COMP3311 Database Management Systems

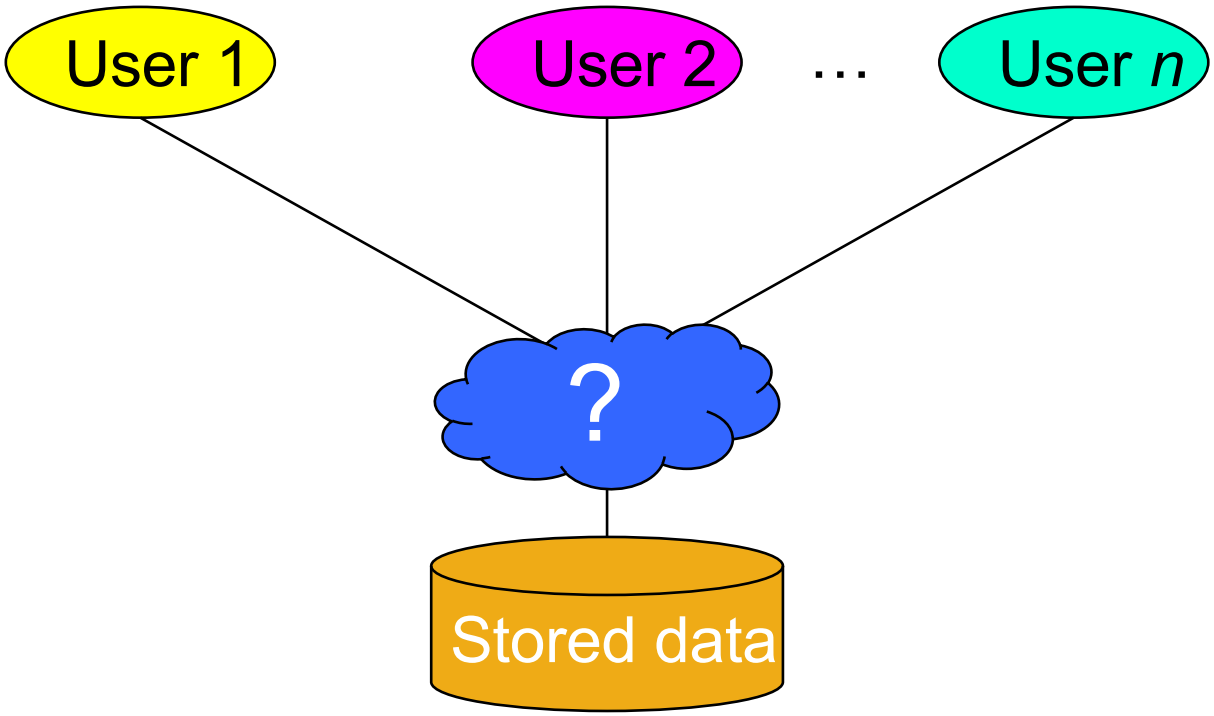
Spring 2022

香港科技大學
THE HONG KONG UNIVERSITY OF SCIENCE AND TECHNOLOGY

DBMS Introduction

Prof Xiaofang Zhou

What is a DBMS?



How to organize, access, share, protect, … stored data?

+ Data and Databases

- Data are facts such as age, salary, name, address, etc.
- A database has the following properties.
  - It usually represents some aspect of the real world
  - The data have some inherent meaning
  - It is designed, built and populated with data for a specific purpose
    - sales, human resources, manufacturing, banking, real estate, stock trading, inventory management, …
- Databases touch all aspects of our lives!

