WelTec

Te Whare Wānanga o te Awakairangi

西安科技大学高新学院

XI'AN KEDAGAOXIN UNIVERSITY

IT5507 Fundamentals of Data Science

Chapter 14
Big Data and NoSQL

CENGAGE

Learning Objectives

- After completing this chapter, you will be able to:
  - Explain the role of Big Data in modern business
  - Describe the primary characteristics of Big Data and how these go beyond the traditional “3 Vs”
  - Explain how the core components of the Hadoop framework operate
  - Identify the major components of the Hadoop ecosystem
  - Summarize the four major approaches of the NoSQL data model and how they differ from the relational model
  - Describe the characteristics of NewSQL databases

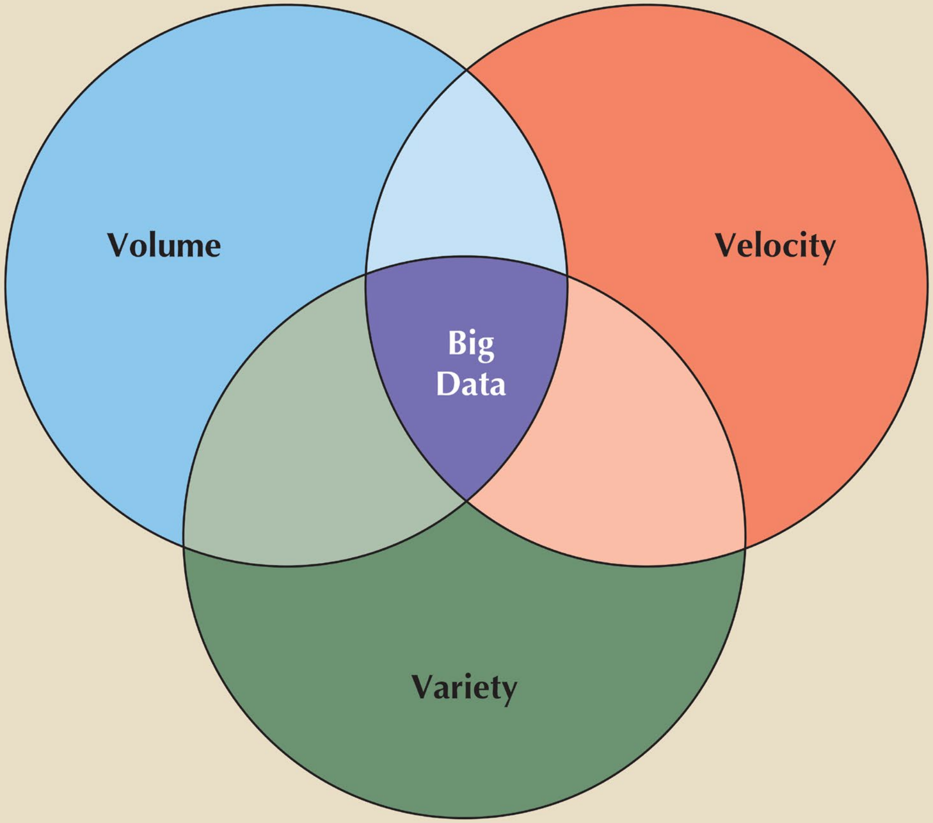
Big Data (1 of 4)

Big data refers to large volumes of structured, semi-structured, and unstructured data that inundates businesses on a daily basis. It encompasses data characterized by its volume, velocity, variety, and complexity, challenging traditional data processing methods but offering valuable insights when effectively analyzed.

- **Volume:** quantity of data to be stored
  - Scaling up: keeping the same number of systems but migrating each one to a larger system
  - Scaling out: 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: analysis of data to produce actionable results
- **Variety:** variations in the structure of data to be stored
  - Structured data: fits into a predefined data model
  - Unstructured data: does not fit into a predefined model

