LECTURE NOTES ON

DATA MINING& DATA WAREHOUSING

Chapter-1

1.1 What Is Data Mining?

Data mining refers to extracting or mining knowledge from large amountsof data. The term is actually a misnomer. Thus, data miningshould have been more appropriately named as knowledge mining which emphasis on mining 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.

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

1.2 The Scope of Data Mining

Data mining derives its name from the similarities between searching for valuable business information in a large database — for example, finding linked products in gigabytes of store scanner data — and mining a mountain for a vein of valuable ore. Both processes require either sifting through an immense amount of material, or intelligently probing it to find exactly where the value resides. Given databases of sufficient size and quality, data mining technology can generate new business opportunities by providing these capabilities:

Automated prediction of trends and behaviors. Data mining automates the process of finding predictive information in large databases. Questions that traditionally required extensive hands-on analysis can now be answered directly from the data — quickly. A typical example of a predictive problem is targeted marketing. Data mining uses data on past promotional mailings to identify the targets most likely to maximize return on investment in future mailings. Other predictive problems include forecasting bankruptcy and other forms of default, and identifying segments of a population likely to respond similarly to given events.

Automated discovery of previously unknown patterns. Data mining tools sweep through databases and identify previously hidden patterns in one step. An example of pattern discovery is the analysis of retail sales data to identify seemingly unrelated products that are often purchased together. Other pattern discovery problems include detecting fraudulent credit card transactions and identifying anomalous data that could represent data entry keying errors.

1.3 Tasks of Data Mining

Data mining involves six common classes of tasks:

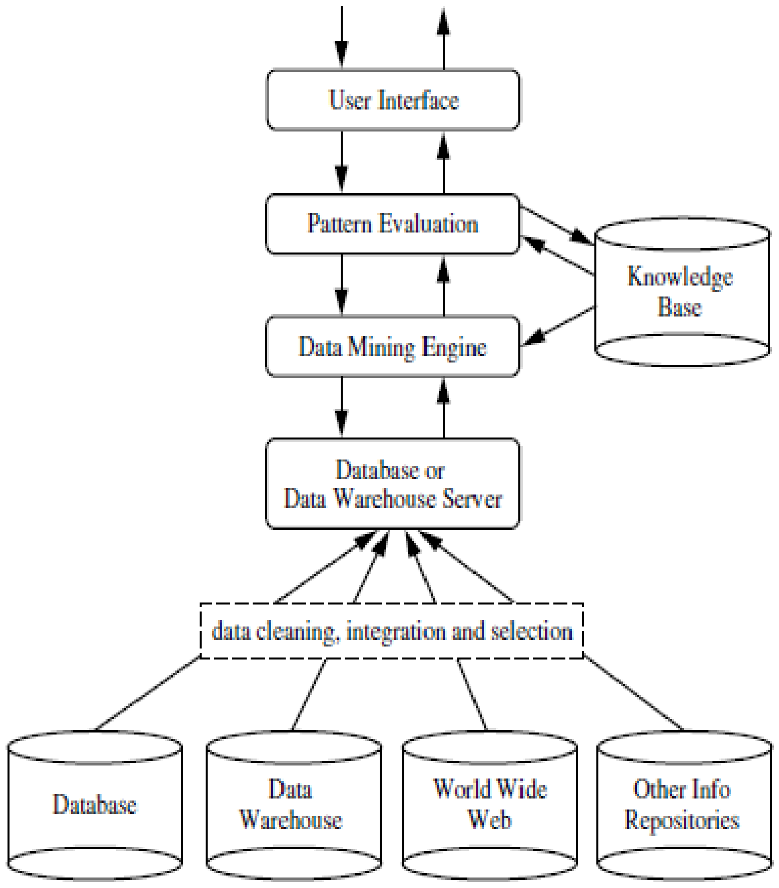
- **Anomaly detection (Outlier/change/deviation detection)** – The identification of unusual data records, that might be interesting or data errors that require further investigation.
- **Association rule learning (Dependency modelling)** – Searches for relationships between variables. For example a 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 which models the data with the least error.

- **Summarization** – providing a more compact representation of the data set, including visualization and report generation.

1.4 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. Other examples of domain knowledge are additional interestingness constraints or thresholds, and metadata (e.g., describing data from multiple heterogeneous sources).

2. **Data Mining Engine:**

   This is essential to the data mining systemand ideally consists ofa set of functional modules for tasks such as characterization, association and correlationanalysis, classification, prediction, cluster analysis, outlier analysis, and evolutionanalysis.

3. **Pattern Evaluation Module:**

   This component typically employs interestingness measures interacts with the data mining modules so as to focus thesearch toward interesting patterns. It may use interestingness thresholds to filterout discovered patterns. Alternatively, the pattern evaluation module may be integratedwith the mining module, depending on the implementation of the datamining method used. For efficient data mining, it is highly recommended to pushthe evaluation of pattern interestingness as deep as possible into the mining processso as to confine the search to only the interesting patterns.

4. **User interface:**

   Thismodule communicates between users and the data mining system,allowing the user to interact with the system by specifying a data mining query ortask, providing information to help focus the search, and performing exploratory datamining based on the intermediate data mining results. In addition, this componentallows the user to browse database and data warehouse schemas or data structures,evaluate mined patterns, and visualize the patterns in different forms.

1.5 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 this step, a modeler usually specifies a set of variables for the unknown dependency and, if possible, a general form of this dependency as an initial hypothesis. There may be several hypotheses formulated for a single problem at this stage. The first step requires the combined expertise of an application domain and a data-mining model. 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. An observational setting, namely, random data generation, is assumed in most data-mining applications. Typically, the sampling distribution is completely unknown after data are collected, or it is partially and implicitly given in the data-collection procedure. It is very important, however, to understand how data collection affects its theoretical distribution, since such a priori knowledge can be very useful for modeling and, later, for the final interpretation of results. Also, it is important to make sure that the data used for estimating a model and the data used later for testing and applying a model come from the same, unknown, sampling distribution. If this is not the case, the estimated model cannot be successfully used in a final application of the results.

3. Preprocessing the data

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

1. **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.

2. 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. Also, application-specific encoding methods usually achieve dimensionality reduction by providing a smaller number of informative features for subsequent data modeling.

These two classes of preprocessing tasks are only illustrative examples of a large spectrum of preprocessing activities in a data-mining process.

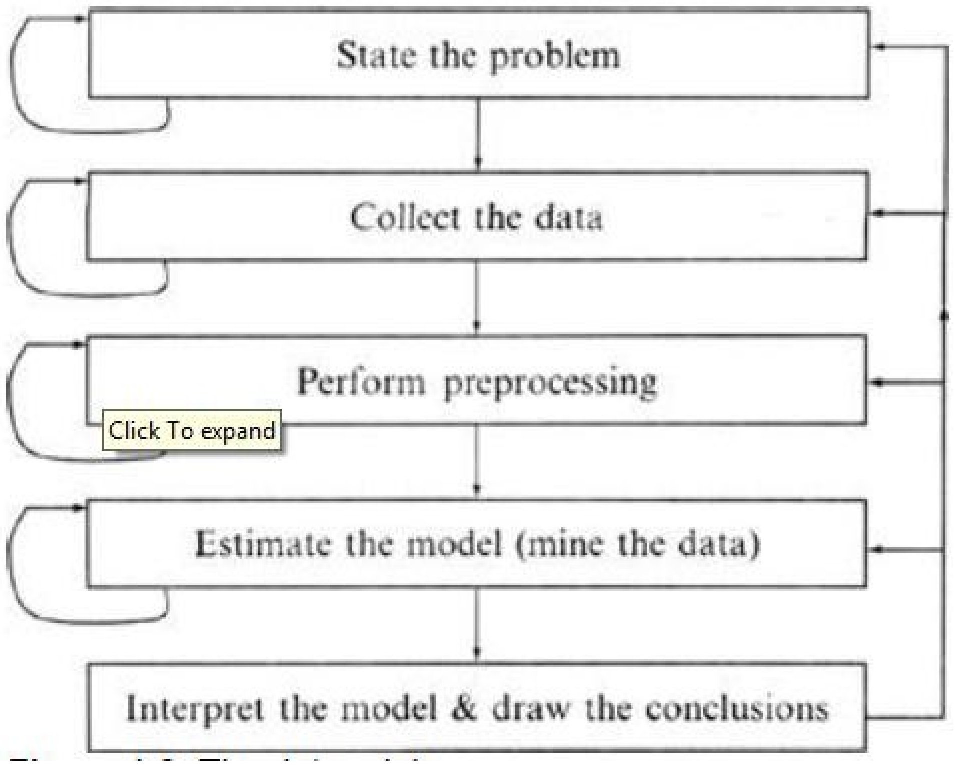
Data-preprocessing steps should not be considered completely independent from other data-mining phases. In every iteration of the data-mining process, all activities, together, could define new and improved data sets for subsequent iterations. Generally, a good preprocessing method provides an optimal representation for a data-mining technique by incorporating a priori knowledge in the form of application-specific scaling and encoding.

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. The basic principles of learning and discovery from data are given in Chapter 4 of this book. Later, Chapter 5 through 13 explain and analyze specific techniques that are applied to perform a successful learning process from data and to develop an appropriate model.

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 in order to be useful because humans are not likely to base their decisions on complex "black-box" models. Note that the goals of accuracy of the model and accuracy of its interpretation are somewhat contradictory. Usually, simple models are more interpretable, but they are also less accurate. Modern data-mining methods are expected to yield highly accurate results using highdimensional models. The problem of interpreting these models, also very important, is considered a separate task, with specific techniques to validate the results. A user does not want hundreds of pages of numeric results. He does not understand them; he cannot summarize, interpret, and use them for successful decision making.


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

The Data mining Process

1.6 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:

- Classification according to kind of databases mined
- Classification according to kind of knowledge mined
- Classification according to kinds of techniques utilized
- Classification according to applications adapted

Classification according to kind of databases mined

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

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

Classification according to kinds of techniques utilized

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

Classification according to applications adapted

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

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

1.7 Major Issues In Data Mining:

•**Mining different kinds of knowledge in databases.** - The need of different users is not the same. And Different user may be in interested in different kind of knowledge. Therefore it is necessary for data mining to cover broad range of knowledge discovery task.

•**Interactive mining of knowledge at multiple levels of abstraction.** - The data mining process needs to be interactive because it allows users to focus the search for patterns, providing and refining data mining requests based on returned results.

•**Incorporation of background knowledge.** - To guide discovery process and to express the discovered patterns, the background knowledge can be used. Background knowledge may be used to express the discovered patterns not only in concise terms but at multiple level of abstraction.

- **Data mining query languages and ad hoc data mining.** - Data Mining Query language that allows the user to describe ad hoc mining tasks, should be integrated with a data warehouse query language and optimized for efficient and flexible data mining.

- **Presentation and visualization of data mining results.** - Once the patterns are discovered it needs to be expressed in high level languages, visual representations. This representations should be easily understandable by the users.

- **Handling noisy or incomplete data.** - The data cleaning methods are required that can handle the noise, incomplete objects while mining the data regularities. If data cleaning methods are not there then the accuracy of the discovered patterns will be poor.

- **Pattern evaluation.** - It refers to interestingness of the problem. The patterns discovered should be interesting because either they represent common knowledge or lack novelty.

- **Efficiency and scalability of data mining algorithms.** - In order to effectively extract the information from huge amount of data in databases, data mining algorithm must be efficient and scalable.

- **Parallel, distributed, and incremental mining algorithms.** - The factors such as huge size of databases, wide distribution of data,and complexity of data mining methods motivate the development of parallel and distributed data mining algorithms. These algorithm divide the data into partitions which is further processed parallel. Then the results from the partitions is merged. The incremental algorithms, updates databases without having mine the data again from scratch.

KDD- Knowledge Discovery in Databases

The term KDD stands for Knowledge Discovery in Databases. It refers to the broad procedure of discovering knowledge in data and emphasizes the high-level applications of specific Data Mining techniques. It is a field of interest to researchers in various fields, including artificial intelligence, machine learning, pattern recognition, databases, statistics, knowledge acquisition for expert systems, and data visualization.

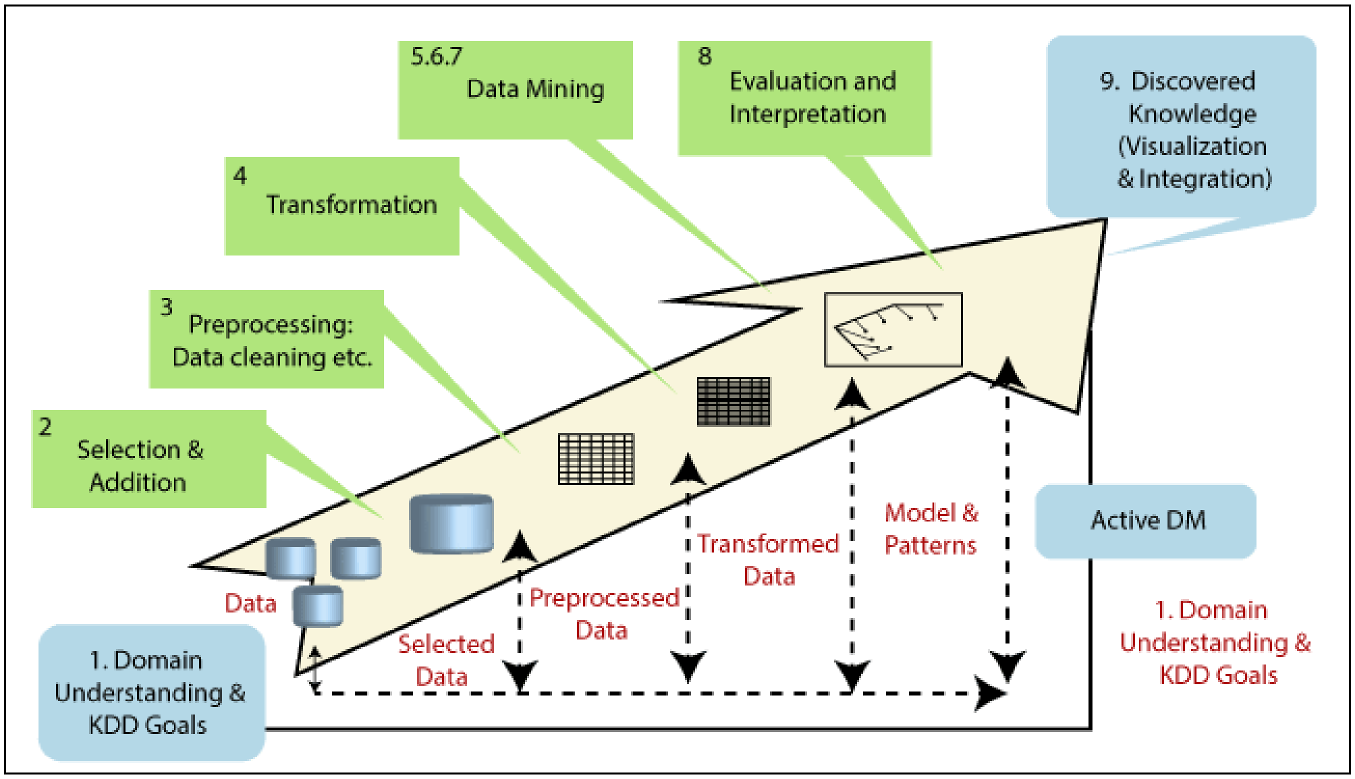
The main objective of the KDD process is to extract information from data in the context of large databases. It does this by using Data Mining algorithms to identify what is deemed knowledge.

KDD is the organized procedure of recognizing valid, useful, and understandable patterns from huge and complex data sets. Data Mining is the root of the KDD procedure, including the inferring of algorithms that investigate the data, develop the model, and find previously unknown patterns. The model is used for extracting the knowledge from the data, analyze the data, and predict the data.

The KDD Process

The knowledge discovery process(illustrates in the given figure) is iterative and interactive, comprises of nine steps. The process is iterative at each stage, implying that moving back to the previous actions might be required. The process has many imaginative aspects in the sense that one cant presents one formula or make a complete scientific categorization for the correct decisions for each step and application type. Thus, it is needed to understand the process and the different requirements and possibilities in each stage.

