Database Management System

Year: III
Part :II

Theory: 80+20
Practical: 50

References:
H.F Korth and A. Silberschatz, “Database System Concepts”, McGraw Hill

Evaluation Scheme

Unit	Hour	Distribution
1	3	4
2	7	12
3	7	12
4	6	12
5	4	8
6	4	8
7	6	12
8	4	6
9	4	6
Total	45	80

1.0 Introduction

1.1 Concepts and Application

Data:

- any raw facts of figures collected from different sources like statistical reports, queries, questionnaires etc.
- They can be numbers, letters, words of symbols which may not have any significant meaning when they stand alone. Eg. Rita, 15000, manager

Information:

- is a processed data using suitable database software.
- give some meaning.

Name	Post	Salary
Rita	Manager	15000

Database:

- A collection of data related to a particular subject or purpose.
- Organization of data in easily accessible manner.
- A user can retrieve information from database in effective and efficient manner

Eg. Telephone directory, marks ledger, dictionary, flight schedule etc.

DBMS (Database Management System)

- a collection of interrelated data and a set of program to access those data.
- primary goal is to provide a way to store and retrieve database information in both efficient and convenient way.
- DBMS must ensure the safety of the information stored, despite system crashes or attempts of unauthorized access of information.
- ensure sharing of information among several users without anomalous results.

Application- Database System

- Banking: all 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 touch all aspects of our lives

History/ Evolution of DB System

Assignment no. 1

Traditional File Based System

- File-based systems were an early attempt to computerize the manual filing system.
- File-based system is a collection of application programs that perform services for the end-users.
- Each program defines and manages its data.
- Early attempts to computerize the manual filing system that are familiar with us

file-based systems proved to be a great improvement over manual filing system

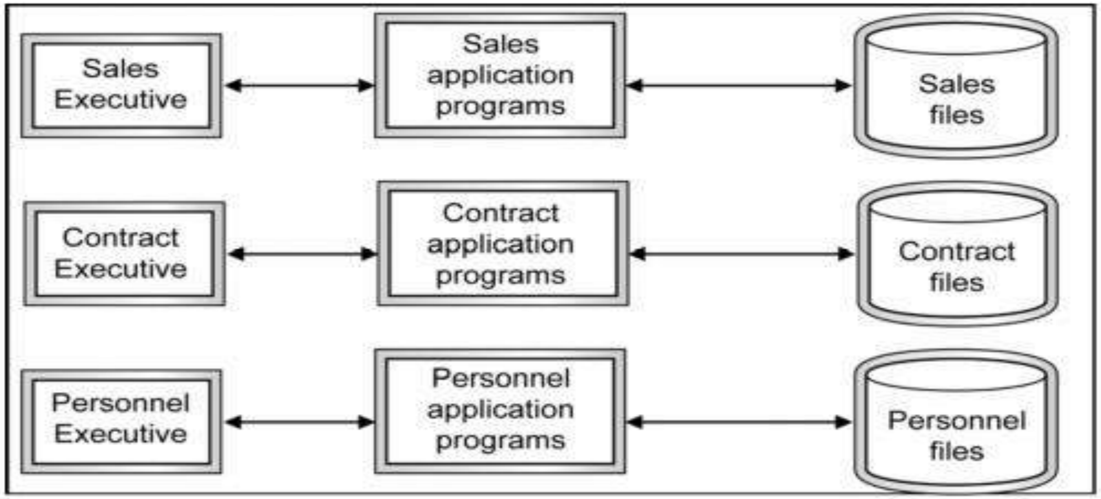


Figure 1.1: File-based system for Make-Believe real estate company

Limitations of File based system

- **Separation and isolation of data**

-When data is isolated in separate files, it is more difficult for us to access data that should be available.

- The application programmer is required to synchronize the processing of two or more files to ensure the correct data is extracted.

- **Duplication of data**

- When employing the decentralized file-based approach, the uncontrolled duplication of data is occurred. Uncontrolled duplication of data is undesirable because:

i. Duplication is wasteful

ii. Duplication can lead to loss of data integrity

Incompatible file formats

file structure provided in one programming language such as direct file, indexed-sequential file which is available in COBOL programming, may be different from the structure generated by other programming language such as C. The direct incompatibility makes them difficult to process jointly.

Fixed queries / proliferation of application programs

File-based systems are very dependent upon the application programmer.

-Normally, a fixed format query or report can only be entertained and no facility for ad-hoc queries if offered.

