Database System Concepts and Architecture

Assoc. Prof. Tran Minh Quang

CSE, HCMUT

quangtran@hcmut.edu.vn

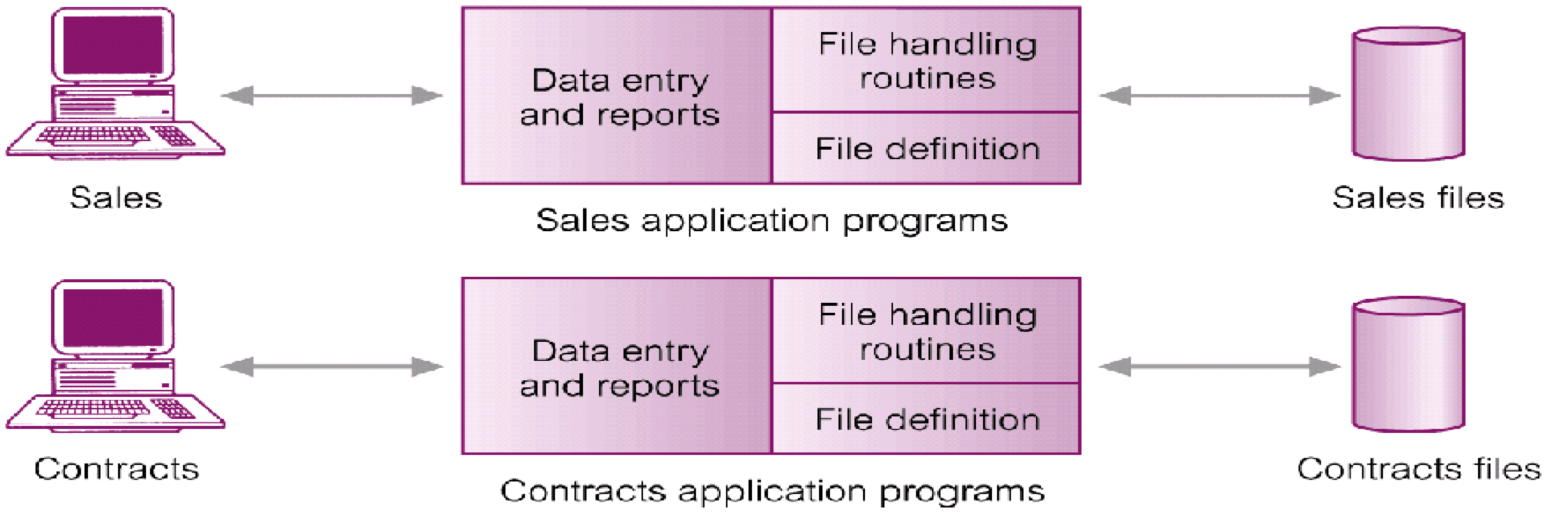
Outline

- File-based Approach
- Database Approach
  - When not to use a DBMS?
- Three-Level Architecture and Data Independence Concepts
- Database Languages
- Data Models, Database Schema and Database State
- Data Management Systems Framework
  - Where are we?
  - Extending database capabilities for new applications
- Reading Suggestion:
  - [1]: Chapters 1, 2
  - Internet

File-based Approach

- Data is stored in one or more separate computer files
- Data is then processed by computer programs - **applications**

File-based Approach



Sales

Data entry and reports

File handling routines

File definition

Sales application programs

Sales files

Contracts

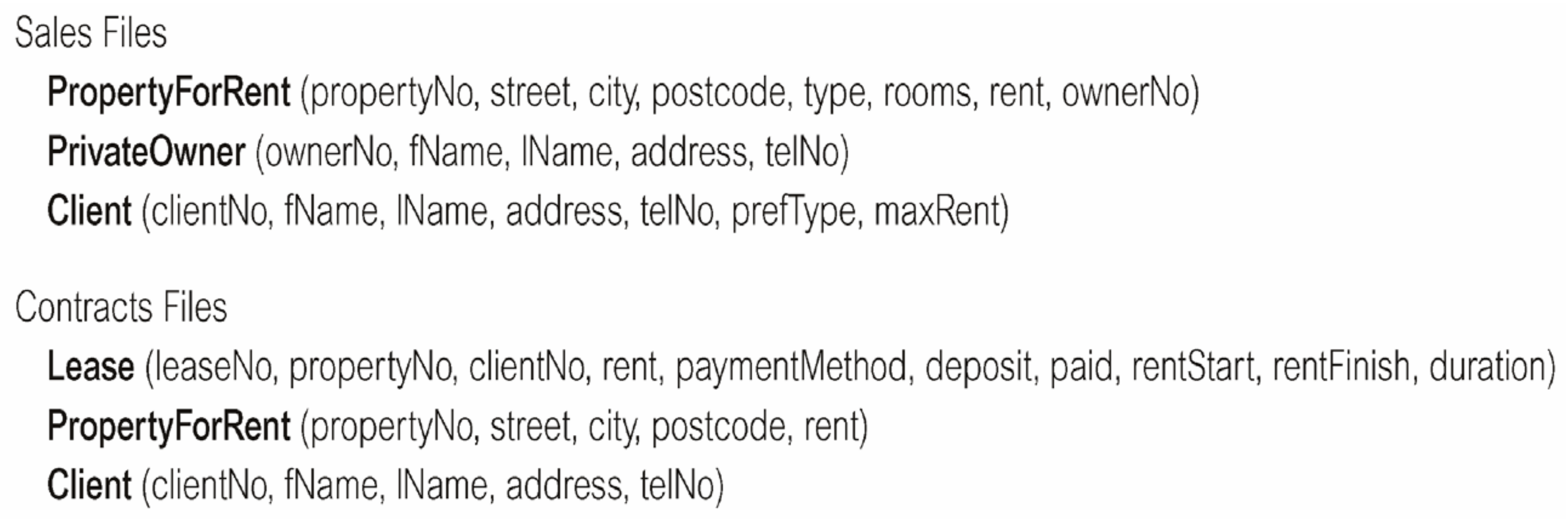
Data entry and reports

File handling routines

File definition

Contracts application programs

Contracts files



Sales Files

PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)

PrivateOwner (ownerNo, fName, lName, address, telNo)

Client (clientNo, fName, lName, address, telNo, prefType, maxRent)

Contracts Files

Lease (leaseNo, propertyNo, clientNo, rent, paymentMethod, deposit, paid, rentStart, rentFinish, duration)

PropertyForRent (propertyNo, street, city, postcode, rent)

Client (clientNo, fName, lName, address, telNo)