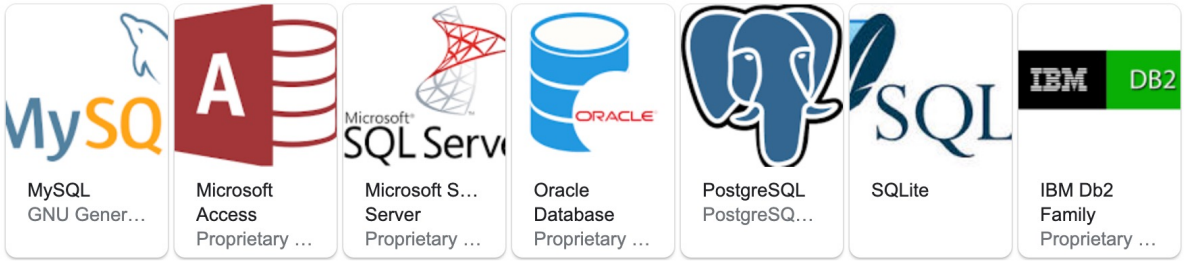
A database is a collection of related data

+ DBMS

- A database management system (DBMS) is a general-purpose software package that manages databases
- A DBMS provides support/facilities for:
  - Defining types, structures, constraints on data
  - Storing data on some storage device
  - Manipulating data (querying, updating)
  - Sharing data among many users
  - Protecting data from loss, corruption, unauthorized access
- A DBMS provides an environment for managing data that is both convenient and efficient to use

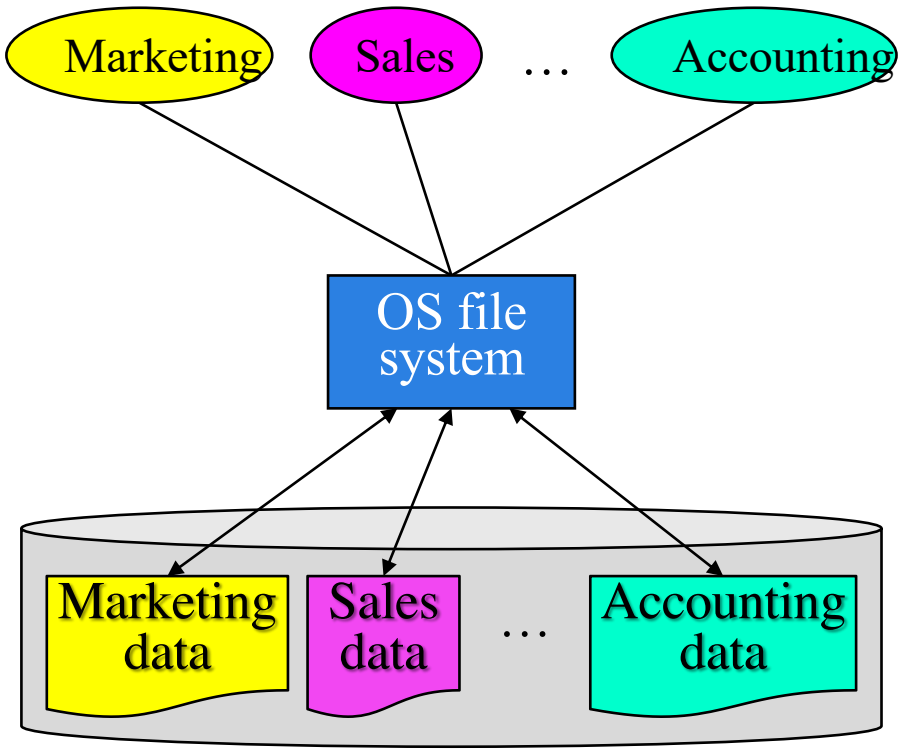
+ Popular DBMS Products

- Oracle
- IBM DB2
- Microsoft SQL Server
- MySQL (MariaDB)
- PostgreSQL
- Microsoft Access
- dBASE
- SQLite…



+ DBMS vs File Systems

In file processing, applications access stored data using the facilities provided by an operating system file system.

