Chapter 1
Overview of Database
Content

- Introduction
- The evolution of database systems
- Characteristics of the database approach
- Database users
- Architecture of a DBMS
- Properties of DBMS
- Data models
- Database languages
Introduction

Examples

- Banking and finance
  - Customer information, accounts, loans, banking transactions
  - Information of holdings, sales and purchases

- Education
  - Student information, course registrations and grades

- Airline
  - Reservations and information of flights and ticket prices

- Human resources
  - Information about employees, salaries, payroll taxes

- ...
Introduction

- Data
  - Facts that can be recorded and have meaning
  - Pieces of data are individual pieces of information

- Example
  - Name, address, phone number of customers
  - Name, address, salary, tax status of employees
  - Printing of reports such as sale, purchase, bill…
  - Tracking inventories of items in warehouses/stores
  - …
Introduction

- Database (DB)
  - A collection of related data
  - Contains information relevant to an enterprise

- Example: Corporate records
  - Sale, purchase
  - Payable and receivable accounts
  - Employees
  - Printing of employee’s weekly paychecks
Introduction

- Database (DB)
  - Represents some aspect of the real world
  - 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
Introduction

- Database Management System (DBMS)
  - A collection of programs that enables users to create and maintain a database

- A general-purpose software system that facilitates
  - **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
Introduction

- Database System

![Diagram showing the components of a database system including Users/Programmers, Applications/Queries, DBMS, Query/Program Processing, Data Accessing, Database Definition, Database, Catalog, and DB system.]
## Example

### Employee

<table>
<thead>
<tr>
<th>LNAME</th>
<th>MNAME</th>
<th>FNAME</th>
<th>SSN</th>
<th>BIRTHDATE</th>
<th>SUPERSSN</th>
<th>DNO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tran</td>
<td>Hong</td>
<td>Quang</td>
<td>987987987</td>
<td>03/09/1969</td>
<td>987654321</td>
<td>4</td>
</tr>
<tr>
<td>Nguyen</td>
<td>Thanh</td>
<td>Tung</td>
<td>333445555</td>
<td>12/08/1955</td>
<td>888665555</td>
<td>5</td>
</tr>
<tr>
<td>Nguyen</td>
<td>Manh</td>
<td>Hung</td>
<td>666884444</td>
<td>09/15/1962</td>
<td>333445555</td>
<td>5</td>
</tr>
<tr>
<td>Tran</td>
<td>Thanh</td>
<td>Tam</td>
<td>453453453</td>
<td>07/31/1972</td>
<td>333445555</td>
<td>5</td>
</tr>
</tbody>
</table>

### Project

<table>
<thead>
<tr>
<th>PNAME</th>
<th>PNUMBER</th>
<th>PLOCATION</th>
<th>DNUM</th>
</tr>
</thead>
<tbody>
<tr>
<td>San pham X</td>
<td>1</td>
<td>VUNG TAU</td>
<td>5</td>
</tr>
<tr>
<td>San pham Y</td>
<td>2</td>
<td>NHA TRANG</td>
<td>5</td>
</tr>
<tr>
<td>San pham Z</td>
<td>3</td>
<td>TP HCM</td>
<td>5</td>
</tr>
<tr>
<td>Tin hoc hoa</td>
<td>10</td>
<td>HA NOI</td>
<td>4</td>
</tr>
</tbody>
</table>

### Works on

<table>
<thead>
<tr>
<th>SSN</th>
<th>PNO</th>
<th>HOURS</th>
</tr>
</thead>
<tbody>
<tr>
<td>123456789</td>
<td>1</td>
<td>32.5</td>
</tr>
<tr>
<td>123456789</td>
<td>2</td>
<td>7.5</td>
</tr>
<tr>
<td>666884444</td>
<td>3</td>
<td>40.0</td>
</tr>
<tr>
<td>453453453</td>
<td>1</td>
<td>20.0</td>
</tr>
</tbody>
</table>
Example

- Company database - project management
  - Definition
    - Specify the structure of records, including data elements, data types
  - Construction
    - Store data to represent an employee, project, department... as a record
  - Manipulation
    - Querying: “Select the employees whose department is 5”
    - Updating: “Move the employee Nguyen Thanh Tung to department 1”
Content

- Introduction
- The evolution of database systems
- Characteristics of the database approach
- Database users
- Architecture of a DBMS
- Properties of DBMS
- Data models
- Database languages
Evolution

- File

An application program has its own data
Evolution

- Limitations
  - Data redundancy
    - Wasted storage space
    - Opportunities of the inconsistency
  - Data sharing is limited
  - Difficult recovery
  - Low security

- But, still be used in some applications
  - Small size DB
    - Storing and accessing data only, not including other processing operations
  - Fee costs less
    - Operation or maintenance
Evolution

