CSC10006 – Introduction to Database

# Chapter 1
# Overview of Database Systems


Instructor: **Vũ Thị Mỹ Hằng, Ph.D.** (**vtmhang**@fit.hcmus.edu.vn)

Lab Assistants: **Tiết Gia Hồng, M.Sc.** (**tghong**@fit.hcmus.edu.vn)

**Lương Hán Cơ, M.Sc.** (**lhco**@fit.hcmus.edu.vn)




fit@hcmus

KHOA CÔNG NGHỆ THÔNG TIN
TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN

# Content

- Introduction of databases
- Evolution of database systems
- Characteristics of the database approach
- Database management systems (DBMS)
- Data models
- Database languages
- Database users

cdio 4.0

fit@hcmus

# Content

- Introduction of databases
- Evolution of database systems
- Characteristics of the database approach
- Database management systems (DBMS)
- Data models
- Database languages
- Database users

# Databases: WHAT and WHY?

fit@hcmus

## A Day In Susan's Life

See how many databases she interacts with each day

*Before leaving for work, Susan checks her Facebook and Twitter accounts*

Where is the data about the friends and groups stored?

Where are the "likes" stored and what would they be used for?


*On her lunch break, she picks up her prescription at the pharmacy*

Where is the pharmacy inventory data stored?

What data about each product will be in the inventory data?

What data is kept about each customer and where is it stored?


*After work, Susan goes to the grocery store*

Where is the product data stored?

Is the product quantity in stock updated at checkout?

Does she pay with a credit card?


*At night, she plans for a trip and buys airline tickets and hotel reservations online*

Where does the online travel website get the airline and hotel data from?

What customer data would be kept by the website?

Where would the customer data be stored?


*Then she makes a few online purchases*


Where are the product and stock data stored?

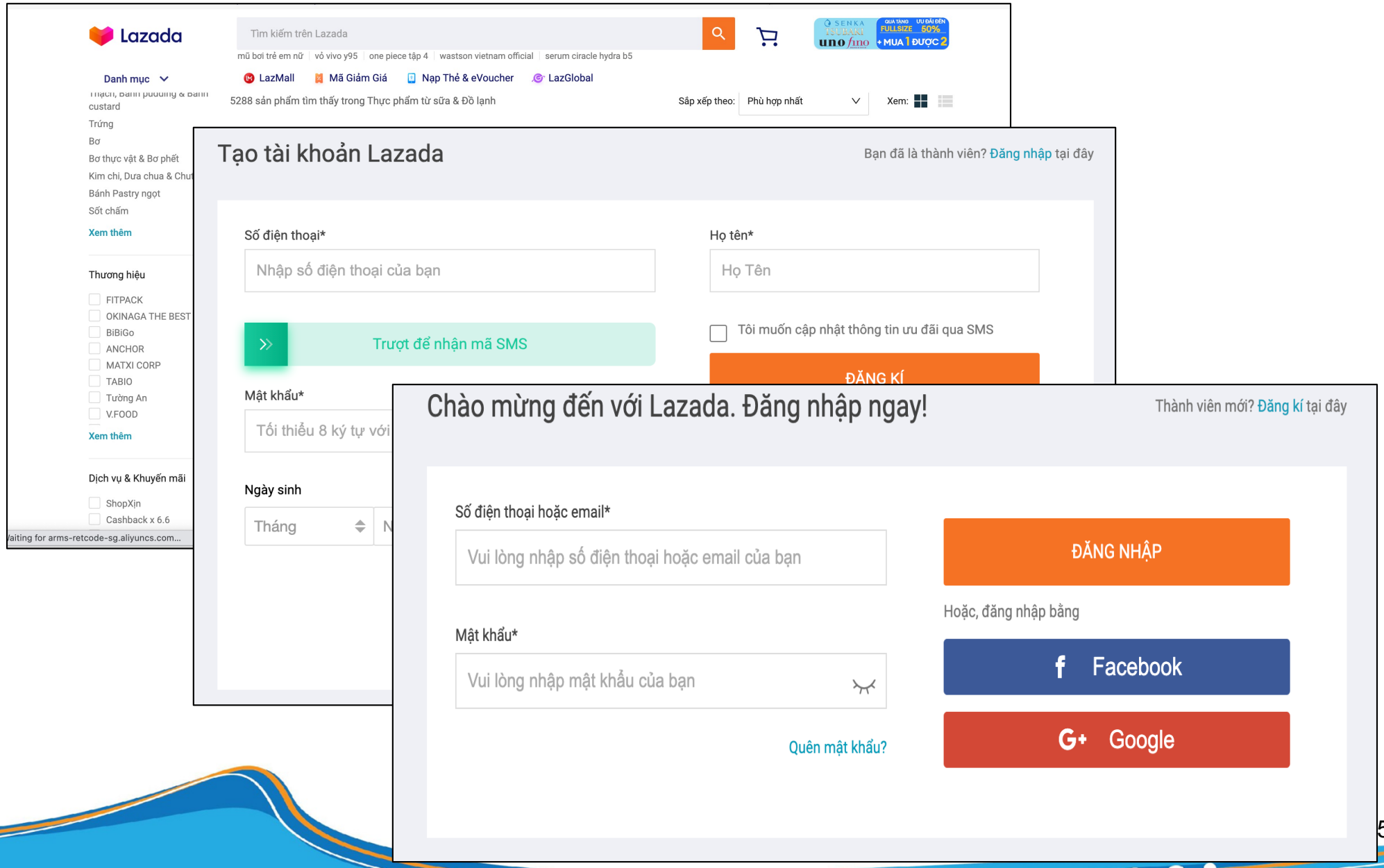
Where does the system get the data to generate product "recommendations" to the customer?

Where would credit card information be stored?

cdio 4.0

fit@hcmus

# Databases: WHAT and WHY?



# Databases: HOW

fit@hcmus

Real business & application (marketing, production, banking, …)

Database Design Process

Develop

Automate & support tasks

Computers

**HOW can computers understand the real-world domain to digitize & support automation?**

# Databases: HOW

fit@hcmus

Real business & application (marketing, production, banking, …)

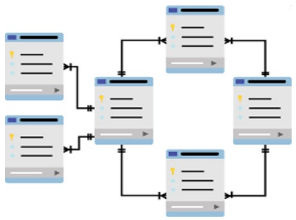

Database Design Process

Develop

Automate & support tasks

Analyze & Design

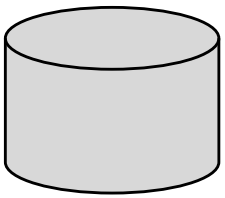
Data



Functionalities, Interfaces

Implementation

Computers

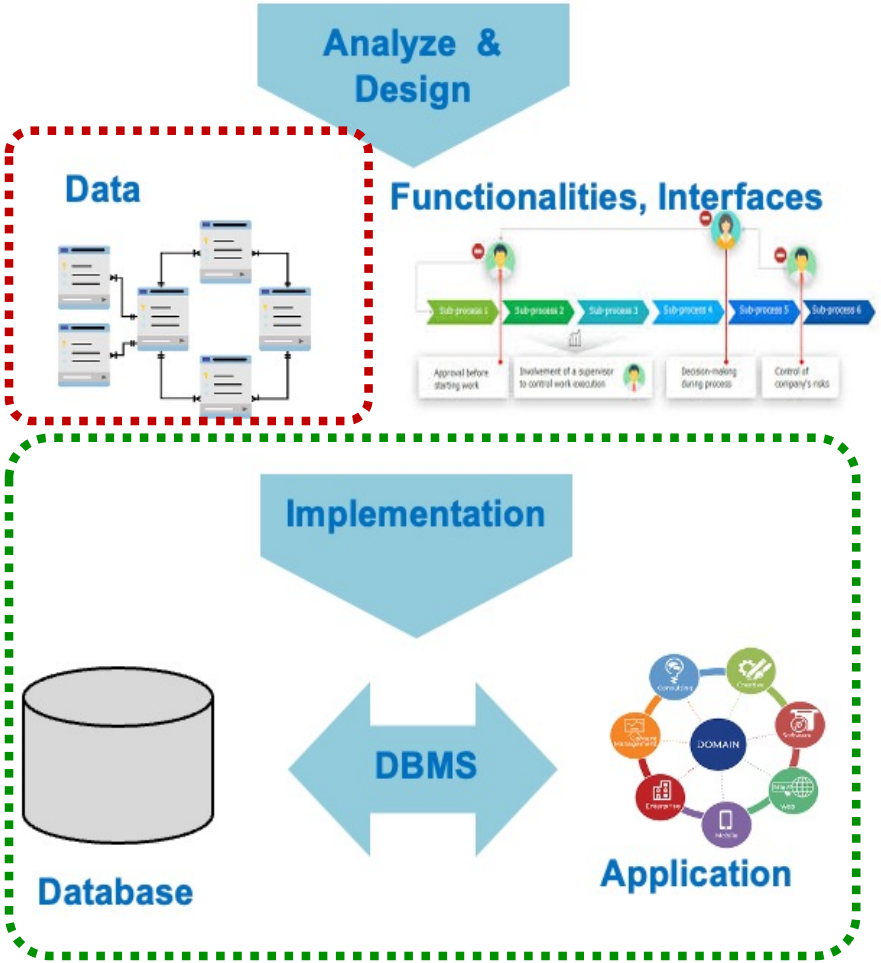


Database

DBMS


Application

# Database Design Process




**HOW** to represent data structures and their characteristics, relations, constraints, etc.?

**Data models**
Entity Relationship Model
Relational Model
Object-oriented Model
Network Model
…

**Database languages**
SQL
Relational Algebra
Relational Calculus
(& Integrity Constraints)

**HOW** to define, operate, and manipulate (relational) database?

# Definition

- **Data**: *Stored* representations of *meaningful objects* and *events*.
  - Structured: numbers, text, dates.
  - Unstructured: images, video, documents.
- **Information**: Data *processed* and *organized* to increase knowledge in the person using the data.
- **Database**: *Collection* of logically *related data* which provides information *relevant to a business/enterprise*.

