Introduction: Database System, Characteristics (Database Vs File System), Advantages of Database Systems, Database Applications, Database Users, Brief Introduction of Different Data Models; Concepts of Schema, Instance and Data Independence; Three Tier Schema Architecture for Data Independence; Database System Structure, Centralized and Client Server Architecture for Database Systems.

Entity Relationship Model: Introduction, Concept of Entities, Attributes, Entity Sets, Relationships, Relationship Sets, Key and Participation Constraints, Class Hierarchies, and Aggregation, Developing E-R diagrams for Databases.

Introduction to Database Systems

Database Management System

Database: A collection of related data. It should support

Definition
Construction
Manipulation

Database Management System: A collection of programs that enable the users to create and maintain a database.

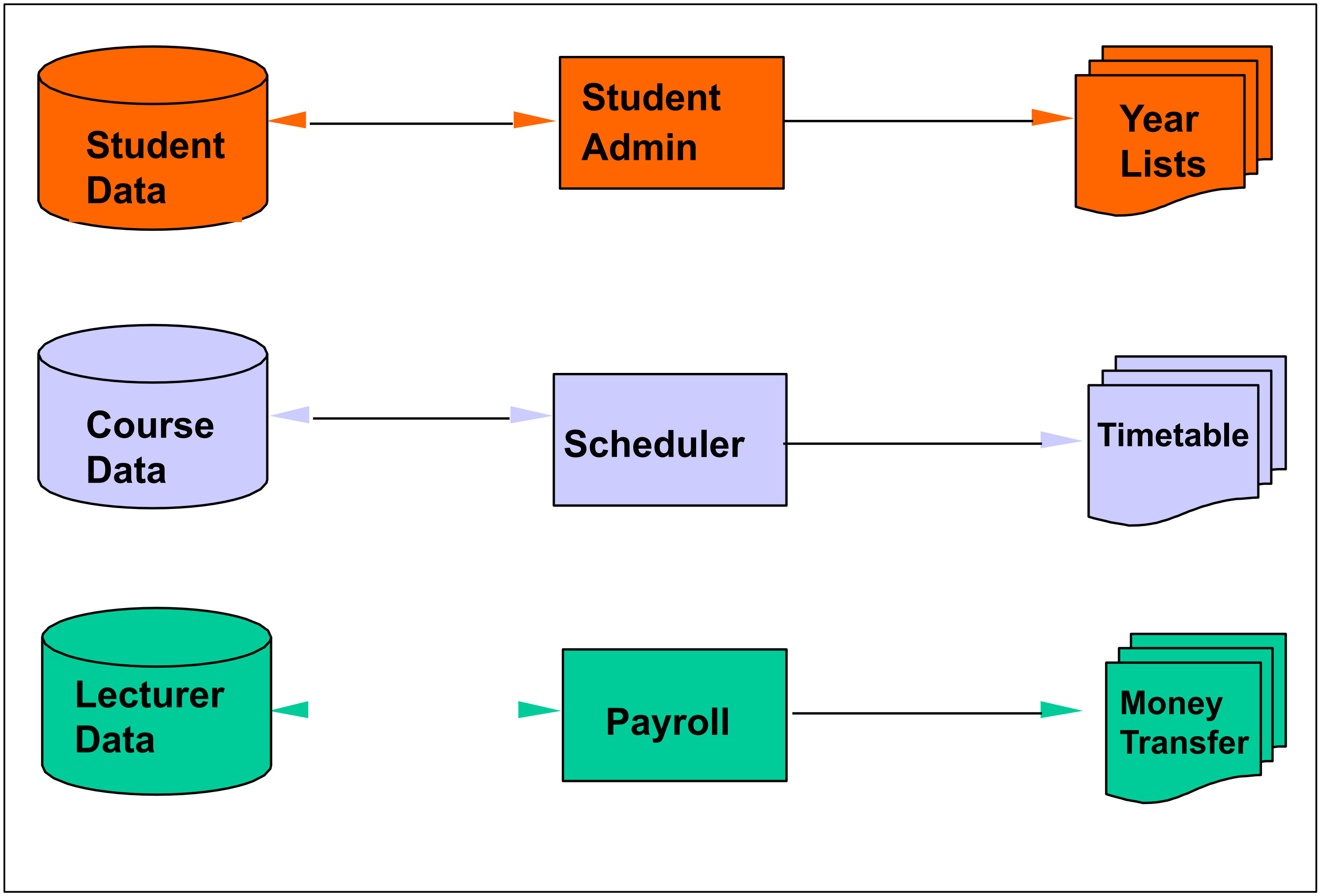
Database Management System (DBMS)

