DATA MINING (COCSC16)

UNIT 2

Data Warehouse

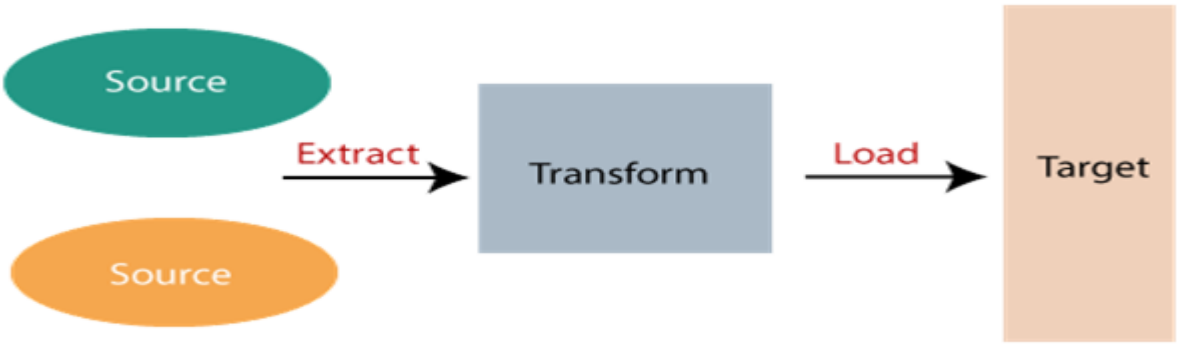
- Data Warehouse is a relational database management system (RDBMS) construct to meet the requirement of transaction processing systems. It can be loosely described as any centralized data repository which can be queried for business benefits.
- It is a database that stores information oriented to satisfy decision-making requests. It is a group of decision support technologies, targets to enabling the knowledge worker (executive, manager, and analyst) to make superior and higher decisions.
- So, Data Warehousing support architectures and tool for business executives to systematically organize, understand and use their information to make strategic decisions.
- Data Warehouse environment contains an extraction, transportation, and loading (ETL) solution, an online analytical processing (OLAP) engine, customer analysis tools, and other applications that handle the process of gathering information and delivering it to business users.

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.
- 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 Warehousing Process

- Data warehouse combines data from numerous sources which ensure the data quality, accuracy, and consistency. Data warehouse boosts system execution by separating analytics processing from transactional databases.
- Data flows into a data warehouse from different databases. A data warehouse works by sorting out data into a pattern that depicts the format and types of data. Query tools examine the data tables using patterns.

Data warehouses and **databases** both are relative data systems, but both are made to serve different purposes. A data warehouse is built to store a huge amount of historical data and empowers fast requests over all the data, typically using **Online Analytical Processing** (OLAP). A database is made to store current transactions and allow quick access to specific transactions for ongoing business processes, commonly known as **Online Transaction Processing** (OLTP).

Characteristics of Data Warehouse

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, 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.

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.

3. **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.

4. **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.

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.

Goals of Data Warehousing

- To help reporting as well as analysis
- 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. Since these people are non-technical, the data may be presented to them in an elementary form.

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.

5) **High response time:** Data warehouse has to be ready for somewhat unexpected loads and types of queries, which demands a significant degree of flexibility and quick response time.

Benefits of Data Warehouse

1. Understand business trends and make better forecasting decisions.
2. Data Warehouses are designed to perform well 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.

Differences between Data Mining and Data Warehousing:

Data Mining	Data Warehousing
Data mining is the process of determining data patterns.	A data warehouse is a database system designed for analytics.
Data mining is generally considered as the process of extracting useful data from a large set of data.	Data warehousing is the process of combining all the relevant data.
Business entrepreneurs carry data mining with the help of engineers.	Data warehousing is entirely carried out by the engineers.
In data mining, data is analyzed repeatedly.	In data warehousing, data is stored periodically.
Data mining uses pattern recognition techniques to identify patterns.	Data warehousing is the process of extracting and storing data that allow easier reporting.

One of the most amazing data mining techniques is the detection and identification of the unwanted errors that occur in the system.	One of the advantages of the data warehouse is its ability to update frequently. That is the reason why it is ideal for business entrepreneurs who want up to date with the latest stuff.
The data mining techniques are cost-efficient as compared to other statistical data applications.	The responsibility of the data warehouse is to simplify every type of business data.
The data mining techniques are not 100 percent accurate. It may lead to serious consequences in a certain condition.	In the data warehouse, there is a high possibility that the data required for analysis by the company may not be integrated into the warehouse. It can simply lead to loss of data.
Companies can benefit from this analytical tool by equipping suitable and accessible knowledge-based data.	Data warehouse stores a huge amount of historical data that helps users to analyze different periods and trends to make future predictions.

Difference between Database System and Data Warehouse:

Database System	**Data Warehouse**
It supports operational processes.	It supports analysis and performance reporting.
Capture and maintain the data.	Explore the data.
Current data.	Multiple years of history.
Data is balanced within the scope of this one system.	Data must be integrated and balanced from multiple system.
Data is updated when transaction occurs.	Data is updated on scheduled processes.
Data verification occurs when entry is done.	Data verification occurs after the fact.
100 MB to GB.	100 GB to TB.
ER based.	Star/Snowflake.

Database System	Data Warehouse
Application oriented.	Subject oriented.
Primitive and highly detailed.	Summarized and consolidated.
Flat relational.	Multidimensional.

Multi-tier Architecture of data warehouse

- There are four layers in multi-tier architecture. These are: **Data Source Layer, ETL Layer, Data Storage Layer, and Data Access Layer.**
- A data warehouse is a complex system. It requires multiple layers to handle the large amount of data involved. There is a need for a multi-level structure.
- Each layer of the system performs its specific function efficiently. A multi-tier architecture provides several benefits like better data quality, faster query response time, better data integration and scalability.

Data Source Layer:

- It is the first layer of a multi-tier architecture. It includes all sources of data that need to be integrated into the data warehouse. These sources can be databases, flat files or external sources such as social media platforms.
- The data source layer is responsible for collecting, validating and organizing the data before passing it on to the next layer.

ETL Layer

- This is the second layer of the multi-tier architecture. It is responsible for extracting data from data sources. It transforms it into a format suitable for a data warehouse.
- It also loads it into the data storage layer. This layer ensures the quality and consistency of the data loaded into the data warehouse.

Data Storage Layer

- This is the third layer of the multi-tier architecture. It is responsible for storing the data that has been transformed and loaded by the ETL Layer.
- This Layer can be divided into two sub-layers: the staging area and the data warehouse.

- The staging area is used to store the data temporarily before it is loaded into the data warehouse. The data warehouse is the final destination for the data and is used for reporting and analysis.

