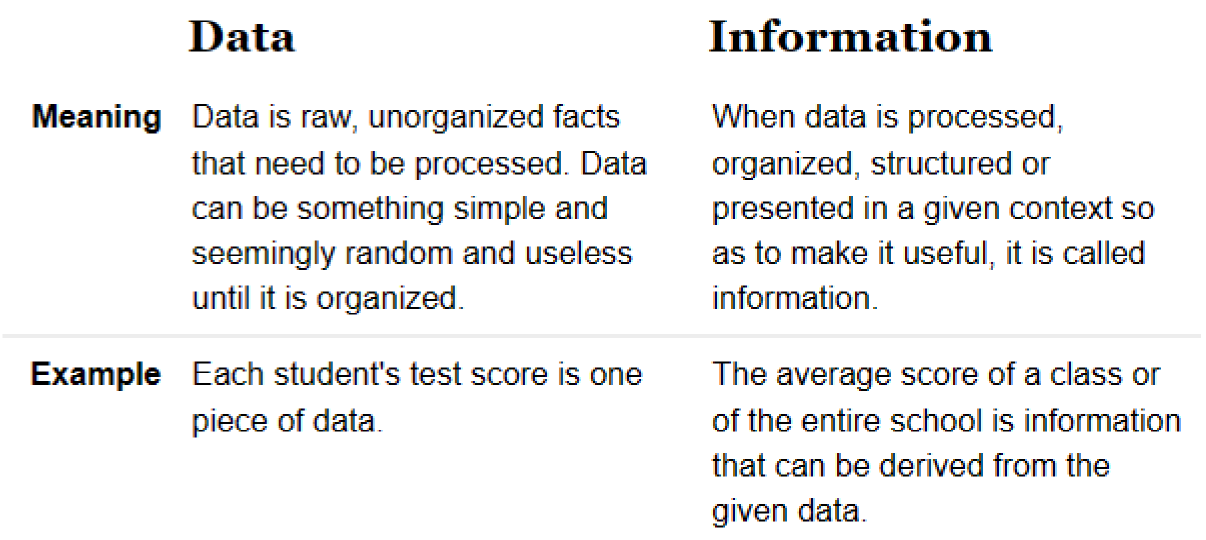
Chapter One: Introduction to Database System

Data are simply facts or figures (unorganized) — bits of information, but not information itself. When data are processed, interpreted, organized, structured or presented so as to make them meaningful or useful, they are called information. Information provides context for data. What the decision maker really needs is information, which is defined as data processed and presented in a meaningful form.



	Data	Information
Meaning	Data is raw, unorganized facts that need to be processed. Data can be something simple and seemingly random and useless until it is organized.	When data is processed, organized, structured or presented in a given context so as to make it useful, it is called information.
Example	Each student's test score is one piece of data.	The average score of a class or of the entire school is information that can be derived from the given data.

Databases and database technology are having a major impact on the growing use of computers. It is fair to say that databases play a critical role in almost all areas where computers are used, including business, engineering, medicine, law, education, and library science, to name a few.

A **database** is a collection of related data. By **data,** we mean known facts that can be recorded and that have implicit meaning. For example, consider the names, telephone numbers, and addresses of the people you know. You may have recorded this data in an indexed address book, or you may have stored it on a diskette, using a personal computer and software such as Microsoft ACCESS, or EXCEL. This is a collection of related data with an implicit meaning and hence is a database. A database has the following implicit properties:

- A database represents some aspect of the real world
- A database is a logically coherent collection of data with some inherent meaning
- A database is designed, built, and populated with data for a specific purpose

A database management system (DBMS) is a collection of programs that enables users to create and maintain a database. The DBMS is hence a general-purpose software system that facilitates the processes of ***defining, constructing,*** and ***manipulating*** databases for various applications. Defining a database involves specifying the data types, structures, and constraints for the data to be stored in the database. Constructing the database is the process of storing the data itself on some storage medium that is controlled by the DBMS. Manipulating a database includes such functions as querying the database to retrieve specific data, updating the database to reflect changes in the miniworld, and generating reports from the data.