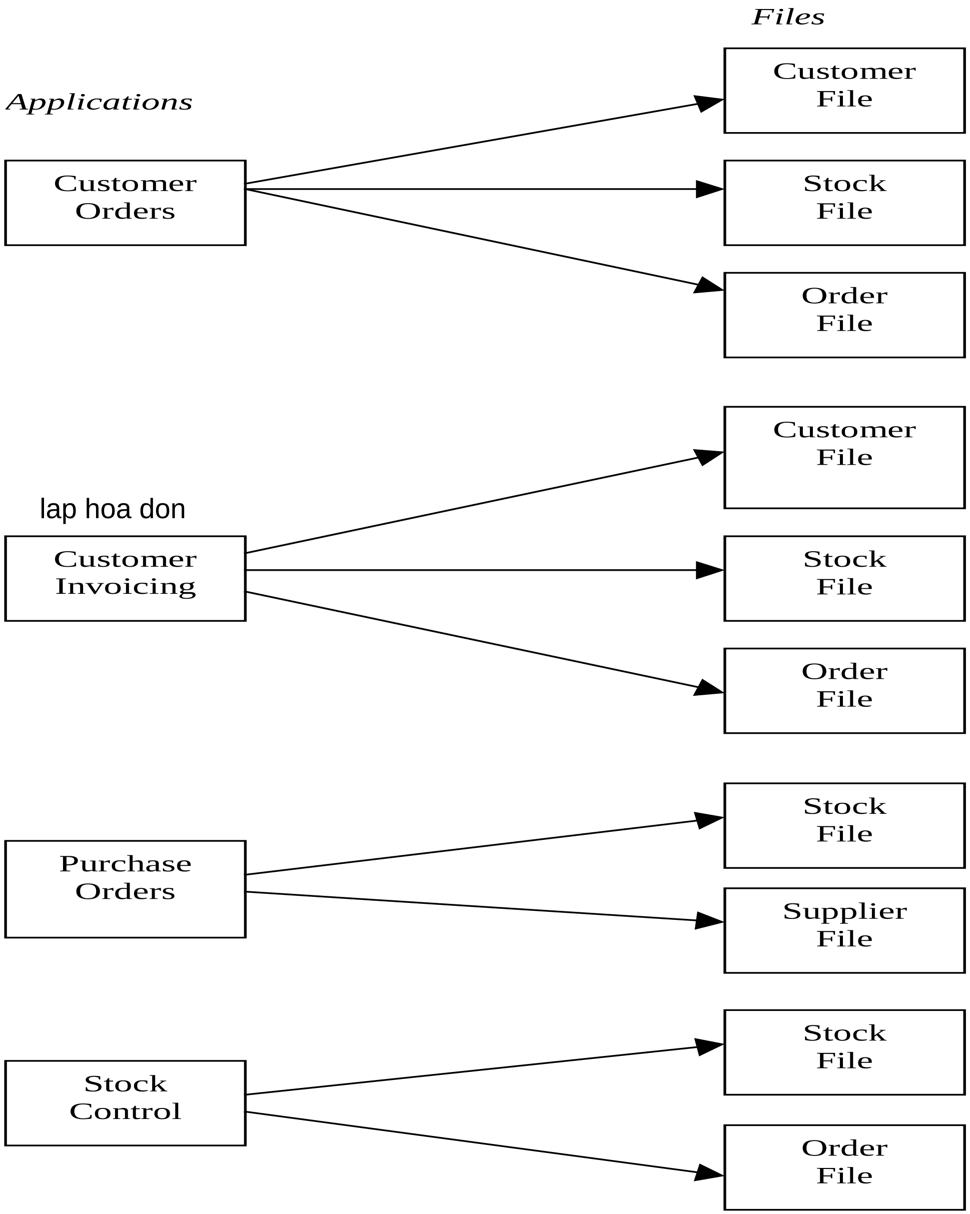
File-based Approach

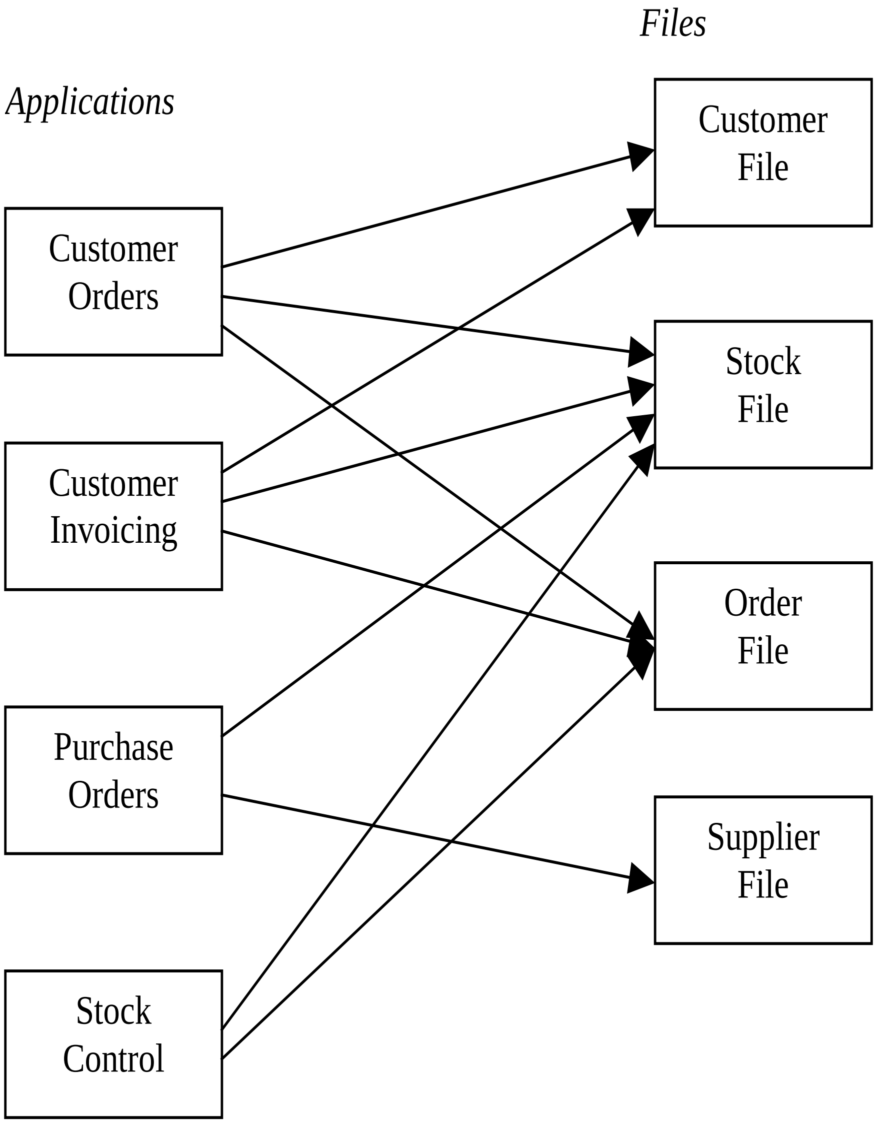
- Real-World Issues in VN universities (till now !!)
    - Staff ID: a staff have several IDs wrt. dept. of education affairs, postgrad dept., and dept. of finance & planning
    - Staff’s other related information: where does a staff really belong to?
    - Scientific CVs
    - Shared data between dept/fac/offices
    - …

