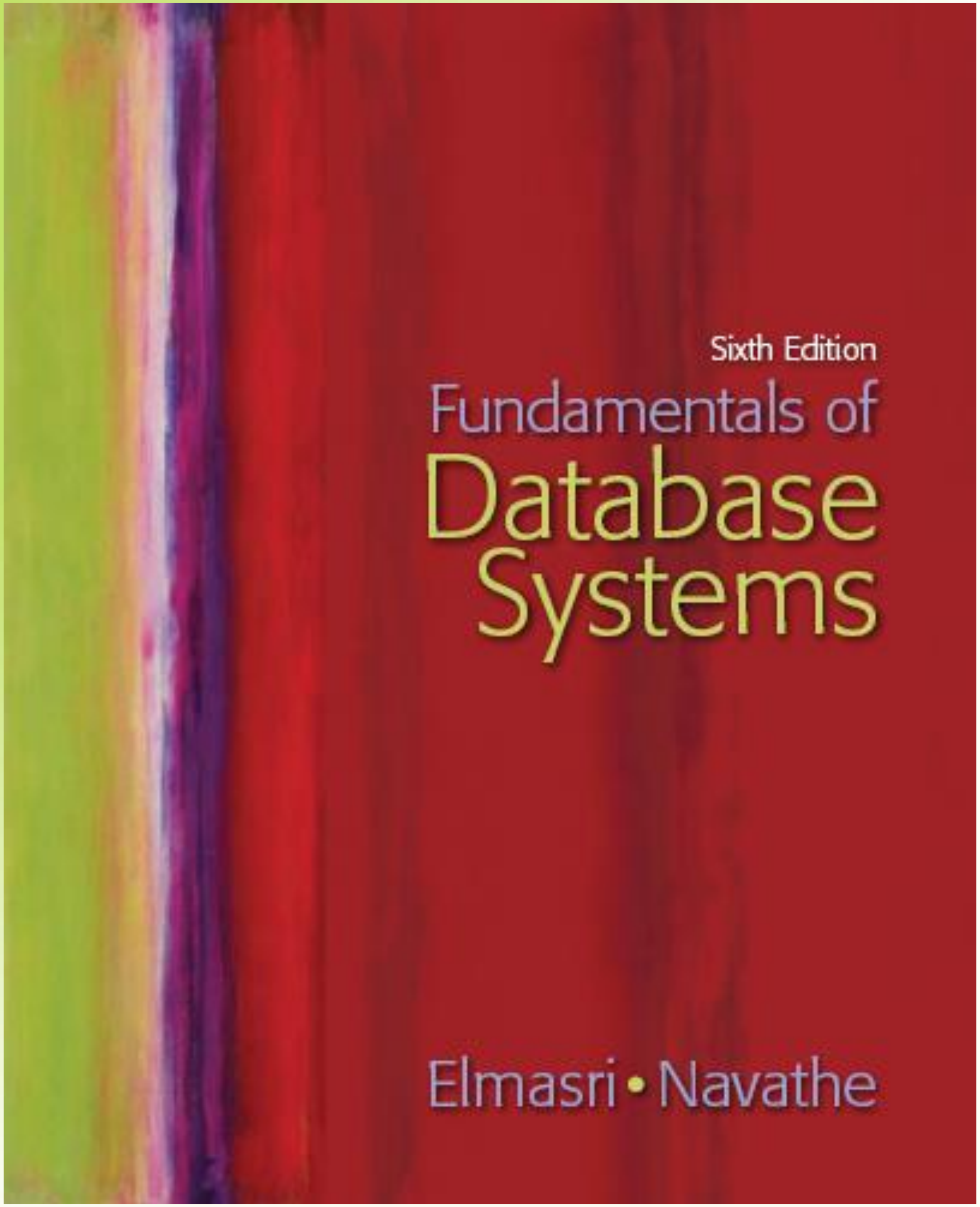
Chapter 1

Databases and Database Users



Addison-Wesley
is an imprint of
PEARSON

Chapter 1 Outline

- Types of Databases and Database Applications
- Basic Definitions
- Typical DBMS Functionality
- Example of a Database (UNIVERSITY)
- Main Characteristics of the Database Approach
- Types of Database Users
- Advantages of Using the Database Approach
- When Not to Use Databases

