Lecture 3b

Data Models and Database Schemas: The Three Schema Architecture

Week 2

Overview

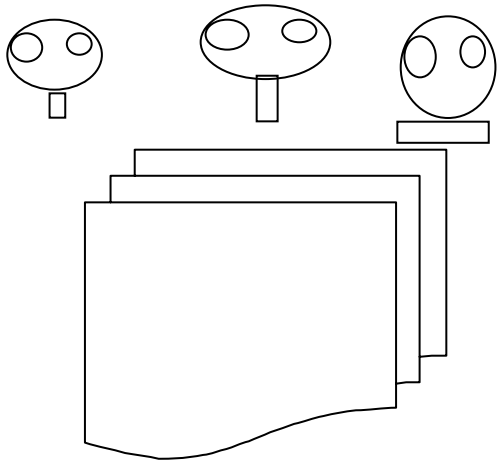
- The ‘three schema’ architecture
- The process of database design
- Logical and physical data independence

The three schema architecture

- A database can be described using three schema levels (not to be confused with data model levels)

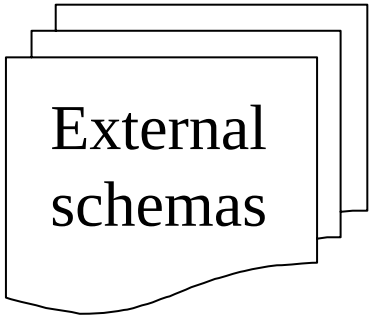
- E.g. using the relational data model we can create
  - external schemas (schemata)
  - conceptual schema
  - internal schema

I

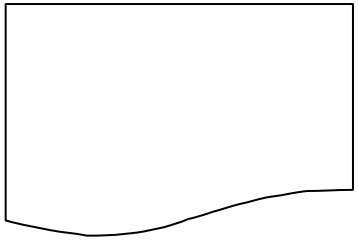


External schemas
for each group of users
(they provide a limited view
of the database)

I

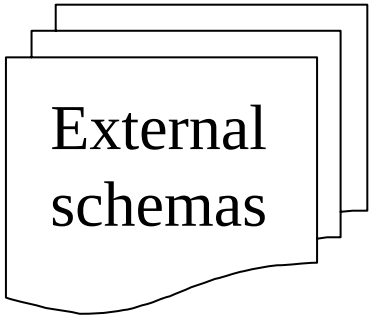


II

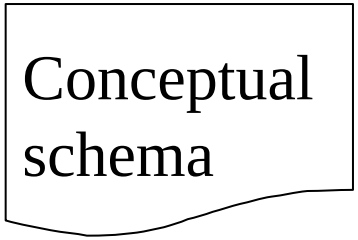


Conceptual schema (describes the database schema as a whole)

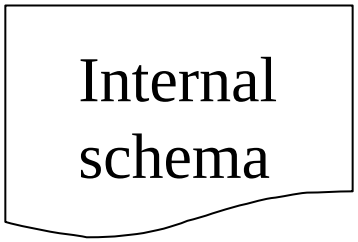
I



II



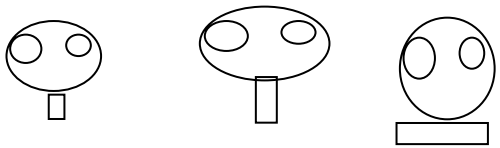
III



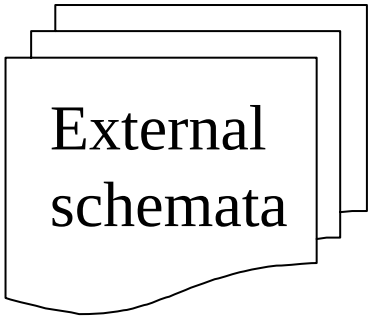
Internal schema
(describes the database's
physical storage properties)

Database design process

- Use conceptual level data model to capture user requirements
- Use conceptual level data model to integrate user requirements into one database schema

