Database Systems

Introduction to Databases

Objectives

- **Some common uses of database systems.**
- **Characteristics of file-based systems.**
- **Database and Database Management System (DBMS).**
- **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.**
- **Components of a DBMS**

Examples of Database Applications

- **Purchases from the supermarket**
- **Purchases using your credit card**
- **Booking a holiday at the travel agents**
- **Using the local library**
- **Taking out insurance**
- **Renting a video**
- **Using the Internet**
- **Studying at university**

File-Based Systems

- **Collection of application programs that perform services for the end users (e.g. reports).**
- **Each program defines and manages its own data.**

File-Based Processing

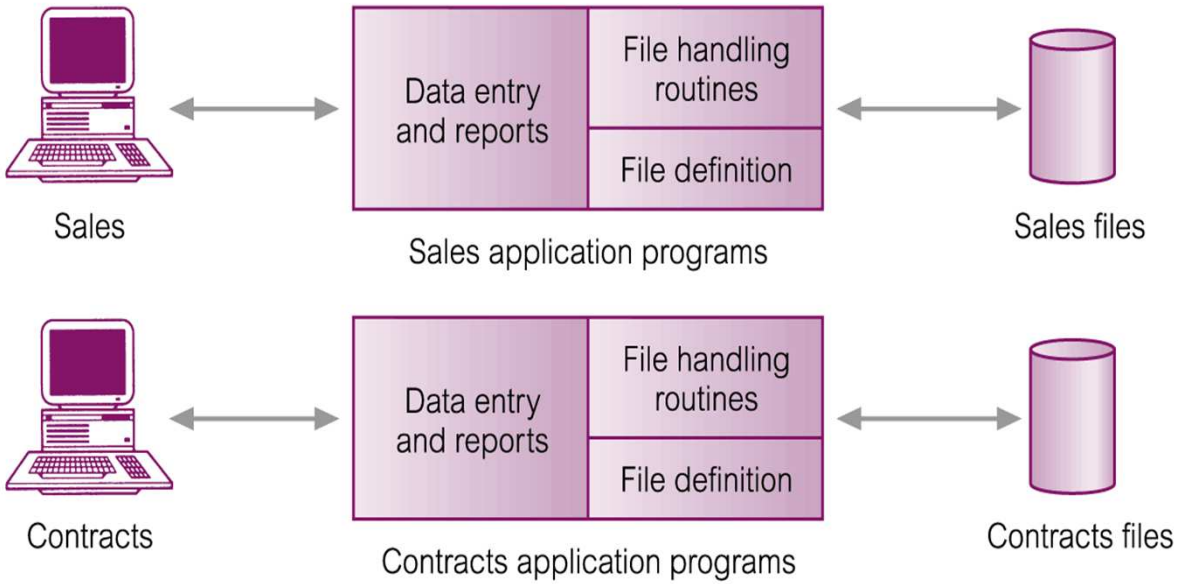


Figure 1.5
File-based processing.

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 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 is defined in the program code.**
- **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 particular functions.**
  - **Any new requirement needs a new program.**