Data management passes through the different levels of development along with the development in technology and services. These levels could best be described by categorizing the levels into three levels of development. Even though there is an advantage and a problem overcome at each new level, all methods of data handling are in use to some extent. The major three levels are;

1. **Manual Approach**
2. **Traditional File Based Approach**
3. **Database Approach**

Manual File Handling Systems

In the manual approach, data storage and retrieval follow the primitive and traditional way of information handling where cards and paper are used for the purpose. The data storage and retrieval will be performed using human labor.

- ✓ Files for as many event and objects as the organization has are used to store information.
- ✓ Each of the files containing various kinds of information is labeled and stored in one or more cabinets.
- ✓ The cabinets could be kept in safe places for security purpose based on the sensitivity of the information contained in it.
- ✓ Insertion and retrieval is done by searching first for the right cabinet then for the right file then the information.
- ✓ One could have an indexing system to facilitate access to the data

Limitations of the Manual approach

- Prone to error
- Difficult to update, retrieve, integrate
- You have the data but it is difficult to compile the information
- Limited to small size information

• Cross referencing is difficult

An alternative approach of data handling is a computerized way of dealing with the information. The computerized approach could also be either decentralized or centralized base on where the data resides in the system.

Two computerized approaches evolved to overcome the limitations of manual data handling

- **File based approach → decentralized**
- **Database approach→ centralized**

File-Based Approach

After the introduction of computer for data processing to the business community, the need to use the device for data storage and processing increase. There were, and still are, several computer applications with file-based processing used for the purpose of data handling. Even though the approach evolved over time, the basic structure is still similar if not identical.

- ✓ File based systems were an early attempt to computerize the manual filing system.
- ✓ This approach is the decentralized computerized data handling method.
- ✓ A collection of application programs perform services for the end-users. In such systems, every application program that provides service to end users define and manage its own data
- ✓ Such systems have number of programs for each of the different applications in the organization.

Limitations of the Traditional File Based approach

As business application become more complex demanding more flexible and reliable data handling methods, the shortcomings of the file-based system became evident. These shortcomings include, but not limited to:

- ✓ **Data Redundancy (Duplication of data)**
  - ➢ **Same data is held by different programs**
  - ➢ **Wasted space**
- ✓ **Separation or Isolation of Data**: Available information in one application may not be known.
  - ➢ Each program maintains its own set of data. Users of one program may be unaware of potentially useful data held by other programs.
- ✓ **Limited data sharing**
- ✓ **Lengthy development and maintenance time**

- ✓ **Data dependency on the application**
- ✓ **Data Inconsistency and confusion**

Database Approach

To become more effective, a new approach was required by the name database approach.

Thus in Database Approach:

• A Database is shared collection of logically related data (and a description of this data), designed to meet the information needs of an organization. (Centralized System). Since it is a shared corporate resource, the database is integrated with minimum amount of or no duplication.

- ➢ **Shared collection** – can be used simultaneously by many departments and users.
- ➢ **Logically related** - comprises the important objects and the relationships between these objects.
- ➢ **Description of the data** – the system catalog (data dictionary or meta-data) provides description of data to enable data independence (program–data independence).

• Database is designed once and used simultaneously by many users.

• Unlike the traditional file-based approach in database approach there is program data independence. That is the separation of the data definition from the application. Thus, the application is not affected by changes made in the data structure and file organization.

Characteristics of database approach

A number of characteristics distinguish the database approach from the traditional approach of programming with files.

1.2.1. Self-Describing Nature of a Database System

1.2.2. Insulation between Programs and Data, and Data Abstraction

1.2.3. Support of Multiple Views of the Data

1.2.4. Sharing of Data and Multiuser Transaction Processing

1.2.1. Self-Describing Nature of a Database System

A fundamental characteristic of the database approach is that the database system contains not only the database itself but also a complete definition or description of the database structure and constraints. This definition is stored in the system **catalog,** which contains information such as the structure of each file, the type and storage format of each data item, and various constraints on the data. The information stored in the catalog is called **meta-data,** and it describes the structure of the primary database

