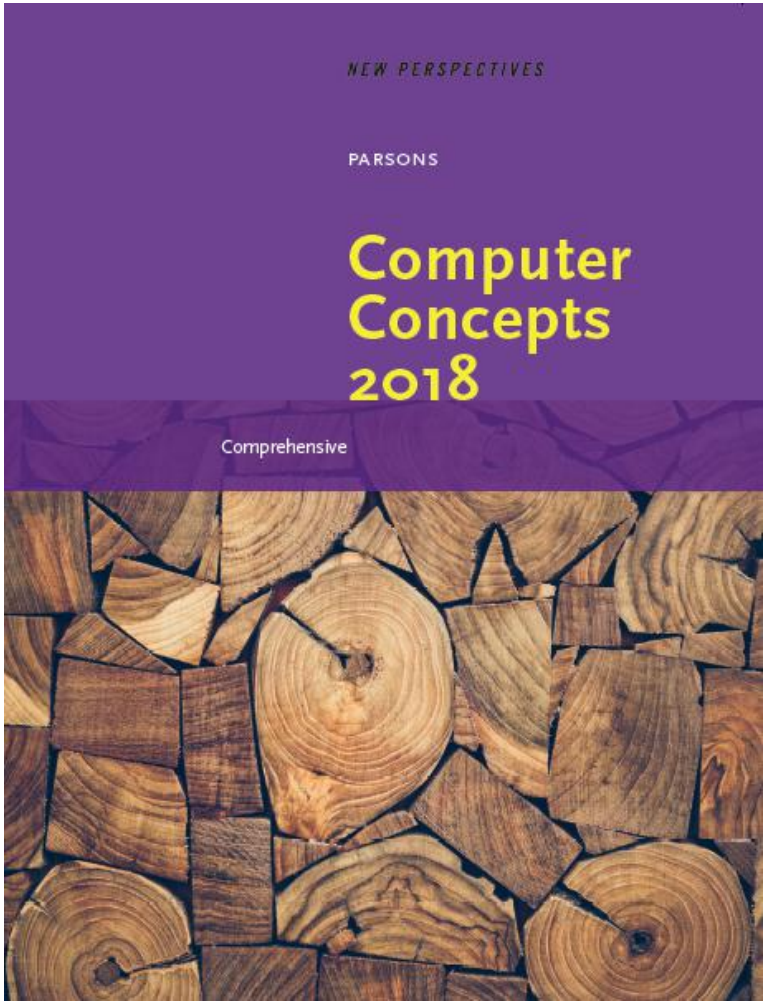
Computer Concepts 2018



Module 10
Databases

Module Contents

- Section A: Database Basics
- Section B: Database Tools
- Section C: Database Design
- Section D: SQL
- Section E: Big Data

Section A: Database Basics

- Operational and Analytical Databases
- Database Models

Section A: Objectives

- Describe the difference between an operational database and an analytical database
- List seven activities associated with operational databases
- Provide at least three examples of analytics applied to databases
- Draw the data structure for a flat file and label each component
- Draw ERDs and provide real-life examples of one-to-one, one-to-many, and many-to-many relationships
- Diagram the data structure for hierarchical, graph, relational, multidimensional, and object database models

Operational and Analytical Databases (1 of 8)

- An **operational database** is used to collect, modify, and maintain data on a daily basis
- An **analytical database** is used to collect data that will be used for spotting trends that offer insights for tactical and strategic business decisions

Operational and Analytical Databases (2 of 8)

- Operational Databases
  - Operational databases are commonly part of an enterprise's TPS, OLTP, CRM, SCM, or ERP information systems.
  - They store data as it is collected from point-of-sale systems, customer loyalty programs, social media signups, and other transactions.
  - The data is typically dynamic. It changes constantly and reflects up-to-the-minute information.

Operational and Analytical Databases (3 of 8)

- Analytical Databases
  - Analytical databases commonly hold historical data copied from one or more transaction processing systems.
  - Unlike an operational database, the data in an analytical database is not being constantly updated. Therefore, it remains relatively static.
  - Because the data is historical, the information that can be inferred is like a snapshot of a point in time.

Operational and Analytical Databases (4 of 8)

- Operational databases perform the following:
  - Collect and store data
  - View data
  - Find data
  - Update data
  - Organize data
  - Distribute data
  - Move or remove data

Operational and Analytical Databases (5 of 8)

- Analytical databases store data that is used by corporate executives, strategic planners, and other workers to examine business metrics
- Decision makers can access analytical databases using an **executive dashboard**, provided by software such as iDashboards, which uses tools for visually displaying query results

Operational and Analytical Databases (6 of 8)

Operational and Analytical Databases (7 of 8)

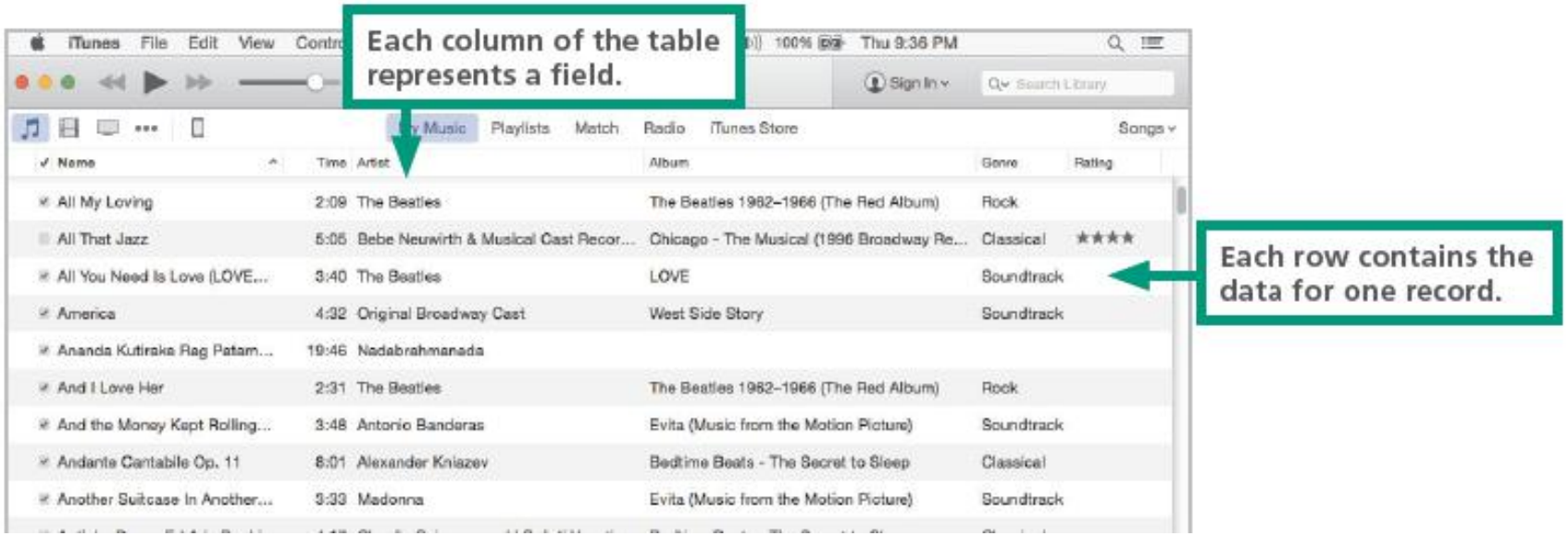
- Analytical databases perform the following:
  - Find relationships and patterns using data mining
  - Make predictions using predictive analytics
  - Examine multiple factors using OLAP (online analytical processing)

Operational and Analytical Databases (8 of 8)

