TEACHING SCHEME

Subject:**Database Management System**								
Program: **B.Tech. CE/CS/IT**				Subject Code:			Semester: **III**	
Teaching Scheme				**Examination Evaluation Scheme**				
Lecture	**Tutorial**	**Practical**	**Credits**	**University Theory Examination**	**University Practical Examination**	**Continuous Internal Evaluation (CIE)- Theory**	**Continuous Internal Evaluation (CIE)- Practical**	**Total**
3	0	2	4	40	60	40	60	200

TEXT BOOK:

- **1. Database System Concepts, Abraham Silberschatz, Henry F. Korth & S. Sudarshan, McGraw Hill.**
- 2. SQL- PL/SQL by Ivan bayross

REFERENCE BOOK:

- **1. An introduction to Database Systems, C J Date, Addition-Wesley.**
- **2. Understanding SQL by Martin Gruber, BPB**

UNIT - 1

- Overview of database management
- Introduction: data, database, database management, database management system, application of database, why database? , Data independence
- Architecture: The three levels of architecture-Levels, mapping, Database users and DBA Brief overview of different types of model
- Entity-Relationship model:
- Introduction, An overview of the E/R model, E/R diagrams, Database design with the E/R model, extended E-R features – generalization, specialization, aggregation, reduction to E-R database schema


ORACLE
foundation
firebird
PostgreSQL
IBM
DB2
Microsoft
SQL Server 2008
SQLite
MySQL
123
TERADATA
CSV

Brief History

Manual System - 1950s

- Data was stored as paper records
- Huge man power involved
- Unnecesary time was wasted like when searching for a particular record
- This was inefficient

1950s and early 1960s

- Data processing using magnetic tapes for storage
- Tapes provide only sequential access
- Punched cards for input

Late 1960s and 1970s

- Hard disks allow direct access to data
- Data stored in files
- Known as File Processing System

86x188mm

Relógio de Ponto

Matrícula	Nome	Mês/Ano
Departamento	Cargo	
Horário de Trabalho		
Observações		

1° Quinzena

Dia	ENT	SAI	ENT	SAI	ENT	SAI	Total
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							
14							
15							

Cartão de Ponto

Relógio de Ponto

Matrícula	Nome	Mês/Ano
Faltas		
Extras		
Horas Noturnas		
DSR		

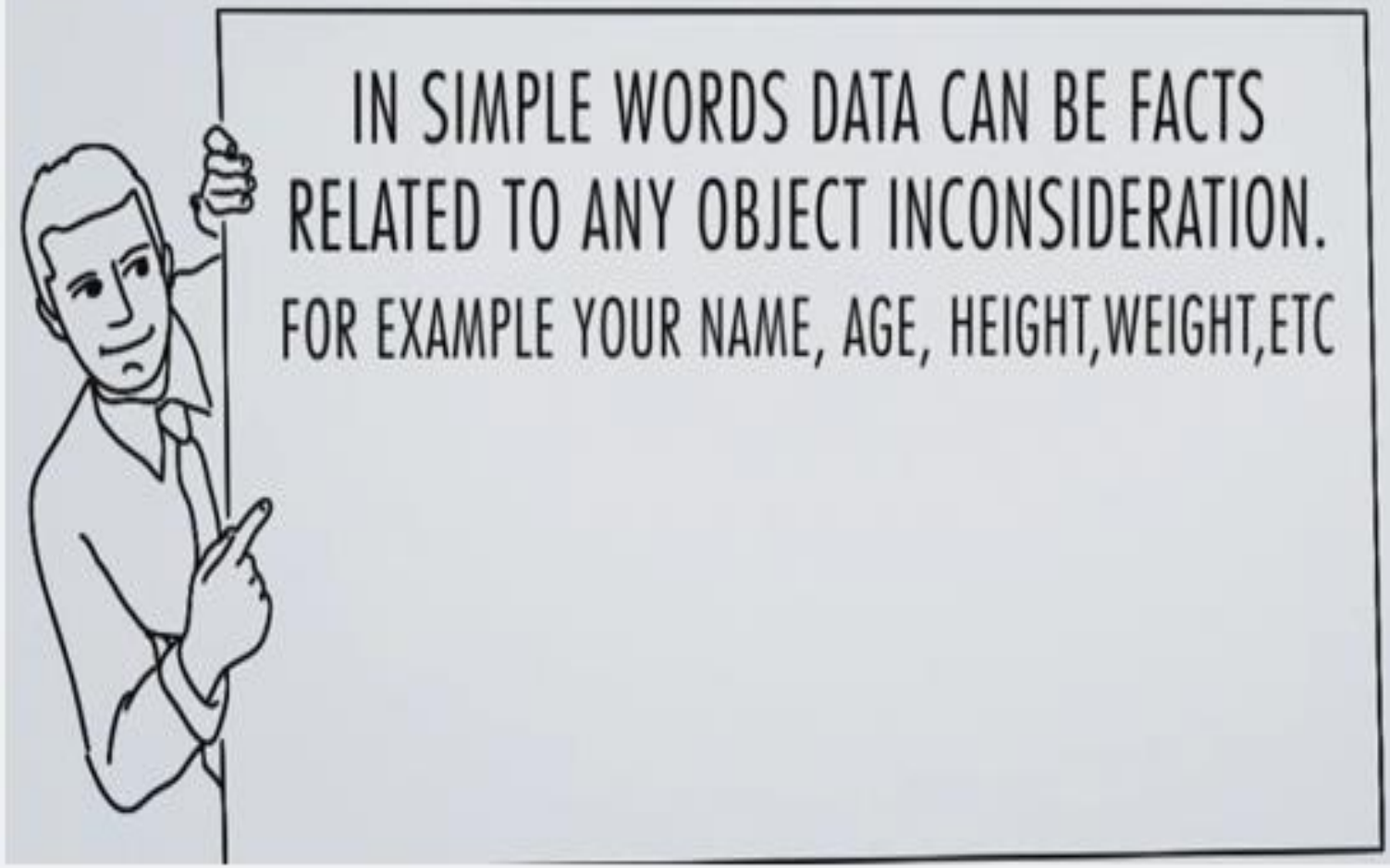
2° Quinzena

Dia	ENT	SAI	ENT	SAI	ENT	SAI	Total
16							
17							
18							
19							
20							
21							
22							
23							
24							
25							
26							
27							
28							
29							
30							
31							

Confirmo os horários registrados neste cartão.

Brief History

The first of the database system the mankind saw was developed as internal program by IBM and was based on binary trees, where the shape was like a tree and relations were only limited between parent and child records. The benefits were numerous; less redundant data, data independence, security and integrity, which all lead to efficient searches. Nonetheless; there were some disadvantages such as; complex implementation, was hard to manage because of the absence of standards, which made it harder to handle many relationships.


IN SIMPLE WORDS DATA CAN BE FACTS
RELATED TO ANY OBJECT INCONSIDERATION.
FOR EXAMPLE YOUR NAME, AGE, HEIGHT,WEIGHT,ETC

What is Data ?

- A collection of raw facts and figures.
- Raw material that can be processed by any computing machine.
- A collection of facts from which conclusions may be drawn.
- Data can be represented in the form of numbers and words which can be stored in computer's language.
  i.e. Paan Singh, Anshul 007

What is Information?

- Systematic and meaningful form of data.
- Knowledge acquired through study or experience.
- Information helps human beings in their decision making.

WE ALREADY KNOW WHAT DATA IS .
BUT THIS DATA COULD BE RANDOM.
A DATABASE IS A SYSTEMATIC
COLLECTION OF DATA. SINCE THE
DATA IN A DATABASE IS ORGANIZED
IT MAKES DATA MANAGEMENT EASY.

DATABASE MANAGEMENT SYSTEM (DBMS) IS A COLLECTION OF PROGRAMS WHICH ENABLES ITS USERS TO ACCESS DATABASE, MANIPULATE DATA, & HELP IN REPRESENTATION OF DATA .

IT ALSO HELPS CONTROL ACCESS TO THE DATABASE BY VARIOUS USERS.

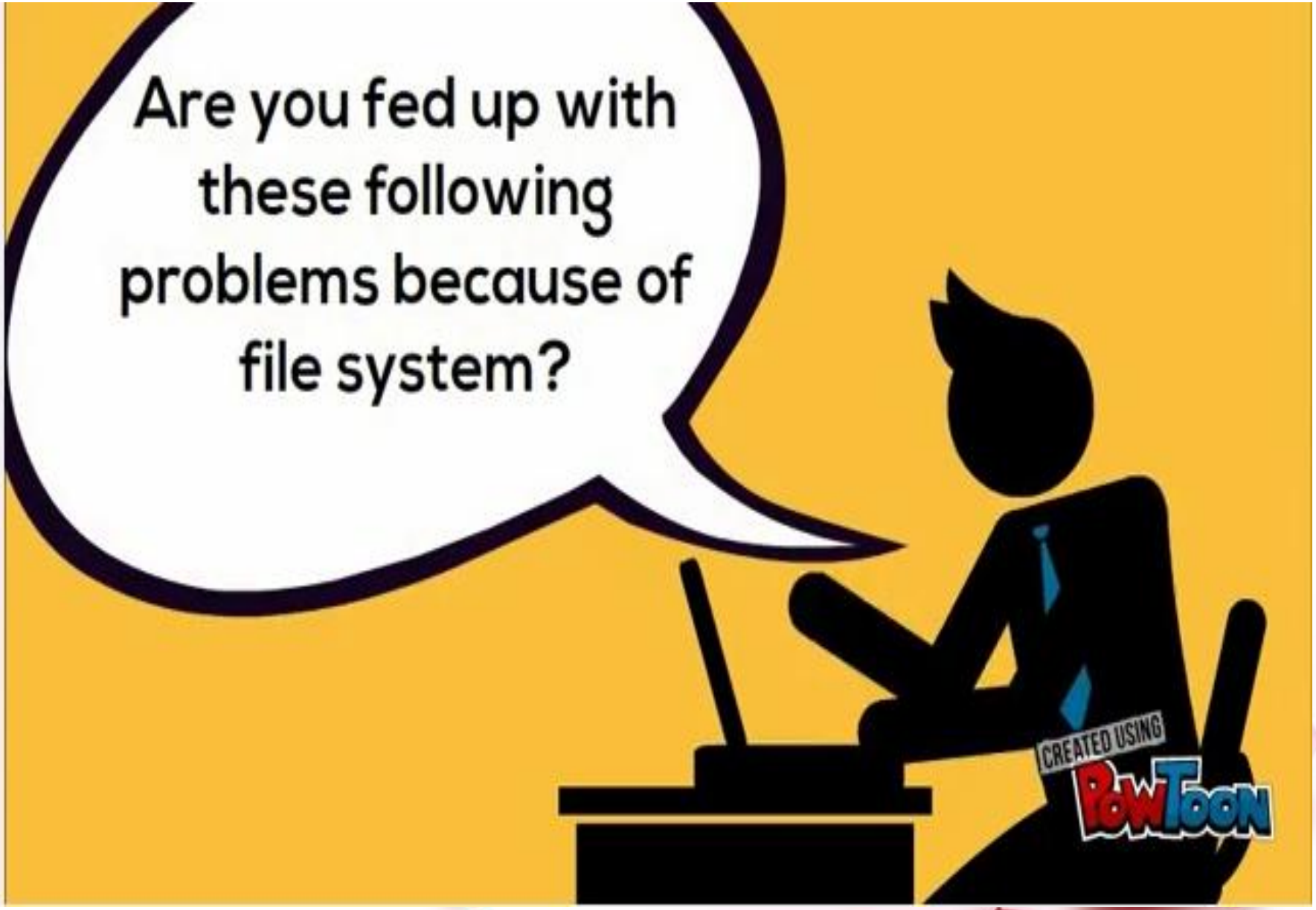
AN ONLINE TELEPHONE DIRECTORY WOULD DEFINITELY USE DATABASE MANAGEMENT SYSTEM TO STORE DATA PERTAINING TO PEOPLE, PHONE NUMBERS,& OTHER CONTACT DETAILS.

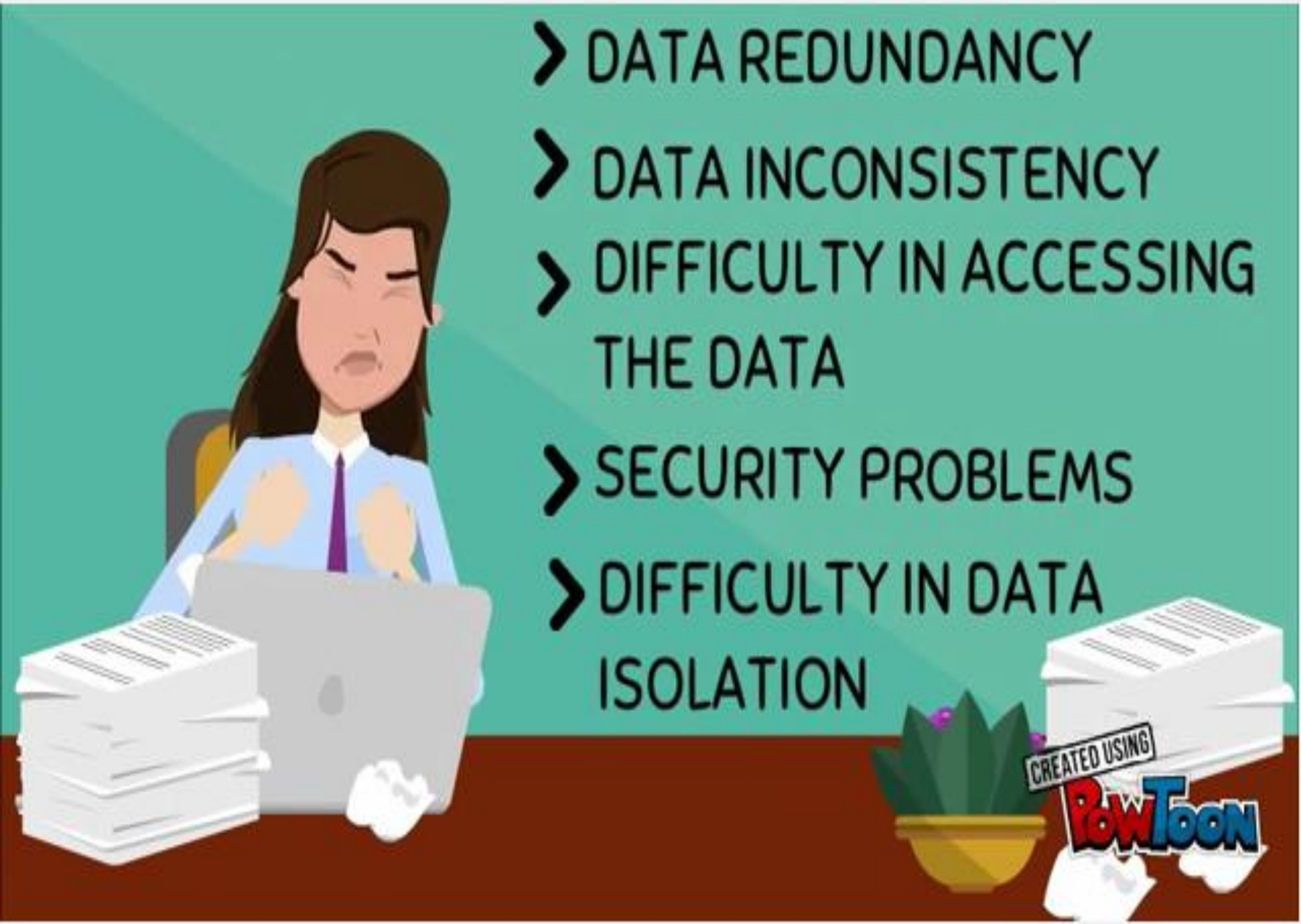
YOUR ELECTRICITY SERVICE PROVIDER IS OBVIOUSLY USING A DBMS TO MANAGE BILLING , CLIENT RELATED ISSUES, TO HANDLE FAULT DATA, ETC.

