Chapter 1

Introduction

Database Management System (DBMS)
4th Semester , BEIT & BCE

Compiled by :

Ankit Bhattarai, Assistant Professor
ankitbhattarai@cosmoscollege.edu.np

Cosmos College of Management & Technology

Instructions

- **Modes of learning** : Lecture Slides, Google Jam board, Online test, Classwork's, Assignment.

- Try to make a **separate copy for DBMS** in which you can track the courses, do classwork & assignments on them.

  [NOTE: It's important as internal marks will be evaluated from it]

- Both of us are **new to the platform**, we need to have good coordination in order to make the platform effective. With time this shall will obviously be efficient.

Syllabus

Full marks: 100

Internal: 50
Final: 50

Internal Marks:

Theory: 30
Practical: 20

Unit	Contents	Hours	Remarks
1.	Introduction	4	
2.	Data Models	4	
3.	Relational Model	4	
4.	Relational Database Query Languages	8	Important
5.	Database Constraints and Relational Database Design	8	Important
6.	Security	3	
7.	Query Processing	3	
8.	File Organization & Indexing	4	
9.	Crash Recovery	3	
10.	Transaction Processing & Concurrency Control	4	
11.	Advanced Database Concepts	3	

Chapter 1: Introduction

1.1 Concepts & Applications

1.2 Objectives & Evolution

1.3 Needs of DBMS

1.4 Data Abstraction

1.5 Data Independence

1.6 Schema & Instances

1.7 Concept of DDL, DML & DCL

1.8 Database manager & users

1.1 Concepts and Applications

Database

- It is defined as collection of logically interrelated data and description of this data, designed to meet the information needs for organization.

- It is typically stored on disk and accessible by possibly many concurrent users. Databases are generally separated into application areas.

- Example: One database may contain HR(employee and payroll) data; another may contain sales data; another may contain accounting data; and soon. Databases are managed by a DBMS.

Continued…

Database Management System (DBMS)

- DBMS refers to the technology of storing and retrieving users data with utmost (maximum) efficiency along with appropriate security measures.

- In other words, It is a set of program/procedure that manages the database and provide an access to database in a form required by any application program.

- It is a collection of programs that allows users to specify the structure of a database, to create , query and modify the data in the database and to control access to it. (Example: limit access to the database so that only relevant staff can access details of enrolling students.)

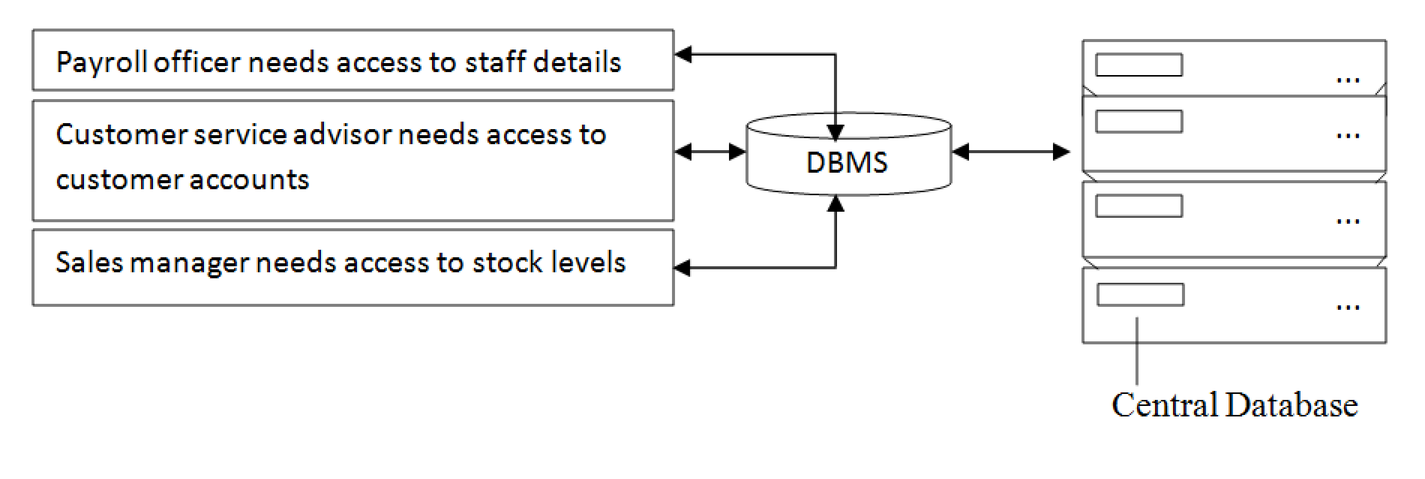
- A DBMS is a storage area that persist the data in files.

- DBMS software are MySQL, MSSQL, Oracle, MongoDB, Microsoft Access etc.

