Database System Concepts and Architecture

Assoc. Prof. Dr. Dang Tran Khanh

Department of Information Systems

Faculty of CSE, HCMUT

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

File-based Approach

- Data is stored in one or more separate computer files

- Data is then processed by computer programs - **applications**

File-based Approach
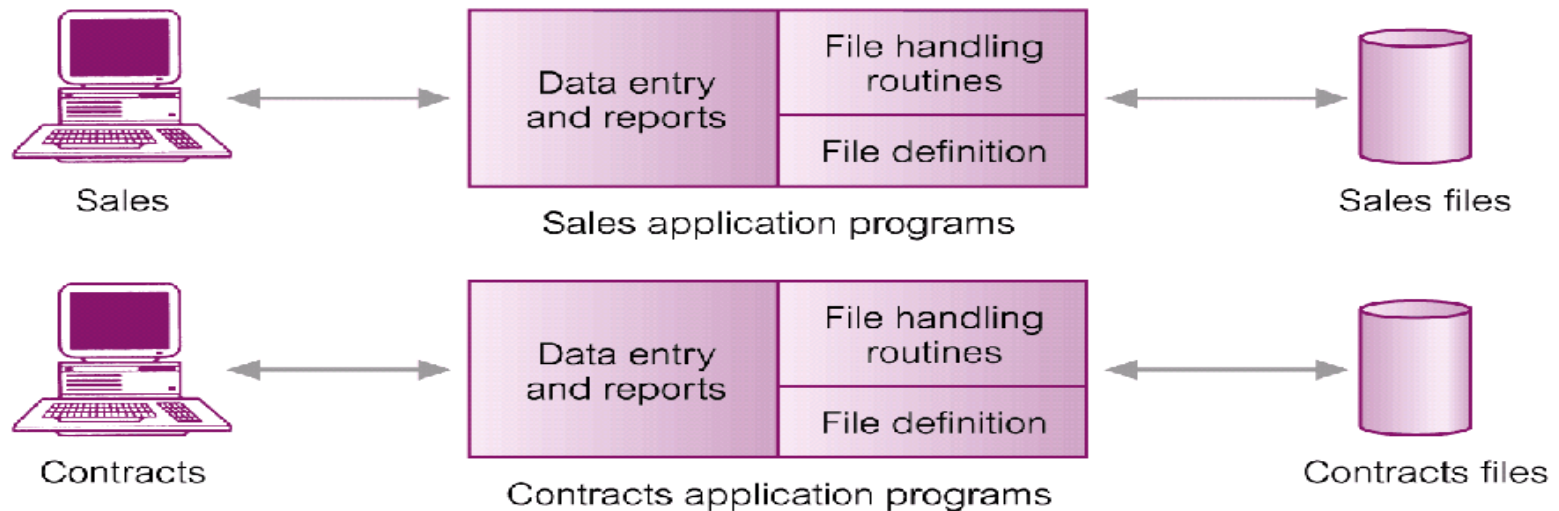
Sales Files

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
  - Data Inconsistency
  - *More information: internet & [1]*

