GATE in Data Science and AI Study Materials
Data Warehousing
By Piyush Wairale

Instructions:

- Kindly go through the lectures/videos on our website www.piyushwairale.com
- Read this study material carefully and make your own handwritten short notes. (Short notes must not be more than 5-6 pages)
- Attempt the question available on portal.
- Revise this material at least 5 times and once you have prepared your short notes, then revise your short notes twice a week
- **If you are not able to understand any topic or required detailed explanation, please mention it in our discussion forum on webiste**
- **Let me know, if there are any typos or mistake in study materials. Mail me at piyushwairale100@gmail.com**

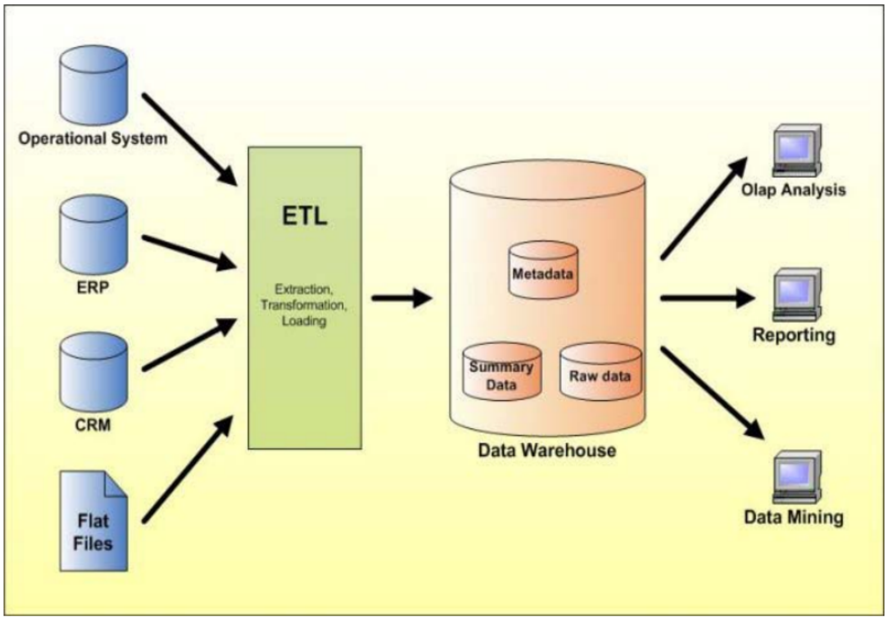
1 Data Warehousing

- A data warehouse is a repository (data and metadata) that contains integrated, cleansed, and reconciled data from disparate sources for decision support applications, with an emphasis on online analytical processing.

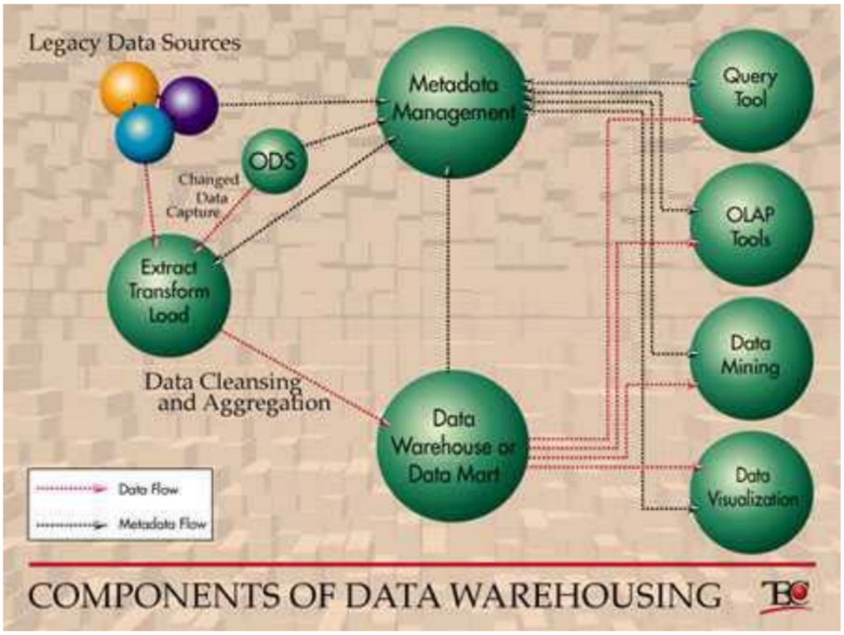
- A data warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision-making process.

  - **Subject-Oriented:** A data warehouse can be used to analyze a particular subject area. For example, "sales" can be a particular subject.
  - **Integrated:** A data warehouse integrates data from multiple data sources. For example, source A and source B may have different ways of identifying a product, but in a data warehouse, there will be only a single way of identifying a product.
  - **Time-Variant:** Historical data is kept in a data warehouse. For example, one can retrieve data from 3 months, 6 months, 12 months, or even older data from a data warehouse. This contrasts with a transactions system, where often only the most recent data is kept. For example, a transaction system may hold the most recent address of a customer, where a data warehouse can hold all addresses associated with a customer.
  - **Non-volatile:** Once data is in the data warehouse, it will not change. So, historical data in a data warehouse should never be altered.

Data Warehouse Architecture



Components of Data Warehousing



1.1 ETL Process in Data Warehouse

ETL stands for Extract, Transform, Load and it is a process used in data warehousing to extract data from various sources, transform it into a format suitable for loading into a data warehouse, and then load it into the warehouse. The process of ETL can be broken down into the following three stages:

1. **Extract:** The first stage in the ETL process is to extract data from various sources such as transactional systems, spreadsheets, and flat files. In this step, data from various source systems is extracted which can be in various formats like relational databases, No SQL, XML, and flat files into the staging area. It is important to extract the data from various source systems and store it into the staging area first and not directly into the data warehouse because the extracted data is in various formats and can be corrupted also. Hence loading it directly into the data warehouse may damage it and rollback will be much more difficult. Therefore, this is one of the most important steps of ETL process

2. **Transform:** In this stage, the extracted data is transformed into a format that is suitable for loading into the data warehouse. In this step, a set of rules or functions are applied on the extracted data to convert it into a single standard format. It may involve following processes/tasks:

Filtering – loading only certain attributes into the data warehouse.
Cleaning – filling up the NULL values with some default values, mapping U.S.A, United States, and America into USA, etc.
Joining – joining multiple attributes into one.
Splitting – splitting a single attribute into multiple attributes.
Sorting – sorting tuples on the basis of some attribute (generally key-attribute).

3. **Load:** After the data is transformed, it is loaded into the data warehouse. This step involves creating the physical data structures and loading the data into the warehouse. In this step, the transformed data is finally loaded into the data warehouse. Sometimes the data is updated by loading into the data warehouse very frequently and sometimes it is done after longer but regular intervals. The rate and period of loading solely depend on the requirements and varies from system to system.

The ETL process is an iterative process that is repeated as new data is added to the warehouse. The process is important because it ensures that the data in the data warehouse is accurate, complete, and up-to-date. It also helps to ensure that the data is in the format required for data mining and reporting.

1.1.1 Advantages of ETL process in data warehousing:

- Improved data quality: The ETL process ensures that the data in the data warehouse is accurate, complete, and up-to-date.

- Better data integration: ETL process helps to integrate data from multiple sources and systems, making it more accessible and usable.

- Increased data security: The ETL process can help to improve data security by controlling access to the data warehouse and ensuring that only authorized users can access the data.

- Improved scalability: The ETL process can help to improve scalability by providing a way to manage and analyze large amounts of data.

- Increased automation: ETL tools and technologies can automate and simplify the ETL process, reducing the time and effort required to load and update data in the warehouse.

1.1.2 Disadvantages of ETL process in data warehousing:

- High cost: ETL process can be expensive to implement and maintain, especially for organizations with limited resources.