LET'S ALSO CONSIDER THE FACEBOOK. IT NEEDS TO STORE, MANIPULATE AND PRESENT DATA RELATED TO MEMBERS, THEIR FRIENDS, MEMBER ACTIVITIES, MESSAGES, ADVERTISEMENTS AND LOT MORE.

DATABASE MANAGEMENT SYSTEMS ARE NOT A NEW CONCEPT AND AS SUCH HAD BEEN FIRST IMPLEMENTED IN 1960S. CHARLES BACHMEN'S INTEGRATED DATA STORE(IDS) IS SAID TO BE THE FIRST DBMS IN HISTORY.

WITH TIME DATABASE TECHNOLOGIES EVOLVED A LOT WHILE USAGE AND EXPECTED FUNCTIONALITIES OF DATABASES HAVE BEEN INCREASED IMMENSELY.


Are you fed up with these following problems because of file system?
CREATED USING
PowToon


DATA REDUNDANCY
DATA INCONSISTENCY
DIFFICULTY IN ACCESSING THE DATA
SECURITY PROBLEMS
DIFFICULTY IN DATA ISOLATION
CREATED USING
POWTOON

DBMS vs File Handling

- Controlling Redundancy
- Restricting Unauthorized Access
- Efficient Query Processing
- Providing Backup and Recovery
- Multiple user interfaces
- Integrity Constraints
- Relationship among Data
- Development Time
- Flexibility


WE HAVE JUST
the solution
for you


USE DATABASE
INSTEAD OF FILE
SYSTEM !
USE DATABASE
INSTEAD OF FILE
SYSTEM !
0:28
CREATED USING


HOW DBMS SOLVES
THESE PROBLEMS...
>MANAGES RECORD OF INFORMATIC
IN A DESIRED MANNER
>PROVIDES EASY ACCESS TO USER
>MINIMUM REDUNDANCY
>PROVIDES EASY WAY TO UP
DATA
CREATED USING
PowToo

File System vs. DBMS

- A company has 500 GB of data on employees, departments, products, sales, & so on..
- Data is accessed concurrently by several employees
- Queries about the data must be answered quickly
- Changes made to the data by different users must be applied consistently
- Access to certain parts of the data be restricted

File System vs. DBMS

- Data stored in operating system files
- Many drawbacks!!!
  - 500 GB of main memory not available to hold all data. Data must be stored on secondary storage devices
  - Even if 500GB of main memory is available, with 32-bit addressing, we cannot refer directly to more than 4GB of data
  - Data redundancy and inconsistency
    - Multiple file formats, duplication of information in different files
  - Special program to answer each query user may ask

File System vs. DBMS

- Many drawbacks!!!
  - Integrity problems
    - Integrity constraints (e.g. account balance > 0) become "buried" in program code rather than being stated explicitly
    - Hard to add new constraints or change existing ones
  - We must protect the data from inconsistent changes made by different users. If application programs need to address concurrency, their complexity increases manifolds
  - Consistent state of data must be restored if the system crashes while changes are being made
  - OS provide only a password mechanism for security. Not flexible enough if users have permission to access subsets of data

Advantages of a DBMS

- **Data Administration**
  - When several users share data , centralizing the administration offers significant improvement
- **Concurrent Access & Crash Recovery**
  - DBMS schedules concurrent access to the data in such a manner that users think of the data as being accessed by only one user at a time
  - DBMS protects users from the ill-effects of system failures
- **Reduced Application Development Time**
  - Many important tasks are handled by the DBMS

Advantages of a DBMS

- **Program-Data Independence**
  - Insulation between programs and data: Allows changing data storage structures and operations without having to change the DBMS access programs.
- **Efficient Data Access**
  - DBMS uses a variety of techniques to store & retrieve data efficiently
- **Data Integrity & Security**
  - Before inserting salary of an employee, the DBMS can check that the dept. budget is not exceeded
  - Enforces access controls that govern what data is visible to different classes of users

Disadvantages of DBMS

- **Cost of Hardware and Software:** Processor with high speed of data processing and memory of large size is required.
- **Cost of Data Conversion:** Very difficult and costly method to convert data of data file into database.
- **Cost of Staff Training:** A lot of amount for the training of staff to run the DBMS.
- **Appointing Technical Staff:** Trained technical persons such as database administrator, application programmers, data entry operators etc. are required to handle the DBMS.
- **Database Damage:** All data is integrated into a single database. If database is damaged due to electric failure or database is corrupted on the storage media, then your valuable data may be lost forever.

People Who Work with Databases

- Database Implementers/ Designers
- DBA
- Application Programmers
- End Users

End Users

- **Casual users**

  These are people who use the database occasionally.

- **Naive users**

  These are users who constantly querying and updating the database.

  Eg. Reservation Clerks of Airline, Railway, Hotel, etc.

  Clerks at receiving station of Courier service, Insurance agencies, etc.

- **Sophisticated Users**

  People who use for their complex requirements.

  Eg. Engineers, Scientists, Business analysts…

- **Standalone Users**

  Who maintain database for personal use.

I. Casual End Users

✓Access data base occasionally when needed
✓They need different information each time
✓They use sophisticated query language to specify their request
✓example: High level Managers who access the data weekly or biweekly.

II. Naive or parametric users

- ✓They make up large section of the database
- ✓ They communicate with the database on regular period
- ✓ communicate with the system by invoking one of the applications programs that have been written previously
- ✓Their job is to constantly querying and updating database using standard queries
- ✓ This is called canned transaction
- ✓Ex: Bank teller, reservation clerks etc

III. Sophisticated users

✓They include business analyst , scientist, engineers, other thoroughly familiar with the system capability

✓ They interact with the system without writing programs, instead they form their requests in a database query language

✓ They submit each query to a query processor

IV. Specialized users

- Also Sophisticated users who write specialized database application that do not fit into the traditional data processing framework
- Write specialized applications like CAD (Computer Aided design),Multimedia database programs

B) S/w Engineers

I. Application programmer

→Writing database programs in some programming languages (Such as COBOL,C++ or some forth generation languages)

→ Application programs access database by issuing the appropriate request to the DBMS.

II. Systems analysts

- Architects, as well as the project leaders, of an information system.
- It is their job
  - ❑ Develop solutions to users problems
  - ❑ Determine the technical and operational feasibility of their solutions
  - ❑ Estimate the costs to develop and implement them.

Application programmer + System Analyst = Software Engineers

C) Database Administrator

- The DBA can be a single person or a team comprising a group of persons

The functions of the DBA include the following:

- ✓ Definition of the Conceptual Schema
- ✓ Definition of the Internal Schema
- ✓ Liaising with users
- ✓ Granting of authorization for data access
- ✓ Defining Integrity constraints

1. Definition of the Conceptual Schema

- To decide exactly what information is to be held in the database.
- Identifies the entities and the information to be recorded about those entities. This process is usually referred to as logical database design.
- Once the DBA has decided the content of the database at an abstract level, he creates the corresponding conceptual schema

2. Definition of the Internal Schema

- Decide how the data is to be represented in the database. This process is usually referred to as physical database design.
- Having done the physical design, the DBA must then create the corresponding storage structure definition.
- In addition, the DBA must also define the associated conceptual/internal mapping

3. Liaising with users

- The DBA liaises with users to ensure that the data they need is available and to write the necessary external schema.
- In addition, the DBA must also define the associated external/conceptual mapping

4. Granting of authorization for data access

- The granting of different types of authorizations (read, write, etc.) allows the DBA to regulate which parts of the database various users can access

5. Defining Integrity constraints

- The data values stored in the database must satisfy certain consistency constraints

External Level

Works at the highest level of abstraction.
Deals with updates and queries

End User

Conceptual Level

Writes application programs

Application Programmers

Defines the Conceptual, Internal and External schema, controls access privileges to users and ensures consistency of the database

Internal Level

Data Base Administrator (DBA)

Database Languages

- **DDL** – Data Definition Language
- **SDL** – Storage Definition Language
- **VDL** – View Definition Language
- **DML** – Data Manipulation Language

  (For data manipulations like insertion, deletion, update, retrieval, etc.)

Data Definition Language (DDL):

- DDL is used for specifying the database schema. Let's take SQL for instance to categorize the statements that comes under DDL.
- To create the database instance – CREATE
- To alter the structure of database – **ALTER**
- To drop database instances – DROP
- To delete tables in a database instance – **TRUNCATE**
- To rename database instances – **RENAME**
- All these commands specify or update the database schema that's why they come under Data Definition language.

Data Manipulation Language (DML):

- DML is used for accessing and manipulating data in a database.
- To read records from table(s) – SELECT
- To insert record(s) into the table(s) – **INSERT**
- Update the data in table(s) – UPDATE
- Delete all the records from the table – DELETE

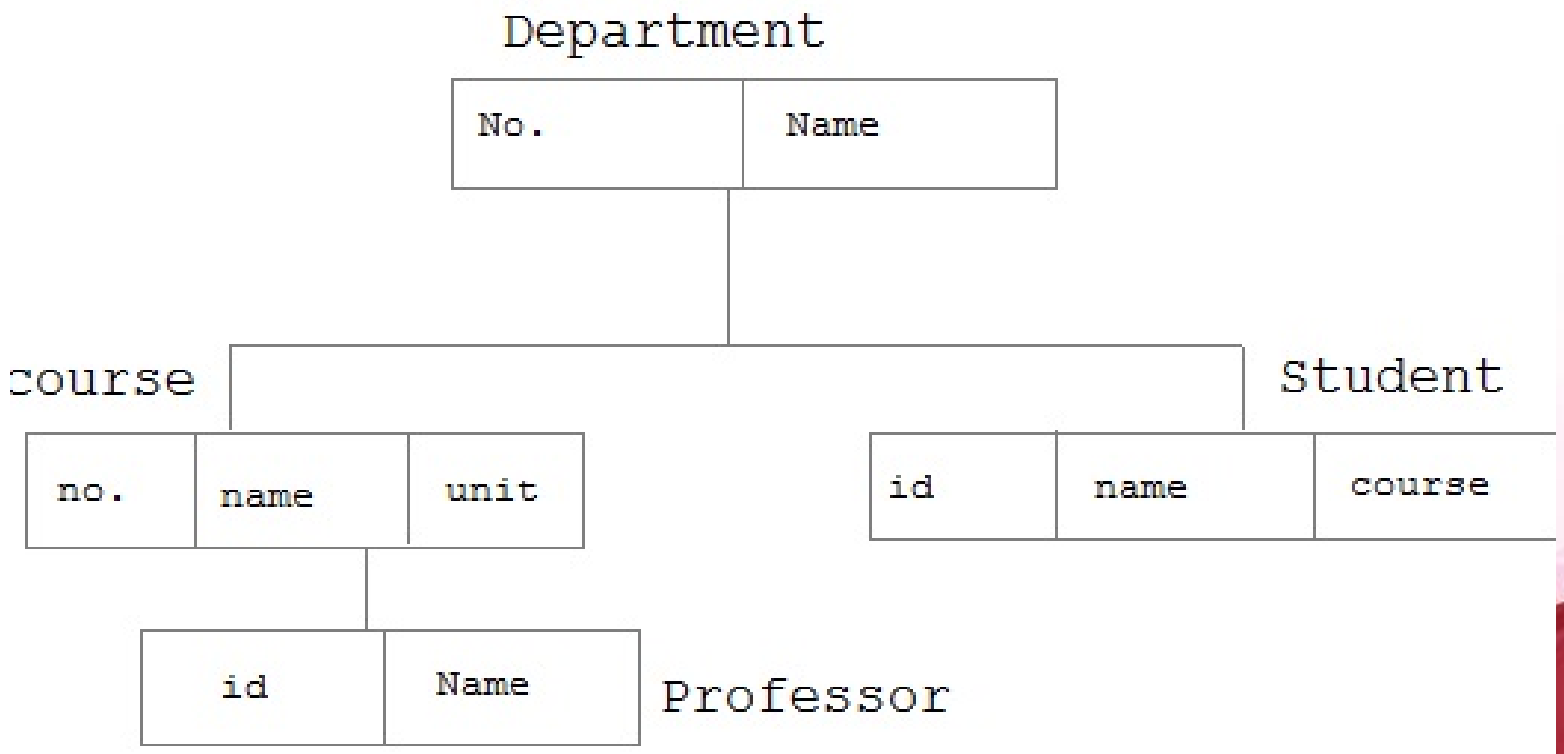
Data Control language (DCL):

- DCL is used for granting and revoking user access on a database –
- To grant access to user – GRANT
- To revoke access from user – REVOKE
- In practical data definition language, data manipulation language and data control languages are not separate language; rather they are the parts of a single database language such as SQL.

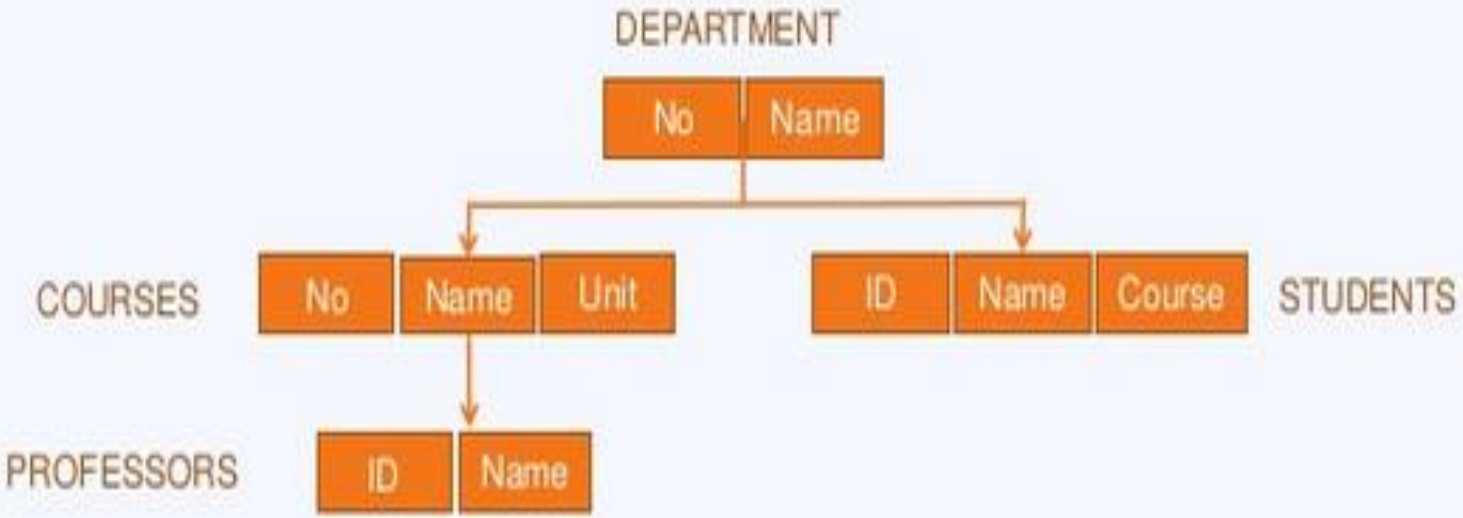
- **Database Model**
- A Database model defines the logical design of data. The model describes the relationships between different parts of the data. Historically, in database design, three models are commonly used. They are,
- Hierarchical Model
- Network Model
- Relational Model

Hierarchical Model

In this model each entity has only one parent but can have several children . At the top of hierarchy there is only one entity which is called **Root**.



