BASIC CONCEPTS

Chapter 1: Databases and Database Users

Outline of Chapter 1

1 Basic Definitions

2 Example of a Database

3 Main Characteristics of Database Technology

4 Databases and People

5 Additional Benefits of Database Technology

6 When Not to Use a DBMS

1. Basic definitions (1)

Database (DB): A collection of related data.

Data: Known facts that can be recorded and have an implicit meaning.

Mini-world: Some part of the real world about which data is stored in a database.
For example, student grades and transcripts at a university.

Database Management System (DBMS): A collection of programs to facilitate the creation and maintenance of a computerized database.

Database System: The DBMS software together with the data itself. Sometimes, the applications are also included.

1. Basic definitions (2)

A **DBMS is a general purpose software system** that facilitates the process of *defining*, *constructing*, and *manipulating* DBs for various applications.

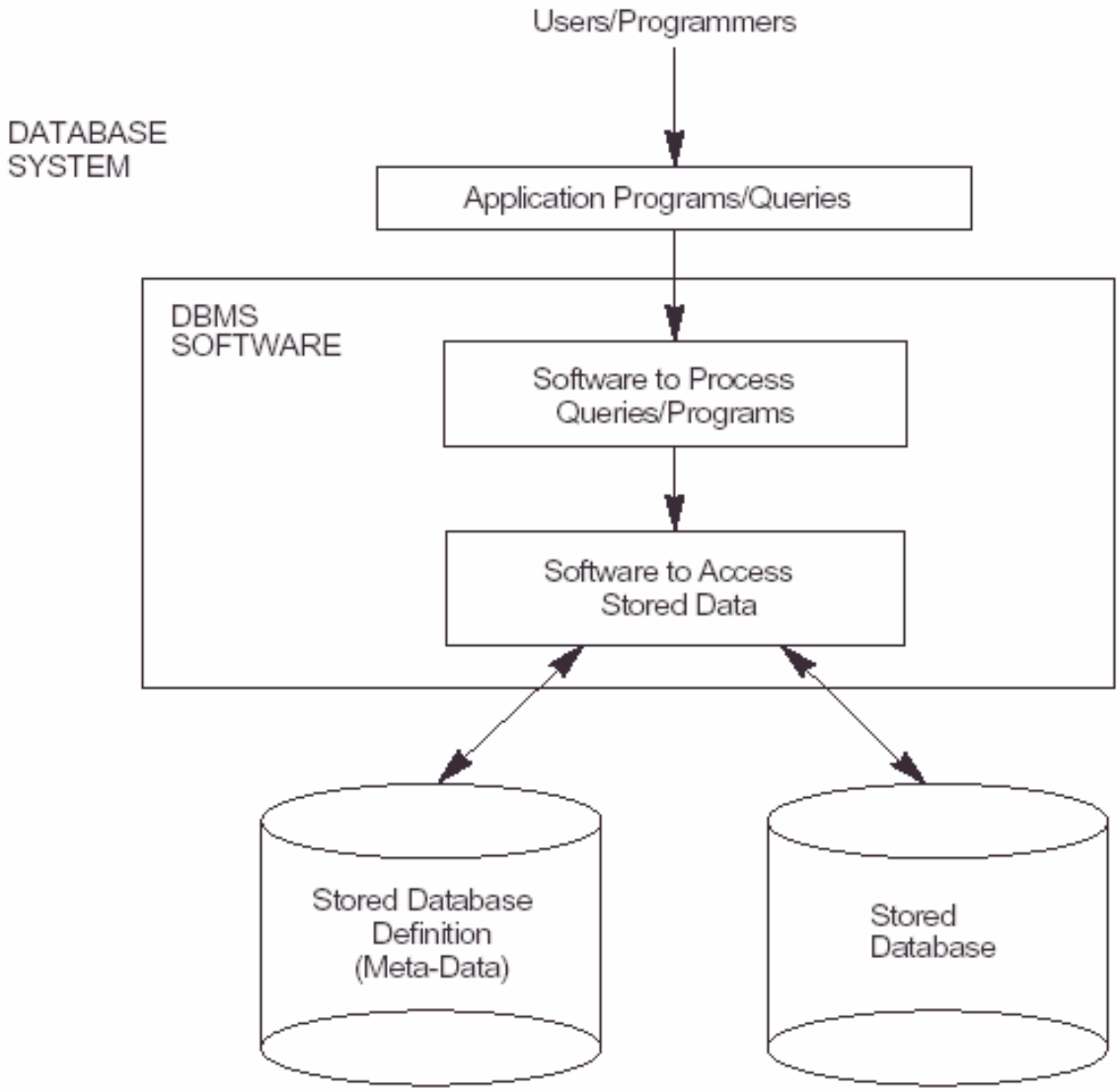
Defining a DB involves specifying the *data types, structures and constraints* for the data to be stored in the DB.

Constructing a DB is the process of storing the data itself on some storage medium that is controlled by the DBMS.

Manipulating a DB involves functions such as *querying* the DB to retrieve specific data, *updating the DB* to reflect changes in the mini-world, and *generating reports* from the data.

1. Basic definitions (3)

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



2. Example of a Database

(Using a Conceptual Data Model)

Mini-world for the example: Part of a UNIVERSITY environment.

Some mini-world *entities*:
- STUDENTs
- COURSEs
- SECTIONs (of COURSEs)
- (academic) DEPARTMENTs
- INSTRUCTORs

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

.

2. Example of a Database (Cont.)

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

3. Characteristics of the database approach (1)

- A number of characteristics distinguishes the *database approach* from the traditional *approach of programming with files*:

- Self-contained nature of a database system: A DBMS **catalog** stores the *description* of the database. (The description is called **meta-data**).

  This allows the DBMS software to work with different databases.

3. Characteristics of the database approach (2)

- Insulation between programs and data: Called **program-data independence**. Allows changing data storage structures and operations without having to change the DBMS access programs.

Example

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

3. Characteristics of the database approach (2 – Cont.)

- The characteristic that allows *program-data independence* is called **data abstraction**.

- A **data model** is a type of *data abstraction* that is used to provide the users with a *conceptual representation* of data that does not include many of the details of how data is stored.

3. Characteristics of the database approach (3)

- 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.

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

3. Characteristics of the database approach (4)

- A multiuser DBMS must allow *multiple users to access the database at the same time*.

- Concurrency control: The DBMS must include **concurrency control software** to ensure that multiple users trying to update the same data do so in a controlled manner so that the result of the update is correct.

4. Databases and People

- *Database Administrators*
- *Database Designers*
- *End Users*
  - Casual End Users
  - Naïve users
  - Sophisticated end users
  - Stand-alone users
- *System analysts and Application Programmers*

5. Additional Benefits of Database Technology

- Controlling redundancy in data storage and in development and maintenance efforts.

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

5. Additional Benefits of Database Technology (Cont.)

- Sharing of data among multiple users.
- Restricting unauthorized access to data.
- Providing multiple interfaces to different classes of users.
- Representing complex relationships among data.
- Enforcing integrity constraints on the database.
- Providing backup and recovery services.
- Potential for enforcing standards.
- Flexibility to change data structures.
- Reduced application development time.
- Availability of up-to-date information.
- Economies of scale.

6. When Not to Use a DBMS

In a few cases the use of a DBMS may incur unnecessary overhead costs due to the following:

- High initial investment and possible need for additional hardware.
- Overhead for providing generality, security, recovery, Integrity, and concurrency control.

In the following cases it may be preferable to use regular files:

- 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.

Sometimes no DBMS may suffice:

- If the database system is not able to handle the complexity of data because of modeling limitations