Database System Concepts and Architecture

Lecture 2

Outline

- Data Models
- Categories of Data Models
- Database Schemas, Instances, and Database State
- DBMS Languages and Interfaces

Data Models

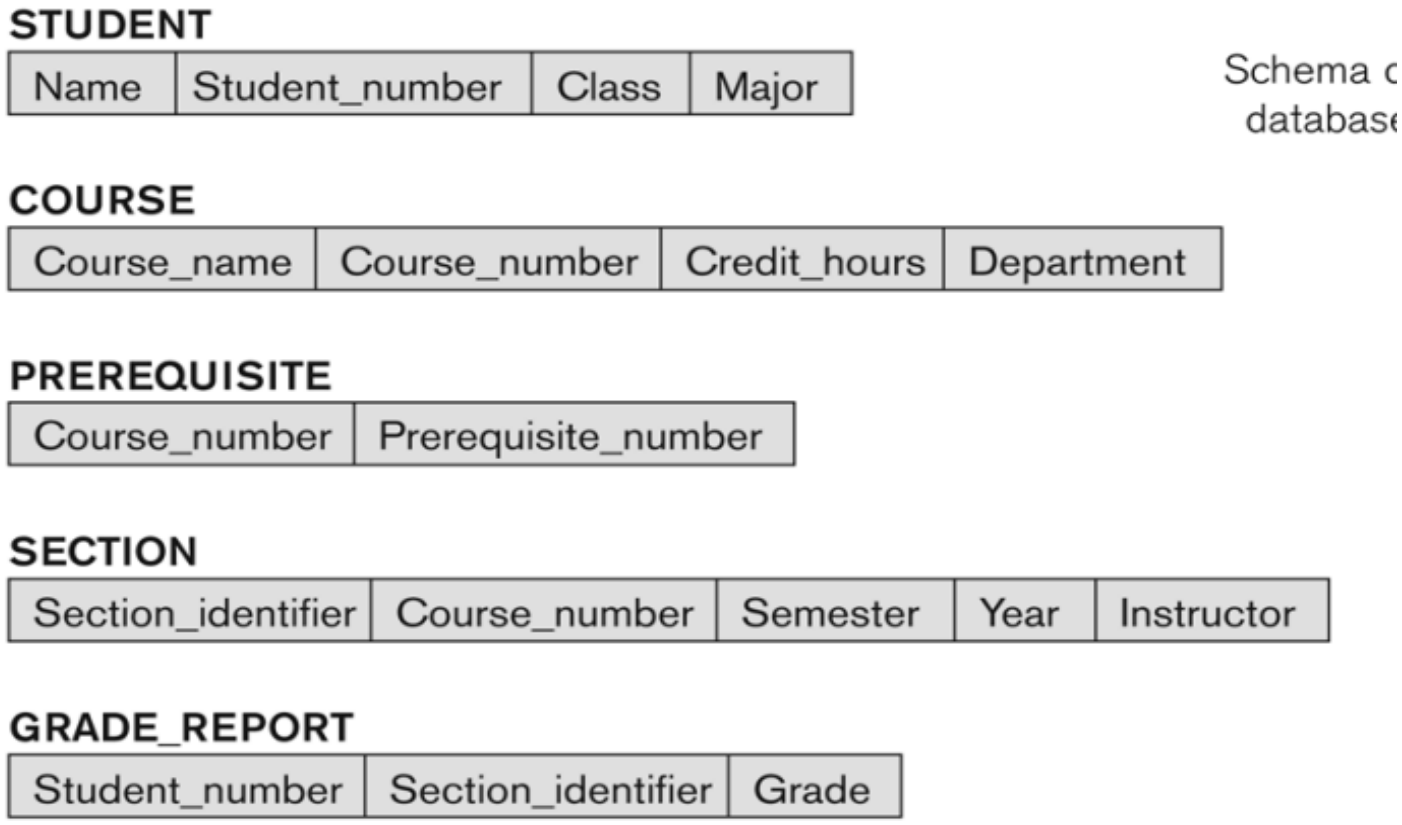
- Data models: a collection of concepts that can be used to describe the structure of a database—provides the necessary means to achieve this abstraction.
- They define how the logical structure of a database is modeled (e.g., the data types, relationships, and constraints that apply to the data).
- They define how data is connected to each other and how they are processed and stored inside the system.

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).

Database Schema

- **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.



Database State

- **Database State:** refers to the content of a database at a moment in time.
- Also called the current set of occurrences or instances in the database.
  - **Initial Database State:** refers to the database state when it is initially loaded into the system.

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.

Database Schema vs. Database State

- 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***.

DBMS Languages

- Data Definition Language (DDL)
  - Used by the DBA and database designers to specify and define the conceptual schema of a database.
- Data Manipulation Language (DML)
  - Used to specify database retrievals and updates
  - High-Level or Non-procedural Languages: these include the relational language SQL
    - May be used in a standalone way or may be embedded in a programming language (java or c++)
- View Definition Language (VDL)
  - Specifies user views/mappings to conceptual schemas.

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
  - Procedure Call Approach: e.g. JDBC for Java, ODBC (Open Database Connectivity) for other programming languages as API's (application programming interfaces)
  - Scripting Languages: PHP (client-side scripting) and Python (server-side scripting) are used to write database programs.
- User-friendly interfaces
  - Web-based interfaces: Menu-based, forms-based, graphics-based, etc.
- Mobile Interfaces
  - interfaces allowing users to perform transactions using mobile apps