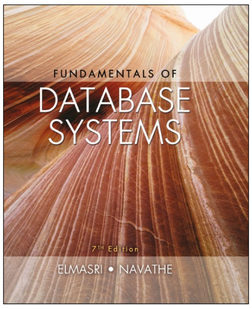
# Fundamentals of Database Systems

Seventh Edition



**Chapter 2**

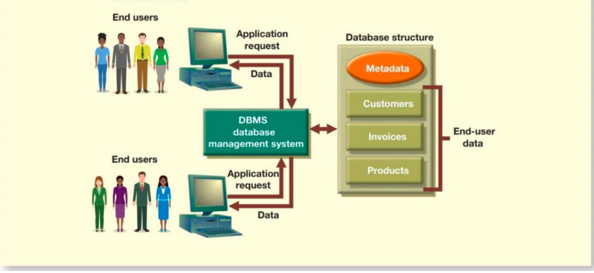
Database System Concepts and Architecture

Pearson

1

# DBMS

FIGURE 1.2 The DBMS manages the interaction between the end user and the database



Pearson

3

# Database Models

- Collection of logical constructs used to represent data structure and relationships within the database
  - Conceptual models: focuses on logical nature of data representation, *what* is represented rather than *how*
    - the entity relationship model
    - the object-oriented model
  - Implementation models: emphasis on how the data are represented in the database or on how the data structures are implemented to represent what is modeled
    - the hierarchical database model
    - the network database model
    - the relational database model

Pearson

11

# Database Models (Con't.)

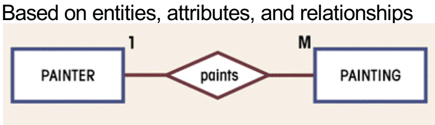
- Conceptual models use 3 types of relationships to describe associations among data;
  - One-to-many (1:M)
    - "PAINTER paints PAINTING" relationship is labeled as 1:M
    - "CUSTOMER generates INVOICE" relationship is labeled as 1:M
  - Many-to-many (M:N)
    - "STUDENT takes COURSE" relationship is labeled as M:N
    - "EMPLOYEE learns SKILL" relationship is labeled as M:N
  - One-to-one (1:1)
    - "EMPLOYEE manages STORE" relationship is labeled as 1:1

Pearson

12

# Entity Relationship Database Model

- A graphical representation of entities and their relationships
- Completes the relational data model concepts
- Represented in an entity relationship diagram (ERD)
- Based on entities, attributes, and relationships



**entity**: a person, place, or thing about which data are collected and stored

Each entity is described by a set of *attributes*.

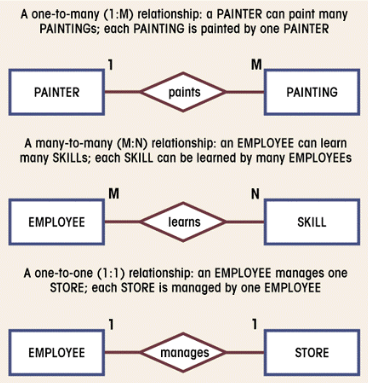
**attribute**: a particular characteristic of the entity

**relationship**: an association among data (**connectivity**)

Pearson

16

Relationship Depiction: The ERD


A one-to-many (1:M) relationship: a PAINTER can paint many PAINTINGs; each PAINTING is painted by one PAINTER

A many-to-many (M:N) relationship: an EMPLOYEE can learn many SKILLs; each SKILL can be learned by many EMPLOYEEs

A one-to-one (1:1) relationship: an EMPLOYEE manages one STORE; each STORE is managed by one EMPLOYEE

Pearson

17

## Relational Database Model

- A collection of tables in which data are stored
- Tables consist of rows and columns
- Tables related to each other by sharing a common characteristic

