Data Warehousing and Data Mining
TBC-604(1)

Syllabus

UNIT 1 : **Introduction to Data Warehousing**

Concept of Data Warehouse, DBMS verses data warehouse, Data Marts, Metadata, Multidimensional data model, Multidimensional database, Data warehouse Measures, their categorization and computation, Multi-dimensional database hierarchies.

UNIT 2 : Data Warehouse Architecture

Operations in OLAP, Advantages of OLAP over OLTP, Three-Tier Data Warehouse architecture, OLAP Guidelines, Multidimensional versus Multirelational OLAP , Categories of Tools, OLAP Tools and the Internet

Unit 3: Introduction to Data Mining

Basic Concepts of Data Mining; Data Mining primitives: Task-relevant data, mining objective, measures and identification of patterns, KDD versus data mining, data mining tools and applications.

Data Mining Query Languages: Data specification, specifying kind of knowledge, hierarchy specification, pattern presentation & visualization specification, data mining languages and standardization of data mining, Architectures of Data Mining Systems.

Syllabus

UNIT 4 : **Data Mining Techniques**

Association rules: Association rules from transaction database & relational database, correlation analysis; Classification and predication using decision tree induction. Introduction to Clustering techniques, partition method, and Hierarchical method.

UNIT 5 : Overview of Advanced Features of Data Mining

Mining complex data objects, Spatial databases, Multimedia databases, Time series and Sequence data; mining Text Databases and mining Word Wide Web.

What Is a Data Warehouse?

A single, complete and consistent store of data obtained from a variety of different sources made available to end users in a what they can understand and use in a business context.

The data warehouse is an informational environment

Provides an integrated and total view of the enterprise

Makes the enterprise's current and historical information easily available for decision making

Makes decision-support transactions possible without hindering operational systems

Renders the organization's information consistently

Presents a flexible and interactive source of strategic information

- “A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management’s decision-making process.”—W. H. Inmon

- Data warehousing:
  - The process of constructing and using data warehouses

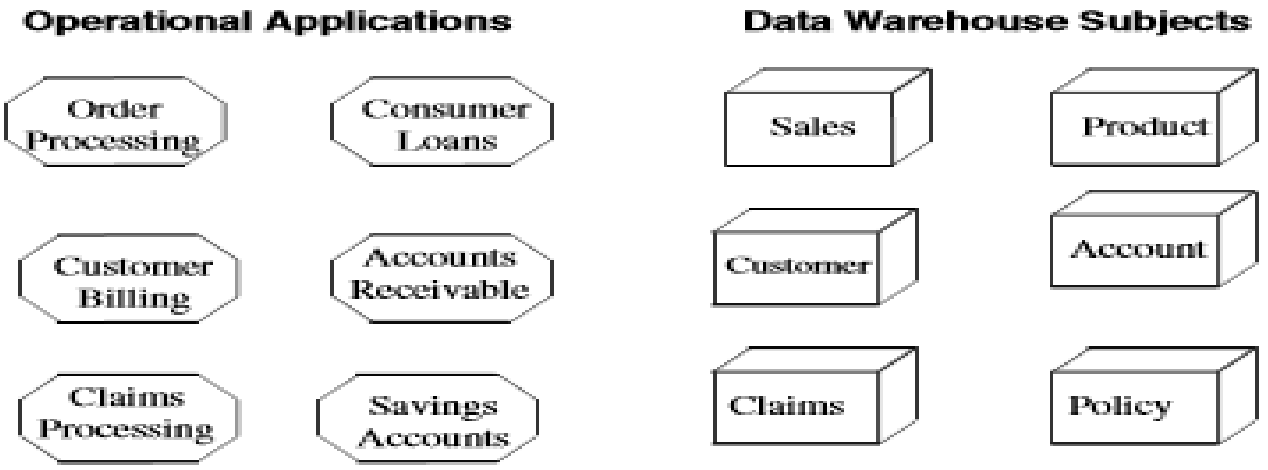
- “A data warehouse is a subject-oriented, integrated, time-variant, and nonvolatile collection of data in support of management’s decision-making process.”—W. H. Inmon

- Data warehousing:
  - The process of constructing and using data warehouses

Features of Data Warehousing

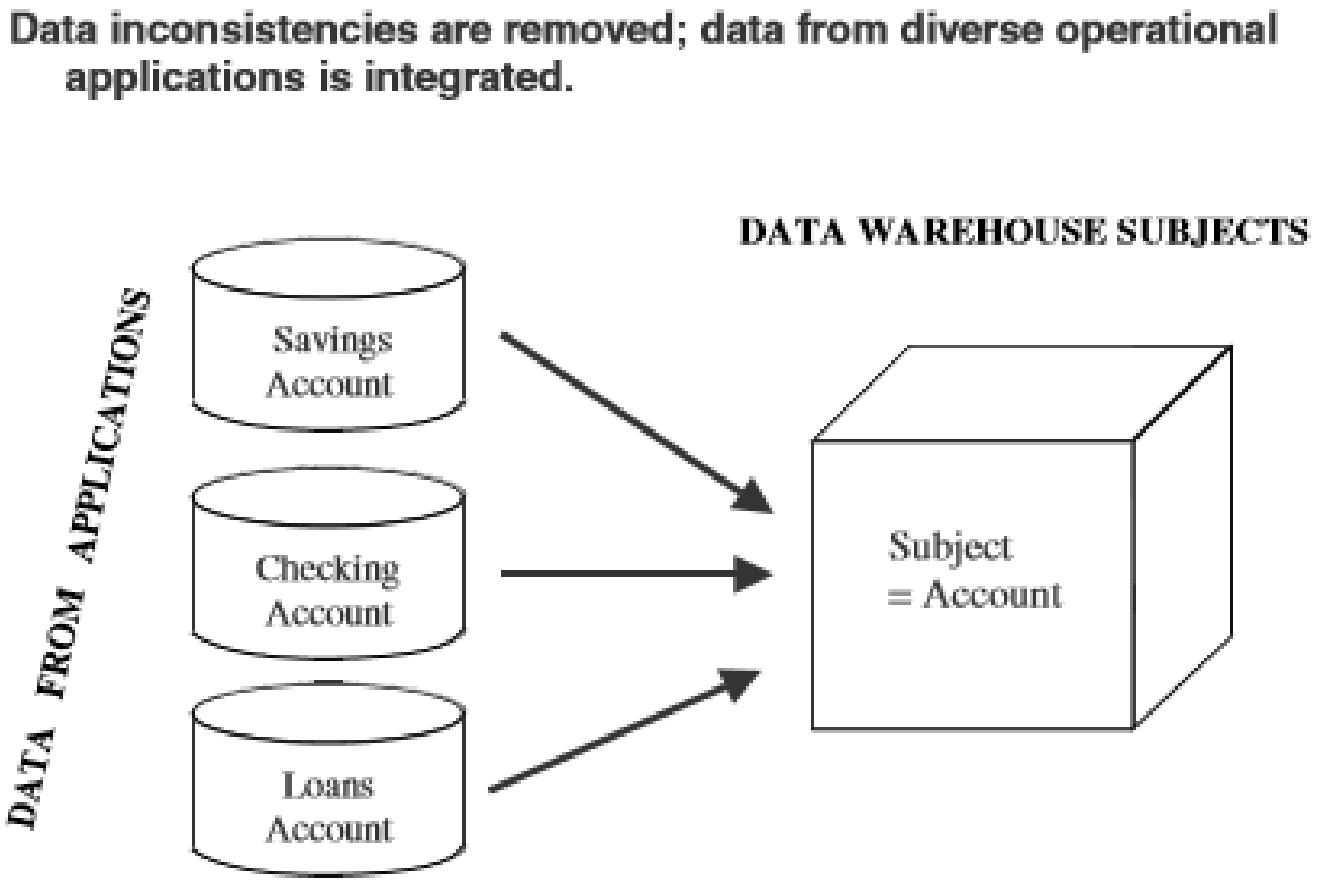
- Subject Oriented: The Data Warehouse is Subject Oriented because it provides us the information around a subject rather the organization's ongoing operations.
- These subjects can be product, customers, suppliers, sales, revenue etc.
- The data warehouse does not focus on the ongoing operations Rather it focuses on modelling and analysis of data for decision making.

In the data warehouse, data is not stored by operational applications, but by business subjects.



Claims under automobile insurance policies are processed in the Auto Insurance application, similarly claim for compensation of workers is organized in workers' compensation application.

- **Integrated:-**The data in DWH is integrated in the sense that data from the disparate system as well as the data from different application will be accumulated and will be stored in the single database called DWH(This process is known as ETL in DWH.) for analysis and decision taking purpose.



Time variant:

- In OLTP the data will be stored according for current values unlike in Data Warehouse, the data will be stored for time which may range from 1 year to 10 years.
- This aspect of data warehouse is quite significant for both the design and implementation phase.

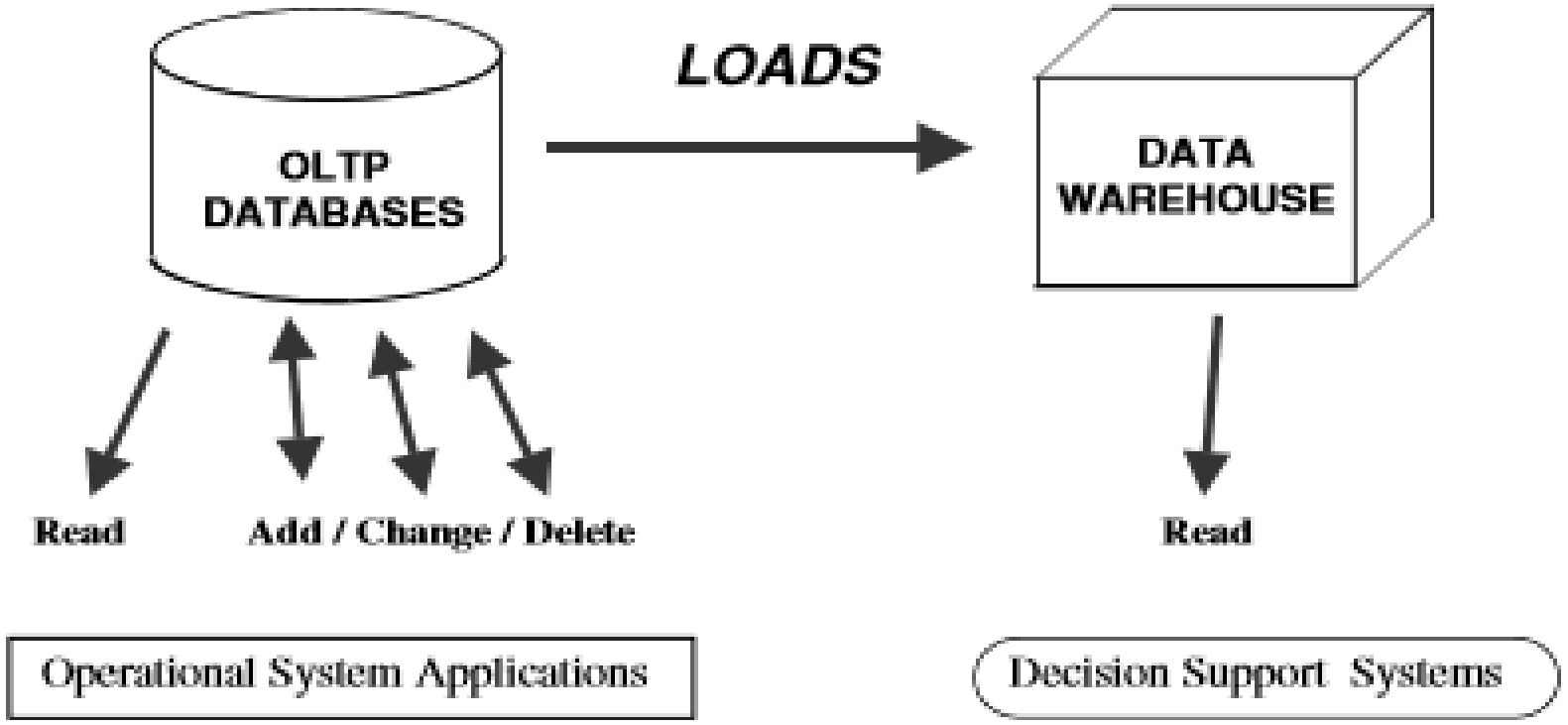
- It helps in analysis of the past, correlates with the present scenario and to predict for the future.

Non-Volatile:

We add change or delete the data from an operational system as each transaction happens.

- We do not delete the data from the data warehouse in the real time.
- Once the data is captured in the data warehouse, we do not run individual transactions to change the data there.
- Data warehouse can only be viewed.
- DWH is used for analysing the data so it periodically get refreshed by picking up the data from various OLTP systems.

Usually the data in the data warehouse is not updated or deleted.



Data warehouse	Operational system
Subject oriented	Transaction oriented
Large (hundreds of GB up to several TB)	Small (MB up to several GB)
Historic data	Current data
De-normalized table structure (few tables, many columns per table)	Normalized table structure (many tables, few columns per table)
Batch updates	Continuous updates
Usually very complex queries	Simple to complex queries

Comparison between OLTP and OLAP system

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

Data Warehouse and Data Mart

- A data mart contains a subset of corporate wide data that is of value to a specific group of users.
- The scope is confined to specific selected subjects.
- The data contained in data marts tend to be summarized.
- Data marts are usually implemented on low-cost departmental servers that are Unix/Linux- or Windows-based. The implementation cycle of a data mart is more likely to be measured in weeks rather than months or years.
- However, it may involve complex integration in the long run if its design and planning were not enterprise-wide

Depending on the source of data, data marts can be categorized as independent or dependent..

Independent Data mart:-Independent data marts are sourced from data captured from one or more operational systems or external information providers, or from data generated locally within a particular department or geographic area.

Dependent data mart:- Dependent data marts are sourced directly from enterprise data warehouses.

Data Warehouse and Data Mart

S.no	Data warehouse	DataMart
1.	Corporate/Enterprise Wide	Departmental
2.	It is a unit of all data marts	It is a single business process
3	Structure for corporate view of data	Structure to suit the departmental view of data
4	Queries on presentation resource	Technology optimal for data access and analysis
5	Data received from staging area Organized on E-R Model	Star-join(Facts & Dimensions)

Metadata

Metadata are data about data. When used in a data warehouse, metadata are the data that define warehouse objects

Metadata in a data warehouse contains the answers to questions about the data in the data warehouse. Metadata are data about data when used in a date warehouse, metadata are the data that define warehouse objects .

Categories of Metadata

1. **Operational Metadata** - Operational metadata contains all of this information about the operational data sources.

2. **Extraction and Transformation Metadata** – Extraction and transformation metadata contain data about the extraction of data from the source systems, namely, the extraction frequencies, extraction methods, and business rules for the data extraction. It also contain information about all the transformation that take place in the data staging area

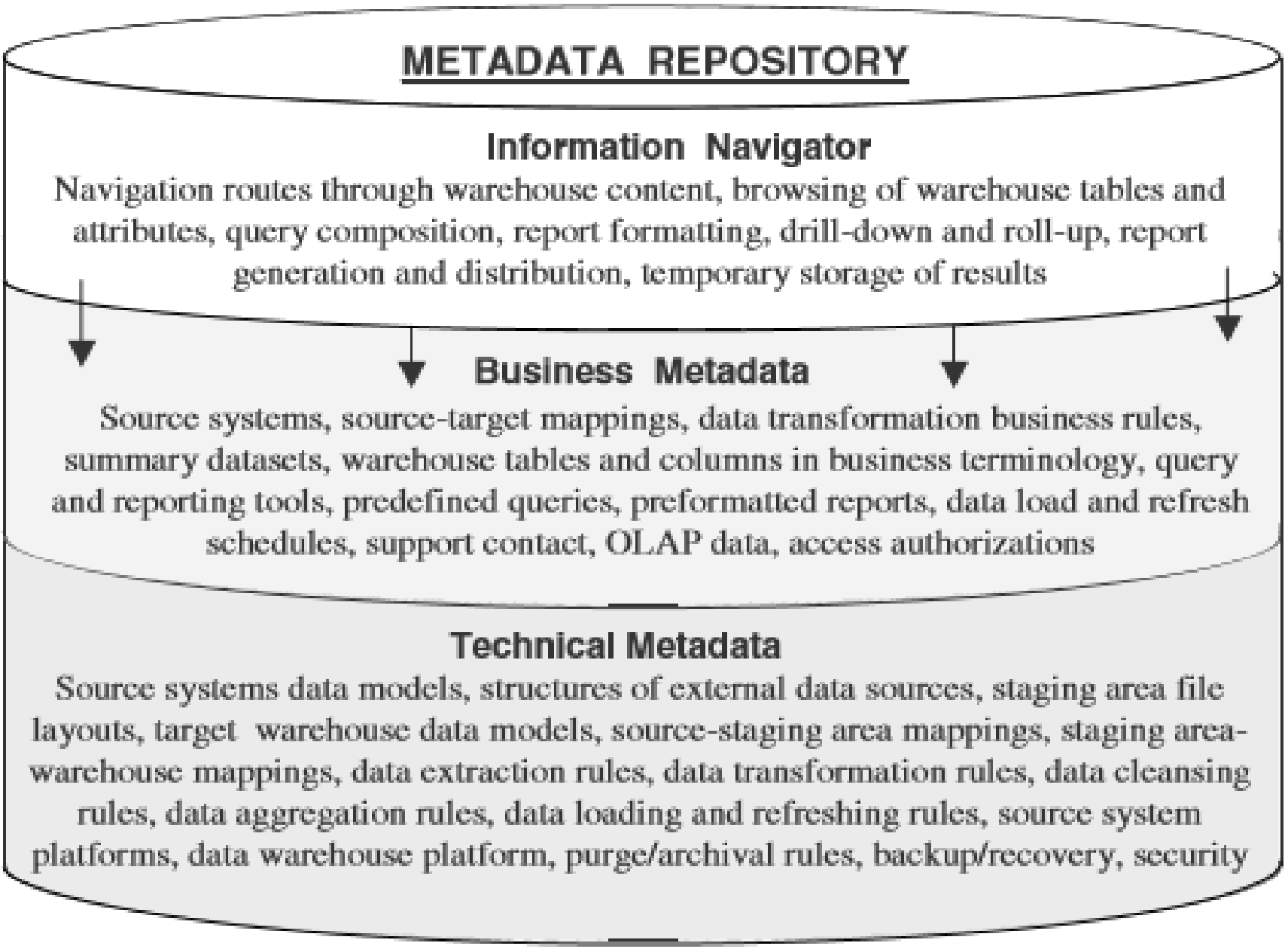
3. **End-User Metadata** – It is the navigation map of the data warehouse. It enable the end users to find information from the data warehouse. The end user metadata allows the end-users to use their own business terminology and look for information in those ways in which they normally think of the business.

Metadata Repository: Metadata repository as a general-purpose information directory or cataloguing device to classify, store, and manage metadata.

Business metadata and **technical metadata** serve different purposes. The end-users need the business metadata; data warehouse developers and administrators require the technical metadata.

Therefore, the metadata repository can be thought of as two distinct information directories, one to store business metadata and the other to store technical metadata.

This division may also be logical within a single physical repository.


METADATA REPOSITORY
Information Navigator
Navigation routes through warehouse content, browsing of warehouse tables and attributes, query composition, report formatting, drill-down and roll-up, report generation and distribution, temporary storage of results
Business Metadata
Source systems, source-target mappings, data transformation business rules, summary datasets, warehouse tables and columns in business terminology, query and reporting tools, predefined queries, preformatted reports, data load and refresh schedules, support contact, OLAP data, access authorizations
Technical Metadata
Source systems data models, structures of external data sources, staging area file layouts, target warehouse data models, source-staging area mappings, staging area-warehouse mappings, data extraction rules, data transformation rules, data cleansing rules, data aggregation rules, data loading and refreshing rules, source system platforms, data warehouse platform, purge/archival rules, backup/recovery, security

Technical Metadata: Technical metadata concentrates on support for the IT staff responsible for development, maintenance, and administration.

Technical metadata is more structured than business metadata.

Technical metadata is like an internal view of the data warehouse showing the inner details in technical terms.

examples of technical metadata:

- Data models of source systems
- Record layouts of outside sources
- Source-to-staging area mappings
- Staging area-to-data warehouse mappings
- Data extraction rules and schedules
- Data transformation rules and versioning
- Data aggregation rules
- Data cleansing rules
- Summarization and derivations
- Data loading and controls
- Job dependencies
- Program names and descriptions
- Data warehouse data model
- Database names
- Table/view names
- Column names and descriptions
- Key attributes
- Business rules for entities and relationships

Business metadata connects your business users to your data warehouse.

Business users need to know what is available in the data warehouse from a perspective different from that of IT professionals like you.

Business metadata is like a roadmap or an easy-to-use information directory showing the contents and how to get there.

Examples

- Connectivity procedures
- Security and access privileges
- The overall structure of data in business terms
- Source systems
- Source-to-target mappings
- Data transformation business rules
- Summarization and derivations
- Table names and business definitions
- Attribute names and business definitions
- Data ownership
- Query and reporting tools
- Predefined queries
- Predefined reports
- Report distribution information
- Common information access routes
- Rules for analysis using OLAP

Multidimensional Data Model

Data warehouses and OLAP tools are based on a multidimensional data model. This model views data in the form of a data cube

Stars, Snowflakes, and Fact Constellations: Schemas for Multidimensional Databases

The entity-relationship data model is commonly used in the design of relational databases, where a database schema consists of a set of entities and the relationships between them.

A data warehouse, however, requires a concise, subject-oriented schema that facilitates on-line data analysis.

The most popular data model for a data warehouse is a multidimensional model. Such a model can exist in the form of a star schema, a snowflake schema, or a fact constellation schema.

Star schema:

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

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

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

The schema graph resembles a starburst, with the dimension tables displayed in a radial pattern around the central fact table.

Snowflake schema:

The snowflake schema is a variant of the star schema model, where some dimension tables are normalized, thereby further splitting the data into additional tables.

The resulting schema graph forms a shape similar to a snowflake.

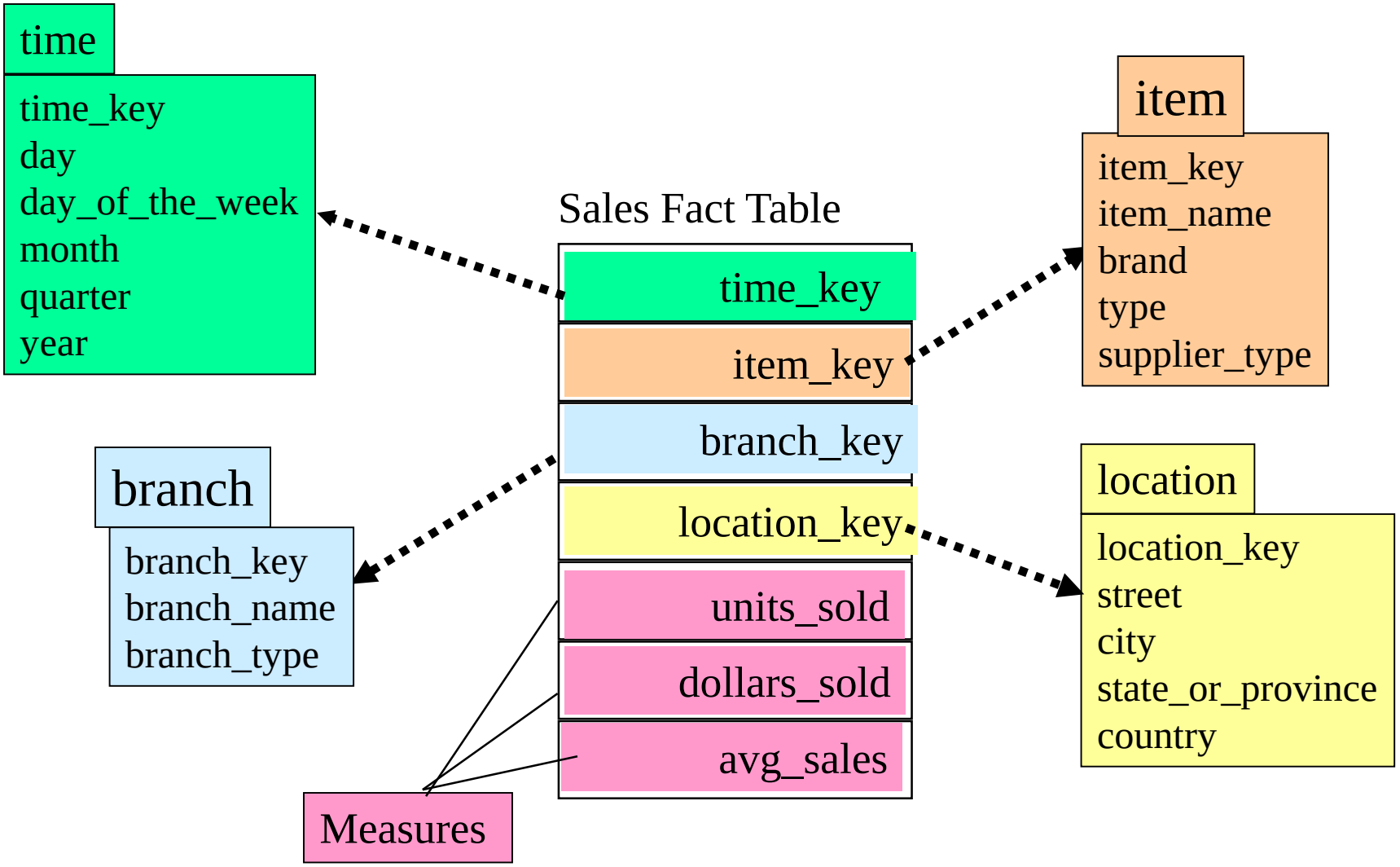
The major difference between the snowflake and star schema models is that the dimension tables of the snowflake model may be kept in normalized form to reduce redundancies.

Such a table is easy to maintain and saves storage space.

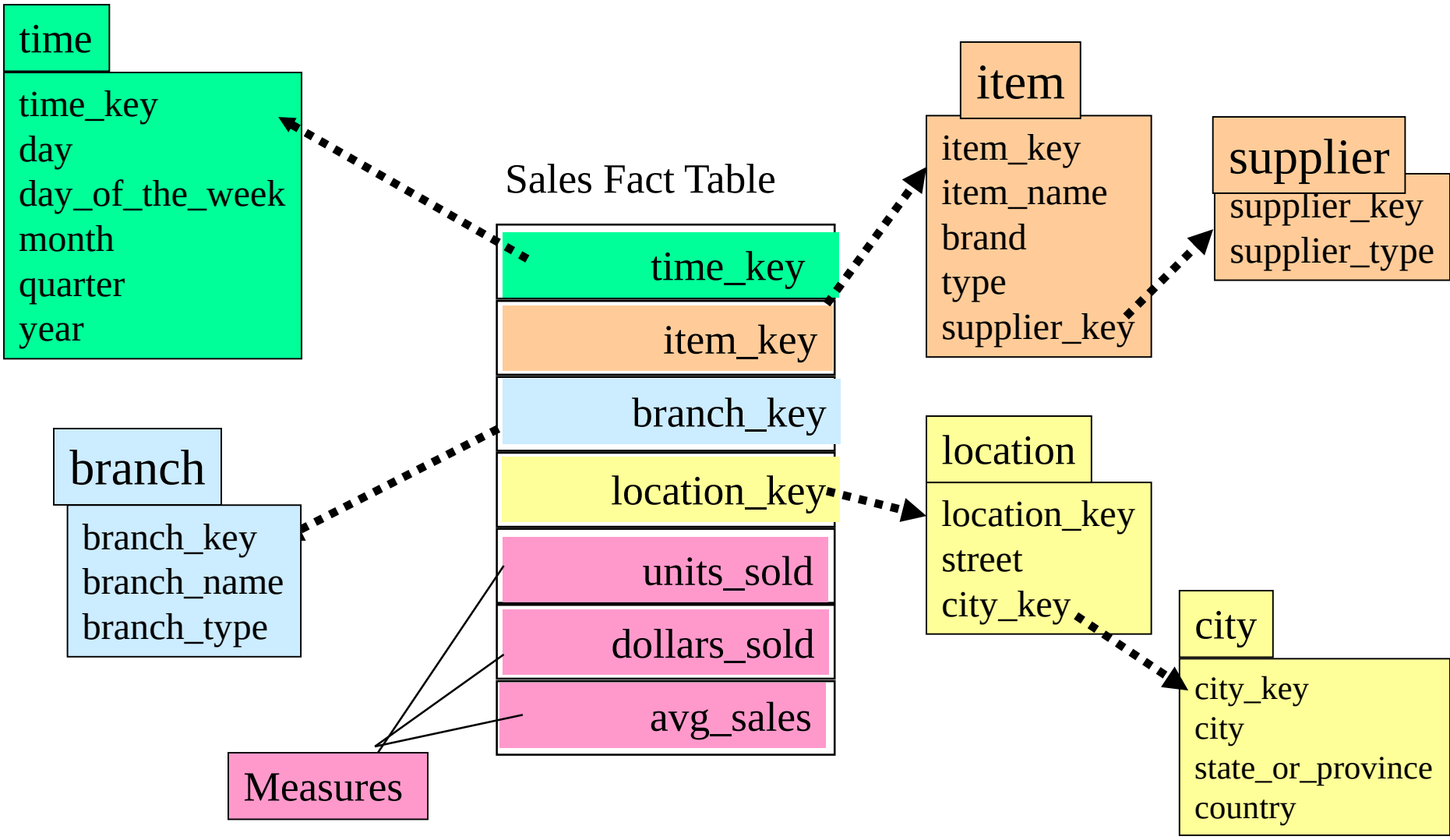
Fact constellation: Sophisticated applications may require multiple fact tables to share dimension tables. This kind of schema can be viewed as a collection of stars, and hence is called a galaxy schema or a fact constellation.