Hierarchical Model



- Hierarchical Model The hierarchical data model organizes data in a tree structure. There is a hierarchy of parent and child data segments. This structure implies that a record can have repeating information, generally in the child data segments. Data in a series of records, which have a set of field values attached to it. It collects all the instances of a specific record together as a record type. These record types are the equivalent of tables in the relational model, and with the individual records being the equivalent of rows. To create links between these record types, the hierarchical model uses Parent Child Relationships. These are a1:N mapping between record types. This is done by using trees, like set theory used in the relational model, "borrowed" from maths.

- For example, an organization might store information about an employee, such as name, employee number, department, salary. The organization might also store information about an employee's children, such as name and date of birth. The employee and children data forms a hierarchy, where the employee data represents the parent segment and the children data represents the child segment. If an employee has three children, then there would be three child segments associated with one employee segment. In a hierarchical database the parent-child relationship is one to many. This restricts a child segment to having only one parent segment. Hierarchical DBMSs were popular from the late1960s, with the introduction of IBM's Information Management System (IMS) DBMS, through the1970s.

a. Hierarchical Database Model

- Developed by IBM, is the Oldest database model.
- Represented using a tree-diagram. (Parent-child relationship)
- Each box is called a 'Node'
- The nodes represent a record type.
- A line connecting nodes represents the link.

Cont…

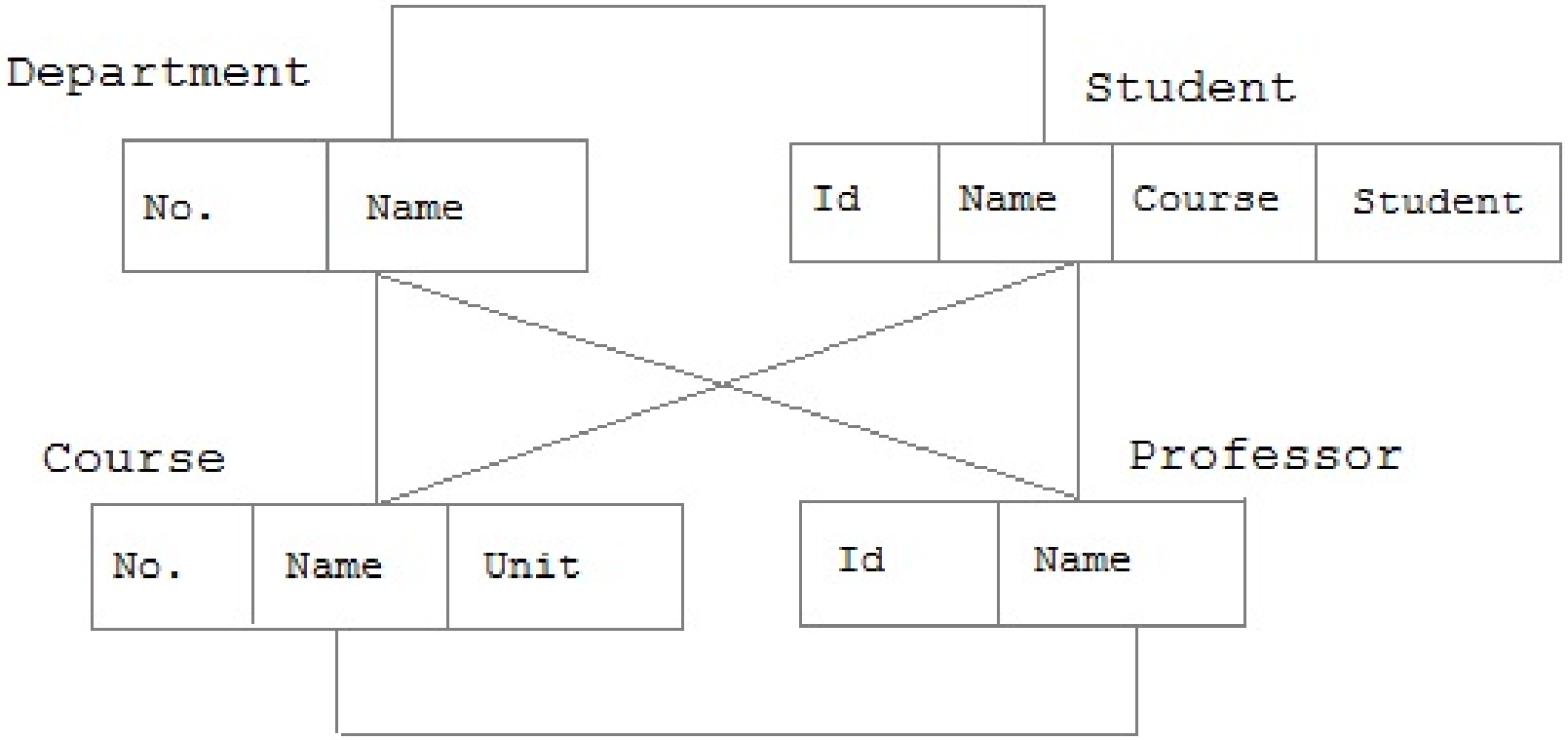
- Parent-child type is suited for One-to-many relationship between two entities.
- But difficult to implement many-to-many relationship.

e.g.:

IMS system from IBM.

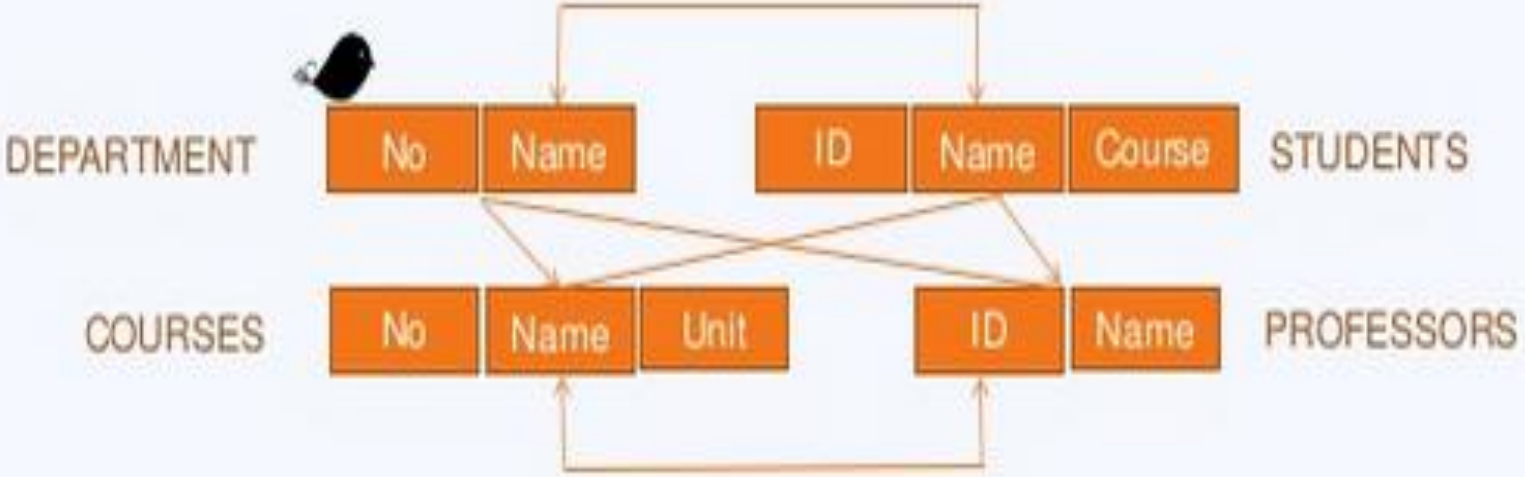
Network Model

the network model, entities are organised in a graph,in which some entities can be accessed through several path



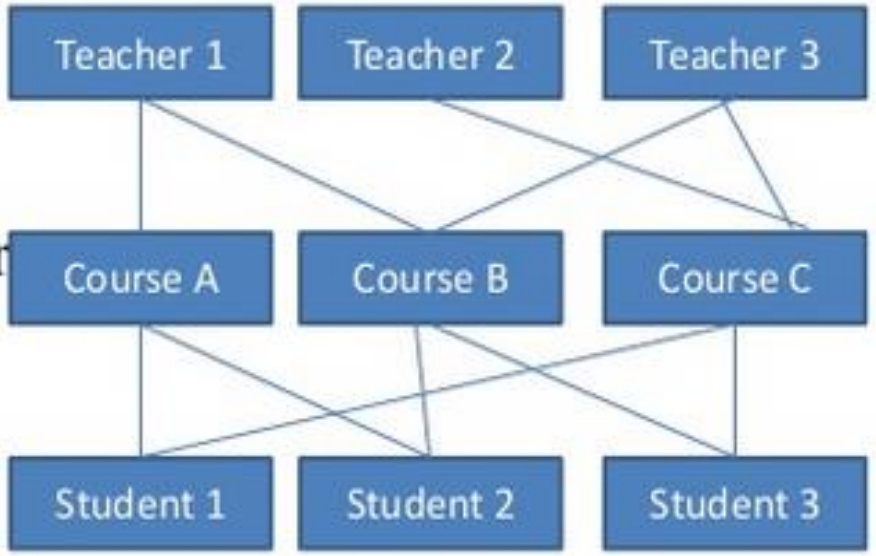
Network Model

In the network model, the entities are organized in a graph, in which some entities can be accessed through several paths



c. Network Database Model

- Represented using a Data-Structure Diagram.
- Boxes represents the records & lines the links.
- Based on 'owner-member relationship.'
- Members of an owner may be many but for many member owner is one.
- Can represent one-to-one and many-to-many as well.

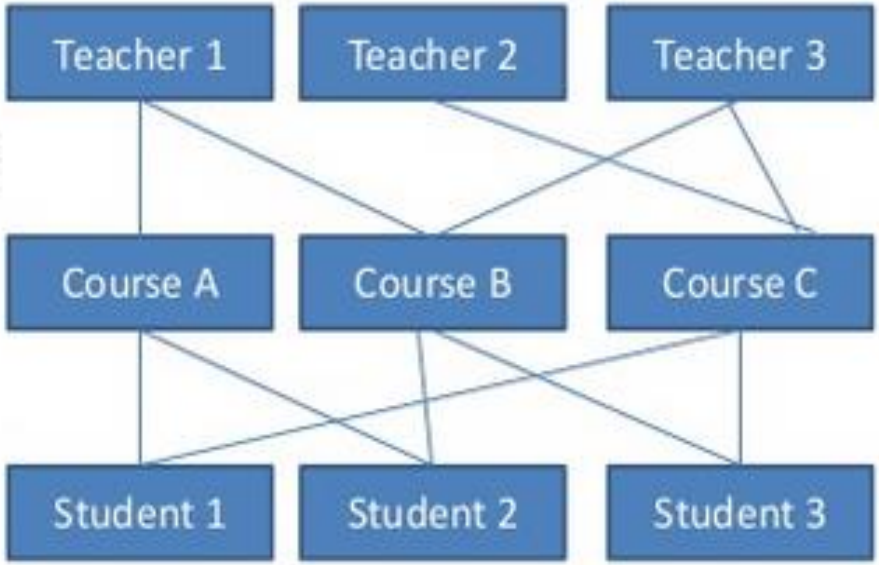


Cont…

- One-to-many relationship is converted into a set of one-to-one.
- Also, many-to-many is converted into 2 or more one-to-many relationship.

  e.g.:

  IDMS, IMAGE.



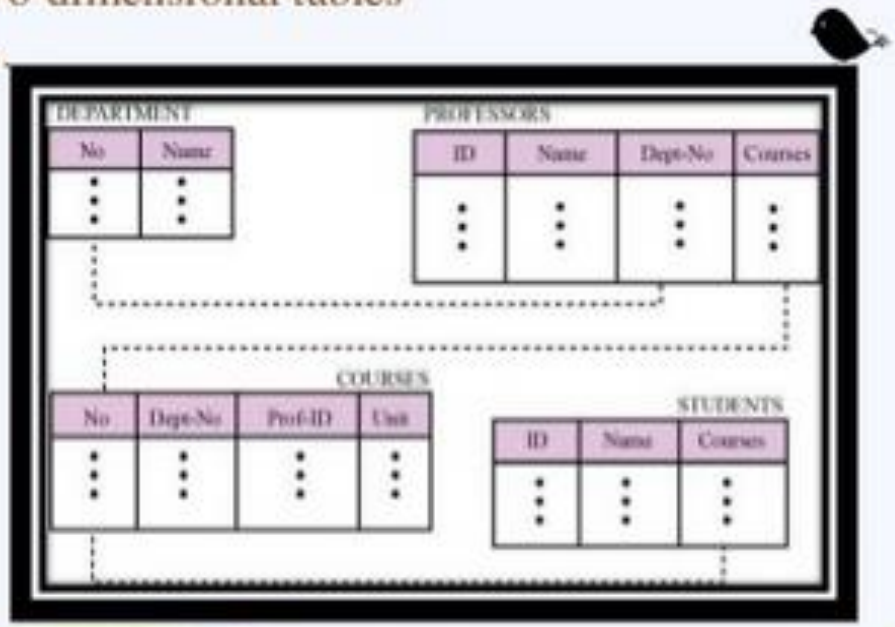
- Network Model The popularity of the network data model coincided with the popularity of the hierarchical data model. Some data were more naturally modeled with more than one parent per child. So, the network model permitted the modeling of many-to-many relationships in data. In1971, the Conference on Data Systems Languages (CODASYL) formally defined the network model. The basic data modeling construct in the network model is the set construct. A set consists of an owner record type, a set name, and a member record type. A member record type can have that role in more than one set, hence the multiparent concept is supported. An owner record type can also be a member or owner in another set. The data model is a simple network, and link and intersection record types (called junction records by IDMS) may exist, as well as sets between them . Thus, the complete network of relationships is represented by several pairwise sets; in each set some (one) record type is owner (at the tail of the network arrow) and one or more record types are members (at the head of the relationship arrow). Usually, a set defines a1:M relationship, although1:1 is permitted. The CODASYL network model is based on mathematical set theory.

Relational Model

- In this model, data is organised in two-dimesional tables called **relations**. The tables or relation are related to each other.

Relational model

- In the relational model, data is organized in two-dimensional tables called relations



Relations


Attributes
No
Course-Name
Unit
CIS15
Intro to C
5
CIS17
Intro to Java
5
CIS19
UNIX
4
CIS51
Networking
5
Tuples


A relation appears as a two-dimensional table.

Relations

- Attributes.

Each column in a relation is called an attribute.

- Tuples

Each row in a relation is called a tuple. A tuple defines a collection of attribute values.

b. Relational Database Model

- Simplest and the most common model.
- Developed in 1970 by E.F. Codd, it became commercial in the 80s.
- Data elements are stored in different tables made up of rows and columns.

Roll No	Name	Surname	Section
1001	Rajkumar	Tomar	D
1002	Rajkumar	Singh	D

Cont…

- Terminologies:
  -Data Values: alphanumeric raw data (Rajkumar)
  -Columns: fields (item or object that holds the data)
  -Rows: record (a group of data for related field)
  -Table: collection (all records & fields)
  -Key: identifier (uniquely identifies a row in the table. It can be value of a single or multiple column.
  e.g.:
  DB2, ORACLE, SQL Server.

Roll No	Name	Surname	Section
1001	Rajkumar	Tomar	D
1002	Rajkumar	Singh	D

History of DBMS

- **1960** – First DBMS designed by Charles Bachman at GE. Integrated Data source(IBS).
- **1970** – Codd introduced IMS. IBMs Information Management System (IMS)
- **1980** – Relational model became popular and accepted as the main database paradigm. SQL, ANSI SQL, etc.
- **1980 to 1990** – New data models, powerful query languages, etc. Popular vendors are Oracle, SQL Server, IBMs DB2, Informix, etc.