- Complexity: ETL process can be complex and difficult to implement, especially for organizations that lack the necessary expertise or resources.

- Limited flexibility: ETL process can be limited in terms of flexibility, as it may not be able to handle unstructured data or real-time data streams.
- Limited scalability: ETL process can be limited in terms of scalability, as it may not be able to handle very large amounts of data.
- Data privacy concerns: ETL process can raise concerns about data privacy, as large amounts of data are collected, stored, and analyzed.

1.2 Data Warehouse Design Process

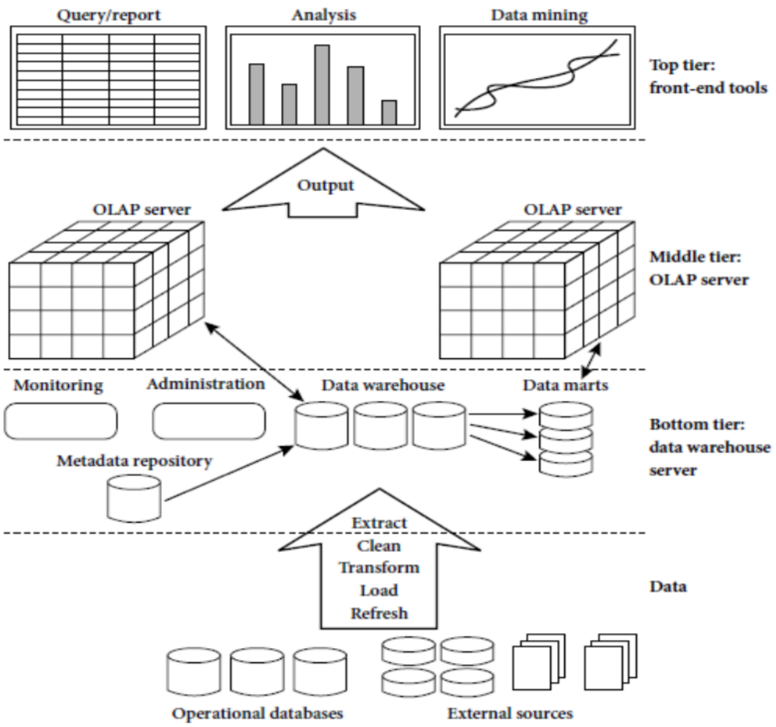
A data warehouse can be built using a top-down approach, a bottom-up approach, or a combination of both.

- The top-down approach starts with the overall design and planning. It is useful in cases where the technology is mature and well known, and where the business problems that must be solved are clear and well understood.
- The bottom-up approach starts with experiments and prototypes. This is useful in the early stage of business modeling and technology development. It allows an organization to move forward at considerably less expense and to evaluate the benefits of the technology before making significant commitments.
- In the combined approach, an organization can exploit the planned and strategic nature of the top-down approach while retaining the rapid implementation and opportunistic application of the bottom-up approach.

The warehouse design process consists of the following steps:

- Choose a business process to model, for example, orders, invoices, shipments, inventory, account administration, sales, or the general ledger. If the business process is organizational and involves multiple complex object collections, a data warehouse model should be followed. However, if the process is departmental and focuses on the analysis of one kind of business process, a data mart model should be chosen.
- Choose the grain of the business process. The grain is the fundamental, atomic level of data to be represented in the fact table for this process, for example, individual transactions, individual daily snapshots, and so on.
- Choose the dimensions that will apply to each fact table record. Typical dimensions are time, item, customer, supplier, warehouse, transaction type, and status.
- Choose the measures that will populate each fact table record. Typical measures are numeric additive quantities like dollars sold and units sold.

1.3 A Three Tier Data Warehouse Architecture



Tier-1:

- The bottom tier is a warehouse database server that is almost always a relational database system.
- Back-end tools and utilities are used to feed data into the bottom tier from operational databases or other external sources (such as customer profile information provided by external consultants). These tools and utilities perform data extraction, cleaning, and transformation (e.g., to merge similar data from different sources into a unified format), as well as load and refresh functions to update thedata warehouse .
- The data are extracted using application program interfaces known as gateways. A gateway is supported by the underlying DBMS andallows client programs to generate SQL code to be executed at a server.
- Examplesof gateways include ODBC (Open Database Connection) and OLEDB (Open Linking and Embedding for Databases) by Microsoft and JDBC (Java Database Connection).

This tier also contains a metadata repository, which stores information about the data warehouse and its contents.

Tier-2:

- The middle tier is an OLAP server that is typically implemented using either a relational OLAP (ROLAP) model or a multidimensional OLAP.
- OLAP model is an extended relational DBMS that operations on multidimensional data to standard relational operations.
- A multidimensional OLAP (MOLAP) model, that is, a special-purpose server that directly implements multidimensional data and operations.

Tier-3:
The top tier is a front-end client layer, which contains query and reporting tools, analysis tools, and/or data mining tools (e.g., trend analysis, prediction, and so on)

1.4 Data Warehouse Models:

There are three data warehouse models.

1. Enterprise warehouse:

- An enterprise warehouse collects all of the information about subjects spanning the entire organization.
- It provides corporate-wide data integration, usually from one or more operational systems or external information providers, and is cross-functional in scope.
- It typically contains detailed data aswell as summarized data, and can range in size from a few gigabytes to hundreds of gigabytes, terabytes, or beyond.
- An enterprise data warehouse may be implemented on traditional mainframes, computer superservers, or parallel architecture platforms. It requires extensive business modeling and may take years to design and build.

2. Data mart:

- A data mart contains a subset of corporate-wide data that is of value to aspecific group of users. The scope is confined to specific selected subjects. For example,a marketing data mart may confine its subjects to customer, item, and sales. The data contained in data marts tend to be summarized.
- Data marts are usually implemented on low-cost departmental servers that are UNIX/LINUX- or Windows-based. The implementation cycle of a data mart is more likely to be measured in weeks rather than months or years. However, it may involve complex integration in the long run if its design and planning were not enterprise-wide.

- Depending on the source of data, data marts can be categorized as independent or dependent. Independent data marts are sourced from data captured from one or more operational systems or external information providers, or from data generated locally within a particular department or geographic area. Dependent data marts are sourced-directly from enterprise data warehouses.

3. Virtual warehouse:

- A virtual warehouse is a set of views over operational databases. For efficient query processing, only some of the possible summary views may be materialized.
- A virtual warehouse is easy to build but requires excess capacity on operational database servers.

1.5 Meta Data Repository

Metadata are data about data. When used in a data warehouse, metadata are the data that defines warehouse objects. Metadata are created for the data names anddefinitions of the given warehouse. Additional metadata are created and captured fortimestamping any extracted data, the source of the extracted data, and missing fieldsthat have been added by data cleaning or integration processes.
A metadata repository should contain the following:

- A description of the structure of the data warehouse, which includes the warehouse schema, view, dimensions, hierarchies, and derived data definitions, as well as data mart locations and contents.
- Operational metadata, which includes data lineage (history of migrated data and the the sequence of transformations applied to it), the currency of data (active, archived, or purged), and monitoring information (warehouse usage statistics, error reports, and audit trails).
- The algorithms used for summarization, which include measure and dimension definition algorithms, data on granularity, partitions, subject areas, aggregation, summarization, and predefined queries and reports.
- The mapping from the operational environment to the data warehouse, which include databases and their contents, gateway descriptions, data partitions, data extraction, cleaning, transformation rules and defaults, data refresh and purging rules, and security (user authorization and access control).
- Data related to system performance, which includes indices and profiles that improved access and retrieval performance, in addition to rules for the timing and scheduling of refresh, update, and replication cycles. Business metadata, which includes business terms and definitions, data ownership information, and charging policies

1.6 Schemas for Multidimensional Databases

The most popular data model for a data warehouse is a multidimensional model, which can exist in the form of a star schema, a snowflake schema, or a fact constellation schema. Let's look at each of these

1. **Star schema:** The most common modeling paradigm is the star schema, in which the data warehouse contains
   (1) a large central table (fact table) containing the bulk of the data, with no redundancy, and
   (2) a set of smaller attendant tables (dimension tables), one for each dimension. The schema graph resembles a starburst, with the dimension tables displayed in a radial pattern around the central fact table.

