Database Management System (DBMS) ITC303

Subject Incharge

Priyanka Patil

Assistant Professor

email: priyankapatil@sfit.ac.in

Module I

Database System Concepts and Architecture

Content

- Introduction
- Characteristics of databases
- File system V/s Database system
- Data abstraction and data Independence
- DBMS system architecture
- Database Administrator

Concept of a database

- **Data**: Known facts that can be recorded and have an implicit meaning.
- **Database**: A collection of related data.
- **Mini-world**: Some part of the real world about which data is stored in a database. For example, student grades and transcripts at a university.
- **Database Management System (DBMS)**: A software package/ system to facilitate the creation and maintenance of a computerized database.
- **Database System**: The DBMS software together with the data itself. Sometimes, the applications are also included.

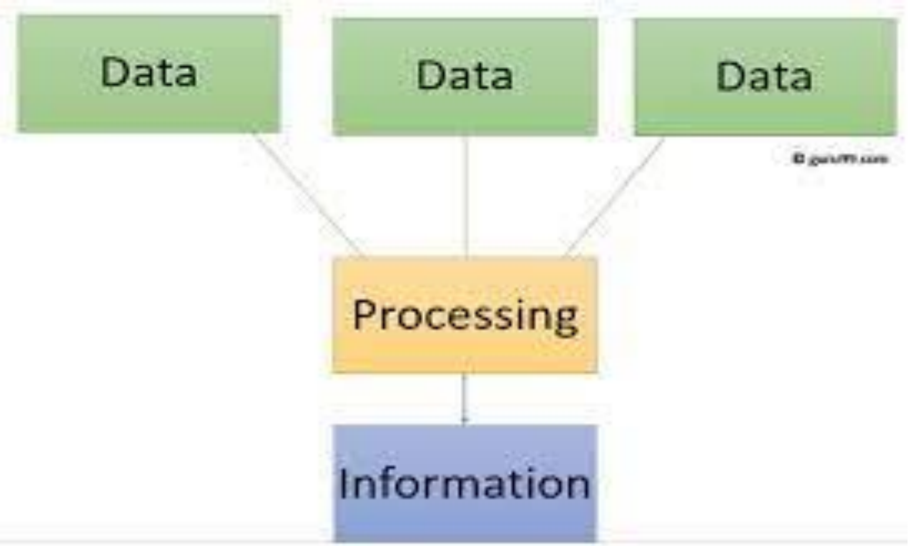
What is Data?

Data can be facts related to any object in consideration.

For example,
Name of person, age, height, weight, etc. are some data related to person.
A picture, image, file, pdf, etc. can also be considered data.

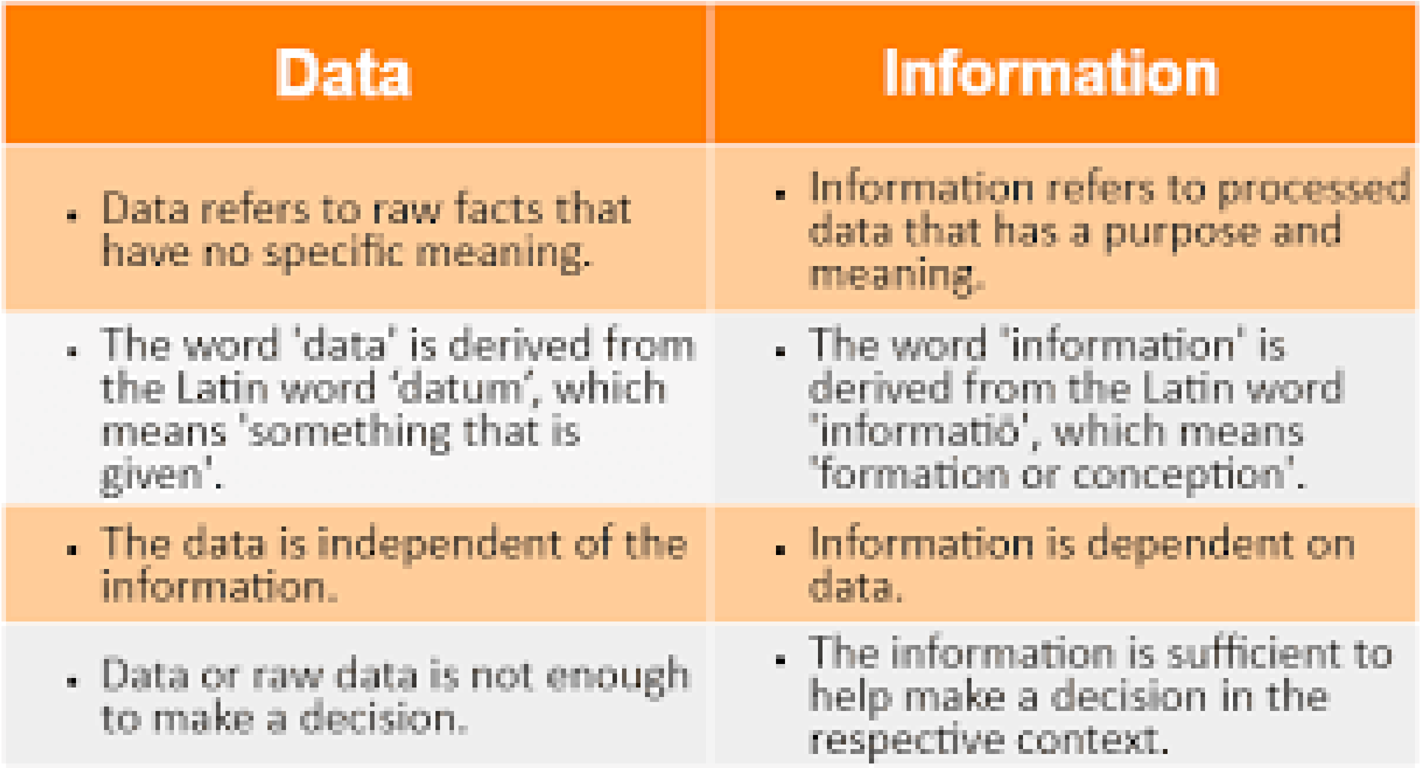
What is Information?

Data is a collection of facts, while information puts those facts into context. While data is raw and unorganized, information is organized. Data points are individual and sometimes unrelated.



© guru99.com

Data Vs Information?



Data	Information
▪ Data refers to raw facts that have no specific meaning.	▪ Information refers to processed data that has a purpose and meaning.
▪ The word 'data' is derived from the Latin word 'datum', which means 'something that is given'.	▪ The word 'information' is derived from the Latin word 'informatiō', which means 'formation or conception'.
▪ The data is independent of the information.	▪ Information is dependent on data.
▪ Data or raw data is not enough to make a decision.	▪ The information is sufficient to help make a decision in the respective context.

What is Database?

- A **database** is an systematic collection of data, so that it can be easily accessed and managed.
- We can organize data into tables, rows, columns, and index it to make it easier to find relevant information.
- The **main purpose** of the database is to operate a large amount of information by storing, retrieving, and managing data.

•Database Management Systems had been first implemented in the **1960**s.

•Charles Bachman along with his team invented the first DBMS known as Integrated Data Store (IDS).

What are the Examples of Databases?

Example of Database?

Example:

1. An online telephone directory uses a database to store data of people, phone numbers, and other contact details.

2. Electricity service provider uses a database to manage billing, client-related issues, handle fault data, etc.

3. Facebook- It needs to store, manipulate, and present data related to members, their friends, member activities, messages, advertisements, and a lot more.

DBMS - Most Popular Database Management Systems

ORACLE DATABASE
MySQL
mongoDB
Microsoft SQL Server
MariaDB
PostgreSQL
Microsoft Access
IBM DB2

www.learncomputerscienceonline.com

What is a Database Management System (DBMS)?

- **Database Management System (DBMS)** is a collection of programs that enable its users to access databases, **manipulate** data and **represent** data.
- It also helps to **control access** to the database.
- Database Management Systems had been first implemented in the 1960s.
- **Charles Bachman's** Integrated Data Store (IDS) is said to be the **first DBMS** in history.

What is a Database Management System (DBMS)?

DBMS is a general-purpose software System, that facilitates the processes of Defining, constructing, manipulating, Sharing databases among various users and applications.

Database Management System (DBMS)

- DBMS contains information about a particular enterprise
  - Collection of interrelated data
  - Set of programs to access the data
- **Database Applications:**
  - 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 can be very large.

When not to use a DBMS

When a DBMS may be unnecessary:

- If the database and applications are simple, well defined, and not expected to change.
- If there are real-time requirements that may not be met because of DBMS overhead.
- If access to data by multiple users is not required.

