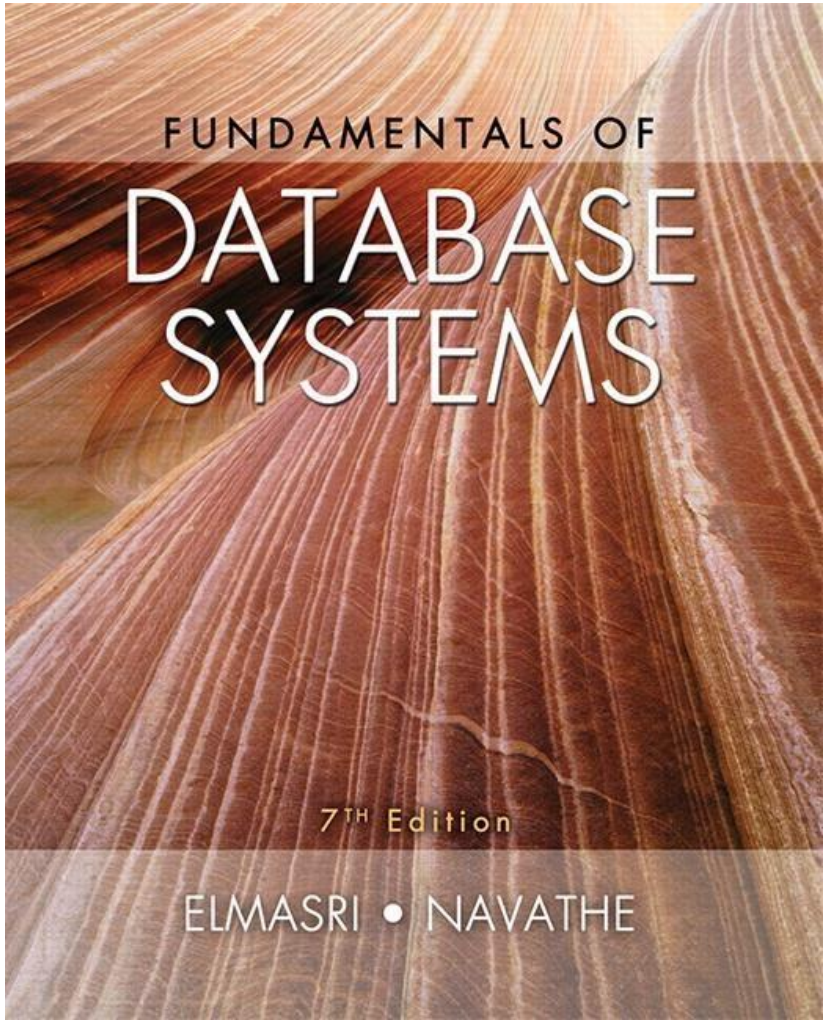

FUNDAMENTALS OF
DATABASE SYSTEMS
7TH Edition
ELMASRI • NAVATHE

UNIT 01

Introduction to Database Systems

Reviewed and Edited by Prof. Suleiman H. Mustafa

OUTLINE

- Basic Definitions and Types of Databases, Database Applications, and database system
- DBMS Functionality
- Example of a Database (UNIVERSITY)
- Main Characteristics and advantages of the Database Approach
- Database Users
- Data Models, Schemas, Instances, and States
- DBMS Languages
- Database System Utilities and Tools

Basic Definitions and Concepts

Database Concepts

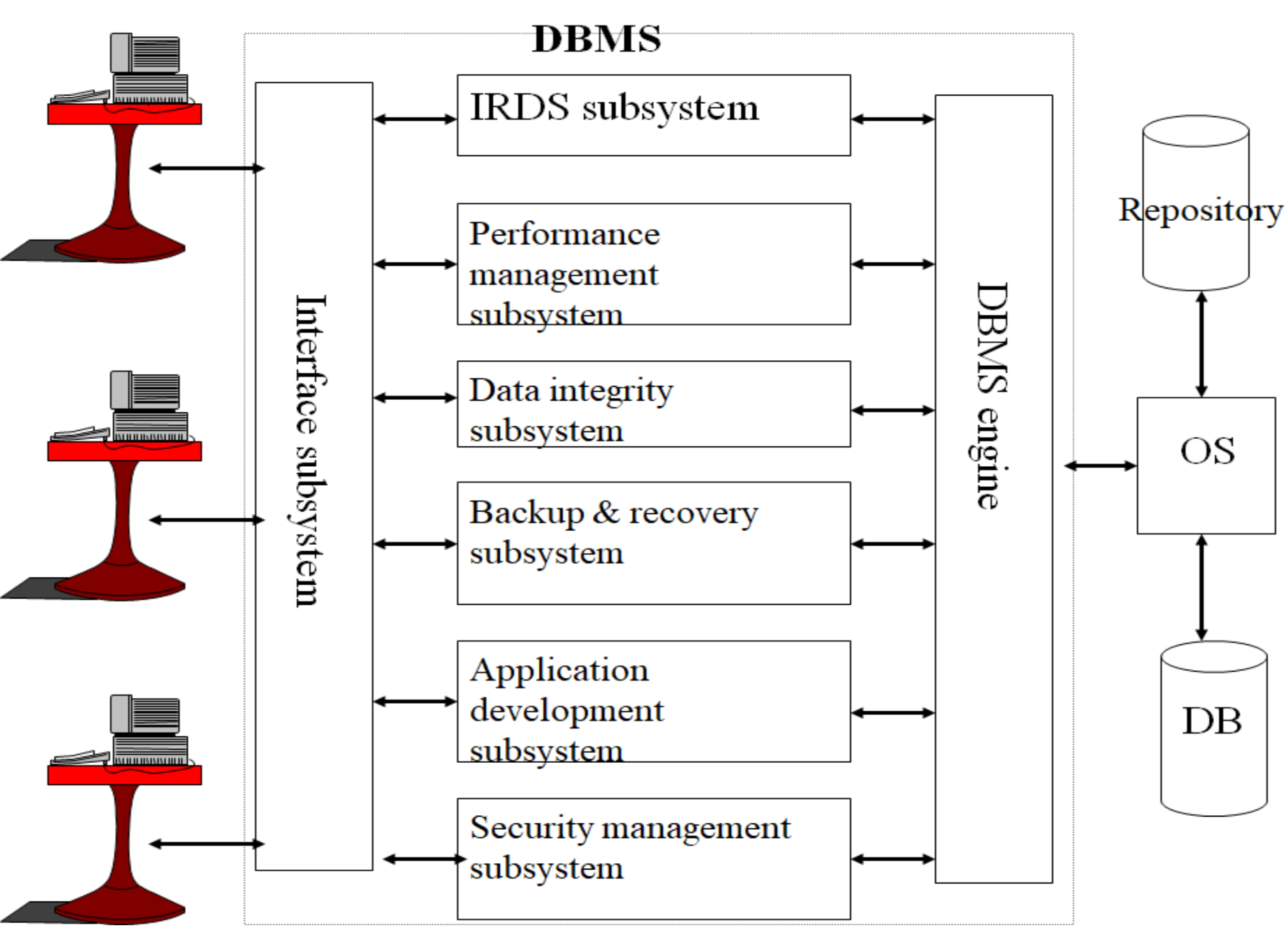
- **Data**
  - Known facts that can be recorded and have an implicit meaning.
- **Database**
  - A collection of organized related data to serve a given purpose.
- **Types of Databases**
  - Numerical Databases
  - Textual Databases
  - Image databases
  - Multimedia Databases
- Databases are used in most human activities and fields of application
  - Examples: Banking, Insurance, Transportation, Healthcare, Retail, Manufacturing, Education, GIS, etc.

