AIT 411 DATA MINING AND BUSINESS INTELLIGENCE (2+0)

(2015 SYLLABUS)

Lecture Notes

Course Teacher

Dr. J. Arockia Stephen Raj
Associate Professor (Computer Science)

Department of Physical Sciences & Information Technolgy
Tamil Nadu Agricultural University
Coimbatore - 641 003

Unit I – Introduction to Data Mining

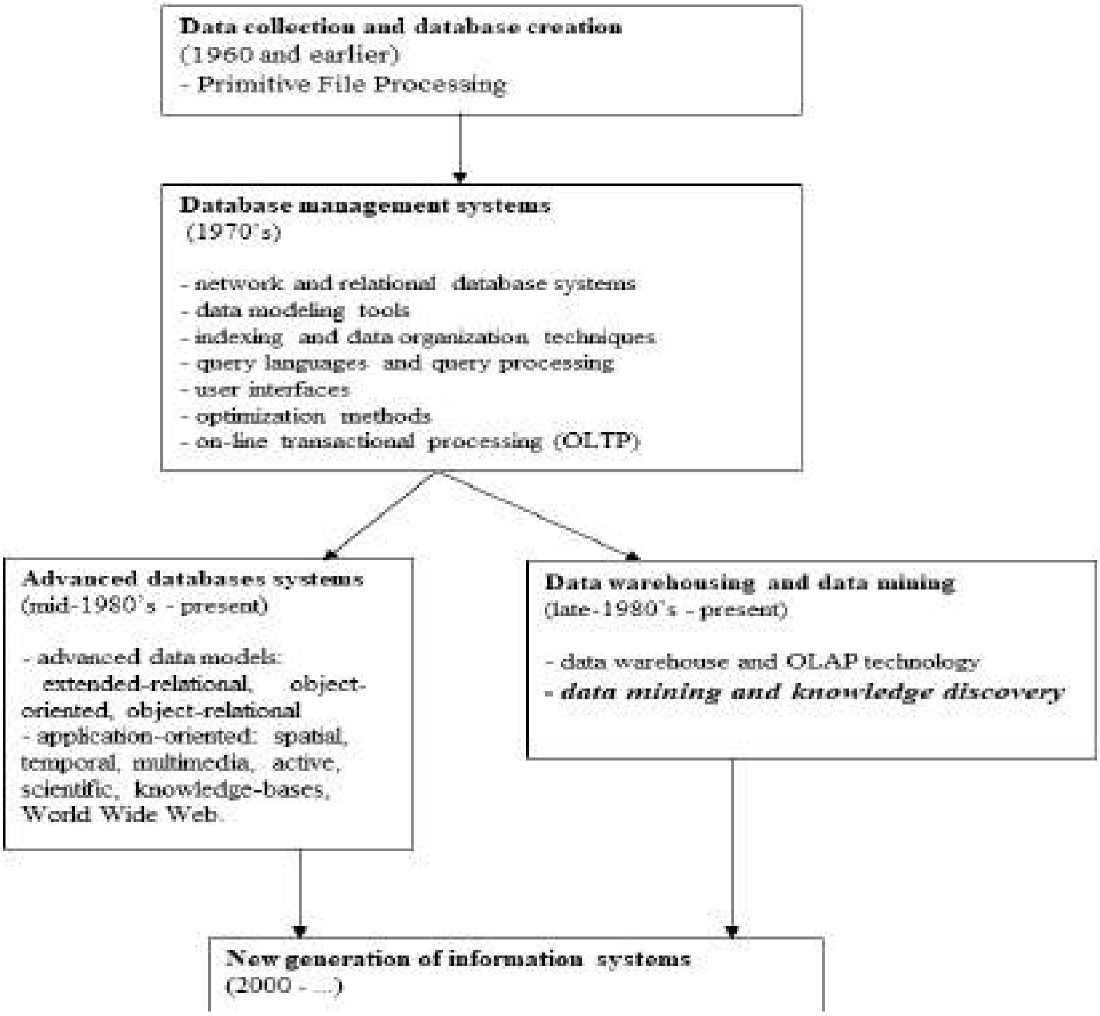
Data Mining

Evolution of Database Technology

- 1960s:Data collection, database creation, IMS and network DBMS
- 1970s: Relational data model, relational DBMS implementation
- 1980s:
    - RDBMS, advanced data models (extended-relational, OO, deductive, etc.)
    - Application-oriented DBMS (spatial, scientific, engineering, etc.)
- 1990s:

    Data mining, data warehousing, multimedia databases, and Web databases
- 2000s
    - Stream data management and mining
    - Data mining and its applications
    - Web technology (XML, data integration) and global information systems

History


Data collection and database creation
(1960 and earlier)
- Primitive File Processing
Database management systems
(1970's)
- network and relational database systems
- data modeling tools
- indexing and data organization techniques
- query languages and query processing
- user interfaces
- optimization methods
- on-line transactional processing (OLTP)
Advanced databases systems
(mid-1980's - present)
- advanced data models:
extended-relational, object-oriented, object-relational
- application-oriented: spatial, temporal, multimedia, active, scientific, knowledge-bases, World Wide Web.
Data warehousing and data mining
(late-1980's - present)
- data warehouse and OLAP technology
- data mining and knowledge discovery
New generation of information systems
(2000 - ...)

Data Mining

- Finding hidden information in a database
- Data Mining has been defined as
  "*The nontrivial extraction of implicit, previously unknown, and potentially useful information from data**".
- Data mining (knowledge discovery from data)
  - Extraction of interesting (non-trivial, implicit, previously unknown and potentially useful) patterns or knowledge from huge amount of data

Alternative names / Similar terms

- Knowledge discovery (mining) in databases (KDD)
- knowledge extraction
- data/pattern analysis
- data archeology
- data dredging
- information harvesting
- business intelligence
- Data driven discovery
- Deductive learning
- Discovery science etc.

Why Data Mining?

The Explosive Growth of Data: from terabytes to petabytes

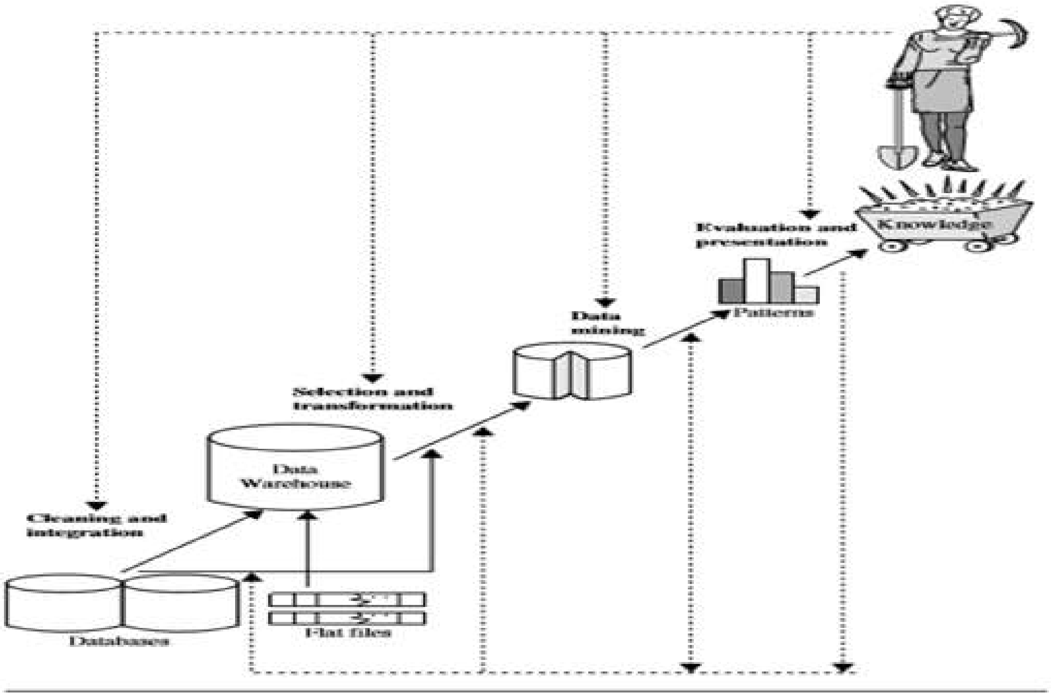
Every day the world creates a few exabytes of data
1 exabyte = 1000 petabytes
1 petabyte = 1000 terabytes
1 terabyte = 1000 gigabytes
Only **4%** of the data is used for any purpose (IBM)
If we could only do something useful with this data
… the field of DATA MINING is born

Evolution of Sciences

- Before 1600, experimental / empirical science
- 1600-1950s, theoretical science
  - Each discipline has grown a *theoretical* component. Theoretical models often motivate experiments and generalize our understanding.
- 1950s-1990s, computational science

    - Over the last 50 years, most disciplines have grown a third, *computational* branch (e.g. empirical, theoretical, and computational ecology, or physics, or linguistics.)
    - Computational Science traditionally meant simulation. It grew out of our inability to find closed-form solutions for complex mathematical models.
- 1990-now, data science
    - The flood of data from new scientific instruments and simulations
    - The ability to economically store and manage petabytes of data online
    - The Internet and computing Grid that makes all these archives universally accessible
    - Scientific info. management, acquisition, organization, query, and visualization tasks scale almost linearly with data volumes. Data mining is a major new challenge!
- Jim Gray and Alex Szalay, *The World Wide Telescope: An Archetype for Online Science*, Comm. ACM, 45(11): 50-54, Nov. 2002

Knowledge Discovery (KDD) Process



This is a view from typical database systems and data warehousing communities
Data mining plays an essential role in the knowledge discovery process

1. Data Cleaning (to remove noise and inconsistent data)
2. Data Integration (where multiple data sources may be combined.
   A popular trend in the Information Industry is to perform data cleaning and data integration as a preprocessing step, where the resulting data are stored in a data warehouse)

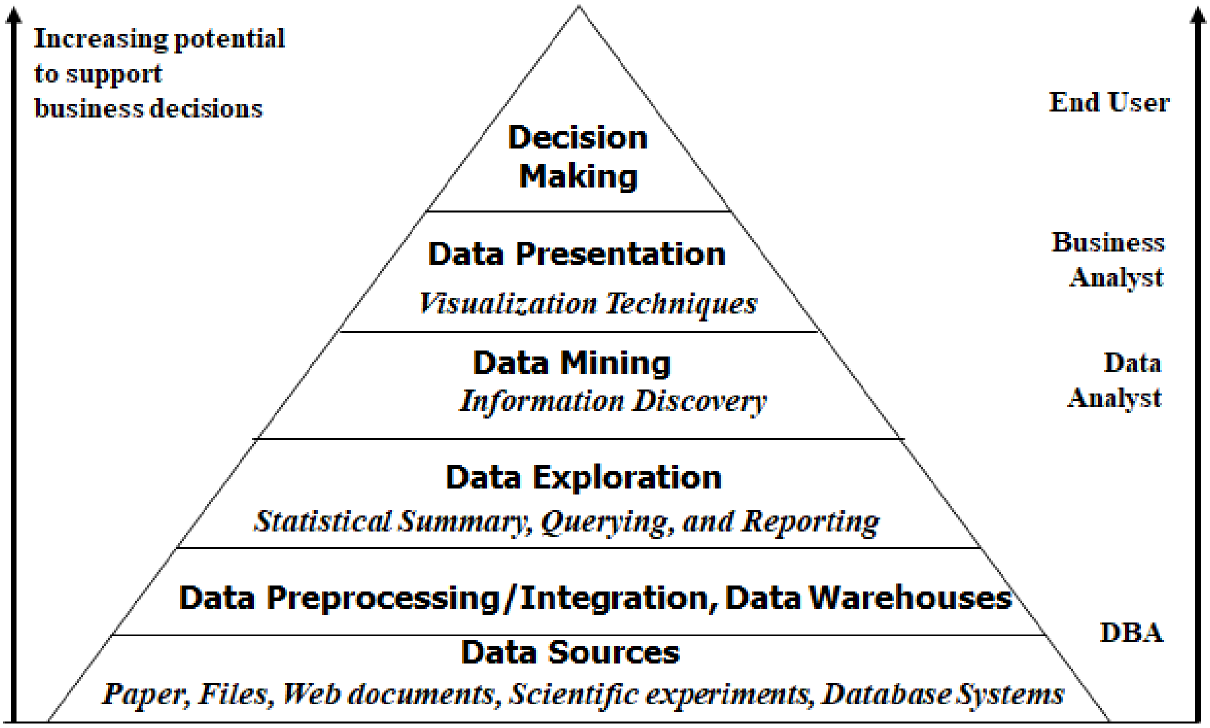
3. Data selection (where data relevant to the analysis task are retrieved from the database)
4. Data transformation (where data are transformed and consolidated into forms appropriate for mining by performing summary or aggregation operations. Sometimes data transformation and consolidation are performed before the data selection process, particularly in the case of data warehousing. Data reduction may also be performed to obtain a smaller representation of the original data without sacrificing its integrity.)
5. Data mining (an essential process where intelligent methods are applied to extract data patterns)
6. Pattern evaluation (to identify the truly interesting patterns representing knowledge based on interestingness measures)
7. Knowledge presentation (where visualization and knowledge representation techniques are used to present mined knowledge to users)

Example: A Web Mining Framework

Web mining usually involves

- Data cleaning
- Data integration from multiple sources
- Warehousing the data
- Data cube construction
- Data selection for data mining
- Data mining
- Presentation of the mining results
- Patterns and knowledge to be used or stored into knowledge-base

Data Mining in Business Intelligence



Increasing potential to support business decisions

Decision Making

Data Presentation

Visualization Techniques

Data Mining

Information Discovery

Data Exploration

Statistical Summary, Querying, and Reporting

Data Preprocessing/Integration, Data Warehouses

Data Sources

Paper, Files, Web documents, Scientific experiments, Database Systems

End User

Business Analyst

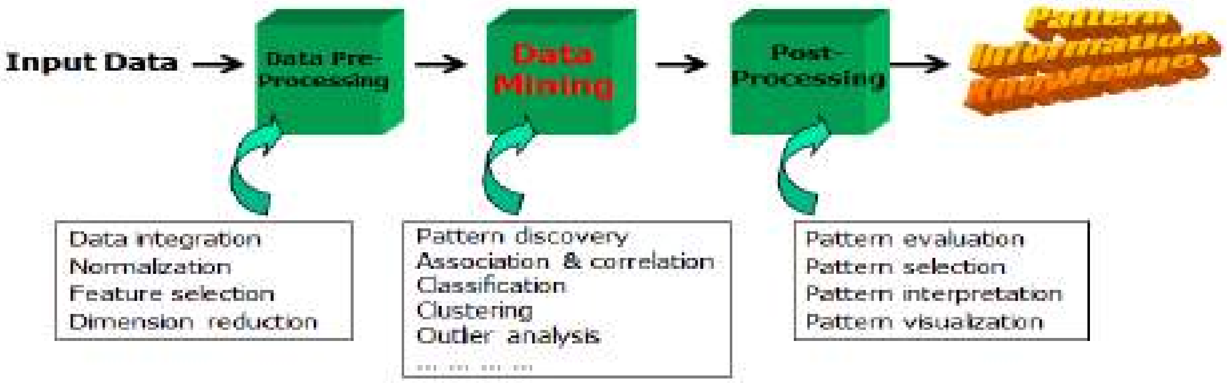
Data Analyst

DBA

Example: Mining vs. Data Exploration

- Business intelligence view
  - Warehouse, data cube, reporting but not much mining
- Business objects vs. data mining tools
- Supply chain example: tools
- Data presentation
- Exploration

KDD Process: A Typical View from ML and Statistics



Input Data → Data Pre-Processing → Data Mining → Post-Processing → Pattern Information Knowledge

Data integration
Normalization
Feature selection
Dimension reduction

Pattern discovery
Association & correlation
Classification
Clustering
Outlier analysis
... ...

Pattern evaluation
Pattern selection
Pattern interpretation
Pattern visualization

- This is a view from typical machine learning and statistics communities

Example: Medical Data Mining

- Health care & medical data mining – often adopted such a view in statistics and machine learning
- Preprocessing of the data (including feature extraction and dimension reduction)
- Classification or/and clustering processes
- Post-processing for presentation

Multi-Dimensional View of Data Mining

- Data to be mined

  Database data (extended-relational, object-oriented, heterogeneous, legacy), data warehouse, transactional data, stream, spatiotemporal, time-series, sequence, text and web, multi-media, graphs & social and information networks
- Knowledge to be mined (or: Data mining functions)
  - Characterization, discrimination, association, classification, clustering, trend/deviation, outlier analysis, etc.
  - Descriptive vs. predictive data mining
  - Multiple/integrated functions and mining at multiple levels
- Techniques utilized

  Data-intensive, data warehouse (OLAP), machine learning, statistics, pattern recognition, visualization, high-performance, etc.
- Applications adapted

  Retail, telecommunication, banking, fraud analysis, bio-data mining, stock market analysis, text mining, Web mining, etc.

Data Mining: On What Kinds of Data?

- Database-oriented data sets and applications
  - Relational database, data warehouse, transactional database
- Advanced data sets and advanced applications
  - Data streams and sensor data
  - Time-series data, temporal data, sequence data (incl. bio-sequences)
  - Structure data, graphs, social networks and multi-linked data
  - Object-relational databases
  - Heterogeneous databases and legacy databases
  - Spatial data and spatiotemporal data
  - Multimedia database
  - Text databases
  - The World-Wide Web

Data Mining Function: (1) Generalization

- Information integration and data warehouse construction
  - Data cleaning, transformation, integration, and multidimensional data model
- Data cube technology
  - Scalable methods for computing (i.e., materializing) multidimensional aggregates
  - OLAP (online analytical processing)
- Multidimensional concept description: Characterization and discrimination
  - Generalize, summarize, and contrast data characteristics, e.g., dry vs. wet region

Data Mining Function: (2) Association and Correlation Analysis

- Frequent patterns (or frequent itemsets)
  - What items are frequently purchased together in your Walmart?
- Association, correlation vs. causality
  - A typical association rule

    Diaper → Beer [0.5%, 75%] (support, confidence)
  - Are strongly associated items also strongly correlated?
- How to mine such patterns and rules efficiently in large datasets?
- How to use such patterns for classification, clustering, and other applications?

Data Mining Function: (3) Classification

- Classification and label prediction
  - Construct models (functions) based on some training examples
  - Describe and distinguish classes or concepts for future prediction
    - E.g., classify countries based on (climate), or classify cars based on (gas mileage)
  - Predict some unknown class labels
- Typical methods
  - Decision trees, naïve Bayesian classification, support vector machines, neural networks, rule-based classification, pattern-based classification, logistic regression, …
- Typical applications:
  - Credit card fraud detection, direct marketing, classifying stars, diseases, web-pages, …

Data Mining Function: (4) Cluster Analysis

- Unsupervised learning (i.e., Class label is unknown)
- Group data to form new categories (i.e., clusters), e.g., cluster houses to find distribution patterns
- Principle: Maximizing intra-class similarity & minimizing interclass similarity
- Many methods and applications

Data Mining Function: (5) Outlier Analysis

- Outlier: A data object that does not comply with the general behavior of the data
- Noise or exception? — One person's garbage could be another person's treasure
- Methods: by product of clustering or regression analysis, …
- Useful in fraud detection, rare events analysis

Time and Ordering: Sequential Pattern, Trend and Evolution Analysis

- Sequence, trend and evolution analysis
  - Trend, time-series, and deviation analysis: e.g., regression and value prediction
  - Sequential pattern mining

    e.g., first buy digital camera, then buy large SD memory cards
  - Periodicity analysis
  - Motifs and biological sequence analysis

    Approximate and consecutive motifs
  - Similarity-based analysis
- Mining data streams
  - Ordered, time-varying, potentially infinite, data streams

Structure and Network Analysis

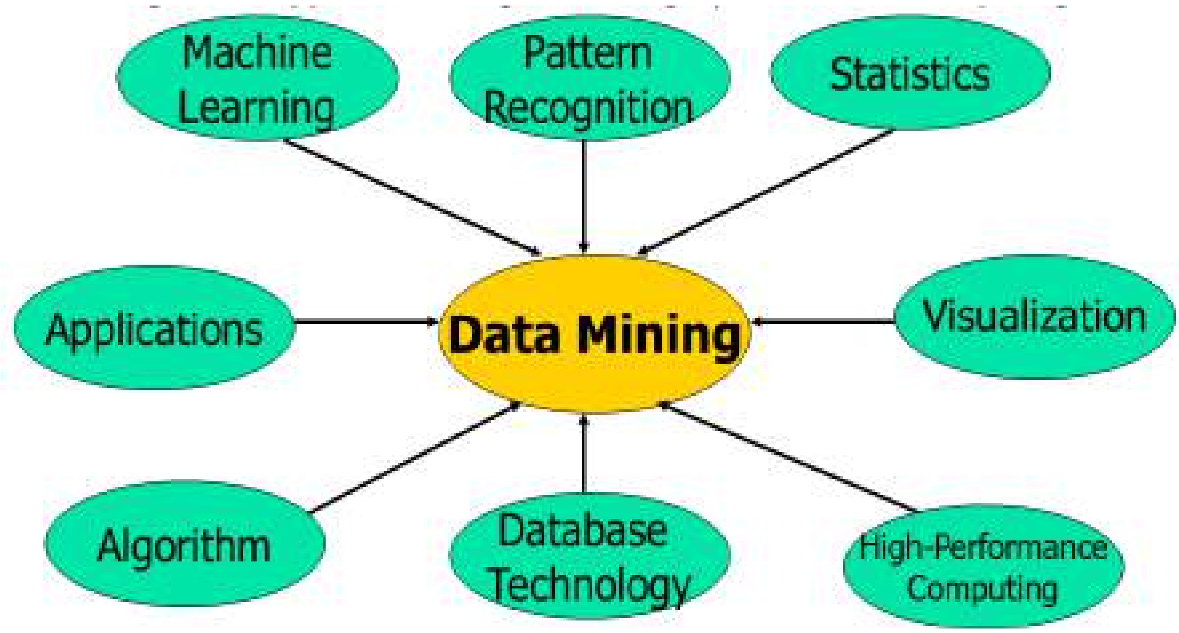
- Graph mining
  - Finding frequent subgraphs (e.g., chemical compounds), trees (XML), substructures (web fragments)
- Information network analysis
  - Social networks: actors (objects, nodes) and relationships (edges)

    e.g., author networks in CS, terrorist networks
  - Multiple heterogeneous networks

    A person could be multiple information networks: friends, family, classmates, …
  - Links carry a lot of semantic information: Link mining
- Web mining
  - Web is a big information network: from PageRank to Google
  - Analysis of Web information networks
- Web community discovery, opinion mining, usage mining, …

Classification of Data Mining Systems

Data mining system is an interdisciplinary field (has incorporated many techniques from other domains) such as database and data warehouse, statistics, machine learning, visualization, pattern recognition, applications, algorithm, high performance computing



Classification of Data Mining System

Data mining systems can be categorized according to various criteria, as follows:

- **Classification according to the kinds of database mined**

  Database data (extended-relational, object-oriented, heterogeneous, legacy), data warehouse, transactional data, stream, spatiotemporal, time-series, sequence, text and web, multi-media, graphs & social and information networks

- **Classification according to the kinds of Knowledge mined (or: Data mining functions)**

  - Characterization, discrimination, association, classification, clustering, trend/deviation, outlier analysis, etc.
  - Descriptive vs. predictive data mining

    Descriptive mining tasks describe the characteristics of the **data** in a target **data** set. On the other hand, **predictive mining** tasks carry out the induction over the current **and** past **data** so that predictions can be made
  - Multiple/integrated functions and mining at multiple levels

- **Classification according to the kinds of Techniques utilized**

  Data-intensive, data warehouse (OLAP), machine learning, statistics, pattern recognition, visualization, high-performance, etc.
- **Classification according to the Applications adapted**

  Retail, telecommunication, banking, fraud analysis, bio-data mining, stock market analysis, text mining, Web mining, etc.

Applications of Data Mining

- Web page analysis: from web page classification, clustering to PageRank & HITS algorithms
- Collaborative analysis & recommender systems
- Basket data analysis to targeted marketing (Market **Basket Analysis** is a modelling technique based upon the theory that if you buy a certain group of items, you are more (or less) likely to buy another group of items.)
- Biological and medical data analysis: classification, cluster analysis (microarray data analysis), biological sequence analysis, biological network analysis
- From major dedicated data mining systems/tools (e.g., SAS, MS SQL-Server Analysis Manager, Oracle Data Mining Tools) to invisible data mining

Major Issues in Data Mining

- Mining Methodology
  - Mining various and new kinds of knowledge
  - Mining knowledge in multi-dimensional space
  - Data mining: An interdisciplinary effort
  - Boosting the power of discovery in a networked environment
  - Handling noise, uncertainty, and incompleteness of data
  - Pattern evaluation and pattern- or constraint-guided mining
- User Interaction
  - Interactive mining
  - Incorporation of background knowledge
  - Presentation and visualization of data mining results
- Efficiency and Scalability
  - Efficiency and scalability of data mining algorithms
  - Parallel, distributed, stream, and incremental mining methods
- Diversity of data types
  - Handling complex types of data
  - Mining dynamic, networked, and global data repositories
- Data mining and society
  - Social impacts of data mining
  - Privacy-preserving data mining
  - Invisible data mining

Data Preprocessing

Data Quality: Why Preprocess the Data?

- Measures for data quality: A multidimensional view
    - Accuracy: correct or wrong, accurate or not
    - Completeness: not recorded, unavailable, …
    - Consistency: some modified but some not, dangling, …
    - Timeliness: timely update?
    - Believability: how trustable the data are correct?
    - Interpretability: how easily the data can be understood?

Major Tasks in Data Preprocessing

- **Data cleaning**
    - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- **Data integration**
    - Integration of multiple databases, data cubes, or files
- **Data reduction**
    - Dimensionality reduction
    - Numerosity reduction
    - Data compression
- **Data transformation and data discretization**
    - Normalization
- Concept hierarchy generation

Data Cleaning

Data in the Real World Is Dirty: Lots of potentially incorrect data, e.g., instrument faulty, human or computer error, transmission error

- Incomplete/missing: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data

    e.g., *Occupation*="" (missing data)
- noisy: containing noise, errors, or outliers

    e.g., *Salary*="−10" (an error)
- inconsistent: containing discrepancies in codes or names,

    e.g.,
    - *Age*="42", *Birthday*="03/07/2010"
    - Was rating "1, 2, 3", now rating "A, B, C"
    - discrepancy between duplicate records
- Intentional (e.g., *disguised missing* data)

    Jan. 1 as everyone's birthday?

Incomplete (Missing) Data

- Data is not always available
  - E.g., many tuples have no recorded value for several attributes, such as customer income in sales data
- Missing data may be due to
  - equipment malfunction
  - inconsistent with other recorded data and thus deleted
  - data not entered due to misunderstanding
  - certain data may not be considered important at the time of entry
  - not register history or changes of the data
- Missing data may need to be inferred

How to Handle Missing Data?

- Ignore the tuple: usually done when class label is missing (when doing classification)—not effective when the % of missing values per attribute varies considerably
- Fill in the missing value manually: tedious + infeasible?
- Fill in it automatically with
  - a global constant : e.g., "unknown", a new class?!
  - the attribute mean
  - the attribute mean for all samples belonging to the same class: smarter
- the most probable value: inference-based such as Bayesian formula or decision tree

Noisy Data

- Noise: random error or variance in a measured variable
- Incorrect attribute values may be due to
  - faulty data collection instruments
  - data entry problems
  - data transmission problems
  - technology limitation
  - inconsistency in naming convention
- Other data problems which require data cleaning
  - duplicate records
  - incomplete data
  - Inconsistent / unpredictable data

How to Handle Noisy Data?

- Binning
    - first sort data and partition(divided) into several buckets/equal frequency (also called bins)
    - then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
- Regression
    - smooth by fitting the data into regression functions
        - y = b + mx
- Clustering
    - Groups(also called Clusters) having similar values are formed
    - detect and remove outliers
- Combined computer and human inspection
    - detect suspicious values and check by human (e.g., deal with possible outliers)

Data Cleaning as a Process

Various steps involved in datamining process

Data discrepancy detection

there may exist noise in data due to the following

- Manual error, system error, field overloading, poorly designed structure etc
- The discrepancy can be detected by metadata (data about data)helps in finding the domain & datatype, range, dependency, distribution

- Data should be analyzed using the following rules.
    - Unique rule (attribute value should be different)
    - Consecutive rule (should be no missing values b/w the lowest & highest value for the attribute, all existing values must be unique)
    - Null rule (use special characters/strings to indicate null condition)
    - Use commercial tools
        - Data scrubbing tools: use simple domain knowledge (e.g., postal code, spell-check) to detect errors and make corrections
        - Data auditing tools: by analyzing data to discover rules and relationship to detect violators (e.g., correlation and clustering to find outliers)

Data transformation (Data migration and integration)

- Data migration tools: allow transformations to be specified
- ETL (Extraction/Transformation/Loading) tools: allow users to specify transformations through a graphical user interface(ex.to replace ‘post’ by ‘designation’)
- Integration of the two processes
- Iterative and interactive (e.g., Potter’s Wheels)

Data Integration

Combines data from multiple sources into a coherent store

- Schema integration: e.g., A.cust-id ≡ B.cust-#

  Integrate metadata from different sources
- Entity identification problem:
  - Identify real world entities from multiple data sources
  - Ex : *customer_id* in one database, *cust-number* in another DB
- Detecting and resolving data value conflicts
  - For the same real world entity, attribute values from different sources are different
  - Possible reasons: different representations, different scales, e.g., metric vs. British units

Handling Redundancy in Data Integration

- Redundant data occur often when integration of multiple databases
  - *Object identification*: The same attribute (or object) may have different names in different databases
  - *Derivable data:* One attribute may be a "derived" attribute in another table, e.g., *age* can be derived from *DOB*
- Redundant attributes may be able to be detected by *correlation analysis* and *covariance analysis*
- Careful integration of the data from multiple sources may help reduce/avoid redundancies and inconsistencies and improve mining speed and quality

Correlation Analysis (Nominal Data)

- **X^2 (chi-square) test**

$$\chi^2 = \sum \frac{(Observed - Expected)^2}{Expected}$$

- The larger the X^2 value, the more likely the variables are related
- The cells that contribute the most to the X^2 value are those whose actual count is very different from the expected count
- Correlation does not imply causality
  - # of hospitals and # of car-theft in a city are correlated
  - Both are causally linked to the third variable: population

Chi-Square Calculation: An Example

	Play chess	**Not play chess**	**Sum (row)**
Like science fiction	250(90)	200(360)	450
Not like science fiction	50(210)	1000(840)	1050
Sum(col.)	300	1200	1500

- X^2 (chi-square) calculation (numbers in parenthesis are expected counts calculated based on the data distribution in the two categories)

$$\chi^2 = \frac{(250-90)^2}{90} + \frac{(50-210)^2}{210} + \frac{(200-360)^2}{360} + \frac{(1000-840)^2}{840} = 507.93$$

- It shows that like_science_fiction and play_chess are correlated in the group

Correlation Analysis (Numeric Data)

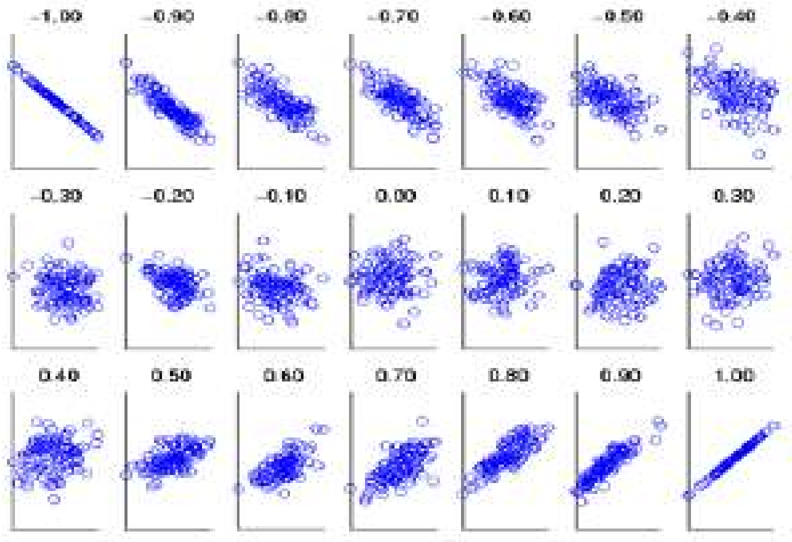
- Correlation coefficient (also called Pearson's product moment coefficient)

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

  where n is the number of tuples, and are the respective means of A and B, σ_A and σ_B are the respective standard deviation of A and B, and $\Sigma(a_i b_i)$ is the sum of the AB cross-product.

- If $r_{A,B} > 0$, A and B are positively correlated (A's values increase as B's). The higher, the stronger correlation.
- $r_{A,B} = 0$: independent;
- $r_{AB} < 0$: negatively correlated

Visually Evaluating Correlation



Scatter plots showing the similarity from −1 to 1.

Correlation (viewed as linear relationship)

- Correlation measures the linear relationship between objects
- To compute correlation, we standardize data objects, A and B, and then take their dot product

$$a'_k = (a_k - mean(A)) / std(A)$$

$$b'_k = (b_k - mean(B)) / std(B)$$

$$correlation(A, B) = A' \bullet B'$$

Covariance (Numeric Data)

- Covariance is similar to correlation

$$Cov(A,B) = E((A-\bar{A})(B-\bar{B})) = \frac{\sum_{i=1}^{n}(a_i-\bar{A})(b_i-\bar{B})}{n}$$

Correlation coefficient: $r_{A,B} = \frac{Cov(A,B)}{\sigma_A \sigma_B}$

- where n is the number of tuples, and are the respective mean or **expected values** of A and B, σ_A and σ_B are the respective standard deviation of A and B.
- **Positive covariance**: If $Cov_{A,B} > 0$, then A and B both tend to be larger than their expected values.
- **Negative covariance**: If $Cov_{A,B} < 0$ then if A is larger than its expected value, B is likely to be smaller than its expected value.
- **Independence**: $Cov_{A,B} = 0$ but the converse is not true:

  Some pairs of random variables may have a covariance of 0 but are not independent. Only under some additional assumptions (e.g., the data follow multivariate normal distributions) does a covariance of 0 imply independence

Co-Variance: An Example

$$Cov(A,B) = E((A-\bar{A})(B-\bar{B})) = \frac{\sum_{i=1}^{n}(a_i-\bar{A})(b_i-\bar{B})}{n}$$

It can be simplified in computation as

$$Cov(A,B) = E(A \cdot B) - \bar{A}\bar{B}$$

- Suppose two stocks A and B have the following values in one week:

  (2, 5), (3, 8), (5, 10), (4, 11), (6, 14).
- Question: If the stocks are affected by the same industry trends, will their prices rise or fall together?

  E(A) = (2 + 3 + 5 + 4 + 6)/ 5 = 20/5 = 4

  E(B) = (5 + 8 + 10 + 11 + 14) /5 = 48/5 = 9.6

  Cov(A,B) = (2×5+3×8+5×10+4×11+6×14)/5 − 4 × 9.6 = 4
- Thus, A and B rise together since Cov(A, B) > 0.

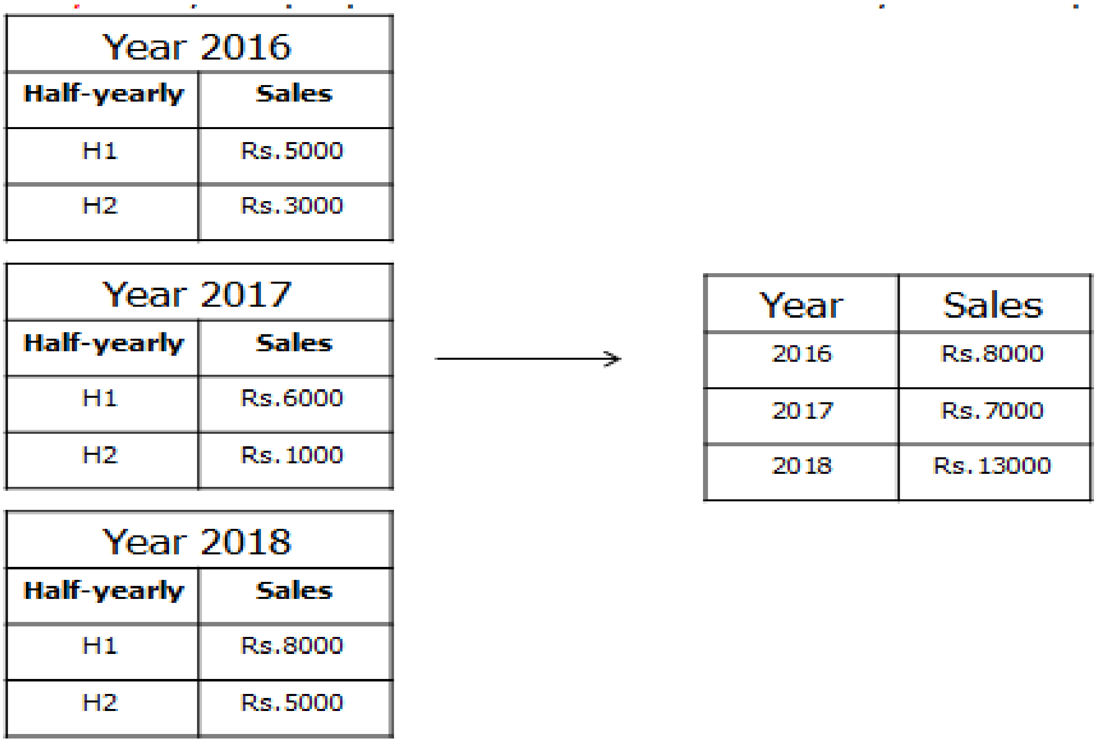
Data Reduction

- **Data reduction**: It is processing technique which helps in obtaining reduced representation of the data set (ie set having much smaller volume of data) from the available data set but yet produces the same (or almost the same) analytical results
- Why data reduction? — A database/data warehouse may store terabytes of data. Complex data analysis may take a very long time to run on the complete data set.
- Data reduction strategies
  - Data cube aggregation
  - Attribute subset selection
  - Dimensionality reduction, e.g., remove unimportant attributes
    - Wavelet transforms
    - Principal Components Analysis (PCA)
  - Numerosity reduction (some simply call it: Data Reduction)
    - Regression and Log-Linear Models
    - Histograms, clustering, sampling
  - Data Compression

Data Cube Aggregation

Where aggregation operations are applied to the data in the construction of a data cube. It is a process in which information is gathered and expressed in a summary form, for purpose such as statistical analysis.

Example



Year 2016

Half-yearly	Sales
H1	Rs.5000
H2	Rs.3000

Year 2017

Half-yearly	Sales
H1	Rs.6000
H2	Rs.1000

Year 2018

Half-yearly	Sales
H1	Rs.8000
H2	Rs.5000

Year	Sales
2016	Rs.8000
2017	Rs.7000
2018	Rs.13000

Attribute Subset Selection

- Irrelevant, weakly relevant, or redundant attributes or dimensions may be detected and removed.
- Redundant attributes
  - Duplicate much or all of the information contained in one or more other attributes
  - E.g., purchase price of a product and the amount of sales tax paid
- Irrelevant attributes
  - Contain no information that is useful for the data mining task at hand
  - E.g., students' ID is often irrelevant to the task of predicting students' GPA

Heuristic Search in Attribute Selection

- For n attributes, there exists 2^n possible subsets.
- Typical heuristic attribute selection methods:
  - Stepwise forward selection
  - Stepwise backward elimination
  - Combination of forward selection and backward elimination
  - Decision tree induction

Stepwise forward selection

- In the beginning of the procedure, the reduced set is empty.
- Then determines the **best attribute** among all the available original attributes and **adds** it to the reduced set.

Forward Selection
Initial Attribute Set: { A1,A2,A3,A4,A5,A6,A7,A8}
Initial Reduced Set : { }
=> {A2}
=> {A2, A5}
=> {A2,A5,A6}
=> Reduced Attribute Set:
=>{A2,A5,A6,A8}

Stepwise backward elimination

- Initially, the procedure starts with the full set of attributes in the reduced set.
- Then, with each successive iteration the **worst attribute is removed** from the attribute set.

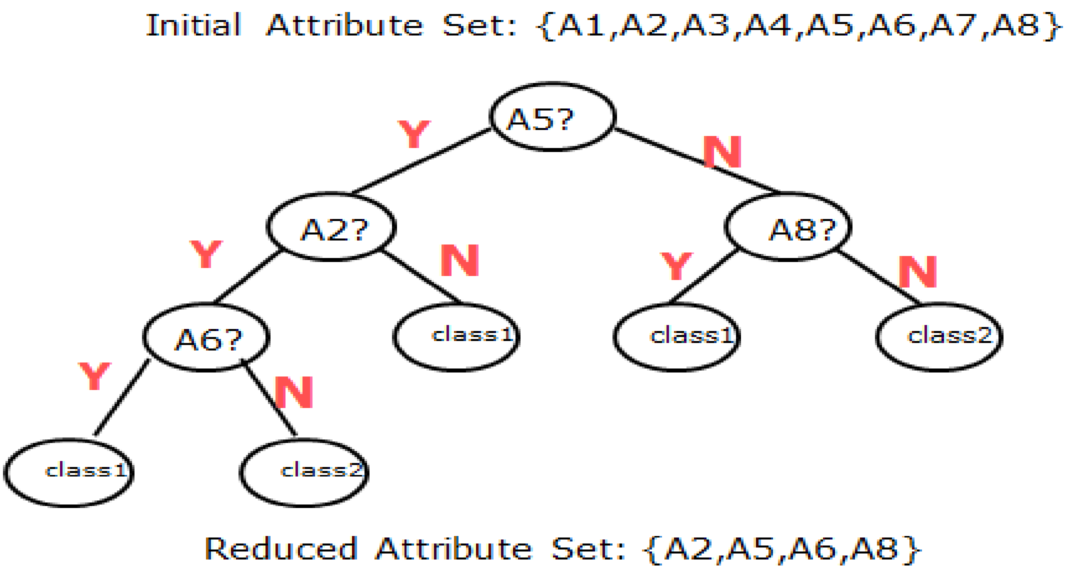
Backward Elimination
Initial Attribute Set: { A1,A2,A3,A4,A5,A6,A7,A8}
=> {A1,A2,A3,A4,A5,A6,A8}
=> {A1,A2,A3,A5,A6,A8}
=> {A1,A2,A5,A6,A8}
=> Reduced Attribute Set:
=>{A2,A5,A6,A8}

Combination of forward selection and backward elimination

This technique with each iteration identifies the best attribute from the original attributes and at the same time removes the worst attributes from the remaining attributes.

Decision Tree Induction

- This technique constructs a tree-like structure on the basis of the available data
- The tree consists of an internal (nonleaf) node, which denotes a test on an attribute, a branch, which represents the result of the test, and an external (leaf) node which denotes predicated class.
- At each node, the algorithm chooses the best attribute such that the data are divided into individual classes.