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

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

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

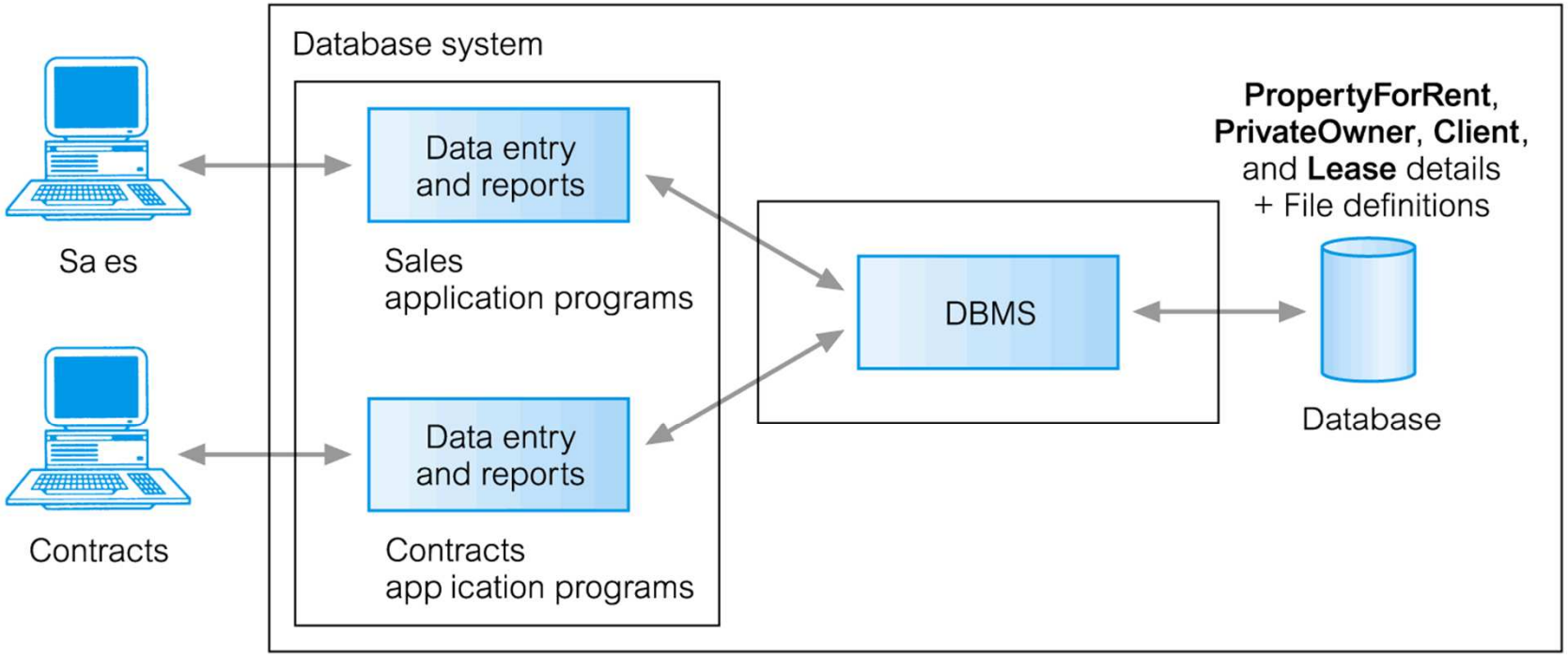
- **Logically related data comprises entities, attributes, and relationships of an organization's information.**

Database Management System (DBMS)

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

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

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)

Database Approach

- **Controlled access to database may include:**
  - **a security system**
  - **an integrity system**
  - **a concurrency control system**
  - **a recovery control system**
  - **a user-accessible catalog.**

Advantages of DBMSs

- **Control of data redundancy**
- **Data consistency**
- **More information from the same amount of data**
- **Sharing of data**
- **Improved data integrity**
- **Improved security**
- **Enforcement of standards**
- **Economy of scale**

Advantages of DBMS

- **Balance conflicting requirements**
- **Improved data accessibility and responsiveness**
- **Increased productivity**
- **Improved maintenance through data independence**
- **Increased concurrency**
- **Improved backup and recovery services**

