Overview of Database Systems

CSC 209 2.0 Database Management Systems

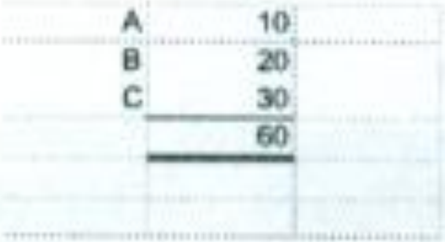
Dr. Ananda Edirisuriya

(Handout 1)

Key Terms

- Data
  - A representation of facts, concepts or instructions in a formalised manner suitable for communication, interpretation or processing by human beings or by automatic means.
  - Raw data which is unprocessed

DATA

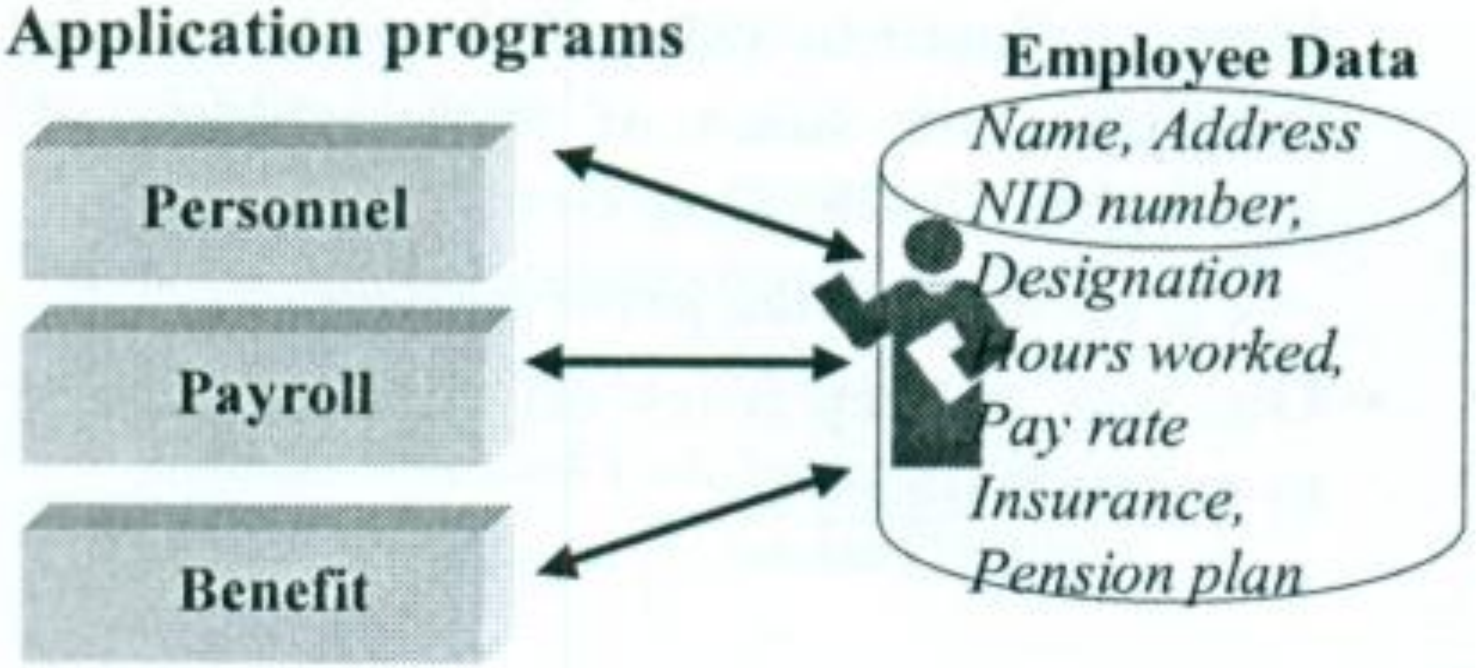


A	10
B	20
C	30
	60

Letters, numbers, text, colours, symbols, shapes, graphics, images, , sound, video or other facts and figures are data suitable for processing.

- Information
  - Processed or organised or summarised data.
  - Knowledge derived from data.
- Process Date of Birth → Age
- Process Name → Surname
- Process Address → City
- Process Salary (all) → Highest paid employee
- Process all → No of employees

- **Database**
  - **A collection of interrelated data items that can be processed by one or more application systems.**



Application programs

Personnel

Payroll

Benefit

Employee Data

Name, Address
NID number,
Designation
Hours worked,
Pay rate
Insurance,
Pension plan

File

- **A collection of related records**

Employee file **(Name, Designation, Depart)**

De Silva	Manager	Personnel
Perera	Secretary	Personnel
Dias	Salesman	Sales
....	....	....

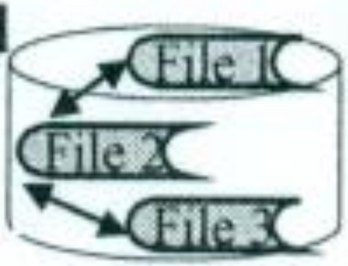
Department file **(Depart, Manager, Dept Addr, Dept Phone)**

Personnel	De Silva	Colombo	589123
Sales	Alwis	Kandy	987275
....	....	....	....

- All data of the database may be in one file (simplest)

E.g. Student information (name, address, registration number, date of birth) and marks of course units in one file

- it may be in a number of files, depending on the way database was designed and the data subsequently represented



E.g. Student information in one file and Course results in another file

Record

A group of related fields. Each field is a data item that is part of a large record

All the student information required for the registration process (Student surname, initials, reference number, date of birth, address, examination centre, entry requirements, registration number)

Field (Data Item)

- The smallest unit of data that has meaning to its users.

Name, Age, address etc.

Byte

- A single character (letter, number, symbol) is represented using a group of bits

