Table of Contents

Chapter 1: Introduction to Data Mining and Data Warehousing 4

1.1 Review of Basic Concepts of Data Mining and Data Warehousing 4

1.2 Data Mining 5

1.2.1 Why Data Mining? 5

1.2.2 Data mining Functions 6

1.2.3 Data mining Architecture 6

1.2.4 Knowledge Discovery process 7

1.2.5 Applications of Data mining 9

1.2.6 Classification of Data mining system 10

1.2.7 Problem and Challenges of Data Mining 10

1.3 Data Warehouse 12

1.3.1 Features of Data Warehouse 13

1.3.2 Difference between Operational Database Systems and Data Warehouses 13

1.3.3 Why Separate Data Warehouse? 14

1.3.4 Data Warehouse Architecture 15

1.3.5 Data warehouse models 18

1.3.6 Benefits of Data warehousing 18

Chapter 2: Data Warehouse Logical Design 19

2.1 A Multidimensional Data Model 19

2.1.1 From Tables and Spreadsheets to Data Cubes 19

2.1.2 Data warehouse schema 22

2.1.4 Design and construction of Data warehouse 26

2.1.5 Process of Data warehouse design 26

2.1.6 Materialized View (Computation of cuboids) 27

Chapter 3: Data Warehouse Physical Design 29

3.1 Physical Design 29

3.1.1 Physical Design Structure 30

3.2 Hardware and I/O Considerations 30

3.3 Parallelism 31

3.3.1When to parallelise? 31

3.3.2 Benefits of Parallelism......31
3.4 Indexing......31
3.4.1 B-tree indexes......32
3.4.2. Bitmap Index......33
3.4.3 Join Index......35
Chapter 4: Data Warehousing technologies and Implementation......36
4.1 Data preprocessing......36
4.1.1 Data cleaning......36
4.1.2 Data Integration......38
4.1.3 Data Transformation......38
4.1.4 Data Reduction......39
4.2 Data warehouse backend tools (ETL process)......39
4.2.1 Data Extraction......40
4.2.2 Transformation and Cleansing......40
4.2.3 Loading......40
Chapter 5: Data Warehouse to Data Mining......41
5.1 Online Analytical Processing (OLAP)......41
5.1.1 Benefits of OLAP......41
5.2 OLAP OPERATIONS......41
5.3 Types of OLAP Server......44
5.3.1 Relational OLAP (ROLAP)......44
5.3.2 Multidimensional OLAP (MOLAP)......45
5.3.3 Hybrid OLAP (HOLAP)......46
5.3.4 Desktop OLAP (DOLAP)......46
Chapter 6 Data Mining Approaches and Methods......47
6.1 Tasks of Data Mining......47
6.1.1 Descriptive Data Mining......47
6.1.2 Predictive Data Mining......48
6.2 Supervised Vs Unsupervised Learning......48
6.3 Class/Concept Description......48
6.3.1 Data Characterization......49
6.3.2 Data Discrimination......49
6.4 Classification and Prediction......49

6.4.1 Classification 49
6.4.2 Prediction.. 54
6.4.3 Issues regarding classification and prediction 56
6.5 Association Rule Mining.. 57
6.5.1 Apriori Algorithm... 59
6.6 Clustering.. 60
6.6.1 Types of Clustering 60
6.6.2 Applications of Cluster Analysis.. 64
6.7 Data Mining Tools 65
Chapter 7 Mining Complex Types of Data.. 67
7.1 Mining Spatial Data 67
7.2 Multimedia Data Mining.. 69
7.3 Text Mining... 70
7.4 Web Mining... 71
Chapter 8 Research Trends in Data Warehousing and Data Mining 73
8.1 Data Mining Systems Products and Research Prototypes.. 73
8.2 Theoretical Foundations of Data Mining 73
8.3 Statistical Data Mining... 74
8.4 Visual and Audio Data Mining 75
8.5 Data Mining and Collaborative Filtering 76
8.6 Social Impact of Data Mining 76
8.6 Trends in Data Mining 77

Chapter 1: Introduction to Data Mining and Data Warehousing

1.1 Review of Basic Concepts of Data Mining and Data Warehousing

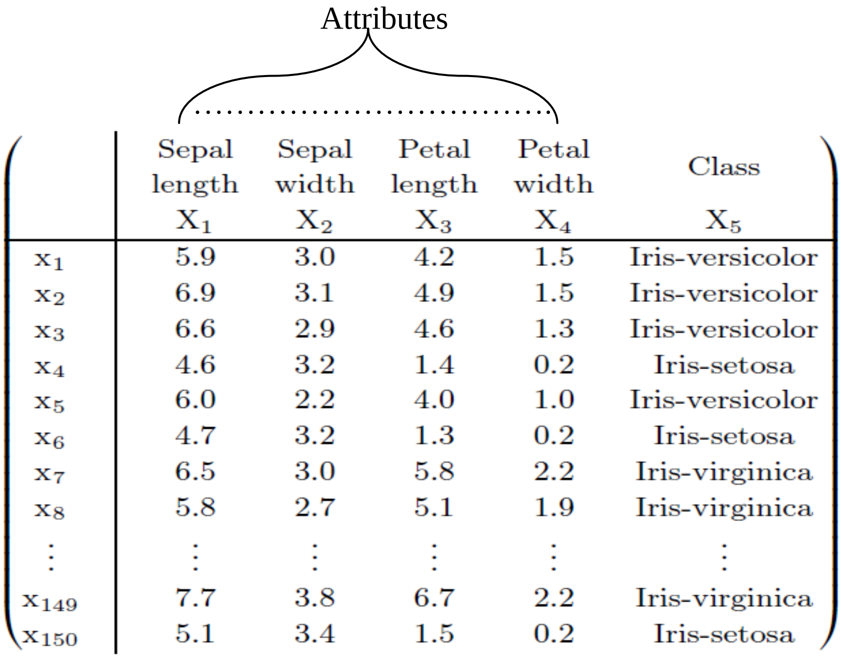
There is a huge amount of data available in the Information Industry. This data is of no use until it is converted into useful information. It is necessary to analyze this huge amount of data and extract useful information from it.

Extraction of information is not the only process we need to perform; data mining also involves other processes such as Data Cleaning, Data Integration, Data Transformation, Data Mining, Pattern Evaluation and Data Presentation. Once all these processes are over, we would be able to use this information in many applications such as Fraud Detection, Market Analysis, Production Control, Science Exploration, etc.

Data

It is a representation of facts, concepts, or instructions in a formal manner suitable for communication, interpretation, or processing by human beings or by computers.

Dataset


Attributes

	Sepal length	Sepal width	Petal length	Petal width	Class
	X_1	X_2	X_3	X_4	X_5
x_1	5.9	3.0	4.2	1.5	Iris-versicolor
x_2	6.9	3.1	4.9	1.5	Iris-versicolor
x_3	6.6	2.9	4.6	1.3	Iris-versicolor
x_4	4.6	3.2	1.4	0.2	Iris-setosa
x_5	6.0	2.2	4.0	1.0	Iris-versicolor
x_6	4.7	3.2	1.3	0.2	Iris-setosa
x_7	6.5	3.0	5.8	2.2	Iris-virginica
x_8	5.8	2.7	5.1	1.9	Iris-virginica
⋮	⋮	⋮	⋮	⋮	⋮
x_{149}	7.7	3.8	6.7	2.2	Iris-virginica
x_{150}	5.1	3.4	1.5	0.2	Iris-setosa

Table 1.1 Extract from the Iris dataset.

Attributes

Attributes refers to the properties of the entity. Attributes may be classified into two main types depending on their domain, that is, depending on the types of values they take on.

a. Numeric Attribute:

A numeric attribute is one that has a real-valued or integer-valued domain. For example, Age with domain(Age) = N, where N denotes the set of natural numbers (non-negative integers), is numeric, and so is petal length in Table 1.1.

b. Categorical Attributes:

A categorical attribute is one that has a set-valued domain composed of a set of symbols. For example, Sex and Education could be categorical attributes with their domains given as

domain(Sex) = {M, F}
domain(Education) = {HighSchool, BS, MS, PhD}

Categorical attributes may be of two types:

I. **Nominal:** The attribute values in the domain are unordered. Eg. domain(Sex) = {M, F}

II. **Ordinal:** The attribute values are ordered. Eg. domain(Education)= {HighSchool, BS, MS, PhD}

1.2 Data Mining

Data mining refers to extracting or "mining" knowledge or information from large amounts of data. The information or knowledge extracted so can be used for any of the following applications −

- Market Analysis
- Fraud Detection
- Customer Retention
- Production Control
- Science Exploration

Data mining is the practice of automatically searching large stores of data to discover patterns and trends that go beyond simple analysis.

Data mining is a art/science of extracting non-trivial, implicit, previously unknown, valuable, and potentially useful information from a large database.

1.2.1 Why Data Mining?

- Data mining helps to turn the huge amount of data into useful information and knowledge that can have different applications.

- Data mining helps in
  a. Automatic discovery of patterns
  b. Prediction of likely outcomes
  c. Creation of actionable information
- Data mining can answer questions that cannot be addressed through simple query and reporting techniques.

1.2.2 Data mining Functions

On the basis of the kind of data to be mined, there are two categories of functions involved in Data Mining −

a. Descriptive

The descriptive function deals with the general properties of data in the database. Here is the list of descriptive functions :

- Class/Concept Description
- Mining of Frequent Patterns
- Mining of Associations
- Mining of Correlations
- Mining of Clusters

b. Classification and Prediction

The list of functions involved in these processes is as follows:

- Classification
- Prediction
- Outlier Analysis
- Evolution Analysis

1.2.3 Data mining Architecture

Architecture of a typical data mining system may have the following major components:

a. **Database, data warehouse, World Wide Web, or other information repository:** This is one or a set of databases, data warehouses, spreadsheets, or other kinds of information repositories.
b. **Database or data warehouse server:** The database or data warehouse server is responsible for fetching the relevant data, based on the user's data mining request.
c. **Knowledge base:** This is the domain knowledge that is used to guide the search or evaluate the interestingness of resulting patterns.
d. **Data mining engine:** This is essential to the data mining system and ideally consists of a set of functional modules for tasks such as characterization, association and

correlation analysis, classification, prediction, cluster analysis, outlier analysis, and evolution analysis.

e. **Pattern evaluation module:** This component typically employs interestingness measures and interacts with the data mining modules so as to focus the search toward interesting patterns.

f. **User interface:** This module communicates between users and the data mining system, allowing the user to interact with the system.

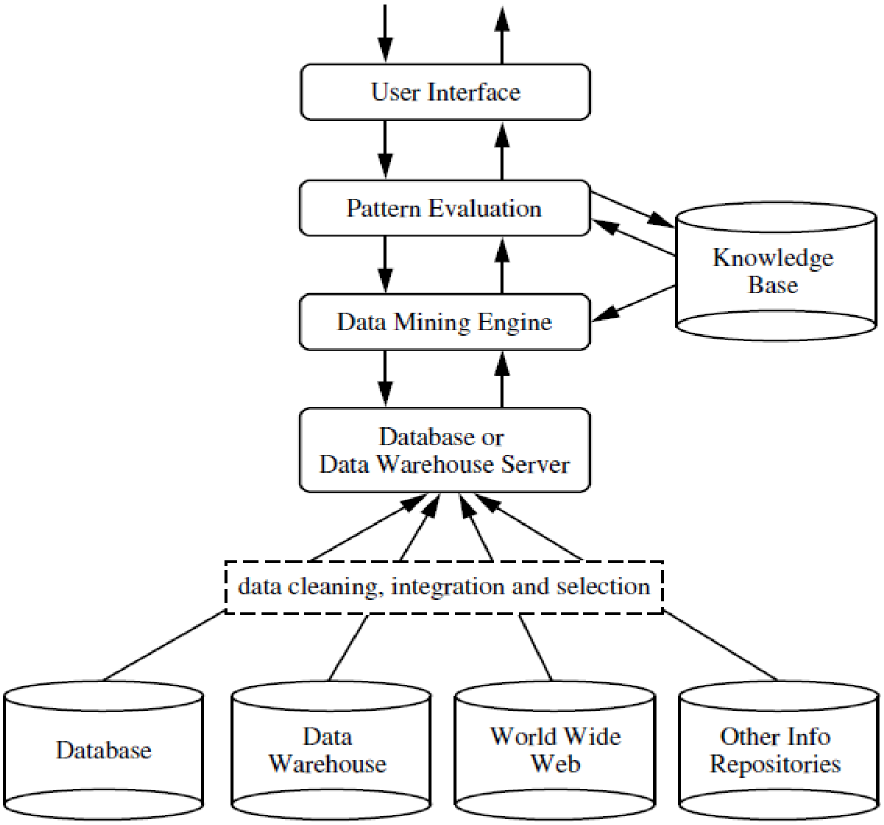


Figure 1.2.3 Architecture of Typical data mining system

1.2.4 Knowledge Discovery process

Data mining is an essential step in the process of knowledge discovery. Knowledge discovery as a process is depicted in Figure 1.2.4 and consists of an iterative sequence of the following steps:

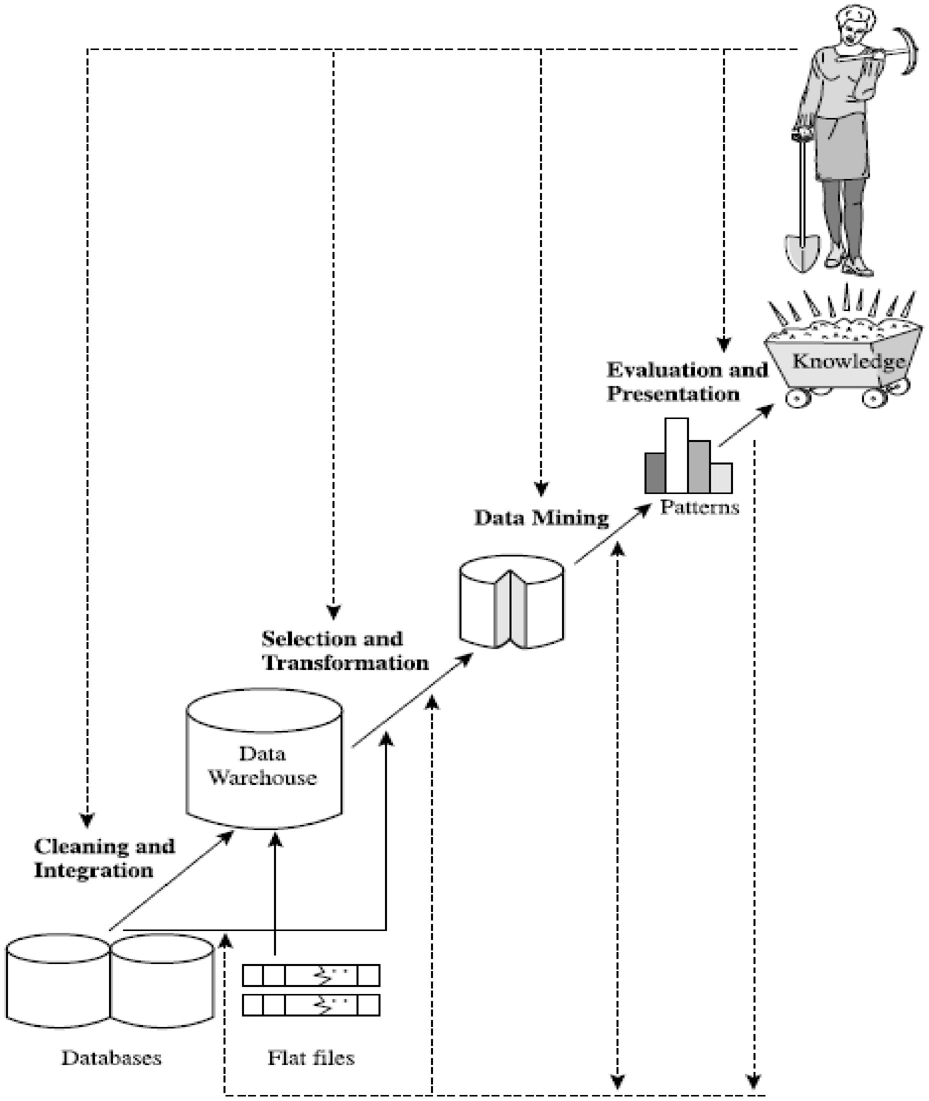


Figure 1.2.4 Data mining as a step in the knowledge discovery process

a. **Data cleaning** (to remove noise and inconsistent data)
a. **Data integration** (where multiple data sources may be combined)
b. **Data selection** (where data relevant to the analysis task are retrieved from the database)
c. **Data transformation** (where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations, for instance)
d. **Data mining** (an essential process where intelligent methods are applied in order to extract data patterns)
e. **Pattern evaluation** (to identify the truly interesting patterns representing knowledge based on some interestingness measures)
f. **Knowledge presentation** (where visualization and knowledge representation techniques are used to present the mined knowledge to the user)

1.2.5 Applications of Data mining

Data mining is highly useful in the following domains −

- Market Analysis and Management
- Corporate Analysis & Risk Management
- Fraud Detection

Apart from these, data mining can also be used in the areas of production control, customer retention, science exploration, sports, astrology, and Internet Web Surf-Aid.

Market Analysis and Management

Listed below are the various fields of market where data mining is used −

- **Customer Profiling** − Data mining helps determine what kind of people buy what kind of products.
- **Identifying Customer Requirements** − Data mining helps in identifying the best products for different customers. It uses prediction to find the factors that may attract new customers.
- **Cross Market Analysis** − Data mining performs association/correlations between product sales.
- **Target Marketing** − Data mining helps to find clusters of model customers who share the same characteristics such as interests, spending habits, income, etc.
- **Determining Customer purchasing pattern** − Data mining helps in determining customer purchasing pattern.
- **Providing Summary Information** − Data mining provides us various multidimensional summary reports.

Corporate Analysis and Risk Management

Data mining is used in the following fields of the Corporate Sector −

- **Finance Planning and Asset Evaluation** − It involves cash flow analysis and prediction, contingent claim analysis to evaluate assets.
- **Resource Planning** − It involves summarizing and comparing the resources and spending.
- **Competition** − It involves monitoring competitors and market directions.

Fraud Detection

- Data mining is also used in the fields of credit card services and telecommunication to detect frauds. In fraud telephone calls, it helps to find the destination of the call, duration of the call, time of the day or week, etc. It also analyzes the patterns that deviate from expected norms.

1.2.6 Classification of Data mining system

Data mining is an interdisciplinary field, the confluence of a set of disciplines, including database systems, statistics, machine learning, visualization, and information science and other disciplines.
Data mining systems can be categorized according to various criteria, as follows:

Classification according to the kinds of databases mined:

Database systems can be classified according to different criteria such as data models, or the types of data or applications involved.

- relational, transactional, object-relational, or data warehouse mining system.
- spatial, time-series, text, stream data, multimedia data mining system, or a World Wide Web mining system

Classification according to the kinds of knowledge mined:

- Data mining systems can be categorized according to the kinds of knowledge they mine, that is, based on data mining functionalities
- Such as characterization, discrimination, association and correlation analysis, classification, prediction, clustering, outlier analysis, and evolution analysis.

Classification according to the kinds of techniques utilized:

- autonomous systems, interactive exploratory systems, query-driven systems
- database-oriented or data warehouse–oriented techniques, machine learning, statistics, visualization, pattern recognition, neural networks, and so on

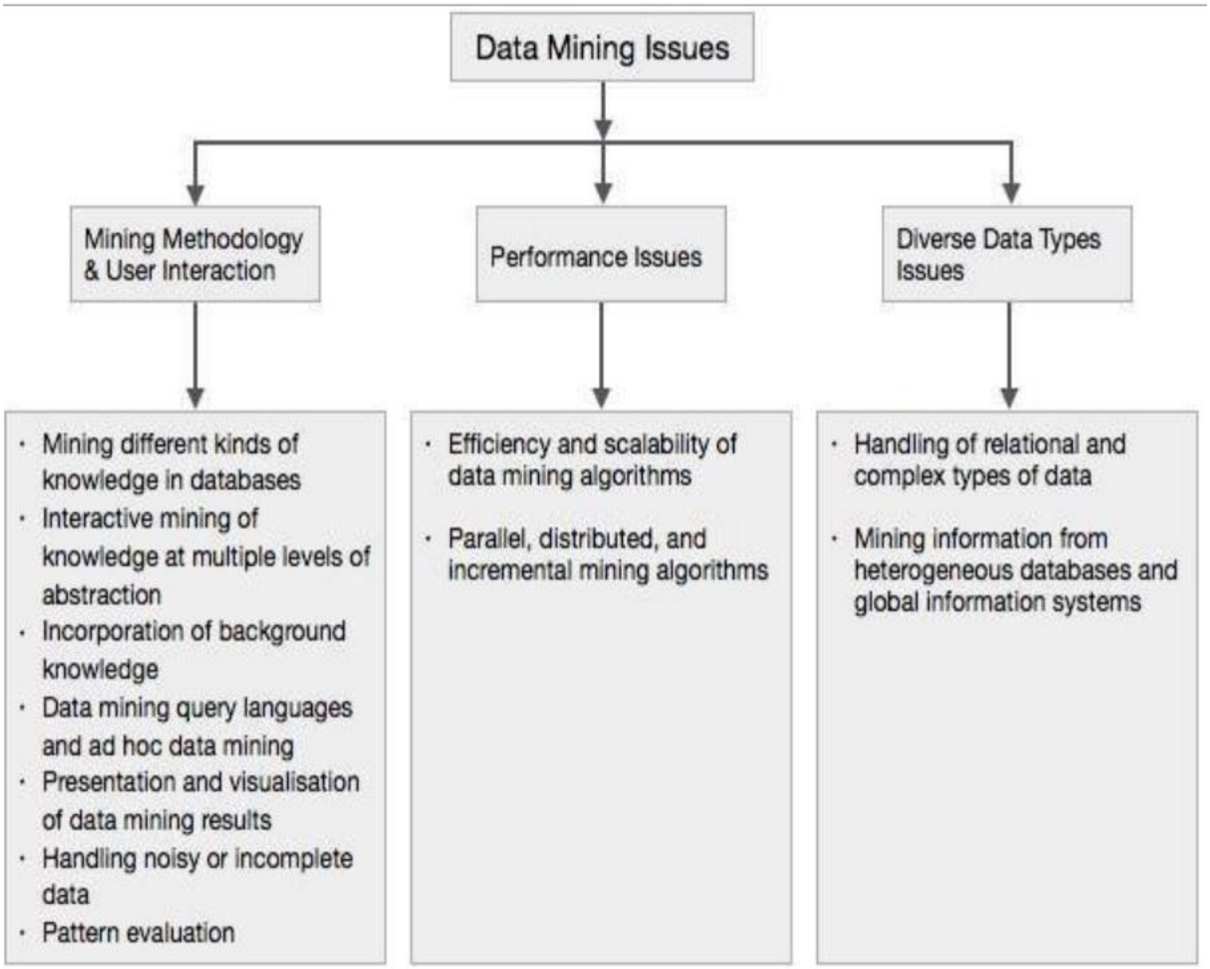
Classification according to the applications adapted:

- Data mining systems may be tailored specifically for finance, telecommunications, DNA, stock markets, e-mail, and so on.
- Different applications often require the integration of application-specific methods.

1.2.7 Problem and Challenges of Data Mining

Data mining is not an easy task, as the algorithms used can get very complex and data is not always available at one place. It needs to be integrated from various heterogeneous data sources. These factors also create some issues.

The following diagram describes the major issues.