Database Concepts

- Data Warehouse
  - A data warehouse is a type of data management system that is designed to
    - enable and support business intelligence (BI) activities, especially data analysis (analytics).
  - Data warehouses are solely intended to
    - perform queries and analysis and
    - often contain large amounts of historical data.

Database Management System (DBMS)

- A computerized database is created and managed using a special software called Database Management System (DBMS)
- **A Database Management System (DBMS) is**
  - A software package/ system to facilitate the creation, access, and maintenance of a database.
- **DBMS Components (see figure next slide)**
  - The DBMS consists of several subsystems each of which is responsible for a given function.
  - The DBMS interacts with the operating system to manage and access data.


DBMS
IRDS subsystem
Performance management subsystem
Data integrity subsystem
Backup & recovery subsystem
Application development subsystem
Security management subsystem
Interface subsystem
DBMS engine
Repository
OS
DB

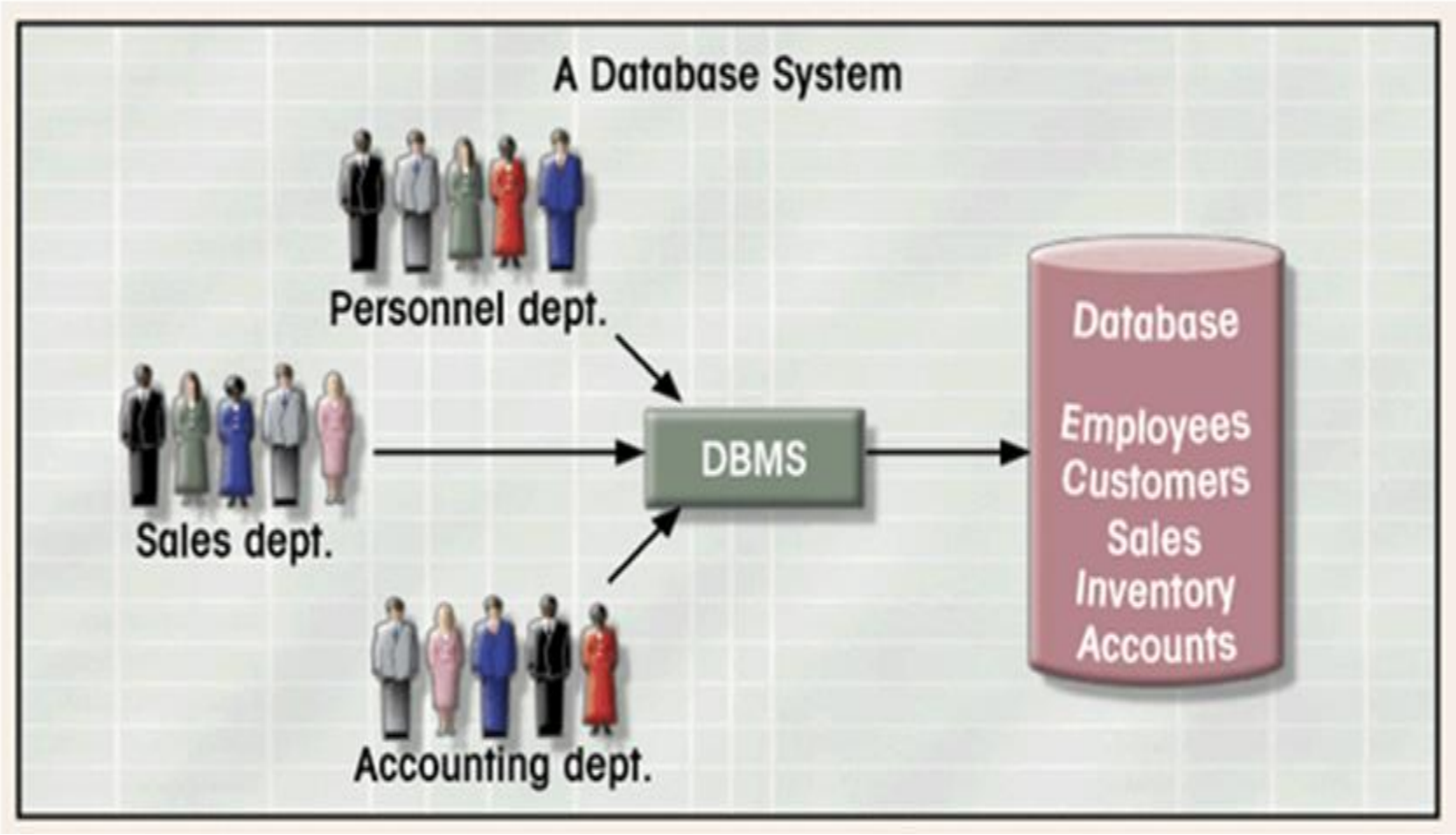
Functions of DBMS

1. *Defining* a particular database in terms of its data types, structures, and constraints
2. *Constructing* or Loading the initial database contents on a secondary storage device.
3. *Manipulating* the database:
   - **Retrieval**: Querying, generating reports, and accessing the database through Web applications
   - **Update**: Insertions, deletions and modifications to its content
4. *Processing* and *Sharing* by a set of concurrent users and application programs (yet, keeping all data valid and consistent)

Functions of DBMS

5. Protecting or providing security measures to prevent unauthorized access
6. Presentation and Visualization of data
7. System and software Maintenance of the database and associated programs over the lifetime of the database application.