Addison-Wesley
is an imprint of
PEARSON

Types of Database Applications

- Traditional Applications:
  - Numeric and Textual Databases in Business Applications
- More Recent Applications:
  - Multimedia Databases (images, videos, voice, etc.)
  - Geographic Information Systems (GIS)
  - Data Warehouses (OLAP)
  - Real-time and Active Databases
  - Many other applications
- Initial part of book focuses on traditional applications
- *A number of recent applications are described later in the book (for example, Chapters 26,27,28,29)*

Addison-Wesley
is an imprint of
PEARSON

Basic Definitions

- **Database:**
  - A collection of related data.
- **Data:**
  - Known facts that can be recorded and have an implicit meaning.
- **Mini-world or Universe of Discourse (UoD):**
  - Some part of the real world about which data is stored in a database. For example, student registration, grades and transcripts at a university.
- **Database Management System (DBMS):**
  - A software package/system to facilitate the creation and maintenance of a computerized database.
- **Database System:**
  - The DBMS software together with the data itself. Sometimes, the application programs and interfaces are also included.

Simplified database system environment (Figure 1.1)

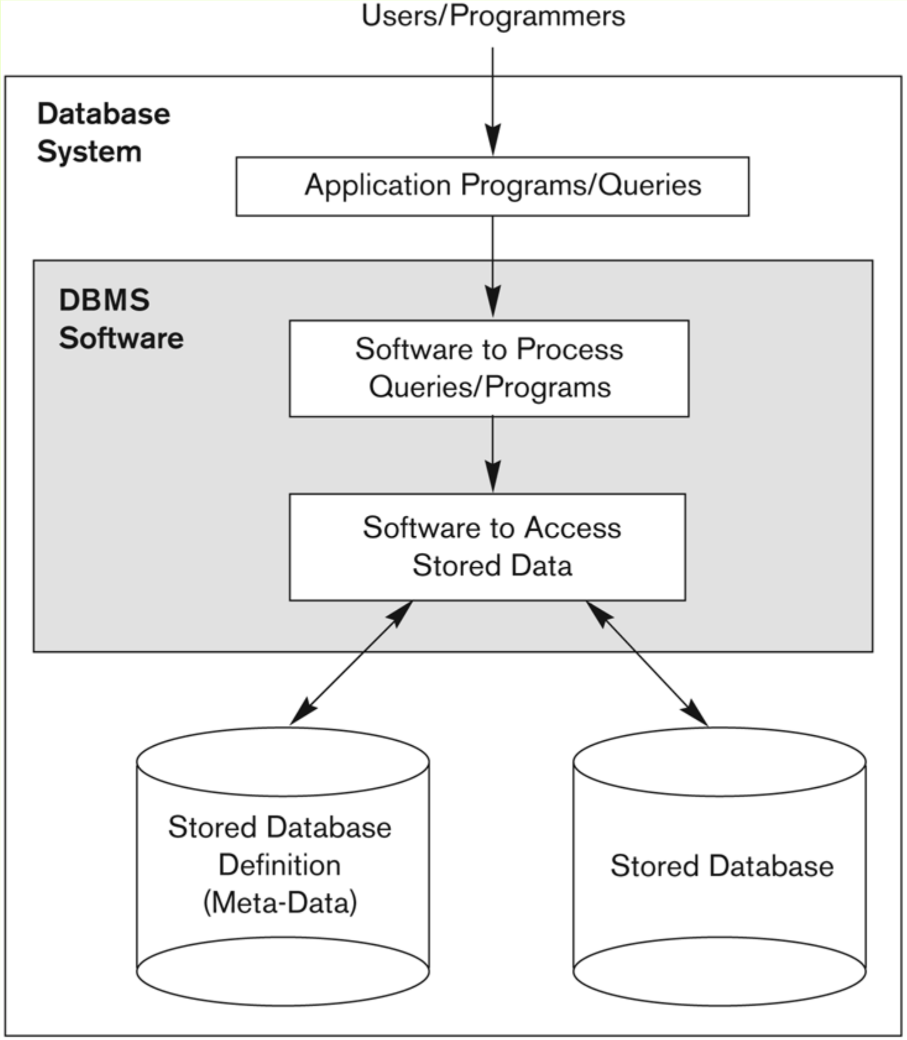

Users/Programmers
Database System
Application Programs/Queries
DBMS Software
Software to Process Queries/Programs
Software to Access Stored Data
Stored Database Definition (Meta-Data)
Stored Database

Figure 1.1
A simplified database system environment.

