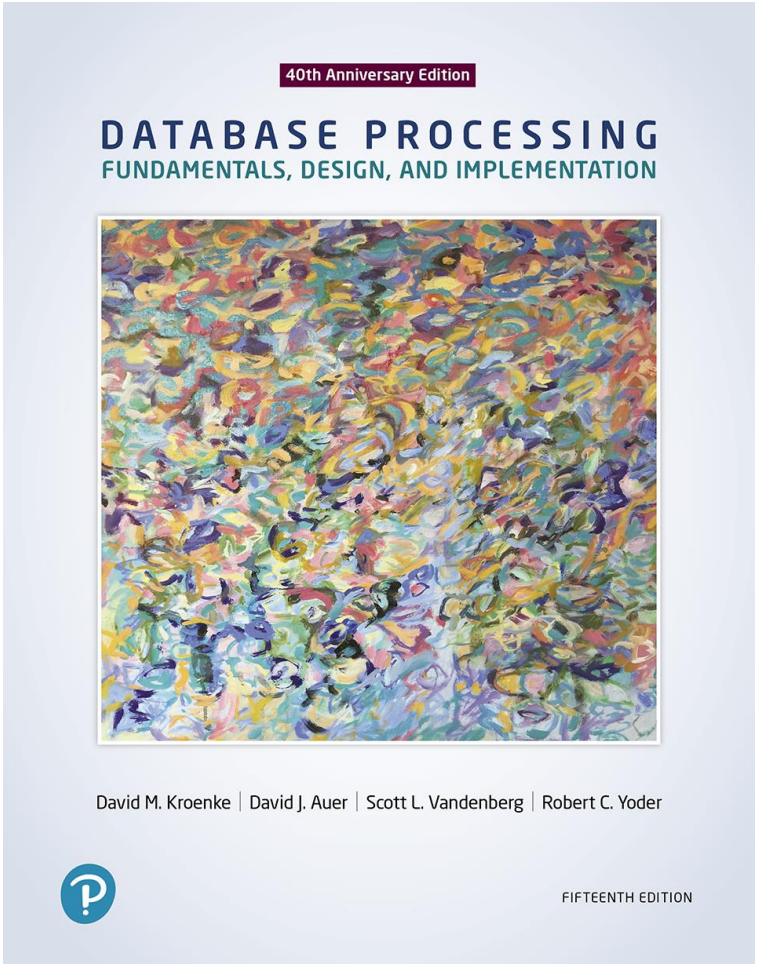
Database Processing: Fundamentals, Design, and Implementation

15th Edition



Chapter One

Introduction

Learning Objectives (1 of 2)

1.1 To understand the importance of databases in Internet Web applications and smart-phone apps

1.2 To understand the nature and characteristics of databases

1.3 To survey some important and interesting database applications

1.4 To gain a general understanding of tables and relationships

1.5 To describe the components of a Microsoft Access database system and explain the functions they perform

1.6 To describe the components of an enterprise-class database system and explain the functions they perform

1.7 To define the term *database management system* (DBMS) and describe the functions of a DBMS

1.8 To define the term *database* and describe what is contained within the database

Learning Objectives (2 of 2)

1.9 To define the term *metadata* and provide examples of metadata

1.10 To define and understand database design from existing data

1.11 To define and understand database design as new systems development

1.12 To define and understand database redesign of an existing database

1.13 To understand the history and development of database processing

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
    - 1983: U.S. 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 1990s: Online retail sites
    - 1995: Amazon
    - Followed by Best Buy
- Early 2000s: Web 2.0

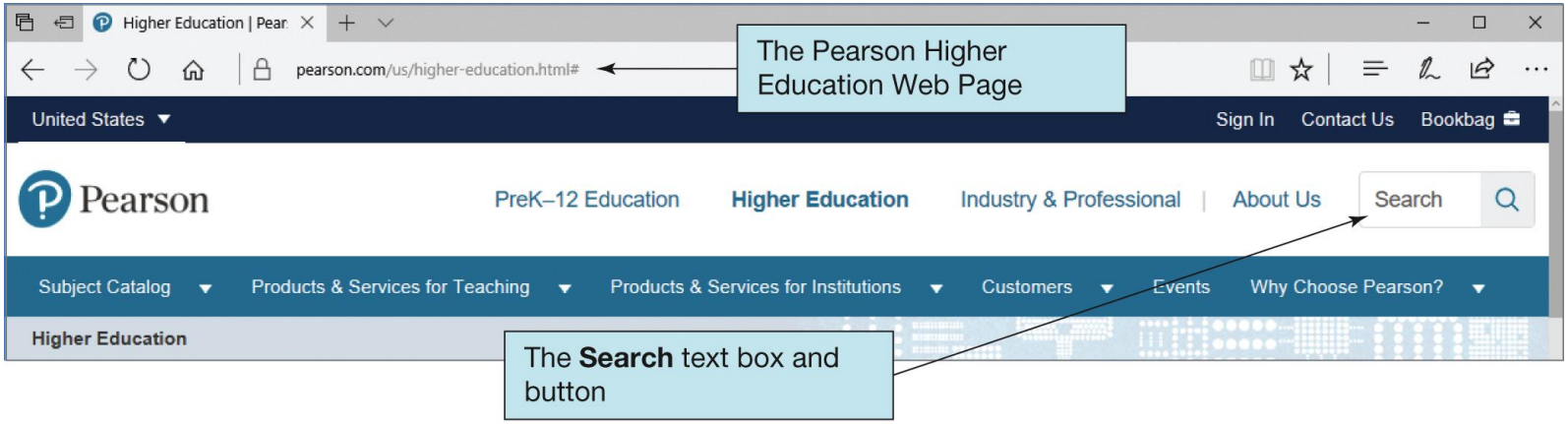
How Did We Get Here?
The Smartphone World

- Mid 1970s: Mobile Phone (Cell Phone)
- Smartphone
  - 2007: Apple iPhone
  - 2008: Google Android Operating System
- Tablets
  - 2010: Apple iPad
- Apps
- All of these examples depend on databases

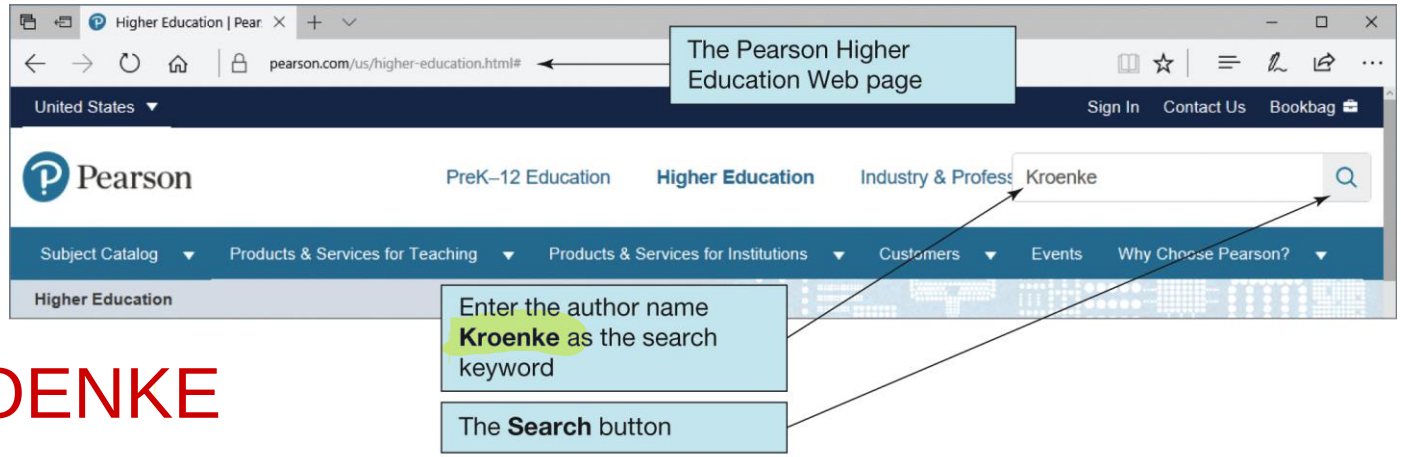
Databases in the Internet and Mobile Device World

- Databases are important because they are everywhere and are used daily:
  - Facebook
    - Posts
    - Likes
  - Twitter
    - Tweets
  - Online shopping
    - Amazon.com – [Do an actual search]

Figure 1-1
Searching a Database in a Web Browser (1 of 2)



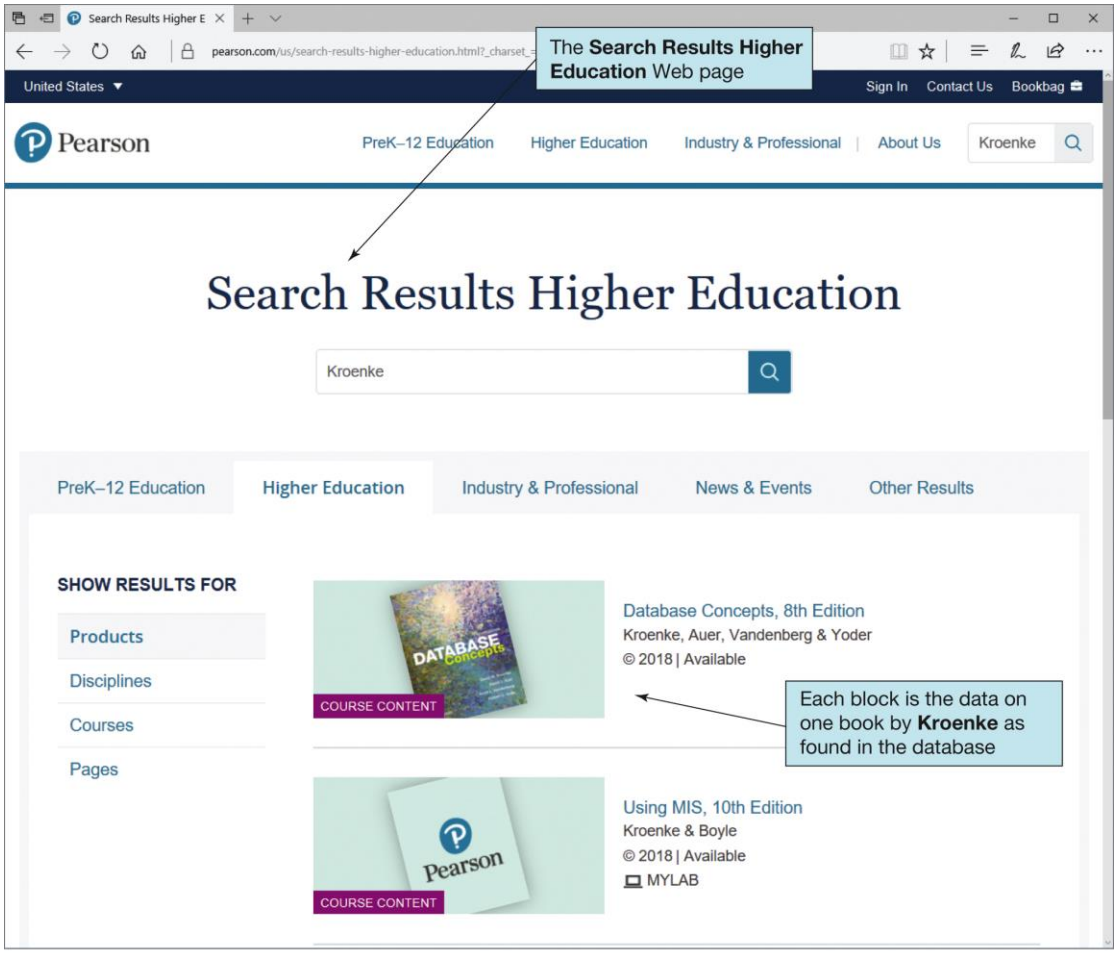
(a) The Pearson Higher Education Web Page



KROENKE

(b) Entering Author Name *Kroenke* as the Search Keyword

Figure 1-1
Searching a Database in a Web Browser (2 of 2)

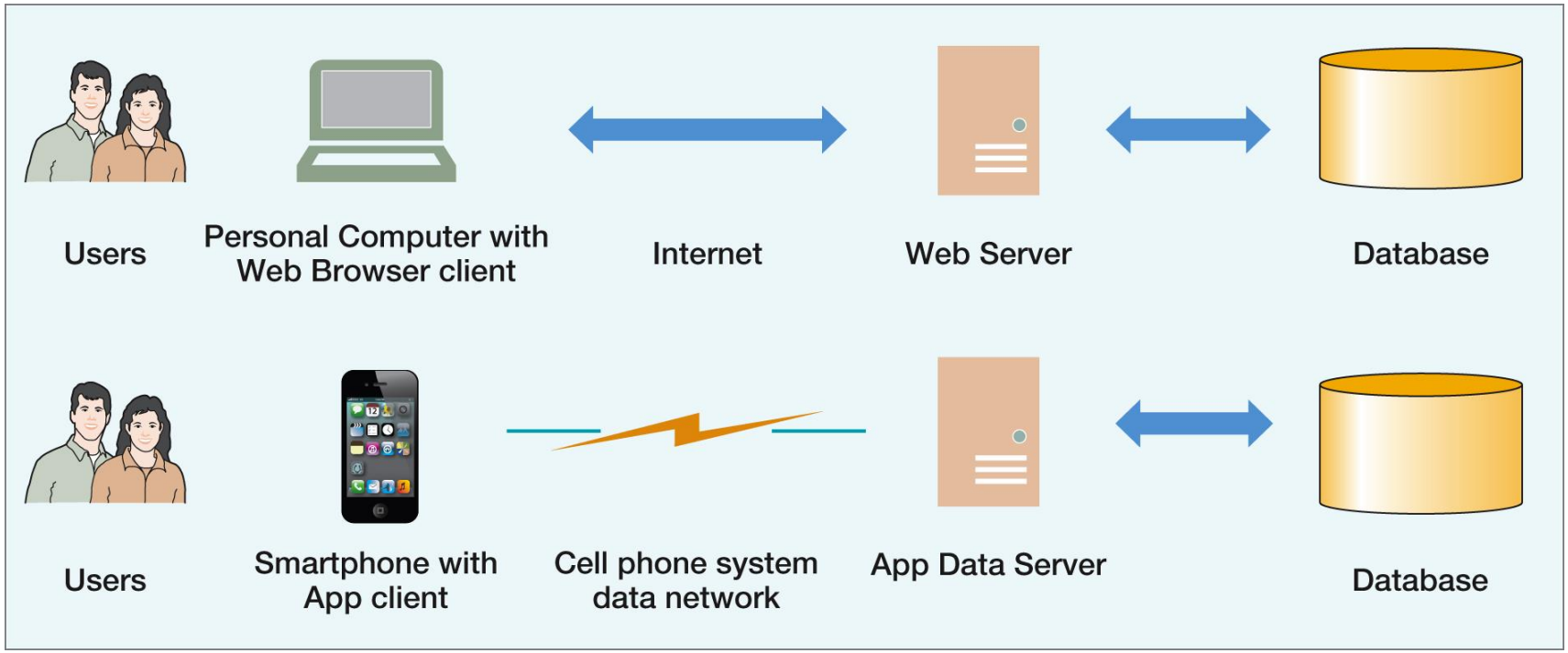


(c) Books by Author *Kroenke* Found in the Database

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**.

