# Database System

## ❖What is a Database System ?

* Database: A collection of related data.
* Database Management System (DBMS): A software package to facilitate the creation and maintenance of a computerized database.
* Database System: DBMS + data.

  Computerized record system.

## ❖Who uses a DB ?

1. bank (account).
2. airline (passenger and seats).
3. hospital (patient).
4. university (students).

## ❖Operations on a DB

1. add/remove files.
2. insert/delete/update data in existing files.
3. retrieve data .

# ❖ Example of a DB & Operations

field (data type, range) --> record (related fields) --> file (multiple instances).

(a) Given table Cellar

| Wine | Year | Bottles |
|---|---|---|
| Zinfandel | 77 | 9 |
| Chardonnay | 82 | 4 |
| Fume Blane | 78 | 2 |
| Pinot Noir | 77 | 3 |

## (b) Operators

* Row subset (retrieval)

```
Select Wine, Year, Bottles
From Cellar
Where Year = 77;
```

| Wine | Year | Bottles |
|---|---|---|
| Zinfandel | 77 | 9 |
| Pinot Noir | 77 | 3 |

* Column subset(retrieval)

```
Select Wine, Bottles
From Cellar;
```

| Wine | Bottles |
|---|---|
| Zinfandel | 9 |
| Chardonnay | 4 |
| Fume Blane | 2 |
| Pinot Noir | 3 |

* Insertion

Insert into Cellar

Values (‘Gammy’, 85, 2);

| Wine | Year | Bottles |
|---|---|---|
| Zinfandel | 77 | 9 |
| Chardonnay | 82 | 4 |
| Fume Blane | 78 | 2 |
| Pinot Noir | 77 | 3 |
| Gammy | 85 | 2 |

* Update

Update Cellar

Set Bottles = 4

Where Wine = ‘Gammy’;

| Wine | Year | Bottles |
|---|---|---|
| Zinfandel | 77 | 9 |
| Chardonnay | 82 | 4 |
| Fume Blane | 78 | 2 |
| Pinot Noir | 77 | 3 |
| Gammy | 85 | 4 |

* Deletion

Delete From Cellar

Where Year = 77;

| Wine | Year | Bottles |
|---|---|---|
| Chardonnay | 82 | 4 |
| Fume Blane | 78 | 2 |
| Gammy | 85 | 2 |

## ❖ Why using a DB ?

a) Speed. (special data organization and access strategy).

b) Centralized control of data (integrated and shared).

# Advantages:

1. Redundancy can be reduced (as compared to each one has its private file -->waste space).
2. Inconsistency can be avoided. (when redundancy is allowed, data update may cause inconsistency.
3. Data can be shared.

   Shared: individual pieces of data in the database can be shared among several different users. They can access the same data at the same time (view in different ways).
4. Standards can be enforced. (data representation)

   (Standardizing stored data formats is desirable as an aid to data interchange.)

5. Security restrictions can be applied.
6. Integrity can be maintained. (check data type or range or non-existing information).

   Integrated: a unification of several distinct data files, with any redundancy among those files either wholly or partly eliminated.

Ex. EMPLOYEE (name, address, department,salary) and Enrollment (name,course);

instead of Enrollment (name, address, course).

