Database System (Dbms):-

Database (collection of related data)
- → Structured (store in particular format)
- → Unstructured

Dbms
- → SQL
- → Oracle

* Database is collection of inter-related data which is used to retrive, insert and delete data effecently.

File System vs Dbms:-

Dbms	File System
collection of structured data.	collection of unstructured data user has
In Dbms, user is not req.	to write the procedure for managing
to write procedure.	database.
no data redundancy (repetition of data).	data - redundancy is there.
searching is easy.	searching is diff.
dbms gives abstract view	It provide details of data representation
of data that hide details.	& storage of data.
→ have crash recovery mechanism (protect data from system failure)	→ no crash recovery mechanism. (if system crash while entering data content of file will lost).
→ good protection mechanism.	very diff to protect file under system
→ wide variety of sophesticated	→ can't effecienlly store &
technique to store & retrive data	retrieve data.
→ take care of concurrent access using some form of locking	→ no this type of mechanism

many user access data at same time

Dbms Architecture:

It depends upon how user are connected to database to get their req done

1 - tier Architecture :-

→ database is readily available on client machine, any req or changes will directly done by a client itself. no tool or any medium is req.

→ It used for deployment of local application, where programmer can directly communicate with database for quick response.

(Ticket booking on station)

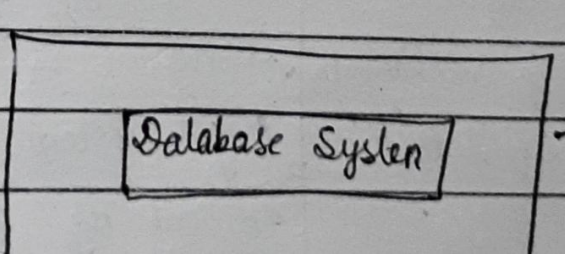
2 - tier :-

→ database system is present at server machine and dbms application is present at client machine. they connected each other through reliable network.

→ whenever a client machine make a req to access database present at server using query lng like. Sql server perform req on database and return result back to client.

→ application Connection interface such as JDBC, ODBC are used for interaction b/w server & client

Advantage - Maintainance



1-tier Architecture :-

→ database is readily available on client machine, any req. or changes will directly done by a client itself. no tool or any medium is req.

→ It used for deployment of local application, where ploglammer can directly communicate with database for quick response.

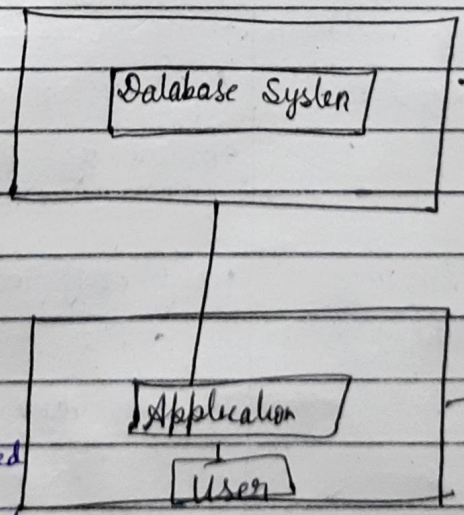
(ticket booking on station)

2-tier :-

→ database system is present at server machine and dbms application is present at client machine. they connected each other through reliable network.

→ whenever a client machine make a req. to access database present at server using query lng like. SQL server perform req on database and return result back to client.

→ application connection interface such as JDBC, ODBC are used for interaction b/w server & client



dvantage - Maintainance

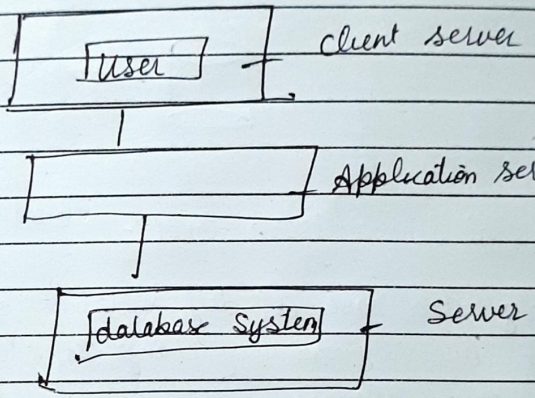
sadvantage - Scalability
Security

mited clients or for authorized
es when clients are increased
en it gives load on server.

(Ticket booking online)

3- tier:

Client Application doesn't communicate directly with database system present at server machine rather than client application communicate with server application or application server and application server directly communicate with database present at server.



Advantage: Scalability
Security

Disadvantage - maintenance
maintainance

Schema :- logical representation of data / no. of table or collection of table. / design of database.