Main Characteristics of the Database Approach

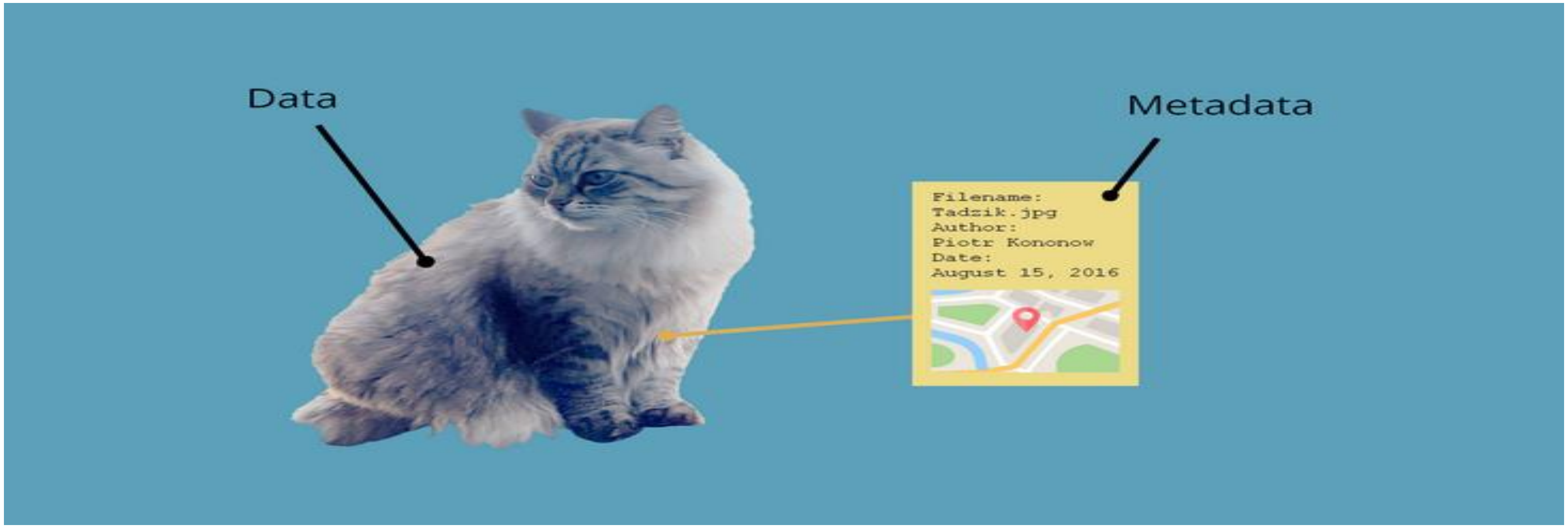
1. **Self-describing nature of a database system**
2. **Insulation between programs and data**
3. **Support of multiple views of the data**
4. **Sharing of data and multiuser transaction processing**

Main Characteristics of the Database Approach

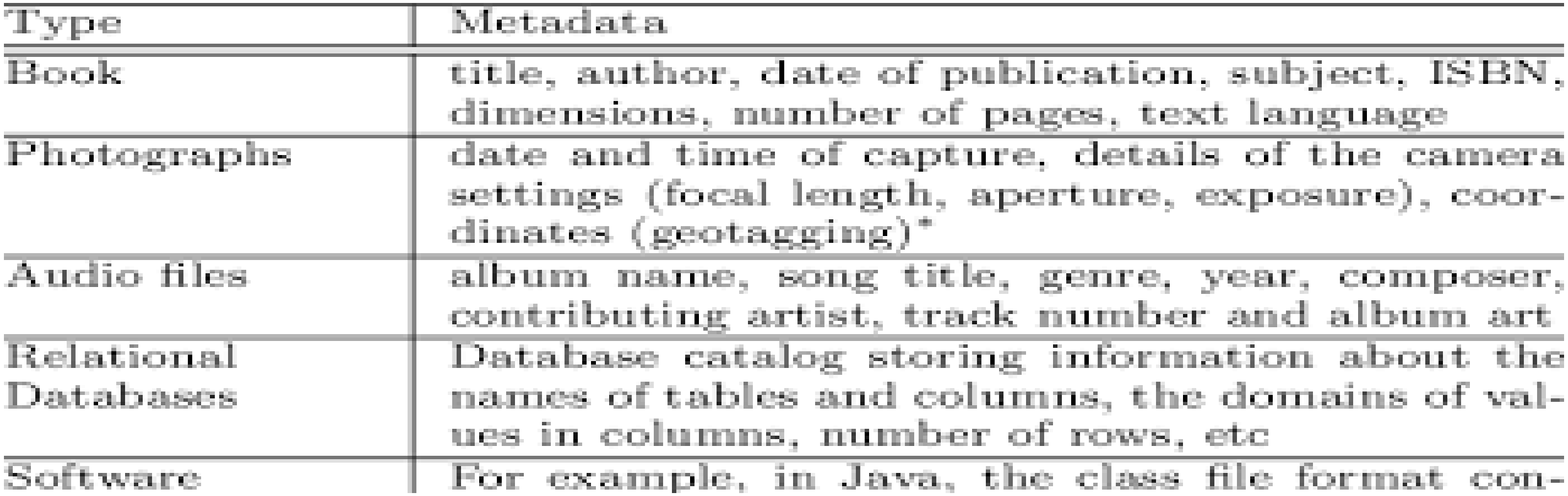
- **Self-describing nature of a database system**: A DBMS **catalog** stores the *description* of the database. (The description is called **meta-data**). This allows the DBMS software to work with different databases.
- **Database System = Database +Meta-Data(Database definition)-> stored in DBMS catalog->used by DBMS s/w & Database Users**

Note: Meta data contains data types,constraints and other information about data

Main Characteristics of the Database Approach



Filename:
Tadzik.jpg
Author:
Piotr Kononow
Date:
August 15, 2016



Type	Metadata
Book	title, author, date of publication, subject, ISBN, dimensions, number of pages, text language
Photographs	date and time of capture, details of the camera settings (focal length, aperture, exposure), coordinates (geotagging)*
Audio files	album name, song title, genre, year, composer, contributing artist, track number and album art
Relational Databases	Database catalog storing information about the names of tables and columns, the domains of values in columns, number of rows, etc
Software	For example, in Java, the class file format con-

Main Characteristics of the Database Approach

- **Insulation between programs and data**:
- In traditional file processing structure of data file is embedded in the application programs.
- In database approach structure of data file is stored in the DBMS catalog which is separate from access programs
- Called **program-data independence**. Allows changing data storage structures and operations without having to change the DBMS access programs.
- The characteristic that allows **program-data independence** is called data abstraction

Main Characteristics of the Database Approach

- **Data Abstraction:** A **data model** is used to hide storage details and present the users with a *conceptual view* of the database.
- **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**.
- **Example:** One user may be interested in **courses** only and other is interested in only **grades** of the students

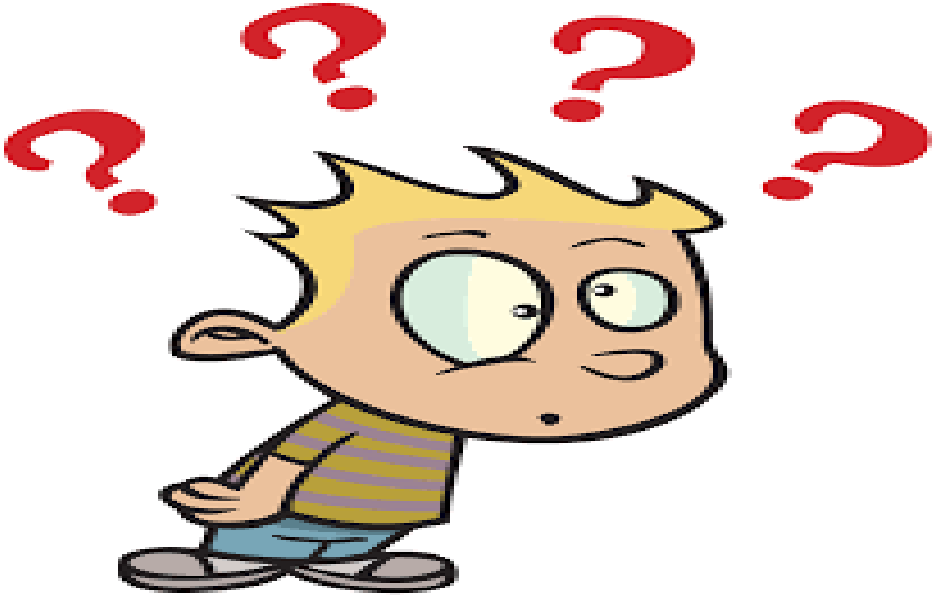
Note: View is a subset of database which contains **virtual data** derived from database

Main Characteristics of the Database Approach

- **Sharing of data and multiuser transaction processing** : allowing a set of concurrent users to retrieve and to update the database. **Concurrency control** within the DBMS guarantees that each **transaction** is correctly executed or completely aborted. OLTP (**O**n**l**ine **T**ransaction **P**rocessing) is a major part of database applications.
- Two users are not allowed to use same data for same time.