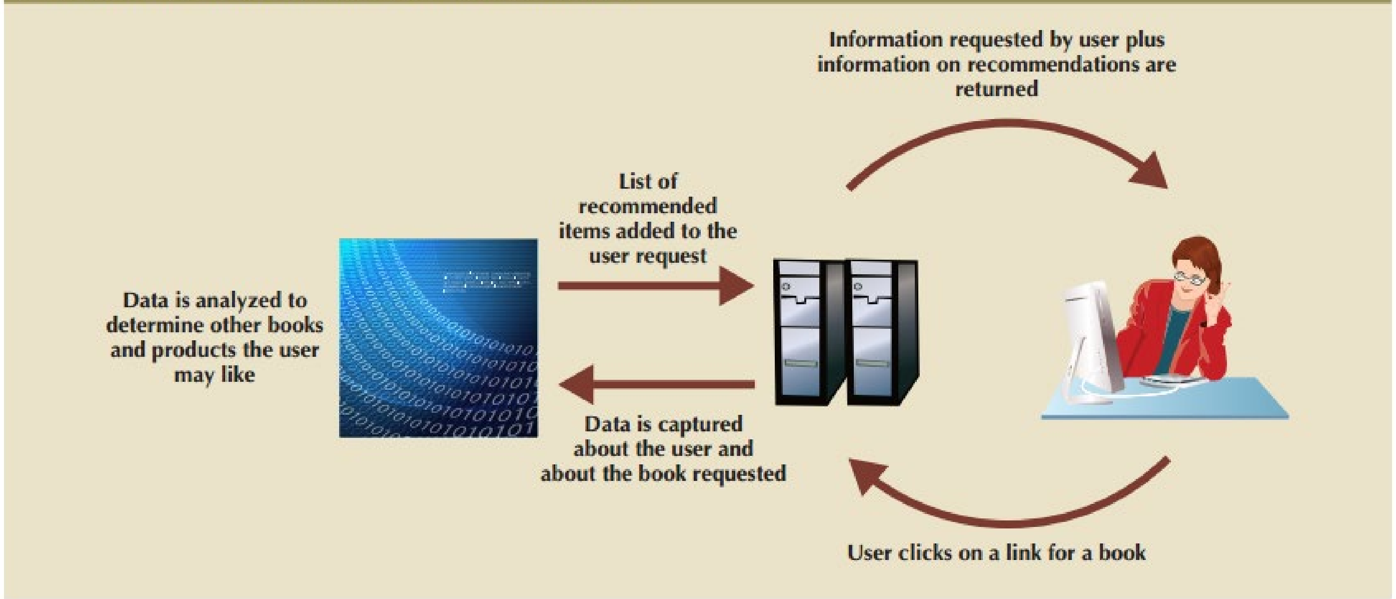
Big Data (2 of 4)

FIGURE 14.1 ORIGINAL VIEW OF BIG DATA



Big Data (3 of 4)

FIGURE 14.3 FEEDBACK LOOP PROCESSING


Information requested by user plus information on recommendations are returned
List of recommended items added to the user request
Data is analyzed to determine other books and products the user may like
Data is captured about the user and about the book requested
User clicks on a link for a book

Feedback loop processing in big data refers to the iterative process of collecting, analyzing, and acting upon insights gained from the data. This involves continuously refining data analysis techniques, adjusting models or algorithms, and implementing changes based on the feedback received. The goal is to improve decision-making, optimize processes, and enhance outcomes over time by leveraging the insights generated from analyzing large volumes of data.

Big Data (4 of 4)

- Other characteristics
  - Variability: changes in the meaning of data based on context
  - Sentimental analysis: attempts to determine if a statement conveys a positive, negative, or neutral attitude about a topic
  - Veracity: trustworthiness of data
  - Value: degree data can be analyzed for meaningful insight
  - Visualization: ability to graphically resent data to make it understandable
- Relational databases are not necessarily the 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

  - Polyglot persistence simply means using multiple types of data storage and management technologies in an organization's systems. Instead of relying on just one type of database, companies might use different ones depending on the specific needs of their data. This approach allows them to choose the best tool for each job, whether it's traditional relational databases, NoSQL databases, or other specialized systems

Hadoop (1 of 8)

- 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
- 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 (2 of 8)

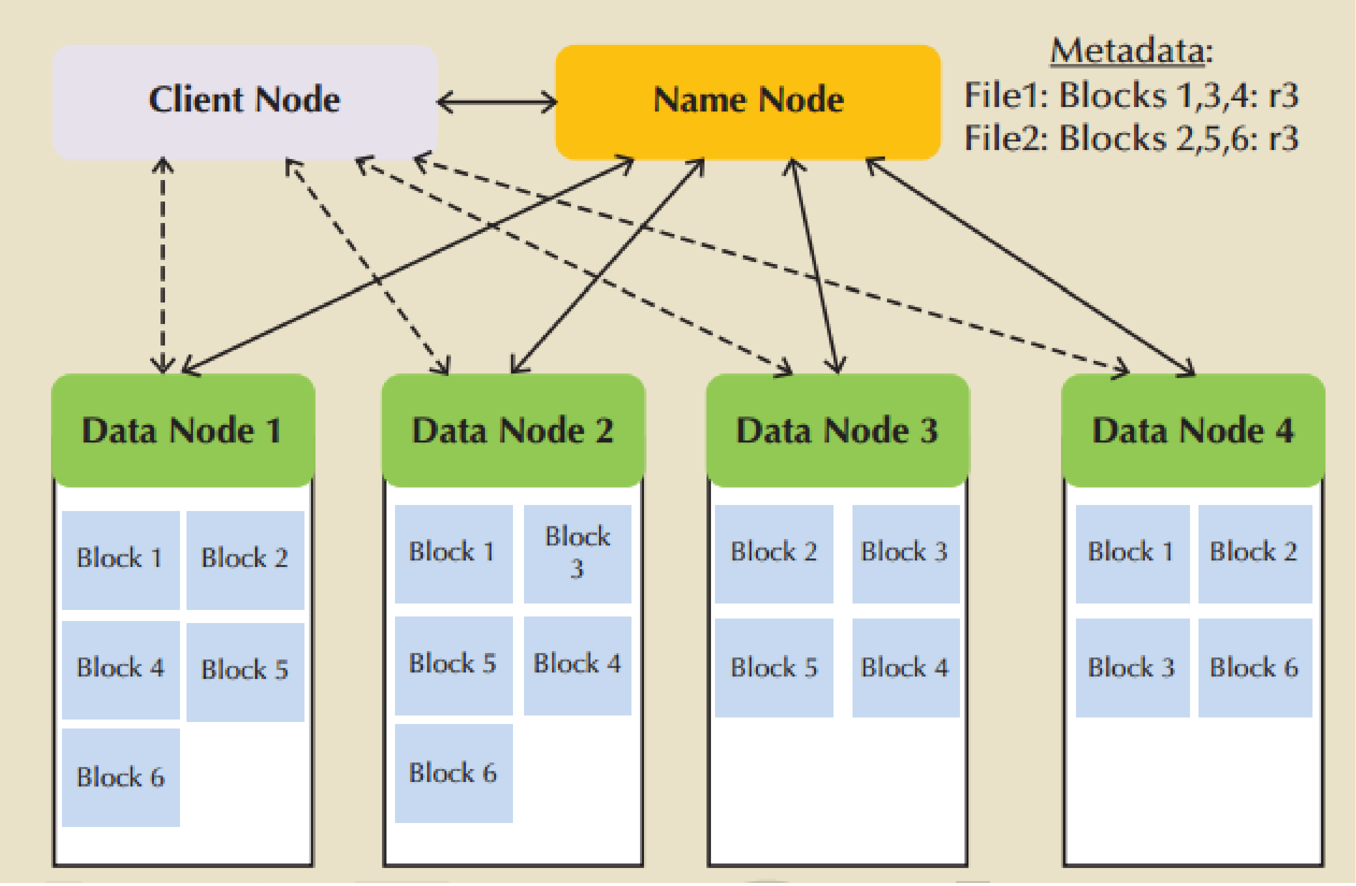
- Hadoop Distributed File System (HDFS)
  - Based on several key assumptions
    - High volume: default block sizes is 64 MB and can be configured to even larger values
    - Write-once, read-many: model simplifies concurrency issues and improves data throughput
    - Streaming access: optimized for batch processing of entire files as a continuous stream of data
    - Fault tolerance: designed to replicate data across many different devices so that when one fails, data is still available from another device