Data Mining Issues
Mining Methodology & User Interaction
Performance Issues
Diverse Data Types Issues
Mining different kinds of knowledge in databases
Interactive mining of knowledge at multiple levels of abstraction
Incorporation of background knowledge
Data mining query languages and ad hoc data mining
Presentation and visualisation of data mining results
Handling noisy or incomplete data
Pattern evaluation
Efficiency and scalability of data mining algorithms
Parallel, distributed, and incremental mining algorithms
Handling of relational and complex types of data
Mining information from heterogeneous databases and global information systems

Mining Methodology and User Interaction Issues

It refers to the following kinds of issues −

- **Mining different kinds of knowledge in databases** − Different users may be interested in different kinds of knowledge. Therefore it is necessary for data mining to cover a broad range of knowledge discovery task.
- **Interactive mining of knowledge at multiple levels of abstraction** − The data mining process needs to be interactive because it allows users to focus the search for patterns, providing and refining data mining requests based on the returned results.
- **Incorporation of background knowledge** − To guide discovery process and to express the discovered patterns, the background knowledge can be used. Background knowledge may be used to express the discovered patterns not only in concise terms but at multiple levels of abstraction.
- **Data mining query languages and ad hoc data mining** − Data Mining Query language that allows the user to describe ad hoc mining tasks, should be integrated with a data warehouse query language and optimized for efficient and flexible data mining.

- **Presentation and visualization of data mining results** − Once the patterns are discovered it needs to be expressed in high level languages, and visual representations. These representations should be easily understandable.
- **Handling noisy or incomplete data** − The data cleaning methods are required to handle the noise and incomplete objects while mining the data regularities. If the data cleaning methods are not there then the accuracy of the discovered patterns will be poor.
- **Pattern evaluation** − The patterns discovered should be interesting because either they represent common knowledge or lack novelty.

Performance Issues

There can be performance-related issues such as follows −

- **Efficiency and scalability of data mining algorithms** − In order to effectively extract the information from huge amount of data in databases, data mining algorithm must be efficient and scalable.
- **Parallel, distributed, and incremental mining algorithms** − The factors such as huge size of databases, wide distribution of data, and complexity of data mining methods motivate the development of parallel and distributed data mining algorithms. These algorithms divide the data into partitions which is further processed in a parallel fashion. Then the results from the partitions is merged. The incremental algorithms, update databases without mining the data again from scratch.

Diverse Data Types Issues

- **Handling of relational and complex types of data** − The database may contain complex data objects, multimedia data objects, spatial data, temporal data etc. It is not possible for one system to mine all these kind of data.
- **Mining information from heterogeneous databases and global information systems** − The data is available at different data sources on LAN or WAN. These data source may be structured, semi structured or unstructured. Therefore mining the knowledge from them adds challenges to data mining.

1.3 Data Warehouse

- **Data warehouse is a repository of multiple heterogeneous data sources organized under a unified schema at a single site in order to facilitate management decision making.**
- **According to William H. Inmon, ” A data warehouse is a subject-oriented, integrated, time variant, and non-volatile collection of data in support of management’s decision making process”.**

- The process of constructing and using data warehouses is known as Data warehousing.
- Data warehouses are constructed via a process of data cleaning, data integration, data transformation, data loading, and periodic data refreshing.

1.3.1 Features of Data Warehouse

The key features of Data Warehouse are:

a. **Subject-oriented**
b. **Integrated**
c. **Time-variant**
d. **Non-volatile**

a. Subject-oriented:
Data warehouses typically provide a simple and concise view around particular subject issues by excluding data that are not useful in the decision support process.

b. Integrated:
A data warehouse is usually constructed by integrating multiple heterogeneous sources, such as relational databases, flat files, and on-line transaction records. Data cleaning and data integration techniques are applied to ensure consistency in naming conventions, encoding structures, attribute measures, and so on.

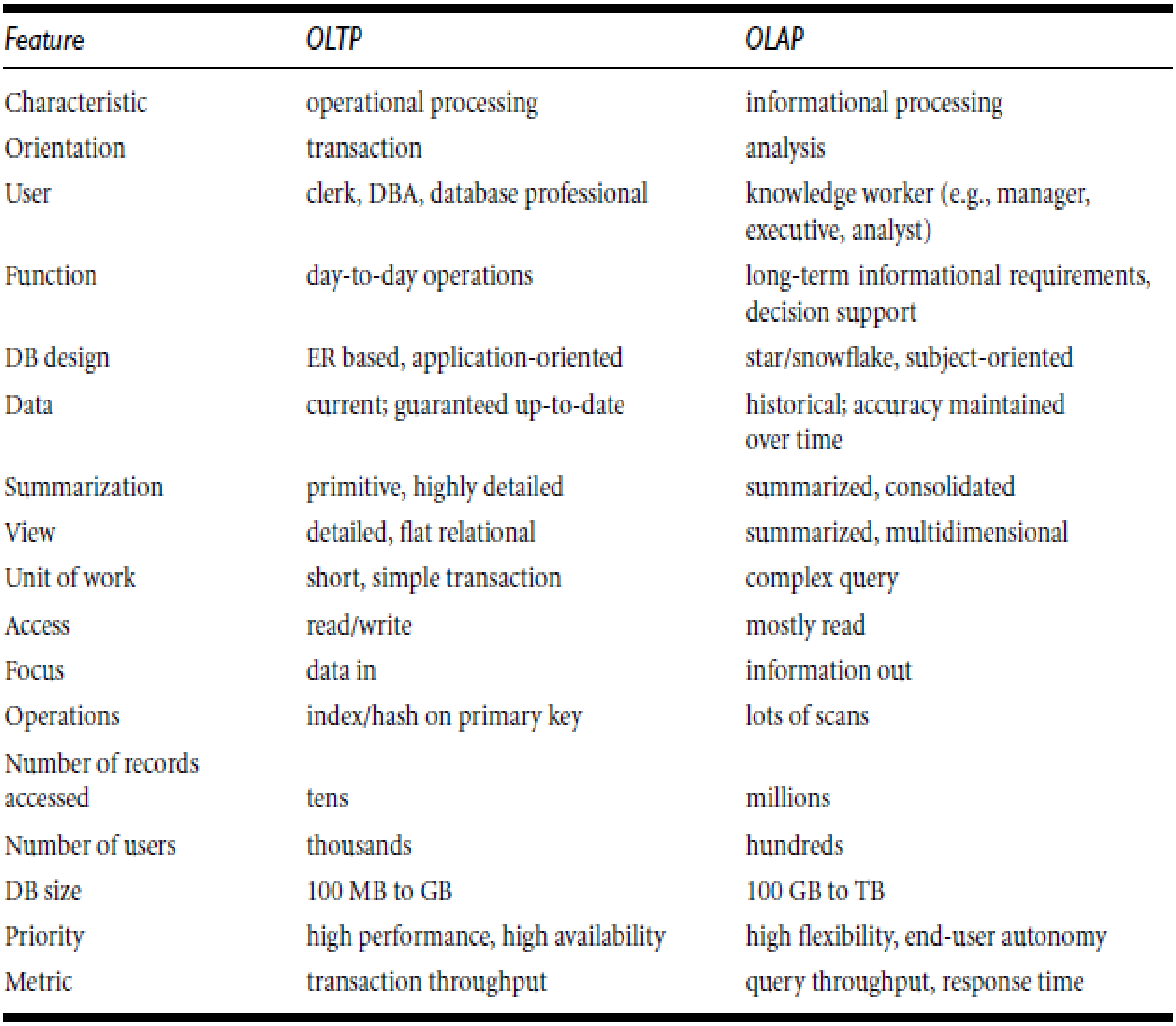
c. Time-variant:
Data are stored to provide information from a historical perspective (e.g., the past 5–10 years).

d. Non-volatile:
Non-volatile means that, once entered into the warehouse, data should not change. This is logical because the purpose of a warehouse is to enable you to analyze what has occurred.

1.3.2 Difference between Operational Database Systems and Data Warehouses

Online transaction processing system (OLTP): OLTP system also known as operational database systems perform online transactions and query processing such as day-to-day operations of an organisations.

Online analytical processing system (OLAP): OLAP system also known as Data warehouse systems performs data analysis and decision making tasks.



Feature	OLTP	OLAP
Characteristic	operational processing	informational processing
Orientation	transaction	analysis
User	clerk, DBA, database professional	knowledge worker (e.g., manager, executive, analyst)
Function	day-to-day operations	long-term informational requirements, decision support
DB design	ER based, application-oriented	star/snowflake, subject-oriented
Data	current; guaranteed up-to-date	historical; accuracy maintained over time
Summarization	primitive, highly detailed	summarized, consolidated
View	detailed, flat relational	summarized, multidimensional
Unit of work	short, simple transaction	complex query
Access	read/write	mostly read
Focus	data in	information out
Operations	index/hash on primary key	lots of scans
Number of records accessed	tens	millions
Number of users	thousands	hundreds
DB size	100 MB to GB	100 GB to TB
Priority	high performance, high availability	high flexibility, end-user autonomy
Metric	transaction throughput	query throughput, response time

Table 2.1.2: Comparison between OLTP and OLAP systems.

1.3.3 Why Separate Data Warehouse?

- Processing OLAP queries in operational databases would substantially degrade the performance of operational tasks.
- An operational database supports the concurrent processing of multiple transactions. Concurrency control and recovery mechanisms, if applied for such OLAP operations, may jeopardize the execution of concurrent transactions and thus substantially reduce the throughput of an OLTP system.
- Decision support requires historical data, whereas operational databases do not typically maintain historical data. In this context, the data in operational databases is not sufficient for decision making process.

1.3.4 Data Warehouse Architecture

Data warehouses often adopt a three-tier architecture, as presented in Figure 1.3.4

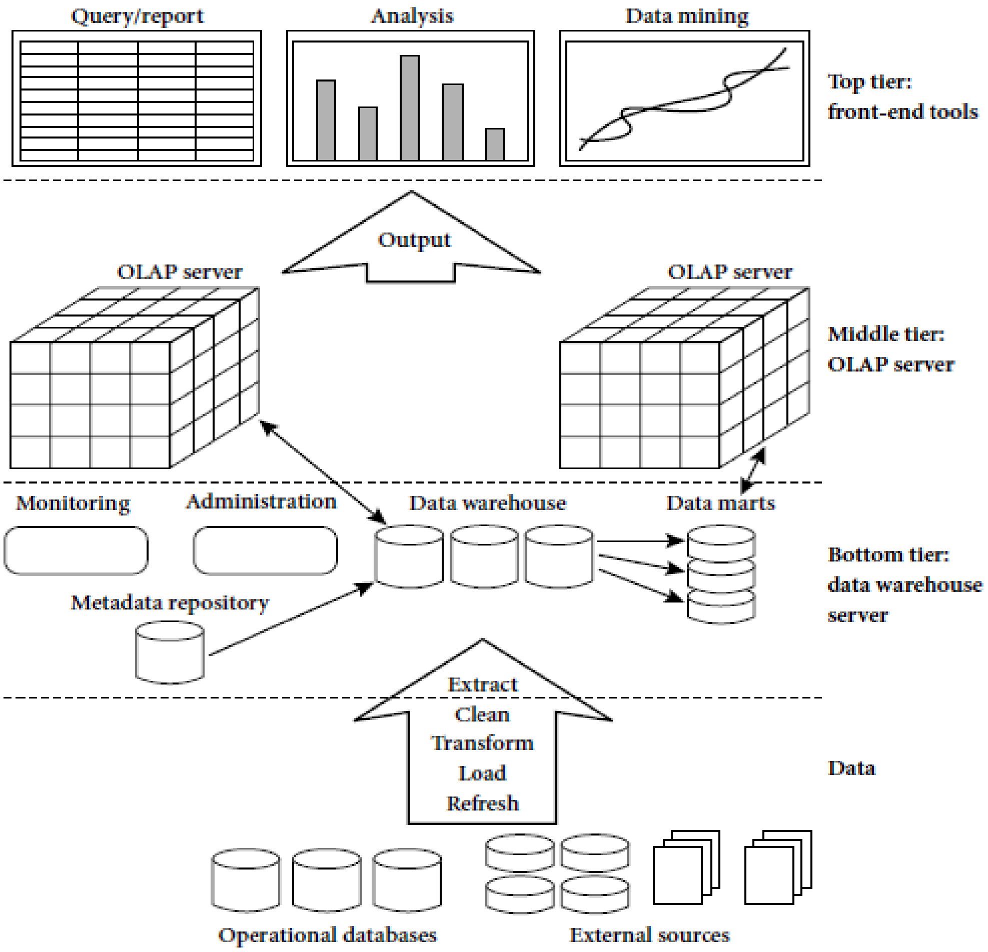
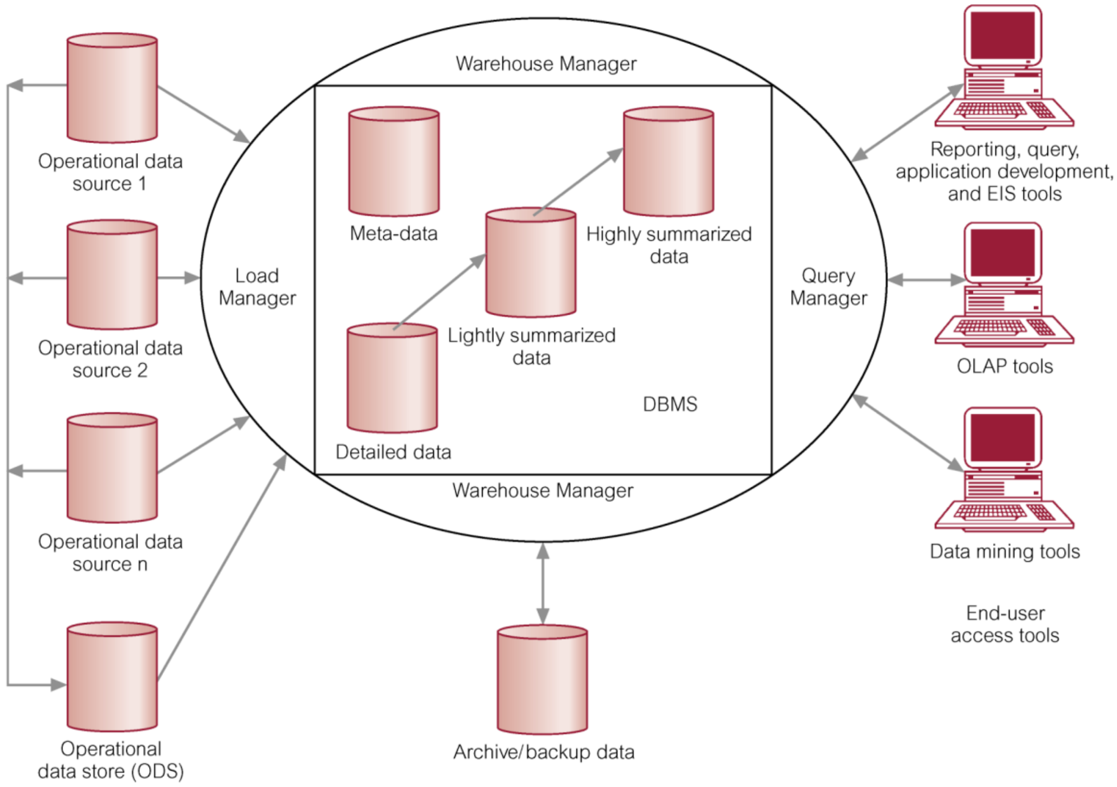


Figure 1.3.4: A three-tier data warehousing architecture.

1. **Data Source:**
   A data warehouse system uses heterogeneous sources of data either from operational databases or from some external sources.
2. **Data warehouse server:**
   Data from heterogeneous sources are stored to one logically centralised single repository: a data warehouse through the extraction, cleaning, transformation, load and refresh functions. Data warehouse can be directly accessed, but it can be also used as a source for creating data marts. Metadata repositories stores information on sources, access procedures, data staging, data mart schema and so on.
3. **OLAP server:**
   OLAP server that is typically implemented using either a relational OLAP (ROLAP) model or a multidimensional OLAP (MOLAP) model.
4. **Front-end client tools:**
   It contains query and reporting tools, analysis tools, and/or data mining tools.

The above architecture can also be shown by following figure:



Load Manager

This component performs the operations required to extract and load process. The size and complexity of the load manager varies between specific solutions from one data warehouse to other. The load manager performs the following functions:

- Extract the data from source system.
- Fast Load the extracted data into temporary data store.
- Perform simple transformations into structure similar to the one in the data warehouse.

Warehouse Manager

A warehouse manager is responsible for the warehouse management process. It consists of third-party system software, C programs, and shell scripts.

It performs the following functions:

- A warehouse manager analyzes the data to perform consistency and referential integrity checks.
- Creates indexes, business views, partition views against the base data.
- Generates new aggregations and updates existing aggregations.
- Transforms and merges the source data into the published data warehouse.

- Backup the data in the data warehouse.
- Archives the data that has reached the end of its captured life.

Query Manager

- Query manager is responsible for directing the queries to the suitable tables.
- By directing the queries to appropriate tables, the speed of querying and response generation can be increased.
- Query manager is responsible for scheduling the execution of the queries posed by the user.

Metadata

Metadata is simply defined as data about data. The data that is used to represent other data is known as metadata. For example, the index of a book serves as a metadata for the contents in the book. In other words, we can say that metadata is the summarized data that leads us to detailed data.

Note: In a data warehouse, we create metadata for the data names and definitions of a given data warehouse. Along with this metadata, additional metadata is also created for time-stamping any extracted data, the source of extracted data.

Categories of Metadata

Metadata can be broadly categorized into three categories:

- **Business Metadata** - It has the data ownership information, business definition, and changing policies.
- **Technical Metadata** - It includes database system names, table and column names and sizes, data types and allowed values. Technical metadata also includes structural information such as primary and foreign key attributes and indices.
- **Operational Metadata** - It includes currency of data and data lineage. Currency of data means whether the data is active, archived, or purged. Lineage of data means the history of data migrated and transformation applied on it.

Role of metadata

The various roles of metadata are explained below.

- Metadata acts as a directory.
- This directory helps the decision support system to locate the contents of the data warehouse.
- Metadata helps in decision support system for mapping of data when data is transformed from operational environment to data warehouse environment.
- Metadata helps in summarization between current detailed data and highly summarized data.
- Metadata also helps in summarization between lightly detailed data and highly summarized data.
- Metadata is used for query tools.

- Metadata is used in reporting tools.
- Metadata is used in extraction and cleansing tools.
- Metadata is used in transformation tools.
- Metadata plays an important role in loading functions.

1.3.5 Data warehouse models

From the architecture point of view, there are three data warehouse models:

a. **Enterprise warehouse**
b. **Data mart**
c. **Virtual warehouse**

a. Enterprise warehouse

An enterprise warehouse collects all of the information about subjects spanning the entire organization. It provides corporate-wide data integration, usually from one or more operational systems or external information providers.

b. Data mart

Data mart is a subset of data warehouse built specifically for department. The scope is confined to specific selected subjects. For example, a marketing data mart may confine its subjects to customer, item, and sales.

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

c. Virtual warehouse

A virtual warehouse is a set of views over operational databases. A virtual warehouse is easy to build but requires excess capacity on operational database servers.

1.3.6 Benefits of Data warehousing

- Queries do not impact Operational systems
- Integrates data from multiple, diverse sources
- Enables multiple interpretations of same data by different users or groups
- Provides thorough analysis of data over a period of time
- Accuracy of Operational systems can be checked
- Provides analysis capabilities to decision makers

Chapter 2: Data Warehouse Logical Design

Logical design is the phase of a database design concerned with identifying the relationships among the data elements. A logical design is conceptual and abstract.

Logical design results in

a. a set of entities and attributes corresponding to fact tables and dimension tables
b. target data warehouse schema.

2.1 A Multidimensional Data Model

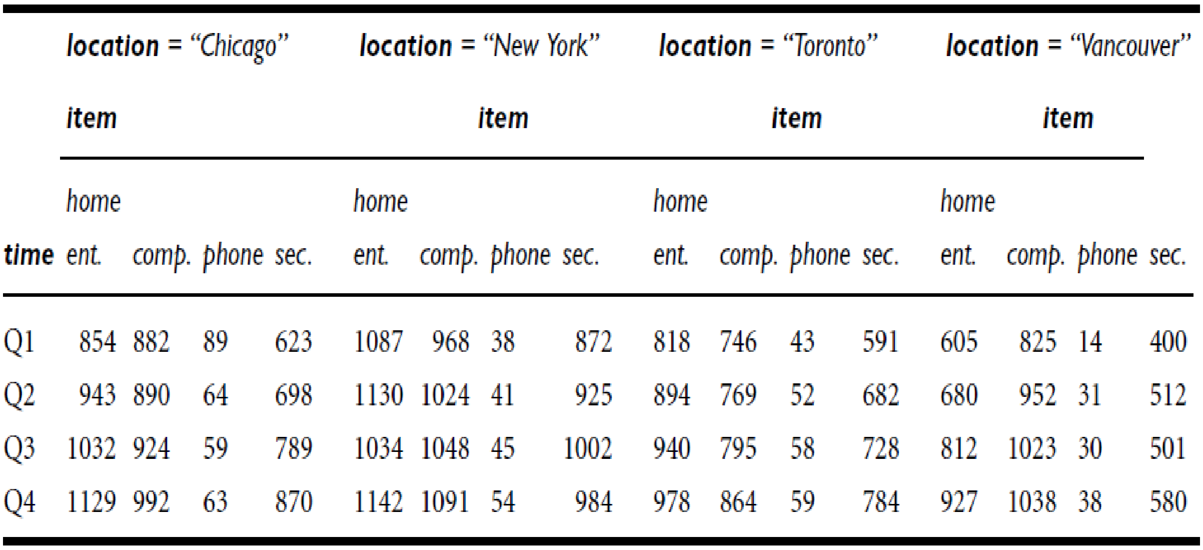
A multidimensional data model is typically used for the design of corporate *data warehouses* and *departmental data marts*. Such a model can adopt a *star schema*, *snowflake schema*, or *fact constellation schema*. Data warehouses and OLAP tools are based on a multidimensional data model which views data in the form of a *data cube*.

2.1.1 From Tables and Spreadsheets to Data Cubes

Data cube

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

Figure below represents dataset as 2-D table (i.e in rows and columns). It shows sales for AllElectronics, according to the dimension time, item, and location.



	location = "Chicago"				**location** = "New York"				**location** = "Toronto"				**location** = "Vancouver"			
	item				**item**				**item**				**item**			
time	home ent.	comp.	phone	sec.	home ent.	comp.	phone	sec.	home ent.	comp.	phone	sec.	home ent.	comp.	phone	sec.
Q1	854	882	89	623	1087	968	38	872	818	746	43	591	605	825	14	400
Q2	943	890	64	698	1130	1024	41	925	894	769	52	682	680	952	31	512
Q3	1032	924	59	789	1034	1048	45	1002	940	795	58	728	812	1023	30	501
Q4	1129	992	63	870	1142	1091	54	984	978	864	59	784	927	1038	38	580

Table: Statistical Table: Two dimensional representation

This form of representing multidimensional tables is very popular in Statistical Data Analyses, because in the early days it was only possible to represent information on paper and thus 2-D restriction. In this type of representation the rows and columns represents more than one dimension, if the dataset contains more than two dimensions. In above table the column contains two dimensions namely location and item.

