Introduction to Database Systems

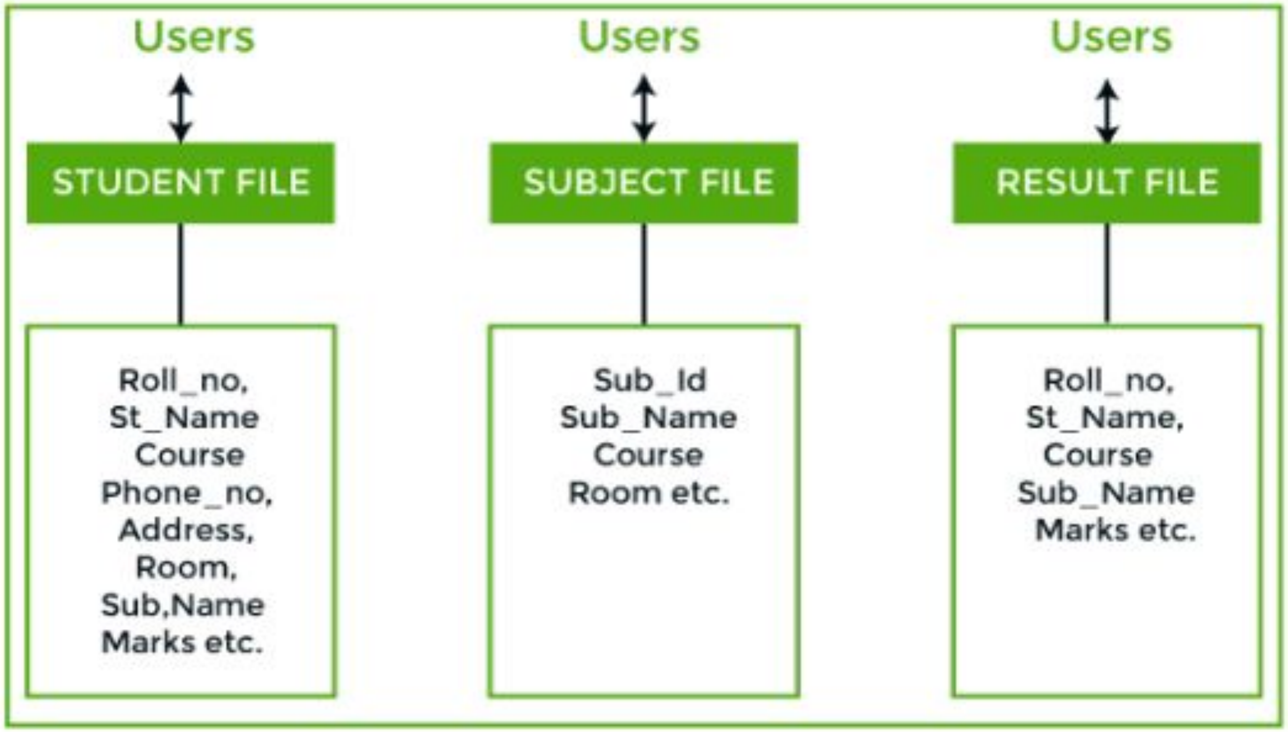
Database Management Systems

Introduction: Database System, Characteristics (Database Vs File System), Advantages of Database Systems, Database Applications, Database Users, Brief Introduction of Different Data Models; Concepts of Schema, Instance and Data Independence; Three Tier Schema Architecture for Data Independence; Database System Structure, Centralized and Client Server Architecture for Database Systems.

Entity Relationship Model: Introduction, Concept of Entities, Attributes, Entity Sets, Relationships, Relationship Sets, Key and Participation Constraints, Class Hierarchies, and Aggregation, Developing E-R diagrams for Databases.

Database Management Systems Vs FMS

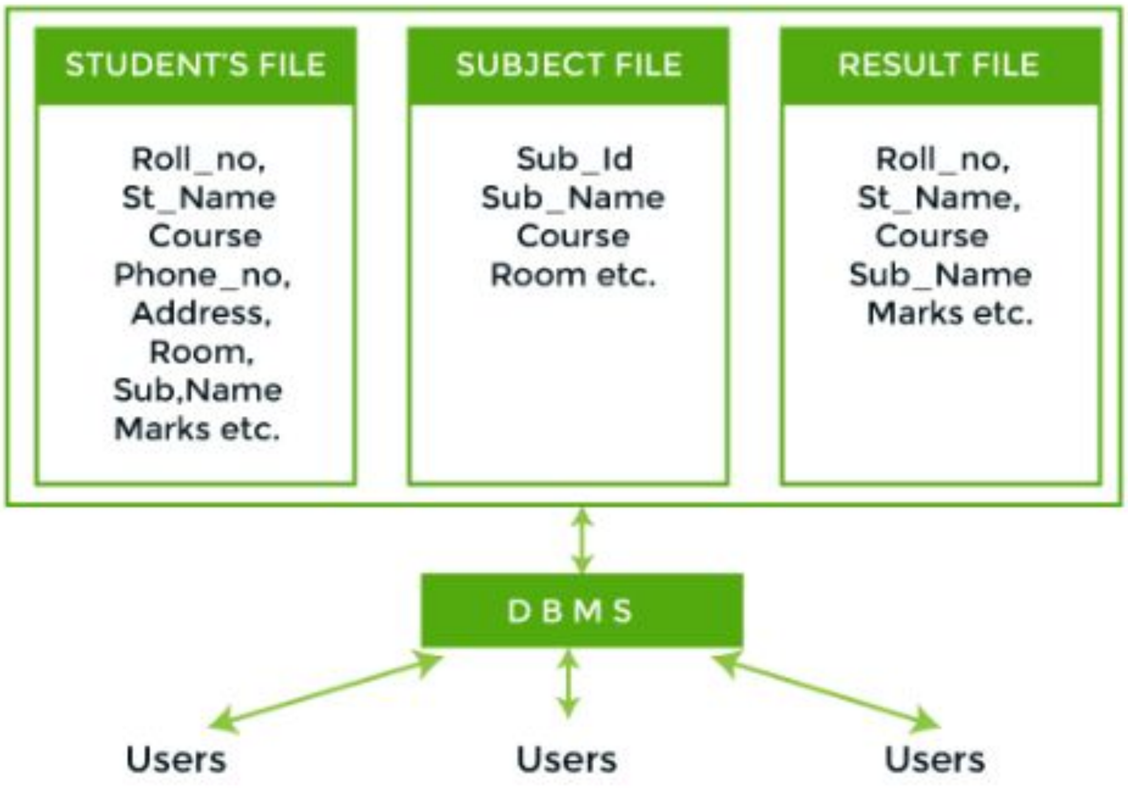
FMS



Users
Users
Users
STUDENT FILE
SUBJECT FILE
RESULT FILE
Roll_no, St_Name Course Phone_no, Address, Room, Sub,Name Marks etc.
Sub_Id Sub_Name Course Room etc.
Roll_no, St_Name, Course Sub_Name Marks etc.