- Modeling data warehouses: dimensions & measures
  - Star schema: A fact table in the middle connected to a set of dimension tables
  - Snowflake schema: A refinement of star schema where some dimensional hierarchy is normalized into a set of smaller dimension tables, forming a shape similar to snowflake
  - Fact constellations: Multiple fact tables share dimension tables, viewed as a collection of stars, therefore called galaxy schema or fact constellation

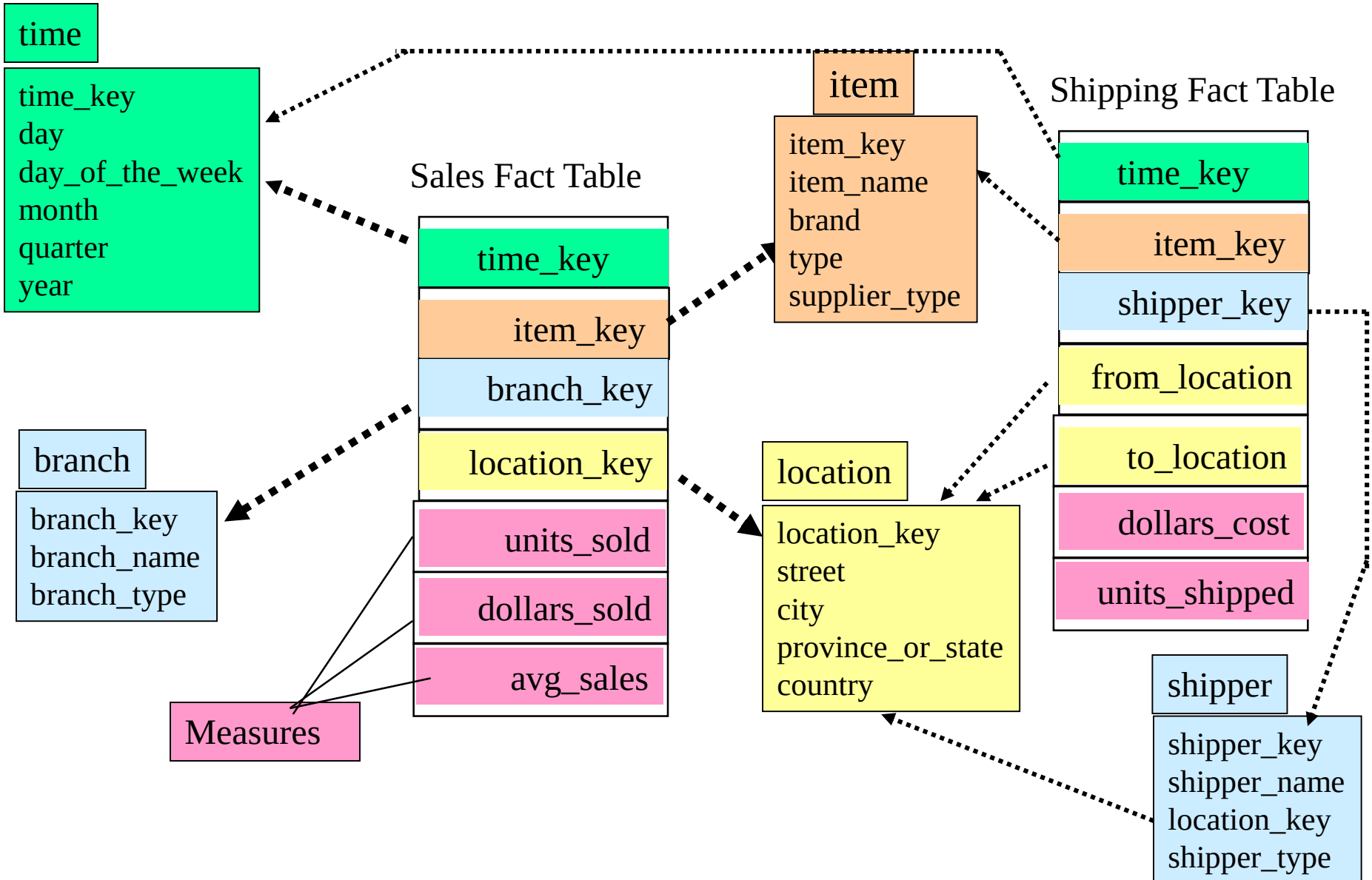
Example of **Star Schema**



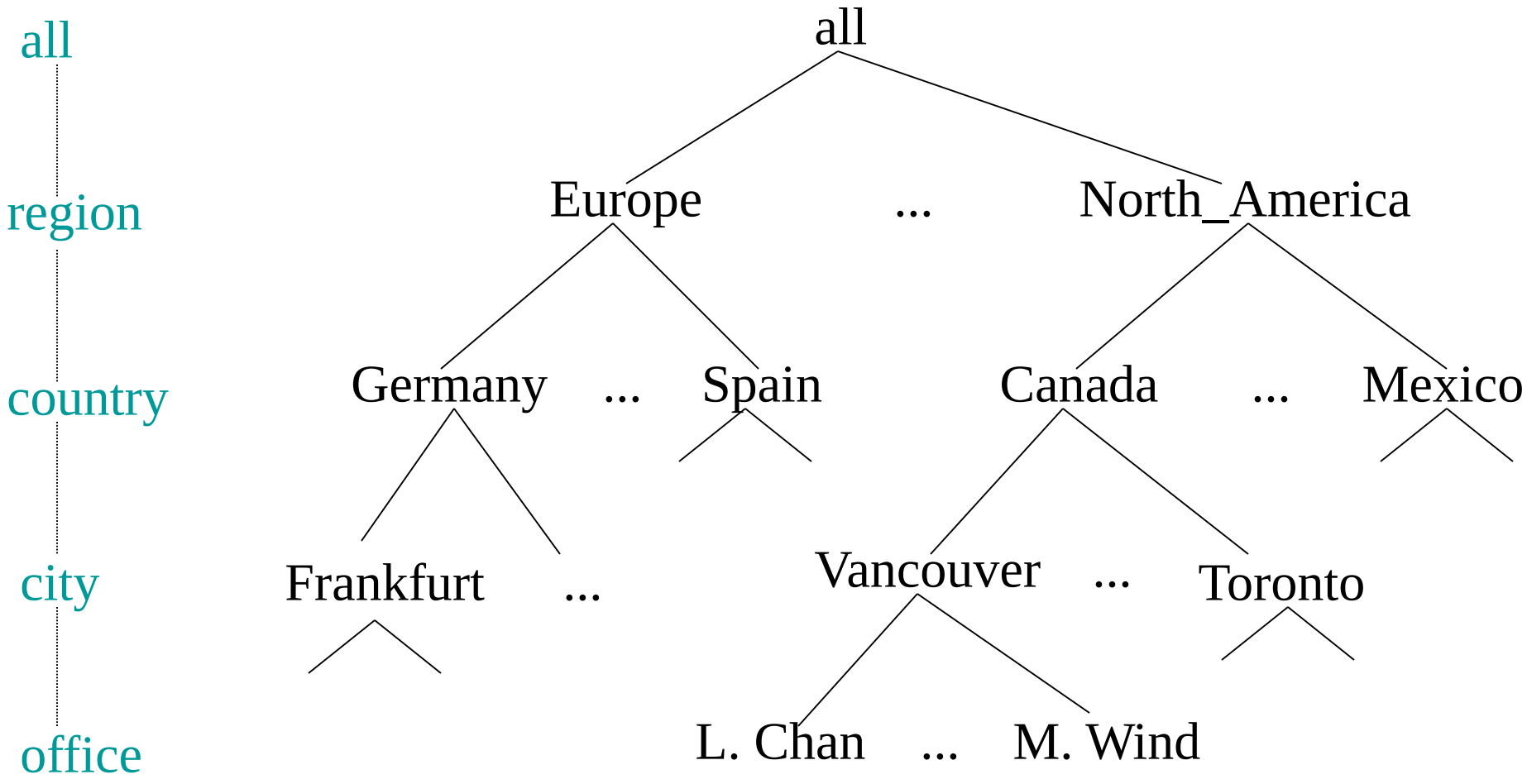
Example of **Snowflake Schema**



Example of **Fact Constellation**



A Concept Hierarchy: **Dimension** (location)



A data cube measure

is a numerical function that can be evaluated at each point in the data cube space. A measure value is computed for a given point by aggregating the data corresponding to the respective dimension-value pairs defining the given point

Measures can be organized into three categories (i.e., distributive, algebraic, holistic), based on the kind of aggregate functions used.

Distributive: An aggregate function is distributive if it can be computed in a distributed manner as follows. Suppose the data are partitioned into n sets. We apply the function to each partition, resulting in n aggregate values. If the result derived by applying the function to the n aggregate values is the same as that derived by applying the function to the entire data set (without partitioning), the function can be computed in a distributed manner. For example, sum() can be computed for a data cube by first partitioning the cube into a set of subcubes, computing sum() for each subcube, and then summing up the counts obtained for each subcube. Hence, sum() is a distributive aggregate function.

Algebraic

An aggregate function is algebraic if it can be computed by an algebraic function with M arguments (where M is a bounded positive integer), each of which is obtained by applying a distributive aggregate function.

For example, avg() (average) can be computed by sum()/count(), where both sum() and count() are distributive aggregate functions.

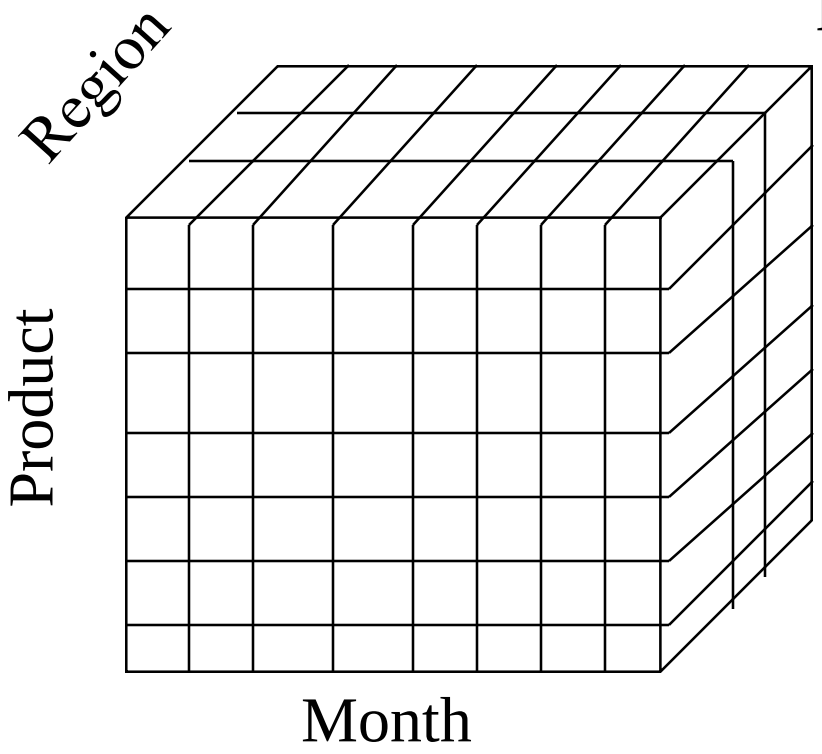
Similarly, it can be shown that min N() and max N() (which find the N minimum and N maximum values, respectively, in a given set) and standard deviation() are algebraic aggregate functions. A measure is algebraic if it is obtained by applying an algebraic aggregate function

Data Cube Measures: Three Categories

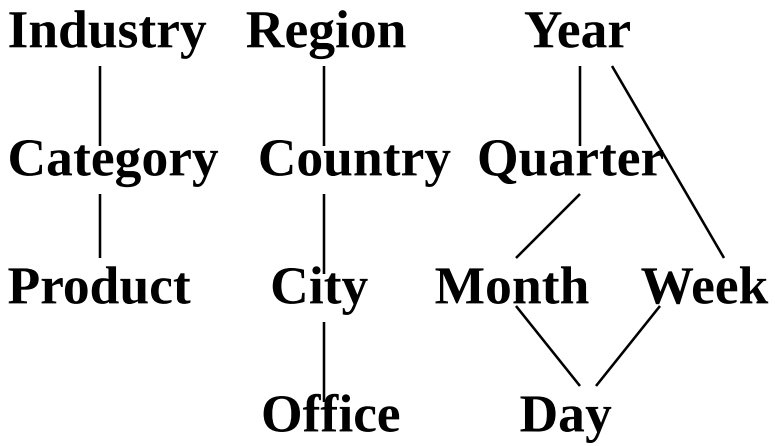
- Distributive: if the result derived by applying the function to *n* aggregate values is the same as that derived by applying the function on all the data without partitioning
  - E.g., count(), sum(), min(), max()
- Algebraic: if it can be computed by an algebraic function with *M* arguments (where *M* is a bounded integer), each of which is obtained by applying a distributive aggregate function
  - E.g., avg(), min_N(), standard_deviation()
- Holistic: if there is no constant bound on the storage size needed to describe a subaggregate.
  - E.g., median(), mode(), rank()

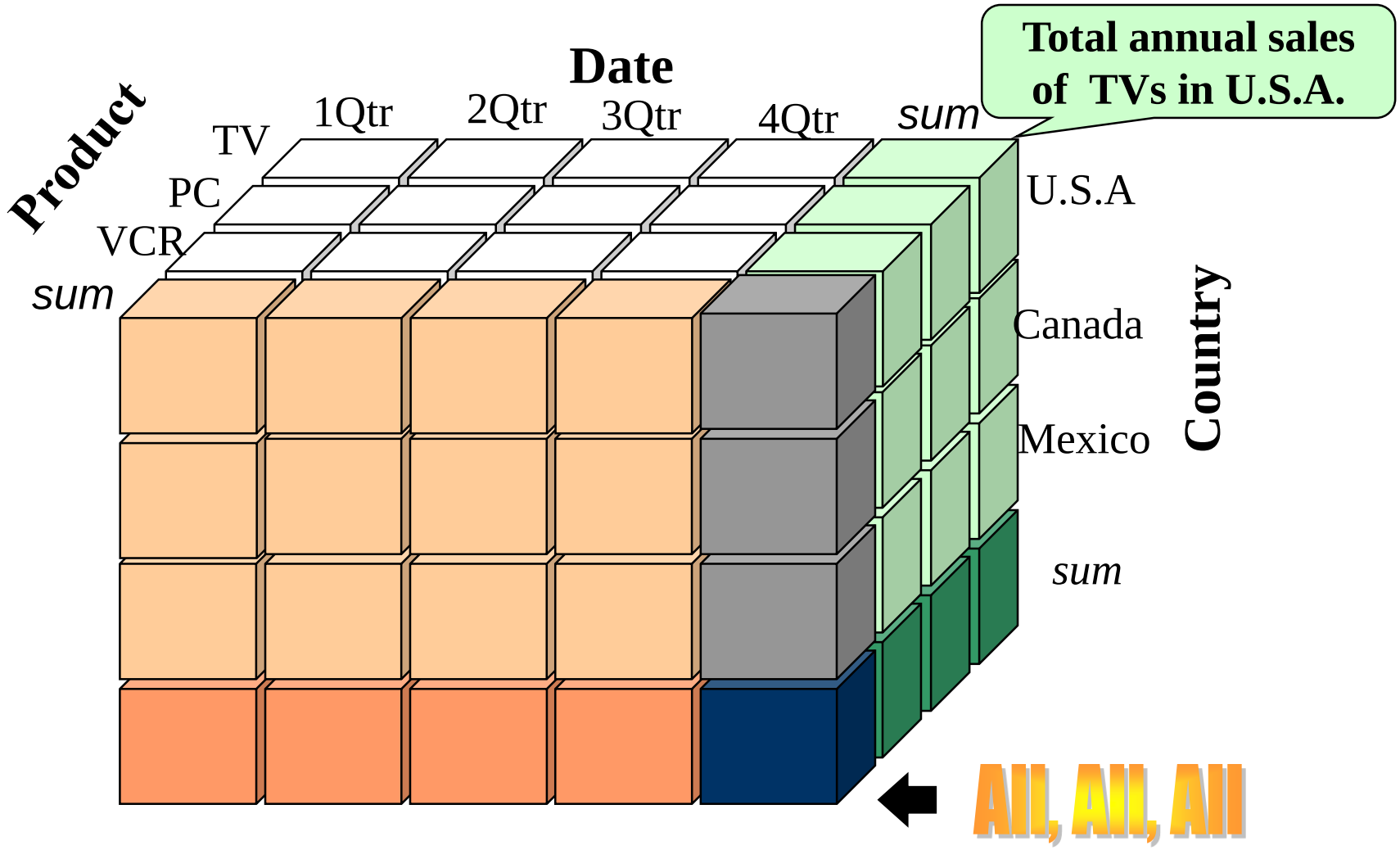
In a data warehouse, a **measure** is a property on which calculations (e.g., sum, count, average, minimum, maximum) can be made. A measure can either be categorical, algebraic or holistic.

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



Dimensions: *Product, Location, Time*
Hierarchical summarization paths




Total annual sales of TVs in U.S.A.
Date
1Qtr
2Qtr
3Qtr
4Qtr
sum
Product
TV
PC
VCR
sum
U.S.A
Canada
Mexico
sum
Country
All, All, All

UNIT 2 : Data Warehouse Architecture

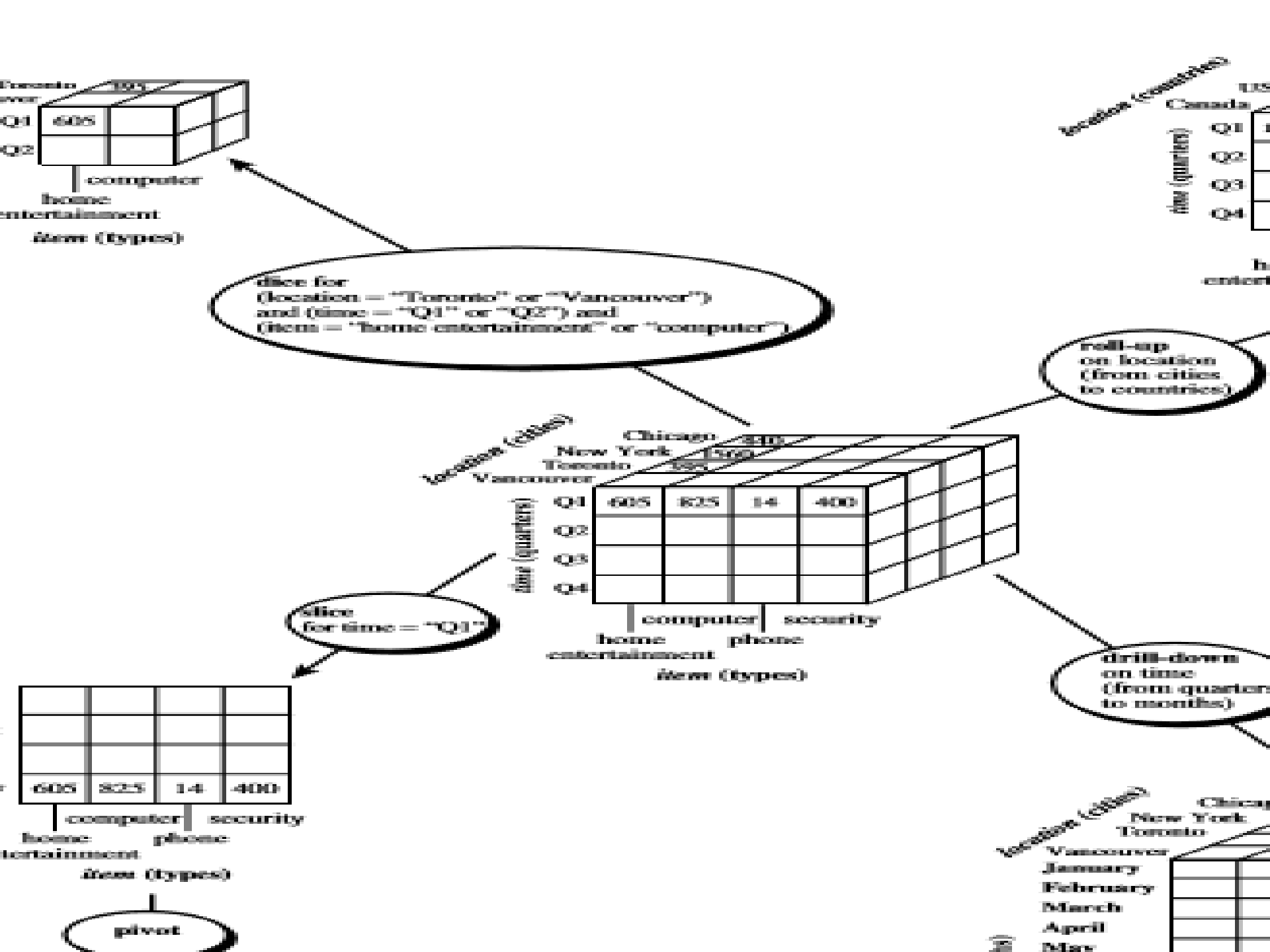
Operations in OLAP, Advantages of OLAP over OLTP, Three-Tier Data Warehouse architecture, OLAP Guidelines, Multidimensional versus Multirelational OLAP , Categories of Tools, OLAP Tools and the Internet

UNIT 2 : Data Warehouse Architecture

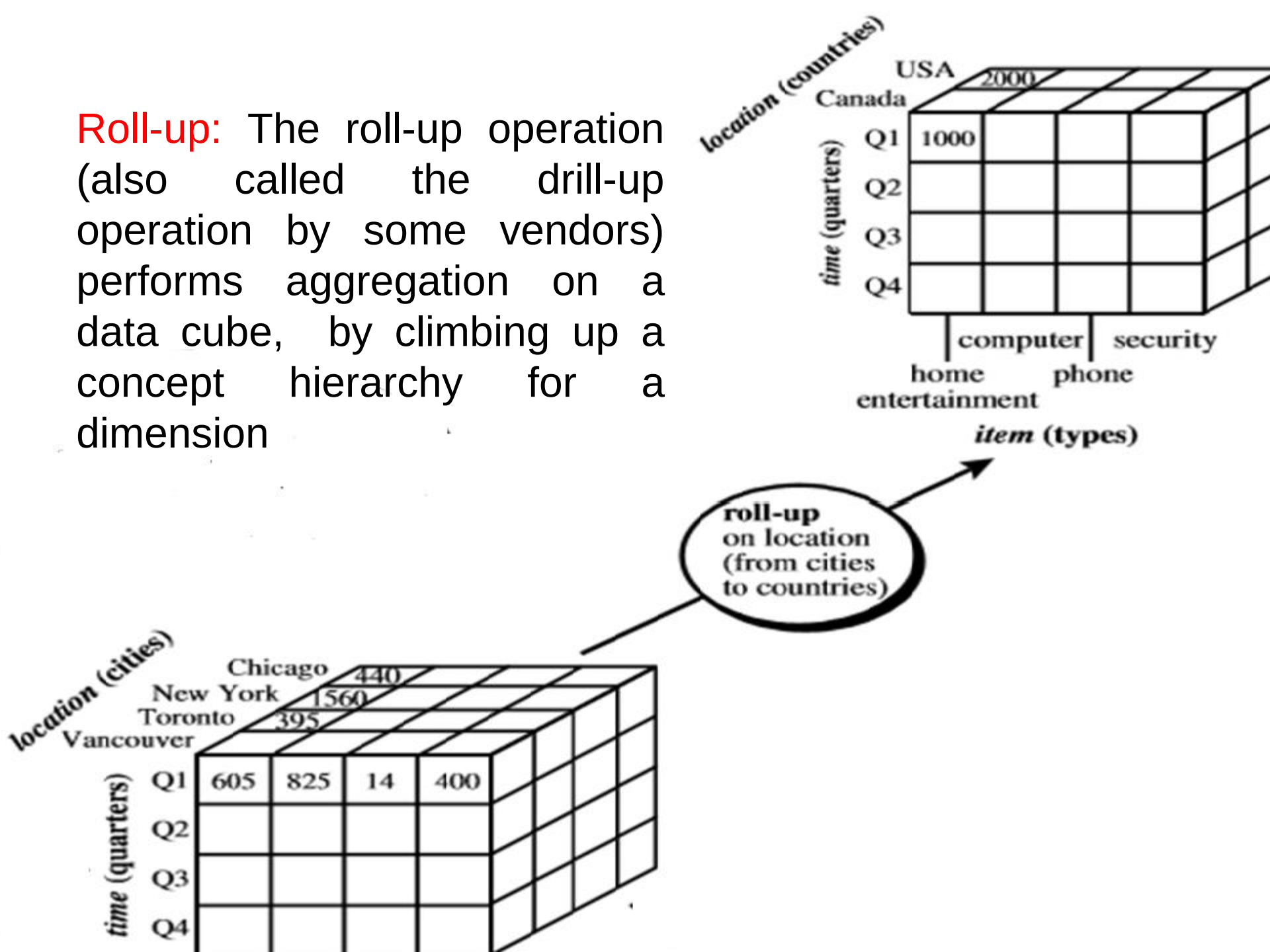
Operations in OLAP, Advantages of OLAP over OLTP, Three-Tier Data Warehouse architecture, OLAP Guidelines, Multidimensional versus Multirelational OLAP , Categories of Tools, OLAP Tools and the Internet

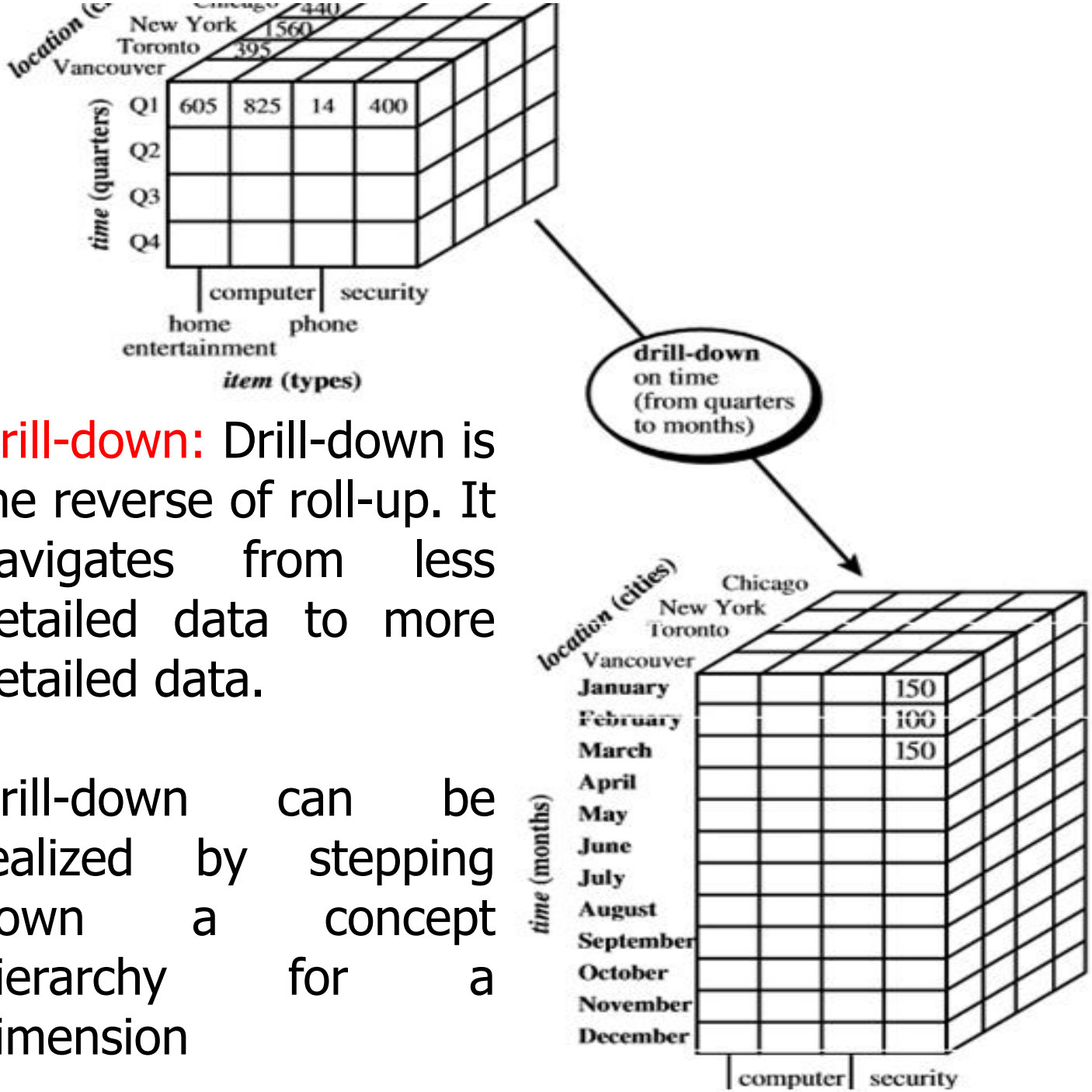
OLAP OPERATIONS

- Roll up (drill-up): summarize data
  - *by climbing up hierarchy or by dimension reduction*
- Drill down (roll down): reverse of roll-up
  - *from higher level summary to lower level summary or detailed data, or introducing new dimensions*
- Slice and dice: *project and select*
- Pivot (rotate):
  - *reorient the cube, visualization, 3D to series of 2D planes*


dice for
(location = "Toronto" or "Vancouver")
and (time = "Q1" or "Q2") and
(item = "home entertainment" or "computer")
roll-up
on location
(from cities
to countries)
slice
for time = "Q1"
drill-down
on time
(from quarters
to months)
pivot
location (cities)
Chicago
New York
Toronto
Vancouver
time (quarters)
Q1
Q2
Q3
Q4
605
825
14
400
computer
security
home
entertainment
phone
item (types)
location (countries)
USA
Canada
January
February
March
April
May

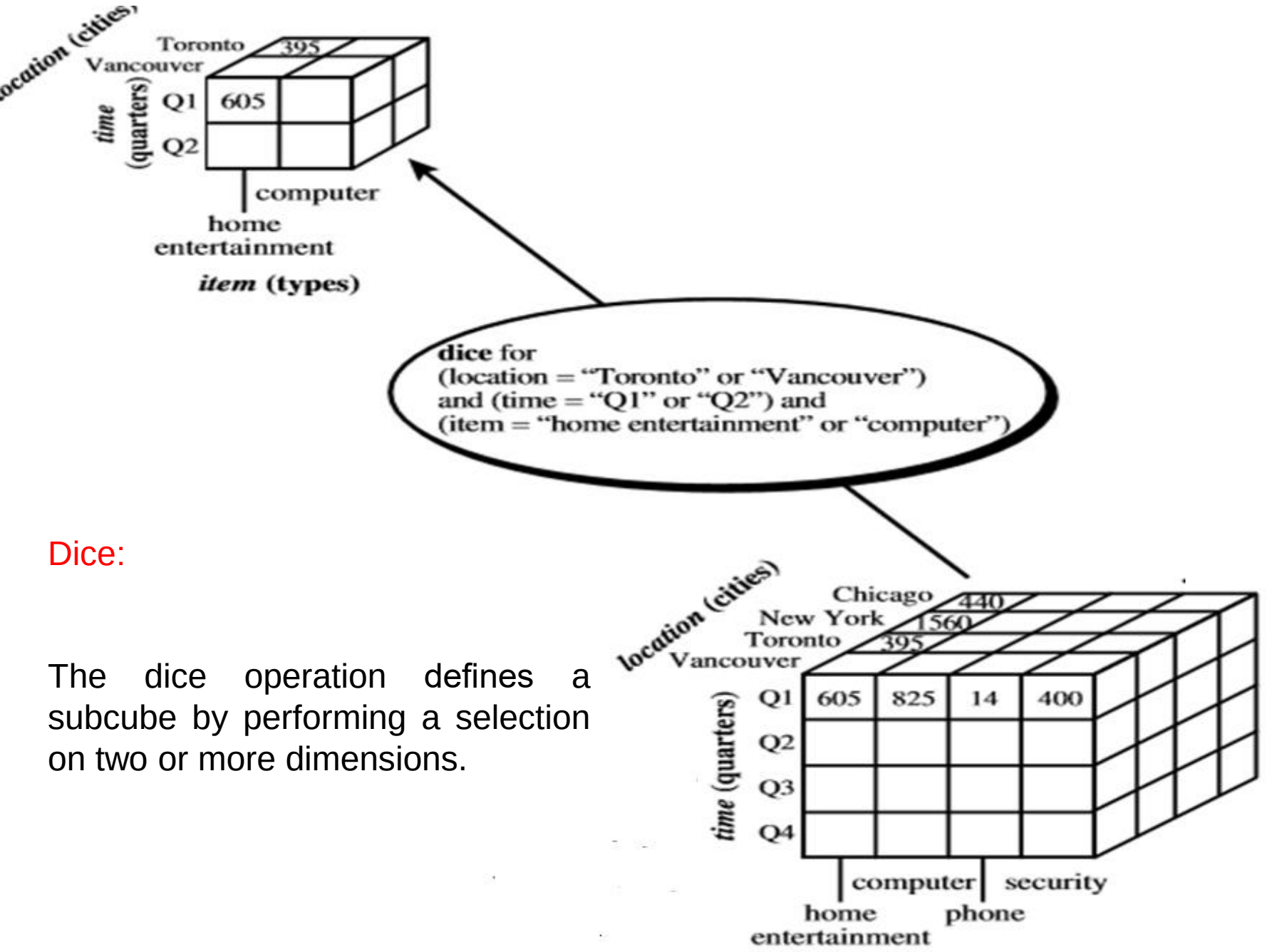
Roll-up: The roll-up operation (also called the drill-up operation by some vendors) performs aggregation on a data cube, by climbing up a concept hierarchy for a dimension





Drill-down: Drill-down is the reverse of roll-up. It navigates from less detailed data to more detailed data.

