Chapter1

Introduction to Database Concepts

Topics to be covered:

- **Quick Recap**
- Flat File System v/s Database Systems
- Definition of DBMS
- Characteristics of Database
- Database Systems - Advantages and Disadvantages
- DBMS users
- Three Tier Architecture, Data Abstraction & Data Independence
- DBMS Architecture

Learning Outcomes:

- To understand the advantages & disadvantages of database systems
- To explain the characteristics of Database with the help of examples
- To List the main functions of a database management system
- To infer the importance of data independence
- To explain the purpose of the three levels of data abstraction: external, conceptual, and internal.
- To draw and define various modules in DBMS Architecture

Flat File System v/s Database Systems

Sr. No.	File System/ Flat File System	DBMS
1	File system is a software that manages and organizes the files in a storage medium within a computer.	DBMS is a software for managing the database.
2	Redundant data can be present in a file system.	In DBMS there is no redundant data.
3	There is less data consistency in file system.	There is more data consistency because of the process of normalization.
4	Difficulty in accessing data (Application programs need to know the location of data / files)	Data Access is easy (Location of data/ tables not required to write Queries)
5	Data is isolated – scattered across the locations (may be in different formats) without any relation	Data in tables is related to each other
6	Data integrity not possible	Data integrity can be achieved with the help of constraints.

Flat File System v/s Database Systems

Sr. No.	File System/ Flat File System	DBMS
7	Not possible to carry out atomic updates	Updates are carried out in atomic manner
8.	Concurrent access of data/ sharing between multiple users not possible	DBMS has concurrency control mechanisms for sharing of data between multiple users
9	No security features provided by this system	Backup and Recovery features are provided. Different levels of privileges can be granted to users.

Database Management System (DBMS)

The **DBMS** is a ***general-purpose*** *software system* that facilitates the processes of ***defining***, ***constructing***, ***manipulating***, *and* ***sharing*** databases among various users and applications.

It is a collection of interrelated data & a set of programs to access those data.

Characteristics of Database

1. **Self describing nature of Database**

Characteristics of Database

1. **Self describing nature of Database**

   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.

Example

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

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.

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.

Characteristics of Database

1. **Self describing nature of Database**

   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.

2. **Program-data independence**

Characteristics of Database

1. **Self describing nature of Database**

   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.

2. **Program-data independence**

   Insulation between program and data. Allows changing data structures and storage organization without having to change the DBMS access programs

Characteristics of Database

1. **Self describing nature of Database**

   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.

2. **Program-data independence**

   Insulation between program and data. Allows changing data structures and storage organization without having to change the DBMS access programs

3. **Support of multiple views of the data**

Characteristics of Database

1. **Self describing nature of Database**

   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.

2. **Program-data independence**

   Insulation between program and data. Allows changing data structures and storage organization without having to change the DBMS access programs

3. **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.

Characteristics of Database

1. **Self describing nature of Database**

   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.

2. **Program-data independence**

   Insulation between program and data. Allows changing data structures and storage organization without having to change the DBMS access programs

3. **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.

4. **Sharing of data and multi-user transaction processing**:

Characteristics of Database

1. **Self describing nature of Database**

   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.

2. **Program-data independence**

   Insulation between program and data. Allows changing data structures and storage organization without having to change the DBMS access programs

3. **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.

4. **Sharing of data and multi-user transaction processing**: Allowing a set of concurrent users to retrieve from and to update the database. Concurrent processing , OLTP

Example

A database in a DBMS could be viewed by lots of different people with different responsibilities.