# Example 1 – Course Management

| MÔNHỌC | TênMH | MãMH | SốTC | Khoa |
|---|---|---|---|---|
| | Khoa học máy tính | CS1310 | 4 | CNTT |
| | Cấu trúc dữ liệu | CS3320 | 4 | CNTT |
| | Toán rời rạc | MATH2410 | 3 | TOÁN |
| | Cơ sở dữ liệu | CS3380 | 3 | CNTT |

| SINHVIÊN | Tên | MSSV | Lớp | Khoa |
|---|---|---|---|---|
| | Trang | 17 | 1 | CNTT |
| | Ngọc | 8 | 2 | CNTT |

| HỌCPHẦN | MãHP | MãMH | HọcKỳ | Năm | GiáoViên |
|---|---|---|---|---|---|
| | 85 | MATH2410 | 1 | 2008 | Anh |
| | 92 | CS1310 | 1 | 2007 | Tiến |
| | 112 | MATH2410 | 2 | 2008 | Anh |
| | 119 | CS1310 | 2 | 2007 | Tiến |

| KẾTQUẢ | MSSV | MãHP | Điểm |
|---|---|---|---|
| | 17 | 112 | 10 |
| | 17 | 119 | 7 |
| | 8 | 85 | 6 |
| | 8 | 92 | 9 |

| ĐIỀUKIỆN | MãMH | MãMH_Trước |
|---|---|---|
| | CS3380 | CS3320 |
| | CS3380 | MATH2410 |
| | CS3320 | CS1310 |

cdio 4.0

fit@hcmus

# Example 2 – Project Management

| EMPLOYEE | LName | MName | FName | SSN | BirthDate | SuperSSN | DNo |
|---|---|---|---|---|---|---|---|
| | Tran | Hong | Quang | 987987987 | 03/09/1969 | 987654321 | 4 |
| | Nguyen | Thanh | Tung | 333445555 | 12/08/1955 | 888665555 | 5 |
| | Nguyen | Manh | Hung | 666884444 | 09/15/1962 | 333445555 | 5 |
| | Tran | Thanh | Tam | 453453453 | 07/31/1972 | 333445555 | 5 |

| PROJECT | PName | PNumber | PLocation | DNum |
|---|---|---|---|---|
| | San pham X | 1 | VUNG TAU | 5 |
| | San pham Y | 2 | NHA TRANG | 5 |
| | San pham Z | 3 | TP HCM | 5 |
| | Tin hoc hoa | 10 | HA NOI | 4 |

| WORKS_ON | SSN | PNo | Hours |
|---|---|---|---|
| | 123456789 | 1 | 32.5 |
| | 123456789 | 2 | 7.5 |
| | 666884444 | 3 | 40.0 |
| | 453453453 | 1 | 20.0 |

# Characteristics of Databases

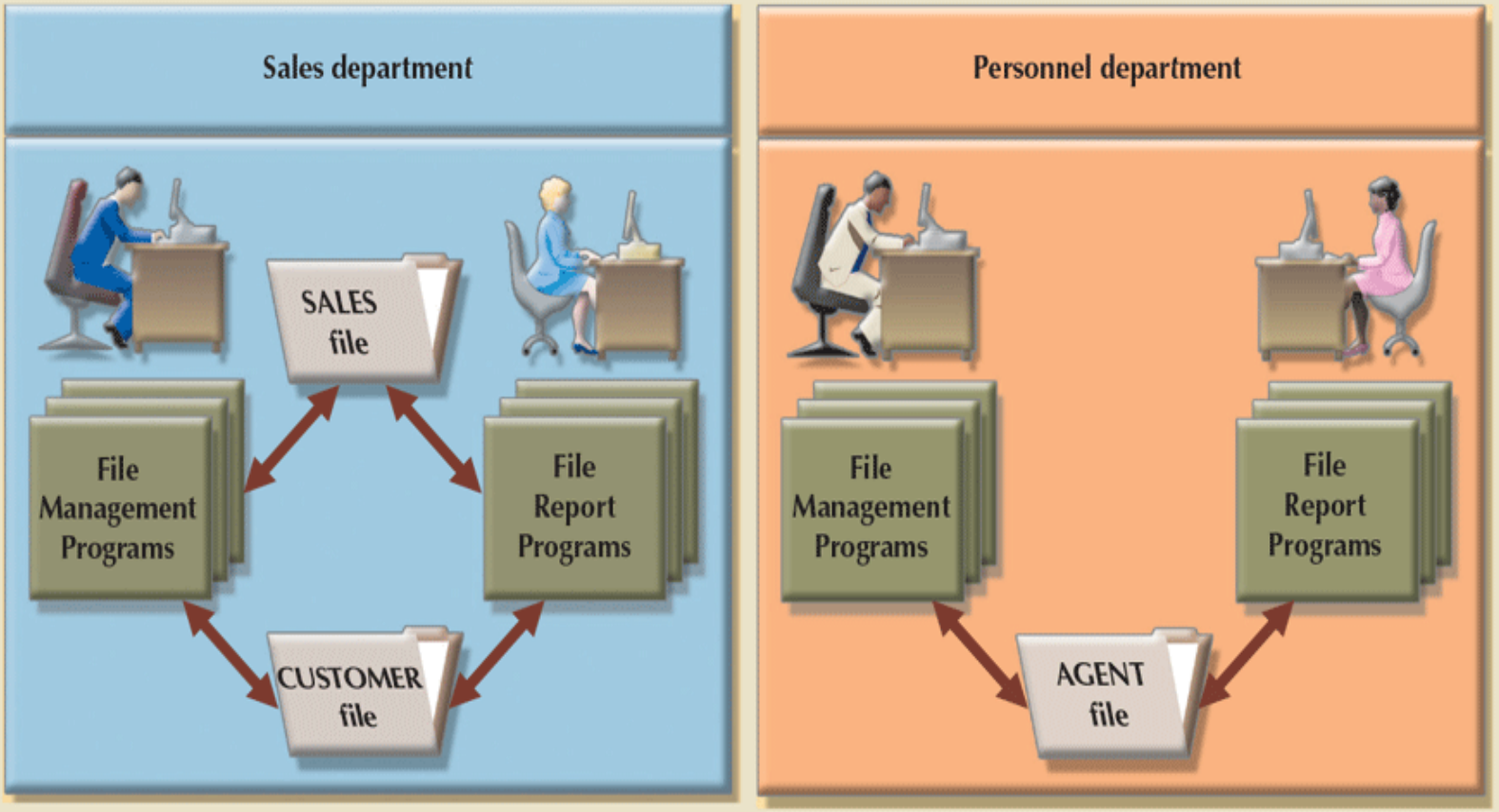
- A logically coherent collection of data with some inherent meaning.
  - Random assortment of data cannot correctly be a database.
- Is designed, built, and populated with data for a specific purpose, for intended group of users or applications.
  - Example: University management, Hospital management
- Data is stored using a structure ➔ structured database
  - Relational databases store data in two-dimensional tables.
- Other types of database: unstructured databases, document databases, graph databases.

# Content

- Introduction of databases
- Evolution of database systems
- Characteristics of the database approach
- Database management systems (DBMS)
- Data models
- Database languages
- Database users