File-based approach

Shared file approach
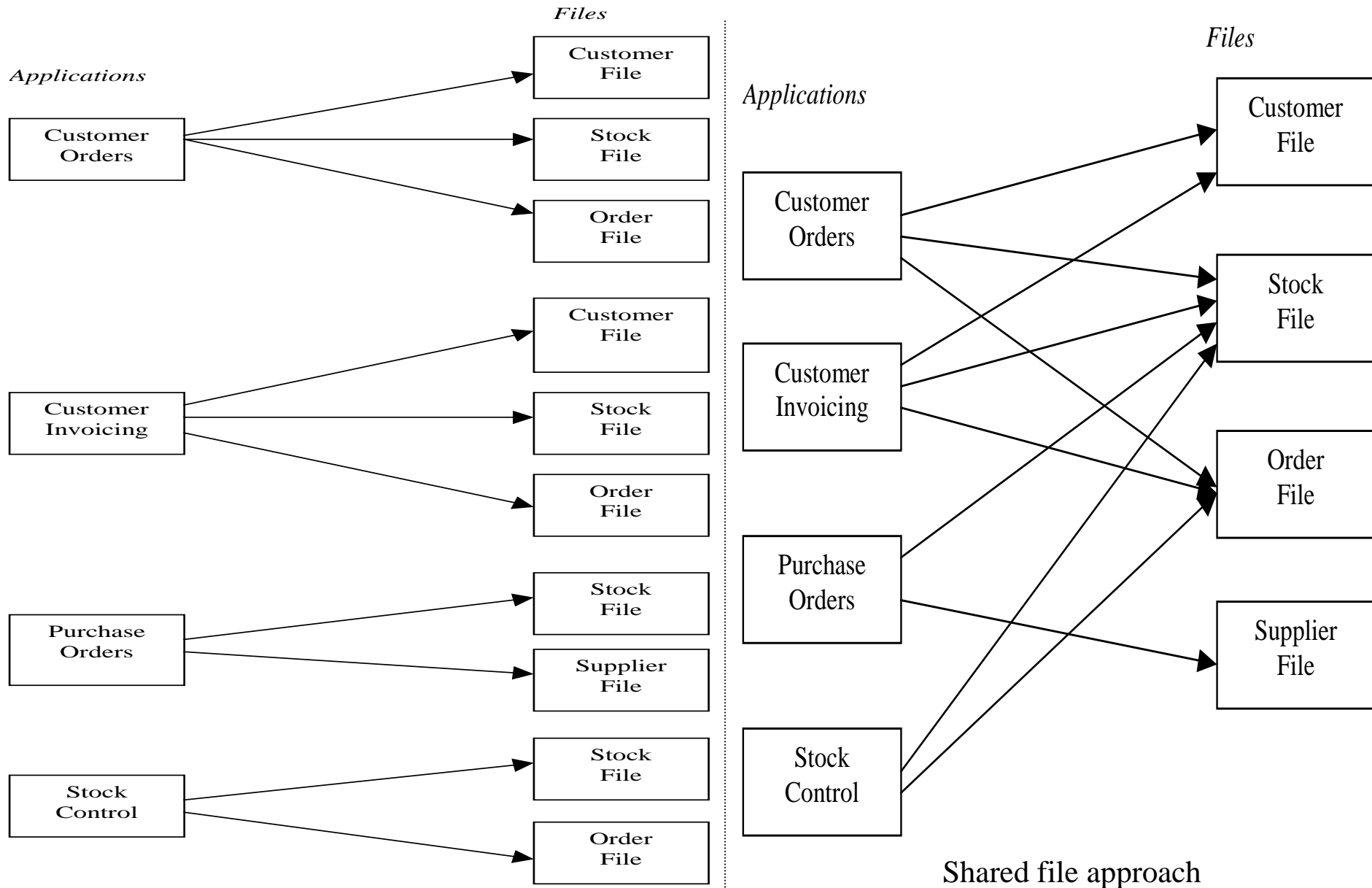
File-based Approach

- Shared File Approach
  - Data (files) is shared between different applications
  - Data redundancy problem is alleviated
  - Data inconsistency problem across different versions of the same file is solved
  - Other problems:
    - →Rigid data structure: If applications have to share files, the file structure that suits one application might not suit another
    - →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
    - →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

- 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:
  - The Database and Database Management System (DBMS).