Pearson

18

## Relational Database Model (Con’t.)

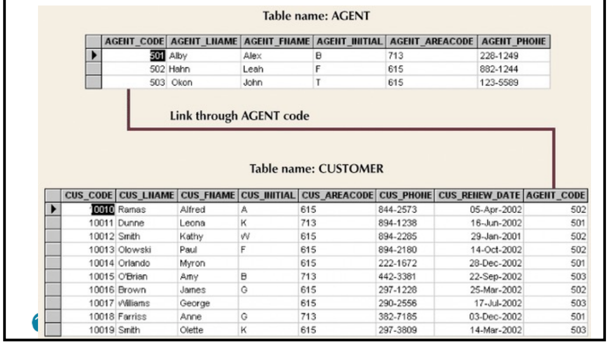


Table name: AGENT

| | AGENT_CODE | AGENT_LNAME | AGENT_FNAME | AGENT_INITIAL | AGENT_AREACODE | AGENT_PHONE |
|---|---|---|---|---|---|---|
| ▶ | 501 | Alby | Alex | B | 713 | 228-1249 |
| | 502 | Hahn | Leah | F | 615 | 882-1244 |
| | 503 | Okon | John | T | 615 | 123-5589 |

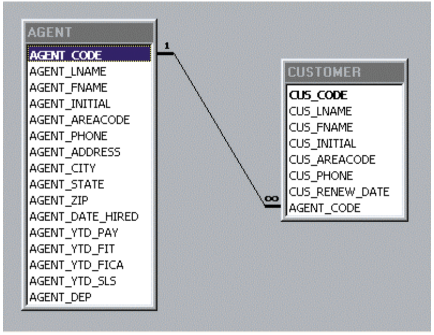
Link through AGENT code

Table name: CUSTOMER

| | CUS_CODE | CUS_LNAME | CUS_FNAME | CUS_INITIAL | CUS_AREACODE | CUS_PHONE | CUS_RENEW_DATE | AGENT_CODE |
|---|---|---|---|---|---|---|---|---|
| ▶ | 10010 | Ramas | Alfred | A | 615 | 844-2573 | 05-Apr-2002 | 502 |
| | 10011 | Dunne | Leona | K | 713 | 894-1238 | 16-Jun-2002 | 501 |
| | 10012 | Smith | Kathy | W | 615 | 894-2285 | 29-Jan-2001 | 502 |
| | 10013 | Olowski | Paul | F | 615 | 894-2180 | 14-Oct-2002 | 502 |
| | 10014 | Orlando | Myron | | 615 | 222-1672 | 28-Dec-2002 | 501 |
| | 10015 | O'Brian | Amy | B | 713 | 442-3381 | 22-Sep-2002 | 503 |
| | 10016 | Brown | James | G | 615 | 297-1228 | 25-Mar-2002 | 502 |
| | 10017 | Williams | George | | 615 | 290-2556 | 17-Jul-2002 | 503 |
| | 10018 | Farriss | Anne | G | 713 | 382-7185 | 03-Dec-2002 | 501 |
| | 10019 | Smith | Olette | K | 615 | 297-3809 | 14-Mar-2002 | 503 |

19

## Relational Database Model (Con’t.)



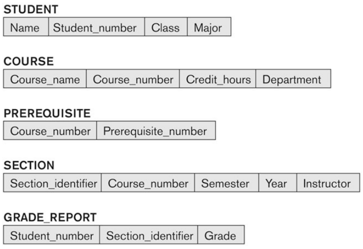
Relational Schema

Pearson

20

## Example of a Database Schema

**Figure 2.1** Schema diagram for the database in Figure 1.2.



STUDENT

| Name | Student_number | Class | Major |
|---|---|---|---|

COURSE

| Course_name | Course_number | Credit_hours | Department |
|---|---|---|---|

PREREQUISITE

| Course_number | Prerequisite_number |
|---|---|

SECTION

| Section_identifier | Course_number | Semester | Year | Instructor |
|---|---|---|---|---|

GRADE_REPORT

| Student_number | Section_identifier | Grade |
|---|---|---|

Pearson

21

## Data Models

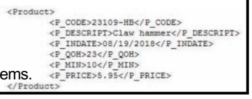
- **Data Model:**
  - A set of concepts to describe the **structure** of a database, the **operations** for manipulating these structures, and certain **constraints** that the database should obey.
- **Data Model Structure and Constraints:**
  - Constructs are used to define the database structure
  - Constructs typically include **elements** (and their **data types**) as well as groups of elements (e.g. **entity, record, table**), and **relationships** among such groups
  - Constraints specify some restrictions on valid data; these constraints must be enforced at all times
- **Data Model Operations:**
  - These operations are used for specifying database retrievals and updates by referring to the constructs of the data model.
  - Operations on the data model may include **basic model operations** (e.g. generic insert, delete, update) and **user-defined operations** (e.g.