Addison-Wesley
is an imprint of
PEARSON

Typical DBMS Functionality

- *Define* a particular database in terms of its data types, structures, and constraints
- *Construct* or Load the initial database contents on a secondary storage medium (typically hard disk)
- *Manipulating* the database:
  - Retrieval: Querying, generating reports
  - Modification: Insertions, deletions and updates to its content
  - Accessing/changing the database through Web applications
- *Processing* and *Sharing* by a set of concurrent users and application programs – yet, keeping all data valid and consistent

Typical DBMS Functionality (cont.)

- Other features:
  - Protection or Security measures to prevent unauthorized access
  - “Active” processing to take internal actions on data
  - Presentation and Visualization of data
  - Maintaining the database and associated programs over the lifetime of the database application
    - Called database, software, and system life-cycle maintenance

Example of a Database UNIVERSITY Application

- **Mini-world for the example:**
  - Part of a UNIVERSITY environment.
- **Some mini-world *entities*:**
  - STUDENTs
  - COURSEs
  - SECTIONs (of COURSEs)
  - (academic) DEPARTMENTs
  - INSTRUCTORs

Addison-Wesley is an imprint of PEARSON

Example of a Database UNIVERSITY Application (cont.)

- **Some mini-world *relationships*:**
  - SECTIONs *are of specific* COURSEs
  - STUDENTs *take* SECTIONs
  - COURSEs *have prerequisite* COURSEs
  - INSTRUCTORs *teach* SECTIONs
  - COURSEs *are offered by* DEPARTMENTs
  - STUDENTs *major in* DEPARTMENTs

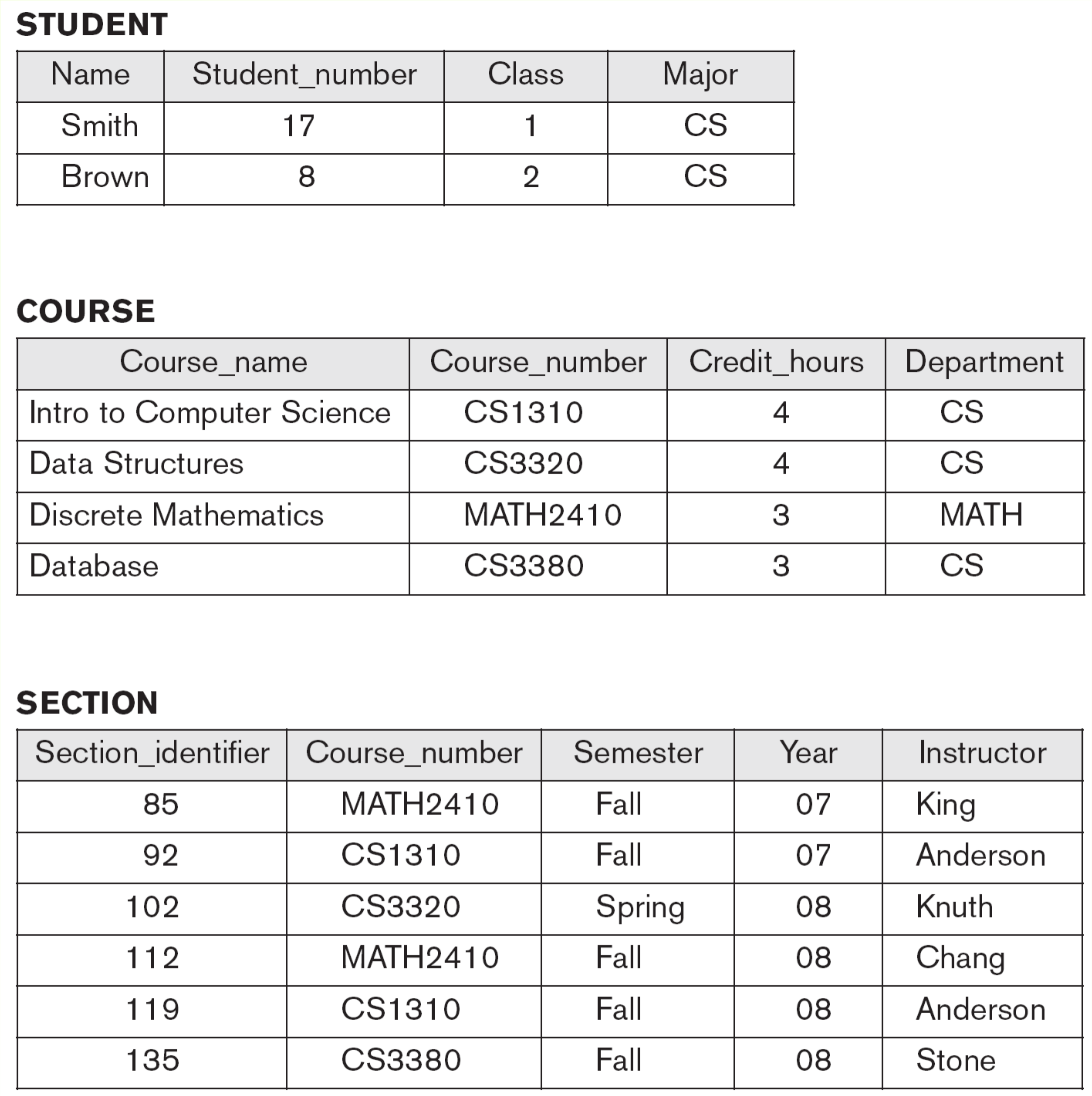
- Note: The above entities and relationships are typically expressed in a conceptual data model, such as the ENTITY-RELATIONSHIP data model (see Chapters 7, 8)