Database Management Systems Vs FMS

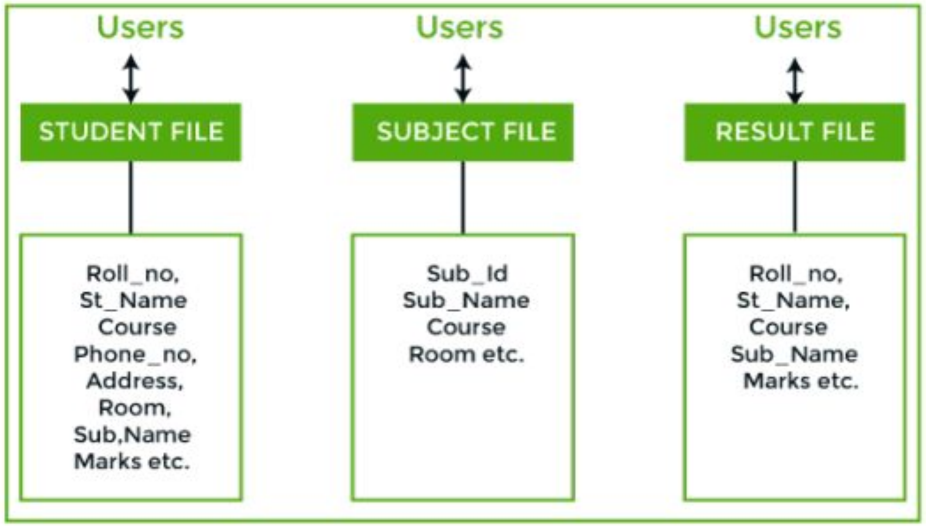
DBMS


STUDENT'S FILE
Roll_no, St_Name Course Phone_no, Address, Room, Sub,Name Marks etc.
SUBJECT FILE
Sub_Id Sub_Name Course Room etc.
RESULT FILE
Roll_no, St_Name, Course Sub_Name Marks etc.
DBMS
Users
Users
Users

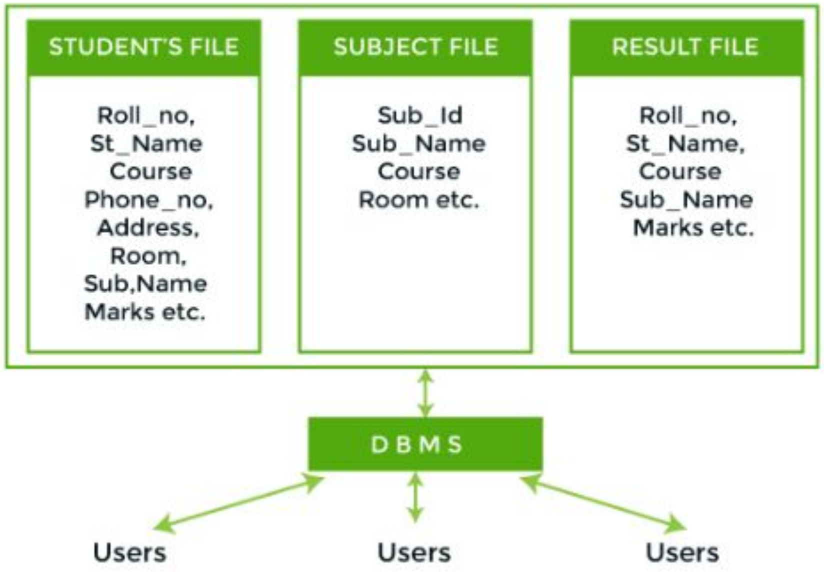
Database Management Systems Vs FMS

FMS

DBMS



Users
Users
Users
STUDENT FILE
SUBJECT FILE
RESULT FILE
Roll_no, St_Name Course Phone_no, Address, Room, Sub,Name Marks etc.
Sub_Id Sub_Name Course Room etc.
Roll_no, St_Name, Course Sub_Name Marks etc.



STUDENT'S FILE
SUBJECT FILE
RESULT FILE
Roll_no, St_Name Course Phone_no, Address, Room, Sub,Name Marks etc.
Sub_Id Sub_Name Course Room etc.
Roll_no, St_Name, Course Sub_Name Marks etc.
DBMS
Users
Users
Users

Basis	DBMS Approach	File System Approach
Meaning	DBMS is a collection of data. In DBMS, the user is not required to write the procedures.	The file system is a collection of data. In this system, the user has to write the procedures for managing the database.
Sharing of data	Due to the centralized approach, data sharing is easy.	Data is distributed in many files, and it may be of different formats, so it isn't easy to share data.
Data Abstraction	DBMS gives an abstract view of data that hides the details.	The file system provides the detail of the data representation and storage of data.
Security and Protection	DBMS provides a good protection mechanism.	It isn't easy to protect a file under the file system.
Recovery Mechanism	DBMS provides a crash recovery mechanism, i.e., DBMS protects the user from system failure.	The file system doesn't have a crash mechanism, i.e., if the system crashes while entering some data, then the content of the file will be lost.
Manipulation Techniques	DBMS contains a wide variety of sophisticated techniques to store and retrieve the data.	The file system can't efficiently store and retrieve the data.
Concurrency Problems	DBMS takes care of Concurrent access of data using some form of locking.	In the File system, concurrent access has many problems like redirecting the file while deleting some information or updating some information.
Where to use	Database approach used in large systems which interrelate many files.	File system approach used in large systems which interrelate many files.
Cost	The database system is expensive to design.	The file system approach is cheaper to design.
Data Redundancy and Inconsistency	Due to the centralization of the database, the problems of data redundancy and inconsistency are controlled.	In this, the files and application programs are created by different programmers so that there exists a lot of duplication of data which may lead to inconsistency.

Basis	DBMS Approach	File System Approach
Structure	The database structure is complex to design.	The file system approach has a simple structure.
Data Independence	In this system, Data Independence exists, and it can be of two types. ◦ Logical Data Independence ◦ Physical Data Independence	In the File system approach, there exists no Data Independence.
Integrity Constraints	Integrity Constraints are easy to apply.	Integrity Constraints are difficult to implement in file system.
Data Models	In the database approach, 3 types of data models exist: ◦ Hierarchal data models ◦ Network data models ◦ Relational data models	In the file system approach, there is no concept of data models exists.
Flexibility	Changes are often a necessity to the content of the data stored in any system, and these changes are more easily with a database approach.	The flexibility of the system is less as compared to the DBMS approach.
Examples	Oracle, SQL Server, Sybase etc.	Cobol, C++ etc.

Building an Application with a DBMS

- Requirements gathering (natural language, pictures)
- Requirements modeling (conceptual data model, ER)
  - Decide what *entities* should be part of the application and how they should be *related*
- Schema design and implementation
  - Decide on a set of *tables*, *attributes*
  - Create the tables in the database system
  - Populate database (insert records/tuples)
- Write application programs using the DBMS
  - … a lot easier now that
    the data management is taken care of

Database Management System (DBMS)

DBMS contains information about a particular enterprise

- Collection of interrelated data
- Set of programs to access the data
- An environment that is both *convenient* and *efficient* to use

Database Applications:

- Banking: 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.