DBMS Functionalities

- **Define a database :** In terms of data types, structures and constraints
- **Construct or Load the Database on a secondary storage medium**
- **Manipulating the database :** Querying, generating reports, insertions, deletions and modifications to its content
- **Concurrent Processing and Sharing by a set of users and programs** – Yet, keeping all data valid and consistent
- **Crash Recovery**
- **Data Security and Integrity**
- **Data Dictionary**
- **Performance**

People Who Work with Databases

- Database Implementers/ Designers
- DBA
- Application Programmers
- End Users

Applications of DBMS

- Various types of data: Images, Text, complex queries, Data Mining, etc.
- Enterprise Resource Planning (ERP)
- Management Resource Planning (MRP)
- Database in Web technologies
- Banking: all transactions
- Airlines: reservations, schedules
- Universities: registration, grades

Current Database trends:

- Multimedia databases
- Interactive video
- Streaming data
- Digital Libraries
- **Databases touch all aspects of our lives**

- DBMS Architecture

Two tier architecture one person handle client..

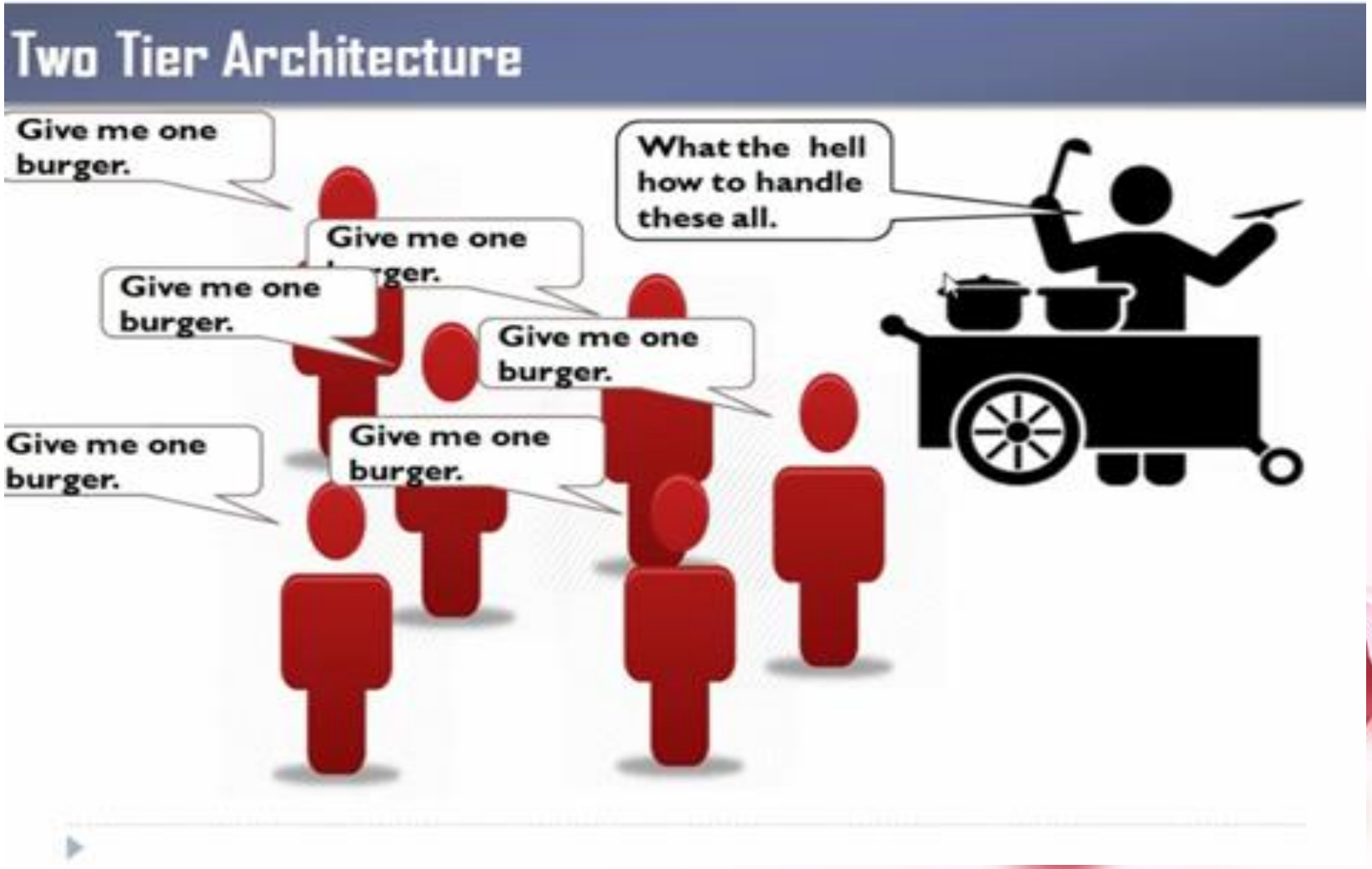
Two Tier Architecture

Guy X has a hot dog shop. He is only person to handle customers. He has to spent less to start this business as compared to opening a restrourant .

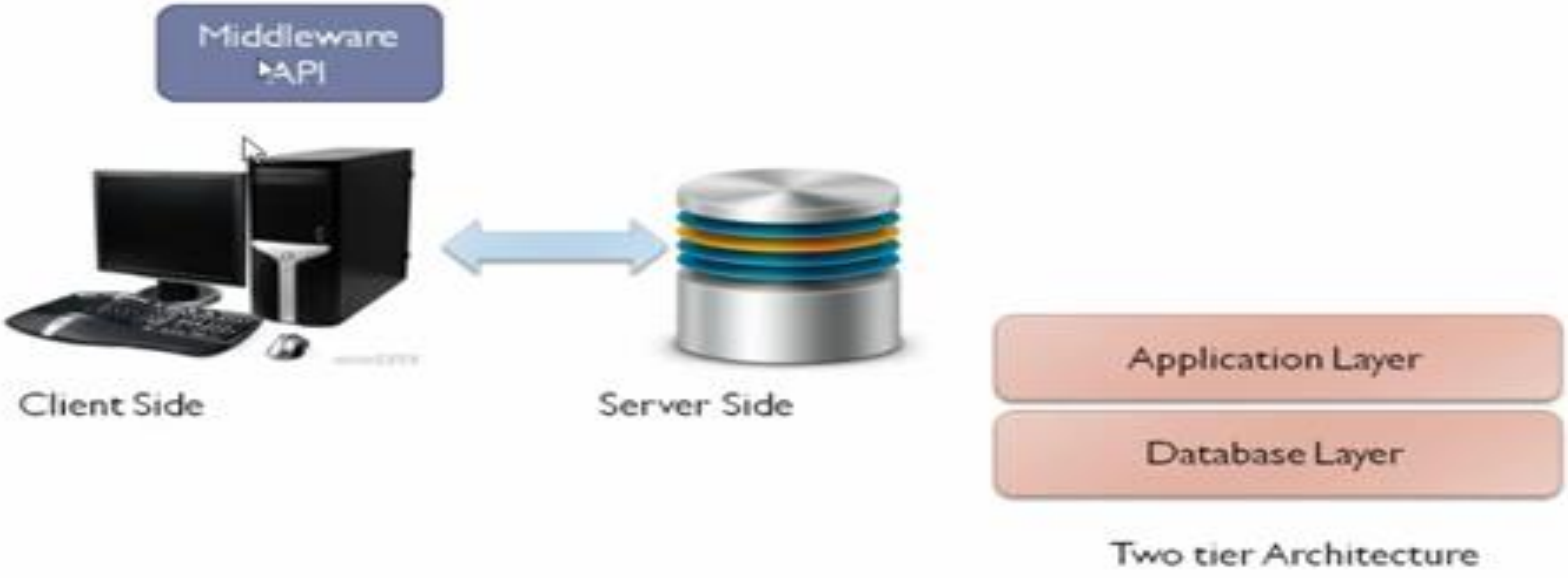
Advantage :

Simple to implement

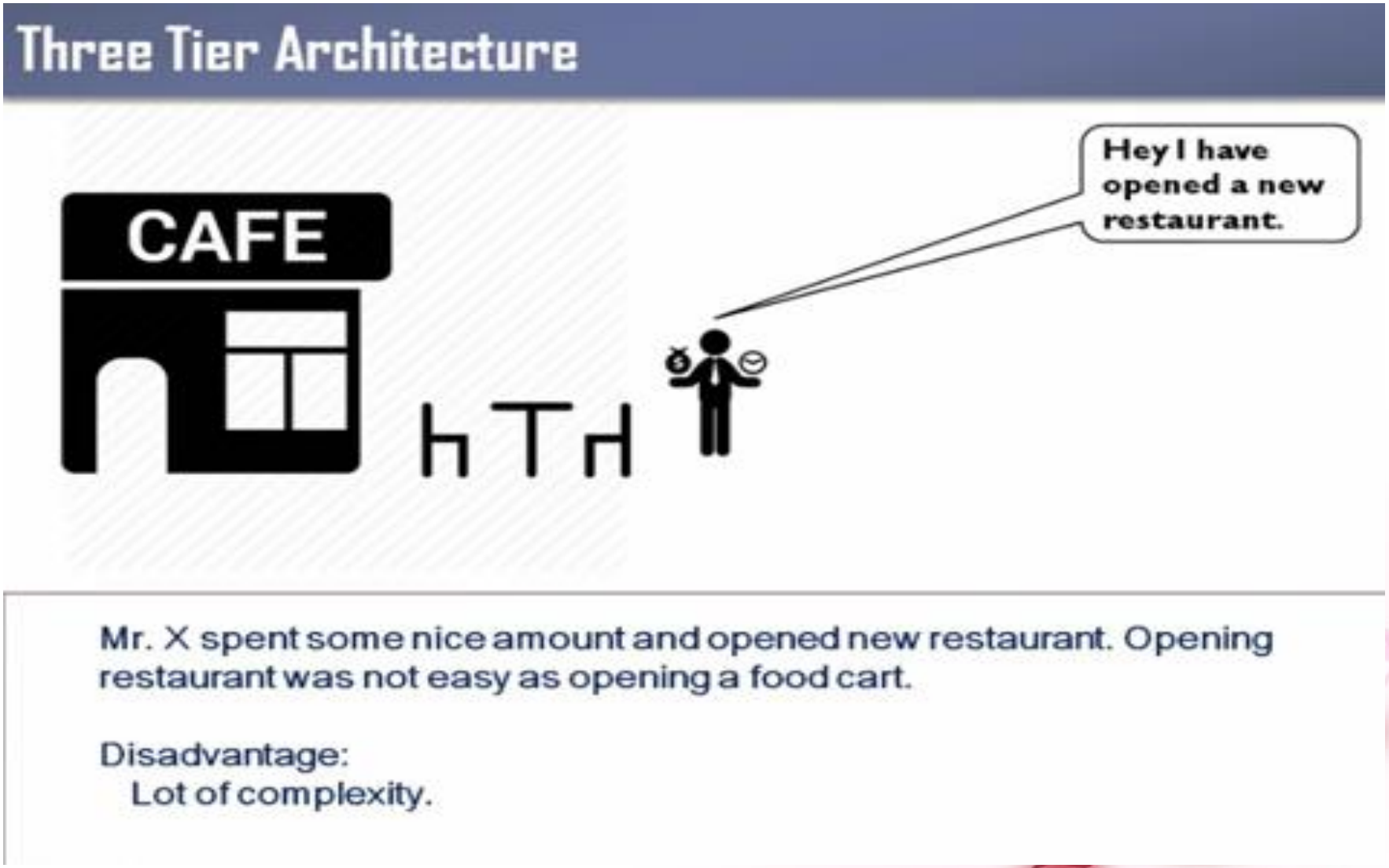
he can’t handle many people at a same time



