

INTRODUCTION TO BUSINESS INTELLIGENCE

Unit Structure

- 1.0 Objectives
- 1.1 Introduction
- 1.2 Business Intelligence Operational and Decision Support System
- 1.3 Data-Information-Knowledge
- 1.4 Decision making-Action cycle
- 1.5 Business Intelligence
- 1.6 Data warehousing
- 1.7 Business Intelligence architecture
- 1.8 Use and benefits of Business Intelligence
- 1.9 Let us Sum Up
- 1.10 List of References
- 1.11 Bibliography
- 1.12 Unit End Exercise

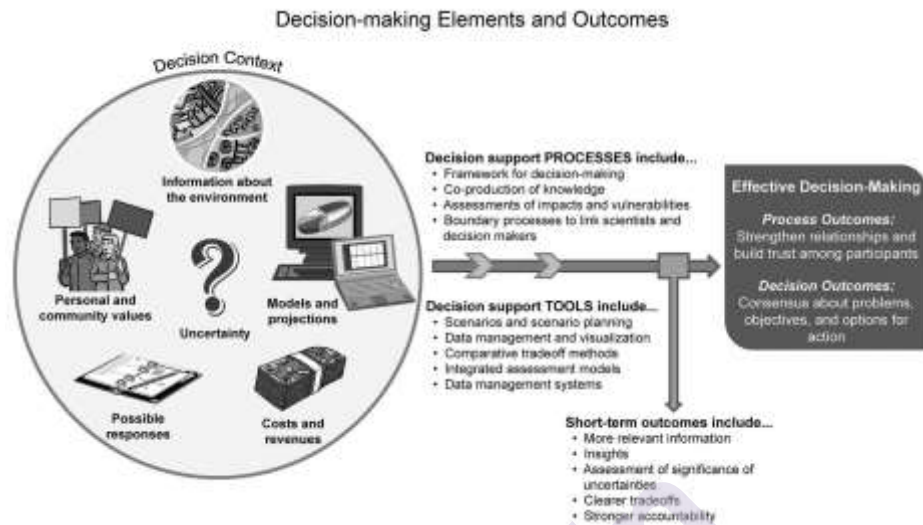
1.0 OBJECTIVES

- To gain knowledge about Business Intelligence.
- To Explore DSS
- To gain insight of DIKW Pyramid
- To explore benefits of BI

1.1 INTRODUCTION

Business intelligence (BI) helps organizations analyse historical and current data, so they can quickly uncover actionable insights for making strategic decisions. Business intelligence tools make this possible by processing large data sets across multiple sources and presenting findings in visual formats that are easy to understand and share.

1.2 BUSINESS INTELLIGENCE OPERATIONAL AND DECISION SUPPORT SYSTEM



A decision support system (DSS) is a computer-based information system that supports business or organizational decision-making activities; typically, this results in ranking, sorting, or choosing from among alternatives. DSSs serve the management, operations, and planning levels of an organization (usually mid and higher management) and help people make decisions about problems that may be rapidly changing and not easily specified in advance.

There are several types of DSSs that include:

1. **Communication-driven DSS** which enables cooperation, supporting more than one person working on a shared task; examples include integrated tools like Google Docs or Microsoft Groove.
2. **Document-driven DSS** which manages, retrieves, and manipulates unstructured information in a variety of electronic formats.
3. **Knowledge-driven DSS** provides specialized problem-solving expertise stored as facts, rules, procedures, or in similar structures
4. **Model-driven DSS** emphasizes access to and manipulation of a statistical, financial, optimization, or simulation model. Model-driven DSS use data and parameters provided by users to assist decision makers in analysing a situation; they are not necessarily data-intensive.
5. **Data-driven DSS** (or data-oriented DSS) emphasizes access to and manipulation of a time series of internal company data and, sometimes, external data. A data driven DSS, which we will focus on, emphasizes access to and manipulation of a time series of internal company data and sometimes external data. Simple file systems accessed by query and retrieval tools provide the most elementary level of functionality. Data warehouse systems that allow the manipulation of data by computerized tools tailored to a specific

task and setting or by more general tools and operators provide additional functionality. Data-driven DSS with online analytical processing (OLAP) provide the highest level of functionality.

Data warehousing combines the best of business practices and information systems technology and requires the cooperation of both business and IT, continuously coordinating in order to align all the needs, requirements, tasks, and deliverables of a successful implementation.

The need for a successful data warehouse implementation arises when reporting from a line of business database and a single report requires multiple table joins to get relevant data hence a slow rate of retrieval, when naming conventions are usually not enforced and thus it is difficult to know where the data you need is stored, when your organization may have several line of business applications working against a single or several databases and thus the data quality is low and not tracked over time.

Creation of a central repository for merged and historical data as opposed to the normalized relational schema represented in the above scenario has several advantages that include: simplified business reporting logic with performance gains, faster aggregations, and the ability to feed Online Analytical Processing systems with Star or snowflake schemas that cover multiple business areas.

The star schema separates business process data into facts and dimensions. Fact tables record measurements for a specific event generally consisting of numeric values, and foreign keys to dimensional data where descriptive information is kept. Fact tables are generally assigned a surrogate key to ensure each row can be uniquely identified. This key is a simple primary key not derived from source data. Dimension tables on the other hand are descriptive attributes related to fact data, usually having a relatively small number of records compared to fact tables, but each record may have a very large number of attributes to describe the fact data.

Dimensions can define a wide variety of characteristics dimension tables are generally assigned a surrogate primary key, usually a single-column integer data type, mapped to the combination of dimension attributes that form the natural key.

A snowflake schema, on the other hand, is an expansion and extension of a star schema to additional secondary dimensional tables. In a star schema, each dimension is typically stored in one table; the snowflake design principle expands a dimension and creates tables for each level of a dimensional hierarchy.

As far as the advantages go, Data Warehouses have demerits too and these include data integrity not being enforced as well as it is in a highly normalized database. One-off inserts and updates can result in data anomalies which normalized schemas are designed to avoid. Generally speaking, star schemas are loaded in a highly controlled fashion via batch processing or near-real time “trickle feeds”, to compensate for the lack of protection afforded by normalization.

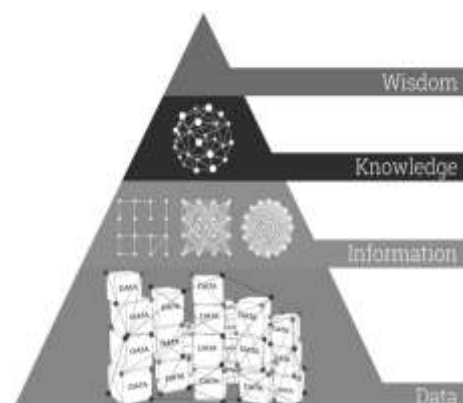
The Star schema is also not as flexible in terms of analytical needs as a normalized data model. Normalized models allow any kind of analytical queries to be executed as long as they follow the business logic defined in the model. Star schemas tend to be more purpose-built for a particular view of the data, thus not really allowing more complex analytics. Star schemas don't support many-to-many relationships between business entities – at least not very naturally. Typically, these relationships are simplified in star schema to conform to the simple dimensional model.

When a data warehouse is included as a component in a data driven DSS, a DSS analyst or data modeler needs to develop a schema or structure for the database and identify analytic software and end-user presentation software to complete the DSS architecture and design. The DSS components need to be linked in an architecture that provides appropriate performance and scalability. In some data driven DSS designs, a second multidimensional database management system (MDBMS) will be included and populated by a data warehouse built using a relational database management system (RDBMS). The MDBMS will provide data for online analytical processing (OLAP). It is common to build a data warehouse using an RDBMS from Microsoft and then use query and reporting and analytical software from a vendor such as Tableau or Business Objects as part of the overall data driven DSS design. What some vendors call “business intelligence software” provide the analytics and user interface functionality for a data driven DSS built with a data warehouse component.

1.3 DATA-INFORMATION-KNOWLEDGE

What is the Data, Information, Knowledge, Wisdom (DIKW) Pyramid?

The DIKW Pyramid represents the relationships between data, information, knowledge, and wisdom. Each building block is a step towards a higher level - first comes data, then is information, next is knowledge and finally comes wisdom. Each step answers different questions about the initial data and adds value to it. The more we enrich our data with meaning and context, the more knowledge, and insights we get out of it so we can take better, informed and data-based decisions.



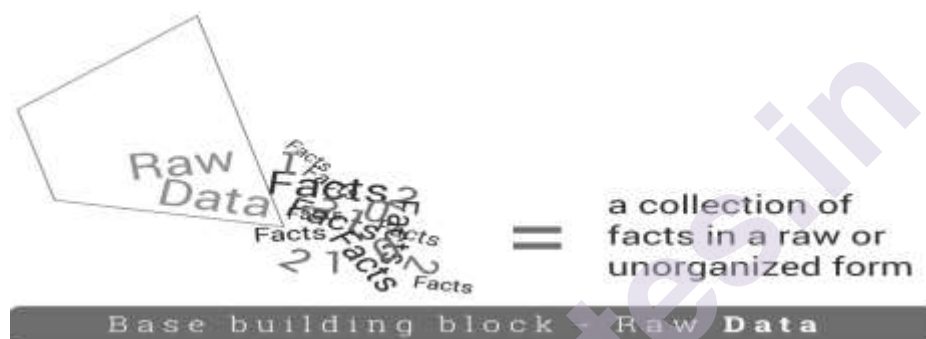
Each step up the pyramid answers questions about and adds value to the initial data.

Knowledge Pyramid, Wisdom Hierarchy and Information Hierarchy are some of the names referring to the popular representation of the relationships between data, information, knowledge and wisdom in the Data, Information, Knowledge, Wisdom (DIKW) Pyramid.

Like other hierarchy models, the Knowledge Pyramid has rigidly set building blocks – data comes first, information is next, then knowledge follows and finally wisdom is on the top.

Each step up the pyramid answers questions about the initial data and adds value to it. The more questions we answer, the higher we move up the pyramid. In other words, the more we enrich our data with meaning and context, the more knowledge, and insights we get out of it. At the top of the pyramid, we have turned the knowledge and insights into a learning experience that guides our actions.

Data

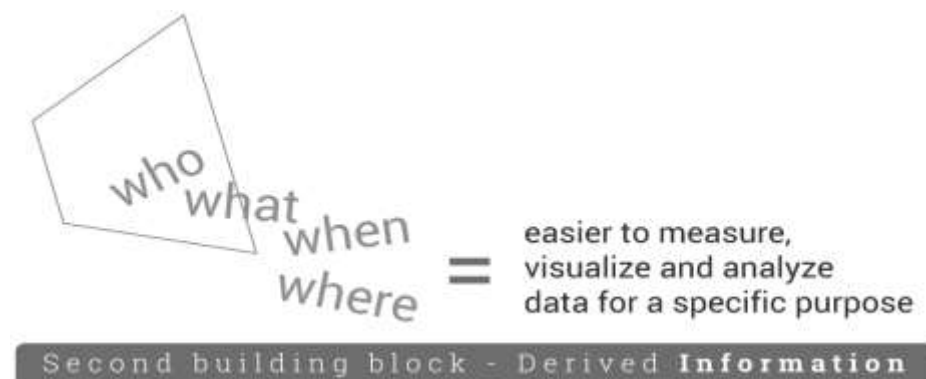


Data is a collection of facts in a raw or unorganized form such as numbers or characters.

However, without context, data can mean little. For example, 12012012 is just a sequence of numbers without apparent importance. But if we view it in the context of 'this is a date', we can easily recognize 12th of January 2012. By adding context and value to the numbers, they now have more meaning.

In this way, we have transformed the raw sequence of numbers into information.

Information



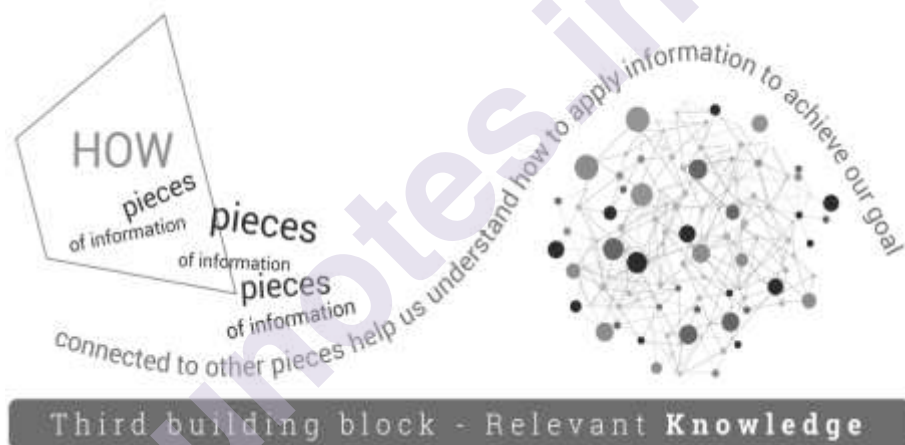
Information is the next building block of the DIKW Pyramid. This is data that has been “cleaned” of errors and further processed in a way that makes it easier to measure, visualize and analyse for a specific purpose.

Depending on this purpose, data processing can involve different operations such as combining different sets of data (aggregation), ensuring that the collected data is relevant and accurate (validation), etc. For example, we can organize our data in a way that exposes relationships between various seemingly disparate and disconnected data points. More specifically, we can analyse the Dow Jones index performance by creating a graph of data points for a particular period of time, based on the data at each day’s closing.

By asking relevant questions about ‘who’, ‘what’, ‘when’, ‘where’, etc., we can derive valuable information from the data and make it more useful for us.

But when we get to the question of ‘how’, this is what makes the leap from information to knowledge.

Knowledge



“How” is the information, derived from the collected data, relevant to our goals? “How” are the pieces of this information connected to other pieces to add more meaning and value? And maybe most importantly, “how” can we apply the information to achieve our goal?

When we don’t just view information as a description of collected facts, but also understand how to apply it to achieve our goals, we turn it into knowledge. This knowledge is often the edge that enterprises have over their competitors. As we uncover relationships that are not explicitly stated as information, we get deeper insights that take us higher up the DIKW pyramid.

But only when we use the knowledge and insights gained from the information to take proactive decisions, we can say that we have reached the final – ‘wisdom’ – step of the Knowledge Pyramid.

Wisdom



Wisdom is the top of the DIKW hierarchy and to get there, we must answer questions such as 'why do something' and 'what is best'. In other words, wisdom is knowledge applied in action.

We can also say that, if data and information are like a look back to the past, knowledge and wisdom are associated with what we do now and what we want to achieve in the future.

1.4 DECISION MAKING-ACTION CYCLE

Decision making is the process to select a course of action from a number of alternatives. Like planning, decision making also requires features like forecasting and action plans. For any organisation, policy documents help in taking managerial decisions. But these are decisions of routine nature, which we also call operational decisions. Strategic or important decisions are obviously taken after considering different alternatives. In order to be a successful manager, one has to necessarily develop decision-making skills.

What is a decision cycle?

It is a sequence of steps used by an organisation on a repeated basis to reach and implement decisions; not necessarily each decision adds to profits, but organisations must learn from mistakes made by them. For growth and sustainability, a business relies on decision cycle. The 'decision cycle' as a phrase is used to broadly categorise various methods of making decisions, going upstream to the need, downstream to the outcomes, and cycling around to connect the outcomes to the needs.

The examples of decision cycle are:

OODA loop:

This tool was coined in the 1950s by US Air Force colonel and military strategist John Boyd as a way to illustrate the action and decision cycle that a fighter pilot goes through during an aerial dogfight; it has since been applied to disciplines as diverse as business, medicine, law and the acquisition process in the military. The quality and speed of those decisions get enhanced by new command-and-control precepts and

advances in information, surveillance and reconnaissance tools, sensors and systems. As a result, military forces have been improving on their ability to observe, orient, decide and attack, better known as the OODA loop.

PDCA (plan-do-check-act): It was made popular by Dr W Edwards Deming, considered the father of modern quality control. Based on scientific method, PDCA can be better explained as “hypothesis” that can be proved under statistical control as a three-step process of specification, production and inspection. It can be specified as a scientific method of hypothesis, experiment and evaluation. According to Dr Deming, during his lectures in Japan in the early 1950s, the Japanese participants shortened the steps to the now traditional plan, do, check and act.

Herbert Simon Model:

The field of decision support systems was pioneered by Herbert Simon. According to Simon and his work with Allen Newell, decision making has distinct stages. He suggested for the first time the decision-making model of humans. It has three stages: Intelligence which deals with problem identification and data collection on the problem; design which deals with the generation of alternative solutions to the problem at hand; and choice which is selecting the “best” solution from amongst the alternative solutions using some criterion. Later, other scholars expanded his framework to five steps (Intelligence-Design-Choice-Implementation-Learning).

Business Analytics:

It refers to the skills, technologies, practices for continuously connecting exploration and investigation of past business performance to gain insight and drive business planning. BA focuses on developing new insights and understanding of business performance based on data and statistical methods. In contrast, business intelligence traditionally focuses on using a consistent set of metrics to both measure past performance and guide business planning, which is also based on data and statistical methods.

Design thinking:

It refers to cognitive, strategic, and practical processes by which design concepts, proposals for new products, buildings, machines, etc, are developed by designers. Many key concepts and aspects of design thinking have been identified through studies of design cognition and design activity in both laboratory and natural contexts. Design thinking includes “building up” ideas, with few or no limits. This helps reduce fear of failure among managers; it encompasses processes such as context analysis, problem finding and framing, ideation and solution, generating, creative thinking, sketching and drawing and portraying and evaluating.

1.5 BUSINESS INTELLIGENCE

What is business intelligence?

Business intelligence is an overarching term for the tools and technology used to analyse, visualize, benchmark, predict, and mine business data to

make better business decisions. BI technology allows businesses to assess current and historical data to gain actionable insights and predictive analysis for business operations.

BI tools may have one or more of these functionalities:

- Reporting
- Analytics and dashboard development
- Online analytical processing
- Data mining and process mining
- Complex event processing
- Business performance management
- Benchmarking
- Predictive and perspective analytics

1.6 DATA WAREHOUSING

What is Data Warehousing?

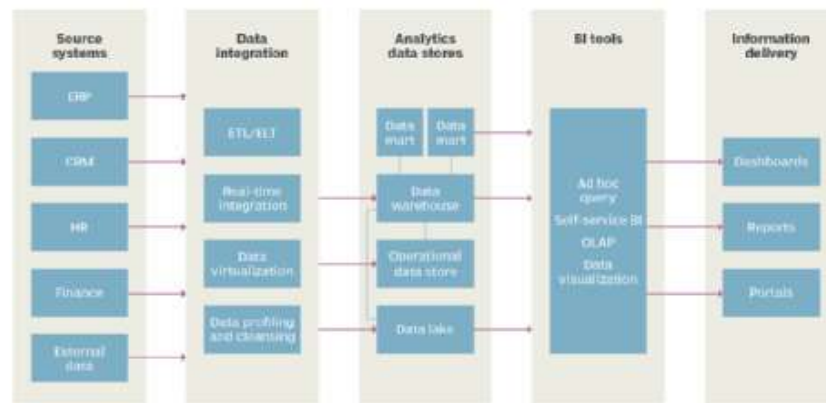
The data warehouse (DWH) is a repository of an organization's electronically stored data extracted from operational systems and made available for ad-hoc queries and scheduled reporting. In contrast, the process of building a data warehouse entail constructing and using a data model that can quickly generate insights. Data stored in the DWH is different from data found in the operational environment. It is organized so that relevant data is clustered together to facilitate day-to-day operations, analysis, and reporting. This helps determine the trends over time and allows users to create plans based on that information. Hence, reinforcing the importance of the use of warehouses in business.

1.7 BUSINESS INTELLIGENCE ARCHITECTURE

A business intelligence architecture is the framework for the various technologies an organization deploys to run business intelligence and analytics applications. It includes the IT systems and software tools that are used to collect, integrate, store, and analyse BI data and then present information on business operations and trends to corporate executives and other business users.

Business intelligence architecture components and diagram

A BI architecture can be deployed in an on-premises data centre or the cloud. In either case, it contains a set of core components that collectively support the different stages of the BI process, from data collection, integration, storage and analysis to data visualization, information delivery and the use of BI data in business decision-making.



Business Intelligence Architecture

As shown in the accompanying business intelligence architecture diagram, the core components include the following items.

- **Source systems.** These are all of the systems that capture and hold the transactional and operational data identified as essential for the enterprise BI program -- for example, ERP, CRM, finance, manufacturing, and supply chain management systems. They can also include secondary sources, such as market data and customer databases from outside information providers. As a result, both internal and external data sources are often incorporated into a BI architecture.

Important criteria in the data source selection process include data relevancy, data currency, data quality and the level of detail in the available data sets. In addition, a combination of structured, semi structured, and unstructured data types may be required to meet the data analysis and decision-making needs of executives and other business users.

- **Data integration and cleansing tools.** To effectively analyse the data collected for a BI program, an organization must integrate and consolidate different data sets to create unified views of them. The most widely used data integration technology for BI applications is extract, transform and load (ETL) software, which pulls data from source systems in batch processes. A variant of ETL is extract, load and transform (ELT), in which data is extracted and loaded as is and transformed later for specific BI uses. Other methods include real-time data integration, such as change data capture and streaming integration to support real-time analytics applications, and data virtualization, which combines data from different source systems virtually.

A BI architecture typically also includes data profiling and data cleansing tools that are used to identify and fix data quality issues. They help BI and data management teams provide clean and consistent data that's suitable for BI uses.

- **Analytics data stores:** This encompasses the various repositories where BI data is stored and managed. The primary one is a data warehouse, which usually stores structured data in a relational,

columnar, or multidimensional database and makes it available for querying and analysis. An enterprise data warehouse can also be tied to smaller data marts set up for individual departments and business units with data that's specific to their BI needs.

In addition, BI architectures often include an operational data store that's an interim repository for data before it goes into a data warehouse; an ODS can also be used to run analytical queries against recent transaction data. Depending on the size of a BI environment, a data warehouse, data marts and an ODS can be deployed on a single database server or separate systems.

A data lake running on a Hadoop cluster or other big data platform can also be incorporated into a BI architecture as a repository for raw data of various types. The data can be analysed in the data lake itself or filtered and loaded into a data warehouse for analysis. A well-planned architecture should specify which of the different data stores is best suited for particular BI uses.

- **BI and data visualization tools.** The tools used to analyse data and present information to business users include a suite of technologies that can be built into a BI architecture -- for example, ad hoc query, data mining and online analytical processing, or OLAP, software. In addition, the growing adoption of self-service BI tools enables business analysts and managers to run queries themselves instead of relying on the members of a BI team to do that for them.

BI software also includes data visualization tools that can be used to create graphical representations of data, in the form of charts, graphs and other types of visualizations designed to illustrate trends, patterns and outlier elements in data sets.

- **Dashboards, portals, and reports.** These information delivery tools give business users visibility into the results of BI and analytics applications, with built-in data visualizations and, often, self-service capabilities to do additional data analysis. For example, BI dashboards and online portals can both be designed to provide real-time data access with configurable views and the ability to drill down into data. Reports tend to present data in a more static format.

Other components that increasingly are part of a business architecture include data preparation software used to structure and organize data for analysis and a metadata repository, a business glossary and a data catalogue, which can all help users find relevant data and understand its lineage and meaning.

1.8 USE AND BENEFITS OF BUSINESS INTELLIGENCE

Use of Business Intelligence

A robust BI solution can help you bring together complex data and make informed business decisions in a span of minutes. Whether you are an industry giant, a midsize business, or an evolving start-up, we can use BI to turn business data into business opportunities.

Benefits of Business Intelligence

1. Enhance Business Productivity

In the race to reach and stay at the top, organizational productivity is often overlooked in business. But with BI tools, you can reach all your quantitative goals such as monthly sales or on-time delivery targets and track the progress of your business daily.

- Spot internal trends and get insight into underperforming processes
- Receive feedback on inefficiencies in your business
- Detect cost-cutting areas in your business
- Monitor your inventory and tweak production accordingly to increase profit margins
- Unearth industry patterns and insights and increase efficiency and forecasting outcomes

2. Improve Access to Crucial Information

With humongous amounts of data generated every second, accessing the right information at the right time to make a crucial business decision can get challenging. BI systems offer visualization tools that provide a better understanding of historical data, real-time updates, forecasts, and trends.

- Convert, merge, and report data with intuitive visuals
- Access important data through dashboards on mobile devices and tablets
- Get instant access to key business metrics for marketing and sales
- Extract crucial details from enormous data at a rapid speed
- Gauge all the trends and decide on the ideal course of action quickly

3. Boost ROI

When companies focus on things that are not aligned with the organizational strategy, they are sure to incur huge costs. BI enables you to establish metrics and KPIs that align with the organizational strategy, offering the needed visibility into business performance and ROI.

- Drive accountability by aligning activities and outcomes with the desired strategic objectives
- Identify areas for cost savings and improve business efficiency
- Leverage the numerous dashboards to improve visibility into inventory, and make better supply chain decisions
- Analyse the manufacturing process and access and collect the data necessary to measure all major productivity and production influencing factors
- Maximize production efficiency throughout the shop floor

4. Fuel Strategic Decision-Making

Making key business decisions based on intuition can spell disaster; it's preferable if they are made through analytics. BI software facilitates easy collection, rendering, and analysis of data using analytic tools.

- Get actionable insights by analysing data from business departments, social media, sales, marketing, and digital initiatives
- Spend less time formulating reports and more time in analysing potential outcomes and driving business decisions
- Minimize costs and efforts spent in preparing state-of-the-art reports and focus on analysing outcomes and driving better, more profitable decisions
- Strengthen your company's core and run a more efficient team with faster, better decisions at the forefront

5. Eliminate Waste

If you want to ensure business success, you need to first eliminate waste – anything that is not adding value to a company. BI systems help in identifying areas of waste, helping you improve your bottom line.

- Get a wider view of your company's statistics and locate areas of waste
- Identify the root causes of defects and eliminate them immediately
- Stay in tune with customer needs, market fluctuations, and business trends, and maintain optimum levels of inventory
- Eliminate bottlenecks in the production process, improve communication, and reduce idle time.
- Reduce gaps in the production process and improve forecasting methods to eliminate work in progress

6. Identify Opportunities

BI systems analyse unstructured data based on both qualitative and quantitative metrics, and aid in understanding what happened, and why it happened.

- Assess business capabilities, and compare strengths and weaknesses
- Identify trends and market conditions and respond quickly to change
- Evaluate performance in terms of customer and competitor experience
- Gain a 360-degree view of prospective business opportunities
- Make more informed choices and maximize profits, while cutting costs

Drive your Business Forward

The more people who have access to the right data, at the right place, and in the right form, the greater the value organizations will derive. IDC forecasts global spending on cognitive systems will reach nearly \$31.3 billion in 2019. Gathering data from your business processes and analysing it can help unearth some surprising and important insights.

BI software allows you to maximize business value by turning every employee into a decision maker. When armed with relevant, real-time information, employees can make data-driven, informed decisions that impact the company's bottom line. Increase the productivity of your business, improve access to crucial information, improve your ROI, drive strategic decision making, eliminate waste, identify business opportunities ahead of time, and take your business to newer, never-reached-before levels.

1.9 LET US SUM UP

- Business intelligence (BI) helps organizations analyse historical and current data, so they can quickly uncover actionable insights for making strategic decisions
- A decision support system (DSS) is a computer-based information system that supports business or organizational decision-making activities; typically, this results in ranking, sorting, or choosing from among alternatives.
- A business intelligence architecture is the framework for the various technologies an organization deploys to run business intelligence and analytics applications

1.10 LIST OF REFERENCES

- Data Strategy: How to Profit from A World of Big Data, Analytics and The Internet of Things by Bernard Marr
- Big Data Demystified: How to Use Big Data, Data Science and AI To Make Better Business Decisions and Gain Competitive Advantage" by David Stephenson PhD
- Performance Dashboards – Measuring, Monitoring, And Managing Your Business" by Wayne Eckerson
- Business Intelligence for Dummies" by Swain Scheps

1.11 BIBLIOGRAPHY

- Business Intelligence for Dummies" by Swain Scheps
- Data Strategy: How to Profit from A World of Big Data, Analytics and The Internet of Things by Bernard Marr

1.12 UNIT END EXERCISE

1. Explain Business Intelligence.
2. Explain in detail Business Intelligence Operational and Decision Support System.
3. Write short notes on:
a. Data b. Information c. Knowledge
4. Briefly explain Data warehousing.
5. Explain Business intelligence architecture components with diagram.
6. Write benefits of Business Intelligence.

munotes.in

KNOWLEDGE DISCOVERY DATABASE (KDD)

Unit Structure

- 2.0 Objectives:
- 2.1 Introduction
 - Knowledge Discovery Database (KDD)
 - Knowledge Discovery Database (KDD) process model
 - Data Pre-processing, Need of Data Pre-processing
 - What is Incomplete, Noisy, Inconsistent Data
 - Data Integration
 - Data Reduction
 - Data Transformation
- 2.2 Data Pre-Processing
- 2.3 Summary
- 2.4 Questions
- 2.5 References

2.1 INTRODUCTION

Knowledge Discovery in Database (KDD) is the process of discovering useful knowledge from the collection of data.

The main aim of KDD is to extract useful knowledge from the complex or huge volumes of data. KDD is the organized process of identifying valid, useful and understandable patterns from large and complex data sets.

Data Mining is the branch which is core of the KDD process. To explore the data, develop the model and discover previously unknown patterns these are the steps followed by the KDD. The Model is used for understanding phenomena from the data, analysis and prediction.

2.1.1 KNOWLEDGE DISCOVERY IN DATABASES PROCESS (KDD PROCESS) MODEL

KDD process model consisting of five steps. Each step of the procedure is iterative as well as interactive, which means that going back to earlier steps may be necessary. In the sense that one cannot offer a single formula or create a complete taxonomy, the process has numerous "artistic" characteristics. For making the best decisions for each step and type of application. As a result, it is necessary to comprehend the procedure, as well as the various requirements and options at each stage.

Knowledge Discovery in Databases application areas includes marketing, fraud detection, telecommunication and manufacturing industries. The primary goal of the KDD process is to extract high level knowledge from the low level data.

KDD applies data mining methods or algorithms to extract or identify what is the deemed knowledge, as per the specification of measures or thresholds using a database along with any required preprocessed, subsampling and transformation of that database.

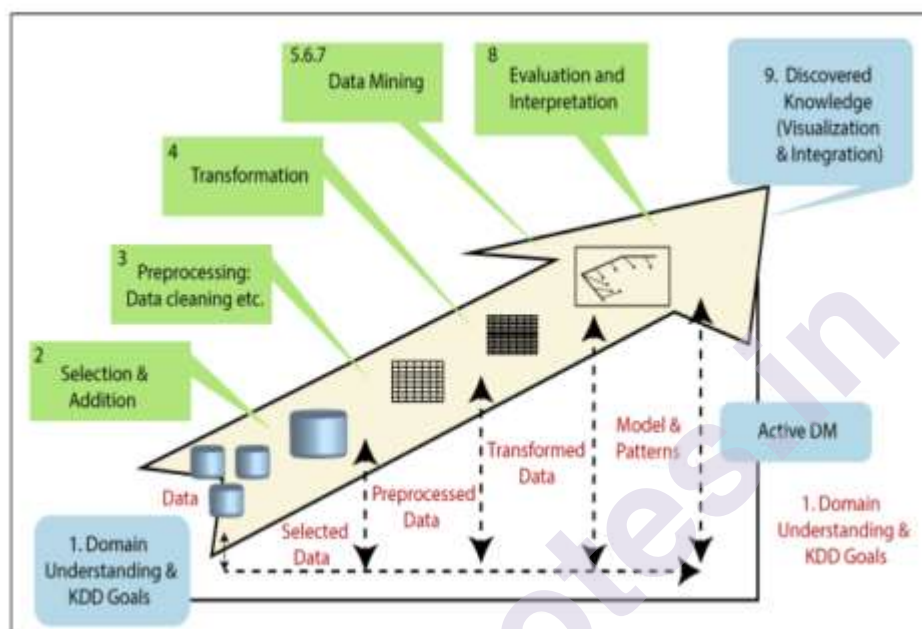


Figure The process of Knowledge Discovery in Databases

The process starts with KDD goals and ends with the implementation of the discovered knowledge. If one loop closes then the next loop starts.

Following are the nine steps for KDD process:

- Step 1) Developing an understanding of the application domain:
- Step 2) Selecting and creating a data section which discovery will be performed:
- Step 3) Preprocessing and cleaning:
- Step 4) Data transformation:
- Step 5) Prediction and description:
- Step 6) Choosing the Data Mining algorithm:
- Step 7) Employing the Data Mining algorithm:
- Step 8) Evaluation:
- Step 9) Using the discovered knowledge:

Step 1) Developing an understanding of the application domain:

This is the initial primary step. It is useful for developing the scene for understanding the problem domain with various decisions like

transformation, algorithms, representation etc. The person in charge of a KDD project need to understand and design the objectives of the problems of end user and the environment which is having the prior knowledge for knowledge discovery process.

Step 2) Selecting and creating a data section which discovery will be performed:

After defining the objectives, the next process of knowledge discovery is data determination. This process discovers, what data is accessible, finding out the important data, and later integration of all the data for knowledge discovery. Quality data is an essential part of it. This process is very important because of Data Mining learns as well as discovers from the data which is accessible. To organize, collect and operate data repositories is expensive. The interactive and iterative aspect KDD takes place. i.e. It starts with the available data sets and then expands, observes the effect in terms of knowledge discovery and modelling.

Step 3) Preprocessing and cleaning

Data dependability is increased in this step. It includes data cleaning, such as dealing with missing values and removing noise or outliers. In this case, it might apply complicated statistical techniques or a Data Mining programme. For example, if one suspects that a particular characteristic is unreliable or has a lot of missing data, that attribute could become the goal of the Data Mining supervised method. Following the creation of a prediction model for these properties, missing data can be forecasted. The extent to which one pays attention to this level is determined by a variety of circumstances. Regardless, it is critical to research the many components.

Step 4) Data transformation

This stage involves preparing and developing appropriate data for Data Mining. Dimension reduction (for example, feature selection and extraction, and record sampling) as well as attribute transformation are used here (for example, discretization of numerical attributes and functional transformation). This stage is often project-specific and can be critical to the success of the entire KDD project. In medical examinations, for example, the quotient of qualities, rather than each one individually, may be the most important component. We may need to consider influences beyond our control, as well as efforts and temporary challenges, in business.

Examining the effects of advertising accumulation, for example. However, if we don't start with the appropriate transformation, we might get a wonderful result that tells us about the transformation we'll need in the following iteration. As a result, the KDD process feeds back on itself, prompting an understanding of the essential transformation.

