15/11/24

TATA CONSULTANCY SERVICES

DBMS.

Data ?? ⇒ Bits / Bytes
↳ Collection of raw bytes.

image → collection of bytes - this is data.

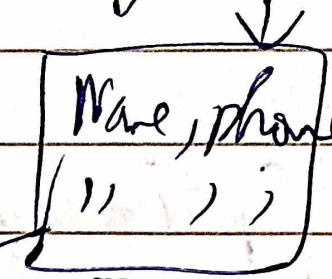
Atomicity problems → To do one thing in a single go.

- Three schema architecture (View of data)

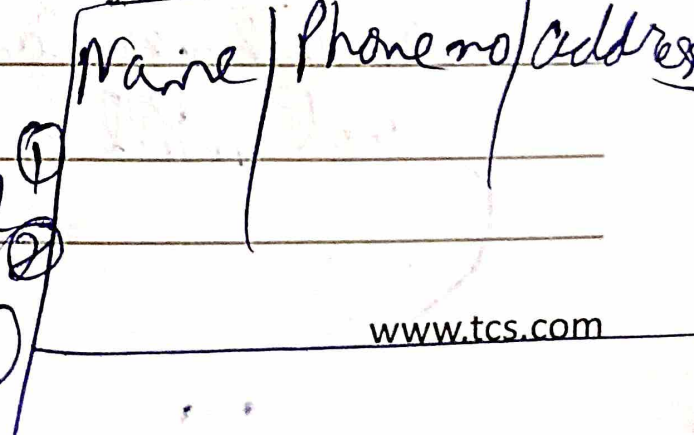
Eg- Facebook → Profile picture img

Conceptual level

Physical level :



file → convert into logical level



TATA CONSULTANCY SERVICES

__/__/___

is internal level data only stores at bit/byte format.

→ Instance of DB.

Student :- logical level

Name	ph	address	batch
1) XYZ	XX	X	X
2) abc	XX	X	XX

3) 12:00 am ⇒ DB?? - 2 Rows

4) Tom 12:00 ⇒ DB = 4 Rows.

→ DB Schema (also logical schema)

① Attributes of table.

② Consistency constraint (primary key not null)

__/__/____

→ logical schema is very important affecting applications.

* Data Models :- Design of db at logical level.

* DBMS - after making it, language comes :- To define db schema like student → contains

Stud-id, NAME, address, phone.

DDL - data definition lang.

like | stud-id | name | add | phone |

insert, update, delete & Retrieval of data (select)

→ DML org - data manipulation - provide insert, update.

→ How app access DB?

App written in java or js

SQL

JDBC
Java database Connectivity API interface Convert Java into SQL Query.

→ Java

↓

App

↓

SQL Query

JDBC

SQL Server

DBMS

TATA CONSULTANCY SERVICES

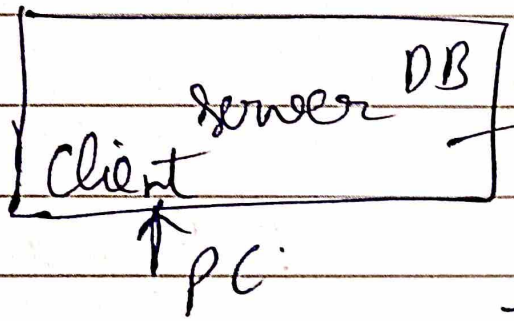
DBA → Database administrator.
DBA makes, logical schema, handle changes at central control, logical/internal level.

* DBMS Application architecture.

Client machine.
↓
end user

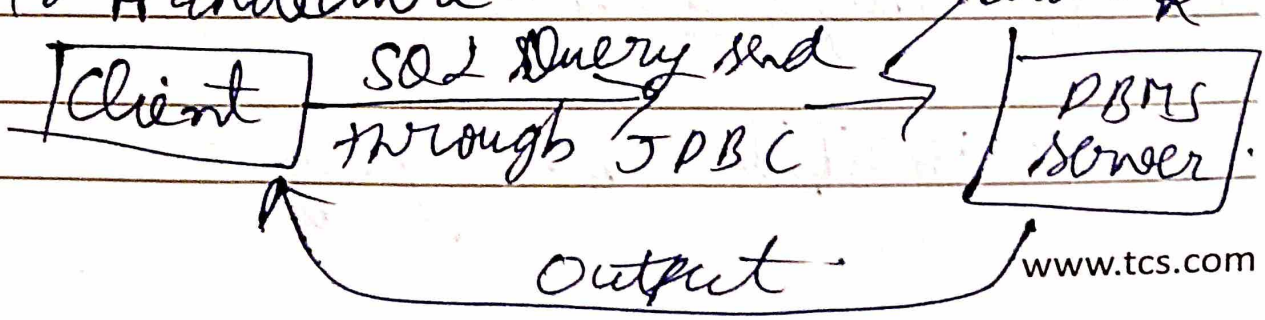
Server Machine
↓
Actual db system work.

