DBMS

What is DBMS? (DataBase Management System)

A Database management system is a collection of programs that enable the users and maintain the database.

The DBMS is a general purpose of system that facilatates the purpose of definning, constructing & manipulating databases for various application.

Database Application

① IRCTC ② Flipkart ③ Amazon ④ SBI

Most Recent DB Appln

i) Data warehousing
ii) Data Minning
iii) GIS
iv) Big Data Analysis
v) Distributed database

DBMS Software

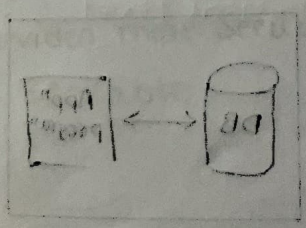
① ORACLE → oracle
② SQL → microsoft
③ mongoDB
④ mySQL → mySQL
5) Supabase

Keyfeatures of database

1) Data storage & Retrieval
2) Concurrency Control
3) Databackup & Recovery
4) Data Integrity & Security

Advantage of DBMS

i) Data Integrity
ii) Concurrent Accers
iii) Data Security
iv) Baekup and recovery
v) Data Sharing.



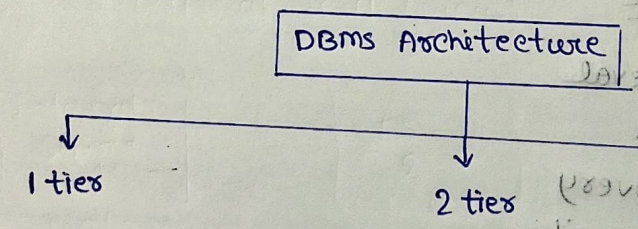
Disadvantage

1) Complexity :- Complex to set up and maintain.
2) performance overhead :- Can add the overshead to the performance of an application.
3) cost :- Cost is high for maintaining and upgrading
4) Limited usage Cause.

	File Processing	DBMS
Defn	File Systems is a way of processing files in a storage	It is software for managing the database.
Data Redundancy	Redundant data can be present	Less Redundancy
Ease of Access data	Accessing file is not easy	Easy to Access
Concurrency Control	No concurrency Control System	Concurrency Control is maintained
Cost	Less Expensive	Higher cost
Data Independance	No data Independance	Data Independance Exist
Example	C++	ORACLE, SQL server.

DBMS Architecture

DBMS Architecture depends upon how users are connected to the database to get their request done.

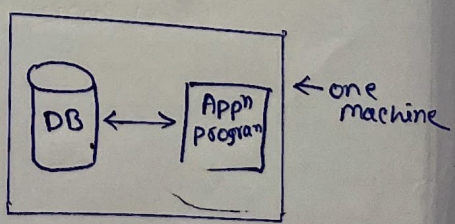


1 tier Architecture

- In 1 tier Architecture, the database is directly available to user.
- The client & Server are present in the same machine
- used to learn SQL or project in one machine.

Advantage

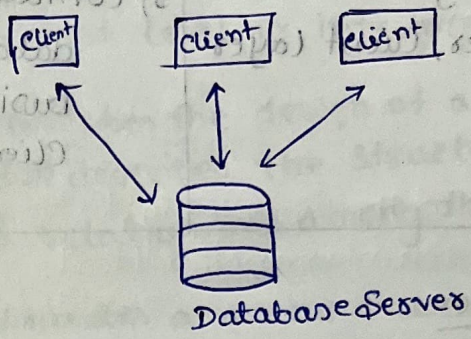
i. Simple to set up and maintain
ii. Cost Effective.
iii. No additional hardware is required.



2 tier Architecture :-

In 2 tier Architecture, applications on the client end directly communicate with the Server side application with help of JDBC ODBC API.

- The user interface and application programs ~~are~~ are run in the client side
- The Server side is responsible to provide the functionality query and transaction processing.



