DSAA 5012
Advanced Data Management for Data Science

LECTURE 1
DATABASE MANAGEMENT SYSTEMS

DATABASE MANAGEMENT SYSTEMS: OUTLINE

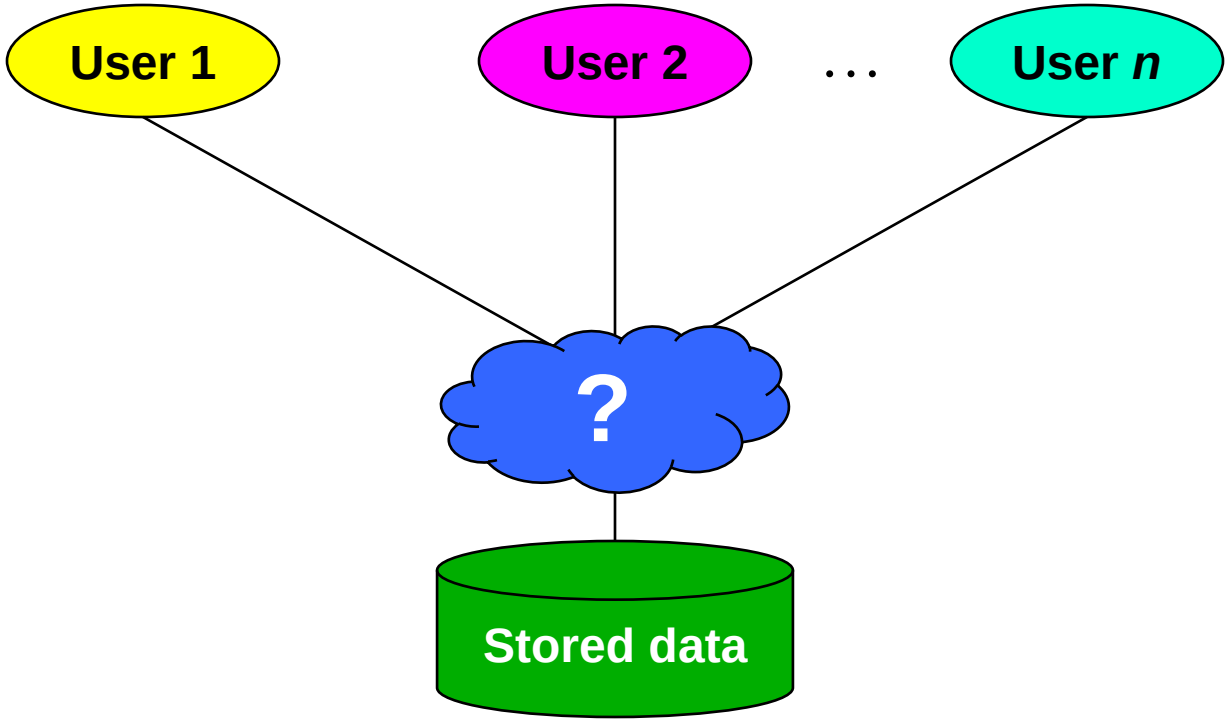
What Is A Database Management System (DBMS)?

Why Do We Need Database Management Systems?

How Does A DBMS Manage Data?

What Are The Major Components Of A DBMS?

THE PROBLEM WE WANT TO ADDRESS



How best to manage stored data?

☞ **organize, access, share, protect, … .**

DATABASE

A *database* is a collection of related data within a specific business process or problem setting.

Data are facts such as age, salary, name, address, etc.

- A database has the following properties.
  - It is designed, built and populated with data for a specific purpose.

    Applications: sales, human resources, manufacturing, banking, real estate, stock trading, inventory management, social media, ride sharing, …
  - It usually represents some aspect of the real world.
  - The data have some inherent meaning.

☞ **Databases touch all aspects of our lives!**

DATABASE MANAGEMENT SYSTEM (DBMS)

A *database management system (DBMS)* is a general-purpose software system used to manage databases.

- A DBMS provides support/facilities for:
  - defining what data to store (types, structures, constraints)
  - storing and managing data on a storage device
  - manipulating data (querying, updating)
  - sharing data among many users
  - protecting data from loss, corruption, unauthorized access

Company	*Product*
Oracle	Oracle Database
IBM	DB2
Microsoft	Access, SQL Server
Sybase	Adaptive Server
Informix	Dynamic Server

A DBMS provides an environment for managing data that is both *convenient* and *efficient* to use.

