SECD2523 DATABASE

TOPIC 1 | INTRODUCTION TO DATABASE

Content adapted from Connolly, T., Begg, C., 2015. Database Systems: A Practical Approach to Design, Implementation, and Management, Global Edition. Pearson Education.

LECTURE LEARNING OUTCOME

By the end of this lecture, students should be able to:

01 **Common uses** of database systems (DB).

02 **Characteristics**, **problems** and **differences** between file-based approach and DB approach.

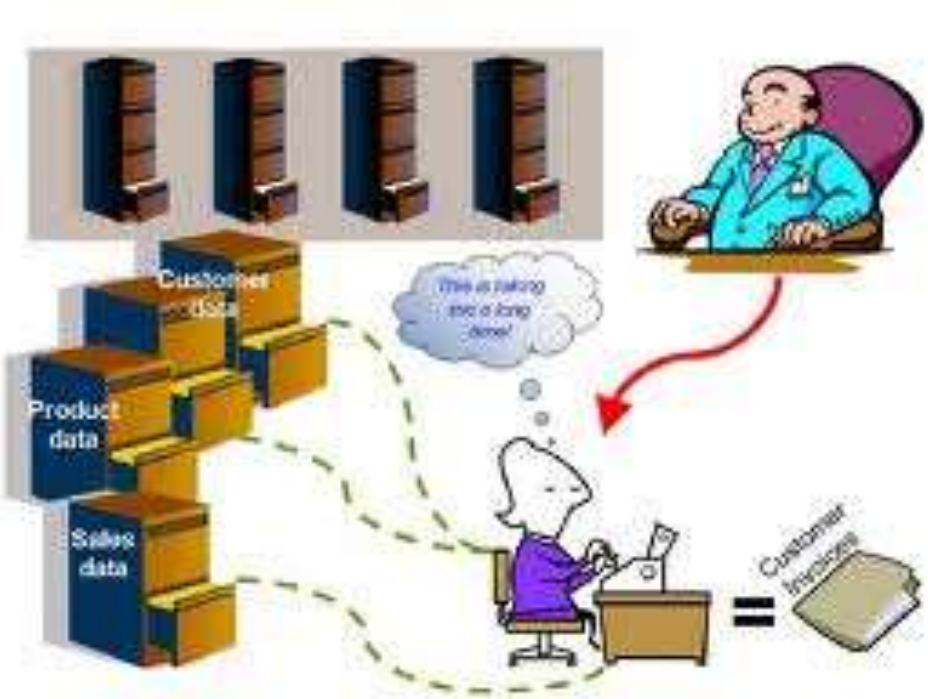
03 Meaning of **terms**: DB, Database Management System (DBMS), DB application system, DB system.

04 **Personnel** involved in the DBMS environment.

05 **Advantages** and **disadvantages** of DBMS.

06 Three-level **ANSI-SPARC** architecture.

Before computer



After computer

Introduction

- Database is now such an integral part of our day-to-day life that often we are not aware we are using one.