Figure 1-2
The Internet and Mobile Device World



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

Figure 1-3
The STUDENT and CLASS Tables

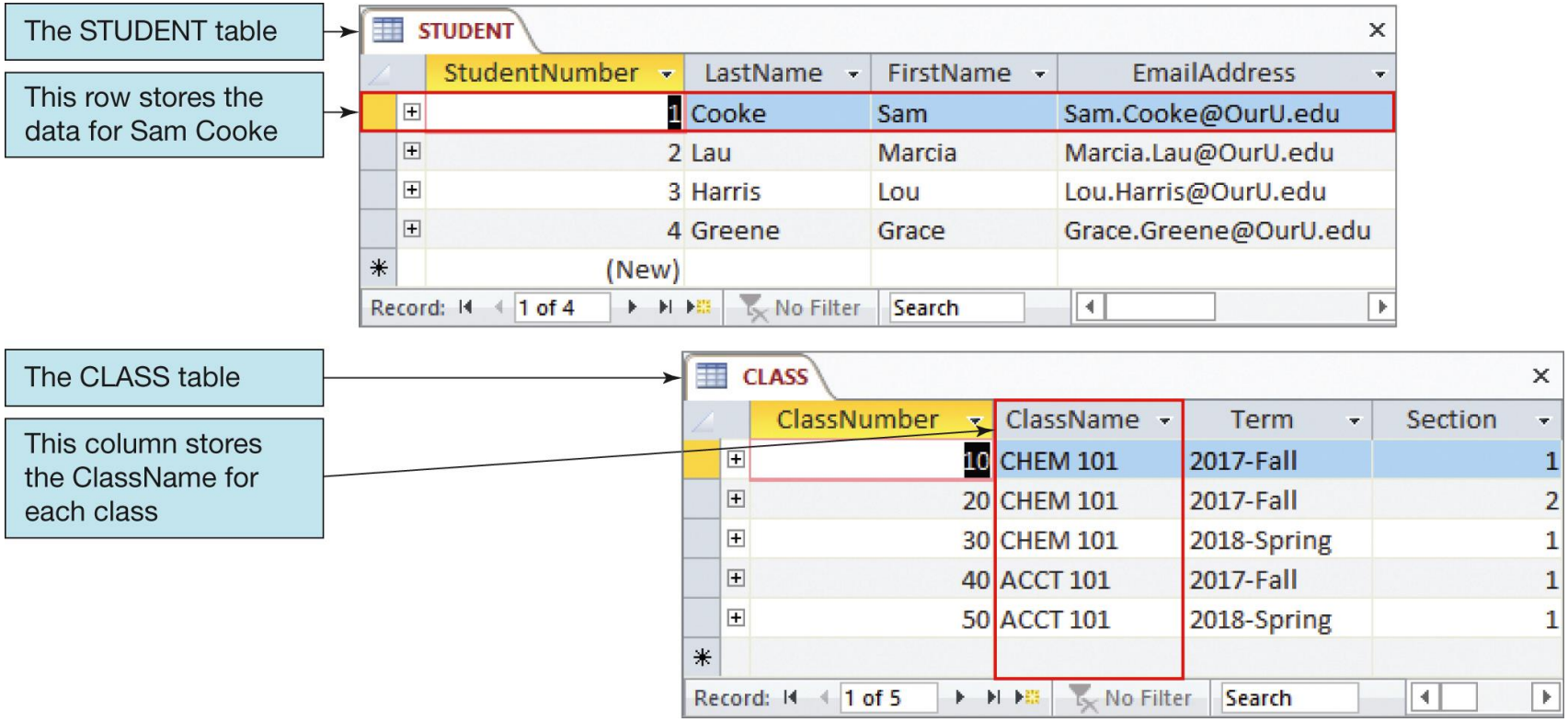


Figure 1-4
The STUDENT, CLASS, and GRADE Tables

The STUDENT 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

The CLASS table

CLASS

ClassNumber	ClassName	Term	Section
10	CHEM 101	2017-Fall	1
20	CHEM 101	2017-Fall	2
30	CHEM 101	2018-Spring	1
40	ACCT 101	2017-Fall	1
50	ACCT 101	2018-Spring	1

Record: 1 of 5 No Filter Search

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

GRADE

Grade
3.7
3.5
3.7
3.1
3.0
3.5
0.0

Record: 1 of 6

Figure 1-5
The Key Database Characteristics: Related Tables

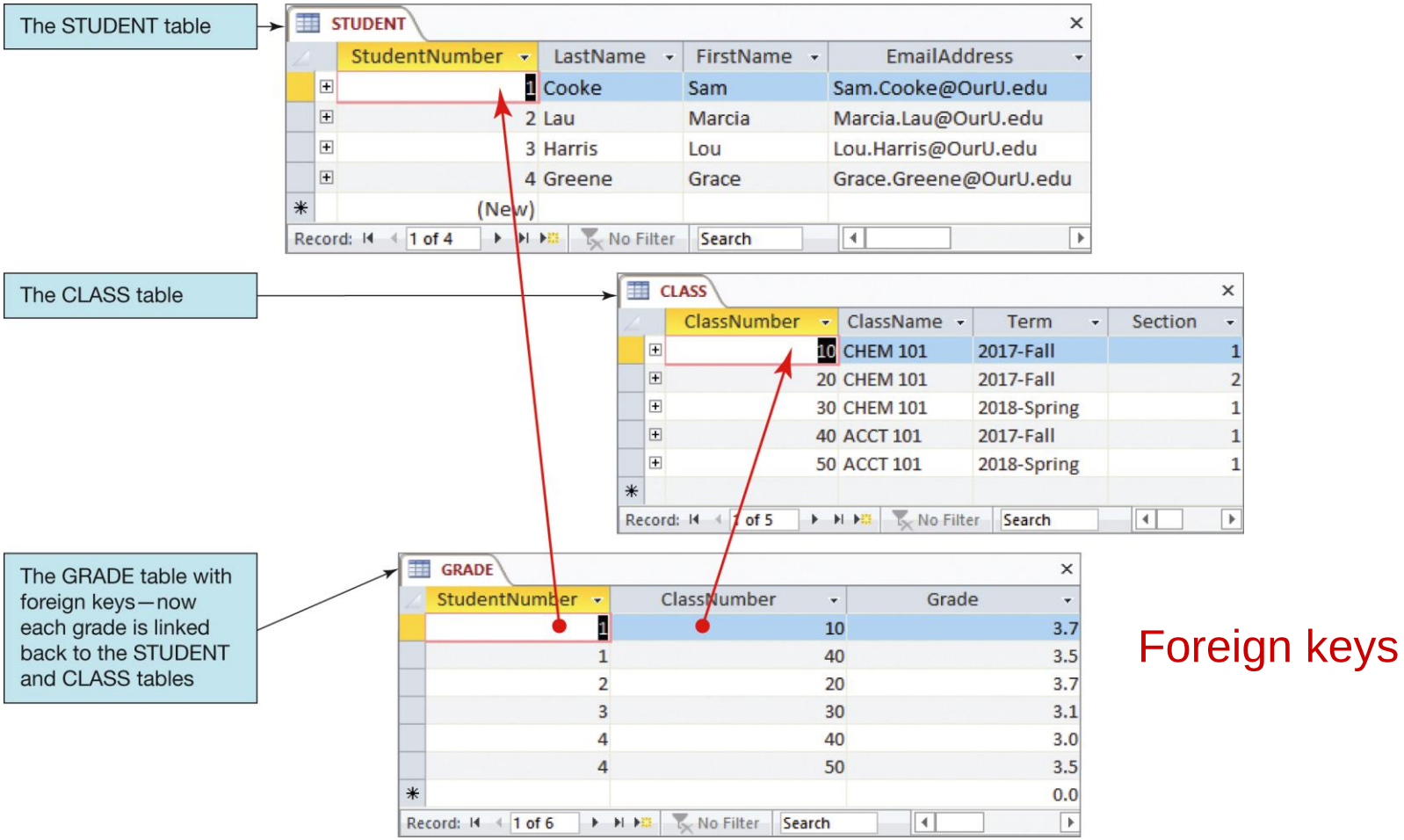
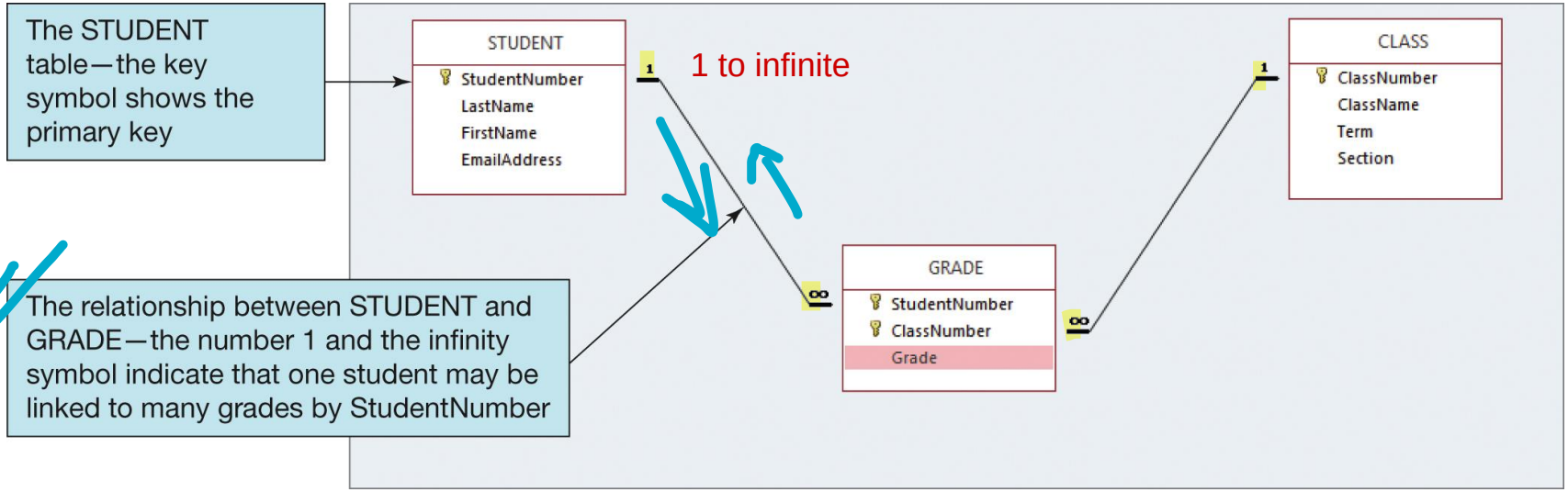


Figure 1-6
Microsoft Access 2016 View of Tables and Relationships



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.

Figure 1-7
Example Database Applications

Application	Example Users	Number of Users	Typical Size	Remarks
Sales Contact Manager	Salesperson	1	2,000 rows	Products such as GioldMine 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	500	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 mining tools.