7. Conflicting requirements can be balanced. (provide overall (best) service.)
8. Data Independence.

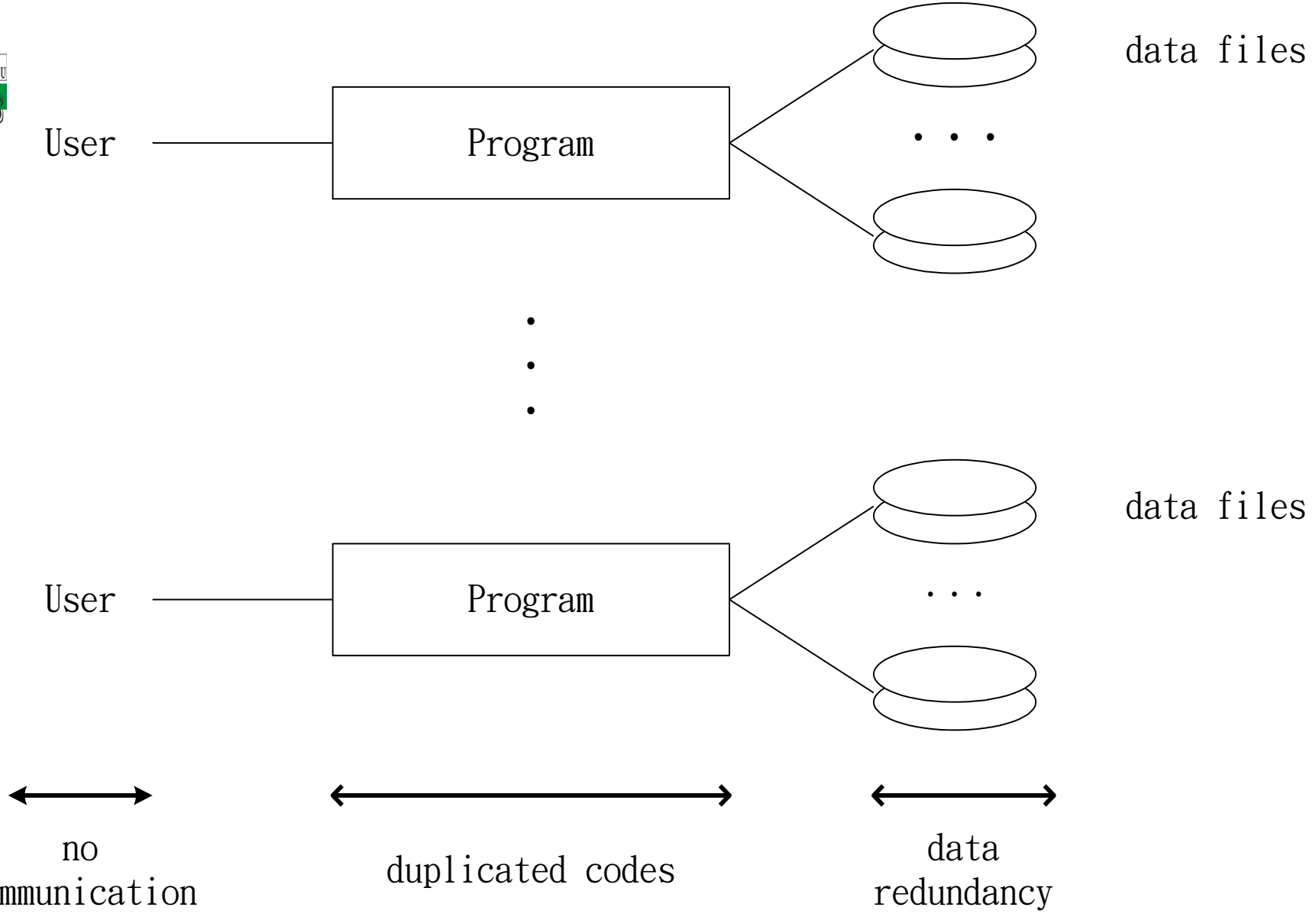



Fig. The Traditional Approach

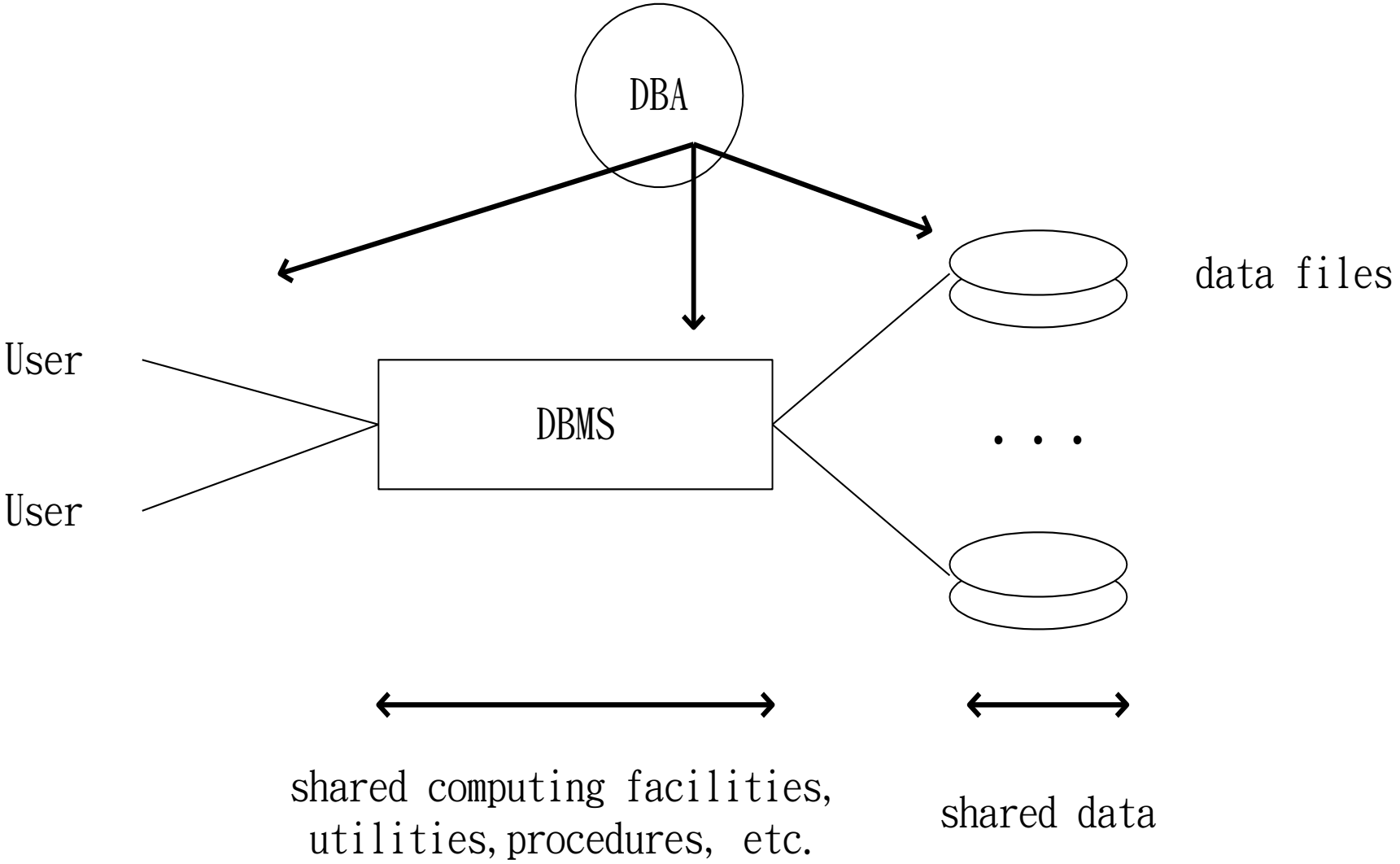



Fig. The DBMS Approach

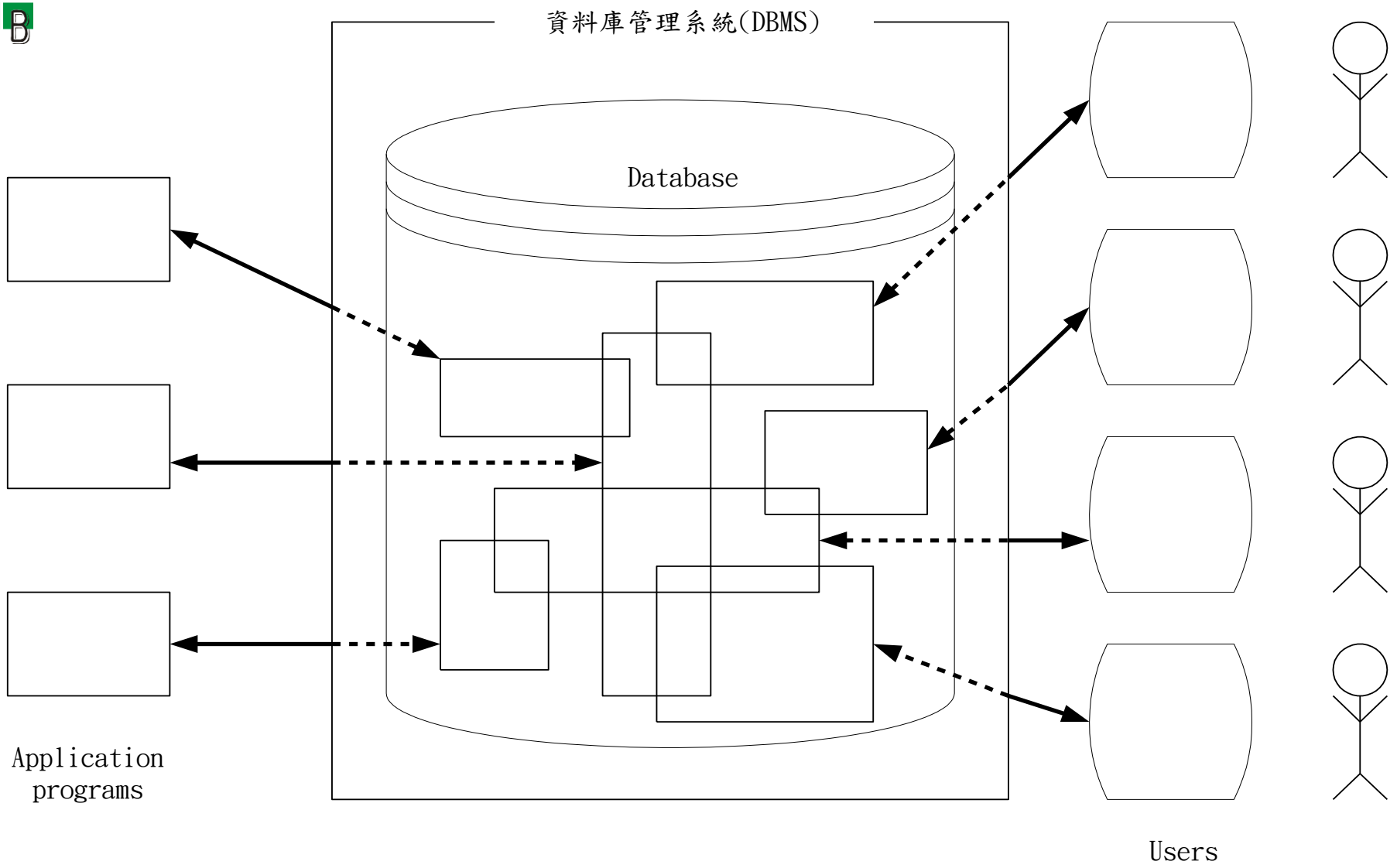



Fig. Simplified picture of a database system

## ❖Data Independence

* The immunity of applications to change in storage structure (and access technique).
  (i.e., changes to the storage structure while the corresponding logical structures do not change.)

## ❖Classes of Database Users

* End Users: The persons that use the database for querying, updating, generating reports, etc. (interface,: menu-driven or command-driven)
* Application Programmer (COBOL, PL/1).
* Database Designers: Responsible for designing the DB.
  Data administrator (DA)

1) Make the strategic and policy decision regarding the data of the enterprise.

2) Decide what data should be stored in the database in the first place, and to establish policies to maintain and deal with data.

* Database Administrators (DBA): Responsible for managing the DBS.

- The technical person responsible for implementing the DA's decision.

# ❖An Architecture for a DB

* Three layers: Internal, Conceptual, External.
* Internal: The way the data is physically stored.
* External: The way the data is viewed by individual users. (individual user view)
* Conceptual: Level of indirection. (community user view) The conceptual view is a representation of the entire information content of the DB.
* Each layer is defined by related schema.

# ❖DB Schema

* Database Schema: The description of a database.

  Includes descriptions of the database structure and the constraints that should hold on the database.

* Database Instance: The actual data stored in a database at a particular moment in time. Also called database state (or occurrence).

* The database schema changes very infrequently. The database state changes every time the database is updated. Schema is also called intension, whereas state is called extension.

