CSCI-C442 Database Systems

Do a self-check

Download resource files?

Read the syllabus?

Find class schedule table?

- Canvas
- dbsguide: https://cisjw.sitehost.iu.edu/dbsguide/

Week 1

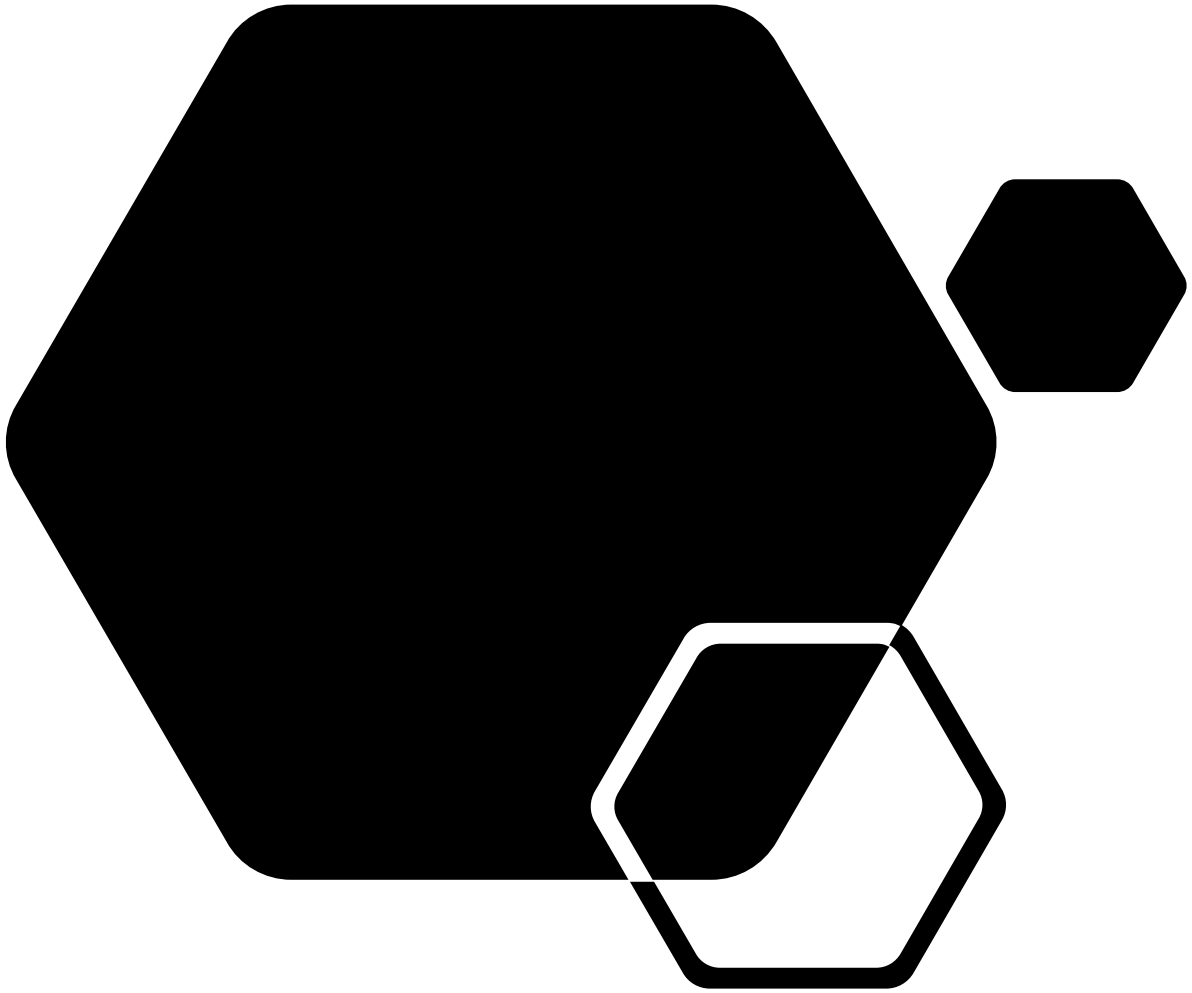
- P1

Week 2:

- Start the SQLite project

Chapter 1

Roles, Database design



The things you can do

Three design scenarios

Roles in a DBS

User	Database Administrator	Programmer developer software engineer	(database, systems) Designer
Data Engineer	Data Analyst (*Data Scientist*)	Product Manager	Machine learning engineer

1. From Existing Data

Someone emailed you a few spreadsheets or several tables in a word doc. Ask you to create a database from them.

How?

Designer

Determine metadata (table structure, column type, constraints)

Split them or join them?

Visualize the model

2. For New Systems

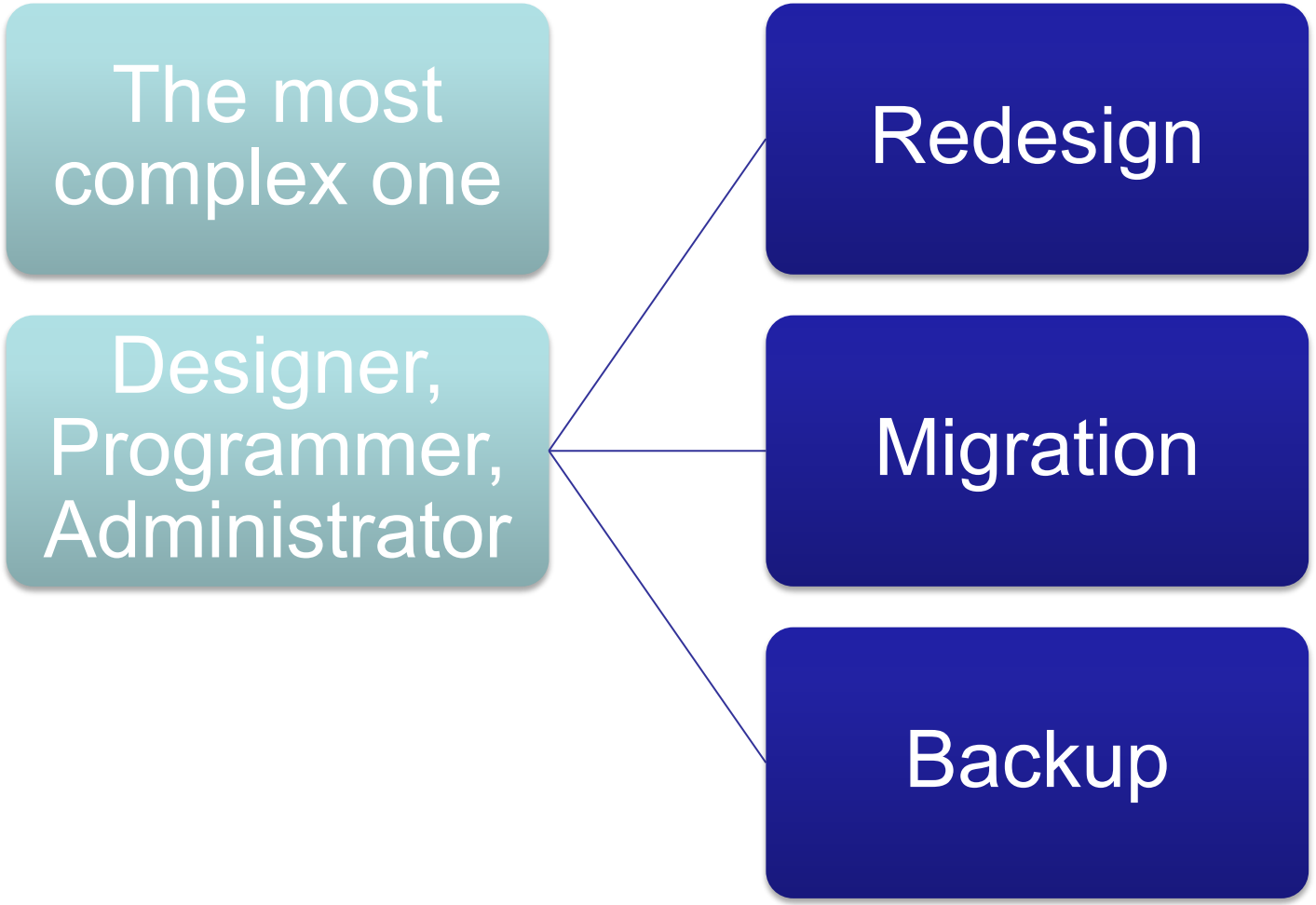
Systems Analyst and Designer

- Requirement analysis
- Data modeling
  - ERD, Enhanced ERD, Class Diagram
- Database design