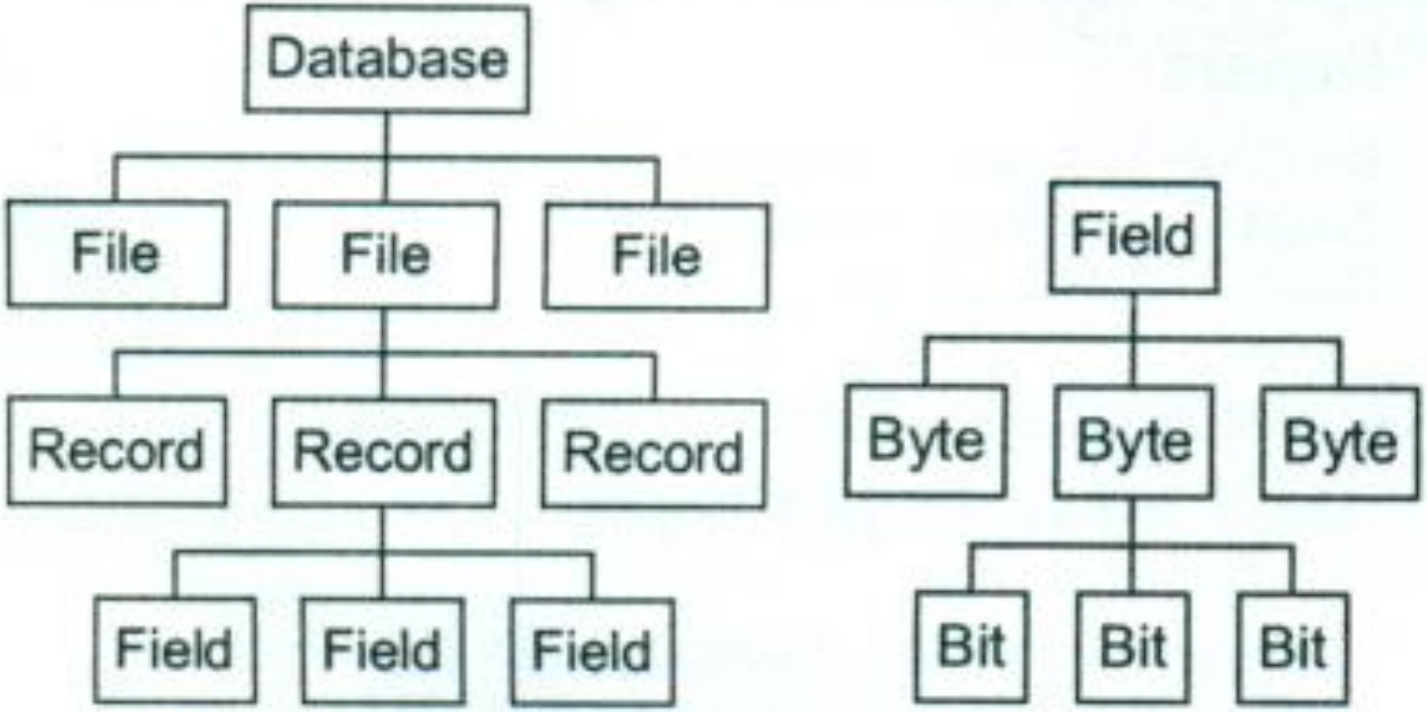
  E.g. 10101010 letter J in ASCII

Bit

- The smallest unit of data

  E.g. 0 or 1

Data Hierarchy



Database

File

File

File

Record

Record

Record

Field

Field

Field

Field

Byte

Byte

Byte

Bit

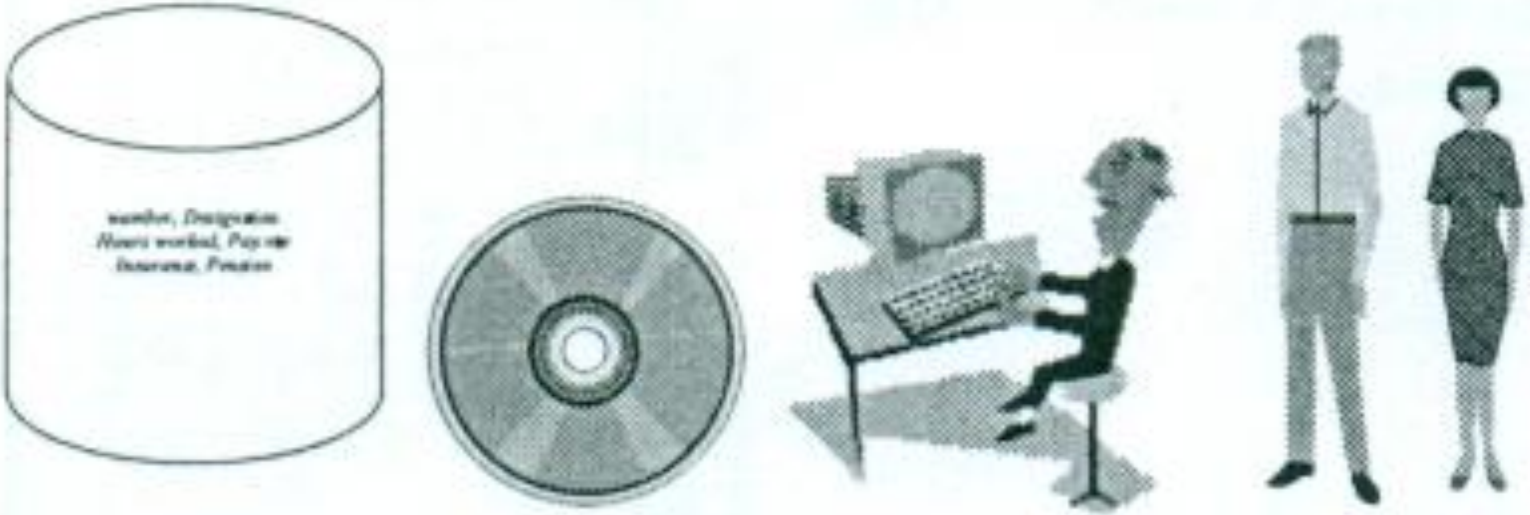
Bit

Bit

Why use a Database?

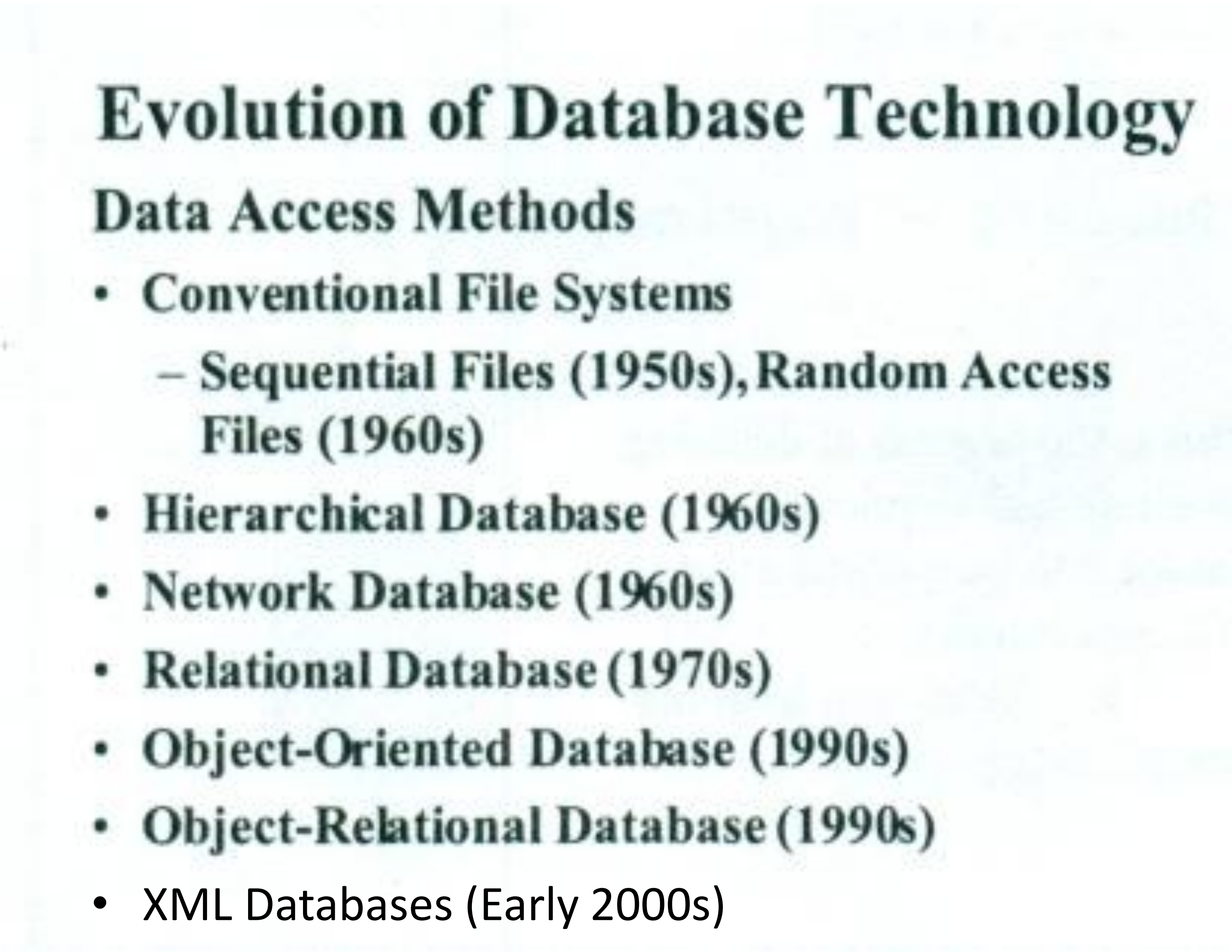
- Many people collect things
- If you collect any thing, you probably are familiar with some of the problems of managing a collection
  - e.g. stamps, photos, paper cuttings
- One way to keep track off a collection is to create a database

- **Database System**
  - **A database, a database management system and appropriate hardware and personnel.**



What is a Database System?

- A computerised record keeping system that organises data into records in one or more databases / files.

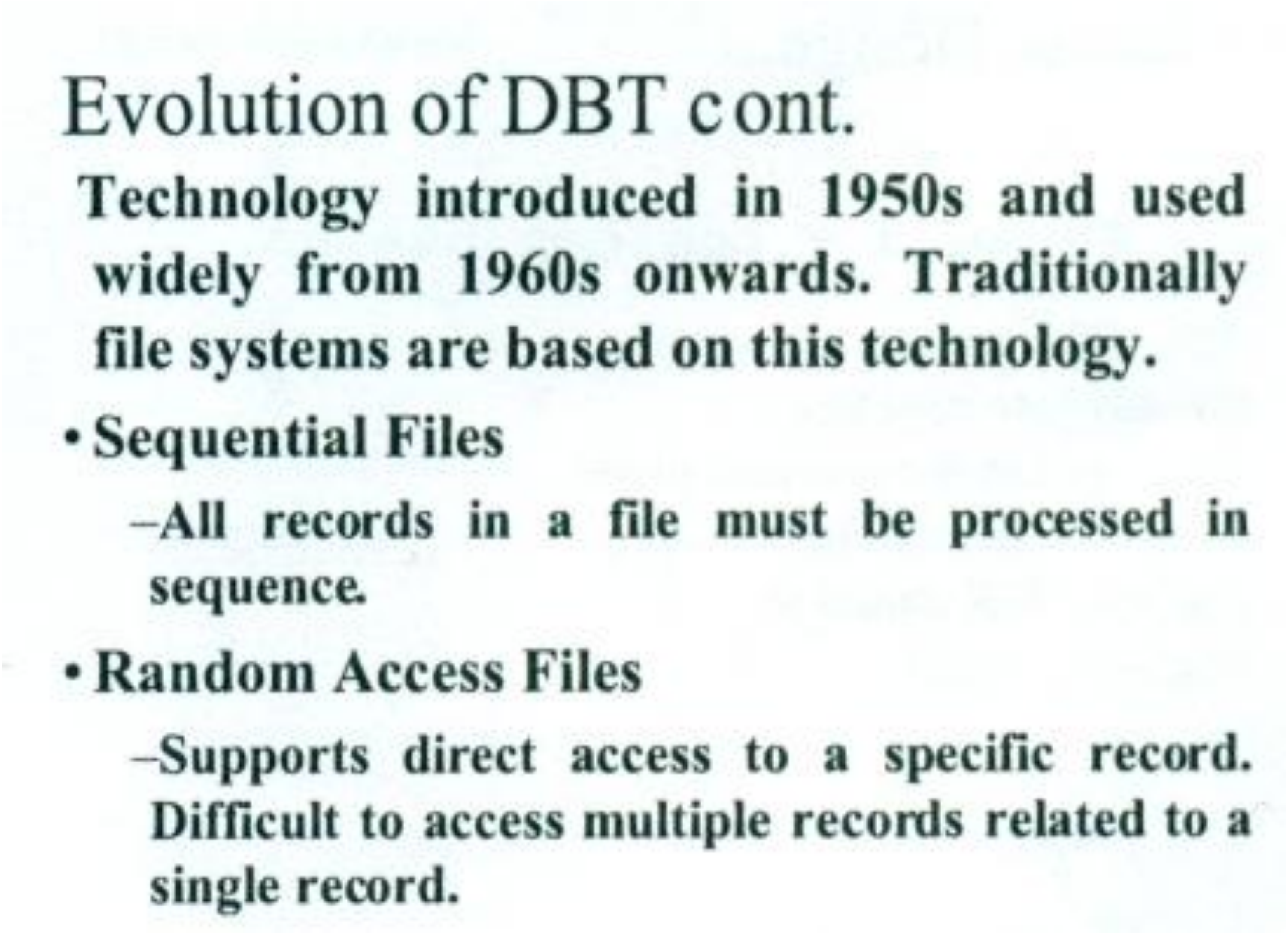


Evolution of Database Technology

Data Access Methods

- Conventional File Systems
  - Sequential Files (1950s), Random Access Files (1960s)
- Hierarchical Database (1960s)
- Network Database (1960s)
- Relational Database (1970s)
- Object-Oriented Database (1990s)
- Object-Relational Database (1990s)
- XML Databases (Early 2000s)