Figure 1-8
The Components of a Database System

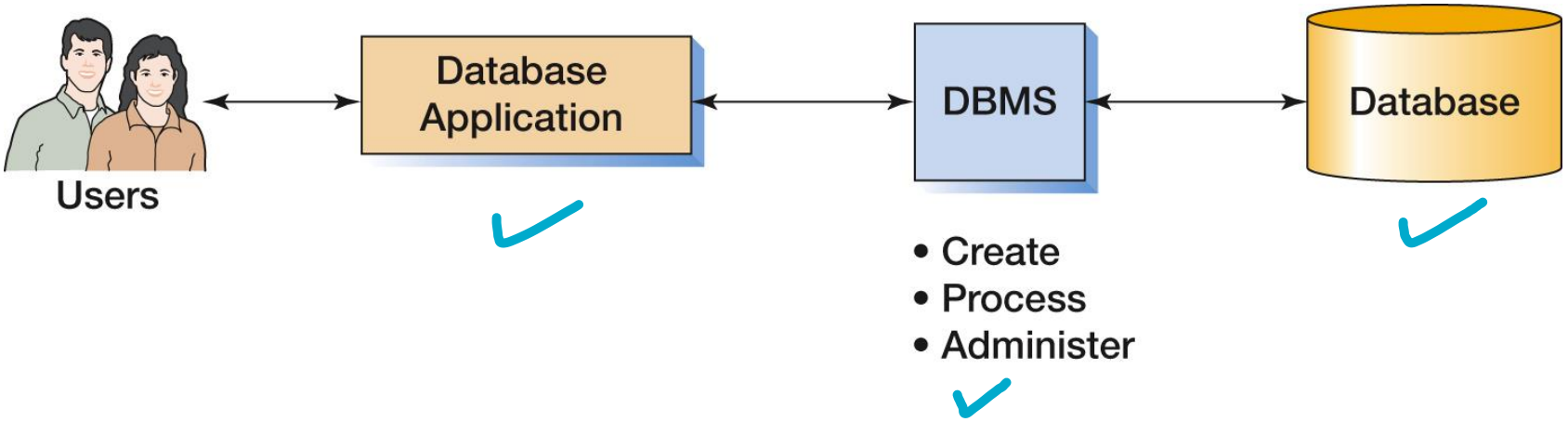
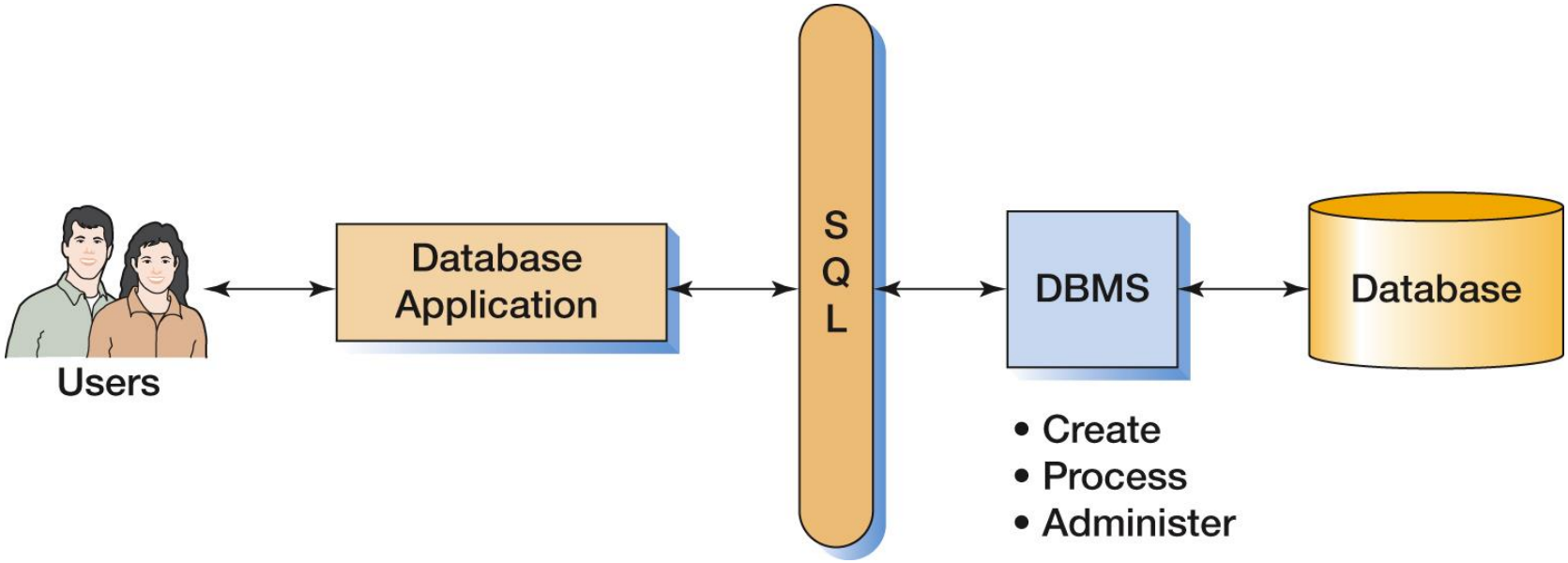


Figure 1-9
The Components of a Database System with SQL



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.

Figure 1-10
Basic Functions of Application Programs

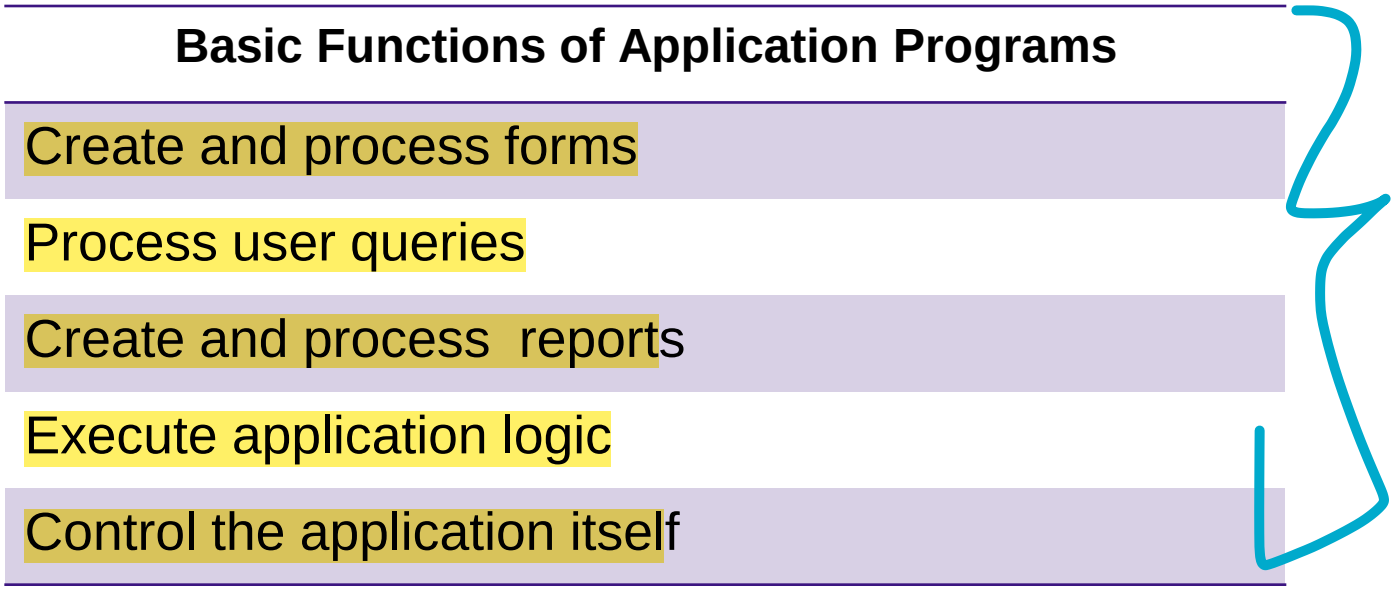


Figure 1-11
An Example Data Entry Form

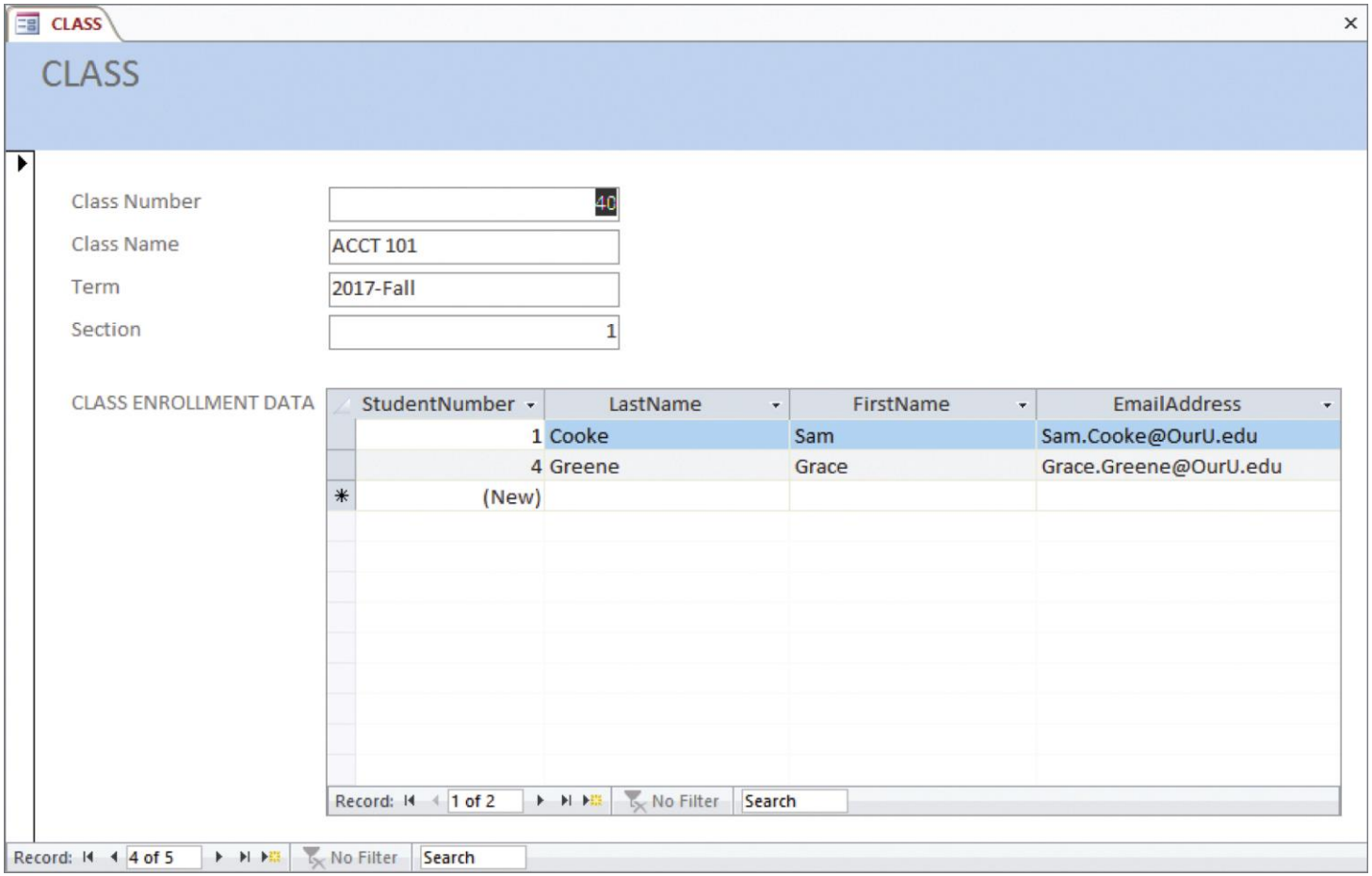


Figure 1-12
Example SQL Query Results

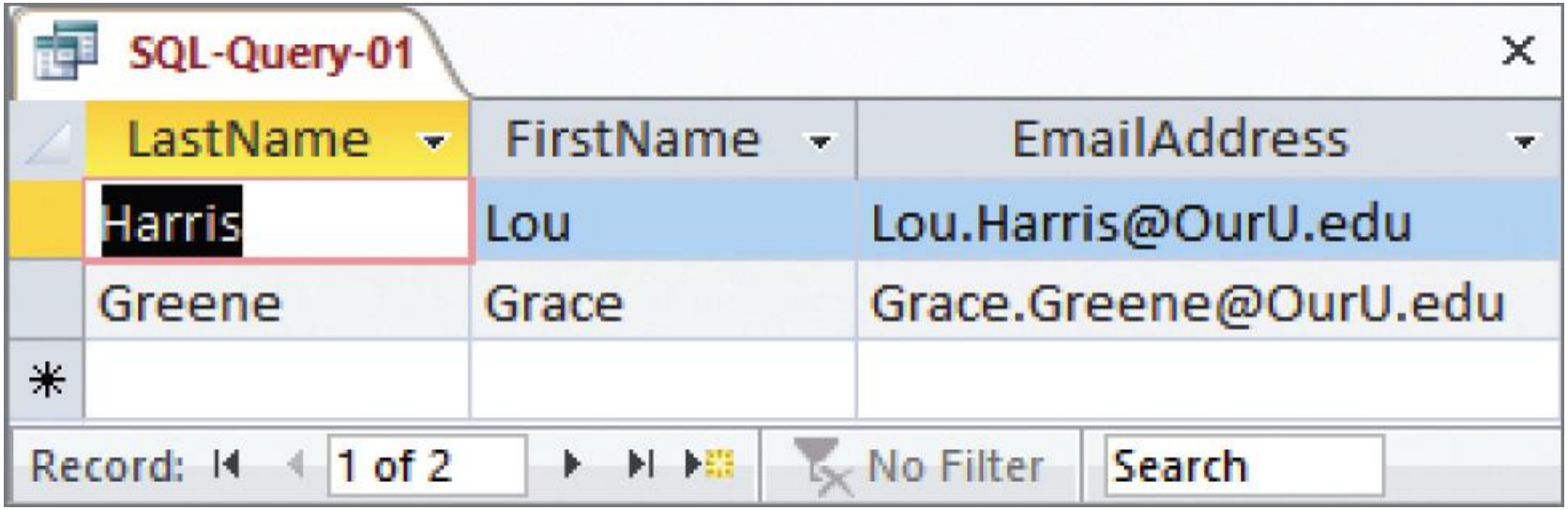
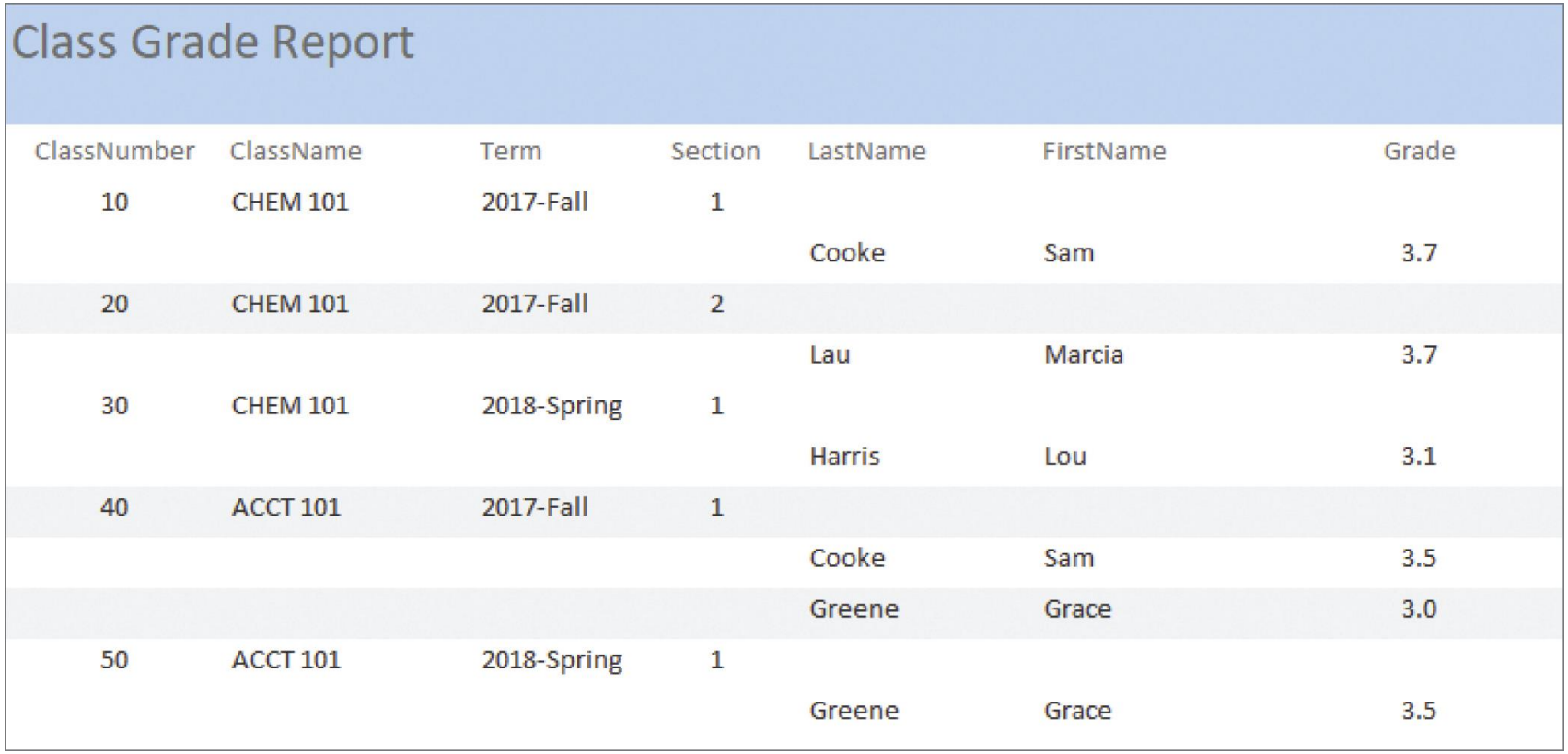


