DBMS
UNIT-I

Data Base Management System (DBMS):

Database + Management System = DBMS.

↓ collection of data

↓ set of program to store and retrieve (access) those data.

Definition:-

DBMS is a collection of interrelated data and a set of programs to access and store those data in an efficient and efficient way. The primary goal of DBMS is to provide efficien

It is a software which is used to manage database

Eg:- MySQL, Oracle etc are popular DBMS softwares used in different applications.

Purpose:-

Previously data was stored in files. We have some drawbacks of filesystem to overcome that we need DBMS.

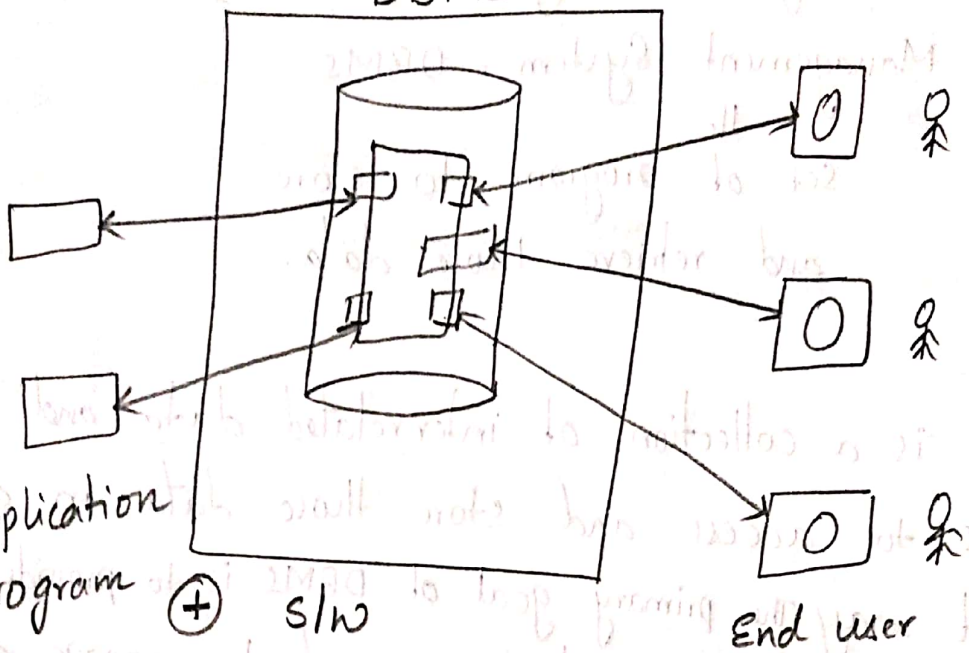
Drawbacks of filesystem are:-

1. Data Redundancy & Inconsistency.
2. Difficulty in accessing data.
3. Data Isolation
4. Data Security
5. Transaction problems.

DBMS solves all the above problems.

Data base :- ⇒ collection of data

DBMS ⇒ s/w ↗ manage the db



DBMS is a software

> oracle, MySQL, SQL, PL SQL

Application program → db.

> C, C++, Java

Historical Perspective :-

- Early 1960 : First General purpose database developed by Charles Bachman from GE, used Network Model data.
- Late 1960 : IBM developed information management System (IMS) used in Airline reservation System.

→ It is used in hierarchical data model

→ Still it is used

- 1970 : Edgar code of IBM developed "RELATIONAL DATA MODEL"
- 1980, 1990 : Types of DBMS introduced

What is DBMS?

A Database Management System (DBMS) is a system software that allows users to efficiently define, create, maintain and share databases.

> Defining a database involves specifying the datatypes, structures and constraints of data to be stored in database

> Creating a database involves storing the data on some storage medium that is controlled by DBMS.

> Maintaining a database involves updating the database whenever required to reflect the changes.

> DBMS also serves as an interface between database and end users or application programs.

Advantages of DBMS over file System:

- Data redundancy and Inconsistency:-

Redundancy means repetition of data. File system cann Control redundancy of data. There may be the possibility tha two users may maintain same file for different applications whereas DBMS controls redundancy problem. by maintaining single repository of data that is defined once and accessed by many users.

- Data sharing:- File system does not allow sharing of data whereas in DBMS data can be shared easily due to central system.

- Data Concurrency:- concurrent access to data means more th one user is accessing the same data at same time. Anomal occur when changes made by one user gets lost because of changes made by other users. Filesystem does not provide

any procedure to stop anomolies. whereas DBMS provides a locking system to stop anomolies.

- Data searching:- For every search operation performed on file system, a different application program has to be written whereas DBMS provides inbuilt searching operations. Users only have to write a small query to retrieve data from database.

Why to use DBMS?

- To develop software applications in less time.
- Data Independence and efficient use of data
- Uniform Data administration (# storage of data & retrieval of data)
- Data Integrity and Security
- Concurrent acces to data, data recovery from crashes.
- Uses user-friendly querry language.

Where is DBMS used?

Universities: registrations, results, grades, etc

Sales: Products, purchases, customers etc

Telecom: calls made, customer details, network usage.

Airlines: Reservations, schedules

Banking: All transactions.

BATU F S → Airlines, Sales, Finance, University, Tele comm., Bank

Advantages of DBMS:

DBMS has many advantages. They are:

1. Data Independence
2. Efficient data access
3. Data Integrity and security
4. Data administration
5. Concurrent access and crash Recovery
6. Reduced application Development Time