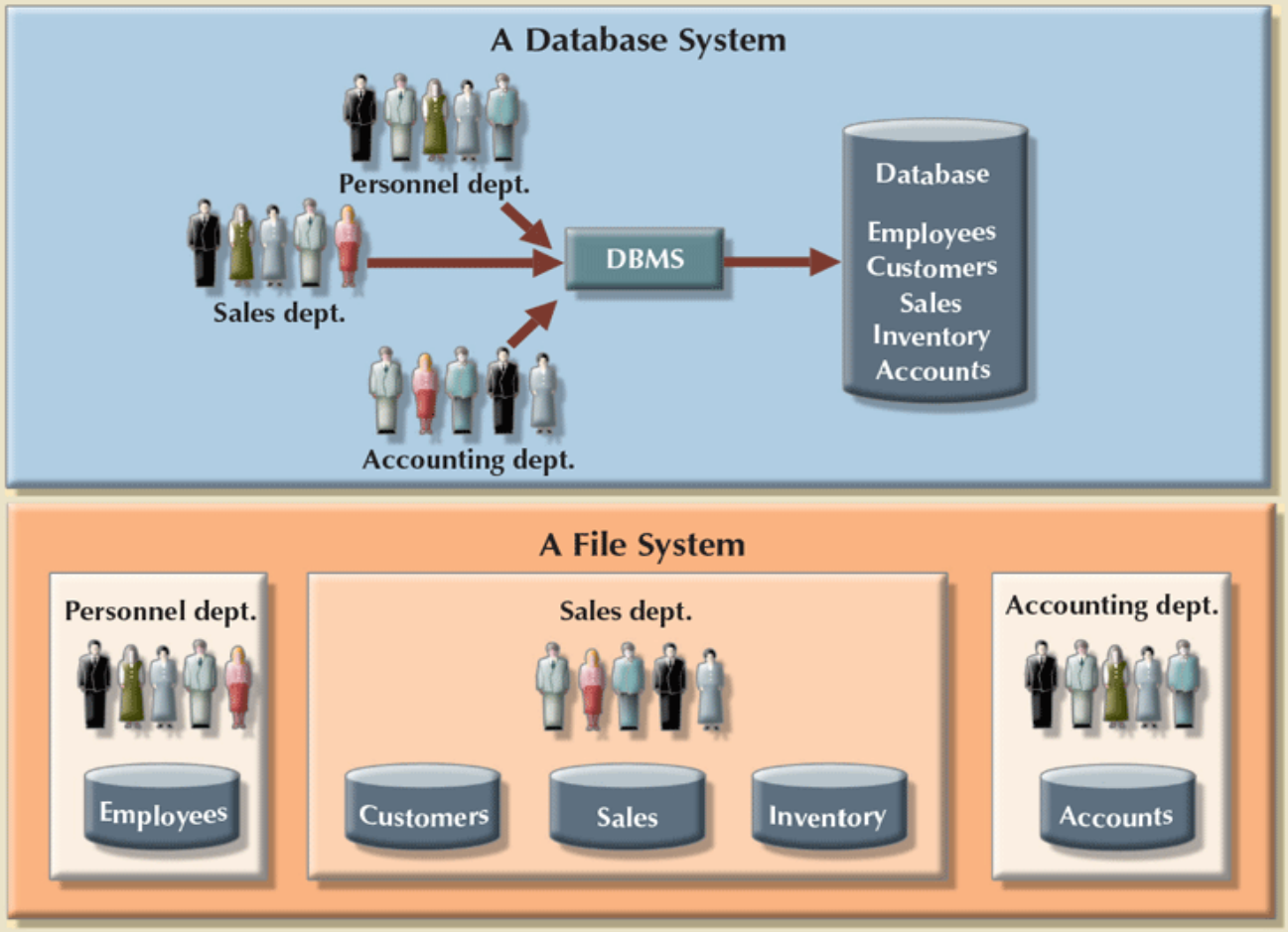
# Evolution of Database Systems

FIGURE 1.9 A SIMPLE FILE SYSTEM



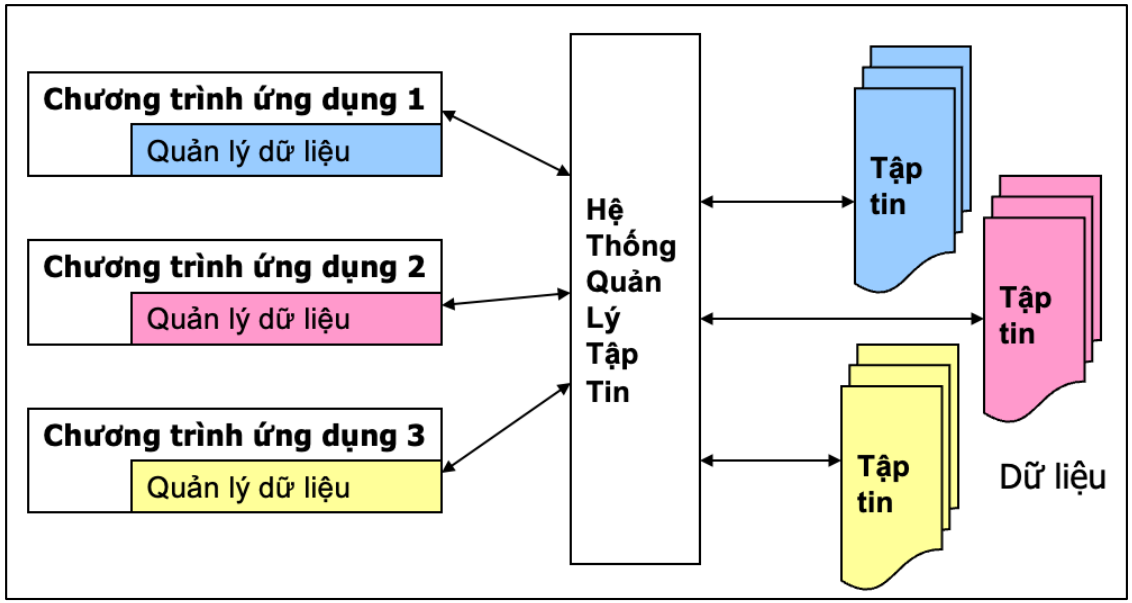
# Evolution of Database Systems

FIGURE 1.10 CONTRASTING DATABASE AND FILE SYSTEMS



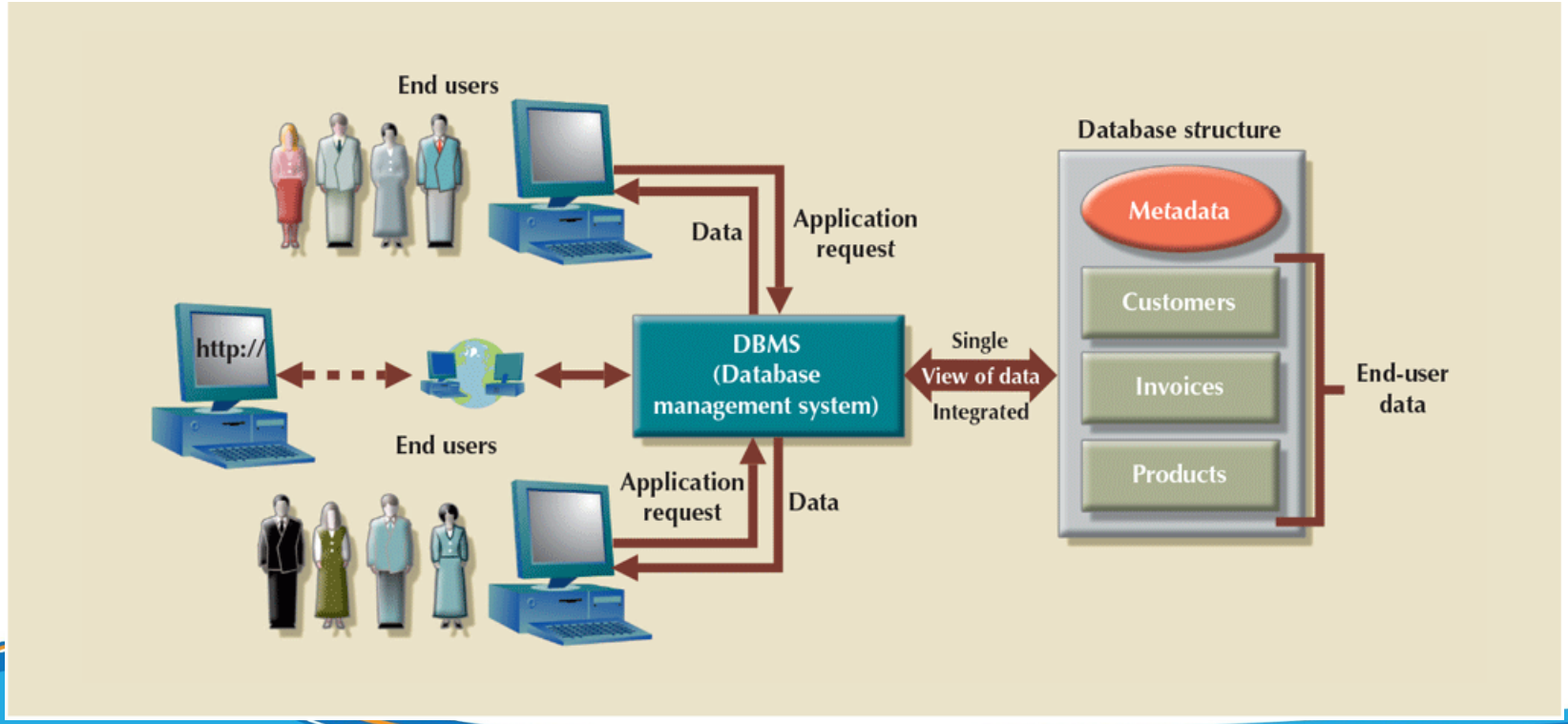
# Disadvantages of File Processing

- Program-Data Dependence
- Duplication of Data
- Lack of Security and Limited Data Sharing
- Expensive and Lengthy Program Development
- Excessive Program Maintenance

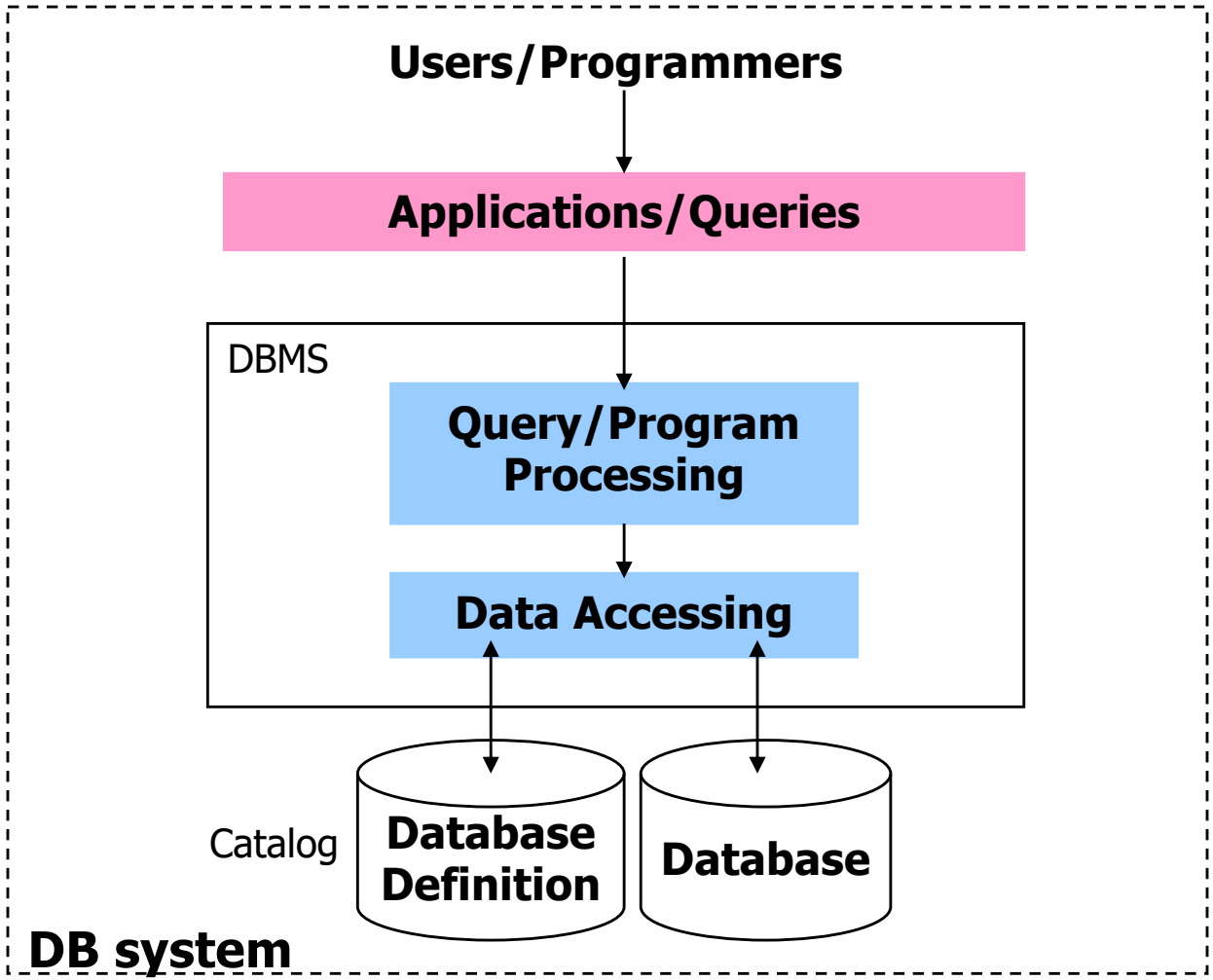


# Database Systems

- Central repositories of shared data
- Data is stored, processed, and managed in a suitable and standardized manner
- Data is managed by a controlling agent
- ➔ Requires a Database Management System (DBMS)



