



Nguyen Tat Thanh Institute of  
International Education (NIIE)

# **DATABASE MANAGEMENT SYSTEMS**

**(Credits 3)**

**MSc. Luong Tran Ngoc Khiet**  
**May - 2021**

## **Chap 1. Overview**

Chap 2. Data storage management

Chap 3. Programming with Cursors

Chap 4. Query optimization

Chap 5. Continuous transaction processing



# Chap 1. Overview

MSc. Luong Tran Ngoc Khiet

NTT Institute of International Education (NIIE)

## 1. Database (DB)

### 1.1 Introduction

### 1.2 Development history

### 1.3 Features of the DB approach (Đ/đ cách tiếp cận CSDL)

### 1.4 The benefits of the DB approach (Lợi ích cách tiếp cận CSDL)

## 2. Database management system (DBMS) – Hệ QT CSDL

### 2.1 Data model types of DBMS

### 2.2 Classification of DBMS

### 2.3 Architecture of DBMS

# 1. Database (DB)

- 2<sup>nd</sup> years, course **Database Design & Development**

- What did you know?

## Relational

Tend to be larger,  
monolithic

ORACLE



## Non-relational

Newer field, lots  
of players



splunk>



Solr



## 2. Database management system



### 2.1 Data model types of DBMS:

- The data model describes how data is organized within the database.
- The data model also describes the data relationships and constraints defined on that data

- Hierarchical data model
- Network data model
- Relational data model
- Hybrid entity model
- Object-oriented data model

### Hierarchical data model example

Level 1:

KQua	
DiemTH	DiemLT

Level 2:

HPhan	
TenHP	SLuong

SVien		
TenSV	Lop	Nganh

Level 3:

MHoc		
TenMH	Khoa	TinChi



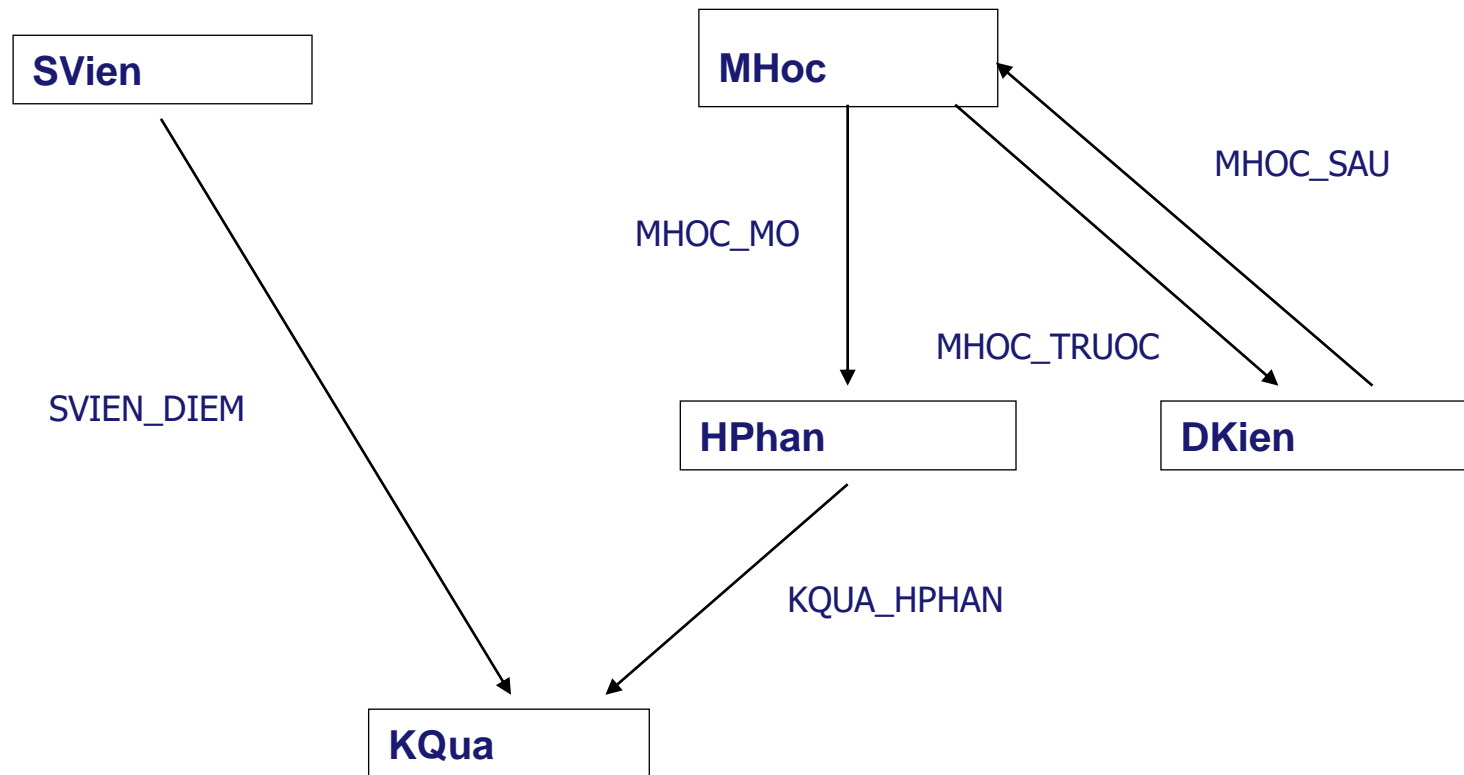
### Hierarchical data model

- The model is a tree.
- Each node of the tree represents an entity
- The child node and parent node are related to each other according to a certain relationship

### Hierarchical data model

- ❑ Type of news: Characterizes a particular type of object
- ❑ Relationship type: separation relationship type
  - ❑ Relationship between master record and member record 1:n
  - ❑ The relationship between member records and master records is 1:1
  - ❑ There is only one relationship between two pieces of information

### Network model example



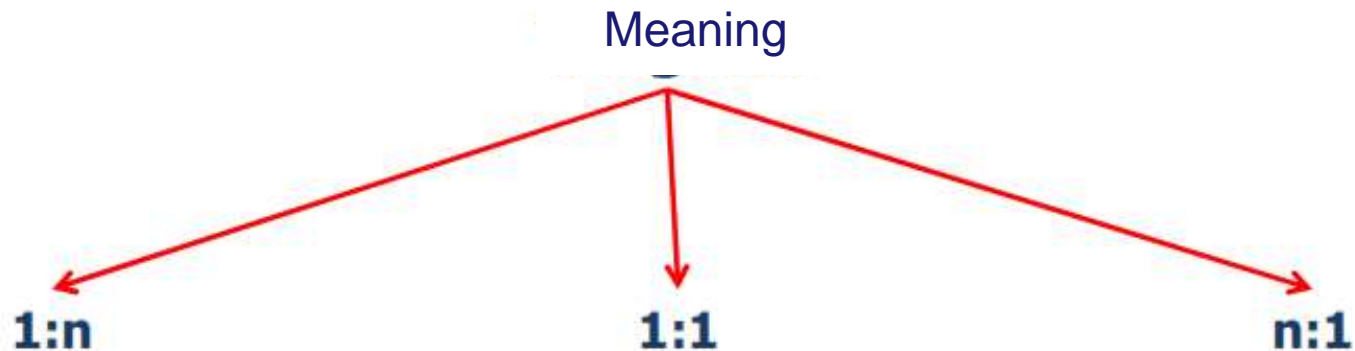
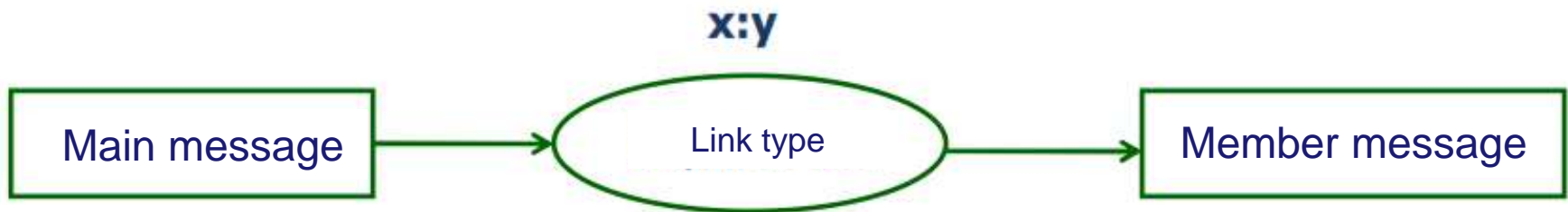
### Network data model

- ❑ Record type: Characterizes a particular type of object
- ❑ Record: Is a representation of a type of record
- ❑ Set type: The association between the master record type and the member record type