Traditional File systems



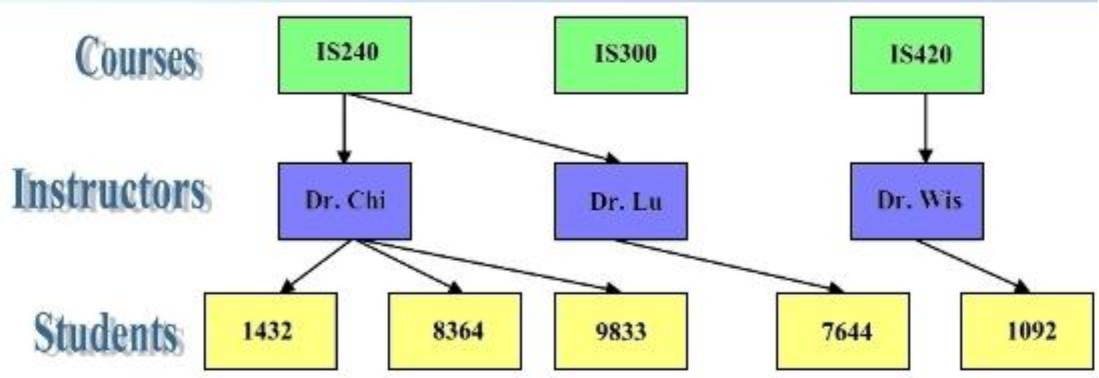
Evolution of DBT cont.

Technology introduced in 1950s and used widely from 1960s onwards. Traditionally file systems are based on this technology.

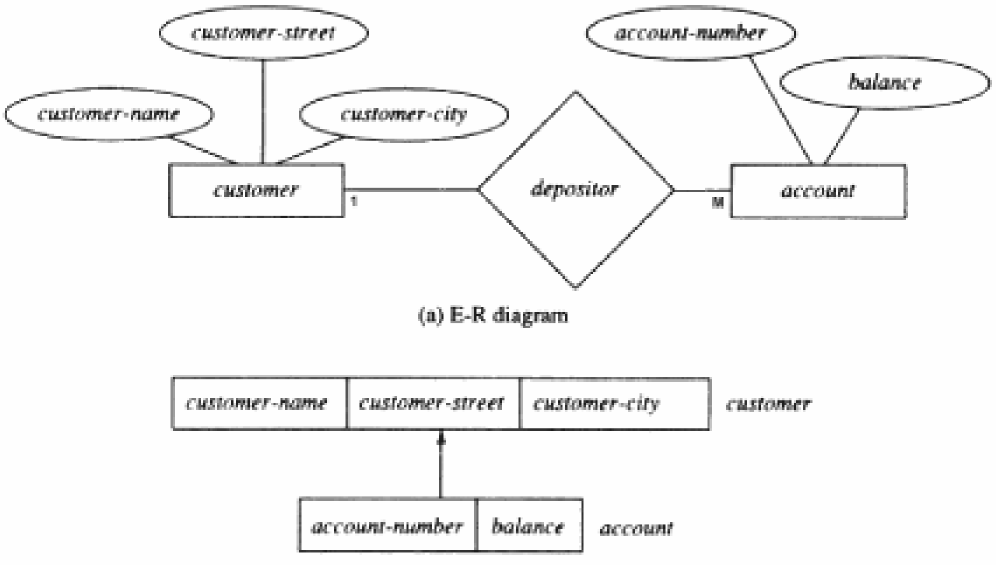
- Sequential Files
  - –All records in a file must be processed in sequence.
- Random Access Files
  - –Supports direct access to a specific record. Difficult to access multiple records related to a single record.

Hierarchical data model

- Records are arranged in a tree structure.
- A parent record may have more than one child, but a child always has only one parent.
- To locate a particular record, starts at the top of the tree with a parent record and trace down the tree to the child.



Hierarchical data model – Tree diagram



(a) E-R diagram

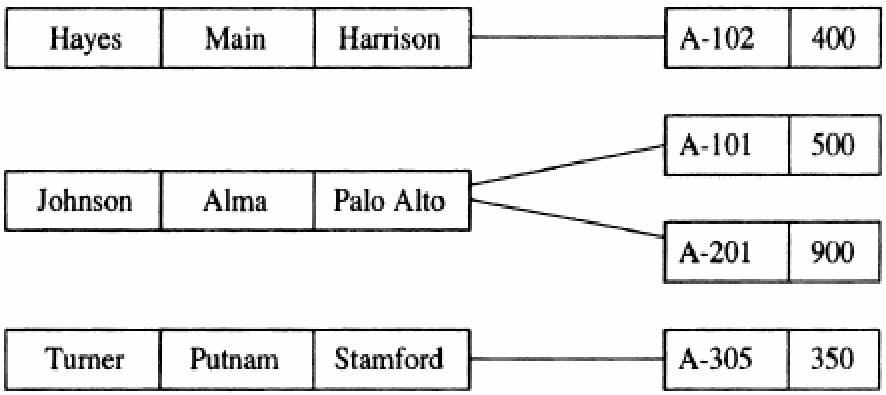
(b) tree-structure diagram

- Data are represented as *collection of records*
- Relationships are represented as *links*
- Each record is a collection of fields:

```
type customer =     record
                          customer-name: string;
                          customer-street: string;
                          customer-city: string;
                    end

type account =      record
                          account-number: string;
                          balance: integer;
                    end
```

Hierarchical data model



Hayes Main Harrison

A-102 400

Johnson Alma Palo Alto

A-101 500

A-201 900

Turner Putnam Stamford

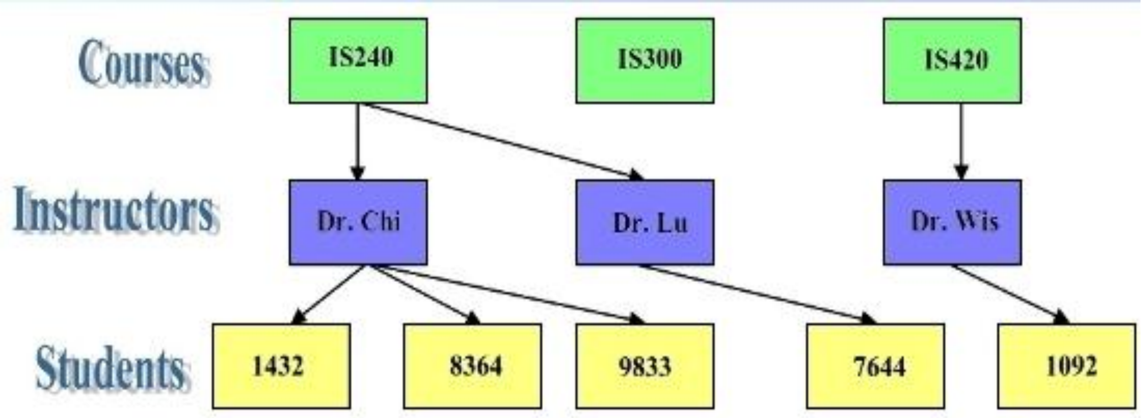
A-305 350

Sample database.

- A parent will have a list of pointers to each of their children.
- Handle one-to-many relationships well but do not handle many-to-many relationships well.

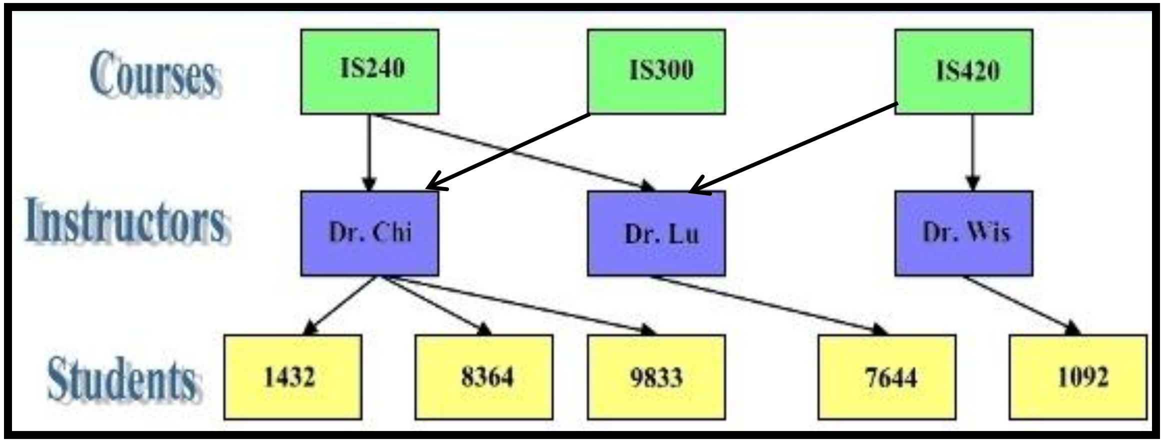
Hierarchical data model

- The relation between course and Instructor is M:N. (This cannot model using Hierarchical data model).