Continued…

Database System (Database + DBMS)

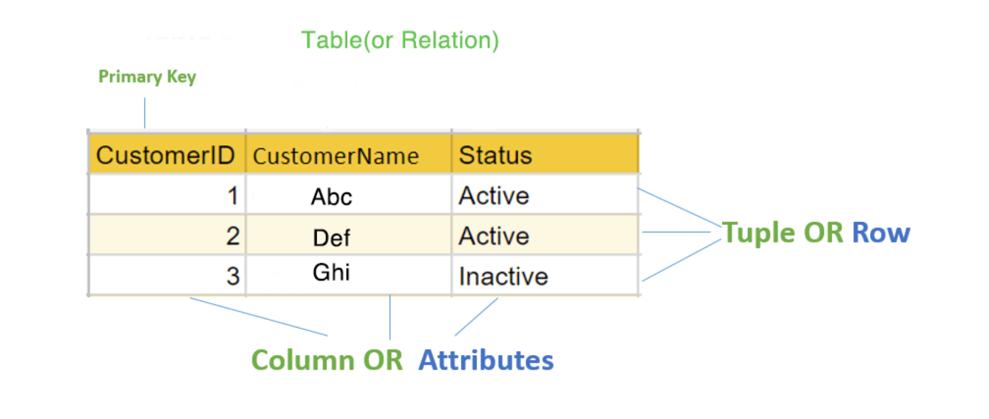
- Both database and database management system in combined form is called database system.
- It shows collection of related files with details about their definition, interpretation, manipulation and maintenance


Payroll officer needs access to staff details
Customer service advisor needs access to customer accounts
Sales manager needs access to stock levels
DBMS
...
...
...
...
Central Database

Continued…

RDBMS

- RDBMS stands for relational database management system.
- It is database management system based on the relational model *(In relational model, the data and relationships are represented by collection of inter-related tables.).*
- RDBMS is the most widely used & easier to implement and execute.
- Every RDBMS is a DBMS but not every DBMS, a RDBMS.
- RDMS Example systems are SQL Server, Oracle, MySQL, Maria DB and SQLite.
- RDBMS imposes integrity constraints.
- RDBMS stores the data in tabular form.



CustomerID	CustomerName	Status
1	Abc	Active
2	Def	Active
3	Ghi	Inactive

Example of a table or relation

Continued…

Applications of DBMS :

Databases are widely used. Some of representative applications are:

- **Banking:** For customer information, accounts and loans and banking transaction.
- **Universities:** For student registrations and grades.
- **Online shopping:** Everyone wants to shop from home,. Everyday new products are added and sold only with the help of DBMS. Purchase information, invoice bills and payment, all of these are done with the help of DBMS.
- **Airlines:** For reservation and schedule information.
- **Credit & transaction**: For purchase on credit cards and generation of monthly statements.

Continued…

Applications of DBMS :

- **Library Management System :** Maintain all the information related to book issue dates, name of the book, author and availability of the book.
- **Telecommunication :** For keeping records of call made, generating monthly bills, maintaining balances on prepaid calling cards.
- **Sales:** For customer, product and purchase information.
- **Finance :** For storing information about purchase information and purchase of financial instruments such as stocks and bonds.
- **Manufacturing :** For management of supply chain and for tracking production of items in factories, inventories of items and orders for items.

1.2. Objectives and Evolution

Objectives

- **Mass storage :** DBMS can store a lot of data in it for all the big firms. DBMS is really ideal technology to use. It can store thousands of records in it & one can fetch all that data whenever it is needed.
- **Remove Duplicate:** If you have lots of data then data duplicity will occur for sure at any instance. DBMS guarantees it that there will be no data duplicity among all the records. While storing new records, DBMS makes sure that same data was not inserted before.
- **Integrity :** Integrity means your data is authentic and consistent. DBMS has various validity checks that make your data completely accurate and consistent.
- **Platform Independent :**One can run DBMS at any platform. No particular platform is required to work on database management system.

Continued….

- **Data Protection:** Information such as bank details, employees salary details and sale purchase details should always be kept secured. Also all the companies need their data secured from unauthorized use. DBMS gives a master level security to their data. No one can alter or modify the information without the privileges of using that data.
- **Data backup & Recovery:** Sometimes database failure occurs so there is no option like one can say that all the data has been lost. There should be a backup of database so that on database failure it can be recovered. DBMS has the ability to back up and recover all the data in database.
- **Everyone can work on database:** There is no need to be a master of programming language if you want to work on DBMS. Any accountant who is having less technical knowledge can work on DBMS. All the definition and descriptions are given in it so that even a non-technical background person can work on it.

Evolution

Dates	Events
1950s & early 1960s	Data processing using magnetic tapes for storage
Late 1960s & 1970s	• Network & hierarchical data models for widespread use • Hard disks allowed for direct access of data • Ted Codd defines relational data model.
1980s	SQL(Sequential Query Language)becomes industrial standard Parallel & distributed database systems Object Oriented database systems
1990s	Large decision support & data mining applications Large multi terabyte data warehouses
Early 2000s	XML & XQuery Standard Automated database administration
Later 2000s	Giant data storage systems Google Big Table, Yahoo, Facebook

Continued….