- **Data dependence**

Using file-based system, the physical structure and storage of the data files and records are defined in the application program code. This characteristic is known as program-data dependence.

Making changes to an existing structure are rather difficult and will lead to a modification of program. Such maintenance activities are time-consuming and subject to error.

Solution???

Database Approach
Advantages

Control of data redundancy

The database approach attempts to eliminate the redundancy by integrating the file.

Data consistency

- By eliminating or controlling redundancy, the database approach reduces the risk of inconsistencies occurring.
- It ensures all copies of the data are kept consistent.

Sharing of data

- Database belongs to the entire organization and can be shared by all authorized users.

More information from the same amount of data

- With the integration of the operated data in the database approach, it may be possible to derive additional information for the same data.

Improved data integrity

- Database integrity provides the validity and consistency of stored data.
- Integrity is usually expressed in terms of constraints, which are consistency rules that the database is not permitted to violate.

Improved security

Database approach provides a protection of the data from the unauthorized users

Enforcement of standards

The integration of the database enforces the necessary standards including data formats, naming conventions, documentation standards, update procedures and access rules.

Improved data accessibility and responsiveness

- data accessing can be crossed departmental boundaries. This feature provides more functionality and better services to the users.

Increased productivity

- provides all the low-level file-handling routines.
- The provision of these functions allows the programmer to concentrate more on the specific functionality required by the users

Increased concurrency

- Database can manage concurrent data access effectively.
- It ensures no interference between users that would not result any loss of information nor loss of integrity.

Improved backing and recovery services

- Modern database management system provides facilities to minimize the amount of processing that can be lost following a failure by using the transaction approach.

Limitations of Database Approach

Complexity

- Database management system is an extremely complex piece of software.
- training for the administrators, designers and users is required.

Size

- consumes a substantial amount of main memory as well as a large number amount of disk space in order to make it run efficiently.

Cost of DBMS

- A multi-user database management system may be very expensive.
- Even after the installation, there is a high recurrent annual maintenance cost on the software.

Cost of conversion

- When moving from a file-base system to a database system, the company is required to have additional expenses on hardware acquisition and training cost.

Performance

- some applications may not run as fast as before.

Higher impact of a failure

- increases the vulnerability of the system due to the centralization.
- As all users and applications reply on the database availability, the failure of any component can bring operations to a halt and affect the services to the customer seriously.