The process begins with determining the KDD objectives and ends with the implementation of the discovered knowledge. At that point, the loop is closed, and the Active Data Mining starts. Subsequently, changes would need to be made in the application domain. For example, offering various features to cell phone users in order to reduce churn. This closes the loop, and the impacts are then measured on the new data repositories, and the KDD process again. Following is a concise description of the nine-step KDD process, Beginning with a managerial step:


5.6.7 Data Mining
8 Evaluation and Interpretation
9. Discovered Knowledge (Visualization & Integration)
4 Transformation
3 Preprocessing: Data cleaning etc.
2 Selection & Addition
Active DM
Model & Patterns
Transformed Data
Preprocessed Data
Selected Data
Data
1. Domain Understanding & KDD Goals
1. Domain Understanding & KDD Goals

1. Building up an understanding of the application domain

This is the initial preliminary step. It develops the scene for understanding what should be done with the various decisions like transformation, algorithms, representation, etc. The individuals who are in charge of a KDD venture need to understand and characterize the objectives of the end-user and the environment in which the knowledge discovery process will occur (involves relevant prior knowledge).

2. Choosing and creating a data set on which discovery will be performed

Once objectives are defined, the data for the knowledge discovery process must be identified, collected, and integrated. This includes determining accessible data, selecting relevant attributes, and preparing a comprehensive dataset. Adequate data is crucial as it forms the foundation for effective models in data mining. Missing key attributes can jeopardize the study, but managing advanced data repositories can be costly. The iterative nature of KDD allows for starting with available data, refining it, and evaluating its impact on knowledge discovery and modeling.

3. Preprocessing and cleansing

This step enhances data reliability through cleaning, handling missing values, and removing noise or outliers, potentially using statistical techniques or Data Mining algorithms. For example, when one suspects that a specific attribute of lacking reliability or has many missing data, at this point, this attribute could turn into the objective of the Data Mining supervised algorithm. A prediction model for these attributes will be created, and after that, missing data can be predicted.

4. Data Transformation

This stage prepares data for Data Mining through techniques like dimension reduction (e.g., feature selection, sampling) and attribute transformation (e.g., discretization, functional changes). These steps are critical and project-specific, influencing success significantly. For instance, in medical studies, attribute ratios may matter more than individual attributes, while in business, external factors like advertising may need consideration. Missteps in transformation can guide refinements in subsequent iterations, emphasizing the iterative nature of the KDD process.

5. Prediction and description

At this stage, the type of Data Mining—such as classification, regression, or clustering—is chosen based on the KDD objectives and prior steps. Data Mining serves two main goals: prediction (supervised learning) and description (unsupervised or visualization). Most techniques use inductive learning, where models generalize from training data to apply to future cases. The approach may also include meta-learning tailored to the available data.

6. Selecting the Data Mining algorithm

This stage involves selecting strategies and specific techniques for pattern discovery, balancing factors like precision (e.g., neural networks) and interpretability (e.g., decision trees). Meta-learning helps identify conditions where a Data Mining algorithm is most effective. Algorithms also require parameter tuning and validation strategies, such as cross-validation, to optimize performance.

7. Utilizing the Data Mining algorithm

At last, the implementation of the Data Mining algorithm is reached. In this stage, we may need to utilize the algorithm several times until a satisfying outcome is obtained. For example, by turning the algorithms control parameters, such as the minimum number of instances in a single leaf of a decision tree.

8. Evaluation

In this step, we assess and interpret the mined patterns, rules, and reliability to the objective characterized in the first step. Here we consider the preprocessing steps as for their impact on the Data Mining algorithm results. For example, including a feature in step 4, and repeat from there. This step focuses on the comprehensibility and utility of the induced model. In this step, the identified knowledge is also recorded for further use. The last step is the use, and overall feedback and discovery results acquire by Data Mining.

9. Using the discovered knowledge

In this step, the discovered knowledge is integrated into a system for further action, allowing changes to be made and their impacts measured. This stage determines the effectiveness of the entire KDD process but comes with challenges. For instance, the static dataset used in discovery may now become dynamic, with changes in data structures, availability, or unexpected attribute values

1.8 Data Warehouse:

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.

1.8.1 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

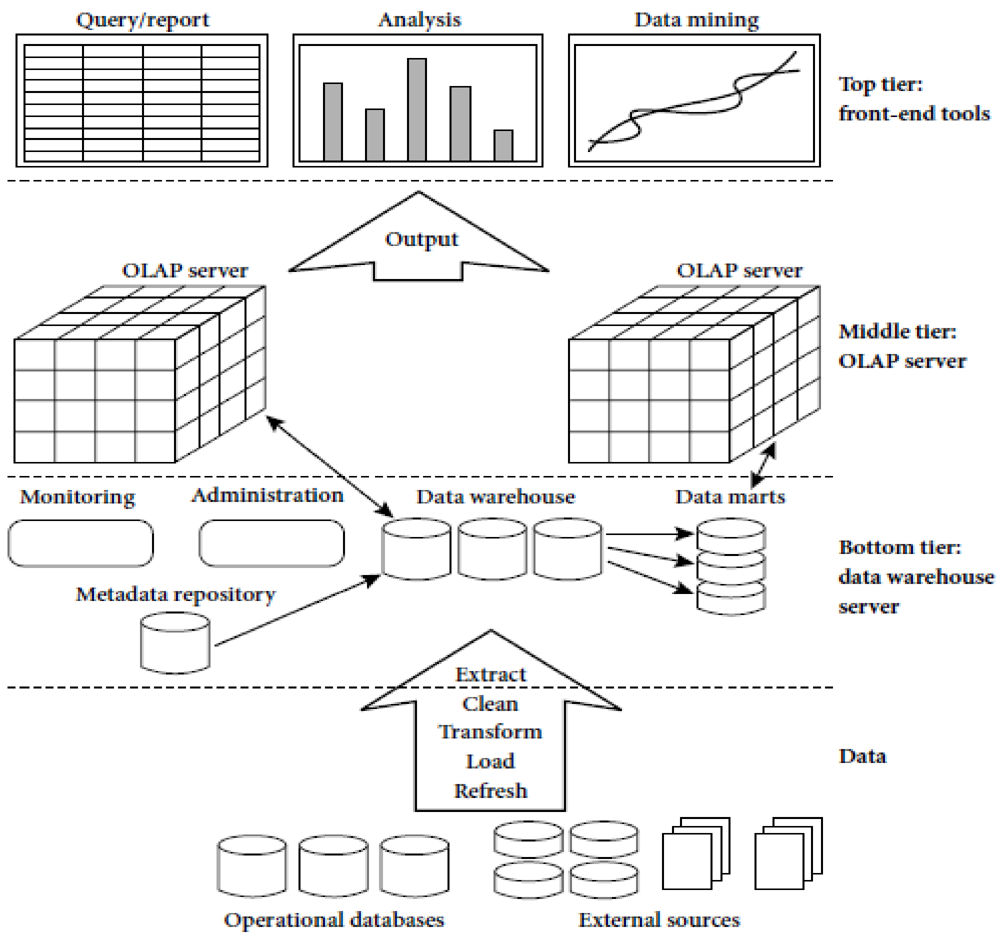
making significant commitments.

- 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.

1.8.2 A Three Tier Data Warehouse Architecture:



Tier-1:

The bottom tier is a warehouse database server that is almost always a relationaldatabase system. Back-end tools and utilities are used to feed data into the bottomtier from operational databases or other external sources (such as customer profileinformation provided by external consultants). These tools and utilities performdataextraction, cleaning, and transformation (e.g., to merge similar data from differentsources into a unified format), as well as load and refresh functions to update thedata warehouse . The data are extracted using application programinterfaces 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 Linkingand Embedding for Databases) by Microsoft and JDBC (Java Database Connection).

This tier also contains a metadata repository, which stores information aboutthe 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 thatmaps 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).

1.8.3 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. Thedata contained in data marts tend to be summarized.

- Data marts are usually implemented on low-cost departmental servers that areUNIX/LINUX- or Windows-based. The implementation cycle of a data mart ismore likely to be measured in weeks rather than months or years. However, itmay involve complex integration in the long run if its design and planning werenot enterprise-wide.

- Depending on the source of data, data marts can be categorized as independent ordependent. Independent data marts are sourced fromdata captured fromone or moreoperational systems or external information providers, or fromdata generated locallywithin a particular department or geographic area. Dependent data marts are sourceddirectly from enterprise data warehouses.

3. Virtual warehouse:

- A virtual warehouse is a set of views over operational databases. Forefficient 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.

1.8.4 Meta Data Repository

Metadata are data about data.When used in a data warehouse, metadata are the data thatdefine 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 include data lineage (history of migrated data and the sequence of transformations applied to it), 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 definitionalgorithms, data on granularity, partitions, subject areas, aggregation, summarization,and predefined queries and reports.
- The mapping from the operational environment to the data warehouse, which includessource databases and their contents, gateway descriptions, data partitions, data extraction, cleaning, transformation rules and defaults, data refresh and purging rules,

andsecurity (user authorization and access control).

- Data related to system performance, which include indices and profiles that improvedata access and retrieval performance, in addition to rules for the timing and scheduling of refresh, update, and replication cycles.
- Business metadata, which include business terms and definitions, data ownershipinformation, and charging policies.

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.

OLAP consists of three basic analytical operations:

- Consolidation (Roll-Up)
- Drill-Down
- Slicing And Dicing

- Consolidation involves the aggregation of data that can be accumulated and computed in one or more dimensions. For example, all sales offices are rolled up to the sales department or sales division to anticipate sales trends.
- The drill-down is a technique that allows users to navigate through the details. For instance, users can view the sales by individual products that make up a region’s sales.
- Slicing and dicing is a feature whereby users can take out (slicing) a specific set of data of the OLAP cube and view (dicing) the slices from different viewpoints.

1.9.1 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 limit 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.

Online Analytical Processing (OLAP)

- Online Analytical Processing Server (OLAP) is based on the multidimensional data model.
- It allows managers, and analysts to get an insight of the information through fast, consistent, and interactive access to information.
- OLAP facilitates users to extract and present multidimensional data from different view.
- OLAP provides a user-friendly environment for interactive data analysis.

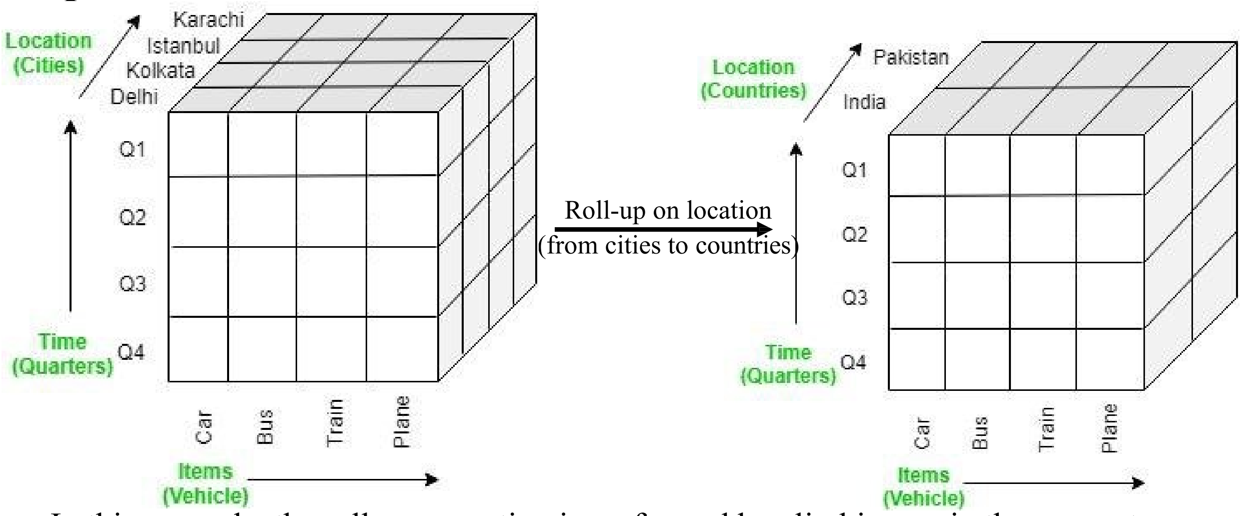
OLAP Operations

There are five basic analytical operations that can be performed on an OLAP cube:

1. ***Roll -up (Drill-up):*** The roll-up operation performs aggregation on a data cube, either by climbing up a concept hierarchy for a dimension or by dimension reduction. When roll up operation is performed one or more dimension is removed from the given cube.

   Note: *Concept hierarchy is a system of grouping things based on their order or level.*

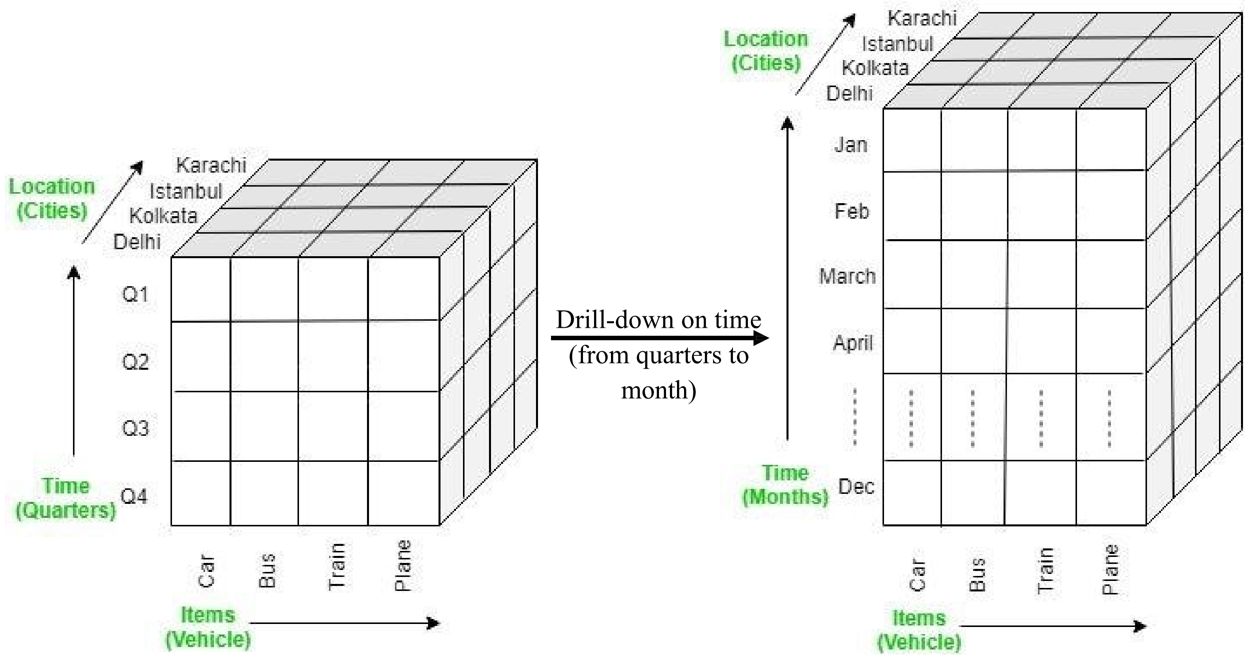
Example:



In this example, the roll-up operation is performed by climbing up in the concept hierarchy of *Location* dimension (City -> Country).

2. ***Drill-down (Roll-down):*** Drill-down is the reverse of roll-up. In drill-down operation, the less detailed data is converted into highly detailed data. It can be done by either stepping down a concept hierarchy for a dimension or introducing additional dimensions. When drill-down is performed, one or more dimensions from the data cube are added.

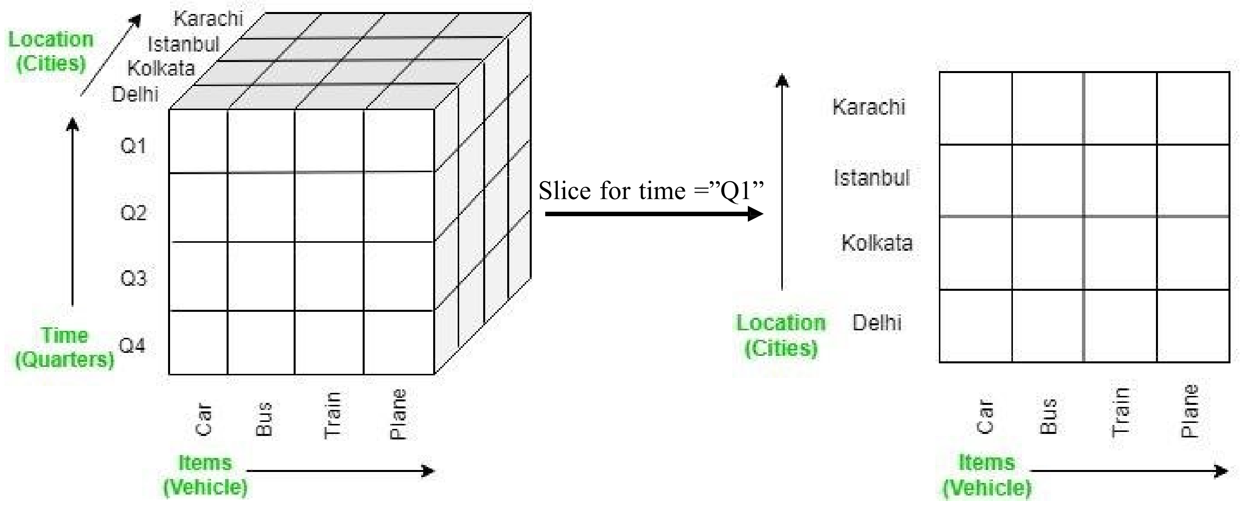
Example:



In this example, the drill down operation is performed by moving down in the concept hierarchy of *Time* dimension (Quarter -> Month).

