CGS 2545: Database Concepts
Spring 2014

Data Warehousing (Chapter 9)

Instructor : Dr. Mark Llewellyn
markl@cs.ucf.edu
HEC 236, 407-823-2790
http://www.cs.ucf.edu/courses/cgs2545/spr2014

Department of Electrical Engineering and Computer Science
Computer Science Division
University of Central Florida

Introduction to Parallel and Distributed Database Systems

- So far in this course, we have considered centralized DBMSs in which all of the data is maintained at a single site. We further assumed that processing individual transactions was essentially sequential.

- One of the most important trends in databases is the increased use of parallel evaluation techniques (parallel DBMS) and data distribution (distributed DBMS).

- We will focus primarily on distributed database management systems as they are related to data warehousing.

Parallel Database Systems

- A parallel database system seeks to improve performance of the database through the parallelization of various operations of the DBMS.

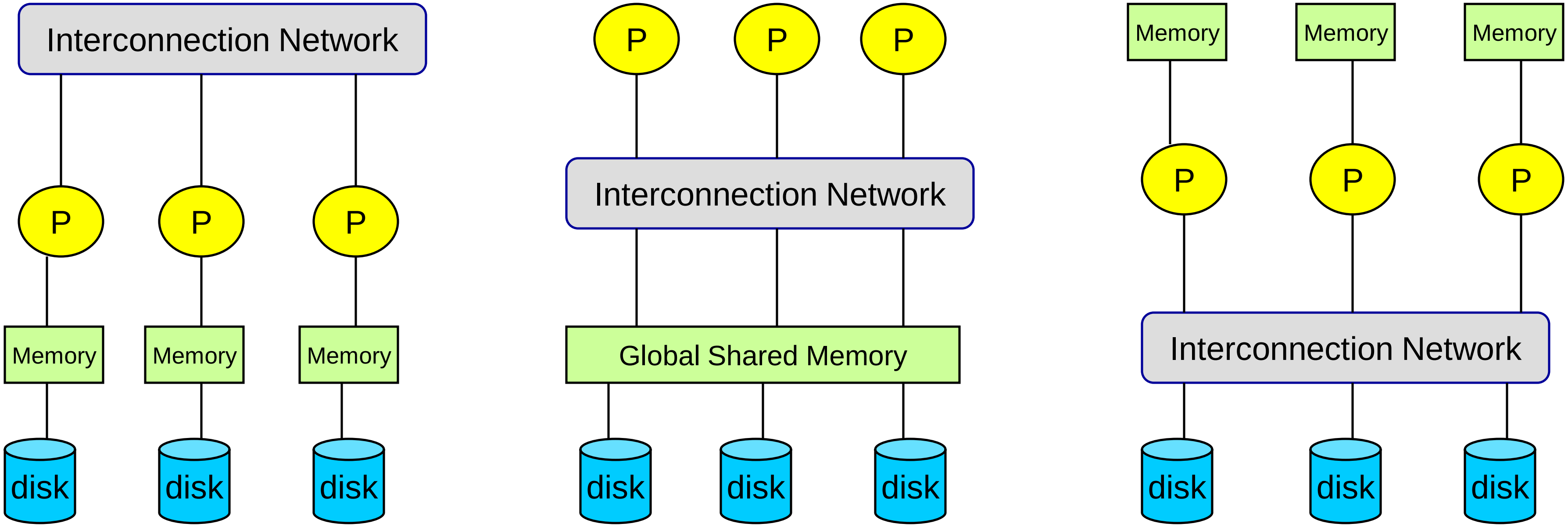
- Parallelization can occur:
  - in the loading of data
  - building/searching indices
  - query evaluation

- Although it is common for data to be distributed in such a system, the distribution is governed solely by performance considerations.

Parallel Database System Architectures

- Three main architectures have been proposed for building parallel DBMSs.
- In a shared-memory system, multiple CPUs are attached to an interconnection network and can access a common region of main memory.
- In a shared-disk system, each CPU has a private memory and direct access to all disks through an interconnection network.
- In a shared-nothing system, each CPU has local main memory and disk space, but no two CPUs can access the same storage area; all communication between CPUs is through a network connection.

Parallel Database System Architectures (cont.)



Shared Nothing

Shared Memory

Shared Disk

The best architecture for parallel DBMSs

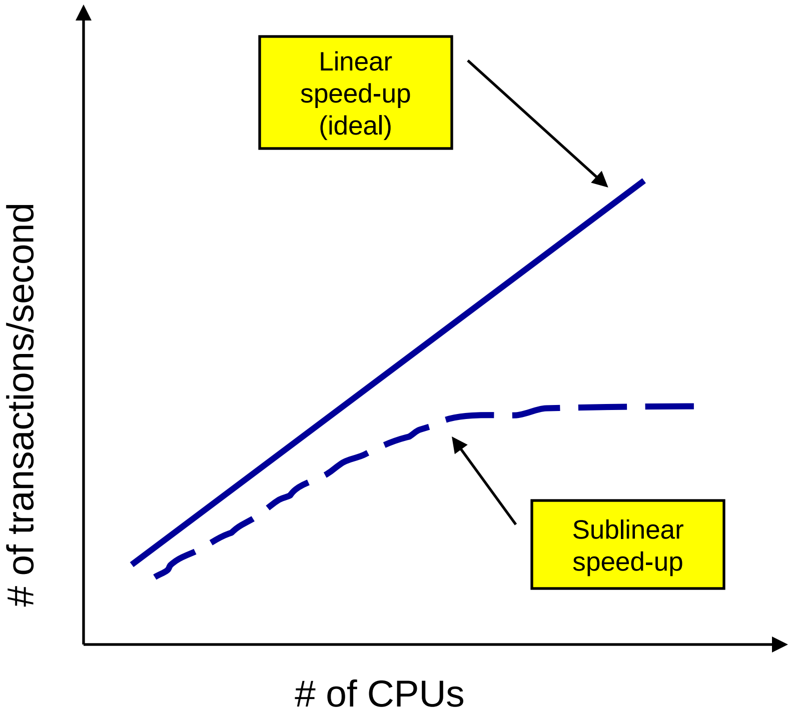
Parallel Database System Architectures (cont.)

- The basic problem with the shared-memory and shared-disk architectures is interference.
- As more CPUs are added, existing CPUs are slowed down because of the increased contention for memory accesses and network bandwidth.
- It has been shown that:
  - An average of 1% slowdown per additional CPU limits the maximum speed-up to a factor of 37.
  - Adding additional CPUs actually slows down the system.
  - A system with 1000 CPUs is only 4% as effective as a single CPU.
- These observations motivated the development of the shared-nothing architecture for large parallel database systems.

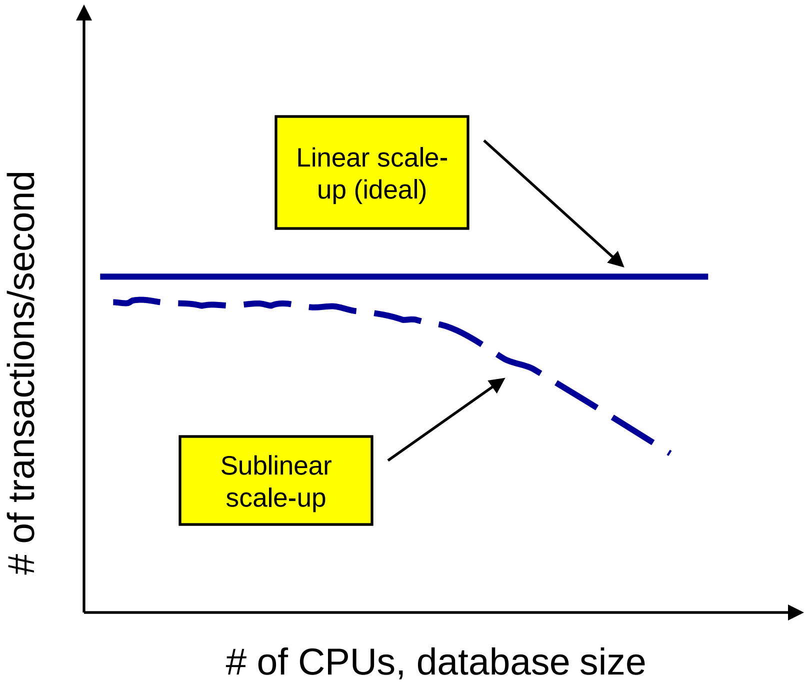
Parallel Database System Architectures (cont.)

- The shared-nothing architecture requires more extensive reorganization of the DBMS code, but it has been shown to provide a linear speed-up and linear scale-up.
- Linear speed-up occurs when the time required by an operation decreases in proportion to the increase in the number of CPUs and disks.
- Linear scale-up occurs when the performance level is sustained if the number of CPUs and disks are increased in proportion to the amount of data.
- As a result, ever-more-powerful parallel database systems can be constructed by taking advantage of the rapidly improving performance for single-CPU systems and connecting as many CPUs as desired.

Parallel Database System Architectures (cont.)



<u>Speed-up</u>



<u>Scale-up</u>

Distributed Database Systems

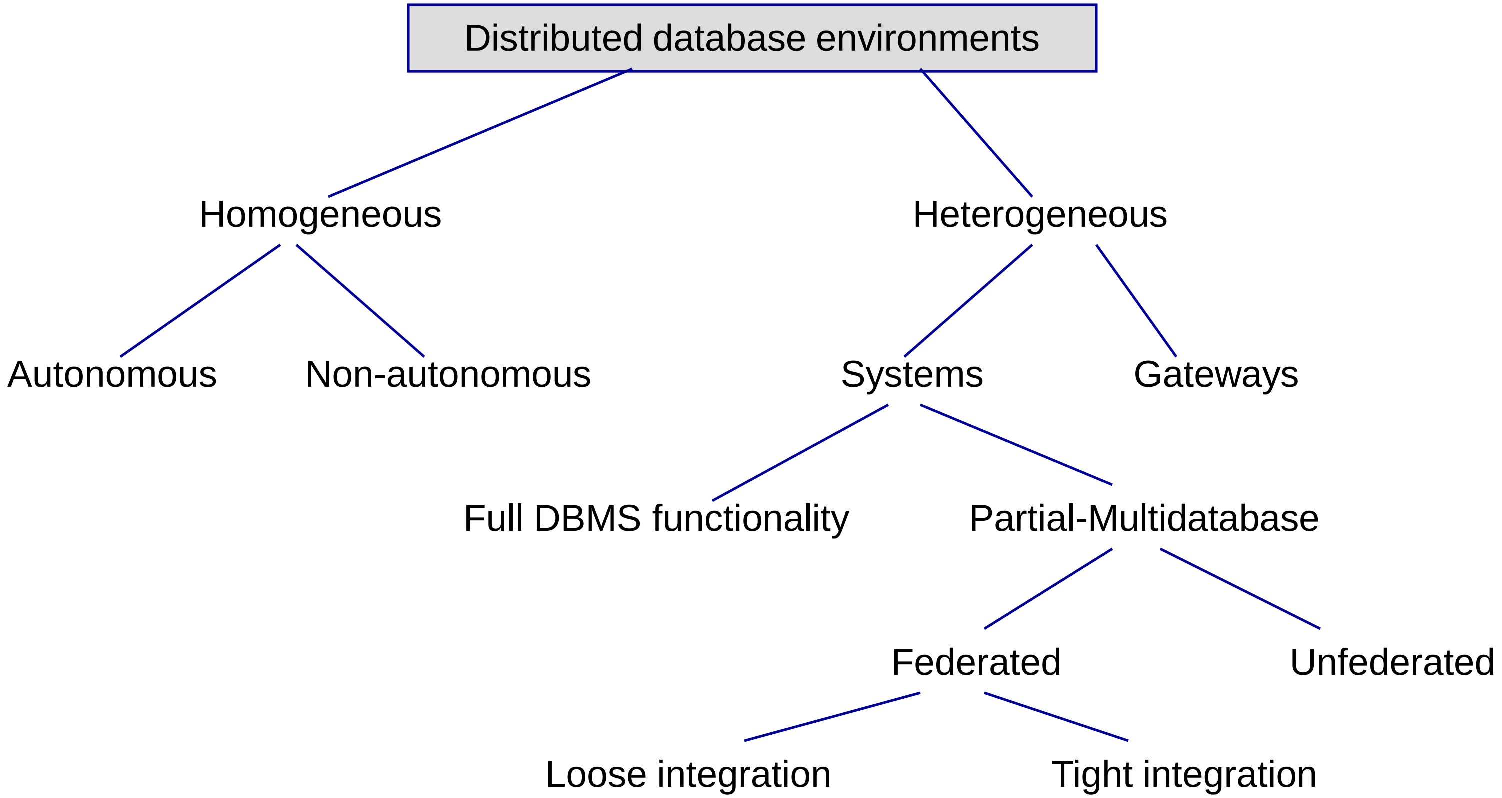
- In a distributed database system, data is physically stored across several sites, and each site is typically managed by a DBMS capable of running independent of the other sites.

- The location of the data items and the degree of autonomy of the individual sites have a significant impact on all aspects of the system, including query processing and optimization, concurrency control, and recovery.

- In contrast to parallel database systems, the distribution of data is governed by factors such as local ownership and increased availability, in addition to performance related issues.

Distributed Database Systems (cont.)

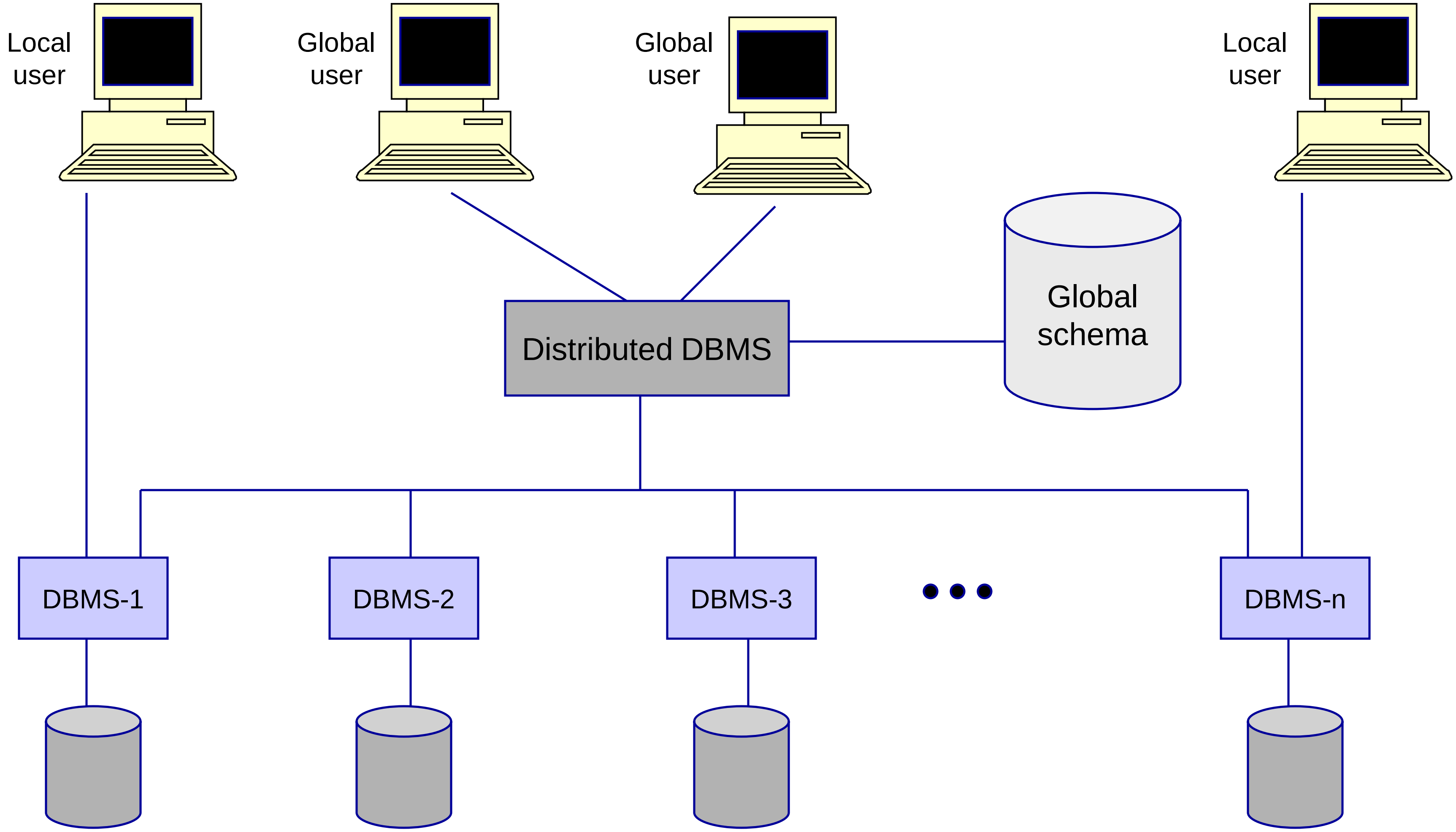
- Distributed database systems have been around since the mid-1980s. As you might expect, a variety of distributed database options exist. The diagram below shows the basic distributed database environments.



A Heterogeneous Distributed Database

- It is difficult in most organizations to force a homogeneous environment, yet heterogeneous environments are much more difficult to manage.
- As the diagram on the previous page illustrates, there are many variations of heterogeneous distributed database environments, however; a typical heterogeneous distributed database environment is defined by the following characteristics:
  - Data are distributed across all the nodes.
  - Different DBMSs may used at each location.
  - Some users require only local access to databases, which can be accomplished using only the local DBMS and schema.
  - A global schema exists, which allows local users to access remote data.

A Heterogeneous Distributed Database System

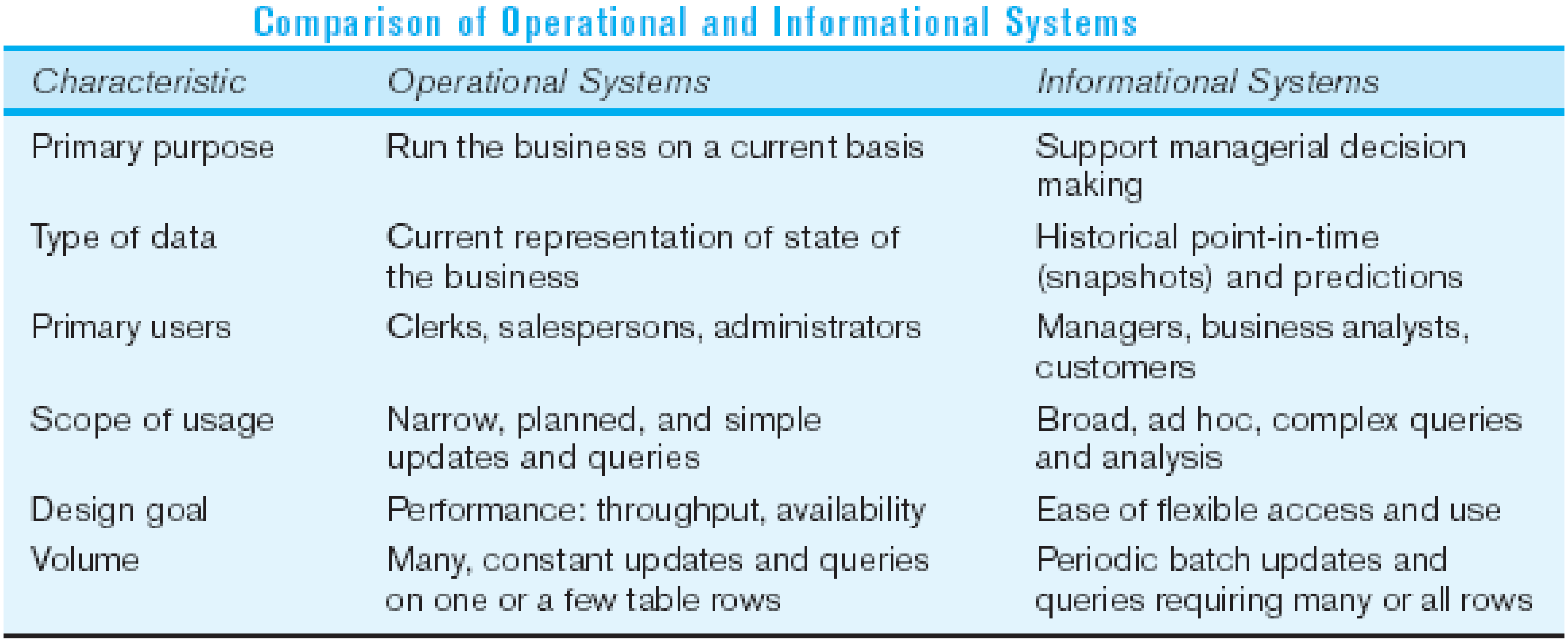


Data Warehousing Definitions

- **Data Warehouse**:
  - A subject-oriented, integrated, time-variant, non-updatable collection of data used in support of management decision-making processes
  - ***Subject-oriented:*** e.g. customers, patients, students, products
  - ***Integrated:*** Consistent naming conventions, formats, encoding structures; from multiple data sources
  - ***Time-variant:*** Can study trends and changes
  - ***Non-updatable:*** Read-only (user viewpoint), periodically refreshed
- **Data Mart**:
  - A data warehouse that is limited in scope