File-based Approach

- Problems/Limitations
  - Data Redundancy su du thua
  - Data Inconsistency du lieu khong nhat quan
  - *More information: internet & [1]*

File-based approach





Shared file approach

File-based Approach

- Shared File Approach
  - Data (files) is shared between different applications
  - Data redundancy problem is alleviated duoc giam bot
  - Data inconsistency problem across different versions of the same file is solved
  - Other problems:

cau truc cung ngat

→Rigid data structure: If applications have to share files, the file structure that suits one application might not suit another

phu thuoc du lieu vat ly

su thay doi

→Physical data dependency: If the structure of the data file needs to be changed in some way, this alteration will need to be reflected in all application programs that use that data file

khong ho tro kiem soat dong thoi

→No support of concurrency control: While a data file is being processed by one application, the file will not be available for other applications or for ad hoc queries

Outline

- File-based Approach
- Database Approach
- Three-Level Architecture and Data Independence Concepts
- Database Languages
- Data Models, Database Schema and Database State
- Data Management Systems Framework
  - Where are we?
  - Extending database capabilities for new applications
- Reading Suggestion:
  - [1]: Chapters 1, 2
  - Internet

Database Approach

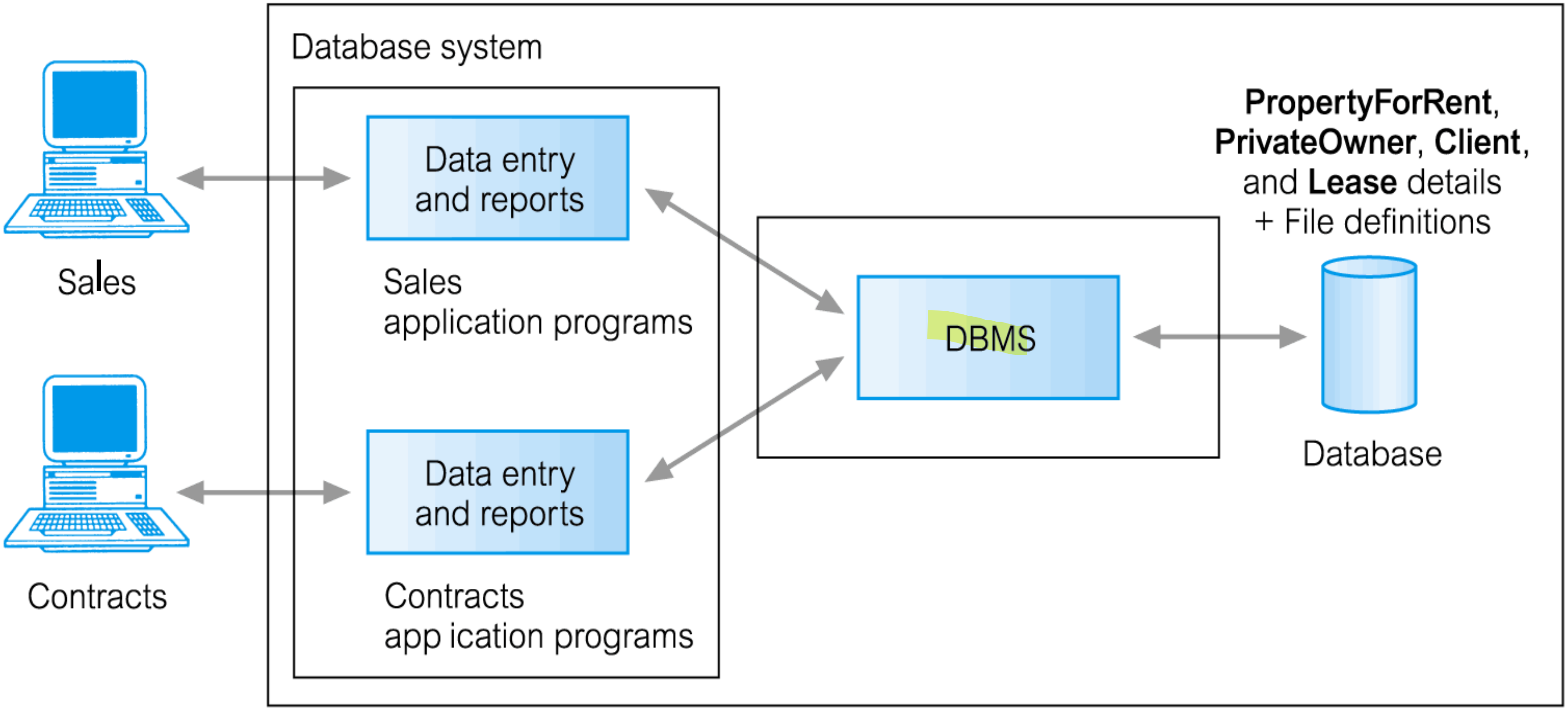
phat sinh

- Arose because:
  - Definition of data was embedded in application programs, rather than being stored separately and independently
  - No control over access and manipulation of data beyond that imposed by application programs
- Result:

khong co quyen trong viec truy cap va thao tac du lieu ngoai do chuong trinh ap dat

  - The Database and Database Management System (DBMS).

Database Approach


Database system
Sales
Data entry and reports
Sales application programs
Contracts
Data entry and reports
Contracts app ication programs
DBMS
PropertyForRent, PrivateOwner, Client, and Lease details + File definitions
Database

PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)

PrivateOwner (ownerNo, fName, lName, address, telNo)

Client (clientNo, fName, lName, address, telNo, prefType, maxRent)

Lease (leaseNo, propertyNo, clientNo, paymentMethod, deposit, paid, rentStart, rentFinish)

Database Approach

- Data
  - Known facts that can be recorded and that have implicit meaning
  - Information? Knowledge?
  - More: http://www.whatis.com ; http://en.wikipedia.org

- Database: Shared collection of logically related data and a description of this data, designed to meet the information needs of an organization

