



Database Management System (DBMS)

(Course code: 22CS2110R)

CO# 1 ,Session# 2:
Session Topic: Database Architecture





Session Objective

Students should be able to understand

- The database architecture
- The database users



Poll Question-01

Q: Which of the following is a function of the DBMS?

- A) Storing data
- B) Providing multi-users access control
- C) Data Integrity
- D) All of the above



Answer: D

Explanation: The purpose of creating DBMS was to store the data. The data stored in the database management system can be accessed by multiple users if the access is provided. The data stored will be accurate and complete hence providing data integrity.



Key Concepts

- ❖ Database Environment
- ❖ Typical DBMS Functionality
- ❖ Database Users
- ❖ Database Architecture

Database system environment

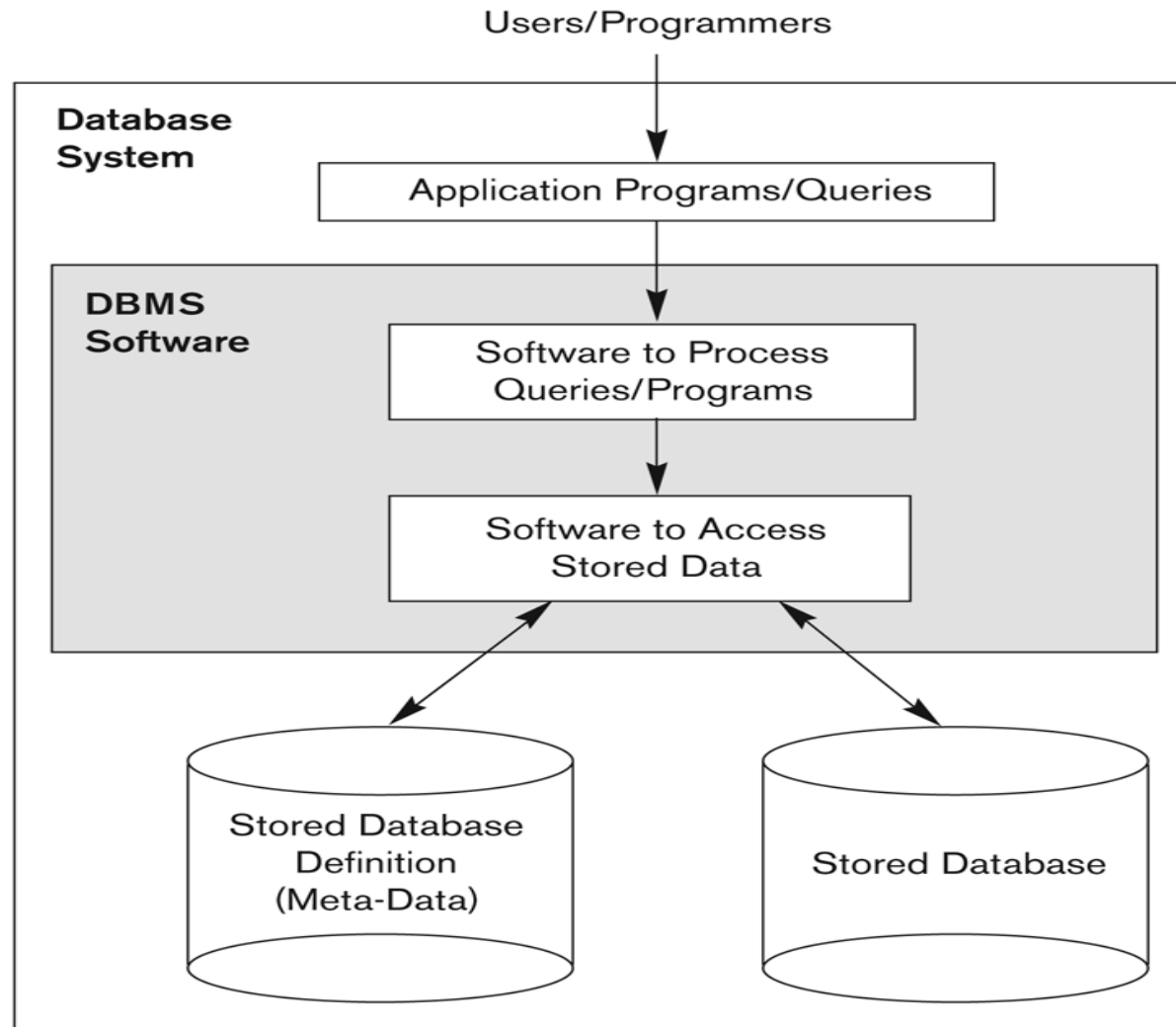


Figure 1.1
A simplified database system environment.