Source: Coronel, 2019

Introduction

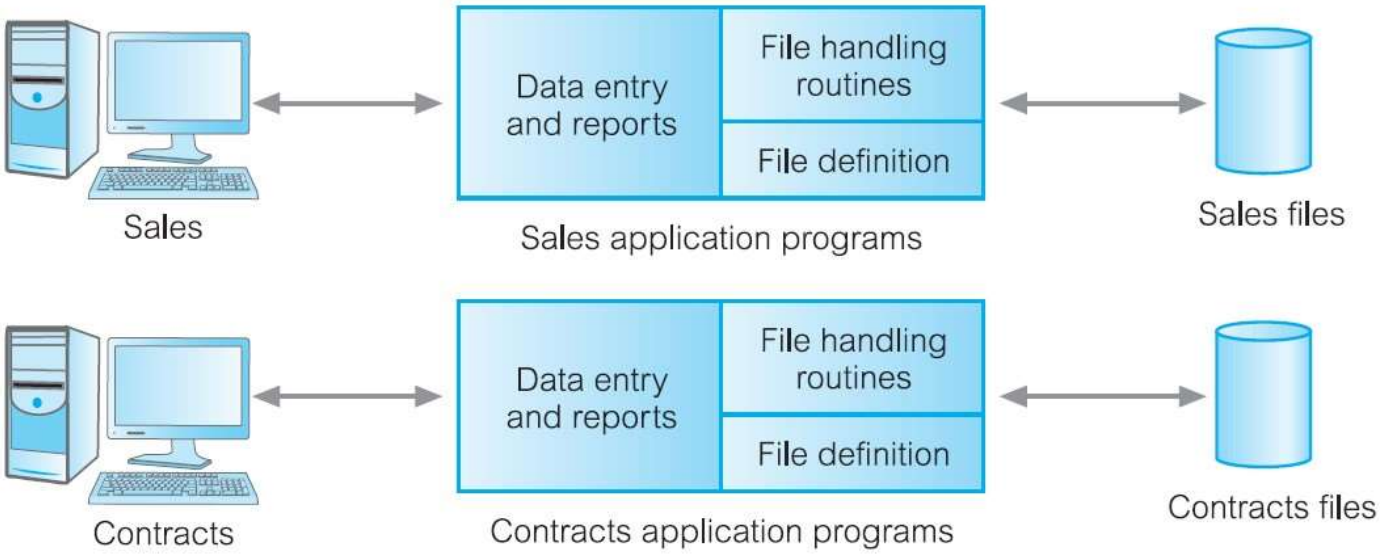
- Some of the terminologies in this topic:

Terms	Description
Database	A collection of related data.
Database Management System (DBMS)	The software that manages & controls access to the database.
Database application	A program that interacts with the database at some point in its execution.
Database system	A collection of application programs that interact with the database along with the DBMS and database itself.

Traditional File-based System

- An early attempt to computerize the manual filing system.
- Definition:
  - Collection of application programs that perform services for the end users (e.g.: reports). Each program defines and manages its own data.
- Works well while the number of items to be stored is small. However, it breaks down when we need to cross-reference or process the information in the files.

File-based Processing



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 Processing

- Figure shows each department accessing their own files through application programs written specially for them.
- Each set of departmental application programs handles data entry, file maintenance, and the generation of a fixed set of specific reports.
- The physical structure & storage of the data files and records are defined in the application code

Limitations of File-based Approach

- **Separation and isolation of data**
  - Each program maintains its own set of data.
  - Users of one program may be unaware of potentially useful data held by other programs.
- **Duplication of data**
  - Same data is held by different programs.
  - Wasted space and potentially different values and/or different formats for the same item.