   A star schema for AllElectronics sales is shown Table 1, Sales are considered along four dimensions: time, item, branch, and location. The schema contains a central fact table for sales that contains keys to each of the four dimensions, along with two measures: dollars sold and units sold. To minimize the size of the fact table, dimension identifiers (e.g., time key and item key) are system generated identifiers. Notice that in the star schema, each dimension is represented by only one table, and each table contains a set of attributes.
   Example: Let, an organization sells products throughout the world. The main four major dimensions are time, location, time, and branch.

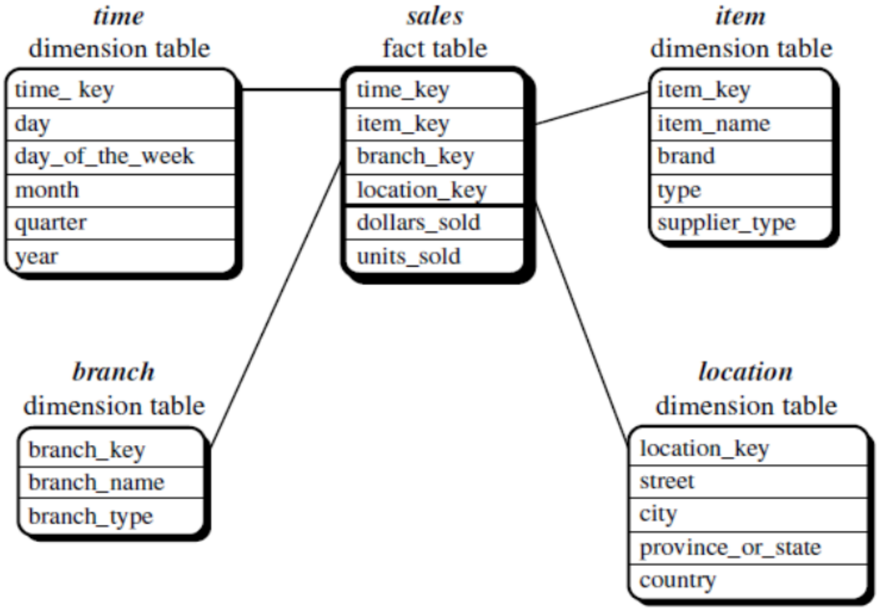


Figure 1

2. **Snowflake schema:**
The snowflake schema is a variant of the star schema model, where some dimension tables are normalized, thereby further splitting the data into additional tables. The resulting schema graph forms a shape similar to a snowflake. The major difference between the snowflake and star schema models is that the dimension tables of the snowflake model may be kept in the normalized form to reduce redundancies. Such a table is easy to maintain and saves storage space. However, this space savings is negligible in comparison to the typical magnitude of the fact table. Furthermore, the snowflake structure can reduce the effectiveness of browsing, since more joins will be needed to execute a query. Consequently, the system performance may be adversely impacted. Hence, although the snowflake schema reduces redundancy, it is not as popular as the star schema in data warehouse design.

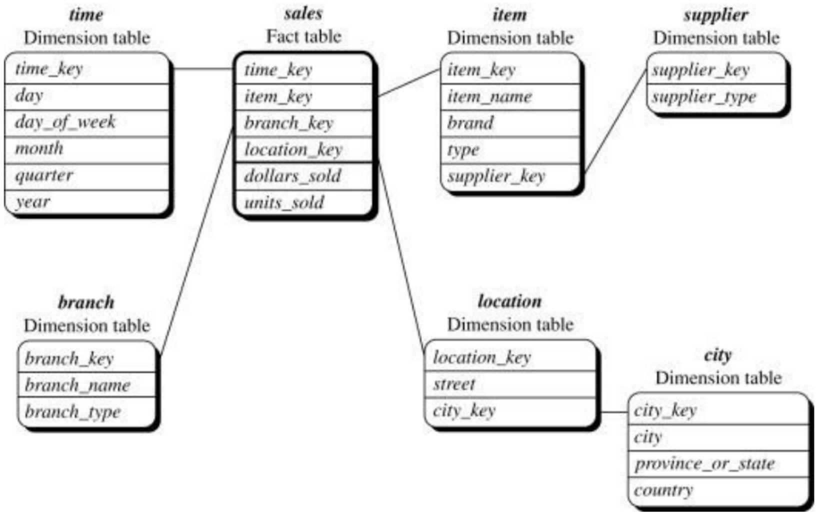


Figure 2

A snowflake schema for AllElectronics sales is given in Figure 2. Here, the sales fact table is identical to that of the star schema in Figure 1. The main difference between the two schemas is in the definition of dimension tables. The single dimension table for an item in the star schema is normalized in the snowflake schema, resulting in new item and supplier tables. For example, the item dimension table now contains the attributes item key, item name, brand, type, and supplier key, where supplier key is linked to the supplier dimension table, containing supplier key and supplier type information. Similarly, the single dimension table for location in the star schema can be normalized into two new tables: location and city. The city key in the new location table links to the city dimension. Notice that, when desirable, further normalization can be performed on province or state and country in the snowflake schema shown in Figure 6.

3. **Fact constellation:** Sophisticated applications may require multiple fact tables to share dimension tables. This kind of schema can be viewed as a collection of stars and hence is called a galaxy schema or a fact constellation.

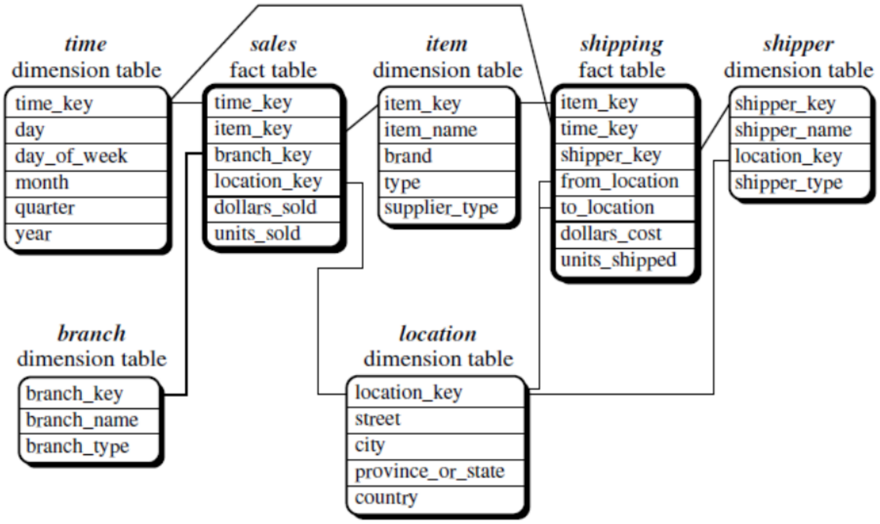


Figure 3

A fact constellation schema is shown in Figure 3. This schema specifies two fact tables, sales, and shipping. The sales table definition is identical to that of the star schema (Figure 1). The shipping table has five dimensions, or keys—item key, time key, shipper key, from location, and to location— and two measures—dollars cost and units shipped. A fact constellation schema allows dimension tables to be shared between fact tables. For example, the dimensions tables for time, item, and location are shared between the sales and shipping fact tables.

1.7 Concept Hierarchies

A concept hierarchy defines a sequence of mappings from a set of low-level concepts to higher-level, more general concepts. Consider a concept hierarchy for the dimension location. City values for location include Vancouver, Toronto, New York, and Chicago. Each city, however, can be mapped to the province or state to which it belongs. For example, Vancouver can be mapped to British Columbia, and Chicago to Illinois. The provinces and states can in turn be mapped to the country (e.g., Canada or the United States) to which they belong. These mappings form a concept hierarchy for the dimension location, mapping a set of low-level concepts (i.e., cities) to higher-level, more general concepts (i.e., countries). This concept of hierarchy is illustrated in Figure 4.

