ENSF 608:
What are database systems and architecture?

Dr. Emily Marasco
Fall 2021
Textbook: Fundamentals of Database Systems, 7th Ed., Elmasri & Navathe

Lesson Content

- Data models and model categories
- Schemas, instances, and states
- Three-schema architecture
- DBMS languages and interfaces

Data Models (1 of 2)

- **Data Model:**
  - A set of concepts to describe the **structure** of a database, the **operations** for manipulating these structures, and certain **constraints** that the database should obey.
- **Data Model Structure and Constraints:**
  - Constructs are used to define the database structure
  - Constructs typically include **elements** (and their **data types**) as well as groups of elements (e.g. **entity, record, table**), and **relationships** among such groups
  - Constraints specify some restrictions on valid data; these constraints must be enforced at all times

Data Models (2 of 2)

- **Data Model Operations:**
  - These operations are used for specifying database retrievals and updates by referring to the constructs of the data model.
  - Operations on the data model may include **basic model operations** (e.g. generic insert, delete, update) and **user-defined operations** (e.g. compute_student_gpa, update_inventory)

Categories of Data Models

- **Conceptual (high-level, semantic) data models:**
  - Provide concepts that are close to the way many users perceive data.
    - (Also called **entity-based** or **object-based** data models.)
- **Physical (low-level, internal) data models:**
  - Provide concepts that describe details of how data is stored in the computer. These are usually specified in an ad-hoc manner through DBMS design and administration manuals
- **Implementation (representational) data models:**
  - Provide concepts that fall between the above two, used by many commercial DBMS implementations (e.g. relational data models used in many commercial systems).
- **Self-Describing Data Models:**
  - Combine the description of data with the data values. Examples include XML, key-value stores and some NOSQL systems.

Schemas Versus Instances (1 of 2)

- Database Schema:
  - The **description** of a database.
  - Includes descriptions of the database structure, data types, and the constraints on the database.
- Schema Diagram:
  - An **illustrative** display of (most aspects of) a database schema.
- Schema Construct:
  - A **component** of the schema or an object within the schema, e.g., STUDENT, COURSE.

Schemas Versus Instances (2 of 2)

- Database State:
  - The actual data stored in a database at a **particular moment in time**. This includes the collection of all the data in the database.
  - Also called database instance (or occurrence or snapshot).
    - The term **instance** is also applied to individual database components, e.g. **record instance, table instance, entity instance**

Database Schema vs Database State (1 of 2)

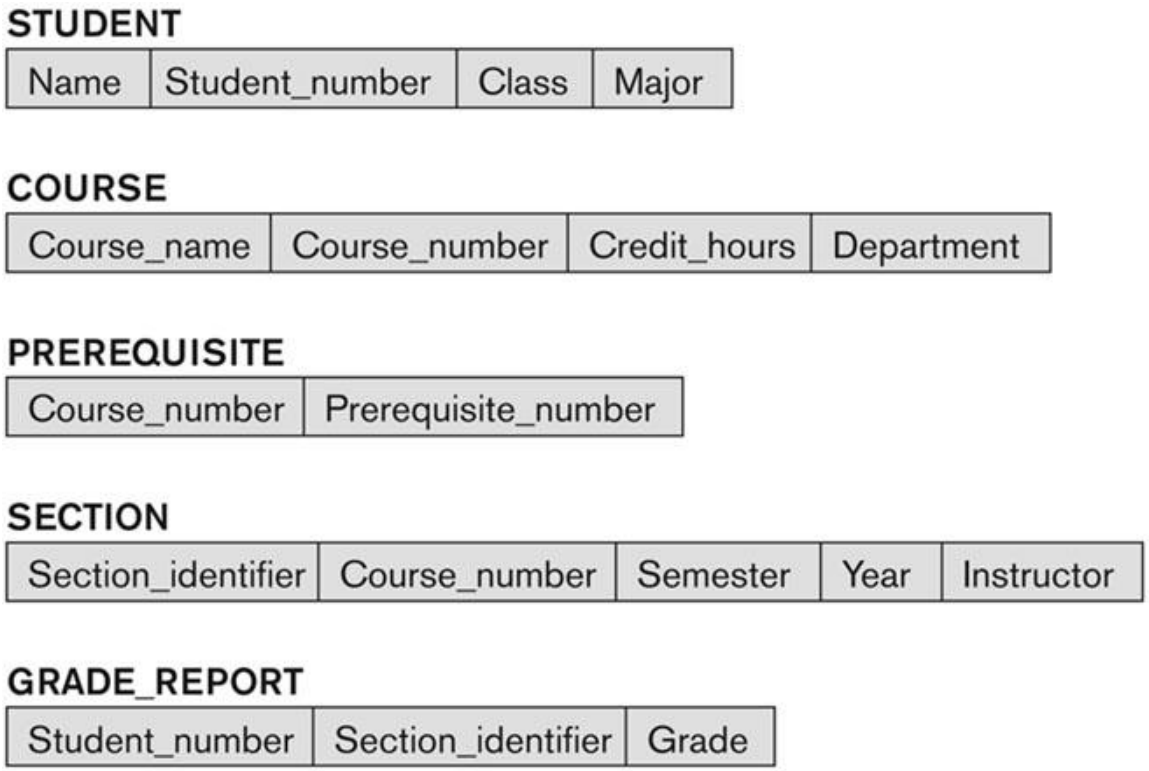
- Database State:
  - Refers to the **content** of a database at a moment in time.
- Initial Database State:
  - Refers to the database state when it is initially loaded into the system.
- Valid State:
  - A state that satisfies the structure and constraints of the database.