## 2.1 Types of models



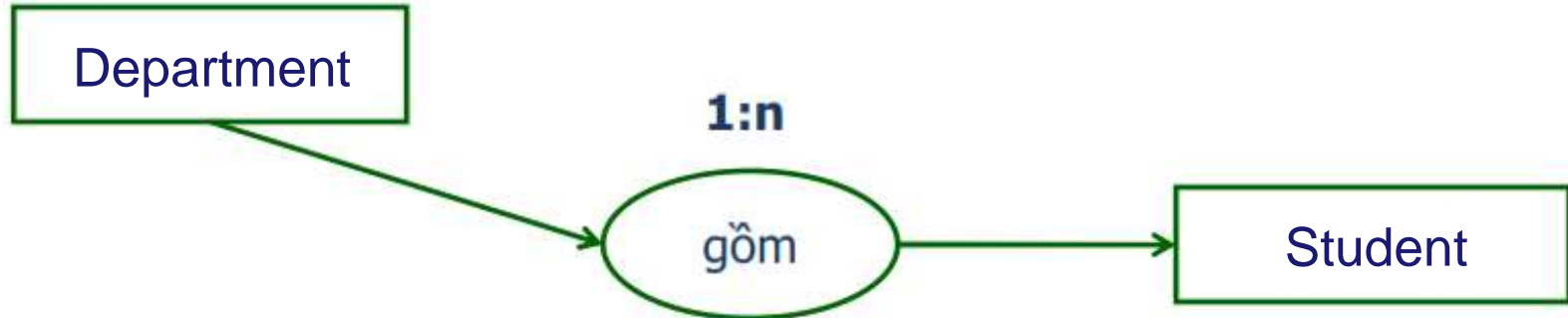
### Network data model



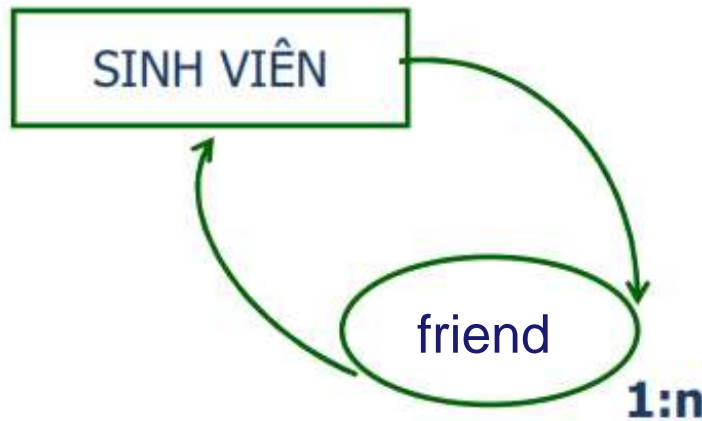
# The downside



## Network data model



A department has many students



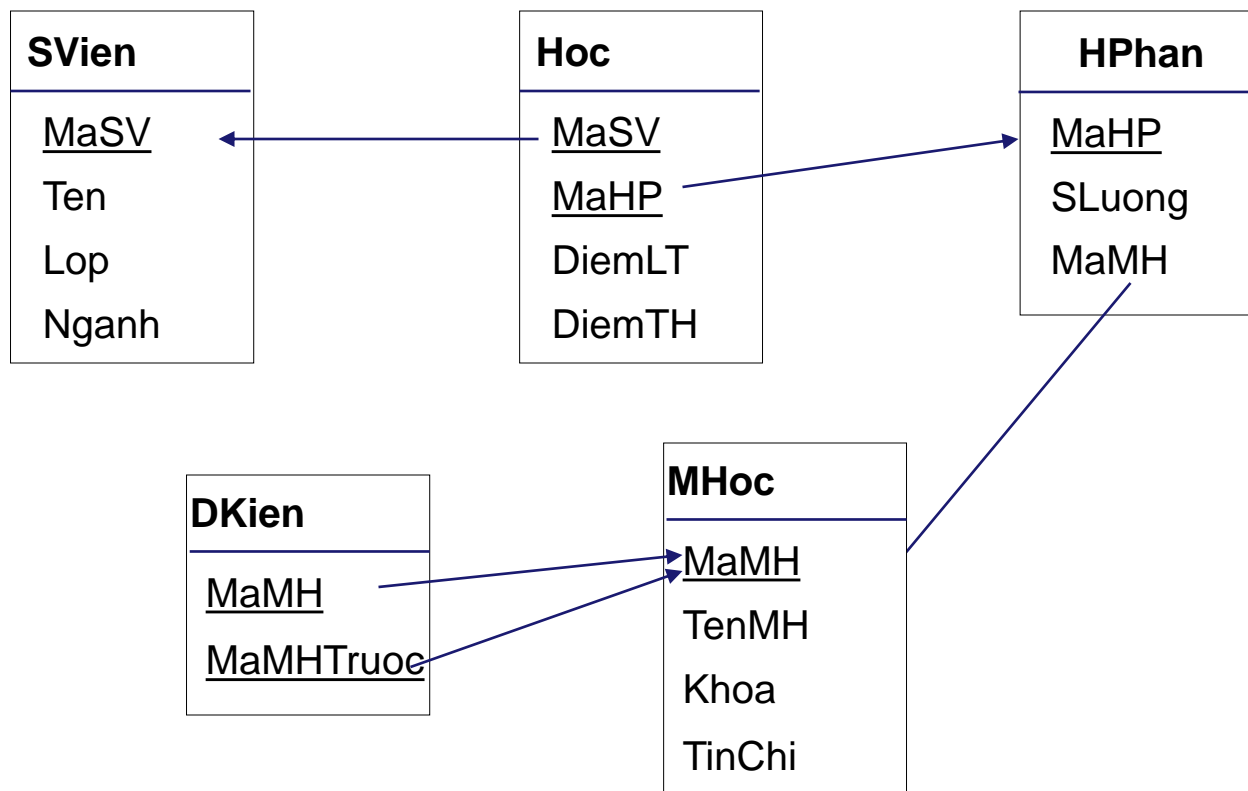
Students can have many friends who are students

### **Hierarchical data model**

### **Network data model**

- ❑ The ability to express semantics is poor compared to the complex relationships of data in reality
- ❑ Low-level representation model close to storage level: difficult when application environment changes

### Relational data model example



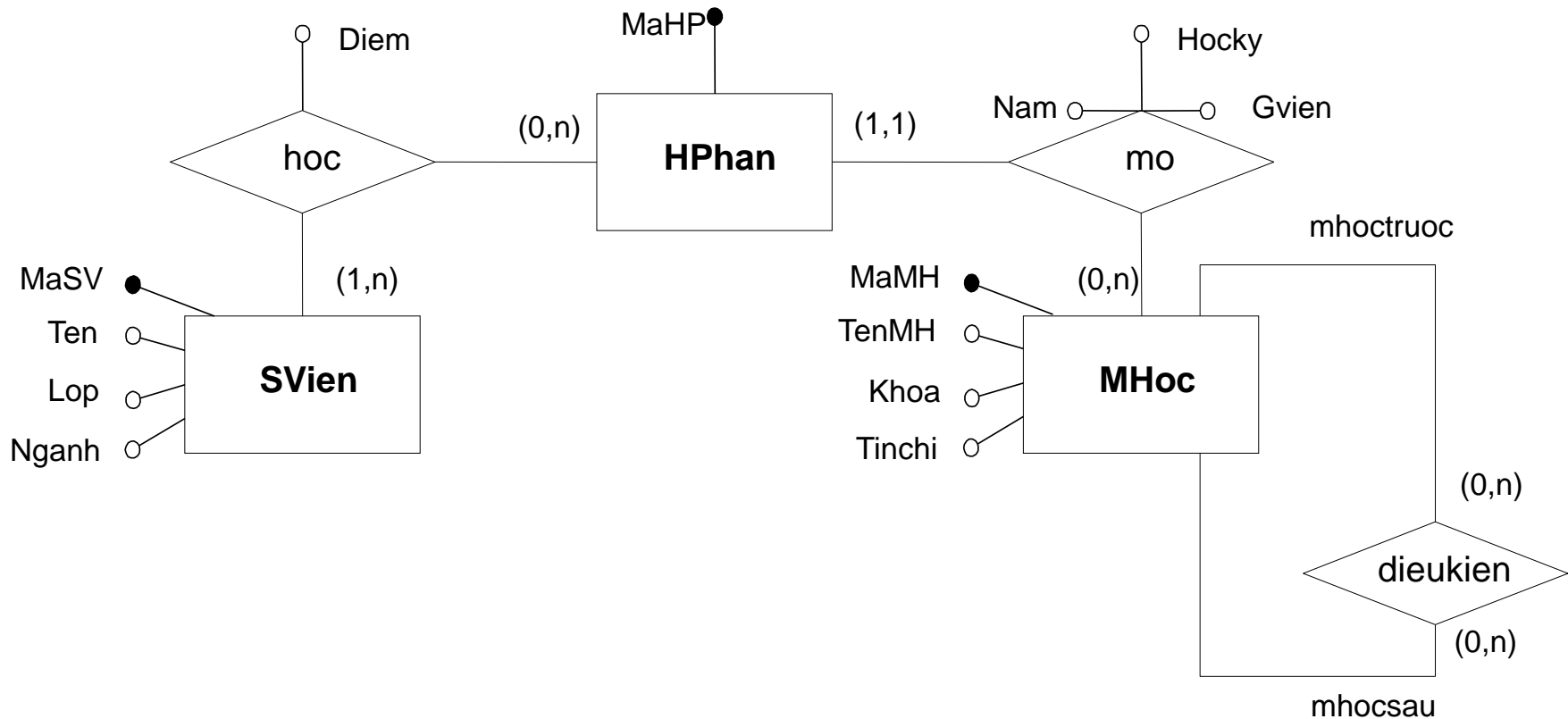


### Relational data model

- Properties
- Relational schema
- Set
- Relationship
- Key

<b>SVien</b>
<u>MaSV</u>
Ten
Lop
Nganh

### Associative entity data model example

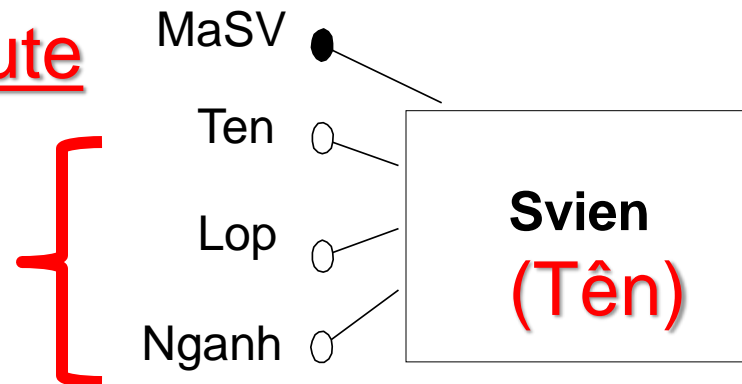


### Entity relationship model

- Entities are objects, places, people... that need to store information.
- The entity is named and drawn in a rectangle.