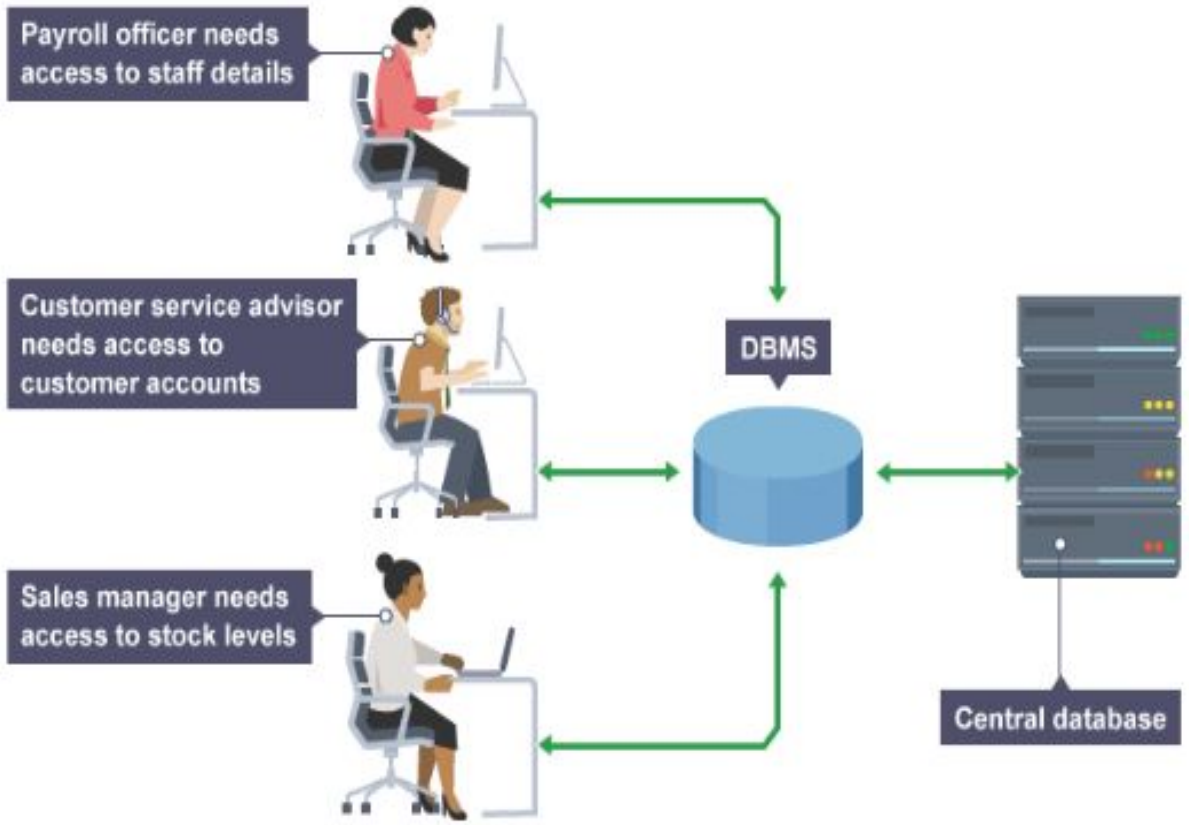


Figure 1.1: Empolyees are accessing Data through DBMS

Database - Advantages

Database - Advantages

- Reduced data redundancy

- Reduced updating errors and increased consistency

- Greater data integrity and independence from applications programs

- Improved data access to users through use of query languages

- Improved data security

- Reduced data entry, storage, and retrieval costs

- Facilitated development of new applications program

Database - Disadvantages

Database - Disadvantages

- Database systems are complex, difficult and time consuming to design.

- Start-up cost of Hardware & Software

- Cost of Data Conversion

- Cost of Staff Training

- Appointing Technical Staff

- Database Failures

Database users

- **Actors on the Scene**
  a) Database Administrator

Workers behind the Scene

Database users

- **Actors on the Scene**
  a) Database Administrator

Database administrator's duties

- Schema definition
- Storage structure and access method definition
- Schema and physical organization modification
- Granting user authority to access the database
- Routine maintenance
  - periodically backing up
  - Ensuring enough free disks available
  - monitoring jobs running on the database

Database users

- **Actors on the Scene**
  a) Database Administrator
  b) Database Designers – data, data structures, views
  c) End users : querying, updating, and generating reports
     - naive /parametric : Canned transactions e.g. – Bank tellers
     - Casual : Middle level / High level managers- **Query language**
     - sophisticated : engineers, scientists, business analysts
     - Standalone: personal database, tax mgmt. system
     - System Analysts and Application Programmers: for naïve and casual users make appln. Programs

