Database System Architecture

Schemas

- Is the description of the database (*not database itself*)
  - Specified during database design
  - Not expected to change frequently
  - A displayed schema is called a schema diagram (Fig 2.1)
- Each object in the schema-such as STUDENT or COURSE-is a schema construct.
- Schema diagram represents only some aspects of a schema (name of record type, data element and some type of constraint)

Instances and Schemas

- Similar to types and variables in programming languages
- **Schema** – the logical structure of the database
  - e.g., the database consists of information about a set of customers and accounts and the relationship between them)
  - Analogous to type information of a variable in a program
  - **Physical schema**: database design at the physical level
  - **Logical schema**: database design at the logical level
- **Instance** – the actual content of the database at a particular point in time
  - Analogous to the value of a variable
- **Physical Data Independence** – the ability to modify the physical schema without changing the logical schema
  - Applications depend on the logical schema
  - In general, the interfaces between the various levels and components should be well defined so that changes in some parts do not seriously influence others.

Data Definition Language (DDL)

- Specification notation for defining the database schema
  - E.g.

    ```
    create table account (
            account-number   char(10),
            balance          integer)
    ```

- DDL compiler generates a set of tables stored in a *data dictionary*
- Data dictionary contains metadata (i.e., data about data)
  - database schema
  - Data *storage and definition* language
    - language in which the storage structure and access methods used by the database system are specified
    - Usually an extension of the data definition language

Data Manipulation Language (DML)

- Language for accessing and manipulating the data organized by the appropriate data model
  - DML also known as query language
- Two classes of languages
  - Procedural – user specifies what data is required and how to get those data
  - Nonprocedural – user specifies what data is required without specifying how to get those data
- SQL is the most widely used query language

SQL

- SQL: widely used non-procedural language
  - E.g. find the name of the customer with customer-id 192-83-7465

```
select   customer.customer-name
from     customer
where    customer.customer-id = '192-83-7465'
```

  - E.g. find the balances of all accounts held by the customer with customer-id 192-83-7465

```
select   account.balance
from     depositor, account
where    depositor.customer-id = '192-83-7465' and
        depositor.account-number = account.account-number
```

- Application programs generally access databases through one of
  - Language extensions to allow embedded SQL
  - Application program interface (e.g. ODBC/JDBC) which allow SQL queries to be sent to a database

Database Users

- Users are differentiated by the way they expect to interact with the system
- Application programmers – interact with system through DML calls
- Sophisticated users – form requests in a database query language
- Specialized users – write specialized database applications that do not fit into the traditional data processing framework
- Naïve users – invoke one of the permanent application programs that have been written previously
  - E.g. people accessing database over the web, bank tellers, clerical staff

Database Administrator

- Coordinates all the activities of the database system; the database administrator has a good understanding of the enterprise’s information resources and needs.
- Database administrator's duties include:
  - Schema definition
  - Storage structure and access method definition
  - Schema and physical organization modification
  - Granting user authority to access the database
  - Specifying integrity constraints
  - Acting as liaison with users
  - Monitoring performance and responding to changes in requirements

Figure 2.1 Schema diagram for the database of Figure 1.2.