# Database Systems

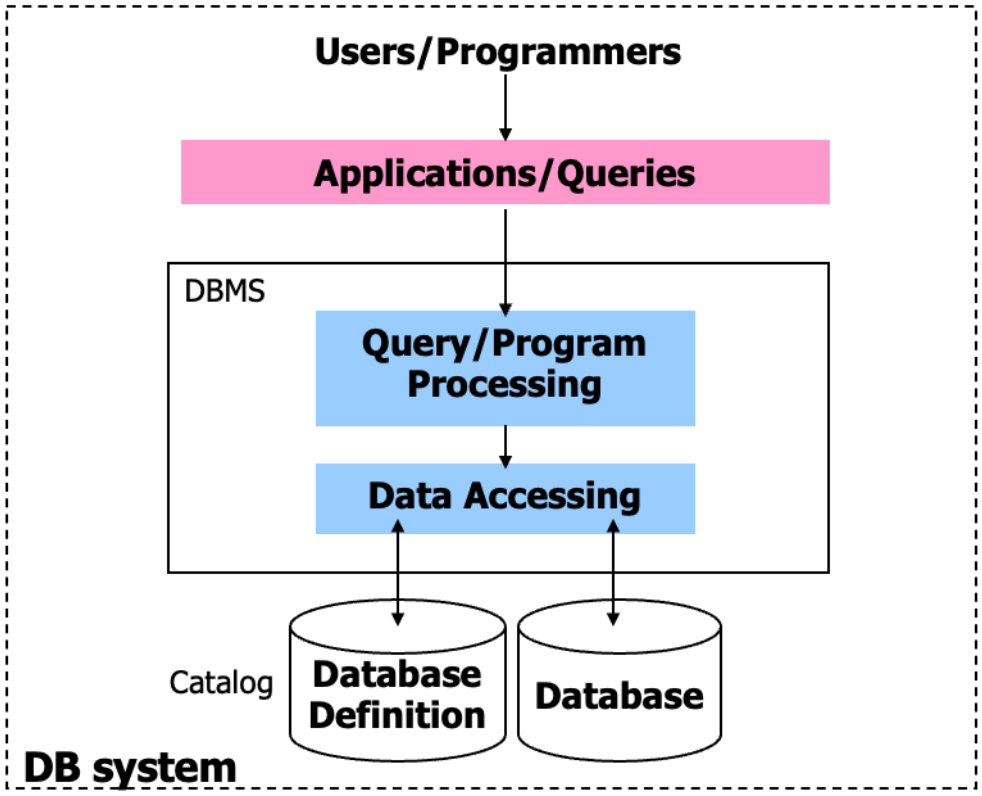


# Content

- Introduction of databases
- Evolution of database systems
- Characteristics of the database approach
- Database management systems (DBMS)
- Data models
- Database languages
- Database users

# Self-Describing (Meta-data)

- A **catalog** called "metadata" contains information such as the structure of data, type and storage format of data items, and constraints on the data.




**RELATION**

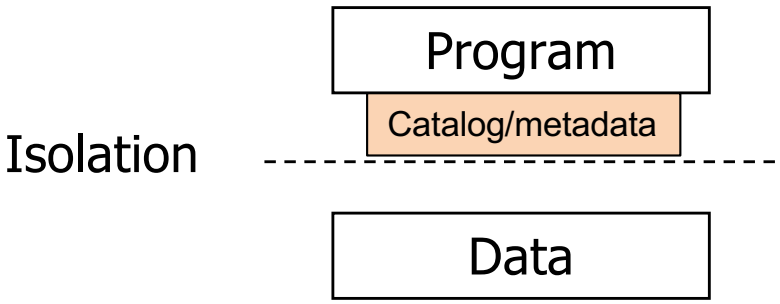
| Relation_name | No_of_columns |
|---|---|
| EMPLOYEE | 7 |
| PROJECT | 4 |
| WORKS_ON | 3 |

**COLUMN**

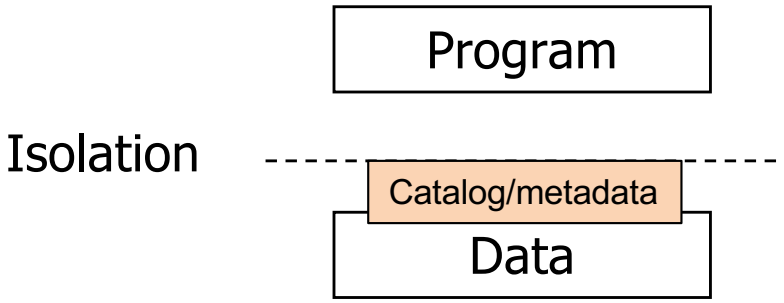
| Column_name | Data_type | Belongs_to_relation |
|---|---|---|
| LName | Character(10) | EMPLOYEE |
| FName | Character(10) | EMPLOYEE |
| ... | ... | ... |

# Isolation
## (Program-Data independence)

- The structure of data is stored in **catalog separately** from the access **programs**.




Program contains catalog □ data structure depends on program



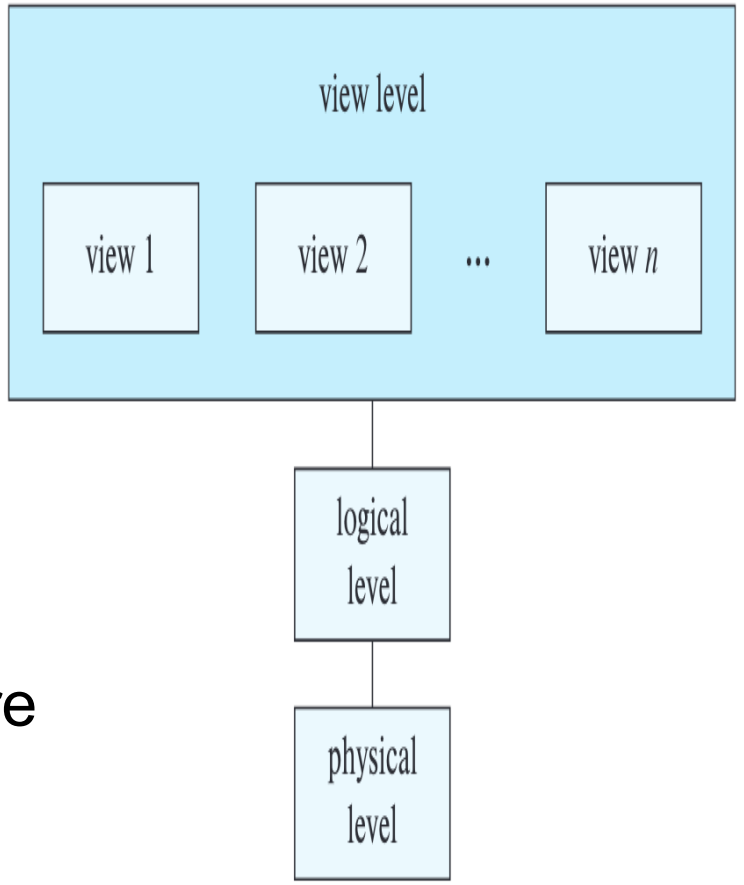

DBMS contains catalog □ data is independent of program

☺ A little change in the structure happens.

☺ Application programs are rarely revised.

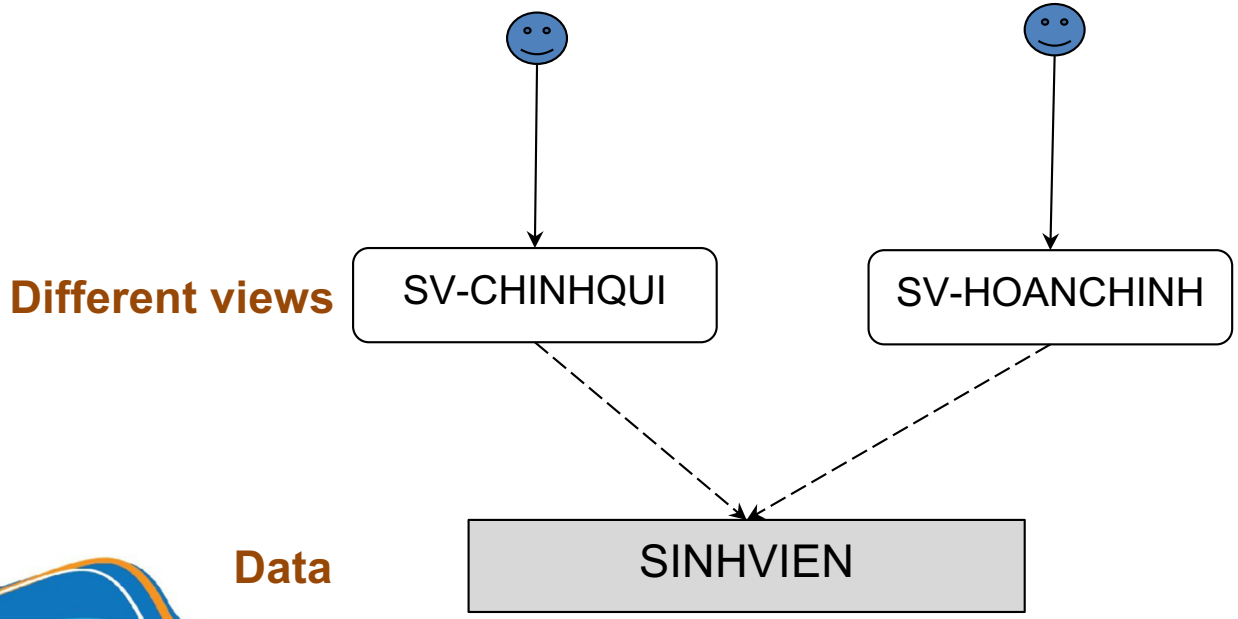
# Data Abstraction
## (Conceptual-Logical-Physical Levels)

- Hide the data structure complexity from users through several levels of data abstraction to simplify users' interaction with the systems.
  - **Physical level**: HOW data is stored.
  - **Logical level**: WHAT data are stored
  - **View level**: PART of the entire database for a specific USERS.



# Data Abstraction
## (Conceptual-Logical-Physical Levels)

- A DB has many users who may require a different perspective or view of the database.
- Example of (user) views:
  - A subset of the database.
  - Aggregate data that are derived from the database.