The above Table can be represented in multi-dimensional view using data cube as follows:

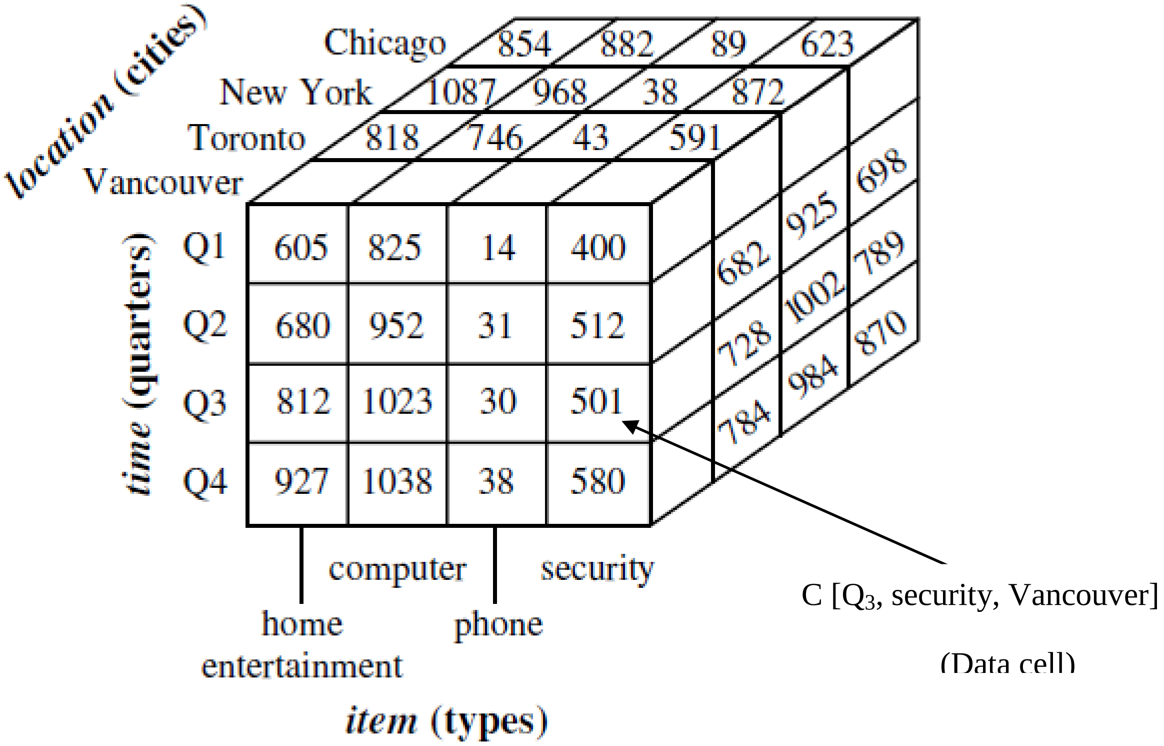


Figure: Multi-dimensional representation of AllElectronics dataset.

Formally, An n-dimensional data cube, C[A_1, A_2,...........,A_n] is a database with n dimensions as A_1, A_2, ..., A_n each of which represents a theme and contains | A_i| number of distinct elements in the dimension A_i. Each distinct element of A_i corresponds to a data row of C. A data cell in the cube C[a_1, a_2, ..., a_n] stores the numeric measures of the data for $A_i = a_i$, for all i. Thus, a data cell corresponds to an instantiation of all dimension.
In above example, C[time, item, location] is the data cube and a data cell C[Q_3, security, Vancouver] stores 501as its associated measure.

In the data warehousing research literature, a data cube such as each of the above is often referred to as a cuboid. Given a set of dimensions, we can generate a cuboid for each of the possible subsets of the given dimensions. The result would form a *lattice* of cuboids, each showing the data at a different level of summarization, or group by. The lattice of cuboids is then referred to as a data cube.
Figure below shows a lattice of cuboids forming a data cube for the dimensions *time*, *item*, *location*, and *supplier*.

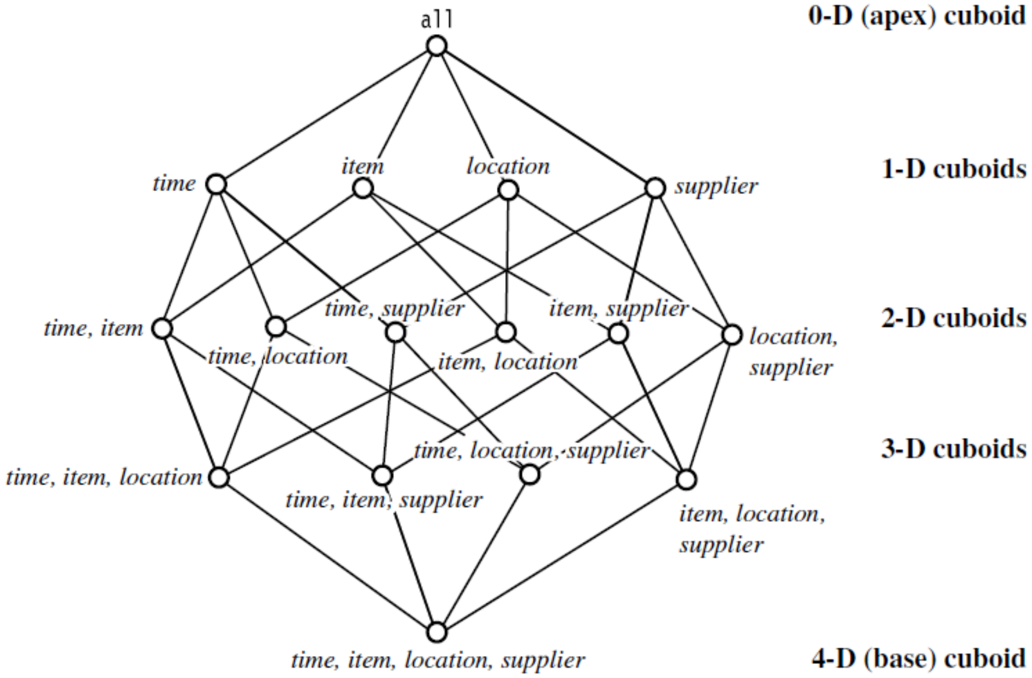


Figure: Lattice of cuboids

Each cuboids represents different degree of summarisation. Generally, n-D cuboid is formed by applying summarisation or group by operation using n-dimension.

Dimensions

Dimensions are the perspectives or entities with respect to which an organization wants to keep records. *Sales* data warehouse may keep records of the store's sales with respect to the dimensions *time*, *item*, *branch*, and *location*.

Dimension table

Dimension table further describes the dimension. For example, a dimension table for *item* may contain the attributes *item name*, *brand*, and *type*.
Syntax:

define dimension <dimension name> as (<attribute or dimension list>)

Fact

Facts are numerical measures which are used to analyse the relationship between dimensions. Examples of facts for a sales data warehouse include *dollars_sold* (sales amount in dollars), *units_sold* (number of units sold), and *amount_budgeted*.

Fact table

The fact table contains the names of the *facts*, or measures, as well as keys to each of the related dimension tables.

Syntax:

define cube <cube name> [<dimension list>]: <measure list>

The define cube statement defines a data cube, which corresponds to the fact table.

2.1.2 Data warehouse schema

A schema is a collection of database objects, including tables, views, indexes, and synonyms. Following are most common schema used in Data Warehouse environment:

a. **Star schema**
b. **Snowflake schema**
c. **Fact constellation schema**

a. **Star schema**

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

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

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

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

The dimension table consists of columns that correspond to the attributes of the dimensions.

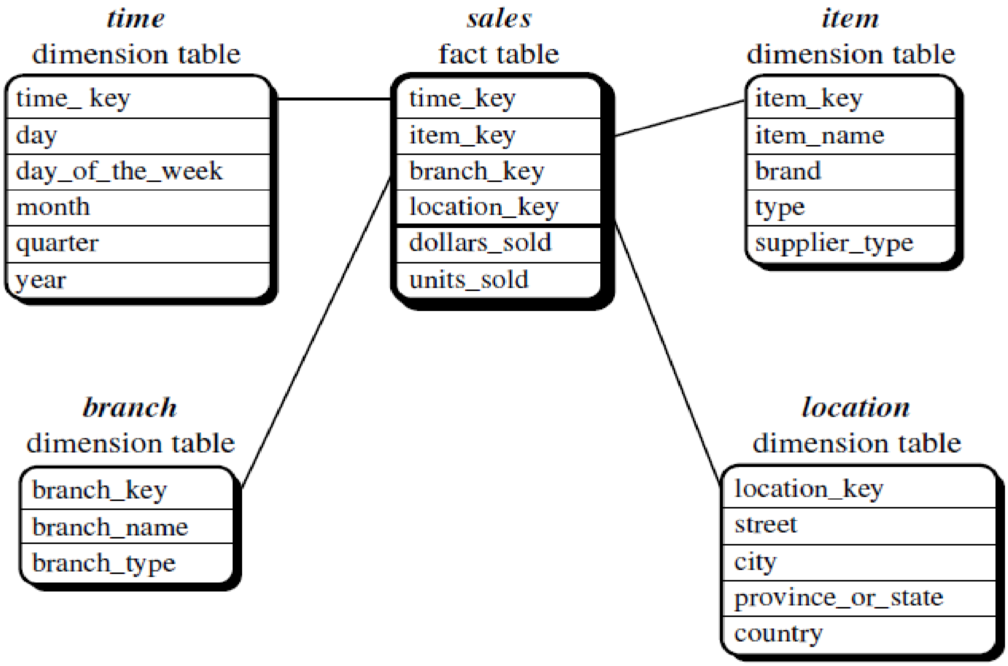


Figure 2.1.2 (a): Star schema for data warehouse for sales.

Example: Star schema. A star schema for *AllElectronics* sales is shown in Figure 2.1.2 (a). Sales are considered along four dimensions, namely, *time, item, branch*, and *location*. The schema contains a central fact table for *sales* that contains keys to each of the four dimensions, along with two measures: *dollars sold* and *units sold*. To minimize the size of the fact table, dimension identifiers (such as *time key* and *item key*) are system-generated identifiers.

Syntax:

define cube sales_star [time, item, branch, location]: dollars_sold = sum(sales_in_dollars), units_sold = count(*)
define dimension time as (time_key, day, day_of_week, month, quarter, year)
define dimension item as (item_key, item_name, brand, type, supplier_type)
define dimension branch as (branch_key, branch_name, branch_type)
define dimension location as (location_key, street, city, province_or_state, country)

Advantages:

i. Easy to understand since all the information about each level is stored in one row.

ii. A star schema optimizes performance by keeping queries simple and providing fast response time as only one join requires to establish the relationship between the fact table and any one of the dimension table.

Disadvanatges:

i. Redundancy of the data hence occupies additional space.

b. Snowflake Schema:

The snowflake schema is a variant of the star schema model, where some dimension tables are *normalized*, thereby further splitting the data into additional tables. The resulting schema graph forms a shape similar to a snowflake.

Snowflake schemas normalize dimensions to eliminate redundancy. While this saves space, it increases the number of dimension tables and requires more foreign key joins. The result is more complex queries and reduced query performance.

Example: Snowflake schema. A snowflake schema for *AllElectronics* sales is given in Figure 2.1.2 (b). Here, the *sales* fact table is identical to that of the star schema in Figure 2.1.2 (a). The main difference between the two schemas is in the definition of dimension tables. The single dimension table for *item* in the star schema is normalized in the snowflake schema, resulting in new *item* and *supplier* tables. For example, the *item* dimension table now contains the attributes *item key, item name, brand, type,* and *supplier key*, where *supplier key* is linked to the *supplier* dimension table, containing *supplier key* and *supplier type* information. Similarly, the single dimension table for *location* in the star schema can be normalized into two new tables: *location* and *city*. The *city key* in the new *location* table links to the *city* dimension. Notice that further normalization can be performed on *province or state* and *country* in the snowflake schema shown in Figure 2.1.2 (b), when desirable.

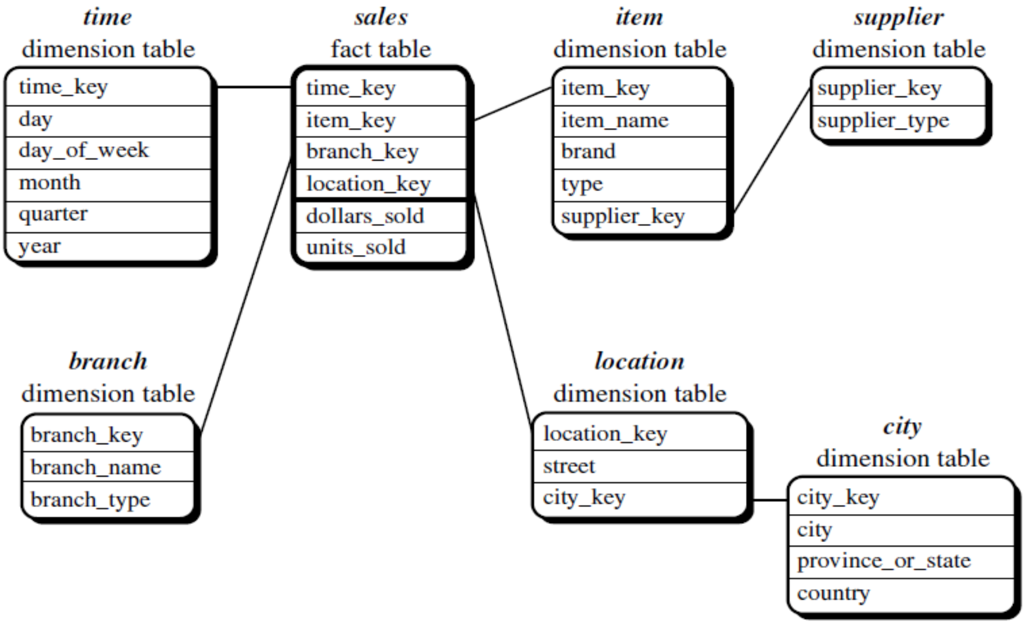


Figure 2.1.2 (b): Snowflake schema of a data warehouse for sales.

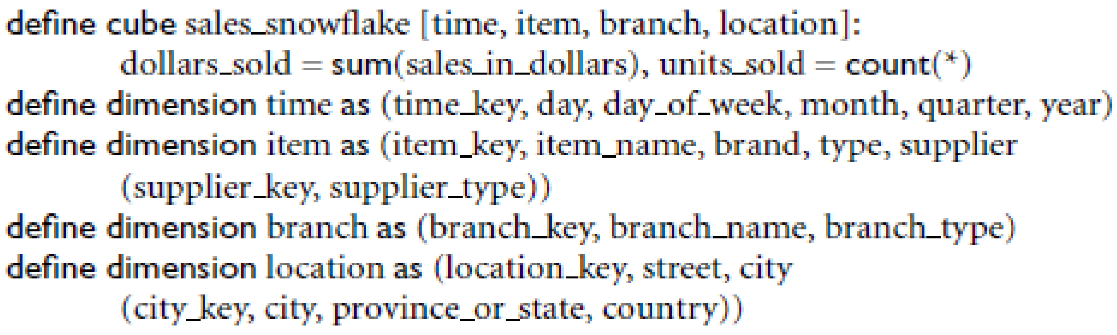
Advantages:

i. Eliminates the redundancies and hence saves the storage space

Disadvantages:

i. it increases the number of dimension tables and requires more foreign key joins. The result is more complex queries and reduced query performance.

Syntax:



```
define cube sales_snowflake [time, item, branch, location]:
          dollars_sold = sum(sales_in_dollars), units_sold = count(*)
define dimension time as (time_key, day, day_of_week, month, quarter, year)
define dimension item as (item_key, item_name, brand, type, supplier
          (supplier_key, supplier_type))
define dimension branch as (branch_key, branch_name, branch_type)
define dimension location as (location_key, street, city
          (city_key, city, province_or_state, country))
```

c. **Fact Constellation schema:**

A Fact constellation schema is a type of schema which consists of more than one fact table sharing to dimension tables. This kind of schema can be viewed as a collection of stars, and hence is called a galaxy schema or a fact constellation.

Example: A fact constellation schema is shown in Figure 2.1.2 (c). This schema specifies two fact tables, *sales* and *shipping*. The *sales* table definition is identical to that of the star schema (Figure 2.1.2 (a)). The *shipping* table has five dimensions, or keys: *item key, time key, shipper key, from location*, and *to location*, and two measures: *dollars cost* and *units shipped*. A fact constellation schema allows dimension tables to be shared between fact tables. For example, the dimensions tables for *time, item*, and *location* are shared between both the *sales* and *shipping* fact tables.

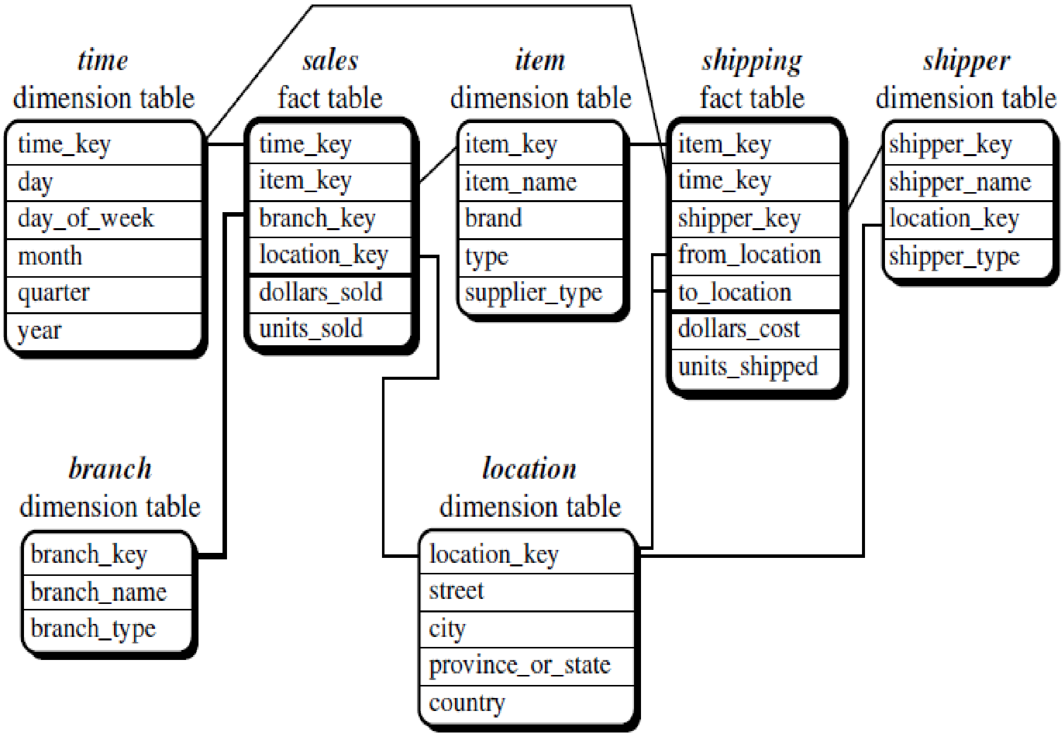
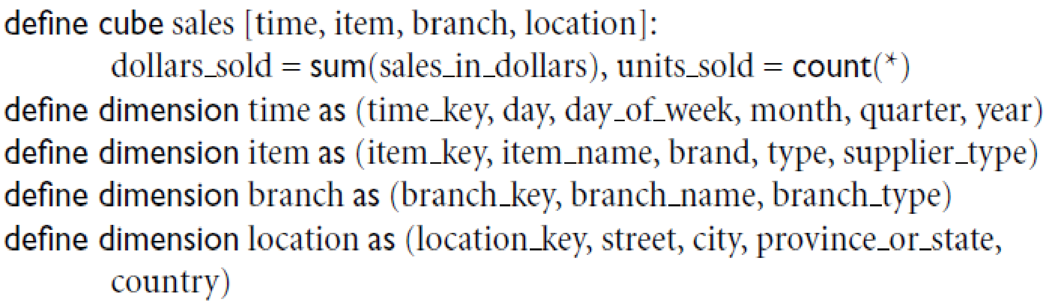
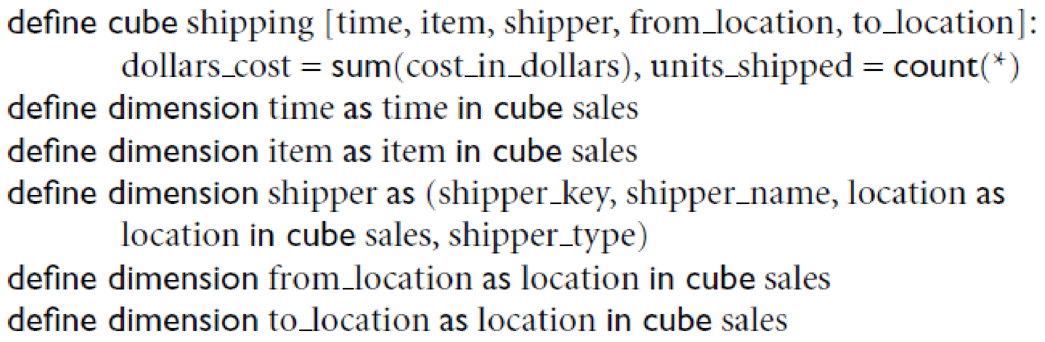


figure 2.1.2 (c) : Fact constellation schema of a data warehouse for sales and shipping.

Syntax:



```
define cube sales [time, item, branch, location]:
          dollars_sold = sum(sales_in_dollars), units_sold = count(*)
define dimension time as (time_key, day, day_of_week, month, quarter, year)
define dimension item as (item_key, item_name, brand, type, supplier_type)
define dimension branch as (branch_key, branch_name, branch_type)
define dimension location as (location_key, street, city, province_or_state,
          country)
```



```
define cube shipping [time, item, shipper, from_location, to_location]:
        dollars_cost = sum(cost_in_dollars), units_shipped = count(*)
define dimension time as time in cube sales
define dimension item as item in cube sales
define dimension shipper as (shipper_key, shipper_name, location as
        location in cube sales, shipper_type)
define dimension from_location as location in cube sales
define dimension to_location as location in cube sales
```

2.1.4 Design and construction of Data warehouse

The construction of a large and complex information system can be viewed as the construction of a large and complex building, for which the owner, architect, and builder have different views. Following four different views regarding the design of a data warehouse must be considered while constructing Data warehouse:

a) ***Top-down view***
   - Allows the selection of the relevant information necessary for the data warehouse.
   - This information matches the current and future business needs.

b) ***Data source view***
   - It exposes the information being captured, stored, and managed by operational systems.

c) ***Data warehouse view***
   - It Includes fact tables and dimension tables.
   - It represents the information that is stored inside the data warehouse, including pre-calculated totals and counts, as well as information regarding the source, date, and time of origin, added to provide historical context.

d) ***Business query view***
   - *It is the perspective of data in the data warehouse from the viewpoint of the end user.*

2.1.5 Process of Data warehouse design

The warehouse design process consists of the following steps:

a) **Choose a *business process* to model**.
   - For example orders, invoices, shipments, inventory, account administration, sales, or the general ledger.
   - If the business process is organizational and involves multiple complex object collections, a data warehouse model should be followed.
   - If the process is departmental and focuses on the analysis of one kind of business process, a data mart model should be chosen.

b) **Choose the *grain* of the business process.**
   - The grain is the fundamental, atomic level of data to be represented in the fact table for the process.
   - For example, individual transactions, individual daily snapshots, and so on.

c) **Choose the *dimensions* that will apply to each fact table record.**
   - Typical dimensions are time, item, customer, supplier, warehouse, transaction type, and status.
   -

d) **Choose the *measures* that will populate each fact table record.**
   - Typical measures are numeric additive quantities like *dollars sold and units sold.*

2.1.6 Materialized View (Computation of cuboids)

- In data warehouses, materialized views are used to pre-compute and store aggregated data such as the sum of sales.
- Materialized views in these environments are often referred to as summaries, because they store summarized data.
- They can also be used to pre-compute joins with or without aggregations.
- A materialized view eliminates the overhead associated with expensive joins and aggregations for a large or important class of queries.

Need of Materialized Views

Materialized views improve query performance by pre calculating expensive join and aggregation operations on the database prior to execution and storing the results in the database. The query optimizer automatically recognizes when an existing materialized view can and should be used to satisfy a request. It then transparently rewrites the request to use the materialized view. Queries go directly to the materialized view and not to the underlying detail tables.