Disadvantages of DBMS

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

Database Environment
Three Level Architecture

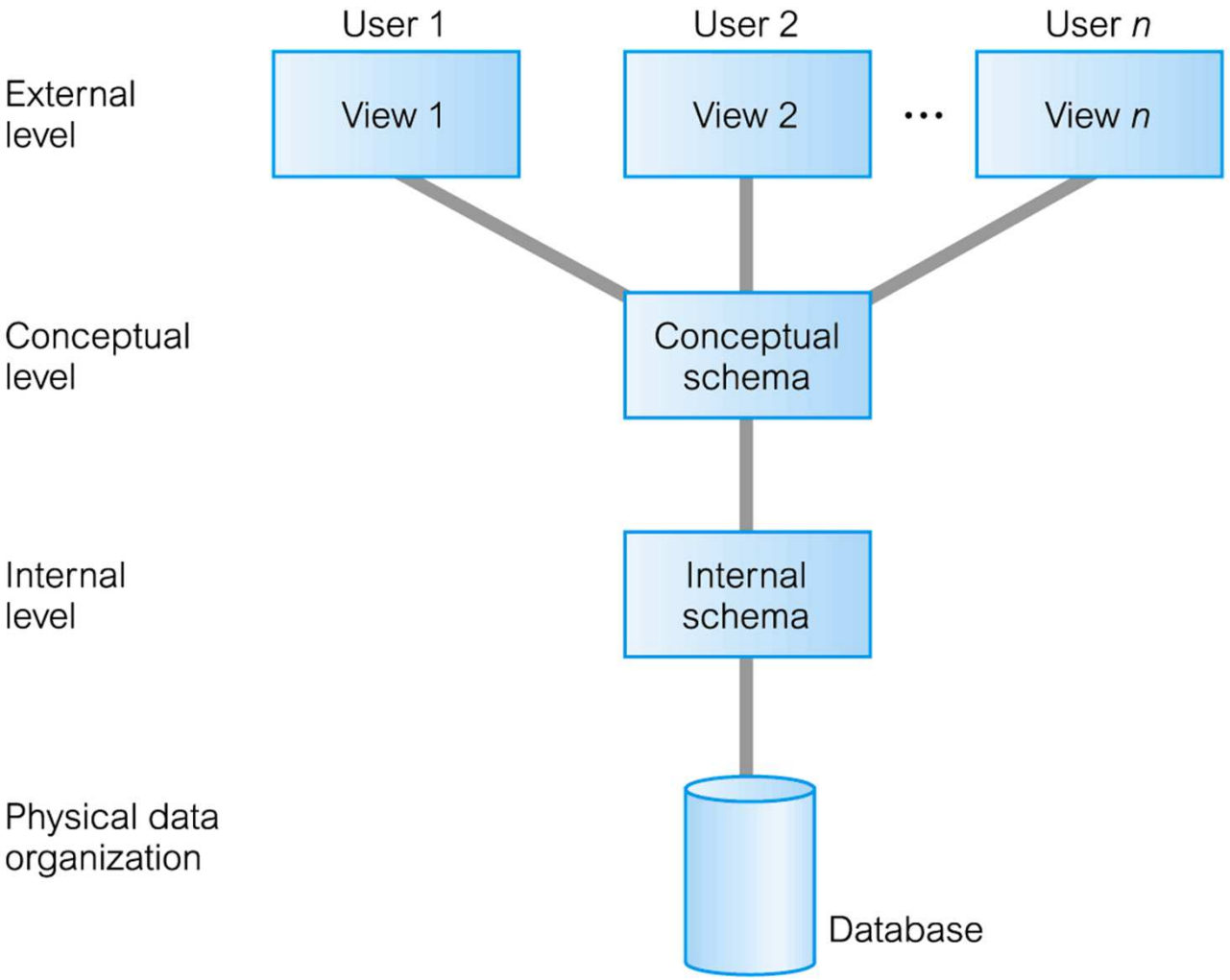
Objectives of Three-Level Architecture

- **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 the physical database storage details.**

