

UNIT - I

**Data Warehouse and OLAP Technology for Data
mining**

**Prepared by
Dr. Lakshmi Priya GG**

Contents

- Introduction to Data Warehouse
- A multidimensional Data Model
- Data Warehouse architecture
- Data pre-processing
- Data cleaning
- Data integration and Transformation.

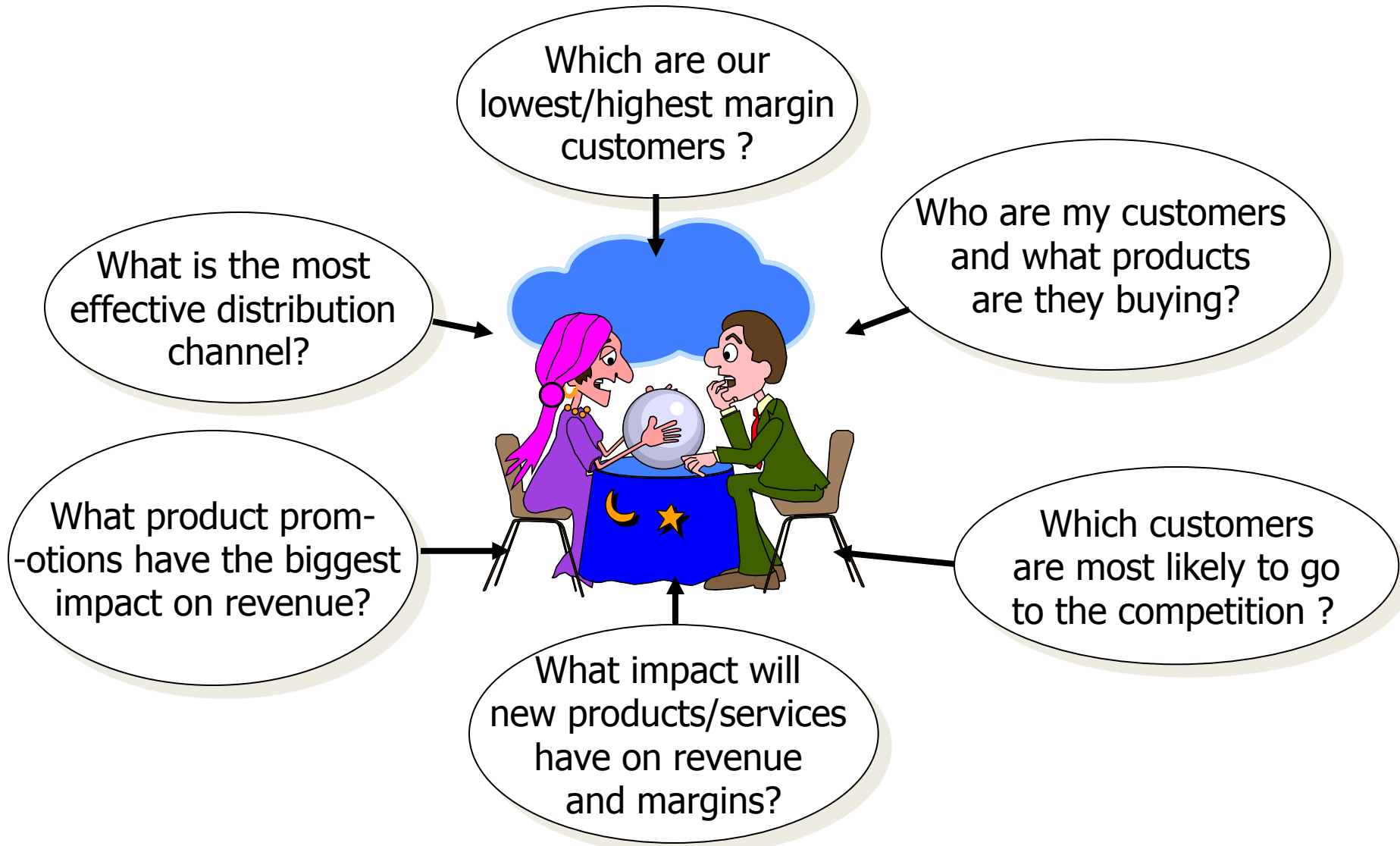
Disucssions

- Overview of Data Warehouse and Business Intelligence –
 - Basic concepts of Data warehouse
 - Data Warehouse Architecture
 - design – implementation issues
- Data warehouse and Business Intelligence modeling
 - data cube
 - OLAP and basic reporting
 - querying-emerging Business Intelligence technology

Database and Data Ware Housing....

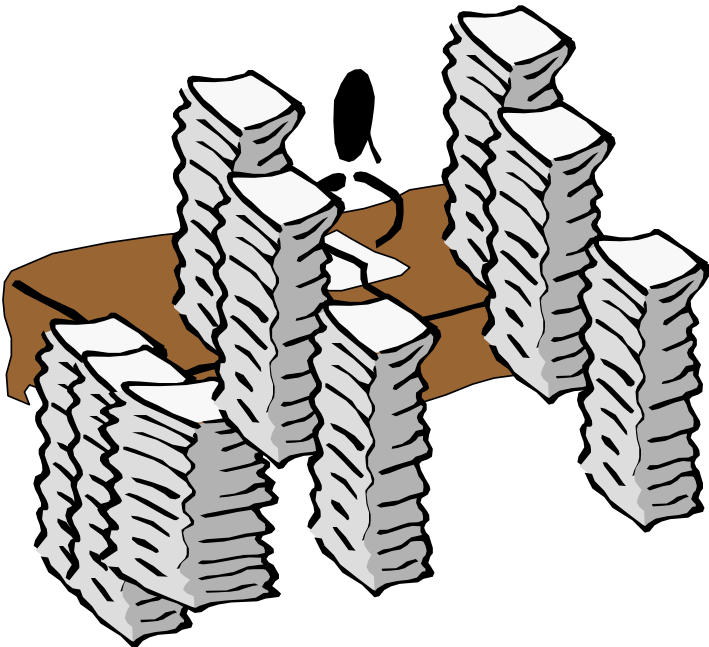
- The Difference...
 - DWH Constitute Entire Information Base For All Time..
 - Database Constitute Real Time Information...
 - DWH Supports DM And Business Intelligence.
 - Database Is Used To Running The Business
 - DWH Is How To Run The Business

A producer wants to know....



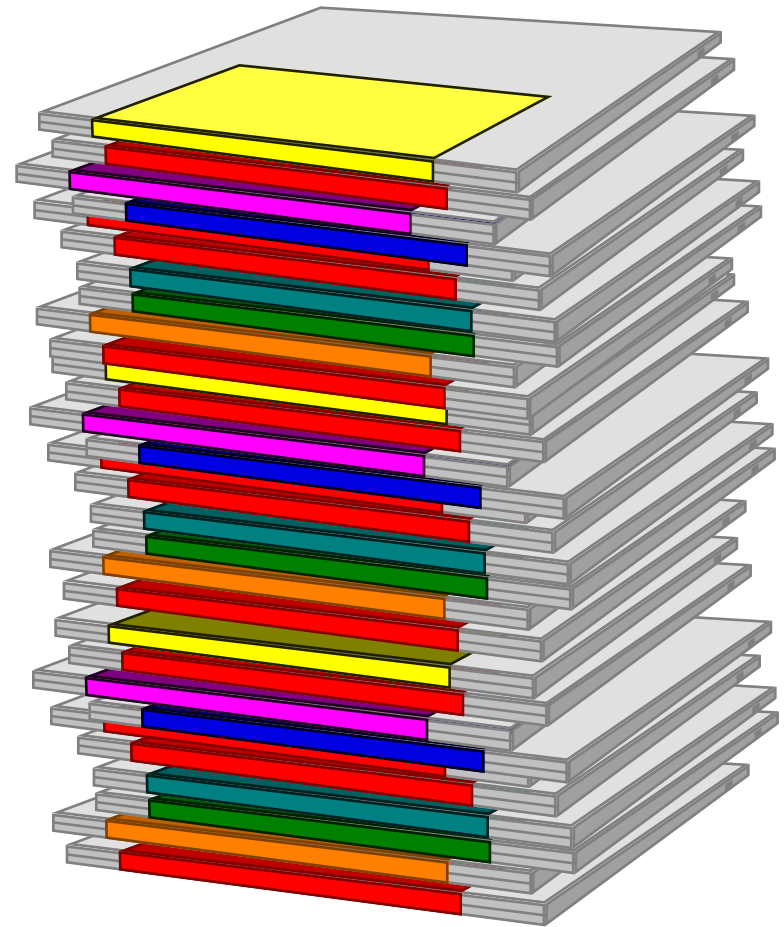
Data, Data everywhere yet ...

- **I can't find the data I need**
 - data is scattered over the network
 - many versions, subtle differences
- **I can't get the data I need**
 - need an expert to get the data
- **I can't understand the data I found**
 - available data poorly documented
- **I can't use the data I found**
 - results are unexpected
 - data needs to be transformed from one form to other



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 way that they can understand and use in a business context.



What is Data Warehousing?

A process of
**transforming data into
information** and
making it available to
users in a timely
enough manner to
make a difference



What is a Data Warehouse

- Many definitions
 - A repository of an organizations electronically stored data. (*Wikipedia*)
 - A central repository for all or significant parts of the data that an enterprise's various business systems collect. (*Search SQL*)
 - A warehouse is a subject-oriented, integrated, time-variant and non-volatile collection of data in support of management's decision making process (*Bill Inmon*)
 - A data warehouse is a database system optimized for reporting. (*Deakin University*)

DW Definition

- More than just a database or single tool
 - Tools (Extract, Load, Transform, Aggregate, Mine, Analyze, Report, Manage)
 - Complex subject specific data models
 - Processes / Governance / Standards
 - People
 - Technology
 - Metadata

Data Warehousing -- a process

- It is a relational or multidimensional database management system
 - designed to support **management decision making**.
- It is a copy of transaction data
 - specifically structured for **querying and reporting**.
- Technique for assembling and managing data from various sources
 - for the purpose of **answering business questions**. Thus making decisions that were not previous possible

- “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 is ...

- **Subject Oriented:** Data that gives information about a particular subject instead of about a company's ongoing operations.
- **Integrated:** Data that is gathered into the data warehouse from a variety of sources and merged into a coherent whole.
- **Time-variant:** All data in the data warehouse is identified with a particular time period.
- **Non-volatile:** Data is stable in a data warehouse. More data is added but data is never removed. This enables management to gain a consistent picture of the business.

Contd....

- Data warehousing is combining data from multiple and usually varied sources into one comprehensive and easily manipulated database.
- Common accessing systems of data warehousing include queries, analysis and reporting.
- Because data warehousing creates one database in the end, the number of sources can be anything you want it to be, provided that the system can handle the volume, of course.
- The final result, however, is homogeneous data, which can be more easily manipulated.

History of data warehousing

- The concept of data warehousing dates back to the late 1980s when IBM researchers Barry Devlin and Paul Murphy developed the "business data warehouse".
- 1960s - General Mills and Dartmouth College, in a joint research project, develop the terms *dimensions* and *facts*.
- 1970s - ACNielsen and IRI provide dimensional data marts for retail sales.
- 1983 – Tera data introduces a database management system specifically designed for decision support.
- 1988 - Barry Devlin and Paul Murphy publish the article *An architecture for a business and information systems* in *IBM Systems Journal* where they introduce the term "business data warehouse".

Data Warehouse vs. Operational DBMS

- OLTP (on-line transaction processing)
 - Major task of traditional relational DBMS
 - Day-to-day operations: purchasing, inventory, banking, manufacturing, payroll, registration, accounting, etc.
- OLAP (on-line analytical processing)
 - Major task of data warehouse system
 - Data analysis and decision making
- Distinct features (OLTP vs. OLAP):
 - User and system orientation: customer vs. market
 - Data contents: current, detailed vs. historical, consolidated
 - Database design: ER + application vs. star + subject
 - View: current, local vs. evolutionary, integrated
 - Access patterns: update vs. read-only but complex queries

<i>Feature</i>	<i>OLTP</i>	<i>OLAP</i>
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

Why Separate Data Warehouse?

- High performance for both systems
 - DBMS— tuned for OLTP: access methods, indexing, concurrency control, recovery
 - Warehouse—tuned for OLAP: complex OLAP queries, multidimensional view, consolidation
- Different functions and different data:
 - [missing data](#): Decision support requires historical data which operational DBs do not typically maintain
 - [data consolidation](#): DS requires consolidation (aggregation, summarization) of data from heterogeneous sources
 - [data quality](#): different sources typically use inconsistent data representations, codes and formats which have to be reconciled
- Note: There are more and more systems which perform OLAP analysis directly on relational databases

To summarize ...

- OLTP Systems are used to *“run”* a business



- The Data Warehouse helps to *“optimize”* the business

Evolution in organizational use of data warehouses

Organizations generally start off with relatively simple use of data warehousing. Over time, more sophisticated use of data warehousing evolves. The following general stages of use of the data warehouse can be distinguished:

- **Off line Operational Database**

- Data warehouses in this initial stage are developed by simply copying the data off an operational system to another server where the processing load of reporting against the copied data does not impact the operational system's performance.

- **Off line Data Warehouse**

- Data warehouses at this stage are updated from data in the operational systems on a regular basis and the data warehouse data is stored in a data structure designed to facilitate reporting.