Types of Materialized Views

There are three type of Materialization:

a) **No materialization**
   - Do not pre-compute any of the “nonbase” cuboids.
   - This leads to computing expensive multidimensional aggregates on the fly, which can be extremely slow.

b) **Full materialization**
   - Precompute all of the cuboids.
   - This choice typically requires huge amounts of memory space in order to store all of the precomputed cuboids.

c) **Partial materialization**

- Selectively compute a proper subset of the whole set of possible cuboids.
- Compute a subset of the cube, which contains only those cells that satisfy some user-specified criterion.
- Partial materialization represents an interesting trade-off between storage space and response time.

Chapter 3: Data Warehouse Physical Design

3.1 Physical Design

- Physical design is the phase of a database design following the logical design that identifies the actual database tables and index structures used to implement the logical design.
- It concerned with designing the effective way of storing and retrieving the objects as well as handling them from a transportation and backup/recovery perspective.
- Physical design decisions are mainly driven by query performance and database maintenance aspects.

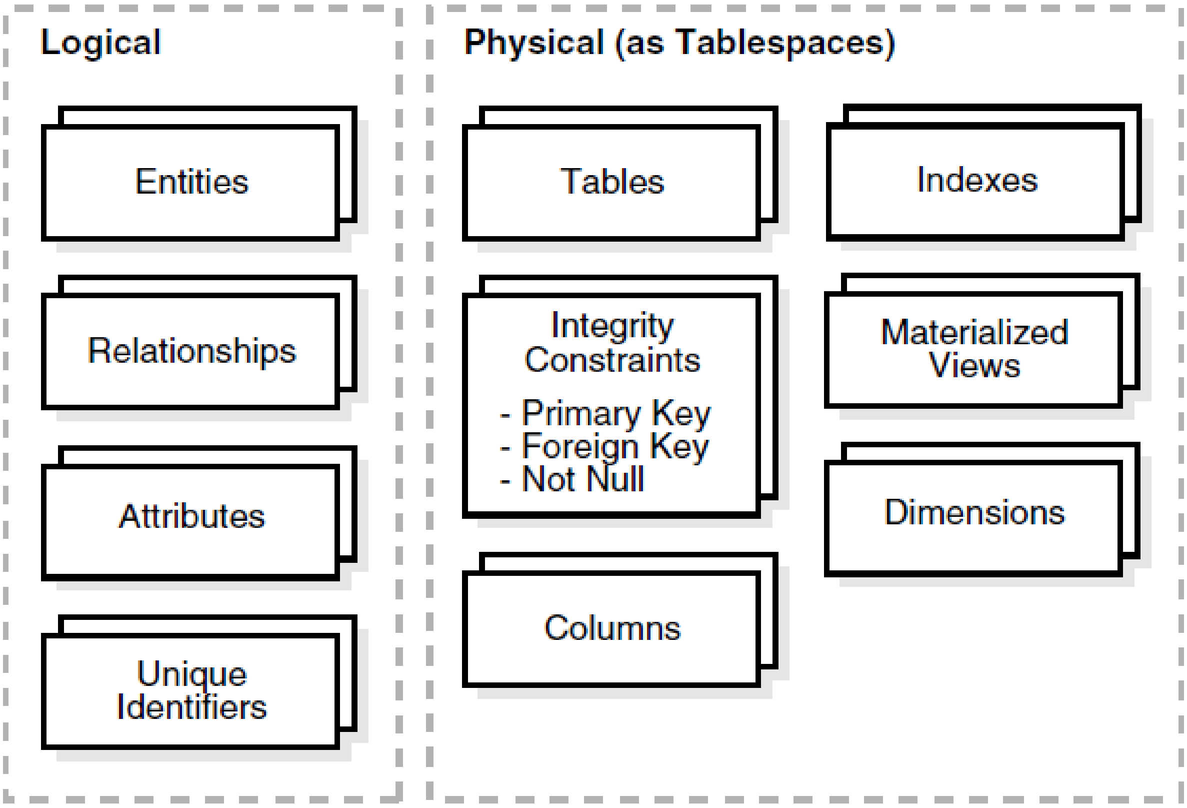


Figure 3.1: Logical Design Compared with Physical Design

- Physical design process translates the expected schema into actual database structure.
- At this phase
  - Entities to tables
  - Relationships to foreign key constraints
  - Attributes to columns
  - Primary unique identifiers to primary key constraints
  - Unique identifiers to unique key constraints

3.1.1 Physical Design Structure

In physical design the following structures are created:

- Tablespaces
- Tables and Partitioned Tables
- Views
- Integrity Constraints

Additionally, the following structures may be created for performance improvement:

- Indexes and Partitioned Indexes
- Materialized Views

3.2 Hardware and I/O Considerations

- I/O performance should always be a key consideration for data warehouse designers and administrators.
- The typical workload in a data warehouse is especially I/O intensive, with operations such as large data loads and index builds, creation of materialized views, and queries over large volumes of data.
- The underlying I/O system for a data warehouse should be designed to meet these heavy requirements.
- In fact, one of the leading causes of performance issues in a data warehouse is poor I/O configuration.
- The I/O configuration used by a data warehouse will depend on the characteristics of the specific storage and server capabilities

There are following five high-level guidelines for data-warehouse I/O configurations:

- **Configure I/O for Bandwidth not Capacity**
  - Storage configurations for a data warehouse should be chosen based on the I/O bandwidth that they can provide, and not necessarily on their overall storage capacity.
- **Stripe Far and Wide**
  - The goal is to ensure that each tablespace is striped across a large number of disks so that any database object can be accessed with the highest possible I/O bandwidth.
- **Use Redundancy**
  - Because data warehouses are often the largest database systems in a company, they have the most disks and thus are also the most susceptible to the failure of a single disk.
  - Therefore, disk redundancy is a requirement for data warehouses to protect against a hardware failure.

- **Test the I/O System Before Building the Database**
  - When creating a data warehouse on a new system, the I/O bandwidth should be tested before creating all of the database datafiles to validate that the expected I/O levels are being achieved.
  - Once the database files are created, it is more difficult to reconfigure the files.
- **Plan for Growth**
  - A data warehouse designer should plan for future growth of a data warehouse.
  - There are many approaches to handling the growth in a system, and the key consideration is to be able to grow the I/O system without compromising on the I/O bandwidth.

3.3 Parallelism

- Data warehouses often contain large tables and require techniques both for managing these large tables and for providing good query performance across these large tables.
- **Parallelism is the idea of breaking down a task so that, instead of one process doing all of the work in a query, many processes do part of the work at the same time.**
- **Parallel execution is sometimes called parallelism.**
- **Parallel execution dramatically reduces response time for data-intensive operations on large databases typically associated with Decision Support Systems (DSS) and data warehouses.**
- An example of this is when four processes handle four different quarters in a year instead of one process handling all four quarters by itself.

3.3.1When to parallelise?

- When the operations access significant amounts of data.
- when operations can be implemented independent of each other "Divide-&-Conquer"

3.3.2 Benefits of Parallelism

Parallelism improves processing for:

- Queries requiring large table scans, joins, or partitioned index scans
- Creation of large indexes
- Creation of large tables (including materialized views)
- Bulk inserts, updates, merges, and deletes

3.4 Indexing

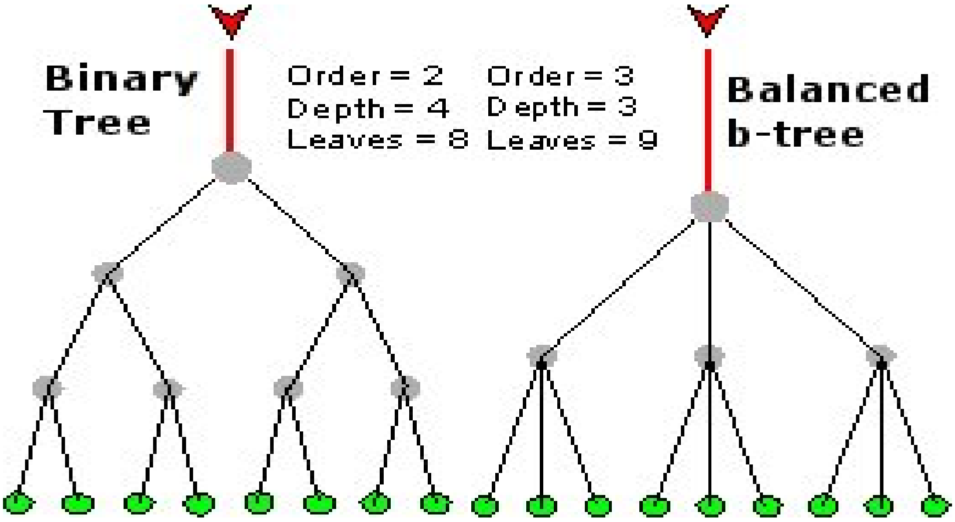
Indexes are optional structures associated with tables and clusters. **Indexes are typically used to speed up the retrieval of records in response to search conditions.** In a query-centric system like the data warehouse environment, the need to process queries faster dominates. Among the various methods to improve performance, indexing ranks very high.

Index structure commonly used in Data warehouse environment are:

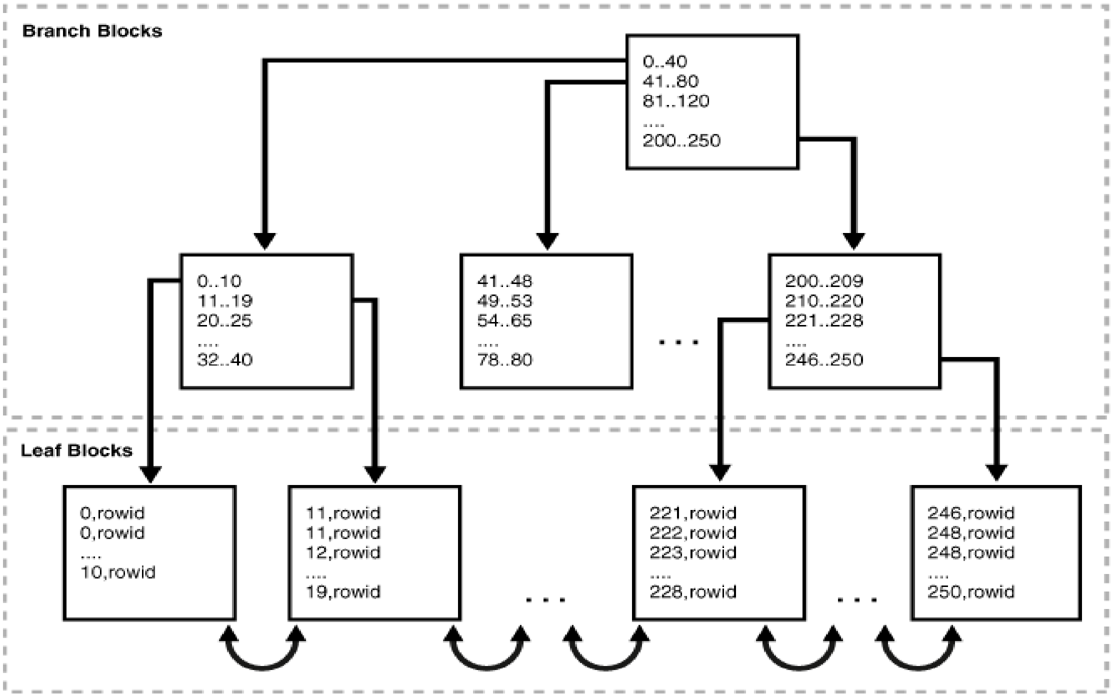
a) **B-tree indexes**
b) **Bitmap indexes**
c) **Join Indexes**

3.4.1 B-tree indexes

- B-trees, short for **balanced trees**, are the most common type and default database index.
- **A B-tree is a tree data structure that keeps data sorted and allows searches, sequential access, insertions, and deletions.**



- **The B-tree is a generalization of a binary search tree in that a node can have more than two children.**
- Figure below shows an example of a B-Tree Index.



- A B-tree index has two types of blocks: **branch blocks** for searching and **leaf blocks** that store values.

- The upper-level branch blocks of a B-tree index contain index data that points to lower-level index blocks.

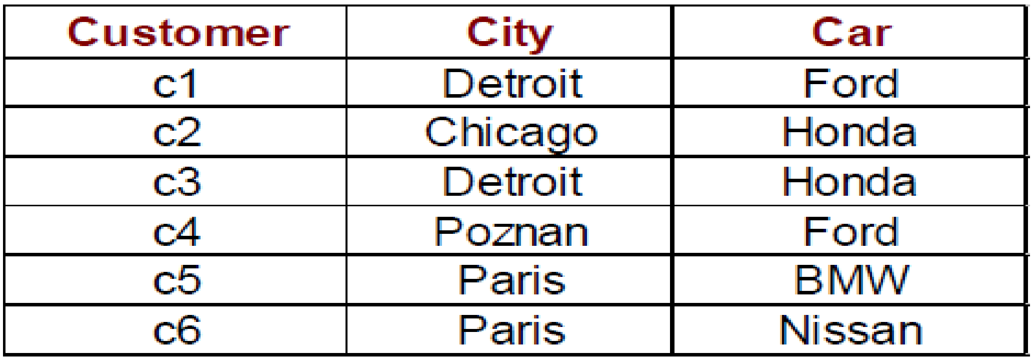
- The lowest level index blocks are called leaf blocks, and these blocks contain every indexed data value and a corresponding ROWID used to locate the actual row.

B-tree indexes are the most common index type used in typical OLTP applications and provide excellent levels of functionality and performance. Used in both OLTP and data warehouse applications, they speed access to table data when users execute queries with varying criteria, such as equality conditions and range conditions. B-tree indexes improve the performance of queries that select a small percentage of rows from a table.

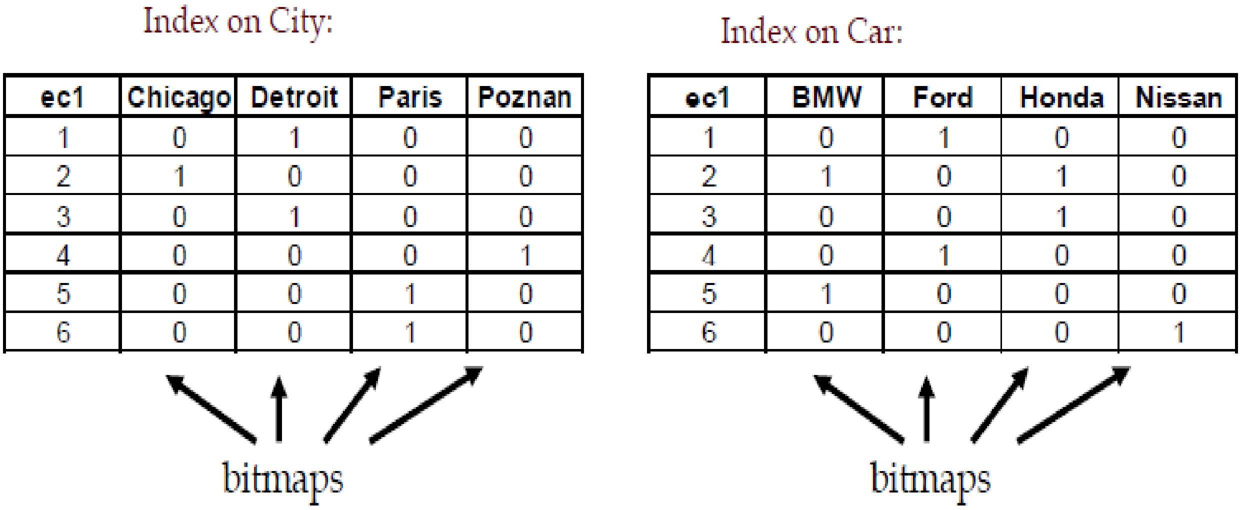
B-tree index is a poor choice for name and text searches because it is case-sensitive and requires a left-to-right match. **B-tree indexes are most commonly used in a data warehouse to index unique or near-unique keys.** In many cases, it may not be necessary to index these columns in a data warehouse, because unique constraints can be maintained without an index, and because typical data warehouse queries may not work better with such indexes. Bitmap indexes should be more common than B-tree indexes in most data warehouse environments.

3.4.2. Bitmap Index

- The concept of bitmap index was first introduced by Professor Israel Spiegler and Rafi Maayan in their research "Storage and Retrieval Considerations of Binary Data Bases", published in 1985.
- A bitmap index is a special kind of database index that uses bitmaps and are used widely in multi-dimensional database implementation.
- Bitmap indexes are primarily intended for data warehousing applications where users query the data rather than update it.
- They are not suitable for OLTP applications with large numbers of concurrent transactions modifying the data.
- Bitmap indexes use bit arrays (commonly called bitmaps) and answer queries by performing bitwise logical operations on these bitmaps.
- In a bitmap index, a bitmap for each key value replaces a list of rowids.
- Each bit in the bitmap corresponds to a possible rowid, and if the bit is set, it means that the row with the corresponding rowid contains the key value.
- Each value in the indexed column has a bit vector (bitmaps).
- The length of the bit vector is the number of records in the base table.
- The i-th bit is set if the i-th row of the base table has the value for the indexed column.



Customer	City	Car
c1	Detroit	Ford
c2	Chicago	Honda
c3	Detroit	Honda
c4	Poznan	Ford
c5	Paris	BMW
c6	Paris	Nissan



Index on City:

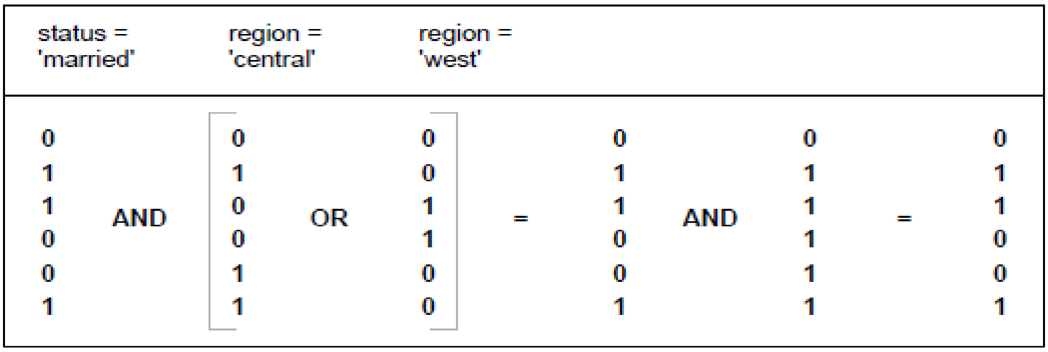
ec1	Chicago	Detroit	Paris	Poznan
1	0	1	0	0
2	1	0	0	0
3	0	1	0	0
4	0	0	0	1
5	0	0	1	0
6	0	0	1	0

Index on Car:

ec1	BMW	Ford	Honda	Nissan
1	0	1	0	0
2	1	0	1	0
3	0	0	1	0
4	0	1	0	0
5	1	0	0	0
6	0	0	0	1

Executing a query using Bitmap Indexes

SELECT COUNT(*) FROM CUSTOMER WHERE status = ‘married’ AND region = ‘central’ OR region = ‘west’



status = 'married' | region = 'central' | region = 'west'

$$\begin{matrix}0\\1\\1\\0\\0\\1\end{matrix} \;\text{AND}\; \left[\begin{matrix}0\\1\\0\\0\\1\\1\end{matrix} \;\text{OR}\; \begin{matrix}0\\0\\1\\1\\0\\0\end{matrix}\right] = \begin{matrix}0\\1\\1\\0\\0\\1\end{matrix} \;\text{AND}\; \begin{matrix}0\\1\\1\\1\\1\\1\end{matrix} = \begin{matrix}0\\1\\1\\0\\0\\1\end{matrix}$$

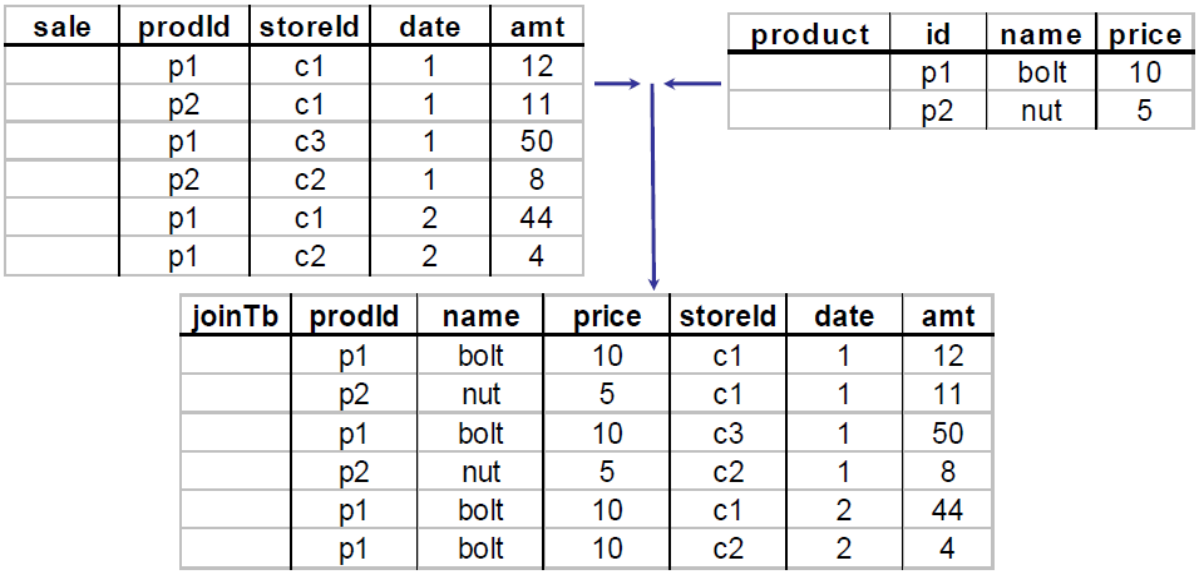
Advantage of Bitmap Indexing

Bitmap indexing provides:

- Reduced response time for large classes of ad hoc queries
- Reduced storage requirements compared to other indexing techniques
- Dramatic performance gains even on hardware with a relatively small number of CPUs or a small amount of memory
- Efficient maintenance during parallel DML and loads

3.4.3 Join Index

- Join indexes map the tuples in the join result of two relations to the source tables.
- In data warehouse cases, join indexes relate the values of the dimensions of a star schema to rows in the fact table.
    - For a warehouse with a Sales fact table and dimension city, a join index on city maintains for each distinct city a list of RIDs of the tuples recording the sales in the city
- Join indexes can span multiple dimensions

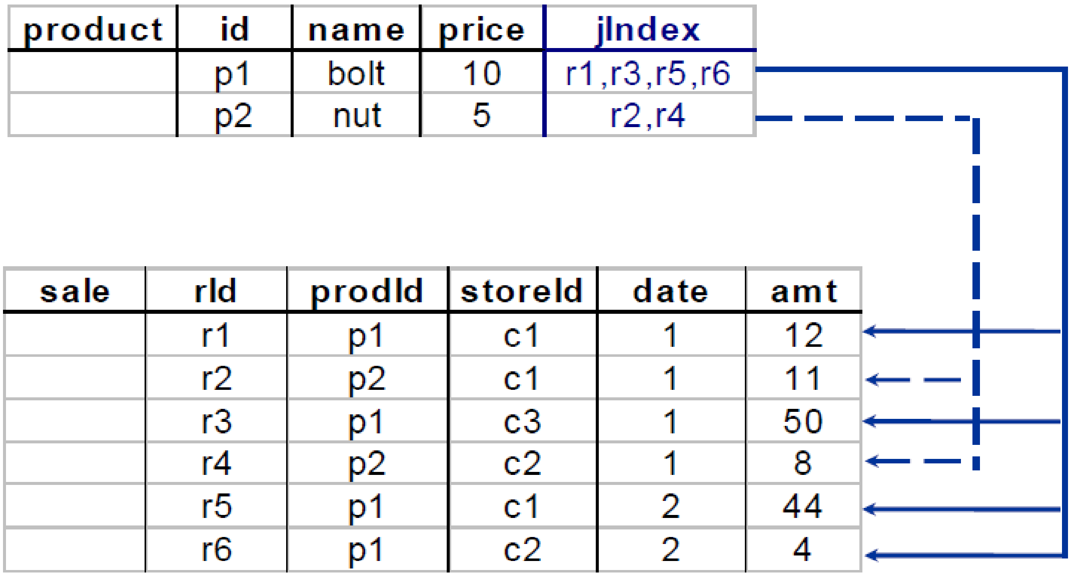


sale	prodId	storeId	date	amt
	p1	c1	1	12
	p2	c1	1	11
	p1	c3	1	50
	p2	c2	1	8
	p1	c1	2	44
	p1	c2	2	4

product	id	name	price
	p1	bolt	10
	p2	nut	5

joinTb	prodId	name	price	storeId	date	amt
	p1	bolt	10	c1	1	12
	p2	nut	5	c1	1	11
	p1	bolt	10	c3	1	50
	p2	nut	5	c2	1	8
	p1	bolt	10	c1	2	44
	p1	bolt	10	c2	2	4

"Combine" SALE, PRODUCT relations
In SQL: SELECT * FROM SALE, PRODUCT



product	id	name	price	jIndex
	p1	bolt	10	r1,r3,r5,r6
	p2	nut	5	r2,r4

sale	rId	prodId	storeId	date	amt
	r1	p1	c1	1	12
	r2	p2	c1	1	11
	r3	p1	c3	1	50
	r4	p2	c2	1	8
	r5	p1	c1	2	44
	r6	p1	c2	2	4

Figure: Join Index.