Database users

- **Actors on the Scene**
  a) Database Administrator
  b) Database Designers – data, data structures, views
  c) End users : querying, updating, and generating reports
     - naive /parametric : Canned transactions e.g. – Bank tellers
     - Casual : Middle level / High level managers- **Query language**
     - sophisticated : engineers, scientists, business analysts
     - Standalone: personal database, tax mgmt. system
     - System Analysts and Application Programmers: for naïve and casual users make appln. Programs

Workers behind the Scene

d) DBMS system designers & implementer (DBMS system architecture modules)
e) Tool developers: database design, performance monitoring, graphical interfaces, prototyping, simulation & test data generation.
f) Operators & maintenance personnel : H/W , S/W environment

Data Abstraction
View of DATA

Physical level: describes how a record (e.g., customer) is stored.

Logical level: describes what data stored in database,
and the relationships among the data.

```
type customer = record
        customer_id : string;
        customer_name : string;
        customer_street : string;
        customer_city : string;
                end;
```

View level: application programs hide details of data types. Views can also hide information (such as an employee's salary) for security purposes.

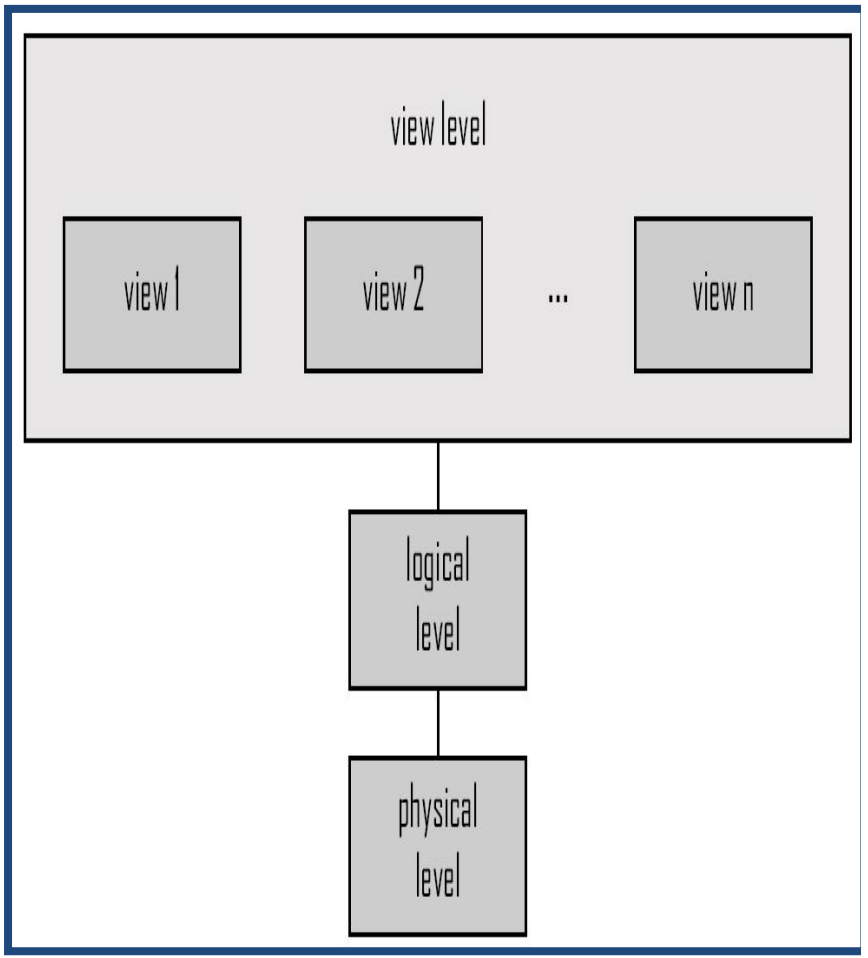


Fig: Levels of Abstraction in a DBMS

Instances and Schemas

- Database changes over time

- **Instance** – the actual content of the database at a particular point in time
  -- also known as "Database state" or "Snapshot "
- **Schema** – the logical structure of the database
  e.g., the database consists of information about a set of customers and accounts and the relationship between them

  Physical schema: database design at the physical level
  Logical schema: database design at the logical level
  Sub schema: database design at the view level