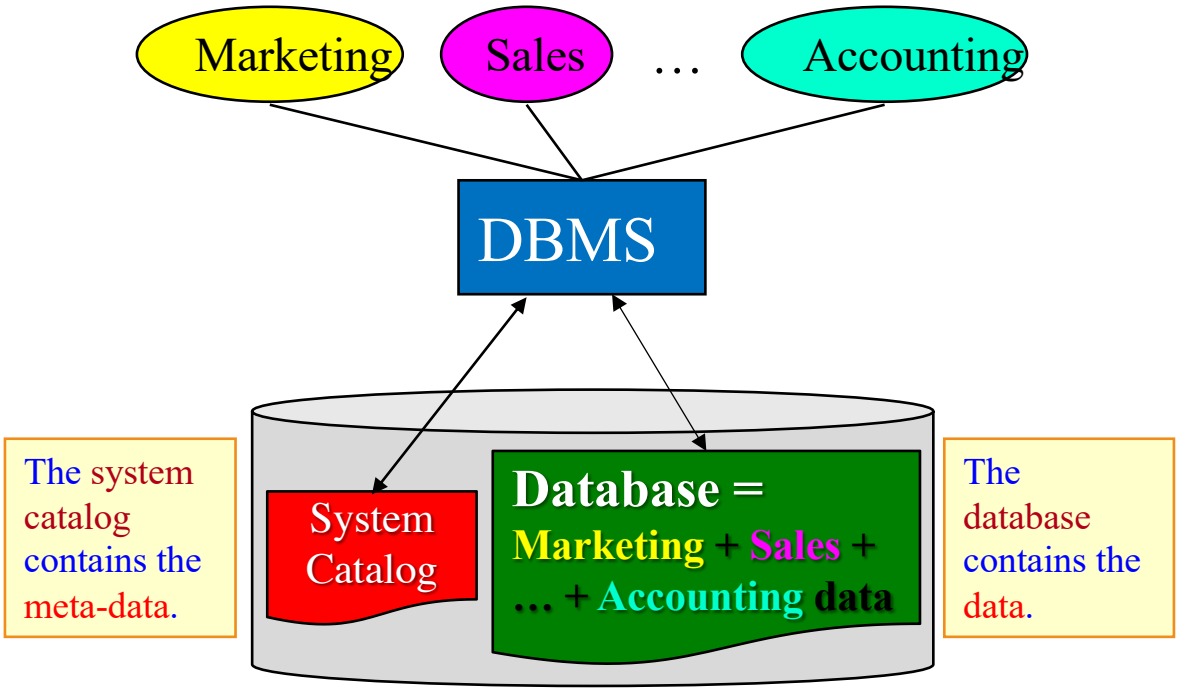


Drawbacks

- Data redundancy & inconsistency
- Difficulty in accessing data
- Data isolation
- Integrity problems
- Atomicity of updates
- Concurrent access
- Security problems

+ DBMS vs File Systems

In database processing, 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.