Initial Attribute Set: {A1,A2,A3,A4,A5,A6,A7,A8}
A5?
Y
N
A2?
A8?
Y
N
Y
N
A6?
class1
class1
class2
Y
N
class1
class2
Reduced Attribute Set: {A2,A5,A6,A8}

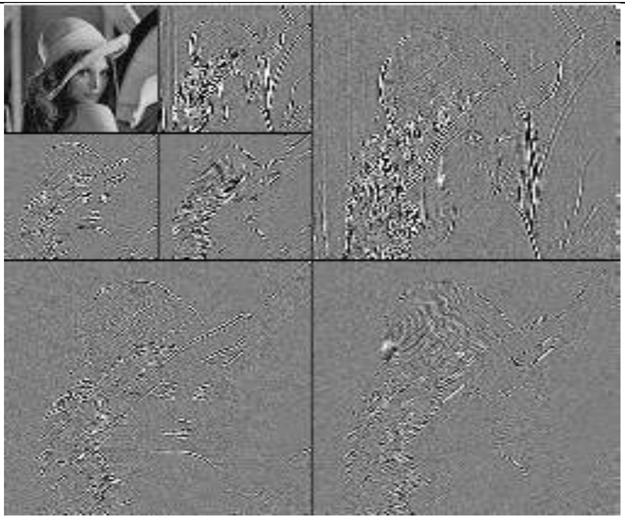
Dimensionality Reduction

- **Curse of dimensionality**
  - When dimensionality increases, data becomes increasingly sparse
  - Density and distance between points, which is critical to clustering, outlier analysis, becomes less meaningful
  - The possible combinations of subspaces will grow exponentially

- **Dimensionality reduction(Uses)**
  - Avoid the curse of dimensionality
  - Help eliminate irrelevant features and reduce noise
  - Reduce time and space required in data mining
  - Allow easier visualization

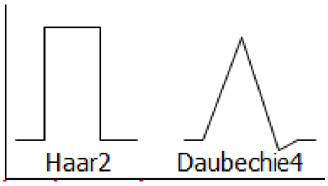
- **Dimensionality reduction techniques**
  - Wavelet transforms
  - Principal Component Analysis (PCA)
- Dimensionality reduction represents the original data in the compressed or reduced form by applying data encoding or transformations on it.
- If the original data can be reconstructed from the compressed data without losing any information, the data reduction is said to be **lossless.**
- If one can reconstruct only an approximation of the original data, the data reduction is said to be **lossy.**
- The two most effective and popular methods of lossy dimensionality reduction are
  - Wavelet transforms
  - Principal Component Analysis(PCA)

What Is Wavelet Transform?

➢ Decomposes a signal into different frequency subbands Applicable to n-dimensional signals ➢ Data are transformed to preserve relative distance between objects at different levels of resolution ➢ Allow natural clusters to become more distinguishable ➢ Used for image compression	

Wavelet Transformation

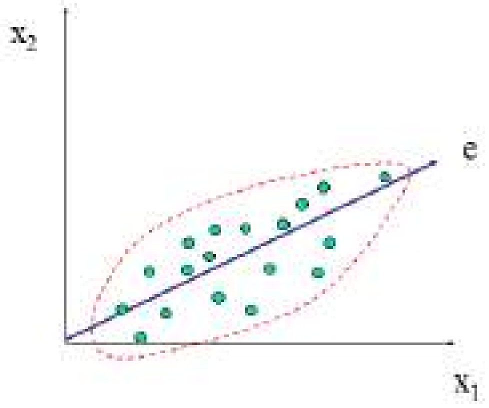
- This lossy dimension reduction method works by using its variant called **Discrete Wavelet Transform** (DWT)



- DWT is a linear signal processing technique, multi-resolution analysis
- Compressed approximation: store only a small fraction of the strongest of the wavelet coefficients
- Similar to **Discrete Fourier Transform** (DFT) , it is a signal processing technique, but better lossy compression, localized in space
- Lossy compression by wavelets gives better result than JPEG compression.
- Some of the popular wavelet transforms are Haar-2, Daubechines-4 and Daubechines-6 transforms.

Principal Component Analysis (PCA)

- Find a projection that captures the largest amount of variation in data
- The original data are projected onto a much smaller space, resulting in dimensionality reduction. We find the eigenvectors of the covariance matrix, and these eigenvectors define the new space



Principal Component Analysis (Steps)

- This method searches for k,
- n-dimensional orthogonal vectors that can be best used to represent the data
- Where k <= n, here n refers to total attributes or dimensions of the data which need to be reduced
- find $k \leq n$ orthogonal vectors (*principal components*) that can be best used to represent data

- Normalize input data: Each attribute falls within the same range
- Compute *k* orthogonal (unit) vectors, i.e., *principal components*
- Each input data (vector) is a linear combination of the *k* principal component vectors
- The principal components are sorted in order of decreasing “significance” or strength
- Since the components are sorted, the size of the data can be reduced by eliminating the *weak components*, i.e., those with low variance (i.e., using the strongest principal components, it is possible to reconstruct a good approximation of the original data)

➢ Works for numeric data only

Numerosity Reduction

It reduces data volume by choosing alternative *smaller forms* of data representation. Such representation can be achieved by two methods

1. **Parametric methods**
   - ➢ Here, only parameters of data and outliers are stored instead of the actual data. (Assume the data fits some model, estimate model parameters, store only the parameters, and discard the data (except possible outliers))
   - ➢ Ex.: Regression and Log-linear models
2. **Non-parametric** methods

Used to store data in reduced forms such as Histograms, Clustering, Sampling, …

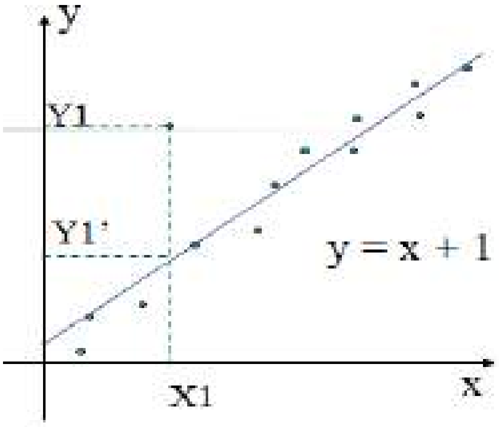
- ➢ Do not assume models

Regression Models

- ➢ Linear regression (*Y = m X + b)*
  - Data modeled to fit a straight line, it uses the formula of a straight line (Y = m X + b) and determines the appropriate values for m and b (called regression coefficients) to predict the value of y (also called response variable) based on a given value of x (also called predictor variable)
  - Often uses the least-square method to fit the line
  - Two regression coefficients, *m* and *b,* specify the line and are to be estimated by using the data at hand
  - Using the least squares criterion to the known values of $Y_1, Y_2, \ldots, X_1, X_2, \ldots.$
- ➢ Multiple regression ($Y = b_0 + b_1 X_1 + b_2 X_{2)}$
  - Allows a response variable Y to be modeled as a linear function of multidimensional feature vector
  - Many nonlinear functions can be transformed into the above

Regression Analysis

- Regression analysis: A collective name for techniques for the modeling and analysis of numerical data consisting of values of a ***dependent variable*** (also called ***response variable*** or *measurement*) and of one or more *independent variables* (aka. ***explanatory variables*** or ***predictors***)
- The parameters are estimated so as to give a "**best fit**" of the data
- Most commonly the best fit is evaluated by using the ***least squares method***, but other criteria have also been used

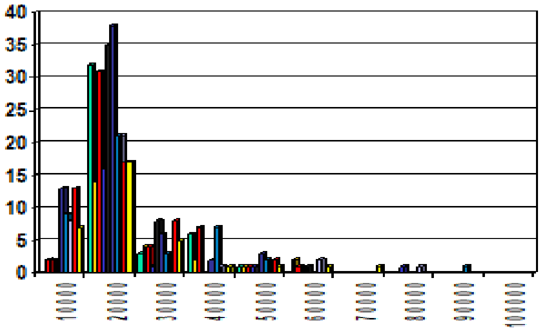


- Used for prediction (including forecasting of time-series data), inference, hypothesis testing, and modeling of causal relationships

Log-Linear Models

- Approximate discrete multidimensional probability distributions
- Estimate the probability of each point (tuple) in a multi-dimensional space for a set of discretized attributes, based on a smaller subset of dimensional combinations
- Useful for dimensionality reduction and data smoothing

Histogram

➢ One of the popular forms of data reduction and uses binning for approximating the distribution of data ➢ Divide data into buckets and store average (sum) for each bucket ➢ Partitioning rules: o Equal-width: equal bucket range o Equal-frequency (or equal-depth)	

- Partitioning rules:
  - Equal-width: equal bucket range
    In this, the width of each bucket range is uniform
  - Equal-frequency (or equal-depth)
- buckets are created that the frequency of each bucket is roughly constant. Ie each bucket is roughly holding the same number of contiguous data samples.

Clustering

- Partition data set into clusters based on similarity, and store cluster representation (e.g., centroid and diameter) only
- The quality of clusters is determined by two factors, namely **cluster diameter and centroid distance**
- **Cluster diameter** is defined as the maximum distance b/w any two objects.
- **Centroid distance** is the average distance of each cluster object from the cluster centroid.
- Can be very effective if data is clustered but not if data is "smeared"
- Can have hierarchical clustering and be stored in multi-dimensional index tree structures
- There are many choices of clustering definitions and clustering algorithms

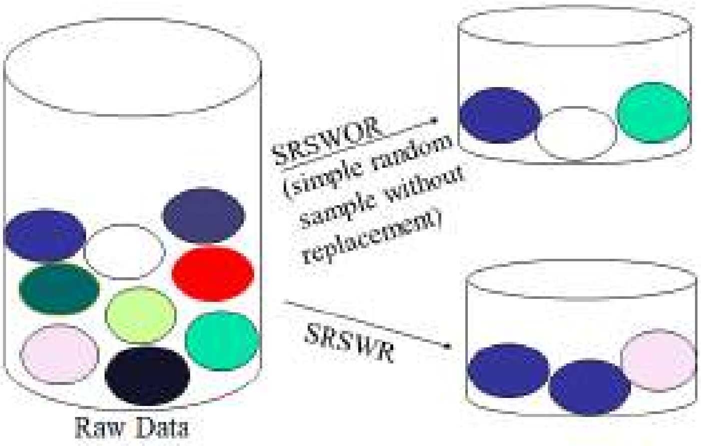
Sampling

- It is data reduction technique which helps in representing a large data set by a much smaller random sample (or subset) of the data.
- Sampling: obtaining a small sample *s* to represent the whole data set *N*
- Allow a mining algorithm to run in complexity that is potentially sub-linear to the size of the data
- Key principle: Choose a representative subset of the data
  - Simple random sampling may have very poor performance in the presence of skew
  - Develop adaptive sampling methods, e.g., stratified sampling:
- Note: Sampling may not reduce database I/Os (page at a time)

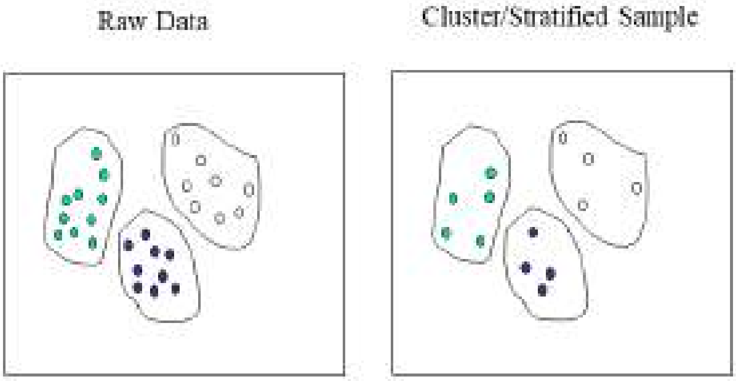
Types of Sampling

- **Simple random Sample without replacement(SRSWOR) of size s**
  Once an object is selected, it is removed from the population
- **Simple random Sample with replacement(SRSWR) of size s**
  A selected object is not removed from the population
- **Stratified sampling(Cluster sample):**
  - Partition the data set, and draw samples from each partition (proportionally, i.e., approximately the same percentage of the data)
  - Used in conjunction with skewed data

Sampling: With or without Replacement



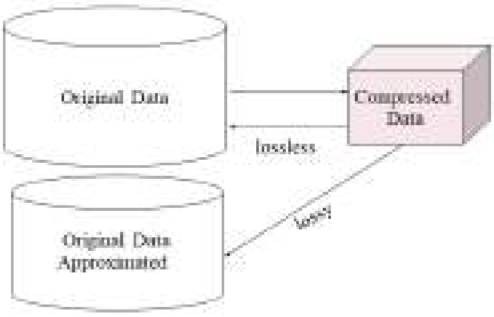
Sampling: Cluster or Stratified Sampling



Data Reduction 3: Data Compression

- String compression
  - There are extensive theories and well-tuned algorithms
  - Typically lossless, but only limited manipulation is possible without expansion
- Audio/video compression
  - Typically lossy compression, with progressive refinement
  - Sometimes small fragments of signal can be reconstructed without reconstructing the whole
- Time sequence is not audio

  Typically short and vary slowly with time
- Dimensionality and numerosity reduction may also be considered as forms of data compression



If the original data can be reconstructed from the compressed data without losing any information, the data reduction is said to be **lossless.**

If one can reconstruct only an approximation of the original data, the data reduction is said to be **lossy.**

Data Transformation

- It is a data processing , here the data are transformed or consolidated into alternate forms , so that the resulting mining process may be more efficient and the patterns found may be easier to understand.
- Strategies (processes/Methods) for data transformation include the following
  - Smoothing: It helps in removing noise from the data. The various techniques used for this purpose are binning, regression and clustering
  - Attribute (or feature construction)
    New attributes are constructed and added from the given set of attributes
    Ex : one may add the attribute *age* based on the attribute *DOB*
  - Aggregation: Summery or aggregation operations are applied to the data.
    Ex. The daily sales data may be aggregated for computing monthly and annual total amounts.
  - Normalization: Where the attributes data are scaled to fall within a smaller range, such as -1.0 to 1.0, or 0.0 to 1.0
    - Min-max normalization
    - Z-score normalization
    - Normalization by decimal scaling
  - Discretization: Where the raw values of a numeric attribute(eg. Age) are replaced by interval labels (e.g. 0 – 10, 11- 20, etc) or conceptual labels (e.g. youth, adult, senior)
  - Concept hierarchy generation for nominal data:
    where attributes such as street can be generalized to higher-level concepts, like *city* or *country*

Normalization

- **Min-max normalization**: to [new_min_A, new_max_A]
  This method linearly transforms the original data. Ex. Consider an attribute A (ex. income) having minimum value min_A (Ex.Rs.12,000), maximum value max_A

(Ex.Rs.98,000) and its original value as v (Ex.Rs.73,600). The min-max normalization maps or transforms the value of the attribute v into v’ in the range [new_min$_A$, new-max$_A$] (ex. [0.0, 1.0]) by the following computation:

$$v' = \frac{v - min_A}{max_A - min_A}(new\_max_A - new\_min_A) + new\_min_A$$

$$\frac{73{,}600 - 12{,}000}{98{,}000 - 12{,}000}(1.0 - 0) + 0 = 0.716$$

- **Z-score normalization** (μ: mean, σ: standard deviation):
  The normalization is based on the mean and standard deviation of attribute A.
  A value of v, of attribute A is normalized by v’ by the following computation

$$v' = \frac{v - \mu_A}{\sigma_A}$$

Ex. Let μ = 54,000, σ = 16,000. Then $\frac{73{,}600 - 54{,}000}{16{,}000} = 1.225$

- **Normalization by decimal scaling:**

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

Where *j* is the smallest integer such that Max(|ν’|) < 1
Suppose that the recorded values of A range from -986 to 917.The maximum absolute value of A is 986.
To normalize by decimal scaling, divide each value by 1000 (ie. j=3)
so that -986 normalizes to -0.986 and 917 normalizes to 0.917

Discretization

- Where the raw values of a numeric attribute(eg. Age) are replaced by interval labels (e.g. 0 – 10, 11- 20, etc) or conceptual labels (e.g. youth, adult, senior)
- Discretization techniques can be categorized based on how the discretization is performed, such as whether it uses class information or which direction it proceeds (ie top-down vs bottom-up).
- If the discretization process uses class information, then we say it is ***supervised discretization.*** Otherwise, it is ***unsupervised discretization***
  - Interval labels can be used to replace actual data values
  - Reduce data size by discretization
  - Supervised vs. unsupervised
  - Split (top-down) vs. merge (bottom-up)
  - Discretization can be performed recursively on an attribute
  - Prepare for further analysis, e.g., classification

- Three types of attributes
  - Nominal—values from an unordered set, e.g., color, profession
  - Ordinal—values from an ordered set, e.g., military or academic rank
  - Numeric—real numbers, e.g., integer or real numbers

Data Discretization Methods

- Typical methods: All the methods can be applied recursively
  - Binning

    Top-down split, unsupervised
  - Histogram analysis

    Top-down split, unsupervised
  - Clustering analysis (unsupervised, top-down split or bottom-up merge)
  - Decision-tree analysis (supervised, top-down split)
  - Correlation (e.g., χ^2) analysis (unsupervised, bottom-up merge)

Discretization by Binning

- Equal-width (distance) partitioning
  - Divides the range into *N* intervals of equal size: uniform grid
  - if *A* and *B* are the lowest and highest values of the attribute, the width of intervals will be: $W = (B - A)/N$.
  - The most straightforward, but outliers may dominate presentation
  - Skewed data is not handled well
- Equal-depth (frequency) partitioning
  - Divides the range into *N* intervals, each containing approximately same number of samples (roughly constant)
  - Good data scaling
  - Managing categorical attributes can be tricky

Binning Methods for Data Smoothing

Sorted data for price (in dollars): 2,10,18,18,19,20,22,25,28

* Partition into equal-frequency (**equi-depth**) bins:

- Bin 1: 2,10,18
- Bin 2:18,19,20
- Bin 3: 22,25,28

* Smoothing by **bin means**:

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

- Smoothing by **bin medians**:
  - Bin 1: 10,10,10
  - Bin 2:19,19,19
  - Bin 3:25,25,25
- Smoothing by **bin boundaries**:
  - Bin 1: 2,2,18
  - Bin 2: 18,18,20
  - Bin 3: 22,22,28

Discretization by Classification & Correlation Analysis

- Classification (e.g., decision tree analysis)
  - Supervised: Given class labels, e.g., cancerous vs. benign
  - Using *entropy* to determine split point (discretization point)
  - Top-down, recursive split
  - Details to be covered in Chapter 7
- Correlation analysis (e.g., Chi-merge: χ^2-based discretization)
  - Supervised: use class information
  - Bottom-up merge: find the best neighboring intervals (those having similar distributions of classes, i.e., low χ^2 values) to merge
  - Merge performed recursively, until a predefined stopping condition

Concept Hierarchy Generation

- **Concept hierarchy** organizes concepts (i.e., attribute values) hierarchically and is usually associated with each dimension in a data warehouse
- Concept hierarchies facilitate drilling and rolling in data warehouses to view data in multiple granularity
- Concept hierarchy formation: Recursively reduce the data by collecting and replacing low level concepts (such as numeric values for *age*) by higher level concepts (such as *youth, adult,* or *senior*)
- Concept hierarchies can be explicitly specified by domain experts and/or data warehouse designers
- Concept hierarchy can be automatically formed for both numeric and nominal data. For numeric data, use discretization methods shown.

Concept Hierarchy Generation for Nominal Data

- **Specification of a partial/total ordering of attributes explicitly at the schema level by users or experts**

  Ex: if the dimension *address* contains a group of attributes, namely *house_no, street, city, state and country* then a hierarchy can be build by specifying the total ordering among these attributes such as

  House_no < street < city < country

- **Specification of portion of a hierarchy by explicit data grouping**
- Ex: if state and country from a hierarchy at schema level, then a user could manually define some intermediate levels such as

  {Saibaba colony, RS Puram} ¢ Coimbatore
- **Specification of only a partial set of attributes**

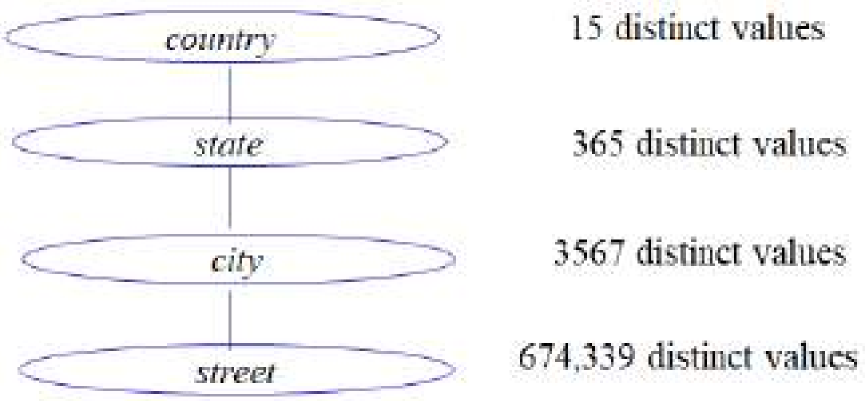
  E.g., only *street* < *city*, not others
- Automatic generation of hierarchies (or attribute levels) by the analysis of the number of distinct values

  E.g., for a set of attributes: {*street*, *city*, *state*, *country*}

Automatic Concept Hierarchy Generation

Some hierarchies can be automatically generated based on the analysis of the number of distinct values per attribute in the data set

- The attribute with the most distinct values is placed at the lowest level of the hierarchy
- Exceptions, e.g., weekday, month, quarter, year



Data Mining Query Language (DMQL)

- The most desirable features of a data mining system is to provide interactivity to the end-users in order **to facilitate flexible and effective knowledge discovery.**
- DMQL is designed on the basis of data mining primitives to incorporate such feature.
- DMQL allows the mining of some specific kinds of knowledge from relational databases and data warehouses at multiple levels of abstraction.
- It helps in standardizing the development of platforms for data mining systems so that the communication with other information systems is possible world-wide.
- This is because the language has adopted an SQL – like syntax which can be easily integrated with the relational query language, SQL.

Example:

Suppose a user wants to describe the general characteristics of students in TNAU university database. The database consists of attributes *name, gender, birth_place, birth_date, stream, address, telephone_number and ogpa.*

For this characterization, a data mining query can be expressed in data mining query language (DMQL) as follows

use TNAU_university_DB
mine characteristics as "AIT_Students"
in relevance to name, gender, birth_place, birth_date, stream,address, telephone_number, ogpa
from student
where status in "post-graduate"

The clause "in relevance to " is used to specify the set of relevant attributes
(Note : the user selects only a few attribute from his/her point of view while missing others important in the description. Ex: let the dimension *birth_place* be defined by attributes *city,state and country*)

where clause indicates that a concept hierarchy exists for the attribute *status* which organizes low-level data (eg. M.Sc. Meteorology, B.Tech AIT) into higher conceptual levels (eg. Postgraduate, graduate)

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

use TNAU_university_DB
select name, gender, birth_place, birth_date, stream,address, telephone_number, gpa
from student
where status in {"M.Sc. Meteorology, M.Sc. Remote sensing"}

Now, this transformed query returns the following result as

name	**gender**	**birth_place**	**steam**	**birth_date**	**Address**	**Tele_no**	**0gpa**
Ram	M	Coimbatore	AIT	14-05-1997	12/3,KK Nagar, Coimbatore	9894123413	8.7
Vimala	F	Erode	Agri	02-03-1998	102, JJ Nagar, Erode	7823456109	8.2
---	----	-----	----	-----	-----	----	---

Unit II – Association and Classification

Mining Association Rules

- Proposed by Rakesh Agrawal et al in 1993.
- It is an important data mining model studied extensively by the database and data mining community.
- The term association describes a relationship between a set of items that people tend to buy together .
- For example, if customers buy two-wheelers, there is a possibility that they also buy some accessories such as seat cover, helmet, gloves, etc.
- The discovery of association rules is one of the major tasks involved in data mining.
- The task of mining association rules is to find interesting relationship among various items in a given data set.
- Let us consider some examples of associations given as follows:
- A person who buys a mobile is also likely to buy some accessories such as mobile cover,hands free,etc.
- A person who buys bread is also likely to buy butter and jam.
- Someone who buys eggs is also likely to buy bread.
- Someone who bought the book Data Structure using C is also likely to buy programming in C.
- An association rule has the form X → Y

  Where X={$X_{1,}X_{2,...}X_n$} and Y={$Y_{1,}Y_{2,...}Y_n$} are the disjoint sets of items,
  that is, $X \cap Y = \varnothing$. It stands that if a person buys an item X, he/she is likely
  to buy an item Y.
- The set $X \cup Y$ is called an ***itemset*** – a set of items a customer tends to buy together. The item X is called the ***antecedent***, while Y is called the ***consequent*** of the rule.
- An example of association rule is

  Mobile → mobile cover, head set

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

■ **Support** (also known as **prevalence**):

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

  support (X → Y) = P ($X \cup Y$)

For example , suppose only 0.002% of the customers buy mobile and chocolates, then the support for the rule *mobile* → *chocolates* will be low.

- **Confidence** (also known as **strength**) ;
  - It is the probability that a customer will buy the items in the set Y if he/she purchases the items in the set X. It is computed as

    $$\text{Confidence } (X \rightarrow Y) = P(Y|X) = \text{support } (X \cup Y) / \text{support } (X)$$

    Ex : the association rule *mobile* → *mobile cover* has he confidence of 80% if 80% of the purchase that include mobile also include mobile cover.
- In general, association rules are said to be interesting or strong if they satisfy both a minimum support threshold (min_sup) and minimum confidence threshold (min_conf).

Association rule mining

- Initially used for Market Basket Analysis to find how items purchased by customers are related.

  Bread → Milk [sup = 5%, conf = 100%]
- Objective:

  Given a transaction T, the problem of mining association rules is to discover all association rules in T that have support and confidence greater or equal to user specified **minimum support (minsup)** and **minimum confidence (minconf).**

The model: data

- $I = \{i_1, i_2, \ldots, i_m\}$: a set of *items*.
- Transaction *t* :

  t a set of items, and $t \subseteq I$.
- Transaction Database *T*: a set of transactions $T = \{t_1, t_2, \ldots, t_n\}$.

Transaction data: supermarket data

- Market basket transactions:

  t1: {bread, cheese, milk}

  t2: {apple, eggs, salt, yogurt}

  … …

  tn: {biscuit, eggs, milk}
- Concepts:
  - An *item*: an item/article in a basket
  - *I*: the set of all items sold in the store
  - A *transaction*: items purchased in a basket; it may have TID (transaction ID)
  - A *transactional dataset*: A set of transactions

The model: rules

- A transaction *t* contains *X*, a set of items (itemset) in *I*, if $X \subseteq t$.
- An association rule is an implication of the form:

  $X \rightarrow Y$, where $X, Y \subset I$, *and* $X \cap Y = \emptyset$
- An itemset is a set of items.

  E.g., X = {milk, bread, butter} is an itemset.
- A *k*-itemset is an itemset with *k* items.

  E.g., {milk, bread, butter} is a 3-itemset

Rule strength measures

- Support: The rule holds with support *sup* in *T* (the transaction data set) if sup% of transactions contain $X \cup Y$.

  $sup = \Pr(X \cup Y)$.
- Confidence: The rule holds in *T* with confidence *conf* if *conf*% of transactions that contain *X* also contain *Y*.

  $conf = \Pr(Y \mid X)$
- An association rule X-> Y states that when *X* occurs, *Y* occurs with certain probability.

Support and Confidence

- Support count: The support count of an itemset *X*, denoted by *X.count*, in a data set *T* is the number of transactions in *T* that contain *X*. Assume *T* has *n* transactions.
- Then,

$$support = \frac{(X \cup Y).count}{n}$$

$$confidence = \frac{(X \cup Y).count}{X.count}$$

Goal and key features

- Goal: Find all rules that satisfy the user-specified *minimum support* (minsup) and *minimum confidence* (minconf).
- Key Features
  - Completeness: find all rules.
  - No target item(s) on the right-hand-side
  - Mining with data on hard disk (not in memory)

An example

- Transaction data →
- Assume:
  minsup = 30%
  minconf = 80%
- An example frequent *itemset (also called large itemset whose support is greater than minsup)*:
  {Chicken, Clothes, Milk} [sup = 3/7]
- Association rules from the itemset:
  Clothes → Milk, Chicken [sup = 3/7, conf = 3/3]
  …
  Clothes, Chicken → Milk, [sup = 3/7, conf = 3/3]

t1:	Beef, Chicken, Milk
t2:	Beef, Cheese
t3:	Cheese, Boots
t4:	Beef, Chicken, Cheese
t5:	Beef, Chicken, Clothes, Cheese, Milk
t6:	Chicken, Clothes, Milk
t7:	Chicken, Milk, Clothes

Transaction data representation

- A simplistic view of shopping baskets,
- Some important information not considered.
  E.g, the quantity of each item purchased and the price paid.

Many mining algorithms

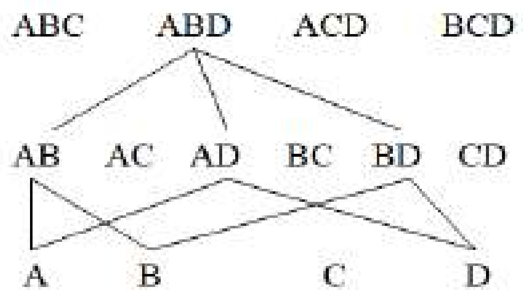
- There are a large number of them!!
- They use different strategies and data structures.
- Their resulting sets of rules are all the same.
  Given a transaction data set *T*, and a minimum support and a minimum confidence, the set of association rules existing in *T* is uniquely determined.
- Any algorithm should find the same set of rules although their computational efficiencies and memory requirements may be different.
- We will discuss only one: the Apriori Algorithm

The Apriori algorithm

- The best known algorithm
- Two steps:
  - Find all itemsets that have minimum support (*frequent itemsets*, also called large itemsets).
  - Use frequent itemsets to generate rules.
- E.g., a frequent itemset
  {Chicken, Clothes, Milk} [sup = 3/7]
  and one rule from the frequent itemset
  Clothes → Milk, Chicken [sup = 3/7, conf = 3/3]

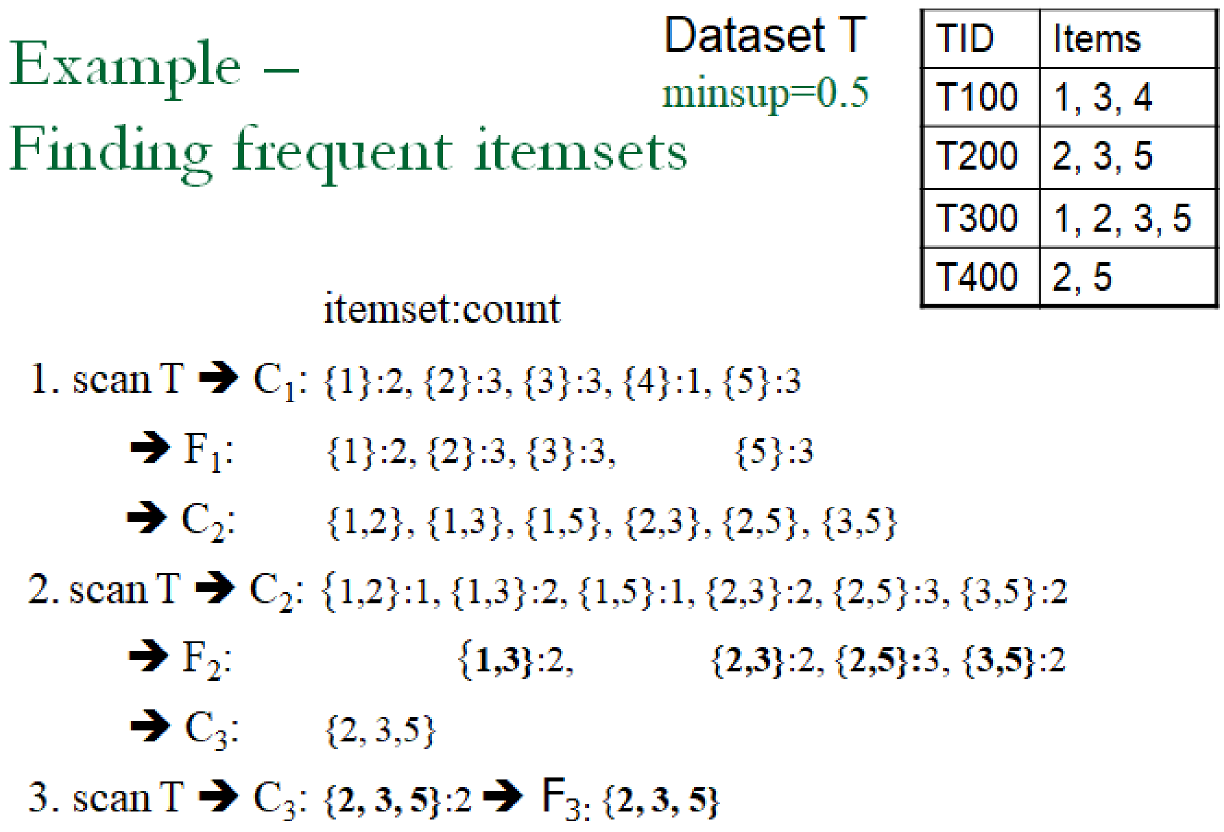
Step 1: Mining all frequent itemsets

- A frequent *itemset* is an itemset whose support is ≥ minsup.
- Key idea: The apriori property (downward closure property): any subsets of a frequent itemset are also frequent itemsets



The Algorithm

- **Iterative algo.** (also called level-wise search): Find all 1-item frequent itemsets; then all 2-item frequent itemsets, and so on.
    - In each iteration *k*, only consider itemsets that contain some *k*-1 frequent itemset.
- Find frequent itemsets of size 1: F_1
- From $k = 2$
    - C_k = candidates of size *k*: those itemsets of size *k* that could be frequent, given F_{k-1}
    - F_k = those itemsets that are actually frequent, $F_k \subseteq C_k$ (need to scan the database once).



Example – Finding frequent itemsets

Dataset T

minsup=0.5

TID	Items
T100	1, 3, 4
T200	2, 3, 5
T300	1, 2, 3, 5
T400	2, 5

itemset:count

1. scan T ➔ C_1: {1}:2, {2}:3, {3}:3, {4}:1, {5}:3

➔ F_1: {1}:2, {2}:3, {3}:3, {5}:3

➔ C_2: {1,2}, {1,3}, {1,5}, {2,3}, {2,5}, {3,5}

2. scan T ➔ C_2: {1,2}:1, {1,3}:2, {1,5}:1, {2,3}:2, {2,5}:3, {3,5}:2

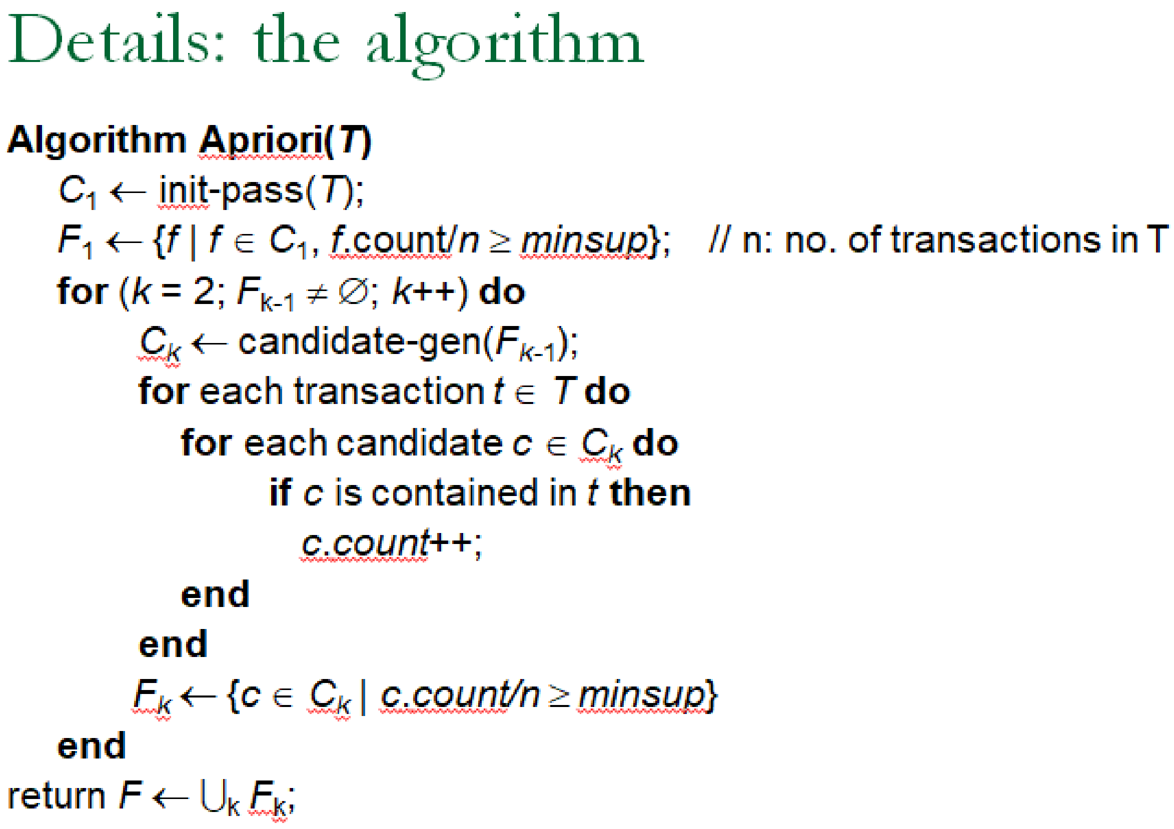
➔ F_2: **{1,3}**:2, **{2,3}**:2, **{2,5}**:3, **{3,5}**:2

➔ C_3: {2, 3,5}

3. scan T ➔ C_3: **{2, 3, 5}**:2 ➔ F_3: **{2, 3, 5}**

Details: ordering of items

- The items in *I* are sorted in lexicographic order (which is a total order).
- The order is used throughout the algorithm in each itemset.
- {*w*[1], *w*[2], …, *w*[*k*]} represents a *k*-itemset *w* consisting of items *w*[1], *w*[2], …, *w*[*k*], where *w*[1] < *w*[2] < … < *w*[*k*] according to the total order.

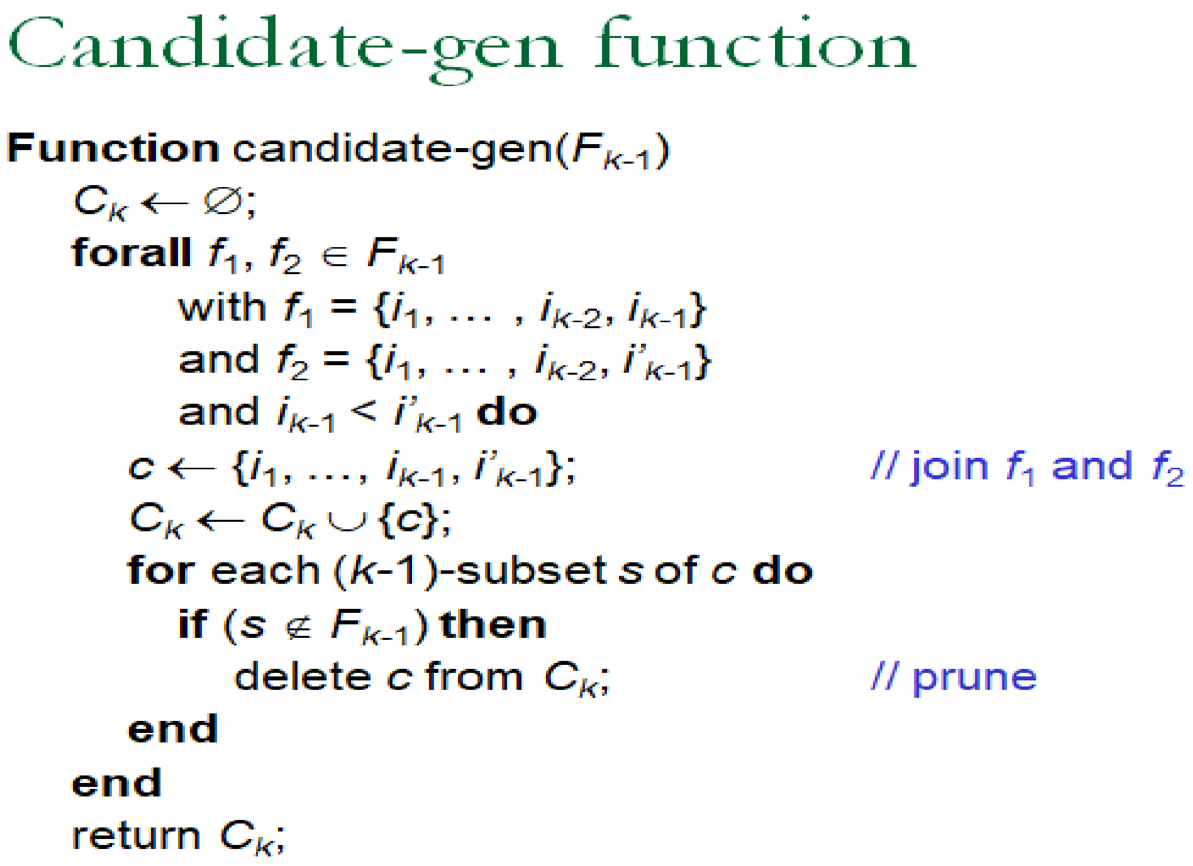


Details: the algorithm

```
Algorithm Apriori($T$)
    $C_1 \leftarrow$ init-pass($T$);
    $F_1 \leftarrow \{f \mid f \in C_1, f.count/n \geq minsup\}$;    // n: no. of transactions in T
    for ($k = 2$; $F_{k-1} \neq \emptyset$; $k$++) do
        $C_k \leftarrow$ candidate-gen($F_{k-1}$);
        for each transaction $t \in T$ do
            for each candidate $c \in C_k$ do
                if $c$ is contained in $t$ then
                    $c.count$++;
            end
        end
        $F_k \leftarrow \{c \in C_k \mid c.count/n \geq minsup\}$
    end
return $F \leftarrow \bigcup_k F_k$;
```

Apriori candidate generation

- The candidate-gen function takes $F_{k\text{-}1}$ and returns a superset (called the candidates) of the set of all frequent k-itemsets. It has two steps
  - *join* step: Generate all possible candidate itemsets C_k of length k
  - *prune* step: Remove those candidates in C_k that cannot be frequent.