Figure 1-13
Example Report



Class Grade Report

ClassNumber	ClassName	Term	Section	LastName	FirstName	Grade
10	CHEM 101	2017-Fall	1			
				Cooke	Sam	3.7
20	CHEM 101	2017-Fall	2			
				Lau	Marcia	3.7
30	CHEM 101	2018-Spring	1			
				Harris	Lou	3.1
40	ACCT 101	2017-Fall	1			
				Cooke	Sam	3.5
				Greene	Grace	3.0
50	ACCT 101	2018-Spring	1			
				Greene	Grace	3.5

Figure 1-14
Functions of a 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

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 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.

Figure 1-15
Typical Metadata Tables (1 of 2)

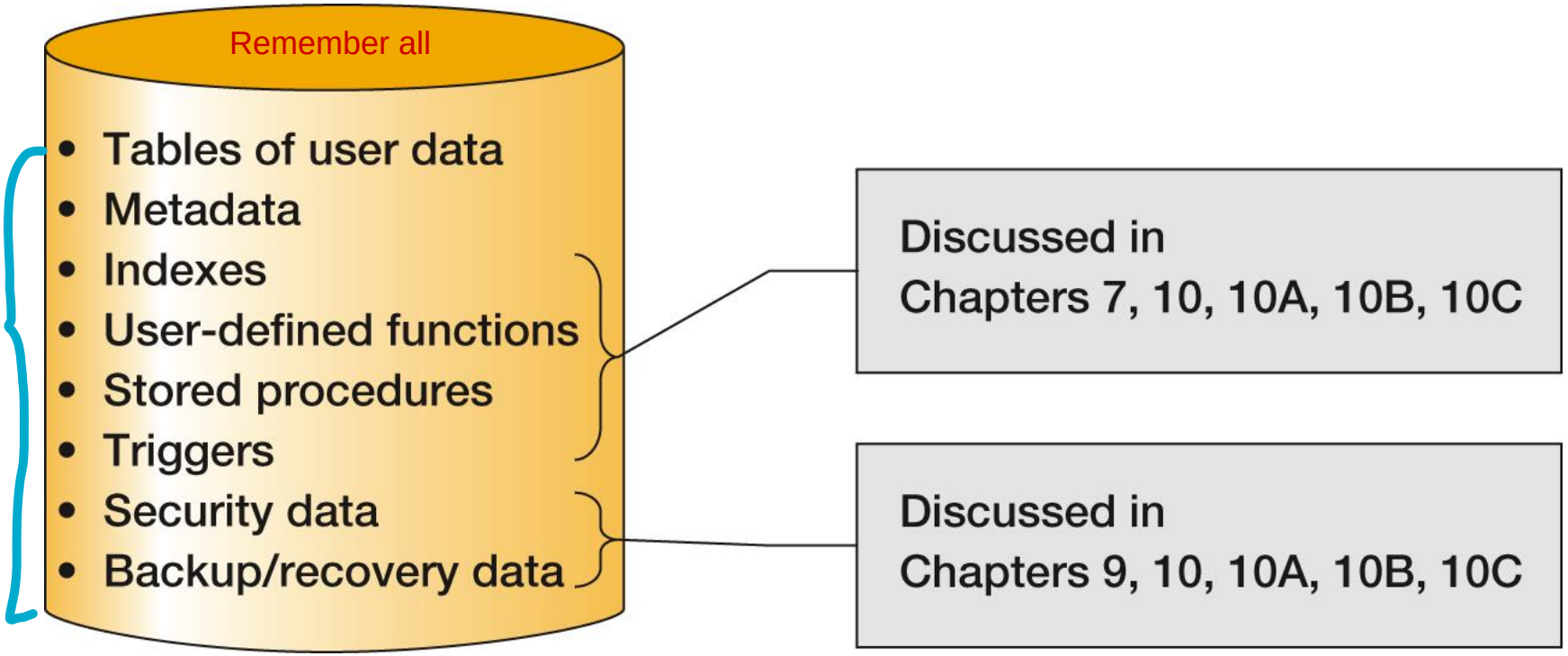
TableName	NumberColumns	PrimaryKey
STUDENT	4	StudentNumber
CLASS	4	ClassNumber
GRADE	3	(StudentNumber, ClassNumber)

Figure 1-15
Typical Metadata Tables (2 of 2)

info abt data itself

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)

Figure 1-16
Typical Metadata Tables



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.
- This is a good strategy for beginners, but not for database professionals.
- NOTE: Microsoft Access 2016 is discussed in detail in Appendix A.

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).
- Microsoft Access 2000 thru 2010 can be used as an application generator for the Microsoft SQL Server DBMS.