Limitations of File-based Approach

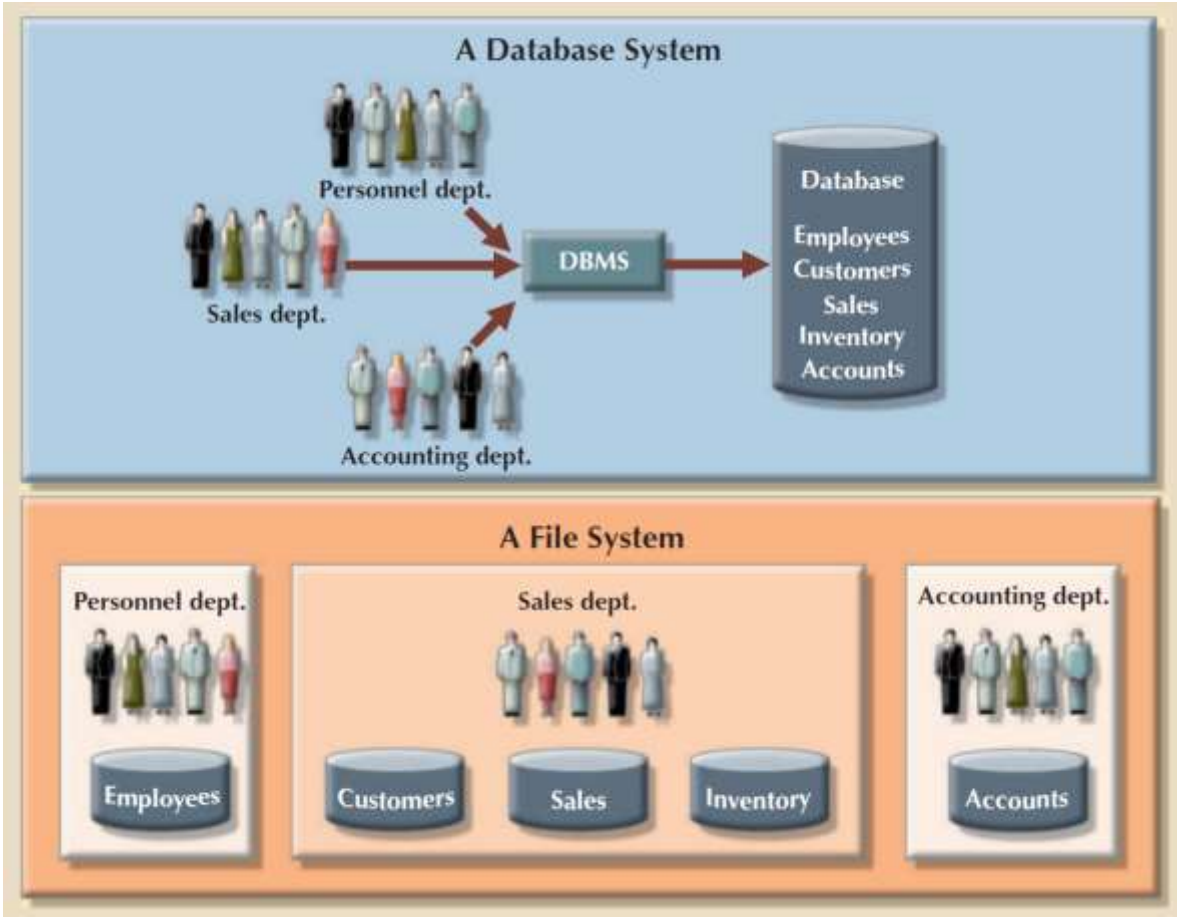
- **Data dependence**
  - File structure (for data file) is defined in the program code.
  - Changes to an existing structure are difficult to make
- **Incompatible file formats**
  - Programs are written in different languages, and so cannot easily access each other's files.
- **Fixed Queries/Proliferation of application programs**
  - Programs are written to satisfy specific functions.
  - Any new requirement needs a new program.

Database Approach

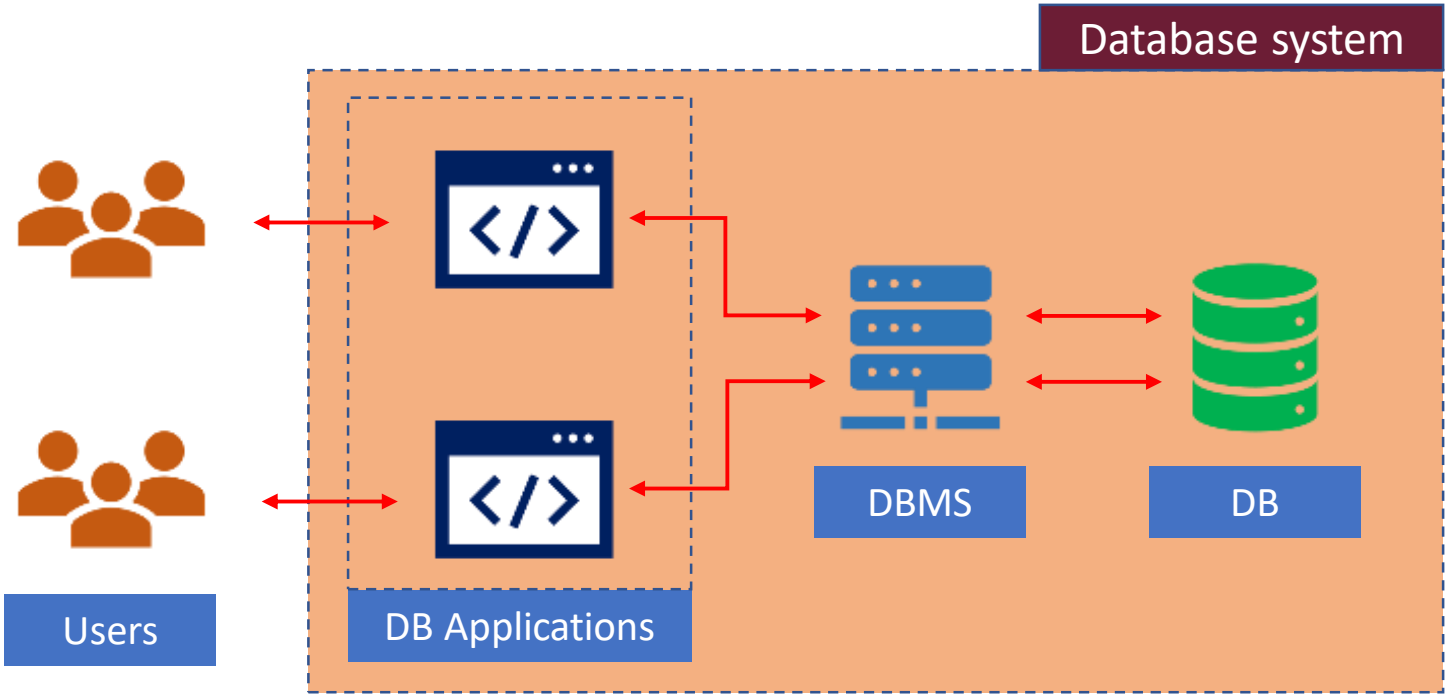
- The limitations of the file-based approach can be attributed to 2 factors:
  - Definition of data was embedded in application programs, rather than being stored separately and independently.
  - No control over access and manipulation of data beyond that imposed by application programs.
- Result, a new approach was required:-
  - the database and Database Management System (DBMS).