- Key attribute

- Attribute



### **Entity relationship model**

- There are two types of entities: independent entities and dependent entities.
- A dependent entity also known as a weak entity is an entity whose existence depends on another entity.
- Weak entities are described by rectangles whose outer edges are double edges.

### Entity relationship model

❑ A relationship represents a collaboration between two entities, represented by a "**diamond**" in the middle containing the relationship name.

❑ Relationships can exist on 2 entities or on the same entity. There can be many relationships between two entities.

❑ Relationship: **1 – 1 | 1 – n | n - n**

### Object oriented data model

Based on an object-oriented approach that includes concepts

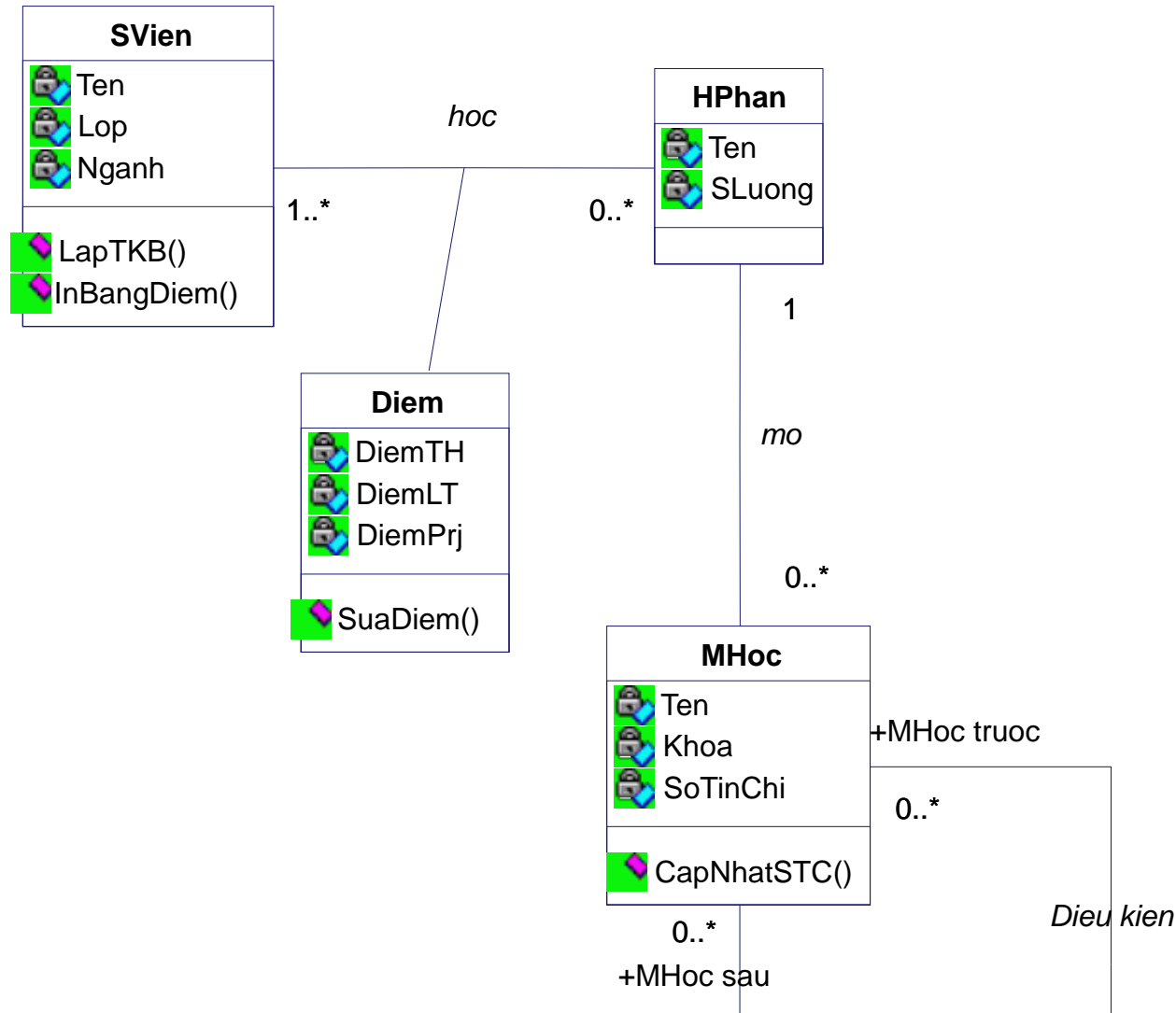
- Lớp (class)
- Kế thừa (inheritance)
- Kế thừa bội (multi-inheritance)
- Tính đóng gói (encapsulation)
- Tính đa hình (polymorphism)
- Tái sử dụng (reuse)

**Basic features**

## 2.1 Types of models



### Example object-oriented data model



### 2.2 Classification of DBMSs

#### Classification criteria

- ☐ Data model
- ☐ Number of users
- ☐ Station number
- ☐ Target



- ❑ Based on the data model
  - Hierarchical data model
  - Network data model
  - Relational data model
  - Hybrid entity model
  - Object-oriented data model

### □ Based on number of users

– Single user

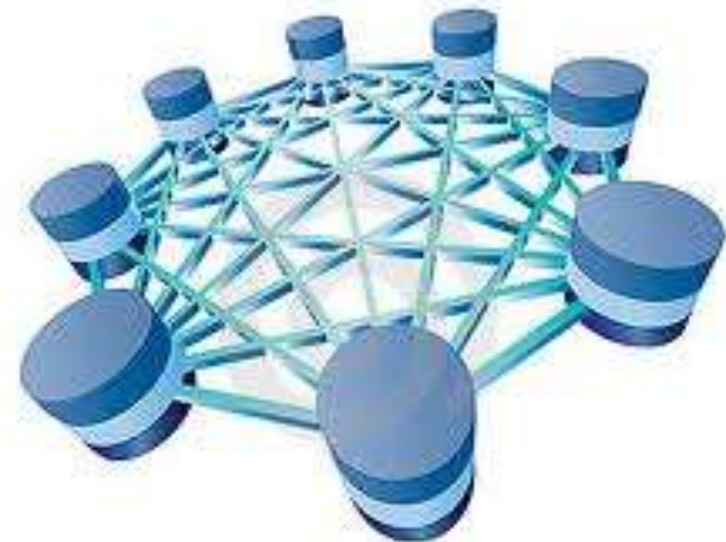


– Multiple user



### ❑ Based on station number

- Centralized database (on 1 computer and 1 database)
- Distributed database (multiple computers, multiple databases)



### ❑ Based on intended use

- General purpose (OLTP)
- Special purpose: airline ticket booking, phone directory system...DBMS needs to support large concurrency dispute resolution functionality
- Data analysis

