Schemas, Instances, and Database State

SONY P, Assistant Professor

Model Engineering College

May 1, 2021

1. Schemas, Instances, and Database State

2. Three-Schema Architecture
   - Internal level
   - Conceptual level
   - External or view level
   - Mappings
   - Data Independence
     - Logical data independence
     - Physical data independence

Schemas, Instances, and Database State

1. The description of a database is called the **database schema**, which is specified during database design and is not expected to change frequently
2. A displayed schema is called a **schema diagram**. We call each object in the schema a schema construct.
3. The data in database at particular instant or moment of time is called **database state or snapshot**

Example of a Database Schema

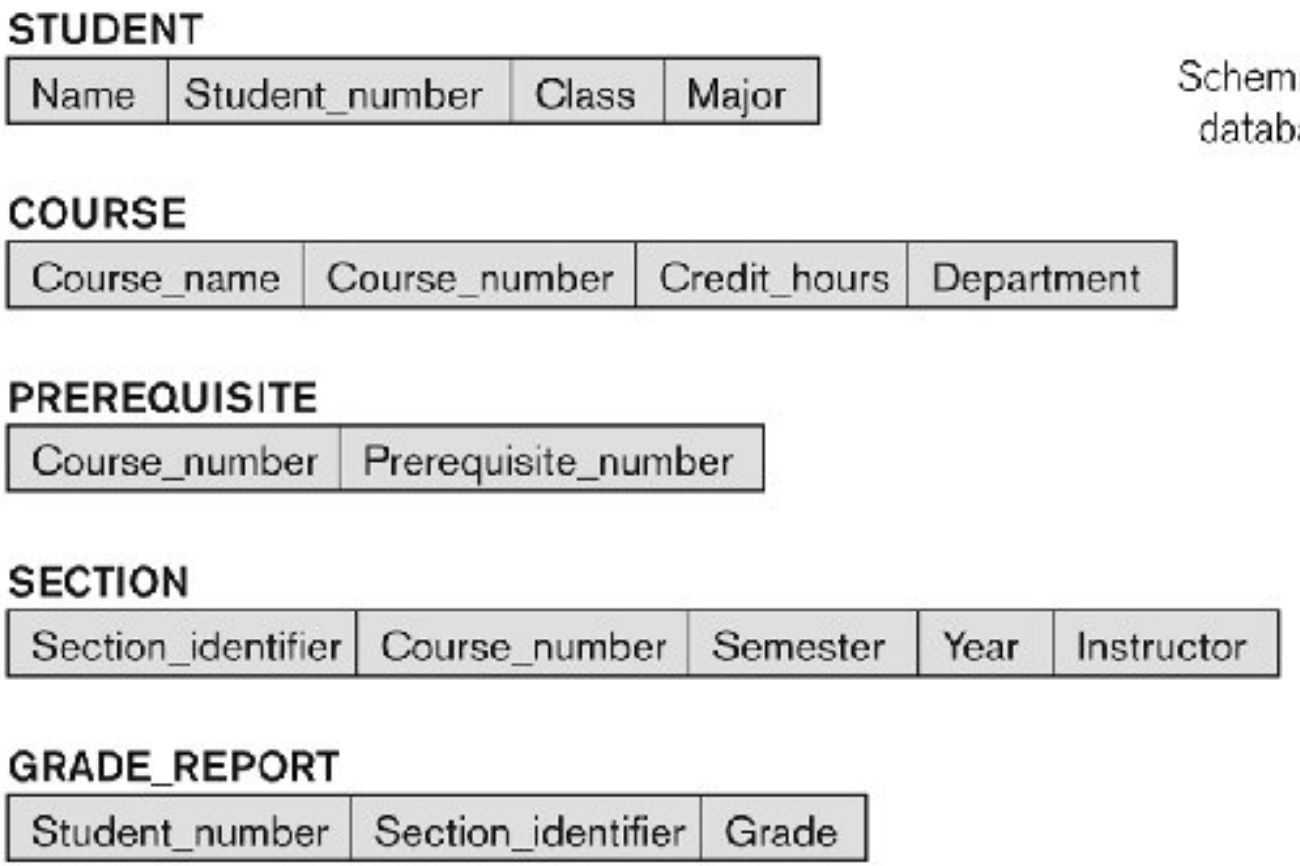

Figure 2.1
Schema diagram for the database in Figure 1.2.