Addison-Wesley
is an imprint of
PEARSON

Example of a Database UNIVERSITY Application (cont.)

- The next two slides (Figure 1.2 from textbook) show what a simple UNIVERSITY database may look like

Addison-Wesley
is an imprint of
PEARSON



STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

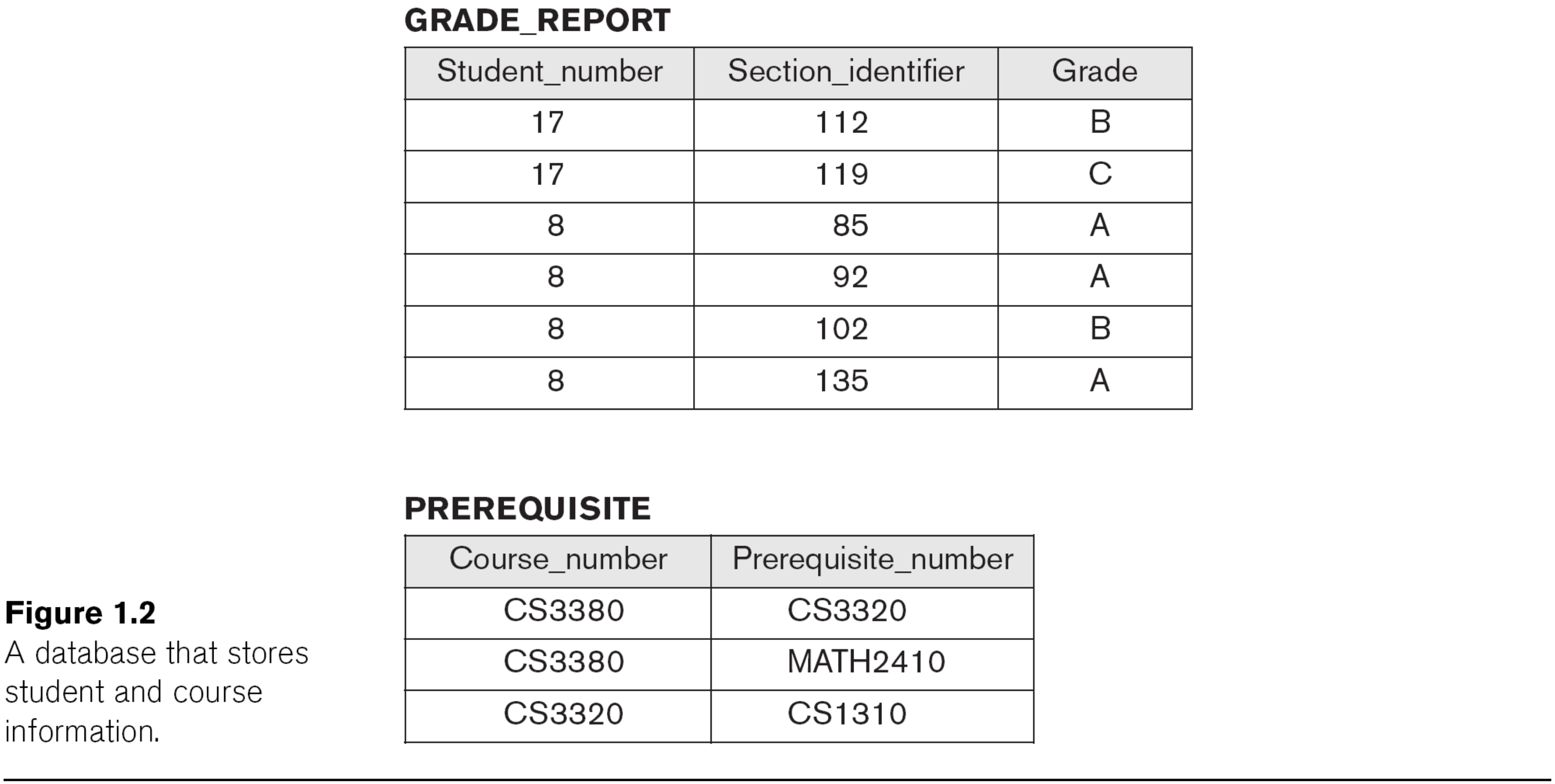
Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