Programmer

- Database implementation

3. For New Requirements



Chapter 1 Objectives

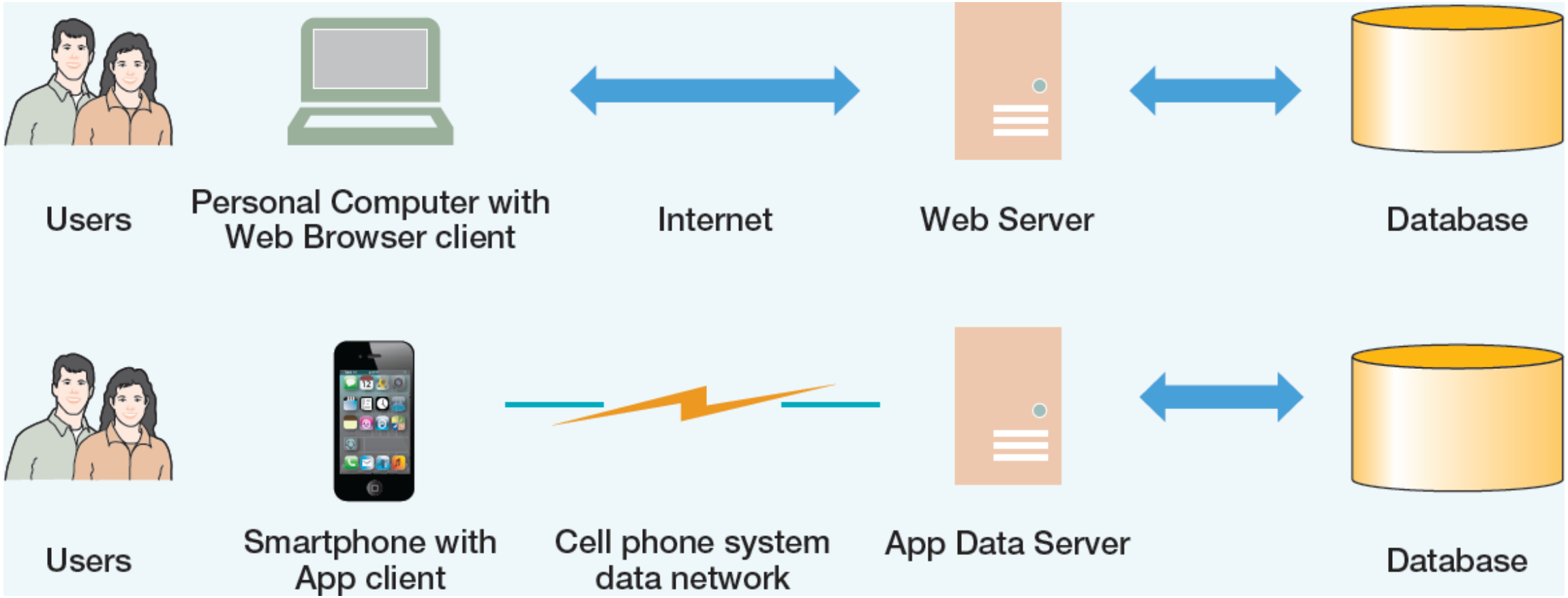
- To understand the importance of databases in Internet Web applications and smartphone apps
- To understand *the nature and characteristics of databases*
- To survey some important and interesting database applications
- To gain a general understanding of tables and relationships
- To describe the components of a database system and explain the functions they perform
- To describe the components of an enterprise-class database system and explain the functions they perform

Chapter 1 Objectives *cont.*

- To define the term *database management system (DBMS)* and describe the functions of a DBMS
- To define the term *database* and describe what is contained within the database
- To define the term *metadata* and provide examples of metadata
- *To define and understand database design from existing data*
- *To define and understand database design as new systems development*
- *To define and understand database design in database redesign*

THE IMPORTANCE

The Internet and Mobile Device World
Client-Server Architecture



Databases in the Internet, IoT, Mobile Apps

- Information sharing:
  - Social media
    - Facebook: Posts Likes
    - Twitter: Tweets
- E-Commerce & Online shopping
  - Amazon.com - [Do an actual search]
- Information fusion:
  - Small and smart devices, sensors for Internet of Things (IoT)

How Did We Get Here?
The Internet World I

- Personal Computers
  - 1977: Apple II
  - 1981: IBM PC
- Local Area Networks
  - Ethernet networking technology
    - Early 1970s: Xerox Palo Alto Research Center
    - 1893: US National Standard

How Did We Get Here?
The Internet World II

- The Internet
  - 1969: ARPANET
- World Wide Web (WWW)
  - 1993: First Web browser (Netscape) available
  - Mid 1990's: Online retail sites
    - 1995: Amazon.com
- Early 2000's: Web 2.0

How Did We Get Here?
The Smartphone World

- Mid 1970s: Mobile Phone (Cell Phone)
- Smartphone
  - 2007: Apple iPhone
  - 2008: Google Android Operating System
- Tablets
  - 2010: Apple iPad
- Apps

WHAT IS THE DATABASE?

A base for data

The Characteristics of Databases

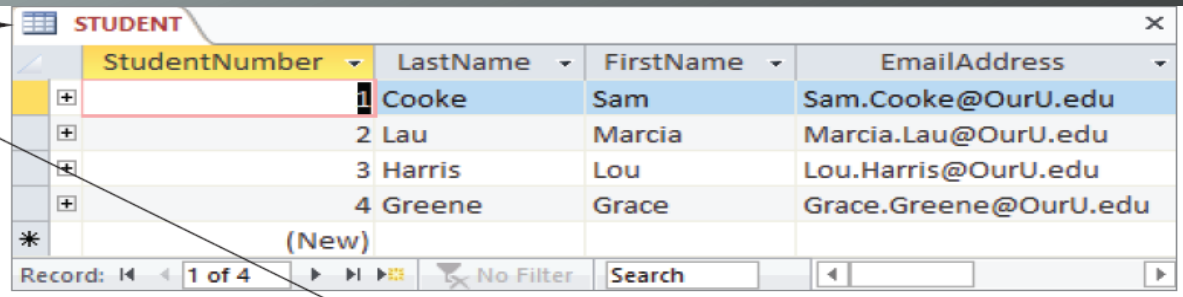
- The purpose of a **database** is to help people *track things of interest to them*.
- Data is stored in ***tables***, which have rows and columns like a spreadsheet.
- A database may have multiple tables, where each table stores data about a different thing.
- Each row in a table stores data about an occurrence or ***instance*** of the thing of interest.
- A database stores ***data*** and ***relationships***.

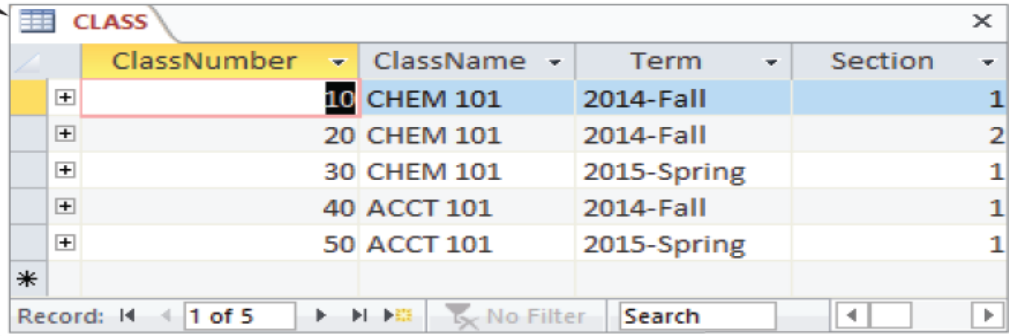
Data in Tables (MS Access)

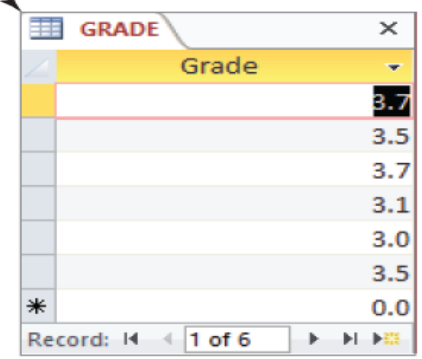
The STUDENT table

The CLASS table

The GRADE table —but who do these grades belong to?







A **Primary Key** is a unique identifier field within a table.

A **Surrogate Key** is a primary key field that is automatically assigned by the computer. An example is the StudentNumber field in the STUDENT table above.

The Key Characteristic of Databases: Related Tables

The STUDENT table

The CLASS table

The GRADE table with foreign keys—now each grade is linked back to the STUDENT and CLASS tables

Foreign Key = A Primary Key in one table that is used as an identifier in another table

STUDENT

StudentNumber	LastName	FirstName	EmailAddress
1	Cooke	Sam	Sam.Cooke@OurU.edu
2	Lau	Marcia	Marcia.Lau@OurU.edu
3	Harris	Lou	Lou.Harris@OurU.edu
4	Greene	Grace	Grace.Greene@OurU.edu
(New)			

Record: 1 of 4 No Filter Search

CLASS

ClassNumber	ClassName	Term	Section
10	CHEM 101	2014-Fall	1
20	CHEM 101	2014-Fall	2
30	CHEM 101	2015-Spring	1
40	ACCT 101	2014-Fall	1
50	ACCT 101	2015-Spring	1

Record: 1 of 5 No Filter Search

GRADE

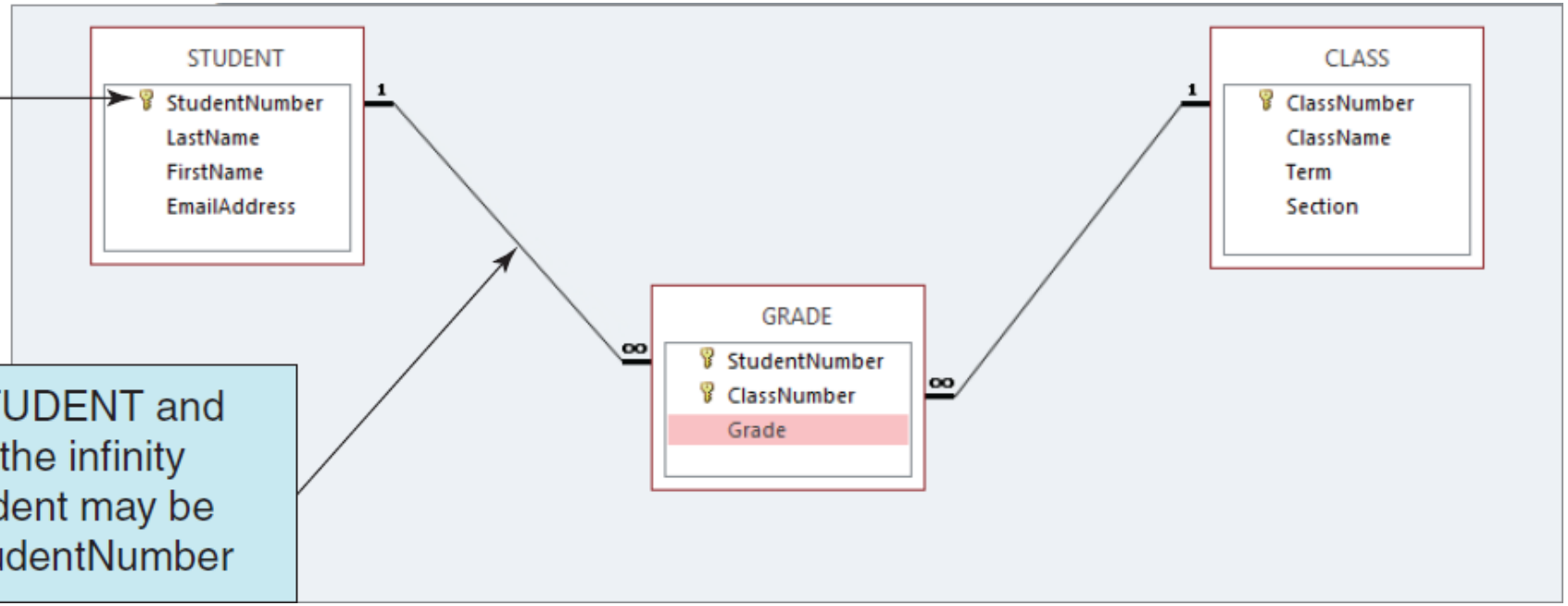
StudentNumber	ClassNumber	Grade
1	10	3.7
1	40	3.5
2	20	3.7
3	30	3.1
4	40	3.0
4	50	3.5
		0.0