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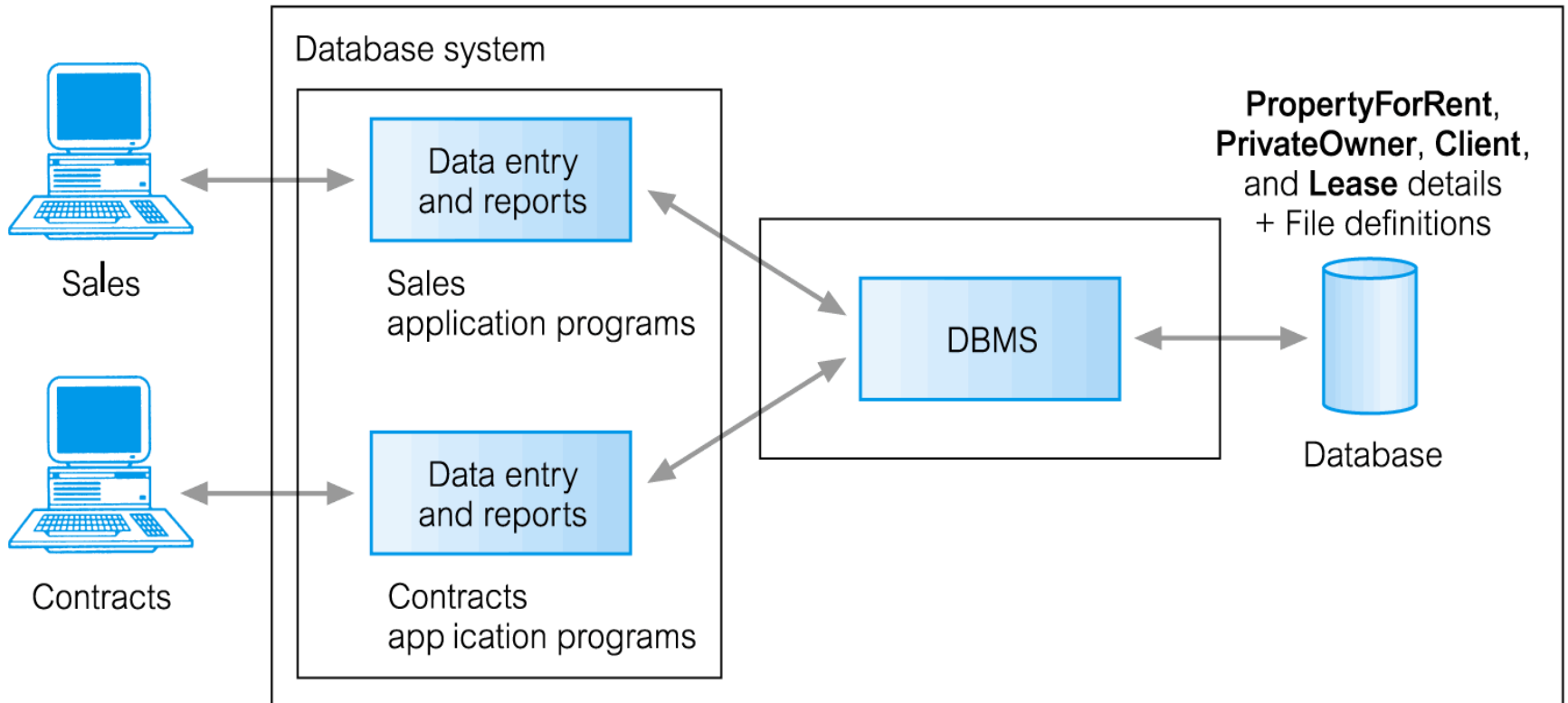
PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)

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

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

Lease (leaseNo, propertyNo, clientNo, paymentMethod, deposit, paid, rentStart, rentF nish)

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

- **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*)