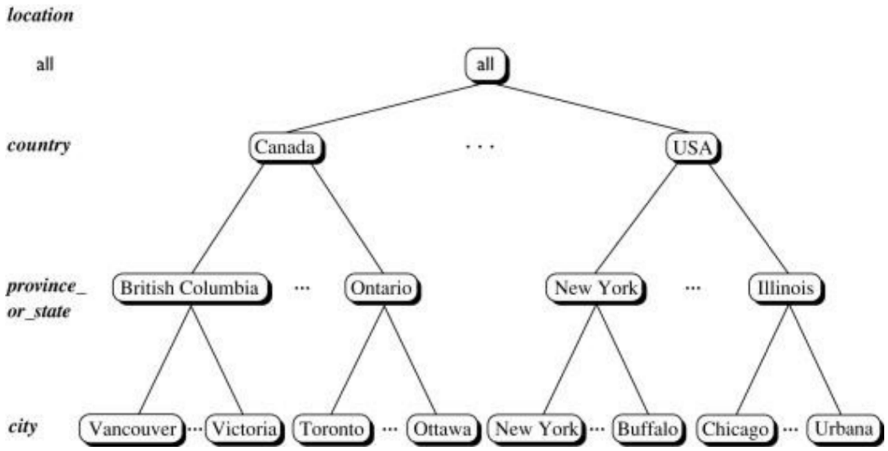


Figure 4

1.8 Measures: Their Categorization and Computation

"How are measures computed?"
To answer this question, we first study how measures can be categorized.
Note that a multidimensional point in the data cube space can be defined by a set dimension–value pairs; for example,(time ="Q1", location = "Vancouver", item ="computer"). A data cube measure is a numeric function that can be evaluated at each point in the data cube space.

A measured value is computed for a given point by aggregating the data corresponding to the respective dimension-value pairs defining the given point.

- **Distributive:** An aggregate function is distributive if it can be computed in a distributed manner as follows. Suppose the data are partitioned into n sets. We apply the function to each partition, resulting in n aggregate values. If the result derived by applying the function to the n aggregate values is the same as that derived by applying the function to the entire data set (without partitioning), the function can be computed in a distributed manner. For example, sum() can be computed for a data cube by first partitioning the cube into a set of sub-cubes, computing sum() for each sub-cube, and then summing up the counts obtained for each sub-cube. Hence, sum() is a distributive aggregate function. For the same reason, count(), min(), and max() are distributive aggregate functions.

- **Algebraic:** An aggregate function is algebraic if it can be computed by an algebraic function with M arguments (where M is a bounded positive integer), each of which is obtained by applying a distributive aggregate function. For example, avg() (average) can be computed by sum()/count(), where both sum() and count() are distributive aggregate functions. Similarly, it can be shown that min N() and max N() (which find the N minimum and N maximum values, respectively, in a given set) and standard deviation() are algebraic aggregate functions. A measure is algebraic if it is obtained by applying an algebraic aggregate function.

- **Holistic:** An aggregate function is holistic if there is no constant bound on the storage size needed to describe a sub-aggregate. That is, there does not exist an algebraic function with M arguments (where M is a constant) that characterizes the computation. Common examples of holistic functions include median(), mode(), and rank(). A measure is holistic if it is obtained by applying a holistic aggregate function.

1.9 OLAP(Online Analytical Processing):

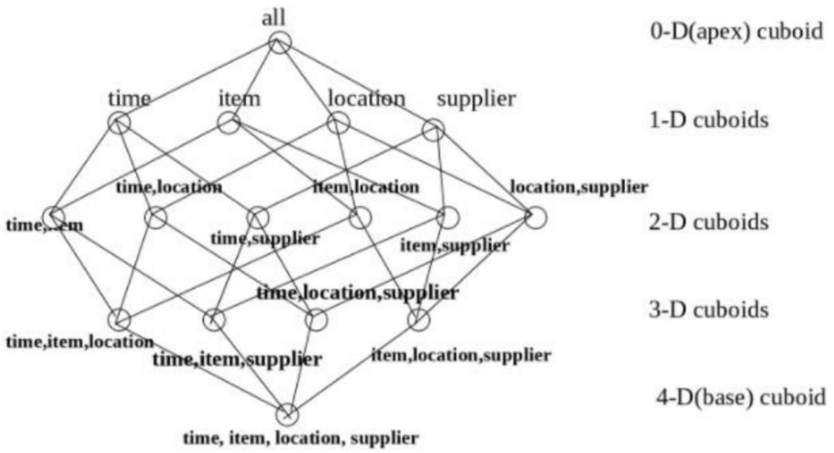
- OLAP is an approach to answering multi-dimensional analytical (MDA) queries swiftly.
- OLAP is part of the broader category of business intelligence, which also encompasses relational database, report writing and data mining.
- OLAP tools enable users to analyze multidimensional data interactively from multiple perspectives.

“A data cube allows data to be modeled and viewed in multiple dimensions. It is defined by dimensions and facts.

Dimensions are the entities concerning which an organization wants to keep records.

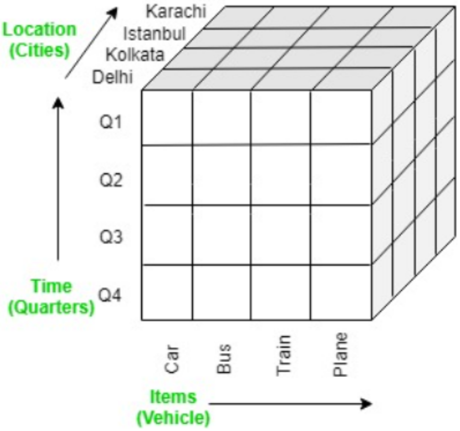
Facts are numerical measures. It is the quantities by which we want to analyze relationships between dimensions.

The data cube is used by the users of the decision support system to see their data. The cuboid that holds the lowest level of summarization is called **the base cuboid.** The 0-D cuboid, which holds the highest level of summarization is called **the apex cuboid.**



1.9.1 OLAP Operations in DBMS

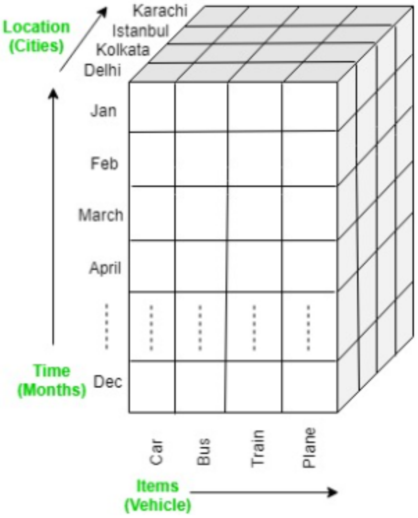
OLAP databases are divided into one or more cubes and these cubes are known as Hypercubes.



OLAP operations:
Five basic analytical operations can be performed on an OLAP cube:

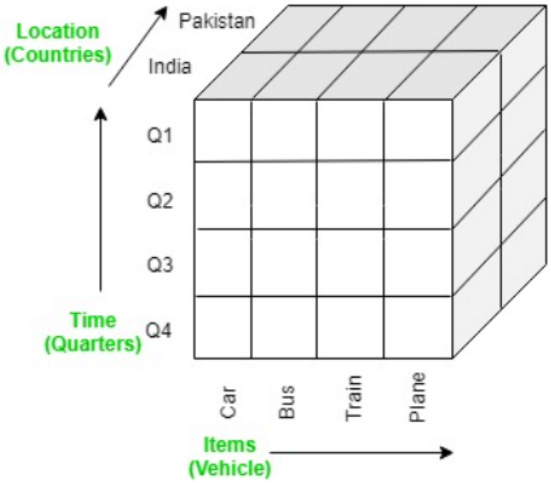
1. **Drill down:** In drill-down operation, the less detailed data is converted into highly detailed data. It can be done by:
   Moving down in the concept hierarchy
   Adding a new dimension

   In the cube given in the overview section, the drill-down operation is performed by moving down in the concept hierarchy of Time dimension (Quarter → Month).

