Analyzing operations - Production/Service Delivery, Marketing & Sales, Customer Service, Financial Management

sources of profit - Revenue Growth, Cost Reduction, Improved Efficiency, Pricing Strategy

The major distinguishing features of OLTP and OLAP are summarized as follows:

Users and system orientation: An OLTP system is customer-oriented and is used for transaction and query processing by clerks, clients, and information technology professionals. An OLAP system is market-oriented and is used for data analysis by knowledge workers, including managers, executives, and analysts.

Data contents: An OLTP system manages current data that, typically, are too detailed to be easily used for decision making. An OLAP system manages large amounts of historic data, provides facilities for summarization and aggregation, and stores and manages information at different levels of granularity. These features make the data easier to use for informed decision making.

Database design: An OLTP system usually adopts an entity-relationship (ER) data model and an application-oriented database design. An OLAP system typically adopts either a star or a snowflake model and a subject-oriented database design.

View: An OLTP system focuses mainly on the current data within an enterprise or department, without referring to historic data or data in different organizations. In contrast, an OLAP system often spans multiple versions of a database schema, due to the evolutionary process of an organization. OLAP systems also deal with information that originates from different organizations, integrating information from many data stores. Because of their huge volume, OLAP data are stored on multiple storage media.

Access patterns: The access patterns of an OLTP system consist mainly of short, atomic transactions. Such a system requires concurrency control and recovery mechanisms. However, accesses to OLAP systems are mostly read-only operations (because most data warehouses store historic rather than up-to-date information), although many could be complex queries.

**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.

In the star schema, each dimension is represented by only one table, and each table contains a set of attributes. For example, the *location* dimension table contains the attribute set f*location key, street, city, province or state, country*g. This constraint may introduce some redundancy. For example, “Urbana” and “Chicago” are both cities in the state of Illinois, USA. Entries for such cities in the *location* dimension table will create redundancy among the attributes *province or state* and *country*; that is, ...., Urbana, IL, USA/ and ...., Chicago, IL, USA/.Moreover, the attributes within a dimension table may form either a hierarchy (total order) or a lattice (partial order).

**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 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.

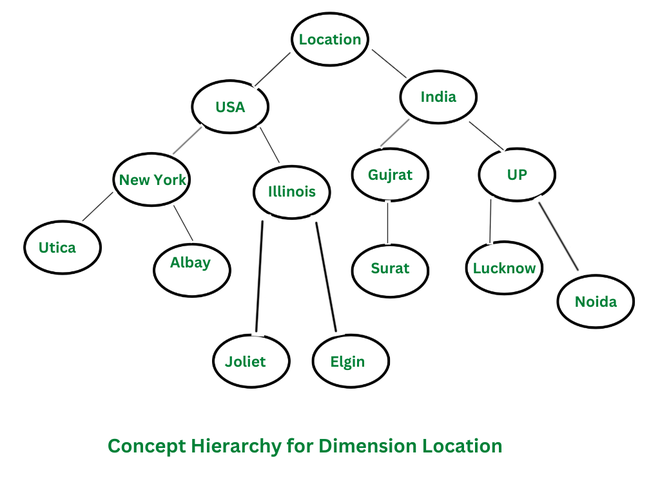
**Fact constellation:** Sophisticated applications may require multiple fact tables to *share* dimension tables. This kind of schema can be viewed as a collection of stars, andhence is called a **galaxy schema** or a **fact constellation**.

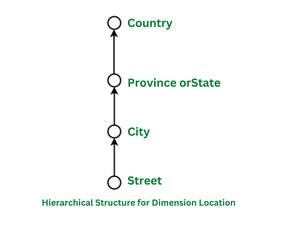
In data warehousing, there is a distinction between a data warehouse and a data mart. A data warehouse collects information about subjects that span the *entire organization*, such as *customers, items, sales, assets*, and *personnel*, and thus its scope is *enterprise-wide*. For data warehouses, the fact constellation schema is commonly used, since it can model multiple, interrelated subjects. A **data mart**, on the other hand, is a department subset of the data warehouse that focuses on selected subjects, and thus its scope is *departmentwide*. For data marts, the *star* or *snowflake* schema is commonly used, since both are geared toward modeling single subjects, although the star schema is more popular and efficient.

**Concept Hierarchy in Data Mining**

In data mining, the concept of a concept hierarchy refers to the organization of data into a tree-like structure, where each level of the hierarchy represents a concept that is more general than the level below it. This hierarchical organization of data allows for more efficient and effective data analysis, as well as the ability to drill down to more specific levels of detail when needed. The concept of hierarchy is used to organize and classify data in a way that makes it more understandable and easier to analyze. The main idea behind the concept of hierarchy is that the same data can have different levels of granularity or levels of detail and that by organizing the data in a hierarchical fashion, it is easier to understand and perform analysis.

**Example:**





**Explanation:**

As shown in the above diagram, it consists of a concept hierarchy for the dimension location, where the user can easily retrieve the data. In order to evaluate it easily the data is represented in a tree-like structure. The top of the tree consists of the main dimension location and further splits into various sub-nodes. The root node is located, and it further splits into two nodes countries ie. USA and India. These countries are further then splitted into more sub-nodes, that represent the province states ie. New York, Illinois, Gujarat, UP. Thus the concept hierarchy as shown in the above example organizes the data into a tree-like structure and describes and represents in more general than the level below it.

The hierarchical structure represents the abstraction level of the dimension location, which consists of various footprints of the dimension such as street, city, province state, and country.