Contrasting File-based Approach and Database Approach

Source: Coronel, 2019



Overview of DB System



The Database

- Definition:
  - Shared collection of logically related data (and a description of this data), designed to meet the information needs of an organization.
- A database is also defined as a self-describing collection of integrated records
  - The description of the data is known as the system catalogue (metadata) to enable program–data independence.
  - The definition of data is separated from the application program.
  - The users of an object only see the external definition and unaware of how the object is defined and how it functioned
- Logically related data comprises entities, attributes, and relationships of an organization's information.

Database Management System (DBMS)

- DBMS is the software that interacts with the user's application programs and the database
- Definition: A software system that enables users to define, create, and maintain the database and that provides controlled access to this database.
  - Define using Data Definition Language (DDL)
  - Create insert, update, delete & retrieve data using Data Manipulation Language (DML)
  - Controlled accessed security, integrity, concurrency control, recovery & user-accessible catalog
- Example of DBMS – Microsoft Access, Microsoft SQL Server, Oracle, Sybase, MongoDB, CouchDB and etc.

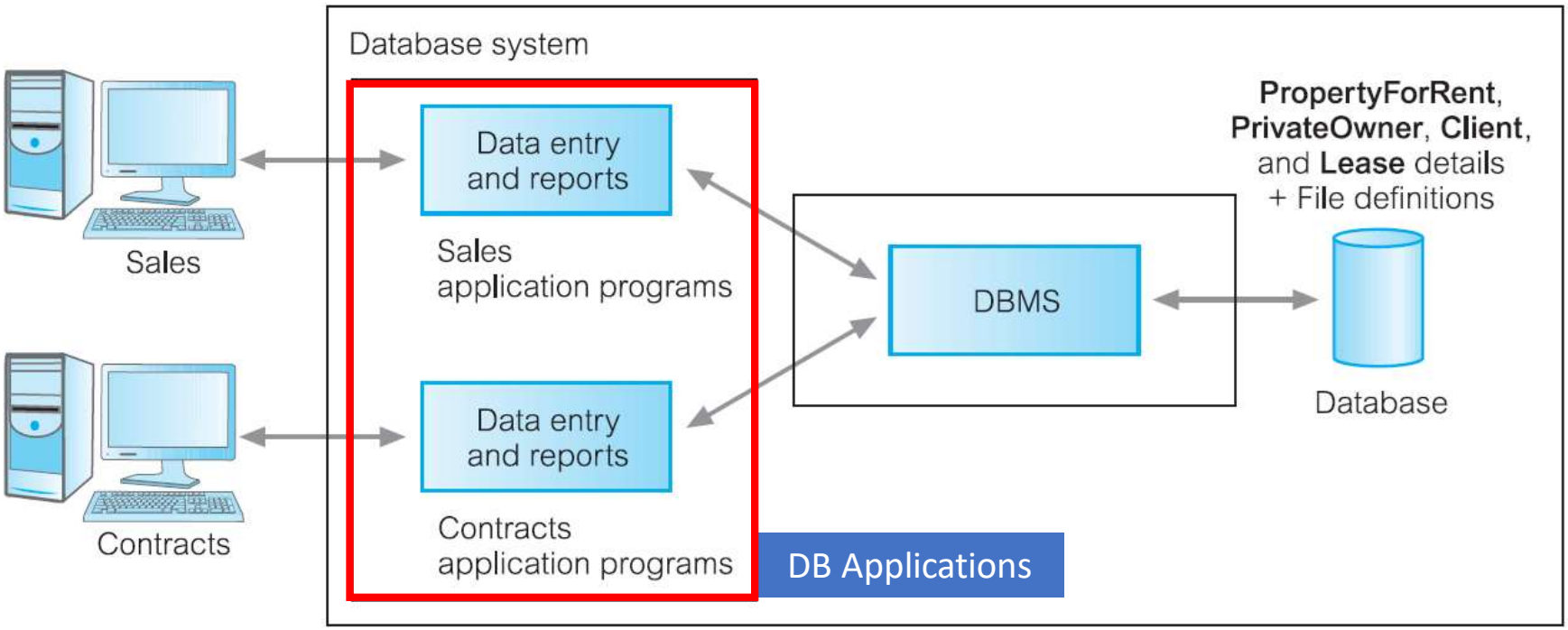
Database Management System (DBMS)