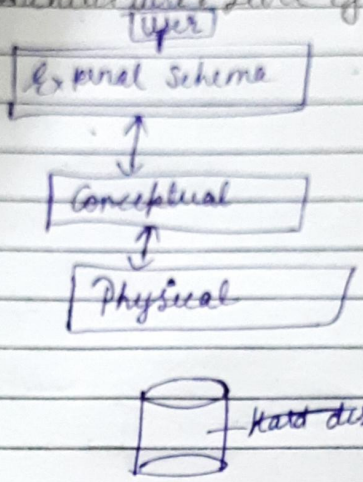
→ Physical ⇒ how the data stored in blocks of storage

→ Logical ⇒ data can be defined as certain type of data records stored in data-structure, however internal details such as implementation is hidden

→ view ⇒ user interaction with database system.

irrelevant
hide the data from the user or to hide the
→ position where data is stored

3- Schema Architecture's Level of Data Abstraction



View Level

Logical Level

Physical Level

External Schema/ View Level:- view of database, how data should be look to user and every user have a diff view pt of database some have a diff authority so they change & edit the data whereas some has only view it.

(E-R model) Conceptual schema :- representation of stored data

Physical Schema:- decide the location of data where data will be stored.

Key:- It is a attribute. key can be uniquely identify two tuple. it means it define the diff b/w two data or which except key other attributes are same or may not be. key is key is only unique attribute that define that person.

Roll no.	Name	City	Age
1.	Ram	Kanpur	20
2.	Ram	"	20
3.	Ram	"	20

all have same values except roll no.

Candidate key :-

★ Collection of all unique key is known as candidate key and from this we select any key as primary key & rest as alternative key.

for Student table :- Aadhar card, voter_id, license no., roll no, phone no. } all are unique for all student so these are candidate key subset.

Primary key :- { unique + NOT NULL } select that candidate key that is correctly entered by the user or data.

from candidate key

★ In database, there is only one primary_key.

Foreign key :- It is an attribute or set of attribute that reference primary key of same table or another table (relation)

★ → maintain referential integrity

Course :- (Referencing table)

Student (Reference table)

Primary key Roll no.	name	Address
1	A	Delhi
2	B	Kanpur
3	C	U.P

Course id	course_name	Foreign key roll no.
C_1	Dbms	1
C_2	Network	2

roll no is a foreign key in course_table because roll no is taken reference from student table. if data is not present in student table than it doesn't enter in course table.

Create table Course (
course_id varchar(10);

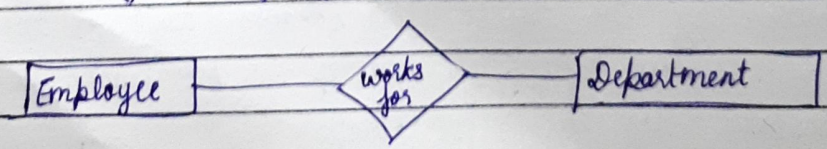
Reference table (base table)	Referencing table
⇒ ★ Insert - no problem	⇒★ Insert - problem may occour (if that roll no is not present in student table)
⇒ ★ deletion - may problem occour (if data present delete of roll no 1 in student table but can't delete in course table then problem occour)	★ deletion → no problem occour
solution:	
★ on delete Cascade (delete data from the both table)	★ updation → may problem occour
★ on delete set NULL (delete data from student table & put NULL on course table)	
★ on delete No Action (no deletion occour)	
⇒ ★ updation - may problem occour ↳ same as delete solution	

Super key :- Super key is a combination of all possible attributes which can uniquely identify 2 tuples in a table.

★ Super key is a collection of candidate key and other attributes.

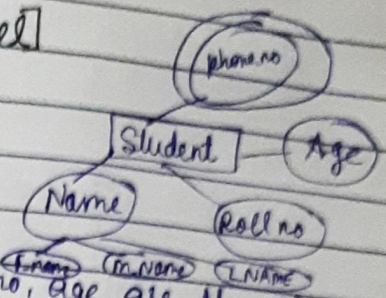
Tuple - Single row of table.

Entity :- It may be any object, class, person represented by rectangle.

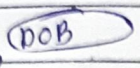


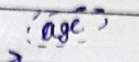
ER model [Entity Relationship Model]

Types of Attributes :-



Student is a entity and Name, roll no, age are the attributes of student entity.





Stored vs Derived :- for student birth year is stored attribute while age is derived. age can be derived from birth year.

Key vs Non-key :- may be unique or not.

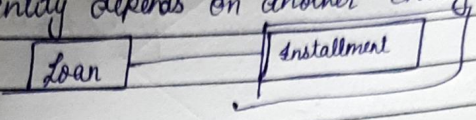
unique attribute (primary key)

Composite :- attribute that composed of many other attribute. Name is an composite attribute it represented by ellipse, and those ellipse connects with other.

Multivalued :- attributes have more than one value. it is represented by double oval.

eg :- phone no.

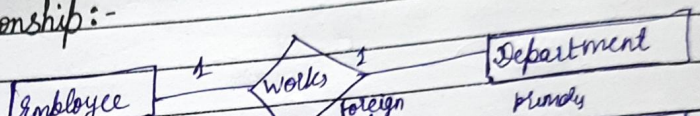
Weak entity:- entity depends on another entity.



★ Attribute is property of entity.