- A DBMS is a software package designed to *store* and *manage* databases
- A DBMS provides *generic functionality* (see previous slide) that otherwise would have to be implemented over and over again

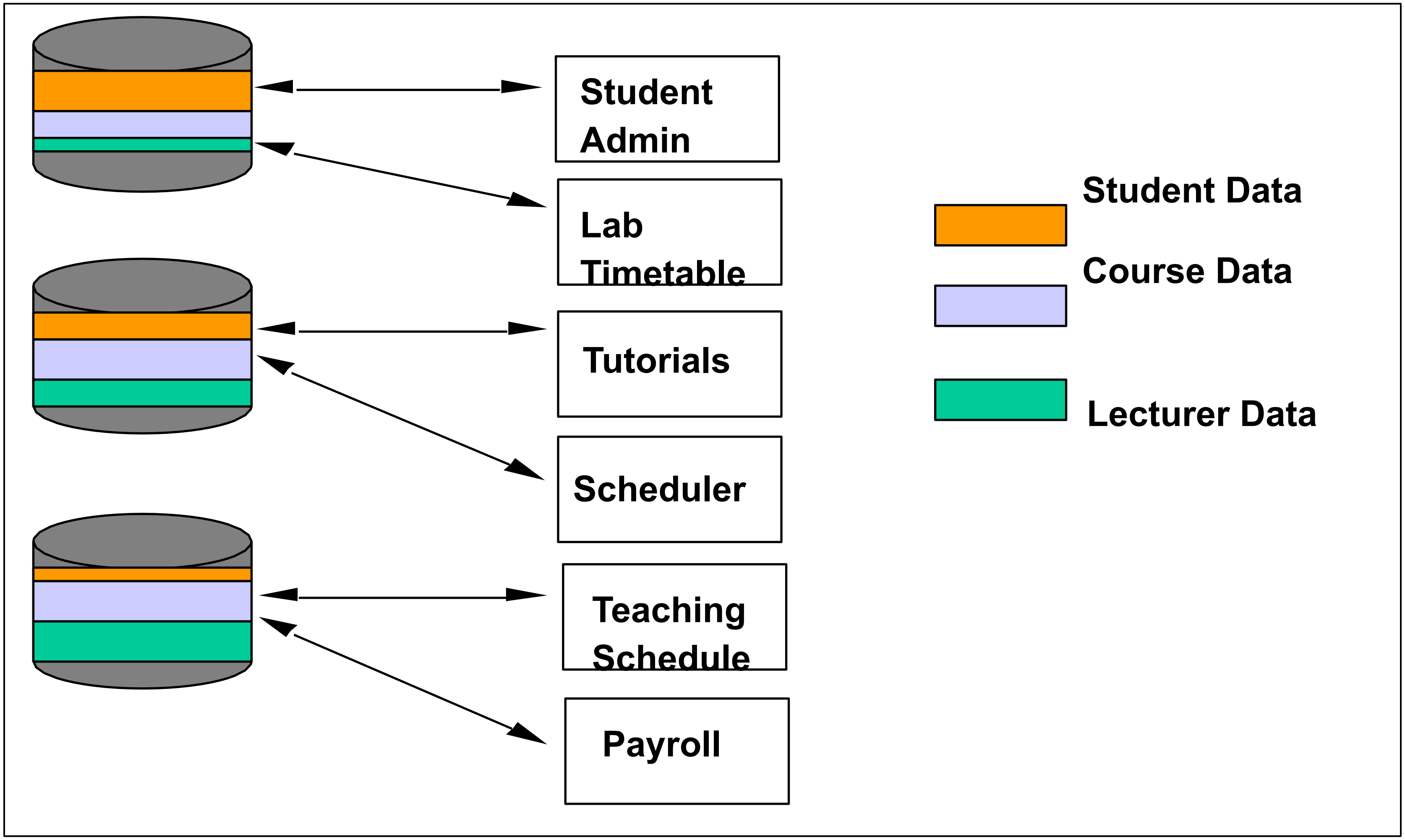
   Reduced application development time

- Several brands, e.g.,
  - Oracle , DB2 (IBM), SQL Server, Access (Microsoft), MySQL, SQLite (open source)