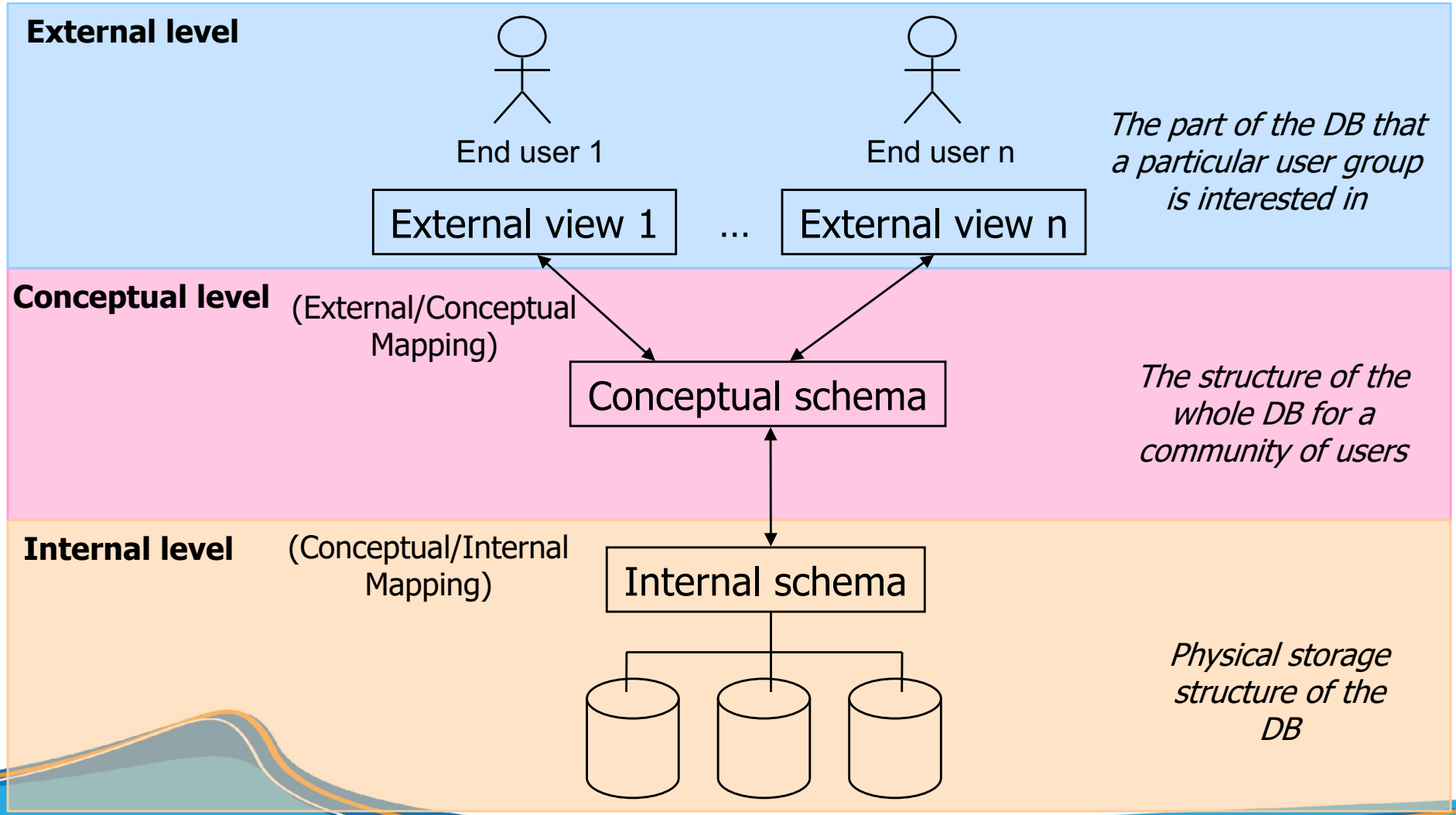


cdio 4.0

fit@hcmus

# Data Abstraction
## (Conceptual-Logical-Physical Levels)

### THREE-SCHEMA ARCHITECTURE



# Sharing Data (Multiusers)

- A multiuser DBMS: Allows multiple users to access the DB at the same time.
  - Data for many applications are integrated and maintained in a single DB.
  - Using concurrency control mechanisms to access the data reasonably
  - Ensure the data will always be valid when they are accessed.

# Content

- Introduction of databases
- Evolution of database systems
- Characteristics of the database approach
- Database management systems (DBMS)
- Data models
- Database languages
- Database users

# DBMS

- A collection of programs that enables users to **create**, **operate**, and **maintain** a database.
  - **Definition**: Specifying the data types, structures, and constraints for the data.
  - **Construction**: Storing the data itself on some storage medium.
  - **Manipulation**: Querying the database to retrieve data, updating the database to reflect changes, generating reports from the data.
  - **Sharing**: Allowing multiple users/programs to access the database concurrently.

# DBMS Functions

- Data dictionary management
  - Data dictionary stores definitions of data elements and their relationships (meta-data)
- Data storage management
  - Performance tuning ensures efficient performance of the database in terms of storage and access speed
- Data transformation and presentation
  - Data is formatted to conform to required data structures
- Security management
  - Enforces user security and data privacy



(Coronel and Morris, 2015) (Hoffer et al., 2019)

# DBMS Functions

- Multiuser access control
  - Sophisticated algorithms ensure that multiple users can access the database concurrently without compromising its integrity.
- Backup and recovery management
  - Enables recovery of the database after a failure
- Data integrity management
  - Minimizes redundancy and maximizes consistency



(Coronel and Morris, 2015) (Hoffer et al., 2019)

# DBMS Functions

- Database access languages and application programming interfaces
  - **Query language**: lets the user specify what must be done without having to specify how
  - **Structured Query Language (SQL)**: a structured query language and data access standard supported by the majority of DBMS vendors
- Database communication interfaces
  - Accept end-user requests via multiple, different network environments



(Coronel and Morris, 2015) (Hoffer et al., 2019)

# DBMS Disadvantages

- Increased costs
- Management complexity
- Maintaining currency
- Vendor dependence
- Frequent upgrade/replacement cycles



(Coronel and Morris, 2015) (Hoffer et al., 2019)

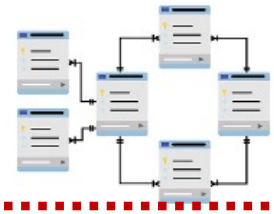
# Content

- Introduction of databases
- Evolution of database systems
- Characteristics of the database approach
- Database management systems (DBMS)
- Data models
- Database languages
- Database users

# Recall: DB Design Process

Analyze & Design

Data



Functionalities, Interfaces

**HOW** to represent data structures and their characteristics, relations, constraints, etc.?

**Data models**

Entity Relationship Model

Relational Model

Object-oriented Model

Network Model

…

**Database languages**

SQL

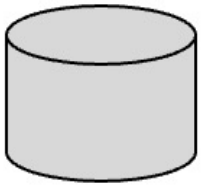
Relational Algebra

Relational Calculus

(& Integrity Constraints)

**HOW** to define, operate, and manipulate (relational) database?

Implementation

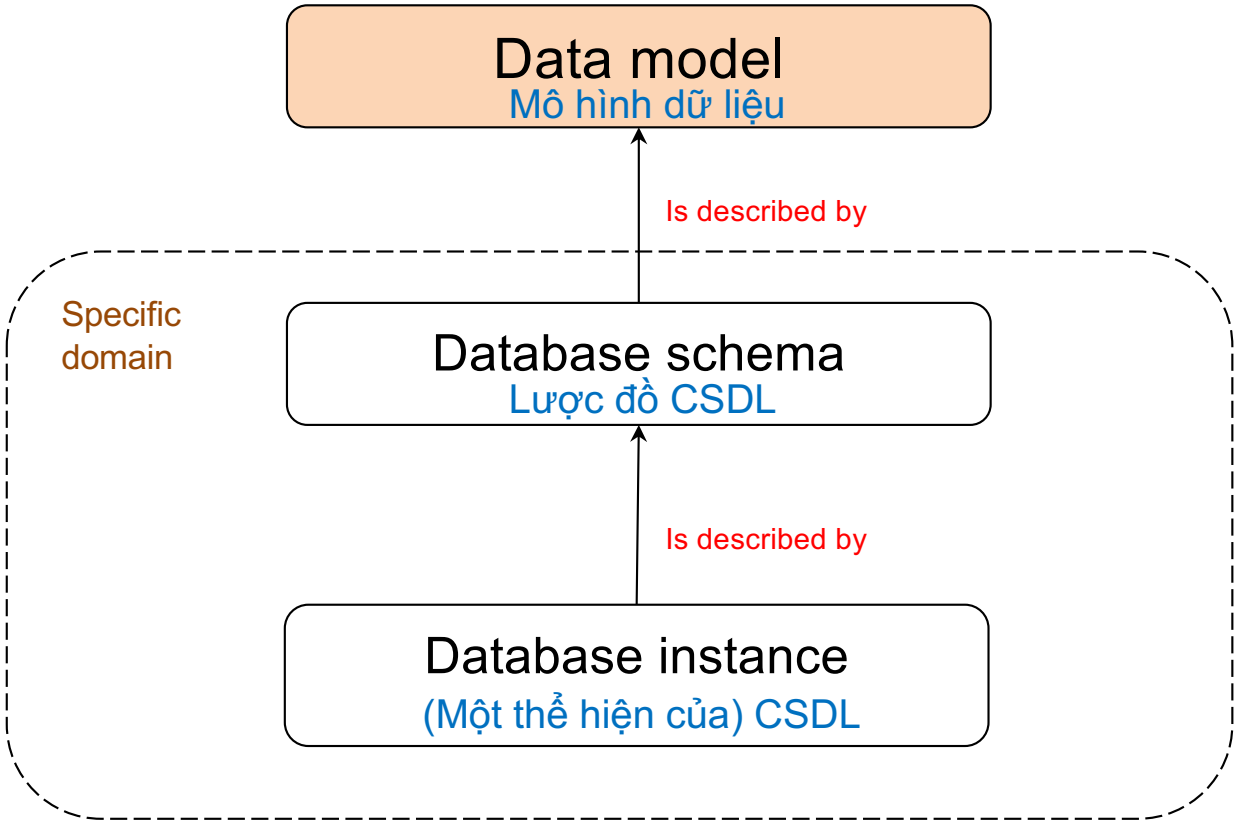


DBMS


Database

Application

# Data Models


Data model
Mô hình dữ liệu
Is described by
Specific domain
Database schema
Lược đồ CSDL
Is described by
Database instance
(Một thể hiện của) CSDL

# Data Models

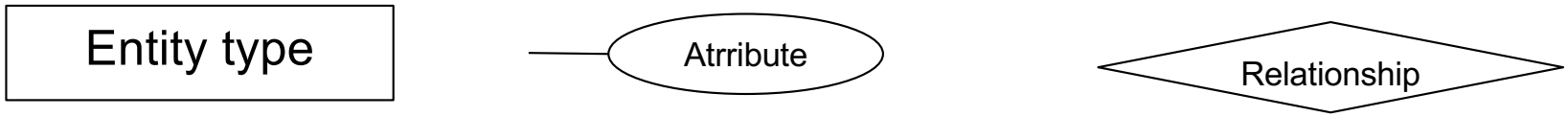
- A collection of **concepts** used to describe the structure of a DB (data types, relationships, and constraints, etc.)
- Including a set of basic **operations** for specifying retrievals and updates on the DB.
- Example:
  - ER model: entity, relationships, cardinality constraints, etc.
  - Relational (data) model: relations, primary keys, etc.

# Data Models

- Categories:
  - High-level (or conceptual) models: Provide concepts that are close to the way (business) users perceive data (ER model, object-oriented model, etc.).
  - Implementation (or logical) data models: Provide concepts that are understood by end users, but not too far from the way data is organized within the computer (relational model, network model, etc.).
  - Low-level (or physical) data models: Provide concepts that describe the details of how data is stored in the computer.