3. ***Slice:*** The slice operation selects one particular dimension from a given cube and provides a new sub-cube. It reduces the dimensionality of the cubes.

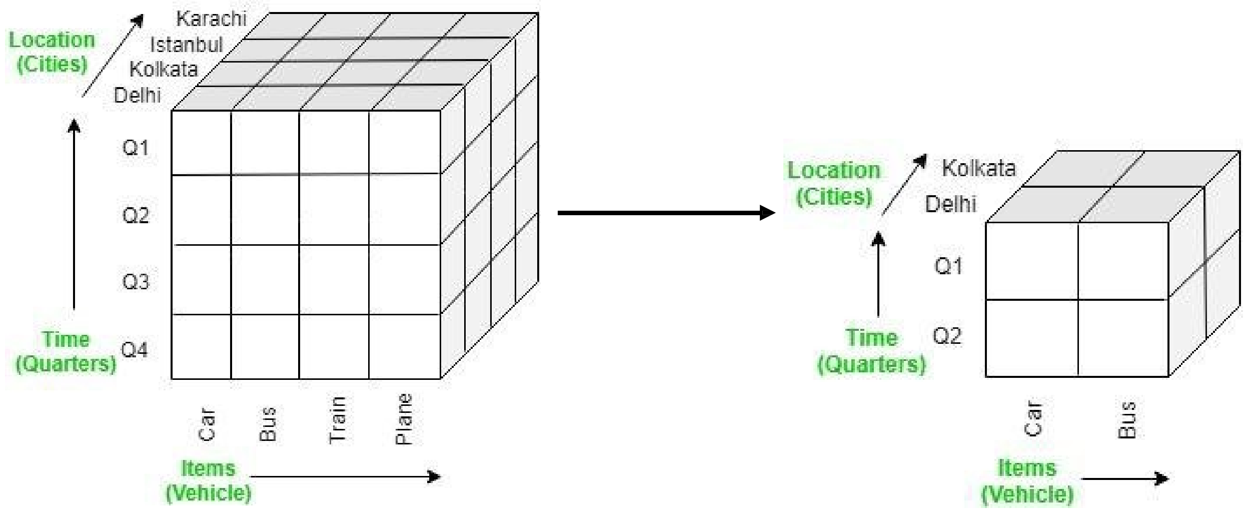
Example:



In this example, slice is performed for the dimension "time" using the criterion time = "Q1".

4. ***Dice:*** Dice selects two or more dimensions from a given cube and provides a new sub- cube.

Example:



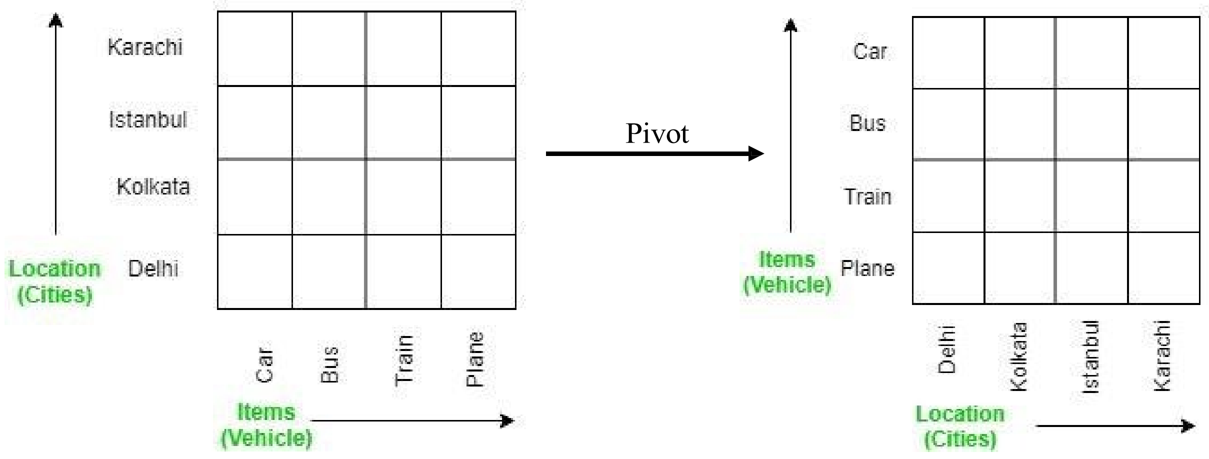
In this example, a sub-cube is selected by selecting following dimensions with criteria:

Location = “Delhi” or “Kolkata”
Time = “Q1” or “Q2”
Item = “Car” or “Bus”

5. ***Pivot:*** The pivot operation is also known as rotation. It rotates the data axis to view the data from different perspectives.

Example:

In the sub-cube obtained after the slice operation, performing pivot operation gives a new view of it.



<u>Types of OLAP Servers</u>

There are three main types of OLAP servers:

- Relational OLAP (ROLAP)
- Multidimensional OLAP (MOLAP)
- Hybrid OLAP (HOLAP)

Relational OLAP (ROLAP)

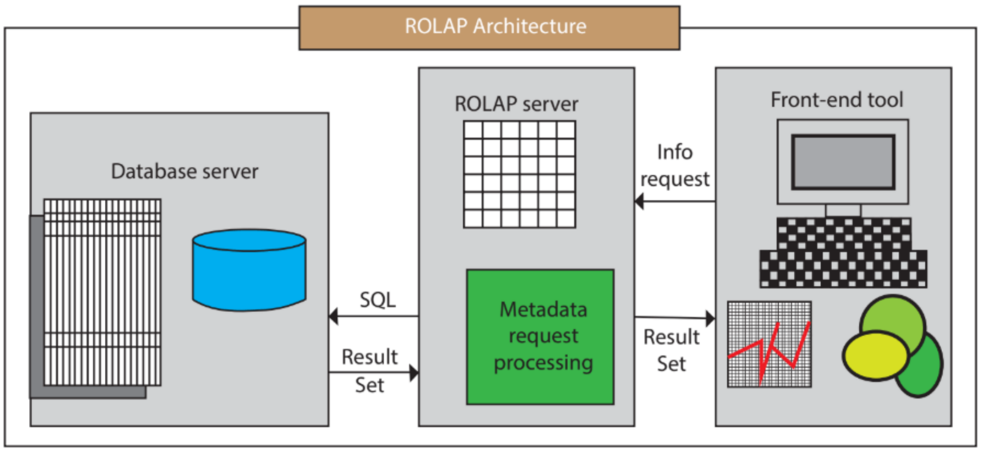
Relational OLAP servers are placed between relational back-end server and client front- end tools. To store and manage the warehouse data, the relational OLAP uses relational or extended-relational DBMS. ROLAP includes the following:

- Implementation of aggregation navigation logic
- Optimization for each DBMS back-end
- Additional tools and services

Examples: Microstrategy, Business Objects, Crystal Holos (ROLAP Mode), Essbase, Microsoft Analysis Services, Oracle Express (ROLAP Mode), Oracle Discoverer.

ROLAP includes the following components:

- Database server
- ROLAP server
- Front-end tool.



Advantages:

- ROLAP servers can be easily used with existing RDBMS.
- ROLAP tools do not use pre-calculated data cubes.
- ROLAP server offers highly scalability.
- Can handle large amounts of information.

Disadvantages:

- ROLAP needs high utilization of manpower, software, and hardware resources.
- Query performance in this model is slow.
- SQL functionality is constrained.

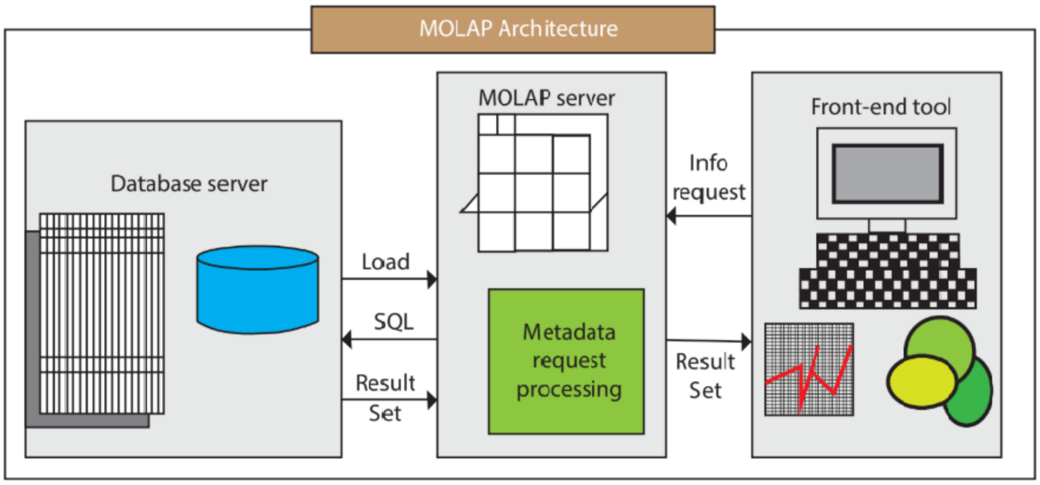
Multidimensional OLAP (MOLAP)

These servers support multidimensional views of data through array-based multidimensional storage engines. They map multidimensional views directly to data cube array structures. The advantage of using a data cube is that it allows fast indexing to pre computed summarized data. With multidimensional data stores, the storage utilization may be low if the data set is sparse. Therefore, many MOLAP servers use two levels of data storage representation to handle dense and sparse data-sets: denser sub- cubes are identified and stored as array structures, while sparse sub-cubes employ compression technology for efficient storage utilization.

Examples: Crystal Holos, Essbase, Microsoft Analysis Services, Oracle Express, Cognos Powerplay

MOLAP includes the following components:

- Database server.
- MOLAP server.
- Front-end tool.



Advantages:

- Fast information retrieval.
- Easier to use, therefore MOLAP is suitable for inexperienced users.
- Suitable for slicing and dicing operations.
- Capable of performing complex calculations.

Disadvantages:

- MOLAP are not capable of containing detailed data.
- The storage utilization may be low if the data set is sparse.
- It is difficult to change the dimensions without re-aggregating.

Hybrid OLAP (HOLAP)

Hybrid OLAP is a mixture of both ROLAP and MOLAP. It offers fast computation of MOLAP and higher scalability of ROLAP. HOLAP server allows to store large data volumes of detailed information. HOLAP uses two databases.

1. Aggregated or computed data is stored in a multidimensional OLAP cube
2. Detailed information is stored in a relational database.

Examples: Oracle Express, Seagate Holos, Speedware Media/M, Microsoft OLAP Services

Advantages:

- HOLAP provides the benefits of both MOLAP and ROLAP.
- It provide quick access at all levels of aggregation.

Disadvantages:

- HOLAP architecture is very complicated because it supports both MOLAP and ROLAP servers.
- There are higher chances of overlapping especially into their functionalities.

Operational Database System (OLTP System)	**Data Warehouse (OLAP System)**
Operational system are generally designed to support high-volume transaction processing.	Data warehousing systems are generally designed to support high volume analytical processing. (i.e. OLAP)
It is used for day-to-day operations.	It is used for long-term informational requirements and decision support.
Operational data are the original sources of the data.	Data comes from various OLTP Databases.
In operational system data is stored with a functional or process orientation.	In data warehousing systems data is stored with a subject orientation.
It provides detailed and flat relational view of data.	It provides summarized and multidimensional view of data.
It focuses on "Data In".	It focuses on Information out.
The tables and joins are complex since they are normalized (for RDMS). This is done to reduce redundant data and to save storage space.	The tables and joins are simple since they are de-normalized. This is done to reduce the response time for analytical queries.
Entity-Relationship modelling techniques are used for RDMS database design.	Data-Modeling techniques are used for the Data Warehouse design.
Performance is low for analysis queries.	High performance for analytical queries.
Data within operational systems are generally updated regularly.	Data within a data warehouse is non-volatile, meaning when new data is added old data is not erased so rarely updates.
Data volumes are less and historical data is generally not maintained.	It involves large data volumes and historical data.
Simple queries are capable of fetching the data.	Complex queries are required to fetch data.
Processing speed is fast.	Processing speed is slow because of large size.
The common users are clerk, DBA, database professional.	The common users are knowledge worker (e.g. manager, executive, analyst)

<u>Schemas for Multidimensional Data Model</u>

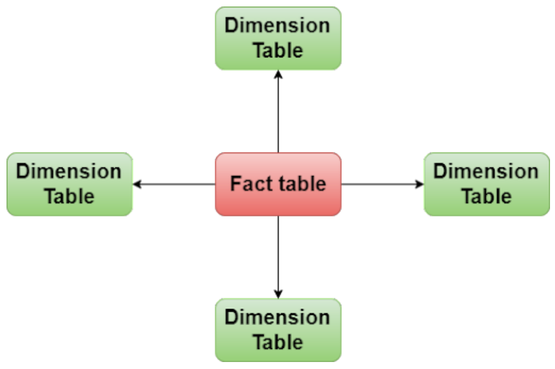
Schema is a logical description of the entire database. It includes the name and description of records of all record types including all associated data-items and aggregates. A data warehouse uses *Star, Snowflake, and Fact Constellation schema*.

Star Schema

The most common modelling paradigm is the star schema, in which the data warehouse contains:

a) large central table (fact table) containing the bulk of the data, with no redundancy, and

b) a set of smaller attendant tables (dimension tables), one for each dimension.



The fact table contains the detailed summary data. Its primary key has one key per dimension. Each tuple of the Fact table consists of a foreign key pointing to each of the dimension tables. It also stores numeric values.
The dimension table consists of columns that correspond to the attributes of the dimensions. The primary key of a dimension table is a foreign key in fact table.

Example:

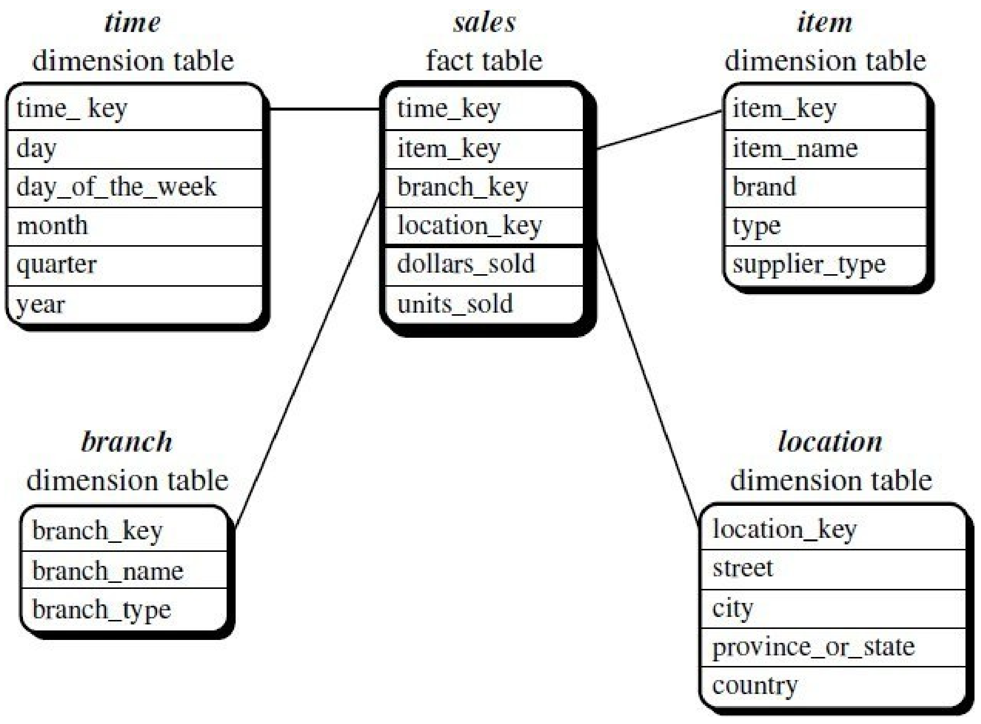


Fig: Star schema for data warehouse for sales.