Database Models (1 of 13)

- The underlying structure of a database is referred to as a **database model**
- One of the simplest models for storing data is a **flat file** that consists of a single, two-dimensional table of data elements
- Spreadsheets are stored as flat files displayed as rows and columns

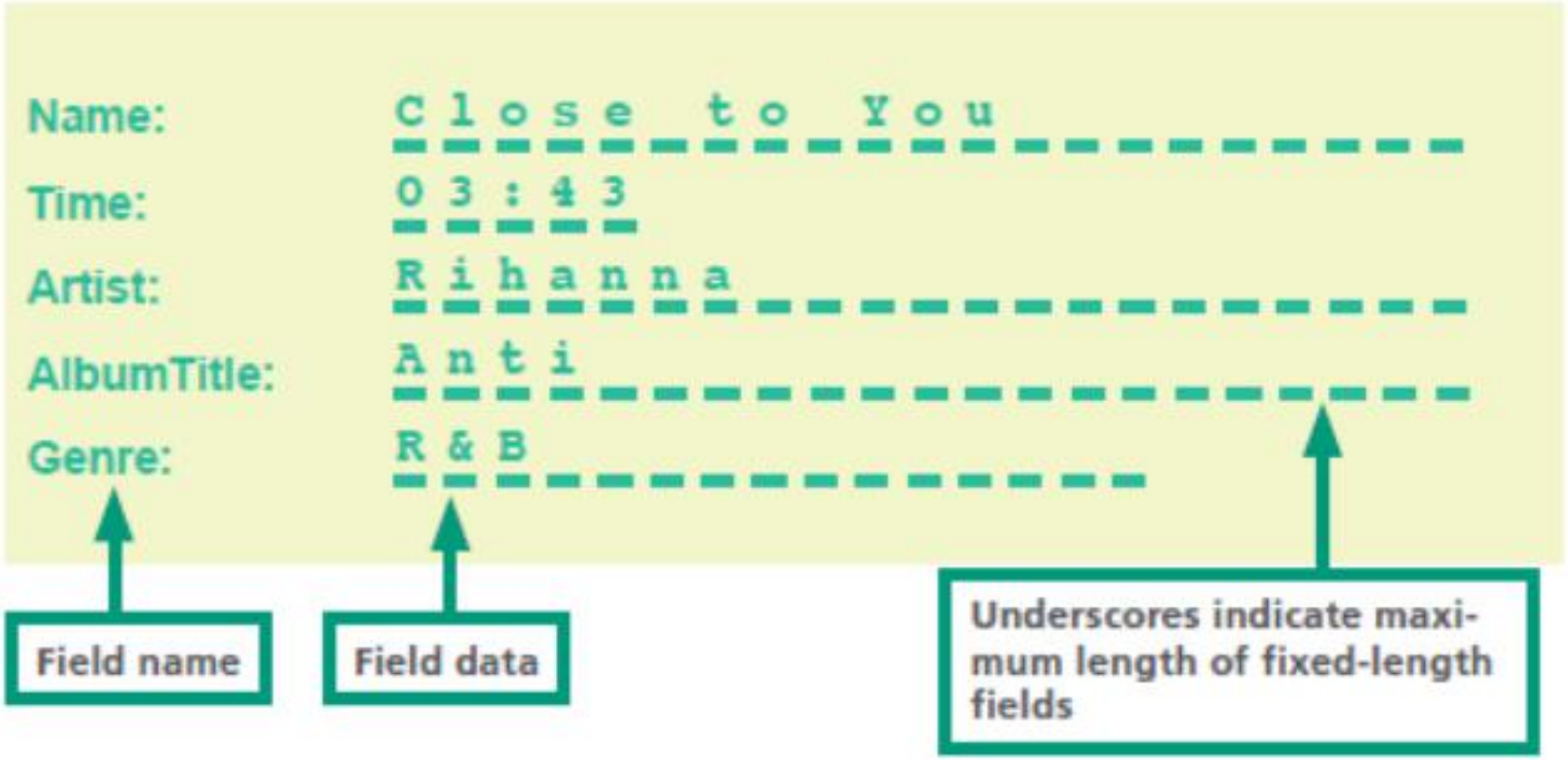
Database Models (2 of 13)



Database Models (3 of 13)

- A **field** contains the smallest unit of meaningful information; it is the basic building block for a structured file or database
- A **variable-length field** is like an accordion—it expands to fit the data you enter
- A **fixed-length field** contains a predetermined number of characters (bytes)
- In the world of databases, a **record** refers to a collection of data fields; the template for a record is a **record type**

Database Models (4 of 13)



Database Models (5 of 13)

- In database jargon, a **relationship** is an association between data that's sorted in different record types
- An important aspect of the relationship between record types is **cardinality**, which refers to the number of associations that can exist between two record types
- The relationship between record types can be depicted graphically with an **entity-relationship diagram** (sometimes called an ER diagram or ERD)

Database Models (6 of 13)

One-to-one relationship

An album has only one description in iTunes.

One-to-many relationship

One genre can be assigned to many albums.

Many-to-many relationship

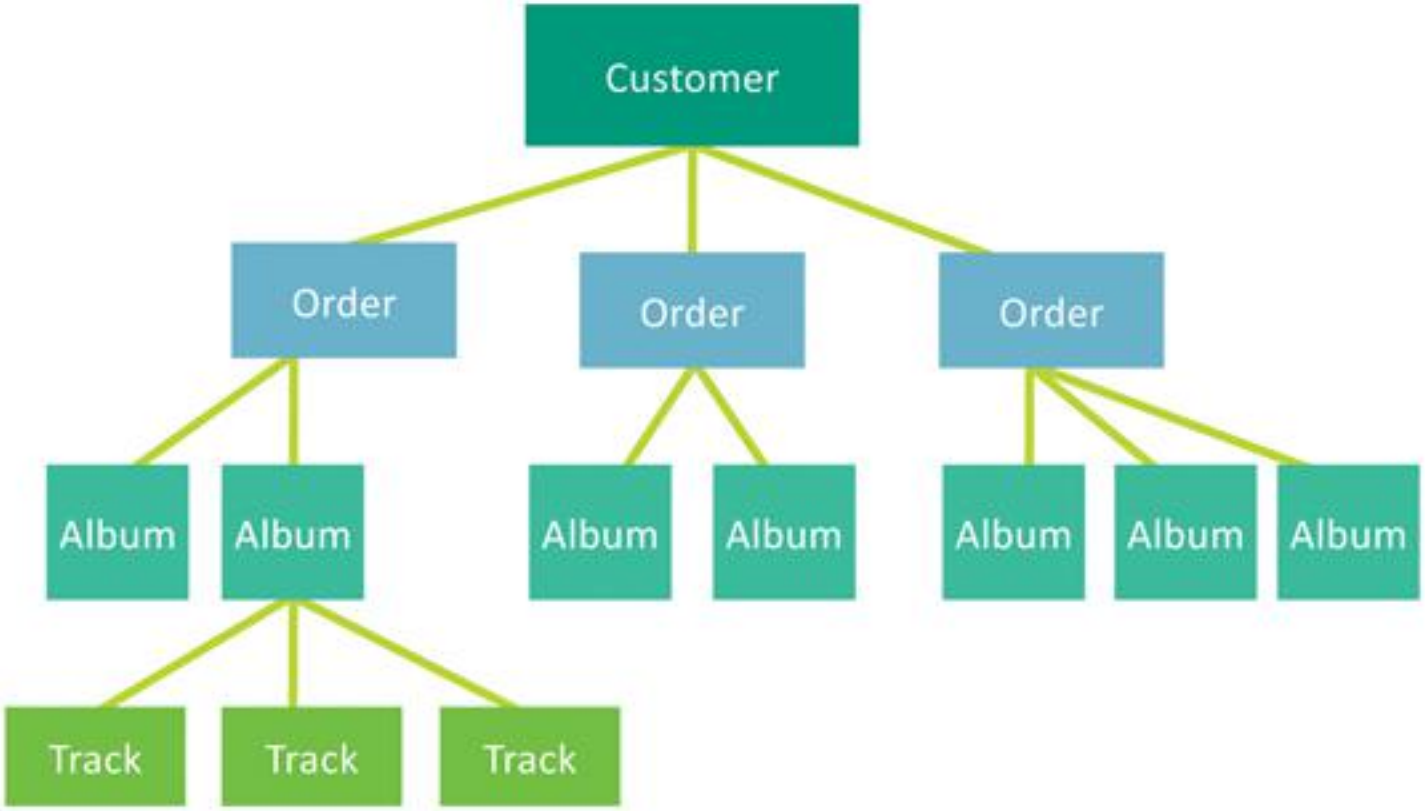
Albums contain many tracks, and tracks can be included on several different albums.