STUDENT

Name	StudentNumber	Class	Major

COURSE

CourseName	CourseNumber	CreditHours	Department

PREREQUISITE

CourseNumber	PrerequisiteNumber

SECTION

SectionIdentifier	CourseNumber	Semester	Year	Instructor

GRADE_REPORT

StudentNumber	SectionIdentifier	Grade

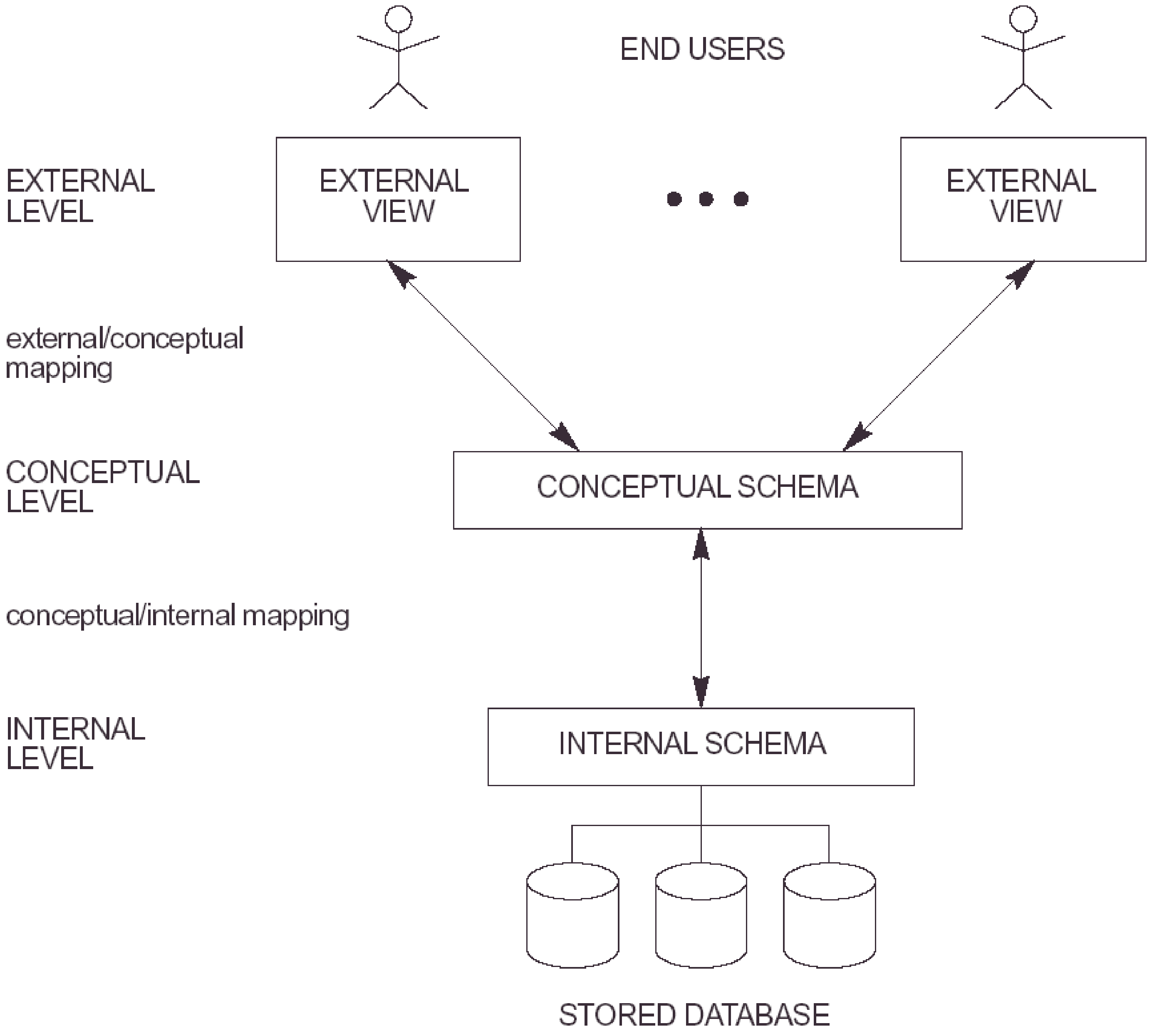
Instances and Database State

- The data in the database at a particular moment in time is called a *database state or snapshot* or *current set of occurrences or instances in the database*
- When we define a new database we have database state is *empty state* (schema specified only in DBMS)
- The *initial state* when the database is first populated
- Then At any point in time, the database has a *current state*
- schema evolution: when we need to change the schema

The Three-Schema Architecture

- Importance of using DB approach
  - insulation of programs and data
  - support of multiple user views
  - use of a catalog to store the database description (schema).
- The aim is to separate the user application and physical DB
- schema can be defined into three levels:
  - The internal level has an internal schema
  - describes the physical storage structure of the database.
  - uses a physical data model

Figure 2.2 Illustrating the three-schema architecture.