- **Real Time Data Warehouse**

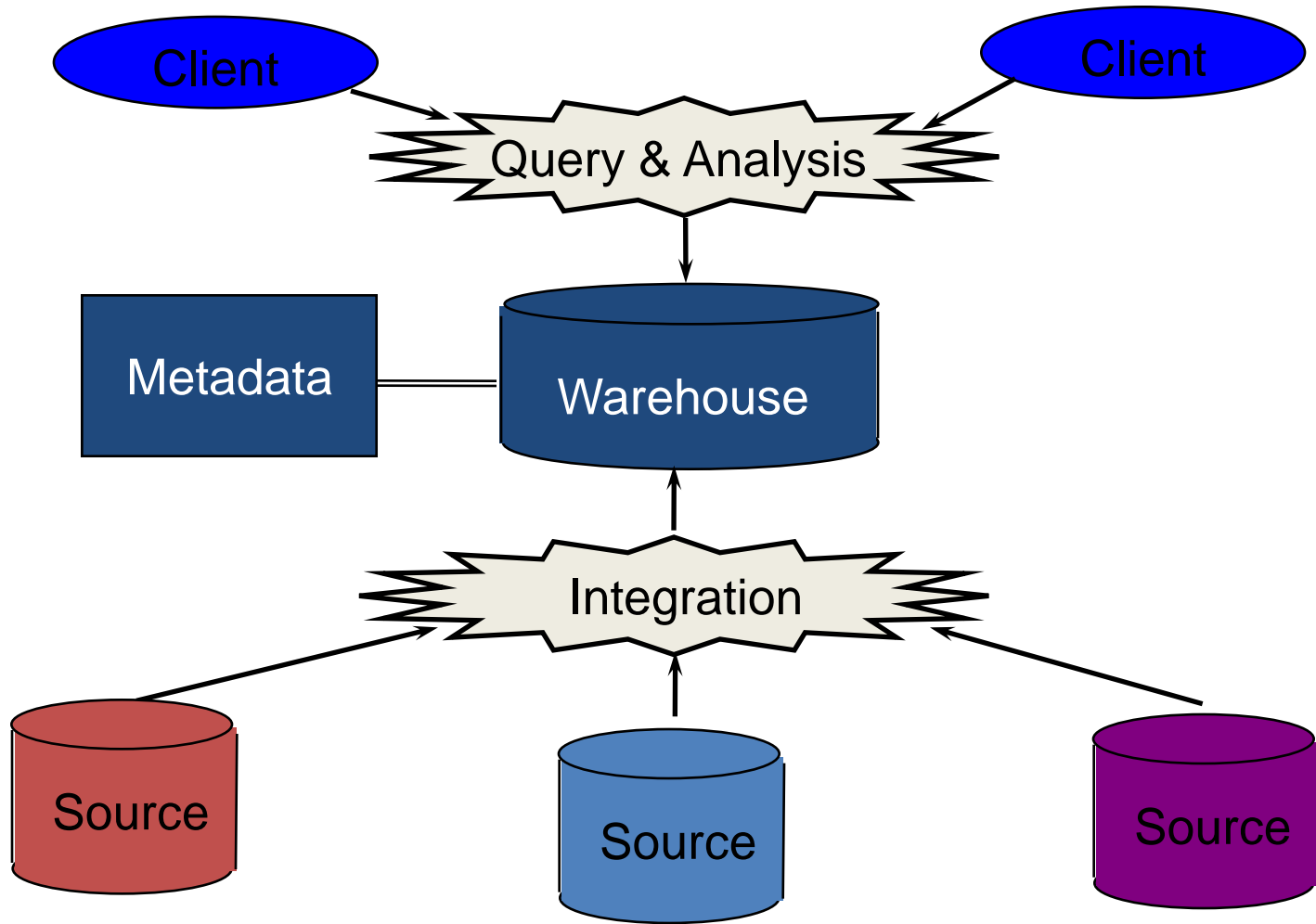
- Data warehouses at this stage are updated every time an operational system performs a transaction (e.g. an order or a delivery or a booking.)

- **Integrated Data Warehouse**

- Data warehouses at this stage are updated every time an operational system performs a transaction. The data warehouses then generate transactions that are passed back into the operational systems.

Data Warehouse Architecture

Data Warehouse Architecture

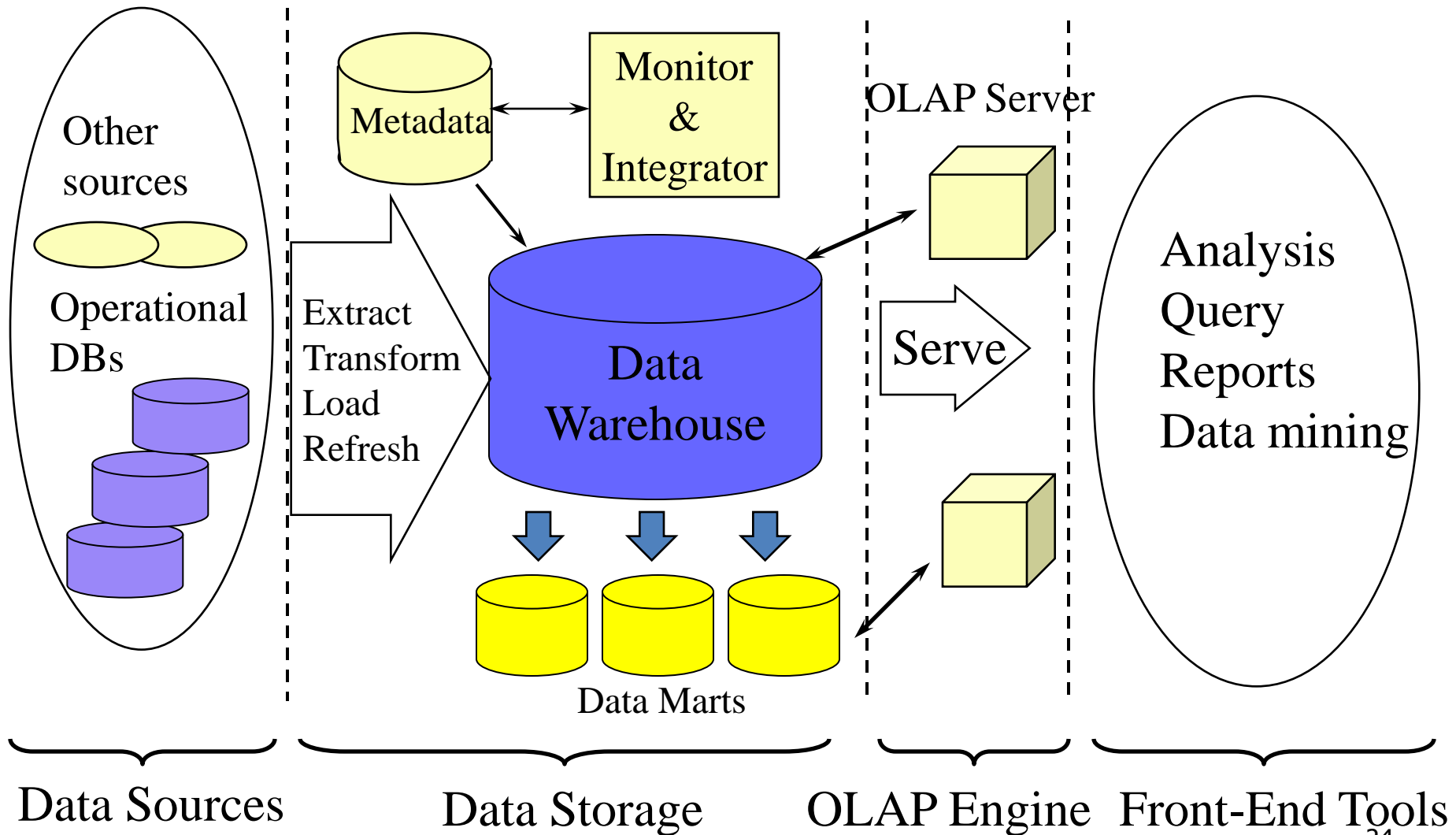


- The data has been selected from various sources and then integrate and store the data in a single and particular format.
- Data warehouses contain current detailed data, historical detailed data, lightly and highly summarized data, and metadata.
- Current and historical data are voluminous because they are stored at the highest level of detail.
- Lightly and highly summarized data are necessary to save processing time when users request them and are readily accessible.
- **Metadata** are “data about data”. It is important for designing, constructing, retrieving, and controlling the warehouse data.

Technical metadata include where the data come from, how the data were changed, how the data are organized, how the data are stored, who owns the data, who is responsible for the data and how to contact them, who can access the data , and the date of last update.

Business metadata include what data are available, where the data are, what the data mean, how to access the data, predefined reports and queries, and how current the data are.

Data Warehouse: A Multi-Tiered Architecture



The data warehouse architecture



The back room

The front room

External sources

Operational
source systems

Extract
Transform
Load

Data warehouse

Data marts

Serve

Analysis/OLAP

Product1	Time1	Value1	Value11
Product2	Time2	Value2	Value21
Product3	Time3	Value3	Value31
Product4	Time4	Value4	Value41

Query/Reporting



Data mining

Faló	söldf
flaöd	aklöd
faló	alksdf

**Operational source
systems (RK)**

Legacy systems
OLTP/TP systems

**Data staging
area (RK)**

Back end tools

**Data presentation
area (RK)**

"The data warehouse"
Presentation (OLAP) servers

Data access tools (RK)

End user applications
Business Intelligence tools

Building A Data Warehouse

- The builders of Data warehouse should take a broad view of the anticipated use of the warehouse.
 - The design should support ad-hoc querying
 - An appropriate schema should be chosen that reflects the anticipated usage.

Building A Data Warehouse

- The Design of a Data Warehouse involves following steps.
 - Acquisition of data for the warehouse.
 - Ensuring that Data Storage meets the query requirements efficiently.
 - Giving full consideration to the environment in which the data warehouse resides.

Functionality of a Data Warehouse

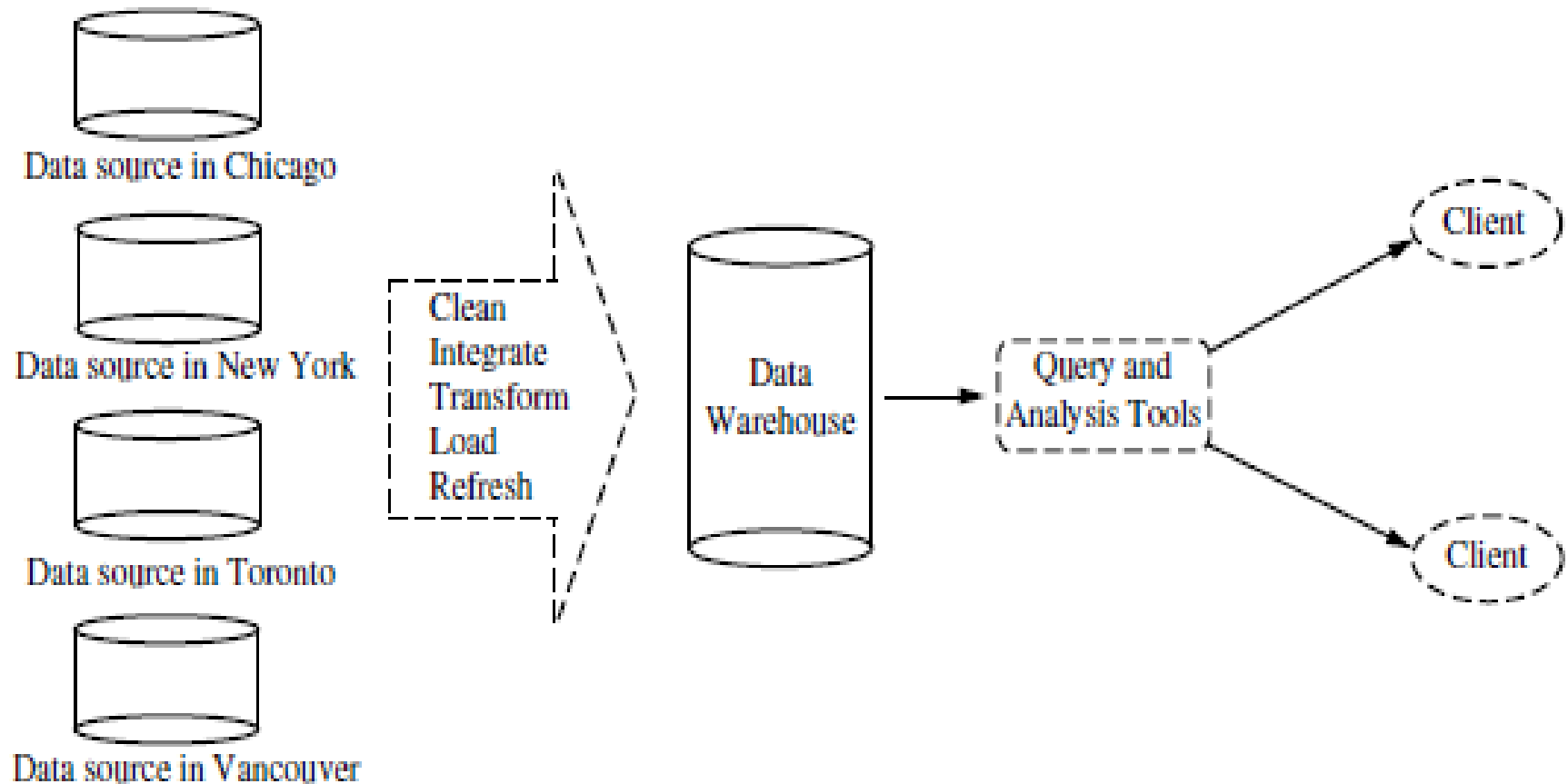
- Functionality that can be expected:
 - **Roll-up**: Data is summarized with increasing generalization
 - **Drill-Down**: Increasing levels of detail are revealed
 - **Pivot**: Cross tabulation is performed
 - **Slice and dice**: Performing projection operations on the dimensions.
 - **Sorting**: Data is sorted by ordinal value.
 - **Selection**: Data is available by value or range.
 - **Derived attributes**: Attributes are computed by operations on stored derived values.

Multidimensional Data Model

From Tables and Spreadsheets to Data Cubes

- A data warehouse is based on a **multidimensional data model** which views data in the form of a data cube
- A data cube, such as **sales**, allows data to be modeled and viewed in multiple dimensions
 - Dimension tables, such as **item (item_name, brand, type)**, or **time(day, week, month, quarter, year)**
 - Fact table contains measures (such as **dollars_sold**) and keys to each of the related dimension tables
- In data warehousing literature, an n-D base cube is called a **base cuboid**. The top most 0-D cuboid, which holds the highest-level of summarization, is called the **apex cuboid**. The lattice of cuboids forms a **data cube**.

Example – Allelectronics Corporation



summarized sales data of *AllElectronics*

- three dimensions:
 - *address* (with city values *Chicago, New York, Toronto, Vancouver*),
 - *time* (with quarter values *Q1, Q2, Q3, Q4*),
 - *item*(with item type values *home entertainment, computer, phone, security*).
- The aggregate value stored in each cell is *sales amount* (in thousands).

A 2-D view of sales data for *AllElectronics* according to the dimensions *time* and *item*, where the sales are from branches located in the city of Vancouver. The measure displayed is *dollars_sold* (in thousands).

<i>location</i> = "Vancouver"				
<i>time</i> (quarter)	<i>item</i> (type)			
	<i>home</i>			
	<i>entertainment</i>	<i>computer</i>	<i>phone</i>	<i>security</i>
Q1	605	825	14	400
Q2	680	952	31	512
Q3	812	1023	30	501
Q4	927	1038	38	580

Table 3.3 A 3-D view of sales data for *AllElectronics*, according to the dimensions *time*, *item*, and *location*. The measure displayed is *dollars_sold* (in thousands).

