Database Management Systems

Shamim Ahmad

Languages

- Every task requires extensive programming in a third-generation language (3GL)
  - Programmer must **specify task and how it must be done**
- Modern databases use fourth-generation languages (4GL)
  - Allow users to **specify what must be done** without specifying **how it is to be done**

Problems with File System Data Management

TABLE 1.3 3GL vs. 4GL Sample Code

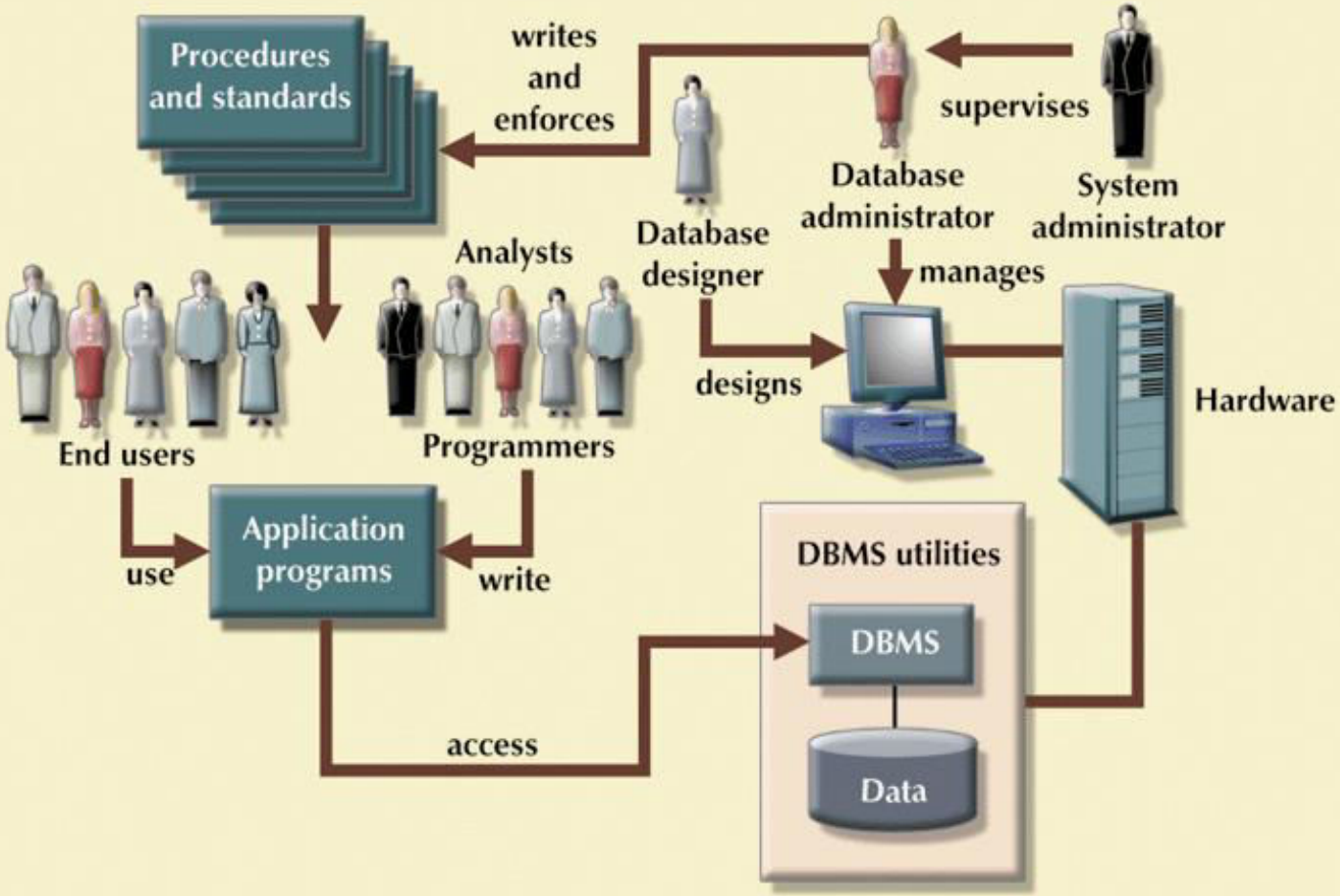
3GL (GENERIC CODE)	4GL (SQL CODE)
DO WHILE NOT EOF() READ CUSTOMER IF CUSTOMER.C_ZIP = "36123" THEN PRINT C_NAME, C_PHONE, C_ZIP; ENDDO;	SELECT C_NAME, C_PHONE, C_ZIP FROM CUSTOMER WHERE CUSTOMER.C_ZIP = '36123';

The Database System Environment

- Database system is composed of five main parts:
  - Hardware
  - Software
    - Operating system software
    - DBMS software
    - Application programs and utility software
  - People
  - Procedures
  - Data

The Database System Environment (continued)

FIGURE 1.7 The database system environment



DBMS Functions

- DBMS performs functions that guarantee integrity and consistency of data
  - Data **dictionary** management
    - defines data elements and their relationships
  - Data **storage** management
    - stores data and related data entry forms, report definitions, etc.