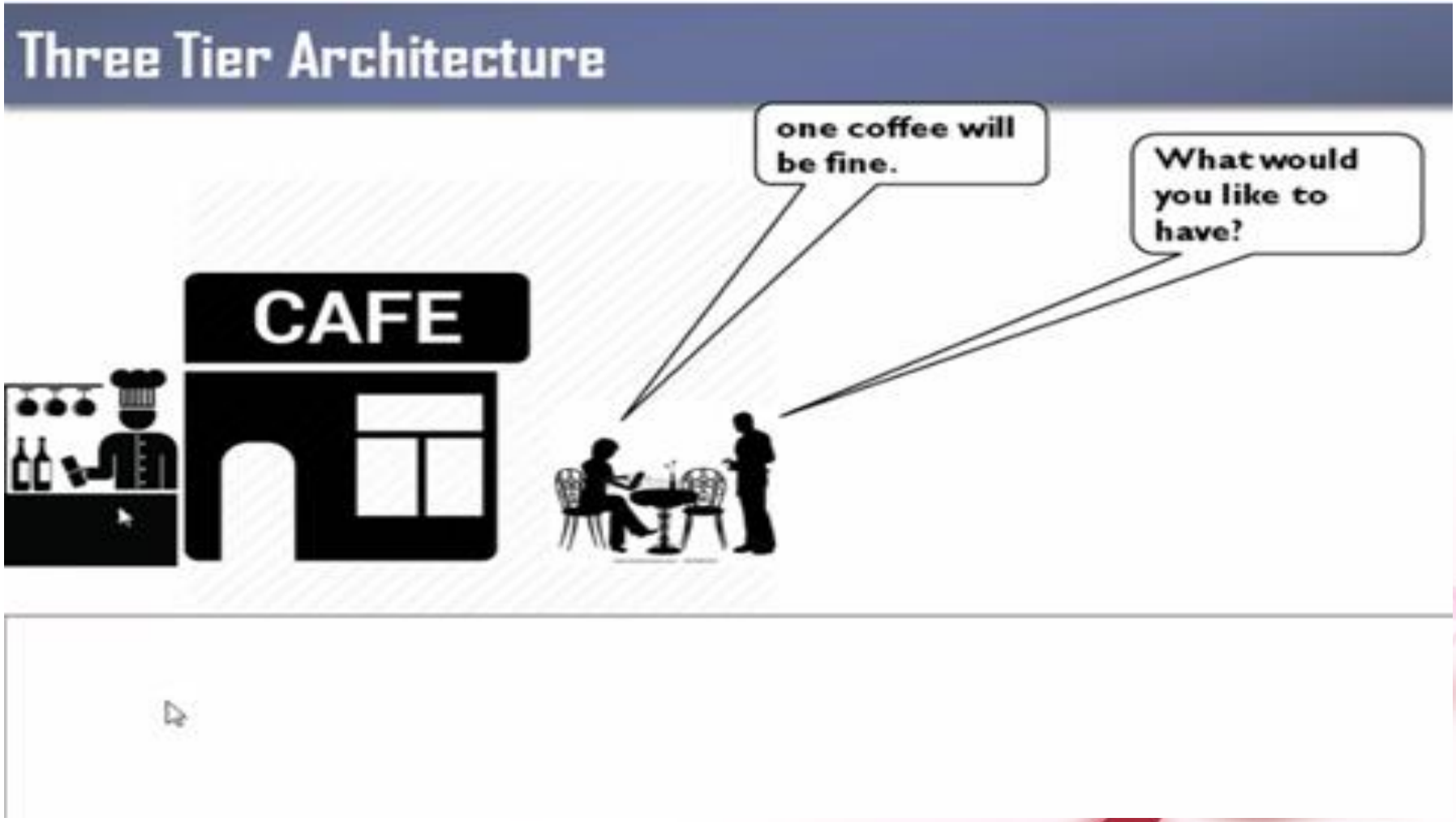
Two Tier Architecture



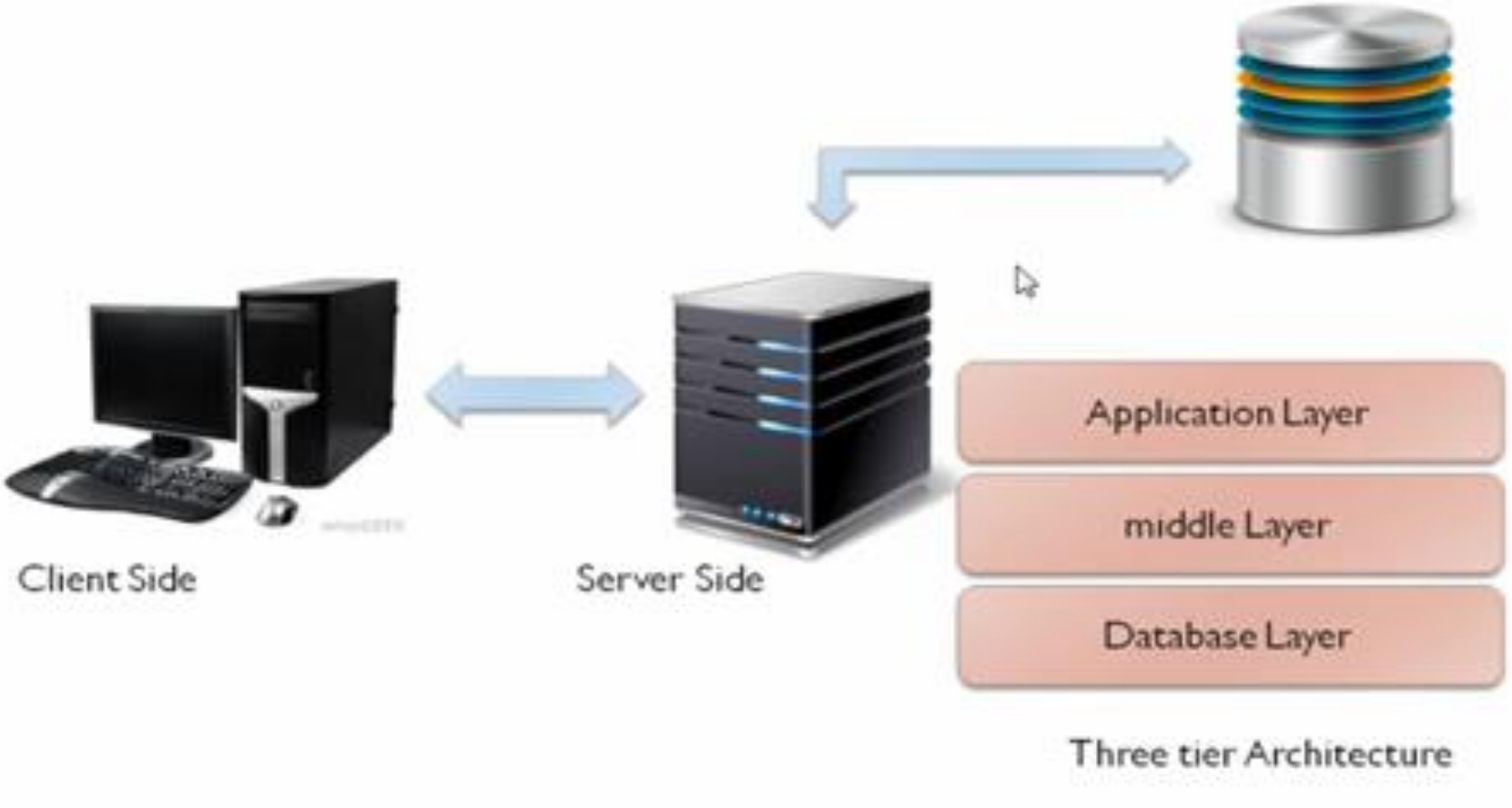
Opened restaurant



Cook has to make food only no need to manage customer waiter manages the customer



Three Tier Architecture



- 1. Physical Level
  2. Conceptual Level
  3. External Level

Architecture

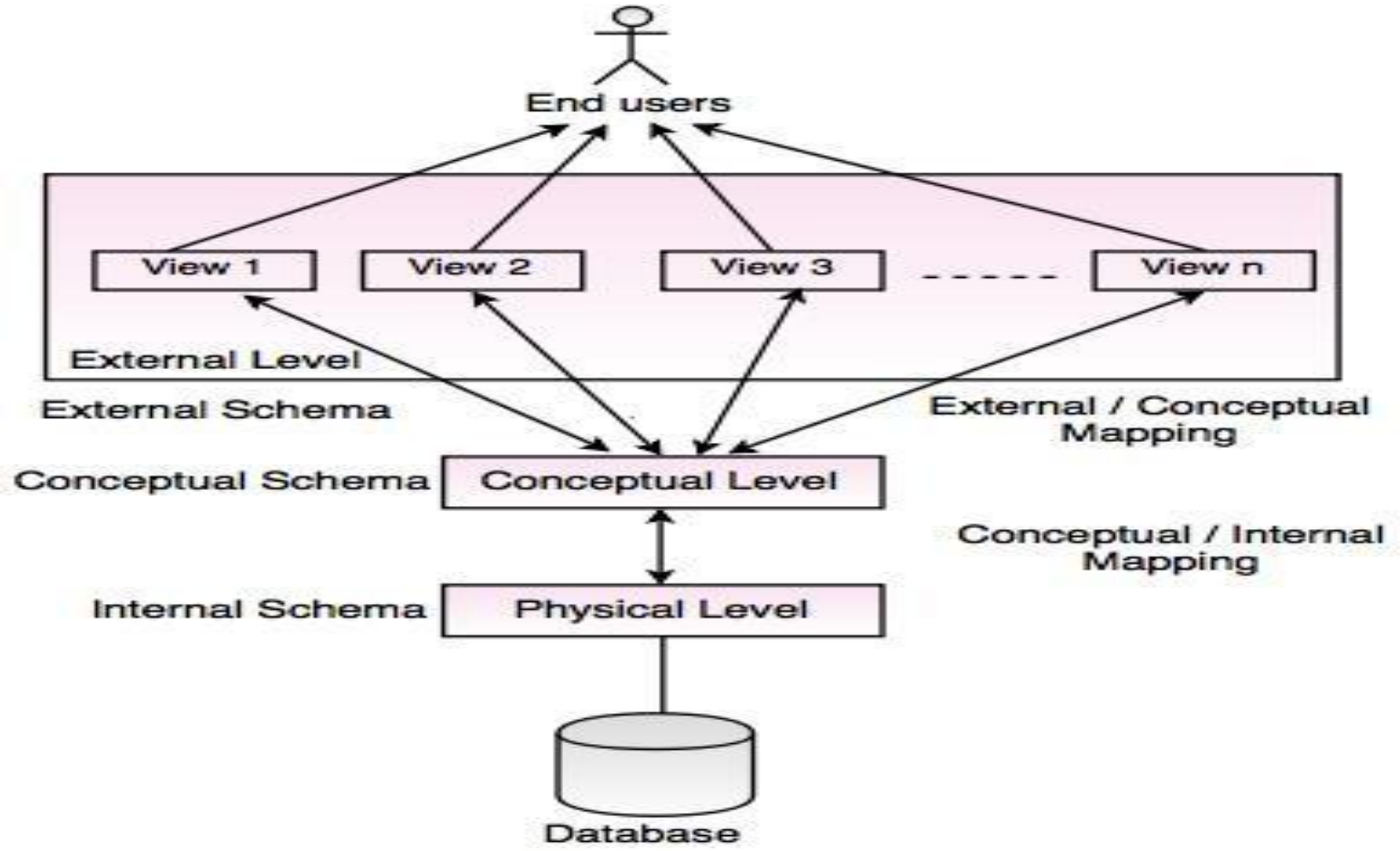
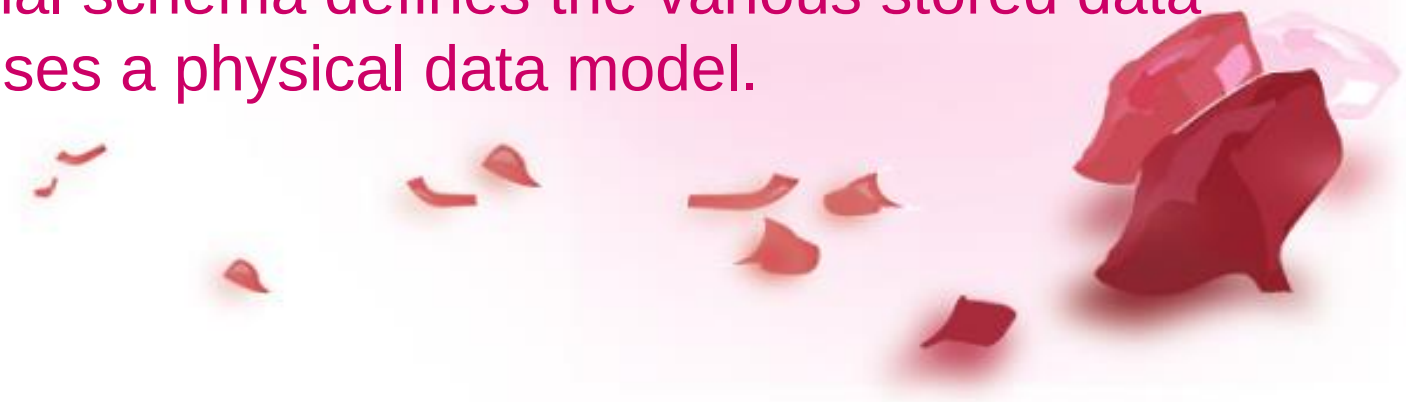


Fig. Three Level Architechture of DBMS

- **In the above diagram,**It shows the architecture of DBMS.
- Mapping is the process of transforming request response between various database levels of architecture.
- Mapping is not good for small database, because it takes more time.
- In External / Conceptual mapping, DBMS transforms a request on an external schema against the conceptual schema.
- In Conceptual / Internal mapping, it is necessary to transform the request from the conceptual to internal levels.

1. Physical Level

- Physical level describes the physical storage structure of data in database.
- It is also known as Internal Level.
- This level is very close to physical storage of data.
- At lowest level, it is stored in the form of bits with the physical addresses on the secondary storage device.
- At highest level, it can be viewed in the form of files.
- The internal schema defines the various stored data types. It uses a physical data model.



2. Conceptual Level

- Conceptual level describes the structure of the whole database for a group of users.
- It is also called as the data model.
- Conceptual schema is a representation of the entire content of the database.
- These schema contains all the information to build relevant external records.
- It hides the internal details of physical storage.

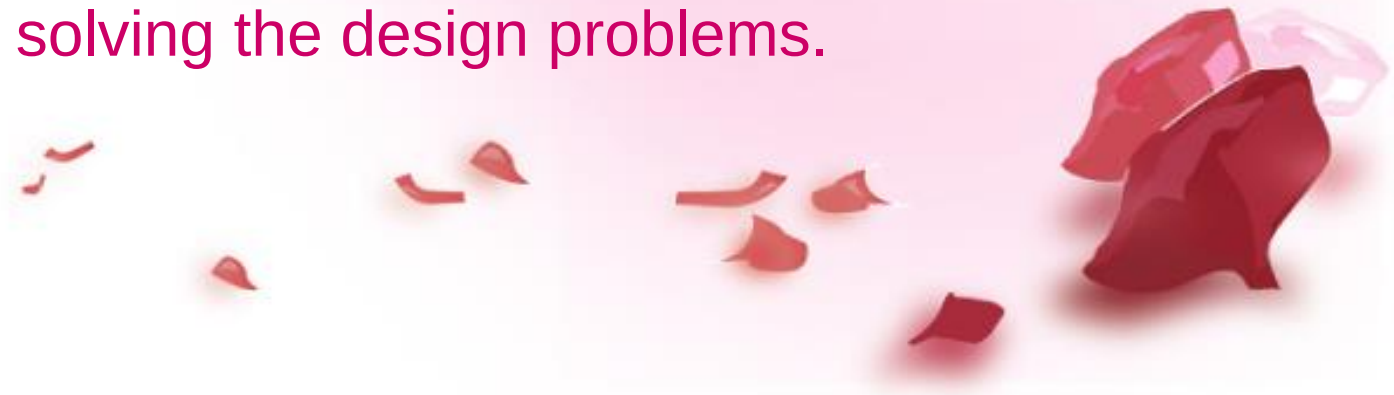
3. External Level

- External level is related to the data which is viewed by individual end users.
- This level includes a no. of user views or external schemas.
- This level is closest to the user.
- External view describes the segment of the database that is required for a particular user group and hides the rest of the database from that user group.

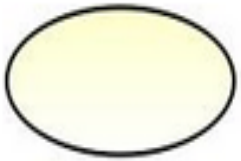
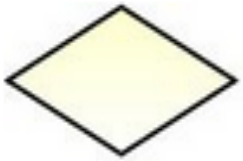
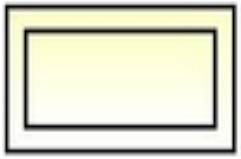
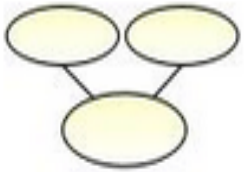
Introduction to ER Model

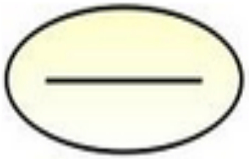
- ER Model is a high-level data model, developed by Chen in 1976. This model defines the data elements and relationships for a specified system. It is useful in developing a conceptual design for the database & is very simple and easy to design logical view of data.**Importance of ER Model**
- ER Model is plain and simple for designing the structure.
- It saves time.

- Without ER diagrams you cannot make a database structure & write production code.
- It displays the clear picture of the database structure.
- **ER Diagrams**
- ERD stands for Entity Relationship diagram.
- It is a graphical representation of an information system.
- ER diagram shows the relationship between objects, places, people, events etc. within that system.
- It is a data modeling technique which helps in defining the business process.
- It used for solving the design problems.



Symbols of ER Diagram

	Rectangle	It represents the Entity.
	Ellipse	It represents the Attribute.
	Diamond	It represents the Relationship.
	Line	It represents the link between attribute and entity set to relationship set.
	Double Rectangle	It represents the weak entity.
	Composite Attribute	It represents composite attribute which can be divided into subparts. For eg. Name can be divided into First Name and Last Name

	Multi valued Attribute	It represents multi valued attribute which can have many values for a particular entity. For eg. Mobile Number.
	Derived Attribute	It represents the derived attribute which can be derived from the value of related attribute.
	Key Attribute	It represents key attribute of an entity which have a unique value in a table. For eg. Employee → EmpId (Employee Id is Unique).

- **Types of Relationship Mapping**
- **Following are the types of Relationship Mapping,**
  1. One - to - One Relationship
  2. One - to - Many Relationship
  3. Many - to - One Relationship
  4. Many - to - Many Relationship

- **1. One - to - One Relationship**In One - to - One Relationship, one entity is related with only one other entity.
- One row in a table is linked with only one row in another table and vice versa.
  For example: A Country can have only one Capital City.

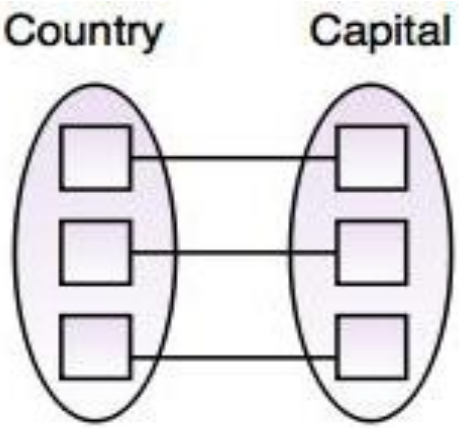


Fig. One to One Mapping

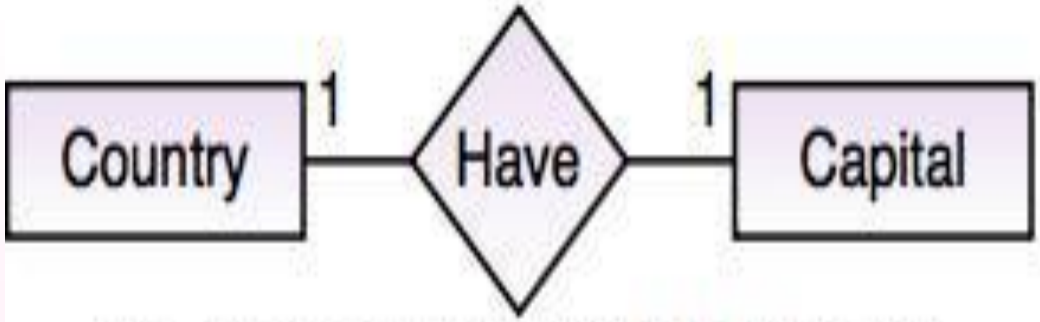


Fig. Representation in ER Diagram

- **2. One - to - Many Relationship**In One - to - Many Relationship, one entity is related to many other entities.
- One row in a table A is linked to many rows in a table B, but one row in a table B is linked to only one row in table A.
  For example: One Department has many Employees.