DATABASE MANAGEMENT SYSTEMS: OUTLINE

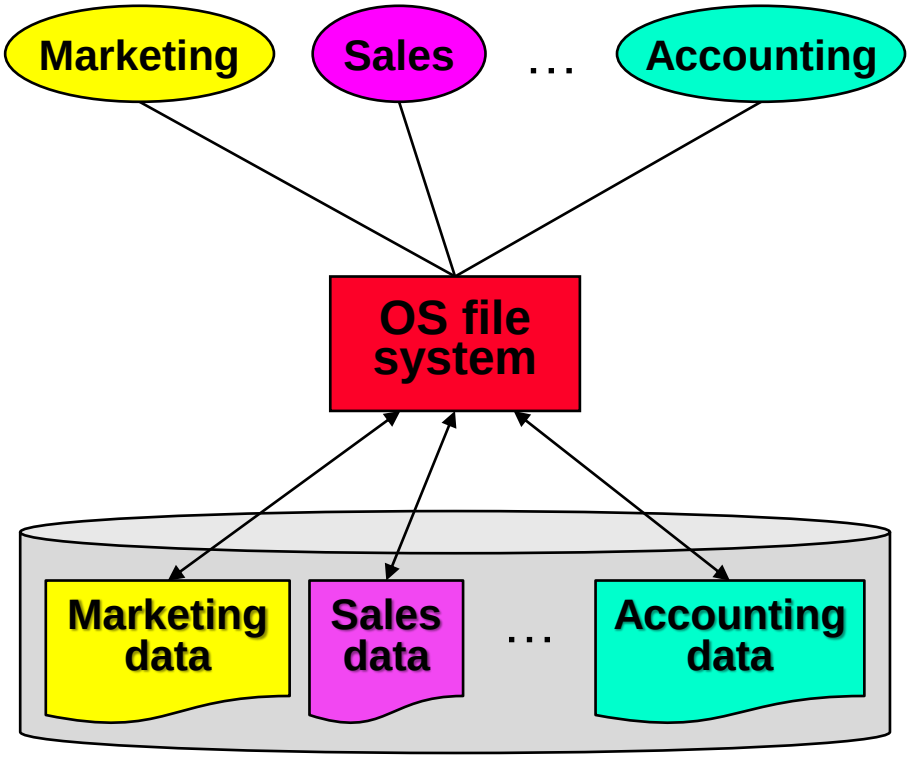
- ✓ What Is A Database Management System (DBMS)?
- ➡ **Why Do We Need Database Management Systems?**

  How Does A DBMS Manage Data

  What Are The Major Components Of A DBMS?

FILE-BASED APPROACH TO MANAGING DATA

Applications access stored data using the facilities provided by an operating system file system.

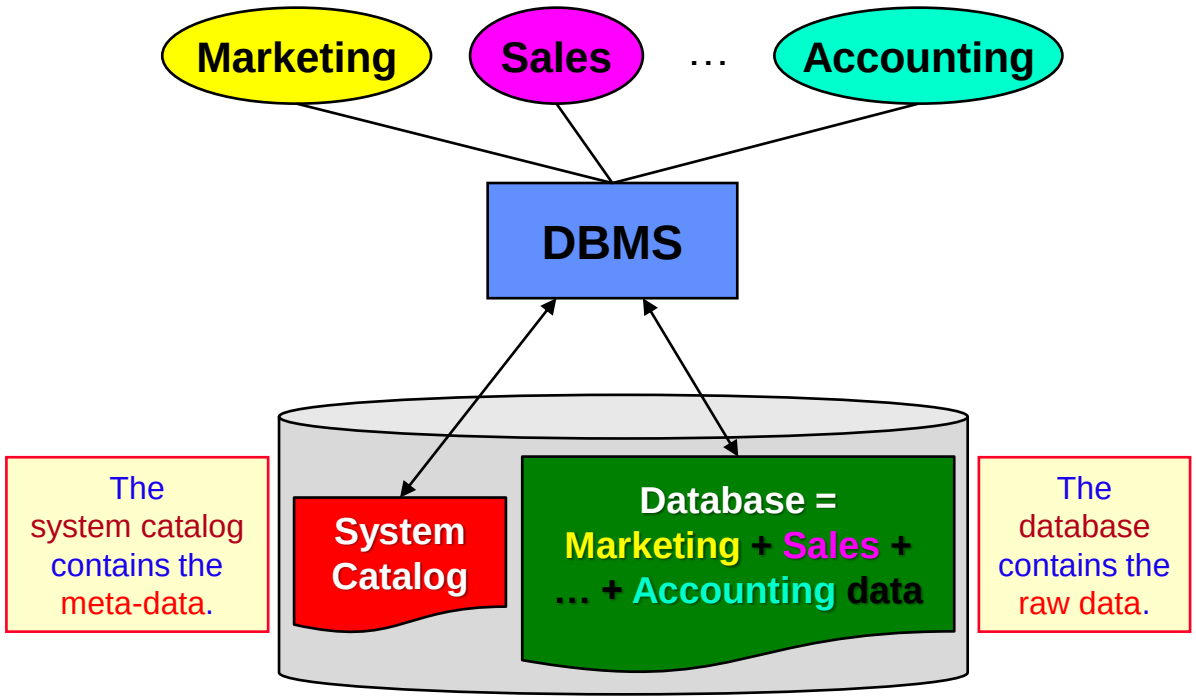


Drawbacks

- Data duplication and inconsistency
- Difficulty meeting unanticipated needs
- Data isolation
- Data integrity problems
- Atomicity of updates
- Concurrent access
- Security problems

DATABASE APPROACH TO MANAGING DATA

Applications access stored data using the facilities provided by a DBMS.