Types of Relationship:-

One to one:-



E_id	E_name	Age
1	A	20
2	B	25

Works (Foreign key)

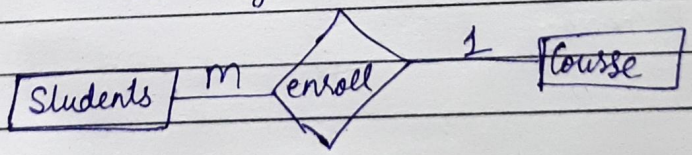
E_ID	D_id
1	D1
2	D2

Primary

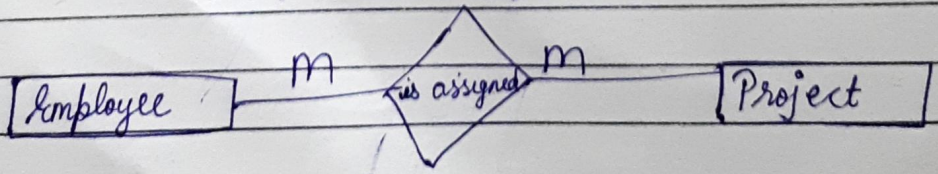
D_id	D_name	D_loc
D_1	TT	Kanpur
D_2	HC	Agra

One to many:- One instance of entity on left and more than one instance of entity on right

Many-to one:- many instance on left side and one instance on right side

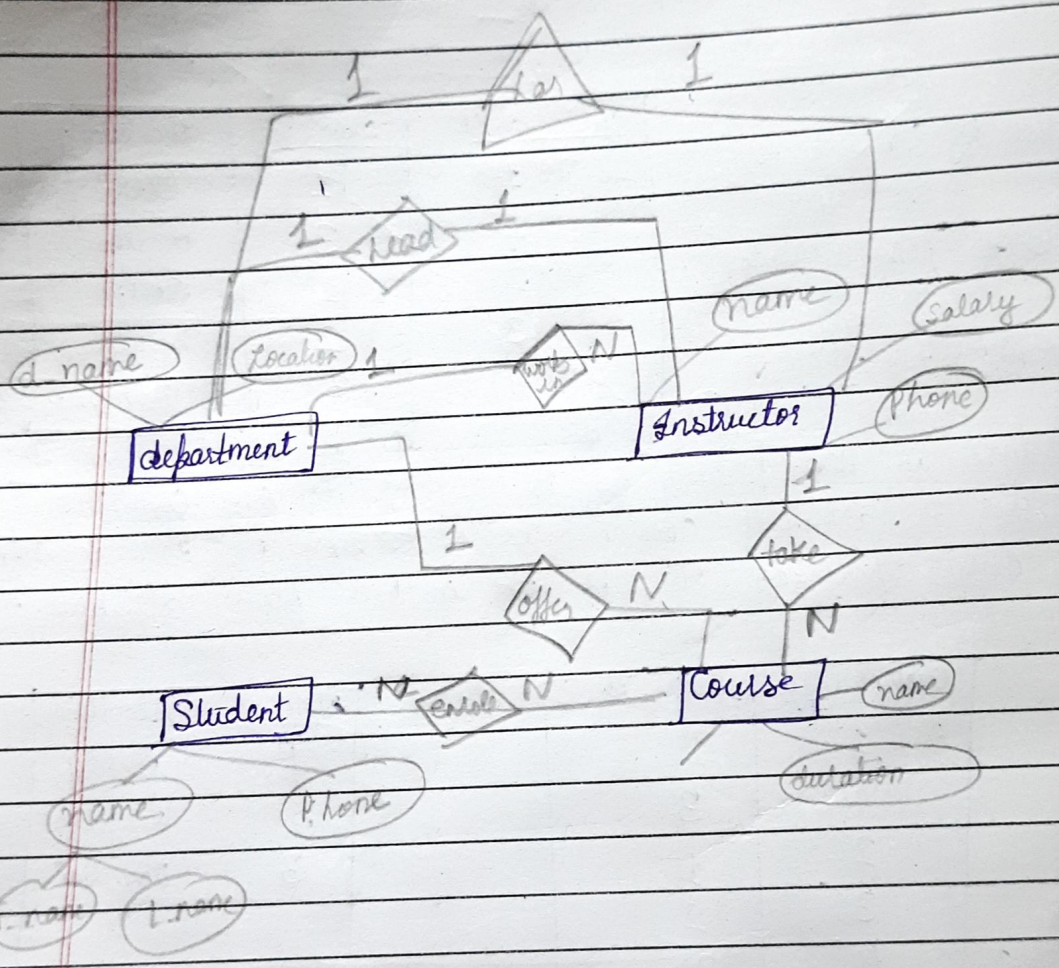


Many to Many:- more than one instance on left and more than one instance on right side.