Continued next page…

Addison-Wesley
is an imprint of
PEARSON



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

PREREQUISITE

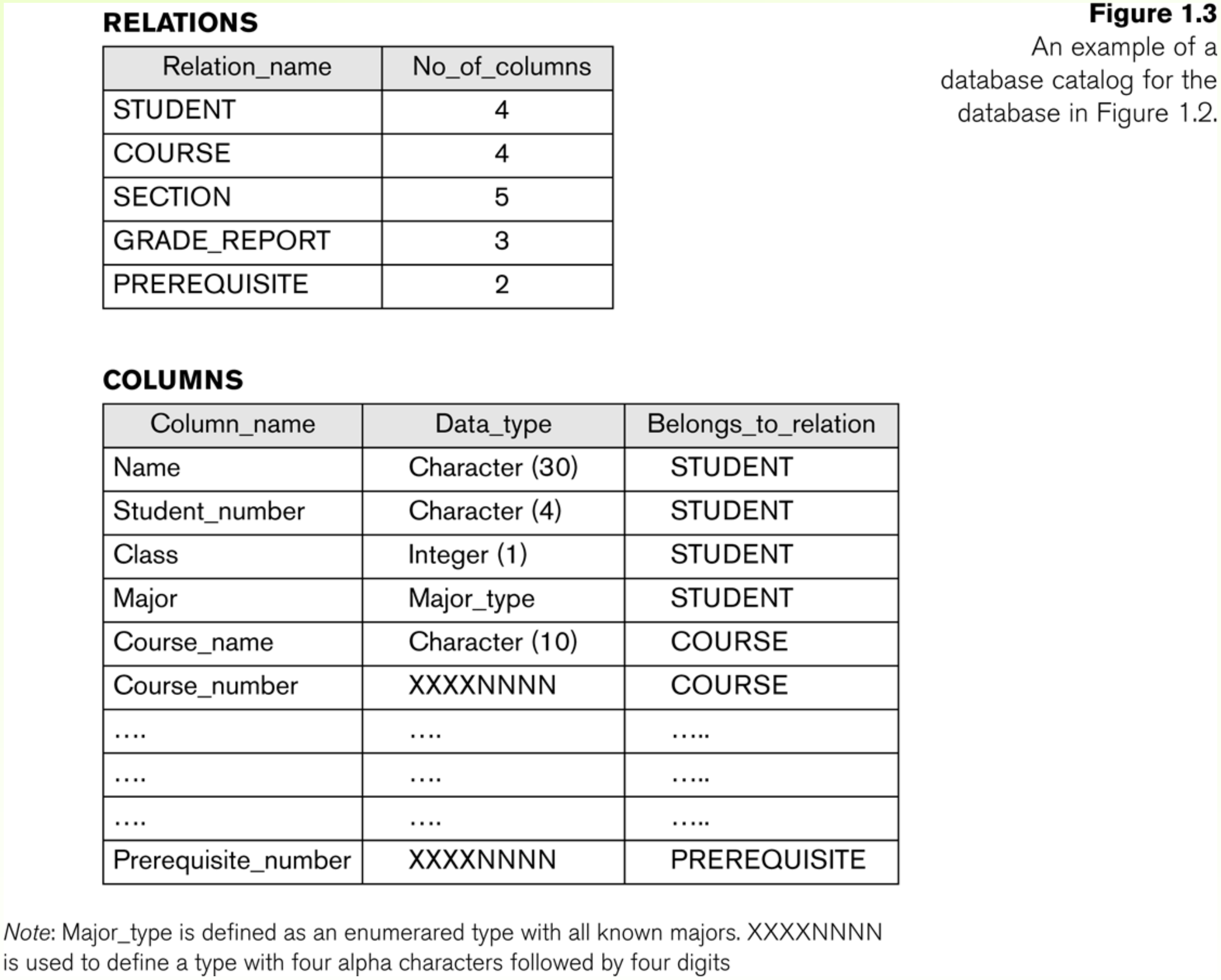
Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Figure 1.2
A database that stores student and course information.