Database Approach

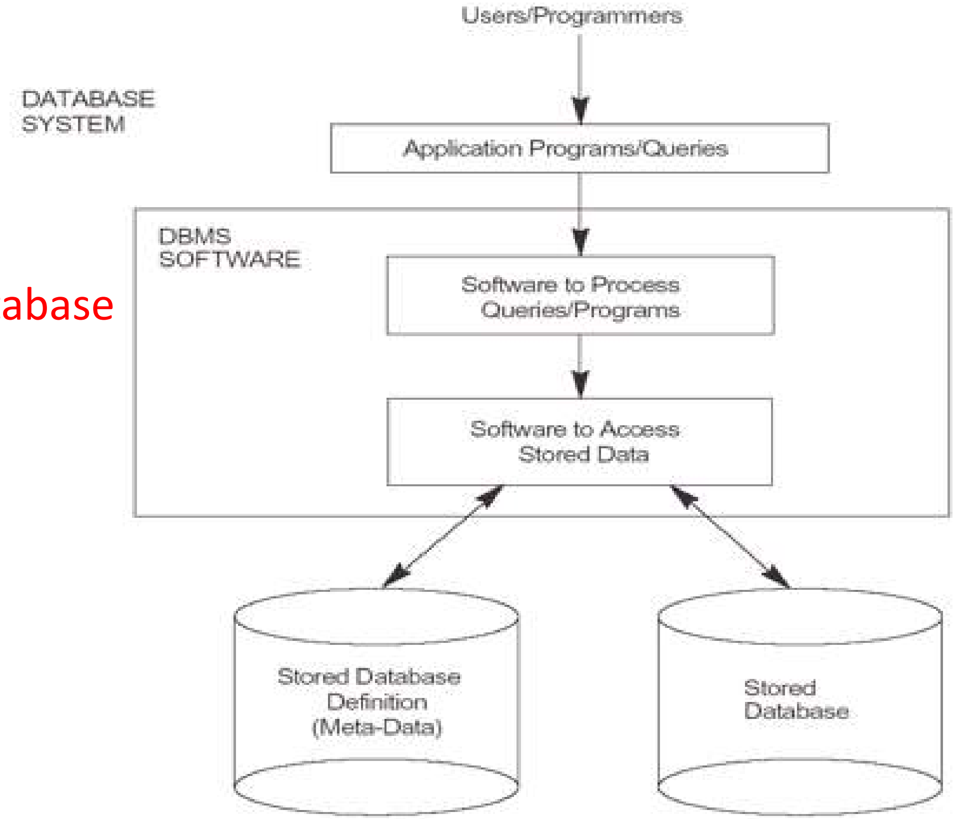

Users/Programmers
DATABASE SYSTEM
Application Programs/Queries
DBMS SOFTWARE
Software to Process Queries/Programs
Software to Access Stored Data
Stored Database Definition (Meta-Data)
Stored Database

Figure: A simplified database system environment

Data Abstraction (Complexity Hiding)

- major purpose of a database system is to provide users with an abstract view of the system.
- The system hides certain details of how data is stored and created and maintained. Complexity should be hidden from database users.
- User of an object see only the external definition and are unware of how object are defined and how it function.

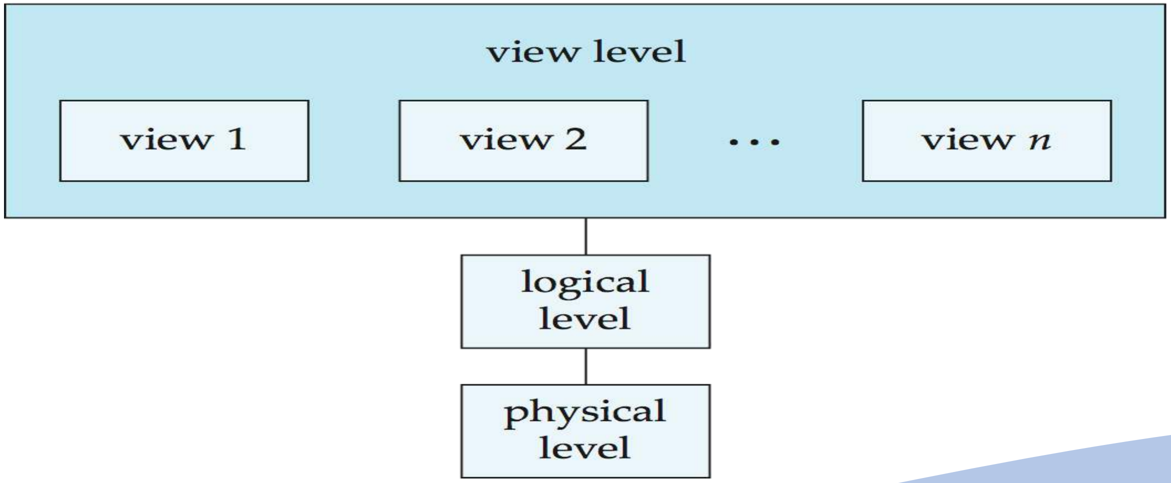


Fig: an architecture for database system (three level of data abstraction)

Physical Level:

- Describes how the data are stored.
- Lowest level of abstraction.
- Complex low-level structures are described in detail.
- E.g. index, B-tree, hashing, record description for storage, data compression and encryption.

Logical Level:

- Next highest level of abstraction.
- Describes what data are stored.
- Describes the relationships among data.
- Database administrator level(decides what information to keep in DB .

E.g. all entities, their attributes & relationships, constraints on data, security etc.

View Level:

- Highest level.
- Describes part of the database for a particular group of users.
- Can be many different views of a database.
- E.g. tellers in a bank get a view of customer accounts, but not of payroll data.

```
Create table student
( RN int,
   Name varchar(50),
   Address varchar (100),
   DOB datetime
   primary key(RN)
);
```

Physical level: ?

Logical level :?

View level:?

Instances and Schema

Instances:

- used to describe a complete database environment, including the RDBMS software, table structure, stored procedures and other functionality.
- It is most commonly used when administrators describe multiple instances of the same database.
- The information in a database at a particular point in time is called an instance of the database.

Schema:

- Database schema is a description of the structure of a database or a directory (file systems).