- **Data definition language (DDL).**
  - Permits specification of data types, structures and any data constraints.
  - All specifications are stored in the database.
- **Data manipulation language (DML).**
  - General enquiry facility (query language) of the data.
  - Using Structured Query Language (SQL) to produce required information.-

Database Application Programs

- Definition of (Database) Application Program
  - A computer program that interacts with the database by issuing an appropriate request (typically an SQL statement) to the DBMS
- Users interact with the database through several application programs (used to create & maintain the database & to generate information) ⍰ written in some programming language

Database Application Programs

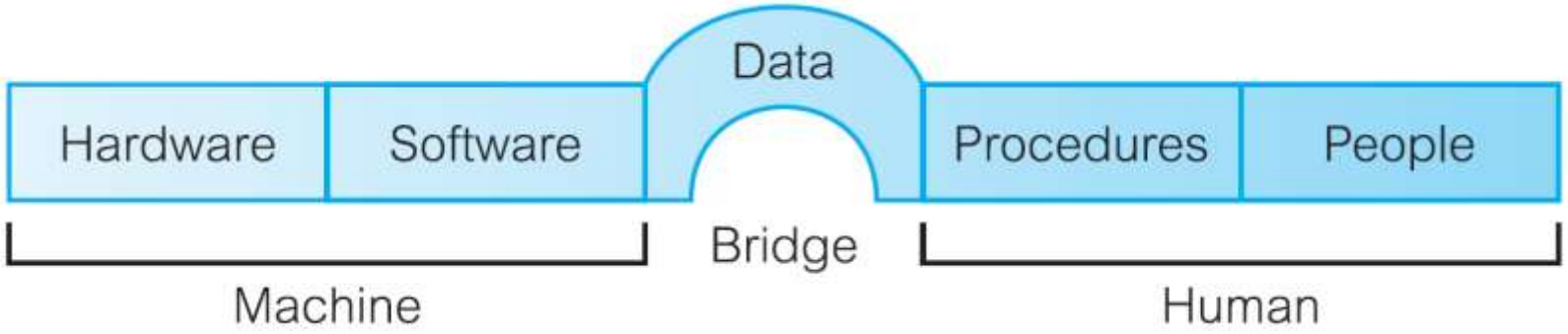


PropertyForRent (propertyNo, street, city, postcode, type, rooms, rent, ownerNo)
PrivateOwner (ownerNo, fName, lName, address, telNo)
Client (clientNo, fName, lName, address, telNo, prefType, maxRent)
Lease (leaseNo, propertyNo, clientNo, paymentMethod, deposit, paid, rentStart, rentFinish)

Database Application Programs

- Figure shows each department using their application programs to access the database through the DBMS.
  - Each set of departmental application programs handles data entry, file maintenance, and the generation of a fixed set of specific reports.
  - The physical structure & storage of the data are now managed by the DBMS
- A view mechanism.
  - Provides users with only the data they want or need to use.