Objectives of 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.**

Three-Level Architecture



Three-Level Architecture

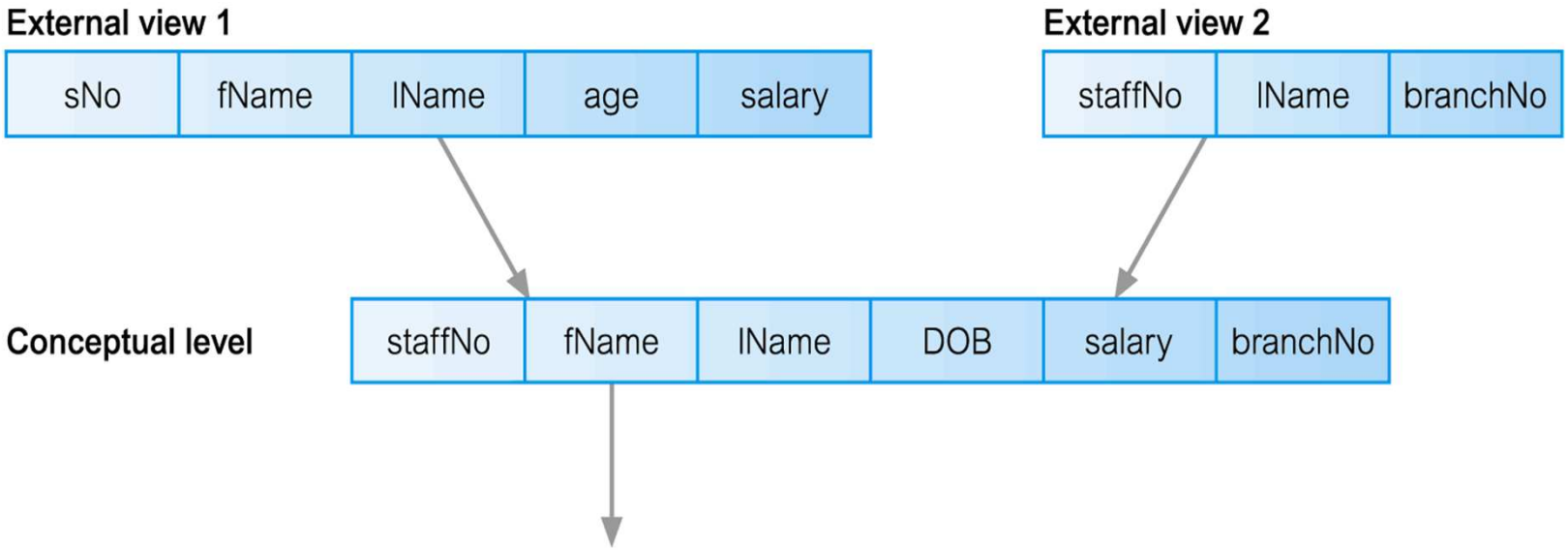
- **External Level**
  - **Users' view of the database.**
  - **Describes that part of database that is relevant to a particular user.**

- **Conceptual Level**
  - **Community view of the database.**
  - **Describes what data is stored in database and relationships among the data.**