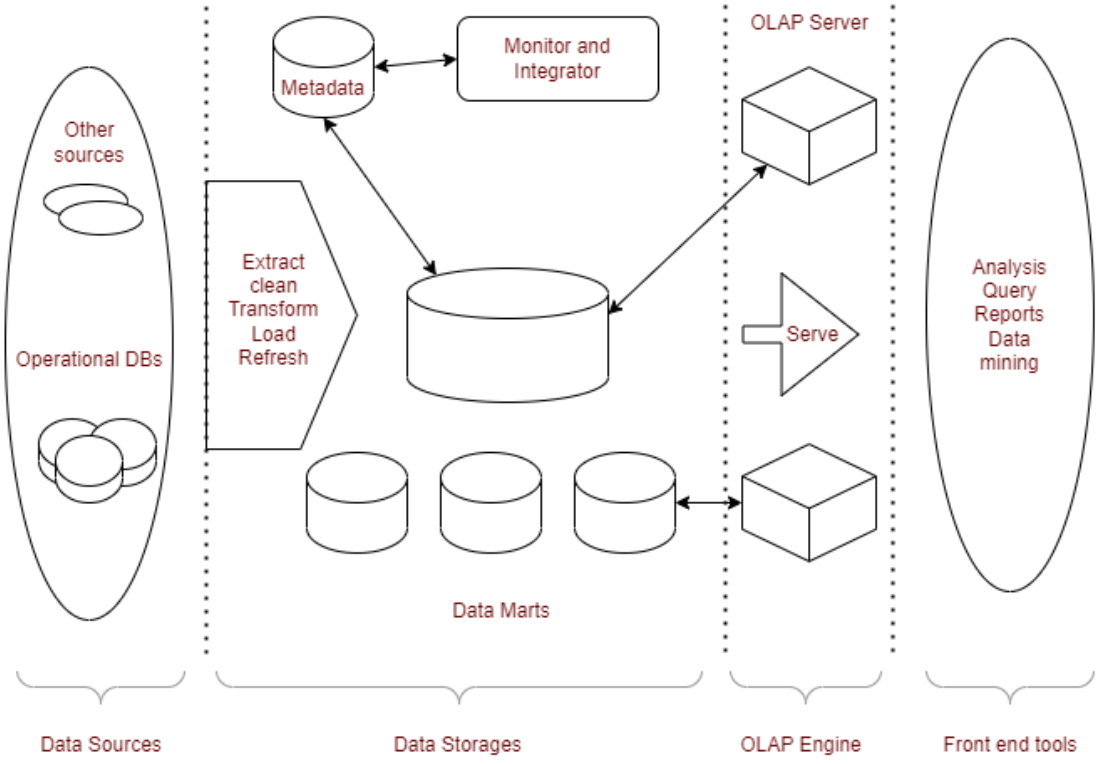
Data Access Layer

It is the fourth layer of the multi-tier architecture. It is responsible for providing users with accessibility to the data. This layer can be divided into two sub-layers −

- Presentation layer provides a user-friendly interface for users to access and analyze data.
- Application layer is responsible for managing the business logic and ensuring the security and integrity of the data.

Multi-Tier Data Warehouse Architecture Components:

Multi-Tier Data Warehouse Architecture has the following components: **Data Sources, Data Integration Layer, Staging Area, Data Warehouse Database, Data Mart, OLAP Cube, Front-End Tools, Metadata Repository.**



Multi-Tier Data Warehouse Architecture can be divided into three main parts. These are: Bottom, Middle and Top tier. These are explained as follows:

Bottom Tier (Data Sources and Data Storage)

This layer consists of Data Sources and Data Storage. It is usually implemented using a warehouse database server, such as RDBMS. Gateways, such as ODBC, OLE-DB, and JDBC, are used to extract data from operational and external sources.

Middle Tier

This layer is an OLAP server. OLAP server can be implemented using either Relational OLAP (ROLAP) model or Multidimensional OLAP (MOLAP) model. ROLAP is an extended relational DBMS. That maps operations from standard data to standard data. While MOLAP is a special-purpose server that directly implements multidimensional data and operations.

Top Tier

This layer is a front-end client layer. It has query and reporting tools, analysis tools, and data mining tools, such as trend analysis and prediction.

Advantages of Multi-tier Architecture

These are main advantages of Multi-Tier Architecture of Data Warehouse −

1. **Scalability**

Components can be added, deleted or updated according to the data warehouse's needs.

2. **Better Performance**

Several layers enable parallel and efficient processing for improved performance and reaction times.

3. **Modularity**

Modular design allows the creation, testing, and deployment of separate components.

4. **Security**

Applying security measures to various layers enhances the data warehouse's overall security.

5. **Improved Resource Management**

Different tiers can be tuned to use proper hardware resources, reducing expenses and increasing effectiveness.

6. **Easier Maintenance**

Individual components can be updated or maintained without affecting the entire data warehouse.

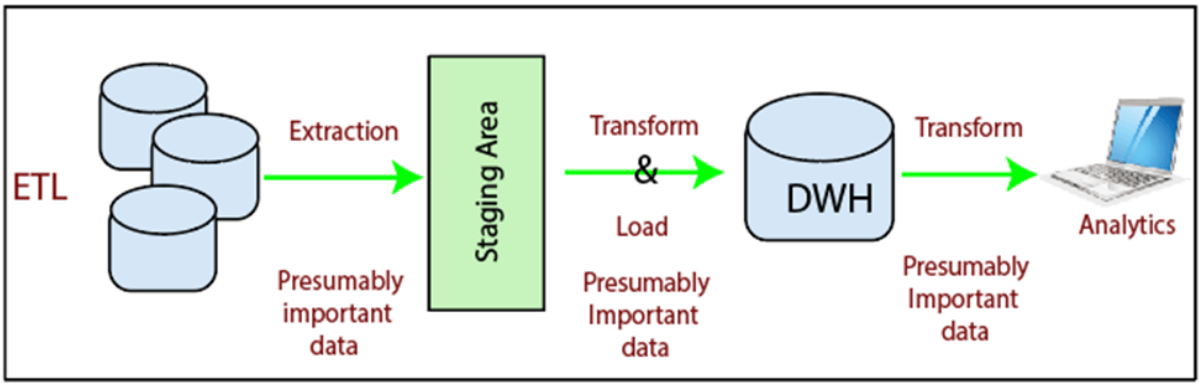
7. **Improved Reliability**

Multi-tier architecture offers redundancy and failover capabilities, enhancing the overall reliability of the data warehouse.

ETL (Extract, Transform, and Load) Process

What is ETL?

- The mechanism of extracting information from source systems and bringing it into the data warehouse is commonly called **ETL**, which stands for **Extraction, Transformation and Loading**.
- The ETL process requires active inputs from various stakeholders, including developers, analysts, testers, top executives and is technically challenging.
- To maintain its value as a tool for decision-makers, Data warehouse technique needs to change with business changes.
- ETL is a recurring method (daily, weekly, monthly) of a Data warehouse system and needs to be agile, automated, and well documented.



How ETL Works?

ETL consists of three separate phases:

Extraction

- Extraction is the operation of extracting information from a source system for further use in a data warehouse environment. This is the first stage of the ETL process.
- Extraction process is often one of the most time-consuming tasks in the ETL.
- The source systems might be complicated and poorly documented, and thus determining which data needs to be extracted can be difficult.

- The data has to be extracted several times in a periodic manner to supply all changed data to the warehouse and keep it up-to-date.

Cleansing

- The cleansing stage is crucial in a data warehouse technique because it is supposed to improve data quality. The primary data cleansing features found in ETL tools are rectification and homogenization.
- They use specific dictionaries to rectify typing mistakes and to recognize synonyms, as well as rule-based cleansing to enforce domain-specific rules and defines appropriate associations between values.

The following examples show the essential of data cleaning:

If an enterprise wishes to contact its users or its suppliers, a complete, accurate and up-to-date list of contact addresses, email addresses and telephone numbers must be available.

If a client or supplier calls, the staff responding should be quickly able to find the person in the enterprise database, but this need that the caller's name or his/her company name is listed in the database.

If a user appears in the databases with two or more slightly different names or different account numbers, it becomes difficult to update the customer's information.