Key to ERD Symbols

Database Models (7 of 13)

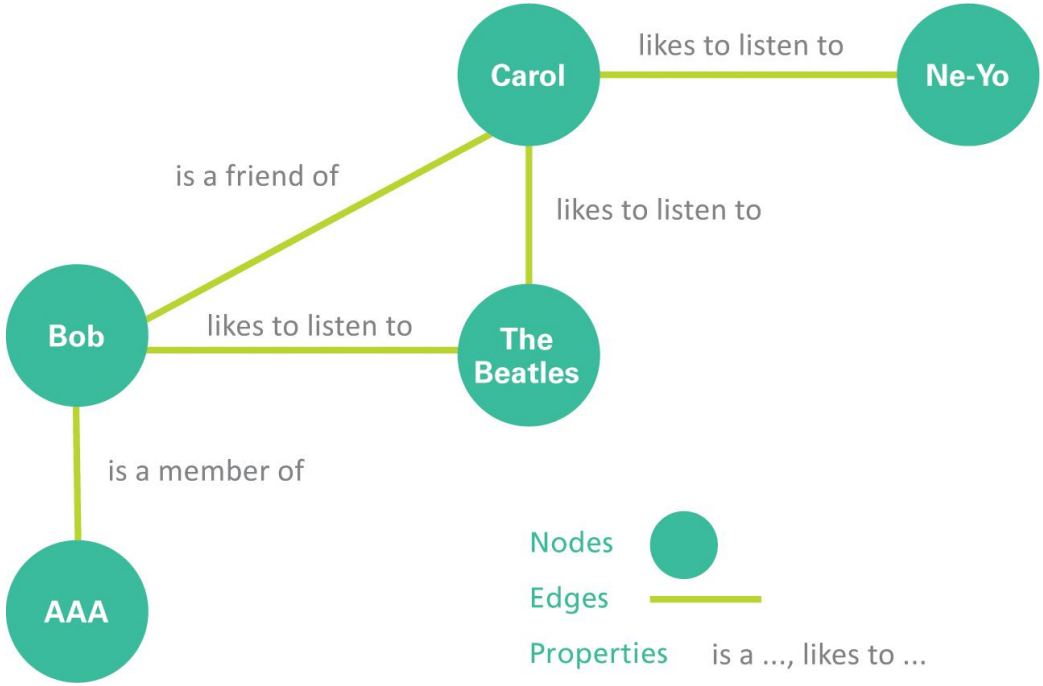
- Many database models keep track of relationships among data, but there are different techniques for doing so
- A hierarchical database allows one-to-one and one-to-many relationships that are linked in a hierarchical structure

Database Models (8 of 13)



Database Models (9 of 13)

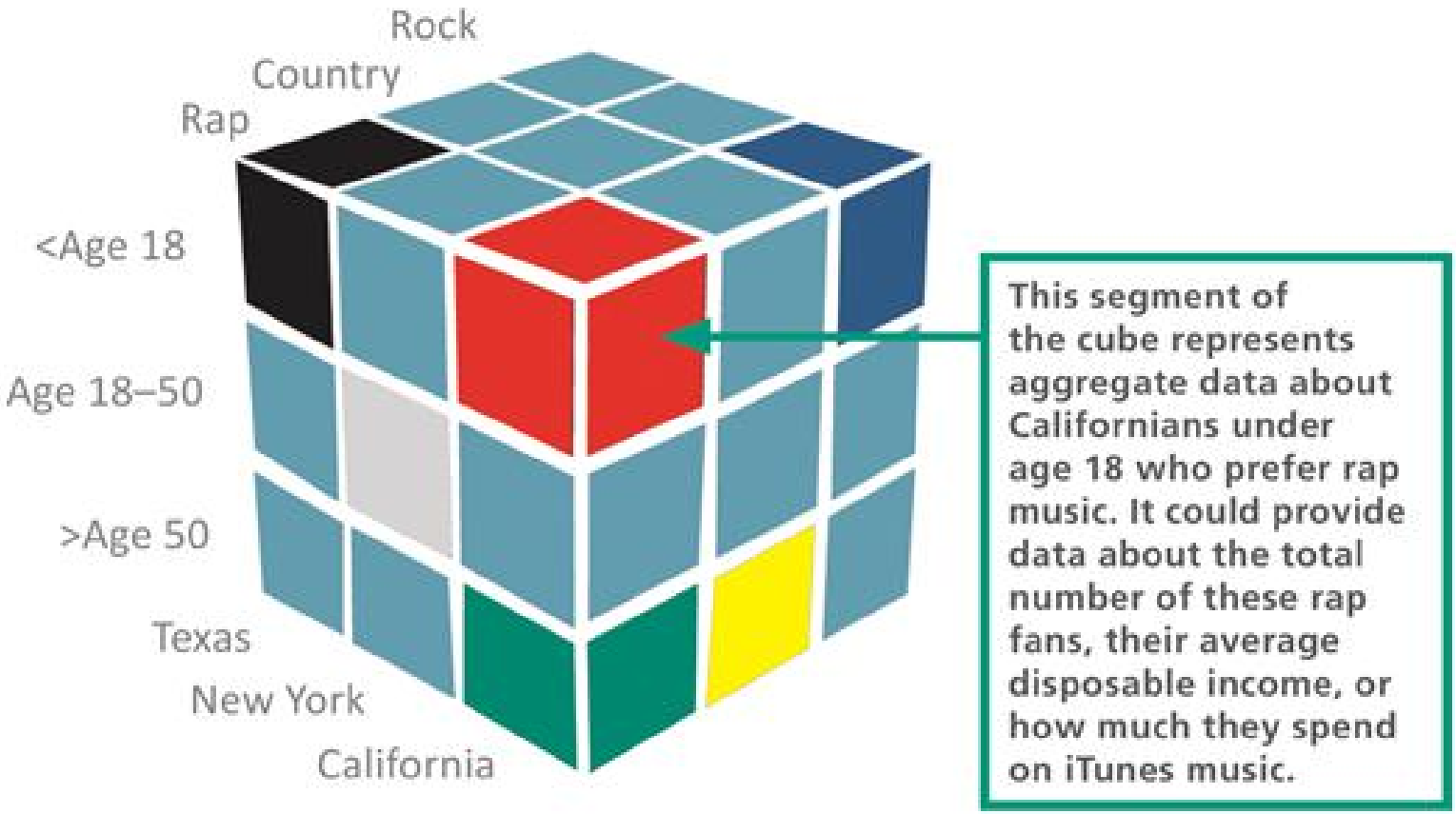
- A **graph database** offers an alternative way to track relationships; its structure resembles sociograms with their interlinked nodes