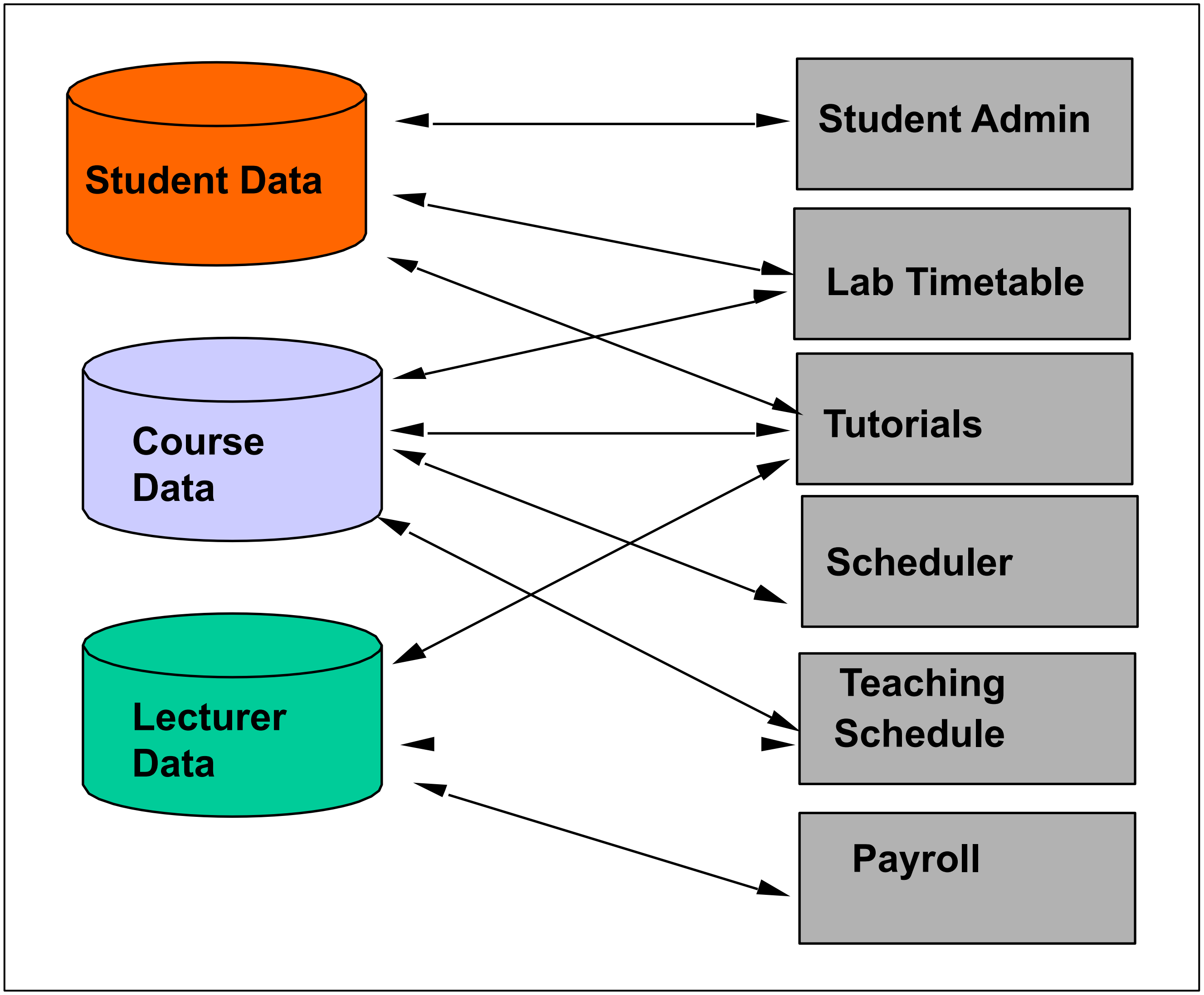
File System: A Physical Interface



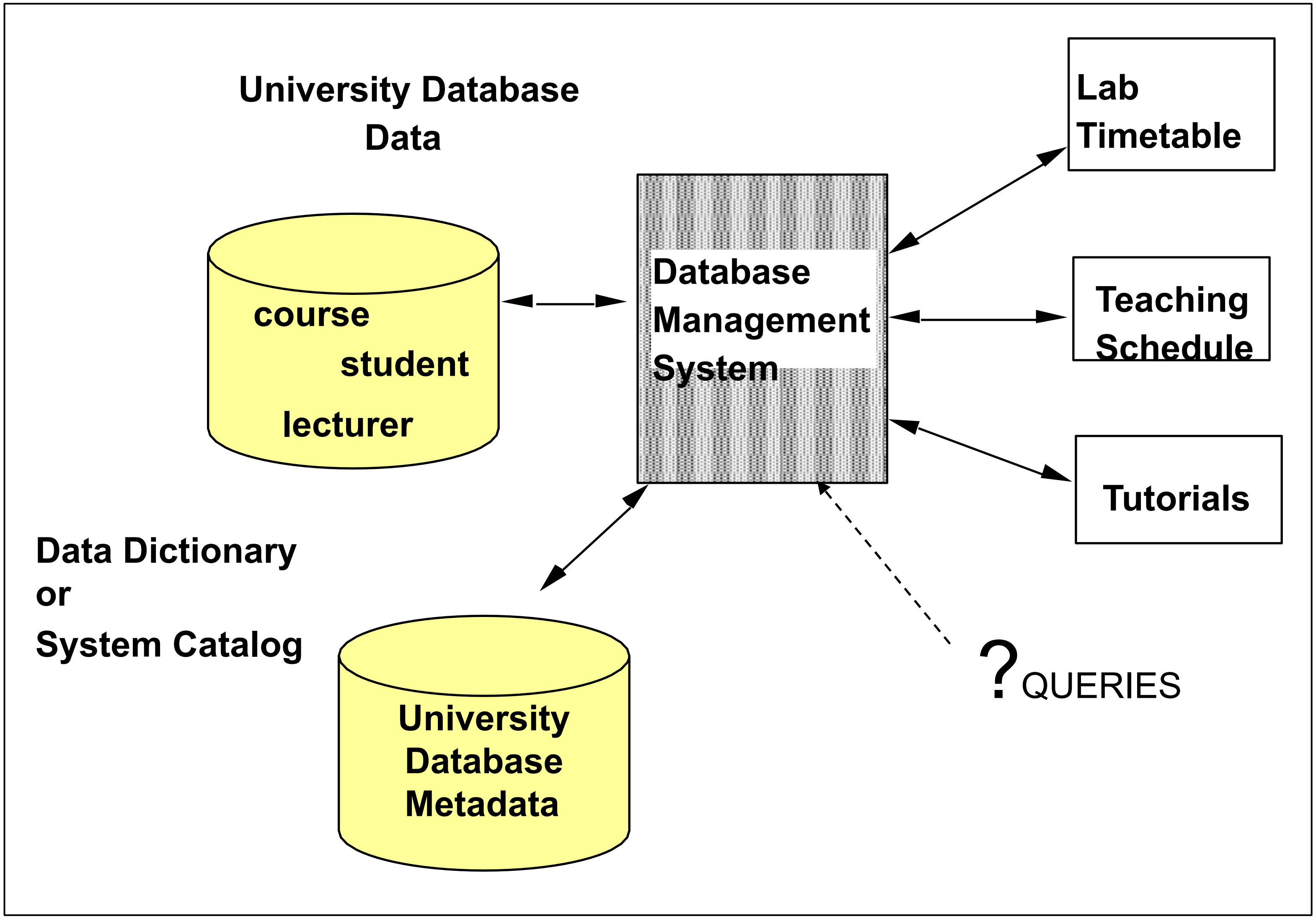
Sharing Data:



Sharing Data and Operations



DBMS: A Logical Interface



File System Approach

- Uncontrolled redundancy
- Inconsistent data
- Inflexibility
- Limited data sharing
- Poor enforcement of standards
- Low programmer productivity
- Excessive program maintenance
- Excessive data maintenance