Example of a database state

COURSE

Course_name	Course_number	Credit_hours	Department
Intro to Computer Science	CS1310	4	CS
Data Structures	CS3320	4	CS
Discrete Mathematics	MATH2410	3	MATH
Database	CS3380	3	CS

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	04	King
92	CS1310	Fall	04	Anderson
102	CS3320	Spring	05	Knuth
112	MATH2410	Fall	05	Chang
119	CS1310	Fall	05	Anderson
135	CS3380	Fall	05	Stone

GRADE_REPORT

Student_number	Section_identifier	Grade
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

Figure 1.2
A database that stores student and course information.

The schema is not supposed to change frequently, but it is not uncommon that changes occasionally need to be applied to the schema as the application requirements change. It is called **schema evolution.**

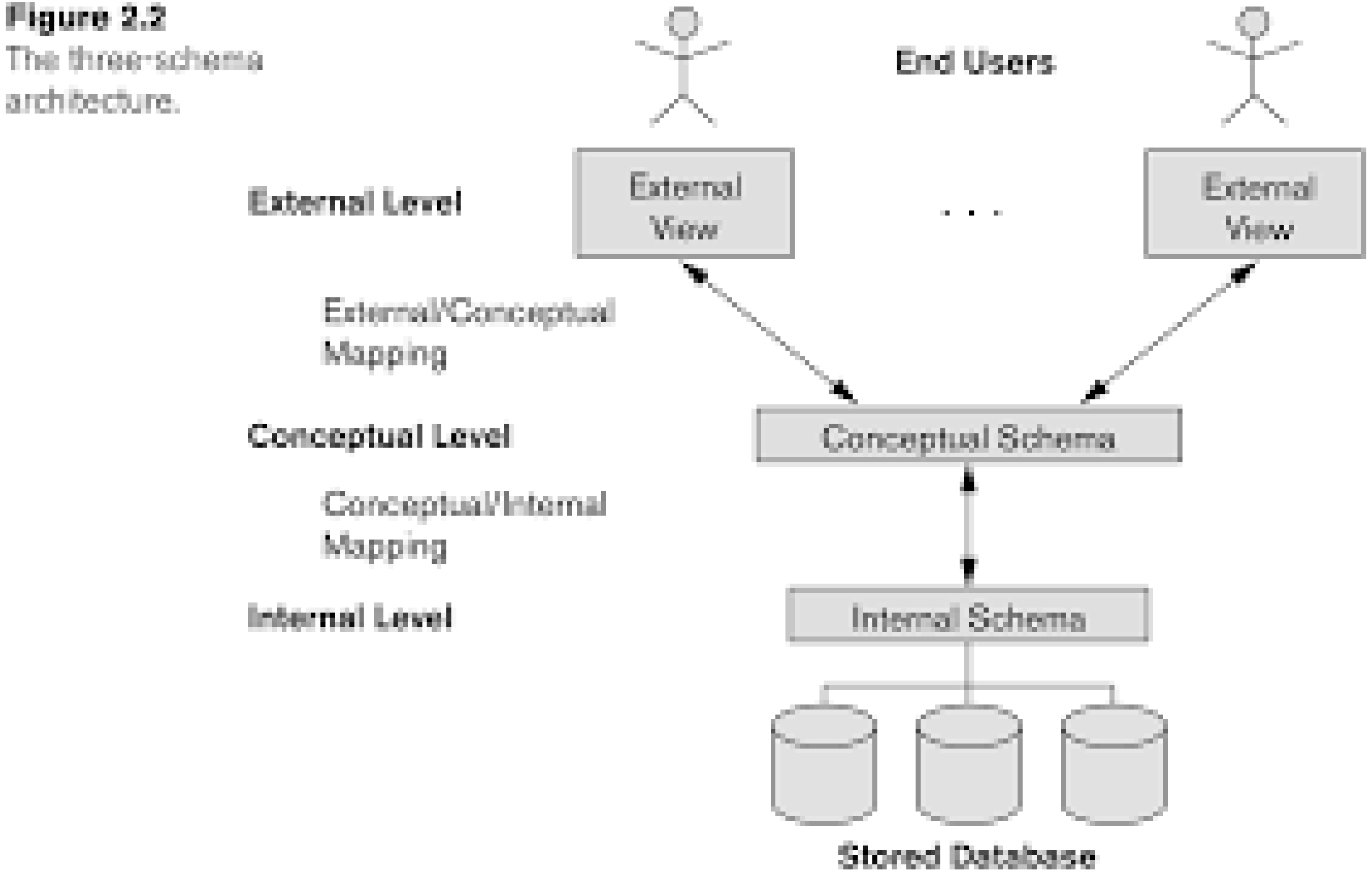

Figure: Three Schema Architecture.