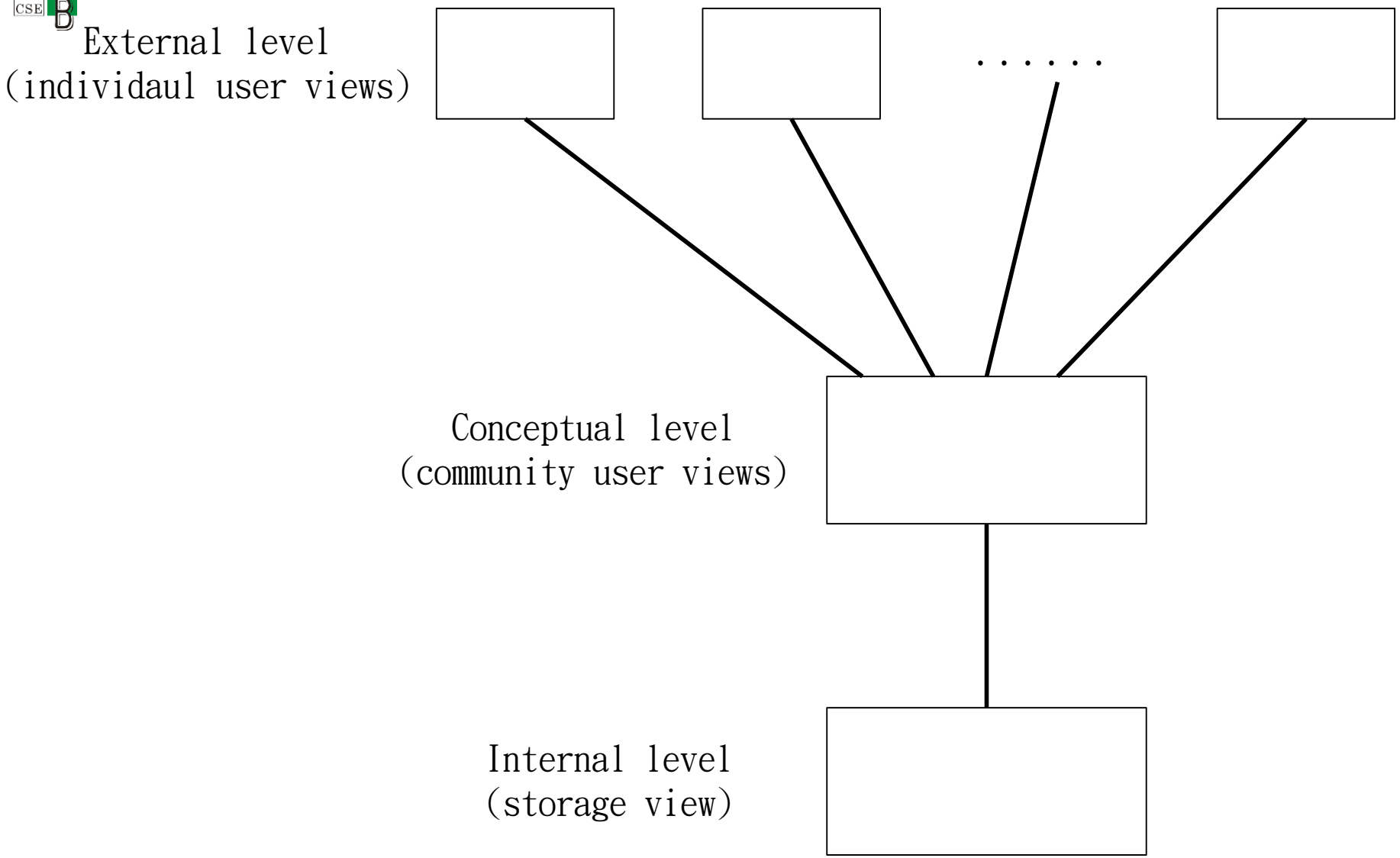



Fig. Database Architecture

External Schema

Male

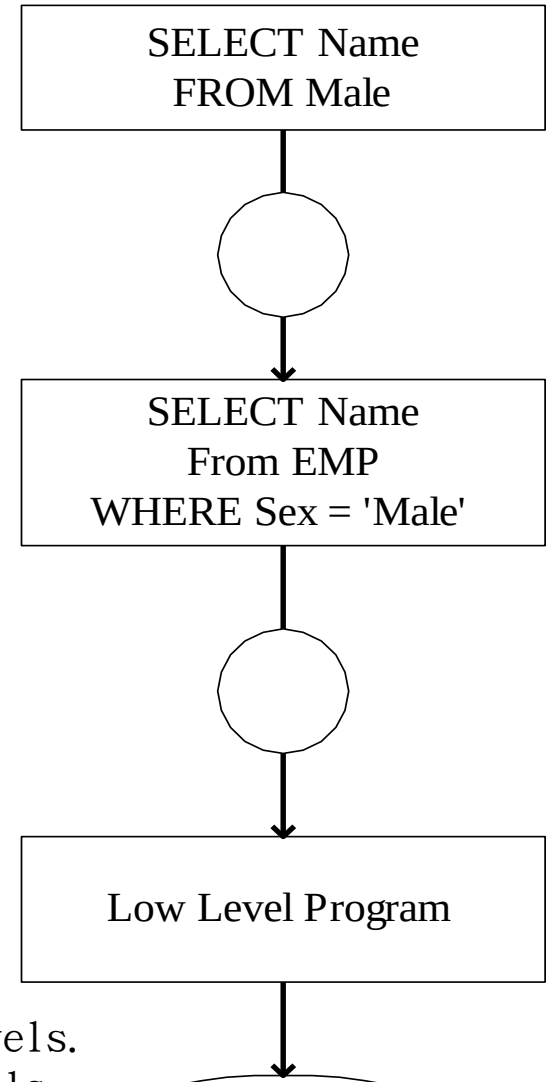
| ID | Name |
|---|---|
| | |

Conceptual Schema

EMP

| SSN | Name | TEL | Sex |
|---|---|---|---|
| | | | |

Internal Schema




1. The 'same' object may have different names at different levels.
2. Objects at a lower levels need not appear at a higher levels.
3. Implementation details appear at the internal level only.

Fig. an example of database architecture

# ❖DBMS Schemas at Three Levels:

--Internal schema at the internal level to describe data storage structures and access paths.
(what index exists, how stored fields are represented, what physical sequence the stored records are in.)

--Conceptual schema at the conceptual level to describe the structure and constraints for the whole database.
(Security checks and integrity checks.)

--External schemas at the external level to describe the various user views.
(definitions of each of the various types of external record in that external view).

* **Logical Data Independence**: The capacity to change the conceptual schema without having to change the external schemas and their application programs.

* **Physical data independence**: The capacity to change the internal schema without having to change the conceptual schema.

* **Mapping**: Whenever the lower level is changed, the mapping also must be changed such that the higher level do not have to be changed.