1.2.2. Insulation between Programs and Data, and Data Abstraction

In traditional file processing, the structure of data files is embedded in the access programs, so any changes to the structure of a file may require *changing all programs* that access this file. By contrast, DBMS access programs do not require such changes in most cases. The structure of data files is stored in the DBMS catalog separately from the access programs. We call this property **program-data independence.** For example, a file access program may be written in such a way that it can access only STUDENT records of the structure shown in Figure 1.1. If we want to add another piece of data to each STUDENT record, say the Birthdate, such a program will no longer work and must be changed. By contrast, in a DBMS environment, we just need to change the description of STUDENT records in the catalog to reflect the inclusion of the new data item Birthdate; no programs are changed. The next time a DBMS program refers to the catalog, the new structure of STUDENT records will be accessed and used. The characteristic that allows program-data independence and program-operation independence is called **data abstraction.**

1.2.3. Support of Multiple Views of the Data

A database typically has many users, each of whom may require a different perspective or **view** of the database. A view may be a subset of the database or it may contain **virtual data** that is derived from the database files but is not explicitly stored. Some users may not need to be aware of whether the data they refer to is stored or derived. A multiuser DBMS whose users have a variety of applications must provide facilities for defining multiple views. For example, one user of the database of Figure 1.1 may be interested only in accessing and printing the transcript of each student; the view for this user is shown in Figure 1.2a. A second user, who is interested only in checking that students have taken all the prerequisites of each course for which they register, may require the view shown in Figure 1.2b.

1.2.4. Sharing of Data and Multiuser Transaction Processing

A multiuser DBMS, as its name implies, must allow multiple users to access the database at the same time. This is essential if data for multiple applications is to be integrated and maintained in a single database. The DBMS must include **concurrency control** software to ensure that several users trying to update the same data in a controlled manner so that the result of the updates is correct.

1.3. Actors on the Scene

- ✓ Database Administrators
- ✓ Database Designers

- ✓ End Users
- ✓ System Analysts and Application Programmers (Software Engineers)

Database Administrators

In any organization where many persons use the same resources, there is a need for a chief administrator to oversee and manage these resources. In a database environment, the primary resource is the database itself and the secondary resource is the DBMS and related software. Administering these resources is the responsibility of the database administrator (DBA). The DBA is responsible for authorizing access to the database, for coordinating and monitoring its use, and for acquiring software and hardware resources as needed. The DBA is accountable for problems such as breach of security or poor system response time. In large organizations, the DBA is assisted by a staff that helps carry out these functions.

Database Designers

Database designers are responsible for identifying the data to be stored in the database and for choosing appropriate structures to represent and store this data. These tasks are mostly undertaken before the database is actually implemented and populated with data. It is the responsibility of database designers to communicate with all prospective database users, in order to understand their requirements, and to come up with a design that meets these requirements. In many cases, the designers are on the staff of the DBA and may be assigned other staff responsibilities after the database design is completed. Database designers typically interact with each potential group of users and develop a view of the database that meets the data and processing requirements of this group. These views are then analyzed and integrated with the views of other user groups. The final database design must be capable of supporting the requirements of all user groups.

End Users

End users are the people whose jobs require access to the database for querying, updating, and generating reports; the database primarily exists for their use. There are several categories of end users:

- ✓ ***Casual end users*** occasionally access the database, but they may need different information each time. They use a sophisticated database query language to specify their requests and are typically middle- or high-level managers or other occasional browsers.
- ✓ ***Naive or parametric end users*** make up a sizable portion of database end users. Their main job function revolves around constantly querying and updating the database, using standard types of queries and updates—called canned transactions—that have been carefully programmed and tested. The tasks that such users perform are varied:
- ✓ ***Sophisticated end users*** include engineers, scientists, business analysts, and others who thoroughly familiarize themselves with the facilities of the DBMS so as to implement their applications to meet their complex requirements.
- ✓ ***Stand-alone users*** maintain personal databases by using ready-made program packages

that provide easy-to-use menu- or graphics-based interfaces. An example is the user of a tax package that stores a variety of personal financial data for tax purposes.

A typical DBMS provides multiple facilities to access a database. Naive end users need to learn very little about the facilities provided by the DBMS; they have to understand only the types of standard transactions designed and implemented for their use. Casual users learn only a few facilities that they may use repeatedly. Sophisticated users try to learn most of the DBMS facilities in order to achieve their complex requirements. Stand-alone users typically become very proficient in using a specific software package.