Internal level

1. The internal level has an internal schema, which describes the **physical storage structure** of the database.
2. The internal schema uses a physical data model and describes the complete details of **data storage and access paths** for the database.

Conceptual level

1. Describes the structure of the whole database for a community of users.
2. The conceptual schema hides the details of physical storage structures and concentrates on describing **entities, data types, relationships, user operations, and constraints**.
3. A high-level data model or an implementation data model can be used at this level.

External or view level

1. **The external or view level** includes a number of external schemas or user views.
2. Each external schema describes the part of the database that a particular user group is interested in and hides the rest of the database from that user group.
3. A high level data model or an implementation data model can be used at this level

External or view level

1. **The external or view level** includes a number of external schemas or user views.
2. Each external schema describes the part of the database that a particular user group is interested in and hides the rest of the database from that user group.
3. A high level data model or an implementation data model can be used at this level

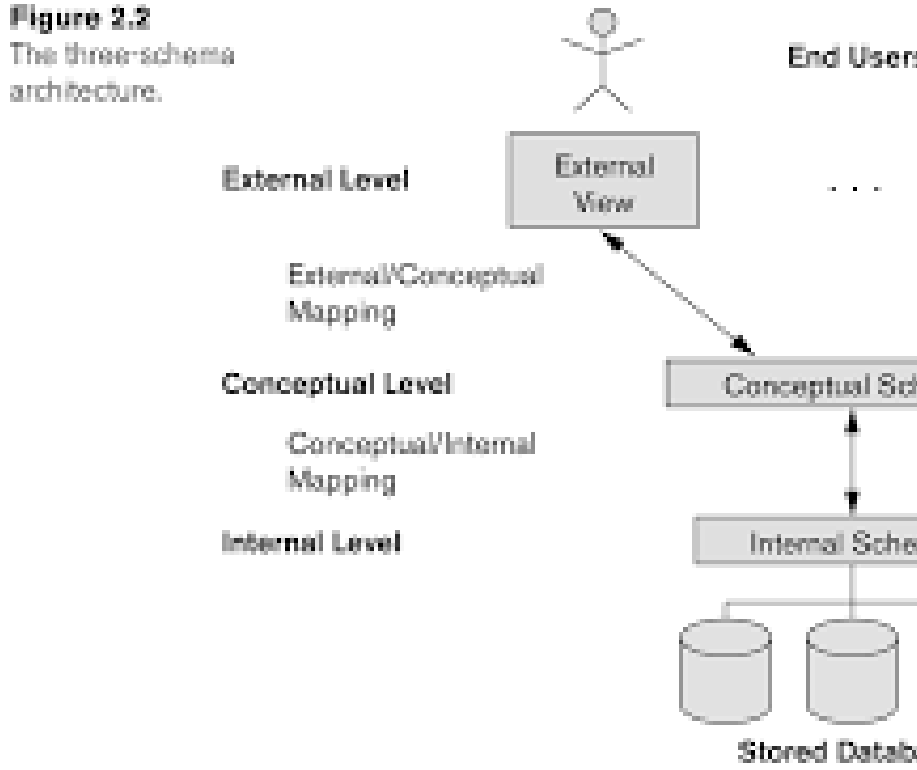

Figure: Three Schema Architecture!

1. In a DBMS based on the three-schema architecture, each user group refers only to its own external schema.
2. Hence, the DBMS must transform a request specified on an external schema into a request against the conceptual schema, and then into a request on the internal schema for processing over the stored database.
3. If the request is a database retrieval, the data extracted from the stored database must be reformatted to match the user's external view.

Mappings

1. The processes of transforming requests and results between levels are called **mappings**
2. These mappings may be time-consuming, so some DBMSs—especially those that are meant to support small databases—do not support external views.
3. a certain amount of mapping is necessary to transform requests between the conceptual and internal levels.

Data Independence

The capacity to change the schema at one level of a database system without having to change the schema at the next higher level. Two types of data independence:
Logical data independence and **Physical data independence**

Logical Data Independence

Logical data independence is the capacity to change the conceptual schema without having to change external schemas or application programs.

Physical data independence

Physical data independence is the capacity to change the internal schema without having to change the conceptual schema.

Data independence occurs because when the schema is changed at some level, the schema at the next higher level remains unchanged; only the mapping between the two levels is changed.