3ISY402 – DATABASE SYSTEMS

Lecture 9 – Database Environment and Architecture

Material from essential text:

T CONNOLLY & C BEGG. Database Systems – A Practical Approach to Design, Implementation and Management, 4th Edition. Addison-Wesley, 2005.

Leena Gulabivala

Lecture - Objectives

- Characteristics of file-based systems and problems.
- Meanings of the terms *databases* and *DBMS*.
- Major components of the DBMS environment.
- Personnel involved in the DBMS environment.
- Advantages and disadvantages of DBMSs.
- Purpose of three-level database architecture.
- Contents of *external*, *conceptual*, and *internal levels*.
- Meaning of *logical* and *physical data independence*.
- Typical functions and services a DBMS should provide.

File-Based Processing

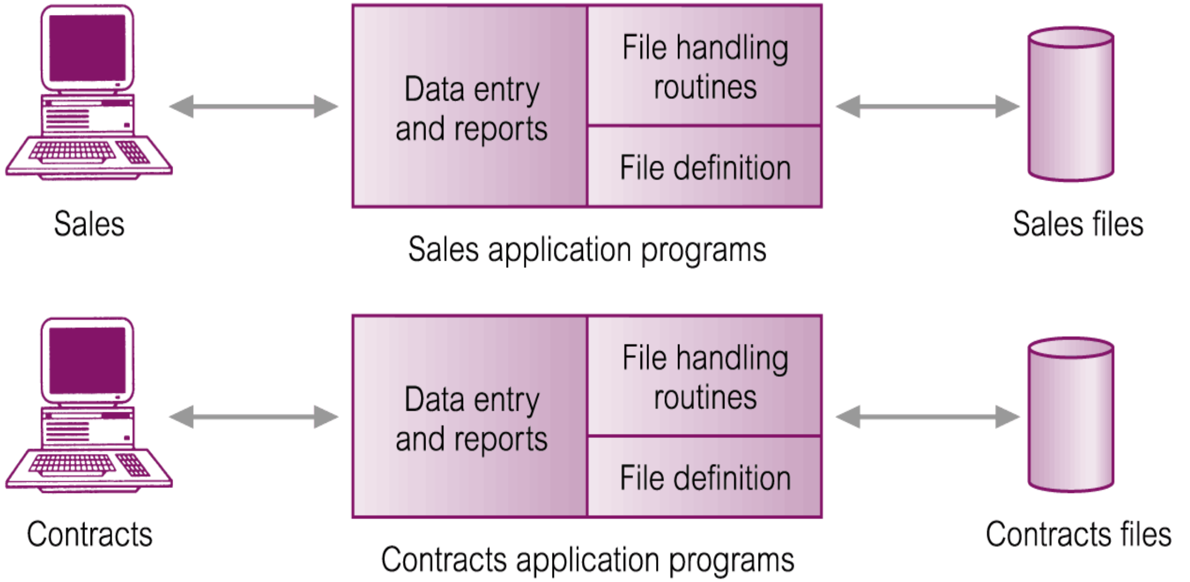


Figure 1.5 File-based processing.

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)

Limitations of File-Based Approach

- *Separation and isolation of data*
  - Each program maintains (defines and manages) 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.
- *Data dependence*
  - File structure is defined in the program code.

Database

- *A database is a shared collection of logically related data (and a description of this data), designed to meet the information needs of an organization.*
- 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.