Candidate-gen function

```
Function candidate-gen($F_{k-1}$)
    $C_k \leftarrow \varnothing$;
    forall $f_1, f_2 \in F_{k-1}$
            with $f_1 = \{i_1, \ldots, i_{k-2}, i_{k-1}\}$
            and $f_2 = \{i_1, \ldots, i_{k-2}, i'_{k-1}\}$
            and $i_{k-1} < i'_{k-1}$ do
        $c \leftarrow \{i_1, \ldots, i_{k-1}, i'_{k-1}\}$;             // join $f_1$ and $f_2$
        $C_k \leftarrow C_k \cup \{c\}$;
        for each $(k-1)$-subset $s$ of $c$ do
            if ($s \notin F_{k-1}$) then
                delete $c$ from $C_k$;                  // prune
        end
    end
    return $C_k$;
```

Example

- $F_3 = \{\{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\},$
  $\{1, 3, 5\}, \{2, 3, 4\}\}$
- After join
  $C_4 = \{\{1, 2, 3, 4\}, \{1, 3, 4, 5\}\}$
- After pruning:
  $C_4 = \{\{1, 2, 3, 4\}\}$
  because $\{1, 4, 5\}$ is not in F_3 ($\{1, 3, 4, 5\}$ is removed)

Step 2: Generating rules from frequent itemsets

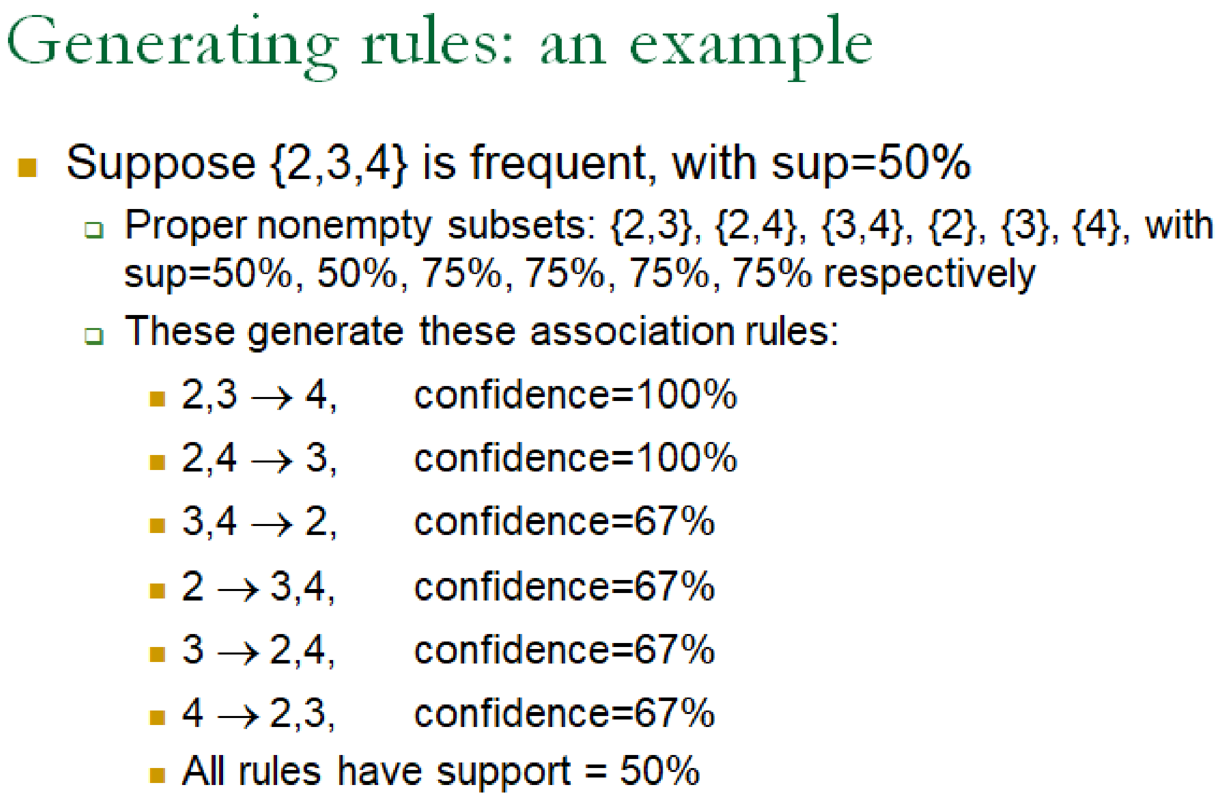
- Frequent itemsets ≠ association rules
- One more step is needed to generate association rules
- For each frequent itemset *X*,
  For each proper nonempty subset *A* of *X*,
  Let *B* = X - *A*
  A → B is an association rule if

confidence(A → B) ≥ minconf,

Where

$$\text{confidence}(A \rightarrow B) = \text{support}(A \rightarrow B) / \text{support}(A)$$

$$\text{support}(A \rightarrow B) = \text{support}(A \cup B) = \text{support}(X)$$



Generating rules: an example

- Suppose {2,3,4} is frequent, with sup=50%
  - Proper nonempty subsets: {2,3}, {2,4}, {3,4}, {2}, {3}, {4}, with sup=50%, 50%, 75%, 75%, 75%, 75% respectively
  - These generate these association rules:
    - 2,3 → 4, confidence=100%
    - 2,4 → 3, confidence=100%
    - 3,4 → 2, confidence=67%
    - 2 → 3,4, confidence=67%
    - 3 → 2,4, confidence=67%
    - 4 → 2,3, confidence=67%
    - All rules have support = 50%

Generating rules: summary

- To recap, in order to obtain A → B, we need to have support($A \cup B$) and support(A)
- All the required information for confidence computation has already been recorded in itemset generation. No need to see the data *T* any more.
- This step is not as time-consuming as frequent itemsets generation.

On Apriori Algorithm

Seems to be very expensive

- Level-wise search
- K = the size of the largest itemset
- It makes at most K passes over data
- In practice, K is bounded (10).
- The algorithm is very fast. Under some conditions, all rules can be found in linear time.
- Downward closure property makes it efficient
- Scale up to large data sets

More on association rule mining

- Clearly the space of all association rules is exponential, O(2^m), where m is the number of items in *I*.
- The mining exploits sparseness of data, and high minimum support and high minimum confidence values.
- Still, it always produces a huge number of rules, thousands, tens of thousands, millions, ...

Classification and Prediction

There are two forms of data analysis that can be used for extracting models describing important classes or to predict future data trends. These two forms are as follows −

- Classification
- Prediction

Classification

- Refers to partitioning the given data into predefined disjoint groups or classes
- A model (also known as classifier) is built to predict the class of a new item
- Classification models predict categorical class labels
- Ex: we can build a classification model to categorize bank loan applications as either safe or risky

Prediction

- prediction models predict continuous valued functions
- Ex: a prediction model to predict the expenditures in dollars of potential customers on computer equipment given their income and occupation.

What is classification?

Following are the examples of cases where the data analysis task is Classification −

- A bank loan officer wants to analyze the data in order to know which customer (loan applicant) are risky or which are safe.
- A marketing manager at a company needs to analyze a customer with a given profile, who will buy a new computer.

In both of the above examples, a model or classifier is constructed to predict the categorical labels. These labels are risky or safe for loan application data and yes or no for marketing data.

What is prediction?

Following are the examples of cases where the data analysis task is Prediction −

Suppose the marketing manager needs to predict how much a given customer will spend during a sale at his company. In this example we are bothered to predict a numeric value. Therefore the data analysis task is an example of numeric prediction. In this case, a model or a predictor will be constructed that predicts a continuous-valued-function or ordered value. i.e., predicts unknown or missing values

Note − Regression analysis is a statistical methodology that is most often used for numeric prediction.

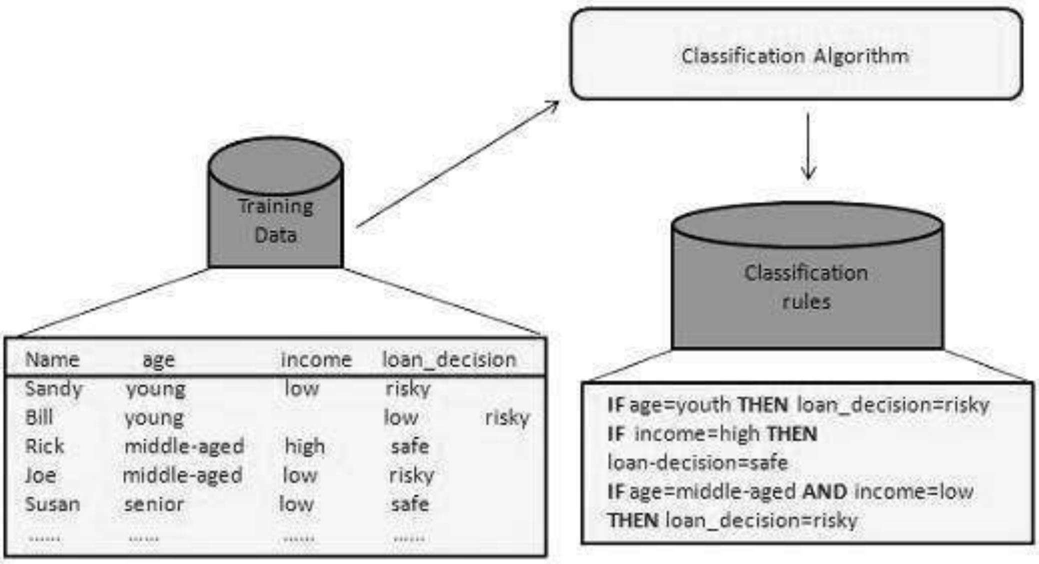
How Does Classification Works?

With the help of the bank loan application that we have discussed above, let us understand the working of classification.

The Data Classification process includes two steps –

1) **Model construction**: describing a set of predetermined classes
   (Building the Classifier or Model)
   - Each tuple is assumed to belong to a predefined class, as determined by the class label attribute (**supervised learning**)
   - The set of tuples used for model construction: training set
   - The model is represented as classification rules, decision trees, or mathematical formulae

- This step is the learning step or the learning phase.
- In this step the classification algorithms build the classifier.
- The classifier is built from the training set made up of database tuples and their associated class labels.
- Each tuple that constitutes the training set is referred to as a category or class. These tuples can also be referred to as sample, object or data points.



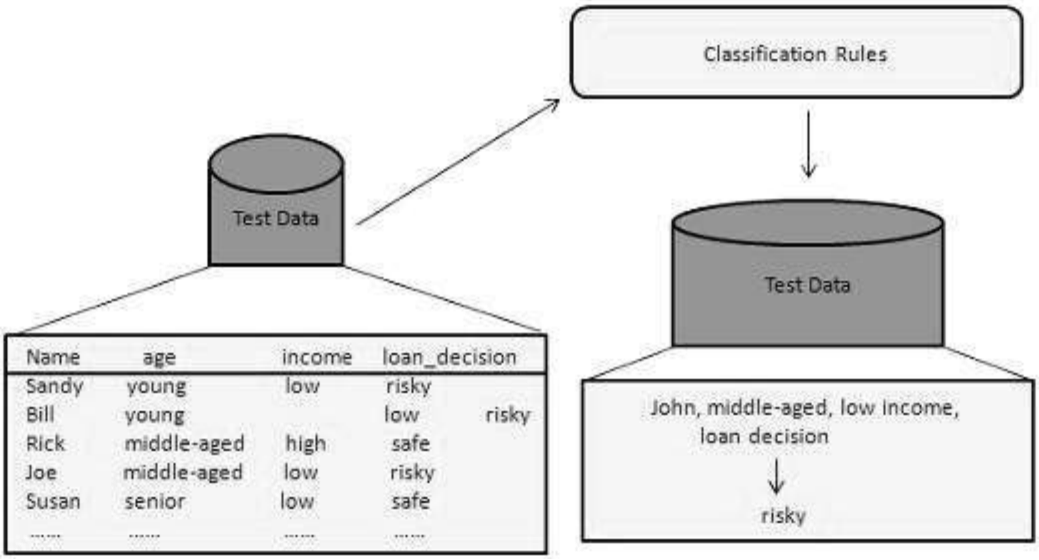
Name	age	income	loan_decision	
Sandy	young	low	risky	
Bill	young		low	risky
Rick	middle-aged	high	safe	
Joe	middle-aged	low	risky	
Susan	senior	low	safe	
......	......	......	......	

IF age=youth **THEN** loan_decision=risky
IF income=high **THEN**
loan-decision=safe
IF age=middle-aged **AND** income=low
THEN loan_decision=risky

2) **Model usage:** for classifying previously unseen objects
(Using Classifier for Classification)

– Estimate accuracy of the model using a test set
  - The known label of test sample is compared with the classified result from the model
  - Accuracy rate is the percentage of test set samples that are correctly classified by the model
  - Test set is independent of training set, otherwise over-fitting will occur

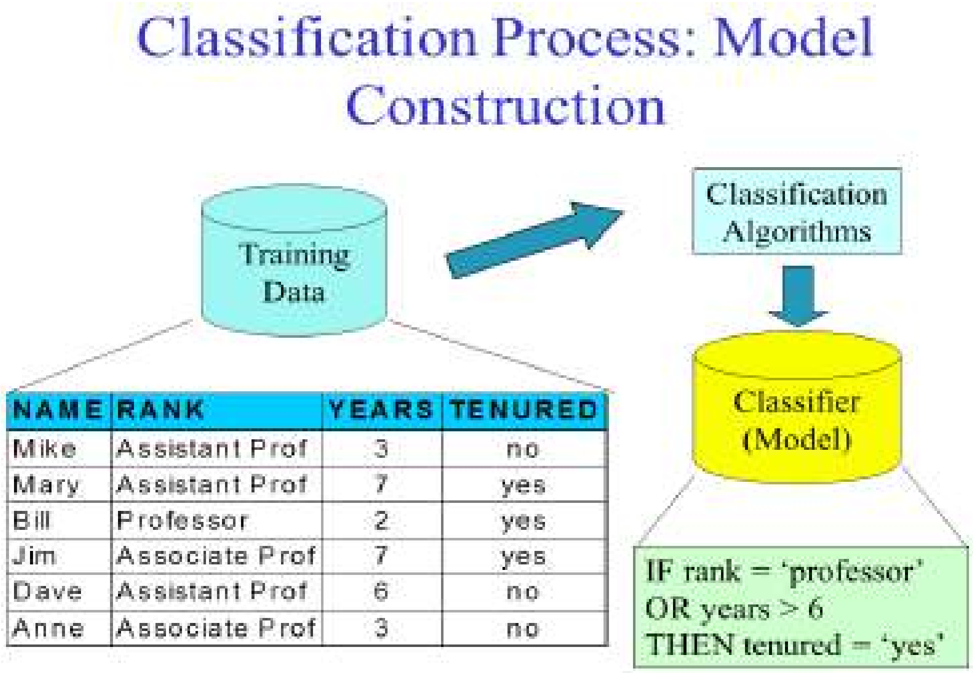
In this step, the classifier is used for classification. Here the test data is used to estimate the accuracy of classification rules. The classification rules can be applied to the new data tuples if the accuracy is considered acceptable.



Name	age	income	loan_decision	
Sandy	young	low	risky	
Bill	young		low	risky
Rick	middle-aged	high	safe	
Joe	middle-aged	low	risky	
Susan	senior	low	safe	
......	......	......	......	

John, middle-aged, low income,
loan decision
↓
risky

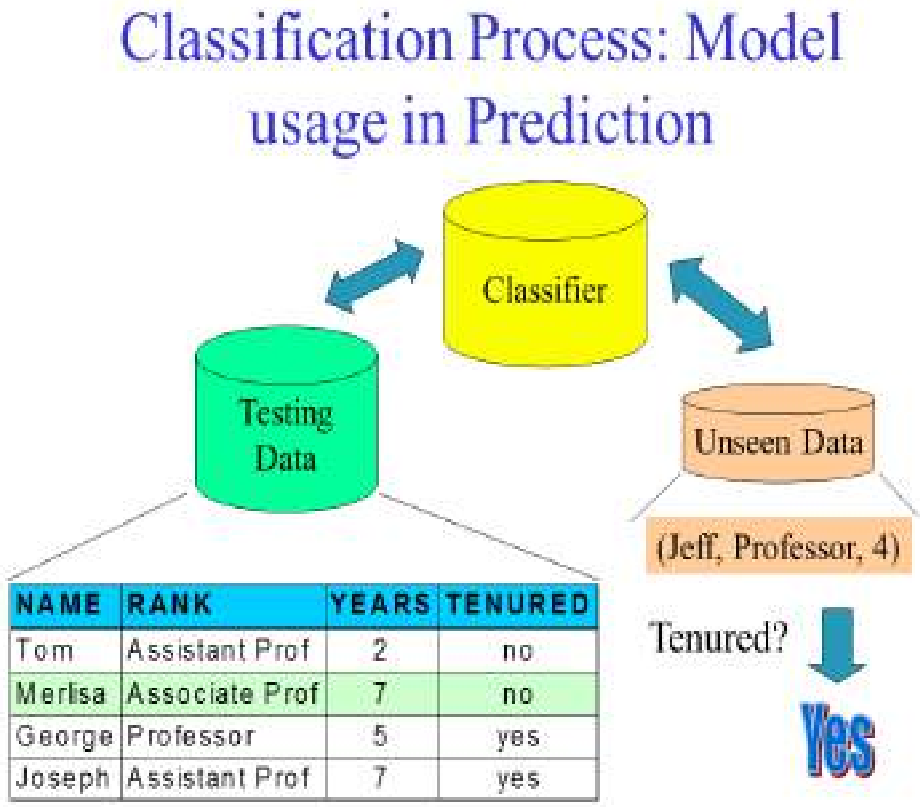
Example : 2



Classification Process: Model Construction

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'



Classification Process: Model usage in Prediction

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

Classification and Prediction Issues

a) Issues regarding classification and prediction: Data Preparation

The major issue is preparing the data for Classification and Prediction. Preparing the data involves the following activities −

- **Data Cleaning** − Data cleaning involves removing the noise and treatment of missing values(handle missing value). The noise is removed by applying smoothing techniques and the problem of missing values is solved by replacing a missing value with most commonly occurring value for that attribute.
- **Relevance Analysis** − Database may also have the irrelevant attributes. Correlation analysis is used to know whether any two given attributes are related.
    - Remove the irrelevant or redundant attributes
- **Data Transformation and reduction** − The data can be transformed by any of the following methods.
    - **Normalization** − The data is transformed using normalization. Normalization involves scaling all values for given attribute in order to make them fall within a small specified range. Normalization is used when in the learning step, the neural networks or the methods involving measurements are used.
    - **Generalization** − The data can also be transformed by generalizing it to the higher concept. For this purpose we can use the concept hierarchies.

Note − Data can also be reduced by some other methods such as wavelet transformation, binning, histogram analysis, and clustering.

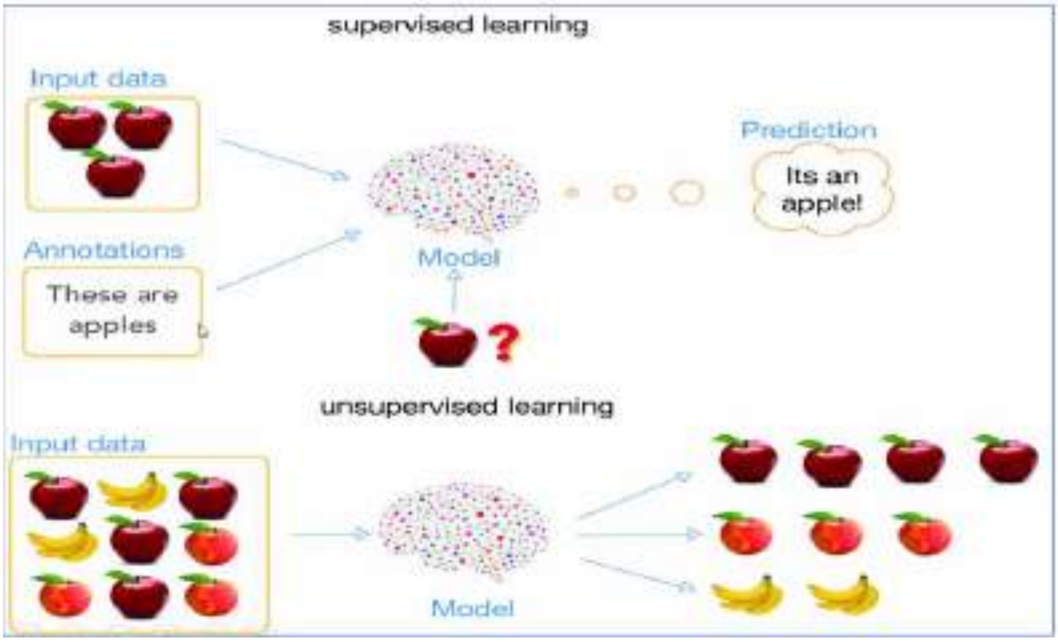
b) Issues regarding classification and prediction: Evaluating Classification Methods

- **Accuracy** − Accuracy of classifier refers to the ability of classifier. It predict the class label correctly and the accuracy of the predictor refers to how well a given predictor can guess the value of predicted attribute for a new data.
- **Speed** − This refers to the computational cost in generating and using the classifier or predictor.
    - time to construct the model
    - time to use the model
- **Robustness** − It refers to the ability of classifier or predictor to make correct predictions from given noisy data.
    - handling noise and missing values

- **Scalability** – Scalability refers to the ability to construct the classifier or predictor efficiently; given large amount of data.
    - efficiency in disk-resident databases
- **Interpretability** – It refers to what extent the classifier or predictor understands.
    - understanding and insight provided by the model
- **Goodness of rules**
    - decision tree size
    - compactness of classification rules

Supervised vs. Unsupervised Learning

- Supervised learning (classification)
    - Supervision: The training data (observation`s, measurements, etc.) are accompanied by labels indicating the class of the observations
    - New data is classified based on the training set
- Unsupervised learning (clustering)
    - The class labels of training data is unknown
    - Given a set of measurements, observations, etc. the aim is to establish the existence of classes or clusters in the data

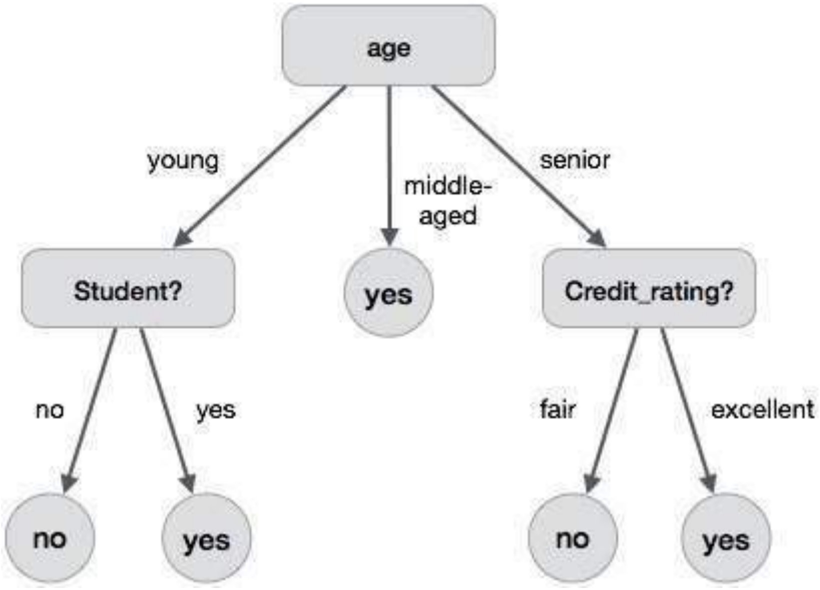


Decision Tree Induction

- It is a graphical representation of the classification rules.
- A decision tree is also known as classification tree.
- Decision tree is a flowchart –like tree structure which relates conditions and actions sequentially.

A decision tree is a structure that includes a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node.

The following decision tree is for the concept buy computer that indicates whether a customer at a company is likely to buy a computer or not. Each internal node represents a test on an attribute. Each leaf node represents a class.



The benefits of having a decision tree are as follows −

- It does not require any domain knowledge.
- It is easy to comprehend.
- The learning and classification steps of a decision tree are simple and fast.

Decision Tree Induction Algorithm

The three decision tree induction algorithms are as follows

1. ID3 (Iterative Dichotomiser) :

A machine researcher named J. Ross Quinlan in 1980 developed a decision tree algorithm known as ID3 (Iterative Dichotomiser).

Which provided a broad approach for learning decision trees from training tuples.

2. **C4.5:**

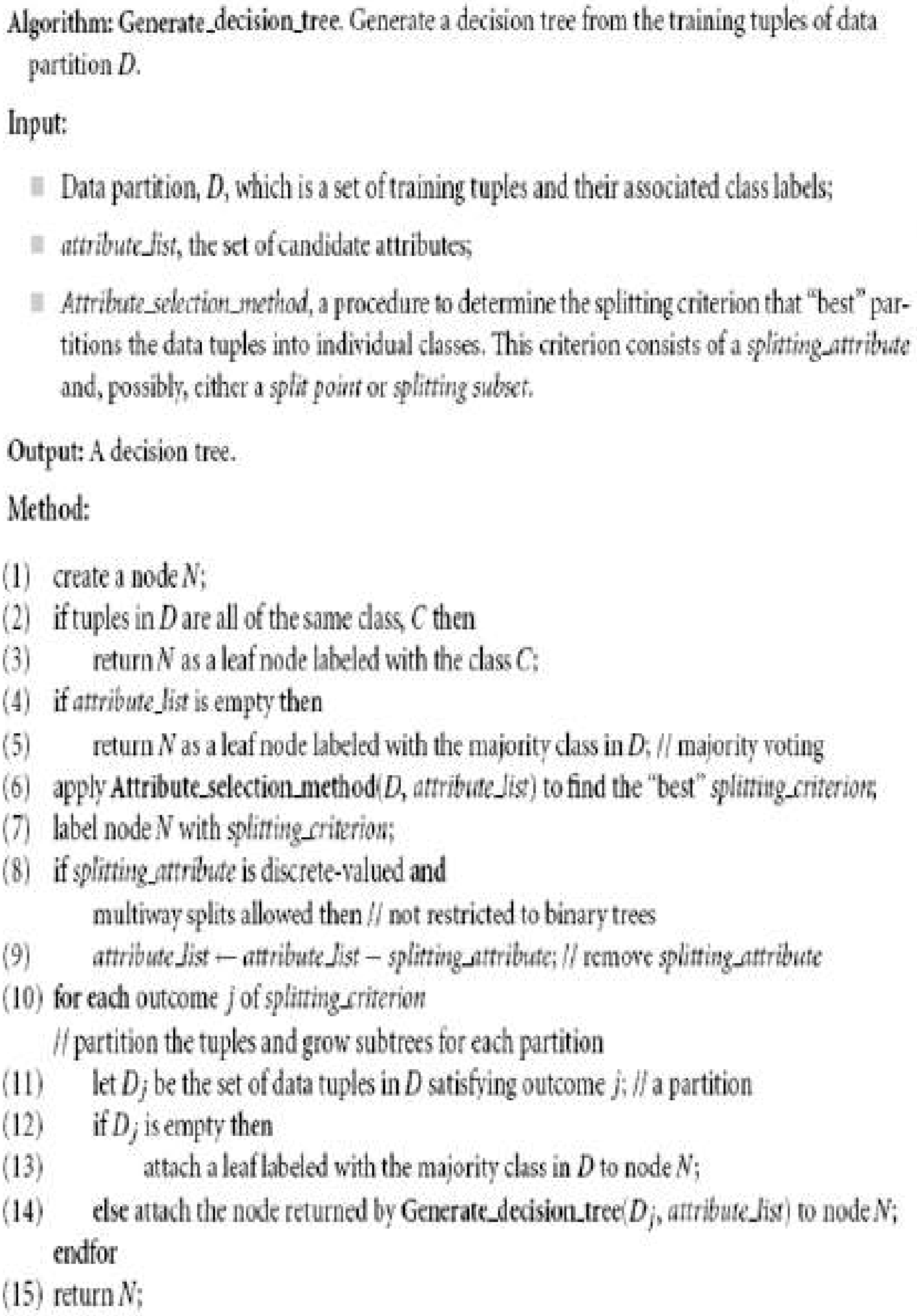
   This algorithm is a successor of ID3 which was developed by Quinlan. It become a bench mark to which newer supervised learning algorithms are often compared.

3. **CART (Classification And Regression Trees)**

   This algorithm was developed by a group of statisticians named L.Breiman, J.Friedman, R.Olshen and C.Stone in 1984. It describes the generation of binary decision trees.

All these algorithms follow a greedy (ie. no backtracking) approach. It means that the trees are constructed in a top-down, recursive, divide-and-conquer manner.

- In such an approach, the training set is recursively partitioned into smaller subsets as the tree is being built.
- The algorithm searches attributes of the training set and extracts the attribute that best partitions the given instances.
- This attribute is called the partitioning(or splitting)attribute.
- If the partitioning attribute, A, perfectly classifies the training set, the algorithm stops, otherwise it recursively selects the partitioning Attribute and partitioning predicate to create further child nodes.
- The basic difference between these algorithms lies in the selection of the attributes for construction of trees and the mechanisms used for pruning.
- A decision tree algorithm (Generate_DT)generates a decision tree which requires one pass over the training tuples in D for each level of tree is as follows:



Algorithm: Generate_decision_tree. Generate a decision tree from the training tuples of data partition *D*.

Input:

- Data partition, *D*, which is a set of training tuples and their associated class labels;
- *attribute_list*, the set of candidate attributes;
- *Attribute_selection_method*, a procedure to determine the splitting criterion that "best" partitions the data tuples into individual classes. This criterion consists of a *splitting_attribute* and, possibly, either a *split point* or *splitting subset*.

Output: A decision tree.

Method:

```
create a node $N$;
if tuples in $D$ are all of the same class, $C$ then
    return $N$ as a leaf node labeled with the class $C$;
if attribute_list is empty then
    return $N$ as a leaf node labeled with the majority class in $D$; // majority voting
apply Attribute_selection_method($D$, attribute_list) to find the "best" splitting_criterion;
label node $N$ with splitting_criterion;
if splitting_attribute is discrete-valued and
    multiway splits allowed then // not restricted to binary trees
    attribute_list $\leftarrow$ attribute_list $-$ splitting_attribute; // remove splitting_attribute
for each outcome $j$ of splitting_criterion
  // partition the tuples and grow subtrees for each partition
    let $D_j$ be the set of data tuples in $D$ satisfying outcome $j$; // a partition
    if $D_j$ is empty then
        attach a leaf labeled with the majority class in $D$ to node $N$;
    else attach the node returned by Generate_decision_tree($D_j$, attribute_list) to node $N$;
  endfor
return $N$;
```

Unit III - Cluster Analysis

Cluster Analysis

Cluster is a group of objects that belongs to the same class. In other words, similar objects are grouped in one cluster and dissimilar objects are grouped in another cluster.

Clustering

Clustering is the process of making a group of abstract objects into classes of similar objects.

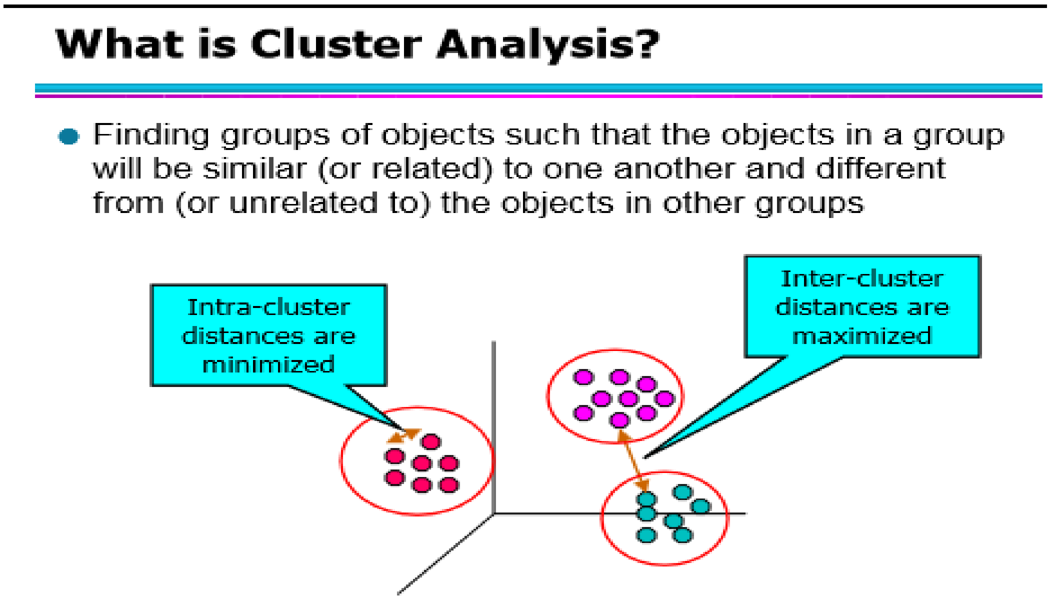
- ➢ Cluster: A collection of data objects
  - similar (or related) to one another within the same group
  - dissimilar (or unrelated) to the objects in other groups

- ➢ Cluster analysis (or *clustering*, *data segmentation, …*)
  - Finding similarities between data according to the characteristics found in the data and grouping similar data objects into clusters

- ➢ Unsupervised learning: no predefined classes (i.e., *learning by observations* vs. learning by examples: supervised)
- ➢ Typical applications
  - As a stand-alone tool to get insight into data distribution
  - As a preprocessing step for other algorithms

Points to Remember

- A cluster of data objects can be treated as one group.
- While doing cluster analysis, we first partition the set of data into groups based on data similarity and then assign the labels to the groups.
- The main advantage of clustering over classification is that, it is adaptable to changes and helps single out useful features that distinguish different groups.



What is Cluster Analysis?

- Finding groups of objects such that the objects in a group will be similar (or related) to one another and different from (or unrelated to) the objects in other groups

Intra-cluster distances are minimized

Inter-cluster distances are maximized

Various application area of Clustering

- ➢ Biology: taxonomy of living things: kingdom, phylum, class, order, family, genus and species
- ➢ Information retrieval: document clustering
- ➢ Land use: Identification of areas of similar land use in an earth observation database
- ➢ Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- ➢ City-plannign: 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
- ➢ Climate: understanding earth climate, find patterns of atmospheric and ocean
- ➢ Economic Science: market research

Requirements of Clustering in Data Mining

The following points throw light on why clustering is required in data mining −

- **Scalability** − We need highly scalable clustering algorithms to deal with large databases.

- **Ability to deal with different kinds of attributes** − Algorithms should be capable to be applied on any kind of data such as interval-based (numerical) data, categorical, and binary data.

- **Discovery of clusters with attribute shape** − The clustering algorithm should be capable of detecting clusters of arbitrary shape. They should not be bounded to only distance measures that tend to find spherical cluster of small sizes.
- **High dimensionality** − The clustering algorithm should not only be able to handle low-dimensional data but also the high dimensional space.
- **Ability to deal with noisy data** − Databases contain noisy, missing or erroneous data. Some algorithms are sensitive to such data and may lead to poor quality clusters.
- **Interpretability** − The clustering results should be interpretable, comprehensible, and usable.

Types of Data in Cluster Analysis

1. **Interval-scaled variables:**

   These are quantitative(or continuous)variable that are measured on a linear scale, and can have both positive and negative values (Eg: weight, height , temperature ,etc)

2. **Binary variables:**

   These variables can have only two values:0 or1.The value 0 depicts that a variable is absent whereas value 1 depicts the presence of the variable. Variables can be either symmetric or asymmetric. **A binary variable** is symmetric if both of its states are considered equally important. For eg, variable gender having the states male and female. On the other hand, a **binary value** is asymmetric if both of its states are not equally important .For eg, Variable HIV test having states positive and negative. That is, if the test outcome of someone comes out to be HIV positive, then its value will be 1, otherwise 0.

   e.g., gender (M/F), has_cancer(T/F), has-COVID-19(T/F)

3. **Nominal variables:**

   These variables are just the generalization of binary variables as they can take on more than two states. For eg, fruit _name is a categorical variable that may have many states such as mango, grapes , apple ,banana and so on. These variables are also known as nominal variables in which there is no specific ordering among states .Because of this reason, one cannot perform logical or arithmetic operations on such variables.

   e.g., religion (Christian, Muslim, Buddhist, Hindu, etc.)

4. Ordinal variables:

These are categorical variables, but states of such variables are ordered in a meaningful sequence. Such variables are comparable only in terms of relative magnitude and not on their actual magnitude. In other words, these variables are useful only for subjective assessment of quality. For eg, we can determine the socio-economic status of families by arranging the states in a sequential order such as **high class, middle class** and **lower class**. From such ranking, one can identify that high class family is richer than middle class, but one cannot say by how much. The ordinal variables can also be obtained from the discretization of interval-scaled quantities by splitting the value range into finite number of classes. Note that the ordinal variables have order, but the intervals between scale points are not uniform. Thus, only logical operations can be performed on the ordinal variables. Arithmetic operations are impossible.

e.g., military rank (soldier, sergeant, captain, etc.)

5. Ratio variables:

These variables are continuous positive measurements on a non- linear scale, such as exponential scale. For eg, the growth of bacterial population (say with a function Ae^{Bt}) and the decay of a radioactive element (with a function Ae^{-Bt}). Here, t represents time, and A and B are positive constants. These functions represent the growth of bacteria or decay of radioactive element by the same ratio in each equal intervals of time ; hence the name is ratio – scaled variable.

e.g., population growth (1,10,100,1000,...)

6. Variables of mixed types :

These variables are a mixture of various types of variables such as interval-scaled, symmetric binary ,asymmetric binary, categorical, ordinal or ratio-scaled.

multiple attributes with various types

7. Vector objects:

These are complex objects (such as documents) containing large number of symbolic entities such as keywords and phrases.

Categorization of Clustering Methods

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

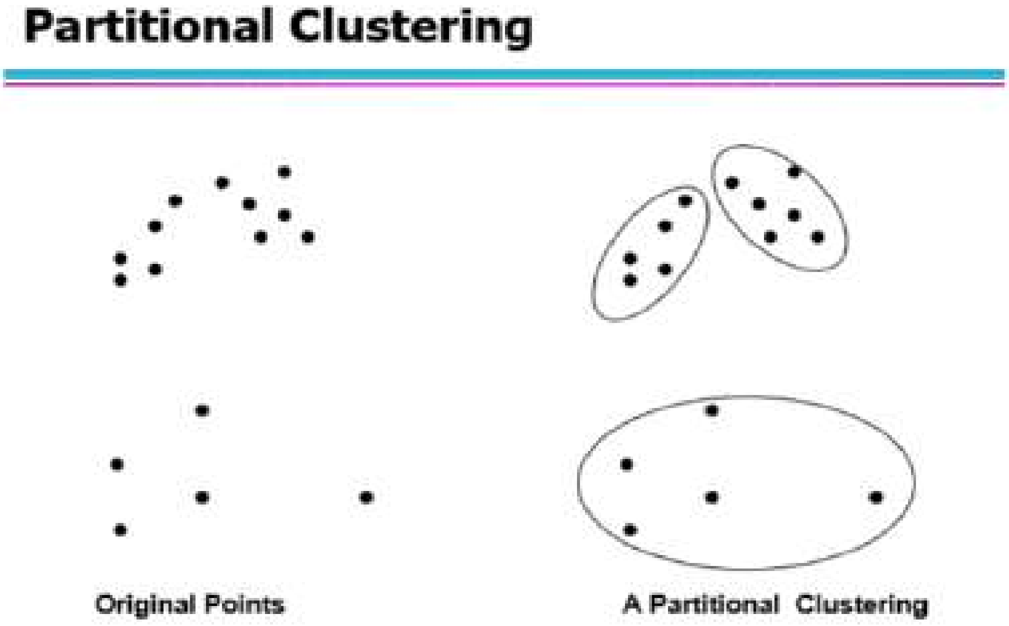
1. Partitioning Method

Suppose we are given a database of **'n'** objects and the partitioning method constructs **'k'** partition of data. Each partition will represent a cluster and k ≤ n. It means that it will classify the data into k groups, which satisfy the following requirements −

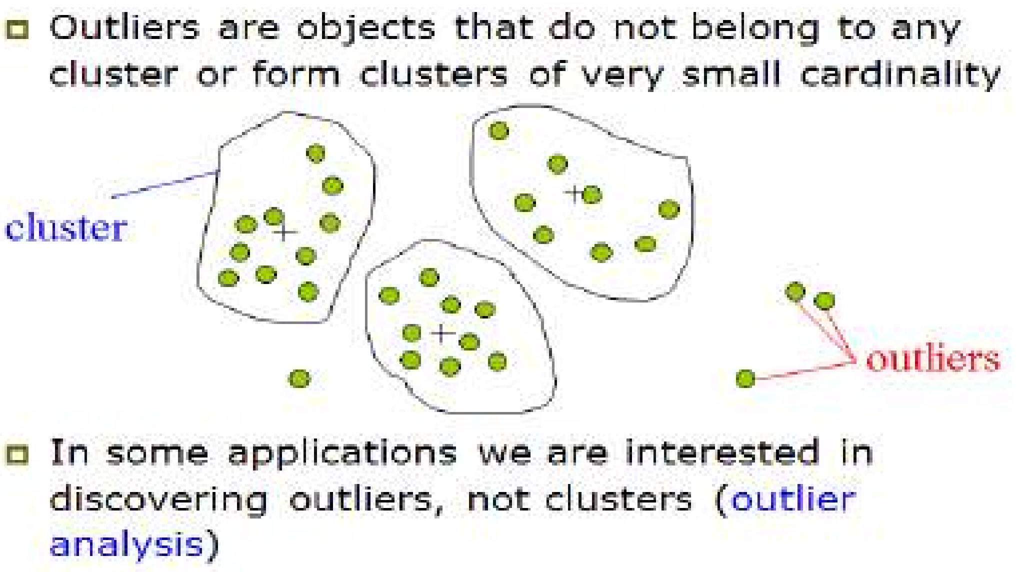
- Each group contains at least one object.
- Each object must belong to exactly one group.

Points to remember −

- For a given number of partitions (say k), the partitioning method will create an initial partitioning.
- Then it uses the iterative relocation technique to improve the partitioning by moving objects from one group to other.



Outliers



- Outliers are objects that do not belong to any cluster or form clusters of very small cardinality

- In some applications we are interested in discovering outliers, not clusters (outlier analysis)

Partitioning Algorithms

<u>Partitioning method:</u> Partitioning a database ***D*** of ***n*** objects into a set of ***k*** clusters, such that the sum of squared distances is minimized (where c_i is the centroid of cluster C_i)

$$E = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

k-means and *k-medoids* algorithms

- ***<u>k-means</u>*** (MacQueen’67, Lloyd’57/’82): Each cluster is represented by the center of the cluster

- ***<u>k-medoids</u>*** **or PAM** (Partition around medoids) (Kaufman & Rousseeuw’87): Each cluster is represented by one of the objects in the cluster

The *K-Means* Clustering Method (*k-means* Algorithm)

- First, It randomly selects k of the objects, each of which initially represents a cluster mean or center.
- An object is assigned to the cluster to which it is the most similar, based on the distance between the object and the cluster mean.
- It then computes the new mean for each cluster
- This process iterates until the criterion function converges.
- The square-error criterion is used as

For each point, the error is the distance to the nearest cluster

To get E, we square these errors and sum them

$$E = \sum_{i=1}^{K} \sum_{x \in C_i} dist^2(m_i, x)$$

Here, E is the sum of square error for all objects in the data set

x is the point in cluster C_i

m_i is the mean of cluster C_i (the centroid of cluster C_i)

Given two clusters, we can choose the one with the smallest error

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.