- **Example:** Booking system for travel agency

Quiz Time



Quiz Time

1. What is a database?

a)Organized collection of information that cannot be accessed, updated, and managed
b) Collection of data or information without organizing
c) Organized collection of data or information that can be accessed, updated, and managed
d) Organized collection of data that cannot be updated

Quiz Time

1. What is a database?

a)Organized collection of information that cannot be accessed, updated, and managed
b) Collection of data or information without organizing
c) **Organized collection of data or information that can be accessed, updated, and managed**
d) Organized collection of data that cannot be updated

2. Which type of data can be stored in the database?

a) Image oriented data
b) Text, files containing data
c) Data in the form of audio or video
d) All of the above

2. Which type of data can be stored in the database?

a) Image oriented data
b) Text, files containing data
c) Data in the form of audio or video
d) **All of the above**

3. Who created the first DBMS?

a) Edgar Frank Codd
b**)** Charles Bachman
c) Charles Babbage
d) Sharon B. Codd

3. Who created the first DBMS?

a) Edgar Frank Codd

b**) Charles Bachman**

c) Charles Babbage

d) Sharon B. Codd

4.Which of the following is not an example of DBMS?

a) MySQL
b) MongoDB
c) IBM DB2
d) Google(Bigtable)

4.Which of the following is not an example of DBMS?

a) MySQL
b) MongoDB
c) IBM DB2
d) Google

Content

- File System vs Database Systems
- Data Abstraction and data independence
- Database System Architecture
- Database users and Database Administrators(DBA)
- Role of DBA

File system vs DBMS

Basis	File System	DBMS
Structure	The file system is software that manages and organizes the files in a storage medium within a computer.	DBMS is software for managing the database.
Data Redundancy	Redundant data can be present in a file system.	In DBMS there is no redundant data.
Backup and Recovery	It doesn’t provide backup and recovery of data if it is lost.	It provides backup and recovery of data even if it is lost.
Query processing	There is no efficient query processing in the file system.	Efficient query processing is there in DBMS.
Consistency	There is less data consistency in the file system.	There is more data consistency because of the process of normalization.
Complexity	It is less complex as compared to DBMS.	It has more complexity in handling as compared to the file system.
Security Constraints	File systems provide less security in comparison to DBMS.	DBMS has more security mechanisms as compared to file systems.

Basis	File System	DBMS
Cost	It is less expensive than DBMS.	It has a comparatively higher cost than a file system.
User Access	Only one user can access data at a time.	Multiple users can access data at a time.

DBMS vs. File Systems?

- Database consists of logically related data stored in a single repository
- Provides advantages over file system management approach
  - Eliminates inconsistency, data anomalies, data dependency, and structural dependency problems
  - Stores data structures, relationships, and access paths

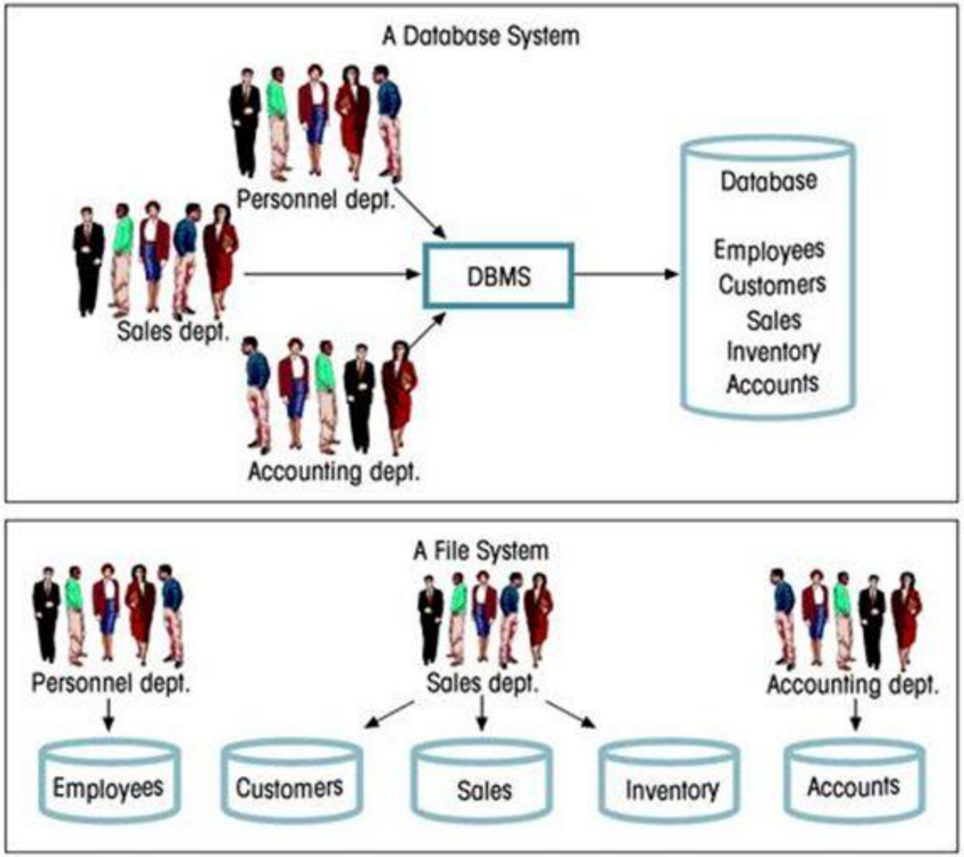


FIGURE 1.6 CONTRASTING DATABASE AND FILE SYSTEMS

DBMS, Ch1 File Systems and Databases, Hachim Haddouti

File System Vs. Database System

Database Systems offers solutions to all File system Problems

1.Controlled Redundancy :

- Storing the data multiple times (Redundancy) avoided through database approach .
- It stores each logical data item in only one place while designing the database.
- This saves storage space, time for multiple updates and ensures consistency.

2.Restricting the unauthorized access:

- DBMS provides a security & authorization subsystem through which DBS staff grant privileges to another user.

File System Vs. Database System

3.Providing Storage Structures for efficient Query processing (2):

- DBMS provides specialized **data structures** to **speed up** disk search for desired record.
- It provides facility **of Indexes** which are typically based on tree or hash data structures.

4.Providing Back –up & Recovery (5):

- By introducing **save point** , it gives **the back up& recovering** facility.

5.Providing Multiple user interfaces (6):

Casual Users - Query Language interface

Application Programmer - Programming language interface

Naïve users – Menu-Driven interfaces

Data Abstraction and Data Independence

Data Abstraction

Data Abstraction is a process of **hiding unwanted or irrelevant details** from the end user.

The database systems consist of complicated data structures and relations. For users to access the data easily, these complications are **kept hidden**, and only the relevant part of the database is made accessible to the users through **data abstraction**.

