Chapter 14
Big Data Analytics and NoSQL

Learning Objectives

- In this chapter, you will learn:
  - What Big Data is and why it is important in modern business
  - The primary characteristics of Big Data and how these go beyond the traditional "3 Vs"
  - How the core components of the Hadoop framework, HDFs and MapReduce operate
  - What the major components of the Hadoop ecosystems are

Learning Objectives

- In this chapter, you will learn:
  - The four major approaches of the NoSQL data model and how the differ from the relational model
  - About data analytics, including data mining and predictive analytics

Figure 14.2 – Current View of Big Data

FIGURE 14.2 CURRENT VIEW OF BIG DATA

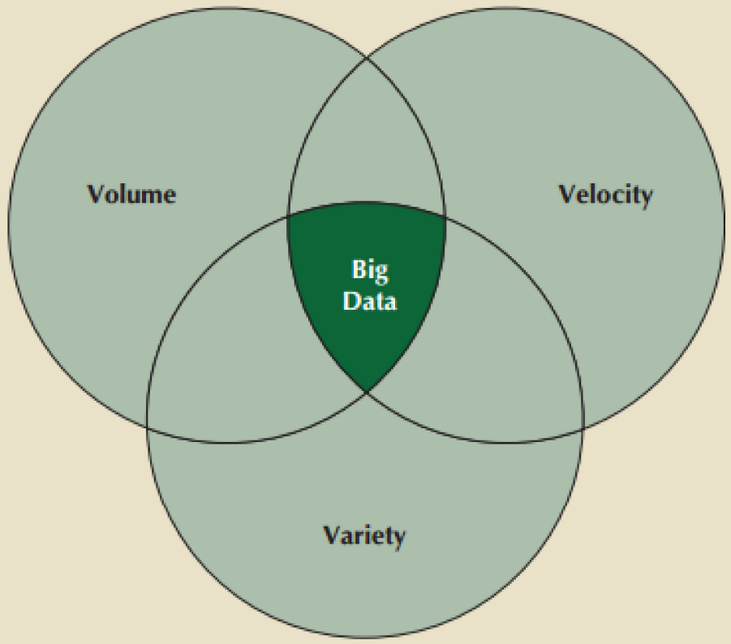


TABLE 14.1

STORAGE CAPACITY UNITS

TERM	CAPACITY	ABBREVIATION
Bit	0 or 1 value	b
Byte	8 bits	B
Kilobyte	1024* bytes	KB
Megabyte	1024 KB	MB
Gigabyte	1024 MB	GB
Terabyte	1024 GB	TB
Petabyte	1024 TB	PB
Exabyte	1024 PB	EB
Zettabyte	1024 EB	ZB
Yottabyte	1024 ZB	YB

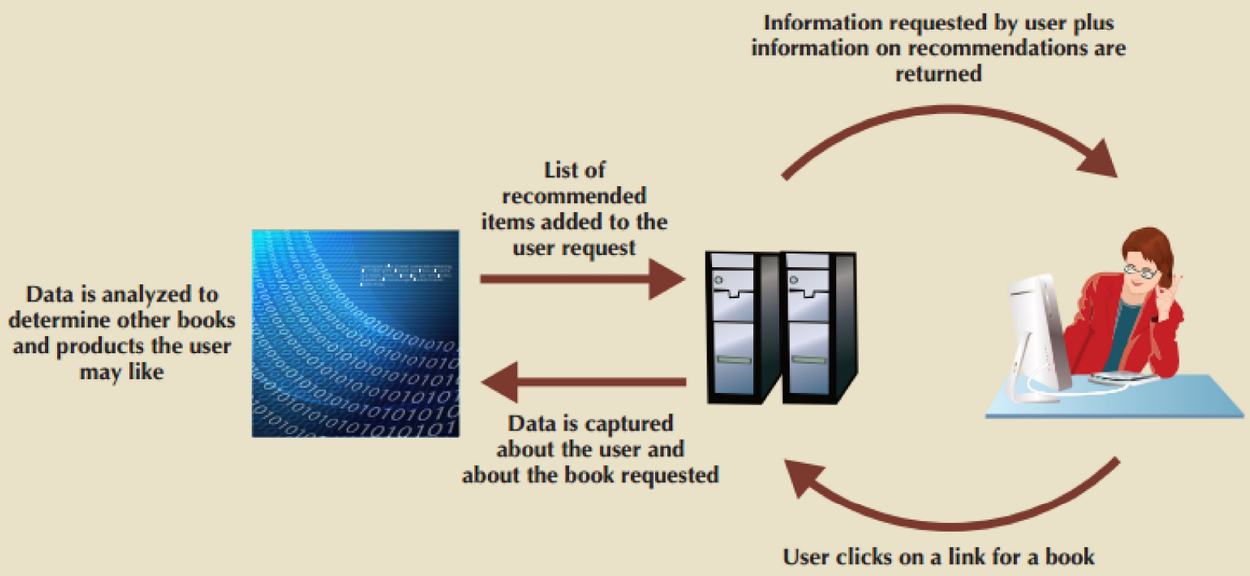
* Note that because bits are binary in nature and are the basis on which all other storage values are based, all values for data storage units are defined in terms of powers of 2. For example, the prefix *kilo* typically means 1000; however, in data storage, a kilobyte = 2^{10} = 1024 bytes.