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

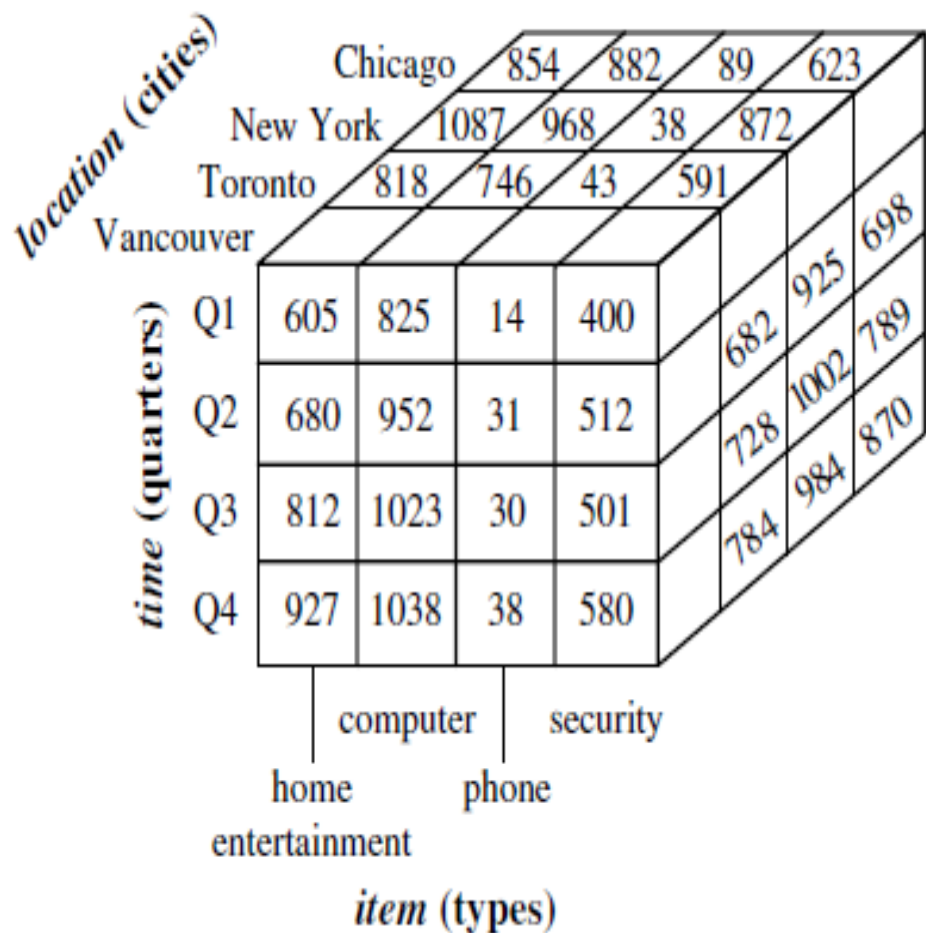
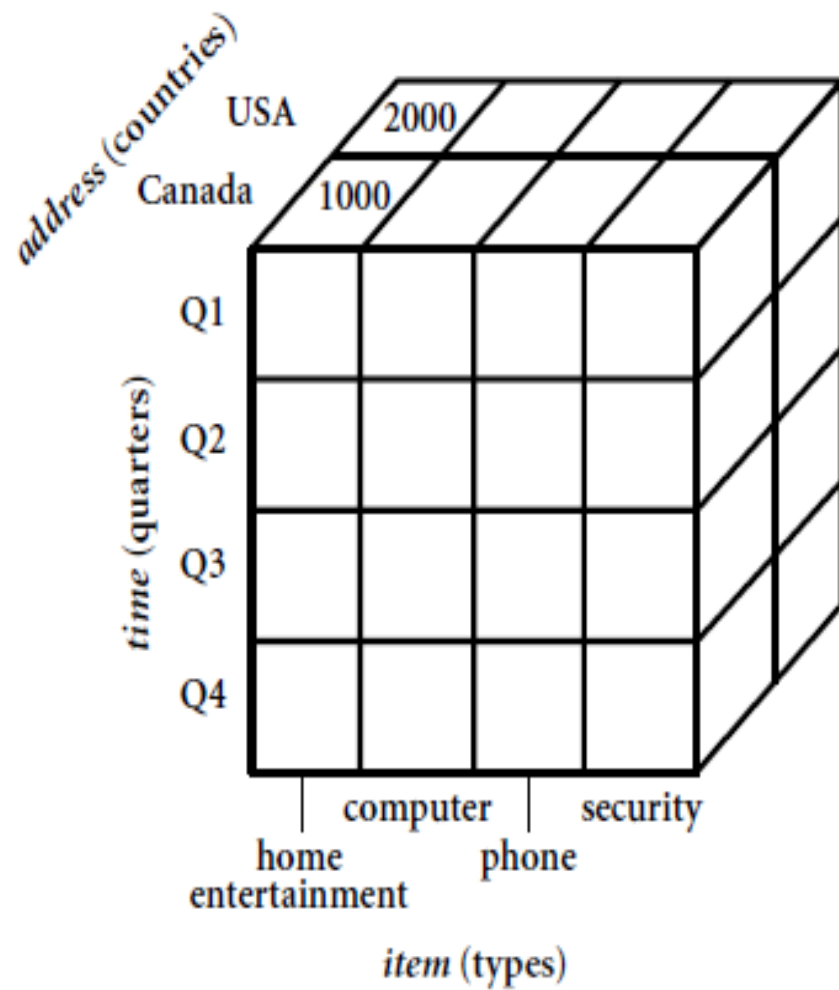
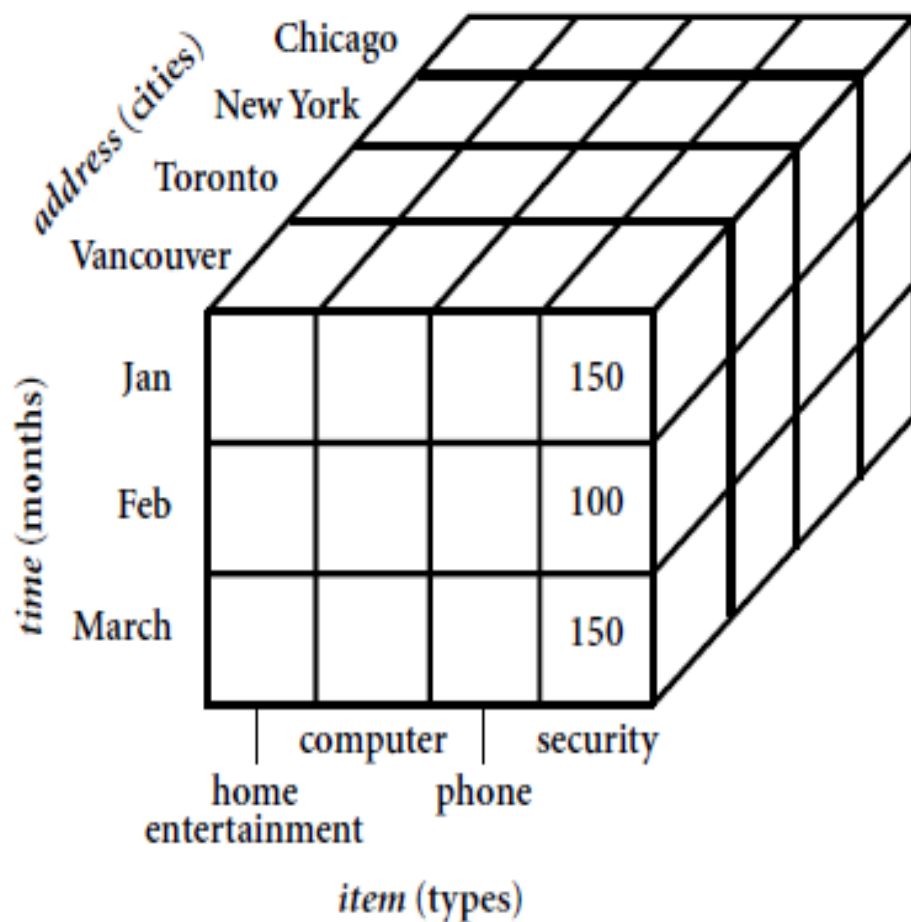
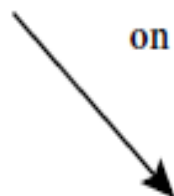
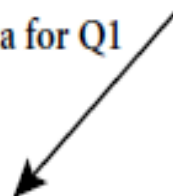


Figure 3.1 A 3-D data cube representation of the data in Table 3.3, according to the dimensions *time*, *item*, and *location*. The measure displayed is *dollars_sold* (in thousands).

(b)

Drill-down
on time data for Q1

Roll-up
on address



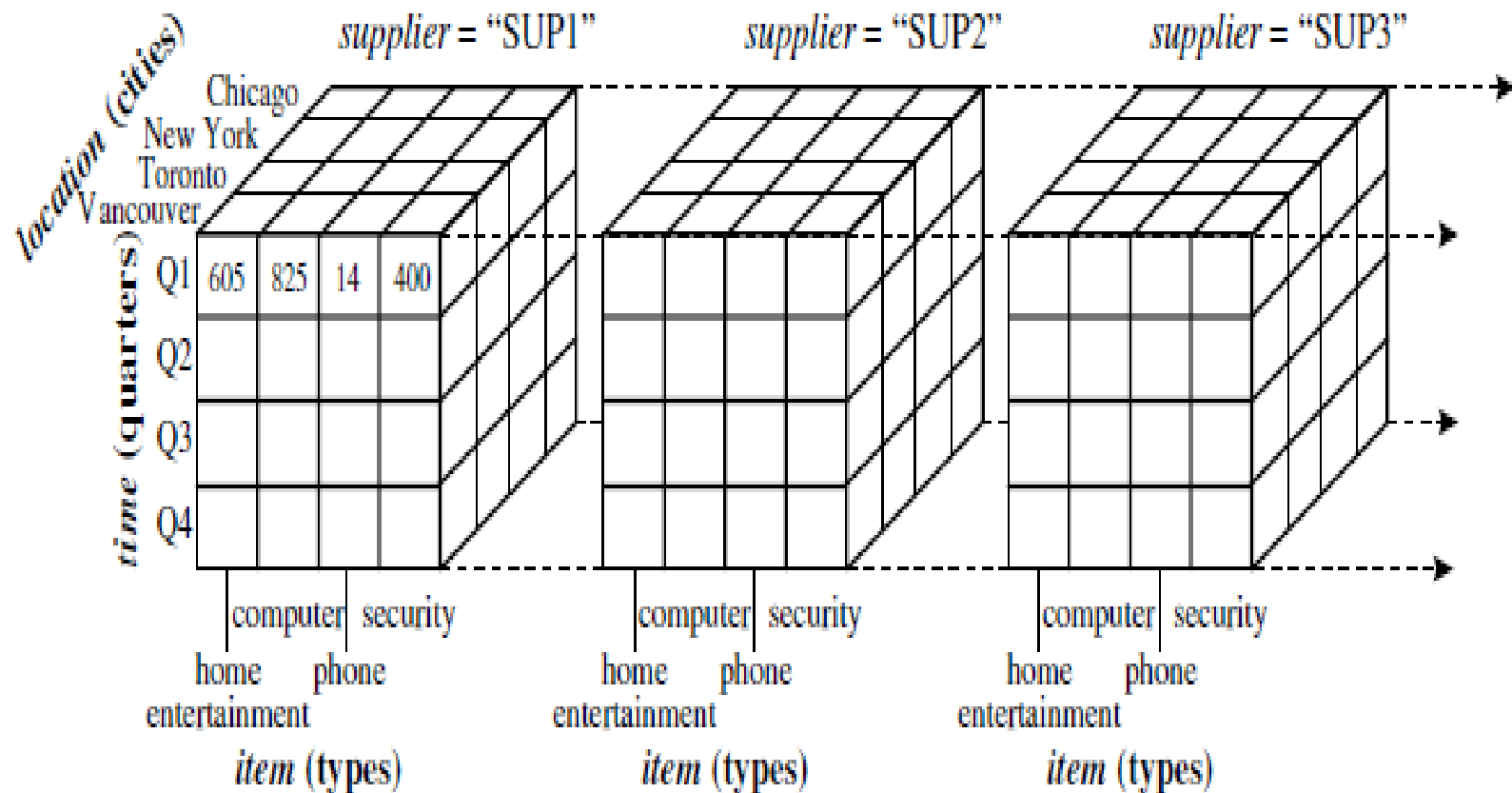
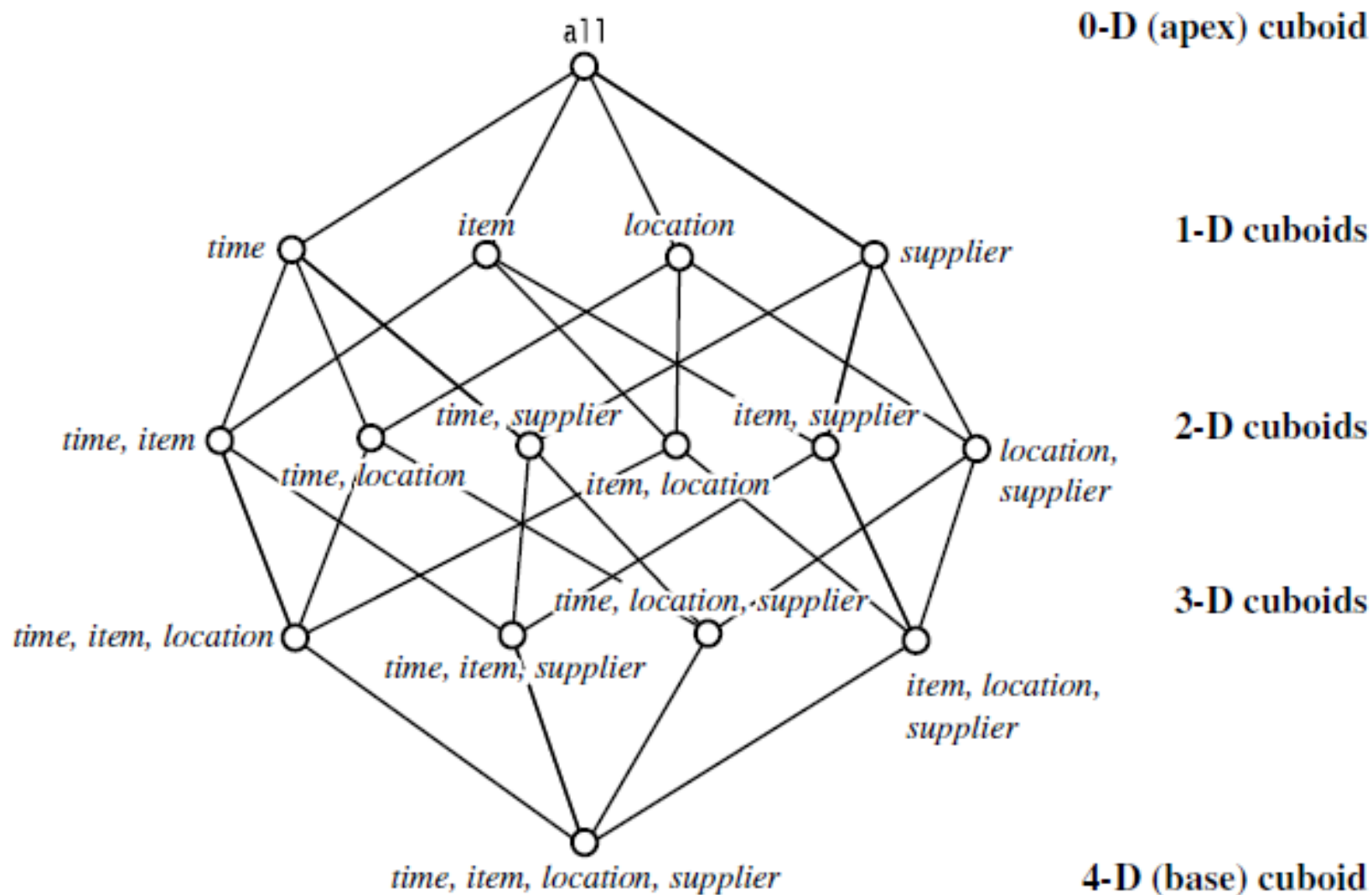


Figure 3.2 A 4-D data cube representation of sales data, according to the dimensions *time*, *item*, *location*, and *supplier*. The measure displayed is *dollars_sold* (in thousands). For improved readability, only some of the cube values are shown.