Hadoop (3 of 8)

- Hadoop uses several types of nodes; computers that perform one or more types of tasks within the system
  - 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 and send back block reports and heartbeats

Hadoop (4 of 8)

FIGURE 14.4 HADOOP DISTRIBUTED FILE SYSTEM (HDFS)



Hadoop (5 of 8)

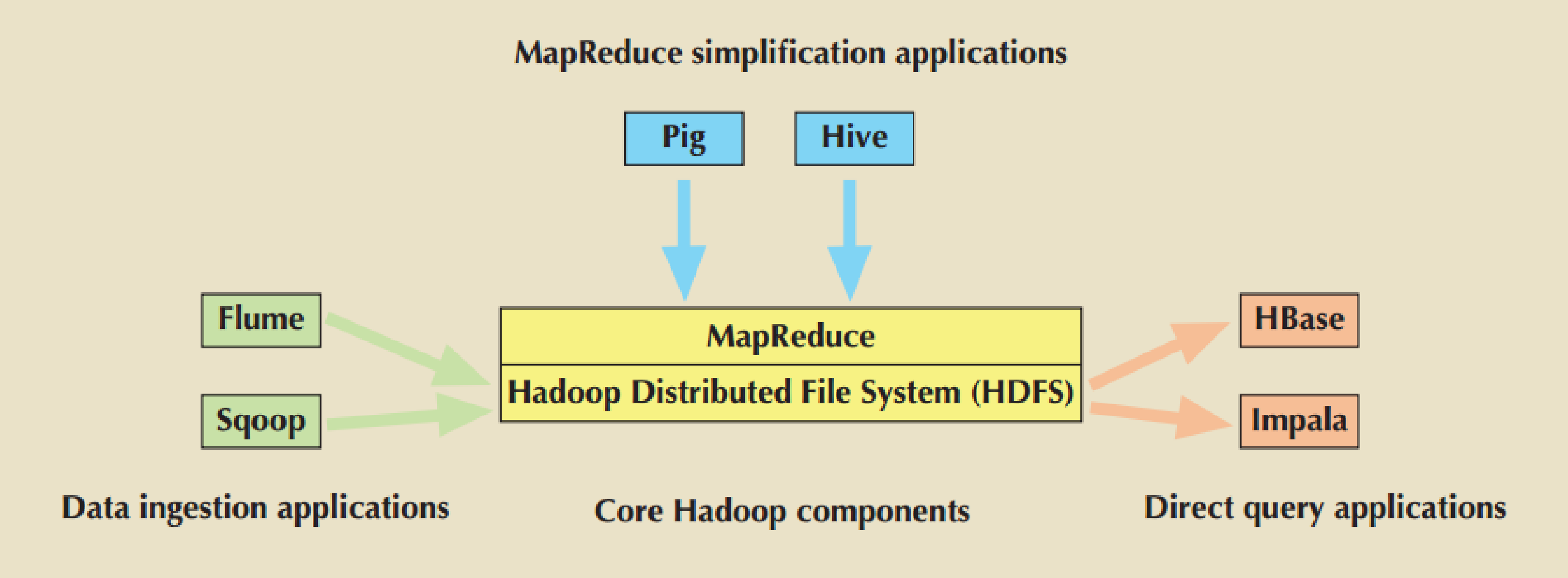
- MapReduce
  - Framework used to process large data sets across clusters
    - Breaks down complex tasks into smaller subtasks, performing the subtasks, and producing a final result
    - Map function takes a collection of data and sorts and filters it into a set of key-value pairs
      - Mapper program performs the map function
    - Reduce summaries results of map function produce a single result
      - Reducer program performs the reduce function
  - Implementation complements HDFS structure
    - Job tracker: central control program
    - Task tracker: reduces tasks on a node
    - Batch processing: runs tasks from beginning to end with no user interaction