### **2.3.1. Three-Schema architecture (3-schema architecture)**

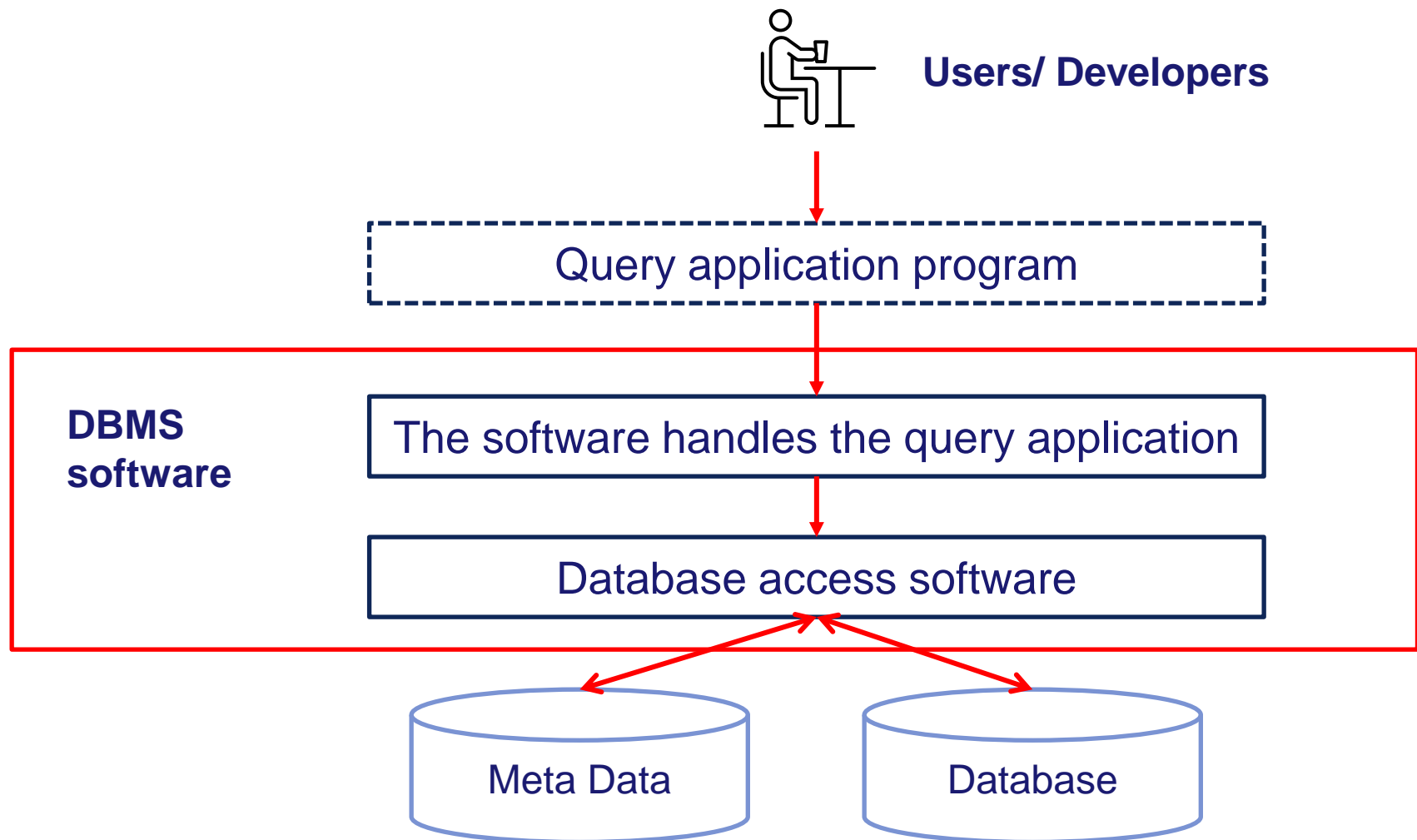
### **2.3.2 Organizing the database system**