Advantages:

- It is easy to understand and small number of tables can join.
- Since star schema contains de-normalized dimension tables, it leads to simpler queries due to lesser number of join operations and it also leads to better system performance.

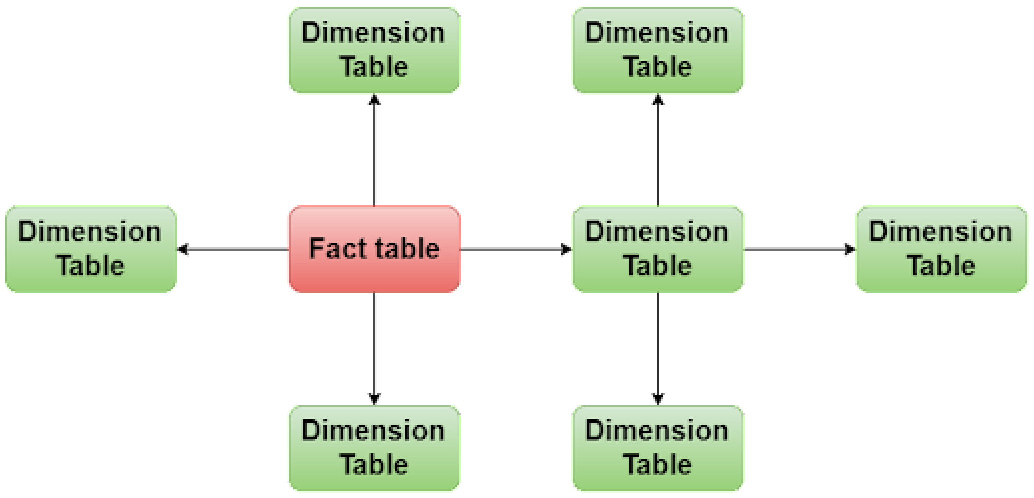
Disadvantages:

- It is difficult to maintain integrity of data in star schema due to de-normalized tables.
- Redundancy of the data hence occupies additional space.

Snowflake Shema

The snowflake schema is a variant of the star schema model, where some dimension tables are *normalized* which splits data into additional tables.

The snowflake schema is represented by centralized fact table which is connected to multiple dimension table and this dimension table can be normalized into additional dimension tables.



Snowflake Schemas for Multidimensional Modal

Snowflake Schema eliminates the redundancies and hence saves the storage space. It increases the number of dimension tables and requires more foreign key joins. The result is more complex queries and reduced query performance.

Example:

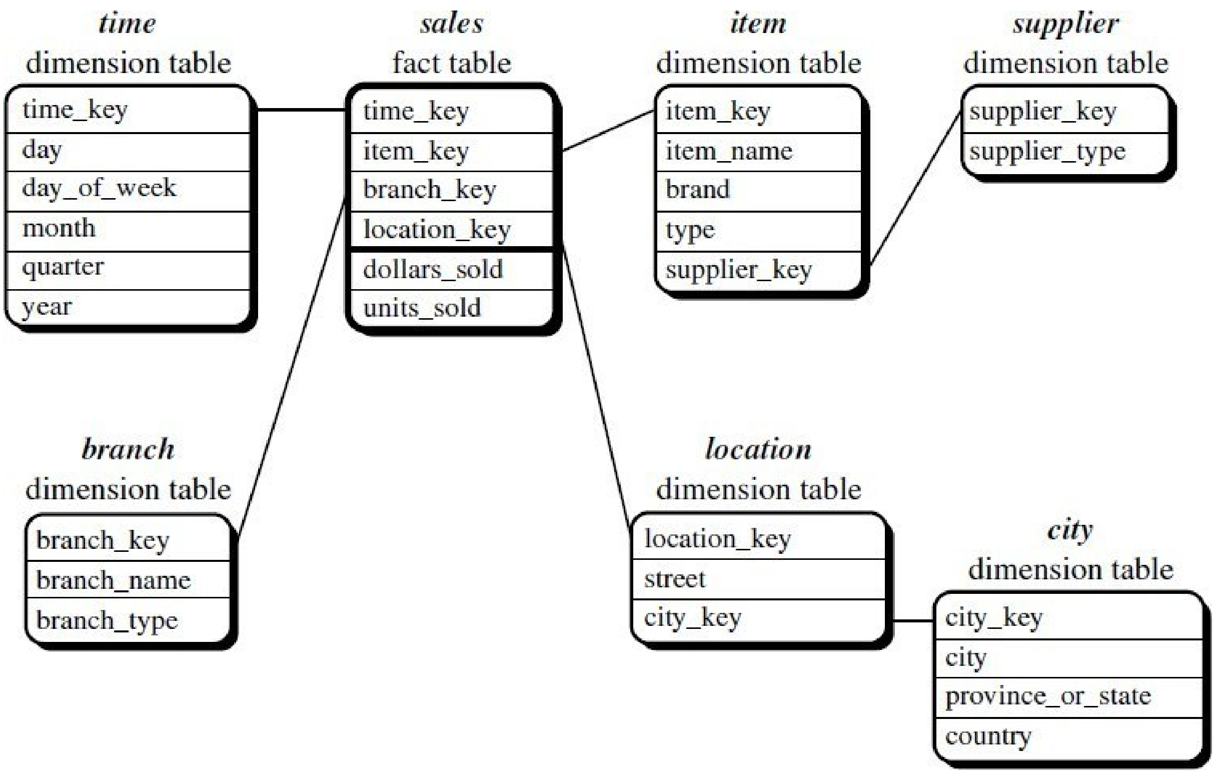
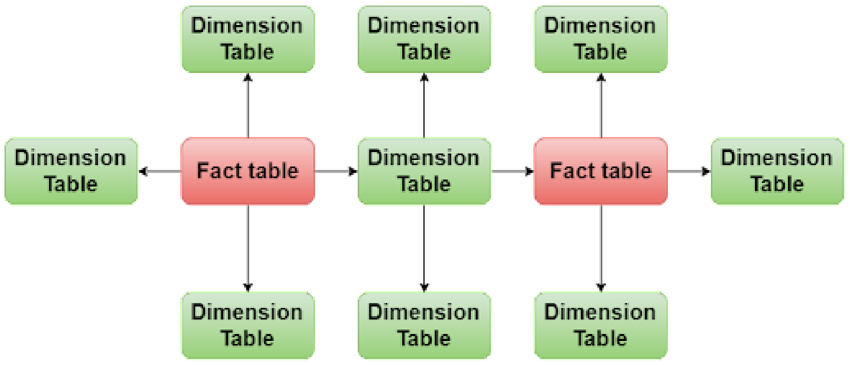


Fig: Snowflake schema of a data warehouse for sales.

Fact Constellation Schema

A Fact constellation schema is a type of schema which consists of more than one fact table that share many 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.



Fact constellation Schemas for Multidimensional Modal

The main disadvantage of fact constellation schemas is its more complicated design.

Example:

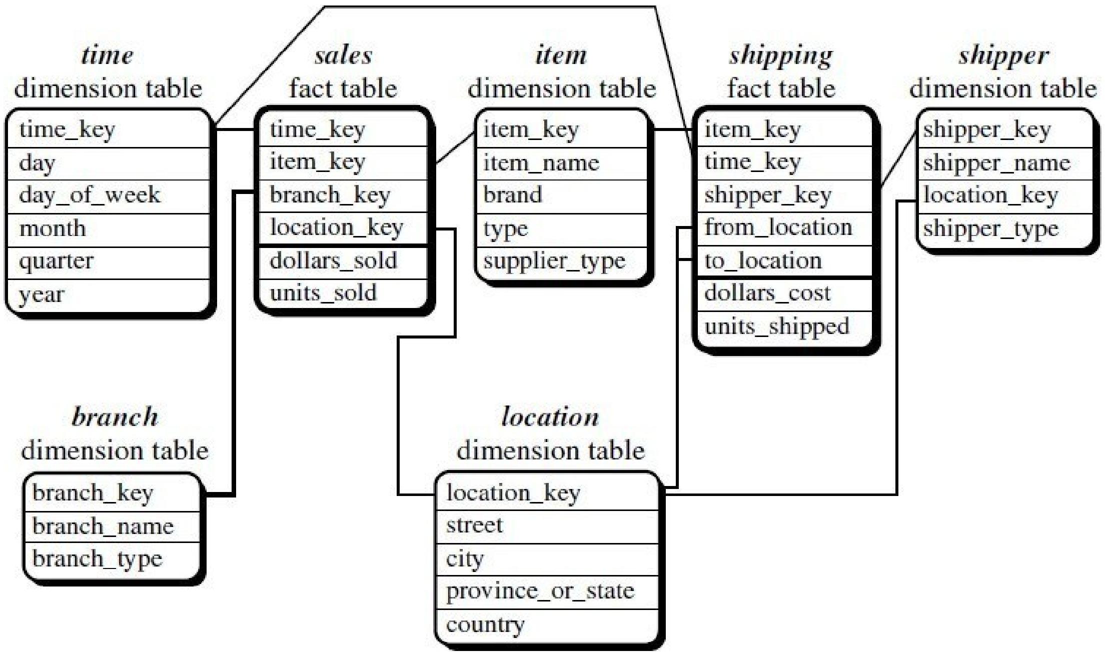
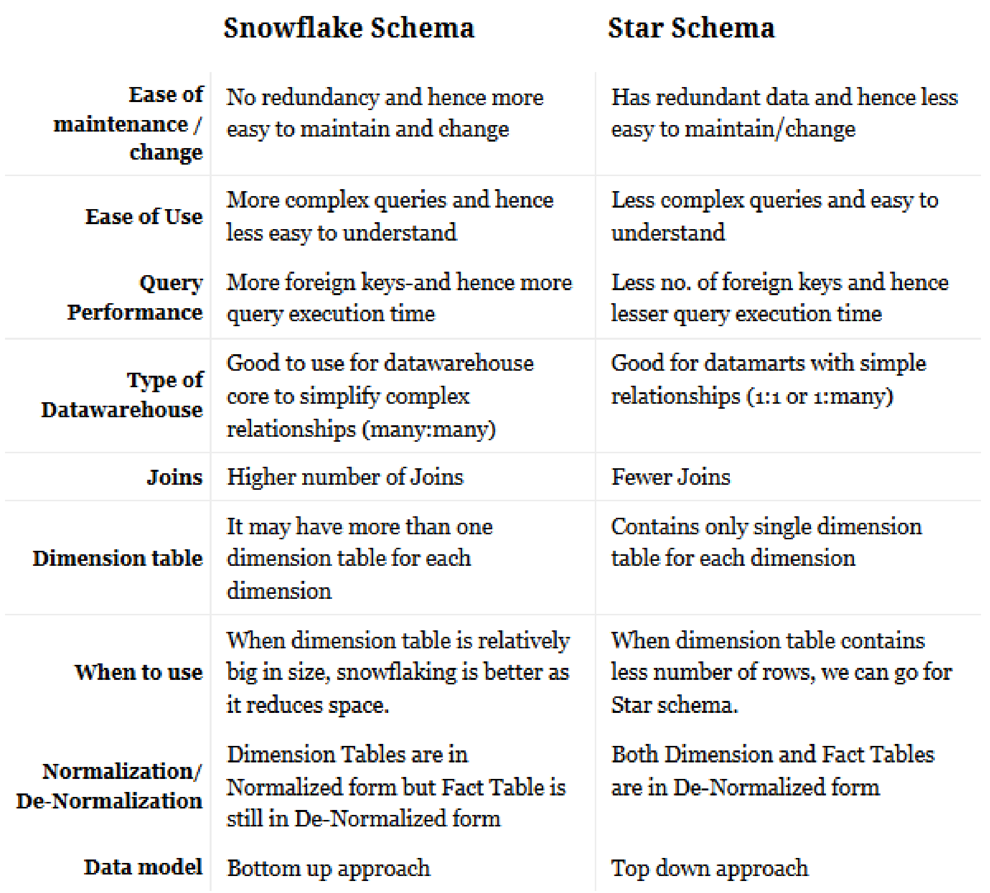


Fig: Fact constellation schema of a data warehouse for sales and shipping.

Snowflake Schema vs. Fact Constellation Schema

Snowflake Schema	Galaxy or Fact Constellation Schema
• Extension of star schema that adds additional dimensions	• Splits one star schema into more star schemas
• Called snowflake because its diagram resembles a snowflake	• Schema is viewed as a collection of star hence named as galaxy schema
• One fact table	• Two or more fact tables
• Data splits into different dimension tables	• The schema is helpful for aggregating fact tables for better understanding.
• One fact table surrounded by dimension table which are in turn surrounded by dimension table.	• Dimensions In this schema are separated into separate dimensions based on various levels of hierarchies.

Star Schema vs. Snowflake Schema



	Snowflake Schema	Star Schema
Ease of maintenance / change	No redundancy and hence more easy to maintain and change	Has redundant data and hence less easy to maintain/change
Ease of Use	More complex queries and hence less easy to understand	Less complex queries and easy to understand
Query Performance	More foreign keys-and hence more query execution time	Less no. of foreign keys and hence lesser query execution time
Type of Datawarehouse	Good to use for datawarehouse core to simplify complex relationships (many:many)	Good for datamarts with simple relationships (1:1 or 1:many)
Joins	Higher number of Joins	Fewer Joins
Dimension table	It may have more than one dimension table for each dimension	Contains only single dimension table for each dimension
When to use	When dimension table is relatively big in size, snowflaking is better as it reduces space.	When dimension table contains less number of rows, we can go for Star schema.
Normalization/ De-Normalization	Dimension Tables are in Normalized form but Fact Table is still in De-Normalized form	Both Dimension and Fact Tables are in De-Normalized form
Data model	Bottom up approach	Top down approach

Q. Let us consider the case of a real estate agency whose database is composed by the following tables:

OWNER (IDOwner, Name, Surname, Address, City, Phone)

ESTATE (IDEstate, IDOwner, Category, Area, City, Province, Rooms, Bedrooms, Garage, Meters)

CUSTOMER (IDCust, Name, Surname, Budget, Address, City, Phone)
AGENT (IDAgent, Name, Surname, Office, Address, City, Phone)
AGENDA (IDAgent, Data, Hour, IDEstate, ClientName)
VISIT (IDEstate, IDAgent, IDCust, Date, Duration)
SALE (IDEstate, IDAgent, IDCust, Date, AgreedPrice, Status)
RENT (IDEstate, IDAgent, IDCust, Date, Price, Status, Time)

Design a Star Schema or Snowflake Schema for the DW.