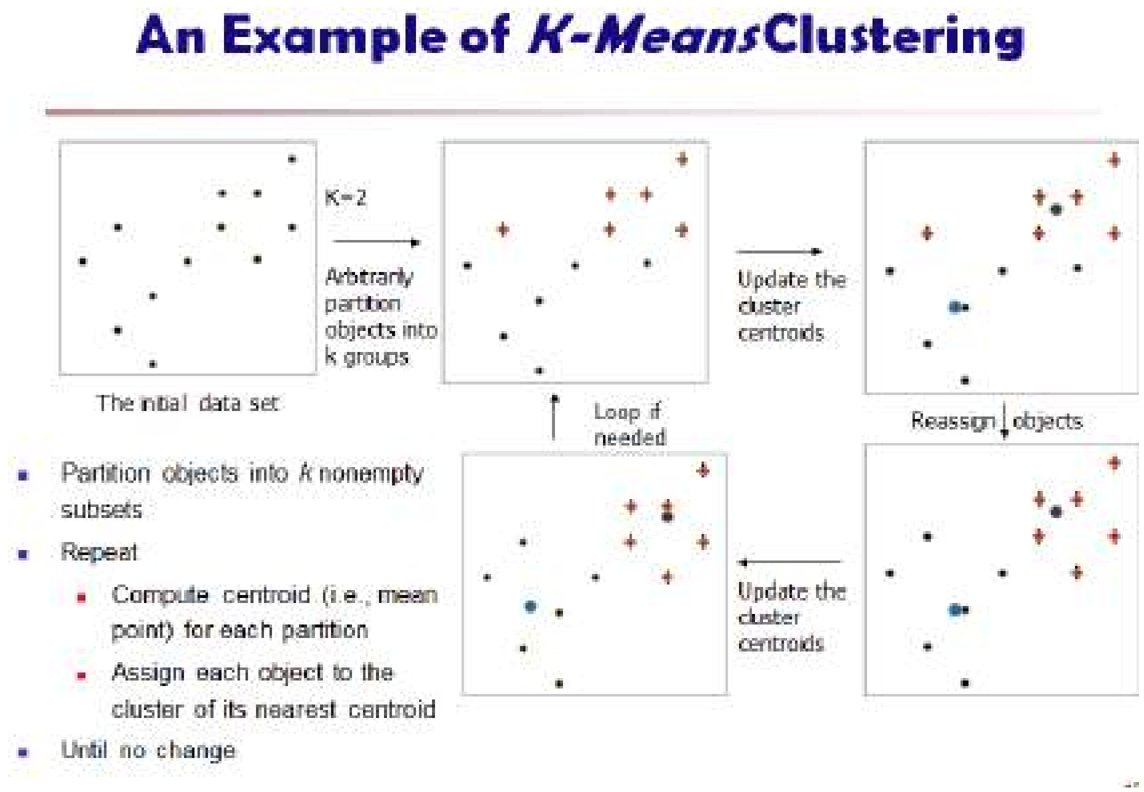
Input :

- 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 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, that is , calculate the mean value of the objects for
           each cluster;
5. Until no change.
```

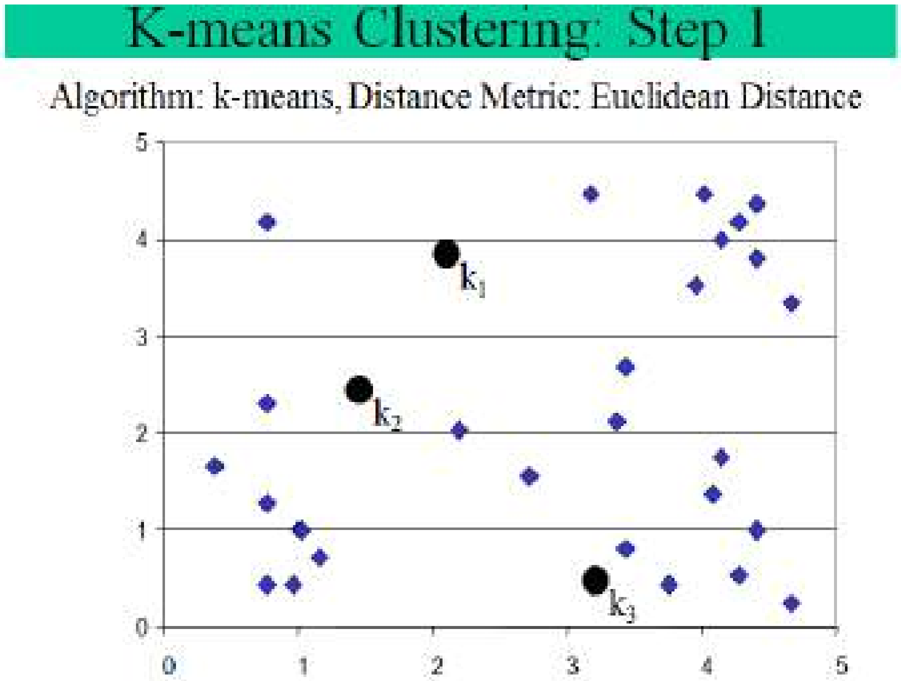


An Example of *K-Means* Clustering

K=2
Arbitrarily partition objects into k groups
The initial data set
Update the cluster centroids
Reassign objects
Loop if needed
Update the cluster centroids

- Partition objects into *k* nonempty subsets
- Repeat
  - Compute centroid (i.e., mean point) for each partition
  - Assign each object to the cluster of its nearest centroid
- Until no change

Example 2

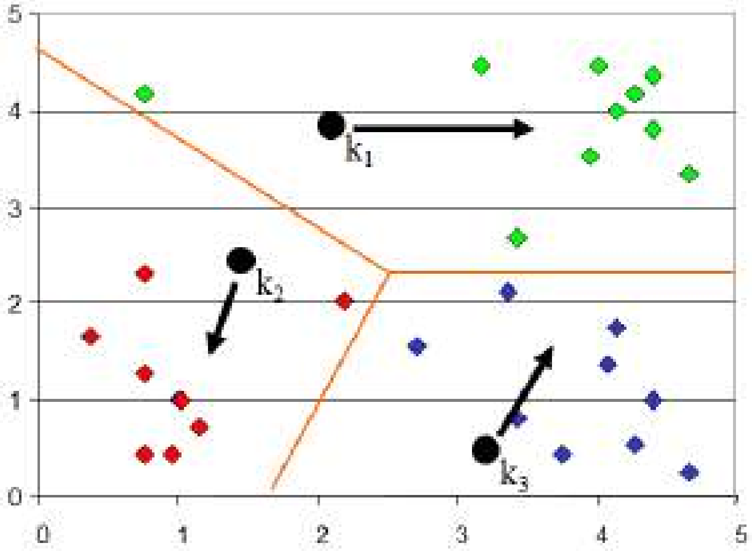


K-means Clustering: Step 1

Algorithm: k-means, Distance Metric: Euclidean Distance

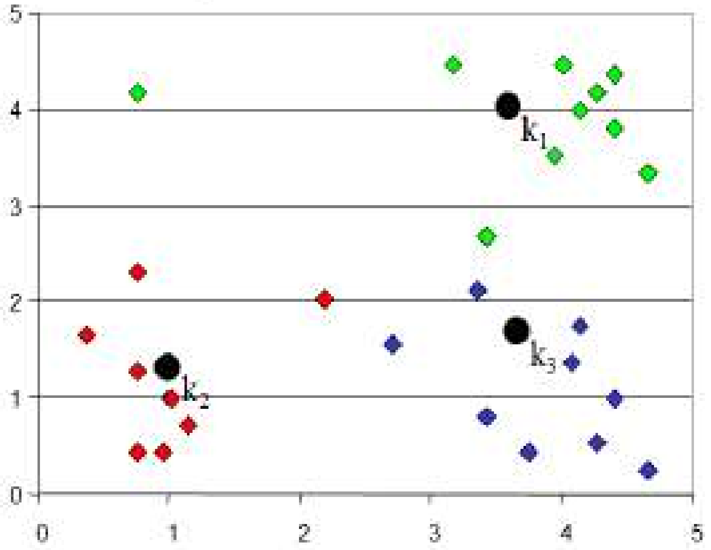
K-means Clustering: Step 2

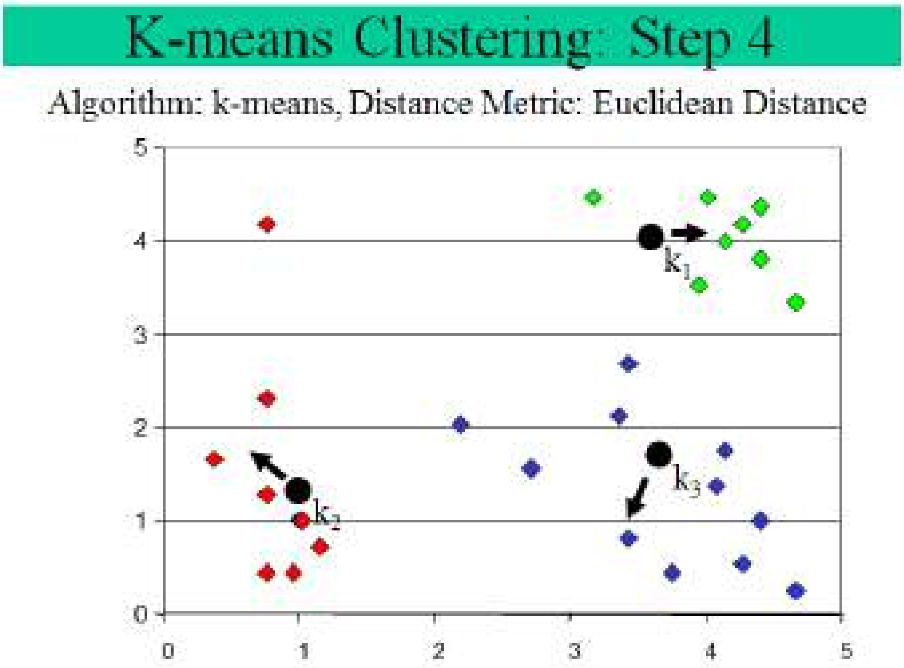
Algorithm: k-means, Distance Metric: Euclidean Distance



K-means Clustering: Step 3

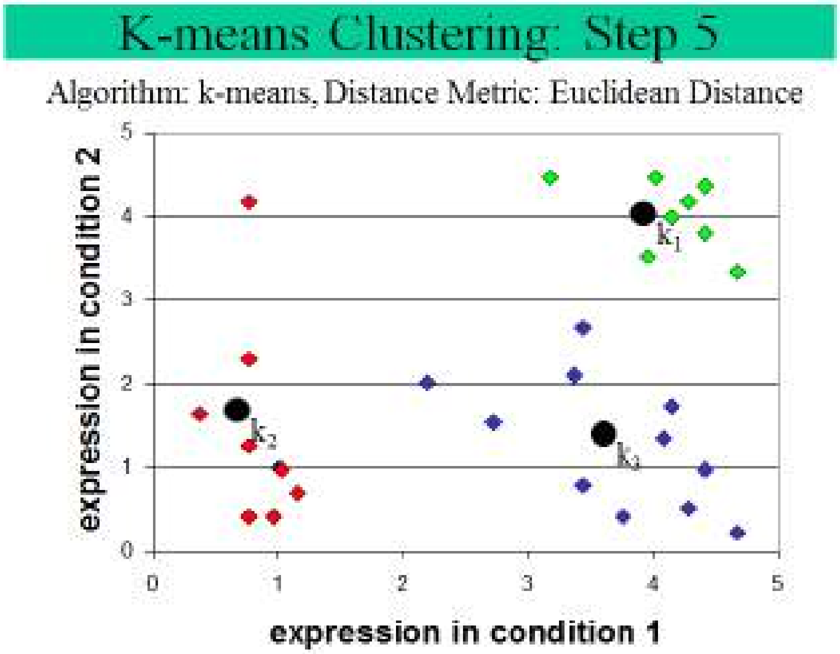
Algorithm: k-means, Distance Metric: Euclidean Distance





K-means Clustering: Step 4

Algorithm: k-means, Distance Metric: Euclidean Distance



K-means Clustering: Step 5

Algorithm: k-means, Distance Metric: Euclidean Distance

Comments on the *K-Means* Method

- **<u>Strength:</u>** *Efficient*: Complexity is *O*(*tkn*), where *n* is # objects, *k* is # clusters, and *t* is # iterations. Normally, *k*, *t* << *n*.

  - Comparing: PAM: O(k(n-k)^2), CLARA: O(ks^2 + k(n-k))
    Complexity of K-Means Algorithms = *O*(*tkn*)
    Complexity of K-Medoids/PAM Algorithms = O(k(n-k)^2)
    Complexity of CLARA Algorithms = O(ks^2 + k(n-k)

- Comment: Often terminates at a *local optimal*.

Weakness

- Applicable only to objects in a continuous n-dimensional space
  - Using the k-modes method for categorical data
  - In comparison, k-medoids can be applied to a wide range of data
- Need to specify *k*, the *number* of clusters, in advance (there are ways to automatically determine the best k
- Sensitive to noisy data and *outliers*
- Not suitable to discover clusters with *non-convex shapes*

What Is the Problem of the K-Means Method?

The k-means algorithm is sensitive to outliers !

- Since an object with an extremely large value may substantially distort the distribution of the data
- K-Medoids: Instead of taking the **mean** value of the object in a cluster as a reference point, **medoids** can be used, which is the **most centrally located** object in a cluster

k-medoids or PAM (Partition around medoids) Algorithm

k-medoids or PAM (Partition around medoids) (Kaufman & Rousseeuw'87): Each cluster is represented by one of the objects in the cluster

In the PAM (Partitioning Around Mediods) alogorithm (also known as K-mediods algorithm),each cluster is represented by a mediod (one of the objects located near the centre of the cluster). This algorithm initially selects K objects arbitrarily from the input data set as mediods and each K object is considered as representative of K classes. The other objects in the database which are not currently mediods are classified on the basis of their distances to these K-mediods. That is ,the algorithm determines whether there is an object that should replace one of the existing mediods. For doing this, it makes use of **two steps**, namely, ***build phase*** and ***swap phase***.

The ***build phase*** works by looking at all pairs of mediods and non-mediod objects and then selecting the pair that best improves the overall quality of the clustering. Suppose there is a cluster k_i represented by mediod t_i . Now, in the build phase the algorithm needs to determine whether the current mediod t_i should be exchanged with a non-mediod object t_h .

In the ***swap phase***, the mediod t_i and non-mediod object t_h are swapped only if the overall impact to the cost shows an improvement . **The cost or the quality is measured by the sum of all the distances from a non-mediod object to the mediod for the cluster it is in** .To calculate the effect of such a swap, a cost C_{jih} is calculated . It is the cost change for an item t_j associated with swapping mediod t_i with non-mediod t_h . This cost is the change to the sum of all the distances from objects to their cluster mediods . While calculating the cost , following four cases must be examined :

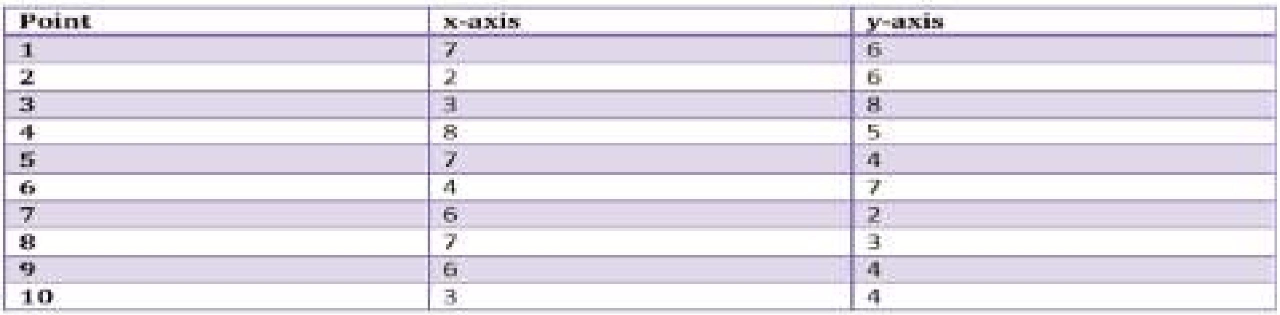
k-medoids or PAM (Partition around medoids) Algorithm is as follows:

```
Input:
        D = {t1, t2, …, tn}   // set of elements
        A                     // Adjacency matrix showing distance between elements
        K                     // Number of desired clusters
Output:
        K = {k1,k2,… kn}      // set of clusters
Procedure :
        Randomly select k mediods from D;
        Repeat
                for each non-mediod $t_h$ do
                        for each mediod $t_i$ do

                            calculate $TC_{ih}$;    // $\sum_{j=1}^{n} Cjih$

                                determine i, h where $TC_{ih}$ is the smallest;
                            if $TC_{ih} < 0$,  then
                                  swap mediod $t_i$ and $t_h$ ;
         until $TC_{ih} \geq 0$;

         for each $t_i \in D$ do

              assign $t_i$ to $k_j$ , where distance ($t_i$ ,$t_j$ ) is the minimum over all mediods;

End
```

Complexity of K-Medoids/PAM Algorithms = $O(k(n-k)^2)$

Example:

For a given **k=2, cluster** the following data set using PAM.



Point	x-axis	y-axis
1	7	6
2	2	6
3	3	8
4	8	5
5	7	4
6	4	7
7	6	2
8	7	3
9	6	4
10	3	4

Let us choose that (3, 4) and (7, 4) are the medoids. Suppose considering the Manhattan distance metric as the distance measure,

So, now if we calculate the distance from each point:

For (7, 6), Calculating the distance from the medoids chosen, this point is nearest to (7, 4)

For (2, 6) , Calculating the distance from the medoids chosen, this point is nearest to (3, 4)

For (3, 8) , Calculating the distance from the medoids chosen, this point is at same distance from both the points. So choosing that it is nearest to (3, 4)

For (8, 5) , Calculating the distance from the medoids chosen, this point is nearest to (7, 4)

For (4, 7) , Calculating the distance from the medoids chosen, this point is nearest to (3, 4)

For (6, 2) , Calculating the distance from the medoids chosen, this point is nearest to (7, 4)

For (7, 3) , Calculating the distance from the medoids chosen, this point is nearest to (7, 4)

For (6, 4) , Calculating the distance from the medoids chosen, this point is nearest to (7, 4)

So, now after the clustering, the clusters formed are: **{(3,4), (2,6), (3,8), (4,7)} and{(7,4), (6,2), (6,4), (7,3), (8,5), (7,6)}**. Now calculating the cost which is nothing but the sum of distance of each non-selected point from the selected point which is medoid of the cluster it belongs to.

Total Cost = cost((3, 4), (2, 6)) + cost((3, 4), (3, 8)) + cost((3, 4), (4, 7)) + cost((7, 4), (6, 2))+ cost((7, 4), (6, 4))+ cost((7, 4), (7, 3))+ cost((7, 4), (8, 5))+ cost((7, 4), (7, 6))

= 3 + 4 + 4 + 3 + 1 + 1 + 2 + 2

= 20.

So, now let us choose some other point to be a medoid instead of (7, 4). Let us randomly choose (7, 3). Not the new medoid set is: (3, 4) and (7, 3). Now repeating the same task as earlier:

So, now if we calculate the distance from each point:

For (7, 6), Calculating the distance from the medoids chosen, this point is nearest to (7, 3)

For (2, 6) , Calculating the distance from the medoids chosen, this point is nearest to (3, 4)

For (3, 8) , Calculating the distance from the medoids chosen, this point is nearest to (3, 4)

For (8, 5) , Calculating the distance from the medoids chosen, this point is nearest to (7, 3)

For (4, 7) , Calculating the distance from the medoids chosen, this point is nearest to (3, 4)

For (6, 2) , Calculating the distance from the medoids chosen, this point is nearest to (7, 3)

For (7, 4) , Calculating the distance from the medoids chosen, this point is nearest to (7, 3)

For (6, 4) , Calculating the distance from the medoids chosen, this point is nearest to (7, 3)

Calculating the total cost = cost((3, 4), (2, 6)) + cost((3, 4), (3, 8)) + cost((3, 4), (4, 7)) + cost((7, 3), (7, 6)) + cost((7, 3), (8, 5)) + cost((7, 3), (6, 2)) + cost((7, 3), (7, 4)) + cost((7, 3), (6, 4))

= 3 + 4 + 4 + 3 + 3 + 2 + 1 + 2

= 22.

The total cost when (7, 3) is the medoid > the total cost when (7, 4) was the medoid earlier.

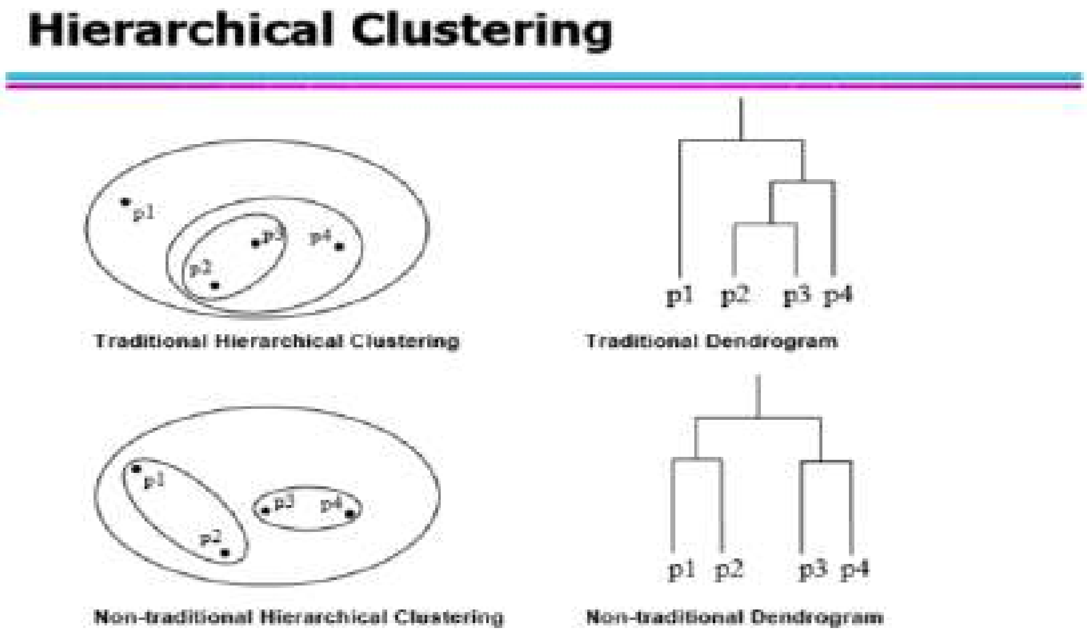
Hence, (7, 4) should be chosen instead of (7, 3) as the medoid. Since there is no change in the medoid set, the algorithm ends here.

Hence the clusters obtained finally are: {(3,4), (2,6), (3,8), (4,7)} and{(7,4), (6,2), (6,4), (7,3), (8,5), (7,6)}.

2. Hierarchical Methods

This method creates a hierarchical decomposition of the given set of data objects. We can classify hierarchical methods on the basis of how the hierarchical decomposition is formed. There are two approaches here −

- Agglomerative Approach
- Divisive Approach



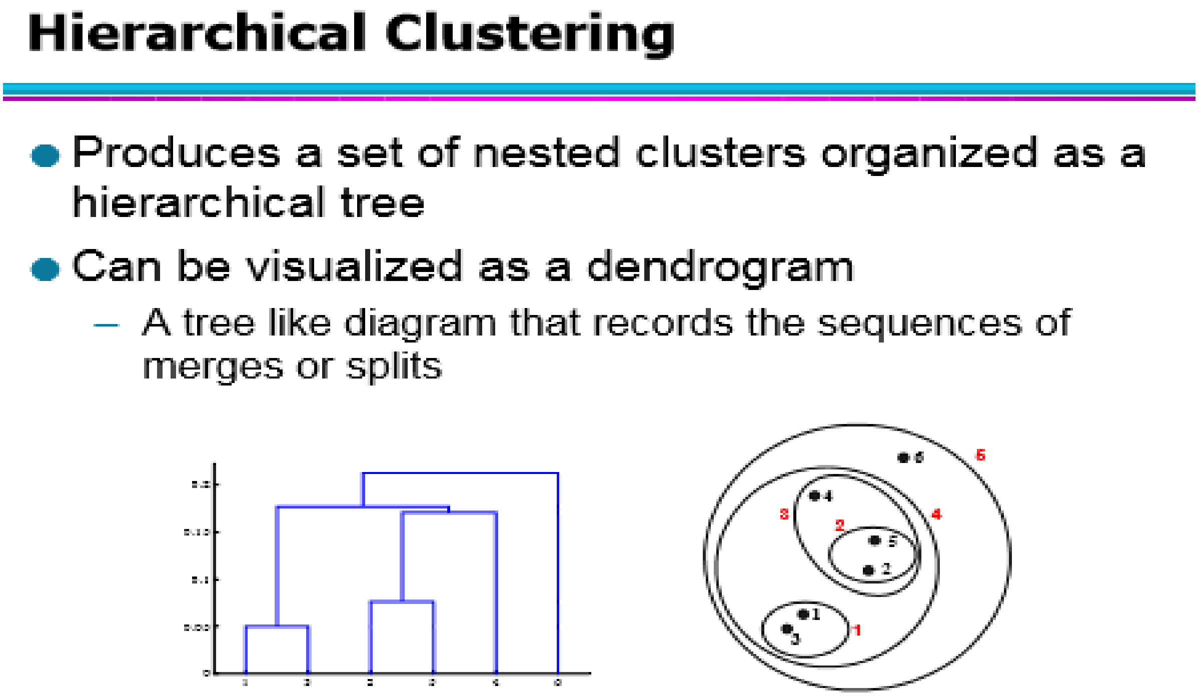
Hierarchical Clustering

Traditional Hierarchical Clustering

Traditional Dendrogram

Non-traditional Hierarchical Clustering

Non-traditional Dendrogram

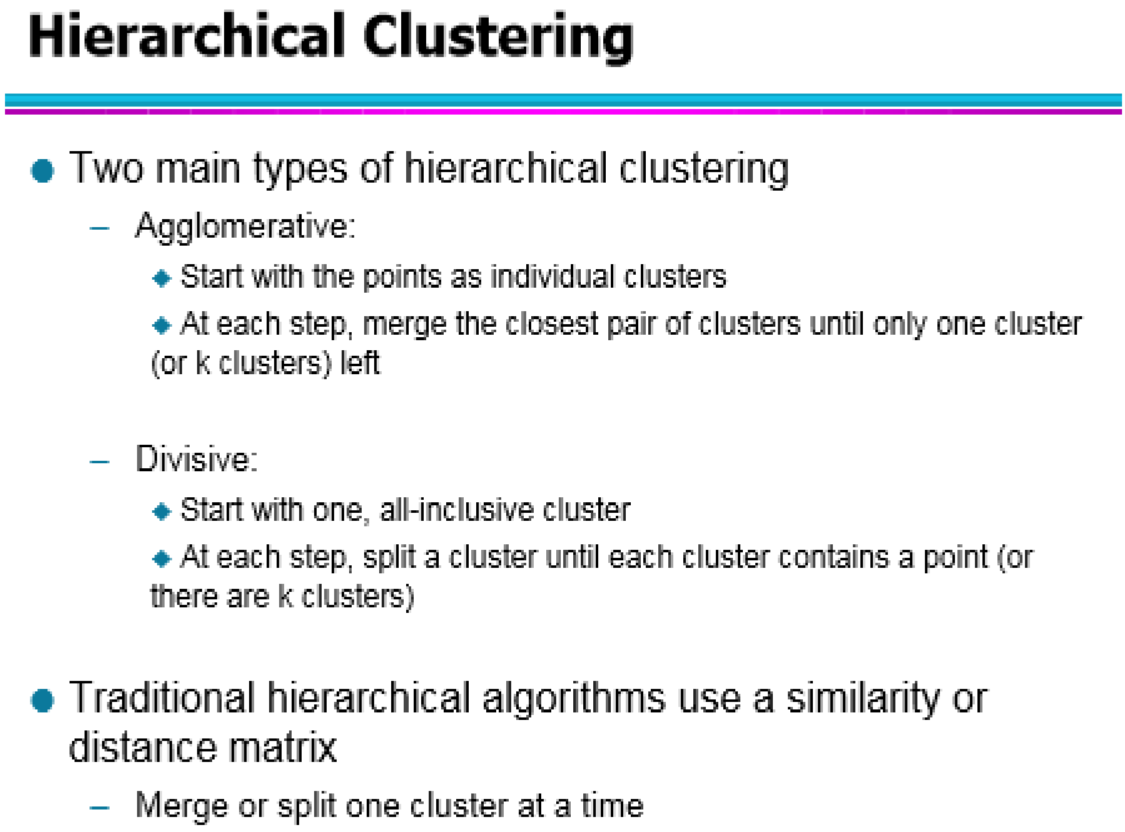


Hierarchical Clustering

- Produces a set of nested clusters organized as a hierarchical tree
- Can be visualized as a dendrogram
  - A tree like diagram that records the sequences of merges or splits

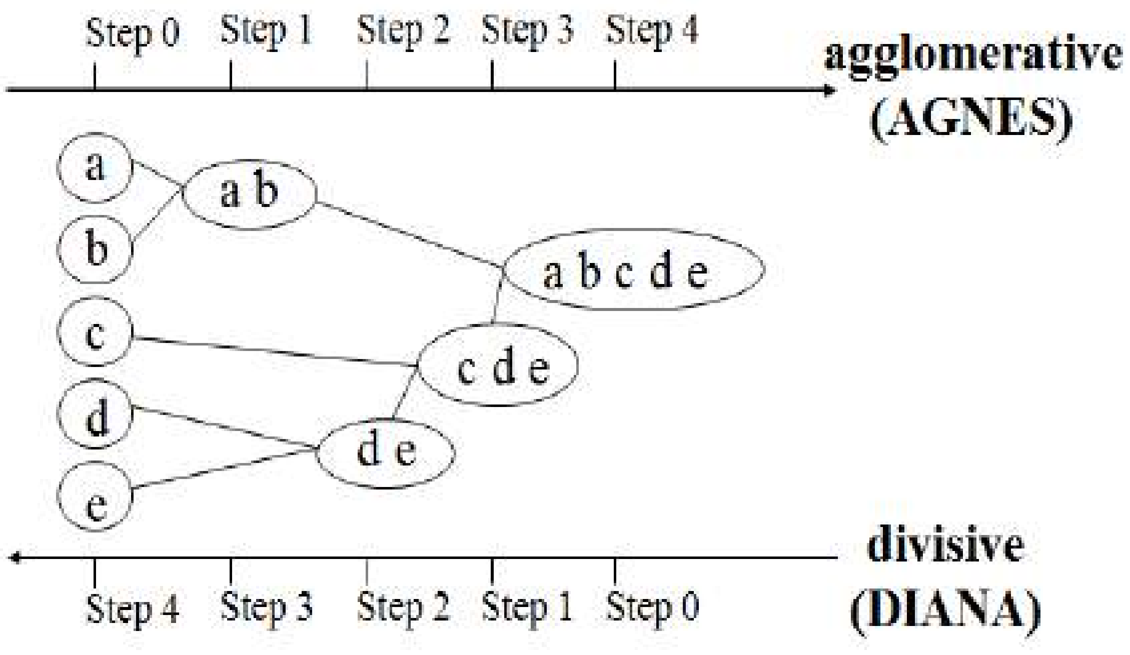
Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram

A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster



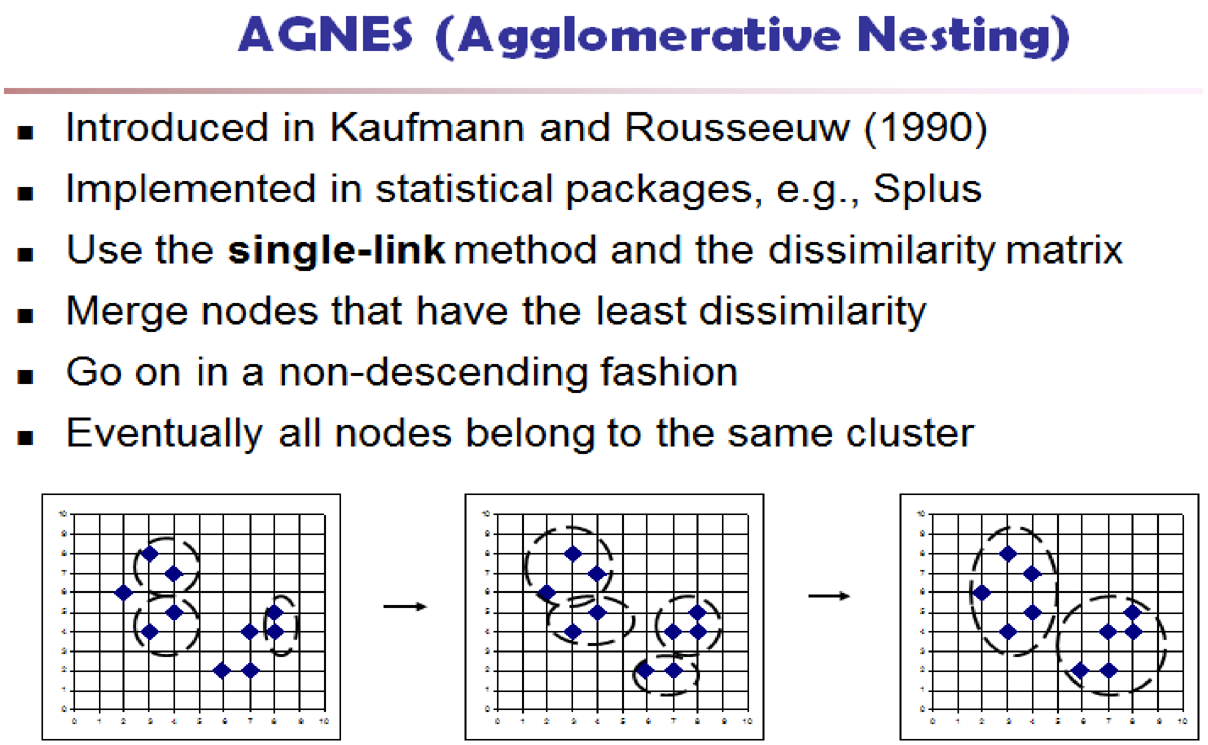
Hierarchical Clustering

- Two main types of hierarchical clustering
  - Agglomerative:
    - Start with the points as individual clusters
    - At each step, merge the closest pair of clusters until only one cluster (or k clusters) left
  - Divisive:
    - Start with one, all-inclusive cluster
    - At each step, split a cluster until each cluster contains a point (or there are k clusters)
- Traditional hierarchical algorithms use a similarity or distance matrix
  - Merge or split one cluster at a time



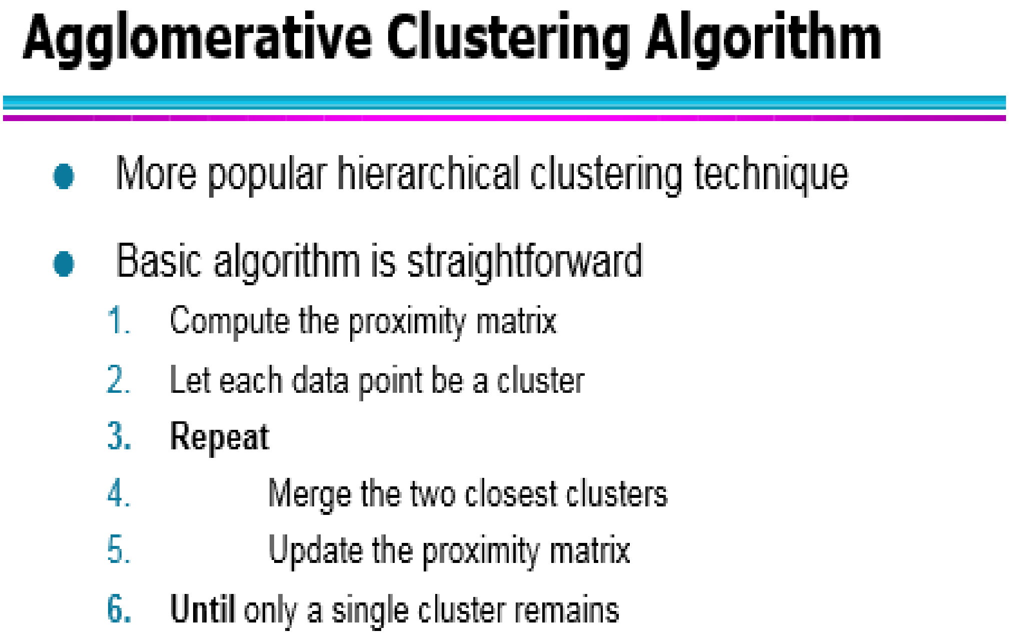
Agglomerative Approach(AGNES)

This approach is also known as the **bottom-up approach**. In this, we start with each object forming a separate group. It keeps on merging the objects or groups that are close to one another. It keep on doing so until all of the groups are merged into one or until the termination condition holds.



AGNES (Agglomerative Nesting)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical packages, e.g., Splus
- Use the **single-link** method and the dissimilarity matrix
- Merge nodes that have the least dissimilarity
- Go on in a non-descending fashion
- Eventually all nodes belong to the same cluster



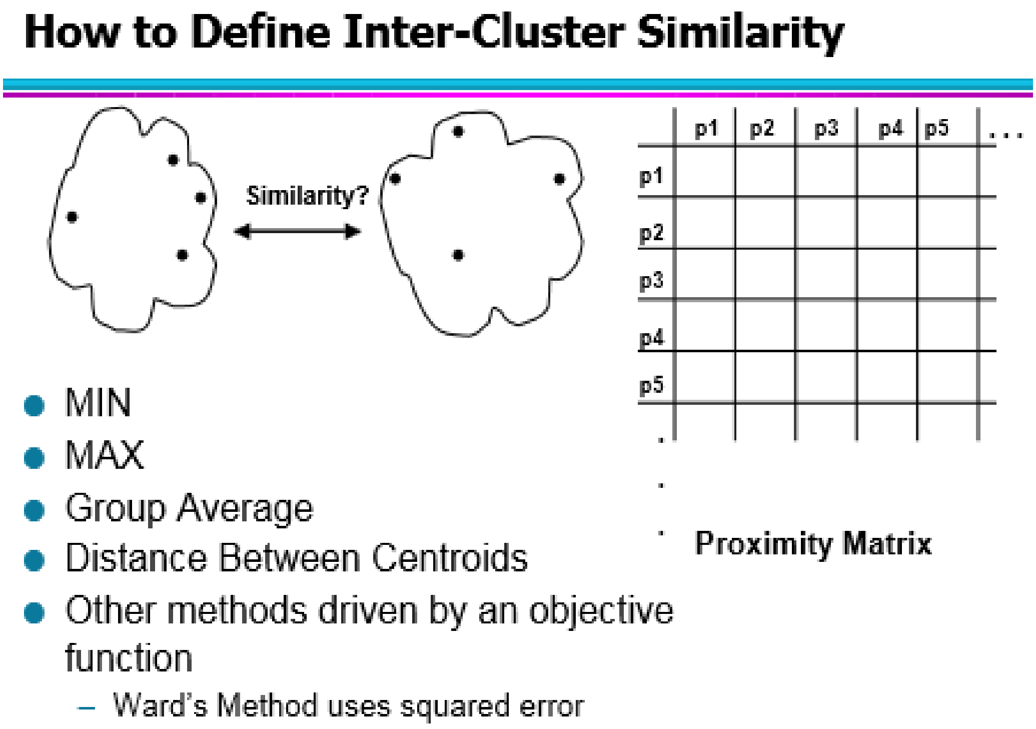
Agglomerative Clustering Algorithm

- More popular hierarchical clustering technique
- Basic algorithm is straightforward