Database Approach

ho tro, tao dien kien thuan loi

muc dich chung

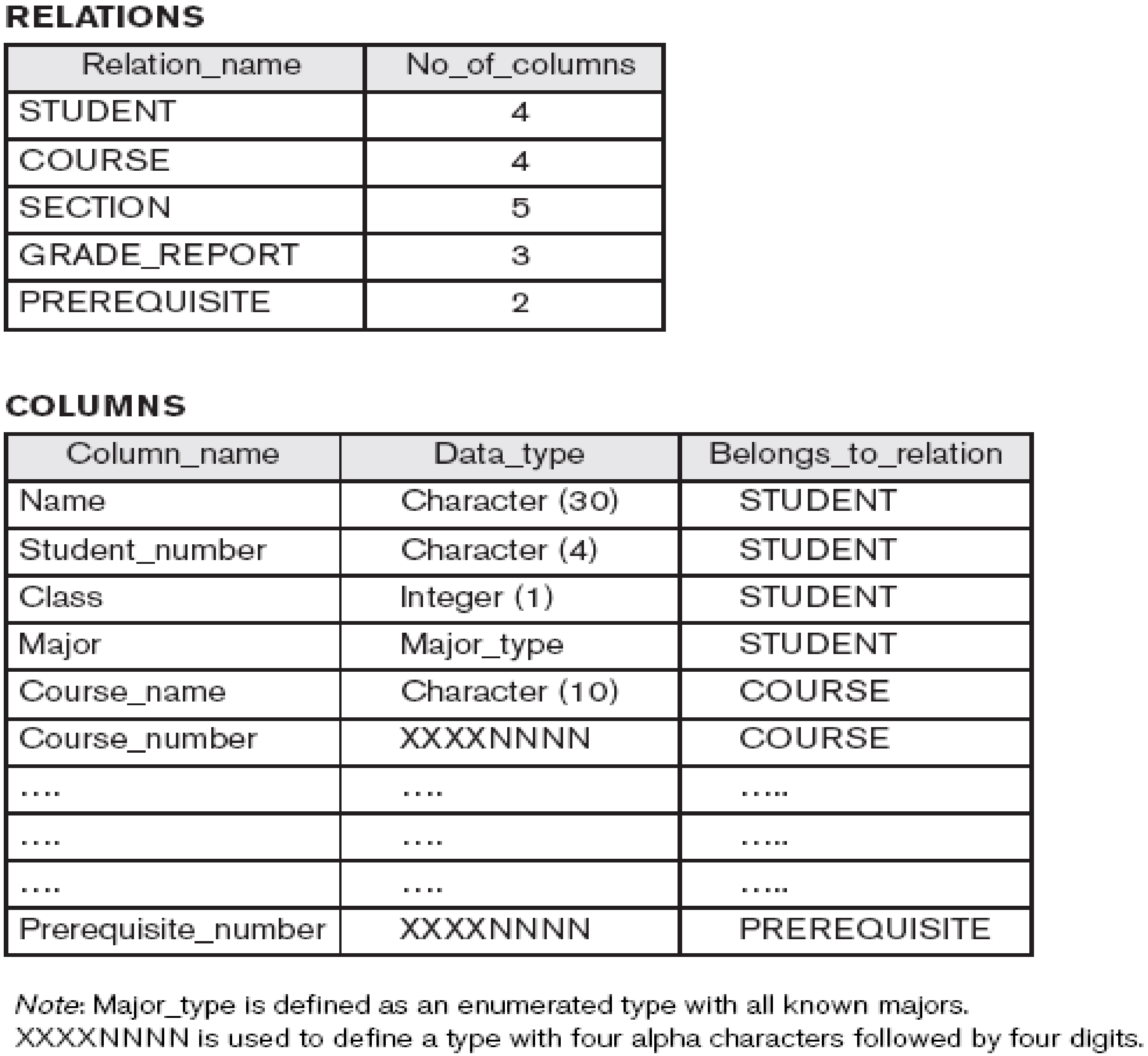
- **D**ata**B**ase **M**anagement **S**ystem (**DBMS**): a general-purpose software system that facilitates the processes of defining, constructing, manipulating, and sharing databases among various users and applications (*or a software system that enables users to define, create, maintain, and control access to the database*)

comprises

- Logically related data comprises entities, attributes, and relationships of an organization's information

- System catalog (metadata) provides description of data to enable program–data independence

Database Approach



RELATIONS

Relation_name	No_of_columns
STUDENT	4
COURSE	4
SECTION	5
GRADE_REPORT	3
PREREQUISITE	2