-
- ↳ Lattice of cuboids, making up a 4-D data cube for the dimensions *time*, *item*, *location*, and *supplier*. Each cuboid represents a different degree of summarization.

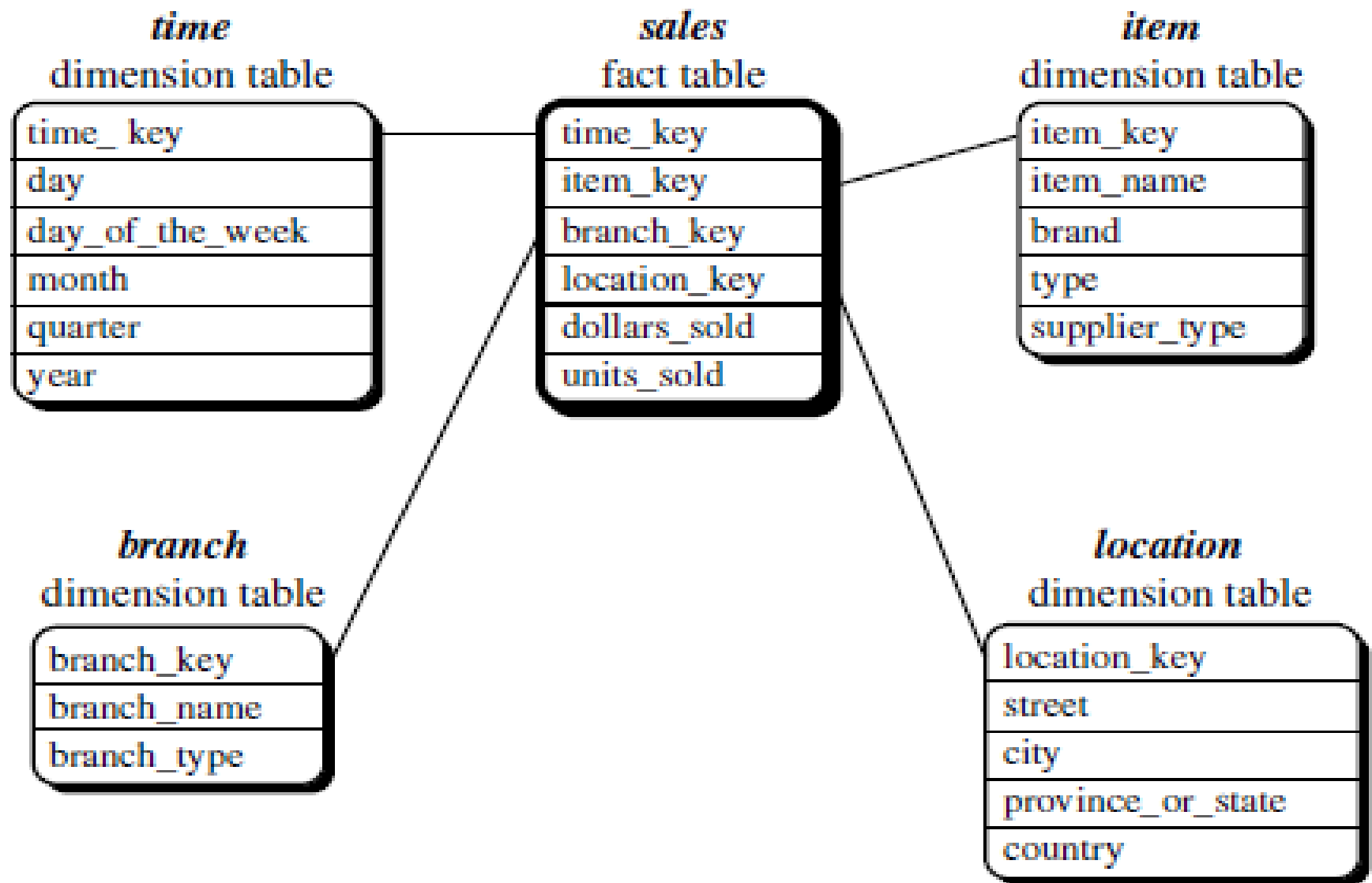
Class Room Practice

- Suppose that a data warehouse consists of the three dimensions *time*, *doctor*, and *patient*, and the two measures *count* and *charge*, where *charge* is the fee that a doctor charges a patient for a visit.

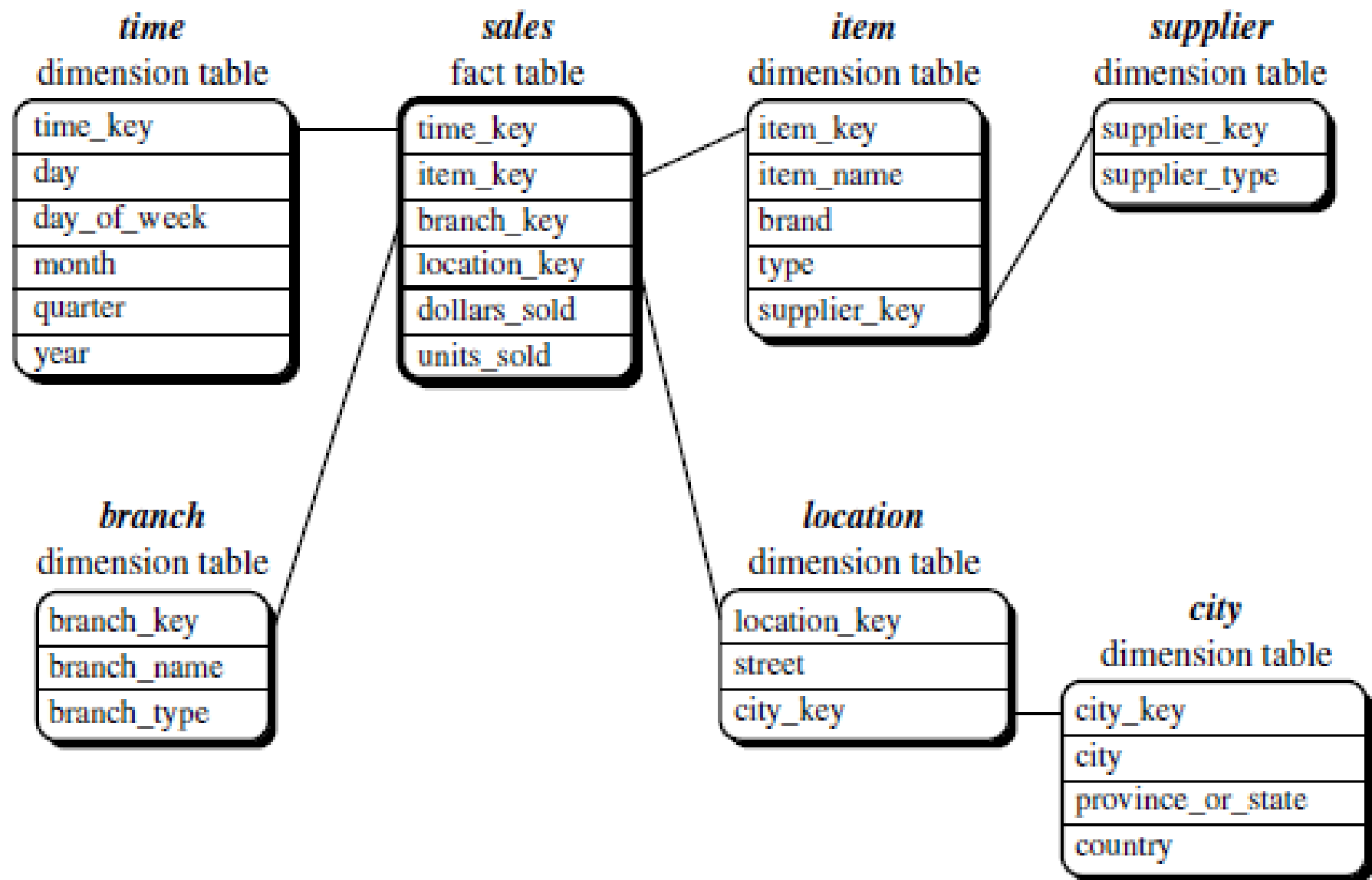
Stars, Snowflakes, and Fact Constellations: Schemas for Multidimensional Databases

Conceptual Modeling of Data Warehouses

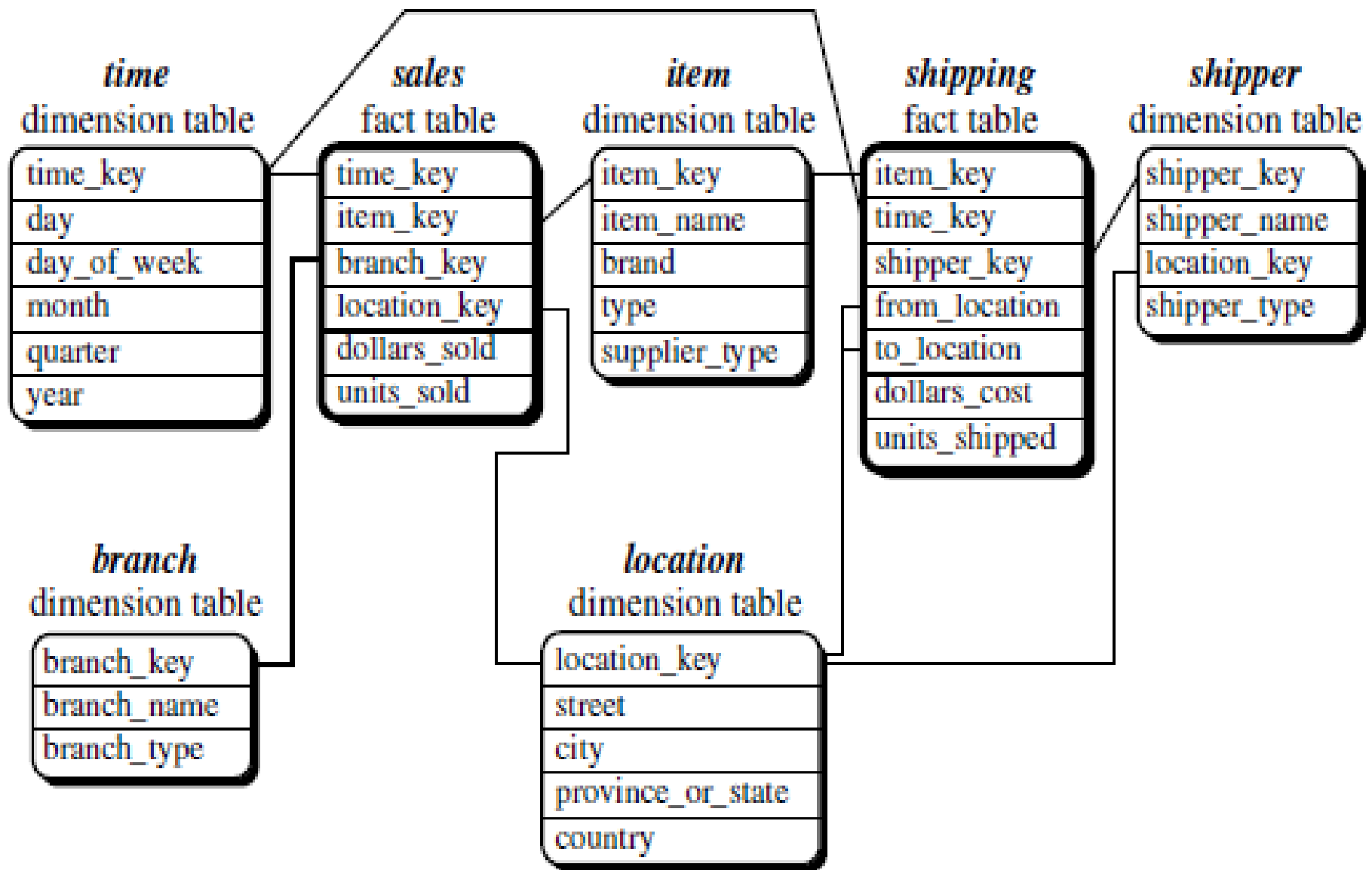
- Modeling data warehouses: dimensions & measures
 - Star schema: A fact table in the middle connected to a set of dimension tables
 - Snowflake schema: A refinement of star schema where some dimensional hierarchy is **normalized** into a set of smaller dimension tables, forming a shape similar to snowflake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



Star schema of a data warehouse for sales.



Snowflake schema of a data warehouse for sales.



Fact constellation schema of a data warehouse for sales and shipping.

Data Preprocessing

Why Data Preprocessing?

- Data in the real world is dirty
 - **incomplete**: lacking attribute values, lacking certain attributes of interest, or containing only aggregate data
 - e.g., occupation=" "
 - **noisy**: containing errors or outliers
 - e.g., Salary="-10"
 - **inconsistent**: containing discrepancies in codes or names
 - e.g., Age="42" Birthday="03/07/1997"
 - e.g., Was rating "1,2,3", now rating "A, B, C"
 - e.g., discrepancy between duplicate records

Why Is Data Dirty?