The Three-Schema Architecture

- The conceptual level has a conceptual schema describing the structure of the whole database for a community of users.
- It hides the details of physical storage structures and concentrates on describing entities, data types, relationships, user operations, and constraints.
- A high-level data model or an implementation data model can be used at this level.
- The external or view level includes a number of external schemas or user views describing the part of the db that a particular user group is interested in and hides the rest of the db from that user group.
- A high-level data model or an implementation data model can be used at this level.

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.

AN ARCHITECTURE FOR DATABASE SYSTEM

- EXTERNAL LEVEL
  How data is viewed by an individual user

  CONCEPTUAL LEVEL
  How data is viewed by a community of users

  INTERNAL LEVEL
  How data is physically stored

- **THE EXTERNAL LEVEL**
- Application programmer uses a HOST LANGUAGE: COBOL, PL/1, C
  Embedded in the host language is a DATA SUBLANGUAGE DSL

  Example: SQL, dBASE

  Data Sublanguage consists of:
  Data Definition Language DDL
  Data Manipulation Language DML

  Data Definition Language declares database objects

  Data Manipulation Language manipulates database objects
  e.g. retrievals and updates

- **THE CONCEPTUAL LEVEL**
- A representation of the entire information content of the database

  The conceptual schema is a definition of the view of the total database content

  Conceptual schema, in most cases, is the union of external schemas

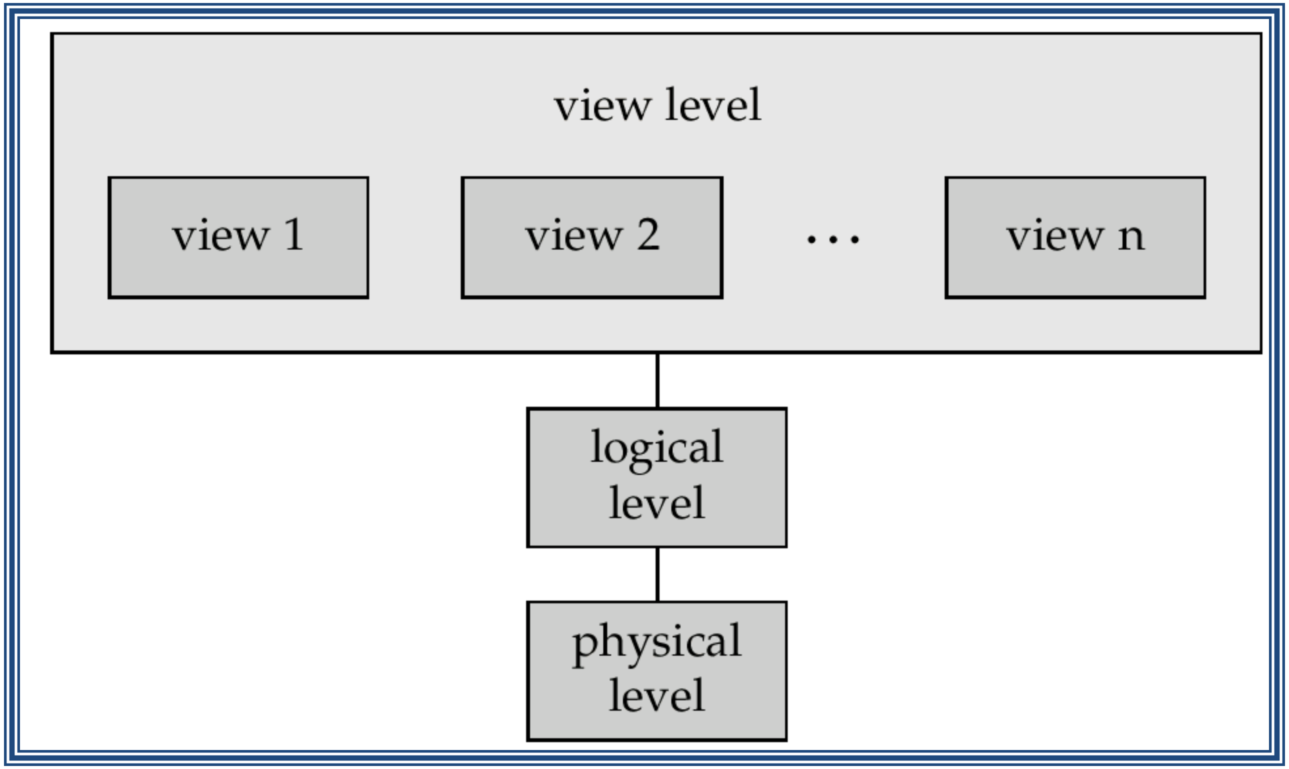
  One can add: security and integrity checks, semantic models and data dictionary