Users Interaction with Databases through their Applications

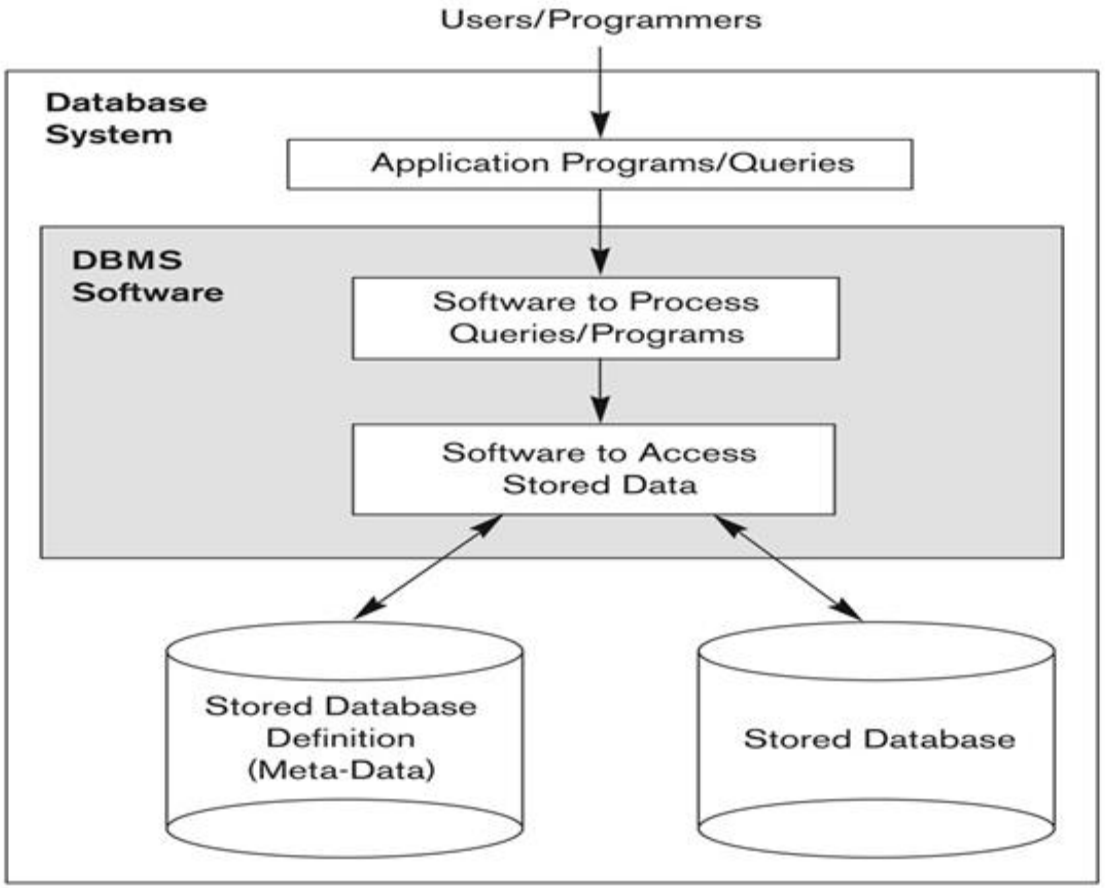


Database Applications

- Database applications are:
  - programs developed by database users to utilize the data stored in a given database
- Applications interact with a database by generating
  - Queries (retrieval): that access different parts of data and formulate the result of a request.
  - Update Transactions (insert, delete, modify): that may read some data and "update" certain values or generate new data and store that in the database
- Applications must not allow unauthorized users to access data and must keep up with changing user requirements against the database

Database System

- **A Database System consists of** the DBMS software together with the data itself and the database applications.



A simplified database system environment.

Example of a Database (with a Conceptual Data Model)

Example of a Database (with a Conceptual Data Model)

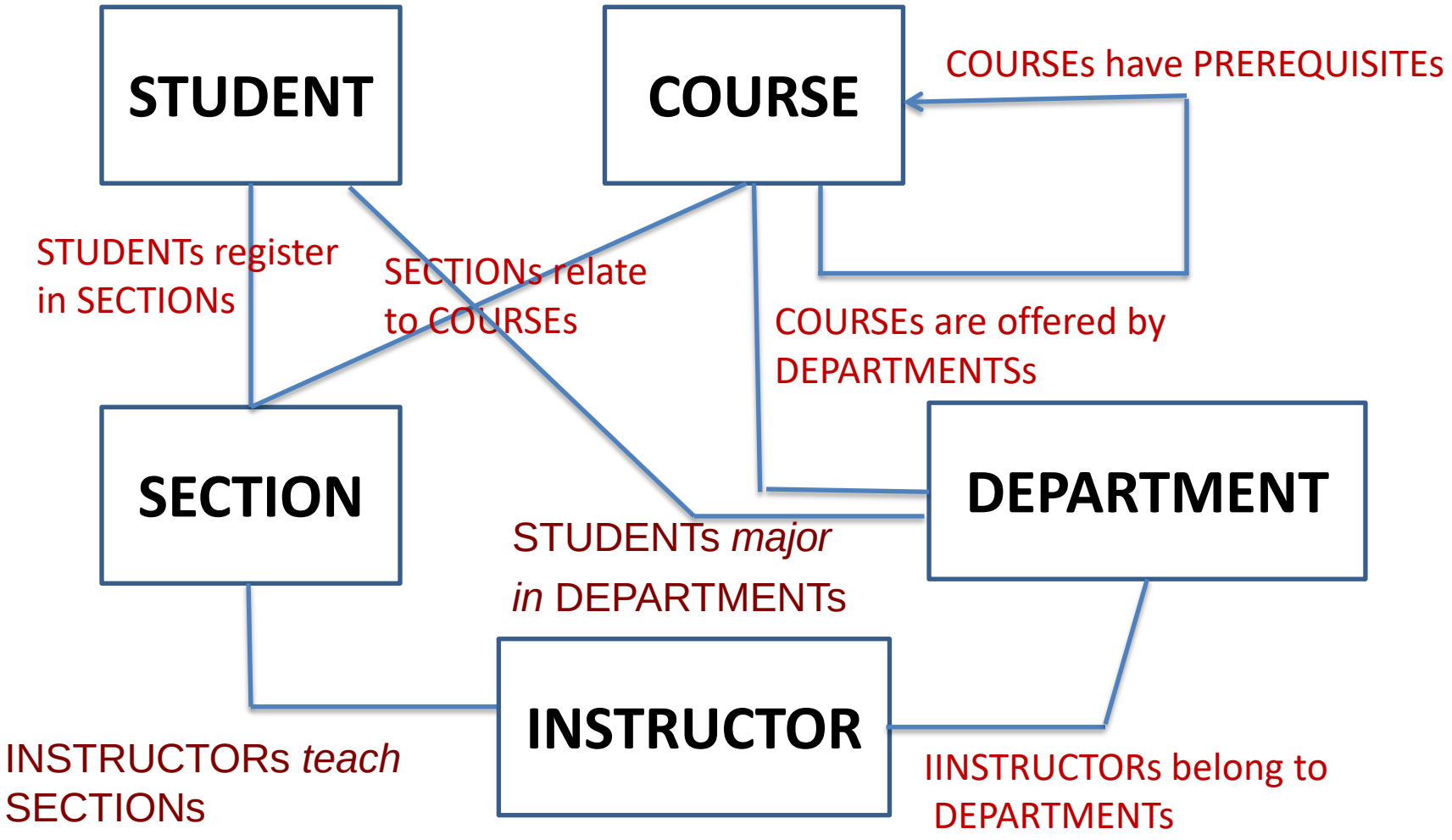
- **Application area of the example:**
  - Part of a UNIVERSITY environment.
- **Some *entities in the example*:**
  - STUDENTs
  - COURSEs
  - SECTIONs (of COURSEs)
  - (academic) DEPARTMENTs
  - INSTRUCTORs

Example of a Database (with a Conceptual Data Model)

- **Some *relationships between entities*:**
  - SECTIONs *are of specific* COURSEs
  - STUDENTs *take* SECTIONs
  - COURSEs *have prerequisite* COURSEs
  - INSTRUCTORs *teach* SECTIONs
  - COURSEs *are offered by* DEPARTMENTs
  - STUDENTs *major in* DEPARTMENTs

- **Note:** The above entities and relationships are typically expressed in a conceptual data model (see next slide), such as the ENTITY-RELATIONSHIP data model (discussed later)

Conceptual View of Entities and Relationships



Example of a simple database

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

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	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	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.