```
1.  Compute the proximity matrix
2.  Let each data point be a cluster
3.  Repeat
4.          Merge the two closest clusters
5.          Update the proximity matrix
6.  Until only a single cluster remains
```

Distance between Clusters

- Key operation is the computation of the proximity of two clusters
  - Different approaches to defining the distance between clusters distinguish the different algorithms

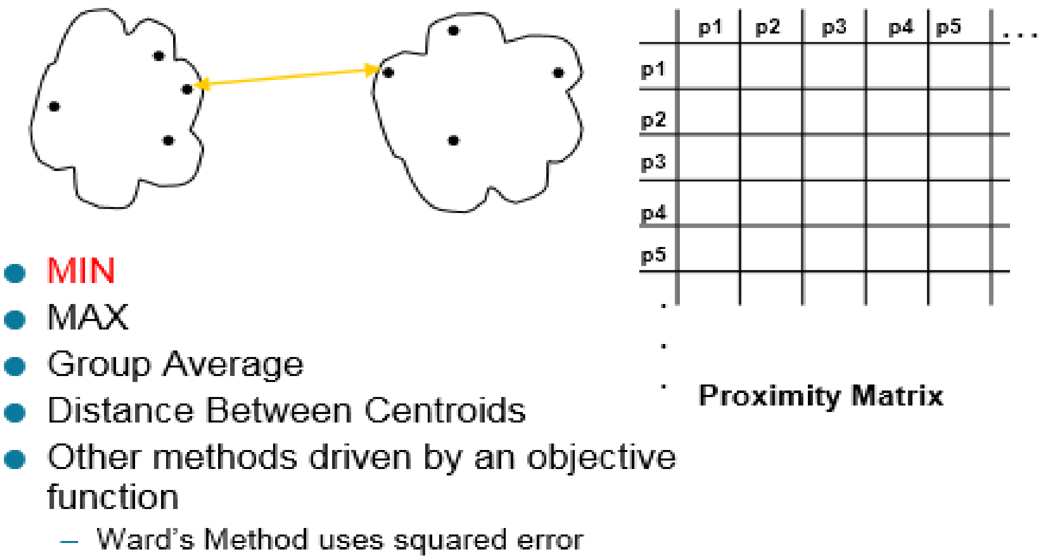


How to Define Inter-Cluster Similarity

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

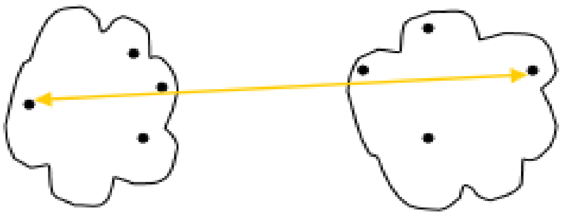
Proximity Matrix



- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

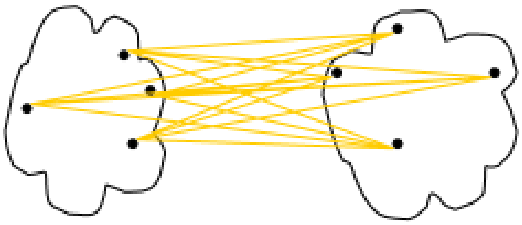
Proximity Matrix



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
. . .						

Proximity Matrix

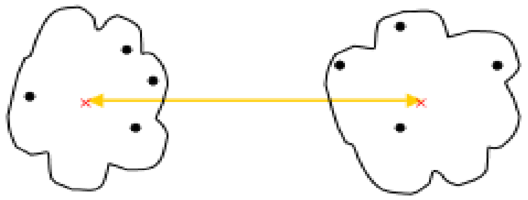
- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error



	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
. . .						

Proximity Matrix

- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error



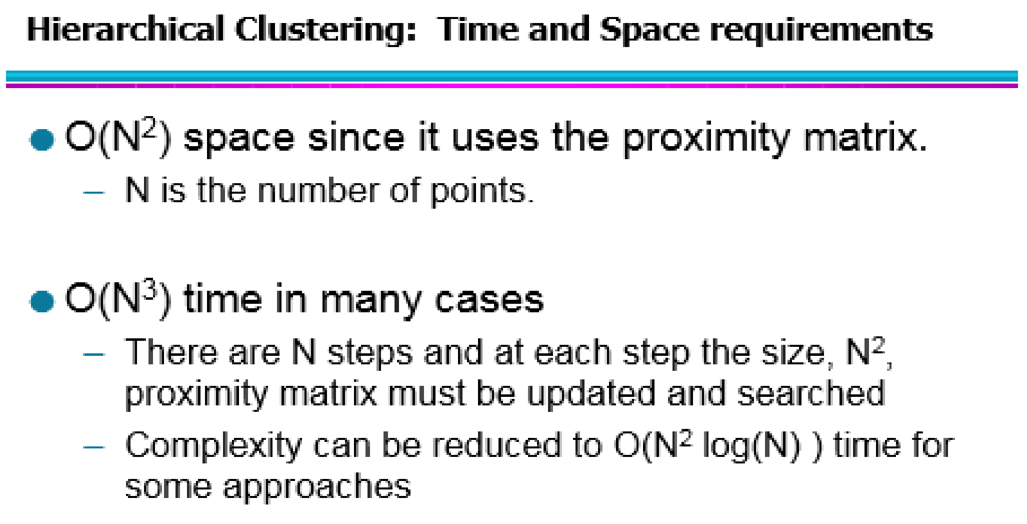
- MIN
- MAX
- Group Average
- Distance Between Centroids
- Other methods driven by an objective function
  - Ward's Method uses squared error

	p1	p2	p3	p4	p5	. . .
p1						
p2						
p3						
p4						
p5						
.						
.						
.						

Proximity Matrix

Distance between Clusters

- Single link: smallest distance between an element in one cluster and an element in the other, i.e., dist(K_i, K_j) = min(t_{ip}, t_{jq})
- Complete link: largest distance between an element in one cluster and an element in the other, i.e., dist(K_i, K_j) = max(t_{ip}, t_{jq})
- Average: avg distance between an element in one cluster and an element in the other, i.e., dist(K_i, K_j) = avg(t_{ip}, t_{jq})
- Centroid: distance between the centroids of two clusters, i.e., dist(K_i, K_j) = dist(C_i, C_j)
- Medoid: distance between the medoids of two clusters, i.e., dist(K_i, K_j) = dist(M_i, M_j)
  - Medoid: a chosen, centrally located object in the cluster

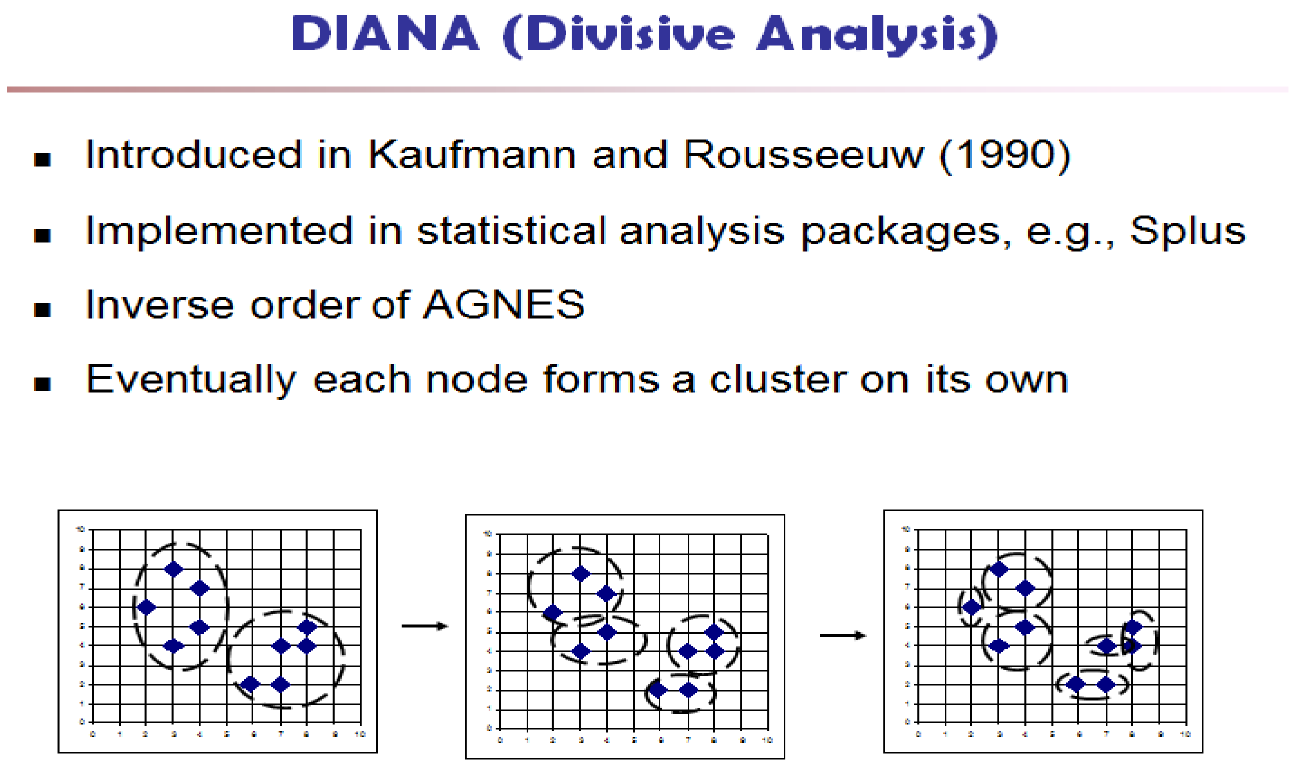


Hierarchical Clustering: Time and Space requirements

- $O(N^2)$ space since it uses the proximity matrix.
  - N is the number of points.
- $O(N^3)$ time in many cases
  - There are N steps and at each step the size, N^2, proximity matrix must be updated and searched
  - Complexity can be reduced to $O(N^2 \log(N))$ time for some approaches

Divisive Approach (DIANA)

This approach is also known as the **top-down approach**. In this, we start with all of the objects in the same cluster. In the continuous iteration, a cluster is split up into smaller clusters. It is down until each object in one cluster or the termination condition holds. This method is rigid, i.e., once a merging or splitting is done, it can never be undone.



DIANA (Divisive Analysis)

- Introduced in Kaufmann and Rousseeuw (1990)
- Implemented in statistical analysis packages, e.g., Splus
- Inverse order of AGNES
- Eventually each node forms a cluster on its own

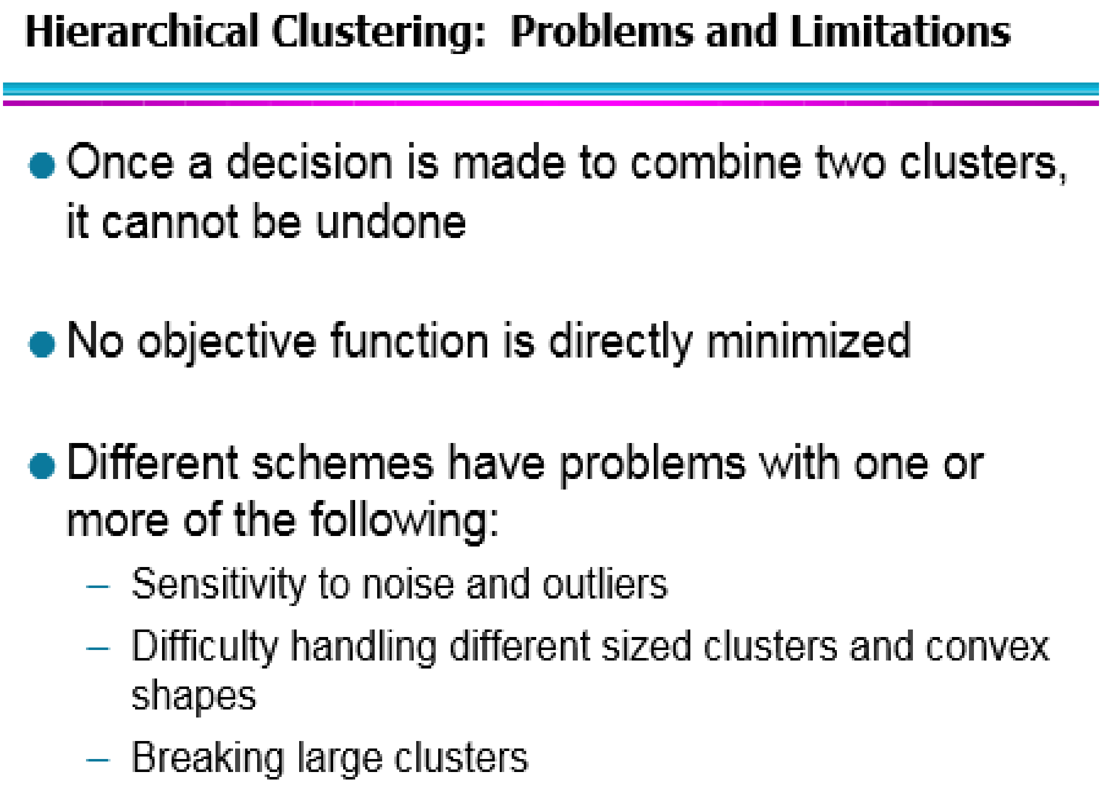
Divisive Clustering Algorithm