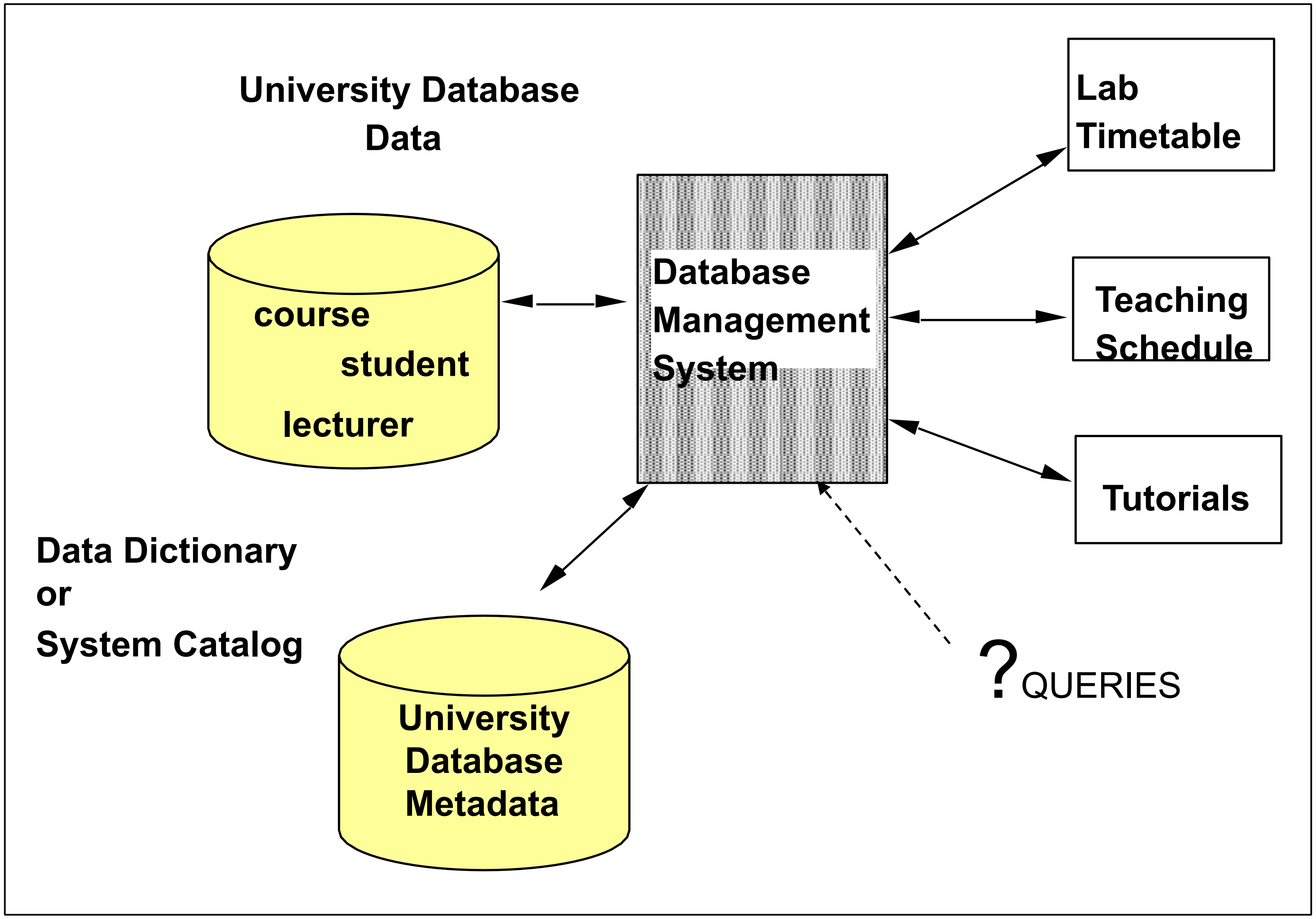
Databases touch all aspects of our lives

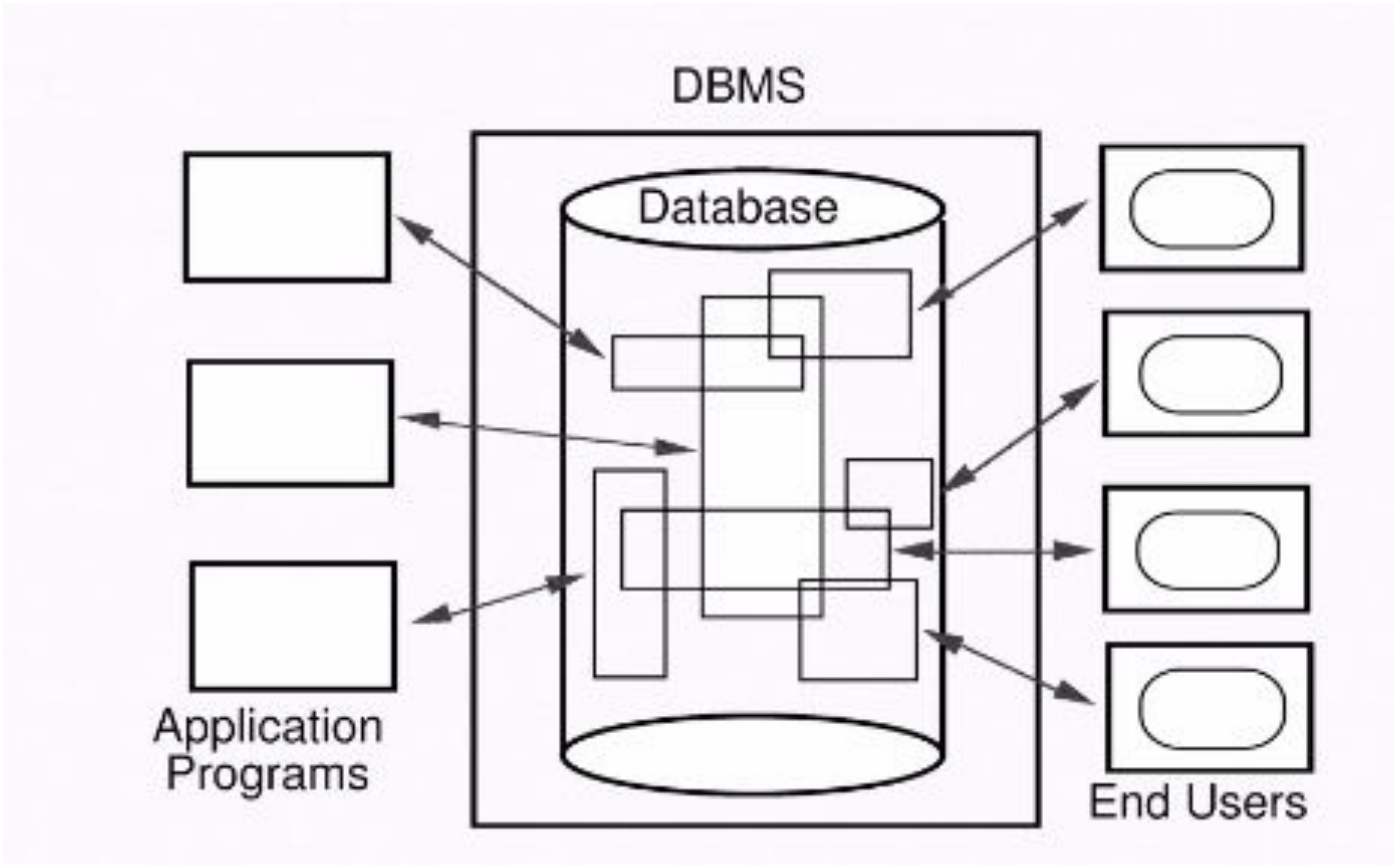
University Database Example

Application program examples

Add new students, instructors, and courses
Register students for courses, and generate class rosters
Assign grades to students, compute grade point averages (GPA) and generate transcripts