Big Data

- **Volume**: Quantity of data to be stored
  - **Scaling up** is keeping the same number of systems but migrating each one to a larger system
  - **Scaling out** means when the workload exceeds server capacity, it is spread out across a number of servers
- **Velocity:** Speed at which data is entered into system and must be processed
  - **Stream processing** focuses on input processing and requires analysis of data stream as it enters the system
  - **Feedback loop processing** refers to the analysis of data to produce actionable results

Figure 14.3 – Feedback Loop Processing

FIGURE 14.3 FEEDBACK LOOP PROCESSING



Big Data

- **Variety**: Variations in the structure of data to be stored
  - **Structured data** fits into a predefined data model
  - **Unstructured data** dies not fit into a predefined model
- Other characteristics:
  - **Variability:** Changes in meaning of data based on context
    - **Sentimental analysis** attempts to determine attitude
  - **Veracity:** Trustworthiness of data
  - **Value:** Degree data can be analyzed for meaningful insight
  - **Visualization:** Ability to graphically resent data to make it understandable to users

Big Data

- Characteristics important in working with data in relational models are universal and also apply to Big Data
- Relational databases not necessarily best for storing and managing all organizational data
  - **Polyglot persistence:** Coexistence of a variety of data storage and management technologies within an organization's infrastructure

Hadoop

- De facto standard for most Big Data storage and processing
- Java-based framework for distributing and processing very large data sets across clusters of computers
- Most important components:
  - **Hadoop Distributed File System (HDFS)**: Low-level distributed file processing system that can be used directly for data storage
  - **MapReduce:** Programming model that supports processing large data sets

Hadoop Distributed File System (HDFS)

- Approach based on several key assumptions:
  - *High volume* - Default block sizes is 64 MB and can be configured to even larger values
  - *Write-once, read-many* - Model simplifies concurrent issues and improves data throughput
  - *Streaming access* - Hadoop is optimized for batch processing of entire files as a continuous stream of data
  - *Fault tolerance* – HDFS is designed to replicate data across many different devices so that when one fails, data is still available from another device

MapReduce

- Framework used to process large data sets across clusters
  - Breaks down complex tasks into smaller subtasks, performing the subtasks and producing a final result
  - **Map** function takes a collection of data and sorts and filters it into a set of key-value pairs
    - **Mapper** program performs the map function
  - **Reduce** summaries results of map function to produce a single result
    - **Reducer** program performs the reduce function