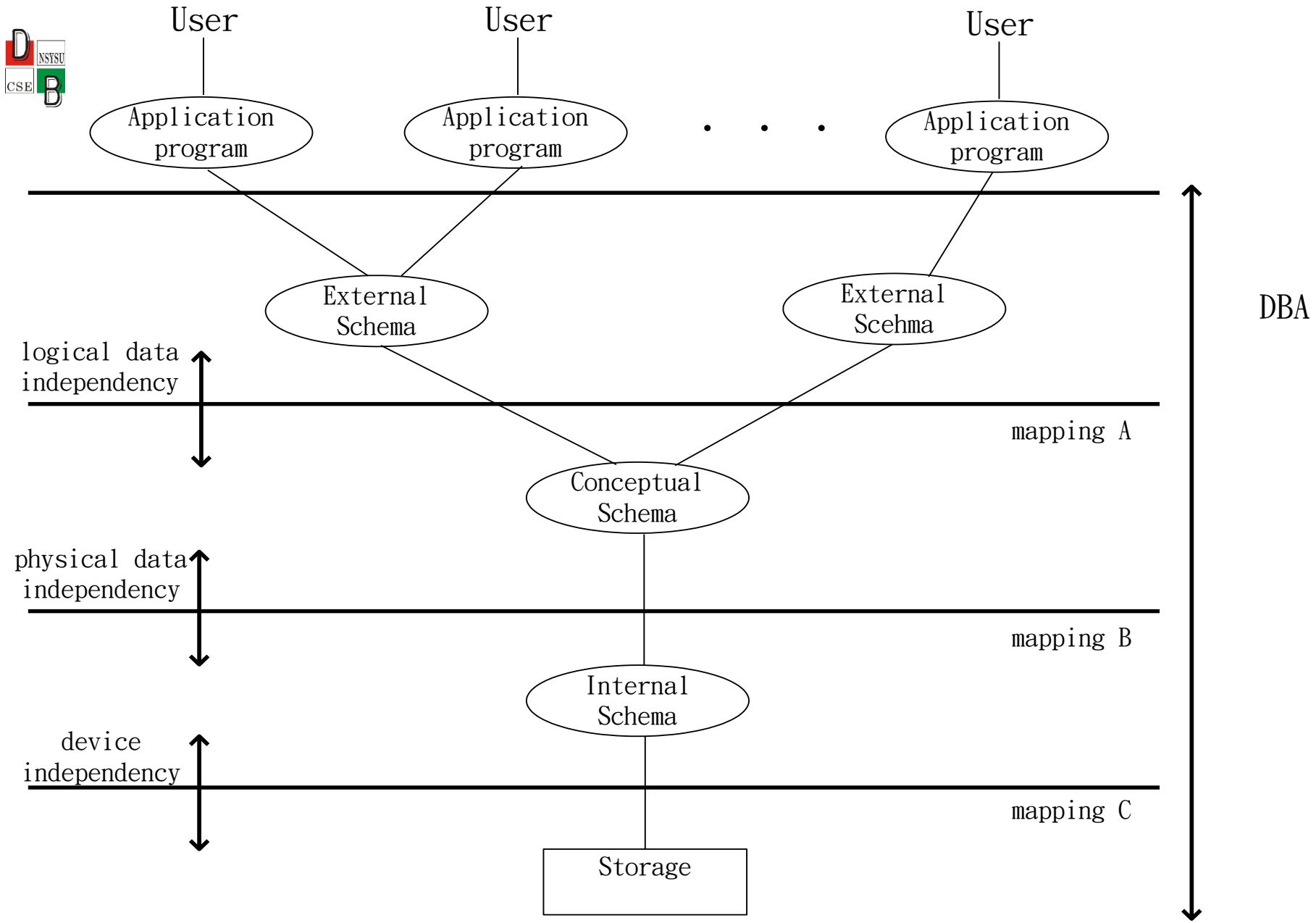



Fig. DBMS Architecture

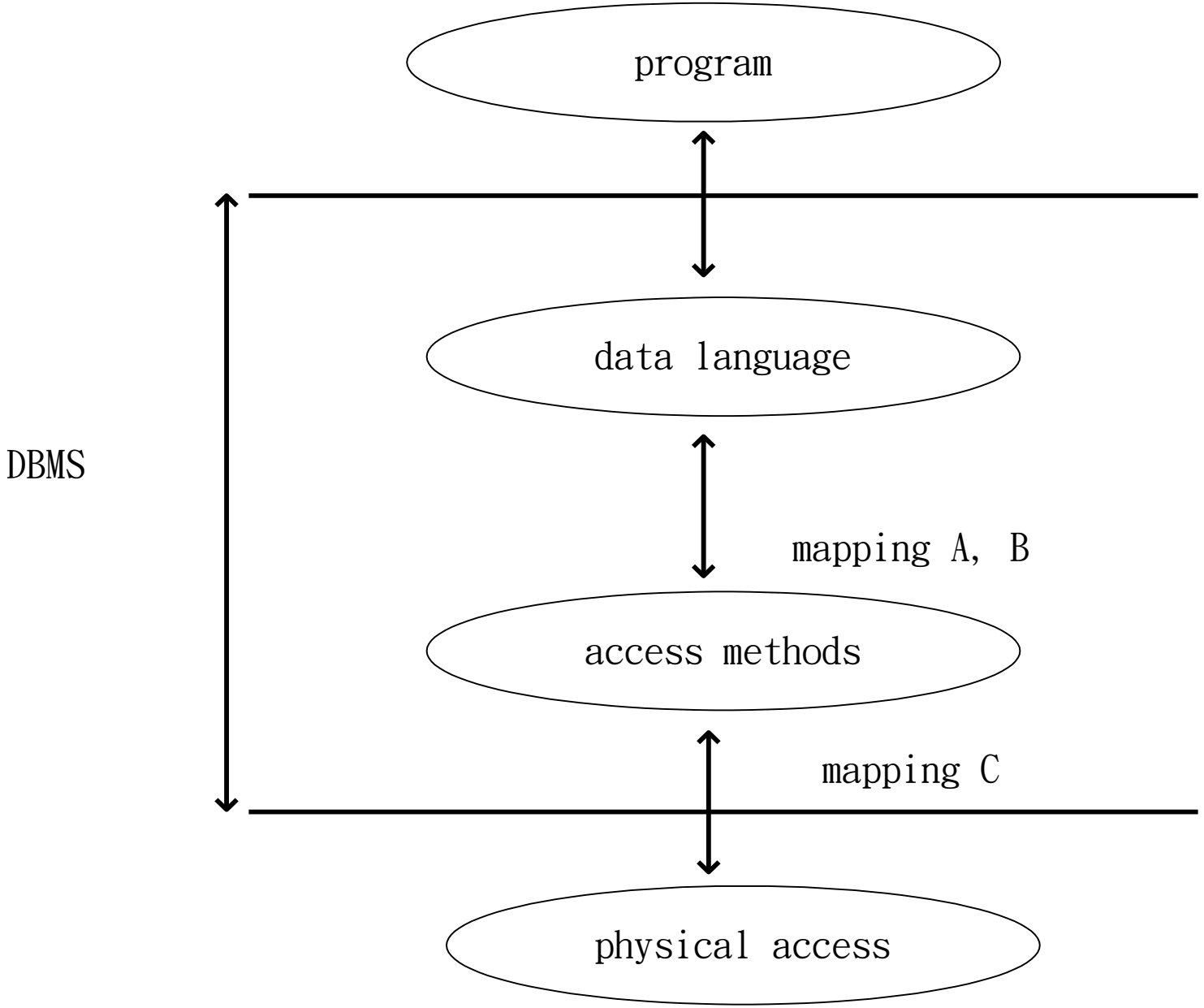



Fig. Data Access

**External (Pascal)**

```
Var
   EMP# : CHAR(6);

   SAL: INTEGER;
```

**External (COBOL)**

```
01 EMPC,
   02 EMPNO PIC X(6),

   02 DEPTNO PIC X(4).
```

**Conceptual**

```
EMPLOYEE
  EMPLOYEE_NUMBER CHARACTER(6)
  DEPARTMENT_NUMBER CHARACTER(4)

  SALARY NUMBER
```

**Internal**

```
 STORED_EMP LENGTH = 20
 PREFIX   TYPE=BYTE(6), OFFSET=0
 EMP#     TYPE=BYTE(6), OFFSET=6, INDEX=EMPX
 DEPT#    TYPE=BYTE(4), OFFSET=12
 PAY      TYPE=FULLWORD, OFFSET=16
```

Fig. An example of database architecture

# ❖ Jobs of a DBA

1) define the conceptual schema. (decide exactly what information is to be held in the DB).
2) define the internal schema. (how the data is represented in the stored database).
3) liaising with users.
4) define security and integrity checks.
5) define backup and recovery structure procedures.
6) monitor performance and responding to changing requirements.

# ❖DB Languages

* Data Definition Language (DDL): Used by the DBA and database designers to specify the conceptual schema of a database.
* Data Manipulation Language (DML): Used to specify database retrievals and updates.

--DML commands (data sublanguage ) can be embedded in a general-purpose programming language (host language), such as COBOL, PL/1 or PASCAL.
--Alternatively, stand-alone DML commands can be applied directly (query language).

# Data Sublanguages

- -The host language is responsible for providing various nondatabase facilities. (if.. then)
- 
- Example. (PL/1 + SQL)

```
DCL Givens # char(5);
DCL Rank fixed bin(15);
DCL City char(15);
....
Exec SQL Declare S Table
                    (S# char(5) not null,
                   Sname char(20),
                     Status smalllist,
                     City char(15));
Exec SQL Include SQLCA;
....
If  A > B then
getstc:
Exec SQL select Status, City
        into : Rank,   : City
        from  S
        where  S# = :Givens;
.....
Put skip list (Rank,City);
```

# DBMS

- Is the software that handles all access to the database.

- Provide the user interface to the database system.