Database Models (10 of 13)

- A **relational database** stores data in a collection of related tables
- Each **table** is a sequence of records, similar to a flat file
- A **multidimensional database** organizes relationships over three or more dimensions; in the context of databases, a dimension is a layer based on a data element, such as a product, place, or customer, that can be used to categorize data

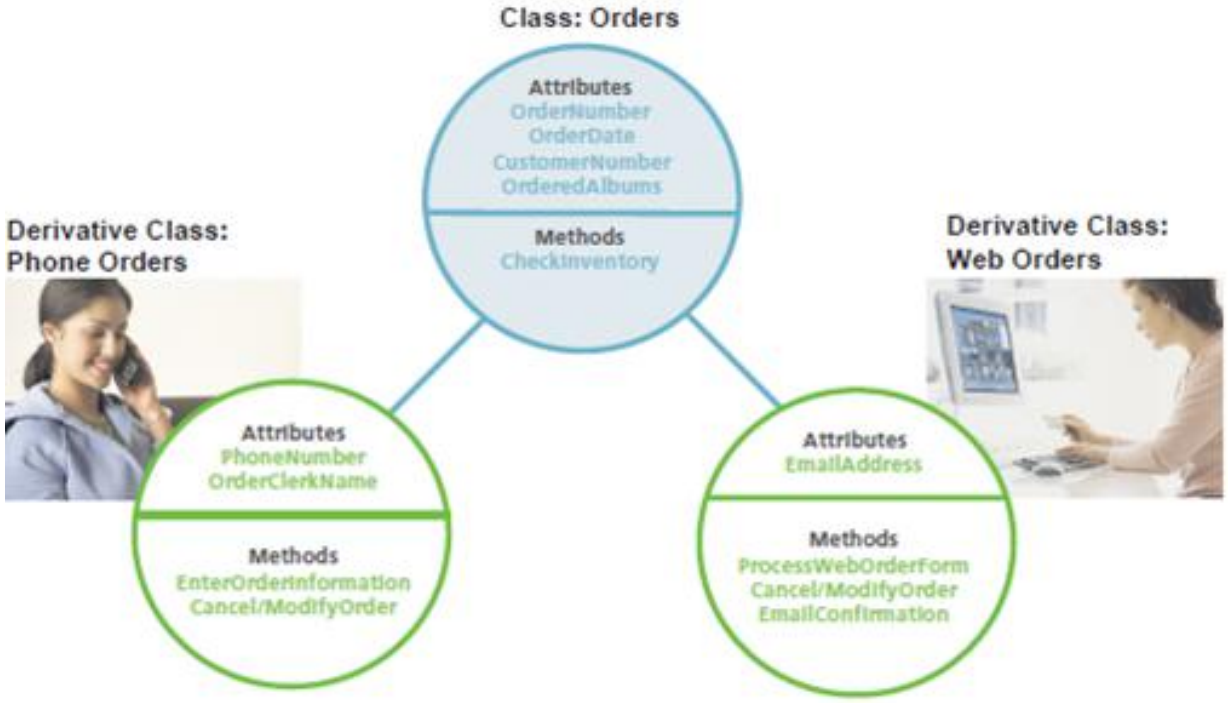
Database Models (11 of 13)



Database Models (12 of 13)

- An **object database**, also called an object-oriented database, stores data as objects, which can be grouped into classes and defined by attributes and methods
- Object databases excel at representing objects that have slightly different attributes, which is the case in many real-world business applications
- A **document-oriented** database stores unstructured data, such as the text of a speech
- **XML** (eXtensible Markup Language) is a popular tool used to format document databases

Database Models (13 of 13)



An object database can easily store data about different types of orders. A class called Orders holds data and methods common to all types of orders. A derivative class called Phone Orders inherits all the characteristics of Orders, but it has attributes and methods unique to orders placed by telephone. Web Orders is a derivative class that has attributes and methods unique to orders placed over the Web.

Section B: Database Tools

- Database Tool Basics
- Dedicated Applications
- Word Processor Data Tools
- Spreadsheet Data Tools
- Database Management Systems

Section B: Objectives (1 of 2)

- Explain why database designers try to avoid data dependence
- Provide examples of at least five dedicated applications that are based on databases
- Describe the data management features available in word processors
- Provide three examples of the kinds of data you might successfully store and manipulate with spreadsheet software

Section B: Objectives (2 of 2)

- Name two DBMSs that are relatively easy to use, and list four companies that produce an enterprise-level DBMS
- Explain how the concept of serializability relates to databases
- Describe the three categories of database clients
- List at least four security measures that are important for database management

Database Tool Basics (1 of 2)

- **Data dependence** is a term that refers to data and program modules being so tightly interrelated that they become difficult to modify
- Modern database tools support **data independence**, which entails separating data from the programs that manipulate data

Database Tool Basics (2 of 2)

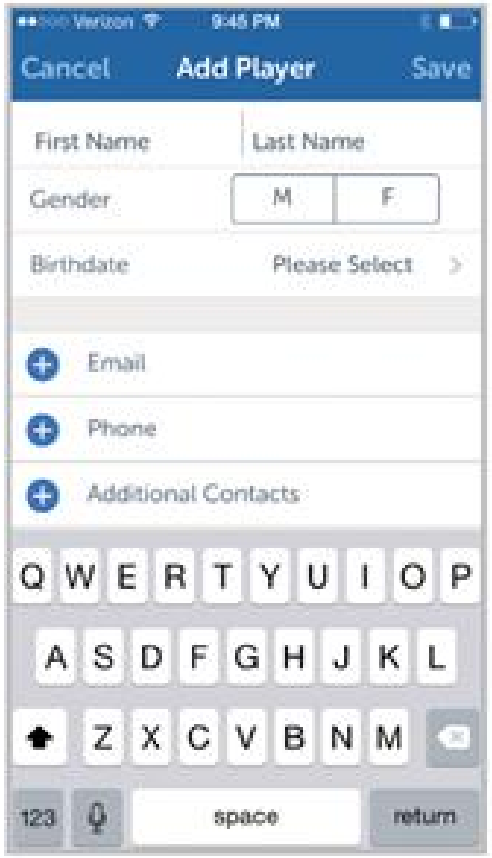
TOOL	COST	VERSATILITY	EASE OF USE
Dedicated software, such as an address book	Shareware available for simple applications is inexpensive; dedicated software for business applications can be costly,	Normally, the software is dedicated to a single type of database.	Easy; minimal setup is required because fields are predefined.
Word processing software	Most consumers have word processing software.	The software is best for simple flat files, such as mailing lists.	Easy; the software uses an interface familiar to most users.
Spreadsheet software	Most consumers have spreadsheet software.	The software is best for simple flat files that involve calculations.	Easy; the software uses an interface familiar to most users.
Database software	Basic shareware database software is inexpensive; high-end database software can be expensive.	High-end packages provide excellent versatility.	High-end database software often has a steep learning curve.