14.5 MAPREDUCE

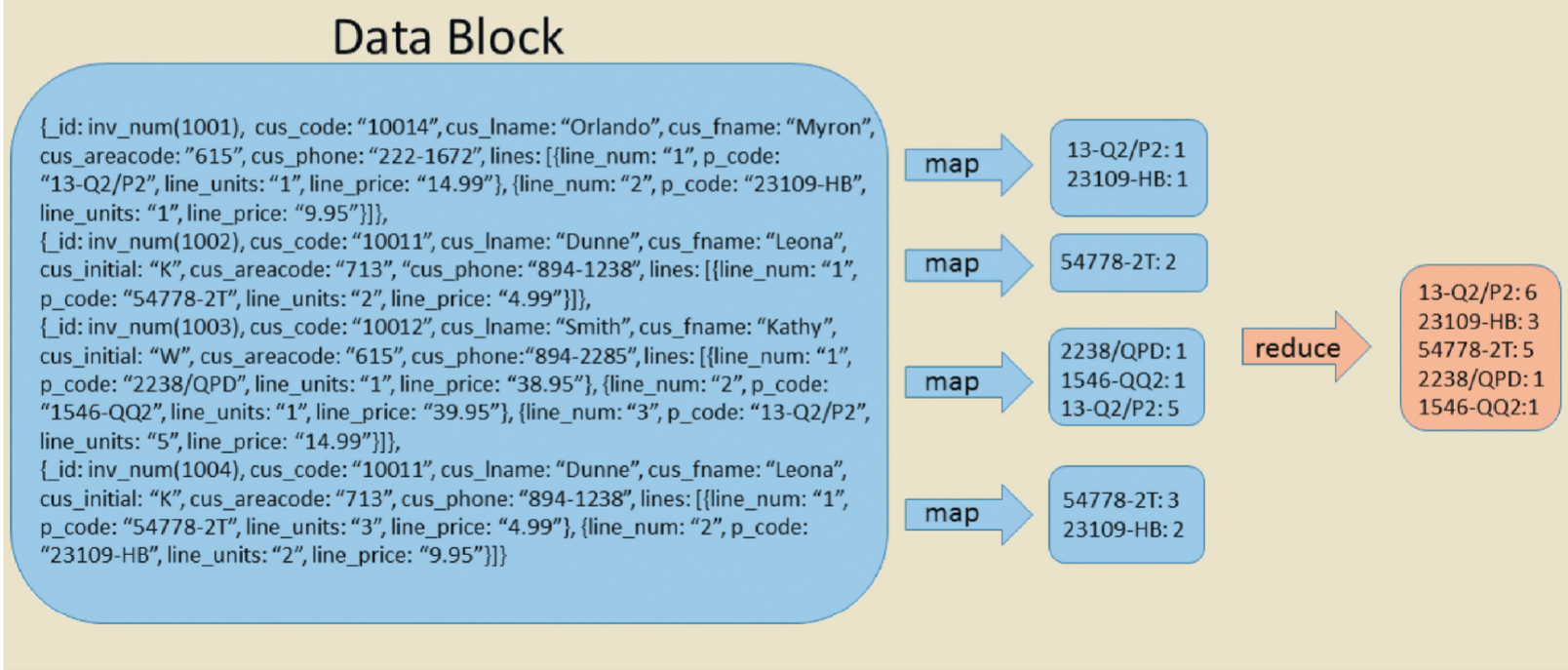
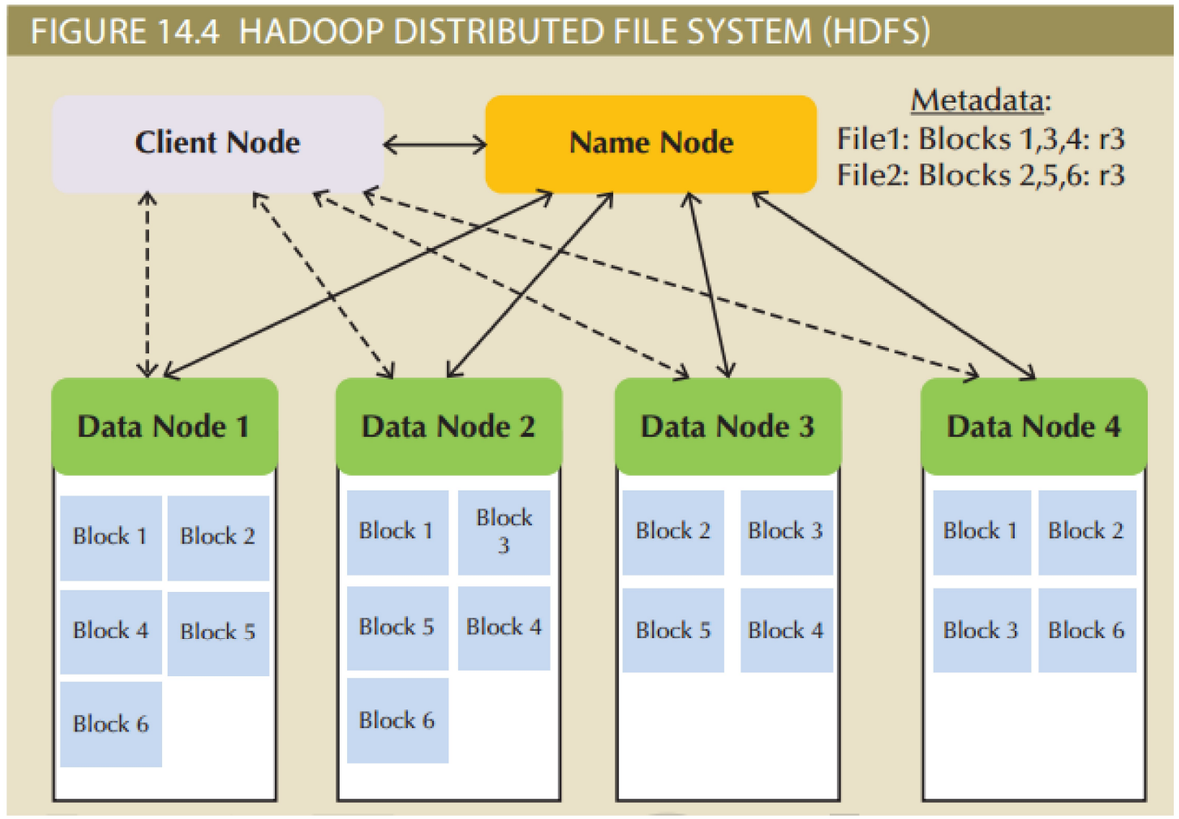