Record: 1 of 6 No Filter Search

A **Composite Key** is a primary key that consists of 2 or more attributes that uniquely identify an entity occurrence.

Microsoft Access 2013
Tables and Relationships

The STUDENT table—the key symbol shows the primary key



The relationship between STUDENT and GRADE—the number 1 and the infinity symbol indicate that one student may be linked to many grades by StudentNumber

Naming Conventions in this Textbook

- **Table names** are written with all capital letters:
  - STUDENT, CLASS, GRADE, COURSE_INFO
- **Column names** are written with an initial capital letter, and compound names are written with a capital letter on each word:
  - Term, Section, ClassNumber, StudentName

Databases Create Information

Data = recorded facts and figures

Information = knowledge derived from data

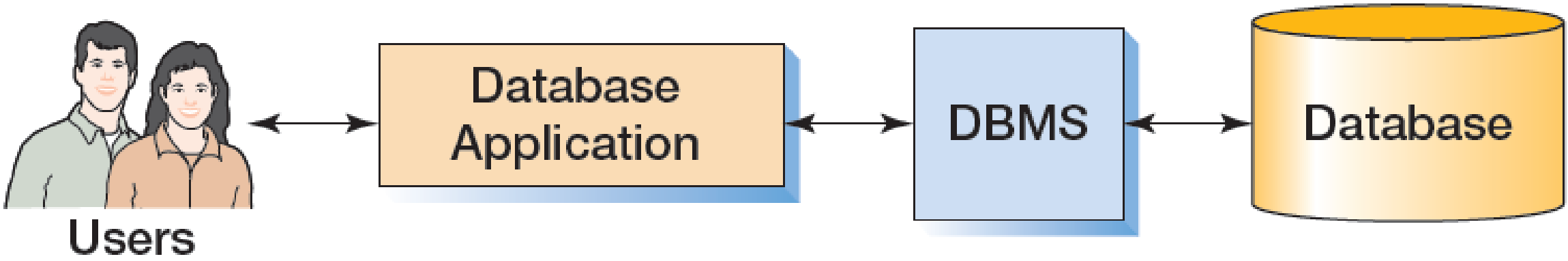
Databases record data, but they do so in such a way that we can produce information from the data.

The data on STUDENTs, CLASSes, and GRADEs could produce information about each student's GPA.

Database Examples

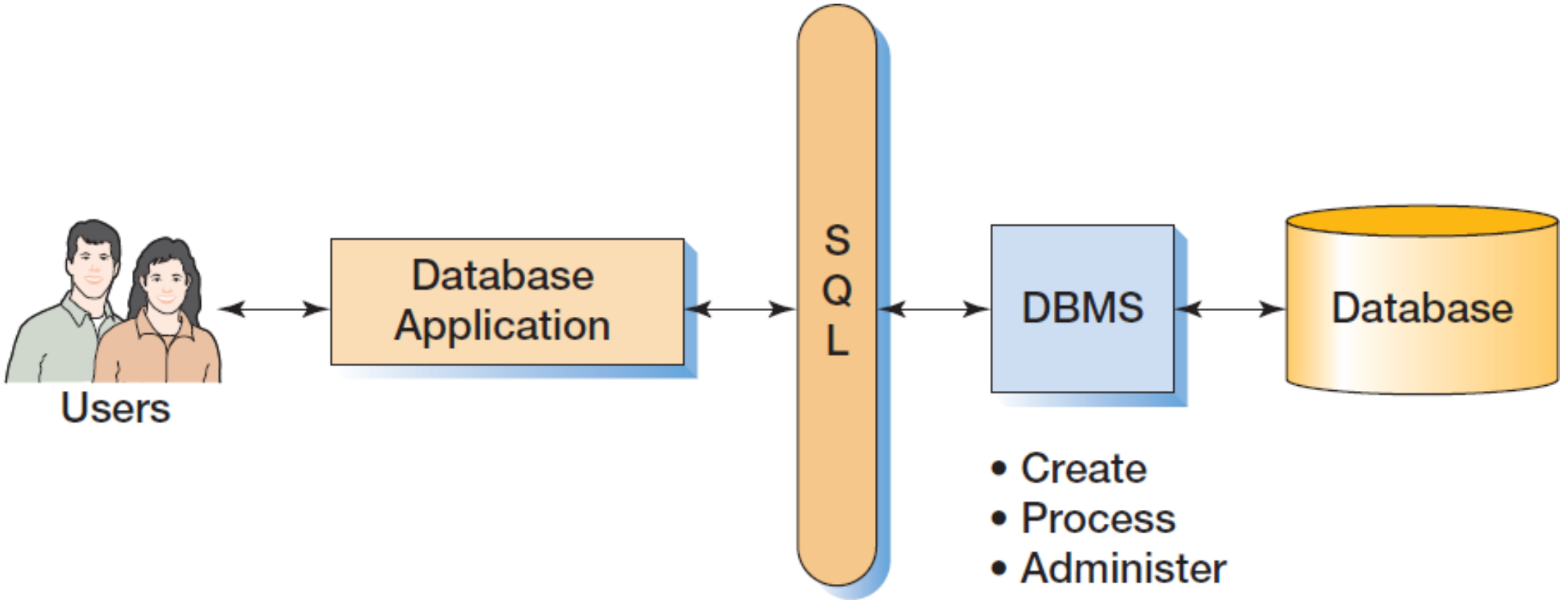
Application	Example Users	Number of Users	Typical Size	Remarks
Sales contact manager	Salesperson	1	2,000 rows	Products such as GoldMine and Act! are database centric.
Patient appointment (doctor, dentist)	Medical office	15 to 50	100,000 rows	Vertical market software vendors incorporate databases into their software products.
Customer relationship management (CRM)	Sales, marketing, or customer service departments	500	10 million rows	Major vendors such as Microsoft and Oracle PeopleSoft Enterprise build applications around the database.
Enterprise resource planning (ERP)	An entire organization	5,000	10 million+ rows	SAP uses a database as a central repository for ERP data.
E-commerce site	Internet users	Possibly millions	1 billion+ rows	Drugstore.com has a database that grows at the rate of 20 million rows per day!
Digital dashboard	Senior managers	500	100,000 rows	Extractions, summaries, and consolidations of operational databases.
Data mining	Business analysts	25	100,000 to millions+	Data are extracted, reformatted, cleaned, and filtered for use by statistical data mining tools.