Dedicated Applications (1 of 3)

- The simplest tools for managing data are **dedicated applications** for specific data management tasks, such as keeping track of appointments or maintaining an address book
- To use one of these tools, simply enter your data. The software includes menus that allow you to manipulate your data once it is entered
- Dedicated applications are easy to use; however, they generally don't allow users to add fields or change field names

Dedicated Applications (2 of 3)

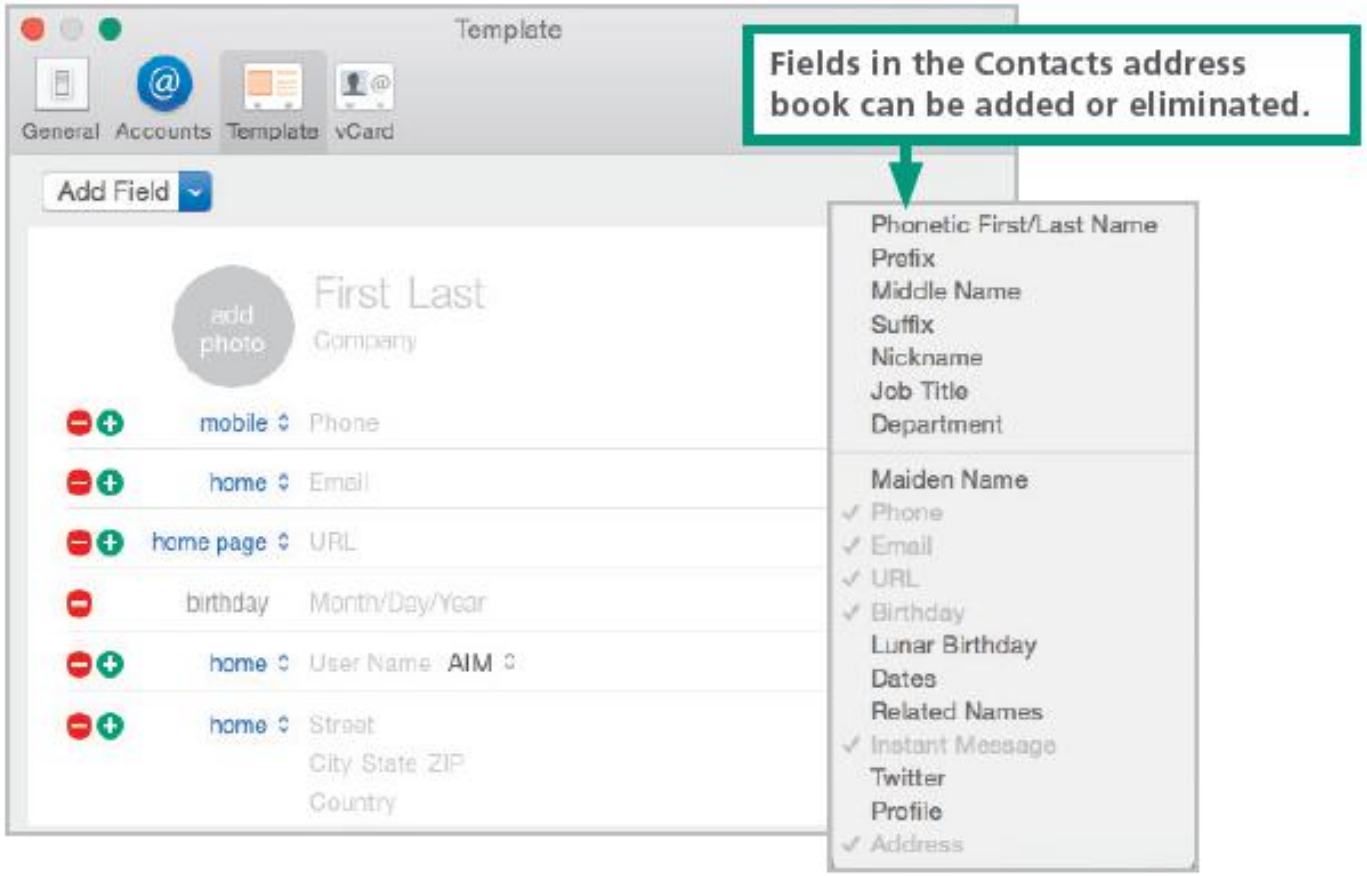
Source: Warranty Purchase tracker/Todd, Michael & James, Inc.



Source: TeamSnap/www.teamsnap.com

Source: Movingday/Mighty Ants

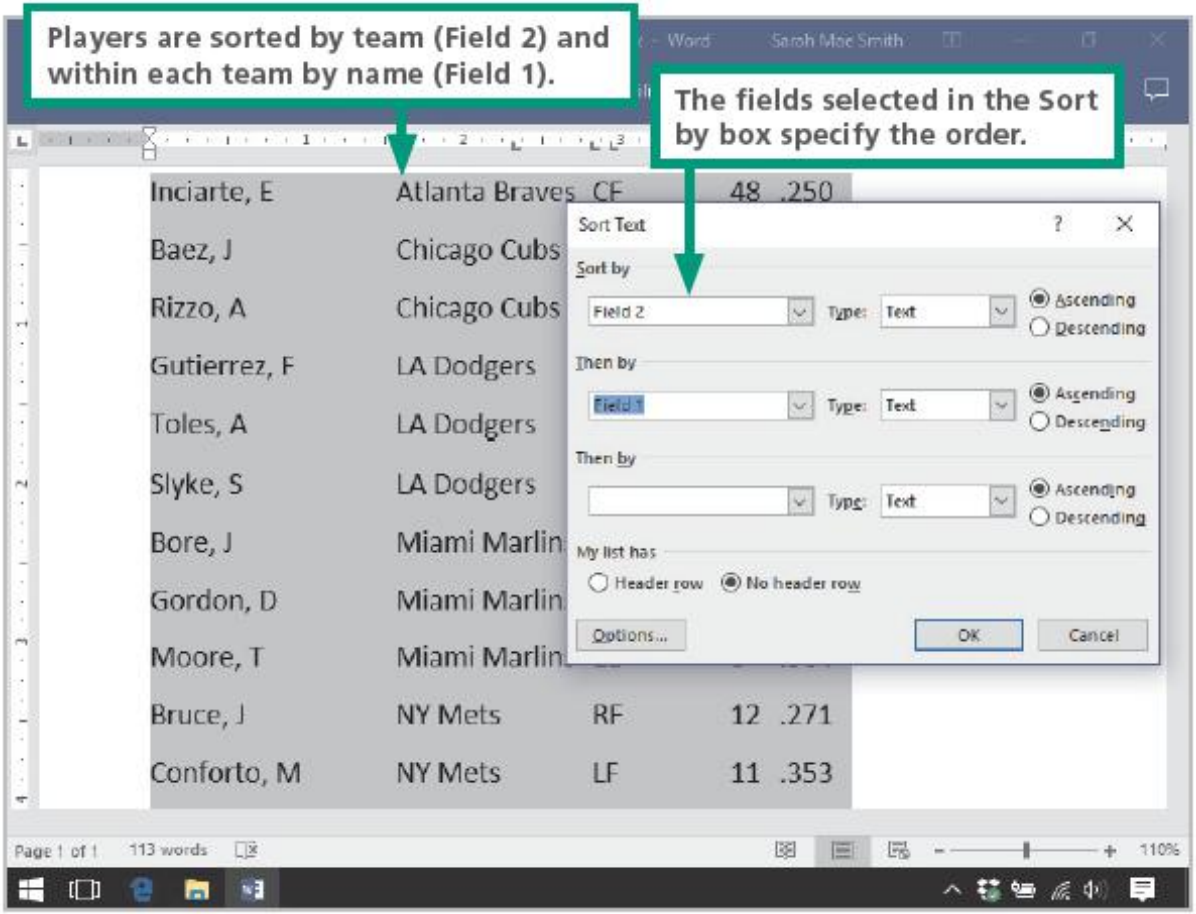
Dedicated Applications (3 of 3)



Word Processor Data Tools (1 of 2)

- Word processing software is designed to produce documents, but it also may include tools for working with unstructured or structured data
- Most word processing software includes a sort feature that can be used to arrange a simple list in alphabetical or numeric order
- A **single-level sort** uses only one field to arrange records
- A **multi-level sort** arranges information by more than one field

