GALGOTIAS UNIVERSITY

School of Computing Science and Engineering

Program: B.Sc.

Course Code: BSCS2312

Course Name: Database Management System

Vision

To be known globally as a premier department of Computer Science and Engineering for value-based education, multidisciplinary research and innovation.

Mission

- **M1:** Developing a strong foundation in fundamentals of computing science with responsiveness towards emerging technologies.
- **M2:** Establishing state-of-the-art facilities and adopt education 4.0 practices to analyze, develop, test and deploy sustainable ethical IT solutions by involving multiple stakeholders.
- **M3:** Establishing Centers of Excellence for multidisciplinary collaborative research in association with industry and academia.

Course Outcomes (COs)

CO Number	Title
CO1	Understand the basic concepts, modeling techniques and architecture of DBMS (K2).
CO2	Apply the concept of ER Model and SQL programming using DDL and DML commands (K3).
CO3	Able to store and analyze data into normalized format. (K4).
CO4	Analyze the transaction processing concept and recovery methods in database (K4)
CO5	Examine the concept of concurrency control techniques in database (K4).
CO6	List out the various contemporary research areas and database tools (K2).

Course Prerequisites

- Knowledge of Mathematics
- Query Languages

Syllabus

Unit 1 - Introduction (6 hours)

- An Overview of Database Management System
- Database System vs File System
- Database System Concept and Architecture
- Data Model Schema and Instances
- Overall Database Structure

Recommended Books

Text books

- Abraham Silberschatz, Henry F. Korth and S. Sudarshan- "Database System Concepts", Fourth Edition, McGraw-Hill, 2002.

Reference Book

- Ramez Elmasri and Shamkant B. Navathe, "Fundamental Database Systems", Third Edition, Pearson Education, 2003.
- Raghu Ramakrishnan, "Database Management System", Tata McGraw- Hill Publishing Company, 2003.
- Hector Garcia–Molina, Jeffrey D.Ullman and Jennifer Widom- "Database System Implementation"- Pearson Education- 2000
- Peter Rob and Corlos Coronel- "Database System, Design, Implementation and Management", Thompson Learning Course Technology- Fifth edition, 2003

Additional online materials

- Coursera - https://www.coursera.org/learn/database-management
- NPTEL- https://nptel.ac.in/courses/106/105/106105175/
- https://www.coursera.org/learn/research-methods
- https://www.coursera.org/browse/physical-science-and-engineering/research-methods

File Processing System (FPS)

- Earlier, computers are used for the scientific purpose.
- Later, computers are used for the commercial applications.
- In commercial application calculation or computation are not involved at a large scale. Only data processing (read data from different devices, store data after minor calculation output can be displayed on monitor or paper in a proper format).
- Commercial applications introduced FPS.
- FPS is a collection of programs that perform services.
- File processing systems was an early attempt to computerize the manual filing system that we are all familiar with.
- A file system is a method for storing and organizing computer files and the data they contain to make it easy to find and access them.
- File systems may use a storage device such as a hard disk or CD-ROM and involve maintaining the physical location of the files.

File Processing System (FPS)

- Consider a university system as depicted below.
- Different system maintain their own database separately and they have their own program to process the data that is program and data independence