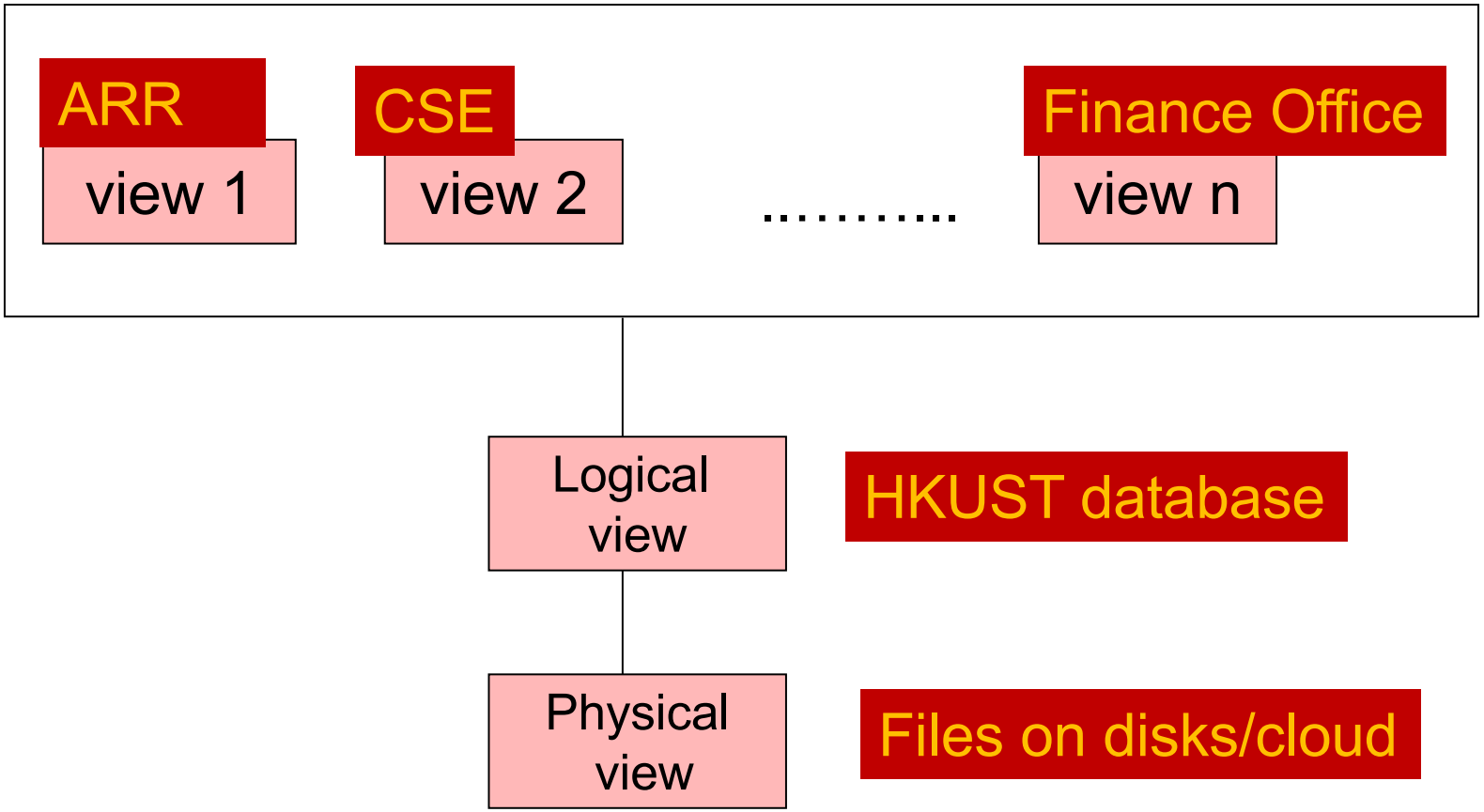
A DBMS provides automated solutions for the data management problems encountered when using file systems

Data Independence

- One big problem in application development is the separation of applications from data
- Do I have to change my program when I …
  - replace my hard drive?
  - store the data in a B-tree instead of a hash file?
  - partition the data into two physical files (or merge two physical files into one)?
  - store salary as floating point number instead of integer?
  - develop other applications that use the same set of data?
  - add more data fields to support other applications?
- Solution: introduce levels of abstraction

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

+ Three Levels of Abstraction



Three Levels of Abstraction

- Physical level: describe how a record is stored on disks
  - e.g., “Divide the customer records into 3 partitions and store them on disks 1, 2 and 3.”
- Logical level: describe how data are structured in database, and the relationships among the data
  - Similar to defining a record type or class in a programming language:

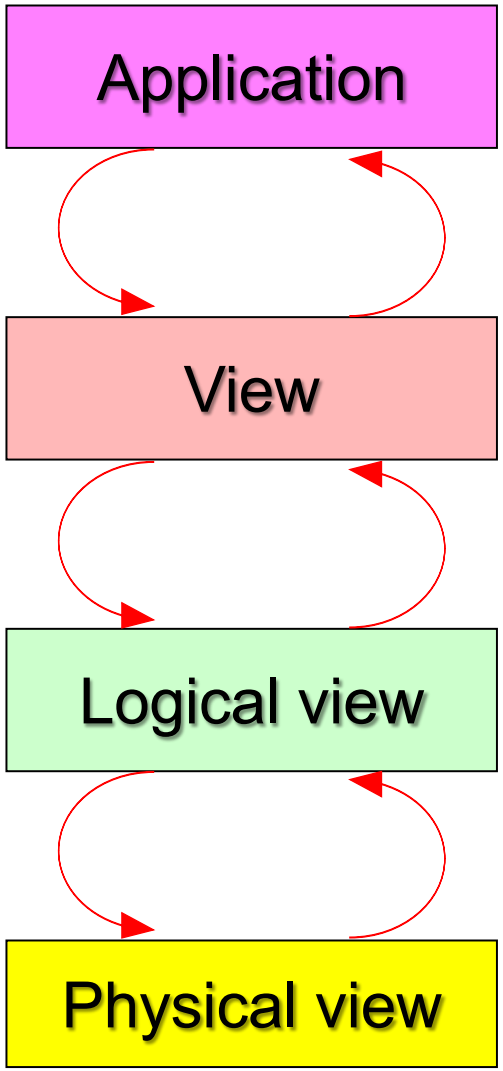
```
class customer {
        string name;
        string street;
        int city;
};
```

- View level: Define a subset of the database for a particular application. Views can hide information (e.g. salary) for security purposes or add information (e.g., age).

+ Instances and Schemas

- Each level is defined by a schema, which describes the data at the corresponding level
  - A logical schema defines the logical structure of the database (e.g., set of customers and accounts and the relationship between them)
  - A physical schema defines the file formats and locations
- A database instance refers to the actual content of the database at a particular point in time.
  - A database instance must conform to the corresponding schema

+ 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)

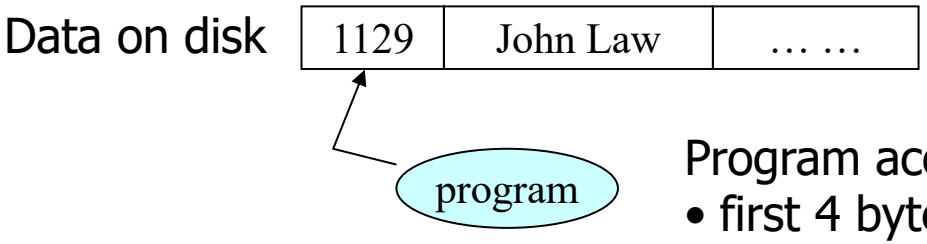
Logical data independence
(shields users from changes in the logical structure)

(described by) Logical schema

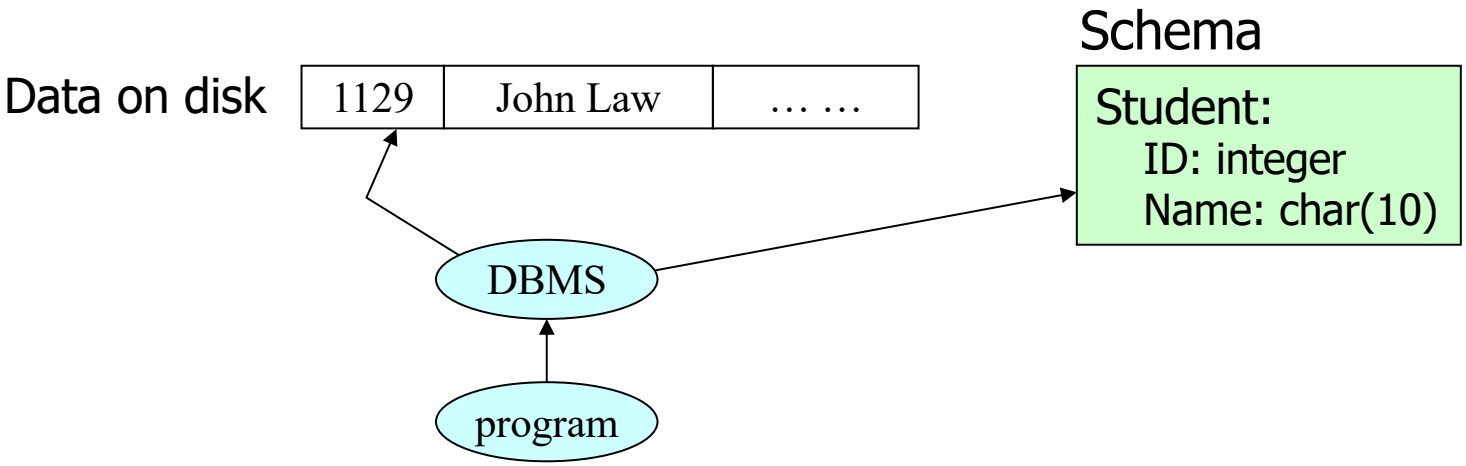
Physical data independence
(shields users from changes in the physical structure)