R (A B C D E F)

F.D { C → F, E → A, }

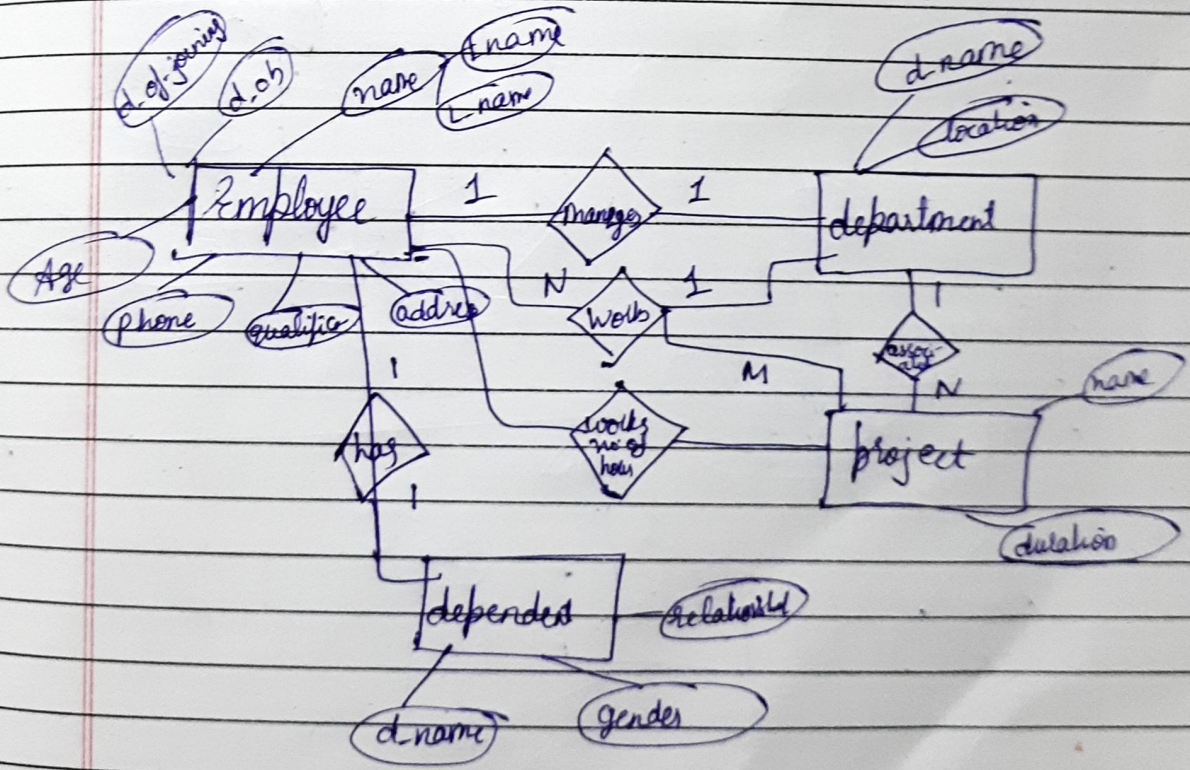


college contains many departments.
each department can offer no. of courses.
many instructor can work in department.
instructor can work only in one department.
for each department there is head.
instructor can be head of only one department.
each instructor can take any no. of courses.
course can be taken by only one instructor.