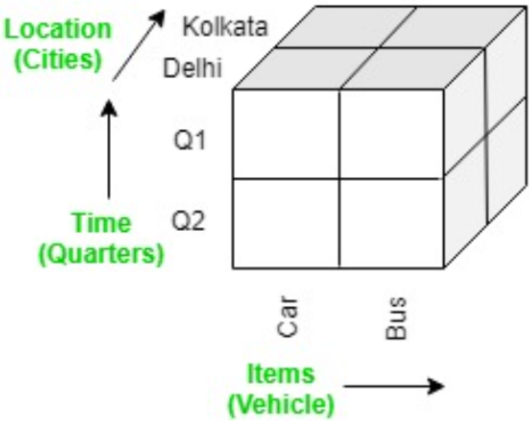


2. **Roll up:** It is just the opposite of the drill-down operation. It performs aggregation on the OLAP cube. It can be done by:
   Climbing up in the concept hierarchy
   Reducing the dimensions

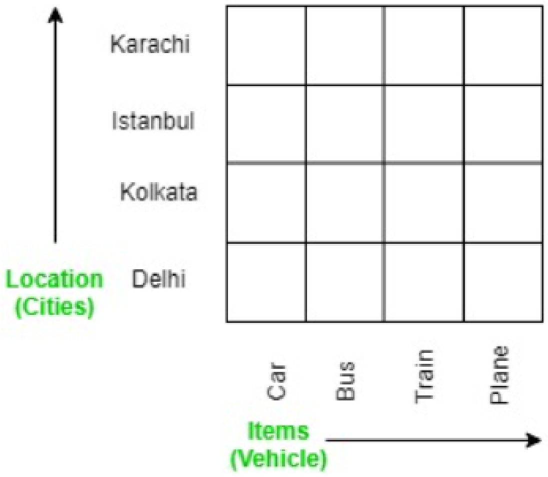
   In the cube given in the overview section, the roll-up operation is performed by climbing up in the concept hierarchy of the Location dimension (City → Country).



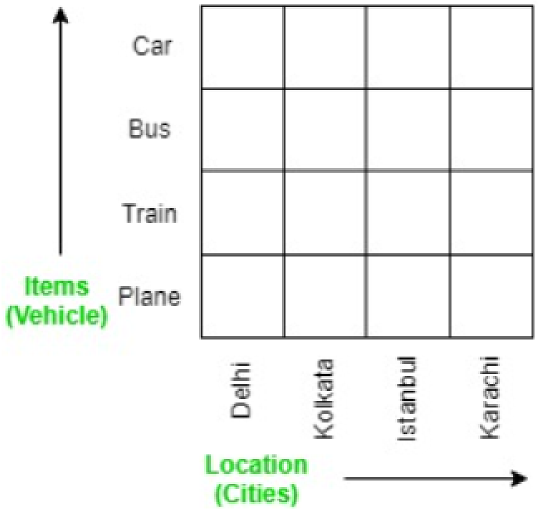
3. **Dice:** It selects a sub-cube from the OLAP cube by selecting two or more dimensions. In the cube given in the overview section, a sub-cube is selected by selecting the following dimensions with criteria:
   Location = “Delhi” or “Kolkata”
   Time = “Q1” or “Q2”
   Item = “Car” or “Bus”



4. Slice: It selects a single dimension from the OLAP cube which results in a new sub-cube creation. In the cube given in the overview section, Slice is performed on the dimension Time = "Q1".



5. Pivot: It is also known as rotation operation as it rotates the current view to get a new view of the representation. In the sub-cube obtained after the slice operation, performing pivot operation gives a new view of it.



Types of OLAP:

1. **Relational OLAP (ROLAP):**

   - ROLAP works directly with relational databases. The base data and the dimension tables are stored as relational tables and new tables are created to hold the aggregated information. It depends on a specialized schema design.
   - This methodology relies on manipulating the data stored in the relational database to give the appearance of traditional OLAP's slicing and dicing functionality. In essence, each action of slicing and dicing is equivalent to adding a "WHERE" clause in the SQL statement.
   - ROLAP tools do not use pre-calculated data cubes but instead pose the query to the standard relational database and its tables in order to bring back the data required to answer the question.
   - ROLAP tools feature the ability to ask any question because the methodology does not limited to the contents of a cube. ROLAP also has the ability to drill down to the lowest level of detail in the database.