(described by) Physical schema

+ An Example of Data Independence



Program accessing data directly has to know:
- first 4 bytes is an ID number
- next 10 bytes is an employee name

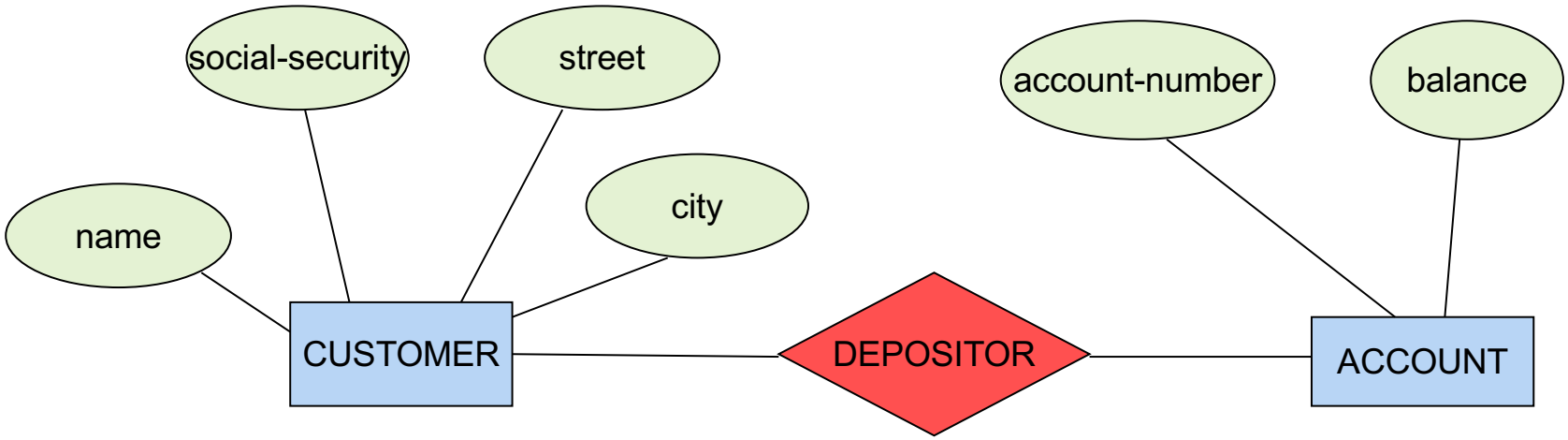


+ Data Models

- A collection of tools for describing:
  - Data
  - Relationships among data
  - Data semantics
  - Constraints on data

+ Entity-Relationship Model

- Example of entity-relationship model



+ Relational Model

Example of tabular data in the relational model:

name	social-security	street	city	account-number
Johnson	192-83-7465	Alma	Palo Alto	A-101
Smith	019-28-3746	North	Rye	A-215
Johnson	192-83-7465	Alma	Palo Alto	A-201
Jones	321-12-3123	Main	Harrison	A-217
Smith	019-28-3746	North	Rye	A-201

account-number	balance
A-101	500
A-201	900
A-215	700
A-217	750

+ Data Definition Language (DDL)

- Notation for defining the database schema
  - Express how data are organized in a formal language
  - Examples:

```
CREATE TABLE customer (
        customer-name   varchar(40),
        social-security     char(11),
        customer-street   varchar(100),
        customer-city       varchar(20),
        account-number varchar(10));

CREATE TABLE account (
        account-number char(10),
        balance              integer);
```