Example of Queries and Updates

- **Examples of Queries:**
  - Retrieve the transcript.
  - List the names of students who took the section of the 'Database' course offered in fall 2008 and their grades in that section.
  - List the prerequisites of the 'Database' course
- **Examples of Update Transactions:**
  - Change the class of 'Smith' to sophomore.
  - Create a new section for the 'Database' course for this semester.
  - Enter a grade of 'A' for 'Smith' in the 'Database' section of last semester

Characteristics and Advantages of the Database Approach

Main Characteristics of the Database Approach for Storing Data

1. Self-describing nature of a database system:

- A DBMS **catalog (or repository or data dictionary)** stores the description of a particular database (e.g. schema descriptions, data structures, types, constraints, design decisions, application program descriptions, user information, etc.)
- This description is called **meta-data** (that is data about data). This allows the DBMS software to work with different database applications.
- S**ee next slide**:

Data Dictionary

- The information stored in the DD include:
  - schema descriptions (see next slide) and
  - other information such as
    - design decisions,
    - application program descriptions,
    - user information,
    - usage standards, etc.

Example of a Simplified Database Catalog (Data Dictionary)

RELATIONS

Relation_name	No_of_columns
STUDENT	4
COURSE	4
SECTION	5
GRADE_REPORT	3
PREREQUISITE	2

COLUMNS

Column_name	Data_type	Belongs_to_relation
Name	Character (30)	STUDENT
Student_number	Character (4)	STUDENT
Class	Integer (1)	STUDENT
Major	Major_type	STUDENT
Course_name	Character (10)	COURSE
Course_number	XXXXNNNN	COURSE
….	….	…..
….	….	…..
….	….	…..
Prerequisite_number	XXXXNNNN	PREREQUISITE

Note: Major_type is defined as an enumerared type with all known majors. XXXXNNNN is used to define a type with four alpha characters followed by four digits

Figure 1.3
An example of a database catalog for the database in Figure 1.2.

Meta-Data Example

Field Name	Data Type	Data Format	Field Size	Description	Example
License ID	Integer	NNNNNN	6	Unique number ID for all drivers	12345
Surname	Text		20	Surname for Driver	Jones
First Name	Text		20	First Name for Driver	Arnold
Address	Text		50	First Name for Driver	11 Rocky st Como 2233
Phone No.	Text		10	License holders contact number	0400111222
D.O.B	Date / Time	DD/MM/YYYY	10	Drivers Date of Birth	08/05/1956

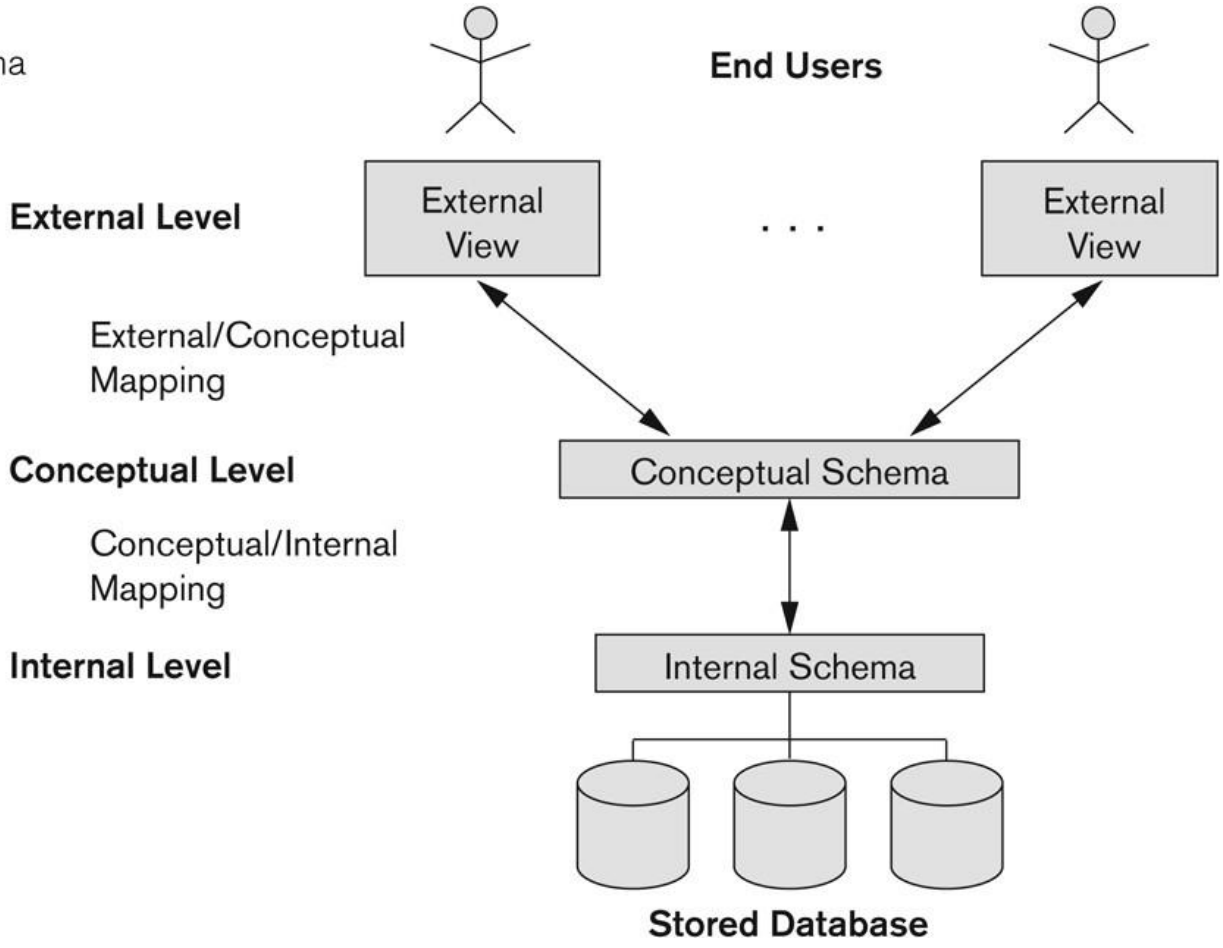
Main Characteristics of the Database Approach for Storing Data

2. Insulation between programs and data:

- This is called **program-data independence**.
- Allows changing data structures and storage organization without having to change the DBMS access programs.
- Logical data independence:
  - The capacity to change the conceptual or logical schemas without having to change the requirements representations and their associated application programs.
- Physical data independence:
  - The capacity to change the physical schema without having to change the conceptual or logical schema. (e.g., adding an index to the physical structure).

The Three-Levels Mapping

Figure 2.2
The three-schema architecture.



Main Characteristics of the Database Approach

3. Data Abstraction:

- A **data model** is used to hide storage details (see next slide) and present the users with a conceptual view of the database.
- Programs refer to the data model constructs rather than data storage details

4. Support of multiple views of the data:

- Each user may see a different view of the database, which describes **only** the data of interest to that user.

Internal Storage Format

Data Item Name	Starting Position in Record	Length in Characters (bytes)
Name	1	30
Student_number	31	4
Class	35	1
Major	36	4

Figure 1.4
Internal storage format for a STUDENT record, based on the database catalog in Figure 1.3.

Main Characteristics of the Database Approach (continued)

5. Sharing of data and multi-user transaction processing:

- Allowing a set of **concurrent users** to retrieve from and to update the database.
- *Concurrency control* within the DBMS guarantees that each **transaction** is correctly executed or aborted
- *Recovery* subsystem ensures each completed transaction has its effect permanently recorded in the database
- **OLTP** (Online Transaction Processing) is a major part of database applications. This allows hundreds of concurrent transactions to execute per second.