Step 5) Prediction and description

We're now ready to choose the type of Data Mining to apply, such as classification, regression, clustering, and so on. This is primarily dependent on the KDD objectives, as well as the preceding steps. Data Mining has two major goals: the first is to make a forecast, and the second is to make a description. Predictive Data Mining is commonly referred to as supervised Data Mining, whereas descriptive Data Mining includes the unsupervised and visualization components of Data Mining. Inductive learning is used in most Data Mining approaches, where a model is generated explicitly or implicitly by generalising from a sufficient number of preparatory models. The inductive approach's key assumption.

Step 6) Choosing the Data Mining algorithm:

Now that we've figured out the technique, it's time to figure out the tactics. This stage entails deciding on a technique to utilise while searching for patterns containing numerous inducers. When it comes to precision vs. understandability, for example, neural networks excel at the former while decision trees excel at the latter. There are numerous approaches to achieving meta-learning success for each system. Meta-learning is concerned with determining what causes a Data Mining algorithm to be successful or unsuccessful in a given situation. As a result, this methodology tries to figure out what kind of situation a Data Mining algorithm is most suited for. There are settings and techniques for each algorithm.

Step 7) Employing the Data Mining algorithm:

In this stage the implementation of algorithms of Data Mining, we can employ the algorithm many times until we reached at a satisfying conclusion or outcome. For example by turning the algorithm control parameter, like minimum number of instances in a single leaf of a decision tree.

Step 8) Evaluation:

In this phase, we evaluate and analyse the mined patterns, rules, and reliability in relation to the first-step goal. In this section, we look at the pretreatment procedures and how they affect the Data Mining method outcomes. Include a feature in step 4, for example, and then repeat the process. This step focuses on the induced model's readability and utility. The recognized knowledge is also documented in this phase for future use. The utilization, as well as overall feedback and discovery outcomes obtained by Data Mining, is the final phase.

Step 9) Using the discovered knowledge:

We are now ready to incorporate the knowledge into a different framework for further action. Knowledge becomes useful when it allows us to make changes to the system and track the results. The success of this stage determines the overall effectiveness of the KDD process. This phase

presents a number of obstacles, including the loss of the "laboratory settings" in which we previously operated. For instance, information was obtained from a static portrayal, which is usually a set of data, but the data has now become dynamic. Certain numbers may become inaccessible, and the data domain may be transformed, such as an attribute that may have a value that was not previously available.

2.1.2 Pros and Cons of KDD

Pros of KDD

- The ultimate goal is to extract high level knowledge from low level data.
- Artificial Intelligence also supports KDD by discovering empirical laws from experimentation and observation.
- KDD includes multidisciplinary activities.
- Cons of KDD
- KDD is a complex process and it fundamentally requires human participation.
- Requirement of expertise (KDD process handling)

2.2 DATA PRE-PROCESSING

Data Pre-processing is a Data Mining technique which is used to transform the raw data into organized format i.e. knowledgeable format.

The data which is available in real world is often incomplete, inconsistent or may lacking the certain behaviors or various pattern/trends, or may have errors.

2.2.1 Need of Data Pre-processing

- Data Pre-processing improves the quality of the data in data warehouse.
- In Data Mining technique where raw data is transforms into an knowledgeable or understandable format is always incomplete or may have errors, that raw data can not sent for the processing. So Data Pre-processing is a data mining technique to transform the data into the understandable or knowledgeable data.
- There are many factors which are comprising data quality, including accuracy, completeness i.e. all required fields are filled with data, consistency, timeliness i.e updation from time to time. Believability i.e. reflects how much the data are trusted by users as well as interpretability (reflects how easy the data are understood).

Incomplete data: Lacking attribute values or certain attributes of interest, or containing only aggregate data.

Inaccurate or noisy data: Data containing errors, or values that different from the expected data.

Inconsistent data: Data containing the discrepancies in the database.

2.2.2 Benefits of Data preprocessing

- Data preprocessing is useful in improving the quality of data in the data warehouse.
- Increases efficiency
- Ease of mining process
- Removes noisy data, inconsistent data as well as incomplete data.

2.2.3 Task of Data Preprocessing

- 1) Data Cleaning
- 2) Data Integration
- 3) Data Reduction
- 4) Data Transformation

2.2.3.1 DATA CLEANING

1) Data Cleaning:

Data cleaning is also known as Data cleansing. Data can have many missing as well as irrelevant part. To handle the irrelevant as well as missing part of the data. It is attempt to fill in missing values or noisy data.

There are three approaches in Data Cleaning

- i) Missing values
- ii) Noisy data
- iii) Inconsistent data

2.2.3.1.1 MISSING VALUES

- i) **Missing values:** This condition arises when some data is missing in the database.

It can be handled with various ways as follows:

step 1) Ignore the tuple.

step 2) Fill in the missing values manually.

step 3) Use a global constant to fill in the missing values e.g. NA

step 4) Use a measure of central tendency for the attribute.

e.g. Mean or Median to fill in the missing value.

Step 5) Use the most probable value to fill in the missing value.

e.g. Using Decision Tree

2.2.3.1.2 NOISY VALUES

- ii) **Noisy data:** Noisy data is a meaningless data. It can not be interpreted by machines. It generated because of faulty collection of data, errors during data entry.

Noisy is related to error or variance in a measured variables.

Incorrect attribute values may due to following reasons

- Faulty data collection instruments

- Data entry problems
- Data transmission problems
- Technology limitation
- Inconsistency in naming convention

For duplicate records, incomplete data and inconsistent data these are the problems occurs in noisy data.

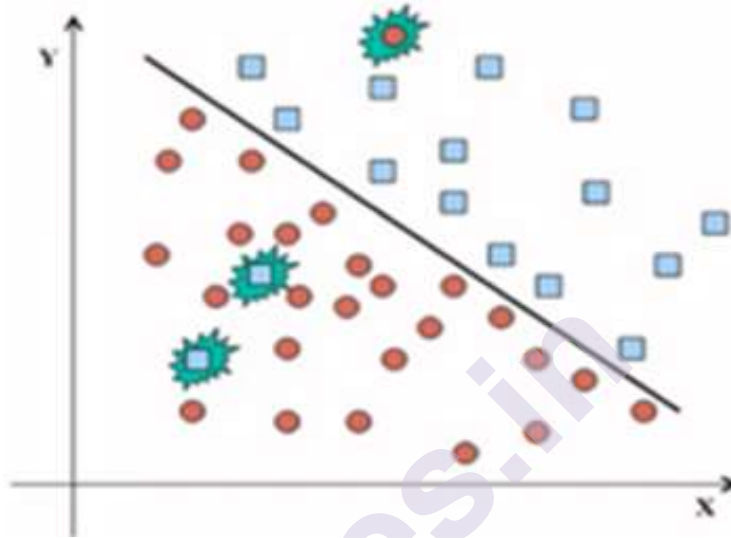


Figure: Noisy Data

It can be handled with the various ways as follows:

- i) Binning Method
- ii) Regression
- iii) Clustering

i) Binning Method:

Binning is the technique used to handle noisy data by data smoothing.

In Binning data is first sorted and then distributed into a number of buckets or bins. As binning methods consult the neighboring values then they perform local smoothing.

Steps for Binning method:

step 1) sort the data

step 2) Divide the data into equal number of intervals having same range.
i.e. equal dept partition.

All the partition must have equal number of elements.

Step 3) To smooth the data by 3 ways

1. Smoothing by bin means
2. Smoothing by bin medians
3. Smoothing by bin boundaries

1) Smoothing by bin means

How to create Bins of the Data.

1) Equal- depth partitioning frequency or equal height

Example:

Data : 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34

We will apply binning method step 1

Step 1) sort the data.

Now data : 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34

Now we will apply binning step 2

Step 2) Divide the data into equal number of intervals having same range.

i.e. equal dept partition.

All the partition must have equal number of elements.

Now Bins : Bin 1 = 4, 8, 9, 15

Bin 2 = 21, 21, 24, 25

Bin 3 = 26, 28, 29, 34

Mean of Bin 1 = $(4+8+9+15)/4 = 36/4 = 9$

So replace each element of Bin 1 to 9

i.e. Bin 1 contains 9, 9, 9, 9

Apply same step to Bin 2 and Bin 3

Bin 2 = 21, 21, 24, 25

Mean of Bin 2 = $(21+21+24+25)/4 = 91/4 = 22.75 = 23$

And replace each element of Bin 2 to 23

i.e. Bin 2 contains 23, 23, 23, 23

Bin 3 = 26, 28, 29, 34

Mean of Bin 3 = $(26+28+29+34)/4 = 117/4 = 29.25 = 29$

i.e. Bin 3 contains 29, 29, 29, 29

2) Smoothing by Bin Median

Bins : Bin 1 = 4, 8, 9, 15

Bin 2 = 21, 21, 24, 25

Bin 3 = 26, 28, 29, 34

Bin 1 : 4, 8, 9, 15

Middle elements are 8, 9 so

Bin 1 Median : $(8+9)/2 = 17/2 = 8.5 = 9$

So replace each element by the median i.e. 9

Bin 1 = 9, 9, 9, 9

Bin 2 = 21, 21, 24, 25

Middle elements are 21, 24 so

Bin 2 Median : $(21+24)/2 = 45/2 = 22.5 = 23$

So replace each element by the median i.e. 23

Bin 2 = 23, 23, 23, 23

Bin 3 = 26, 28, 29, 34

Middle elements are 28, 29 so

Bin 3 Median : $(28+29)/2 = 57/2 = 28.5 = 29$

So replace each element by the median i.e. 29

Bin 3 = 29, 29, 29, 29

3) Smoothing by Bin Boundaries

The boundaries of the bin values. i.e. left and right value will remain same.

Bin 1 values : 4, 8, 9, 15

Boundary values 4 and 15 so it will remain as it is.

In Bin 1: 4 15

Bin 2 values : 21, 21, 24, 25

Boundary values 21 and 25 so it will remain as it is.

In Bin 2: 21 25

Bin 3 values : 26, 28, 29, 34

Boundary values 26 and 34 so it will remain as it is.

In Bin 3: 26 34

Check the middle value and find out the nearest value and replace it by that boundaries.

In Bin 1 : 4, 8, 9, 15

Keep the boundaries as it is i.e. 4 15

Now compare 4 and 8 as well as 8 and 15 so which one is less we will consider it and replace with it other than boundaries values.

Now in Bin 1 : 4, 4, 4, 15

In Bin 2 : 21, 21, 24, 25

Keep the boundaries as it is i.e. 21 25

Now compare 21 and 21 as well as 24 and 25 so which one is less we will consider it and replace with it other than boundaries value.

Now in Bin 2 : 21, 21, 21, 25

In Bin 3 : 26, 28, 29, 34

Keep the boundaries as it is i.e. 26 34

Now compare 26 and 28 as well as 29 and 34 so which one less we will

Consider it and replace with it other than boundaries value.

Now in Bin 3 : 26, 26, 26, 34

ii) Regression :

Regression is the used to describe, predict and control the dependent variable on the basis of the independent variable.

Following are the types of Regression

1) Linear Regression –

In this method data are modelled to fit a straight line. Linear regression uses in the least-square method to the line.

Linear regression equation $Y = \alpha + \beta X$

Parameters α and β specifies the line for estimation by using the data which is in hand. Least square method is used in linear regression method and the known values of $Y_1, Y_2, \dots, X_1, X_2, \dots$

2) Multiple Regression –

In this method it allows a response variable Y modelled with the linear function of multidimensional feature vector.

Multiple Regression equation $Y = b_0 + b_1 X_1 + b_2 X_2$

By using this method many nonlinear functions can be transformed into multiple regression.

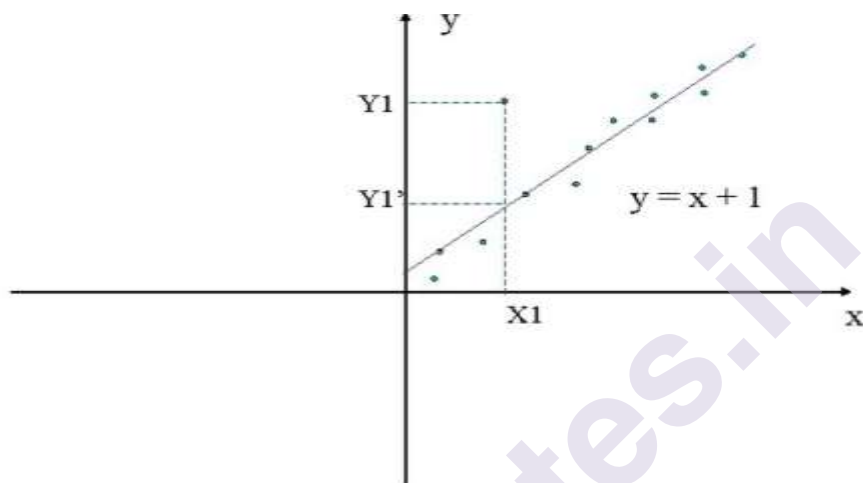


Figure:Regression

iii) Clustering :

Clustering is a collection of items (e.g., objects, events, etc., provided in a structured data set) is clustered into segments (or natural groupings) whose members share comparable properties. In contrast to classification, class labels are not used in clustering.

The clusters are formed as the chosen algorithm travels through the data set, detecting the commonalities of things based on their properties. Because of this, Clusters are determined by a heuristic algorithm, and because different algorithms produce different results, Before the results of the clustering are known, it's possible that multiple sets of clusters will emerge for the same data set.

It may be required for an e-commerce site to apply clustering algorithms.

After identifying reasonable clusters, they can be utilised to classify and understand additional data.

Clustering approaches, predictably, entail optimization. Clustering has a specific objective. That is to divide people into groups so that each group's members are as similar as possible.

Members of different groups share the smallest amount of similarities. K-means (from statistics) and self-organizing maps

(from machine learning) are two of the most often utilised clustering approaches.

Kohonen invented a novel neural network design called learning (1982).

Market segmentation with cluster analysis is a common usage of data mining tools by businesses.

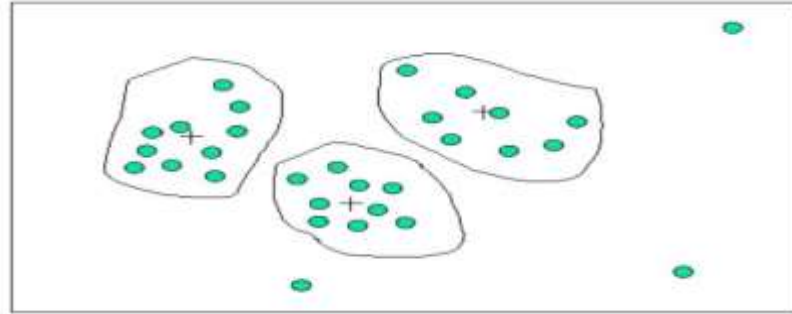


Figure : Clustering

2.2.3.1.3 INCONSISTENT DATA

iv) Inconsistent data:

For some transactions, there may be inconsistencies in the data recorded. Some data errors can be manually addressed by referring to external sources. A paper trace, for example, can be used to correct data entry mistakes. This might be combined with procedures to help address inconsistencies in code usage. Knowledge engineering techniques could also be used to detect data constraints violations. For instance, known functional connections between characteristics can be used to identify values that defy the functional restrictions.

There may also be inconsistencies due to data integration, where a given attribute can have different names in different databases. Redundancies may also exist.

Removal of Inconsistent data

- Manually, by using external references
- Knowledge engineering tools

Student ID	Student Name	Age	GPA	Classification
100122014	Joseph	21	3.5	Junior
100232015	Patrick	200	3.2	Sophomore
100122012	Seller	24	3.0	Senior
100342013	Roger	23	234	Senior
100942012	Davis	2.8	3.7	Sophomore
	Travis	23	3.4	Sr
100982015	Alex	27		Sophomore
100982013	Trevor	-22	4.0	Senior
AUC2016XC	Aman	30	3.5	Jr

Missing Data

Inconsistent Data

Noisy Data

2.2.4 OUTLIERS

Outliers:

Extreme values that deviate from the other observations in the given dataset. It occurs because of incorrect entry or calculation errors. Data points inconsistent with the majority of data.

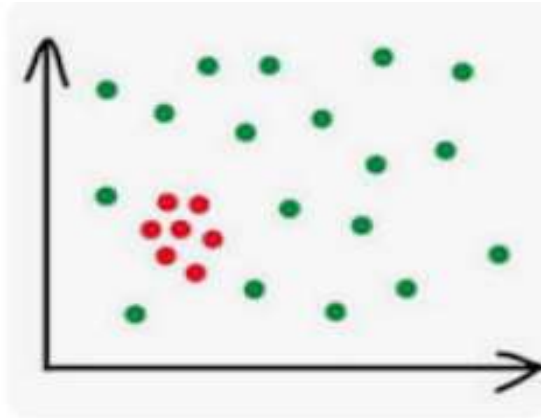


Figure: Outliers

2.2.5 DATA INTEGRATION

2) Data Integration

Data Integration is the technique for combining of data from the multiple sources into a coherent data store which is known as data warehouse. In Integration process multiple databases, data cubes, or files are used. It integrates metadata from the different sources.

Data Integration - Problems

Entity identification problem - It is that to identify real world entities from multiple data sources. E.g. A.student-id \equiv B.student-#.

Redundancy- It is a problem where same data is in multiple times occurs. i.e same attribute may have different names in different databases.

Redundant data may be able to detected by correlational analysis.

When we will do the carefully integration of data from multiple sources, it will help in reducing or avoiding redundancies and inconsistencies as well improvement in Data mining speed as well as quality.

Duplication of records-It is also a problem in Data Integration where the same tuples in the databases occurs multiple times.

Detection and resolving data value conflicts – It is the problem where different representations, different scales for example metric verses British units as well as for the real world entity, attribute values from the various sources are different. So, difficulty occurs in combining the multiple sources data.

2.2.6 DATA REDUCTION

3) Data Reduction

Data Reduction is a technique used to reduced representation of the data set which is much smaller in volume, in that way it will maintain the integrity of the original data.

i.e. Mining on the reduced data set must be efficient to produce the same or almost the same for analytical calculations.

Data warehouse may store terabytes of data which is having complex data analysis that time process of mining may take a very long time to run with the complete data set. So Reduction of data is essential process in mining. Data Reduction gives the result in reduced representation of the data set i.e. much compress or smaller in volume but still it produces the same or near about same analytical results.

Data Reduction – Strategies

- Data cube aggregation
 - It is aggregated data for an individual field of data i.e. the lowest level of data cube. Example A customer in a phone calling data warehouse.
 - Reduces the size of data to deal with multiple levels of aggregation in data cubes.
 - Use the smallest representation which is efficient to resolve the task by using reference related levels.
 - Solving different queries which are related to aggregated information that may be solved by using data cube.

The diagram illustrates the process of data cube aggregation. On the left, there are three stacked tables representing quarterly sales data for the years 2002, 2003, and 2004. The table for Year 2002 is expanded to show quarterly breakdowns. An arrow points to the right, where a single table shows the aggregated yearly sales totals.

Year 2004	
Quarter	Sales
Q1	0
Q2	0
Q3	0
Q4	0

Year 2003	
Quarter	Sales
Q1	0
Q2	0
Q3	0
Q4	0

Year 2002	
Quarter	Sales
Q1	\$224,000
Q2	\$408,000
Q3	\$350,000
Q4	\$586,000

Year	Sales
2002	\$1,568,000
2003	\$2,356,000
2004	\$3,594,000

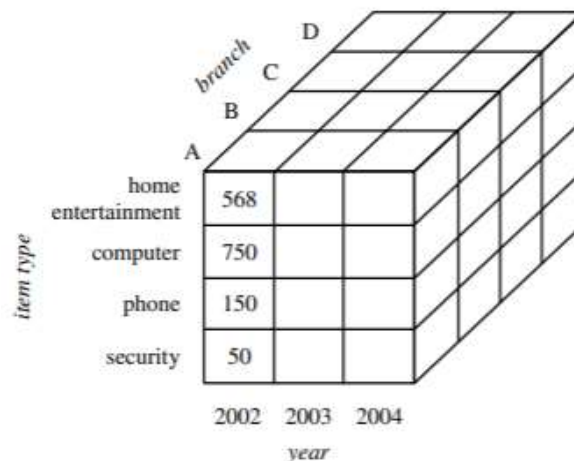


Figure : Data cube aggregation

- Attribute Subset Selection
 - Data sets for analysis may contain hundreds of attributes, from of which may be redundant, irrelevant for the mining process.
 - Example Student details have the attributes roll no, name, subject marks, birth date, blood group etc. for computing the percentage birth date and blood group attributes are irrelevant.
 - It is irrelevant, weakly relevant, or redundant attribute sets.
 - Dimensions may be detected and removed.

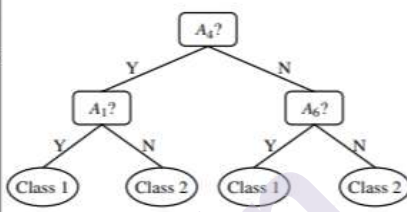
Forward selection	Backward elimination	Decision tree induction
Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ Initial reduced set: $\{\}$ $\Rightarrow \{A_1\}$ $\Rightarrow \{A_1, A_4\}$ \Rightarrow Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$ $\Rightarrow \{A_1, A_3, A_4, A_5, A_6\}$ $\Rightarrow \{A_1, A_4, A_5, A_6\}$ \Rightarrow Reduced attribute set: $\{A_1, A_4, A_6\}$	Initial attribute set: $\{A_1, A_2, A_3, A_4, A_5, A_6\}$  \Rightarrow Reduced attribute set: $\{A_1, A_4, A_6\}$

Figure : Attribute subset selection

- Dimensionality reduction
 - Feature Selection
 - It is selection of attribute subset. i.e. selection of minimum number of set of features so that the probability distribution of different classes given the values for those features is as close as possible to the original distribution given the values of all features.
 - It also reduces the number of patterns so that easy to understand.
- Heuristic methods
 - It is number of choices due to exponential.
 - Step-wise forward selection
 - Step-wise backward elimination
 - Combining forward selection and backward elimination
 - Decision-tree induction

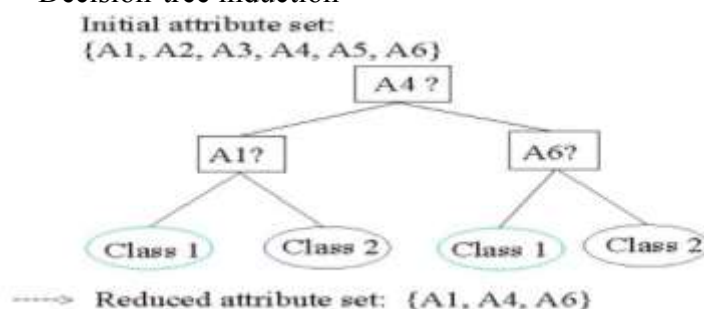


Figure : Decision tree induction

- **Numerosity reduction**

- There are two methods

- 1) Parametric methods

In this method assumption of data which fits with some model parameters, it stores only the parameters and removes the data which has outliers.

Long-linear model is a part of parametric method where obtained value at a point in $m - D$ space as the product on appropriate marginal substance.

- 2) Non-Parametric methods

In this method no assumption of data or models.

It uses histograms, clustering and sampling methods.

i) Histograms : It is a data reduction technique. It divides data into buckets and stores average or sum for each bucket. It is useful for constructed optimally for one dimensional data with the help of dynamic programming. It is related to quantization problems.

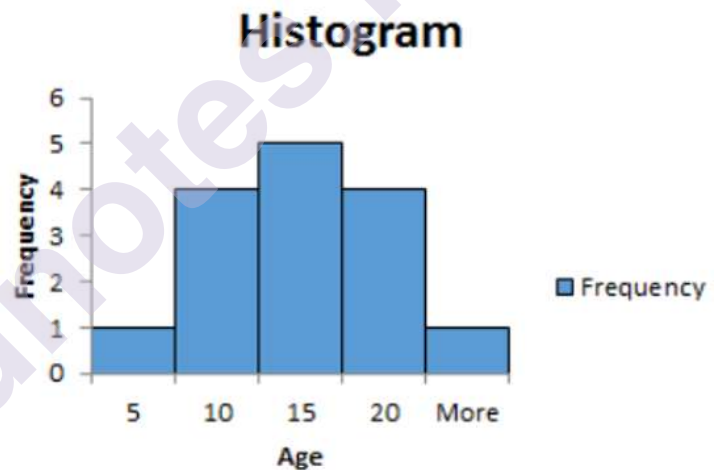


Figure : Histogram

- 2) **Clustering :**

It is a technique where data tuples consider as objects.

It partition data set or objects into different groups or clusters, and one can store cluster representation only.

Similar and dissimilar objects are prepared. Similar tends to how close the object for that data set and dissimilar object tends to vice versa.

The quality of a cluster represented by its diameter, the maximum distance between any two objects in the cluster.

Centroid distance is another method for measuring of cluster quality i.e. the average distance of each cluster object from the cluster centroid.

3) Sampling :

This technique is used for data reduction where it allows a large data set to be represented by much smaller random sample or subset of the data.

It allows a mining algorithm to run in complexity i.e. depends on size of the data. Sampling may not reduce database input/output.

Stratified Sampling: It uses approximate the percentage of each class or subpopulation of interest from the overall database. It uses conjunction with skewed data.

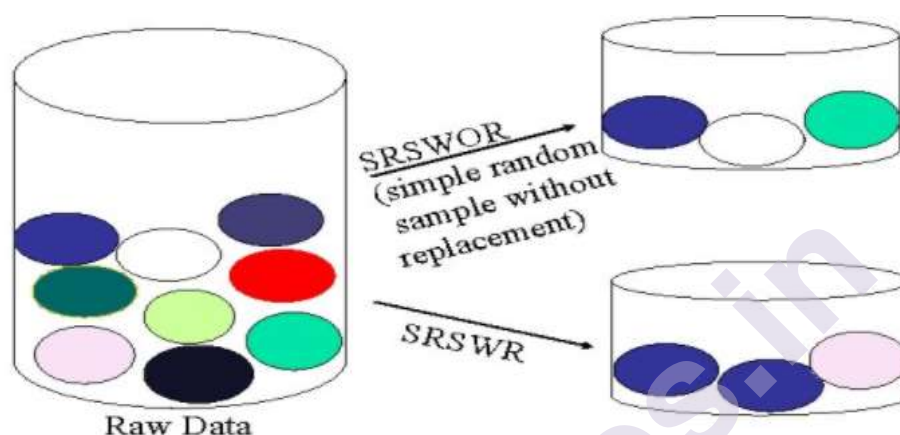
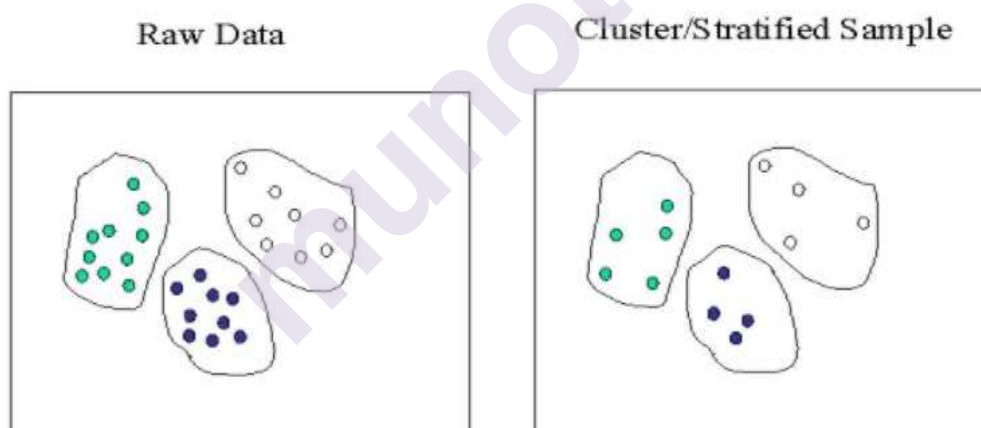


Figure : Sampling



Sampling can be used as a data reduction technique because it allows a large data set to be represented by a much smaller random sample (or subset) of the data. Suppose that a large data set, D , contains N tuples. Let's look at the most common ways that we could sample D for data reduction, as illustrated in following Figure

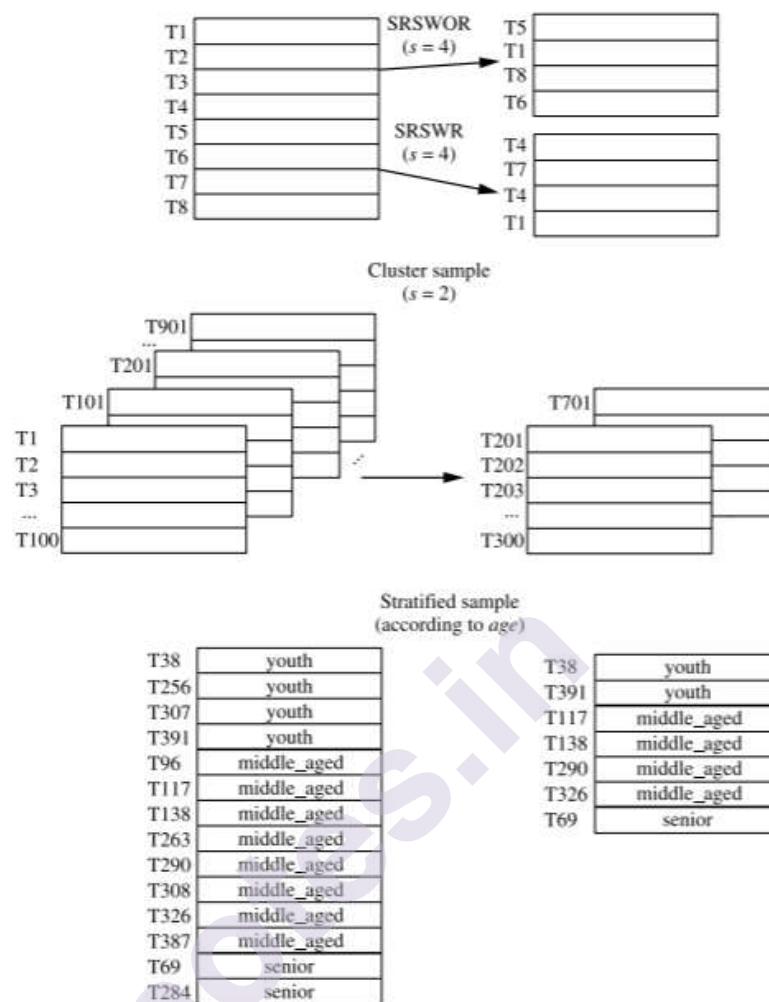


Figure : Sampling

- Simple random sample without replacement (SRSWOR) of size s : This is created by drawing s of the N tuples from D ($s < N$), where the probability of drawing any tuple in D is $1/N$, that is, all tuples are equally likely to be sampled.
- Simple random sample with replacement (SRSWR) of size s : This is similar to SRSWOR, except that each time a tuple is drawn from D , it is recorded and then replaced. That is, after a tuple is drawn, it is placed back in D so that it may be drawn again.
- Cluster sample: If the tuples in D are grouped into M mutually disjoint “clusters,” then an SRS of s clusters can be obtained, where $s < M$. For example, tuples in a database are usually retrieved a page at a time, so that each page can be considered a cluster. A reduced data representation can be obtained by applying, say, SRSWOR to the pages, resulting in a cluster sample of the tuples. Other clustering criteria conveying rich semantics can also be explored. For example, in a spatial database, we may choose to define clusters geographically based on how closely different areas are located.
- Stratified sample: If D is divided into mutually disjoint parts called strata, a stratified sample of D is generated by obtaining an SRS at

each stratum. This helps ensure a representative sample, especially when the data are skewed. For example, a stratified sample may be obtained from customer data, where a stratum is created for each customer age group. In this way, the age group having the smallest number of customers will be sure to be represented.

- An advantage of sampling for data reduction is that the cost of obtaining a sample is proportional to the size of the sample, s , as opposed to N , the data set size. Hence, sampling complexity is potentially sublinear to the size of the data. Other data reduction techniques can require at least one complete pass through D . For a fixed sample size, sampling complexity increases only linearly as the number of data dimensions, n , increases, whereas techniques using histograms, for example, increase exponentially in n .
- When applied to data reduction, sampling is most commonly used to estimate the answer to an aggregate query. It is possible (using the central limit theorem) to determine a sufficient sample size for estimating a given function within a specified degree of error. This sample size, s , may be extremely small in comparison to N . Sampling is a natural choice for the progressive refinement of a reduced data set. Such a set can be further refined by simply increasing the sample size
- Discretization and concept hierarchy generation:

Discretization

It divides the range of a continuous attribute into intervals. Interval labels can be used for replacing the actual data values.

Classification algorithm are useful for categorical attributes. It reduces the data size by discretization.

There are three types of attributes in Discretization

- 1) Nominal – Values from an unordered set.
- 2) Ordinal – Values from an ordered set.
- 3) Continuous – Real numbers

Hierarchy generation

It reduces the data by collecting and replacing the low level concepts by higher level concepts example : replace numeric values for the attribute age by young, middle-age or senior.

2.2.7 DATA TRANSFORMATION

4) Data Transformation

Data Transformation or consolidating data into mining appropriate form is known as Data Transformation.

Following are the steps for Data Transformation

- **Smoothing :**

This technique is used to remove noise from data. It consist of binning, regression and clustering.

- **Aggregation :**
This technique is used to summary or aggregation operations which are applied to the given data sets. By using summarization we can create data cube. For example : Daily Dmart sales data may be aggregated so that we can able to calculate monthly and annual total amount of sales. This technique is used in constructing a data cube for analysis of data at multiple granularities.
- **Generalization :**
In this technique concept hierarchy climbing is used where low-level or raw data are get replaced by higher level data concepts. For example : Data which is categorical attributes like street can be generalized to higher-level concept to city and country. As well as we can generalized numerical attribute to the categorical attribute. Example age (low level numeric attribute) generalized to youth, middle-aged or senior (higher level categorical attribute).
- **Normaltization :**
In this technique the attributes scaled to fall within a small, specified range i.e. -1.0 to 1.0 or 0.0 to 1.0
- **Min-max normalization**
It performs a linear transformation on the original data. Suppose $\min A$ and $\max A$ are the minimum and maximum values of an attribute. A Min-Max normalization maps a value, v , of A to v' in the range $[\text{new_min}A, \text{new_max}A]$
Min-max normalization is the relationship between the original data values. It will be encounter an “out-of-bounds” error if a future input case for normalization falls outside of the original data range for A .
- **Z-score normalization**
It is also known as zero-mean normalization. Attribute $\text{mean}A$, the values are normalized based on the $\text{mean}A$ and standard deviation of attribute mean. $\text{Mean}A$ value, v , of is normalized to v' by computing. This method of normalization is useful when the actual minimum and maximum of attribute A are unknown, or when there are outliers that dominate the min-max normalization.
- **Normalization by decimal scaling**
In this method it normalizes by moving the decimal point of values of attribute A . The number of decimal points moved depends on the maximum absolute value of A , A value v , of A is normalized to v' by formula as follows:
- **Attribute/feature construction**
In this technique new attribute constructed from the given data set. Attribute or the features which are already exist as well as new attributes which are required for the mining process is constructed in this phase.