Pearson

22

## Categories of Data Models

- **Conceptual (high-level, semantic) data models:**
  - Provide concepts that are close to the way many users perceive data.
    - (Also called **entity-based** or **object-based** data models.)
- **Physical (low-level, internal) data models:**
  - Provide concepts that describe details of how data is stored in the computer. These are usually specified in an ad-hoc manner through DBMS design and administration manuals
- **Implementation (representational) data models:**
  - Provide concepts that fall between the above two, used by many commercial DBMS implementations (e.g. relational data models used in many commercial systems).
- **Self-Describing Data Models:**
  - Combine the description of data with the data values. Examples include XML, key-value stores and some NOSQL systems.



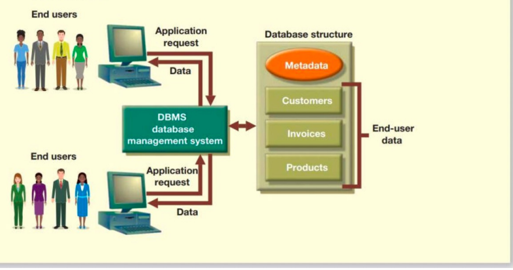
```
<Product>
        <P_CODE>23109-HB</P_CODE>
        <P_DESCRIPT>Claw hammer</P_DESCRIPT>
        <P_INDATE>08/19/2018</P_INDATE>
        <P_QOH>23</P_QOH>
        <P_MIN>10</P_MIN>
        <P_PRICE>5.95</P_PRICE>
</Product>
```

Pearson

24

## DBMS

**FIGURE 1.2** The DBMS manages the interaction between the end user and the database



## Schemas Versus Instances (1 of 2)

- Database Schema:
  - – The **description** of a database.
  - – Includes descriptions of the database structure, data types, and the constraints on the database.
- Database State:
  - – The actual data stored in a database at a **particular moment in time**. This includes the collection of all the data in the database.
  - – Also called database instance (or occurrence or snapshot).
    - ▪ The term **instance** is also applied to individual database components, e.g. **record instance, table instance, entity instance**

## Database Schema Vs Database State (1 of 2)

- Database State:
  - – Refers to the **content** of a database at a moment in time.
- Distinction
  - – The **database schema** changes very infrequently.
  - – The **database state** changes every time the database is updated.

## Example of a Database Schema

**Figure 2.1** Schema diagram for the database in Figure 1.2.

STUDENT

| Name | Student_number | Class | Major |
|---|---|---|---|

COURSE

| Course_name | Course_number | Credit_hours | Department |
|---|---|---|---|

PREREQUISITE

| Course_number | Prerequisite_number |
|---|---|

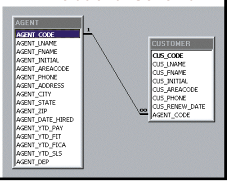
SECTION

| Section_identifier | Course_number | Semester | Year | Instructor |
|---|---|---|---|---|

GRADE_REPORT

| Student_number | Section_identifier | Grade |
|---|---|---|

Relational Schema



## Example of a Database State (1 of 4)

**Figure 1.2** A database that stores student and course information.

**COURSE**

| Course_name | Course_number | Credit_hours | Department |
|---|---|---|---|
| Intro to Computer Science | CS1310 | 4 | CS |
| Data Structures | CS3320 | 4 | CS |
| Discrete Mathematics | MATH2410 | 3 | MATH |
| Database | CS3380 | 3 | CS |

## Example of a Database State (2 of 4)

**Figure 1.2** A database that stores student and course information.

**SECTION**

| Section_identifier | Course_number | Semester | Year | Instructor |
|---|---|---|---|---|
| 85 | MATH2410 | Fall | 04 | King |
| 92 | CS1310 | Fall | 07 | Anderson |
| 102 | CS3320 | Spring | 05 | Knuth |
| 112 | MATH2410 | Fall | 05 | Chang |
| 119 | CS1310 | Fall | 05 | Anderson |
| 135 | CS3380 | Fall | 05 | Stone |

## Example of a Database State (3 of 4)

**Figure 1.2** A database that stores student and course information.

**GRADE_REPORT**

| Student_number | Section_identifier | Grade |
|---|---|---|
| 17 | 112 | B |
| 17 | 119 | C |
| 8 | 85 | A |
| 8 | 92 | A |
| 8 | 102 | B |
| 8 | 135 | A |

35

## Example of a Database State (4 of 4)

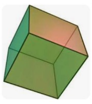
**Figure 1.2** A database that stores student and course information.