Major Principles

- **integrates** an organization's data.
- **separates** meta-data (description of data) and data.
- **supports** multiple views of data.
- **controls** definition and access of data centrally.

A DBMS provides automated solutions for the data management problems encountered when using file systems.

DATABASE MANAGEMENT SYSTEMS: OUTLINE

- ✓ What Is A Database Management System (DBMS)?
- ✓ Why Do We Need Database Management Systems?
- ➡ **How Does A DBMS Manage Data?**

  What Are The Major Components Of A DBMS?

DATA MODELS

A *data model* is a set of concepts for describing data that defines
– properties – relationships – semantics – constraints

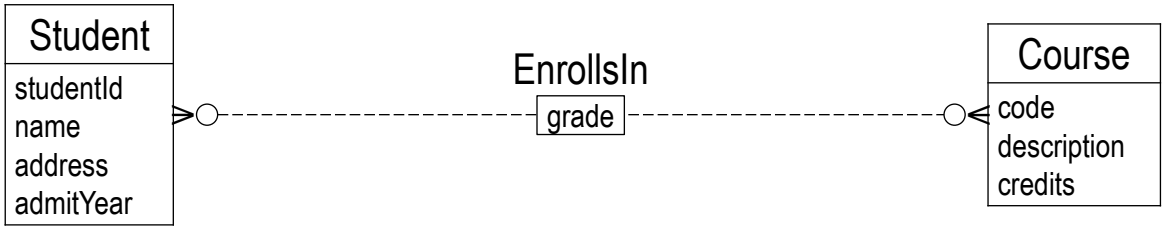
A data model is the fundamental mechanism used by a DBMS to logically describe and organize data and consists of:

1. **data structure types** ⟹ specify logical organization (properties, relationships and semantics)
2. **integrity constraints** ⟹ specify constraints (restrictions on properties and relationships)
3. **operations** ⟹ specify how data is accessed (e.g., R,I,U,D—Read, Insert, Update, Delete)

A data model is used to describe and organize data as well as to state any restrictions on the data.