2.3 SUMMARY

This chapter gives the details about Knowledge Discovery in Database (KDD), how we can extract the understandable or useful information and patterns in data. The process of extracting the details is known as KDD process. KDD process has nine steps, and each step is iterative and interactive.

Data Preprocessing is Data Mining technique of getting the quality data. How to remove noise, incomplete data, missing values by using data cleaning process. By using Data reduction how we can reduce the data with the help of different techniques like histogram, clustering and sampling. From various sources how we can integrate data by using a technique Data integration. Data Transformation is the technique smoothing, aggregation, generalization and normalization to get quality data after removing of all dirt from data for the process of mining.

2.4 QUESTIONS

- Q.1) What is KDD? Explain the process of KDD in detail.
- Q.2) Give the pros and cons of KDD.
- Q.3) What is Noisy data?
- Q.4) What is Binning?
- Q.5) Write a note on data smoothing.
- Q.6) Explain the binning technique for the following numbers
- 1) 2, 4, 7, 8, 10, 12, 13, 13, 17, 22, 27, 32
 - 2) 5, 8, 6, 10, 13, 18, 18, 22, 27, 29, 33, 39
 - 3) 9, 3, 5, 12, 16, 19, 19, 21, 24, 28, 32, 34
- Q.7) Explain in detail regression, clustering and sampling methods.
- Q.8) Write a note on:
- 1) Histogram
 - 2) Data Transformation
 - 3) Data Integration
 - 4) Outliers
- Q.9) What is Inconsistency in data? Explain with example.
- Q.10) Explain Data reduction in detail.

TEXT BOOKS

- 1) Business Intelligence and Analytics -Systems for Decision Support (10th Edition), Ramesh Sharda, Dursun Delen, Efraim Turban, Pearson publication
- 2) Business Intelligence (2nd Edition), Efraim Turban, Ramesh Sharda, Dursun Delen, David Kind, Pearson (2013)
- 3) Business Intelligence for Dummies, Swain Scheps, Wiley Publications (2009)

- 4) Data Mining: Introductory and Advanced Topics, Dunham, Margaret H, Prentice Hall (2006)

2.5 REFERENCES

- Data Modeling Techniques for Data Warehousing by IBM; International Technical Support Organization, Chuck Ballard, Dirk Herreman, Don Schau, Rhonda Bell, Eunsang Kim, Ann Valencic :<http://www.redbooks.ibm.com>
- Data Mining: Concepts and Techniques, The Morgan Kaufmann Series in Data Management Systems, Han J. and Kamber M. Morgan Kaufmann Publishers, (2000).

munotes.in

DATA TRANSFORMATION

Unit Structure

- 3.0 Objectives
- 3.1 Introduction
- 3.2 Binning for data discretization
- 3.3 Data Transformation By Normalization
- 3.4 Smoothing

3.0 OBJECTIVES

The chapter also deals with data transformation techniques like Discretization and Normalization. This chapter also introduces concepts dealing with smoothing which is used to reduce noisy data.

3.1 INTRODUCTION

One of the ways of data transformation is where the raw values of a numeric attribute (e.g., age) are replaced by interval labels. For example, we can have labels like 0–5, 6–10, etc. or have conceptual labels like youth, middle age, senior citizen etc. The labels, in turn, can be recursively organized into higher-level concepts, resulting in a concept hierarchy for the numeric attribute. The process through which we convert numeric data into interval labels is called discretization. Digitization will convert a large number of values into a few values or labels.

Figure 9.1 shows an example of discretization in terms of concept hierarchy for the attribute “price”. In the figure the price attribute can have different labels like \$0-\$200, \$200-\$400, \$400-\$600 etc., which can be further classified as \$0-\$100, \$100-\$200 etc. One can also define more than one concept hierarchy for the same attribute depending on the requirement.

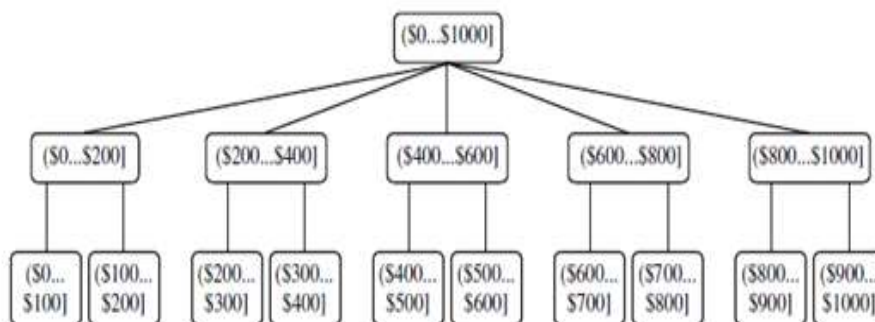


Figure 9.1 Source: Data Mining Concepts and Techniques Third Edition
 Jiawei Han University of Illinois at Urbana–Champaign Micheline
 Kamber Jian Pei Simon Fraser University

Discretization can be categorized as two types:

- Supervised –uses class information, e.g. Decision trees
- Unsupervised- uses direction of labelling (i.e., top-down vs. bottom-up) e.g. Binning, Histogram analysis, Cluster analysis

If the process starts by first finding one or a few points (called split points or cut points) to split the entire attribute range, and then repeats this recursively on the resulting intervals, it is called top-down discretization or splitting. This contrasts with bottom-up discretization or merging, which starts by considering the continuous values as potential split-points, removes some by merging neighborhood values to form intervals, and then recursively applies this process to the resulting intervals.

Data discretization and concept hierarchy generation are also forms of data reduction. The raw data are replaced by a smaller number of interval or concept labels. This simplifies the original data and makes the mining more efficient. The resulting patterns mined are typically easier to understand.

3.2 BINNING FOR DATA DISCRETIZATION

Binning is a method used for discretization. Here Binning is used for achieving data reduction and concept hierarchy generation. Binning is a top-down splitting technique based on a specified number of bins. The following two binning methods can be used for discretization:

- Equal-width binning
- Equal-frequency binning

Binning does not use class information and is therefore an unsupervised discretization technique.

3.2.1 Equal-width binning

In this method, the data is divided into k intervals of equal size. After the binning, all bins have equal width, or represent an equal range of the original variable values, no matter how many cases are in each bin.

The width of intervals is: $w = (\max - \min) / k$

And the interval boundaries are:

$\min + w, \min + 2w, \dots, \min + (k-1)w$

For example, given data **0,4,12,16,16,18,24,26,28** and the number of bins, **$k=3$**

Data: 0,4,12,16,16,18,24,26,28

Min = 0

Max = 28

$K=3$

$w = 28 - 0 / 3 = 28 / 3 = 9.33 = 10$ (rounded off to 10)

Interval boundaries are: $\min + w, \min + 2w, \dots, \min + (k-1)w = 0 + 10, 0 + 2 * 10, 0 + 3 * 10$

Interval boundaries are: 10, 20, 30. So there will be 3 bins: 0-10, 10-20, 20-30

Bins generated:

Bin1(upto 10): [0,4]

Bin 2 (Between 10,20) : [12,16,16,18]

Bin 3 (20+) ;[24,26,28]

3.2.2 Equal-frequency binning

In this method the data is divided into k groups in which each group contains approximately same number of values.

For example, given data **0,4,12,16,16,18,24,26,28** and the number of bins, $k=3$

Data: 0,4,12,16,16,18,24,26,28

Bins generated:

Bin1(upto 14): [0,4,12]

Bin 2 (Between 14,21) : [16,16,18]

Bin 3 (21+);[24,26,28]

3.3 DATA TRANSFORMATION BY NORMALIZATION

The measurement unit used can affect the data analysis. For example, changing measurement units from meters to inches for height, or from kilograms to pounds for weight, may lead to very different results. In general, expressing an attribute in smaller units will lead to a larger range for that attribute, and thus tend to give such an attribute greater weight. To help avoid dependence on the choice of measurement units, the data should be normalized or standardized.

Normalization involves transforming the data to fall within a smaller or common range such as $[-1,1]$ or $[0.0, 1.0]$. Normalizing the data is done to give all attributes an equal weight. Normalization is particularly useful for classification algorithms involving neural networks or distance measurements such as nearest-neighbor classification and clustering. In neural network backpropagation algorithm for classification mining, normalizing the input values for each attribute measured in the training tuples will help speed up the learning phase.

For distance-based methods, normalization helps prevent attributes with initially large ranges (e.g., income) from outweighing attributes with initially smaller ranges (e.g., binary attributes). It is also useful when given no prior knowledge of the data.

Some of the common methods data normalization are:

- min-max normalization
- z-score normalization,
- Normalization by decimal scaling

Let us consider A be a numeric attribute with n observed values, v_1, v_2, \dots, v_n .

3.3.1 Min-max normalization: This normalization performs a linear transformation on the original data. Suppose that \min_A and \max_A are the

minimum and maximum values of an attribute, A. Min-max normalization maps a value, v_i , of A to v' in the range $[\text{new_min}_A, \text{new_max}_A]$ by computing

$$v' = \frac{v - \min_A}{\max_A - \min_A} (\text{new_max}_A - \text{new_min}_A) + \text{new_min}_A$$

Source: <https://t4tutorials.com/data-normalization-before-data-mining/>
Min-max normalization preserves the relationships among the original data values. It will encounter an “out-of-bounds” error if a future input case for normalization falls outside of the original data range for A.

Example: Suppose that the minimum and maximum values for the attribute income are Rs.12,000 and Rs.98,000, respectively. The income needs to be mapped to the range $[0.0, 1.0]$. By min-max normalization, a value of Rs.73,600 for income is transformed to $73,600 - 12,000$ divided $98,000 - 12,000$ $(1.0 - 0.0) + 0.0 = 0.716$.

3.3.2 z-score normalization (or zero-mean normalization): Here the values for an attribute, A, are normalized based on the mean (i.e., average) and standard deviation of A. A value, v_i , of A is normalized to v' by computing $z = (x - \mu) / \sigma$

where μ and σ are the mean and standard deviation, respectively, of attribute A. This method of normalization is useful when the actual minimum and maximum of attribute A are unknown, or when there are outliers that dominate the min-max normalization.

Example: Suppose that the mean and standard deviation of the values for the attribute income are Rs.54,000 and Rs16,000, respectively. With z-score normalization, a value of Rs.73,600 for income is transformed to $73,600 - 54,000$ divided $16,000 = 1.225$.

3.3.3 Normalization by decimal scaling: This is done by moving the decimal point of values of attribute A. The number of decimal points moved depends on the maximum absolute value of A. A value, v_i , of A is normalized to v' by computing

$$\text{Normalized value of attribute} = (v_i / 10^j)$$

where j is the smallest integer such that $\max(|v_i|) < 1$

Example: Suppose that the recorded values of A range from -986 to 917 . The maximum absolute value of A is 986 . To normalize by decimal scaling, we therefore divide each value by 1000 (i.e., $j = 3$) so that -986 normalizes to -0.986 and 917 normalizes to 0.917 .

3.4 SMOOTHING

Smoothing is a data transformation technique which works to remove noise from the data. Given some numeric attribute such as, say, price, smoothing can be used to “smooth” out the data to remove the noise. Binning, Regression, Clustering are the techniques used in smoothing.

3.4.1 Binning for data smoothing

Binning methods smooth a sorted data value by consulting its “neighborhood”, that is, the values around it. It is sensitive to the user-specified number of bins, as well as the presence of outliers. The sorted values are distributed into a number of “buckets,” or bins. Because binning methods consult the neighborhood of values, they perform local smoothing. The following methods are used for smoothing using binning method

- Partition into (equal frequency) bins
- smoothing by bin means, median
- smoothing by bin boundaries

To explain binning, take the following sample data

Data: Sorted data for price (in Rupees): 4, 8, 15, 21, 21, 24, 25, 28, 34

Partition into (equal frequency) bins

The data for price are first sorted and then partitioned into equal-frequency bins of size 3 (i.e., each bin contains three values).

Bin 1:[4,8,15]

Bin 2:[21,21,24]

Bin 3:[25,28,34]

Smoothing by bin means and median: Each value in a bin is replaced by the mean value of the bin. To find the mean of a bin, add together all of data points and then divide that sum by the total number of data points in the bin. For example, the mean of the values 4, 8, and 15 in Bin 1 is 9. Therefore, each original value in this bin is replaced by the value 9.

Bin 1:[4,8,15]

Bin 2:[21,21,24]

Bin 3:[25,28,34]

Bin 1: mean of the values 4, 8, and 15 : $4+8+15/3=9$

So after transformation, values in Bin1:[9,9,9]

Bin2: mean of the values 21,21,24: $21+21+24/3=66/3=22$

So after transformation, values in Bin2:[22,22,22]

Bin3: mean of the values 25,28,34: $87/3=29$

So after transformation, values in Bin3:[29,29,29]

Similarly, smoothing by **bin medians** can be employed, in which each bin value is replaced by the bin median.

Smoothing by bin boundaries

In smoothing by bin boundaries, the minimum and maximum values in a given bin are identified as the bin boundaries. Each bin value is then replaced by the closest boundary value.

Bin 1:[4,4,15]

Bin 2:[21,21,24]

Bin 3:[25,25,34]

3.4.2 Regression

Data smoothing can also be done by regression, a technique that conforms data values to a function. Linear regression involves finding the “best” line to fit two attributes (or variables) so that one attribute can be used to predict the other. Multiple linear regression is an extension of linear regression, where more than two attributes are involved and the data are fit to a multidimensional surface.

3.4.3 Clustering

Clustering techniques consider data tuples as objects. In clustering data is partitioned into groups, or clusters, so that the data objects which lie within a cluster are similar. They should also be dissimilar to objects in other clusters. Similarity is based on closeness of the data in space, based on a distance function. The quality of a cluster may be represented by its diameter and the maximum distance between any two data objects in the cluster. Centroid distance is an alternative measure of cluster quality and is defined as the average distance of each cluster data from the cluster centroid (denoting the “average object,” or average point in space for the cluster).

In data reduction, the cluster representations of the data are used to replace the actual data. The effectiveness of this technique depends on the data’s nature. It is much more effective for data that can be organized into distinct clusters than for smeared data.

Conclusion

This chapter examined some of the concepts related to data transformation like discretization and normalization. It introduced Equal-width binning, Equal-frequency binning for data discretization.

The chapter also introduced Min-max normalization, z-score normalization, normalization by decimal scaling. Smoothing which is used for cleaning noisy data was also explored.

MCQ Questions

1. _____ is a data transformation technique which works to remove noise from the data.

a. Smoothing b. tuning c. normalisation d. discretization

Ans a. Smoothing

2. _____ will convert a large number of values into a few values or labels.

a. Smoothing b. tuning c. normalisation d. discretization

Ans d. discretization

3. _____ a technique that conforms data values to a function.

a. regression b. tuning c. normalisation d. discretization

Ans a. regression

4. _____ is Supervised technique for discretization

a. Decision trees b. Binning c. Histogram analysis d. Cluster analysis

Ans a. Decision trees

Descriptive Questions

1. What is discretization? What are the various techniques used for discretization?
2. Explain binning for data discretization?
3. What is Normalization? What are the types Normalization?
4. Explain the different types of Smoothing with examples?

munotes.in

INTRODUCTION TO BUSINESS DATA WAREHOUSE

Unit Structure

- 4.0 Objective
- 4.1 Introduction
- 4.2 Definition of Data warehouse,
- 4.3 Logical architecture of Data Warehouse,
- 4.4 Data Warehouse model-
- 4.5 Populating business
- 4.6 Summary
- 4.5 Exercise

4.0 OBJECTIVE

In this chapter we are going to,

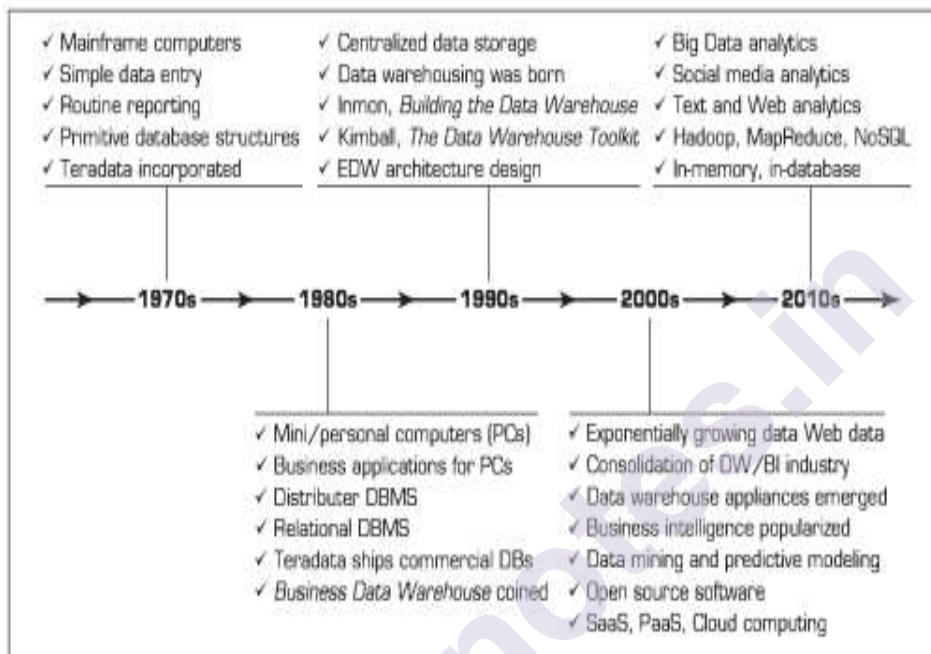
- Understand the basic definitions and concepts of data warehouses
- Explain the role of data warehouses in decision support
- Understand data warehousing architectures
- Explain data integration and the extraction, transformation , and load (ETL) processes
- Describe the processes used in developing and managing data warehouses
- Describe real-time (active) data warehousing
- Explain data warehousing operations
- Understand data warehouse administration and security issues

4.1 INTRODUCTION

- The concept of data warehousing has been around since the late 1980s.
- This chapter provides the foundation for an important type of database, called a data warehouse, which is primarily used for decision support and provides improved analytical capabilities.
- In computing, a data warehouse (DW or DWH), also known as an enterprise data warehouse (EDW), is **a system used for reporting and data analysis** and is considered a core component of business intelligence.
- DWs are central repositories of integrated data from one or more disparate sources.

- Decision makers require concise, dependable information about current operations, trends, and changes.
- Data are often fragmented in distinct operational systems, so managers often make decisions with partial information , at best.
- Data warehousing cuts through this obstacle by accessing, integrating, and organizing key operational data in a form that is consistent, reliable, timely, and readily available, wherever and whenever needed.

A List of Events That Led to Data Warehousing Development.



4.2 DEFINITION OF DATA WAREHOUSE

Definition:

In simple terms, a data warehouse (DW) is a pool of data produced to support decision making; it is also a repository of current and historical data of potential interest to managers throughout the organization.

- Data is usually structured to be available in a form ready for analytical processing activities (i.e., online analytical processing [OLAP], data mining, querying, reporting, and other decision support applications).
- A data warehouse is a subject-oriented, integrated, time-variant, nonvolatile collection of data in support of management's decision-making process.

Benefits of a data warehouse:

Organizations that use a data warehouse to assist their analytics and business intelligence see a number of substantial benefits:

- **Better data**
 - Adding data sources to a data warehouse enables organizations to ensure that they are collecting consistent and relevant data from that source.
 - They don't need to wonder whether the data will be accessible or inconsistent as it comes into the system.
 - This ensures higher data quality and data integrity for sound decision making.
- **Faster decisions**
 - Data in a warehouse is in such consistent formats that it is ready to be analyzed.
 - It also provides the analytical power and a more complete dataset to base decisions on hard facts.
 - Therefore, decision makers no longer need to rely on hunches, incomplete data, or poor quality data and risk delivering slow and inaccurate results.

Characteristics of Data Warehousing:

A common way of introducing data warehousing is to refer to its fundamental characteristics (see Inmon, 2005):

- **Subject oriented:**
 - Data are organized by detailed subject, such as sales, products, or customers, containing only information relevant for decision support.
 - Subject orientation enables users to determine not only how their business is performing, but why.
 - A data warehouse differs from an operational database in that most operational databases have a product orientation and are tuned to handle transactions that update the database.
 - Subject orientation provides a more comprehensive view of the organization.
- **Integrated:**
 - Integration is closely related to subject orientation.
 - Data warehouses must place data from different sources into a consistent format.
 - To do so, they must deal with naming conflicts and discrepancies among units of measure.
 - A data warehouse is presumed to be totally integrated.
- **Time variant (time series):**
 - A warehouse maintains historical data.
 - The data do not necessarily provide current status (except in real-time systems).

- They detect trends, deviations, and long-term relationships for forecasting and comparisons, leading to decision making.
- Every data warehouse has a temporal quality. Time is the one important dimension that all data warehouses must support.
- Data for analysis from multiple sources contains multiple time points (e.g., daily, weekly, monthly views).
- **Nonvolatile:**
 - After data is entered into a data warehouse, users cannot change or update the data. Obsolete data is discarded, and changes are recorded as new data.
- **Web based:**
 - Data warehouses are typically designed to provide an efficient computing environment for Web-based applications.
- **Relational/multidimensional:**
 - A data warehouse uses either a relational structure or a multidimensional structure.
 - A recent survey on multidimensional structures can be found in Romero and Abell6 (2009).
- **Client Server:**
 - A data warehouse uses the client/server architecture to provide easy access for end users.
- **Real time:**
 - Newer data warehouses provide real-time, or active, data-access and analysis capabilities (see Basu, 2003; and Bonde and Kuckuk, 2004).
- **Include metadata:**
 - A data warehouse contains metadata (data about data) about how the data are organized and how to effectively use them.

Comparison between Database and Data Warehouse:

Database	Data Warehouse
It is data collected for multiple transactional purposes.	It is aggregated transactional data, transformed and stored for analytical purposes.
Optimized for read/write access.	Optimized for aggregation and retrieval of large data sets.
Databases are made to quickly record and retrieve information.	Data warehouses store data from multiple databases, which makes it easier to analyze.

Databases are used in data warehousing. However, the term usually refers to an online, transactional processing database. There are other types as well, including csv, html, and Excel spreadsheets used for database purposes.

A data warehouse is an analytical database that layers on top of transactional databases to allow for analytics.

4.3 LOGICAL ARCHITECTURE OF DATA WAREHOUSE

- A logical data warehouse (LDW) is a data management architecture in which an architectural layer sits on top of a traditional data warehouse, enabling access to multiple, diverse data sources while appearing as one “logical” data source to users.
- Essentially, it is an analytical data architecture that optimizes both traditional data sources (databases, enterprise data warehouses, data lakes, etc.) and other data sources (applications, big data files, web service, and the cloud) to meet every analytics use case.
- The term was introduced in 2009 and continues to gain traction in the market as data complexity becomes a growing problem for many companies.
- The logical data warehouse is being called the next generation of data warehouse with the ability to meet companies growing data management needs.
- Combining multiple engines and various data sources across the enterprise, logical data warehouse components can be combined in one place logically instead of physically.
- The modern LDW has advanced to support today's wide variety of available data sources, data platforms, and business use cases.
- It helps organizations digitally reinvent, enable real-time streaming analytics, and optimize operations with smarter, data-driven decision making.

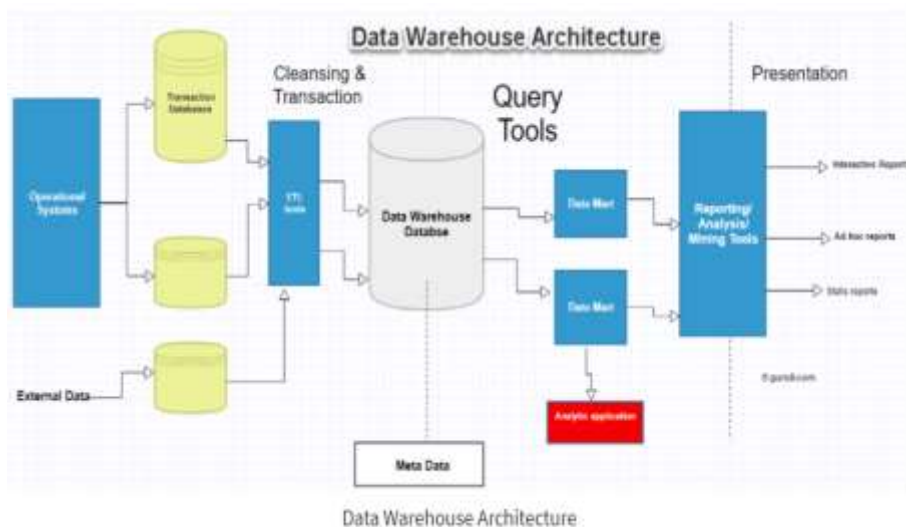
Common Characteristics of a Logical Data Warehouse:

Modern LDW tools now usually include the following characteristics:

- Application access through a single interface
- Existing enterprise data warehouse remains
- Contains one or more data lakes as repositories
- Uses an operational data store (ODS)
- Ensure consistency with data marts
- Set metadata and governance policies

Data Warehouse Architecture:

Data Warehouse Architecture is complex as it's an information system that contains historical and commutative data from multiple sources.



There are 3 approaches for constructing Data Warehouse layers:

1. Single Tier,
2. Two tier and
3. Three tier.

This 3 tier architecture of Data Warehouse is explained as below.

Single-tier architecture

- The objective of a single layer is to minimize the amount of data stored.
- This goal is to remove data redundancy. This architecture is not frequently used in practice.

Two-tier architecture

- Two-layer architecture is one of the Data Warehouse layers which separates physically available sources and data warehouses.
- This architecture is not expandable and also not supporting a large number of end-users.
- It also has connectivity problems because of network limitations.

Three-Tier Data Warehouse Architecture

- This is the most widely used Architecture of Data Warehouse.
- It consists of the Top, Middle and Bottom Tier.

Bottom Tier:

- The database of the Datawarehouse servers as the bottom tier.
- It is usually a relational database system.
- Data is cleansed, transformed, and loaded into this layer using back-end tools.

Middle Tier:

- The middle tier in Data warehouse is an OLAP server which is implemented using either ROLAP or MOLAP model.

- For a user, this application tier presents an abstracted view of the database.
- This layer also acts as a mediator between the end-user and the database.

Top-Tier:

- The top tier is a front-end client layer.
- Top tier is the tools and API that you connect and get data out from the data warehouse.
- It could be Query tools, reporting tools, managed query tools, Analysis tools and Data mining tools.

Data Warehouse Components:

1. Data Warehouse Database
2. Sourcing, Acquisition, Clean-up and Transformation Tools (ETL)
3. Metadata
4. Query Tools
5. Data warehouse Bus Architecture

1. Data Warehouse Database

- The central database is the foundation of the data warehousing environment.
- This database is implemented on the RDBMS technology.
- ad-hoc query, multi-table joins, aggregates are resource intensive and slow down performance.
- Hence, alternative approaches to Database are used as listed below-
 - In a datawarehouse, relational databases are deployed in parallel to allow for scalability.
 - Parallel relational databases also allow shared memory or shared nothing model on various multiprocessor configurations or massively parallel processors.
 - New index structures are used to bypass relational table scan and improve speed.
 - Use of multidimensional database (MDDBs) to overcome any limitations which are placed because of the relational Data Warehouse Models.
 - Example: Essbase from Oracle.

2. Sourcing, Acquisition, Clean-up and Transformation Tools (ETL)

- The data sourcing, transformation, and migration tools are used for performing all the conversions, summarizations, and all the changes needed to transform data into a unified format in the datawarehouse.
- They are also called Extract, Transform and Load (ETL) Tools.

- Their functionality includes:
 - Anonymize data as per regulatory stipulations.
 - Eliminating unwanted data in operational databases from loading into Data warehouse.
 - Search and replace common names and definitions for data arriving from different sources.
 - Calculating summaries and derived data
 - In case of missing data, populate them with defaults.
 - De-duplicated repeated data arriving from multiple datasources.
- These Extract, Transform, and Load tools may generate cron jobs, background jobs, Cobol programs, shell scripts, etc. that regularly update data in datawarehouse.
- These tools are also helpful to maintain the Metadata.
- These ETL Tools have to deal with challenges of Database & Data heterogeneity.

3. Metadata

- The name Meta Data suggests some high-level technological Data Warehousing Concepts. However, it is quite simple.
- Metadata is data about data which defines the data warehouse.
- It is used for building, maintaining and managing the data warehouse.
- In the Data Warehouse Architecture, meta-data plays an important role as it specifies the source, usage, values, and features of data warehouse data. It also defines how data can be changed and processed.
- It is closely connected to the data warehouse.

For example, a line in sales database may contain:

4030 KJ732 299.90

This is a meaningless data until we consult the Meta that tell us it was

Model number: 4030

Sales Agent ID: KJ732

Total sales amount of \$299.90

- Therefore, Meta Data are essential ingredients in the transformation of data into knowledge.
- Metadata can be classified into following categories:
- **Technical Meta Data:**
This kind of Metadata contains information about warehouse which are used by Data warehouse designers and administrators.

- **Business Meta Data:**

This kind of Metadata contains detail that gives end-users a way to easily understand information stored in the data warehouse.

4. Query Tools

- One of the primary objects of data warehousing is to provide information to businesses to make strategic decisions.
- Query tools allow users to interact with the data warehouse system.
- These tools fall into four different categories:
 - a. Query and reporting tools
 - Query and reporting tools can be further divided into
 - Reporting tools
 - Reporting tools can be further divided into production reporting tools and desktop report writer.
 - Report writers: This kind of reporting tool are tools designed for end-users for their analysis.
 - Production reporting: This kind of tools allows organizations to generate regular operational reports. It also supports high volume batch jobs like printing and calculating. Some popular reporting tools are Brio, Business Objects, Oracle, PowerSoft, SAS Institute.
 - Managed query tools
 - This kind of access tools helps end users to resolve snags in database and SQL and database structure by inserting meta-layer between users and database.
 - b. Application Development tools
 - Sometimes built-in graphical and analytical tools do not satisfy the analytical needs of an organization.
 - In such cases, custom reports are developed using Application development tools.
 - c. Data mining tools
 - Data mining is a process of discovering meaningful new correlation, patters, and trends by mining large amount data.
 - Data mining tools are used to make this process automatic.
 - d. OLAP tools
 - These tools are based on concepts of a multidimensional database.
 - It allows users to analyse the data using elaborate and complex multidimensional views.

5. Data warehouse Bus Architecture

- Data warehouse Bus determines the flow of data in your warehouse.

- The data flow in a data warehouse can be categorized as Inflow, Upflow, Downflow, Outflow and Meta flow.
- While designing a Data Bus, one needs to consider the shared dimensions, facts across data marts.

4.4 DATA WAREHOUSE MODEL

From the perspective of data warehouse architecture, we have the following data warehouse models –

- Virtual Warehouse
- Data mart
- Enterprise Warehouse

Virtual Warehouse

- The view over an operational data warehouse is known as a virtual warehouse.
- It is easy to build a virtual warehouse.
- Building a virtual warehouse requires excess capacity on operational database servers.

Data Mart

- Data mart contains a subset of organization-wide data.
- This subset of data is valuable to specific groups of an organization.
- In other words, we can claim that data marts contain data specific to a particular group.
- For example, the marketing data mart may contain data related to items, customers, and sales.
- Data marts are confined to subjects.
- Window-based or Unix/Linux-based servers are used to implement data marts.
- They are implemented on low-cost servers.
- The implementation data mart cycles is measured in short periods of time, i.e., in weeks rather than months or years.
- The life cycle of a data mart may be complex in the long run, if its planning and design are not organization-wide.
- Data marts are small in size.
- Data marts are customized by department.
- The source of a data mart is a departmentally structured data warehouse.
- Data mart are flexible.

Enterprise Warehouse

- An enterprise warehouse collects all the information and the subjects spanning an entire organization.
- It provides us enterprise-wide data integration.
- The data is integrated from operational systems and external information providers.

- This information can vary from a few gigabytes to hundreds of gigabytes, terabytes or beyond.

4.5 POPULATING BUSINESS

Methods for populating a data warehouse:

- A data mart or data warehouse that is based on those tables needs to reflect these changes.
- When moving data into a data warehouse, taking it from a source system is the first step in the ETL process.
- Once extracted from the source, the data can be cleaned and transformed so it can be loaded into a staging table or directly into the data warehouse.
- The source system for a data warehouse is typically an online transaction processing (OLTP) application, such as an ERP system, payroll application, order entry system, CRM, etc.
- Designing and creating the process to extract the data from the source system is usually the most time-consuming task in the ETL process if not the entire data warehousing process.
- Source systems are usually very complex, with tables and fields in the databases that are difficult to understand and poorly documented (many popular ERP systems use numbers for table names).
- This makes determining the data which needs to be extracted a challenge.
- And usually the data needs to be extracted on a daily basis to supply all changed data to the data warehouse in order to keep it up-to-date.
- Moreover, the source systems usually cannot be modified, or its performance or availability adjusted, to accommodate the needs of the data warehouse extraction process.

4.6 SUMMARY

- Data warehouse is an information system that contains historical and commutative data from single or multiple sources.
- These sources can be traditional Data Warehouse, Cloud Data Warehouse or Virtual Data Warehouse.
- A data warehouse is subject oriented as it offers information regarding the subject instead of organization's ongoing operations.
- In Data Warehouse, integration means the establishment of a common unit of measure for all similar data from the different databases
- Data warehouse is also non-volatile means the previous data is not erased when new data is entered in it.
- A Datawarehouse is Time-variant as the data in a DW has a high shelf life.
- There are mainly 5 components of Data Warehouse Architecture:

- 1) Database
 - 2) ETL Tools
 - 3) Meta Data
 - 4) Query Tools
 - 5) DataMarts
- These are four main categories of query tools
 1. Query and reporting, tools
 2. Application Development tools,
 3. Data mining tools
 4. OLAP tools
 - The data sourcing, transformation, and migration tools are used for performing all the conversions and summarizations.
 - In the Data Warehouse Architecture, meta-data plays an important role as it specifies the source, usage, values, and features of data warehouse data.