**Types of Concept Hierarchies**

1. **Schema Hierarchy**:  Schema Hierarchy is a type of concept hierarchy that is used to organize the schema of a database in a logical and meaningful way, grouping similar objects together. A schema hierarchy can be used to organize different types of data, such as tables, attributes, and relationships, in a logical and meaningful way. This can be useful in data warehousing, where data from multiple sources needs to be integrated into a single database.
2. **Set-Grouping Hierarchy**: Set-Grouping Hierarchy is a type of concept hierarchy that is based on set theory, where each set in the hierarchy is defined in terms of its membership in other sets. Set-grouping hierarchy can be used for data cleaning, data pre-processing and data integration. This type of hierarchy can be used to identify and remove outliers, noise, or inconsistencies from the data and to integrate data from multiple sources.
3. **Operation-Derived Hierarchy**: An Operation-Derived Hierarchy is a type of concept hierarchy that is used to organize data by applying a series of operations or transformations to the data. The operations are applied in a top-down fashion, with each level of the hierarchy representing a more general or abstract view of the data than the level below it. This type of hierarchy is typically used in data mining tasks such as clustering and dimensionality reduction. The operations applied can be mathematical or statistical operations such as aggregation, normalization
4. **Rule-based Hierarchy**: Rule-based Hierarchy is a type of concept hierarchy that is used to organize data by applying a set of rules or conditions to the data. This type of hierarchy is useful in data mining tasks such as classification, decision-making, and data exploration. It allows to the assignment of a class label or decision to each data point based on its characteristics and identifies patterns and relationships between different attributes of the data.

**Need of Concept Hierarchy in Data Mining**

There are several reasons why a concept hierarchy is useful in data mining:

1. **Improved Data Analysis**: A concept hierarchy can help to organize and simplify data, making it more manageable and easier to analyze. By grouping similar concepts together, a concept hierarchy can help to identify patterns and trends in the data that would otherwise be difficult to spot. This can be particularly useful in uncovering hidden or unexpected insights that can inform business decisions or inform the development of new products or services.
2. **Improved Data Visualization and Exploration**: A concept hierarchy can help to improve data visualization and data exploration by organizing data into a tree-like structure, allowing users to easily navigate and understand large and complex data sets. This can be particularly useful in creating interactive dashboards and reports that allow users to easily drill down to more specific levels of detail when needed.
3. **Improved Algorithm Performance**: The use of a concept hierarchy can also help to improve the performance of data mining algorithms. By organizing data into a hierarchical structure, algorithms can more easily process and analyze the data, resulting in faster and more accurate results.
4. **Data Cleaning and Pre-processing**: A concept hierarchy can also be used in data cleaning and pre-processing, to identify and remove outliers and noise from the data.
5. **Domain Knowledge**: A concept hierarchy can also be used to represent the domain knowledge in a more structured way, which can help in a better understanding of the data and the problem domain.

**Applications of Concept Hierarchy**

There are several applications of concept hierarchy in data mining, some examples are:

1. **Data Warehousing**: Concept hierarchy can be used in data warehousing to organize data from multiple sources into a single, consistent and meaningful structure. This can help to improve the efficiency and effectiveness of data analysis and reporting.
2. **Business Intelligence**: Concept hierarchy can be used in business intelligence to organize and analyze data in a way that can inform business decisions. For example, it can be used to analyze customer data to identify patterns and trends that can inform the development of new products or services.
3. **Online Retail**: Concept hierarchy can be used in online retail to organize products into categories, subcategories and sub-subcategories, it can help customers to find the products they are looking for more quickly and easily.
4. **Healthcare**: Concept hierarchy can be used in healthcare to organize patient data, for example, to group patients by diagnosis or treatment plan, it can help to identify patterns and trends that can inform the development of new treatments or improve the effectiveness of existing treatments.
5. **Natural Language Processing**: Concept hierarchy can be used in natural language processing to organize and analyze text data, for example, to identify topics and themes in a text, it can help to extract useful information from unstructured data.
6. **Fraud** **Detection**: Concept hierarchy can be used in fraud detection to organize and analyze financial data, for example, to identify patterns and trends that can indicate fraudulent activity.

OLAP Operations

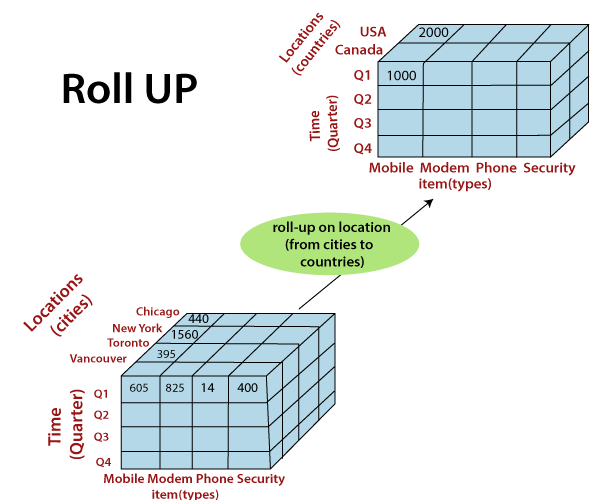
<https://www.javatpoint.com/olap-operations>

Roll-Up

The roll-up operation **(also known as drill-up or aggregation operation)**performs aggregation on a data cube, by climbing down concept hierarchies, i.e., dimension reduction. Roll-up is like **zooming-out** on the data cubes. Figure shows the result of roll-up operations performed on the dimension location. The hierarchy for the location is defined as the Order Street, city, province, or state, country. The roll-up operation aggregates the data by ascending the location hierarchy from the level of the city to the level of the country.

When a roll-up is performed by dimensions reduction, one or more dimensions are removed from the cube. For example, consider a sales data cube having two dimensions, location and time. Roll-up may be performed by removing, the time dimensions, appearing in an aggregation of the total sales by location, relatively than by location and by time.

The following diagram illustrates how roll-up works.