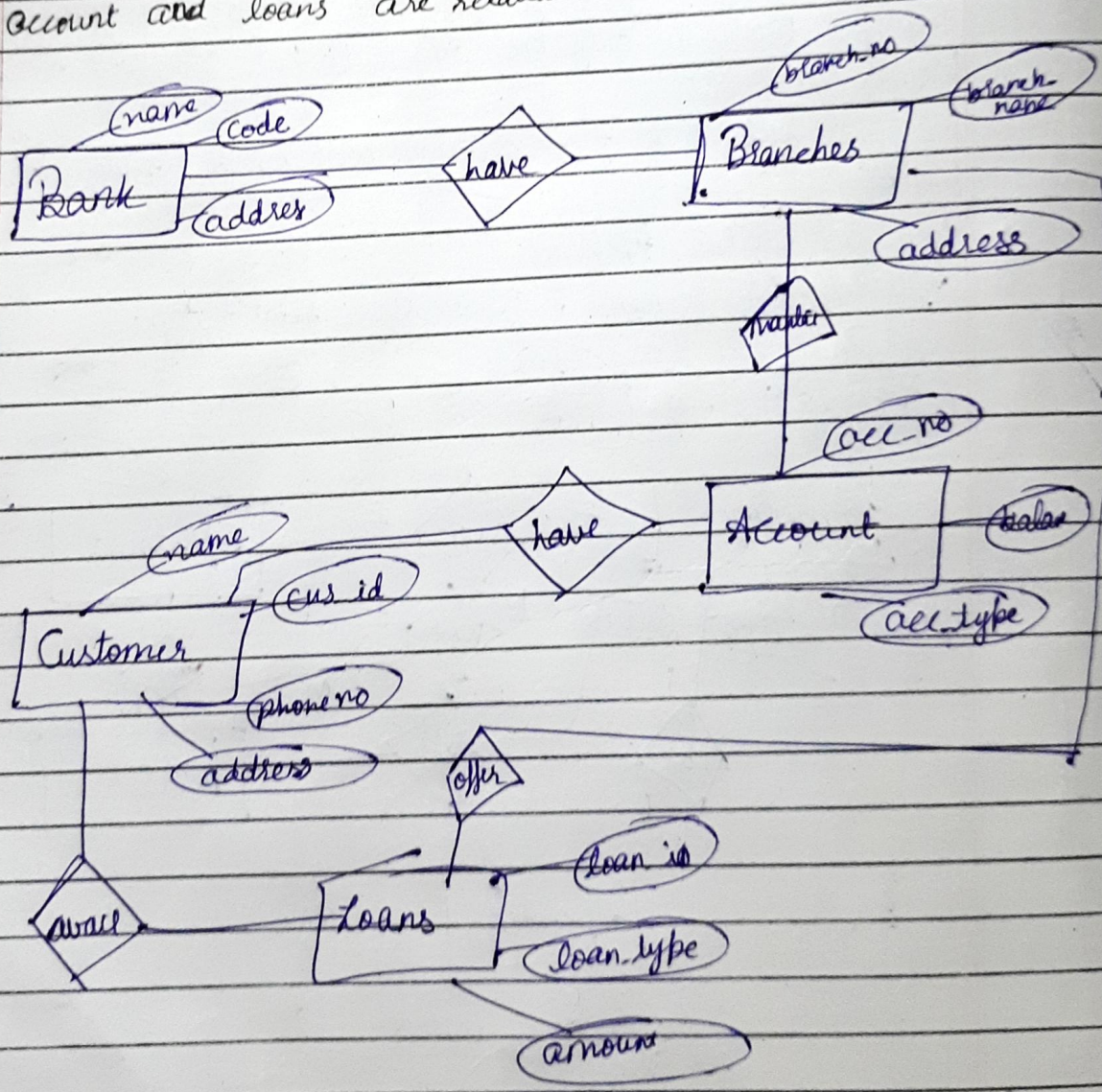
1 o o 1

student can enroll for any no. of courses.
each course can have no. of students.

☆ Company has several departments.
each departments may have sarcal location.
department are identified by a name, d_no, location.
manegar control a particular department.
each department is associated with no. of projects.
employee are identified by name, id, address, dob, date_of_joining.
employee works in only one department but can work on several project.
We also keep track of no. of hours work by an employee on a single project.
Each employee has dependent.
dependent has d_name, gender and relationship.

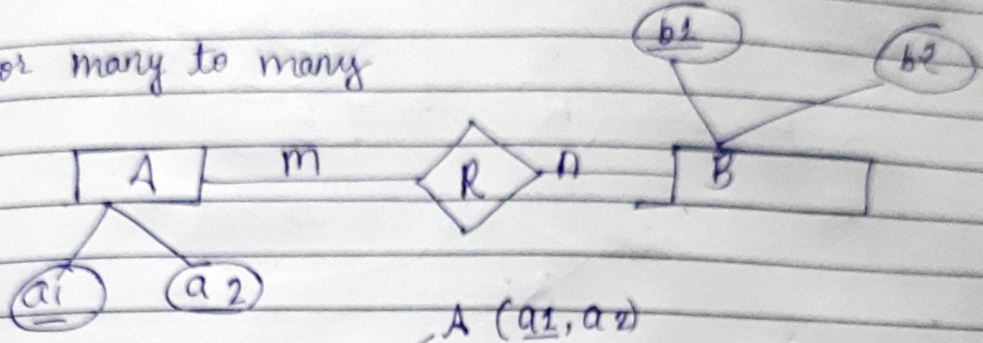


bank have Customer.
banks are identified by name, code, address of main office
banks have branches.
branches are identified by branch_no, branch_name, address.
Customers are identified by name, cust_id, phone no, add.
customer can have one or more accounts
accounts are identified by acc_no, acc_type, balance.
customer can avail loans.
loans are identified by loan_id, loan_type & amount.
account and loans are related to banks branch.



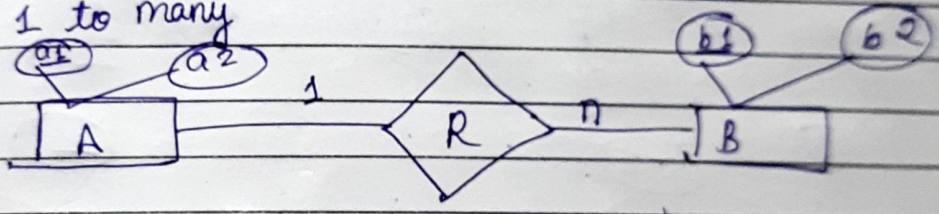
Convert ER into table :-

1) for many to many



3 tables are req. <
- A (a1, a2)
- B (b1, b2)
- R (a1, b1)

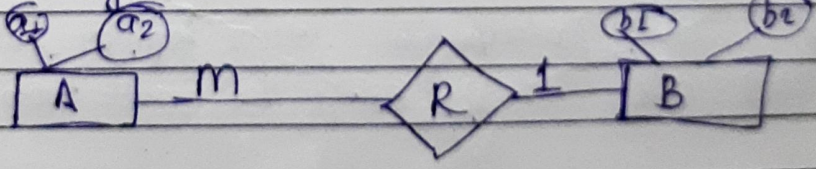
2) for 1 to many



2 tables are req. <
- A (a1, a2)
- BR (b1, b2, a1)