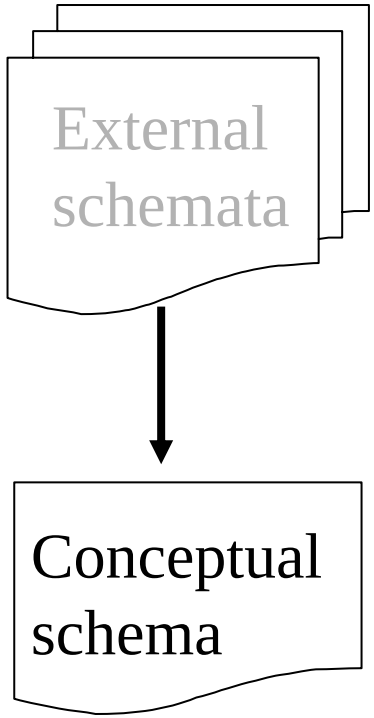


Provide requirements



I

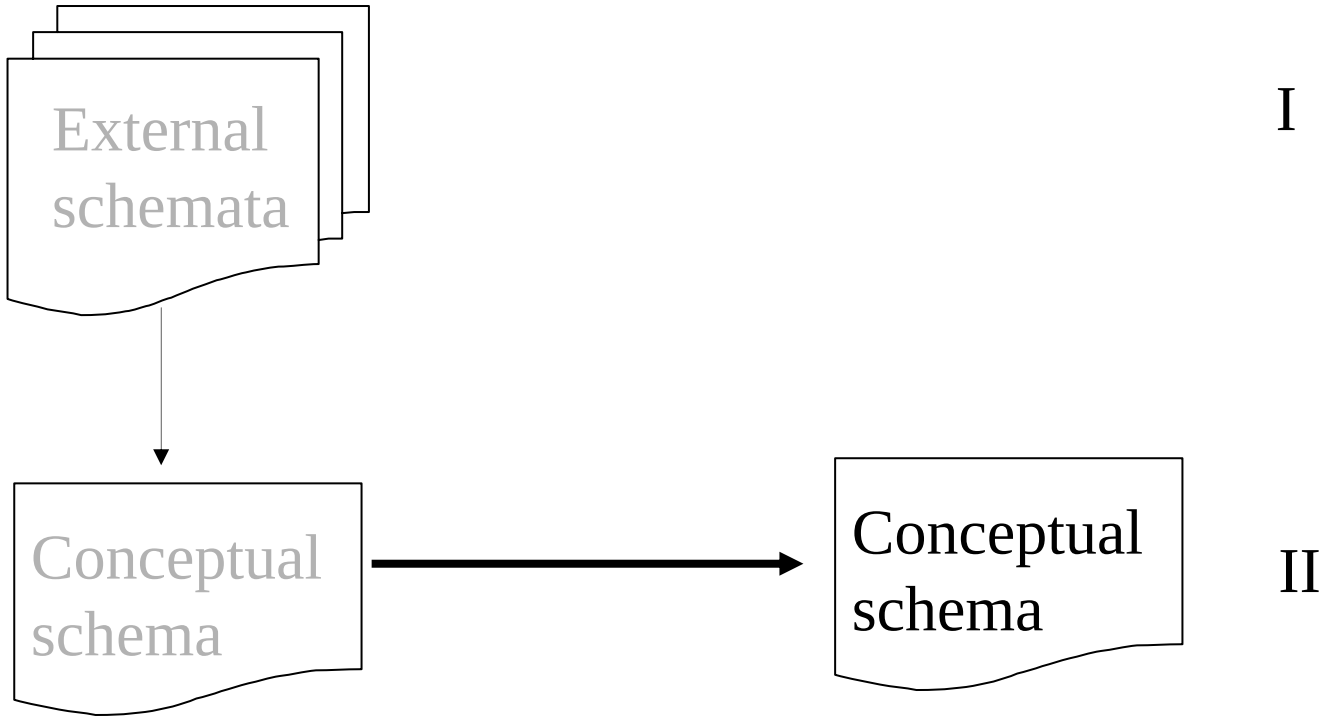
1. Capture user requirements as external schemas written in a conceptual level data model