4.7 EXERCISE

Answer the following:

1. What is a data warehouse?
2. How does a data warehouse differ from a database?
3. What is an ODS?
4. Differentiate among a data mart, an ODS, and an EDW.
5. Explain the importance of metadata.
6. Describe the data warehousing process.
7. Describe the major components of a data warehouse.
8. Identify and discuss the role of middleware tools.

DATA WAREHOUSE ETL

Unit Structure

- 5.0 Objectives:
- 5.1 Introduction to Data Warehouse
- 5.2 Evolution of Data Warehouse
- 5.3 Benefits of Data Warehouse
- 5.4 Data Warehouse Architecture
 - 5.4.1 Basic Single-Tier Architecture
 - 5.4.2 Two-Tier Architecture
 - 5.4.3 Three-Tier Architecture
- 5.5 Properties of Data Warehouse Architectures
- 5.6 ETL Process in Data Warehouse
- 5.7 Cloud-based ETL Tools vs. Open Source ETL Tools
- 5.8 ETL and OLAP Data Warehouses
 - 5.8.1 The Technical Aspects of ETL
- 5.9 Data Warehouse Design Approaches
 - 5.9.1 Bill Inmon – Top-down Data Warehouse Design Approach
 - 5.9.2 Ralph Kimball – Bottom-up Data Warehouse Design Approach
- 5.10 Summary
- 5.11 References for further reading

5.0 OBJECTIVES

This chapter will make the readers understand the following concepts:

- Meaning of data warehouse
- Concept behind Data Warehouse
- History and Evolution of Data Warehouse
- Different types of Data Warehouse Architectures
- Properties of data warehouse
- Concept of Data Staging
- ETL process
- Design approaches to Data Warehouse

5.1 INTRODUCTION TO DATA WAREHOUSE

As organizations grow, they usually have multiple data sources that store different kinds of information. However, for reporting purposes, the organization needs to have a single view of the data from these different

sources. This is where the role of a Data Warehouse comes in. **A Data Warehouse helps to connect and analyse data that is stored in various heterogeneous sources.** The process by which this data is collected, processed, loaded, and analysed to derive business insights is called Data Warehousing.

The data that is present within various sources in the organization can provide meaningful insights to the business users if analysed in a proper way and can assist in making data as a strategic tool leading to improvement of processes. Most of the databases that are attached to the sources systems are transactional in nature. This means that these databases are used typically for storing transactional data and running operational reports on it. The data is not organized in a way where it can provide strategic insights. A data warehouse is designed for generating insights from the data and hence, helps to convert data into meaningful information that can make a difference.

Data from various operational source systems is loaded onto the Data Warehouse and is therefore a central repository of data from various sources that can provide cross functional intelligence based on historic data. Since the Data Warehouse is separated from the operational databases, it removes the dependency of working with transactional data for intelligent business decisions.

While the primary function of the Data Warehouse is to store data for running intelligent analytics on the same, it can also be used as a central repository where historic data from various sources is stored.

In order to be able to provide actionable intelligence to the end users, it is important can the Data Warehouse consists of information from different sources that can be analysed as one for deriving business intelligence for the organization as a whole. For example, in case of an insurance company, to be able to find out the customers who have more propensity provide a fraud claim, the insurance company must be able to analyse data from the various sources like the policy system, claims systems, CRM systems, etc.

In most cases, the data is these disparate systems is stored in different ways and hence cannot be taken as it is and loaded onto the data warehouse. Also, the purpose for which a data warehouse is built is different from the one for which the source system was built. In the case of our insurance company above, the policy system was built to store information with regards to the policies that are held by a customer. The CRM system would have been designed to store the customer information and the claims system was built to store information related to all the claims made by the customers over the years. For use to be able to determine which customers could potentially provide fraud claims, we need to be able to cross reference information from all these source systems and then make intelligent decisions based on the historic data.

Hence, the data has to come from various sources and has to be stored in a way that makes it easy for the organization to run business intelligence tools over it. There is a specific process to extract data from various source systems, translate this into the format that can be uploaded onto the data warehouse and the load the data on the data warehouse. This process for

extraction, translation and loading of data is explained in detail subsequently in the chapter.

Besides the process of ensuring availability of the data in the right format on the data warehouse, it is also important to have the right business intelligence tools in place to be able to mine data and then make intelligent predictions based on this data. This is done with the help of business intelligence and data visualization tools that enable converting data into meaningful information and then display this information in a way that is easy for the end users to understand.

With the improvement in technology and the advent of new tools, an enormous amount of data is being collected from various sources. This could be data collected from social media sites where every click of the user is recorded for further analysis. Such enormous amount of data creates a big data situation that is even more complex to store and analyse. Specialised tools are required to analyse such amounts of data.

The kind of analysis that is done on the data can vary from high level aggregated dashboards that provide a cockpit view to a more detailed analysis that can provide as much drill down of information as possible. Hence, it is important to ensure that the design of the data warehouse takes into consideration the various uses of the data and the amount of granularity that is needed for making business decisions.

Most times, the kind of analysis that is done using the data that is stored in the data warehouse is time-related. This could mean trends around sales numbers, inventory holding, profit from products or specific segments of customers, etc. These trends can then be utilized to forecast the future with the use of predictive tools and algorithms. The Data Warehouse provides the basic infrastructure and data that is needed by such tools to be able to help the end-users in their quest for information.

In order to understand Data Warehouse and the related concepts in more details, it is important for us to understand a few more related terms:

1. An Operational Data Store (ODS)
2. Data Marts
3. Data Lakes

Operational Data Store (ODS)

As the name suggests, an Operational Data Store or ODS is primarily meant to store data that near current operational data from various systems. The advantage of such a data store is that it allows querying of data which is more real-time as compared to a data warehouse. However, the disadvantage is that the data cannot be used to do complex and more time-consuming queries that can be run on a data warehouse. This is because the data on the operational data store has not yet gone through the process of transformation and is not structured for the purpose of complex queries. It provides a way to query data without having to burden the actual transactional system.

Data Marts

Data marts are like a mini data warehouse consisting of data that is more homogenous in nature rather than a varied and heterogeneous nature of a data warehouse. Data marts are typically built for the use within an

department or business unit level rather than at the overall organizational level. It could aggregate data from various systems within the same department or business unit. Hence, data marts are typically smaller in size than data warehouses.

Data Lakes

A concept that has emerged more recently is the concept of data lakes that store data in a raw format as opposed to a more structured format in the case of a data warehouse. Typically, a data lake will not need much transformation of data without loading onto the data lake. It is generally used to store bulk data like social media feeds, clicks, etc. One of the reasons as to why the data is not usually transformed before loading onto a data lake is because it is not usually known what kind of analysis would be carried out on the data. More often than not, a data scientist would be required to make sense of the data and to derive meaningful information by applying various models on the data.

Table 1 - Data Mart v/s Data Lake v/s Data Warehouse

Data Store	Primary Use	Amount of Data and Cost of Setup
<i>Data Mart</i>	Meant for use within a business unit or function	Lesser than Data Warehouse and Data Lake
<i>Data Warehouse</i>	Meant for use at organizational level across business units	More than Data Mart but less than Data Lake
<i>Data Lake</i>	Meant for advanced and predictive analytics	Greater than Data Mart and Data Warehouse

Some of the other names of a Data Warehouse system are Decision Support System, Management Information System, Business Intelligence System or Executive Information System.

5.2 EVOLUTION OF DATA WAREHOUSE

As the information systems within the organizations grew more and more complex and evolved over time, the systems started to develop and handle more and more amount of information. The need for an ability to analyze the data coming out from the various systems became more evident over time.

The initial concept of a “Business Data Warehouse” was developed by IBM researchers Barry Devlin and Paul Murphy in late 1980s. It was intended to provide an architectural model as to how the data would flow from an operational system to an environment that could support decision making for the business. The evolution of Data Warehouse can be traced back to the 1960 when Dartmouth and General Mills developed the terms like

dimension and facts in a joint research paper. In 1970, A. Nielsen and IRI used this concept to introduce the dimensional data marts for retail sales. It was much later in the year 1983, that Tera Data Corporation introduced a Database Management System that was designed specifically for the decision support process.

Later, in the late 1980s, IBM researchers developed the Business Data Warehouse. Inmon Bill was considered as a father of data warehouse. He had written about a variety of topics for building, usage, and maintenance of the warehouse & the Corporate Information Factory.

5.3 BENEFITS OF DATA WAREHOUSE

There are numerous benefits that a data warehouse can provide organizations. Some of these benefits are listed below:

1. Enhancing business intelligence within the organization:

A data warehouse is able to bring data from various source systems into a single platform. This allows the users to make better business decisions that are based on data which cuts across different system and can provide an integrated view rather than an isolated view of the data.

This is made possible since the data has been extracted, translated, and then loaded onto the data warehouse platform from various cross-functional and cross-departmental source systems. Information that provides such an integrated view of the data is extremely useful for the senior management in making decisions at the organizational level.

2. Right information at the right time

Given the ability of the Data Warehouse to be able to provide information requested on demand, it is able to provide the right information to the organizational users at the time when it is required the most. Time is usually of essence when it comes to business decisions. Organizations not only need to spend valuable time and effort in collating information from various sources. Manual collation of such data not only takes time but is also error prone and cannot be completely trusted.

A Data warehouse platform can take care of all such issues since the data is already loaded and can be queried upon as desired. Thereby saving precious time and effort for the organizational users.

3. Improving the quality of data

A data warehouse platform consists of data that is extracted from various systems and has been translated to the required format for the data warehouse. Second disk significantly improves the quality of the data second thereby increases the quality of decisions that are made based on such data.

Given that the data in the data warehouse is usually automatically uploaded, the chances of errors to creep into the process are quite minimal. This is not a manual process which is prone to errors.

4. **Return on investment**

Building a data warehouse is usually an upfront cost for the organization. However, the return that it provides in terms of information and the ability to make right decisions at the right time provides a return on investment that is usually manyfold with respect to the amount that has been invested upfront. In the long run, a data warehouse helps the organization in multiple ways to generate new revenue and save costs.

5. **Competitive edge**

A data warehouse is able to provide the top management within the organization a capability to make business decisions that are based on data that cuts across the organizational silos. It is therefore more reliable and the decisions that are based on such data are able to provide a competitive edge to the organization viz-a-viz their competition

6. **Better decision-making process**

Use of a data warehouse that provides the capability to integrate information from various systems across the organization can lead to better decision-making process within the organization. The senior management will have an integrated view of information coming from video source systems and therefore will be able to make decisions that are not limited by a siloed view.

7. **Predict with more confidence**

The data that is stored within the data warehouse provides better quality and consistency than any manual process. This can give more confidence to the users that any predictions that are driven from this data warehouse would be more accurate and can be trusted with more confidence than manual processes.

8. **Streamlined data flow within the organization**

A data warehouse is able to integrate data from multiple sources within the organization and therefore streamlines and provides a consistent view of data that is stored in various systems – bringing them into a single repository.

5.4 DATA WAREHOUSE ARCHITECTURE

As we seek to understand what the data warehouse is, it is important for us to understand the different types of deployment architectures by way of which a data warehouse can be implemented within an organization. Every data warehouse implementation is different from each other. However, there are certain elements that can be common between all of them.

The data warehouse architecture defines the way in which information is processed, transformed, loaded and then presented to the end users for the purpose of generating business insights. In order to understand the data

warehouse architecture, we need to understand the some of the terminologies associated with it.

Day-to-day operations of an organization are typically run by production systems such as payroll, HR, finance, etc. that generate data and transactions on a daily basis and are usually called Online Transaction Processing (OLTP) systems. Such applications are usually the sources of data for a data warehousing platform. On the other hand, a data warehouse is primarily designed to support analytical capabilities on top of data that comes from all of these various source systems and is therefore termed as an Online Analytical Processing (OLAP) system. The online analytical processing system provides users with the capability to produce ad hoc reports as required and on demand.

As can be seen that the online transaction processing systems are usually updated regularly based on the data and transactions that happen daily on that system. In contrast, an online analytical processing system or the data warehouse is usually updated through an ETL process that extracts the data from the source systems on a regular basis, transforms the data into a format that will be required for the data warehouse and then loads the data onto the data warehouse as per the pre-defined processes.

It may be noticed that the data in the data warehouse is typically not real time data and there is usually a delay in moving the data from these source systems to the data warehouse. However, this is something that most businesses are fine with as long as they get an integrated view of data from across different functions of the organization and as long as the data is automatically uploaded on the data warehouse for generation of these insights on demand.

A data warehouse architecture may be implemented in many different ways. Some of the common ways of implementing the data warehouse architecture are listed below.

- Basic architecture for a Data Warehouse or a single tier architecture
- Staging-area based architecture for a Data Warehouse or a two tier architecture
- Staging area and data-mart based architecture for a Data Warehouse or a three-tier architecture



Figure 1 - Various Implementation Architectures for Data Warehouse

5.4.1 Basic Single-Tier Architecture

This type of architecture is not used much but it does provide a good idea of how basic data warehouse can be implemented. It aims to remove data redundancy. In this basic architecture, the only physical layer available is the source systems.

This means that the data warehouse is implemented as a multidimensional view of operational data created by specific middleware, or an intermediate processing layer.

The figure below shows the implementation of a basic data warehouse architecture which has the sources systems abstracted by a middleware that aims to remove the provide a separation between transaction and analytical capabilities.

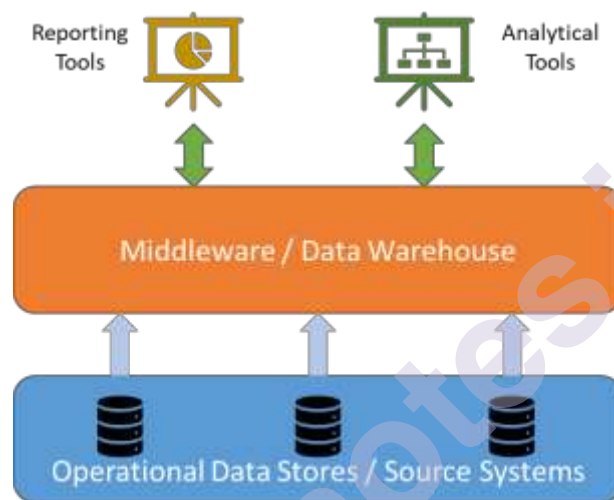


Figure 2 - Basic Data Warehouse Architecture

5.4.2 Two-Tier Architecture

The need for separation plays a crucial role in defining the two-tier architecture for a data warehouse system, as shown in the figure below:

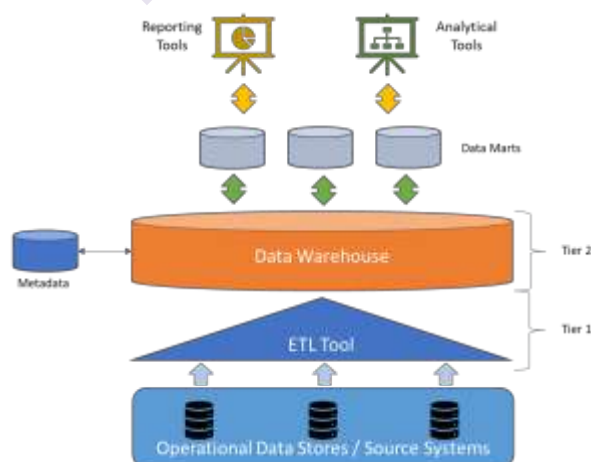


Figure 3 - Two Tier architecture for Data Warehouse

The two the two-layer architectures highlights a separation between physically available resources and data warehouse thus it is divided into four different four different stages which are according to the dataflow. these different stages are mentioned as below.

1. **Source Layer:** as discussed earlier the data warehouse uses heterogeneous sources of data the data which is initially stored in a corporate relational databases or legacy databases or it may come from any source within the organization or outside the organization.
2. **Data Staging:** The data which we are going to store should be extracted, cleans to remove any inconsistencies integrity to merge heterogeneous sources into one standard schema. Thus extraction, transformation, loading tools ETL can combine heterogeneous schema by extracting, cleaning, transforming, validating and load data into data warehouse.
3. **Data Warehouse layer:** Information is saved to one logically centralized individual repository: a data warehouse. The data warehouses can be directly accessed, but it can also be used as a source for creating data marts, which partially replicate data warehouse contents and are designed for specific enterprise departments. Meta-data repositories store information on sources, access procedures, data staging, users, data mart schema, and so on.
4. **Analysis:** In this layer, integrated data is efficiently, and flexible accessed to issue reports, analyse information, and simulate business scenarios. It should feature information navigators, complex query optimizers, and customer-friendly GUIs.

5.4.3 Three-Tier Architecture

The three-tier architecture consists of the source layer (containing multiple source system), the reconciled layer and the data warehouse layer (containing both data warehouses and data marts).

The reconciled layer is between the source data and data warehouse. It creates a standard reference model for the whole enterprise. And, at the same time it separates the problem of data extraction and integration from data warehouse. This layer is also directly used to perform better operational tasks e.g. producing daily reports or generating data flows periodically to benefit from cleaning and integration.

While this architecture is useful for extensive global enterprise systems, the major disadvantage is the use of extra file storage space because of redundant reconciled layer that makes the analytical tools little further away from being real time.

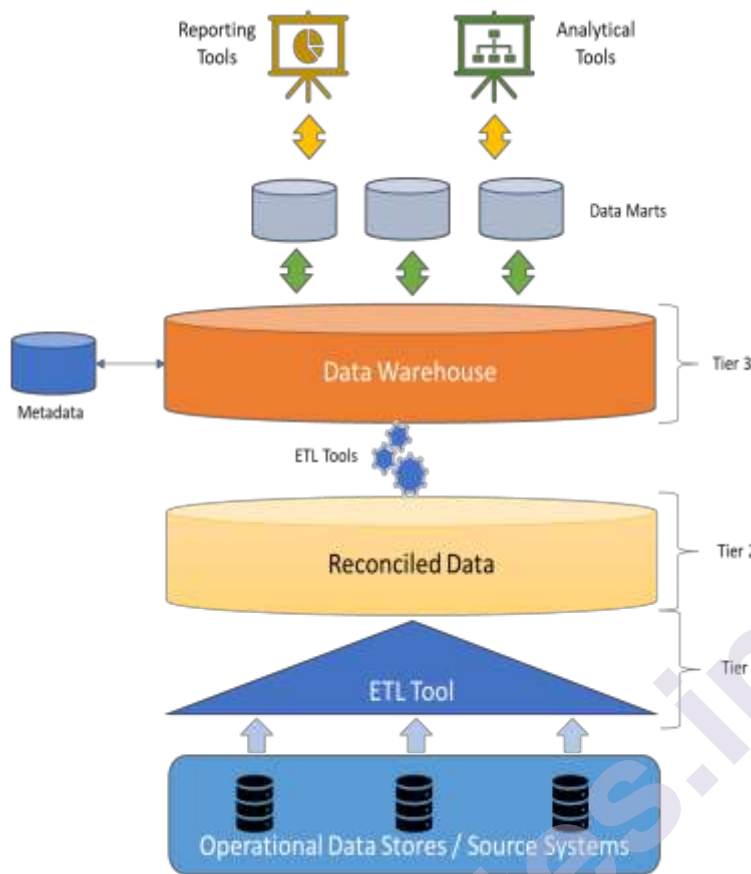


Figure 4 - Three Tier architecture for Data Warehouse

5.5 PROPERTIES OF DATA WAREHOUSE ARCHITECTURES

The following architecture properties are necessary for a data warehouse system:

1. **Separation:** There should be separation between analytical and transactional processing as much as possible.
2. **Scalability:** To upgrade the data volumes which has to be managed and processed and number of user requirements which have to be met we need hardware and software architectures that should be simple to upgrade.
3. **Extensibility:** The architecture should be able to perform new operations and technologies without redesigning the whole system.
4. **Security:** Security plays a very important role in information technology .Monitoring accesses providing passwords are necessary because of the strategic data stored in the data warehouses.
5. **Administrability:** Data Warehouse management should not be complicated.



Figure 5 – Properties of Data Warehouse Architecture

5.6 ETL Process in Data Warehouse

ETL (or Extract, Transform, Load) is a process of data integration that encompasses three steps — extraction, transformation, and loading. In a nutshell, ETL systems take large volumes of raw data from multiple sources, converts it for analysis, and loads that data into your warehouse.

It is a process in which an ETL tool extracts the data from various data source systems, transforms it in the staging area and then finally, loads it into the Data Warehouse system.

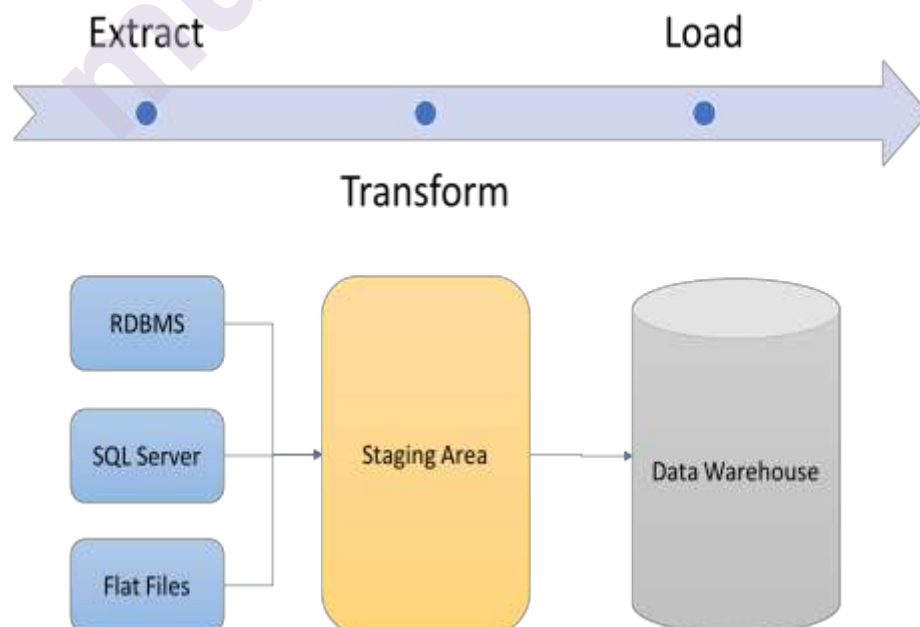


Figure 6 - High Level ETL Process flow

Extraction:

In this step data is extracted from various sources into a staging area. This area acts as a buffer between the data warehouse and source systems. As we know the data comes from various sources, hence the data will be in different formats and we cannot directly transfer this data into data warehouse. The staging area is used by companies for data cleaning.

A major challenge during this extraction process is how ETL tool differentiates structured and unstructured data. All unstructured items such as emails web pages etc can be difficult to extract without the right tool. It is important to extract the data from various source systems and store it into staging area first and not indirectly into data warehouse because of their various formats. It is, therefore, one of the major steps of ETL process.

Transformation:

The second step of the ETL process is transformation. In this step, a set of rules or functions are applied on the extracted data to convert it into a single standard format. All the data from multiple source systems is normalized and converted to a single system format — improving data quality and compliance. ETL yields transformed data through these methods:

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

Loading:

The third and final step of the ETL process is loading. In this step, the transformed data is finally loaded into the data warehouse. Sometimes the data is updated by loading into the data warehouse very frequently and sometimes it is done after longer but regular intervals. The rate and period of loading solely depends on the requirements and varies from system to system.

ETL process can also use the pipelining concept i.e. as soon as some data is extracted, it can be transformed and during that period some new data can be extracted. And while the transformed data is being loaded into the data warehouse, the already extracted data can be transformed.

Finally, data that has been extracted to a staging area and transformed is loaded into your data warehouse. Depending upon your business needs, data can be loaded in batches or all at once. The exact nature of the loading will depend upon the data source, ETL tools, and various other factors.

The block diagram of the pipelining of ETL process is shown below:

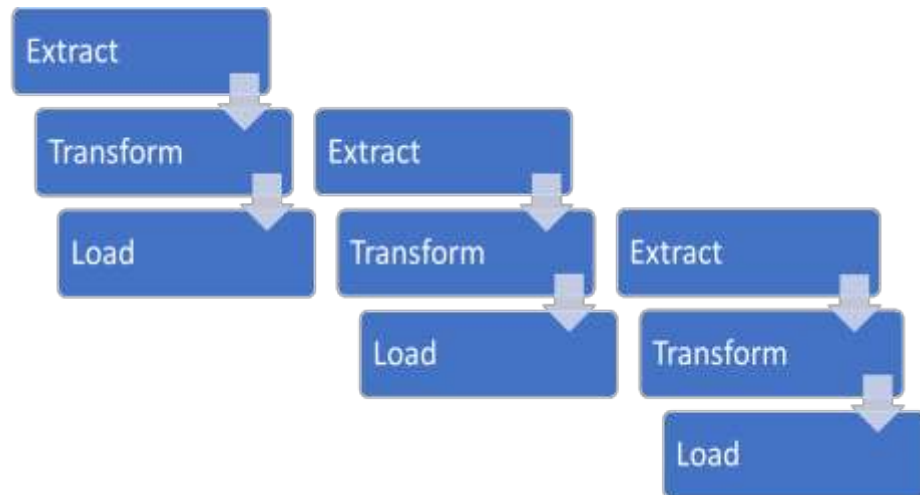


Figure 7 - ETL Pipeline View

ETL Tools: Most used ETL tools are

- Sybase
- Oracle Warehouse builder,
- CloverETL
- MarkLogic.

5.7 CLOUD-BASED ETL TOOLS VS. OPEN SOURCE ETL TOOLS

ETL is a critical component of overall data warehouse architecture so choosing the right one is very crucial there are various different options available and one can choose depending upon the overall ETL needs, data schemas and operational structure

Cloud-based ETL tools like Xplenty offer rapid, real-time streaming, quick integrations, and easy pipeline creation. The primary benefit of cloud-based ETL tools is that they work immediately out-of-the-box. Plus, they're hyper-useful for a variety of ETL needs, especially if most of your warehouse exists in the cloud (i.e., Redshift, Snowflake, or Big Query).

Open source ETL tools come in a variety of shapes and sizes. There are ETL frameworks and libraries that you can use to build ETL pipelines in Python. There are tools and frameworks you can leverage for GO and Hadoop. Really, there is an open-source ETL tool out there for almost any unique ETL need.

5.8 ETL AND OLAP DATA WAREHOUSES

Data engineers have been using ETL for over two decades to integrate diverse types of data into online analytical processing (OLAP) data warehouses. The reason for doing this is simple: to make data analysis easier.

Normally, business applications use online transactional processing (OLTP) database systems. These are optimized for writing, updating, and

editing the information inside them. They're not good at reading and analysis. However, online analytical processing database systems are excellent at high-speed reading and analysis. That's why ETL is necessary to transform OLTP information, so it can work with an OLAP data warehouse.

During the ETL process, information is:

- Extracted from various relational database systems (OLTP or RDBMS) and other sources.
- Transformed within a staging area, into a compatible relational format, and integrated with other data sources.
- Loaded into the online analytical processing (OLAP) data warehouse server.

5.8.1 The Technical Aspects of ETL

It's important to pay close attention to the following when designing your ETL and ELT processes:

- **Ensure accurate logging:** It's vital to make sure your data system provides "accurate logging" of new information. To ensure accurate logging, you'll need to audit data after loading to check for lost or corrupt files. With proper auditing procedures, you can debug your ETL/ELT process when data integrity challenges arise (as they invariably do).
- **Flexibility to work with diverse sources of structured and unstructured data:** Your data warehouse may need to integrate information from a lot of incompatible sources like PostgreSQL, Salesforce, Cassandra, and in-house financial applications. Some of this information could lack the data structures required for analysis. You need to design your ETL/ELT process to deal with all forms of data—structured and unstructured alike.
- **Stability and reliability:** ETL/ELT pipelines get overloaded, crash, and run into problems. Your goal should be to build a fault-tolerant system that can recover after a shutdown so your data can move without getting lost or corrupted even in the face of unexpected issues.
- **Designing an alert system:** To ensure the accuracy of your business insights, an alert system that notifies you of potential problems with the ETL/ELT process is essential. For example, you'll want to receive notifications and reports for expired API credentials, bugs related to third-party APIs, connector errors, general database errors, and more.
- **Strategies to speed up the flow of data:** When data warehouses and BI platforms have access to information that is up-to-date, they offer better, more accurate insights at a moment's notice. Therefore, it's important to focus on reducing data latency, i.e., the time it takes for a data packet to move from one area of the system to the next.

- **Growth flexibility:** Your ETL/ELT solution should be flexible to scale up and down according to your organization's changing data needs. This will save money on cloud-server processing and storage fees, while providing the ability to scale up as required.
- **Support for incremental loading:** Using change data capture (CDC) speeds up the ETL process by permitting incremental loading. This lets you update only a small part of your data warehouse while ensuring data synchronicity.

5.9 DATA WAREHOUSE DESIGN APPROACHES

Very important aspect of building data warehouses is the design of data warehouse. Selection of right data Warehouse saves lot of time, efforts, and project cost.

The two different approaches are normally followed when designing a data warehouse solution and based on the requirement of the project we can choose one that suits the particular scenario.

These methodologies are a result of research from Bill Inmon and Ralph Kimball.

5.9.1 Bill Inmon – Top-down Data Warehouse Design Approach

“Bill Inmon” is sometimes also referred to as the “father of data warehousing”; his design methodology is based on a top-down approach. In the top-down approach, the data warehouse is designed first and then data mart are built on top of data warehouse.

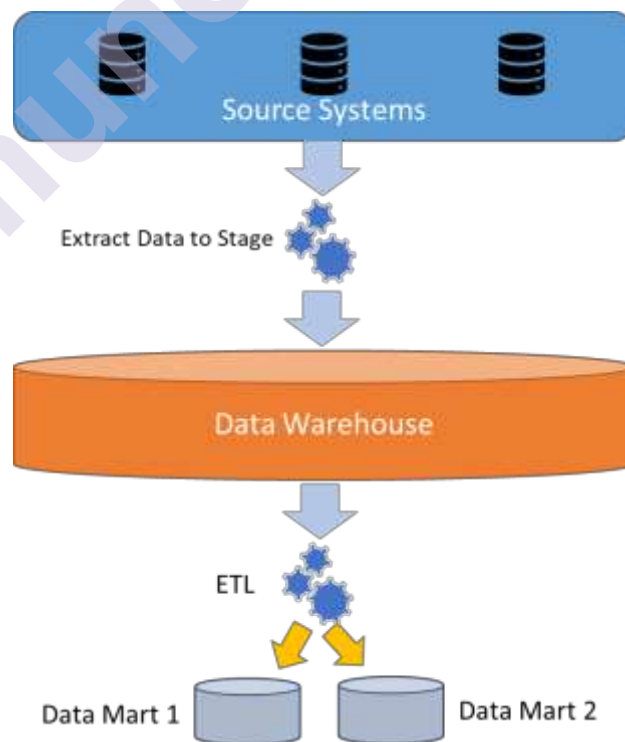


Figure 8 - Top Down Approach

Below are the steps that are involved in top-down approach:

- Data is extracted from the various source systems. The extracts are loaded and validated in the stage area. Validation is required to make sure the extracted data is accurate and correct. You can use the ETL tools or approach to extract and push to the data warehouse.
- Data is extracted from the data warehouse in regular basis in stage area. At this step, you will apply various aggregation, summarization techniques on extracted data and loaded back to the data warehouse.
- Once the aggregation and summarization is completed, various data marts extract that data and apply the some more transformation to make the data structure as defined by the data marts.

5.9.2 Ralph Kimball – Bottom-up Data Warehouse Design Approach

Ralph Kimball is a renowned author on the subject of data warehousing. His data warehouse design approach is called dimensional modelling or the Kimball methodology. This methodology follows the bottom-up approach. As per this method, data marts are first created to provide the reporting and analytics capability for specific business process, later with these data marts enterprise data warehouse is created.

Basically, Kimball model reverses the Inmon model i.e. Data marts are directly loaded with the data from the source systems and then ETL process is used to load in to Data Warehouse.

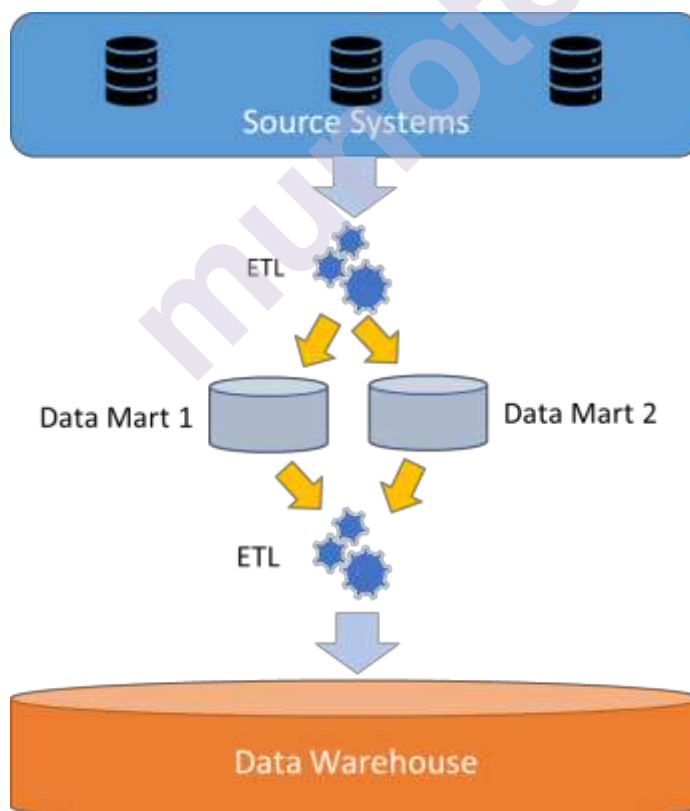


Figure 9 - Bottom up Approach

Below are the steps that are involved in bottom-up approach:

- The data flow in the bottom-up approach starts from extraction of data from various source system into the stage area where it is processed and loaded into the data marts that are handling specific business process.
- After data marts are refreshed the current data is once again extracted in stage area and transformations are applied to create data into the data mart structure. The data is the extracted from Data Mart to the staging area is aggregated, summarized and so on loaded into EDW and then made available for the end user for analysis and enables critical business decisions.