**PREREQUISITE**

| Course_number | Prerequisite_number |
|---|---|
| CS3380 | CS3320 |
| CS3380 | MATH2410 |
| CS3320 | CS1310 |

36

## Three-Schema Architecture (1 of 3)



- Proposed to support DBMS characteristics of:
  - **Program-data independence**.
  - Support of **multiple views** of the data.
- Not explicitly used in commercial DBMS products, but has been useful in explaining database system organization

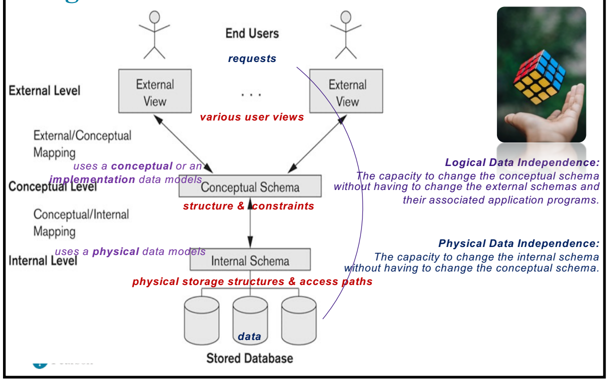
37

## Three-Schema Architecture (2 of 3)

- Defines DBMS schemas at **three** levels:
  - **Internal schema** at the internal level to describe physical storage structures and access paths (e.g indexes).
    - Typically uses a **physical** data model.
  - **Conceptual schema** at the conceptual level to describe the structure and constraints for the whole database for a community of users.
    - Uses a **conceptual** or an **implementation** data model.
  - **External schemas** at the external level to describe the various user views.
    - Usually uses the same data model as the conceptual schema.

38

## Figure 2.2 The Three-Schema Architecture


End Users
requests
External Level
External View
. . .
External View
various user views
External/Conceptual Mapping
uses a **conceptual** or an **implementation** data models
Conceptual Level
Conceptual Schema
structure & constraints
Conceptual/Internal Mapping
uses a **physical** data models
Internal Level
Internal Schema
physical storage structures & access paths
data
Stored Database

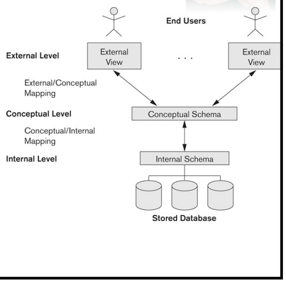
***Logical Data Independence:***
*The capacity to change the conceptual schema without having to change the external schemas and their associated application programs.*

***Physical Data Independence:***
*The capacity to change the internal schema without having to change the conceptual schema.*

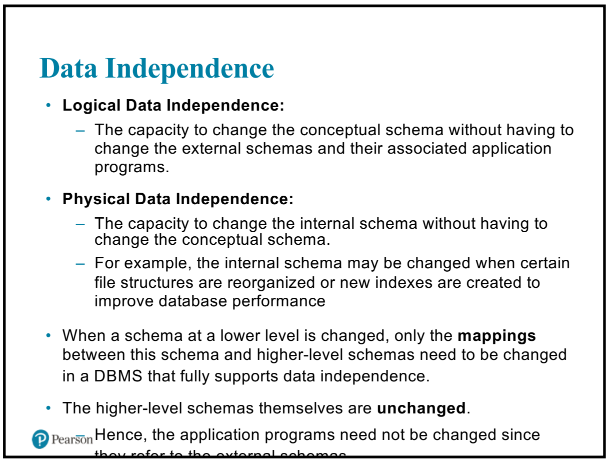
39

## Three-Schema Architecture (3 of 3)

- Mappings among schema levels are needed to transform requests and data.
  - Programs refer to an external schema, and are mapped by the DBMS to the internal schema for execution.
  - Data extracted from the internal DBMS level is reformatted to match the user's external view (e.g. formatting the results of an SQL query for display in a Web page)