+ Data Manipulation Language (DML)

- Language for accessing and manipulating the data organized by the data model
- Two types of theoretical DML
  - Algebra (Procedural) - user specifies what data is required and how to get those data.
  - Calculus (Nonprocedural) - user specifies what data is required without specifying how to get those data
- DML in practice
  - SQL

+ SQL

- Most common language – used in all commercial DBMSs
- Including DML, DDL and more
- Example

```
SELECT account-number
FROM   account
WHERE balance <= 0
```

+ Transaction Management

- A transaction is a collection of operations that performs a single logical function in the database
  - Example: ATM withdrawal

    Read account record

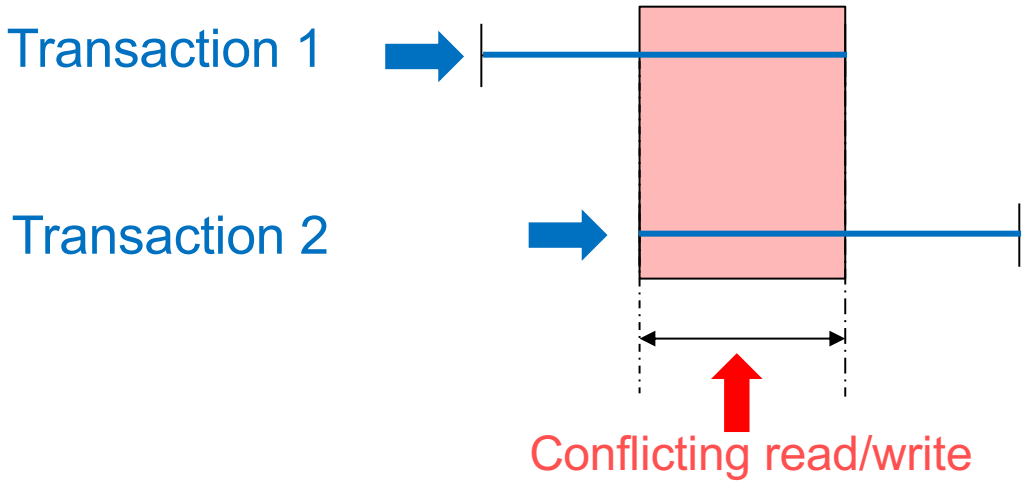
    Modify balance

    Write back modified record

- The transaction management (TM) component ensures that the database remains in a consistent (correct) state despite system failures (e.g., power failures and operating system crashes) and transaction failures