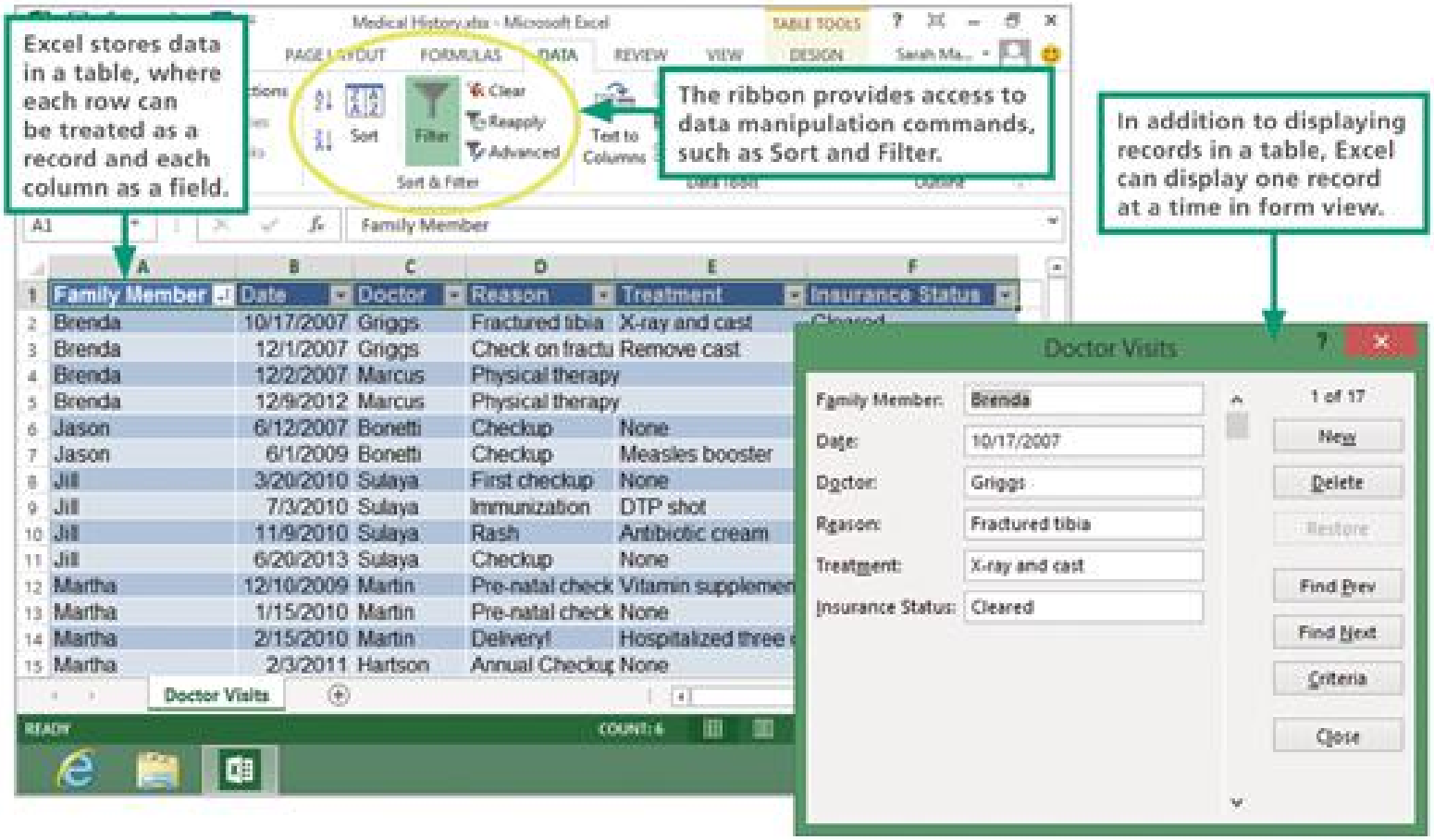
Word Processor Data Tools (2 of 2)



Spreadsheet Data Tools (1 of 2)

- Spreadsheets are organized in table format, so it makes sense that they can be used for sorting data
- Depending on the spreadsheet software, it may be possible to sort records, validate data, search for records, perform simple statistical functions, and generate graphs based on the data

Spreadsheet Data Tools (2 of 2)



Database Management Systems (1 of 4)

- When a word processor or spreadsheet isn't sufficient to handle a data set, a DBMS is an option that offers a set of development tools for creating and accessing databases
- The term **DBMS** (database management system) refers to software that manages data stored in a database
- Filemaker Pro and Microsoft Access are easy-to-use DBMSs that are a good fit for small businesses and individuals

Database Management Systems (2 of 4)

- Modern DBMSs work with many kinds of data including text, numbers, images, PDFs, and audio files
- Today, databases might reside on a in-house server, or on a cloud-based server, or on distributed servers scattered throughout the world
- DBMSs handle the details of how to most efficiently arrange data on a storage medium for optimal access speed
- DBMSs require security features to ensure data confidentiality, protect against insider threats, and block unauthorized access

Database Management Systems (3 of 4)

Prevention

- User rights management
  - Allows access to data on an asneeded basis
- Encryption
  - Scrambles data that resides in storage, so that it is useless to thieves who acquire stolen devices containing databases
- Database assessment
  - Identifies sensitive data and database vulnerabilities in order to secure them

Database Management Systems (4 of 4)

- Redaction
  - Masks confidential data such as credit card numbers
- Intermediary servers
  - Prevent users from directly accessing the database, but instead allow users access only to a query processor

Section C: Database Design

- Defining Fields
- Data Types
- Normalization
- Sorting and Indexing
- Designing the Interface
- Designing Report Templates

Section C: Objectives

- Explain how computed fields save storage space
- List three common data entry errors and the strategies that database designers use to reduce the number of them
- List eight data types and provide a real-world example for each one
- Explain why database designers normalize a database and how they do it
- Distinguish the difference between a sort and an index
- List at least three best principles for designing database interfaces
- Explain the purpose of a report template

Defining Fields (1 of 2)

- There are three core elements in a relational database: **fields**, **tables**, and **relationships**
- The term **database structure** refers to the arrangement of fields, tables, and relationships in a database
- The first step in structuring a relational database is to determine what data should be collected and stored
- A **computed field** is a calculation that a DBMS performs, similar to the way a spreadsheet computes a formula
- A **field format** is a template that adds the correct formatting as data is entered

Defining Fields (2 of 2)

- A **field validation** rule is a specification that the database designer sets up to filter data entered into a particular field
- A **lookup routine** validates a field entry by checking data in an in-house or third-party database

Data Types (1 of 2)

- The data that can be entered into a field depends on the field's data type
- A **data type** specifies the way data is represented on physical storage media and RAM
- Data types:
  - **Real** – used for fields that contain numbers with decimal places
  - **Integer** – used for fields that contain whole numbers
  - **Date** – stores dates in a format that allows them to be manipulated

Data Types (2 of 2)

- **Text** – assigned to fixed-length fields that hold character data
- **Memo** – provides a variable-length field for user comments
- **Logical (Boolean)** – used for true/false and yes/no data
- **BLOB (binary language object)** – can be any type of data
- **Hyperlink** – stores URLs used to link from a database to a Web page

Normalization

- A process called **normalization** helps database designers create a database structure that minimizes storage space and increases processing efficiency
- The goal of normalization is to minimize **data redundancy**—the amount of data that is duplicated in a database

Sorting and Indexing

- A table's **physical sort order** is the order in which data is arranged on storage devices
- A **sort key** is the column of data that is used as the basis for rearranging the data
- Sorted tables produce faster queries and updates using clever algorithms to find data
- A **database index** contains a list of keys, and each key provides a pointer to the data that contains the rest of the fields related to that key