- **Distinguish between – description of database & database itself**

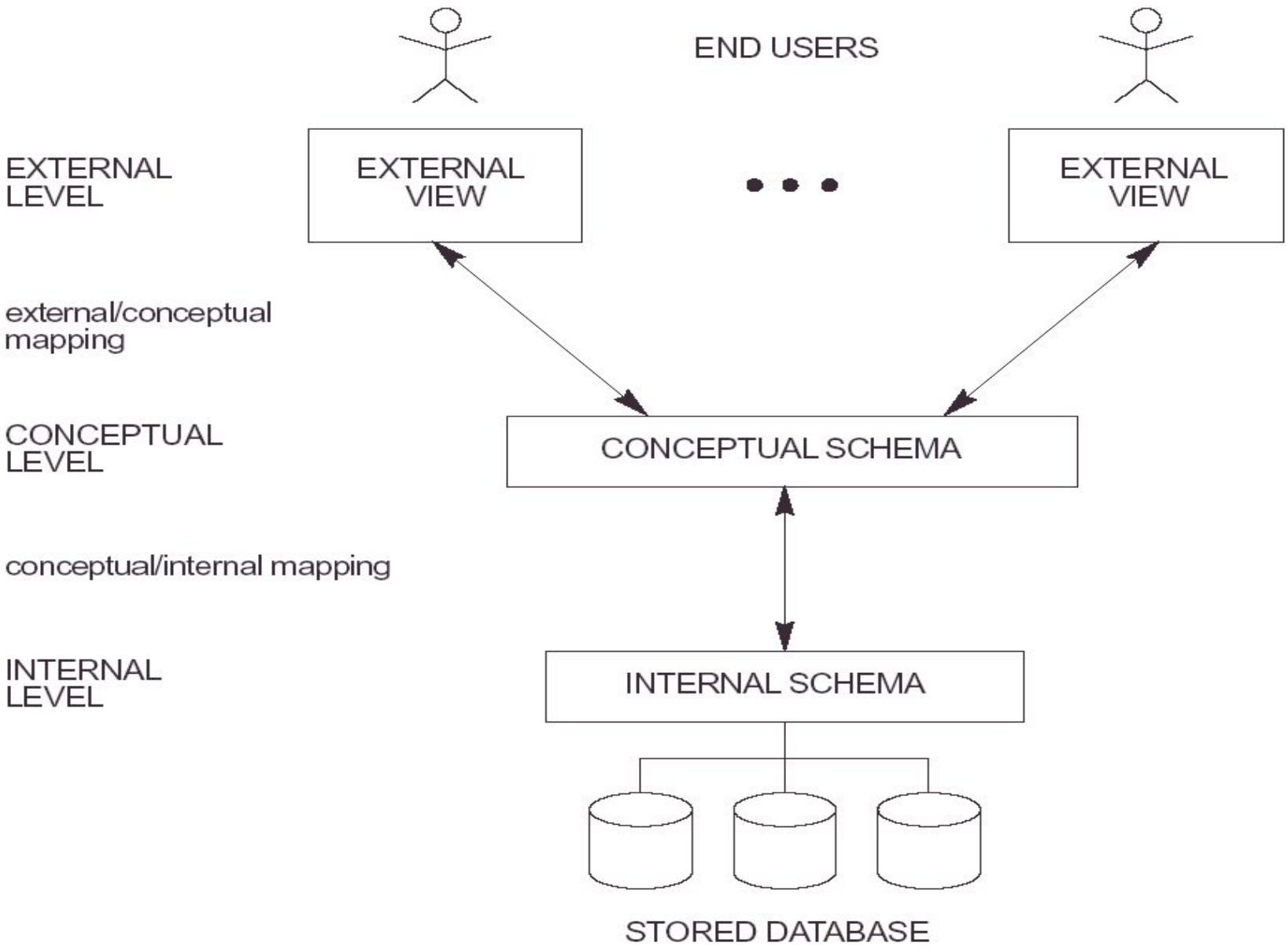
The Three-Schema Architecture

- Importance of using DB approach
  - insulation of programs and data
  - support of multiple user views
  - use of a catalog to store the database description (schema).
- The aim is to separate the user application and physical DB
- schema can be defined into three levels:
  - The internal level has an internal schema
    - describes the physical storage structure of the database.
    - uses a physical data model

The Three-Schema Architecture

- The conceptual level has a conceptual schema describing the structure of the whole database for a community of users.
  - It hides the details of physical storage structures and concentrates on describing entities, data types, relationships, user operations, and constraints.
  - A high-level data model or an implementation data model can be used at this level.
- The external or view level includes a number of external schemas or user views describing the part of the db that a particular user group is interested in and hides the rest of the db from that user group.
  - A high-level data model or an implementation data model can be used at this level.