Chapter 4: Data Warehousing technologies and Implementation

4.1 Data preprocessing

Data preprocessing is an important issue for both data warehousing and data mining, as real-world data tend to be incomplete, noisy, and inconsistent. Incomplete data can occur for a number of reasons. Attributes of interest may not always be available, such as customer information for sales transaction data. Other data may not be included simply because it was not considered important at the time of entry. There are many possible reasons for noisy data (having incorrect attribute values). The data collection instruments used may be faulty, human or computer errors occurring at data entry, Errors in data transmission. Thus, analysing these real world data without pre-processing is difficult and if somehow performed its results are not accurate, reliable, and effective.

Data preprocessing is done to improve the quality of Data and consequently the mining result. It is used to improve the efficiency and ease of the mining process. Data preprocessing includes:

- **Data cleaning,**
- **Data integration,**
- **Data transformation**
- **Data reduction.**

4.1.1 Data cleaning

Real-world data tend to be incomplete, noisy, and inconsistent. *Data cleaning* (or *data cleansing*) routines attempt to fill in missing values, smooth out noise while identifying outliers, and correct inconsistencies in the data.

4.1.1.1 Handling missing values

- Ignore the tuple
- Fill in the missing value manually
- Use a global constant to fill in the missing value: use "unknown" or -∞.
- Use the attribute mean to fill in the missing value
- Use the attribute mean for all samples belonging to the same class as the given tuple
- Use the most probable value to fill in the missing value

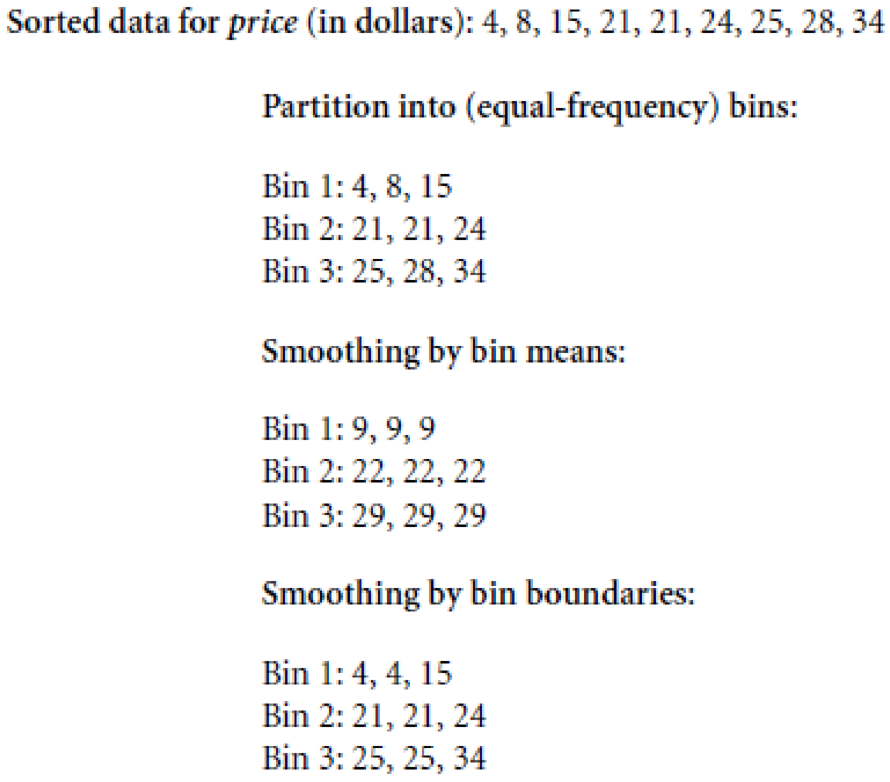
4.1.1.2 Handling Noisy Data

Noise is a random error or variance in a measured variable. Following data smoothing techniques are used to remove noise from data:

a) **Binning**

Binning methods smooth a sorted data value by consulting its “neighborhood,” that is, the values around it. The sorted values are distributed into a number equal-frequency “buckets,” or *bins*. Then, one of the following binning techniques is used for smoothing:

- **Smoothing by bin means**
  - Each value in a bin is replaced by the mean value of the bin.
- **Smoothing by bin medians**
  - Each bin value is replaced by the bin median
- **Smoothing by bin boundaries**
  - Minimum and maximum values in a given bin are identified as the *bin boundaries*.
  - Each bin value is then replaced by the closest boundary value.



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

Partition into (equal-frequency) bins:

Bin 1: 4, 8, 15
Bin 2: 21, 21, 24
Bin 3: 25, 28, 34

Smoothing by bin means:

Bin 1: 9, 9, 9
Bin 2: 22, 22, 22
Bin 3: 29, 29, 29

Smoothing by bin boundaries:

Bin 1: 4, 4, 15
Bin 2: 21, 21, 24
Bin 3: 25, 25, 34

Figure 4.1.1.2 (a): Binning methods for data smoothing

b) **Regression**

Data can be smoothed by fitting the data to a function, such as with regression. ***Linear regression*** involves finding the “best” line to fit two attributes (or variables), so that one attribute can be used to predict the other. ***Multiple linear regression*** is an extension of linear regression, where more than two attributes are involved and the data are fit to a multidimensional surface.

c) **Clustering**

Outliers may be detected by clustering, where similar values are organized into groups, or "clusters." Intuitively, values that fall outside of the set of clusters may be considered outliers.

4.1.2 Data Integration

Data integration combines data from multiple sources to form a coherent data store. These sources may include multiple databases, data cubes, or flat files.
There are a number of issues to consider during data integration such as:

- **Heterogeneous data**: This has no common key
- **Different definition**: This is intrinsic, that is, same data with different definition, such as a different database schema
- **Time synchronization**: This checks if the data is gathered under same time periods
- **Legacy data**: This refers to data left from the old system
- **Sociological factors**: This is the limit of data gathering

There are several approaches that deal with the above issues:

- **Entity identification problem**: Schema integration and object matching are referred to as the entity identification problem. Metadata helps to solve this problem.
- **Redundancy and correlation analysis**: Some redundancies can be detected by correlation analysis. Given two attributes, such an analysis can measure how strongly one attribute implies the other, based on the available data.
- **Tuple Duplication**: Duplication should be detected at the tuple level to detect redundancies between attributes
- **Data value conflict detection and resolution**: Attributes may differ on the abstraction level, where an attribute in one system is recorded at a different abstraction level

4.1.3 Data Transformation

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

- **Selection**
- **Splitting/Joining**
- **Conversion**
- **Summarisation**
- **Enrichment**

- **Selection**
  - This takes place at the beginning of the whole process of data transformation.
  - It selects the whole records or parts of several records from the source system.
  - The task of selection usually forms part of the extraction function itself.

- However, in some cases, the composition of the source structure may not be supporting selection of the necessary parts during data extraction.
- In these cases, it is advised to extract the whole record and then do the selection as part of the transformation function.

- **Splitting/Joining**

4.1.4 Data Reduction

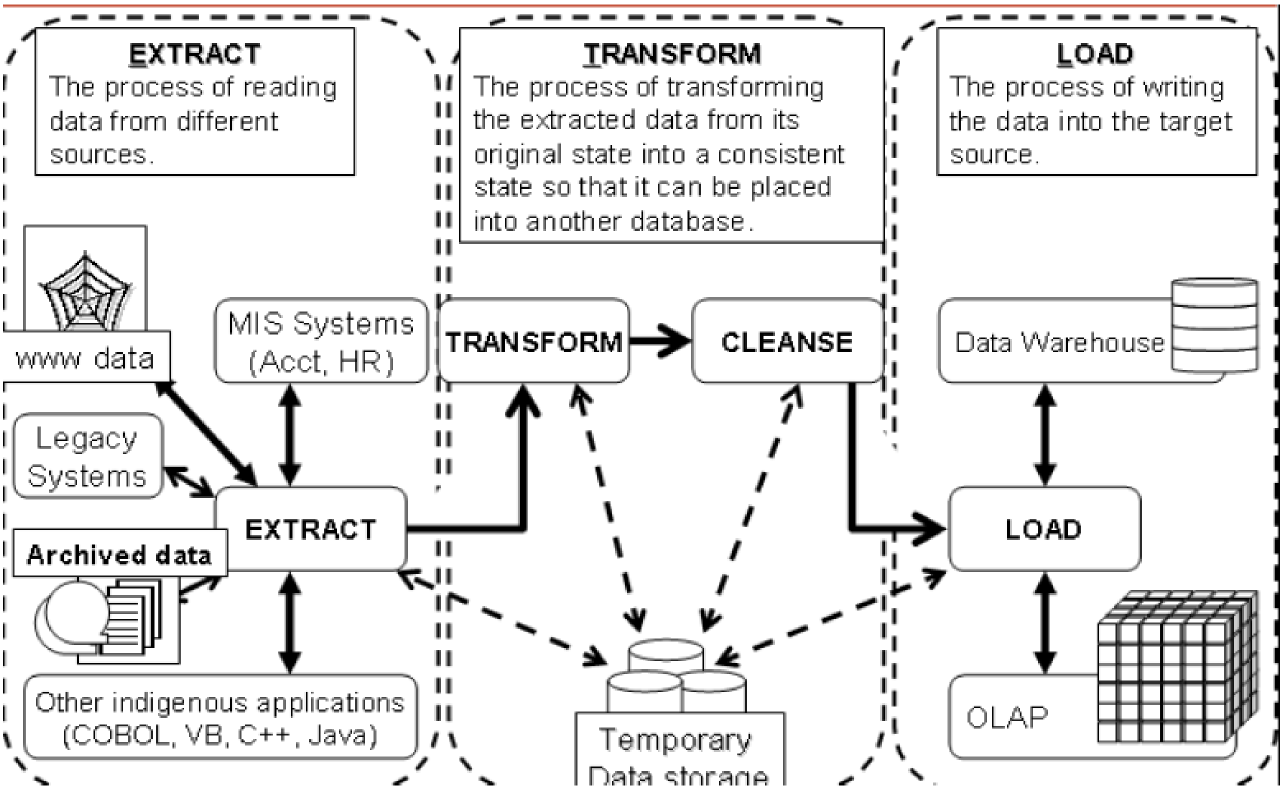
Data reduction techniques can be applied to obtain a reduced representation of the data set that is much smaller in volume, yet closely maintains the integrity of the original data.
Strategies for data reduction include the following:

- **Data cube aggregation,** where aggregation operations are applied to the data in the construction of a data cube.
- **Attribute subset selection,** where irrelevant, weakly relevant, or redundant attributes or dimensions may be detected and removed.
- **Dimensionality reduction,** where encoding mechanisms are used to reduce the data set size.
- **Numerosity reduction,** where the data are replaced or estimated by alternative, smaller data representations.
- **Discretization and concept hierarchy generation,** where raw data values for attributes are replaced by ranges or higher conceptual levels.

4.2 Data warehouse backend tools (ETL process)

The large amount of data produced in the organization is converted into data warehouse through the ETL process. It consists of following three main activities:

1. Extraction
2. Transformation & Cleansing
3. Loading & refreshing


EXTRACT
The process of reading data from different sources.
TRANSFORM
The process of transforming the extracted data from its original state into a consistent state so that it can be placed into another database.
LOAD
The process of writing the data into the target source.
www data
MIS Systems (Acct, HR)
Legacy Systems
Archived data
EXTRACT
Other indigenous applications (COBOL, VB, C++, Java)
TRANSFORM
CLEANSE
Temporary Data storage
Data Warehouse
LOAD
OLAP

4.2.1 Data Extraction

Data Extraction is the process of extracting data for the warehouse from various sources. The data may come from various sources, such as

- Production data
- Legacy data
- Internal office system
- External systems
- Metadata

4.2.2 Transformation and Cleansing

Refer to section 4.1.1 and 4.1.3.

4.2.3 Loading

It is the process of moving the data into data warehouse repository is known as data loading. Loading can be carried out in following ways:

- **Initial Load:** populating all the data warehouse tables for the very first time.
- **Refresh**: Data warehouse data is completely rewritten. This means that the older data is completely replaced.
- **Update:** Only those changes applied to source data are added to the data warehouse. Update is carried out without deleting or modifying preexisting data.

Chapter 5: Data Warehouse to Data Mining

5.1 Online Analytical Processing (OLAP)

- Online Analytical Processing Server (OLAP) is based on the multidimensional data model.
- It allows managers and analysts to get an insight of the information through fast, consistent, and interactive access to information.
- **OLAP** facilitates users to extract and present multidimensional data from different view.
- **OLAP** provides a user-friendly environment for interactive data analysis.
- It enables users to gain a deeper understanding and knowledge about various aspects of their corporate data through fast, consistent, interactive access to a wide variety of possible views of the data.
- **OLAP** provides you with a very good view of *what is happening*, but cannot predict *what will happen in the future* or *why it is happening*.

5.1.1 Benefits of OLAP

- Increased productivity of end-users.
- Retention of organizational control over the integrity of corporate data.
- Improved potential revenue and profitability.

5.2 OLAP OPERATIONS

Following are the typical OLAP operations that are used for multidimensional analysis of data:

a. **Roll-up**
b. **Drill-down**
c. **Slice and dice**
d. **Pivot (rotate)**
e. **Other OLAP operations: drill-across, drill-through**

a. **Roll-up (Drill-up)**

- The roll-up (also called the *drill-up) operation* performs aggregation on a data cube, either by *climbing up a concept hierarchy for* a dimension or by *dimension reduction.*
- dimension reduction: e.g., total sales by city and year -> total sales by city (here we reduce the dimension year).
- summarization over aggregate hierarchy: e.g., total sales by city and year -> total sales by country and by year (.climbing up from city to country).

b. Drill-down (Roll-down)

- Drill-down is the reverse of roll-up.
- It navigates from less detailed data to more detailed data by either *stepping down a concept hierarchy* for a dimension or *introducing additional dimensions.*
- *Example: Sales per month rather than summarising them by quater*.

c. Slice and dice

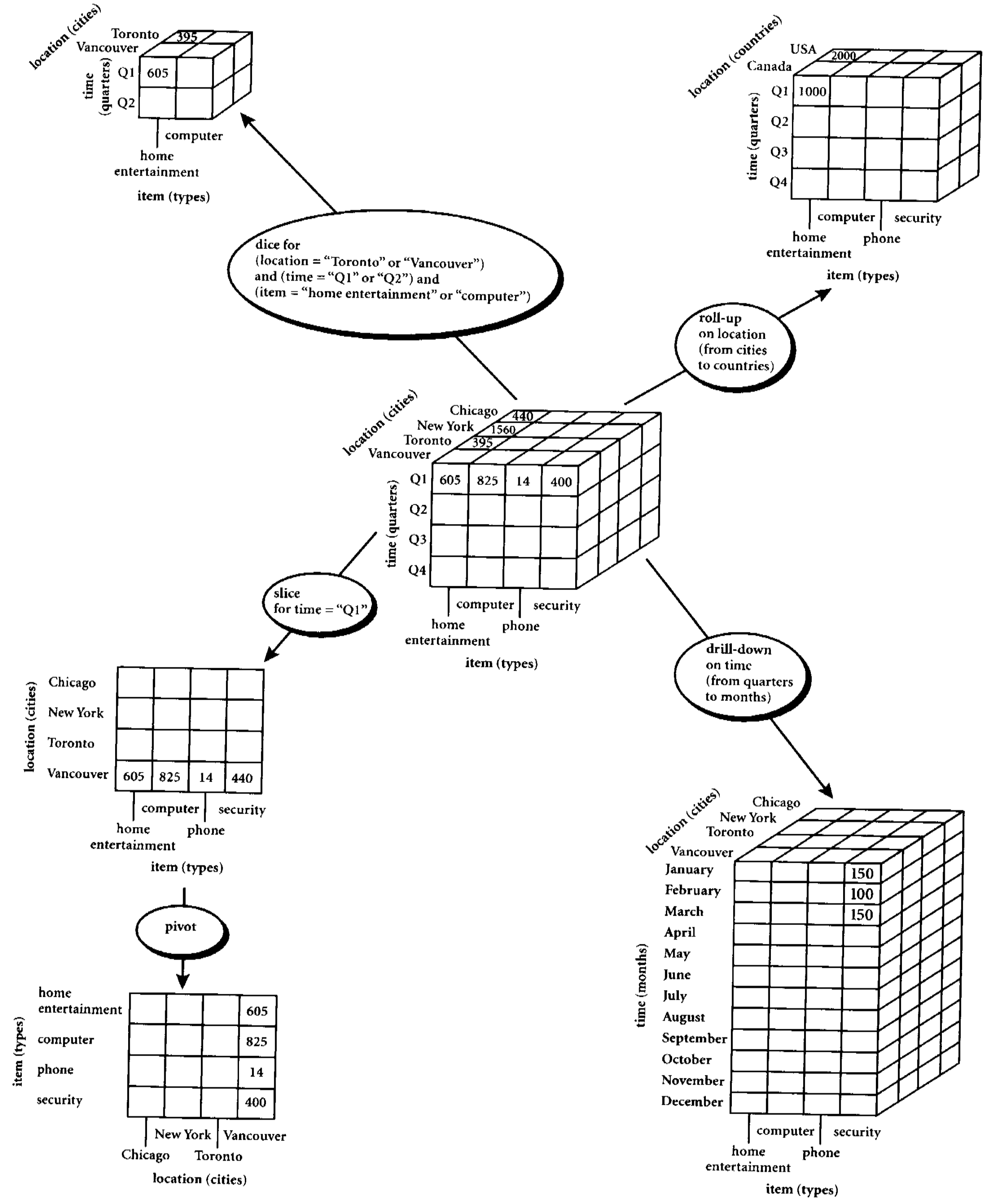
- **Slice:** Slice operation performs a selection on one dimension of the given cube, resulting in a subcube.
- **Dice:** The *dice operation defines a subcube by performing a* selection on two or more dimensions.

d. Pivot (Rotate)

- ***Pivot*** *(also called rotate) is a visualization operation that rotates the data* axes in view in order to provide an alternative presentation of the data.
- Examples include rotating the axes in a 3-D cube, or transforming a 3-D cube into a series of 2-D planes.

e. Other operations

- ***Drill across:*** *executes queries involving (across) more than one fact table*.
- ***Drill through:*** *through the bottom level of the cube to its back-end relational tables (using SQL).*


location (cities)
Toronto
Vancouver
395
time (quarters)
Q1
Q2
605
computer
home entertainment
item (types)
dice for
(location = "Toronto" or "Vancouver")
and (time = "Q1" or "Q2") and
(item = "home entertainment" or "computer")
location (countries)
USA
Canada
2000
Q1
Q2
Q3
Q4
1000
computer
security
home entertainment
phone
item (types)
roll-up
on location
(from cities
to countries)
location (cities)
Chicago
New York
Toronto
Vancouver
440
1560
395
time (quarters)
Q1
Q2
Q3
Q4
605
825
14
400
computer
security
home entertainment
phone
item (types)
slice
for time = "Q1"
location (cities)
Chicago
New York
Toronto
Vancouver
605
825
14
440
computer
security
home entertainment
phone
item (types)
pivot
item (types)
home entertainment
computer
phone
security
605
825
14
400
New York
Vancouver
Chicago
Toronto
location (cities)
drill-down
on time
(from quarters
to months)
location (cities)
Chicago
New York
Toronto
Vancouver
time (months)
January
February
March
April
May
June
July
August
September
October
November
December
150
100
150
computer
security
home entertainment
phone
item (types)

5.3 Types of OLAP Server

OLAP servers present business users with multidimensional data from data warehouses or data marts, without concerns regarding how or where the data are stored.

OLAP tools are categorized according to the architecture of the underlying database.

There are four types of OLAP servers:

a. **Relational OLAP (ROLAP)**
b. **Multidimensional OLAP (MOLAP)**
c. **Hybrid OLAP (HOLAP)**
d. **Desktop OLAP (DOLAP)**

5.3.1 Relational OLAP (ROLAP)

- These are the intermediate servers that stand in between a relational back-end server and client front-end tools.
- They use a *relational or extended-relational DBMS to store and manage warehouse data.*
- ROLAP or a Relational OLAP provides access to information via a relational database using ANSI standard SQL.
- Examples: Microstrategy, Business Objects, Crystal Holos (ROLAP Mode), Essbase, Microsoft Analysis Services, Oracle Express (ROLAP Mode), Oracle Discoverer.