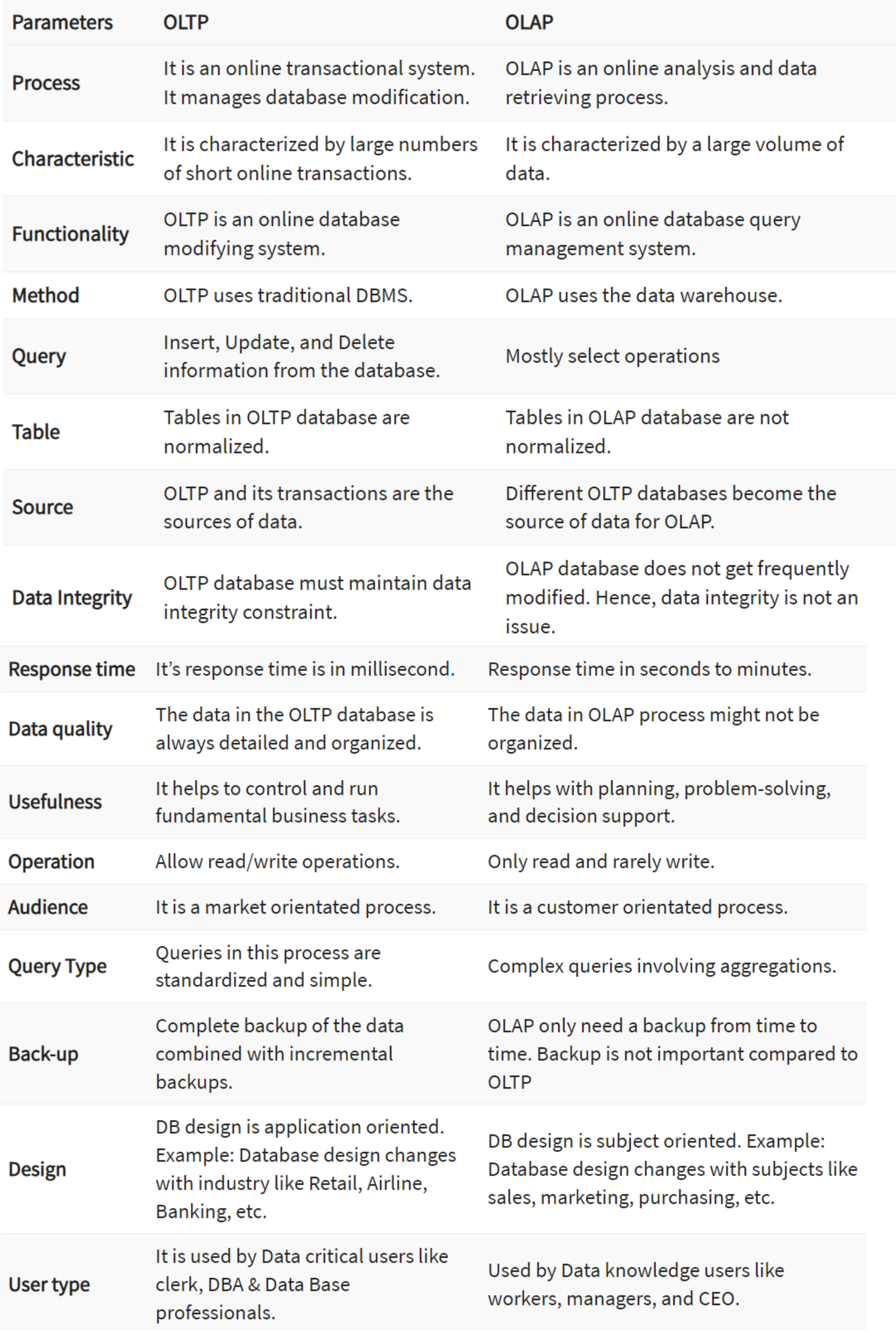
2. **Multidimensional OLAP (MOLAP):**

   - MOLAP is the 'classic' form of OLAP and is sometimes referred to as just OLAP.
   - MOLAP stores this data in an optimized multi-dimensional array storage, rather than in a relational database. Therefore it requires the pre-computation and storage of information in the cube - the operation known as processing.
   - MOLAP tools generally utilize a pre-calculated data set referred to as a data cube. The data cube contains all the possible answers to a given range of questions.
   - MOLAP tools have a very fast response time and the ability to quickly write back data into the data set.

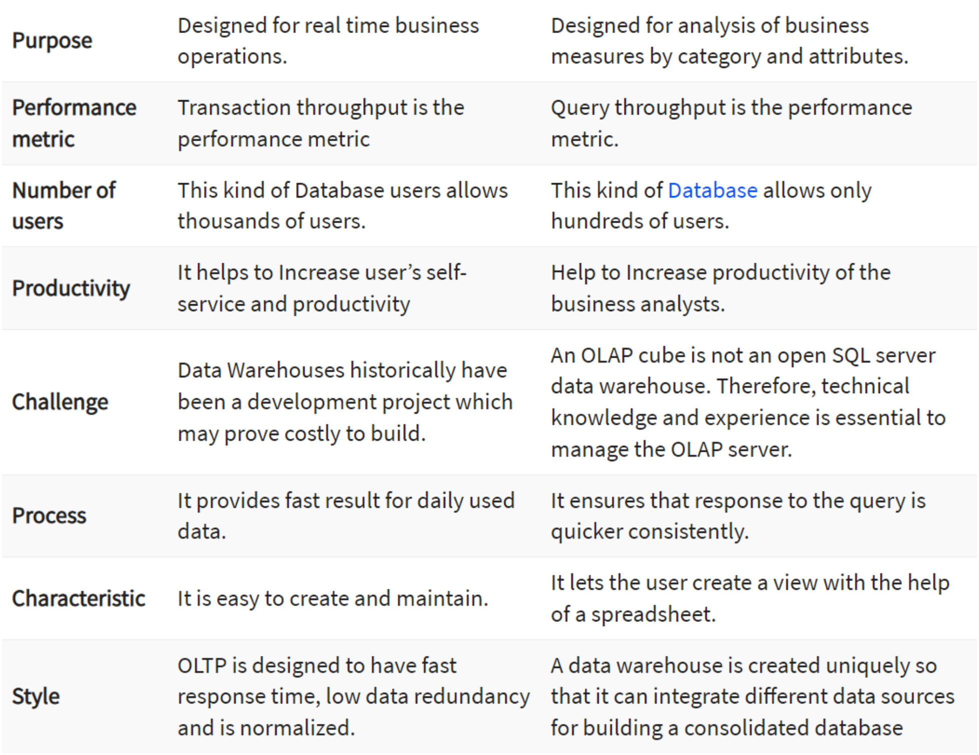
3. **Hybrid OLAP (HOLAP):**

   - There is no clear agreement across the industry as to what constitutes Hybrid OLAP, except that a database will divide data between relational and specialized storage.
   - For example, for some vendors, a HOLAP database will use relational tables to hold the larger quantities of detailed data, and use specialized storage for at least some aspects of the smaller quantities of more-aggregate or less-detailed data.
   - HOLAP addresses the shortcomings of MOLAP and ROLAP by combining the capabilities of both approaches.
   - HOLAP tools can utilize both pre-calculated cubes and relational data sources

OLTP vs OLAP



Parameters	OLTP	OLAP
Process	It is an online transactional system. It manages database modification.	OLAP is an online analysis and data retrieving process.
Characteristic	It is characterized by large numbers of short online transactions.	It is characterized by a large volume of data.
Functionality	OLTP is an online database modifying system.	OLAP is an online database query management system.
Method	OLTP uses traditional DBMS.	OLAP uses the data warehouse.
Query	Insert, Update, and Delete information from the database.	Mostly select operations
Table	Tables in OLTP database are normalized.	Tables in OLAP database are not normalized.
Source	OLTP and its transactions are the sources of data.	Different OLTP databases become the source of data for OLAP.
Data Integrity	OLTP database must maintain data integrity constraint.	OLAP database does not get frequently modified. Hence, data integrity is not an issue.
Response time	It's response time is in millisecond.	Response time in seconds to minutes.
Data quality	The data in the OLTP database is always detailed and organized.	The data in OLAP process might not be organized.
Usefulness	It helps to control and run fundamental business tasks.	It helps with planning, problem-solving, and decision support.
Operation	Allow read/write operations.	Only read and rarely write.
Audience	It is a market orientated process.	It is a customer orientated process.
Query Type	Queries in this process are standardized and simple.	Complex queries involving aggregations.
Back-up	Complete backup of the data combined with incremental backups.	OLAP only need a backup from time to time. Backup is not important compared to OLTP
Design	DB design is application oriented. Example: Database design changes with industry like Retail, Airline, Banking, etc.	DB design is subject oriented. Example: Database design changes with subjects like sales, marketing, purchasing, etc.
User type	It is used by Data critical users like clerk, DBA & Data Base professionals.	Used by Data knowledge users like workers, managers, and CEO.



Purpose	Designed for real time business operations.	Designed for analysis of business measures by category and attributes.
Performance metric	Transaction throughput is the performance metric	Query throughput is the performance metric.
Number of users	This kind of Database users allows thousands of users.	This kind of Database allows only hundreds of users.
Productivity	It helps to Increase user's self-service and productivity	Help to Increase productivity of the business analysts.
Challenge	Data Warehouses historically have been a development project which may prove costly to build.	An OLAP cube is not an open SQL server data warehouse. Therefore, technical knowledge and experience is essential to manage the OLAP server.
Process	It provides fast result for daily used data.	It ensures that response to the query is quicker consistently.
Characteristic	It is easy to create and maintain.	It lets the user create a view with the help of a spreadsheet.
Style	OLTP is designed to have fast response time, low data redundancy and is normalized.	A data warehouse is created uniquely so that it can integrate different data sources for building a consolidated database

2 Data Mining

2.1 What is data mining?

- Data mining refers to extracting or mining knowledge from large amounts of data.
- It is the computational process of discovering patterns in large data sets involving methods at the intersection of artificial intelligence, machine learning, statistics, and database systems.
- The overall goal of the data mining process is to extract information from a data set and transform it into an understandable structure for further use.
- Data mining, also known as knowledge discovery in data (KDD), is the process of uncovering patterns and other valuable information from large data sets.