Transformation

Transformation is the core of the reconciliation phase. It converts records from its operational source format into a particular data warehouse format. If we implement a three-layer architecture, this phase outputs our reconciled data layer.

The following points must be rectified in this phase:

- Loose texts may hide valuable information. For example, XYZ PVT Ltd does not explicitly show that this is a Limited Partnership company.
- Different formats can be used for individual data. For example, data can be saved as a string or as three integers.

Following are the main transformation processes aimed at populating the reconciled data layer:

- Conversion and normalization that operate on both storage formats and units of measure to make data uniform.
- Matching that associates equivalent fields in different sources.
- Selection that reduces the number of source fields and records.

Loading

The **Load** is the process of writing the data into the target database. During the load step, it is necessary to ensure that the load is performed correctly and with as little resources as possible.

Loading can be carried in two ways:

1. **Refresh:** Data Warehouse data is completely rewritten. This means that older file is replaced. Refresh is usually used in combination with static extraction to populate a data warehouse initially.
2. **Update:** Only those changes applied to source information are added to the Data Warehouse. An update is typically carried out without deleting or modifying preexisting data. This method is used in combination with incremental extraction to update data warehouses regularly.

Selecting an ETL Tool

- Selection of an appropriate ETL Tools is an important decision that has to be made in choosing the importance of an ODS or data warehousing application.
- The ETL tools are required to provide coordinated access to multiple data sources so that relevant data may be extracted from them.
- An ETL tool would generally contains tools for data cleansing, re-organization, transformations, aggregation, calculation and automatic loading of information into the object database.

What is Meta Data?

Metadata is data about the data or documentation about the information which is required by the users. In data warehousing, metadata is one of the essential aspects.

Metadata includes the following:

1. The location and descriptions of warehouse systems and components.
2. Names, definitions, structures, and content of data-warehouse and end-users views.
3. Identification of authoritative data sources.
4. Integration and transformation rules used to populate data.
5. Integration and transformation rules used to deliver information to end-user analytical tools.
6. Subscription information for information delivery to analysis subscribers.
7. Metrics used to analyze warehouses usage and performance.
8. Security authorizations, access control list, etc.

Metadata is used for building, maintaining, managing, and using the data warehouses. Metadata allow users access to help understand the content and find data.

Several examples of metadata are:

1. A library catalog may be considered metadata. The directory metadata consists of several predefined components representing specific attributes of a resource, and each item can have one or more values. These components could be the name of the author, the name of the document, the publisher's name, the publication date, and the methods to which it belongs.
2. The table of content and the index in a book may be treated metadata for the book.
3. Suppose we say that a data item about a person is 80. This must be defined by noting that it is the person's weight and the unit is kilograms. Therefore, (weight, kilograms) is the metadata about the data is 80.
4. Another example of metadata are data about the tables and figures in a report like this book. A table (which is a record) has a name (e.g., table titles), and there are column names of the tables that may be treated metadata. The figures also have titles or names.

Why is metadata necessary in a data warehouses?

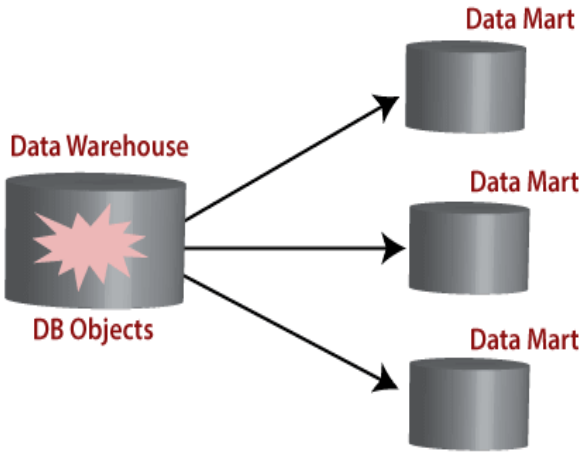
- First, it acts as the glue that links all parts of the data warehouses.
- Next, it provides information about the contents and structures to the developers.
- Finally, it opens the doors to the end-users and makes the contents recognizable in their terms.

Metadata is Like a **Nerve Center**. Various processes during the building and administering of the data warehouse generate parts of the data warehouse metadata. Another uses parts of metadata generated by one process. In the data warehouse, metadata assumes a key position and enables communication among various methods. It acts as a nerve centre in the data warehouse.

What is Data Mart?

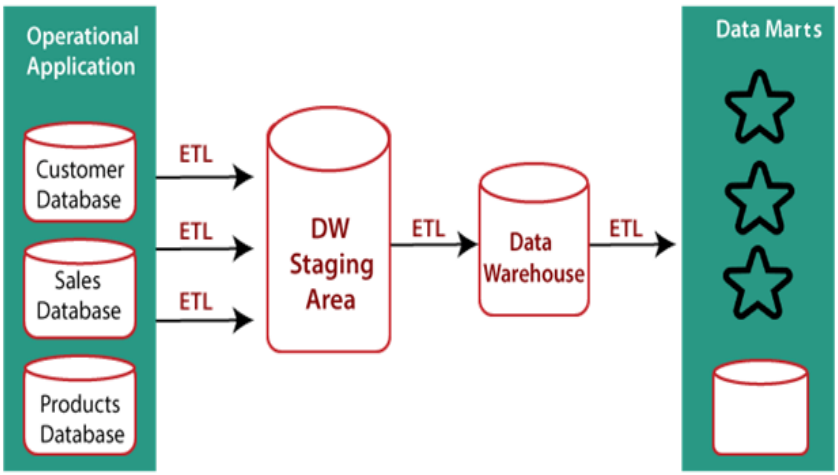
- A **Data Mart** is a subset of a directorial information store, generally oriented to a specific purpose or primary data subject which may be distributed to provide business needs.
- Data Marts are analytical record stores designed to focus on particular business functions for a specific community within an organization.
- Data marts are derived from subsets of data in a data warehouse, though in the bottom-up data warehouse design methodology, the data warehouse is created from the union of organizational data marts.
- The fundamental use of a data mart is **Business Intelligence (BI)** applications. **BI** is used to gather, store, access, and analyze record.

- It can be used by smaller businesses to utilize the data they have accumulated since it is less expensive than implementing a data warehouse.



Reasons for creating a data mart

- Creates collective data by a group of users
- Easy access to frequently needed data
- Ease of creation
- Improves end-user response time
- Lower cost than implementing a complete data warehouse
- Potential clients are more clearly defined than in a comprehensive data warehouse
- It contains only essential business data and is less cluttered.