- Database

- Application 1
- Application 2
- Application 3

DBMS

Database
Content

- Introduction
- The evolution of database systems
- **Characteristics of the database approach**
  - Self-describing
  - Insulation between programs and data
  - Data abstraction
  - Views of data
  - Sharing of data
- Database users
- Architecture of a DBMS
- Properties of DBMS
- Data models
- Database languages
Self-Describing

- The DB system contains not only the DB itself, but also a complete definition/description of the DB structure
- The definitions are stored in *catalog*
  - Contains information such as the structure of data, type and storage format of data items, and constraints on the data
- Information stored in catalog is called *meta-data* (data of data)
- Many applications can access to the DB
  - Refer to catalog, knowing the structure of files in specific DB (type and format of data)
Insulation

- The structure of data is stored in *catalog* separately from the access programs
  - Program-Data independence

- A little change in the structure happens
  - Application programs are rarely revised
Data abstraction

- The DB system provides a **conceptual representation** of the data to hide certain details of how the data are stored and maintained.

- **Example**
  - *Data model* is a type of data abstraction
    - Objects
    - Properties
    - Relationships
  - These logical concepts are easier for user to understand than computer storage concepts.
Views of data

- A DB has many users
- Each user may require a different *perspective or view* of the database

- A view may be
  - A subset of the database
  - Aggregate data that are derived from the database
Sharing of data

- A multiuser DBMS
  - Allows users to access the DB at the same time
  - Data for many applications are to be integrated and maintained in *a single DB*

- Using concurrency control mechanisms to access the data reasonably
  - Avoid data contention
  - Ensure the data will always be valid when they are accessed
Content

- Introduction
- The evolution of database systems
- Characteristics of the database approach

**Database users**
- Database administrator (DBA)
- Database designer
- End user

- Architecture of a DBMS
- Properties of DBMS
- Data models
- Database languages
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 come up with 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
End user

- People whose jobs require to access to the DB
  - Querying, updating, generating reports

- Categories
  - Casual end user
    - Occasionally access the DB
    - Need different information each time
    - Use sophisticated DB query language to specify requests
    - Middle or high level manager
  - Naïve or parametric end user
  - Sophisticated end user
End user

- People whose jobs require to access to the DB
  - Querying, updating, generating reports

- Categories
  - Casual end user
  - Naïve or parametric end user
    - Constantly query and update the DB
    - Use standard types of queries and updates that have been programmed and tested
    - Employee
  - Sophisticated end user
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
    - Be familiar with the facilities of the DBMS
    - Implement the applications to meet the complex requirements
    - Engineers, scientists, business analysts
Content

- Introduction
- The evolution of database systems
- Characteristics of the database approach
- Database users
  - Architecture of a DBMS
- Properties of DBMS
- Data models
- Database languages
Architecture

- Three-schema architecture

**External level**

- End user 1
- External view 1
- ... 
- External view n

**Conceptual level**

- Conceptual schema

**Internal level**

- Internal schema

The part of the DB that a particular user group is interested in

The structure of the whole DB for a community of users

Physical storage structure of the DB
Introduction

Data independence

- Logic data independence
  - The capability to change the conceptual schema without any change external schemas or application programs
  - Example
    - Adding/removing a record type or data item (expand/reduce DB)
    - Changing constrains

- Physical data independence
  - The capability to change the internal schema without having any change the conceptual schema
  - Example
    - Physical files had to be reorganized to improve the performance of retrieval or update
Architecture of a DBMS

Naive Users
- Forms

Casual Users
- Application Front ends

Application Programmers
- DML Interface
  - Plan Executor
  - Parser
  - Optimizer
  - Operator Evaluator

Database Administrator
- DDL
  - DDL Compiler
  - Data Definition Language
  - Call Language Interface

DBMS Concurrency Control
- Transaction Manager
- Lock Manager

Query Evaluation Engine
- SQL Commands

File/Access Methods
- Disk Management
- Buffer Management

System Catalog
- Index files
- Data files

SQL Commands

Data Manipulation Language

Query Evaluation Engine

Data Definition Language

Call Language Interface

DDL

Naive Users

Casual Users

Application Programmers

Database Administrator
Content

- Introduction
- The evolution of database systems
- Characteristics of the database approach
- Database users
- Architecture of a DBMS
- Properties of DBMS
- Data models
- Database languages
Properties of DBMS

- **Controlling redundancy**
  - By placing all the data together, we do not have to search multiple files to collect this data

- **Data sharing**
  - In multiple user environment, concurrency data access is allowed

- **Restricting unauthorized access**
  - Users or user groups are given account numbers protected by passwords to gain access to the DB

- **Providing multiple user interfaces**
  - Provide query languages for casual users, programming language interfaces for programmers, forms and command codes for parametric users
Properties of DBMS

- Enforcing integrity constraints
  - Integrity constraints
    - Rules/conditions are derived from the meaning/semantics of the data or the miniworld it represents
  - Some constraints
    - Can be specified to the DBMS and automatically enforced
    - May have to be checked by update programs

