, and Territory. The DepartmentID attribute links with the **Employee**table with the **Department**dimension table. The **Department**dimension is used to provide detail about each department, such as the Name and Location of the department. The **Customer**dimension table now contains the attributes: CustomerID, CustomerName, Address, and CityID. The CityID attributes link the **Customer**dimension table with the **City**dimension table. The **City**dimension table has details about each city such as city name, Zipcode, State, and Country.

## Advantage of Snowflake Schema

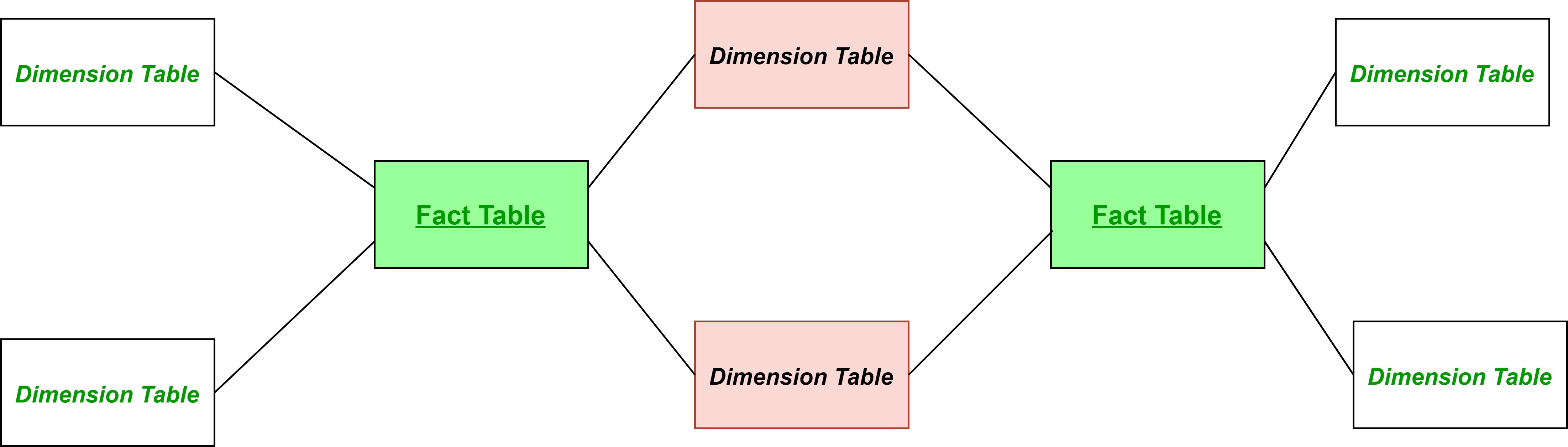
1. It provides greater scalability in the interrelationship between dimension levels and components.
2. No redundancy, so it is easier to maintain.
3. It provides structured data which reduces the problem of data integrity
4. It uses small disk space because data are highly structured.

## Disadvantage of Snowflake Schema

1. The primary disadvantage of the snowflake schema is the additional maintenance efforts required due to the increasing number of lookup tables. It is also known as a multi fact star schema.
2. There are more complex queries and hence, difficult to understand.
3. More tables more join so more query execution time.

# Fact Constellation Schema

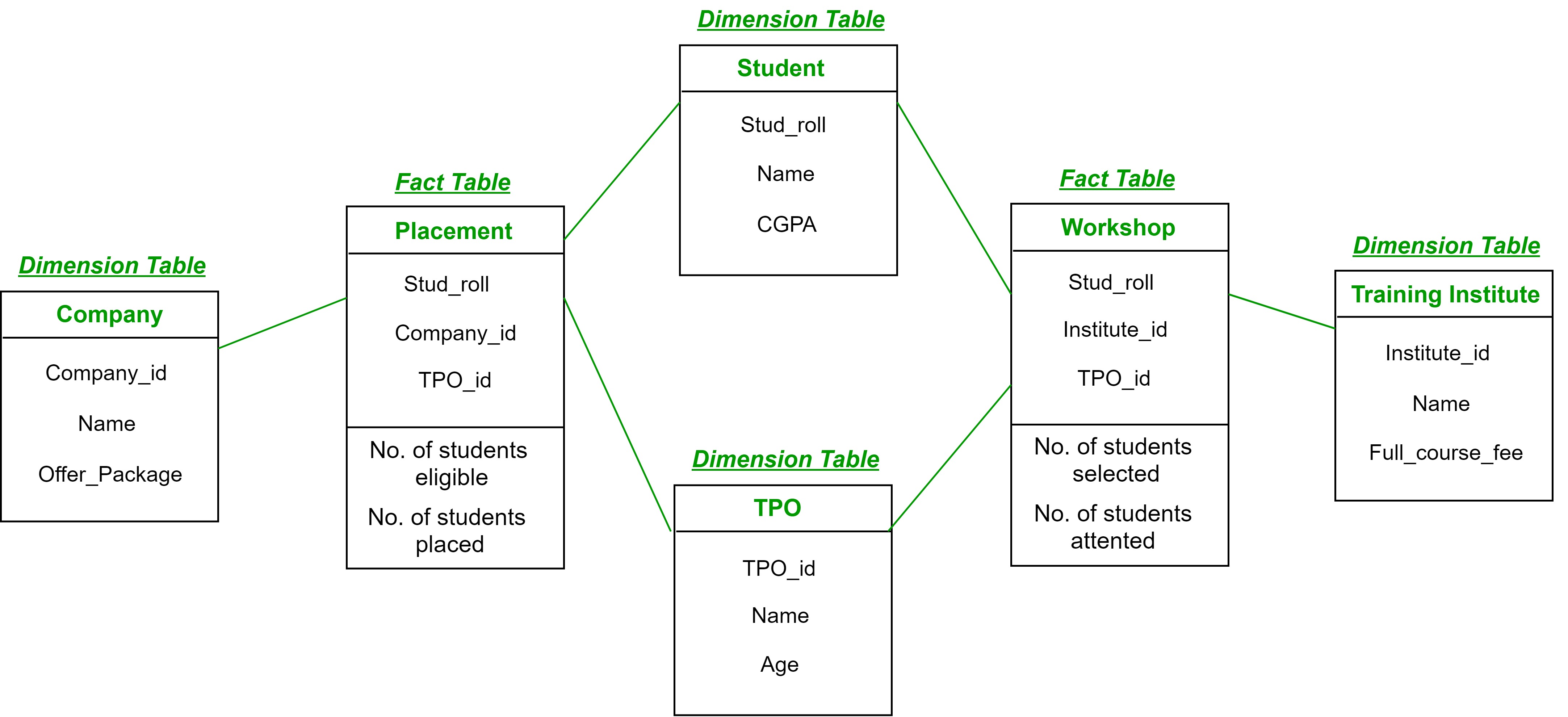
**Fact Constellation** is a schema for representing multidimensional model. It is a collection of multiple fact tables having some common dimension tables. It can be viewed as a collection of several star schemas and hence, also known as Galaxy schema. It is one of the widely used schema for Data warehouse designing and it is much more complex than star and snowflake schema. For complex systems, we require fact constellations.