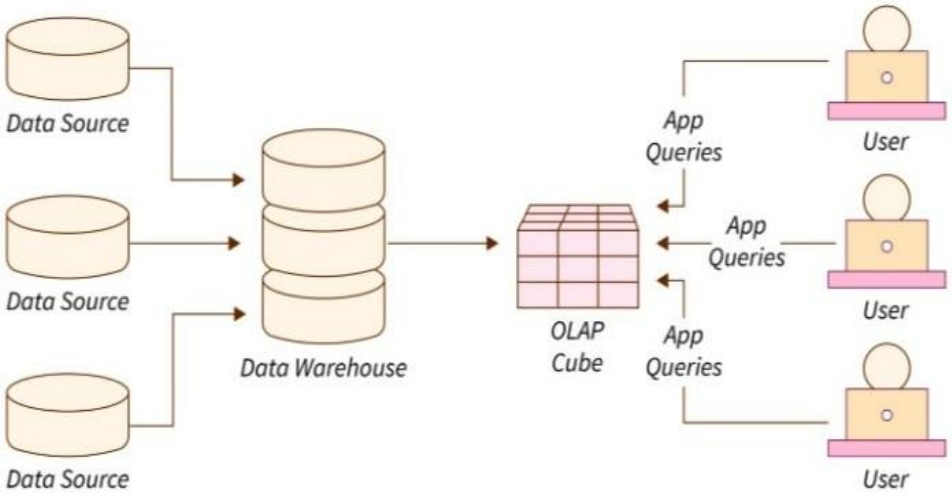


Difference between Data Warehouse and Data Mart

Data Warehouse	Data Mart
A Data Warehouse is a vast repository of information collected from various organizations or departments within a corporation.	A data mart is an only subtype of a Data Warehouses. It is architecture to meet the requirement of a specific user group.
It may hold multiple subject areas.	It holds only one subject area. For example, Finance or Sales.
It holds very detailed information.	It may hold more summarized data.
Works to integrate all data sources	It concentrates on integrating data from a given subject area or set of source systems.
In data warehousing, Fact constellation is used.	In Data Mart, Star Schema and Snowflake Schema are used.
It is a Centralized System. It is a Decentralized System.	
Data Warehousing is the data-oriented.	Data Marts is a project-oriented.

Online Analytical Processing Server (OLAP)

- Online Analytical Processing Server (OLAP) is a software. Users can analyze information from many different databases all at once.
- It uses a multidimensional data model where users can ask questions based on multiple dimensions at the same time.
- For example, a user could ask for sales data from Delhi in the year 2018. OLAP databases are split up into cubes, which are also called hyper-cubes.



The main characteristics of OLAP are as follows:

1. **Multidimensional conceptual view:** OLAP systems let business users have a dimensional and logical view of the data in the data warehouse. It helps in carrying slice and dice operations.
2. **Multi-User Support:** Since the OLAP techniques are shared, the OLAP operation should provide normal database operations, containing retrieval, update, adequacy control, integrity, and security.
3. **Accessibility:** OLAP acts as a mediator between data warehouses and front-end. The OLAP operations should be sitting between data sources (e.g., data warehouses) and an OLAP front-end.
4. **Storing OLAP results:** OLAP results are kept separate from data sources.
5. **Uniform documenting performance:** Increasing the number of dimensions or database size should not significantly degrade the reporting performance of the OLAP system.
6. OLAP provides for distinguishing between zero values and missing values so that aggregates are computed correctly.
7. OLAP system should ignore all missing values and compute correct aggregate values.
8. OLAP facilitate interactive query and complex analysis for the users.
9. OLAP allows users to drill down for greater details or roll up for aggregations of metrics along a single business dimension or across multiple dimensions.
10. OLAP provides the ability to perform intricate calculations and comparisons.
11. OLAP presents results in a number of meaningful ways, including charts and graphs.

Benefits of OLAP

1. OLAP helps managers in decision-making through the multidimensional record views that it is efficient in providing, thus increasing their productivity.
2. OLAP functions are self-sufficient owing to the inherent flexibility support to the organized databases.
3. It facilitates simulation of business models and problems, through extensive management of analysis-capabilities.
4. In conjunction with data warehouse, OLAP can be used to support a reduction in the application backlog, faster data retrieval, and reduction in query drag.

OLAP operations

These are used to analyze data in an OLAP cube. There are five basic operations:

Drill down

This makes the data more detailed by moving down the concept hierarchy or adding a new dimension. For example, in a cube showing sales data by Quarter, drilling down would show sales data by Month.

Roll up

This makes the data less detailed by climbing up the concept hierarchy or reducing dimensions. For example, in a cube showing sales data by City, rolling up would show sales data by Country.

Dice

This selects a sub-cube by choosing two or more dimensions and criteria. For example, in a cube showing sales data by Location, Time, and Item, dicing could select sales data for Delhi or Kolkata, in Q1 or Q2, for Cars or Buses.

Slice

This selects a single dimension and creates a new sub-cube. For example, in a cube showing sales data by Location, Time, and Item, slicing by Time would create a new sub-cube showing sales data for Q1.

Pivot

This rotates the current view to get a new representation. For example, after slicing by Time, pivoting could show the same data but with Location and Item as rows instead of columns.

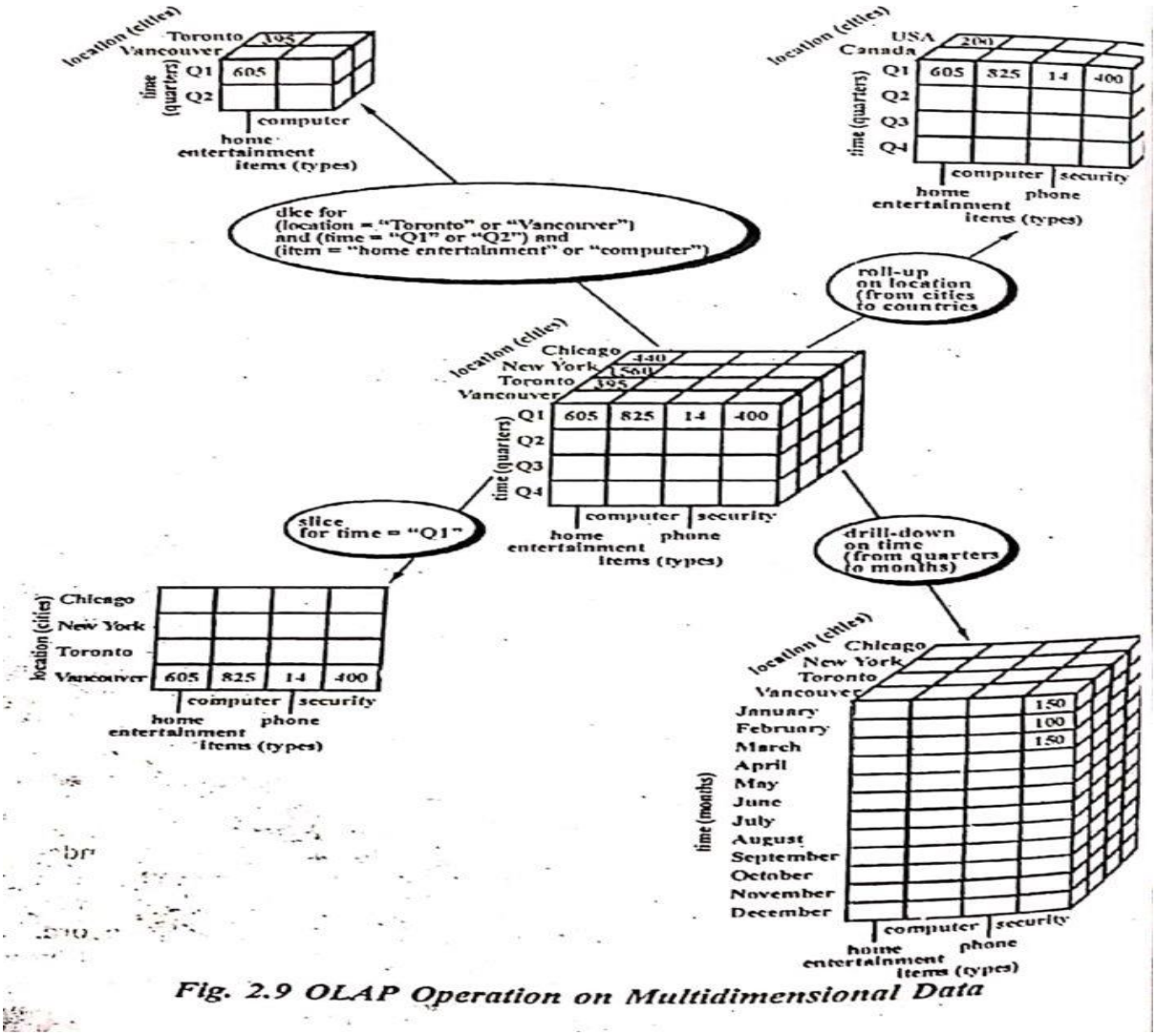
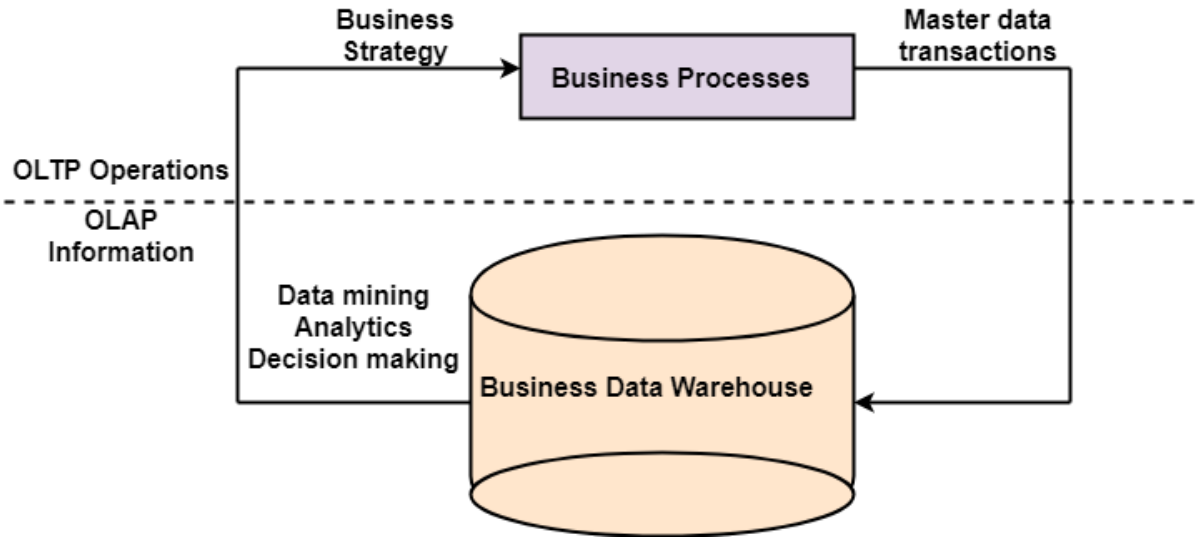