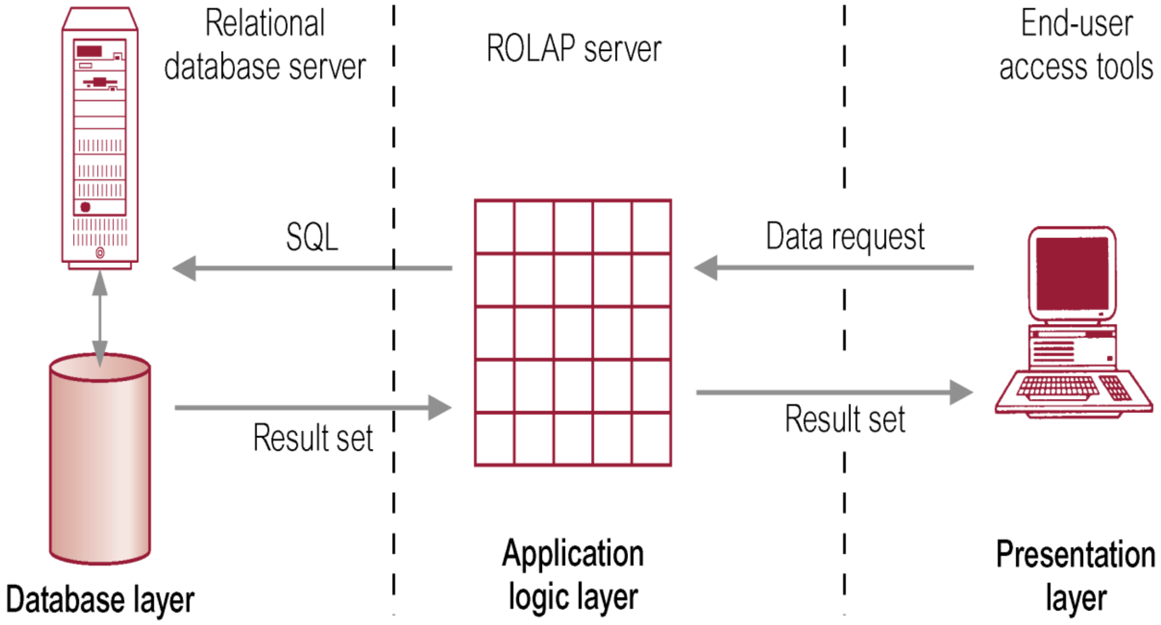


Figure 5.3.1: Typical Architecture of ROLAP

Features of ROLAP:

- Ask any question (not limited to the contents of the cube)
- Ability to drill down

Downsides of ROLAP:

- Slow Response
- Some limitations on scalability

5.3.2 Multidimensional OLAP (MOLAP)

- These servers support multidimensional views of data through *array-based multidimensional storage engines.*
- MOLAP physically builds "cubes" for direct access - usually in the proprietary file format of a multi-dimensional database (MDD) or a user defined data structure. Therefore ANSI SQL is not supported.
- Data is typically aggregated and stored according to predicted usage to enhance query performance.
- The main advantage of an MDDB over an RDBMS is that an MDDB can provide information quickly since it is calculated and stored at the appropriate hierarchy level in advance.
- However, this limits the flexibility of the MDDB since the dimensions and aggregations are predefined.
- Examples: Crystal Holos, Essbase, Microsoft Analysis Services, Oracle Express, Cognos Powerplay

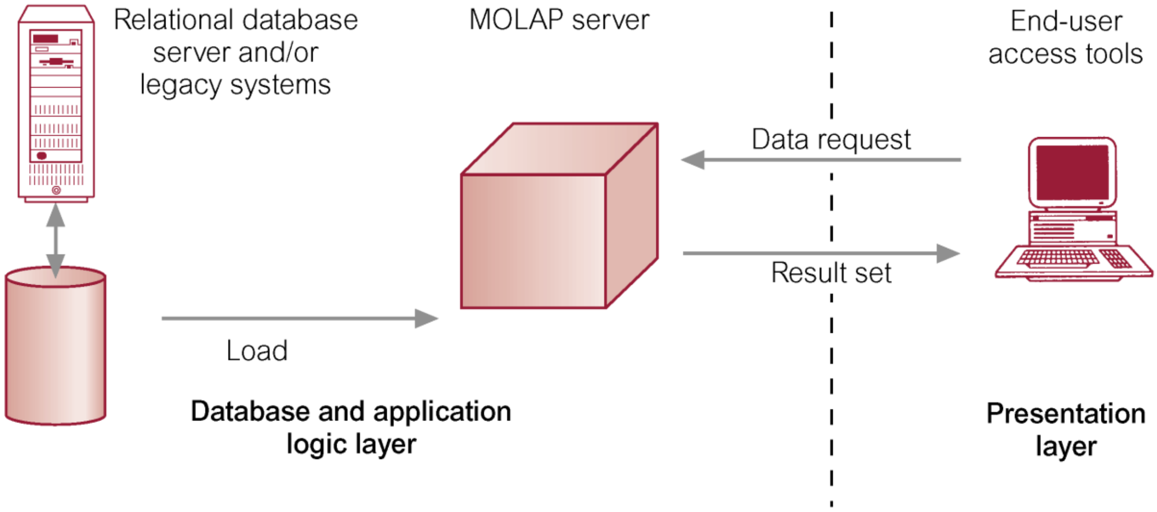

Relational database server and/or legacy systems
MOLAP server
End-user access tools
Data request
Result set
Load
Database and application logic layer
Presentation layer

Figure 5.3.2 : Typical Architecture of MOLAP

Features:

- Very fast response
- Ability to quickly write data into the cube

Downsides:

- Limited Scalability
- Inability to contain detailed data

5.3.3 Hybrid OLAP (HOLAP)

- The hybrid OLAP approach combines ROLAP and MOLAP technology, benefiting from the greater scalability of ROLAP and the faster computation of MOLAP.
- HOLAP server may allow large volumes of detail data to be stored in a relational database, while aggregations are kept in a separate MOLAP store.
- Example: Oracle Express, Seagate Holos, Speedware Media/M, Microsoft OLAP Services

5.3.4 Desktop OLAP (DOLAP)

- The desktop OLAP market resulted from the need for users to run business queries using relatively small data sets extracted from production systems.
- Most desktop OLAP systems were developed as extensions of production system report writers.
- Desktop OLAP systems are popular and typically require relatively little IT investment to implement.
- They also provide highly mobile OLAP operations for users who may work remotely or travel extensively.
- Examples: Brio.Enterprise, Business Objects, Cognos PowerPlay

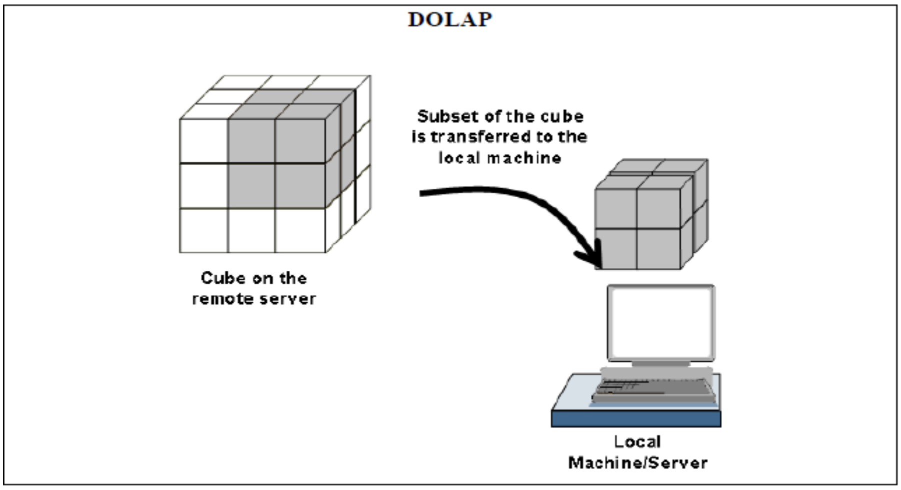

DOLAP
Subset of the cube is transferred to the local machine
Cube on the remote server
Local Machine/Server

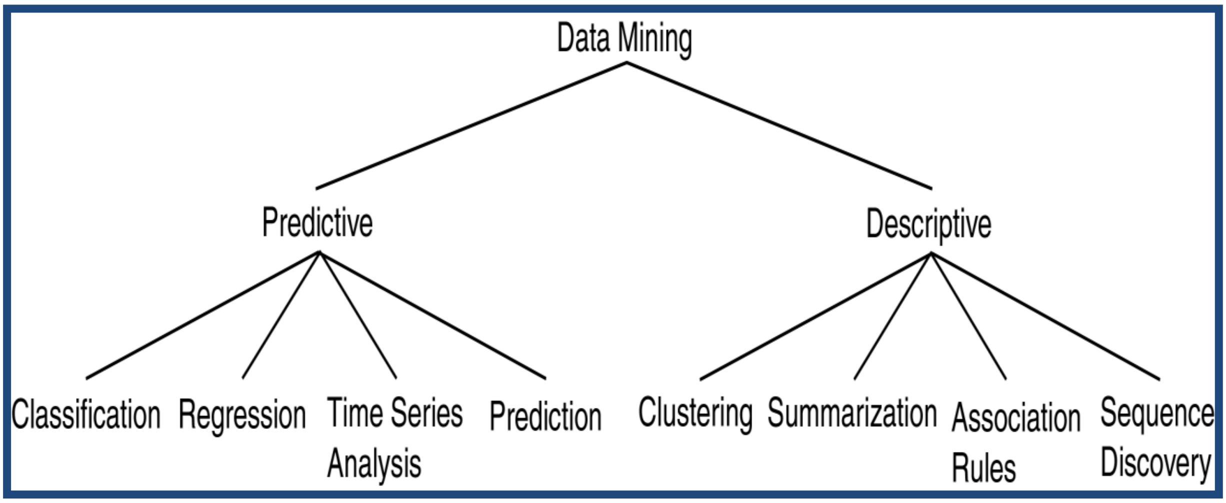
Figure 5.3.4: Concept of DOLAP

Chapter 6 Data Mining Approaches and Methods

6.1 Tasks of Data Mining

On the basis of the kind of data to be mined, there are two categories of functions involved in Data Mining:

- **Predictive**
- **Descriptive**



6.1.1 Descriptive Data Mining

It describes concepts or task-relevant datasets in concise, summarative, informative, discriminative forms. Descriptive Data Mining includes following functions:

- **Clustering**:
  It is referred as unsupervised learning or segmentation/partitioning. In clustering groups are not pre-defined.
- **Summarization**:
  Data is mapped into subsets with simple descriptions. Also termed as Characterization or generalization.
- **Sequence Discovery**:
  Sequential analysis or sequence discovery utilized to find out sequential patterns in data. Similar to association but relationship is based on time.
- **Association Rules**- A model which identifies specific types of data associations.

6.1.2 Predictive Data Mining

It is based on data and analysis, constructs models for the database, and predicts the trend and properties of unknown data. Predictive Data Mining includes following functions:

- **Classification:**
  Data is mapped into predefined groups or classes. It is also termed as supervised learning as classes are established prior to examination of data.
- **Regression:**
  Mapping of data item into known type of functions. These may be linear, logistic functions etc.
- **Time Series Analysis:**
  Value of an attribute are examined at evenly spaced times, as it varies with time.
- **Prediction:**
  It means fore telling future data states based on past and current data.

6.2 Supervised Vs Unsupervised Learning

Supervised learning algorithms are trained on **labelled** examples, i.e., input where the desired output is known. The supervised learning algorithm attempts to generalise a function or mapping from inputs to outputs which can then be used to speculatively generate an output for previously unseen inputs.

- Type and number of classes are known in advance
- Eg: Classification technique

Unsupervised learning algorithms operate on **unlabelled** examples, i.e., input where the desired output is unknown. Here the objective is to discover structure in the data (e.g. through a cluster analysis), not to generalise a mapping from inputs to outputs.

- Type and number of classes are **NOT** known in advance
- Eg: Clustering

6.3 Class/Concept Description

- Data can be associated with classes or concepts.

- For example, in the *AllElectronics store*, classes of items for sale include *computers and printers, and concepts of customers include bigSpenders and budgetSpenders.*

- Such descriptions of a class or a concept are called **class/concept descriptions.**

- These descriptions can be derived via:

1. **Data Characterization**
2. **Data Discrimination**
3. **Both Data Characterization and Data Description**

6.3.1 Data Characterization

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

- **Example**: The characteristics of customers who spend more than $1000 a year at XYZ store. The result can be a general profile such as age, employment status or credit ratings.

6.3.2 Data Discrimination

- Data discrimination is a comparison of the general features of target class data objects with the general features of objects from one or a set of contrasting classes.

- **Example**: The user may like to compare the general features of software products whose sales increased by 10% in the last year with those whose sales decreased by about 30% in the same duration.

6.4 Classification and Prediction

- Classification and prediction are two forms of data analysis that can be used to extract models describing important data classes or to predict future data trends.

- Whereas ***classification*** predicts categorical (discrete, unordered) labels, ***prediction*** models continuous valued functions.

- For example, we can build a classification model to categorize bank loan applications as either safe or risky, or a prediction model to predict the expenditures in dollars of potential customers on computer equipment given their income and occupation.

6.4.1 Classification

- Classification is supervised learning paradigm in which object are assigned into a predefined group or class based on a number of observed attributes related to that object.

- Classification algorithm finds the class of the unknown instance with the help of their attributes.

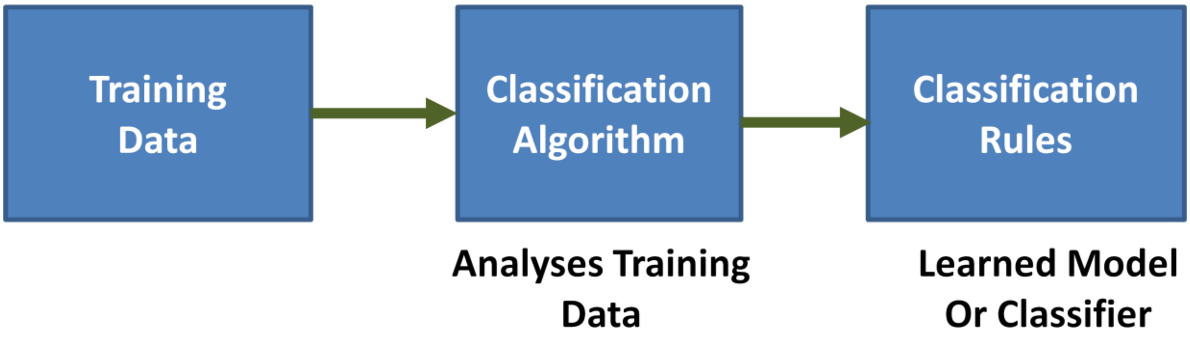
- **Application:** Stock market prediction, Weather forecasting, Bankruptcy prediction, Medical diagnosis, Speech recognition, Character recognitions.

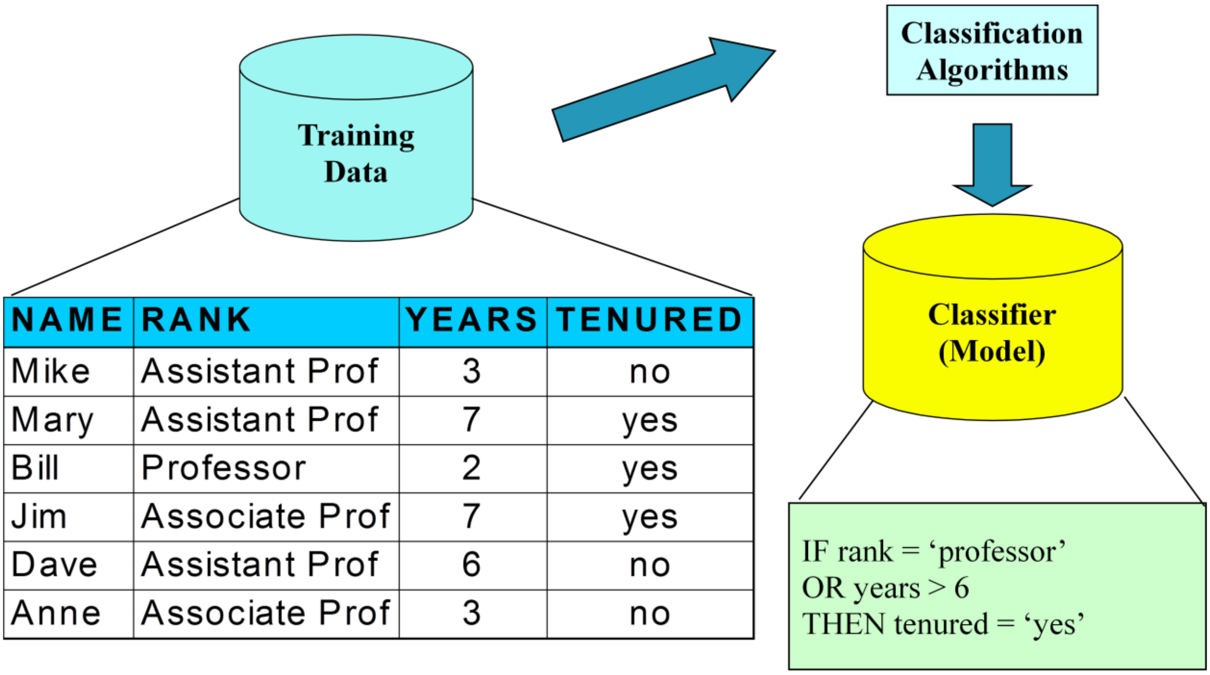
- The Data Classification process includes two steps:

    1. **Building the Classifier or Model**
    2. **Using Classifier for Classification**

Building the Classifier or Model (Model Construction)

- Training data are analyzed by a classification algorithm.
- A classifier is built describing a predetermined set of data classes or concepts.
- Also called as training phase or learning stage.



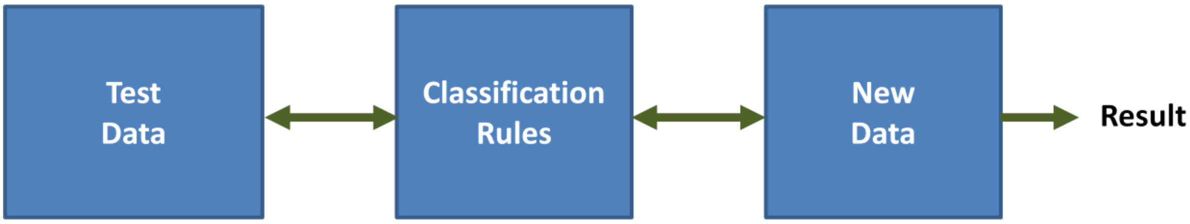


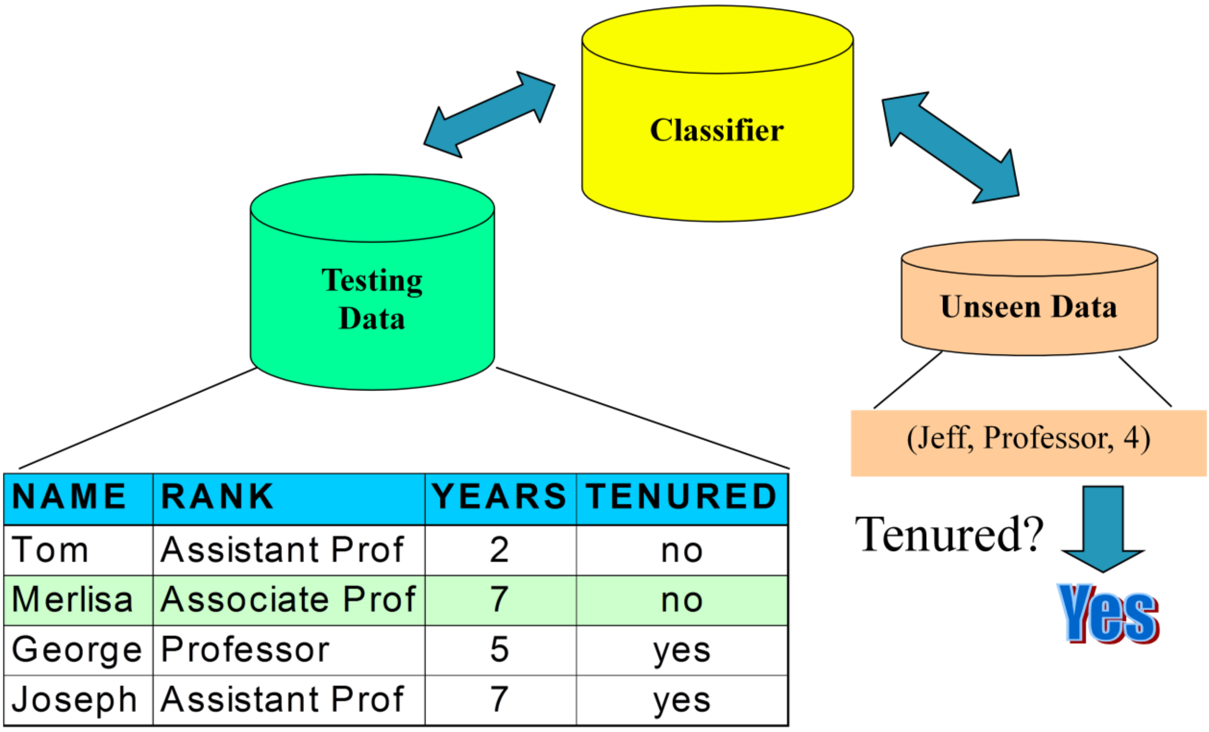
NAME	RANK	YEARS	TENURED
Mike	Assistant Prof	3	no
Mary	Assistant Prof	7	yes
Bill	Professor	2	yes
Jim	Associate Prof	7	yes
Dave	Assistant Prof	6	no
Anne	Associate Prof	3	no

IF rank = 'professor'
OR years > 6
THEN tenured = 'yes'

Using Classifier for classification (Model Usage)

- Test data are used to estimate the accuracy of the classification rules. If the accuracy is considered acceptable, the rules can be applied to the classification of new data tuples.





NAME	RANK	YEARS	TENURED
Tom	Assistant Prof	2	no
Merlisa	Associate Prof	7	no
George	Professor	5	yes
Joseph	Assistant Prof	7	yes

Examples of Classification Algorithms

- Decision Tree
- Bayesian Network
- Neural Network
- Genetic Algorithm
- K-Nearest Neighbor

6.4.1.1 Decision Tree

A decision tree is a flowchart like tree structure which consists:

- Root node: The top most node in the tree.
- Internal node: Denotes test on an attribute.
- Branch : Denotes the outcome of a test.
- Leaf node: Holds the class label.

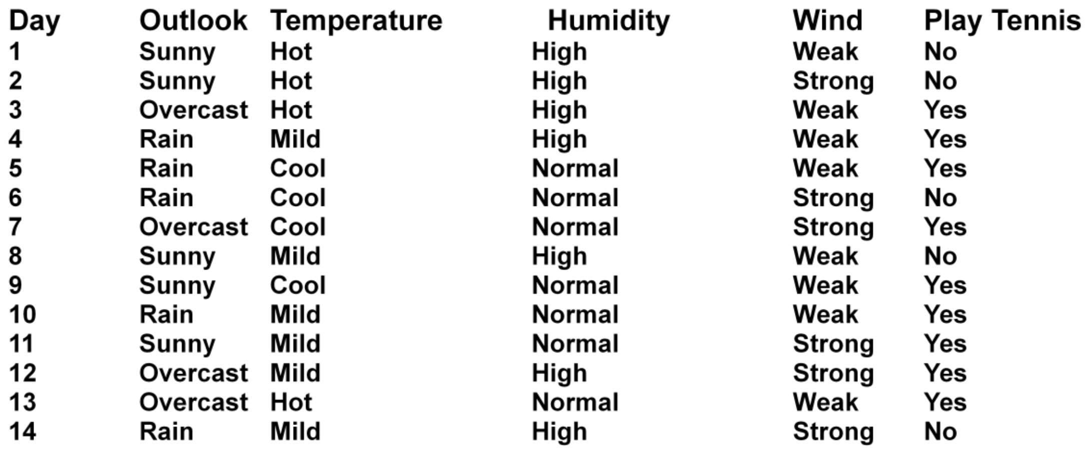
In order to classify an unknown sample the attribute values of the tuple are tested against the decision tree. A path is traced from the root to a leaf node, which holds the class prediction for that tuple. Decision trees can easily be converted to classification rules.