Advantages of Using the Database Approach

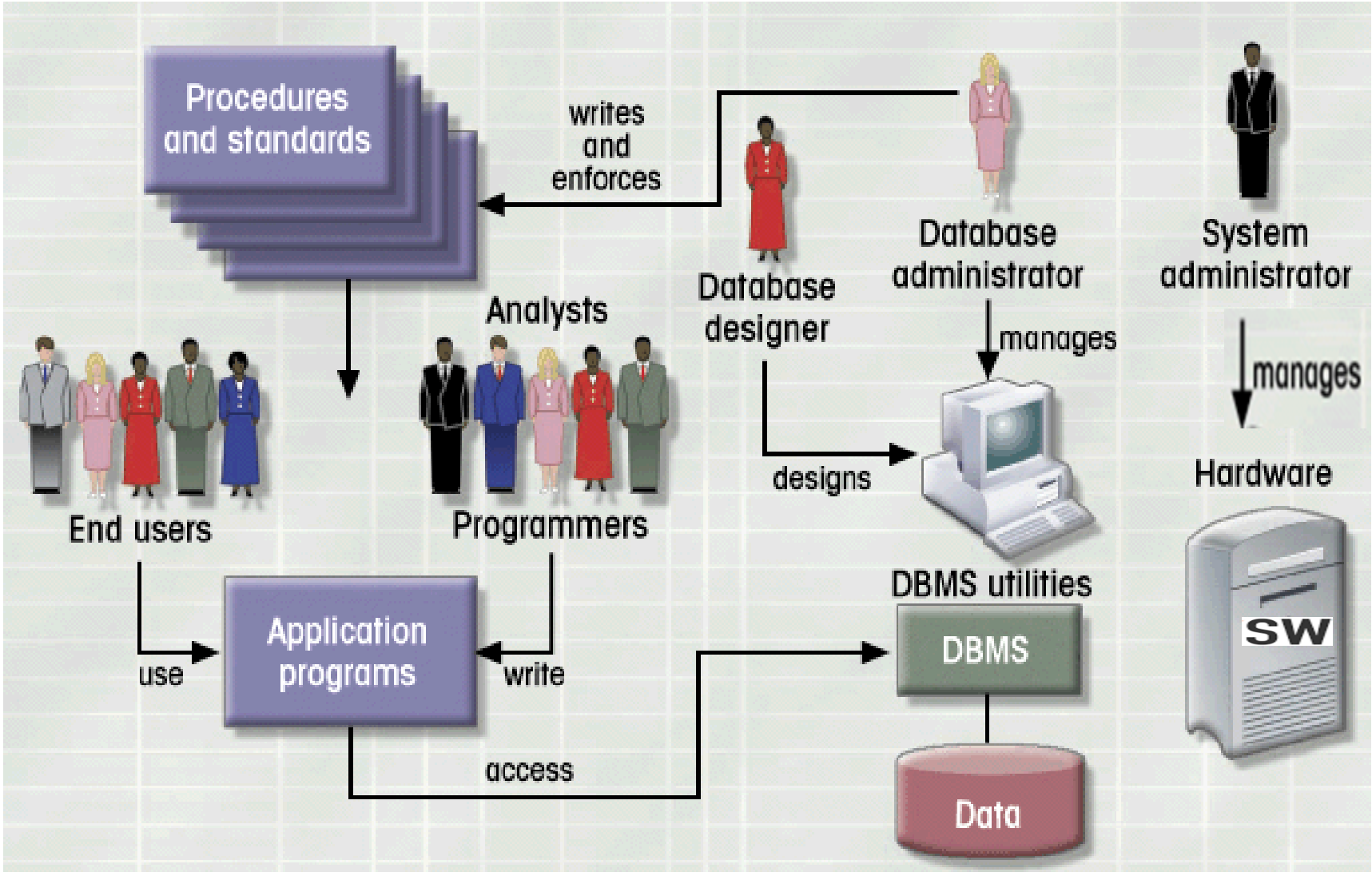
- Controlling redundancy in data storage and in development and maintenance efforts.
- Representing complex relationships among data and providing storage structures (e.g. indexes) for efficient query processing.
- Providing backup and recovery services.
- Restricting unauthorized access to data
- Sharing of data among multiple users
- Optimizing queries for efficient processing.
- Enforcing integrity constraints on the database.
- Providing multiple interfaces to different classes of users.

Additional Implications of Using the Database Approach

- Potential for enforcing standards
- Reduced application development time
- Economies of scale across applications and departments
- Flexibility to change data structures
- Availability of current information

Database Users

Database System Environment and Users



Database Users

- **Database Administrators (DBA):** Responsible for
  - authorizing access to the database,
  - Coordinating, Controlling, and monitoring its use,
  - acquiring software and hardware resources,
  - monitoring efficiency of database operations.
- **Database Analysts and Designers:**
  - Responsible for defining the content, the structure, the constraints, and functions or transactions against the database.
  - They must communicate with the end-users and understand their needs.

Database Users

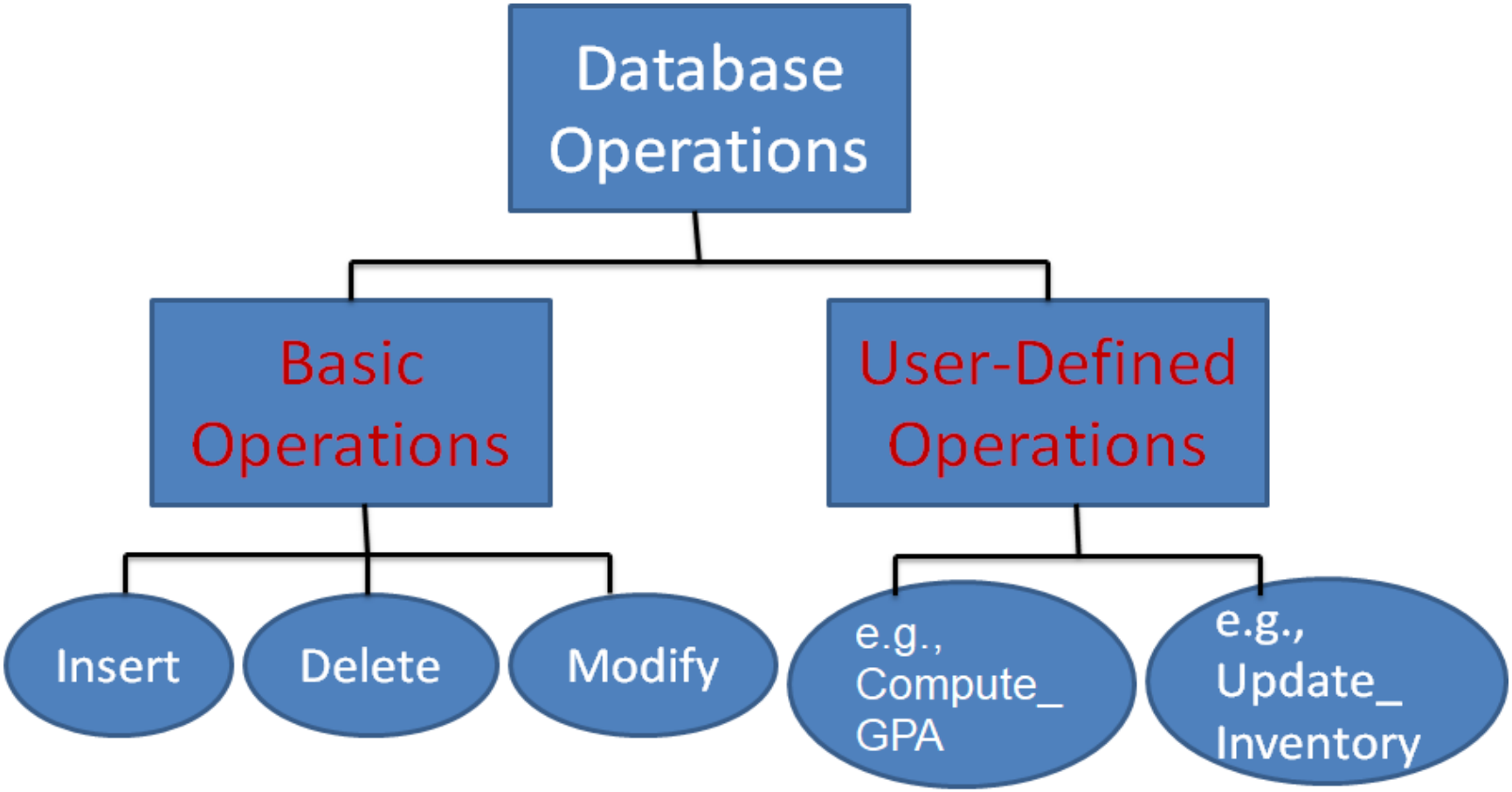
- **Database Application Developers:**
  - This group includes: system analysts, application programmers, and business analysts.
  - They write application programs.
- **Database End-users:**
  - They use the data for queries, reports and some of them update the database content.
- **System Administrator, Operators and Maintenance Personnel:**
  - They manage the actual running and maintenance of the database system hardware and software environment.

