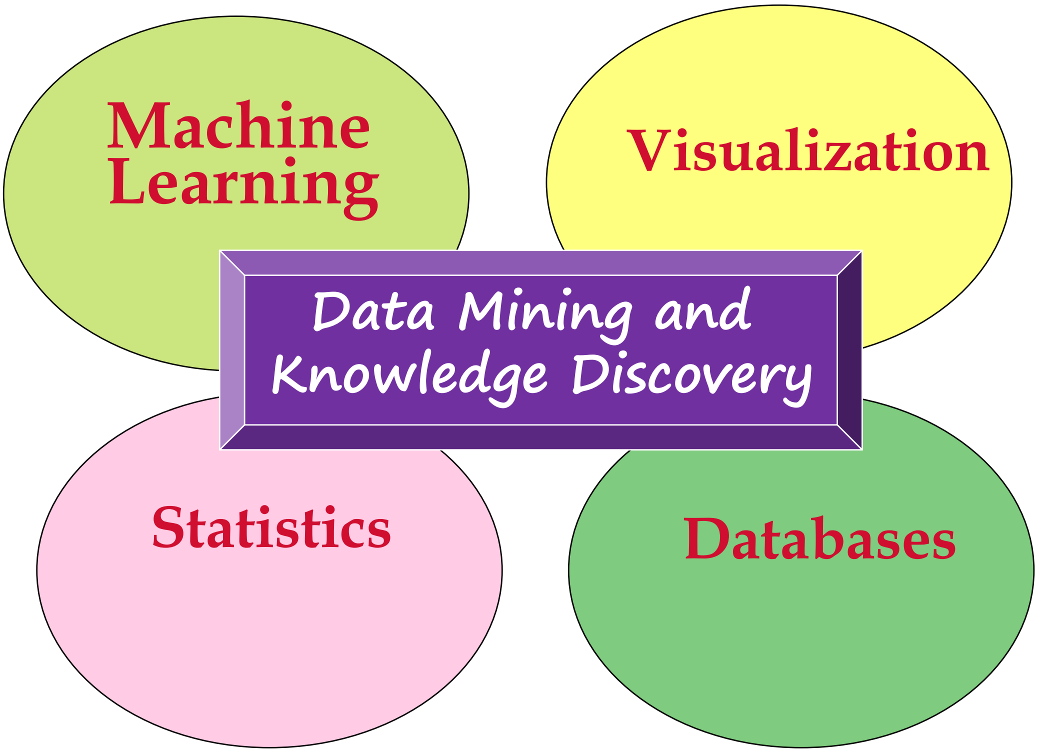
# Data Mining, Big Data and Analytics

CSC 375 Fall 2019

Is this the place to learn about mining?

> "To find signals in data, we must learn to reduce the noise ~ not just the noise that resides in the data, but also the noise that resides in us. It is nearly impossible for noisy minds to perceive anything but noise in data."
>
> — Stephen Few

## Related Fields



Machine Learning

Visualization

Data Mining and Knowledge Discovery

Statistics

Databases

Source: Gregory Piatetsky-Shapiro

# Introduction

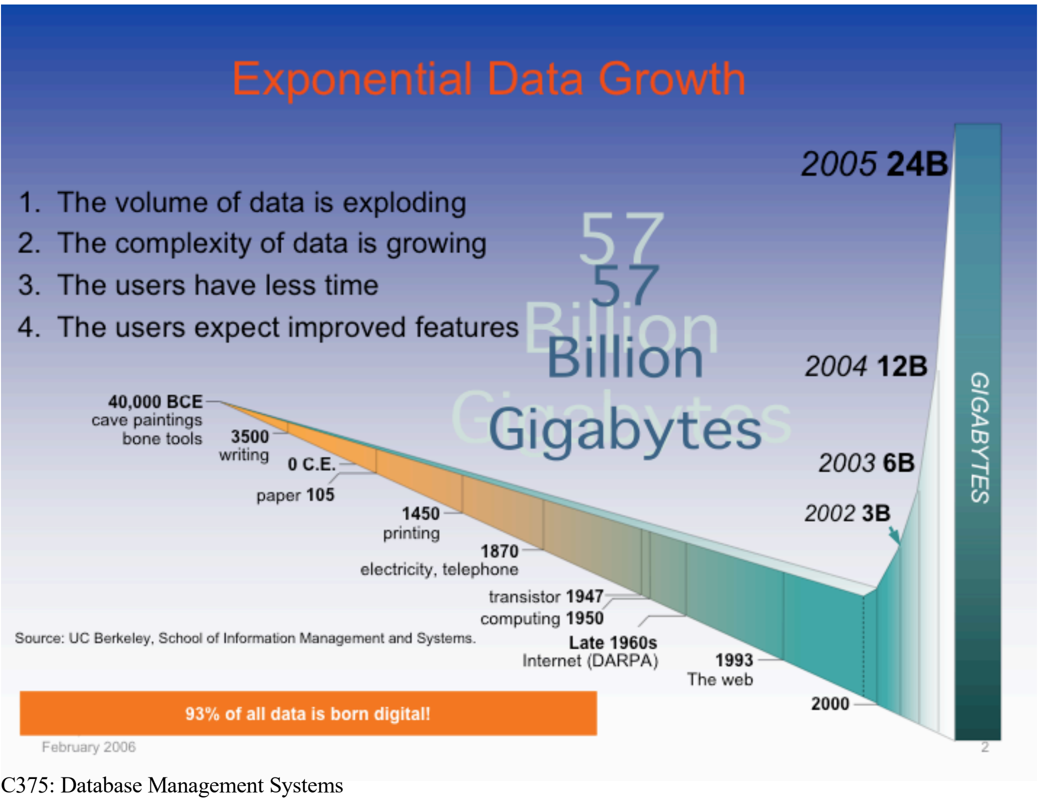
- **Big Data**
  - Data that exist in very large volumes and many different varieties (data types) and that need to be processed at a very high velocity (speed).
- **Analytics**
  - Systematic analysis and interpretation of data—typically using mathematical, statistical, and computational tools—to improve our understanding of a real-world domain.

# Big Data and Databases: Famous Quotes

"640K ought to be enough for anybody."

*Bill Gates, 1981*

# Digitization of Everything: the Zettabytes are coming



Exponential Data Growth

1. The volume of data is exploding
2. The complexity of data is growing
3. The users have less time
4. The users expect improved features

Source: UC Berkeley, School of Information Management and Systems.

93% of all data is born digital!

February 2006

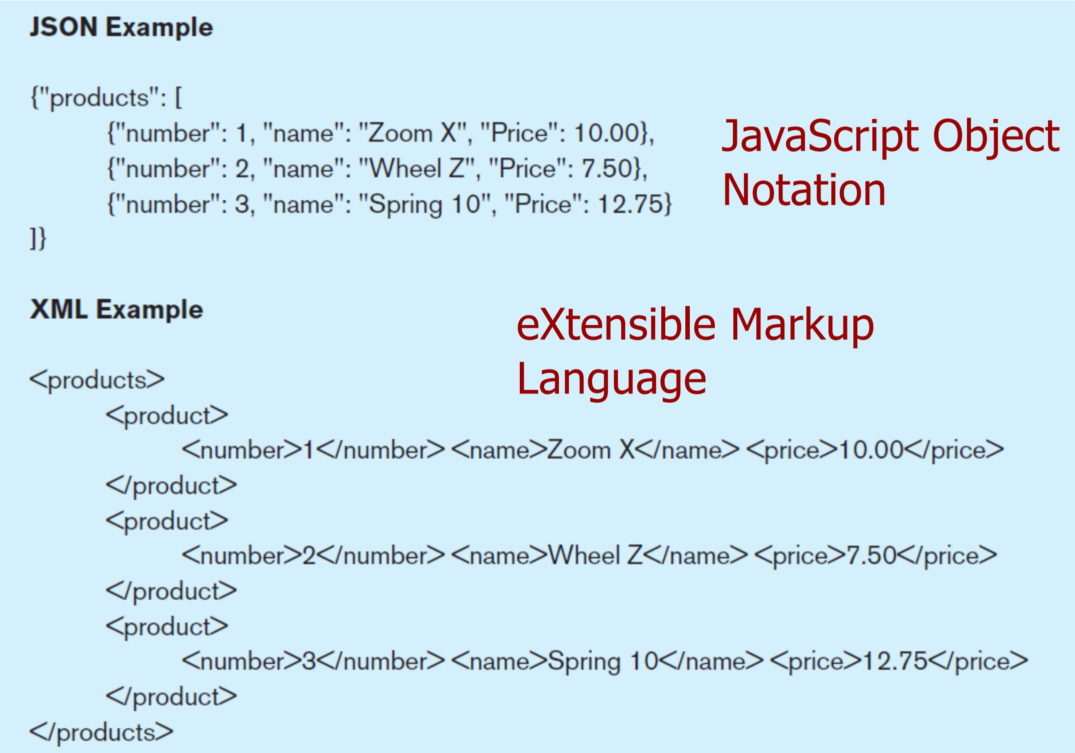
# Characteristics of Big Data

- **The Five Vs of Big Data**
  - Volume – much larger quantity of data than typical for relational databases
  - Variety – lots of different data types and formats
  - Velocity – data comes at very fast rate (e.g. mobile sensors, web click stream)
  - Veracity – traditional data quality methods don’t apply; how to judge the data’s accuracy and relevance?
  - Value – big data is valuable to the bottom line, and for fostering good organizational actions and decisions

# Characteristics of Big Data

- **Schema on Read, rather than Schema on Write**
  - Schema on Write– preexisting data model, how traditional databases are designed (relational databases)
  - Schema on Read – data model determined later, depends on how you want to use it (XML, JSON)
  - Capture and store the data, and worry about how you want to use it later
- **Data Lake**
  - A large integrated repository for internal and external data that does not follow a predefined schema
  - Capture everything, dive in anywhere, flexible access

# Examples of JSON and XML



**JSON Example**

```
{"products": [
        {"number": 1, "name": "Zoom X", "Price": 10.00},
        {"number": 2, "name": "Wheel Z", "Price": 7.50},
        {"number": 3, "name": "Spring 10", "Price": 12.75}
]}
```

