Ch 1. Databases and Database Users

- Database: a collection of data with the following properties

  (a) it represents some aspect of the real world;
  changes to the real world are reflected in the database

  (b) logically coherent collection of data (inter-related)

  (c) it is designed, built and populated with data for
  a specific purpose (intended users and applications)

  (d) typically large in size

- Database Management System (DBMS): collection of programs that enables users to create and maintain a database. Activities: **Defining** the database (structure), **Constructing** the database (populating), **Manipulating** the database (query, applications, updates)
  Figure 1.1
  Figure 1.2

Figure 1.1 A simplified database system environment, illustrating the concepts and terminology discussed in Section 1.1.

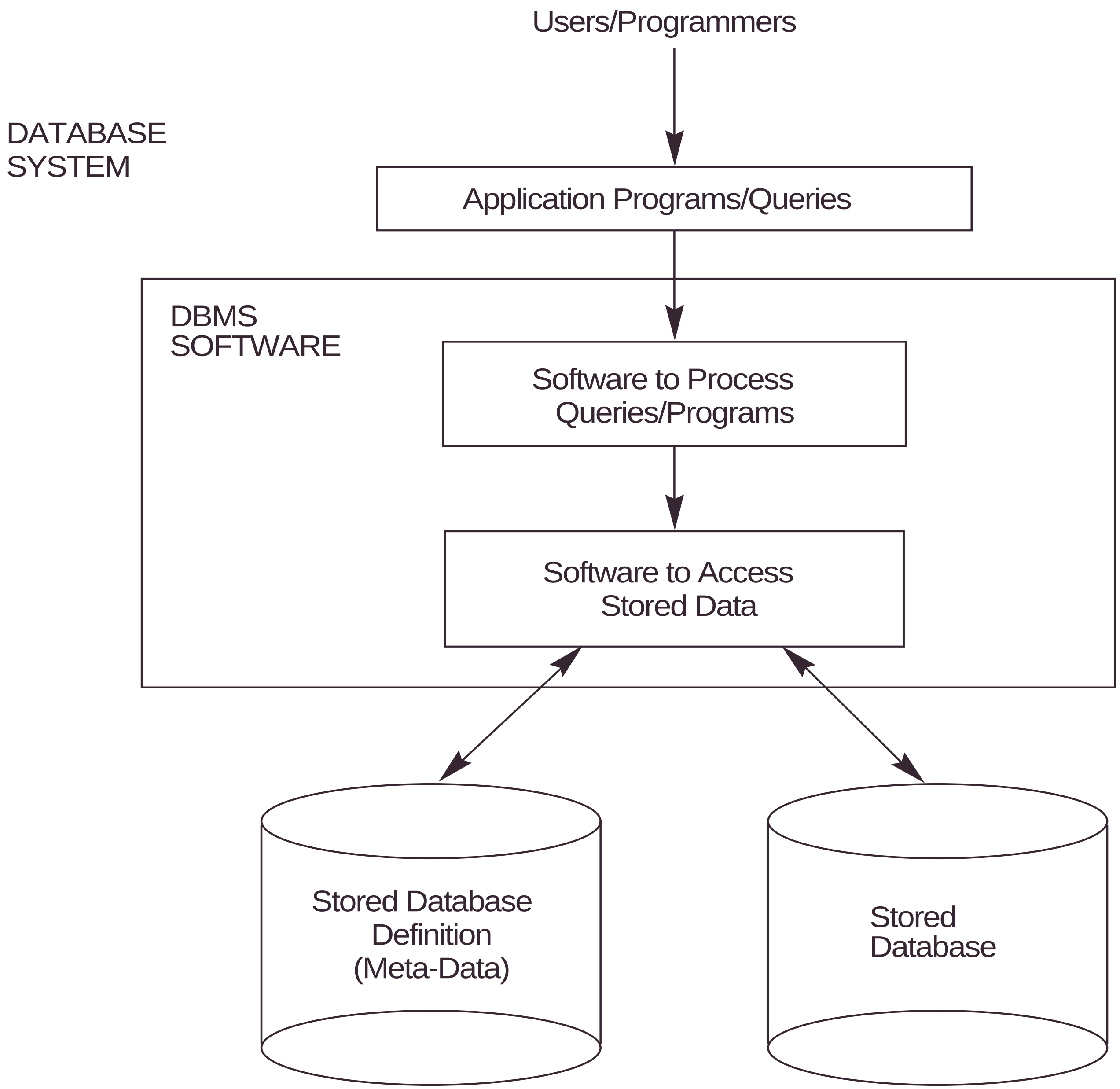


Figure 1.2 An example of a database that stores student records and their grades.