DBMS: A Logical Interface

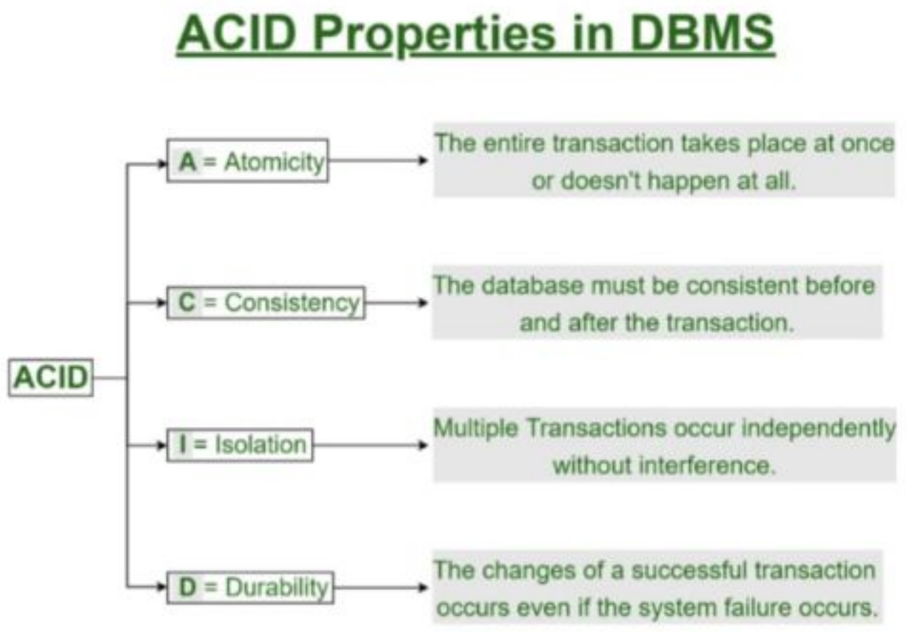



DBMS
Database
Application Programs
End Users

Characteristics of DBMS

Characteristics of DBMS

- It uses a digital repository established on a server to store and manage the information.
- It can provide a clear and logical view of the process that manipulates data.
- DBMS contains automatic backup and recovery procedures.
- It contains ACID properties which maintain data in a healthy state in case of failure

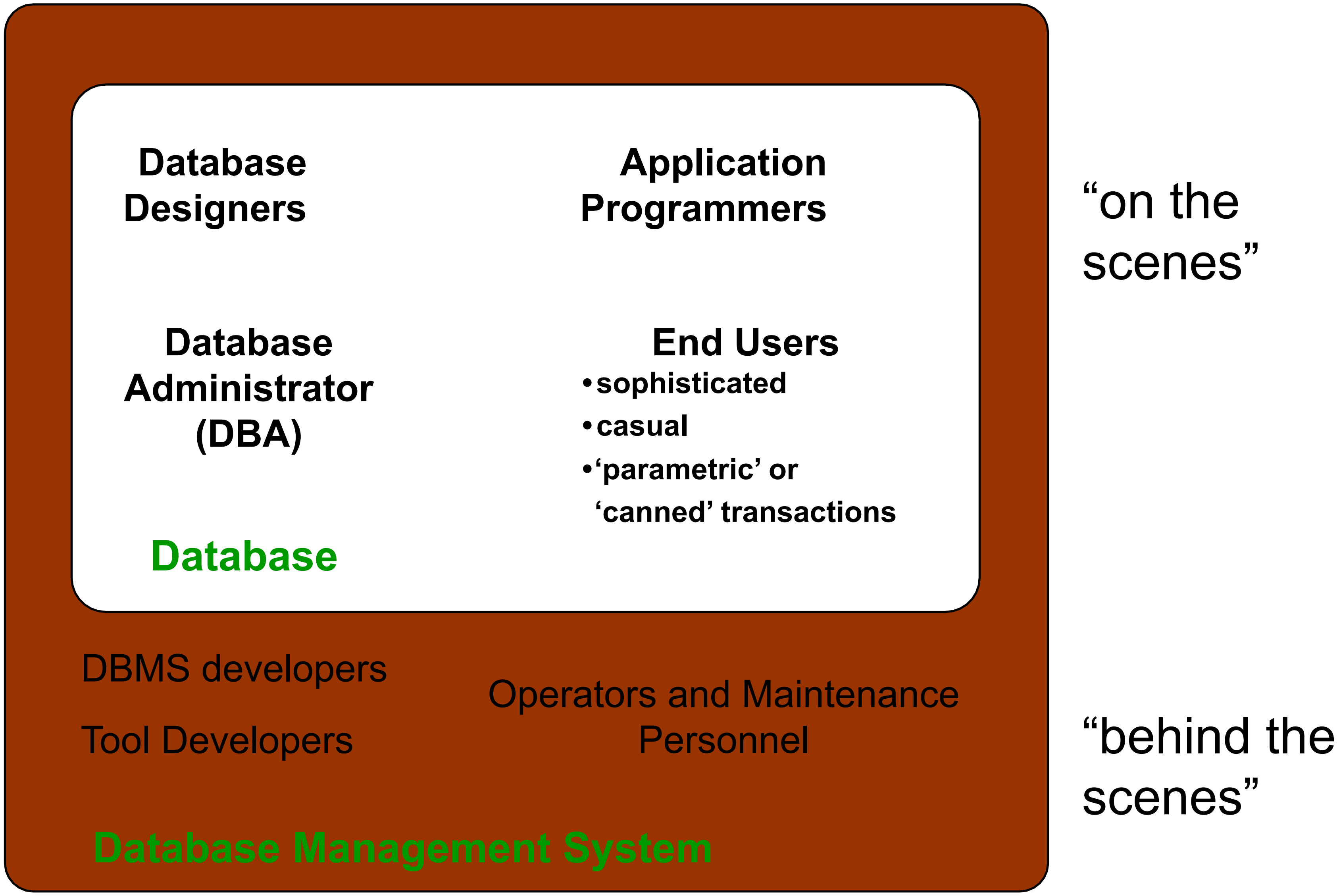

ACID Properties in DBMS
ACID
A = Atomicity
The entire transaction takes place at once or doesn't happen at all.
C = Consistency
The database must be consistent before and after the transaction.
I = Isolation
Multiple Transactions occur independently without interference.
D = Durability
The changes of a successful transaction occurs even if the system failure occurs.

Characteristics of DBMS

Characteristics of DBMS

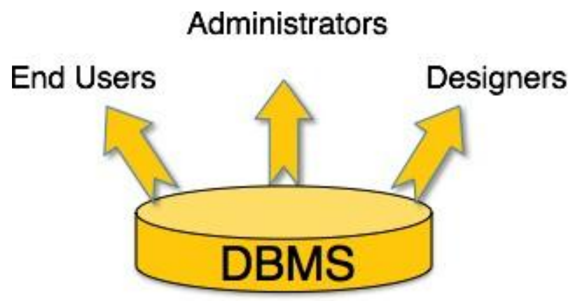
- It can reduce the complex relationship between data.
- It is used to support manipulation and processing of data. It is used to provide security of data.
- It can view the database from different viewpoints according to the requirements of the user.