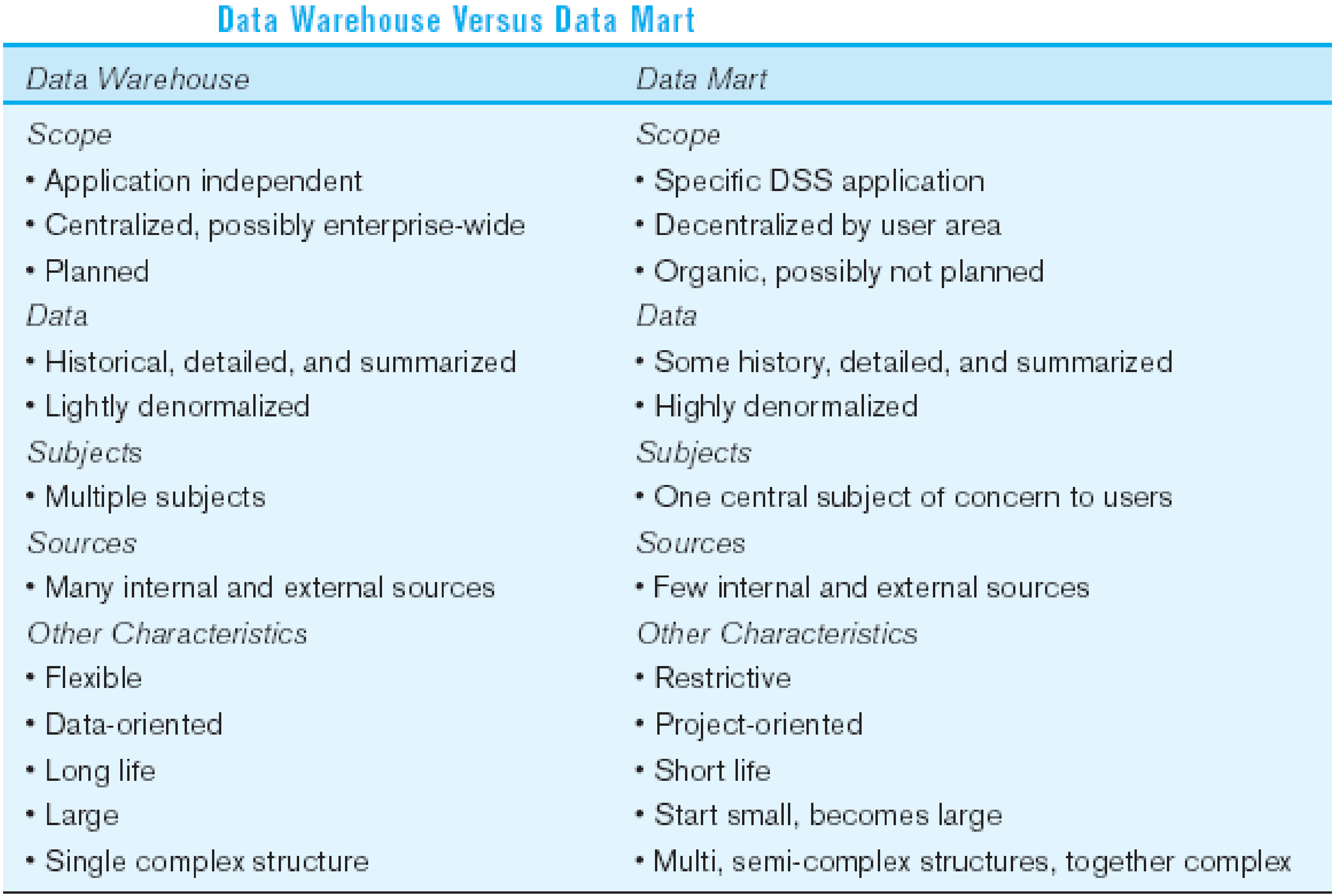
The Need for Data Warehousing

- Integrated, company-wide view of high-quality information (from disparate databases).
- Separation of *operational* and *informational* systems and data (for improved performance).



Comparison of Operational and Informational Systems

Characteristic	*Operational Systems*	*Informational Systems*
Primary purpose	Run the business on a current basis	Support managerial decision making
Type of data	Current representation of state of the business	Historical point-in-time (snapshots) and predictions
Primary users	Clerks, salespersons, administrators	Managers, business analysts, customers
Scope of usage	Narrow, planned, and simple updates and queries	Broad, ad hoc, complex queries and analysis
Design goal	Performance: throughput, availability	Ease of flexible access and use
Volume	Many, constant updates and queries on one or a few table rows	Periodic batch updates and queries requiring many or all rows



Data Warehouse Versus Data Mart

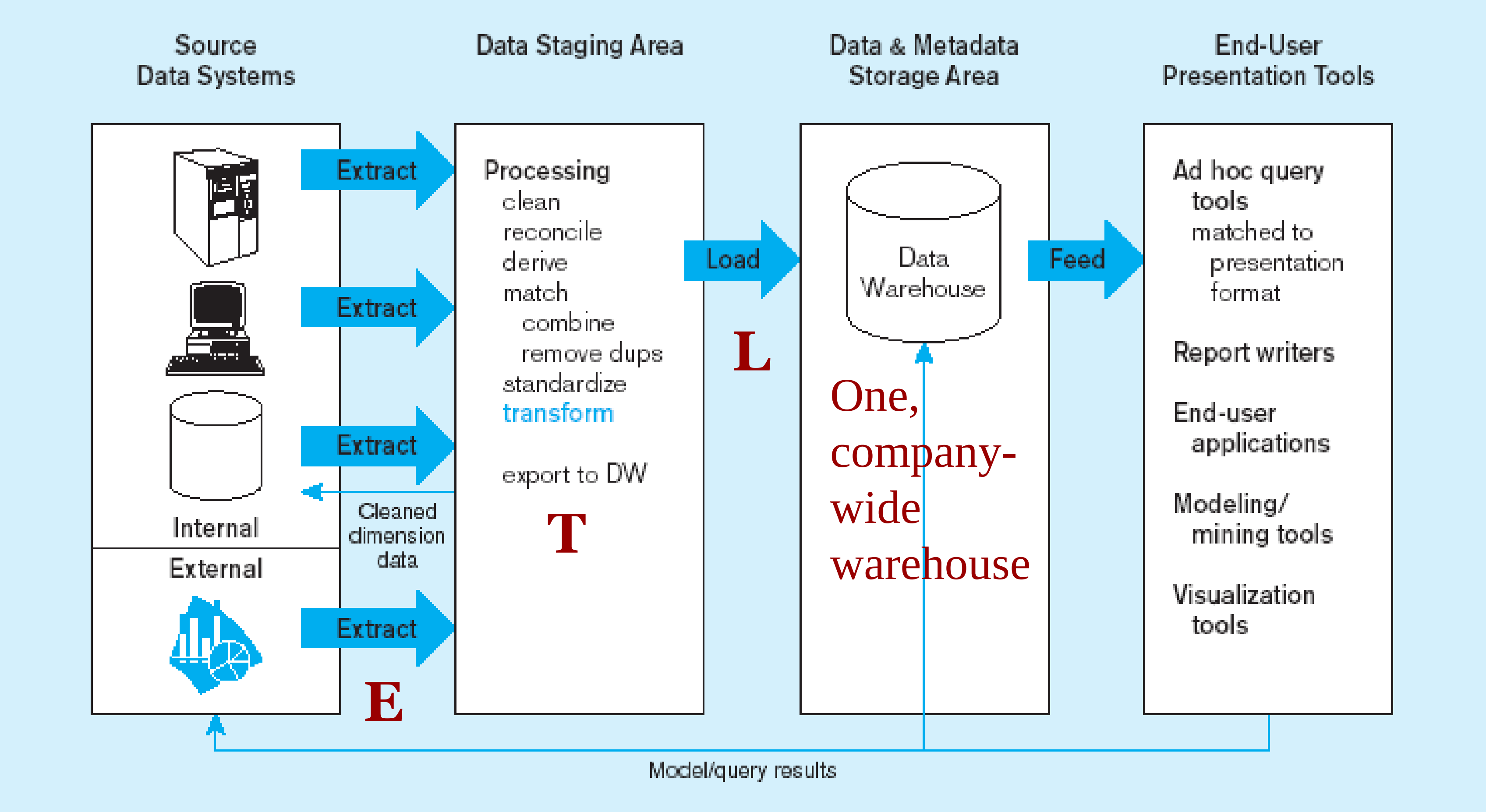
Data Warehouse	*Data Mart*
Scope	*Scope*
• Application independent	• Specific DSS application
• Centralized, possibly enterprise-wide	• Decentralized by user area
• Planned	• Organic, possibly not planned
Data	*Data*
• Historical, detailed, and summarized	• Some history, detailed, and summarized
• Lightly denormalized	• Highly denormalized
Subjects	*Subjects*
• Multiple subjects	• One central subject of concern to users
Sources	*Sources*
• Many internal and external sources	• Few internal and external sources
Other Characteristics	*Other Characteristics*
• Flexible	• Restrictive
• Data-oriented	• Project-oriented
• Long life	• Short life
• Large	• Start small, becomes large
• Single complex structure	• Multi, semi-complex structures, together complex

Data Warehouse Architectures

- Generic Two-Level Architecture
- Independent Data Mart
- Dependent Data Mart and Operational Data Store
- Logical Data Mart and Real-Time Data Warehouse
- Three-Layer architecture

All involve some form of ***extraction***, ***transformation*** and ***loading*** (**ETL**).

Generic two-level data warehousing architecture

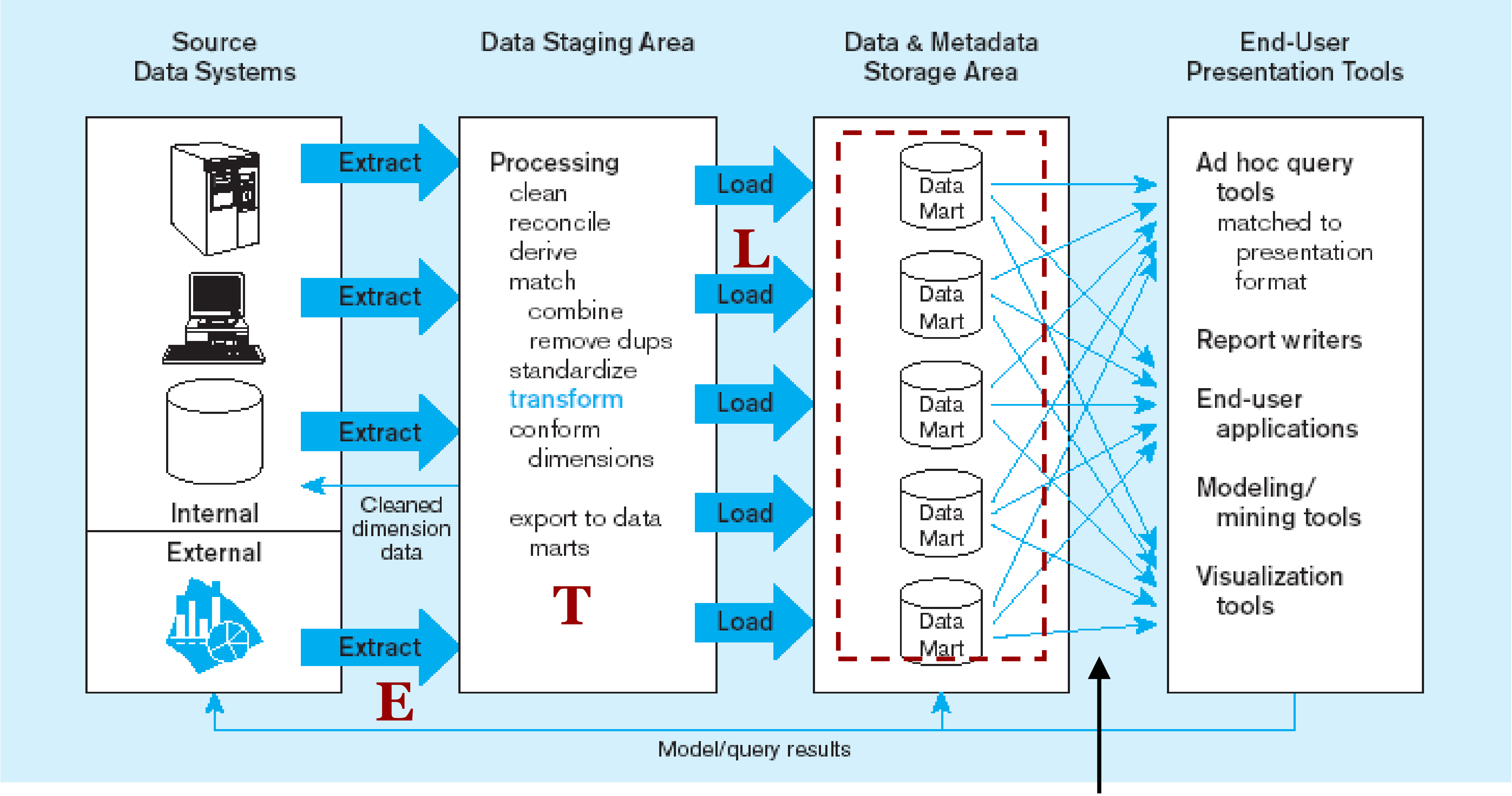


Periodic extraction ➔ data is not completely current in warehouse

Independent data mart data warehousing architecture

Data marts:

Mini-warehouses, limited in scope

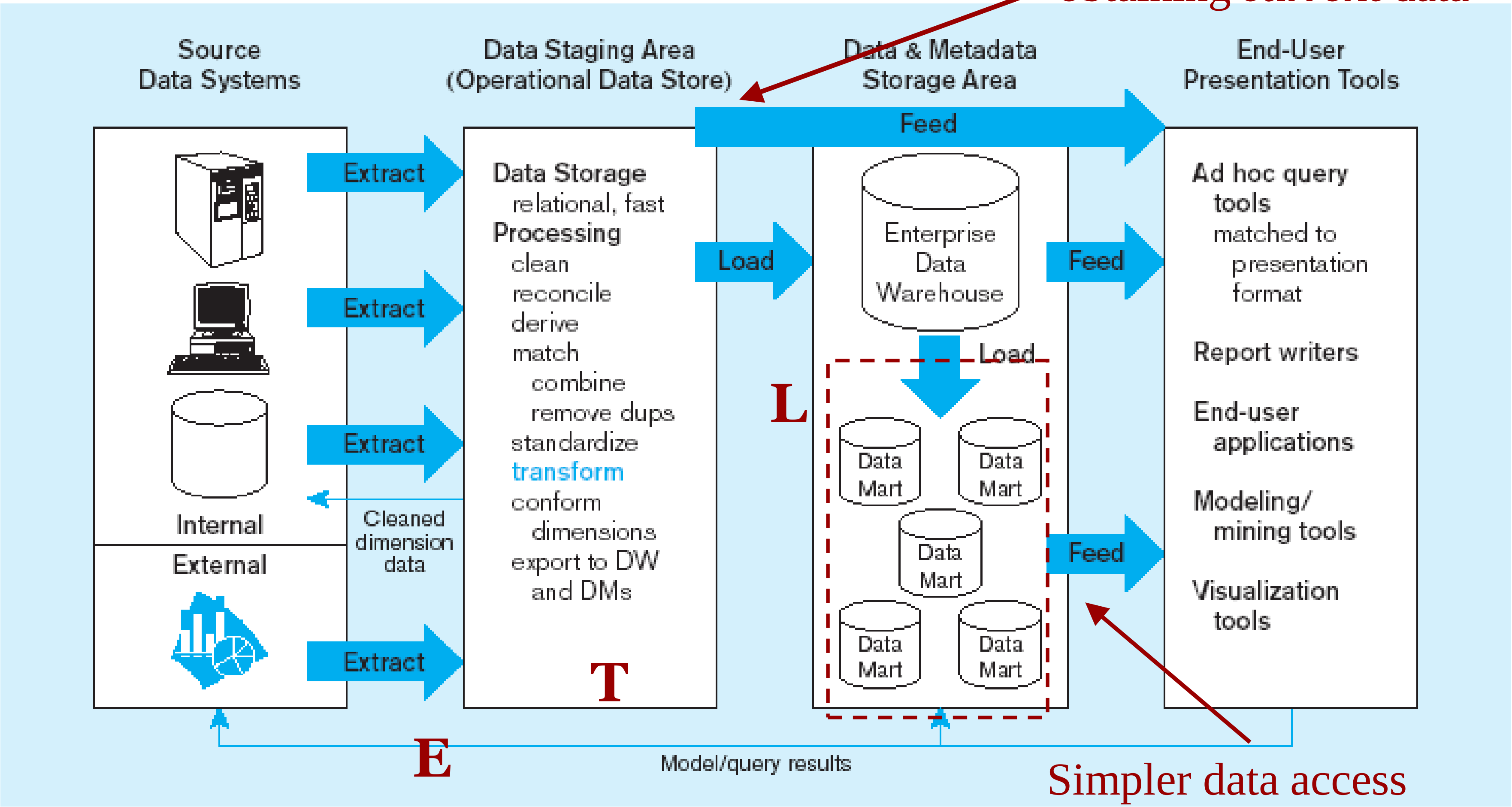


Separate ETL for each ***independent*** data mart

Data access complexity due to ***multiple*** data marts

Dependent data mart with operational data store: a three-level architecture

ODS provides option for obtaining ***current*** data



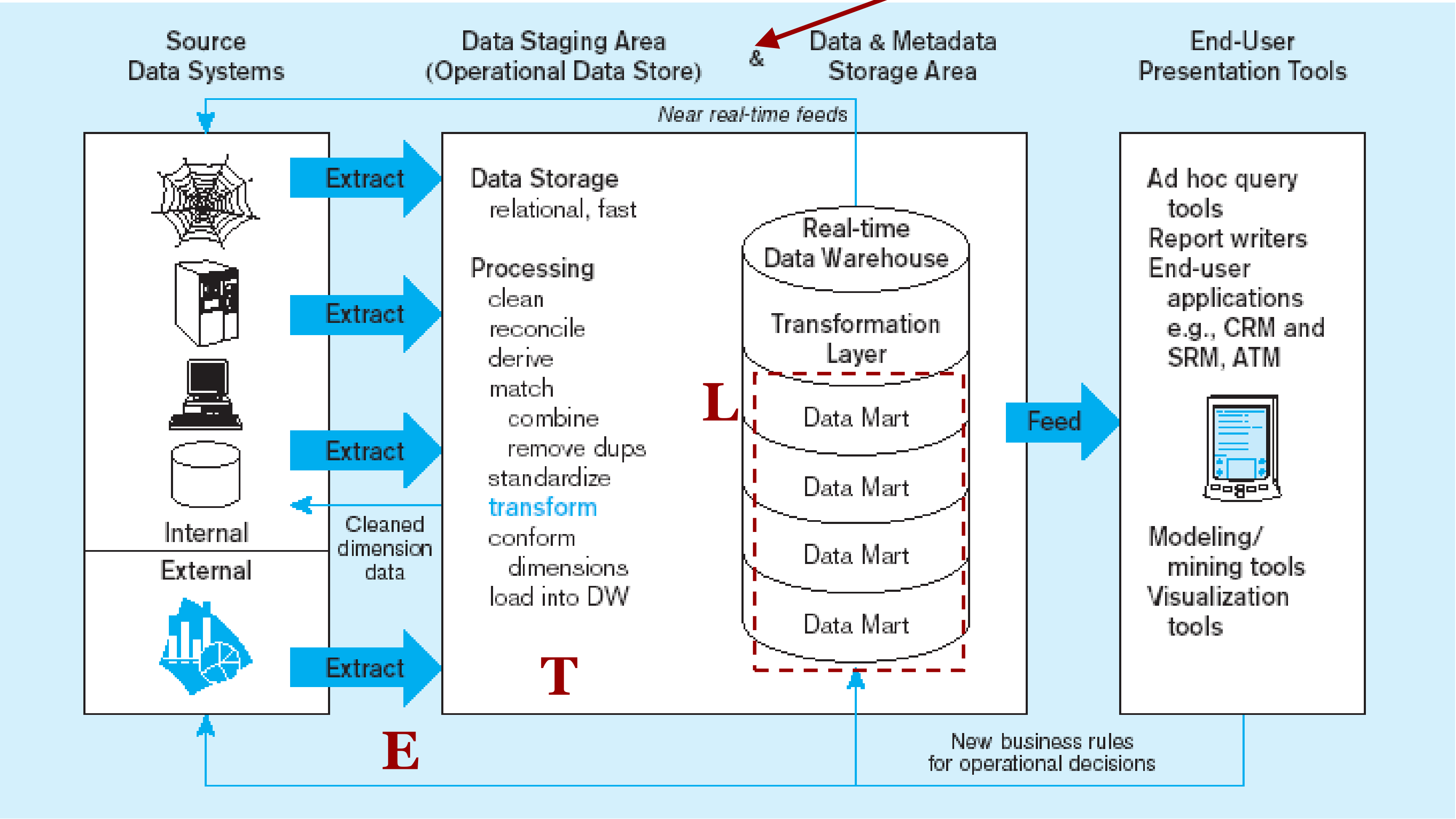
Simpler data access

Single ETL for ***enterprise data warehouse (EDW)***

Dependent data marts loaded from EDW

Logical data mart and real time warehouse architecture

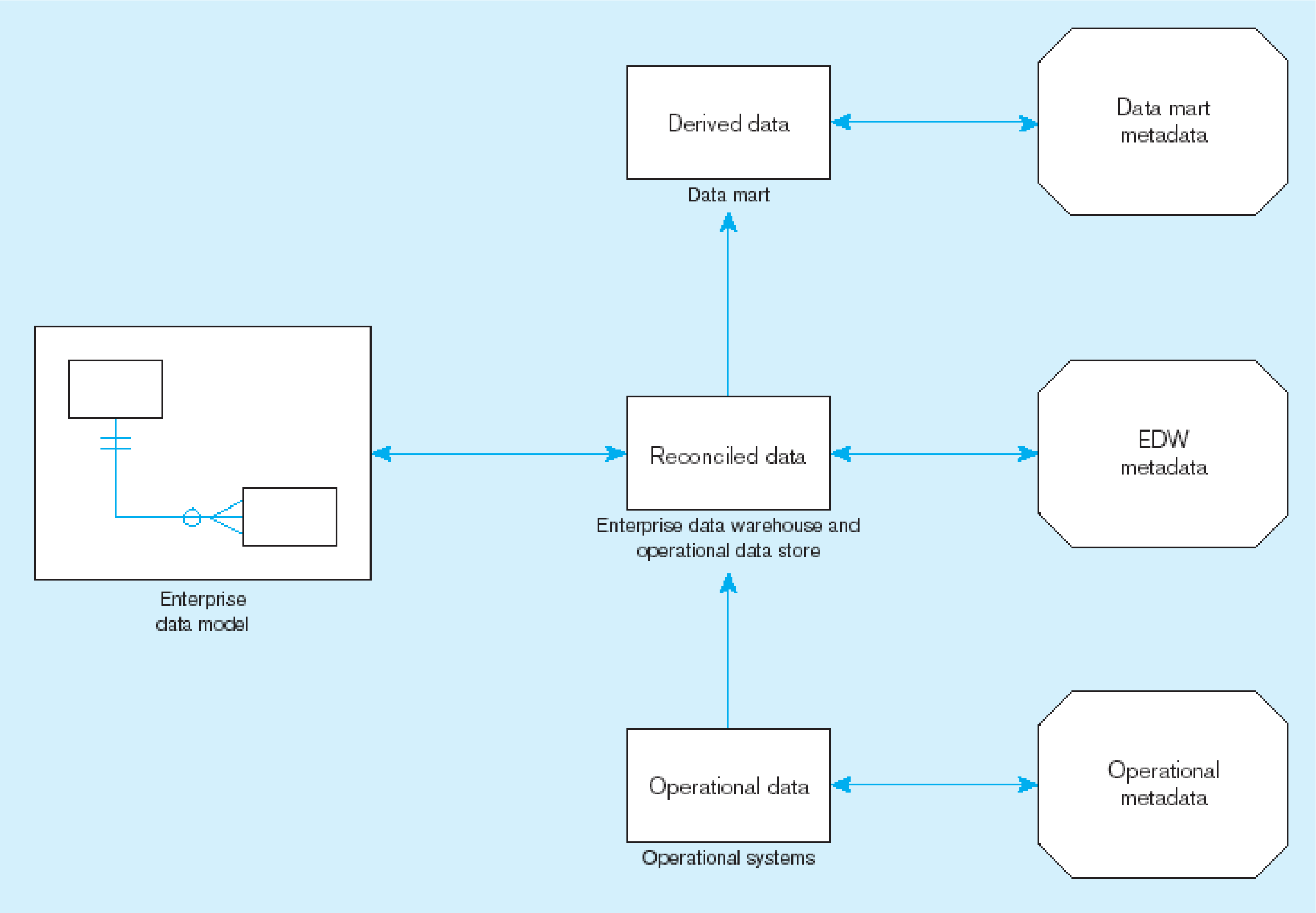
ODS and **data warehouse** are one and the same



Near real-time ETL for
Data Warehouse

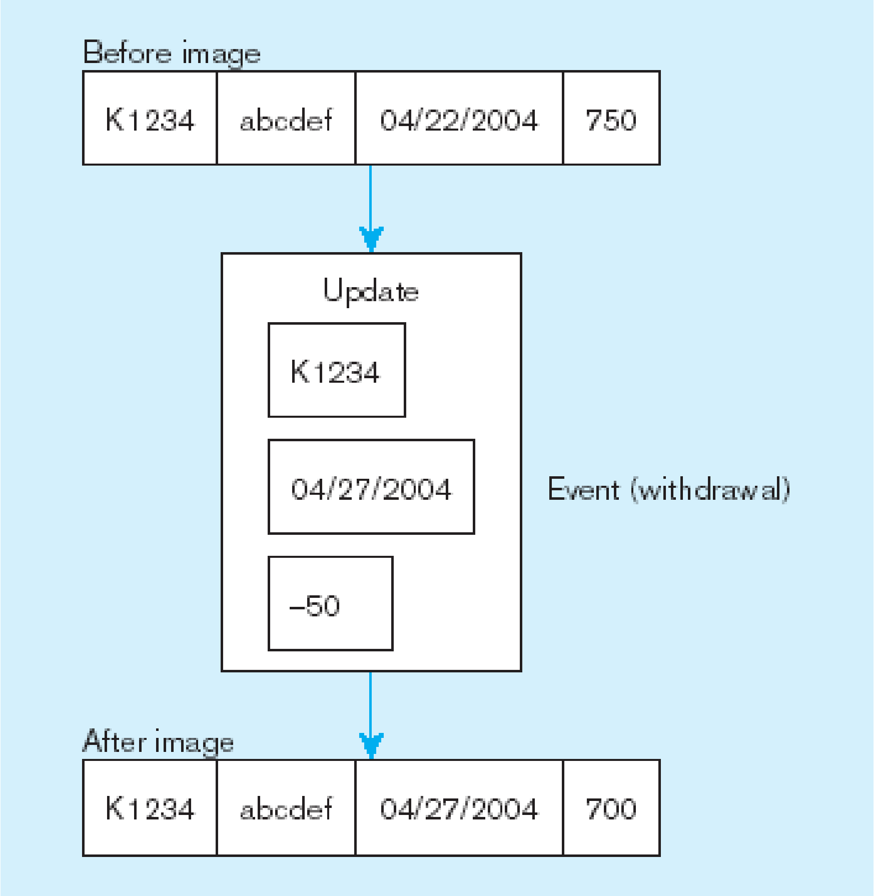
Data marts are NOT separate databases, but logical ***views*** of the data warehouse
➔ Easier to create new data marts

Three-layer data architecture for a data warehouse



Data Characteristics Status vs. Event Data

Example of DBMS log entry



Status

Event = a database action (create/update/delete) that results from a transaction

Status

Transient
operational data

Data Characteristics

Transient vs. Periodic Data

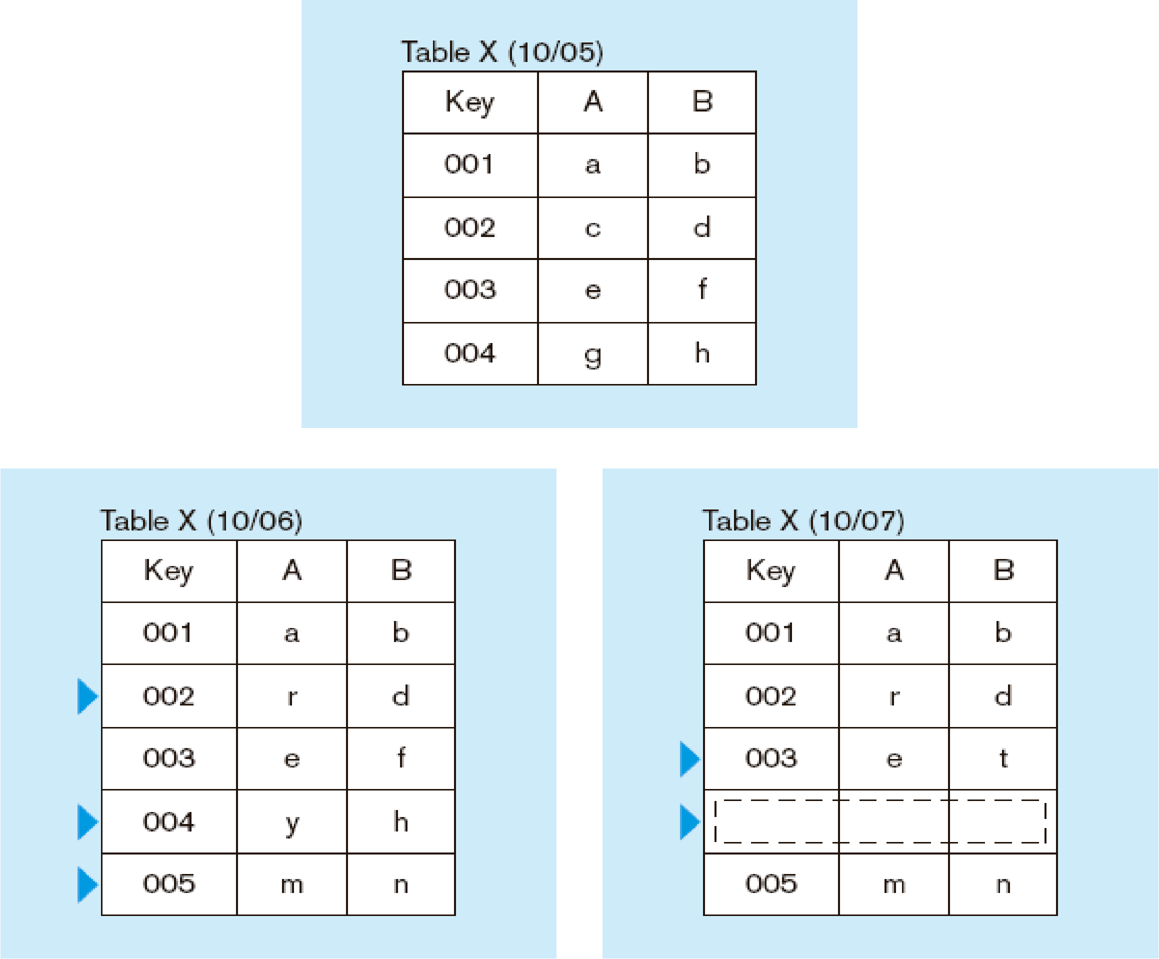


Table X (10/05)

Key	A	B
001	a	b
002	c	d
003	e	f
004	g	h

Table X (10/06)

	Key	A	B
	001	a	b
▶	002	r	d
	003	e	f
▶	004	y	h
▶	005	m	n

Table X (10/07)

	Key	A	B
	001	a	b
	002	r	d
▶	003	e	t
▶			
	005	m	n

With transient data, changes to existing records are written over previous records, thus destroying the previous data content

Periodic warehouse data

Data Characteristics

Transient vs. Periodic Data

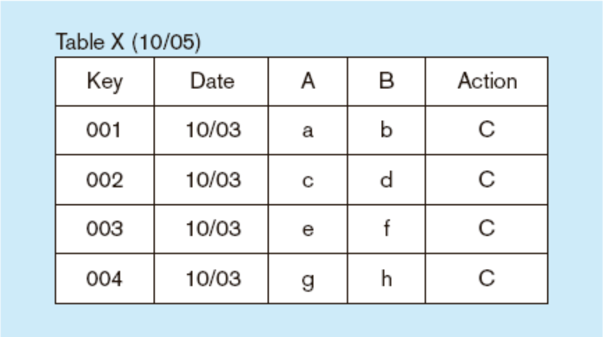


Table X (10/05)

Key	Date	A	B	Action
001	10/03	a	b	C
002	10/03	c	d	C
003	10/03	e	f	C
004	10/03	g	h	C

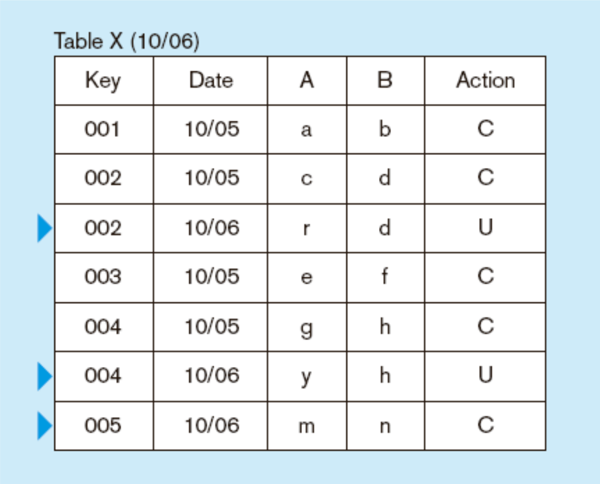


Table X (10/06)

	Key	Date	A	B	Action
	001	10/05	a	b	C
	002	10/05	c	d	C
▶	002	10/06	r	d	U
	003	10/05	e	f	C
	004	10/05	g	h	C
▶	004	10/06	y	h	U
▶	005	10/06	m	n	C

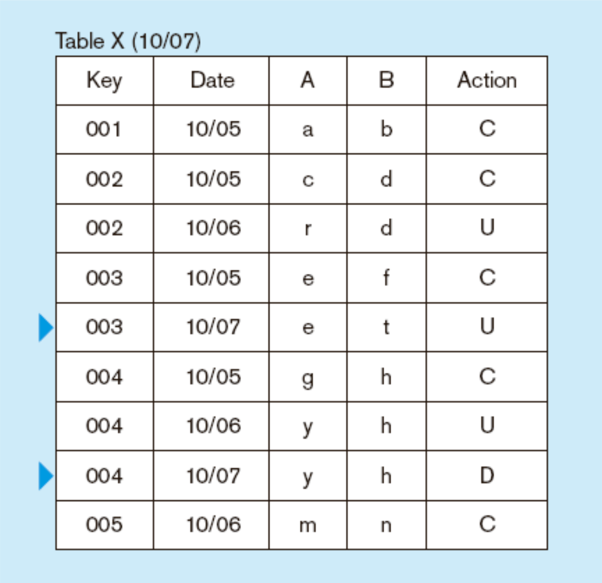


Table X (10/07)

	Key	Date	A	B	Action
	001	10/05	a	b	C
	002	10/05	c	d	C
	002	10/06	r	d	U
	003	10/05	e	f	C
▶	003	10/07	e	t	U
	004	10/05	g	h	C
	004	10/06	y	h	U
▶	004	10/07	y	h	D
	005	10/06	m	n	C

Periodic data are never physically altered or deleted once they have been added to the store

Other Data Warehouse Changes

- New descriptive attributes.
- New business activity attributes.
- New classes of descriptive attributes.
- Descriptive attributes become more refined.
- Descriptive data are related to one another.
- New source of data.

The Reconciled Data Layer

- Typical operational data is:
  - Transient–not historical
  - Not normalized (perhaps due to denormalization for performance)
  - Restricted in scope–not comprehensive
  - Sometimes poor quality–inconsistencies and errors
- After ETL, data should be:
  - Detailed–not summarized yet
  - Historical–periodic
  - Normalized–3rd normal form or higher
  - Comprehensive–enterprise-wide perspective
  - Timely–data should be current enough to assist decision-making
  - Quality controlled–accurate with full integrity

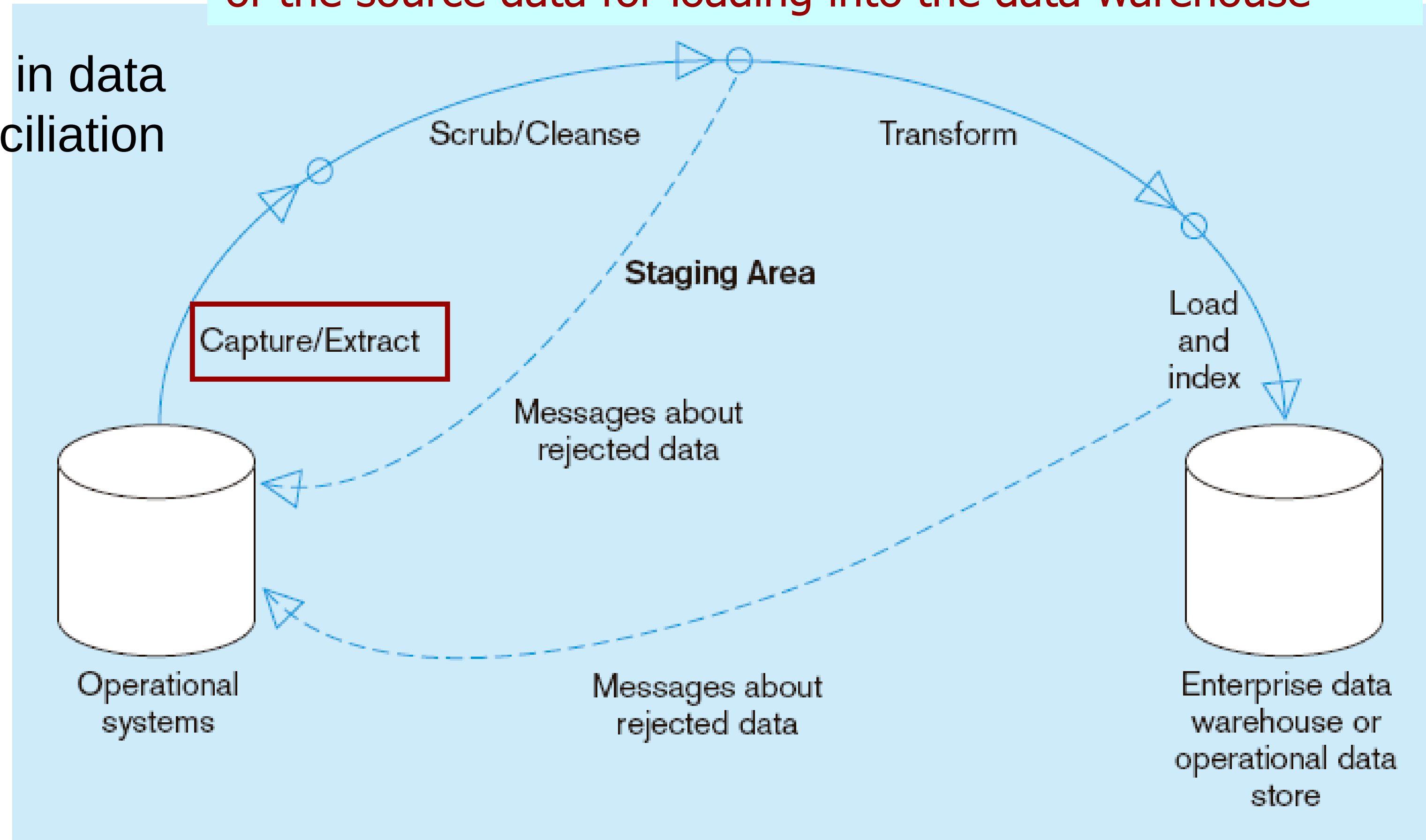
The ETL Process

- Capture/Extract
- Scrub or data cleansing
- Transform
- Load and Index

ETL = Extract, transform, and load

Capture/Extract…obtaining a snapshot of a chosen subset of the source data for loading into the data warehouse

Steps in data reconciliation

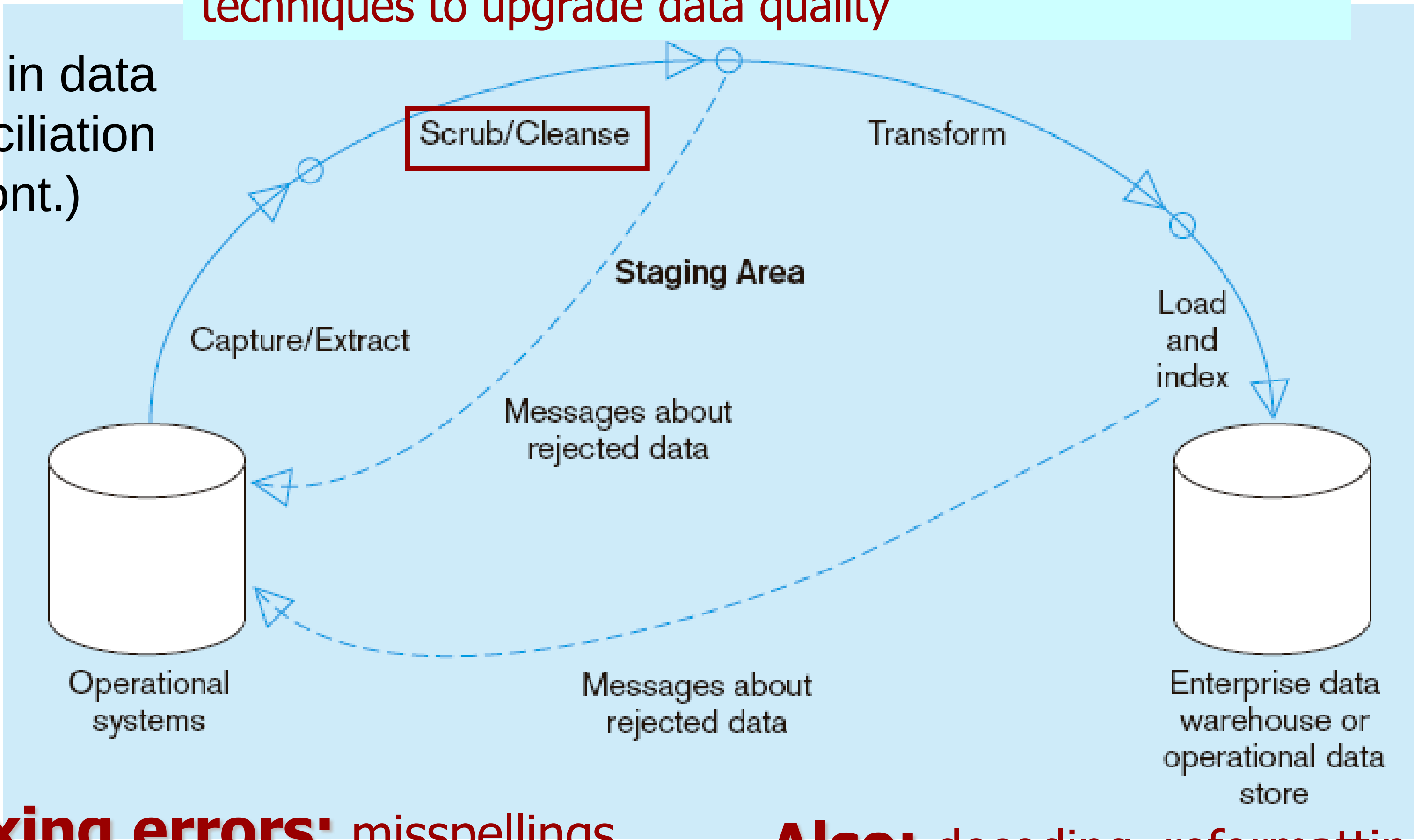

Scrub/Cleanse
Transform
Staging Area
Capture/Extract
Load and index
Messages about rejected data
Operational systems
Messages about rejected data
Enterprise data warehouse or operational data store

Static extract = capturing a snapshot of the source data at a point in time

Incremental extract = capturing changes that have occurred since the last static extract

Scrub/Cleanse…uses pattern recognition and AI techniques to upgrade data quality

Steps in data reconciliation (cont.)