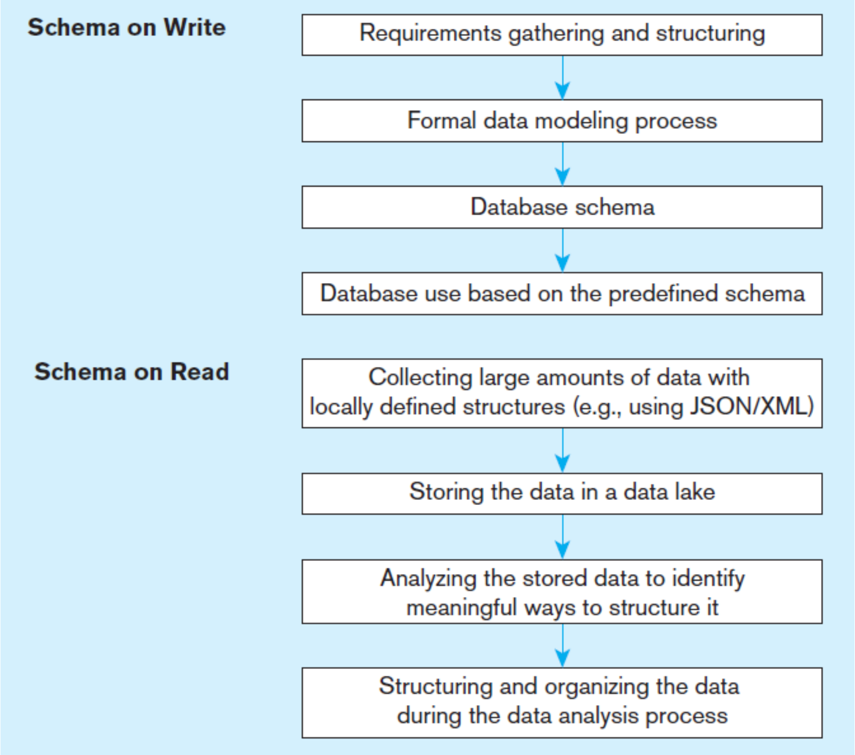
JavaScript Object Notation

**XML Example**

eXtensible Markup Language

```
<products>
        <product>
                <number>1</number> <name>Zoom X</name> <price>10.00</price>
        </product>
        <product>
                <number>2</number> <name>Wheel Z</name> <price>7.50</price>
        </product>
        <product>
                <number>3</number> <name>Spring 10</name> <price>12.75</price>
        </product>
</products>
```

# Schema on write vs. schema on read



**Schema on Write**

Requirements gathering and structuring

Formal data modeling process

Database schema

Database use based on the predefined schema

**Schema on Read**

Collecting large amounts of data with locally defined structures (e.g., using JSON/XML)

Storing the data in a data lake

Analyzing the stored data to identify meaningful ways to structure it

Structuring and organizing the data during the data analysis process

Traditional database design

The big data approach

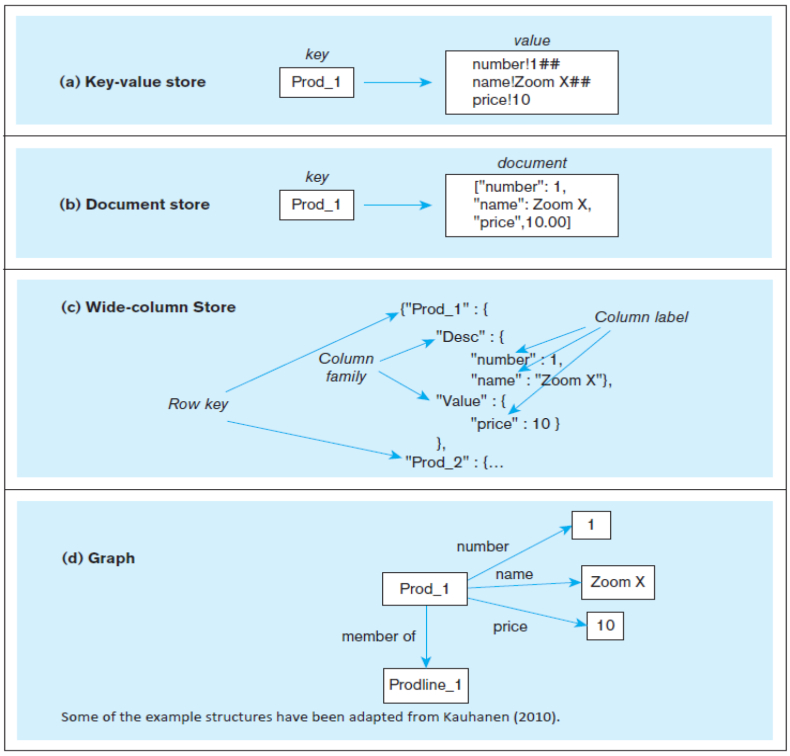
# NoSQL

- **NoSQL = Not Only SQL**
- **A category of recently introduced data storage and retrieval technologies not based on the relational model**
- **Scaling out rather than scaling up**
- **Natural for a cloud environment**
- **Supports schema on read**
- **Largely open source**
- **Not ACID compliant!**
- **BASE – basically available, soft state, eventually consistent**

# NoSQL Classifications

- **Key-value stores**
  - A simple pair of a key and an associated collection of values. Key is usually a string. Database has no knowledge of the structure or meaning of the values.
- **Document stores**
  - Like a key-value store, but “document” goes further than “value”. Document is structured so specific elements can be manipulated separately.
- **Wide-column stores**
  - Rows and columns. Distribution of data based on both key values (records) and columns, using “column groups/families”
- **Graph-oriented database**
  - Maintain information regarding the relationships between data items. Nodes with properties, Connections between nodes (relationships) can also have properties.

# Four-part figure illustrating NoSQL databases



Some of the example structures have been adapted from Kauhanen (2010).

# NoSQL Comparison

- **Redis – Key-value store DBMS**
- **MongoDB – document store DBMS**
- **Apache Cassandra – wide-column store DBMS**
- **Neo4j – graph DBMS**

# NoSQL Comparison

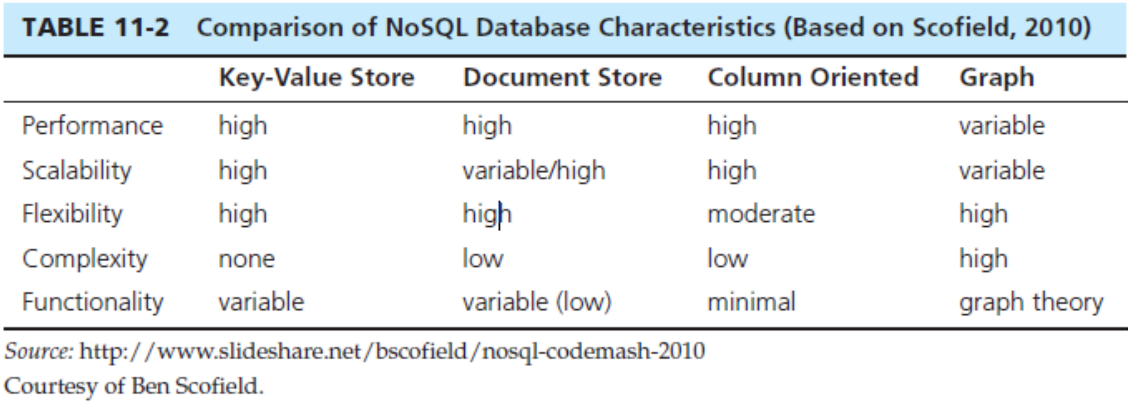


**TABLE 11-2** Comparison of NoSQL Database Characteristics (Based on Scofield, 2010)

| | Key-Value Store | Document Store | Column Oriented | Graph |
|---|---|---|---|---|
| Performance | high | high | high | variable |
| Scalability | high | variable/high | high | variable |
| Flexibility | high | high | moderate | high |
| Complexity | none | low | low | high |
| Functionality | variable | variable (low) | minimal | graph theory |

*Source:* http://www.slideshare.net/bscofield/nosql-codemash-2010
Courtesy of Ben Scofield.

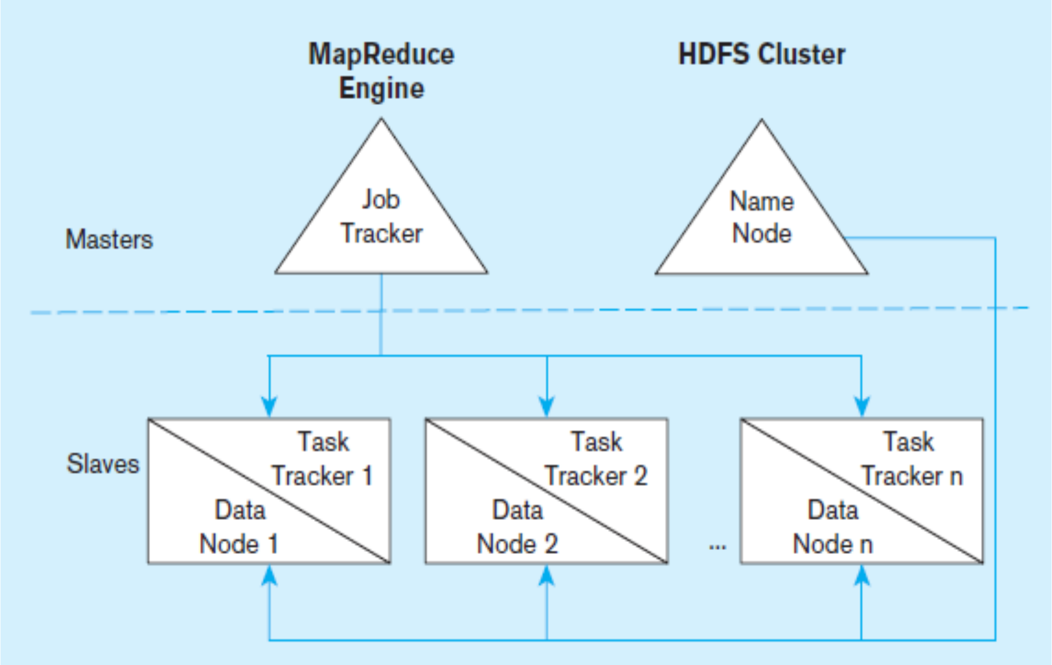
# Hadoop

- Hadoop is an open source implementation framework of MapReduce
- MapReduce is an algorithm for massive parallel processing of various types of computing tasks
- Hadoop Distributed File System (HDFS) is a file system designed for managing a large number of potentially very large files in a highly distributed environment
- Hadoop is the most talked about Big-Data data management product today
- Hadoop is a good way to take a big problem and allow many computers to work on it simultaneously