Fig. 2.9 OLAP Operation on Multidimensional Data

Difference between OLTP and OLAP

OLTP (On-Line Transaction Processing) is featured by a large number of short on-line transactions (INSERT, UPDATE, and DELETE). The primary significance of OLTP operations is put on very rapid query processing, maintaining record integrity in multi-access environments, and effectiveness consistent by the number of transactions per second. In the OLTP database, there is an accurate and current record, and schema used to save transactional database is the entity model (usually 3NF).

OLAP (On-line Analytical Processing) is represented by a relatively low volume of transactions. Queries are very difficult and involve aggregations. For OLAP operations, response time is an effectiveness measure. OLAP applications are generally used by Data Mining techniques. In OLAP database there is aggregated, historical information, stored in multi-dimensional schemas (generally star schema).



Following are the difference between OLAP and OLTP system.

1) Users: OLTP systems are designed for office worker while the OLAP systems are designed for decision-makers. Therefore, while an OLTP method may be accessed by hundreds or even thousands of clients in a huge enterprise, an OLAP system is suitable to be accessed only by a select class of manager and may be used only by dozens of users.

Functions: OLTP systems are mission-critical. They provide day-to-day operations of an enterprise and are largely performance and availability driven. These operations carry out simple repetitive operations. **OLAP** systems are management-critical to support the decision of enterprise support tasks using detailed investigation.

3) Nature: Although **SQL** queries return a set of data, OLTP methods are designed to step one record at the time, for example, a data related to the user who may be on the phone or in the store. OLAP system is not designed to deal with individual customer records. Instead, they include queries that deal with many data at a time and provide summary or aggregate information to a manager. OLAP applications include data stored in a data warehouses that have been extracted from many tables and possibly from more than one enterprise database.

4) Design: OLTP database operations are designed to be application-oriented while **OLAP** operations are designed to be subject-oriented. OLTP systems view the enterprise record as a collection of tables (possibly based on an entity-relationship model). **OLAP** operations view enterprise information as multidimensional).

5) Data: OLTP systems usually deal only with the current status of data. For example, a record about an employee who left three years ago may not be feasible on the Human Resources System. The old data may have been achieved on some type of stable storage media and may not be accessible online. On the other hand, OLAP systems needed historical data over several years since trends are often essential in decision making.

6) Kind of use: OLTP methods are used for reading and writing operations while OLAP methods usually do not update the data.

7) View: An **OLTP** system focuses primarily on the current data within an enterprise or department, which does not refer to historical data or data in various organizations. In contrast,

an **OLAP** system spans multiple versions of a database schema, due to the evolutionary process of an organization. OLAP system also deals with information that originates from different organizations, integrating information from many data stores. Because of their huge volume, these are stored on multiple storage media.

8) Access Patterns: The access pattern of an OLTP system consist primarily of short, atomic transactions. Such a system needed concurrency control and recovery techniques. However, access to OLAP systems is mostly read-only operations because these data warehouses store historical information.

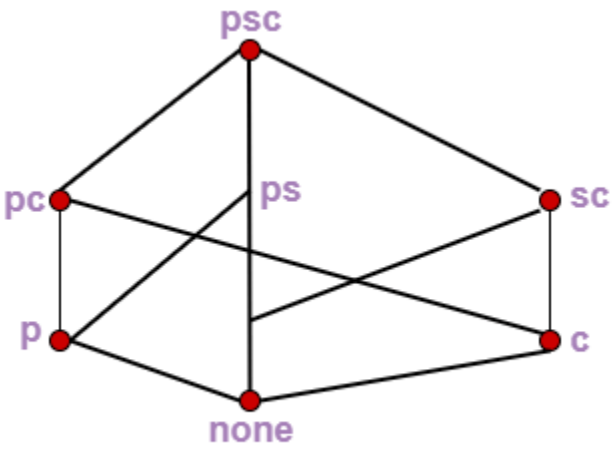
The biggest difference between an OLTP and OLAP system is the amount of data analysed in a single transaction. Whereas an OLTP handles many concurrent customers and queries touching only a single data or limited collection of records at a time, an OLAP system must have the efficiency to operate on millions of data to answer a single query.

What is Multi-Dimensional Data Model?

- A multidimensional model views data in the form of a data-cube. A data cube enables data to be modelled and viewed in multiple dimensions. It is defined by dimensions and facts.
- The dimensions are the perspectives or entities concerning which an organization keeps records. For example, a shop may create a sales data warehouse to keep records of the store's sales for the dimension time, item, and location.
- These dimensions allow to keep track of things, for example, monthly sales of items and the locations at which the items were sold.
- Each dimension has a table related to it, called a dimensional table, which describes the dimension further. For example, a dimensional table for an item may contain the attributes item_name, brand, and type.
- **A multidimensional data model is organized around a central theme, for example, sales. This theme is represented by a fact table. Facts are numerical measures. The fact table contains the names of the facts or measures of the related dimensional tables.**

What is Data Cube?