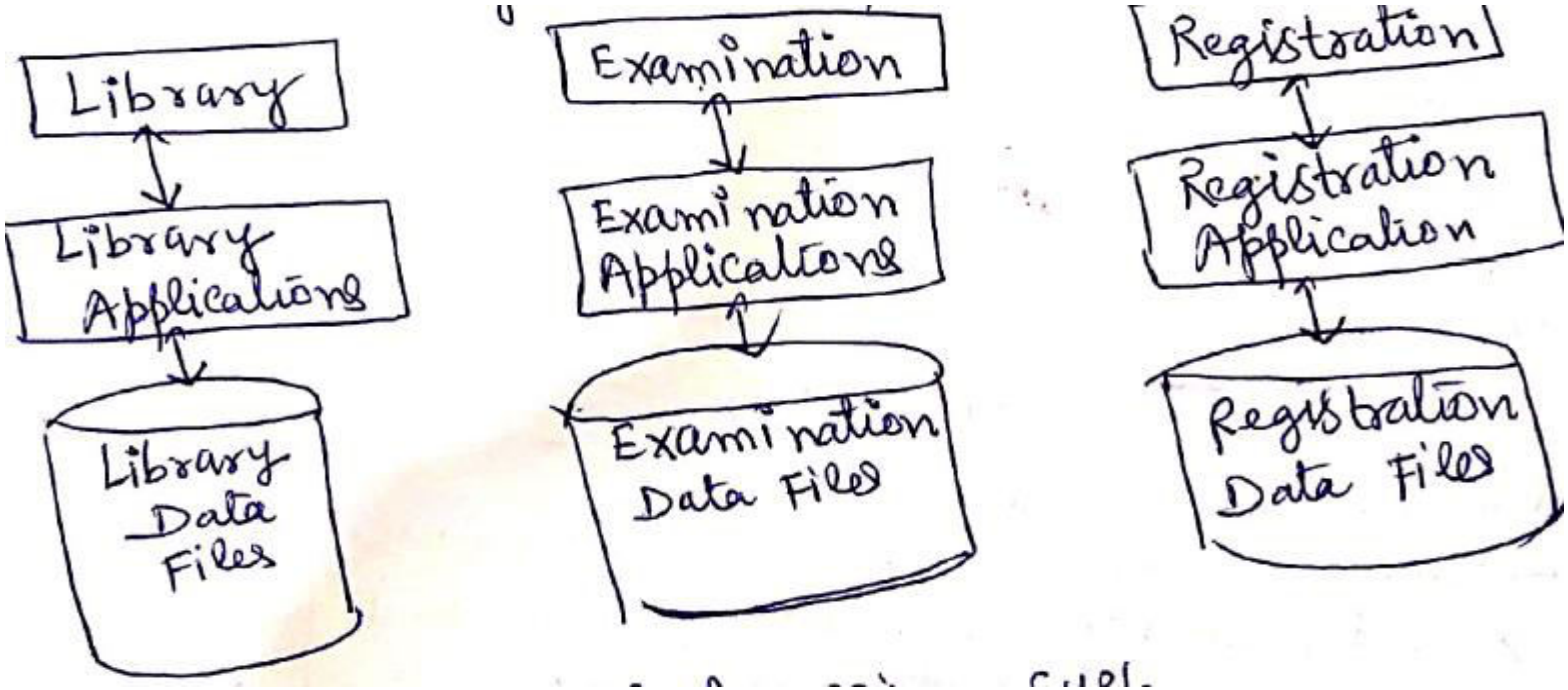


Figure 1: Traditional File Storage System of University Database

Characteristics of FPS

- It is a group of files storing data of an organization.
- Each file is independent from one another.
- Each file is called a flat file.
- Each file contained and processed information for one specific function, such as accounting or inventory.
- Files are designed by using programs written in programming languages such as C, C++, etc.
- The physical implementation and access procedures are written into database application; therefore, physical changes resulted in intensive rework on the part of the programmer.
- As systems became more complex, file processing systems offered little flexibility, presented many limitations, and were difficult to maintain.

Limitations of FPS

- **Separated and Isolated Data:** To make a decision, a user might need data from two separate files.
- **Duplication of data:** Often the same information is stored in more than one file. It leads to loss of data integrity.
- **Data Dependence:** Files and records were described by specific physical formats that were coded into the application program by programmers. If the format of a certain record was changed, the code in each file containing that format must be updated. Let us consider a student file, where information of students is stored in text file and each field is separated by blank space such as **I Rama 22 India**

  Now, if the delimiter of the field changes from blank space to semicolon such as **1; Rama; 22; India**

  Then, the application programs using this file must be modified, because now it will token the field on semicolon; but earlier it was blank space.

Limitations of FPS

- **Difficulty in representing data from the user's view:** To create useful applications for the user, often data from various files must be combined. In file processing it was difficult to determine relationships between isolated data in order to meet user requirements.
- **Data Inflexibility:** Program-data interdependency and data isolation, limited the flexibility of file processing systems in providing users with ad-hoc information requests.
- **Data Security.** The security of data is low in file based system because, the data is maintained in the flat file(s) is easily accessible.
- **Concurrency Problems**