Drill-down can be realized by stepping down a concept hierarchy for a dimension

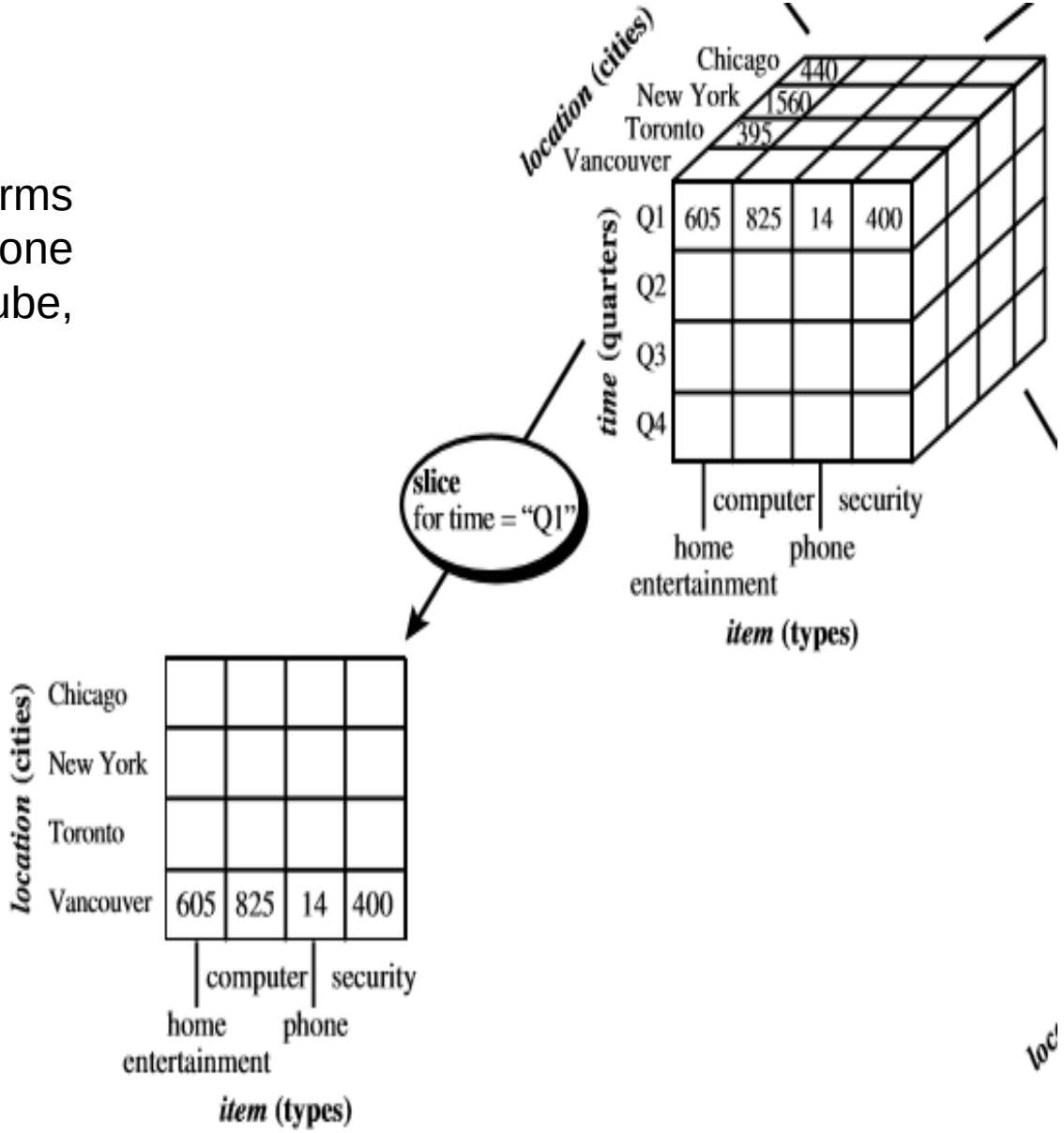


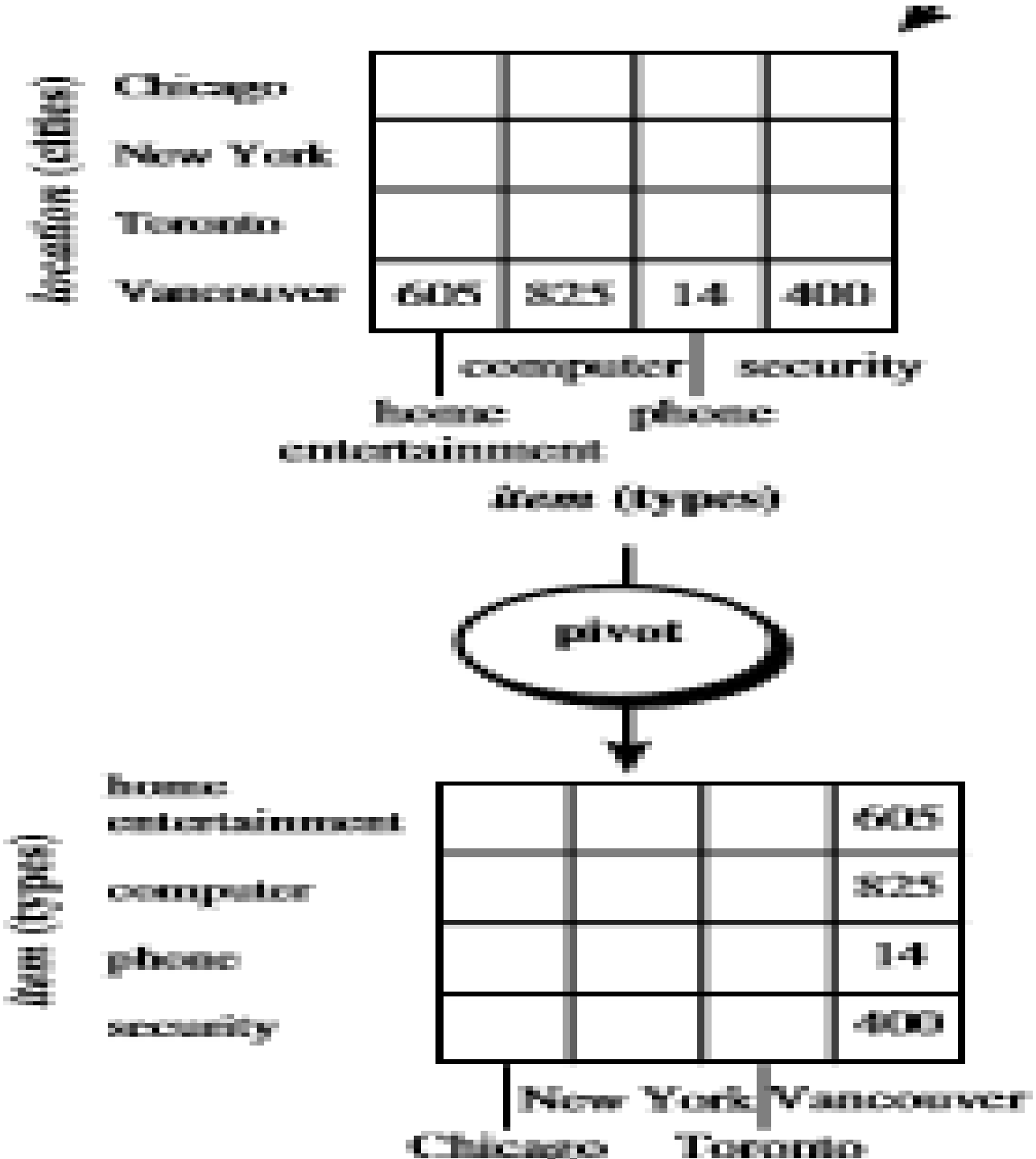
Dice:

The dice operation defines a subcube by performing a selection on two or more dimensions.

Slice :

The slice operation performs a selection on one dimension of the given cube, resulting in a subcube.




location (cities)
Chicago
New York
Toronto
Vancouver
605
825
14
400
computer
security
home entertainment
phone
item (types)
pivot
home entertainment
computer
phone
security
605
825
14
400
New York
Vancouver
Chicago
Toronto

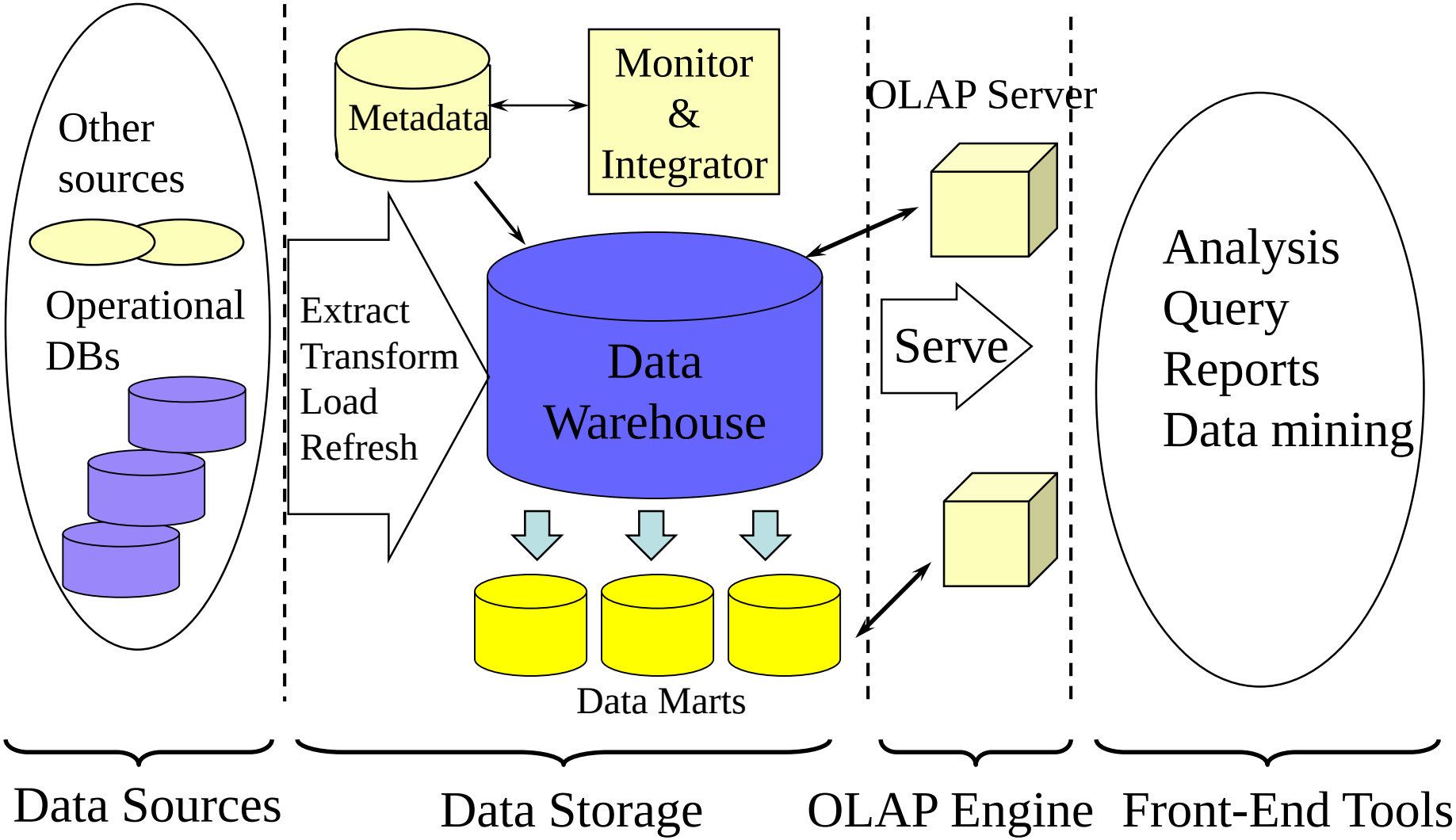
Types of OLAP Servers: ROLAP versus MOLAP versus HOLAP

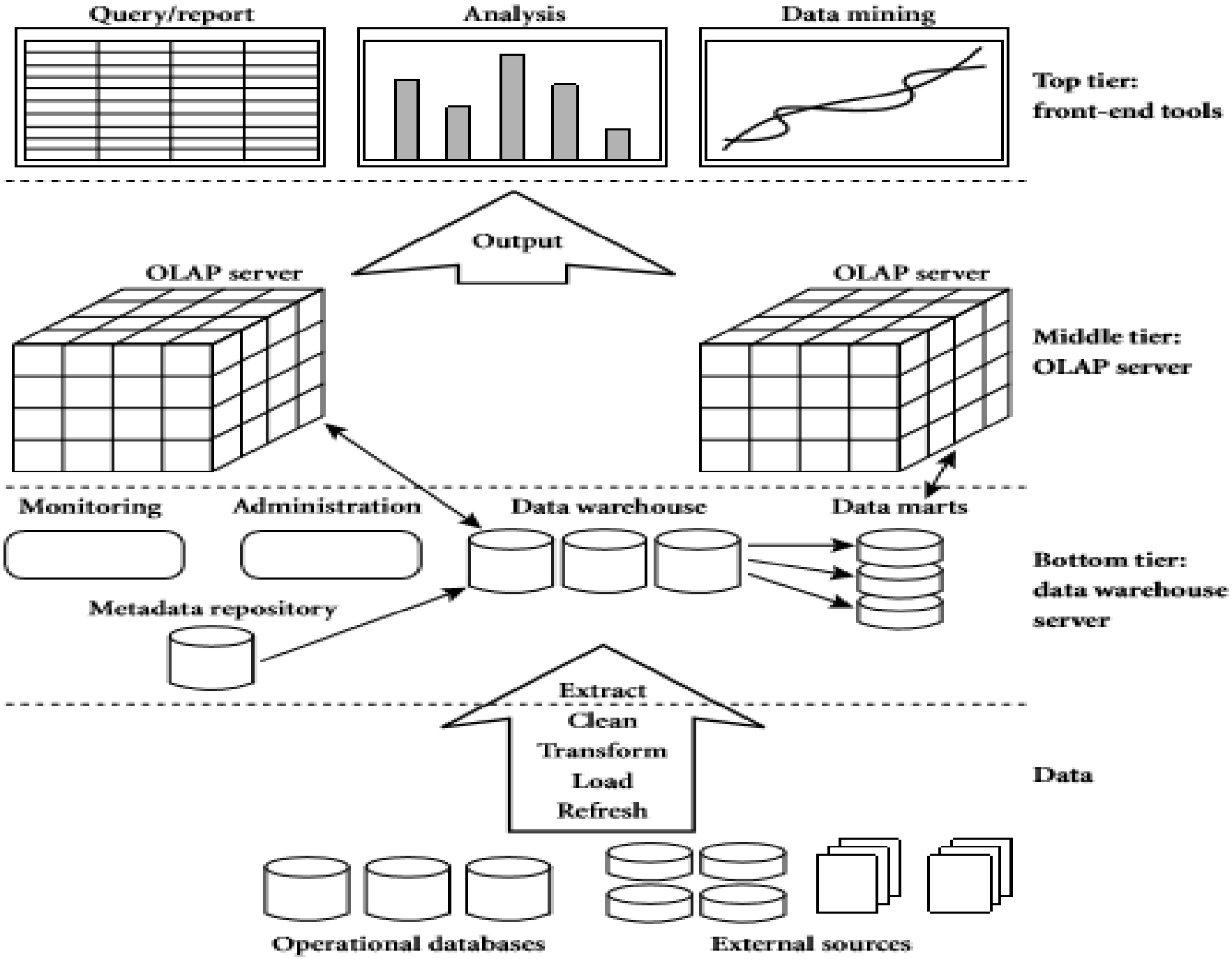
Relational OLAP (ROLAP) servers: These are the intermediate servers that stand in between a relational back-end server and client front-end tools. They use a relational or extended- relational DBMS to store and manage warehouse data, and OLAP middleware to support missing pieces.

Multidimensional OLAP(MOLAP)servers: These servers support multidimensional views of data through array-based multi dimensional storage engines. They map multi dimensional views directly to data cube array structures

Hybrid OLAP (HOLAP) servers: The hybrid OLAP approach combines ROLAP and MOLAP technology, benefiting from the greater scalability of ROLAP and the faster computation of MOLAP.

Data Warehouse: A Multi-Tiered Architecture




Query/report
Analysis
Data mining
Top tier:
front-end tools
Output
OLAP server
OLAP server
Middle tier:
OLAP server
Monitoring
Administration
Data warehouse
Data marts
Bottom tier:
data warehouse
server
Metadata repository
Extract
Clean
Transform
Load
Refresh
Data
Operational databases
External sources

A Three-Tier Data Warehouse Architecture

Data warehouses often adopt a three-tier architecture

1. **The bottom tier** is a warehouse database server that is almost always a relational database system. Back-end tools and utilities are used to feed data into the bottom tier from operational databases or other external sources.

These tools and utilities perform data extraction, cleaning, and transformation as well as load and refresh functions to update the data warehouse.

2. **The middle tier** is an OLAP server that is typically implemented using either

(1)a relational OLAP (ROLAP) model, that is, an extended relational DBMS that maps operations on multidimensional data to standard relational operations; or

(2) a multidimensional OLAP (MOLAP) model, that is, a special-purpose server that directly implements multidimensional data and operations.

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

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

the enterprise warehouse,

the data mart,

and the virtual warehouse.

- Enterprise warehouse
  - collects all of the information about subjects spanning the entire organization
- Data Mart
  - a subset of corporate-wide data that is of value to a specific groups of users. Its scope is confined to specific, selected groups, such as marketing data mart
    - Independent vs. dependent (directly from warehouse) data mart
- Virtual warehouse
  - A set of views over operational databases
  - Only some of the possible summary views may be materialized

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

OLAP

LINE	TOTAL SALES
Clothing	$12,836,450
Electronics	$16,068,300
Video	$21,262,190
Kitchen	$17,704,400
Appliances	$19,600,800
Total	$87,472,140

1 High level summary by product line

2 Drill down by year

LINE	1998	1999	2000	TOTAL
Clothing	$3,457,000	$3,590,050	$5,789,400	$12,836,450
Electronics	$5,894,800	$4,078,900	$6,094,600	$16,068,300
Video	$7,198,700	$6,057,890	$8,005,600	$21,262,190
Kitchen	$4,875,400	$5,894,500	$6,934,500	$17,704,400
Appliances	$5,947,300	$6,104,500	$7,549,000	$19,600,800
Total	$27,373,200	$25,725,840	$34,373,100	$87,472,140

3 Rotate columns to rows

YEAR	Clothing	Electronics	Video	Kitchen	Appliances	TOTAL
1998	$3,457,000	$5,894,800	$7,198,700	$4,875,400	$5,947,300	$27,373,200
1999	$3,590,050	$4,078,900	$6,057,890	$5,894,500	$6,104,500	$25,725,840
2000	$5,789,400	$6,094,600	$8,005,600	$6,934,500	$7,549,000	$34,373,100
Total	$12,836,450	$16,068,300	$21,262,190	$17,704,400	$19,600,800	$87,472,140

Figure 15-3 Simple OLAP session.

The term OLAP was introduced in a paper entitled “ providing On-Line Analytical Processing to User Analysts, “ by Dr. E.F. Codd. The paper published in 1993.

Online analytical processing is a category of software technology that enables an analysts, managers and executives to gain insight into data through fast, consistent, interactive access in a wide variety of possible views of information that has been transformed from raw data to reflect the real dimensionality of the enterprise as understood by the user

OLAP Guidelines

Multidimensional Conceptual View

OLAP system should provide multidimensional conceptual view of the data. This view should be based on the needs of the user and not on the physical data storage.

Transparency
System should be transparent to the user. But data abstraction property should be preserved.

Accessibility
System should provide easy access to the data. Users can access the data through user-friendly interface.

Consistent Reporting Performance
System should provide consistent reporting performance regardless of the complexity of the query or the amount of data being analyzed.

Consistent Reporting Performance
System should provide consistent reporting performance regardless of the complexity of the query or the amount of data being analyzed.

Client-Server Architecture
System should be based on client-server architecture. It has multiple users to access the system at the same time.

Generic Dimensionality
The system should support generic dimensionality. System can handle any number of dimensions and any type of data.

Dynamic Sparse Matrix Handling
System should be able to handle dynamic sparse matrices. System can handle data that is not regularly populated.

Multi-User Support
System should support multi-user access. Multiple users can access and analyze data at the same time.

Unrestricted Cross-Dimensional Operations
System should allow unrestricted cross-dimensional operations. System should allow users to analyze data from different dimensions without restrictions.

Intuitive Data Manipulation
System should provide intuitive data manipulation tools. Users can manipulate and analyze data in a user-friendly manner.

Flexible Reporting
System should provide flexible reporting capabilities. Users can generate reports in various formats and with different levels of detail.

Unlimited Dimensions and Aggregation Levels
System should support unlimited dimensions and aggregation levels. System should can handle any number of dimensions and any level of aggregation

Multidimensional versus Multi relational OLAP

In the MOLAP model, online analytical processing is best implemented by storing the data multidimensional that is easily viewed in a multidimensional way. Here the data structure is fixed so that the logic to process multidimensional analysis ca be based on well defined methods of establishing data storage coordinates. Usually, multidimensional databases (MDDBs) are vendors proprietary systems.

On the other hand , the ROLAP model relies on the existing relational DBMS of data warehouse. OLAP Features are provided against the relational database.

Multidimensional vs. Multirelational OLAP:

MOLAP (Multidimensional OLAP): Stores data in pre-aggregated "cubes" optimized for fast query response, ideal for complex analysis with pre-defined dimensions and hierarchies.

ROLAP (Relational OLAP): Utilizes standard relational databases and SQL queries, offering greater flexibility for ad-hoc analysis but potentially slower performance for complex queries.

As mentioned earlier, multidimensional database management systems are proprietary software systems. These systems provide the capability to consolidate and fabricate summarized cubes during the process that loads data into the MDDBs from the main data warehouse. The users who need summarized data enjoy fast response times from the preconsolidated data.

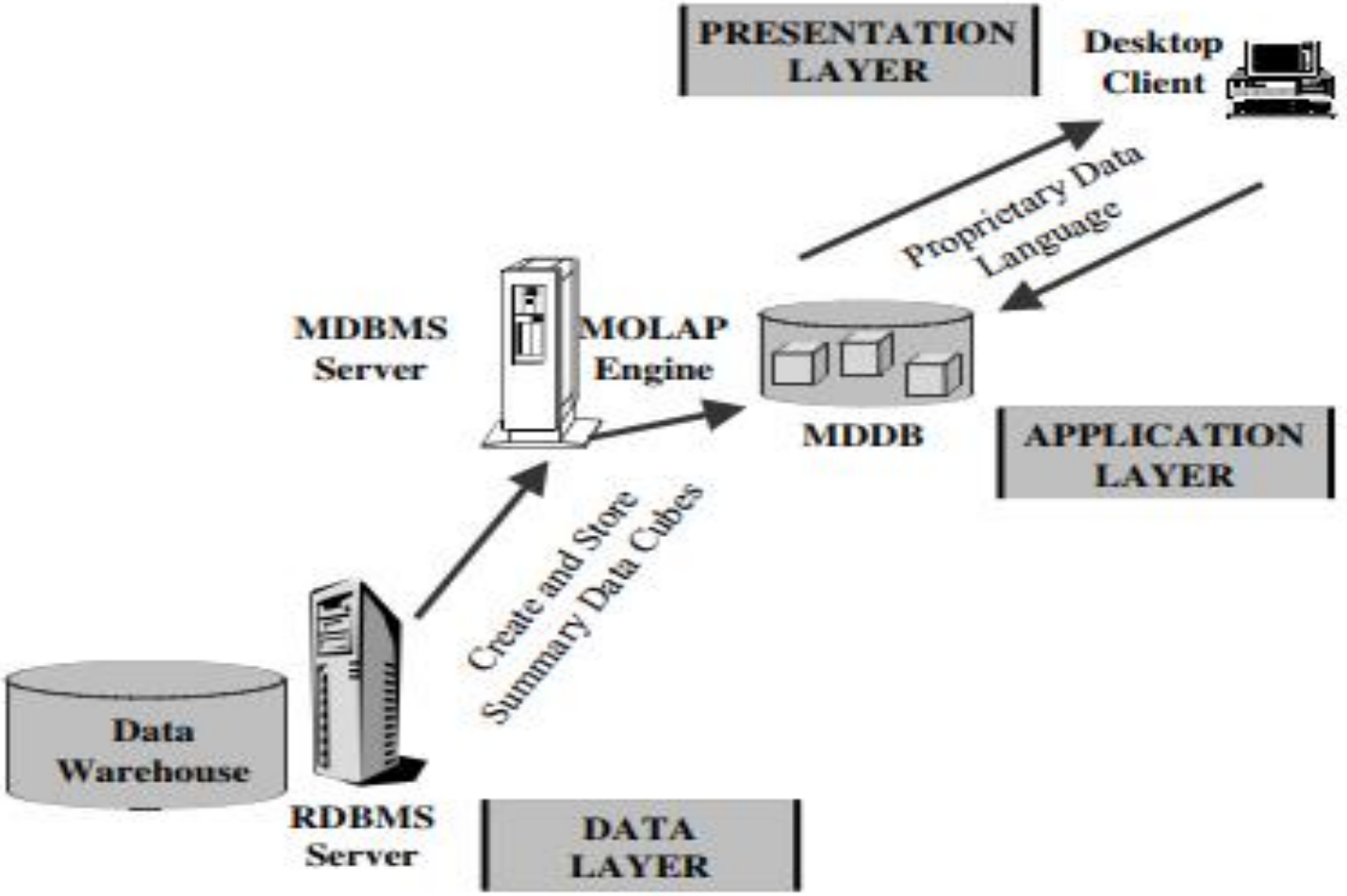


Figure 15-16 The MOLAP model.

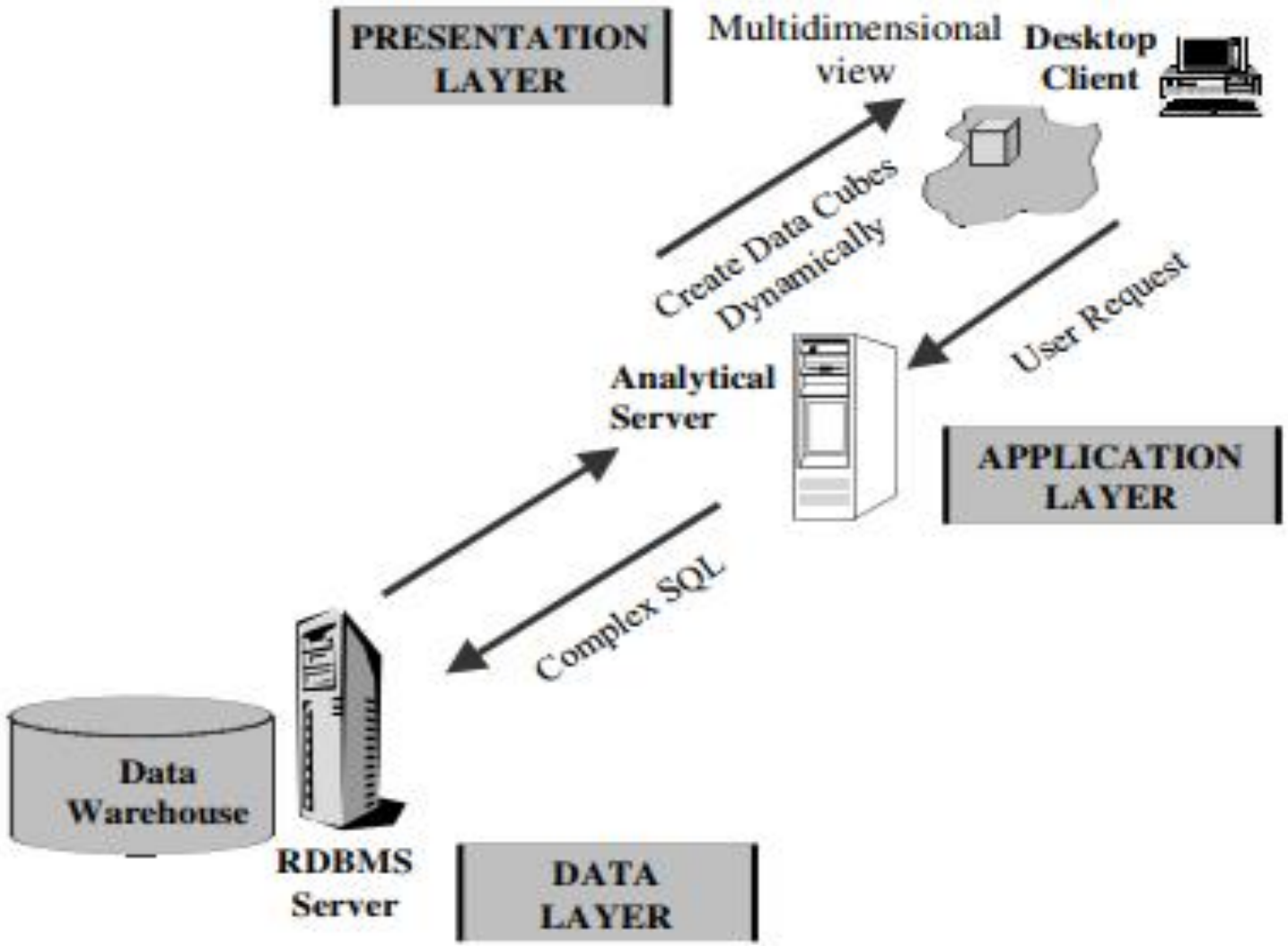


Figure 15-17 The ROLAP model.

ROLAP	MOLAP
ROLAP stands for **Relational Online Analytical Processing.**	While MOLAP stands for **Multidimensional Online Analytical Processing.**
ROLAP is used for large data volumes.	While it is used for limited data volumes.
The access of ROLAP is slow.	While the access of MOLAP is fast.
In ROLAP, Data is stored in relation tables.	While in MOLAP, Data is stored in multidimensional array.
In ROLAP, Data is fetched from data-warehouse.	While in MOLAP, Data is fetched from MDDBs database.
In ROLAP, Complicated sql queries are used.	While in MOLAP, Sparse matrix is used.
In ROLAP, Static multidimensional view of data is created.	While in MOLAP, Dynamic multidimensional view of data is created.