The key properties of data mining are

- Automatic discovery of patterns
- Prediction of likely outcomes
- Creation of actionable information
- Focus on large datasets and databases

2.2 Tasks of Data Mining

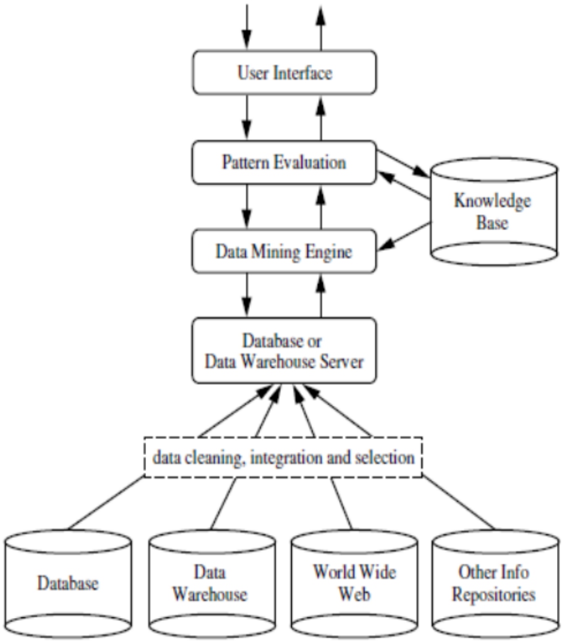
Data mining involves six common classes of tasks:

- **Anomaly detection (Outlier/change/deviation detection)** – The identification of unusual data records, that might be interesting or data errors that require further investigation.
- **Association rule learning (Dependency modelling)** – Searches for relationships between variables. For example a supermarket might gather data on customer purchasing habits. Using association rule learning, the supermarket can determine which products are frequently bought together and use this information for marketing purposes. This is sometimes referred to as market basket analysis.
- **Clustering** – is the task of discovering groups and structures in the data that are in some way or another ”similar”, without using known structures in the data.
- **Classification** – is the task of generalizing known structure to apply to new data. For example, an e-mail program might attempt to classify an e-mail as ”legitimate” or as ”spam”. Regression – attempts to find a function that models the data with the least error.
- **Summarization** – providing a more compact representation of the data set, including visualization and report generation

2.3 Architecture of Data Mining

A typical data mining system may have the following major components.

1. **Knowledge Base:** This is the domain knowledge that is used to guide the search orevaluate the interestingness of resulting patterns. Such knowledge can include concepthierarchies, used to organize attributes or attribute values into different levels of abstraction. Knowledge such as user beliefs, which can be used to assess a pattern's interestingness based on its unexpectedness, may also be included
2. **Data Mining Engine:** This is essential to the data mining system and ideally consists of a set of functional modules for tasks such as characterization, association and correlation analysis, classification, prediction, cluster analysis, outlier analysis, and evolution analysis.
3. **Pattern Evaluation Module:** This component typically employs interestingness measures interacting with the data mining modules so as to focus the search toward interesting patterns. It may use interestingness thresholds to filter out discovered patterns. Alternatively, the pattern evaluation module may be integrated with the



mining module, depending on the implementation of the data mining method used. For efficient data mining, it is highly recommended to push the evaluation of pattern interestingness as deep as possible into the mining process so as to confine the search to only the interesting patterns.

4. **User interface:** This module communicates between users and the data mining system, allowing the user to interact with the system by specifying a data mining query or task, providing information to help focus the search, and performing exploratory data mining based on the intermediate data mining results. In addition, this component allows the user to browse database and data warehouse schemas or data structures, evaluate mined patterns, and visualize the patterns in different forms.

2.4 Data Mining Process:

Data Mining is a process of discovering various models, summaries, and derived values from a given collection of data.
The general experimental procedure adapted to data-mining problems involves the following steps:

1. State the problem and formulate the hypothesis
   Most data-based modeling studies are performed in a particular application domain. Hence, domain-specific knowledge and experience are usually necessary in order to come up with a meaningful problem statement. Unfortunately, many application studies tend to focus on the data-mining technique at the expense of a clear problem

statement. In practice, it usually means a close interaction between the data-mining expert and the application expert. In successful data-mining applications, this cooperation does not stop in the initial phase; it continues during the entire data-mining process.

2. Collect the data
   This step is concerned with how the data are generated and collected. In general, there are two distinct possibilities. The first is when the data-generation process is under the control of an expert (modeler): this approach is known as a designed experiment. The second possibility is when the expert cannot influence the data- generation process: this is known as the observational approach.

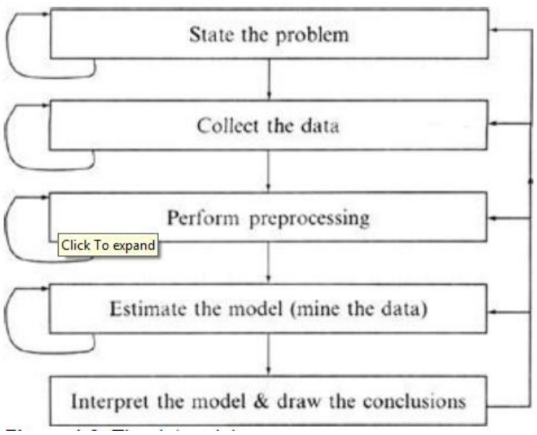
3. Preprocessing the data
   In the observational setting, data are usually "collected" from existing databases, data warehouses, and data marts. Data preprocessing usually includes at least two common tasks:

   - Outlier detection (and removal) – Outliers are unusual data values that are not consistent with most observations. Commonly, outliers result from measurement errors, coding and recording errors, and, sometimes, are natural, abnormal values. Such nonrepresentative samples can seriously affect the model produced later. There are two strategies for dealing with outliers:
     a. Detect and eventually remove outliers as a part of the preprocessing phase, or
     b. Develop robust modeling methods that are insensitive to outliers.

   - Scaling, encoding, and selecting features – Data preprocessing includes several steps such as variable scaling and different types of encoding. For example, one feature with the range [0, 1] and the other with the range [-100, 1000] will not have the same weights in the applied technique; they will also influence the final data-mining results differently. Therefore, it is recommended to scale them and bring both features to the same weight for further analysis

4. Estimate the model
   The selection and implementation of the appropriate data-mining technique is the main task in this phase. This process is not straightforward; usually, in practice, the implementation is based on several models, and selecting the best one is an additional task.

5. Interpret the model and draw conclusions
   In most cases, data-mining models should help in decision-making. Hence, such models need to be interpretable to be useful because humans are not likely to base their decisions on complex "black-box" models. Note that the goals of the accuracy of the model and the accuracy of its interpretation are somewhat contradictory. Usually, simple models are more interpretable, but they are also less accurate.


State the problem
Collect the data
Perform preprocessing
Click To expand
Estimate the model (mine the data)
Interpret the model & draw the conclusions

2.5 Classification of Data mining Systems:

The data mining system can be classified according to the following criteria:

- Database Technology
- Statistics
- Machine Learning
- Information Science
- Visualization
- Other Disciplines

Some Other Classification Criteria:

1. Classification according to kind of databases mined
2. Classification according to kind of knowledge mined
3. Classification according to kinds of techniques utilized
4. Classification according to applications adapted
5. Classification according to kind of databases mined

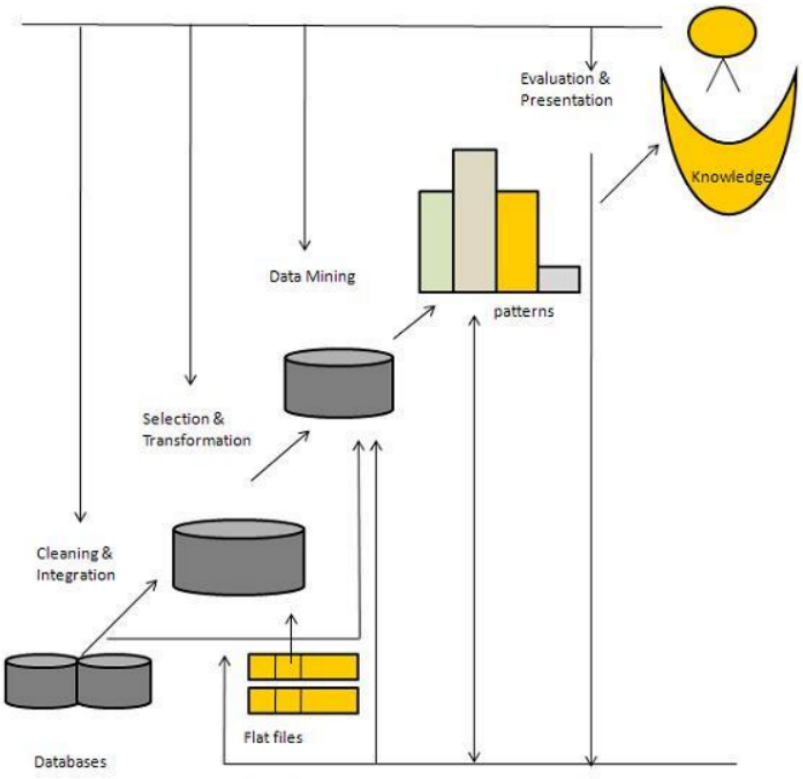
Classification according to kind of knowledge mined
We can classify the data mining system according to kind of knowledge mined. It is means data mining system are classified on the basis of functionalities such as: Characterization, Discrimination, Association and Correlation Analysis, Classification, Prediction, Clustering, Outlier Analysis, Evolution Analysis