When not to use database ?

- There may be a situation that involves unnecessary overhead cost of using a database system. The overhead cost of using database are:

1. High initial investment in hardware, software and training.
2. Overhead for providing security concurrency control, recovery etc.

Additional problem may arise if database designer and DBA (Database Administrator) don’t properly design the database. Hence, **file system** is used in the following situation due to:

1. The database and application are simple, small, well defined & not expected to change.
2. Multiple User access to data is not required.

1.3 Needs of DBMS

File Processing System (or Early Information System)

- In traditional approach, before to computer, all information was stored in papers
- After computer the data were stored in the form of files. The way we stored in files is similar to traditional file system
- For example, we have notepad file named Student_info on UI (User Interface), Then we can
  search file
  open file
  show file
  close file
- Now if we want to see specific student detail from the file, then we have to use different programming languages like C, C++ etc to get details.
- But the problem is , we have to write a lengthy program to complete this task or any other task like, open the file, append the file, modify the file, going to specific line of the file, delete the detail of file etc

Traditional FPS

- In this figure, we can see that each application program contains its own data files
- Due to which, common data files or data will be repeated due to non sharing of data.

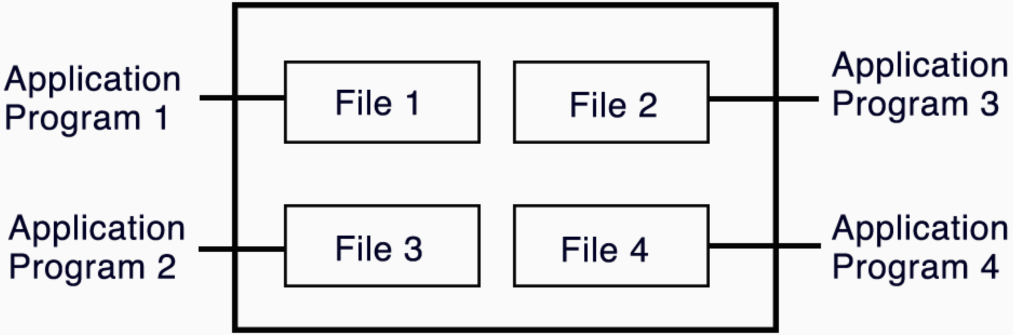
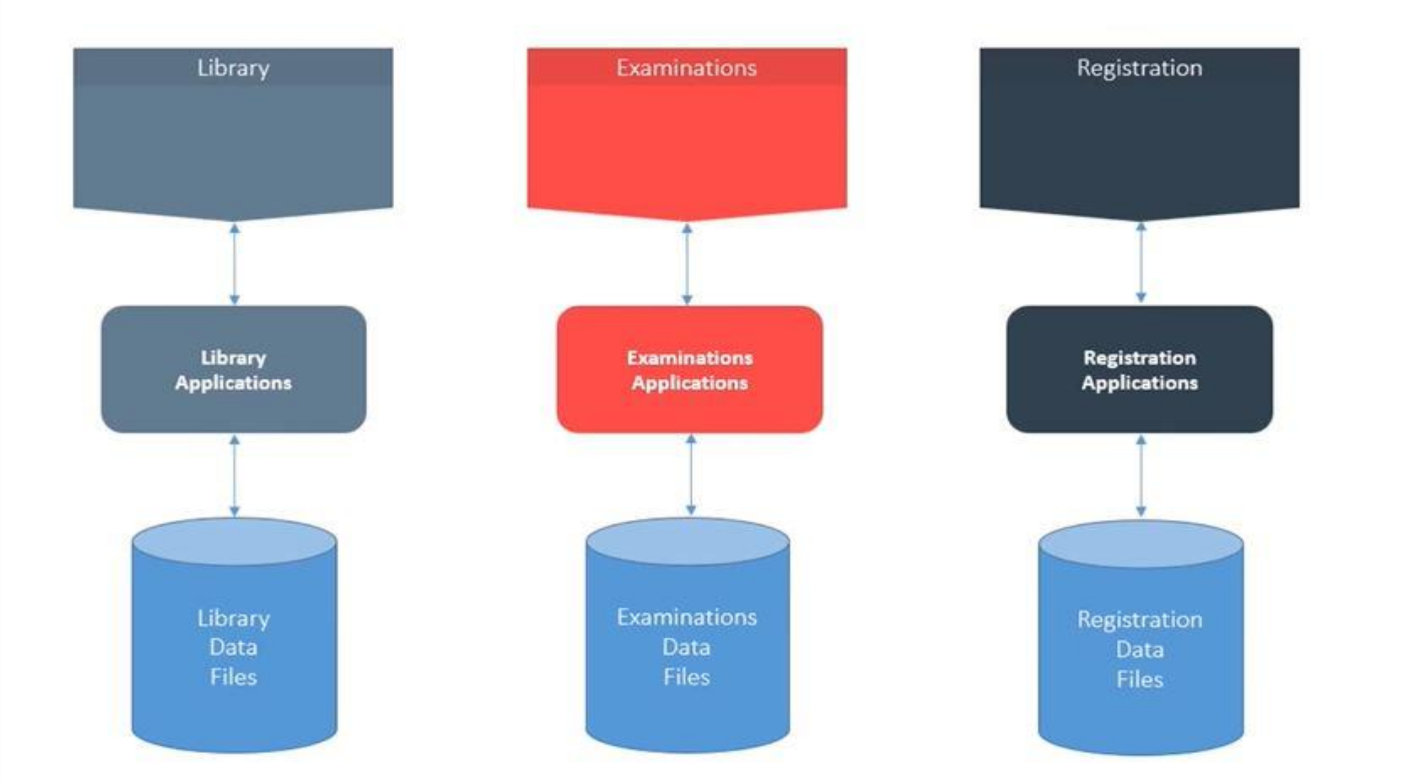