Data Models and Database Schemas

Data Models

- **Data Model:**
  - A set of concepts to describe
    - the ***structure*** of a database (its elements and relationships)
    - the ***operations*** for manipulating these structures, and
    - certain ***constraints*** that the database should obey.
  - **Constraints** specify some restrictions on valid data; these constraints must be enforced at all times.
  - Data model **operations** are used for
    - specifying database *retrievals* and
    - specifying *updates (insert, delete, modify)*
    - *See next slide*

Data Models



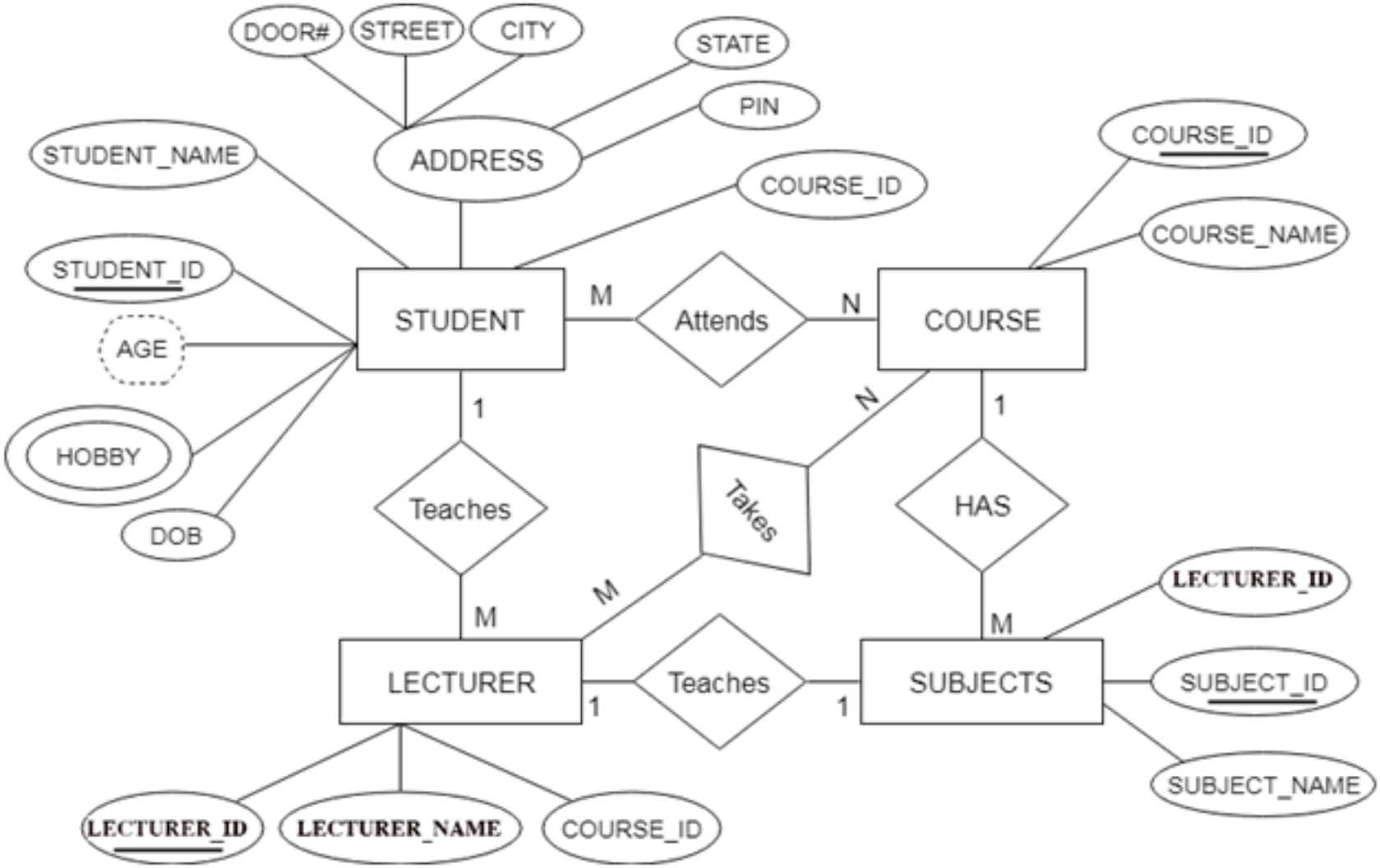
Database Schemas

- 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.