Main Characteristics of the Database Approach

- **Self-describing nature of a database system:**
  - A DBMS **catalog** stores the *description* of a particular database (e.g. data structures, types, and constraints)
  - The description is called **meta-data** (see next slide).
  - This allows the DBMS software to work with *different* database applications (university, bank, airlines, etc.)
- **Insulation between programs and data:**
  - Called **program-data independence**.
  - Allows changing data structures and data storage organization without having to change the DBMS access programs.

Example of meta-date in a simplified database catalog (Figure 1.3)



RELATIONS

Relation_name	No_of_columns
STUDENT	4
COURSE	4
SECTION	5
GRADE_REPORT	3
PREREQUISITE	2

COLUMNS

Column_name	Data_type	Belongs_to_relation
Name	Character (30)	STUDENT
Student_number	Character (4)	STUDENT
Class	Integer (1)	STUDENT
Major	Major_type	STUDENT
Course_name	Character (10)	COURSE
Course_number	XXXXNNNN	COURSE
….	….	…..
….	….	…..
….	….	…..
Prerequisite_number	XXXXNNNN	PREREQUISITE

Figure 1.3
An example of a database catalog for the database in Figure 1.2.

Note: Major_type is defined as an enumerared type with all known majors. XXXXNNNN is used to define a type with four alpha characters followed by four digits

Addison-Wesley
is an imprint of
PEARSON

Main Characteristics of the Database Approach (cont.)

- **Insulation between programs and data (cont.):**
  - Accomplished through **data abstraction**
  - A **data model** is used to hide storage details and present the users with a conceptual view of the database.
  - Programs refer to the data model constructs rather than data storage details
- **Support of multiple views of the data:**
  - Each user may see a different view of the database, which describes **only** the data of interest to that user.

Main Characteristics of the Database Approach (cont.)

- **Sharing of data and multi-user transaction processing:**
  - Allowing a set of **user transactions** to access and update the database ***concurrently*** (at the same time).
  - *Concurrency control* within the DBMS guarantees that each **transaction** is correctly executed or aborted
  - *Recovery* subsystem ensures each completed transaction has its effect permanently recorded in the database
  - **OLTP** (Online Transaction Processing) is a major part of database applications (allows hundreds of concurrent transactions to execute per second)

Addison-Wesley
is an imprint of
PEARSON

Types of Database Users

- Users may be divided into
  - Those who actually use and control the database content, and those who design, develop and maintain database applications (called "Actors on the Scene"), and
  - Those who design and develop the DBMS software and related tools, and the computer systems operators (called "Workers Behind the Scene").

Database Users: End-users

- Actors on the scene
  - **End-users:** Use the database for queries, reports, and updating the database content. Can be categorized into:
    - **Casual end-users**: access database occasionally when needed
    - **Naïve** (or Parametric) **end-users**: largest section of end-user population.
      - Use previously implemented and tested programs (called "canned transactions") to access/update the database.
      - Examples are bank-tellers or hotel reservation clerks or sales clerks.

Addison-Wesley
is an imprint of
PEARSON

Database Users: End-users (cont)

- **Sophisticated end-users:**
  - These include business analysts, scientists, engineers, others thoroughly familiar with the system capabilities.
  - Many use tools in the form of software packages that work closely with the stored database.
- **Stand-alone end-users:**
  - Mostly maintain personal databases using ready-to-use packaged applications.
  - An example is a tax program user that creates its own internal database.
  - Another example is a user that maintains an address book

Addison-Wesley
is an imprint of
PEARSON

Database Users: DB Administrators (DBAs) and Designers

- Actors on the scene (continued)
  - **Database administrators:**
    - Responsible for authorizing/controlling access to the database; coordinating and monitoring its use; acquiring software and hardware resources; and monitoring efficiency of operations.
  - **Database Designers:**
    - Responsible for defining database structure, constraints, and transactions; communicate with users to understand their needs.