System Analysts and Application Programmers (Software Engineers)

System analysts determine the requirements of end users, especially naive and parametric end users, and develop specifications for canned transactions that meet these requirements. Application programmers implement these specifications as programs; then they test, debug, document, and maintain these canned transactions. Such analysts and programmers (nowadays called software engineers) should be familiar with the full range of capabilities provided by the DBMS to accomplish their tasks.

1.4. Basic Database Terminologies

• Enterprise

– an organization : A library, a bank, a university, etc.

• Entity

– Person, place, thing, or event (property of an entity)

– An "object" in the real world that we are interested in:

– The object student is an entity

• Attribute (Field)

– A character or group of characters (alphabetic or numeric), that has a specific meaning.

Eg. Name, age, telephone, grade, sex, etc.

• Record

– A logically connected set of one or more Attributes that describe a person, place or thing. (Logically related data)

• File

– A collection of related records. For example, a file might contain data about customers; or students of a certain department in a university.

• Database

– Collection of Files

1.5. Advantages of Using the DBMS Approach

Controlling Redundancy

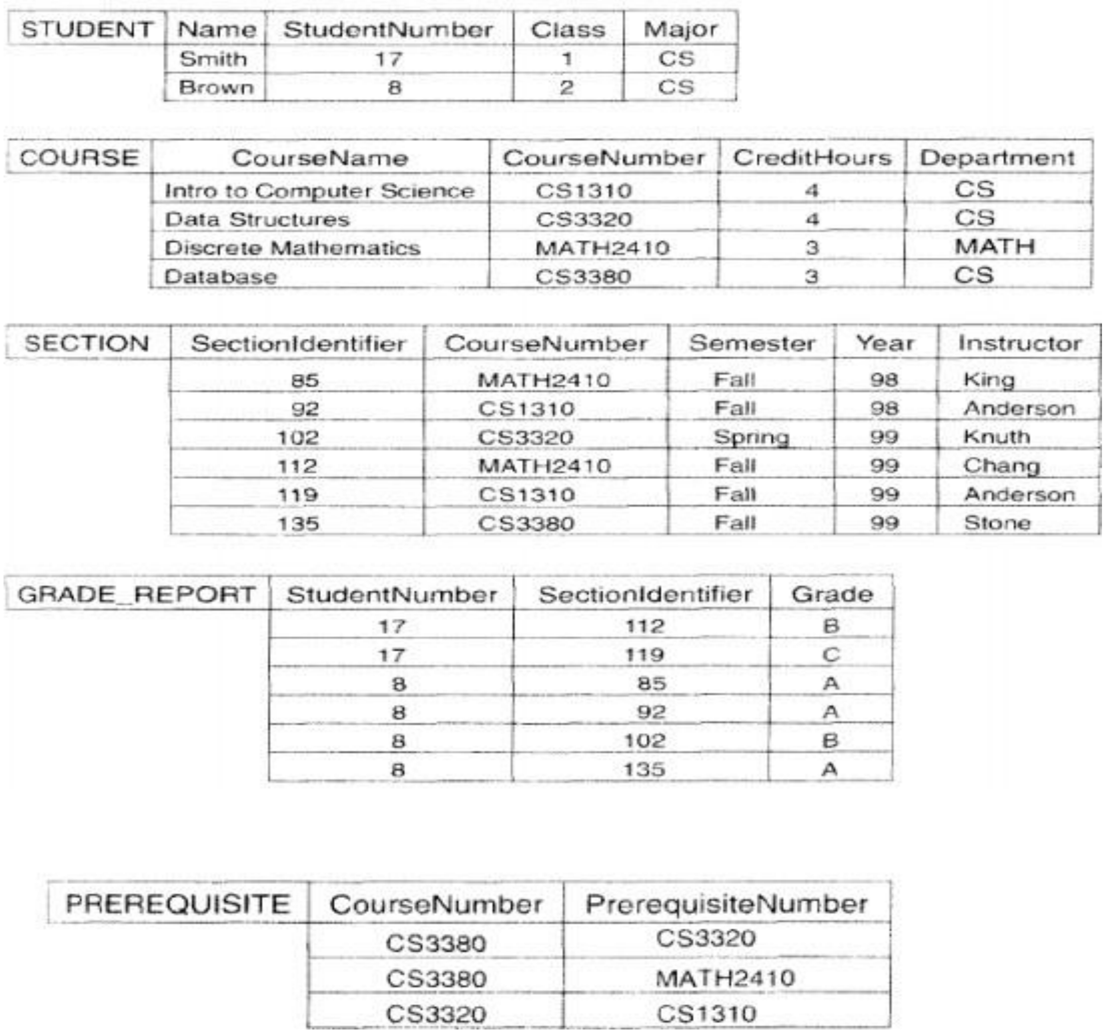
Redundancy in storing the same data multiple times leads to several problems. First, there is the need to perform a single logical update. Second, storage space is wasted when the same data is stored repeatedly, and this problem may be serious for large databases. Third, files that represent the same data may be come inconsistent. This may happen because an update is applied to some of the files but not to others.