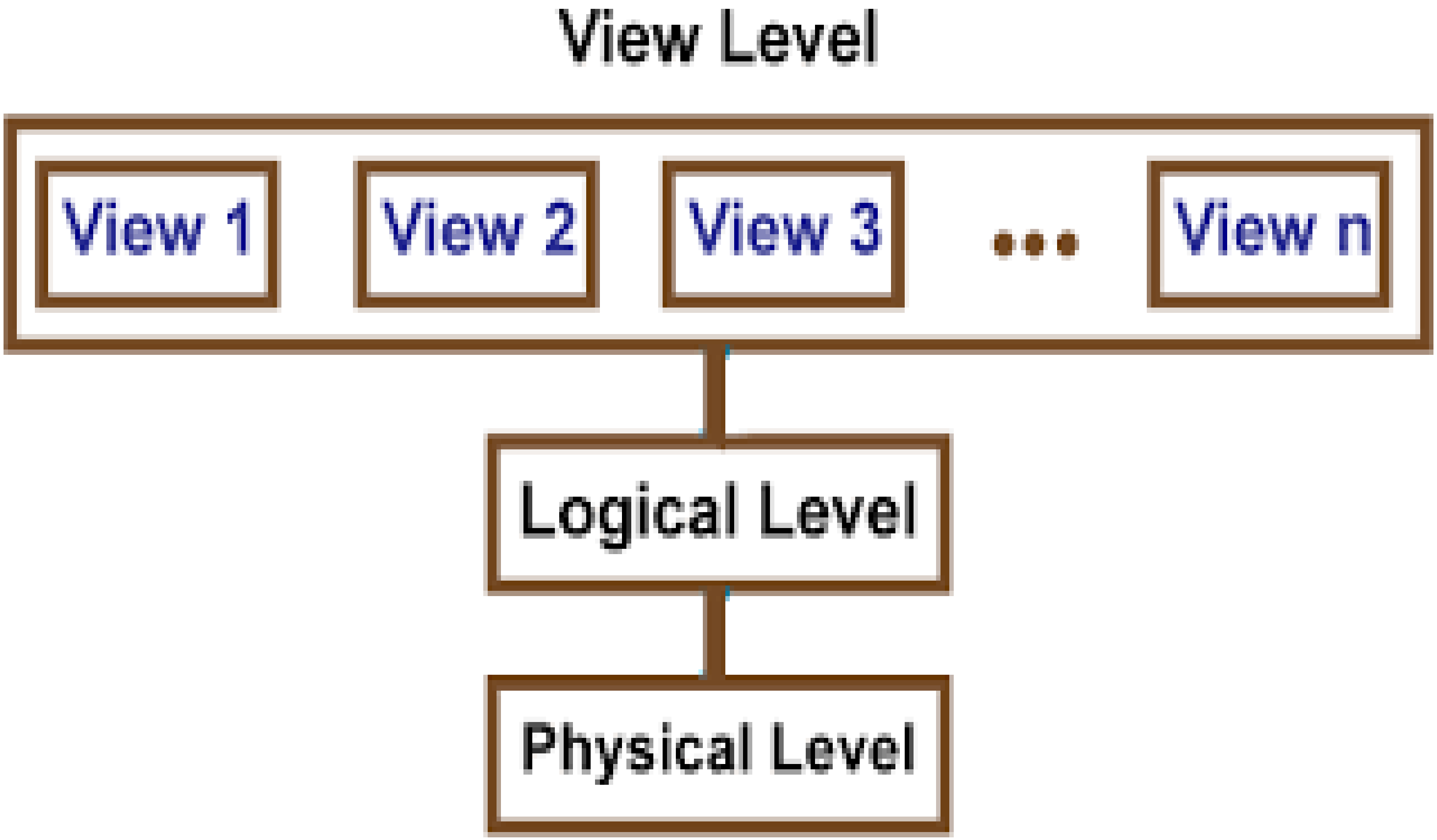
Example

Making coffee with a coffee machine is a good example of abstraction. You need to know how to use your coffee machine to make coffee. You need to provide water and coffee beans, switch it on and select the kind of coffee you want to get

Levels of abstraction for DBMS

Mainly there are **three levels** of abstraction for DBMS, which are as follows –

1. Physical or Internal Level
2. Logical or Conceptual Level
3. View or External Level


View Level
View 1
View 2
View 3
...
View n
Logical Level
Physical Level

1. Physical or Internal Level

- It is the **lowest level of abstraction** for DBMS which **defines how the data is actually stored**, it defines **data-structures to store data** and access methods used by the database. Actually, it is decided by developers or database application programmers how to store the data in the database.
- So, overall, the entire database is described in this level that is physical or internal level. It is a very **complex level to understand**.

- **For example,** customer's information is stored in tables and data is stored in the form of blocks of storage such as bytes, gigabytes etc.

2. Logical or Conceptual Level

- ❑ Logical level is the **intermediate level** or **next higher level.** It describes **what data is stored** in the database and **what relationship** exists among those data. It tries to describe the entire or whole data because it describes **what tables to be created** and what are the **links** among those tables that are created.
- ❑ It is less complex than the physical level. Logical level is **used by developers or database administrators (DBA).** So, overall, the logical level contains **tables (fields and attributes)** and **relationships** among table attributes.

3.View or External Level

- ❑ It is the **highest level**. In view level, there are different levels of views and every view only defines a part of the entire data. It also simplifies interaction with the user and it provides many views or multiple views of the same database.
- ❑ View level can **be used by all users** (all levels' users). This level is the least complex and **easy to understand**.
- ❑ **For example,** a user can interact with a system using GUI that is view level and can enter details at GUI or screen and the user does not know **how data is stored** and what data is stored, this detail is hidden from the user.

Levels of abstraction for DBMS

1.Physical or Internal Level - **describes how a record** (e.g., instructor) is **stored**.

2.Logical or Conceptual Level - describes data stored in **database**, and the **relationships among** the data.

```
type instructor = record
                  ID : string;
name : string;
dept_name : string;
salary : integer;
     end;
```

3.View or External Level - application programs hide details of data types. Views can also hide information (such as an employee's salary) for security purposes.

Data Independence

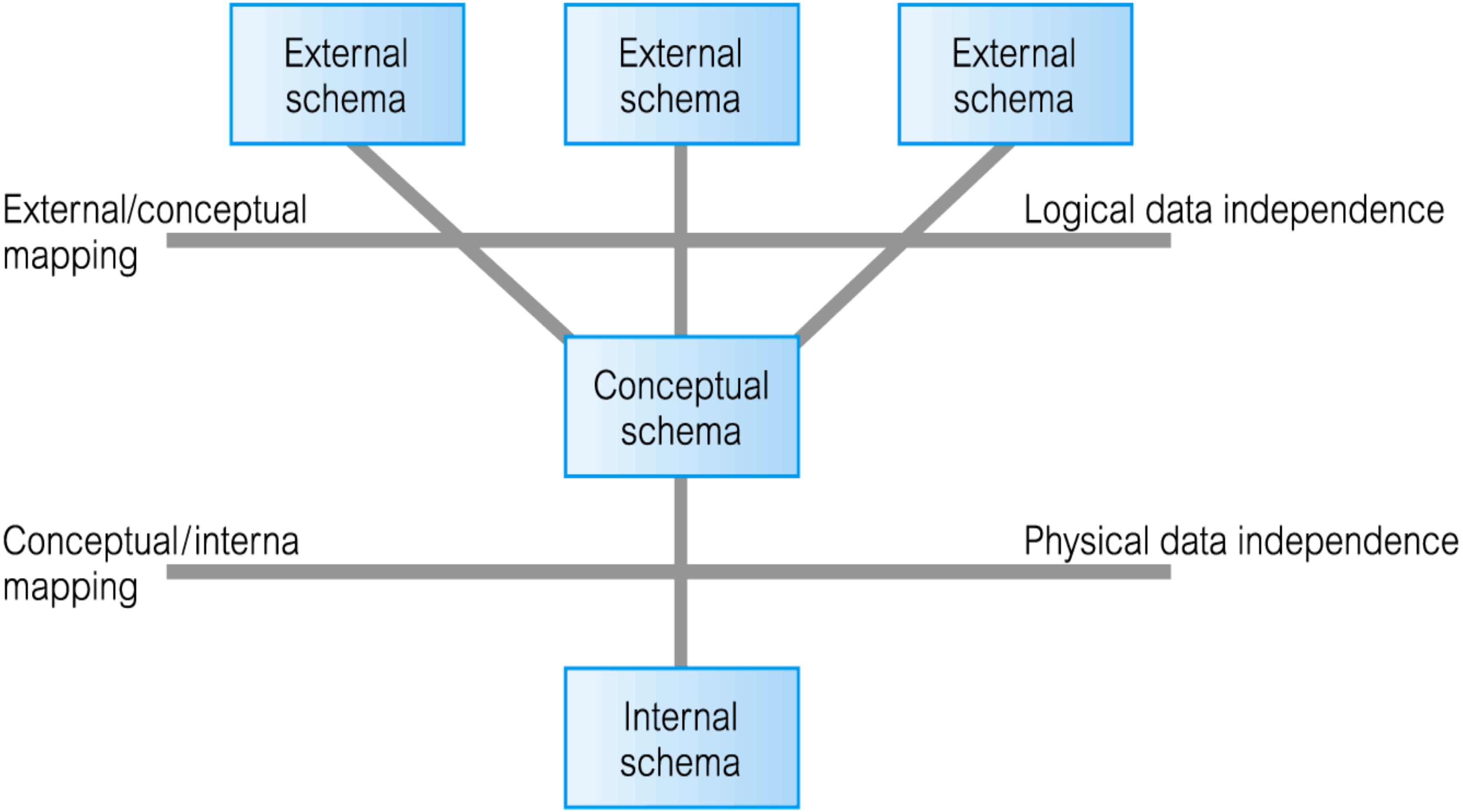
Data Independence is defined as a **property of DBMS that helps you to** **change the Database schema** at one level of a database system **without requiring to change the schema** at the next higher level.