+ Concurrency Control

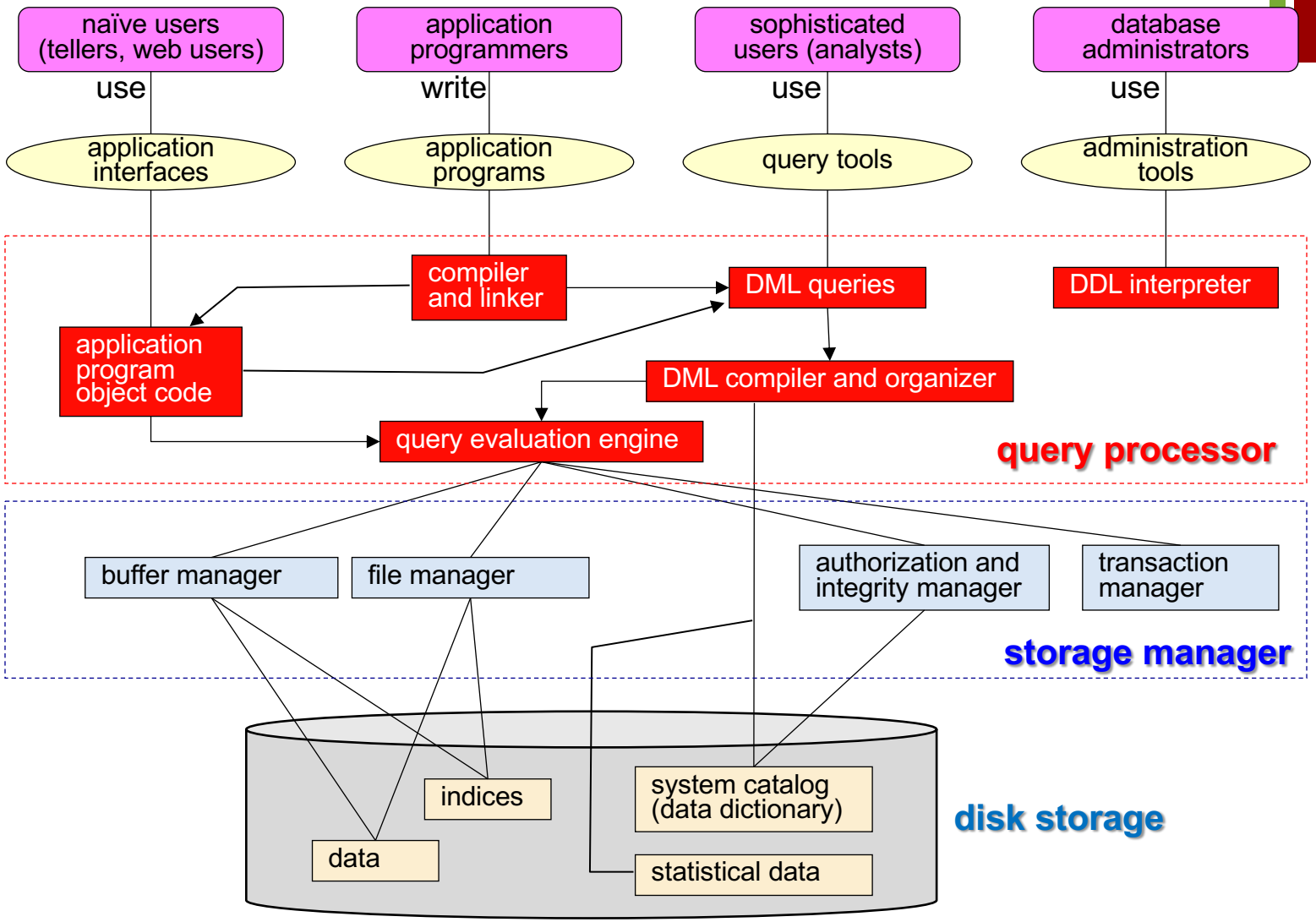
- Concurrency control (CC) manager controls the interaction among the concurrent transactions, to ensure the consistency of the database



+ Storage/Buffer Management

- The storage manager provides an interface to the buffer manager in the DBMS to access the data stored on disk
- The buffer manager is responsible for fetching data from the storage manager into main memory (the buffer) and deciding what data to keep in the buffer

+ Overall System Architecture



+ Database Users

- End Users
  - Naïve users
    - Invoke existing application programs (e.g., print monthly sales report).
    - Interact with applications through a graphical user interface (GUI).
  - Application programmers
    - Develop applications that interact with DBMS through DML calls.
  - Sophisticated users
    - Issue queries either directly using a database query language (e.g., SQL) or via tools such as data analysis software.
- Database Administrator (DBA)
  - Coordinates all activities of the database system.
    - Defines and maintains the schemas.
    - Defines and maintains the physical organization.
    - Monitors and optimizes the database performance.
    - Monitors access and grants access rights

A DBA must have a good understanding of an enterprise's information resources and needs

+ Summary

- Database management systems (DBMSs) address the limitations of file systems for managing an enterprise's data
- Data independence is fundamental to understanding a database at different abstraction levels
- Data models are the foundation for developing a database
  - The entity-relationship (E-R) model and relational model are commonly used in practice
- Database languages (DDL and DML) are an integral part of a DBMS
  - SQL is the common database language for relational DBMSs
- A DBMS provides many facilities to efficiently manage the data management and access needs of various users
  - Query processing, storage management, transaction management