- Incomplete data may come from
 - “Not applicable” data value when collected
 - Different considerations between the time when the data was collected and when it is analyzed.
 - Human/hardware/software problems
- Noisy data (incorrect values) may come from
 - Faulty data collection instruments
 - Human or computer error at data entry
 - Errors in data transmission
- Inconsistent data may come from
 - Different data sources
 - Functional dependency violation (e.g., modify some linked data)
- Duplicate records also need data cleaning

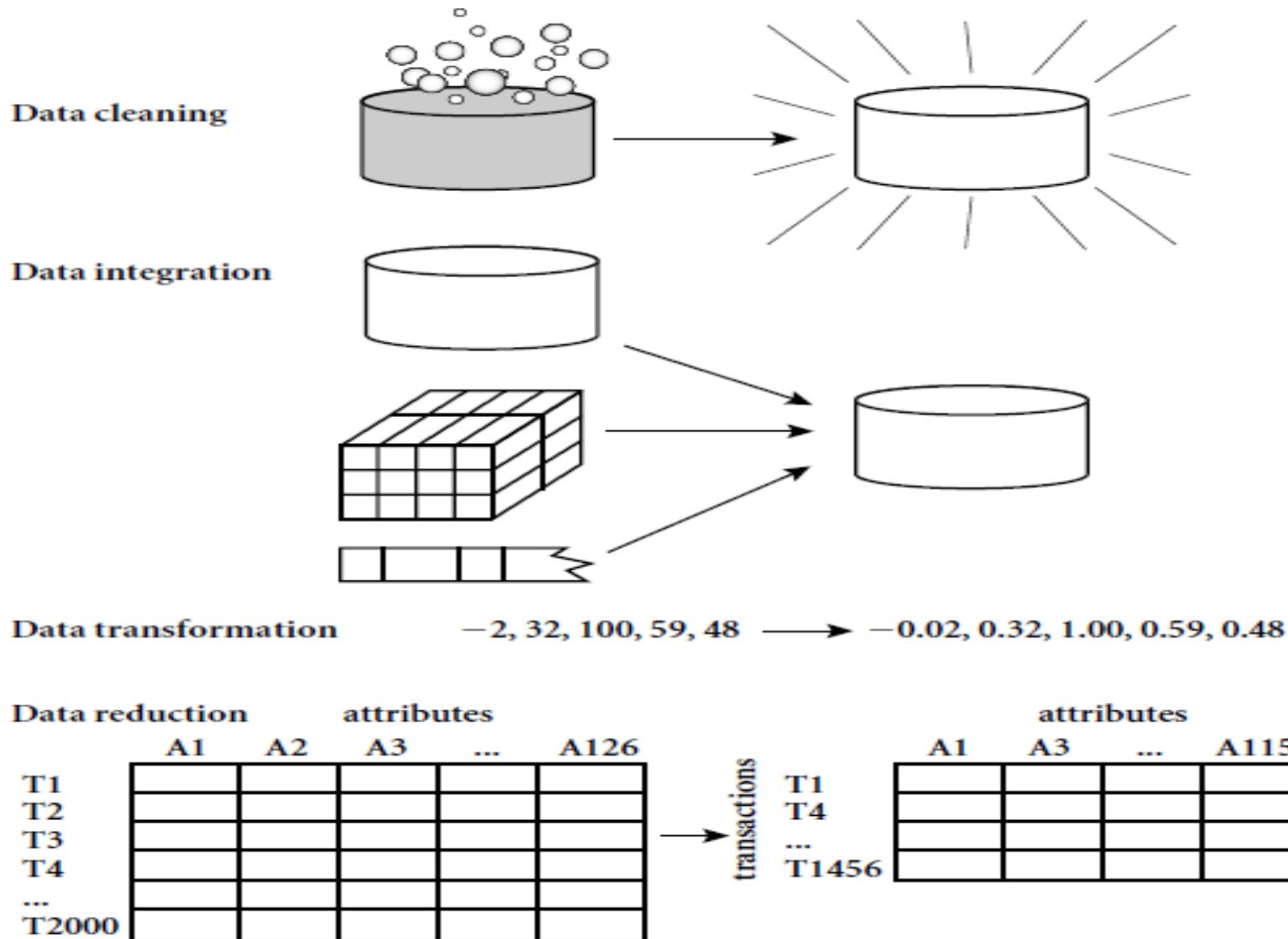
Why Is Data Preprocessing Important?

- No quality data, no quality mining results!
 - Quality decisions must be based on quality data
 - e.g., duplicate or missing data may cause incorrect or even misleading statistics.
 - Data warehouse needs consistent integration of quality data
- Data extraction, cleaning, and transformation comprises the majority of the work of building a data warehouse

Major Tasks in Data Preprocessing

- Data cleaning
 - Fill in missing values, smooth noisy data, identify or remove outliers, and resolve inconsistencies
- Data integration
 - Integration of multiple databases, data cubes, or files
- Data transformation
 - Normalization and aggregation
- Data reduction
 - Obtains reduced representation in volume but produces the same or similar analytical results
- Data discretization
 - Part of data reduction but with particular importance, especially for numerical data

Forms of Data Preprocessing



Data Cleaning

- Importance
 - “Data cleaning is one of the three biggest problems in data warehousing”—Ralph Kimball
 - “Data cleaning is the number one problem in data warehousing”—DCI survey
- Data cleaning tasks
 - Fill in missing values
 - Identify outliers and smooth out noisy data
 - Correct inconsistent data
 - Resolve redundancy caused by data integration

Missing Data

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

How to Handle Missing Data?

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

Noisy Data

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

How to Handle Noisy Data?

- Binning
 - first sort data and partition into (equal-frequency) bins
 - then one can smooth by bin means, smooth by bin median, smooth by bin boundaries, etc.
- Regression
 - smooth by fitting the data into regression functions
- Clustering
 - detect and remove outliers
- Combined computer and human inspection
 - detect suspicious values and check by human (e.g., deal with possible outliers)

Simple Discretization Methods: Binning

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

Binning Methods for Data Smoothing

❑ Sorted data for price (in dollars): 4, 8, 9, 15, 21, 21, 24, 25, 26, 28, 29, 34

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

- Bin 1: 4, 8, 9, 15
- Bin 2: 21, 21, 24, 25
- Bin 3: 26, 28, 29, 34

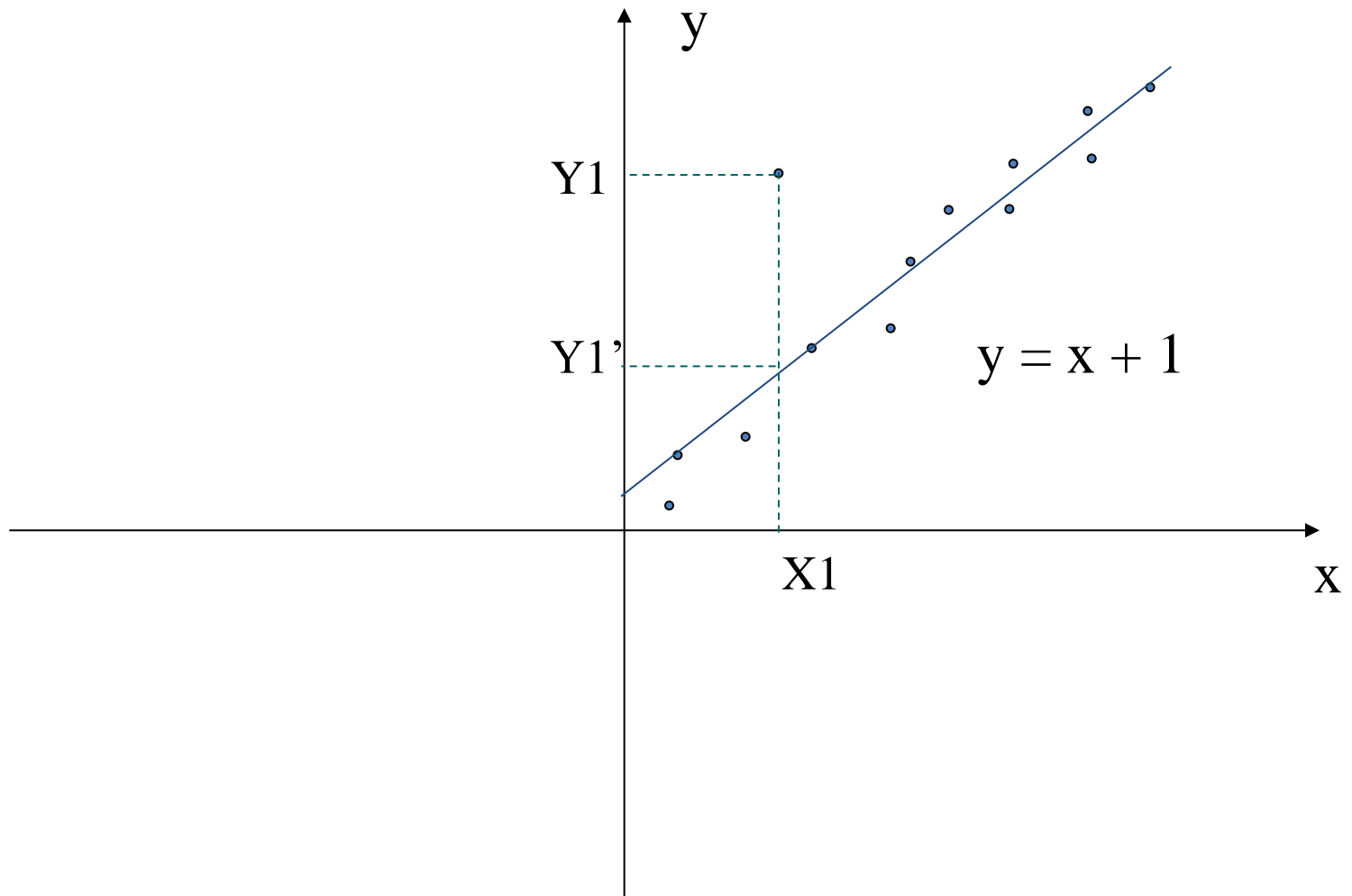
* Smoothing by bin means:

- Bin 1: 9, 9, 9, 9
- Bin 2: 23, 23, 23, 23
- Bin 3: 29, 29, 29, 29

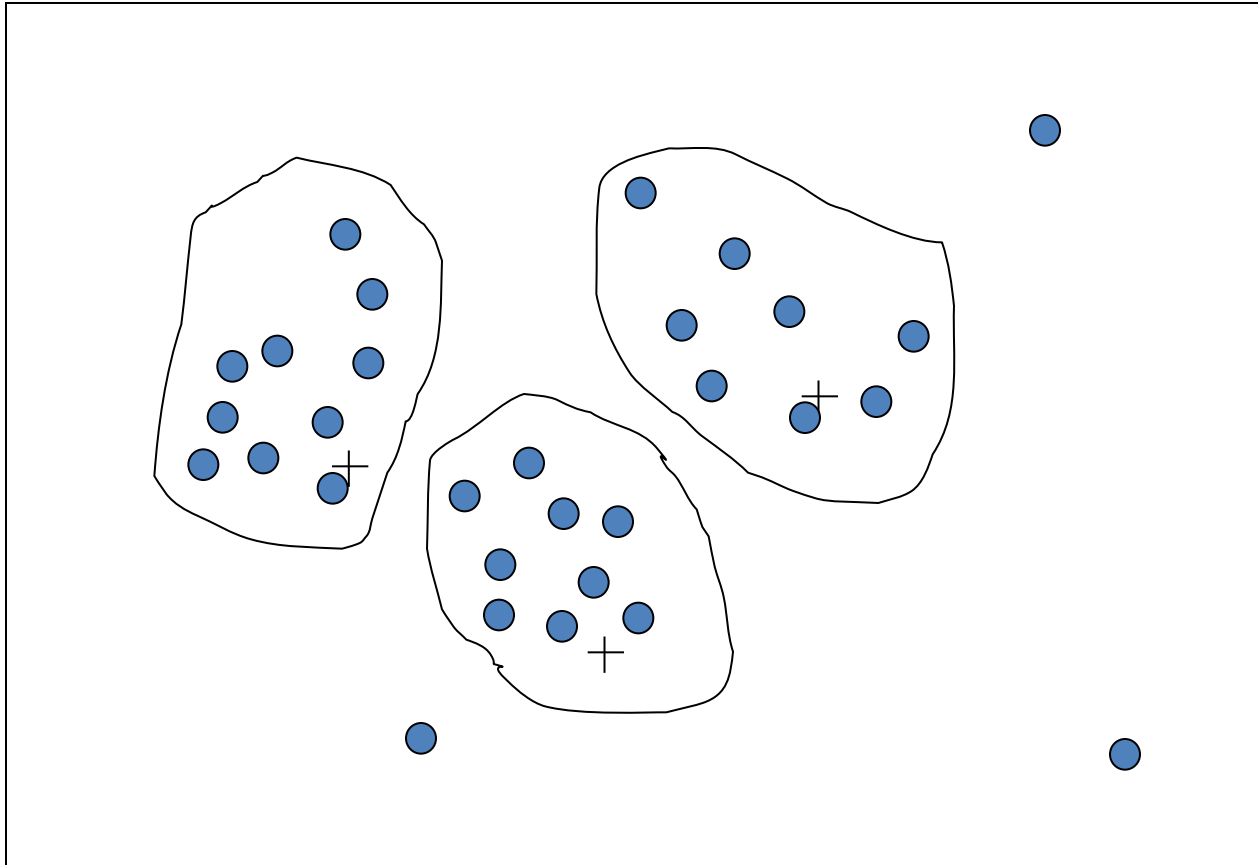
* Smoothing by bin boundaries:

- Bin 1: 4, 4, 4, 15
- Bin 2: 21, 21, 25, 25
- Bin 3: 26, 26, 26, 34

Regression



Cluster Analysis



Data Cleaning as a Process

- Data discrepancy detection
 - Use metadata (e.g., domain, range, dependency, distribution)
 - Check field overloading
 - Check uniqueness rule, consecutive rule and null rule
 - Use commercial tools
 - Data scrubbing: use simple domain knowledge (e.g., postal code, spell-check) to detect errors and make corrections
 - Data auditing: by analyzing data to discover rules and relationship to detect violators (e.g., correlation and clustering to find outliers)
- Data migration and integration
 - Data migration tools: allow transformations to be specified
 - ETL (Extraction/Transformation/Loading) tools: allow users to specify transformations through a graphical user interface
- Integration of the two processes
 - Iterative and interactive (e.g., Potter's Wheels)

Data Integration

- Data integration:
 - Combines data from multiple sources into a coherent store
- Schema integration: e.g., $A.cust-id \equiv B.cust-\#$
 - Integrate metadata from different sources
- Entity identification problem:
 - Identify real world entities from multiple data sources, e.g.,
Bill Clinton = William Clinton
- Detecting and resolving data value conflicts
 - For the same real world entity, attribute values from different sources are different
 - Possible reasons: different representations, different scales, e.g., metric vs. British units

Handling Redundancy in Data Integration

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

Correlation Analysis (Numerical Data)

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

$$r_{A,B} = \frac{\sum (A - \bar{A})(B - \bar{B})}{(n-1)\sigma_A \sigma_B} = \frac{\sum (AB) - n\bar{A}\bar{B}}{(n-1)\sigma_A \sigma_B}$$

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

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

Correlation Analysis (Categorical Data)

- χ^2 (chi-square) test

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

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

Chi-Square Calculation: An Example

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

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

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

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

Data Transformation

- Smoothing: remove noise from data
- Aggregation: summarization, data cube construction
- Generalization: concept hierarchy climbing
- Normalization: scaled to fall within a small, specified range
 - min-max normalization
 - z-score normalization
 - normalization by decimal scaling
- Attribute/feature construction
 - New attributes constructed from the given ones

Data Transformation: Normalization

- Min-max normalization: to $[new_min_A, new_max_A]$

$$v' = \frac{v - min_A}{max_A - min_A} (new_max_A - new_min_A) + new_min_A$$

- Ex. Let income range \$12,000 to \$98,000 normalized to $[0.0, 1.0]$.