Database Actors / Users



Roles in DBMS

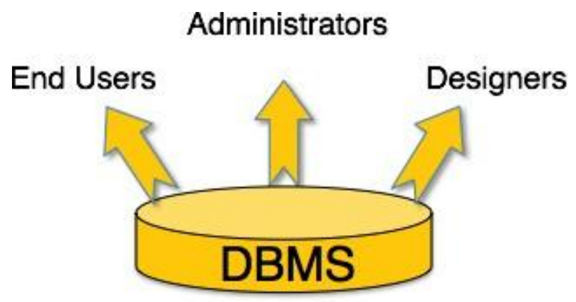
DBMS has users with different rights and permissions who use it for different purposes. Some users retrieve data and some back it up. The users of a DBMS can be broadly categorized as follows :



- **Administrators** : One who maintains the DBMS and are responsibility's are :
  - Administrating the Database
  - To look after its usage and by whom it should be used and Create access profiles for users.
  - Apply limitations to maintain isolation and force security.
  - System license, required tools, and Other software and hardware related maintenance

Roles in DBMS

DBMS has users with different rights and permissions who use it for different purposes. Some users retrieve data and some back it up. The users of a DBMS can be broadly categorized as follows :



- **Designers :** are the group of people who actually work on the designing part of the database. They keep a close watch on **what data should be kept and in what format.** They identify and design the whole set of entities, relations, constraints, and views.
- **Users :** End users are those who actually reap the benefits of having a DBMS. End users can range from simple viewers who pay attention to the logs or market rates to sophisticated users such as business analysts.

Historical Development and Database Technology

Early Database Applications: Hierarchical and Network Models (in mid 1960’s).

Relational Model based Systems: Researched and experimented with in IBM and the universities (in 1970).

Object-oriented applications: OODBMSs (in late 1980’s and early 1990’s)

History of Data Models

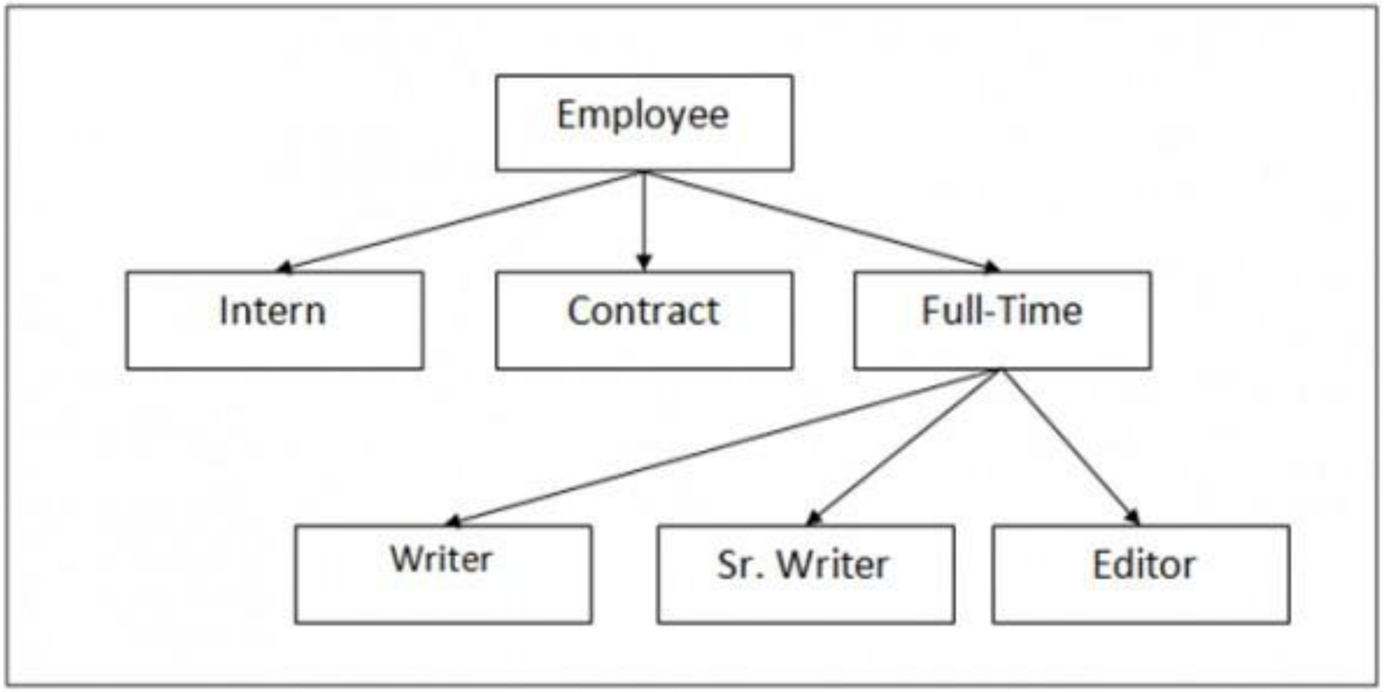
Hierarchical Data Model: implemented in a joint effort by IBM and North American Rockwell around 1965. Resulted in the IMS family of systems. The most popular model. Other system based on this model: System 2k (SAS inc.)

Network Model: the first one to be implemented by Honeywell in 1964-65 (IDS System). Adopted heavily due to the support by CODASYL (CODASYL - DBTG report of 1971). Later implemented in a large variety of systems - IDMS (Cullinet - now CA), DMS 1100 (Unisys), IMAGE (H.P.), VAX -DBMS (Digital
Equipment Corp.).

Relational Model: proposed in 1970 by E.F. Codd (IBM), first commercial system in 1981-82. Now in several commercial products (DB2, ORACLE, SQL Server, SYBASE, INFORMIX)

Hierarchical Data Model

- In Hierarchical Model, a hierarchical relation is formed by collection of relations and forms a
- tree-like structure.
- The relationship can be defined in the form of parent child type. One of the first and most popular Hierarchical Model is Information Management System (IMS), developed by IBM