❖ Types of Data Independence

In DBMS there are two types of data independence

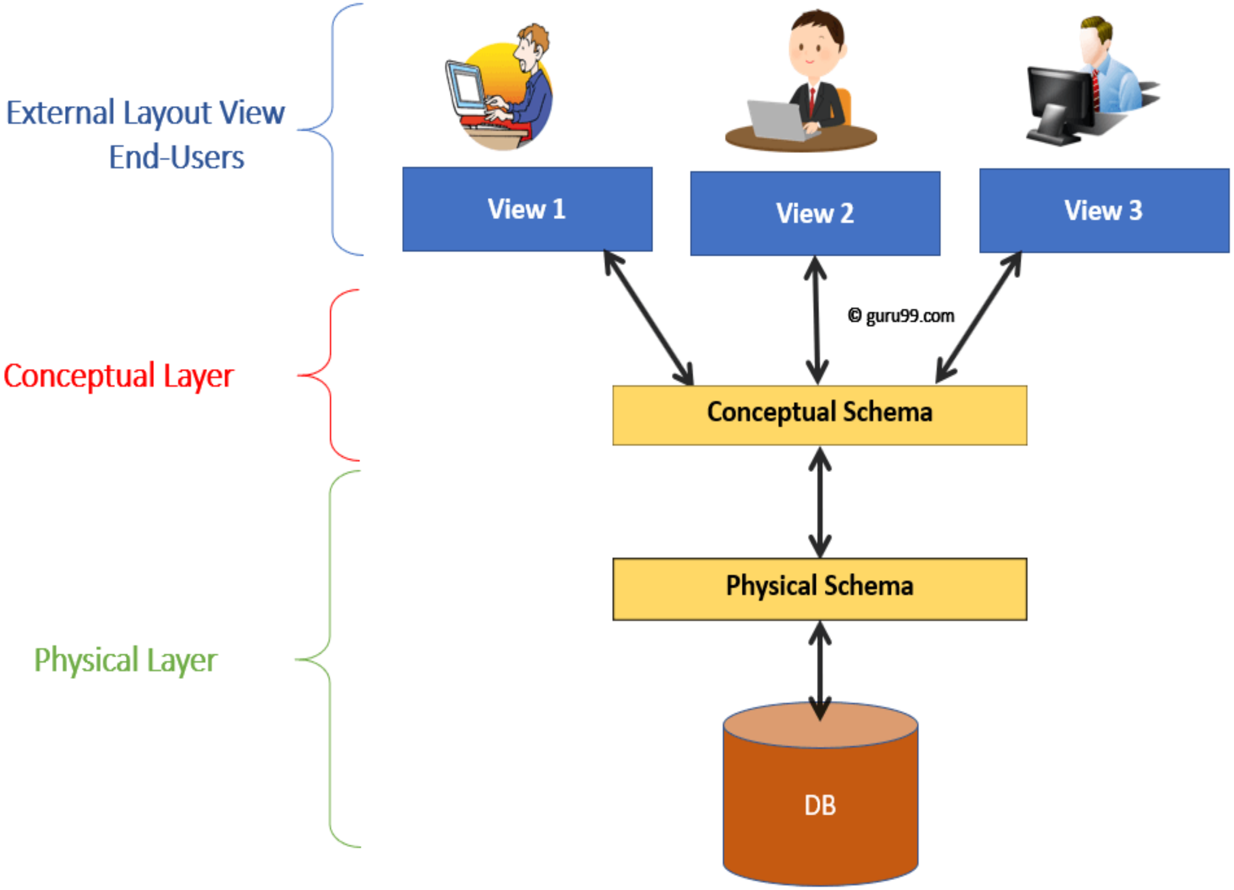
1.Physical data independence
2.Logical data independence

Data Independence


External schema
External schema
External schema
External/conceptual mapping
Logical data independence
Conceptual schema
Conceptual/interna mapping
Physical data independence
Internal schema

Levels of Database

1. Physical/Internal
2. Conceptual
3. External



Consider an Example of a University Database. At the different levels this is how the implementation will look like:

Type of Schema	Implementation
External Schema	**View 1**: Course info(cid:int,cname:string) **View 2**: studeninfo(id:int. name:string)

Conceptual Shema	Students(id: int, name: string, login: string, age: integer) Courses(id: int, cname:string, credits:integer)
	Enrolled(id: int, grade:string)

Physical Schema	•Relations stored as unordered files. •Index on the first column of Students.

Physical Data Independence

- ❑ Physical data independence refers to the **ability to change the data's physical structure** without affecting the conceptual level.

- ❑ Physical changes include using a **new storage device or moving the database's location**, **changing the data structure,** or **altering indexes** to speed up data retrieval.

- ❑ **Example:** The patient database could be moved from drive C to drive D, but the conceptual schema and external views remain unchanged because of physical data independence.

Logical Data Independence

- ❑ Logical data independence, on the other hand, **allows users to change the conceptual schema** without changing the external views.

- ❑ **For example,** the hospital billing department could add a column to the database table for each patient’s insurance policy number. Having logical data independence means that the view of the food services department and other users are not changed even though modifications were made to the conceptual level.

Due to Logical independence, any of the below change will not affect the external layer.

1.Add/Modify/Delete a new attribute, entity or relationship is possible without a rewrite of existing application programs
2.Merging two records into one
3.Breaking an existing record into two or more records

Goal of 3 level /schema of Database

Objectives of using Three schema Architecture:

1. Every user should be **able to access** the same data but able to see a **customized view** of the data.
2. The user need **not to deal directly** with **physical database storage** detail.
3. The DBA should be **able to change** the database storage structure **without disturbing the user's views**
4. The internal structure of the database should remain **unaffected** when changes made to the physical aspects of storage.

QUIZ

Quiz Time

1. With DBMS , Multiple user can access the same data at a same time

A. True

B. False

1. With DBMS , Multiple user can access the same data at a same time

A.True

B.False

2. ____________is a process of hiding unwanted or irrelevant details from the end user.

A. Data Abstraction

B. Data Independence

C. Data Restoration

D. Data Cleaning

2. ____________is a process of hiding unwanted or irrelevant details from the end user.

A. **Data Abstraction**

B. Data Independence

C. Data Restoration

D. Data Cleaning

3. Physical changes does not include_____

A. using a new storage device
B. moving the database’s location
C. changing the data structure
D. Merging two records into one

3. Physical changes does not include_____

A. using a new storage device
B. moving the database’s location
C. changing the data structure
D. Merging two records into one

4. Logical data independence, allows users to change the conceptual schema without changing the _____

A. External views

B. Conceptual views

C. Physical Views

4. Logical data independence, allows users to change the conceptual schema without changing the _____

A. **External views**

B. Conceptual views

C. Physical Views

Three-Schema Architecture

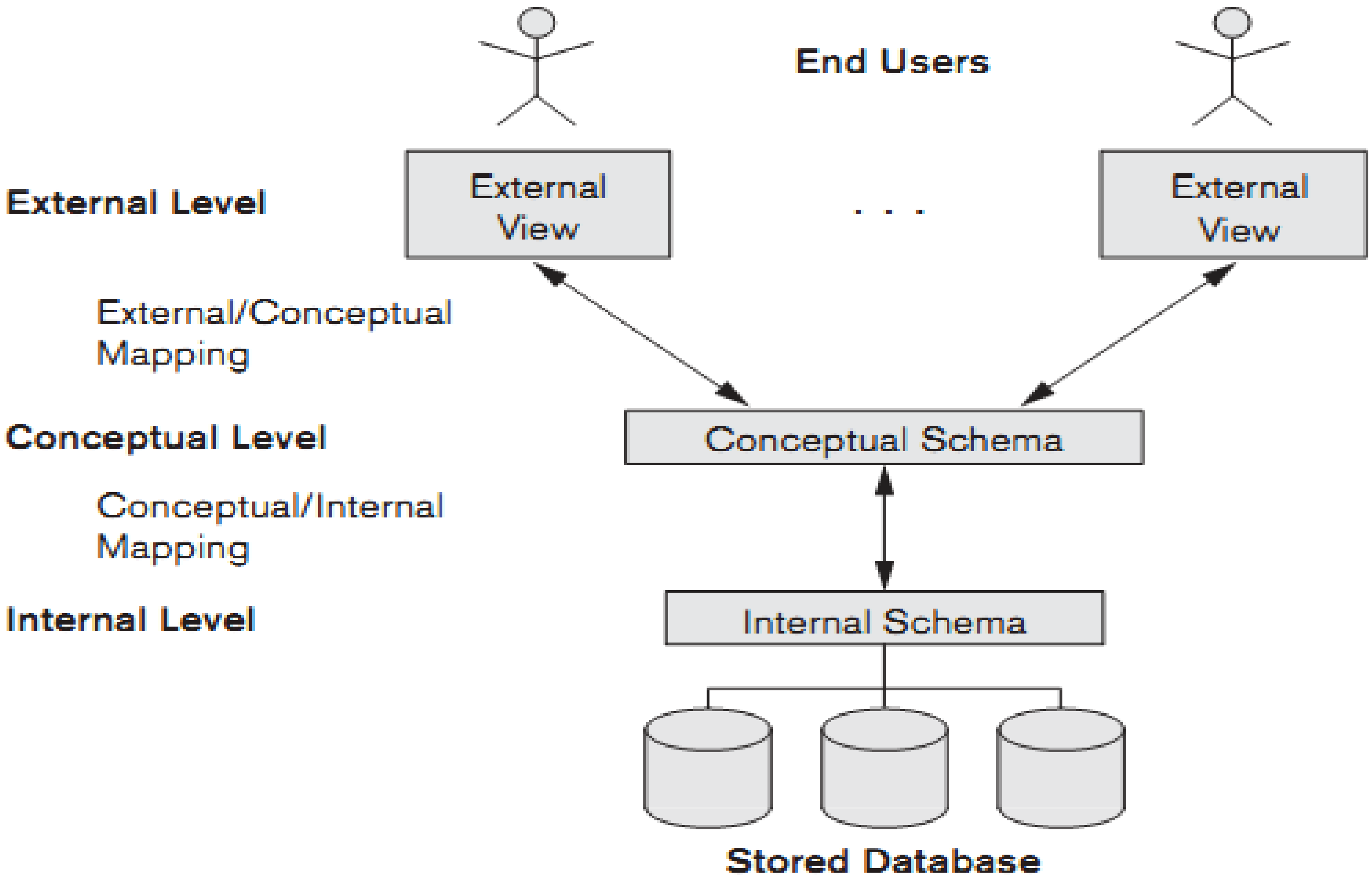
- Defines DBMS schemas at *three levels*:
  - **Internal schema** at the internal level to describe physical storage structures and access paths. Typically uses a ***physical*** data model.

