Database

Instructor: Hoa Dinh Nguyen, PhD.
hoand@ptit.edu.vn
0942807711

Introduction

Course Format

- Lectures: 16 sessions
  Content: tutorials, individual exercises, term project
- Practice: 2 sessions
  - Lab room
- Grading:
  - Attendance: 10%
  - Term project: 30%
  - Final exam: 60%

Communications

- Classroom:
  - Syllabus is there
  - Lecture slides will be available there
  - Other announcements will be available there
- Email:
  - Email: hoand@ptit.edu.vn
  - Report submission

Textbook

Main textbook:

- *Database Systems: The Complete Book*,
  Hector Garcia-Molina,
  Jeffrey Ullman,
  Jennifer Widom
  Second edition.

Covers most, but **not all**, of course content

Term Project

Design and implement a database system:

- Work in group to design a database system assigned by the lecturer
- Apply a database management system to implement and manage a database system including:
  - Setup fundamental functions for the system
  - Database Input/output
  - Making reports
- Report all your works in details.

Outline of Today's Lecture

- Overview of database
  - Definitions of data, database, database management systems, database systems.
  - Why they are helpful
  - What are some of their key features
  - What are some of their key concepts

Motivation

- The world is drowning in data
  - affects almost every app / service
- Need professionals to help manage it
  - help domain scientists achieve new discoveries
  - help companies provide better services
  - help governments become more efficient

Data Management is Universal

- Managing data is at the core of most apps / services
  - whether they store small or large amounts of data
  - whether they are modern systems or older ones
- Hard problems even with small amounts of data
  - we'll see examples later on…
- Doing it right typically makes everything else easier

An Example: Online Bookseller

- What data do we need?
  - Data about books, customers, pending orders, order histories, trends, preferences, etc.
  - Data about sessions (clicks, pages, searches)
  - Note: data must be persistent! Outlive application
  - Also note that data is large… won't fit all in memory
- What capabilities on the data do we need?
  - Insert/remove books, find books by author/title/etc., analyze past order history, recommend books, …
  - Data must be accessed efficiently, by many users
  - Data must be safe from failures, malicious users, and bugs!

Database

What is a database?

- “data” in a database can encompass a wide variety of objects from numbers, text, graphics, video, etc.
- A collection of files storing related data
  - a logically coherent collection of data with some inherent meaning
  - Random data is not typically referred to as a database

Examples of databases

- Accounts database; payroll database; PTIT’s students database; Amazon’s products database; airline reservation database

Data vs. Information

In its “raw” form, data has little meaning.
In this case it simply looks like a couple of lists of integer numbers.
There is no context on which to base the data.

Data:	
0	11,500
5	12,300
10	12,800
15	10,455
20	12,200
25	13,900
30	14,220

Data vs Information

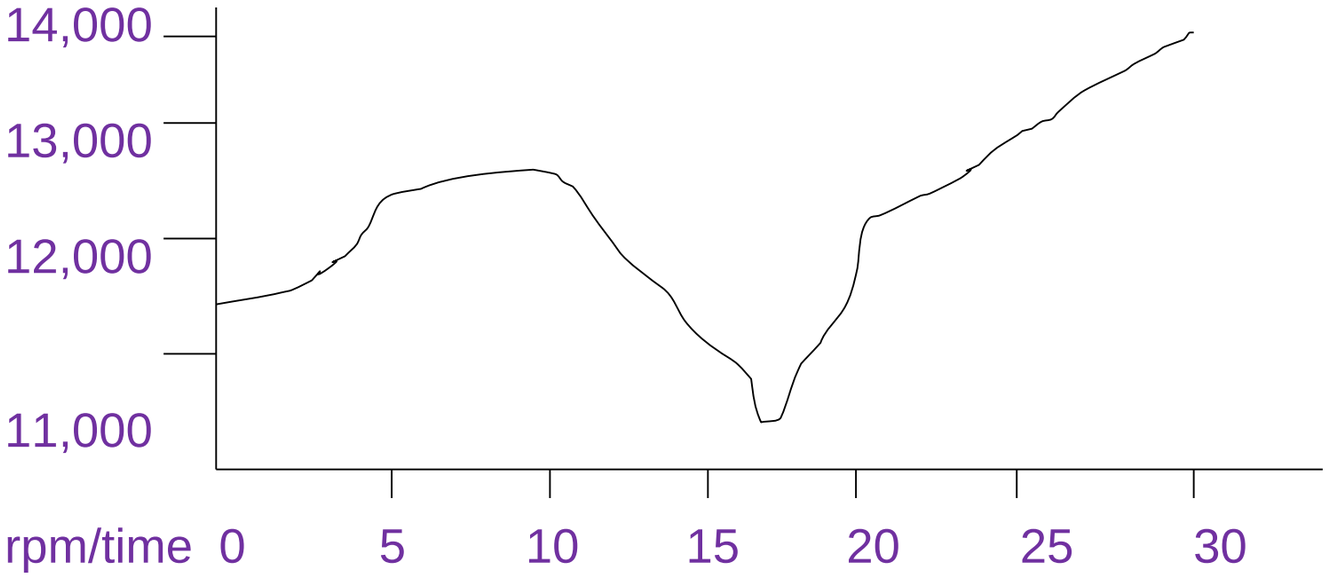
“Processing” the data is to transform it into something with more meaning.
In this example, the data is processed by putting it in context. Now the data begins to take on more meaning.