Hadoop (6 of 8)

- Hadoop ecosystem
  - Most organizations that use Hadoop also use a set of other related products that interact and complement each other to produce an entire ecosystem of applications and tools
  - Like any ecosystem, the interconnected pieces are constantly evolving and their relationships are changing, so it is a rather fluid situation

Hadoop (7 of 8)

FIGURE 14.6 A SAMPLE OF THE HADOOP ECOSYSTEM


MapReduce simplification applications
Pig
Hive
Flume
Sqoop
MapReduce
Hadoop Distributed File System (HDFS)
HBase
Impala
Data ingestion applications
Core Hadoop components
Direct query applications

Hadoop (8 of 8)

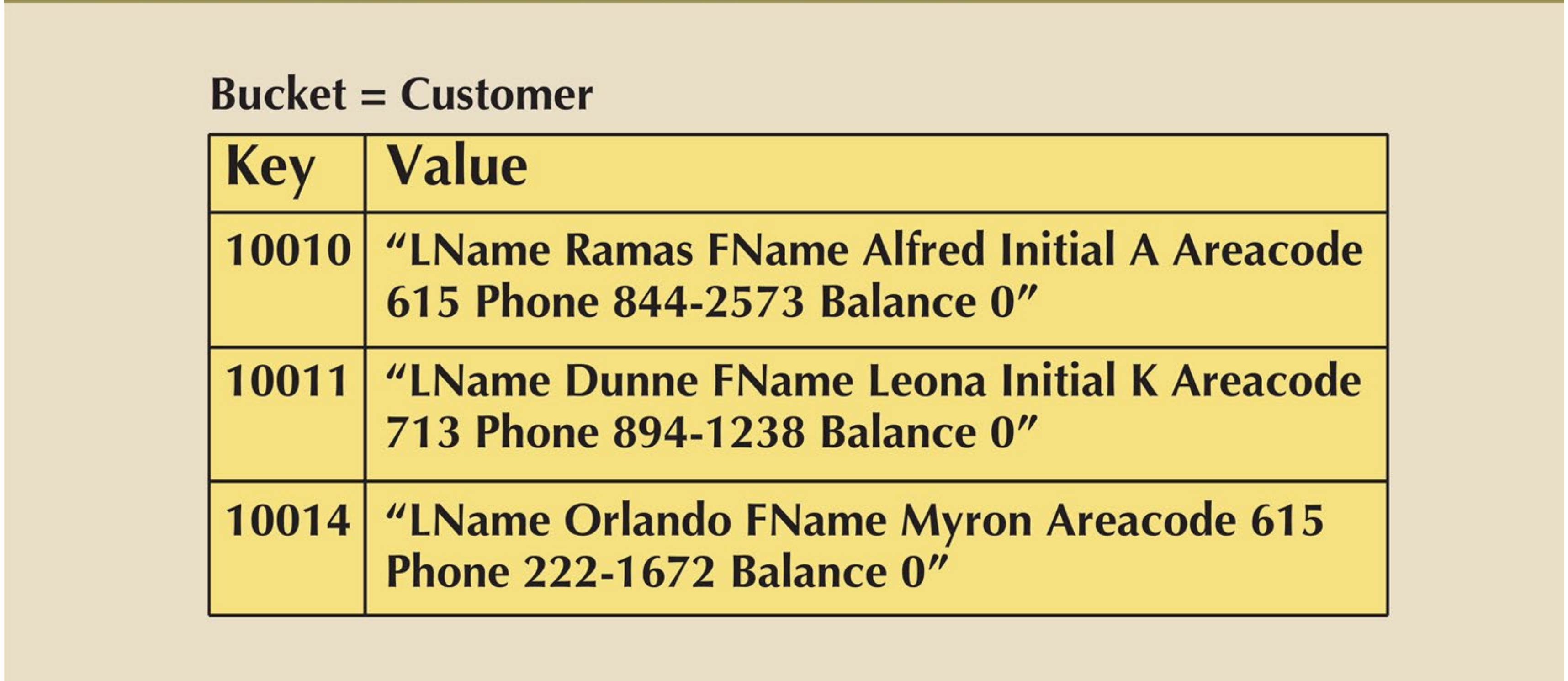
- Map reduce simplification applications
  - Hive: data warehousing system that sites on top of HDFS and supports its own SQL-like language
  - Pig: tool that compiles a high-level scripting language, named Pig Latin, into MapReduce jobs for executing in Hadoop
- Data ingestion applications
  - Flume: component for ingesting data in Hadoop
  - Sqoop: tool for converting data back and forth between a relational database and the HDFS
- Direct query applications
  - Hbase: column-oriented NoSQL database designed to sit on top of the HDFS that quickly processes sparse datasets
  - Impala: the first SQL on Hadoop application