5.10 SUMMARY

- A data warehouse is a data base which is kept separate from organizations operational database.
- It possesses consolidated historical data which helps organizations to analyse its business
- Data warehouse helps in consolidated historical data analysis
- An operational database query allows to read and modify operations while an OLAP query needs read only access of stored data
- An operational database second maintains current data while on the other hand a data warehouse maintains historical data
- OLAP systems are used by knowledge workers such as executives, managers, and analysts
- ETL stands for extract, transform and load
- ETL provides a method of moving the data from various sources into data warehouse
- In the first step, extraction, data is extracted from the source system into the staging area.
- In the transformation step, the data extracted from source is cleaned and transformed.
- In the third step, loading, data into the target data warehouse

5.11 REFERENCES FOR FURTHER READING

Reference books:

1. Ponniah, Paulraj, Data warehousing fundamentals: a comprehensive guide for IT professionals, John Wiley & Sons, 2004.
2. Dunham, Margaret H, Data mining: Introductory and advanced topics, Pearson Education India, 2006.
3. Gupta, Gopal K, Introduction to data mining with case studies, PHI Learning Pvt. Ltd., 2014.

4. Han, Jiawei, Jian Pei, and Micheline Kamber, Data mining: concepts and techniques, Second Edition, Elsevier, Morgan Kaufmann, 2011.
5. Ramakrishnan, Raghu, Johannes Gehrke, and Johannes Gehrke, Database management systems, Vol. 3, McGraw-Hill, 2003
6. Elmasri, Ramez, and Shamkant B. Navathe, Fundamentals of Database Systems, Pearson Education, 2008, (2015)
7. Silberschatz, Abraham, Henry F. Korth, and Shashank Sudarshan, Database system concepts, Vol. 5, McGraw-Hill, 1997.

Web References:

1. <https://www.guru99.com/data-mining-vs-datawarehouse.html>
2. https://www.tutorialspoint.com/dwh/dwh_overview
3. <https://www.geeksforgeeks.org/>
4. <https://blog.eduonix.com/internet-of-things/web-mining-text-mining-depth-mining-guide>

munotes.in

DESIGNING BUSINESS DATA WAREHOUSE

Unit Structure

- 6.0 Objectives
- 6.1 Introduction
- 6.2 OLTP and OLAP SYSTEMS
- 6.3 Designing business information warehouse
 - 6.3.1 Need of Data Warehouse
 - 6.3.2 Approaches to Build Data Warehouse
 - 6.3.2 Benefits of Data Warehouse
- 6.4 Principals of dimensional modelling
 - 6.4.1 Elements of Dimensional Data Model
 - 6.4.2 Steps of Dimensional Modelling
 - 6.4.3 Advantages of Dimensional Modelling
- 6.5 Data cube operations
 - 6.5.1 OLAP Operations
- 6.6 Data cube schemas
 - 6.6.1 Star Schema
 - 6.6.2 Snowflake Schema
- 6.7 References
- 6.8 Summary

6.0 OBJECTIVES

After going through the unit, the learner will be able to:

- Gain the basic knowledge about Online Transaction Processing (OLTP).
- Impart understanding of Online Analytical Processing (OLAP).
- Elaborate the key differences between OLTP and OLAP.
- Design business information warehouse.
- Gain the knowledge about Dimensional modelling.
- Understand different Data cube schemes and operations.

6.1 INTRODUCTION

OLTP was formerly restricted to real-world exchanges in which something was exchanged—money, goods, information, service requests, and so on. However, the definition of transaction in this sense has evolved over time, particularly since the internet's introduction, to include any type of digital

connection or engagement with a business that can be initiated from anywhere in the world and via any web-connected sensor. It also includes any type of interaction or action that a business must record in order to better serve their consumers, such as downloading pdfs from a web page, watching a certain movie, or automatic maintenance triggers or comments on social media. Businesses usually have two types of data processing capabilities: OLTP and OLAP.

A data warehouse is a single data repository that integrates records from several data sources for online business analytical processing (OLAP). This means that a data warehouse must suit the needs of all business stages across the whole organization. As a result, data warehouse design is a very complex, time-consuming, and error-prone procedure. Furthermore, business analytical functions evolve with time, resulting in shifts in system requirements. As a result, data warehouse and OLAP systems are dynamic, and design is ongoing. Many OLAP systems employ a dimensional model as their data model.

OLAP databases are divided into one or more cubes. The cubes are designed in such a way that creating and viewing reports become easy. Analysts frequently need to group, aggregate and join data. These OLAP operations in data mining are resource intensive. With OLAP data can be pre-calculated and pre-aggregated, making analysis faster. A schema is a logical description of a database's contents. It contains the name and description of all record kinds, as well as all associated data items and aggregates. A data warehouse, like a database, requires the maintenance of a schema. The relational model is used in databases, while the Star, Snowflake, and Fact Constellation structure are used in data warehouses. The schemas used in a data warehouse will be discussed in this chapter.

6.2 OLTP AND OLAP SYSTEMS

We can divide IT systems as transactional (OLTP) and analytical (OLAP). In general, we can assume that OLTP systems provide source data to data warehouses, whereas OLAP systems help to analyze it.

OLTP (On-line Transaction Processing) is characterized by a large number of short on-line transactions (INSERT, UPDATE, DELETE). The main emphasis for OLTP systems is put on very fast query processing, maintaining data integrity in multi-access environments and an effectiveness measured by number of transactions per second. In OLTP database there is detailed and current data, and schema used to store transactional databases is the entity model (usually 3NF).

OLAP (On-line Analytical Processing) is characterized by relatively low volume of transactions. Queries are often very complex and involve aggregations. For OLAP systems a response time is an effectiveness measure. OLAP applications are widely used by Data Mining techniques. In OLAP database there is aggregated, historical data, stored in multi-dimensional schemas (usually star schema). For example, a bank storing years of historical records of check deposits could use an OLAP database

to provide reporting to business users. OLAP databases are divided into one or more cubes. The cubes are designed in such a way that creating and viewing reports become easy. At the core of the OLAP concept, is an OLAP Cube. The OLAP cube is a data structure optimized for very quick data analysis. The OLAP Cube consists of numeric facts called measures which are categorized by dimensions. OLAP Cube is also called the **hypercube**.

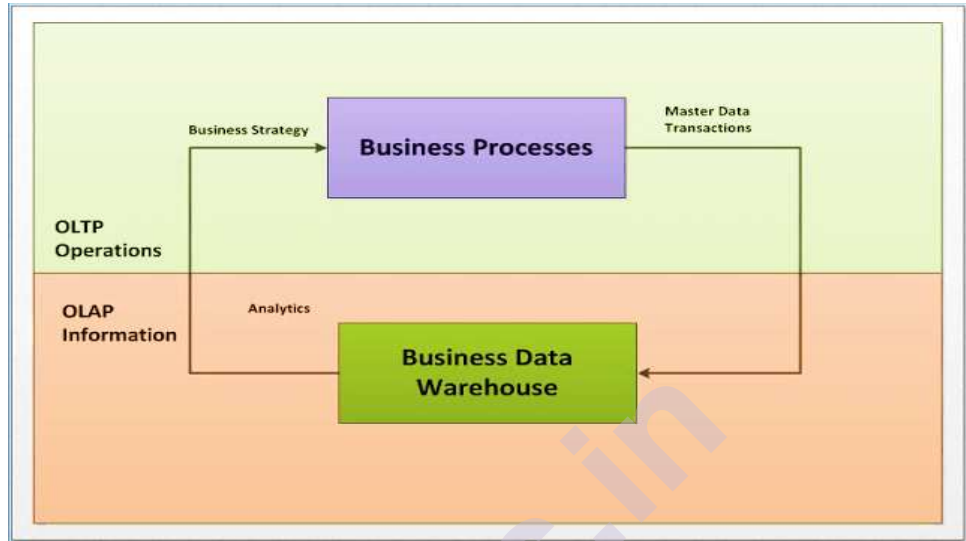


Fig. 6.1 OLTP Vs OLAP Operations

The following table summarizes the major differences between OLTP and OLAP system design.

Table 6.1: OLTP Vs OLAP

Parameters	OLTP	OLAP
Process	It is an online transactional system. It manages database modification.	OLAP is an online analysis and data retrieving process.
Characteristic	It is characterized by large numbers of short online transactions.	It is characterized by a large volume of data.
Functionality	OLTP is an online database modifying system.	OLAP is an online database query management system.
Method	OLTP uses traditional DBMS.	OLAP uses the data warehouse.
Query	Insert, Update, and Delete information from the database.	Mostly Select operations
Table	Tables in OLTP database are normalized.	Tables in OLAP database are not normalized.

Source	OLTP and its transactions are the sources of data.	Different OLTP databases become the source of data for OLAP.
Storage	The size of the data is relatively small as the historical data is archived. For e.g. MB, GB.	Large amount of data is stored typically in TB, PB.
Data Integrity	OLTP database must maintain data integrity constraint.	OLAP database does not get frequently modified. Hence, data integrity is not an issue.
Response time	It's response time is in millisecond.	Response time in seconds to minutes.
Data quality	The data in the OLTP database is always detailed and organized.	The data in OLAP process might not be organized.
Usefulness	It helps to control and run fundamental business tasks.	It helps with planning, problem-solving, and decision support.
Operation	Allow read/write operations.	Only read and rarely write.
Audience	It is a market orientated process.	It is a customer orientated process.
Query Type	Queries in this process are standardized and simple.	Complex queries involving aggregations.
Back-up	Complete backup of the data combined with incremental backups.	OLAP only need a backup from time to time. Backup is not important compared to OLTP
Design	DB design is application oriented. Example: Database design changes with industry like Retail, Airline, Banking, etc.	DB design is subject oriented. Example: Database design changes with subjects like sales, marketing, purchasing, etc.
User type	It is used by Data critical users like clerk, DBA & Data Base professionals.	Used by Data knowledge users like workers, managers, and CEO.
Purpose	Designed for real time business operations.	Designed for analysis of business measures by category and attributes.
Performance metric	Transaction throughput is the performance metric	Query throughput is the performance metric.

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

6.3 DESIGNING BUSINESS INFORMATION WAREHOUSE

Ralph Kimball created the notion of Dimensional Modelling, which is made up of facts and dimension tables. The SELECT OPERATION is optimized because the major purpose of this modelling is to improve data retrieval. The benefit of employing this approach is that we can store data in a data warehouse in a way that makes it easier to store and retrieve data.

6.3.1 Need of Data Warehouse

Data Warehouse is needed for the following reasons:

1. **Business User:** Business users require a data warehouse to view summarized data from the past. Since these people are non-technical, the data may be presented to them in an elementary form.
2. **Store historical data:** Data Warehouse is required to store the time variable data from the past. This input is made to be used for various purposes.
3. **Make strategic decisions:** Some strategies may be depending upon the data in the data warehouse. So, data warehouse contributes to making strategic decisions.

4. **For data consistency and quality:** Bringing the data from different sources at a commonplace, the user can effectively undertake to bring the uniformity and consistency in data.
5. **High response time:** Data warehouse has to be ready for somewhat unexpected loads and types of queries, which demands a significant degree of flexibility and quick response time.

3.3.2 Approaches to Build Data Warehouse

1. Top-Down Approach

- This is the big-picture approach to building the overall, massive, enterprise-wide data warehouse.
- There is no collection of information sources here.
- The data warehouse is large and well-integrated.
- This approach, on the other hand, would take longer to build and has a higher failure rate.
- This approach could be dangerous if you do not have experienced professionals on your team.
- It will also be difficult to sell this approach to senior management and sponsors.
- They are unlikely to see results soon enough.

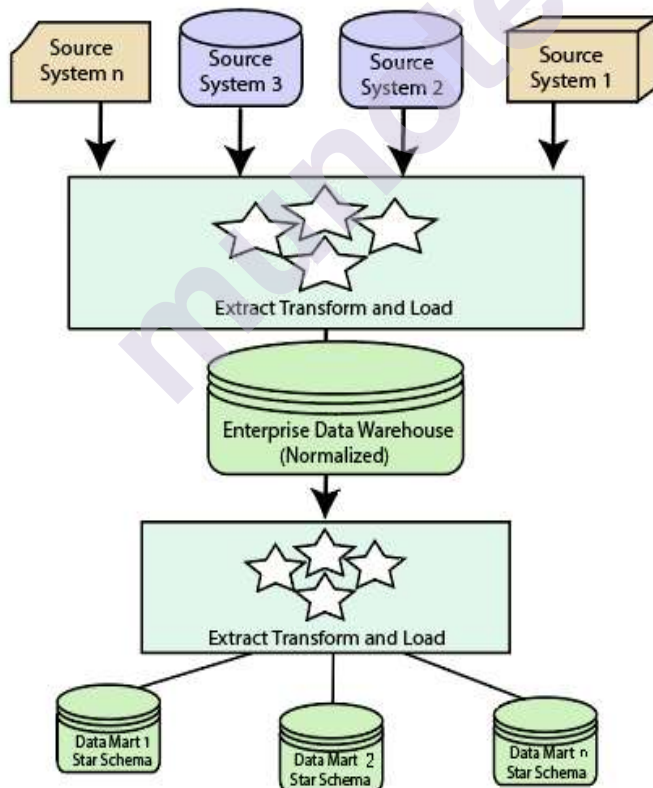


Fig. 6.2 Top-Down Approach of Data Warehouse Design

Advantages

- Represents a data view from the perspective of the enterprise.

- Inherently designed—not a mash-up of disparate data marts.
- Data about the content is stored in a single, central location.
- Centralized control and rules.

Disadvantages

- Even with an iterative strategy, building takes longer.
- High failure risk/exposure
- Requires a high level of cross-functional expertise
- Expenses are high without proof of concept.

2. Bottom-Up Approach

- You create departmental data marts one by one using this bottom-up method.
- To figure out which data marts to build first, you'd create a priority list.
- The most serious disadvantage of this method is data fragmentation.
- Each data mart will be blind to the organization's overarching requirements.

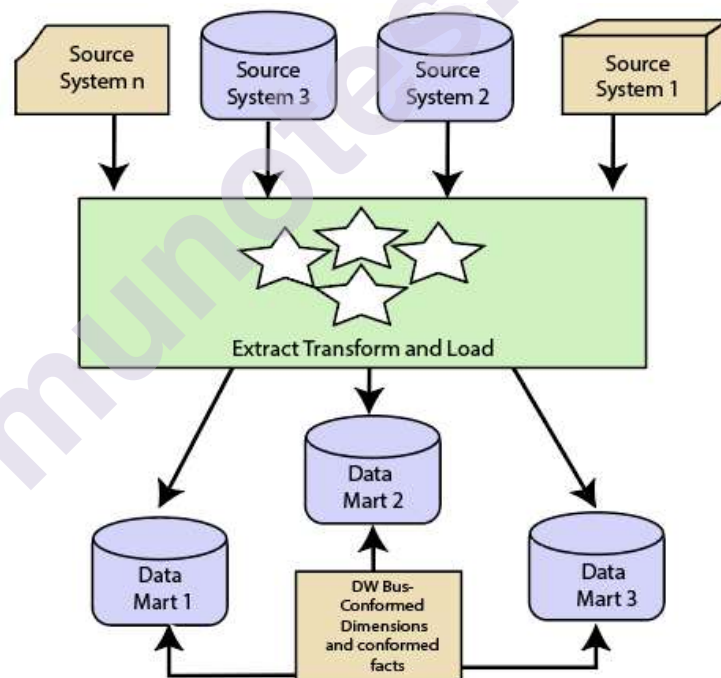


Fig. 6.3 Bottom-Up Approach of Data Warehouse Design

Advantages

- Implementation of small portions is faster and easier.
- Favourable return on investment and proof of concept.
- There is a lower chance of failure.
- Inherently incremental; significant data marts can be scheduled first.
- Allows the project team to grow and develop.

Disadvantages

- Each data mart has its own skewed perspective on information.
- Every data mart is flooded with redundant information.
- Perpetuates data that is inconsistent and irreconcilable.
- Increases the number of unmanageable interfaces.

6.3.2 Benefits of Data Warehouse

1. **Delivers enhanced business intelligence:** By having access to information from various sources from a single platform, decision makers will no longer need to rely on limited data or their instinct. Additionally, data warehouses can effortlessly be applied to a business's processes, for instance, market segmentation, sales, risk, inventory, and financial management.
2. **Saves times:** A data warehouse standardizes, preserves, and stores data from distinct sources, aiding the consolidation and integration of all the data. Since critical data is available to all users, it allows them to make informed decisions on key aspects. In addition, executives can query the data themselves with little to no IT support, saving more time and money.
3. **Enhances data quality and consistency:** A data warehouse converts data from multiple sources into a consistent format. Since the data from across the organization is standardized, each department will produce results that are consistent. This will lead to more accurate data, which will become the basis for solid decisions.
4. **Generates a high Return on Investment (ROI):** Companies experience higher revenues and cost savings than those that haven't invested in a data warehouse.
5. **Provides competitive advantage:** Data warehouses help get a holistic view of their current standing and evaluate opportunities and risks, thus providing companies with a competitive advantage.
6. **Improves the decision-making process:** Data warehousing provides better insights to decision makers by maintaining a cohesive database of current and historical data. By transforming data into purposeful information, decision makers can perform more functional, precise, and reliable analysis and create more useful reports with ease.
7. **Enables organizations to forecast with confidence:** Data professionals can analyze business data to make market forecasts, identify potential KPIs, and gauge predicated results, allowing key personnel to plan accordingly.
8. **Streamlines the flow of information:** Data warehousing facilitates the flow of information through a network connecting all related or non-related parties.

6.4 PRINCIPALS OF DIMENSIONAL MODELLING

- Dimensional Modelling (DM) is a data structure technique optimized for data storage in a Data warehouse. The purpose of dimensional modelling is to optimize the database for faster retrieval of data. The concept of Dimensional Modelling was developed by Ralph Kimball and consists of “fact” and “dimension” tables. Dimensional table records information on each dimension, and fact table records all the “fact”, or measures.
- A dimensional model in data warehouse is designed to read, summarize, analyse numeric information like values, balances, counts, weights, etc. in a data warehouse. In contrast, relation models are optimized for addition, updating and deletion of data in a real-time Online Transaction System.
- These dimensional and relational models have their unique way of data storage that has specific advantages.
- For instance, in the relational model, normalization and ER models reduce redundancy in data. On the contrary, dimensional model in data warehouse arranges data in such a way that it is easier to retrieve information and generate reports.
- Hence, Dimensional models are used in data warehouse systems and not a good fit for relational systems.

6.4.1 Elements of Dimensional Data Model

1. **Fact:** Facts are the measurements/metrics or facts from your business process. For a Sales business process, a measurement would be quarterly sales number.
2. **Dimension:** Dimension provides the context surrounding a business process event. In simple terms, they give who, what, where of a fact. In the Sales business process, for the fact quarterly sales number, dimensions would be:
 - Who – Customer Names
 - Where – Location
 - What – Product Name

In other words, a dimension is a window to view information in the facts.

3. **Attributes:** The Attributes are the various characteristics of the dimension in dimensional data modeling. In the Location dimension, the attributes can be
 - State
 - Country
 - Zipcode, etc.

Attributes are used to search, filter, or classify facts. Dimension Tables contain Attributes

4. **Fact Table:** A fact table is a primary table in dimension modelling. A Fact Table contains
 - Measurements/facts
 - Foreign key to dimension table
5. **Dimension Table**
 - A dimension table contains dimensions of a fact.
 - They are joined to fact table via a foreign key.
 - Dimension tables are de-normalized tables.
 - The dimension attributes are the various columns in a dimension table.
 - Dimensions offers descriptive characteristics of the facts with the help of their attributes.
 - No limit set for given for number of dimensions.
 - The dimension can also contain one or more hierarchical relationships

6.4.2 Steps of Dimensional Modelling

The accuracy in creating your dimensional modeling determines the success of your data warehouse implementation. The model should describe the Why, How much, When/Where/Who and What of your business process. Here are the steps to create dimension model.

1. Identify Business Process

- Identifying the actual business process, a data warehouse should cover. This could be Marketing, Sales, HR, etc. as per the data analysis needs of the organization. The selection of the business process also depends on the quality of data available for that process. It is the most important step of the data modelling process, and a failure here would have cascading and irreparable defects.
- To describe the business process, you can use plain text or use basic Business Process Modelling Notation (BPMN) or Unified Modelling Language (UML).

2. Identify the Grain

- The grain describes the level of detail for the business problem/solution. It is the process of identifying the lowest level of information for any table in your data warehouse. If a table contains sales data for every day, then it should be daily granularity. If a table contains total sales data for each month, then it has monthly granularity.
- During this stage, you answer questions like -
 1. Do we need to store all the available products or just a few types of products? This decision is based on the business processes selected for data warehouse.

2. Do we store the product sale information on a monthly, weekly, daily or hourly basis? This decision depends on the nature of reports requested by executives.
3. How do the above two choices affect the database size?
 - Example of Grain: The CEO at an MNC wants to find the sales for specific products in different locations on a daily basis. So, the grain is "product sale information by location by the day."
3. **Identify Dimensions and Attributes**
 - Dimensions are nouns like date, store, inventory, etc. These dimensions are where all the data should be stored. For example, the date dimension may contain data like a year, month and weekday.
 - Example of Dimensions: The CEO at an MNC wants to find the sales for specific products in different locations on a daily basis.
Dimensions: Product, Location and Time
Attributes: For Product: Product key (Foreign Key), Name, Type, Specifications
Hierarchies: For Location: Country, State, City, Street Address, Name
4. **Identify Facts**
 - This step is co-associated with the business users of the system because this is where they get access to data stored in the data warehouse. Most of the fact table rows are numerical values like price or cost per unit, etc.
 - Example of Facts: The CEO at an MNC wants to find the sales for specific products in different locations on a daily basis. The fact here is Sum of Sales by product by location by time.
5. **Build Schema**
 - In this step, you implement the Dimension Model. A schema is nothing but the database structure (arrangement of tables). There are two popular schemas
 1. Star Schema
 2. Snowflake Schema

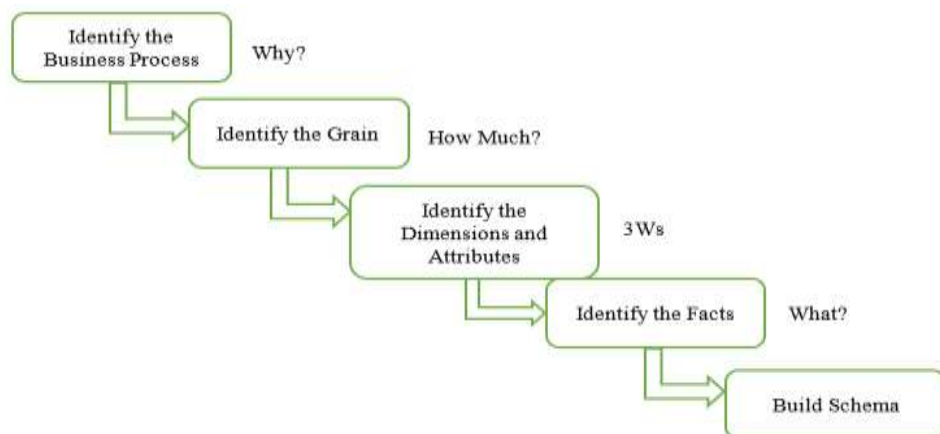


Fig. 6.4 Steps in Dimensional Modeling

6.4.3 Advantages of Dimensional Modelling

1. The standardization of dimensions enables easy reporting across areas of business.
2. The history of dimensional information is stored in dimension tables.
3. It enables the addition of a completely new dimension to the fact table without causing substantial disturbances.
4. Dimensional data is also used to store data in a way that makes it easier to retrieve information from it once it has been saved in a database.
5. The dimensional table is easier to understand than the normalized model.
6. The information is organized into easy-to-understand business categories.
7. Dimensional models can accommodate change easily. Dimension tables can have more columns added to them without affecting existing business intelligence applications using these tables.
8. The business can easily understand the dimensional model. The business understands what each fact, dimension, or characteristic implies because this model is based on business concepts.
9. For quick data searching, dimensional models are deformed and optimised. This approach is recognised by many relational database platforms, which optimise query execution plans to improve performance.
10. In a data warehouse, dimensional modelling generates a schema that is designed for optimum performance. It reduces the number of joins and reduces data redundancy.
11. In addition, the dimensional model aids query performance. Because it is more denormalized, it is better for querying.

6.5 DATA CUBE OPERATIONS

An OLAP Cube is at the heart of the OLAP idea. The OLAP cube is a data structure that is designed to allow for rapid data analysis. The OLAP Cube is made up of measurements, which are numerical data that are categorized by dimensions. The hypercube is another name for the OLAP Cube.

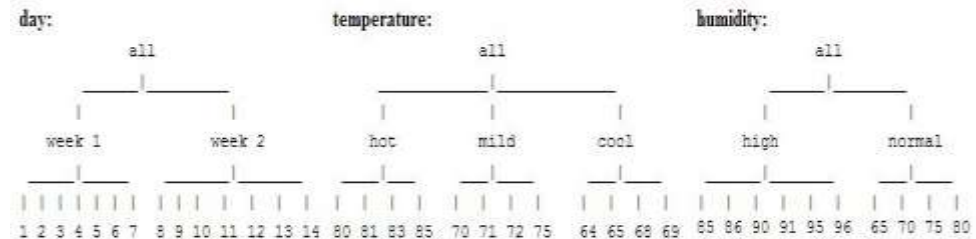
Data processing and analysis are usually done with a simple spreadsheet, which has data values organized in a row and column structure. For two-dimensional data, this is ideal. OLAP, on the other hand, comprises multidimensional data, which is typically collected from a separate and unrelated source. Using a spreadsheet isn't the best solution. The cube can logically and orderly store and evaluate multidimensional data.

A data warehouse extracts data from a variety of sources and formats, including text files, excel sheets, multimedia files, and so on. The data is cleaned and modified after it has been extracted. Data is fed into an OLAP server (or OLAP cube), which calculates information ahead of time for future analysis.

6.5.1 OLAP Operations

In the multidimensional model, data are organized into multiple dimensions, and each dimension contains multiple levels of abstraction defined by concept hierarchies. This organization provides users with the flexibility to view data from different perspectives.

For example, we have attributes as day, temperature and humidity, we can group values in subsets and name these subsets, thus obtaining a set of hierarchies as shown in figure below.



OLAP provides a user-friendly environment for interactive data analysis. A number of OLAP data cube operations exist to materialize different views of data, allowing interactive querying and analysis of the data.

The most popular end user operations on dimensional data are:

1. Roll-Up

The roll-up operation (also called drill-up or aggregation operation) performs aggregation on a data cube, either by climbing up a concept hierarchy for a dimension or by climbing down a concept hierarchy, i.e. dimension reduction. Let us explain roll up with an example:

Consider the following cubes illustrating temperature of certain days recorded weekly:

Temperature	64	65	68	69	70	71	72	75	80	81	83	85
Week1	1	0	1	0	1	0	0	0	0	0	1	0
Week2	0	0	0	1	0	0	1	2	0	1	0	0

Consider that we want to set up levels (hot (80-85), mild (70-75), cool (64-69)) in temperature from the above cubes.

To do this, we have to group column and add up the value according to the concept hierarchies. This operation is known as a roll-up.

By doing this, we get the following cube:

Temperature	Cool	mild	hot
Week1	2	1	1
Week2	2	1	1

The concept hierarchy can be defined as hot→day→week. The roll-up operation groups the data by levels of temperature.

2. Drill-Down

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

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

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

Temperature	Cool	mild	Hot
Day 1	0	0	0
Day 2	0	0	0
Day 3	0	0	1
Day 4	0	1	0
Day 5	1	0	0
Day 6	0	0	0
Day 7	1	0	0
Day 8	0	0	0
Day 9	1	0	0
Day 10	0	1	0
Day 11	0	1	0
Day 12	0	1	0
Day 13	0	0	1
Day 14	0	0	0

3. Slice

A slice is a subset of the cubes corresponding to a single value for one or more members of the dimension. For example, a slice operation is executed when the customer wants a selection on one dimension of a three-dimensional cube resulting in a two-dimensional site. So, the slice operations perform a selection on one dimension of the given cube, thus resulting in a sub-cube. It will form a new sub-cubes by selecting one or more dimensions.

For example, if we make the selection, temperature=cool we will obtain the following cube:

Temperature	Cool
Day 1	0
Day 2	0
Day 3	0
Day 4	0
Day 5	1
Day 6	1
Day 7	1
Day 8	1
Day 9	1
Day 11	0
Day 12	0
Day 13	0
Day 14	0

4. Dice

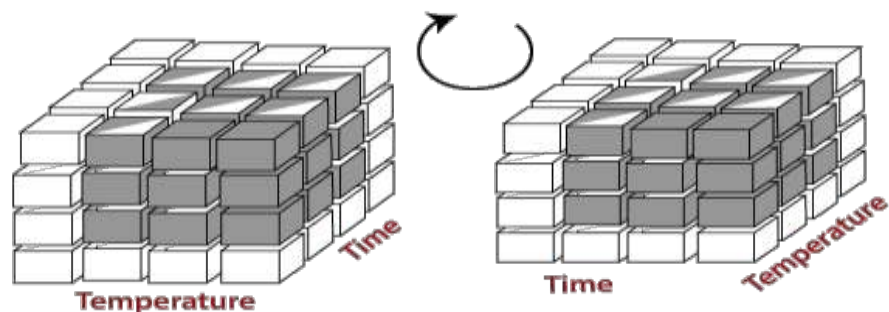
The dice operation describes a sub-cube by operating a selection on two or more dimension.

For example, Implement the selection (time = day 3 OR time = day 4) AND (temperature = cool OR temperature = hot) to the original cubes we get the following sub-cube (still two-dimensional)

Temperature	Cool	hot
Day 3	0	1
Day 4	0	0

5. Pivot

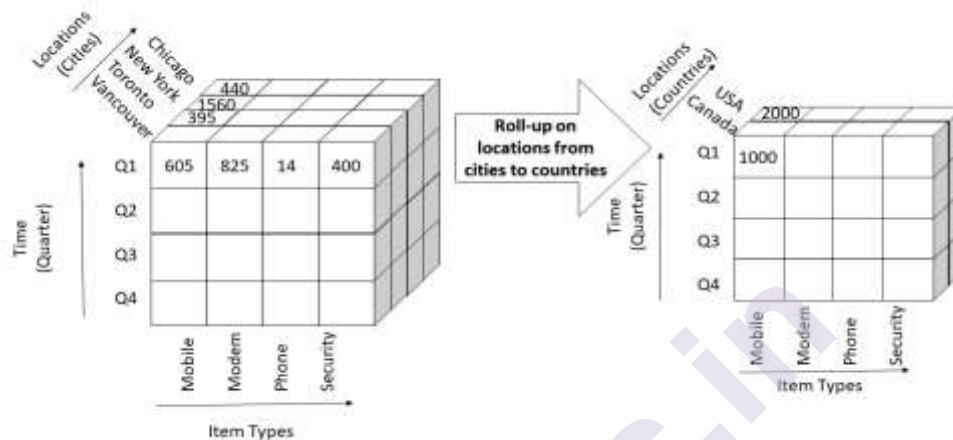
The pivot operation is also called a rotation. Pivot is a visualization operation which rotates the data axes in view to provide an alternative presentation of the data. It may contain swapping the rows and columns or moving one of the row-dimensions into the column dimensions.



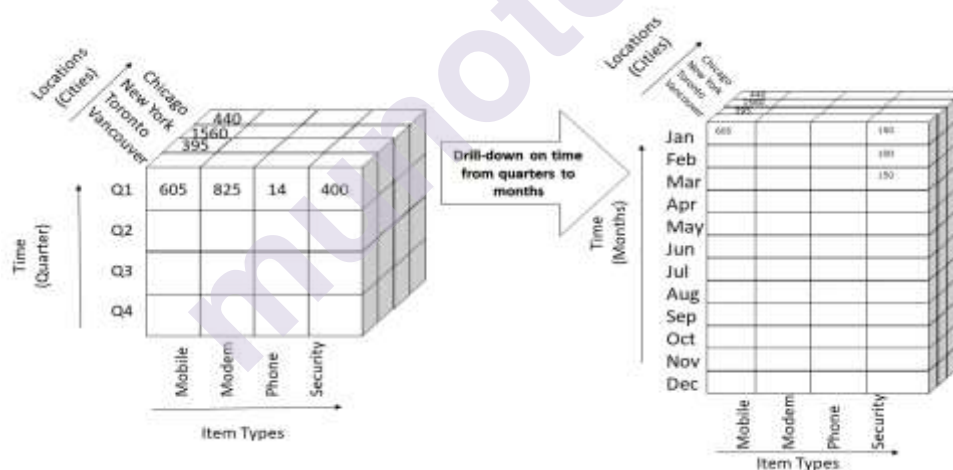
Example: Let's look at some typical OLAP operations for multidimensional data. Each of the following operations described is illustrated below. In the figure is a data cube for Digi1 Electronics

sales. The cube contains the dimension location, time, and item, where location is aggregated with respect to city values, time is aggregated with respect to quarters, and item is aggregated with respect to item types. The measure displayed is dollars sold (in thousands). (For improved readability, only some of the cubes' cell values are shown.) The data examined are for the cities Chicago, New York, Toronto, and Vancouver.

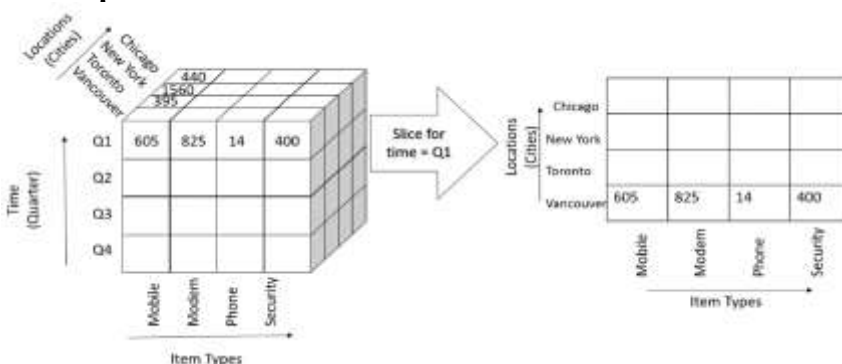
1. Roll-up Operation



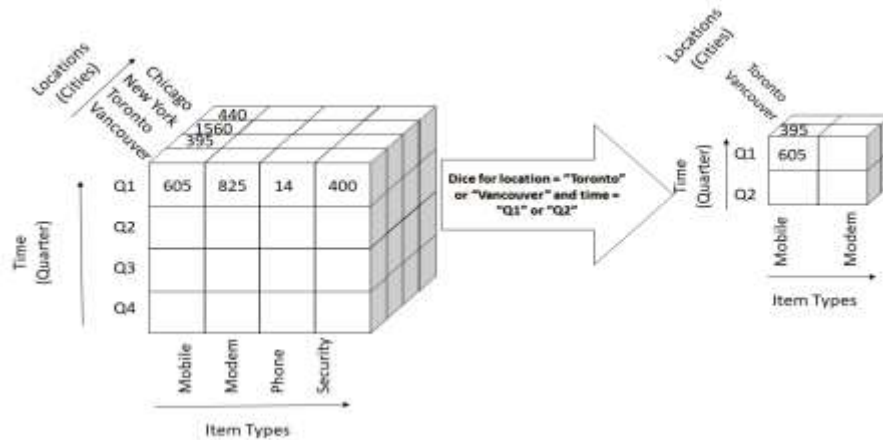
2. Drill-down Operation



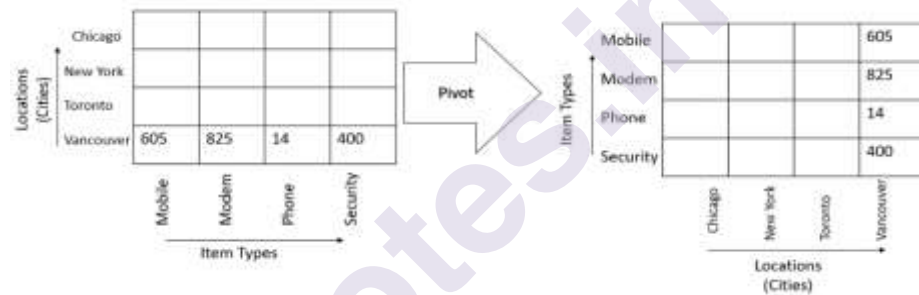
3. Slice Operation



4. Dice Operation



5. Pivot Operation



6.6 DATA CUBE SCHEMAS

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. Much like a database, a data warehouse also requires to maintain a schema. A database uses relational model, while a data warehouse uses Star, Snowflake, and Fact Constellation schema.