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

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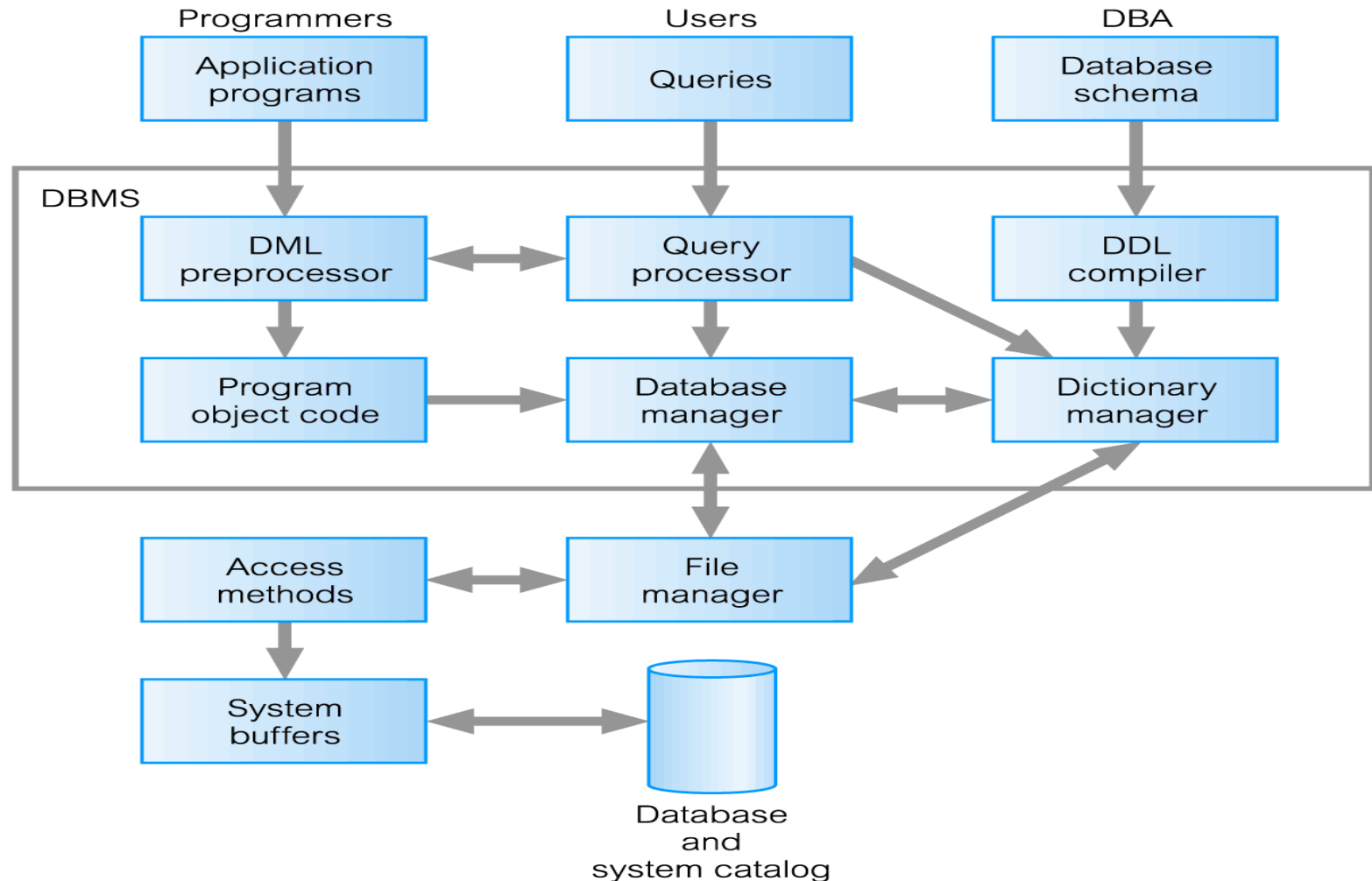
Database Approach

- Data Definition Language (DDL)
  - 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)

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

Database Approach

- Roles in the Database Environment
  - Database Administrator (DBA): responsible for authorizing access to DB, coordinating & monitoring its use, and for acquiring software and hardware resources as needed
  - Database Designers: responsible for identifying the data to be stored in DB, choosing appropriate structures to represent and store this data
  - Application Programmers
  - End Users
  - More details: see [1]-chapter 1

Database Approach

- DBMS components: see [1] for the details

Database Approach

- ▪ Characteristics of the Database Approach
  - • Self-describing nature of a database system
  - • Insulation between programs and data, and data abstraction
    - →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]-1.6