1) T1 architecture



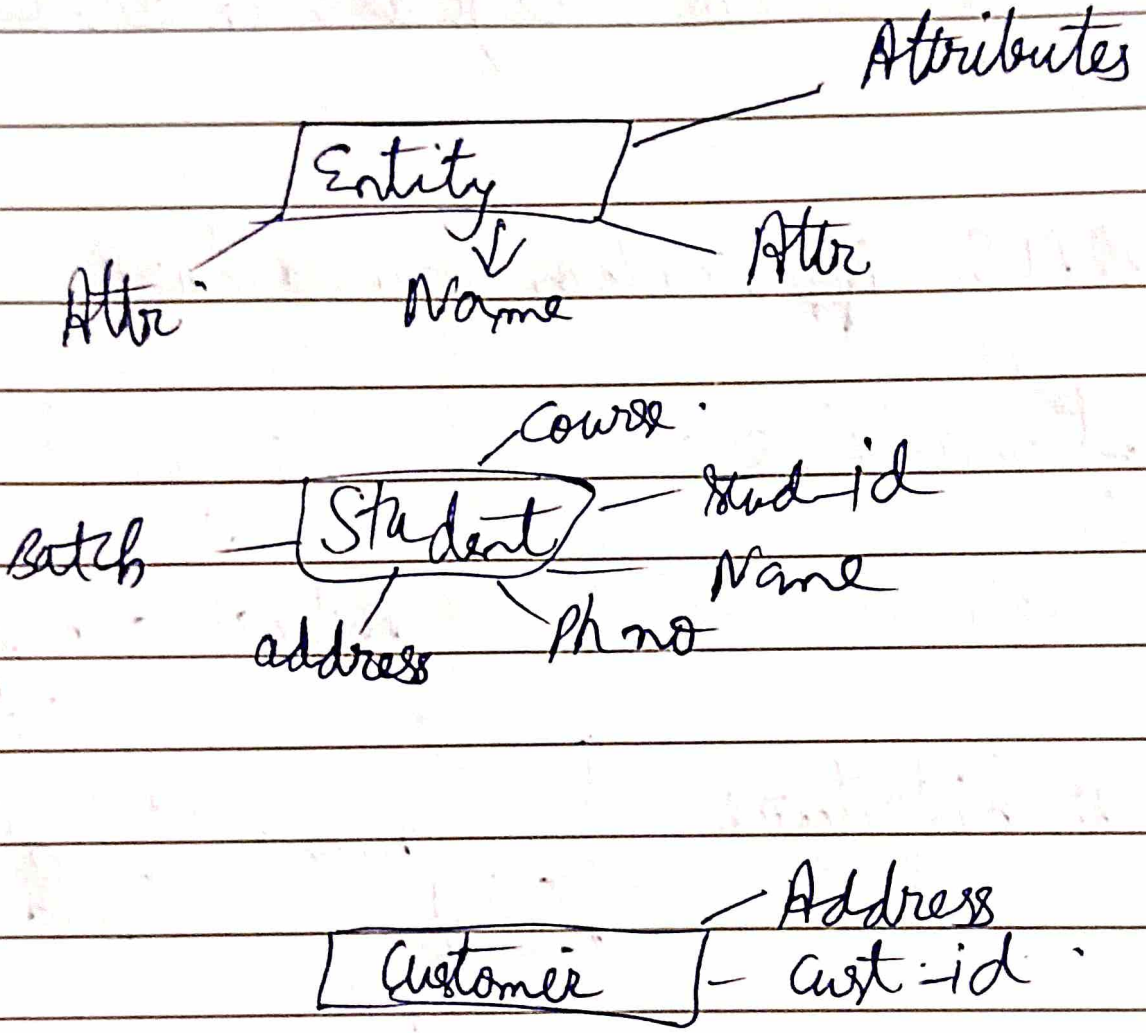
all three are in one PC

2) T2 Architecture:-



3) T3 Architecture

ER Model :- Entity Relationship model

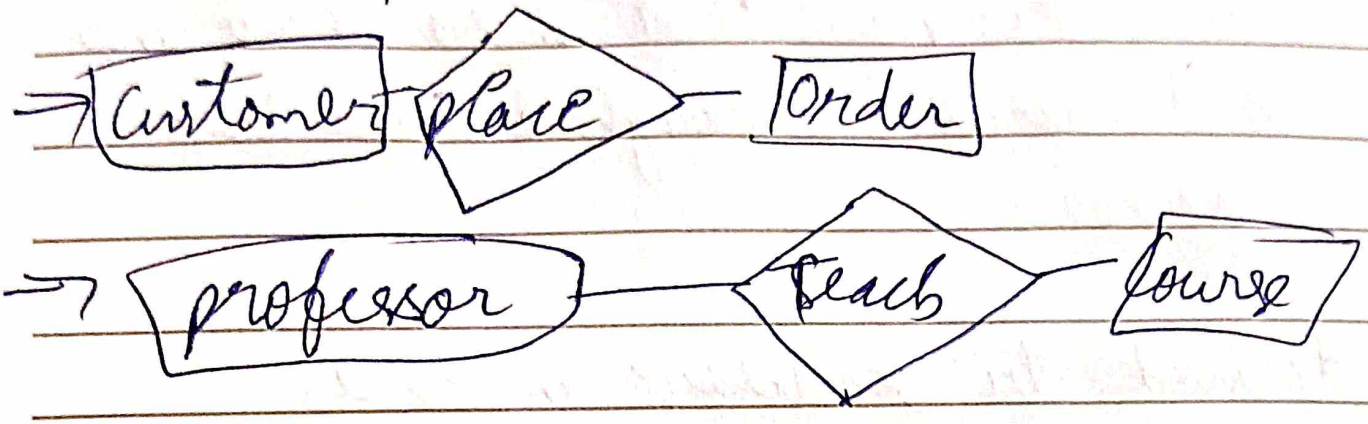


→ Identify (Entity) is Uniquely identify

Unique identified → Primary key

TATA CONSULTANCY SERVICES

Relation represent as diamond.