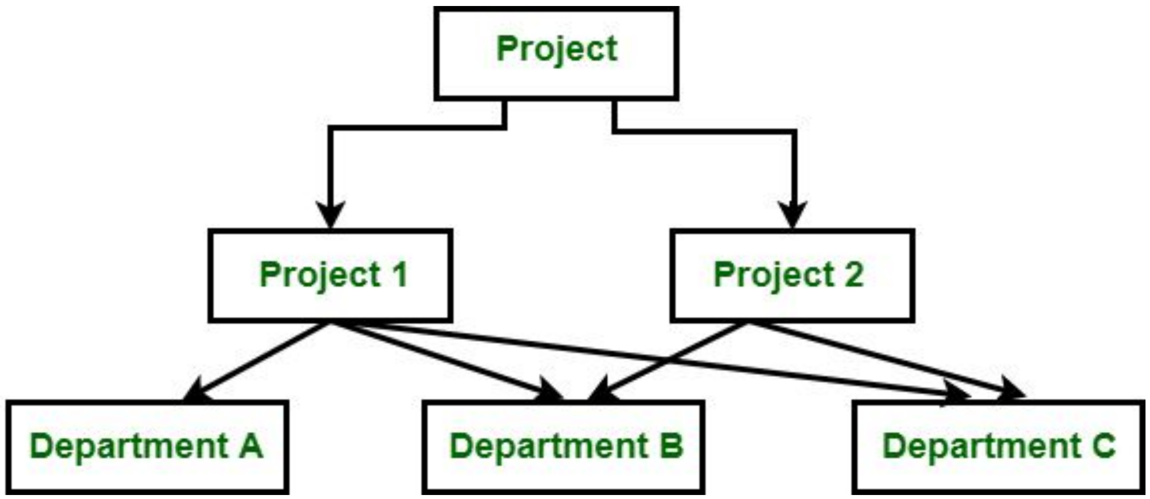


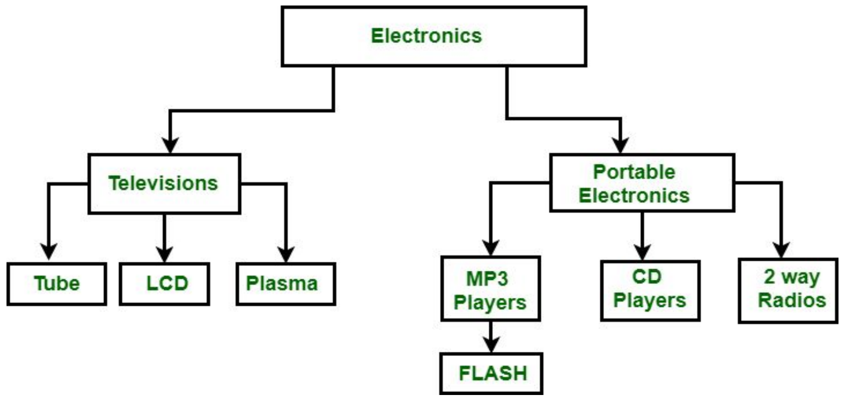
Hierarchical Model:

In Hierarchical Model, data are represented by records and relationships among data are represented by links. But unlike in Network model, data are organized in an ordered tree structure, which is called Hierarchical structure.

What is the Hierarchical Model in DBMS?

The Hierarchical Model was the first database management system model. This concept uses a hierarchical tree structure to organize the data. The hierarchy begins at the root, which contains root data, and then grows into a tree as child nodes are added to the parent node





Features of a Hierarchical Model

1. Parent-Child Relationship
A parent node exists for each child node. However, a parent node might have several child nodes. It is not permitted to have more than one parent.

2. One-to-many Relationship

The data is organized in a tree-like form, with the data types having a one-to-many relationship. There can only be one path from any node to its parent.

3. Deletion Problem
When a parent node is removed, the child node is removed as well.

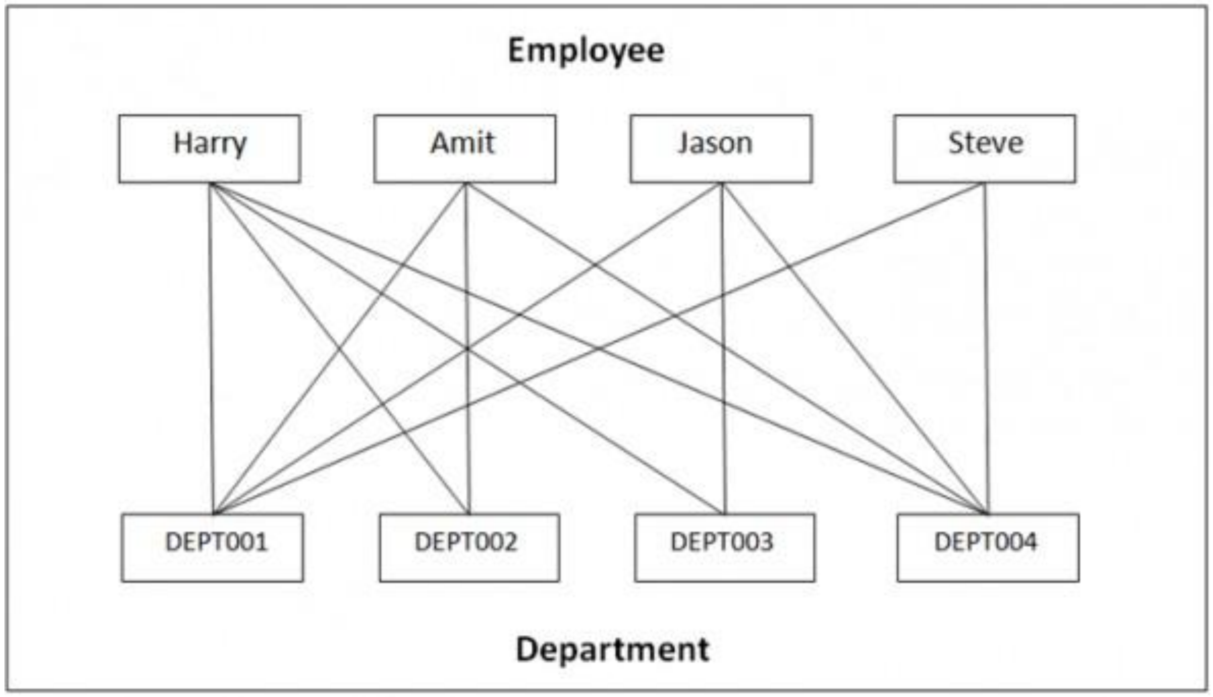
4. Pointers
Pointers are used to connect the parent and child nodes and to traverse and navigate between the stored data

Merits and De-Merits of Hierarchical Data Model

- **Merits**
  - The design of the hierarchical model is simple.
  - Provides Data Integrity since it is based on parent/ child relationship
  - Data sharing is feasible since the data is stored in a single database.
  - Even for large volumes of data, this model works perfectly.
- **De-Merits**
  - Implementation is complex.
  - This model has to deal with anomalies like Insert, Update and Delete.
  - Maintenance is difficult since changes done in the database may want you to do changes in the entire database structure.

Network Data Model

- The Hierarchical Model creates hierarchical tree with parent/ child relationship, whereas the Network Model has graph and links.
- The relationship can be defined in the form of links and it handles many-to-many relations. This itself states that a record can have more than one parent.