FPS vs DBMS

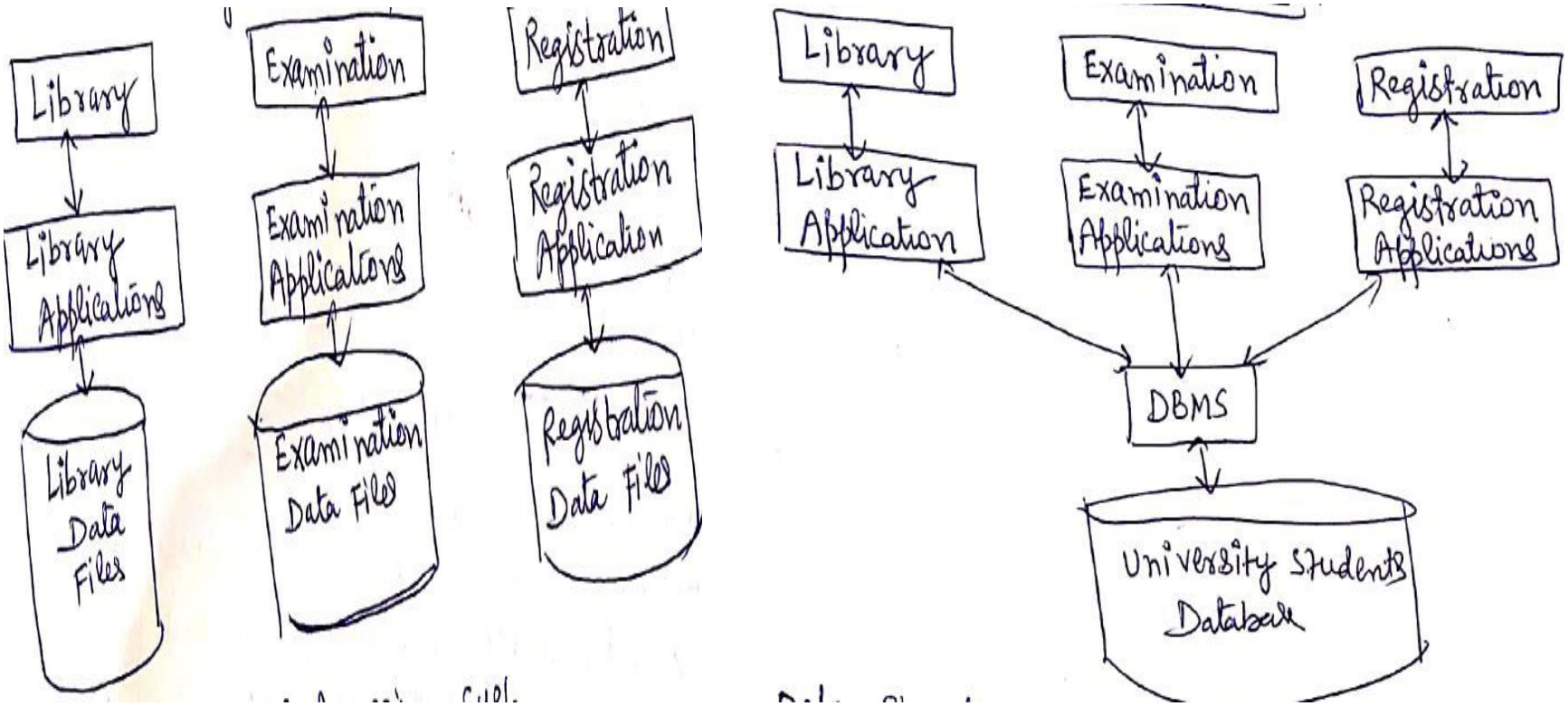


Figure 2: FPS vs DBMS

FPS vs DBMS

DBMS	FPS
Multi-user access	It does not support multi-user access
Design to fulfill the need for small and large businesses	It is only limited to smaller DBMS system.
Remove redundancy and Integrity	Redundancy and Integrity issues
Expensive. But in the long term Total Cost of Ownership is cheap	It's cheaper
Easy to implement complicated transactions	No support for complicated transactions