```
1.  Compute the proximity matrix
2.  Let all of the objects in the same cluster
3.  Repeat
4.        Split a cluster into smaller clusters
5.        Update the proximity matrix
6.   Until each object in one cluster.
```

Approaches to Improve Quality of Hierarchical Clustering

Here are the two approaches that are used to improve the quality of hierarchical clustering

- Perform careful analysis of object linkages at each hierarchical partitioning.
- Integrate hierarchical agglomeration by first using a hierarchical agglomerative algorithm to group objects into micro-clusters, and then performing macro-clustering on the micro-clusters.

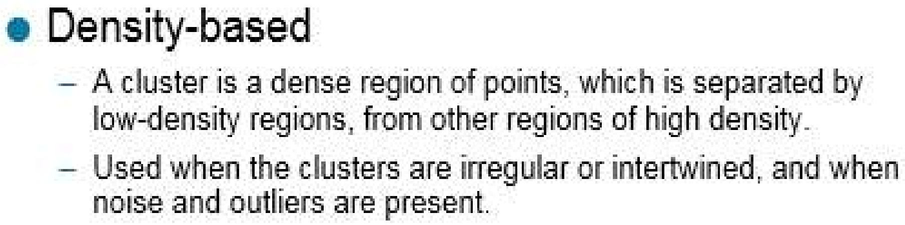


Hierarchical Clustering: Problems and Limitations

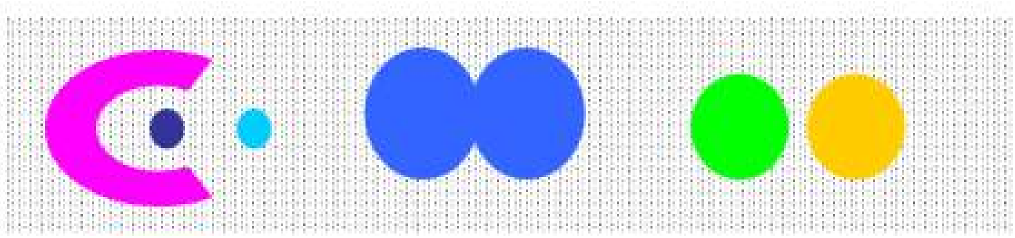
- Once a decision is made to combine two clusters, it cannot be undone
- No objective function is directly minimized
- Different schemes have problems with one or more of the following:
  - Sensitivity to noise and outliers
  - Difficulty handling different sized clusters and convex shapes
  - Breaking large clusters

3. Density-based Method

This method is based on the **notion of density** **(**connectivity and density functions**)**. The basic idea is to continue growing the given cluster as long as the density in the neighbourhood exceeds some threshold, i.e., for each data point within a given cluster, the radius of a given cluster has to contain at least a minimum number of points.



- Density-based
  - A cluster is a dense region of points, which is separated by low-density regions, from other regions of high density.
  - Used when the clusters are irregular or intertwined, and when noise and outliers are present.

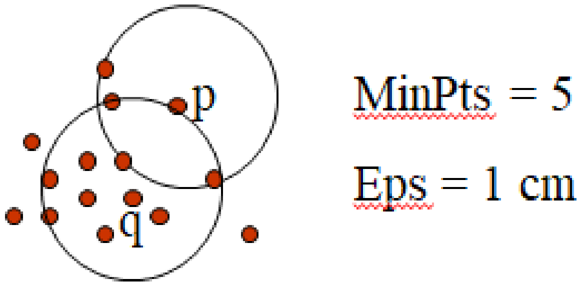


6 density-based clusters

- Clustering based on density (local cluster criterion), such as density-connected points or based on an explicitly constructed density function
- Major features:
  - Discover clusters of arbitrary shape
  - Handle noise
  - One scan
  - Need density parameters as termination condition
- Several interesting studies:
  - DBSCAN: Ester, et al. (KDD'96)
  - OPTICS: Ankerst, et al (SIGMOD'99).
  - DENCLUE: Hinneburg & D. Keim (KDD'98)
  - CLIQUE: Agrawal, et al. (SIGMOD'98) (more grid-based

Density-Based Clustering: Basic Concepts

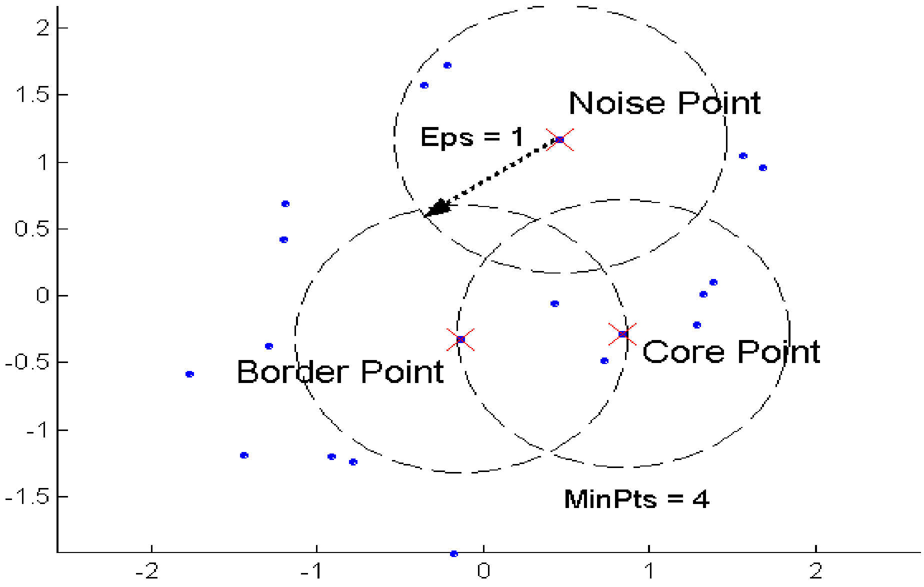
- Two parameters*:*
  - ***Eps*****:** Maximum radius of the neighbourhood
  - ***MinPts*****:** Minimum number of points in an Eps-neighbourhood of that point

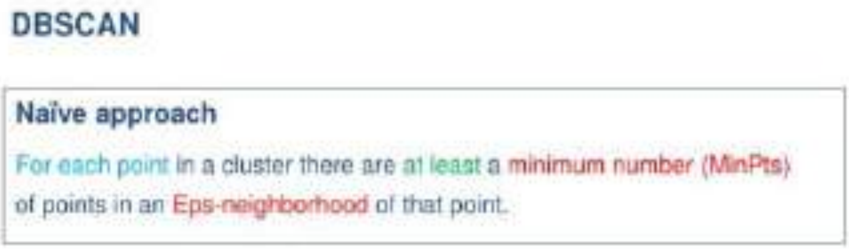


DBSCAN is a density-based algorithm

- Density = number of points within a specified radius r (Eps)
- A point is a core point if it has more than a specified number of points (MinPts) within Eps
    - These are points that are at the interior of a cluster
- A border point has fewer than MinPts within Eps, but is in the neighborhood of a core point
- A noise point is any point that is not a core point or a border point.

DBSCAN: Core, Border, and Noise Points

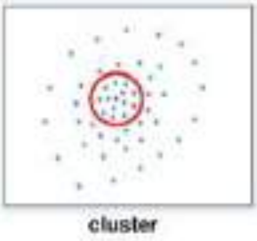




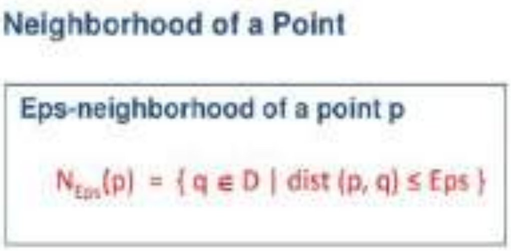
DBSCAN

Naïve approach

For each point in a cluster there are at least a minimum number (MinPts) of points in an Eps-neighborhood of that point.



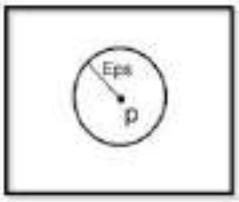
- $N_{Eps}(p)$: {q belongs to D | dist(p,q) ≤ Eps}



Neighborhood of a Point

Eps-neighborhood of a point p

$N_{Eps}(p) = \{ q \in D \mid dist\ (p, q) \leq Eps \}$



- Directly density-reachable: A point *p* is directly density-reachable from a point *q* w.r.t. *Eps*, *MinPts* if
  - *p* belongs to $N_{Eps}(q)$
  - core point condition:
    - $|N_{Eps}(q)| \geq MinPts$

Better idea

For every point p in a cluster C there is a point q ∊ C, so that

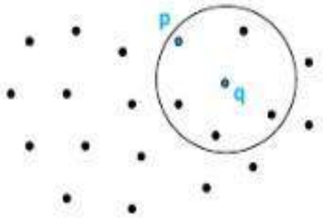
(1) p is inside of the Eps-neighborhood of q

and

(2) $N_{Eps}(q)$ contains at least MinPts points.

border points are connected to core points

core points = high density

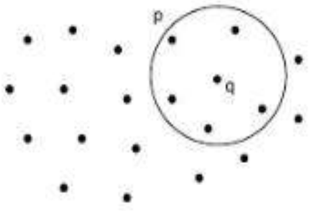


Definition

A point p is directly density-reachable from a point q with regard to the parameters Eps and MinPts, if

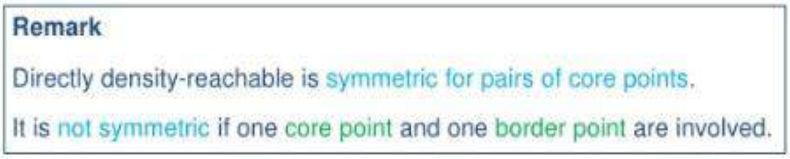
1) $p \in N_{Eps}(q)$ (reachability)

2) $| N_{Eps}(q) | \geq$ MinPts (core point condition)



MinPts = 5

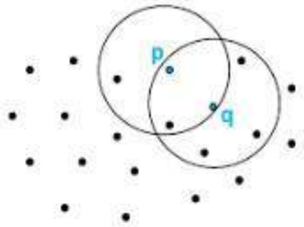
$| N_{Eps}(q) | = 6 \geq 5 =$ MinPts (core point condition)

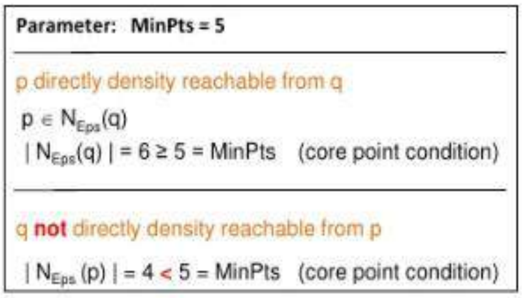


Remark

Directly density-reachable is symmetric for pairs of core points.

It is not symmetric if one core point and one border point are involved.





Parameter: MinPts = 5

p directly density reachable from q

$p \in N_{Eps}(q)$

$| N_{Eps}(q) | = 6 \geq 5 = MinPts$ (core point condition)

q **not** directly density reachable from p

$| N_{Eps}(p) | = 4 < 5 = MinPts$ (core point condition)

DBSCAN Algorithm

1. Arbitrary select a point *p*
2. Retrieve all points density-reachable from *p* w.r.t. *Eps* and *MinPts*
3. If *p* is a core point, a cluster is formed
4. If *p* is a border point, no points are density-reachable from *p* and DBSCAN visits the next point of the database
5. Continue the process until all of the points have been processed

(OR) Another way

1. Create a graph whose nodes are the points to be clustered
2. For each core-point C create an edge from C to every point p in the ε-neighborhood of C
3. Set N to the nodes of the graph;
4. If N does not contain any core points terminate
5. Pick a core point C in N
6. Let X be the set of nodes that can be reached from C by going forward;
    1. create a cluster containing X∪{C}
    2. N=N/(X∪{C})
7. Continue with step 4

Unit IV - Introduction to Business Intelligence

OLAP

(Online-Analytical Processing)

- It was introduced by E.F. Codd, who has been described as "the father of relational database model".
- OLAP (OnLine Analytical Processing) is computer processing that enables a user to ea sily and selectively extract and view data from different points of view.
- OLAP allows users to analyze database information from multiple database systems at one time.
- OLAP data is stored in multidimensional databases.
- OLAP is a category of software technology that it allows /enables analysts, managers, and executives to analyze the complex data derived from the data warehouse.
- The online indicates that the analysts, managers, and executives must be able to request new summaries and get the responses online, within a few seconds.
- Technology used to perform complex analysis of the data in a data warehouse
- Online Analytical Processing (OLAP) is based on the multidimensional data model.
- Nigel Pendse has suggested that an alternative and perhaps more descriptive term to describe the concep t of OLAP is **Fast Analysis of Shared Multidimensional Information (FASMI)**
- That is, most fundamentals definition of OLAP covering its all features is named as **FASMI.**

Fast Analysis of Shared Multidimensional Information (FASMI)

Fast : to achieve a high response speed, various techniques such as the use of pre-calculations specialized data storage should be considered.

Analysis: It lets users to enter query interactively for performing statistical analysis.

Shared : It allows multiple users to access the same data concurrently.

Multidimensional: It allows business users to have a multidimensional and logical view of the data in the data warehouse for making effective decision.

Information: It enables users to see the results in number of meaningful ways such as charts and graphs. It includes all the data and calculated information required by the users.

OLAP Operations

Since OLAP servers are based on multidimensional view of data, we will discuss OLAP operations in multidimensional data.

Here is the list of OLAP operations −

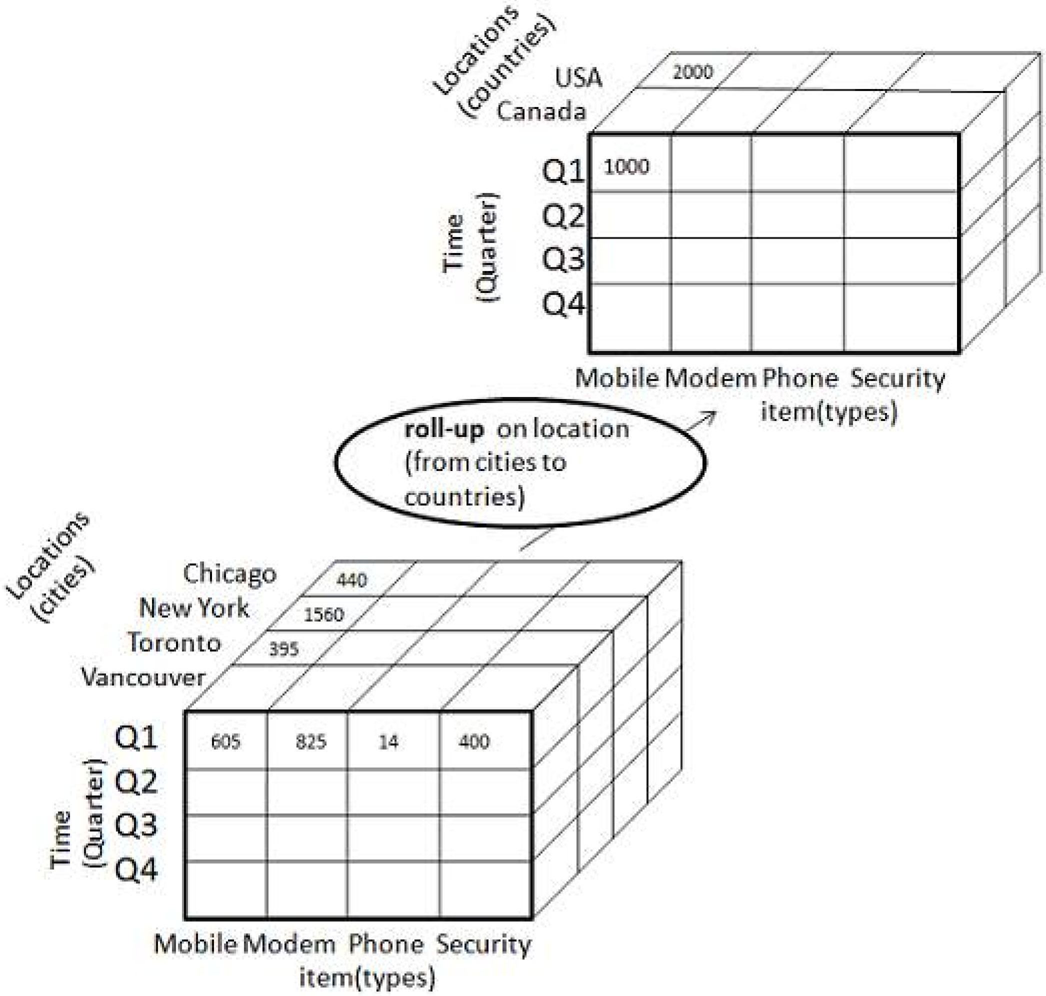
- Roll-up
- Drill-down
- Slice and dice
- Pivot (rotate)

Roll-up: summarize data

Roll-up performs aggregation on a data cube in any of the following ways −

- By climbing up a concept hierarchy for a dimension
- By dimension reduction

The following diagram illustrates how roll-up works.



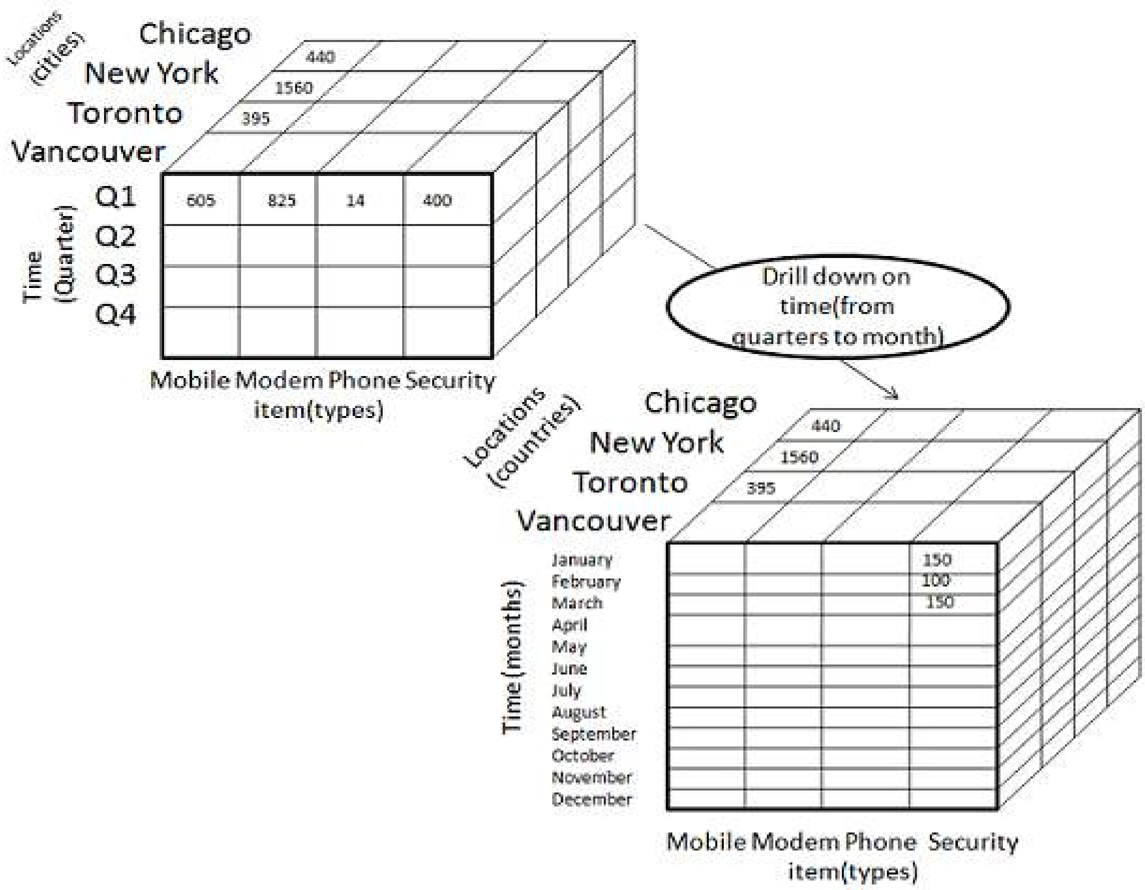
- Roll-up is performed by climbing up a concept hierarchy for the dimension location.
- Initially the concept hierarchy was "street < city < province < country".
- On rolling up, the data is aggregated by **ascending** the location hierarchy from the level of city to the level of country.
- The data is grouped into cities rather than countries.
- When roll-up is performed, one or **more dimensions from the data cube are removed.**

Drill-down : reverse of roll-up

Drill-down is the reverse operation of roll-up. It is performed by either of the following ways −

- By stepping down a concept hierarchy for a dimension (from higher level summary to lower level summary or detailed data)
- By introducing a new dimension.

The following diagram illustrates how drill-down works −

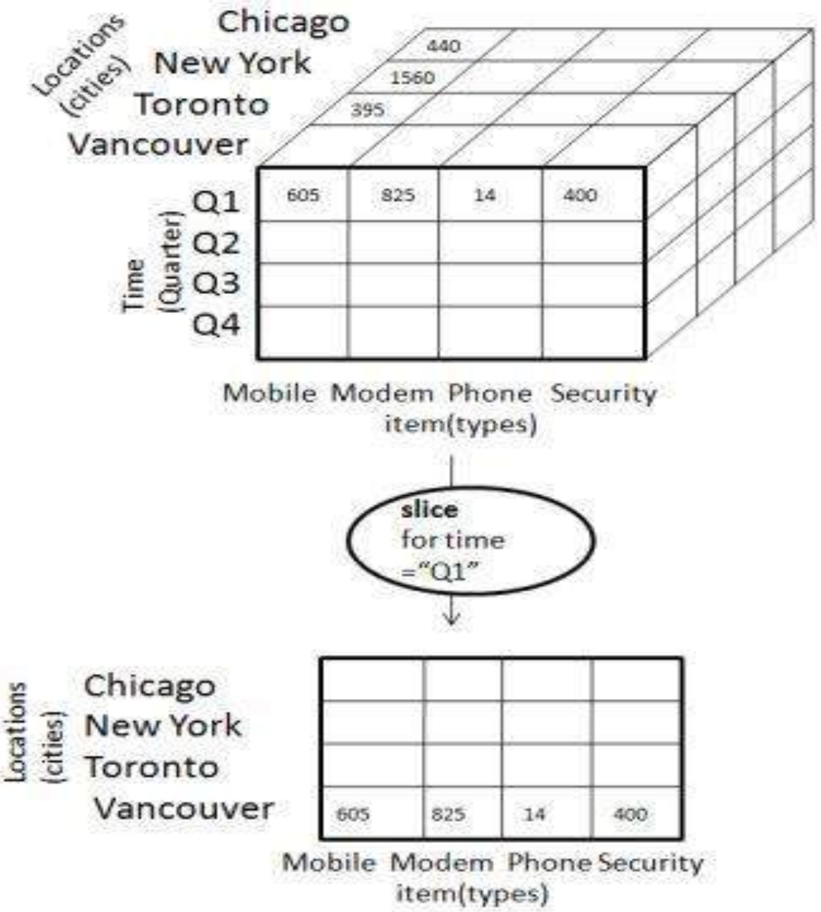


- Drill-down is performed by stepping down a concept hierarchy for the dimension time.

- Initially the concept hierarchy was "day < month < quarter < year."
- On drilling down, the time dimension is **descended** from the level of quarter to the level of month.
- When drill-down is performed, one or **more dimensions from the data cube are added.**
- It navigates the data from less detailed data to highly detailed data.

Slice

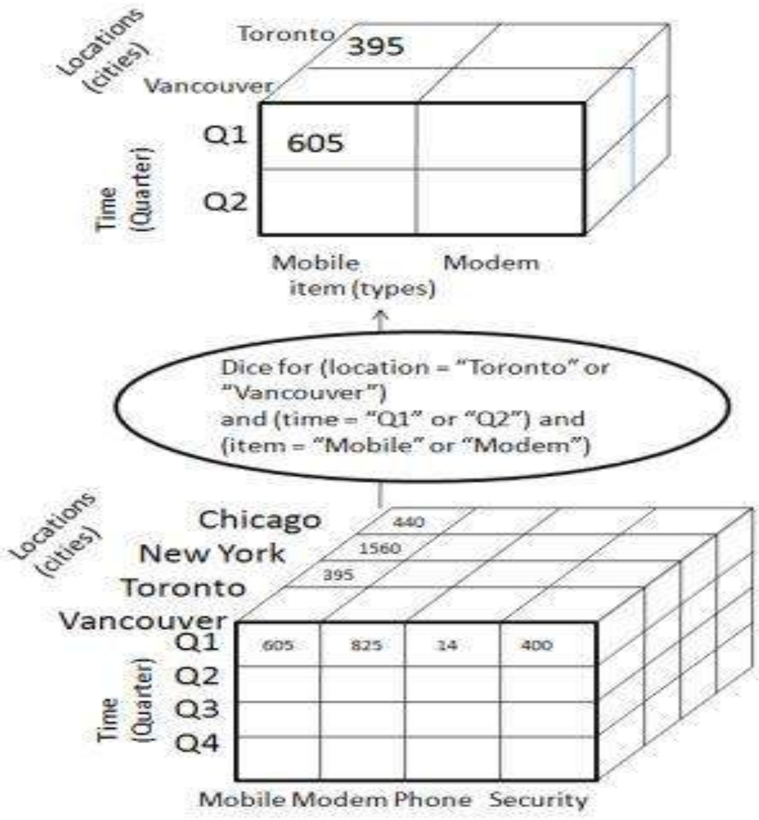
The slice operation selects one particular dimension from a given cube and provides a **new sub-cub**e. Consider the following diagram that shows how slice works.



- Here Slice is performed for the dimension "time" using the criterion time = "Q1".
- Slice will form a new sub-cube by selecting one dimension.

Dice

Dice selects two or more dimensions from a given cube and provides a new sub-cube. Consider the following diagram that shows the dice operation.

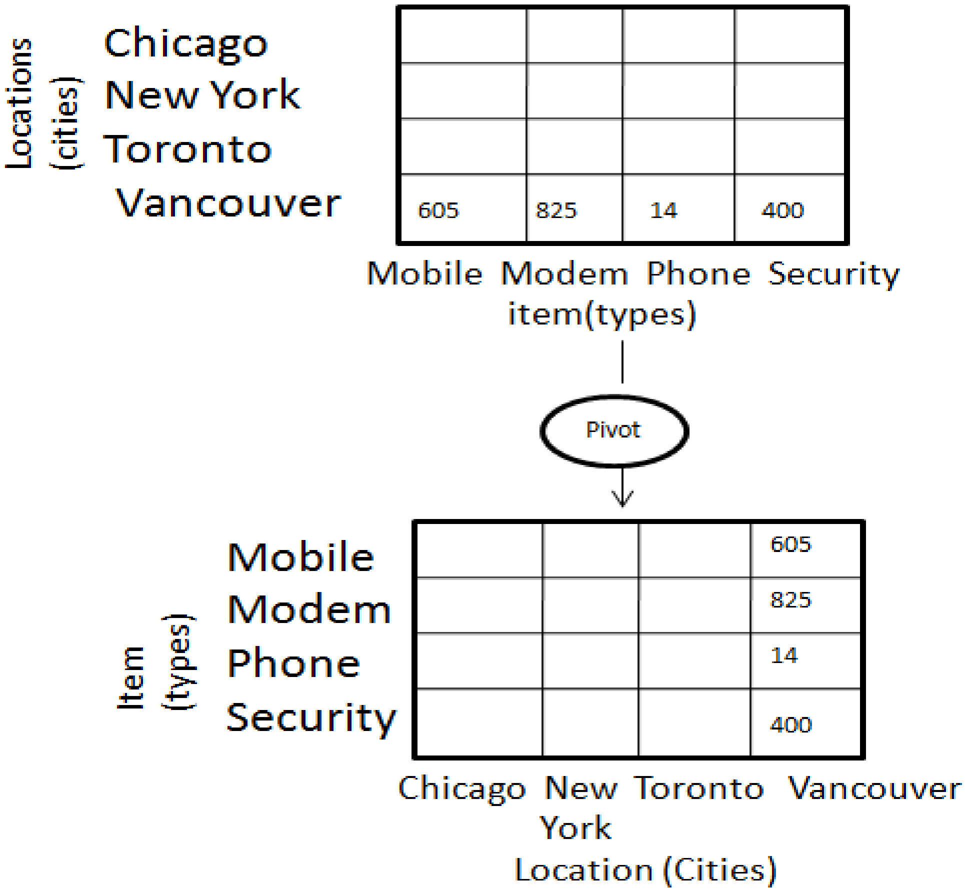


The dice operation on the cube based on the following selection criteria involves three dimensions.

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

Pivot (rotate):

The pivot operation is also known as rotation. It rotates the data axes in view in order to provide an alternative presentation of data. Consider the following diagram that shows the pivot operation.



Types of OLAP Servers

We have four types of OLAP servers –

- Relational OLAP (ROLAP)
- Multidimensional OLAP (MOLAP)
- Hybrid OLAP (HOLAP)
- Specialized SQL Servers

Relational OLAP

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

ROLAP includes the following −

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

Multidimensional OLAP

MOLAP uses array-based multidimensional storage engines (sparse matrix techniques) for multidimensional views of data. With multidimensional data stores, the storage utilization may be low if the data set is sparse. Therefore, many MOLAP server use two levels of data storage representation to handle dense and sparse data sets.

Hybrid OLAP

Hybrid OLAP is a combination of both ROLAP and MOLAP. It offers higher scalability of ROLAP and faster computation of MOLAP. HOLAP servers allows to store the large data volumes of detailed information. The aggregations are stored separately in MOLAP store.

Specialized SQL Servers

Specialized SQL servers provide advanced query language and query processing support for SQL queries over star and snowflake schemas in a read-only environment.

OLTP
(Online-Transaction Processing)

This technology used to perform updates on operational or transactional systems (e.g., point of sale systems)

(OLTP vs OLAP) AT A GLANCE…

OLTP	OLAP
Short Transaction both query and updates (e.g., update account balance, enroll is courses)	Long transactions, usually Complex queries. (e.g., all statistics about sales, grouped by department and month).
Queries are Simple (e.g., find account balance, find grade in courses)	“Data mining” operations
Updates are frequent (e.g., Concert tickets, seat reservations, shopping carts)	Infrequent Updates.

OLTP vs OLAP

Operational Database (OLTP)	**Data Warehouse (OLAP)**
Involves day-to-day processing.	Involves historical processing of information.
OLTP systems are used by clerks, DBAs, or database professionals.	OLAP systems are used by knowledge workers such as executives, managers and analysts.
Useful in running the business.	Useful in analyzing the business.
It focuses on Data in.	It focuses on Information out.
Based on Entity Relationship Model.	Based on Star Schema, Snowflake, Schema and Fact Constellation Schema.
Contains current data.	Contains historical data.
Provides primitive and highly detailed data.	Provides summarized and consolidated data.
Provides detailed and flat relational view of data.	Provides summarized and multidimensional view of data.
Number of users is in thousands.	Number or users is in hundreds.
Number of records accessed is in tens.	Number of records accessed is in millions.
Database size is from 100 MB to 1 GB.	Database size is from 100 GB to 1 TB
Provides high performance.	Highly flexible.

DIFFERENCE BETWEEN OLTP & OLAP

Item	**OLTP**	**OLAP**
User	Clerk, IT Professional	Knowledge worker
Functional	Daily task (day to day operation)	Decision Making / support
DB Design	Application oriented	subject oriented
Data	Current, up to date, detail, relational	Historical, multidimensional, integrated, summarised, consolidated
Access	Read/write	Read only

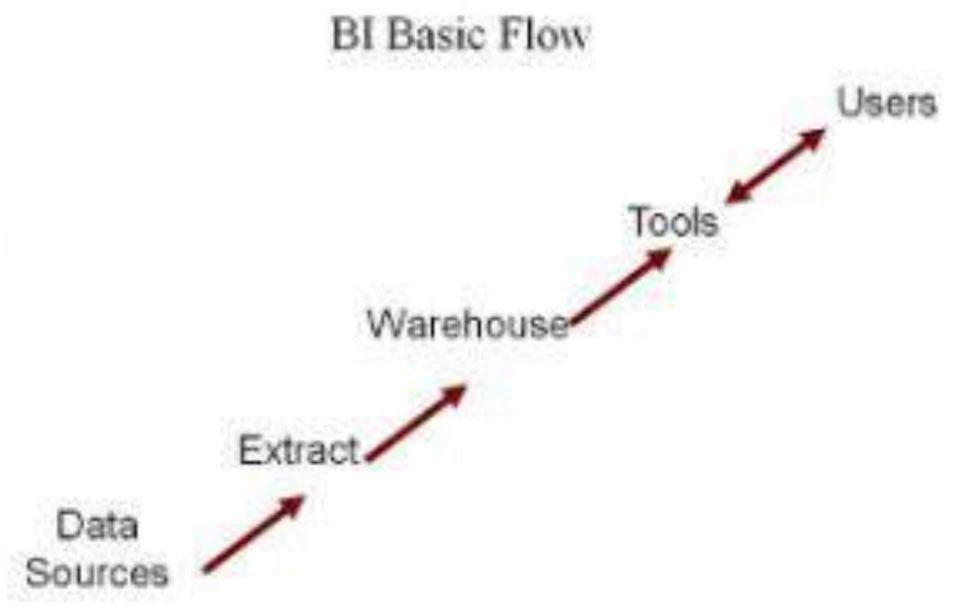
DB Size	100 MB-GB	100 GB-TB
Unit of work	Short, simple transaction	Complex query
#Record accessed	Tens	millions
usage	Structured, repetitive	Ad hoc

Business Intelligence(BI)

Business Intelligence Definitions and Concepts

The term **Business Intelligence (BI)** refers to technologies, applications and practices for the collection, integration, analysis, and presentation of business information. The purpose of Business Intelligence is to support better business decision making. Essentially, Business Intelligence systems are data-driven Decision Support Systems (DSS). Business Intelligence is sometimes used interchangeably with briefing books, report and query tools and executive information systems.

Business Intelligence (BI) is a computer based technique to identified, extracting and analyzing business data. For example senior management of an industry can inspect sales revenue by products and/or departments, or by associated costs and incomes. BI technologies provide historical, current and predictive views of business operations. So, management can take some strategic or operation decision easily.



Business intelligence (BI) is an umbrella term that includes a variety of IT applications that are used to analyze an organization's data and communicate the information to relevant users. Its major components are data warehousing, data mining, querying, and reporting (Figure 2.1).

Why BI?

BI is used for reporting, online analytical processing, data mining, process mining, complex event processing, business performance management, benchmarking, text mining and productive analysis. By using BI, management can monitor objectives from high level , understand what is happening, why is happening and can take necessary steps why the objectives are not full filled. Business intelligence aims to support better business decision-making. Thus a BI system can be called a decisions support system (DSS).

Business Intelligence can be applied to the following business purposes (MARCKM), in order to drive business value:

MARCKM means – Measurement, Analytics, Reporting/Enterprise Reporting, Collaboration/Collaboration Platform, and Knowledge Management.

The nature of life and businesses is to grow. Information is the lifeblood of business. Businesses use many techniques for understanding their environment and predicting the future for their own benefit and growth. Decisions are made from facts and feelings. Data-based decisions are more effective than those based on feelings alone. Actions based on accurate data, information, knowledge, experimentation, and testing, using fresh insights, can more likely succeed and lead to sustained growth.

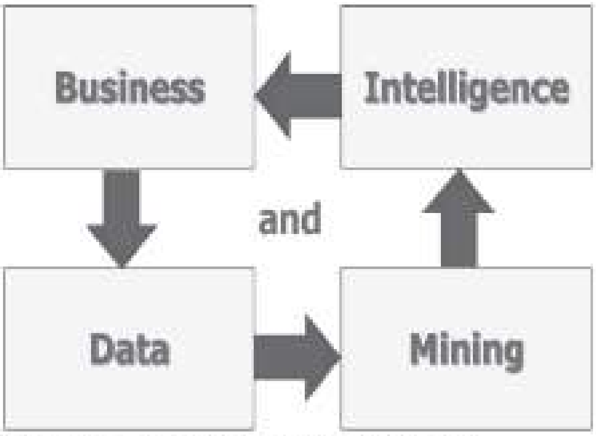


Figure 2.1 Business intelligence and data mining cycle

BI Tools

Many BI tools are available now. Most of the organization follows-

- Spreadsheet
- Reporting and querying software: tools that extract, sort, summarize, and present selected data
- Dashboards
- Data mining
- Data warehousing
- Decision Engineering
- Process Mining
- Business Performance management
- Local Information System

Microsoft introduced a new BI tool name-Dashboard. Dashboard is a visual display of the most important information needed to achieve one or more objectives which fits in a single computer screen so it can be monitored at a glance – Stephen Few, Information Dashboard design.

BI includes a variety of software tools and techniques to provide the managers with the information and insights needed to run the business. Information can be provided about the current state of affairs with the capability to drill down into details, and also insights about emerging patterns which lead to projections into the future. BI tools include data warehousing, online analytical processing, social media analytics, reporting, dashboards, querying, and data mining.

BI tools can range from very simple tools that could be considered end-user tools, to very sophisticated tools that offer a very broad and complex set of functionality. Thus, Even executives can be their own BI experts, or they can rely on BI specialists to set up the BI mechanisms for them. Thus, large organizations invest in expensive sophisticated BI solutions that provide good information in real time.

A spreadsheet tool, such as Microsoft Excel, can act as an easy but effective BI tool by itself. Data can be downloaded and stored in the Business Intelligence Concepts and Applications 25 spreadsheet, then analyzed to produce insights, then presented in the form of graphs and tables. This system offers limited automation using macros and other features. The analytical features include basic statistical and financial functions. Pivot tables help do sophisticated what-if analysis. Add-on modules can be installed to enable moderately sophisticated statistical analysis.

A dashboarding system, such as Tableau, can offer a sophisticated set of tools for gathering, analyzing, and presenting data. At the user end, modular dashboards can be designed and redesigned easily with a graphical user interface. The back-end data analytical capabilities include many statistical functions. The dashboards are linked to data warehouses at the back end to ensure that the tables and graphs and other elements of the dashboard are updated in real time (Figure 2.2).

Data mining systems, such as IBM SPSS Modeler, are industrial strength systems that provide capabilities to apply a wide range of analytical models on large data sets. Open source systems, such as Weka, are popular platforms designed to help mine large amounts of data to discover patterns.

Figure 2.2 Sample executive dashboard

BI Applications

BI applications can be divided into:

- **Technology solutions**
  - – DSS
  - – EIS
  - – OLAP
  - – Managed Query and Reporting
  - – Data Mining

- **Business Solutions**
  - – Performance Analysis
  - – Customer Analysis
  - – Market Place Analysis
  - – Productivity Analysis
  - – Sales Channel Analysis
  - – Behavioral Analysis
  - – Supply Chain Analysis

BI tools are required in almost all industries and functions. The nature of the information and the speed of action may be different across businesses, but every manager today needs access to BI tools to have up-todate metrics about business performance. Businesses need to embed new insights into their operating processes to ensure that their activities continue to evolve with more efficient practices. The following are some areas of applications of BI and data mining

Customer Relationship Management

A business exists to serve a customer. A happy customer becomes a repeat customer. A business should understand the needs and sentiments of the customer, sell more of its offerings to the existing customers, and also, expand the pool of customers it serves. BI applications can impact many aspects of marketing.

1. Maximize the return on marketing campaigns: Understanding the customer's pain points from data-based analysis can ensure that the marketing messages are fine-tuned to better resonate with customers.

2. Improve customer retention (churn analysis): It is more difficult and expensive to win new customers than it is to retain existing customers. Scoring each customer on their likelihood to quit can help the business design effective interventions, such as discounts or free services, to retain profitable customers in a cost-effective manner.

3. Maximize customer value (cross-selling, upselling): Every contact with the customer should be seen as an opportunity to gauge their current needs. Offering a customer new products and solutions based on those imputed needs can help increase revenue per customer. Even a customer complaint can be seen as an opportunity to wow the customer. Using the knowledge of the customer's history and value, the business can choose to sell a premium service to the customer.

4. Identify and delight highly valued customers: By segmenting the customers, the best customers can be identified. They can be proactively contacted, and delighted, with greater attention and better service. Loyalty programs can be managed more effectively.

5. Manage brand image: A business can create a listening post to listen to social media chatter about itself. It can then do sentiment analysis of the text to understand the nature of comments and respond appropriately to the prospects and customers.

Health Care and Wellness

Health care is one of the biggest sectors in advanced economies. Evidence-based medicine is the newest trend in data-based health care management. BI applications can help apply the most effective diagnoses and prescriptions for various ailments. They can also help manage public health issues, and reduce waste and fraud.

1. Diagnose disease in patients: Diagnosing the cause of a medical condition is the critical first step in a medical engagement. Accurately diagnosing cases of cancer or diabetes can be a matter of life and death for the patient. In addition to the patient's own current situation, many other factors can be considered, including the patient's health history, medication history, family's history, and other environmental factors. This makes diagnosis as much of an art form as it is science. Systems, such as IBM Watson, absorb all the medical research to date and make probabilistic diagnoses in the form of a decision tree, along with a full explanation for their recommendations. These systems take away most of the guess work done by doctors in diagnosing ailments.

2. Treatment effectiveness: The prescription of medication and treatment is also a difficult choice out of so many possibilities. For example, there are more than 100 medications for hypertension (high blood pressure) alone. There are also interactions in terms of which drugs work well with others and which drugs do not. Decision trees can help doctors learn about and prescribe more effective treatments. Thus, the patients could recover their health faster with a lower risk of complications and cost.

3. Wellness management: This includes keeping track of patient health records, analyzing customer health trends, and proactively advising them to take any needed precautions.

4. Manage fraud and abuse: Some medical practitioners have unfortunately been found to conduct unnecessary tests and/or overbill the government and health insurance companies. Exception-reporting systems can identify such providers, and action can be taken against them.

5. Public health management: The management of public health is one of the important responsibilities of any government. By using effective forecasting tools and techniques, governments can better predict the onset of disease in certain areas in real time. They can thus be better prepared to fight the diseases. Google has been known to predict the movement of certain diseases by tracking the search terms (like flu, vaccine) used in different parts of the world.

Education

As higher education becomes more expensive and competitive, it is a great user of data-based decision-making. There is a strong need for efficiency, increasing revenue, and improving the quality of student experience at all levels of education.

1. Student enrolment (recruitment and retention): Marketing to new potential students requires schools to develop profiles of the students that are most likely to attend. Schools can develop models of what kinds of students are attracted to the school, and then reach out to those students. The students at risk of not returning can be flagged, and corrective measures can be taken in time.

2. Course offerings: Schools can use the class enrolment data to develop models of which new courses are likely to be more popular with students. This can help increase class size, reduce costs, and improve student satisfaction.

3. Alumni pledges: Schools can develop predictive models of which alumni are most likely to pledge financial support to the school. Schools can create a profile for alumni more likely to pledge donations to the school. This could lead to a reduction in the cost of mailings and other forms of outreach to alumni.

Retail

Retail organizations grow by meeting customer needs with quality products, in a convenient, timely, and cost-effective manner. Understanding emerging customer shopping patterns can help retailers organize their products, inventory, store layout, and web presence in order to delight their customers, which in turn would help increase revenue and profits. Retailers generate a lot of transaction and logistics data that can be used to solve problems.

1. Optimize inventory levels at different locations: Retailers need to manage their inventories carefully. Carrying too much inventory imposes carrying costs, while carrying too little inventory can cause stockouts and lost sales opportunities. Predicting sales trends dynamically can help retailers move inventory to where it is most in demand. Retail organizations can provide their suppliers with real-time information about sales of their items so that the suppliers can deliver their product to the right locations and minimize stock-outs.

2. Improve store layout and sales promotions: A market basket analysis can develop predictive models of which products sell together often. This knowledge of affinities between products can help retailers co-locate those products. Alternatively, those affinity products could be located farther apart to make the customer walk the length and breadth of the store, and thus be exposed to other products. Promotional discounted product bundles can be created to push a non-selling item along with a set of products that sell well together

3. Optimize logistics for seasonal effects: Seasonal products offer tremendously profitable short-term sales opportunities, yet they also offer the risk of unsold inventories at the end of the season. Understanding which products are in season in which market can help retailers dynamically manage prices to ensure their inventory is sold during the season. If it is raining in a certain area, then the inventory of umbrella and ponchos could be rapidly moved there from non-rainy areas to help increase sales.

4. Minimize losses due to limited shelf life: Perishable goods offer challenges in terms of disposing off the inventory in time. By tracking sales trends, the perishable

products at risk of not selling before the sell-by date can be suitably discounted and promoted.

Banking

Banks make loans and offer credit cards to millions of customers. They are most interested in improving the quality of loans and reducing bad debts. They also want to retain more good customers and sell more services to them.

1. Automate the loan application process: Decision models can be generated from past data that predict the likelihood of a loan proving successful. These can be inserted in business processes to automate the financial loan application process.

2. Detect fraudulent transactions: Billions of financial transactions happen around the world every day. Exception-seeking models can identify patterns of fraudulent transactions. For example, if money is being transferred to an unrelated account for the first time, it could be a fraudulent transaction.

3. Maximize customer value (cross-selling, upselling): Selling more products and services to existing customers is often the easiest way to increase revenue. A checking account customer in good standing could be offered home, auto, or educational loans on more favorable terms than other customers, and thus, the value generated from that customer could be increased.

4. Optimize cash reserves with forecasting: Banks have to maintain certain liquidity to meet the needs of depositors who may like to withdraw money. Using past data and trend analysis, banks can forecast how much to keep, and invest the rest to earn interest.

Financial Services

Stock brokerages are an intensive user of BI systems. Fortunes can be made or lost based on access to accurate and timely information.

1. Predict changes in bond and stock prices: Forecasting the price of stocks and bonds is a favorite pastime of financial experts as well as lay people. Stock transaction data from the past, along with other variables, can be used to predict future price patterns. This can help traders develop long-term trading strategies.

2. Assess the effect of events on market movements: Decision models using decision trees can be created to assess the impact of events on changes in market volume and prices. Monetary policy changes (such as Fed Reserve interest rate change) or geopolitical changes (such as war in a part of the world) can be factored into the predictive model to help take action with greater confidence and less risk.

3. Identify and prevent fraudulent activities in trading: There have unfortunately been many cases of insider trading, leading to many prominent financial industry stalwarts going to jail. Fraud detection models can identify and flag fraudulent activity patterns.

Insurance

This industry is a prolific user of prediction models in pricing insurance proposals and managing losses from claims against insured assets.

1. Forecast claim costs for better business planning: When natural disasters, such as hurricanes and earthquakes, strike, loss of life and property occurs. By using the best available data to model the likelihood (or risk) of such events happening, the insurer can plan for losses and manage resources and profits effectively.

2. Determine optimal rate plans: Pricing an insurance rate plan requires covering the potential losses and making a profit. Insurers use actuary tables to project life spans and disease tables to project mortality rates, and thus price themselves competitively yet profitably.

3. Optimize marketing to specific customers: By micro segmenting potential customers, a data-savvy insurer can cherry-pick the best customers and leave the less profitable customers to its competitors. Progressive Insurance is a U.S.-based company that is known to actively use data mining to cherry-pick customers and increase its profitability.