Figure 14.4 – Hadoop Distributed File System (HDFS)



Hadoop Distributed File System (HDFS)

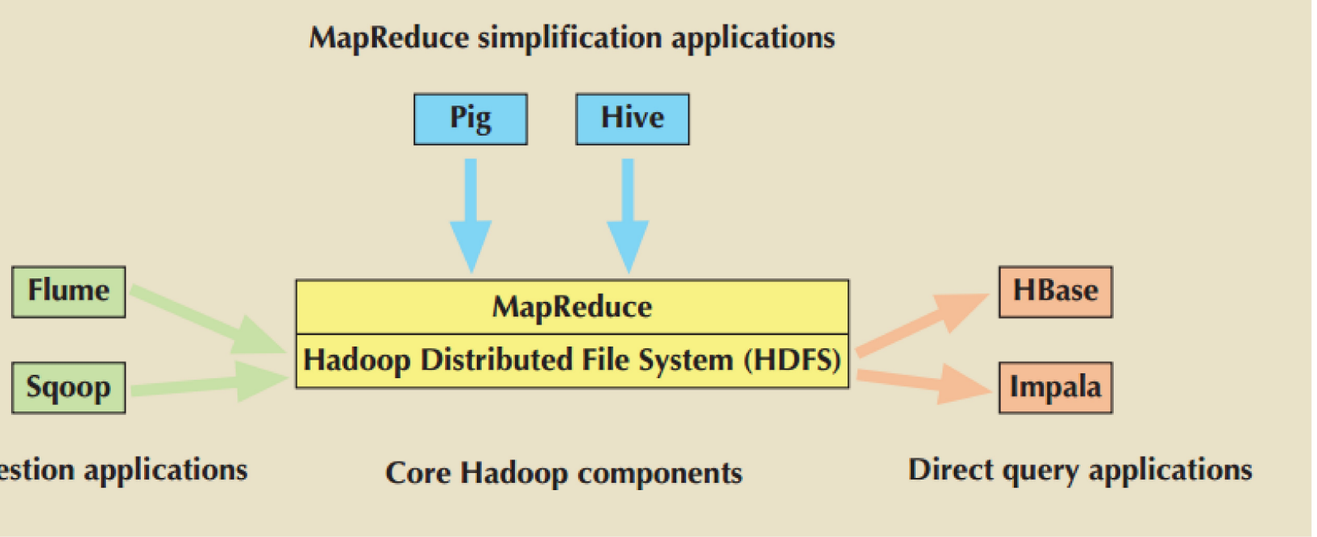
- Uses several types of nodes (computers):
  - Data node store the actual file data
  - Name node contains file system metadata
  - Client node makes requests to the file system as needed to support user applications
  - Data node communicates with name node by regularly sending **block reports** and **heartbeats**

MapReduce

- Implementation complements HDFS structure
- Uses a **job tracker** or central control program to accept, distribute, monitor and report on jobs in a Hadoop environment
- **Task tracker** is a program in MapReduce responsible for reducing tasks on a node
- System uses **batch processing** which runs tasks from beginning to end with no user interaction