STUDENT	Name	StudentNumber	Class	Major
	Smith	17	1	CS
	Brown	8	2	CS

COURSE	CourseName	CourseNumber	CreditHours	Department
	Intro to Computer Science	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Database	CS3380	3	CS

SECTION	SectionIdentifier	CourseNumber	Semester	Year	Instructor
	85	MATH2410	Fall	98	King
	92	CS1310	Fall	98	Anderson
	102	CS3320	Spring	99	Knuth
	112	MATH2410	Fall	99	Chang
	119	CS1310	Fall	99	Anderson
	135	CS3380	Fall	99	Stone

GRADE_REPORT	StudentNumber	SectionIdentifier	Grade
	17	112	B
	17	119	C
	8	85	A
	8	92	A
	8	102	B
	8	135	A

PREREQUISITE	CourseNumber	PrerequisiteNumber
	CS3380	CS3320
	CS3380	MATH2410
	CS3320	CS1310

Database Approach vs. Traditional File Processing

- Self contained nature of database systems (database contains both data and meta-data).
- Data Independence: application programs and queries are independent of how data is actually stored. (Figure 1.3 add Birthdate)
- Data sharing.
- Controlling redundancies and inconsistencies.
- Secure access to database; Restricting unauthorized access.
- Enforcing Integrity Constraints.
- Backup and Recovery from system crashes.
- Support for multiple-users and concurrent access (data sharing).
- Support for multiple views of data (Figure 1.4)

Figure 1.3 Internal storage format for a STUDENT record.

Data Item Name	Starting Position in Record	Length in Characters (bytes)
Name	1	30
StudentNumber	31	4
Class	35	4
Major	39	4

Figure 1.4 Two views derived from the example database shown in Figure 1.2. (a) The student transcript view. (b) The course prerequisite view.

(a)

TRANSCRIPT	StudentName	Student Transcript				
		CourseNumber	Grade	Semester	Year	SectionId
	Smith	CS1310	C	Fall	99	119
		MATH2410	B	Fall	99	112
	Brown	MATH2410	A	Fall	98	85
		CS1310	A	Fall	98	92
		CS3320	B	Spring	99	102
		CS3380	A	Fall	99	135

(b)

PREREQUISITES	CourseName	CourseNumber	Prerequisites
	Database	CS3380	CS3320
			MATH2410
	Data Structures	CS3320	CS1310

Actors on the scene

- Database Administrator (DBA)
- Database Designers
- End Users:

```
Casual users  (managers using SQL)
Naive (or parametric end users)  (bank tellers/clerks)
Sophisticated end users (engineers, scientists, analysts)
Stand-alone users (personal db, single user, ex. tax package)
```

- System Analysts and Application Programmers

Workers behind the Scene

- DBMS designers and implementors
- Tool developers
- System administrators

Intended uses of a DBMS

```
-- Controlling redundancy (Figure 1.5)
-- Restricting unauthorized access
-- Persistent storage of program objects and data
-- Deductive inferences of data
-- Multiple user interfaces (SQL, C-SQL, Forms, Menus, Web)
-- Complex relationships among data (1-many, many-many, 1-1)
-- Enforcing integrity constraints
-- Backup and recovery
```

Figure 1.5 The redundant storage of data items. (a) *Controlled redundancy:* Including StudentName and CourseNumber in the grade_report file. (b) *Uncontrolled redundancy:* A GRADE_REPORT record that is inconsistent with the STUDENT records in Figure 1.2, because the Name of student number 17 is Smith, not Brown.

(a)

GRADE_REPORT	StudentNumber	StudentName	SectionIdentifier	CourseNumber	Grade
	17	Smith	112	MATH2410	B
	17	Smith	119	CS1310	C
	8	Brown	85	MATH2410	A
	8	Brown	92	CS1310	A
	8	Brown	102	CS3320	B
	8	Brown	135	CS3380	A

(b)

GRADE_REPORT	StudentNumber	StudentName	SectionIdentifier	CourseNumber	Grade
	17	Brown	112	MATH2410	B

Implications of Database approach

```
-- potential for enforcing standards (naming/formatting convention,
         display formats, report structures)

-- reduced application development time (once database is up and running)

-- flexibility (changing database structure)

-- Up-to-date information (e.g. airline reservations)

-- Economies of scale (invest in one server, centralized database)
```

Data Models, Schemas, Instances

```
Data Model: set of concepts used to describe the structure
   of a database (abstraction tool); It may also include
   a set of operations for specifying updates and queries

- High-level or Conceptual Data Model  (ER)

- Representational or Implementation Data Model
  (Record-based; Value-based/object-based)
  Relational, Object-Oriented, Deductive, Hierarchical, Network

- Low-level or Physical Data Model (access paths)
```

Database Schema vs Database Instance

Schemas (Intension) and Instances (Extension)

Schema: structure (meta-data; schema diagram; schema constructs)
Figure 2.1

Instances: actual data (occurrences or instances) database state

Figure 2.1 Schema diagram for the database of Figure 1.2.