Components of DBMS Environment



Roles in the Database Environment

- **Data Administrator (DA)**
  - responsible for the **management of the data resource**
    - Database planning
    - Development and maintenance of standards, policies, procedures
    - Conceptual/logical database design
- **Database Administrator (DBA)**
  - responsible for the **physical realization** of the database
    - Physical database design and implementation
    - Security & integrity control
    - Maintenance of operational control
    - Ensuring satisfactory performance of applications for users

Roles in the Database Environment

- **Database Designers (Logical and Physical)**
  - **Logical** : is concerned with identifying the data, the relationships between the data, & the constraint on the data that is to be stored in the database
  - **Physical** : decides how the logical database design is to be physically realized
- **Application Programmers**
  - **build the application programs** that provide the required functionality for the end-users
- **End Users**
  - **naive** and **sophisticated**

File-based Approach

Limitation of File Based Approach :

- Separation and isolation of data
- Duplication of data
- Data dependence
- Incompatible file formats
- Fixed Queries/Proliferation of application programs

Database Approach

Advantages of Database Approach :

- Control of data redundancy
- Data consistency
- More information from the same amount of data
- Sharing of data
- Improved data integrity & security
- Improved data accessibility and responsiveness
- Increased productivity
- Improved maintenance through data independence
- Increased concurrency
- Improved backup and recovery services

Disadvantage of Database Approach :

- Complexity
- Size
- Cost of DBMSs
- Additional hardware costs
- Cost of conversion
- Performance
- Greater impact of a failure