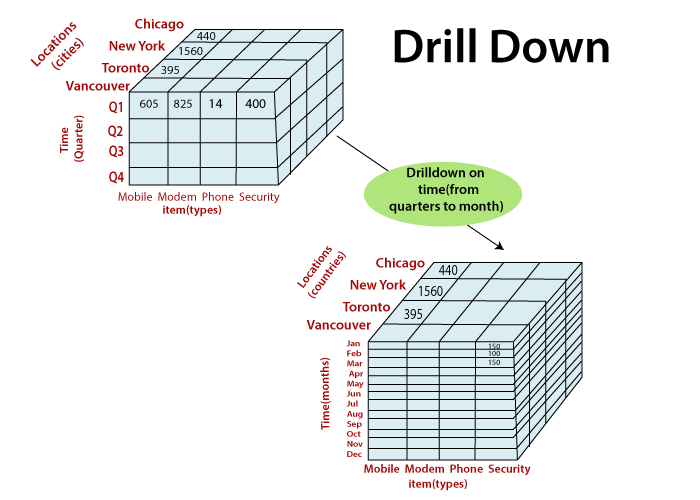
Drill-Down

The drill-down operation **(also called roll-down)** is the reverse operation of **roll-up**. Drill-down is like **zooming-in** on the data cube. It navigates from less detailed record to more detailed data. Drill-down can be performed by either **stepping down** a concept hierarchy for a dimension or adding additional dimensions.

Figure shows a drill-down operation performed on the dimension time by stepping down a concept hierarchy which is defined as day, month, quarter, and year. Drill-down appears by descending the time hierarchy from the level of the quarter to a more detailed level of the month.

Because a drill-down adds more details to the given data, it can also be performed by adding a new dimension to a cube. For example, a drill-down on the central cubes of the figure can occur by introducing an additional dimension, such as a customer group.

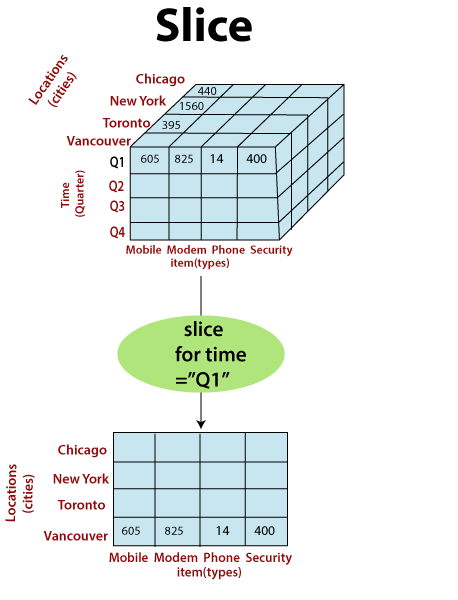
The following diagram illustrates how Drill-down works.



Slice

A **slice** is a subset of the cubes corresponding to a single value for one or more members of the dimension. For example, a slice operation is executed when the customer wants a selection on one dimension of a three-dimensional cube resulting in a two-dimensional site. So, the Slice operations perform a selection on one dimension of the given cube, thus resulting in a subcube.

**The following diagram illustrates how Slice works.**



Here Slice is functioning for the dimensions "time" using the criterion time = "Q1".

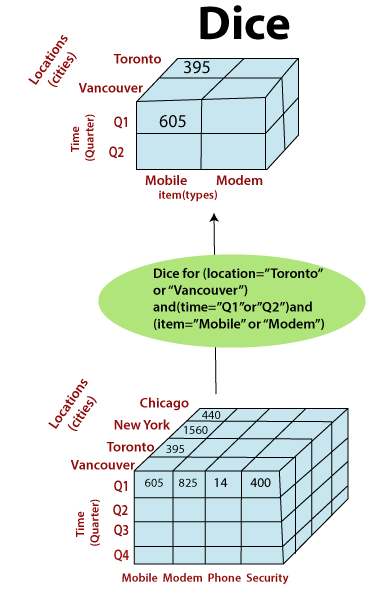
It will form a new sub-cubes by selecting one or more dimensions.

**Dice**

The dice operation describes a subcube by operating a selection on two or more dimension.

The dice operation on the cubes based on the following selection criteria involves three dimensions.

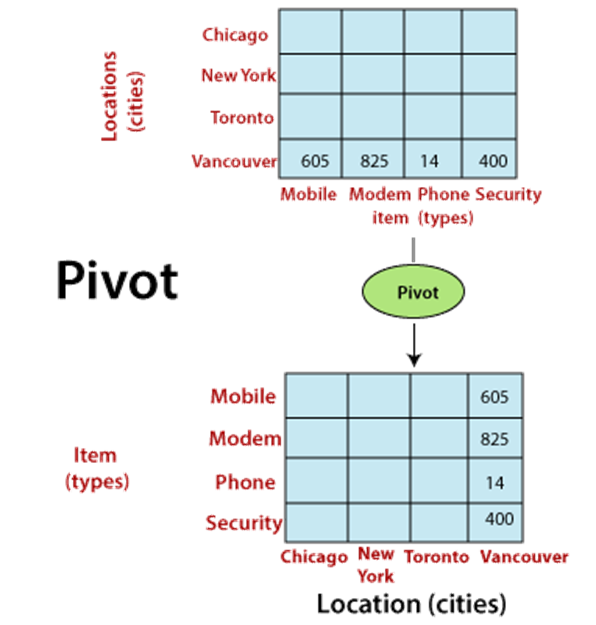
* (location = "Toronto" or "Vancouver")
* (time = "Q1" or "Q2")
* (item =" Mobile" or "Modem")



Pivot

The pivot operation is also called a rotation. Pivot is a visualization operations which rotates the data axes in view to provide an alternative presentation of the data. It may contain swapping the rows and columns or moving one of the row-dimensions into the column dimensions.

Consider the following diagram, which shows the pivot operation.



Other OLAP Operations

executes queries containing more than one fact table. The drill-through operations make use of relational SQL facilitates to drill through the bottom level of a data cubes down to its back-end relational tables.

Other OLAP operations may contain ranking the top-N or bottom-N elements in lists, as well as calculate moving average, growth rates, and interests, internal rates of returns, depreciation, currency conversions, and statistical tasks.

**Introduction to KDD(Knowledge Discovery in Databases) Process**

In the context of computer science, “Data Mining” can be referred to as knowledge mining from data, knowledge extraction, data/pattern analysis, data archaeology, and data dredging. Data Mining also known as Knowledge Discovery in Databases, refers to the nontrivial extraction of implicit, previously unknown and potentially useful information from data stored in databases.