STUDENT

Name	StudentNumber	Class	Major

COURSE

CourseName	CourseNumber	CreditHours	Department

PREREQUISITE

CourseNumber	PrerequisiteNumber

SECTION

SectionIdentifier	CourseNumber	Semester	Year	Instructor

GRADE_REPORT

StudentNumber	SectionIdentifier	Grade

2.2 DBMS Architecture and Data Independence

Figure 2.2

Internal Level (physical structure)
Conceptual Level (logical structure)
External Level (Views)

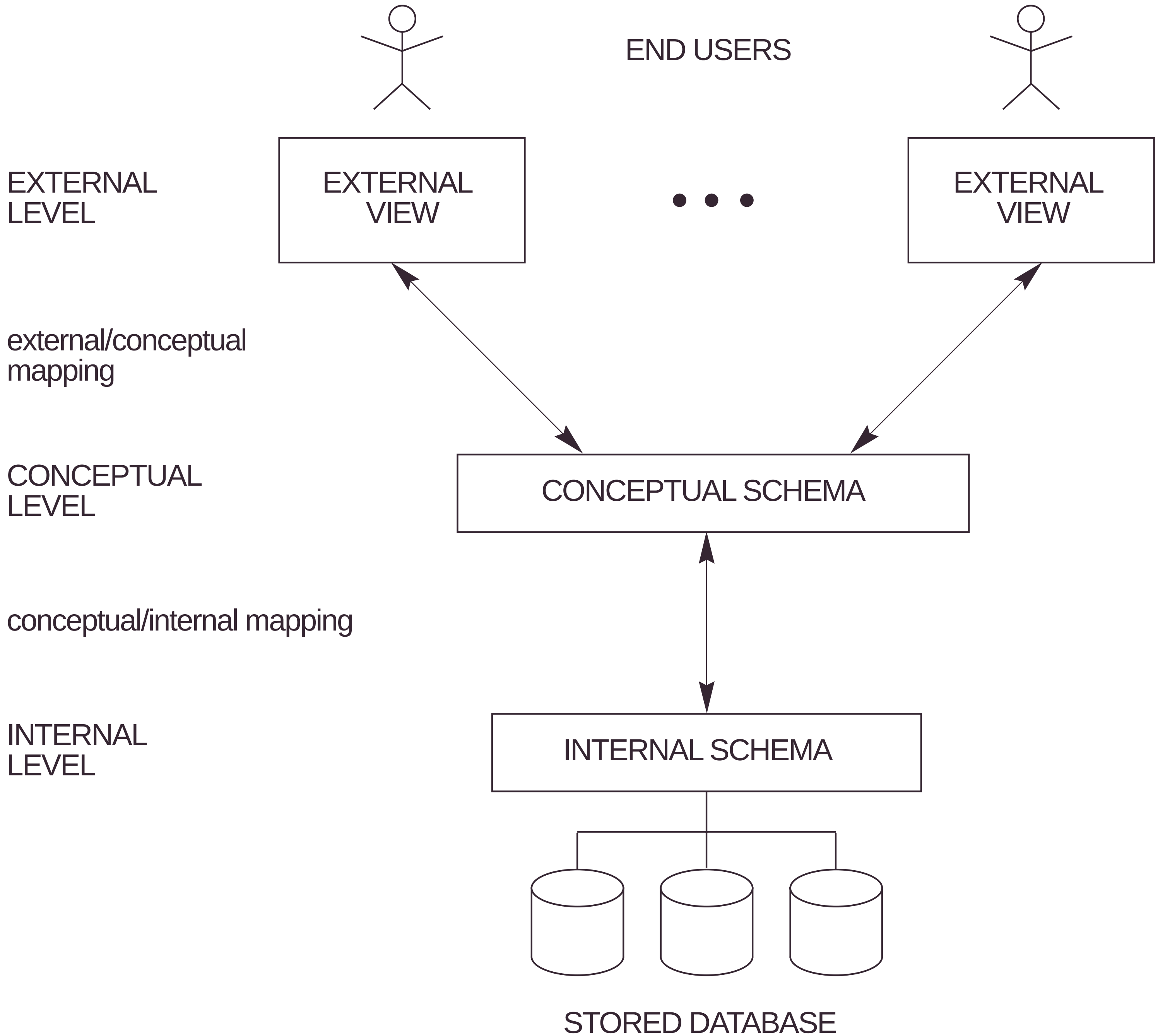
mappings: from one level to another

Data Independence:

Logical Data Independence: ability to change conceptual schema without having to change external views or application programs

Physical Data Independence: ability to change internal schema without having to change conceptual or external schema.

Figure 2.2 Illustrating the three-schema architecture.