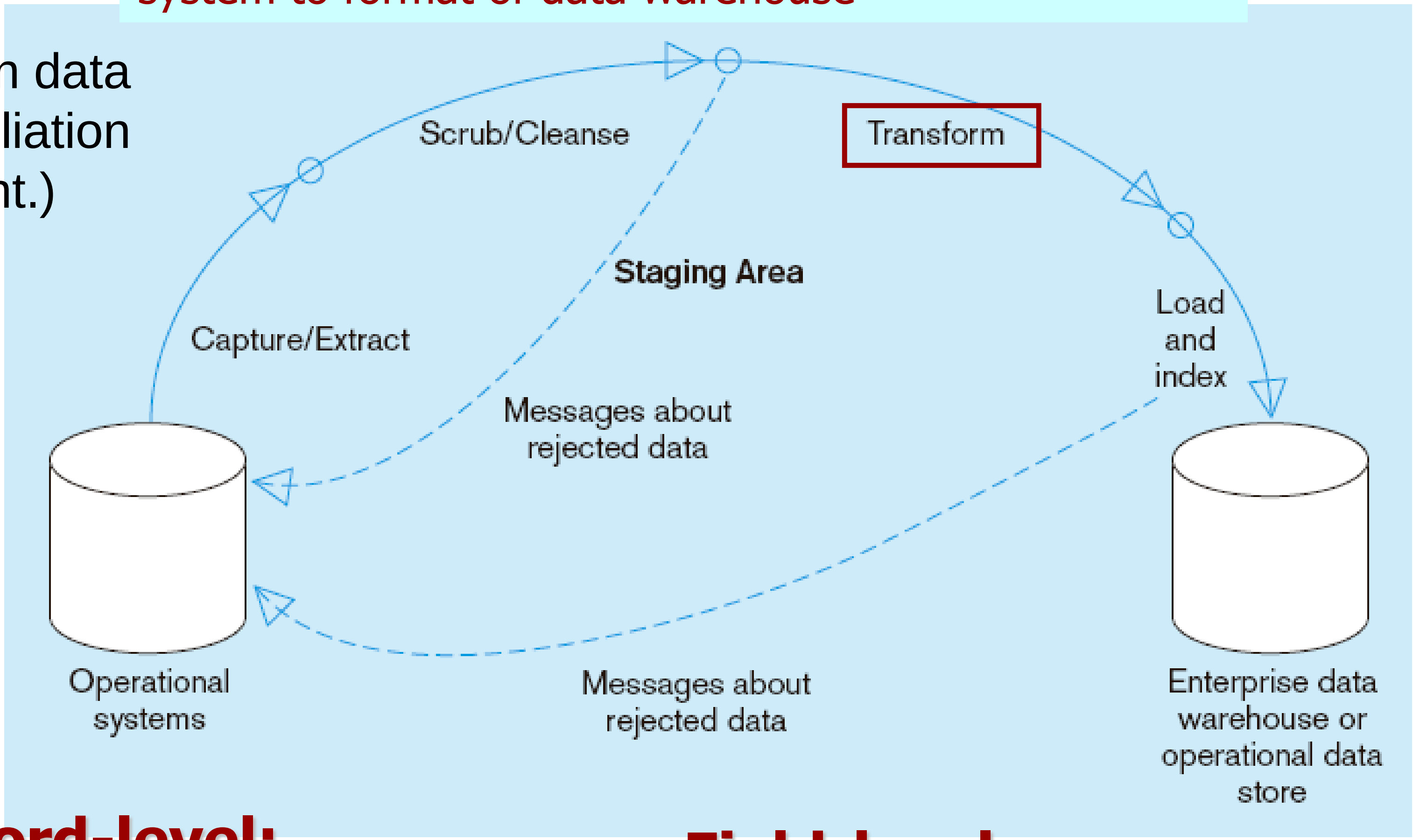


Fixing errors: misspellings, erroneous dates, incorrect field usage, mismatched addresses, missing data, duplicate data, inconsistencies

Also: decoding, reformatting, time stamping, conversion, key generation, merging, error detection/logging, locating missing data

Transform = convert data from format of operational system to format of data warehouse

Steps in data reconciliation (cont.)


Scrub/Cleanse
Transform
Staging Area
Capture/Extract
Load and index
Messages about rejected data
Operational systems
Messages about rejected data
Enterprise data warehouse or operational data store

Record-level:

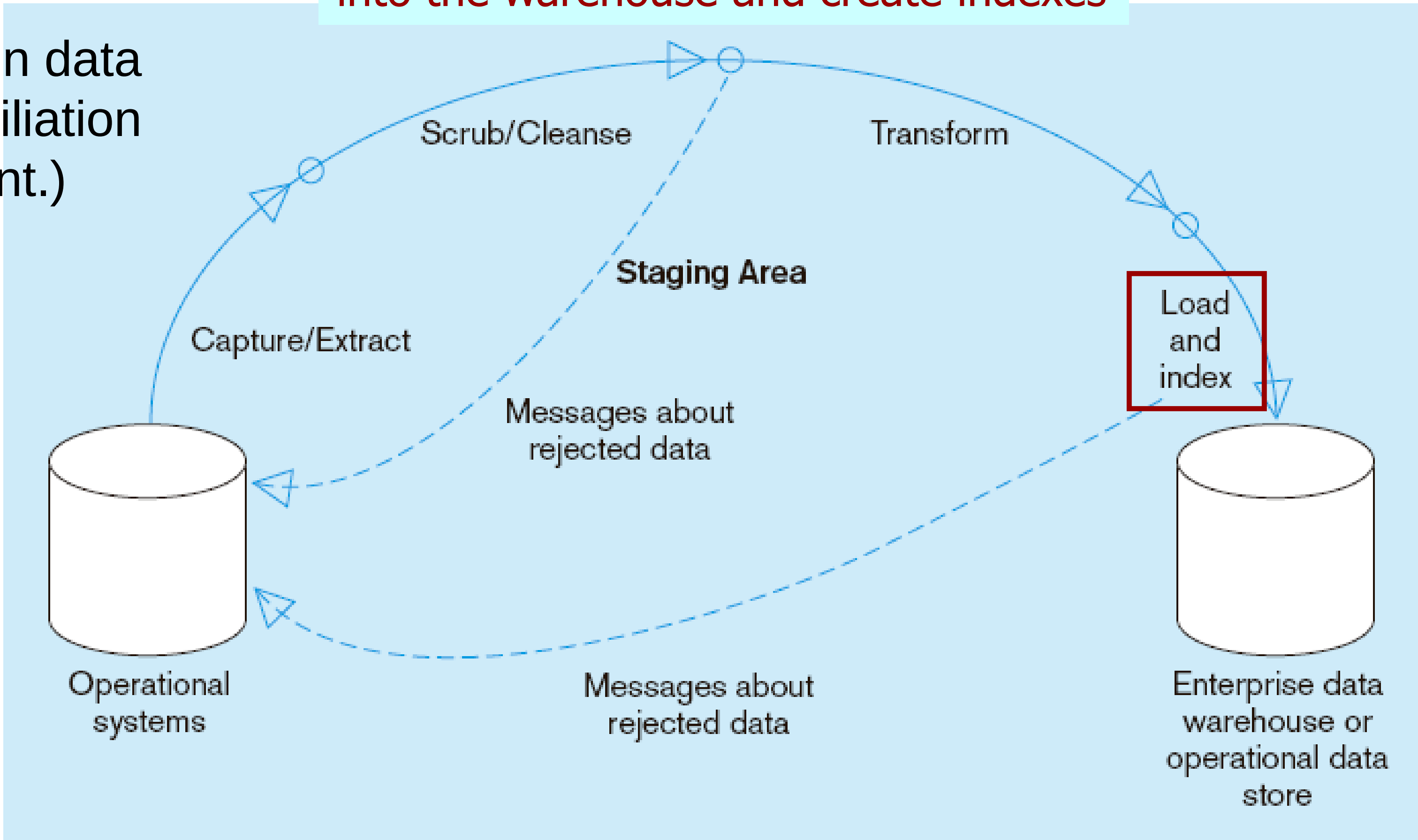
Selection–data partitioning
Joining–data combining
Aggregation–data summarization

Field-level:

single-field–from one field to one field
multi-field–from many fields to one, or one field to many

Steps in data reconciliation (cont.)

Load/Index= place transformed data into the warehouse and create indexes

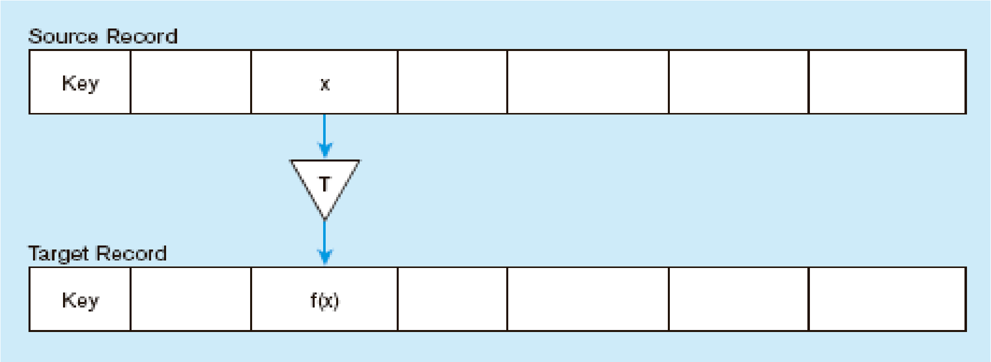


Scrub/Cleanse
Transform
Staging Area
Capture/Extract
Load and index
Messages about rejected data
Operational systems
Messages about rejected data
Enterprise data warehouse or operational data store

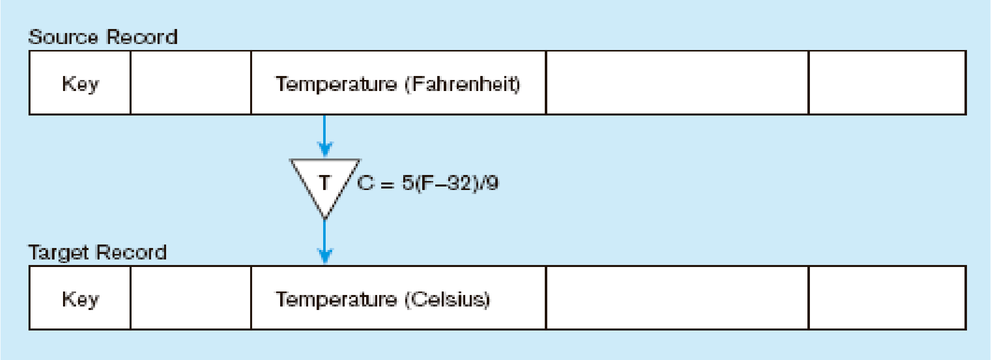
Refresh mode: bulk rewriting of target data at periodic intervals

Update mode: only changes in source data are written to data warehouse

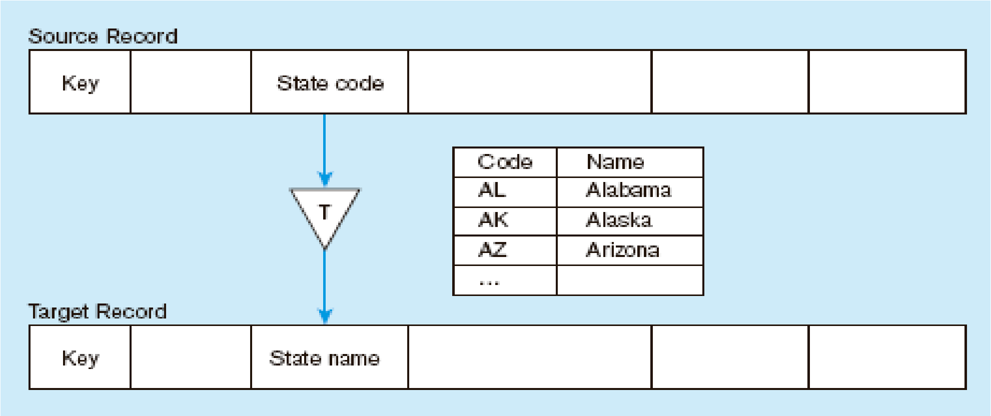
Single-field transformation



In general–some transformation function translates data from old form to new form



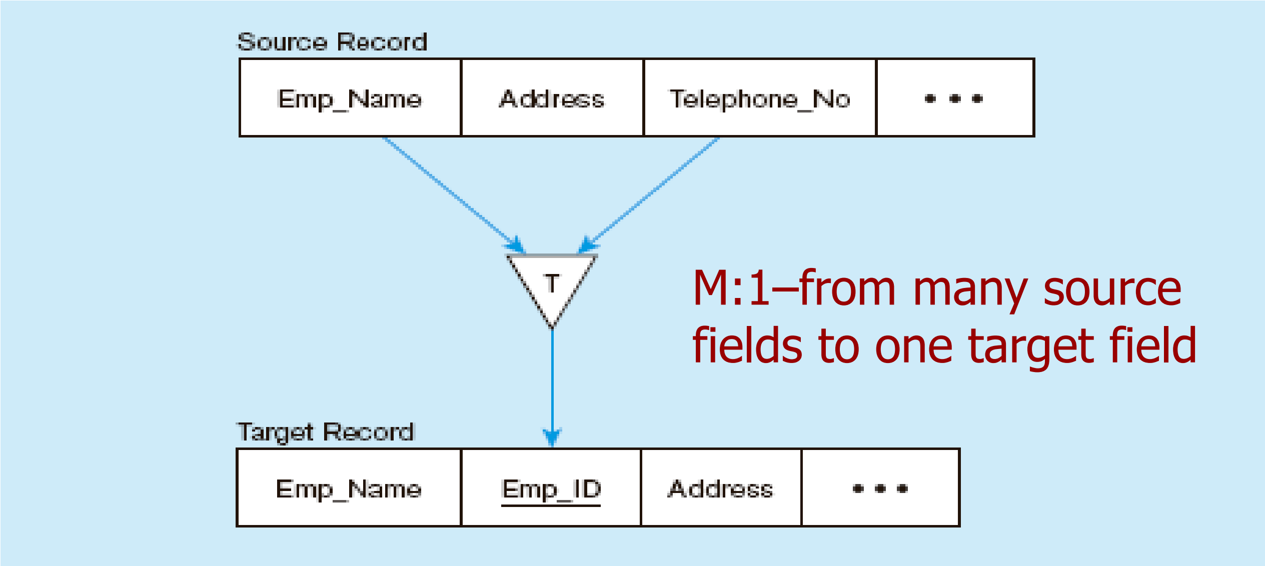
Algorithmic transformation uses a formula or logical expression



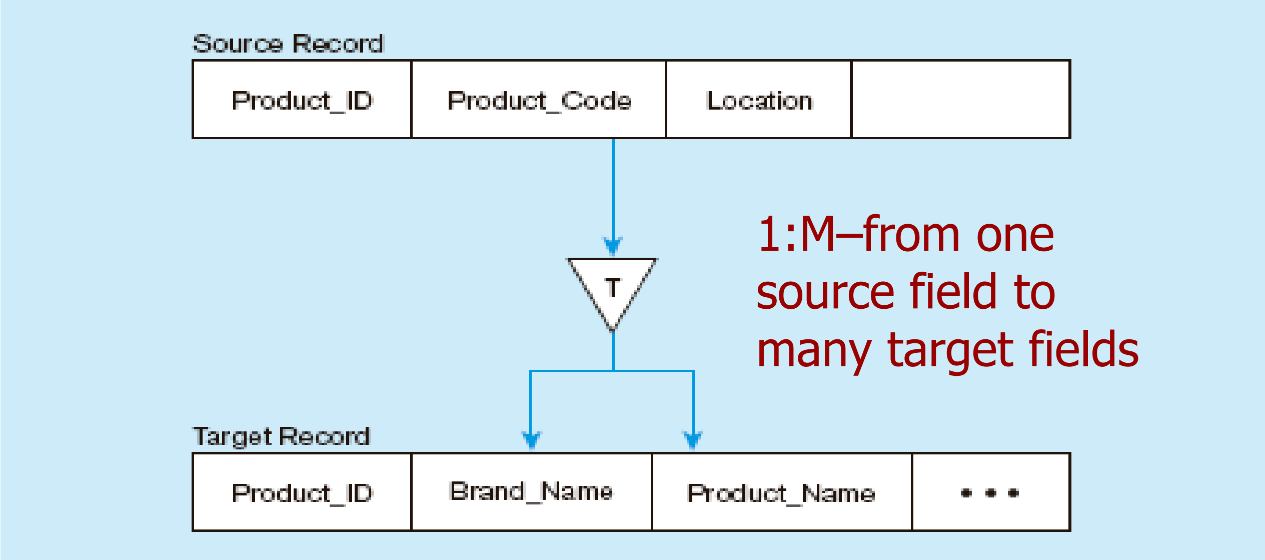
Code	Name
AL	Alabama
AK	Alaska
AZ	Arizona
…	

Table lookup–another approach, uses a separate table keyed by source record code

Multi-field transformation



M:1–from many source fields to one target field



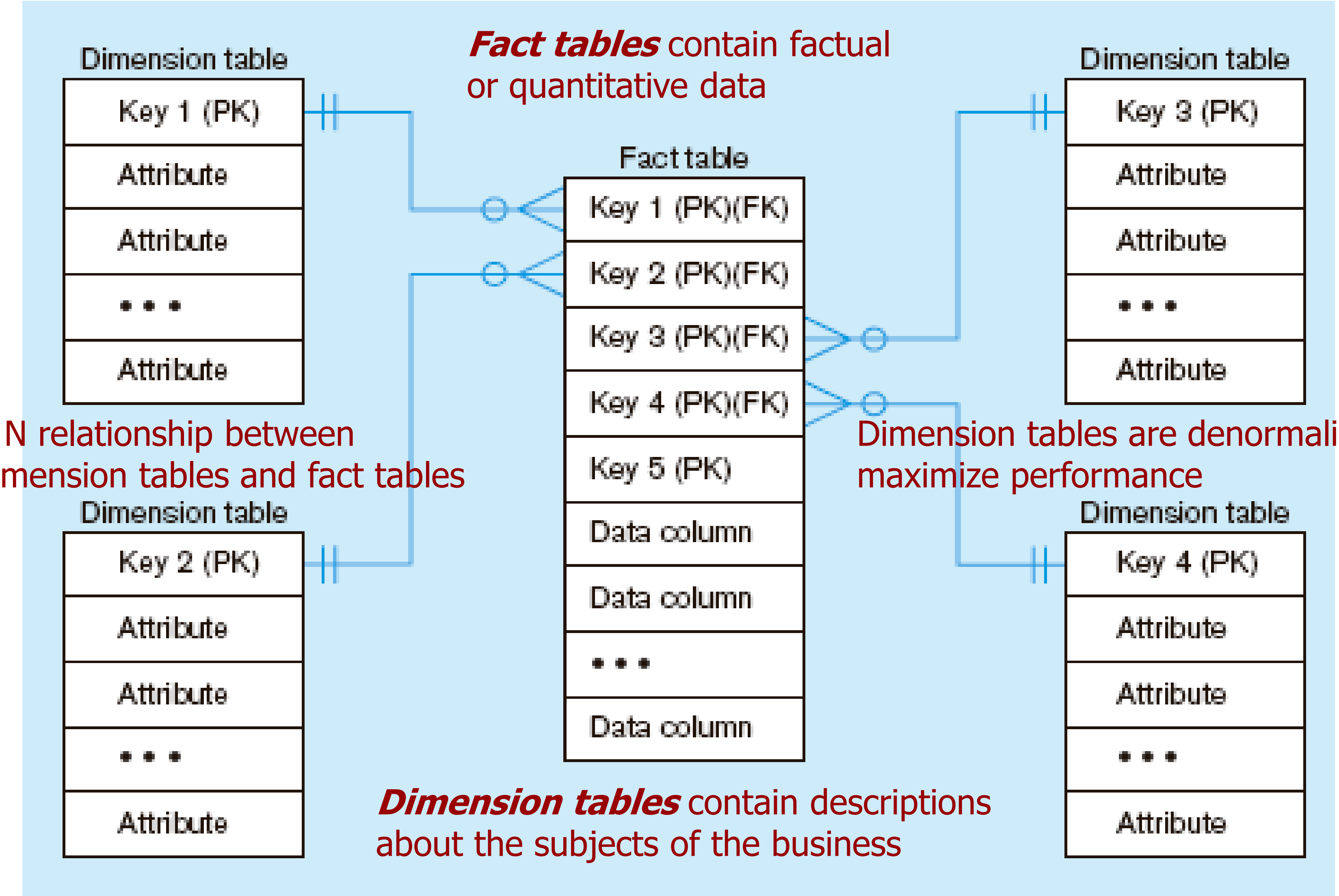
1:M–from one source field to many target fields