Designing the Interface (1 of 3)

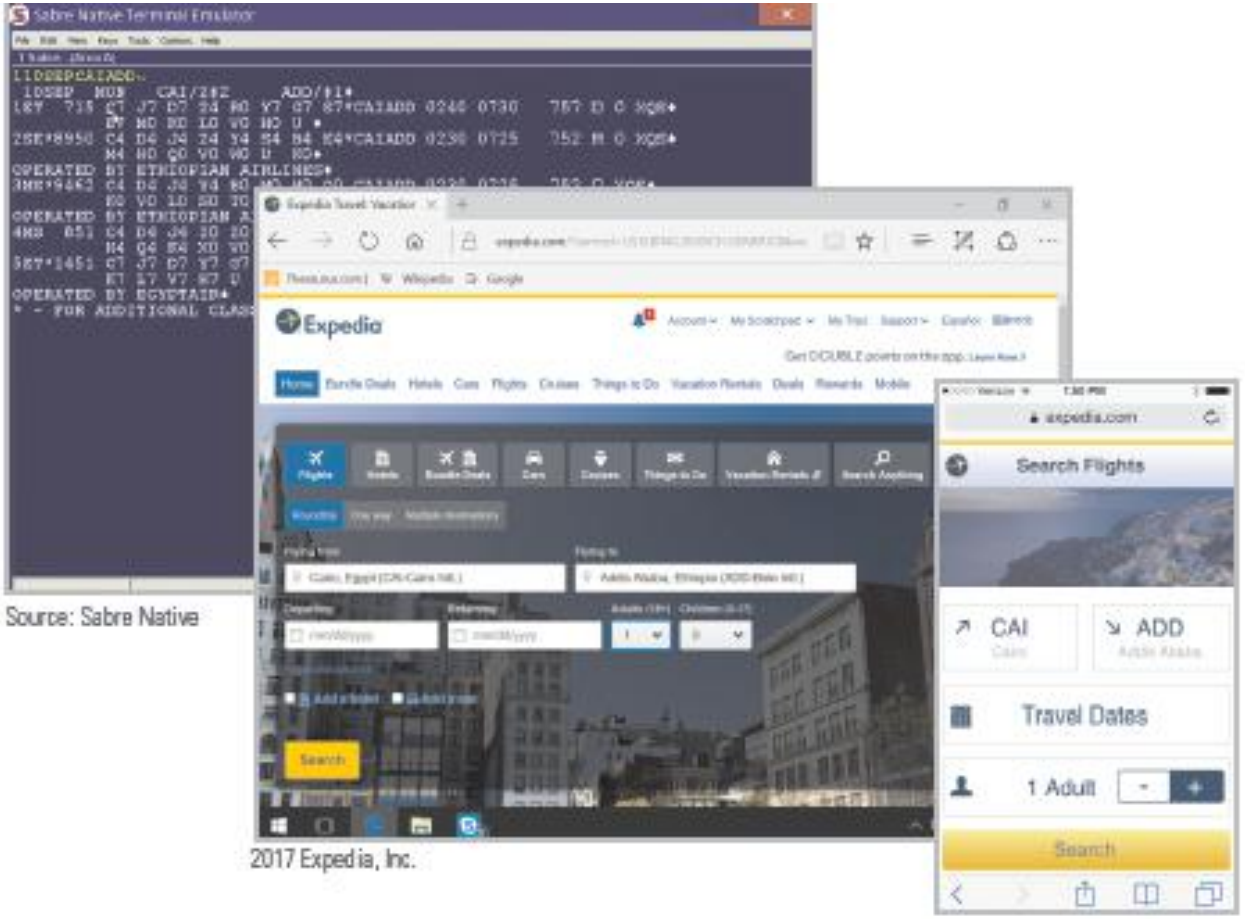
The following guidelines list strategies for producing well-designed database interfaces:

- Arrange fields in a logical order beginning at the upper-left corner of the screen. The first fields should be those used most often or those that come first in the data entry sequence
- Provide visual clues to the entry areas. A box, a line, or shading can delineate data entry areas
- Entry areas should appear in a consistent position relative to their labels
- By convention, labels are placed to the left of the entry areas or above them

Designing the Interface (2 of 3)

- Provide a quick way to move through the fields in order. By convention, the Tab key performs this function on desktop and laptop computers
- If all fields do not fit on a single screen, use scrolling or create a second screen
- Provide buttons or other easy-to-use controls for moving from one record to another
- Stay aware of the platform; controls for a touchscreen device have to be large, well spaced, and easy to operate
- Supply on-screen instructions to help ensure that data is entered correctly. Web databases can benefit from links to help pages

Designing the Interface (3 of 3)



Source: Sabre Native

2017 Expedia, Inc.

Designing Report Templates

- A report is a printed or screen-based list of some or all of the data in a database
- Most DBMSs include a **report generator**, which is a software tool for specifying the content and format for a database report
- A **report template** contains the outline or general specifications for a report

Section D: SQL

- SQL Basics
- Adding Records
- Searching for Information
- Updating Fields
- Joining Tables

Section D: Objectives

- Explain how SQL relates to databases
- Describe how SQL injection works
- List at least five SQL command words and describe what they do
- Describe how to use Boolean operators in a query and provide examples
- Provide an example in which a global update is used efficiently
- Illustrate how SQL can be used to join two tables using dot notation

SQL Basics (1 of 3)

- Commands processed by the DBMS are issued using computer programming languages designed for databases
- These languages are sometimes called **query languages** because one of their main capabilities is to request data from a database
- The database client software collects input from the user and then converts it into an **SQL query**, which can operate directly on the database to carry out the user's instructions

SQL Basics (2 of 3)

- The SQL query language provides a collection of special command words called **SQL keywords**, such as SELECT, FROM, INSERT, and WHERE
- Most SQL queries can be divided into three simple elements that specify an action, the name of a database table, and a set of parameters
- An SQL query begins with an action keyword, or **command**, which specifies the operation you want carried out
- **Parameters** are detailed specifications for a command

SQL Basics (3 of 3)

COMMAND	DESCRIPTION	EXAMPLE
CREATE	Create a database or table.	CREATE TABLE Albums
DELETE	Remove a record from a table.	DELETE FROM Tracks WHERE TrackTitle = 'Blue Suede Shoes'
INSERT	Add a record.	INSERT INTO AlbumDescription (Cat#, Condition) VALUES ('LPM-2256', 'Mint condition; no visible scratches; original album cover')
JOIN	Use the data from two tables.	SELECT FROM Albums JOIN Tracks ON Albums.Cat# = Tracks.Cat#
SELECT	Search for records.	SELECT FROM Albums WHERE Artist = 'Beatles'
UPDATE	Change data in a field.	UPDATE Albums SET Price = 15.95 WHERE Cat# = 'LPM-2256'

Adding Records

- A database record contains information about an entity, such as a customer, an online purchase, an ATM withdrawal, or a social media post
- The data is bundled into an SQL statement that is handled by the DBMS
- Using the **INSERT** command, a user can add data to a record

Searching for Information

- One of the most common database operations is to query for particular record or a group of records by using the **SELECT** command
- The database client software uses a search specification to create the SQL query; a result is generated for this query
- SQL uses **Boolean operators** such as AND, OR, and NOT to form complex queries

Updating Fields

- Updates and modifications to the contents of a database field are made by using the SQL **UPDATE** command
- The UPDATE function works only for records that have similar characteristics
- Custom programming is required to perform global operations on information that does not have any similar characteristics