The need of data mining is to extract useful information from large datasets and use it to make predictions or better decision-making. Nowadays, data mining is used in almost all places where a large amount of data is stored and processed.

For examples: Banking sector, Market Basket Analysis, Network Intrusion Detection.

**KDD Process**

KDD (Knowledge Discovery in Databases) is a process that involves the extraction of useful, previously unknown, and potentially valuable information from large datasets. The KDD process is an iterative process and it requires multiple iterations of the above steps to extract accurate knowledge from the data.

The following steps are included in KDD process:

**Data Cleaning**

Data cleaning is defined as removal of noisy and irrelevant data from collection.

1. Cleaning in case of **Missing values**.
2. Cleaning **noisy** data, where noise is a random or variance error.
3. Cleaning with **Data discrepancy detection** and **Data transformation tools**.

**Data Integration**

Data integration is defined as heterogeneous data from multiple sources combined in a common source(DataWarehouse). Data integration using **Data Migration tools, Data Synchronization tools and ETL**(Extract-Load-Transformation) process.

**Data Selection**

Data selection is defined as the process where data relevant to the analysis is decided and retrieved from the data collection. For this we can use  **Neural network, Decision Trees, Naive bayes, Clustering**, and **Regression**methods.

**Data Transformation**

Data Transformation is defined as the process of transforming data into appropriate form required by mining procedure. Data Transformation is a two step process:

1. **Data Mapping**: Assigning elements from source base to destination to capture transformations.
2. **Code generation**: Creation of the actual transformation program.

**Data Mining**

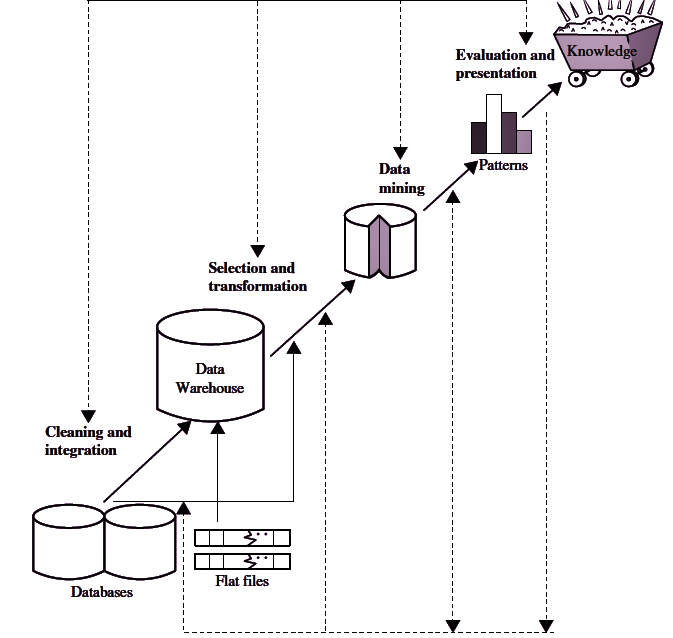
Data mining is defined as techniques that are applied to extract patterns potentially useful. It transforms task relevant data into **patterns, and d**ecides purpose of model using **classification** or **characterization**.

**Pattern Evaluation**

Pattern Evaluation is defined as identifying strictly increasing patterns representing knowledge based on given measures. It find **interestingness score** of each pattern, and uses **summarization** and **Visualization** to make data understandable by user.

**Knowledge Representation**

This involves presenting the results in a way that is meaningful and can be used to make decisions.



**Note**: KDD is an **iterative process** where evaluation measures can be enhanced, mining can be refined, new data can be integrated and transformed in order to get different and more appropriate results.**Preprocessing of databases** consists of **Data cleaning** and **Data Integration**.

**Advantages of KDD**

1. **Improves decision-making:** KDD provides valuable insights and knowledge that can help organizations make better decisions.
2. **Increased efficiency:** KDD automates repetitive and time-consuming tasks and makes the data ready for analysis, which saves time and money.
3. **Better customer service:** KDD helps organizations gain a better understanding of their customers’ needs and preferences, which can help them provide better customer service.
4. **Fraud detection:**KDD can be used to detect fraudulent activities by identifying patterns and anomalies in the data that may indicate fraud.
5. **Predictive modeling:**KDD can be used to build predictive models that can forecast future trends and patterns.

**Disadvantages of KDD**

1. **Privacy concerns:**KDD can raise privacy concerns as it involves collecting and analyzing large amounts of data, which can include sensitive information about individuals.
2. **Complexity:**KDD can be a complex process that requires specialized skills and knowledge to implement and interpret the results.
3. **Unintended consequences:** KDD can lead to unintended consequences, such as bias or discrimination, if the data or models are not properly understood or used.
4. **Data Quality:** KDD process heavily depends on the quality of data, if data is not accurate or consistent, the results can be misleading
5. **High cost:** KDD can be an expensive process, requiring significant investments in hardware, software, and personnel.
6. **Overfitting:** KDD process can lead to overfitting, which is a common problem in machine learning where a model learns the detail and noise in the training data to the extent that it negatively impacts the performance of the model on new unseen data.

**Data Mining**

Data mining is the process of extracting knowledge or insights from large amounts of data using various statistical and computational techniques. The data can be structured, semi-structured or unstructured, and can be stored in various forms such as databases, data warehouses, and data lakes.

The primary goal of data mining is to discover hidden patterns and relationships in the data that can be used to make informed decisions or predictions. This involves exploring the data using various techniques such as clustering, classification, regression analysis, association rule mining, and anomaly detection.

Data mining has a wide range of applications across various industries, including marketing, finance, healthcare, and telecommunications. For example, in marketing, data mining can be used to identify customer segments and target marketing campaigns, while in healthcare, it can be used to identify risk factors for diseases and develop personalized treatment plans.