DATABASE SYSTEM, APPLICATION, DBMS, SQL



- Create
- Process
- Administer

Components of a Database System



Components of a Database System with SQL

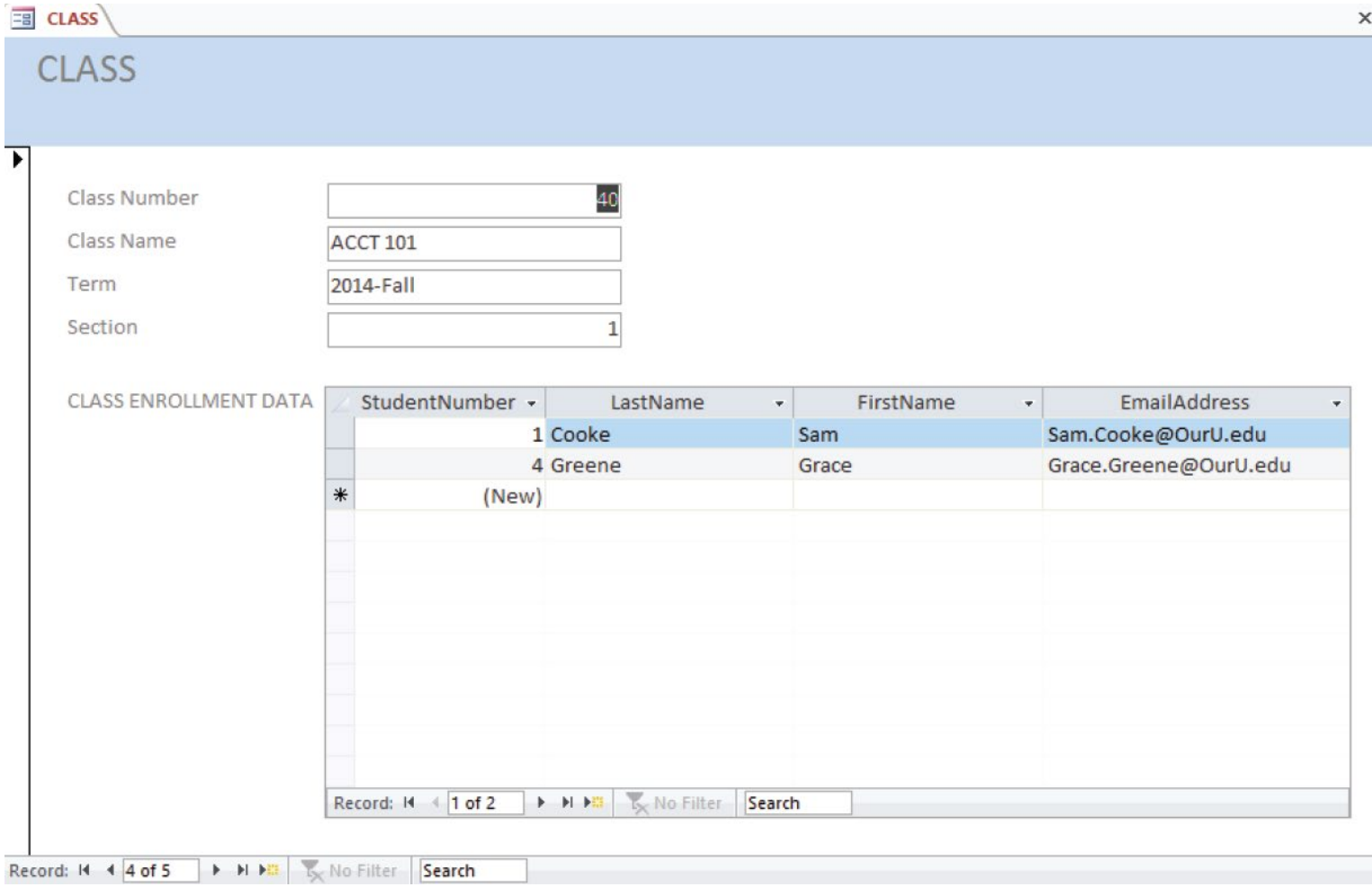
Applications, the DBMS, and SQL

- **Applications** are the computer programs that users work with.
- The **Database Management System (DBMS)** creates, processes, and administers databases.
- **Structured Query Language (SQL)** is an internationally recognized standard database language that is used by all commercial DBMSs.

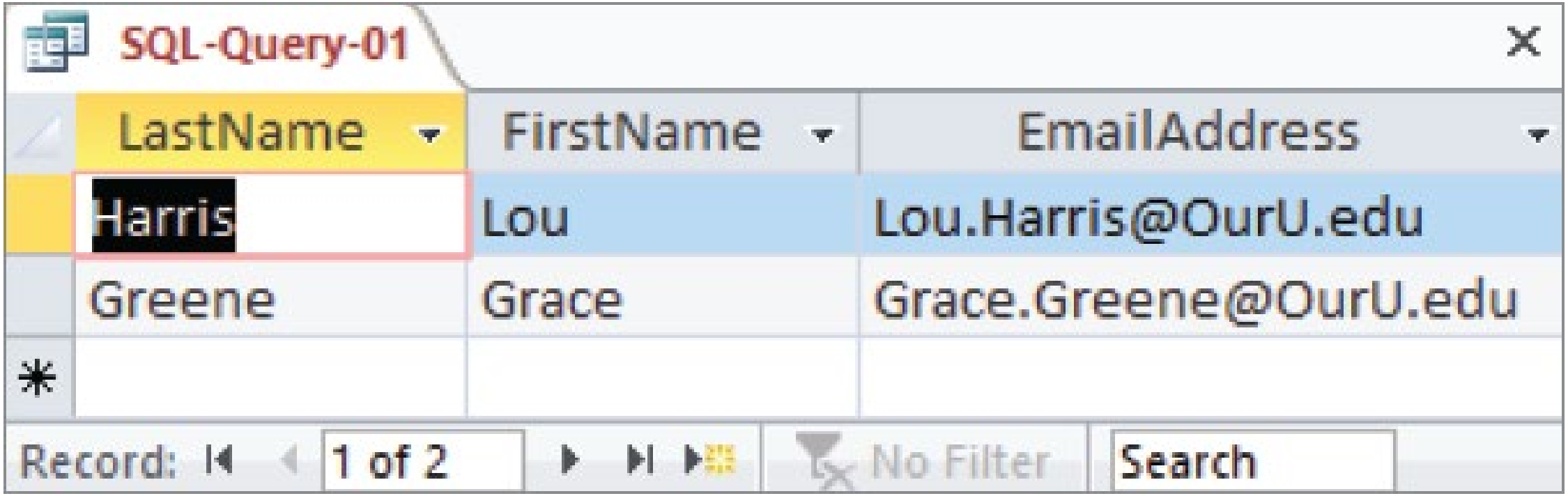
Database Applications

Basic Functions of Application Programs
Create and process forms
Process user queries
Create and process reports
Execute application logic
Control the application itself

Database Applications—Forms



Jie Wang 2021



Database Applications—Queries

- SELECT LastName, FirstName, EmailAddress
- FROM STUDENT
- WHERE StudentNumber > 2;

Class Grade Report

ClassNumber	ClassName	Term	Section	LastName	FirstName	Grade
10	CHEM 101	2014-Fall	1			
				Cooke	Sam	3.7
20	CHEM 101	2014-Fall	2			
				Lau	Marcia	3.7
30	CHEM 101	2015-Spring	1			
				Harris	Lou	3.1
40	ACCT 101	2014-Fall	1			
				Cooke	Sam	3.5
				Greene	Grace	3.0
50	ACCT 101	2015-Spring	1			

Database—Reports

The DBMS

Functions of a DBMS
Create database
Create tables
Create supporting structures (e.g., indexes)
Modify (insert, update, or delete) database data
Read database data
Maintain database structures
Enforce rules
Control concurrency
Perform backup and recovery

RELATIONAL DATABASE

The Database

- A **database** is a self-describing collection of integrated tables.
- The tables are called **integrated** because they store data about the relationships between the rows of data.
- A database is called **self-describing** because it stores a description of itself.
- The self-describing data is called **metadata**, which is data about data.

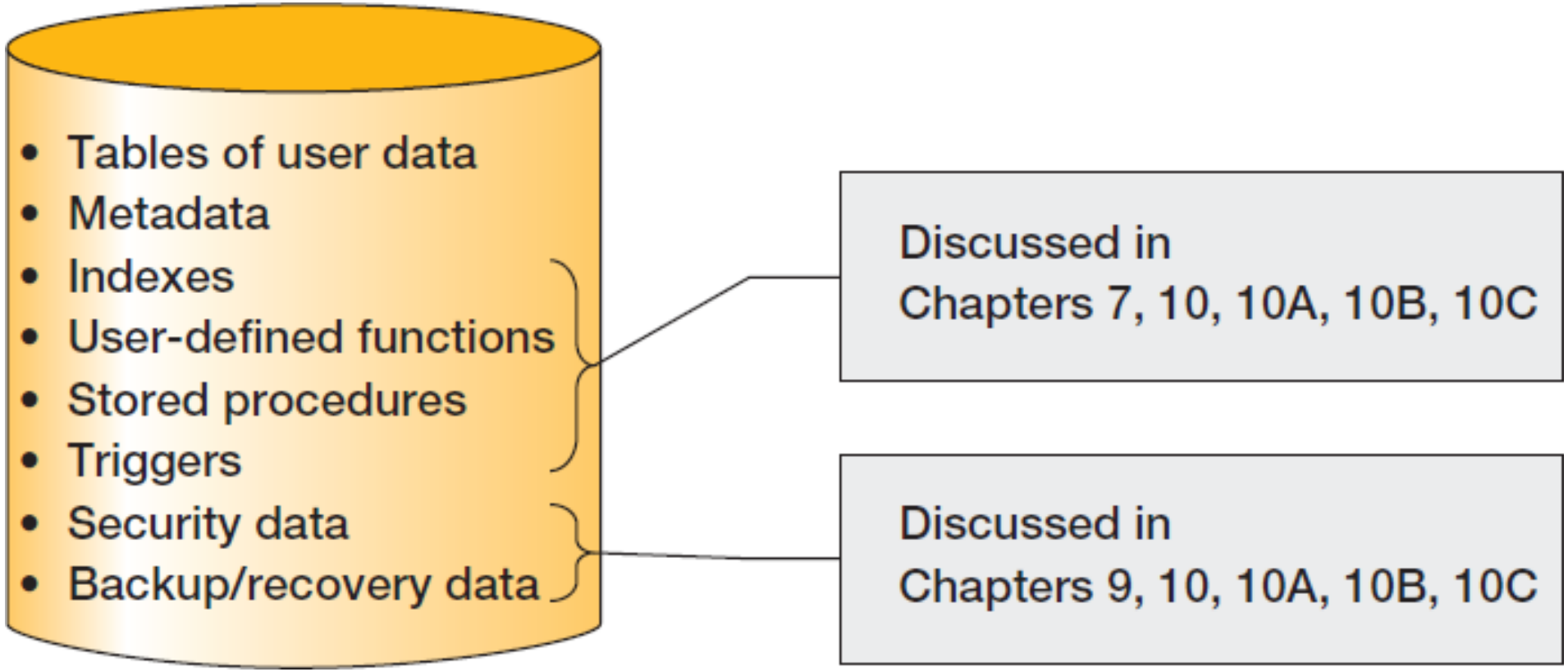
Typical Metadata Tables

USER_TABLES Table

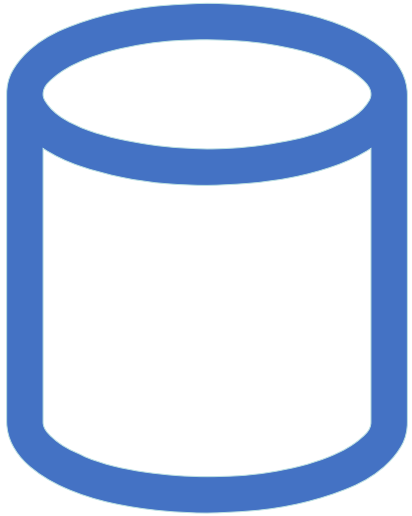
TableName	NumberColumns	PrimaryKey
STUDENT	4	StudentNumber
CLASS	4	ClassNumber
GRADE	3	(StudentNumber, ClassNumber)