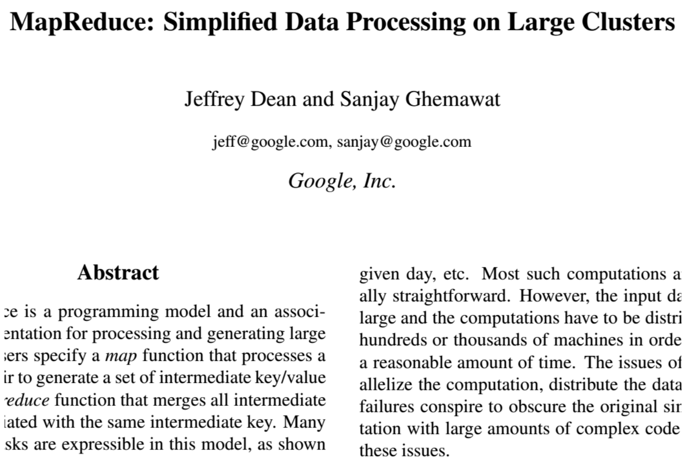
# Hadoop Distributed File System (HDFS)

- A file system, not a DBMS, not relational
- Breaks data into blocks and distributes them on various computers (nodes) throughout a Hadoop cluster
- Each cluster consists of a NameNode (master server) and some DataNodes (slaves)
- Overall control through YARN (“yet another resource allocator”)
- No updates to existing data in files, just appending to files
- “Move computation to the data”, not vice versa

# Four part figure illustrating NoSQL databases



# MapReduce



**MapReduce: Simplified Data Processing on Large Clusters**

Jeffrey Dean and Sanjay Ghemawat

jeff@google.com, sanjay@google.com

*Google, Inc.*

**Abstract**

ce is a programming model and an associ-
entation for processing and generating large
sers specify a *map* function that processes a
ir to generate a set of intermediate key/value
*educe* function that merges all intermediate
iated with the same intermediate key. Many
sks are expressible in this model, as shown

given day, etc. Most such computations a
ally straightforward. However, the input da
large and the computations have to be distri
hundreds or thousands of machines in orde
a reasonable amount of time. The issues of
allelize the computation, distribute the data
failures conspire to obscure the original sir
tation with large amounts of complex code
these issues.

hadoop

Google 2004

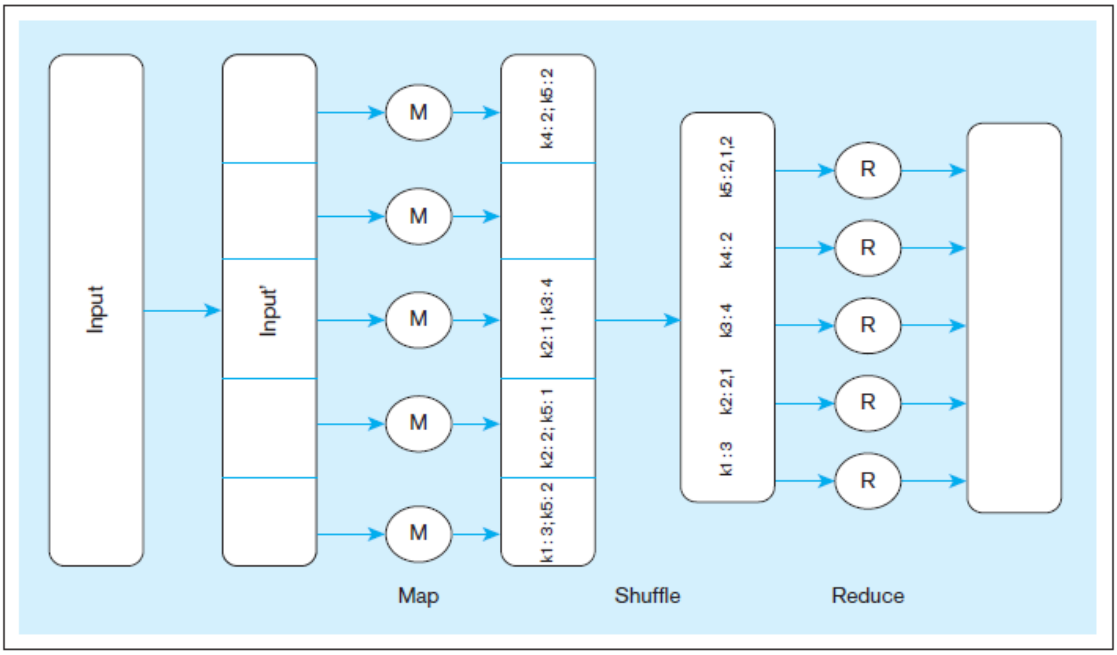
Build search index
Compute PageRank

Hadoop: Open-source at Yahoo, Facebook

# MapReduce

- **Enables parallelization of data storage and computational problem solving in an environment consisting of a large number of commodity servers**
- **Programmers don’t have to be experts at parallel processing**
- **Core idea – divide a computing task so that a multiple nodes can work on it at the same time**
- **Each node works on local data doing local processing.**
- **Two stages:**
  - **Map** stage – divide for local processing
  - **Reduce** stage – integrate the results of the individual map processes

# Schematic representation of MapReduce



MapReduce: Simplified Data Processing on Large Clusters, Jeff Dean, Sanjay Ghemawat, Google, Inc.,http://research.google.com/archive/mapreduce-osdi04-slides/index-auto-0007.html.Courtesy of the authors.

# MapReduce Programming Model

**Data type: Each record is (key, value)**

**Map function:**

$$(K_{in}, V_{in}) \rightarrow \text{list}(K_{inter}, V_{inter})$$

**Reduce function:**

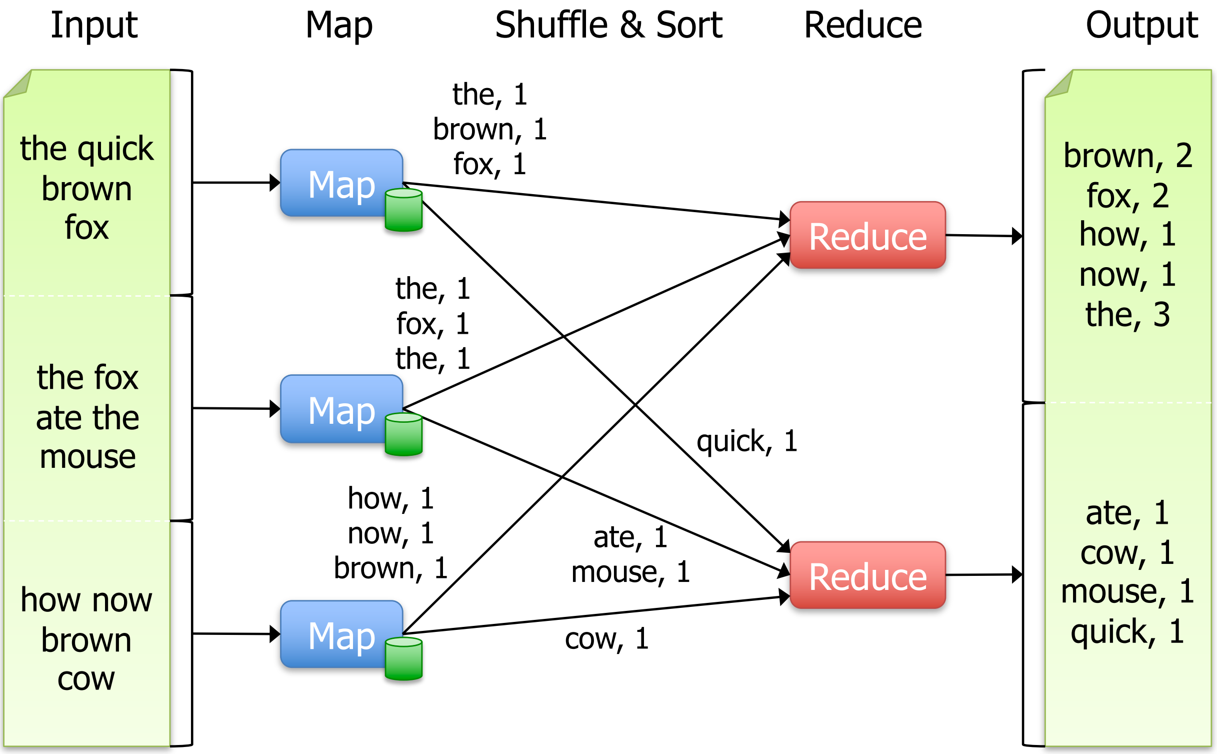
$$(K_{inter}, \text{list}(V_{inter})) \rightarrow \text{list}(K_{out}, V_{out})$$

# Example: Word Count

```
def mapper(line):
    for word in line.split():
        output(word, 1)


def reducer(key, values):
    output(key, sum(values))
```