**Data Mining on what kinds of data ?**

1. ***Relational Databases***
   * A [Relational database](https://www.geeksforgeeks.org/relational-model/) is defined as the collection of data organized in tables with rows and columns.
   * Physical schema in Relational databases is a schema which defines the structure of tables.
   * Logical schema in Relational databases is a schema which defines the relationship among tables.
   * Standard API of relational database is [SQL](https://www.geeksforgeeks.org/sql-tutorial/).

* A relational database is a type of structured data that organizes data into one or more tables, with each table consisting of rows and columns. The rows represent individual records, and the columns represent fields or attributes within those records.
* The main feature of a relational database is the ability to establish relationships between different tables using a common field called a primary key. This allows data to be linked and queried across multiple tables, enabling more efficient data retrieval and manipulation.
* Relational databases are widely used in many different industries, such as finance, healthcare, retail and e-commerce. They are also used to support transactional systems, data warehousing, and business intelligence.
* Relational databases are typically managed by a database management system (DBMS) such as MySQL, Oracle, SQL Server, and PostgreSQL. The DBMS provides tools for creating, modifying, and querying the database, as well as for managing access and security.

1. ***DataWarehouse***
   * A datawarehouse is defined as the collection of data integrated from multiple sources that will queries and decision making.
   * There are three types of datawarehouse: **Enterprise** datawarehouse, **Data Mart** and **Virtual** Warehouse.
   * Two approaches can be used to update data in DataWarehouse: **Query-driven** Approach and **Update-driven** Approach.
   * **Application**: Business decision making, Data mining, etc.
2. ***Transactional Databases***
   * Transactional databases is a collection of data organized by time stamps, date, etc to represent transaction in databases.
   * This type of database has the capability to roll back or undo its operation when a transaction is not completed or committed.
   * Highly flexible system where users can modify information without changing any sensitive information.
   * Follows [ACID property](https://www.geeksforgeeks.org/acid-properties-in-dbms/) of DBMS.
   * **Application**: Banking, Distributed systems, Object databases, etc.
3. ***Multimedia Databases***
   * Multimedia databases consists audio, video, images and text media.
   * They can be stored on Object-Oriented Databases.
   * They are used to store complex information in a pre-specified formats.
   * **Application**: Digital libraries, video-on demand, news-on demand, musical database, etc.
4. ***Spatial Database***
   * Store geographical information.
   * Stores data in the form of coordinates, topology, lines, polygons, etc.
   * **Application**: Maps, Global positioning, etc.
5. ***Time-series Databases***
   * Time series databases contains stock exchange data and user logged activities.
   * Handles array of numbers indexed by time, date, etc.
   * It requires real-time analysis.
   * **Application**: eXtremeDB, Graphite, InfluxDB, etc.
6. ***WWW***
   * WWW refers to World wide web is a collection of documents and resources like audio, video, text, etc which are identified by Uniform Resource Locators (URLs) through web browsers, linked by HTML pages, and accessible via the Internet network.
   * It is the most heterogeneous repository as it collects data from multiple resources.
   * It is dynamic in nature as Volume of data is continuously increasing and changing.
   * **Application**: Online shopping, Job search, Research, studying, etc.
7. **Structured Data:** This type of data is organized into a specific format, such as a database table or spreadsheet. Examples include transaction data, customer data, and inventory data.
8. **Semi-Structured Data:** This type of data has some structure, but not as much as structured data. Examples include XML and JSON files, and email messages.
9. **Unstructured Data:**This type of data does not have a specific format, and can include text, images, audio, and video. Examples include social media posts, customer reviews, and news articles.
10. **External Data:**This type of data is obtained from external sources such as government agencies, industry reports, weather data, satellite images, GPS data, etc.
11. **Time-Series Data:**This type of data is collected over time, such as stock prices, weather data, and website visitor logs.
12. **Streaming Data:**This type of data is generated continuously, such as sensor data, social media feeds, and log files.
13. **Relational Data:** This type of data is stored in a relational database, and can be accessed through SQL queries.
14. **NoSQL Data:** This type of data is stored in a NoSQL database, and can be accessed through a variety of methods such as key-value pairs, document-based, column-based or graph-based.
15. **Cloud Data**: This type of data is stored and processed in cloud computing environments such as AWS, Azure, and GCP.
16. **Big Data:**This type of data is characterized by its huge volume, high velocity, and high variety, and can be stored and processed using big data technologies such as Hadoop and Spark.

**Difference between KDD and Data Mining**

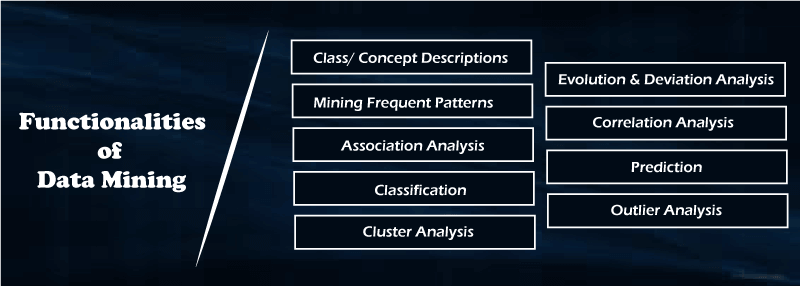
| **Parameter** | **KDD** | **Data Mining** |
| --- | --- | --- |
| Definition | KDD refers to a  process of identifying valid, novel, potentially useful, and ultimately understandable patterns and relationships in data. | Data Mining refers to a  process of extracting useful and valuable information or patterns from large data sets. |
| Objective | To find useful knowledge from data. | To extract useful information from data. |
| Techniques Used | Data cleaning, data integration, data selection, data transformation, data mining, pattern evaluation, and knowledge representation and visualization. | Association rules, classification, clustering, regression, decision trees, neural networks, and dimensionality reduction. |
| Output | Structured information, such as rules and models, that can be used to make decisions or predictions. | Patterns, associations, or insights that can be used to improve decision-making or understanding. |
| Focus | Focus is on the discovery of useful knowledge, rather than simply finding patterns in data. | Data mining focus is on the discovery of patterns or relationships in data. |
| Role of domain expertise | Domain expertise is important in KDD, as it helps in defining the goals of the process, choosing appropriate data, and interpreting the results. | Domain expertise is less critical in data mining, as the algorithms are designed to identify patterns without relying on prior knowledge. |

**Data Mining Functionalities**

Data mining functionalities specify the kind of patterns to be found in data mining tasks.In general, data mining tasks can be classified into two categories: descriptive and predictive.