Figure 1-17
Components of a Microsoft Access Database System

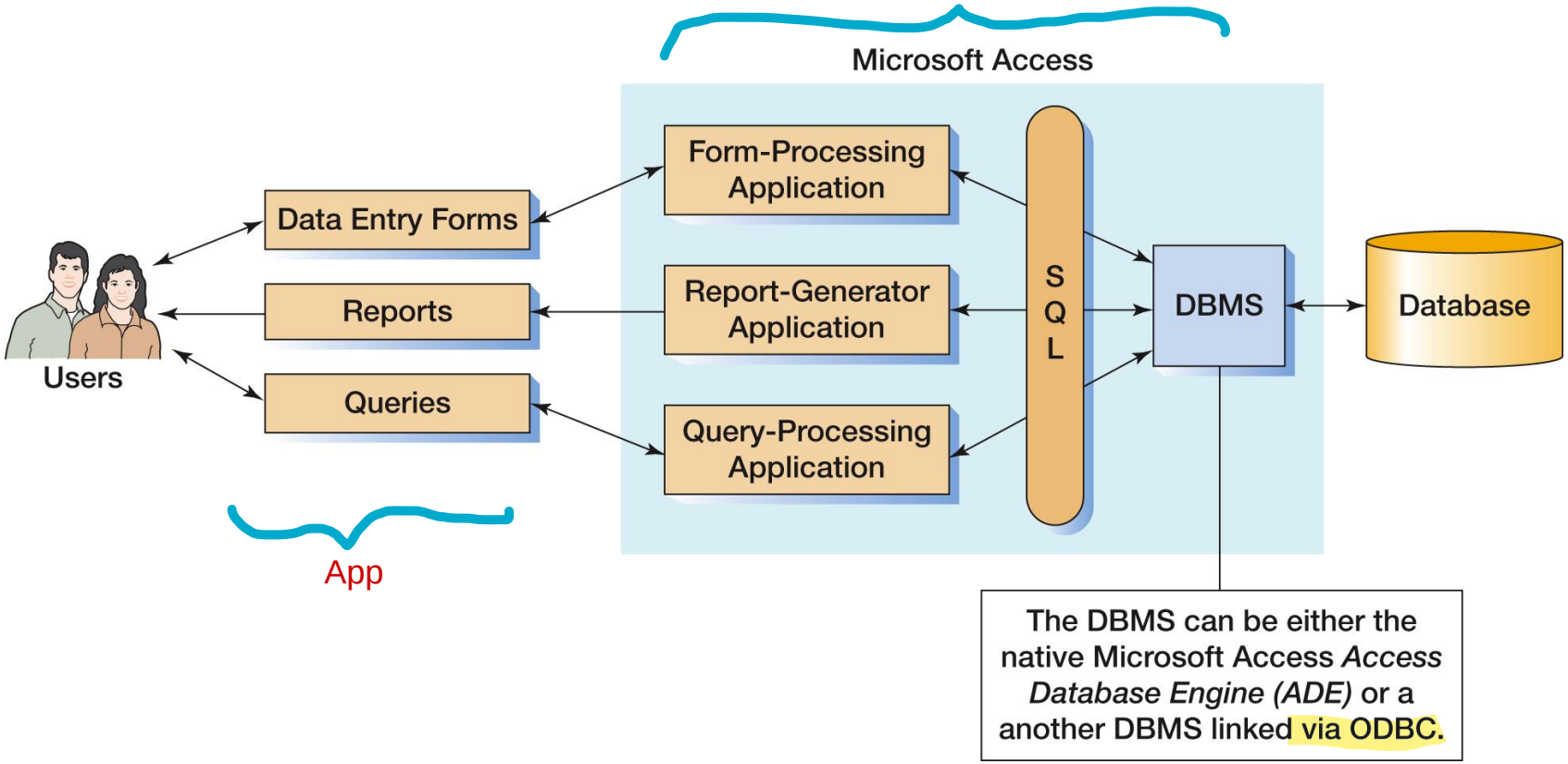


Figure 1-18
Components of an Enterprise-Class Database System

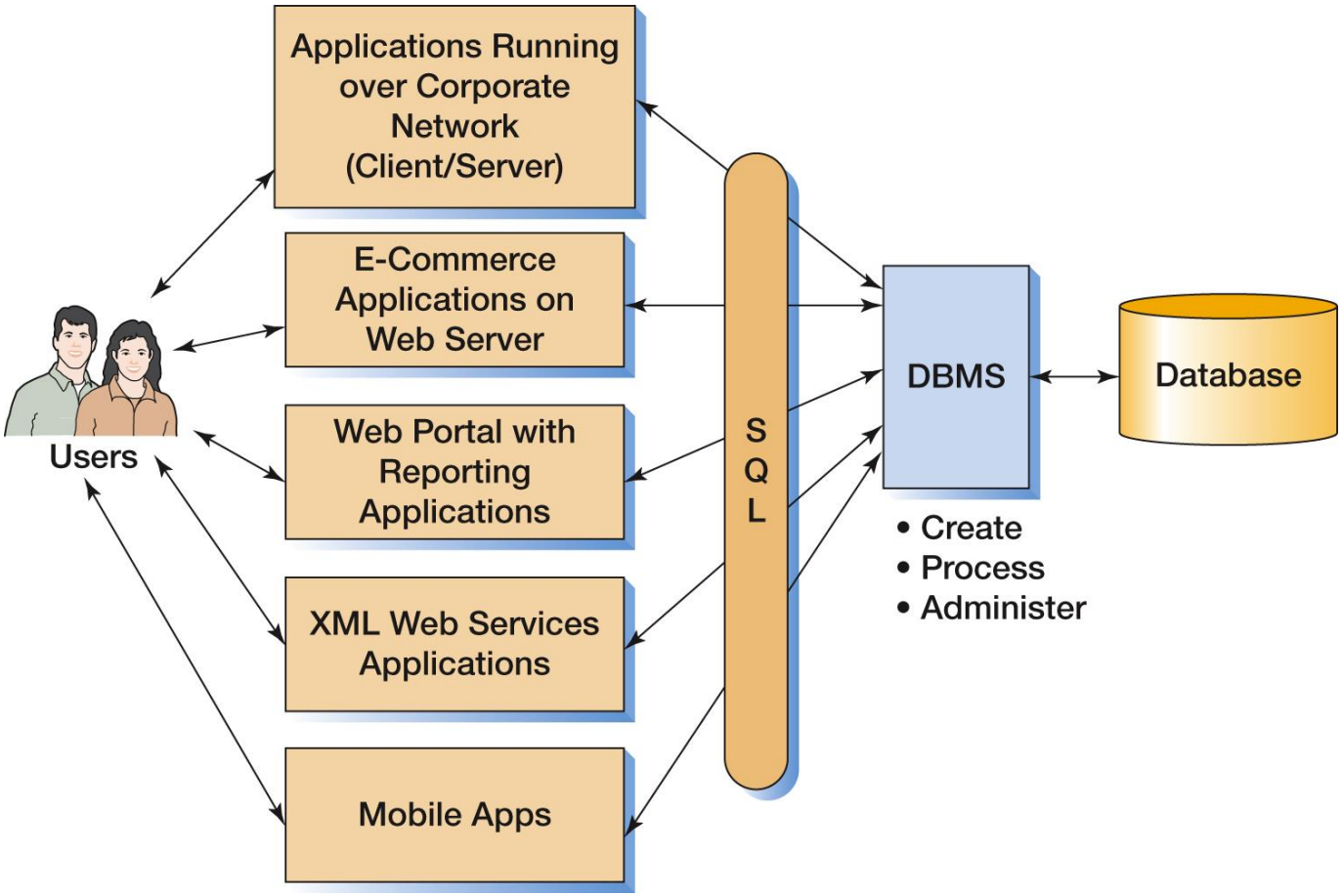


Figure 1-19
Common Professional View of DBMS Products

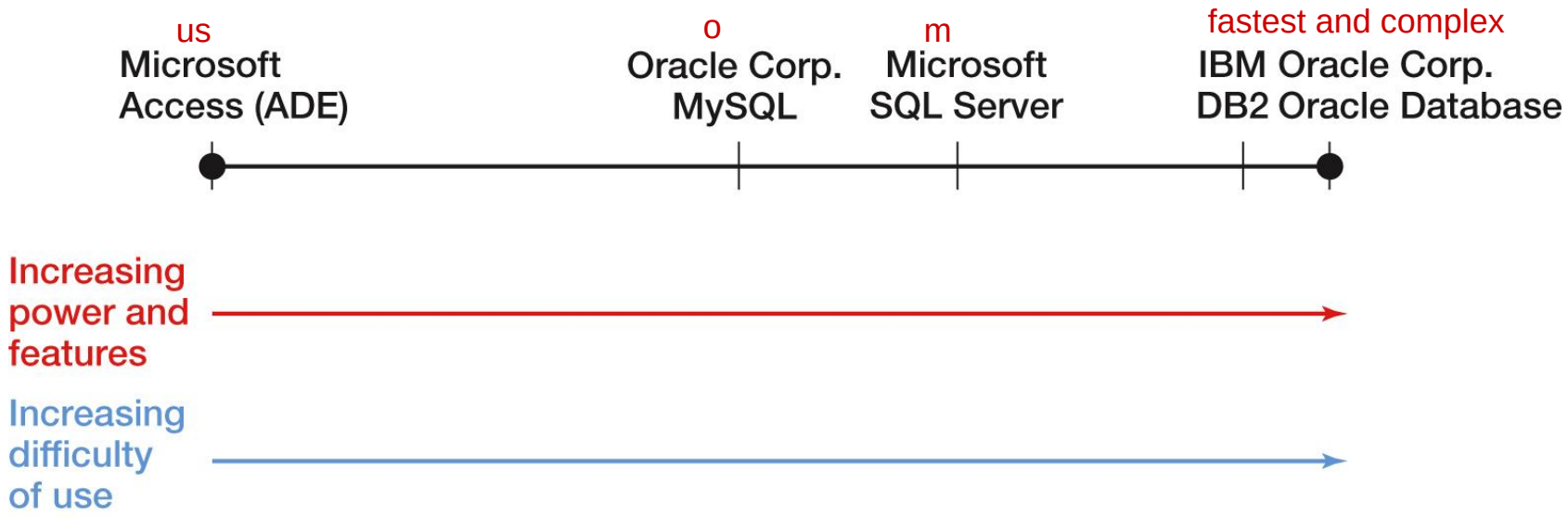


Figure 1-20
Three Types of Database Design

existing

new

existing + new (modify In new)

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)
Crete 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 database using normalization principles and data model transformation

Figure 1-21
Databases Originating from Existing Data

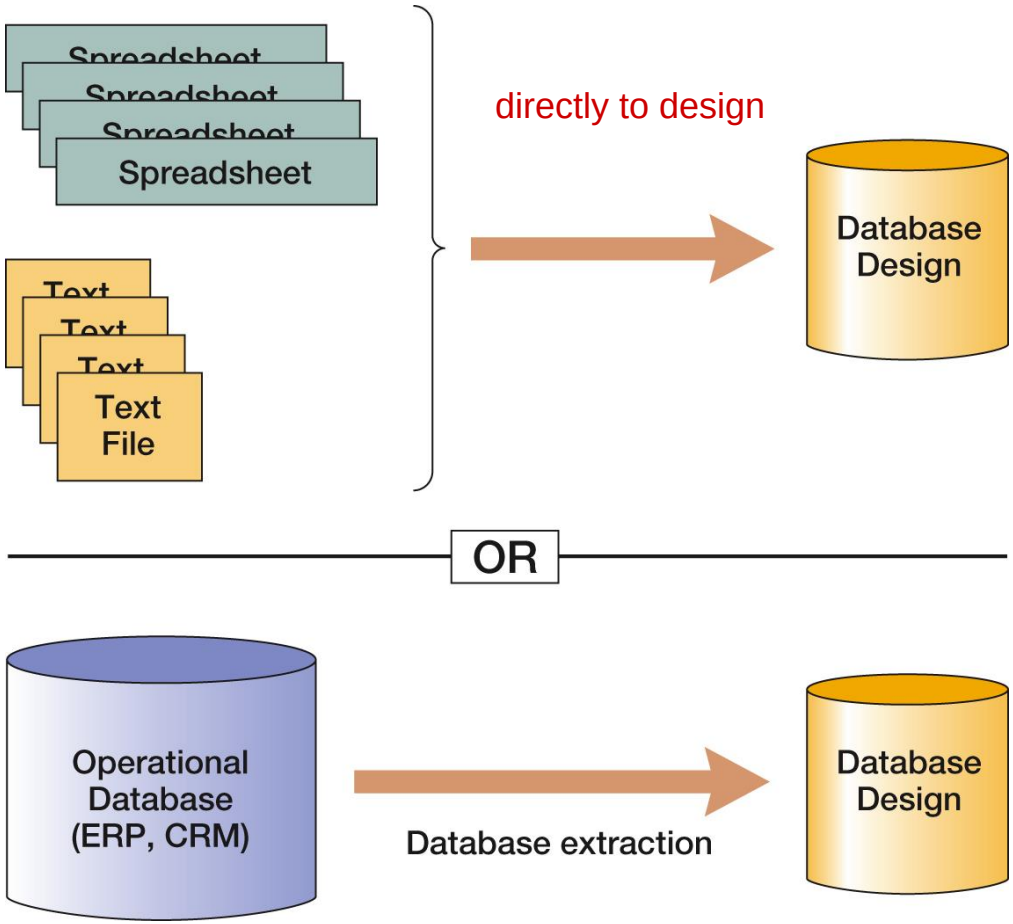


Figure 1-22
Data Import: One or Two Tables?

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

(a) One-Table Design

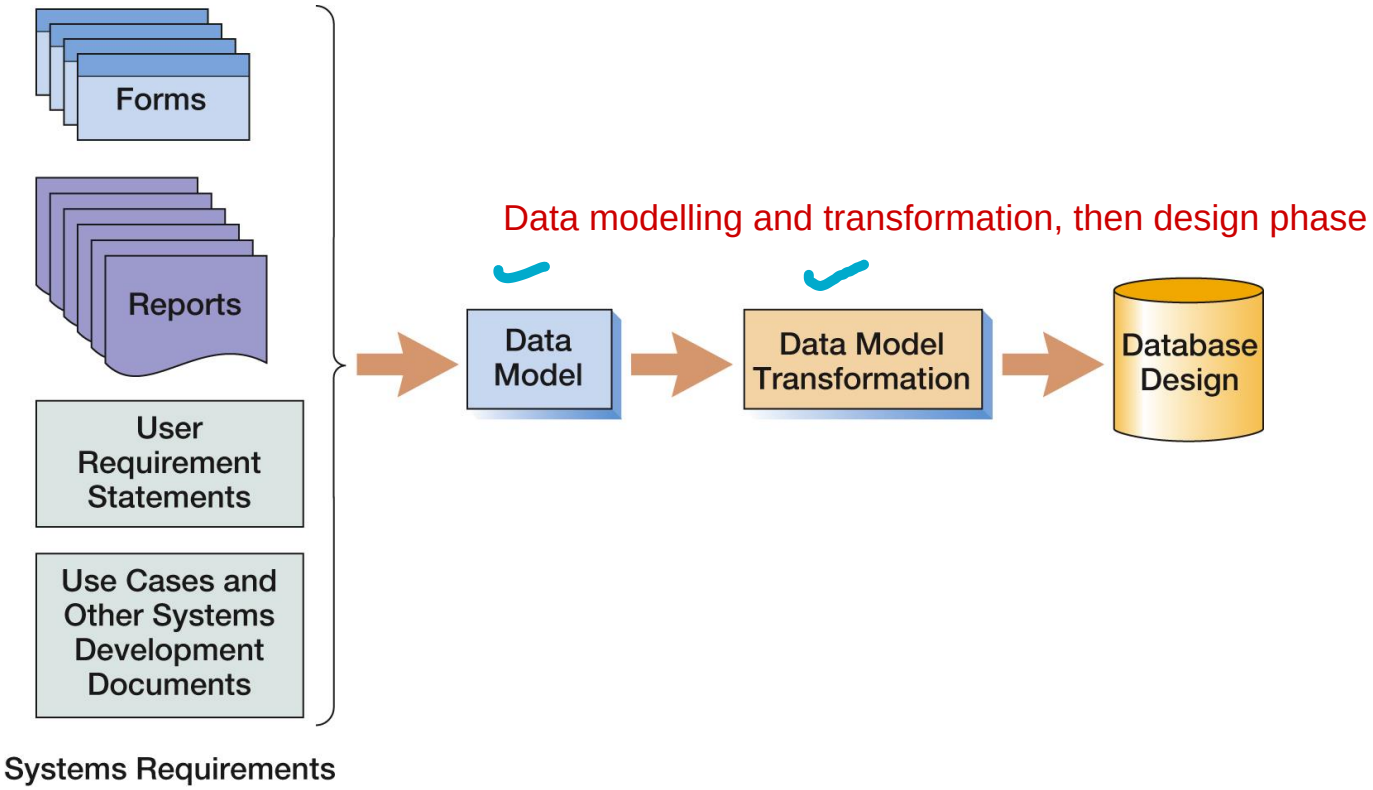
OR?

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

Figure 1-23
Database Originating 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.