Database System Environment

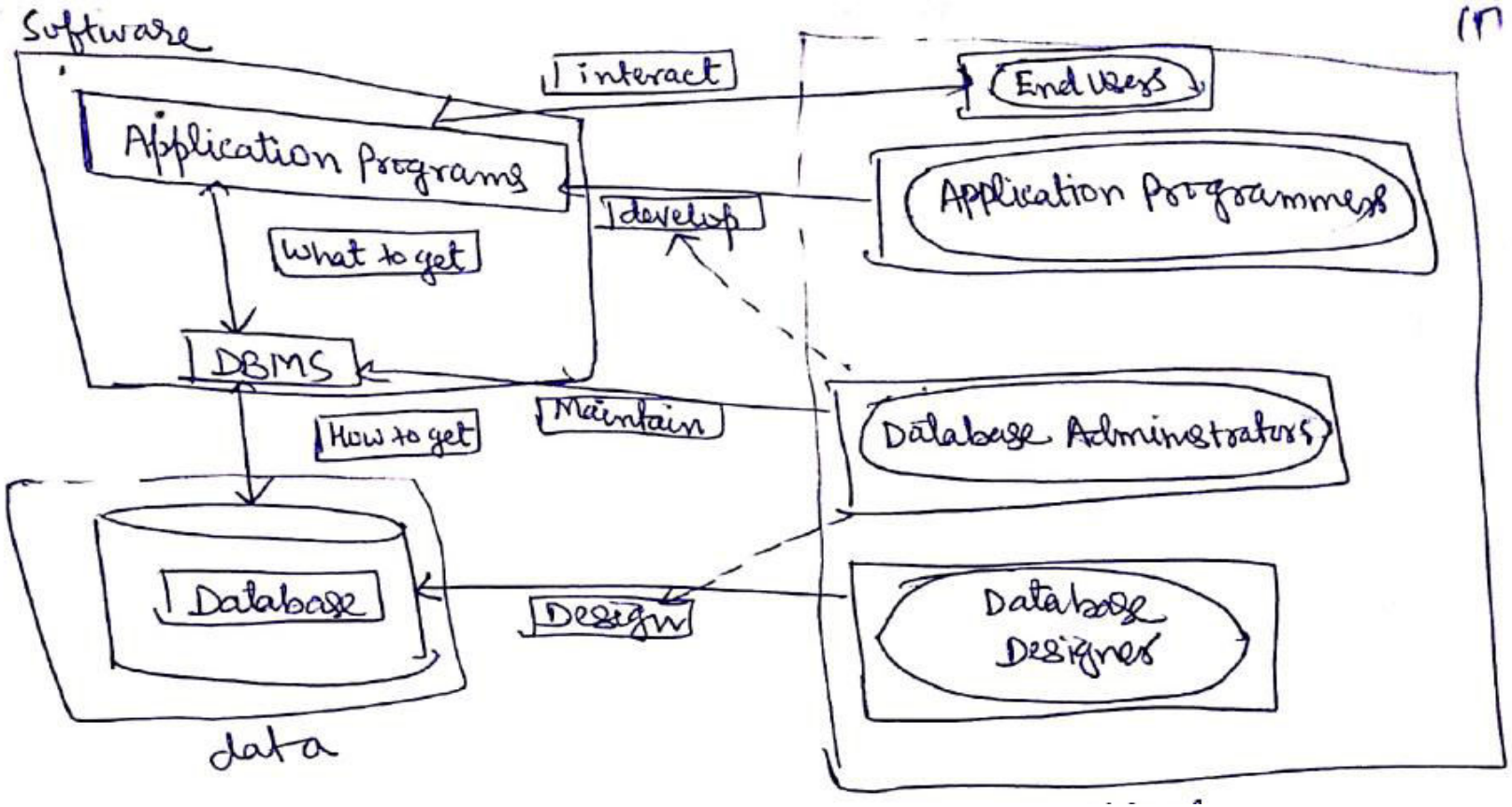


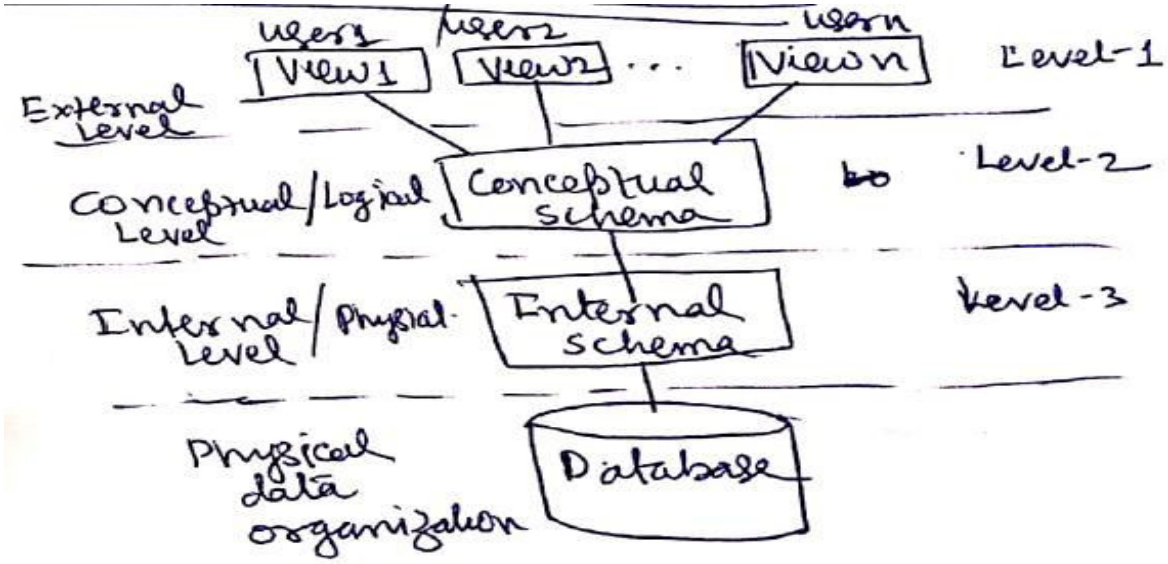
Figure 3: Database System Architecture

Users in Database System Environment

- **Application Programmers:** The Application programmers write programs in various programming languages to interact with databases.
- **Database Administrators:** Database Admin is responsible for managing the entire DBMS system. He/She is called Database admin or DBA.
- **End-Users:** The end users are the people who interact with the database management system. They conduct various operations on database like retrieving, updating, deleting, etc.

Database Schema / Data Abstraction

- A database schema is the skeleton structure that represents the logical view of the entire database.
- It defines how the data is organized and how the relations among them are associated. It formulates all the constraints that are to be applied on the data.
- It's the database designers who design the schema to help programmers understand the database and make it useful.