Workers behind the scene

- **DBMS system designers and implementers:**
  - Design and implement DBMS modules, modules for catalog, query language, accessing, concurrency control, recovery and security.
- **Tool developers:**
  - Tools for database modeling and design.
- **Operators and maintenance personnel**

Addison-Wesley
is an imprint of
PEARSON

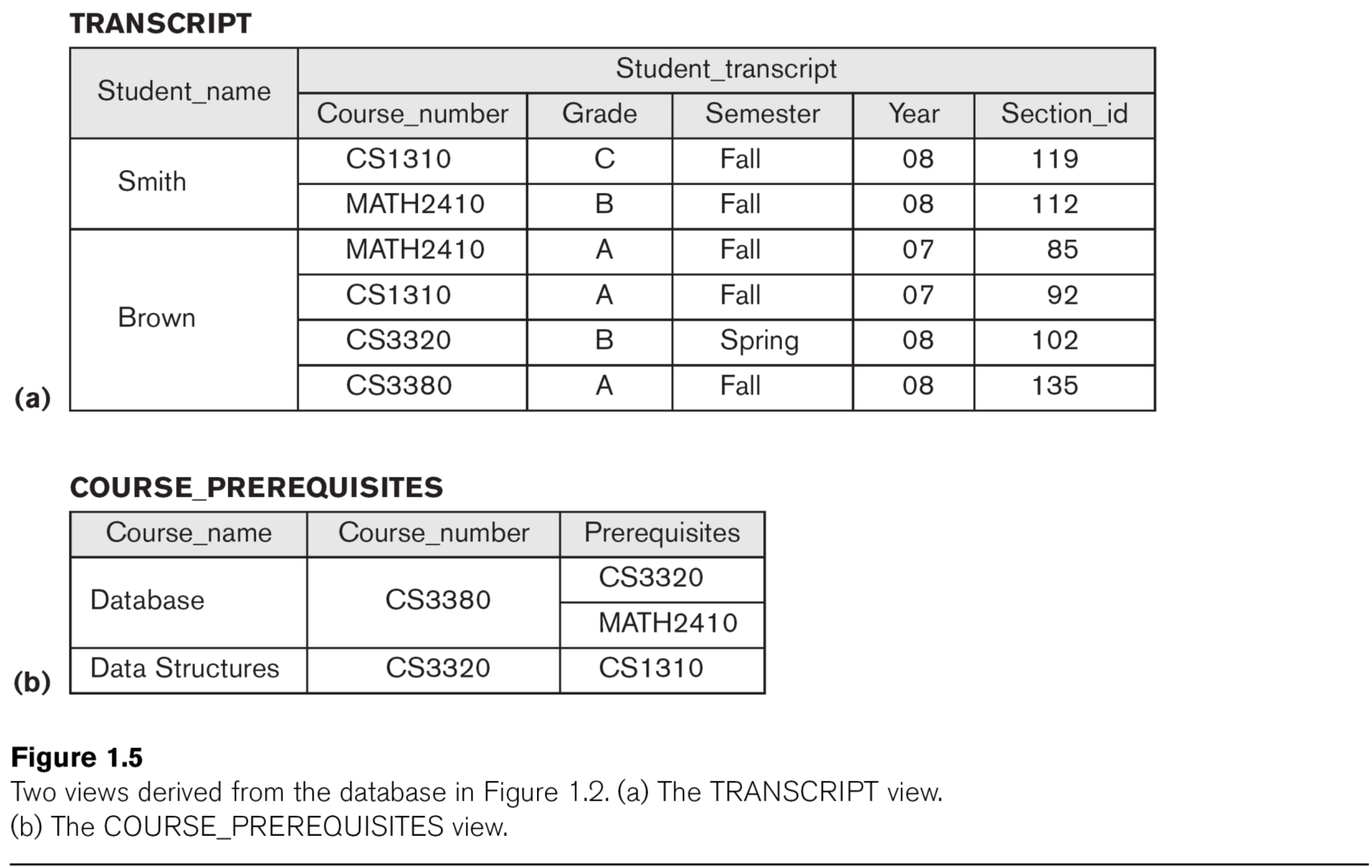
Advantages of Using the Database Approach

- Controlling redundancy in data storage and in development and maintenance efforts.
  - Sharing of data among multiple users.
- Restricting unauthorized access to data.
- Providing persistent storage for program Objects
  - In Object-oriented DBMSs – see Chapter 11
- Providing Storage Structures (e.g. indexes) for efficient Query Processing

Advantages of Using the Database Approach (cont.)