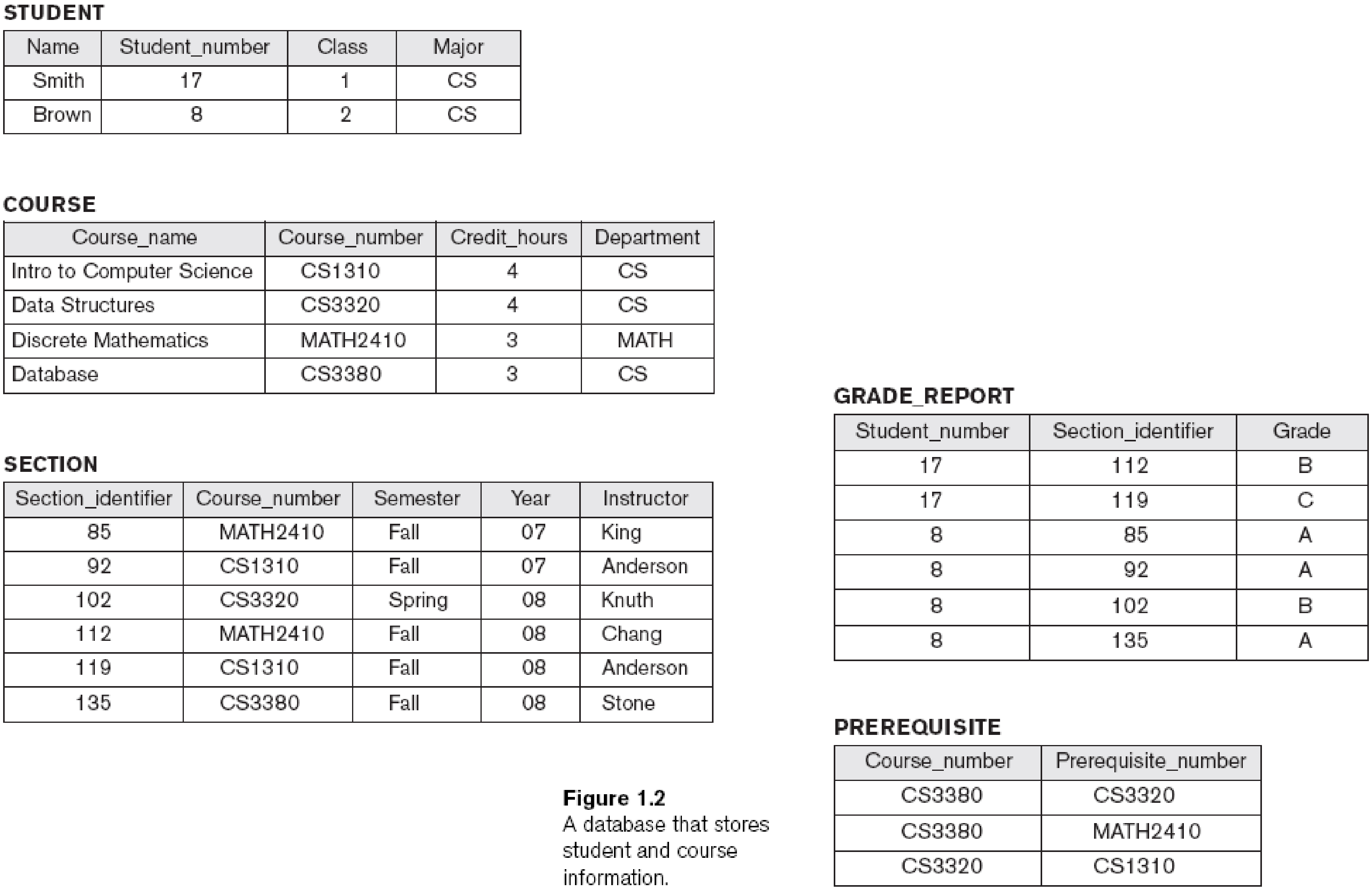
COLUMNS

Column_name	Data_type	Belongs_to_relation
Name	Character (30)	STUDENT
Student_number	Character (4)	STUDENT
Class	Integer (1)	STUDENT
Major	Major_type	STUDENT
Course_name	Character (10)	COURSE
Course_number	XXXXNNNN	COURSE
….	….	…..
….	….	…..
….	….	…..
Prerequisite_number	XXXXNNNN	PREREQUISITE

Note: Major_type is defined as an enumerated type with all known majors.
XXXXNNNN is used to define a type with four alpha characters followed by four digits.

Figure 1.3
An example of a database catalog for the database in Figure 1.2.

Database Approach



STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Figure 1.2
A database that stores student and course information.

Database Approach

- Data Definition Language (DDL) cho phep
  - Permits specification of data types, structures and any data constraints to be stored in the database
  - All specifications are stored in the database
- Data manipulation language (DML).
  - Query language: retrieve (query), update (insert, delete, modify)

truy van lay lai

- (In this course) Database System = Database + DBMS

Database Approach

- Roles in the Database Environment
  - uy quyen

    Database Administrator (DBA): responsible for authorizing access to DB, coordinating & monitoring its use, and for acquiring software and hardware resources as needed

    dieu phoi va giam sat

    co duoc, mua duoc
  - Database Designers: responsible for identifying the data to be stored in DB, choosing appropriate structures to represent and store this data
  - System analysts: determine requirements of end users
  - Tool developers: design and implement tools
  - Application Programmers
  - End Users, …
  - More details: see [1]-chapter 1

Database Approach

- DBMS components: see [1] for the details

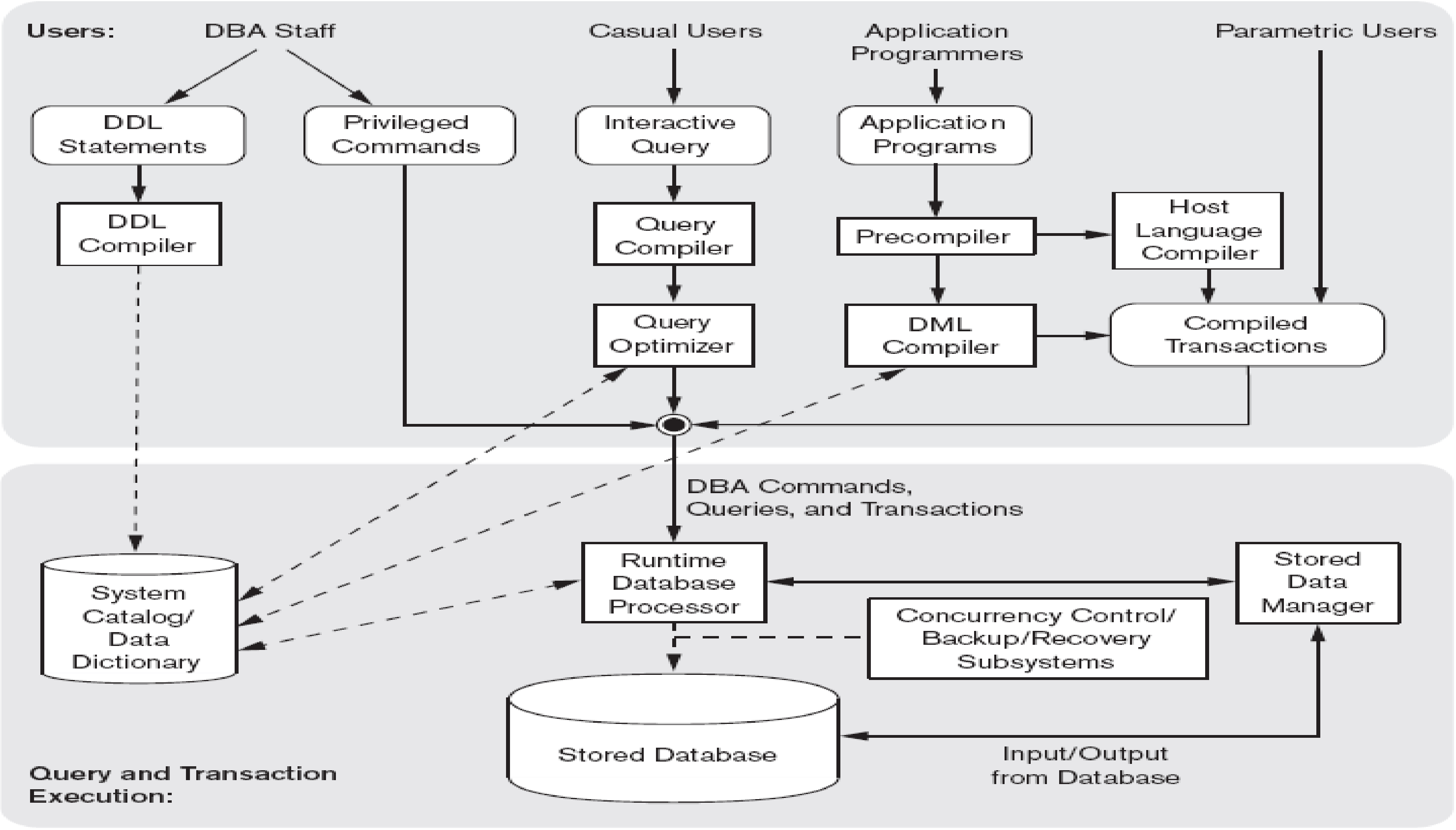


Figure 2.3
Component modules of a DBMS and their interactions.