The following ideas will be used during the solution of the exercise:

- supervisors should be able to control the sales of the agency
  FACT Sales
  MEASURES OfferPrice, AgreedPrice, Status
  DIMENSIONS EstateID, OwnerID, CustomerID, AgentID, TimeID

- supervisors should be able to control the work of the agents by analyzing the visits to the estates, which the agents are in charge of
  FACT Viewing
  MEASURES Duration DIMENSIONS EstateID, CustomerID, AgentID, TimeID

Solution:

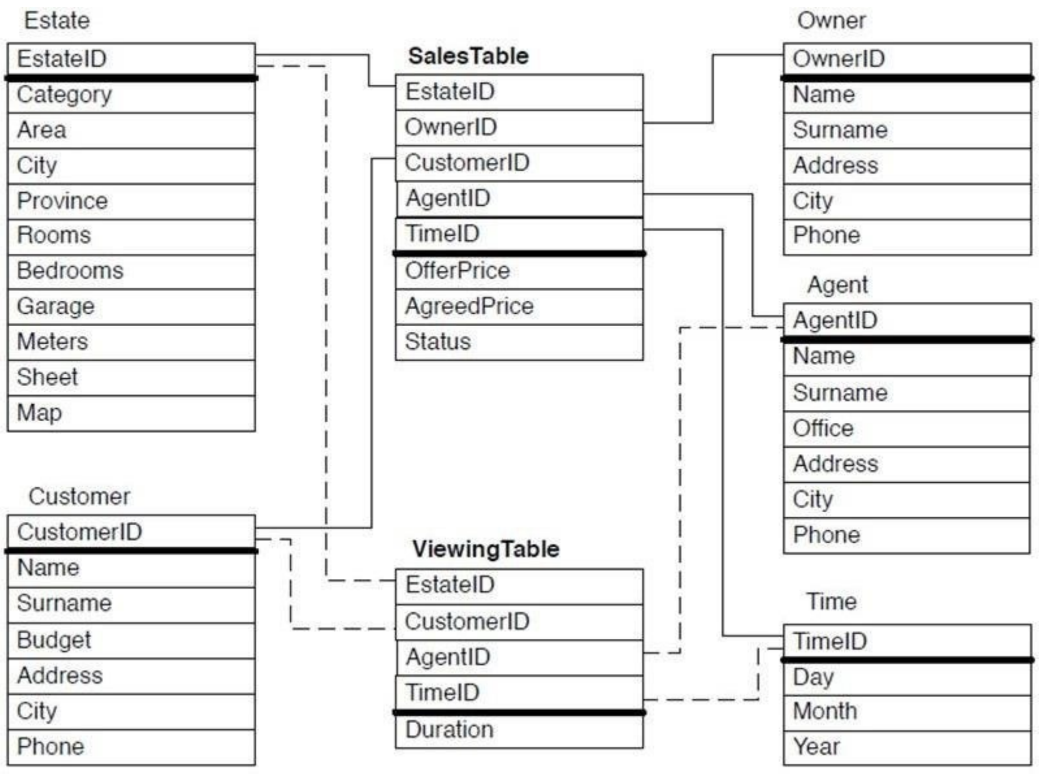


Figure: Star schema

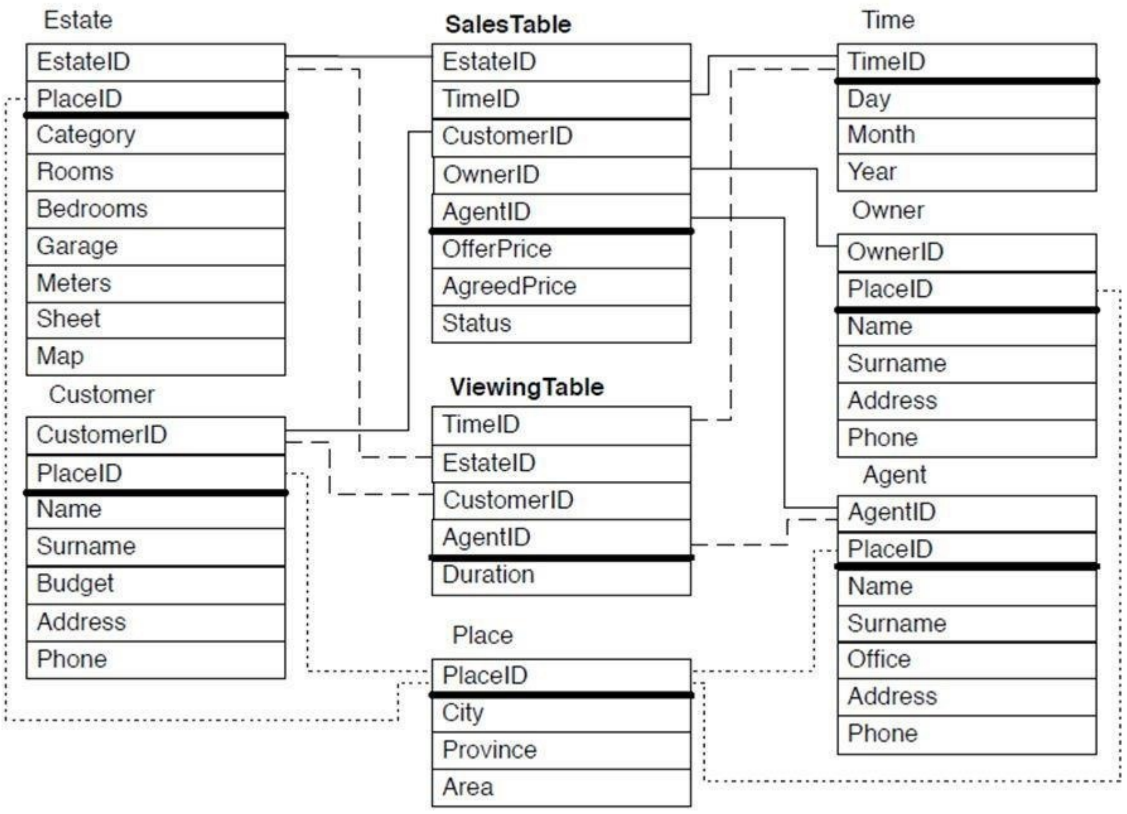


Figure: Snowflake schema

Q. Suppose that a data warehouse consists of the three dimensions time, doctor, and patient, and the two measures count and charge, where charge is the fee that a doctor charges a patient for a visit.

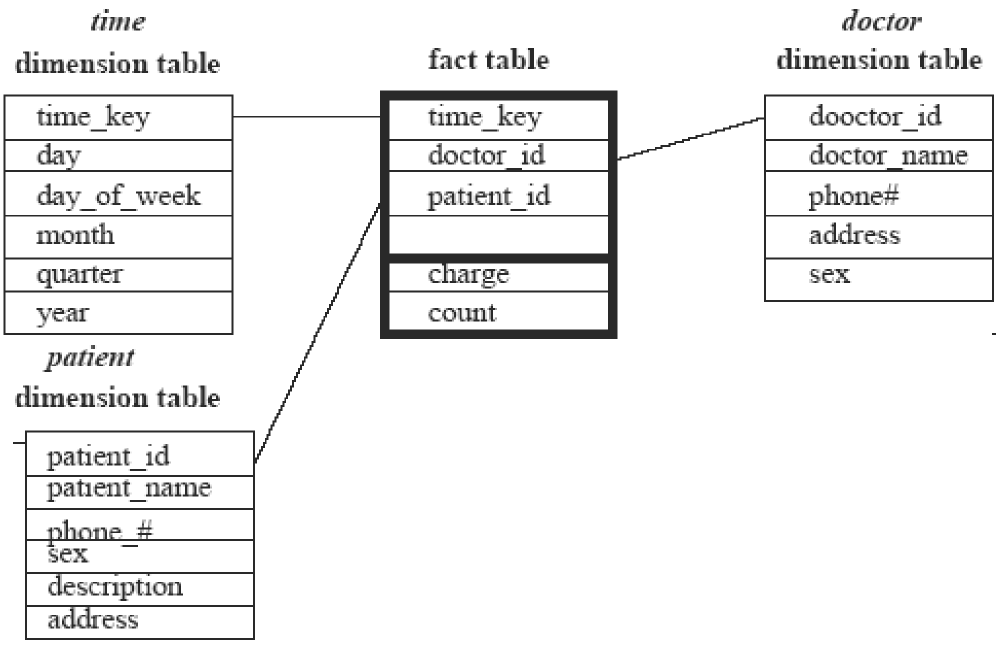
a) Draw a schema diagram for the above data warehouse using one of the schemas. [star, snowflake, fact constellation]

b) Starting with the base cuboid [day, doctor, patient], what specific OLAP operations should be performed in order to list the total fee collected by each doctor in 2004?

c) To obtain the same list, write an SQL query assuming the data are stored in a relational database with the schema fee (day, month, year, doctor, hospital, patient, count, charge)

Solution:

a) Star Schema is shown in figure below:



b) First, we should use roll-up operation to get the year 2004(rolling-up from day then month to year). After getting that, we need to use slice operation to select (2004). Second, we should use roll-up operation again to get all patients. Then, we need to use slice operation to select (all). Finally, we get list the total fee collected by each doctor in 2004. So,

1. roll up from day to month to year
2. slice for year = "2004"
3. roll up on patient from individual patient to all
4. slice for patient = "all"
5. get the list of total fee collected by each doctor in 2004

c)

```
Select doctor, Sum(charge)
From fee
Where year = 2004
Group by doctor
```

Q. Suppose that a data warehouse for Big-University consists of the following four dimensions: student, course, semester, and instructor, and two measures count and avg_grade. When at the lowest conceptual level (e.g., for a given student, course, semester, and instructor combination), the avg_grade measure stores the actual course grade of the student. At higher conceptual levels, avg_grade stores the average grade for the given combination.

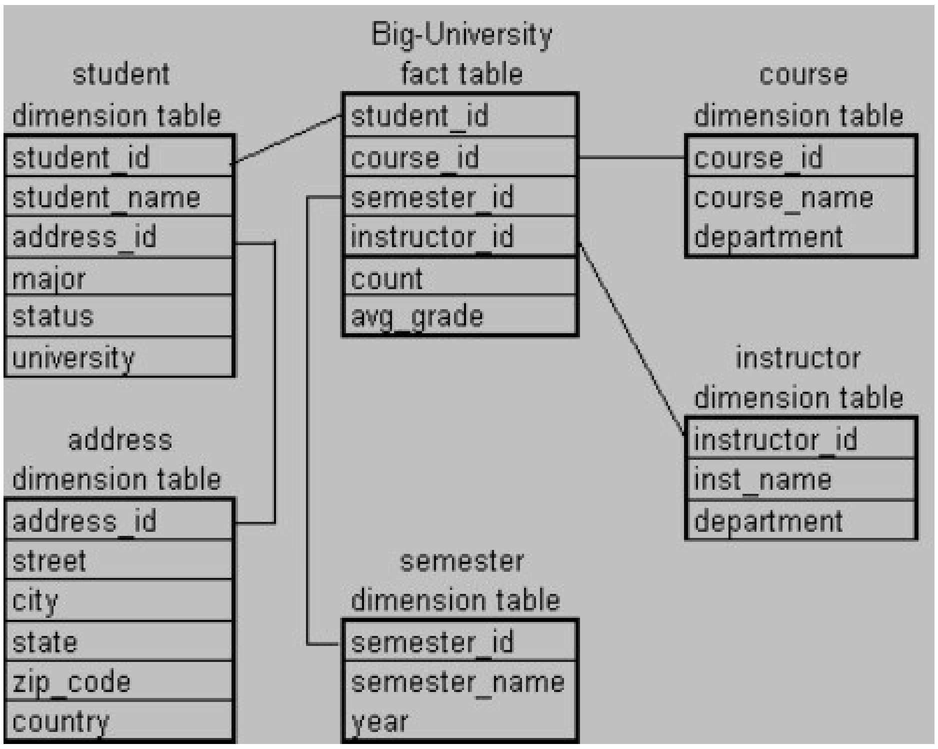
a) Draw a snowflake schema diagram for the data warehouse.

b) Starting with the base cuboid [student, course, semester, instructor], what specific OLAP operations (e.g., roll-up from semester to year) should one perform in order to list the average grade of CS courses for each BigUniversity student.

c) If each dimension has five levels (including all), such as "student < major < status < university < all", how many cuboids will this cube contain (including the base and apex cuboids)?

Solution:

a) A Snowflake Shema is shown in figure below:



b) The specific OLAP operations to be performed:

1. Roll-up on course from course_id to department.
2. Roll-up on student from student_id to university.
3. Dice on course, student with department ="CS" and university = "biguniversity"
4. Drill-down on student from university to student_name.

c) N = 4 dimensions

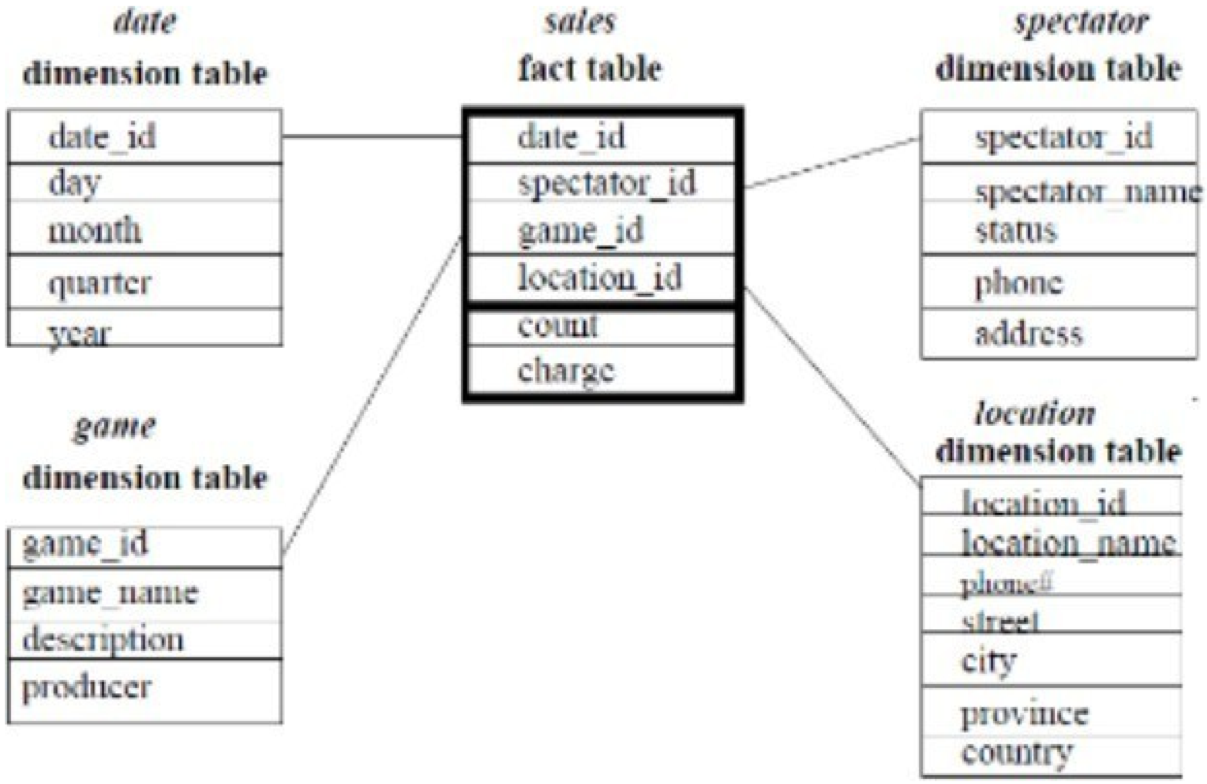
The cube will contain $5^4 = 625$ cuboids.

Q. Suppose that a data warehouse consists of the four dimensions; date, spectator, location, and game, and the two measures, count and charge, where charge is the fee that a spectator pays when watching a game on a given date.
Spectators may be students, adults, or seniors, with each category having its own charge rate.

a) Draw a star schema diagram for the data
b) Starting with the base cuboid [date; spectator; location; game], what specific OLAP operations should perform in order to list the total charge paid by student spectators at GM Place in 2004?

Solution:

a)



b) The specific OLAP operations to be performed are:

1. Roll-up on date from date id to year.
2. Roll-up on spectator from spectator id to status.
3. Roll-up on location from location id to location name.
4. Roll-up on game from game id to all.
5. Dice with status= “students”, location name= “GM Place”, and year=2004

1.10 Data Preprocessing:

1.10.1 Data Integration:

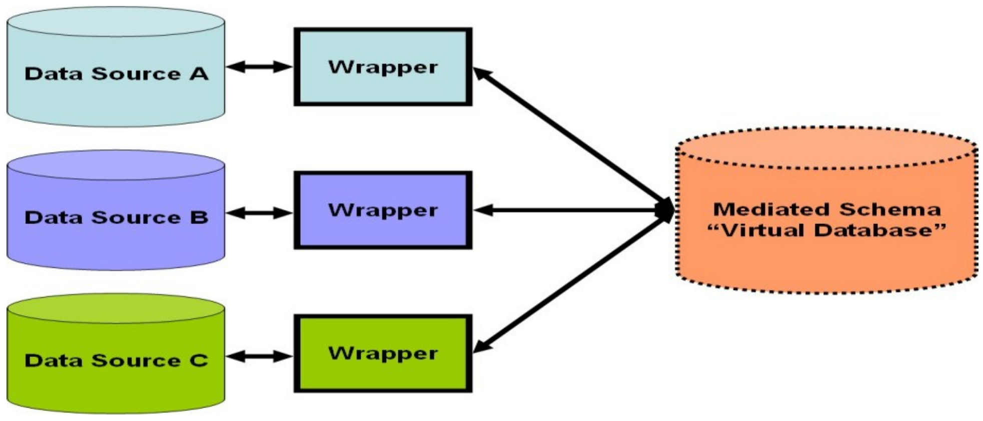
It combines data from multiple sources into a coherent data store, as in data warehousing. These sources may include multiple databases, data cubes, or flat files.

The data integration systems are formally defined as triple<G,S,M>

Where G: The global schema

S:Heterogeneous source of schemas

M: Mapping between the queries of source and global schema



1.10.2 Issues in Data integration:

1. **Schema integration and object matching:**
   How can the data analyst or the computer be sure that customer id in one database and customer number in another reference to the same attribute.
2. **Redundancy:**
   An attribute (such as annual revenue, forinstance) may be redundant if it can be derived from another attribute or set ofattributes. Inconsistencies in attribute or dimension naming can also cause redundanciesin the resulting data set.
3. **detection and resolution of datavalue conflicts:**
   For the same real-world entity, attribute values fromdifferent sources may differ.