	Data Storage	Underlying Technologies	Functions and Features
ROLAP	Data stored as relational tables in the warehouse. Detailed and light summary data available. Very large data volumes. All data access from the warehouse storage.	Use of complex SQL to fetch data from warehouse. ROLAP engine in analytical server creates data cubes on the fly. Multidimensional views by presentation layer.	Known environment and availability of many tools. Limitations on complex analysis functions. Drill-through to lowest level easier. Drill-across not always easy.
MOLAP	Data stored as relational tables in the warehouse. Various summary data kept in proprietary databases (MDDBs) Moderate data volumes. Summary data access from MDDB, detailed data access from warehouse.	Creation of pre-fabricated data cubes by MOLAP engine. Proprietary technology to store multidimensional views in arrays, not tables. High speed matrix data retrieval. Sparse matrix technology to manage data sparsity in summaries.	Faster access. Large library of functions for complex calculations. Easy analysis irrespective of the number of dimensions. Extensive drill-down and slice-and-dice capabilities.

Figure 15-19 ROLAP versus MOLAP.

- **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

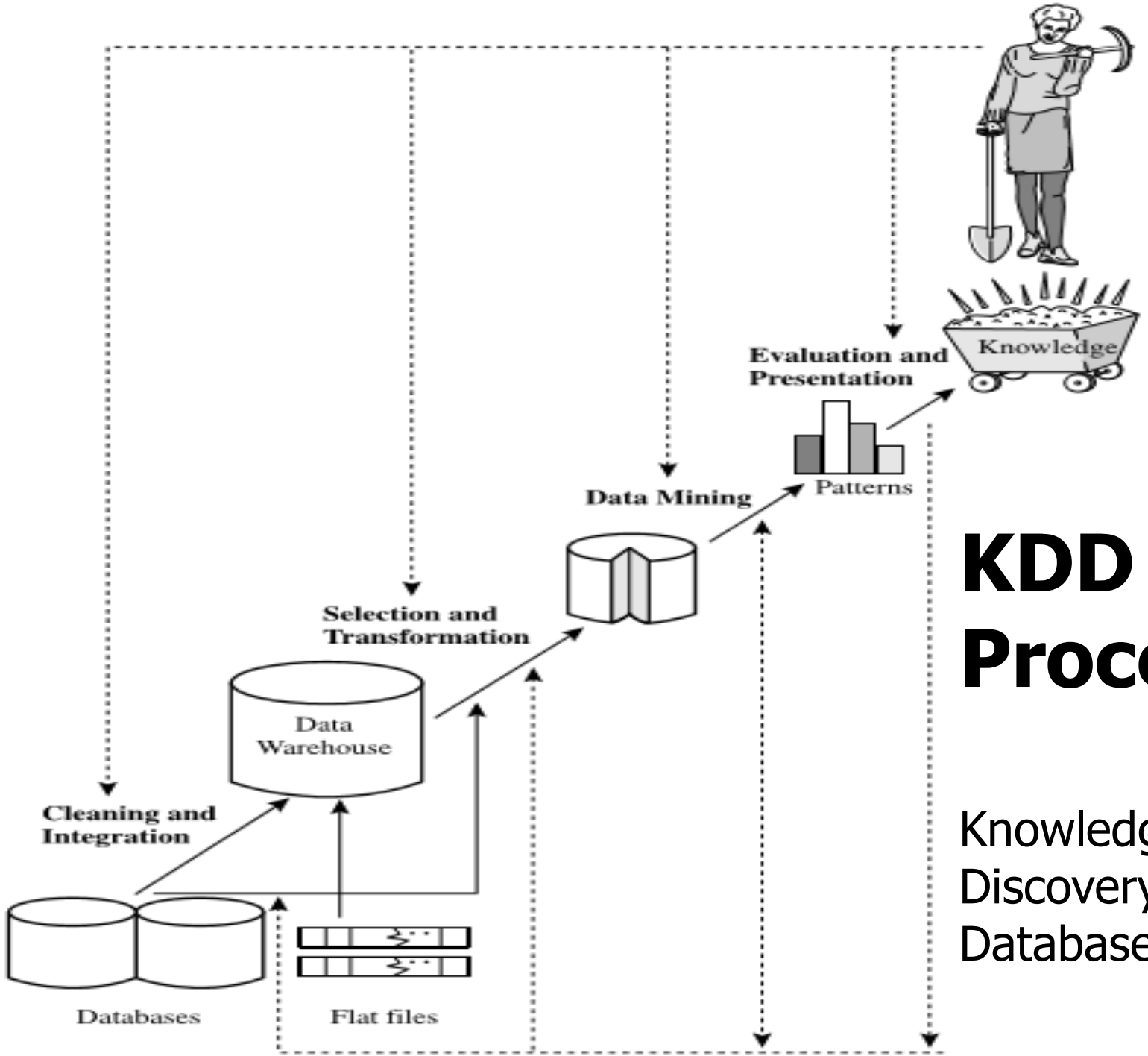
- **Alternative names**

  Knowledge discovery (mining) in databases (KDD), knowledge extraction, data/pattern analysis, data archeology, data dredging, information harvesting, business intelligence, etc.

Unit 3: Introduction to Data Mining

Basic Concepts of Data Mining; Data Mining primitives: Task-relevant data, mining objective, measures and identification of patterns, KDD versus data mining, data mining tools and applications.

Data Mining Query Languages: Data specification, specifying kind of knowledge, hierarchy specification, pattern presentation & visualization specification, data mining languages and standardization of data mining, Architectures of Data Mining Systems.



KDD Process

Knowledge Discovery in Databases

Many people treat data mining as a synonym for another popularly used term, Knowledge Discovery from Data.

Steps in KDD

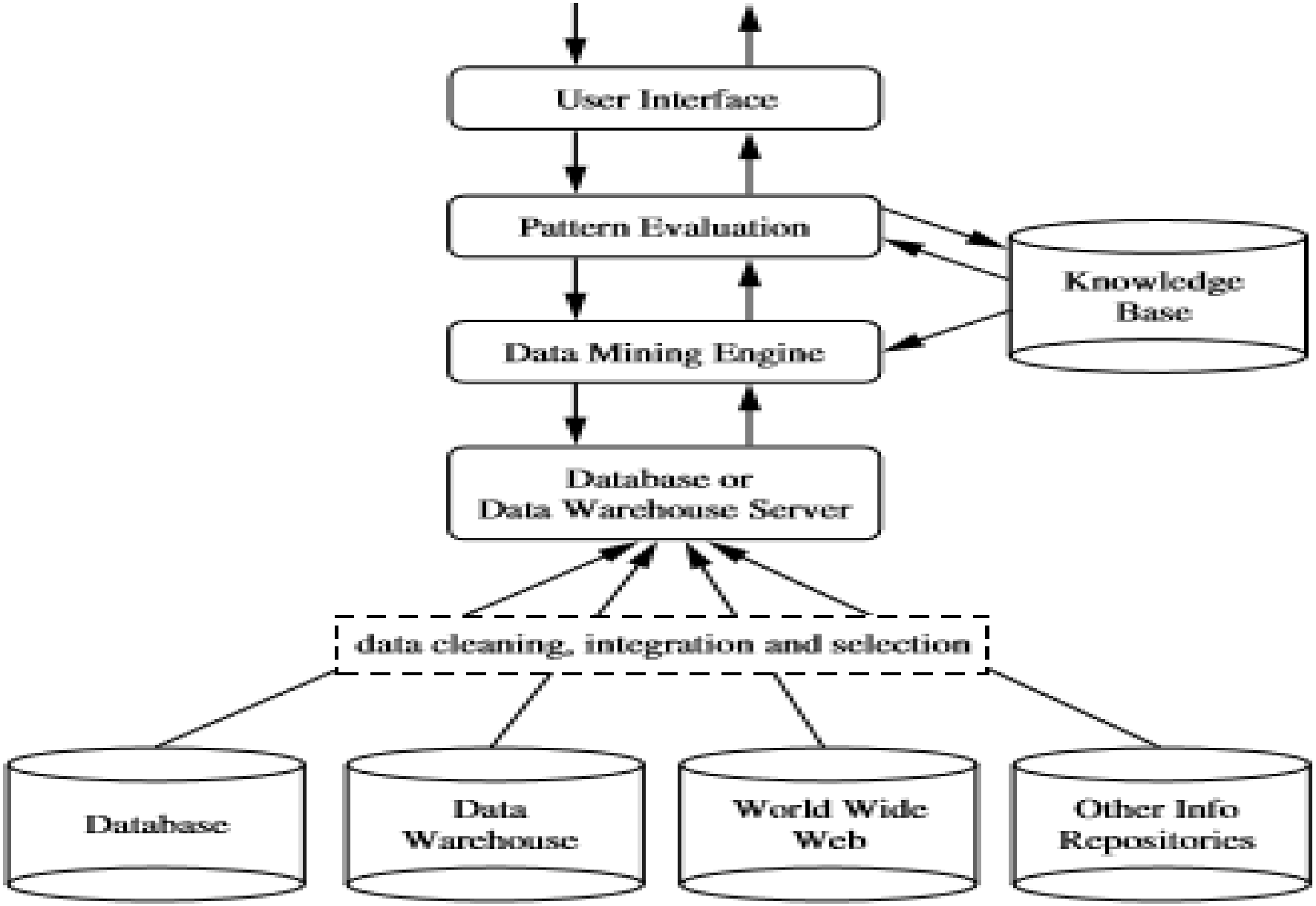
- **Data cleaning** (to remove noise and inconsistent data).
- **Data integration** (where multiple data sources may be combined)
- **Data selection** (where data relevant to the analysis task are retrieved from the database)
- **Data transformation** (where data are transformed or consolidated into forms appropriate for mining by performing summary or aggregation operations, for instance).

- **Data mining** (an essential process where intelligent methods are applied in order to extract data patterns).

- **Pattern evaluation** (to identify the truly interesting patterns representing knowledge based on some interestingness measures.

- **Knowledge presentation** (where visualization and knowledge representation techniques are used to present the mined knowledge to the user).

Architecture of a Data Mining System



The architecture of a typical data mining system may have the following major components

Database, data warehouse, World Wide Web, or other information repository: This is one or a set of databases, data warehouses, spread sheets, or other kinds of information repositories. Data cleaning and data integration techniques may be performed on the data.

Database or data warehouse server: The database or data warehouse server is responsible for fetching the relevant data, based on the user's data mining request.

Knowledge base: This is the domain knowledge that is used to guide the search or evaluate the interestingness of resulting patterns.

Such knowledge can include concept hierarchies, used to organize attributes or attribute values into different levels of abstraction.

Knowledge such as user beliefs, which can be used to assess a pattern's interestingness based on its unexpectedness, may also be included.

Data mining engine: This is essential to the data mining system and ideally consists of a set of functional modules for tasks such as characterization, association and correlation analysis, classification, prediction, cluster analysis, outlier analysis.

Pattern evaluation module: This component typically employs interestingness measures and interacts with the data mining modules so as to focus the search toward interesting patterns. It may use interestingness thresholds to filter out discovered patterns.

User interface: This module communicates between users and the datamining system, allowing the user to interact with the system by specifying a data mining query or task

Data Mining—On What Kind of Data?

In principle, data mining is not specific to one type of media or data. Data mining should be applicable to any kind of information repository.

Flat files: Flat files are actually the most common data source for data mining algorithms, especially at the research level.

Flat files are simple data files in text or binary format with a structure known by the data mining algorithm to be applied.

The data in these files can be transactions, time-series data, scientific measurements, etc.

Relational Databases: Briefly, a relational database consists of a set of tables containing either values of entity attributes, or values of attributes from entity relationships.

Tables have columns and rows, where columns represent attributes and rows represent tuples.

A tuple in a relational table corresponds to either an object or a relationship between objects and is identified by a set of attribute values representing a unique key.

- **Data Warehouses:** A data warehouse as a storehouse is a repository of data collected from multiple data sources (often heterogeneous) and is intended to be used as a whole under the same unified schema.

- A data warehouse gives the option to analyze data from different sources under the same roof.

- **Transaction Databases**: A transaction database is a set of records representing transactions, each with a time stamp, an identifier and a set of items.
- Associated with the transaction files could also be descriptive data for the items.

- **Multimedia Databases**: Multimedia databases include video, images, audio and text media. Multimedia is characterized by its high dimensionality, which makes data mining even more challenging.

Spatial Databases: Spatial databases are databases that, in addition to usual data, store geographical information like maps, and global or regional positioning.

Data mining functionalities, and the kinds of patterns they can discover

Characterization and Discrimination

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

The data corresponding to the user-specified class are typically collected by a database query.

For example, to study the characteristics of software products whose sales increased by 10% in the last year

Data discrimination: Comparison of the target class with one or a set of comparative classes.

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

The target and contrasting classes can be specified by the user, and the corresponding data objects retrieved through database queries.

For example, the user may like to compare the general features of software products whose sales increased by 10% in the last year with those whose sales decreased by at-least 30% during the same period.

Mining Frequent Patterns, Associations, and Correlations

Frequent patterns, as the name suggests, are patterns that occur frequently in data. There are many kinds of frequent patterns, including itemsets, subsequences, and substructures.

Mining frequent patterns leads to the discovery of interesting associations and correlations within data.

Market Basket Analysis

Frequent itemset mining leads to the discovery of associations and correlations among items in large transactional or relational data sets.

With massive amounts of data continuously being collected and stored, many industries are becoming interested in mining such patterns from their databases.

The discovery of interesting correlation relationships among huge amounts of business transaction records can help in many business decision-making processes.

A typical example of frequent itemset mining is market basket analysis.

This process analyzes customer buying habits by finding associations between the different items that customers place in their “shopping baskets”.

The discovery of such associations can help retailers develop marketing strategies by gaining insight into which items are frequently purchased together by customers.

Data Mining Query Language:

The Data Mining Query Language (DMQL) was proposed by Han, Fu, Wang, et al. for the DBMiner data mining system. The Data Mining Query Language is actually based on the Structured Query Language (SQL). Data Mining Query Languages can be designed to support ad hoc and interactive data mining. This DMQL provides commands for specifying primitives. The DMQL can work with databases and data warehouses as well. DMQL can be used to define data mining tasks

A desired feature of data mining systems is the ability to support ad hoc and interactive data mining in order to facilitate the flexible and effective knowledge discovery. Data mining query languages can be designed to support such a feature.

Key features of DMQL

DMQL supports the specification of:

- **Data to be mined**

- **Type of knowledge to be mined** (e.g., association rules, classification rules, clustering)
- **Background knowledge** (e.g., concept hierarchies)

- **Interestingness constraints** (e.g., support/confidence thresholds)

Advantages of DMQL

Declarative: Focus on *what* to mine, not *how*.

User-friendly: Simple syntax for complex tasks.

Flexible: Supports different kinds of knowledge.

Integrates well with databases: Easy to apply on existing data.

Syntax for DMQL

- Syntax for specification of
  - task-relevant data
  - the kind of knowledge to be mined
  - concept hierarchy specification
  - interestingness measure
  - pattern presentation and visualization
- Putting it all together — a DMQL query

Syntax for task-relevant data specification

- *use database* database_name, or *use data warehouse* data_warehouse_name
  - directs the data mining task to the database or data warehouse specified
- *from relation*(s)/cube(s) [*where* condition]
  - specify the database tables or data cubes involved and the conditions defining the data to be retrieved
- *in relevance to* att_or_dim_list
  - Lists attributes or dimensions for exploration

Syntax for task-relevant data specification

- *order by* order_list
  - Specifies the sorting order of the task relevant data
- *group by* grouping_list
  - Specifies criteria for grouping the data
- *having* condition
  - Specifies the condition by which groups of data are considered relevant

Specification of task-relevant data

Example 4.11 This example shows how to use DMQL to specify the task-relevant data described in Example 4.1 for the mining of associations between items frequently purchased at *AllElectronics* by Canadian customers, with respect to customer *income* and *age*. In addition, the user specifies that she would like the data to be grouped by date. The data are retrieved from a relational database.

```
use database AllElectronics_db
in relevance to I.name, I.price, C.income, C.age
from customer C, item I, purchases P, items_sold S
where I.item_ID = S.item_ID and S.trans_ID = P.trans_ID and P.cust_ID = C.cust_ID
        and C.address = "Canada"
group by P.date
```

□

Key Components of DMQL

1. Data Specification : Defines the **target dataset** for mining.
Chooses database, table, and attributes.
Uses filtering conditions (WHERE clause).
Example:

```
use database bank_db;

select * from customer_data

where age > 30;
```

2. Kind of Knowledge to Be Mined

Specifies the **type of pattern or model** to discover.
Types include:
Association Rules
Classification
Clustering
Sequential Patterns
Outlier Detection
Prediction

```
mine association_rules
from transactions
with min_support = 0.2
and min_confidence = 0.6;
```

3. Hierarchy Specification

Allows the use of **concept hierarchies** for generalization (e.g., City < State < Country).

- Supports **multilevel mining**
- Enables mining at different abstraction levels

```
define hierarchy for location as
{
  city < state < country;
};
```

4. Pattern Presentation & Visualization

Specifies how the **output** should be formatted and displayed.
Formats:
Rule form (If-Then)
Decision trees
Tables
Charts and graphs

display as decision_tree;
sort by support desc;

5. Visualization Specification

Purpose: Enhances the **interpretability** of results using graphical formats.

Often integrated with pattern presentation for:

- Charts (bar, pie, line)
- Tree structures
- Interactive visuals

display chart as bar_graph;

Component	Function
Data Specification	Selects data and conditions for mining
Kind of Knowledge	Specifies the mining goal (rules, clusters, etc.)
Hierarchy Specification	Provides background knowledge for generalization
Pattern Presentation	Defines how results are shown (tree, table, rules)
Visualization Specification	Adds graphical/visual interpretation of patterns

Datawarehouse indexing

Bitmap indexing is a space- and time-efficient indexing technique used in data warehouses and OLAP systems, especially when dealing with **low-cardinality** columns (columns with few distinct values).

A bitmap index uses bitmaps (bit vectors) to represent the presence or absence of attribute values in a table. Each distinct value of a column gets a bitmap where:
1 indicates that the row has the value
0 indicates that it does not.

Row	Gender
1	Male
2	Female
3	Male
4	Female
5	Male

Distinct Values: Male, Female

Bitmap Indexes:

Row	Male	Female
1	1	0
2	0	1
3	1	0
4	0	1
5	1	0

Advantages

- **Fast for querying:** Bitwise operations are very efficient.
- **Space-efficient:** Bitmaps are compact, especially for low-cardinality columns.
- **Supports complex conditions:** Great for combining multiple conditions quickly.

Disadvantages

- **Not suitabl**e for high-cardinality columns: Bitmaps become large and sparse.
- **Slow updates**: Every update may affect multiple bitmaps, making them inefficient for frequently updated tables.

Feature	Joint Indexing	Bitmap Indexing
Definition	An indexing technique that creates a **composite index** over multiple columns.	An indexing method that uses **bitmaps** to represent the presence or absence of a value in a column.
Structure	Stores index entries combining values from **two or more columns**.	Uses **bit vectors** (arrays of 0s and 1s) for **each distinct value** of an attribute.
Best For	Queries that involve **combinations** of column values (e.g., WHERE col1 AND col2).	Columns with **low cardinality** (few distinct values, e.g., gender, status).
Efficiency	Faster for **multi-column** filters but requires more space and maintenance.	Extremely efficient in **space** and **bitwise operations**, especially for read-heavy workloads.
Storage Space	May consume more space if columns have many combinations.	Very compact for low-cardinality attributes.
Update Performance	Can be costly to update; index must reflect changes in multiple columns.	Updates are **expensive** for high-cardinality columns.
Example Use Case	(gender, department) used together in search filters.	gender = 'F' or married = 'Yes'.

Association Rules

The problem

- When we go grocery shopping, we often have a standard list of things to buy.
- Each shopper has a distinctive list, depending on one's needs and preferences.
- A housewife might buy healthy ingredients for a family dinner, while a bachelor might buy cold drinks and chips.
- Understanding these buying patterns can help to increase sales in several ways. If there is a pair of items, X and Y, that are frequently bought together:

- Both X and Y can be placed on the same shelf, so that buyers of one item would be prompted to buy the other.
- Promotional discounts could be applied to just one out of the two items.
- Advertisements on X could be targeted at buyers who purchase Y.

- While we may know that certain items are frequently bought together, the question is, how do we uncover these associations?

Definition

- Association rules analysis is a technique to uncover how items are associated to each other. There are tw0 common ways to measure association.

- **Measure 1: Support**. This says how popular an itemset is, as measured by the proportion of transactions in which an itemset appears.

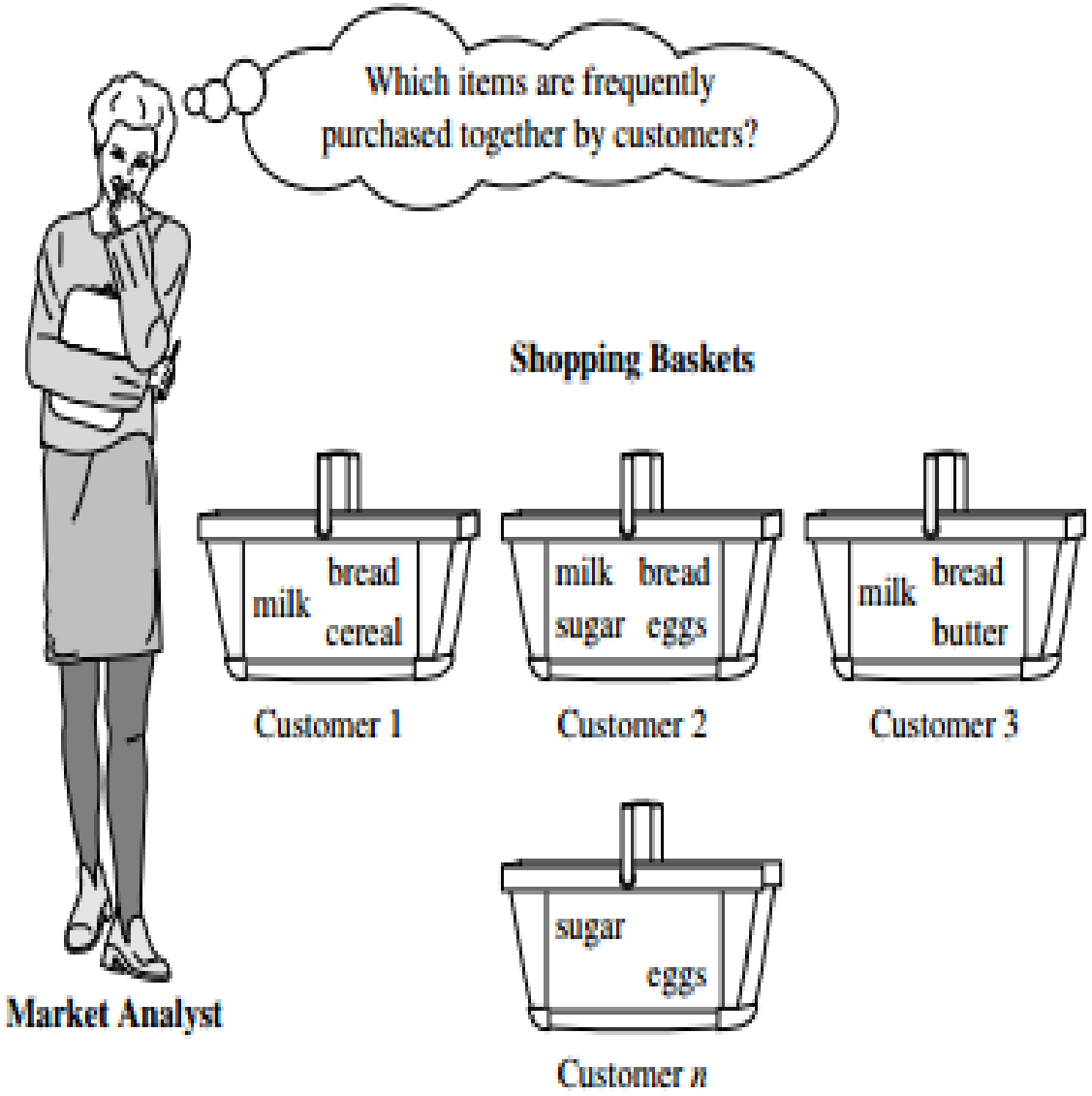
Frequent Itemset Mining

With the quick growth in e-commerce applications, there is an accumulation vast quantity of data in months not in years.

Data Mining, also known as Knowledge Discovery in Databases(KDD), to find anomalies, correlations, patterns, and trends to predict outcomes.

Apriori algorithm is a classical algorithm in data mining. It is used for mining frequent itemsets and relevant association rules.

Frequent itemset mining leads to the discovery of associations and correlations among items in large transactional or relational data sets.


Which items are frequently purchased together by customers?
Shopping Baskets
milk
bread
cereal
Customer 1
milk
bread
sugar
eggs
Customer 2
milk
bread
butter
Customer 3
sugar
eggs
Customer n
Market Analyst

The patterns can be represented in the form of association rules. For example, the information that customers who purchase computers also tend to buy antivirus software at the same time is represented in the following association rule: $computer \Rightarrow antivirus\_software\ [support = 2\%, confidence = 60\%].$

A support of 2% for means that 2% of all the transactions under analysis show that computer and antivirus software are purchased together.

A confidence of 60% means that 60% of the customers who purchased a computer also bought the software. It signifies the likelihood of item Y being purchased when item X is purchased.

Support

$$supp(X) = \frac{\text{Number of transaction in which } X \text{ appears}}{\text{Total number of transactions}}$$

$$supp(Onion) = \frac{4}{6} = 0.66667$$

Confidence

$$conf(X \longrightarrow Y) = \frac{supp(X \cup Y)}{supp(X)}$$

Apriori Algorithm: Finding Frequent Itemsets by Confined Candidate Generation

The name of the algorithm is based on the fact that the algorithm uses prior knowledge of frequent itemset properties.

Apriori employs an iterative approach known as a level-wise search, where k-itemsets are used to explore (k + 1) itemsets.

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

Mining Frequent Itemsets

- Find the *frequent itemsets*: the sets of items that have the minimum support
  - A subset of a frequent itemset must also be a frequent itemset
    - i.e., if {*AB*} is a frequent itemset, both {*A*} and {*B*} should be a frequent itemset
  - Iteratively find frequent itemsets with cardinality from 1 to *k* (*k*-itemset)
- Use the frequent itemsets to generate association rules.

The Apriori Algorithm

- Join Step: C_k is generated by joining $L_{k-1}$with itself
- Prune Step: Any (k-1)-itemset that is not frequent cannot be a subset of a frequent k-itemset
- Pseudo-code:

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

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

Apriori is an algorithm proposed by R. Agrawal and R. Srikant in 1994 for mining frequent itemsets for Boolean association rules.

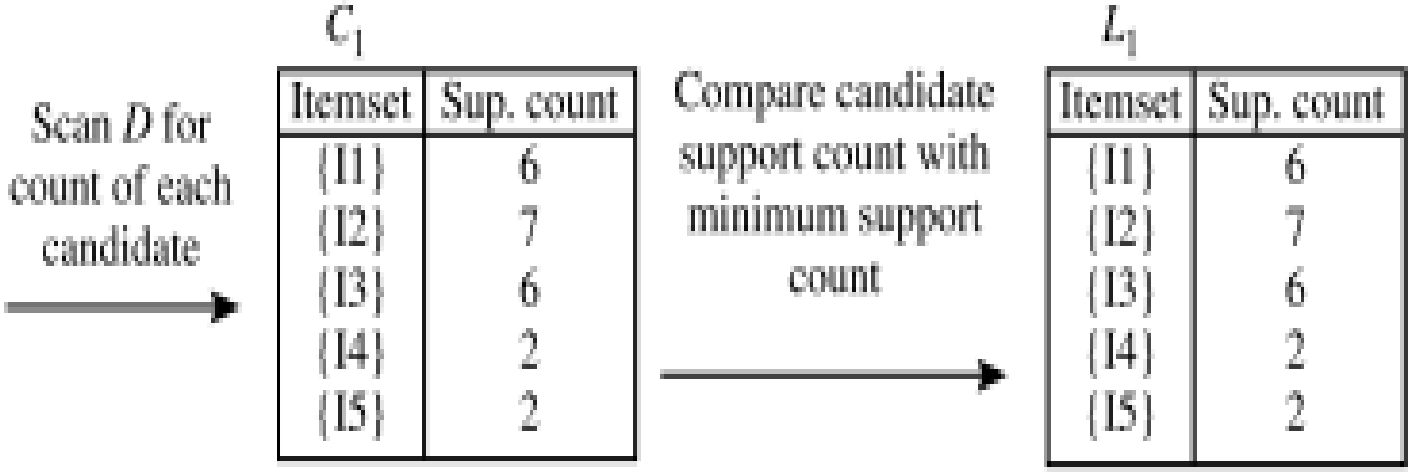
Apriori property: *All nonempty subsets of a frequent itemset must also be frequent.*

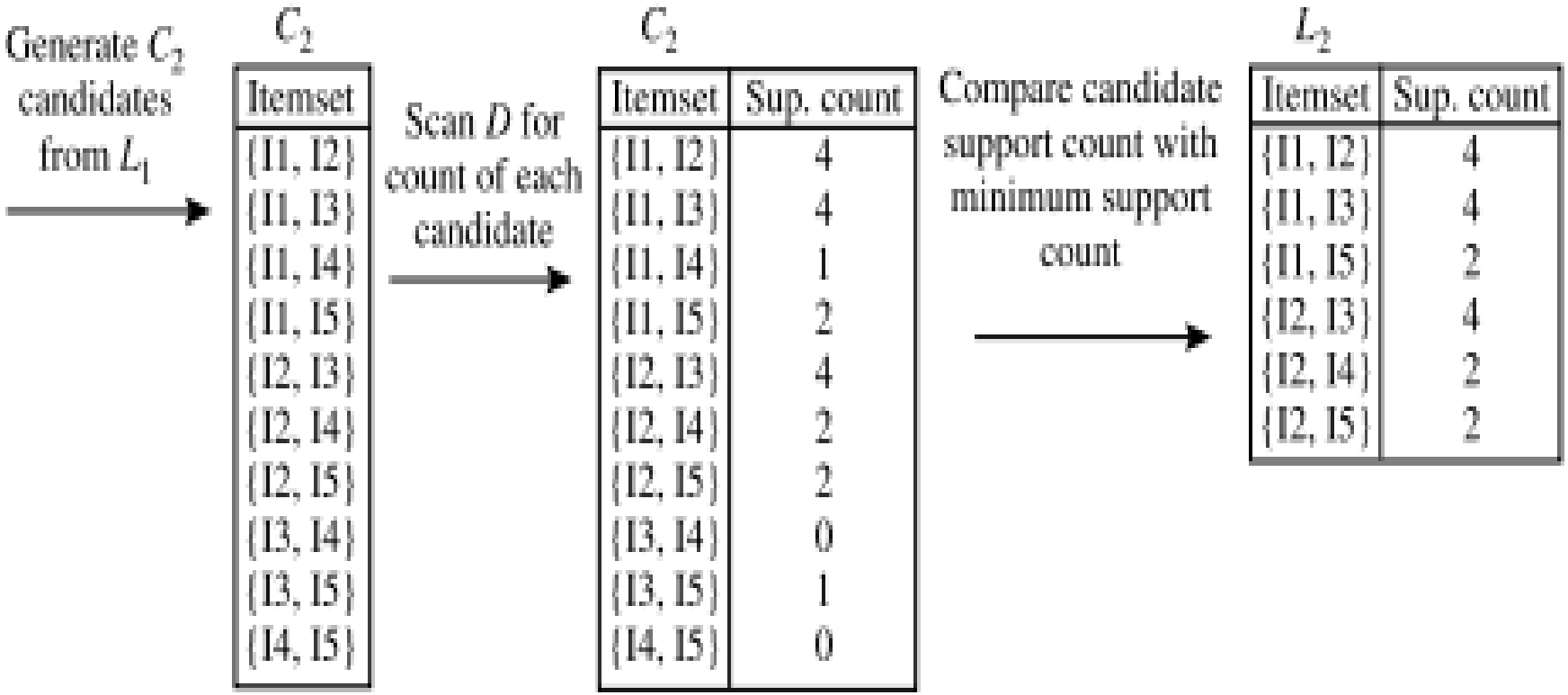
Transactional Data

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

Suppose that the minimum support count required is 2, that is, min sup is 2

The Apriori Algorithm: Finding Frequent Itemsets Using Candidate Generation





Similarly generate candidate set with possible combination of 3 itemset

Example 5.4 Generating association rules. Let's try an example based on the transactional data for *AllElectronics* shown in Table 5.1. Suppose the data contain the frequent itemset $l = \{I1, I2, I5\}$. What are the association rules that can be generated from l? The nonempty subsets of l are $\{I1, I2\}$, $\{I1, I5\}$, $\{I2, I5\}$, $\{I1\}$, $\{I2\}$, and $\{I5\}$. The resulting association rules are as shown below, each listed with its confidence:

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

If the minimum confidence threshold is, say, 70%, then only the second, third, and last rules above are output, because these are the only ones generated that are strong. Note that, unlike conventional classification rules, association rules can contain more than one conjunct in the right-hand side of the rule. ■

Find the frequent item sets using Apriori Algorithm. Also generate the association rules from the frequent item sets.