Fig. File Processing System

Continued….

- Example of FPS:



Advantages/ Disadvantages of FPS

Advantages of FPS:

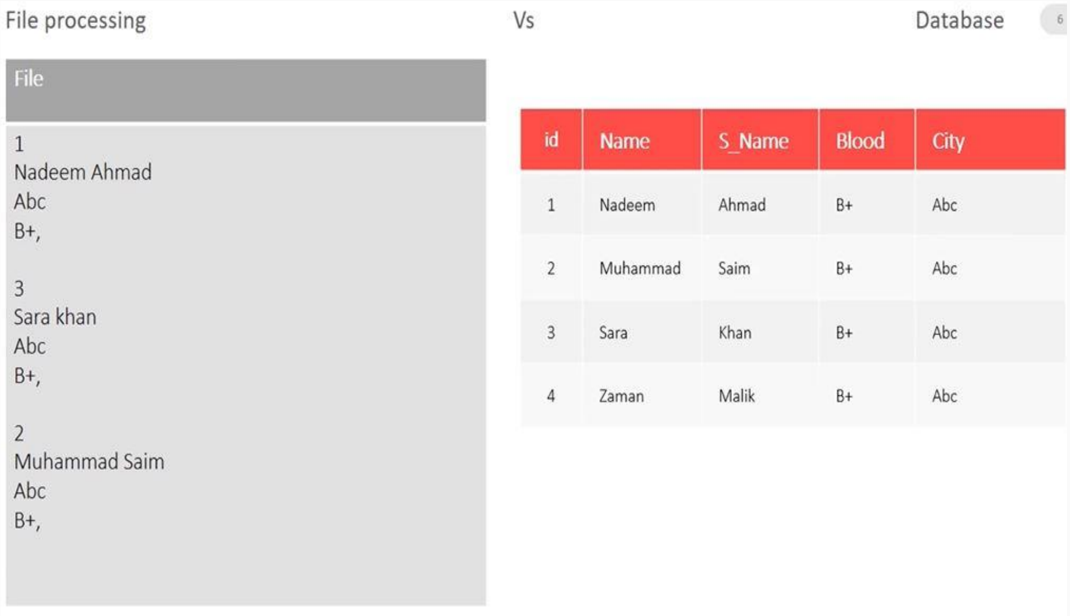
1. No need of external storage.
2. No need of highly technical person to handle the files or data.

Disadvantages of FPS:

1. Non sharing of data
2. Hard to update
3. Hard to handle
4. Difficulty to back up & migrate
5. Uncontrolled redundancy: Duplication of data
6. Time consuming

FPS vs DBMS

Comparison table



File processing Vs Database 6

File

1
Nadeem Ahmad
Abc
B+,

3
Sara khan
Abc
B+,

2
Muhammad Saim
Abc
B+,

id	Name	S_Name	Blood	City
1	Nadeem	Ahmad	B+	Abc
2	Muhammad	Saim	B+	Abc
3	Sara	Khan	B+	Abc
4	Zaman	Malik	B+	Abc

Advantages/ Disadvantages of DBMS

Advantages of DBMS:

1. **Minimal file redundancy**: Duplication of data is minimized. So the waste of storage is reduced.
2. **Sharing of data**: Data can be shared among the existing applications and newly developed application without having to create any additional stored data.
3. **Highly Secure**: It creates a security system that enforces user security and data privacy within the database.
4. **Backup & recovery**: Helps to create a copy of original data so that the data can be recovered in case of data loss.
5. **Easy in accessing data:** Helps non-technical person to easily manipulate data through the system.

Continued…

Disadvantages of DBMS:

1. **Extra cost of hardware & software** to facilitate DBMS.
2. **Failures of database** in case of centralized system may result to huge loss of data and money.
3. **Technical appointments** like programmers, database administrator, etc may be necessary in to manage DBMS which results in increase of system cost.
4. **Backup & recovery** complexity, as additional cost & manpower may be required.
5. **Memory size** must be large in order to implement DBMS.