**Figure –** General structure of Fact Constellation

Here, the pink coloured Dimension tables are the common ones among both the star schemas. Green coloured fact tables are the fact tables of their respective star schemas.

**Example:**

  
In above demonstration:

* **Placement**is a fact table having attributes: (Stud\_roll, Company\_id, TPO\_id) with facts: (Number of students eligible, Number of students placed).
* **Workshop** is a fact table having attributes: (Stud\_roll, Institute\_id, TPO\_id) with facts: (Number of students selected, Number of students attended the workshop).
* **Company** is a dimension table having attributes: (Company\_id, Name, Offer\_package).
* **Student** is a dimension table having attributes: (Student\_roll, Name, CGPA).
* **TPO** is a dimension table having attributes: (TPO\_id, Name, Age).
* **Training Institute** is a dimension table having attributes: (Institute\_id, Name, Full\_course\_fee).

So, there are two fact tables namely, **Placement** and **Workshop**which are part of two different star schemas having dimension tables – Company, Student and TPO in Star schema with fact table Placement and dimension tables – Training Institute, Student and TPO in Star schema with fact table Workshop. Both the star schema have two dimension tables common and hence, forming a fact constellation or galaxy schema.

**Advantage:**Provides a flexible schema.  
**Disadvantage:**It is much more complex and hence, hard to implement and maintain.

## **Defining data marts**

A [data mart](https://www.oracle.com/in/autonomous-database/departmental-data-warehouse/) is a simple form of data warehouse focused on a single subject or line of business.

With a data mart, teams can access data and gain insights faster, because they don’t have to spend time searching within a more complex data warehouse or manually aggregating data from different sources.

A data mart provides easier access to data required by a specific team of business within an organization. For example, if marketing team is looking for data to help improve campaign performance during the holiday season, sifting through and combining data scattered across multiple systems could prove costly in terms of time, accuracy, and ultimately, money.

Teams forced to locate data from various sources most often rely on spreadsheets to share this data and collaborate. This usually results in human errors, confusion, complex reconciliations, and multiple sources of truth—the so-called “spreadsheet nightmare.” Data marts have become popular as a centralized place where the necessary data is collected and organized before reports, dashboards, and visualizations are created.

Benefits of a Data Mart

Data Marts are built to enable business users to access the most relevant data in the shortest time. With its small size and focused design, data mart offers several benefits to the end-user, including:

* It contains data that is valuable to specific groups within an organization
* It is more Cost-effective to build a data mart than a data warehouse.
* It Allows simplified data access. Data marts contain a small subset of data, so users can easily retrieve data as needed compared to sifting through broader data set from a data warehouse.
* Quick access to data insights. Insights gained from a data mart impacts decisions at the department level. Teams can use these focused insights with specific goals in mind, resulting in faster business processes and higher productivity.
* Data mart needs less Implementation Time compared to data warehouse because you only need to focus on a small subset of data. Implementation tends to be more efficient and less time-consuming.
* It contains historical data, which helps data analysts to predict data trends.

## Types of Data Marts

There are three main types of data mart:

1. Dependent Data Mart - Built by drawing data directly from an existing data warehouse. All business data is stored in a centralized repository, and then a well-defined set of data is extracted when needed for analysis. The specific data set is aggregated from the warehouse, restructured, and populated into the data mart for querying. It is usually a logical view or physical subset of the data warehouse.

2. Independent Data Mart – stand-alone system, built without the use of a central data warehouse. Independent Data Marts are ideal for smaller units within an organization. Data is obtained from internal or external data sources, processed, loaded, and stored in the data mart until queried later for [business analytics.](https://www.simplilearn.com/business-analytics-as-a-lucrative-career-option-ccr60-article" \o "business analytics. " \t "https://www.simplilearn.com/_blank)

3. Hybrid Data Mart – it combines data from the data warehouse and other operational sources. A hybrid data mart is best suited for multiple database environments with a fast implementation turnaround. The system requires the least data cleansing effort.