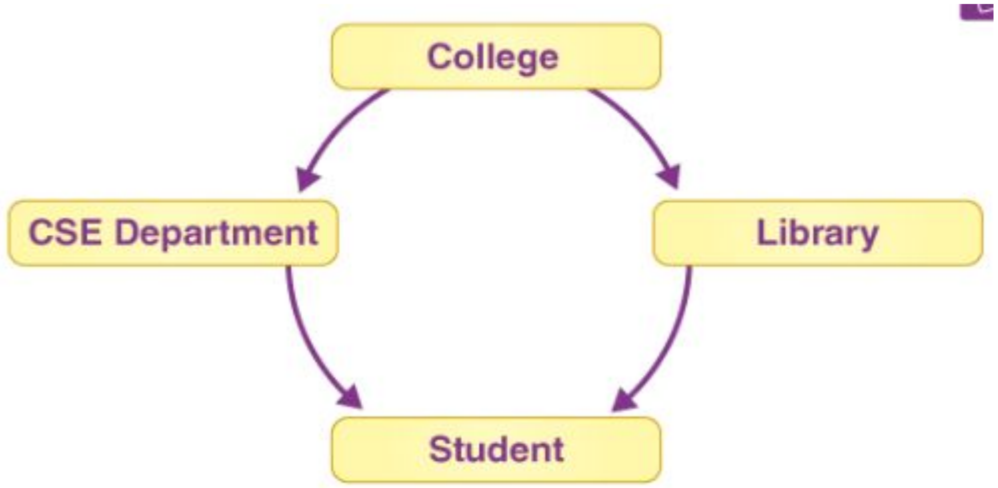


Network Model:

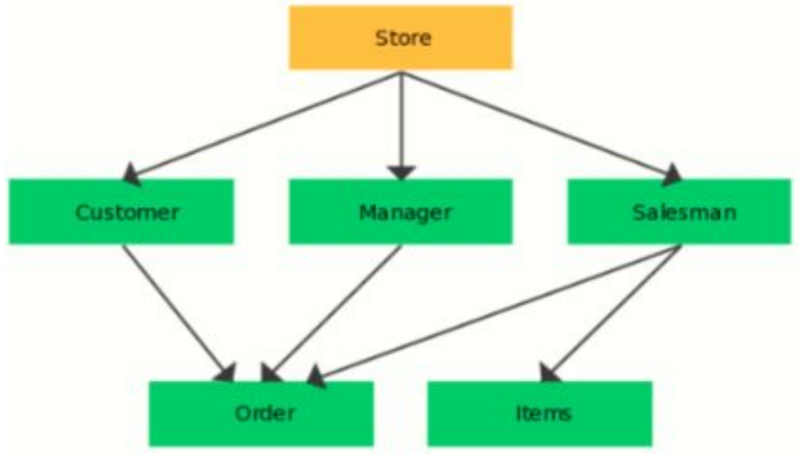
The network model uses two different structures. The data are represented by a collection of records and the relationships among data are represented by links.

What is the Network Model in DBMS?

The hierarchical model is extended in the network model. Prior to the relational model, it was the most popular model. To increase database performance and standards, the network model was devised to express complicated data relationships more effectively than hierarchical models.



Network Model



Merits and De-Merits of Network Model

- **Merits :**
  - Easy to design the Network Model
  - The model can handle one-one, one-to-many, many-to-many relationships.
  - It isolates the program from other details.
  - Based on standards and conventions.
- **De-Merits :**
  - Pointers bring complexity since the records are based on pointers and graphs.
  - Changes in the database isn't easy that makes it hard to achieve structural independence.

Relational Model:

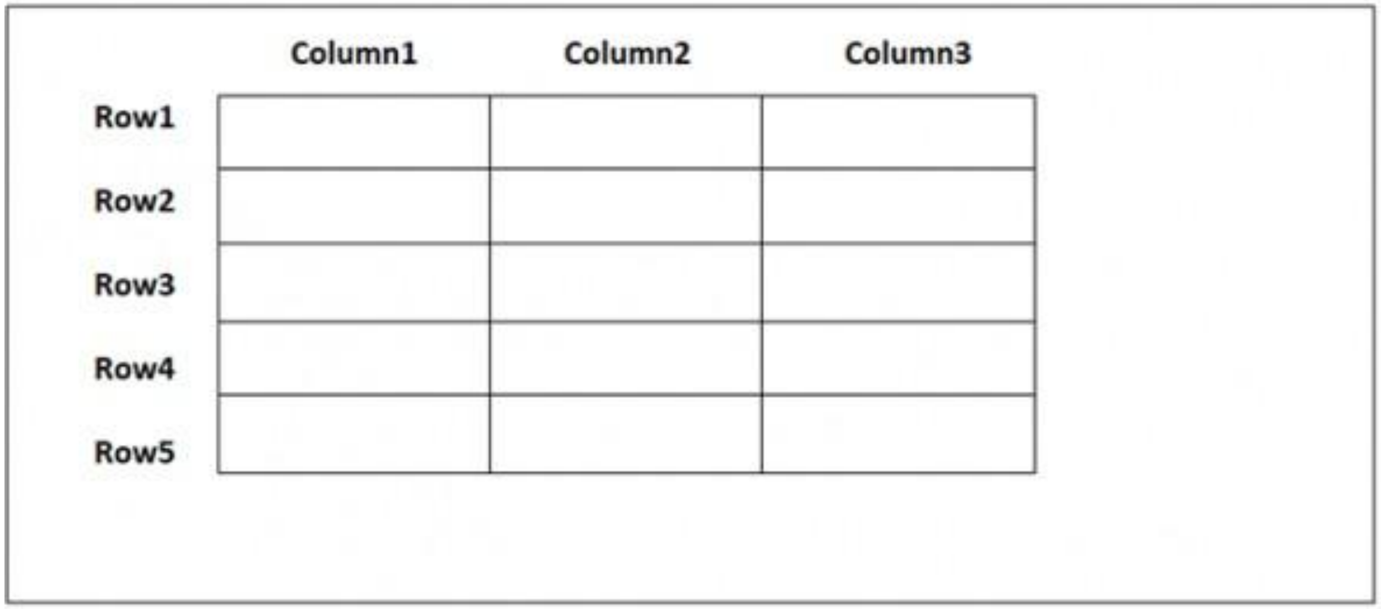
A data model in which both data and their relationships are represented by means of tables is called Relational Model.

The relation is the only data structure used in this model to represent both entities and their interrelationships. A relation is a two dimensional table with a unique name.

Each row of a table is called a tuple and each column of a table is called an attribute. The set of all possible values in an attribute is called the domain of the attribute.