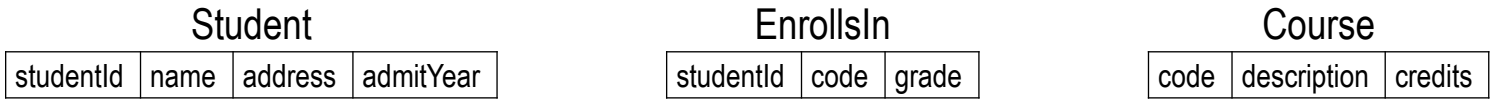
DATA MODELS: EXAMPLE

Entity-Relationship (E-R) model



☞ **Users view data as entities and explicit relationships among entities.**

Relational model



☞ **Users view data as tables and implicit relationships among tables.**

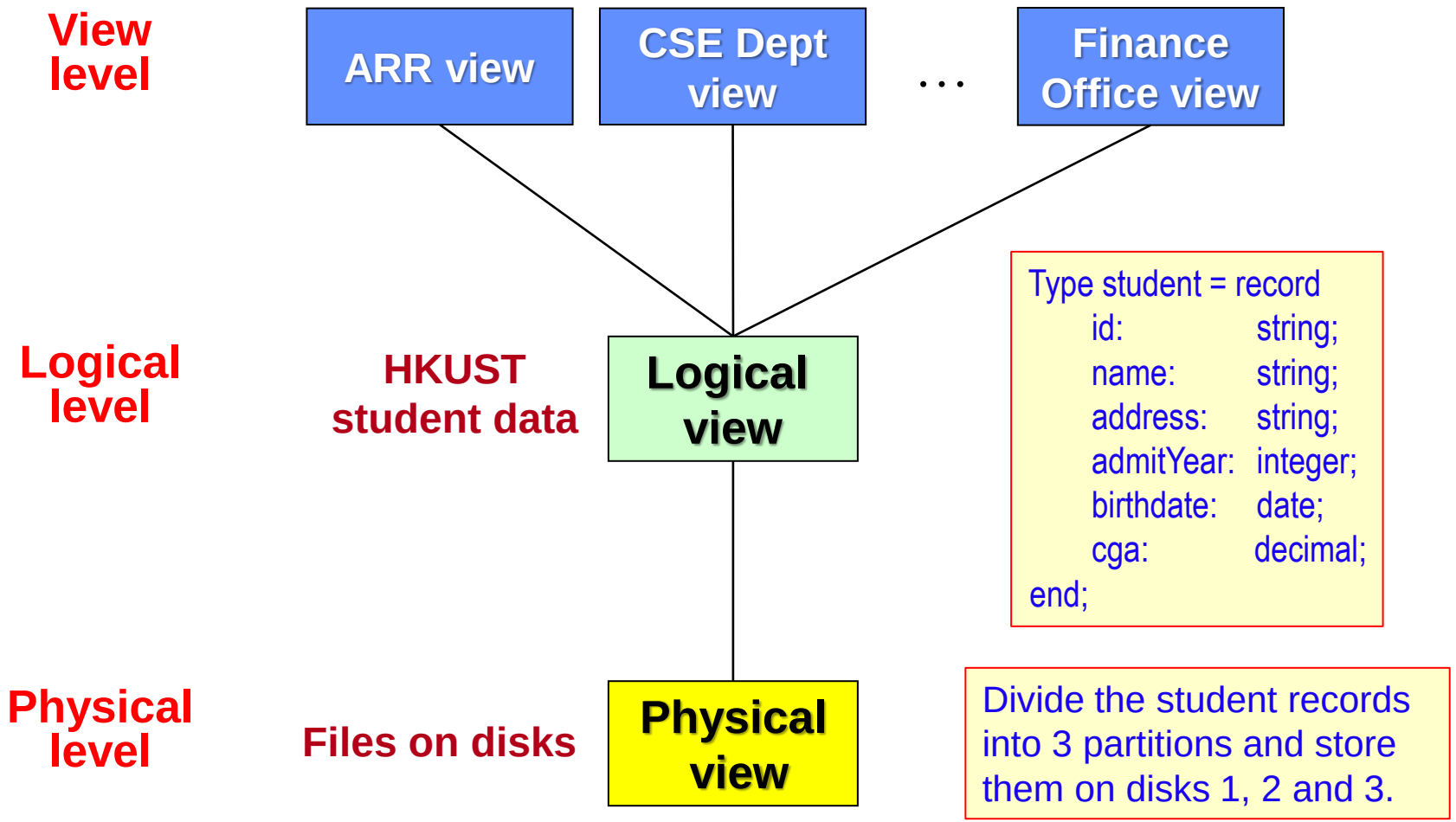
LEVELS OF ABSTRACTION

- One big problem in application development is the *separation* of application programs (i.e., code) from the data that they access.

- **Do I have to change my application program when I …**
  - replace my hard drive?
  - partition the data into two physical files (or merge two physical files into one)?
  - store salary as a floating-point number instead of an integer?
  - develop other applications that use the same data?
  - add more data fields to support other applications?
  - index the data using a B^+-tree instead of a hash index?

A DBMS provides separation of application programs and data via several levels of abstraction.

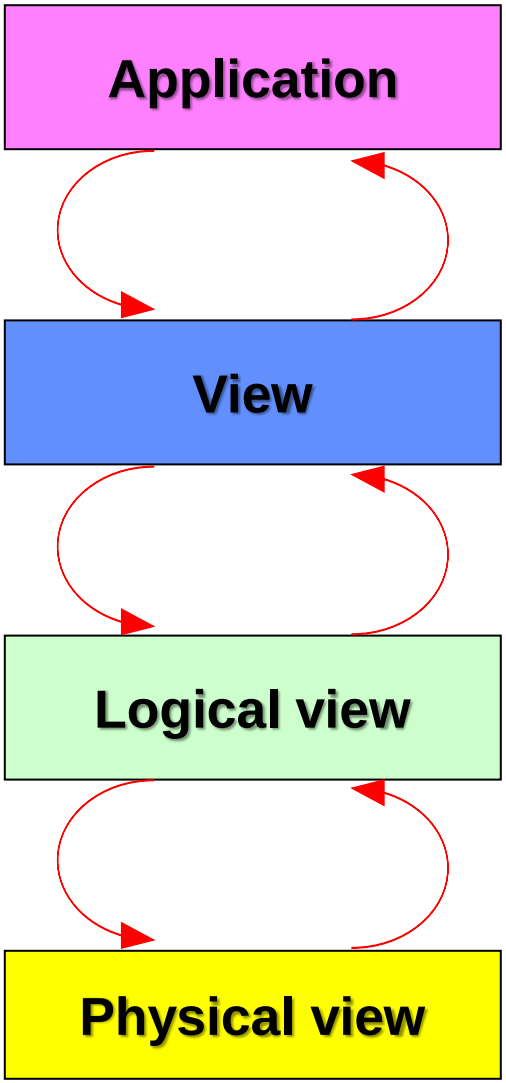
LEVELS OF ABSTRACTION (cont'd)



SCHEMAS AND INSTANCES

- A database *schema* describes the overall design of a database according to a data model.
  - ☞ **Stored in the system catalog; changes infrequently, if at all.**
- A DBMS uses several schemas, one for each level of abstraction, which *describes* the data at the corresponding level.
  - A view (subschema) describes the data that a user can access.
  - A logical schema describes the logical structure of the database (e.g., the set of students, courses and the relationship between them).
  - A physical schema describes the file formats and locations where the data are stored on disk.
- A database *instance* refers to the actual content of the database at a particular point in time.
  - ☞ **Conforms to its corresponding schema; changes frequently as data are changed.**

DATA INDEPENDENCE



Data independence **is the ability to modify a schema definition in one level of abstraction without affecting a schema definition in a higher level.**

(described by) ↔ View (subschema)