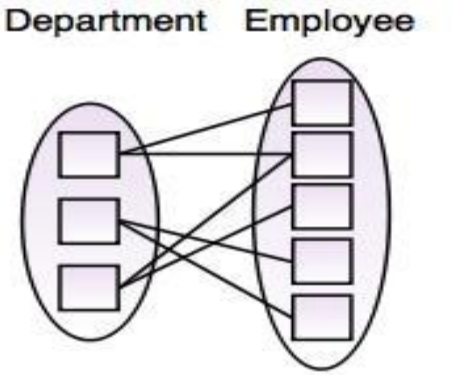


Fig. One to Many Mapping

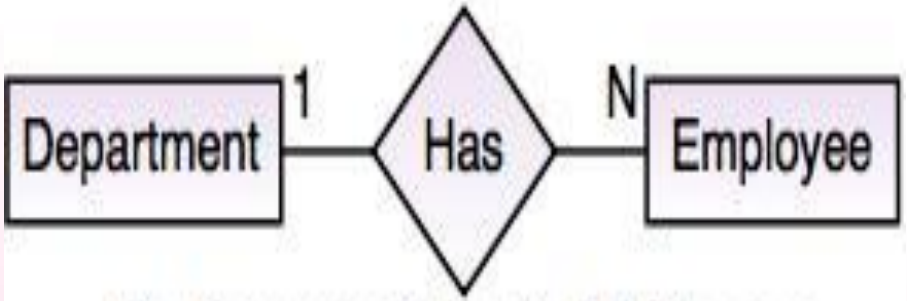


Fig. Representation in ER Diagram

- **3. Many - to - One Relationship**In Many - to - One Relationship, many entities can be related with only one other entity.
  For example: No. of Employee works for Department.
- Multiple rows in Employee table is related with only one row in Department table.

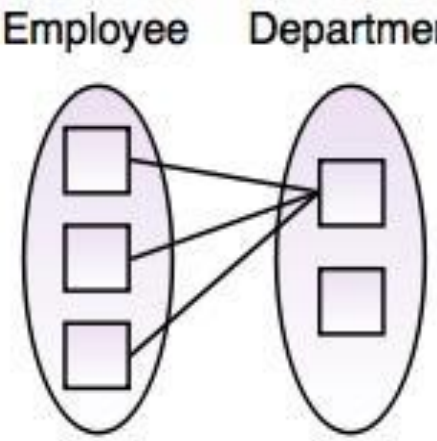


Fig. Many to One Mapping

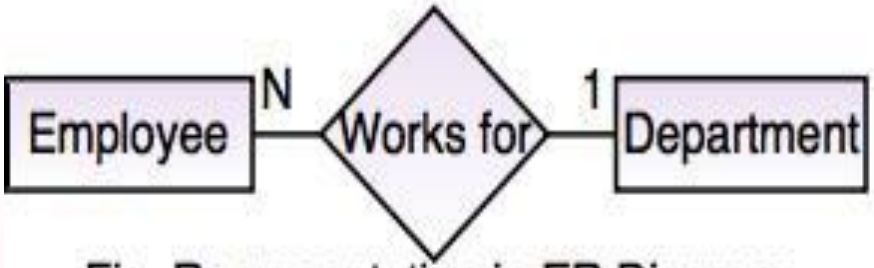
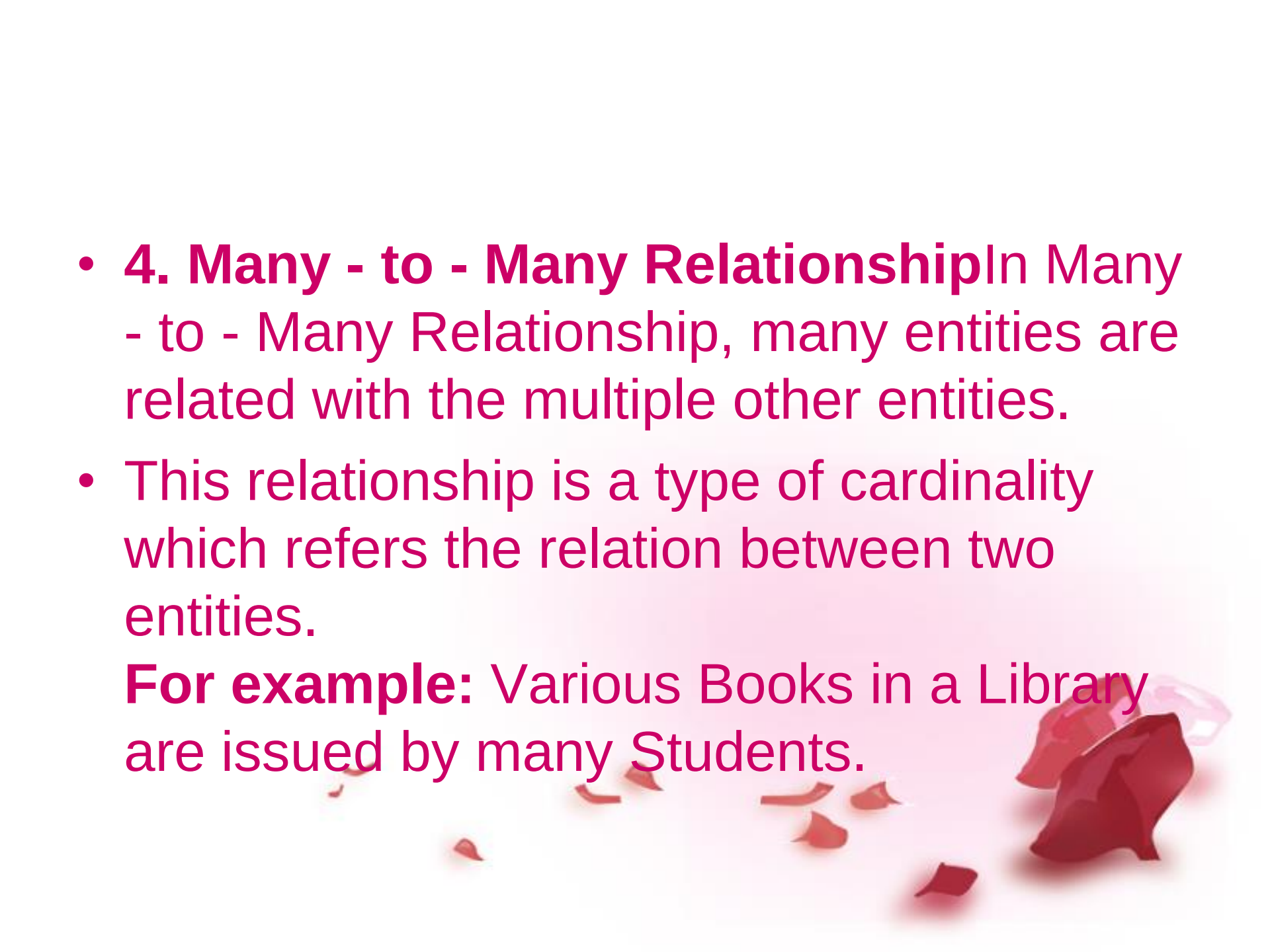


Fig. Representation in ER Diagram

- **4. Many - to - Many Relationship**In Many - to - Many Relationship, many entities are related with the multiple other entities.
- This relationship is a type of cardinality which refers the relation between two entities.
  For example: Various Books in a Library are issued by many Students.



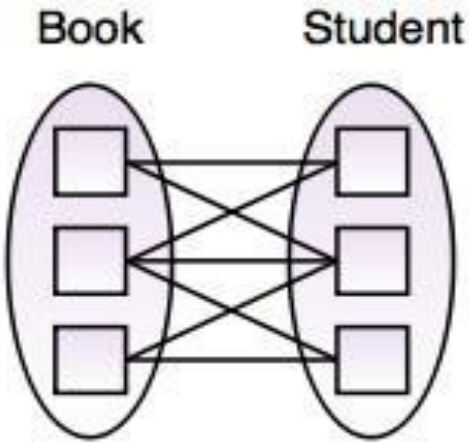


Fig. Many to Many Mapping

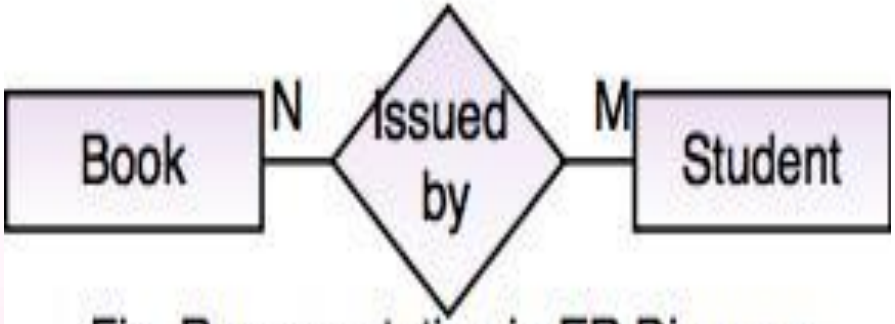


Fig. Representation in ER Diagram

Participation Constraints

- **Following are the two types of Participation Constraints,**
  1. Total Participation
  2. Partial Participation

  1. Total Participation In Total Participation, every entity in the set is involved in some association of the relationship.
- It is indicated by a double line () between entity and relationship.
  For example: Every Department must have a Manager.

For example: Every Department must have a Manager.

Fig. Total Participation

2. Partial Participation

- In Partial Participation, not all entities in the set are involved in association of the relationship.
- It is indicated by a single line (——) between entity and relationship.

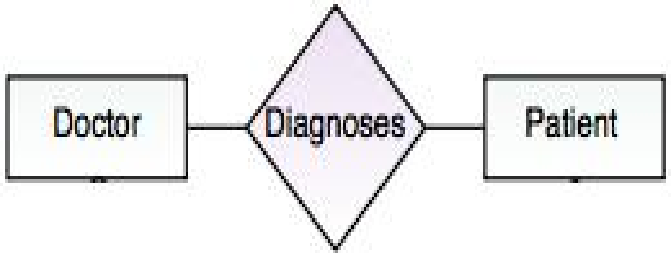
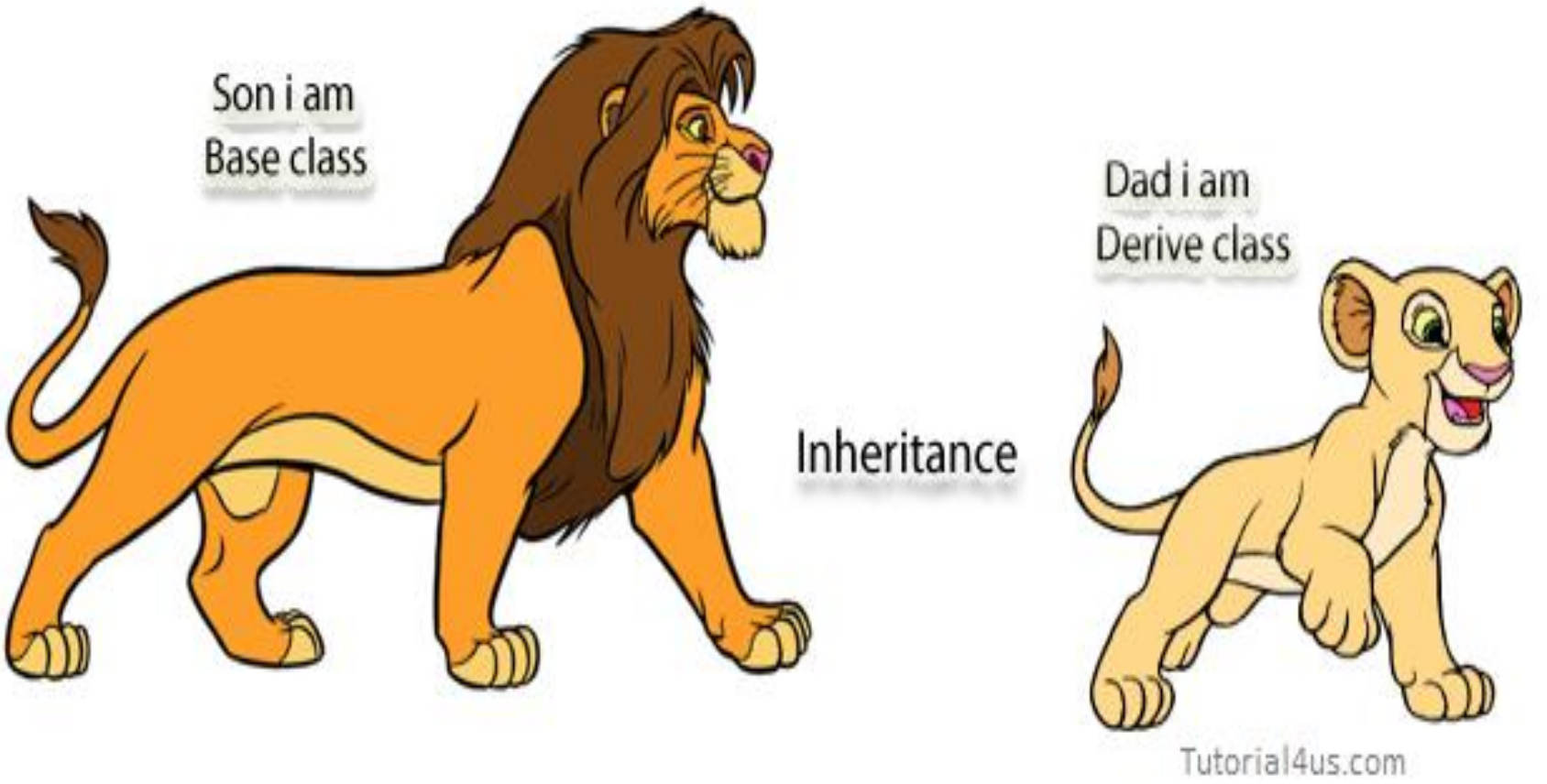


Fig. Partial Participation

EER Model

- EER is a high-level data model that incorporates the extensions to the original ER model.
  It is a diagrammatic technique for displaying the following concepts Sub Class and Super Class
- Specialization and Generalization
- Union or Category
- Aggregation
- These concepts are used when the comes in EER schema and the resulting schema diagrams called as EER Diagrams.**Features of EER Model**
- EER creates a design more accurate to database schemas.
- It reflects the data properties and constraints more precisely.
- It includes all modeling concepts of the ER model.
- Diagrammatic technique helps for displaying the EER schema.
- It includes the concept of specialization and generalization.
- It is used to represent a collection of objects that is union of objects of different of different entity types.