Database System Environment

Contd...

Database system refers to an organization of components that defines and regulate the following:

- Collection
- Storage
- Management
- Use of data within the environment



The Database System involves the following components:

- Hardware
- Software
- People
- Procedure
- Data.



Typical DBMS Functionality

- **Define a particular database** in terms of its data types, structures, and constraints
- **Construct or Load the initial database contents** on a secondary storage medium
- **Manipulating the database:**
 - **Retrieval:** Querying, generating reports
 - **Modification:** Insertions, deletions and updates to its content
 - **Accessing** the database through Web applications
- **Processing and Sharing by a set of concurrent users** and application programs – yet, keeping all data valid and consistent



Typical DBMS Functionality

- Other features:
 - **Protection or Security** measures to prevent unauthorized access
 - **“Active” processing** to take internal actions on data
 - **Presentation and Visualization** of data
 - **Maintaining the database** and associated programs over the lifetime of the database application
 - Called database, software, and system maintenance



Example of a Database (with a Conceptual Data Model)

- **Mini-world for the example:**
 - Part of a UNIVERSITY environment.
- **Some mini-world *entities*:**
 - STUDENTs
 - COURSEs
 - SECTIONs (of COURSEs)
 - (academic) DEPARTMENTs
 - INSTRUCTORs



Example of a Database (with a Conceptual Data Model)

- **Some mini-world *relationships*:**
 - 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, such as the ENTITY-RELATIONSHIP data model (see Chapters 3, 4)



Example of a simple database

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.



Main Characteristics of the Database Approach

- **Self-describing nature of a database system:**
 - A DBMS **catalog** stores the description of a particular database (e.g. data structures, types, and constraints)
 - The description is called **meta-data**.
 - This allows the DBMS software to work with different database applications.
- **Insulation between programs and data:**
 - Called **program-data independence**.
 - Allows changing data structures and storage organization without having to change the DBMS access programs.



Example of a simplified database catalog

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 enumerated 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.



Main Characteristics of the Database Approach (continued)

- **Data Abstraction:**

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

- **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.



Main Characteristics of the Database Approach (continued)

- **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.