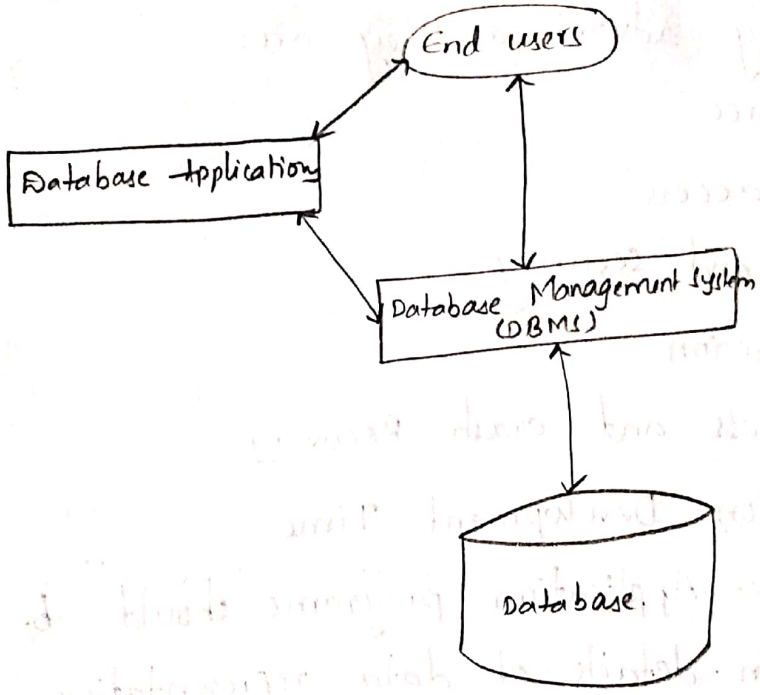
Data Independence:- Application programs should be as free as possible from details of data representation and storage. DBMS provides abstract view of data that hides such details.

Efficient Data access:- DBMS utilizes a mixture of sophisticated concepts and techniques for storing and retrieving data efficiently. This feature is especially important if data is stored on External storage devices.

Data Integrity and security:- If data is accessed through DBMS, DBMS can enforce integrity constraints.

Data administration:- When several users share the data centralizing the administration of data can offer significant improvement. Experienced professionals understand the nature of data being managed and can be responsible for organizing data representation to reduce redundancy and make data to retrieve efficiently.

Components of DBMS:-



Users: users may be any kind of such as DB administrator, system developer or database users.

Database applications:- Database application may be Departmental, Personal, organizations.

DBMS:- software that allows users to create and manipulate database access.

Database: collection of logical data as a single unit.

Data Models:-

Underlying the structure of a database is data model. It is a collection of conceptual tools for describing data, data relationships, data semantics and consistency constraints. A datamodel provides a way to describe design of database at physical, logical and view levels.

There are different data models. They are classified into four different categories

- Relational Model
- Entity - Relationship Model
- Object - based data Model
- Semistructured Data Model

Network Datamodels.
Hierarchical model

Relational Model:-

The relational model uses a collection of tables to represent both data and the relationships among those data. Each table has multiple columns and each column has unique name. Tables are also known as relations.

The relational model is an example of record-based model. Each table contains records of particular type. Each record type defines a fixed number of fields or attribute. This is the most widely used data model and majority of current databases systems are based on relational model.

Table name
Student

column name

Roll No	Name	Phone
1	Alex	444123
2	Aryan	421456
3	Parker	414259
4	John	456785

Tuple / Row
Table / Relation
Attribute / column

Relational model terms

Relation: A relation is a table with columns and rows

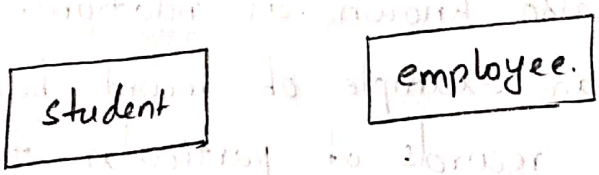
Attribute: An attribute is a named column of a relation

Tuple: A tuple is a row of relation.

Entity - Relationship model:

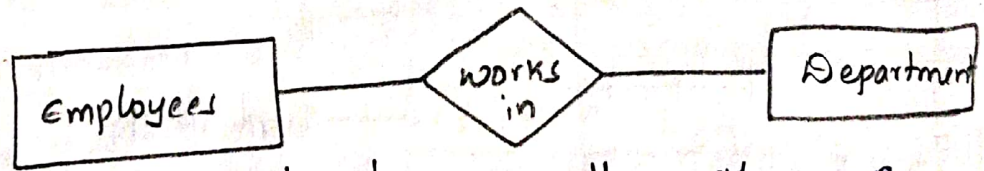
The entity-relationship (E-R) data model uses a collection of basic objects called entities and relationships among these objects. An entity is a thing or object in the real world. The ER model is widely used in database design.

Entities are represented using rectangle shape box.



Entities.

Relationships are represented by diamond-shaped box. All the entities participating in a relation gets connected using a line. Relationship is an association among two or more entities.



Four types of relationships are there. They are

- One-to-One
- One-to-many
- Many-to-one
- Many-to-Many.

Object-base data model:-
(UML) in next semister It is an extension of E-R model with notions of encapsulation and object identity.

Levels of abstraction:-

The data in DBMS is described at three levels.

They are:

1. Physical level
2. Logical level / conceptual
3. view level.

* The process of hiding irrelevant details or information from an user is data abstraction.

*Physical level:- The lowest level of abstraction describes how the data are actually stored. The physical level describes complex low-level data structures in detail.

*Logical level: The next higher level of abstraction describes what data are stored in database and what relationships exist among those data. It describes the entire database in terms of small number of simple structures. Database administrator who must decide what information to keep in database.

*view level: The highest level of abstraction describes only part of entire database. The system provides many views for same database.