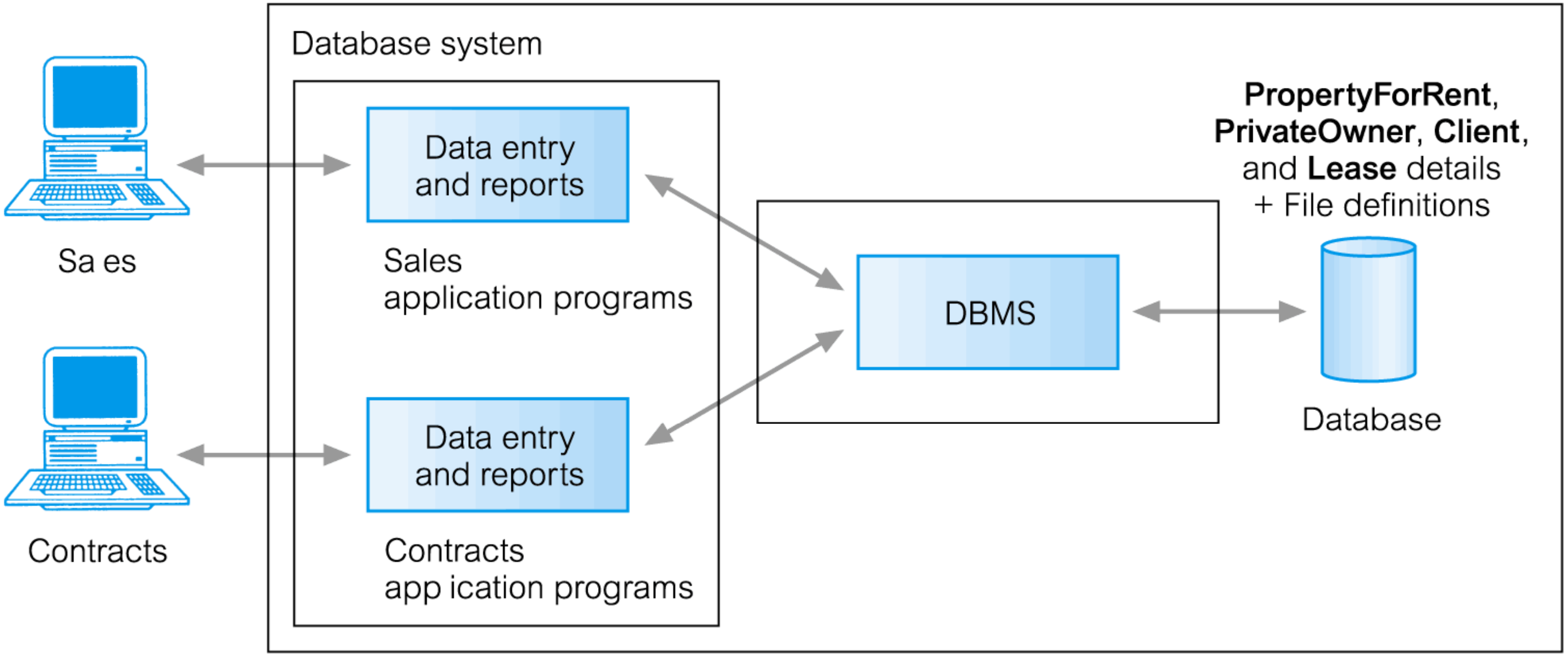
Database Management System (DBMS)

- A *DBMS* is a software system that enables users to define, create, maintain, and control access to the database.

- A *database application program*: a computer program that interacts with database by issuing an appropriate request (SQL statement) to the DBMS.