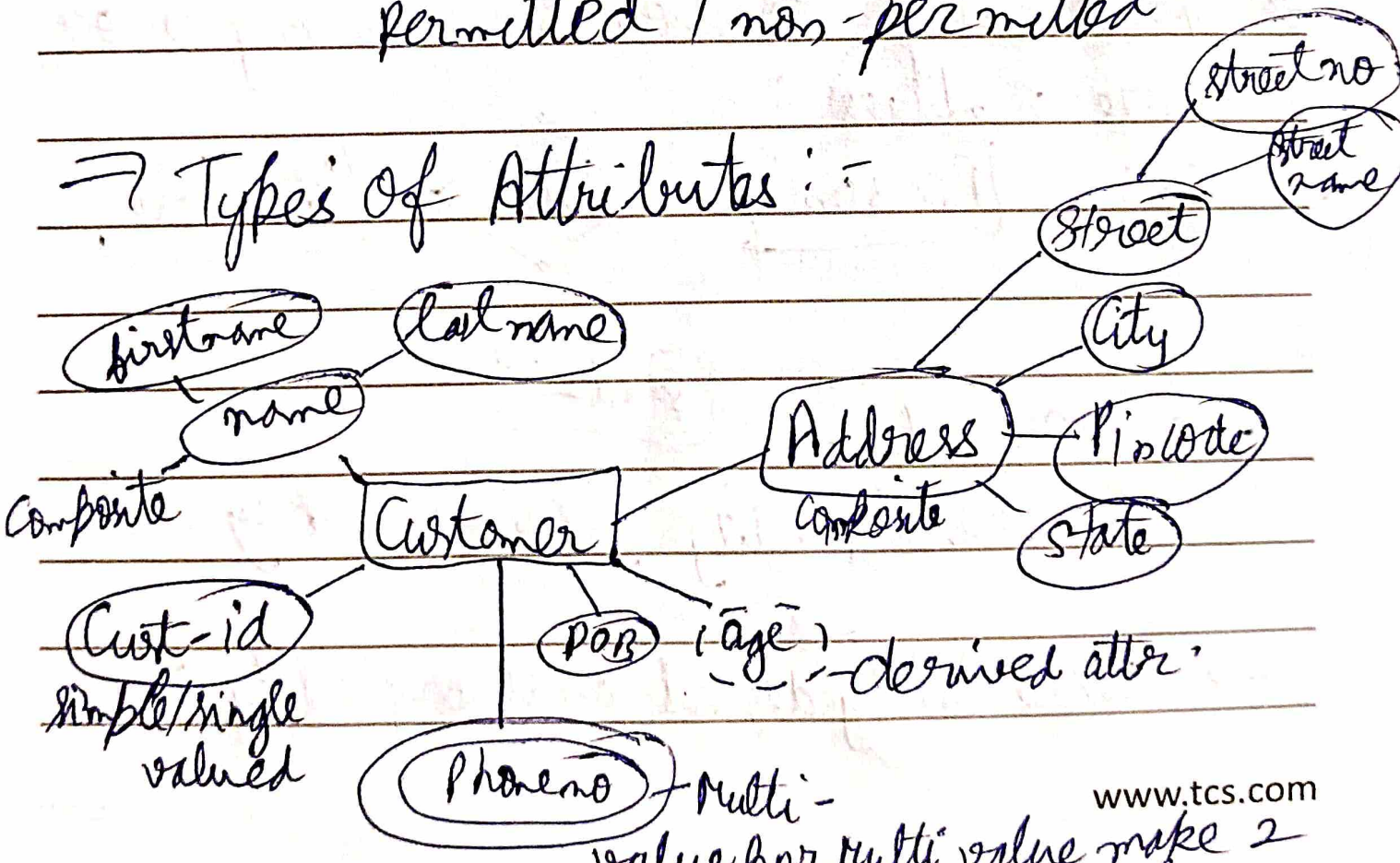


In

Attributes → Customer name will not be null

Attributes → domain, values permitted / non-permitted

→ Types of Attributes :-



→ (phone no) is Multivalued as Customer can have 2 or more phone no & it represent through 2 Concentric circles.

→ derived attr is always make as dotted so from dob attr, age derived attr comes -

* Relationship :- association among 2 or more entities :-

it is

eg :- [Person] — <has> — [Vehicle]

→ [Student] — (Stud-id)

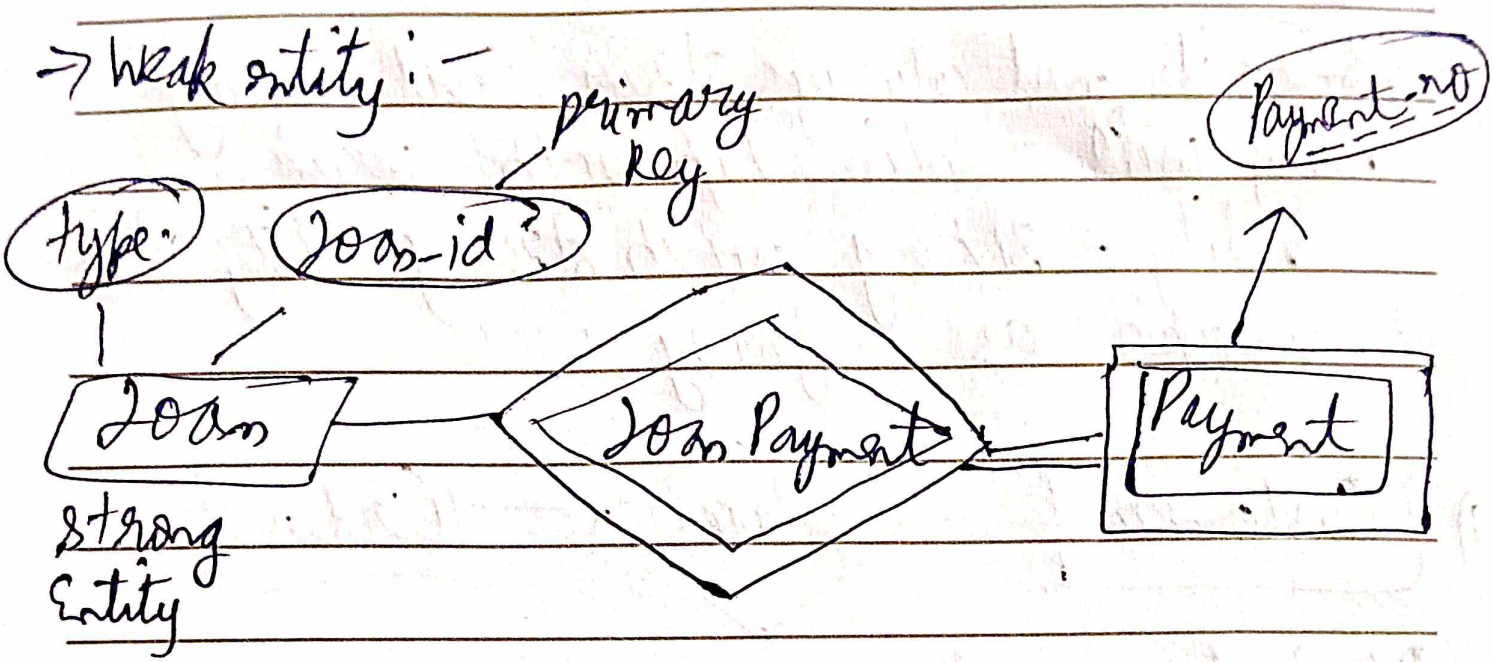
Strong entity (Primary-Key)

→ Weak Entity depend on strong entity.