# Word Count Execution



Input

Map

Shuffle & Sort

Reduce

Output

the quick brown fox

the fox ate the mouse

how now brown cow

Map

Map

Map

the, 1
brown, 1
fox, 1

the, 1
fox, 1
the, 1

how, 1
now, 1
brown, 1

quick, 1

ate, 1
mouse, 1

cow, 1

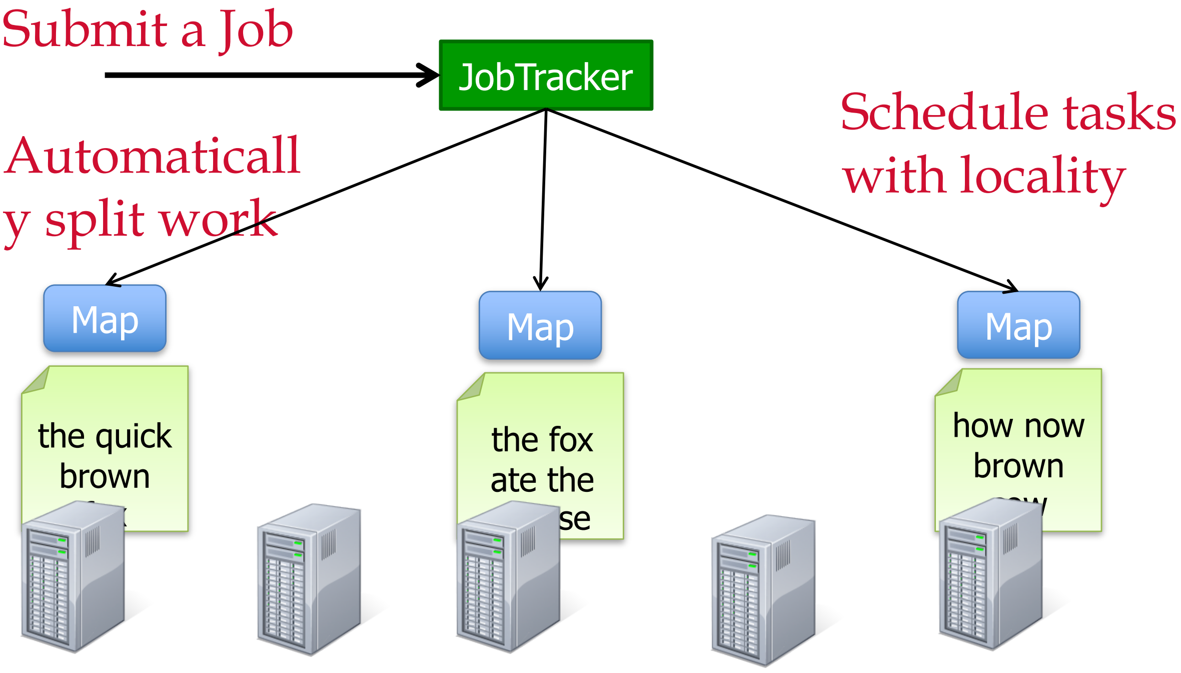
Reduce

Reduce

brown, 2
fox, 2
how, 1
now, 1
the, 3

ate, 1
cow, 1
mouse, 1
quick, 1

# Word Count Execution



Submit a Job

JobTracker

Automaticall
y split work

Schedule tasks
with locality

Map

Map

Map

the quick
brown

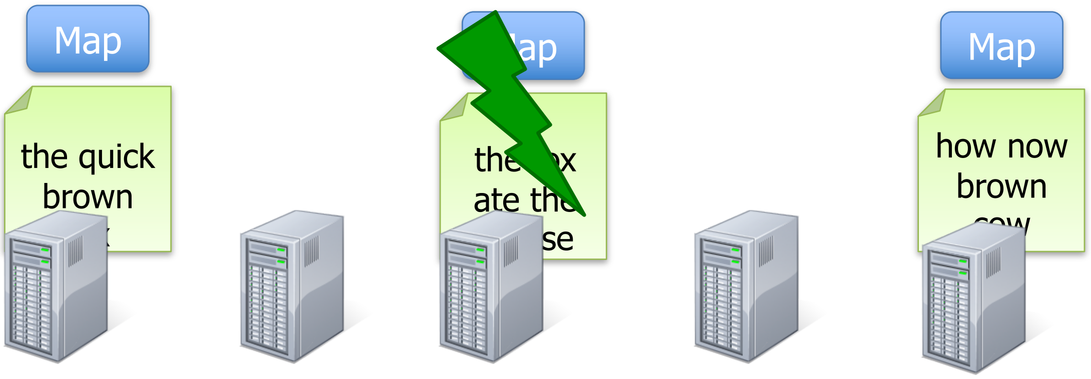
the fox
ate the
se

how now
brown

# Fault Recovery

**If a task crashes:**

- Retry on another node
- If the same task repeatedly fails, end the job

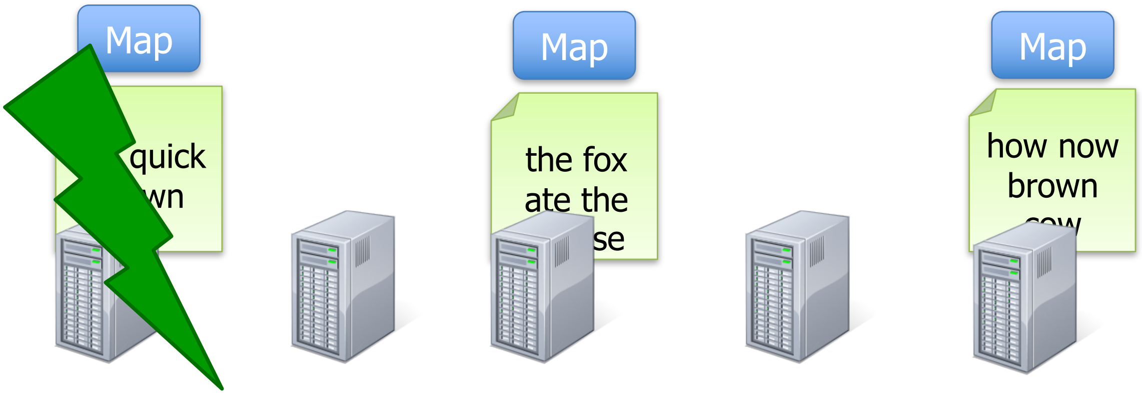


Requires user code to be **deterministic**

# Fault Recovery

**If a node crashes:**

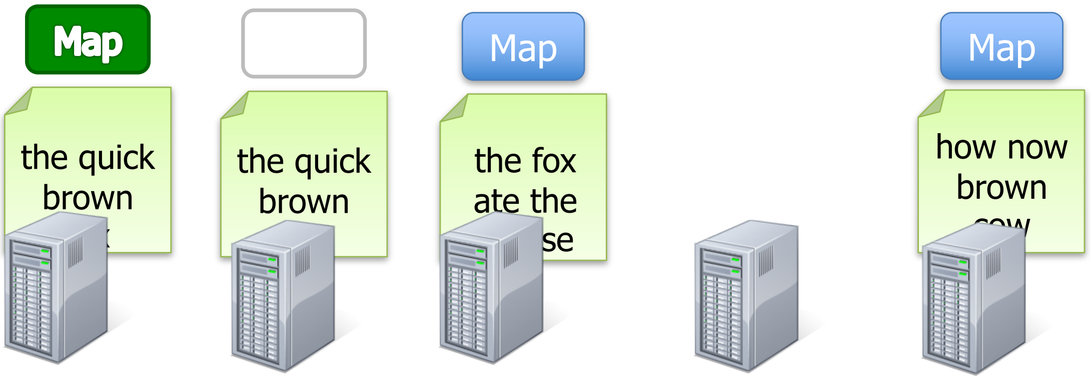
- Relaunch its current tasks on other nodes
- Relaunch tasks whose outputs were lost



# Fault Recovery

**If a task is going slowly (straggler):**

- Launch second copy of task on another node
- Take the output of whichever finishes first



# Applications

# 1. Search

**Input: (lineNumber, line) records**

**Output: lines matching a given pattern**

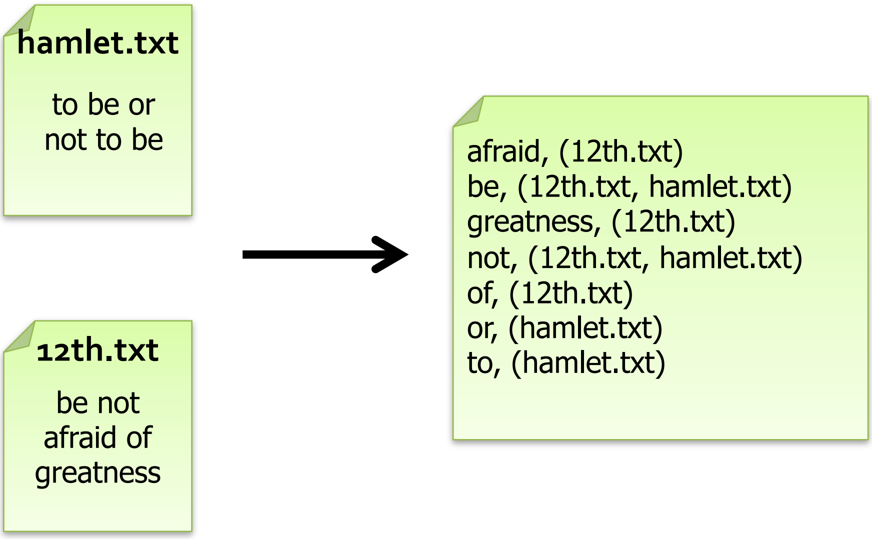
**Map:**

```
    if(line matches pattern):
        output(line)
```

**Reduce: Identity function**

- Alternative: no reducer (map-only job)

# 2. Inverted Index



**hamlet.txt**

to be or
not to be

**12th.txt**

be not
afraid of
greatness

afraid, (12th.txt)
be, (12th.txt, hamlet.txt)
greatness, (12th.txt)
not, (12th.txt, hamlet.txt)
of, (12th.txt)
or, (hamlet.txt)
to, (hamlet.txt)