- When data is grouped or combined in multidimensional matrices called Data Cubes. The data cube method has a few alternative names or a few variants, such as "Multidimensional databases," "materialized views," and "OLAP (On-Line Analytical Processing)."
- The general idea of this approach is to materialize certain expensive computations that are frequently inquired.
- **For example,** a relation with the schema sales (part, supplier, customer, and sale-price) can be materialized into a set of eight views as shown in fig, where **psc** indicates a view consisting of aggregate function value (such as total-sales) computed by grouping three attributes part, supplier, and customer, **p** indicates a view composed of the corresponding aggregate function values calculated by grouping part alone, etc.

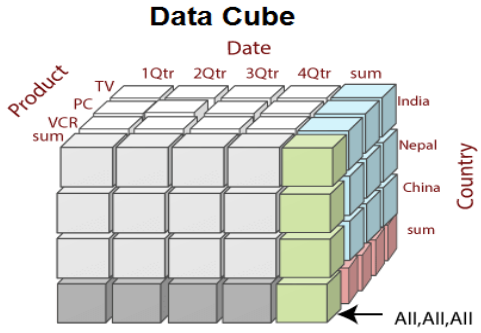


Eight views of data cubes for sales information.

- A data cube is created from a subset of attributes in the database. Specific attributes are chosen to be measure attributes, i.e., the attributes whose values are of interest.
- Other attributes are selected as dimensions or functional attributes. The measure attributes are aggregated according to the dimensions.

For example, XYZ may create a sales data warehouse to keep records of the store's sales for the dimensions time, item, branch, and location. These dimensions enable the store to keep track of things like monthly sales of items, and the branches and locations at which the items were sold. Each dimension may have a table identify with it, known as a dimensional table, which describes the dimensions. For example, a dimension table for items may contain the attributes item_name, brand, and type.

- Data cube method is an interesting technique with many applications. Data cubes could be sparse in many cases because not every cell in each dimension may have corresponding data in the database.
- If a query contains constants at even lower levels than those provided in a data cube, it is not clear how to make the best use of the precomputed results stored in the data cube.
- The model view data in the form of a data cube. OLAP tools are based on the multidimensional data model. Data cubes usually model n-dimensional data.
- A data cube enables data to be modeled and viewed in multiple dimensions. A multidimensional data model is organized around a central theme, like sales and transactions.
- A fact table represents this theme. Facts are numerical measures. Thus, the fact table contains measure (such as Rs_sold) and keys to each of the related dimensional tables.



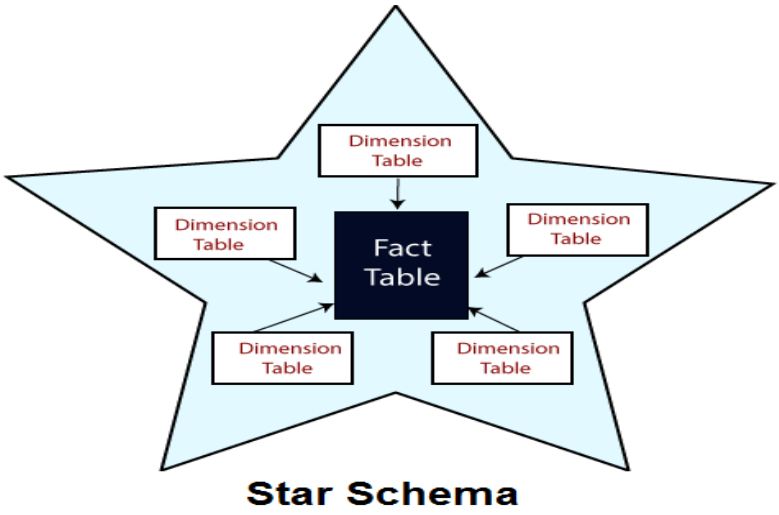
Example: In the **2-D representation**, we will look at the All Electronics sales data for **items sold per quarter** in the city of Vancouver. The measured display in dollars sold (in thousands).

2-D view of Sales Data

location ="Vancouver"				
time (quarter)	item (type)			
	home entertainment	computer	phone	security
Q1	605	825	14	400
Q2	680	952	31	512
Q3	812	1023	30	501
Q3	927	1038	38	580

What is Star Schema?

- A star schema is the elementary form of a dimensional model, in which data are organized into **facts** and **dimensions**.
- A fact is an event that is counted or measured, such as a sale or log in.
- A dimension includes reference data about the fact, such as date, item, or customer.
- A star schema is a relational schema where a relational schema whose design represents a multidimensional data model.
- The star schema is the explicit data warehouse schema. It is known as **star schema** because the entity-relationship diagram of this schemas simulates a star, with points, diverge from a central table.
- The center of the schema consists of a large fact table, and the points of the star are the dimension tables.



Star Schema

Fact Tables

- A table in a star schema which contains facts and connected to dimensions.
- A fact table has two types of columns: those that include fact and those that are foreign keys to the dimension table. The primary key of the fact tables is generally a composite key that is made up of all of its foreign keys.
- A fact table might involve either detail level fact or fact that have been aggregated (fact tables that include aggregated fact are often instead called summary tables). A fact table generally contains facts with the same level of aggregation.

Dimension Tables

- A dimension is an architecture usually composed of one or more hierarchies that categorize data.
- If a dimension has not got hierarchies and levels, it is called a **flat dimension** or **list**. The primary keys of each of the dimension's table are part of the composite primary keys of the fact table.
- Dimensional attributes help to define the dimensional value. They are generally descriptive, textual values. Dimensional tables are usually small in size than fact table.
- Fact tables store data about sales while dimension tables data about the geographic region (markets, cities), clients, products, times, channels.

Characteristics of Star Schema

The star schema is intensely suitable for data warehouse database design because of the following features:

- It creates a DE-normalized database that can quickly provide query responses.
- It provides a flexible design that can be changed easily or added to throughout the development cycle, and as the database grows.
- It provides a parallel in design to how end-users typically think of and use the data.
- It reduces the complexity of metadata for both developers and end-users.

Advantages of Star Schema

Star Schemas are easy for end-users and application to understand and navigate. With a well-designed schema, the customer can instantly analyze large, multidimensional data sets.

The main advantage of star schemas in a decision-support environment are:

1. **Query Performance**

A star schema database has a limited number of table and clear join paths, the query run faster than they do against OLTP systems. Small single-table queries, frequently of a dimension table, are almost instantaneous. Large join queries that contain multiple tables takes only seconds or minutes to run.

In a star schema database design, the dimension is connected only through the central fact table. When the two-dimension table is used in a query, only one join path, intersecting the fact tables, exist between those two tables. This design feature enforces authentic and consistent query results.

2. **Load performance and administration**

Structural simplicity also decreases the time required to load large batches of record into a star schema database. By describing facts and dimensions and separating them into the various table, the impact of a load structure is reduced. Dimension table can be populated once and occasionally refreshed. We can add new facts regularly and selectively by appending records to a fact table.

3. **Built-in referential integrity**

A star schema has referential integrity built-in when information is loaded. Referential integrity is enforced because each data in dimensional tables has a unique primary key, and all keys in the fact table are legitimate foreign keys drawn from the dimension table. A record in the fact table which is not related correctly to a dimension cannot be given the correct key value to be retrieved.