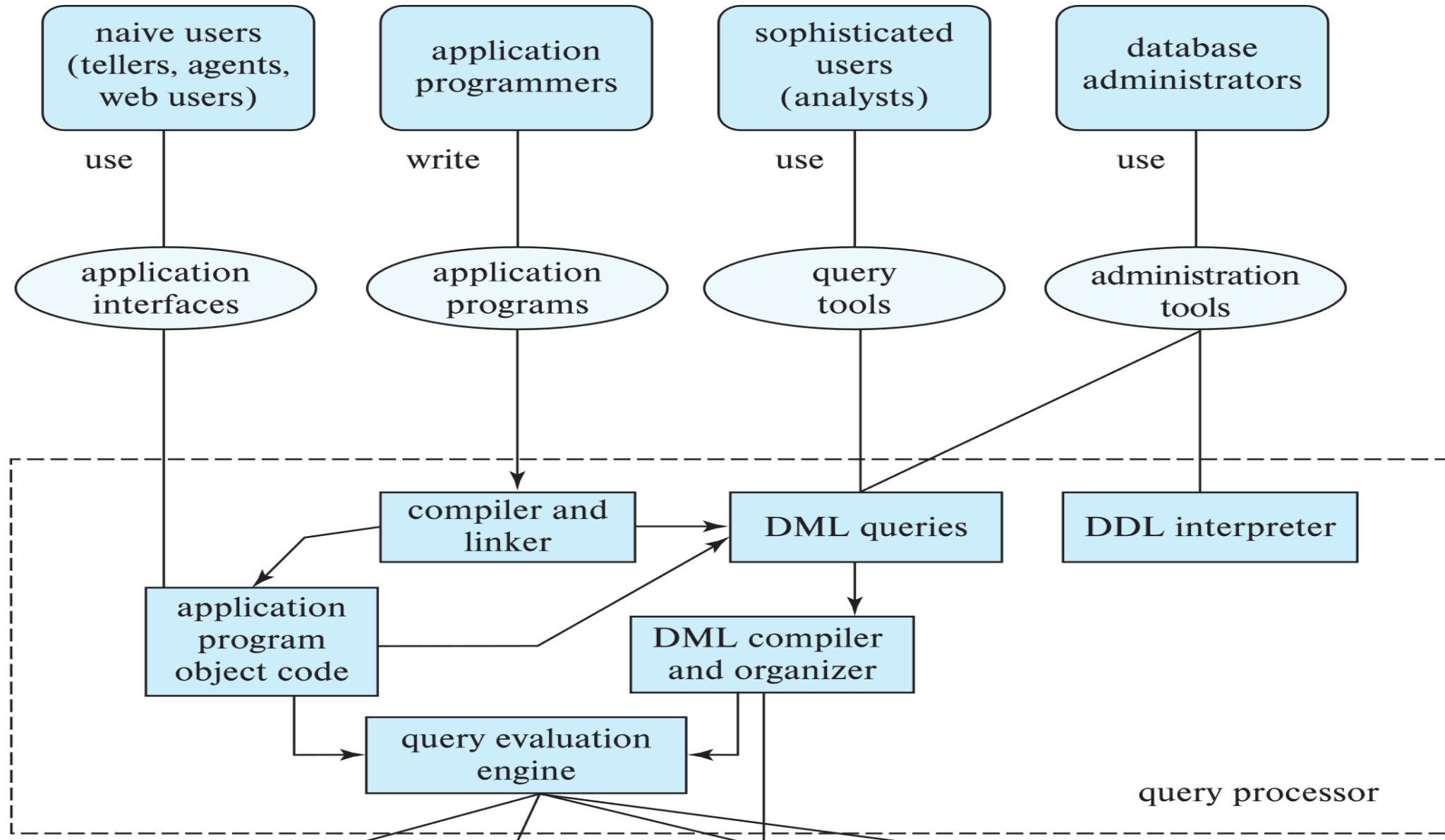


Database Users

- Users may be divided into
 - Those who actually use and control the database content, and those who design, develop and maintain database applications (called “Actors on the Scene”), and
 - Those who design and develop the DBMS software and related tools, and the computer systems operators (called “Workers Behind the Scene”).



Database Users





Database Users

- Actors on the scene
 - **Database administrators:**
 - Responsible for authorizing access to the database, for coordinating and monitoring its use, acquiring software and hardware resources, controlling its use and monitoring efficiency of operations.
 - **Database Designers:**
 - Responsible to define the content, the structure, the constraints, and functions or transactions against the database. They must communicate with the end-users and understand their needs.

Categories of End-users

- Actors on the scene (continued)
 - **End-users:** They use the data for queries, reports and some of them update the database content. End-users can be categorized into:
 - **Casual:** access database occasionally when needed
 - **Naïve** or Parametric: they make up a large section of the end-user population.
 - They use previously well-defined functions in the form of “canned transactions” against the database.
 - Examples are bank-tellers or reservation clerks who do this activity for an entire shift of operations.



Categories of End-users (continued)

- **Sophisticated:**
 - These include business analysts, scientists, engineers, others thoroughly familiar with the system capabilities.
 - Many use tools in the form of software packages that work closely with the stored database.
- **Stand-alone:**
 - Mostly maintain personal databases using ready-to-use packaged applications.
 - An example is a tax program user that creates its own internal database.
 - Another example is a user that maintains an address book



Database Administrator

A person who has central control over the system is called a **database administrator (DBA)**. Functions of a DBA include:

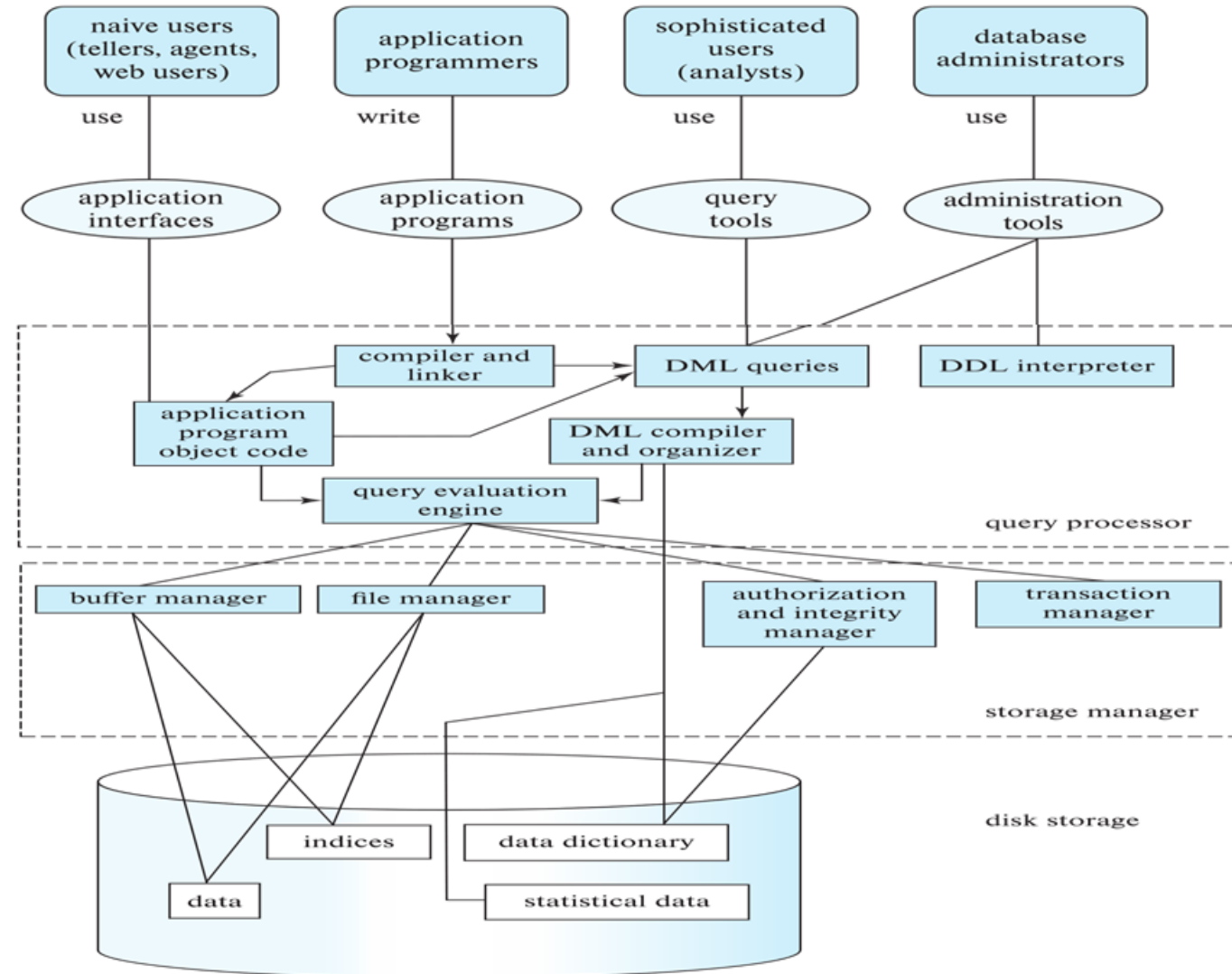
- Schema definition
- Storage structure and access-method definition
- Schema and physical-organization modification
- Granting of authorization for data access
- Routine maintenance
- Periodically backing up the database
- Ensuring that enough free disk space is available for normal operations, and upgrading disk space as required
- Monitoring jobs running on the database