Components of DBMS

A DBMS is an intermediate link between the physical database, computer and operating system

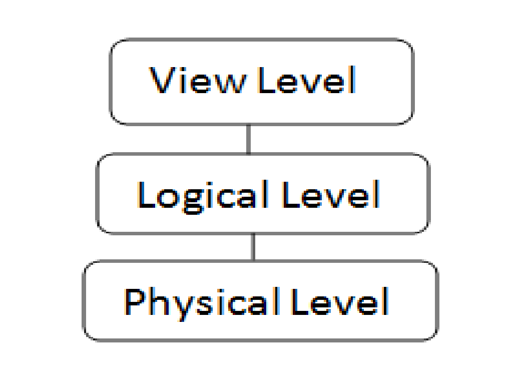
and on the other hand users. These components are:

1. **Hardware** : Computer system, memory elements, I/O devices
2. **Software**: Data can be shared among the existing applications and newly developed application without having to create any additional stored data.
3. **Data**: Actual data to work on it.
4. **Users:** Different users with different privileges
5. **Procedures:** Steps to work on the DBMS

1.4 Data Abstraction

Data Abstraction

- Data abstraction is a process of hiding the complexities from users (through several layers of abstraction) and makes users interaction with the system easier.
- For the system to be usable, it must retrieve data efficiently.
- The need for efficiency has led designer to use complex data structures to represent data in the database.
- Developers hide complexity from users through several levels of abstraction to simplicity user interactions with the system.



Continued…

Physical Level

- The lowest level of abstraction
- It describes how the data are actually stored.
- Database specialist use this level of abstraction.

Logical Level

- It describes what data are stored in the database and what relationship exist among those data.
- Describe the entire database in terms of small number of simple structure.
- Database administrator use logical level of abstraction.

Continued…

View Level

- The highest level of abstraction
- Database users uses view level to access only part of the entire database that they are authorized.
- The view level of abstraction exists to simplicity their interaction with the system.
- The system may provide many view for the database in order to hide the data for security purposes.

Continued…

Example of data abstraction:

```
 struct  staff
{
int staff_no;
int depart_no;
char name[30];
float salary;
};
```

Database specialist will know (Physical level)

staff_no	depart_no	name	salary

Database Users will see

(View level)

Continued…

Example of data abstraction:

staff_no	depart_no	name	salary
21	002	Ashwin	20000

- If we want to increase the salary of a staff_no= 21 by 10%, then the Database Administrator will update the data for the staff with staff_no = 21.

] (Logical level)

After Update,

staff_no	depart_no	name	salary
21	002	Ashwin	20000 + 10% of 20000

1.5 Data Independence

- Data Independence means upper levels are unaffected by changes in lower level.
- Changes in some part don’t seriously change or influence other.

- Two types:

1. Physical Data independence
2. Logical Data independence

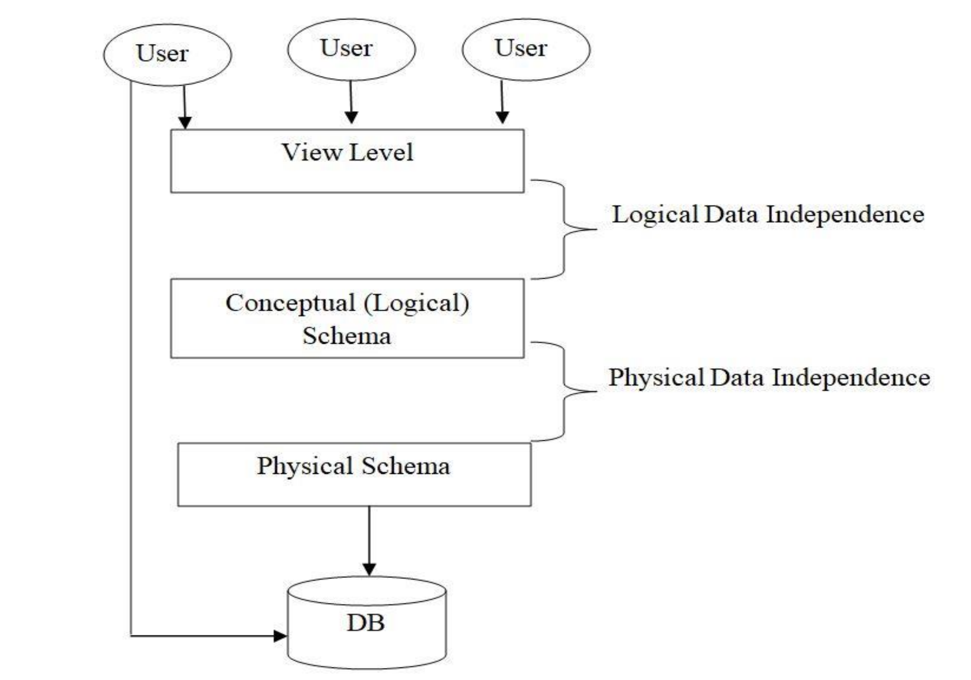

User
User
User
View Level
Logical Data Independence
Conceptual (Logical) Schema
Physical Data Independence
Physical Schema
DB

Fig. Data independence

Continued…

Physical Data independence

- It indicates that the physical storage structure or devices could be changed without affecting logical or conceptual schema.
- The ability to modify physical schema without changing the logical schema.
- Example:

  1. Change of location of database files from say C drive to D Drive or other drives.
  2. Using a new storage device like Hard Drive or Magnetic Tapes or SDD
  3. Changes that are performed in compression techniques or hashing algorithms.