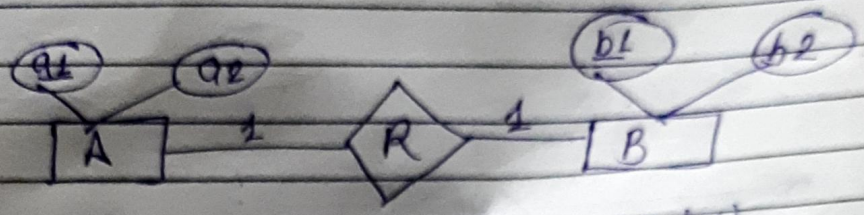
* entity having many site merged with relation.

for many to one - same as 1 to many



2 tables <
- AR (a1, a2, b1)
- b (b1, b2)

4) for 1 - 1

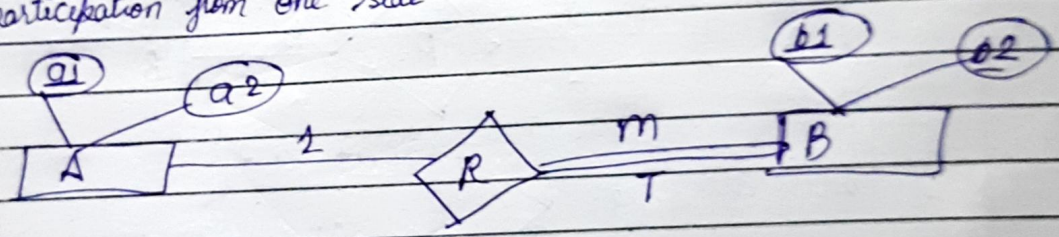


2 ways =

AR (a1, a2, b1)
B (b1, b2)

BR (b1, b2, a1)
A (a1, a2)

5) Binary relationship with cardinality constraints and total participation from one side -



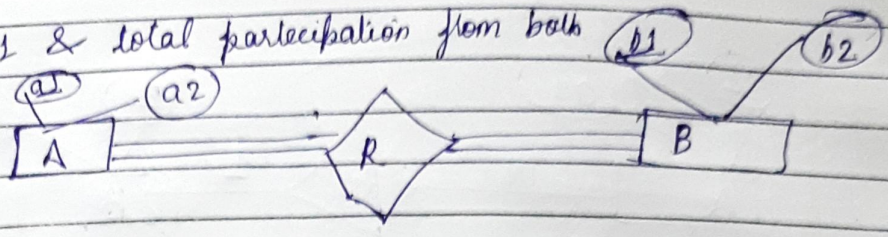
2 table

A (a1, a2)

BR (b1, b2, a1)

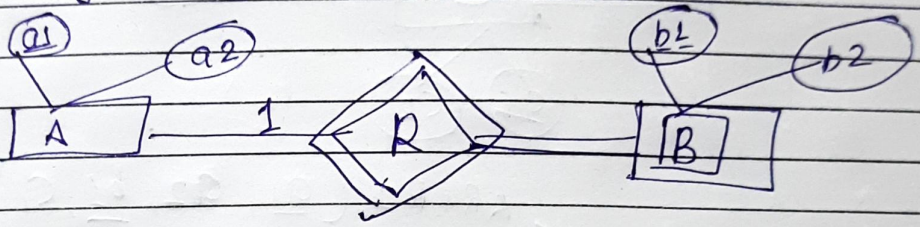
because of total participation, foreign key a1 has acquire NOT NULL cons. so, it can't be null now.

1 to 1 & total participation from both



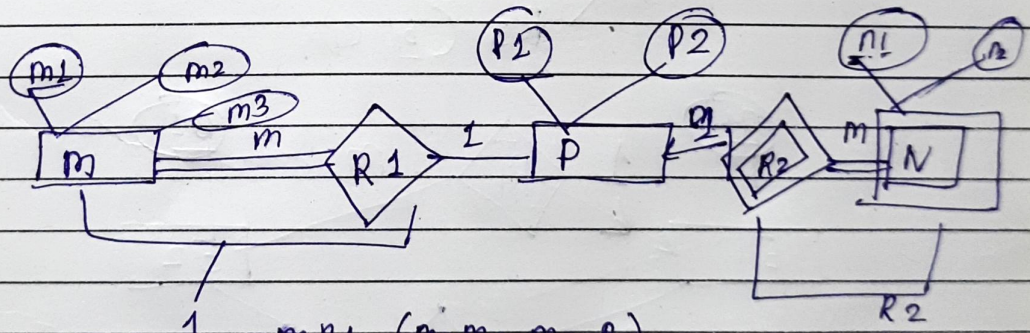
one table is req. ARB (a1, a2, b1, b2).