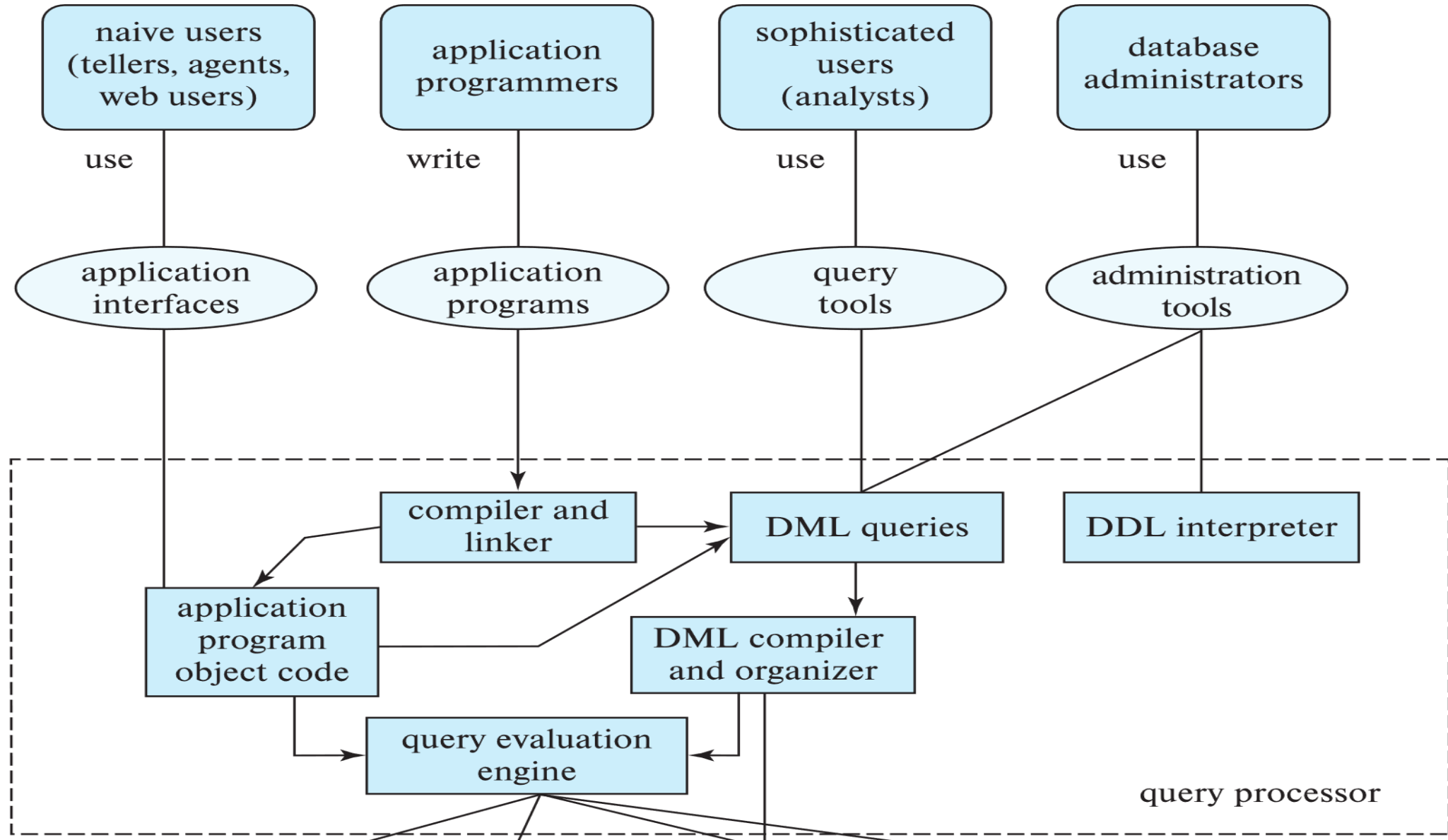


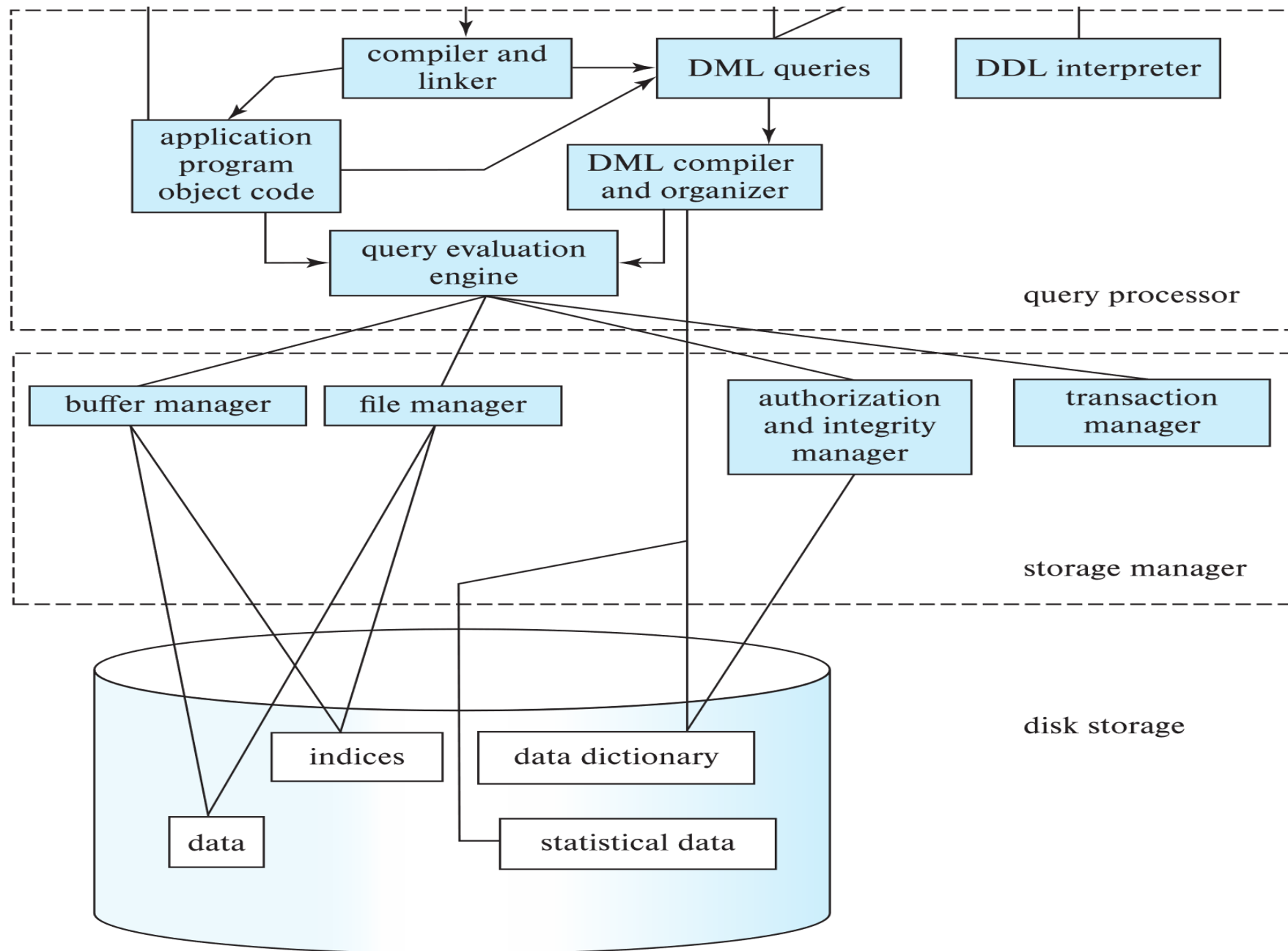
Database Architecture

- Centralized databases
 - One to a few cores, shared memory
- Client-server,
 - One server machine executes work on behalf of multiple client machines.
- Parallel databases
 - Many core shared memory
 - Shared disk
 - Shared nothing
- Distributed databases
 - Geographical distribution
 - Schema/data heterogeneity

Database Architecture (Centralized/Shared-Memory)