USER_COLUMNS Table

ColumnName	TableName	DataType	Length (bytes)
StudentNumber	STUDENT	Integer	4
LastName	STUDENT	Text	25
FirstName	STUDENT	Text	25
EmailAddress	STUDENT	Text	100
ClassNumber	CLASS	Integer	4
Name	CLASS	Text	25
Term	CLASS	Text	12
Section	CLASS	Integer	4
StudentNumber	GRADE	Integer	4
ClassNumber	GRADE	Integer	4
Grade	GRADE	Decimal	(2, 1)

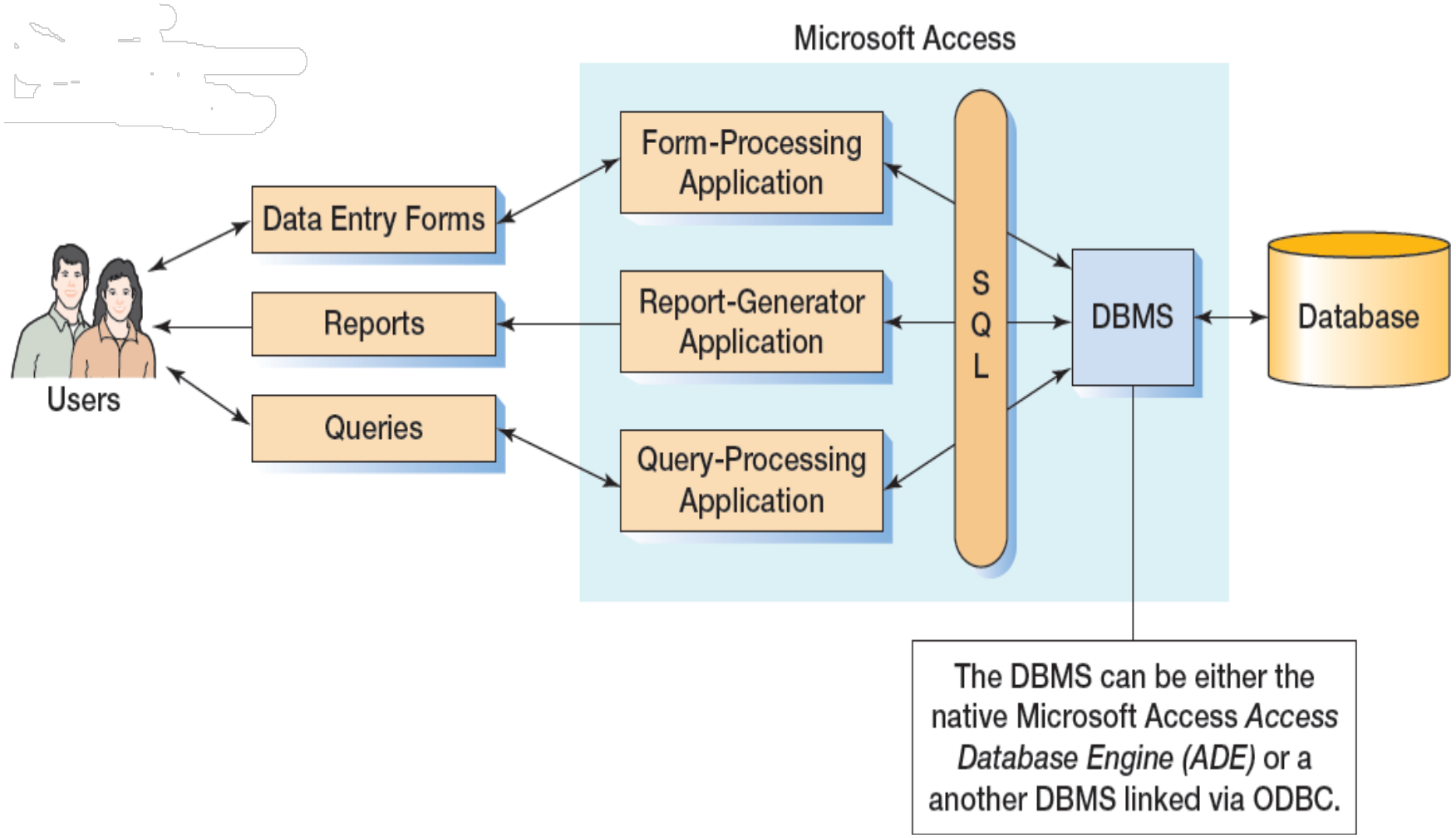


Database Contents



Types of Database Systems

Personal Database Systems: Microsoft Access



Microsoft Access

Microsoft Access is a low-end product intended for individual users and small workgroups.

Microsoft Access tries to hide much of the underlying database technology from the user.

A good strategy for beginners, but not for database professionals.

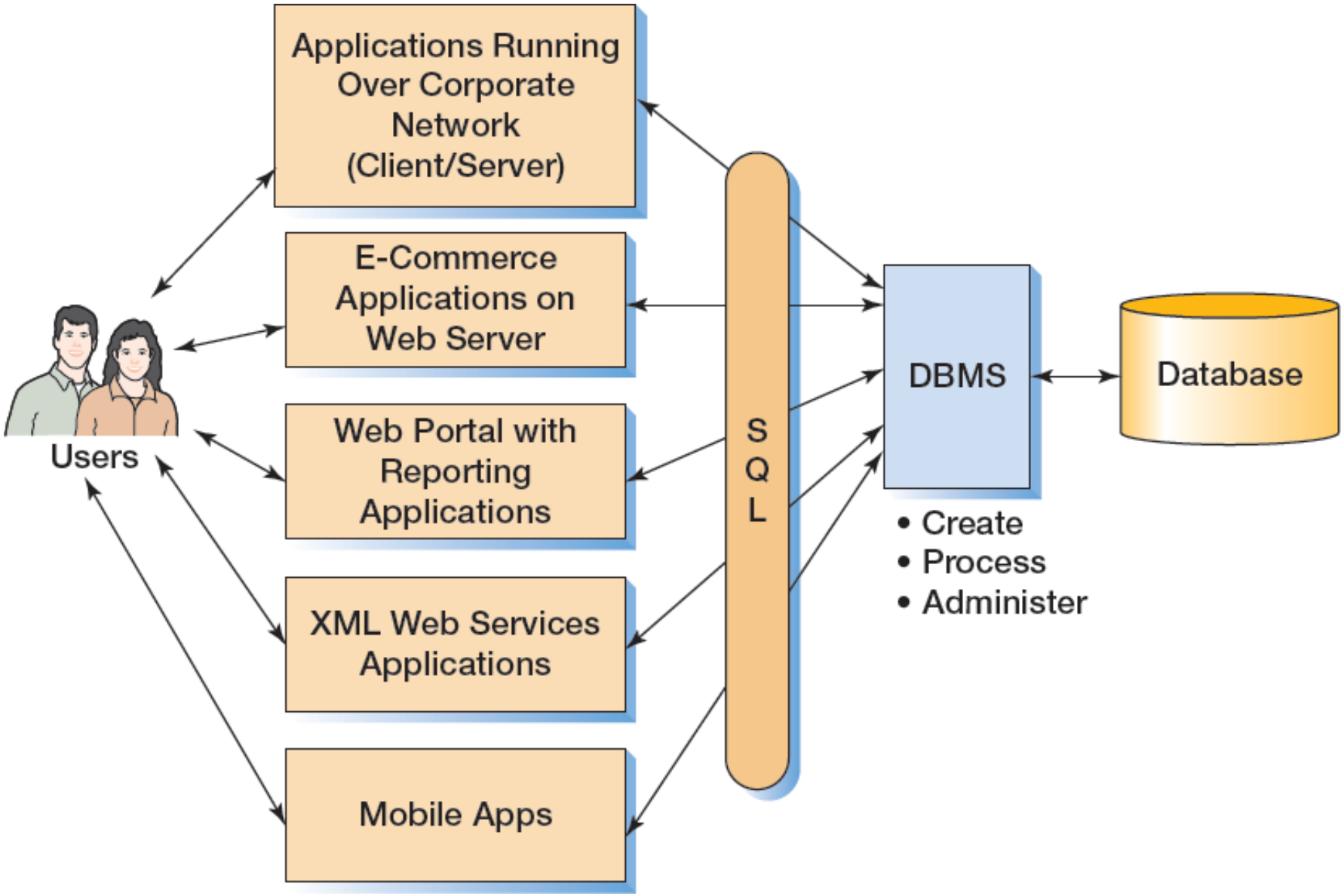
What Is Microsoft Access?