Derived Data

- Objectives
  - Ease of use for decision support applications
  - Fast response to predefined user queries
  - Customized data for particular target audiences
  - Ad-hoc query support
  - Data mining capabilities
- Characteristics
  - Detailed (mostly periodic) data
  - Aggregate (for summary)
  - Distributed (to departmental servers)

Most common data model = **star schema**
(also called “dimensional model”)

Components of a **star schema**



Fact tables contain factual or quantitative data

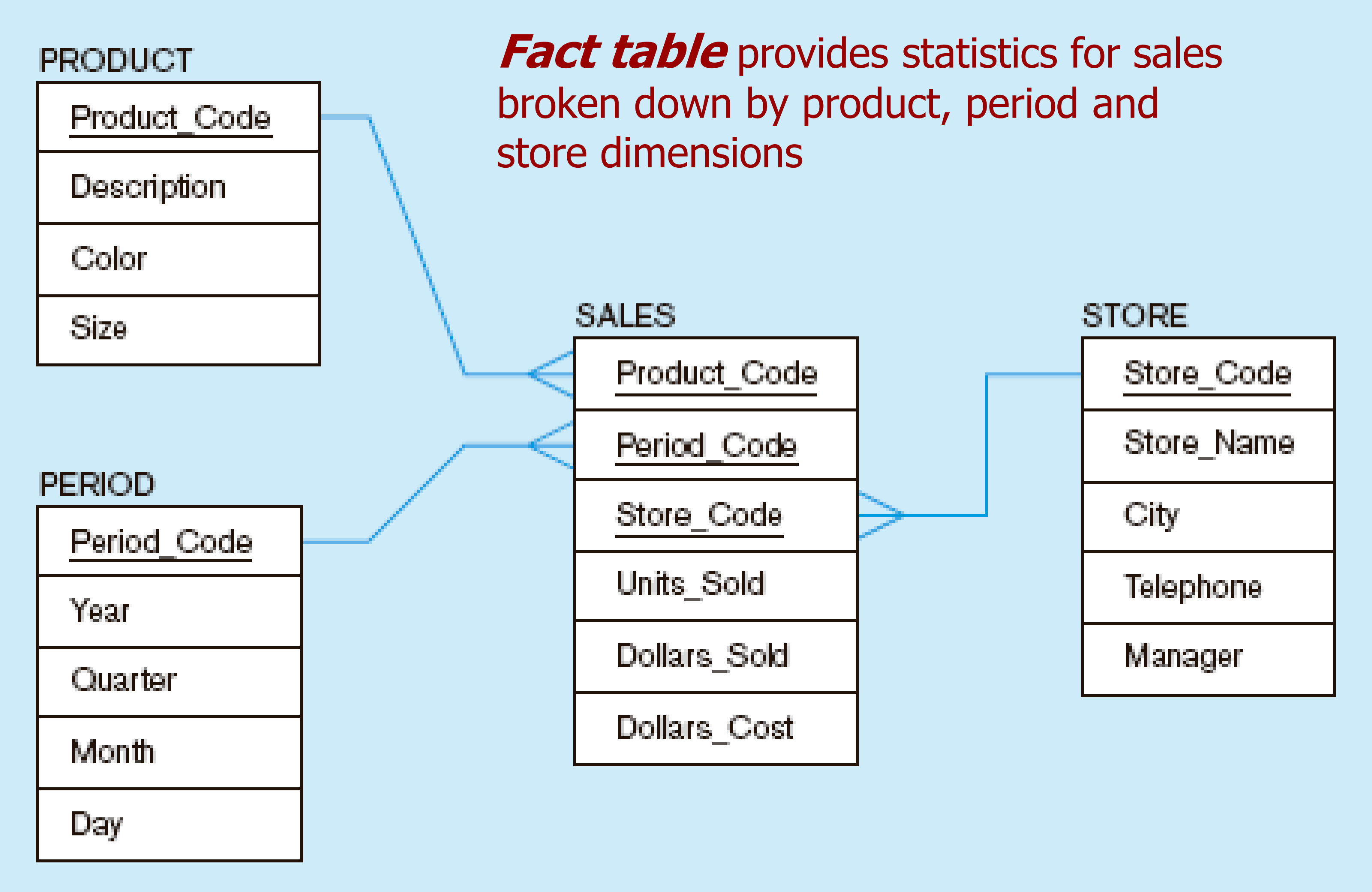
1:N relationship between dimension tables and fact tables

Dimension tables are denormalized to maximize performance

Dimension tables contain descriptions about the subjects of the business

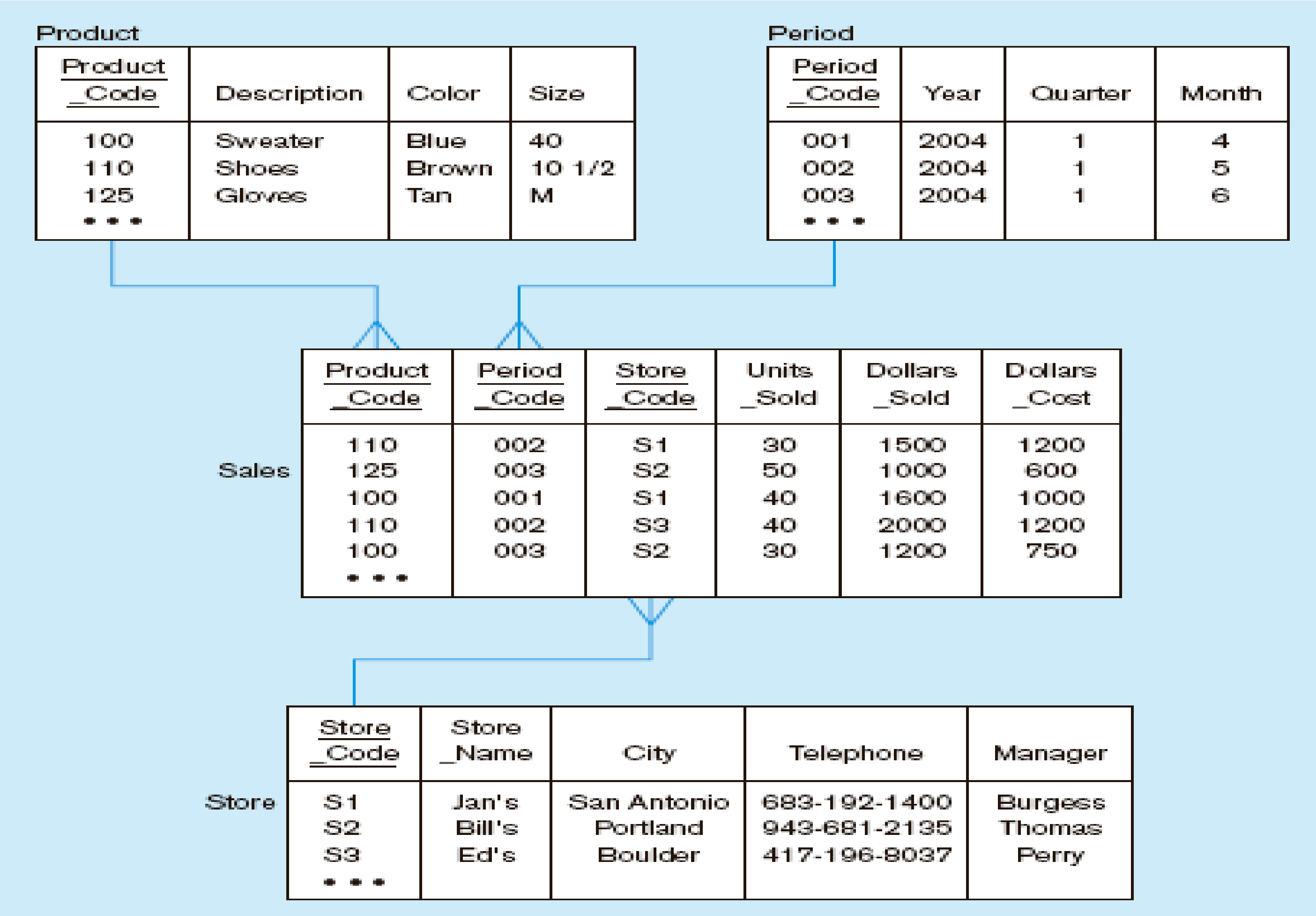
Excellent for ad-hoc queries, but bad for online transaction processing

Star schema example



Fact table provides statistics for sales broken down by product, period and store dimensions

Star schema with sample data



Product

Product_Code	Description	Color	Size
100	Sweater	Blue	40
110	Shoes	Brown	10 1/2
125	Gloves	Tan	M
• • •			

Period

Period_Code	Year	Quarter	Month
001	2004	1	4
002	2004	1	5
003	2004	1	6
• • •			

Sales

Product_Code	Period_Code	Store_Code	Units_Sold	Dollars_Sold	Dollars_Cost
110	002	S1	30	1500	1200
125	003	S2	50	1000	600
100	001	S1	40	1600	1000
110	002	S3	40	2000	1200
100	003	S2	30	1200	750
• • •					

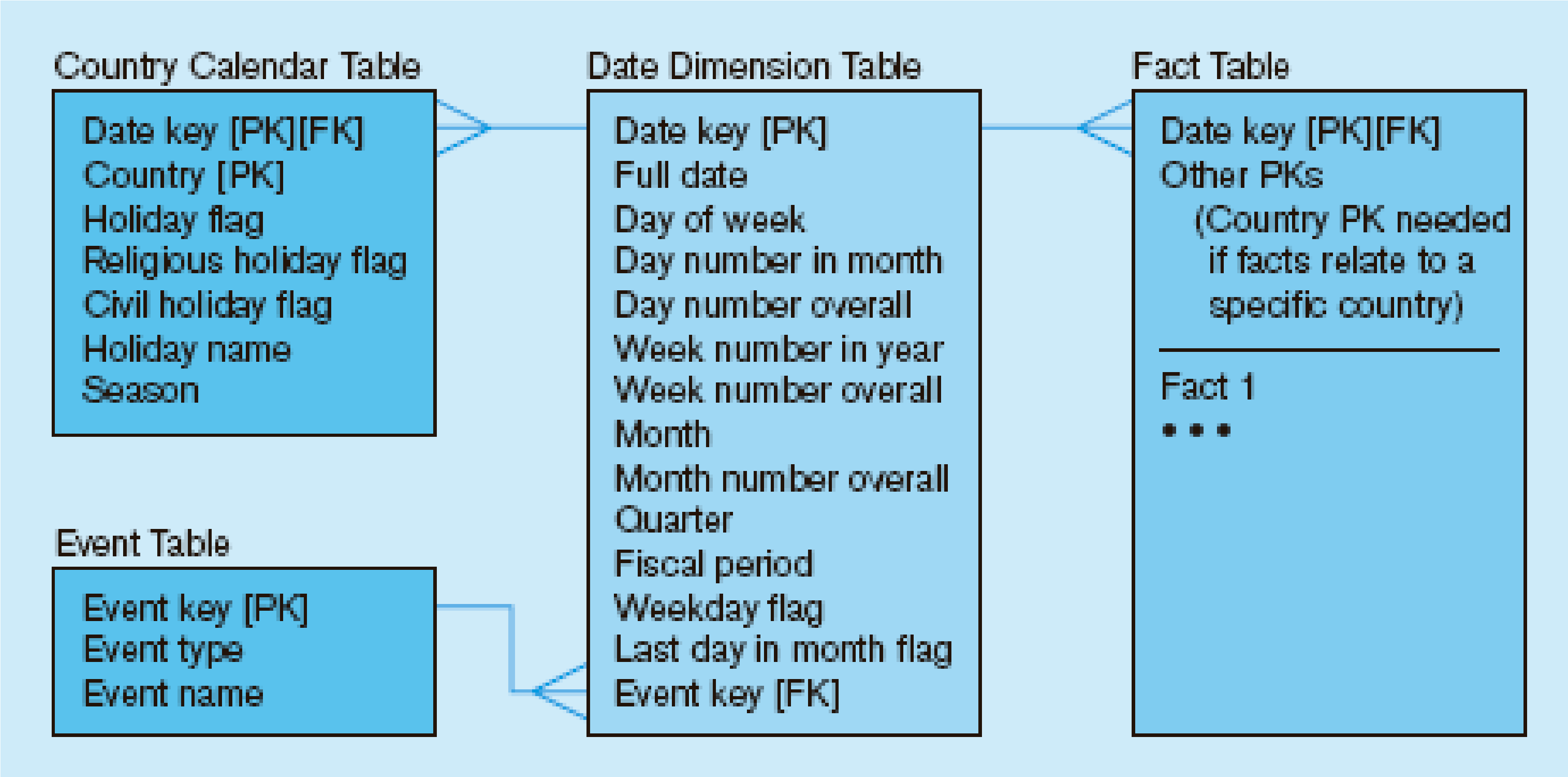
Store

Store_Code	Store_Name	City	Telephone	Manager
S1	Jan's	San Antonio	683-192-1400	Burgess
S2	Bill's	Portland	943-681-2135	Thomas
S3	Ed's	Boulder	417-196-8037	Perry
• • •				

Issues Regarding Star Schema

- Dimension table keys must be ***surrogate*** (non-intelligent and non-business related), because:
  - Keys may change over time
  - Length/format consistency
- Granularity of Fact Table–what level of detail do you want?
  - Transactional grain–finest level
  - Aggregated grain–more summarized
  - Finer grains ➔ better ***market basket analysis*** capability
  - Finer grain ➔ more dimension tables, more rows in fact table
- Duration of the database–how much history should be kept?
  - Natural duration–13 months or 5 quarters
  - Financial institutions may need longer duration
  - Older data is more difficult to source and cleanse

Modeling dates



Country Calendar Table

- Date key [PK][FK]
- Country [PK]
- Holiday flag
- Religious holiday flag
- Civil holiday flag
- Holiday name
- Season

Date Dimension Table

- Date key [PK]
- Full date
- Day of week
- Day number in month
- Day number overall
- Week number in year
- Week number overall
- Month
- Month number overall
- Quarter
- Fiscal period
- Weekday flag
- Last day in month flag
- Event key [FK]

Fact Table

- Date key [PK][FK]
- Other PKs
  (Country PK needed
  if facts relate to a
  specific country)
- Fact 1
- • • •

Event Table

- Event key [PK]
- Event type
- Event name

Fact tables contain time-period data
➔ Date dimensions are important

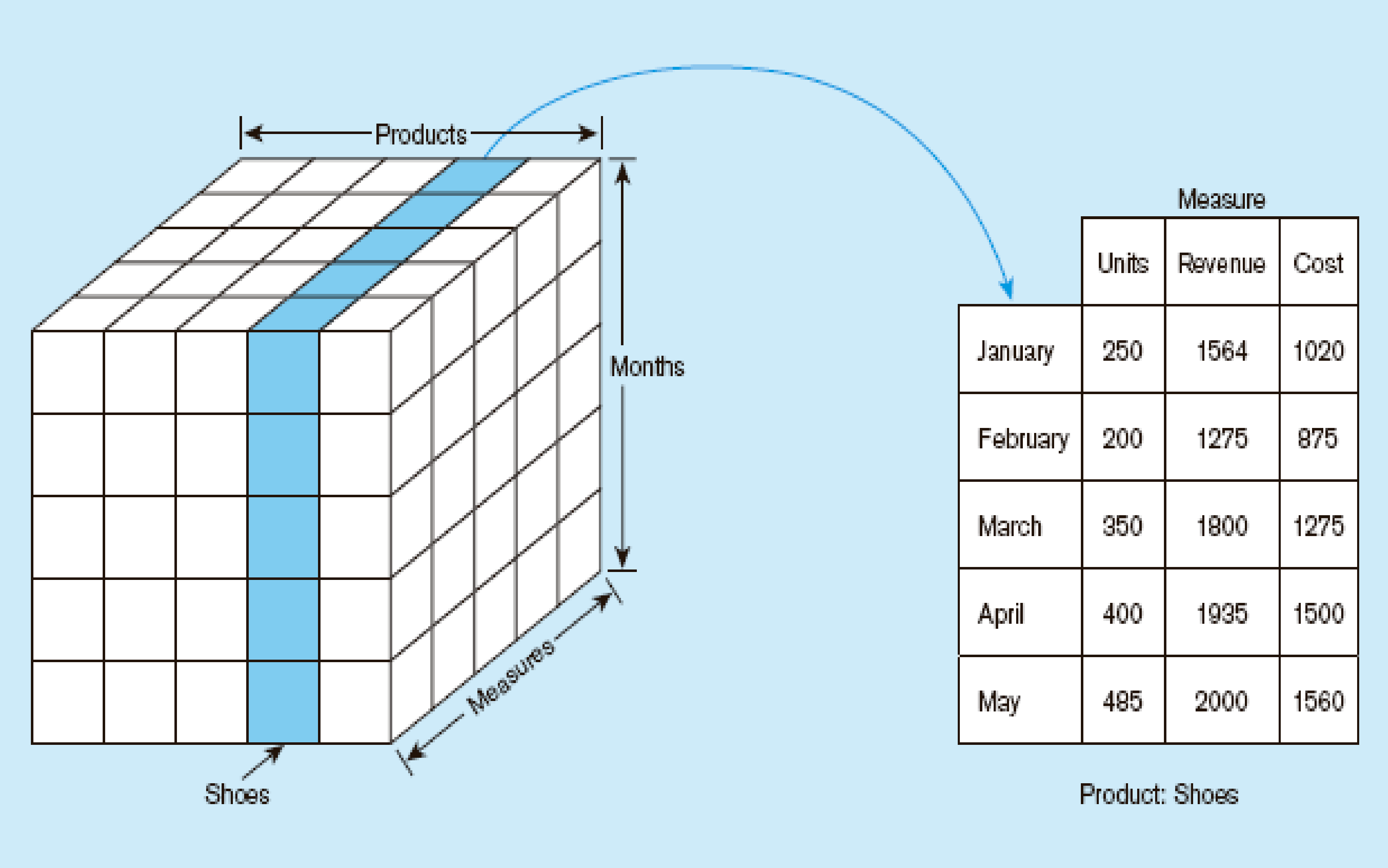
The User Interface
Metadata (data catalog)

- Identify subjects of the data mart
- Identify dimensions and facts
- Indicate how data is derived from enterprise data warehouses, including derivation rules
- Indicate how data is derived from operational data store, including derivation rules.
- Identify available reports and predefined queries.
- Identify data analysis techniques (e.g. drill-down).
- Identify responsible people.

On-Line Analytical Processing (OLAP) Tools

- The use of a set of graphical tools that provides users with multidimensional views of their data and allows them to analyze the data using simple windowing techniques
- ***Relational OLAP (ROLAP)***
  - Traditional relational representation
- ***Multidimensional OLAP (MOLAP)***
  - **Cube** structure
- OLAP Operations
  - ***Cube slicing***–come up with 2-D view of data
  - ***Drill-down***–going from summary to more detailed views

Slicing a data cube



Measure

	Units	Revenue	Cost
January	250	1564	1020
February	200	1275	875
March	350	1800	1275
April	400	1935	1500
May	485	2000	1560

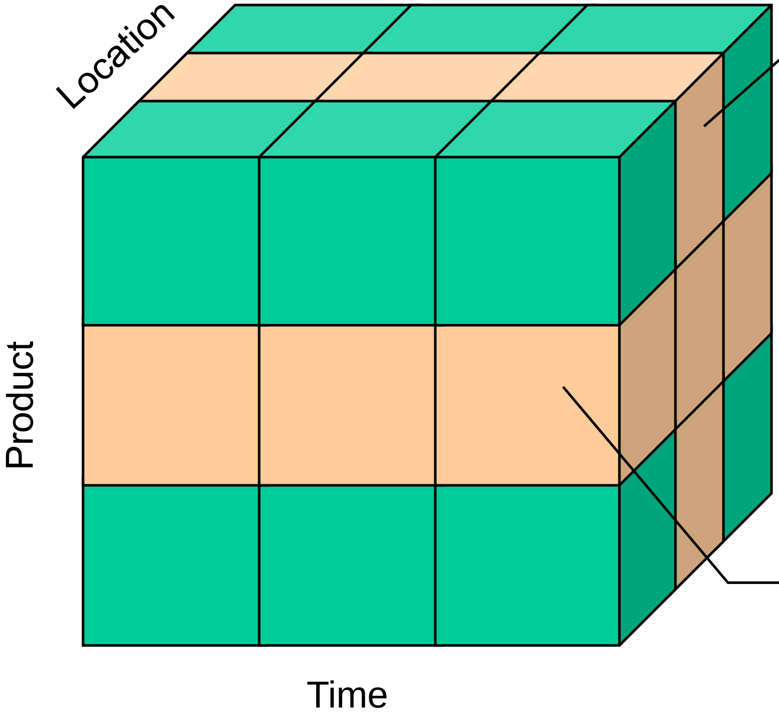
Product: Shoes

Three Dimensional View of Data

Location: possible attributes – region, state, city, store, etc.