I

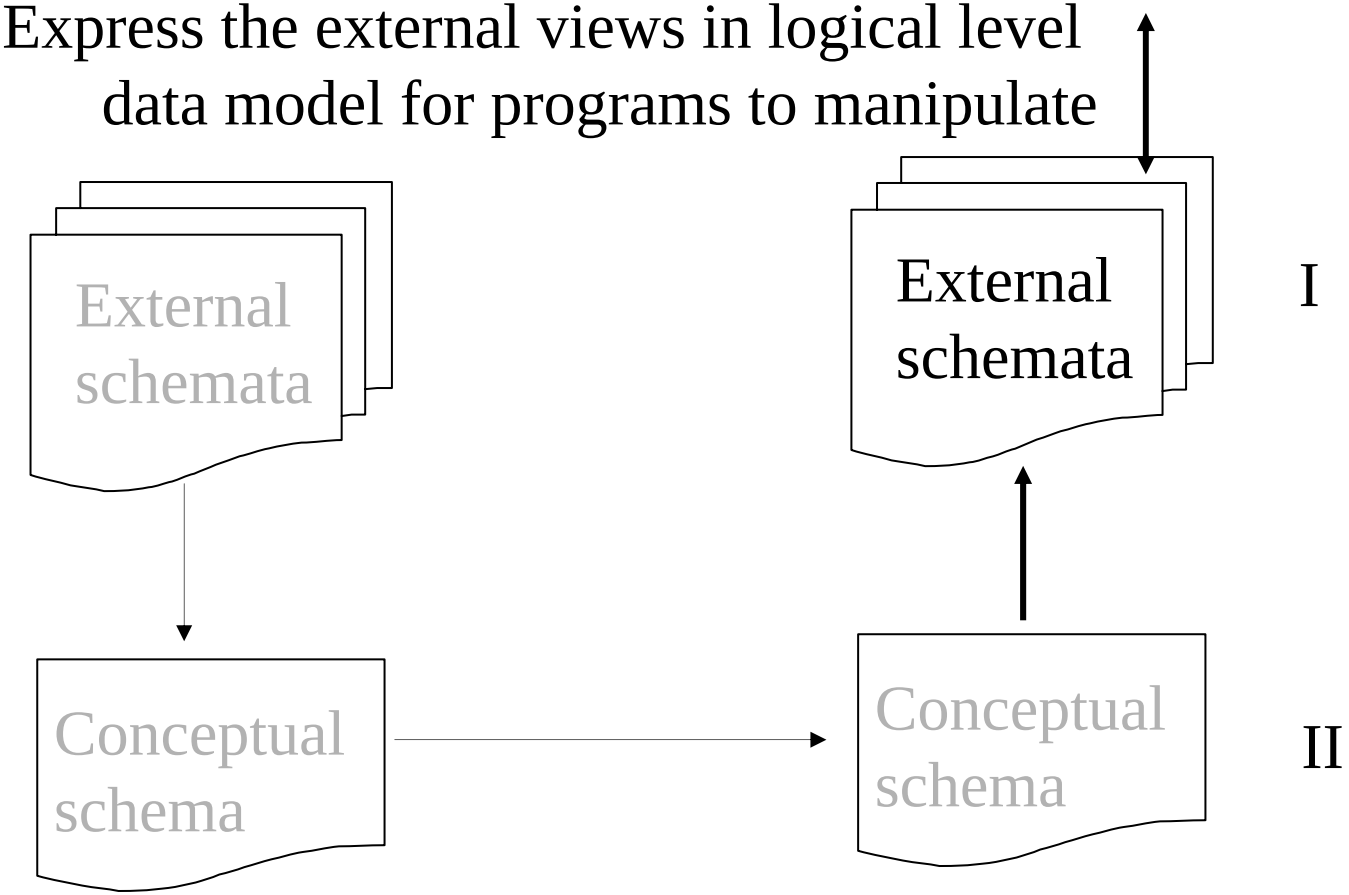
II

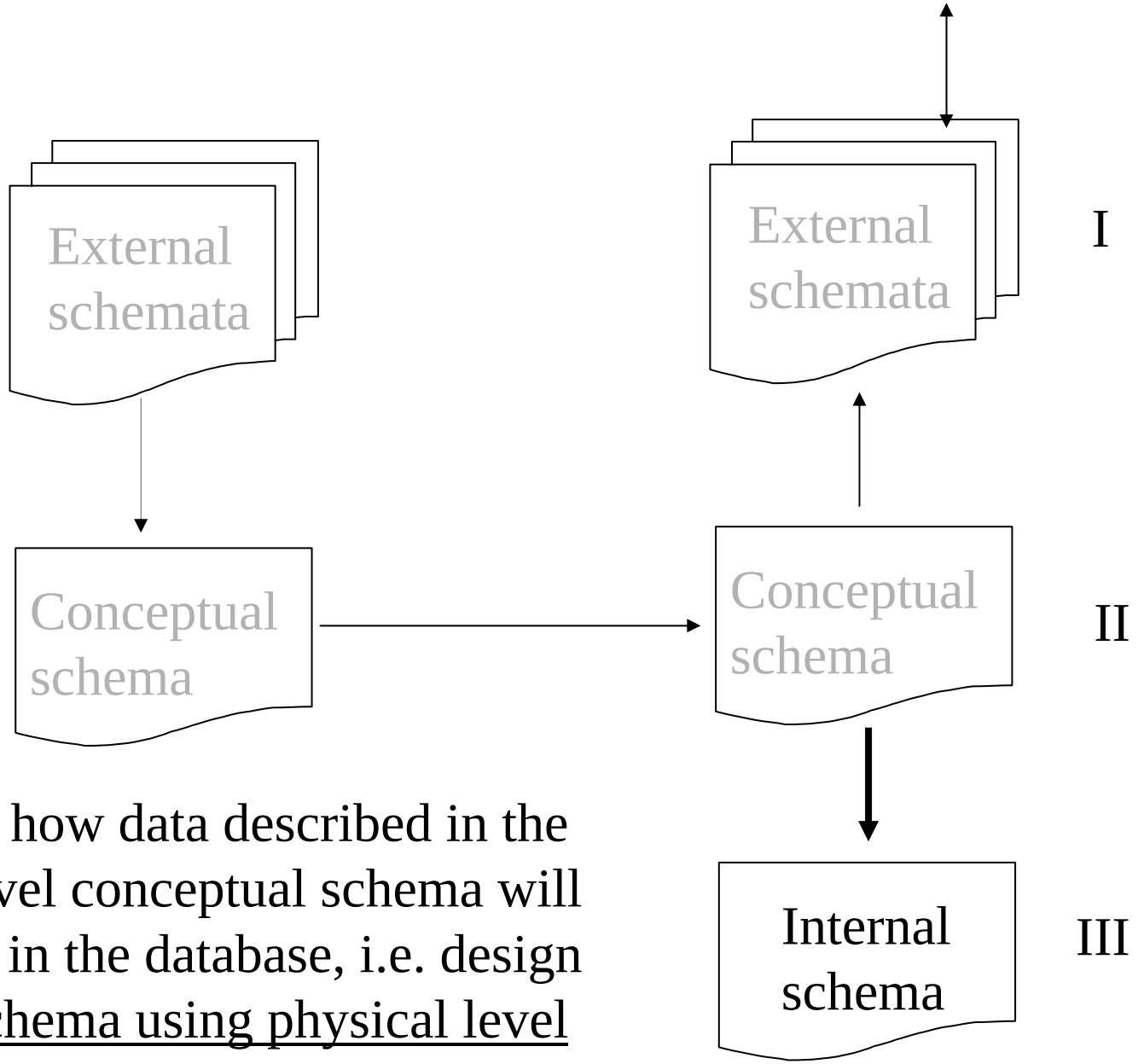
2. Integrate external schemata using conceptual level data model into one conceptual level conceptual schema