TATA CONSULTANCY SERVICES

__/__/____

→ Weak entity :-



Payment Entity exists only when there is strong entity of loan entity

Weak entity doesn't have its own primary key

Here is payment no → dotted line represent weak primary key as only it have value like 1, 2, 3 only no not any specific values

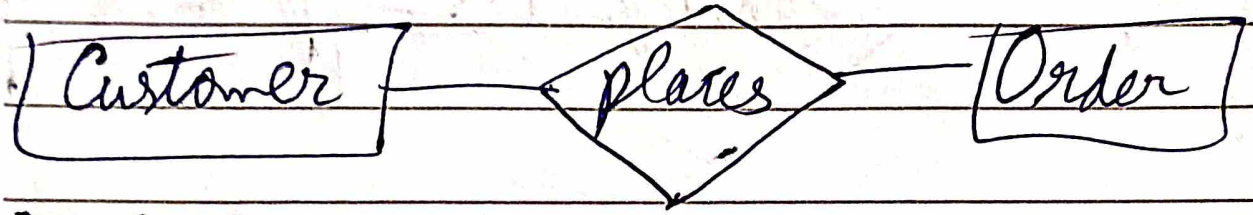
Payment entity can't be uniquely identified.

Weak entity represent as double concentric rectangle and relationship shows as weak double diamond.

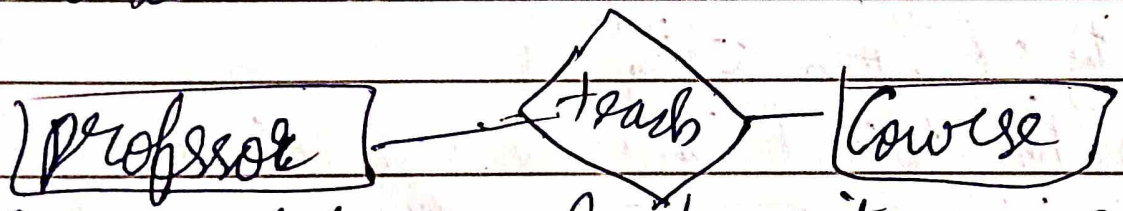
__/__/___

Imp

→ ~~Same~~ its not always true that strong entity will relationship with weak entity. Strong with strong entity also works as :- for eg.

1) 

As Customer also have its own Customer-id & order also " " " order-id.

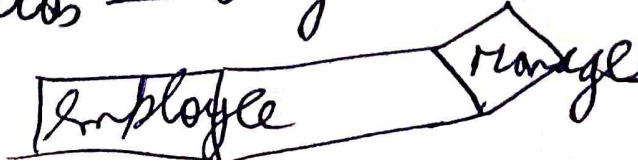
2) 

Same professor also have its own professor-id and course " " " " course-id

Degree Of Relation

In the relation how many entities there is known as degree of relation.

① Unary Relation — Only one entity



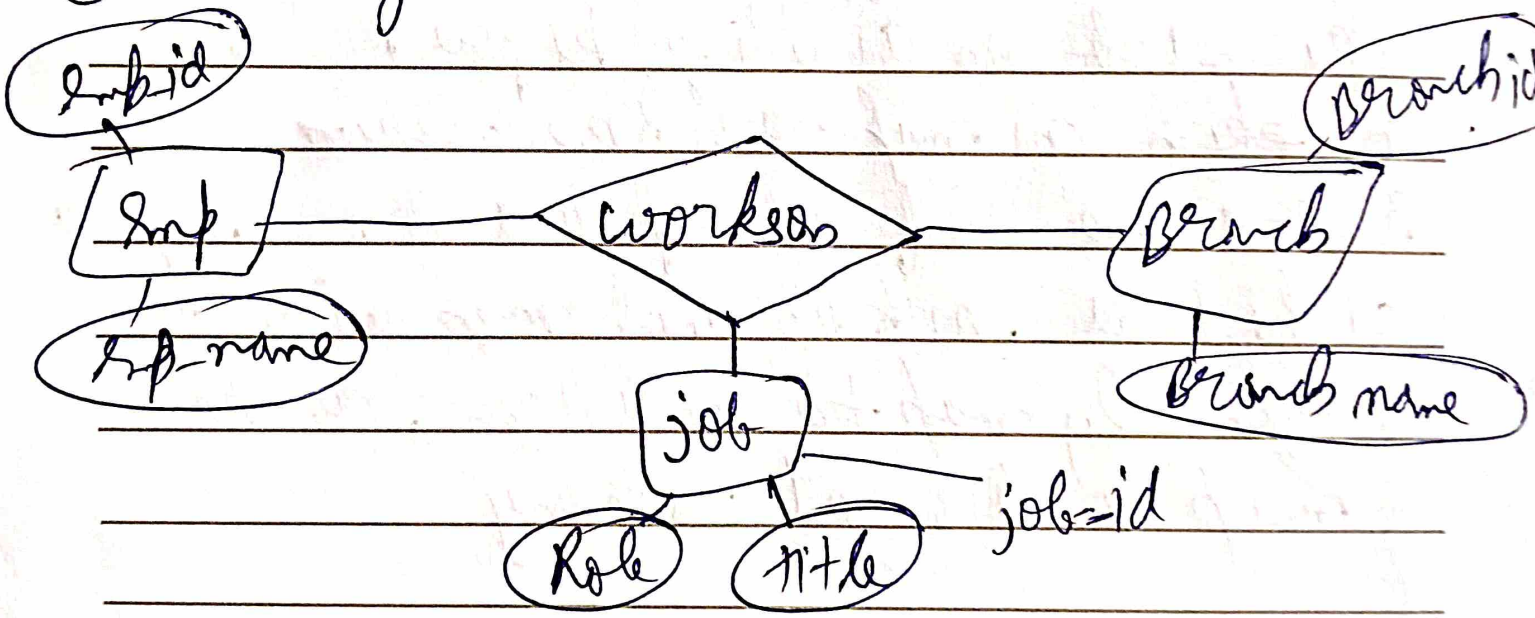
TATA CONSULTANCY SERVICES

__/__/____

any
like in organisation employee manages himself so its unary.