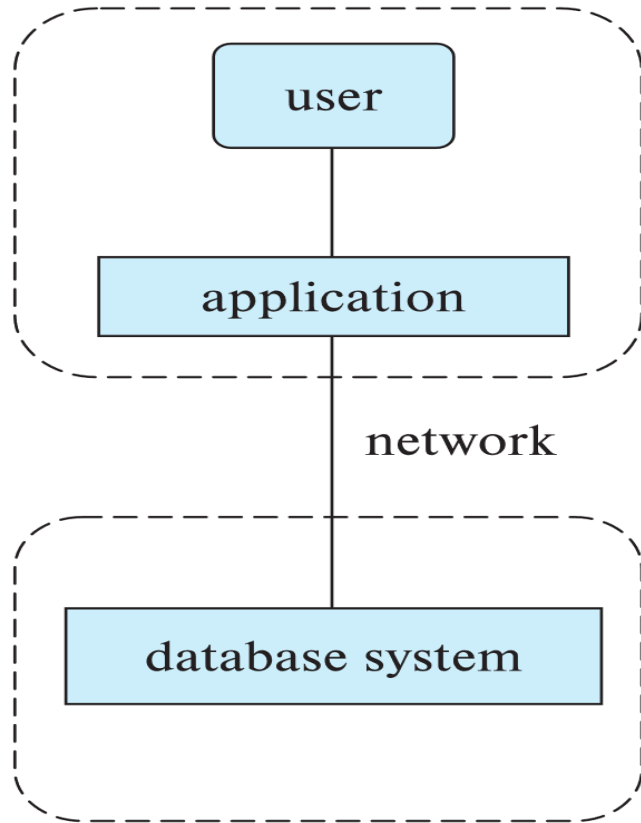


Database Applications

Database applications are usually partitioned into two or three parts

- Two-tier architecture -- the application resides at the client machine, where it invokes database system functionality at the server machine
- Three-tier architecture -- the client machine acts as a front end and does not contain any direct database calls.
 - The client end communicates with an application server, usually through a forms interface.
 - The application server in turn communicates with a database system to access data.

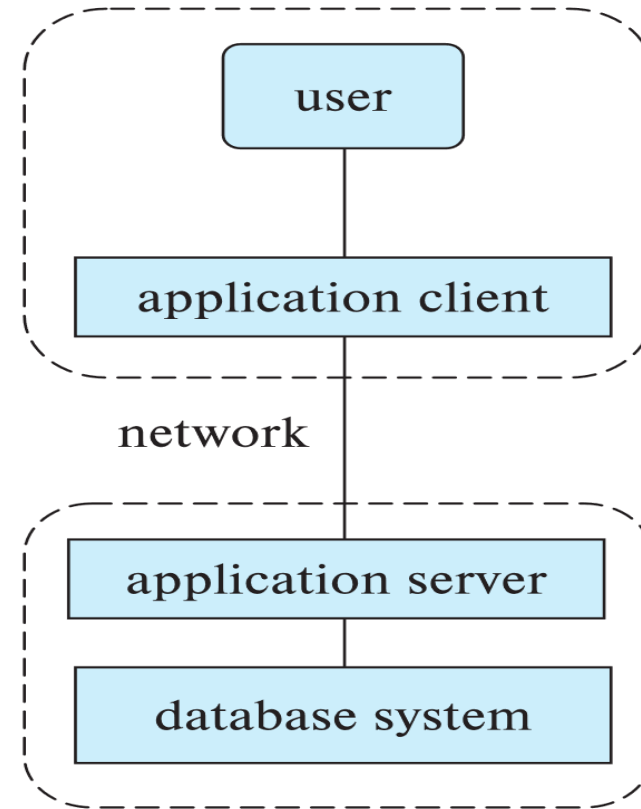
Two-tier and three-tier architectures



(a) Two-tier architecture

client

server



(b) Three-tier architecture



Advantages of Using the Database Approach

- Controlling redundancy in data storage and in development and maintenance efforts.
 - Sharing of data among multiple users.
- Restricting unauthorized access to data.
- Providing persistent storage for program Objects
 - In Object-oriented DBMSs – see Chapters 20-22
- Providing Storage Structures (e.g. indexes) for efficient Query Processing

Advantages of Using the Database Approach (continued)



- Providing backup and recovery services.
- Providing multiple interfaces to different classes of users.
- Representing complex relationships among data.
- Enforcing integrity constraints on the database.
- Drawing inferences and actions from the stored data using deductive and active rules

Additional Implications of Using the Database Approach



- Potential for enforcing standards:
 - This is very crucial for the success of database applications in large organizations. **Standards** refer to data item names, display formats, screens, report structures, meta-data (description of data), Web page layouts, etc.
- Reduced application development time:
 - Incremental time to add each new application is reduced.

Additional Implications of Using the Database Approach (continued)



- Flexibility to change data structures:
 - Database structure may evolve as new requirements are defined.
- Availability of current information:
 - Extremely important for on-line transaction systems such as airline, hotel, car reservations.
- Economies of scale:
 - Wasteful overlap of resources and personnel can be avoided by consolidating data and applications across departments.



References

1. Database System Concepts, Sixth Edition, Abraham Silberschatz, Yale University Henry, F. Korth Lehigh University, S. Sudarshan Indian Institute of Technology, Bombay.
2. Fundamentals of Database Systems, 7th Edition, RamezElmasri, University of Texas at Arlington, Shamkant B. Navathe, University of Texas at Arlington.