1) user issues an access request (by data sublanguage).
2) DBMS intercepts that request and analyze it.
3) DBMS, external schema --> mapping --> conceptual schema --> mapping --> storage structure definition.
4) DBMS executes the necessary operations on the stored database.

# ❖ Functions:

1) Data definition: accept data definition in source form and convert them into the appropriate object form. (language processor for data definition language--DDL).
2) Data manipulation: handle requests, data manipulation language processor--DML.
3) Data security and integrity: monitor user requests and reject any attempts to violate the security and integrity checks defined by DMA.
4) Data Recovery and concurrency. (transaction manager)
5) Data dictionary. data about data (cross-reference).
6) Performance. (efficiency).

# ❖ Objects of Database Design

1) easy to use.
2) easy to maintain.
3) cost effective.

# ❖Research Areas of DBS

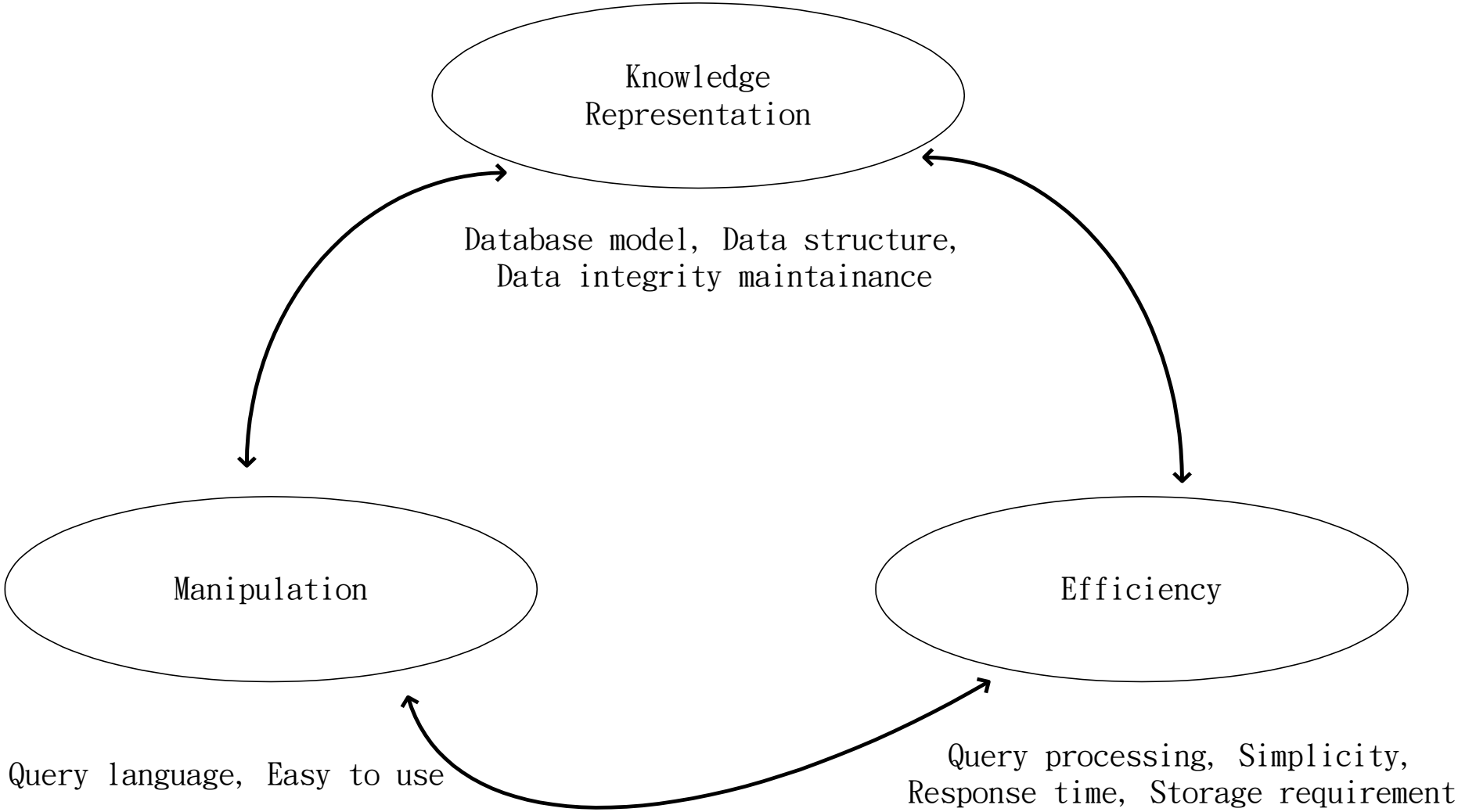



Fig. Research areas of DBMS

# ❖Distributed DBS

- A database that is not stored in its entirely at a single physical location, but rather is spread across a network of computers that are geographically dispersed and connected via communication links.
- It should look like a centralized system to the user.