4. **Easily Understood**

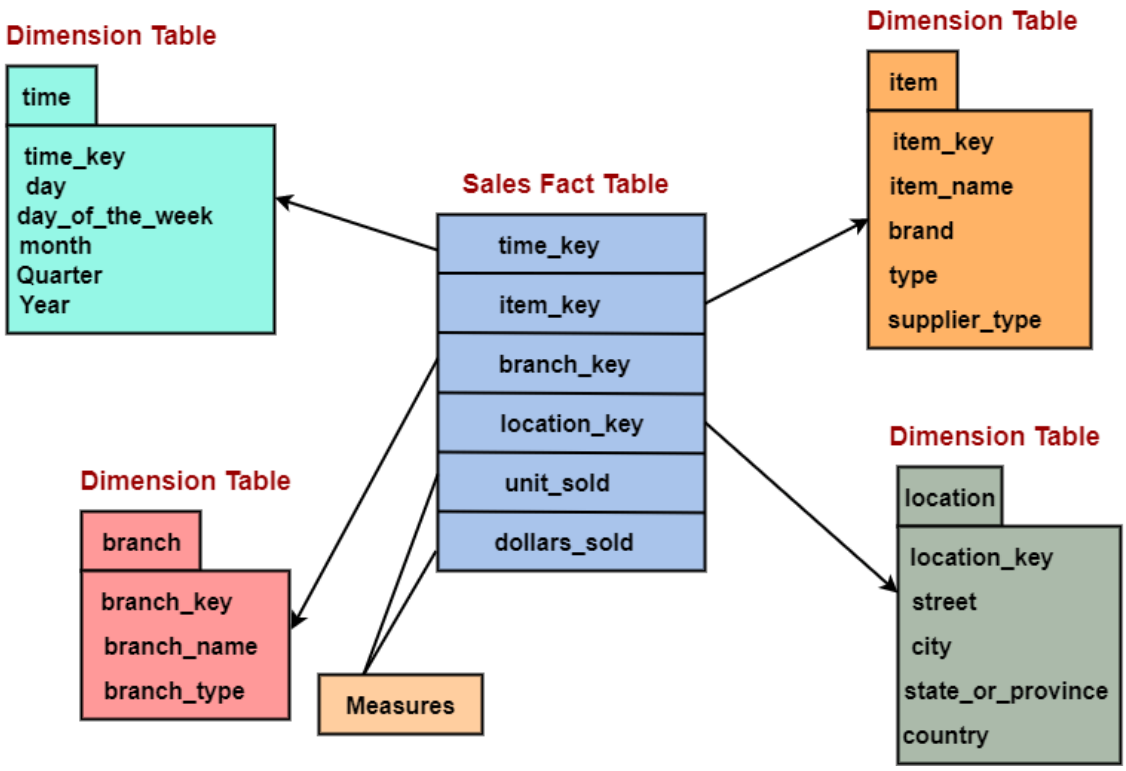
A star schema is simple to understand and navigate, with dimensions joined only through the fact table. These joins are more significant to the end-user because they represent the fundamental relationship between parts of the underlying business. Customer can also browse dimension table attributes before constructing a query.

Disadvantage of Star Schema

There is some condition which cannot be meet by star schemas like the relationship between the user, and bank account cannot describe as star schema as the relationship between them is many to many.

Example: Suppose a star schema is composed of a fact table, SALES, and several dimension tables connected to it for time, branch, item, and geographic locations.

The TIME table has a column for each day, month, quarter, and year. The ITEM table has columns for each item_Key, item_name, brand, type, supplier_type. The BRANCH table has columns for each branch_key, branch_name, branch_type. The LOCATION table has columns of geographic data, including street, city, state, and country.



In this scenario, the SALES table contains only four columns with IDs from the dimension tables, TIME, ITEM, BRANCH, and LOCATION, instead of four columns for time data, four columns for ITEM data, three columns for BRANCH data, and four columns for LOCATION data. Thus, the size of the fact table is significantly reduced. When we need to change an item, we need only make a single change in the dimension table, instead of making many changes in the fact table.

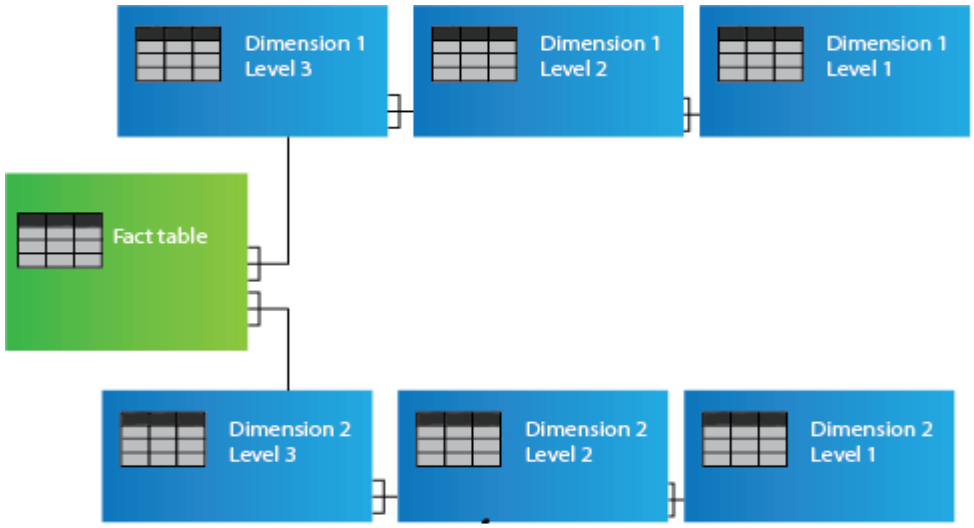
We can create even more complex star schemas by normalizing a dimension table into several tables. The normalized dimension table is called a **Snowflake**.

What is Snowflake Schema?

- A snowflake schema is equivalent to the star schema. "A schema is known as a snowflake if one or more-dimension tables do not connect directly to the fact table but must join through other dimension tables."
- 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 schema. 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.

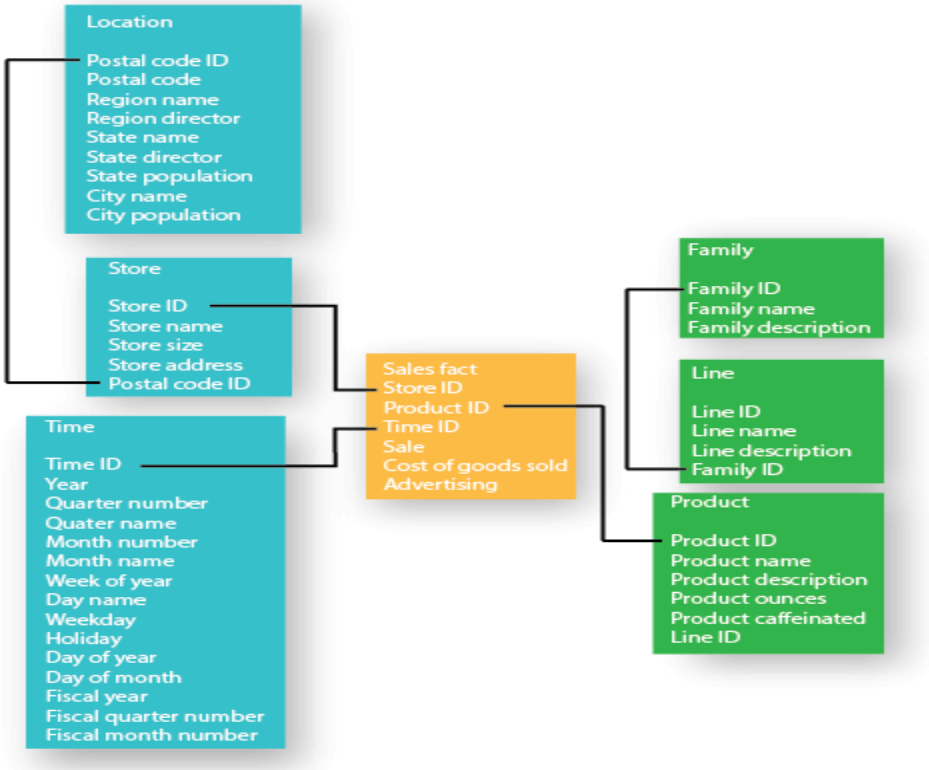
- The snowflake schema consists of one fact table which is linked to many dimension tables, which can be linked to other dimension tables through a many-to-one relationship.