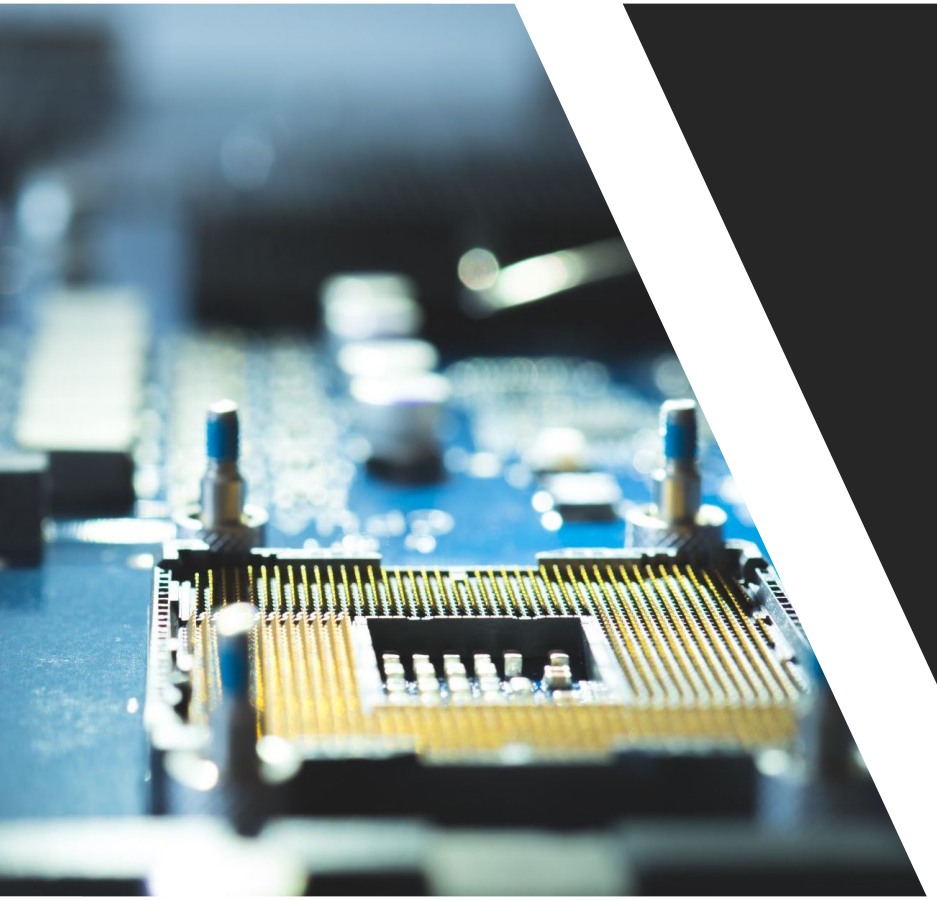
Microsoft Access is a **DBMS plus an application generator**:

- The DBMS creates, processes, and administers Microsoft Access databases.
- The application generator includes query, form, and report components.

The Microsoft Access DBMS engine is called the **Access Data Engine (ADE)**.

Enterprise-Class Database Systems





DBMS PRODUCTS

Refer to the first week's slides

Compatibility: Operating System (OS)

Microsoft Windows

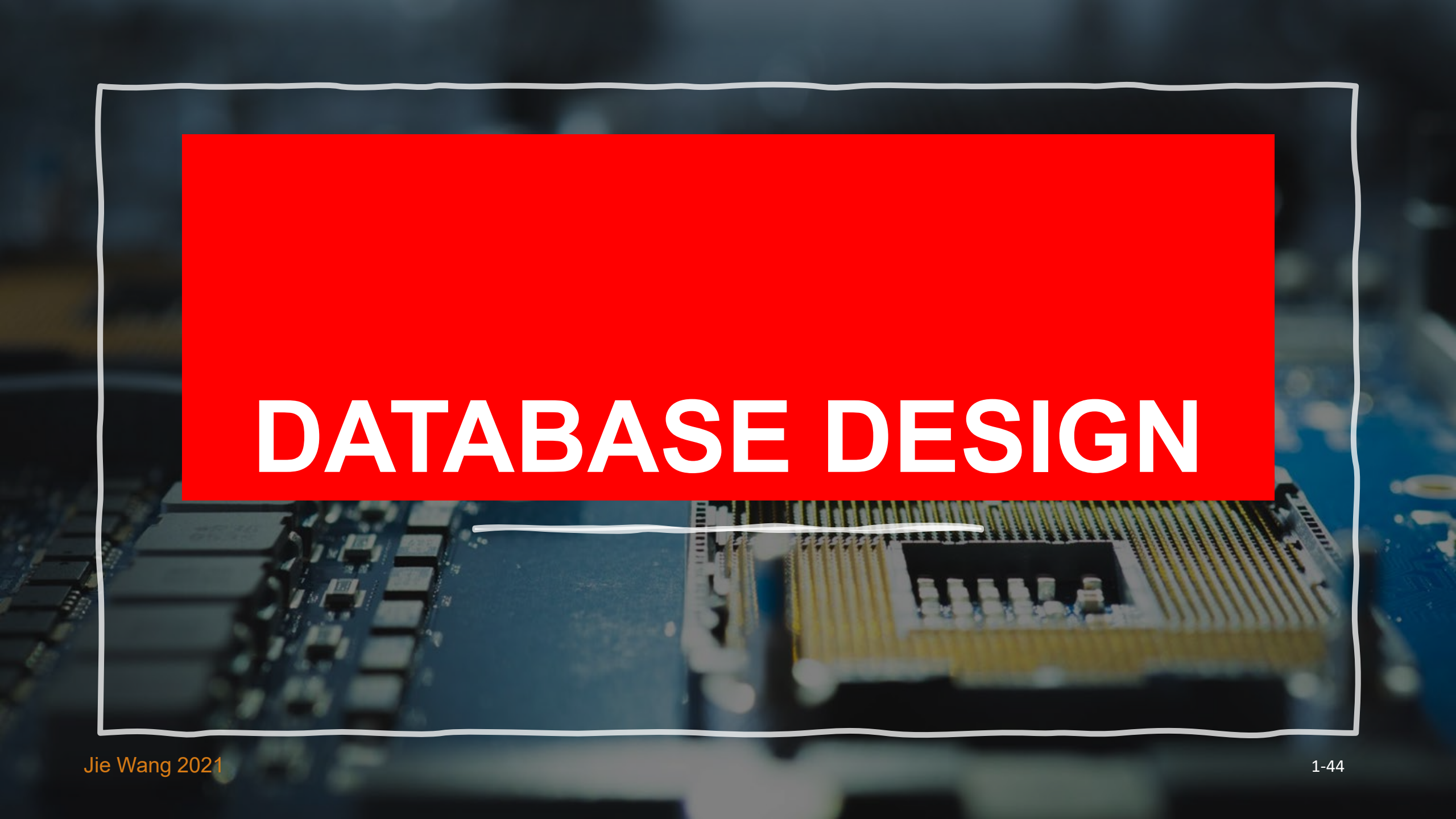
- Microsoft Access
- Microsoft SQL Server
- Oracle Database
- MySQL

Linux

- Oracle Database
- MySQL

Apple OS X

- MySQL

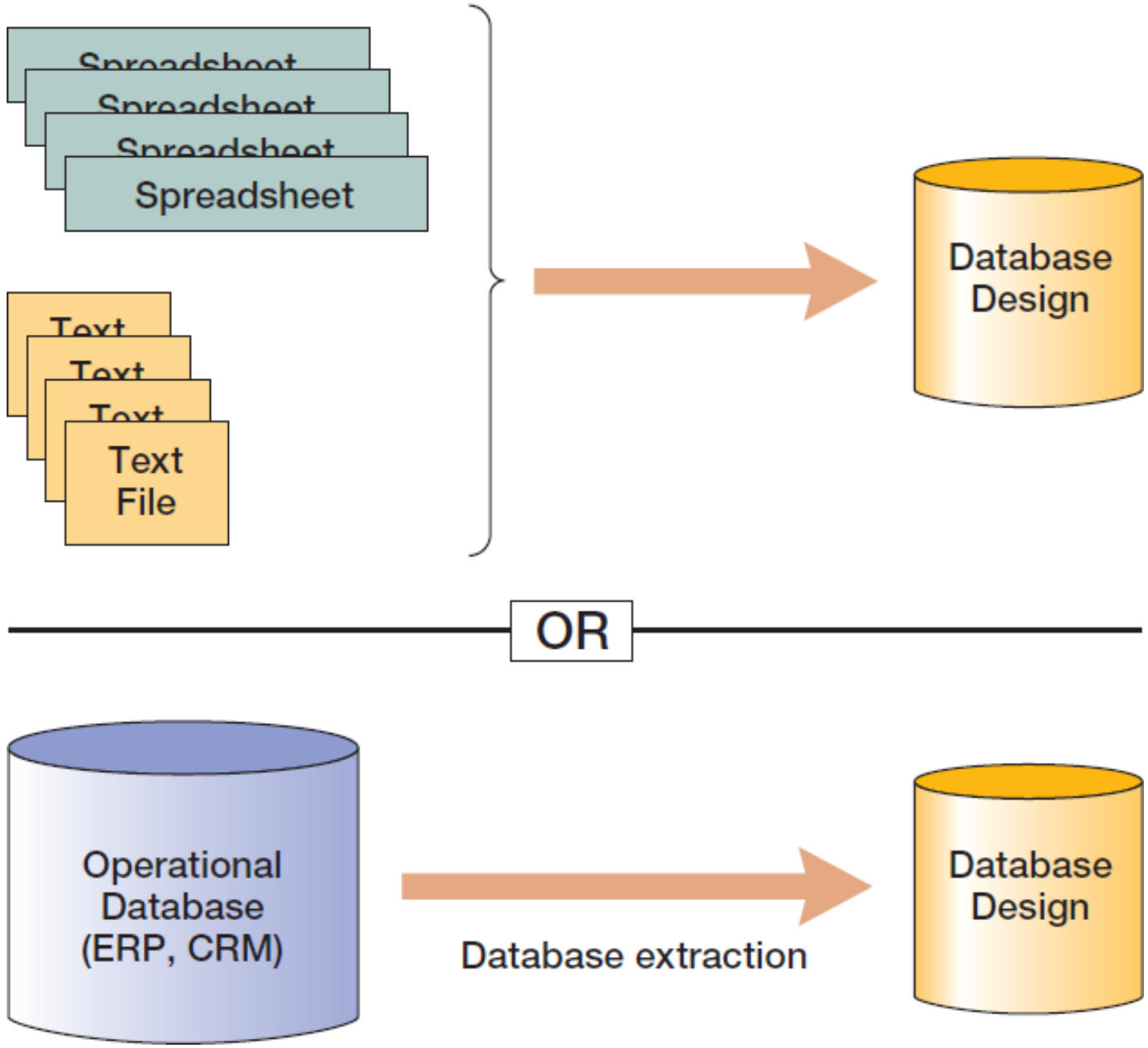

DATABASE DESIGN

Types of Database Design Process
• From existing data (Chapters 3 and 4)
Analyze spreadsheets and other data tables
Extract data from other databases
Design using normalization principles
• New systems development (Chapters 5 and 6)
Create data model from application requirements
Transform data model into database design
• Database redesign (Chapter 8)
Migrate databases to newer databases
Integrate two or more databases
Reverse engineer and design new databases using normalization principles and data model transformation

Note: Chapter 7 discusses database implementation using SQL. You need that knowledge before you can understand database redesign.