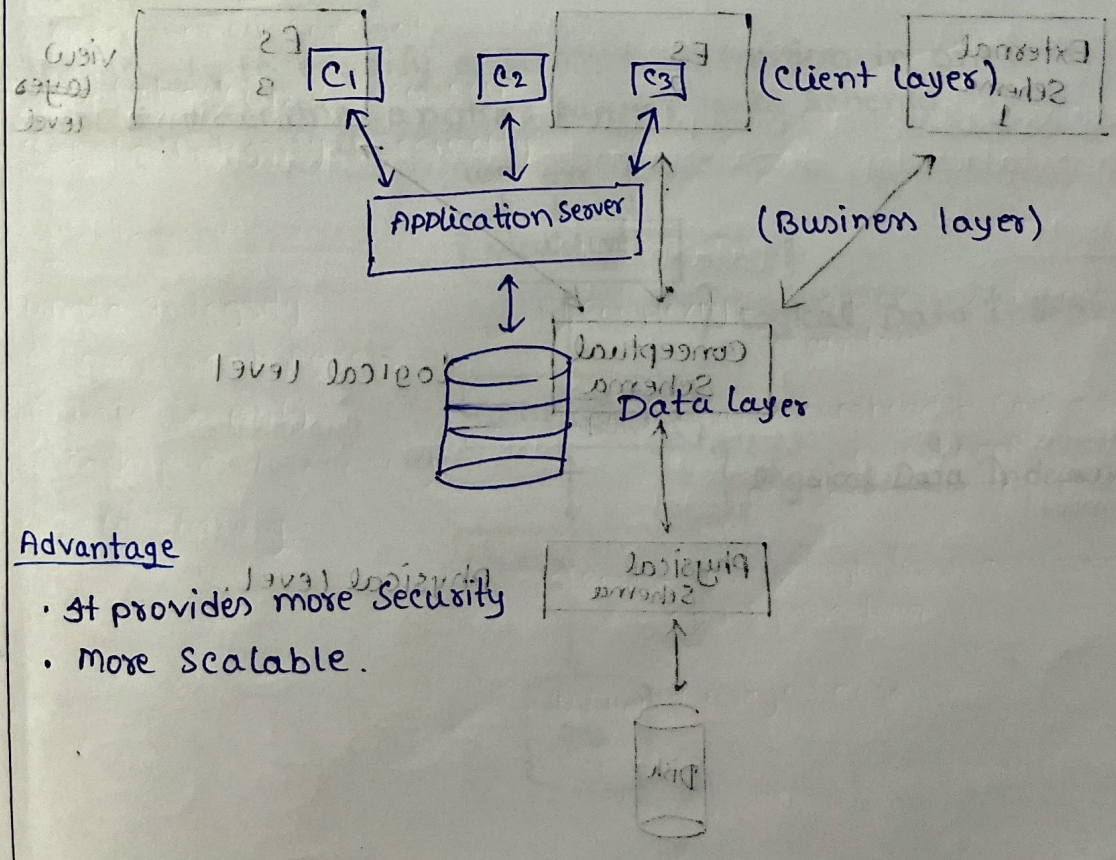
Advantage

i) Easy to access
ii) low cost than 3 tier.

3 tier Architecture :-

In this Architecture, client can't directly communicate with the Server.

- The application on the client end interacts with Application Server to connect with the database System.
- It provides GUI which make ~~the~~ the System more accessible.



Advantage

- It provides more Security
- More Scalable.