Joining Tables

- In SQL terminology, creating a relationship between tables is referred to as **joining tables**
- The SQL **JOIN** command allows users to temporarily join and simultaneously access the data in more than one table
- When joining two tables, the convention is to use dot notation for field names; SQL uses dot notation to make distinctions between data

Section E: Big Data

- Big Data Basics
- Big Data Analytics
- NoSQL

Section E: Objectives (1 of 2)

- List the elements that define the 3rd platform of computing
- List the five Vs that characterize big data
- Describe at least three examples of datasets that would be considered big data
- List the four characteristics of NoSQL
- Explain the difference between scaling up and scaling out
- Explain how dynamic scaling works and how it relates to big data

Section E: Objectives (2 of 2)

- Support or refute the statement that NoSQL tools are schema-less
- Describe and diagram an example of a key-value data model
- Demonstrate the different retrieval strategies for data stored in a relational database and data stored in a column-oriented database
- Give at least three examples of large datasets that would be best handled by a graph schema
- Describe Hadoop and MapReduce

Big Data Basics (1 of 2)

- **Big data** refers to the huge collections of data that are difficult to process, analyze, and manage using conventional database tools
- An example of big data is the 1 million transactions generated by Walmart sales registers every hour
- Big data is a relatively new phenomenon that businesses are just beginning to deal with

Big Data Basics (2 of 2)

- Big data is characterized as having:
  - High volume
  - High velocity
  - Diversified variety
  - Unknown veracity
  - Low-density value (low-density data refers to large volumes of data containing unimportant details)

Big Data Analytics (1 of 3)

- Mainstream big data exploration produces commercial benefits
- A high percentage of today's expenditures on big data are for technologies that enhance the customer experience and provide targeted marketing solutions
- Real-time analysis and decision making are popular reasons to invest in big data technologies

Big Data Analytics (2 of 3)

- Government
  - Threat prediction
  - Cybersecurity
  - Compliance and regulatory analysis
- Retail
  - Shopper behavior analysis
  - Loyalty program management
  - Supply chain optimization

Big Data Analytics (3 of 3)

- Health care
  - Track infectious diseases
  - Genetic analysis
  - Design proactive care plans
- Communications
  - Retain customers
  - Call record analysis
  - Infrastructure optimization

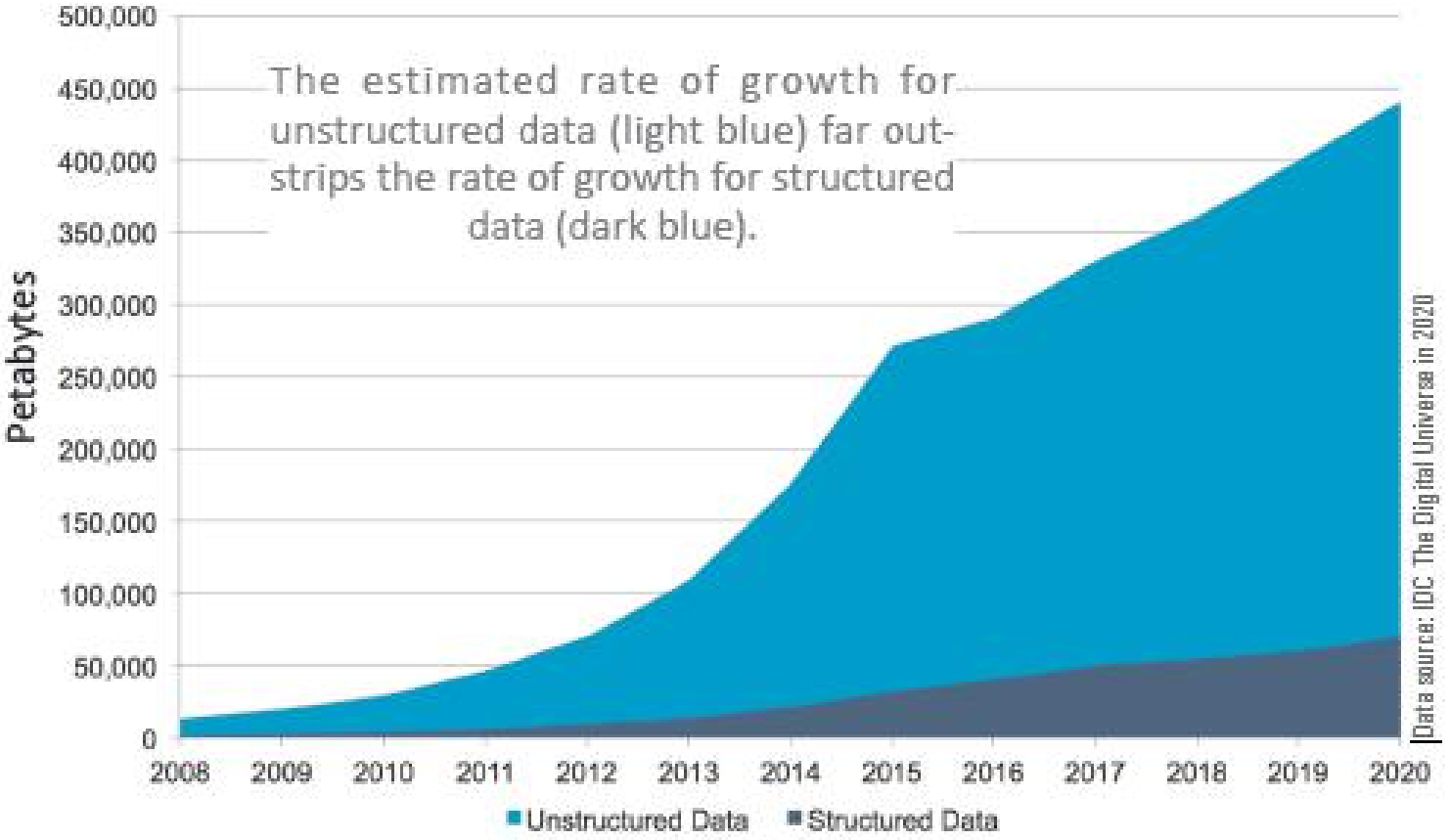
NoSQL (1 of 6)

- The term **NoSQL** is used to refer to a group of technologies for managing databases that do not adhere to the relational model and standard SQL query language
- NoSQL technologies are effective for building and managing non-relational databases containing big data that may be unstructured and may be distributed across multiple servers

NoSQL (2 of 6)

- Distributed
  - Handles data that is stored across many devices
- Dynamically scaling
  - Easy to add storage devices as the database grows or as the velocity of incoming data accelerates
- Flexible data
  - Handles a variety of data types, as well as data that is structured, semi-structured, and unstructured
- Non-relational
  - Uses data models other than the standard relational models and SQL

NoSQL (3 of 6)



NoSQL (4 of 6)

- Unstructured and semi-structured data—such as tweets, email messages, blog posts, and videos—are difficult to mold into fixed structures
- Relational databases are organized according to a **schema**, which is a blueprint for its structure; rows, columns, and tables of a database are part of its schema
- NoSQL tools create **schema-less databases**, allowing data structures such as fields to be added

NoSQL (5 of 6)

- The simplest structure for storing data in a NoSQL database is the **key-value data model**; each data item has a key that is a unique identifier similar to a relational database key such as CustomerID
- The **column-oriented data model** stores data in columns, rather than in rows, so it works well in situations where the focus is on analysis of chunks of data

NoSQL (6 of 6)

- Popular NoSQL tools include:
  - MongoDB
  - Cassandra
  - Hbase
  - Hive
  - Presto
  - Google BigTable
  - Spark
  - Voldemort
- The two most popular are Hadoop and MapReduce