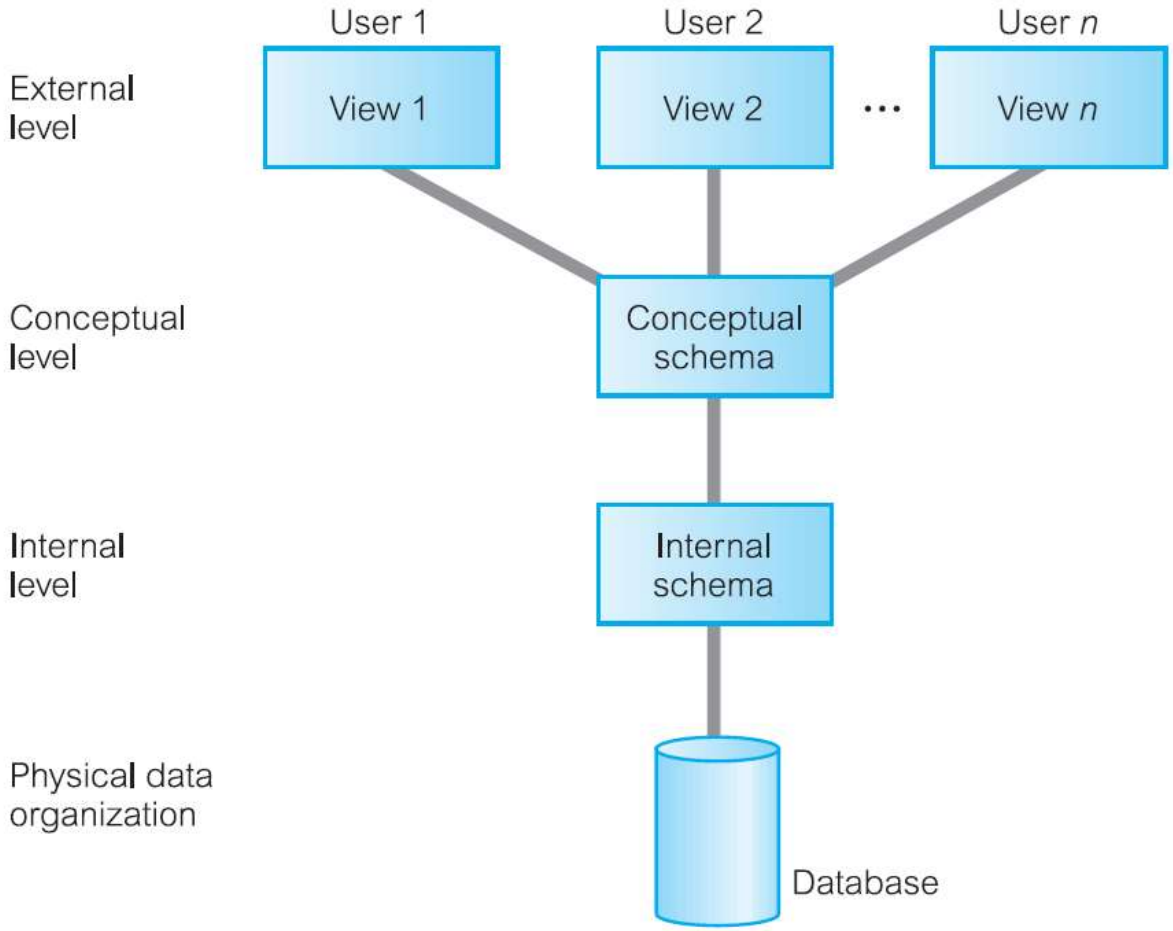
Three-Level Architecture

- All DBMS are built based on the **3-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.

Three-Level Architecture

- 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.

ANSI-SPARC Three-Level Architecture



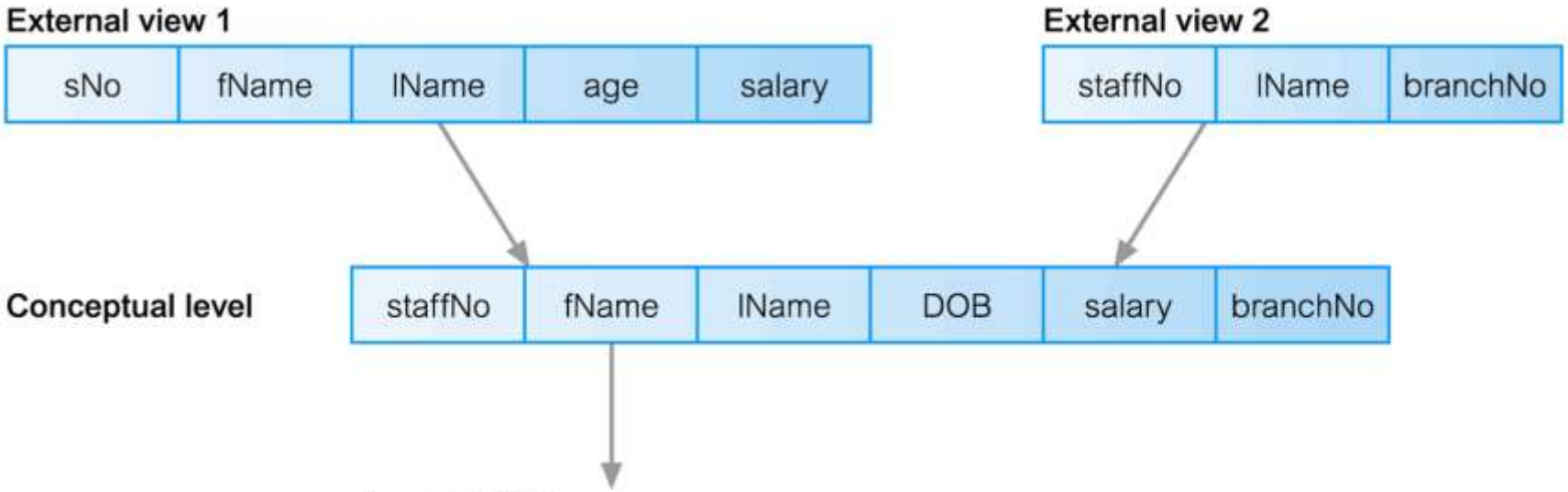
ANSI-SPARC Three-Level Architecture

- **External Level**
  - Users' views 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.

ANSI-SPARC Three-Level Architecture

- **Internal Level**
  - Physical representation of the database on the computer.
  - Describes how the data is stored in the database.