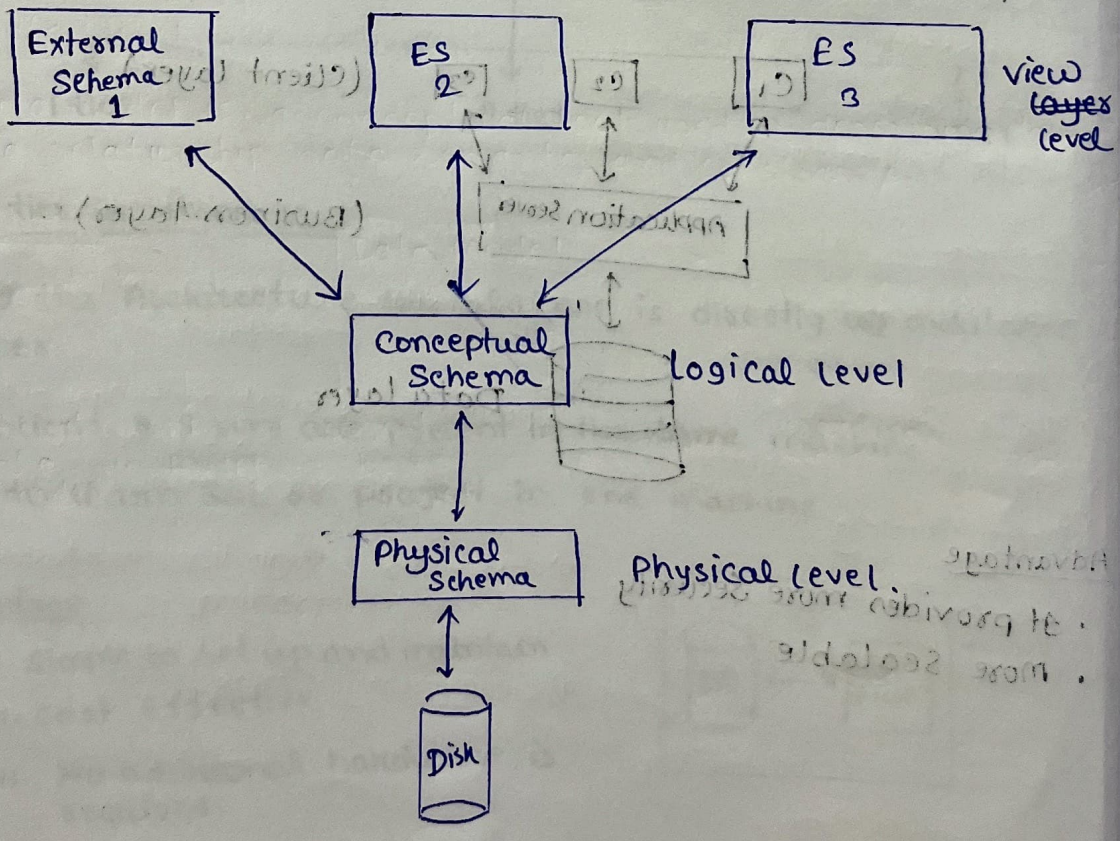
2 tier	3 tier
1) client server application	1. web based application
2) Easy to build and maintain	2. Complex to build and maintain
3) It runs slower	3) It runs faster
4) client can directly communicate with server	4) Client can not directly Communicate with Server
5) Contain 2 layer – Data layer, Client layer	5) contain 3 layer – data layer, Business layer, Client layers.

Data Abstraction

Data Abstraction refers to the process hiding unnecessary data from the end users.

It has three levels.

1) Internal level (Physical)
2) Conceptual level
3) External level (view)



1) Physical or Internal level

a) lowest level of abstraction for DBMS

The internal level has an internal schema which describes the physical storage structure of the database.

It uses physical data model that is used to define that how the data will be stored.

2) Logical level or Conceptual level :-

Logical level is the next highest level or intermediate level.

The conceptual schema describes the design of a database at conceptual level. It also describes the structure of the whole data base and relationship among the datas.

. Define the tables, attributes and how tables are connected.

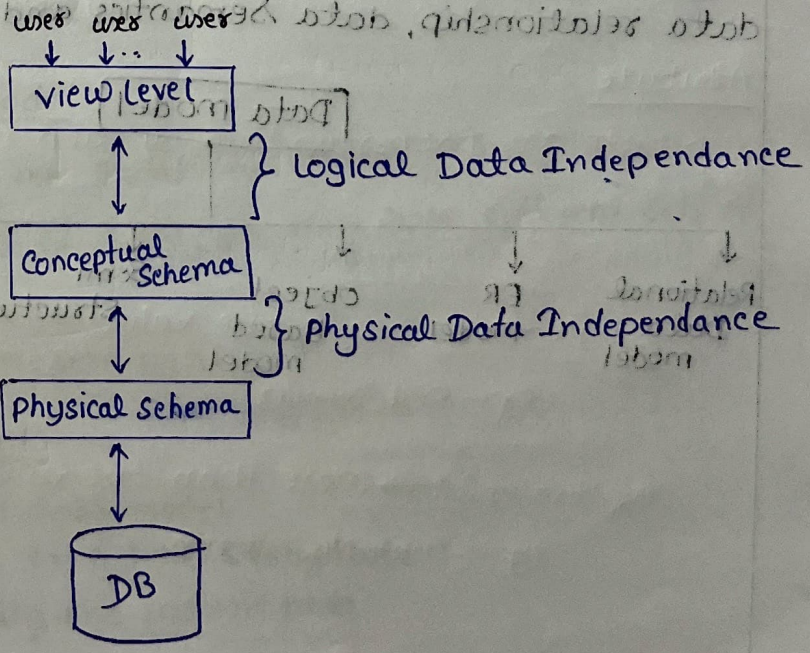
3) view level :-

This is the top level.