Figure 14.6 – A Sample of the Hadoop Ecosystem

FIGURE 14.6 A SAMPLE OF THE HADOOP ECOSYSTEM



Hadoop Ecosystem

- Map Reduce Simplification Applications:
  - *Hive* is a data warehousing system that sites on top of HDFS and supports its own SQL-like language
  - *Pig* compiles a high-level scripting language (Pig Latin) into MapReduce jobs for executing in Hadoop
- Data Ingestion Applications:
  - *Flume* is a component for ingesting data in Hadoop
  - *Sqoop* is a tool for converting data back and forth between a relational database and the HDFS

Hadoop Ecosystem

- Direct Query Applications:
  - *HBase* is a column-oriented NoSQL database designed to sit on top of the HDFS that quickly processes sparse datasets
  - *Impala* was the first SQL-on-Hadoop application

NoSQL

- Name given to non-relational database technologies developed to address Big Data challenges
- **Key-value (KV) databases** store data as a collection of key-value pairs organized as **buckets** which are the equivalent of tables
- **Document databases** store data in key-value pairs in which the value components are tag-encoded documents grouped into logical groups called **collections**

NoSQL

- **Column-oriented databases** refers to two technologies:
  - **Column-centric storage**: Data stored in blocks which hold data from a single column across many rows
  - **Row-centric storage:** Data stored in block which hold data from all columns of a given set of rows
- **Graph databases** store data on relationship-rich data as a collection of **nodes** and **edges**
  - **Properties** are the attributes of a node or edge of interest to a user
- **Traversal** is a query in a graph database

TABLE 14.2

NoSQL DATABASES

NoSQL CATEGORY	EXAMPLE DATABASES
Key-value database	Dynamo Riak Redis Voldemort
Document databases	MongoDB CouchDB OrientDB RavenDB
Column-oriented databases	HBase Cassandra Hypertable
Graph databases	Neo4J ArangoDB GraphBase

Figure 14.7- Key-Value Database Storage

FIGURE 14.7 KEY-VALUE DATABASE STORAGE

Bucket = Customer

Key	Value
10010	"LName Ramas FName Alfred Initial A Areacode 615 Phone 844-2573 Balance 0"
10011	"LName Dunne FName Leona Initial K Areacode 713 Phone 894-1238 Balance 0"
10014	"LName Orlando FName Myron Areacode 615 Phone 222-1672 Balance 0"

Figure 14.8- Document Database Tagged Format

FIGURE 14.8 DOCUMENT DATABASE TAGGED FORMAT

Collection = Customer

Key	Document
10010	{LName: "Ramas", FName: "Alfred", Initial: "A", Areacode: "615", Phone: "844-2573", Balance: "0"}
10011	{LName: "Dunne", FName: "Leona", Initial: "K", Areacode: "713", Phone: "894-1238", Balance: "0"}
10014	{LName: "Orlando", FName: "Myron", Areacode: "615", Phone: "222-1672", Balance: "0"}

Figure 14.9- Comparison of Row-Centric and Column-Centric Storage

FIGURE 14.9 COMPARISON OF ROW-CENTRIC AND COLUMN-CENTRIC STORAGE

CUSTOMER relational table

Cus_Code	Cus_LName	Cus_FName	Cus_City	Cus_State
10010	Ramas	Alfred	Nashville	TN
10011	Dunne	Leona	Miami	FL
10012	Smith	Kathy	Boston	MA
10013	Olowski	Paul	Nashville	TN
10014	Orlando	Myron		
10015	O'Brian	Amy	Miami	FL
10016	Brown	James		
10017	Williams	George	Mobile	AL
10018	Farriss	Anne	Opp	AL
10019	Smith	Olette	Nashville	TN