The following diagram shows a snowflake schema with two dimensions, each having three levels. A snowflake schemas can have any number of dimension, and each dimension can have any number of levels.



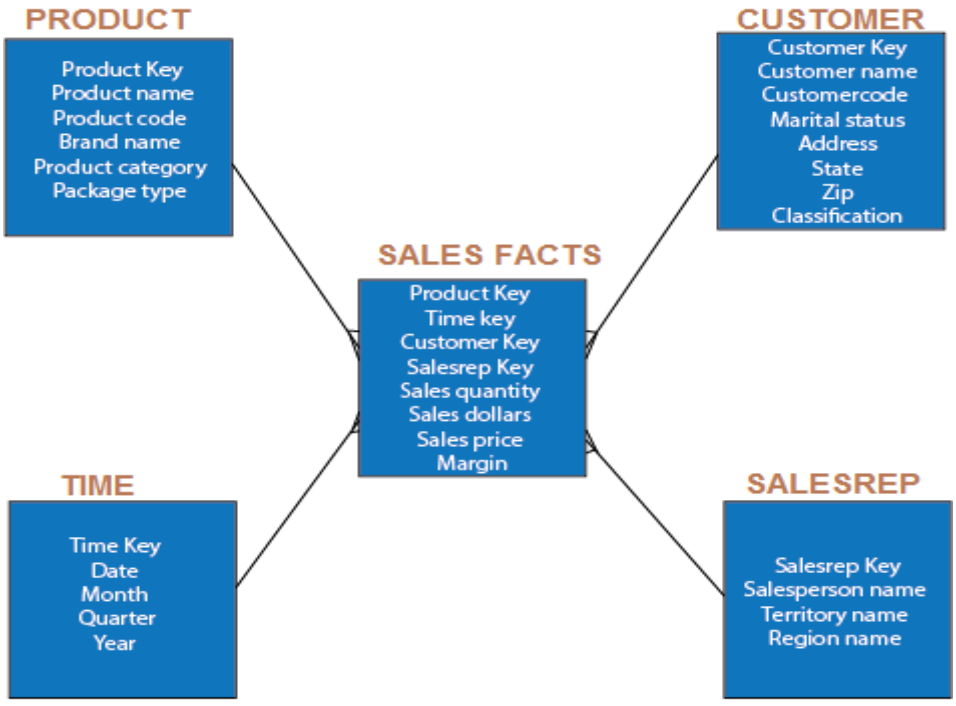
Snowflake Schema

Example: Figure shows a snowflake schema with a Sales fact table, with Store, Location, Time, Product, Line, and Family dimension tables. The Market dimension has two dimension tables with Store as the primary dimension table, and Location as the outrigger dimension table. The product dimension has three-dimension tables with Product as the primary dimension table, and the Line and Family table are the outrigger dimension tables.



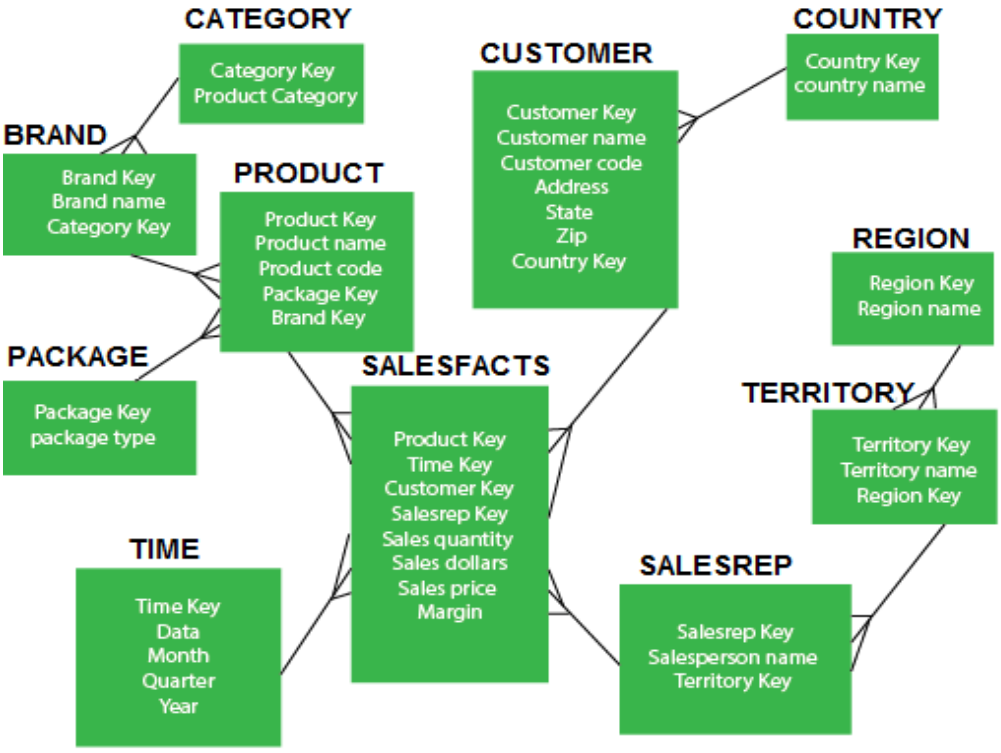
- A star schema stores all attributes for a dimension into one denormalized table. This needed more disk space than a more normalized snowflake schema.
- Snowflaking normalizes the dimension by moving attributes with low cardinality into separate dimension tables that relate to the core dimension table by using foreign keys. Snowflaking for the sole purpose of minimizing disk space is not recommended, because it can adversely impact query performance.
- In snowflake, schema tables are normalized to delete redundancy. In snowflake dimension tables are damaged into multiple dimension tables.

Figure shows a simple STAR schema for sales in a manufacturing company. The sales fact table include quantity, price, and other relevant metrics. SALESREP, CUSTOMER, PRODUCT, and TIME are the dimension tables.



STAR Schema

The STAR schema for sales, as shown above, contains only five tables, whereas the normalized version now extends to eleven tables. We will notice that in the snowflake schema, the attributes with low cardinality in each original dimension tables are removed to form separate tables. These new tables are connected back to the original dimension table through artificial keys.



Snowflake Schema

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.

Advantage of Snowflake Schema

1. The primary advantage of the snowflake schema is the development in query performance due to minimized disk storage requirements and joining smaller lookup tables.
2. It provides greater scalability in the interrelationship between dimension levels and components.
3. No redundancy, so it is easier to maintain.

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.