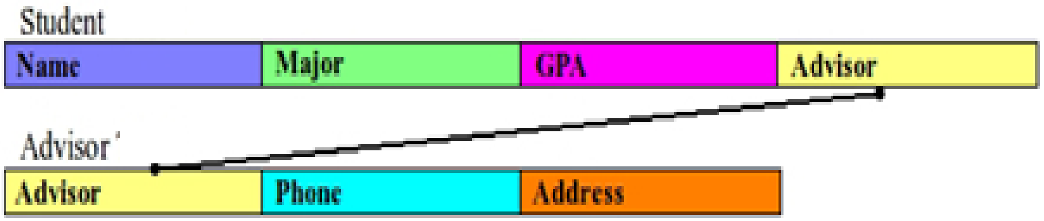
Network data model

- Records are arranged in a graph structure.
- A parent record may have more than one child, and a child may have more than one parent.
- Can handle M:N relationships.



Relational data model

- Data are stored in tables.
- Relates or connects data in different files through the use of a key field, or common data element.



Object data model

Abstraction

- **Identify essential aspects of an entity and ignore the unimportant properties.**
- **Concentrate on what an object is and what it does.**
- **Delay implementation details.**
- **2 aspects:**

  Encapsulation

  An object contains data structure and operations.

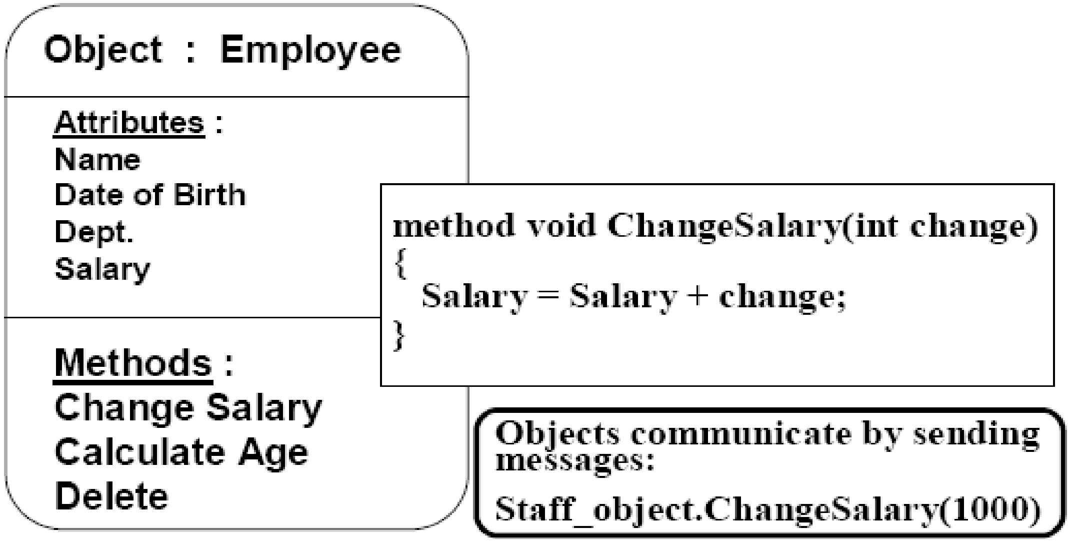
  Information hiding

  We present external aspects of an object to the outside world and hide its internal details.

Class

- **Similar objects can be grouped as a class (cf. a type or a structure in C)**
- **Each such object is called an instance (cf. a variable)**

Object data model



Object : Employee

Attributes :
Name
Date of Birth
Dept.
Salary

Methods :
Change Salary
Calculate Age
Delete

```
method void ChangeSalary(int change)
{
   Salary = Salary + change;
}
```

Objects communicate by sending messages:

```
Staff_object.ChangeSalary(1000)
```

- Use "objects" as elements within database files.
- An object consists of text, sound, images and instructions on the action to be taken on the data.

Example: Traditional data models such as Hierarchical, Network and Relational data models can contain only numeric and text data. An object-oriented database might also contain the pictures and videos. Object would store operations, called "methods" that perform actions on the data.

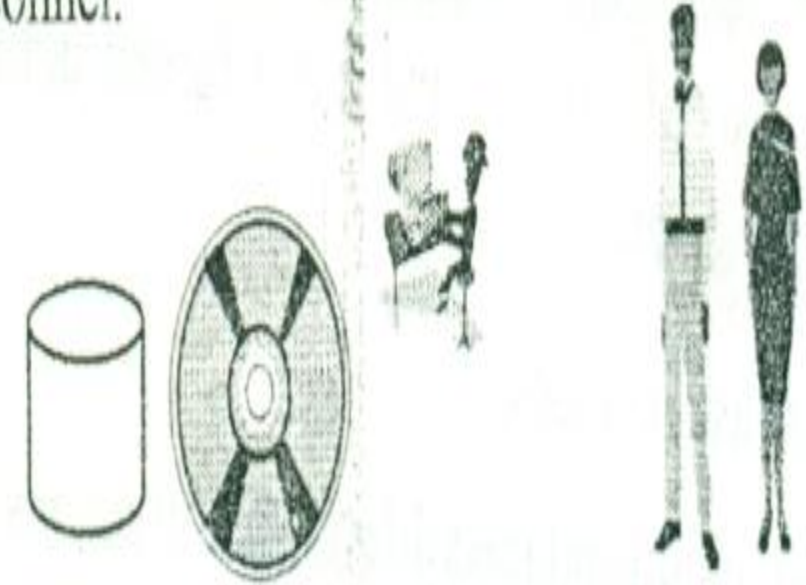
XML data model

- Use XML technology.
- Data are stored using XML tags.

```
<University>
 <Student>
  <Name> Anil</Name>
  <Age> 22</Age>
  <GPA> 3.5</GPA>
</Student>
 <Student>
  <Name> Kamal</Name>
  <Age> 23</Age>
  <GPA> 3.8</GPA>
</Student>
 <Student>
  <Name> Kumari</Name>
  <Age> 24</Age>
  <GPA> 4.0</GPA>
</Student>
</University>
```

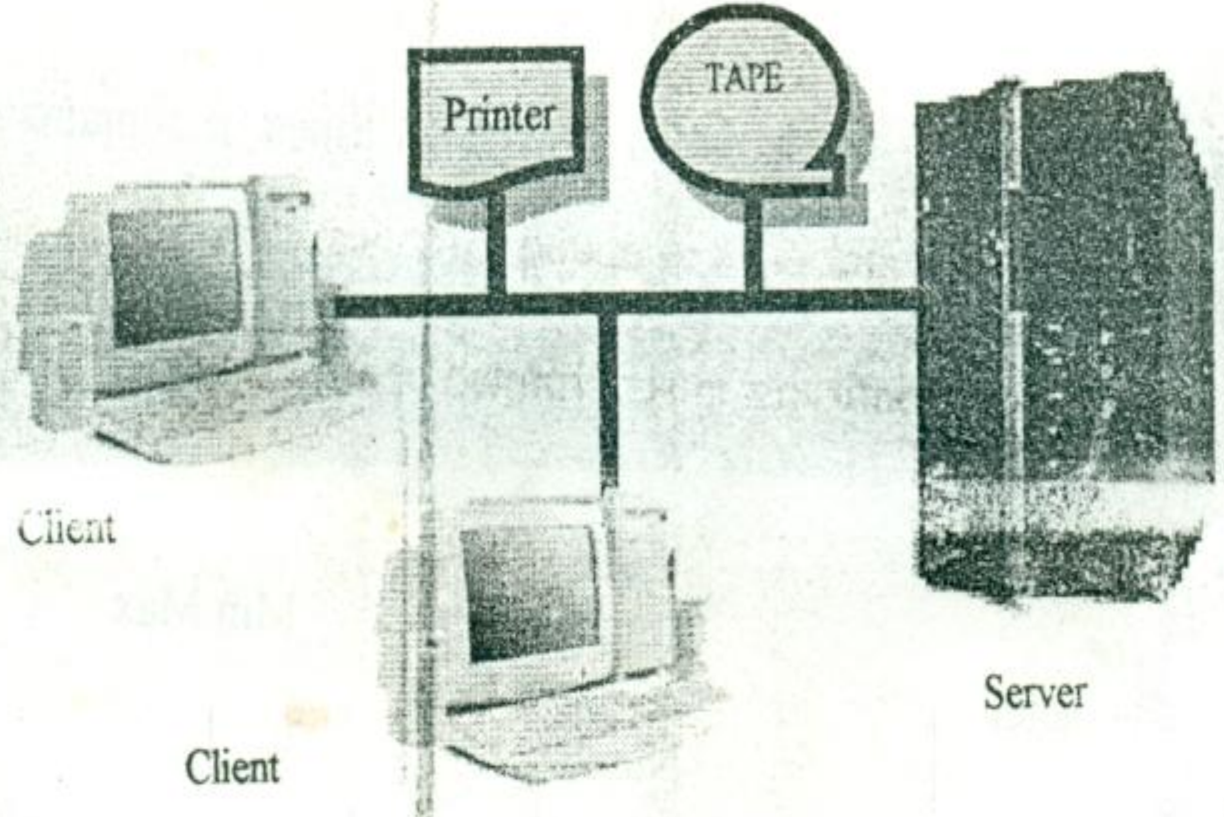
Database System Components

- **Database System**
  - A database, a database management system and appropriate hardware and personnel.