② Binary relation :- 2 entities

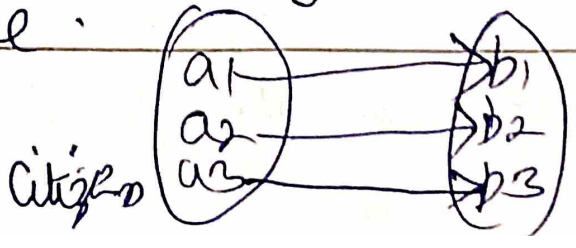
③ Ternary relations – 3 entities (less).



* Relationship constraint :-

① Mapping cardinality.

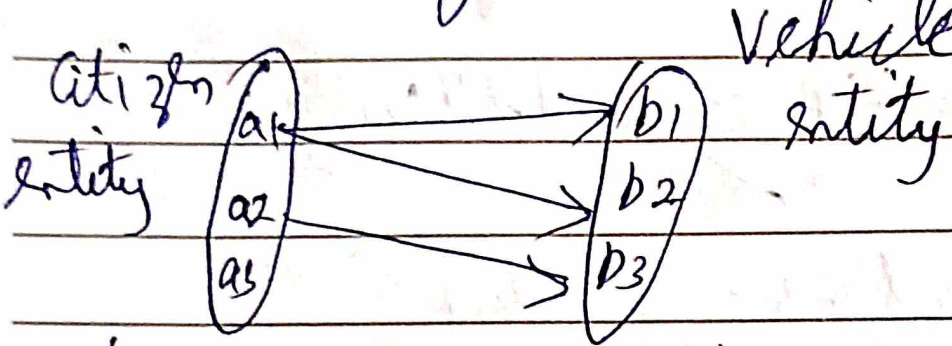
(i) One to one.



www.tcs.com

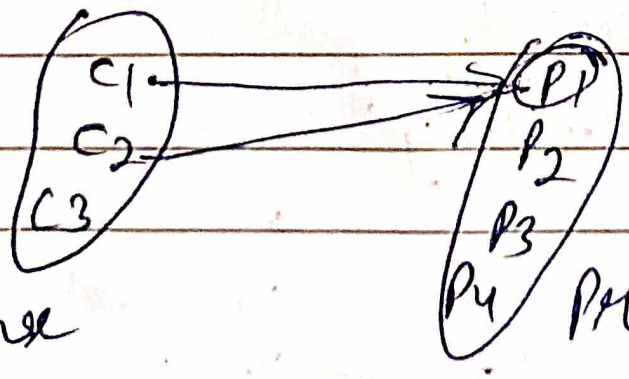
One citizen set — set to — aadhar entity set (One to One).

(ii) One - Many.



iska matlb ke a1 citizen ke pas b1 & b2 cars h lekin a1 map to b1 & b2, lekin reverse me sirf b1 map to a1 & b2 map to a1 hi hoga, b1 & b2 not map with other citizens only map to a1 citizens; as like a2 map to b3 not with any.

(iii) Many to One.

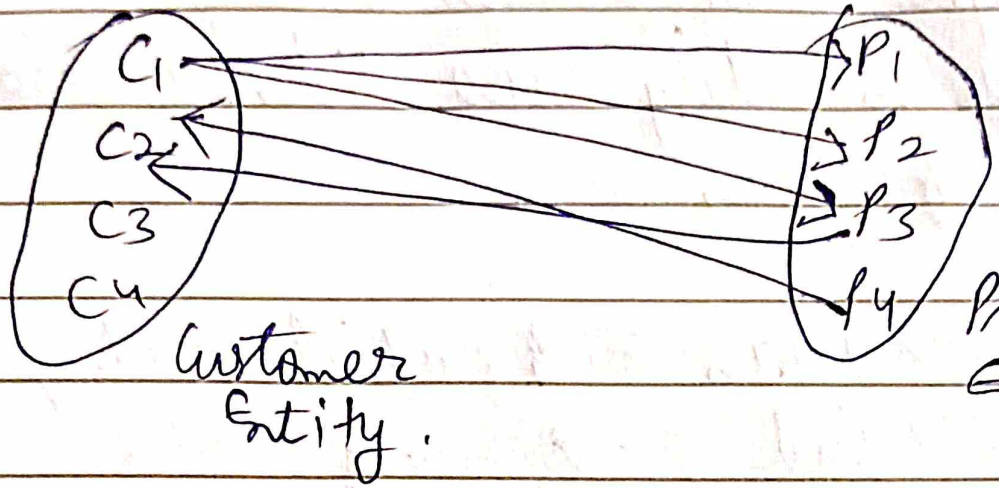


One professor teaches multiple course.

TATA CONSULTANCY SERVICES

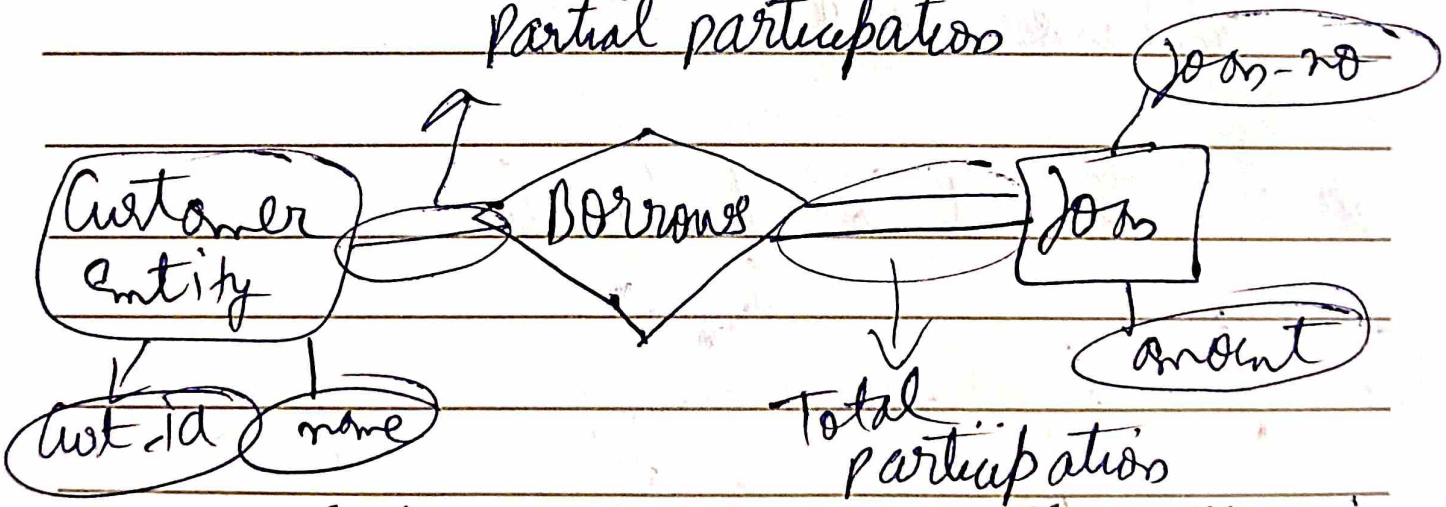
__/__/____

(iv) Many to Many



eg :- Student attend Course.