40

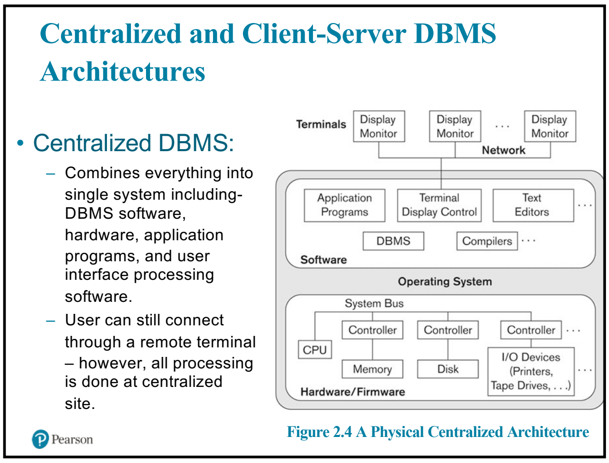


## Data Independence

- **Logical Data Independence:**
  - The capacity to change the conceptual schema without having to change the external schemas and their associated application programs.
- **Physical Data Independence:**
  - The capacity to change the internal schema without having to change the conceptual schema.
  - For example, the internal schema may be changed when certain file structures are reorganized or new indexes are created to improve database performance
- When a schema at a lower level is changed, only the **mappings** between this schema and higher-level schemas need to be changed in a DBMS that fully supports data independence.
- The higher-level schemas themselves are **unchanged**.
  - Hence, the application programs need not be changed since they refer to the external schemas.

Pearson

41



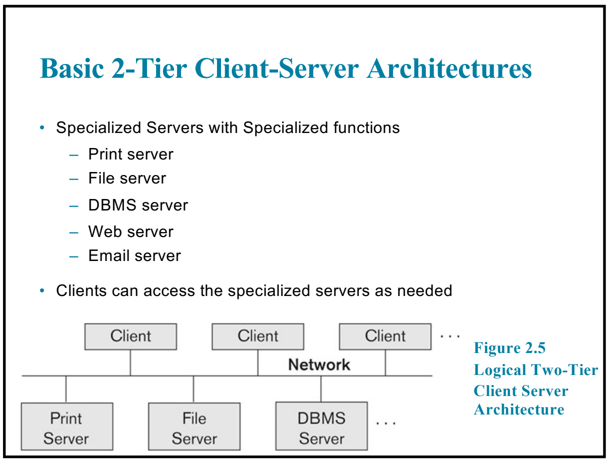
## Centralized and Client-Server DBMS Architectures

- Centralized DBMS:
  - Combines everything into single system including- DBMS software, hardware, application programs, and user interface processing software.
  - User can still connect through a remote terminal – however, all processing is done at centralized site.

Figure 2.4 A Physical Centralized Architecture

Pearson

55

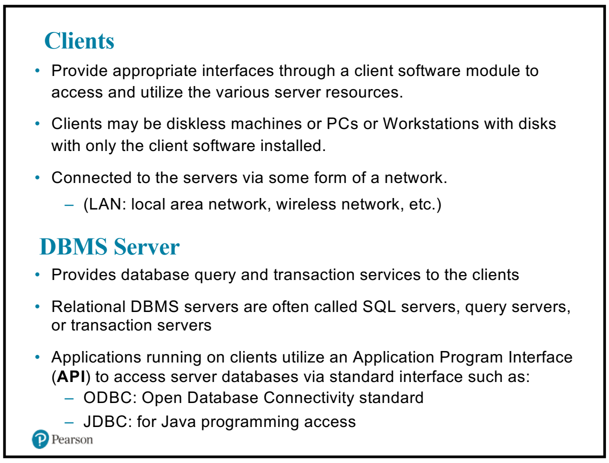


## Basic 2-Tier Client-Server Architectures

- Specialized Servers with Specialized functions
  - Print server
  - File server
  - DBMS server
  - Web server
  - Email server
- Clients can access the specialized servers as needed