Descriptive mining tasks characterize the general properties of the data in the target data set.

Predictive mining tasks perform inference on the current data in order to make predictions.

****

**Concept/Class Description**

Data can be associated with classes or concepts.

*Class :* A collection of things sharing a common attribute

Classes of items for sale include *computers and printers*

*Concept:* An abstract or general idea inferred or derived from specific instances  
 Concepts of customers include *bigSpenders and budgetSpenders.*

Summarized, concise and precise descriptions of individual classes and concepts are called *class/concept descriptions.*

These descriptions can be derived using

(1) *data characterization*, by summarizing the data of the class under study (often called the target class) in general terms, or

(2) *data discrimination*, by comparison of the target class with one or a set of comparative classes (often called the contrasting classes), or

(3) both data characterization and discrimination.

**Data characterization** is a summarization of the general characteristics or features of a target class of data.

The data corresponding to the user-specified class are typically collected by a query.

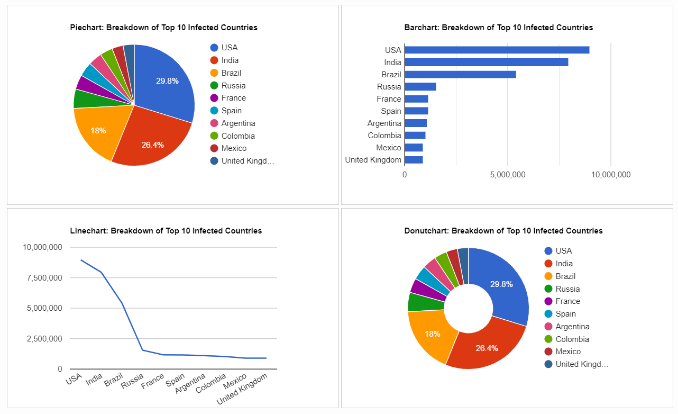
For example, to study the characteristics of **software products with sales that increased by 10% in the previous year**, the data related to such products can be collected by executing an SQL query on the sales database.

There are several methods for effective data summarization and characterization.

* Simple data summaries can be done based on statistical measures and plots.
* The data cube–based OLAP roll-up operation can be used to perform data summarization along a specified dimension.
* An *attribute-oriented induction* technique can be used to perform data generalization and characterization without step-by-step user interaction

The output of data characterization can be presented in various forms.

* **Eg: pie charts**, **bar charts**, **curves**, **multidimensional data cubes**, and **multidimensional tables**, including crosstabs.



* The resulting descriptions can also be presented as **generalized relations** or in rule form (called **characteristic rules**).

Eg: **Data characterization**

A customer relationship manager at *AllElectronics* may order the following data mining task:

*“Summarize the characteristics of customers who spend more than $5000 a year at AllElectronics*.”

The result is a general profile of these customers, such as that they are 40 to 50 years old, employed, and have excellent credit ratings.

The data mining system should allow the customer relationship manager to drill down on any dimension, such as on *occupation* to view these customers according to their type of employment

**Data discrimination**

* **Data discrimination** is a comparison of the general features of the target class data objects against the general features of objects from one or multiple contrasting classes.
* The target and contrasting classes can be specified by a user, and the corresponding data objects can be retrieved through database queries.

*For example, a user may want to compare the general features of software products with sales that increased by 10% last year against those with sales that decreased by at least 30% during the same period.*

* The methods used for data discrimination are similar to those used for data characterization.
* The forms of output presentation are similar to those for characteristic descriptions, although discrimination descriptions should include comparative measures that help to distinguish between the target and contrasting classes.
* Discrimination descriptions expressed in the form of rules are referred to as **discriminant rules**.

Eg:**Data discrimination**

*A customer relationship manager at AllElectronics may want to compare two groups of customers—those who shop for computer products regularly (e.g., more than twice a month) and those who rarely shop for such products (e.g., less than three times a year).*

The resulting description provides a general comparative profile of these customers, such as that

* *80% of the customers who frequently purchase computer products are between 20 and 40 years old and have a university education,*
* *whereas 60% of the customers who infrequently buy such products are either seniors or youths, and have no university degree.*

Drilling down on a dimension like occupation,or adding a new dimension like income level, may help to find even more discriminative features between the two classes.

**Mining Frequent Patterns**

**Frequent patterns** are patterns that occur frequently in data.

*Frequent itemset* - refers to a set of items that often appear together in a transactional data set;

*Eg: milk and bread, which are frequently bought together in grocery stores by many customers.*

*Sequential pattern* A frequently occurring subsequence.

*Eg:customers, tend to purchase first a laptop, followed by a digital camera, and then a memory card*

*Frequent substructure* refer to different structural forms (e.g., graphs, trees, or lattices) that may be combined with itemsets or subsequences. If a substructure occurs frequently, it is called a (*frequent*) *structured pattern*.

Mining frequent patterns leads to the discovery of interesting associations and correlations within data.

*Frequent itemset mining* is a fundamental form of frequent pattern mining.

**Association analysis**

Suppose that, as a marketing manager at *AllElectronics*, you want to know which items are frequently purchased together (i.e., within the same transaction).

An example of such a rule, mined from the *AllElectronics* transactional database, is:



where *X* is a variable representing a customer.

A **confidence**, or certainty, of 50% means that if a customer buys a computer, there is a 50% chance that she will buy software as well.

A 1% **support** means that 1% of all the transactions under analysis show that computer and software are purchased together.

The association rule involves a single attribute or predicate (i.e., *buys*) that repeats.