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

- Z-score normalization (μ : mean, σ : standard deviation):

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

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

- Normalization by decimal scaling

$$v' = \frac{v}{10^j} \quad \text{Where } j \text{ is the smallest integer such that } \text{Max}(|v'|) < 1$$

Data Reduction Strategies

- Why data reduction?
 - A database/data warehouse may store terabytes of data
 - Complex data analysis/mining may take a very long time to run on the complete data set
- Data reduction
 - Obtain a reduced representation of the data set that is much smaller in volume but yet produce the same (or almost the same) analytical results
- Data reduction strategies
 - Data cube aggregation:
 - Dimensionality reduction — e.g., remove unimportant attributes
 - Data Compression
 - Numerosity reduction — e.g., fit data into models
 - Discretization and concept hierarchy generation

Data Cube Aggregation

- The lowest level of a data cube (base cuboid)
 - The aggregated data for an **individual entity of interest**
 - E.g., a customer in a phone calling data warehouse
- Multiple levels of aggregation in data cubes
 - Further reduce the size of data to deal with
- Reference appropriate levels
 - Use the smallest representation which is enough to solve the task
- Queries regarding aggregated information should be answered using data cube, when possible

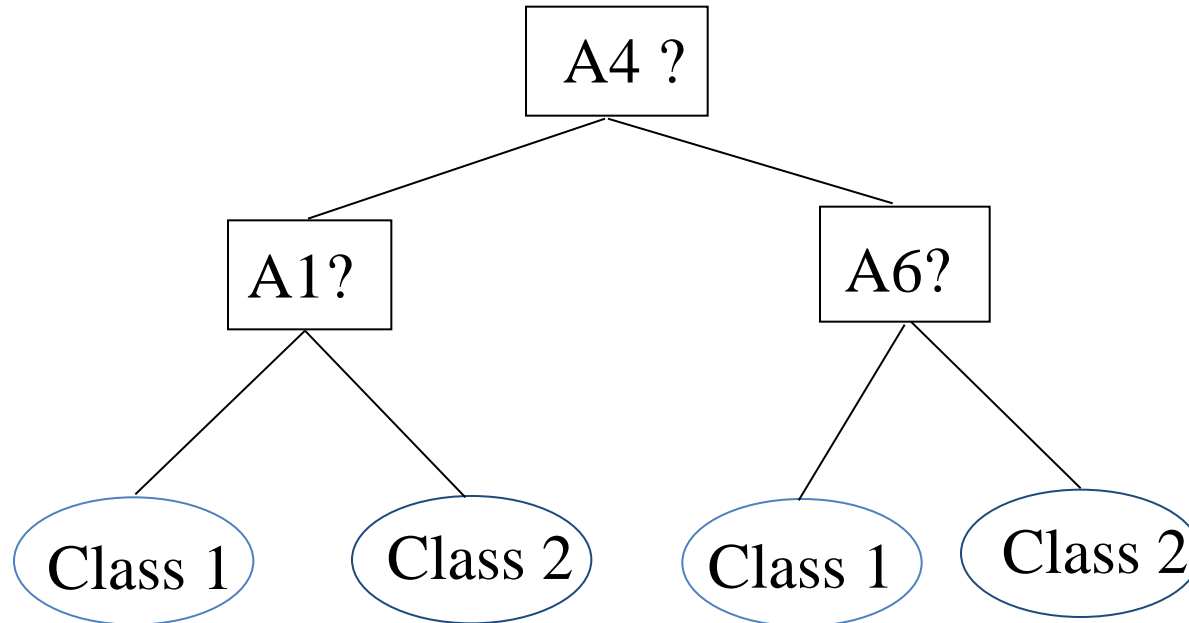
Attribute Subset Selection

- Feature selection (i.e., attribute subset selection):
 - Select a minimum set of features such that the probability distribution of different classes given the values for those features is as close as possible to the original distribution given the values of all features
 - reduce # of patterns in the patterns, easier to understand
- Heuristic methods (due to exponential # of choices):
 - Step-wise forward selection
 - Step-wise backward elimination
 - Combining forward selection and backward elimination
 - Decision-tree induction

Example of Decision Tree Induction

Initial attribute set:

{A1, A2, A3, A4, A5, A6}



-----> Reduced attribute set: {A1, A4, A6}

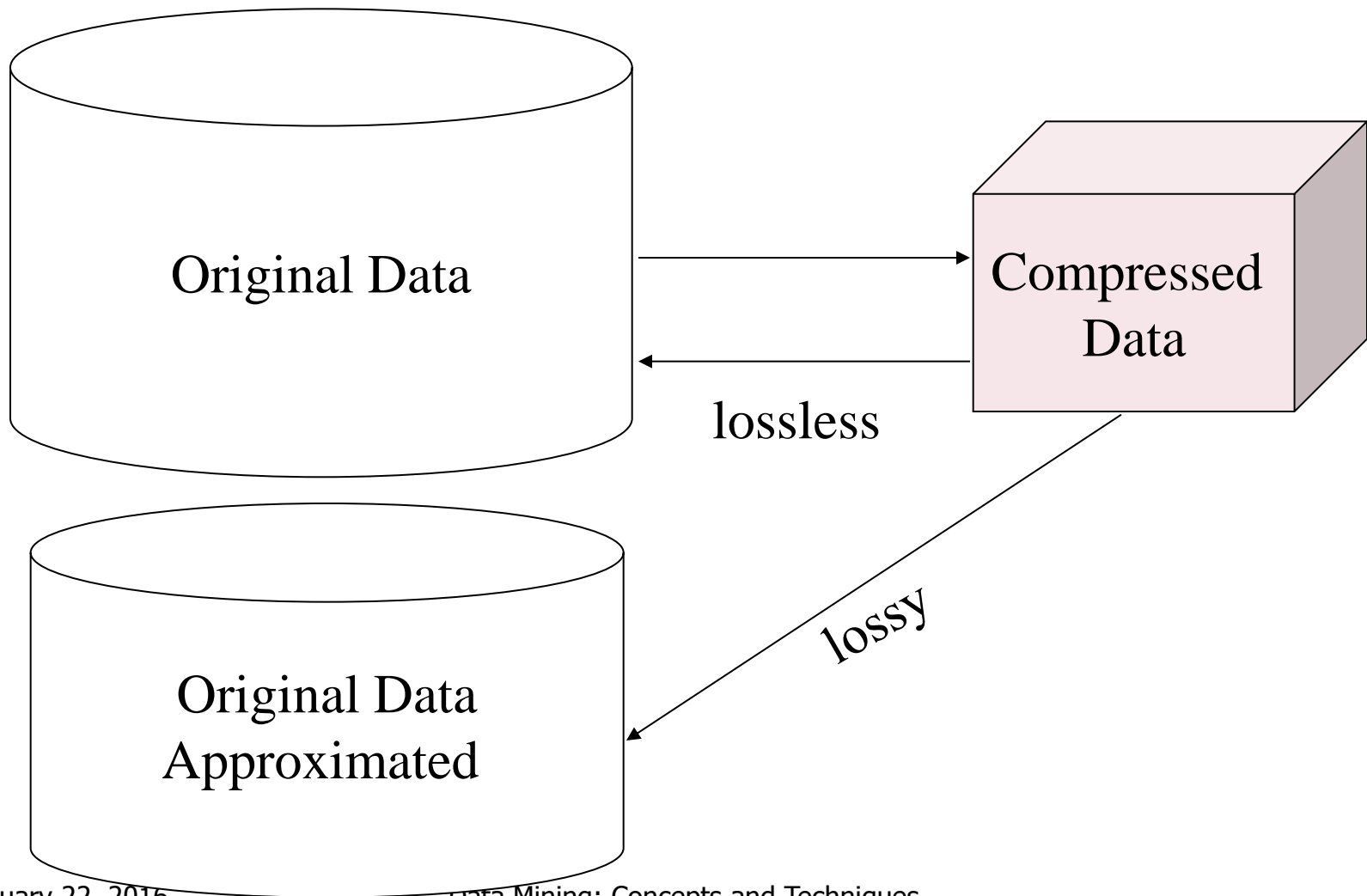
Heuristic Feature Selection Methods

- There are 2^d possible sub-features of d features
- Several heuristic feature selection methods:
 - Best single features under the feature independence assumption: choose by significance tests
 - Best step-wise feature selection:
 - The best single-feature is picked first
 - Then next best feature condition to the first, ...
 - Step-wise feature elimination:
 - Repeatedly eliminate the worst feature
 - Best combined feature selection and elimination
 - Optimal branch and bound:
 - Use feature elimination and backtracking

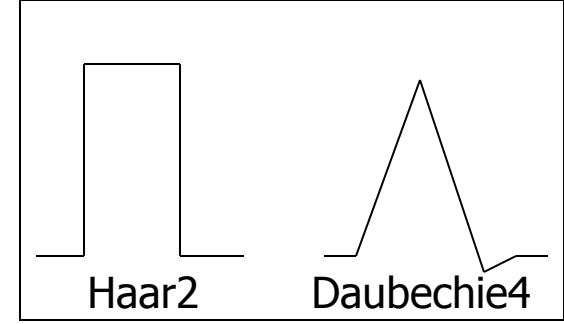
Data Compression

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

Data Compression

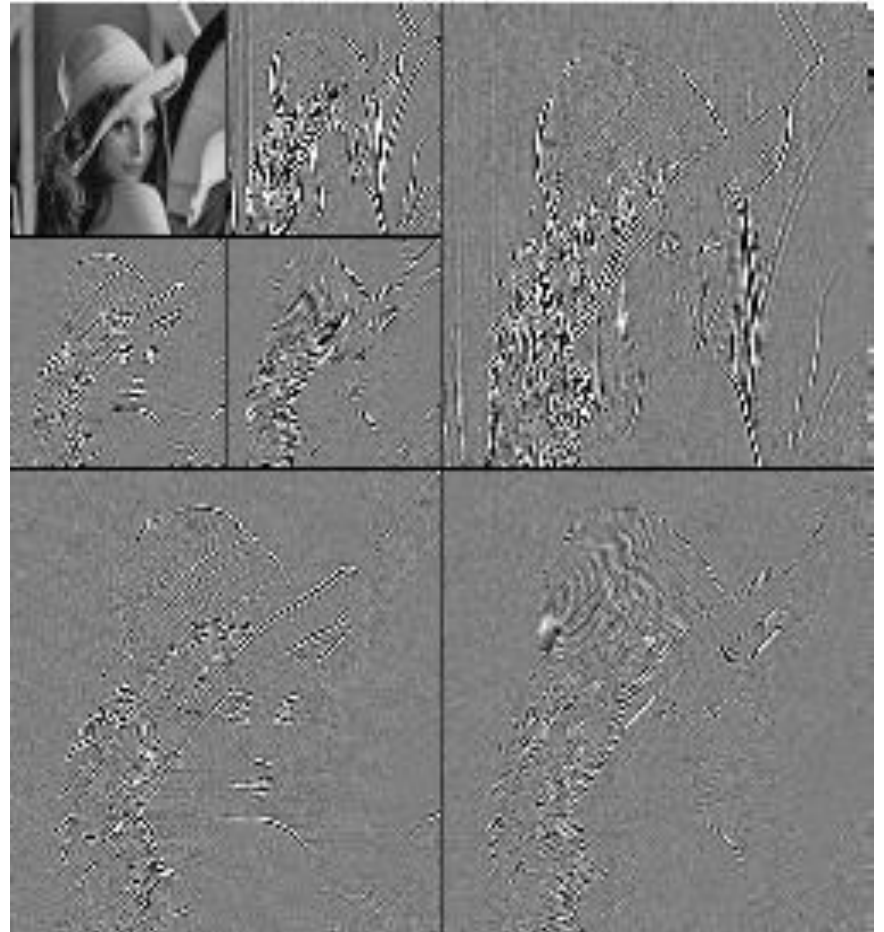
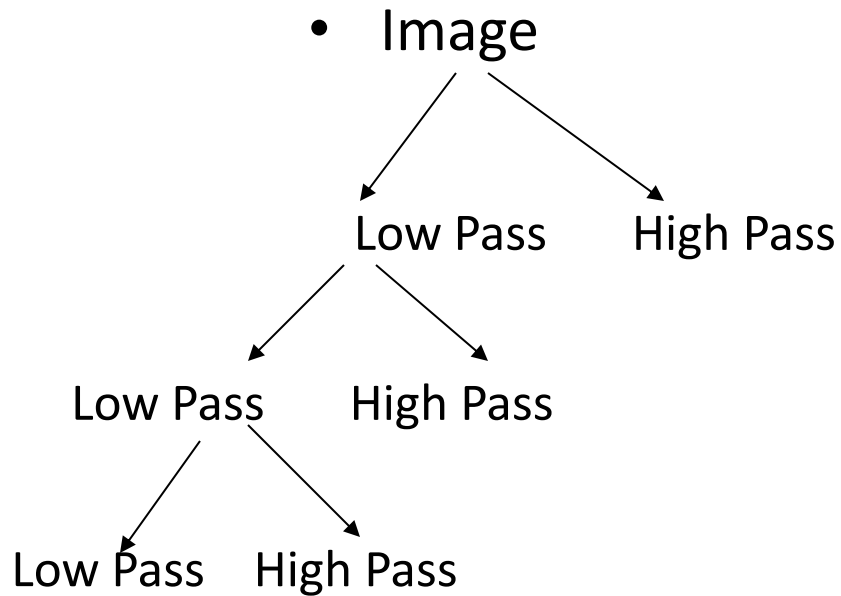


Dimensionality Reduction: Wavelet Transformation



- Discrete wavelet transform (DWT): linear signal processing, multi-resolutional analysis
- Compressed approximation: store only a small fraction of the strongest of the wavelet coefficients
- Similar to discrete Fourier transform (DFT), but better lossy compression, localized in space
- Method:
 - Length, L , must be an integer power of 2 (padding with 0's, when necessary)
 - Each transform has 2 functions: smoothing, difference
 - Applies to pairs of data, resulting in two set of data of length $L/2$
 - Applies two functions recursively, until reaches the desired length