Database Schema

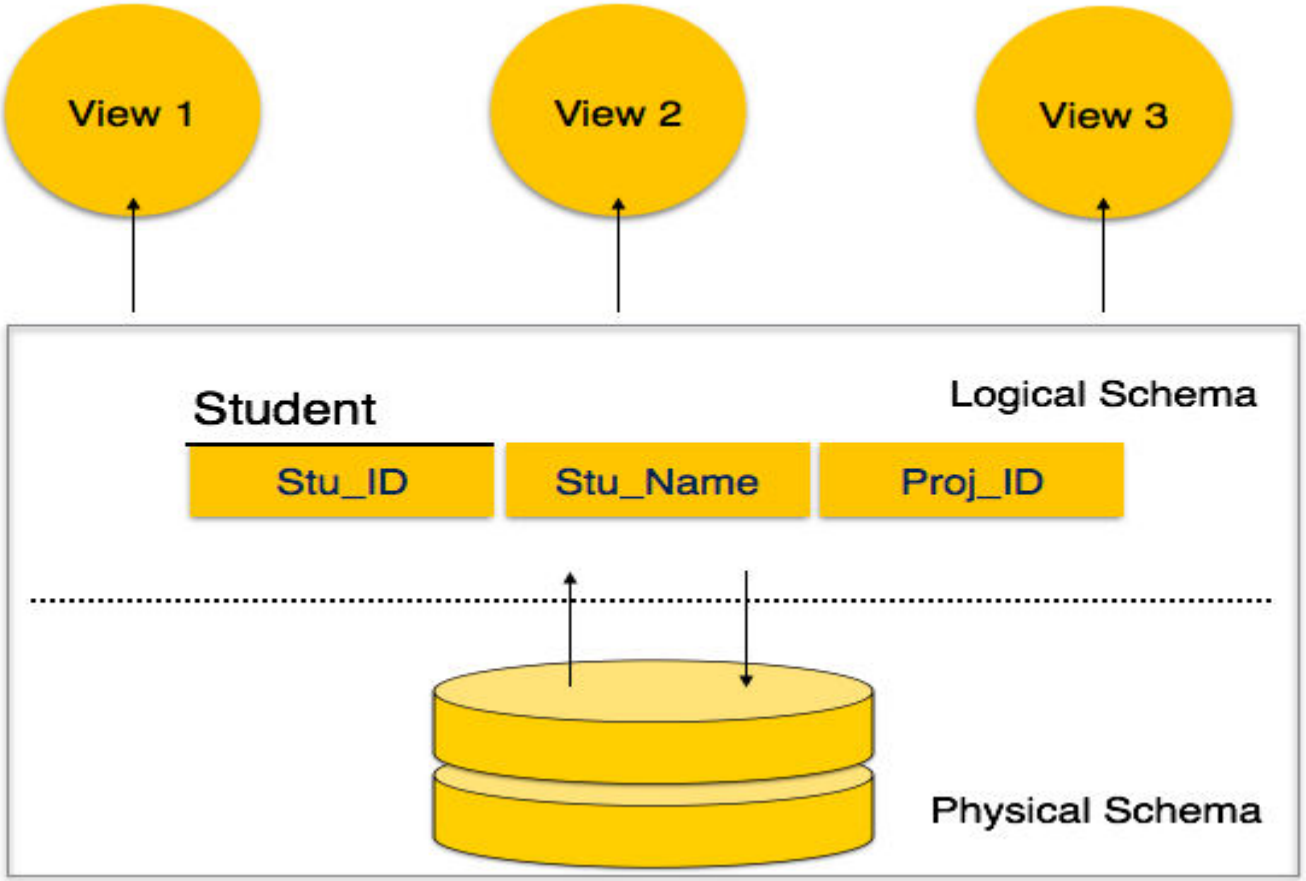


Figure 3: Database Schema

External View (Level-1)

* Top most level of DBMS architecture.
* represents the way each user group view the data
* Each user has a view of the db limited to the appropriate portion of the user's perspective of reality.
* Users may have different views of the same data eg. date, time, etc.
* DBMS uses external views to create user interface for different users which is both the facility and barrier.
* External Schema evolves as user needs are modified over time.

External View (Level-1)