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



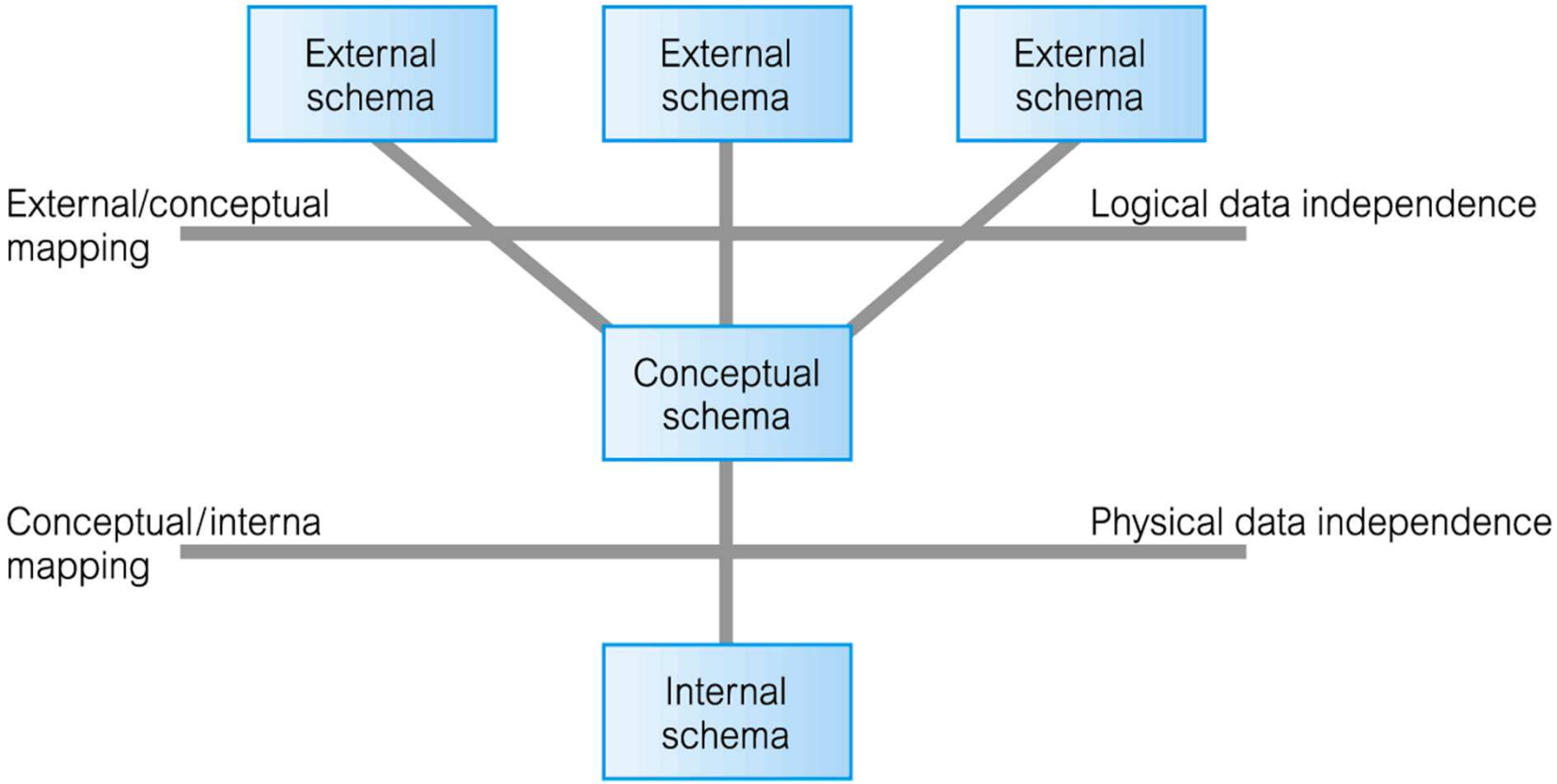
Data Independence

- **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.**

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.**

Data Independence and the Three-Level Architecture



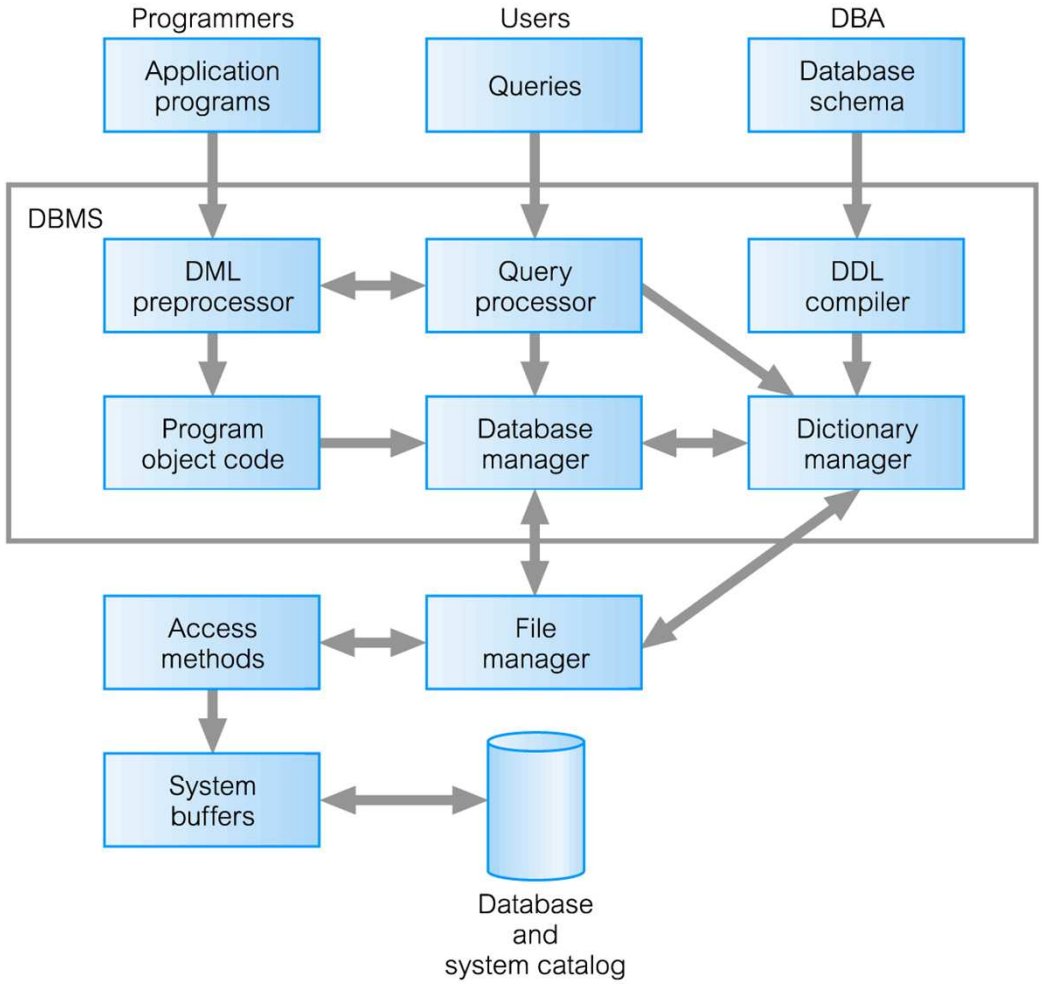
Functions of a DBMS (1)

- **Data Storage, Retrieval, and Update.**
  - **Ability to store, retrieve and update data**
- **A User-Accessible Catalog.**
  - **Description of data items and access specification**
- **Transaction Support.**
  - **ACID properties maintained**
- **Concurrency Control Services.**
  - **Updates leave database in a consistent state**
- **Recovery Services.**
  - **Ability to restore database to a valid state**

Functions of a DBMS (2)

- **Authorization Services.**
  - **Only authorised users access database**
- **Support for Data Communication.**
  - **Integration of DBMS with communication software**
- **Integrity Services.**
  - **Conformance to standards and rules**
- **Services to Promote Data Independence.**
  - **Independence of programs from database structure**
- **Utility Services**
  - **Provision of utilities e.g. monitoring, statistics etc.**

Components of a DBMS (1)