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]-section 1.7

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

- 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
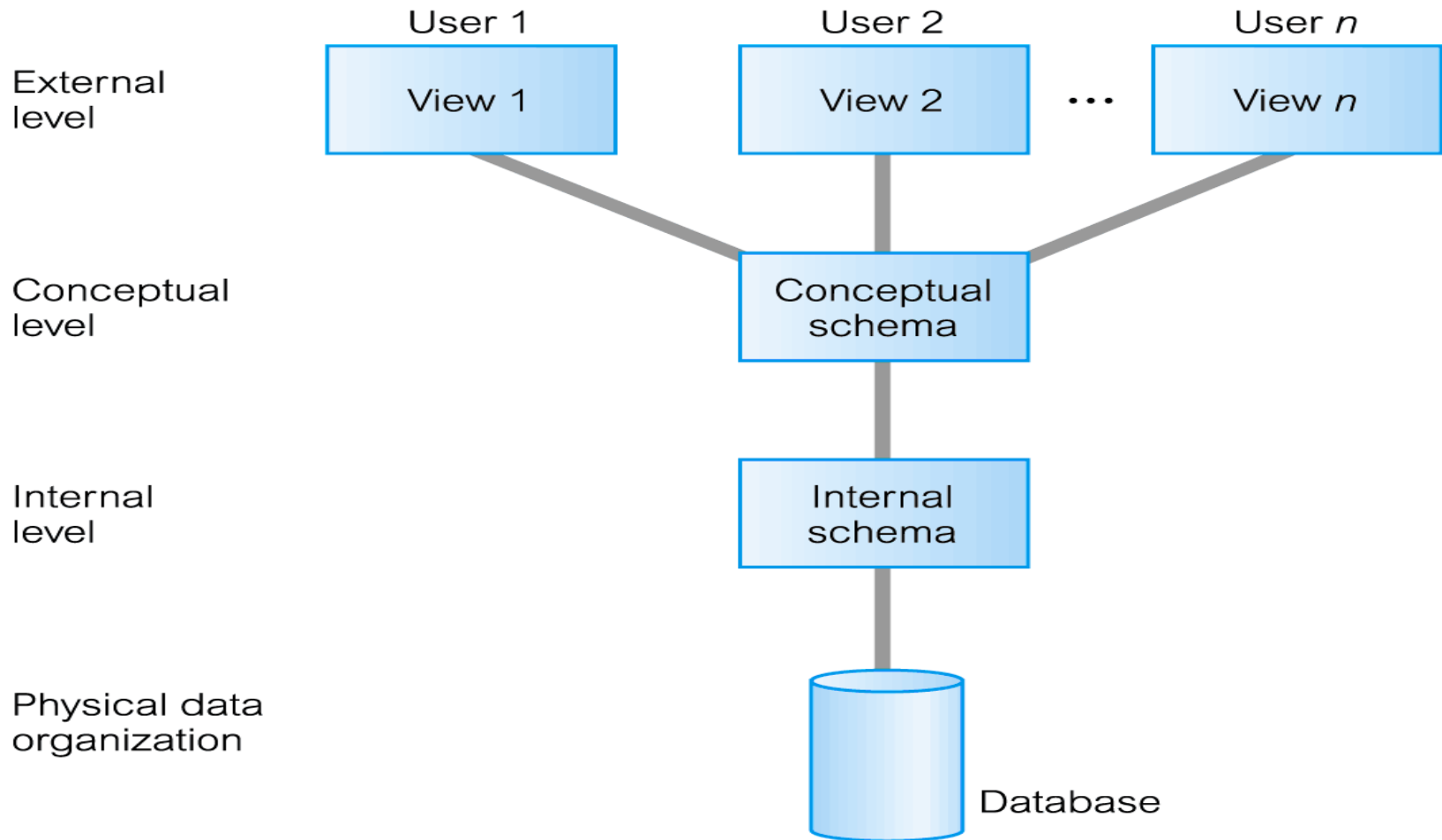
Three-Level Architecture and Data Independence