4. Identify and prevent fraudulent claim activities: Patterns can be identified as to where and what kinds of fraud are more likely to occur. Decision-tree-based models can be used to identify and flag fraudulent claims.

Manufacturing

Manufacturing operations are complex systems with interrelated subsystems. From machines working right, to workers having the right skills, to the right components arriving with the right quality at the right time, to money to source the components, many things have to go right. Toyota’s famous lean manufacturing company works on just-in-time inventory systems to optimize investments in inventory and to improve flexibility in their product mix.

1. Discover novel patterns to improve product quality: Quality of a product can also be tracked, and this data can be used to create a predictive model of product quality deteriorating. Many companies, such as automobile companies, have to recall their products if they have found defects that have a public safety implication. Data mining can help with root cause analysis that can be used to identify sources of errors and help improve product quality in the future.

2. Predict/prevent machinery failures: Statistically, all equipment is likely to break down at some point in time. Predicting which machine is likely to shut down is a complex process. Decision models to forecast machinery failures could be constructed using past data. Preventive maintenance can be planned, and manufacturing capacity can be adjusted, to account for such maintenance activities.

Telecom

BI in telecom can help with churn management, marketing/customer profiling, network failure, and fraud detection.

1. Churn management: Telecom customers have shown a tendency to switch their providers in search for better deals. Telecom companies tend to respond with many incentives and discounts to hold on to customers. However, they need to determine which customers are at a real risk of switching and which others are just negotiating for a better deal. The level of risk should to be factored into the kind of deals and discounts that should be given. Millions of such customer calls happen every month. The telecom companies need to provide a consistent and data-based way to predict the risk of the customer switching, and then make an operational decision in real time while the customer call is taking place. A decision-tree- or a neural network based system can be used to guide the customer-service call operator to make the right decisions for the company, in a consistent manner.

2. Marketing and product creation: In addition to customer data, telecom companies also store call detail records (CDRs), which precisely describe the calling behavior of each customer. This unique data can be used to profile customers and then can be used for creating new products/services bundles for marketing purposes. An American telecom company, MCI, created a program called Friends & Family that allowed calls with one’s friends and family on that network to be totally free and thus, effectively locked many people into their network.

3. Network failure management: Failure of telecom networks for technical failures or malicious attacks can have devastating impacts on people, businesses, and society. In telecom infrastructure, some equipment will likely fail with certain mean time between failures. Modeling the failure pattern of various components of the network can help with preventive maintenance and capacity planning.

4. Fraud management: There are many kinds of fraud in consumer transactions. Subscription fraud occurs when a customer opens an account with the intention of never paying for the services. Superimposition fraud involves illegitimate activity by a person other than the legitimate account holder. Decision rules can be developed to analyze each CDR in real time to identify chances of fraud and take effective action.

Government

Government gathers a large amount of data by virtue of their regulatory function. That data could be analyzed for developing models of effective functioning.

1. Law enforcement: Social behavior is a lot more patterned and predictable than one would imagine. For example, Los Angeles Police Department (LAPD) mined the data from its 13 million crime records over 80 years and developed models of what kind of crime going to happen when and where. By increasing patrolling in those particular areas, LAPD was able to reduce property crime by 27 percent. Internet chatter can be analyzed to learn of and prevent any evil designs.

2. Scientific research: Any large collection of research data is amenable to being mined for patterns and insights. Protein folding (microbiology), nuclear reaction analysis (subatomic physics), disease control (public health) are some examples where data mining can yield powerful new insights.

How Business Intelligence systems are implemented?

Here are the steps:

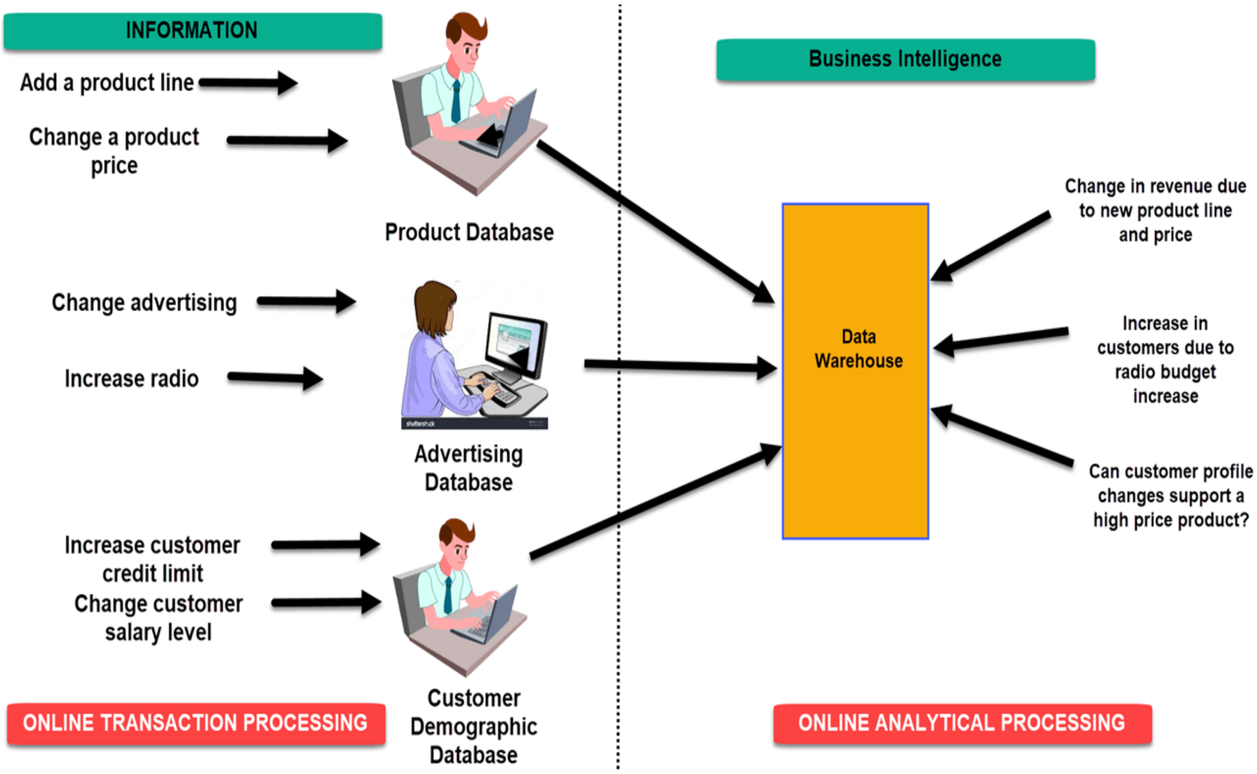
Step 1) Raw Data from corporate databases is extracted. The data could be spread across multiple systems heterogeneous systems.

Step 2) The data is cleaned and transformed into the data warehouse. The table can be linked, and data cubes are formed.

Step 3) Using BI system the user can ask quires, request ad-hoc reports or conduct any other analysis.

Examples of Business Intelligence System used in Practice

Example 1:


INFORMATION
Add a product line
Change a product price
Product Database
Change advertising
Increase radio
Advertising Database
Increase customer credit limit
Change customer salary level
Customer Demographic Database
ONLINE TRANSACTION PROCESSING
Business Intelligence
Data Warehouse
Change in revenue due to new product line and price
Increase in customers due to radio budget increase
Can customer profile changes support a high price product?
ONLINE ANALYTICAL PROCESSING

In an Online Transaction Processing (OLTP) system information that could be fed into product database could be

- add a product line
- change a product price

Correspondingly, in a Business Intelligence system query that would be executed for the product subject area could be did the addition of new product line or change in product price increase revenues

In an advertising database of OLTP system query that could be executed

- Changed in advertisement options
- Increase radio budget

Correspondigly, in BI system query that could be executed would be how many new clients added due to change in radio budget

In OLTP system dealing with customer demographic data bases data that could be fed would be

- increase customer credit limit
- change in customer salary level

Correspondingly in the OLAP system query that could be executed would be can customer profile changes support support higher product price

Example 2:

A hotel owner uses BI analytical applications to gather statistical information regarding average occupancy and room rate. It helps to find aggregate revenue generated per room.

It also collects statistics on market share and data from customer surveys from each hotel to decides its competitive position in various markets.

By analyzing these trends year by year, month by month and day by day helps management to offer discounts on room rentals.

Example 3:

A bank gives branch managers access to BI applications. It helps branch manager to determine who are the most profitable customers and which customers they should work on.

The use of BI tools frees information technology staff from the task of generating analytical reports for the departments. It also gives department personnel access to a richer data source.

Four types of BI users

Following given are the four key players who are used Business Intelligence System:

1. The Professional Data Analyst:

The data analyst is a statistician who always needs to drill deep down into data. BI system helps them to get fresh insights to develop unique business strategies.

2. The IT users:

The IT user also plays a dominant role in maintaining the BI infrastructure.

3. The head of the company:

CEO or CXO can increase the profit of their business by improving operational efficiency in their business.

4. The Business Users"

Business intelligence users can be found from across the organization. There are mainly two types of business users

1. Casual business intelligence user
2. The power user.

The difference between both of them is that a power user has the capability of working with complex data sets, while the casual user need will make him use dashboards to evaluate predefined sets of data.

Advantages of Business Intelligence

Here are some of the advantages of using Business Intelligence System:

1. Boost productivity

With a BI program, It is possible for businesses to create reports with a single click thus saves lots of time and resources. It also allows employees to be more productive on their tasks.

2. To improve visibility

BI also helps to improve the visibility of these processes and make it possible to identify any areas which need attention.

3. Fix Accountability

BI system assigns accountability in the organization as there must be someone who should own accountability and ownership for the organization's performance against its set goals.

4. It gives a bird's eye view:

BI system also helps organizations as decision makers get an overall bird's eye view through typical BI features like dashboards and scorecards.

5. It streamlines business processes:

BI takes out all complexity associated with business processes. It also automates analytics by offering predictive analysis, computer modelling, benchmarking and other methodologies.

6. It allows for easy analytics.

BI software has democratized its usage, allowing even nontechnical or non-analysts users to collect and process data quickly. This also allows putting the power of analytics from the hand's many people.

BI System Disadvantages

1. Cost:

Business intelligence can prove costly for small as well as for medium-sized enterprises. The use of such type of system may be expensive for routine business transactions.

2. Complexity:

Another drawback of BI is its complexity in implementation of data warehouse. It can be so complex that it can make business techniques rigid to deal with.

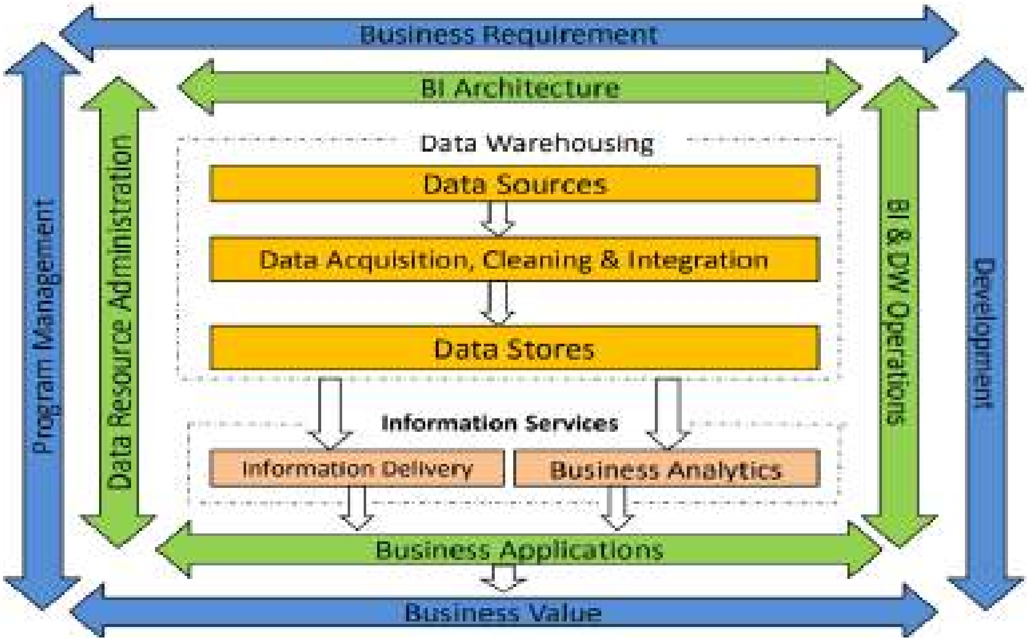
3. Limited use

Like all improved technologies, BI was first established keeping in consideration the buying competence of rich firms. Therefore, BI system is yet not affordable for many small and medium size companies.

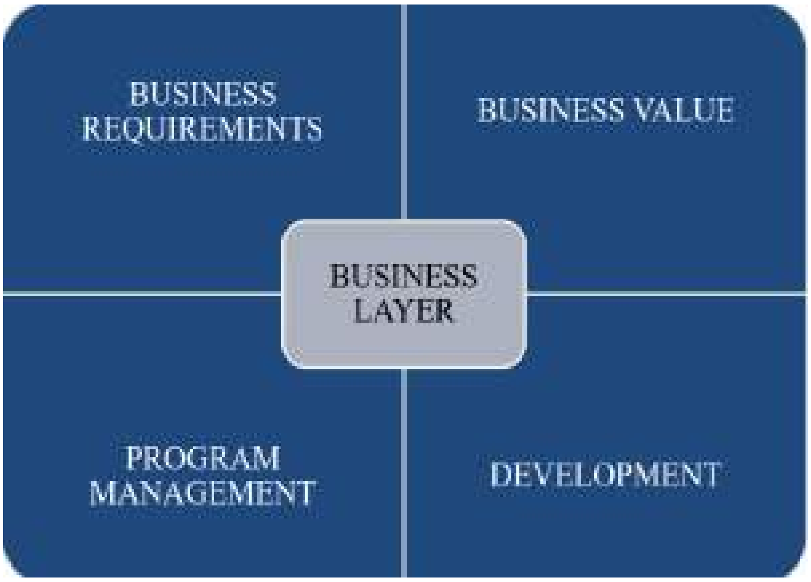
4. Time Consuming Implementation

It takes almost one and half year for data warehousing system to be completely implemented. Therefore, it is a time-consuming process.

BI Framework



Business Layer



Business Layer

Business requirements: The requirements are a product of three steps of a process that includes:

- Business drivers (the impulses that initiate the need to act).

  Examples: changing workforce, changing labor laws, changing economy, changing technology, etc.

- Business goals (the targets to be achieved in response to the business drivers).

  Examples: increased productivity, improved market share, improved profit margins, improved customer satisfaction, cost reduction, etc.

- Business strategies (the planned course of action that will help achieve the set goals).

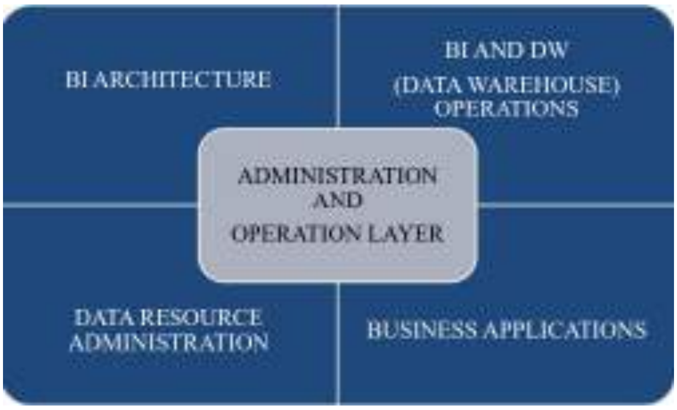
  Examples: outsourcing, global delivery model, partnerships, customer retention programs, employee retention programs, competitive pricing, etc

Business Value: Business value can be measured in terms of ROI (Return on Investment), ROA (Return on Assets), TCO (Total Cost of Ownership), TVO (Total Value of Ownership), etc.

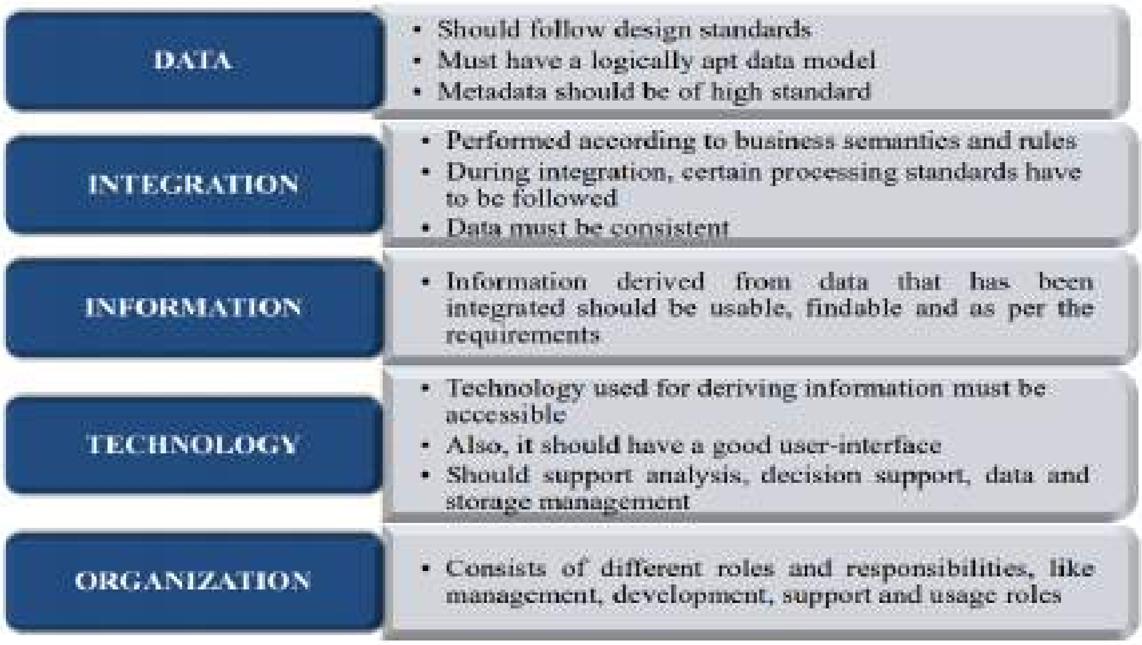
Program management: It is the component that ensures people, projects and priorities work in a manner in which individual processes are compatible with each other; so as to ensure seamless integration and smooth functioning of the entire program.

Development: The process of development consists of database/datawarehouse development (consisting of ETL, data profiling, data cleansing and database tools), data integration system development (consists of data integration tools and data quality tools) and business analytics development (about processes and various technologies used).

Administration and Operation Layer


BI ARCHITECTURE
BI AND DW (DATA WAREHOUSE) OPERATIONS
ADMINISTRATION AND OPERATION LAYER
DATA RESOURCE ADMINISTRATION
BUSINESS APPLICATIONS

BI Architecture



DATA	• Should follow design standards • Must have a logically apt data model • Metadata should be of high standard
INTEGRATION	• Performed according to business semantics and rules • During integration, certain processing standards have to be followed • Data must be consistent
INFORMATION	• Information derived from data that has been integrated should be usable, findable and as per the requirements
TECHNOLOGY	• Technology used for deriving information must be accessible • Also, it should have a good user-interface • Should support analysis, decision support, data and storage management
ORGANIZATION	• Consists of different roles and responsibilities, like management, development, support and usage roles

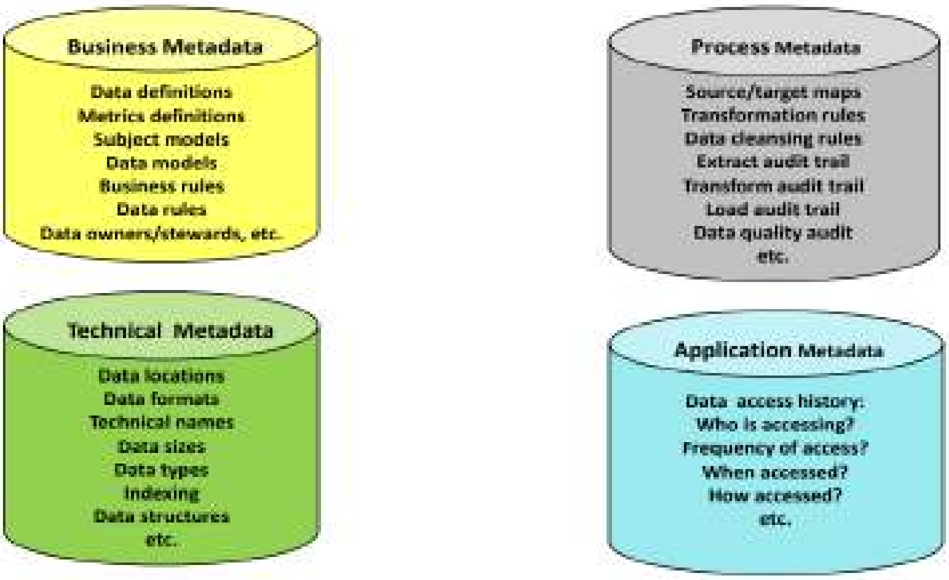
BI and DW operations: Data warehouse administration requires the usage of various tools to monitor the performance and usage of the warehouse, and perform administrative tasks on it. Some of these tools would be:

- Backup and restore
- Security
- Configuration management
- Database management

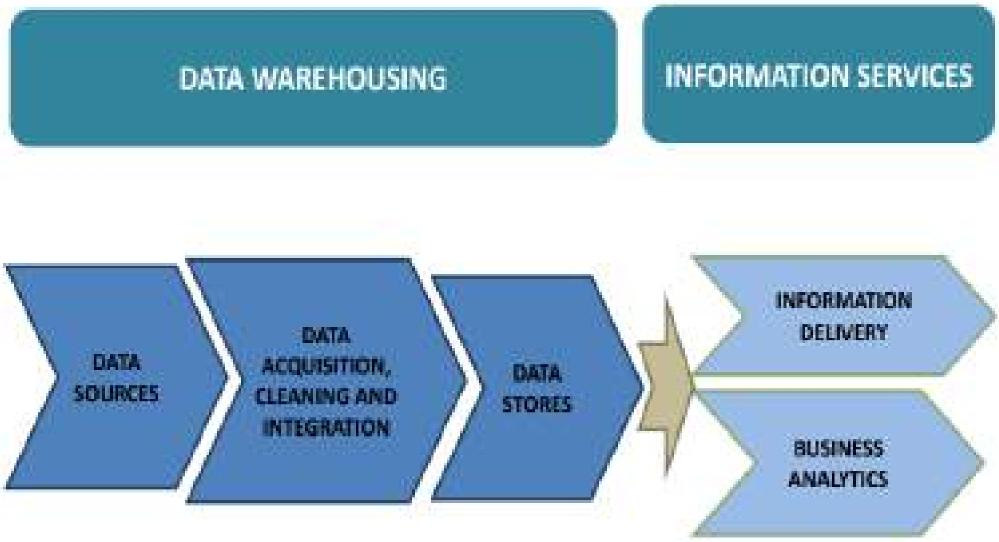
Data resource administration: Involves data governance and metadata management. Data governance is a technique for controlling data quality, which is used to assess, improve, manage and maintain information. It helps to define standards that are required to maintain data quality. The distribution of roles for governance of data is as follows:

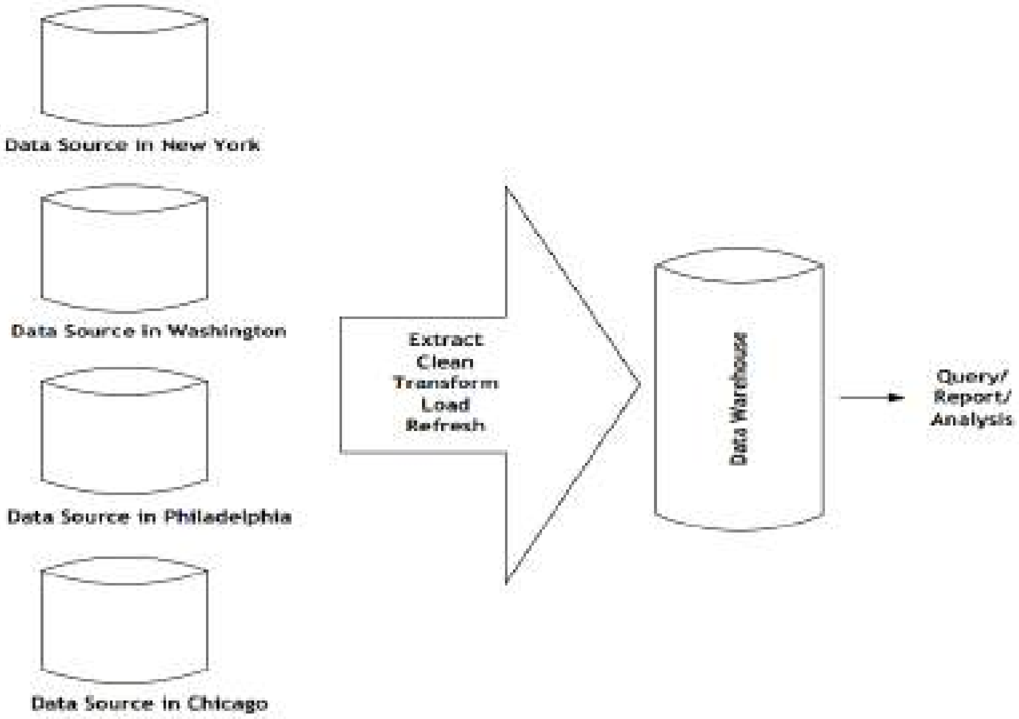
- Data ownership
- Data stewardship
- Data custodianship

Metadata management: Metadata is data about data. Metadata can be divided into four groups: – Business metadata – Process metadata – Technical metadata – Application metadata



Implementation Layer

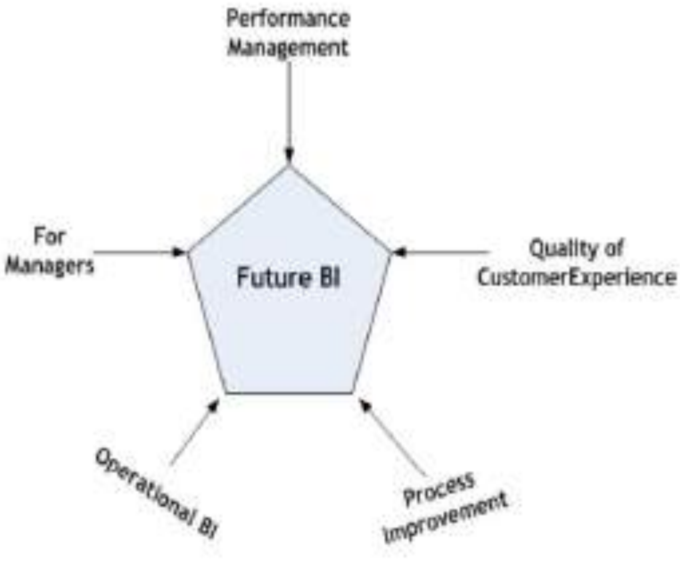




Information services:

- It is not only the process of producing information; rather, it involves ensuring that the information produced is aligned with business requirements and can be acted upon to produce value for the company.
- Information is delivered in the form of KPI's, reports, charts, dashboards or scorecards, etc., or in the form of analytics.
- Data mining is a practice used to increase the body of knowledge.
- Applied analytics is generally used to drive action and produce outcomes.

Who is BI for?



Types of BI Users

Type of user	**Casual users/ Information consumers**	**Power users/Information producers**
Example of such users	Executives, managers, customers, suppliers, field/operation workers, etc.	SAS, SPSS developers, administrators, business analysts, analytical modelers, IT professionals, etc.
Usage	Information consumers	Information producers
Data Access	Tailor made to suit the needs of their respective role	Ad hoc/exploratory
Tools	Pre-defined reports/dashboards	Advanced Analytical/ Authoring tools
Sources	Data warehouse/Data Marts	Data Warehouse/Data Marts (both internal and external)

BI Roles and Responsibilities

Program Roles	**Project Roles**
	Business Manager
BI Program Manager	BI Business Specialist
BI Data Architect	BI Project Manager
BI ETL Architect	Business Requirements Analyst
BI Technical Architect	Decision Support Analyst
Metadata Manager	BI Designer
BI Administrator	ETL Specialist
	Data Administrator

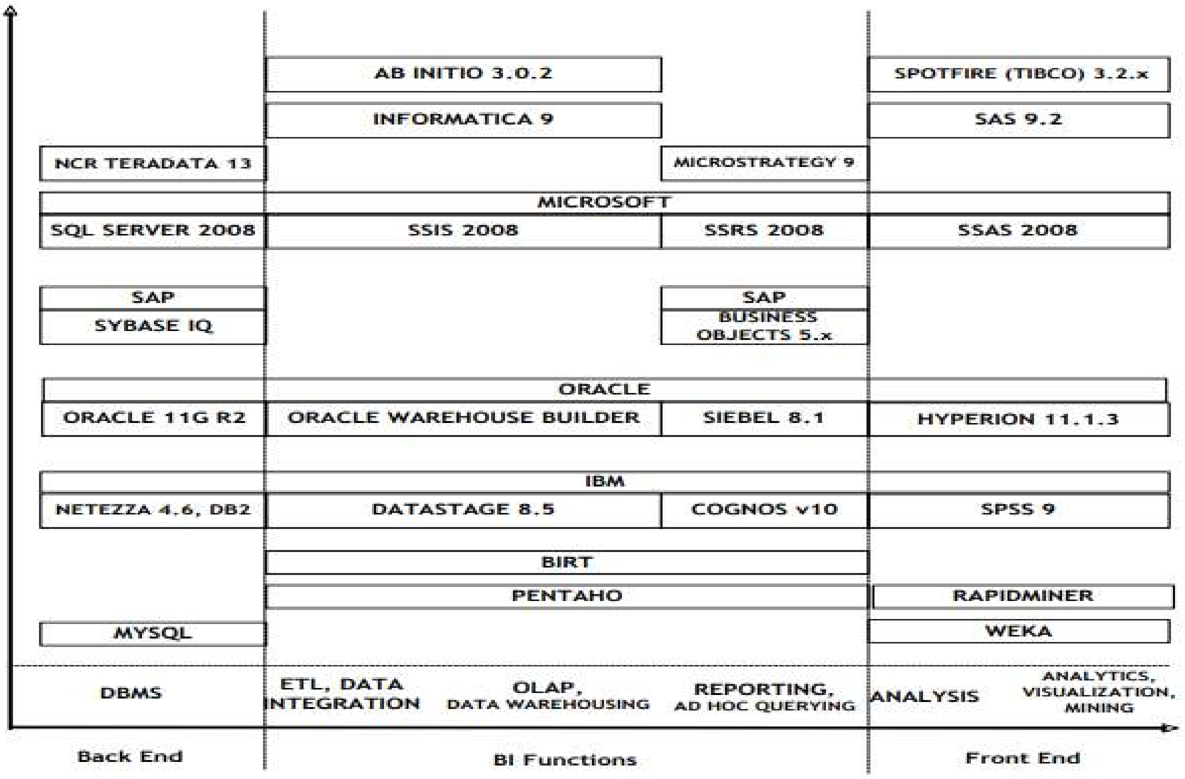
BI DW Best Practices

- Practice “User First” Design
- Create New Value
- Attend to Human Impacts
- Focus on Information and Analytics
- Practice Active Data Stewardship
- Manage BI as a long term investment
- Reach out with BI/DW solutions
- Make BI a business Initiative
- Measure Results
- Attend to strategic Positioning

Open Source BI Tools

RDBMS	MySQL, Firebird
ETL Tools	Pentaho Data Integration (formerly called Kettle), SpagoBI
Analysis Tools	Weka, RapidMiner, SpagoBI
Reporting Tools/Ad Hoc Querying/Visualization	Pentaho, BIRT, Actuate, Jaspersof

Popular BI Tools



Unit V - Introduction to Multi-Dimensional Data Modeling

Introduction

Two data modeling techniques that are relevant in a data warehousing environment are ER modeling and Multidimensional modeling.

ER Modeling

ER modeling produces a data model of the specific area of interest, using two basic concepts: entities and the relationships between those entities. The ER model is an abstraction tool because it can be used to understand and simplify the ambiguous data relationships in the business world and complex systems environments. An ER model is represented by an ER diagram, which uses three basic graphic symbols to conceptualize the data: entity, relationship, and attribute.

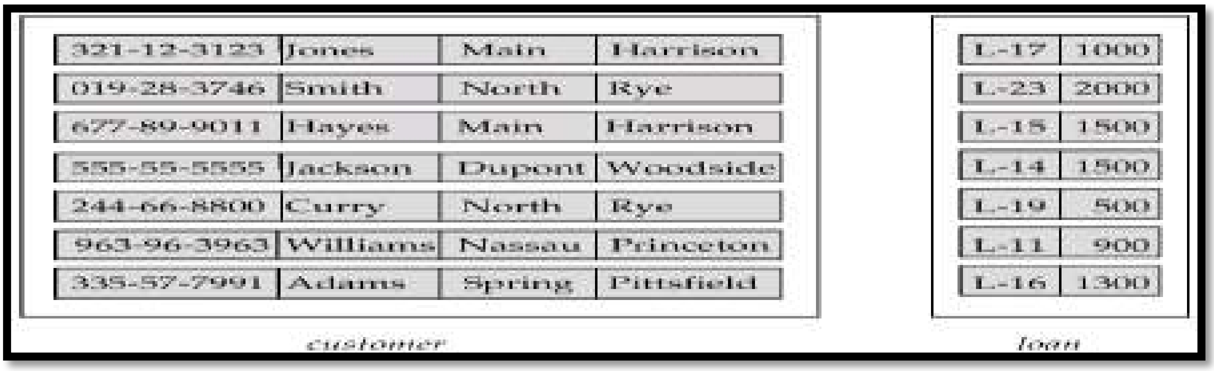
- ER model consists of set of three basic objects called
  - ❖ Entities
  - ❖ Attributes
  - ❖ Relationships
- E-R model is popular for high level database design
- It provides a means for representing the relationship between entities.
- The entity-relationship (E-R) data model perceives the real world as a collection of entities and relationships.

ENTITY :

An entity is defined to be a person, place, thing, or event of interest to the business or the organization. An entity represents a class of objects, which are things in the real world that can be observed and classified by their properties and characteristics.

- An entity consists of a set of attributes.
  - o Examples: In a customer table, customer id, name, street and city
- An entity is an object that has its existence in the real world. Example: The set of all persons who are customers at a given bank (can be defined as an entity called *customer*)

- Entity *customer* and *loan*



customer-id	customer-name	customer-street	customer-city
321-12-3123	Jones	Main	Harrison
019-28-3746	Smith	North	Rye
677-89-9011	Hayes	Main	Harrison
555-55-5555	Jackson	Dupont	Woodside
244-66-8800	Curry	North	Rye
963-96-3963	Williams	Nassau	Princeton
335-57-7991	Adams	Spring	Pittsfield

customer

loan-number	amount
L-17	1000
L-23	2000
L-15	1500
L-14	1500
L-19	500
L-11	900
L-16	1300

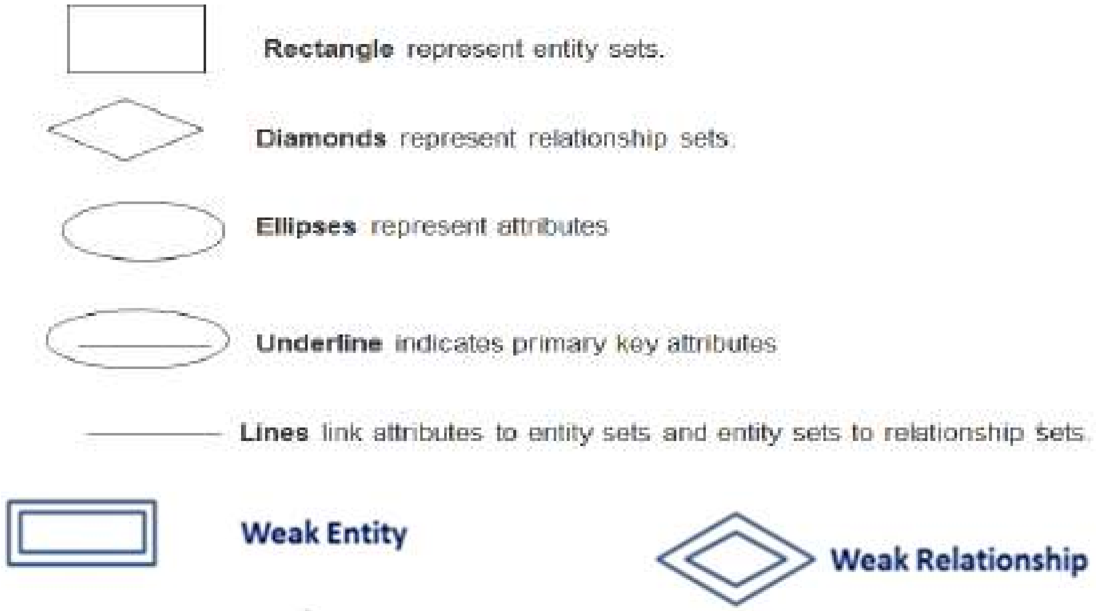
loan

ATTRIBUTES:

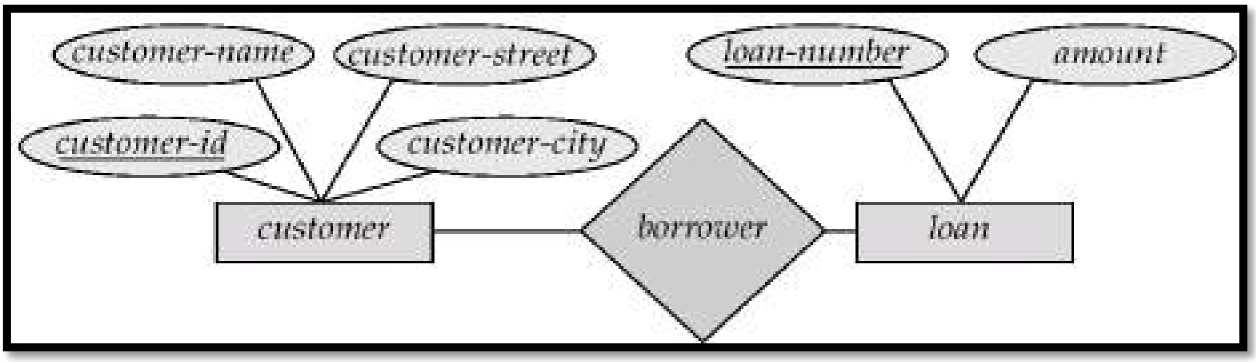
Attributes describe the characteristics of properties of the entities. For clarification, attribute naming conventions are very important. An attribute name should be unique in an entity and should be self-explanatory. When an instance has no value for an attribute, the minimum cardinality of the attribute is zero, which means either null able or optional. In ER modeling, if the maximum cardinality of an attribute is more than 1, the modeler will try to normalize the entity and finally elevate the attribute to another entity. Therefore, normally the maximum cardinality of an attribute is 1.

An entity is represented by a set of attributes. These attributes are commonly shared by all members in an entity set.

e.g. *customer* = (*customer-id*, *customer-name*, *customer-street*, *customer-city*)
loan = (*loan-number*, *amount*)


Rectangle represent entity sets.
Diamonds represent relationship sets.
Ellipses represent attributes
Underline indicates primary key attributes
Lines link attributes to entity sets and entity sets to relationship sets.
Weak Entity
Weak Relationship

E-R Diagrams

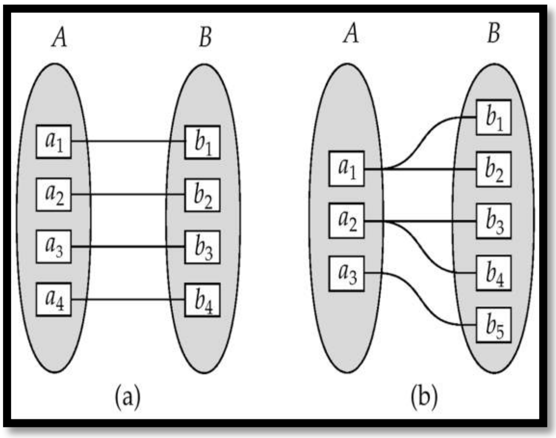


RELATIONSHIP:

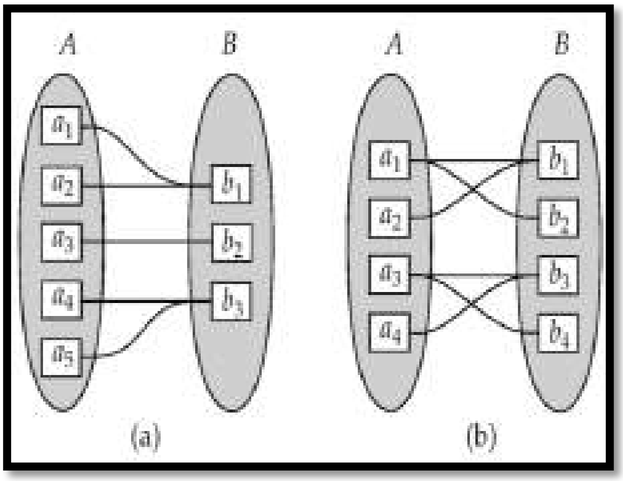
A relationship is represented with lines drawn between entities. It depicts the structural interaction and association among the entities in a model. A relationship is designated grammatically by a verb, such as owns, belongs, and has. The relationship between two entities can be defined in terms of the cardinality. This is the maximum number of instances of one entity that are related to a single instance in another table and vice versa. The possible cardinalities are: one-to-one (1:1), one-to-many (1:M), and many-to-many (M:M).

Types of Relationships (Mapping Cardinalities)

1. Express the number of entities to which another entity can be associated via a relationship set.
2. There are four types of relationships viz.,
    - One to one
    - One to many
    - Many to one
    - Many to many



One to one One to many



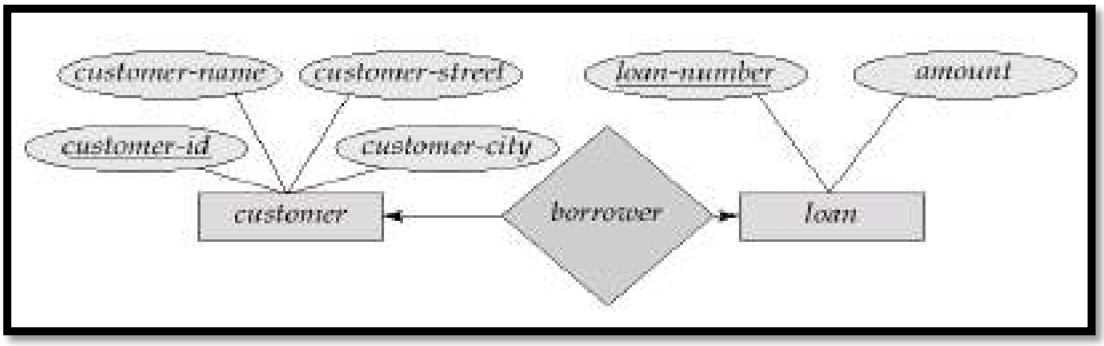
Many to one Many to many

Representing Mapping Cardinality Constraints

We express cardinality constraints by drawing either a directed line (→), signifying "one," or an undirected line (—), signifying "many," between the relationship set and the entity set.

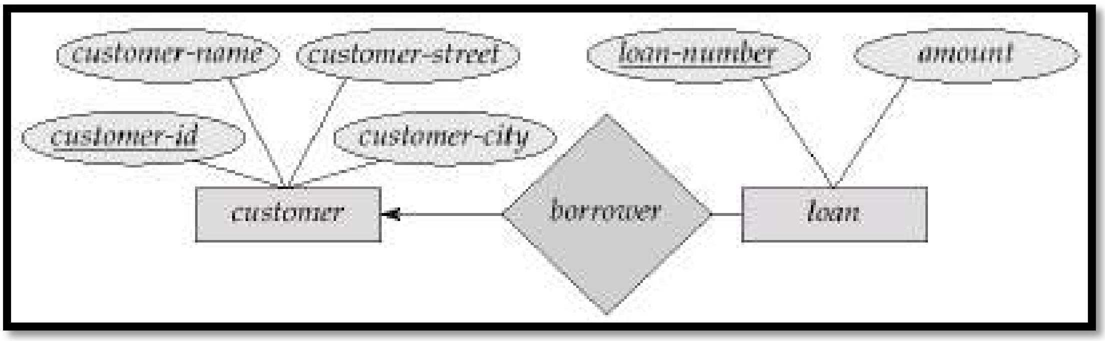
One-To-One Relationship:

- A customer is associated with at most one loan via the relationship *borrower*
- A loan is associated with at most one customer via *borrower*



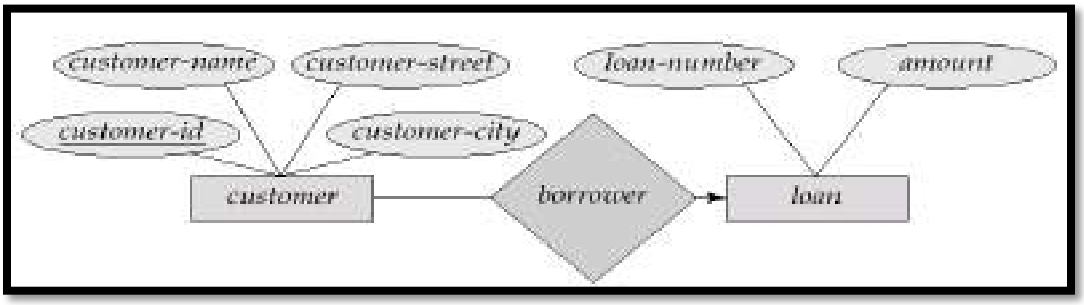
One-To-Many Relationship:

- a loan is associated with at most one customer via *borrower*, a customer is associated with several (including 0) loans via *borrower*.
  - So a customer may have no loan
  - It is also possible that a loan does not have a corresponding customer. But if it has, it has only one corresponding customer.



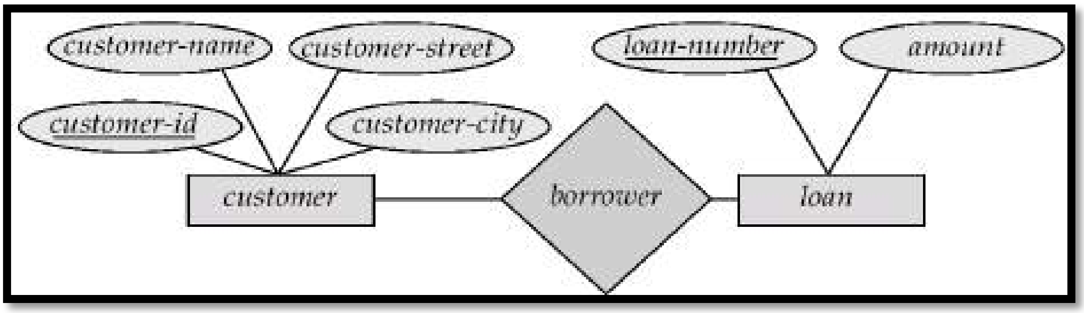
Many-To-One Relationships:

In a many-to-one relationship a loan is associated with several (including 0) customers via *borrower*, a customer is associated with at most one loan via *borrower*



Many-To-Many Relationship:

- A customer is associated with several (possibly 0) loans via borrower
- A loan is associated with several (possibly 0) customers via borrower

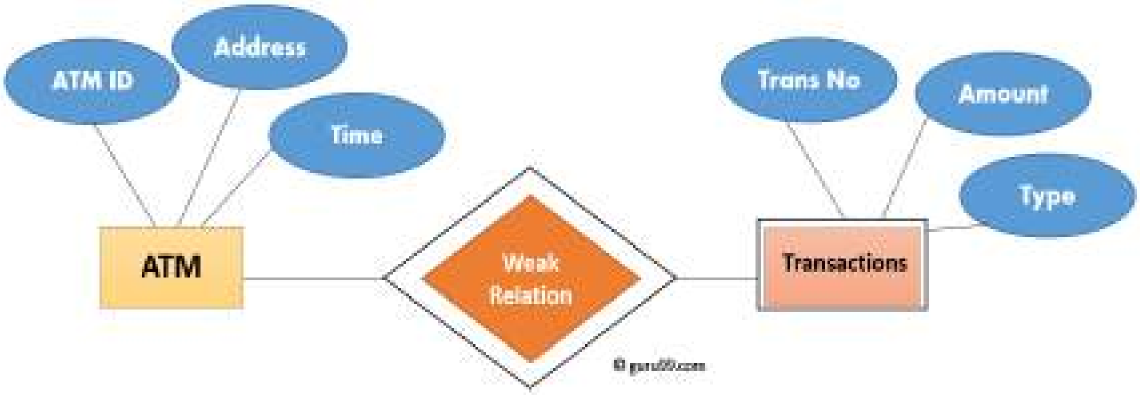


Weak Entity Set

It does not have enough attributes to build a primary key. It is represented by a double rectangle symbol.

Weak Relationship

The relationship between one strong and a weak entity set shown by using the double diamond symbol.



Dimensional Data Modeling

Dimensional Data Modeling is one of the data modeling techniques used in data warehouse design. The concept of Dimensional Modeling was developed by **Ralph Kimball** which is comprised of *facts(measures) and dimension(context)* tables.

Dimensional modeling represents data with a cube operation, making more suitable logical data representation with OLAP data management. In dimensional modeling, the transaction record is divided into either "**facts,**" which are frequently numerical transaction data, or "**dimensions,**" which are the reference information that gives context to the facts

Since the main goal of this modeling is to improve the data retrieval so it is optimized for *SELECT OPERATION*. The advantage of using this model is that we can store data in such a way that it is easier to store and retrieve the data once stored in a data warehouse. Dimensional model is the data model used by many OLAP systems.

It is the design concept used by many data warehouse designers to build their data warehouse. Dimensional modelling (DM) names a set of techniques and concepts used in data warehouse design. It is simpler, more expressive and easier to understand than ER modelling.

It is especially useful for summarizing and rearranging the data and presenting views of the data to support data analysis. Dimensional modelling focuses on numeric data such as values, counts, weights, balances, and occurrences. It does not necessarily involve a relational database

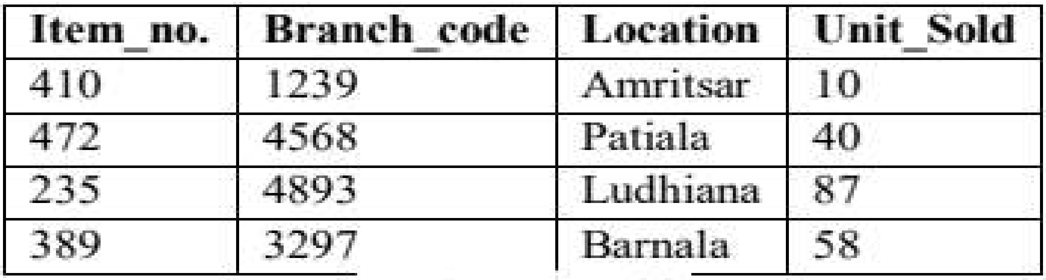
Dimensional modelling always uses the concepts of facts (measures) and dimensions (context). Facts are typically (but not always) numeric values that can be aggregated and dimensions are groups of hierarchies and descriptors that define the facts. For example, sales amount is a fact; timestamp, product, register, store, etc. are elements of dimensions. Dimensional models are built by business process area, e.g. store sales, inventory, claims, etc.

In this model, all data is contained in two types of tables called **Fact Table** and **Dimension Table**

Fact Table:

In a Dimensional Model, Fact table contains the measurements or metrics or facts of business processes. If your business process is Sales, then a measurement of this business process such as "monthly sales number" is captured in the fact table. In addition to the measurements, the only other things a fact table contains are foreign keys for the dimension tables. The fact is a set of related data, contains analytical context data and measures. It used to represents business items or business transactions. A fact table is the central table in a star schema of a data warehouse. A fact table stores quantitative information for analysis and is often denormalized. Suppose an electronic shop sells its product. Thus, every sale is a fact that happens and the fact table is used to record these facts.

For example:

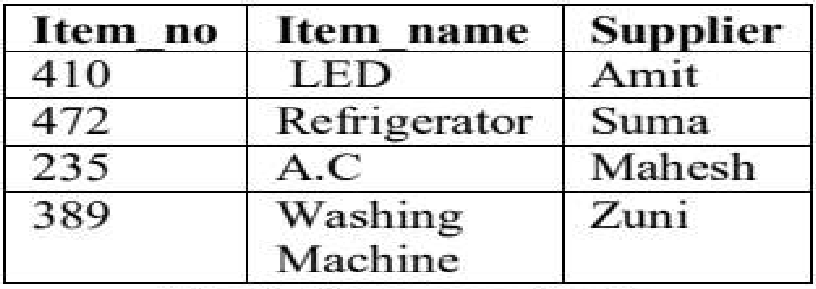


Item_no.	Branch_code	Location	Unit_Sold
410	1239	Amritsar	10
472	4568	Patiala	40
235	4893	Ludhiana	87
389	3297	Barnala	58

Fig 2. Fact Table

Dimension Table:

In a Dimensional Model, frameworks of the measurements are represented in dimension tables. A dimension table is a table in a star schema of a data warehouse. A dimension table stores attributes or dimensions that describe the objects in a fact table. From the above example Item_no dimension, the attributes can be Item_name, supplier, etc. Generally the Dimension Attributes are used in report labels, and query constraints such as where Supplier='Amit'.



Item_no	Item_name	Supplier
410	LED	Amit
472	Refrigerator	Suma
235	A.C	Mahesh
389	Washing Machine	Zuni

Fig 3. Dimension Table

It help us store the data in such a way that it is relatively easy to retrieve the data from the database.

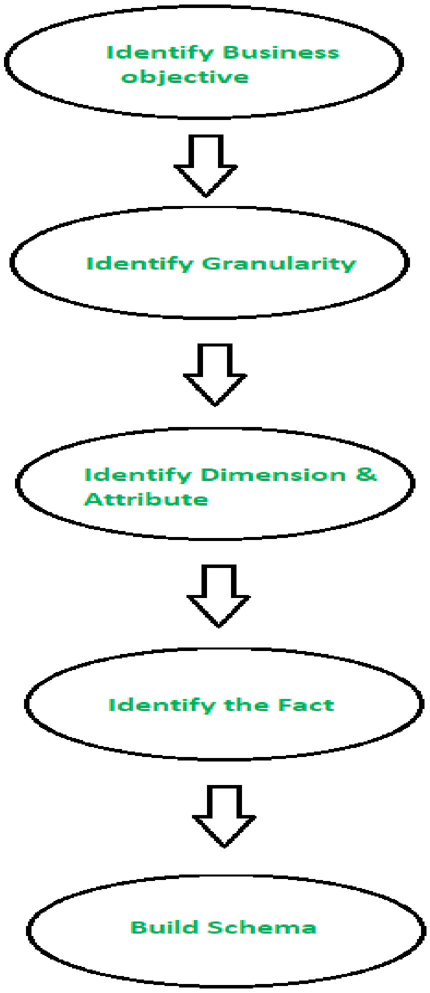


Figure – Steps for Dimensional Model

Steps to Create Dimensional Data Modeling:

Step-1: Identifying the business objective –

The first step is to identify the business objective. Sales, HR, Marketing, etc. are some examples as per the need of the organization. Since it is the most important step of Data Modelling the selection of business objective also depends on the quality of data available for that process.

Step-2: Identifying Granularity –

Granularity is the lowest level of information stored in the table. The level of detail for business problem and its solution is described by Grain.

Step-3: Identifying Dimensions and its Attributes –

Dimensions are objects or things. Dimensions categorize and describe data warehouse facts and measures in a way that support meaningful answers to business questions. A data warehouse organizes descriptive attributes as columns in dimension tables. For Example, the data dimension may contain data like a year, month and weekday.

Step-4: Identifying the Fact –

The measurable data is hold by the fact table. Most of the fact table rows are numerical values like price or cost per unit, etc.

Step-5: Building of Schema –

We implement the Dimension Model in this step. A schema is a database structure. There are two popular schemes: Star Schema and Snowflake Schema.

Schema

Schema is a logical description of the entire database. It includes the name and description of records of all record types including all associated data-items and aggregates. Much like a database, a data warehouse also requires to maintain a schema. A database uses relational model, while a data warehouse uses Star, Snowflake, and Fact Constellation schema. In this chapter, we will discuss the schemas used in a data warehouse. A schema is a database structure.

There are two popular schemes:

1.Star Schema

2.Snowflake Schema

Star Schema

In data warehousing, A star schema is the simplest form of a dimensional model, in which data is organized into facts and dimensions. A fact is an event that is counted or measured, such as a sale or login. A dimension contains reference information about the fact, such as date, product, or customer. Star schema has become a common term used to connote a dimensional model. Database designers have long used the term star schema to describe dimensional models because the resulting structure looks like a star.

A star schema is characterized by one or more very large fact tables that contain the primary information in the data warehouse, and a number of much smaller dimension tables (or lookup tables), each of which contains information about the entries for a particular attribute in the fact table .

The main feature of a star schema is a fact table at the centre surrounded by dimensional tables; each one contains information about the entries for a particular attribute in the fact table.

Example 1:

The following diagram illustrates the star schema of a company which sells various products:

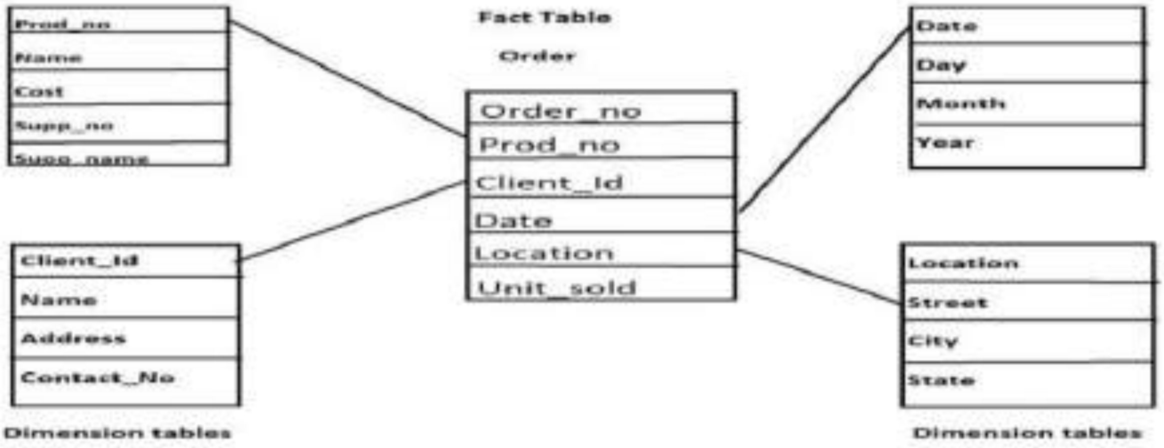
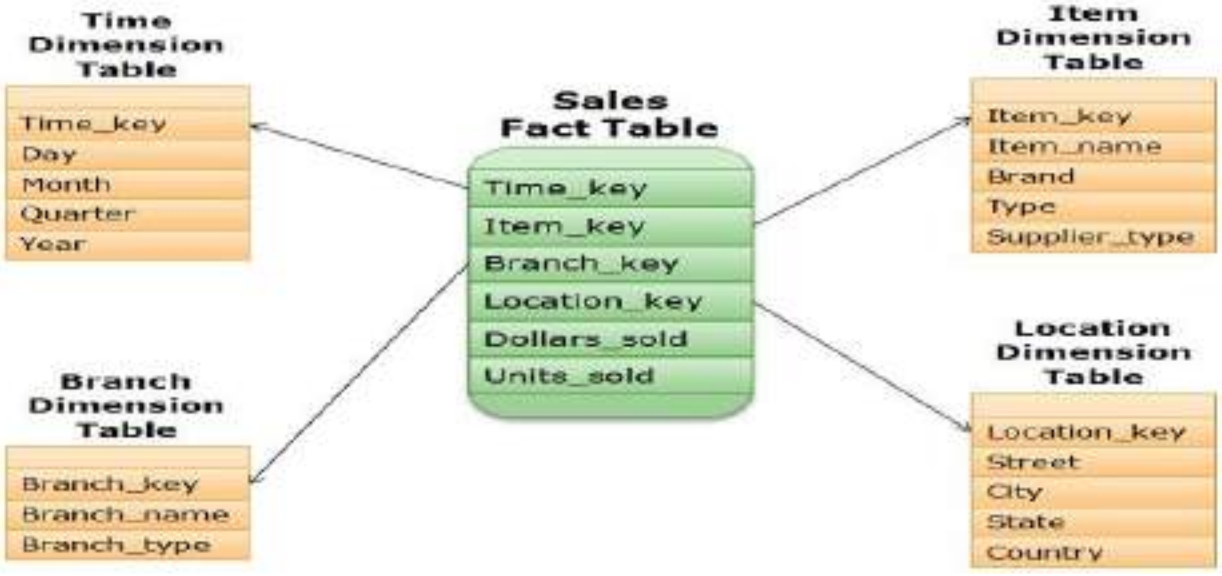


Fig 4. Star Schema

- Each dimension in a star schema is represented with only one-dimension table.
- This dimension table contains the set of attributes.

Example 2:

- The following diagram shows the sales data of a company with respect to the four dimensions, namely time, item, branch, and location.



- There is a fact table at the center. It contains the keys to each of four dimensions.
- The fact table also contains the attributes, namely dollars sold and units sold.

Note − Each dimension has only one dimension table and each table holds a set of attributes. For example, the location dimension table contains the attribute set {location_key, street, city, province_or_state,country}. This constraint may cause data redundancy. For example, "Vancouver" and "Victoria" both the cities are in the Canadian province of British Columbia. The entries for such cities may cause data redundancy along the attributes province_or_state and country.

Characteristics of Star Schema:

- Every dimension in a star schema is represented with the **only one-dimension table**.
- The dimension table should contain the set of attributes.
- The dimension table is joined to the fact table using a **foreign key.**
- **The dimension table are not joined to each other.**
- Fact table would contain **key and measure**.
- The Star schema is easy to understand and provides optimal disk usage.
- The dimension tables are **not normalized**.
- The schema is widely **supported by BI Tools.**

Snowflake Schema

The snowflake schema is an extension of the star schema, where each point of the star explodes into more points. In a star schema, each dimension is represented by a single dimensional table, whereas in a snowflake schema, that dimensional table is normalized into multiple lookup tables, each representing a level in the dimensional hierarchy. The snowflake schema architecture is a more complex because the dimensional tables are normalized.

It is an enhancement of star schema. It normalizes dimensions to eliminate redundancy. The decomposed snowflake structure visualizes the hierarchical structure of dimensions very well. The snowflake model is easy for data modellers to understand and for database designers to use for the analysis of dimensions.