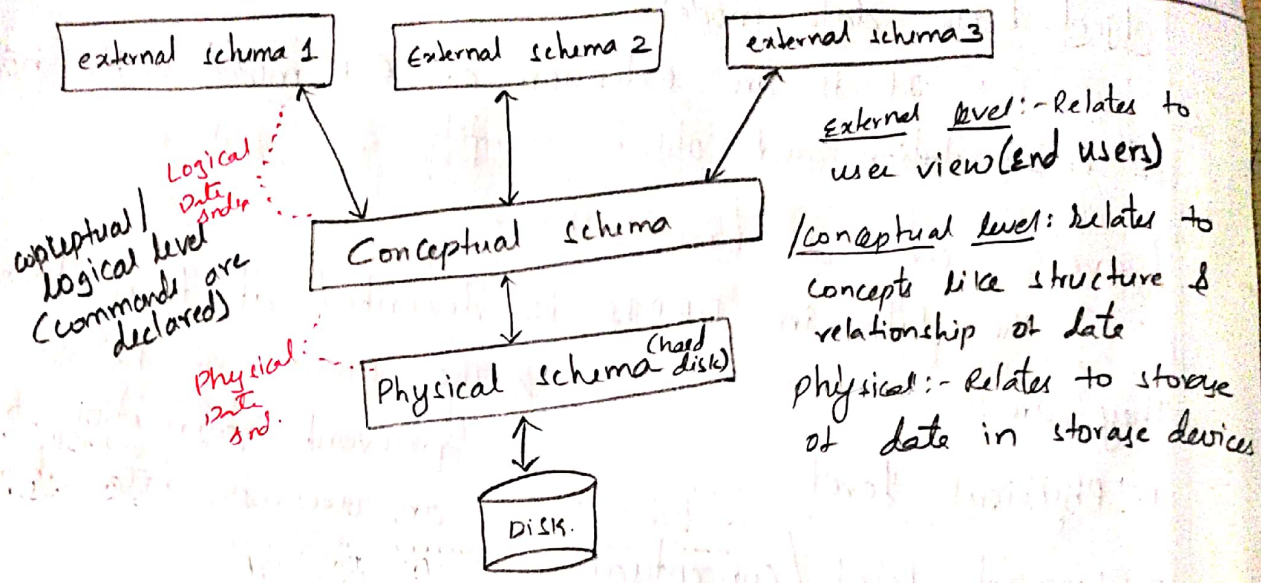


Fig: levels of abstraction

Data Independence:-

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

In this we have two types. They are

1. Physical data Independence.
2. Logical data independence

Physical data Independence:- It helps you to seperate conceptual level from internal/physical levels. It allows you to provide a logical description of database without the need to specify physical structure.

Logical data Independence:-

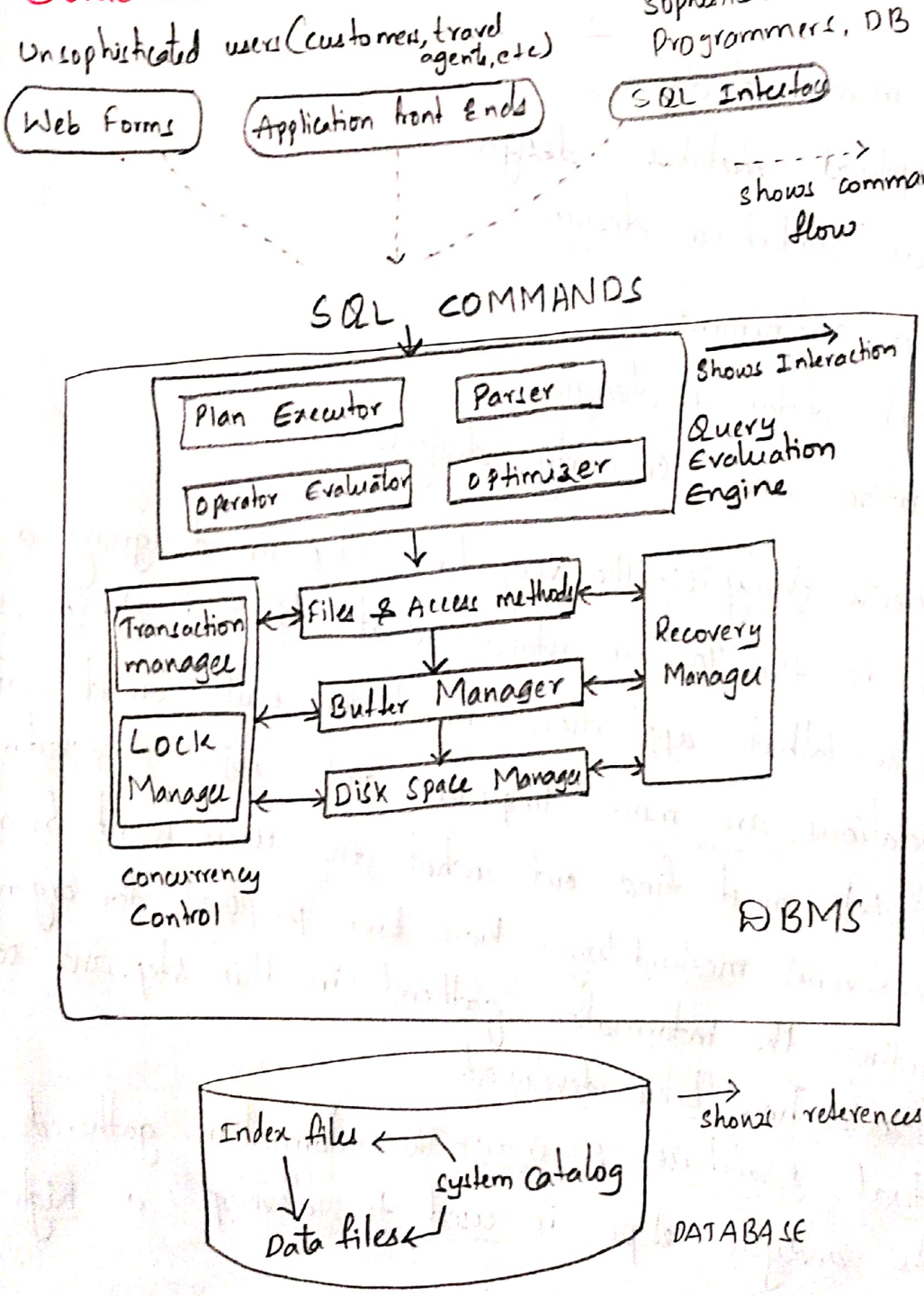
It is the ability to change the conceptual scheme without changing

① External views ② External API or programs.

Importance of Data Independence:

- Helps to improve quality of data.
- Database system maintainance becomes affordable.
- Enforcement of standards and improvement in data base Security.

Structure of DBMS:-



Database Design:-

Database design is the organization of data according to a database model. Properly designed databases are easy to maintain, improves data consistency. The main objective of database design is to produce logical and physical designs. The database design process can be divided into 6 steps. They are:

1. Requirement Analysis
2. Conceptual database design
3. Logical database design
4. Schema refinement
5. Physical database design
6. Application and security design.

1. Requirements Analysis:- The very first step in designing a database application is this. To understand what data is to be stored in database. What applications must be built on it and what operations are most frequent and subject to performance requirements. We must find out what the users want from the data base. Several methodologies have been proposed for organizing and presenting the information gathered in this step, and some automated tools have been developed.

2. Conceptual Database Design:- The information gathered in the requirements analysis steps is used to develop a high-level

• Parser:
It checks the syntax of statement by breaking it into basic units. It ensures that each statement consists of proper component syntax of user query is written here.