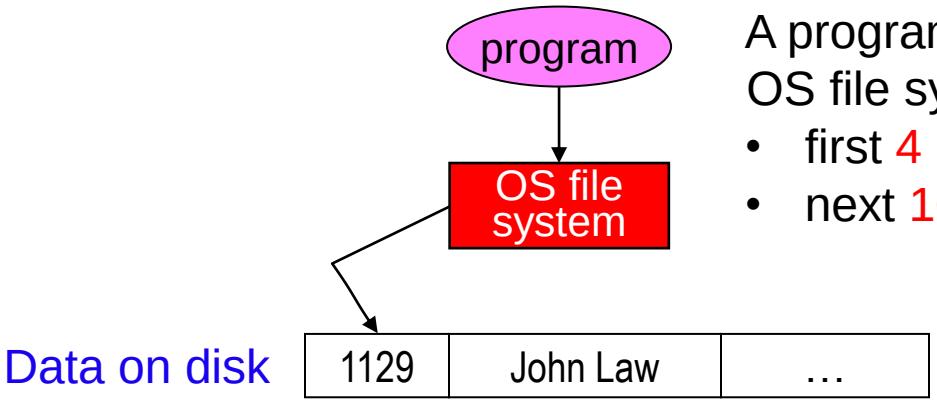
Logical data independence
(shields users from changes in the *logical structure*)

(described by) ↔ Logical schema

Physical data independence
(shields users from changes in the *physical structure*)

(described by) ↔ Physical schema

DATA INDEPENDENCE: EXAMPLE

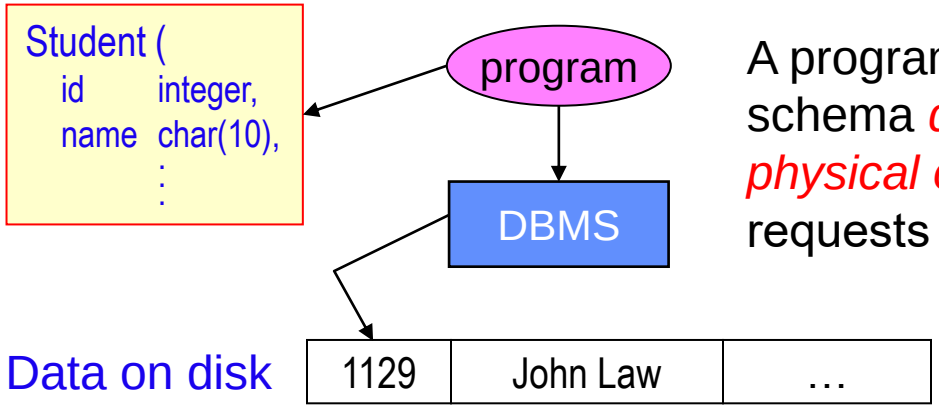


A program accessing the data via the OS file system must know:

- first 4 bytes is an id (a number)
- next 10 bytes is a name (a string)