The main advantage of the snowflake schema is the improvement in query performance due to minimized disk storage requirements and joining smaller lookup tables. The main disadvantage of the snowflake schema is the additional maintenance efforts needed due to the increase number of lookup tables.

Example 1:

The diagram of snowflake schema is as follows:-

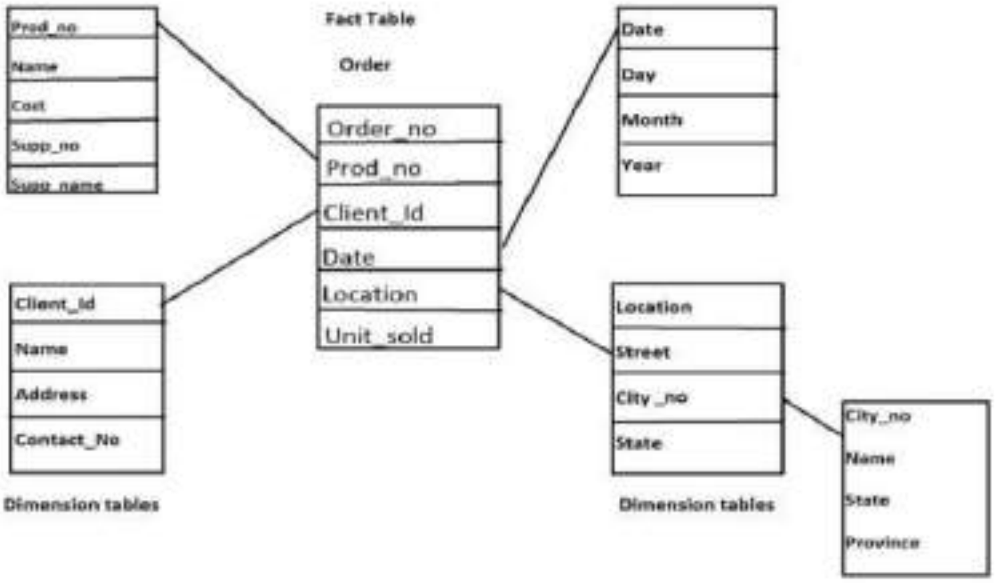
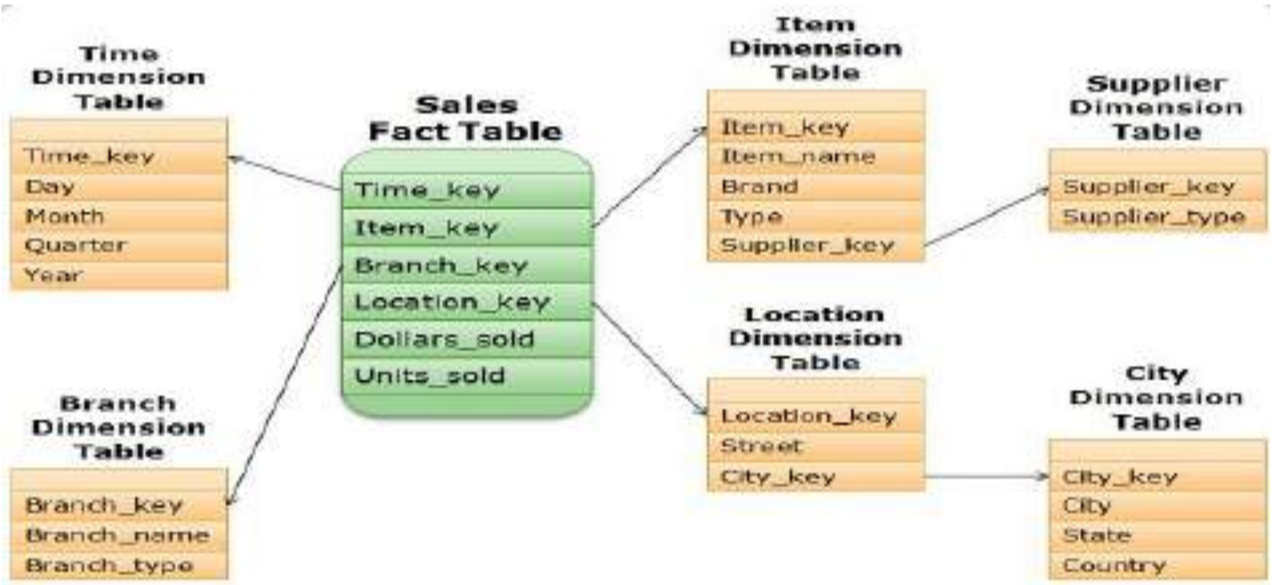


Fig 5.Snowflake Schema Diagram

- Some dimension tables in the Snowflake schema are normalized.
- The normalization splits up the data into additional tables.
- Unlike Star schema, the dimensions table in a snowflake schema are normalized.

Example 2:

For example, the item dimension table in star schema is normalized and split into two dimension tables, namely item and supplier table.



- Now the item dimension table contains the attributes item_key, item_name, type, brand, and supplier-key.
- The supplier key is linked to the supplier dimension table. The supplier dimension table contains the attributes supplier_key and supplier_type.

Note − Due to normalization in the Snowflake schema, the redundancy is reduced and therefore, it becomes easy to maintain and the save storage space.

Characteristics of Snowflake Schema:

- ➢ The main benefit of the snowflake schema it uses **smaller disk space**.
- ➢ Easier to implement a dimension is added to the Schema
- ➢ Due to multiple tables query performance is reduced
- ➢ The primary challenge that you will face while using the snowflake Schema is that you need to perform more maintenance efforts because of the more lookup tables.

Key Differences Between Star and Snowflake Schema:

- ➢ Star schema contains just **one dimension table** for one dimension entry while there may exist dimension and sub-dimension table for one entry.
- ➢ **Normalization** is used in snowflake schema which eliminates the data redundancy. As against, normalization is not performed in star schema which results in data redundancy.
- ➢ Star schema is simple, easy to understand and involves less intricate queries. On the contrary, snowflake schema is hard to understand and involves complex queries.
- ➢ The data model approach used in a star schema is **top-down** whereas snowflake schema uses bottom-up
- ➢ Star schema uses a fewer number of joins. On the other hand, snowflake schema uses a large number of joins.
- ➢ The space consumed by star schema is more as compared to snowflake schema.
- ➢ The time consumed for executing a query in a star schema is less. Conversely, snowflake schema consumes more time due to the excessive use of joins.

Comparison Chart

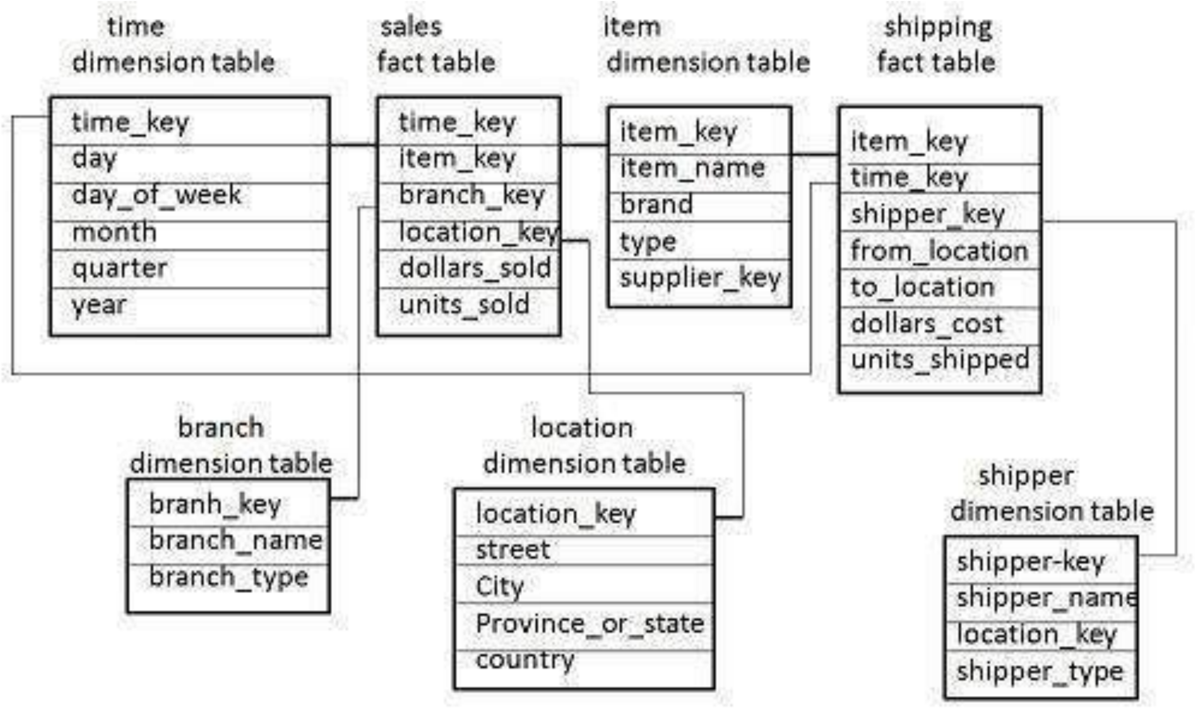
BASIS FOR COMPARISON	STAR SCHEMA	SNOWFLAKE SCHEMA
Structure of schema	Contains fact and dimension tables.	Contains sub-dimension tables including fact and dimension tables.
Use of normalization	Doesn't use normalization.	Uses normalization and denormalization.
Ease of use	Simple to understand and easily designed.	Hard to understand and design.
Data model	Top-down	Bottom-up
Query complexity	Low	High
Foreign key join used	Fewer	Large in number
Space usage	More	Less
Time consumed in query execution	Less	More comparatively due to excessive use of join.

Fact Constellation Schema

For each star schema or snowflake schema it is possible to construct a fact constellation schema. This kind of schema can be viewed as a collection of stars and hence it is called Fact Constellation. This schema is more complex than star or snowflake architecture, which is because it contains multiple fact tables. This allows dimension tables to be shared amongst many fact tables. That solution is very flexible, however it may be hard to manage and support.

The fact constellation architecture contains multiple fact tables that share many dimension tables. It is used mainly for the aggregate fact tables and for better understanding. The main disadvantage of the fact constellation schema is a more complicated design because many variants of aggregation must be considered.

- A fact constellation has multiple fact tables. It is also known as galaxy schema.
- The following diagram shows two fact tables, namely sales and shipping.



- The sales fact table is same as that in the star schema.
- The shipping fact table has the five dimensions, namely item_key, time_key, shipper_key, from_location, to_location.
- The shipping fact table also contains two measures, namely dollars sold and units sold.
- It is also possible to share dimension tables between fact tables. For example, time, item, and location dimension tables are shared between the sales and shipping fact table.

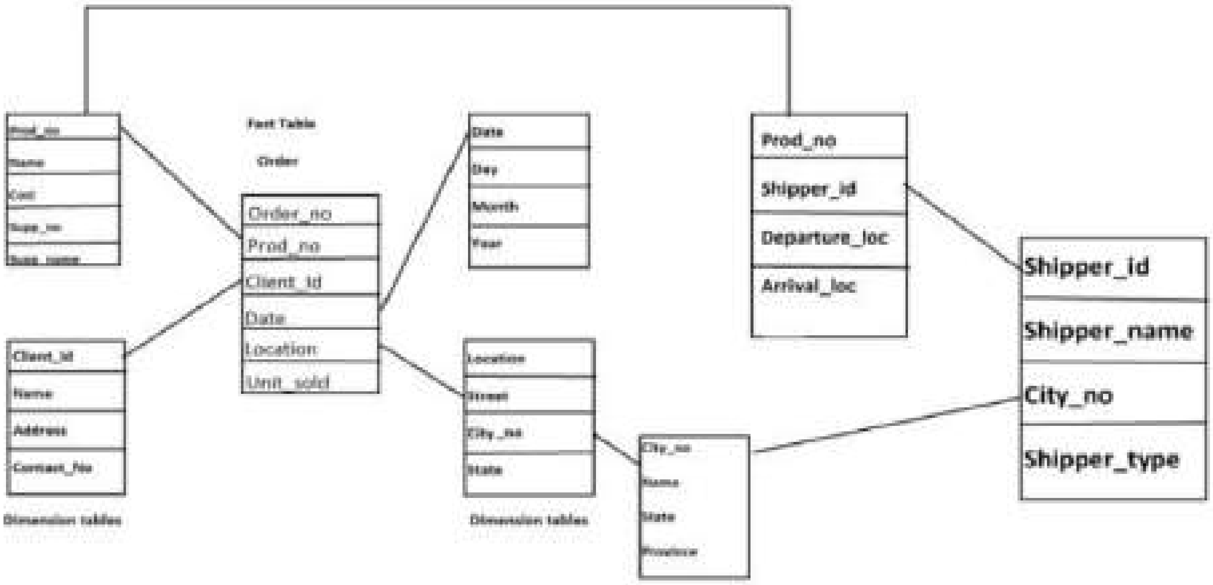


Fig 6. Fact constellation

Multidimensional Data model

A multidimensional model views data in the form of a data-cube. A data cube enables data to be modeled and viewed in multiple dimensions. It is defined by dimensions and facts.

The dimensions are the perspectives or entities concerning which an organization keeps records. For example, a shop may create a sales data warehouse to keep records of the store's sales for the dimension time, item, and location. These dimensions allow the save to keep track of things, for example, monthly sales of items and the locations at which the items were sold. Each dimension has a table related to it, called a dimensional table, which describes the dimension further. For example, a dimensional table for an item may contain the attributes item_name, brand, and type.

A multidimensional data model is organized around a central theme, for example, sales. This theme is represented by a fact table. Facts are numerical measures. The fact table contains the names of the facts or measures of the related dimensional tables.

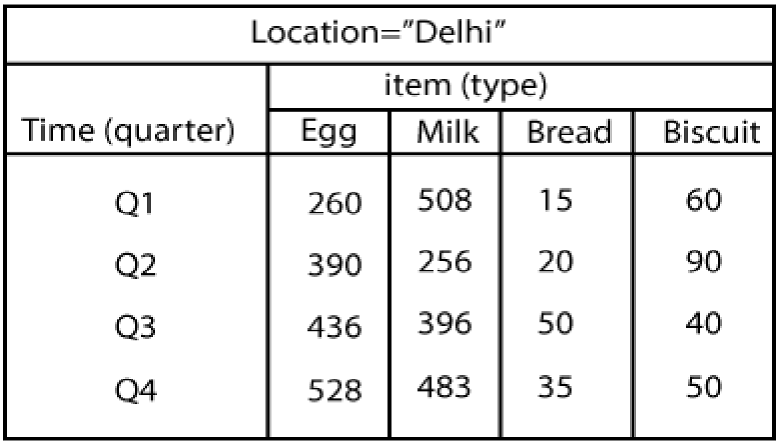
Multidimensional data model - From Tables and Spreadsheets to Data Cubes

A data warehouse is based on a multidimensional data model which views data in the form of a data cube

A data cube, such as sales, allows data to be modeled and viewed in multiple dimensions

Sales table has dimension **time, item and location.**

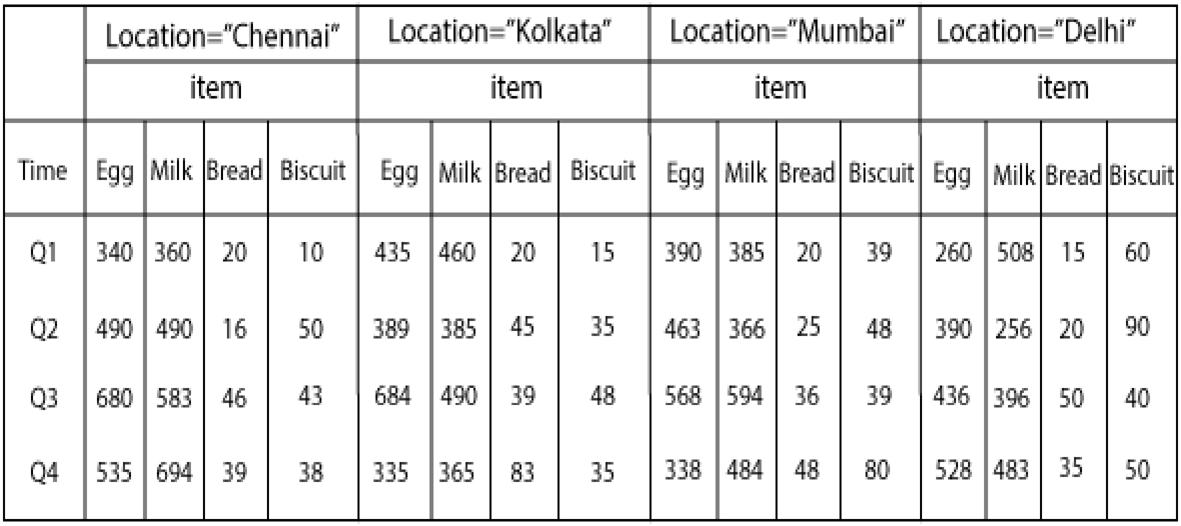
Consider the data of a shop for items sold per quarter in the city of Delhi. The data is shown in the table. In this 2D representation, the sales for Delhi are shown for the time dimension (organized in quarters) and the item dimension (classified according to the types of an item sold). The fact or measure displayed in rupee_sold (in thousands).



Location="Delhi"				
	item (type)			
Time (quarter)	Egg	Milk	Bread	Biscuit
Q1	260	508	15	60
Q2	390	256	20	90
Q3	436	396	50	40
Q4	528	483	35	50

Now, if we want to view the sales data with a third dimension, For example, suppose the data according to time and item, as well as the location is considered for the cities Chennai, Kolkata, Mumbai, and Delhi. These 3D data are shown in the table. The 3D data of the table are represented as a series of 2D tables.

3-D data cube representation in table

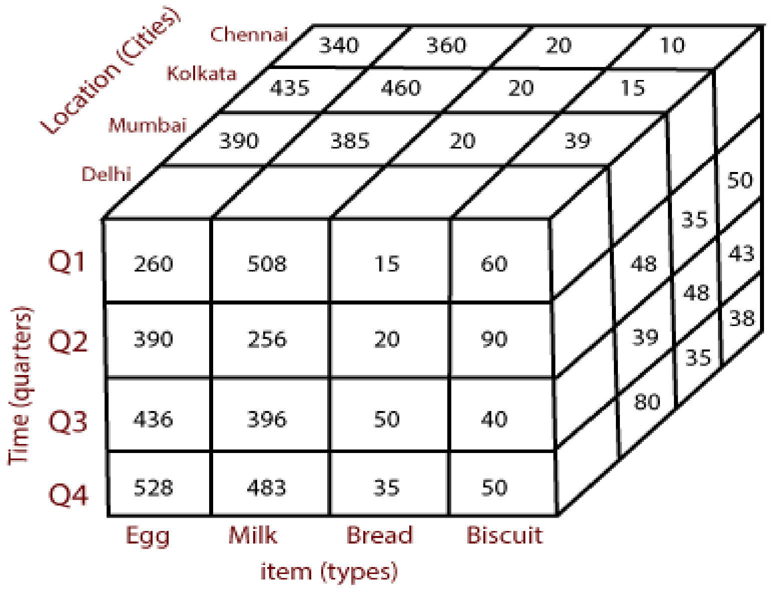


	Location="Chennai"				Location="Kolkata"				Location="Mumbai"				Location="Delhi"			
	item				item				item				item			
Time	Egg	Milk	Bread	Biscuit	Egg	Milk	Bread	Biscuit	Egg	Milk	Bread	Biscuit	Egg	Milk	Bread	Biscuit
Q1	340	360	20	10	435	460	20	15	390	385	20	39	260	508	15	60
Q2	490	490	16	50	389	385	45	35	463	366	25	48	390	256	20	90
Q3	680	583	46	43	684	490	39	48	568	594	36	39	436	396	50	40
Q4	535	694	39	38	335	365	83	35	338	484	48	80	528	483	35	50

Conceptually, it may also be represented by the same data in the form of a 3D data cube, as shown in fig:

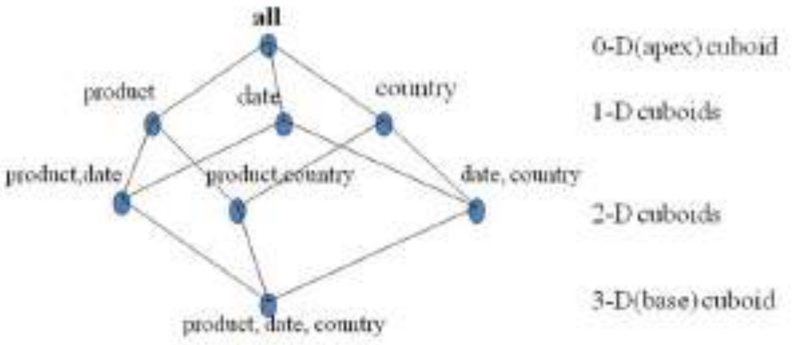
Multidimensional Data

Sales volume as a function of product, month, and region

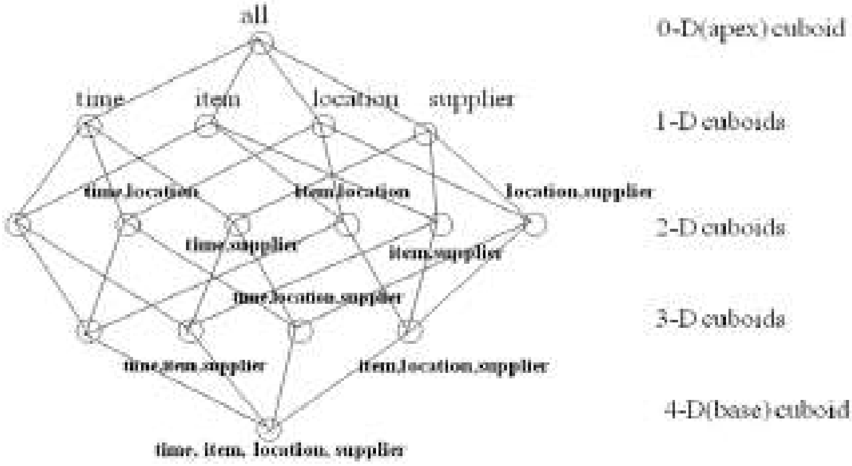


- We may display n-D data as series of (n-1)-D cubes.
- We may construct a lattice of cubiods, each showing the data at different level of summarization, or group by
- In data warehousing literature, an n-D base cube is called a base cuboid.
- The lattice of cuboids forms a data cube.
- The cubiod that holds the lowest level of summarization is called base cubiod
- The top most 0-D cuboid, which holds the highest-level of summarization, is called the apex cuboid.

Cuboids Corresponding to the Cube



Cube: A Lattice of Cuboids



MULTIDIMENSIONAL MODEL Vs ENTITY RELATIONSHIP MODEL

ER is a logical design technique that seeks to remove the redundancy in data. This coupled with normalization of data enables easy maintainability and improves data integrity which is a necessity for transaction processing applications. End user comprehension and the data retrieval are major show stoppers; as such a database is proliferated with dozens of tables that are linked together by a bewildering spider web of joins.

Use of the ER modeling technique defeats the basic allure of data warehousing, namely intuitive and high performance retrieval of data.

MD is a logical design technique that seeks to present the data in a standard, intuitive framework that allows for high-performance access. Every Multidimensional model is composed of one table with a multipart key, called the fact table, and a set of smaller tables called dimension tables.

Each dimension table has a single-part primary key that corresponds exactly to one of the components of the multipart key in the fact table. This characteristic "star-like" structure is often called a star join. Each dimensional table is logical and user identifiable and serves a business purpose by serving as an object of interest to the user. It is also maintained by the ETL process of the data ware housing application .Hence it is considered as an internal Logical file and included in the data function count.

Business Metric

A **Business Metric** is a quantifiable measure that is used to **track, monitor and assess the status of a specific business process.**

Used to track, monitor and assess the success or failure of various business processes. The main goal of measuring business metrics is to track cost management.

It's important to note that business metrics should be employed to address key audiences surrounding a business, such as investors, customers, and different types of employees, such as executives and middle managers.

Every area of business has specific performance metrics that should be monitored – marketers track marketing and social media metrics, such as campaign and program statistics, sales teams monitor sales performance metrics such as new opportunities and leads, and executives look at big picture financial metrics.

Key Performance Indicator (KPI)

Definition

A Key Performance Indicator is a measurable value that demonstrates how effectively a company is achieving key business objectives. Organizations use KPIs at multiple levels to evaluate their success at reaching targets. High-level KPIs may focus on the overall performance of the business, while low-level KPIs may focus on processes in departments such as sales, marketing, HR, support and others.

Examples of KPIs are −

- Sales department of an organization use a KPI to measure monthly gross profit against projected gross profit.
- Accounting department measure monthly expenditures against revenue to evaluate costs.
- Human resources department measure quarterly employee turnover.
- Business professionals frequently use KPIs that are grouped together in a business scorecard to obtain a quick and accurate historical summary of business success or to identify trends.

A Key Performance Indicator (KPI) embodies a strategic objective and measures performance against a goal. KPIs are applied to Business Intelligence (BI) to gauge trends and assess tactical courses of action.

It is common to divide KPIs into two categories: outcomes and drivers.

Outcome: Outcome KPIs measure the output of past activity

Driver: Driver KPIs measure activity in its current and future state.

Note: Not all metrics are KPIs, so it's important to think clearly about what metrics really drive your business.

Types of KPIs

KPIs differ from organization to organization based on business priorities.

For example, one of the key performance indicators for a public company will likely be its stock price, while a KPI for a privately held startup may be the number of new customers added each quarter.

Quantitative indicators that can be presented with a number.

Qualitative indicators that can't be presented as a number.

Leading indicators that can predict the outcome of a process

Lagging indicators that present the success or failure *post hoc*

Input indicators that measure the amount of resources consumed during the generation of the outcome

Process indicators that represent the efficiency or the productivity of the process

Output indicators that reflect the outcome or results of the process activities

Practical indicators that interface with existing company processes.

Directional indicators specifying whether or not an organization is getting better.

Actionable indicators are sufficiently in an organization's control to effect change.

Financial indicators used in performance measurement and when looking at an operating index.

Identifying the KPIs

The first and the most crucial step in KPI analysis is to identify the KPIs that effectively monitor the required trends in the organization. This requires complete understanding of the objectives and requires proper communication channels between the analysts and those who are responsible for fulfilling the objectives.

There are a number of KPIs to choose from, but the success in monitoring relies on the right choice of those that are relevant to the objectives. The KPIs differ from organization to organization and from department to department. It is effective only when they lead to improvement in the performance.

You can evaluate the relevance of a KPI using the SMART criteria, i.e. the KPI should be **S**pecific, **M**easurable, **A**ttainable, **R**elevant and **T**ime-bound. In other words, the KPI chosen should meet the following criteria −

- The KPI reflects your **S**pecific objective.
- The KPI enables you to **M**easure progress towards that goal.
- The goal for which the KPI is being defined is realistically **A**ttainable.
- The goal that the KPI is targeting is **R**elevant to the organization.
- You can set a **T**ime-frame for achieving the goal so that the KPI reveals how near the goal is as compared to the time that is left.

The defined KPIs are to be evaluated from time to time to find their relevance as the time progresses. If required, different KPIs need to be defined and monitored. The KPIs might have to be edited as the time progresses. Only then, your KPI monitoring will be relating to the current organization needs.

What makes a KPI effective?

KPI stands for key performance indicator it is only as valuable as the action it inspires. Too often, organizations blindly adopt industry-recognized KPIs and then wonder why that KPI doesn't reflect their own business and fails to affect any positive change. One of the most important, but often overlooked, aspects of KPIs is that they are a form of communication. As such, they abide by the same rules and best-practices as any other form of communication. Succinct, clear and relevant information is much more likely to be absorbed and acted upon.

In terms of developing a strategy for formulating KPIs, the team should start with the basics and understand what an organizational objectives are, how is our plan on achieving them, and who can act on this information. This should be an iterative process that involves feedback from analysts, department heads and managers. As this fact finding mission unfolds, you will gain a better understanding of which business processes need to be measured with a KPI dashboard and with whom that information should be shared.

Characteristics of effective KPIs

Delivering high-impact KPIs is crucial to an organization’s performance and growth. Here are six characteristics of successful KPIs:

- Simple
- Cascading
- Measurable
- Actionable

- Timely
- Visible

1. Simple

KPIs should be both simple to understand and to measure. Employees must able to know what the KPI is measuring and how it is being calculated. KPIs are also made simple when they are sparse. Focusing on a small number of KPIs enables employees to understand at a deep level what behaviors the KPI is driving and modify the KPI to deliver better results.

2. Cascading

Effective KPIs cascade from strategic dashboards to tactical and operational dashboards. This means that KPIs should trickle down from the overall strategic goals of the organization to the daily operations of the employees that are affecting the KPIs. When KPIs share a link from the C-level to the entry level, they support unified goals and actions.

3. Measurable

KPIs must be measurable for employees to analyze their performance. A KPI doesn’t have to be quantitative to be measurable. For example a qualitative KPI such as “how satisfied are our customers with this product” can be measured through customer feedback surveys.

4. Actionable

Business analytics expert Jay Liebowitz says that an effective KPI is one that “prompts decisions, not additional questions”. When employees clearly understand what they need to do to influence a KPI, they can manage to do it.

5. Timely

The results of KPIs should be reported frequently enough so that employees can make timely decisions but not too frequently so that they are overwhelmed with data. Organizations should consider how urgent, sensitive, accurate, and costly measuring the KPI is before deciding how often to report on it. Additionally they should ensure that the results of a report are being acted on in a timely fashion.

6. **Visible**

Making KPIs visible across an organization communicates to employees how their work is affecting the organization's overall goals. It will incentivize them to work harder and be more productive.

Importance of KPI

Key performance indicators- how well a business is doing. Without KPIs, it would be difficult for a company's leaders to evaluate that in a meaningful way, and to then make operational changes to address performance problems.

Keeping employees focused on business initiatives and tasks that are central to organizational success could also be challenging

In addition to highlighting business successes or issues based on measurements of current and historical performance, KPIs can point to future outcomes, giving executives early warnings on possible business problems or advance guidance on opportunities to maximize return on investment.

Armed with such information, they can manage business operations more proactively, with the potential to gain competitive advantages over less data-driven rivals.

Why are KPI's important?

Not only are company **key performance indicators** critical for monitoring financial performance, they can also help to improve employee morale, customer satisfaction and other, more personal objectives **important** to the growth and success of your business

The reasons why KPIs are important include:

- **Goal measurement:** KPIs are the measurements by which you know if your business is achieving its strategic goals or not.

- **Providing information and feedback:** They provide a simple, insightful snapshot of a company's overall performance, as well as reliable, real-time information for effective decision-making.

- **Education:** KPIs create an atmosphere of learning in an organisation as they generate conversations between staff that can lead to innovation and a better understanding of the business strategy.

- **Staff morale:** Receiving positive feedback or incentives for meeting KPIs can be rewarding and motivating for staff. Without measuring outcomes, quality work can easily be overlooked.
- **Consistency and continuity:** People, priorities, and goals in a business may change over time, but the measurement of a KPI should remain consistent. This is essential for monitoring long-term strategic goals.

How to write and develop KPI

When writing or developing a KPI, you need to consider how that key performance indicator relates to a specific business outcome or objective. Key performance indicators need to be customized to your business situation, and should be developed to help achieve goals. Follow these steps when writing one:

Write a clear objective for each one

Share them with all stakeholders

Review them on a weekly or monthly basis

Make sure they are actionable

Evolve them to fit the changing needs of the business

Check to see that they are attainable (but add a stretch goal)

Update your objectives as needed

Creating Cubes using SSAS (SQL Server Analysis Services)

SSAS:

- SQL Server Analysis Services (SSAS) is the technology from the Microsoft Business Intelligence stack, to develop Online Analytical Processing (OLAP) solutions.
- SSAS to create cubes using data from data marts / data warehouse for deeper and faster data analysis.
- Cubes are multi-dimensional data sources that have dimensions and facts (also known as measures) as its basic constituents. From a relational perspective, dimensions can be thought of as master tables and facts can be thought of as measureable details. These details are generally stored in a pre-aggregated

proprietary format and users can analyze huge amounts of data and slice this data by dimensions very easily.

CUBE: It is a multidimensional object constructed with dimensions and facts in a particular design for taking multidimensional decisions.

The basic concepts of OLAP include:

- Cube
- Dimension table
- Dimension
- Hierarchy
- Level
- Fact table
- Measure
- Schema

Cube:

- ➢ The basic unit of storage and analysis in Analysis Services is the *cube*. A cube is a collection of data that has been aggregated to allow queries to return data quickly. For example, a cube of order data might be aggregated by time period and by title, making the cube fast when you ask questions concerning orders by week or orders by title.
- ➢ Cubes are ordered into *dimensions* and *measures*. The data for a cube comes from a set of staging tables, sometimes called a star-schema database. Dimensions in the cube come from *dimension tables* in the staging database, while measures come from *fact tables* in the staging database.

Dimension tables:

Dimension tables contain a primary key and any other attributes that describe the entities stored in the table. Examples would be a Customers table that contains city, state and postal code information to be able to analyze sales geographically, or a Products table that contains categories and product lines to break down sales figures.

Dimension

Each cube has one or more *dimensions*, each based on one or more dimension tables. A dimension represents a category for analyzing business data. Dimension has a natural hierarchy so that lower results can be "rolled up" into higher results. For example, in a geographical level you might have city totals aggregated into state totals, or state totals into country totals.

Hierarchy

A *hierarchy* can be best visualized as a node tree. A company's organizational chart is an example of a hierarchy. Each dimension can contain multiple hierarchies; some of them are *natural* hierarchies (the parent-child relationship between attribute values occur naturally in the data), others are *navigational* hierarchies (the parent-child relationship is established by developers.)

Level

Each layer in a hierarchy is called a *level*. For example, a week level or a month level in a fiscal time hierarchy, and a city level or a country level in a geography hierarchy.

Fact table

A *fact table* lives in the staging database and contains the basic information that you wish to summarize. The fact tables contain fields for the individual facts as well as foreign key fields relating the facts to the dimension tables.

Measure

Every cube will contain one or more *measures*, each based on a column in a fact table. In the cube of book order information, for example, the measures would be things such as unit sales and profit.

Schema

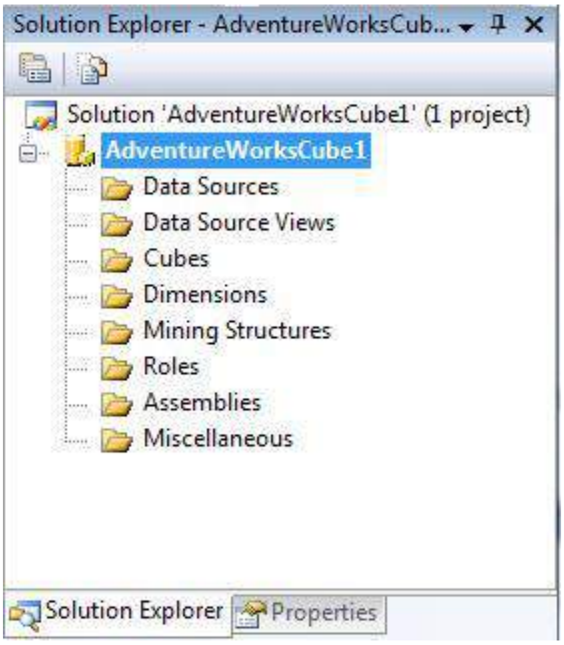
Fact tables and dimension tables are related, which is hardly surprising, given that you use the dimension tables to group information from the fact table. The relations within a cube form a *schema*. There are two basic OLAP schemas: star and snowflake. In a *star schema*, every dimension table is related directly to the fact table. In a *snowflake schema*, some dimension tables are related indirectly to the fact table. For example, if your cube includes Order Details as a fact table, with Customers and Orders as dimension tables, and Customers is related to Orders, which in turn is related to Order Details, then you are dealing with a snowflake schema.

Steps involved in creating a data cube:

Creating a New Analysis Services Project:

To create a new Analysis Services project, follow these steps:

1. Select Microsoft SQL Server 2008 > SQL Server Business Intelligence Development Studio from the Programs menu to launch Business Intelligence Development Studio.
2. Select File > New > Project.
3. In the New Project dialog box, select the Business Intelligence Projects project type.
4. Select the Analysis Services Project template.
5. Name the new project AdventureWorksCube1 and select a convenient location to save it.
6. Click OK to create the new project.

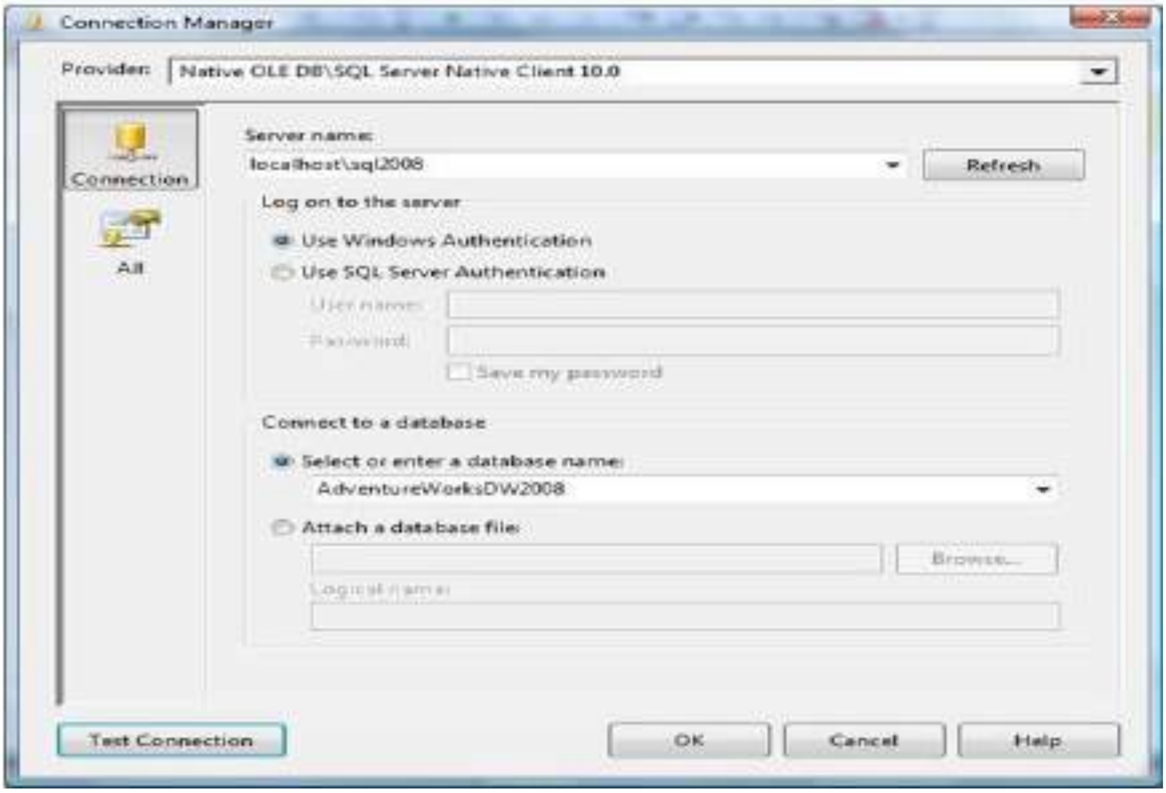


Defining a Data Source:

A data source provides the cube's connection to the staging tables, which the cube uses as source data. To define a data source, you will use the Data Source Wizard. You can launch this wizard by right clicking on the Data Sources folder in your new Analysis Services project. The wizard will walk you through the process of defining a data source for your cube, including choosing a connection and specifying security credentials to be used to connect to the data source.

To define a data source for the new cube, follow these steps:

- Right-click on the Data Sources folder in Solution Explorer and select New Data Source.
- Read the first page of the Data Source Wizard and click next.
- You can base a data source on a new or an existing connection. Because you do not have any existing connections, click New.
- In the Connection Manager Dialog box, select the server containing your analysis services sample database from the Server Name combo box.
- Fill in your authentication information.
- Select the Native OLE DB\SQL Native Client provider (this is the default provider).
- Select the AdventureWorksDW2008 database



Connection Manager

Provider: Native OLE DB\SQL Server Native Client 10.0

Server name: localhost\sql2008

Log on to the server

● Use Windows Authentication

○ Use SQL Server Authentication

User name:

Password:

□ Save my password

Connect to a database

● Select or enter a database name: AdventureWorksDW2008

○ Attach a database file:

Logical name:

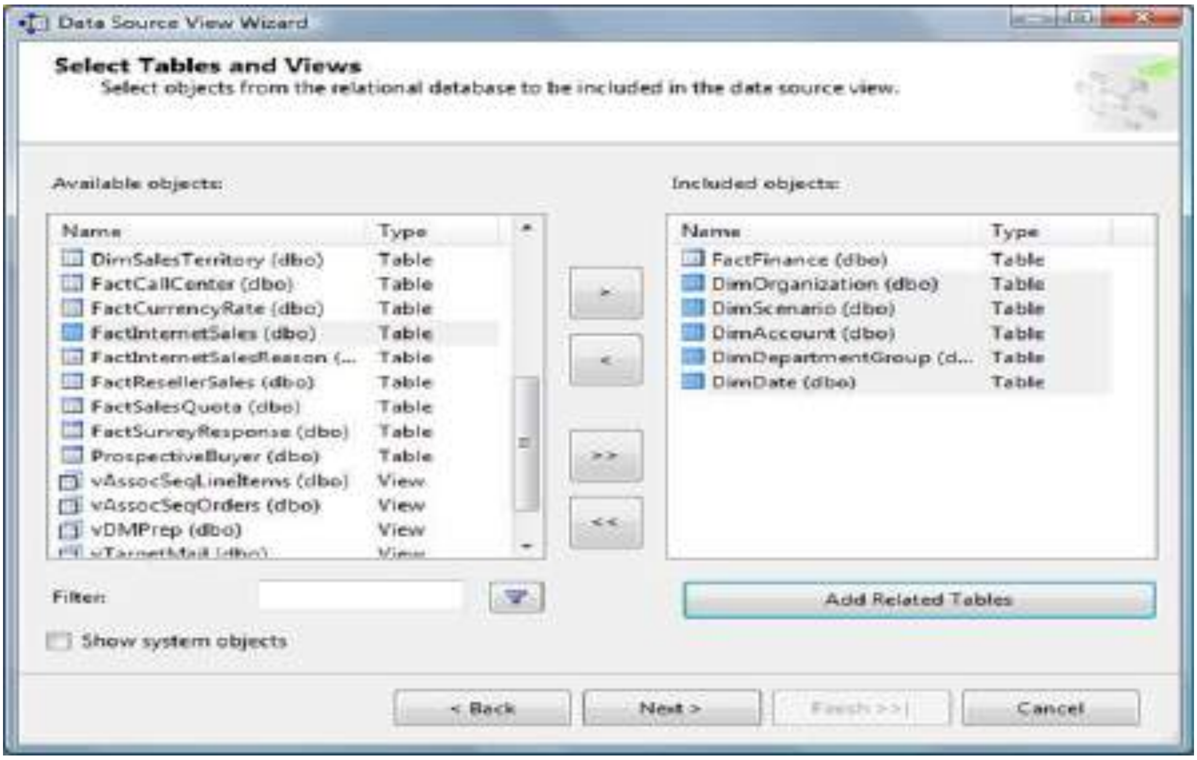
- Click OK to dismiss the Connection Manager Dialog box.
- Click Next.
- Select Use the Service Account impersonation information and click next.
- Accept the default data source name and click Finish.

Defining a Data Source View:

A data source view is a persistent set of tables from a data source that supply the data for a particular cube. It lets you combine tables from as many data sources as necessary to pull together the data your cube needs. BIDS also includes a wizard for creating data source views, which you can invoke by right clicking on the Data Source Views folder in Solution Explorer.

To create a new data source view, follow these steps:

- Right-click on the Data Source Views folder in Solution Explorer and select New Data Source View.
- Read the first page of the Data Source View Wizard and click Next.
- Select the Adventure Works DW data source and click Next. Note that you could also launch the Data Source Wizard from here by clicking New Data Source.
- Select the FactFinance(dbo) table in the Available Objects list and click the > button to move it to the Included Object list. This will be the fact table in the new cube.
- Click the Add Related Tables button to automatically add all of the tables that are directly related to the dbo.FactFinance table. These will be the dimension tables for the new cube



Data Source View Wizard

Select Tables and Views

Select objects from the relational database to be included in the data source view.

Available objects:

Name	Type
DimSalesTerritory (dbo)	Table
FactCallCenter (dbo)	Table
FactCurrencyRate (dbo)	Table
FactInternetSales (dbo)	Table
FactInternetSalesReason (...	Table
FactResellerSales (dbo)	Table
FactSalesQuota (dbo)	Table
FactSurveyResponse (dbo)	Table
ProspectiveBuyer (dbo)	Table
vAssocSeqLineItems (dbo)	View
vAssocSeqOrders (dbo)	View
vDMPrep (dbo)	View
vTargetMail (dbo)	View

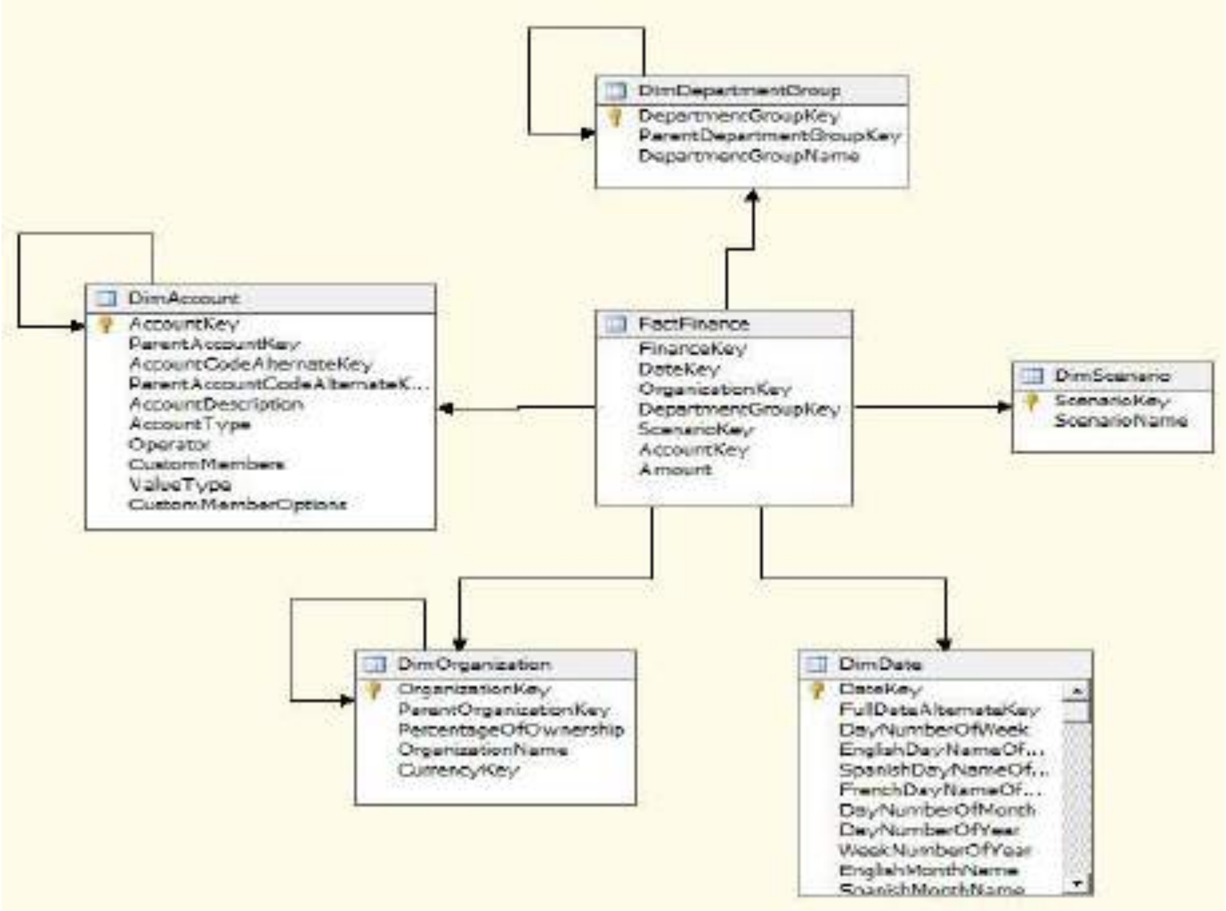
Included objects:

Name	Type
FactFinance (dbo)	Table
DimOrganization (dbo)	Table
DimScenario (dbo)	Table
DimAccount (dbo)	Table
DimDepartmentGroup (d...	Table
DimDate (dbo)	Table

Filter:

□ Show system objects

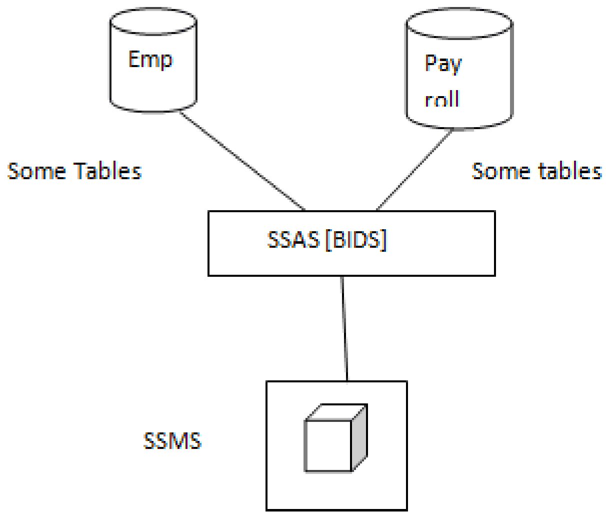
- Click Next.
- Name the new view Finance and click Finish. BIDS will automatically display the schema of the new data source view



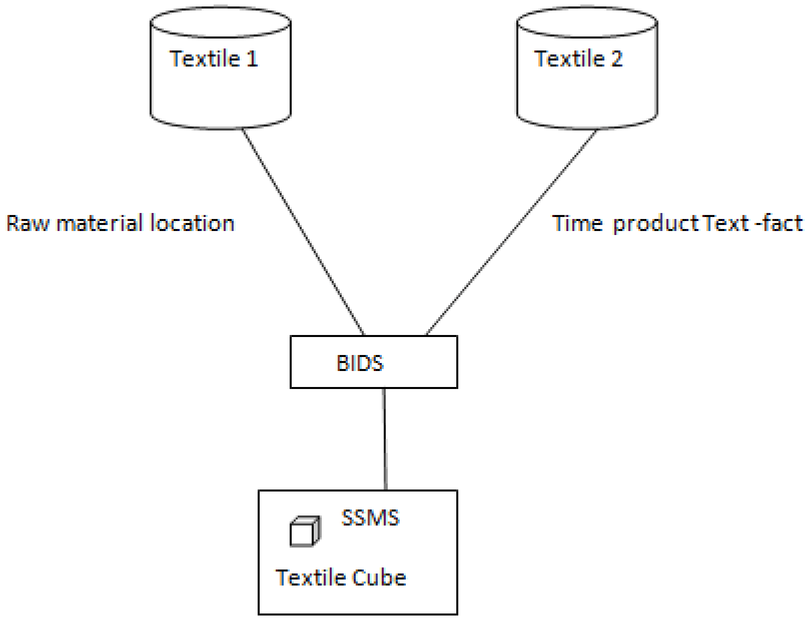
CUBE Creation

Steps:

1. Open BIDS
2. Create data source
3. Create data source view
4. Provide relationship between dimensions and facts
5. Create a cube
6. Manipulate the components (action, KPI ….. ETC)
7. Deploy the cube
8. Browser cube (or) perform re conclusion (or) unit testing



Class Room Example:



1. OpenBIDS
2. File → new → project → template → analysis → service project → project name
   Name:TEXTILE_CUBE
   Location:C:Documentsand settingsvinayaka
   Solution name: TEXTILE_CUBE
3. View → solution explorer
4. Create two data sources DS_textile 1
   DS_textile R with the below procedure
   Data sources → RC → New data source → Next → New → SERVER NAME →

LOCAL HOST
Select or enter database name
LOCAL HOST: TEST TLES → OK → NEXT
Inherit → Data source name: DS_Test tiles 1 Finish
Like this create another data source DS_Text tiles 2

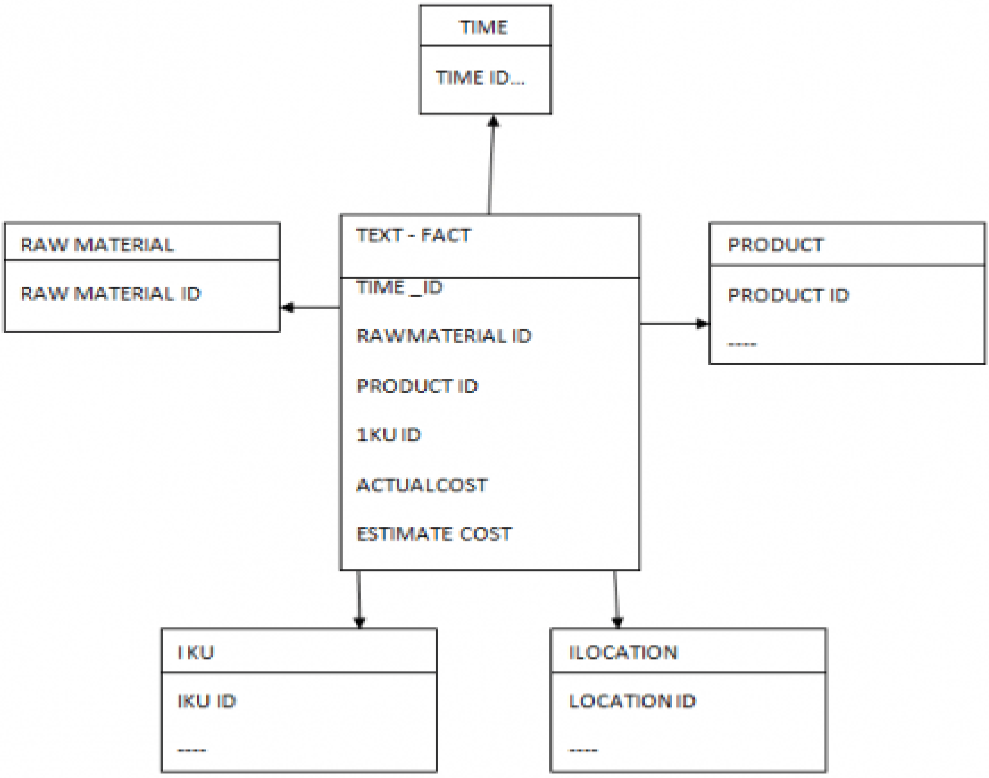
5. Data source views → RC → new data source view → next →
Relational data sources
DS_text tiles 1 → select → next –>
Create logical relationships by matching columns
↓
Next → selects available objects
RAW MATERIAL LOCATION 1ku >RAW MATERIAL LOCATION 1 KU
↓
Next
NAME: DSV_TEXT → FINISH

6. Go to DSV_CUBE_DB, For taking remaining (Time, Product, Text fact) tables into it follow this process DSV_CUBE_DB DESIGN → ADD/REMOVE table
↓
Data source: DS_text tiles 2
Available objects: Included object

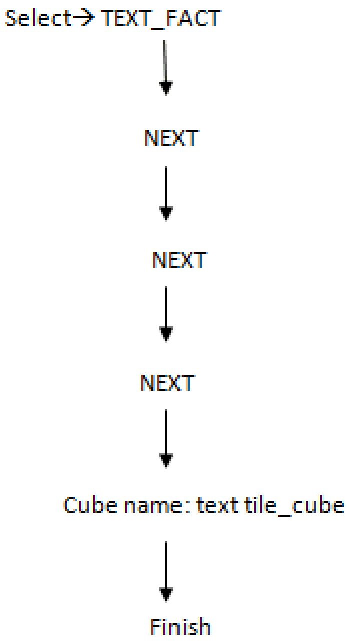
TIME		TIME
PRODUCT	>	PRODUCT
TEXT_FACT		TEXT FACT

↓
CLICK OK

7. Provide relationship between fact table to remaining dimension tables by dragging and drop column mappings from fact table columns to dimension column
While connecting from fact column to dimension column it displays a message, click ok
The destination table of the newly created relationship had no primary key defined. Would you like to define a logical primary key based on the column used in this relationship?
↓
Yes
After all dimensions column connections
DS_Cube_DB → RC → arrange tables, then it looks like this



8. CUBES → RC → NEW CUBE → NEXT → Using Existing tables → next → Measure group tables



Now various tabs opened and we can see the cube structure as well
Build → Deploy → TEST the cube.