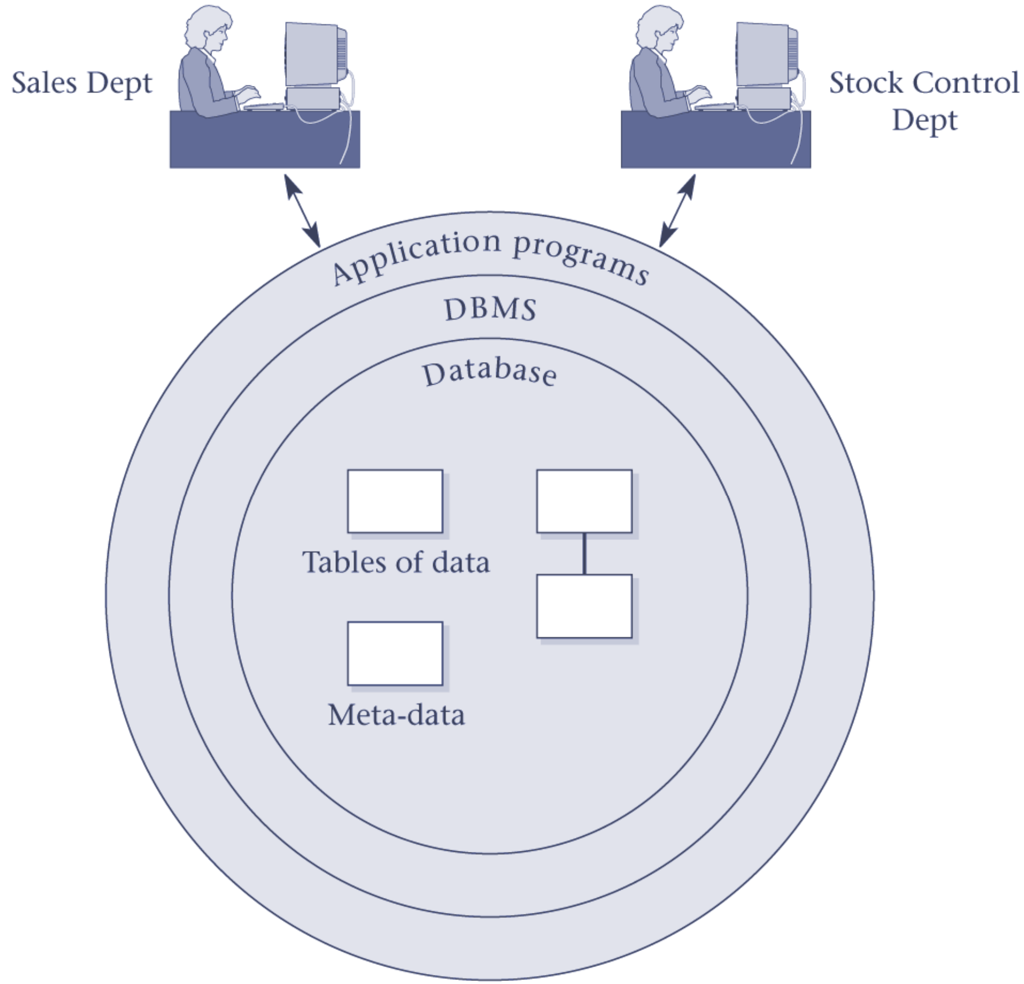
- *Database System* is used to define a collection of application programs that interact with the database, along with the DBMS and the database itself.

Database Management System (DBMS)



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)

DBMS


Sales Dept
Stock Control Dept
Application programs
DBMS
Database
Tables of data
Meta-data

Components of DBMS Environment

- *Hardware*
  - Can range from a PC to a network of computers.
- *Software*
  - DBMS, operating system, network software (if necessary) and also the application programs.
- *Data*
  - Used by the organization and a description of this data called the schema (or meta-data).
- *Procedures*
  - Instructions and rules that should be applied to the design and use of the database and DBMS.
- *People*

Roles in the Database Environment

- Data Administrator (DA)
- Database Administrator (DBA)
- Database Designers (Logical and Physical)
- Application Programmers
- End Users

Data Administration and Database Administration

- Data Administrator (DA) and Database Administrator (DBA) are responsible for managing and controlling activities associated with corporate data and corporate database, respectively.
- *DA is more concerned with early stages of lifecycle and DBA is more concerned with later stages.*

Data Administration (DA)

- Management and control of corporate data, including:
  - database planning
  - development and maintenance of standards, policies, and procedures
  - conceptual and logical database design

Database Administration (DBA)

- Management and control of physical realization of a database system, including:
  - physical database design and implementation
  - setting security and integrity controls
  - monitoring system performance
  - reorganizing the database

Comparison of Data and Database Administration

Data administration	Database administration
Involved in strategic IS planning	Evaluates new DBMSs
Determines long-term goals	Executes plans to achieve goals
Determines standards, policies, and procedures	Enforces standards, policies, and procedures
Determines data requirements	Implements data requirements
Develops logical database design	Develops physical database design
Develops and maintains corporate data model	Implements physical database design
Coordinates database development	Monitors and controls database use
Managerial orientation	Technical orientation
DBMS independent	DBMS dependent