There are 3 kind of schemas in Databases -

* conceptual schema which is a map of concepts and their relationships

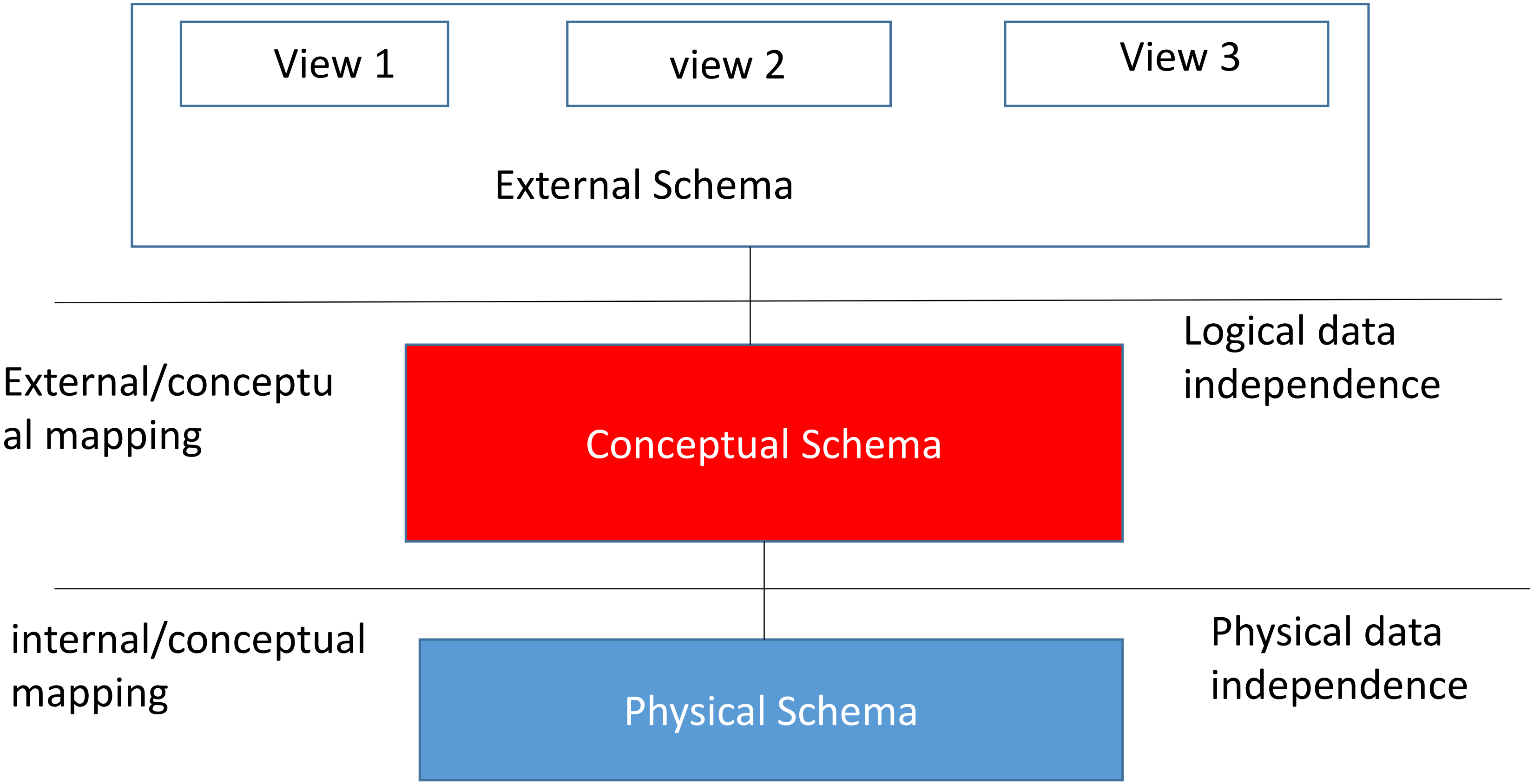
* logical schema which is a map of entities and their attributes and relations

* physical schema which is a particular implementation of a logical schema

Analogy with programming languages:

Data type definition - scheme

Value of a variable - instance



View 1

view 2

View 3

External Schema

External/conceptual mapping

Conceptual Schema

Logical data independence

internal/conceptual mapping

Physical Schema

Physical data independence

Fig: Three schema DBMS architecture

Mapping

- Relationship between the three types of schema. DBMS must also check schemas for consistency.
- DBMS must check that each external schema is derivable from conceptual schema and it must use the information in the conceptual schema to map between external schema and physical schema.