Categories of Data Models

- **Conceptual (High-level) data models:**
  - Provide semantic concepts that are close to the way many users perceive data.
  - e.g. Entity-relationship model (see figure)
- **Logical (Intermediate-level) data models:**
  - Provide concepts for representing data in the database.
  - **e.g**. relational data model used in many commercial database management systems. (see figure)
- **Physical (low-level) data models:**
  - Provide concepts that describe details of how data is stored in the computer.
  - **e.g**. DBDL (database definition language) (see figure)

Conceptual Data Model using ER (conceptual schema example)



Logical Data Modeling using the Relational Model (Logical database schema)

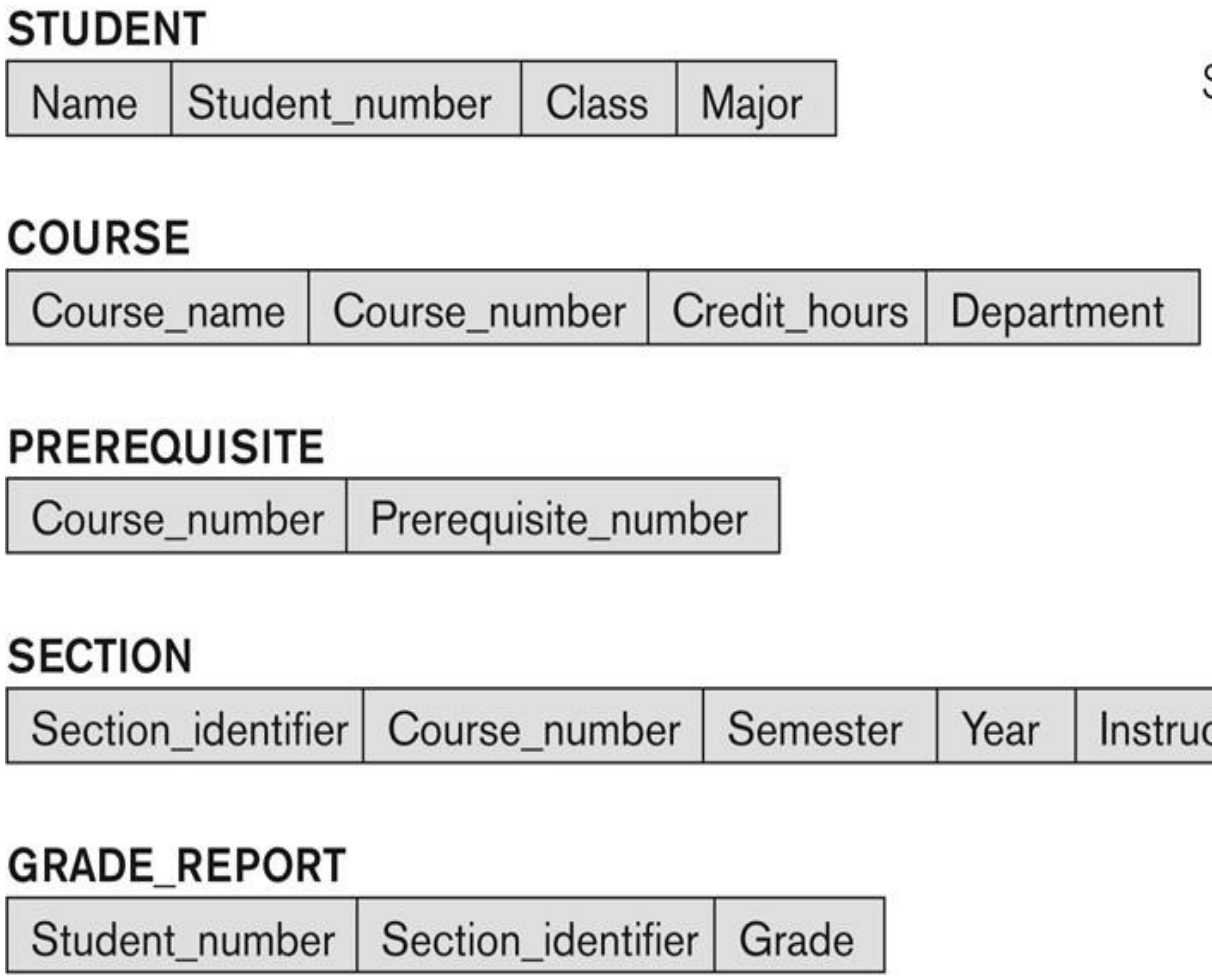


Figure 2.1
Schema diagram for the database in Figure 1.2.

Physical Data Model using DBDL (Physical Database Schema)

```
domain Branch_Numbers       fixed length character string length 4
domain Street_Names         variable length character string maximum length 30
domain City_Names           variable length character string maximum length 20
domain State_Codes          fixed length character string length 2
domain Zip_Codes            fixed length character string length 5
domain Staff_Numbers        fixed length character string length 5

Branch(  branchNo      Branch_Numbers      NOT NULL,
         street        Street_Names        NOT NULL,
         city          City_Names          NOT NULL,
         state         State_Names         NOT NULL,
         zipCode       Zip_Codes           NOT NULL,
         mgrStaffNo    Staff_Numbers       NOT NULL)
         Primary Key branchNo
         Alternate Key zipCode
         Foreign Key mgrStaffNo References Staff(staffNo) ON UPDATE CASCADE ON DELETE NO ACTION
```

Database State

Database State

- Database State:
  - The actual data (i.e., the collection of all the data) stored in a database at a ***particular moment in time***.
  - 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 State

- 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 State Vs. Database Schema
  - The ***database schema*** changes very infrequently.
  - The **database state** changes every time the database is updated.

Example of a Database State (Instance)

STUDENT

Name	Student_number	Class	Major
Smith	17	1	CS
Brown	8	2	CS

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	07	King
92	CS1310	Fall	07	Anderson
102	CS3320	Spring	08	Knuth
112	MATH2410	Fall	08	Chang
119	CS1310	Fall	08	Anderson
135	CS3380	Fall	08	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 Design

Database Design and Applications Design

- The process of designing a database system involves two main activities:
  - Database design and
  - Applications design
- Database design focuses on the steps of constructing the database structure starting from user requirements to the physical specification of data on the storage devices.
- Applications design focuses on the programs and interfaces that access the database
  - Generally considered part of software engineering

Database System Design Phases

- The database design involves three phases that are mapped to each other in the following sequence (see next figure):
  - Conceptual design,
  - Logical design, and
  - Physical design