Restricting Unauthorized Access

When multiple users share a large database, it is likely that most users will not be authorized to access all information in the database. Typically, users or user groups are given account numbers protected by passwords, which they can use to gain access to the database. ADBMS should provide a security and authorization subsystem, which the DBA uses to create accounts and to specify account restrictions. The DBMS should then enforce these restrictions automatically. Notice that we can apply similar controls to the DBMS software. For example, only the DBA's staff may be allowed to use certain privileged software, such as the software for creating new accounts. Similarly, parametric users may be allowed to access the database only through the canned transactions developed for their use.

Providing Backup and Recovery

A DBMS must provide facilities for recovering from hardware or software failures. The backup and recovery subsystem of the DBMS is responsible for recovery. For example, if the computer system fails in the middle of a complex update transaction, the recovery subsystem is responsible for making sure that the database is restored to the state it was in before the transaction started executing. Alternatively, the recovery subsystem could ensure that the transaction is resumed from the point at which it was interrupted so that its full effect is recorded in the database.



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

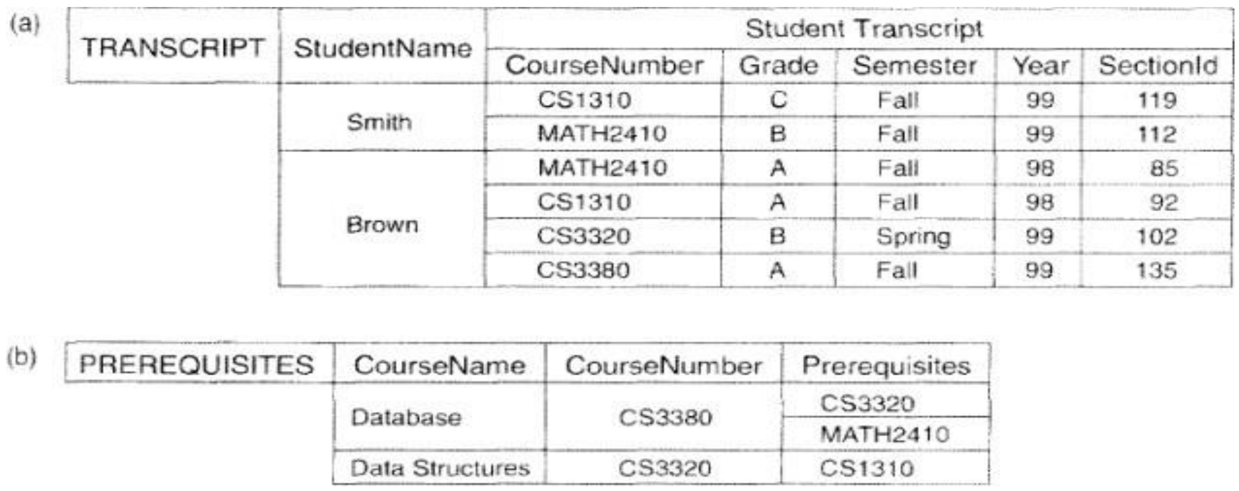
COURSE	CourseName	CourseNumber	CreditHours	Department
	Intro to Computer Science	CS1310	4	CS
	Data Structures	CS3320	4	CS
	Discrete Mathematics	MATH2410	3	MATH
	Database	CS3380	3	CS

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

Figure 1.1: A database that stores student and course information



(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

Figure 1.2: Two views derived from the database in figure 1.1