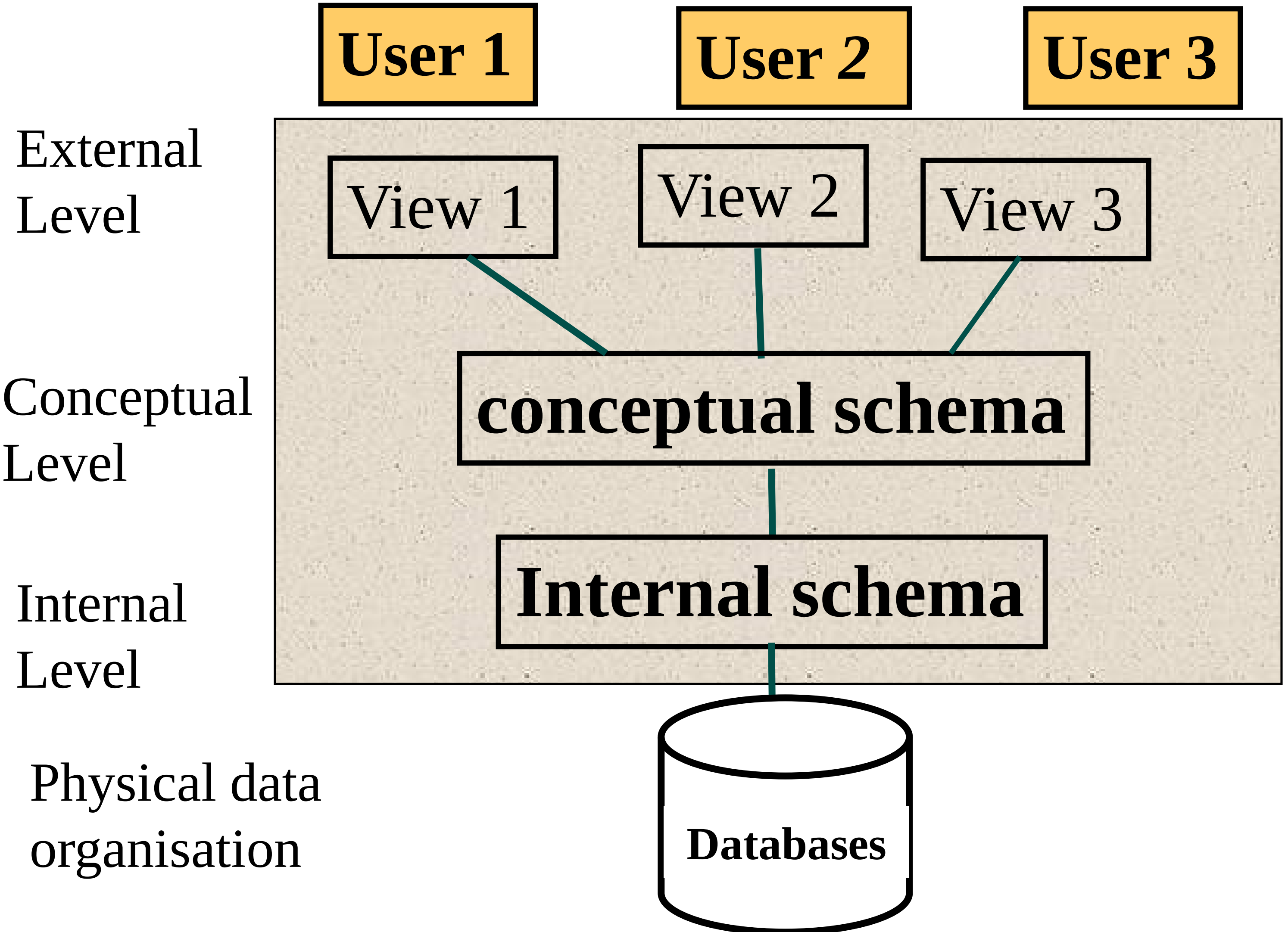
Database Environment

Dr. Enosha Hettiarachchi

Structure

- Three-level database architecture.
- Contents of external, conceptual, and internal levels.
- Purpose of external/conceptual and conceptual/internal mappings.
- Meaning of logical and physical data independence.
- Distinction between DDL and DML.
- Purpose/importance of conceptual modeling.
- Typical functions and services a DBMS should provide.
- Software components of a DB manager.

Three-level ANSI/SPARC architecture



3 Level ANSI/SPARC Architecture

- **External level**
  - User’s view of the database.
- **Conceptual level**
  - Describes what data is stored in the database and the relationships among the data.
- **Internal level**
  - Describes how the data is stored in the database.