Overview of the Database Design Process

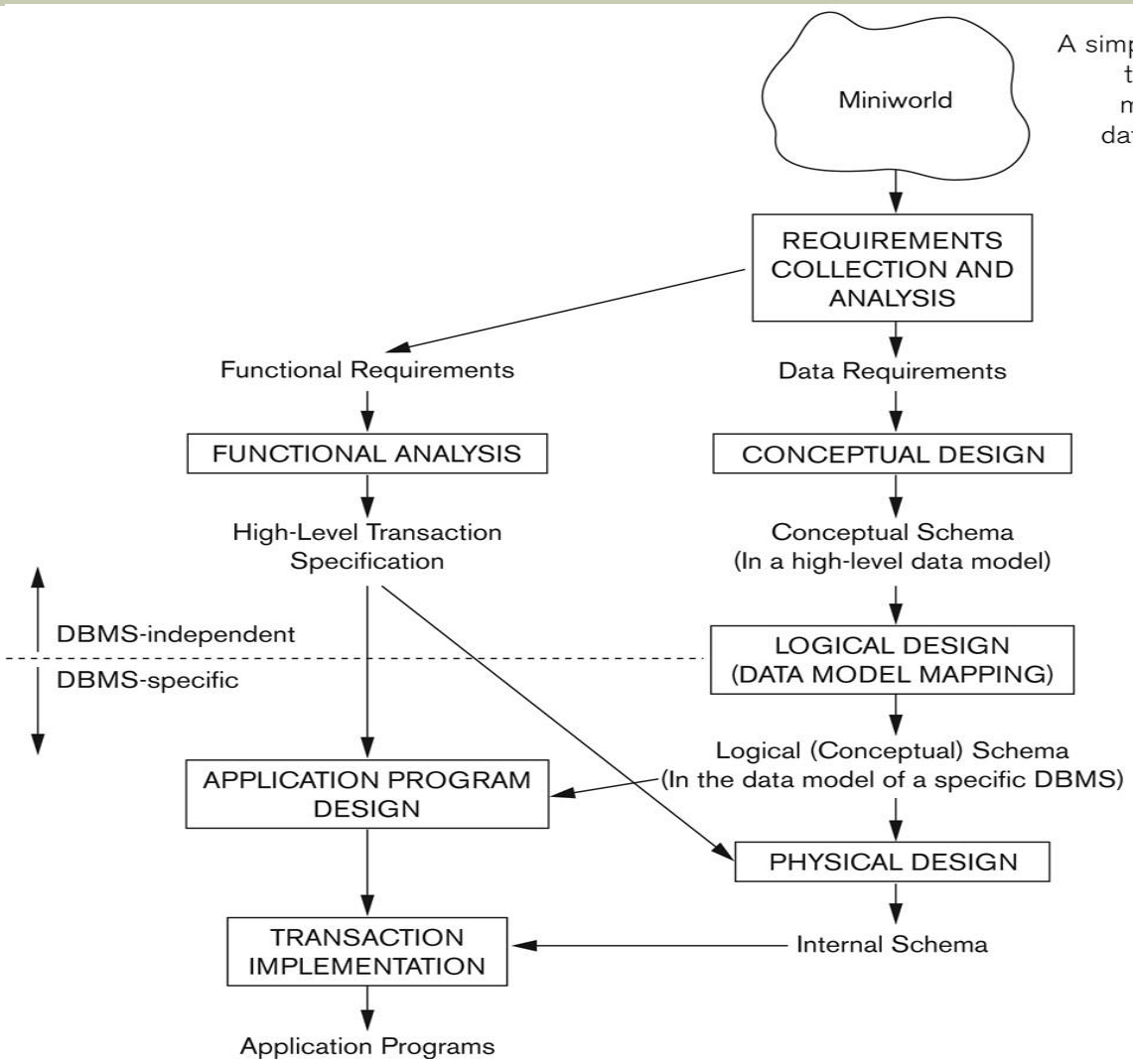
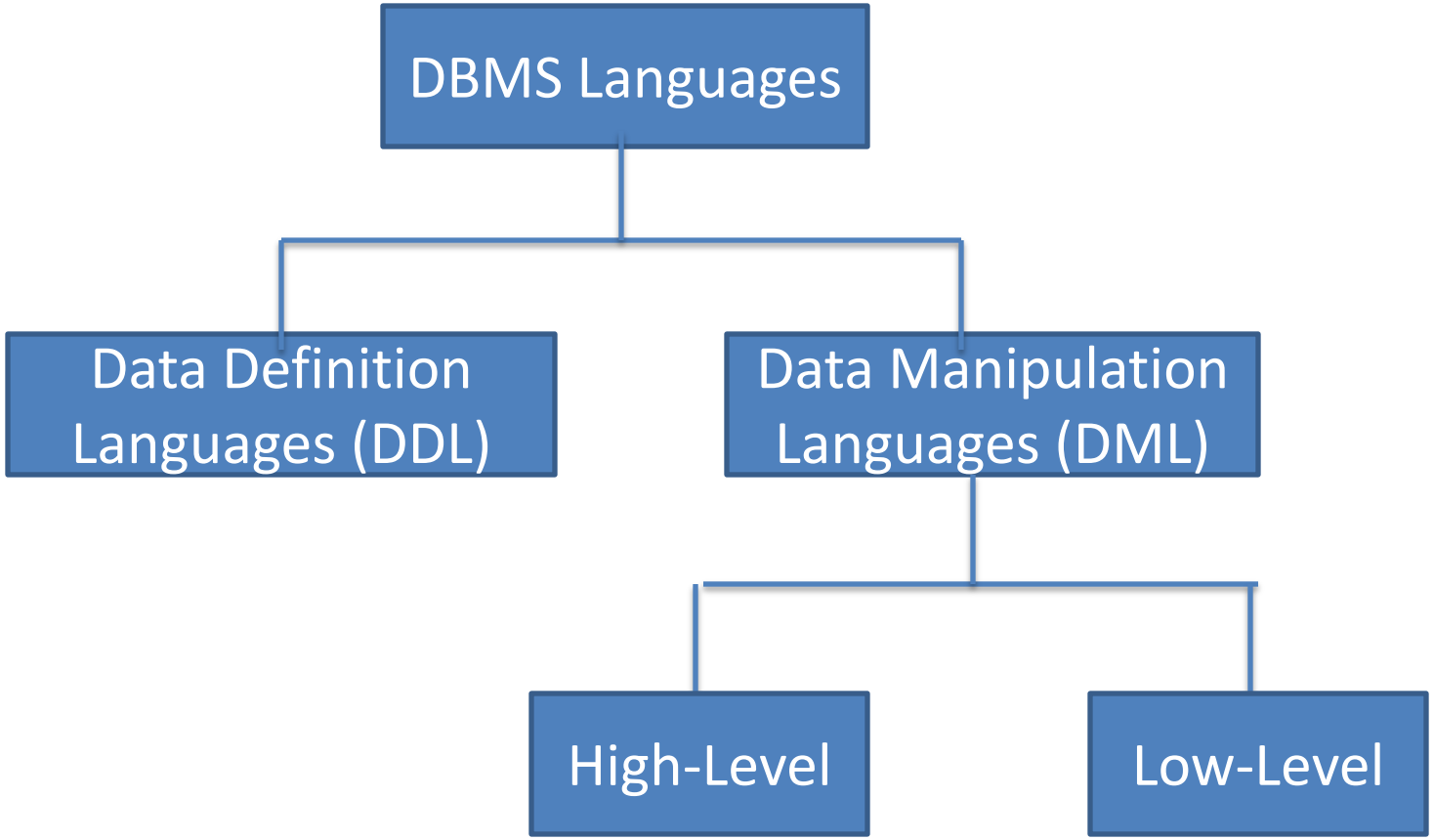


Figure 3.1
A simplified diagram to illustrate the main phases of database design.

DBMS Languages and Utilities

DBMS Languages



DBMS Languages

1. Data Definition Languages (DDL)

- Used by the DBA and database designers to specify the physical schema of a database.
- Example: CREATE TABLE (….); in SQL

2. Data Manipulation Languages (DML)

- 2.1 High-Level or Non-procedural Languages: These include the relational language SQL
  - Examples: SQL Plus, MySQL
- 2.2 Low Level or Procedural Languages:
  - These must be embedded in a programming Language (Example: PL/SQL in Oracle DBMS)

Database System Utilities

- To perform certain functions such as:
  - Loading data stored in files into a database. Including data conversion
  - Backing up the database periodically on secondary storage devices
  - Reorganizing database file structures.
  - Performance monitoring
  - Report generation
  - Other functions, such as sorting, user monitoring, data compression, … etc.

ADE and CASE Tools

- Application Development Environments (ADE) and Computer-Aided Software Engineering (CASE) tools:
- Examples:
  - PowerBuilder (Sybase)
  - JBuilder (Borland)
  - JDeveloper 10G (Oracle)

Summary

- Basic Definitions
- DBMS Functionality
- Example of a Database (UNIVERSITY)
- Main Characteristics and advantages of the Database Approach
- Types of Database Users
- Data Models, Schemas, Instances, and States
- DBMS Languages.
- Database System Utilities and Tools
- Database design phases