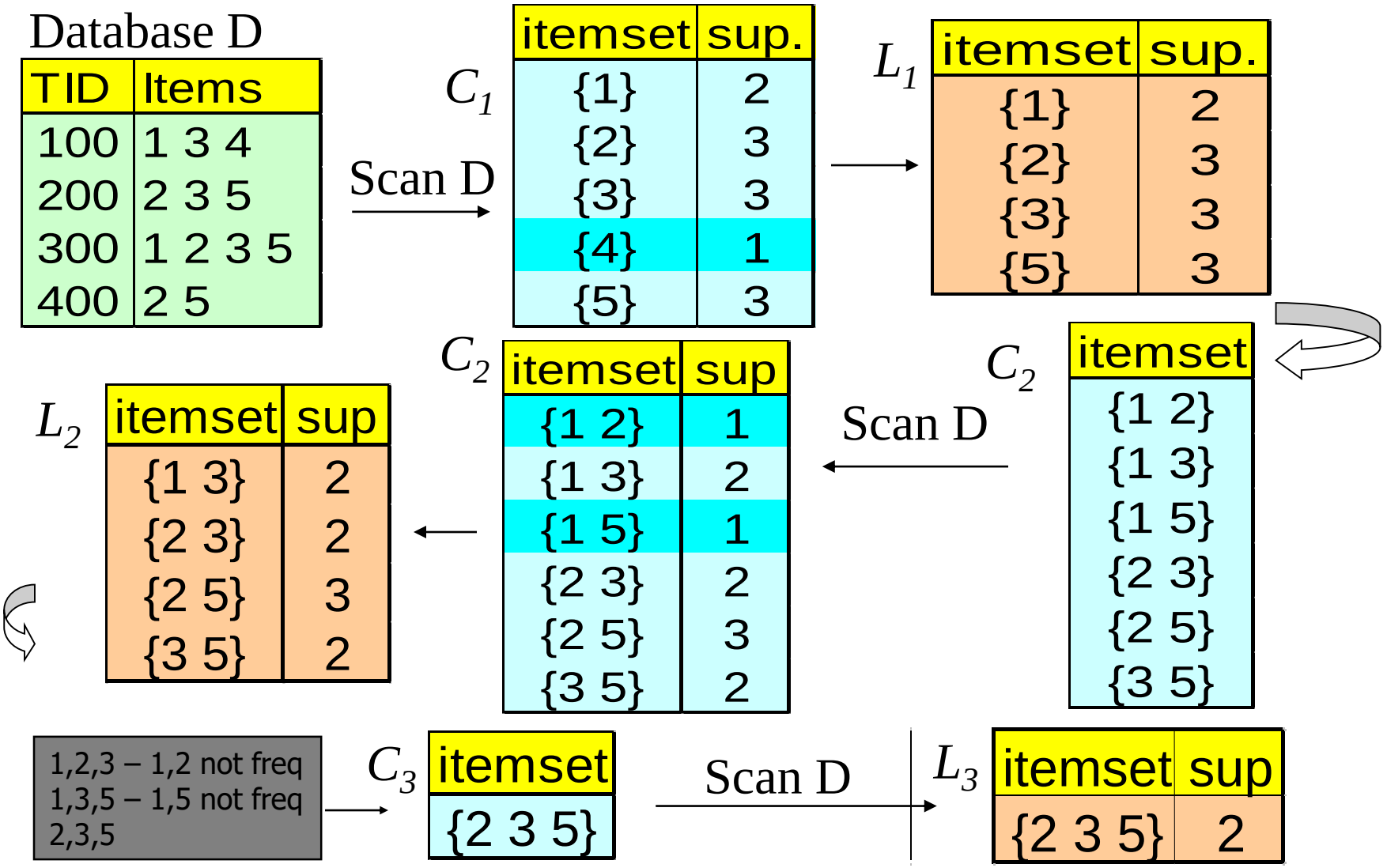
TID	Items
100	1 3 4
200	2 3 5
300	1 2 3 5
400	2 5

Minimum support count = 2

minimum confidence threshold = 80%

The Apriori Algorithm — Example

Minimum support count = 2



How to Generate Association Rules from Frequent Itemsets?

- Done using the following equation for confidence
  - Where the conditional probability is expressed in terms of itemset support counts
- Confidence (A=>B) = P(B/A) = support_count(A $\cup$B) /support_count(A)
  - support_count(A $\cup$ B) = # transactions containing itemsets A $\cup$ B
  - support_count(A) = # transactions containing the itemset A
- Generate as follows :
  - For each frequent itemset l, generate all nonempty subsets of l
  - For every nonempty subset of I, output the rule “s=>(l-s)” if support_count(l)/support_count(s) >= min_conf (minimum confidence threshold)

Consider a transactional database where 1, 2, 3, 4, 5, 6, 7 are items.

Suppose the minimum support is 60%. Find all frequent itemsets. Indicate each candidate set Ck, k = 1, 2, …, the candidates that are pruned by each pruning step, and the resulting frequent itemsets Lk.

ID	Items
t_1	1, 2, 3, 5
t_2	1, 2, 3, 4, 5
t_3	1, 2, 3, 7
t_4	1, 3, 6
t_5	1, 2, 4, 5, 6

Minimum support = 5 × 60% = 3. Minimum confidence be 75%

C_1

Itemset	Support
{1}	5
{2}	4
{3}	4
{4}	2
{5}	3
{6}	2
{7}	1

→

L_1

Itemset	Support
{1}	5
{2}	4
{3}	4
{5}	3

C_2

Itemset	Support
{1, 2}	4
{1, 3}	4
{1, 5}	3
{2, 3}	3
{2, 5}	3
{3, 5}	2

→

L_2

Itemset	Support
{1, 2}	4
{1, 3}	4
{1, 5}	3
{2, 3}	3
{2, 5}	3

C_3

Itemset	Support
{1, 2, 3}	3
{1, 2, 5}	3
{1, 3, 5}	
{2, 3, 5}	

→

L_3

Itemset	Support
{1, 2, 3}	3
{1, 2, 5}	3

FP Growth Algorithm

An efficient and scalable method to find frequent patterns. It allows **frequent itemset** discovery without candidate itemset generation.

Following are the steps for FP Growth Algorithm

- Scan DB once, find frequent 1-itemset (single item pattern)
- Sort frequent items in frequency descending order, f-list
- Scan DB again, construct FP-tree
- Construct the conditional FP tree in the sequence of reverse order of F - List - generate frequent item set

Mining Frequent Itemsets without Candidate Generation

An interesting method in this attempt is called frequent-pattern growth, or simply FP-growth, which adopts a divide-and-conquer strategy.

First, it compresses the dataset representing frequent items into a frequent-pattern tree, or FP-tree, which retains the item set association information.

It then divides the compressed dataset into a set of conditional dataset, each associated with one frequent item or "pattern fragment," and mines each such dataset separately.

FP-tree Example: step 1

Step 1: Scan DB for the first time to ge

TID	*Items bought*
100	{f, a, c, d, g, i, m, p}
200	{a, b, c, f, l, m, o}
300	{b, f, h, j, o}
400	{b, c, k, s, p}
500	{a, f, c, e, l, p, m, n}

Min Support count is 3

FP-tree Example: step 1

Step 1: Scan DB for the first time to generate L

TID	*Items bought*
100	{*f, a, c, d, g, i, m, p*}
200	{*a, b, c, f, l, m, o*}
300	{*b, f, h, j, o*}
400	{*b, c, k, s, p*}
500	{*a, f, c, e, l, p, m, n*}

L

Item	*frequency*
f	*4*
c	*4*
a	*3*
b	*3*
m	*3*
p	*3*

By-Product of First Scan of Database

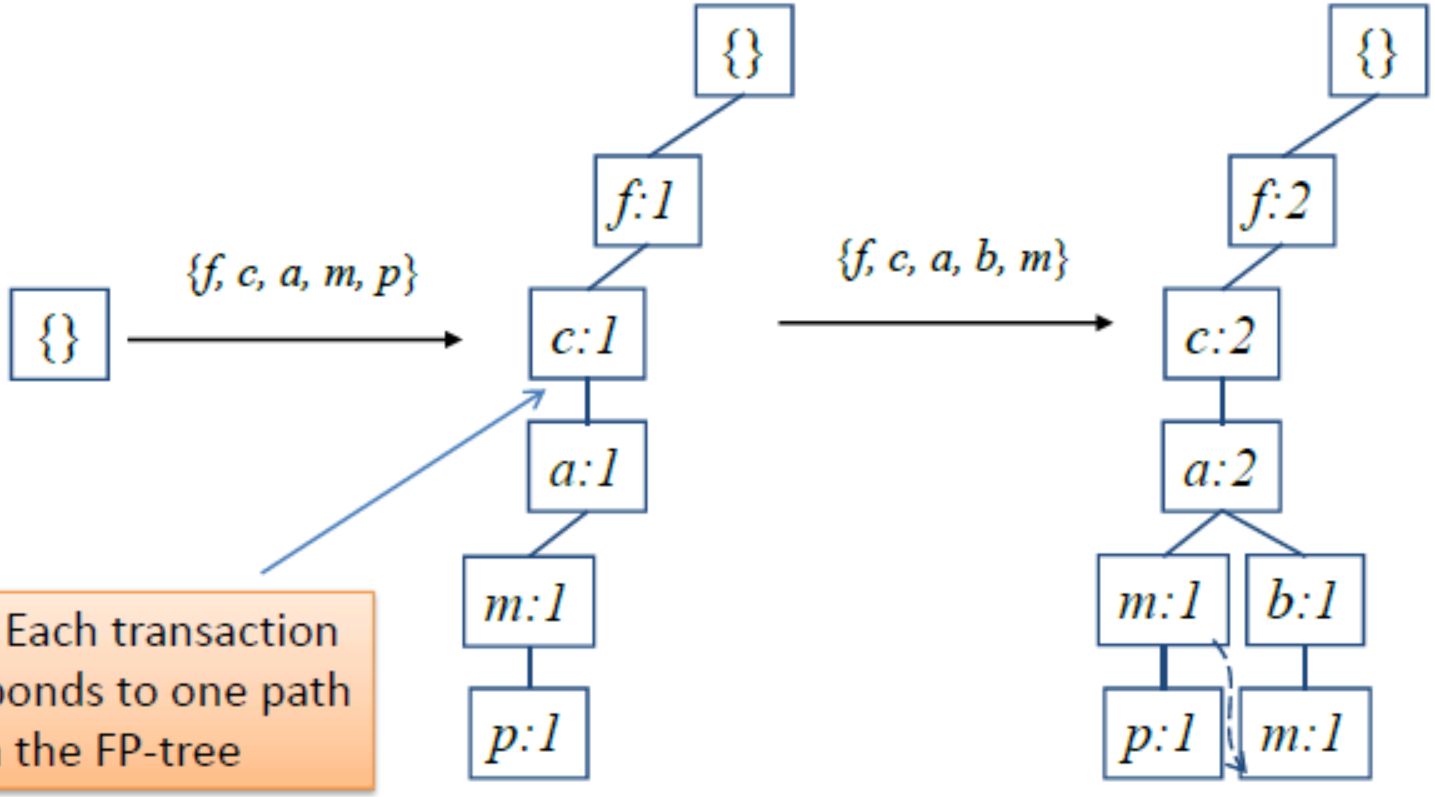
FP-tree Example: step 2

Step 2: scan the DB for the second time, order frequent items in each transaction

TID	*Items bought*	*(ordered) frequent items*
100	{*f, a, c, d, g, i, m, p*}	{*f, c, a, m, p*}
200	{*a, b, c, f, l, m, o*}	{*f, c, a, b, m*}
300	{*b, f, h, j, o*}	{*f, b*}
400	{*b, c, k, s, p*}	{*c, b, p*}
500	{*a, f, c, e, l, p, m, n*}	{*f, c, a, m, p*}

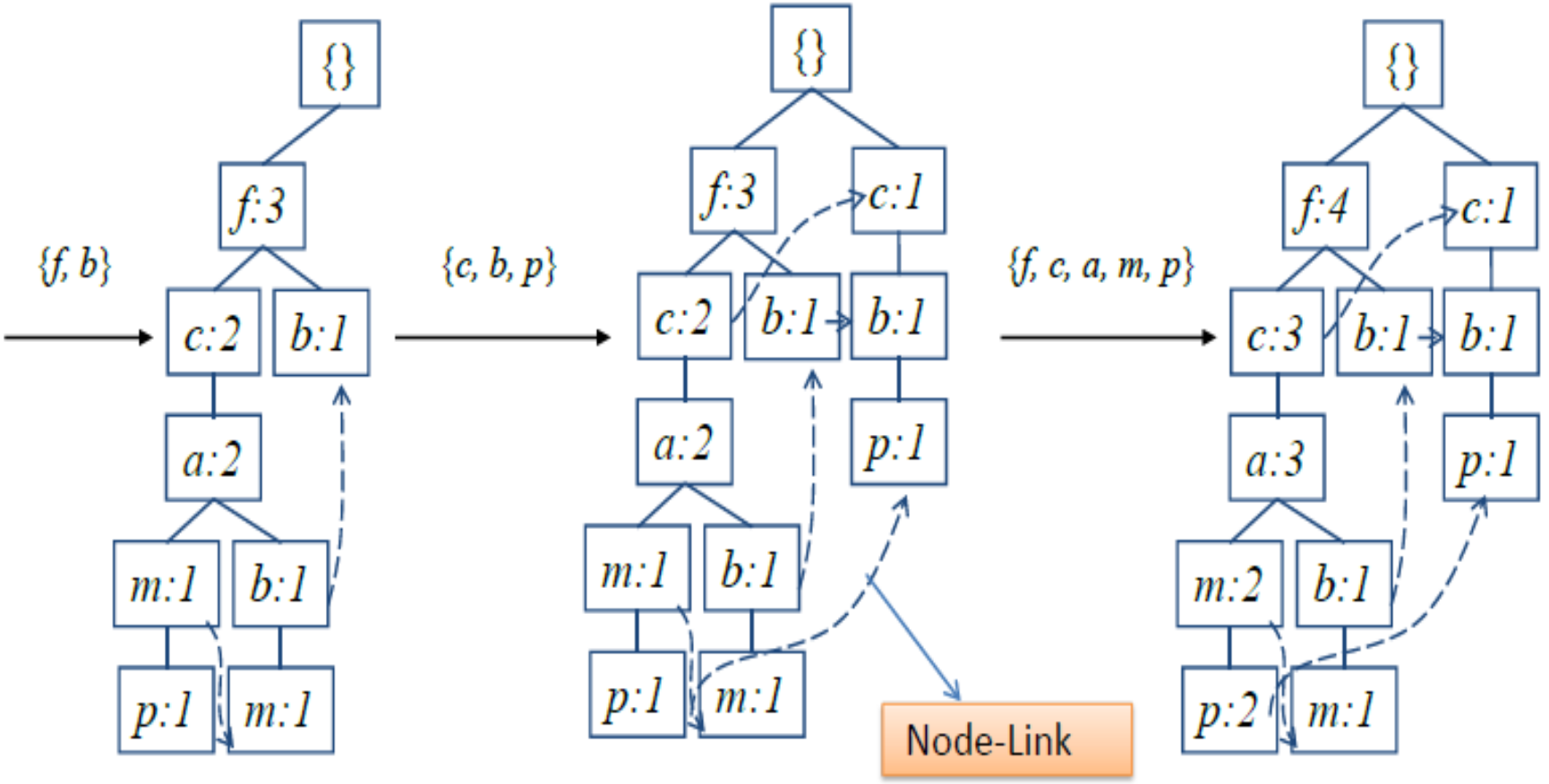
FP-tree Example: step 2

Step 2: construct FP-tree



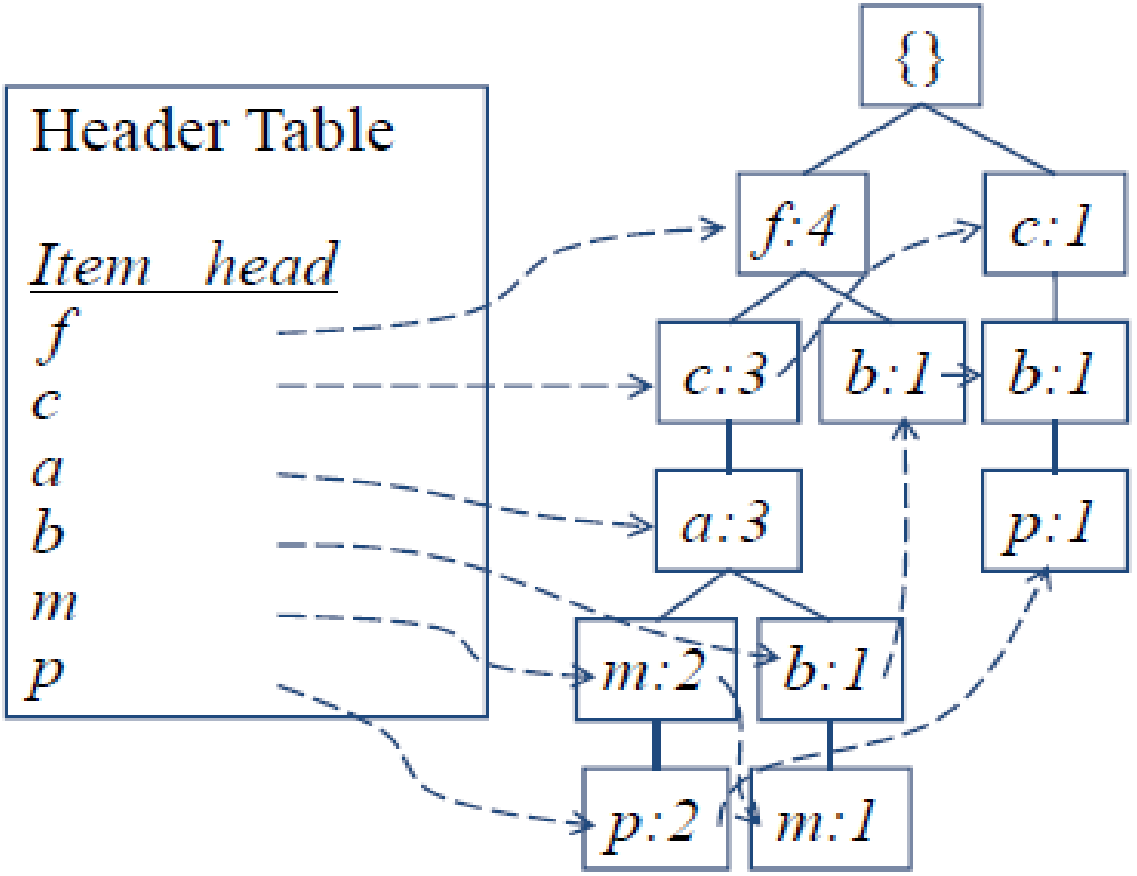
NOTE: Each transaction corresponds to one path in the FP-tree

Step 2: construct FP-tree



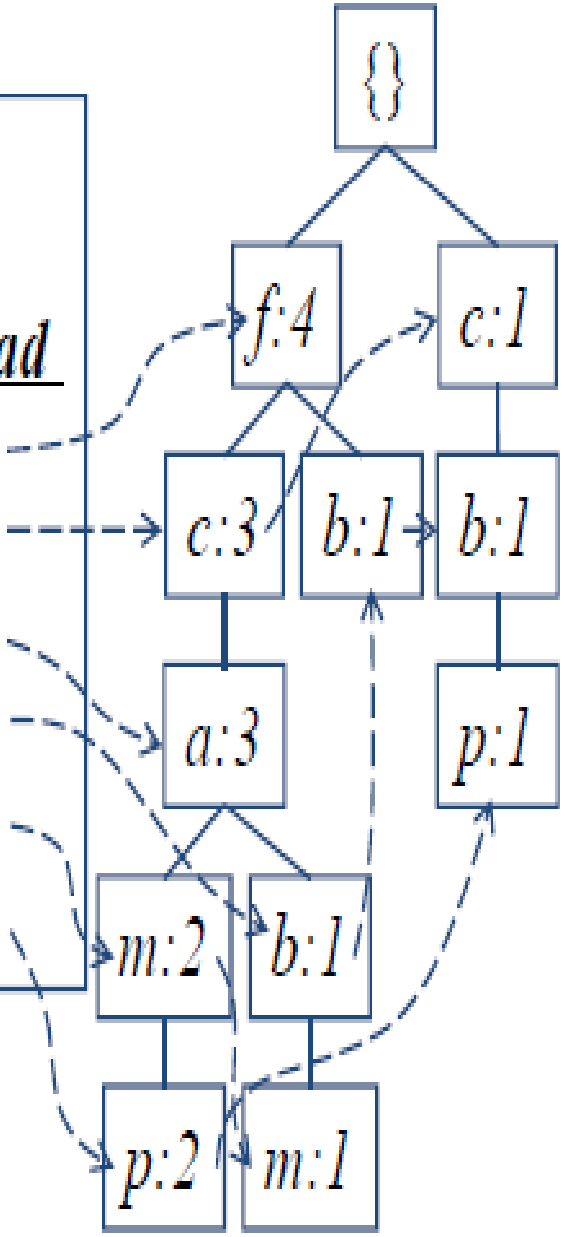
Construction Example

Final FP-tree



Header Table

Item	frequency	head
f	4	
c	4	
a	3	
b	3	
m	3	
p	3	



Conditional pattern bases

item	*cond. pattern base*
c	*f:3*
a	*fc:3*
b	*fca:1, f:1, c:1*
m	*fca:2, fcab:1*
p	*fcam:2, cb:1*

Conditional Pattern Bases and Conditional FP-Tree

Item	Conditional pattern base	Conditional FP-tree
p	{(fcam:2), (cb:1)}	{(c:3)}|p
m	{(fca:2), (fcab:1)}	{(f:3, c:3, a:3)}|m
b	{(fca:1), (f:1), (c:1)}	Empty
a	{(fc:3)}	{(f:3, c:3)}|a
c	{(f:3)}	{(f:3)}|c
f	Empty	Empty

order of L

A **Conditional Pattern Base** is the collection of **prefix paths** in the FP-tree that lead to a specific item, forming the **conditional database** for that item. It consists of **prefix itemsets** (paths in the tree) along with their associated **support counts**.

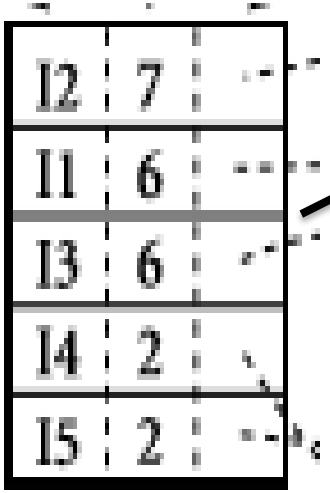
Instead of scanning the entire dataset multiple times (as in Apriori), FP-Growth extracts a smaller, more relevant tree to find frequent patterns.

It reduces computation by focusing only on relevant transactions.

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

Minimum support count be =2.

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



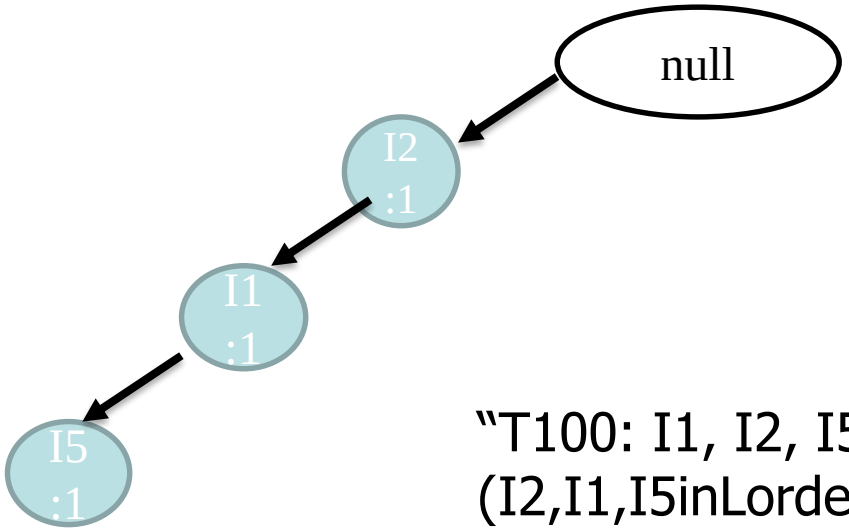
Let Minimum support count be 2.

The set of frequent items is sorted in the order of descending support count.

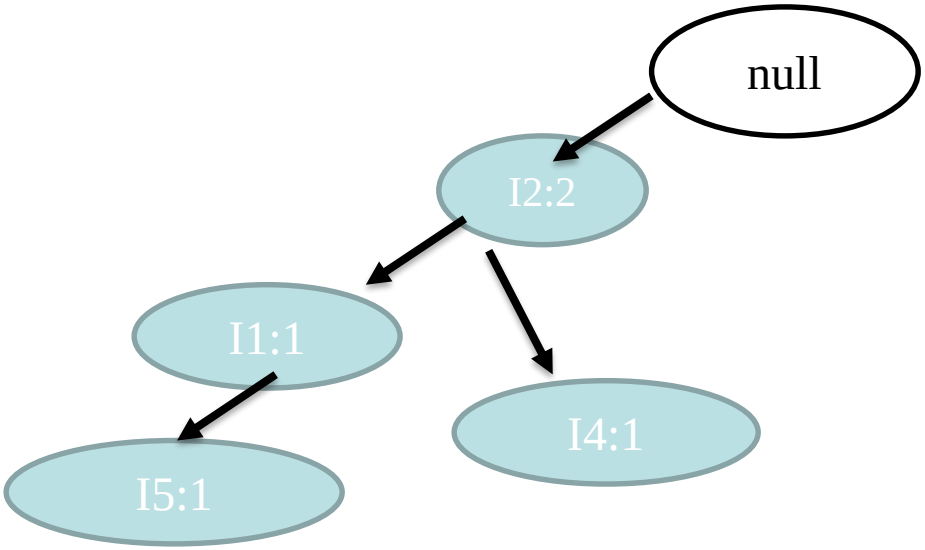
TID	List of Items
T100	I2, I1, I5
T200	I2, I4
T300	I2,I3
T400	I2, I1, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I2, I1, I3, I5

First, create the root of the tree, labeled with "null."

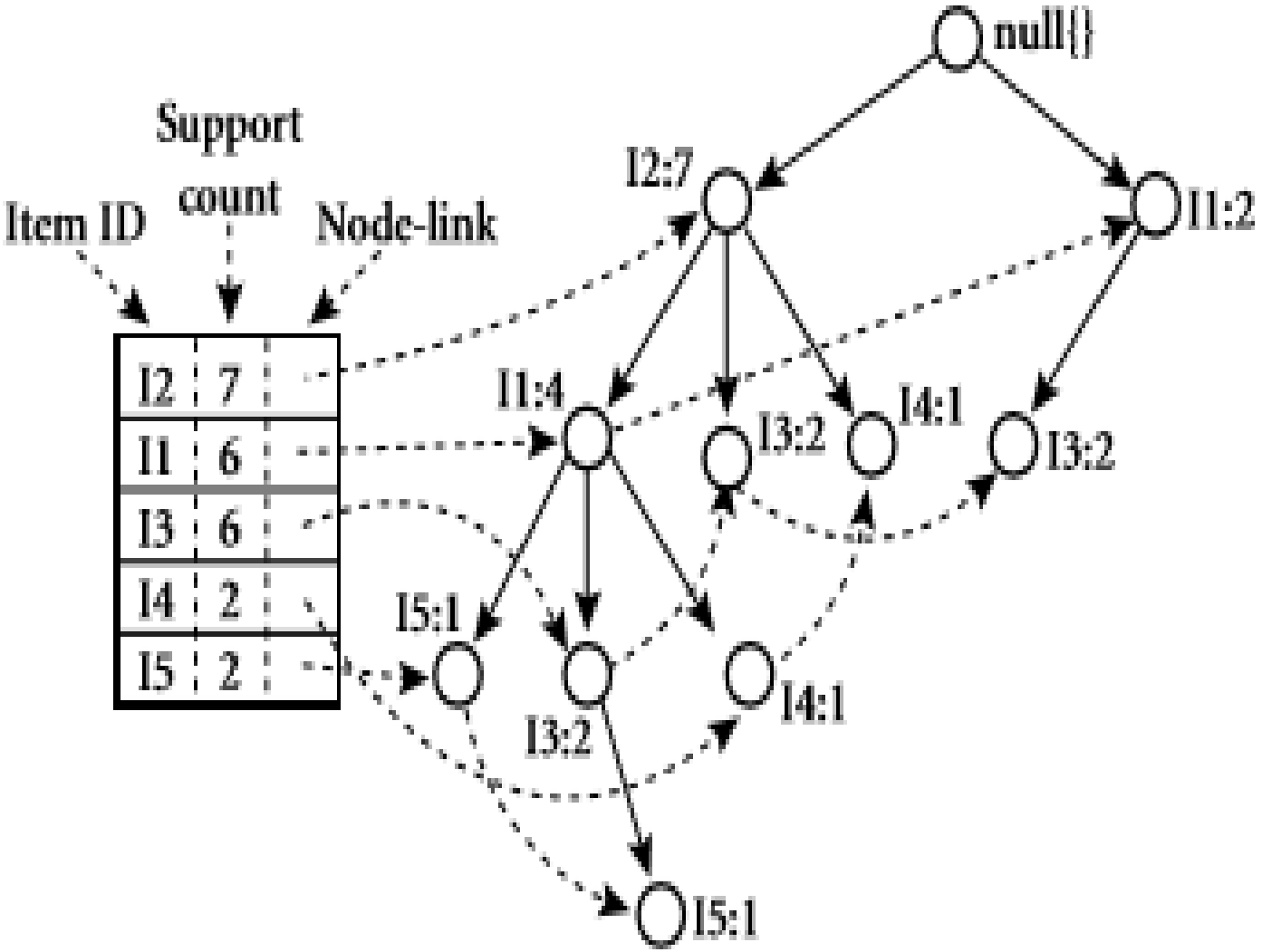
Scan database D a second time. The items in each transaction are processed in Lorder (i.e., sorted according to descending support count), and a branch is created for each transaction.