⋮

Schema

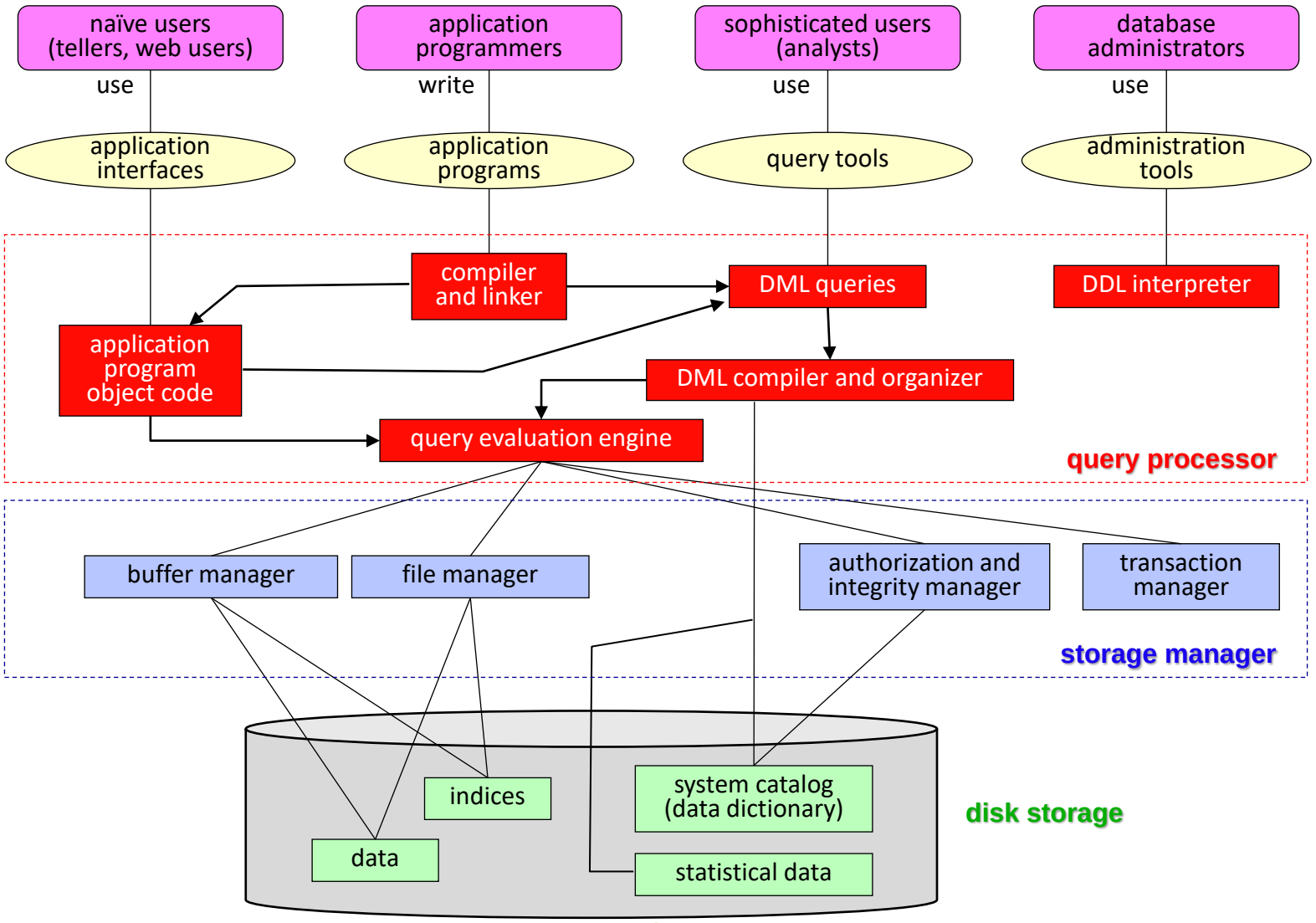


A program accessing the data via a schema *does not need to know its physical organization*; it simply requests the student's id, name, etc.