Product: possible attributes – product type, id, brand, color, size.

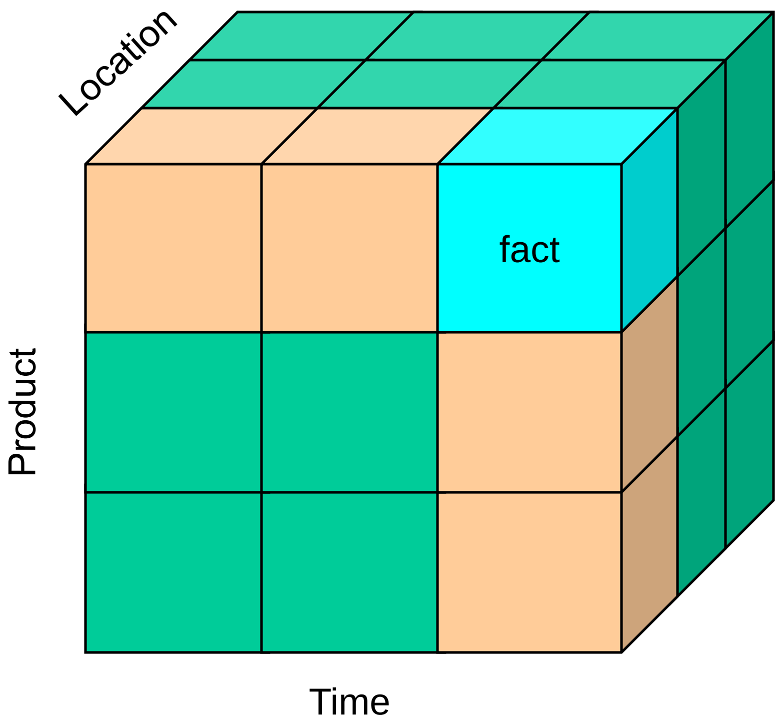
Time: possible attributes – year, quarter, month, week, day, time of day, etc.



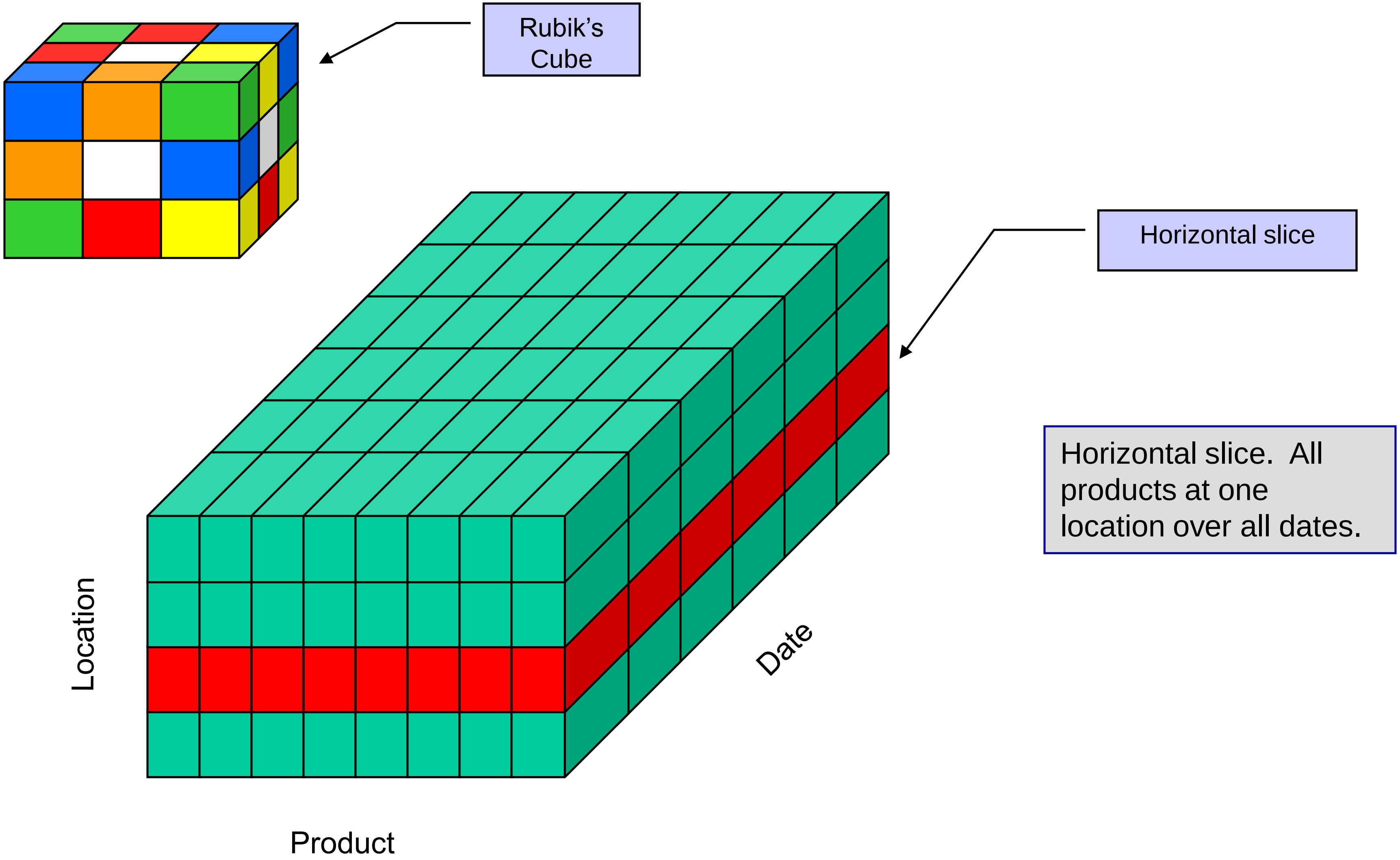
Sales manager's view of sales data

Product manager's view of sales data

Slice and Dice Operation

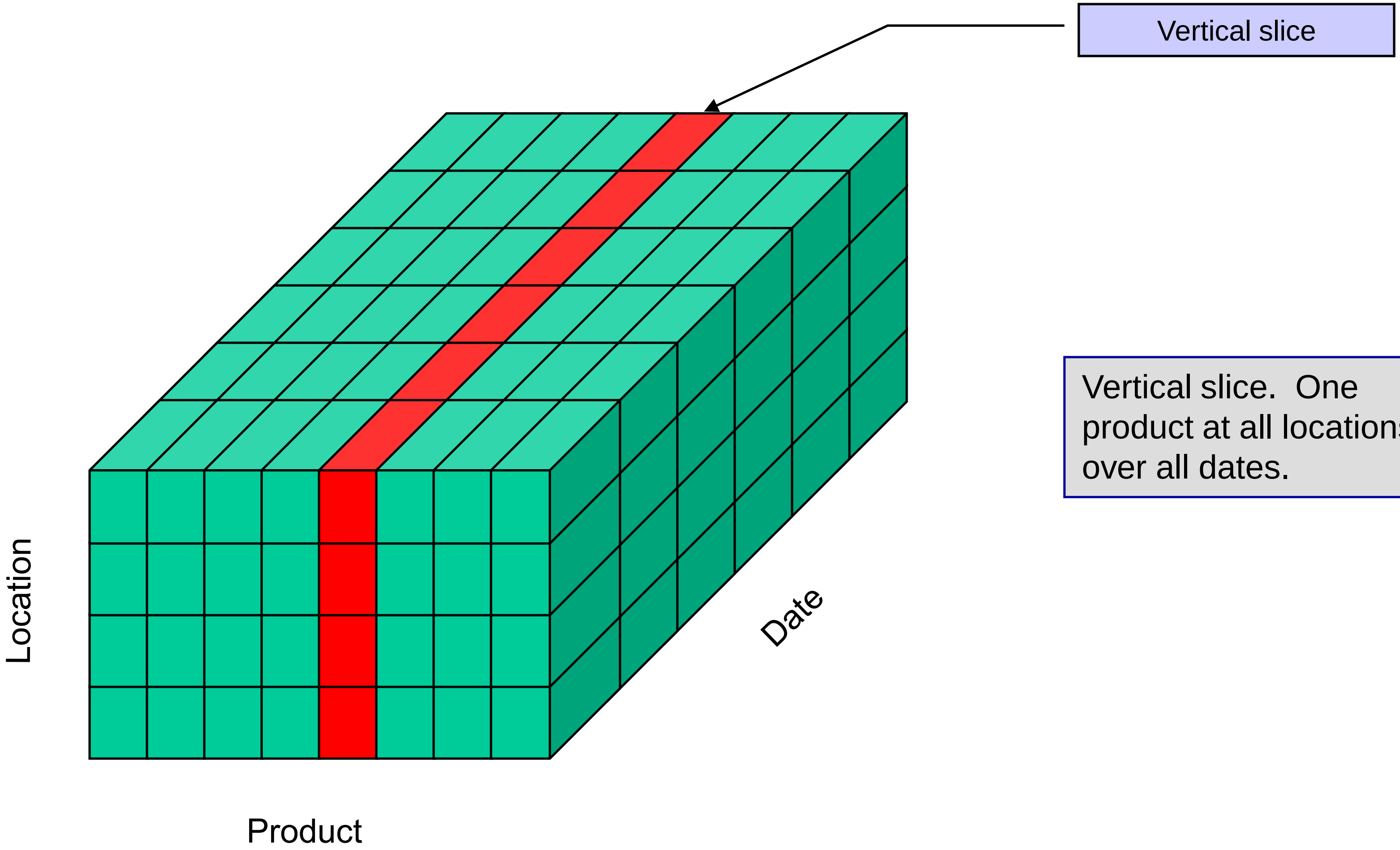


Three Dimensional View of Data



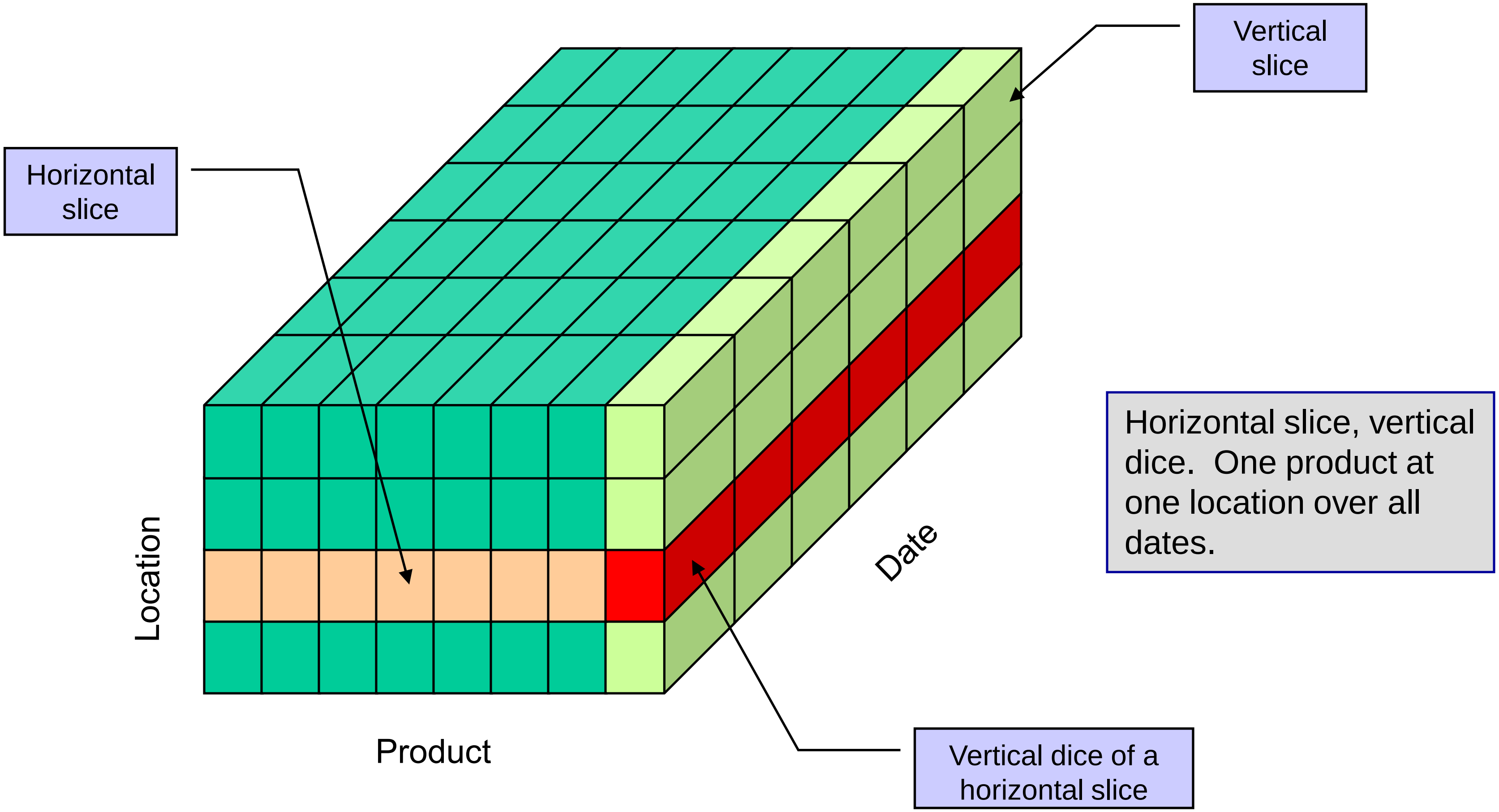
Horizontal slice. All products at one location over all dates.

Three Dimensional View of Data (cont.)



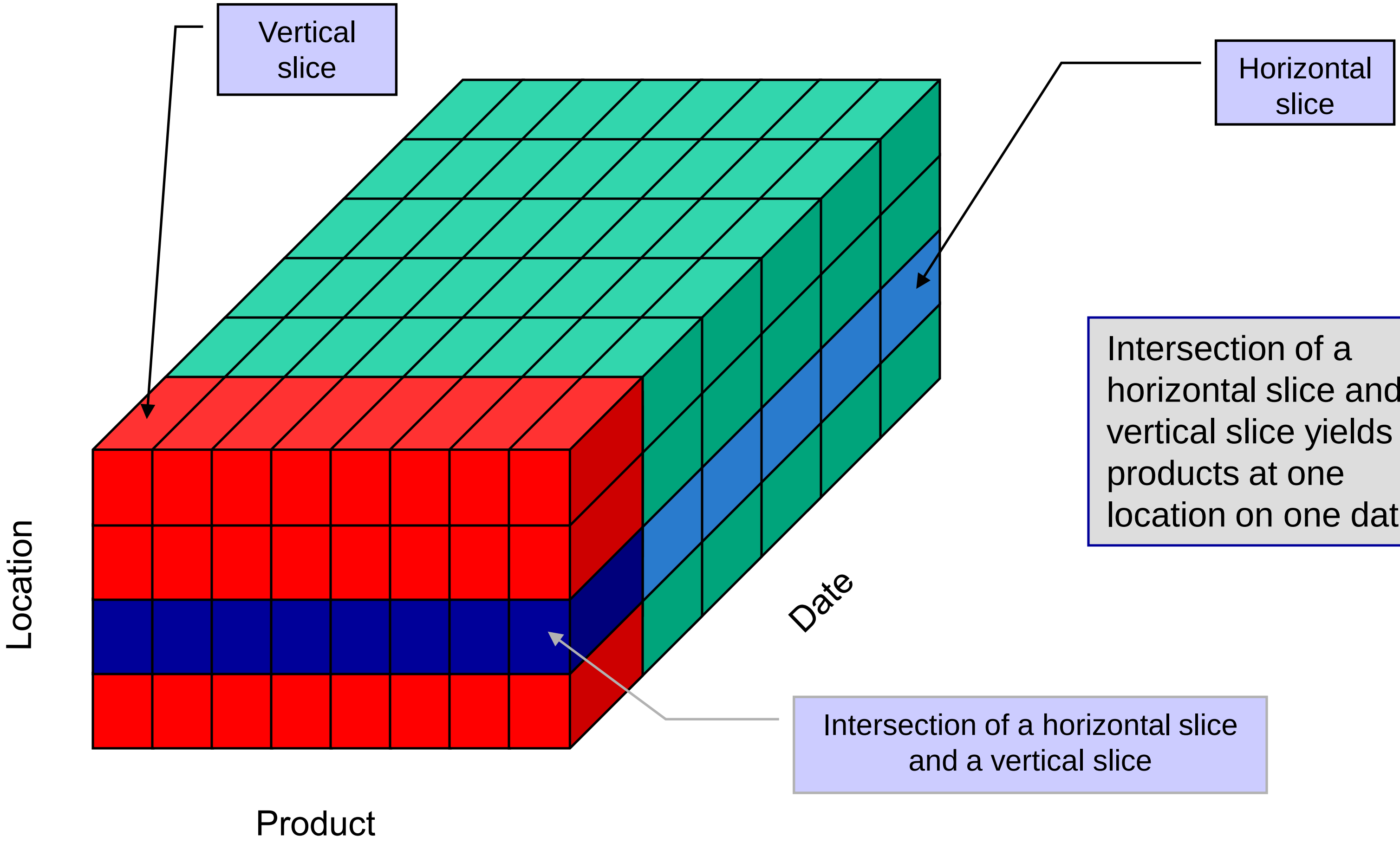
Vertical slice. One product at all locations over all dates.

Three Dimensional View of Data



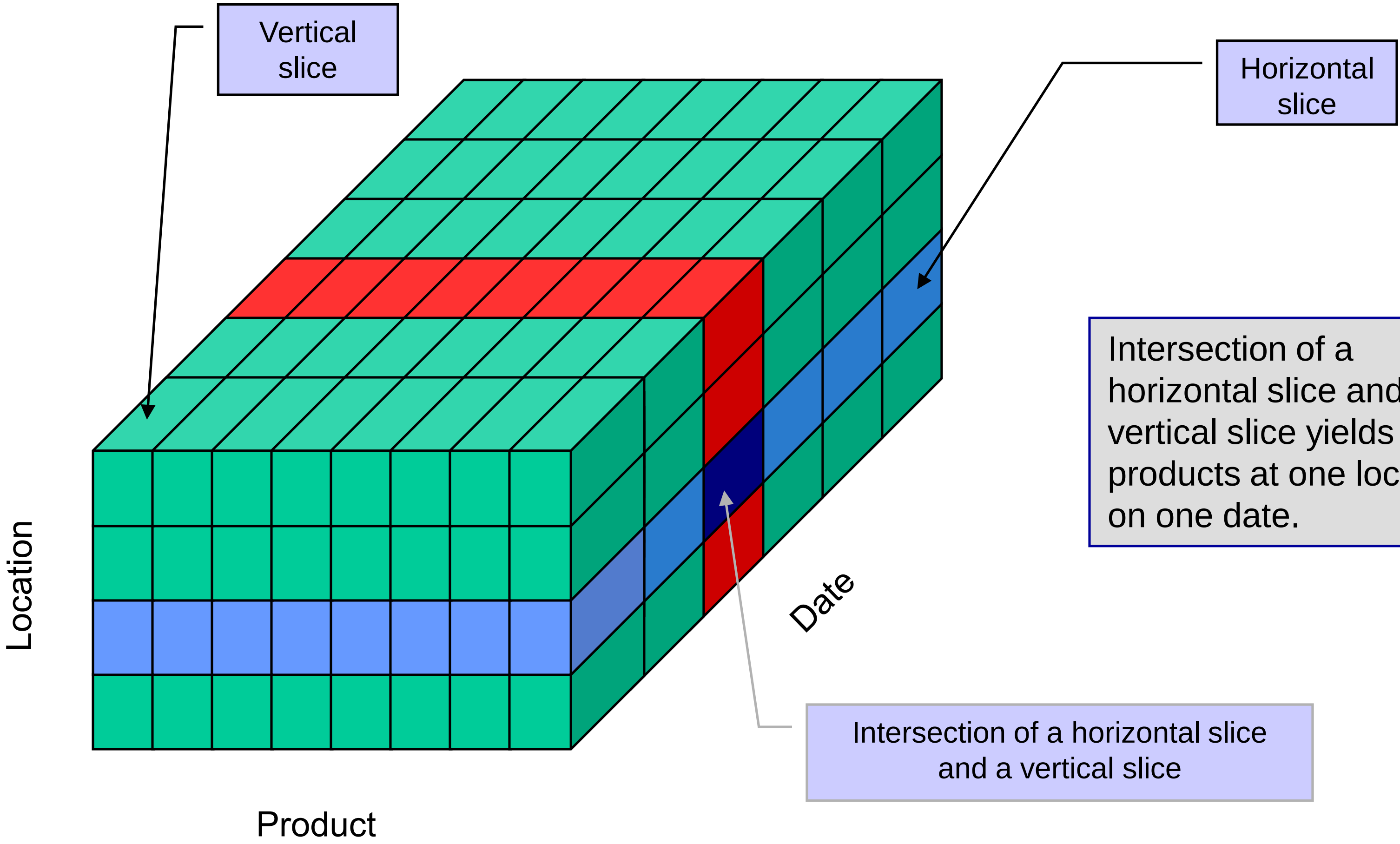
Horizontal slice, vertical dice. One product at one location over all dates.