DBMS Approach

- Controlled redundancy
  - consistency of data & integrity constraints
- Integration of data
  - self-contained
  - represents semantics of application
- Data and operation sharing
  - multiple interfaces
- Services & controls
  - security & privacy controls
  - backup & recovery
  - enforcement of standards
- Flexibility
  - data independence
  - data accessibility
  - reduced program maintenance
- Ease of application development

If an application is

- simple
- stringent real-time
- single user
- static,

files are the option of choice

- In a file system, data is *physically accessed* and *not integrated*

- In a DBMS, data is *logically accessed* and *integrated*:
  - query language
  - data dictionary

Limitations of File-based

Separation and isolation

Duplication

Program & data dependence

Fixed queries

Propagation of application programs

Drawbacks of using file systems to store data

Data redundancy and inconsistency

Multiple file formats, duplication of information in different files

Difficulty in accessing data

Need to write a new program to carry out each new task

Data isolation

Multiple files and formats

Integrity problems

Integrity constraints (e.g., account balance > 0) become "buried" in program code rather than being stated explicitly

Hard to add new constraints or change existing ones

Drawbacks of using file systems to store data (Cont.)

Atomicity of updates

Failures may leave database in an inconsistent state with partial updates carried out

Example: Transfer of funds from one account to another should either complete or not happen at all

Concurrent access by multiple users

Concurrent access needed for performance

Uncontrolled concurrent accesses can lead to inconsistencies

Example: Two people reading a balance (say 100) and updating it by withdrawing money (say 50 each) at the same time

Security problems

Hard to provide user access to some, but not all, data

Database systems offer solutions to all the above problems

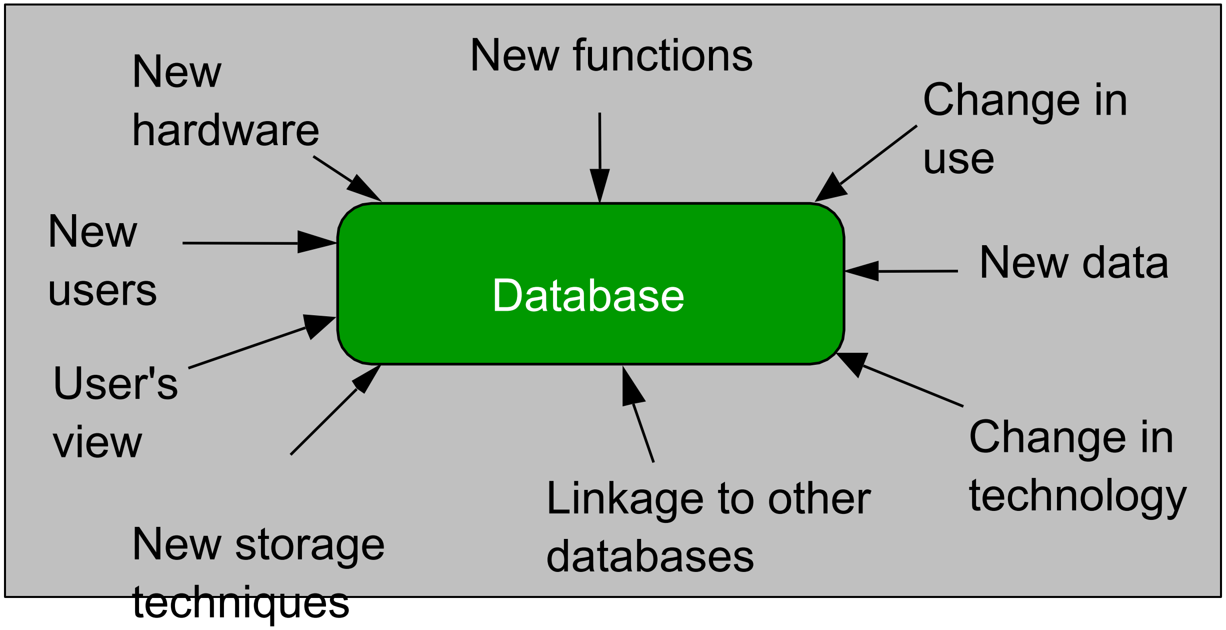
Characteristics of the DB Approach

- *Insulation* of application programs and data from each other

- Use of a *catalogue* to store the schema

- Support of *multiple user views*

Data Independence



- **Logical** data independence
  - change the logical schema without having to change the external schemas
- **Physical** data independence
  - change the internal schema without having to change the logical schema