# 2. Inverted Index

**Input: (filename, text) records**

**Output: list of files containing each word**

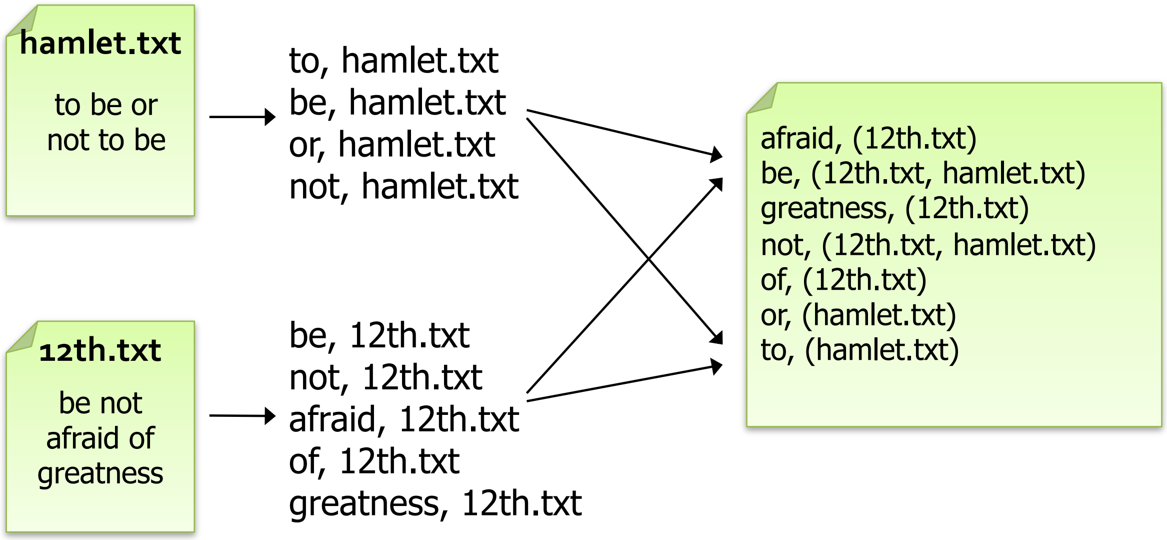
**Map:**

```
foreach word in text.split():
    output(word, filename)
```

Reduce:

```
def reduce(word, filenames):
    output(word, unique(filenames))
```

# 2.Inverted Index



**hamlet.txt**

to be or
not to be

to, hamlet.txt
be, hamlet.txt
or, hamlet.txt
not, hamlet.txt

**12th.txt**

be not
afraid of
greatness

be, 12th.txt
not, 12th.txt
afraid, 12th.txt
of, 12th.txt
greatness, 12th.txt

afraid, (12th.txt)
be, (12th.txt, hamlet.txt)
greatness, (12th.txt)
not, (12th.txt, hamlet.txt)
of, (12th.txt)
or, (hamlet.txt)
to, (hamlet.txt)

# MPI Versus MapReduce

### MPI

- **Parallel process model**
- **Fine grain control**
- **High Performance**

### MapReduce

- **High level data-parallel**
- **Automate locality, data transfers**
- **Focus on fault tolerance**

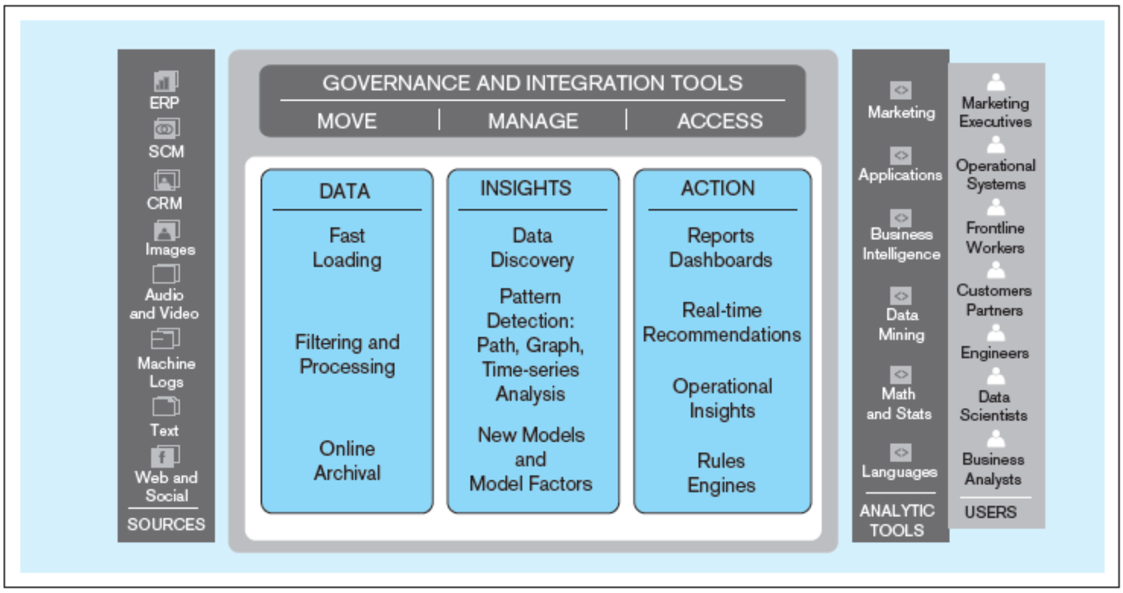
# Other Hadoop Components

- **Pig**
  - A tool that integrates a scripting language and an execution environment intended to simplify the use of MapReduce
  - Useful development tool
- **Hive**
  - An Apache project that supports the management and querying of large data sets using HiveQL, an SQL-like language that provides a declarative interface for managing data stored in Hadoop.
  - Useful for ETL tasks
- **HBase**
  - A wide-column store database that runs on top of HDFS
  - Not as popular as Cassandra

# Integrated Analytics and Data Science Platforms

- **Some vendors are bringing together traditional data warehousing and big data capabilities**
- **Examples**
  - HP HSAVEn – Hewlett Packard technologies combined with Hadoop open source and an analytics engine
  - Teradata Aster – integrate SQL, graph analysis, MapReduce, R
  - IBM Big Data Platform – combine IBM technologies with Hadoop, JSON Query Language (JAQL),DB2, Netezza

# Integrated Data Architecture



SOURCES: ERP, SCM, CRM, Images, Audio and Video, Machine Logs, Text, Web and Social

GOVERNANCE AND INTEGRATION TOOLS: MOVE | MANAGE | ACCESS

DATA: Fast Loading; Filtering and Processing; Online Archival

INSIGHTS: Data Discovery; Pattern Detection: Path, Graph, Time-series Analysis; New Models and Model Factors

ACTION: Reports Dashboards; Real-time Recommendations; Operational Insights; Rules Engines

ANALYTIC TOOLS: Marketing, Applications, Business Intelligence, Data Mining, Math and Stats, Languages

USERS: Marketing Executives, Operational Systems, Frontline Workers, Customers Partners, Engineers, Data Scientists, Business Analysts

UNIFIED DATA ARCHITECTURE, 10.14, EB 7805,http://www.teradata.com/Resources/White-Papers/Teradata-Unified-Data-Architecture-in-Action. Courtesy of Teradata Corporation

Teradata Unified Data Architecture – logical view

# Integrated Data Architecture

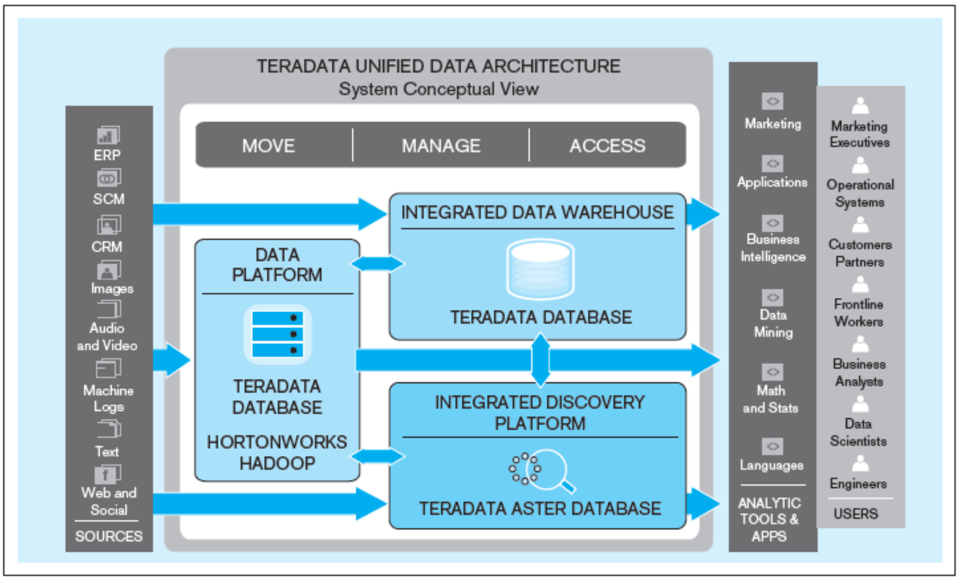


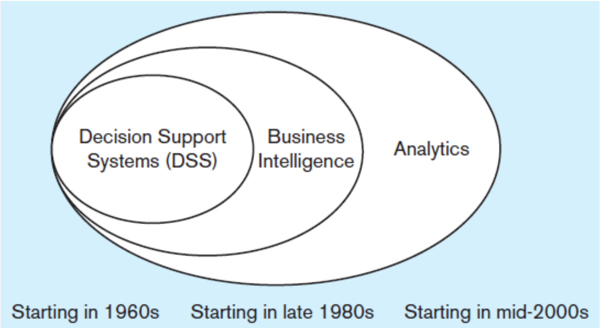
**FIGURE 11-9** **Teradata Unified Data Architecture – system conceptual view**

UNIFIED DATA ARCHITECTURE, 10.14, EB 7805,http://www.teradata.com/Resources/White-Papers/Teradata-Unified-Data-Architecture-in-Action. Courtesy of Teradata Corporation

Teradata Unified Data Architecture – system conceptual view

# Analytics

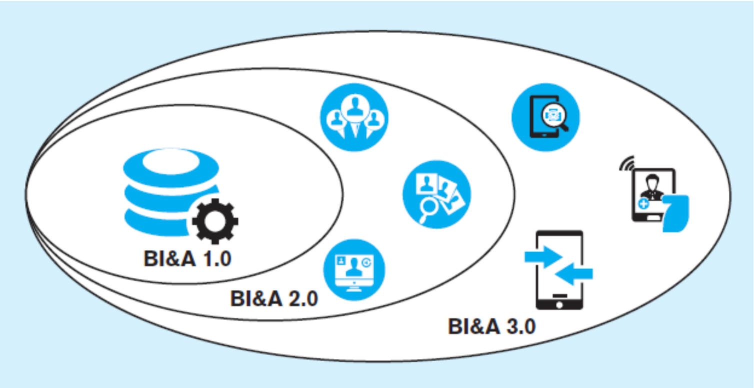
- **Historical precedents to analytics:**
  - Management information systems (MIS) ➔ Decision Support Systems (DSS) ➔ Executive Information Systems (EIS)
  - DSS idea evolved into Business Intelligence (BI)
- **Business Intelligence – a set of methodologies, processes, architectures, and technologies that transform raw data into meaningful and useful information.**
- **Analytics encompasses more than BI**
  - Umbrella term that includes BI
  - Transform data to useful form
  - Infrastructure for analysis
  - Data cleanup processes
  - User interfaces