Database Approach

dac trung

- Characteristics of the Database Approach
  - Self-describing nature of a database system
  - Insulation between programs and data, and data abstraction cach ly giua cac chtrinh va du lieu va du lieu truu tuong

    →Program-data independence + Program-operation independence = Data abstraction

    →A data model is a type of data abstraction
  - Support of multiple views of the data
  - Sharing of data and multi-user transaction processing

  - Other advantages of using the DBMS approach: [1]

Database Approach

- History of database systems
  - First generation: Hierarchical and Network
  - Second generation: Relational
  - Third generation: Object-Relational, Object-Oriented
- Brief history of database applications
  - See [1] & internet
  - Some details below …

A Brief History of Database Applications

co thu bac

- Early database applications using hierarchical and network systems
  - Large numbers of records of similar structure
- Providing data abstraction and application flexibility with relational databases
  - Separates physical storage of data from its conceptual representation
  - Provides a mathematical foundation for data representation and querying

A Brief History of Database Applications (cont'd.)

- Object-oriented applications and the need for more complex databases
  - Used in specialized applications: engineering design, multimedia publishing, and manufacturing systems

trao doi

- Interchanging data on the Web for e-commerce using XML
  - Extended markup language (XML) primary standard for interchanging data among various types of databases and Web pages

A Brief History of Database Applications (cont'd.)

- Extending database capabilities for new applications
  - Extensions to better support specialized requirements for applications

    lap ke hoach nguon luc doanh nghiep
  - Enterprise resource planning (ERP)
  - Customer relationship management (CRM)
- Databases versus information retrieval
  - Information retrieval (IR)
    - Deals with books, manuscripts, and various forms of library-based articles

      ban thao
    - Information needs rather than data needs
- Big data and emerging applications

Database Approach

When Not to Use a DBMS?

mong muon

- More desirable to use regular files for:
  - Simple, well-defined database applications not expected to change at all
  - nghiem ngat
  - Stringent, real-time requirements that may not be met because of DBMS overhead
  - Embedded systems with limited storage capacity
  - No multiple-user access to data

Outline

- File-based Approach
- Database Approach
- Three-Level Architecture and Data Independence Concepts
- Database Languages
- Data Models, Database Schema and Database State
- Data Management Systems Framework
  - Where are we?
  - Extending database capabilities for new applications
- Reading Suggestion:
  - [1]: Chapters 1, 2
  - Internet

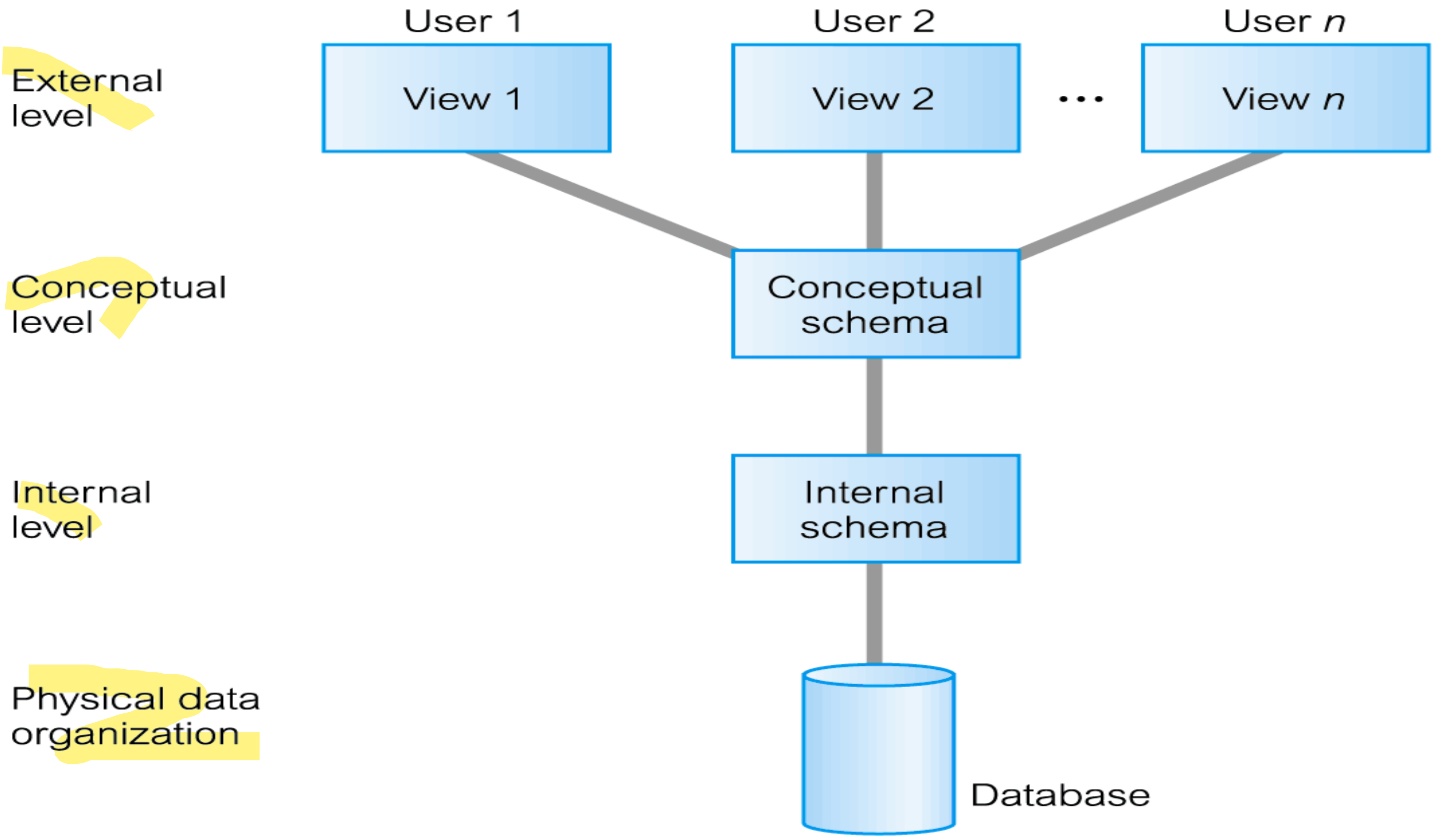
Three-Level Architecture and Data Independence

luoc do

- Objectives of Three-Schema/Level Architecture
  - All users should be able to access same data
  - A user’s view is immune to changes made in other views
  - Users should not need to know physical database storage details
  - DBA should be able to change database storage structures without affecting the users’ views
  - Internal structure of database should be unaffected by changes to physical aspects of storage
  - DBA should be able to change conceptual structure of database without affecting all users