### **2.3.3 Modules of DBMS**

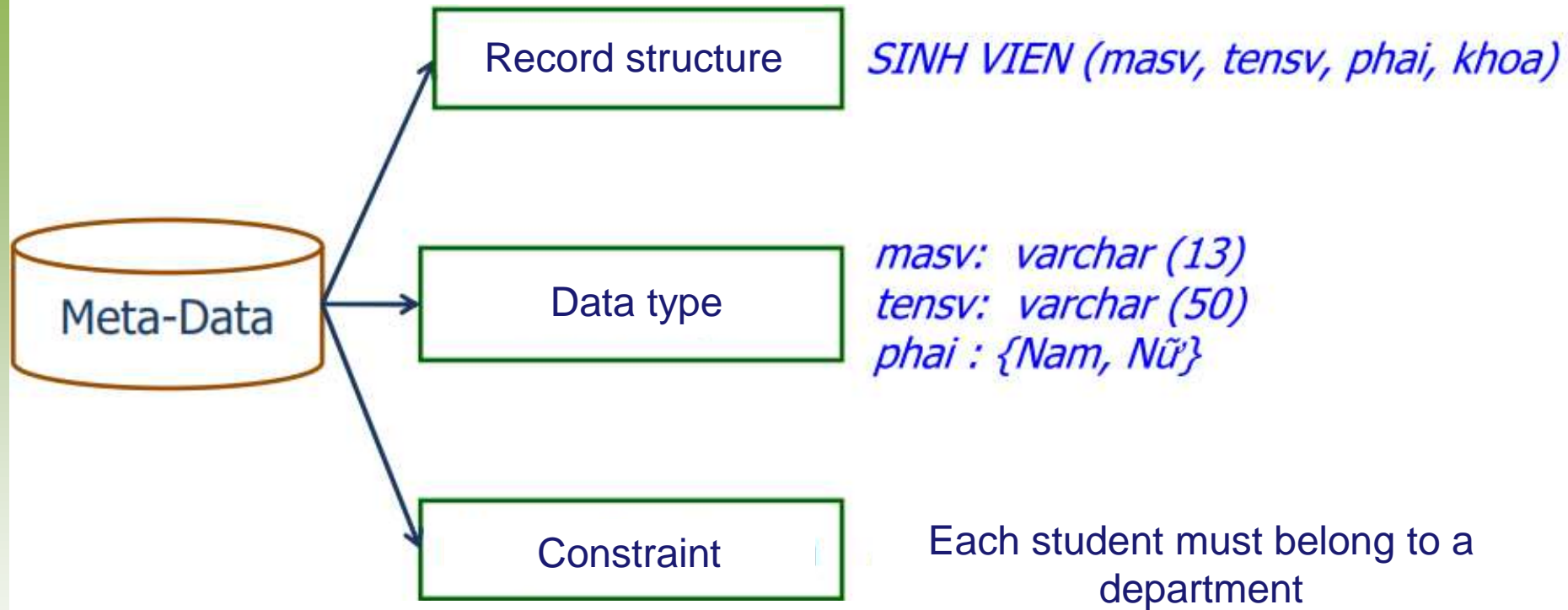
## 2.3 Architecture of DBMS



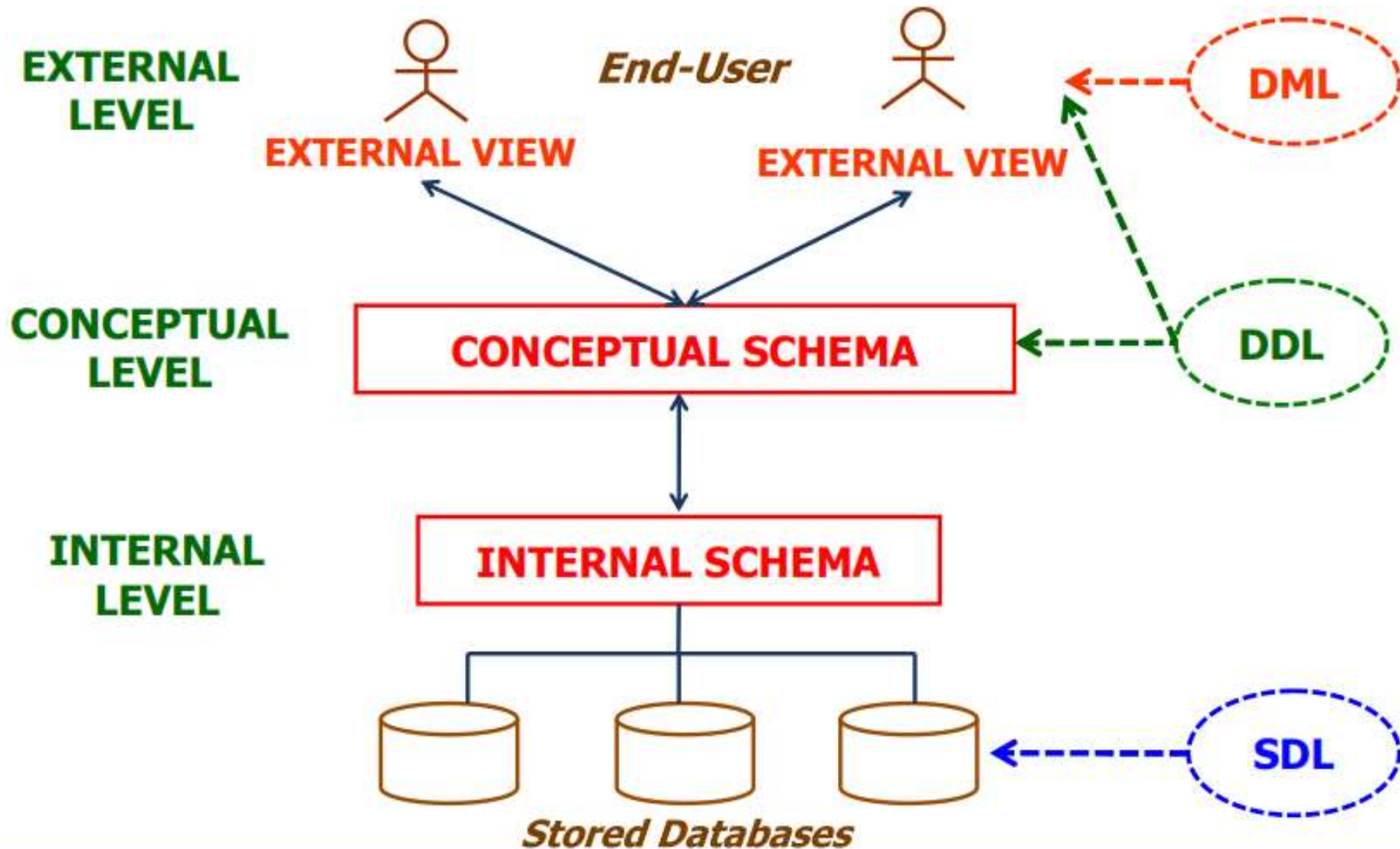
**Database system = Database + DBMS**



## 2.3 Architecture of DBMS



## 2.3.1. Three-Schema

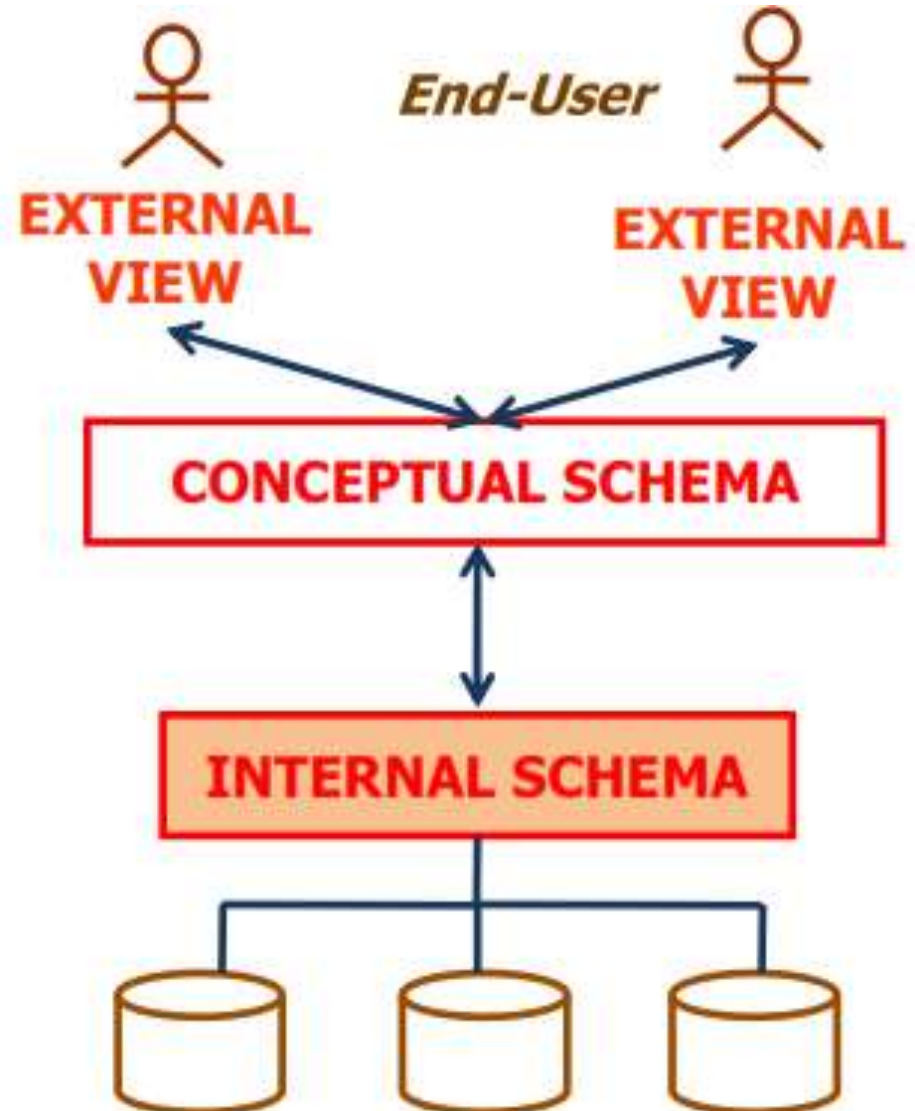




### Internal Level

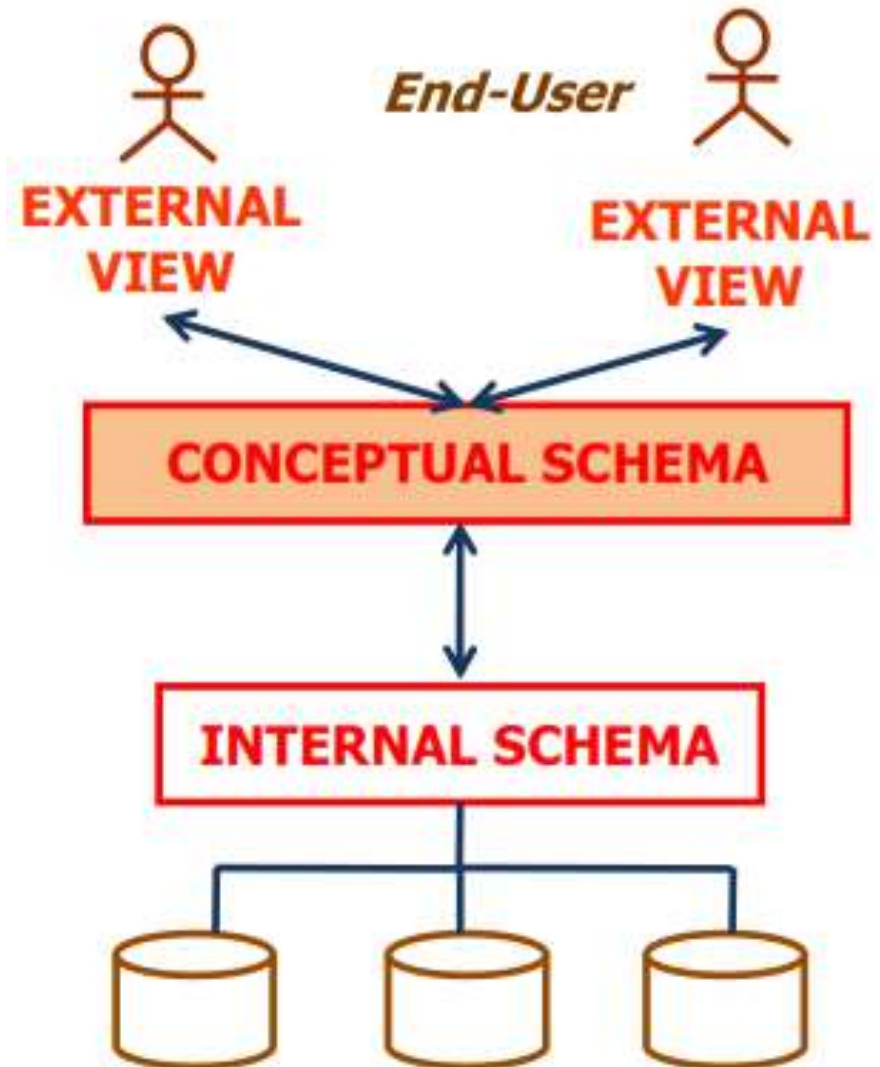
Describe the physical organization of the database

- Describe the storage structure used for efficient information retrieval
- Use physical data schema



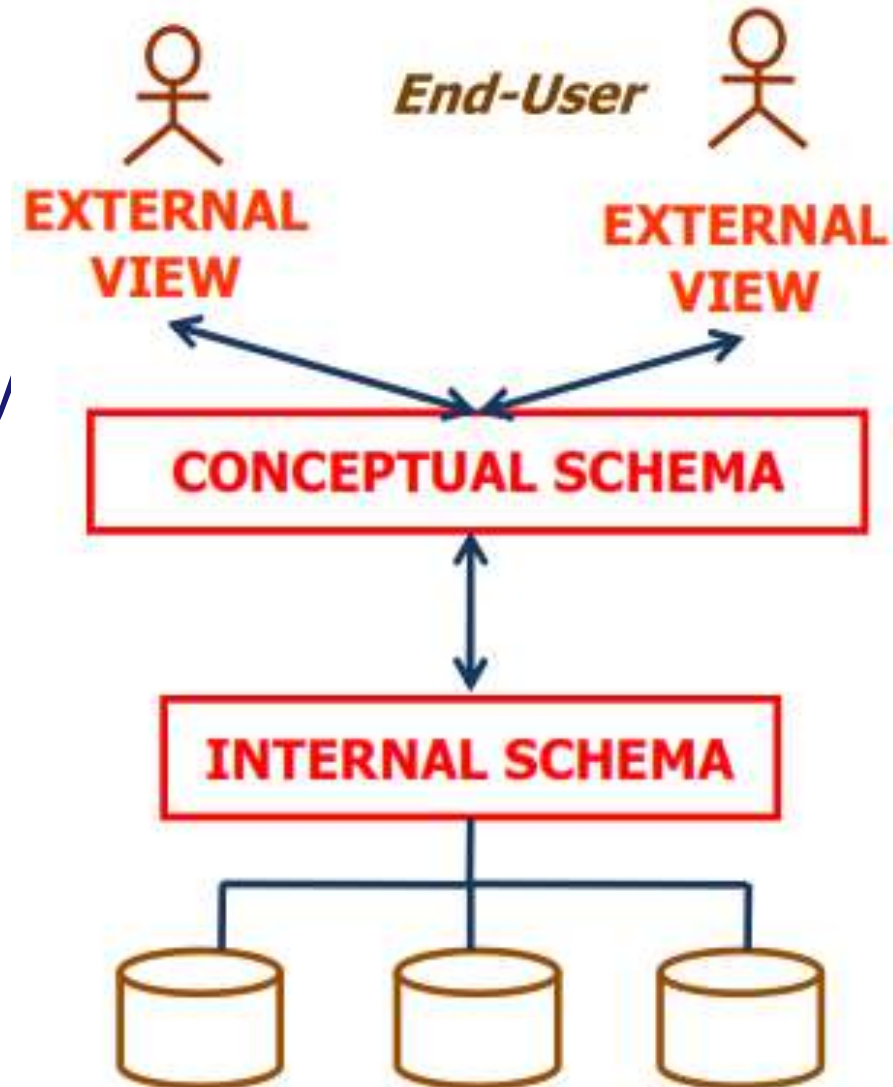
### Conceptual Level

- Modeling the real world
- Define structures and data types
- Use the conceptual data model (Conceptual Schema)