Continued…

Logical Data independence

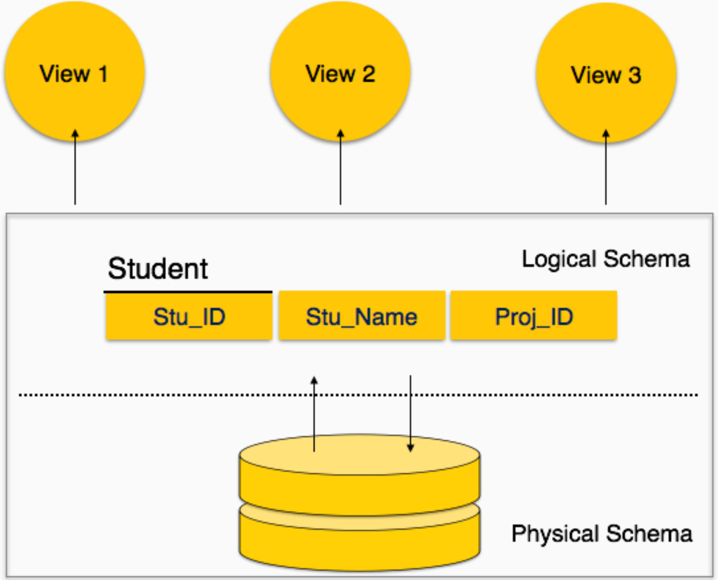
- The capacity to change the conceptual schema without having to change the external views or their application program.
- It is difficult as the retrieving of data is mainly dependent on the logical structure of data.
- Sometimes the changes in the Application program are needed if new fields are added or deleted from the database.
- Example:

    1. Add/Modify/Delete a new attribute, entity or relationship is possible without a change of existing application programs.
    2. Combining two records into one

1.6 Schema & Instances

Schema

- It defines how the data is organized and how the relations among them are associated.
- Its formulates all the constraints that are to be applied on the data.



Column Name	Data Type	Size
Stu_ID	int	4
Stu_Name	char (50)	50
Proj_ID	int	4

Fig. Logical Schema

Continued….

- **Physical Database Schema:** This schema pertains to the actual storage of data and its form of storage like files, indices, etc. It defines how the data will be stored in a secondary storage.

- **Logical Database Schema:** This schema defines all the logical constraints that need to be applied on the data stored. It defines tables, views, and integrity constraints.

- **Database Instance :** The data in the database at a particular moment in time is called a database state or snapshot. It is also called the current set of occurrences or instances in the database. Database instances tend to change with time.

1.7 Concept of DDL, DML, DCL & TCL

- A database system provides a data definition language to specify the database schema and a data manipulation language to express database queries and updates.

Data definition language (DDL)

- Deals with database schema and description of how the data should reside in the database.
- It is used to define create and modify the structure of database object in database.

Example:

CREATE - to create objects in database

Ex1: *Create database Test*; // database name is Test

Ex2:
```
CREATE TABLE Student(          id INT,
                                name VARCHAR(100),
                                age INT
                    );
```

id	name	age

Continued….

- **ALTER**

Alters (add, rename, drop change datatype) the structure of the existing database

Ex: *ALTER TABLE Student address(address VARCHAR(200));*

id	name	age	address

- **RENAME:** to rename database instances

  Ex: *RENAME TABLE Student to Students_info*; // Now table name is //Student_info

- **DROP:** delete objects from the database

  Ex: *DROP TABLE Student_info*; *//table will be deleted or dropped//*

Continued….

DML (Data manipulation language)

- It’s used to retrieve, store, modify, delete, insert and update data in the database.

INSERT: Insert data into a table

Ex: *INSERT INTO Student VALUES(101, ‘A’, 15, ‘ktm’);*

id	name	age	address
101	A	15	ktm

SELECT: Retrieve data from the database

Ex: *SELECT *from Student*;

id	name	age	address
101	A	15	ktm

Continued….

Where Clause

EX: *SELECT * from Student Where age=16*;

id	name	age	address

UPDATE: updates existing data in table. Ex: We have

id	name	age	address
101	A	15	ktm
102	B	15	pokhara
103	C	16	bhaktpur

In table we have to update age of id=102 from 15 to 17. Then

***UPDATE Student SET age=17 where id=102*;**

id	name	age	address
101	A	15	ktm
102	B	17	pokhara
103	C	16	bhaktpur

Continued….