Three-Level Architecture and Data Independence

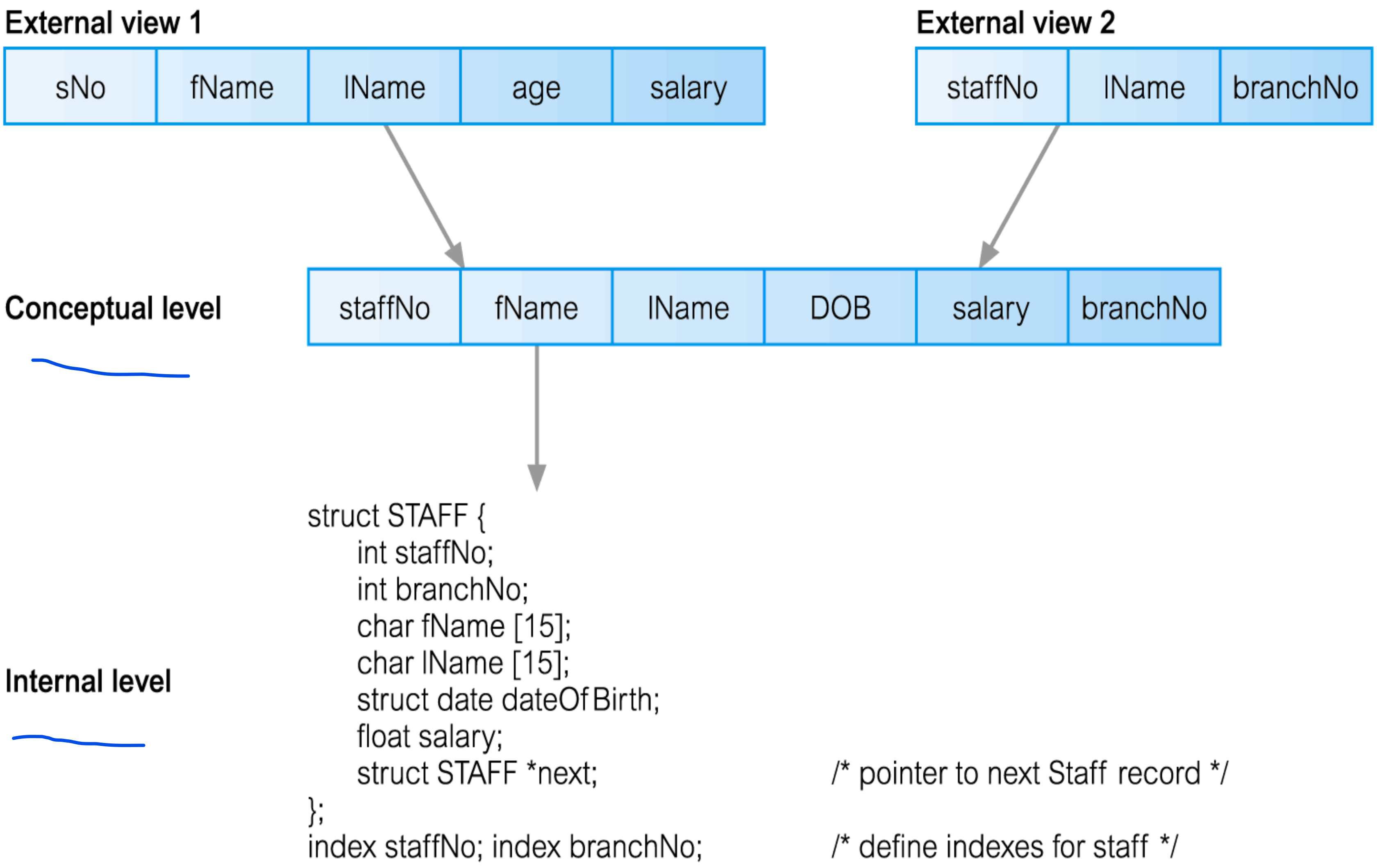
- Three-level architecture and data independence



Three-Level Architecture and Data Independence

- External Level
  - Users' view of the database
  - lien quan
  - Describes what part of database that is relevant to a particular user
- Conceptual Level
  - Community view of the database
  - Describes what data is stored in database and relationships among the data
- Internal Level
  - Physical representation of the database on the computer
  - Describes how the data is stored in the database

Three-Level Architecture and Data Independence



External view 1

sNo	fName	lName	age	salary

External view 2

staffNo	lName	branchNo

Conceptual level

staffNo	fName	lName	DOB	salary	branchNo

Internal level

```
struct STAFF {
      int staffNo;
      int branchNo;
      char fName [15];
      char lName [15];
      struct date dateOfBirth;
      float salary;
      struct STAFF *next;                  /* pointer to next Staff record */
};
index staffNo; index branchNo;             /* define indexes for staff */
```

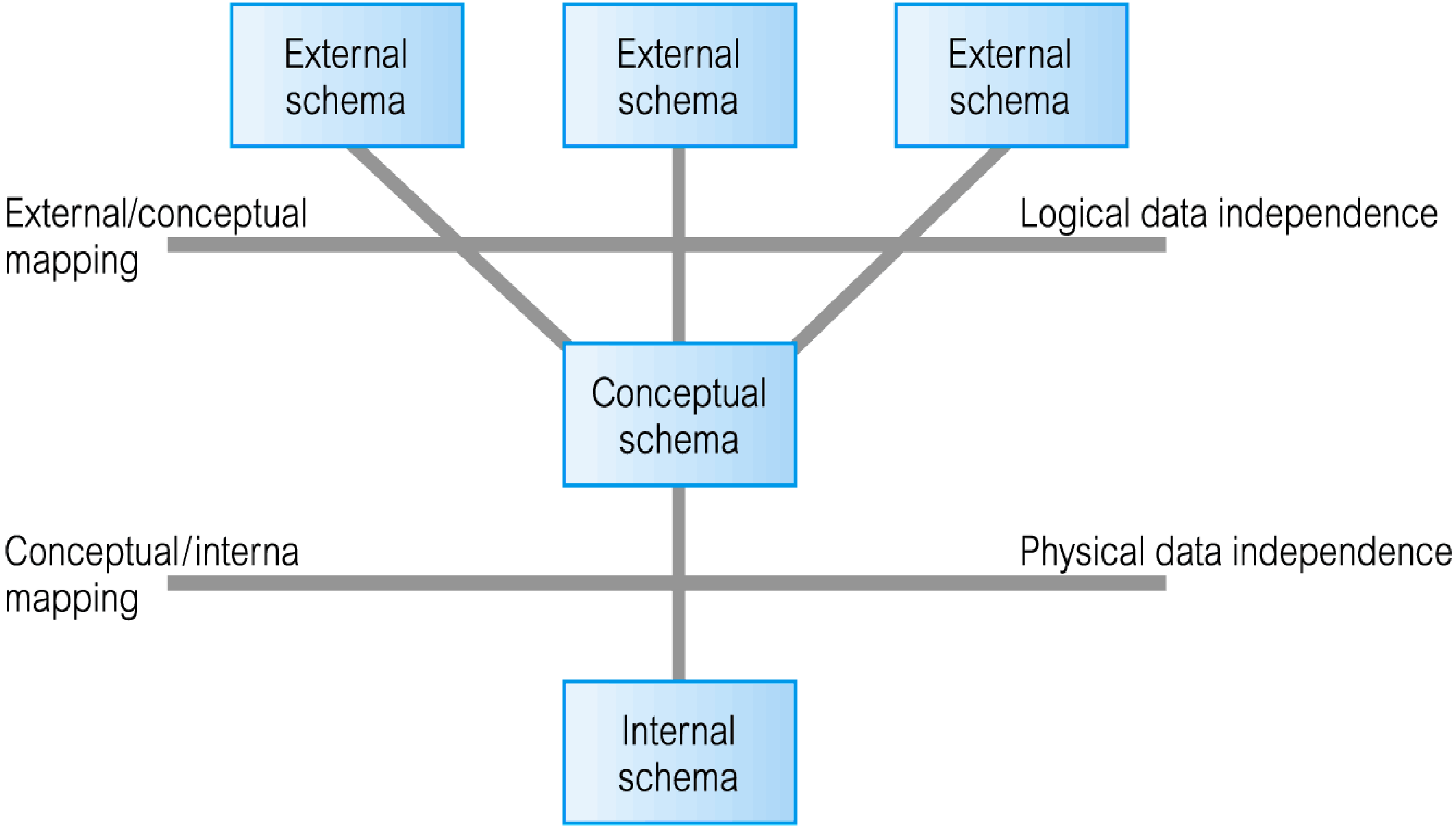
Three-Level Architecture and Data Independence