Son i am
Base class
Inheritance
Dad i am
Derive class
Tutorial4us.com

- **A. Sub Class and Super Class**Sub class and Super class relationship leads the concept of Inheritance.
- The relationship between sub class and super class is denoted with symbol.
- **1. Super Class**Super class is an entity type that has a relationship with one or more subtypes.
- An entity cannot exist in database merely by being member of any super class.
  For example: Shape super class is having sub groups as Square, Circle, Triangle.
- **2. Sub Class** Sub class is a group of entities with unique attributes.
- Sub class inherits properties and attributes from its super class.
  For example: Square, Circle, Triangle are the sub class of Shape super class.

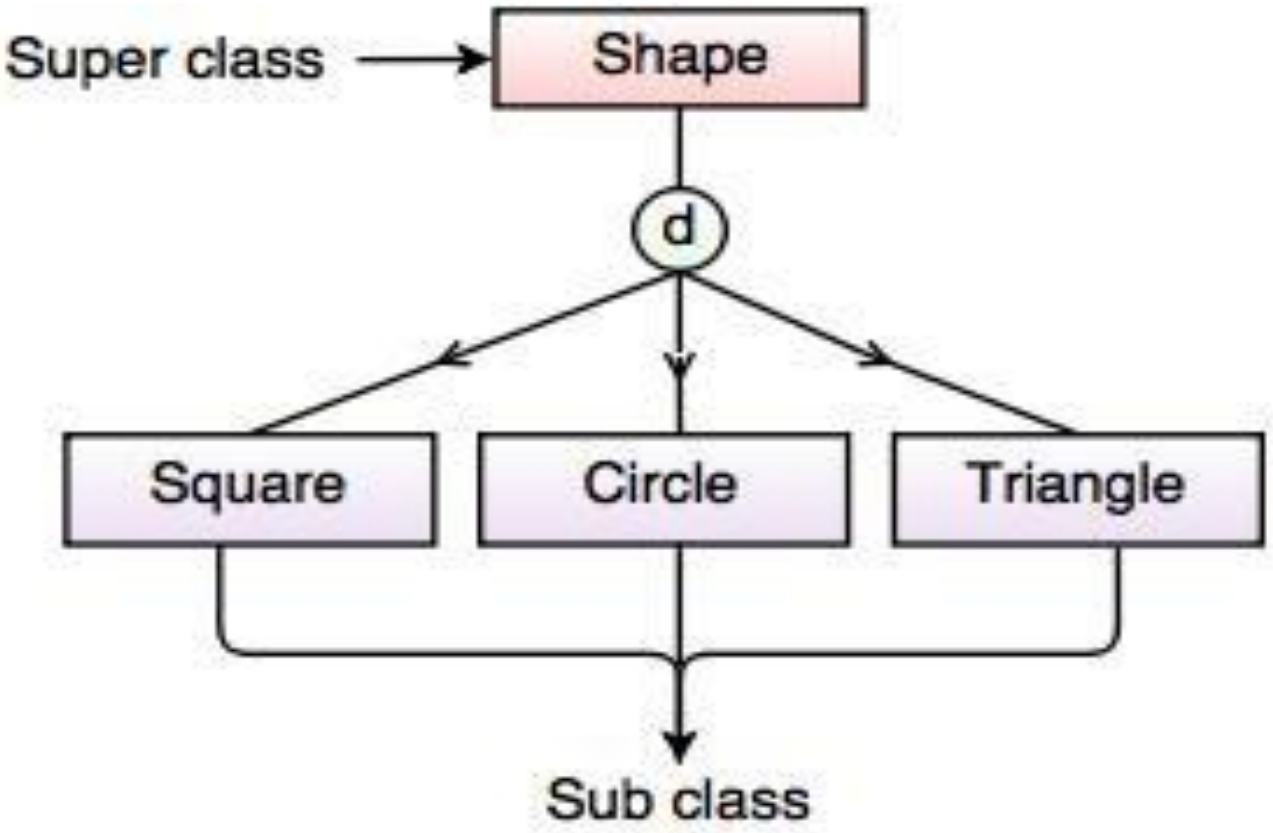


Fig. Super class/Sub class Relationship

B. Specialization and Generalization

1. Generalization Generalization is the process of generalizing the entities which contain the properties of all the generalized entities.

- It is a bottom approach, in which two lower level entities combine to form a higher level entity.
- Generalization is the reverse process of Specialization.
- It defines a general entity type from a set of specialized entity type.
- It minimizes the difference between the entities by identifying the common features.

For example:

In the above example, Tiger, Lion, Elephant can all be generalized as Animals.

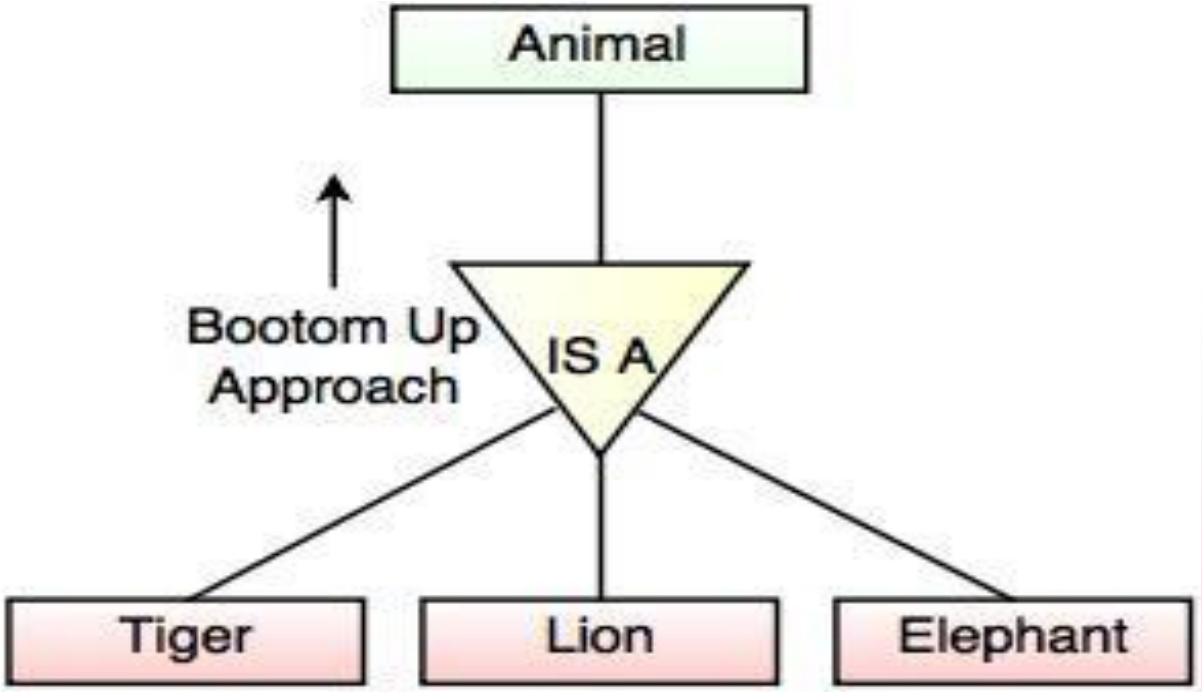


Fig. Generalization

2. Specialization

- Specialization is a process that defines a group entities which is divided into sub groups based on their characteristic.
- It is a top down approach, in which one higher entity can be broken down into two lower level entity.
- It maximizes the difference between the members of an entity by identifying the unique characteristic or attributes of each member.
- It defines one or more sub class for the super class and also forms the superclass/subclass relationship.

In the above example, Employee can be specialized as Developer or Tester, based on what role they play in an Organization.

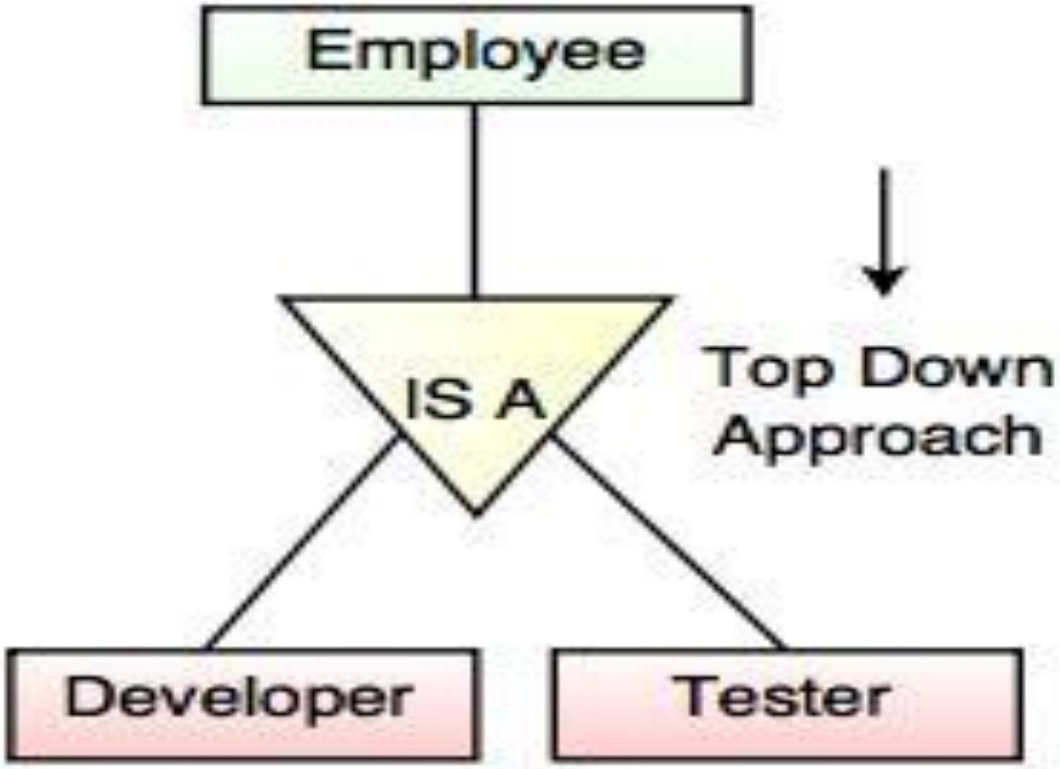


Fig. Specialization

C. Category or Union Category

- represents a single super class or sub class relationship with more than one super class.
- It can be a total or partial participation.
  For example Car booking, Car owner can be a person, a bank (holds a possession on a Car) or a company. Category (sub class) → Owner is a subset of the union of the three super classes → Company, Bank, and Person. A Category member must exist in at least one of its super classes.

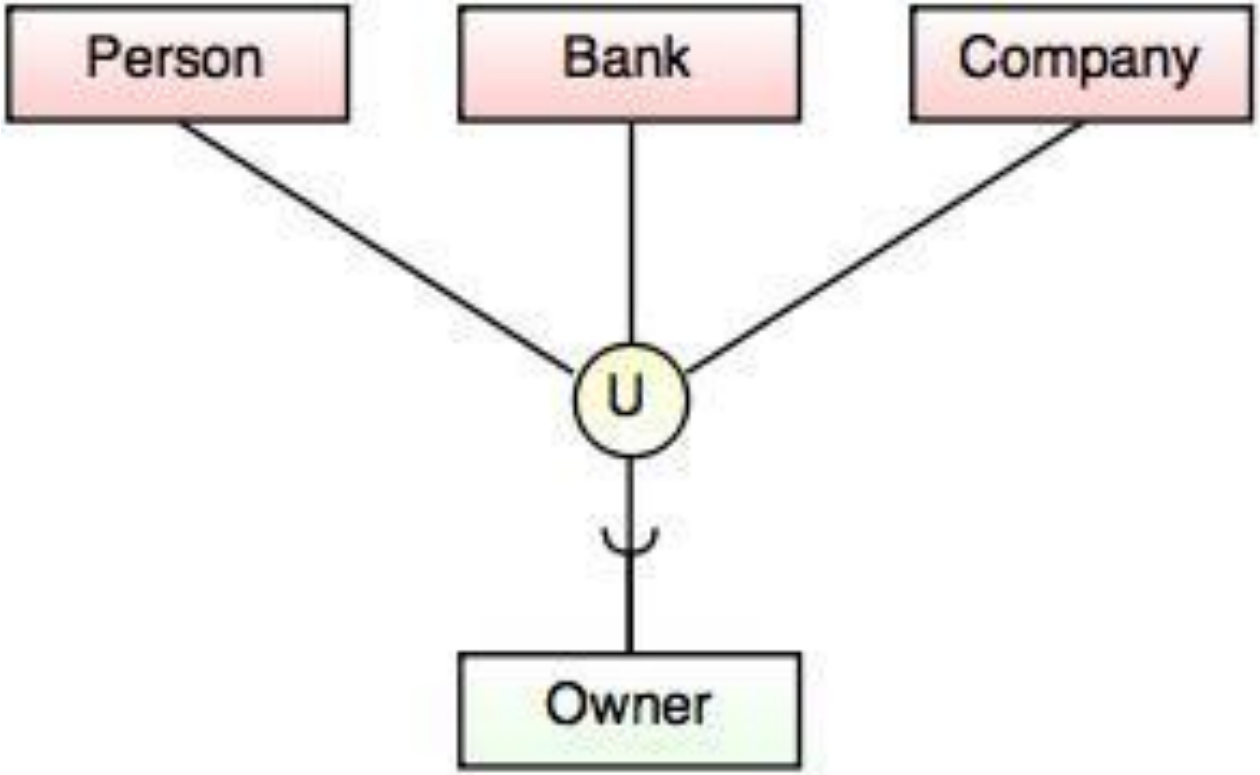


Fig. Categories (Union Type)

- **D. Aggregation**Aggregation is a process that represent a relationship between a whole object and its component parts.
- It abstracts a relationship between objects and viewing the relationship as an object.
- It is a process when two entity is treated as a single entity.

In the above example, the relation between College and Course is acting as an Entity in Relation with Student.