Relational Data Model

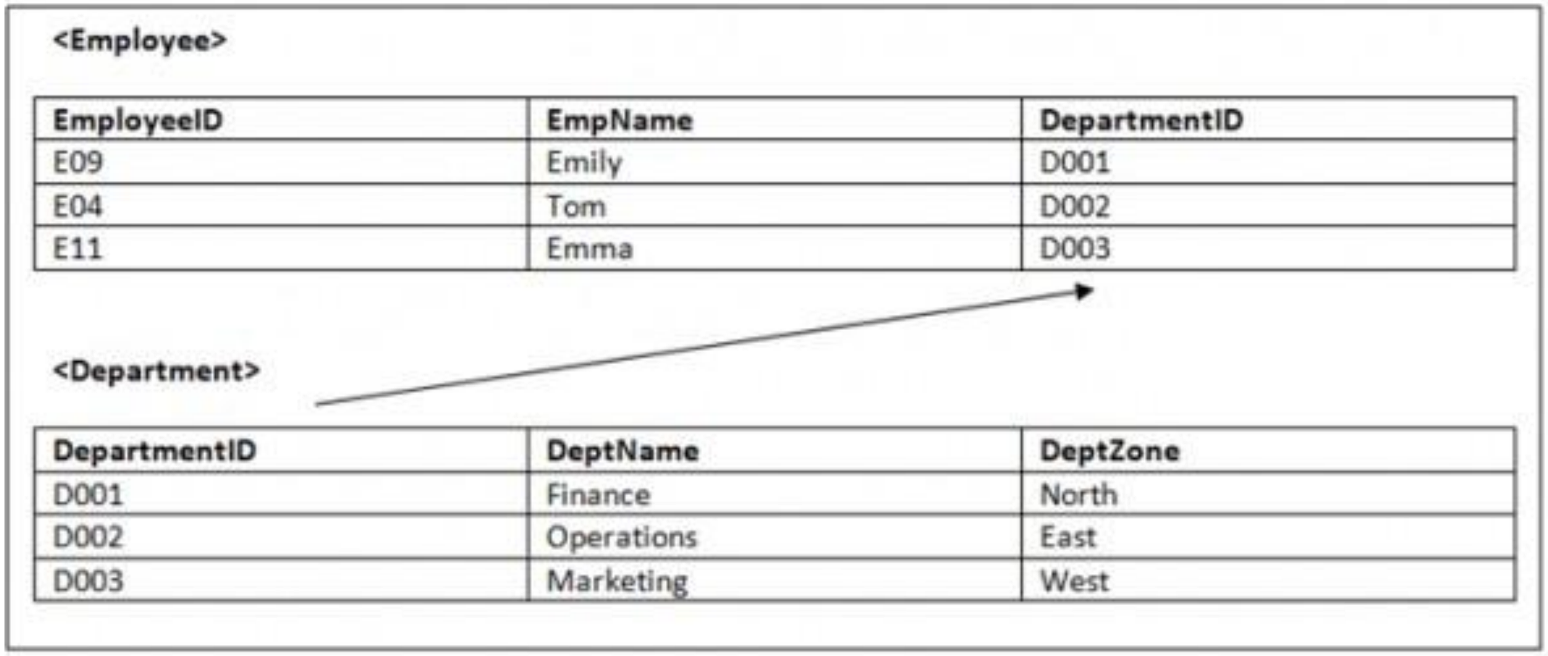
- A relational model groups data into one or more tables. These tables are related to each other using common records.
- The data is represented in the form of rows and columns i.e. tables:



	Column1	Column2	Column3
Row1			
Row2			
Row3			
Row4			
Row5			

Relational Data Model

- **Example :** Let us see an example of two relations <**Employee**> and <**Department**> linked to each other, with DepartmentID, which is **Foreign Key** of <**Employee**> table and Primary key of <Department> table.



<Employee>

EmployeeID	EmpName	DepartmentID
E09	Emily	D001
E04	Tom	D002
E11	Emma	D003

<Department>

DepartmentID	DeptName	DeptZone
D001	Finance	North
D002	Operations	East
D003	Marketing	West

Example of tabular data in the relational model:

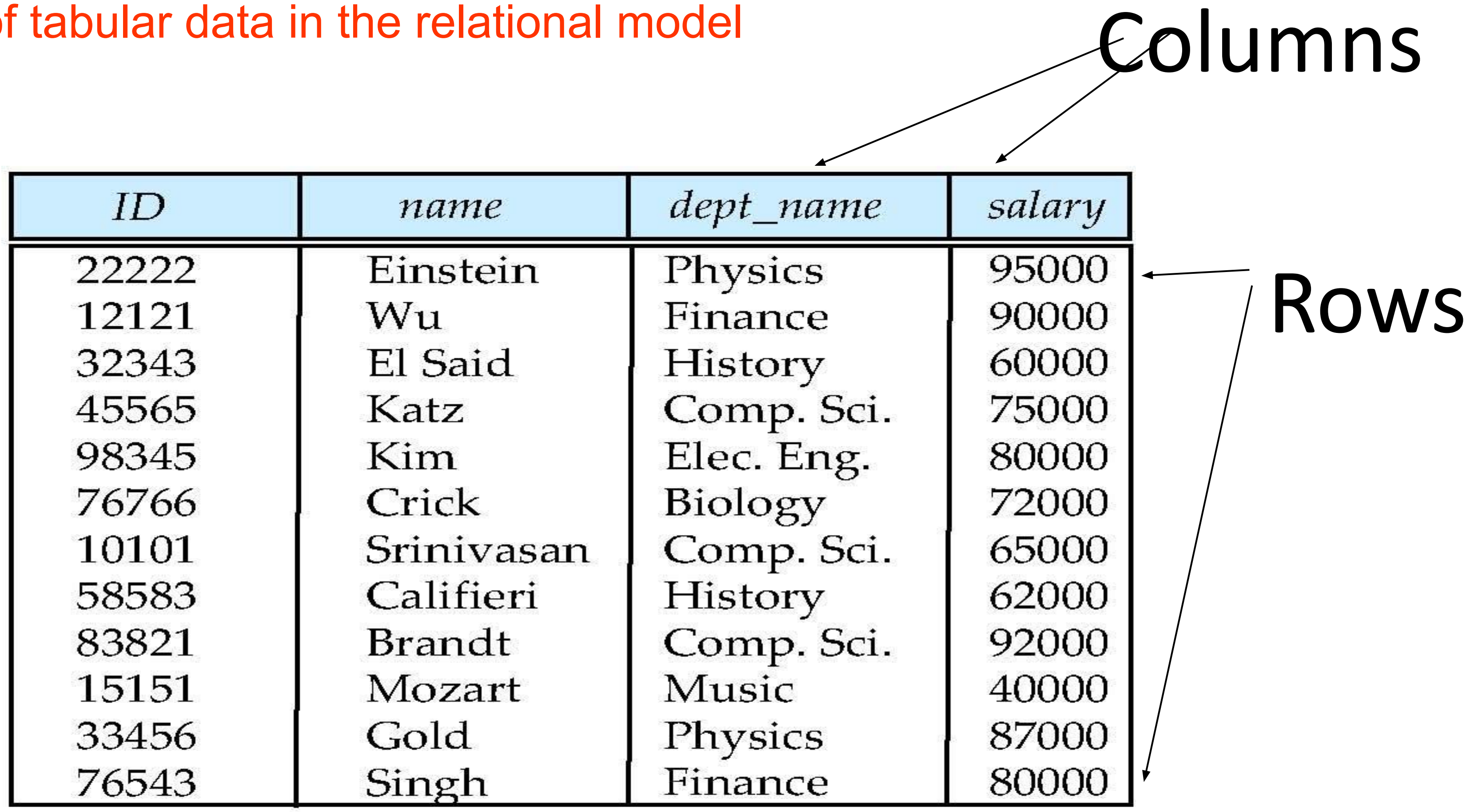
Relational Model

name	ssn	street	city	account-number
Johnson	192-83-7465	Alma	Palo Alto	A-101
Smith	019-28-3746	North	Rye	A-215
Johnson	192-83-7465	Alma	Palo Alto	A-201
Jones	321-12-3123	Main	Harrison	A-217
Smith	019-28-3746	North	Rye	A-201

account-number	balance
A-101	500
A-201	900
A-215	700
A-217	750

Relational Model

All the data is stored in various tables.
Example of tabular data in the relational model


Columns
Rows

ID	*name*	*dept_name*	*salary*
22222	Einstein	Physics	95000
12121	Wu	Finance	90000
32343	El Said	History	60000
45565	Katz	Comp. Sci.	75000
98345	Kim	Elec. Eng.	80000
76766	Crick	Biology	72000
10101	Srinivasan	Comp. Sci.	65000
58583	Califieri	History	62000
83821	Brandt	Comp. Sci.	92000
15151	Mozart	Music	40000
33456	Gold	Physics	87000
76543	Singh	Finance	80000

(a) The *instructor* table