"T100: I1, I2, I5," which contains three items (I2,I1,I5inLorder), leads to the construction of the first branch of the tree with three nodes,{I2: 1}, {I1:1}, and {I5: 1}, where I2 is linked as a child of the root, I1 is linked to I2, and I5 is linked to I1.



For T200 = I2, I4


Item ID
Support count
Node-link
I2 7
I1 6
I3 6
I4 2
I5 2
null{}
I2:7
I1:2
I1:4
I3:2
I4:1
I3:2
I5:1
I3:2
I4:1
I5:1

Mining the FP-tree by creating conditional (sub-)pattern bases.

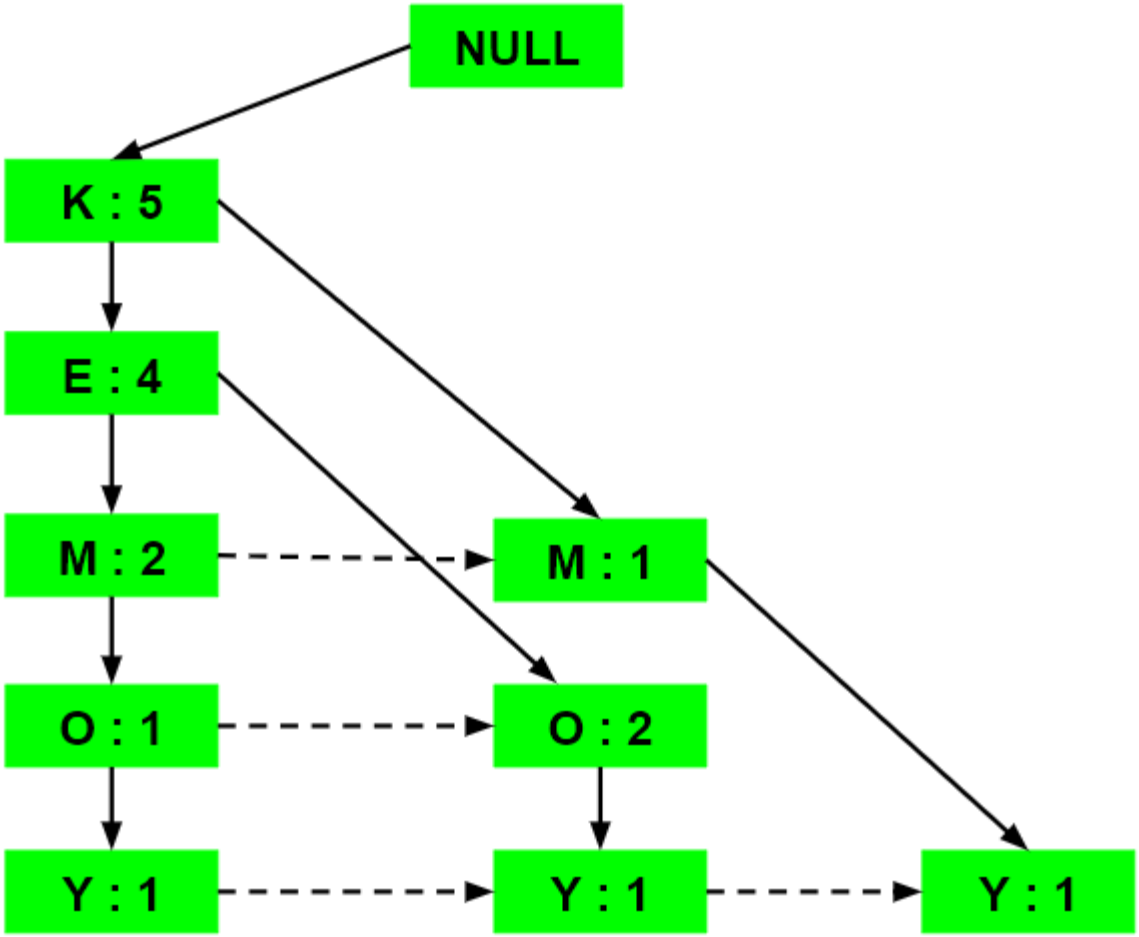
Item	*Conditional Pattern Base*	*Conditional FP-tree*	*Frequent Patterns Generated*
I5	{{I2, I1: 1}, {I2, I1, I3: 1}}	⟨I2: 2, I1: 2⟩	{I2, I5: 2}, {I1, I5: 2}, {I2, I1, I5: 2}
I4	{{I2, I1: 1}, {I2: 1}}	⟨I2: 2⟩	{I2, I4: 2}
I3	{{I2, I1: 2}, {I2: 2}, {I1: 2}}	⟨I2: 4, I1: 2⟩, ⟨I1: 2⟩	{I2, I3: 4}, {I1, I3: 4}, {I2, I1, I3: 2}
I1	{{I2: 4}}	⟨I2: 4⟩	{I2, I1: 4}

Transaction ID	Items
T1	{E,K,M,N,O,Y}
T2	{D,E,K,N,O,Y}
T3	{A,E,K,M}
T4	{C,K,M,U,Y}
T5	{C,E,I,K,O,O}

Item	Frequency
A	1
C	2
D	1
E	4
I	1
K	5
M	3
N	2
O	3
U	1
Y	3

L = {K : 5, E : 4, M : 3, O : 3, Y : 3}

Transaction ID	Items	Ordered-Item Set
T1	{E,K,M,N,O,Y}	{K,E,M,O,Y}
T2	{D,E,K,N,O,Y}	{K,E,O,Y}
T3	{A,E,K,M}	{K,E,M}
T4	{C,K,M,U,Y}	{K,M,Y}
T5	{C,E,I,K,O,O}	{K,E,O}


NULL
K : 5
E : 4
M : 2
O : 1
Y : 1
M : 1
O : 2
Y : 1
Y : 1

Items	Conditional Pattern Base
Y	{{K,E,M,O : 1}, {K,E,O : 1}, {K,M : 1}}
O	{{K,E,M : 1}, {K,E : 2}}
M	{{K,E : 2}, {K : 1}}
E	{K : 4}
K	

Items	Conditional Pattern Base	Conditional Frequent Pattern Tree
Y	{{K,E,M,O : 1}, {K,E,O : 1}, {K,M : 1}}	{K : 3}
O	{{K,E,M : 1}, {K,E : 2}}	{K,E : 3}
M	{{K,E : 2}, {K : 1}}	{K : 3}
E	{K : 4}	{K : 4}
K		

Items	Frequent Pattern Generated
Y	{<K,Y : 3>}
O	{<K,O : 3>, <E,O : 3>, <E,K,O : 3>}
M	{<K,M : 3>}
E	{<E,K : 3>}
K	

Generate Association Rule of the below transaction database where

Min support =40%, confidence = 60%.

Transaction ID	ITEMS
100	Milk, Bread, Butter
200	Milk Bread
300	Bread, Butter
400	Milk, Butter
500	Bread

Find **frequent itemsets** using **FP-Growth** with **min support = 40% (2 transactions)**

Transaction ID	Items Bought
1	A, B, D
2	B, C, E
3	A, B, C, E
4	B, E
5	A, B, C, E

INTRODUCTION TO CLUSTERING TECHNIQUES,

- Pattern Recognition
- Spatial Data Analysis
  - create thematic maps in GIS by clustering feature spaces
  - detect spatial clusters and explain them in spatial data mining
- Image Processing
- Economic Science (especially market research)
- WWW
  - Document classification
  - Cluster Weblog data to discover groups of similar access patterns

- Marketing: Help marketers discover distinct groups in their customer bases, and then use this knowledge to develop targeted marketing programs
- Land use: Identification of areas of similar land use in an earth observation database
- Insurance: Identifying groups of motor insurance policy holders with a high average claim cost
- City-planning: Identifying groups of houses according to their house type, value, and geographical location
- Earth-quake studies: Observed earth quake epicenters should be clustered along continent faults

- A good clustering method will produce high quality clusters with
  - high intra-class similarity
  - low inter-class similarity
- The quality of a clustering result depends on both the similarity measure used by the method and its implementation.
- The quality of a clustering method is also measured by its ability to discover some or all of the hidden patterns.

Clustering is the process of grouping data into classes, or clusters, so that objects within a cluster have high similarity in comparison to one another, but are very dissimilar to objects in other clusters.

Although classification is an effective means for distinguishing groups or classes of objects, it requires the often costly collection and labeling of a large set of training tuples or patterns, which the classifier uses to model each group.

It is often more desirable to proceed in the reverse direction: First partition the set of data into groups based on data similarity (e.g., using clustering), and then assign labels to the relatively small number of groups.

Clustering is also called data segmentation in some applications because clustering partitions large datasets into groups according to their similarity.

Clustering can also be used for outlier detection, where outliers (values that are “far away” from any cluster) may be more interesting than common cases.

In machine learning, clustering is an example of unsupervised learning. Unlike classification, clustering and unsupervised learning do not rely on predefined classes and class-labeled training examples.

For this reason, clustering is a form of learning by observation, rather than learning by examples.

The following are typical requirements of clustering in data mining:

Scalability: Many clustering algorithms work well on small datasets containing fewer than several hundred data objects; however, a large database may contain millions of objects. Clustering on a sample of a given large data set may lead to biased results. Highly scalable clustering algorithms are needed.

Ability to deal with different types of attributes: Many algorithms are designed to cluster interval-based (numerical) data. However, applications may require clustering other types of data, such as binary, categorical (nominal), and ordinal data, or mixtures of these data types.

Discovery of clusters with arbitrary shape: Many clustering algorithms determine clusters based on Euclidean or Manhattan distance measures. Algorithms based on such distance measures tend to find spherical clusters with similar size and density. However, a cluster could be of any shape. It is important to develop algorithms that can detect clusters of arbitrary shape.

Ability to deal with noisy data: Most real-world databases contain outliers or missing, unknown, or erroneous data. Some clustering algorithms are sensitive to such data and may lead to clusters of poor quality.

High dimensionality: A database or a data warehouse can contain several dimensions or attributes. Many clustering algorithms are good at handling low-dimensional data, involving only two to three dimensions

- Given two objects represented by the tuples (22, 1, 42, 10) and (20, 0, 36, 8):
- (a) Compute the *Euclidean distance between the two objects.*
- (b) Compute the *Manhattan distance between the two objects.*
- (c) Compute the *Minkowski distance between the two objects, using p = 3.*

- Compute the *Euclidean distance between the*

Euclidean distance is defined as

$$d(i,j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{in} - x_{jn})^2}.$$

Where i=(xi1,,xi2,….xin) and j=(xj1,sj2,…xjn) are two n-dimentional data objects

(22, 1, 42, 10) and (20, 0, 36, 8)

$$\begin{aligned} d(i,j) &= \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{in} - x_{jn})^2} \\ &= \sqrt{|22-20|^2 + |1-0|^2 + |42-36|^2 + |10-8|^2} = 6.71 \end{aligned}$$

Manhattan (or city block) distance, is defined as

$$d(i,j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{in} - x_{jn}|$$

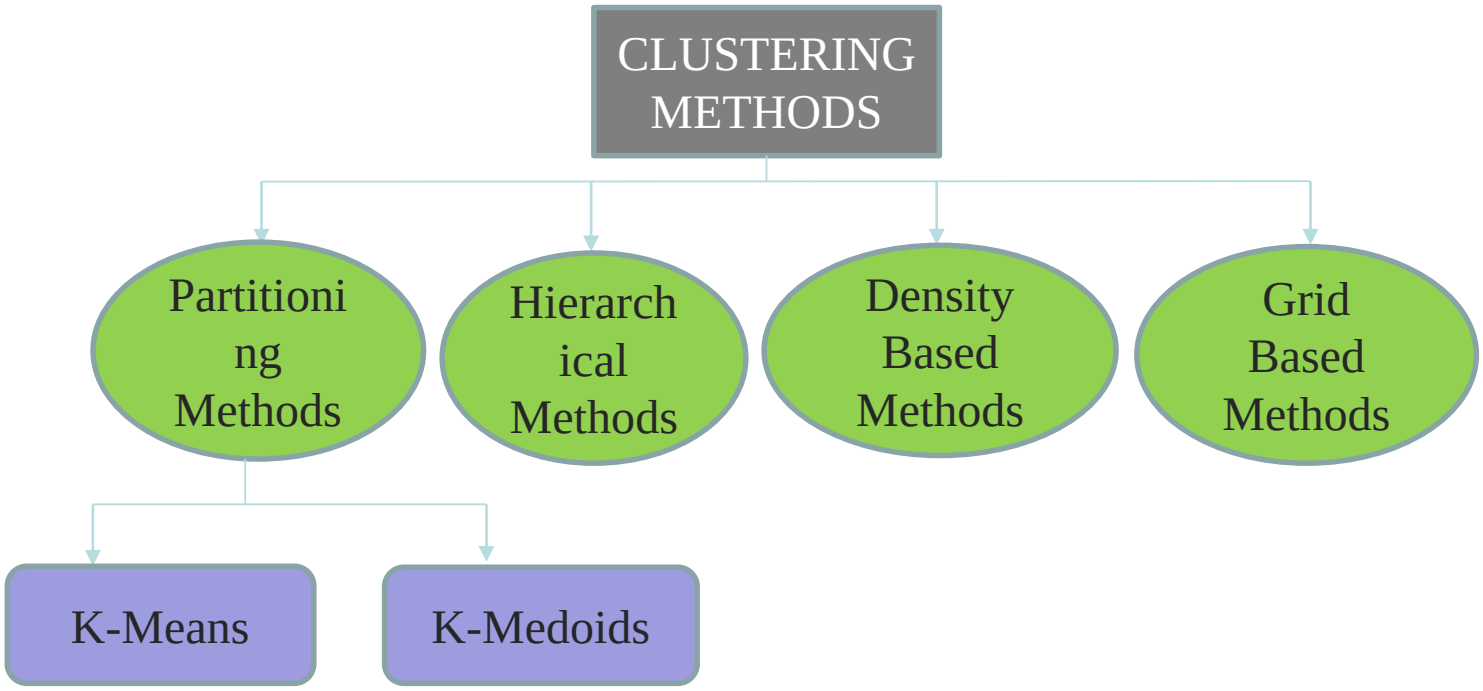
$$= |22 - 20| + |1 - 0| + |42 - 36| + |10 - 8| = 11$$

Compute the *Minkowski distance between the two objects, using p = 3.*

$$d(i,j) = \left(|x_{i1} - x_{j1}|^q + |x_{i2} - x_{j2}|^q + \cdots + |x_{in} - x_{jn}|^p\right)^{1/p}$$

$$= \left(|22 - 20|^3 + |1 - 0|^3 + |42 - 36|^3 + |10 - 8|^3\right)^{1/3} = 6.15$$

Clustering Methods



Partitioning Method

Suppose we are given a database of n objects, the partitioning method construct k partition of data.

Each partition will represents a cluster and k≤n. It means that it will classify the data into k groups, which satisfy the following requirements:

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

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.

Partitioning Method

- **Types of Partitioning Methods**
- K- Means: A Centroid-Based Technique
- K- Medoids: A Representative Object Based Technique

K- Means and K- Medoids

- **Example:**

Amazon:

- Cluster 1: Utensils(Bowls, Spoons, Forks etc.)
- Cluster 2: Mobile Accessories(Earphones, Headphones, Charger etc)
- Cluster 3: Home Accessories(Bedsheets, Pillow Covers etc)
- Cluster 4: Books

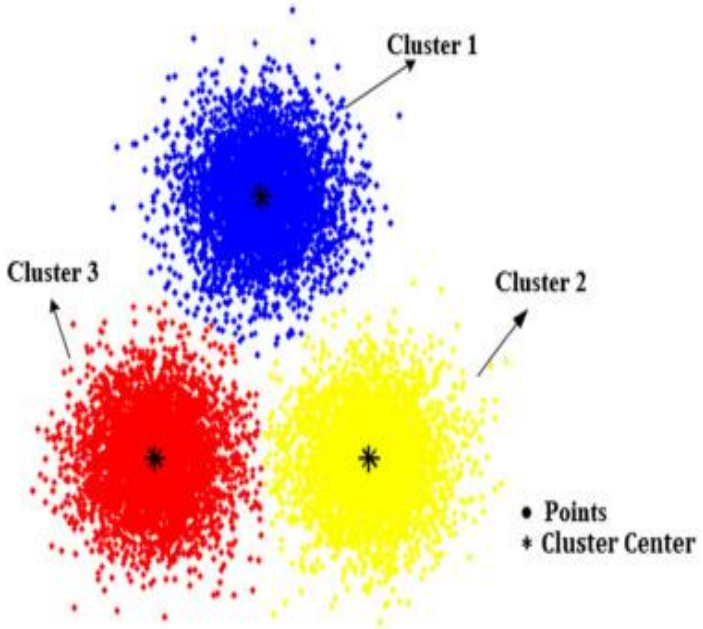
Classical Partitioning Methods: k-Means and k- Medoids

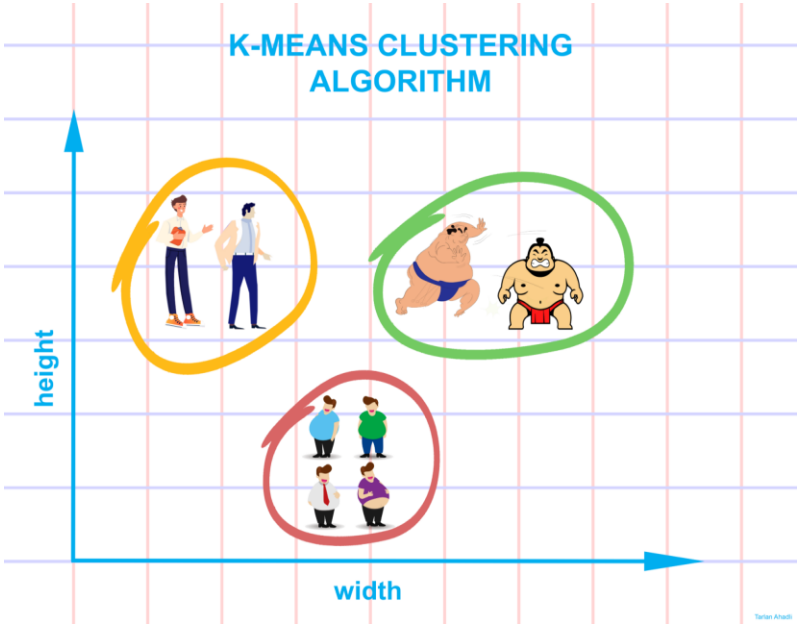
The most well-known and commonly used partitioning methods are k-means, k-medoids

Centroid-Based Technique: The k-Means Method

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

Cluster similarity is measured in regard to the mean value of the objects in a cluster, which can be viewed as the cluster’s centroid or center of gravity.


Cluster 1
Cluster 3
Cluster 2
Points
Cluster Center


K-MEANS CLUSTERING
ALGORITHM
height
width
Tarlan Ahadli

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

Input:

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

Output: A set of *k* clusters.

Method:

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

Consider the data set with the following data points. Apply K-Means Clustering Algorithm take 4 as the center for cluster 1 and 12 as center for cluster 2.

D= { 1,2,3,8,9,10)

S.No.	Objects	C=1 Center = 3	C=2 Center =10	Clusters
1	1	3-1=2	10-1=9	c1
2	2	3-2=1	10-2=8	C1
3	3	3-3=0	10-3=7	c1
4	8	3-8=5	10-8=2	c2
5	9	3-9=6	10-9=1	c2
6	10	3-10=7	10-10=0	c2

Cluster 1 = {1,2,3} cluster 2={ 8,9,10}

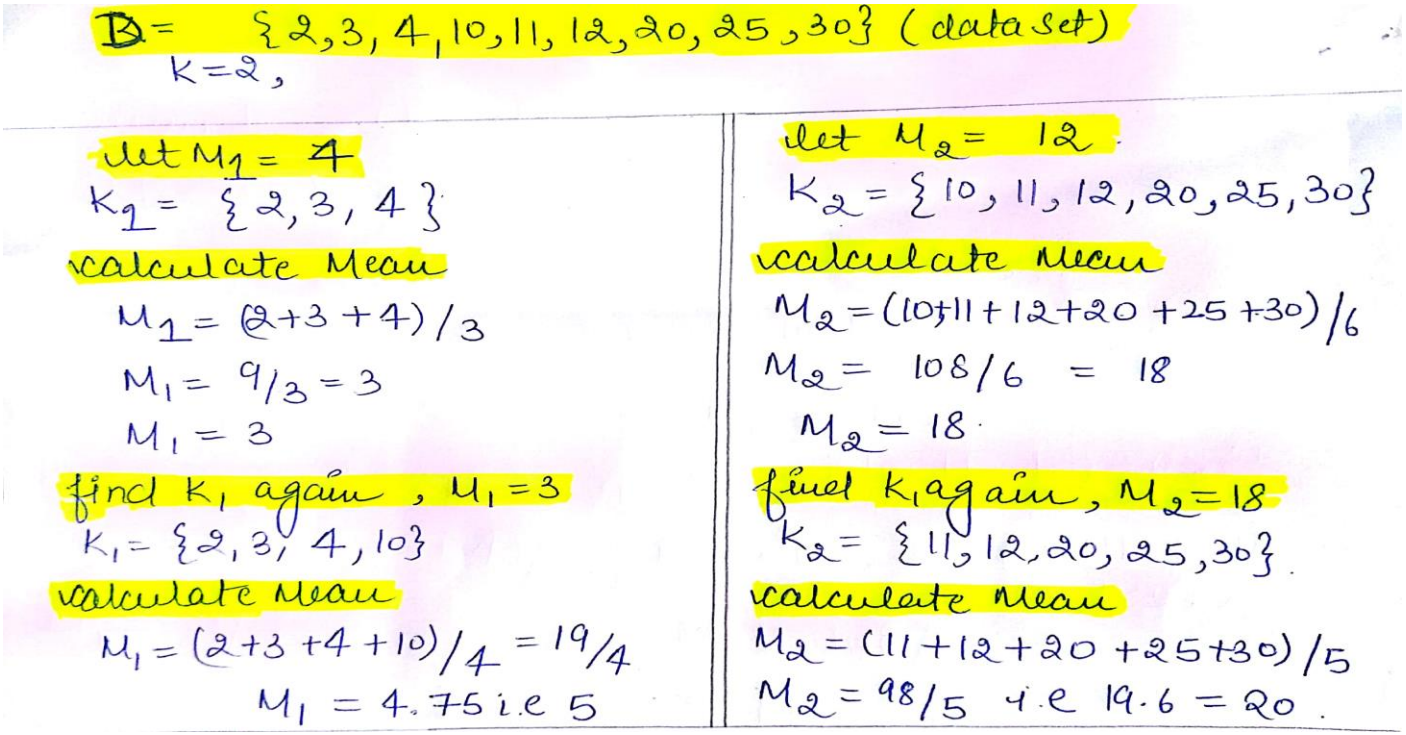
(1+2+3/3)= 2 8+9+10/3 = 9

{ 1,2,3,8,9,10}

C1=2
C2=9

S.No.	Objects	C=1 Center = 2	C=2 Center =9	Clusters
1	1	1-2=1	1-9=8	c1
2	2	2-2=0	2-9=7	c1
3	3	3-2=1	3-9=6	c1
4	8	8-2=6	8-9=1	c2
5	9	9-2=7	9-9=0	c2
6	10	10-2=8	10-9=1	c2

K-Mean Example



K-Mean Example

find K_1 again

$K_1 = \{2, 3, 4, 10, 11, 12\}$

calculate Mean

$M_1 = (2+3+4+10+11+12)/6$

$M_1 = 42/6 = 7$

$M_1 = 7$

find K_1 again

$K_1 = \{2, 3, 4, 10, 11, 12\}$

find K_2 again

$K_2 = \{20, 25, 30\}$

calculate Mean

$M_2 = (20+25+30)/3$

$= 75/3 = 25$

$M_2 = 25$

find K_1 again

$K_2 = \{20, 25, 30\}$

Distance functions

K-Means Example

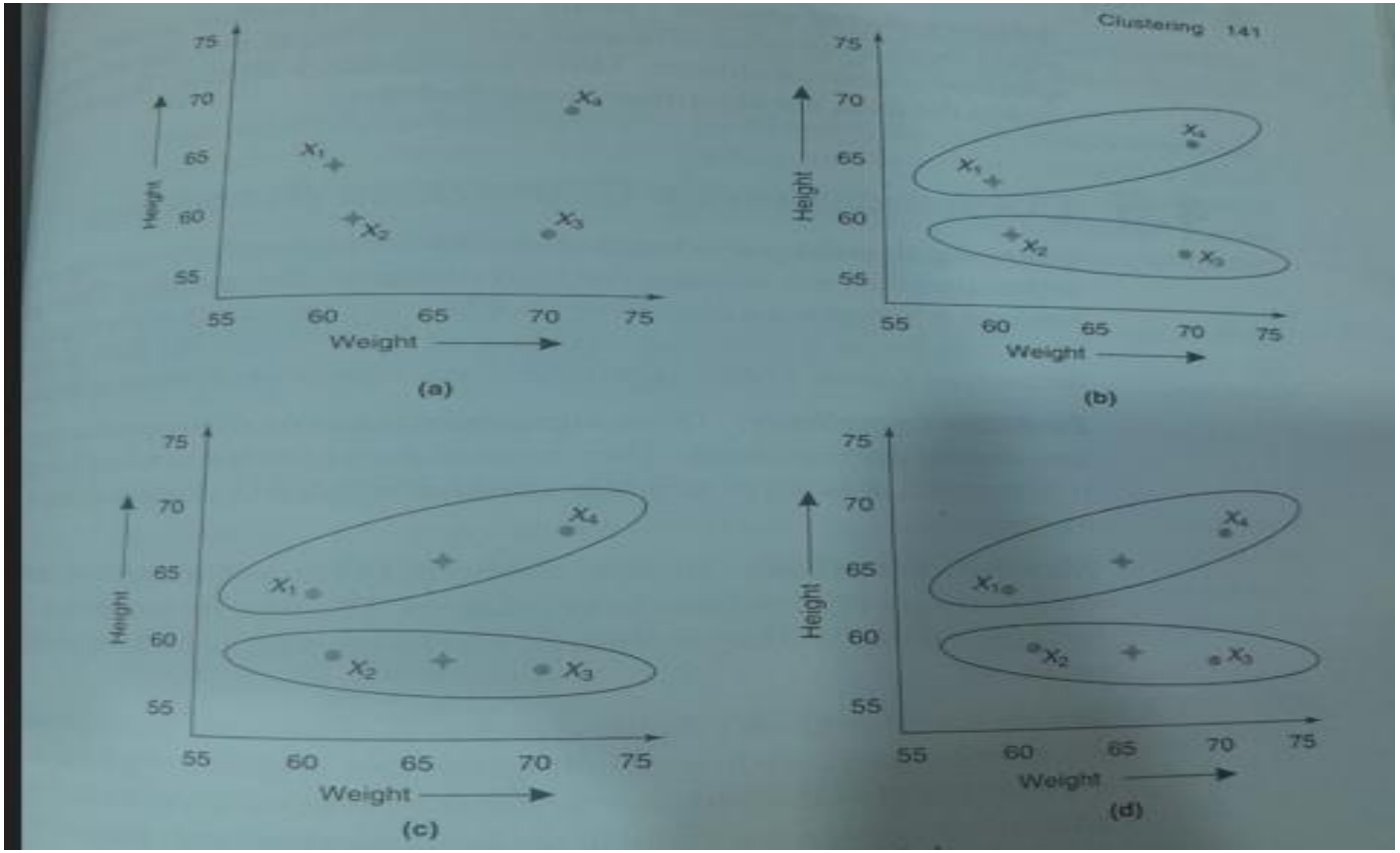
Euclidean

$$\sqrt{\sum_{i=1}^{k}(x_i - y_i)^2}$$

ID	Name	Height	Weight
X1	A	64	60
X2	B	60	61
X3	C	59	70
X4	D	68	71

Since we want 2 clusters, k=2

Initially let us assume algo selects points X1 and X2 as the cluster means. The distance of remaining points needs to be calculated from X1 and X2 using Euclidean distance


75
70
65
60
55
Height
55
60
65
70
75
Weight
X_1
X_2
X_3
X_4
(a)
(b)
(c)
(d)

X1 *	64	60
X2 *	60	61
X3	59	70
X4	68	71

$$E = \sum_{i=1}^{N} \sum_{x \in C} d(x, \bar{x}_i)^2$$

X1 and x2 as cluster means

$$d(x_3, x_1) = \sqrt{[(59-64)^2 + (70-60)^2]} = 11.18$$

$$d(x_3, x_2) = \sqrt{[(59-60)^2 + (70-61)^2]} = \mathbf{9.06}$$

$$d(x_4, x_1) = \sqrt{[(68-64)^2 + (71-60)^2]} = \mathbf{11.7}$$

$$d(x_4, x_2) = \sqrt{[(68-60)^2 + (71-61)^2]} = 12.81$$

Clearly, x_3 is closer to x_2 than x_1. Hence, it should belong to the cluster for which x_2 is a prototype. Similarly, x_4 should belong to the cluster for which x_1 is a prototype.

Computing cluster quality

$E=d(x_3,x_2)^2 + d(x_4,x_1)^2 = \mathbf{219}$

In the next iteration, the algorithm re-computes by calculating the cluster means. For the cluster with X_1 the mean is computed as (64+68)/2, (60+71)/2 which is 66 and 65.5. Similarly for cluster with X_2 the mean is (59.5, 65.5)

$$d(x_1, (66, 65.5)) = \sqrt{[(66-64)^2 + (65.5-60)^2]} = 5.85$$
$$d(x_1, (59.5, 65.5)) = \sqrt{[(59.5-64)^2 + (65.5-60)^2]} = 7.1$$
$$d(x_2, (66, 65.5)) = \sqrt{[(66-60)^2 + (65.5-61)^2]} = 7.5$$
$$d(x_2, (59.5, 65.5)) = \sqrt{[(59.5-60)^2 + (65.5-61)^2]} = 4.53$$
$$d(x_3, (66, 65.5)) = \sqrt{[(66-59)^2 + (65.5-70)^2]} = 8.32$$
$$d(x_3, (59.5, 65.5)) = \sqrt{[(59.5-59)^2 + (65.5-70)^2]} = 4.53$$
$$d(x_4, (66, 65.5)) = \sqrt{[(66-68)^2 + (65.5-71)^2]} = 5.85$$
$$d(x_4, (59.5, 65.5)) = \sqrt{[(59.5-68)^2 + (65.5-71)^2]} = 10.12$$

X1	**66**	**64**	**60**
X2	**59**	**60**	**61**
X3	**59**	**59**	**70**
X4	**66**	**68**	**71**

E=d(x1,(66,65.5))+d(x2,(59.5,65.5))+x3,(59.5.65.5))+d(x4,(66.5,65.5))=**109.49**

Quality of cluster is measured by computing Squared Error, which should be minimum.

Squared error Criterion

If some point x is erroneously placed in wrong cluster Ci, then distance between x and the centre of Ci(which is d(x,Ci) is going to be larger than the distance if x was placed in its correct cluster. Thus this distance can be considered as an error that needs to be minimized over all the possible choices of Ci

$$E = \sum_{i=1}^{N} \sum_{x \in C} d(x, \bar{x}_i)^2$$

d is the distance function and $\bar{x}_i$ is the centre of the cluster Ci. The algorithm that finds cluster in such a way that the above sum is minimized would be a good clustering algorithm

Absolute Error Criterion

The squared error criterion has one significant drawback, it is heavily affected by the presence of outliers (data points with extreme values) in the dataset. The distance of an outlier point from its cluster centre will be quite large. The square of this distance will be even larger. To avoid drawback, squaring the distances of points from their cluster centres can be avoided. The resulting measure is kbown as the absolute error criterion

$$E = \sum_{i=1}^{N} \sum_{x \in C} d(x, \bar{x}_i)$$

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

Input:

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

Output: A set of k clusters.