* Total Participation constraint :-



double bar represent as total participation so it means that in Loan entity there will be not only type of loan which is not taken by customer. Each loan taken by one

80×20000
= 10,00000 - 4/2u

Customer: One single bar represent as Participl participation, any Customer in entity who didn't take any loan.

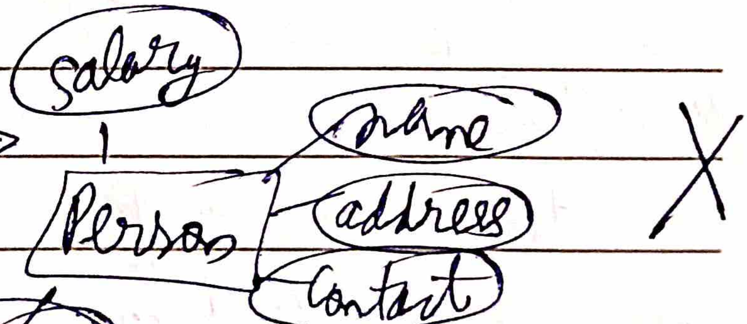
Cust → Loan = partial participation
Loan → Cust = total "

Imp Note :- Weak Entity has total participation constraint, But strong entity may not have total or Can have total.

Extended ER features:-

① Specialization

if we do like this →

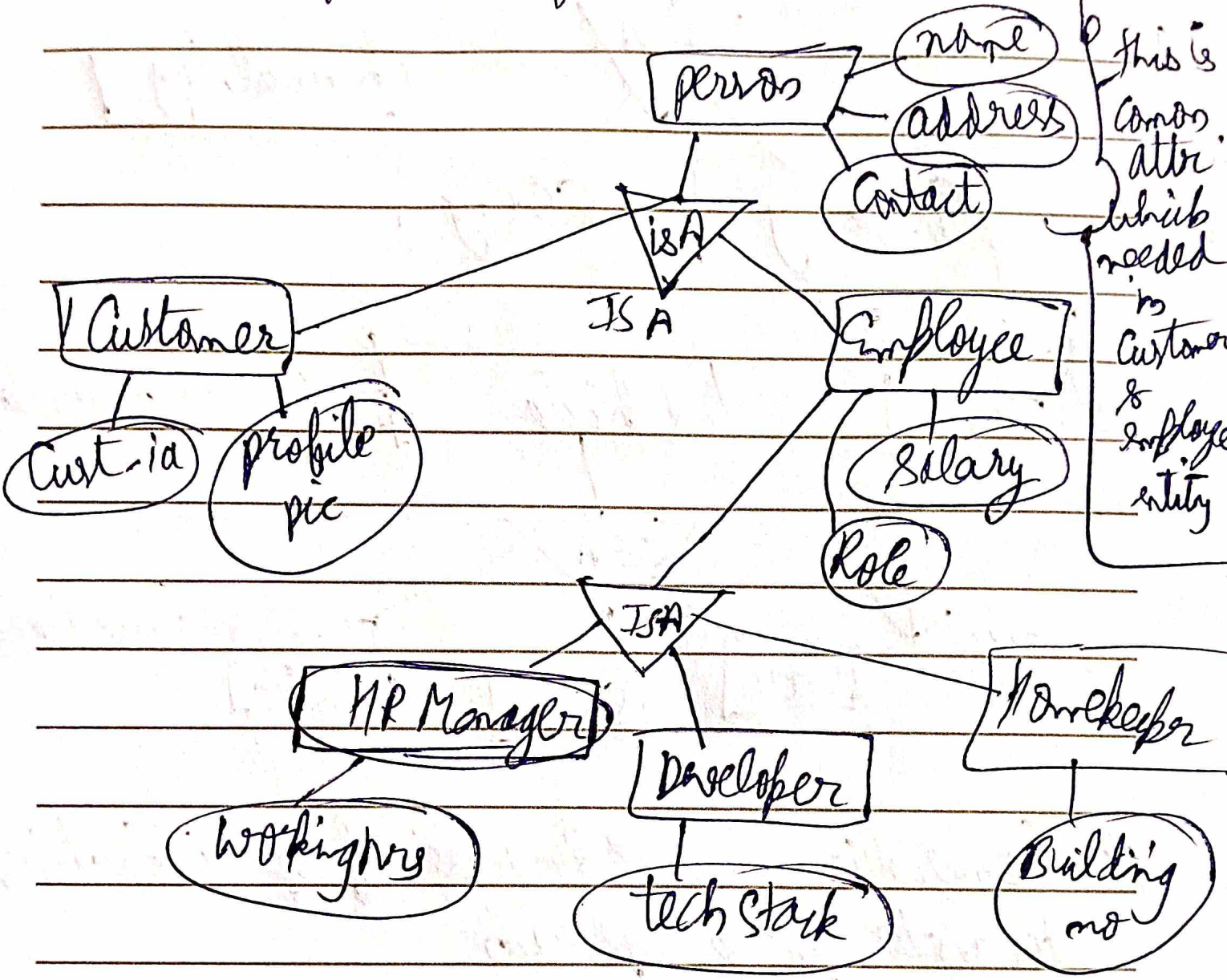


on person entity then we have put load of "lafattu" ke attr aa gye aur (Rutsia) - redundancy bhad gyi, toh yeh galath.