Figure 2.5 Logical Two-Tier Client Server Architecture

57



## Clients

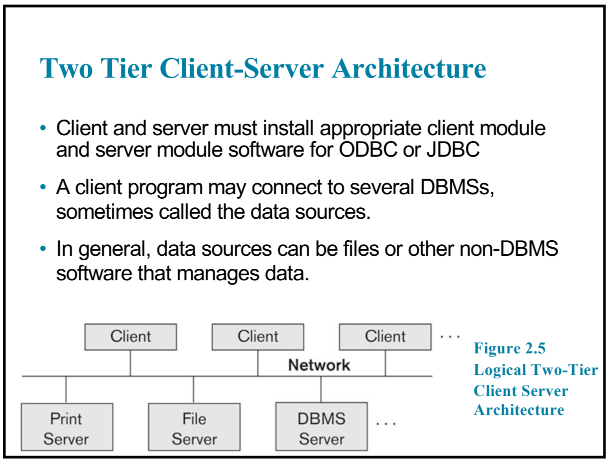
- Provide appropriate interfaces through a client software module to access and utilize the various server resources.
- Clients may be diskless machines or PCs or Workstations with disks with only the client software installed.
- Connected to the servers via some form of a network.
  - (LAN: local area network, wireless network, etc.)

## DBMS Server

- Provides database query and transaction services to the clients
- Relational DBMS servers are often called SQL servers, query servers, or transaction servers
- Applications running on clients utilize an Application Program Interface (**API**) to access server databases via standard interface such as:
  - ODBC: Open Database Connectivity standard
  - JDBC: for Java programming access

Pearson

59

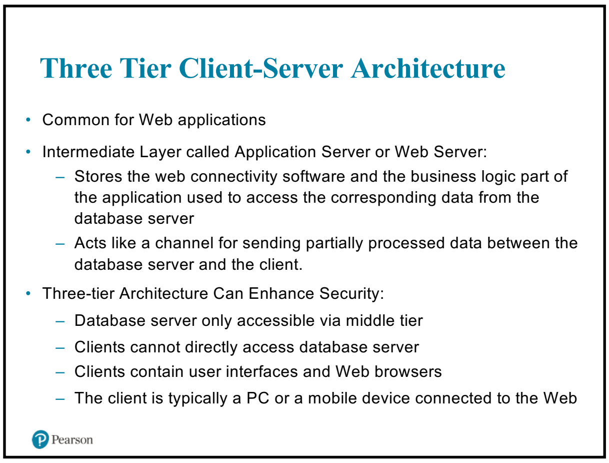


## Two Tier Client-Server Architecture

- Client and server must install appropriate client module and server module software for ODBC or JDBC
- A client program may connect to several DBMSs, sometimes called the data sources.
- In general, data sources can be files or other non-DBMS software that manages data.

Figure 2.5 Logical Two-Tier Client Server Architecture

61



## Three Tier Client-Server Architecture

- Common for Web applications
- Intermediate Layer called Application Server or Web Server:
  - Stores the web connectivity software and the business logic part of the application used to access the corresponding data from the database server
  - Acts like a channel for sending partially processed data between the database server and the client.
- Three-tier Architecture Can Enhance Security:
  - Database server only accessible via middle tier
  - Clients cannot directly access database server
  - Clients contain user interfaces and Web browsers
  - The client is typically a PC or a mobile device connected to the Web

Pearson

62

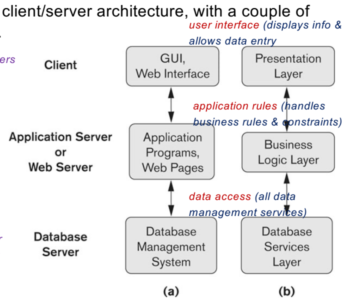
## Three-Tier Client-Server Architecture

**Figure 2.7** Logical three-tier client/server architecture, with a couple of commonly used terminology.

*Clients contain user interfaces and Web browsers*
*The client is typically a PC or a mobile device connected to the Web*

*Stores the web connectivity software and the business logic part of the application used to access the corresponding data from the database server*

*Database server only accessible via middle tier*
*Clients cannot directly access database server*



*user interface (displays info & allows data entry*

*application rules (handles business rules & constraints)*

*data access (all data management services)*

63

## Classification of DBMSs

- Based on the data model used
  - Legacy: Network, Hierarchical.
  - Currently Used: Relational, Object-oriented, Object-relational
  - Recent Technologies: Key-value storage systems, NOSQL systems: document based, column-based, graph-based and key-value based. Native XML DBMSs.
- Other classifications
  - Single-user (typically used with personal computers) vs multi-user (most DBMSs).
  - Centralized (uses a single computer with one database) vs distributed (multiple computers, multiple DBs)

64

## History of Data Models (Additional Material)

- Network Model
- Hierarchical Model
- Relational Model
- Object-oriented Data Models
- Object-Relational Models

68

## History of Data Models

- **Relational Model:**
  - Proposed in 1970 by E.F. Codd (IBM), first commercial system in 1981-82.
  - Now in several commercial products (e.g. DB2, ORACLE, MS SQL Server, SYBASE, INFORMIX).
  - Several free open source implementations, e.g. MySQL, PostgreSQL
  - Currently most dominant for developing database applications.
  - SQL relational standards: SQL-89 (SQL1), SQL-92 (SQL 2), SQL-99, SQL3, …
- Current trend by Relational DBMS vendors is to extend relational DBMSs with capability to process XML, Text and other data types.