Figure 1-24
Databases Originating from Database Redesign

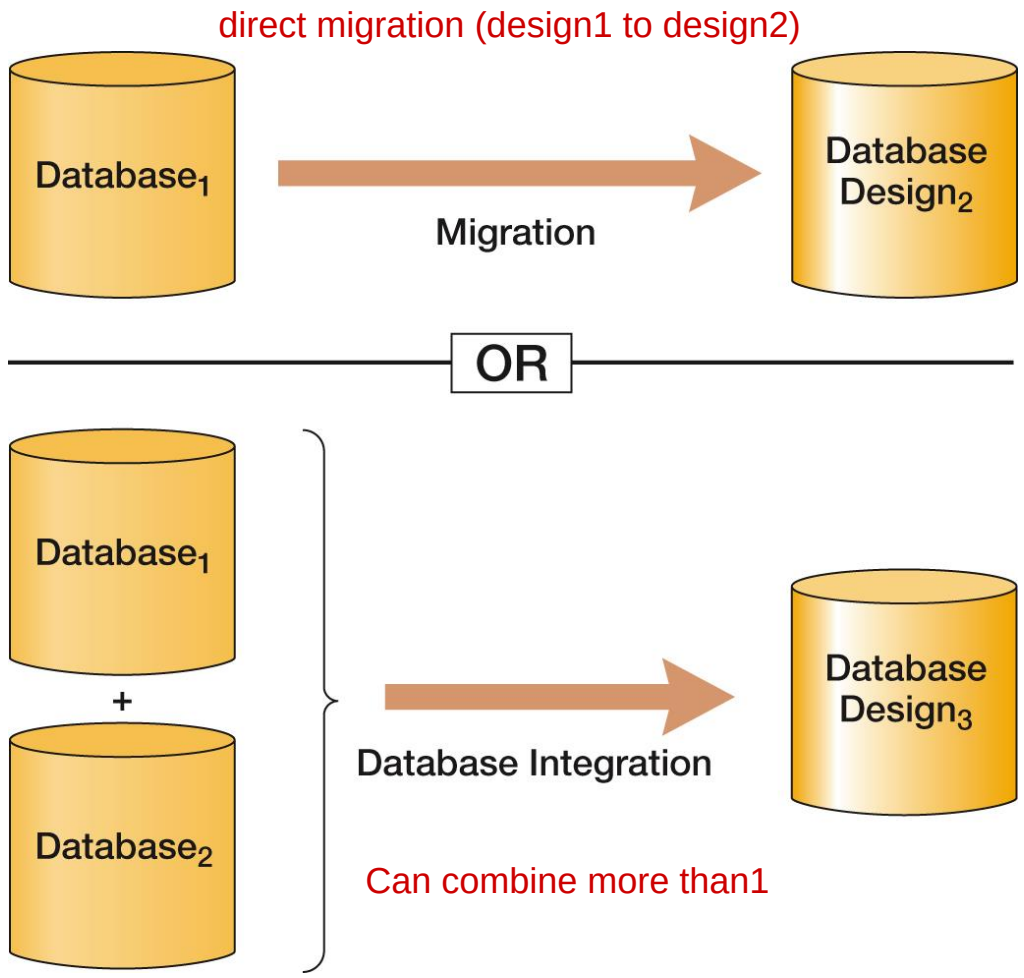


Figure 1-25

Working Domains of Knowledge Workers, Programmers, and Database Administrators

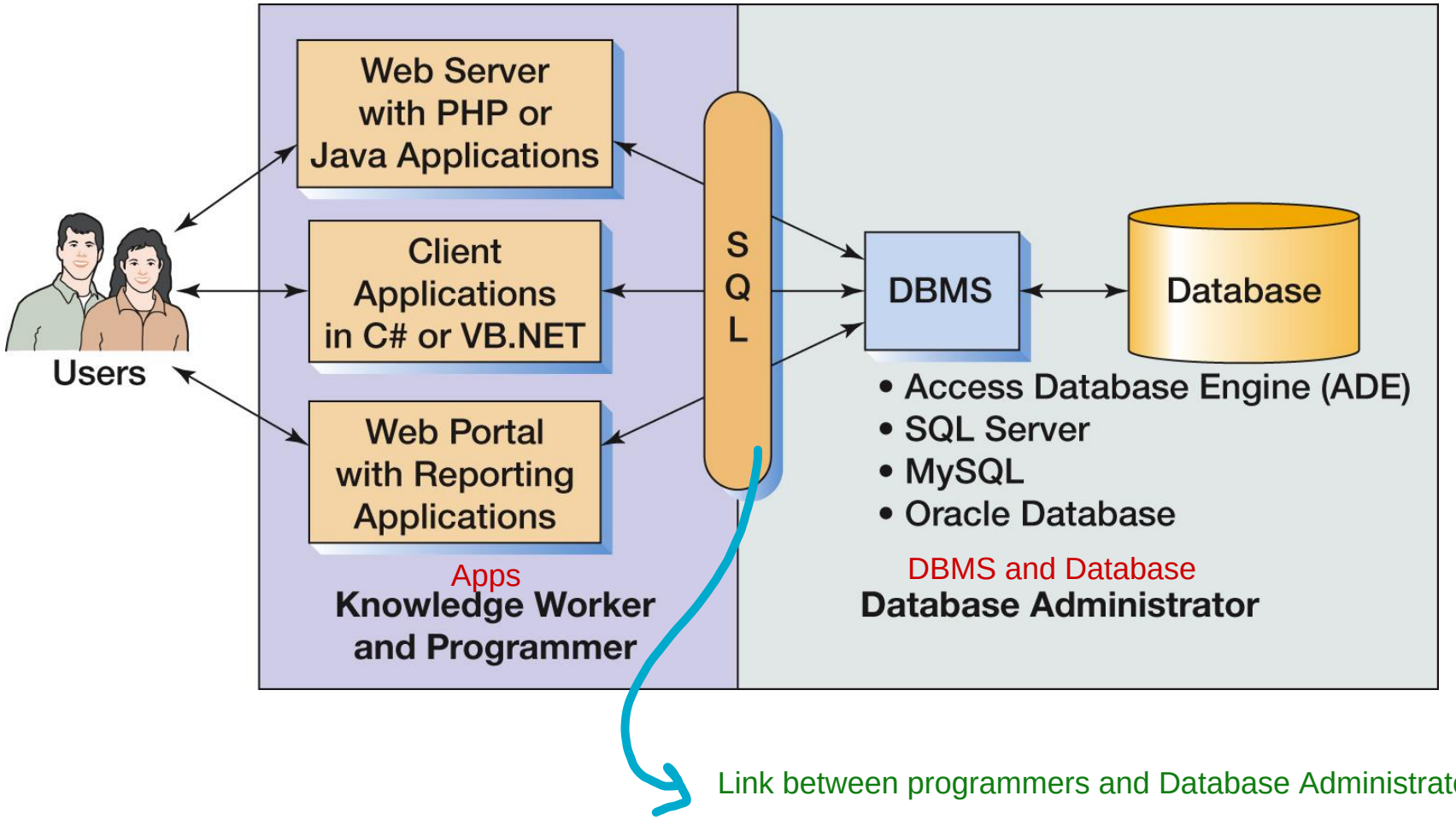


Figure 1-26
Priorities of What You Need to Know

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

Figure 1-27
Database History (1 of 2)

Important pages !!

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 Database, Ingres	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, Microsoft 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, Gemstone, O2, Versant	Never caught on. Required relational database to be converted. Too much work for perceived benefit.

Figure 1-27
Database History (2 of 2)

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 Can replace relational	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 I.
Big Data and the NoSQL movement	2009-present	Hadoop, Cassandra, Hbase, CouchDB, Arango DB, MongoDB, JSON and other products	Web applications such as Facebook and Twitter use Big Data technologies. The NoSQL movement is geared toward processing large data sets using NoSQL data models which replace relational databases with nonrelational data structures such as XML and JSON, and which may supplant relational databases during your career. See Chapter 12 and Appendices K and L.

The Relational Database Model

- The dominant database model is the **relational database model**—all current major DBMS products are based on it.
- It was created by IBM engineer **E. F. Codd** in 1970.
- It was based on mathematics called **relational algebra**.
- This text examines and explains the relational database mode.

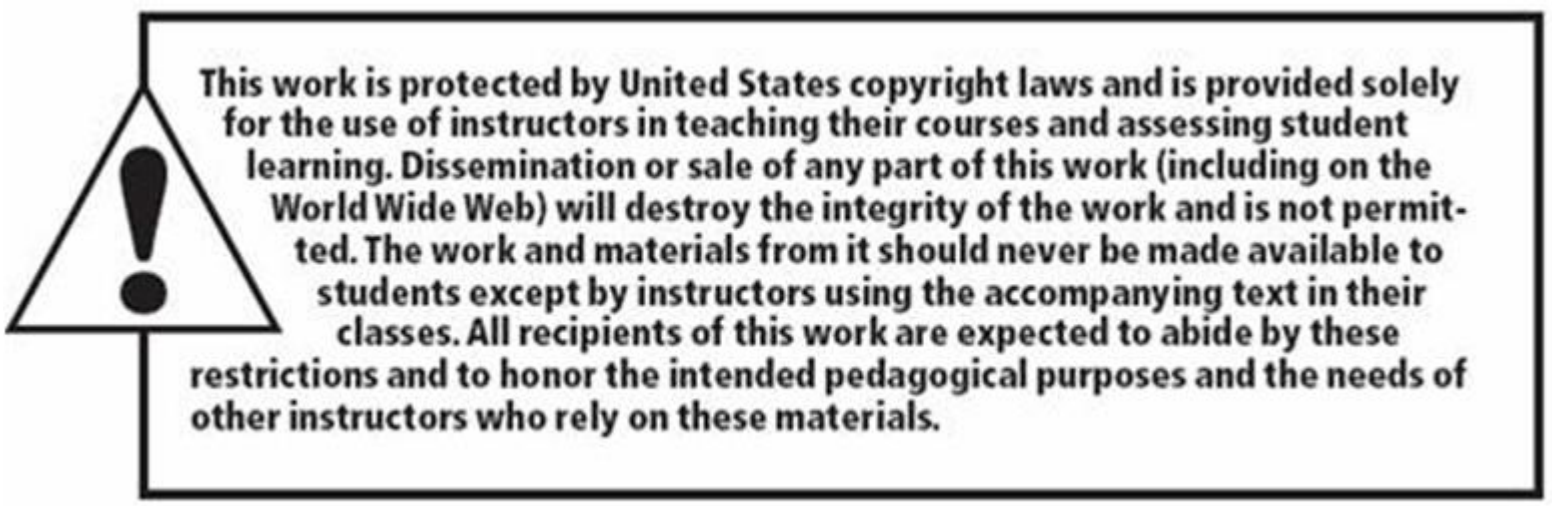
The NoSQL Movement and Big Data

- Recent developments in Internet and mobile computing have resulted in the development of non-relational DBMSs.
  - NoSQL movement
  - Big Data
- These do not replace the relational model, but rather complement it.
- These topics are discussed in Chapter 12 and Appendix I.

Database Processing
Fundamentals, Design, and Implementation (15th Edition)

End of Presentation:

Chapter One



ALGONQUIN COLLEGE

CST 2355 – Database Systems

Week 2

Topics Covered:

- Microsoft Access – User Forms
- Microsoft Access – Reports
- Microsoft Access – Queries
- Microsoft Access – Grouping
- Microsoft Access – External Data

Microsoft Access – User Forms …

- **Interactive** – allows data input using text fields, radio buttons, combo-boxes (pull-down lists),
- Can contain Buttons to invoke macros – run stored procedures (queries) or navigate to other forms
  - Open, Close, Hide, Show, …
- If the relationships (key/foreign key) constraints have been identified then the Wizard can suggest layouts for interactively browsing/updating data Layouts itself as per foreign key relationships
  - Constraints most easily entered using the "Database Tools > Relationships" E-R diagram.

… Microsoft Access – User Forms

- AutoExec macro can be used to launch initial form or can use a "Start Form"; can build an initial form as a Control Panel of options.
- If you are familiar with .Net programming, the programming environment and selection of widgets is natural
  - Microsoft Look and Feel

- *Demo of Relationships, User Form Creation, Form View, Layout View, Design View, Buttons, …*

Microsoft Access – Reports

- **Display of information** – could look like a "form" with text boxes, etc
- Used for grouping of data; e.g., Alphabetical directories of information, sorted data using a hierarchy of sort fields, etc
- Layout of the form and options for sorting are extensive (e.g., tabs, nested forms)
- If the relationships (key/foreign key) constraints have been identified then the Wizard can suggest layouts for displaying selected data
  - Constraints most easily entered using the "Database Tools > Relationships" E-R diagram.

Layouts itself as per foreign key relationships

- *Demo of Report Creation, Report View, Print Preview, Layout View, Design View, …*

Microsoft Access – Queries

- **A general SQL statement** – could be SELECT, UPDATE, DELETE, etc.
- IF "Simple" query, then the Query Wizard saves a SELECT statement and can then use it like a VIEW (i.e., virtual table)

```
e.g.,
SELECT field_names FROM my_table LEFT JOIN my_query ON join_condition;
```

  - Query Wizard handles multiple types of queries:
    - SELECT [Simple, Crosstab (aggregates numeric data across rows and columns), Find duplicates, or Find unmatched]
    - Make Table, Append, Update, Delete, …
  - *Demo of Query Creation: Simple, Make Table, Update, ….*

Microsoft Access – Grouping …

- **Grouping of data** – useful for Alphabetical directories of information, sorted data using a hierarchy of sort fields, etc
- **Underlying "GROUP BY"** – forms, reports, and queries have associated SQL that they are using to get the data for display.
  - Recall: general SQL syntax (**PostgreSQL**)

```
SELECT field_names
FROM tables_including_joins
WHERE some_condition_among_fields
GROUP BY some_grouping_fields
ORDER BY some_sort_fields
LIMIT number_of_items
;
```

… Microsoft Access – Grouping …

All have different grouping techniques

- **By the Way: LIMIT is not portable! – different databases have different mechanisms for limiting the number of rows returned**

MySQL and PostgreSQL –
select * from table limit *number_of_rows*

Microsoft SQL Server and Access –
select top *number_of_rows* * from table

LIMIT is not portable !!!!!!!!!!!!!!!

Oracle –
select * from table where rownum <= *number_of_rows*

DB2 –
select * from table fetch first *number_of_rows* rows only

… Microsoft Access – Grouping

- If the relationships (key/foreign key) constraints have been identified then the Wizards will suggest groupings for displaying selected data
  - Constraints are often most easily entered using the “Database Tools > Relationships” E-R diagram. Layouts itself as per foreign key relationships
- Can use the Query Design tools to access the SQL View and then build a custom query to use as the data source for a form or report.
  - In the design view of the Query tool, you can specify the grouping and ordering as attributes of each field in the query

    Design view allows to specify the grouping attributes of each field
- *Demo of Grouping in Query tool, …*

Microsoft Access – External Data

- **External Storage of information** – could be input or output or both
  - Database, MS-Excel worksheets, Comma-Separated-Value (CSV) files, …
  - Can be locally registered Open Database Connectivity data sources (more on this next week)
    - ODBC is a Windows-level list of connectors (one list for 32 bit software packages and another list for 64 bit software packages) and is very similar to Java Data Base Connectivity (JDBC); connection-based sessions with username/passwords.
    - MANY databases support ODBC by supplying an ODBC adapter for use with Windows.
- This means that an Access application can use the data "tables" or objects (e.g., views) that are either local to the workstation running Access, or remote, or any combination of local and remote objects.

access app can use data that can be local, remote or both !!

- *Demo of Excel input/output, ODBC connections, …*

ALGONQUIN COLLEGE

CST 2355 – Database Systems

Week 3

Topics Covered:

- Microsoft Access – Reports
- Microsoft Access – Queries (esp. Updates Queries)
- Microsoft Access – Triggers
- ODBC and connectivity

Microsoft Access – Reports (continued)

- **Display of information** – could look like a “form” with text boxes, etc
- Can have parameterized reports using “filter” parameters to provide specific values for parameters to be used in the data source for a report
  - Often a combo-box (i.e., value selected from a list generated from a query) is provided to the user to select a value
  - Can use Visual Basic for Applications (VBA) to build OnClick events, etc.
- *Demo of Report Selection Dialog (Northwind Sales reports example)*
  - *Use of Domain functions for lookups, counts, …*

Microsoft Access –Queries …

- **Query Types:**
  - **Select: query selects data from tables and nested queries (for use)**
    - **Similar to "CREATE VIEW *viewname* AS *query;"**
    - **Can access the query as a virtual table**
  - **Make Table: query selects data from the other tables and queries and saves the resulting data as a new table**
    - **Creates a snapshot of the data**
    - **Similar to "CREATE MATERIALIZED VIEW *mvname* AS *query;"**
  - **Append: query selects data from tables and queries and inserts the resulting data as new records in a particular table**
    - **Inserts a snapshot of the data**
    - **Similar to "INSERT INTO *mytable* (*fieldlist*) *query;"**
      - **Note: *query* must have fields as given in the *fieldlist***
  - **Update: updates selected records in a particular table based on given criteria to new value(s)**
  - **Delete: deletes selected records in a particular table based on given criteria**
  - **Crosstab: creates a result set aggregated by two "dimension" sets of values; one down the left side of the datasheet and the other across the top**
    - **Often used in generating data for display in financial reports; e.g., sum of number of sales with two dimensions as (product, location)**

Microsoft Access – … Queries

- **Best Practice:**
  - **First Step**
    - **Create a "SELECT" query (add tables, queries and choose the required fields)**
      - **Test the query by running it to make sure it selects the appropriate rows and fields**
  - **THEN:**
    - **Modify the type of the query to the type required (e.g., append, update, delete)**
    - **Save the query**
    - **Run the query**