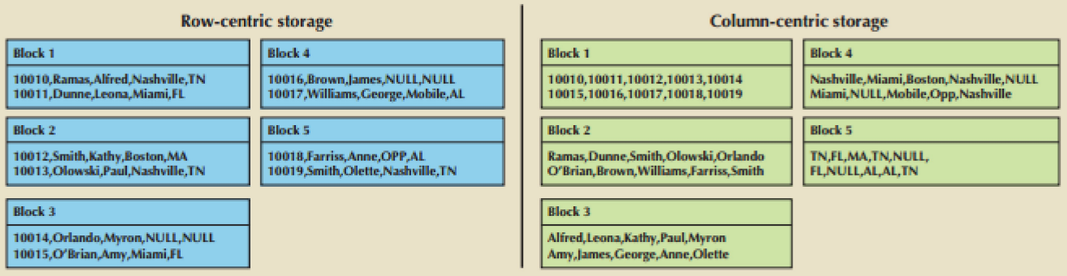
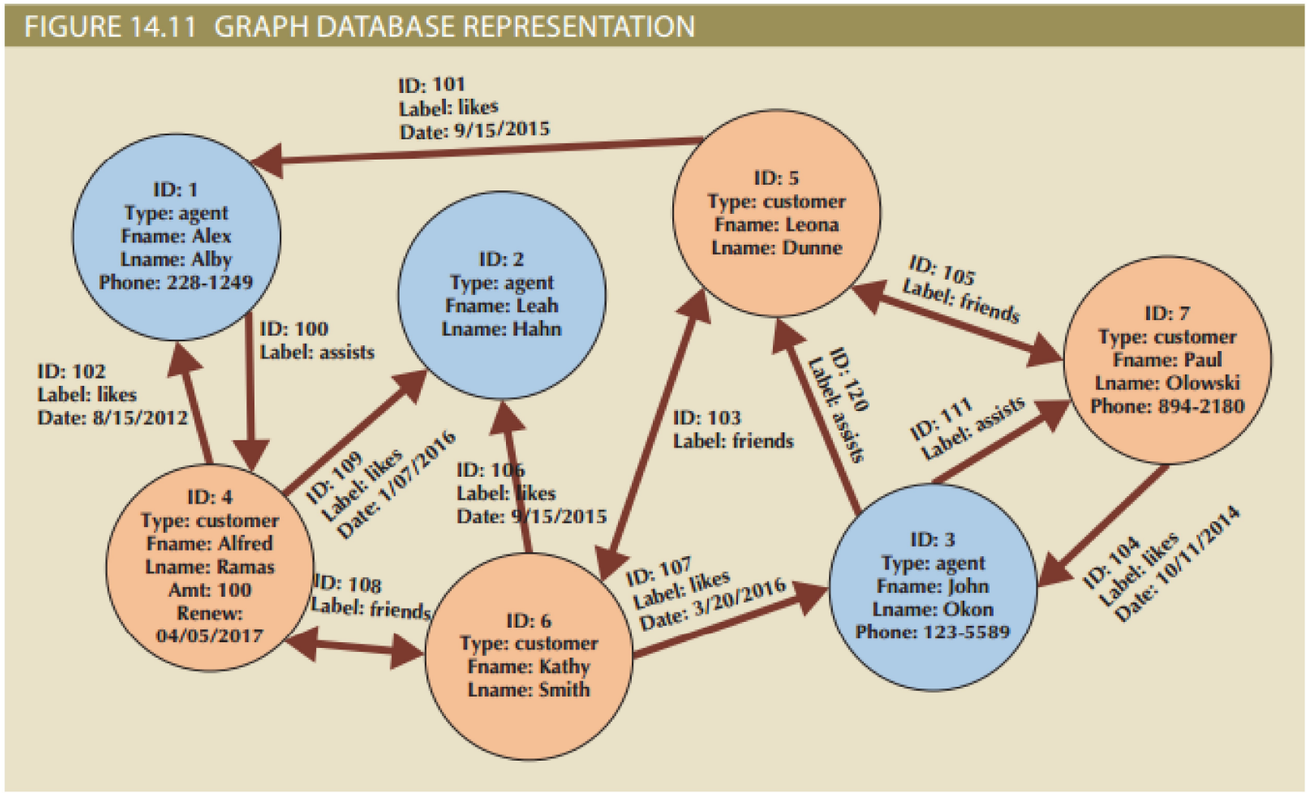


Figure 14.10- Graph Database Representation

FIGURE 14.11 GRAPH DATABASE REPRESENTATION



NewSQL Databases

- Database model that attempts to provide ACID-compliant transactions across a highly distributed infrastructure
  - Latest technologies to appear in the data management area to address Big Data problems
  - No proven track record
  - Have been adopted by relatively few organizations