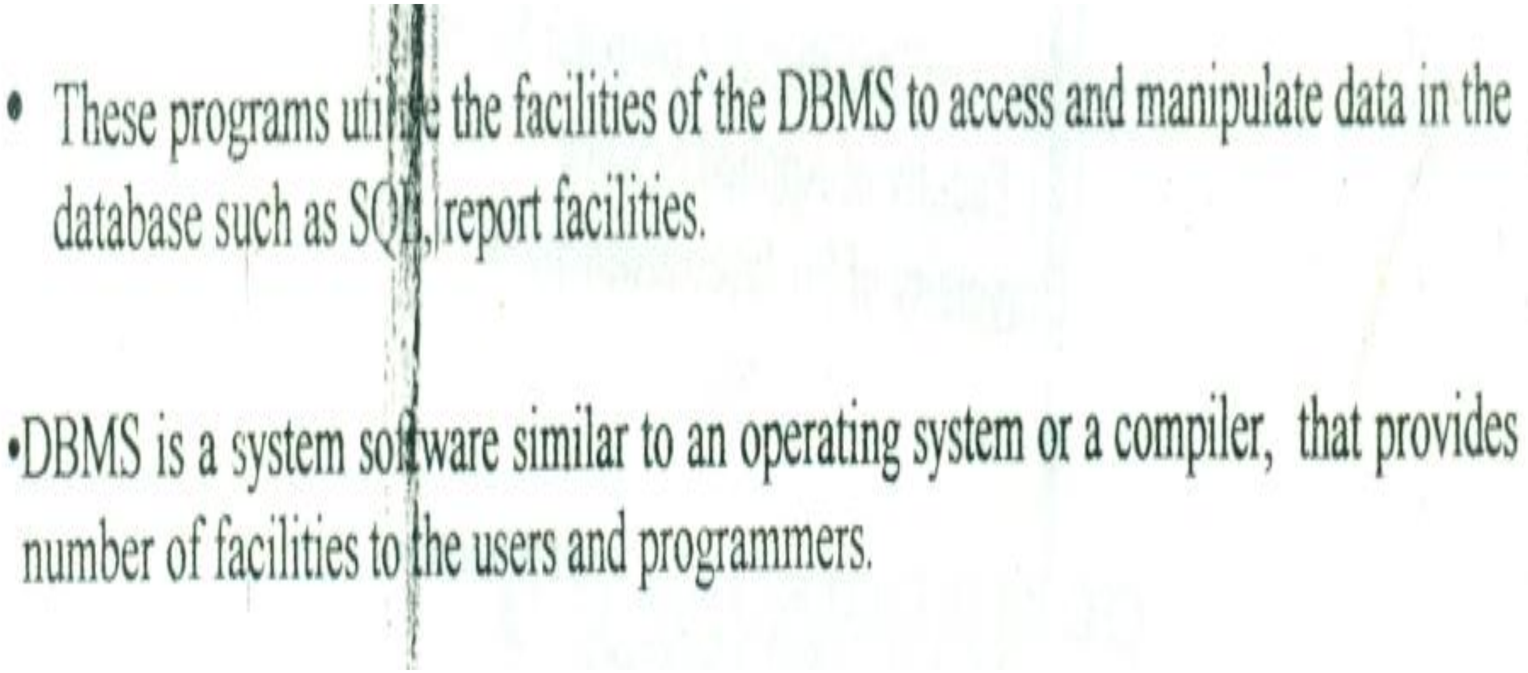
Database System: Hardware

- Set of physical devices on which a database resides. It consists of one or more computers, disk drives, CRT terminals, printers, tape drives, connecting cables and other auxiliary and connecting hardware.



Database System: Software

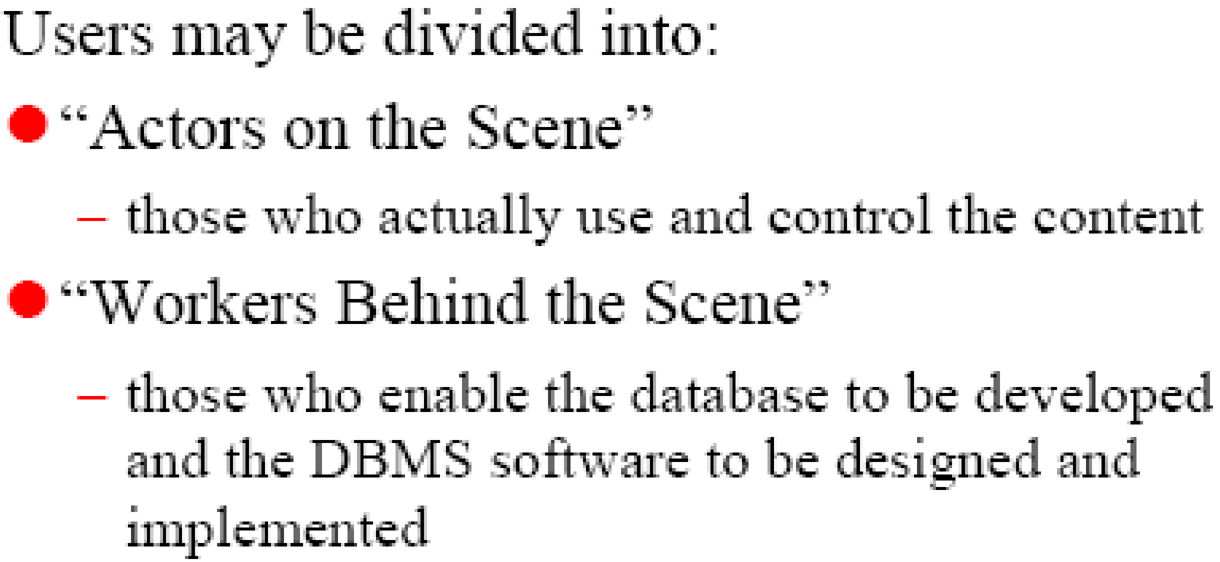
- A database software includes two types of software
  - General-purpose database management software, usually called the database management system (DBMS)
  - Application software that uses DBMS facilities to manipulate the database to achieve a specific business function, such as providing reports or documents, which can be used by users.
- Application software is generally written standard programming language such as C, VB, or it may be written in a language (commonly called a fourth-generation language) supplied with the DBMS.



- These programs utilize the facilities of the DBMS to access and manipulate data in the database such as SQL, report facilities.

•DBMS is a system software similar to an operating system or a compiler, that provides number of facilities to the users and programmers.

Database system: Users



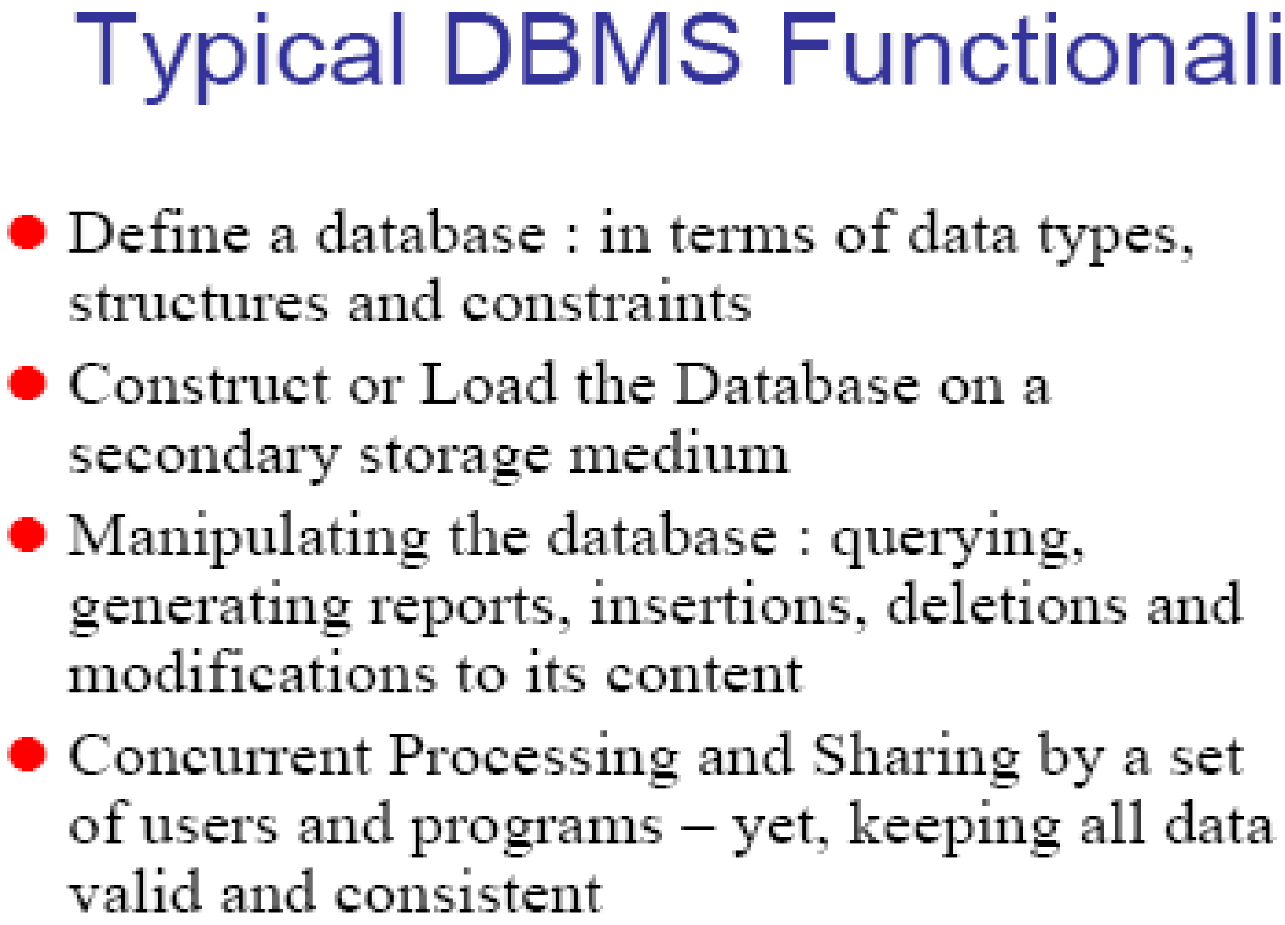
Users may be divided into:

- “Actors on the Scene”
  - those who actually use and control the content
- “Workers Behind the Scene”
  - those who enable the database to be developed and the DBMS software to be designed and implemented

Database Users

Actors on the scene

- **Database administrators:** responsible for authorizing access to the database, for co-ordinating and monitoring its use, acquiring software, and hardware resources, controlling its use and monitoring efficiency of operations.
- **Database Designers:** responsible to define the content, the structure, the constraints, and functions or transactions against the database. They must communicate with the end-users and understand their needs.
- **End-users:** they use the data for queries, reports and some of them actually update the database content.