Database Management System Facility

Data definition language (DDL)

Data manipulation language (DML)

Structured query language (SQL)

Security system

Integrity system

Concurrency control system

Backup & recovery system

View mechanism

DBMS Environment

Hardware

Client-server architecture

Software

dbms, os, network, application

Data

Schema, subschema, table, attribute

People

Data administrator & database administrator

Database designer: logical & physical

Application programmer

End-user: naive & sophisticated

Procedure

Start, stop, log on, log off, back up, recovery

Advantages of DBMS

Control redundancy

Consistency

Integrity

Security

Concurrency control

Backup & recovery

Data standard

More information

Data sharing & conflict control

Productivity & accessibility

Economy of scale

Maintenance

Limitations of DBMS

Complexity

Size

Cost

- Software
- Hardware
- Conversion

Performance

Vulnerability

Features of DBMS

1. Data storage, retrieval, and update: The ability to store, retrieve, and update the data that are in the database.

2. User-accessible catalog: where descriptions of database components are stored and are accessible to the users

3. Shared update support: A mechanism to ensure accuracy when several users are updating the database at the same time

4. Backup and Recovery Services: Mechanisms for recovering the database in the event that a database is damaged somehow.

5. Security Services: Mechanisms to ensure that certain rules are followed with regard to data in the database and any changes that are made in the data

Features of DBMS

5. Integrity services: Mechanisms to ensure that certain rules are followed with regard to data in the database and any changes that are made in the data.

6. Data Independence: Facilities to support the independence of programs from the structure of the database.

7. Replication support: A facility to manage copies of the same data at multiple locations.

8. Utility Services: DBMS provided services that assist in the general maintenance of the database.

Security

Protection against unauthorized access: either intentional or accidental.

Three main features for protection

Passwords: Allows only authorized users to access the database. Access privileges can be provided based on access needs

Encryption: Encodes data to non-decipherable. Data decoded on demand to prevent hackers from accessing data

Views: Different snapshot of the data ensures that users only get access to data they need

Integrity

Integrity Constraints are the conditions that data must satisfy during initial input & updates.

There are four categories of constraints

Data Type

Legal Values

Format

Key Constraints

Entity Integrity Constraints (Primary Key)

Enforces the uniqueness of the primary key

Referential Integrity Constraints (Foreign Key)

Value of foreign key must match the value of primary key for some row in another table

Replication

Duplication of data at multiple physical locations

Each replica of the data can be changed independently

Periodically the replicas update their data to the master database – this process is called synchronization

Data Dictionary

Contains information describing the database

- Schema for the database
- Characteristic for each field
- Possible values for each field
- Description of the data
- Relationships
- Description of the programs

Database Languages

- Data Definition Language (DDL)
  - Commands for setting up the schema of a database
  - The process of designing a schema can be complex, may use a design methodology and/or tool
- Data Manipulation Language (DML)
  - Commands to manipulate data in database:

    `RETRIEVE, INSERT, DELETE, MODIFY`

  - Also called “query language”

Building an Application with a DBMS

- Requirements gathering (natural language, pictures)
- Requirements modeling (conceptual data model, ER)
  - Decide what *entities* should be part of the application and how they should be *related*
- Schema design and implementation
  - Decide on a set of *tables*, *attributes*
  - Create the tables in the database system
  - Populate database (insert records/tuples)
- Write application programs using the DBMS
  - … a lot easier now that
    the data management is taken care of

Database Management System (DBMS)

DBMS contains information about a particular enterprise

- Collection of interrelated data
- Set of programs to access the data
- An environment that is both *convenient* and *efficient* to use

Database Applications:

- Banking: transactions
- Airlines: reservations, schedules
- Universities: registration, grades
- Sales: customers, products, purchases
- Online retailers: order tracking, customized recommendations
- Manufacturing: production, inventory, orders, supply chain
- Human resources: employee records, salaries, tax deductions

Databases can be very large.