Method:

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

k-medoids algorithm

- Use real object to represent the cluster
  - Select ***k*** representative objects arbitrarily
  - repeat
    - Assign each remaining object to the cluster of the nearest medoid
    - Randomly select a nonmedoid object
    - Compute the total cost, S, of swapping o_j with o_{random}
    - If $S < 0$ then swap o_j with o_{random}
  - until there is no change

K-Medoids Algorithm

- **Algorithm:**

1. Select 2 Medoids
2. Calculate the distance between data points and both medoids.
3. Calculate the total cost involved in forming the clusters using medoids
4. Again choose some other medoids and repeat step 1 and step 2.

If we will get low cost, then again do the step 1 and step 2 otherwise stop

K-Medoids Example

For a given K=2, Find the cluster by using following data sets using K-Medoids

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

For a given k=2, cluster the following data set using k-Medoids. by taking Medoids (3,4) & (7,4). After that change the medoid (7,4) to (7,3)

Points	x-axis	y-axis	Manhattan distance (3,4)	Manhattan distance (7,4)
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		

For a given k=2, cluster the following data set using k-Medoids. by taking Medoids (3,4) & (7,4). After that change the medoid (7,4) to (7,3)

Points	x-axis	y-axis	Manhattan distance (3,4)	Manhattan distance (7,4)
1	7	6	$\|7-3\|+\|6-4\| = 4+2 = 6$	$\|7-7\|+\|6-4\| = 0+2 = 2$
2	2	6	$\|2-3\|+\|6-4\| = 1+2 = 3$	$\|2-7\|+\|6-4\| = 5+2 = 7$
3	3	8	$\|3-3\|+\|8-4\| = 0+4 = 4.$	$\|3-7\|+\|8-4\| = 4+4 = 8$
4	8	5	$\|8-3\|+\|5-4\| = 5+1 = 6$	$\|8-7\|+\|5-4\| = 1+1 = 2$
5	7 ⟶	4	$\|7-3\|+\|4-4\| = 4+0 = 4$	$\|7-7\|+\|4-4\| = 0+0 = 0.$
6	4	7	$\|4-3\|+\|7-4\| = 1+3 = 4$	$\|4-7\|+\|7-4\| = 3+3 = 6$
7	6	2	$\|6-3\|+\|2-4\| = 3+2 = 5$	$\|6-7\|+\|2-4\| = 1+2 = 3$
8	7	3	$\|7-3\|+\|3-4\| = 4+1 = 5$	$\|7-7\|+\|3-4\| = 0+1 = 1$
9	6	4	$\|6-3\|+\|4-4\| = 3+0 = 3$	$\|6-7\|+\|4-4\| = 1+0 = 1$
10	3 ⟶	4	$\|3-3\|+\|4-4\| = 0+0 = 0$	$\|3-7\|+\|4-4\| = 4+0 = 4.$

Now

$K_1 = \{(3,4),(4,6),(3,8),(4,7)\}$

Now

$K_2 = \{(7,4),(7,6),(8,5),(6,4),(7,$

$(6,4)\}$

Now

$K_1 = \{(3,4), (2,6), (3,8), (4,7)\}$

Cost for k_1 by (3,4)

$= \{(|3-3|+|4-4|)+(|3-2|+|4-6|)$
$+(|3-3|+|4-8|)+(|3-4|+(4-7|)\}$

$= 0+(1+2)+(4)+(1+3)$

$= 0+3+4+4 = 11$

Cost $(K_1) = 11$

Now

$K_2 = \{(7,4), (7,6), (8,5), (6,2), (7,$
$(6,4)\}$

Cost for K_2 by (7,4).

$= \{(|7-7|+|4-4|)+(|7-7|+|4-6|)$
$+(|7-8|+|4-5|)+(|7-6|+|4-2|)$
$+(|7-7|+|4-3|)+(|7-6|+(4-4|)$

$= 0+(0+2)+(1+1)+(1+2)$
$+(0+1)+(1+0)$

$= 0+2+2+3+1+1$

$= 9$

Cost $(K_2) = 9$.

Total cost $(K_1 \& K_2)$ = $11+9$ ie $=20$.
by medoids (3,4) & (7,4)

Now change the medoids range.

i.e 3,4 — same i.e (3,4)

7,4 — (7,3)

Points	x-axis	y-axis	Manhattan distance (3,4)	Manhattan distance (7,3)
1	7	6	6	$\|7-7\|+\|6-3\| = 0+3 = 3$
2	2	6	3	$\|2-7\|+\|6-3\| = 5+3 = 8$
3	3	8	4	$\|3-7\|+\|8-3\| = 4+5 = 9$
4	8	5	6	$\|8-7\|+\|5-3\| = 1+2 = 3$
5	7	4	4	$\|7-7\|+\|4-3\| = 0+1 = 1$
6	4	7	4	$\|4-7\|+\|7-3\| = 3+4 = 7$
7	6	2	5	$\|6-7\|+\|2-3\| = 1+1 = 2$
8	7 →	3	5	$\|7-7\|+\|3-3\| = 0+0 = 0$
9	6	4	3	$\|6-7\|+\|4-3\| = 1+1 = 2$
10	3 →	4	0	$\|3-7\|+\|4-3\| = 4+1 = 5$

Now

$K_1 = \{(3,4), (2,6), (3,8), (4,7)\}$

Now

$K_2 = \{(7,3), (7,6), (8,5), (7,4), (6,2), (6,4)\}$.

K-Medoids Example

Cost K_1 = 0 + 3 + 4 + 4

cost K_1 = 11

cost(K_2) = 0 + 3 + 3 + 1 + 2 + 2

= 11

total cost i.e $K_1 + K_2$ = 11 + 11 = 22

hence Medoids will be same as (3,4) and (7,4) because cost is lower in these medoids as compared to (3,4) and (7,3) medoids.

Apply k-**Medoids** algorithm take **C1 -(4, 5)** and **C2 -(8, 5) as cluster initial medoids. Calculate the cost**

Then swap c2 medoids (8,4) with (8,5) and find the cost of swap.

	X	Y
0	8	7
1	3	7
2	4	9
3	9	6
4	**8**	**5**
5	5	8
6	7	3
7	8	4
8	7	5
9	**4**	**5**

Swap Cost (s) = New Cost – Previous Cost
If s>0 undo swap.

Point	Coordinates
A1	(2, 6)
A2	(3, 8)
A3	(4, 7)
A4	(6, 2)
A5	(6, 4)
A6	(7, 3)
A7	(7,4)
A8	(8, 5)
A9	(7, 6)
A10	(3, 4)

Initial centers are

C1 = (3, 4)

C2 = (7, 3)

Then swap with

C1=(3, 4) and C2= (6, 4).

Hierarchical Methods

A hierarchical clustering method works by grouping data objects into a tree of clusters.

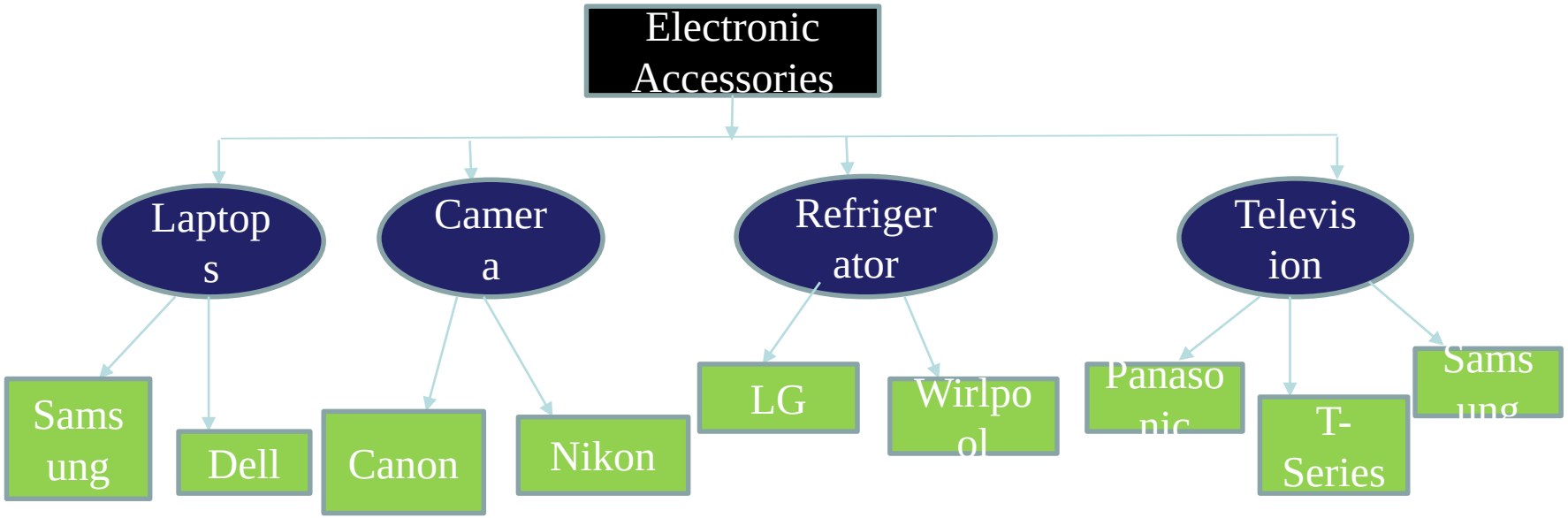
Hierarchical clustering methods can be further classified as either agglomerative or divisive, depending on whether the hierarchical decomposition is formed in a bottom-up (merging) or top-down (splitting) fashion.

Agglomerative and Divisive Hierarchical Clustering

In general, there are two types of hierarchical clustering methods:

Agglomerative hierarchical clustering: This bottom-up strategy starts by placing each object in its own cluster and then merges these atomic clusters into larger and larger

Hierarchical Methods



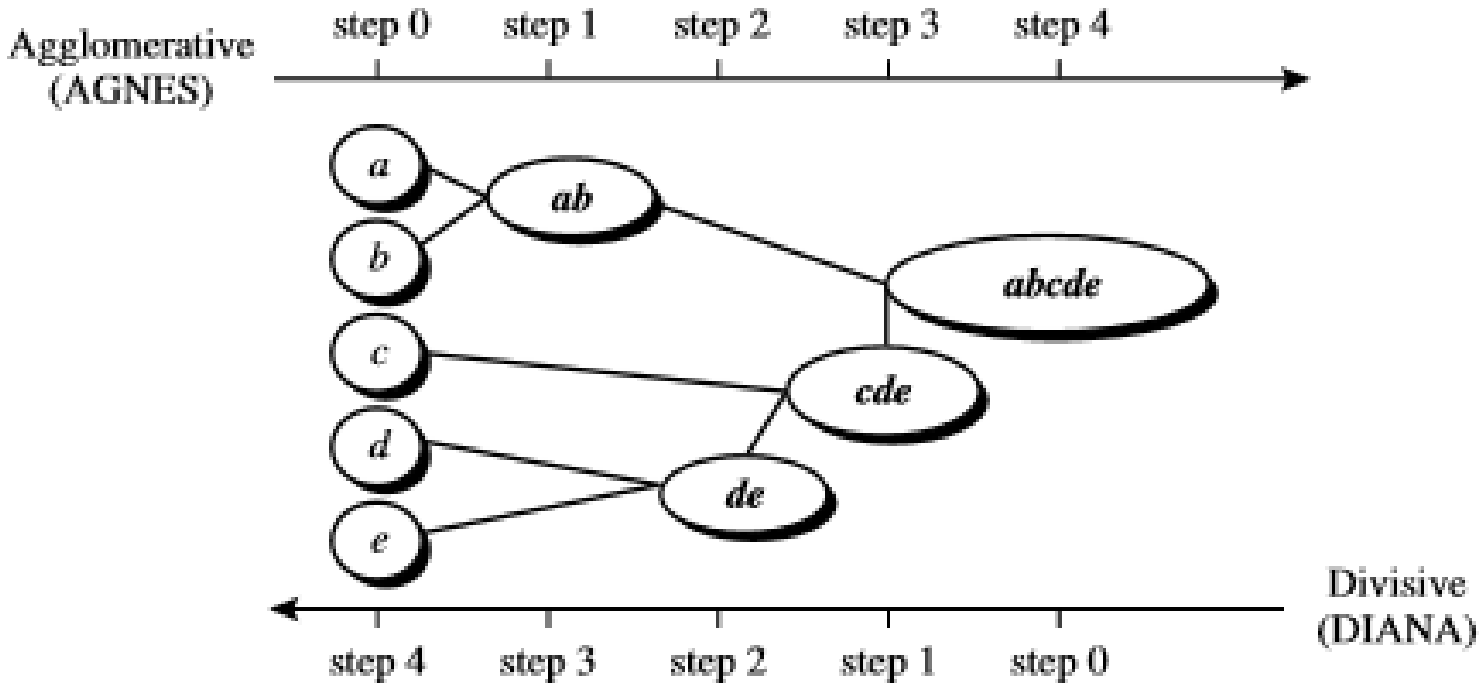
clusters, until all of the objects are in a single cluster or until certain termination conditions are satisfied.

Divisive hierarchical clustering: This top-down strategy does the reverse of agglomerative hierarchical clustering by starting with all objects in one cluster.

It subdivides the cluster into smaller and smaller pieces, until each object forms a cluster on its own or until it satisfies certain termination conditions, such as a desired number of clusters is obtained or the diameter of each cluster is within a certain threshold.

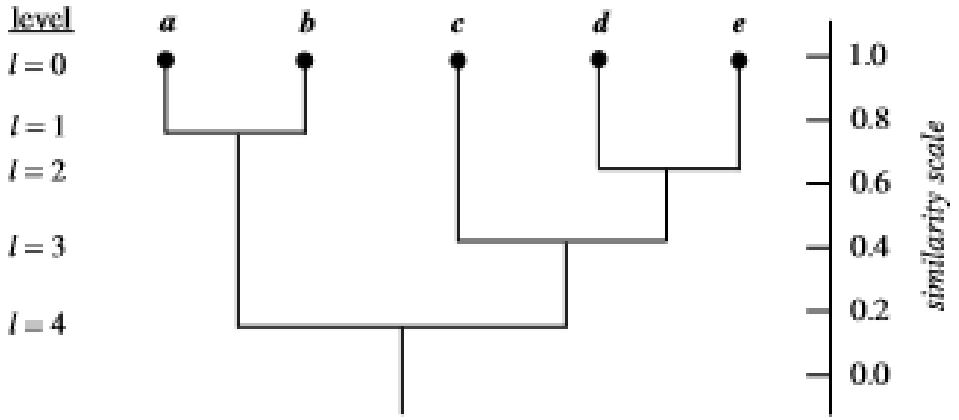
Disadvantage

This method is rigid i.e. once merge or split is done, It can never be undone.



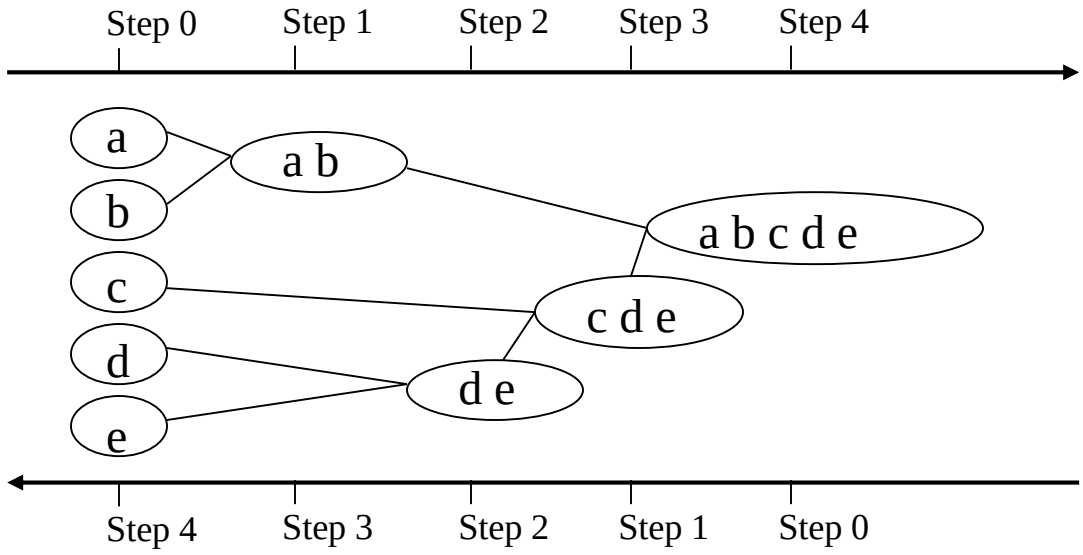
AGNES (AGglomerative NESting)

A tree structure called a dendrogram is commonly used to represent the process of hierarchical clustering. It shows how objects are grouped together step by step.



Dendrogram representation for hierarchical clustering of data objects $\{a, b, c, d, e\}$.

- Use distance matrix as clustering criteria. This method does not require the number of clusters ***k*** as an input, but needs a termination condition



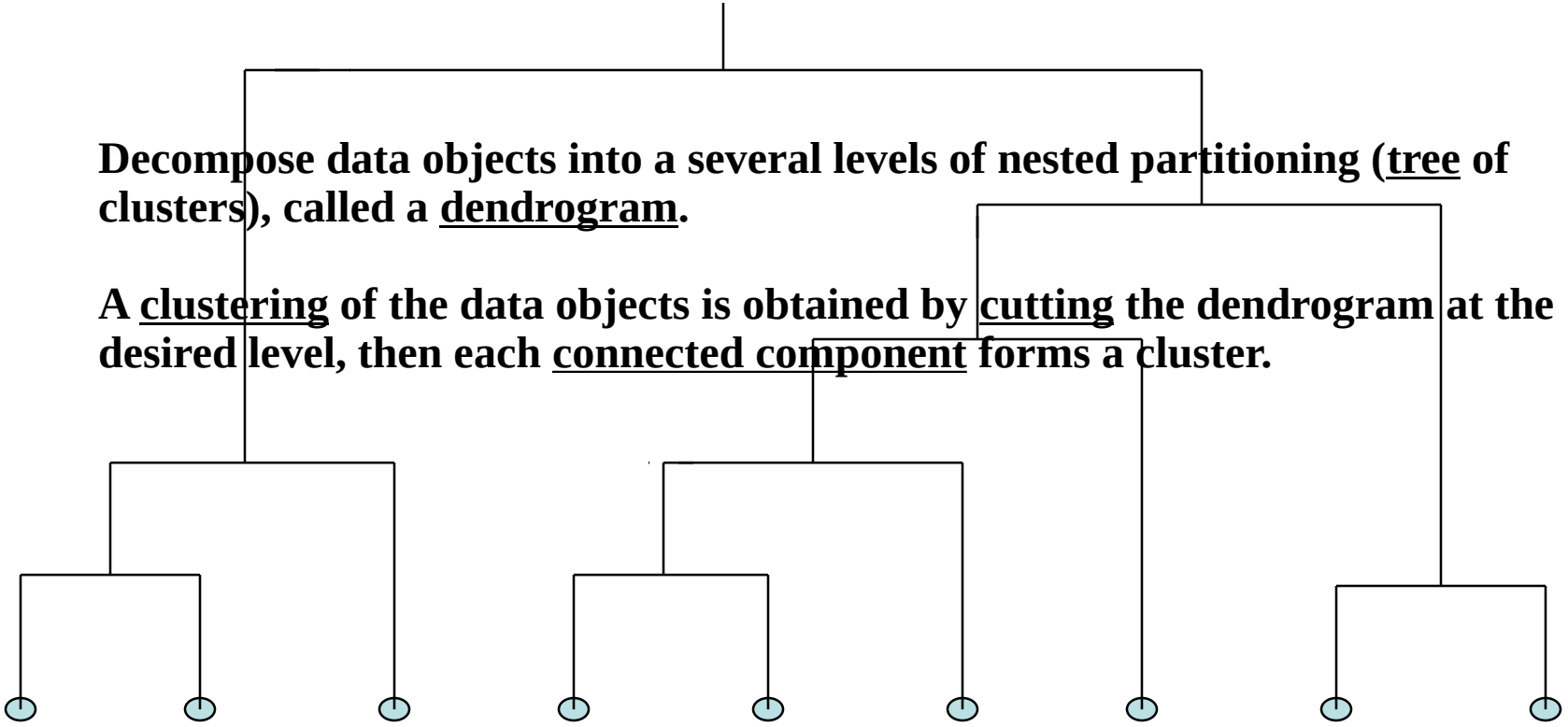
agglomerative (AGNES)

Bottom-up
Places each object in its own cluster & then merges

Top-down
All objects in a single cluster & then subdivides

divisive (DIANA)

A *Dendrogram* Shows How the Clusters are Merged Hierarchically



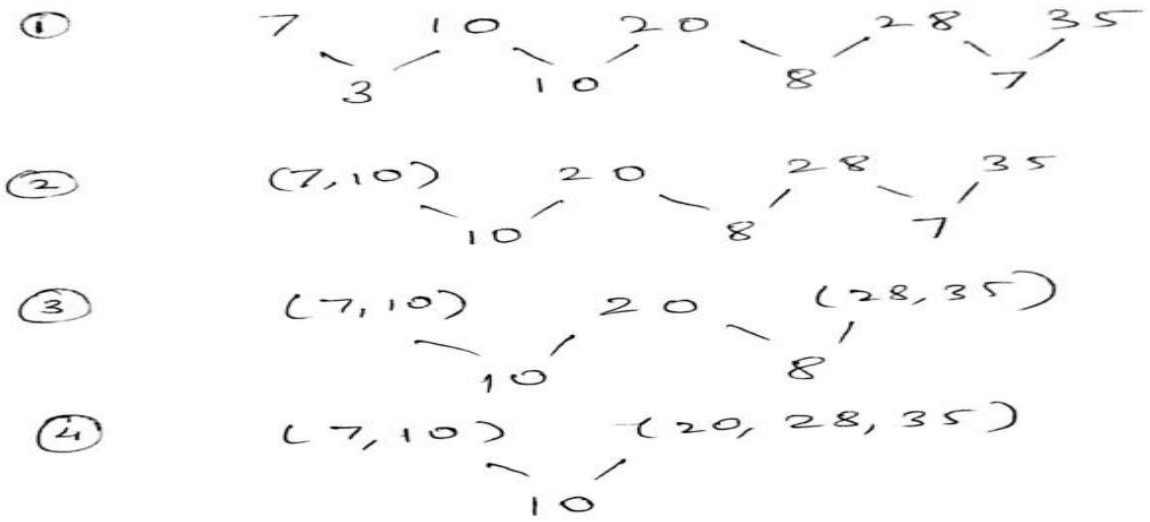
Decompose data objects into a several levels of nested partitioning (tree of clusters), called a dendrogram.

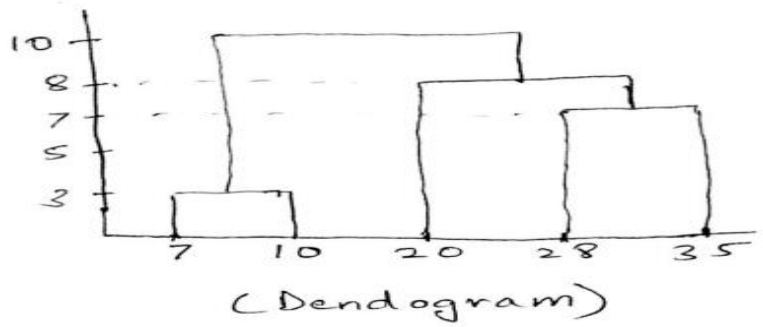
A clustering of the data objects is obtained by cutting the dendrogram at the desired level, then each connected component forms a cluster.

For the one dimensional data set **{7,10,20,28,35}**, perform hierarchical clustering and plot the dendogram to visualize it.

Single Linkage : In single link hierarchical clustering, we merge in each step the two clusters, whose two closest members have the smallest distance.

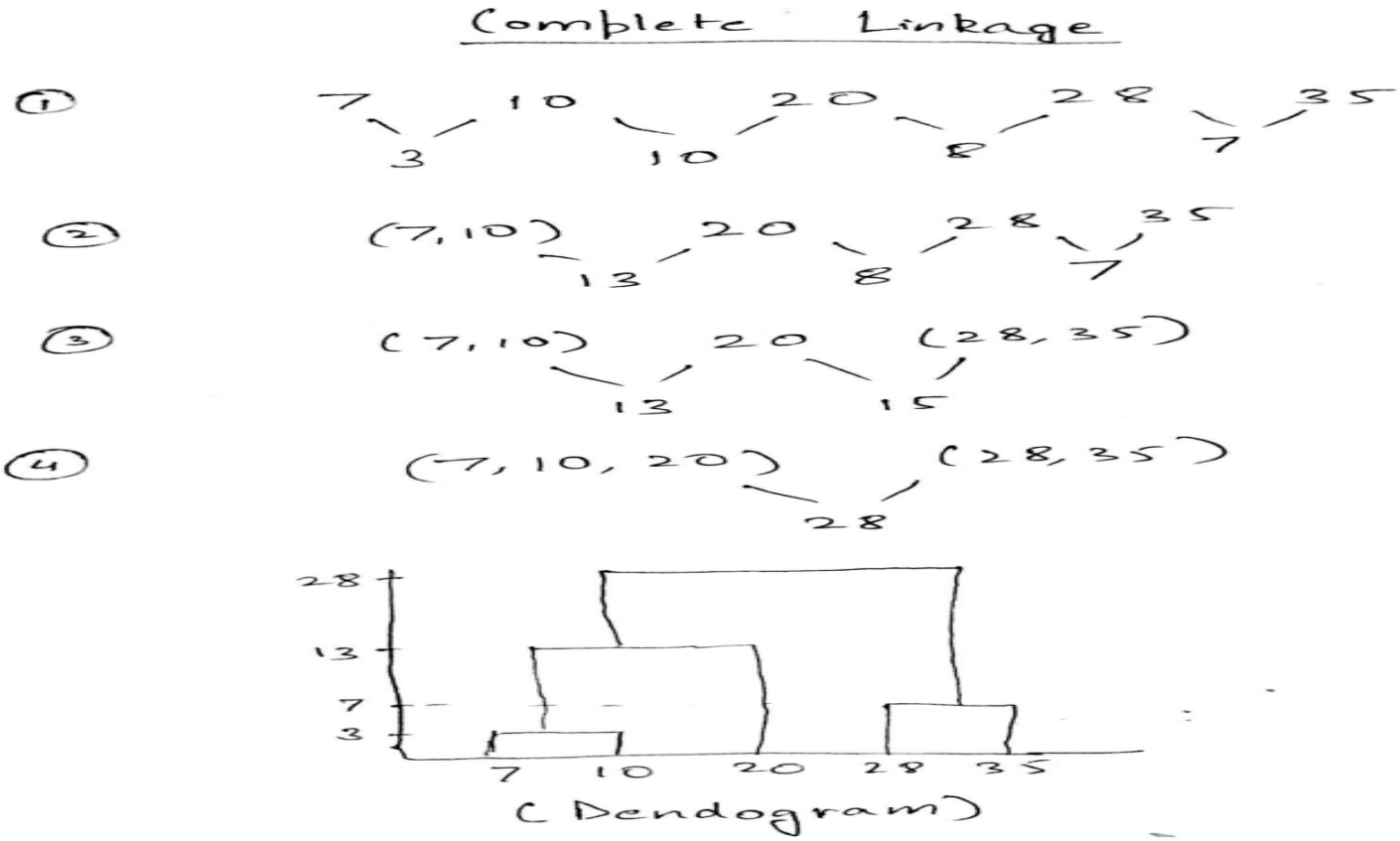
Single Linkage





(Dendogram)

2. **Complete Linkage :** In complete link hierarchical clustering, we merge in the members of the clusters in each step, which provide the smallest maximum pairwise distance.



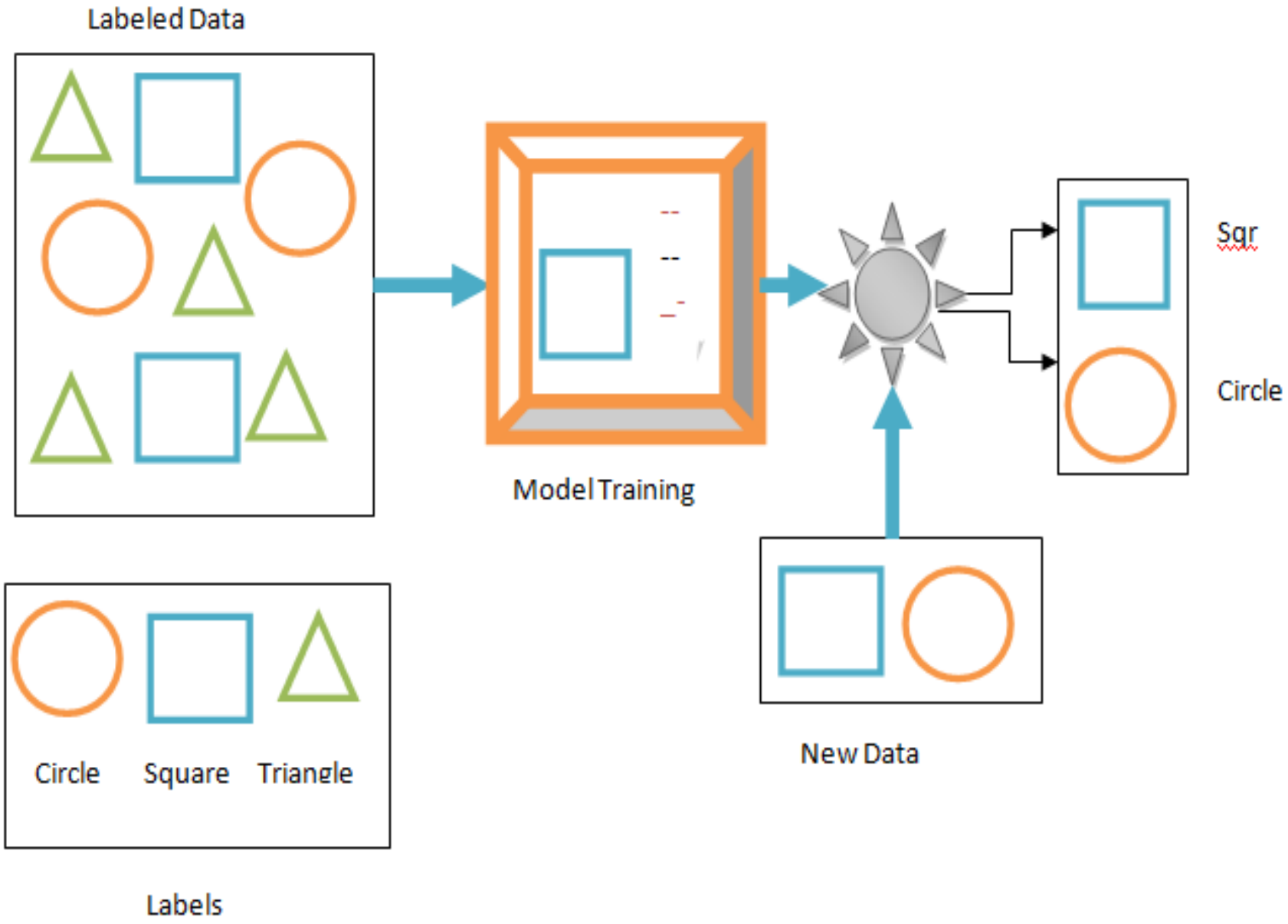
Supervised learning is an approach to machine learning that uses labeled data sets to train algorithms in order to properly classify data and predict outcomes.