NoSQL (1 of 7)

- Nosql: non-relational database technologies developed to address Big Data challenges
  - Name does not describe what the NoSQL technologies are, but rather what they are not (poor job of that as well)
- Key-value (KV) databases: conceptually the simplest of the NoSQL data models
  - Store data as a collection of key-value pairs organized as buckets which are the equivalent of tables
- Document databases: similar to key-value databases and can almost be considered a subtype of KV databases
  - Store data in key-value pairs in which the value components are encoded documents grouped into large groups called collections

NoSQL (2 of 7)

FIGURE 14.7 KEY-VALUE DATABASE STORAGE

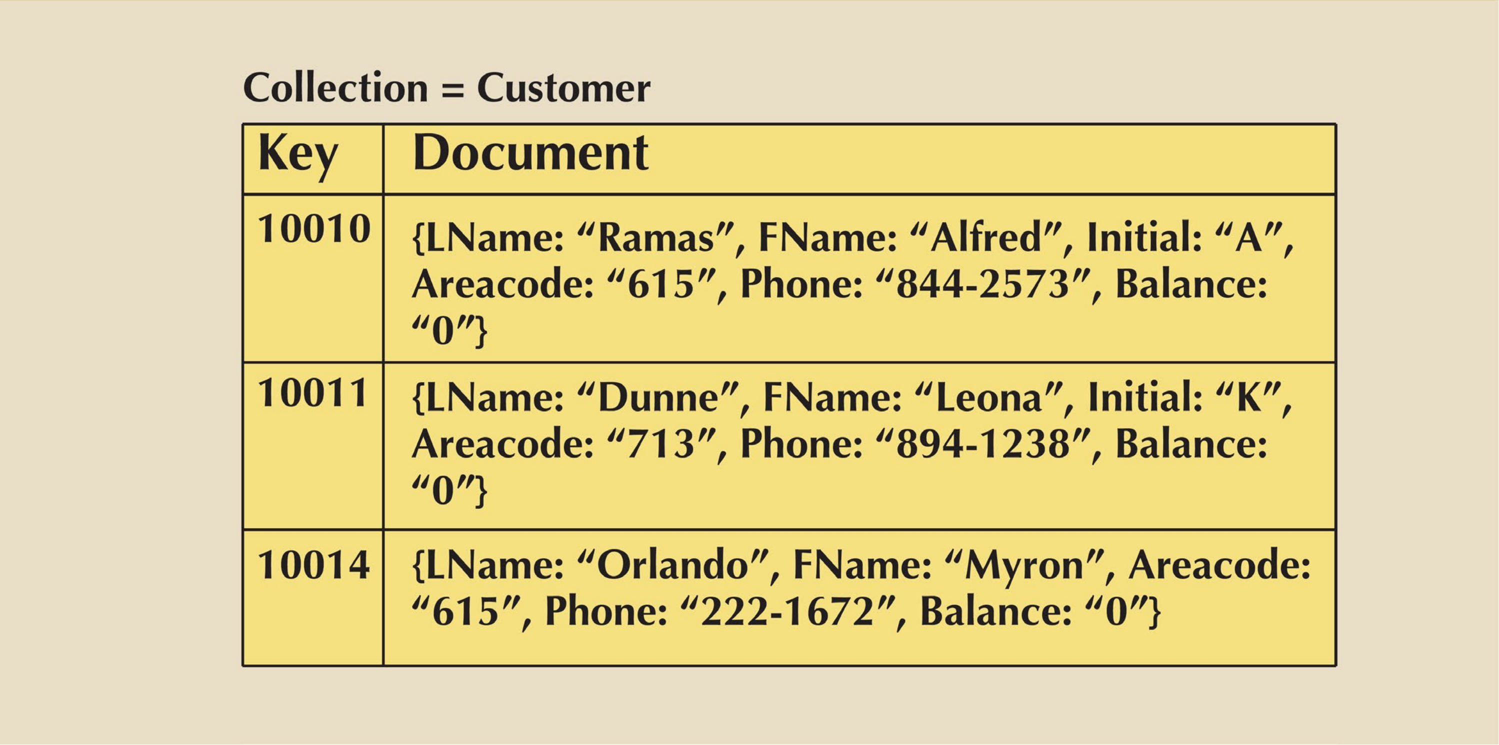


Bucket = Customer

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"

NoSQL (3 of 7)

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”}

NoSQL (4 of 7)