3.6.1 Star Schema

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

- The center of the schema consists of a large fact table, and the points of the star are the dimension tables.

Fact Tables

- It is a table in a star schema which contains facts and connected to dimensions.
- A fact table has two types of columns: those that include fact and those that are foreign keys to the dimension table.
- The primary key of the fact tables is generally a composite key that is made up of all of its foreign keys.

Dimension Tables

- A dimension is an architecture usually composed of one or more hierarchies that categorize data.
- If a dimension has not got hierarchies and levels, it is called a **flat dimension** or **list**.
- The primary keys of each of the dimension table are part of the composite primary keys of the fact table.
- Dimensional attributes help to define the dimensional value. They are generally descriptive, textual values.
- Dimensional tables are usually small in size than fact table.

Note: Fact tables are deep whereas dimension tables are wide as fact tables will have a higher number of rows and a lesser number of columns. A primary key defined in the fact table is primarily to identify each row separately. The primary key is also called a Composite key in fact table. A dimension table contains a higher granular information so have less no of records and it needs to have all the necessary details (more columns) related to the grain of the table. On the other side, a fact table has the lowest level grain of a subject area. Lower grain causes more number of rows in the Fact table.

6.6.1.1 Characteristics of Star Schema

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

6.6.1.2 Keys in Star Schema

1. **Primary Keys:** The primary key of the dimension table identifies each row in a dimension table. Example: In a student dimension table, student_id is the primary key which identifies each student uniquely.
2. **Surrogate Keys:** System generated sequence numbers are called surrogate keys. They do not have any built in meanings.

3. **Foreign Keys:** Every dimension table has one-to-one relationship with the fact table. The primary key in the dimension table acts as a foreign key in the fact table.

6.6.1.3 Advantages of Star Schema

1. Query Performance

- A star schema database has a limited number of table and clear join paths, the query run faster than they do against OLTP systems. Small single-table queries, frequently of a dimension table, are almost instantaneous. Large join queries that contain multiple tables takes only seconds or minutes to run.
- In a star schema database design, the dimension is connected only through the central fact table. When the two-dimension table is used in a query, only one join path, intersecting the fact tables, exist between those two tables. This design feature enforces authentic and consistent query results.

2. Load performance and administration

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

3. Built-in referential integrity

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

4. Easily Understood

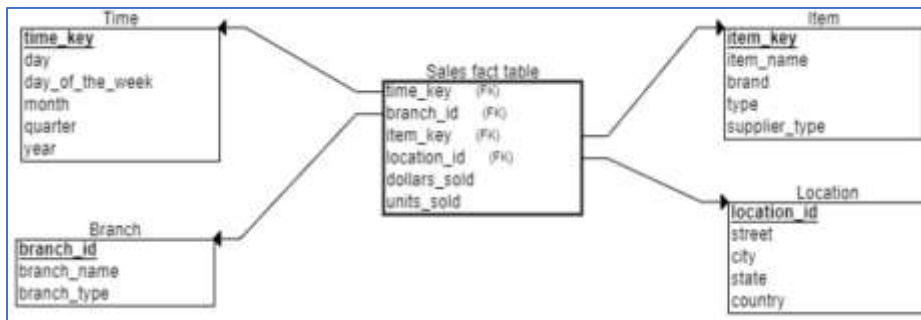
- A star schema is simple to understand and navigate, with dimensions joined only through the fact table. These joins are more significant to the end-user because they represent the fundamental relationship between parts of the underlying business. Customer can also browse dimension table attributes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

6.6.1.4 Disadvantages of Star Schema

- Data integrity is not enforced well since in a highly de-normalized schema state.
- Not flexible in terms if analytical needs as a normalized data model.

- Star schemas don't reinforce many-to-many relationships within business entities, at least not frequently.

6.6.1.5 Example: A star schema for Digi1 Electronics sales is shown. Sales are considered along four dimensions: time, item, branch, and location. The schema contains a central fact table for sales that contains keys to each of the four dimensions, along with two measures: dollars sold and units sold.



6.6.2 Snowflake Schema

- The snowflake schema is a variant of the star schema.
- Here, the centralized fact table is connected to multiple dimensions.
- In the snowflake schema, dimensions are present in a normalized form in multiple related tables.
- The snowflake structure is materialized when the dimensions of a star schema are detailed and highly structured, having several levels of relationship, and the child tables have multiple parent table.
- The snowflake schema affects only the dimension tables and does not affect the fact tables.
- In other words, a dimension table is said to be snowflaked if the low-cardinality attribute of the dimensions has been divided into separate normalized tables.
- These tables are then joined to the original dimension table with referential constraints (foreign key constraint).

6.6.2.1 Characteristics of Snowflake Schema

- The snowflake schema uses small disk space.
- It is easy to implement dimension that is added to schema.
- There are multiple tables, so performance is reduced.
- The dimension table consist of two or more sets of attributes which define information at different grains.
- The sets of attributes of the same dimension table are being populated by different source systems.

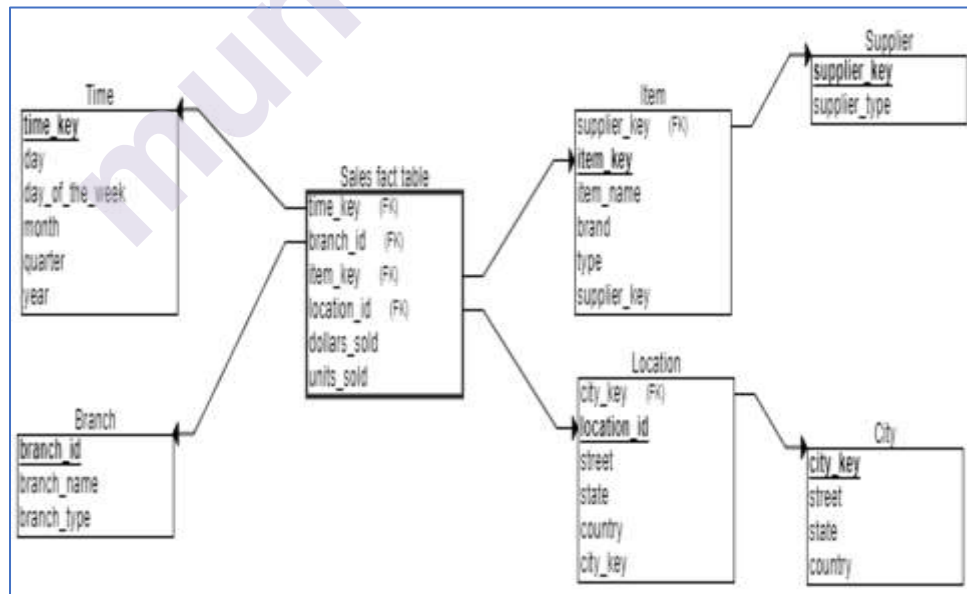
6.6.2.2 Advantages of Snowflake Schema

- It provides structured data which reduces the problem of data integrity.
- It uses small disk space because data are highly structured.

6.6.2.3 Disadvantages of Snowflake Schema

- Snowflaking reduces space consumed by dimension tables, but compared with the entire data warehouse the saving is usually insignificant.
- Avoid snowflaking or normalization of a dimension table, unless required and appropriate.
- Do not snowflake hierarchies of one-dimension table into separate tables. Hierarchies should belong to the dimension table only and should never be snowflaked.

6.6.2.4 Example: A snowflake schema for Digi1 Electronics sales is given. Here, the sales fact table is identical to that of the star schema in Figure 4.6. The main difference between the two schemas is in the definition of dimension tables. The single dimension table for item in the star schema is normalized in the snowflake schema, resulting in new item and supplier tables. For example, the item dimension table now contains the attributes item_key, item_name, brand, type, and supplier_key, where supplier_key is linked to the supplier dimension table, containing supplier_key and supplier_type information. Similarly, the single dimension table for location in the star schema can be normalized into two new tables: location and city. The city_key in the new location table links to the city dimension. Notice that, when desirable, further normalization can be performed on state and country in the snowflake schema.



3.6.2 Star Schema Vs Snowflake Schema

Table 6.2: Star Schema Vs Snowflake Schema

Sr. No.	Basis	Star Schema	Snowflake Schema
1	Ease of Maintenance/change	It has redundant data and hence less easy to maintain/change.	No redundancy and therefore more easy to maintain and change.
2	Ease of Use	Less complex queries and simple to understand.	More complex queries and therefore less easy to understand.
3	Parent table	In a star schema, a dimension table will not have any parent table.	In a snowflake schema, a dimension table will have one or more parent tables.
4	Query Performance	Less number of foreign keys and hence lesser query execution time.	More foreign keys and thus more query execution time.
5	Normalization	It has De-normalized tables.	It has normalized tables.
6	Type of Data Warehouse	Good for data marts with simple relationships (one to one or one to many).	Good to use for data warehouse core to simplify complex relationships (many to many).
7	Joins	Fewer joins	Higher number of joins.
8	Dimension Table	It contains only a single dimension table for each dimension.	It may have more than one dimension table for each dimension.
9	Hierarchies	Hierarchies for the dimension are stored in the dimensional table itself in a star schema.	Hierarchies are broken into separate tables in a snowflake schema. These hierarchies help to drill down the information from topmost hierarchies to the lowermost hierarchies.

10	When to use	When the dimensional table contains less number of rows, we can go for Star schema.	When dimensional table store a huge number of rows with redundancy information and space is such an issue, we can choose snowflake schema to store space.
11	Data Warehouse system	Work best in any data warehouse/ data mart.	Better for small data warehouse/data mart.

6.7 REFERENCES

- Business Intelligence (2nd Edition), Efraim Turban, Ramesh Sharda, Dursun Delen, David King, Pearson
- Business Intelligence for Dummies, Swain Scheps, Wiley Publications
- Building the Data Warehouse, Inmon, Wiley Publications
- <https://www.geeksforgeeks.org/>
- <https://www.javatpoint.com/>

6.8 SUMMARY

OLTP stands for online data modification, whereas OLAP stands for online historical multidimensional data store, which is used to access massive volumes of data for analysis. OLAP is used to analyze data that has been recorded by one or more OLTP systems. In the industries, view materialization is not the same as data warehouse design. It considers data warehouses to be database systems with specific requirements, such as responding to management inquiries. The goal of the design is to determine how records from numerous data sources should be extracted, transformed, and loaded (ETL) into a data warehouse database.

Dimensional modelling uses a cube operation to represent data, making OLAP data management more suitable for logical data representation. Ralph Kimball created the Dimensional Modeling perception, which consists of "fact" and "dimension" tables. We have also discussed different data cube operations and star, snowflake schema.

DATA MINING BASICS

Unit Structure

- 7.0 Objectives
- 7.1 Introduction to Data Mining
- 7.2 Sources of Data that can be Mined
- 7.3 Kind of Patterns to be Mined
- 7.4 Data Mining Technologies
- 7.5 Difference between Data Mining and Data Warehouse
- 7.6 Data Mining Task Primitives
- 7.7 Data Mining Architecture
- 7.8 KDD Process
- 7.9 Issues in Data Mining
- 7.10 Applications of Data Mining
- 7.11 Benefits of Data Mining
- 7.12 Disadvantages of Data Mining
- 7.13 Summary
- 7.14 Test your skills
- 7.15 Descriptive Questions
- 7.16 Reference for Further Reading

7.0 OBJECTIVES

- To understand the fundamentals of data mining.
- To identify the appropriateness and need of mining the data.
- To understand data mining architecture and KDD process.
- To learn issues, applications, benefits and disadvantages of data mining.

7.1 INTRODUCTION TO DATA MINING

- The Information Industry has a massive amount of data available. This data is useless until it is transformed into usable information. This massive amount of data must be evaluated and meaningful information must be extracted from it.
- William J Frawley, Gregory Piatetsky-Shapiro and Christopher J Matheus define **data mining** as “**The non-trivial extraction of implicit, previously unknown, and potentially useful information from data.**”
- The process of extracting information to identify patterns, trends, and useful data that would allow the business to take the data-driven decision from huge sets of data is called Data Mining.

- The computer is responsible for finding the patterns by identifying the underlying rules and features in the data.
- In other words, data mining is the process of investigating hidden patterns of information from various perspectives for categorization into useful data, which is collected and assembled in specific areas such as data warehouses, efficient analysis, data mining algorithms, assisting decision making, and other data requirements to eventually cost-cutting and revenue generation.

7.2 SOURCES OF DATA THAT CAN BE MINED

The data from multiple sources are integrated into a common source known as Data Warehouse. Let's discuss what type of data can be mined:

1. Flat Files

- Flat files are defined as data files in text form or binary form with a structure that can be easily extracted by data mining algorithms.
- Data stored in flat files have no relationship or path among themselves, like if a relational database is stored on flat file, then there will be no relations between the tables.
- Flat files are represented by data dictionary. E.g.: CSV file.
- **Application** : Used in Data Warehousing to store data, used in carrying data to and from server, etc.

2. Relational Databases

- A Relational database is defined as the collection of data organized in tables with rows and columns.
- Physical schema in Relational databases is a schema which defines the structure of tables.
- Logical schema in Relational databases is a schema which defines the relationship among tables.
- Standard API of relational database is SQL.
- **Application**: Data Mining, ROLAP model, etc.

3. Data Warehouse

- A data warehouse is defined as the collection of data integrated from multiple sources that will queries and decision making.
- There are three types of data warehouse: Enterprise data warehouse, Data Mart and Virtual Warehouse.
- Two approaches can be used to update data in Data Warehouse: Query-driven Approach and Update-driven Approach.
- **Application**: Business decision making, Data mining, etc.

4. Transactional Databases

- Transactional database is a collection of data organized by time stamps, date, etc. to represent transaction in databases.
- This type of database has the capability to roll back or undo its operation when a transaction is not completed or committed.
- Highly flexible system where users can modify information without changing any sensitive information.
- Follows ACID (Atomicity, Consistency, Isolation and Durability) property of DBMS.
- **Application:** Banking, Distributed systems, Object databases, etc.

5. Multimedia Databases

- Multimedia databases consists audio, video, images and text media.
- They can be stored on Object-Oriented Databases.
- They are used to store complex information in a pre-specified formats.
- **Application:** Digital libraries, video-on demand, news-on demand, musical database, etc.

6. Spatial Database

- Store geographical information.
- Stores data in the form of coordinates, topology, lines, polygons, etc.
- **Application:** Maps, Global positioning, etc.

7. Time-series Databases

- Time series databases contains stock exchange data and user logged activities.
- Handles array of numbers indexed by time, date, etc.
- It requires real-time analysis.
- **Application:** eXtremeDB, Graphite, InfluxDB, etc.

8. WWW

- WWW refers to World wide web is a collection of documents and resources like audio, video, text, etc. which are identified by Uniform Resource Locators (URLs) through web browsers, linked by HTML pages, and accessible via the Internet network.
- It is the most heterogeneous repository as it collects data from multiple resources.
- It is dynamic in nature as Volume of data is continuously increasing and changing.
- **Application:** Online shopping, Job search, Research, studying, etc.

7.3 KIND OF PATTERNS TO BE MINED

Data mining is highly effective, so long as it draws upon one or more of these techniques:

1. **Tracking patterns** : Ability to identify patterns in your data sets is one of the most basic strategies in data mining. This is frequently an identification of a periodic aberration in your data or a time-dependent ebb and flow of a variable. For example, you might notice an increase in sales of a certain product immediately before the holidays, or that the warmer weather attracts more visitors to your website.
2. **Classification** : Classification is a more advanced data mining technique that requires you to group together diverse attributes into identifiable groups, which you can then use to make additional conclusions or perform a specific job. When analysing data on individual customers' financial backgrounds and purchasing histories, for example, you may be able to classify them as "low," "medium," or "high" credit risks. These classifications might then be used to learn even more about those customers.
3. **Association** : Tracking patterns are connected to association, although dependently linked variables are more specific. In this situation, you'll seek for certain events or attributes that are strongly linked to another event or attribute; for example, you might discover that when your consumers buy one thing, they frequently buy another, related item. This is commonly used to populate online store "people also bought" sections.
4. **Outlier detection** : In many circumstances, simply finding the overall pattern will not provide you with a complete picture of your data. You must also be able to spot anomalies, sometimes known as outliers, in your data. If, for example, your buyers are nearly all male but there's a sharp rise in female buyers during one unexpected week in July, you'll want to research the rise and figure out what caused it, so you can either reproduce it or better understand your audience.
5. **Clustering** : Clustering is similar to classification in that it involves putting together groups of data based on their similarities. For example, you may group different demographics of your audience into distinct packages based on their pocket money or how frequently they purchase at your store.
6. **Regression** : Regression is a type of planning and modelling that is used to determine the probability of a particular variable given the presence of other variables. You may use it, for example, to forecast a price based on other criteria such as availability, consumer demand, and competition. Regression is primarily concerned with determining the exact relationship between two (or more) variables in a given data set.

7. **Prediction** : One of the most significant data mining approaches is prediction, which is used to forecast the types of data you'll see in the future. In many circumstances, simply recognising and comprehending historical tendencies is sufficient to create a reasonably accurate forecast of what will occur in the future. For example, you might look at a consumer's credit history and previous transactions to see if they're at credit risk in the future.

7.4 DATA MINING TECHNOLOGIES

Several techniques are used in the development of data mining methods. Some of them are mentioned below:

1. Statistics

- It uses the mathematical analysis to express representations, model and summarize empirical data or real world observations.
- Statistical analysis involves the collection of methods, applicable to large amount of data to conclude and report the trend.

2. Machine learning

- *Arthur Samuel* defined machine learning as “a field of study that gives computers the ability to learn without being programmed”.
- When the new data is entered in the computer, algorithms help the data to grow or change due to machine learning.
- In machine learning, an algorithm is constructed to predict the data from the available database (Predictive analysis).
- It is related to computational statistics.
- The four types of machine learning are:
- **Supervised learning**: It is based on the classification. It is also called as inductive learning. In this method, the desired outputs are included in the training dataset.
- **Unsupervised learning**: Unsupervised learning is based on clustering. Clusters are formed on the basis of similarity measures and desired outputs are not included in the training dataset.
- **Semi-supervised learning**: Semi-supervised learning includes some desired outputs to the training dataset to generate the appropriate functions. This method generally avoids the large number of labeled examples (i.e. desired outputs).
- **Active learning**: Active learning is a powerful approach in analyzing the data efficiently. The algorithm is designed in such a way that; the desired output should be decided by the algorithm itself (the user plays important role in this type).

3. Information retrieval

- Information deals with uncertain representations of the semantics of objects (text, images). For example: Finding relevant information from a large document.

4. Database systems and data warehouse

- Databases are used for the purpose of recording the data as well as data warehousing.
- Online Transactional Processing (OLTP) uses databases for day to day transaction purpose.
- To remove the redundant data and save the storage space, data is normalized and stored in the form of tables.
- Entity-Relational modeling techniques are used for relational database management system design.
- Data warehouses are used to store historical data which helps to take strategic decision for business.
- It is used for online analytical processing (OLAP), which helps to analyze the data.

5. Decision support system

- Decision support system is a category of information system. It is very useful in decision making for organizations.
- It is an interactive software based system which helps decision makers to extract useful information from the data, documents to make the decision.

7.5 DIFFERENCE BETWEEN DATA MINING AND DATA WAREHOUSE

Table 7.1 : Data Mining Vs Data Warehouse

Sr. No.	Data Mining	Data Warehouse
1.	Data mining is the process of analyzing unknown patterns of data.	A data warehouse is database system which is designed for analytical instead of transactional work.
2.	Data mining is a method of comparing large amounts of data to finding right patterns.	Data warehousing is a method of centralizing data from different sources into one common repository.
3.	Data mining is usually done by business users with the assistance of engineers.	Data warehousing is a process which needs to occur before any data mining can take place.

Sr. No.	Data Mining	Data Warehouse
4.	Data mining is the considered as a process of extracting data from large data sets.	On the other hand, Data warehousing is the process of pooling all relevant data together.
5.	One of the most important benefits of data mining techniques is the detection and identification of errors in the system.	One of the pros of Data Warehouse is its ability to update consistently. That's why it is ideal for the business owner who wants the best and latest features.
6.	Data mining helps to create suggestive patterns of important factors. Like the buying habits of customers, products, sales. So that, companies can make the necessary adjustments in operation and production.	Data Warehouse adds an extra value to operational business systems like CRM systems when the warehouse is integrated.
7.	The Data mining techniques are never 100% accurate and may cause serious consequences in certain conditions.	In the data warehouse, there is great chance that the data which was required for analysis by the organization may not be integrated into the warehouse. It can easily lead to loss of information.
8.	The information gathered based on Data Mining by organizations can be misused against a group of people.	Data warehouses are created for a huge IT project. Therefore, it involves high maintenance system which can impact the revenue of medium to small-scale organizations.
9.	After successful initial queries, users may ask more complicated queries which would increase the workload.	Data Warehouse is complicated to implement and maintain.
10	Organisations can benefit from this analytical tool by equipping pertinent and usable knowledge-based information.	Data warehouse stores a large amount of historical data which helps users to analyze different time periods and trends for making future predictions.

Sr. No.	Data Mining	Data Warehouse
11.	Organisations need to spend lots of their resources for training and Implementation purpose. Moreover, data mining tools work in different manners due to different algorithms employed in their design.	In Data warehouse, data is pooled from multiple sources. The data needs to be cleaned and transformed. This could be a challenge.
12.	The data mining methods are cost-effective and efficient compares to other statistical data applications.	Data warehouse's responsibility is to simplify every type of business data. Most of the work that will be done on user's part is inputting the raw data.
13.	Another critical benefit of data mining techniques is the identification of errors which can lead to losses. Generated data could be used to detect a drop-in sale.	Data warehouse allows users to access critical data from the number of sources in a single place. Therefore, it saves user's time of retrieving data from multiple sources.
14.	Data mining helps to generate actionable strategies built on data insights.	Once you input any information into Data warehouse system, you will unlikely to lose track of this data again. You need to conduct a quick search, helps you to find the right statistic information.

7.6 DATA MINING TASK PRIMITIVES

- Each user will have a data mining task in mind, that is, some form of data analysis that he or she would like to have performed.
- A data mining task can be specified in the form of a data mining query, which is input to the data mining system.
- A data mining query is defined in terms of data mining task primitives.
- These primitives allow the user to interactively communicate with the data mining system during discovery in order to direct the mining process, or examine the findings from different angles or depths.
- Here is the list of Data Mining Task Primitives.

1. **The set of task-relevant data to be mined :** 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).

2. **The kind of knowledge to be mined :** 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. **The background knowledge to be used in the discovery process :** This knowledge about the domain to be mined is useful for guiding the knowledge discovery process and for evaluating the patterns found. Concept hierarchies are a popular form of background knowledge, which allow data to be mined at multiple levels of abstraction.
4. **The interestingness measures and thresholds for pattern evaluation :** 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, interestingness measures for association rules include support and confidence. Rules whose support and confidence values are below user-specified thresholds are considered uninteresting.
5. **The expected representation for visualizing the discovered patterns :** This refers to the form in which discovered patterns are to be displayed, which may include rules, tables, charts, graphs, decision trees, and cubes.

7.7 DATA MINING ARCHITECTURE

The significant components of data mining systems are a data source, data mining engine, data warehouse server, the pattern evaluation module, graphical user interface, and knowledge base.

- **Data Source**
 - The actual source of data is the Database, data warehouse, World Wide Web (WWW), text files, and other documents. You need a huge amount of historical data for data mining to be successful.
 - Organizations typically store data in databases or data warehouses. Data warehouses may comprise one or more databases, text files spreadsheets, or other repositories of data.
 - Sometimes, even plain text files or spreadsheets may contain information. Another primary source of data is the World Wide Web or the internet.



Fig. 7.1: Data Mining Architecture

- **Data Cleaning, Integration and Selection**
 - Before passing the data to the database or data warehouse server, the data must be cleaned, integrated and selected.
 - As the information comes from various sources and in different formats, it can't be used directly for the data mining procedure because the data may not be complete and accurate. So, the first data requires to be cleaned and unified.
 - More information than needed will be collected from various data sources, and only the data of interest will have to be selected and passed to the server. These procedures are not as easy as we think. Several methods may be performed on the data as part of selection, integration, and cleaning.
- **Database or Data Warehouse Server**
 - The database or data warehouse server consists of the original data that is ready to be processed. Hence, the server is cause for retrieving the relevant data that is based on data mining as per user request.
- **Data Mining Engine**
 - The data mining engine is a major component of any data mining system. It contains several modules for operating data mining tasks, including association, characterization, classification, clustering, prediction, time-series analysis, etc.
 - In other words, we can say data mining is the root of our data mining architecture. It comprises instruments and software used to obtain insights and knowledge from data collected from various data sources and stored within the data warehouse.

- **Pattern Evaluation Module**
 - The Pattern evaluation module is primarily responsible for the measure of investigation of the pattern by using a threshold value. It collaborates with the data mining engine to focus the search on exciting patterns.
 - This segment commonly employs stake measures that cooperate with the data mining modules to focus the search towards fascinating patterns. It might utilize a stake threshold to filter out discovered patterns.
 - On the other hand, the pattern evaluation module might be coordinated with the mining module, depending on the implementation of the data mining techniques used. For efficient data mining, it is abnormally suggested to push the evaluation of pattern stake as much as possible into the mining procedure to confine the search to only fascinating patterns.
- **Graphical User Interface**
 - The graphical user interface (GUI) module communicates between the data mining system and the user. This module helps the user to easily and efficiently use the system without knowing the complexity of the process.
 - This module cooperates with the data mining system when the user specifies a query or a task and displays the results.
- **Knowledge Base**
 - The knowledge base is helpful in the entire process of data mining. It might be helpful to guide the search or evaluate the stake of the result patterns.
 - The knowledge base may even contain user views and data from user experiences that might be helpful in the data mining process.
 - The data mining engine may receive inputs from the knowledge base to make the result more accurate and reliable.
 - The pattern assessment module regularly interacts with the knowledge base to get inputs, and also update it.

GQ. Suppose your task as a software engineer at DB-University is to design a data mining system to examine their university course database, which contains the following information: the name, address, and status (e.g., undergraduate or graduate) of each student, the courses taken, and their cumulative grade point average (GPA). Describe the architecture you would choose. What is the purpose of each component of this architecture?

A data mining architecture that can be used for this application would consist of the following major components :

- A database, data warehouse, or other information repository, which consists of the set of databases, data warehouses,

spreadsheets, or other kinds of information repositories containing the student and course information.

- A database or data warehouse server which fetches the relevant data based on users' data mining requests.
- A knowledge base that contains the domain knowledge used to guide the search or to evaluate the interestingness of resulting patterns. For example, the knowledge base may contain metadata which describes data from multiple heterogeneous sources.
- A data mining engine, which consists of a set of functional modules for tasks such as classification, association, classification, cluster analysis, and evolution and deviation analysis.
- A pattern evaluation module that works in tandem with the data mining modules by employing interestingness measures to help focus the search towards interestingness patterns.
- A graphical user interface that allows the user an interactive approach to the data mining system.

7.8 KDD PROCESS

- Knowledge discovery in the database (KDD) is the process of searching for hidden knowledge in the massive amounts of data that we are technically capable of generating and storing.
- The basic task of KDD is to extract knowledge (or information) from a lower level data (databases).
- It is the non-trivial (significant) process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data.
- The goal is to distinguish between unprocessed data something that may not be obvious but is valuable or enlightening in its discovery.
- The overall process of finding and interpreting patterns from data involves the repeated application of the following steps :

1. Data Cleaning

- Removal of noise, inconsistent data, and outliers
- Strategies to handle missing data fields.

2. Data Integration

- Data from various sources such as databases, data warehouse, and transactional data are integrated.
- Multiple data sources may be combined into a single data format.

3. Data Selection

- Data relevant to the analysis task is retrieved from the database.
- Collecting only necessary information to the model.

- Finding useful features to represent data depending on the goal of the task.

4. Data Transformation

- Data are transformed and consolidated into forms appropriate for mining by performing summary or aggregation operations.
- By using transformation methods invariant representations for the data is found.

5. Data Mining

- An essential process where intelligent methods are applied to extract data patterns.
- Deciding which model and parameter may be appropriate.

6. Pattern Evaluation

- To identify the truly interesting patterns representing knowledge based on interesting measures.

7. Knowledge Presentation

- Visualization and knowledge representation techniques are used to present mined knowledge to users.
- Visualizations can be in form of graphs, charts or table.

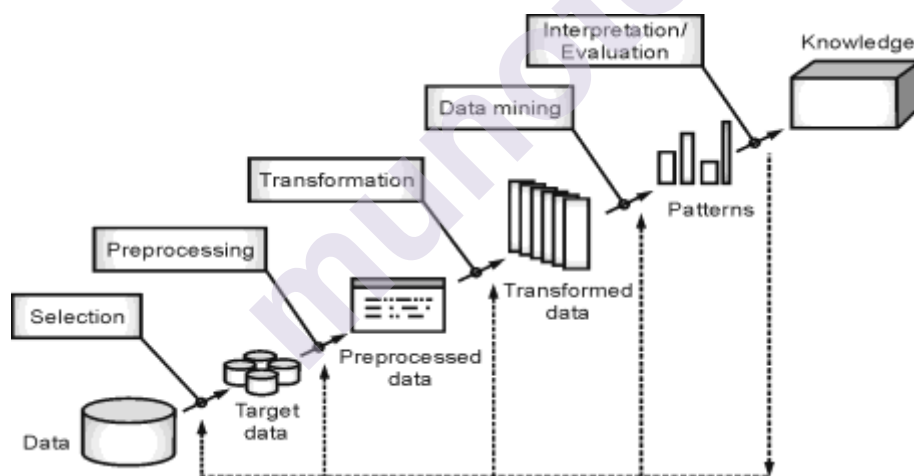


Fig. 7.2: KDD Process

7.9 ISSUES IN DATA MINING

Data mining systems face a lot of challenges and issues in today's world. Some of them are:

1. Mining methodology and user interaction issues
2. Performance issues
3. Issues relating to the diversity of database types

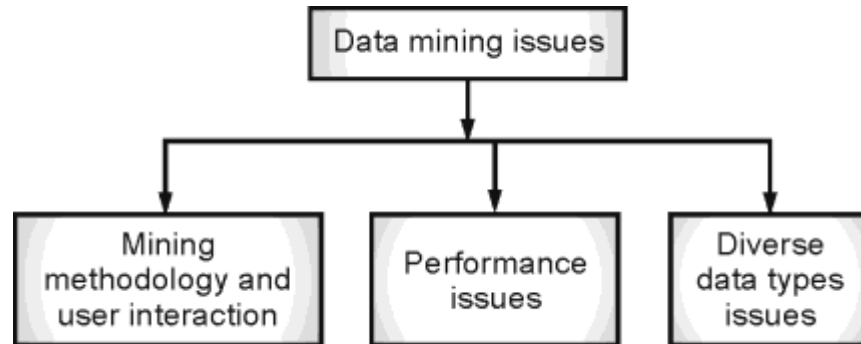


Fig. 7.3 : Data Mining Issues

1. Mining Methodology and User Interaction Issues

The issues in this category are as follows:

- **Mining different kinds of knowledge in databases:** This issue is responsible for addressing the problems of covering a big range of data in order to meet the needs of the client or the customer. Due to the different information or a different way, it becomes difficult for a user to cover a big range of knowledge discovery task.
- **Interactive mining of knowledge at multiple levels of abstraction:** Interactive mining is very crucial because it permits the user to focus the search for patterns, providing and refining data mining requests based on the results that were returned. In simpler words, it allows user to focus the search on patterns from various different angles.
- **Incorporation of background of knowledge:** The main work of background knowledge is to continue the process of discovery and indicate the patterns or trends that were seen in the process. Background knowledge can also be used to express the patterns or trends observed in brief and precise terms. It can also be represented at different levels of abstraction.
- **Data mining query languages and ad hoc data mining:** Data Mining Query language is responsible for giving access to the user such that it describes ad hoc mining tasks as well and it needs to be integrated with a data warehouse query language.
- **Presentation and visualization of data mining results:** In this issue, the patterns or trends that are discovered are to be rendered in high level languages and visual representations. The representation has to be written so that it is simply understood by everyone.
- **Handling noisy or incomplete data:** For this process, the data cleaning methods are used. It is a convenient way of handling the noise and the incomplete objects in data mining. Without data cleaning methods, there will be no accuracy in the discovered patterns. And then these patterns will be poor in quality.

2. Performance Issues

It has been noticed several times there are performance related issues in data mining as well. These issues are as follows:

- **Efficiency and Scalability of data mining algorithm :** Efficiency and Scalability is very important when it comes to data mining process. It is also very necessary because with the help of using this, the user can withdraw the information from the data in a more effective and productive manner. On top of that, the user can withdraw that information effectively from the large amount of data in various databases.
- **Parallel, distributed and incremental mining algorithm :** There are a lot factors which can be responsible for the development of parallel and distributed algorithms in data mining. These factors are large in size of database, huge distribution of data, and data mining method that are complex. In this process, in the first and foremost step, the algorithm divides the data from database into various partition. In the next step, that data is processed such that it is situated in parallel manner. Then in the last step, the result from the partition is merged.