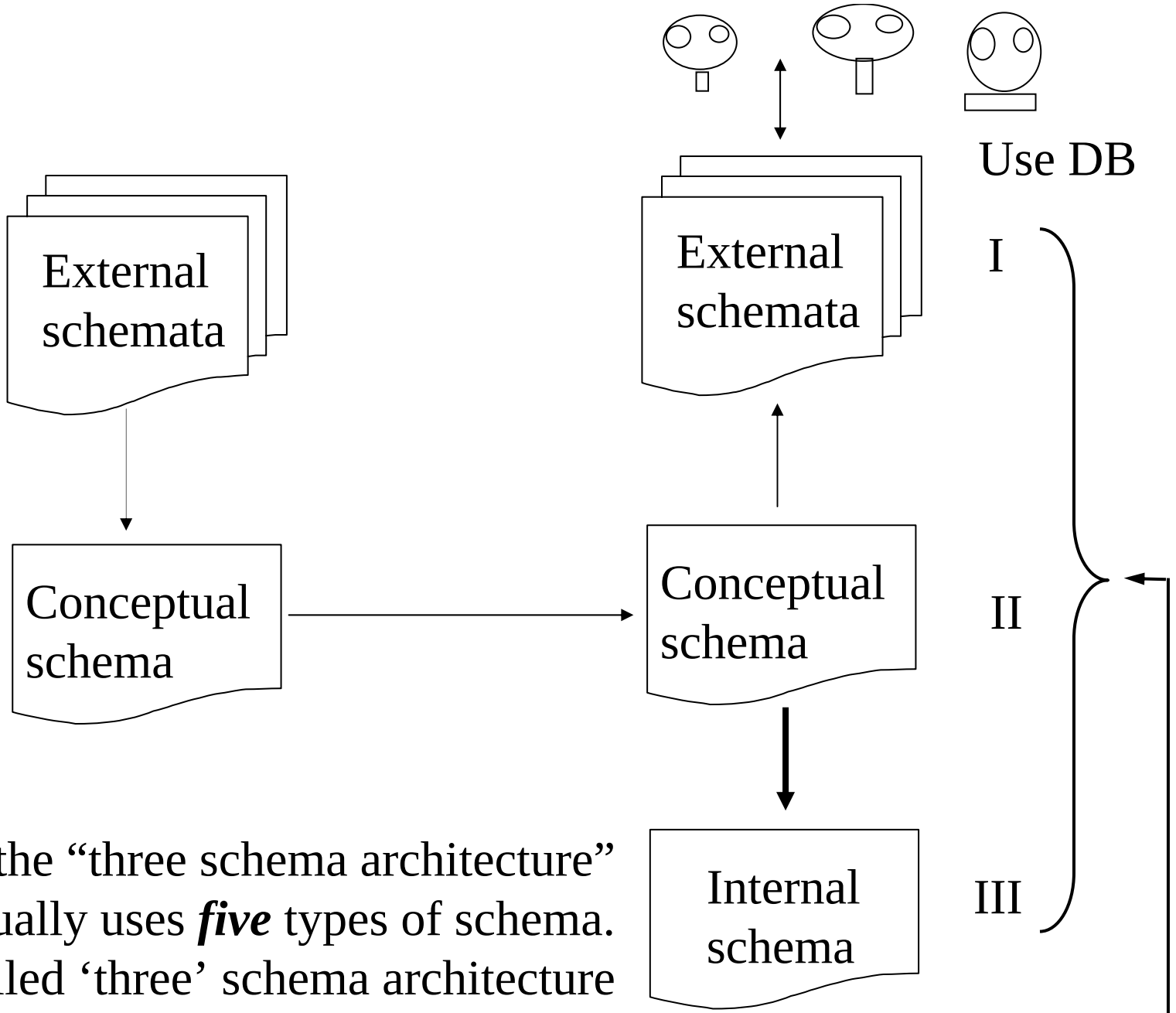
3. Map the conceptual schema to an equivalent conceptual schema expressed in a logical level data model