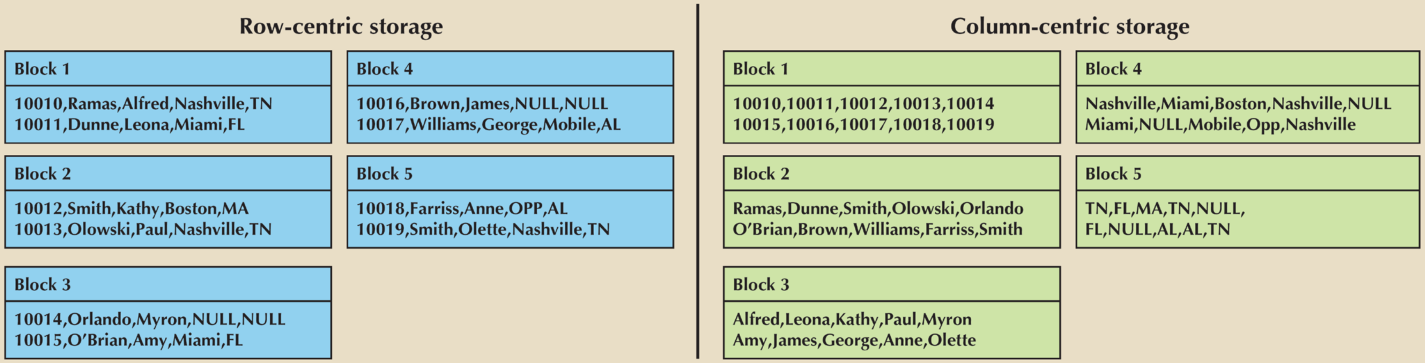
- Column-oriented databases refers to two technologies
  - Column-centric storage: data stored in blocks which hold data from a single column across many rows
  - Row-centric storage: data stored in block which hold data from all columns of a given set of rows
- Graph databases store data on relationship-rich data as a collection of nodes and edges
  - Properties: like attributes; they are the data that we need to store about the node
  - Traversal: query in a graph database

NoSQL (5 of 7)

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



Row-centric storage

Block 1
10010,Ramas,Alfred,Nashville,TN
10011,Dunne,Leona,Miami,FL

Block 4
10016,Brown,James,NULL,NULL
10017,Williams,George,Mobile,AL

Block 2
10012,Smith,Kathy,Boston,MA
10013,Olowski,Paul,Nashville,TN

Block 5
10018,Farriss,Anne,OPP,AL
10019,Smith,Olette,Nashville,TN

Block 3
10014,Orlando,Myron,NULL,NULL
10015,O'Brian,Amy,Miami,FL

Column-centric storage

Block 1
10010,10011,10012,10013,10014
10015,10016,10017,10018,10019

Block 4
Nashville,Miami,Boston,Nashville,NULL
Miami,NULL,Mobile,Opp,Nashville

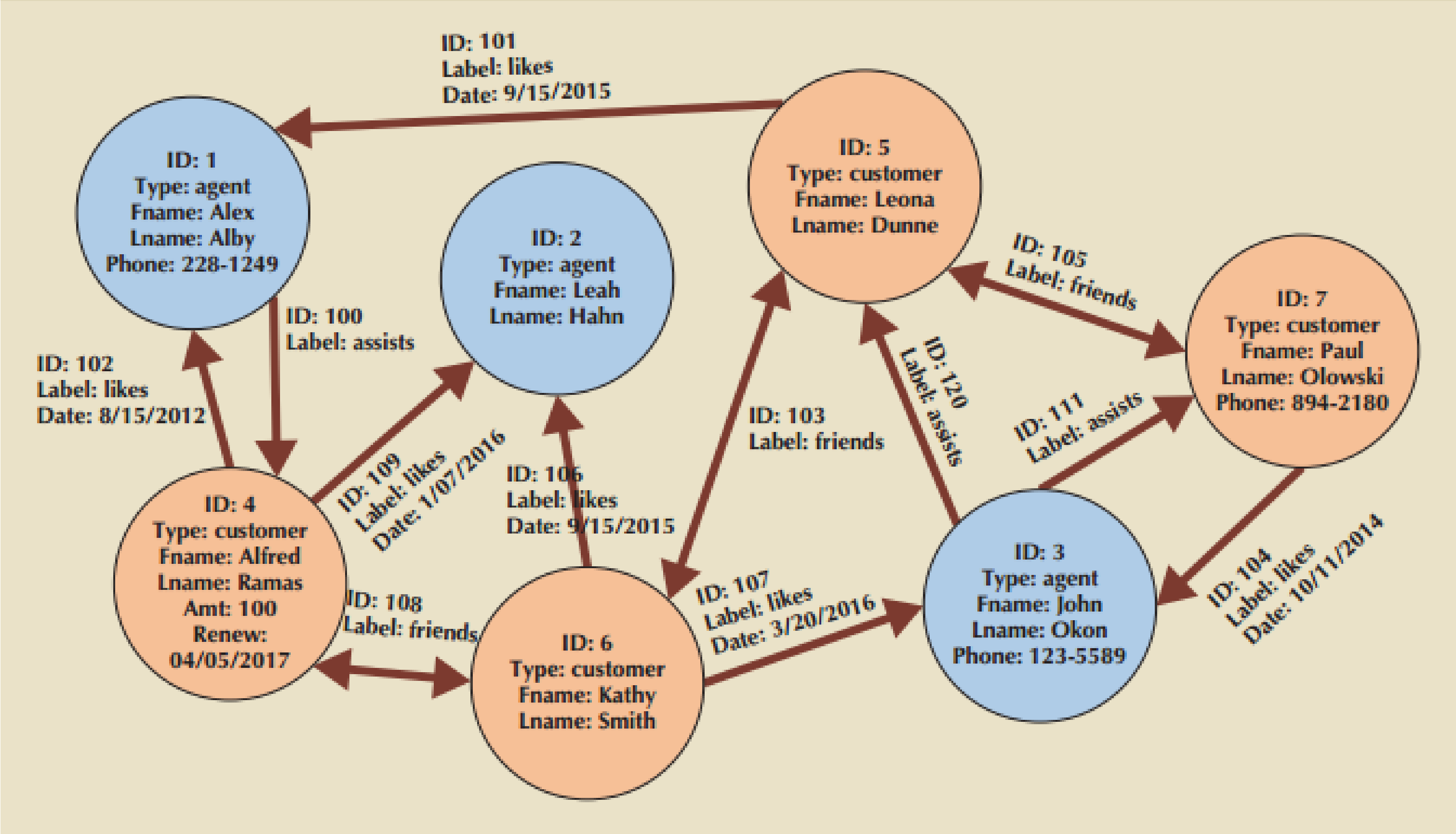
Block 2
Ramas,Dunne,Smith,Olowski,Orlando
O'Brian,Brown,Williams,Farriss,Smith