Figure 14.12- Extracting Knowledge From Data

FIGURE 14.12 EXTRACTING KNOWLEDGE FROM DATA

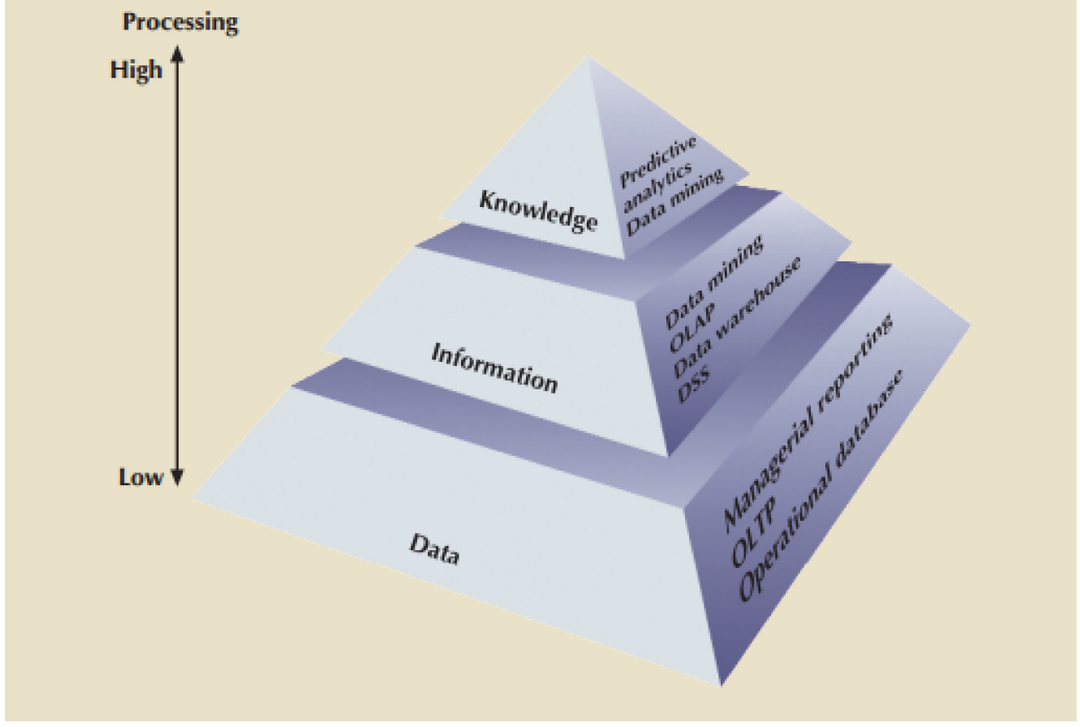
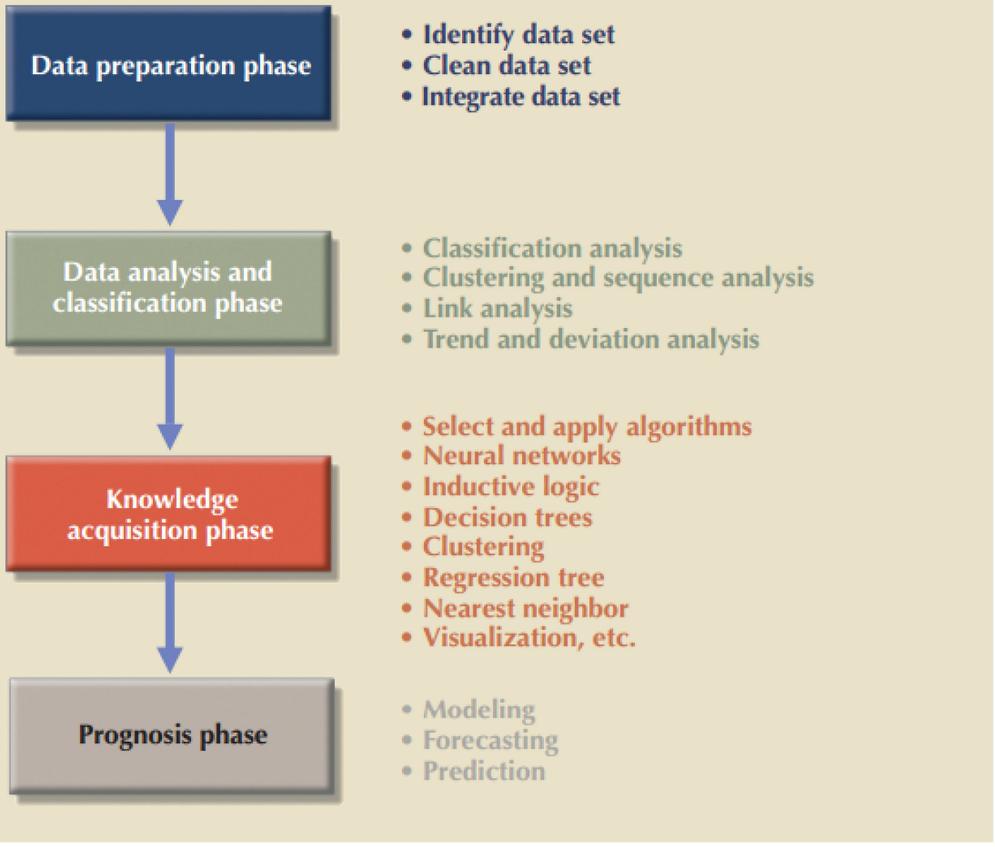