Microsoft Access – Triggers

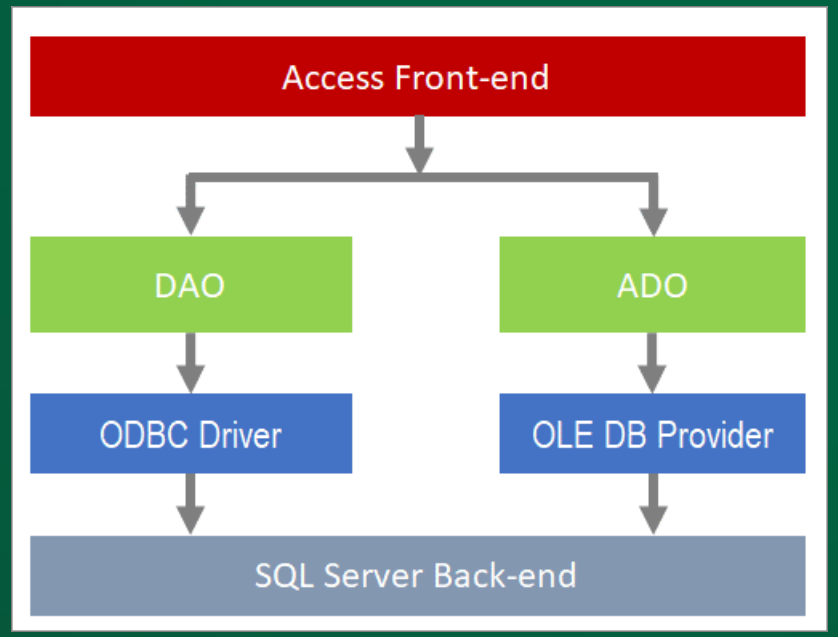
- In Access the event handlers are associated with a FORM (not the individual tables…)!
  - Events occur for records on forms when you move the focus to a different record, update data in a record, delete an existing record or records, or create a new record.
- Best Practice:
  - Create a form that allows updates to the table
  - Create event handlers associated with the form
- Order of Events for Database Objects
  - See Microsoft documentation at:
    - *Https://support.microsoft.com/en-us/office/order-of-events-for-database-objects-e76fbbfe-6180-4a52-8787-ce86553682f9*
    - *Or for just the form events:*
    - *https://support.microsoft.com/en-us/office/order-of-events-for-database-objects-e76fbbfe-6180-4a52-8787-ce86553682f9#bm2*
- *Demo of Form events*

Microsoft Access – ODBC and Connectivity …

- ODBC – according to Microsoft:
  - As at:
    - https://docs.microsoft.com/en-us/sql/odbc/microsoft-open-database-connectivity-odbc?view=sql-server-ver15#:~:text=The%20Microsoft%20Open%20Database%20Connectivity,specifically%20for%20relational%20data%20stores.
  - The Microsoft Open Database Connectivity (ODBC) interface is a C programming language interface that makes it possible for applications to access data from a variety of database management systems (DBMSs). ODBC is a low-level, high-performance interface that is designed specifically for relational data stores.
  - The ODBC interface allows maximum interoperability-an application can access data in diverse DBMSs through a single interface. Moreover, that application will be independent of any DBMS from which it accesses data. Users of the application can add software components called drivers, which interface between an application and a specific DBMS.
- MS-Access can use the ODBC drivers that are installed on your Windows environment, to connect to external data
- ODBC provides full access to the dbms – to run ANY sql statement…. Including data control statements
- *Demo of Connection to External Data using ODBC, …*

… Microsoft Access – ODBC and Connectivity

- Connecting Access to a Back-end server using ODBC is described at:
  - https://support.microsoft.com/en-us/office/connect-access-to-sql-server-050d88f3-b2d6-4e76-b6f9-f3c556f139ea



ODBC Drivers in Access/Windows

- Install the ODBC driver for the back-end database
- Create a “DSN” for the particular database instance
  - Data Source Name
- Use the Windows-level DSN when linking or downloading items from the external data source from Access

- *Demo of ODBC connections,…*

ALGONQUIN COLLEGE

CST 2355 – Database Systems

Week 4

Topics Covered:

- Microsoft SQL Server
  - Features
  - Security
  - Upsizing
- ODBC and connectivity

Microsoft SQL Server - Introduction

- https://en.wikipedia.org/wiki/Microsoft_SQL_Server
- "Microsoft SQL Server is a relational database management system developed by Microsoft. As a database server, it is a software product with the primary function of storing and retrieving data as requested by other software applications—which may run either on the same computer or on another computer across a network (including the Internet). Microsoft markets at least a dozen different editions of Microsoft SQL Server, aimed at different audiences and for workloads ranging from small single-machine applications to large Internet-facing applications with many concurrent users."
- Major Features:
  - Built for client-server applications: secure connections over network to remote server
    - Has variety of adapters (e.g., ODBC (32 bit, 64 bit), JDBC, ETC.!)
  - Toolkits for Data warehousing, analysis, etc.
  - Scalability:
    - Large numbers of users, Petabyte-sized databases
  - Security:
    - Users, roles, encrypted connections
  - Concurrency:
    - Supports large number of concurrent users with an efficient locking strategy
  - Availability:
    - Has high-availability options for single-site installations or multi-site replication

Microsoft SQL Server - Features

- **All the usual database features:** triggers, stored procedures, …
- **Watch the video at: What's new in SQL Server 2019 (15.x)**

  https://docs.microsoft.com/en-us/sql/sql-server/what-s-new-in-sql-server-ver15?view=sql-server-ver15&viewFallbackFrom=sqlallproducts-allversions
- **Editions and supported features of SQL Server 2019 (15.x)**

  https://docs.microsoft.com/en-us/sql/sql-server/editions-and-components-of-sql-server-version-15?view=sql-server-ver15&viewFallbackFrom=sql-server-2017
- *Demo of SQL Server Management Studio … basic features*
  - *Projects, Database, Security, Diagrams, …*

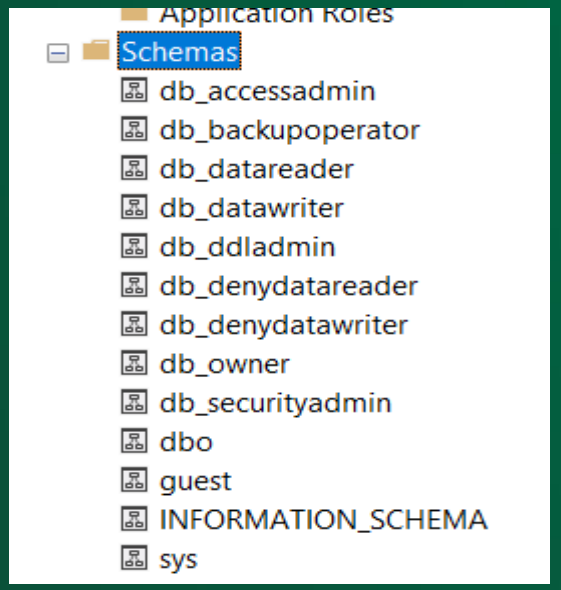
Microsoft SQL Server - Security

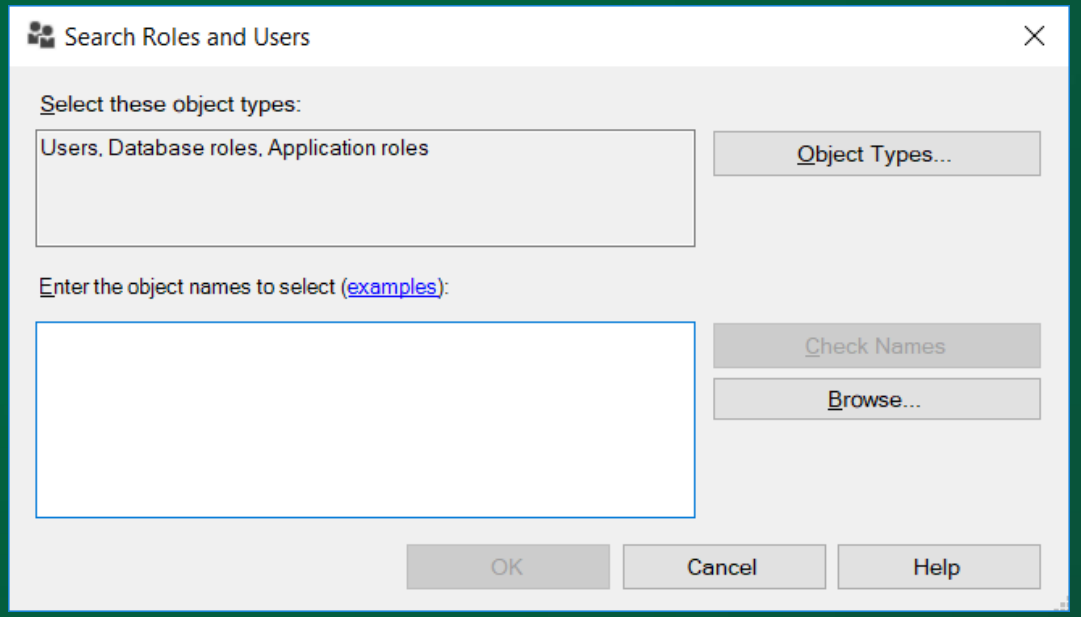
- Applications connect to the SQL Server as a “login”
  - Each database has users with “user name” associated with a login.
  - Typically: applicationAdminUser, applicationUser
  - **NOT** logins for each end user of the application…
  - Each user has individual default database

- System administrators of the database have very different sets of privileges
- Control privilege assignment through “roles”
  - Assign privileges to role, then add users to the role through “membership”

Microsoft SQL Server - Schema

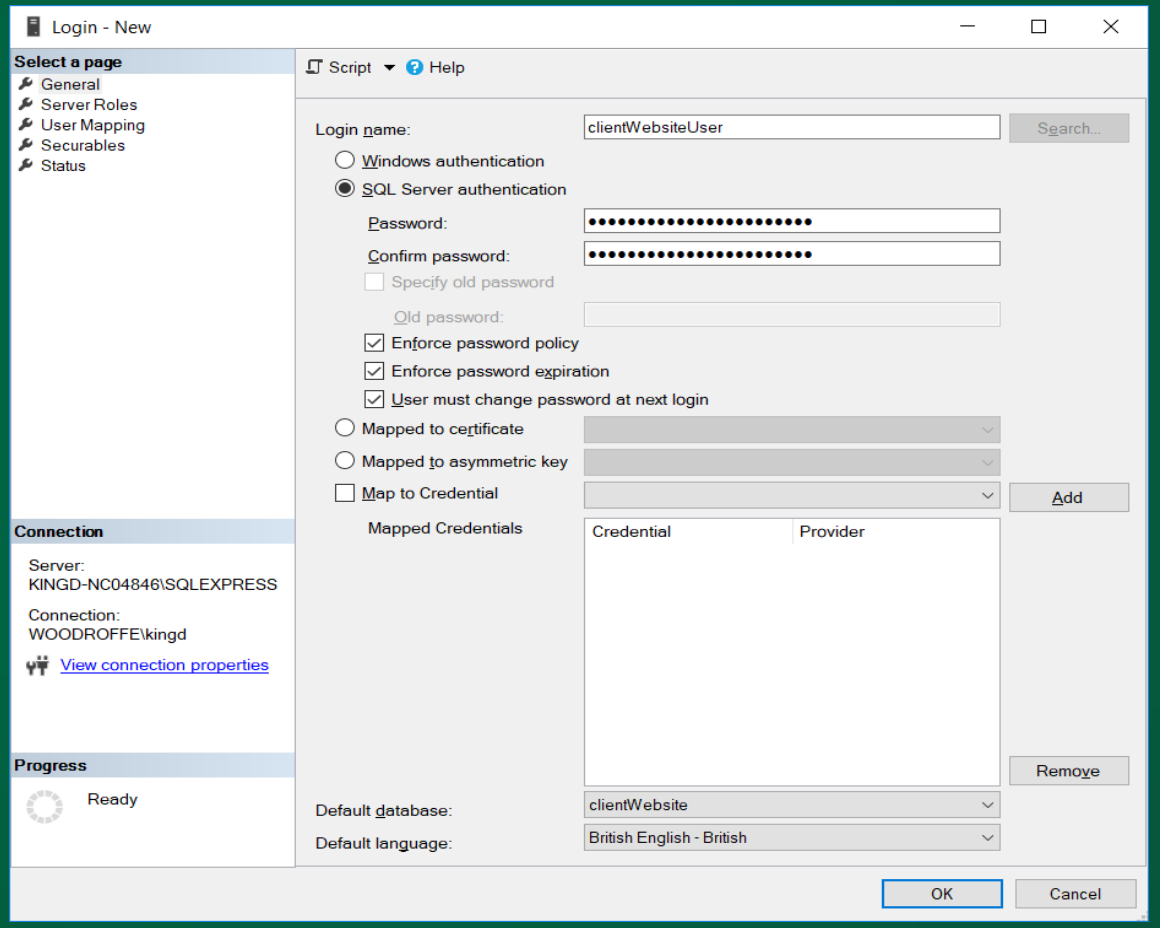
- Can group users by roles
- Can group database objects using “Schema”
  - Create using SSMS under “Security” within database: owned by user, database role, or application role





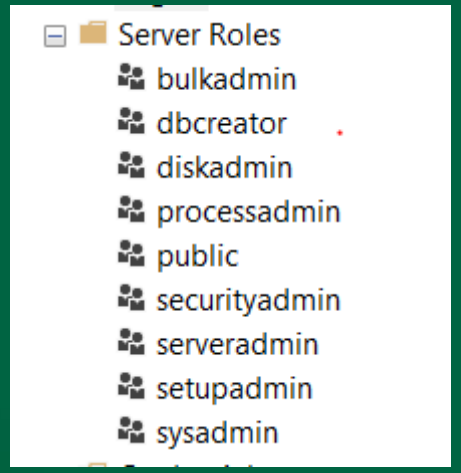
Microsoft SQL Server - Security

- Creating user login:
- Note:
  - Can use Public Key Infrastructure Certificates to have individualized data encryption on connections