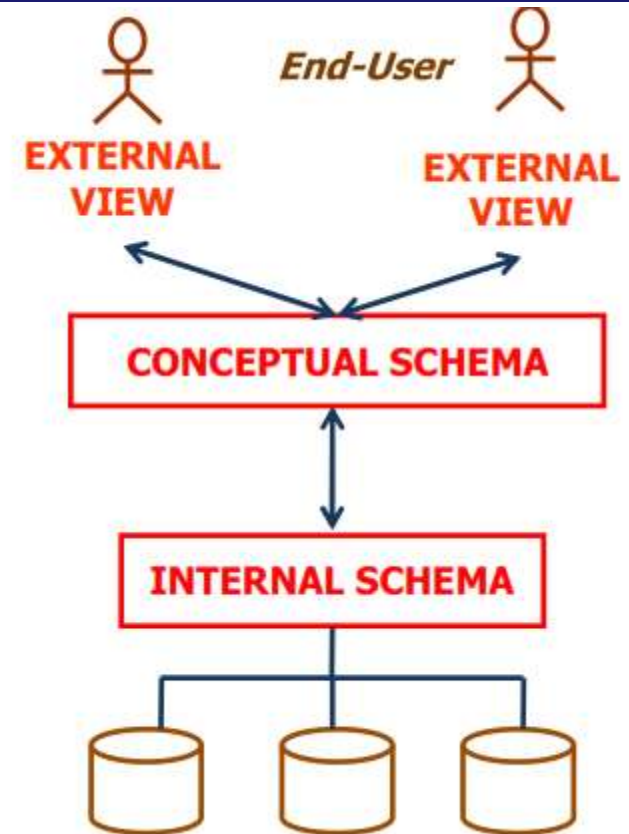
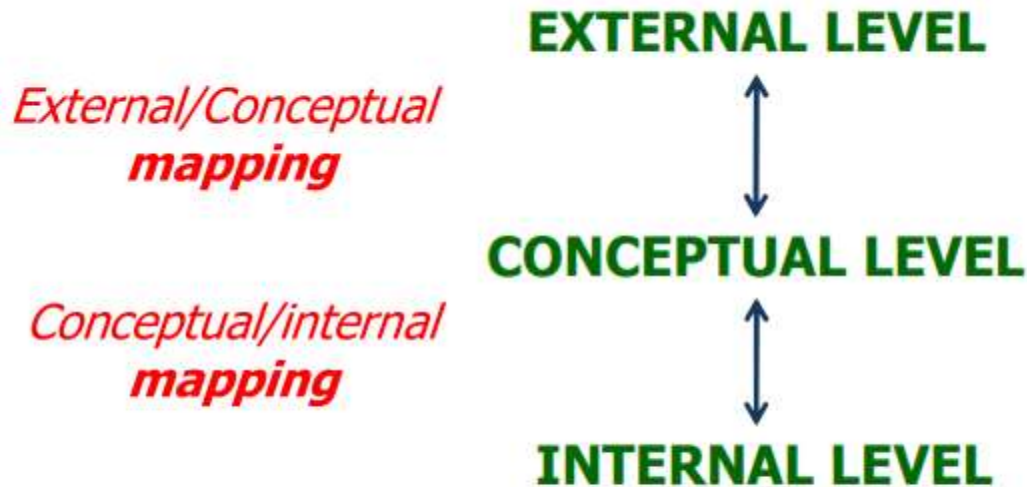


### External Level

- Describes the end user's external views
- Ensure data security: only authorized people can access relevant data



## 2.3.1. Three-Schema



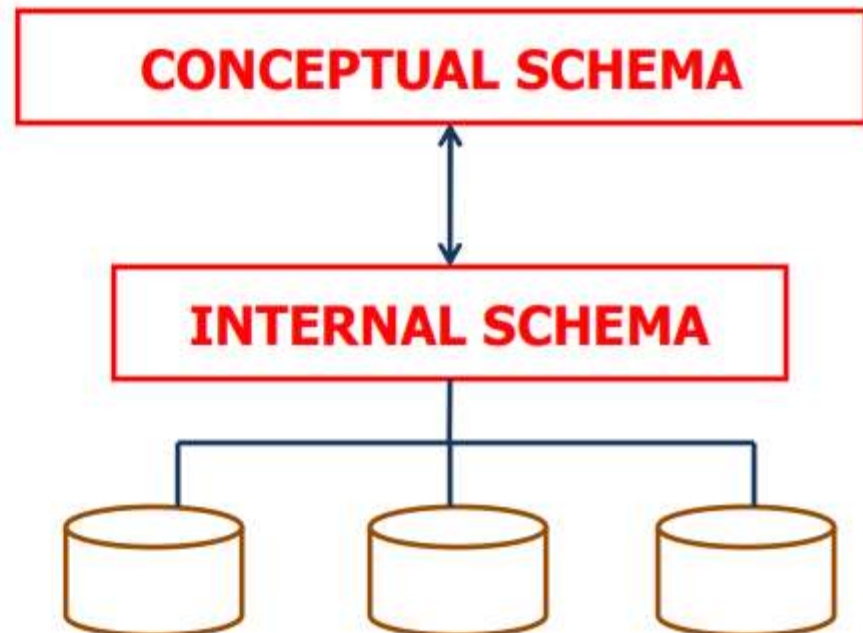
### Ảnh xạ (Mapping)

The process of conveying requests and returned results between levels

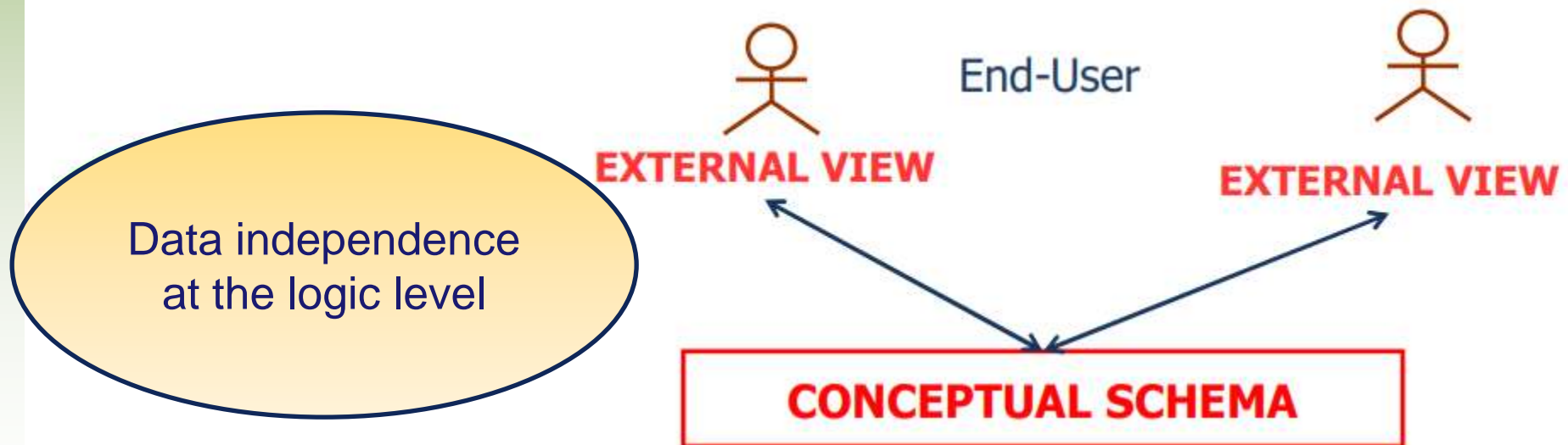
Three-Schema architecture demonstrates  
data independence:

- Data independence at the physical level
- Data independence at the logical level

Reorganizing or changing the physical database only changes the computational efficiency of the application program without requiring rewriting the program.



Changing the conceptual schema such as adding information, changing constraints, adding entities... does not change or affect the application program.





## 2.3.2 Organize the DB system



Centralized  
DBMS

Client/Server

Three-Tier  
Client/Server





### Centralized DBMS

1 computer all included:

- DBMS (SQL Server, Oracle, ...)
- Deployment software + utilities (.NET, ...)
- Application programs (Education management, ...)