Typical DBMS Functionality

- Define a database : in terms of data types, structures and constraints
- Construct or Load the Database on a secondary storage medium
- Manipulating the database : querying, generating reports, insertions, deletions and modifications to its content
- Concurrent Processing and Sharing by a set of users and programs – yet, keeping all data valid and consistent

- Data are been shared by different users and programs.

- Concurrent processing refers to the simultaneous execution of programs

Typical DBMS Functionality

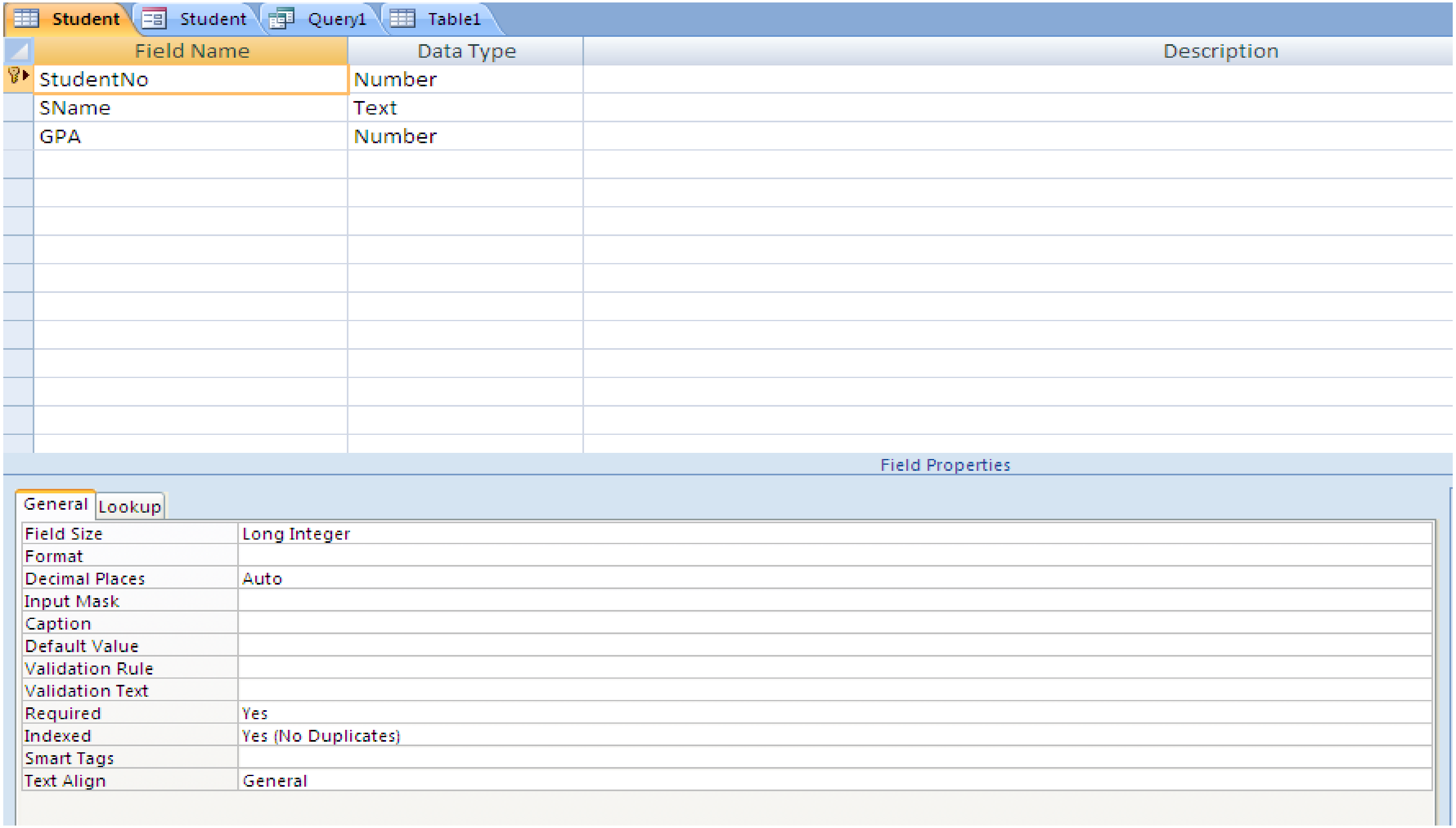
Other features:

- Protection or Security measures to prevent unauthorized access
- Presentation and Visualization of data

Main Characteristics of the Database Approach

- Self-describing 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.
- Independence between programs and data
  - Called **program-data independence**
  - Allows changing data storage structures and operations without having to change the DBMS

System Catalog



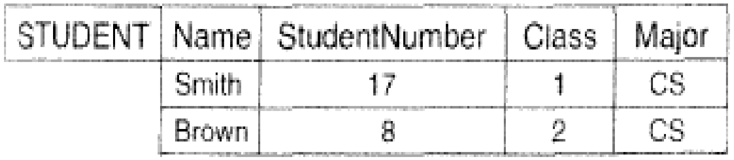
Field Name	Data Type	Description
StudentNo	Number	
SName	Text	
GPA	Number	

Field Size	Long Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	Yes
Indexed	Yes (No Duplicates)
Smart Tags	
Text Align	General

Main Characteristics of the Database Approach

- Data Abstraction
  - A **data model** is used to hide storage details and present the users with a *conceptual view* of the database.
- 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.

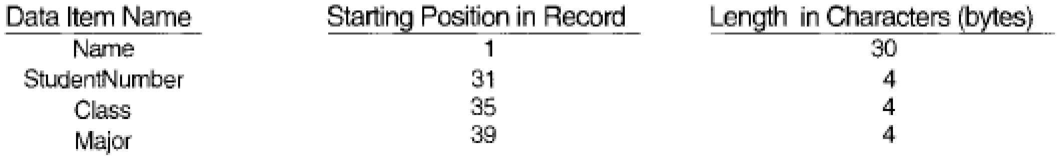
Data Abstraction



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

Student(Name: String, StudentNumber: Numeric, Class: Numeric, Major: String)

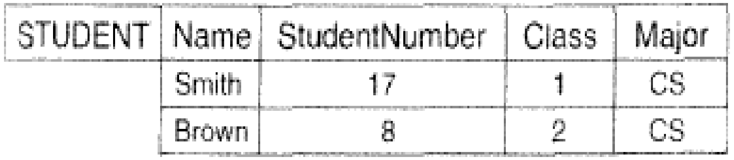
Data model for student record(High level view)



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

Internal storage format for a STUDENT record.