Illustrating the three-schema architecture.



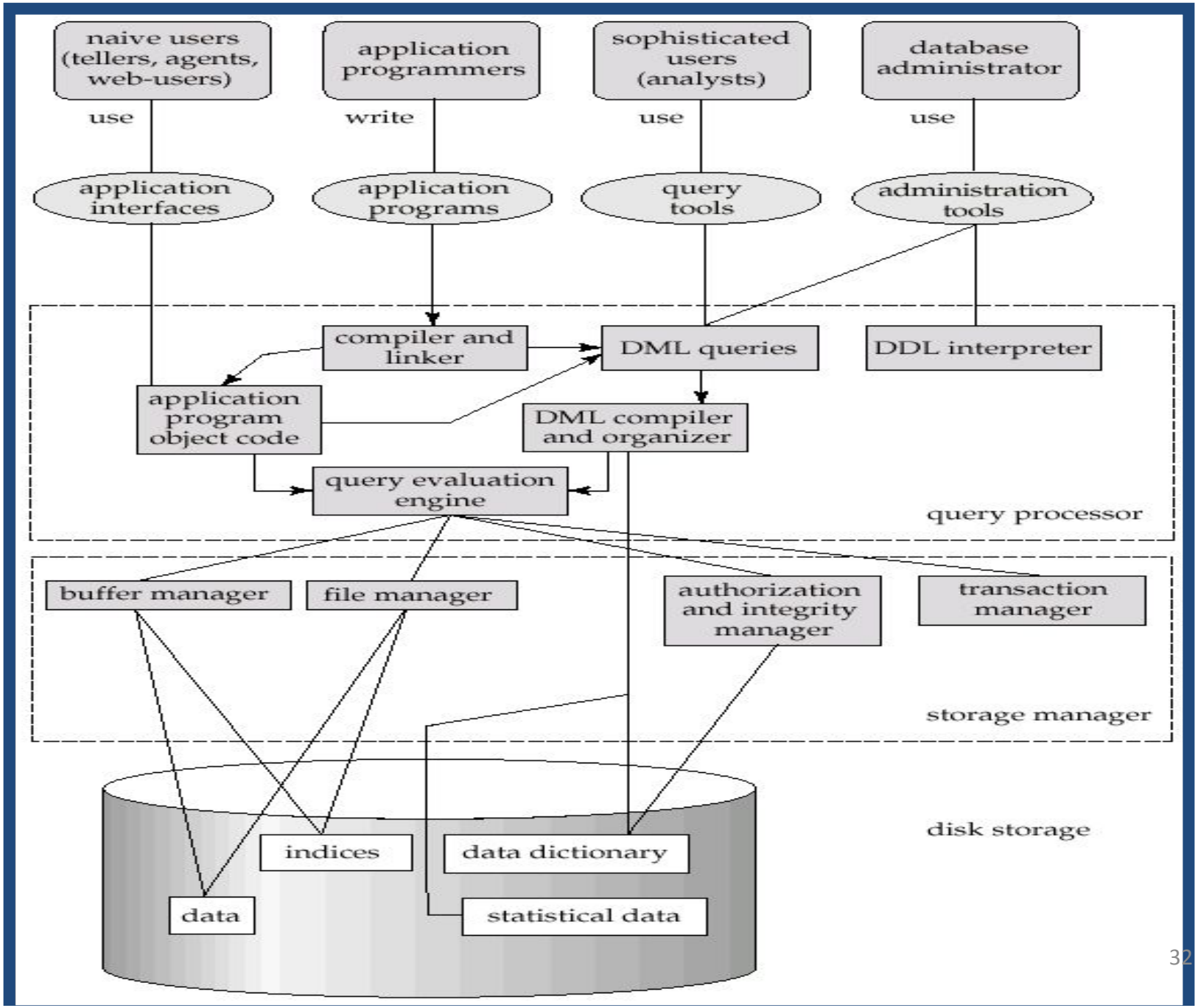
Data Independence

- Is the capacity to change the schema at one level of a database system without having to change the schema at the next higher level.
- Logical data independence: capacity to change the conceptual schema without having to change external schemas or application programs.
- Physical data independence: capacity to change the internal schema without having to change the conceptual (or external) schemas

Database Management System Structure

- Storage management
- Query processing
- Transaction processing

Overall System Structure



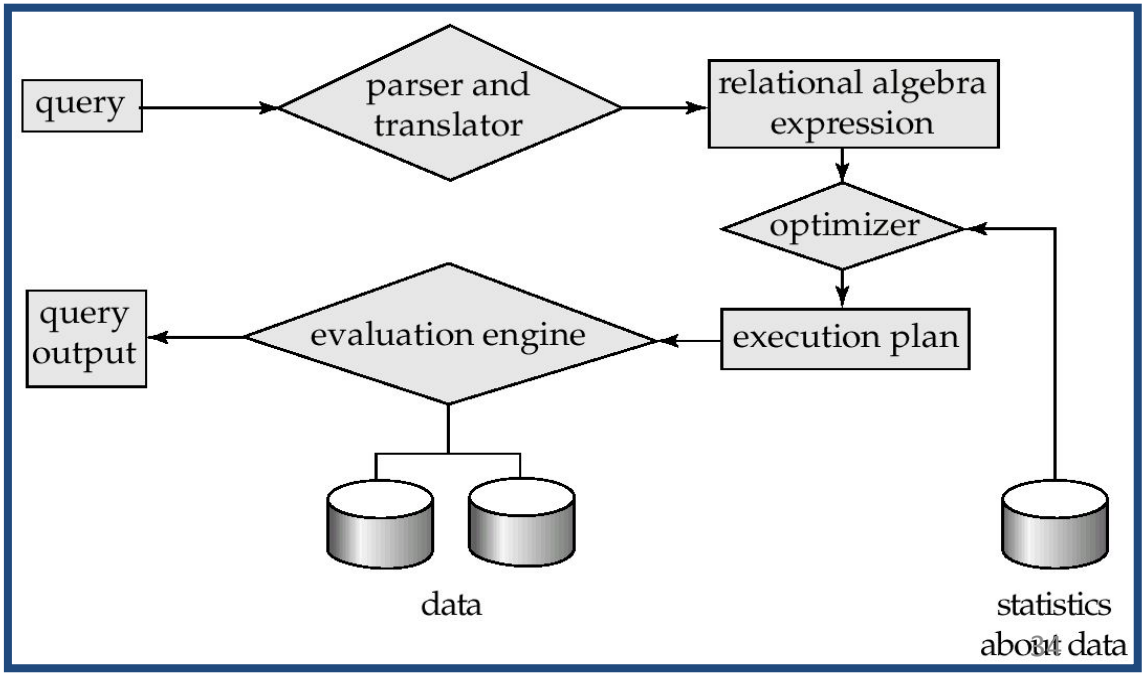
Storage Management

- **Storage manager** is a program module that provides the interface between the low-level data stored in the database and the application programs and queries submitted to the system.
- It has following components:

- Authorization and integrity manager
- Transaction manager
- File manager
- Buffer manager

Query Processing

1. Parsing and translation
2. Optimization
3. Evaluation



Query Processing (Cont.)

- Alternative ways of evaluating a given query
  - Equivalent expressions
  - Different algorithms for each operation
- Cost difference between a good and a bad way of evaluating a query can be enormous
- Need to estimate the cost of operations
  - Depends critically on statistical information about relations which the database must maintain
  - Need to estimate statistics for intermediate results to compute cost of complex expressions

Transaction Management

- A **transaction** is a collection of operations that performs a single logical function in a database application
- **Transaction-management 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 manager** controls the interaction among the concurrent transactions, to ensure the consistency of the database.