- **External Level**
  - Users' view of the database
  - Describes that 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 */
```
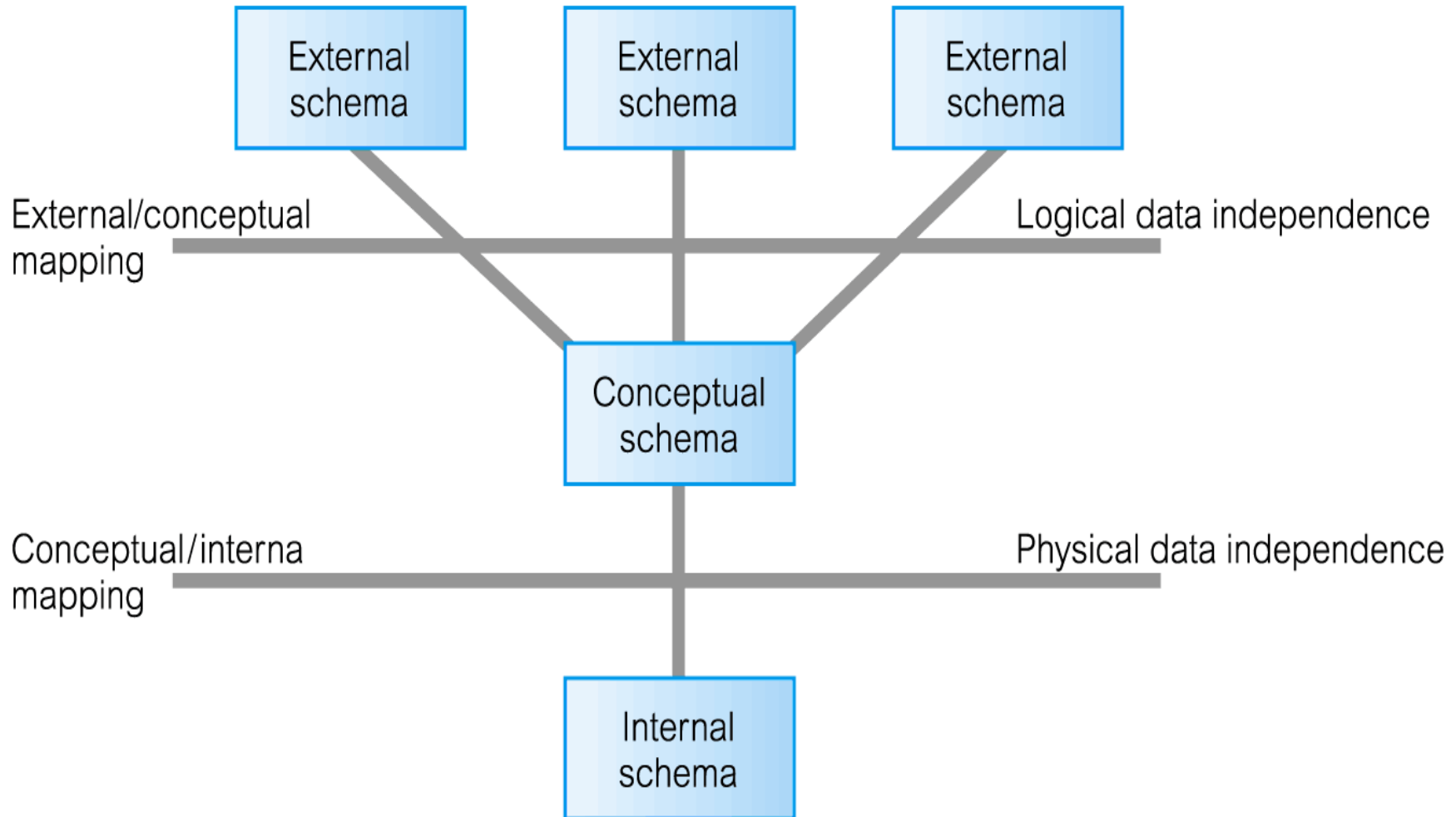
Three-Level Architecture and Data Independence

- 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

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

- 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

Database Languages

- Procedural DML allows user to tell system exactly how to manipulate data (e.g., Network and hierarchical DMLs)
- 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)
- 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

- **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 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, streaming data, …
  - More complex data structures than relational representation
  - New data types except for the basic numeric and character string types
  - New operations and query languages for new data types
  - New storage and retrieval methods
  - New security mechanisms
  - …

Summary

- File-based Approach
- Database Approach
- Three-Schema Architecture and Data Independence
- Database Languages
- Data Models Database Schema and Database State
- Data Management Systems Framework (where are we?)
- Reading Suggestion & Homework: do not forget !!
- Next lecture: ER Model

Q&A