  - **Conceptual schema** at the conceptual level to describe the structure of the *whole* database for a community of users. It concentrates on describing entities, data types, relationships, user operations and constraints. Uses a ***conceptual* or an *implementation*** data model.

Three-Schema Architecture Cont…

- **External schemas** at the external level to describe the various user views. Each external schema describes the part of the database that a particular user group is interested in and hides the rest of the database .Usually uses the same data model as the conceptual level.

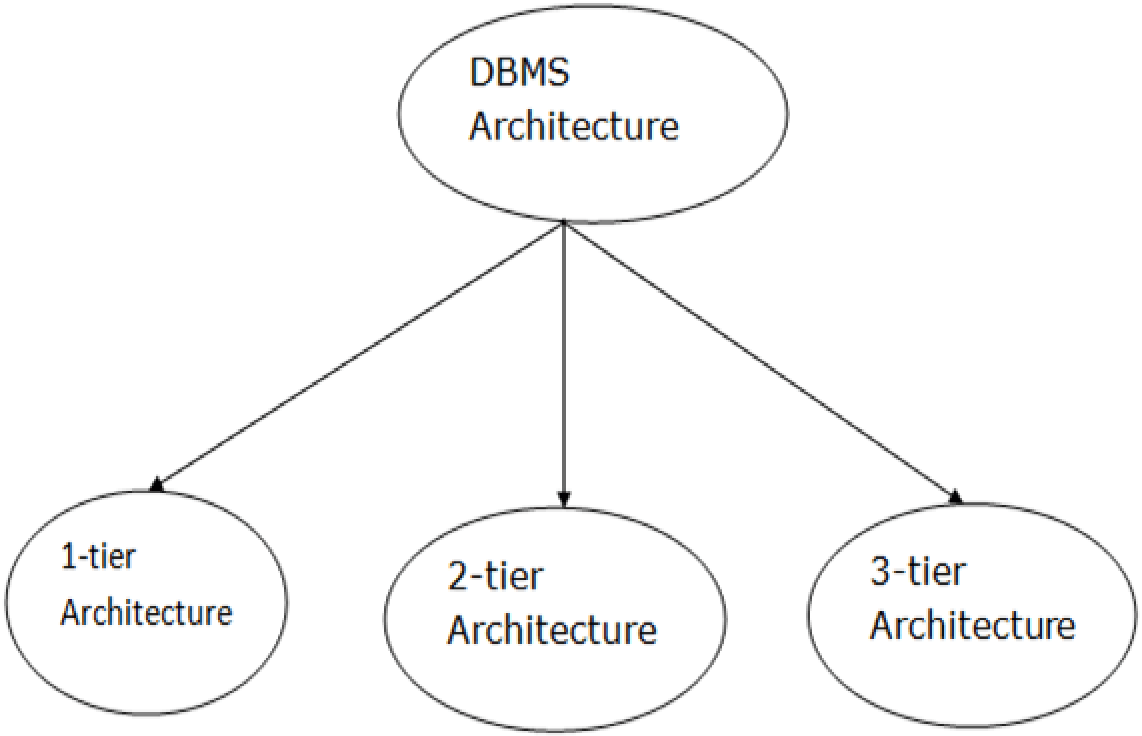
Three-Schema Architecture



Application Architecture

Types of DBMS Architecture/Application Architectures

1. One Tier Architecture (Single Tier Architecture)
2. Two Tier Architecture
3. Three Tier Architecture

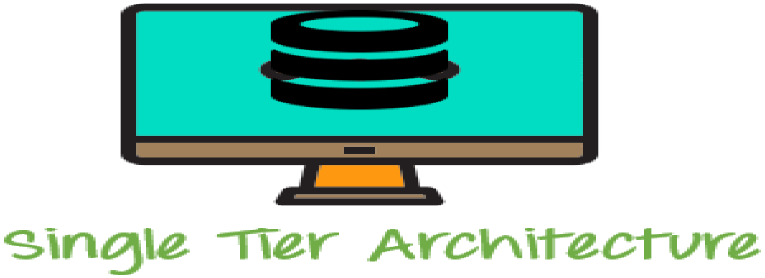

DBMS Architecture
1-tier Architecture
2-tier Architecture
3-tier Architecture

1-Tier Architecture

1 Tier Architecture in DBMS is the simplest architecture of Database in which where the user interface, application logic, and data storage are all combined into a single unit on same machine.

- In this architecture, the database is directly available to the user.
- Any changes done here will directly be done on the database itself.
- The 1-Tier architecture is used for development of the local application, where programmers can directly communicate with the database for the quick response.

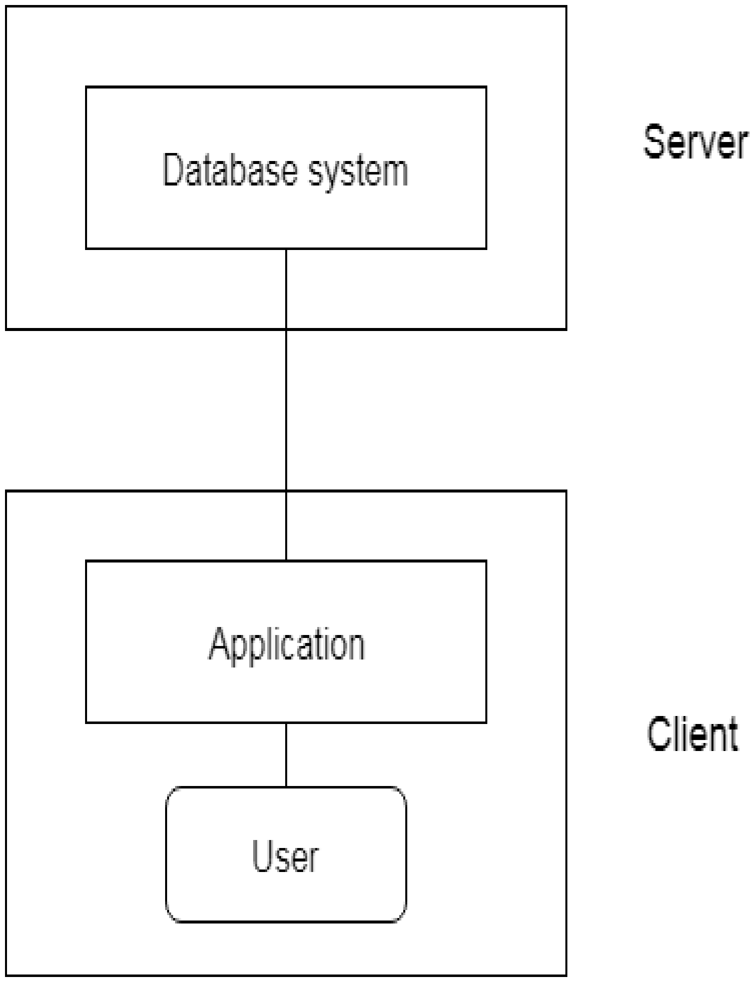
Applications that handle all the three tiers such as Phone Book.Address Book Application, MS Office



2-Tier Architecture

- The 2-Tier architecture is same as **basic client-server**. In the two-tier architecture, applications on the client end can **directly communicate** with the database at the server side.
- The user interfaces and application programs are run on the client-side.
- The server side is responsible to provide the functionalities like: **query processing and transaction management**.
- To communicate with the DBMS, client-side application establishes a connection with the server side.