### **Client/Server Architecture**

Client connects to server (LAN, WAN) to participate in performing a number of functions:

- Communication between user and system
- Send a request to access resources on the server

### **Client/Server Architecture**

Server performs specialized functions:

- File Server
- Printer Server
- Web Server
- Email Server

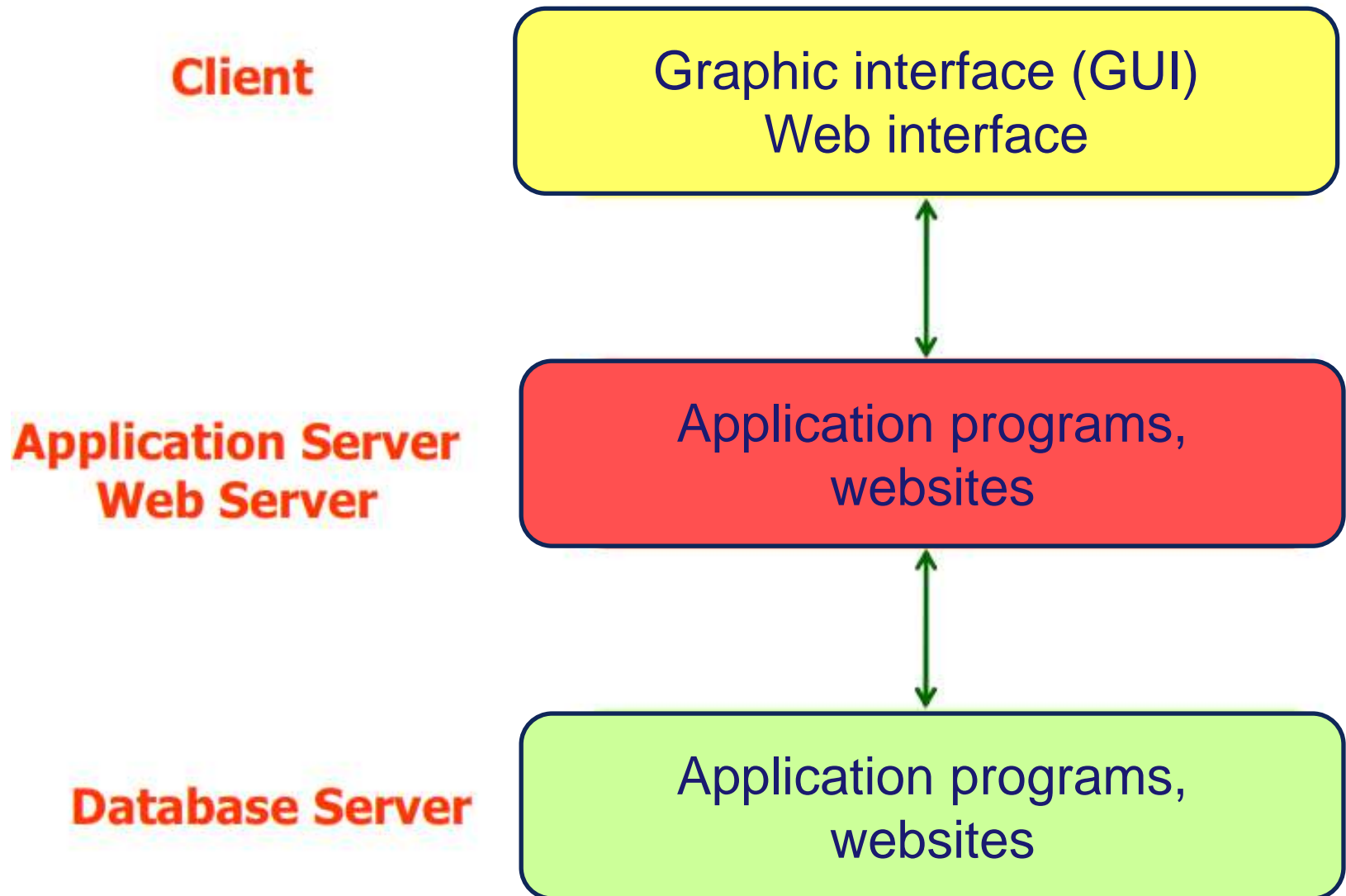
### Client/Server Architecture

- ❑ **ODBC (Open Database Connectivity):**  
provide API (Application Program Interface) that allows Clients to call and execute applications on HQT-DB
- ❑ From Client can connect to many DBMSs
- ❑ Most HQT-DBs today provide ODBC drivers

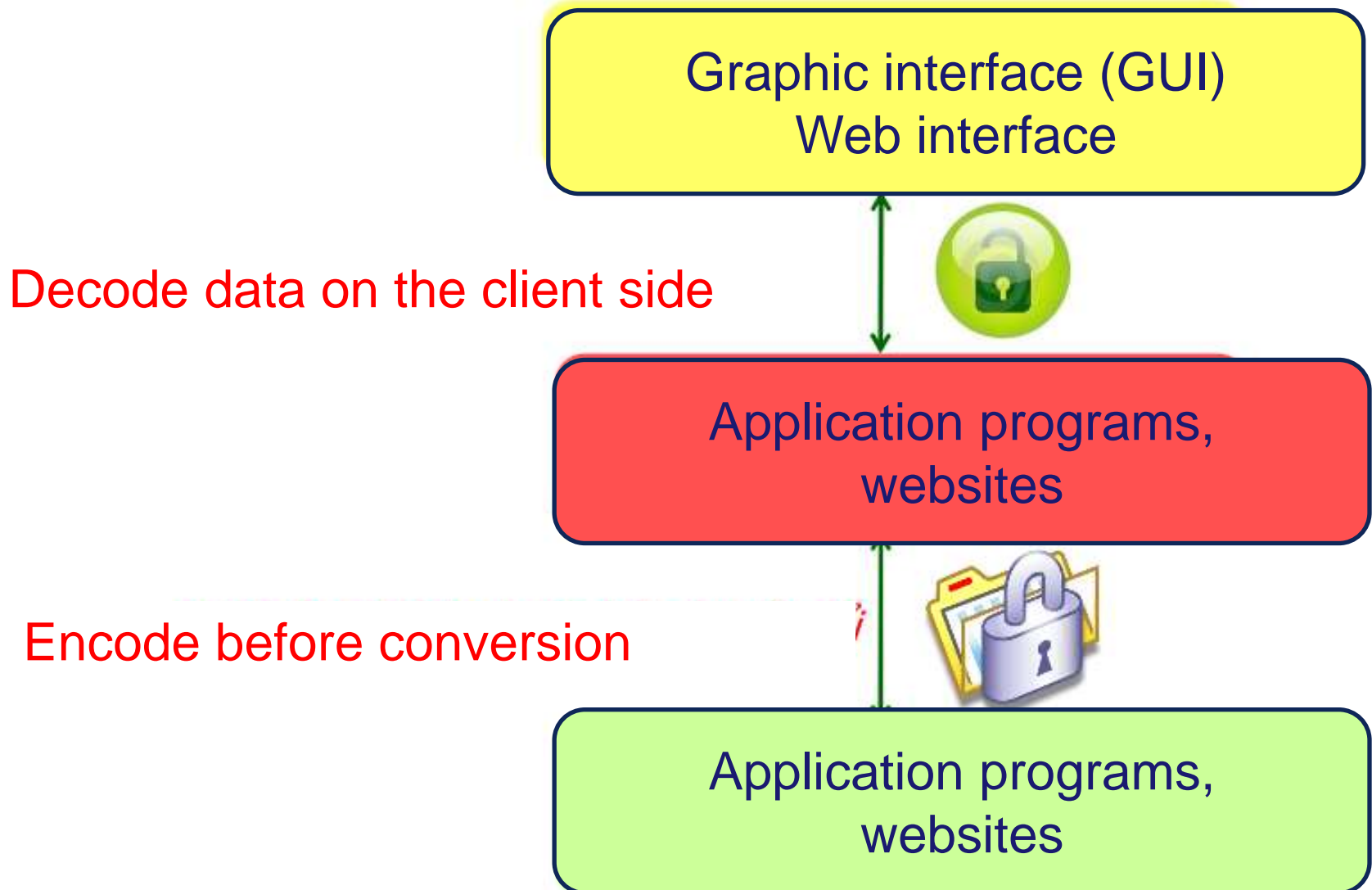
### **3-layer Client/Server architecture**

- ❑ Popular for Web applications
- ❑ Adding a middle layer (intermediate) called  
Application Server or Web Server
  - Contains connection software, data binding rules, etc.
  - Acts as an intermediary bridge between information conversion from client to server