- The internal view is a low-level representation of the entire database

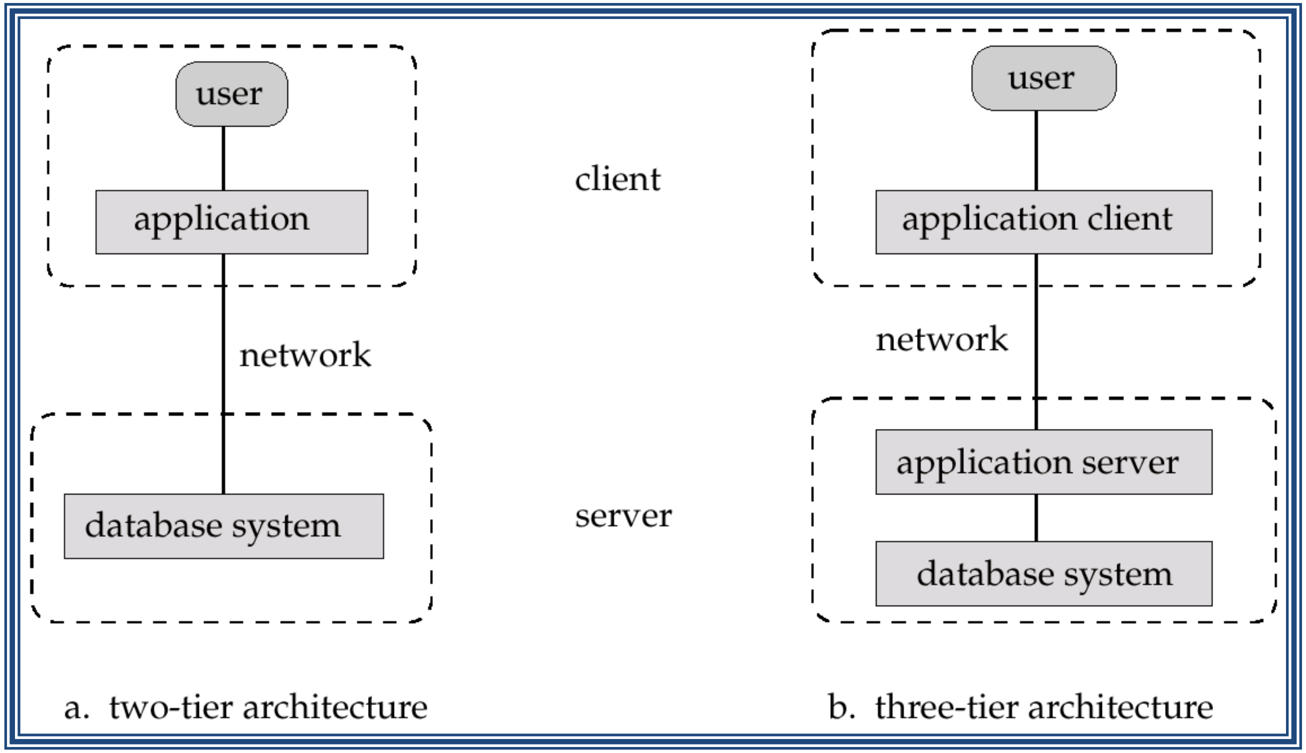
  Internal record, or stored record, is built upon physical records, or pages and blocks