4. Express the external views in logical level data model for programs to manipulate

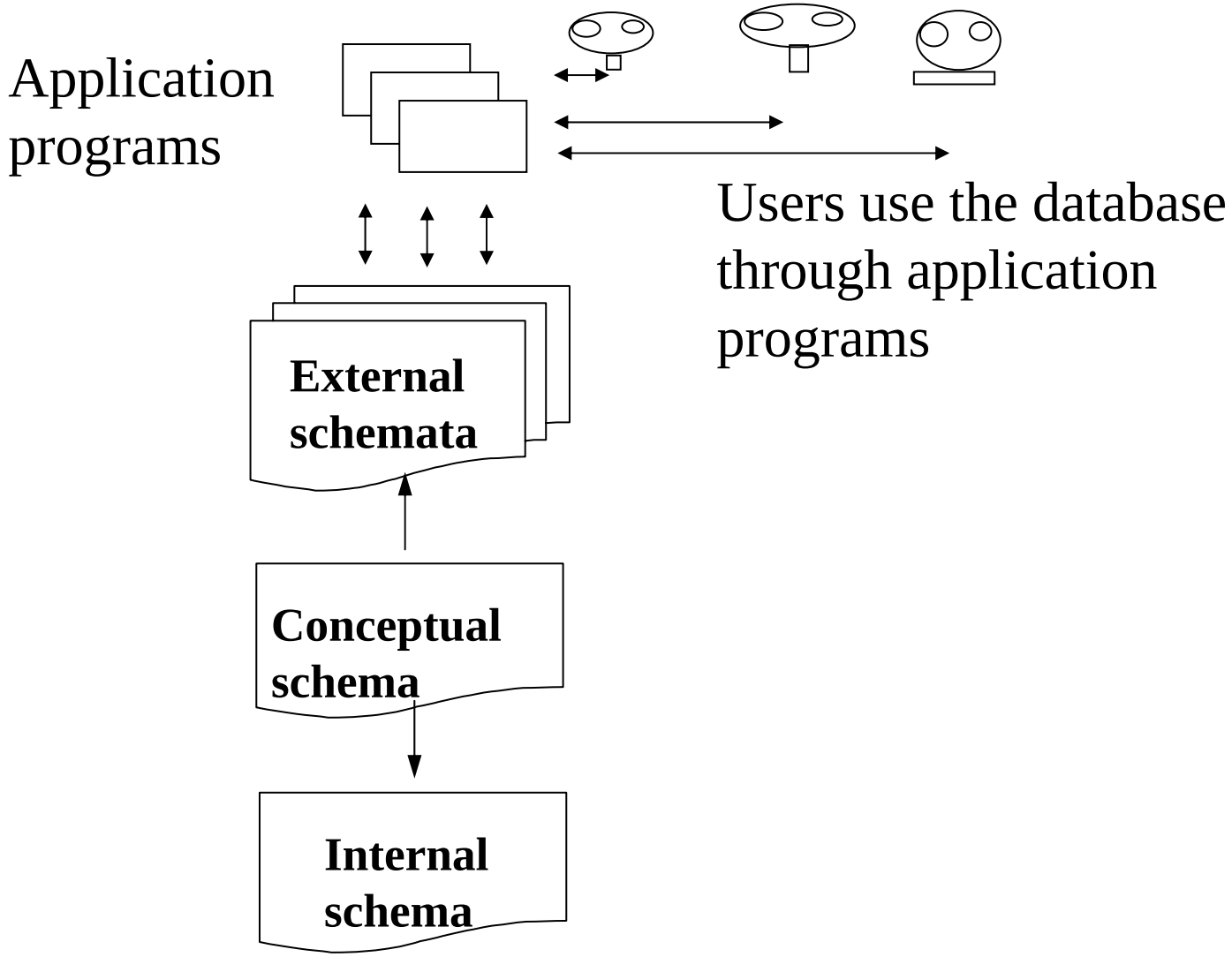




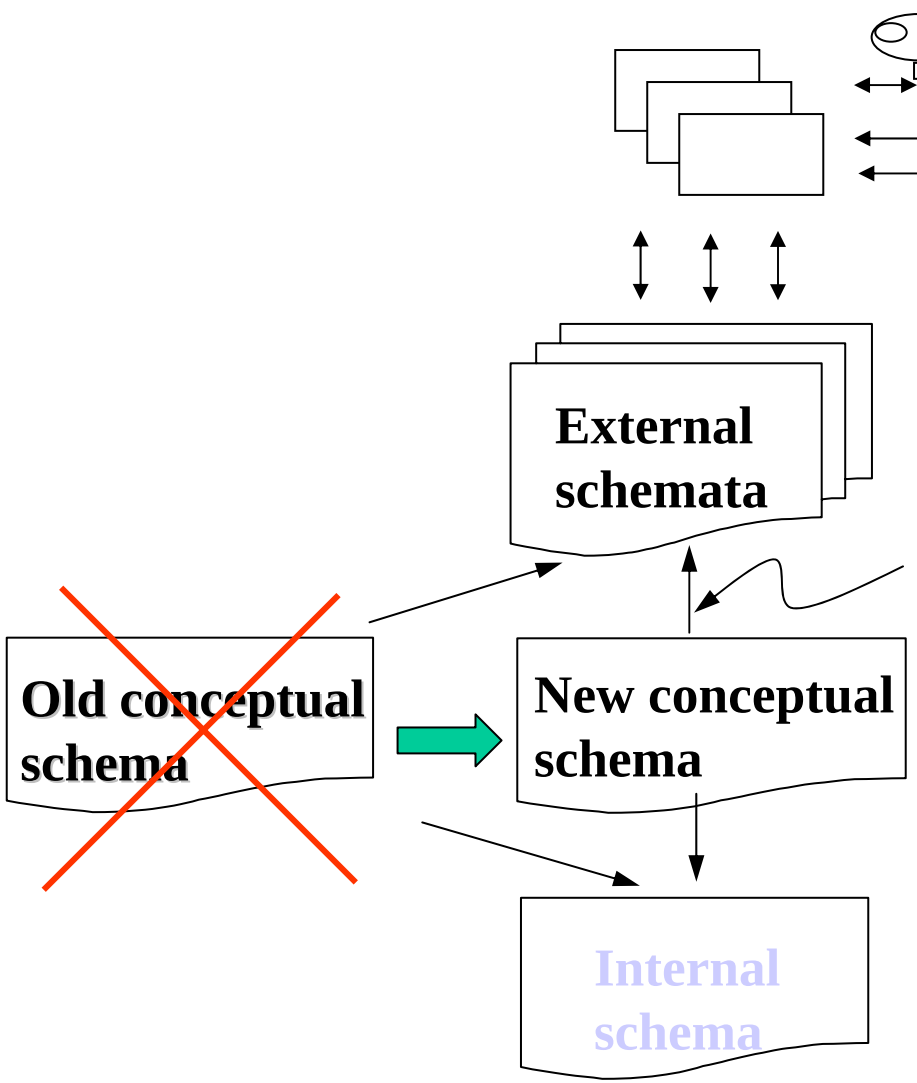
5. Design how data described in the logical level conceptual schema will be stored in the database, i.e. design internal schema using physical level data model (logical level + storage)