Internal Level

- The physical representation of the database on the computer to achieve optimal runtime performance and storage space utilization.

  - Covers data structures and file organisations used to store data on the storage device.

  - Storage space allocation for data and indexes.

Conceptual Level

- This level contains the logical structure of the entire database. Provides a complete view of the data requirements of the organisation that is independent of any storage considerations.
- The conceptual level represents:
  - All entities, their attributes and their relationships
  - The constraints on the data
  - Security and integrity information

External Level

- Describes the part of the database that is relevant to the user.
- The external view includes only the entities, attributes or relationships in the ‘real world’ that the user is interested in.
- Different views have different representations of the same data.

External Level

- **External Views Allow to**
  - **hide unauthorised data**
    - e.g. *salary*, *dob*
  - **provide user view**
    - e.g. view employee *name*, *designation*, *department* data taken from *employee* and *department* files
  - **derive new attributes**
    - e.g. *age* derived from *dob* or *nid*

External Level

- **External Views Allow to**
  - **change unit of measurement**
    - e.g. show *age* in years or months
  - **define security levels**
    - e.g. update access to *employee* file read-only to *department* file

Three-Level Architecture

- All users should be able to access same data but have a customized view of the data.
- A user’s view is immune to changes made in other views.
- Users should not need to know physical database storage details (e.g. indexing or hashing).

Objectives of Three-Level Architecture

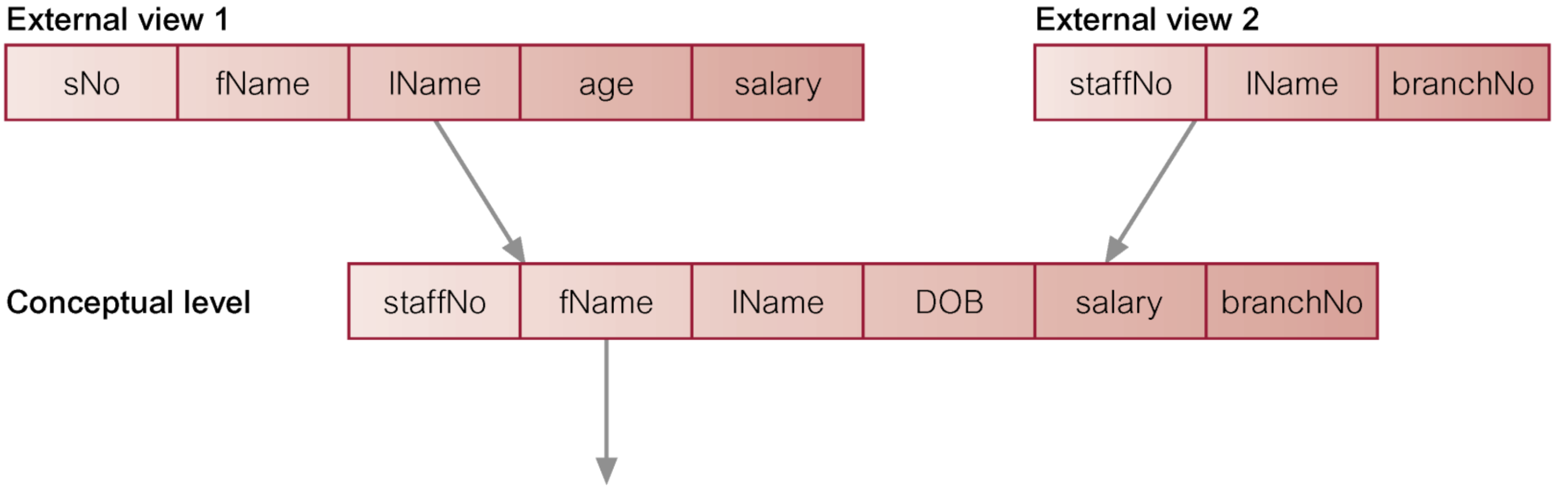
- DBA should be able to change database storage structures without affecting the users' views.
- DBA should be able to change conceptual structure of database without affecting all users.

Physical Level

- Managed by the operating system under the direction of the DBMS.

- Consist of items only the OS knows.