Database Schema vs Database State (2 of 2)

- Distinction
  - The **database schema** changes very infrequently.
  - The **database state** changes every time the database is updated.
- **Schema** is also called **intension**.
- **State** is also called **extension**.

Example of a Database Schema

Figure 2.1 Schema diagram for the database in Figure 1.2.



Example of a Database State (1 of 4)

Figure 1.2 A database that stores student and course information.

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

Example of a Database State (2 of 4)

Figure 1.2 A database that stores student and course information.

SECTION

Section_identifier	Course_number	Semester	Year	Instructor
85	MATH2410	Fall	04	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	05	Knuth
112	MATH2410	Fall	05	Chang
119	CS1310	Fall	05	Anderson
135	CS3380	Fall	05	Stone

Example of a Database State (3 of 4)

Figure 1.2 A database that stores student and course information.

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

Example of a Database State (4 of 4)

Figure 1.2 A database that stores student and course information.

PREREQUISITE

Course_number	Prerequisite_number
CS3380	CS3320
CS3380	MATH2410
CS3320	CS1310

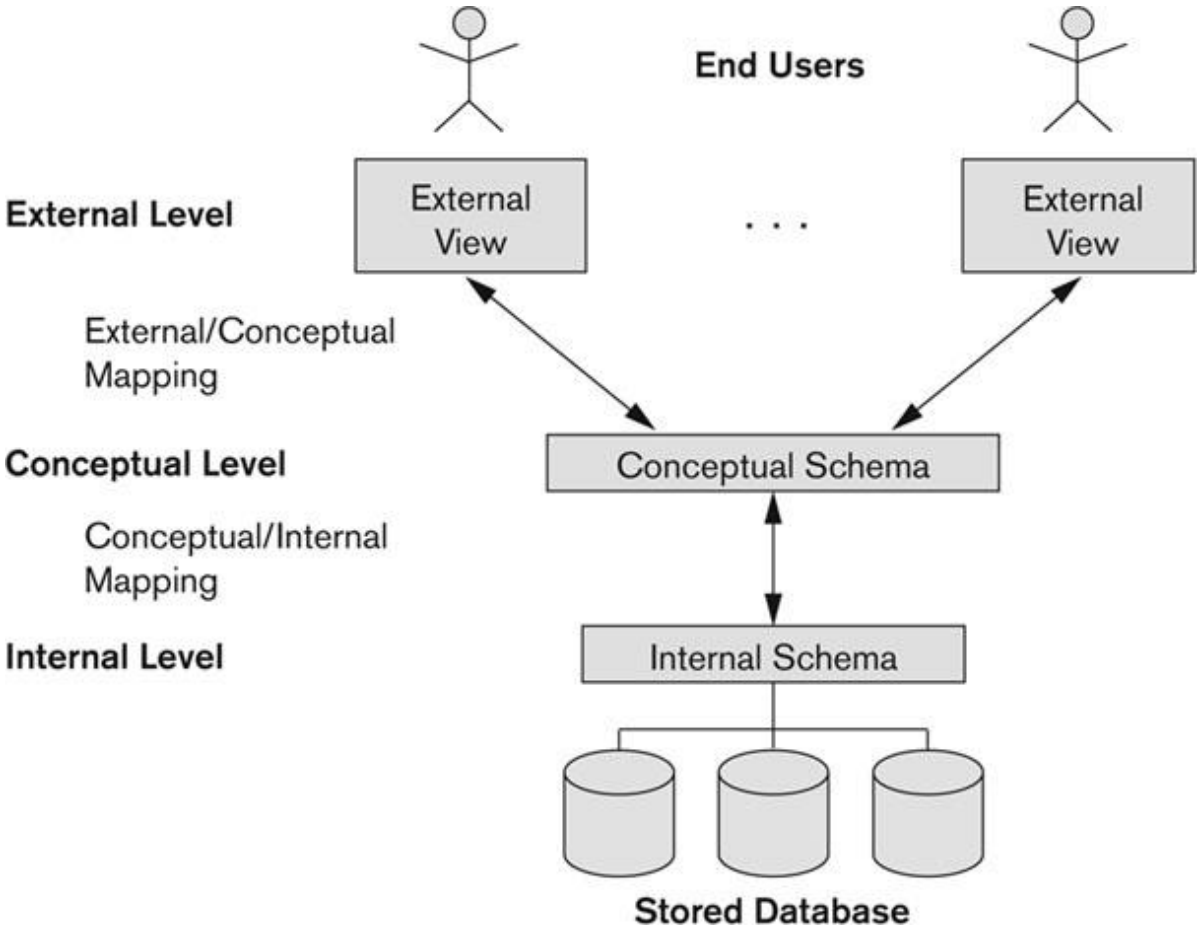
Three-Schema Architecture (1 of 2)