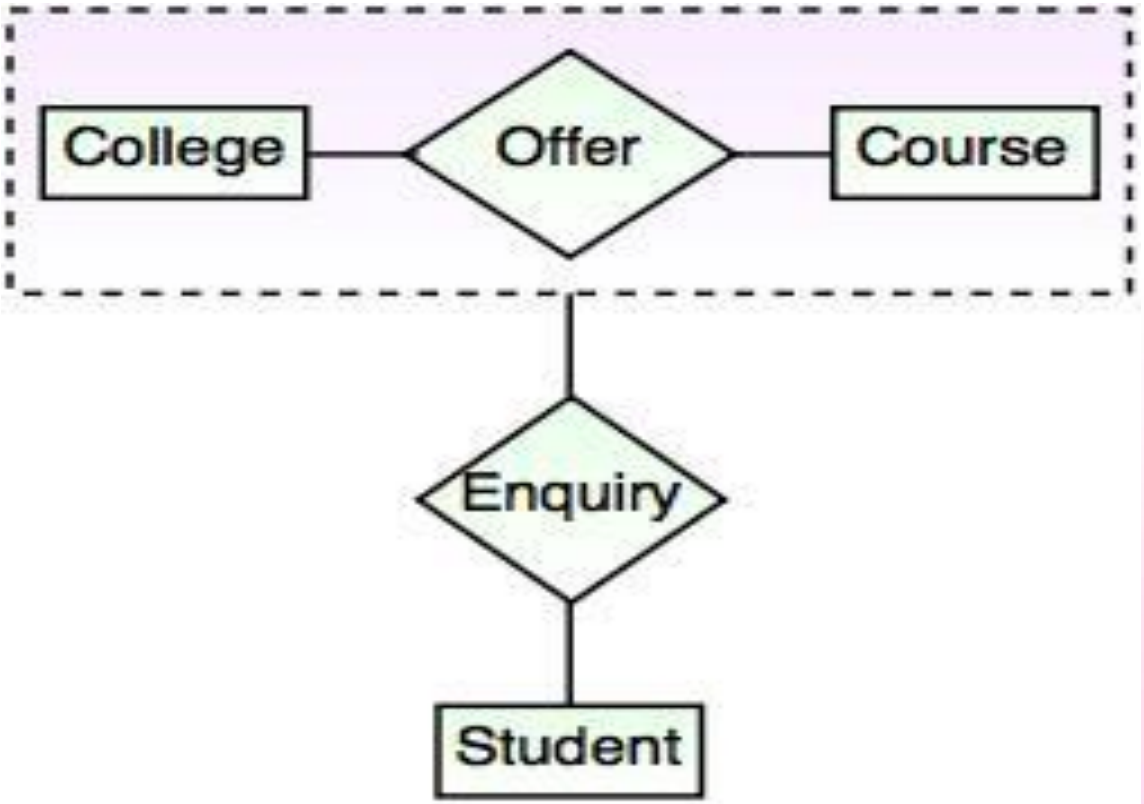
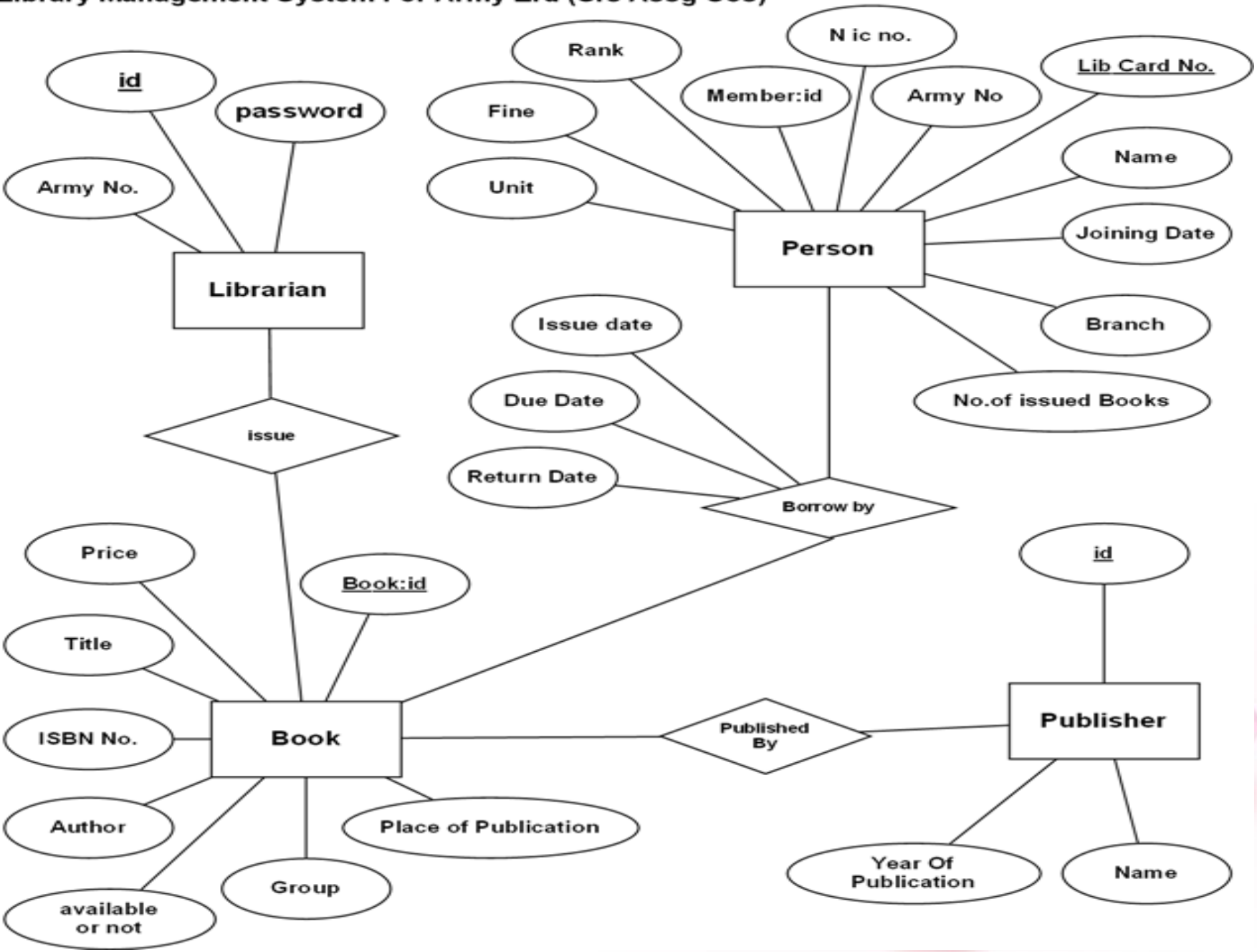
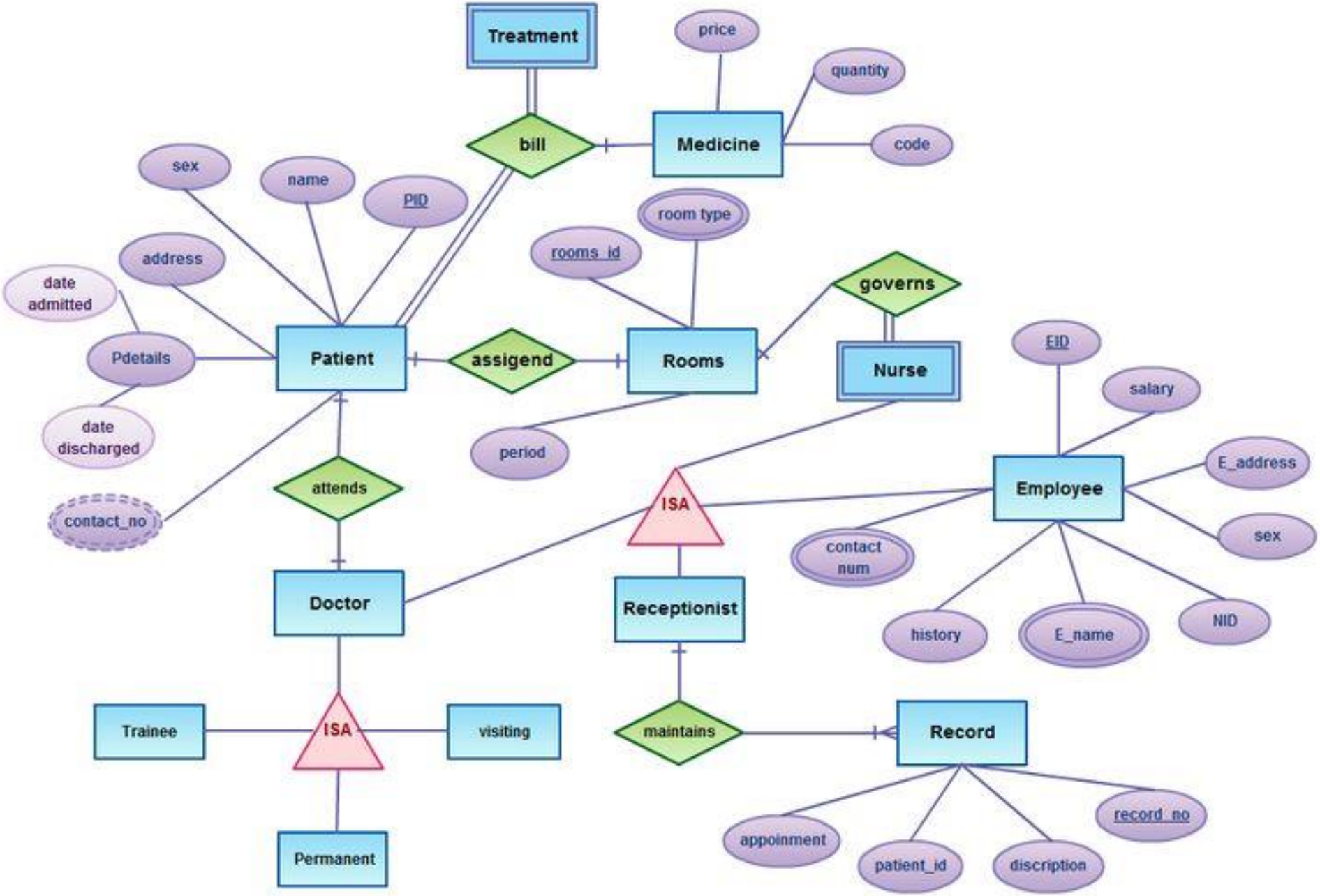


Fig. Aggregation

Library Management System For Army Erd (Srs Assg Uos)

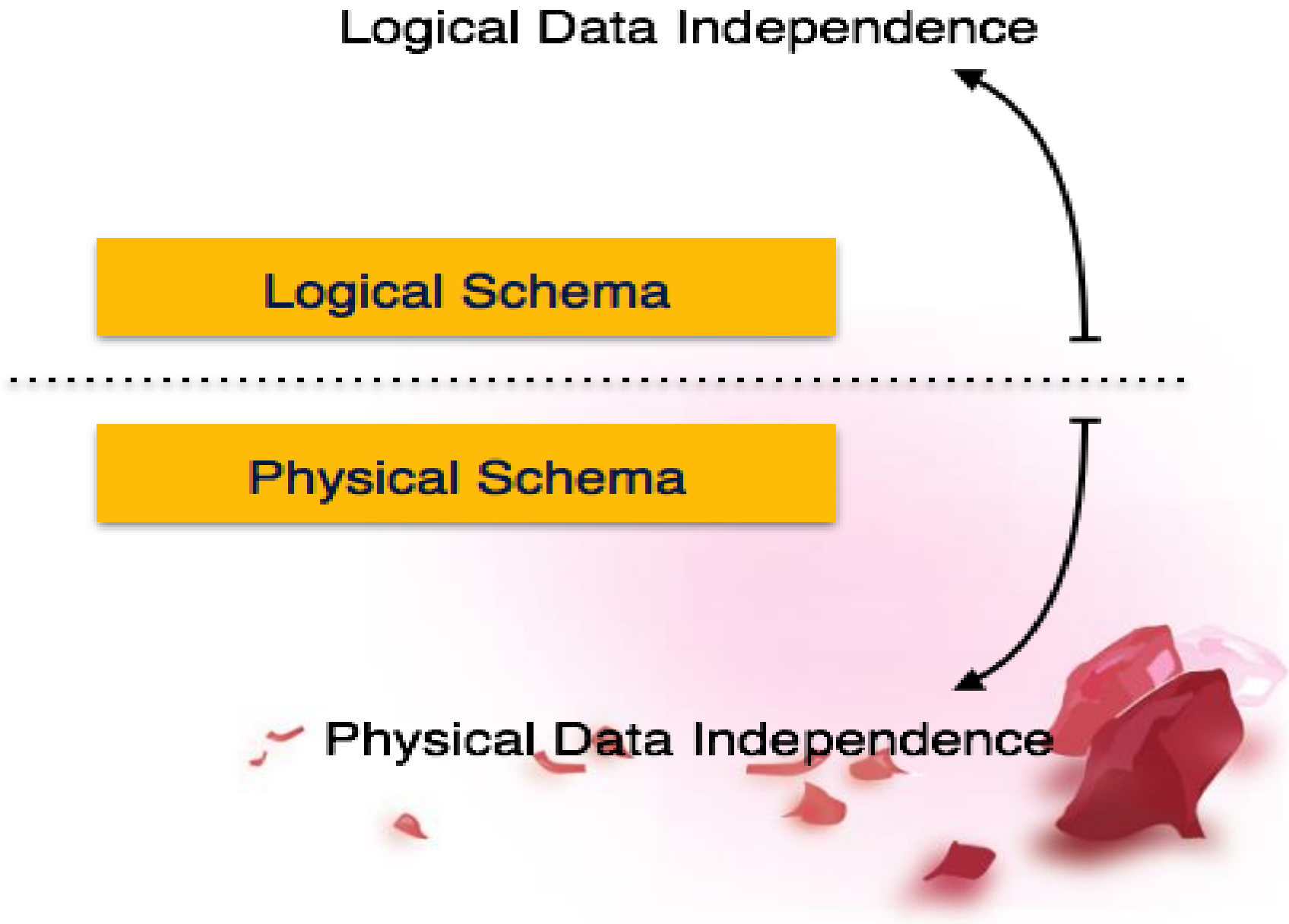


E-R Diagram for Hospital Management System



DATA INDEPENDENCE

- **Data Independence**
- A database system normally contains a lot of data in addition to users' data. For example, it stores data about data, known as metadata, to locate and retrieve data easily. It is rather difficult to modify or update a set of metadata once it is stored in the database. But as a DBMS expands, it needs to change over time to satisfy the requirements of the users. If the entire data is dependent, it would become a tedious and highly complex job.
- Metadata itself follows a layered architecture, so that when we change data at one layer, it does not affect the data at another level. This data is independent but mapped to each other.


Logical Data Independence
Logical Schema
Physical Schema
Physical Data Independence

Logical Data Independence

- Logical data is data about database, that is, it stores information about how data is managed inside. For example, a table (relation) stored in the database and all its constraints, applied on that relation.
- Logical data independence is a kind of mechanism, which liberalizes itself from actual data stored on the disk. If we do some changes on table format, it should not change the data residing on the disk.

Physical Data Independence

- All the schemas are logical, and the actual data is stored in bit format on the disk. Physical data independence is the power to change the physical data without impacting the schema or logical data.
- For example, in case we want to change or upgrade the storage system itself − suppose we want to replace hard-disks with SSD − it should not have any impact on the logical data or schemas.

mapping

- (i) External-Conceptual mapping (ii) Conceptual-Internal mapping
- External-Conceptual Mapping
- An external-conceptual mapping defines the correspondence between a particular external view and the conceptual view. The external-conceptual mapping tells the DBMS which objects on the conceptual level correspond to the objects requested on a particular user's external view. If changes are made to either an external view or conceptual view, then mapping must be changed accordingly.

- Conceptual-Internal Mapping
- The conceptual-internal mapping defines the correspondence between the conceptual view and the internal view, i.e. database stored on the physical storage device. It describes how conceptual records are stored and retrieved to and from the storage device. This means that conceptual-internal mapping tells the DBMS that how the conceptual! records are physically represented. If the structure of the stored database is changed, then the mapping must be changed accordingly. It is the responsibility of DBA to manage such changes.

Reduction to E-R database schema

Questions

1) Define: Data, Database, DBMS, Data Redundancy.

3) Explain the purpose and application of DBMS.

4) List the benefits of database approach.

5) Describe various disadvantages of file system compared to Data base management system.

7) Explain different database users.

8) What are the responsibilities of a DBA?

11) Explain three level architecture of database system.

13) What is data independence? Explain the difference between physical and logical data independence with example

Questions

- 1) What is mapping cardinalities? Explain it with real time examples.
- 3) Explain types of attribute with example
- 4) Explain Specialization feature of ER diagram with example.
- 5) Explain Generalization feature of E-R Diagram.
- 6) Explain aggregation operation of ER diagram.
- 7) Draw an ER diagram for exam system of INDUS UNIVERSITY.
- 8) Construct E-R diagram for a hospital with a set of patients and medical doctors.
- Associate with each patient a log of various tests and examinations conducted.

9) Construct E-R diagram of the bank. It provides different kinds of bank accounts. And loans. It operates number of branches.

10) Draw E – R Diagram for the School Management System.

11) Give Symbol used in E-R Diagram and Draw the E-R diagram of Library Management System.

12) Draw E-R diagram for supplier who supplies different parts. The parts are used in different projects. Explain the mapping cardinality used. Assume suitable attributes.

13) Construct an E-R Diagram for an insurance company with a set of customers, each of whom owns number of cars, also each can have number of recorded accidents associated with it.

14) What is constraint in database? Explain types of constraints with suitable example.

15) Draw symbols for following in E-R diagram:

- a. Weak Entity set, Derived attribute, Multivalued
- b. Relationship Set, and Primary key attribute