Block 5
TN,FL,MA,TN,NULL,
FL,NULL,AL,AL,TN

Block 3
Alfred,Leona,Kathy,Paul,Myron
Amy,James,George,Anne,Olette

NoSQL (6 of 7)

FIGURE 14.11 GRAPH DATABASE REPRESENTATION



CENGAGE

NoSQL (7 of 7)

- Aggregate awareness: data is collected or aggregated around a central topic or entity
  - Aggregate aware database models achieve clustering efficiency by making each piece of data relatively independent
- Graph databases, like relational databases, are aggregate ignorant
  - Do not organize the data into collections based on a central entity

NewSQL Databases (1 of 2)

- 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

NewSQL databases are a class of relational database management systems (RDBMS) that aim to combine the benefits of traditional SQL databases with the scalability and performance of NoSQL databases. They offer features such as horizontal scalability, distributed processing, and high availability while maintaining ACID compliance and support for SQL queries and transactions.

NewSQL Databases (2 of 2)

- NewSQL databases support:
  - SQL as the primary interface
  - ACID-compliant transactions (Atomicity, Consistency, Isolation, and Durability)

ACID compliance refers to a set of properties that ensure reliability and consistency in database transactions:

- Atomicity: Ensures that all operations within a transaction are completed successfully or none at all.
- Consistency: Guarantees that the database remains in a valid state before and after a transaction.
- Isolation: Prevents interference between concurrent transactions, ensuring that they operate independently.
- Durability: Ensures that committed transactions persist even in the event of system failures.

- Similar to NoSQL, NewSQL databases also support:
  - Highly distributed clusters
  - Key-value or column-oriented data stores

Working with Document Databases Using MongoDB (1 of 2)

- Popular document database
  - Among the NoSQL databases currently available, MongoDB has been one of the most successful in penetrating the database market
- MongoDB, comes from the word humongous as its developers intended their new product to support extremely large data sets
  - High availability
  - High scalability
  - High performance

Working with Document Databases Using MongoDB (2 of 2)

- Importing Documents in MongoDB
  - Refer to the text for an importation example and considerations
- Example of a MongoDB Query Using find()
  - Methods are programed functions to manipulate objects
    - Find() method retrieves objects from a collection that match the restrictions provided
    - Pretty() method is used to improve readability of the documents by placing key:value pairs on separate lines
  - Refer to the text for a query example

Working with Graph Databases Using Neo4j (1 of 3)