At the External level, a database contains several schemas called Subschema. It is used to describe the different view of the database.

Data Independance :-

The ability to modify a schema definition in one level without effecting another higher level schema.



1) Physical Data Independance :-

It refers to the ability to modify the physical schema without affecting the conceptual schema.

11) Logical Data Independance :-

It refers to the ability to modify the conceptual schema without affecting the External schema.

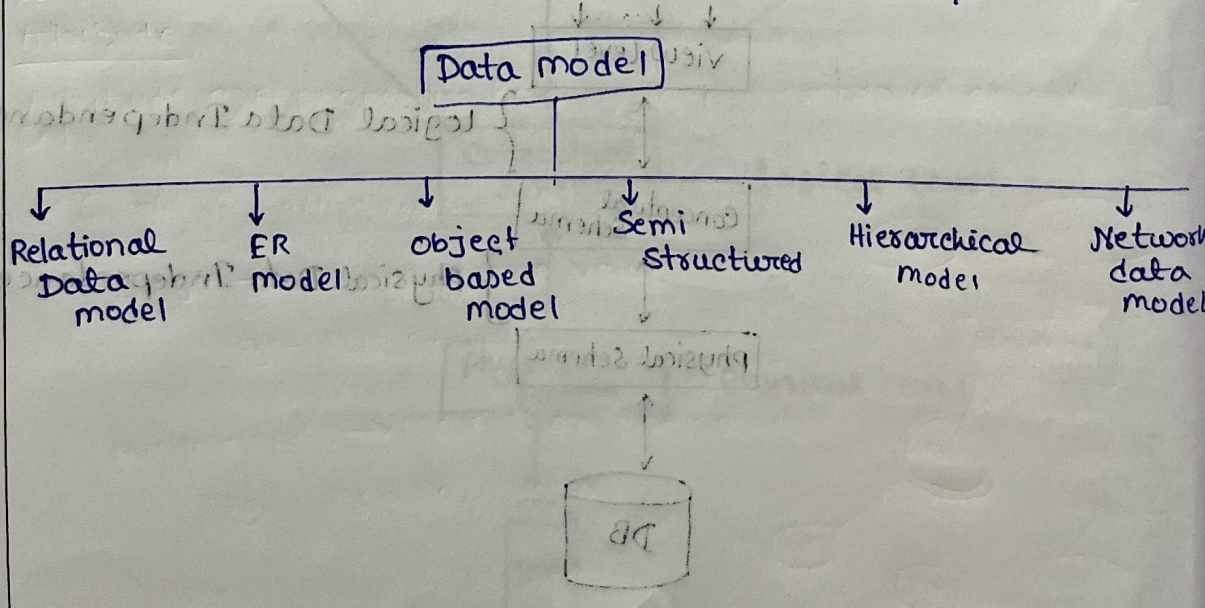
It represent how data is viewed by application programs.

Advantage

1) It reduces the complexity for the user
11) reduces maintenance effort.

☐ Data model

A data model is a conceptual tool used for describing data, data relationship, data semantics and constraints.



1. Relational Model :-

In this model, data are kept in the form of table for representing data and relationships amongs those data.

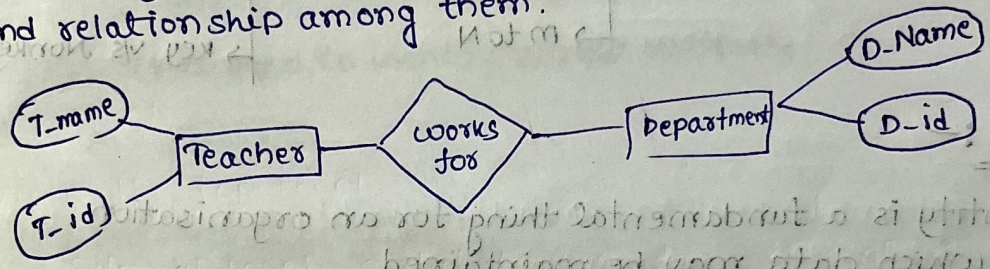
- Each table is group of rows and columns

Table | Student

Student id	Student Name	Age
2570188	Sayan Pal	22
2570190	Shlok Arora	23

2. ER model :-

- ER model is a high level data model.
- It is a blueprint of a database.
- It is a logical representation of the data such as an object and relationship among them.