DBMS Functions (continued)

- Data transformation and presentation
  - translates **logical requests** into commands **to physically locate** and retrieve the requested data
- Security management
  - enforces **user security and data privacy** within database

DBMS Functions (continued)

- Multiuser access control
  - uses sophisticated algorithms to ensure multiple users can access the database **concurrently** without compromising the integrity of the database
- Backup and recovery management
  - provides **backup and data recovery** procedures
- Data integrity management
  - promotes and enforces **integrity rules**

DBMS Functions (continued)

- Database access languages and application programming interfaces
  - provide data access through a **query language**
- Database communication interfaces
  - allow database to accept end-user requests via **multiple, different network environments**

Data Abstraction

Data Abstraction

- Major aim of a DBMS is to provide users with an abstract view of data
- Hides certain details of how the data are stored & maintained
- DBMS must retrieve data efficiently
- Need for efficiency has led designers to use complex data structures to represent the data in the database
- Most DB users are not computer trained, developers hide complexity through several levels of abstraction to simplify user's interaction with the systems

Levels of Abstraction

- Physical level describes how a record (e.g., customer) is stored.
- Logical level: describes data stored in database, and the relationships among the data.

```
type customer = record
                    name : string;
                    street : string;
                    city : integer;
                end;
```

- View level: application programs hide details of data types. Views can also hide information (e.g., salary) for security purposes.

Levels of Abstraction

- Physical level describes how a record (e.g., customer) is stored.
- Logical level: describes data stored in database, and the relationships among the data.

```
type customer = record
                    name : string;
                    street : string;
                    city : integer;
                end;
```

- View level: application programs hide details of data types. Views can also hide information (e.g., salary) for security purposes.

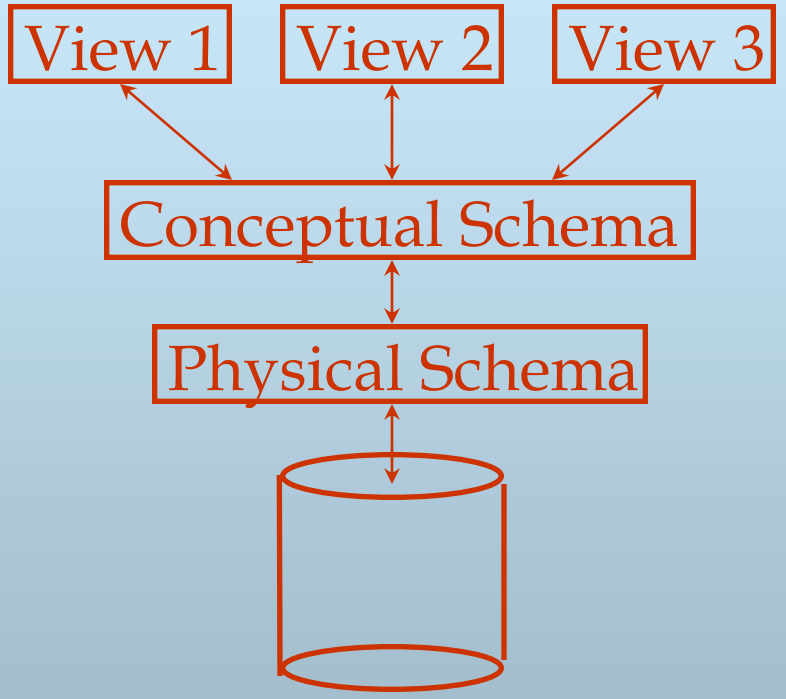
3 Levels of Abstraction

- **Logical or Conceptual Level**
  - Describes what data are stored in the DB & what relationships exist among those data
  - Describes the entire DB in terms of relatively simpler structures
  - Implementation of these simple structures at this level may involve complex physical-level structures
  - Users of the logical level need not be aware of this complexity
  - DBAs, who decide what information to keep in DB, use the logical level of abstraction