Databases touch all aspects of our lives

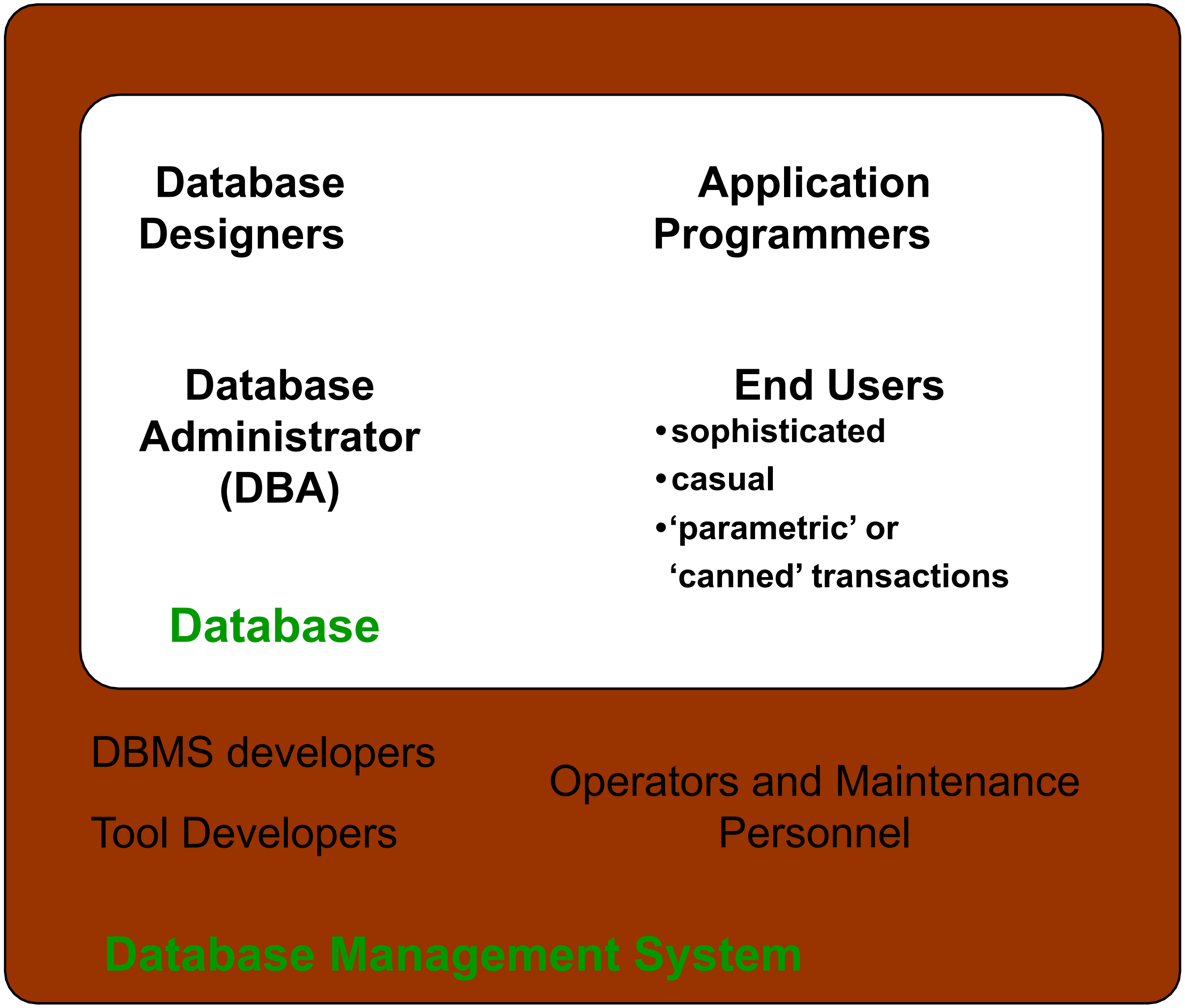
University Database Example

Application program examples

Add new students, instructors, and courses
Register students for courses, and generate class rosters
Assign grades to students, compute grade point averages (GPA) and generate transcripts

In the early days, database applications were built directly on top of file systems

Database Actors



Database Designers

Application Programmers

Database Administrator (DBA)

End Users

- **sophisticated**
- **casual**
- **‘parametric’ or ‘canned’ transactions**

Database

DBMS developers

Tool Developers

Operators and Maintenance Personnel

Database Management System

“on the scenes”

“behind the scenes”