### 3-layer Client/Server architecture



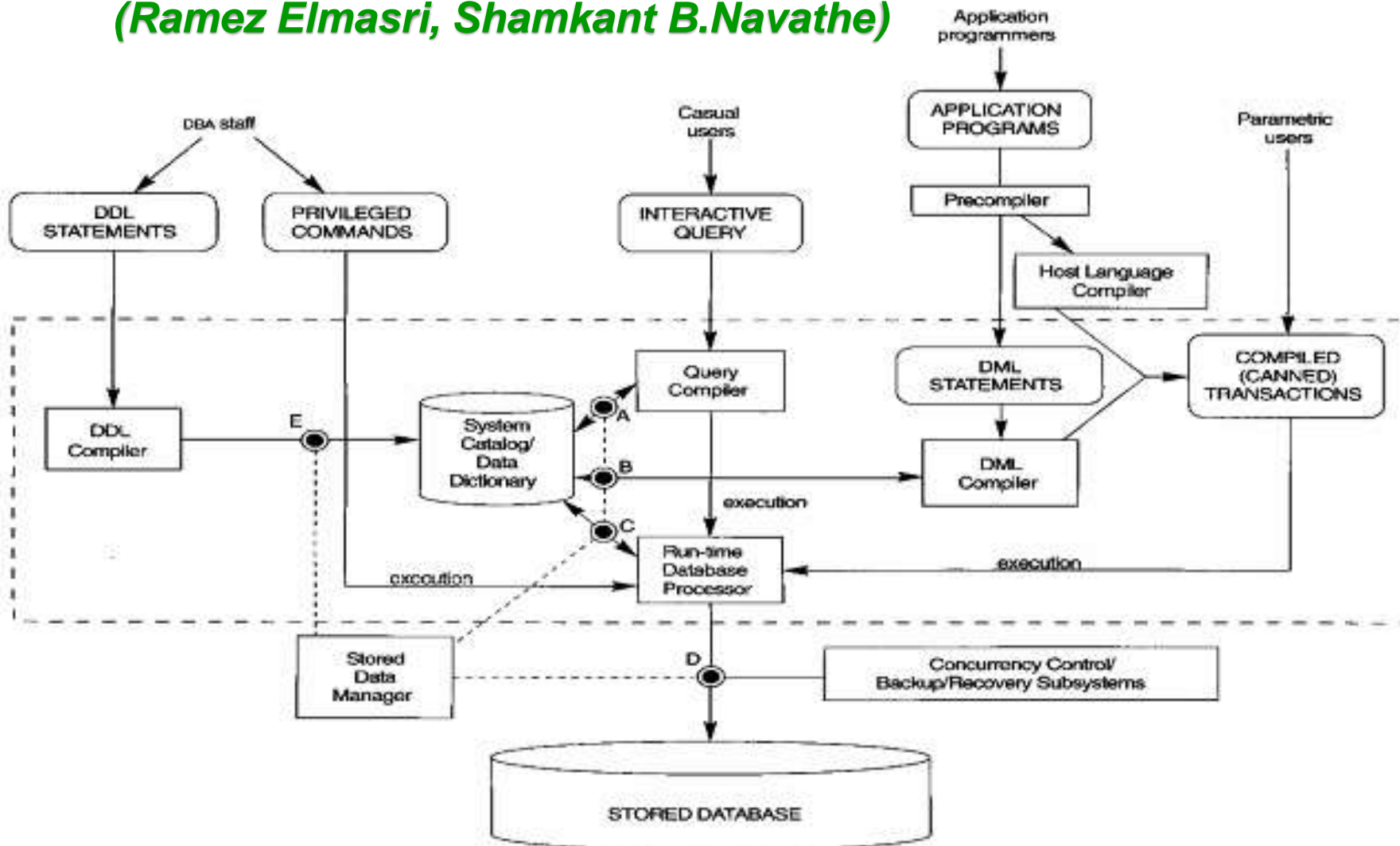
### 3-layer Client/Server architecture



# 3.3 Modules of DBMS



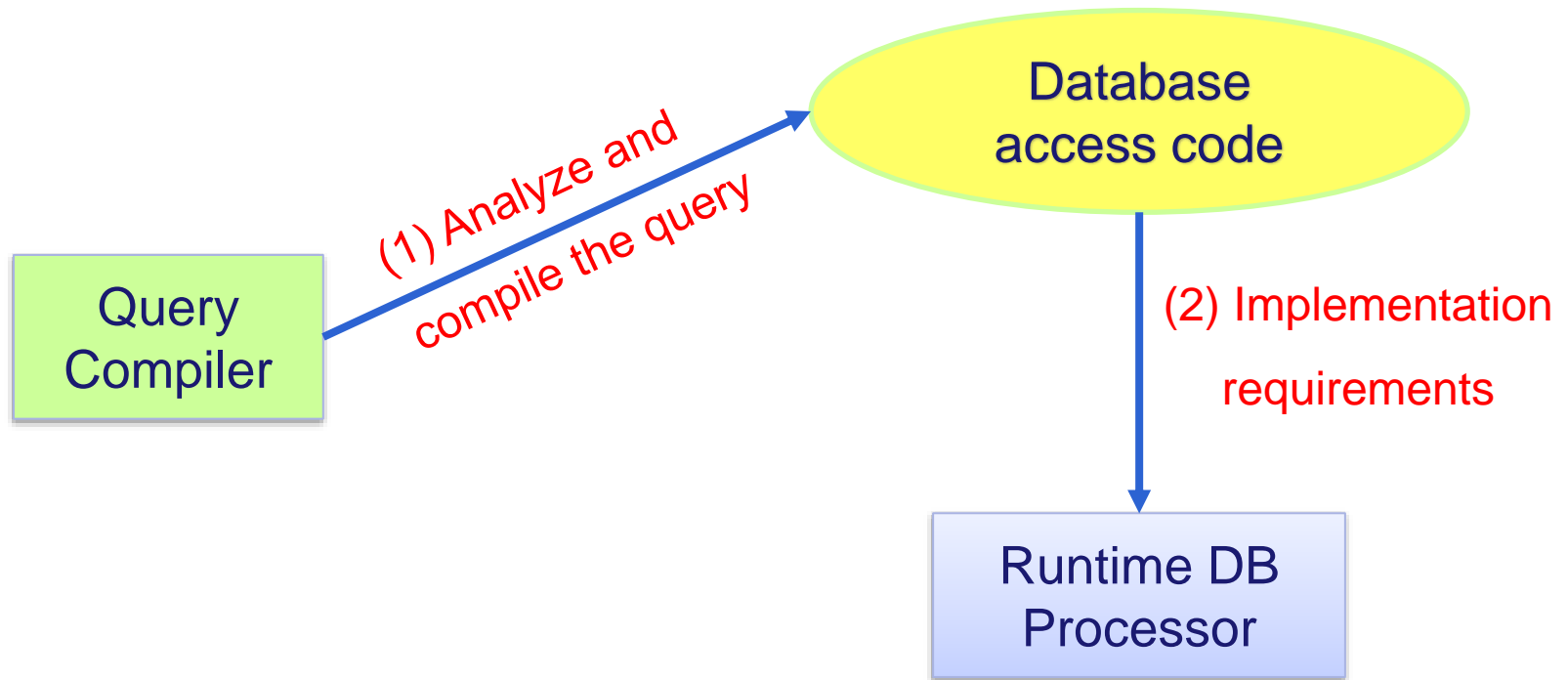
**Reference: Fundamentals of DB Systems 4th Edition  
(Ramez Elmasri, Shamkant B. Navathe)**





- ❑ **DML compiler:** performs schema definition (meta-data) in the DBMS catalog
- ❑ **DBMS catalog:** contains information about the name, file size, data type, storage location of each file, and constraints ...
- ❑ **Runtime DB processor:** manage DB access in real time, record update activities on the DB

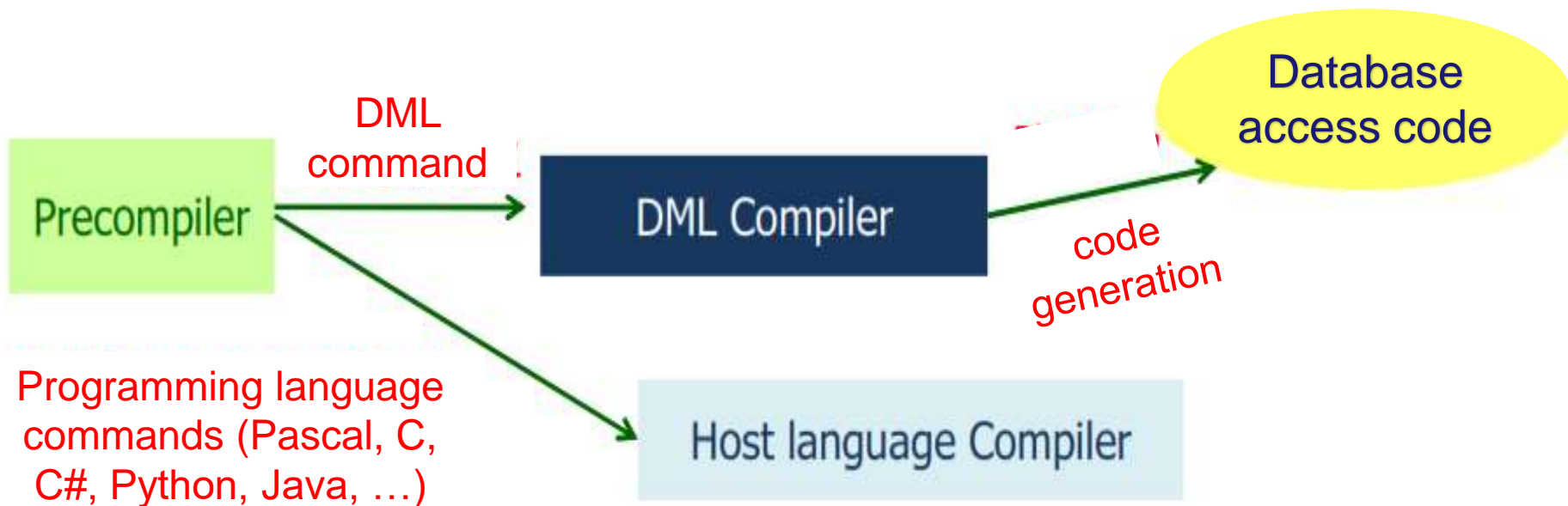
- ❑ **Query Compiler:** compiles interactive queries that users enter



## 3.3 Modules of DBMS



❑ **Precompiler:** retrieves DML statements from an application program written in a general programming language



# Discussion