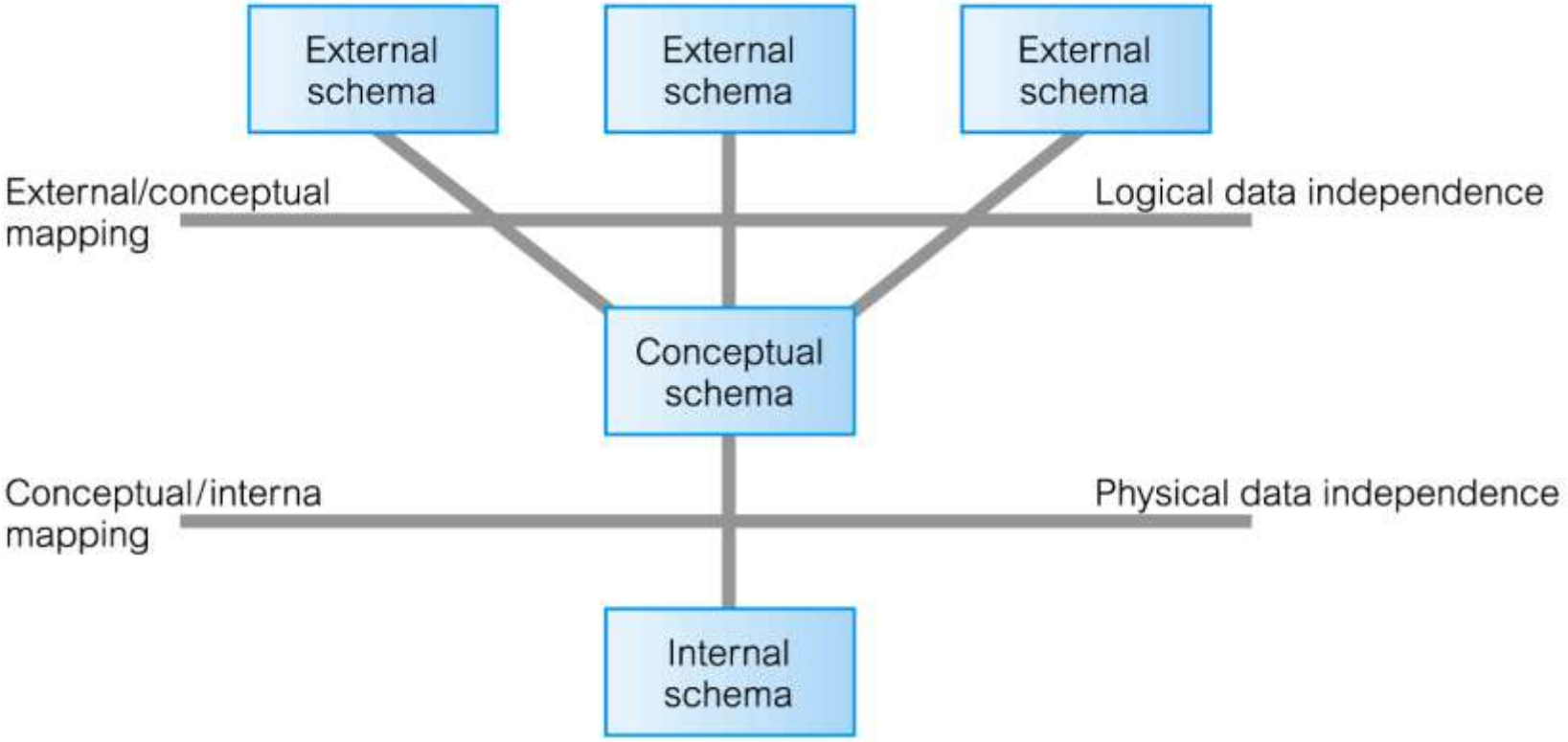
Differences between the three levels



Internal level

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

Data Independence and the ANSI-SPARC Three-Level Architecture



Data Independence

- Referring to mapping between external, conceptual & internal levels
- **Logical Data Independence**
  - The 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.

Data Independence

- **Physical Data Independence**
  - The 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 change to conceptual or external schemas.

Summary

- Importance of databases
  - Database approach VS file-based approach
- Important terms in database fields
  - Database, database applications, database systems, DBMS
- Database architectures – ANSI-SPARC levels and associations with data independence

Innovating Solutions
Menginovasi Penyelesaian