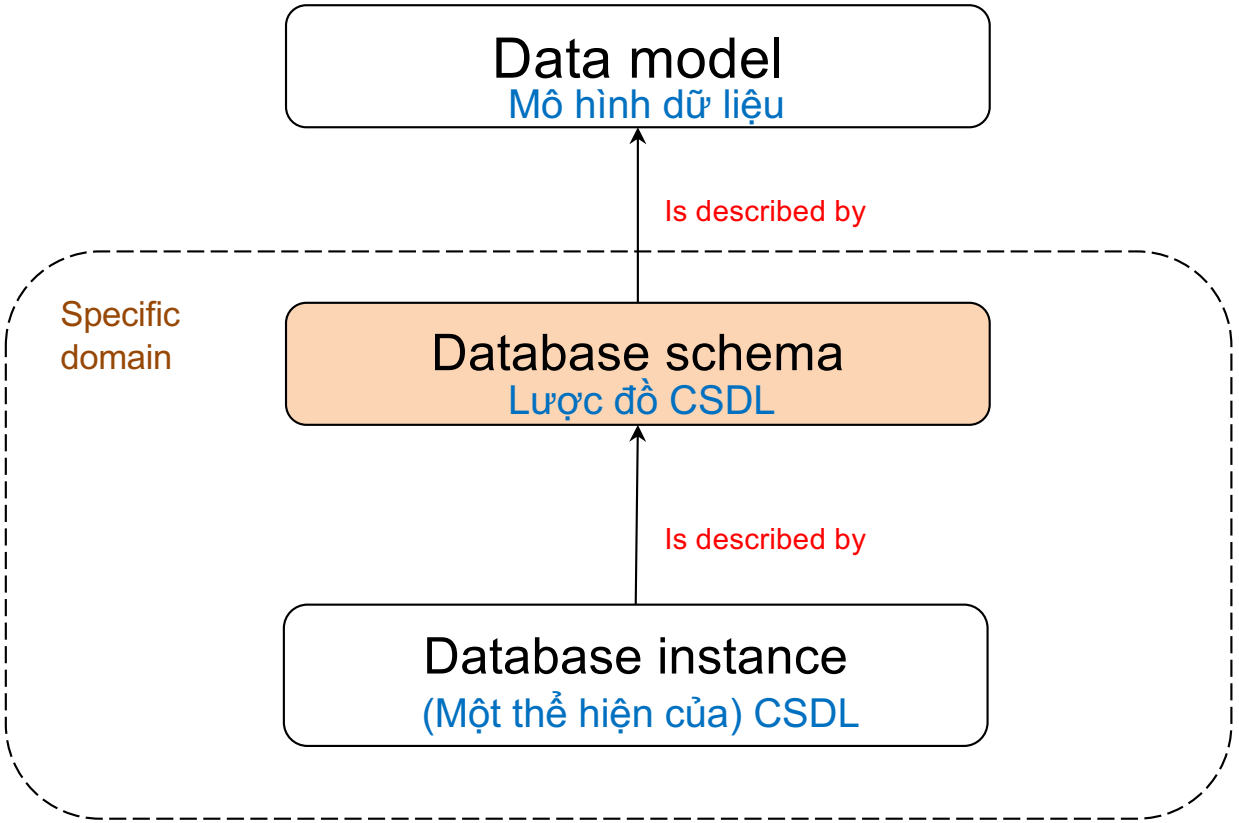
# Examples of Data Models

- Entity Relationship Diagram



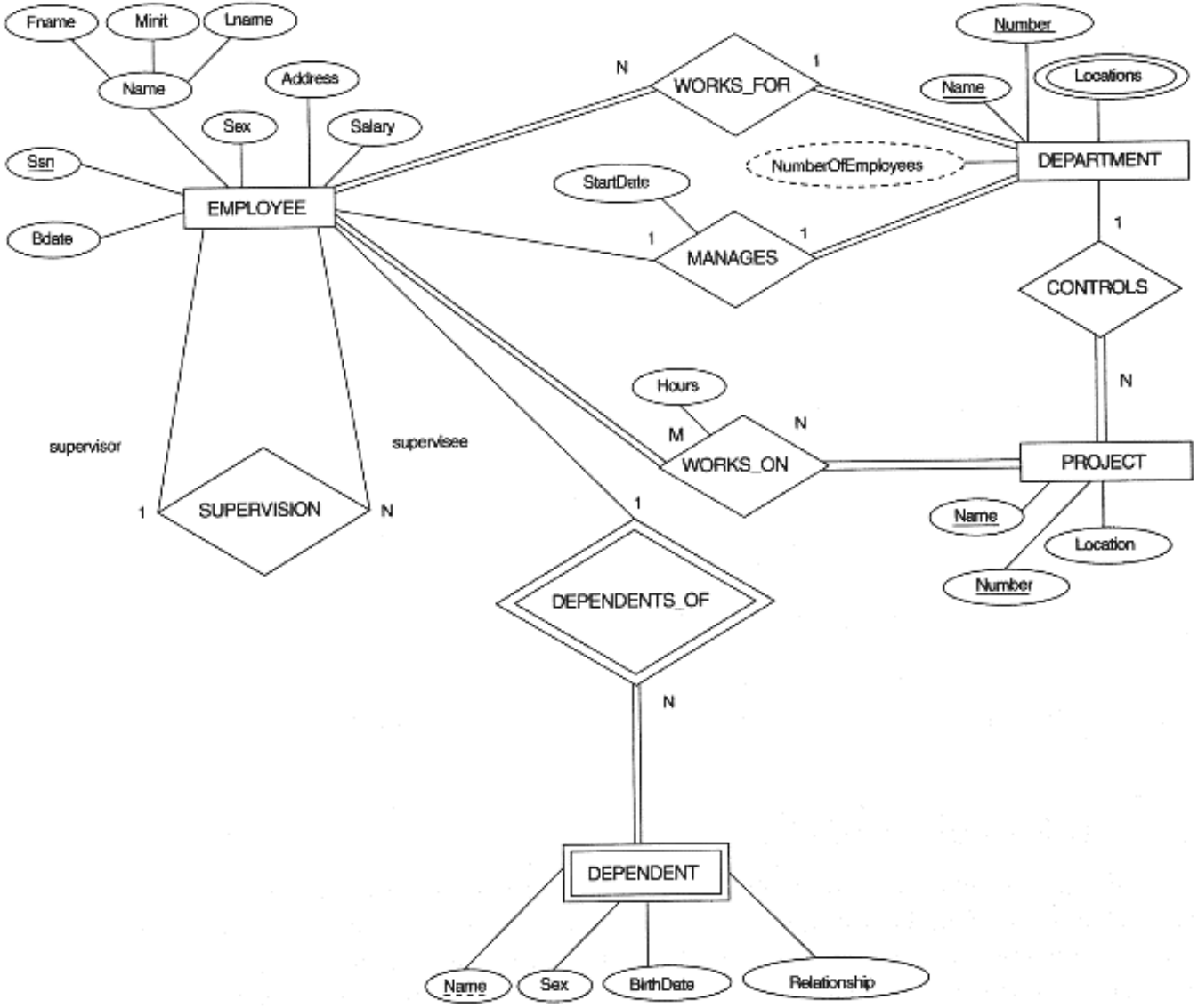

- Network data model

# Database Schema




- Description of the **structure** and **constraints** on the **database** about a **particular domain** (banking, education, marketing, etc.) based on a data model.

# Example of ER Database Schema (ER Diagram)



# Example of (Relational) Database Schema

| SINH VIÊN | TÊNSV | MÃSV | LỚP | NGÀNH |
|---|---|---|---|---|

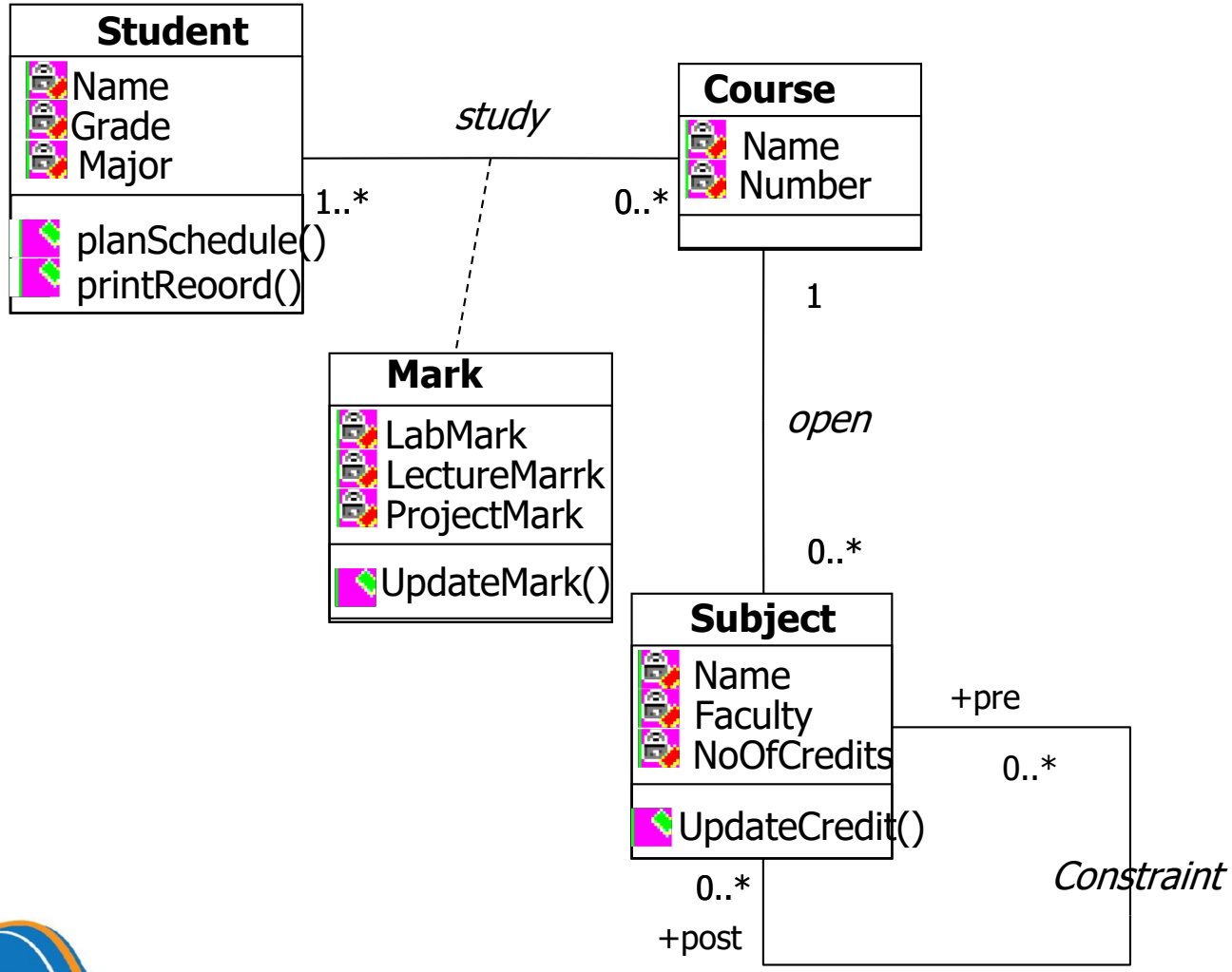
| MÔN HỌC | TÊNMH | MÃMH | KHOA | TÍNCHỈ |
|---|---|---|---|---|