• Query optimizer:
When a user issues a query the parsed query is presented to query optimizer which uses information about how data is stored to produce an efficient execution plan for evaluating a query.

• Plan executor:
It is a blueprint for evaluating a query usually represented as tree of relational operators.

• Query evaluation engine:
It is important part of SQL (Structured Query Lang) because all query evaluated in SQL with help of query evaluation engine it executes low level instruction and provides specific output.

• Transaction Manager:
DBMS supports concurrency and crash recovery by carefully scheduling user requests and maintaining all logs of all changes to database. DBMS components associated with concurrency control and recovery includes this.

• Lock Manager:
It ensures that transactions request and release locks according to a suitable locking protocol and keeps the track of requests for locks.

• Recovery Manager:-

It is responsible for maintaining a log and restoring the system to a consistent state after a crash.

The disk space manager, buffer manager and file and access layers must be interacted with these components.

• Disk Space Manager:-

The lowest layer of DBMS software deals with management of space on disk, where data is stored. Higher layers allocate, deallocate, read and write pages through this.

• Concurrency Control:

It is the process of managing simultaneous execution of transactions in a shared database, to ensure the serializability of transactions.

* Purpose:-

To preserve database consistency

To resolve read-write and write-write conflicts.

description of data to be stored in database, along with the constraints known to hold over this data. This step is often carried out using ER model. The ER model is one of the several high level. The goal is to create a simple description of data that closely matches how users and developers think of data.

3. Logical Database Design:-

We must choose DBMS to implement our database design and convert conceptual database design into database schema in the data model of chosen DBMS. We will consider only relational DBMSs, the task in logical design step is to convert ER schema into relational database schema.

4. Schema Refinement:- This step in database design is to analyze the collection of relations in our relational databases schema to identify potential problems and to refine it. Schema refinement can be guided by some elegant and powerful theory.

5. Physical Database Design:- We consider typical expected workloads that our database must support and further refine the database design to ensure that it meets desired performance criteria. This step involves building indexes on some tables and clustering some tables

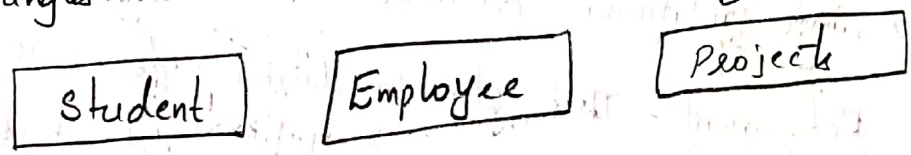
6. Application and Security design:- Any software project that involves a DBMS must consider aspects of application. Design methodologies like UML try to address complete s/w design and development cycle.

We must identify the entities (eg:- users, user groups, DBA) and processes involved in the application. We must describe role of each entity in every process that is related.

ER Diagram:-

Entity Relation diagram is a pictorial representation of data that describes how data is communicated and related to each other. Any object such as entities, object, attributes of an entity, sets of relationships and other attributes of relationship can be characterized with the help of the ER diagram.

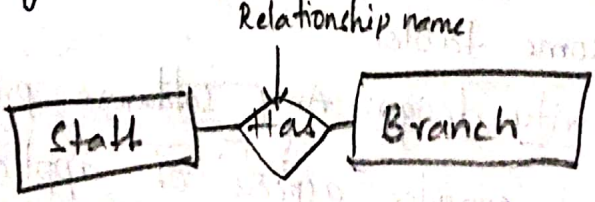
Entity:- They are represented using rectangle shape box. These rectangles are named with the entity set they represent.



Entities

ER modeling is an important technique for any database designer

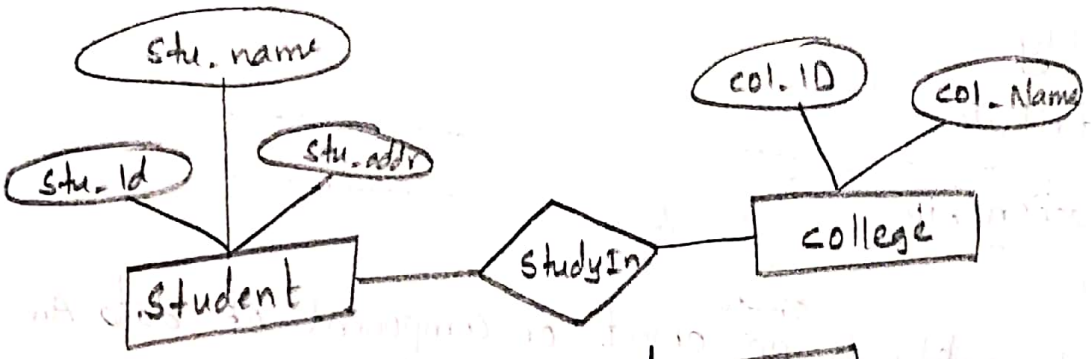
Relationship Type:- A relation ship type is a set of associations between one or more participating entity types. Each relationship type is given a name.



We must identify the entities (eg:- users, user groups, departments) and processes involved in the application. We must describe role of each entity in every process that is reflected.

Simple ER Diagram:-

In the below diagram we have two entities student and college and their relationship. The relationship between student and college is many to one as a college can have ~~multiple~~ many students however a student cannot study in multiple colleges at same time. student Entity has attributes, such as stu_id, stu_name & stu_addr and college entity has attributes such as col_ID & col_Name



Rectangle — Represents entity sets

Ellipses — Attributes

Diamonds — Relationship set

Lines — link attributes to entity sets & entity sets to Relationship set.