- Providing backup and recovery
  - Provide facilities for recovering from hardware and software failures
  - Make sure the DB is restored to the state it was before
Properties of DBMS

- **Others**
  - Potential for enforcing standards
    - Permit DBA to define and enforce standards among database users in a large organization
  - Flexibility
    - It may be necessary to change the structure of a DB as requirements change without affecting the stored data and the existing application programs
  - Reduced application development time
  - Availability of up-to-date information
    - As soon as one user’s update is applied to the DB, all other users can immediately see this update
Content

- Introduction
- The evolution of database systems
- Characteristics of the database approach
- Database users
- Architecture of a DBMS
- Properties of DBMS
- Data models
- Database languages
Data models

Definition
- A collection of **concepts** that can be used to describe the **structure of a DB**
  - Data types, relationships, and constraints
- Including a set of basic **operations** for specifying retrievals and updates on the DB

Categories
- High level or conceptual data models
- Representational or implementation data models
- Low level or physical data models
Data models

- High level data model
  - Provide concepts that are close to the way users perceive data
  - Eg: entity relationship model, object-oriented model…

- Implementation data model
  - Provide concepts that may be understood by end users, but that are not too far from the way data is organized within the computer
  - Eg: relational model, network and hierarchical models…

- Low level data model
  - Provide concepts the describe the details of how data is stored in the computer
Example of ER Model

**ER Diagram**

- **Student**
  - ID
  - Name
  - Grade
  - Major

- **Course**
  - ID
  - Name
  - Faculty
  - NoOfCredits

- **Subject**
  - ID
  - Name
  - Faculty

- **Study**
  - Mark

- **Open**
  - ID

- **Term**
  - Year
  - Instructor

Constraints:
- (0,n) relationship between **Study** and **Course**
- (1,1) relationship between **Course** and **Term**
- (0,n) relationship between **Subject** and **Open**
- (0,n) relationship between **Term** and **Open**
Example of Object-Oriented Model

- Student
  - Name
  - Grade
  - Major
  - planSchedule()
  - printReord()

- Mark
  - LabMark
  - LectureMark
  - ProjectMark
  - UpdateMark()

- Course
  - Name
  - Number
  - 0..*

- Subject
  - Name
  - Faculty
  - NoOfCredits
  - UpdateCredit()
  - 0..*
  - Constraint
    - +pre
    - 0..*
    - +post

- Study
  - 1..*
  - open
  - 0..*

Example of Object-Oriented Model

- Student
  - Name
  - Grade
  - Major
  - planSchedule()
  - printReord()

- Mark
  - LabMark
  - LectureMark
  - ProjectMark
  - UpdateMark()

- Course
  - Name
  - Number
  - 0..*

- Subject
  - Name
  - Faculty
  - NoOfCredits
  - UpdateCredit()
  - 0..*
  - Constraint
    - +pre
    - 0..*
    - +post

- Study
  - 1..*
  - open
  - 0..*
### Example of relational model

#### Student
- **StuID**
- **Name**
- **Grade**
- **Major**

#### Study
- **StuID**
- **CouID**
- **LabMark**
- **LectureMark**

#### Course
- **CouID**
- **Number**
- **SubID**

#### Constraint
- **SubID**
- **PreviousID**

#### Subject
- **SubID**
- **Name**
- **Faculty**
- **NoOFCredits**

---

[CuuDuongThanhCong.com](https://fb.com/tailieudientucntt)
Example of network data model

- Student
- Subject
- Course
- Constraint

Connections:
- Student_Mark
- Subject_Open
- Result_Course
- PreSubject
- PostSubject

CuuDuongThanCong.com
https://fb.com/tailieudientucntt
Example of hierarchical data model

Level 1:

*Result*

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>LabMark</td>
<td>LectureMark</td>
</tr>
</tbody>
</table>

Level 2:

*Course*

<table>
<thead>
<tr>
<th>Name</th>
<th>Number</th>
</tr>
</thead>
</table>

*Student*

| Name | Grade | Major |

Level 3:

*Subject*

| Name | Faculty | NoOfCredits |

https://fb.com/tailieudientucntt
Content

- Introduction
- The evolution of database systems
- Characteristics of the database approach
- Database users
- Architecture of a DBMS
- Properties of DBMS
- Data models
- Database languages
Database language

- **DDL – Data Definition Language**
  - Identify descriptions of the schema constructs
  - Store the schema description in the DBMS catalog

- **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
Database language

- **DML – Data Manipulation Language**
  - Provide a set of operations including retrieval, insertion, deletion and modification of the data

- **Two types**
  - **High level (nonprocedural)**
    - Entered interactively from a display monitor/terminal
    - Embedded in a general-purpose programming language
  - **Low level (procedural)**
    - Must be embedded in a general-purpose programming language
Discussion

- When will we use or not use the DB approach?