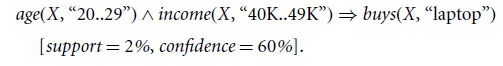
Association rules that contain a single predicate are referred to as **single-dimensional association rules**.

Dropping the predicate notation, the rule can be written simply as



Example: Multi dimensional Association rules

*AllElectronics* relational database related to purchases, a data mining system may find association rules like



Of the *AllElectronics* customers under study

* *2% are 20 to 29 years old with an income of $40,000 to $49,000 and have purchased a laptop (computer) at AllElectronics.*
* *There is a 60% probability that a customer in this age and income group will purchase a laptop.*

An association involving more than one attribute or predicate (i.e., *age, income*, and *buys*).

Each attribute is referred to as a dimension-> referred to as a ***multidimensional association rule.***

**Classification and Regression for Predictive Analysis**

Classification

* ***Classification*** is the process of finding a ***model*** (or function) that describes and distinguishes data classes or concepts.
* The model are derived based on the analysis of a set of ***training data*** (i.e., data objects for which the class labels are known).
* The model is *used to predict the class label of objects* for which the the class label is unknown.
* How is the derived model presented?
  + *classification rules* (i.e., *IF-THEN rules*)
  + *Decision trees*
  + *Mathematical formulae*
  + *neural networks*

**Decision tree**

* A flowchart-like tree structure
* Each node denotes a test on an attribute value
* Each branch represents an outcome of the test
* Tree leaves represent classes or class distributions.
* Decision trees can easily be converted to classification rules.

N**eural network**

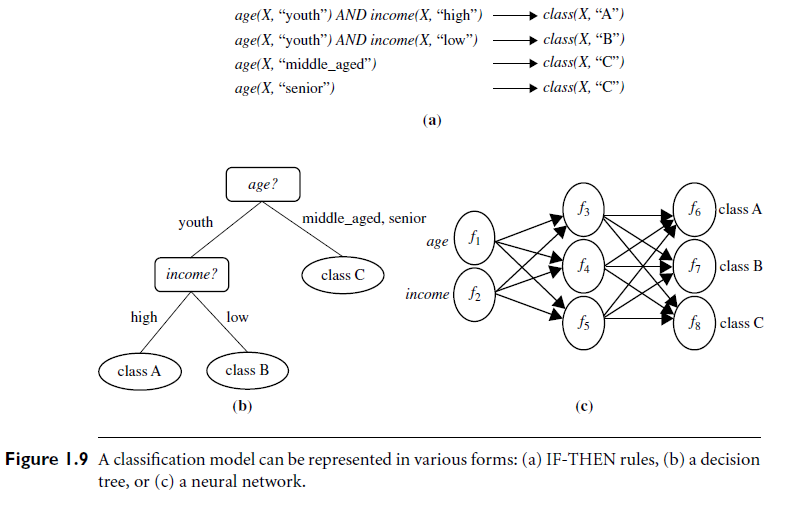
* used for classification
* A collection of neuron-like processing units with weighted connections between the units.

**Other Classification Models:**

Naïve Bayesian classification

Support Vector Machines

*k*-nearest-neighbor classification.



Eg: Classification

Suppose as a sales manager of *AllElectronics* you want to classify a large set of items in the store, based on three kinds of responses to a sales campaign:*good response*, *mild response* and *no response*.

Derive a model for each of these three classes based on the descriptive features of the items, such as *price*, *brand, place made, type*, and *category*.

The resulting classification should maximally distinguish each class from the others, presenting an organized picture of the data set.

The resulting classification is expressed as a decision tree. The decision tree, for instance, may identify *price* as being the single factor that best distinguishes the three classes. The tree may reveal that, in addition to *price*, other features that help to further distinguish objects of each class fromone another include *brand* and *place made*. Such a decision tree may help you understand the impact of the given sales campaign and design a more effective campaign in the future.

**Regression**

Regression models continuous-valued functions.

Used to predict missing or unavailable *numerical data values* rather than (discrete) class labels.

*Prediction* -> both numeric prediction and class label prediction.

*Regression analysis* - a statistical methodology that is most often used for numeric prediction.

Regression also encompasses the identification of distribution *trends* based on the available data.

**Relevance analysis**

* Classification and regression may need to be preceded by **relevance analysis.**
* Attempts to identify attributes that are significantly relevant to the classification and regression process.
* Other attributes, which are irrelevant, can then be excluded from consideration.

Eg: Regression

* Predict the amount of revenue that each item will generate during an upcoming sale at *AllElectronics*, based on the previous sales data.
* An example of regression analysis because the regression model constructed will predict a continuous function (or ordered value.)

Cluster Analysis

* **Clustering** analyzes data objects without consulting class labels.
* Clustering can be used to generate class labels for a group of data.
* The objects are clustered or grouped based on the principle of *maximizing the intraclass similarity and minimizing the interclass similarity*.
* Objects within a cluster have high similarity in comparison to one another, but are rather dissimilar to objects in other clusters.
* Each cluster so formed can be viewed as a class of objects, from which rules can be derived.
* Clustering facilitate **taxonomy formation ->** the organization of observations into a hierarchy of classes that group similar events together.

Example:

* Cluster analysis can be performed on *AllElectronics* customer data to identify homogeneous subpopulations of customers.
* These clusters may represent individual target groups for marketing.
* Three clusters of data points are evident.
* 

**Outlier Analysis**

* A data set may contain objects that do not comply with the general behavior or model of the data- ***Outliers.***
* Many data mining methods discard outliers as noise or exceptions.
* In some applications the rare events can be more interesting than the more regularly occurring ones.
* The analysis of outlier data is referred to as **outlier analysis** or **anomaly mining**.
* Detected using:
  + statistical tests that assume a distribution or probability model for the data
  + distance measures where objects that are remote from any other cluster are considered outliers.
  + Density-based methods may identify outliers in a local region, although they look normal from a global statistical distribution view.

**Example -Outlier analysis**

* Outlier analysis may uncover fraudulent usage of credit cards by detecting purchases of unusually large amounts for a given account number in comparison to regular charges incurred by the same account.
* Outlier values may also be detected with respect to the locations and types of purchase, or the purchase frequency.