Decision tree generation consists of two phases:

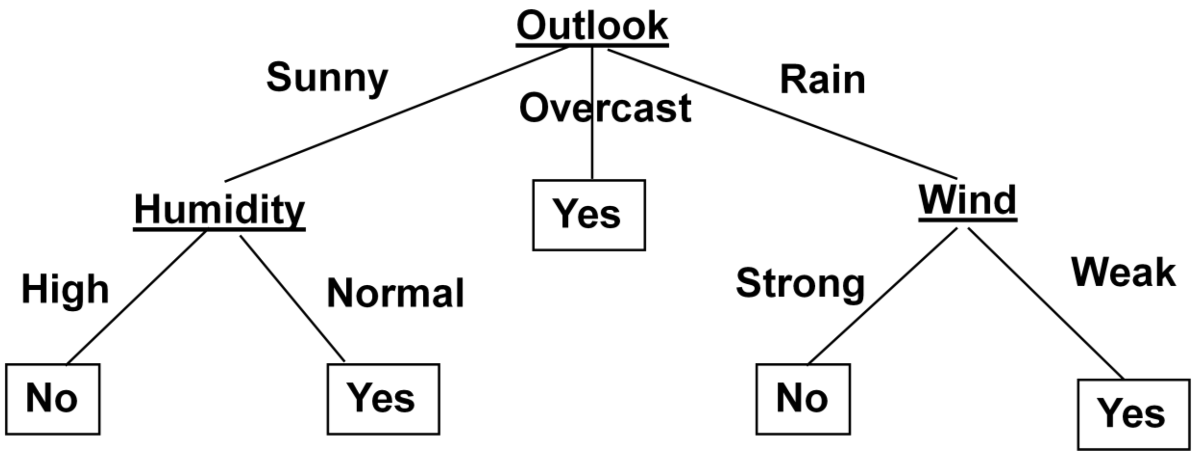
- **Tree construction**
  - At start, all the training examples are at the root

- Partition examples recursively based on selected attributes

- **Tree pruning**
  - Identify and remove branches that reflect noise or outliers
  - **Pre-pruning -** The tree is pruned by halting its construction early.
  - **Post-pruning -** This approach removes a sub-tree from a fully grown tree.



Day	Outlook	Temperature	Humidity	Wind	Play Tennis
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No



Attributes = {Outlook, Temperature, Humidity, Wind}   Play Tennis = {yes, no}

Examples of Decision Tree Algorithms: ID3, CART, C4.5

Decision Tree Induction (ID3) Algorithm

- **ID3 (Iterative Dichotomiser)** is a simple decision tree learning algorithm developed by Ross Quinlan (1983).
- ID3 follow non-backtracking approach in which decision trees are constructed in a top-down recursive “divide and conquer” manner to test each attribute at every tree node.
- This approach starts with a training set of tuples and their associated class labels.

- Training set is recursively partitioned into smaller subsets as the tree is being built.

Algorithm:

```
(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 labeled with the class C;
(4) if attribute list is empty then
(5)  return N as a leaf node labeled with the majority class in D; // majority voting
(6) select splitting-attribute, the attribute among attribute-list with the highest information gain;
(7) label node N with splitting attribute;
(9) attribute list = attribute list - splitting attribute; // remove splitting attribute
(10) for each outcome j of splitting attribute
(11)        partition the tuples and grow subtrees for each partition
(12) let Dj be the set of data tuples in D satisfying outcome j; // a partition
(13) if Dj is empty then
(14)        attach a leaf labeled with the majority class in D to node N;
(15) else attach the node returned by Generate decision tree(Dj, attribute list) to node    N;
```

Advantages of using ID3

- Understandable prediction rules are created from the training data.
- Builds the fastest tree.
- Builds a short tree.
- Only need to test enough attributes until all data is classified.
- Finding leaf nodes enables test data to be pruned, reducing number of tests.
- Whole dataset is searched to create tree.

Disadvantage of using ID3

- Data may be over-fitted or over-classified, if a small sample is tested.
- Only one attribute at a time is tested for making a decision.
- Classifying continuous data may be computationally expensive, as many trees must be generated to see where to break the continuum.

Pros and Cons of Decision Tree

Pros

- can handle real and nominal inputs
- speed and scalability
- robustness to outliers and missing values
- interpretability
- They are easy to use.
- Generated rules are easy to understand.

Cons

- several tuning parameters to set with little guidance
- decision boundary is non-continuous
- Cannot handle continuous data.
- Incapable of handling many problems which cannot be divided into attribute domains.
- Can lead to over-fitting as the trees are constructed from training data.

6.4.2 Prediction

- It is used to predict missing or unavailable numerical data values rather than class labels.
- Prediction can also be used for identification of distribution trends based on available data.
- **Regression Analysis** is generally used for prediction.

6.4.2.1 Regression Analysis

Regression analysis is used to model the relationship between one or more independent or predictor variables and a dependent or response variable. In the context of Data Mining, predictor variables are attributes of interest describing the tuple. Regression analysis can be divided into two categories:

- **Linear Regression**
- **Non-linear Regression**

Linear Regression

A linear regression technique approximates the relationship between the predictors and the target with a straight line. Linear Regression can be of two types: **Uni-variate Linear Regression** and **Multi-variate Linear Regression.**

Uni-variate Linear Regression

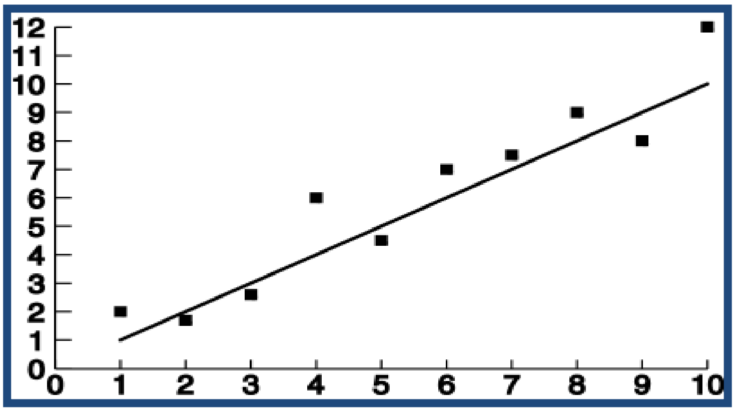
- Linear Regression which involves only one predictor variable (attribute) is known as Uni-variate Linear Regression.

- It has the form

$$y = a + bx$$

Where, y is response variable and x is single predictor variable, a and b are regression coefficients.

- These coefficients are solved by the method of least squares, which estimates the best fitting straight line as the one that minimizes the error between the actual data and the estimate of the line.



Multi-variate Linear Regression

- It involves more than one predictor variables (attributes).

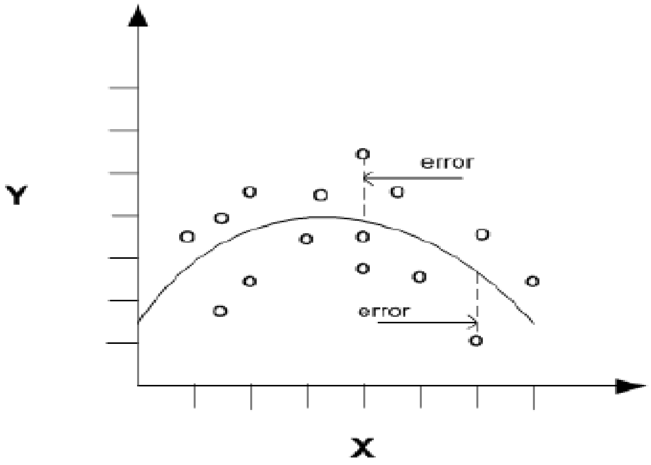
- It has the form

$$y = a_0 \;\; + a_1\, x_1 + a_2\, x_2 + a_3\, x_3 + \ldots.$$

Where (x_1 , x_2 , x_3 , ...) are predictor variables and (a_1 , a_2 , a_3 , ...) are regression coefficents.

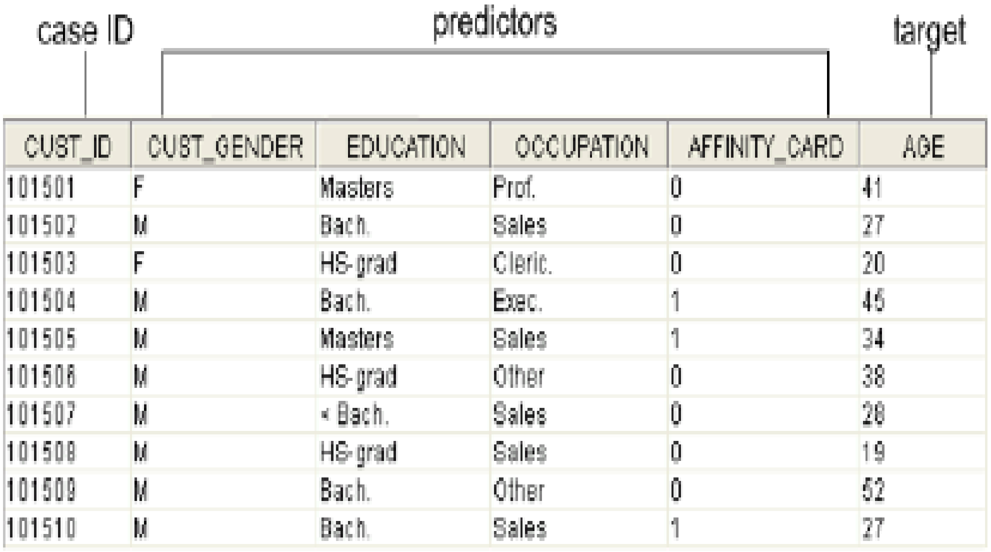
<u>Non-Linear Regression</u>

- In many cases the the relationship between x and *y cannot be approximated with a straight line.*
- For such cases, a nonlinear regression technique may be used.
- Nonlinear regression models define *y as a function of* ***x using an equation that is more*** complicated than the linear regression equation.



Regression Problem Example

Suppose to learn more about the purchasing behavior of customers of different ages. Building a model to predict the ages of customers as a function of various demographic characteristics and shopping patterns is Regression problem since the model will predict a number (age).


case ID
predictors
target

CUST_ID	CUST_GENDER	EDUCATION	OCCUPATION	AFFINITY_CARD	AGE
101501	F	Masters	Prof.	0	41
101502	M	Bach.	Sales	0	27
101503	F	HS-grad	Cleric.	0	20
101504	M	Bach.	Exec.	1	46
101505	M	Masters	Sales	1	34
101506	M	HS-grad	Other	0	38
101507	M	« Bach.	Sales	0	28
101508	M	HS-grad	Sales	0	19
101509	M	Bach.	Other	0	52
101510	M	Bach.	Sales	1	27

6.4.3 Issues regarding classification and prediction

Issues (1): Data Preparation

- Data cleaning

  Preprocess data in order to reduce noise and handle missing values

- Relevance analysis (feature selection)

  Remove the irrelevant or redundant attributes

- Data transformation

  Generalize and/or normalize data

Issues (2): Evaluating Classification Methods

- ❖ Predictive accuracy
- ❖ Speed and scalability

  time to construct the model

  time to use the model
- ❖ Robustness

  handling noise and missing values
- ❖ Scalability

  efficiency in disk-resident databases
- ❖ Interpretability

  understanding and insight provided by the model
- ❖ Goodness of rules

  decision tree size

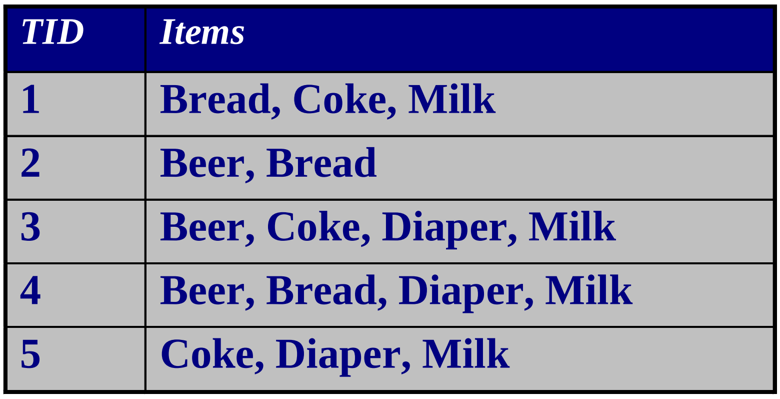
  compactness of classification rules

6.5 Association Rule Mining

- It is an important data mining model studied extensively by the database and data mining community.
- It was proposed by Proposed by Agrawal et al in 1993.
- Initially used for Market Basket Analysis to find how items purchased by customers are related.
- It produces dependency rules which will predict occurrence of an item based on occurrences of other items.

Some Basic Terms

- **Itemsets:** An itemset is a set of items.
  E.g., X = {milk, bread, coke} is an itemset.
- A ***k*-itemset** is an itemset with *k* items.
  E.g., {milk, bread, coke} is a **3-itemset**
- **Support count (σ):** Frequency of occurrence of and itemset
  σ ({milk, coke}) = 3
  σ ({ beer, Diaper}) = 2
- **Support:** Fraction of transactions that contain an itemset
- s ({milk, coke}) = 3/5



TID	*Items*
1	**Bread, Coke, Milk**
2	**Beer, Bread**
3	**Beer, Coke, Diaper, Milk**
4	**Beer, Bread, Diaper, Milk**
5	**Coke, Diaper, Milk**

s ({beer, diapher}) = 2/5

- **Frequent itemset:** An itemset whose support is greater than or equal to a minimum support threshold (minsup)
- ***Frequent items are represented in the form of association rules.***

What's an Association Rule?

- An association rule is an implication of two itemsets:

$$X \Rightarrow Y$$

- To measure the interestingness of association rules two measures are used:
  - **<u>Support(s)</u>:** The occurring frequency of the rule, i.e., percentage of transactions that contain both X and Y

    $$s = \frac{\sigma(X \cup Y)}{\#\text{ of trans.}}$$

    = P(XUY)

    Example: s(beer-> diapher) = 2/5
  - **<u>Confidence(c)</u>:** The strength of the association, i.e, is the percentage of transactions *containing X that also contain Y*

    $$c = \frac{\sigma(X \cup Y)}{\sigma(X)}$$

    = P (Y/X)

    Example: c(beer->diapher) = 2/3

<u>Association Rules Mining Steps</u>

In general, association rule mining can be viewed as a two-step process:

1. **Find all frequent itemsets:** Each of these itemsets will occur at least as frequently as a predetermined minimum support count, *min sup*.

2. **Generate strong association rules from the frequent itemsets:** These rules must satisfy minimum support and minimum confidence.

Example of Association Rule mining technique: Apriori Algorithm.

6.5.1 Apriori Algorithm

- Apriori is a seminal algorithm proposed by R. Agrawal and R. Srikant in 1994 for mining frequent itemsets.
- The Apriori principle: Any subset of a frequent itemset must be frequent.

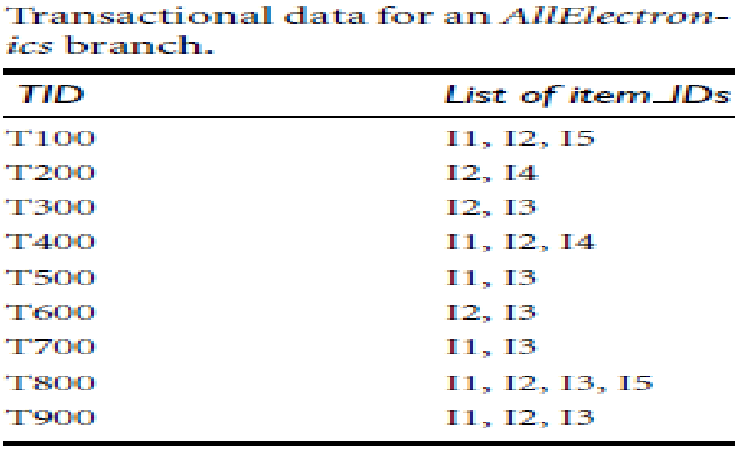
Steps:

- **Join Step**:
  Set of candidate k-itemsets (C_k) is generated by joining L_{k-1} with itself.
- **Prune Step:**
  Generate L_k by selecting the candidates from C_k having a count no less than the minimum support count.

Pseudo-code

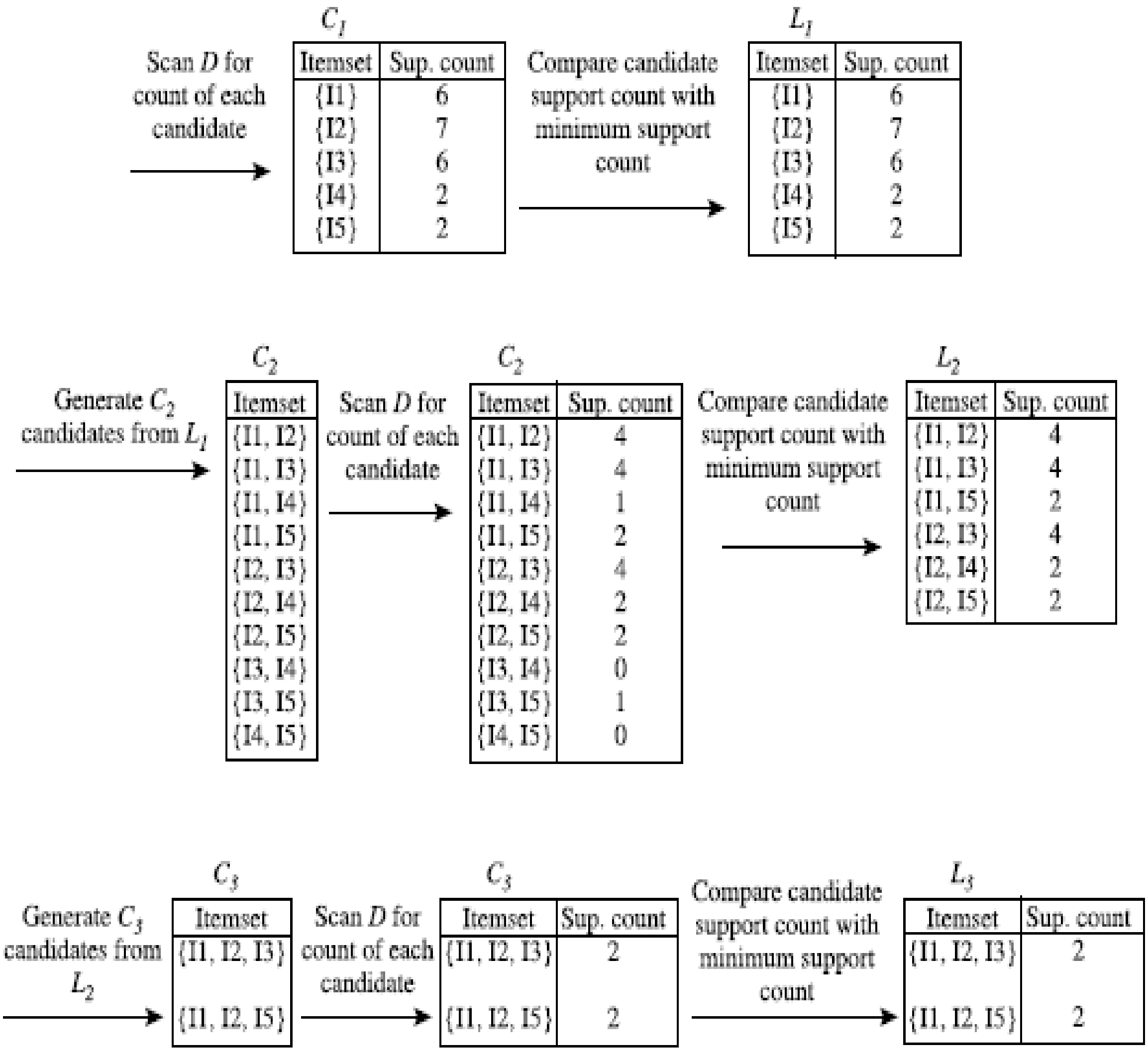
```
$C_k$: Candidate itemset of size k
$L_k$ : frequent itemset of size k
$L_1$ = {frequent items};
for ($k$ = 1; $L_k$ !=$\emptyset$; $k$++) do begin
        $C_{k+1}$ = candidates generated from $L_k$;
        for each transaction $t$ in database do
                increment the count of all candidates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

Example:



Transactional data for an *AllElectronics* branch.

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


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

C_1

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

L_1

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

C_2

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

C_2

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

L_2

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

C_3

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

C_3

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

L_3

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

6.6 Clustering

- A cluster is a collection of data objects that are *similar to one another* within the same cluster and are *dissimilar to the objects in other clusters.*
- The process of grouping a set of physical or abstract objects into classes of *similar objects* is called clustering.
- Clustering is also called data segmentation in some applications because clustering partitions large data sets into groups according to their *similarity.*

6.6.1 Types of Clustering

Clustering methods can be classified into the following categories:

- ❖ Partitioning Method
- ❖ Hierarchical Method

- ❖ Density-based Method
- ❖ Grid-Based Method
- ❖ Model-Based Method
- ❖ Constraint-based Method

Partitioning Method

- It data objects into non-overlapping subsets (clusters) such that each data object is in exactly one subset.
- It construct various partitions and then evaluate them by some criterion, e.g., minimizing the sum of square errors.
- Typical methods: k-means, k-medoids, CLARA (Clustering LARge Applications)

Hierarchical Method

- Groups the data objects into a tree of clusters.
- Two types: Agglomerative & Divisive.
- Typical methods: DiAna (Divisive Analysis), AgNes (Agglomerative Nesting), BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies), ROCK (RObust Clustering using linKs), CAMELEON
- **Agglomerative:**
  Works on bottom up approach by placing each object in its own cluster and then merges these atomic clusters into larger and larger clusters, until all of the objects are in a single cluster or until certain termination conditions are satisfied.
- **Divisive**:
  Works on top down approach by placing all objects into one cluster and subdivides the cluster into smaller and smaller pieces, until each object forms a cluster on its own or until it satisfies certain termination conditions.

Density-based Method

- Based on connectivity and density functions.
- Typical methods: DBSACN (Density Based Spatial Clustering of Applications with Noise), OPTICS (Ordering Points To Identify the Clustering Structure), DenClue (DENsity-based CLUstEring).

Grid-Based Method

- It quantizes the object space into a finite number of cells that form a grid structure on which all of the operations for clustering are performed.
- Typical methods: STING (STatistical INformation Grid), WaveCluster, CLIQUE (Clustering In QUEst).

Model-Based Method

- Model-based clustering methods attempt to optimize the fit between the given data and some mathematical model.

- Typical methods: EM (Expectation Maximization), SOM (Self-Organizing Map), COBWEB

Constraint-based Method

- Constraint-based clustering finds clusters that satisfy user-specified preferences or constraints.
- Typical methods: COD, constrained clustering

K-means Clustering

- The *k-means algorithm takes the input parameter, k, and partitions a set of n objects into k clusters so that the resulting intracluster similarity is high but the intercluster similarity* is low.

- **Algorithm:** *k*-means. The *k*-means algorithm for partitioning, where each cluster's center is represented by the mean value of the objects in the cluster.

  Input:

  k: the number of clusters,

  D: a data set containing *n* objects.

  Output: A set of *k* clusters.

  Method:

```
(1) arbitrarily choose k objects from D as the initial cluster centers;
(2) repeat
(3) (re)assign each object to the cluster to which the object is the most similar, based
      on the mean value of the objects in the cluster;
(4) update the cluster means, i.e., calculate the mean value of the objects for each
     cluster;
(5) until no change
```

Issues and Limitations of K-means Clustering

- How to choose initial centers?
- How to choose K?
- How to handle Outliers?
- Clusters different in

- Shape
- Density
- Size

Pros and Cons of K-means Algorithm

Pros

- *Simple*
- *Fast for low dimensional data*

Cons

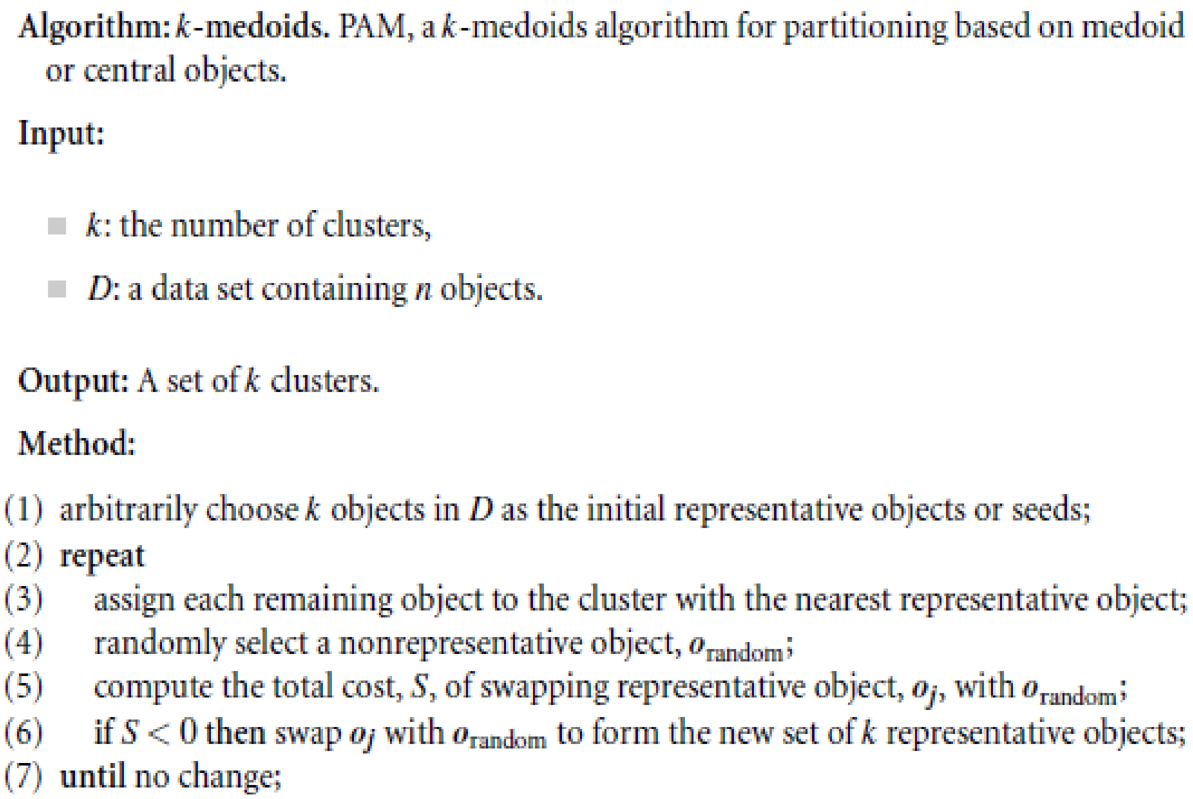
- *K-Means will not identify outliers*
- *K-Means is restricted to data which has the notion of a center (centroid)*
- *Applicable only when mean is defined, then what about categorical data?*
- *Need to specify k, the number of clusters, in advance*
- *Unable to handle noisy data and outliers*

<u>K-mediod Clustering</u>

- The *k-means algorithm is sensitive to outliers.*
- *To diminish such sensivity, K-mediods pick* actual objects as a reference point to represent the clusters, rather than the mean value of the clusters.
- The partitioning method is then performed based on the principle of minimizing the sum of the dissimilarities between each object and its corresponding reference point.
- That is, an absolute-error criterion is used, defined as

  $$E = \sum_{j=1}^{k} \sum_{p \in C_j} |p - o_j|,$$

  where *E is the sum of the absolute error for all objects in the data set; p is the point in* space representing a given object in cluster*Cj; and oj is the representative object of Cj.*
- Algorithm iterates until, eventually, each representative object is actually the medoid, or most centrally located object, of its cluster.



Algorithm: *k*-**medoids.** PAM, a *k*-medoids algorithm for partitioning based on medoid or central objects.

Input:

- *k*: the number of clusters,
- *D*: a data set containing *n* objects.

Output: A set of *k* clusters.

Method:

```
arbitrarily choose $k$ objects in $D$ as the initial representative objects or seeds;
repeat
    assign each remaining object to the cluster with the nearest representative object;
    randomly select a nonrepresentative object, $o_{random}$;
    compute the total cost, $S$, of swapping representative object, $o_j$, with $o_{random}$;
    if $S < 0$ then swap $o_j$ with $o_{random}$ to form the new set of $k$ representative objects;
until no change;
```

6.6.2 Applications of Cluster Analysis

- Pattern Recognition
- Spatial Data Analysis
  - Create thematic maps in GIS by clustering feature spaces
  - Detect spatial clusters or for other spatial mining tasks
- Image Processing
- Economic Science (especially market research)
- WWW
  - Document classification
  - Cluster Weblog data to discover groups of similar access patterns
- customer bases, and then use this knowledge to develop targeted marketing programs
- Land use: Identification of areas of similar land use in an earth observation database
- Insurance: Identifying groups of motor insurance policy holders with a high average claim cost

- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults

6.7 Data Mining Tools

There are no. of data mining tools available in the market. Some of them are described below:

WEKA

- (Waikato Environment for Knowledge Analysis) is a popular suite of machine learning software written in Java, developed at the University of Waikato, New Zealand.
- WEKA is free software available under the GNU General Public License.
- **Features:**
  - Written in JAVA
  - Has graphical user interfaces
  - Contains a collection of visualization tools and algorithms for data analysis and predictive modeling
  - Supports standard data mining tasks like data preprocessing, clustering, classification, regression, visualization, and feature selection
- **Usage:**
  - Apply a learning method to a dataset & analyze the result
  - Use a learned model to make predictions on new instances
  - Apply different learners to a dataset & compare results

Microsoft SQL Server 2005

- Integrate DB and OLAP with mining
- Support OLEDB for DM standard

IBM Intelligent Miner

- Intelligent Miner is an IBM data-mining product
- A wide range of data mining algorithms
- Scalable mining algorithms
- Toolkits: neural network algorithms, statistical methods, data preparation, and data visualization tools
- Tight integration with IBM's DB2 relational database system

SAS Enterprise Miner

- SAS Institute Inc. developed Enterprise Miner
- A variety of statistical analysis tools
- Data warehouse tools and multiple data mining algorithms

SGI MineSet

- Silicon Graphics Inc. (SGI) developed MineSet
- Multiple data mining algorithms and advanced statistics
- Advanced visualization tools

DBMiner

- DBMiner Technology Inc developed DBMiner.
- It provides multiple data mining algorithms including discovery-driven OLAP analysis, association, classification, and clustering

SPSS Clementine

- Integral Solutions Ltd. (ISL) developed Clementine
- Clementine has been acquired by SPSS Inc.
- An integrated data mining development environment for end-users and developers
- Multiple data mining algorithms and visualization tools including rule induction, neural nets, classification, and visualization tools

Chapter 7 Mining Complex Types of Data

In previous studies data mining techniques have focused on mining relational databases, transactional databases, and data warehouses formed by the transformation and integration of structured data. Vast amount of data in various complex forms (e.g., structured and unstructured, hypertext and multimedia) have been growing explosively owing to the rapid progress of data collection tools, advanced database system technologies and World –Wide Web (WWW) technologies. Therefore, an increasingly important task in data mining is to mine complex types of data.

Complex types of Data include:

- Object data
- Spatial data
- Multimedia data
- Time-series data
- Text data
- Web data

7.1 Mining Spatial Data

Spatial Data

- Spatial data refer to any data about objects that occupy real physical space.
- Spatial data can contain both spatial and non-spatial features.
- Spatial information includes geometric metadata (e.g., location, shape, size, distance, area, perimeter) and topological metadata (e.g., "neighbor of", "adjacent to", "included in", "includes").
- Non-spatial attributes are used to characterize non-spatial features of objects, such as name, population, and unemployment rate for a city.

Spatial Database

- Spatial Database is the repository of spatial data.
- It stores a large amount of space-related data, such as maps, preprocessed remote sensing or medical imaging data, and VLSI chip layout data.

Spatial Data Mining

- Spatial data mining refers to the extraction of knowledge, spatial relationships, or other interesting patterns not explicitly stored in spatial database.

- It can be used for understanding spatial data, discovering spatial relationships and relationships between spatial and nonspatial data, constructing spatial knowledge bases, reorganizing spatial databases, and optimizing spatial queries.

Applications

- Geographic information systems
- Geomarketing
- Remote sensing
- Image database exploration
- Medical imaging
- Navigation
- Traffic control
- Environmental studies

Spatial Data Mining Tasks

- Spatial classification
- Spatial Trend Analysis
- Spatial clustering
- Spatial association rules analysis

Spatial Classification

- Spatial classification analyzes spatial objects to derive classification schemes in relevance to certain spatial properties, such as the neighbourhood of a district, highway, or river.
- Example: To classify regions in a province into rich versus poor according to the average family income several properties of the spatial objects are analyzed such as hosting a university, containing interstate highways, being near a lake or ocean, and so on.
- Uses conventional supervised learning algorithms
  - e.g., Decision trees

Spatial Trend Analysis

- Spatial trend analysis deals with the detection of changes and trends along a spatial dimension.
- It is used to analyse the patterns that changes with space and time.
- Spatial trend analysis replaces time with space and studies the trend of nonspatial or spatial data changing with space.
- Such analysis can be done using regression and correlation analysis.
- Example:
- Trend of changes in economic situation when moving away from the center of a city.
- Trend of changes of the climate or vegetation with the increasing distance from an ocean.

Spatial Clustering

- Spatial clustering is a process of grouping a set of spatial objects into clusters so that objects within a cluster have high similarity in comparison to one another, but are dissimilar to objects in other clusters.
- Clustering is used to determine the “hot spots” in crime analysis and disease tracking.

Spatial Association rules Analysis

- A spatial association rule is of the form $A \Rightarrow B$ [s%,c%], where A and B are sets of spatial or nonspatial predicates, s% is the support of the rule, and c% is the confidence of the rule.
- For example, the following is a spatial association rule:

  is_a(*X*,“*school*”)∧ *close_to*(*X*,“*sports center*”) ⇒ *close_to*(*X*,“*park*”) [0.5%,80%].

  This rule states that 80% of schools that are close to sports centers are also close to parks, and 0.5% of the data belongs to such a case.
- Progressive refinement technique is used for Spatial association analysis.

7.2 Multimedia Data Mining

- Multimedia data mining is used for extracting interesting information for multimedia data sets, such as audio, video, images, graphics, speech, text and combination of several types of data set which are all converted from different formats into digital media.

- Multimedia mining is a subfield of data mining which is used to find interesting information of implicit knowledge from multimedia databases.

- Multimedia data are classified into five types:

  - ❖ Text data
  - ❖ Image data
  - ❖ Audio data
  - ❖ Video data and
  - ❖ Electronic and digital ink

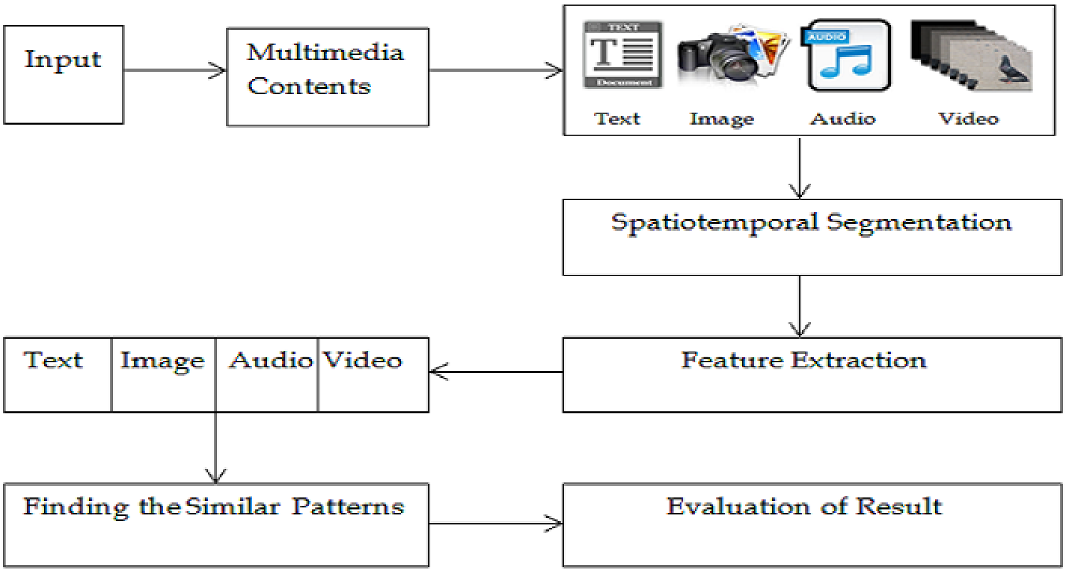
Architecture of Multimedia Data Mining

It consists of following components:

- ***Input*** stage comprises multimedia database iused for finding the patterns and to perform data mining process.

- ***Multimedia Content*** is the data selection stage which requires the user to select the databases, subset of fields or data to be used for data mining.

- ***Spatio-temporal segmentation*** is the process of changing videos to image sequence and it is useful for object segmentation.



- ***Feature extraction*** is the pre-processing step that involves integrating data from various sources and making choices regarding characterizing or coding certain data fields to serve when inputs to the pattern finding stage.

- ***Finding the similar pattern*** stage uncovered the hidden pattern and trends. Some approaches of finding similar pattern stage contain association, classification, clustering, regression, time-series analysis and visualization.

- ***Evaluation of Results*** is a data mining process used to evaluate the results

<u>Multimedia Data Mining Tasks</u>

- **Classification:** Hidden Markov Model used for classifying the multimedia data such as images and video.

- **Clustering:** In multimedia mining, clustering technique can be applied to group similar images, objects, sounds, videos and texts.

- **Association:** There are three different types of associations in multimedia mining:
    - Associations between image content and non-image content features
    - Associations among image contents that are not related to spatial relationships
    - Associations among image contents that are not related to spatial relationships

7.3 Text Mining

- **Text mining** is the procedure of synthesizing information, by analyzing relations, patterns, and rules among textual data.

- Text mining system can be of following types based on the types of input they take:

- **Keyword based Approach:** Input is keyword or terms in document
- **Tagging Approach:** Input is set of tags
- **Information-extraction Approach:** Input is semanatic information such as events, facts or entities

Text Mining Tasks

- Document Clustering
- Classification
- Information extraction
- Association Analysis
- Trend Analysis

Document Classification

- Document Classification organises documents into classes to facilitate document retrieval and subsequent analysis.
- Document classification has been used in automated topic tagging (i.e., assigning labels to documents), topic directory construction, identification of the document writing styles etc.
- Common Classificaiton method: Nearest-neighbor classification, Feature selection methods, Bayesian classification, Support vector machines, and association based classification.

Document Clustering

- Document clustering is one of the most crucial techniques for organizing documents in
  an unsupervised manner.
- Common methods: Spectral clustering, mixture model clustering, clustering using Latent Semantic Indexing, and clustering using Locality Preserving Indexing.

Association Analysis

- Association analysis collects keywords or terms that occur frequently together and find association or co-relationship among them.

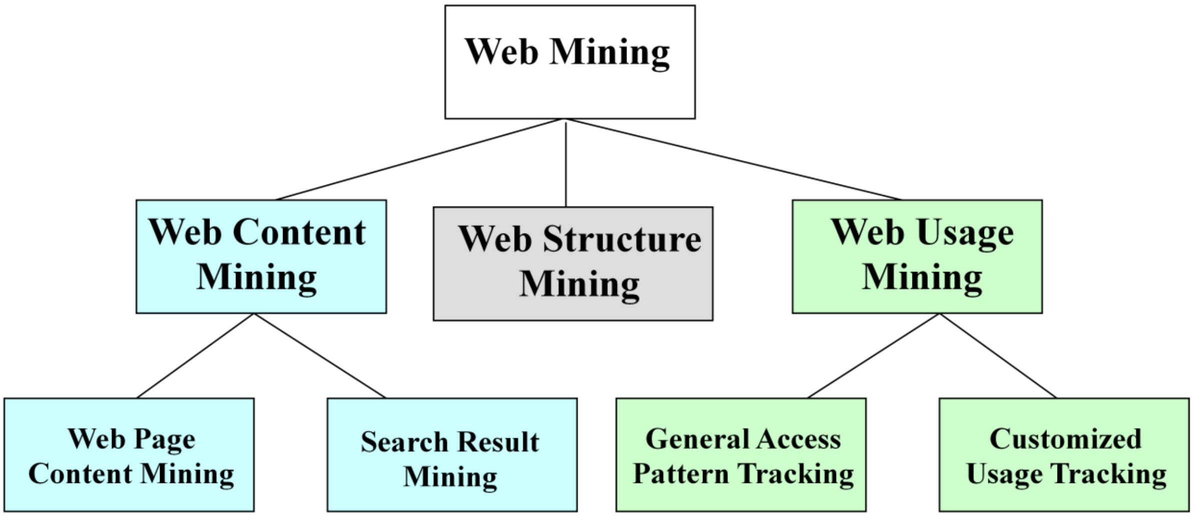
7.4 Web Mining

- The term **Web Mining** was coined by Orem Etzioni (1996) to denote the use of data mining techniques to automatically discover Web documents and services, extract information from Web resources, and uncover general patterns on the Web.
- The World Wide Web is a rich, enormous knowledge base that can be useful to many applications. The WWW is huge, widely distributed, global information service centre for news, advertisements, consumer information, financial management, education, government, e-commerce, hyperlink information, access and usage information.

Web Mining Taxonomy

Web Mining research can be classified into three categories:

- **Web content mining** refers to the discovery of useful information from Web contents, including text, images, audio, video, etc.
- **Web structure mining** studies the model underlying the link structures of the Web. It has been used for search engine result ranking and other Web applications.
- **Web usage mining** focuses on using data mining techniques to analyze search logs to find interesting patterns. One of the main applications of Web usage mining is its use to learn user profiles.



Chapter 8 Research Trends in Data Warehousing and Data Mining

8.1 Data Mining Systems Products and Research Prototypes

As a discipline, data mining has a relatively short history and is constantly evolving, efforts toward the standardization of data mining language are still underway.

How to Choose a Data Mining System?

Data mining systems should be assessed based on the following multiple features:

- **Data type:** Type of the data you are going to mine. (Cateogorical, numerical, spatial, time-series data, stream data, biological data, web data etc).
- **System issues:**
  - Type of the OS you want to run
  - Architecture (Client-server)
- **Data Source:**
  - Depends upon the type of the Data formats used by the Data mining system
  - ASCII text files, relational data, or data warehouse data
- **Data mining functions and methodologies:**
  - Depending upon the types of the data mining functions and methodologies provided
- **Coupling data mining with database and/or data warehouse systems**
- **Scalability**
  - Depending on how the system performs when size and attributes of dataset increases.
- **Visualisation tools**
  - Depends upon how effectively the results are provided to the user.
- **Data mining query language and graphical user interface**

8.2 Theoretical Foundations of Data Mining

Several theories for the basis of data mining include the following:

- **Data reduction**
  - The basis of data mining is to reduce the data representation
  - Trades accuracy for speed in response
- **Data compression**
  - The basis of data mining is to compress the given data by encoding in terms of bits, association rules, decision trees, clusters, etc.

- ❖ **Pattern discovery**

    - The basis of data mining is to discover patterns occurring in the database, such as associations, classification models, sequential patterns, etc.

- ❖ **Probability theory**

    - The basis of data mining is to discover joint probability distributions of random variables

- ❖ **Microeconomic view**

    - A view of utility: the task of data mining is finding patterns that are interesting only to the extent in that they can be used in the decision-making process of some enterprise

- ❖ **Inductive databases**

    - Data mining is the problem of performing inductive logic on databases,
    - The task is to query the data and the theory (i.e., patterns) of the database
    - Popular among many researchers in database systems

8.3 Statistical Data Mining

There are many well-established statistical techniques for data analysis, particularly for numeric data. Some of them are:

Regression:

- predict the value of a response (dependent) variable from one or more predictor (independent) variables where the variables are numeric

Generalized linear models

- allow a categorical response variable (or some transformation of it) to be related to a set of predictor variables
- include logistic regression and Poisson regression

Mixed-effect models

- describe relationships between a response variable and some covariates in data grouped according to one or more factors

Analysis of variance

- Analyze experimental data for two or more populations described by a numeric response variable and one or more categorical variables (factors)

Factor analysis

- determine which variables are combined to generate a given factor
- e.g., for many psychiatric data, one can indirectly measure other quantities (such as test scores) that reflect the factor of interest

Discriminant analysis

- predict a categorical response variable, commonly used in social science
- Attempts to determine several discriminant functions (linear combinations of the independent variables) that discriminate among the groups defined by the response variable

8.4 Visual and Audio Data Mining

Visual Data Mining

- Visual data mining discovers implicit and useful knowledge from large data sets using data and/or knowledge visualization techniques.
- Visual data mining can be viewed as an integration of two disciplines: data visualization and data mining.
- In general, data visualization and data mining can be integrated in the following ways:
    - Data visualization
    - Data mining result visualization
    - Data mining process visualization
    - Interactive visual data mining

Data visualization

- Data can be presented in various visual forms, such as boxplots, 3-D cubes, data distribution charts, curves, surfaces, link graphs, and so on.

Data mining result visualization

- Visualization of data mining results is the presentation of the results or knowledge obtained from data mining in visual forms. Eg: scatter plots and boxplots as well as decision trees, association rules, clusters, outliers

Data mining process visualization

- This type of visualization presents the various processes of data mining in visual forms so that users can see how the data are extracted and from which database or data warehouse they are extracted,as well as how the selected data are cleaned, integrated, preprocessed, and mined**.**

Interactive visual data mining

- Visualization tools can be used in the data mining process to help users make smart data mining decisions.

Audio Data Mining

- Audio data mining uses audio signals to indicate the patterns of data or the features.
- Visual Mining requires users to concentrate on watching patterns and identifying interesting or novel features within them. This can sometimes be quite tiresome. If patterns can be transformed into sound and music, then instead of watching pictures, we can listen to pitches, rhythms, tune, and melody in order to identify anything interesting or unusual.
- Therefore, audio data mining is an interesting complement to visual mining.

8.5 Data Mining and Collaborative Filtering

- A collaborative filtering approach is commonly used, in which products are recommended based on the opinions of other customers.
- Collaborative recommender systems may employ data mining or statistical techniques to search for similarities among customer preferences.
- A collaborative recommender system works by finding a set of customers, referred to as *neighbors*, that have a history of agreeing with the target customer (such as, they tend to buy similar sets of products, or give similar ratings for certain products).

8.6 Social Impact of Data Mining

- **Social Impacts: Threat to Privacy**
  - Profiling information is collected every time
    - You use your credit card, debit card, supermarket loyalty card, or frequent flyer card, or apply for any of the above
    - You surf the Web, reply to an Internet newsgroup, subscribe to a magazine, rent a video, join a club, fill out a contest entry form,
    - You pay for prescription drugs, or present you medical care number when visiting the doctor
  - Collection of personal data may be beneficial for companies and consumers, there is also potential for misuse

- **Protect Privacy and Data Security**
    - Fair information practices
        - International guidelines for data privacy protection
        - Cover aspects relating to data collection, purpose, use, quality, openness, individual participation, and accountability
        - Purpose specification and use limitation
        - Openness: Individuals have the right to know what information is collected about them, who has access to the data, and how the data are being used
    - Develop and use data security-enhancing techniques
        - Blind signatures
        - Biometric encryption
        - Anonymous databases

8.6 Trends in Data Mining

- **Application exploration**
    - development of application-specific data mining system
    - Invisible data mining (mining as built-in function)
- **Scalable data mining methods**
    - Constraint-based mining: use of constraints to guide data mining systems in their search for interesting patterns
- **Integration of data mining with database systems, data warehouse systems, and Web database systems**
- **Standardization of data mining language**
    - A standard will facilitate systematic development, improve interoperability, and promote the education and use of data mining systems in industry and society
- **Visual data mining**
- **New methods for mining complex types of data**
    - More research is required towards the integration of data mining methods with existing data analysis techniques for the complex types of data
- **Web mining**
- **Privacy protection and information security in data mining**