Three Dimensional View of Data



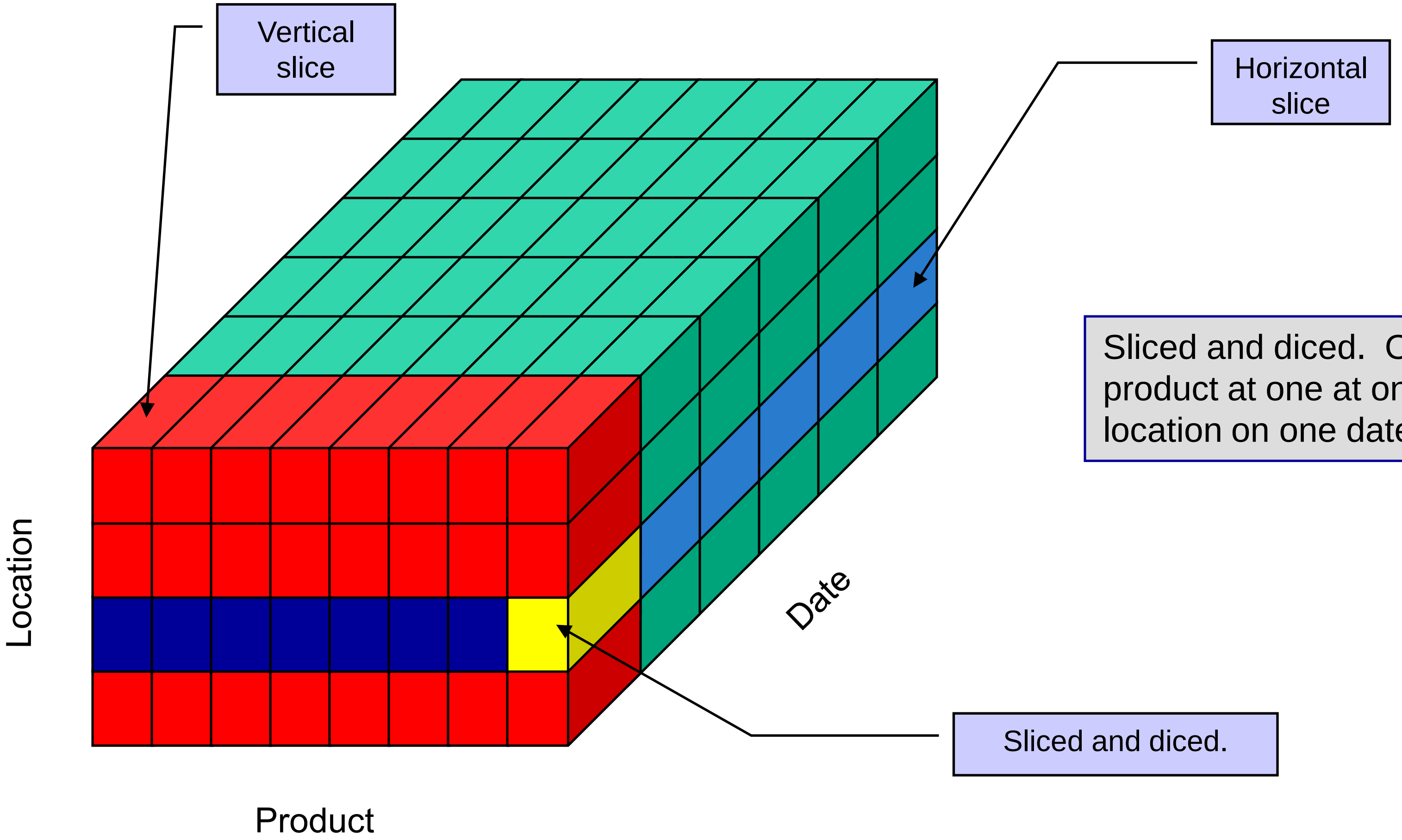
Intersection of a horizontal slice and vertical slice yields all products at one location on one date.

Three Dimensional View of Data



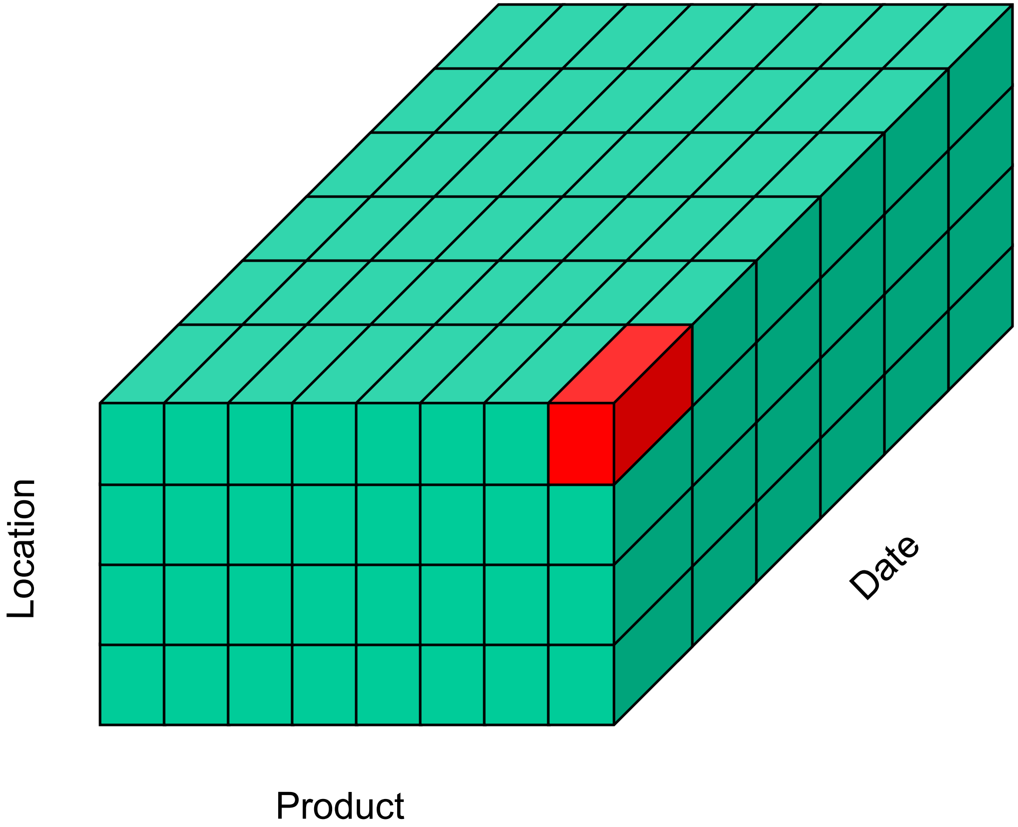
Intersection of a horizontal slice and vertical slice yields all products at one location on one date.

Three Dimensional View of Data



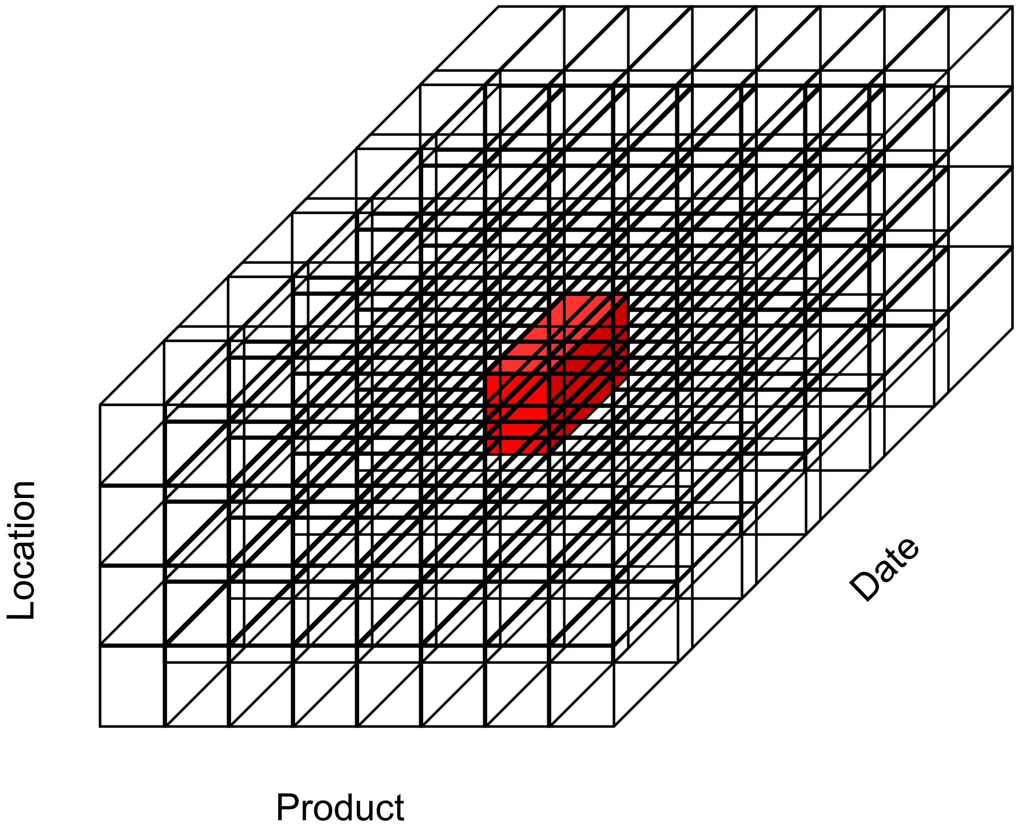
Sliced and diced. One product at one at one location on one date.

Three Dimensional View of Data



Sliced and diced. One product at one location on one date.

Three Dimensional View of Data

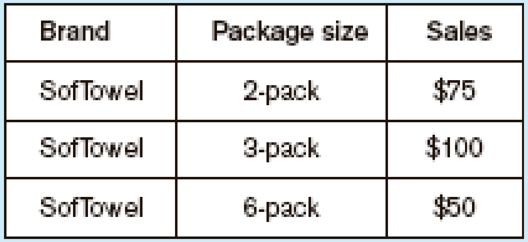


Sliced and diced. One product at one location on one date.

Example of drill-down

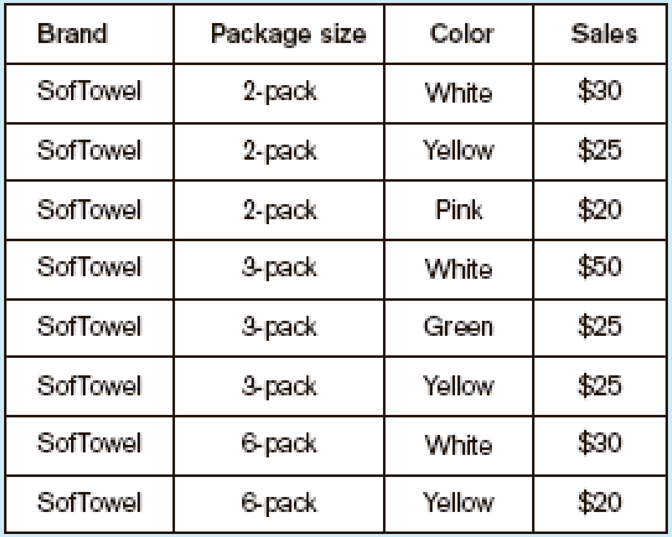
Starting with summary data, users can obtain details for particular cells

Summary report



Brand	Package size	Sales
SofTowel	2-pack	$75
SofTowel	3-pack	$100
SofTowel	6-pack	$50

Drill-down with color added



Brand	Package size	Color	Sales
SofTowel	2-pack	White	$30
SofTowel	2-pack	Yellow	$25
SofTowel	2-pack	Pink	$20
SofTowel	3-pack	White	$50
SofTowel	3-pack	Green	$25
SofTowel	3-pack	Yellow	$25
SofTowel	6-pack	White	$30
SofTowel	6-pack	Yellow	$20

Data Mining and Visualization

- Knowledge discovery using a blend of statistical, AI, and computer graphics techniques
- Goals:
  - Explain observed events or conditions
  - Confirm hypotheses
  - Explore data for new or unexpected relationships
- Techniques
  - Statistical regression
  - Decision tree induction
  - Clustering and signal processing
  - Affinity
  - Sequence association
  - Case-based reasoning
  - Rule discovery
  - Neural nets
  - Fractals
- Data visualization–representing data in graphical/multimedia formats for analysis

Introduction to Data Mining

- The amount of data maintained in computer files and databases is growing at a phenomenal rate.

- At the same time, the users of these data are expecting more sophisticated information from them.

  - A marketing manager is no longer satisfied with a simple listing of marketing contacts, but wants detailed information about customers' past purchases as well as predictions of future purchases.

- Simple structured/query language queries are not adequate to support these increased demands for information.

- Data mining has evolved as a technique to support these increased demands for information.

Introduction to Data Mining (cont.)

- Data mining is often defined as finding hidden information in a database.

- Alternatively, it has been called exploratory data analysis, data driven discovery, and deductive learning.

- We'll look at a somewhat more focused definition that was provided by Simoudis (1996, *IEEE Expert*, Oct, 26-33) who defines data mining as:

> The process of extracting valid, previously unknown, comprehensible, and actionable information from large database and using that information to make crucial business decisions.

Introduction to Data Mining (cont.)

- Traditional database queries access a database using a well-defined query state in a language such as SQL. The output of the query consists of the data from the database that satisfies the query. The output is usually a subset of the database, but it may also be an extracted view or contain aggregations.

- Data mining access of the database differs from this traditional access in three major areas:

1. Query: The query might not be well formed or precisely stated. The data miner might not even be exactly sure of what they want to see.
2. Data: The data access is usually a different version from that of the operational database (it typically comes from a data warehouse). The data must be cleansed and modified to better support mining operations.
3. Output: The output of the data mining query probably is not a subset of the database. Instead it is the output of some analysis of the contents of the database.

Introduction to Data Mining (cont.)

- The current state of the art in data mining is similar to that of database query processing in the late 1960s and early 1970s. Over the next decade or so, there will undoubtedly be great strides in extending the state of the art with respect to data mining.
- We will probably see the development of "query processing" models, standards, and algorithms targeting data mining applications.
- In all likelihood we will also see new data structures designed for the storage of database being using specifically for data mining operations.
- Although data mining is still a relatively young discipline, the last decade has witnessed a proliferation of mining algorithms, applications, and algorithmic approaches to mining.

A Brief Data Mining Example

- Credit card companies must determine whether to authorize credit card purchases. Suppose that based on past historical information about purchases, each purchase is placed into one of four classes: (1) authorized, (2) ask for further identification before authorization, (3) do not authorize, and (4) do not authorize and contact the police.

- The data mining functions here are twofold.
  - First, the historical data must be examined to determine how the data fit into the four classes. That is, how all of the previous credit card purchases should be classified.
  - Second, once classified the problem is to apply this model to each new purchase.

- The second step above can be stated as a simple database query if things are properly set-up, the first problem cannot be solved with a simple query.

Introduction to Data Mining (cont.)

- Data mining involves many different algorithms to accomplish different tasks. All of these algorithms attempt to fit a model to the data.

- The algorithms examine the data and determine a model that is the closest fit to the characteristics of the data being examined.

- Data mining algorithms can be viewed as consisting of three main parts:

1. Model: The purpose of the algorithms is to fit a model to the data.
2. Preference: Some criteria must be used to fit one model over another.
3. Search: All algorithms require some technique to search the data.

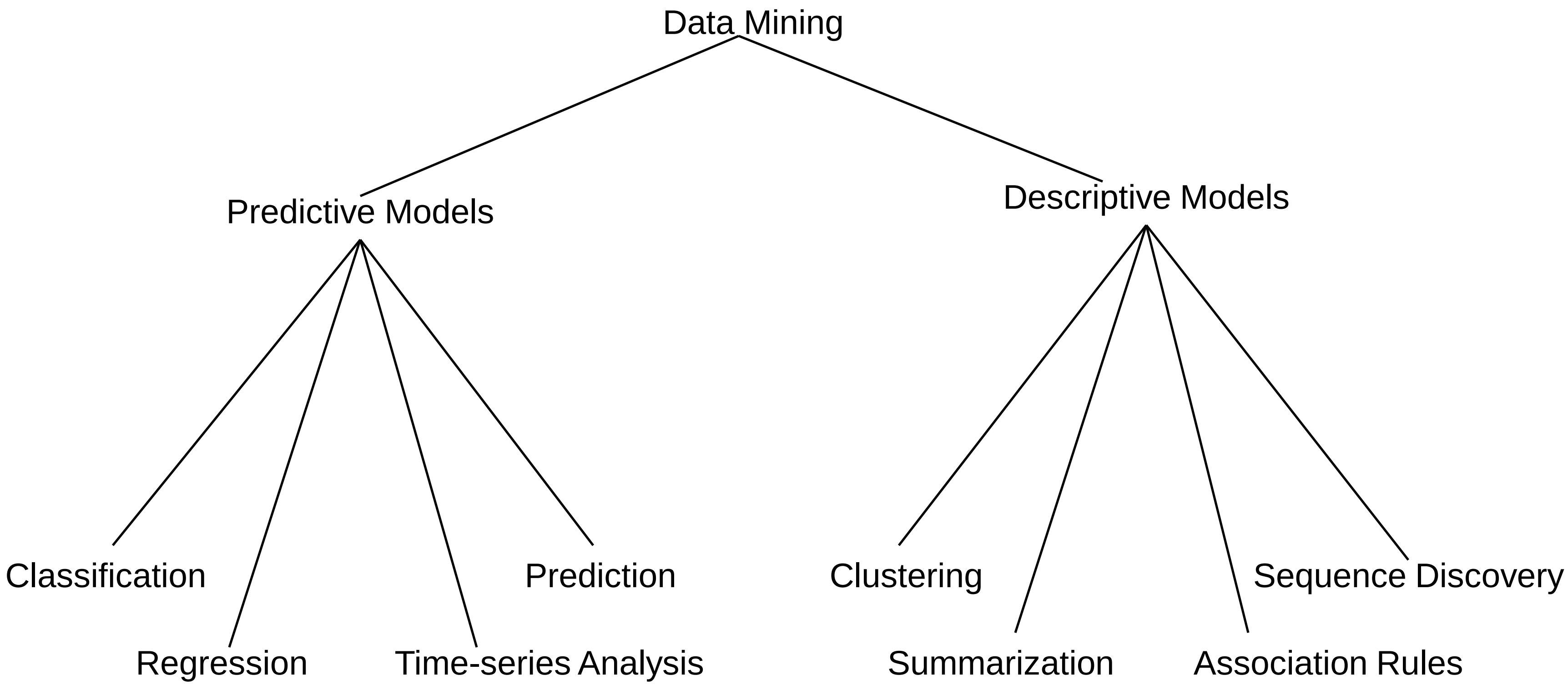
Data Mining Models

- A predictive model makes a prediction about values of data using known results found from different data. Predictive modeling is commonly based on the use of other historical data.
  - For example, a credit card use might be refused not because of the user’s own credit history, but because a current purchase is similar to earlier purchases that were subsequently found to be made with stolen cards.
  - Predictive model data mining tasks include classification, regression, time series analysis, and prediction (as a specific data mining function).

Data Mining Models (cont.)

- A descriptive model identifies patterns or relationships in data. Unlike the predictive model, a descriptive model serves as a way to explore the properties of the data examined, not to predict new properties.
  - For example, a credit card purchase may be not authorized because the amount of the charge is way out of line with your typical charges. In other words, if you have a past history where your average charge amount is $100.00 and the current transaction is for $5000.00 the charge might not be authorized using this model. This is a summarization technique.
  - Clustering, summarizations, association rules, and sequence discovery are usually viewed as descriptive in nature.

Data Mining Models and Tasks



Data Mining

Predictive Models

Classification

Regression

Time-series Analysis

Prediction

Descriptive Models

Clustering

Summarization

Association Rules

Sequence Discovery

Data mining models and some typical tasks. Not an exhaustive listing.
Combinations of these tasks yield more sophisticated mining operations.

Basic Data Mining Tasks

Classification (predictive model)

- Classification maps data into predefined groups or classes. It is often referred to as supervised learning because the classes are not determined before examining the data.

- Two examples of classification applications are determining whether to make a bank loan and identifying credit risks.

- Classification algorithms require that the classes be defined based on data attribute values. They often describe these classes by looking at the characteristics of data already known to belong to the classes.

  – Supervised learning normally consists of two phases: training and testing. Training builds a model using a large sample of historical data called a training set, while testing involves trying out the model on new, previously unseen data to determine its accuracy and physical performance characteristics.

Basic Data Mining Tasks (cont.)

Classification (cont.)

- Pattern recognition is a type of classification where an input pattern is classified into one of several classes based on its similarity to these predefined classes.
- The example on page 6 is an example of a general classification problem.
- An example of pattern recognition would be an airport security system used to determine if passengers are potential terrorists or criminals. Each passenger's face is scanned and its basic pattern (distance between eyes, size and shape of mouth, shape of head, etc.) is identified. This pattern is compared to entries in a database to see if it matches any patterns that are associated with known offenders.

Basic Data Mining Tasks (cont.)

Classification (cont.)

- There are two major types of classification algorithms: tree induction and neural induction.