**Are All Patterns Interesting?**

A data mining system has the potential to generate thousands or even millions of patterns, or rules.

*“Are all of the patterns interesting?” N*o—only a small fraction of the patterns potentially generated would actually be of interest to a given user.

This raises some serious questions for data mining.

*“What makes a pattern interesting?*

*Can a data mining system generate all of the interesting patterns?*

*Can the system generate only the interesting ones?”*

A pattern is **interesting** if it is

(1) *easily understood* by humans

(2) *valid* on new or test data with some degree of *certainty*

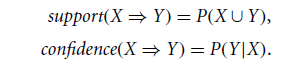
(3) potentially *useful*

(4) *novel*.

A pattern is also interesting if it validates a hypothesis that the user *sought to confirm*. An interesting pattern represents **knowledge**.

**Objective measures of pattern interestingness**

* Based on the structure of discovered patterns and the statistics underlying them.
* An objective measure for association rules of the form *X* =>*Y* is rule **Support**,
* Represent the percentage of transactions from a transaction database that the given rule satisfies.
* This is taken to be the probability *P*(*X U Y)*, where *X* U *Y* indicates that a transaction contains both *X* and *Y*, that is, the union of itemsets *X* and *Y*.
* **Confidence**, which assesses the degree of certainty of the detected association.
* This is taken to be the conditional probability *P*(*Y*|*X*), that is, the probability that a transaction containing *X* also contains *Y*.
* More formally, support and confidence are defined as



* **Accuracy** -the percentage of data that are correctly classified by a rule.
* **Coverage** is similar to support- the percentage of data to which a rule applies.
* Although objective measures help identify interesting patterns, they are often insufficient unless combined with subjective measures that reflect a particular user’s needs and interests.
* For example, *patterns describing the characteristics of customers who shop frequently at AllElectronics should be interesting to the marketing manager, but may be of little interest to other analysts studying the same database for patterns on employee performance.*
* Many patterns that are interesting by objective standards may represent common sense and, therefore, are actually uninteresting.

**Subjective interestingness measures**

* Based on user beliefs in the data.
* These measures find patterns interesting if the patterns are **unexpected** (contradicting a user’s belief) or offer strategic information on which the user can act(**actionable patterns)**.
* For example, patterns like “a large earthquake often follows a cluster of small quakes” may be highly actionable if users can act on the information to save lives.
* Patterns that are **expected** can be interesting if they confirm a hypothesis that the user wishes to validate or they resemble a user’s hunch.
* Eg: During a clinical trial for a new medication, researchers might expect the medication group to show improvement in certain symptoms compared to the placebo group. Observing this expected pattern strengthens the evidence for the medication's effectiveness.

*Can a data mining system generate all of the interesting patterns?*

* Refers to the **completeness** of a data mining algorithm.
* It is often unrealistic and inefficient for data mining systems to generate all possible patterns.
* Instead, user provided constraints and interestingness measures should be used to focus the search.
* For some mining tasks, such as association, this is often sufficient to ensure the completeness of the algorithm.
* Association rule mining is an example where the use of constraints and interestingness measures can ensure the completeness of mining.

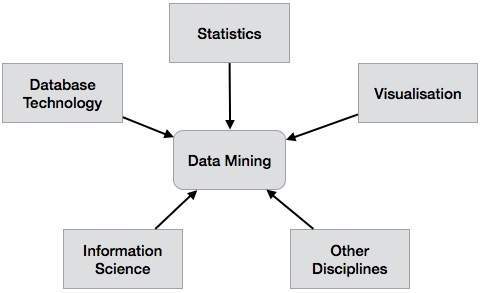
*Can a data mining system generate only interesting patterns?*

* An optimization problem in data mining.
* It is highly desirable for data mining systems to generate only interesting patterns.
* Users and data mining systems would have to search through the patterns generated to identify the truly interesting ones.
* Progress made but optimization remains a challenging issue in data mining.
* Measures of pattern interestingness are essential for the efficient discovery of patterns by target users.
* Such measures can be used after the data mining step to rank the discovered patterns according to their interestingness, filtering out the uninteresting ones.
* Can be used to guide and constrain the discovery process, improving the search efficiency by pruning away subsets of the pattern space that do not satisfy pre-specified interestingness constraints.

Data Mining System Classification

A data mining system can be classified according to the following criteria −

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



Apart from these, a data mining system can also be classified based on the kind of (a) databases mined, (b) knowledge mined, (c) techniques utilized, and (d) applications adapted.

**Classification Based on the Databases Mined**

We can classify a data mining system according to the kind of databases mined. Database system can be classified according to different criteria such as data models, types of data, etc. And the data mining system can be classified accordingly.

For example, if we classify a database according to the data model, then we may have a relational, transactional, object-relational, or data warehouse mining system.

**Classification Based on the kind of Knowledge Mined**

We can classify a data mining system according to the kind of knowledge mined. It means the data mining system is classified on the basis of functionalities such as −

* Characterization
* Discrimination
* Association and Correlation Analysis
* Classification
* Prediction
* Outlier Analysis
* Evolution Analysis

**Classification Based on the Techniques Utilized**

We can classify a data mining system according to the kind of techniques used. We can describe these techniques according to the degree of user interaction involved or the methods of analysis employed.

Machine learning, visualization, pattern recognition, neural networks, database-oriented or data-warehouse oriented techniques.

Classification by User Interaction:

* + **Supervised Learning:** Decision Trees, Support Vector Machines (SVMs)
  + **Unsupervised Learning**: Clustering, Association Rule Learning

Classification by Analysis Methods:

* + **Statistical Techniques:** Linear Regression, Logistic Regression
  + **Machine Learning Techniques:** Artificial Neural Networks (ANNs), Random Forests

**Classification Based on the Applications Adapted**

We can classify a data mining system according to the applications adapted. These applications are as follows −

* Finance
* Telecommunications
* DNA
* Stock Markets
* E-mail