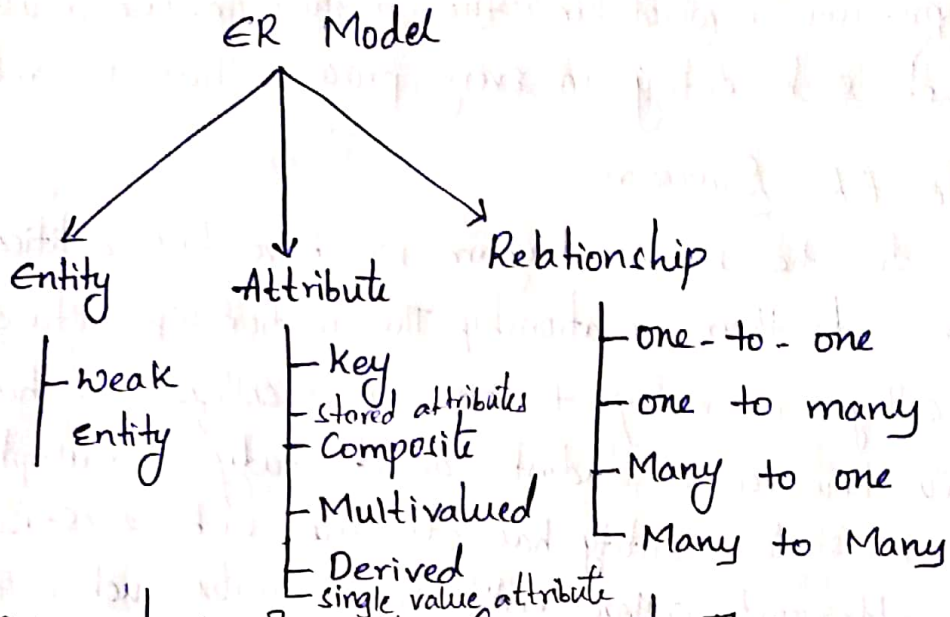
Double Ellipses — Multivalued attributes

Dashed Ellipses — Derived attributes

Double Rectangles — weak entity sets

Double lines — Total participation of Entity in relationship.

Components of ER Diagram.



ER diagram has 3 main components. They are

1. Entity
2. Attribute
3. Relationship

* **Entity:-**

An entity is an object or component of data. An entity is represented as rectangle in an ER diagram.

eg:- We have 2 entities student and college and these two entities have many to one relationship as many students study in a single college.

Weak Entity:-

An entity that cannot be uniquely identified by its own attributes and relies on the relationship with other entity is called weak entity.

Attribute:-

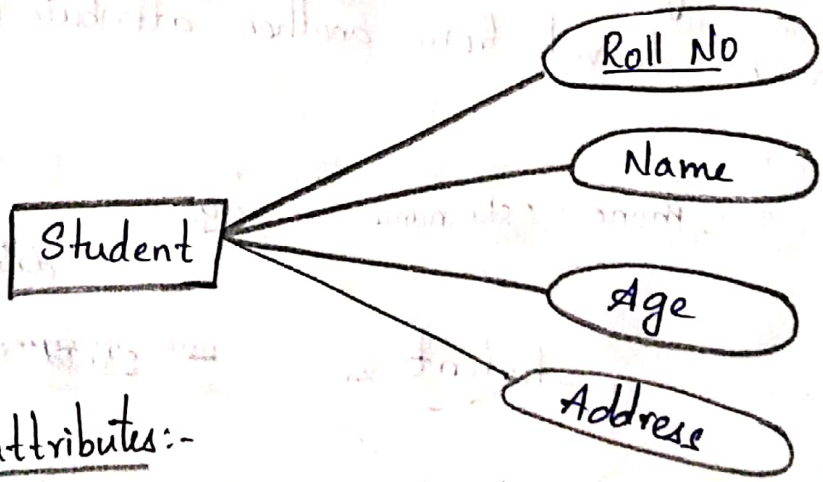
An attribute describes the property of an entity. An attribute is represented as oval in ER diagram. There are four types of attributes.

1. Key attribute
2. Composite attribute
3. Multivalued attribute
4. Derived attribute

- Simple attribute: Attributes that can't be further divided. *atomic value fixed size
  Eg:- PIN code
- single value attribute: It contains single value.
  Eg:- social security number.
  → DOB

1. Key attribute:-

A key attribute can uniquely identify an entity from entity set. For example, student roll number can uniquely identify a student from a set of students. Key attribute is represented by oval same as other attributes however the text of key attribute is underlined.

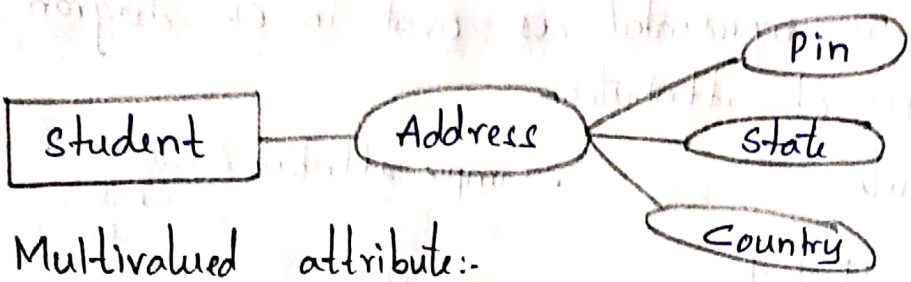


Composite attributes:-

An attribute that is combination of other attributes is known as composite attributes.

Eg:- In student entity, the student address is composite attribute as address is composed of other attributes such as pincode, state, country

Address is a Composite attribute.



3. Multivalued attribute:-

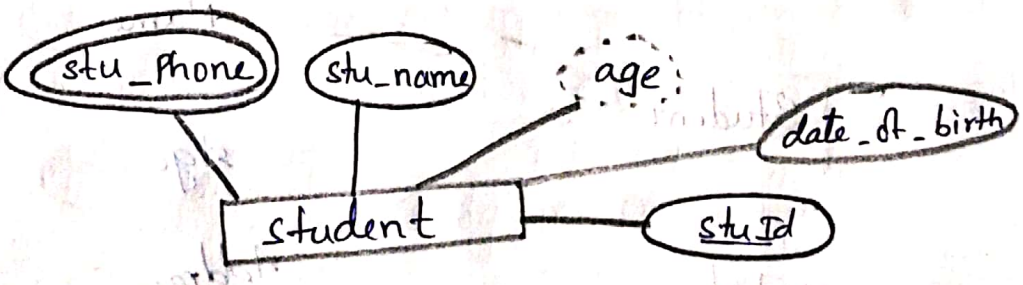
An attribute that can hold multiple values is known as multivalued attribute. It is represented with double ovals in ER diagram.

Eg:- A person can have more than one phone numbers so the phone number attribute is multivalued.

4. Derived attributes:

A derived attribute is one whose value is dynamic and derived from another attribute. It is represented by dashed oval in an ER diagram.

Eg:- Person age is derived attribute as it changes over time and can be derived from another attribute (Date of birth).



* Relationship:-

A relationship is represented by diamond shape in ER diagram. it shows the relationship among entities. There are four types of relationships.