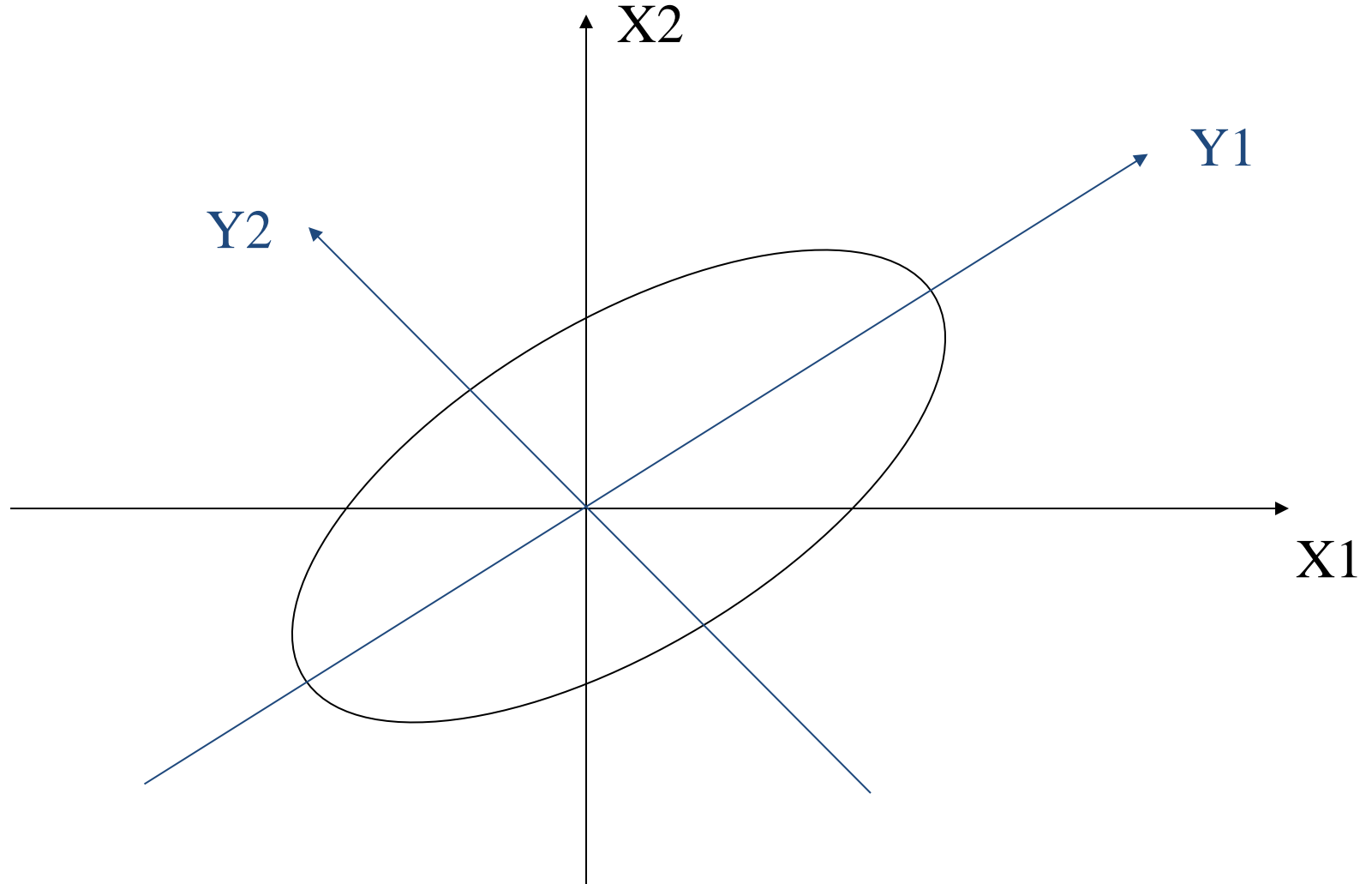
DWT for Image Compression



Dimensionality Reduction: Principal Component Analysis (PCA)

- Given N data vectors from n -dimensions, find $k \leq n$ orthogonal vectors (*principal components*) that can be best used to represent data
- Steps
 - Normalize input data: Each attribute falls within the same range
 - Compute k orthonormal (unit) vectors, i.e., *principal components*
 - Each input data (vector) is a linear combination of the k principal component vectors
 - The principal components are sorted in order of decreasing “significance” or strength
 - Since the components are sorted, the size of the data can be reduced by eliminating the weak components, i.e., those with low variance. (i.e., using the strongest principal components, it is possible to reconstruct a good approximation of the original data)
- Works for numeric data only
- Used when the number of dimensions is large

Principal Component Analysis



Numerosity Reduction

- Reduce data volume by choosing alternative, smaller forms of data representation
- Parametric methods
 - Assume the data fits some model, estimate model parameters, store only the parameters, and discard the data (except possible outliers)
 - Example: Log-linear models—obtain value at a point in m - D space as the product on appropriate marginal subspaces
- Non-parametric methods
 - Do not assume models
 - Major families: histograms, clustering, sampling

Data Reduction Method (1): Regression and Log-Linear Models

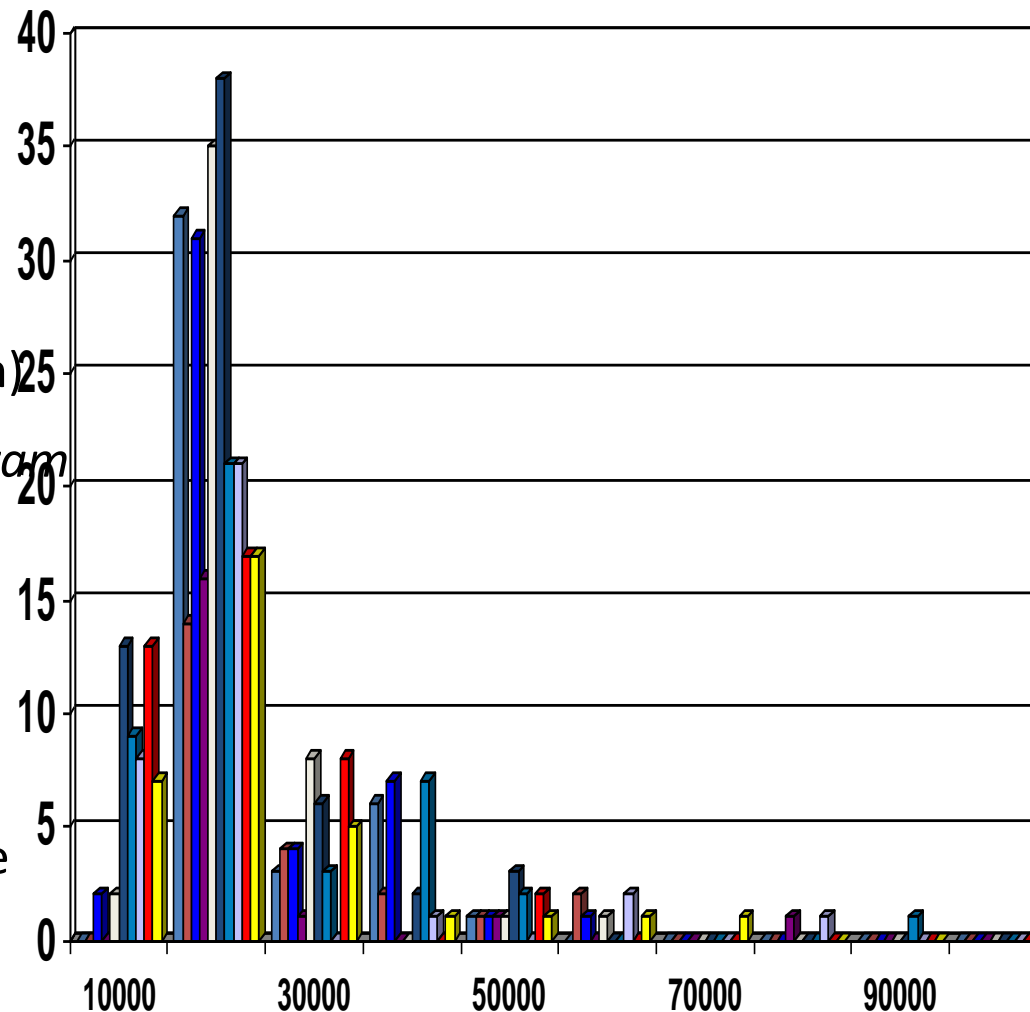
- Linear regression: Data are modeled to fit a straight line
 - Often uses the least-square method to fit the line
- Multiple regression: allows a response variable Y to be modeled as a linear function of multidimensional feature vector
- Log-linear model: approximates discrete multidimensional probability distributions

Regress Analysis and Log-Linear Models

- Linear regression: $Y = w X + b$
 - Two regression coefficients, w and b , specify the line and are to be estimated by using the data at hand
 - Using the least squares criterion to the known values of $Y_1, Y_2, \dots, X_1, X_2, \dots$
- Multiple regression: $Y = b_0 + b_1 X_1 + b_2 X_2$.
 - Many nonlinear functions can be transformed into the above
- Log-linear models:
 - The multi-way table of joint probabilities is approximated by a product of lower-order tables
 - Probability: $p(a, b, c, d) = \alpha_{ab} \beta_{ac} \chi_{ad} \delta_{bcd}$

Data Reduction Method (2): Histograms

- Divide data into buckets and store average (sum) for each bucket
- Partitioning rules:
 - Equal-width: equal bucket range
 - Equal-frequency (or equal-depth)
 - V-optimal: with the least *histogram variance* (weighted sum of the original values that each bucket represents)
 - MaxDiff: set bucket boundary between each pair for pairs have the $\beta-1$ largest differences



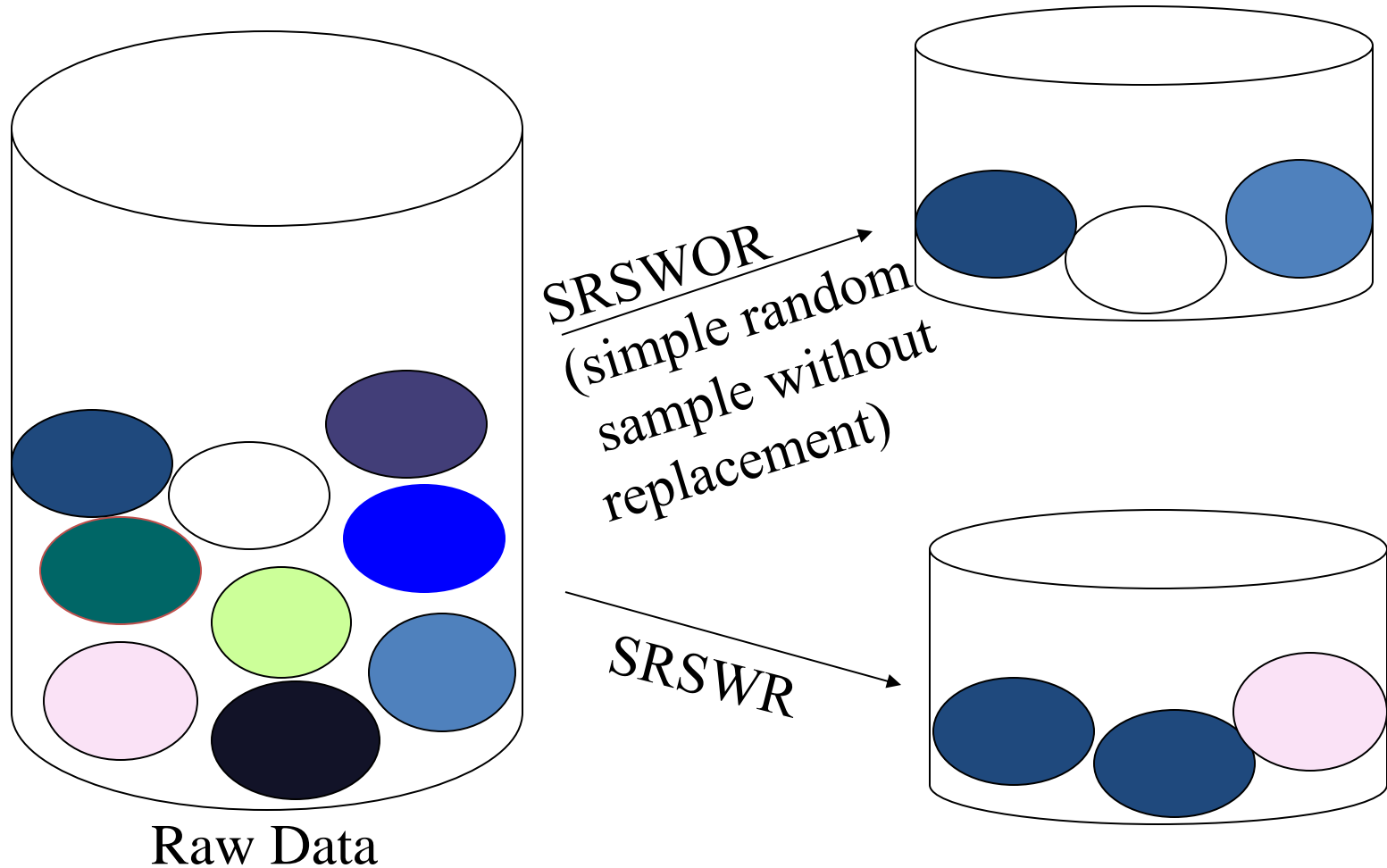
Data Reduction Method (3): Clustering

- Partition data set into clusters based on similarity, and store cluster representation (e.g., centroid and diameter) only
- Can be very effective if data is clustered but not if data is “smeared”
- Can have hierarchical clustering and be stored in multi-dimensional index tree structures
- There are many choices of clustering definitions and clustering algorithms
- Cluster analysis will be studied in depth in Chapter 7

Data Reduction Method (4): Sampling

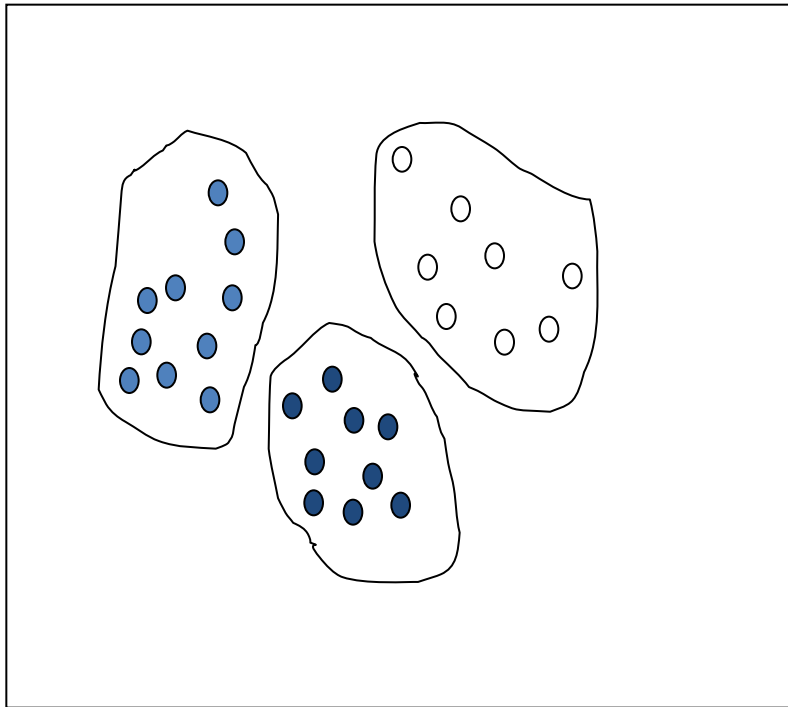
- Sampling: obtaining a small sample s to represent the whole data set N
- Allow a mining algorithm to run in complexity that is potentially sub-linear to the size of the data
- Choose a **representative** subset of the data
 - Simple random sampling may have very poor performance in the presence of skew
- Develop adaptive sampling methods
 - Stratified sampling:
 - Approximate the percentage of each class (or subpopulation of interest) in the overall database
 - Used in conjunction with skewed data
- Note: Sampling may not reduce database I/Os (page at a time)