Advantages of DBMSs

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

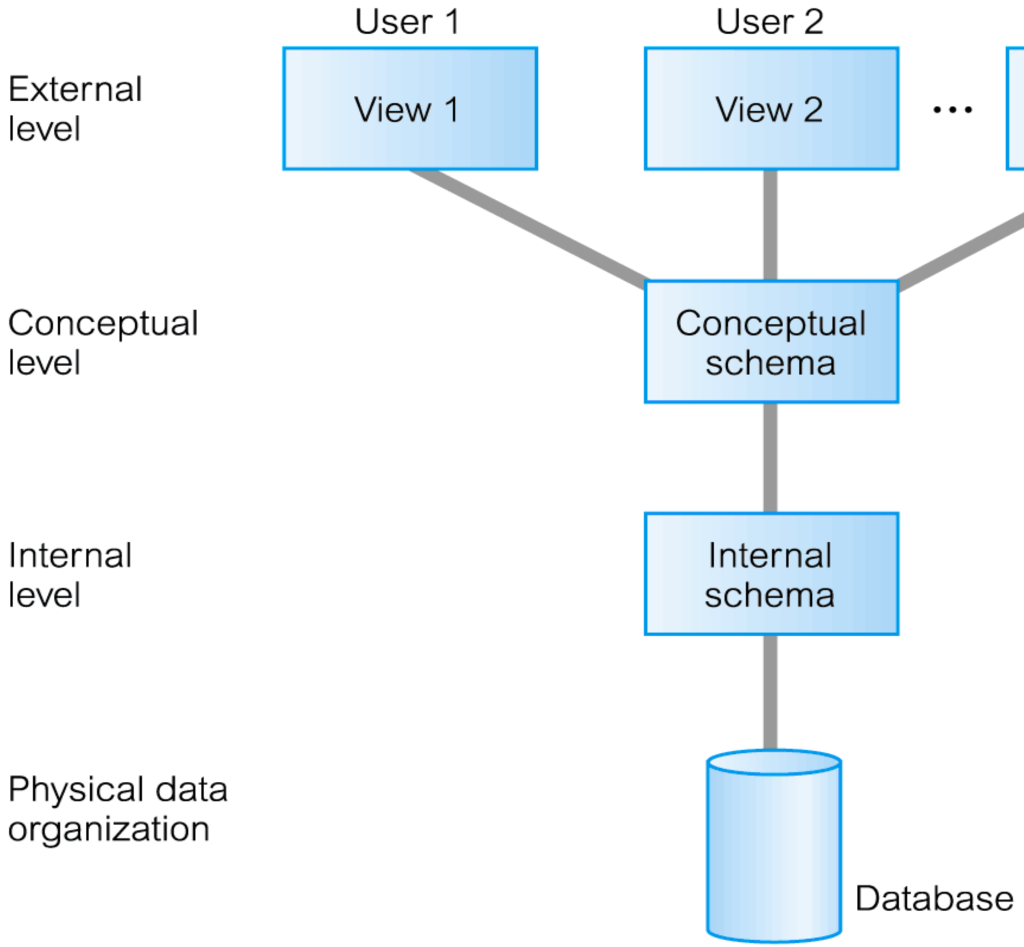
Disadvantages of DBMSs

- Complexity.
- Cost of DBMS.
- Additional hardware costs.
- Cost of conversion.
- Higher impact of a failure.
- Performance.

Objectives of Three-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.