TATA CONSULTANCY SERVICES

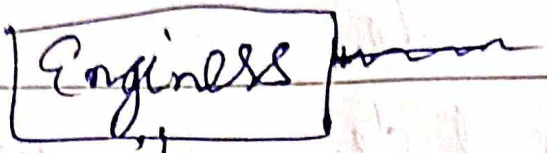
db.

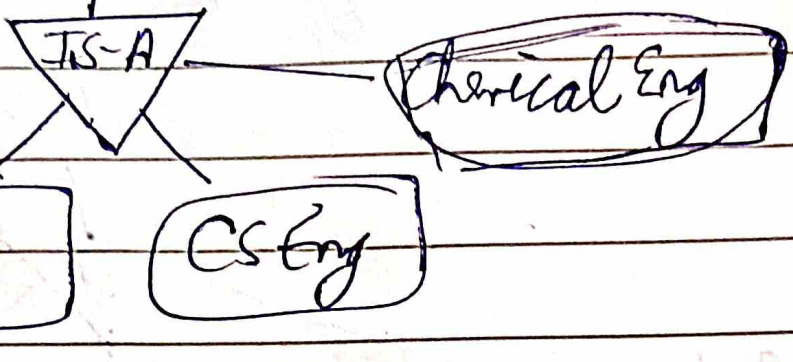
So to make good design we make it like this ↓



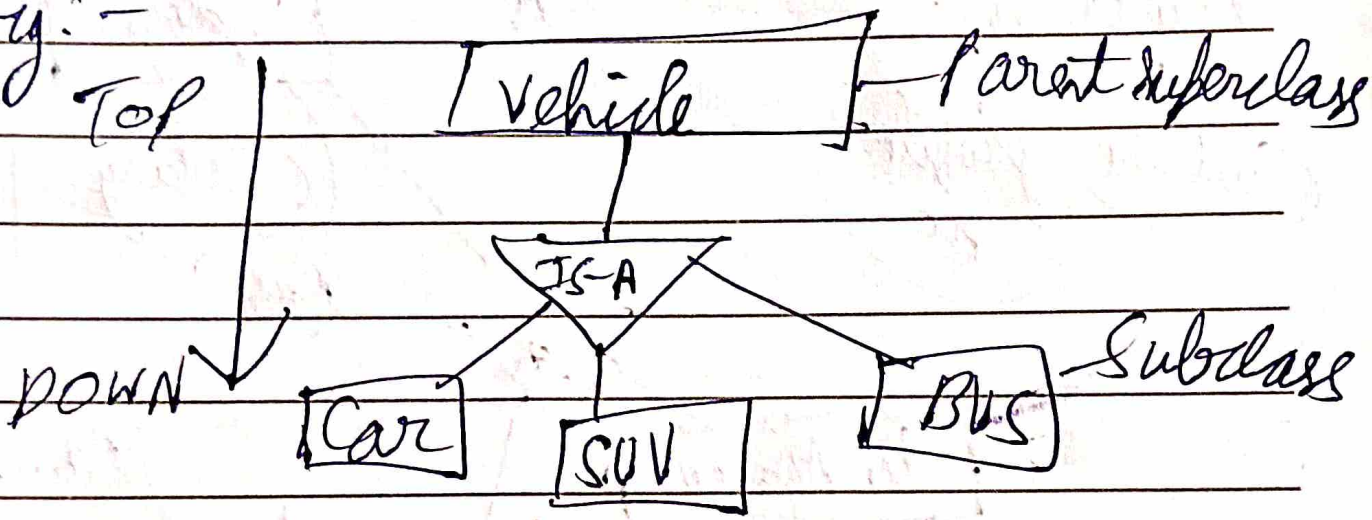
Full DB system Using "Specialisation"
It is "top-down Approach.

More Specialisation sets.