2.6 Knowledge Discovery in Databases(KDD)

Some people treat data mining same as Knowledge discovery while some people view data mining essential step in process of knowledge discovery. Here is the list of steps involved in the knowledge discovery process:

- **Data Cleaning** - In this step the noise and inconsistent data is removed.
- **Data Integration** - In this step multiple data sources are combined.
- **Data Selection** - In this step relevant to the analysis task are retrieved from the database.
- **Data Transformation** - In this step data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations.
- **Data Mining** - In this step, intelligent methods are applied in order to extract data patterns.
- **Pattern Evaluation** - In this step, data patterns are evaluated.
- **Knowledge Presentation** - In this step, knowledge is represented.

The following diagram shows the process of knowledge discovery process:



2.7 Data Transformation

In data transformation, the data are transformed or consolidated into forms appropriatefor mining. Data transformation can involve the following:

- **Smoothing**, which works to remove noise from the data. Such techniques include binning, regression, and clustering.
- **Aggregation**, where summary or aggregation operations are applied to the data. For example, the daily sales data may be aggregated to compute monthly and annual total amounts. This step is typically used in constructing a data cube for analysis of the data at multiple granularities.
- **Generalization of the data**, where low-level or —primitive (raw) data are replaced by higher-level concepts through the use of concept hierarchies. For example, categorical attributes, like streets, can be generalized to higher-level concepts, like city or country.
- **Normalization,** where the attribute data are scaled to fall within a small specified range, such as 1:0 to 1:0, or 0:0 to 1:0.
- **Attribute construction** (or feature construction), where new attributes are constructed and added from the given set of attributes to help the mining process.

2.8 Data Reduction/Compression:

Data reduction techniques can be applied to obtain a reduced representation of the data set that is much smaller in volume, yet closely maintains the integrity of the original data. That is, mining on the reduced data set should be more efficient yet produce thesame (or almost the same) analytical results.
Strategies for data reduction include the following:

- Data cube aggregation, where aggregation operations are applied to the data in the construction of a data cube.
- Attribute subset selection, where irrelevant, weakly relevant, or redundant attributesor dimensions may be detected and removed.
- Dimensionality reduction, where encoding mechanisms are used to reduce the dataset size.
- Numerosityreduction, where the data are replaced or estimated by alternative, smaller data representations such as parametric models (which need to store only the model parameters instead of the actual data) or nonparametric methods such as clustering, sampling, and the use of histograms.

- Discretization and concept hierarchy generation, where rawdata values for attributes are replaced by ranges or higher conceptual levels. Data discretization is a form of numerosity reduction that is very useful for the automatic generation of concept hierarchies. Discretization and concept hierarchy generation are powerful tools for data mining, in that they allow the mining of data at multiple levels of abstraction.

2.9 Data Normalization

Data normalization is a technique used in data mining to transform the values of a dataset into a common scale. This is important because many machine learning algorithms are sensitive to the scale of the input features and can produce better results when the data is normalized.

Several different normalization techniques can be used in data mining, including:

1. Min-Max normalization: This technique scales the values of a feature to a range between 0 and 1. This is done by subtracting the minimum value of the feature from each value, and then dividing by the range of the feature.

2. Z-score normalization: This technique scales the values of a feature to have a mean of 0 and a standard deviation of 1. This is done by subtracting the mean of the feature from each value, and then dividing by the standard deviation.

3. Decimal Scaling: This technique scales the values of a feature by dividing the values of a feature by a power of 10.

4. Logarithmic transformation: This technique applies a logarithmic transformation to the values of a feature. This can be useful for data with a wide range of values, as it can help to reduce the impact of outliers.

5. Root transformation: This technique applies a square root transformation to the values of a feature. This can be useful for data with a wide range of values, as it can help to reduce the impact of outliers. It's important to note that normalization should be applied only to the input features, not the target variable and that different normalization techniques may work better for different types of data and models.

2.10 Data Discretization

Data discretization is defined as a process of converting continuous data attribute values into a finite set of intervals with minimal loss of information and associating with each interval some specific data value or conceptual labels.

Data discretization refers to a method of converting a huge number of data values into smaller ones so that the evaluation and management of data become easy. In other words, data discretization is a method of converting attributes values of continuous data into a finite set of intervals with minimum data loss. There are two forms of data discretization first is supervised discretization, and the second is unsupervised discretization. Supervised discretization refers to a method in which the class data is used. Unsupervised discretization refers to a method depending upon the way which operation proceeds. It means it works on the top-down splitting strategy and bottom-up merging strategy.

There are different techniques of discretization:

- **Discretization by binning:** It is an unsupervised method of partitioning the data based on equal partitions, either by equal width or by equal frequency
- **Discretization by Cluster:** clustering can be applied to discretize numeric attributes. It partitions the values into different clusters or groups by following top-down or bottom-up strategy
- **Discretization By decision tree:** it employs top-down splitting strategy. It is a supervised technique that uses class information.
- **Discretization By correlation analysis:** ChiMerge employs a bottom-up approach by finding the best neighboring intervals and then merging them to form larger intervals, recursively
- **Discretization by histogram:** Histogram analysis is unsupervised learning because it doesn't use any class information like binning. There are various partition rules used to define histograms.

Characteristic:

Histograms are a very effective technique of data reduction that can work on sparse and dense data as well as uniform and highly skewed data. Multidimensional histograms can be used to capture data up to five attributes and are effective in determining dependencies between attributes.

Importance of Discretization:

A discretization is important because it is useful:

- To generate concept hierarchies.
- Transform numeric data.
- To ease evaluation and management of data.
- To minimize data loss.
- To produce a better result.
- Generate a more understandable structure viz. decision tree.

2.11 Data Sampling

- Data Sampling is a statistical technique that involves selecting a representative subset of data from a larger dataset. The selected subset, known as a sample, is used to perform analysis, processing, or testing. The sample is chosen in such a way that it accurately represents the characteristics of the larger dataset. By analyzing the sample, businesses can draw conclusions and make informed decisions without the need to process or analyze the entire dataset.

- Data Sampling involves randomly or systematically selecting a subset of data points from a larger dataset.

- The selection process can be based on various criteria, such as random selection, stratified sampling, or cluster sampling.

- Random selection involves selecting data points randomly without any specific criteria.

- Stratified sampling involves dividing the dataset into homogeneous groups, known as strata, and selecting samples from each stratum.

- Cluster sampling involves dividing the dataset into clusters and selecting entire clusters as samples.

- Once the sample is selected, statistical techniques can be applied to analyze the sample and draw conclusions about the larger dataset. The accuracy of the conclusions depends on the representativeness of the sample and the sampling method used.

References

- Data Warehousing Guide by Oracle
- https://www.geeksforgeeks.org/olap-operations-in-dbms/
- https://www.javatpoint.com/data-warehouse
- https://www.guru99.com/oltp-vs-olap.html