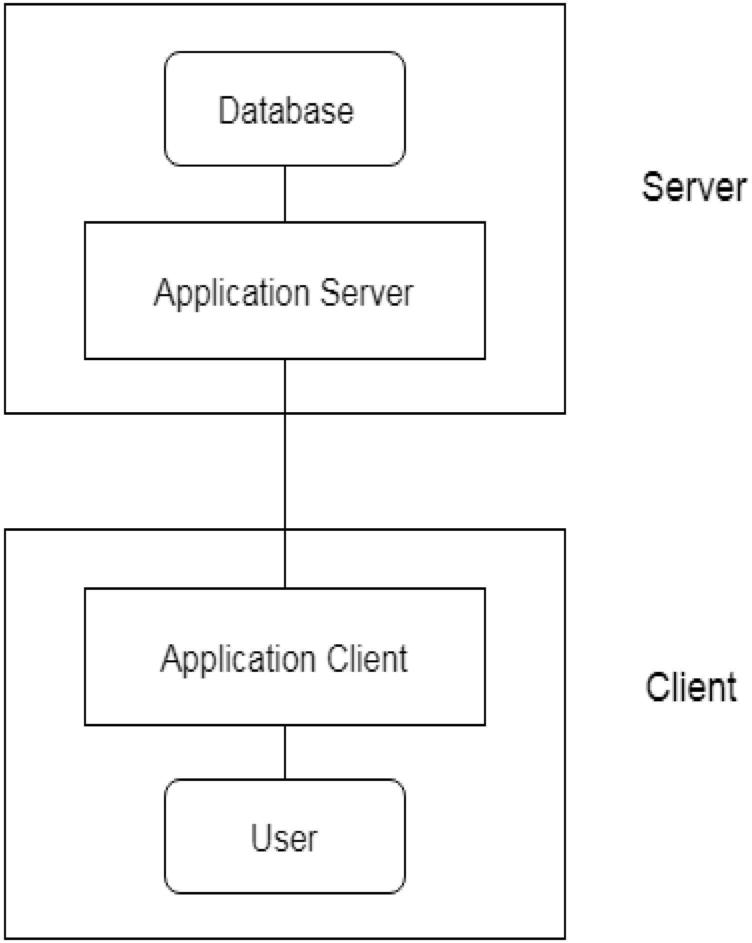
Examples :Small Business Inventory Management System,Desktop applications, excel sheets, word document, desktop games.


Database system
Server
Application
Client
User

3-Tier Architecture

- The 3-Tier architecture contains another layer between the client and server. In this architecture, **client can't directly communicate** with the **server.**
- The application on the client-end interacts with an application server which further communicates with the database system.
- End user has no idea about the existence of the database beyond the application server. The database also has no idea about any other user beyond the application.
- The 3-Tier architecture is used in case of **large web application.**

- **Example:-**E-commerce Website


Database
Application Server
Server
Application Client
Client
User

DBMS Architecture

A **Database Architecture** is a representation of DBMS design. It helps to design, develop, implement, and maintain the database management system.

What are DBMS Languages?

1. DDL (Create, Alter, drop, Truncate)
2. DML (delete, insert, update)
3. Data Retrieval Statement (Select) ->DQL
4. DCL (Grant, Revoke)
5. TCL (Commit, Roll back, Save Point)

Components of a Database System

A database system consists of the items listed below:

- **The Database(data)**
- **A DBMS(Software)**
- **A DDL and a DML (part of the DBMS)**
- **Application Programs**

Database Users

There are two types of database users, Users, and Administrators.

- **Database administrators**
- **Naïve users**
- **Application programmers**
- **Sophisticated users**
- **Specialized users**

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

- **Schema definition** - The DBA creates the original database schema by executing a set of data definition statements in the DDL.
- **Storage structure and access-method definition**
- **Schema and physical-organization modification -** The DBA carries out changes to the schema and physical organization to reflect the changing needs of the organization, or to alter the physical organization to improve performance.
- **Granting of authorization for data access -** By granting different types of authorization, the database administrator can regulate which parts of the database various users can access. The authorization information is kept in a special system structure that the database system consults when ever someone attempts to access the data in the system.
- **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 Users (Cont…)

Naïve : are unsophisticated users who interact with the system by **invoking one of the application programs** that have been written previously.

They do not need to understand the DBMS or the underlying structure of the d/b.

For Example, people accessing database over the web, bank tellers, clerical staff

For example, A Bank cashier or a clerk in the university who needs to add a new instructor to department *A* invokes a program called *new hire*. This program asks the clerk for the name of the new instructor, her new *ID*, the name of the department (that is, *A*), and the salary.

Naive users may also simply read *reports* generated from the database.

Database Users (Cont…)

- **Application programmers** are computer professionals who **write application programs**. This programs are appropriately tested, documented and maintained.
- Application programmers can choose from many tools to develop user interfaces.
- **Rapid application development (RAD)** tools are tools that enable an application programmer to construct forms and reports with minimal programming effort.

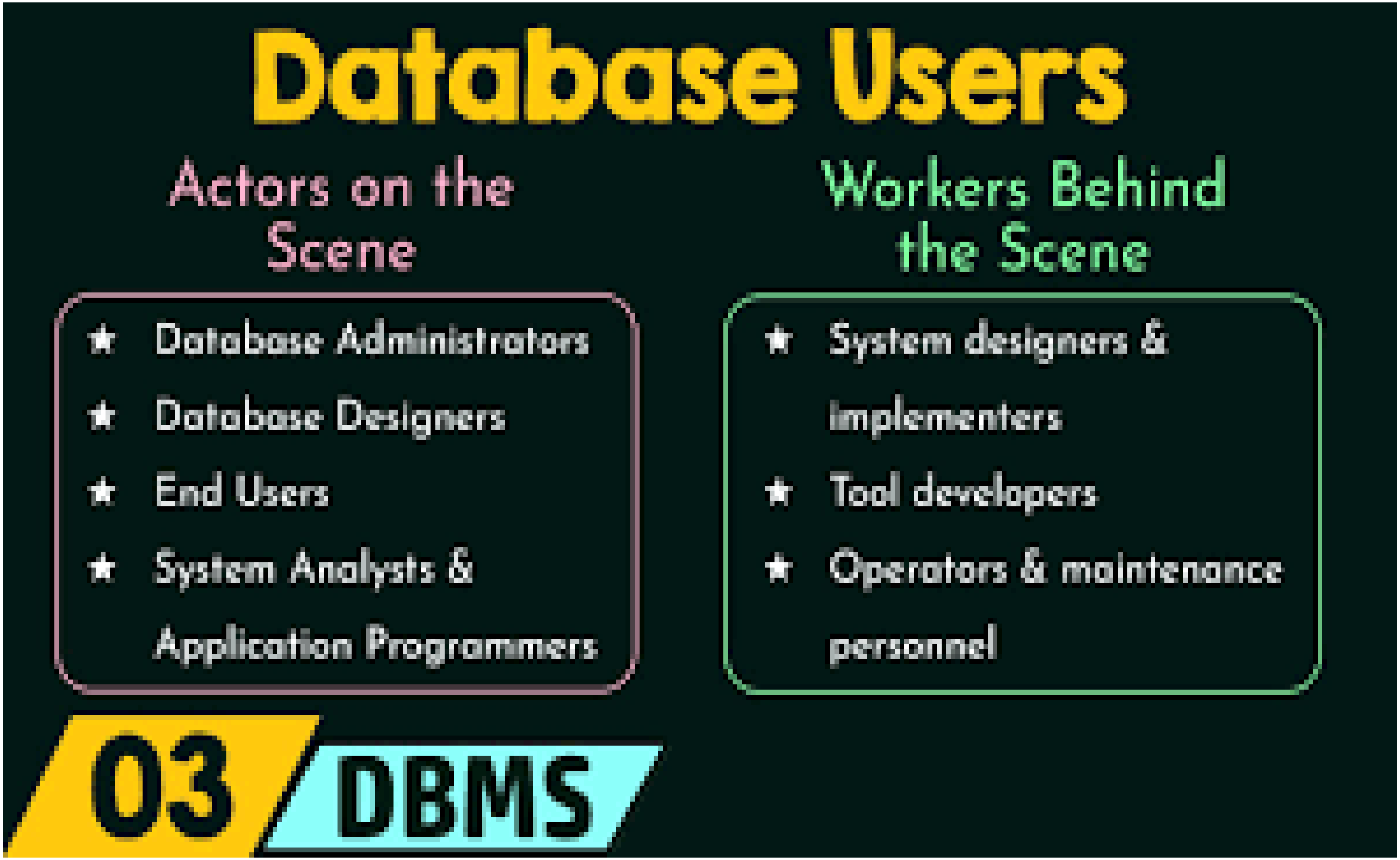
Database Users (Cont…)

- **Sophisticated users - This users are familiar with the DBMS and underlying structure of the database.**
- **Sophisticated users** interact with the system without writing programs. Instead, they form their requests either using a database **query language** or by using tools such as **data analysis software**.
- Analysts who submit queries to explore data in the database fall in this category.

Database Users (Cont…)

- **Specialized users** are sophisticated users who write **specialized database applications** that do not fit into the traditional data-processing framework.
- Among these applications are computer-aided design systems, knowledgebase and expert systems, systems that store data with **complex data types**
- Example, graphics data and audio data, and environment-modeling systems.