Figure 14.13- Data-Mining Phases

FIGURE 14.13 DATA-MINING PHASES



Data Analytics

- Subset of business intelligence (BI) functionality that encompasses mathematical, statistical, and modeling techniques used to extract knowledge from data
  - Continuous spectrum of knowledge acquisition that goes from discovery to explanation to prediction
- **Explanatory analytics** focuses on discovering and explaining data characteristics based on existing data
- **Predictive analytics** focuses on predicting future data outcomes with a high degree of accuracy

Data Mining

- Focuses on the discovery and explanation stages of knowledge acquisition by:
  - Analyzing massive amounts of data to uncover hidden trends, patterns, and relationships
  - Forming computer models to simulate and explain findings and using them to support decision making
- Can be run in two modes:
  - *Guided* – End-user decides techniques to apply to data
  - *Automated* – End-user sets up the tool to run automatically and the data-mining tool applies multiple techniques to find significant relationships

Predictive Analytics

- Refers to the use of advanced mathematical, statistical, and modeling tools to predict future business outcomes with a high degree of accuracy
  - Focuses on creating actionable models to predict future behaviors and events
  - Most BI vendors are dropping the term *data mining* and replacing it with *predictive analytics*
- Models used in customer service, fraud detection, targeted marketing and optimized pricing
  - Can add value in many different ways but needs to be monitored and evaluated to determine return on investment

```
<?xml version=“1.0”?>
<ancient_wonders>
</ancient_wonders>
```

```
<?xml version=“1.0”?>
<ancient_wonders>
<wonder>
<name>Colosus of Rhodes</name>
</wonder>
</ancient_wonders>
```

```
<?xml version=“1.0”?>
<ancient_wonders>
<wonder>
<name language=“English”>Colosus of Rhodes</name>
<location> Rhodes, Greece</location>
<height units=“feet”> 107</height>
</wonder>
</ancient_wonders>
```

```
<?xml version=“1.0”?>
<ancient_wonders>
<wonder>
<name >Colosus of Rhodes</name>
<location> Rhodes, Greece</location>
<height units=“feet”>107</height>
<main_image file=“colossus.jpg” w=“528” h=“349”/>
</wonder>
</ancient_wonders>
```

```
<?xml version="1.0"?>
<ancient_wonders>
<wonder>
<name language="English">Colosus of Rhodes</name>
<location> Rhodes, Greece</location>
<height units="feet">107</height>
<main_image file="colossus.jpg" w="528" h="349"/>
<source sectionid="101" newspaperid="21"></source>
</wonder>
</ancient_wonders>
```

```
<?xml version=“1.0”?>
<ancient_wonders>
<wonder>
<name language=“English”>Colosus of Rhodes</name>
<location> Rhodes, Greece</location>
<height units=“feet”>107</height>
<main_image file=“colossus.jpg” w=“528” h=“349”/>
<!—this example comes from
Visual Quickstart Guide XML -->
<source sectionid=“101” newspaperid=“21”></source>
</wonder>
</ancient_wonders>
```

```
<?xml version=“1.0”?>
<ancient_wonders>
<wonder>
<name language=“English”>Colosus of Rhodes</name>
<location> Rhodes, Greece</location>
<height units=“feet”>&lt; 107</height>
<main_image file=“colossus.jpg” w=“528” h=“349”/>
<!—this example comes from
Visual Quickstart Guide XML -->
<source sectionid=“101” newspaperid=“21”></source>
</wonder>
</ancient_wonders>
```