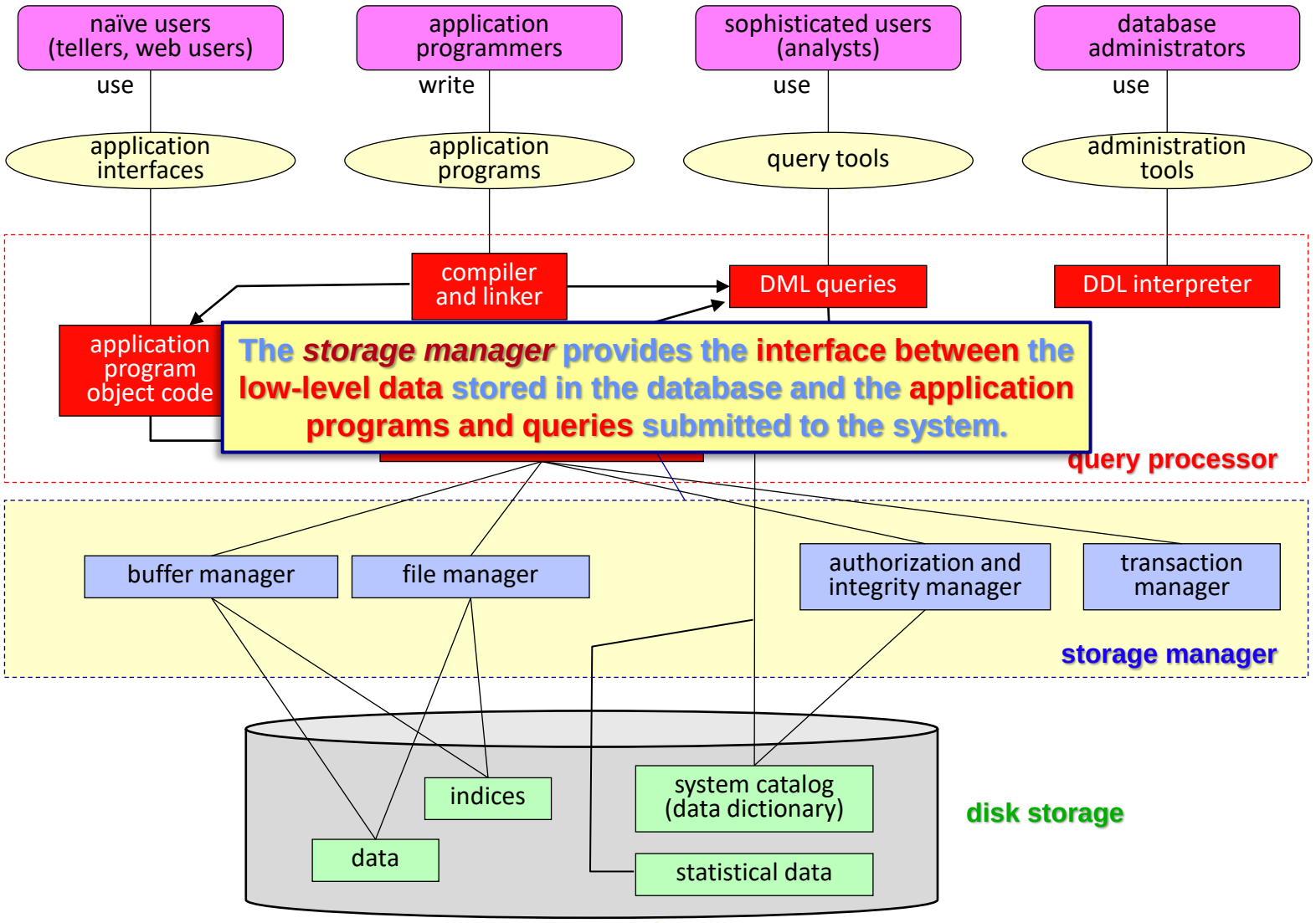
DATABASE MANAGEMENT SYSTEMS: OUTLINE

- ✓ What Is A Database Management System (DBMS)?
- ✓ Why Do We Need Database Management Systems?
- ✓ How Does A DBMS Manage Data?
- ➡ **What Are The Major Components Of A DBMS?**