Database Users (Cont…)



Database Users

Actors on the Scene

- Database Administrators
- Database Designers
- End Users
- System Analysts & Application Programmers

Workers Behind the Scene

- System designers & implementers
- Tool developers
- Operators & maintenance personnel

03 DBMS

DATABASE SYSTEM ARCHITECTURE

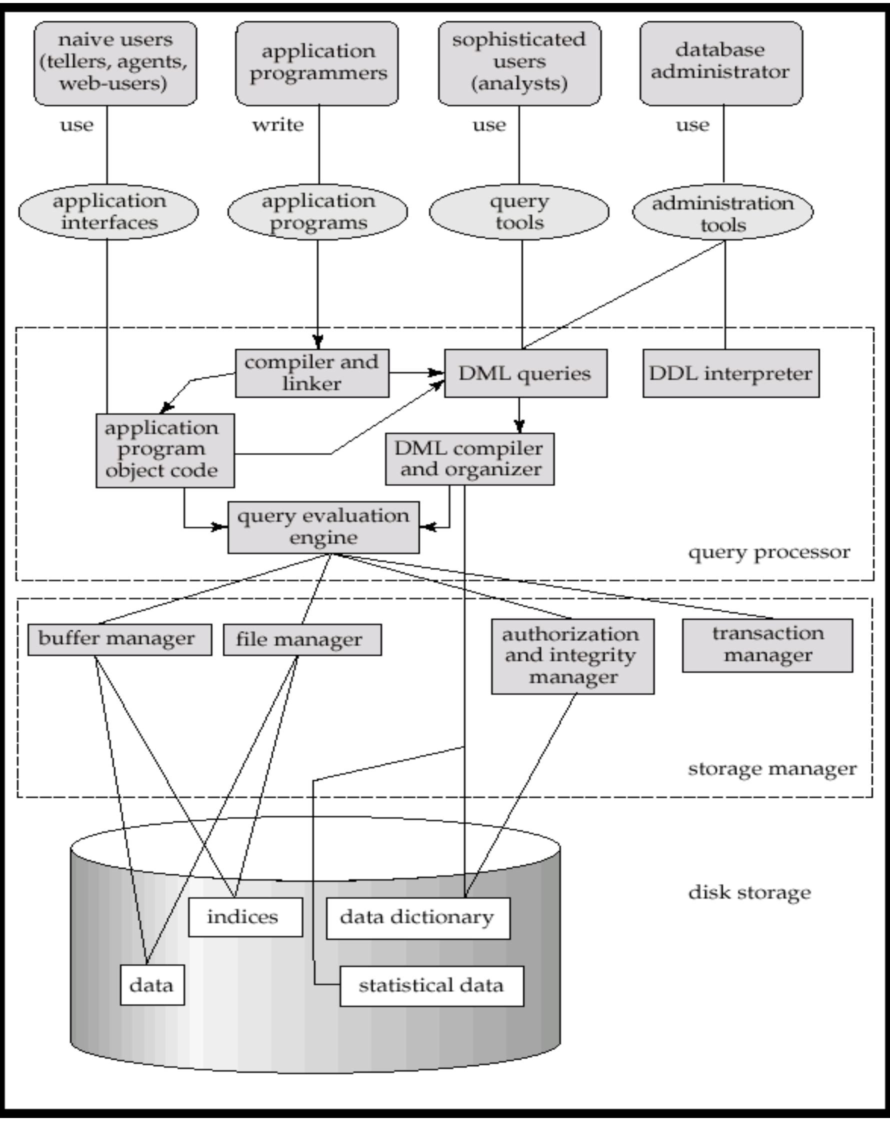
OR

OVERALL SYSTEM ARCHITECTURE

Database Management System Internals

- ❑Database Users
- ❑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.
- The storage manager is responsible to the following tasks:
  - Interaction with the file manager
  - Efficient **storing, retrieving and updating** of data

Issues:

- Storage access
- File organization
- Indexing and hashing

Storage Management

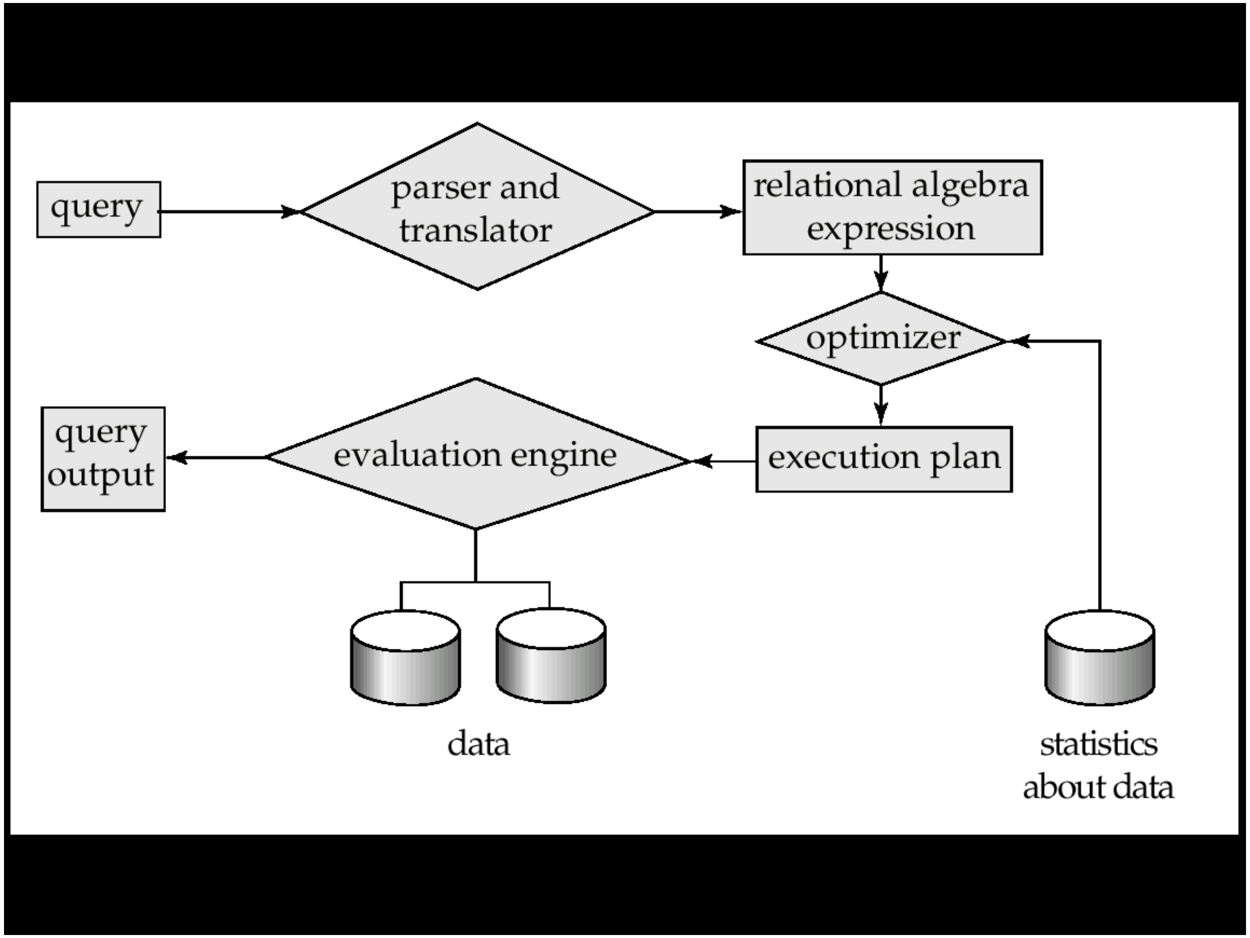
- Buffer Manager - Manage the movement of data **between the disk storage and the main memory** (RAM) of a computer. The buffer manager uses a portion of the main memory as a buffer, which is a temporary storage space for holding frequently accessed data from the database, to reduce the need for repetitive disk reads.
- File Manager - It provides an **abstraction layer** between the higher-level components of the DBMS (e.g., query processor, buffer manager, transaction manager) and the underlying operating system, managing **how data is stored and organized on disk.**
- Authorization and Integrity Manager - Maintaining data integrity, Access Control, User Authentication.
- Transaction Manager – Responsible for all the transcations.

Transaction Management

- Ensures data integrity, consistency, and reliability in multi-user environments. Transaction management handles four properties, often referred to as ACID properties:
- Atomicity - Atomicity ensures that a transaction is treated as a single indivisible unit of work. It means that **either all** the operations within a transaction are executed successfully, or **none of them** are.
- Consistency: Consistency guarantees that a transaction will bring the database from **one consistent state to another consistent state.**
- Isolation: Isolation ensures that **multiple transactions** can **execute concurrently** without interfering with each other.
- Durability: Durability guarantees that once a transaction is committed successfully, its **changes are permanent** and will survive any subsequent failures, including system crashes.

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

Quiz time

Q.1 A Database Administrator is individual or person responsible for controlling, maintenance, coordinating, and operation of database management system.

A. True
B. False

Q.2

In which of the following architecture client, server, and Database all reside on the same machine.

A. 1 Tier
B. 2 Tier
C. 3 Tier

Thank You