Types:

External/ conceptual schema:

- relates external schema and conceptual schema
- Enables the DBMS to map in the user’s view on to the relevant part of the conceptual schema

Conceptual/internal mapping:

- relates internal schema and conceptual schema
- Allows any differences in the entity names, attributes name and orders to be resolved

Data Independence

The ability to modify a scheme definition in one level without affecting a scheme definition in a higher level is called data independence.

There are two kinds:

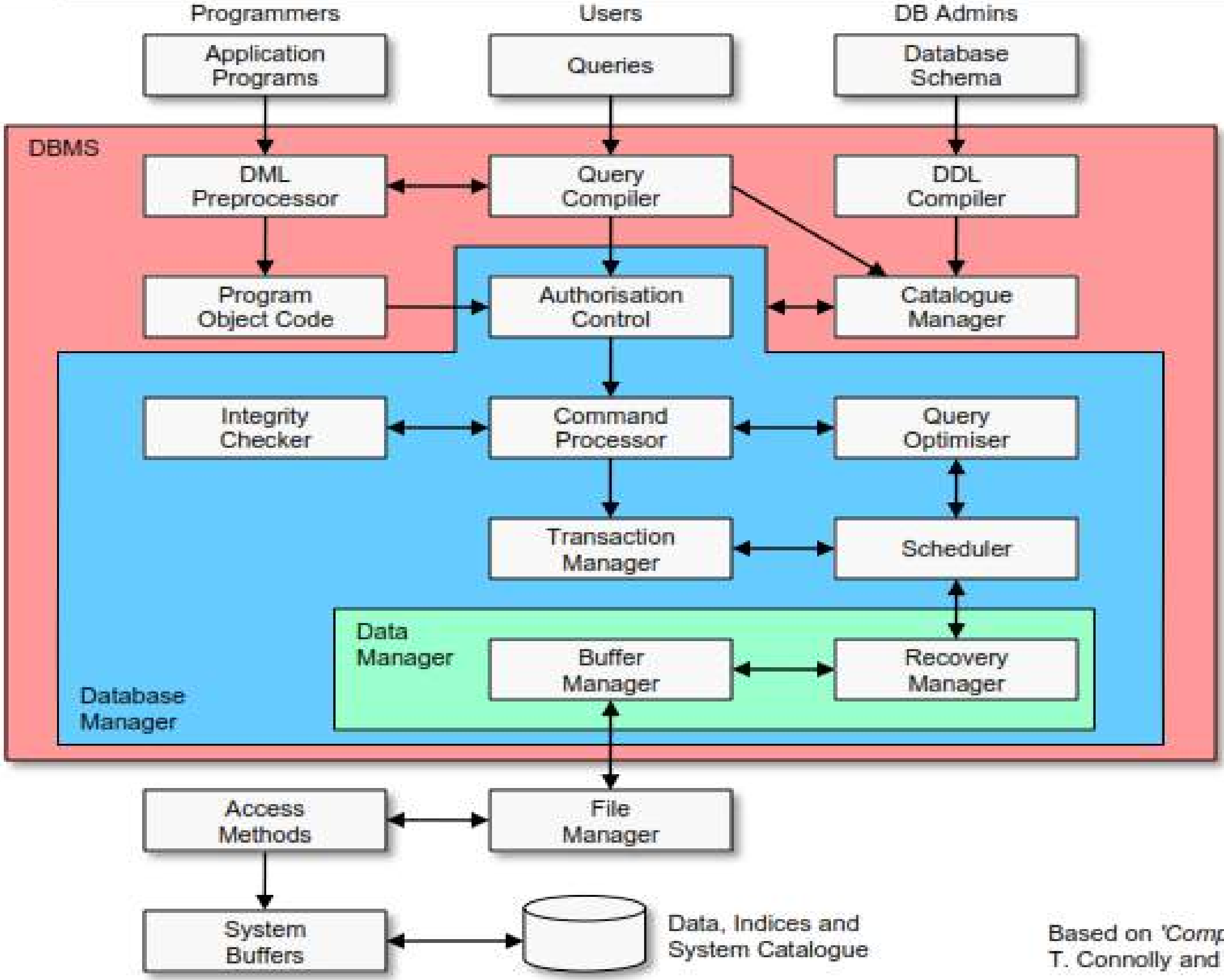
Physical data independence

- The ability to modify the physical scheme without causing application programs to be rewritten
- Modifications at this level are usually to improve performance

Logical data independence

- The ability to modify the conceptual scheme without causing application programs to be rewritten
- Usually done when logical structure of database is altered
- It is harder to achieve as the application programs are usually heavily dependent on the logical structure of the data.