1.10.3 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 includebinning, regression, and clustering.
- **Aggregation**, where summary or aggregation operations are applied to the data. For example, the daily sales data may be aggregated so as to compute monthly and annualtotal 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 byhigher-level concepts through the use of concept hierarchies. For example, categoricalattributes, like street, can be generalized to higher-level concepts, like city or country.
- **Normalization**, where the attribute data are scaled so as to fall within a small specifiedrange, such as 1:0 to 1:0, or 0:0 to 1:0.
- **Attribute construction** (or feature construction),wherenewattributes are constructedand added from the given set of attributes to help the mining process.

1.10.4 Data Reduction:

Data reduction techniques can be applied to obtain a reduced representation of thedata set that ismuch smaller in volume, yet closely maintains the integrity of the originaldata. 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 theconstruction 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, smallerdata representations such as parametric models (which need store only the modelparameters 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 attributesare replaced by ranges or higher conceptual levels. Data discretization is a form ofnumerosity reduction that is very useful for the automatic generation of concept hierarchies.Discretization and concept hierarchy generation are powerful tools for datamining, in that they allow the mining of data at multiple levels of abstraction.

1.10.5 Data Mining Tools

Different types of data mining tools are available in the marketplace, each with their own strengths and weaknesses. These tools use artificial intelligence, machine learning and other techniques to extract data. Most data mining tools can be classified into one of three categories: traditional data mining tools, dashboards, and text-mining tools.

- ***Traditional Data Mining Tools:*** Traditional data mining programs help companies establish data patterns and trends by using a number of complex algorithms and techniques. Some of these tools are installed on the desktop to monitor the data and highlight trends and others capture information residing outside a database. The majority are available in both Windows and UNIX versions, although some specialize in one operating system only. In addition, while some may concentrate on one database type, most will be able to handle any data using online analytical processing or a similar technology.
- ***Dashboards:*** Installed in computers to monitor information in a database, dashboards reflect data changes and updates onscreen — often in the form of a chart or table — enabling the user to see how the business is performing. Historical data also can be referenced, enabling the user to see where things have changed (e.g., increase in sales from the same period last year). This functionality makes dashboards easy to use and particularly appealing to managers who wish to have an overview of the company's performance.
- ***Text-mining Tools:*** The third type of data mining tool sometimes is called a text-

mining tool because of its ability to mine data from different kinds of text — from Microsoft Word and Acrobat PDF documents to simple text files, for example. These tools scan content and convert the selected data into a format that is compatible with the tool's database, thus providing users with an easy and convenient way of accessing data without the need to open different applications. Scanned content can be unstructured (i.e., information is scattered almost randomly across the document, including e-mails, Internet pages, audio and video data) or structured (i.e., the data's form and purpose is known, such as content found in a database). Capturing these inputs can provide organizations with a wealth of information that can be mined to discover trends, concepts, and attitudes.

Data Mining Task Primitives

We can specify a data mining task in the form of a data mining query. This query is input to the system.
A data mining query is defined in terms of data mining task primitives. These primitives allow us to communicate in an interactive manner with the data mining system.

The data mining task primitives are:

1. ***Task-relevant data:*** This specifies the portions of the database or the set of data in which the user is interested. This includes the database attributes or data warehouse dimensions of interest (referred to as the relevant attributes or dimensions). The data selected for mining is typically a subset of the overall data available, as not all data may be necessary or relevant for the task. For example: Extracting the database name, database tables, and relevant required attributes from the dataset from the provided input database.
2. ***The Kind of knowledge to be mined:*** *It refers to the type of information or insights that are being sought through the use of data mining techniques.* This specifies the data mining functions to be performed, such as characterization, discrimination, association or correlation analysis, classification, prediction, clustering, outlier analysis, or evolution analysis.
3. ***Background Knowledge:*** Background knowledge is information about the domain to be mined that can be useful in the discovery process and for evaluating the patterns found. The use of background knowledge can help to improve the accuracy and relevance of the insights obtained from the data mining process. Concept hierarchies are a popular form of background knowledge, which allow data to be mined at multiple levels of abstraction.
4. ***Interestingness measures:*** These functions are used to separate uninteresting patterns from knowledge. They may be used to guide the mining process, or after

discovery, to evaluate the discovered patterns. Different kinds of knowledge may have different interestingness measures. For example: Evaluating the interestingness and interestingness measures such as utility, certainty, and novelty for the data and setting an appropriate threshold value for the pattern evaluation.

5. ***Presentation and visualization of discovered patterns:*** It refers to the methods used to represent the patterns or insights discovered through data mining in a way that is easy to understand and interpret. Visualization techniques such as charts, graphs, and maps are commonly used to represent the data and can help to highlight important trends, patterns, or relationships within the data. Visualizing the discovered pattern helps to make the insights obtained from the data mining process more accessible and understandable to a wider audience, including non-technical stakeholders. For example Presentation and visualization of discovered pattern data using various visualization techniques such as barplot, charts, graphs, tables, etc.

Chapter-2

2.1 Association Rule Mining:

- Association rule mining is a popular and well researched method for discovering interesting relations between variables in large databases.
- It is intended to identify strong rules discovered in databases using different measures of interestingness.
- Based on the concept of strong rules, RakeshAgrawal et al. introduced association rules.

Problem Definition:

The problem of association rule mining is defined as:

Let $I = \{i_1, i_2, \ldots, i_n\}$be a set of n binary attributes called *items*.

Let $D = \{t_1, t_2, \ldots, t_m\}$be a set of transactions called the *database*.

Each transaction in Dhas a unique transaction ID and contains a subset of the items in I.

A *rule* is defined as an implication of the form $X \Rightarrow Y$

where $X, Y \subseteq I$and $X \cap Y = \emptyset$.

The sets of items (for short *itemsets*) $X$and $Y$are called *antecedent* (left-hand-side or LHS) and *consequent* (right-hand-side or RHS) of the rule respectively.

Example:

To illustrate the concepts, we use a small example from the supermarket domain. The set of items is $I = \{\mathrm{milk}, \mathrm{bread}, \mathrm{butter}, \mathrm{beer}\}$and a small database containing the items (1 codes presence and 0 absence of an item in a transaction) is shown in the table.

An example rule for the supermarket could be $\{\mathrm{butter}, \mathrm{bread}\} \Rightarrow \{\mathrm{milk}\}$meaning that if butter and bread are bought, customers also buy milk.

Example database with 4 items and 5 transactions

Transaction ID	milk	bread	butter	beer
1	1	1	0	0
2	0	0	1	0
3	0	0	0	1
4	1	1	1	0
5	0	1	0	0

2.1.1 Important concepts of Association Rule Mining:

- The **support** $\mathrm{supp}(X)$ of an itemset X is defined as the proportion of transactions in the data set which contain the itemset. In the example database, the itemset $\{\mathrm{milk}, \mathrm{bread}, \mathrm{butter}\}$ has a support of $1/5 = 0.2$ since it occurs in 20% of all transactions (1 out of 5 transactions).

- The **confidence** of a rule is defined

$$\mathrm{conf}(X \Rightarrow Y) = \mathrm{supp}(X \cup Y)/\mathrm{supp}(X).$$

For example, the rule $\{\mathrm{butter}, \mathrm{bread}\} \Rightarrow \{\mathrm{milk}\}$ has a confidence of $0.2/0.2 = 1.0$ in the database, which means that for 100% of the transactions containing butter and bread the rule is correct (100% of the times a customer buys butter and bread, milk is bought as well). Confidence can be interpreted as an estimate of the probability $P(Y|X)$, the probability of finding the RHS of the rule in transactions under the condition that these transactions also contain the LHS.

- The ***lift*** of a rule is defined as

$$\mathrm{lift}(X \Rightarrow Y) = \frac{\mathrm{supp}(X \cup Y)}{\mathrm{supp}(X) \times \mathrm{supp}(Y)}$$

or the ratio of the observed support to that expected if X and Y were independent. The rule $\{\mathrm{milk, bread}\} \Rightarrow \{\mathrm{butter}\}$ has a lift of $\frac{0.2}{0.4 \times 0.4} = 1.25$.

- The **conviction** of a rule is defined as

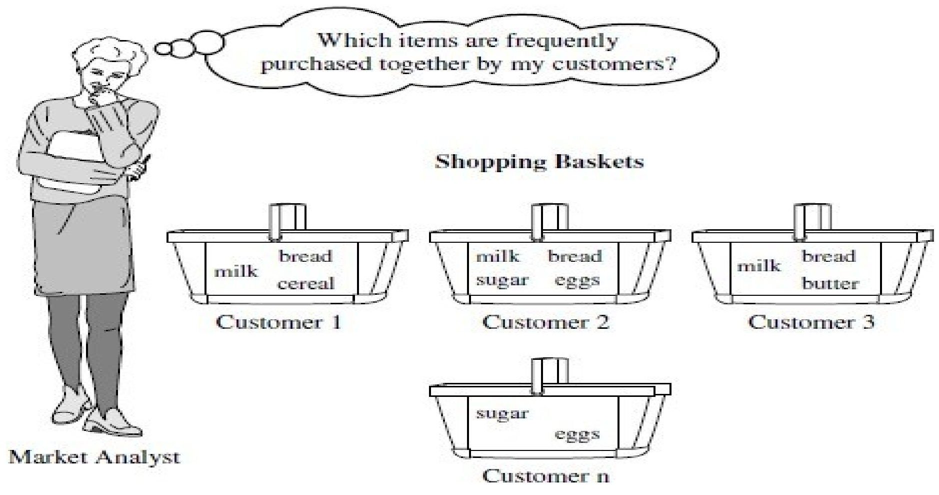
$$\mathrm{conv}(X \Rightarrow Y) = \frac{1 - \mathrm{supp}(Y)}{1 - \mathrm{conf}(X \Rightarrow Y)}.$$

The rule $\{\mathrm{milk, bread}\} \Rightarrow \{\mathrm{butter}\}$ has a conviction of $\frac{1 - 0.4}{1 - .5} = 1.2$,

and can be interpreted as the ratio of the expected frequency that X occurs without Y (that is to say, the frequency that the rule makes an incorrect prediction) if X and Y were independent divided by the observed frequency of incorrect predictions.

2.2 Market basket analysis:

This processanalyzes customer buying habits by finding associations between the different items thatcustomers place in their shopping baskets. The discovery of such associationscan help retailers develop marketing strategies by gaining insight into which itemsare frequently purchased together by customers. For instance, if customers are buyingmilk, how likely are they to also buy bread (and what kind of bread) on the same trip to the supermarket. Such information can lead to increased sales by helping retailers doselective marketing and plan their shelf space.



Example:

If customers who purchase computers also tend to buy antivirussoftware at the same time, then placing the hardware display close to the software displaymay help increase the sales of both items. In an alternative strategy, placing hardware andsoftware at opposite ends of the store may entice customers who purchase such items topick up other items along the way. For instance, after deciding on an expensive computer,a customer may observe security systems for sale while heading toward the software displayto purchase antivirus software and may decide to purchase a home security systemas well. Market basket analysis can also help retailers plan which items to put on saleat reduced prices. If customers tend to purchase computers and printers together, thenhaving a sale on printers may encourage the sale of printers *as well as* computers.

2.3 Frequent Pattern Mining:

Frequent patternmining can be classified in various ways, based on the following criteria:

1. **Based on the completeness of patterns to be mined:**
   - We can mine the complete set of frequent itemsets, the closed frequent itemsets, and the maximal frequent itemsets, given a minimum support threshold.
   - We can also mine constrained frequent itemsets, approximate frequent itemsets,near-match frequent itemsets, top-k frequent itemsets and so on.

2. **Based on the levels of abstraction involved in the rule set:**

   Some methods for associationrule mining can find rules at differing levels of abstraction.

   For example, supposethat a set of association rules mined includes the following rules where X is a variablerepresenting a customer:

   buys(X, ―computer‖))=>buys(X, ―HP printer‖) (1)

   buys(*X*, ―*laptop computer*‖)) =>*buys*(*X*, ―*HP printer*‖) (2)

   In rule (1) and (2), the items bought are referenced at different levels ofabstraction (e.g., ―*computer*‖ is a higher-level abstraction of ―*laptop computer*‖).

3. **Based on the number of data dimensions involved in the rule:**

- If the items or attributes in an association rule reference only one dimension, then it is a single-dimensional association rule.
  buys(X, ―computer‖))=>buys(X, ―antivirus software‖)

- If a rule references two or more dimensions, such as the dimensions age, income, and buys, then it is amultidimensional association rule. The following rule is an exampleof a multidimensional rule:
  age(X, ―30,31…39‖) ^ income(X, ―42K,…48K‖))=>buys(X, ―high resolution TV‖)

4. **Based on the types of values handled in the rule:**
   - If a rule involves associations between the presence or absence of items, it is a Boolean association rule.
   - If a rule describes associations between quantitative items or attributes, then it is a quantitative association rule.

5. **Based on the kinds of rules to be mined:**
   - Frequent pattern analysis can generate various kinds of rules and other interesting relationships.
   - Association rule mining cangenerate a large number of rules, many of which are redundant or do not indicatea correlation relationship among itemsets.
   - The discovered associations can be further analyzed to uncover statistical correlations,

leading to correlation rules.

6. **Based on the kinds of patterns to be mined:**

- Many kinds of frequent patterns can be mined from different kinds of data sets.
- Sequential pattern mining searches for frequent subsequences in a sequence data set, where a sequence records an ordering of events.
- For example, with sequential pattern mining, we can study the order in which items are frequently purchased. For instance, customers may tend to first buy a PC, followed by a digitalcamera,and then a memory card.
- Structuredpatternminingsearches for frequent substructuresin a structured data set.
- Single items are the simplest form of structure.
- Each element of an itemsetmay contain a subsequence, a subtree, and so on.
- Therefore, structuredpattern mining can be considered as the most general formof frequent pattern mining.

Association Rule Mining as two sub-processes

Association rule mining consists of two sub-processes: finding frequent item sets and generating association rules from those item sets.

- **Frequent itemset**: Frequent itemset is a set of items whose support is greater than the user specified minimum support.
- **Association rule**: An association rule is an implication or if-then-rule which is supported by data and can be represented in the form $X \Rightarrow Y$. An association rule must satisfy user-set minimum support (min_sup) and minimum confidence (min_conf). The rule $X \Rightarrow Y$ is called a strong association rule if support $\geq$ min_sup and confidence $\geq$ min_conf.

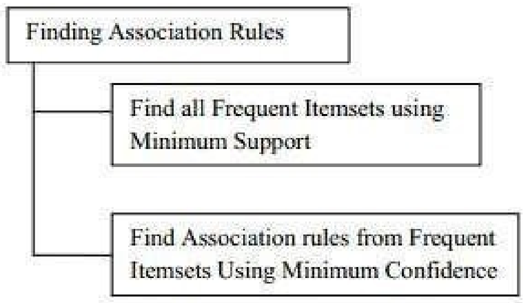

Finding Association Rules
Find all Frequent Itemsets using Minimum Support
Find Association rules from Frequent Itemsets Using Minimum Confidence

Figure 1: Generating Association Rules

Application of ARM (Why Association Rule Mining?)

- To find frequent patterns, associations, correlations, or causal structures among sets of items or objects in transaction databases, relational databases, and other information repositories.
- To discover relationships among various items in the database.
- To measure the interestingness of user based on support and confidence.
- Association rule mining can be used in application like: catalog design, market basket analysis, cross-marketing, clustering, classification etc.

Advantages and Disadvantages of Association Rules

Advantages:

- It is one of the important concept of data mining and machine learning.
- It is used in market basket analysis, apriori algorithms to find frequent itemstes.
- It is used for analyzing and predicting customer behavior.
- It is understandable and easy to use.
- It helps to uncover all relationship between items from huge database.
- It has flexible data formats.

Disadvantages:

- The number of rule obtained in often very large.
- Difficult to isolate interesting pattern.
- No widely accepted procedure.
- Needs deep knowledge for pattern understanding.