| ĐIỀU KIỆN | MÃMH_TRƯỚC | MÃMH |
|---|---|---|

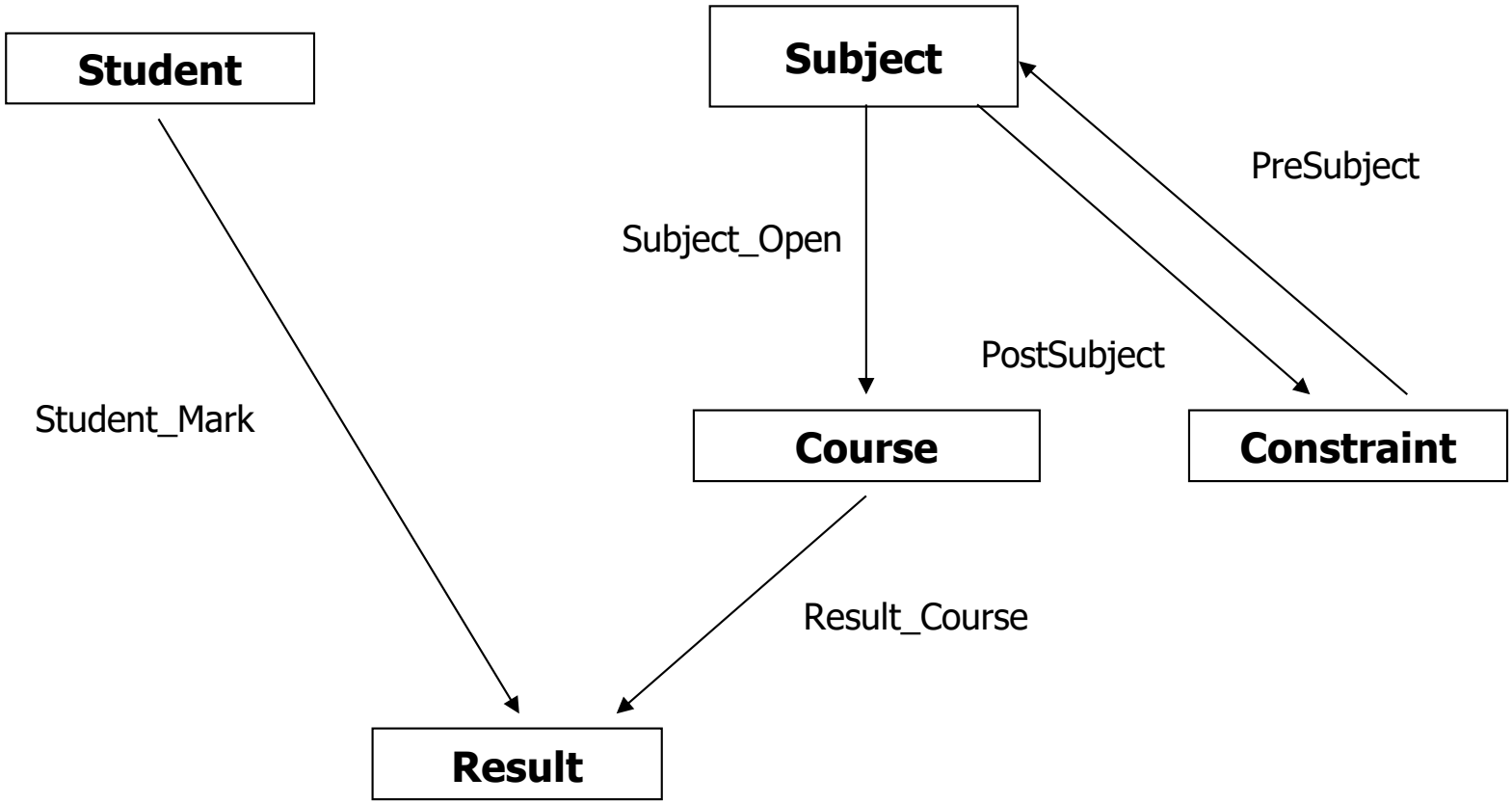
| HỌC PHẦN | MÃHP | GIÁOVIÊN | HỌCKỲ | NĂM |
|---|---|---|---|---|

| KQ_HỌC | MÃSV | MÃHP | ĐIỂM |
|---|---|---|---|

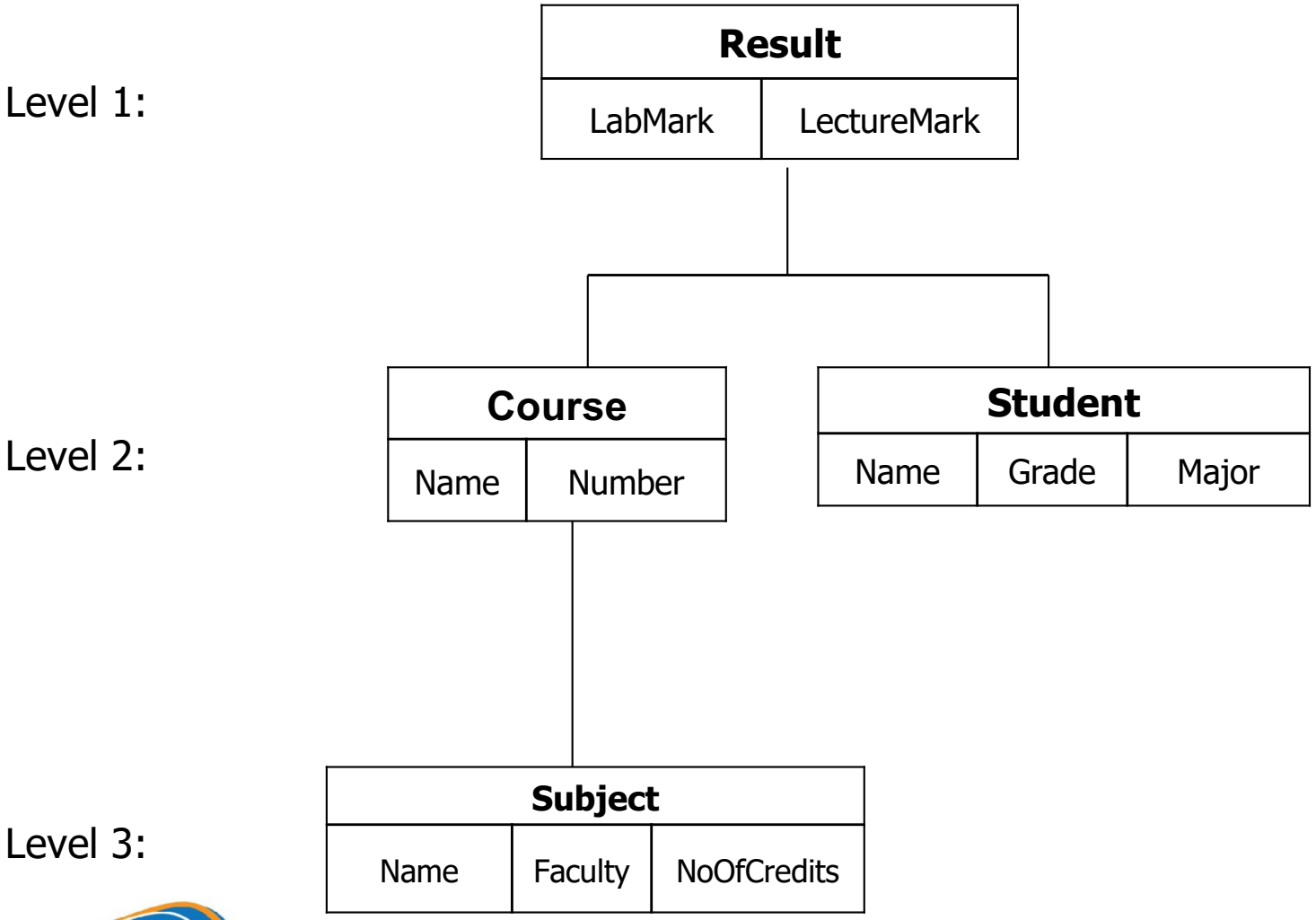
# Example of Object-Oriented Database Schema



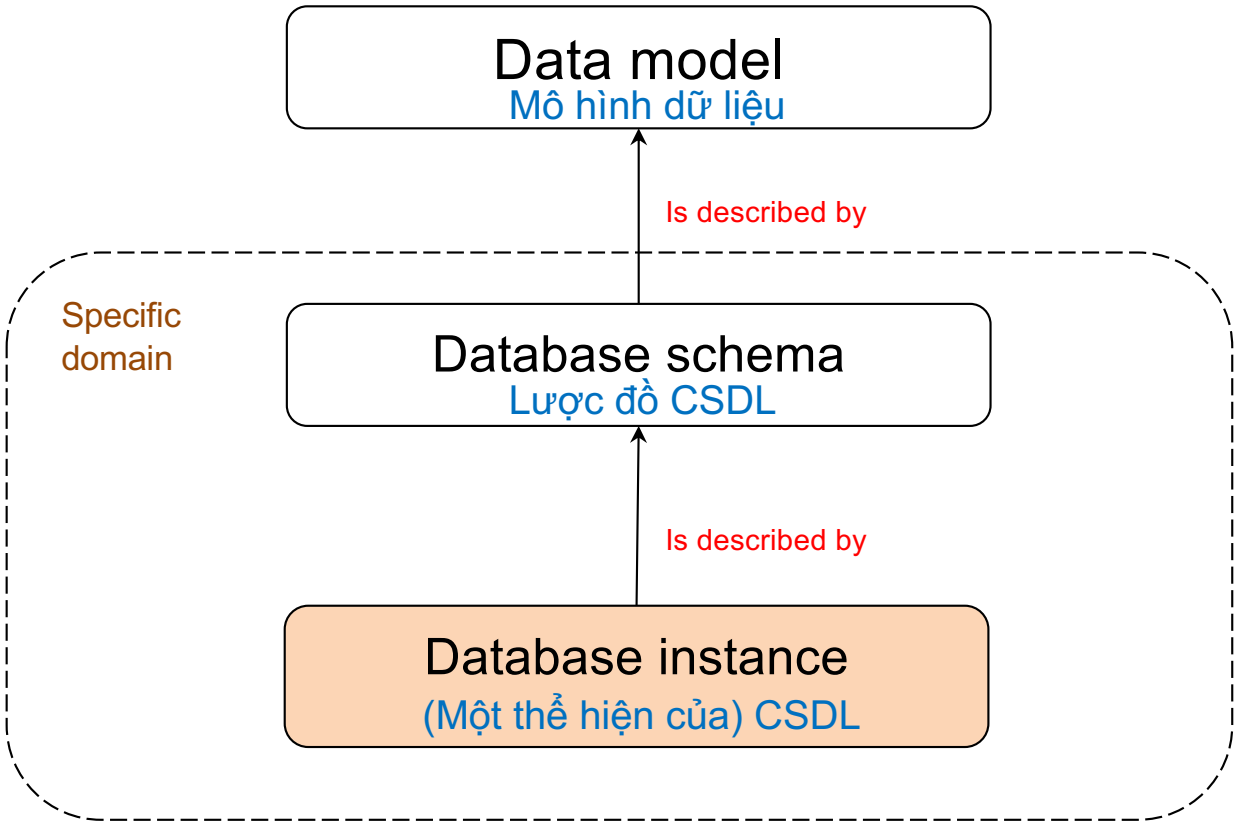
# Example of Network Database Schema



# Example of Hierarchical Database Schema



# Database Instances




- The **data** stored in a database at **a particular moment** of time is called an instance of the database.