Multiple view

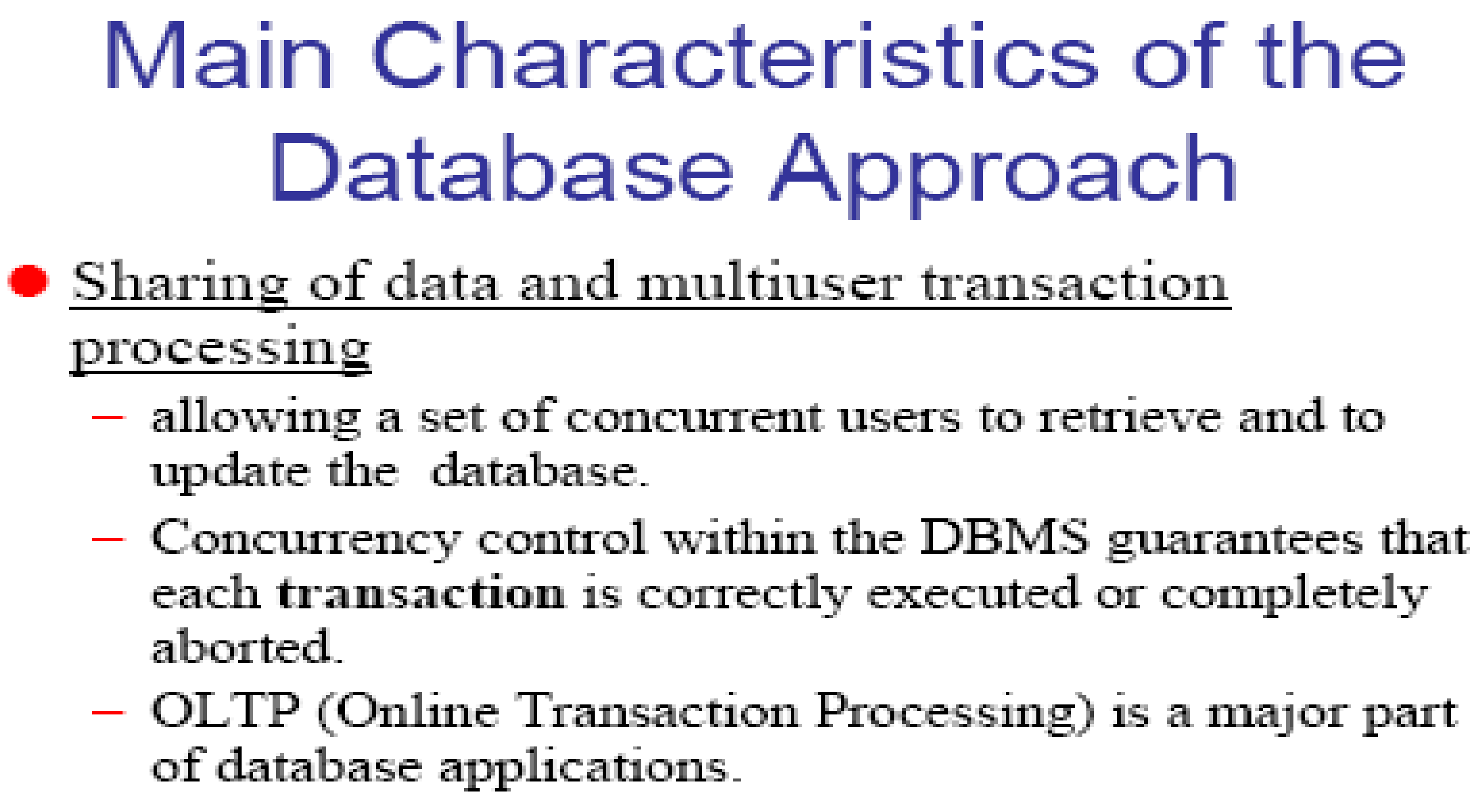


STUDENT	Name	StudentNumber	Class	Major
	Smith	17	1	CS
	Brown	8	2	CS

Student(Name: String, StudentNumber: Numeric, Class: Numeric, Major: String)

Student_Name(Name: String)

Student_Major(StudentNumber: Numeric, Major: String)



Main Characteristics of the Database Approach

- Sharing of data and multiuser transaction processing
  - allowing a set of concurrent users to retrieve and to update the database.
  - Concurrency control within the DBMS guarantees that each transaction is correctly executed or completely aborted.
  - OLTP (Online Transaction Processing) is a major part of database applications.

The DBMS include concurrency control software to control data access. (Several users trying to update the same data, it should be done in a controlled manner so that the result of the updates is correct.)

For example, when several reservation clerks try to assign a seat on an airline flight, the DBMS should ensure that each seat can be accessed by only one clerk at a time for assignment to a passenger.

Advantages of Using the Database Approach

- Controlling redundancy in data storage·
- Sharing of data among multiple users.
- Restricting unauthorized access to data.
- Providing Storage Structures for efficient Query Processing

Advantages of Using the Database Approach

- 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 using rules

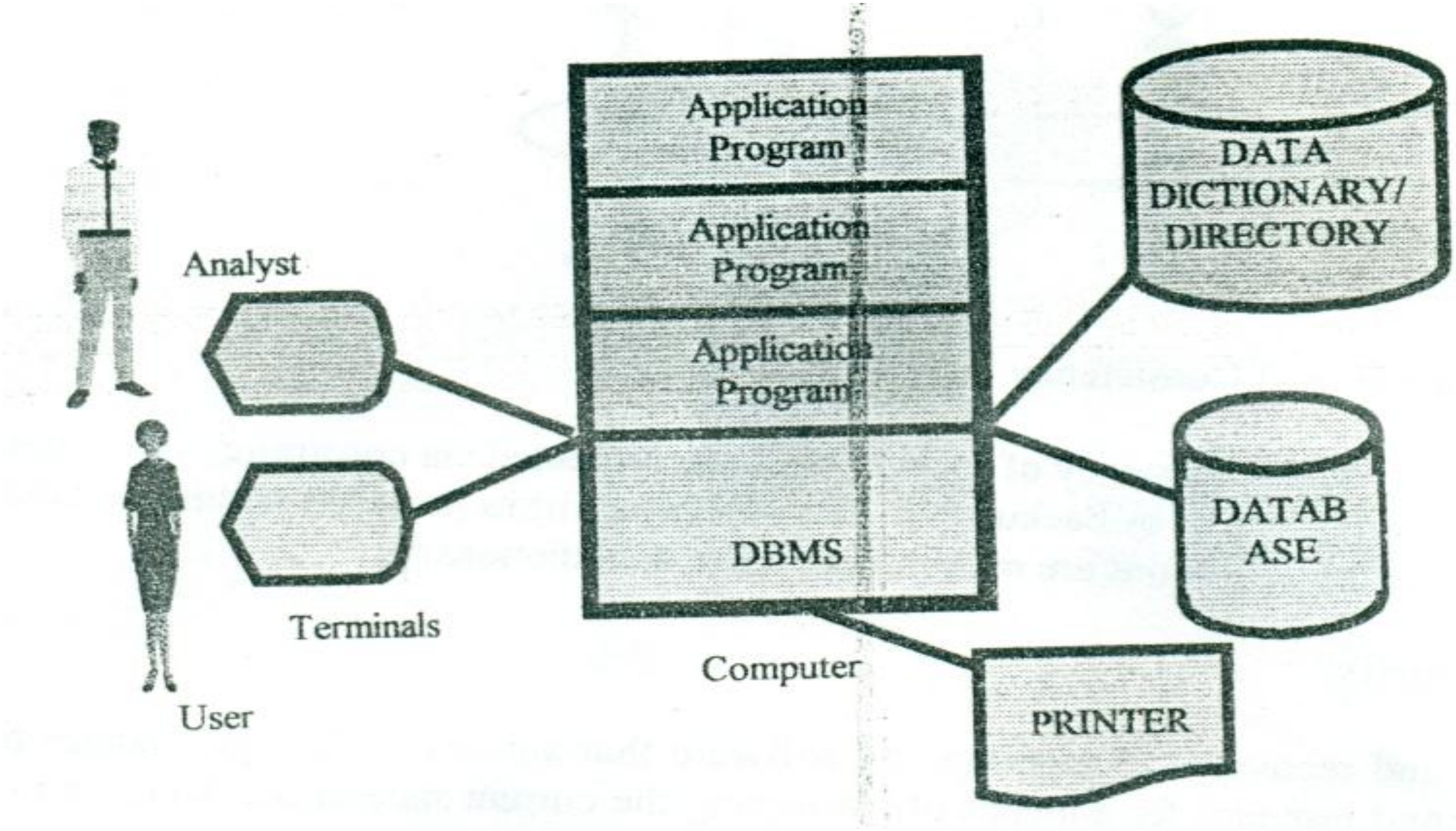
Additional Implications of Using the Database Approach

- Potential for enforcing standards: this is very 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) etc.
- Reduced application development time: incremental time to add each new application is reduced.

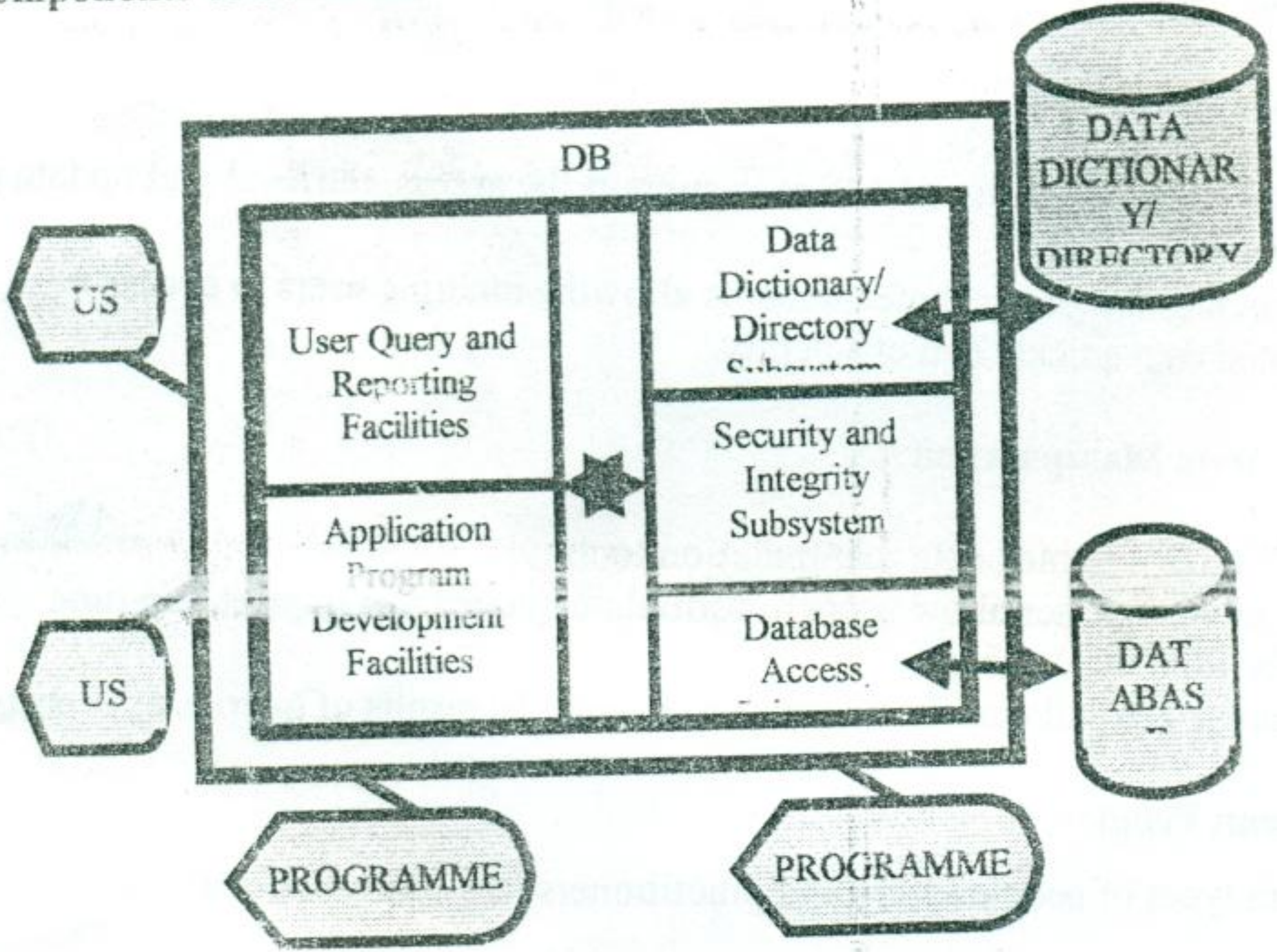
Additional Implications of Using the Database Approach

- Flexibility to change data structures: database structure may evolve as new requirements are defined.
- Availability of up-to-date information – very important for on-line transaction systems such as airline, hotel, car reservations.
- Economies of scale: by consolidating data and applications across departments wasteful overlap of resources and personnel can be avoided.