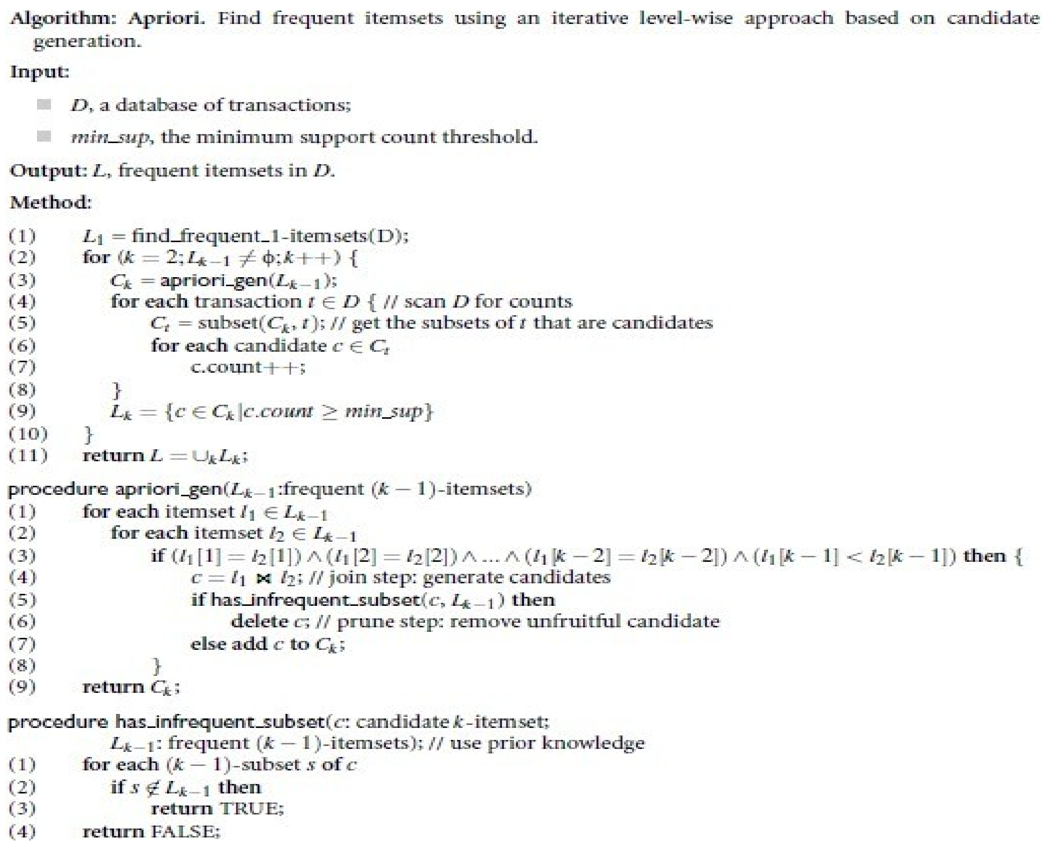
2.4 Efficient Frequent Itemset Mining Methods:

2.4.1 Finding Frequent Itemsets Using Candidate Generation:The Apriori Algorithm

- Apriori is a seminal algorithm proposed by R. Agrawal and R. Srikant in 1994 for mining frequent itemsets for Boolean association rules.
- The name of the algorithm is based on the fact that the algorithm uses *prior knowledge* of frequent itemset properties.
- Apriori employs an iterative approach known as a *level-wise* search, where k-itemsets are used to explore (k+1)-itemsets.
- First, the set of frequent 1-itemsets is found by scanning the database to accumulate the count for each item, and collecting those items that satisfy minimum support. The resulting set is denoted $L1$.Next, $L1$ is used to find $L2$, the set of frequent 2-itemsets, which is used

to find *L*3, and so on, until no more frequent *k*-itemsets can be found.

- The finding of each $L_k$requires one full scan of the database.
- A two-step process is followed in Aprioriconsisting of joinand prune action.



Algorithm: Apriori. Find frequent itemsets using an iterative level-wise approach based on candidate generation.

Input:

- *D*, a database of transactions;
- *min_sup*, the minimum support count threshold.

Output: *L*, frequent itemsets in *D*.

Method:

```
L_1 = find_frequent_1-itemsets(D);
for (k = 2; L_{k-1} ≠ φ; k++) {
    C_k = apriori_gen(L_{k-1});
    for each transaction t ∈ D { // scan D for counts
        C_t = subset(C_k, t); // get the subsets of t that are candidates
        for each candidate c ∈ C_t
            c.count++;
    }
    L_k = {c ∈ C_k | c.count ≥ min_sup}
}
return L = ∪_k L_k;

procedure apriori_gen(L_{k-1}:frequent (k − 1)-itemsets)
    for each itemset l_1 ∈ L_{k-1}
        for each itemset l_2 ∈ L_{k-1}
            if (l_1[1] = l_2[1]) ∧ (l_1[2] = l_2[2]) ∧ ... ∧ (l_1[k−2] = l_2[k−2]) ∧ (l_1[k−1] < l_2[k−1]) then {
                c = l_1 ⋈ l_2; // join step: generate candidates
                if has_infrequent_subset(c, L_{k-1}) then
                    delete c; // prune step: remove unfruitful candidate
                else add c to C_k;
            }
    return C_k;

procedure has_infrequent_subset(c: candidate k-itemset;
        L_{k-1}: frequent (k − 1)-itemsets); // use prior knowledge
    for each (k − 1)-subset s of c
        if s ∉ L_{k-1} then
            return TRUE;
    return FALSE;
```

Example:

TID	**List of item IDs**
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I1, I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

There are nine transactions in this database, that is, |D| = 9.

Steps:

1. In the first iteration of the algorithm, each item is a member of the set of candidate1- itemsets, C1. The algorithm simply scans all of the transactions in order to countthe number of occurrences of each item.

2. Suppose that the minimum support count required is 2, that is, min sup = 2. The set of frequent 1-itemsets, L1, can thenbe determined. It consists of the candidate 1-itemsets satisfying minimum support.In our example, all of the candidates in C1 satisfy minimum support.

3. To discover the set of frequent 2-itemsets, L2, the algorithm uses the join L1 on L1 togenerate a candidate set of 2-itemsets, C2.No candidates are removed fromC2 during the prune step because each subset of thecandidates is also frequent.

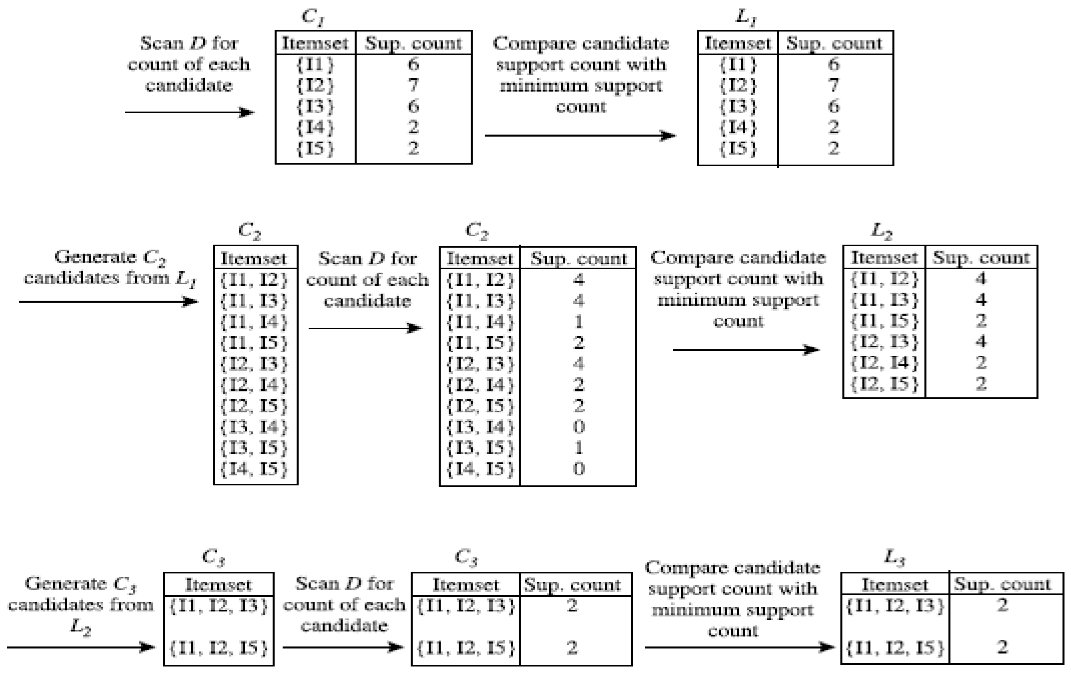
4. Next, the transactions inDare scanned and the support count of each candidate itemsetInC2 is accumulated.

5. The set of frequent 2-itemsets, L2, is then determined, consisting of those candidate2-itemsets in C2 having minimum support.

6. The generation of the set of candidate 3-itemsets,C3, Fromthejoin step, we first getC3 =L2x L2 = ({I1, I2, I3}, {I1, I2, I5}, {I1, I3, I5}, {I2, I3, I4},{I2, I3, I5}, {I2, I4, I5}. Based on the Apriori property that all subsets of a frequentitemsetmust also be frequent, we can determine that the four latter candidates cannotpossibly be frequent.

7. The transactions in D are scanned in order to determine L3, consisting of those candidate 3-itemsets in C3 having minimum support.

8. The algorithm uses L3x L3 to generate a candidate set of 4-itemsets, C4.


Scan D for count of each candidate
Compare candidate support count with minimum support count
Generate C_2 candidates from L_1
Scan D for count of each candidate
Compare candidate support count with minimum support count
Generate C_3 candidates from L_2
Scan D for count of each candidate
Compare candidate support count with minimum support count

C_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

L_1

Itemset	Sup. count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

C_2

Itemset
{I1, I2}
{I1, I3}
{I1, I4}
{I1, I5}
{I2, I3}
{I2, I4}
{I2, I5}
{I3, I4}
{I3, I5}
{I4, I5}

C_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I4}	1
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2
{I3, I4}	0
{I3, I5}	1
{I4, I5}	0

L_2

Itemset	Sup. count
{I1, I2}	4
{I1, I3}	4
{I1, I5}	2
{I2, I3}	4
{I2, I4}	2
{I2, I5}	2

C_3

Itemset
{I1, I2, I3}
{I1, I2, I5}

C_3

Itemset	Sup. count
{I1, I2, I3}	2
{I1, I2, I5}	2

L_3

Itemset	Sup. count
{I1, I2, I3}	2
{I1, I2, I5}	2

Generation of candidate itemsets and frequent itemsets, where the minimum support count is 2.

(a) Join: $C_3 = L_2 \bowtie L_2$ = {{I1, I2}, {I1, I3}, {I1, I5}, {I2, I3}, {I2, I4}, {I2, I5}} $\bowtie$
{{I1, I2}, {I1, I3}, {I1, I5}, {I2, I3}, {I2, I4}, {I2, I5}}
= {{I1, I2, I3}, {I1, I2, I5}, {I1, I3, I5}, {I2, I3, I4}, {I2, I3, I5}, {I2, I4, I5}}.

(b) Prune using the Apriori property: All nonempty subsets of a frequent itemset must also be frequent. Do any of the candidates have a subset that is not frequent?

- The 2-item subsets of {I1, I2, I3} are {I1, I2}, {I1, I3}, and {I2, I3}. All 2-item subsets of {I1, I2, I3} are members of L_2. Therefore, keep {I1, I2, I3} in C_3.
- The 2-item subsets of {I1, I2, I5} are {I1, I2}, {I1, I5}, and {I2, I5}. All 2-item subsets of {I1, I2, I5} are members of L_2. Therefore, keep {I1, I2, I5} in C_3.
- The 2-item subsets of {I1, I3, I5} are {I1, I3}, {I1, I5}, and {I3, I5}. {I3, I5} is not a member of L_2, and so it is not frequent. Therefore, remove {I1, I3, I5} from C_3.
- The 2-item subsets of {I2, I3, I4} are {I2, I3}, {I2, I4}, and {I3, I4}. {I3, I4} is not a member of L_2, and so it is not frequent. Therefore, remove {I2, I3, I4} from C_3.
- The 2-item subsets of {I2, I3, I5} are {I2, I3}, {I2, I5}, and {I3, I5}. {I3, I5} is not a member of L_2, and so it is not frequent. Therefore, remove {I2, I3, I5} from C_3.
- The 2-item subsets of {I2, I4, I5} are {I2, I4}, {I2, I5}, and {I4, I5}. {I4, I5} is not a member of L_2, and so it is not frequent. Therefore, remove {I2, I4, I5} from C_3.

(c) Therefore, C_3 = {{I1, I2, I3}, {I1, I2, I5}} after pruning.

Generation and pruning of candidate 3-itemsets, C_3, from L_2 using the Apriori property.

2.4.2 Generating Association Rules from Frequent Itemsets:

Once the frequent itemsets from transactions in a database *D* have been found, it is straightforward to generate strong association rules from them.

$$confidence(A \Rightarrow B) = P(B|A) = \frac{support\_count(A \cup B)}{support\_count(A)}.$$

The conditional probability is expressed in terms of itemset support count, where $support\_count(A \cup B)$ is the number of transactions containing the itemsets $A \cup B$, and $support\_count(A)$ is the number of transactions containing the itemset A. Based on this equation, association rules can be generated as follows:

- For each frequent itemset l, generate all nonempty subsets of l.
- For every nonempty subset s of l, output the rule "$s \Rightarrow (l - s)$" if $\frac{support\_count(l)}{support\_count(s)} \geq min\_conf$, where min_conf is the minimum confidence threshold.

Example:

Generating association rules. Let's try an example based on the transactional data for *AllElectronics* shown in Table 5.1. Suppose the data contain the frequent itemset l = {I1, I2, I5}. What are the association rules that can be generated from l? The nonempty subsets of l are {I1, I2}, {I1, I5}, {I2, I5}, {I1}, {I2}, and {I5}. The resulting association rules are as shown below, each listed with its confidence:

$$\begin{array}{ll}
I1 \wedge I2 \Rightarrow I5, & confidence = 2/4 = 50\% \\
I1 \wedge I5 \Rightarrow I2, & confidence = 2/2 = 100\% \\
I2 \wedge I5 \Rightarrow I1, & confidence = 2/2 = 100\% \\
I1 \Rightarrow I2 \wedge I5, & confidence = 2/6 = 33\% \\
I2 \Rightarrow I1 \wedge I5, & confidence = 2/7 = 29\% \\
I5 \Rightarrow I1 \wedge I2, & confidence = 2/2 = 100\%
\end{array}$$

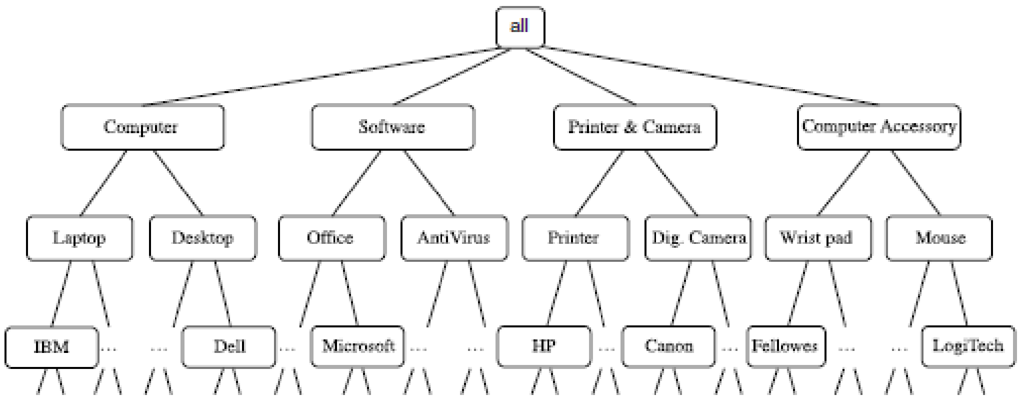
2.5 Mining Multilevel Association Rules:

- For many applications, it is difficult to find strong associations among data items at low or primitive levels of abstraction due to the sparsity of data at those levels.
- Strong associations discovered at high levels of abstraction may represent commonsense knowledge.
- Therefore, data mining systems should provide capabilities for mining association rules at multiple levels of abstraction, with sufficient flexibility for easy traversal amongdifferentabstraction spaces.

- Association rules generated from mining data at multiple levels of abstraction arecalled multiple-level or multilevel association rules.
- Multilevel association rules can be mined efficiently using concept hierarchies under a support-confidence framework.
- In general, a top-down strategy is employed, where counts are accumulated for the calculation of frequent itemsets at each concept level, starting at the concept level 1 and working downward in the hierarchy toward the more specific concept levels,until no more frequent itemsets can be found.

A concepthierarchy defines a sequence of mappings froma set of low-level concepts to higherlevel,more general concepts. Data can be generalized by replacing low-level conceptswithin the data by their higher-level concepts, or ancestors, from a concept hierarchy.