DELETE: used to delete data from a table

Ex: *DELETE from Student;*// will delete all rows//

WHERE clause

DELETE from Student where id=103; //will delete a row which has id=103

id	name	age	address
101	A	15	ktm
102	B	17	pokhara

Data control language (DCL)

- It includes commands such as GRANT and REVOKE mostly concerned with rights, permissions and other controls of the database system.

  GRANT - allows users access privileges to database

  REVOKE - withdraw users access privileges given using the GRANT command.

Continued….

Transaction control language (TCL)

- Deals with a transaction within a database
- TCL commands are to keep a check on other commands and their affect on the database
- These commands can annul changes made by other. It can also make changes permanent

Example **: COMMIT** - to permanently save or save work done in transactions

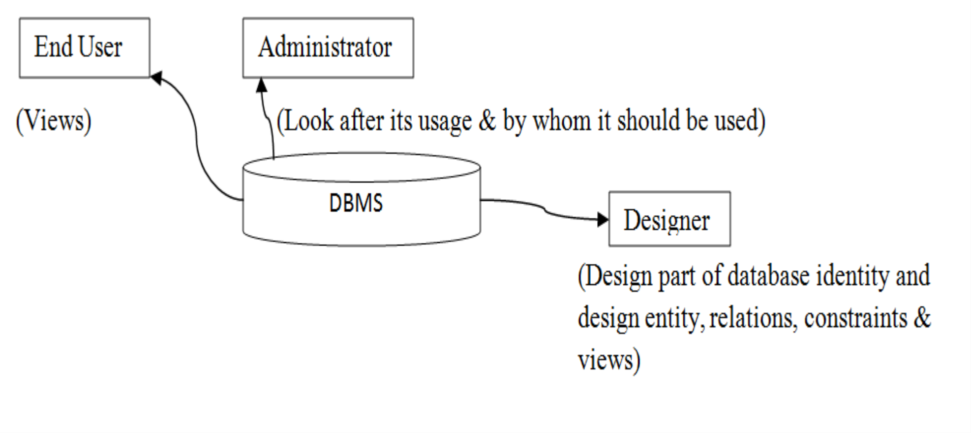
ROLLBACK - to undo changes or restores database to original state since the last COMMIT command in transactions

SAVE POINT or SAVE TRANSACTION- to save temporarily or to create a save point within a transaction.

1.8 Database manager & users

Database Administrator

- It's the person responsible for overall control of the system at technical level.
- They look after DBMS resources like system license, require tools and software and other hardware related maintenance.
- They are the one responsible for granting authorization to data access to different users.


End User
Administrator
(Views)
(Look after its usage & by whom it should be used)
DBMS
Designer
(Design part of database identity and design entity, relations, constraints & views)

1.8 Database manager & users

Database Designer

- They are the group of people who actually work on designing part of the database.
- They are responsible on what data should be kept and in what format.
- They identify and deign the whole set of entities, objects, relations, constraints and views.

End Users

- They are those who get the benefit of using the database system.
- They can range from simple to high level business analyst.

# 3-tier architecture

- **Three layers**:

  Presentation Tier : *HTML/CSS*

  Application Tier: *PHP/JAVA*

  Database Tier: *MYSQL/SQL*

- **Why do we need it ?**

i. Separation of user applications and physical database which enables data independence.
ii. Proposed to support DBMS characteristics & its advantages
iii. Helps in enabling security features.
iv. Multiple views support of the data

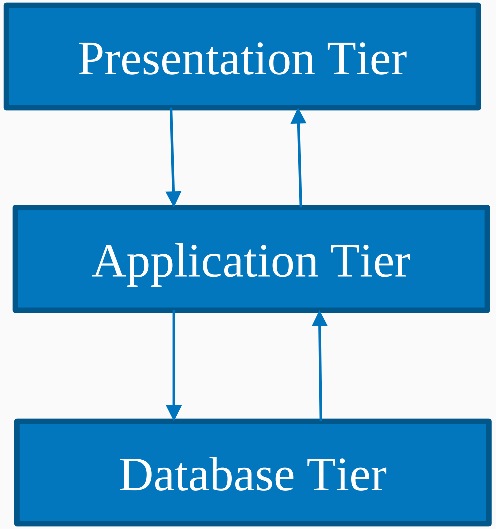


Fig. 3-tier architecture

# Steps in setting up a database:

1. Define the **high level requirement** of the application. (This generates a document: software requirement specification).
2. Define a model containing all appropriate types of data & data relationship
3. Define the integrity constraint on the data,
4. Define the physical level.
5. For each known problem to be solved on a regular basis define a user interface, format to carry out the task and write the necessary application program format to implement a user interface.
6. Create or initialize the database.