- Providing backup and recovery services.
- Providing multiple interfaces to different classes of users.
- Representing complex relationships among data.
- Enforcing integrity constraints on the database.
- Drawing inferences and actions from the stored data using deductive and active rules
- Allowing multiple “views” of the same data (see next slide, Figure 1.5 from textbook)

Addison-Wesley
is an imprint of
PEARSON



TRANSCRIPT

Student_name	Student_transcript				
	Course_number	Grade	Semester	Year	Section_id
Smith	CS1310	C	Fall	08	119
	MATH2410	B	Fall	08	112
Brown	MATH2410	A	Fall	07	85
	CS1310	A	Fall	07	92
	CS3320	B	Spring	08	102
	CS3380	A	Fall	08	135

(a)

COURSE_PREREQUISITES

Course_name	Course_number	Prerequisites
Database	CS3380	CS3320
		MATH2410
Data Structures	CS3320	CS1310

(b)

Figure 1.5
Two views derived from the database in Figure 1.2. (a) The TRANSCRIPT view.
(b) The COURSE_PREREQUISITES view.

Additional Implications of Using the Database Approach

- Potential for enforcing standards:
  - Crucial for the success of database applications in large organizations. **Standards** refer to data item names, display formats, screens, report structures, meta-data (description of data), Web page layouts, etc.
- Reduced application development time:
  - The time needed to add each new application is reduced.

Additional Implications of Using the Database Approach (cont.)

- Flexibility to change data storage structures:
  - Storage structures may evolve to improve performance, or because of new requirements.
- Availability of up-to-date information:
  - Extremely important for on-line transaction systems such as airline, hotel, car reservations.
- Economies of scale:
  - Wasteful overlap of resources and personnel can be avoided by consolidating data and applications across departments.

Addison-Wesley
is an imprint of
PEARSON

Historical Development of Database Technology

- Early Database Applications:
  - The Hierarchical and Network Models were introduced in mid 1960s and dominated during the seventies.
  - Some worldwide database processing still occurs using these models; particularly, the hierarchical model.
- Relational Model based Systems:
  - Relational model was introduced in 1970, and heavily researched and experimented with at IBM Research and several universities.
  - Relational DBMS Products emerged in the early 1980s and now dominate the market.

Historical Development of Database Technology (cont.)

- Object-oriented and emerging applications:
  - Object Databases (ODBs) were introduced in late 1980s and early 1990s to cater to the need of complex data and applications, and the proliferation of object-oriented programming languages.
    - Their use has not taken off much.
  - Many relational DBMSs have incorporated object database concepts, leading to a new category called *object-relationa*l databases (ORDBs) (see Ch. 11)
  - *Extended relational* systems add further capabilities (e.g. for multimedia data, XML, spatial, and other data types)

Historical Development of Database Technology (cont.)

- Data on the Web and E-commerce Applications:
  - *Static* Web pages often specified in HTML (Hypertext markup language) with links among pages.
  - *Dynamic* Web pages have portions of their content extracted from databases, and allow user interaction with databases by typing in form boxes.
  - Script programming languages such as PHP and JavaScript allow generation of dynamic Web pages (see Ch. 14), and provide for user querying of databases by typing selection keywords (e.g. flight number or student id).
    - Also allow database updates through Web pages

Addison-Wesley
is an imprint of
PEARSON

Extending Database Capabilities

- New functionality is being added to DBMSs in the following areas:
    - Scientific Applications
    - XML (eXtensible Markup Language)
    - Image Storage and Management
    - Audio and Video Data Management
    - Data Warehousing and Data Mining
    - Spatial Data Management and Geographic Information Systems
    - Time Series and Historical Data Management
    - Collecting and fusing data from distributed sensors

- The above led to *new research and development* in incorporating new data types, complex data structures, new operations/query languages, and new storage and indexing schemes (see Chapter 26).

When not to use a DBMS

- Main inhibitors (costs) of using a DBMS:
  - High initial investment and possible need for additional hardware.
  - Overhead for providing generality, security, concurrency control, recovery, and other functions.
- When a DBMS may be unnecessary:
  - If the database and applications are simple, well defined, and not expected to change.
  - If there are stringent real-time requirements that may not be met because of DBMS overhead.
  - If access to data by multiple users is not required.

Addison-Wesley
is an imprint of
PEARSON

When not to use a DBMS (cont.)

- When no DBMS may suffice:
  - If the database system is not able to handle the complexity of data because of modeling limitations
  - If the database users need special operations not supported by the DBMS
  - When DBMS overhead makes it impossible to achieve the needed application performance

Addison-Wesley
is an imprint of
PEARSON

Chapter 1 Summary

- Types of Databases and Database Applications
- Basic Definitions
- Typical DBMS Functionality
- Example of a Database (UNIVERSITY)
- Main Characteristics of the Database Approach
- Different Types of Database Users
- Advantages of Using the Database Approach
- When Not to Use Databases

Addison-Wesley
is an imprint of
PEARSON