TID	*Items Purchased*
T100	IBM-ThinkPad-T40/2373, HP-Photosmart-7660
T200	Microsoft-Office-Professional-2003, Microsoft-Plus!-Digital-Media
T300	Logitech-MX700-Cordless-Mouse, Fellowes-Wrist-Rest
T400	Dell-Dimension-XPS, Canon-PowerShot-S400
T500	IBM-ThinkPad-R40/P4M, Symantec-Norton-Antivirus-2003
...	...



A concept hierarchy for *AllElectronics* computer items.

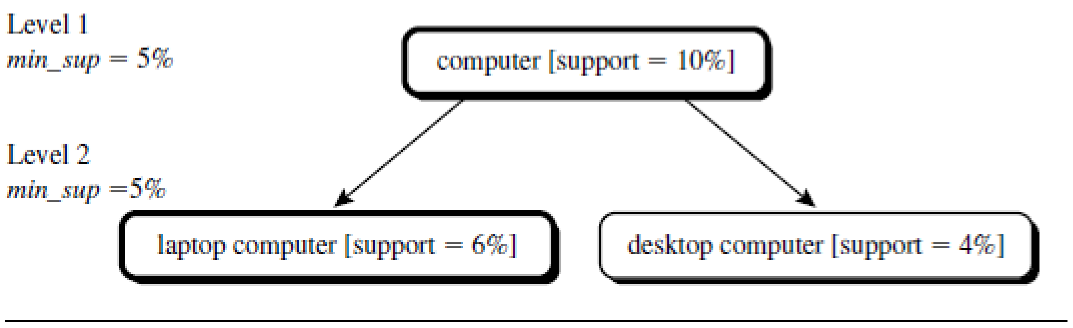
The concept hierarchy has five levels, respectively referred to as levels 0to 4, starting with level 0 at the root node for all.

- Here, Level 1 includes computer, software, printer&camera, and computer accessory.
- Level 2 includes laptop computer, desktop computer, office software, antivirus software
- Level 3 includes IBM desktop computer, . . . , Microsoft office software, and so on.
- Level 4 is the most specific abstraction level of this hierarchy.

2.5.1 Approaches ForMining Multilevel Association Rules:

1.UniformMinimum Support:

- The same minimum support threshold is used when mining at each level of abstraction.
- When a uniform minimum support threshold is used, the search procedure is simplified.
- The method is also simple in that users are required to specify only one minimum support threshold.
- The uniform support approach, however, has some difficulties. It is unlikely thatitems at lower levels of abstraction will occur as frequently as those at higher levelsof abstraction.
- If the minimum support threshold is set too high, it could miss somemeaningful associations occurring at low abstraction levels. If the threshold is set too low, it may generate many uninteresting associations occurring at high abstractionlevels.

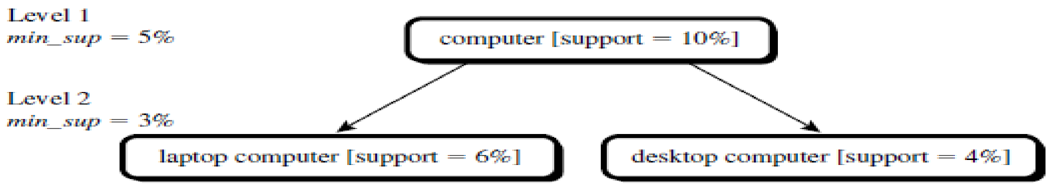


Multilevel mining with uniform support.

2.Reduced Minimum Support:

- Each level of abstraction has its own minimum support threshold.

- The deeper the level of abstraction, the smaller the corresponding threshold is.
- For example,the minimum support thresholds for levels 1 and 2 are 5% and 3%,respectively. In this way, ―computer,‖ ―laptop computer,‖ and ―desktop computer‖ areall considered frequent.



3. Group-Based Minimum Support:

- Because users or experts often have insight as to which groups are more important than others, it is sometimes more desirable to set up user-specific, item, or group based minimal support thresholds when mining multilevel rules.
- For example, a user could set up the minimum support thresholds based on product price, or on items of interest, such as by setting particularly low support thresholds for laptop computersand flash drives in order to pay particular attention to the association patterns containing items in these categories.

2.6 Mining Multidimensional Association Rules from Relational Databases and Data Warehouses:

- ***Single Dimensional Rule:*** It contains a single predicate (*e.g. purchase*) with its multiple occurrences (i.e., the predicate occurs more than once within the rule). For example:
  $purchase(X, \text{"milk"}) \Rightarrow purchase(X, \text{"bread"})$

- ***Multi-dimensional Rule:*** It contains two or more predicate. Each predicate occurs only once. For example:
  $age(X, \text{"}19-25\text{"}) \wedge occupation(X, \text{"}student\text{"}) \Rightarrow buys(X, \text{"}laptop\text{"})$

- ***Hybrid dimensional Association Rule:*** It is a multidimensional association rule with repeated predicates, which contain multiple occurrences of some predicates. For example:
  $age(X, \text{"}19-25\text{"}) \wedge buys(X, \text{"}laptop\text{"}) \Rightarrow buys(X, \text{"}printer\text{"})$

2.7 Mining Quantitative Association Rules:

- Quantitative association rules are multidimensional association rules in which the numeric attributes are *dynamically* discretized during the mining process so as to satisfy some mining criteria, such as maximizing the confidence or compactness of the rules mined.
- In this section, we focus specifically on how to mine quantitative association rules having two quantitative attributes on the left-hand side of the rule and one categorical attribute on the right-hand side of the rule. That is

*Aquan*1 ^*Aquan*2 =>*Acat*

where*Aquan*1 and *Aquan*2 are tests on quantitative attribute interval

*Acat*tests a categorical attribute fromthe task-relevantdata.

- Such rules have been referred to as two-dimensional quantitative association rules, because they contain two quantitative dimensions.
- For instance, suppose you are curious about the association relationship between pairs of quantitative attributes, like customer age and income, and the type of television (such as *high-definition TV,* i.e., *HDTV*) that customers like to buy.

An example of such a 2-D quantitative association rule is

age(*X*, ―30…39‖)^*income*(*X*, ―42*K*…48*K*‖)=>*buys*(*X*, ―*HDTV*‖)

2.8 From Association Mining to Correlation Analysis:

- A correlation measure can be used to augment the support-confidence framework for association rules. This leads to *correlation rules* of the form *A*=>*B* [*support*, *confidence*, *correlation*]
- That is, a correlation rule is measured not only by its support and confidence but alsoby the correlation between itemsets*A* and *B*. There are many different correlation measuresfrom which to choose. In this section, we study various correlation measures todetermine which would be good for mining large data

sets.

- Lift is a simple correlation measure that is given as follows. The occurrence of itemset A is independent of the occurrence of itemset$B$ if $P(A \cup B)$= $P(A)P(B)$; otherwise, itemsets$A$ and B are dependent and correlated as events. This definition can easily be extended to more than two itemsets.

The lift between the occurrence of A and B can bemeasured by computing

$$lift(A, B) = \frac{P(A \cup B)}{P(A)P(B)}.$$

- If the lift(A,B) is less than 1, then the occurrence of A is negatively correlated with the occurrence of B.
- If the resulting value is greater than 1, then A and B are positively correlated, meaning that the occurrence of one implies the occurrence of the other.
- If the resulting value is equal to 1, then A and B are independent and there is no correlation between them.

Apriori Algorithm

Apriori algorithm is a sequence of steps to be followed to find the most frequent itemset in the given database.

Apriori Property: All subsets of a frequent itemset must be frequent.

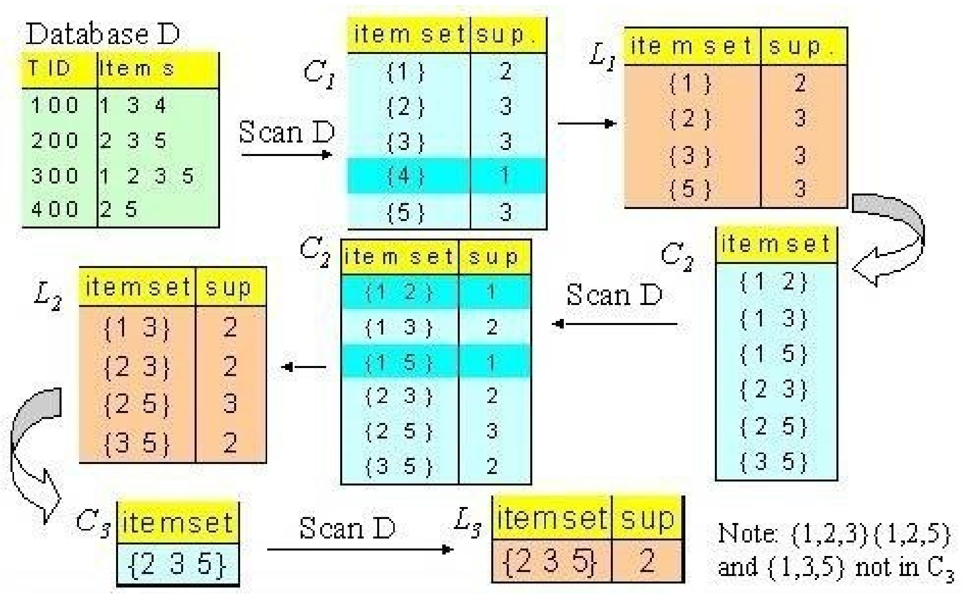
Apriori pruning principle: If there is any itemset which is infrequent, its superset should not be generated/tested.

Pseudo-code:

```
$C_k$: Candidate itemset of size $k$
$L_k$: frequent itemset of size $k$

$L_1$ = {frequent items};
for ($k = 1$; $L_k != \emptyset$; $k++$) do begin
    $C_{k+1}$ = candidates generated from $L_k$;
    for each transaction $t$ in database do
        increment the count of all candidates in $C_{k+1}$ that are contained in $t$
    $L_{k+1}$= candidates in $C_{k+1}$ with min_support
    End
return $\cup_k L_k$;
```

Example:



Database D

TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

C_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{4}	1
{5}	3

L_1

itemset	sup.
{1}	2
{2}	3
{3}	3
{5}	3

C_2

itemset
{1 2}
{1 3}
{1 5}
{2 3}
{2 5}
{3 5}

C_2

itemset	sup
{1 2}	1
{1 3}	2
{1 5}	1
{2 3}	2
{2 5}	3
{3 5}	2

L_2

itemset	sup
{1 3}	2
{2 3}	2
{2 5}	3
{3 5}	2

C_3

itemset
{2 3 5}

L_3

itemset	sup
{2 3 5}	2

Generating Association Rules from Frequent Itemsets:

```
For each frequent itemset, f, generate all non-empty subsets of f.
For every non-empty subset s of f do
    output rules $s \Rightarrow (f - s)$ if $\frac{support(f)}{support(s)} \geq \min\_confidence$
end
```

Here,

Frequent itemset

= {2, 3, 5} Now,

Rules	Confidence
$I_2 \Rightarrow I_3 \wedge I_5$	2/3 = 66%
$I_3 \Rightarrow I_2 \wedge I_5$	2/3 = 66%
$I_5 \Rightarrow I_2 \wedge I_3$	2/3 = 66%
$I_2 \wedge I_3 \Rightarrow I_5$	2/2 = 100%
$I_2 \wedge I_5 \Rightarrow I_3$	2/3 = 66%
$I_3 \wedge I_5 \Rightarrow I_2$	2/2 = 100%

As the given threshold or minimum confidence is 75%, so the rules $I_2 \wedge I_3 \Rightarrow I_5$ and $I_3 \wedge I_5 \Rightarrow I_2$ can be considered as the strong association rules for the given problem.

Q. ***You are given the transaction data shown below from a fast food restaurant. There are 9 distinct transactions (order 1 to order 9). There are total 5 meal (M_1 to***

M_5) involved in transactions.

Meal Items	***List of IDs***	***Meal Items***	***List of IDs***
Order 1	M_1, M_2, M_5	***Order 6***	M_2, M_3
Order 2	M_2, M_4	***Order 7***	M_1, M_3
Order 3	M_2, M_3	***Order 8***	M_1, M_2, M_3, M_5
Order 4	M_1, M_2, M_4	***Order 9***	M_1, M_2, M_3
Order 5	M_1, M_3		

Minimum support = 2, Minimum confidence = 0.7
Apply the Apriori algorithm to the database to identify frequent k-itemset and find all strong association rules.

Solution:

Step 1: Calculating C_1 and L_1

Candidate 1-itemsets (C_1)

Itemset	Sup_count
{M_1}	6
{M_2}	7
{M_3}	6
{M_4}	2
{M_5}	2

Frequent 1-itemsets (L_1)

Items	Sup_count
{M_1}	6
{M_2}	7
{M_3}	6
{M_4}	2
{M_5}	2

Step 2: Calculating C_2 and L_2

Candidate 2-itemsets (C_2)

Itemset	Sup_count
{M_1, M_2 }	4
{M_1, M_3 }	4
{M_1, M_4 }	1
{M_1, M_5 }	2
{M_2, M_3}	4
{M_2, M_4}	2
{M_2, M_5}	2
{M_3, M_4}	0
{M_3, M_5}	1
{M_4, M_5}	0

Frequent 2-itemsets (L_2)

Items	Sup_count
{M_1, M_2 }	4
{M_1, M_3 }	4
{M_1, M_5 }	2
{M_2, M_3}	4
{M_2, M_4}	2
{M_2, M_5}	2

Step 3: Calculating C_3 and L_3

Candidate 3-itemsets (C_3)

Itemset	Sup_count

Frequent 3-itemsets (L_3)

Items	Sup_count

$\{M_1, M_2, M_3\}$	2
$\{M_1, M_2, M_5\}$	2

$\{M_1, M_2, M_3\}$	2
$\{M_1, M_2, M_5\}$	2

Step 4: Calculating C_4 and L_4

A candidate set of 4-itemsets, C_4 is $\{M_1, M_2, M_3, M_5\}$. This item set is pruned since its subset $\{M_2, M_3, M_5\}$ is not frequent. Thus $C_4 = \phi$, and algorithm terminates .

∴ Frequent Itemsets = $\{M_1, M_2, M_3\}$ and $\{M_1, M_2, M_5\}$

Now,

Generating Association Rules from Frequent Itemsets:

Taking $\{M_1, M_2, M_3\}$

Rules	**Confidence**
$M_1 \Rightarrow M_2 \wedge M_3$	2/6 = 0.333
$M_2 \Rightarrow M_1 \wedge M_3$	2/7 = 0.285
$M_3 \Rightarrow M_1 \wedge M_2$	2/6 = 0.333
$M_2 \wedge M_3 \Rightarrow M_1$	2/4 = 0.5
$M_1 \wedge M_3 \Rightarrow M_2$	2/4 = 0.5
$M_1 \wedge M_2 \Rightarrow M_3$	2/4 = 0.5

Taking $\{M_1, M_2, M_5\}$

Rules	**Confidence**
$M_1 \Rightarrow M_2 \wedge M_5$	2/6 = 0.333
$M_2 \Rightarrow M_1 \wedge M_5$	2/7 = 0.285
$M_5 \Rightarrow M_1 \wedge M_2$	2/2 = 1
$M_2 \wedge M_5 \Rightarrow M_1$	2/2 = 1
$M_1 \wedge M_5 \Rightarrow M_2$	2/2 = 1
$M_1 \wedge M_2 \Rightarrow M_5$	2/4 = 0.5

As the given threshold or minimum confidence is 0.7, so the rules $M_5 \Rightarrow M_1 \wedge M_2$, $M_2 \wedge M_5 \Rightarrow M_1$ and $M_1 \wedge M_5 \Rightarrow M_2$ can be considered as the strong association rules for the given problem.

Advantages of Apriori Algorithm

- This is easy to understand algorithm.
- This algorithm has least memory consumption.
- Easy implementation.
- It uses Apriori property for pruning therefore, itemsets left for further support checking remain less.

Disadvantages of Apriori Algorithm

- It requires many scans of database.
- It allows only a single minimum support threshold.
- It is favourable only for small database.

- It explains only the presence or absence of an item in the database.
- Obtaining non interesting rules
- Huge number of discovered rules
- Low algorithm performance

Methods to Improve Apriori's Efficiency

- **Hash-based itemset counting**: A k-itemset whose corresponding hashing bucket count is below the threshold cannot be frequent.
- **Transaction reduction**: A transaction that does not contain any frequent k-itemset is useless in subsequent scans.
- **Partitioning**: An itemset that is potentially frequent in DB must be frequent in at least one of the partitions of DB.
- **Sampling**: Mining on a subset of given data, lower support threshold and a method to determine completeness.
- **Dynamic itemset counting**: Add new candidate itemsets only when all of their subsets are estimated to be frequent.