DBMS Components



Based on *'Components of a DBMS'*, Database Systems, T. Connolly and C. Begg, Addison-Wesley 2010

DML preprocessor

- transforms embedded SQL statements into statements of the host language
- interacts with the query compiler to generate the appropriate host language code

Query compiler

- transforms queries into a set of low-level instructions (query plan)which are forwarded to the database manager component

DDL compiler

- converts a set of DDL statements into a set of tables
- tables and metadata are stored in the system catalogue (catalogue manager)

Catalogue manager

- provides access and manages the system catalogue
- used by most DBMS components

Database manager

- processes user-submitted queries
- interfaces with application programs
- contains a set of components
  - query optimiser
  - transaction manager

Authorisation control

- checks whether as user has the necessary rights to execute a specific operation

Command processor

- executes the steps of a given query plan handed over by the authorisation control component

Integrity checker

- ensures that the operation is not going to violate any integrity constraints (e.g. key constraints)

Query optimiser

- computes an optimal query execution strategy
- transforms the initial query plan into the best available sequence of operations on the given data

Transaction manager

- processes any transaction-specific operations

Scheduler

- manages the relative order in which transaction operations are executed

Recovery manager

- deals with commits and aborts of transactions
- ensures that the database remains in a consistent state in case of failures

Buffer manager

- transfers data between main memory and secondary storage

Participation of people in DB Environment

Data Administrator

- Responsible for the management of data resources including database planning, development and maintenance of standards, policies and procedures and conceptual/logical design.
- Advices senior managers – ensures database development supports corporate objectives

Database Administrator

- Responsible for physical realization of database including physical database design and implementation, security, integrity control, maintenance of operational system
- Ensures satisfactory performance of application for users.

Database Designer

❑ Logical DB Designer

- Identifying entities and attributes
- Relationship between data and constraints
- Implements business rules

❑ Physical DB Designer

- Designing security measures on data
- Selecting storage structure and access method
- Mapping logical database design into a set of tables and integrity constraint

Application Developers

- Develop application programs and implement database to provide required functionality for the end users

End users

- Clients
- Information is served to end users

❑Naïve User: -unware of DBMS

- access DB via application program

- use menu to operate commands

❑Sophisticated user: - are familiar with structure of database

-may use SQL to perform required operation

Application Architecture

- Application architecture is the organizational design of an entire software application, including all subcomponents and external applications interchanges.
- There are several design patterns that are used to define application architecture. These patterns help to communicate how an application will complete the necessary business processes as defined in the system requirements.

1-Tier Architecture

- simple form of standalone application architecture where everything resides in a single program. 1-Tier Architecture is the simplest, single tier on single user, and is the equivalent of running an application on a personal computer.
- entire required components to run the application are located within it. User interface, business logic, and data storage are all located on the same machine.
- easiest to design, but the least scalable. Because they are not part of a network, they are useless for designing web applications

2-Tier Architecture

- 2-Tier Architectures supply a basic network between a client and a server. For example, the basic web model is a 2-Tier Architecture. A web browser makes a request from a web server, which then processes the request and returns the desired response, in this case, web pages.
- improves scalability and divides the user interface from the data layers.

- **3-Tier Architecture**
- 3-Tier Architecture is most commonly used to build web applications. In this model, the browser acts like a client, middleware or an application server contains the business logic, and database servers handle data functions.
- This approach separates business logic from display and data. But, it does not specialize functional layers. Its fine for prototypical or very simple web applications, but it doesn’t measure up to the complexity demanded of web applications.

N-Tier Architecture

- N-Tier (Multi-Tier) architecture is a client-server architecture in which, the presentation, the application processing and the data management are logically separate processes. For example, an application that uses middleware to service data requests between a user and a database employs multi-tier architecture.
- The most widespread use of "multi-tier architecture" refers to three-tier architecture.

Database Languages

DDL(Data Definition Language)

CREATE, ALTER DROP

DML(Data Manipulation Language)

SELECT, INSERT, UPDATE, DELETE

DCL(Data Control Language)

GRANT, REVOKE, DENY

Multi user DBMS architecture

Assignment 2:

Teleprocessing

File-server architecture

Two-tier client server architecture

Three-tier client server architecture

Transaction processing monitors (TPM)