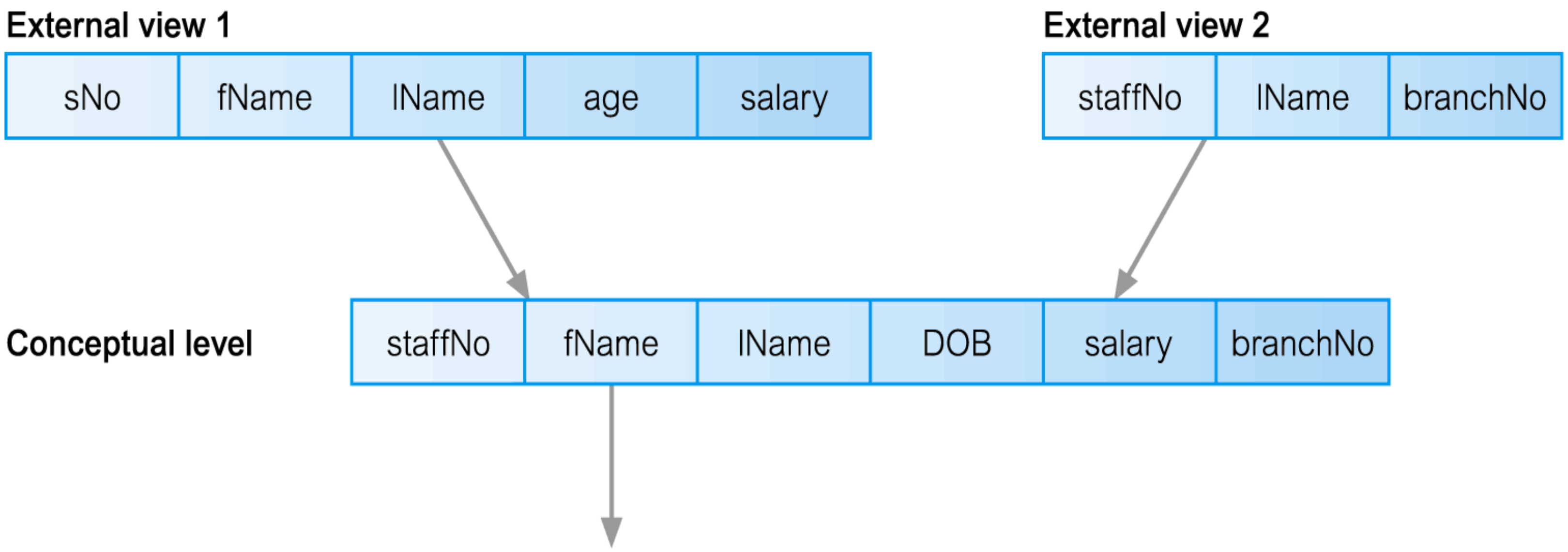
ANSI-SPARC Three-Level Architecture



ANSI-SPARC Three-Level Architecture

- 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.