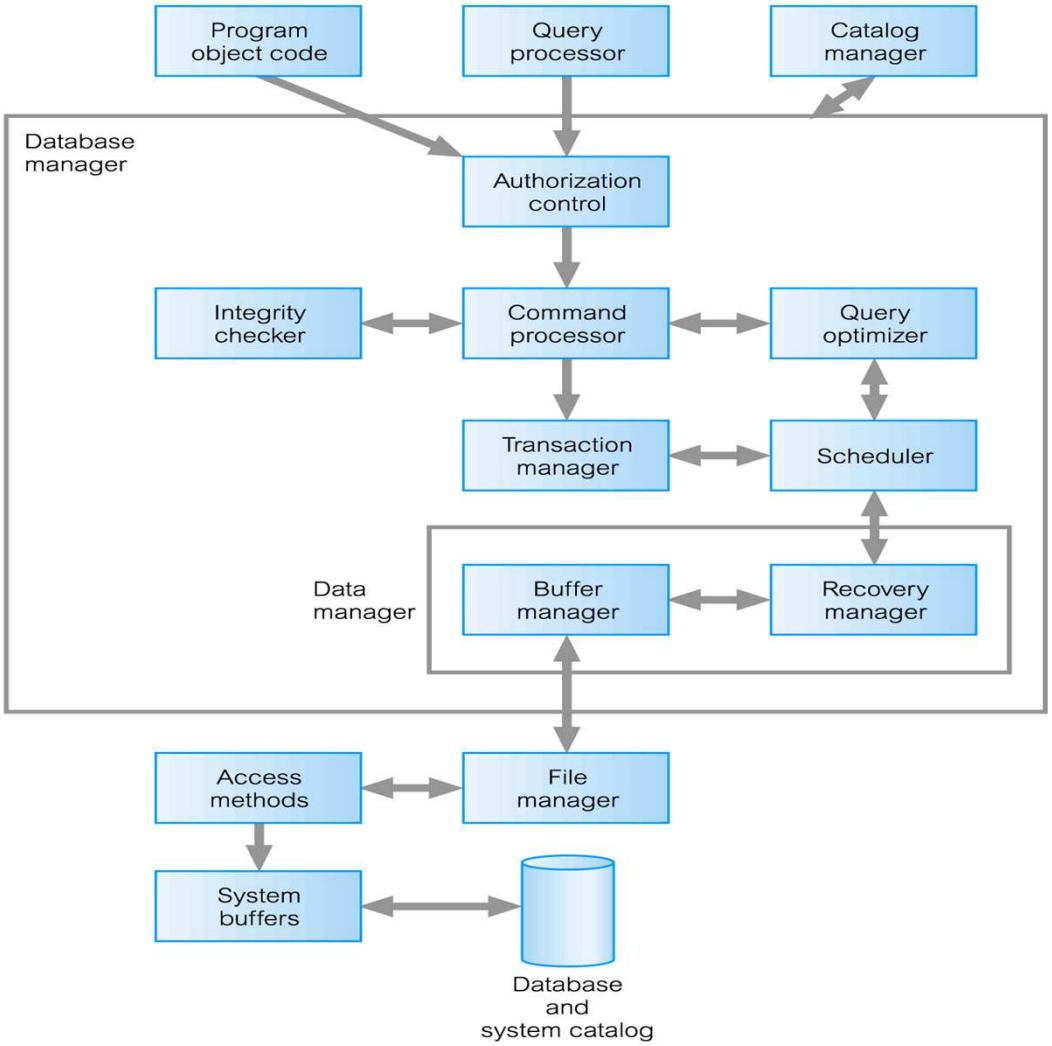
Components of a DBMS (2)

- **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.**

Components of a DBMS (3)

- **Query processor**
  - **Transforms queries into a low-level instructions**
- **File manager**
  - **Manages underlying storage files, allocation of space on disk and maintains list of structures defined in internal schema**
- **DML processor**
  - **Converts embedded DML statements into standard function calls in the host language**
- **DDL compiler**
  - **Converts DDL statements into a set of tables containing metadata, and then stored in the catalog**

Components of Database Manager (1)



Components of Database Manager (2)

- **Authorisation control**
  - **Checks if user has necessary authorisation to carry out required operation**
- **Command processor**
  - **Initiates execution of operation**
- **Integrity checker**
  - **Checks required operation satisfies integrity constraints**
- **Query optimiser**
  - **Determines optimal strategy for query optimisation**

Components of Database Manager (3)

- **Transaction manager**
  - **Processes transaction operations**
- **Scheduler**
  - **Controls concurrent transaction operations**
- **Recovery manager**
  - **Ensures that database is brought to a consistent state after a failure**
- **Buffer manager**
  - **Manages transfer of data between memory and disk**

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.**

Database Languages

- **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.**
- **Non-Procedural DML**
  - **allows user to state what data is needed rather than how it is to be retrieved.**
- **Fourth Generation Languages (4GLs)**