# Steps in setting up a database:

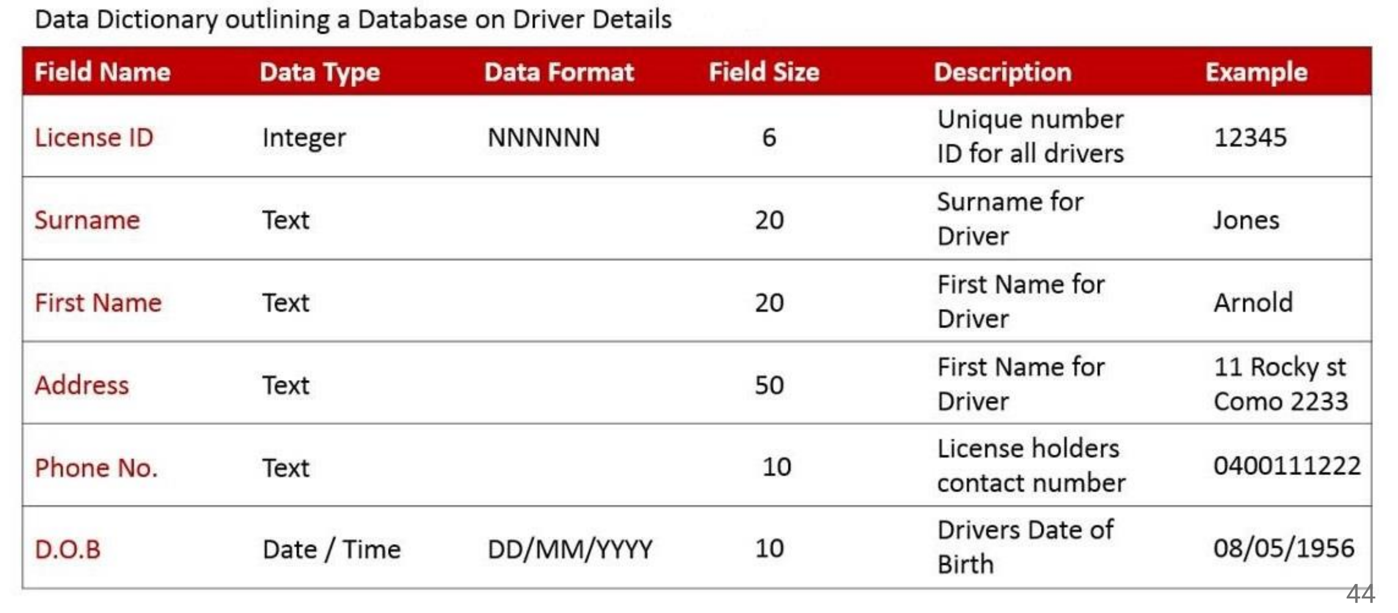
Question (Classwork)

As an Computer/IT engineer of TIA you are required to set up a **airline database system**. Prepare a report that contains the steps in setting up the database.
(Try to give examples in steps that you defined)

# Data Dictionary:

- A data dictionary is a software module and database containing descriptions and definitions concerning the structure, data elements, interrelationships, and other characteristics of an organization's database.

Example:



Data Dictionary outlining a Database on Driver Details

Field Name	Data Type	Data Format	Field Size	Description	Example
License ID	Integer	NNNNNN	6	Unique number ID for all drivers	12345
Surname	Text		20	Surname for Driver	Jones
First Name	Text		20	First Name for Driver	Arnold
Address	Text		50	First Name for Driver	11 Rocky st Como 2233
Phone No.	Text		10	License holders contact number	0400111222
D.O.B	Date / Time	DD/MM/YYYY	10	Drivers Date of Birth	08/05/1956

# Management Information System (MIS):

- MIS is a general term for software designed to facilitate the storage, organization and retrieval of information.
- MIS gives the business manager the information that they need to make decision.
- MIS provide a variety of information to managers.
- Periodic Scheduled Report, Exception report, Demand report and response report etc. are the part of MIS in business enterprise.

# Questions:

1. Point out the applications of DBMS. Mention about different types of users that uses database
2. Define RDMS. Compare and contrast DDL, DML and DCL with examples of each.
3. Differentiate between data and information. Explain the reasons of overhead cost of using DBMS in simple and small database.
4. As a database designer of an IT company you are assigned to carry out the task of designing a database for an application. Now list out the steps that you would take in setting up a database for that application.
5. Trace the evolution of database system & the needs to use this type of system in the present context.
6. Define RDMS. Explain the statement “DBA has central control over the database system’.
7. Compare & contrast DDL, DML and DCL with examples of each.

# Questions:

12. Differentiate between DBMS and file system. Also, mention the case when not to use the database system.

13. Mention the Role of DBA and explain different types of database users.

14. Define database system. How does DBMS differ than FPS?

15. Write short notes :

i. Data Abstraction

ii. Database Administrator

iii. Data Independence

iv. Pros & cons of database

v. FPS (File Processing System)

THANK YOU

Any Queries ?