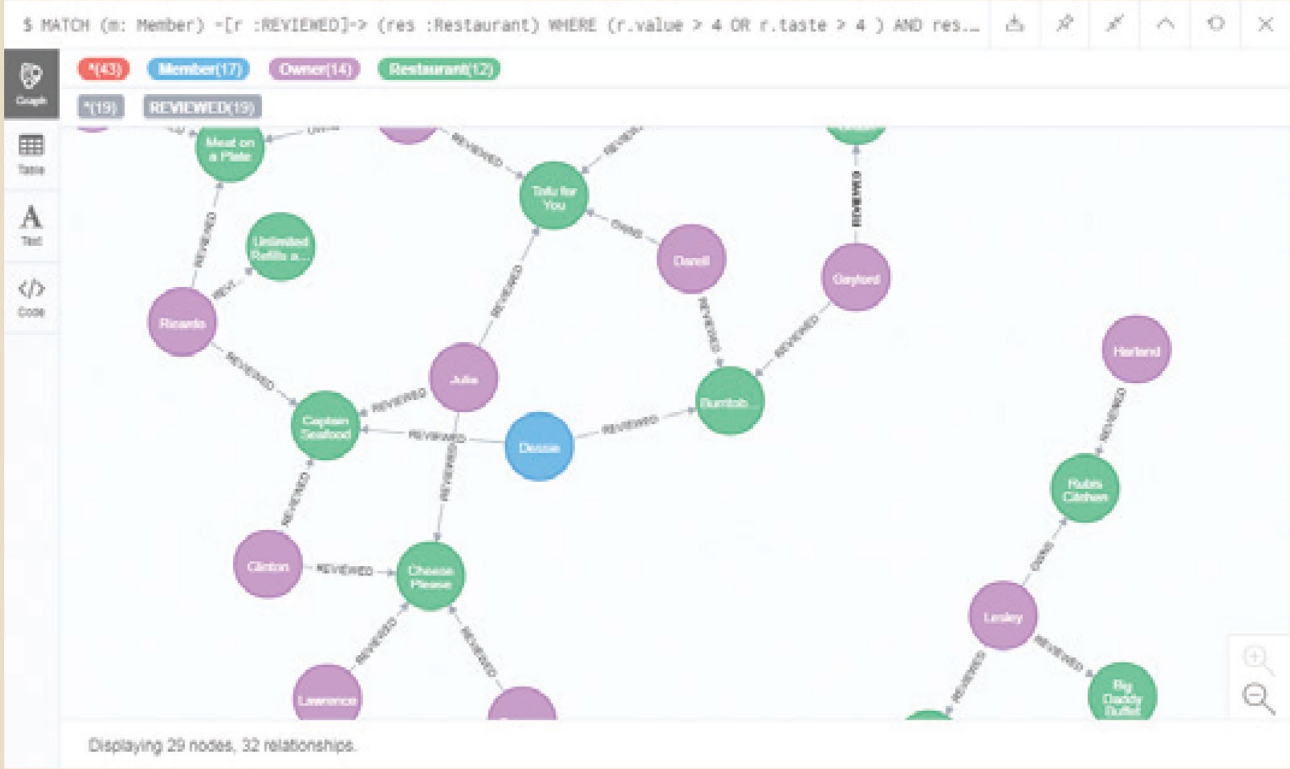
- Even though Neo4j is not yet as widely adopted as MongoDB, it has been one of the fastest growing NoSQL databases
  - Graph databases still work with concepts similar to entities and relationships
    - Focus is on the relationships
  - Graph databases are used in environments with complex relationships among entities
    - Heavily reliant on interdependence among their data
  - Neo4j provides several interface options
    - Designed with Java programming in mind
- Creating nodes in Neo4j
  - Nodes in a graph database correspond to entity instances in a relational database
  - Cypher is the interactive, declarative query language in Neo4j
  - Nodes and relationships are created using a CREATE command

Working with Graph Databases Using Neo4j (2 of 3)

- Refer to the text for examples
  - Using the CREATE command to create a member node
  - Retrieving node data with MATCH and WHERE
  - Retrieving relationship data with MATCH and WHERE

Working with Graph Databases Using Neo4j (3 of 3)

FIGURE 14.13 NEO4J QUERY USING MATCH/WHERE/RETURN



```
$ MATCH (m: Member) -[r :REVIEWED]-> (res :Restaurant) WHERE (r.value > 4 OR r.taste > 4 ) AND res...
```

Displaying 29 nodes, 32 relationships.

- MongoDB, NoSQL, NewSQL, and Neo4j are all different types of databases designed to address specific needs and use cases:
- MongoDB: A NoSQL database that uses a flexible document-based model to store data. It's known for its scalability, high performance, and ease of development, making it suitable for applications with rapidly evolving schemas and large volumes of data.
- NoSQL: A category of databases that diverge from traditional relational databases, offering more flexibility and scalability for handling unstructured or semi-structured data. NoSQL databases come in various types, including document-based (like MongoDB), key-value stores, column-oriented, and graph databases.
- NewSQL: A class of relational databases that aim to combine the benefits of traditional SQL databases (like ACID compliance and strong consistency) with the scalability and performance of NoSQL databases. NewSQL databases are designed to handle large-scale transactional workloads while maintaining the relational model.
- Neo4j: A graph database that uses graph structures with nodes, edges, and properties to represent and store data. It's optimized for handling complex relationships and querying graph data, making it ideal for applications like social networks, recommendation engines, and network analysis.

Summary (1 of 2)

- Big Data is characterized by data of such volume, velocity, and/or variety that the relational model struggles to adapt to it
- Volume, velocity, and variety are collectively referred to as the 3 Vs of Big Data
- The Hadoop framework has quickly emerged as a standard for the physical storage of Big Data
- NoSQL is a broad term to refer to any of several nonrelational database approaches to data management
- Key-value databases store data in key-value pairs
- Document databases also store data in key-value pairs, but the data in the value component is an encoded document

Summary (2 of 2)

- Column-oriented databases, also called column family databases, organize data into key-value pairs in which the value component is composed of a series of columns, which are themselves key-value pairs
- Graph databases are based on graph theory and represent data through nodes, edges, and properties
- NewSQL databases attempt to integrate features of both RDBMS (providing ACID-compliant transactions) and NoSQL databases (using a highly distributed infrastructure)
- MongoDB is a document database that stores documents in JSON format
- Neo4j is a graph database that stores data as nodes and relationships, both of which can contain properties to describe them