for binary relationship with weak entity & one side participation



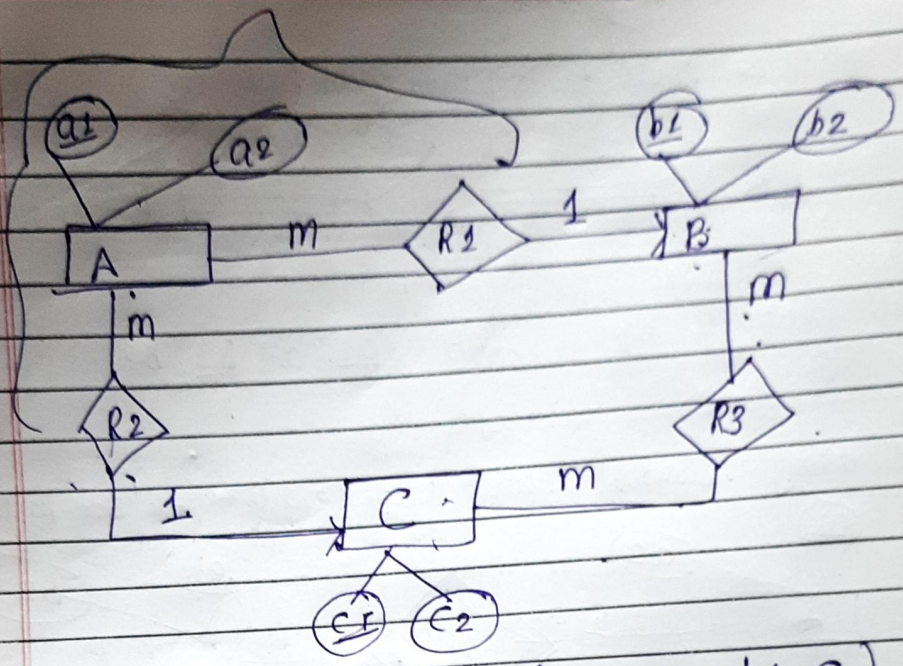
2 table
- A (a1, a2)
- BR (a1, b1, b2)

Q→



3 table
- mR1 (m1 m2 m3, P1)
- P (P1, P2)
- mR2 (P1 n1, n2)

→ for one to
— for many



(5)

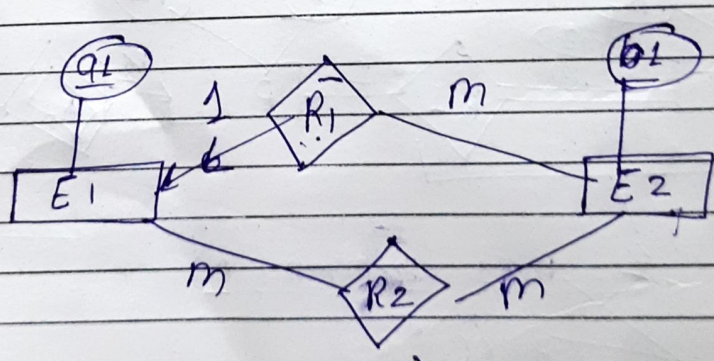
A R1 R2 ($\underline{a_1}$, a_2, $\underline{b_1}$, c_1)

3 table

C ($\underline{c_1}$, c_2)

B R3 ($\underline{b_1}$, b_2, c_1)

R3 ($\underline{b_1}$, c_1)