- To illustrate the differences and similarity in these two techniques, consider the following example:

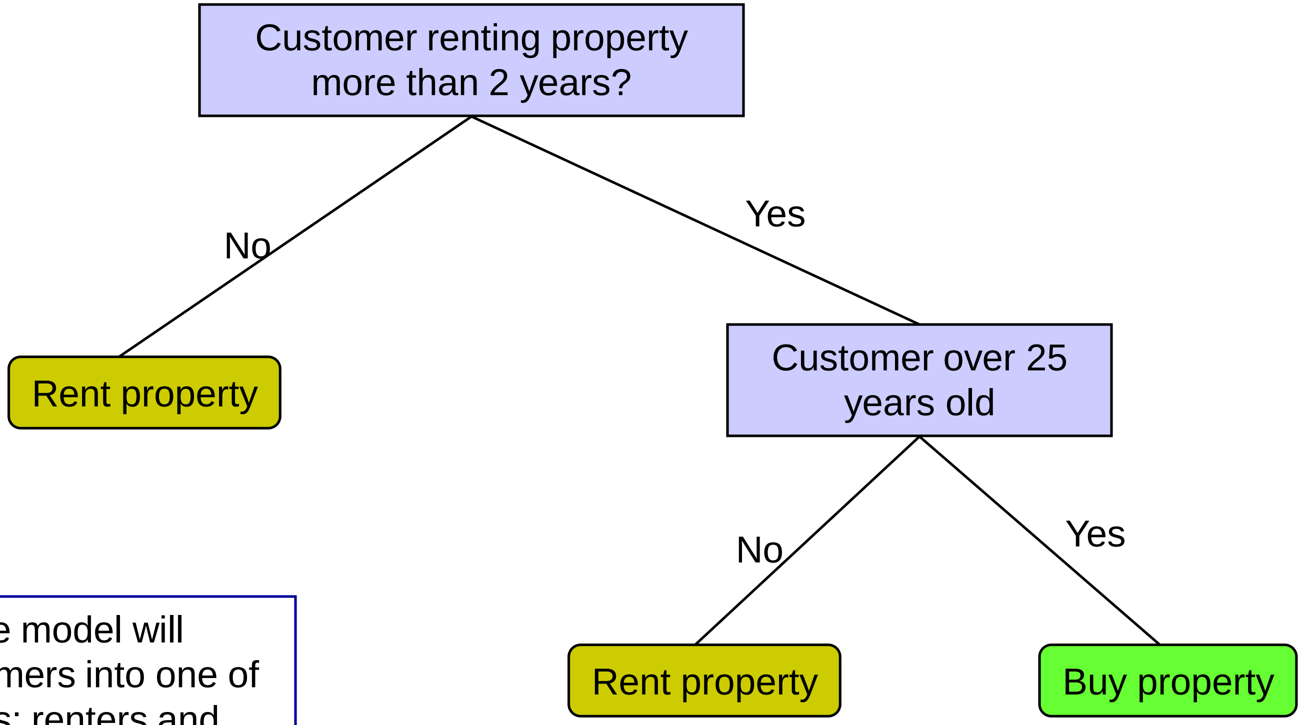
  – Suppose that we are interested in predicting whether a customer who is currently renting property is likely to be interested in buying property.

  – Assume that a predictive model has determined that only two variables are of interest: the length of time the customer has rented property and the age of the customer.

  – Tree induction presents the analysis in an intuitive way, using a decision tree (similar in some ways to a flow chart). A possible classification using tree induction is shown in the following diagram:

Basic Data Mining Tasks (cont.)

Classification (cont.)


Customer renting property more than 2 years?
No
Yes
Rent property
Customer over 25 years old
No
Yes
Rent property
Buy property

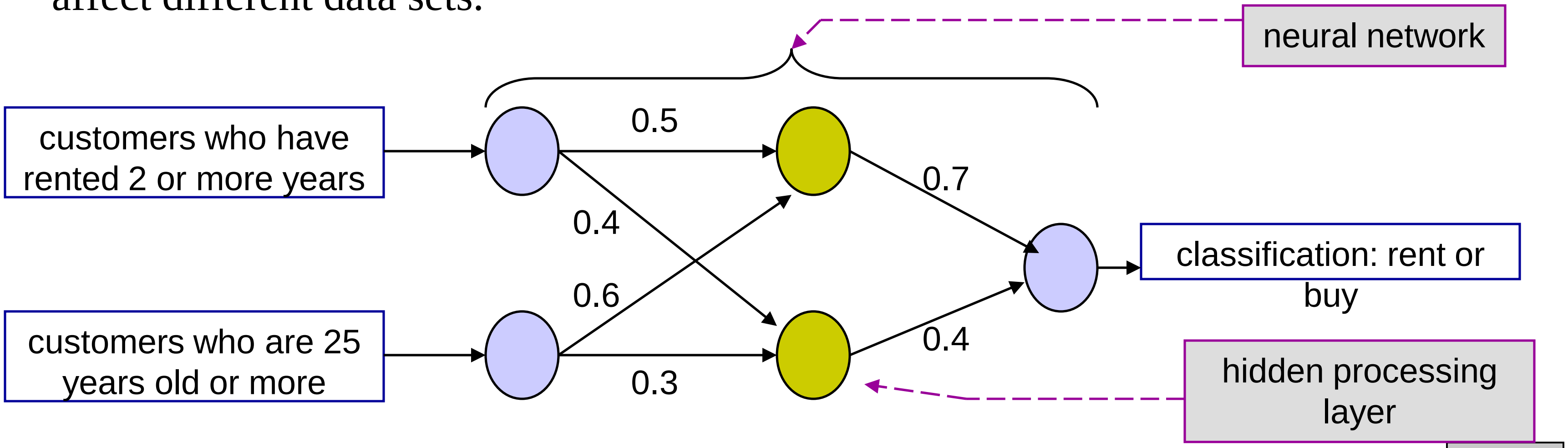
This predictive model will classify customers into one of two categories: renters and buyers. The model will predict that customers who are over 25 years old and have rented for more than 2 years will buy property, others will rent.

Classification Using An Induction Tree

Basic Data Mining Tasks (cont.)

Classification (cont.)

- Using neural induction, for the same example, would require the use of a neural network. A neural network contains collections of connected nodes with input, output, and processing at each node. Between the visible input and output layers may be a number of hidden processing layers. Each processing unit (the circles in the diagram) in one layer is connected to each processing unit in the next layer by a weighted value, expressing the strength of the relationship. The network attempts to mirror the way the human brain works in recognizing patterns by arithmetically combining all the variables associated with a given data point. In this way, it is possible to develop nonlinear predictive models that "learn" by studying combinations of variables and how different combinations of variables affect different data sets.



neural network
customers who have rented 2 or more years
customers who are 25 years old or more
0.5
0.4
0.6
0.3
0.7
0.4
classification: rent or buy
hidden processing layer

Basic Data Mining Tasks (cont.)

Regression (predictive model)

- Regression is used to map a data item to a real valued prediction variable.
- In actuality, regression involves the learning of the function that does this mapping.
- Regression assumes that the target data fit into some known type of function (i.e., linear, logistic, etc.) and then determines the best function of this type that models the given data.
- Some type of error analysis is used to determine which function is "best", i.e., produces the least total error.
- As an example of simple linear regression let's suppose that you are maintaining a retirement savings portfolio and wish to reach a certain level of savings before retirement. Periodically, you will predict what your savings will be based on the current amount and several past amounts. Using a simple linear regression formula you then predict what the value will be in the future by fitting the past values to a linear function and then use that function to predict values at points in the future. Based on these values, you then alter (or not) your investment portfolio.

Basic Data Mining Tasks (cont.)

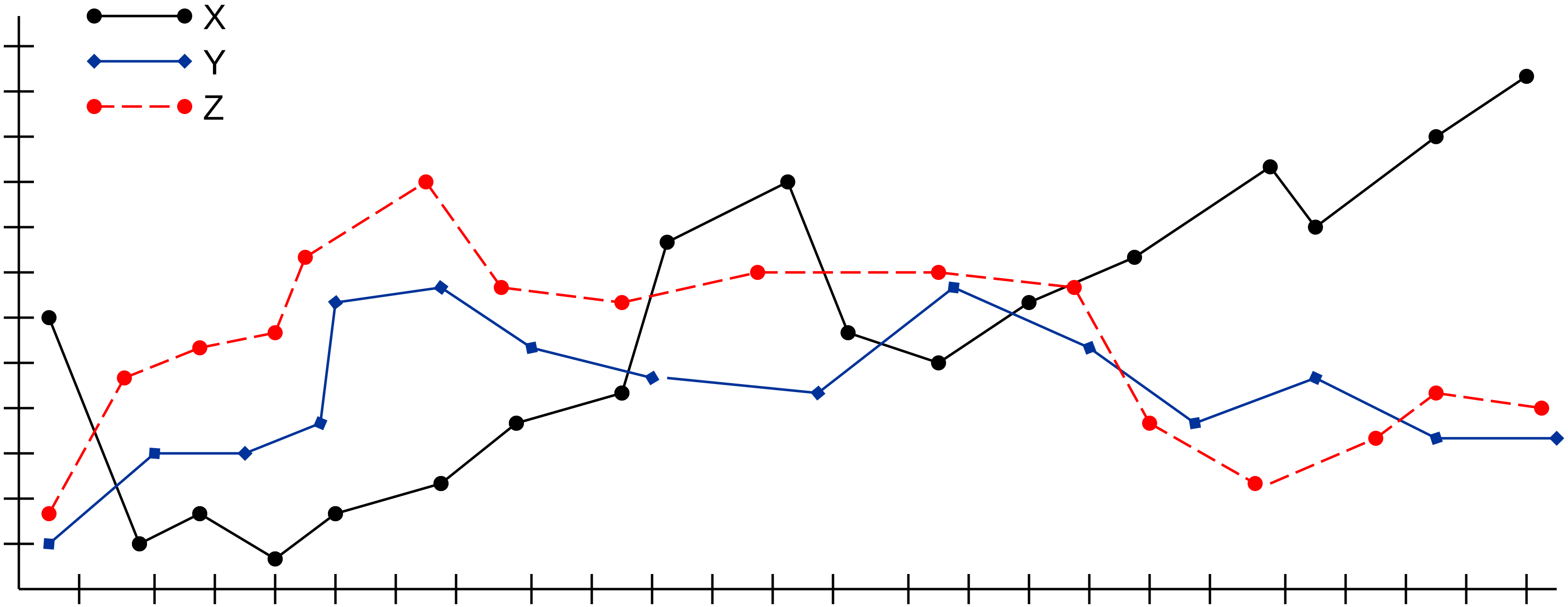
Regression (cont.)

- Linear regression attempts to fit a straight line through the plot of the data, such that the line is the best representation of the average of all observations at that point in the plot.
- The problem with linear regression is that the technique only works well with linear data and is sensitive to the presence of outliers (data values which do not conform to the expected norm).
- Although nonlinear regression avoids the main problems of linear regression, it is still not flexible enough to handle all possible shapes of the data plot.
- This is where the traditional statistical analysis methods and data mining methods begin to diverge. Statistical measurements are fine for building linear models that describe predictable data points, however, most data is not linear in nature.
- Data mining requires statistical methods that can accommodate nonlinearity, outliers, and non-numeric data.

Basic Data Mining Tasks (cont.)

Time Series Analysis (predictive model)

- With time series analysis, the value of an attribute is examined as it varies over time. The values usually are obtained as evenly spaced time points (daily, weekly, hourly, etc.).
- A time series plot is used to visualize the time series. In the example below, the plots for Y and Z appear to have similar behavior, while X appears less similar.



Basic Data Mining Tasks (cont.)

Time Series Analysis (cont.)

- There are three basic functions performed in time series analysis.
- In one case, distance measures are used to determine the similarity between different time series. For example, using the time series on the previous page we could look at the difference in daily stock prices between the three companies, or perhaps the difference between their beginning and ending prices, etc..
- In the second case, the structure of the line is examined to determine (and perhaps classify) its behavior. This could be a generality, such as X appears to be trending upwards, or it could use very specific curve fitting techniques.
- A third case would occur when historical time series plots are used to predict future values. Various extrapolation techniques could be applied.

Basic Data Mining Tasks (cont.)

Time Series Analysis (cont.)

- As an example of how to use time series analysis, suppose that you are deciding whether to purchase stock from Companies X, Y, or Z. Assuming that the time series plots illustrated on page 14 were tracking the daily stock prices for each company, you might decide to purchase stock in either Y or Z because they appear to be less volatile (fluctuate less on a daily basis) that does the stock for company X. On the other hand you might decide to purchase stock in company X because it shows an overall growth which is larger than either of the other two stocks.

Basic Data Mining Tasks (cont.)

Prediction (predictive model)

- Many real-world data mining application can be seen as predicting future data states based on past and current data.
- Prediction can be also be viewed as a type of classification. Note that this is a data mining task which is different from the prediction model, although the prediction task is a type of the prediction model. The difference is that prediction is predicting a future state rather than a current state.
- An example of prediction can be illustrated with the application of the prediction of flooding. In general predicting flooding is a difficult problem. One approach uses monitors placed at various points along a river. The monitors collect data relevant to flood prediction such as water levels, rain amounts, time, humidity, etc.. Then the water level at a potential flooding point in the river can be predicted based on the data collected by the sensors upriver from this point. The prediction must be made with respect to the time the data were collected.

Basic Data Mining Tasks (cont.)

Clustering (descriptive model)

- Clustering is similar to classification except that the groups are not predefined, but rather defined by the data alone.
- Clustering is alternatively referred to as unsupervised learning or segmentation (actually, segmentation is a special case of clustering although many people refer to them synonymously).
- Clustering can be thought of as partitioning or segmenting the data into groups that might or might not be disjoint.
- Clustering is usually accomplished by determining the similarity among the data on predefined attributes. The most similar data are grouped into clusters.
- Since clusters are not predefined, a domain expert is often required to interpret the meaning of the created clusters.
- As an example of clustering, suppose that you are an instructor for COP 3502 and you have 10 different lab sections for the course. Students attend a particular lab section. If you have a database in which each student's lab quiz scores are recorded, then you can cluster (segment) the database using the lab section as a clustering attribute and cluster students attending the same lab section together.

Basic Data Mining Tasks (cont.)

Summarization (descriptive model)

- Summarization maps data into subsets with associated simple descriptions. It extracts or derives representative information about the database.
- This is commonly accomplished by actually retrieving portions of the data. Alternatively, summary type information (e.g., the mean of some numeric attribute) can be derived from the data.
- Summarization succinctly characterizes the contents of the database.
- Summarization is also called characterization or generalization.
- An example of summarization would be one of the many criteria used to compare universities by U.S. News and World Report which is average SAT score. This summarization is used to estimate the type and intellectual level of a student body.

Basic Data Mining Tasks (cont.)

Association Rules (descriptive model)

- Association is also called link analysis or affinity analysis, and refers to the data mining task of uncovering relationships among the data.
- The best example of this type of application is to determine association rules. An association rule is a model that identifies specific types of data associations. These associations are often used in the retail sales world to identify items that are frequently purchased together. This is commonly referred to as market basket analysis.
  - As an example of association rules, suppose that a grocery store manager is trying to decide whether or put bread on sale. To help determine the impact of this decision, the manager generates association rules that show what other products are frequently purchased with bread. Suppose that they discover that 60% of the time bread is purchased with pretzels and 70% of the time bread is purchased with jelly. Based on these facts, the manager attempts to capitalize on the association between bread, pretzels and jelly by placing some pretzels and jelly on the end of the aisle where the bread is located. In addition, he decides never to place both of these items on sale at the same time!
- Associations are also used in many other applications such as predicting the failure of telecommunication switches.

Basic Data Mining Tasks (cont.)

Association Rules (cont.)

- When using association rules, one must remember that these are not casual relationships. They doe not represent and relationship inherent in the actual data as is the case with functional dependencies, or in the real world.

- There is probably no relationship between bread pretzels that causes them to be purchased together. Furthermore, there is no guarantee that this association will apply in the future.

- However, association rules are heavily used in the retail sector in creating effective advertising, marketing and inventory control strategies.

Basic Data Mining Tasks (cont.)

Sequence Discovery (descriptive model)

- Sequential analysis or sequence discovery is used to determine sequential patterns in data. These patterns are based on a time sequence of actions.
- These patterns are similar to associations in that the data (or events) are found to be related, but the relationship is based on time. This is different from market basket analysis, which requires the related objects to be purchased at the same time. In sequence discovery, the items are purchased over some period of time in some order.
- For example, most people who purchase a DVD player may be found to purchase DVDs within one week.
- Temporal association rules really fall into this category although some people try to force the issue and maintain them as strict association rules.

Knowledge Discovery in Databases vs. Data Mining

- The terms knowledge discovery in databases (KDD) and data mining are often used interchangeably. However, over the last few years KDD has been used to refer to a process consisting of many steps, while data mining is only one of these steps.

- Data mining has become a specific operation in the wider arena of knowledge discovery.

> Knowledge discovery in databases (KDD) is the process of finding useful information and patterns in data. Data mining is the use of algorithms to extract the information and patterns derived by the KDD process.

- KDD is a process that involved many different steps. The input to this process is the data and the output is the useful information desired by the users. However, the objective may be unclear or inexact. The process itself is interactive and may require much elapsed time.

- To ensure the accuracy and usefulness of the results, interaction throughout the process with both domain experts and technical experts may be needed.

The KDD Process

- The KDD process consists of the following five basic steps:

1. Selection: The data needed for the data mining process is obtained from many different and heterogeneous data sources.
2. Preprocessing: The data to be used by the process may have incorrect or missing data. There may be anomalous data from multiple sources involving different data types and metrics. There may be many different activities performed during this step. Erroneous data may be corrected or removed, whereas missing data must be supplied or predicted (often using data mining tools).
3. Transformation: Data from different sources must be converted into a common format for processing. Some data may be encoded or transformed into more usable formats. Data reduction may be used to reduce the number of possible data values being considered.
4. Data mining: Based on the data mining task being performed, this step applies the algorithms to the transformed data to generate the desired results.
5. Interpretation/evaluation: How the data mining results are presented to the users is extremely important because the usefulness of the results is dependent on it. Various visualization and GUI strategies are used in this last step.

Data Mining Issues

- There are many important implementation issues associated with data mining:

1. Human interaction: Since data mining problems are often not precisely stated, interfaces may be needed with both domain and technical experts. Technical experts are used to formulate the queries and assist in interpreting the results. Users must identify training data and desired results.

2. Overfitting: When a model is generated that is associated with a given database state, it is desirable that the model also fit future database states. Overfitting occurs when the model does not fit future states. This may be caused by assumptions that are made about the data or may simply be caused by the small size of the training database. For example, a classification model for an employee database may be developed to classify employees as short, medium, or tall. If the training database is quite small, the model might erroneously indicate that a short person is anyone under 5' 8" because there is only one entry in the training database under 5' 8". In this case, many future employees would be erroneously classified as short. Overfitting can arise under other circumstances as well, even though the data are not changing.

Data Mining Issues (cont.)

3. Outliers: There are often many data entries that do not fit nicely into the derived model. This becomes even more of an issue with VLDBs. If a model is developed that includes these outliers, then the model may not behave well for data that are not outliers.

4. Interpretation of results: Currently, data mining output may require experts to correctly interpret the results, which might otherwise be meaningless to the average database user.

5. Visualization of the results: To easily view and understand the output of data mining algorithms, visualization of the results is essential. Selection of the appropriate tool becomes critical to aid in the interpretation.

6. Large datasets: The massive datasets associated with data mining create problems when applying algorithms designed for small datasets. Many modeling applications grow exponentially on the dataset size and thus are too inefficient for larger datasets. Sampling and parallelization are effective tools to attack this scalability problem.

Data Mining Issues (cont.)

7. High dimensionality: A conventional database schema may be composed of many different attributes. The problem here is that not all attributes may be needed to solve a given data mining problem. In fact, the use of some attributes may interfere with the correct completion of a data mining task. The use of other attributes may simply increase the overall complexity and decrease the efficiency of an algorithm. This problem is sometimes referred to as the dimensionality curse, meaning that there are many attributes (dimensions) involved and it is difficult to determine which ones should be used. One solution to this high dimensionality problem is to reduce the number of attributes, which in known as dimensionality reduction. However, determining which attributes are not needed is not always easy to do.

8. Multimedia data: Most previous data mining algorithms are targeted to traditional data types (numeric, character, text, etc.). The use of multimedia data such as found in GIS databases complicates or invalidates many proposed algorithms.

Data Mining Issues (cont.)

9. Missing data: During the preprocessing phase of KDD, missing data may be replaced with estimates. This and other approaches to handling missing data can lead to invalid results in the data mining step.

10. Irrelevant data: Some attributes in the database might not be of interest to the data mining task being developed.

11. Noisy data: Some attribute values might be invalid or incorrect. These values are often corrected before running data mining applications.

12. Changing data: Databases cannot be assumed to be static. However, most data mining algorithms do assume a static database. This requires that the algorithms be completely rerun anytime the database changes.

13. Integration: The KDD process is not currently integrated into normal data processing activities. KDD requests may be treated as special, unusual, or one-time needs. This makes them inefficient, ineffective and not general enough to be used on an ongoing basis. Integration of data mining functions into traditional DBMSs is certainly a desirable goal.

14. Application: Determining the intended use for the information obtained from the data mining function is a challenge. How business executives can effectively use the output is sometimes considered the more difficult part, not the running of the algorithms themselves. Because the data are of a type that has not previously been known, business practices may have to be modified to determine how to effectively use the information uncovered.