3. Diverse Data Types Issues

The issues in this category are given below:

- **Handling of relational and complex types of data :** The database may contain the various data objects. For example, complex, multimedia, temporal data, or spatial data objects. It is very difficult to mine all these data with the help of a single system.
- **Mining information from heterogeneous databases and global information systems :** The problem in this kind of issue is to mine the knowledge from various data sources. These data are not available as a single source instead these data are available at the different data sources on LAN or WAN. The structures of these data are different as well.

7.10 APPLICATIONS OF DATA MINING

The importance of data mining and analysis is growing day by day in our real life. Today most organizations use data mining for analysis of Big Data. Let us see how this technology benefit different users.

1. Mobile Service Providers

- Data mining is used by mobile service providers to create marketing campaigns and keep clients from switching providers.
- Data mining techniques can forecast "churn" from a huge amount of data, such as billing information, email, text messages, web data transmissions, and customer service, and provide a probability score based on the results.

- Customers who are at a higher risk of churning can then be offered incentives or special offers by mobile service providers. Major service companies, such as broadband, phone, and gas suppliers, frequently use this type of mining.

2. Retail Sector

- Data mining aids supermarket and retail sector owners in understanding customer preferences. Customers' purchasing preferences are revealed via data mining technologies based on their purchase history.
- These findings are used by supermarkets to plan product placement on shelves and to introduce special offers such as coupons for matching products and special discounts on specific items.
- RFM grouping is used in these campaigns. Recency, frequency, and monetary grouping are all abbreviated as RFM. These segments have their own promotions and marketing activities. The customer who spends a lot but very less frequently will be treated differently from the customer who buys every 2-3 days but of less amount.
- Data Mining can be used for product recommendation and cross-referencing of items.

3. Artificial Intelligence

- Relevant patterns are fed into a system to make it artificially intelligent. These patterns are the result of data mining. Data mining techniques are used to assess the significance of the artificially intelligent systems' outputs.
- When customers interact with machines, recommender systems use data mining techniques to produce customized recommendations. Artificial intelligence is employed on mined data to make product recommendations based on a customer's previous purchase history such as in Amazon.

4. E-commerce

- Many E-commerce sites use data mining to offer cross-selling and upselling of their products. The shopping sites such as Amazon, Flipkart show “People also viewed”, “Frequently bought together” to the customers who are interacting with the site.
- These recommendations are provided using data mining over the purchasing history of the customers of the website.

5. Science and Engineering

- With the advent of data mining, scientific applications are now moving from statistical techniques to using “collect and store data” techniques, and then perform mining on new data, output new results and experiment with the process. A large amount of data is collected from scientific domains such as astronomy, geology, satellite sensors, global positioning system, etc.

- Data mining in computer science helps to monitor system status, improve its performance, find out software bugs, discover plagiarism and find out faults. Data mining also helps in analyzing the user feedback regarding products, articles to deduce opinions and sentiments of the views.

6. Crime Prevention

- Data Mining detects outliers across a vast amount of data. The criminal data includes all details of the crime that has happened. Data Mining will study the patterns and trends and predict future events with better accuracy.
- The agencies can find out which area is more prone to crime, how much police personnel should be deployed, which age group should be targeted, vehicle numbers to be scrutinized, etc.

7. Research

- Researchers use Data Mining tools to explore the associations between the parameters under research such as environmental conditions like air pollution and the spread of diseases like asthma among people in targeted regions.

8. Farming

- Farmers use Data Mining to find out the yield of vegetables with the amount of water required by the plants.

9. Automation

- By using data mining, the computer systems learn to recognize patterns among the parameters which are under comparison. The system will store the patterns that will be useful in the future to achieve business goals. This learning is automation as it helps in meeting the targets through machine learning.

10. Dynamic Pricing

- Data mining helps the service providers such as cab services to dynamically charge the customers based on the demand and supply. It is one of the key factors for the success of companies.

11. Transportation

- Data Mining helps in scheduling the moving of vehicles from warehouses to outlets and analyze the product loading patterns.

12. Insurance

- Data mining methods help in forecasting the customers who buy the policies, analyze the medical claims that are used together, find out fraudulent behavior and risky customers.

7.11 BENEFITS OF DATA MINING

- Data mining technique helps companies to get knowledge-based information.
- Data mining helps organizations to make the profitable adjustments in operation and production.
- The data mining is a cost-effective and efficient solution compared to other statistical data applications.
- Data mining helps with the decision-making process.
- Facilitates automated prediction of trends and behaviors as well as automated discovery of hidden patterns.
- It can be implemented in new systems as well as existing platforms
- It is the speedy process which makes it easy for the users to analyze huge amount of data in less time.

7.12 DISADVANTAGES OF DATA MINING

- There are chances of companies may sell useful information of their customers to other companies for money. For example, American Express has sold credit card purchases of their customers to the other companies.
- Many data mining analytics software is difficult to operate and requires advance training to work on.
- Different data mining tools work in different manners due to different algorithms employed in their design. Therefore, the selection of correct data mining tool is a very difficult task.
- The data mining techniques are not accurate, and so it can cause serious consequences in certain conditions.

7.13 SUMMARY

- Necessity is the mother of invention. Data mining meets the urgent need for effective, scalable, and adaptable data analysis in our society as data grows in every application. Data mining is a natural evolution of information technology that brings together a variety of disciplines and application fields.
- The technique of extracting interesting patterns from large volumes of data is known as data mining. Data cleansing, data integration, data selection, data transformation, pattern identification, pattern evaluation, and knowledge presentation are all common steps in the knowledge discovery process.
- Data mining has many applications in various industries, such as health informatics, finance, and digital libraries.

- There are many challenges involved in data mining research. Understanding these issues and overcoming them will help you formulate effective mining techniques.

7.14 TEST YOUR SKILLS

Q. 1.1 Which of the following is the data mining task?

- (a) Registering of an online course
- (b) Online money payment through a bank
- (c) Predicting if a student will pass an online course
- (d) Downloading the course certificate from the website

Q. 1.2 _____ performs data mining tasks.

- (a) Knowledge base
- (b) Data mining engine
- (c) Pattern evaluation module
- (d) Data warehouse

Q. 1.3 What is KDD in data mining?

- (a) Knowledge Discovery Database
- (b) Knowledge Discovery Data
- (c) Knowledge Data Definition
- (d) Knowledge Data Discovery

Q. 1.4 In KDD and data mining, noise is referred to as _____.

- (a) Complex Data (b) Meta Data
- (c) Error (d) Repeated Data

Q. 1.5 Which of the following is an essential process in which the intelligent methods are applied to extract data patterns?

- (a) Warehousing (b) Data Mining
- (c) Text Mining (d) Data Selection

7.15 DESCRIPTIVE QUESTIONS

- [1] What is data mining? Describe the steps involved in data mining when viewed as a process of knowledge discovery.
- [2] Explain the architecture of data mining.
- [3] Compare and contrast data warehousing with data mining.
- [4] State the applications of data mining.
- [5] Explain the issues in data mining.

7.16 REFERENCE FOR FURTHER READING

- [1] Data Mining: Introductory and Advanced Topics, Dunham, Margaret H, Prentice Hall (2006)
- [2] Data Mining: Practical Machine Learning Tools and Techniques, Second Edition, Witten, Ian and Eibe Frank, Morgan Kaufmann (2011)
- [3] Data Mining: Concepts and Techniques, The Morgan Kaufmann Series in Data Management Systems, Han J. and Kamber M. Morgan Kaufmann Publishers, (2000).

munotes.in

ASSOCIATION ANALYSIS

Unit Structure

- 8.0 Objectives
- 8.1 Market Basket Analysis
- 8.2 Frequent Itemsets, closed Itemsets and Association Rule
- 8.3 Frequent Pattern Mining
- 8.4 Apriori Algorithm
- 8.5 Improving Efficiency of Apriori Algorithm
- 8.6 Mining Multilevel Association Rule
- 8.7 Mining Multidimensional Association Rule
- 8.8 Hash Based Apriori Algorithm
- 8.9 FP-Growth Algorithm
- 8.10 Summary
- 8.11 Test Your Skills
- 8.12 Descriptive Questions
- 8.13 Reference for Further Reading

8.0 OBJECTIVES

- To understand the fundamentals of market-basket analysis.
- To learn Apriori Algorithm for frequent itemset mining with candidate key generation.
- To understand mining of multilevel and multidimensional association rules.
- To learn FP-Growth Algorithm for frequent itemset mining without candidate key generation.

8.1 MARKET BASKET ANALYSIS

- Market Basket Analysis is a data mining technique that is used to uncover purchase patterns in any retail setting.
- The goal of Market Basket Analysis is to understand consumer behavior by identifying relationships between the items that people buy.
- It which identifies the strength of association between pairs of products purchased together and identify patterns of co-occurrence. A co-occurrence is when two or more things take place together.
- Market Basket Analysis creates *If-Then* scenario rules, for example, if item A is purchased then item B is likely to be purchased.
- For example, people who buy green tea are also likely to buy honey. So Market Basket Analysis would quantitatively establish that there

is a relationship between Green Tea and Honey. The same goes for bread, butter, and jam.

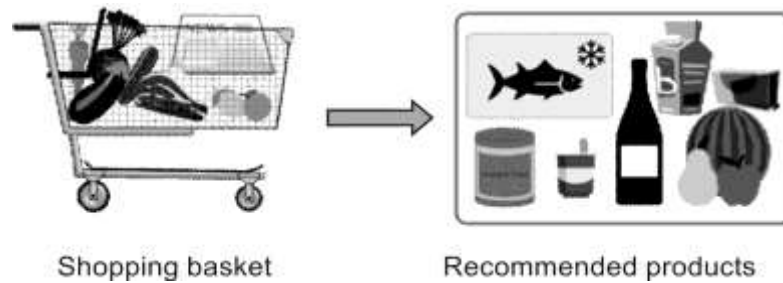


Fig. 8.1.1: Market Basket Analysis

- The rules are probabilistic in nature or, in other words, they are derived from the frequencies of co-occurrence in the observations. Frequency is the proportion of baskets that contain the items of interest.
- The technique determines relationships of what products were purchased with which other product(s). These relationships are then used to build profiles containing If-Then rules of the items purchased.
- The rules could be written as: If {A} Then {B}
- The *If* part of the rule (the {A} above) is known as the antecedent and the *THEN* part of the rule is known as the consequent (the {B} above).
- The antecedent is the condition and the consequent is the result.
- The association rule has three measures that express the degree of confidence in the rule. They are Support, Confidence, Lift and Conviction.
- The **support** is the number of transactions that include items in the {A} and {B} parts of the rule as a percentage of the total number of transactions. It is a measure of how frequently the collection of items occur together as a percentage of all transactions. Also called the **occurrence frequency, frequency, support count, or count**.

$$\text{Support (A} \rightarrow \text{B)} = \frac{\text{Number of transactions containing both A and B}}{\text{Total number of transactions}}$$

- The **confidence** of the rule is the ratio of the number of transactions that include all items in {B} as well as the number of transactions that include all items in {A} to the number of transactions that include all items in {A}.

$$\text{Confidence (A} \rightarrow \text{B)} = \frac{\text{Number of transactions containing both A and B}}{\text{Number of transactions containing A}}$$

- The **lift or lift ratio** is the ratio of confidence to expected confidence. Expected confidence is the confidence divided by the frequency of B. The Lift tells us how much better a rule is at predicting the result than just assuming the result in the first place. Greater lift values indicate stronger associations.

$$\text{Lift (A} \rightarrow \text{B)} = \frac{\frac{\text{Number of transactions containing both A and B}}{\text{Number of transactions containing A}}}{\frac{\text{Number of transactions containing B}}{\text{Total number of transactions}}}$$

- **Conviction** measures the implication strength of the rule from statistical independence. Conviction is defined as,

$$\text{Conviction (A} \rightarrow \text{B)} = \frac{1 - \text{Support(B)}}{1 - \text{Confidence(A} \rightarrow \text{B)}} = \frac{P(A)P(\bar{B})}{P(A \cup \bar{B})}$$

Conviction compares the probability that A appears without B if they were dependent with the actual frequency of the appearance of A without B. Unlike confidence, conviction factors in both $P(A)$ and $P(B)$ and always has a value of 1 when the relevant items are completely unrelated. In contrast to lift, conviction is a directed measure because it also uses the information of the absence of the consequent. Hence, conviction is monotone in confidence and lift.

- **Example :** Consider there are nine baskets containing varying combinations of milk, cheese, apples, and bananas.

Basket	Product 1	Product 2	Product 3
1	Milk	Cheese	
2	Milk	Apples	Cheese
3	Apples	Banana	
4	Milk	Cheese	
5	Apples	Banana	
6	Milk	Cheese	Banana
7	Milk	Cheese	
8	Cheese	Banana	
9	Cheese	Milk	

$$\frac{\text{Support (Milk} \rightarrow \text{Cheese)}}{\text{Number of baskets containing both Milk and Cheese}} = \frac{6}{9} = 0.67$$

$$\frac{\text{Confidence (Milk} \rightarrow \text{Cheese)}}{\text{Number of baskets containing both Milk and Cheese}} = \frac{6}{6} = 1.00$$

$$\text{Lift (Milk} \rightarrow \text{Cheese)} = \frac{\frac{\text{Number of baskets containing both Milk and Cheese}}{\text{Number of baskets containing Milk}}}{\frac{\text{Number of baskets containing Cheese}}{\text{Total number of baskets}}} = \frac{\frac{6}{6}}{\frac{7}{9}} = 1.29$$

$$\text{Conviction (Milk} \rightarrow \text{Cheese)} = \frac{1 - \text{Support}(\text{Cheese})}{1 - \text{Confidence}(\text{Milk} \rightarrow \text{Cheese})}$$

$$= \frac{1 - 7}{1 - 1} = \infty$$

8.1. 1 Applications of Market Basket Analysis

- **Retail** : In Retail, Market Basket Analysis can help determine what items are purchased together, purchased sequentially, and purchased by season. This can assist retailers to determine product placement and promotion optimization (for instance, combining product incentives). Does it make sense to sell soda and chips or soda and crackers?
- **Telecommunications** : In Telecommunications, where high churn rates continue to be a growing concern, Market Basket Analysis can be used to determine what services are being utilized and what packages customers are purchasing. They can use that knowledge to direct marketing efforts at customers who are more likely to follow the same path. For instance, Telecommunications these days is also offering TV and Internet. Creating bundles for purchases can be determined from an analysis of what customers' purchase, thereby giving the company an idea of how to price the bundles. This analysis might also lead to determining the capacity requirements.
- **Banks** : In Financial (banking for instance), Market Basket Analysis can be used to analyze credit card purchases of customers to build profiles for fraud detection purposes and cross-selling opportunities.
- **Insurance** : In Insurance, Market Basket Analysis can be used to build profiles to detect medical insurance claim fraud. By building profiles of claims, you are able to then use the profiles to determine if more than 1 claim belongs to a particular claimer within a specified period of time.
- **Medical** : In Healthcare or Medical, Market Basket Analysis can be used for comorbid conditions and symptom analysis, with which a profile of illness can be better identified. It can also be used to reveal biologically relevant associations between different genes or between environmental effects and gene expression.

8.2 FREQUENT ITEMSETS, CLOSED ITEMSETS AND ASSOCIATION RULE

8.2.1 Frequent ItemSets

- A set of items together is called an **itemset**.
- If any itemset has k-items it is called a **k-itemset**.
- The set {Milk, Cheese} is a 2-itemset.
- An itemset consists of two or more items.
- The occurrence frequency of an itemset is the number of transactions that contain the itemset.
- An itemset that occurs frequently is called a frequent itemset.
- A set of items is called **frequent** if it satisfies a minimum threshold value for support and confidence.

8.2.2 Closed Itemsets

- An itemset is **closed** if none of its immediate supersets have support count same as Itemset.
- An itemset is **closed frequent** itemset if it is both closed and frequent.
- **Identification**
 1. First identify all frequent itemsets.
 2. Then from this group find those that are closed by checking to see if there exists a superset that has the same support as the frequent itemset, if there is, the itemset is disqualified, but if none can be found, the itemset is closed.
- **Maximal frequent** itemsets are the sets S such that no proper superset of S is frequent.
- To illustrate this concept, consider the example given below :

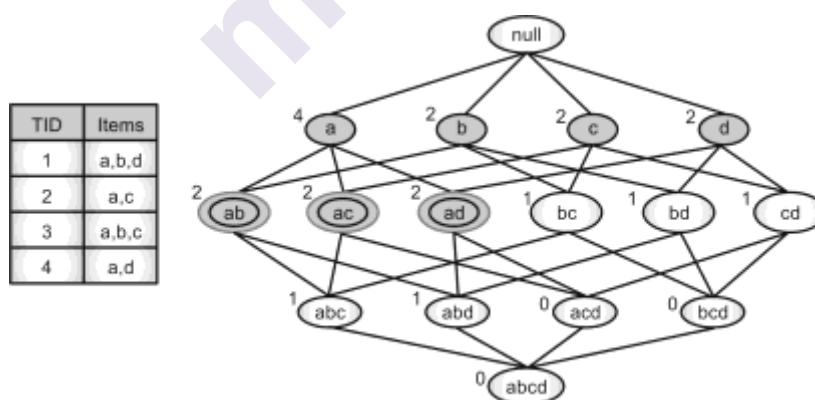


Fig. 8.2.1: Lattice Structure for Closed, Maximal and Frequent Itemsets

- The support counts are shown on the top left of each node.
- Assume **support count threshold = 50%**, that is, each item must occur in 2 or more transactions.

- Based on that threshold, the frequent itemsets are: a, b, c, d, ab, ac and ad (shaded nodes).
- Out of these 7 frequent itemsets, 3 are identified as maximal frequent (having double circles):
 - i. ab: Immediate supersets abc and abd are infrequent.
 - ii. ac: Immediate supersets abc and acd are infrequent.
 - iii. ad: Immediate supersets abd and acd are infrequent.
- The remaining 4 frequent nodes (a, b, c and d) cannot be maximal frequent because they all have at least 1 immediate superset that is frequent.

8.2.3 Association Rules

- An implication expression of the form $X \rightarrow Y$, where X and Y are any 2 itemsets.
- Example: {Milk, Diaper} \rightarrow {Beer}
- These rules must satisfy the confidence value.

ASSOCIATION RULE GENERATION

- Once the frequent itemsets from transactions in the database D are found, we can generate strong association rules from them.
- Strong association rules satisfy both minimum support and minimum confidence.

$$\text{Confidence } (A \rightarrow B) = \frac{\text{Support_count } (A \cup B)}{\text{Support_count } (A)}$$

8.3 FREQUENT PATTERN MATCHING

Frequent pattern mining is classified in the various ways based on following criteria:

1. Completeness of the pattern 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. Levels of abstraction involved in the rule set

- Some methods for association rule mining can find rules at differing levels of abstraction.
- For example, suppose that a set of association rules mined includes the following rules where X is a variable representing a customer:

buys(X, "computer") \rightarrow buys(X, "Canon Printer")
... (1)

buys(X, "laptop computer") \rightarrow buys(X, "Canon Printer")
... (2)

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

3. 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**.

$\text{buys}(X, \text{“computer”}) \rightarrow \text{buys}(X, \text{“Canon printer”})$

- If a rule references two or more dimensions, such as age, income, and buys, then it is a **multidimensional association rule**. The following rule is an example of a multidimensional rule:

$\text{age}(X, \text{“30,31...39”}) \wedge \text{income}(X, \text{“42K,...48K”}) \rightarrow \text{buys}(X, \text{“Apple Smartphone”})$

4. Types of valued 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. Kinds of rules to be mined

- Frequent pattern analysis can generate various kinds of rules and other interesting relationships.
- Association rule mining can generate a large number of rules, many of which are redundant or do not indicate a correlation relationship among itemsets.
- The discovered associations can be further analyzed to uncover statistical correlations, leading to correlation rules.

6. 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 sub-sequences 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 laptop, followed by a smartphone, and then a smartwatch.
- **Structured pattern mining** searches for frequent substructures in a structured data set.
- Single items are the simplest form of structure. Each element of an itemset may contain a subsequence, a subtree, and so on.
- Therefore, structured pattern mining can be considered as the most general form of frequent pattern mining.

7. Application domain-specific semantics

- Because of the huge diversity in data and applications, the patterns to be mined can differ largely based on their domain-specific semantics.
- Application data can be of different types like spatial data, temporal data, spatiotemporal data, multimedia data, text data, time-series data, DNA and biological sequences, software programs, web structures, sensor network data, social and information networks data, and so on.

8. Data analysis usages

- For improved data understanding, pattern-based classification and pattern-based clustering can be used for semantic annotation or contextual analysis.
- Pattern analysis can be also used in recommender systems, which recommend items that are likely to be of interest to the user based on user's patterns.

8.4 APRIORI ALGORITHM: MINING FREQUENT ITEMSETS WITH CANDIDATE KEY GENERATION

- Apriori algorithm was the first algorithm that was proposed for frequent itemset mining.
- It was later improved by R Agarwal and R Srikant and came to be known as Apriori.
- It is an iterative approach to discover the most frequent itemsets.
- This algorithm uses two steps “join” and “prune” to reduce the search space.
- **Join Step** : This step generates (K+1) itemset from K-itemsets by joining each item with itself.
- **Prune Step** : This step scans the count of each item in the database. If the candidate item does not meet minimum support, then it is regarded as infrequent and thus it is removed. This step is performed to reduce the size of the candidate itemsets.

8.4.1 Steps in Apriori Algorithm

Apriori algorithm is a sequence of steps to be followed to find the most frequent itemset in the given database. This data mining technique follows the join and the prune steps iteratively until the most frequent itemset is achieved. A minimum support threshold is given in the problem or it is assumed by the user.

1. In the first iteration of the algorithm, each item is taken as a 1-itemsets candidate. The algorithm will count the occurrences of each item.
2. Let there be some minimum support, min_sup (e.g. 2). The set of 1-itemsets whose occurrence is satisfying the min_sup are determined. Only those candidates which count more than or equal to min_sup, are taken ahead for the next iteration and the others are pruned.

3. Next, 2-itemset frequent items with min_sup are discovered. For this in the join step, the 2-itemset is generated by forming a group of 2 by combining items with itself.
4. The 2-itemset candidates are pruned using min_sup threshold value. Now the table will have 2-itemsets with min_sup only.
5. The next iteration will form 3-itemsets using join and prune step. This iteration will follow antimonotone property where the subsets of 3-itemsets, that is the 2-itemset subsets of each group fall in min_sup . If all 2-itemset subsets are frequent then the superset will be frequent otherwise it is pruned.
6. Next step will follow making 4-itemset by joining 3-itemset with itself and pruning if its subset does not meet the min_sup criteria. The algorithm is stopped when the most frequent itemset is achieved.

8.4.2 Apriori Algorithm given by Jiawei Han, Micheline Kamber and Jian Pei

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 :

- (1) $L_1 = \text{find_frequent_1-itemsets}(D)$;
- (2) for ($k = 2$; $L_{k-1} \neq \emptyset$; $k++$) {
- (3) $C_k = \text{apriori_gen}(L_{k-1})$;
- (4) for each transaction $t \in D$ { // scan D for counts
- (5) $C_t = \text{subset}(C_k, t)$; // get the subsets of t that are candidates
- (6) for each candidate $c \in C_t$
- (7) $c.\text{count}++$;
- (8) }
- (9) $L_k = \{c \in C_k \mid c.\text{count} \geq \text{min_sup}\}$
- (10) }
- (11) return $L = \cup_k L_k$;

procedure apriori_gen (L_{k-1} :frequent ($k-1$) - itemsets)

- (1) for each itemset $l_1 \in L_{k-1}$
- (2) for each itemset $l_2 \in L_{k-1}$
- (3) if $((l_1[1] = l_2[1]) \wedge (l_1[2] = l_2[2]) \wedge \dots \wedge (l_1[k-2] = l_2[k-2]) \wedge (l_1[k-1] < l_2[k-1]))$ then {
- (4) $c = l_1 \bowtie l_2$; // join step: generate candidates
- (5) if $\text{has_infrequent_subset}(c, L_{k-1})$ then
- (6) delete c; // prune step: remove unfruitful candidate
- (7) else add c to C_k
- (8) }

- (9) return C_k ;
procedure has_infrequent_subset(c : candidate k -itemset;
 L_{k-1} : frequent $(k-1)$ -itemsets); // use prior knowledge
 (1) for each $(k-1)$ -subset s of c
 (2) if $s \notin L_{k-1}$ then
 (3) return TRUE;
 (4) return FALSE;

8.4.3 Flowchart for Apriori Algorithm

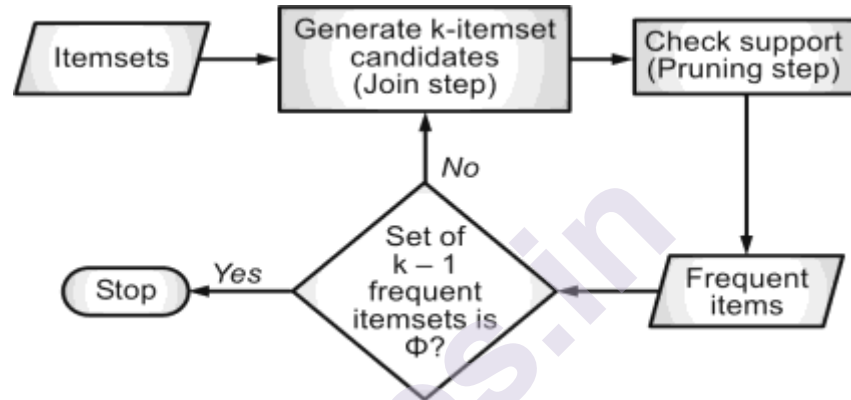


Fig. 8.4.1: Apriori Algorithm Flowchart

8.4.4 Advantages

- Easy to understand algorithm.
- Join and Prune steps are easy to implement on large itemsets in large databases.

8.4.5 Disadvantages

- It requires high computation if the itemsets are very large and the minimum support is kept very low.
- The entire database needs to be scanned.

Ex. 8.4.1 : Given the following data, apply the apriori algorithm. Given **Support threshold=50%, Confidence= 60%.**

Transaction	List of items
T1	I1, I2, I3
T2	I2, I3, I4
T3	I4, I5
T4	I1, I2, I4
T5	I1, I2, I3, I5
T6	I1, I2, I3, I4

Soln. :

Support threshold = 50%

Therefore, min_sup = $0.5 \times \text{number of transactions} = 0.5 \times 6 = 3$

Thus, $\text{min_sup} = 3$

1. **Count of Each Itemset (C_1) by scanning the database.**

Itemset	Count
{I1}	4
{I2}	5
{I3}	4
{I4}	4
{I5}	2

2. **Prune Step (L_1) :** C_1 shows that I5 itemset does not meet $\text{min_sup}=3$, thus it is deleted, only I1, I2, I3, I4 meet min_sup count.

Itemset	Count
{I1}	4
{I2}	5
{I3}	4
{I4}	4

3. **Join Step :** Form C_2 from L_1 using $L_1 \bowtie L_1$ and find out their occurrences.

Itemset	Count
{I1,I2}	4
{I1,I3}	3
{I1,I4}	2
{I2,I3}	4
{I2,I4}	3
{I3,I4}	2

4. **Prune Step (L_2) :** C_2 shows that itemset {I1, I4} and {I3, I4} does not meet min_sup , thus it is deleted.

Itemset	Count
{I1,I2}	4
{I1,I3}	3
{I2,I3}	4
{I2,I4}	3

5. **Join Step** : Form C_3 from L_2 using $L_2 \bowtie L_2$ and find out their occurrences.

Itemset	Count
{I1,I2, I3}	3
{I1,I2, I4}	2
{I1,I3, I4}	1
{I2,I3, I4}	2

6. **Prune Step (L_3)** : C_3 shows that itemset {I1, I2, I4}, {I1, I3, I4} and {I2, I3, I4} does not meet min_sup, thus it is deleted.

Itemset	Count
{I1,I2, I3}	3

Thus {I1, I2, I3} is frequent.

7. **Generate Association Rules**: From the frequent itemset discovered above, the association could be:

- {I1, I2} \rightarrow {I3}
Confidence = support {I1, I2, I3} / support {I1, I2} = (3/ 4) \times 100 = 75%
- {I1, I3} \rightarrow {I2}
Confidence = support {I1, I2, I3} / support {I1, I3} = (3/ 3) \times 100 = 100%
- {I2, I3} \rightarrow {I1}
Confidence = support {I1, I2, I3} / support {I2, I3} = (3/ 4) \times 100 = 75%
- {I1} \rightarrow {I2, I3}
Confidence = support {I1, I2, I3} / support {I1} = (3/ 4) \times 100 = 75%
- {I2} \rightarrow {I1, I3}
Confidence = support {I1, I2, I3} / support {I2} = (3/ 5) \times 100 = 60%
- {I3} \rightarrow {I1, I2}
Confidence = support {I1, I2, I3} / support {I3} = (3/ 4) \times 100 = 75%
- This shows that all the above association rules are strong if minimum confidence threshold is 60%.

Ex. 8.4.2 : A database has four transactions. Let min_sup = 60%, min conf = 80%. Apply Apriori algorithm to find the frequent itemsets and the strong association rules.

Transaction	Date	List of items
T100	10/15/99	{K,A,D,B}
T200	10/15/99	{D,A,C,E,B}
T300	10/19/99	{C,A,B,E}
T400	10/22/99	{B,A,D}

• **Soln. :**

Support threshold = 60%

Therefore, min_sup = $0.6 \times \text{number of transactions} = 0.6 \times 4$
= 2.4

Thus, min_sup = 3

1. **Count of Each Itemset (C₁) by scanning the database.**

Itemset	Count
{A}	4
{B}	4
{C}	2
{D}	3
{E}	2
{K}	1

2. **Prune Step (L₁) :** C₁ shows that C, E, K item does not meet min_sup=3, thus it is deleted, only A, B, D meet min_sup count.

Itemset	Count
{A}	4
{B}	4
{D}	3

3. **Join Step :** Form C₂ from L₁ using L₁ ⋈ L₁ and find out their occurrences.

Itemset	Count
{A,B}	4
{A,D}	3
{B,D}	3

4. **Prune Step (L₂):** C₂ shows that all item meet support count, so nothing is deleted.

Itemset	Count
{A,B}	4
{A,D}	3
{B,D}	3

5. **Join Step:** Form C₃ from L₂ using $L_2 \bowtie L_2$ and find out their occurrences.

Itemset	Count
{A, B, D}	3

6. **Prune Step (L₃) :** C₃ shows that {A, B, D} meet min_sup.

Itemset	Count
{A, B, D}	3

Thus {A, B, D} is frequent.

7. **Generate Association Rules :** From the frequent itemset discovered above, the association could be:

- {A,B} → {D}
Confidence = support {A, B, D} / support {A, B} = (3/ 4) × 100 = 75%
- {A,D} → {B}
Confidence = support {A, B, D} / support {A, D} = (3/ 3) × 100 = 100%
- {B,D} → {A}
Confidence = support {A, B, D} / support {B, D} = (3/ 3) × 100 = 100%
- {A} → {B,D}
Confidence = support {A, B, D} / support {A} = (3/ 4) × 100 = 75%
- {B} → {A,D}
Confidence = support {A, B, D} / support {B} = (3/ 4) × 100 = 75%
- {D} → {A,B}
Confidence = support {A,B,D} / support {D} = (3/ 3) × 100 = 100%
- This shows that association rules {A,D} → {B}, {B,D} → {A} and {D} → {A,B} are strong as they satisfy minimum confidence threshold of 80%.

Ex. 8.4.3: Consider the transaction data given below. Use Apriori Algorithm with min_sup count = 2 and min_confidence = 70% to find all frequent itemsets and strong association rules.

TID	List of Items
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

Soln. :

Given : min_sup = 2

1. Count of Each Itemset (C₁) by scanning the database.

Itemset	Count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

2. Prune Step (L₁) : C₁ shows that I4 itemset does not meet min_sup=3, thus it is deleted, only I1, I2, I3, I5 meet min_sup count.

Itemset	Count
{I1}	6
{I2}	7
{I3}	6
{I4}	2
{I5}	2

3. **Join Step** : Form C_2 from L_1 using $L_1 \bowtie L_1$ and find out their occurrences.

Itemset	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

4. **Prune Step (L_2)** : C_2 shows that itemset {I1, I4}, {I3, I4}, {I3, I5} and {I4, I5} does not meet min_sup, thus it is deleted.

Itemset	Count
{I1,I2}	4
{I1,I3}	4
{I1,I5}	2
{I2,I3}	4
{I2,I4}	2
{I2,I5}	2

5. **Join Step**: Form C_3 from L_2 using $L_2 \bowtie L_2$ and find out their occurrences.

Itemset	Count
{I1,I2,I3}	2
{I1,I2,I5}	2
{I1,I3,I5}	1
{I2,I3,I4}	0
{I2,I3,I5}	1
{I2,I4,I5}	0

6. **Prune Step (L_3)** : C_3 shows that itemset {I1,I3,I5}, {I2,I3,I4}, {I2, I3, I5} and {I2,I4,I5} does not meet min_sup, thus it is deleted.

Itemset	Count
{I1,I2,I3}	2
{I1,I2,I5}	2

7. **Join Step**: Form C_4 from L_3 using $L_3 \bowtie L_3$ and find out their occurrences.