Information: Engine RPM Data: Roebling Road 10/4/2003 – Yamaha Heavy

Lap 12:	time	rpm
	0	11,500
	5	12,300
	10	12,800
	15	10,455
	20	12,200
	25	13,900
	30	14,220

Data vs Information

The same data can also by processed by graphing.



Graph: Partial Lap 12 – Roebling Road 10/4/2003 – Yamaha Heavy

Database Management System

- Is a software that allows us to manage efficiently a large database and allows it to persist over long periods of time
  - Allowing users to create new databases: using data definition languages (DDLs).
  - Allow users to query the database via data manipulation languages (DMLs).
  - Support the storage of very large amounts of data for very long periods of time.
  - Maintain database security and integrity.
  - Control access to data from many users simultaneously.

Advantages of a DBMS

control of data redundancy	economy of scale
data consistency	balance of conflicting requirements
more information from same data	improved data accessibility
amount of data available	increased productivity
sharing of data	improved maintenance
improved data integrity	increased concurrency
improved data security	improved backup and recovery
enforcement of standards	improved responsiveness

Disadvantages of a DBMS

complexity
size
cost of DBMSs
additional hardware costs
cost of conversion
performance (specific cases)
higher impact of failure

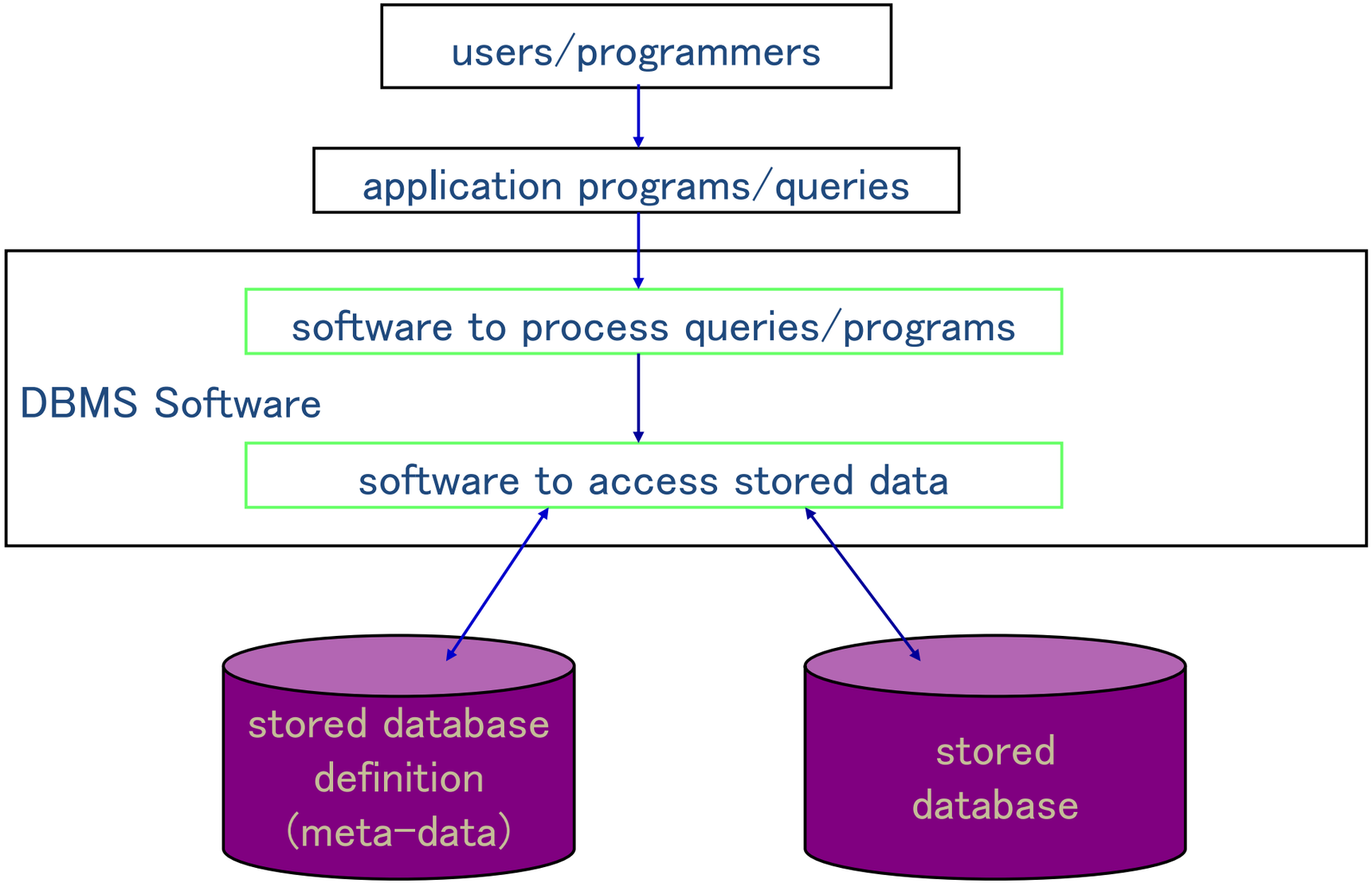
Database Management System

Examples of DBMSs

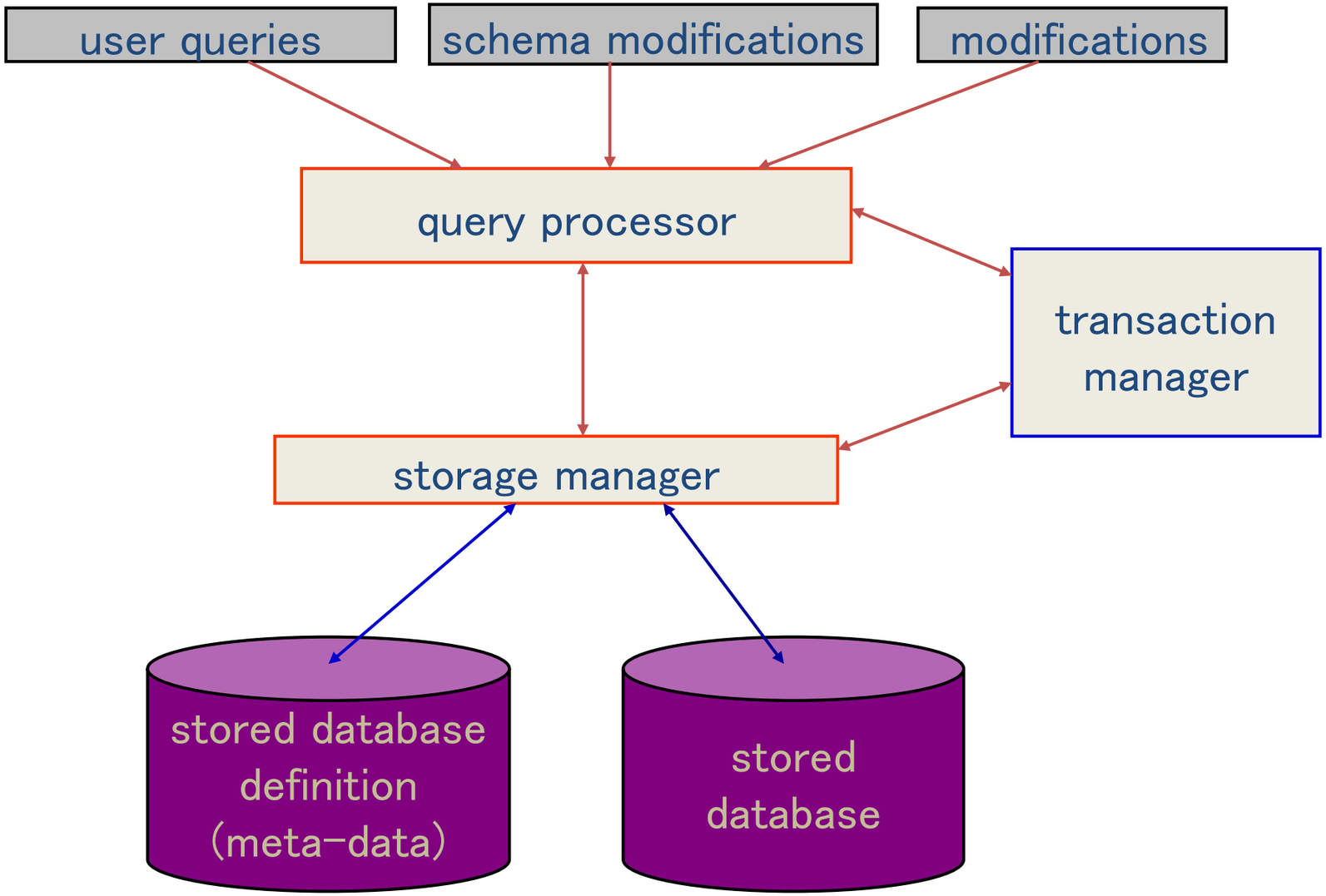
- Oracle, IBM DB2, Microsoft SQL Server, Vertica, Teradata
- Open source: MySQL (Sun/Oracle), PostgreSQL, AsterixDB
- Open source library: SQLite

A database managed by a database management system is typically referred to as a ***database system***.

Components of a Database System



Architecture of a Database system



Required Functionality for a Database system

1. Describe real-world entities in terms of stored data
2. Persistently store large datasets
3. Efficiently query & update
   - Must handle complex questions about data
   - Must handle sophisticated updates
   - Performance matters (users can feel 200ms latency)
4. Easily change structure (e.g., add attributes)
5. Enable simultaneous updates
6. Crash recovery
7. Security and integrity

Key Roles of People

- **DB application developer**: writes programs that query and modify data
- **DB designer**: establishes schema
- **DB administrator**: loads data, tunes system, keeps whole thing running
- **Data analyst**: data mining, data integration
- **DBMS implementer**: builds the DBMS

Key Concepts

- **Data models**: how to describe real-world data
  - Relational, XML, JSon
- **Schema vs data**
- **Declarative query language**
  - Say what you want, not how to get it
- **Data independence**
  - Physical independence: Can change how data is stored on disk without affecting applications
  - Logical independence: can change schema w/o affecting apps
- **Query optimizer** and compiler
- **Transactions**: isolation and atomicity

Database Languages

A data sublanguage consists of two parts: a Data Definition Language (DDL) and a Data Manipulation Language (DML).

These languages are called *data sublanguages* because they do not include constructs for all computing needs such as conditional or iterative statements

Most DBMSs have a facility for embedding the sublanguage in a high-level programming language such as COBOL, Pascal, C, C++, Java, or Visual Basic which is then called the *host language*

Most data sublanguages also provide a non-embedded or interactive version of the language to be input directly from a terminal.

Database system classification

Two types of database system architectures:

- Centralized database systems:
  - Personal database systems
  - Center database systems
  - Client/server database systems
- Distributed database systems
  - Homogeneous distributed database systems
  - Heterogeneous distributed database systems