Differences between Three Levels of ANSI-SPARC Architecture



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 */
```

Database Schema

- Overall description of the database is called the database schema.
- There are three different types of schema
  - External (Sub) Schema
    - defines the external view of data

      as seen by a user or program
  - Conceptual Schema
    - defines the logical view of data

      as seen by all users and programs
  - Internal Schema
    - defines the physical view of data

      as seen by a DBMS

Mapping between Levels

- DBMS map or translate from one level to another.
  - External ⇔ Conceptual

  External schema is related to the conceptual schema
  - Conceptual ⇔ Internal

  Conceptual schema is related to the internal schema.

Data Independence

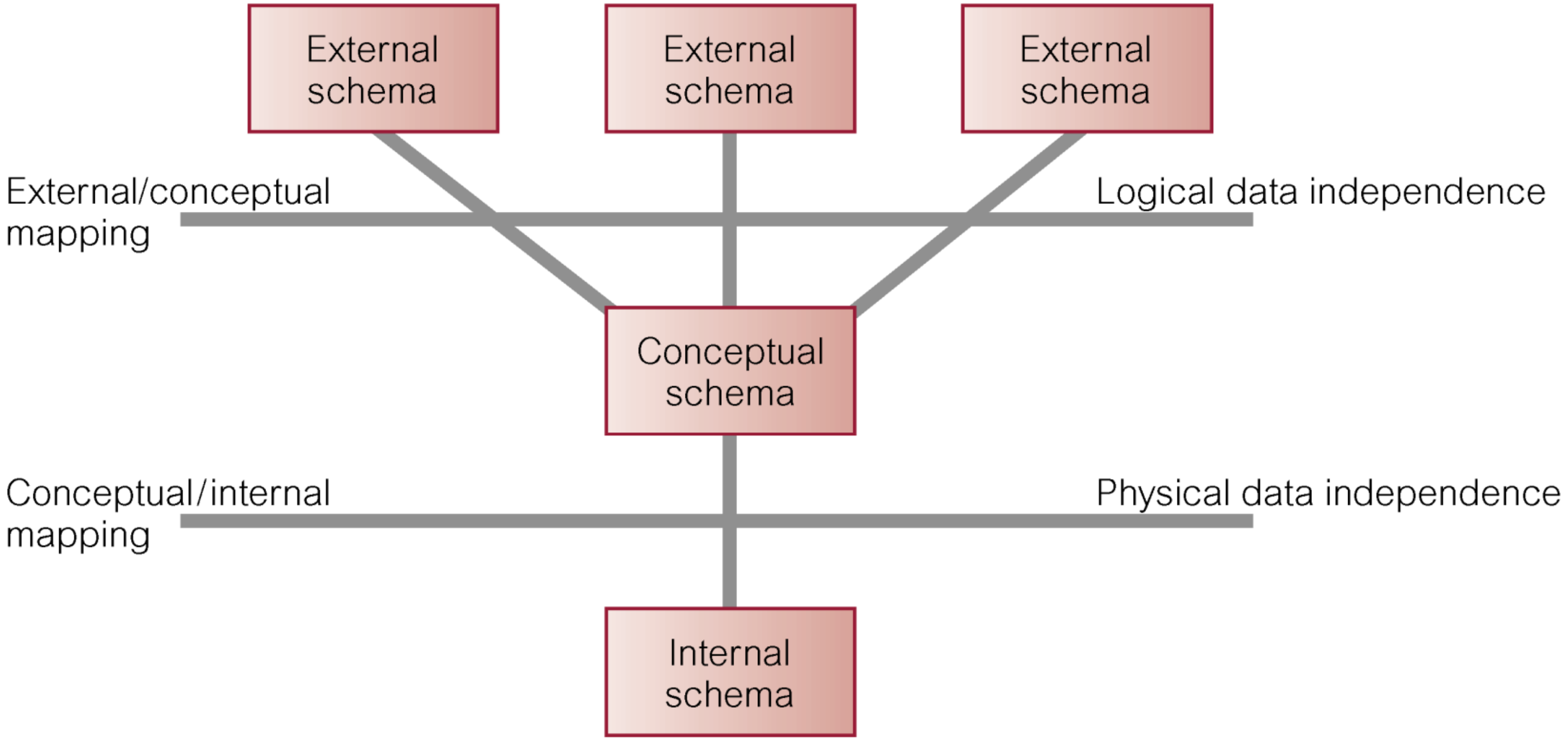
- **Logical Data Independence**
  - Refers to immunity of external schemas to changes in conceptual schema.
  - Conceptual schema changes

    (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 ANSI-SPARC Three-Level Architecture


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

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.
- **Non-Procedural DML**
  - Allows user to state what data is needed rather than how it is to be retrieved.
- **Procedural DML**
  - Allows user to tell system exactly how to manipulate data.

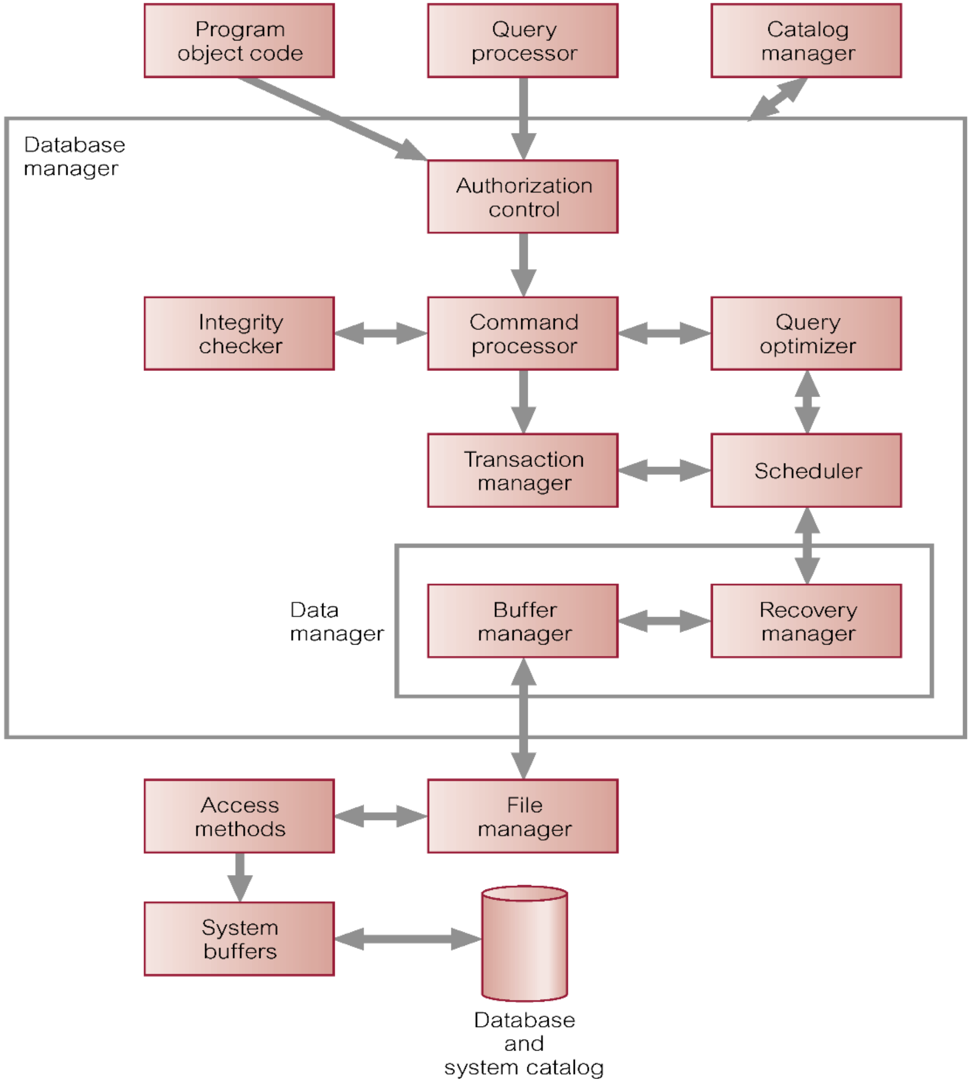
Functions of a DBMS

- Data Storage, Retrieval, and Update.
- A User-Accessible Catalog.
- Transaction Support.
- Concurrency Control Services.
- Recovery Services.

Functions of a DBMS

- Authorization Services.
- Integrity Services.