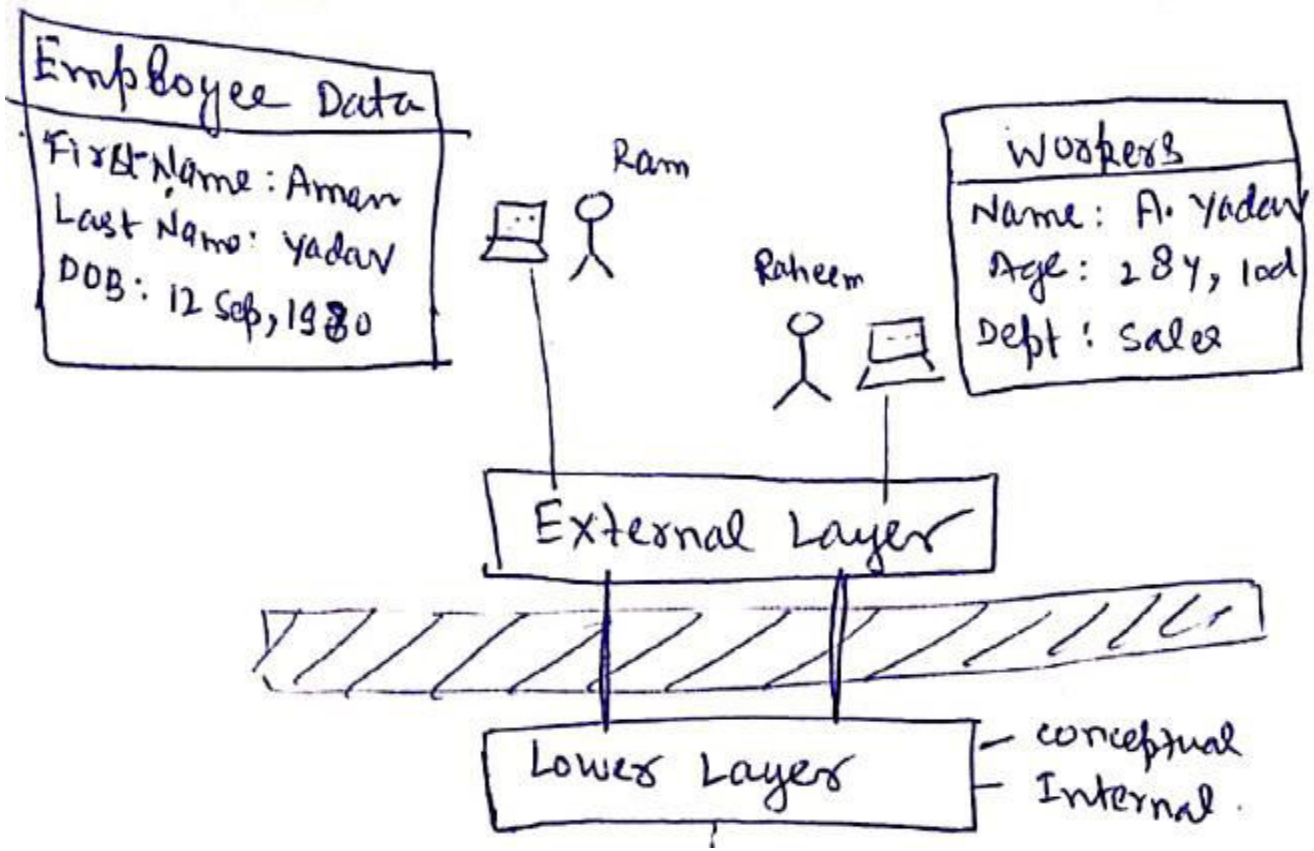


Figure 3: External View

Conceptual View/Logical Level (Level-2)

- Describes structure of whole database, entities, their relationships and constraints.