# Types of Analytics

- **Descriptive analytics – describes the past status of the domain of interest using a variety of tools through techniques such as reporting, data visualization, dashboards, and scorecards**
- **Predictive analytics – applies statistical and computational methods and models to data regarding past and current events to predict what might happen in the future**
- **Prescriptive analytics –uses results of predictive analytics along with optimization and simulation tools to recommend actions that will lead to a desired outcome**

# Generations of Business Intelligence and Analytics



*Adapted from Chen et al., 2012*

**BI&A 1.0**

Focus on structured quantitative data largely from relational databases

**BI&A 2.0**

Include data from the Web (web interaction logs, customer reviews, social media)

**BI&A 2.0**

Include data from mobile devices, (location, sensors, etc.) as well as Internet of Things

# Use of Descriptive Analytics

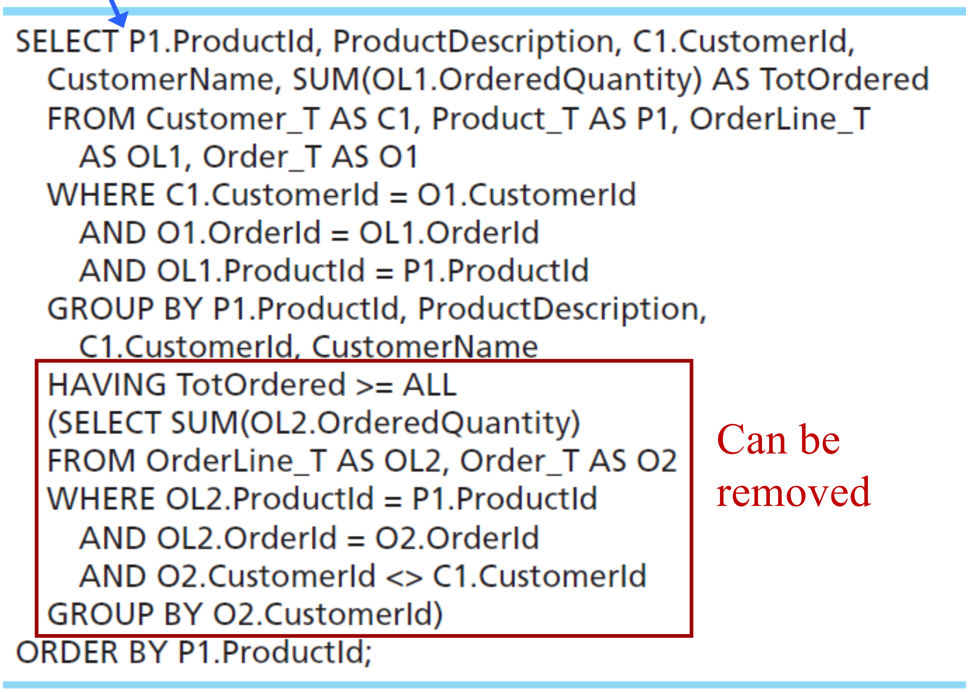
- **Descriptive analytics was the original emphasis of BI**
- **Reporting of aggregate quantitative query results**
- **Tabular or data visualization displays**
- **Dashboard – a few key indicators**
- **Scorecard – like a dashboard, but broader range**
- **OLAP – online analytical processing**

# SQL OLAP Querying

- **SQL is generally not an analytic language, but it can be used for analysis.**
- **However, OLAP extensions to SQL make this easier.**
- **OLAP queries should support:**
  - Categorization – e.g. group data by dimension characteristics
  - Aggregation – e.g. create averages per category
  - Ranking – e.g. find customer in some category with highest average monthly sales

# Regular SQL Query

TOP 1

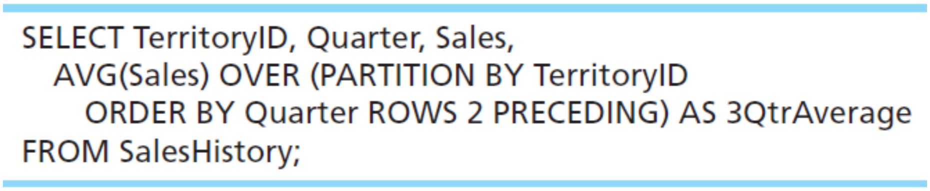


```
SELECT P1.ProductId, ProductDescription, C1.CustomerId,
   CustomerName, SUM(OL1.OrderedQuantity) AS TotOrdered
   FROM Customer_T AS C1, Product_T AS P1, OrderLine_T
      AS OL1, Order_T AS O1
   WHERE C1.CustomerId = O1.CustomerId
      AND O1.OrderId = OL1.OrderId
      AND OL1.ProductId = P1.ProductId
   GROUP BY P1.ProductId, ProductDescription,
      C1.CustomerId, CustomerName
   HAVING TotOrdered >= ALL
   (SELECT SUM(OL2.OrderedQuantity)
   FROM OrderLine_T AS OL2, Order_T AS O2
   WHERE OL2.ProductId = P1.ProductId
      AND OL2.OrderId = O2.OrderId
      AND O2.CustomerId <> C1.CustomerId
   GROUP BY O2.CustomerId)
ORDER BY P1.ProductId;
```

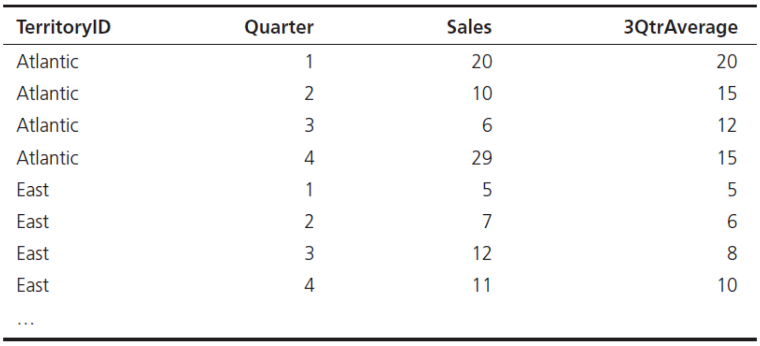
Can be removed

# OLAP SQL Query

Consider a SalesHistory table (columns TerritoryID, Quarter, and Sales) and the desire to show a three-quarter moving average of sales.



```
SELECT TerritoryID, Quarter, Sales,
   AVG(Sales) OVER (PARTITION BY TerritoryID
      ORDER BY Quarter ROWS 2 PRECEDING) AS 3QtrAverage
FROM SalesHistory;
```



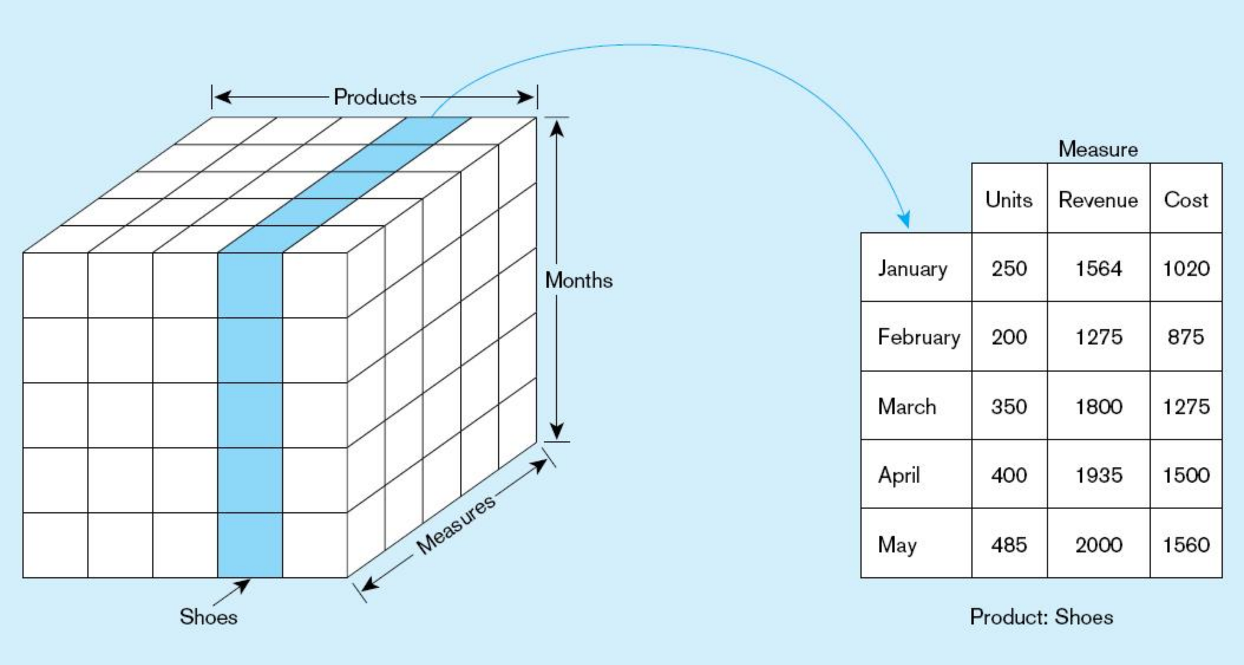
| TerritoryID | Quarter | Sales | 3QtrAverage |
|---|---|---|---|
| Atlantic | 1 | 20 | 20 |
| Atlantic | 2 | 10 | 15 |
| Atlantic | 3 | 6 | 12 |
| Atlantic | 4 | 29 | 15 |
| East | 1 | 5 | 5 |
| East | 2 | 7 | 6 |
| East | 3 | 12 | 8 |
| East | 4 | 11 | 10 |
| … | | | |

OVER (also called WINDOW) is a special clause that provide a “sliding view” of rows from a query. PARTITION BY is like a GROUP by for OVER.