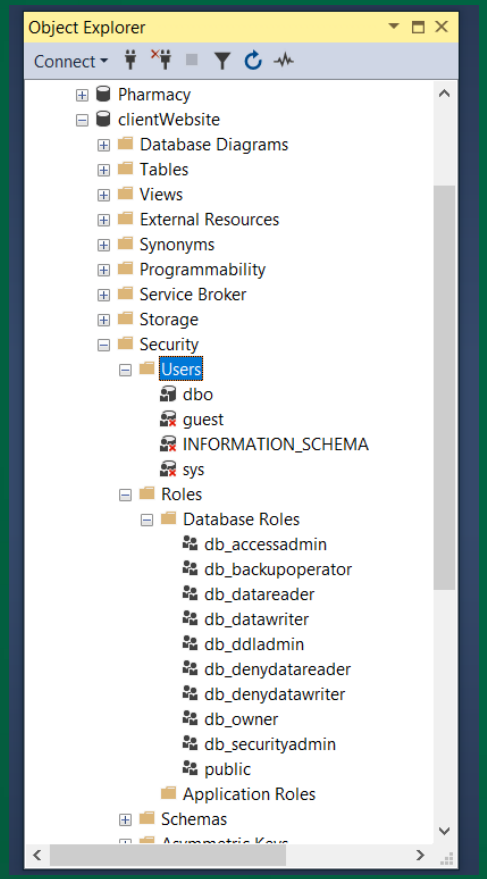
Microsoft SQL Server – Server Roles

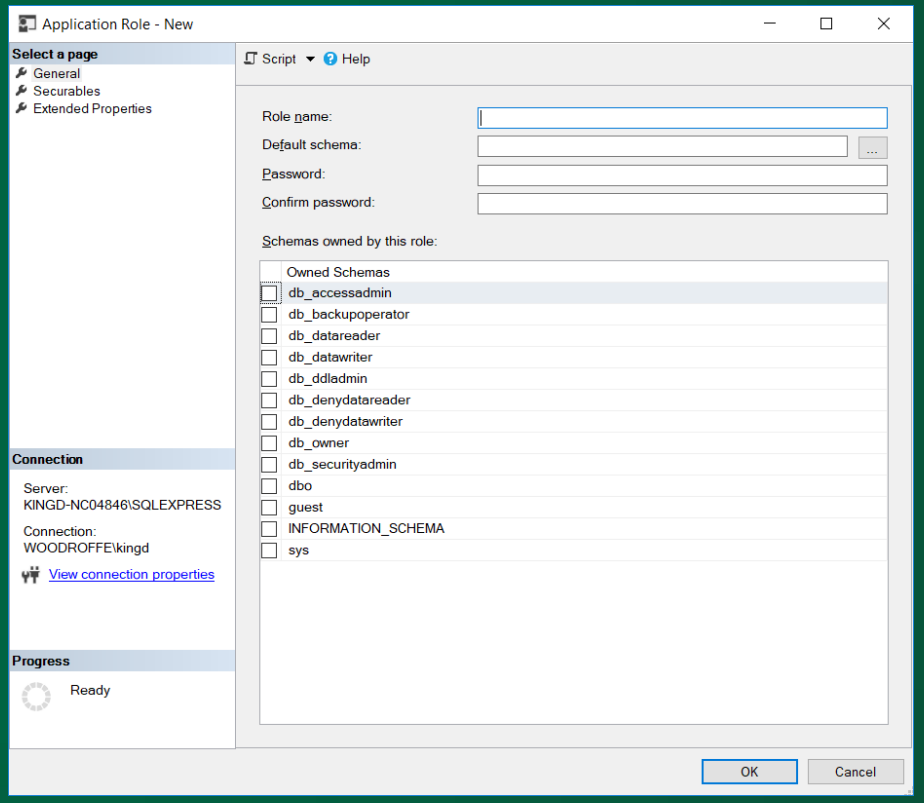
- Built-in Server Roles:



Microsoft SQL Server – Roles

- Database Roles:
- Application Roles:





Microsoft Access – Upsizing

- **Can import data directly using the Data Import Wizard in Microsoft SQL Server (Excel, Access, ODBC sources, SQL Native, …)**
  - **Either import from 32 bit data sources using the built-in Wizard,**
  - **Or, use the 64 bit version of the importation wizard in Windows**
    - **"SQL Server 2019 Import and Export Data (64bit)"**
- **If you are moving from Microsoft Access and want to get all the foreign key relationships (ETC.) imported "easily", you can install "Microsoft SQL Server Migration Assistant for Access"**
  - **Documentation:**

    https://docs.microsoft.com/en-us/sql/ssma/access/sql-server-migration-assistant-for-access-accesstosql?view=sql-server-ver15
  - **Download:**

    https://www.microsoft.com/en-us/download/details.aspx?id=54255

SQL Server – ODBC

- SQL Server has both 32 bit and 64 bit Open Database Connectivity adapters
  - **Recall**: ODBC is a Windows-level list of connectors (one list for 32 bit software packages and another list for 64 bit software packages) and is very similar to Java Data Base Connectivity (JDBC); connection-based sessions with username/passwords.
    - MANY databases support ODBC by supplying an ODBC adapter for use with Windows.
- **Using ODBC allows applications to access SQL Server through a "standardized" Application Programming Interface**

ODBC Drivers in Access/Windows

- Install the ODBC driver for the back-end database
- Create a “DSN” for the particular database instance
  - Data Source Name
- Use the Windows-level DSN when linking or downloading items from the external data source from Access

- *Demo of ODBC connections,…*

ODBC Connection Model

- https://en.wikipedia.org/wiki/Open_Database_Connectivity

  “In computing, Open Database Connectivity (ODBC) is a standard application programming interface (API) for accessing database management systems (DBMS). The designers of ODBC aimed to make it independent of database systems and operating systems.[citation needed] An application written using ODBC can be ported to other platforms, both on the client and server side, with few changes to the data access code.

  ODBC accomplishes DBMS independence by using an ODBC driver as a translation layer between the application and the DBMS. The application uses ODBC functions through an ODBC driver manager with which it is linked, and the driver passes the query to the DBMS. An ODBC driver can be thought of as analogous to a printer driver or other driver, providing a standard set of functions for the application to use, and implementing DBMS-specific functionality. An application that can use ODBC is referred to as "ODBC-compliant". Any ODBC-compliant application can access any DBMS for which a driver is installed. Drivers exist for all major DBMSs, many other data sources like address book systems and Microsoft Excel, and even for text or comma-separated values (CSV) files.Install the ODBC driver for the back-end database”

- Connection-based: connect with username/password, execute commands …, close connection.

- *Microsoft also provides a JDBC driver for SQL Server*

ALGONQUIN COLLEGE

CST 2355 – Database Systems

Week 5

Topics Covered:

- Microsoft SQL Server
  - Modelling
  - Administration
  - Security
  - Functions

Microsoft SQL Server - Modelling

- Microsoft SQL Server is inherently a **relational database**
- **Consider the Entity-Relationship model**
  - **Entities**
    - **Real-world objects (things) and events**
      - **Who, what, where, why, when, how**
  - **Relations**
    - **What the entities share**
      - **Part-of, attribute-of, is-a**
- **Important considerations**
  - **Do we need multi-valued fields?**
    - **Historical (time-based) versions of data (e.g., previous address)**
    - **Multi-lingual text fields**

Modelling Common Sense

- If we start with real-world entities and relationships
  - Then group the entities into tables containing the single-valued attributes
  - Then put multi-valued attributes into their own tables; each with either
    - If 1:N: Add a foreign key in the child (attribute) table containing the parent key, or
    - If N:N: Add a new table containing the groups (typ. Pairs) of parent keys and child keys.
  - Then add either individual fields or tables to capture the relationships between objects
    - If 1:N: Add a foreign key in the child (attribute) table containing the parent key, or
    - If N:N: Add a new table containing the pairs of parent keys and child keys.
- This will result in a 3rd Normal Form (3NF) model and if there are no hidden dependencies among fields (e.g., Postal Codes determine the City and Province…) then it would be Boyce-Codd Normal Form (BCNF)
- Handling all multi-valued fields as separate tables instead of as multi-column fields ensures 4th Normal Form.
- Moving all storage for attributes shared among objects (e.g., City Names in a particular language) into a separate table will ensure 5th Normal Form.
  - E.g., each field in a separate table; allowing sharing of attributes among objects and preventing ANY duplication

Advantages of Normalization

- No duplicate data (or minimal duplicate data) will simplify & speed-up updates
  - We want to avoid mistakes: “ANOMALIES” that could happen if there were service interruptions in the middle of a transaction
    - Update, Delete, Insert
- We will not have to guess which copy is the “Source of truth”?
  - Even when doing a “SELECT” we are trusting that all duplicates of a particular data field have been updated consistently
    - E.g., change a person’s lastname: If it is only stored in one place, then there is no possible inconsistency. If it has been stored in multiple locations, the updates are more complicated and require a **transaction**
    - **(Atomic, Consistent, Isolated, Durable) ACID transactions.**

Disadvantages of Normalization

- No duplicate data (or minimal duplicate data) will mean that we need to have JOINS across tables to retrieve related data
  - This will be somewhat slower than if the data was already grouped
- **Data that was "shared" in error; when updated will affect all related objects**
  - **E.g., if two people have the same phone number, and then one person updates their phone number – does it affect both people? It complicates the update process in that sometimes data needs to be retained for some of the related objects.**
- Both relationships and the data values are updatable

Microsoft SQL Server – Diagrams

- Using Microsoft SQL Server Management Studio: can edit tables and relationships through diagrams
  - Create diagram(s)
  - Add tables
    - Set Keys
  - Add relationships
  - Save…..

- *Demo of SQL Server Management Studio diagram creation …*

Physical Data Models

- Divide data into Databases
  - Divide database into Schemas (typ. “Subject Areas”)
  - Divide Subject Areas into Normalized Tables
    - Each object or event type becomes a table
    - Each relationship becomes either a field within the parent table or a separate named table, perhaps with a 3rd table containing the N:N relationship grouping.
      - 1:1 (using 1 table); 1:N (2 tables); N:N (3 tables)
    - Multi-lingual and multi-value typically means 1:N relationships
- Use security access controls based on table, schema, database
  - Create views, stored procedures, or triggers on tables (or with JOINed tables) to control field-level access
  - Create materialized views to provide historical data snapshots.
  - *Can provide field-level control by disabling access to a table and then allowing access to a view that contains only a subset of the table’s fields*

Microsoft SQL Server - Administration

- Provide a set of “login”s on the server (locally connected – use Windows; remote – username/password)
  - When connected, the login has a default database.
- Each database has a set of users mapped to the server logins
- The database users are members of a “role”: either Database or Application role
- For a new application:
  1. Create a database
  2. Create users and roles within the database (mapped to server logins)
  3. Create schemas within the database – groupings of related data
     - Could be based on subjects, backup groupings (e.g., product catalog) or based on security groupings
  4. Create physical tables. Indexes, triggers, etc. associated with the schemas
  5. Grant privileges to the roles
  6. Add users to the roles

Microsoft SQL Server - Security

- Granting privileges to roles (or directly to users)
  - E.g, “GRANT SELECT ON HRSCHEMA.EMPLOYEES TO WEBAPPUSER;”

```
GRANT system_privilege
TO rolename_or_username;
```

```
GRANT object_privilege ON  database_object_name
TO rolename_or_username;
```

- A long list of privileges (depening upon the object) including:
  - SELECT, INSERT, UPDATE, DELETE, DROP TABLE, CREATE VIEW, ….
- Try these interesting queries:

```
SELECT TABLE_SCHEMA, TABLE_NAME FROM information_schema.tables;

SELECT 'GRANT select ON ' + '"' + TABLE_SCHEMA + '"."' + TABLE_NAME + '" TO myusername '
FROM information_schema.tables;
```

Microsoft SQL Server - Functions

- MANY built-in functions
  - See the W3C Schools site:
  - https://www.w3schools.com/sql/sql_ref_sqlserver.asp

- Can create your own functions to do server-side “calculations”
  - Simple “Scalar” functions that return a single field, or “Table Valued” returning a table!

```
SELECT  e.lastname, e.firstname, dbo.friendcount(e.ID) ‘Number of possible friends’
FROM  employees e;
```

  - **Nice tutorial at: Create User-defined Functions (Database Engine)**
  - https://docs.microsoft.com/en-us/sql/relational-databases/user-defined-functions/create-user-defined-functions-database-engine?view=sql-server-ver15

Microsoft SQL Server – Stored Procedures

- **Create a Stored Procedure**
  - https://docs.microsoft.com/en-us/sql/relational-databases/stored-procedures/create-a-stored-procedure?view=sql-server-ver15
- An example: Stored in Schema created under security in database

```
USE AdventureWorks2012;
GO
CREATE PROCEDURE HumanResources.uspGetEmployeesTest2
    @LastName nvarchar(50),
    @FirstName nvarchar(50)
AS
    SET NOCOUNT ON;
    SELECT FirstName, LastName, Department
    FROM HumanResources.vEmployeeDepartmentHistory
    WHERE FirstName = @FirstName AND LastName = @LastName
    AND EndDate IS NULL;
GO
```

Microsoft SQL Server – Triggers

Lots of examples at:

https://docs.microsoft.com/en-us/sql/t-sql/statements/create-trigger-transact-sql?view=sql-server-ver15

- "a special type of stored procedure that automatically runs when an event occurs in the database server. You can modify data before it is saved, write audit logs, send email, etc.
- **DML triggers** run when a user tries to modify data through a data manipulation language (DML) event. DML events are INSERT, UPDATE, or DELETE statements on a table or view. These triggers fire when any valid event fires, whether table rows are affected or not. For more information, see DML Triggers.
- **DDL triggers** run in response to a variety of data definition language (DDL) events. These events primarily correspond to Transact-SQL CREATE, ALTER, and DROP statements, and certain system stored procedures that perform DDL-like operations.
- **Logon triggers** fire in response to the LOGON event that's raised when a user's session is being established. You can create triggers directly from Transact-SQL statements or from methods of assemblies that are created in the Microsoft .NET Framework common language runtime (CLR) and uploaded to an instance of SQL Server.
- SQL Server lets you create multiple triggers for any specific statement."

Microsoft SQL Server – Trigger Example

```
-- This trigger prevents a row from being inserted in the Purchasing.PurchaseOrderHeader
-- table when the credit rating of the specified vendor is set to 5 (below average).

CREATE TRIGGER Purchasing.LowCredit ON Purchasing.PurchaseOrderHeader
AFTER INSERT
AS
IF (ROWCOUNT_BIG() = 0)
RETURN;
IF EXISTS (SELECT *  FROM Purchasing.PurchaseOrderHeader AS p
        JOIN inserted AS i   ON p.PurchaseOrderID = i.PurchaseOrderID
        JOIN Purchasing.Vendor AS v   ON v.BusinessEntityID = p.VendorID
        WHERE v.CreditRating = 5  )
BEGIN
RAISERROR ('A vendor''s credit rating is too low to accept new purchase orders.', 16, 1);
ROLLBACK TRANSACTION;
RETURN
END;
GO
```