- Data Independence is the capacity to change the schema at one level of a database system without having to change the schema at the next higher level
- Logical Data Independence
  - Refers to immunity of external schemas to changes in conceptual schema
  - Conceptual schema changes (e.g. addition/removal of entities) should not require changes to external schema or rewrites of application programs

Three-level Architecture and Data Independence

- Physical Data Independence
  - Refers to immunity of conceptual schema to changes in the internal schema
  - Internal schema changes (e.g. using different file organizations, storage structures/devices) should not require changes to conceptual or external schemas

Three-Level Architecture and Data Independence


External schema
External schema
External schema
External/conceptual mapping
Logical data independence
Conceptual schema
Conceptual/interna mapping
Physical data independence
Internal schema

Outline

- File-based Approach
- Database Approach
- Three-Level Architecture and Data Independence Concepts
- Database Languages
- Data Models, Database Schema and Database State
- Data Management Systems Framework
  - Where are we?
  - Extending database capabilities for new applications
- Reading Suggestion:
  - [1]: Chapters 1, 2
  - Internet

Database Languages

- Data Definition Language (DDL) allows the DBA or user to describe and name entities, attributes, and relationships required for the application plus any associated integrity and security constraints
  tinh toan ven bao mat
- Data Manipulation Language (DML) provides basic data manipulation operations on data held in the database
- Data Control Language (DCL) defines activities that are not in the categories of those for the DDL and DML, such as granting privileges to users, and defining when proposed changes to a databases should be irrevocably made
  cap dac quyen
  de xuat thay doi
  thuc hien khong the thay doi/huy bo

Database Languages

thuc tuc cua DML

- Procedural DML allows user to tell system exactly how to manipulate data (e.g., Network and hierarchical DMLs)
  - Must be embedded in a general-purpose programming language
  - Record-at-a-time
- Non-Procedural DML (declarative language) allows user to state what data is needed rather than how it is to be retrieved (e.g., SQL, QBE)
  - Can be used on its own to specify complex database operations concisely
  - Set-at-a-time or set-oriented
- Fourth Generation Languages (4GLs)
  - Non-procedural languages: SQL, QBE, etc.
  - Application generators, report generators, etc. (see internet)

Outline

- File-based Approach
- Database Approach
- Three-Level Architecture and Data Independence Concepts
- Database Languages
- Data Models, Database Schema and Database State
- Data Management Systems Framework
  - Where are we?
  - Extending database capabilities for new applications
- Reading Suggestion:
  - [1]: Chapters 1, 2
  - Internet

Data Models, Database Schema and Database State

tich hop

- Data Model: An integrated collection of concepts for describing data, relationships between data, and constraints on the data in an organization
- Categories of data models include:
  - Object-based (Conceptual)
    →ER, Object-Oriented, …
  - Record-based (Representational)
    →Relational, Network, Hierarchical

  Describe data at the conceptual & external levels

  - Physical: used to describe data at the internal level

- Homework: study the network & hierarchical data models (resources: [1] & the Web)

Data Models, Database Schema and Database State

- Database Schema: the description of a database, which is specified during database design and is not expected to change frequently
  - Schema Diagram: a displayed schema
- Database State (Snapshot): the data in the database at a particular moment in time

Outline

- File-based Approach
- Database Approach
- Three-Level Architecture and Data Independence Concepts
- Database Languages
- Data Models, Database Schema and Database State
- Data Management Systems Framework
  - Where are we?
  - Extending database capabilities for new applications
- Reading Suggestion:
  - [1]: Chapters 1, 2
  - Internet

Data Management Systems Framework

- Where are we?

Application Layer	Visualization, Collaborative Computing, Mobile Computing, Knowledge-based Systems
Data Management Layer	**Layer 3: information extraction & sharing** Data Warehousing, Data Mining, Internet DBs, Collaborative, P2P & Grid Data Management
	Layer 2: interoperability & migration kha nang tuong tac va di chuyen Heterogeneous DB Systems, Client/Server DBs, Multimedia DB Systems, Migrating Legacy DBs
	Layer 1: DB technologies DB Systems, Distributed DB Systems
Supporting Layer	Networking, Mass Storage, Agents, Grid Computing Infrastructure, Parallel & Distributed Processing, Distributed Object Management

Data Management Systems Framework

- Extending database capabilities for new applications
  - Example applications: storage and retrieval of images, videos, data mining (large amounts of data need to be stored and analyzed), spatial databases, time series applications, …

    khong gia
  - More complex data structures than relational representation
  - New data types except for the basic numeric and character string types

    ngoai tru
  - New operations and query languages for new data types
  - New storage and retrieval methods
  - New security mechanisms
  - …
- Big data: beyond DBMS capabilities

Summary

- File-based Approach
- Database Approach
  - Database: Collection of related data (recorded facts)
- Three-Schema Architecture and Data Independence
- Database Languages, Data Models. Database Schema and Database State
- Data Management Systems Framework (where are we?)
- Database applications have evolved, and current trends: IR, Web, big data
- Reading Suggestion & Homework: do not forget !!
- Next lecture: ER Model

Q&A

Question ?