- Utility Services.

Components of Database Management System



Catalog Manager

- The catalog manager manages access to and maintain the system catalog.

System Catalog

- Repository of information (metadata) describing the data in the database.
- Typically stores:
  - Names of authorized users;
  - Names, types, and sizes of data items in the database;
  - Names of relationships
  - Constraints on each data item;

System Catalog

- Data items accessible by a user and the type of access allowed (e.g. insert, delete, update or read access).
- External, conceptual and internal schemas and the mappings between the schemas.
- Used by modules such as Authorization Control and Integrity Checker.

System Catalog - Advantages

- Maintains control over the data as a resource.
- Helps other users understand the purpose of the data.
- Communication is simplified.
- Identifies the user/s who own or access the data.
- Security can be enforced.
- Integrity can be ensured

Components of DB Manager

- **Authorization Control**

  This module checks that the user has the necessary authorization to carry out the required operation.

- **Integrity Checker**

  For an operation that changes the database, the integrity checker checks that the requested operation satisfies all necessary integrity constraints.

Components of DB Manager

- **Scheduler**

  Responsible for ensuring that concurrent operations on the database proceed without conflicting with one another. It controls the relative order in which the transaction operations are executed.

- **Recovery Manager**

  Ensures that the database remains in a consistent state in the presence of failure.

END