This is how the “three schema architecture” actually uses ***five*** types of schema. (called ‘three’ schema architecture because of the three levels)



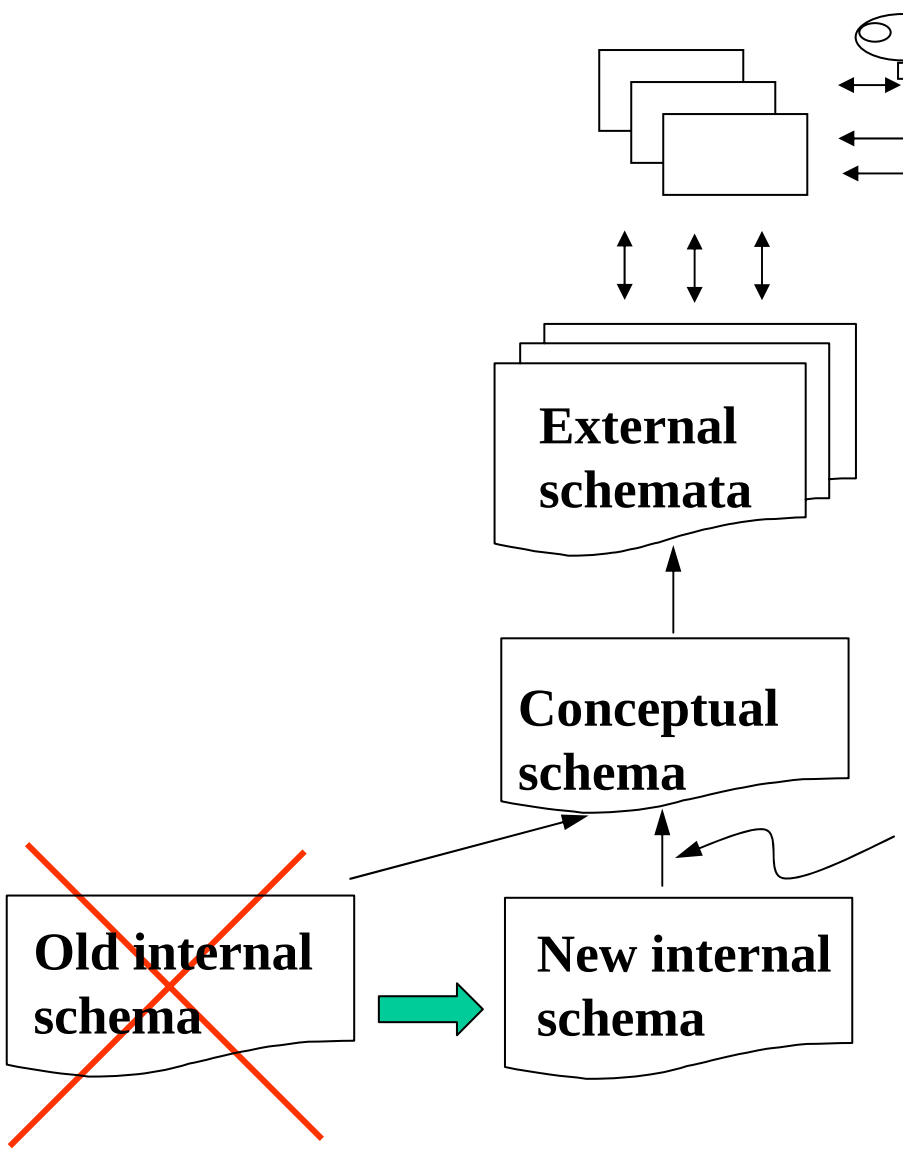
Logical data independence



If the conceptual schema changes the database administrator may define a *new way* to derive the *same* external schemata

therefore application programs will not notice that the conceptual schema changed

Physical data independence



If the physical schema changes the database administrator defines a *new way to store* the *same* conceptual schema therefore the conceptual schema is immune from implementation changes made for efficiency reasons

The end