Three Types of Database Design

Database Design from Existing Data



EmpNum	EmpName	DeptNum	DeptName
100	Jones	10	Accounting
150	Lau	20	Marketing
200	McCauley	10	Accounting
300	Griffin	10	Accounting

(a) One-Table Design

OR?

DeptNum	DeptName
10	Accounting
20	Marketing

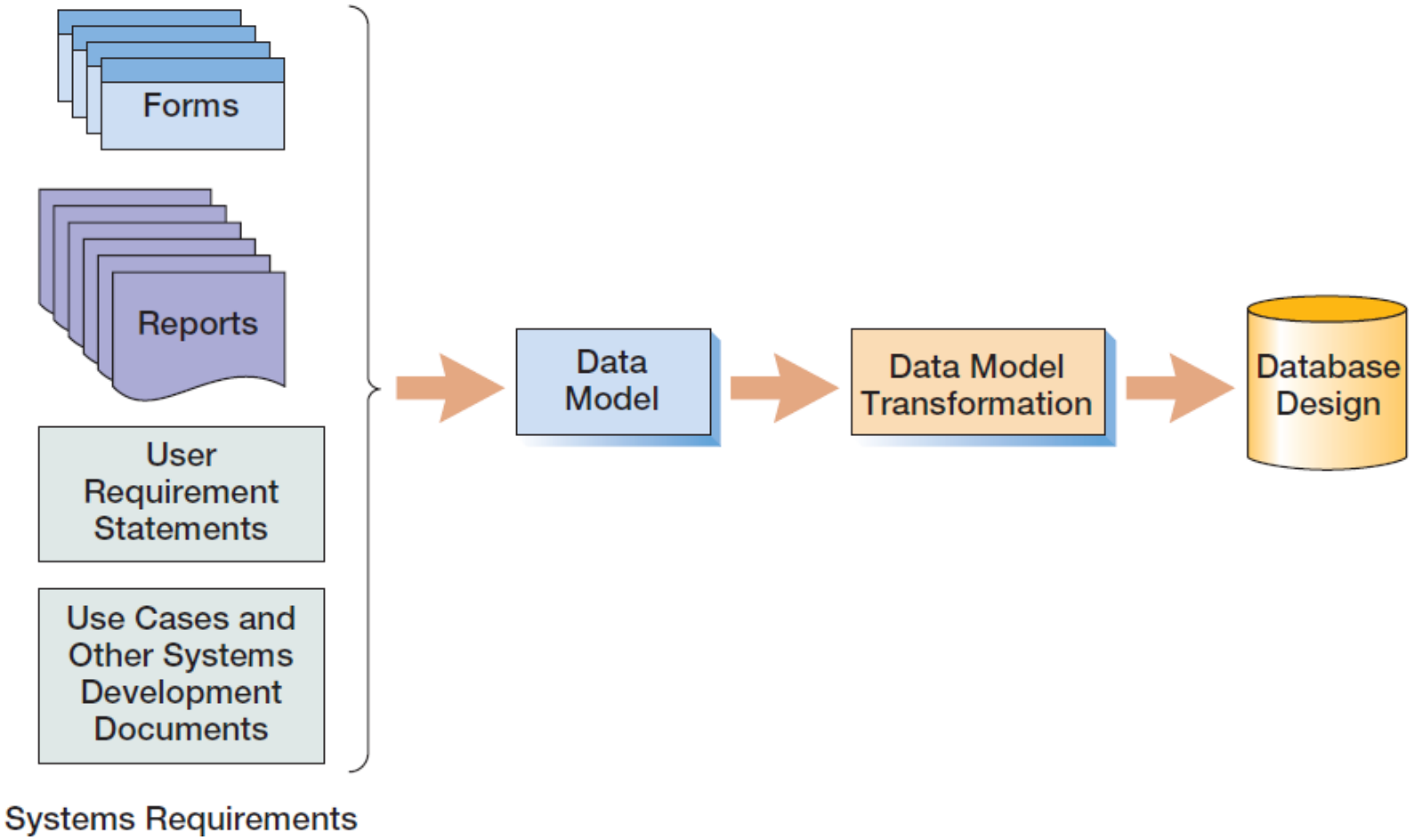
EmpNum	EmpName	DeptNum
100	Jones	10
150	Lau	20
200	McCauley	10
300	Griffin	10

(b) Two-Table Design

Data Import: One or Two Tables?

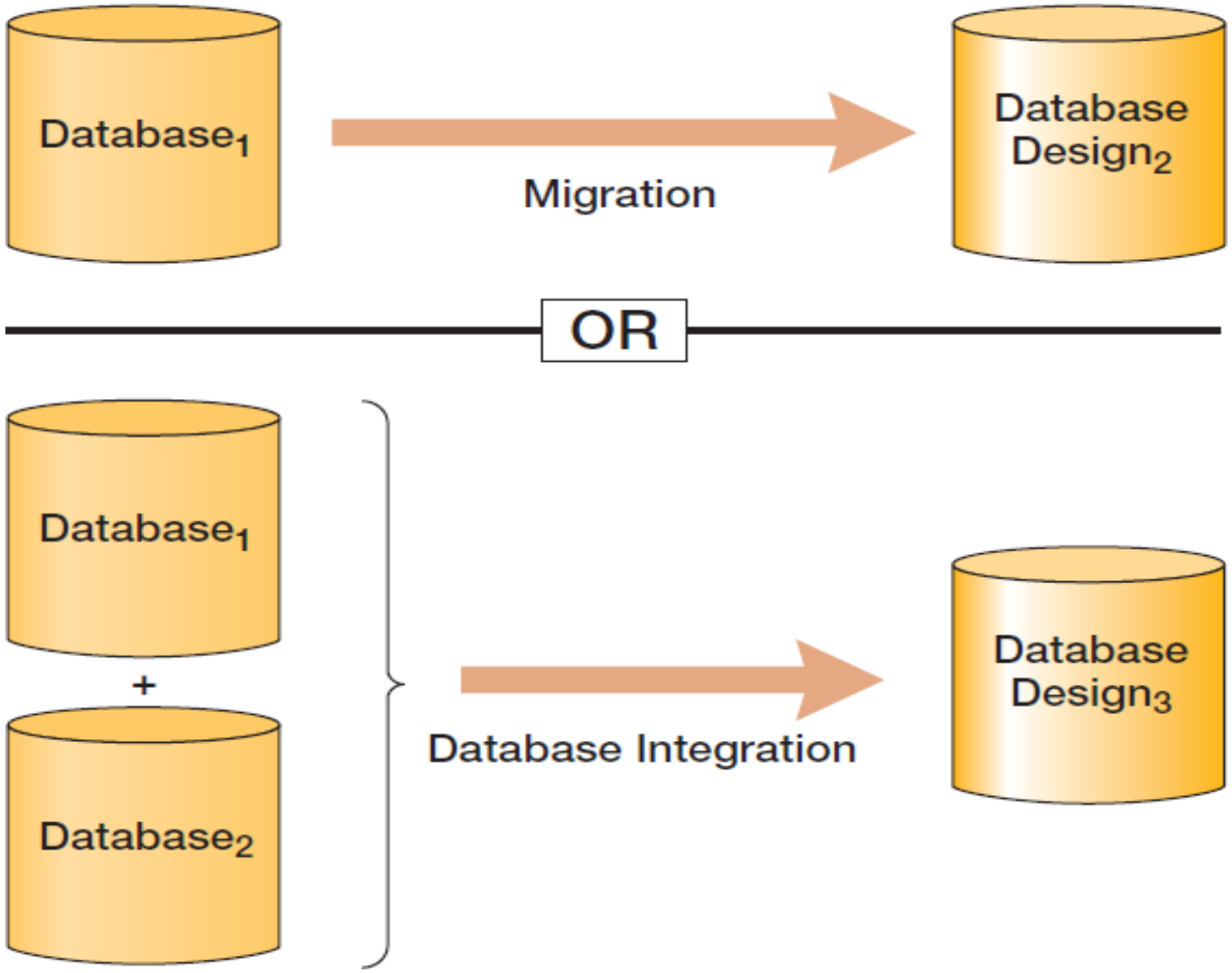
- This is an important decision, and based on a set of rules known as normalization (which is covered in Chapter 3).