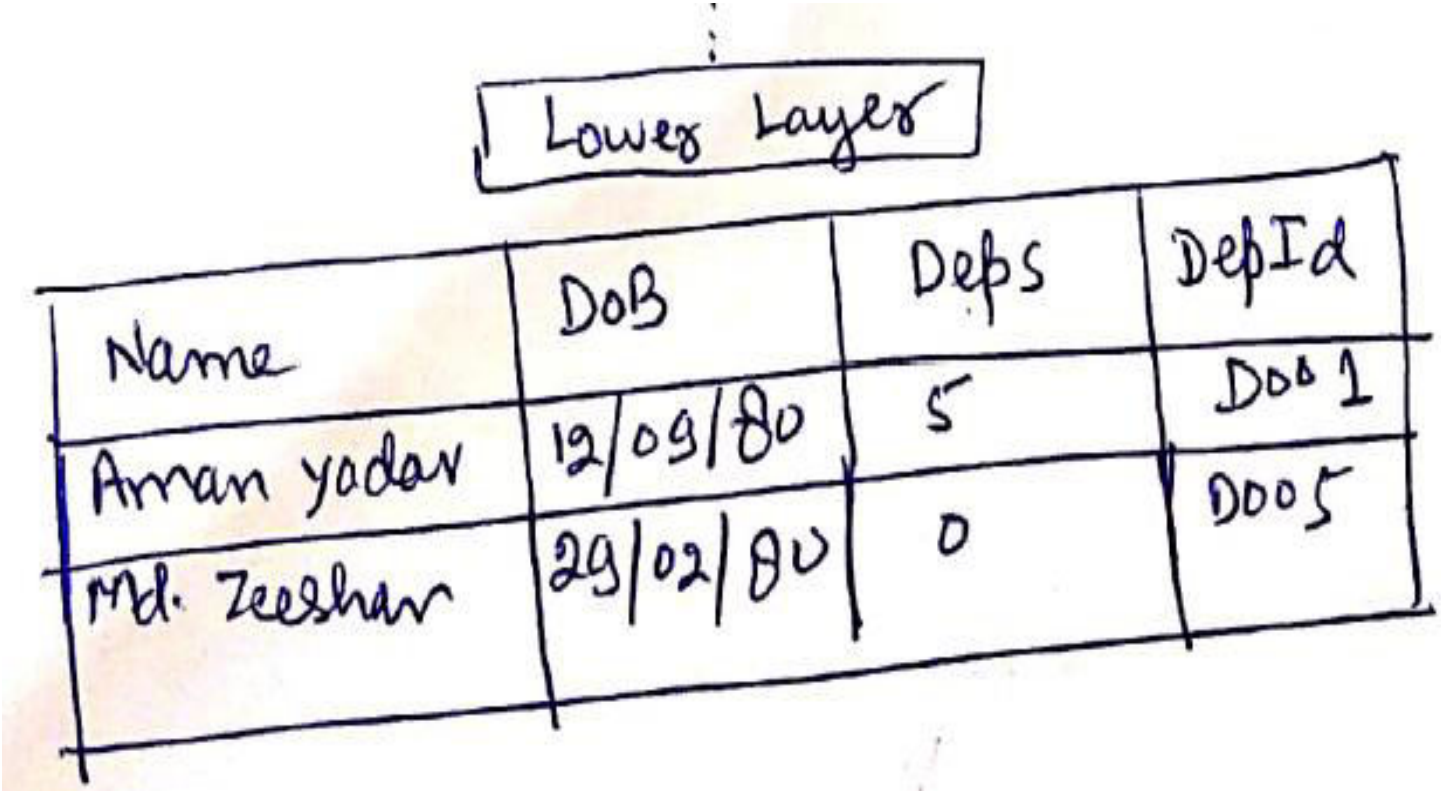


Figure 4: Conceptual View

Physical View/Internal Level (Level-3)

- This level describes how the data is actually stored in the storage devices. This level is also responsible for allocating space to the data. This is the lowest level of the architecture.
- It would also describe the data structures used by the database.

Data Independence

- Data independence refers characteristic of being able to modify the schema at one level of the database system without altering the schema at the next higher level.
- There are two types of data independence: Logical Data Independence and Physical Data Independence

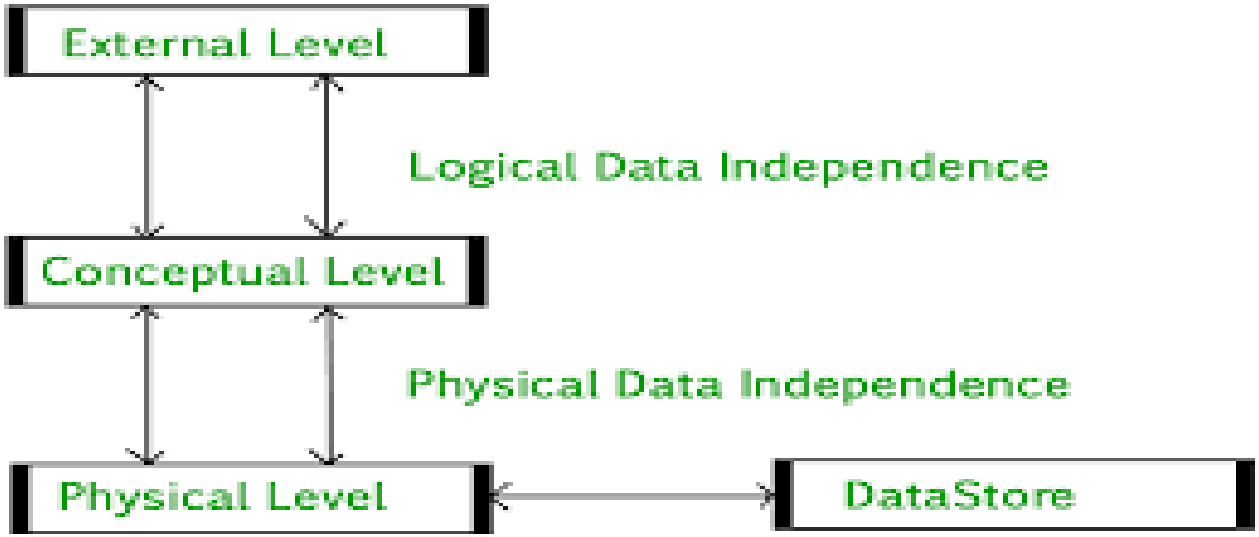


Figure 5: Data Independence

Logical Data Independence

- Logical data independence refers characteristic of being able to change the conceptual schema without having to change the external schema.
- Logical data independence is used to separate the external level from the conceptual view.
- If we do any changes in the conceptual view of the data, then the user view of the data would not be affected.
- Logical data independence occurs at the user interface level.

Physical Data Independence

- Physical data independence can be defined as the capacity to change the internal schema without having to change the conceptual schema.
- If we do any changes in the storage size of the database system server, then the Conceptual structure of the database will not be affected.
- Physical data independence is used to separate conceptual levels from the internal levels.
- Physical data independence occurs at the logical interface level.

Questions

- List four significant differences between a File-processing system and a DBMS?
- List four major advantages of a Database System.
- Explain the difference between physical and logical data independence?
- Write five main functions of a database administrator?
- list six major steps that you would take in setting up a database for a particular enterprise?
- Explain three levels of data abstraction?


Thank You