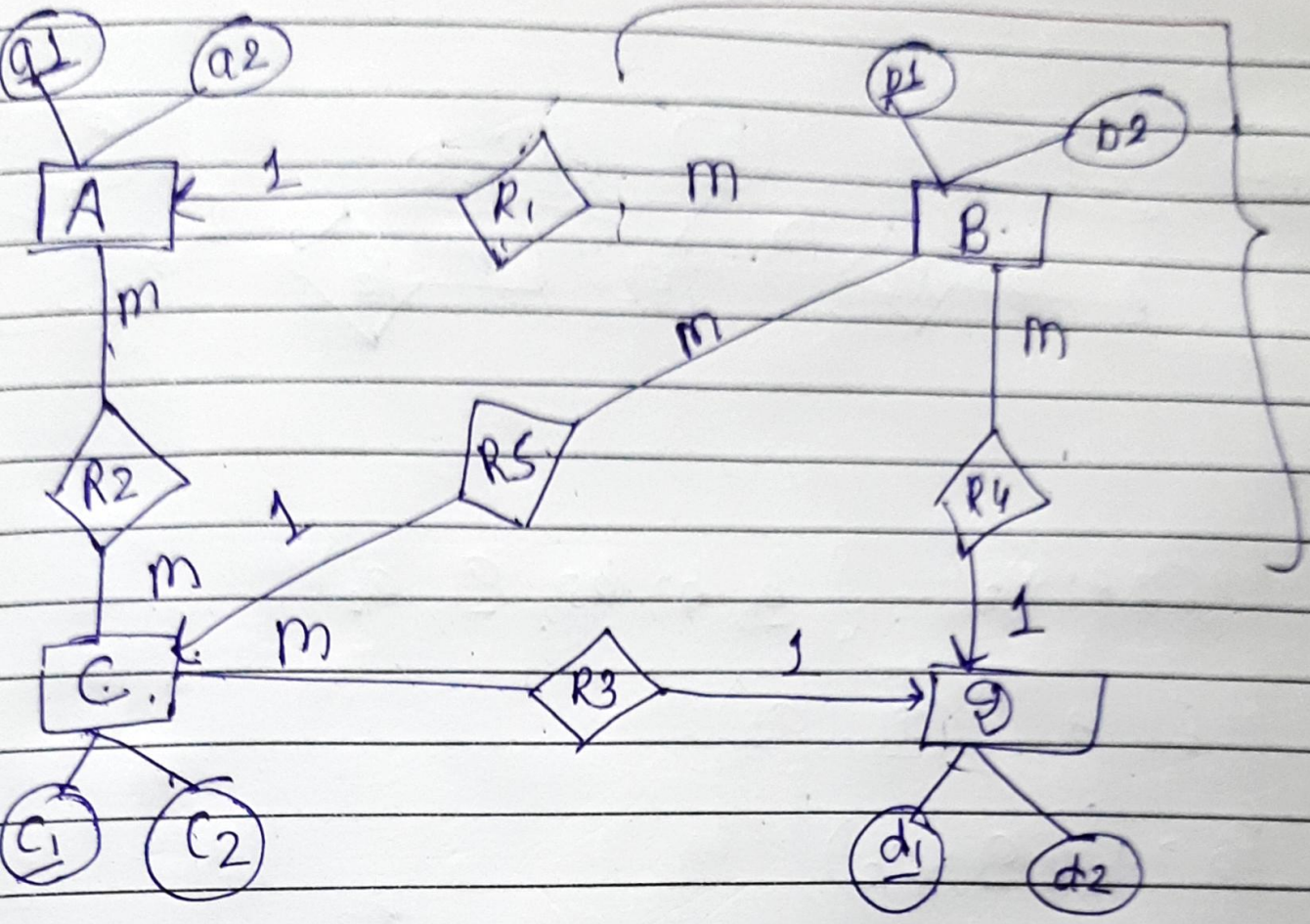


E2 R1 ($\underline{b1}$, a_1)

3 table

E1 ($\underline{a1}$)

R2 ($\underline{a_1}$, a_2)



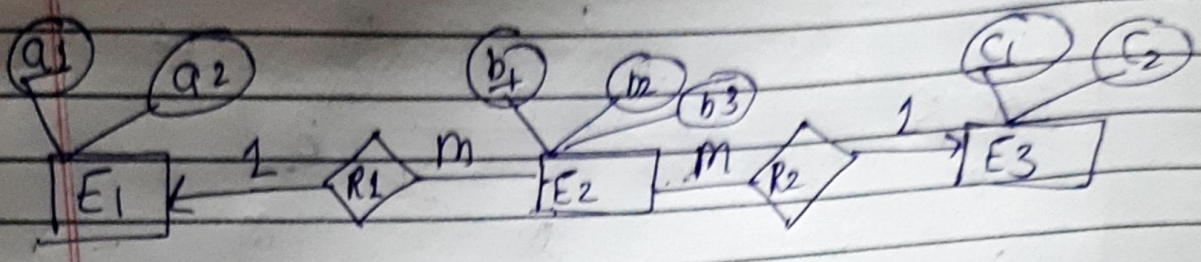
B $R_1, R_5, R_4 \rightarrow \{\underline{b_1}, b_2, \underline{a_1}, \underline{c_1}, \underline{d_1}\}$

C $R_3 \rightarrow \{\underline{c_1}, c_2, \underline{d_1}\}$

a $\rightarrow \{\underline{a_1}, a_2\}$

d $\rightarrow \{\underline{d_1}, d_2\}$

$R_2 \rightarrow \{\underline{a_1}, c_1\}$



E2 R1 R2 = { b1, b2, b3, a1, a2 }

E1 → (a1, a2)

E3 → { (c1, c2) }

Object oriented relationship database Model :-

It is a data model in which data is stored in form of objects which are instance of class. class and obj combine together to make this model.

Object :- object referred as attribute. It is real world entity that contain certain attributes that makes the object structure

Variable → store data of an object.

Method → when msg is passed body of code is executed. When method is executed it return value as output

Message → it acts as a communication medium b/w object & outside world.

- Read-only → no change in value of variable
- update → change the value of variable

- Read-only → when value of variable is not affected by method
- Update → value of variable change by method

class :- Object is an instance of class. first we need to define a class then obj are made which diff in value stored in same class definition

Inheridance :- may be many classes with same methods.

encapsulation :- binding up the imp. data in single class.

Relational Model :- how data is stored in the form of a table. we convert conceptual model into relational model using an RDBMS lng. like Oracle, Sql, etc.

Roll_no	Name	Address	Age
1	Ram	delhi	18
2	Ramesh	Kanpur	18
3	Sumit	U.P	19
4	Suresh	M.P	25

Concepts :-

Relation :- 2-d table used to store a collection of data elements.

Tuple :- row of relation / table
Attribute - column of table.
degree - total no. of attribute present in relation.
Cardinality :- the no. of entity takes part in a relation.
Relation schema - logical blue-print of relation / table

Student (Name, Roll no, age);

It makes some rules to ensure accuracy & accessibility of data. these rules are known as Relational Integrity constraints

Domain Constraints:- every attribute is bound to have a value that lied inside a specific range of value

Key Constraints:- primary key (never be NVLL)

Referential Integrity Constraints:-
it is defined b/w 2 inter-related table. it states that if a given relation refers to key attribute of a diff or same table then key must exist in given relation

Advantage.-	disadvantage
Simple & easy to use	performance depends upon no of table
manageability	no of table increases memory increases
data integrity	structure become complex
	take more response time