3. Object - based Data model :-

- It is also known as object oriented data model.
- In object based data model is .exe of ER model with some object identity
- In object oriented data model, data and relationships are present in Single structure.

4. Semi - Structured Data model :-

- It is an evolved from the Relational model.
- Individual data items of same type may have different sets of attribute
- XML is used for representing this model.

5. Hierarchical data model :-

- This is old form of data model
- This model use a tree like structure
- Each entity has only one parent node.

6. Network Model :-

Data is represented using graph and it can have more than one parent node.

It supports many to many relationship.

Entity Relationship Diagram :-

ER diagram can expres the overall logical structure of the database.

Three components are →

- **Entity**
  - → Strong
  - → weak
- **Relationship**
  - → 1 to 1
  - → 1 to m
  - → m to 1
  - → m to N
- **Attribute**
  - → Single v/s Multivalued
  - → Single v/s Composite
  - → Stored v/s Derived
  - → key v/s Nonkey

Entity :-

An Entity is a fundamental thing for an organisation about which data may be maintained.

- □ → Entity
- ◇ → Relationship
- — → Link betn entity and relationship
- ⬭ → Attributes
- ⬭ → multivalued attributes
- ⬭ (dashed) → Derived attributes
- ▣ → weak Entity

Attribute

Attribute are properties hold by each member of an entity set.

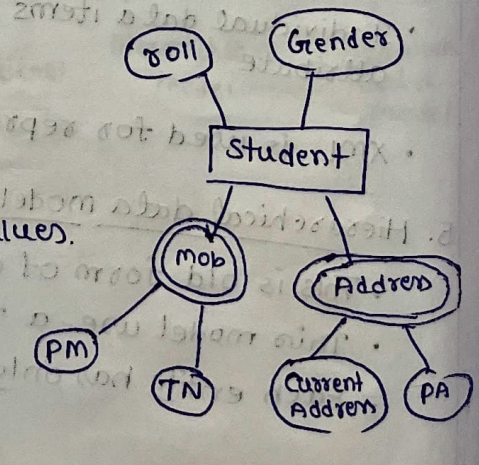
* Single v/s Multivalued :-

Single → It has a one value

Eg :- Roll, Gender

Multivalued :→ more than one values.

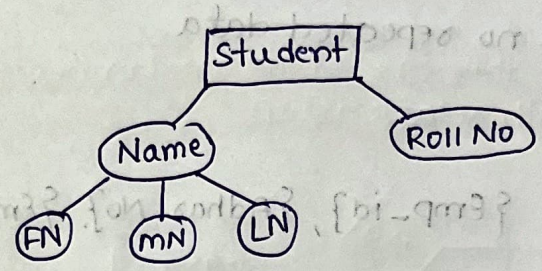
Eg :- mob No, Address



* Simple v/s composite

Simple → can not be divided into subparts

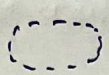
Composite → can be divided into subparts



* Stored v/s Derived :-

Stored → It's value is stored manually in database.

Derived → It's value can be calculated from other attribute



* Key v/s Non key :-

key :- It is used to identify a row uniquely.

Eg :- (Roll No)

Non key → (Age), (DOB)

Types key :-

1) Super key :- A Super key is the combination of all possible attributes that can uniquely identify the rows (tuples).

· Super set of any candidate key is Super key

Super key

1. {Emp id}
2. {Adhar_No}
3. {Email_id}
4. {Emp_id, Adhar_No}
5. {Adhar No, Email_id}
6. {Emp_id, Email_id}
7. {Emp_id, Adhar_No, Email_id}
8. {Emp_id, Name}

Emp_id	Name	Adhar_No	Email_id
01	Sayan	5024	xyz@
02	Samata	5025	ABC@
04	Sreenika	2507	pqr@

2. Candidate key :-

A candidate key is a minimal set of attribute that uniquely identifies each tuple within a table.

It is a super key with no repeated data.

Ex:- In previous table,

Candidate key :→ {Emp_id}, {Adhar_No}, {Email_id}

3. Primary key

It is first key used to identify one and only one instance of an Entity uniquely.

Primary key = {unique + NotNull}

Ex:- {Emp_id} is primary key.