74

## Relational Database Model

- A collection of tables in which data are stored
- Tables consist of rows and columns
- Tables related to each other by sharing a common characteristic

75

## Relational Database Model (Con't.)

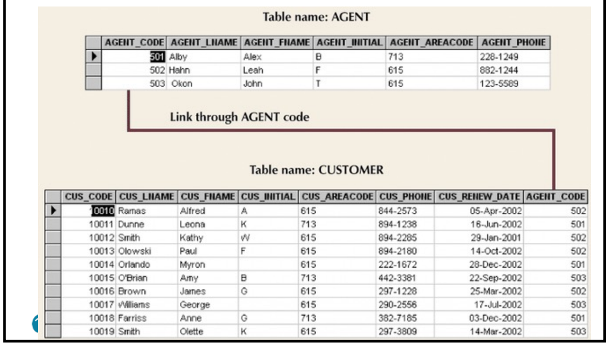


Table name: AGENT

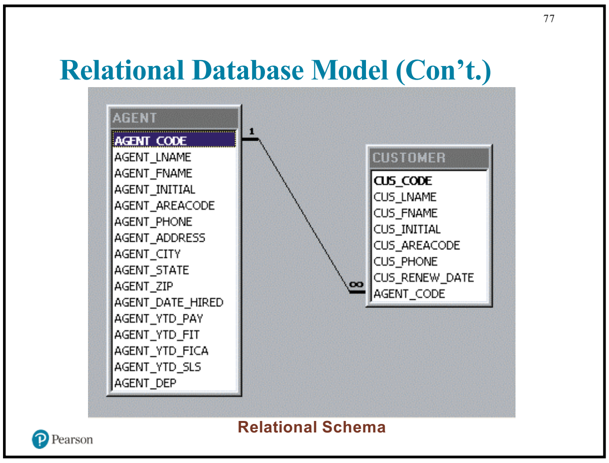
| AGENT_CODE | AGENT_LNAME | AGENT_FNAME | AGENT_INITIAL | AGENT_AREACODE | AGENT_PHONE |
|---|---|---|---|---|---|
| 501 | Alby | Alex | B | 713 | 228-1249 |
| 502 | Hahn | Leah | F | 615 | 882-1244 |
| 503 | Okon | John | T | 615 | 123-5589 |

Link through AGENT code

Table name: CUSTOMER

| CUS_CODE | CUS_LNAME | CUS_FNAME | CUS_INITIAL | CUS_AREACODE | CUS_PHONE | CUS_RENEW_DATE | AGENT_CODE |
|---|---|---|---|---|---|---|---|
| 10010 | Ramas | Alfred | A | 615 | 844-2573 | 05-Apr-2002 | 502 |
| 10011 | Dunne | Leona | K | 713 | 894-1238 | 16-Jun-2002 | 501 |
| 10012 | Smith | Kathy | W | 615 | 894-2285 | 29-Jan-2001 | 502 |
| 10013 | Olowski | Paul | F | 615 | 894-2180 | 14-Oct-2002 | 502 |
| 10014 | Orlando | Myron |  | 615 | 222-1672 | 28-Dec-2002 | 501 |
| 10015 | O'Brian | Amy | B | 713 | 442-3381 | 22-Sep-2002 | 503 |
| 10016 | Brown | James | G | 615 | 297-1228 | 25-Mar-2002 | 502 |
| 10017 | Williams | George |  | 615 | 290-2556 | 17-Jul-2002 | 503 |
| 10018 | Farriss | Anne | G | 713 | 382-7185 | 03-Dec-2002 | 501 |
| 10019 | Smith | Olette | K | 615 | 297-3809 | 14-Mar-2002 | 503 |

76



## Relational Database Model (Con’t.)

Relational Schema

Pearson