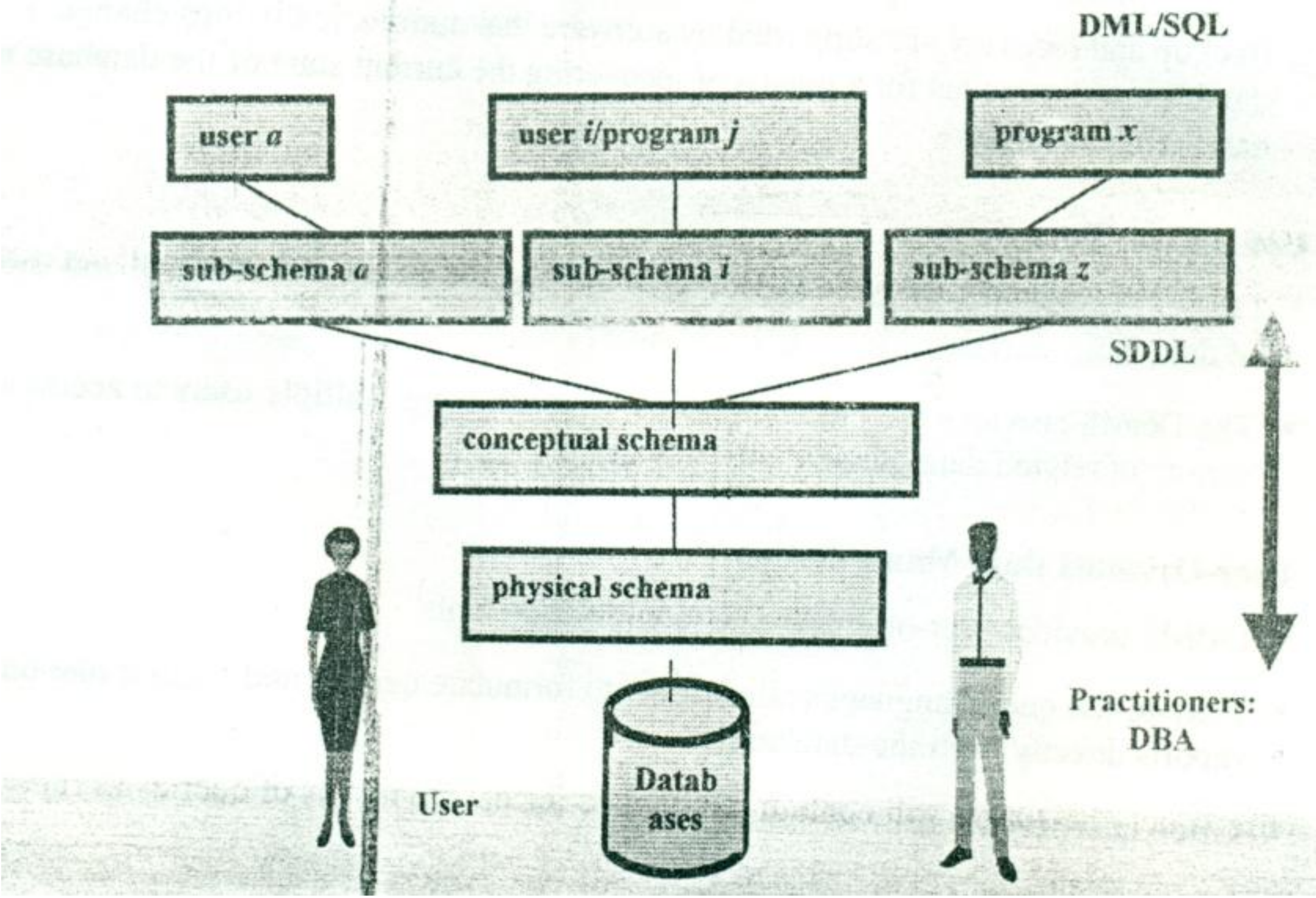
Component of a database system

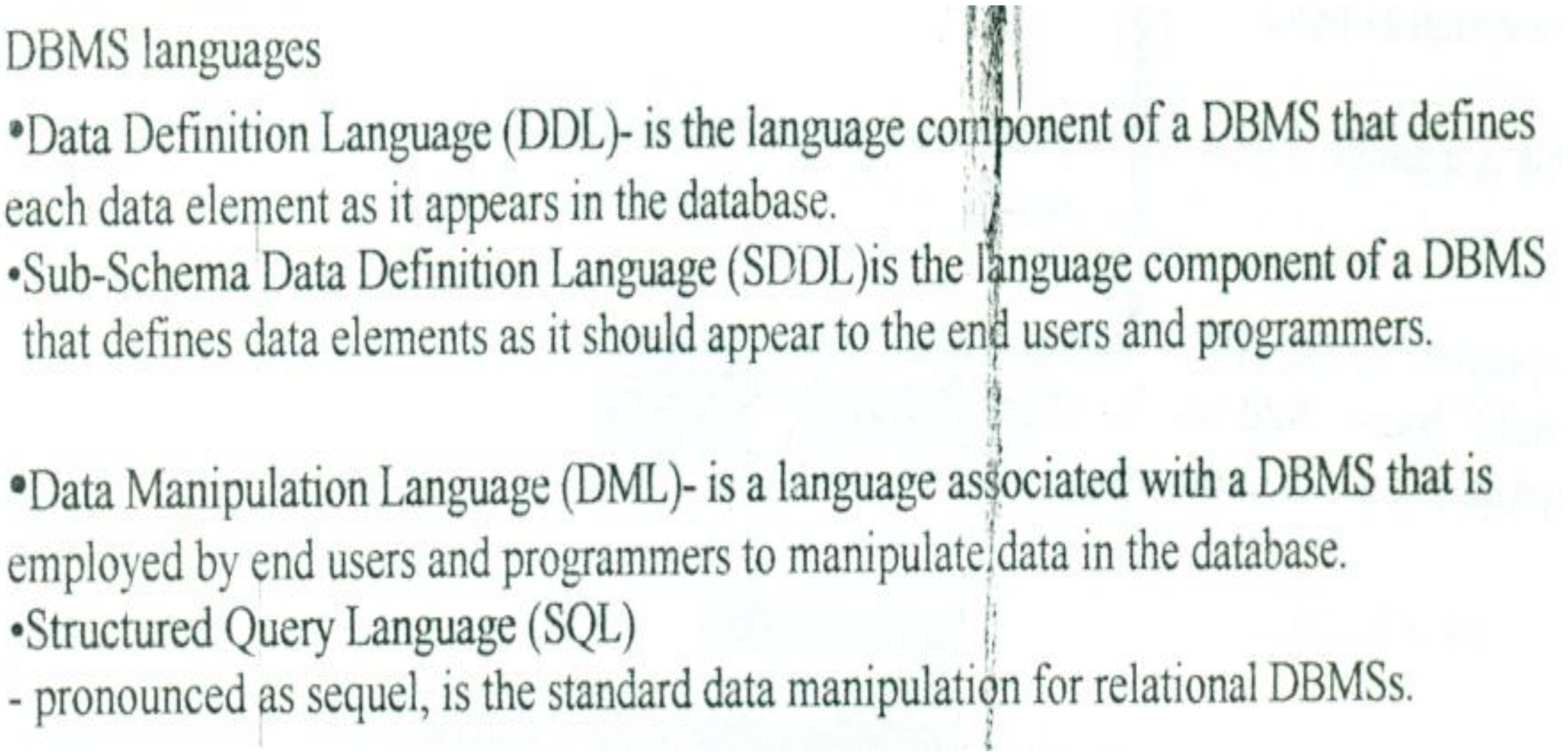


Components of a DBMS


DB
User Query and Reporting Facilities
Application Program Development Facilities
Data Dictionary/ Directory Subsystem
Security and Integrity Subsystem
Database Access
DATA DICTIONAR Y/ DIRECTORY
DAT ABAS
US
US
PROGRAMME
PROGRAMME

3 Level Architecture





DBMS languages

•Data Definition Language (DDL)- is the language component of a DBMS that defines each data element as it appears in the database.

•Sub-Schema Data Definition Language (SDDL)is the language component of a DBMS that defines data elements as it should appear to the end users and programmers.

•Data Manipulation Language (DML)- is a language associated with a DBMS that is employed by end users and programmers to manipulate data in the database.

•Structured Query Language (SQL)

- pronounced as sequel, is the standard data manipulation for relational DBMSs.

Disadvantages

- DBMS are more vulnerable than file-based system because of the centralised nature of a large integrated database.
- If a failure occurs the recovery process is more complex and some times may results in lost transactions.
- Hardware, software and personnel cost are higher for DBMS.