# Online Analytical Processing (OLAP) Tools

- **Online Analytical Processing (OLAP) -- 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) – OLAP tools that view the database as a traditional relational database in either a star schema or other normalized or denormalized set of tables**
- **Multidimensional OLAP (MOLAP) –OLAP tools that load data into an intermediate structure, usually a three- or higher-dimensional array.**

# Slicing a data cube



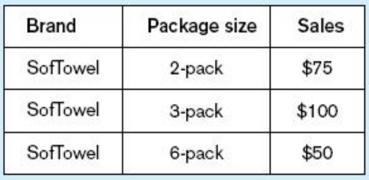
| | Measure | | |
|---|---|---|---|
| | Units | Revenue | Cost |
| January | 250 | 1564 | 1020 |
| February | 200 | 1275 | 875 |
| March | 350 | 1800 | 1275 |
| April | 400 | 1935 | 1500 |
| May | 485 | 2000 | 1560 |

Product: Shoes

Slicing, dicing, pivoting, and drill-down are useful cube operations

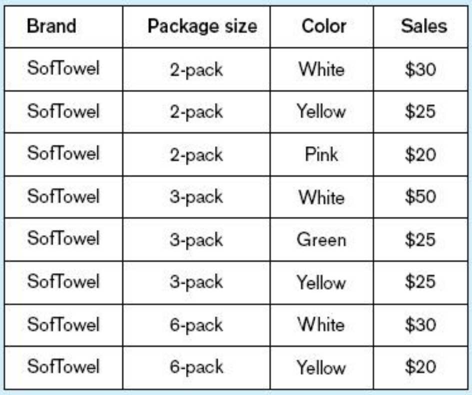
# Example of drill-down

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



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

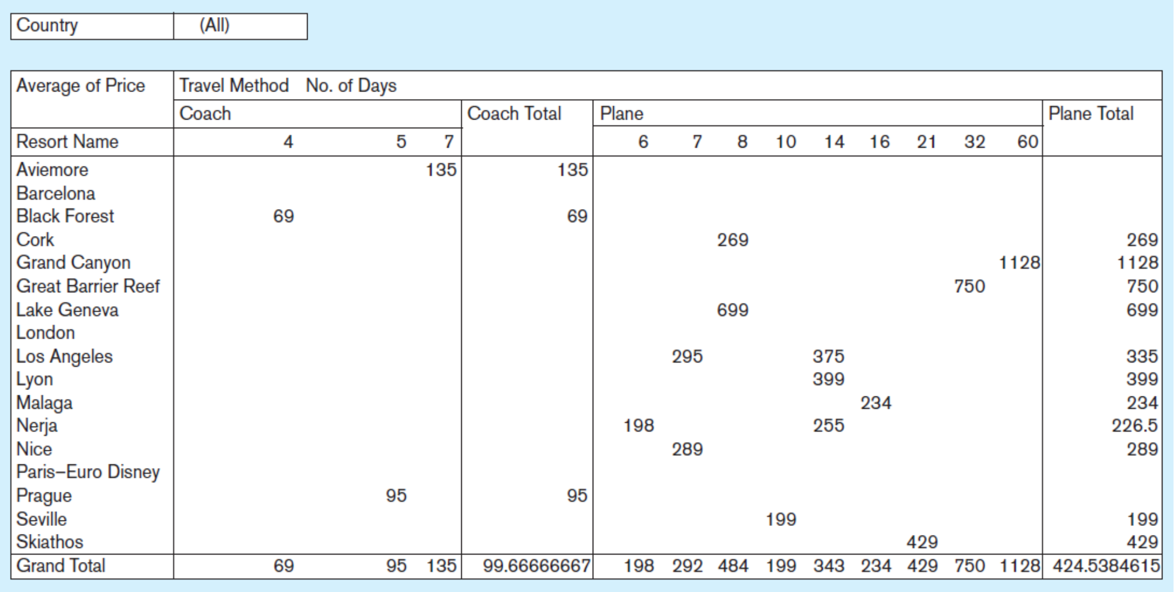
Summary report



| 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 |

Drill-down with color added

# Sample pivot table with four dimensions: Country (pages), Resort Name (rows), Travel Method, and No. of Days (columns)



| Country | (All) |
|---|---|

| Average of Price | Travel Method No. of Days | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | Coach | | | Coach Total | Plane | | | | | | | | | Plane Total |
| Resort Name | 4 | 5 | 7 | | 6 | 7 | 8 | 10 | 14 | 16 | 21 | 32 | 60 | |
| Aviemore | | | 135 | 135 | | | | | | | | | | |
| Barcelona | | | | | | | | | | | | | | |
| Black Forest | 69 | | | 69 | | | | | | | | | | |
| Cork | | | | | | | 269 | | | | | | | 269 |
| Grand Canyon | | | | | | | | | | | | | 1128 | 1128 |
| Great Barrier Reef | | | | | | | | | | | | 750 | | 750 |
| Lake Geneva | | | | | | | 699 | | | | | | | 699 |
| London | | | | | | | | | | | | | | |
| Los Angeles | | | | | | 295 | | | 375 | | | | | 335 |
| Lyon | | | | | | | | | 399 | | | | | 399 |
| Malaga | | | | | | | | | | 234 | | | | 234 |
| Nerja | | | | | 198 | | | | 255 | | | | | 226.5 |
| Nice | | | | | | 289 | | | | | | | | 289 |
| Paris–Euro Disney | | | | | | | | | | | | | | |
| Prague | | 95 | | 95 | | | | | | | | | | |
| Seville | | | | | | | | 199 | | | | | | 199 |
| Skiathos | | | | | | | | | | | 429 | | | 429 |
| Grand Total | 69 | 95 | 135 | 99.66666667 | 198 | 292 | 484 | 199 | 343 | 234 | 429 | 750 | 1128 | 424.5384615 |

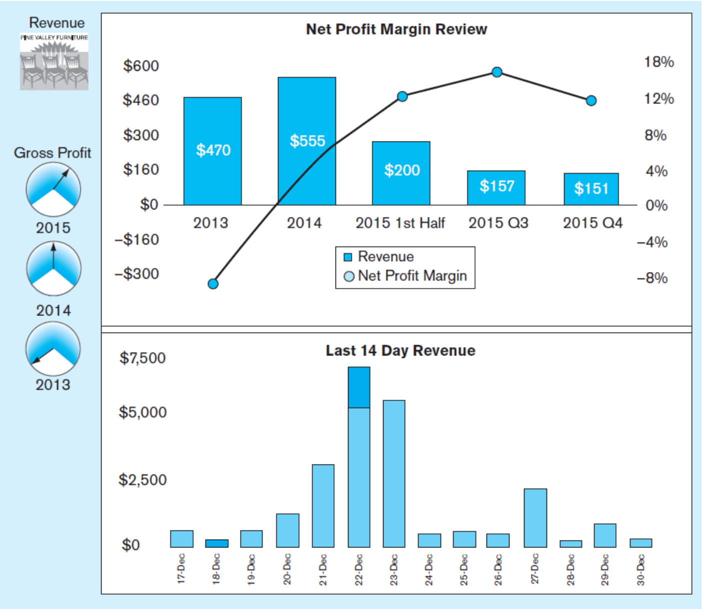
Although the screen is only two dimensions, you can include more dimensions by combining multiple in a row or column, and by including paging

# Data Visualization

- **Representation of data in graphical and multimedia formats for human analysis**
- **"A picture tells a thousand words"**
- **Without showing precise values, graphs and charts can depict relationships in the data**
- **Often used in dashboards, as shown in next slide**

# Business Performance Mgmt (BPM): Sample Dashboard

- **BPM systems allow managers to:**
  - Measure,
  - Monitor
  - Manage key activities and processes to achieve organizational goals
- **Dashboards are often used to provide an information system in support of BPM.**



*Charts like these are examples of data visualization, the representation of data in graphical and multimedia formats for human analysis.*

# Predictive Analytics

- **Statistical and computational methods that use data regarding past and current events to form models regarding what might happen in the future**
- **Examples: classification trees, linear and logistic regression analysis, machine learning, neural networks, time series analysis, Bayesian modeling**

# Data Mining Tools

- **Knowledge discovery using a sophisticated blend of techniques from traditional statistics, artificial intelligence, and computer graphics**
- **Goals:**
  - Explanatory – explain observed events or conditions
  - Confirmatory – confirm hypotheses
  - Exploratory –analyze data for new or unexpected relationships
- **Text mining – Discovering meaningful information algorithmically based on computational analysis of unstructured textual information**