Supervised Learning

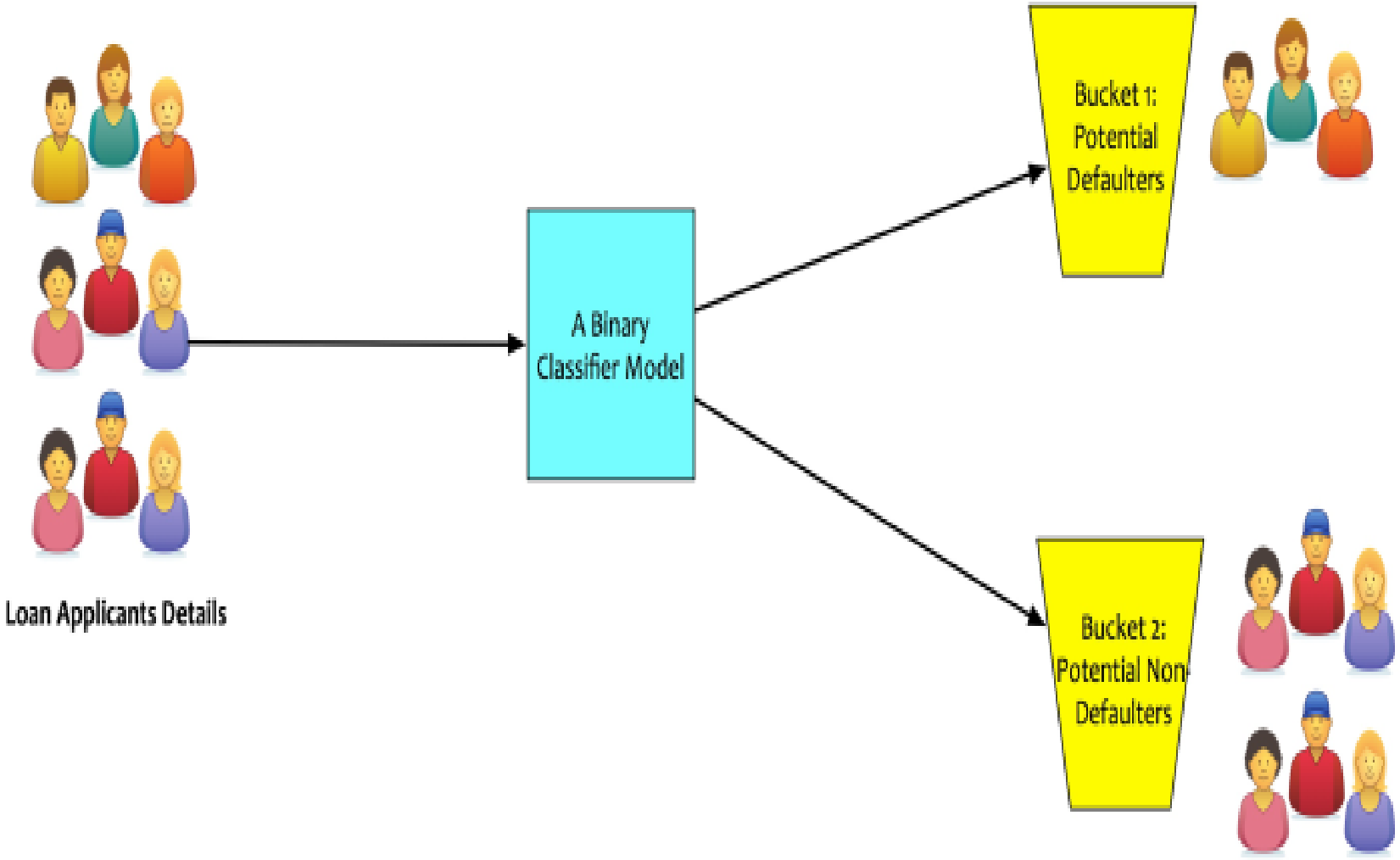
Uses labeled data to build a predictive model (answer is known).

Classification: It is about predicting a label or class.

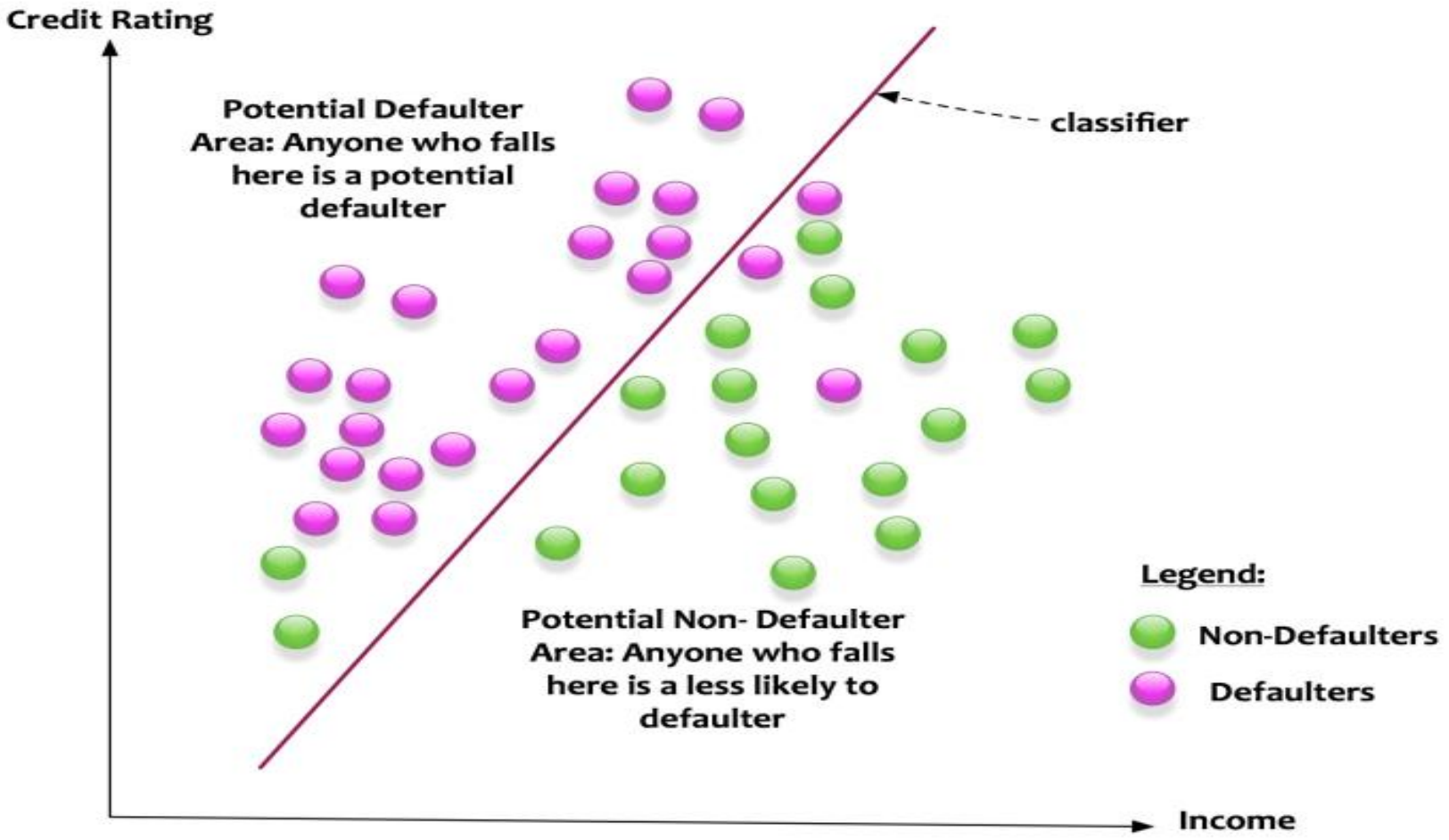
Regression: It is about predicting a continuous quantity


Labeled Data
Model Training
Sqr
Circle
New Data
Circle
Square
Triangle
Labels

Supervised Learning



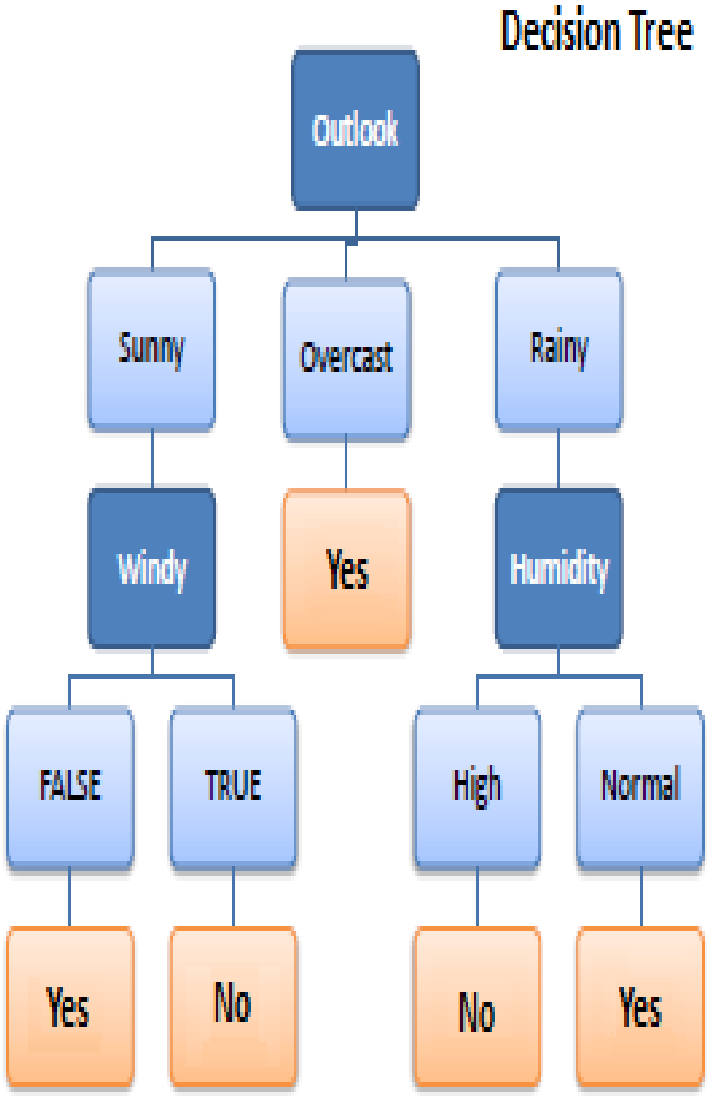
Supervised Learning



Supervised Learning : Decision Tree Example

Predictors | Target

Outlook	Temp.	Humidity	Windy	Play Golf
Rainy	Hot	High	False	No
Rainy	Hot	High	True	No
Overcast	Hot	High	False	Yes
Sunny	Mild	High	False	Yes
Sunny	Cool	Normal	False	Yes
Sunny	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Rainy	Mild	High	False	No
Rainy	Cool	Normal	False	Yes
Sunny	Mild	Normal	False	Yes
Rainy	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Sunny	Mild	High	True	No



UNIT 5 : Overview of Advanced Features of Data Mining

Mining complex data objects, Spatial databases, Multimedia databases, Time series and Sequence data; mining Text Databases and mining Word Wide Web.

Introduction to Complex Data Objects

Traditional data mining deals with structured, relational data.

Complex data includes:

- Spatial data - Information about the physical location and shape of objects.
- Temporal data
- Multimedia data (images, videos)
- Text and web data

Spatial Databases

Definition: Databases optimized to store and query data related to objects in space (e.g., maps, satellite images, GPS data).

Store spatial objects such as points (e.g., city), lines (e.g., rivers), and regions (e.g., country borders).

Spatial Data Types

Geometric Data:

- Points (e.g., store location)
- Lines (e.g., roads)
- Polygons (e.g., land parcels)

Raster Data:

- Grid-based, often from satellite imagery (e.g., temperature, elevation)

Spatial Data Mining Tasks

Spatial Association Rules:

E.g., "If a region has high rainfall, then high crop yield“

Spatial Clustering:

Discovering groups of spatial objects close to each other (e.g., crime hotspots)

Spatial Classification:

Classifying regions based on spatial and non-spatial attributes

Spatial Outlier Detection:

Identifying objects that deviate from spatial neighborhood norms

Applications of Spatial Data Mining

- Urban planning and management
- Environmental monitoring
- Disaster management (e.g., flood risk zones)
- Location-based services
- Agriculture and land use analysis

Multimedia databases store media content in various forms:

Text: Documents, web pages, subtitles.

Images: Photographs, medical scans, satellite pictures.

Audio: Music, voice recordings, podcasts.

Video: Movies, security footage, YouTube content.

Mixed Media: Slideshows, animations, or presentations.

Classification – Categorizing images, videos, etc. (e.g., tagging animals in pictures)

Clustering – Grouping similar items (e.g., finding similar songs)

Association Rule Mining – Discovering links (e.g., users who like X image also like Y)

Similarity Search/Retrieval – Content-based retrieval using features like color, shape, texture, audio tone

Summarization – Automatic summarization of video or audio

Annotation – Automatically labeling multimedia content

Applications

- **Surveillance systems**
- **Medical imaging**
- **Social media analysis**
- **Video content recommendation**
- **Voice assistants**
- **Autonomous vehicles (sensor + image fusion)**

Time Series Data

A time series is a sequence of data points indexed in time order, typically at regular intervals (e.g., stock prices, sensor data, heart rate, weather).

Key Tasks:
Trend Analysis – Long-term movement over time

Seasonality Detection – Patterns that repeat (daily, weekly, etc.)

Anomaly Detection – Finding unexpected patterns (e.g., a sudden spike in CPU usage)
Forecasting – Predicting future values (e.g., sales, temperatures)

Segmentation – Dividing time series into meaningful parts

Classification – Categorizing entire sequences (e.g., ECG patterns for disease)

Sequence Data Mining

Sequence data represents items in a particular order, which may not be timestamped. Examples:

DNA sequences

Clickstream data (user behavior)

Shopping baskets (market basket analysis over time)

Text/logs/chat messages

Techniques & Tools:

ARIMA / SARIMA – Classical forecasting

Fourier / Wavelet Transforms – Frequency analysis

Dynamic Time Warping (DTW) – Similarity between sequences

LSTM / GRU (RNNs) – Deep learning for sequences

Prophet (by Meta) – Time series forecasting tool

TSFresh / Kats / Darts – Python libraries for time series mining

Text mining is the process of extracting useful patterns, trends, or knowledge from unstructured or semi-structured **text data**. Examples of sources:

- News articles
- Research papers
- Social media posts
- Emails
- Reviews

Text Database

Ttext database is a large collection of text documents.

Examples: News articles, emails, books, webpages, tweets
.
Unlike structured databases, text data is semi-structured or unstructured.

Basic Process of Text Mining

Step 1: Text Preprocessing

Tokenization: Split text into words (tokens).

Stopword removal: Remove common words (the, is, at, etc.).

Stemming / Lemmatization: Reduce words to root form (e.g., "running" → "run").

Step 2: Text Representation

Bag of Words (BoW) model:

Represent each document by the **frequency** of words.

Ignores grammar and word order.

TF-IDF (Term Frequency - Inverse Document Frequency):Gives more importance to **rare but meaningful** words.

TF-IDF is a **numerical statistic** used to **evaluate how important a word is** to a **document** in a **collection (corpus)**.

TF (Term Frequency): How often a word appears in a document.
IDF (Inverse Document Frequency): How rare or common a word is across all documents.

If a word appears in **every document** (like "the", "is"), it's probably **not useful** for distinguishing documents.

If a word is **unique** or **rare**, it probably tells us **more about that document's topic**.

Term Frequency (TF)
Measures **how often** a word appears in a document:

Measures how often a word appears in a document:

$$TF(t, d) = \frac{\text{Number of times term } t \text{ appears in document } d}{\text{Total number of terms in document } d}$$

Example:
If "data" appears 3 times in a document of 100 words:
TF(data)=3/100 =0.03

Applications of TF-IDF
Search engines: Ranking documents based on query relevance.
Text classification: Feature extraction for machine learning models.
Keyword extraction: Finding important terms in documents.
Information retrieval: Matching user queries to relevant documents.

Word Embeddings (advanced):

Word embeddings are a way to **represent words** as **dense vectors** (real numbers) in a **continuous vector space** where **similar words** have **similar representations**.

mathematical way to **capture the meaning and relationship between words**.

n a good embedding space:
"king" is close to "queen"
"Paris" is close to "France"
"walking" is close to "running"

Word2Vec, GloVe, BERT — capture **semantic meaning** of words.

Popular Word Embedding Techniques

Technique	Highlights
Word2Vec (Google)	Predicts context words from a target word (Skip-gram) or target word from context (CBOW).
GloVe (Stanford)	Captures global word co-occurrence statistics (counts how often words appear together).
FastText (Facebook)	Improves Word2Vec by considering subword information (good for rare words).
ELMo (Allen Institute)	Generates **contextual** embeddings — the meaning of a word changes depending on sentence context.
BERT (Google)	Deep contextual embeddings from Transformer models — uses attention mechanisms.

Step 3: Text Mining Tasks
Classification: Categorize documents (e.g., spam vs. not spam).

Clustering: Group similar documents together without labels.

Association rule mining: Find interesting patterns (e.g., "people who read X often read Y").

Summarization: Automatically create summaries of documents.

Information extraction: Extract structured information (like names, dates) from text.

Topic modeling: Discover abstract topics (e.g., LDA - Latent Dirichlet Allocation).

Applications of Text Mining

- Sentiment analysis (detecting emotions in reviews).
- Automatic translation.
- Chatbots and virtual assistants.
- Recommender systems (e.g., suggesting articles).
- Fraud detection (e.g., analyzing emails or reports).

Information Retrieval (IR) – Find relevant documents (e.g., search engines)

Text Classification – Label documents (e.g., spam detection, sentiment analysis)

Clustering – Group similar documents (e.g., topic modeling)
Sentiment Analysis – Detect opinions/emotions

Named Entity Recognition (NER) – Extract names, dates, locations

Summarization – Extractive/abstractive

summariesTopic Modeling – Discover themes/topics (e.g., LDA)

Relation Extraction – Identify relationships between entities

Web Mining

Web mining is the application of data mining techniques to discover patterns from web data: content, structure, and usage.

1. Web Content Mining
Extracting data from web pages (text, images, video, audio)
Example: Scraping product reviews and analyzing sentiment
2. Web Structure Mining
Analyzing links (like in a graph/network)
Example: PageRank, identifying hubs and authorities
3. Web Usage Mining
Analyzing user behavior from logs
Example: Clickstream analysis, personalization, recommendation systems

Role in Data Mining

Data mining in multimedia databases involves discovering interesting patterns, correlations, and relationships from media data. Here's how:

1. Content-Based Retrieval

Using **features** like color, texture, shape (for images), or pitch and tempo (for audio), content-based mining helps find similar content.

Example: "Find all images similar to this one."

2. Pattern Recognition

Identifying recurring patterns in multimedia content, such as objects in videos or spoken words in au dio files.

Applications include facial recognition or gesture detection in videos.

3. Classification & Clustering

Grouping multimedia objects into categories.

E.g., Clustering audio files into genres.

Classifying medical images into “normal” and “abnormal”.

4. Association Rules

Mining rules like: “If a video contains beach scenes, it likely also contains people in swimwear.”

5. Temporal & Spatial Mining

Videos and audio have a time dimension; spatial data like satellite images have location info.

Mining such data involves tracking changes over time or identifying geospatial patterns.

Web Mining

Introduction

- With the huge amount of information available online, the World Wide Web is a fertile area for data mining research.
- WWW is a popular and interactive medium to circulate information today.
- The Web is huge, diverse, and dynamic.

The WWW is huge, widely distributed, global information service centre and, therefore, constitutes a rich source for data mining

- Intelligent Web Search
- Personalization, Recommendation Engines
- Web-commerce applications
- Building the Semantic Web
- Web page classification and categorization
- News classification and clustering
- Information trend monitoring
- Analysis of online communities
- Web and mail spam filtering

Abundance and authority crisis

- Liberal and informal culture of content generation and dissemination
- Redundancy and non-standard form and content
- Millions of qualifying pages for most broad queries
- No authoritative information about the reliability of a site
- Little support for adapting to the background of specific users
- Pages added continuously and average page changes in a few weeks

Web mining is the use of data mining techniques to automatically discover and extract information from Web documents/services.

Web mining is to apply data mining techniques to extract and uncover knowledge from *web documents* and *services*.

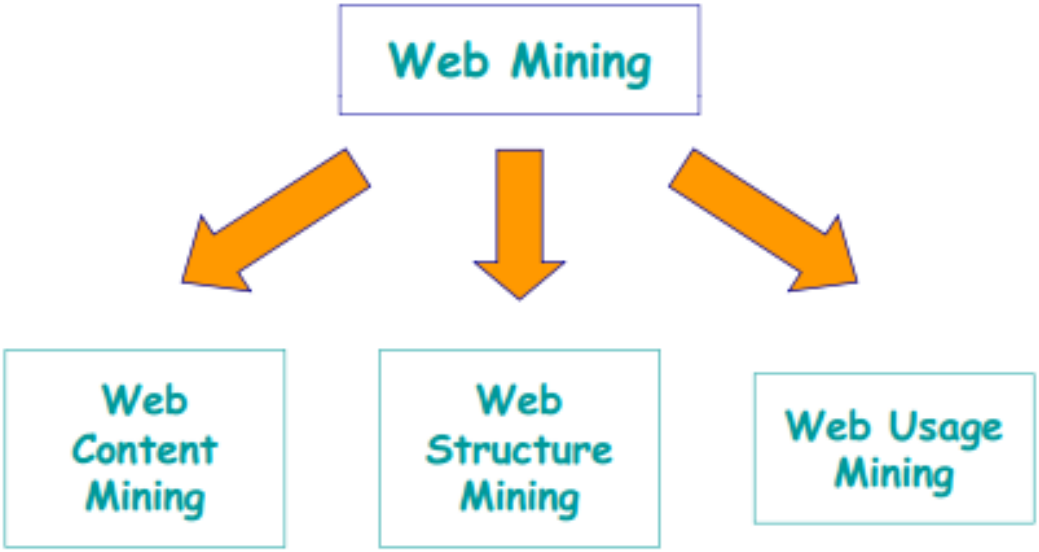
“Web mining refers to the overall process of discovering potentially useful and previously unknown information or knowledge from the Web data.”

- ✓Can be viewed as four subtasks

Web Mining: Subtasks

- Resource finding
  - ✓Retrieving intended web documents
- Information selection and pre-processing
  - ✓Select and pre-process specific information from selected documents
  - ✓Kind of transformation processes of the original data retrieved in the IR process
  - ✓This transformation could be a kind of pre-processing
- Generalization
  - ✓Discover general patterns within and across web sites
- Analysis

Web Mining Taxonomy



Web Mining Taxonomy

Web content mining: Web content mining is the process of extracting useful information from the contents of web documents.

Content data is the collection of facts a web page is designed to contain.

It may consist of text, images, audio, video, or structured records such as lists and tables.

Web Content Data Structure

- Web content consists of several types of data
  - Text, image, audio, video, hyperlinks.
- Unstructured – free text
- Semi-structured – HTML
- More structured – Data in the tables or database generated HTML pages

Note: much of the Web content data is unstructured text data.

Web Content Mining: IR View

- Unstructured Documents
  - ✓ Bag of words to represent unstructured documents
    - ✓ Takes single word as feature
    - ✓ Ignores the sequence in which words occur
  - ✓ Features could be
    - ✓ Boolean
      - ✓ Word either occurs or does not occur in a document
    - ✓ Frequency based
      - ✓ Frequency of the word in a document
  - ✓ Variations of the feature selection include
      - ✓ Removing the case, punctuation, infrequent words and stop words
  - ✓ Features can be reduced using different feature selection techniques:
    - ✓ Information gain, mutual information, cross entropy.
    - ✓ Stemming: which reduces words to their morphological roots.

Web Content Mining: IR View

- Semi-Structured Documents
  - ✓Uses richer representations for features
    - ✓Due to the additional structural information in the hypertext document (typically HTML and hyperlinks)
  - ✓Uses common data mining methods (whereas unstructured might use more text mining methods)
  - ✓Application:
    - ✓ Hypertext classification or categorization and clustering,

Web Content Mining: DB View

- The database techniques on the Web are related to the problems of managing and querying the information on the Web.
- DB view tries to infer the structure of a Web site or transform a Web site to become a database
  - ✓ Better information management
  - ✓ Better querying on the Web
- Can be achieved by:
  - ✓ Finding the schema of Web documents
  - ✓ Building a Web warehouse
  - ✓ Building a Web knowledge base
  - ✓ Building a virtual database

Web Content Mining: DB View

- DB view mainly uses the Object Exchange Model (OEM)
  - ✓ Represents semi-structured data by a labeled graph
  - ✓ The data in the OEM is viewed as a graph, with objects as the vertices and labels on the edges
    - ✓ Each object is identified by an object identifier [oid] and
    - ✓ Value is either atomic or complex
- Process typically starts with manual selection of Web sites for doing Web content mining
- Main application:
  - The task of finding frequent substructures in semi-structured data
  - The task of creating multi-layered database

Web structure mining: aims at developing techniques to take advantage of the collective judgement of web page quality which is available in the form of hyperlinks.

The structure of a typical web graph consists of web pages as nodes, and hyperlinks as edges connecting related pages. Web structure mining is the process of discovering structure information from the web.

This can be further divided into two kinds based on the kind of structure information used.

Hyperlinks

A hyperlink is a structural unit that connects a location in a web page to a different location, either within the same web page or on a different web page.

A hyperlink that connects to a different part of the same page is called an intra-document hyperlink, and a hyperlink that connects two different pages is called an inter-document hyperlink.

Document Structure

In addition, the content within a Web page can also be organized in a tree structured format, based on the various HTML and XML tags within the page.

Web Structure Mining

- Interested in the structure of the hyperlinks within the Web
- Inspired by the study of social networks and citation analysis
  - Can discover specific types of pages(such as hubs, authorities, etc.) based on the incoming and outgoing links.
- Application:
  - Discovering micro-communities in the Web ,
  - measuring the “completeness” of a Web site

Web usage mining: focuses on techniques to study the user behaviour when navigating the web. (also known as Web log mining and clickstream analysis).

It is the application of data mining techniques to discover interesting usage patterns from web usage data, in order to understand and better serve the needs of web-based applications.

Usage data captures the identity or origin of web users along with their browsing behavior at a web site.

Website Usage Analysis

Why analyze Website usage?

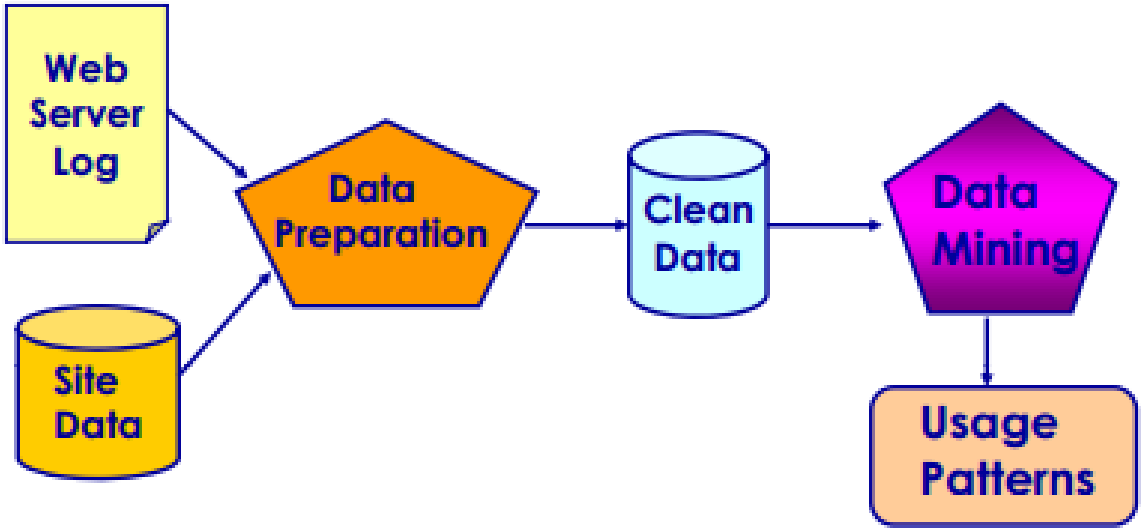
- Knowledge about how visitors use Website could
  - Provide guidelines to web site reorganization; Help prevent disorientation
  - Help designers place important information where the visitors look for it
  - Pre-fetching and caching web pages
  - Provide adaptive Website (Personalization)
  - Questions which could be answered
    - What are the differences in usage and access patterns among users?
    - What user behaviours change over time?
    - How usage patterns change with quality of service (slow/fast)?
    - What is the distribution of network traffic over time?

Web Server Data
User logs are collected by the web server and typically include IP address, page reference and access time.

Application Server Data
Commercial application servers such as Weblogic, StoryServer have significant features to enable E-commerce applications to be built on top of them with little effort. A key feature is the ability to track various kinds of business events and log them in application server logs.

Application Level Data
New kinds of events can be defined in an application, and logging can be turned on for them — generating histories of these events.

Web Usage Mining Process



Questions for practice

Consider the transactional database, find the frequent itemsets and generate association rules (min support =3 , min confidence =75%

1	Milk, Tea , jam
2	Eggs, Tea, Cold drink
3	Milk, Eggs, Tea, Cold Drink
4	Eggs, Cold Drink
5	Tea, Bread, Milk
6	Eggs, Milk, oats
7	Tea, Milk, Egg, Biscuit
8	Biscuit, cold drink, chips
9	Chips, cold drink, oats
10	Eggs, milk, oats

Consider the database, find the class of student with id 8 using k-nn algorithm
K=3

ID	M1	M2	Class
1	89	67	A
2	34	23	B
3	86	79	A
4	50	55	A
5	44	48	B
6	90	56	A
7	2	5	B
8	45	56	?

Consider the transactional database, find the frequent itemsets and generate association rules (min support =2 , min confidence =60%

ID	ITEMS
1	Bread, butter, Milk, rice
2	Flaxseeds, Eggs, Tea, rice, flaxseeds
3	Eggs, cheese, flaxseeds
4	Tea, Bread, Milk, flaxseeds
5	Eggs, Milk, oats, rice
6	Tea, Milk, Egg, Biscuit
7	Biscuit, cheese, chips
8	Chips, cheese, oats
9	flaxseeds, milk, oats, rice
10	Milk , cheese, Biscuit, flaxseeds

Consider the database, find the class of student with id 8 using k-nn algorithm
K=3

ID	M1	M2	Class
1	89	67	A
2	34	23	B
3	86	79	A
4	50	55	A
5	44	48	B
6	90	56	A
7	2	5	B
8	45	56	?

Consider the database, find the class of student with id 8 using k-nn algorithm
K=3

ID	M1	M2	Class
1	99	98	T
2	34	23	S
3	89	77	T
4	76	87	T
5	34	36	S
6	90	56	T
7	23	15	S
8	34	87	?