Cardinality of a Relationship

1. One to one Relationship:-

When a single instance of an entity is associated with single instance of another entity then it is called one-to-one relationship.

Eg:- A person has only one passport and passport is given to one person.

2. One to many Relationship:-

When a single instance of an entity is associated with more than one instances of other entity then it is called one to many relationship.

Eg:- customer can place many orders (but a order cannot be placed by many customers.)

3. Many to one Relationship:-

When more than one instances of an entity is associated with single instance of another entity then it is called many to one relationship.

Eg:- Many students can study in one college.

Many to Many relationship:-

When more than one instances of an entity is associated with more than one instances of another entity then it is called many to many relationship.

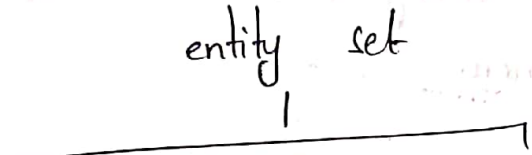
Eg:- students can be assinged to many projects and project can be assinged to many students.

Entity set:-

It is a set of entities of same type.

Eg:- All persons having an account at bank.

Entity set may be of two types.



① represented as □

② key attribute

③ No Total Participation

No key attribute

Total Participation.

Eg:-

If he is an Employee then may not have family.

Total family may have Employee

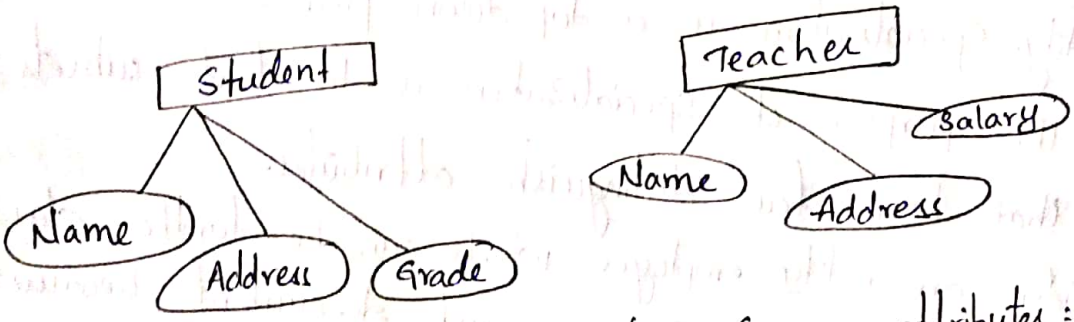
Additional features of ER Model:

- Specialization
- Generalization
- Higher level and lower level entity set
- Attribute Inheritance
- Aggregation

Generalization:-

It is a process in which the common attributes of more than one entities form a new entity. This newly formed entity is called generalized entity.

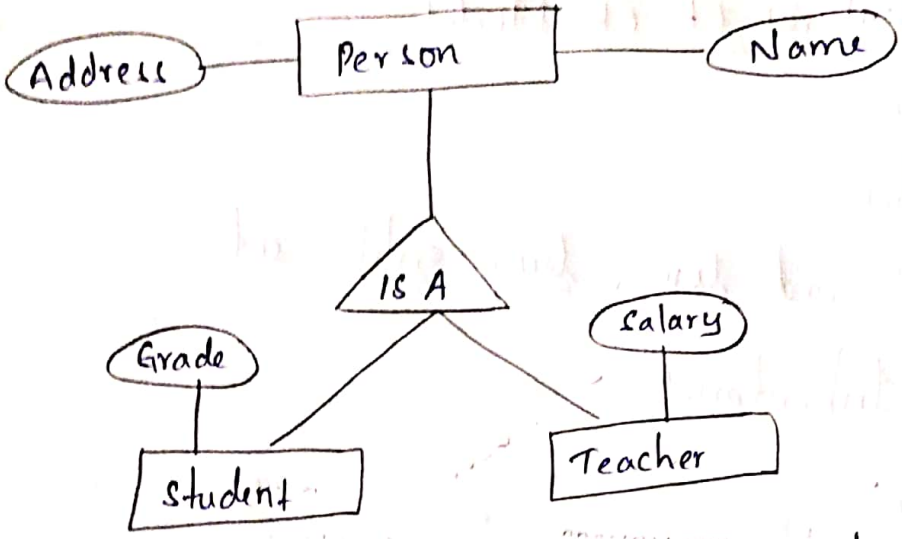
eg:- We have two entities student and Teacher.



The above two entities have Common attributes: name & address. We can make generalized entity with these Common attributes.

Lets have ER model after generalization.

student and teacher only has the specialized attributes Grades & Salary respectively and their Common attributes (name & address) are now associated with new entity person which is in relationship with both entities.



→ Generalization uses bottom-up approach where two or more level entities combine together to form a higher level new entity.

Specialization:-

It is a process in which an entity is divided into sub-entities. It is a reverse process of generalization, in generalization two entities combine together to form a new higher level entity. Specialization is a top-down process.

The purpose of Specialization is to find subsets of entities that have few distinguish attributes.

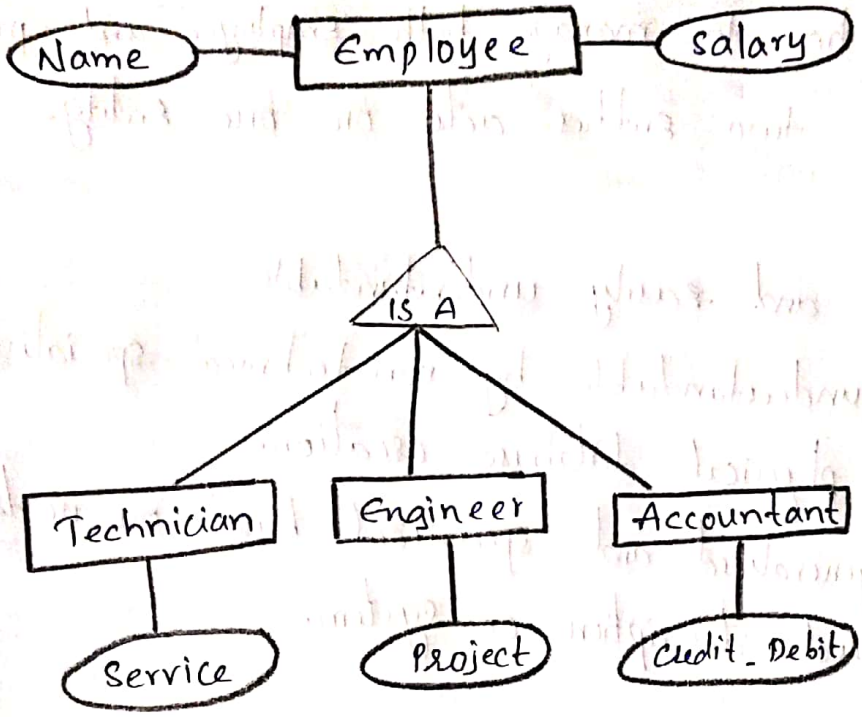
Eg:- Consider an entity employee which can be further classified as sub-entities Technician, Engineer and Accountant because those subentities have some distinguish attributes.