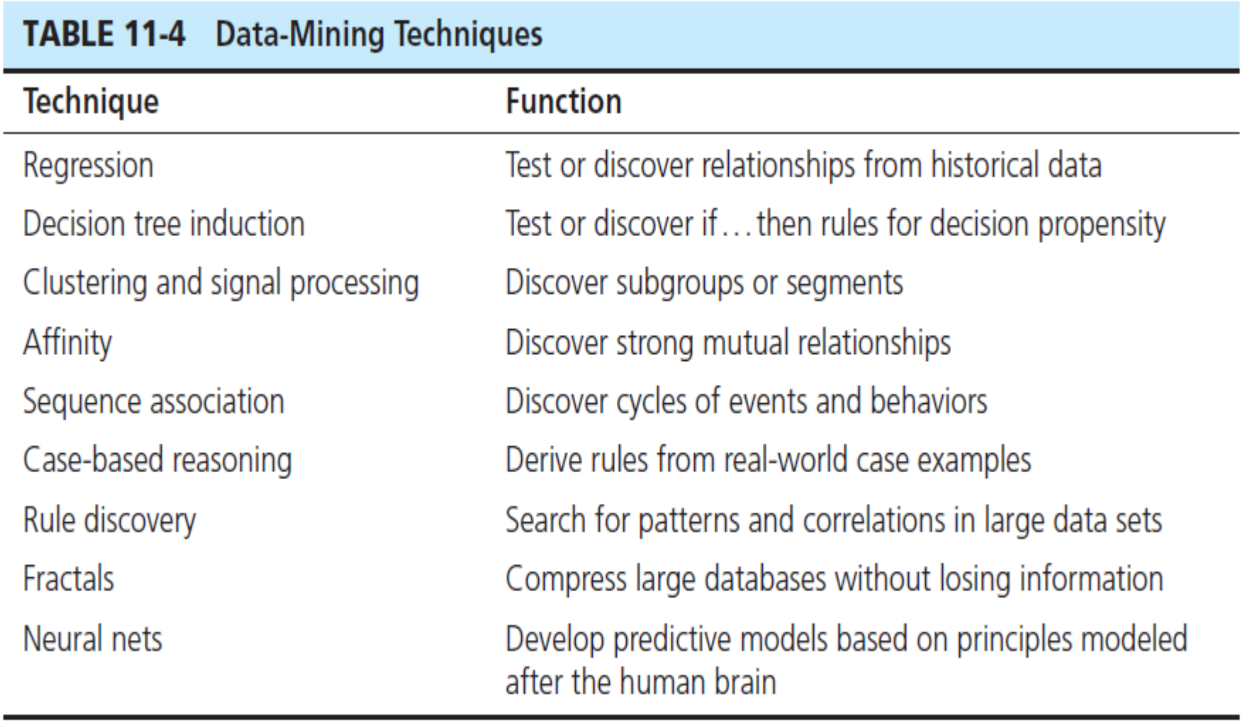


TABLE 11-4 Data-Mining Techniques

| Technique | Function |
|---|---|
| Regression | Test or discover relationships from historical data |
| Decision tree induction | Test or discover if . . . then rules for decision propensity |
| Clustering and signal processing | Discover subgroups or segments |
| Affinity | Discover strong mutual relationships |
| Sequence association | Discover cycles of events and behaviors |
| Case-based reasoning | Derive rules from real-world case examples |
| Rule discovery | Search for patterns and correlations in large data sets |
| Fractals | Compress large databases without losing information |
| Neural nets | Develop predictive models based on principles modeled after the human brain |

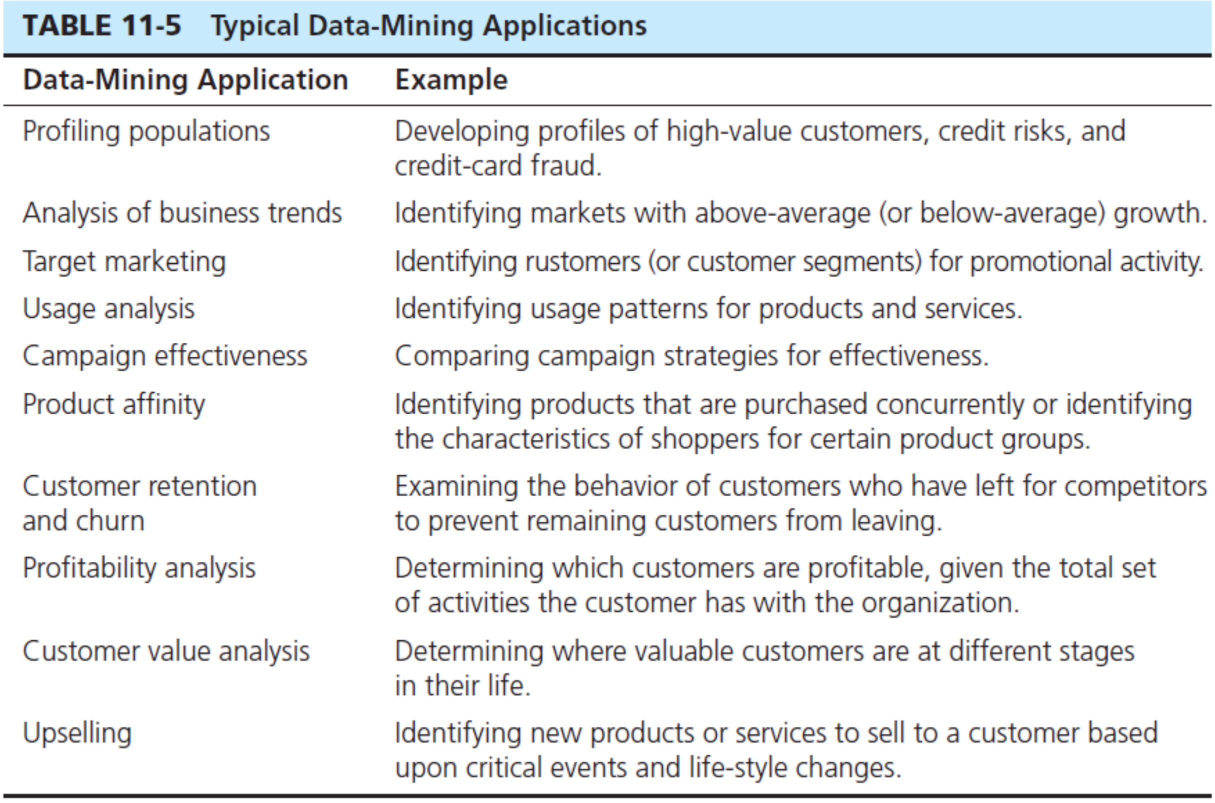


TABLE 11-5 Typical Data-Mining Applications

| Data-Mining Application | Example |
|---|---|
| Profiling populations | Developing profiles of high-value customers, credit risks, and credit-card fraud. |
| Analysis of business trends | Identifying markets with above-average (or below-average) growth. |
| Target marketing | Identifying rustomers (or customer segments) for promotional activity. |
| Usage analysis | Identifying usage patterns for products and services. |
| Campaign effectiveness | Comparing campaign strategies for effectiveness. |
| Product affinity | Identifying products that are purchased concurrently or identifying the characteristics of shoppers for certain product groups. |
| Customer retention and churn | Examining the behavior of customers who have left for competitors to prevent remaining customers from leaving. |
| Profitability analysis | Determining which customers are profitable, given the total set of activities the customer has with the organization. |
| Customer value analysis | Determining where valuable customers are at different stages in their life. |
| Upselling | Identifying new products or services to sell to a customer based upon critical events and life-style changes. |

*Source:* Based on Dyché (2000).

# KNIME Example of Predictive Analytics

- **Credit scoring**
  - Takes past financial data to produce a credit score
  - Starts with decision trees, neural networks, and support vector machine (SVM) algorithms for initial model
  - Next uses Predictive Modeling Markup Language (PMML)
- **Marketing**
  - Churn analysis – predicting which customers will leave using clustering via k-means algorithm
  - Social media analysis using association rules

# Use of Prescriptive Analytics

- **Use of optimization and simulation tools for prescribing the best action to take**
- **Example applications**
  - Making trading decisions in securities and stock market
  - Making pricing decisions for airlines and hotels
  - Making product recommendations (e.g. Amazon and Netflix)
- **Often requires predictive analytics and game theory**

# Analytics Data Management Infrastructure

- **Important criteria: scalability, parallelism, low latency, and data optimization**
- **These criteria ensure speed, availability, and access**

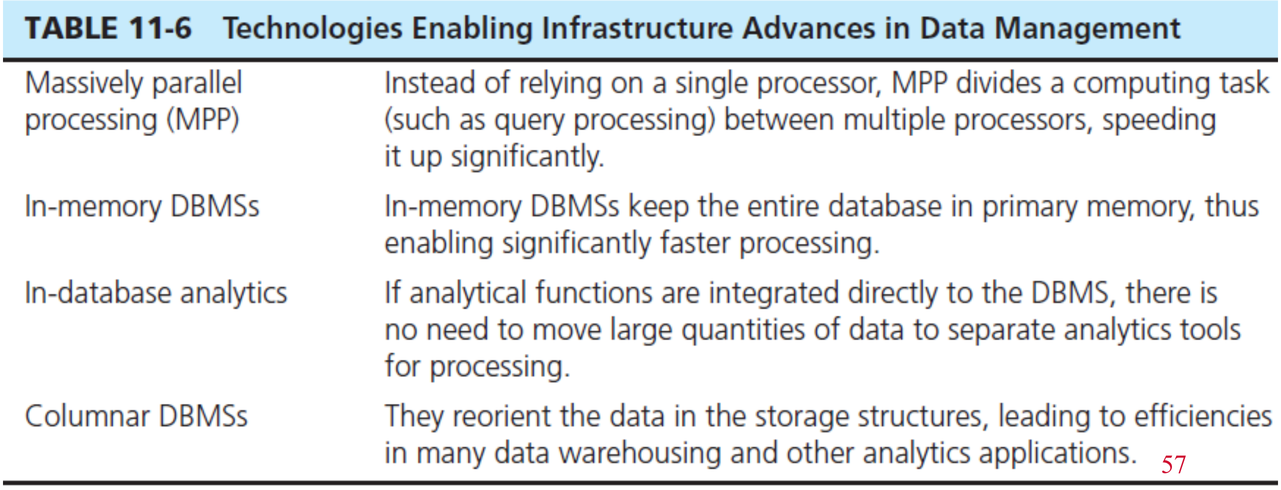


**TABLE 11-6 Technologies Enabling Infrastructure Advances in Data Management**

| | |
|---|---|
| Massively parallel processing (MPP) | Instead of relying on a single processor, MPP divides a computing task (such as query processing) between multiple processors, speeding it up significantly. |
| In-memory DBMSs | In-memory DBMSs keep the entire database in primary memory, thus enabling significantly faster processing. |
| In-database analytics | If analytical functions are integrated directly to the DBMS, there is no need to move large quantities of data to separate analytics tools for processing. |
| Columnar DBMSs | They reorient the data in the storage structures, leading to efficiencies in many data warehousing and other analytics applications. |

# Big Data and Analytics Impact: Applications

- **Business**
- **E-government and politics**
- **Science and technology**
- **Smart health and well-being**
- **Security and public safety**

# Big Data and Analytics Impact: Social Implications

- **Personal privacy vs. collective benefit**
- **Ownership and access**
- **Data/algorithm quality and reuse**
- **Transparency and validation**
- **Demands for workforce capabilities and education**

*Slides adapted from: Jeff Hoffer, Ramesh Venkataraman, Heikki Topi*