Itemset	Count
{I1,I2,I3,I5}	1

$\{I1, I2, I3, I5\}$ does not meet \min_sup , thus it is deleted. So we move back to step 6 and find that **$\{I1, I2, I3\}$ and $\{I1, I2, I5\}$ is frequent.**

8. **Generate Association Rules :** From the frequent itemset discovered above, the association could be:

$$\{I1, I2\} \rightarrow \{I3\}$$

$$\text{Confidence} = \text{support } \{I1, I2, I3\} / \text{support } \{I1, I2\} = 2/4 \times 100 = 50\%$$

$$\{I1, I3\} \rightarrow \{I2\}$$

$$\text{Confidence} = \text{support } \{I1, I2, I3\} / \text{support } \{I1, I3\} = 2/4 \times 100 = 50\%$$

$$\{I2, I3\} \rightarrow \{I1\}$$

$$\text{Confidence} = \text{support } \{I1, I2, I3\} / \text{support } \{I2, I3\} = 2/4 \times 100 = 50\%$$

$$\{I1, I2\} \rightarrow \{I5\}$$

$$\text{Confidence} = \text{support } \{I1, I2, I5\} / \text{support } \{I1, I2\} = 2/4 \times 100 = 50\%$$

$$\{I1, I5\} \rightarrow \{I2\}$$

$$\text{Confidence} = \text{support } \{I1, I2, I5\} / \text{support } \{I1, I5\} = 2/2 \times 100 = 100\%$$

$$\{I2, I5\} \rightarrow \{I1\}$$

$$\text{Confidence} = \text{support } \{I1, I2, I5\} / \text{support } \{I2, I5\} = 2/2 \times 100 = 100\%$$

$$\{I3\} \rightarrow \{I1, I2\}$$

$$\text{Confidence} = \text{support } \{I1, I2, I3\} / \text{support } \{I1\} = 2/6 \times 100 = 33.33\%$$

$$\{I2\} \rightarrow \{I1, I3\}$$

$$\text{Confidence} = \text{support } \{I1, I2, I3\} / \text{support } \{I2\} = 2/7 \times 100 = 28.57\%$$

$$\{I1\} \rightarrow \{I2, I3\}$$

$$\text{Confidence} = \text{support } \{I1, I2, I3\} / \text{support } \{I1\} = 2/6 \times 100 = 33.33\%$$

$$\{I5\} \rightarrow \{I1, I2\}$$

$$\text{Confidence} = \text{support } \{I1, I2, I5\} / \text{support } \{I5\} = 2/2 \times 100 = 100\%$$

$$\{I2\} \rightarrow \{I1, I5\}$$

$$\text{Confidence} = \text{support } \{I1, I2, I5\} / \text{support } \{I2\} = 2/7 \times 100 = 28.57\%$$

$$\{I1\} \rightarrow \{I2, I5\}$$

$$\text{Confidence} = \text{support } \{I1, I2, I5\} / \text{support } \{I1\} = 2/6 \times 100 = 33.33\%$$

This shows that association rules $\{I1, I5\} \rightarrow \{I2\}$, $\{I2, I5\} \rightarrow \{I1\}$, and $\{I5\} \rightarrow \{I1, I2\}$ are strong as they satisfy minimum confidence threshold of 70%.

8.5 IMPROVING EFFICIENCY OF APRIORI ALGORITHM

Many variations of Apriori algorithm have been proposed that focus on improving the efficiency of the original algorithm. Several of these variations are summarized as follows:

1. **Hash-Based Technique** : This method uses a hash-based structure called a hash table for generating the k-itemsets and its corresponding count. It uses a hash function for generating the table.
2. **Transaction Reduction** : This method reduces the number of transactions scanning in iterations. The transactions which do not contain frequent items are marked or removed.
3. **Partitioning** : This method requires only two database scans to mine the frequent itemsets. It says that for any itemset to be potentially frequent in the database, it should be frequent in at least one of the partitions of the database.
4. **Sampling** : This method picks a random sample S from Database D and then searches for frequent itemset in S. It may be possible to lose a global frequent itemset. This can be reduced by lowering the min_sup.

8.6 MINING MULTILEVEL ASSOCIATION RULES

- Association rules generated from mining data at multiple levels of abstraction are called 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 approach is used, with counts accumulated for the computation of frequent itemsets at each concept level, starting at concept level 1 and continuing down the hierarchy toward more detailed concept levels until no more itemsets can be discovered.
- For each level, any algorithm for discovering frequent itemsets may be used, such as Apriori or its variations.

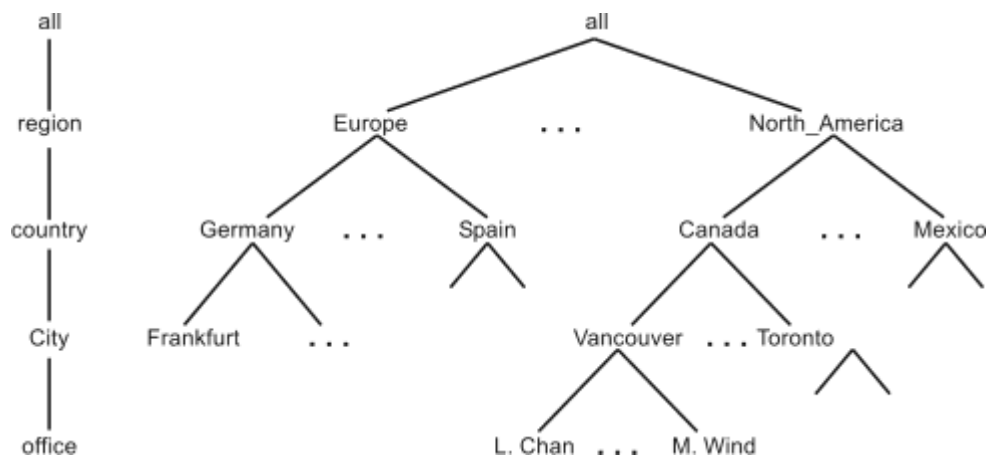


Fig. 8.6.1: Concept Hierarchy

- There are different variations to this approach, where each variation involves “playing” with support threshold in a slightly different way.

8.6.1 Support and Confidence of Multilevel Association Rules

- Generalizing / specializing values of attributes affects support and confidence of an item.
- Support of rules increases from specialized to generalized itemsets.
- Support of rules decreases from generalized to specialized itemsets.
- Confidence is not affected for general or specialized.
- If the support is below the threshold value, then that rule becomes invalid.

8.6.2 Approaches of Multilevel Association Rules

1. Using uniform support level for all levels

- Consider the same minimum support for all levels of hierarchy.
- There is only one minimum support threshold, so no need to examine itemsets.
- If support threshold is too high, then low level associations may get missed.
- If the support threshold is too low, it may generate too many high level associations.

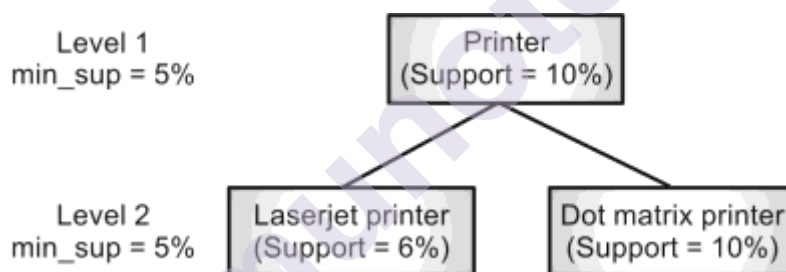


Fig. 8.6.2: Multilevel Mining with Uniform Support

2. Using reduced minimum support at lower level

- Consider separate minimum support for all levels of hierarchy.
- At every level of abstraction, there is its own minimum support threshold; So minimum support at lower levels reduces.

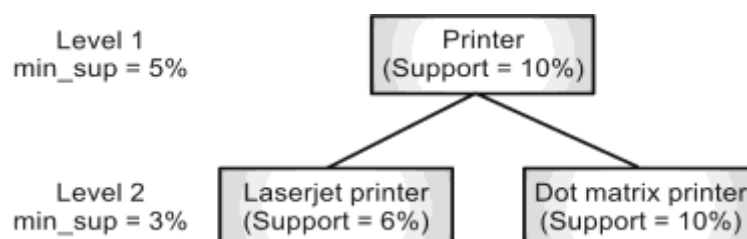


Fig. 8.6.3: Multilevel Mining with Reduced Support

- For mining multiple-level associations with reduced support, there are a number of alternative search strategies:
 - (a) **Level-by-Level independent:** This is a full-breadth search, where no background knowledge of frequent itemsets is used for pruning. Each node is examined, regardless of whether or not its parent node is found to be frequent.
 - (b) **Level -cross-filtering by single item:** An item at the i^{th} level is examined if and only if its parent node at the $(i - 1)^{\text{th}}$ level is frequent. In other words, we investigate a more specific association from a more general one. If a node is frequent, its children will be examined; otherwise, its descendants are pruned from the search.
 - (c) **Level-cross filtering by k-itemset:** A k-itemset at the i^{th} level is examined if and only if its corresponding parent k-itemset at the $(i - 1)^{\text{th}}$ level is frequent.
- 3. **Using item or group-based minimum support**
 - When mining multilevel rules, it's sometimes preferable to build up user-specific, item-based, or group-based minimal support criteria because users or experts sometimes have insight into which groups are more significant than others.
 - For example, a user could establish minimal support criteria based on product pricing or items of interest, such as setting extremely low support thresholds for laptop computers and flash drives to focus on association patterns comprising products from these categories.

8.7 MINING MULTIDIMENSIONAL ASSOCIATION RULES

Following are the terminologies used in multidimensional database.

- **Single – dimension rules :** It contains the single distinct predicate like “buys” in the example given.
 $\text{buys}(X, \text{“milk”}) \rightarrow \text{buys}(X, \text{“bread”})$
- **Multi-dimensional rule :** It contains more than one predicate
 1. **Inter-dimension association rule:** It has no repeated predicate
 $\text{age}(X, \text{“19-25”}) \wedge \text{occupation}(X, \text{“student”}) \rightarrow \text{buys}(X, \text{“coke”})$.
 2. **Hybrid dimension association rules:** It contains multiple occurrence of the same predicate like “buys” in the below example.
 $\text{age}(X, \text{“19-25”}) \wedge \text{buys}(X, \text{“popcorn”}) \rightarrow \text{buys}(X, \text{“coke”})$
- **Categorical Attributes :** This have finite number of possible values, no ordering among values. Example: brand, color.
- **Quantitative Attributes :** These are numeric and implicit ordering among values Example; age, income.

8.7.1 Techniques for Mining Multidimensional Associations

- Database attributes can be categorical or quantitative.
- Categorical attributes have a finite number of possible values, with no ordering among the values.
- Quantitative attributes are numeric and have an implicit ordering among values.
- Techniques for mining multidimensional association rules can be categorized into two basic approaches regarding the treatment of quantitative attributes:

(i) Static Discretization of Quantitative Attributes

- Quantitative attributes are discretized using predefined concept hierarchies in this method. This discretization takes place prior to mining.
- For instance, a concept hierarchy for income may be used to replace the original numeric values of this attribute by interval labels, such as “0.....20K”, “21K.....30K”, “31K.....40K”, and so on. Here, discretization is static and predetermined.
- The discretized numeric attributes, with their interval labels, can then be treated as categorical attributes (where each interval is considered a category).
- We refer to this as mining multidimensional association rules using static discretization of quantitative attributes.

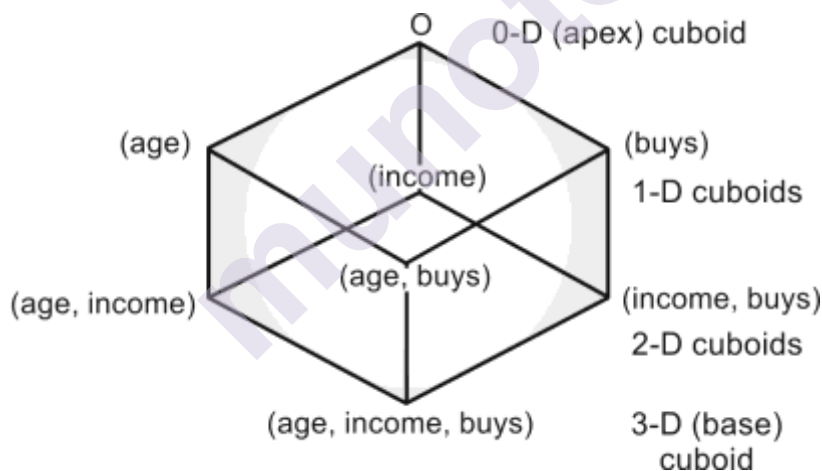


Fig. 8.7.1 : Lattice of cuboids to form a 3-D data cube

(ii) Dynamic Quantitative Association Rules

- In this approach, quantitative attributes are discretized or clustered into "bins" based on the distribution of the data.
- These bins may be further combined during the mining process.
- The discretization process is dynamic and set up to meet certain mining criteria, such as increasing the confidence in the rules mined.
- Association rules derived from this procedure are referred to as (dynamic) quantitative association rules since the numeric

attribute values are treated as quantities rather than predetermined ranges or categories.

- The strong association rules obtained are mapped a 2-D grid as shown.

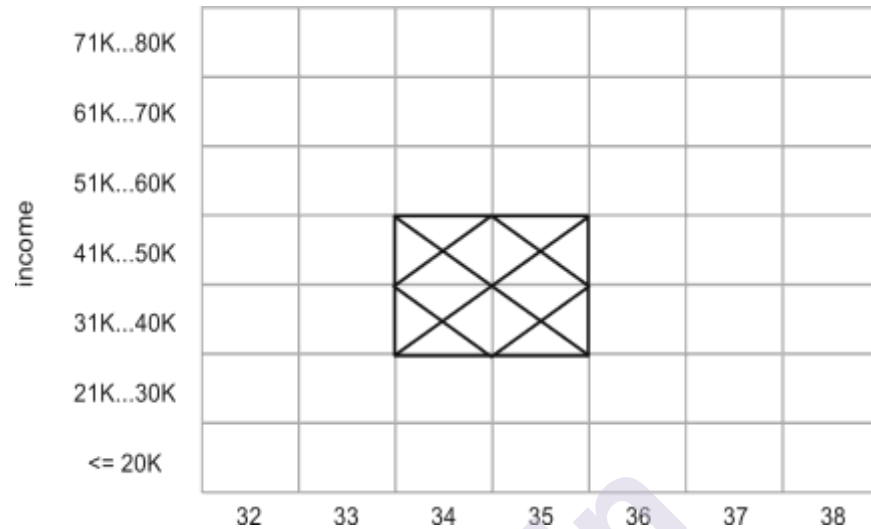


Fig. 8.7.2 : A 2-D grid for tuples representing customers purchasing SUV

- Following four customers correspond to the rules:
 $\text{age}(X, 34) \wedge \text{income}(X, "30k-40K") \rightarrow \text{buys}(X, "SUV")$
 $\text{age}(X, 35) \wedge \text{income}(X, "30k-40K") \rightarrow \text{buys}(X, "SUV")$
 $\text{age}(X, 34) \wedge \text{income}(X, "40k-50K") \rightarrow \text{buys}(X, "SUV")$
 $\text{age}(X, 35) \wedge \text{income}(X, "40k-50K") \rightarrow \text{buys}(X, "SUV")$
- Above rules are close to each other, they can be clustered to form the following rule:
 $\text{age}(X, "34-35") \wedge \text{income}(X, "30k-50K") \rightarrow \text{buys}(X, "SUV")$

8.8 HASH BASED APRIORI ALGORITHM

- Hash based Apriori implementation, uses a data structure that directly represents a hash table.
- This algorithm overcoming some of the weaknesses of the Apriori algorithm by reducing the number of candidate k-itemsets. In particular, the 2-itemsets, since that is the key to improving performance.
- This algorithm uses a hash based technique to reduce the number of candidate itemsets generated in the first pass.
- It is claimed that the number of itemsets in C_2 generated using hashing can be reduced, so that the scan required to determine L_2 is more efficient.

- For example, when scanning each transaction in the database to generate the frequent 1-itemsets, L_1 , from the candidate 1-itemsets in C_1 , we can generate all of the 2-itemsets for each transaction, hash (i.e. map) them into the different buckets of a hash table structure, and increase the corresponding bucket counts.
- A 2-itemset whose corresponding bucket count in the hash table is below the support threshold cannot be frequent and thus should be removed from the candidate set. Such a hash based Apriori may substantially reduce the number of the candidate k-itemsets examined.

8.8.1 Algorithm

1. Scan all the transaction. Create possible 2-itemsets.
2. Let the Hash table of size 8.
3. For each bucket assign a candidate pairs using the ASCII values of the itemsets.
4. Each bucket in the hash table has a count, which is increased by 1 each item an item set is hashed to that bucket.
5. If the bucket count is equal or above the minimum support count, the bit vector is set to 1. Otherwise it is set to 0.
6. The candidate pairs that hash to locations where the bit vector bit is not set are removed.
7. Modify the transaction database to include only these candidate pairs.
 - In this algorithm, each transaction counting all the 1-itemsets.
 - At the same time all the possible 2-itemsets in the current transaction are hashed to a hash table. It uses a hash table to reduce the number of candidate itemsets.
 - When the support count is established the algorithm determines the frequent itemsets. It generates the candidate itemsets as like the Apriori algorithm.

8.9 FREQUENT PATTERN GROWTH OR FP-GROWTH ALGORITHM: MINING FREQUENT ITEMSETS WITHOUT CANDIDATE KEY GENERATION

- This algorithm is an improvement to the Apriori method.
- A frequent pattern is generated without the need for candidate generation.
- FP growth algorithm represents the database in the form of a tree called a frequent pattern tree or FP tree.
- This tree structure will maintain the association between the itemsets.
- The database is fragmented using one frequent item. This fragmented part is called “pattern fragment”. The itemsets of these fragmented patterns are analyzed.

- Thus with this method, the search for frequent itemsets is reduced comparatively.

8.9.1 FP Tree

- Frequent Pattern Tree is a tree-like structure that is made with the initial itemsets of the database.
- The purpose of the FP tree is to mine the most frequent pattern.
- Each node of the FP tree represents an item of the itemset.
- The root node represents null while the lower nodes represent the itemsets.
- The association of the nodes with the lower nodes i.e. the itemsets with the other itemsets are maintained while forming the tree.

8.9.2 Steps in FP Growth Algorithm

1. The first step is to scan the database to find the occurrences of the itemsets in the database. This step is the same as the first step of Apriori. The count of 1-itemsets in the database is called support count or frequency of 1-itemset.
2. The second step is to construct the FP tree. For this, create the root of the tree. The root is represented by null.
3. The next step is to scan the database again and examine the transactions. Examine the first transaction and find out the itemset in it. The itemset with the max count is taken at the top, the next itemset with lower count and so on. It means that the branch of the tree is constructed with transaction itemsets in descending order of count.
4. The next transaction in the database is examined. The itemsets are ordered in descending order of count. If any itemset of this transaction is already present in another branch (for example in the 1st transaction), then this transaction branch would share a common prefix to the root. This means that the common itemset is linked to the new node of another itemset in this transaction.
5. Also, the count of the itemset is incremented as it occurs in the transactions. Both the common node and new node count is increased by 1 as they are created and linked according to transactions.
6. The next step is to mine the created FP Tree. For this, the lowest node is examined first along with the links of the lowest nodes. The lowest node represents the frequency pattern length 1. From this, traverse the path in the FP Tree. This path or paths are called a conditional pattern base. Conditional pattern base is a sub-database consisting of prefix paths in the FP tree occurring with the lowest node (suffix).
7. Construct a Conditional FP Tree, which is formed by a count of itemsets in the path. The itemsets meeting the threshold support are considered in the Conditional FP Tree.
8. Frequent Patterns are generated from the Conditional FP Tree.

8.9.3 FP-Growth Algorithm by Jiawei Han, Micheline Kamber and Jian Pei

Algorithm : FP growth. Mine frequent itemsets using an FP-tree by pattern fragment growth.

Input

D, a transaction database;

min_sup, the minimum support count threshold.

Output

The complete set of frequent patterns.

Method

1. The FP-tree is constructed in the following steps:
 - (a) Scan the transaction database D once. Collect F, the set of frequent items, and their support counts. Sort F in support count descending order as L, the list of frequent items.
 - (b) Create the root of an FP-tree, and label it as “null.” For each transaction Trans in D do the following.

Select and sort the frequent items in Trans according to the order of L. Let the sorted frequent item list in Trans be [p|P], where p is the first element and P is the remaining list.

Call **insert_tree([p|P], T)**, which is performed as follows.

If T has a child N such that N.item-name = p.item-name, then increment N's count by 1; else create a new node N, and let its count be 1, its parent link be linked to T, and its node-link to the nodes with the same item-name via the node-link structure.

If P is nonempty, call **insert_tree(P, N)** recursively.

2. The FP-tree is mined by calling **FP_growth(FP tree, null)**, which is implemented as follows.

procedure **FP_growth (Tree, α)**

- (1) **if** Tree contains a single path P **then**
- (2) **for each** combination (denoted as β) of the nodes in the path P
- (3) generate pattern $\beta \cup \alpha$ with support_count = minimum support count of nodes in β ;
- (4) **else for each** a_i in the header of Tree {
- (5) generate pattern $\beta = a_i \cup \alpha$ with support_count = a_i .support_count;
- (6) construct β 's conditional pattern base and then β 's conditional FP_tree Tree β ;
- (7) **if** Tree $\beta = \phi$ **then**
- (8) call **FP_growth(Tree β , β)**;

8.9.4 Advantages of FP Growth Algorithm

1. This algorithm needs to scan the database only twice when compared to Apriori which scans the transactions for each iteration.
2. The pairing of items is not done in this algorithm and this makes it faster.
3. The database is stored in a compact version in memory.

4. It is efficient and scalable for mining both long and short frequent patterns.

8.9.5 Disadvantages of FP Growth Algorithm

1. FP Tree is more cumbersome and difficult to build than Apriori.
2. It may be expensive.
3. When the database is large, the algorithm may not fit in the shared memory.

Ex. 8.9.1 : A database has five transactions. Let $\text{min_sup} = 60\%$ and $\text{min_conf} = 80\%$. Find all the frequent itemsets using FP-growth.

TID	Items_bought
T100	{M,O,N,K,E,Y}
T200	{D,O,N,K,E,Y}
T300	{M,A,K,E}
T400	{M,U,C,K,Y}
T500	{C,O,O,K,I,E}

Soln. :

Given : $\text{min_sup} = 60\%$

$\therefore \text{sup_count to be satisfied} = 5 \times 0.6 = 3$

Step 1: Scan the database for count of each itemset.

Itemset	sup_count
{A}	1
{C}	2
{D}	1
{E}	4
{I}	1
{K}	5
{M}	3
{N}	2
{O}	4
{U}	1
{Y}	3

Step 2 : Sort the set of frequent itemsets in the order of descending support count and denote that lists as L.

L :

Itemset	sup_count
{K}	5
{E}	4
{O}	4
{M}	3
{Y}	3

Step 3 : Scan the database for second time and sort items in each transaction according to descending support count.

TID	List of Items
T100	{K,E,M,O,Y}
T200	{K,E,O,Y}
T300	{K,E,M}
T400	{K,M,Y}
T500	{K,E,O}

Step 4 : Construct the FP-tree.

4.1 Create a root node with label “NULL”.



Fig. P. 8.9.1(a)

4.2 Scan T100 and construct branch with nodes K:1, E:1, M:1, O:1, Y:1 linked to each other from root node.

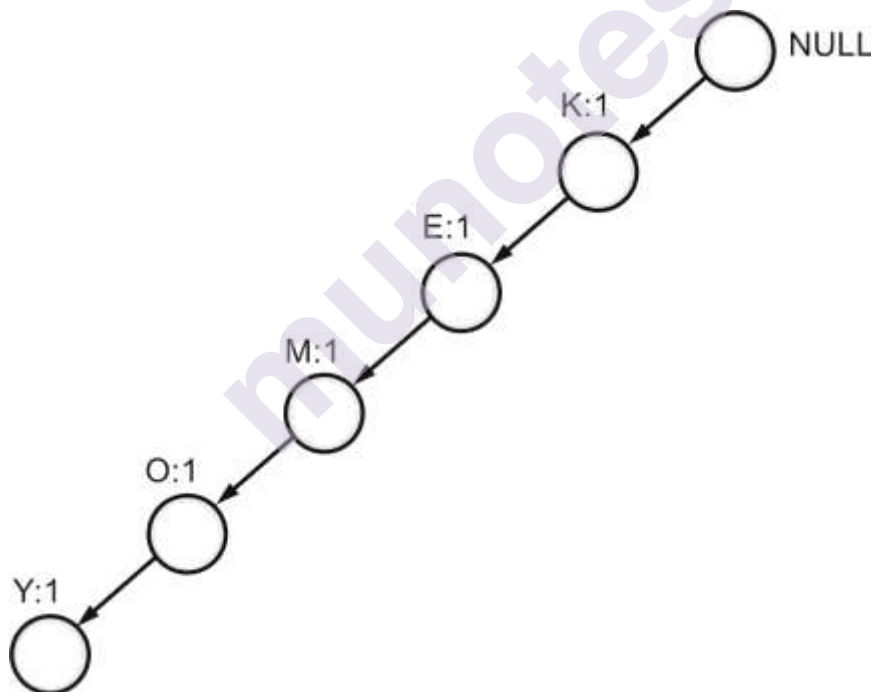


Fig. P. 8.9.1(b)

4.3 Scan T200. It contains itemsets K, E, O, Y in L-order. Nodes K and E already exists. Increment their count as K:2, E: 2 and make a branch for O:1 and Y:1 from E:2.

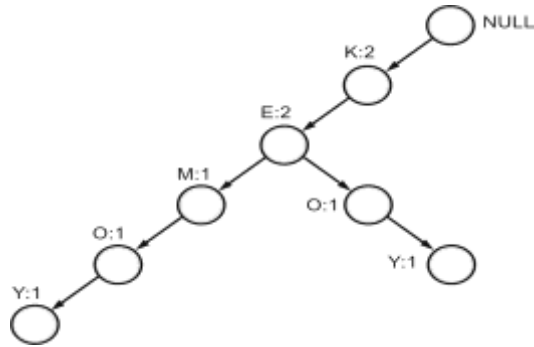


Fig. P. 8.9.1(c)

- 4.4 Scan T300. It contains itemsets K, E, M in L-order. Branch with nodes K, E and M already exists. Increment their count as K:3, E:3 and M:2.

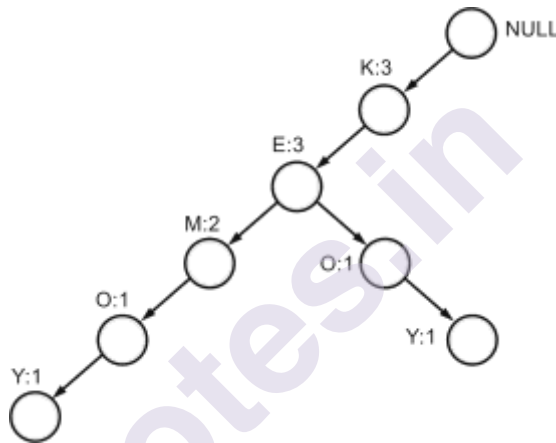


Fig. P. 8.9.1(d)

- 4.5 Scan T400. It contains itemsets K, M, Y in L-order. Node K already exists. Increment its count by 1 as K:4 and make branch for M:1, Y:1 from K:4.

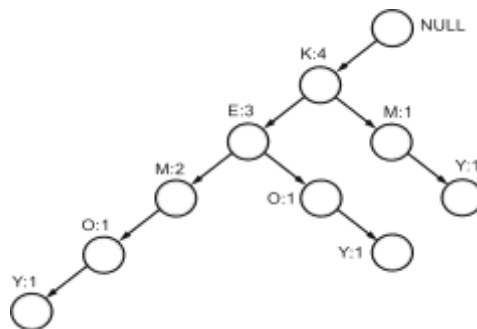


Fig. P. 8.9.1(e)

- 4.6 Scan T500. It contains itemsets K, E, O in L-order. Branch with nodes K, E and O exists. Just increment their count.

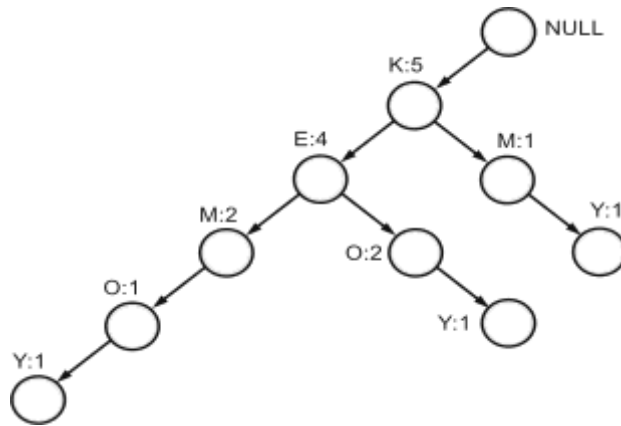


Fig. P. 8.9.1(f)

4.7 Now also connect all the similar nodes.

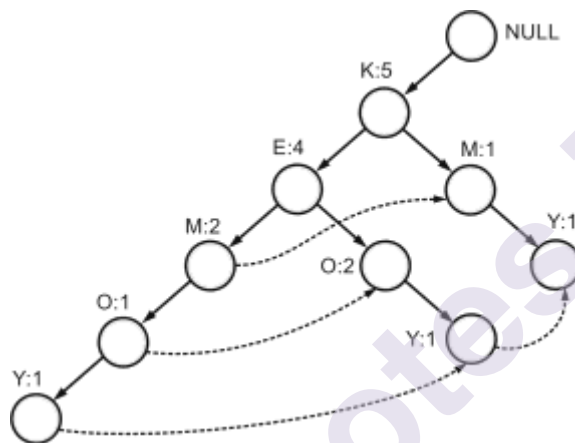


Fig. 8.9.1(g)

Step 5 : Mining FP-tree.

Start from each frequent length-1 pattern, construct its conditional pattern base, then construct its conditional FP-tree, and perform mining recursively on the tree. Start with the last itemset in L.

Note : For generating frequent patterns, consider the items which satisfy $\text{min_sup} = 3$ (given) criteria from conditional FP-tree.

Itemset	Conditional Pattern base	Conditional FP-tree	Frequent Patterns Generated
{Y}	{{K,E,M,O:1},{K,E,O:1},{K,M:1}}	(K:3)	{K,Y:3}
{O}	{{K,E,M:1}, {K,E:2}}	(K:3,E:3)	{K,O:3}, {E,O:3}, {K,E,O:3}
{M}	{{K,E:2},{K:1}}	(K:3)	{K,M:3}
{E}	{{K:4}}	(K:4)	{K,E:4}
{K}	-	-	-

8.10 SUMMARY

- The discovery of patterns and associations among massive amounts of data can be used for various applications. One example is market basket analysis, which is a study of buying habits.
- The process of mining association rules begins with the discovery of frequent itemsets (sets of items, such as A and B, that satisfy a minimum support threshold, or percentage of the task-relevant tuples), from which strong association rules in the form of $A \rightarrow B$ are generated. These rules also meet a minimum confidence level (a pre-specified probability of satisfying B under the condition that A is satisfied).
- Many efficient and scalable algorithms for frequent itemset mining have been developed, from which association rules can be derived. These algorithms can be divided into two types: (1) Apriori-like algorithms; (2) algorithms based on frequent pattern growth, such as FP-growth.

8.11 TEST YOUR SKILLS

Q.2.1 A collection of one or more items is called as ____.

- (a) Itemset (b) Support (c) Confidence
(d) Support Count

Q.2.2 How do you calculate Confidence ($A \rightarrow B$)?

- (a) $\text{Support}(A \cap B) / \text{Support}(A)$
(b) $\text{Support}(A \cap B) / \text{Support}(B)$
(c) $\text{Support}(A \cup B) / \text{Support}(A)$
(d) $\text{Support}(A \cup B) / \text{Support}(B)$

Q.2.3 For the question given below consider the data Transactions:

1. I1, I2, I3, I4, I5, I6
2. I7, I2, I3, I4, I5, I6
3. I1, I8, I4, I5
4. I1, I9, I10, I4, I6
5. I10, I2, I4, I11, I5

With support as 0.6 find all frequent itemsets?

- (a) $\langle I1 \rangle, \langle I2 \rangle, \langle I4 \rangle, \langle I5 \rangle, \langle I6 \rangle, \langle I1, I4 \rangle, \langle I2, I4 \rangle, \langle I2, I5 \rangle, \langle I4, I5 \rangle, \langle I4, I6 \rangle, \langle I2, I4, I5 \rangle$
(b) $\langle I2 \rangle, \langle I4 \rangle, \langle I5 \rangle, \langle I2, I4 \rangle, \langle I2, I5 \rangle, \langle I4, I5 \rangle, \langle I2, I4, I5 \rangle$
(c) $\langle I1 \rangle, \langle I4 \rangle, \langle I5 \rangle, \langle I6 \rangle, \langle I1, I4 \rangle, \langle I5, I4 \rangle, \langle I1, I5 \rangle, \langle I4, I6 \rangle, \langle I2, I4, I5 \rangle$
(d) $\langle I1 \rangle, \langle I4 \rangle, \langle I5 \rangle, \langle I6 \rangle$

Q.2.4 If {A, B, C, D} is a frequent itemset, candidate rules which is not possible is

- (a) $C \rightarrow A$ (b) $D \rightarrow ABCD$
 (c) $A \rightarrow BC$ (d) $B \rightarrow ADC$

Q.2.5 Consider the data transactions given below:

- T1: {F, A, D, B}
 T2: {D, A, C, E, B}
 T3: {C, A, B, E}
 T4: {B, A, D}

With minimum support = 60% and the minimum confidence = 80%, which of the following is not valid association rule?

- (a) $A \rightarrow B$ (b) $B \rightarrow A$
 (c) $D \rightarrow A$ (d) $A \rightarrow D$

8.12 DESCRIPTIVE QUESTIONS

- [1] Explain Market Basket Analysis with an example.
- [2] Explain the terms: Frequent Itemsets, Closed Itemsets and Association Rule
- [3] Explain Apriori algorithm with its advantages and disadvantages.
- [4] Explain the techniques to improve efficiency of Apriori Mining.
- [5] Explain FP-Growth algorithm with its advantages and disadvantages.
- [6] Explain multidimensional and multilevel association rules with example.
- [7] Consider the transaction database given in table below. Apply Apriori Algorithm with minimum support of 50% and confidence of 50%. Find all frequent itemsets and all the association rules.

Tid	Items
100	1,3,4
200	2,3,5
300	1,2,3,5
400	2,5
500	1,2,3
600	3,5
700	1,2,3,5
800	1,5
900	1,3

- [8] Consider the transaction database given in table below. Apply Apriori Algorithm with minimum support 2 and confidence of 60%. Find all frequent itemsets and all the association rules.

Tid	Items Bought
1	Milk, Tea, Cake

2	Eggs, Tea, Cold Drink
3	Milk, Eggs, Tea, Cold Drink
4	Eggs, Cold Drink
5	Juice

- [9] A database has 10 transactions. Let $\text{min_sup} = 2$. Find all frequent itemsets using FP-Growth.

Tid	Items
1	{a, b}
2	{b, c, d}
3	{a, c, d, e}
4	{a, d, e}
5	{a, b, c}
6	{a, b, c, d}
7	{a}
8	{a, b, c}
9	{a, b, d}
10	{b, c, e}

8.13 REFERENCE FOR FURTHER READING

- [1] Data Mining: Introductory and Advanced Topics, Dunham, Margaret H, Prentice Hall (2006)
- [2] Data Mining: Practical Machine Learning Tools and Techniques, Second Edition, Witten, Ian and Eibe Frank, Morgan Kaufmann (2011)
- [3] Data Mining: Concepts and Techniques, The Morgan Kaufmann Series in Data Management Systems, Han J. and Kamber M. Morgan Kaufmann Publishers, (2000).