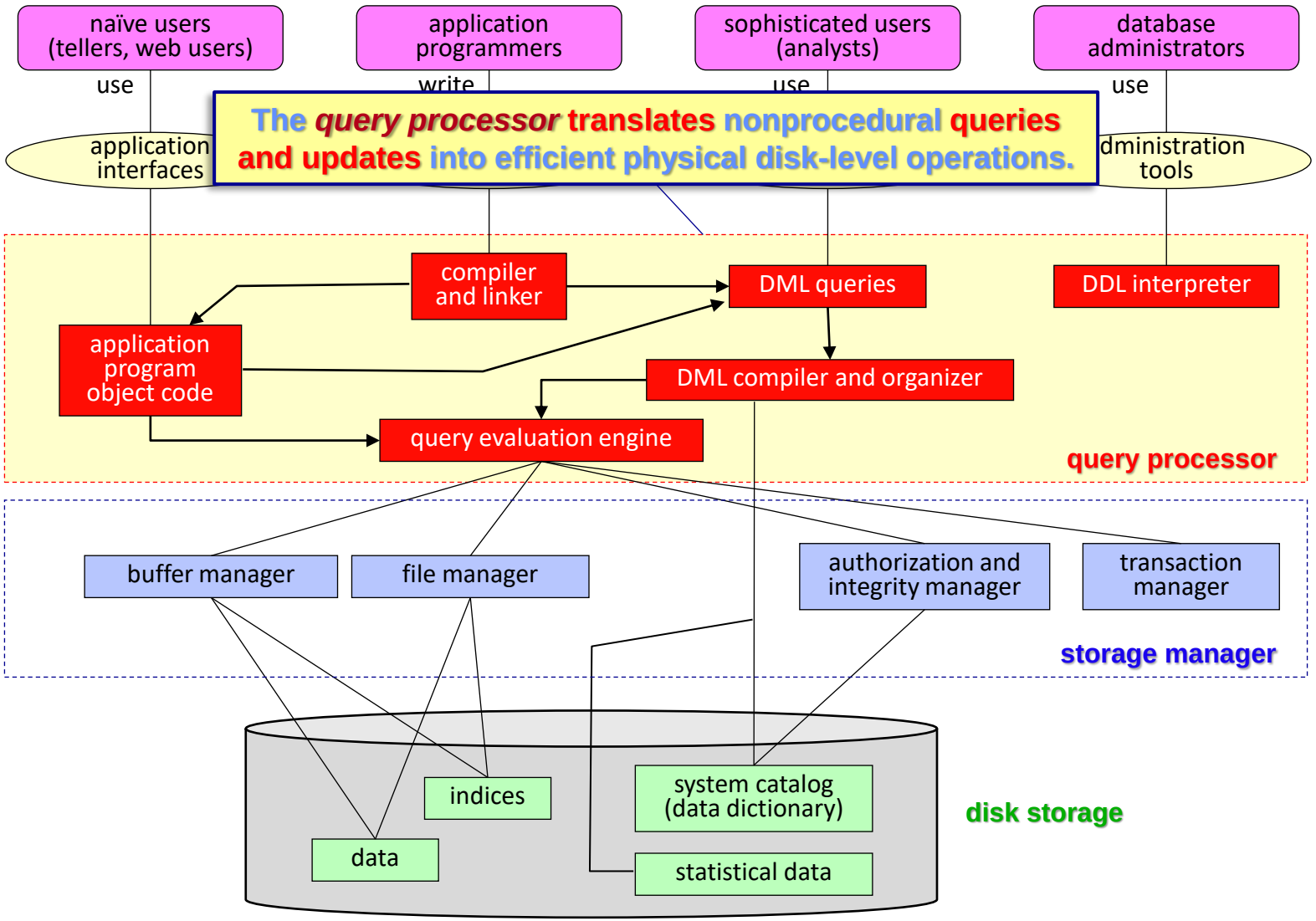
DBMS ARCHITECTURE



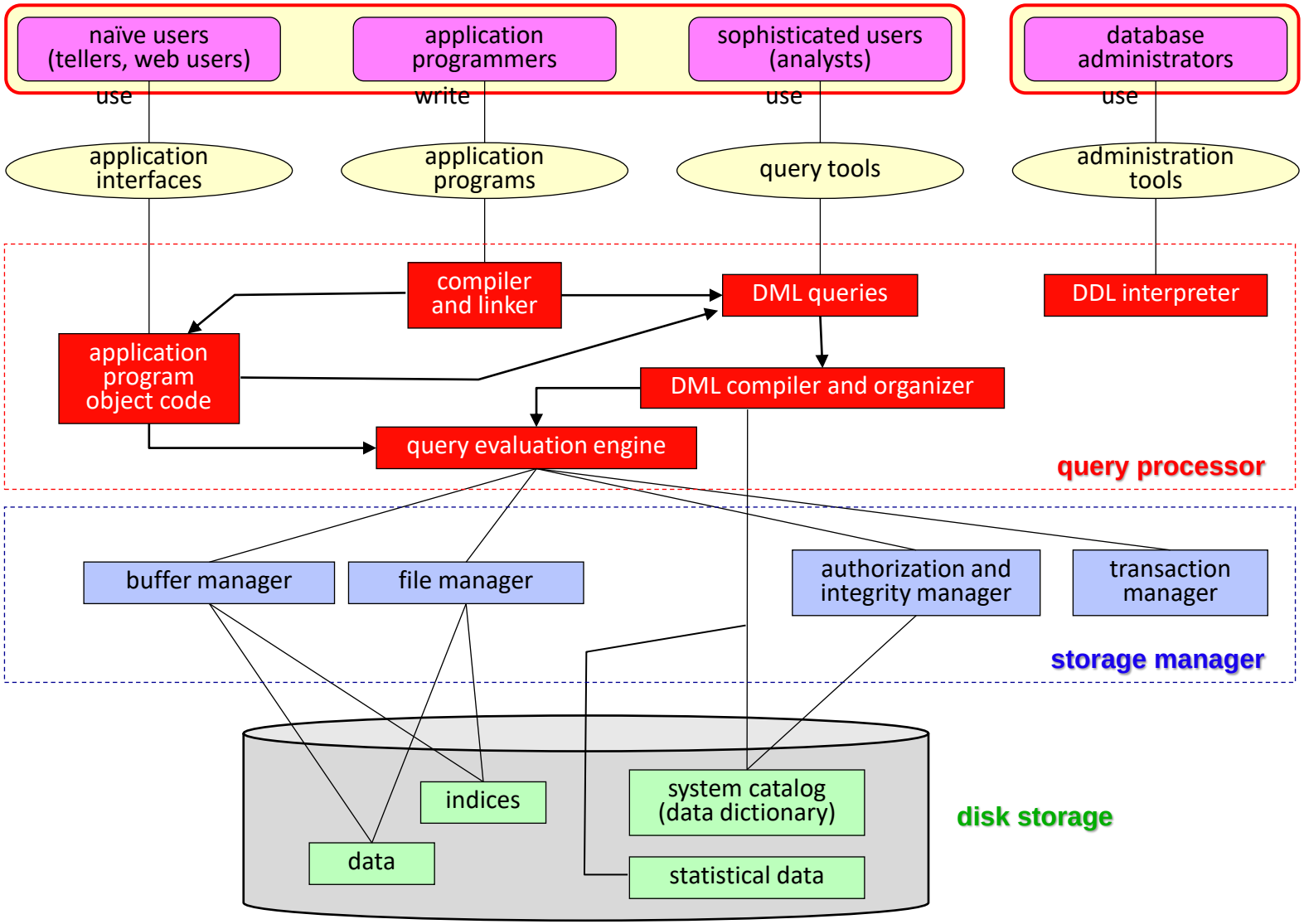
DBMS ARCHITECTURE



DBMS ARCHITECTURE



DBMS USERS


naïve users (tellers, web users)
application programmers
sophisticated users (analysts)
database administrators
use
write
use
use
application interfaces
application programs
query tools
administration tools
compiler and linker
DML queries
DDL interpreter
application program object code
DML compiler and organizer
query evaluation engine
query processor
buffer manager
file manager
authorization and integrity manager
transaction manager
storage manager
indices
system catalog (data dictionary)
data
statistical data
disk storage

DATABASE MANAGEMENT SYSTEMS: SUMMARY

- **Database management systems (DBMSs)** address the limitations of OS file systems for managing an enterprise's data.
- **Data models** are the foundation for developing a database—the entity-relationship (E-R) model and relational model are commonly used in practice.
- **Data independence** is fundamental to understanding how a DBMS manages data at different abstraction levels.
- **A DBMS provides many facilities** for query processing and storage management to efficiently handle the data management and data access needs of various users.

DSAA 5012: SYLLABUS

- ✓ Database Management Systems
- ➡ **Entity-Relationship (E-R) Model and Database Design**
- Relational Algebra
- Structured Query Language (SQL)
- Relational Database Design
- Storage and File Structure
- **Indexing**
- **Query Processing**
- **Query Optimization**
- **Transactions**
- **Concurrency Control**
- **Recovery System**
- **Graph Databases**
- **Uncertain Databases**