Levels of Abstraction

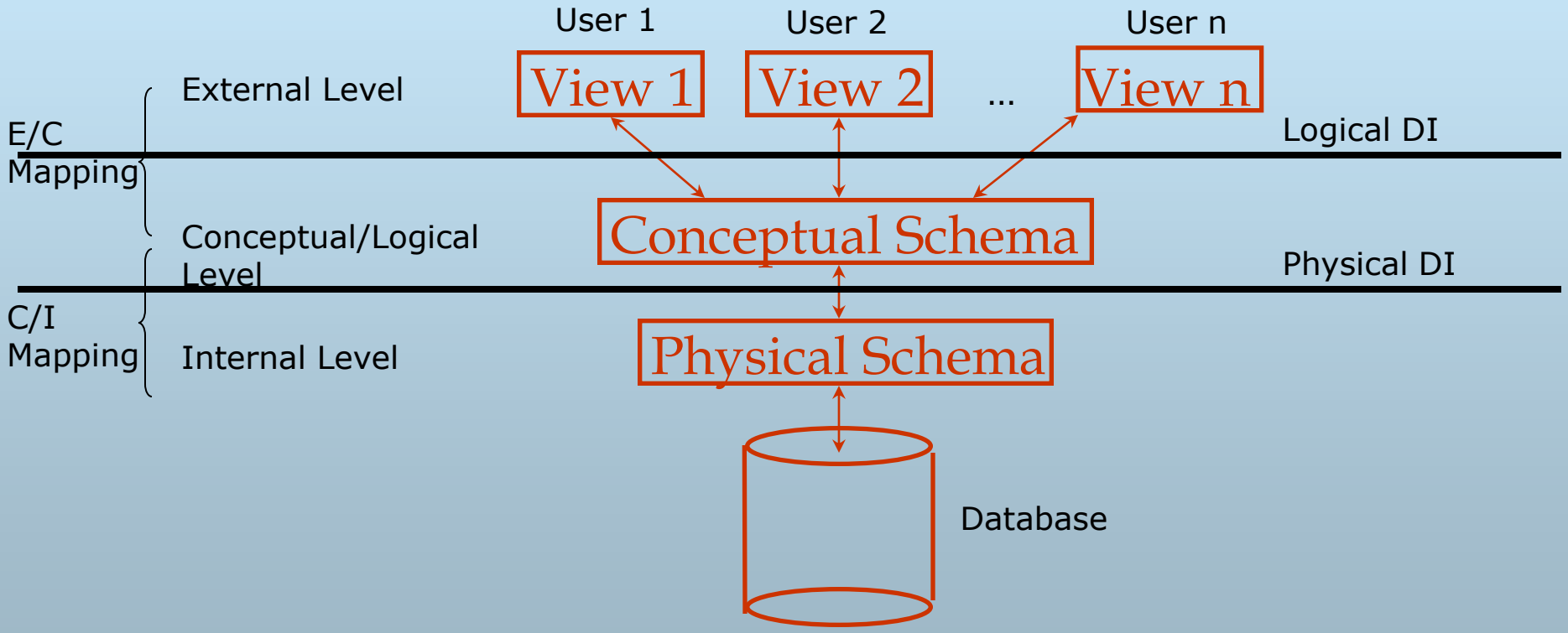
Many *views*, single *conceptual (logical) schema* and *physical schema*.

- **Views describe how users see the data.**
- **Conceptual schema defines logical structure**
- **Physical schema describes the files and indexes used.**



- *Schemas are defined using DDL; data is modified/queried using DML.*

ANSI/SPARC 3-Tier Architecture



Instances & Schemas

Collection of information stored in the DB at a particular moment is called an INSTANCE

The overall design of the DB is called a SCHEMA

A DB has many schemas

- **Physical**
- **Conceptual/Logical**
- **Sub-schemas**

Example: University Database

- **Logical schema:**
  - ***Students(sid: string, name: string, login: string, age: integer, gpa:real)***
  - ***Faculty(fid:string, fname:string, sal:real)***
  - ***Courses(cid: string, cname:string, credits:integer)***
  - ***Enrolled(sid:string, cid:string, grade:string)***
- **Physical schema:**
  - **Relations stored as unordered files.**
  - **Index on first column of Students.**
- **External Schema (View):**
  - ***Course_info(cid:string,fname:string, enrollment:integer)***

Data Models

- A collection of tools for describing
  - data
  - data relationships
  - data semantics
  - data constraints
- Entity-Relationship model
- Relational model
- Other models:
  - object-oriented model
  - semi-structured data models
  - Older models: network model and hierarchical model