View of Data

An architecture for a database system



Application Architectures



- **Two-tier architecture**: E.g. client programs using ODBC/JDBC to communicate with a database
- **Three-tier architecture**: E.g. web-based applications, and applications built using "middleware"

Data Independence

Logical data independence

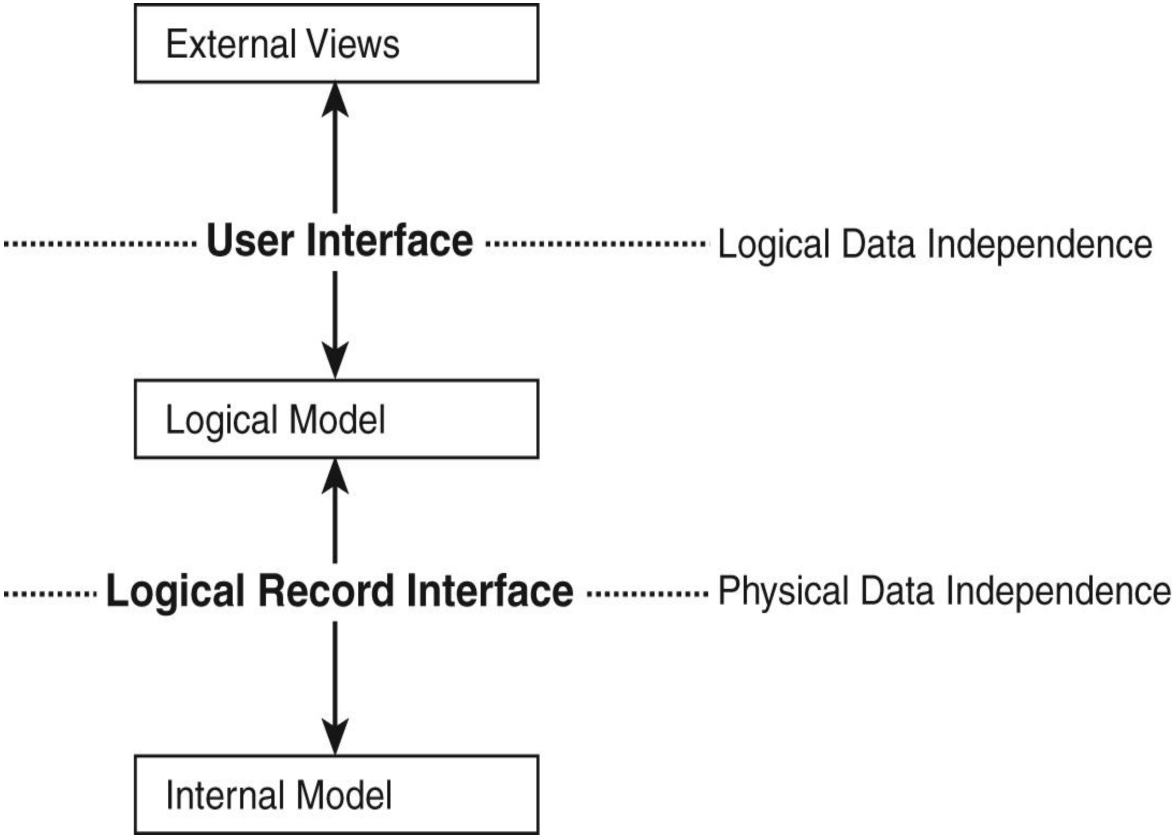
Immunity of external models to changes in the logical model

Occurs at user interface level

Physical data independence

Immunity of logical model to changes in internal model

Occurs at logical interface level



External Views

User Interface

Logical Data Independence

Logical Model

Logical Record Interface

Physical Data Independence

Internal Model