- Proposed to support DBMS characteristics of:
  - **Program-data independence**.
  - Support of **multiple views** of the data.
- Not explicitly used in commercial DBMS products, but has been useful in explaining database system organization

Three-Schema Architecture (2 of 2)

- Defines DBMS schemas at **three** levels:
  - **Internal schema** at the internal level to describe physical storage structures and access paths (e.g indexes).
    - Typically uses a **physical** data model.
  - **Conceptual schema** at the conceptual level to describe the structure and constraints for the whole database for a community of users.
    - Uses a **conceptual** or an **implementation** data model.
  - **External schemas** at the external level to describe the various user views.
    - Usually uses the same data model as the conceptual schema.

Figure 2.2 The Three-Schema Architecture



DBMS Languages

- Data Definition Language (DDL)
  - Define the conceptual schema
- Data Manipulation Language (DML)
  - Retrievals and updates
  - Manipulation and querying the state
  - High-level or non-procedural
    - Include relational language SQL
    - May be standalone or embedded
  - Low-level or procedural languages
    - Must be embedded in a programming language

DBMS Interfaces

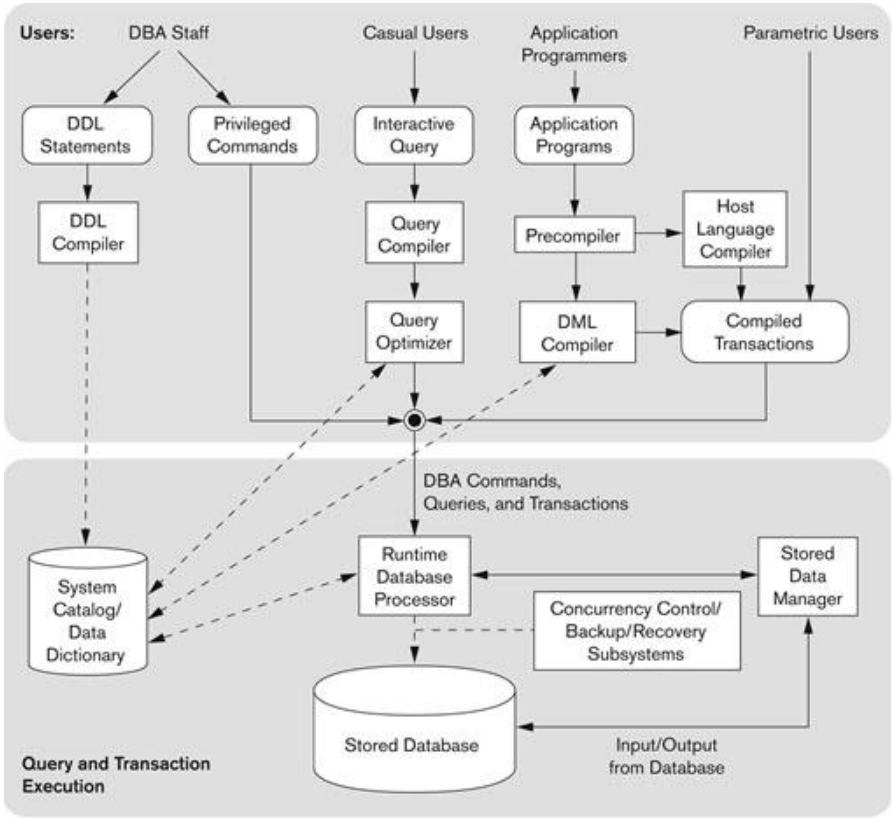
- Stand-alone query language interfaces
  - Example: Entering SQL queries at the DBMS interactive SQL interface (e.g. SQL *Plus in ORACLE)
- Programmer interfaces for embedding DML in programming languages
- User-friendly interfaces
  - Menu-based, forms-based, graphics-based, etc.
- Mobile Interfaces: interfaces allowing users to perform transactions using mobile apps

Other DBMS Interfaces

- Natural language: free text as a query
- Speech: Input query and output response
- Web browser with keyword search
- Parametric interfaces, e.g., bank tellers using function keys.
- Interfaces for the DBA:
  - Creating user accounts, granting authorizations
  - Setting system parameters
  - Changing schemas or access paths

Typical DBMS Component Modules

Figure 2.3 Component modules of a DBMS and their interactions.



Topic Summary

- Data models and model categories
- Schemas, instances, and states
- Three-schema architecture
- DBMS languages and interfaces

Copyright