Higher level entity → employee

sub entities → Technician, Engineer and Accountant

Technician handles service, Engineer works on project

Accountant handles credit & debit details.

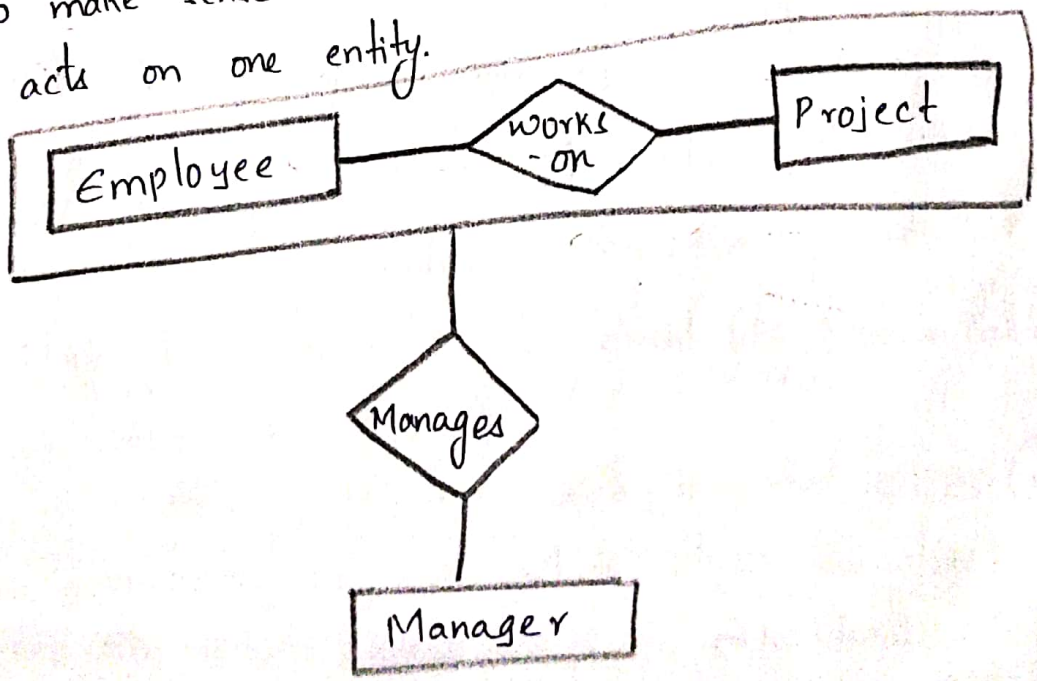


All those three employees have common attributes such as name and salary.

Aggregation:-

It is a process in which a single entity alone is not able to make sense in relationship so the relationship of two entity acts on one entity.

eg:-



Manager has to manage both employee and project. Relationship of two entities acts on one entity.

Advantages:-

- It is simple and easily understandable.
- It can be understandable by non-technical specialist.
- It helps in physical database creation.
- It can be generalized and specialized based on needs.
- Gives higher level description of system.

* DBMS Architecture:-

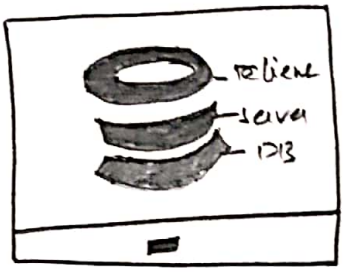
DBMS architecture will help us to understand the components of database system and relation among them.

Types of DBMS architecture:-

1. Single tier architecture
2. Two tier architecture
3. Three tier architecture

1. Single tier architecture:-

In this type of architecture client, server and Database all reside on same machine. Anytime you install a DB in your system and access it to practice SQL querries. It is used rarely in production.

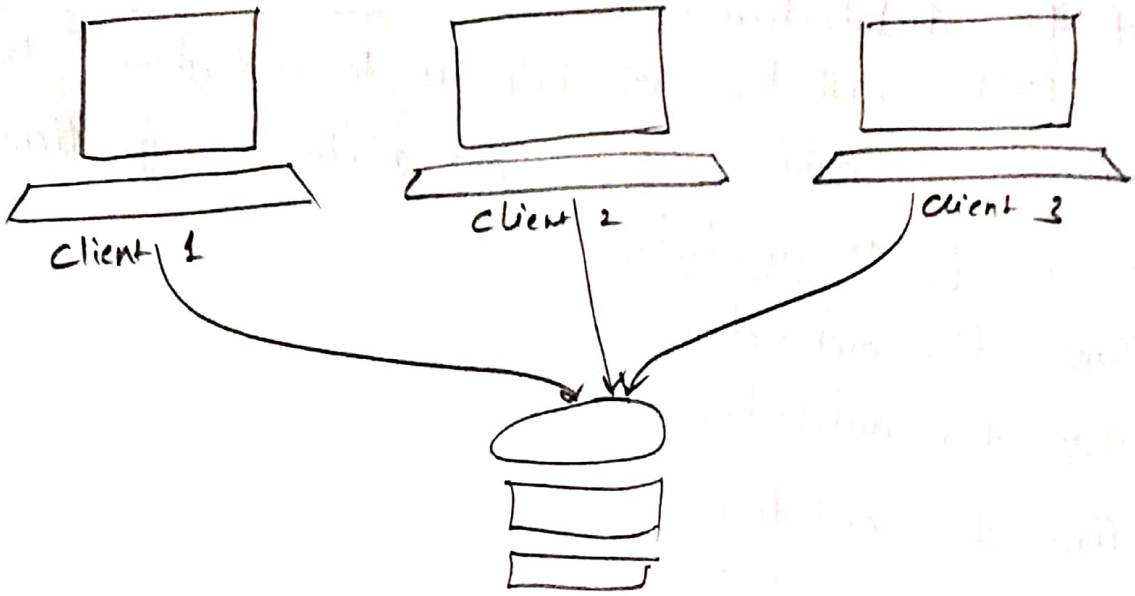


2. Two tier Architecture:-

It is a database architecture where

1. Presentation layer runs on a client (PC, Mobile, Tablet, etc)
2. Data is stored on server.

An application interface which is called ODBC (Open Database Connectivity) an API which allows the client side program to call DBMS. Today most of the DBMS offers ODBC drivers for their DBMS. 2 tier architecture provides added security to DBMS as it is not exposed to end user directly.