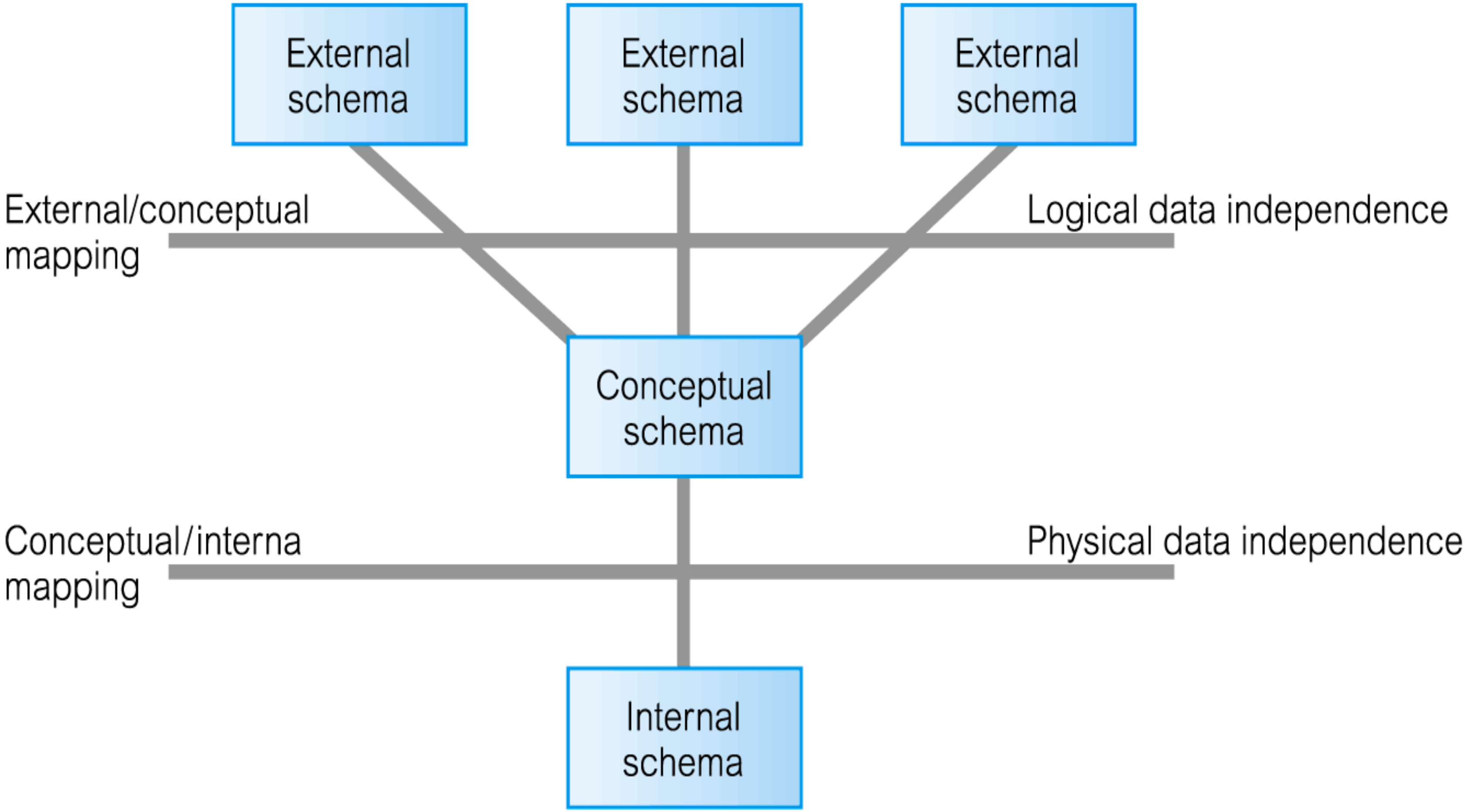
Differences between Three Levels of ANSI-SPARC Architecture



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 */
```

Data Independence and the ANSI-SPARC Three-Level Architecture



Data Independence

- *Logical Data Independence:*
  - Refers to immunity of external schemas to changes in conceptual schema.
  - Conceptual schema changes (e.g. addition of entities or attributes).
  - Should not require changes to external schema or rewrites of application programs.
  - *Only the mappings between the conceptual schema and the external schemas need to be changed.*

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 change to conceptual or external schemas.
  - *Only the mappings between the internal schema and the higher level schemas need to be changed.*

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/security constraints.
- *Data Manipulation Language (DML)* – provides basic data manipulation operations on data held in the database.
- *Procedural DML* – allows user to tell system exactly *how* to manipulate data (PL/SQL)
- *Non-Procedural DML* - allows user to state *what* data is needed rather than how it is to be retrieved (SQL)

Functions of a DBMS

- Data storage, retrieval, and update.
- A user-accessible catalog.
- Transaction support (COMMIT and ROLLBACK).
- Concurrency control services (multi-users).
- Recovery services.
- Authorization services (GRANT and REVOKE).
- Integrity services.
- Support for data communication.
- Services to promote data independence.
- Utility services.

System Catalog

- Repository of information (metadata) describing the data in the database.
- One of the fundamental components of DBMS.
- Typically stores:
  - names, types, and sizes of data items,
  - constraints on the data,
  - names of authorized users,
  - data items accessible by a user and the type of access,
  - usage statistics.

References and Further Reading:

- T CONNOLLY & C BEGG. *Database Systems – A Practical Approach to Design, Implementation and Management,* 4th Edition. Addison-Wesley, 2005.
  - Chapter 1 – Introduction to Databases.
  - Chapter 2 – Database Environment
- T CONNOLLY & C BEGG. *Database Solutions – A step-by-step guide to building databases,* 2nd Edition. Addison-Wesley, 2004.
  - Chapter on Introduction to Databases.