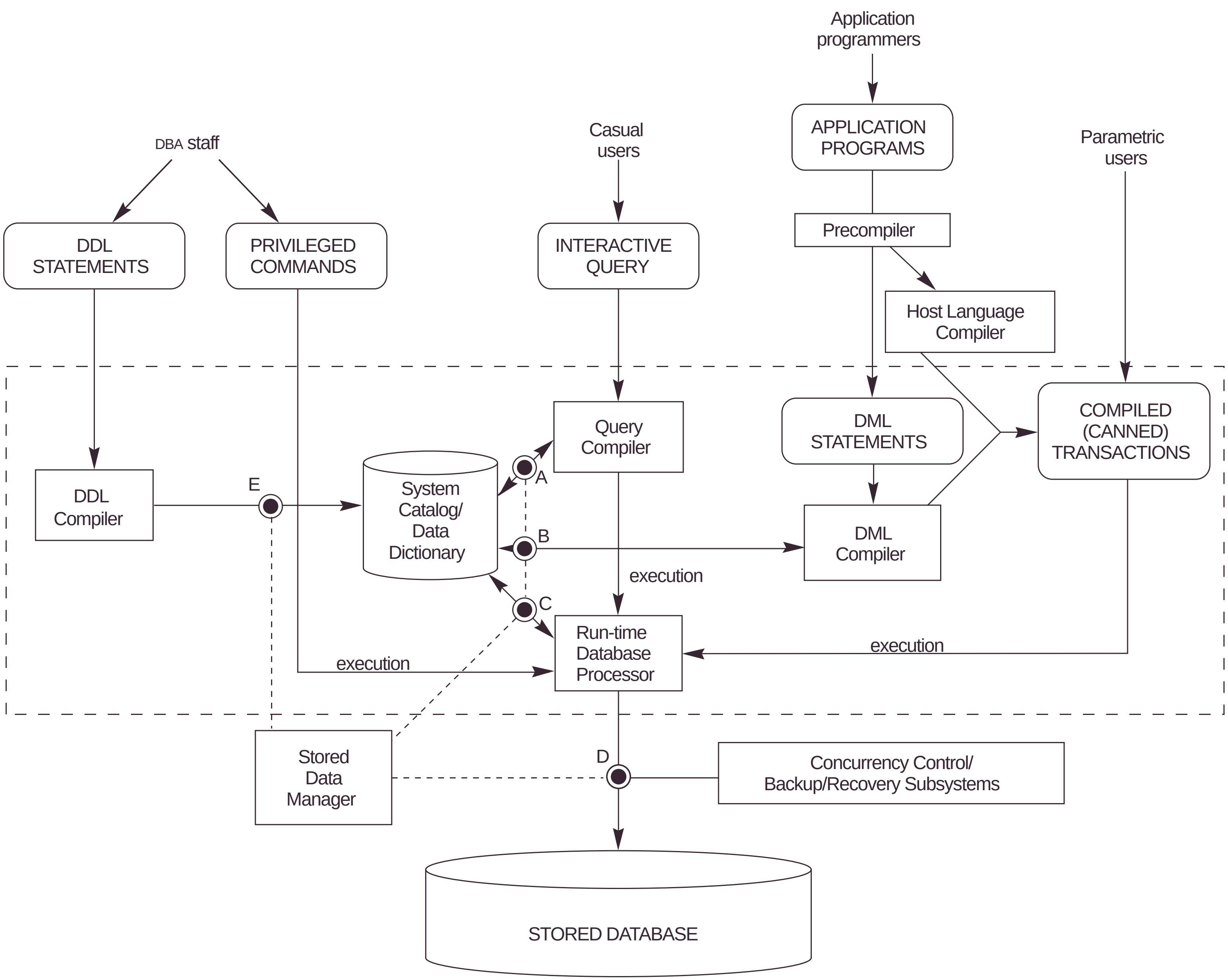
2.3 Database Languages and Interfaces

```
DBMS Languages: DDL (for conceptual level)
                  SDL (Storage) and VDL (Views) when three
                  levels are separated by system.
                DML (for queries and manipulation)
                   - Query Language (non-procedural/declarative)
                      set at a time
                   - procedural DML
                      record at a time
                      host language (embedding)
                User Interfaces (Menu-based,GUI,Forms-based,
                     Natural-language))
```

2.4 Database System Environment

Figure 2.3

Figure 2.3 Typical component modules of a DBMS. Dotted lines show accesses that are under the control of the stored data manager.



Classification of DBMS

```
-- Based on Data Model
    Relational, Network, Hierarchical, Object-Oriented, Others

-- Based on number of users
    Single-user, Multi-user

-- Based on number of sites
    Centralized, Distributed (Homogenous/Heterogenous)
```

Figure 2.4 The schema of Figure 2.1 in the notation of the network data model.