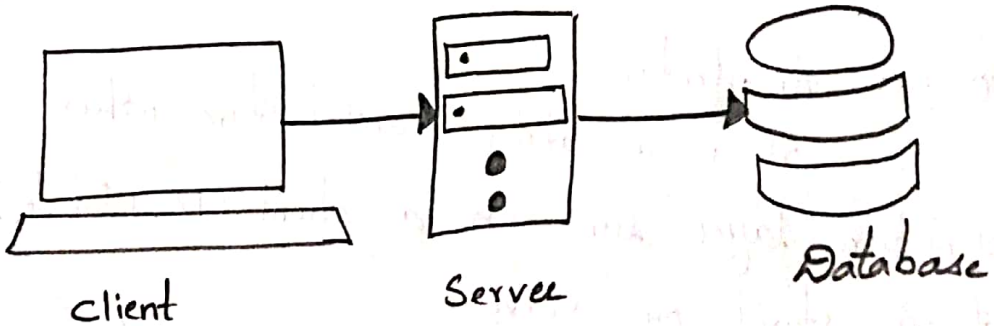


In the above server is connected with clients 1, 2m 3. The architecture provides Direct and faster communication.

Three-tier Architecture:-

It is an extension of two-tier architecture.

1) Presentation layer
2) Application layer
3) Database server.



This DBMS architecture contains an application layer between the user and DBMS, which is responsible for Communicati the users request to DBMS system and send the responce from DBMS to user.

*Three tier architecture is most popular DBMS architecture.

The goal of three tier achitecture is:

- To seperate the user applications and physical database.
- Proposed to support DBMS characteristics.
- Program - Data independence
- Support multiple views of data.

What is a file system?

File system is the traditional way to keep your data organised in way which is easy for physical access.

eg:- Drives.

What is schema and instance?

The overall design/description of databae is known as schema.

eg:- employee schema, student schema.

Data stored at a particular moment is called instance.

→ schema can be divided into two

(i) Physical schema - describing physical level

(ii) Conceptual schema - " conceptual level.

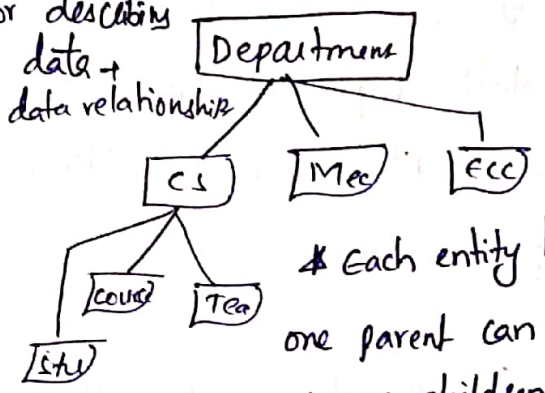
Data models:- It defines the structure of database.

Hierarchical Data Model:- comprises the collection of conceptual tools for describing data → data relationship

* It looks. is a tree like structure

→ relation is Parent & child.

↓ Dept. ↓ Course.



* Each entity has only one parent can have many children.

→ Main parent is root.

* Data is represented in records as trees → collection of records.

Advantages:

1. Parent - child relation
2. Security is highly provided.
3. one to Many relationship.
4. Data sharing is supported.

Disadvantage

1. DDL, DML language ✗
2. Not flexible.
3. Physical data knowledge is needed.
4. If one is changed it reflects.

Object oriented Database Models.

- It works on OOP's
- Represent data → objects & classes.
- It is also known as Hybrid application.

OOPS: Inheritance, Polymorphism, encapsulation

DBMS: Integrity, Concurrency, Querry processing

→ Data + relationship are organized in single structure known as object.

→ An object is the abstraction of real world entity.

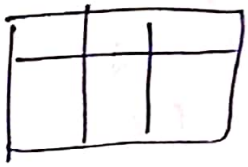
Person — object/class

Name, age, DOB → Properties / attributes

Relational Data model:-

It uses a collection of tables to represent both data and relationships among those data.

→ each table has multiple columns and each column has unique name



→ schema vs Instance → Particular moment

↳ OOM ↳ overall design/structure of relation

class:- objects that are similar to characteristics are grouped

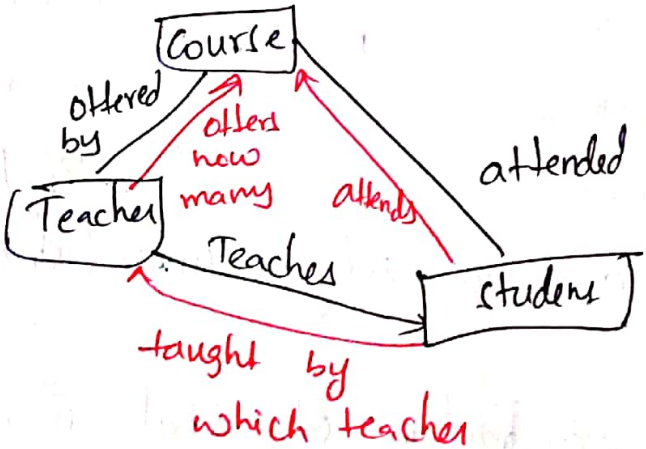
ER model Categories
- object based logical model (Data is treated as real world objects
- Record " " " (Data as treated as records or tuples.

OOM.
↓
C++, Java – Program.
class → data is stored.
↓function
it is operated

- Hierarchical
- N/W
- relational model.

N/W:-

Data is records in Graphs.



Relation:- Many to Many.

→ we use links.