4. Alternate Key :-

An Alternate key can uniquely identify a row but is not selected as the primary key.

Ex:- {Adhar_No} {Email_id}

5. Foreign key :-

A Foreign key is an attribute that is used to link two tables together via the primary key.

Primary key → Foreign key

ID	Name	course
2041	Tom	Java
2204	John	C++
2043	Bob	Python
2032	Alice	Oracle

Student details

ID	Marks
2041	65
2204	55
2043	72
2032	95

Student Marks

6. Composite key / Compound key :-

A key that has more than one attribute is known as Composite key. It is also known as Compound key.

Cust_id	Order_id	Product_code	Product_count
C01	001	P111	5
C02	012	P111	8
C02	012	P222	6
C01	001	P333	9

Ex:- {cust_id, product_code}

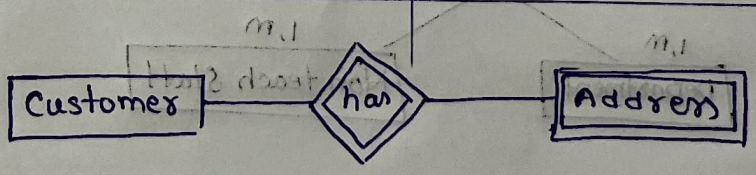
Strong and weak Entity

An Entity set which does not have specific attributes to form a primary key is said to be weak Entity. It depends on other entities in Schema.

▸ An Entity set which has a primary key is called strong Entity.

*** The relationship between weak and strong Entity represented by

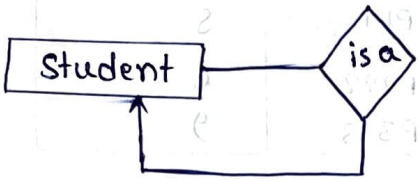
Strong Entity	weak Entity
1. Strong Entity always has a primary key	1. weak Entity has a partial discriminator key.
2. Strong Entity is not dependant on any Entity	2. weak Entity depends on Strong Entity
3. Strong Entity is represented by a single rectangle.	3. weak Entity is represented by double rectangle.



Degree of Relationship

It denotes the number of entity types that participate in a relationship.

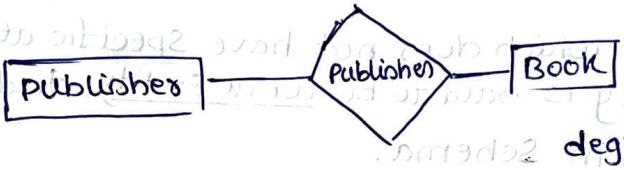
1. **Unary Relationship**:- In this one entity are connected with relationship.



degree = 1

2. **Binary Relationship**:-

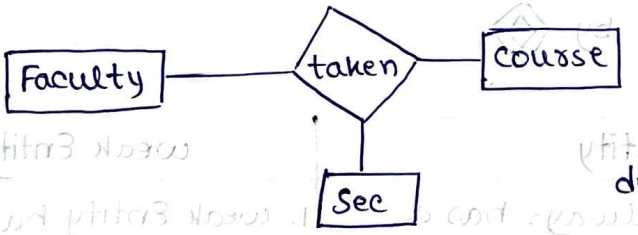
It refers to the relationship between two entities.



degree = 2

3. **Ternary Relationship**

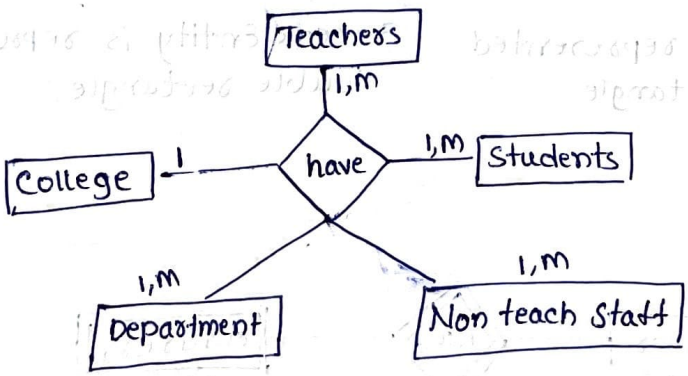
The relationship between three entities



degree = 3

4. **N-ary Relationship**:-

Here 'n' entity sets participates in a relationship.



Here degree = 5