# Example of a (Relational) DB Instance

**fit@hcmus**

The (relational) database schema of the database Student Management

| | | | | |
|---|---|---|---|---|
| SINH VIÊN | TÊNSV | MÃSV | LỚP | NGÀNH |
| MÔN HỌC | TÊNMH | MÃMH | KHOA | TÍNCHỈ |
| ĐIỀU KIỆN | MÃMH_TRƯỚC | MÃMH | | |
| HỌC PHẦN | MÃHP | GIÁOVIÊN | HỌCKỲ | NĂM |
| KQ_HỌC | MÃSV | MÃHP | ĐIỂM | |

Is described by

An instance of the database Student Management (at a moment t)

| MÔNHỌC | TênMH | MãMH | SốTC | Khoa |
|---|---|---|---|---|
| | Khoa học máy tính | CS1310 | 4 | CNTT |
| | Cấu trúc dữ liệu | CS3320 | 4 | CNTT |
| | Toán rời rạc | MATH2410 | 3 | TOÁN |
| | Cơ sở dữ liệu | CS3380 | 3 | CNTT |

| KẾTQUẢ | MSSV | MãHP | Điểm |
|---|---|---|---|
| | 17 | 112 | 10 |
| | 17 | 119 | 7 |
| | 8 | 85 | 6 |
| | 8 | 92 | 9 |

| ĐIỀUKIỆN | MãMH | MãMH_Trước |
|---|---|---|
| | CS3380 | CS3320 |
| | CS3380 | MATH2410 |
| | CS3320 | CS1310 |

| HỌCPHẦN | MãHP | MãMH | HọcKỳ | Năm | GiáoViên |
|---|---|---|---|---|---|
| | 85 | MATH2410 | 1 | 2008 | Anh |
| | 92 | CS1310 | 1 | 2007 | Tiến |
| | 112 | MATH2410 | 2 | 2008 | Anh |
| | 119 | CS1310 | 2 | 2007 | Tiến |

| SINHVIÊN | Tên | MSSV | Lớp | Khoa |
|---|---|---|---|---|
| | Trang | 17 | 1 | CNTT |
| | Ngọc | 8 | 2 | CNTT |

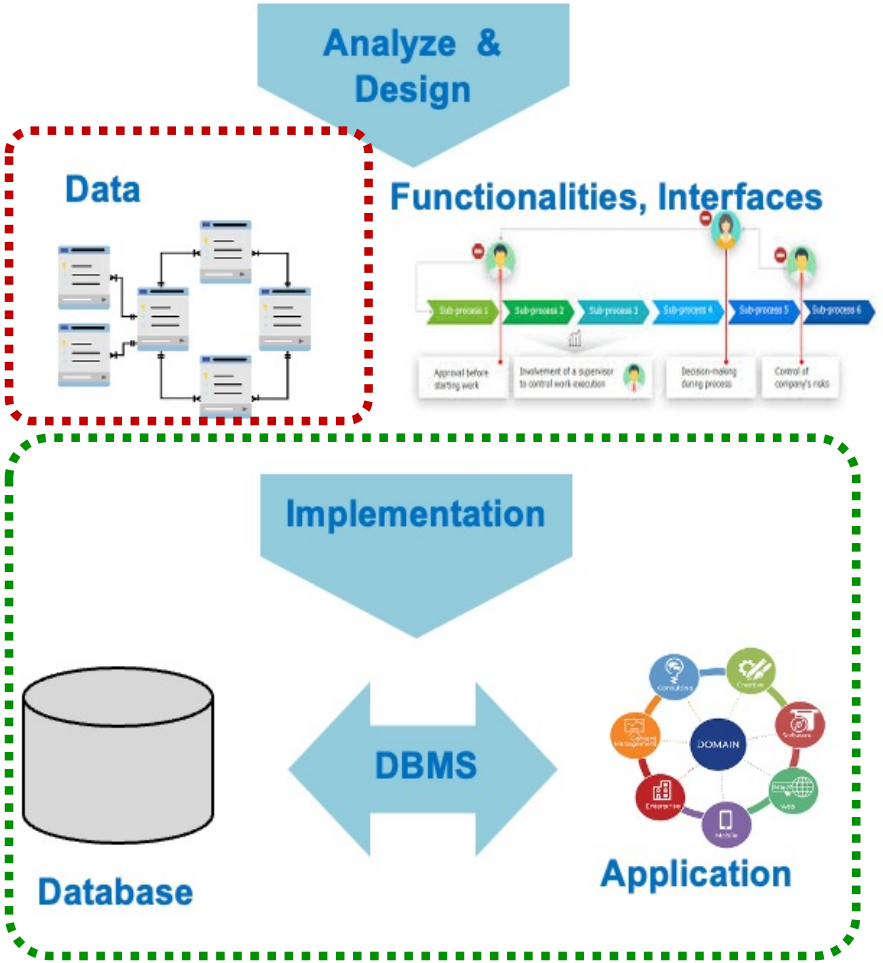
# Content

- Introduction of databases
- Evolution of database systems
- Characteristics of the database approach
- Database management systems (DBMS)
- Data models
- Database languages
- Database users

# Databases Design Process

**HOW** to represent data structures and their characteristics, relations, constraints, etc.?

**Data models**
Entity Relationship Model
Relational Model
Object-oriented Model
Network Model
…




**Database languages**
**SQL**
Relational Algebra
Relational Calculus
(& Integrity Constraints)

**HOW** to define, operate, and manipulate (relational) database?

# Data Languages
## (for Relational Model)

- High-level (practical) language
  - Entered interactively from a display monitor/terminal or embedded in a general-purpose programming language.
  - E.g.: **SQL** Standard (extension: T-SQL, PL/SQL).
- Low-level (formal) language
  - Embedded in a general-purpose programming language.
  - Offer the fundamental principles for the practical language.
  - E.g.: **Relational Algebra**, **Relation Calculus**

# SQL Standard

- **DDL** – Data Definition Language
  - Identify descriptions of the **schema constructs and store** them in the DBMS catalog.
- **DML** – Data Manipulation Language
  - Provide a set of operations including **retrieval**, **insertion**, **deletion** and **modification** of the data.
- **SDL** – Storage Definition Language
  - Specify the internal schema and the mappings between two schemas.
- **VDL** – View Definition Language.
  - Specify user views and their mapping to the conceptual schema.

# Example of SQL DDL

```
CREATE TABLE GIAOVIEN (
        MAGV        CHAR(9) PRIMARY KEY,
        HOTEN       NVARCHAR(50) NOT NULL,
        LUONG       INT DEFAULT (1000),
        PHAI        CHAR(3) CHECK (PHAI IN( 'Nam' , 'Nu' )),
        NGAYSINH    DATETIME,
        SONHA       NVARCHAR(10),
        DUONG       NVARCHAR(50),
        QUAN        NVARCHAR(50),
        THANHPHO    NVARCHAR(50),
        GVQLCM      CHAR(9),
        MABM        CHAR(9)
)
```

cdio 4.0

fit@hcmus

# Example of SQL DML
## (Create – Read – Update - Delete)

**C**

```
INSERT INTO THAMGIADT VALUES('002', '001', 2, 1.2, NULL)
```

**R**

```
SELECT MAKHOA, TENKHOA, PHONG FROM KHOA
WHERE PHONG='I53' AND NAMTL = '1995'
```

**U**

```
UPDATE GIAOVIEN SET LUONG=LUONG*1.1
```

**D**

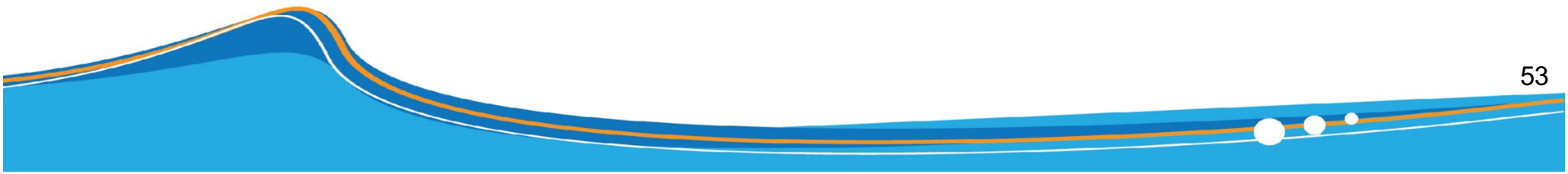
```
DELETE FROM GIAOVIEN WHERE MAGV = 'GV001'
```

# Content

- Introduction of databases
- Evolution of database systems
- Characteristics of the database approach
- Database management systems (DBMS)
- Data models
- Database languages
- Database users

# Database Administrator

- Many people use the same resources
  - Need a chief administrator to oversee and manage
- Responsibility
  - Administering the DB
  - Authorizing access to the DB
  - Coordinating and monitoring the use of DB
  - Acquiring software and hardware resources as needed



# Database Designer

- Responsibility
  - Identifying the data to be stored in the DB
  - Choosing appropriate structures to represent and store the DB
  - Communicating with all DB users to understand their requirements, to produce a design that meet the requirements.

- Can be
  - Staff of the DBA
  - Other staffs taking responsibilities after the DB designed is completed

# End User

- People whose jobs require to access to the DB
  - Querying, updating, generating reports
- Categories
  - Casual end user
  - Naïve or parametric end user
  - Sophisticated end user