Sampling: with or without Replacement

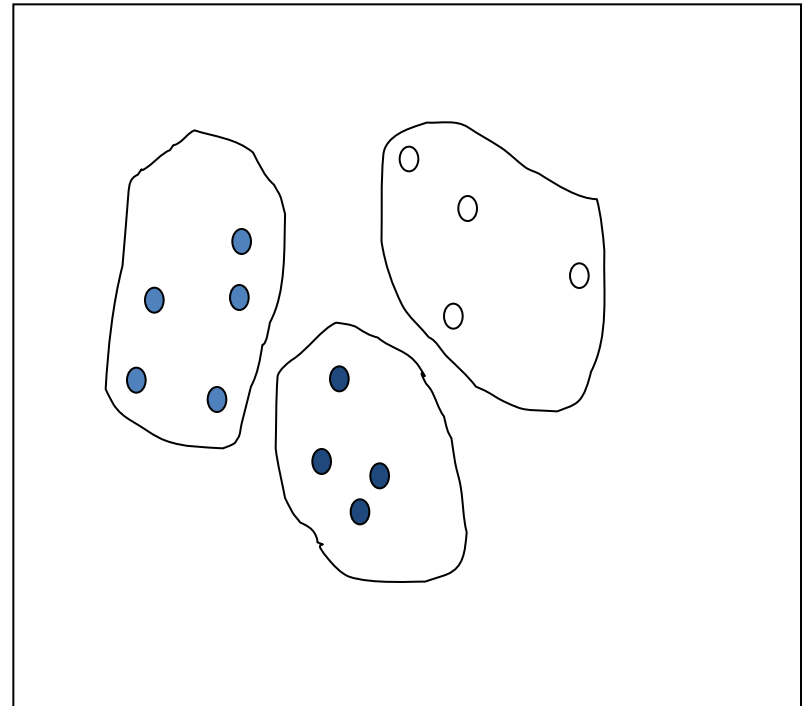


Sampling: Cluster or Stratified Sampling

Raw Data



Cluster/Stratified Sample



Difficulties of implementing Data Warehouses

- Lead time is huge in building a data warehouse
 - Potentially it takes years to build and efficiently maintain a data warehouse.
- Both quality and consistency of data are major concerns.
- Revising the usage projections regularly to meet the current requirements.
 - The data warehouse should be designed to accommodate addition and attrition of data sources without major redesign
- Administration of data warehouse would require far broader skills than are needed for a traditional database.

Open Issues in Data Warehousing

- Data cleaning, indexing, partitioning, and views could be given new attention with perspective to data warehousing.
- Automation of
 - data acquisition
 - data quality management
 - selection and construction of access paths and structures
 - self-maintainability
 - functionality and performance optimization
- Incorporating of domain and business rules appropriately into the warehouse creation and maintenance process more intelligently.

Disadvantages of data warehouses

- Data warehouses are not the optimal environment for unstructured data.
- Because data must be extracted, transformed and loaded into the warehouse, there is an element of latency in data warehouse data.
- Over their life, data warehouses can have high costs. Maintenance costs are high.
- Data warehouses can get outdated relatively quickly. There is a cost of delivering suboptimal information to the organization.
- There is often a fine line between data warehouses and operational systems. Duplicate, expensive functionality may be developed. Or, functionality may be developed in the data warehouse that, in retrospect, should have been developed in the operational systems and vice versa.

Data Marts

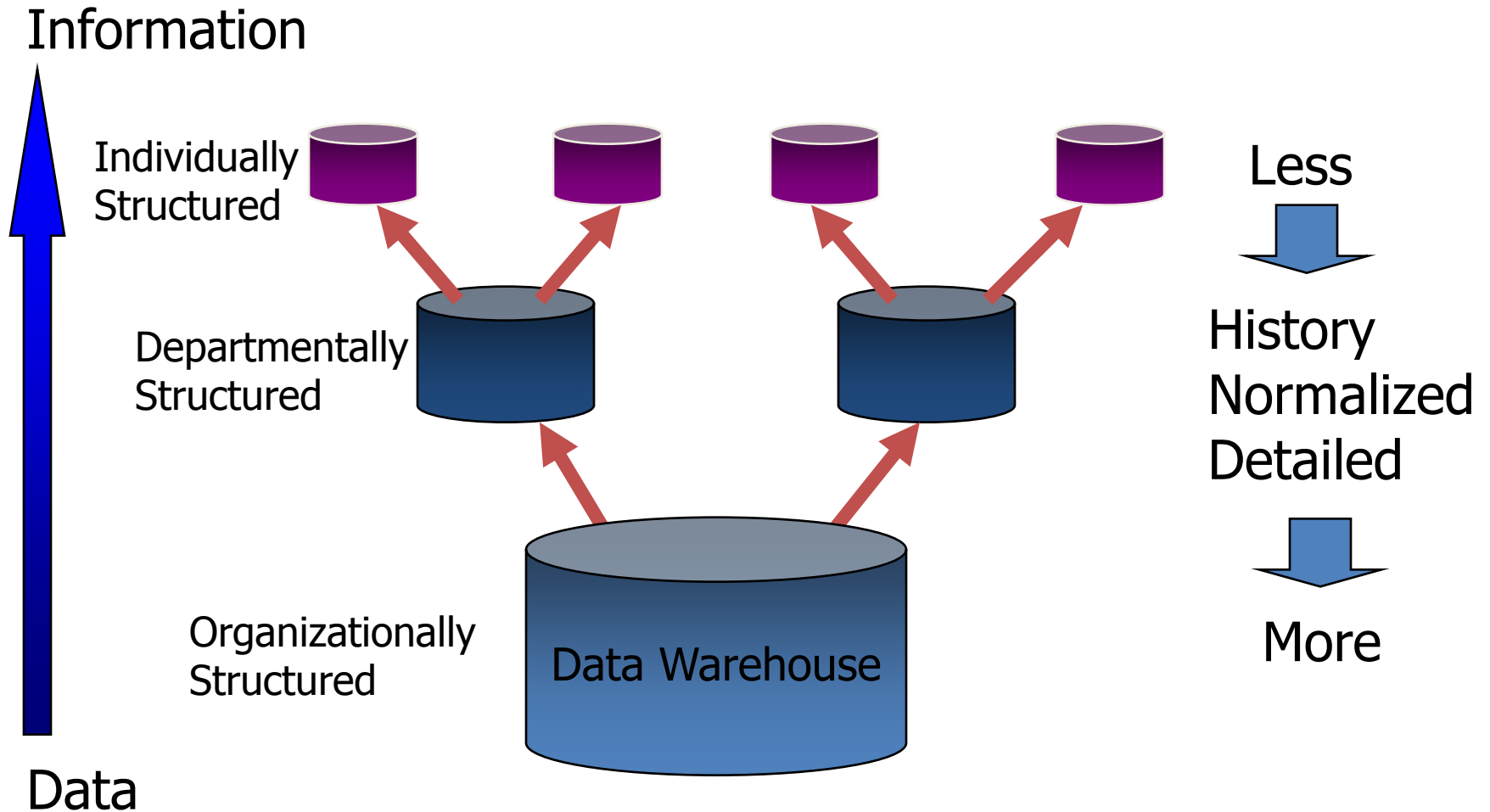
Data Marts

- A data mart is a scaled down version of a data warehouse that focuses on a particular subject area.
- A **data mart** is a subset of an organizational data store, usually oriented to a specific purpose or major data subject, that may be distributed to support business needs.
- Data marts are analytical data stores designed to focus on specific business functions for a specific community within an organization.
- Usually designed to support the unique business requirements of a specified department or business process
- Implemented as the first step in proving the usefulness of the technologies to solve business problems

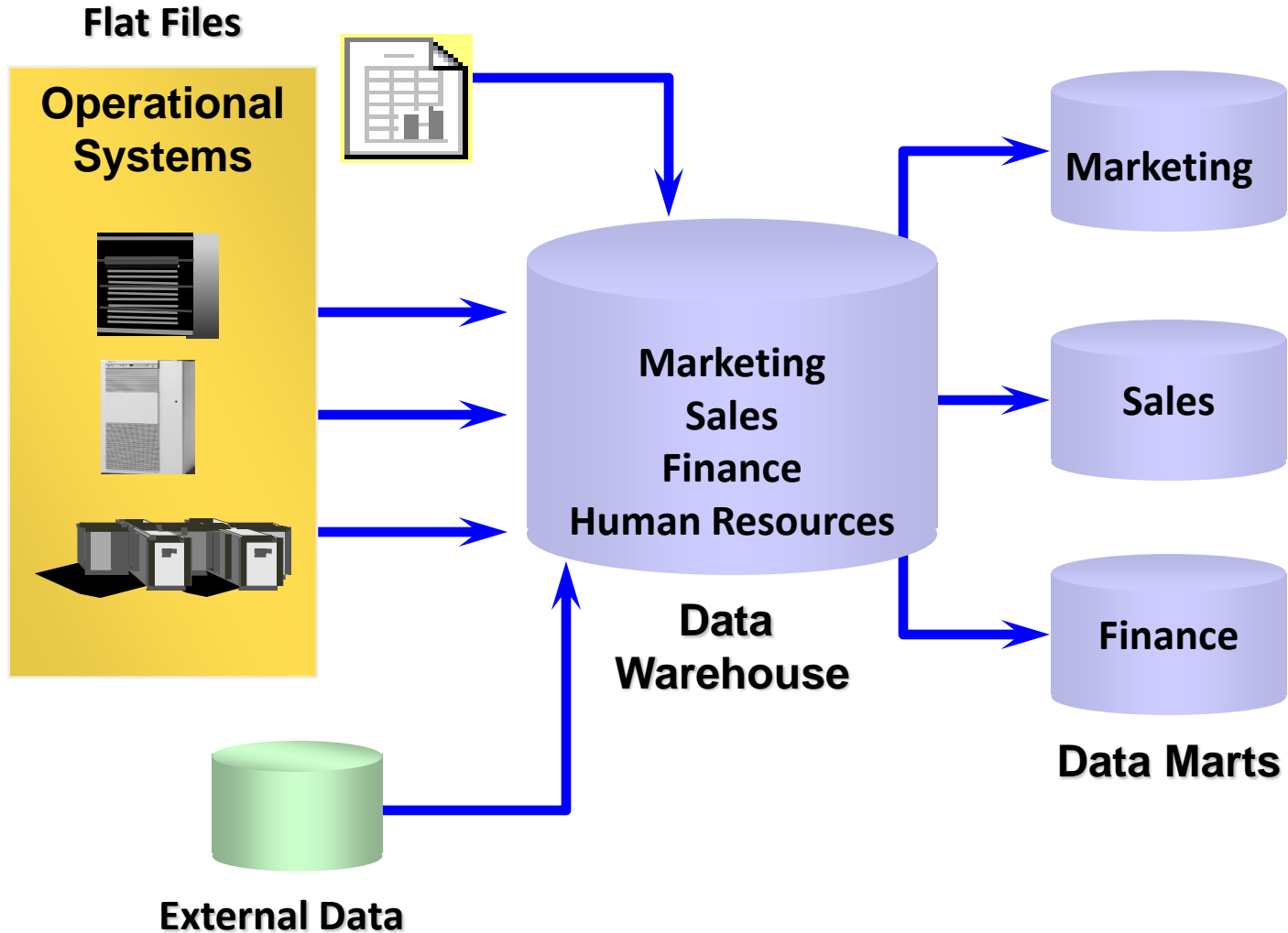
Reasons for creating a data mart

- Easy access to frequently needed data
- Creates collective view by a group of users
- Improves end-user response time
- Ease of creation in less time
- Lower cost than implementing a full Data warehouse
- Potential users are more clearly defined than in a full Data warehouse

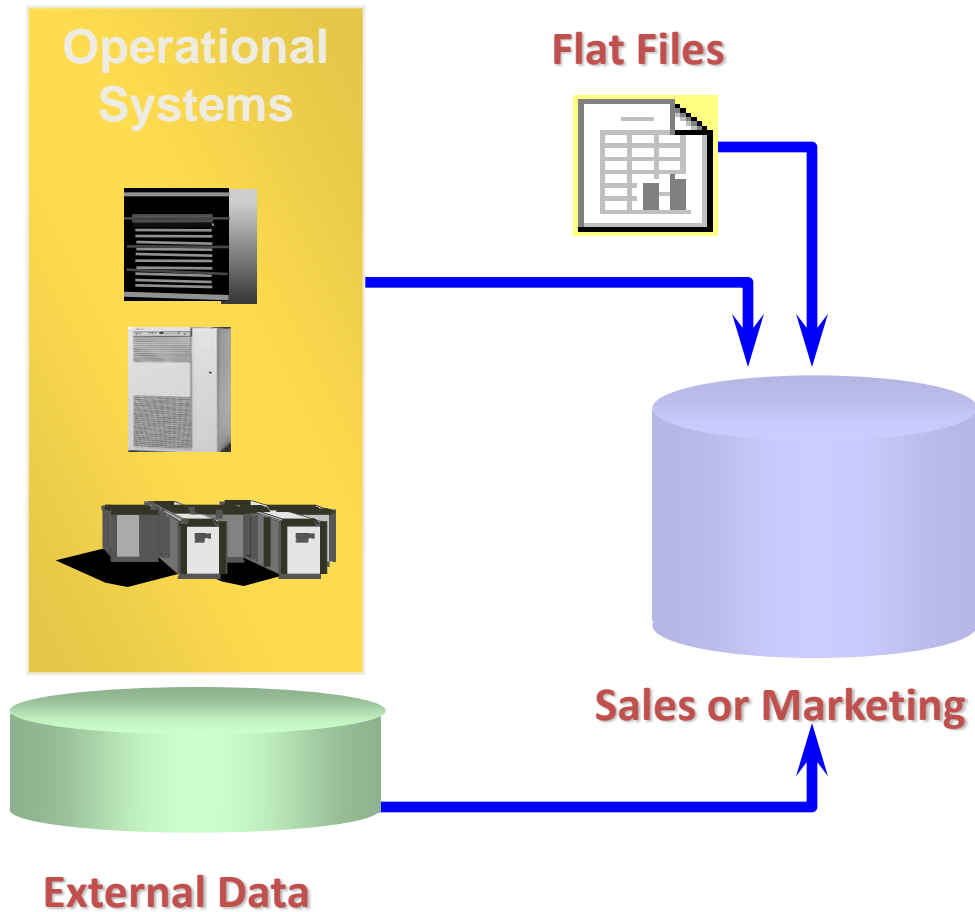
From the Data Warehouse to Data Marts



Dependent Data Mart



Independent Data Mart



END of UNIT I