Database Design from New Systems Development

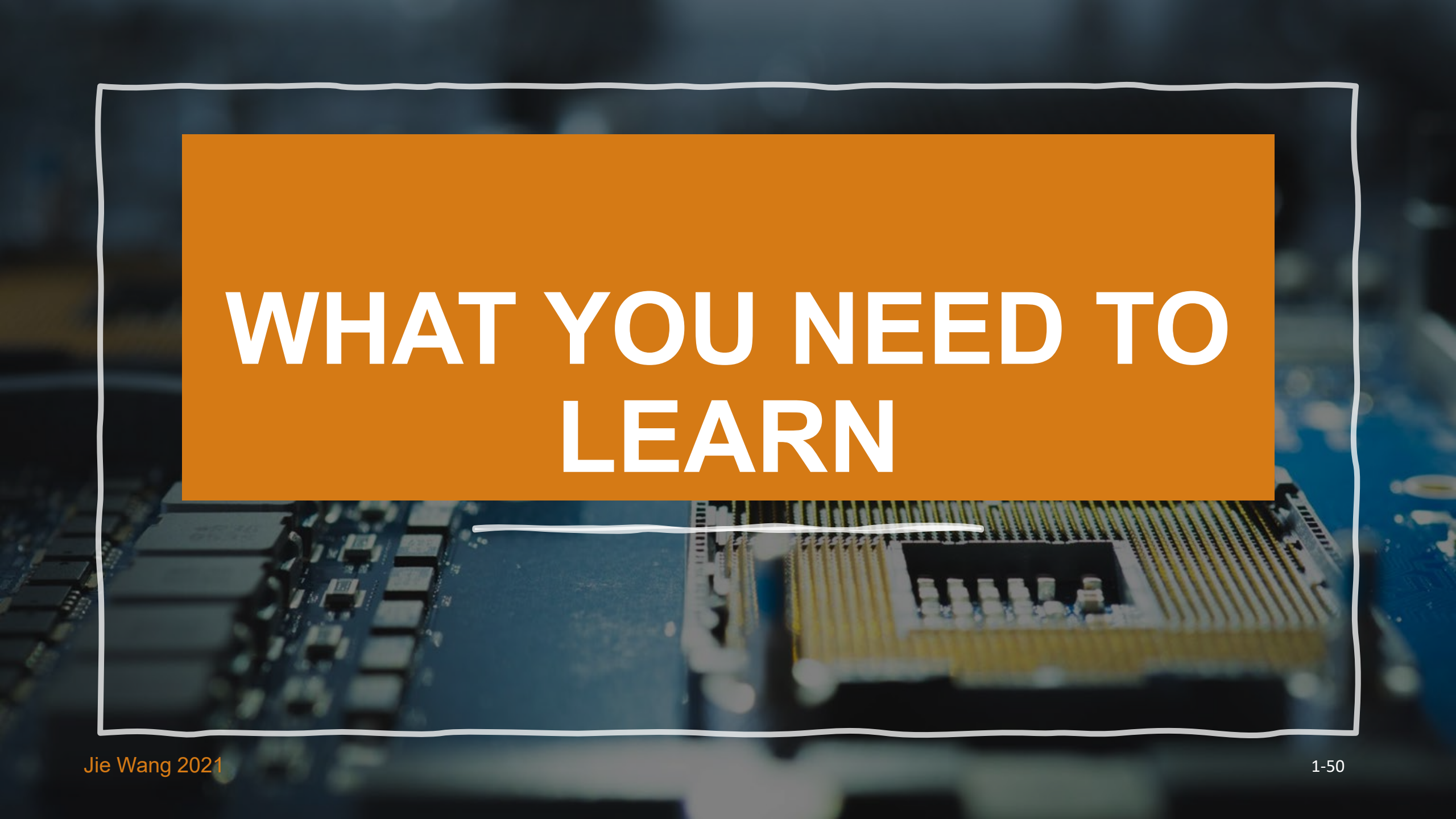


- Entity-Relationship data modeling is covered in Chapter 5, and data model transformations to database designs are covered in Chapter 6.

Database Design from Database Redesign

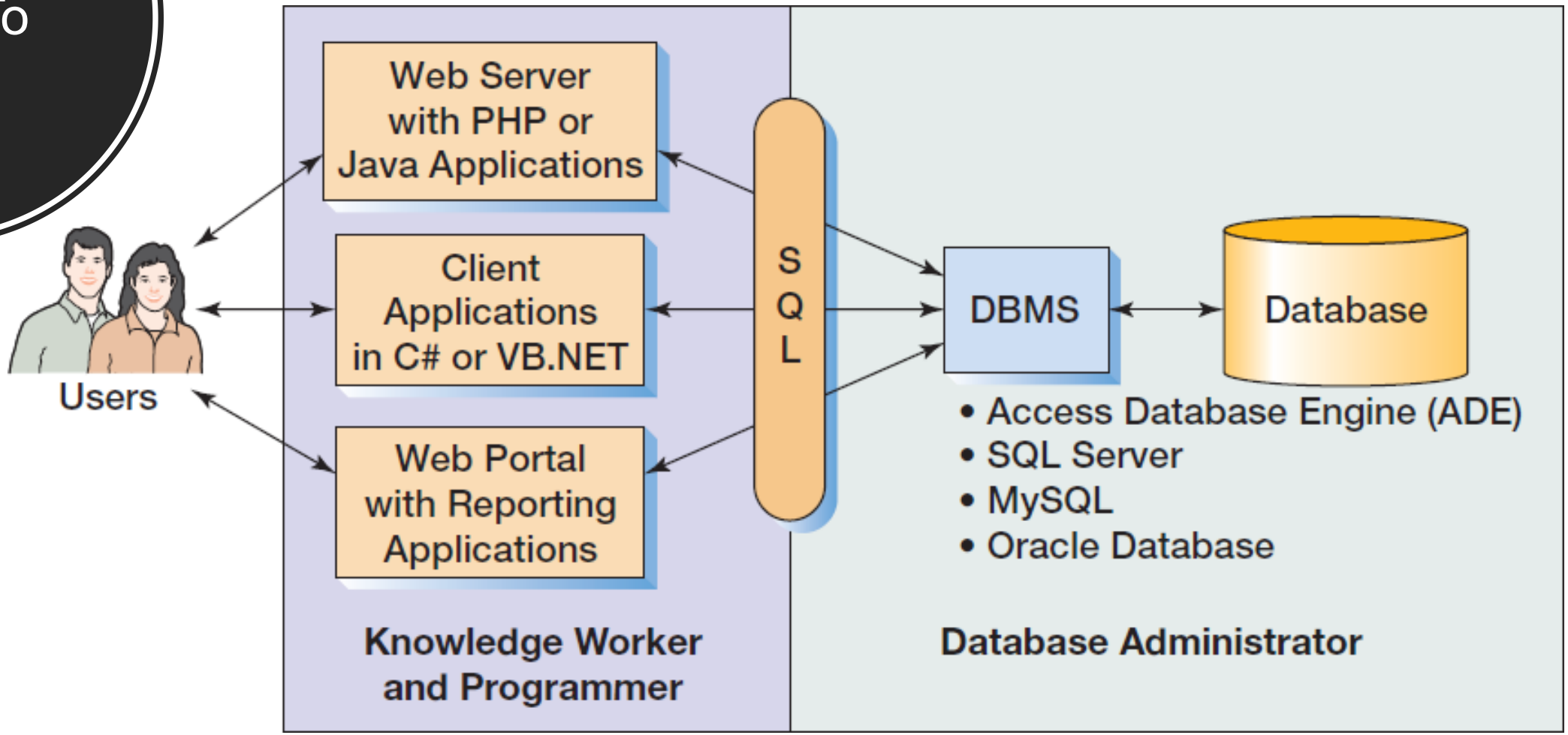


- Database redesign is covered in Chapter 8, after coverage of SQL in Chapter 7.



WHAT YOU NEED TO LEARN

What You Need To Learn



Topic	Chapter	Importance to Database Administrator	Importance to Knowledge Worker and Programmer
Basic SQL	Chapter 2	1	2
The relational database model	Chapter 3	2	2
Design via normalization	Chapter 4	2	1
Data models	Chapter 5	2	1
Data model transformation	Chapter 6	2	1
SQL DDL and constraint enforcement	Chapter 7	3	1
Database redesign	Chapter 8	3	1
Database administration	Chapter 9	3	1
SQL Server, Oracle, MySQL specifics	Chapters 10, 10A, 10B, 10C	3	1
Database application technology	Chapters 11, 12	1	3

1 = Very important; 2 = Important; 3 = Less important

Warning: Opinions vary, ask your instructor for his or hers.

Knowledge Priorities

A Brief History of Database Processing I

Era	Years	Important Products	Remarks
Predatabase	Before 1970	File managers	All data were stored in separate files. Data integration was very difficult. File storage space was expensive and limited.
Early database	1970–1980	ADABAS, System2000, Total, IDMS, IMS	First products to provide related tables. CODASYL DBTG and hierarchical data models (DL/I) were prevalent.
Emergence of relational model	1978–1985	DB2, Oracle	Early relational DBMS products had substantial inertia to overcome. In time, the advantages weighed out.
Microcomputer DBMS products	1982–1992+	dBase-II, R:base, Paradox, Access	Amazing! A database on a micro. All micro DBMS products were eliminated by Microsoft Access in the early 1990s.
Object-oriented DBMS	1985–2000	Oracle ODBMS and others	Never caught on. Required relational database to be converted. Too much work for perceived benefit.

A Brief History of Database Processing II

Rest API over HTTP
Web service
JSON

Era	Years	Important Products	Remarks
Web databases	1995–present	IIS, Apache, PHP, ASP.NET, and Java	Stateless characteristic of HTTP was a problem at first. Early applications were simple one-stage transactions. Later, more complex logic developed.
Open source DBMS products	1995–present	MySQL, PostgresQL, and other products	Open source DBMS products provide much of the functionality and features of commercial DBMS products at reduced cost.
XML and Web services	1998–present	XML, SOAP, WSDL, UDDI, and other standards	XML provides tremendous benefits to Web-based database applications. Very important today. May replace relational databases during your career. See Chapter 11 and Appendix K.
Big Data and the NoSQL movement	2009–present	Hadoop, Cassandra, Hbase, CouchDB, MongoDB, and other products	Web applications such as Facebook and Twitter use Big Data technologies, often using Hadoop and related products. The NoSQL movement is really a NoRelationalDB movement that replaces relational databases with non-relational data structures. See Chapter 12 and Appendix K.