Eg:-

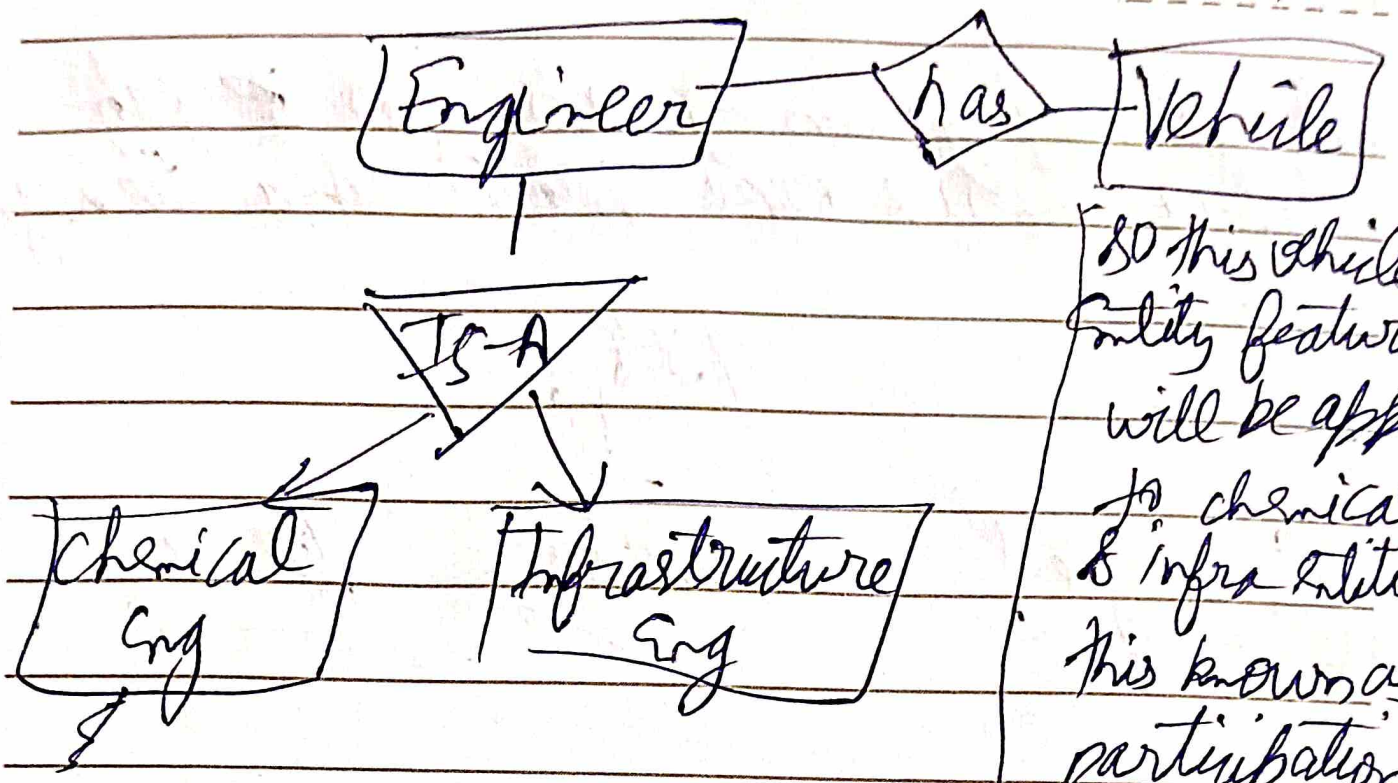


* Generalisation → Reverse of specialisation, down to up/top approach

→ Participation inheritance mtlb agr hume parent superclass entity ka sath koi aur relationship banaya toh vo child entity ke sath bhi ban jayega.

Bcoz :-

TATA CONSULTANCY SERVICES

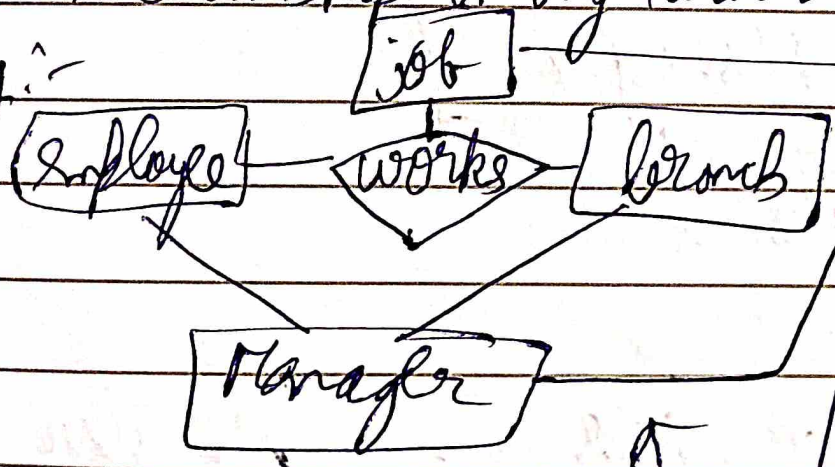


so this vehicle safety features will be applied to chemical & infra entities this known as participation inheritance.

Aggregation — Imp concept :-

Relationship among relationship.

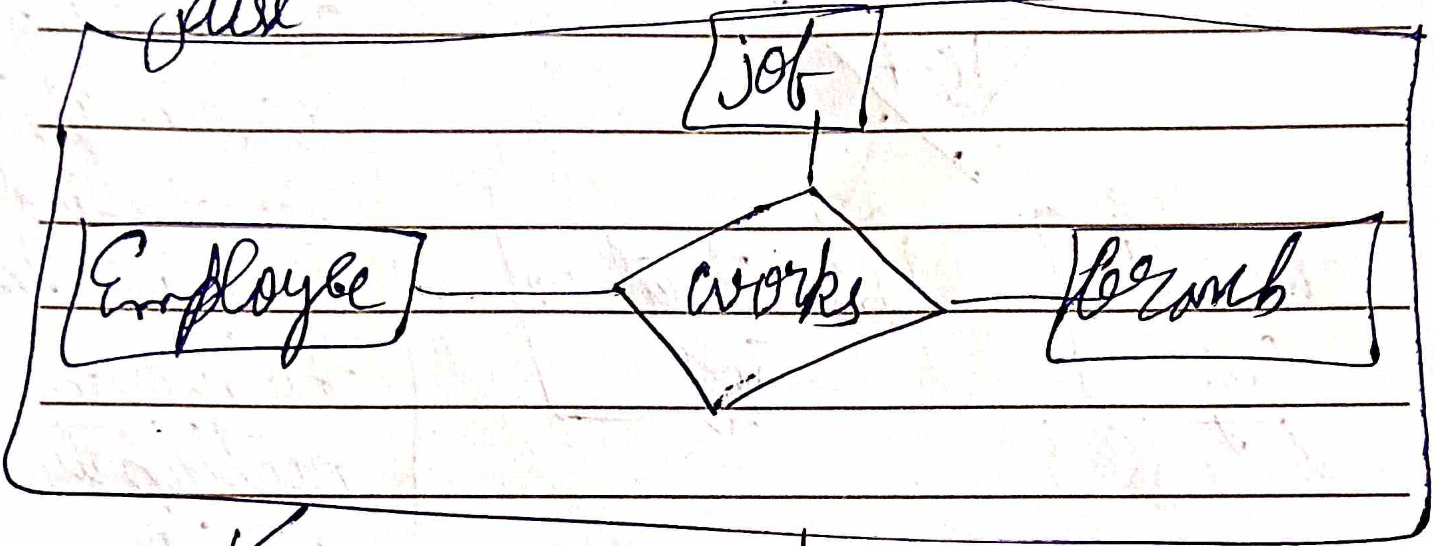
eg :-



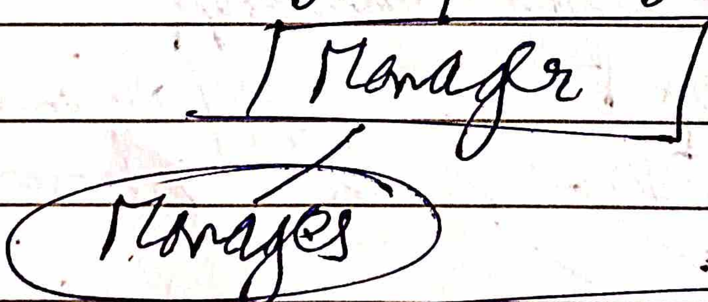
ab isme agr emp-branch job ko manage krne ke liye ek manager chahiye toh agr hum isse aise jodte h. toh redundancy lagegi isliye hi aggregation use hota h.

Aggregation

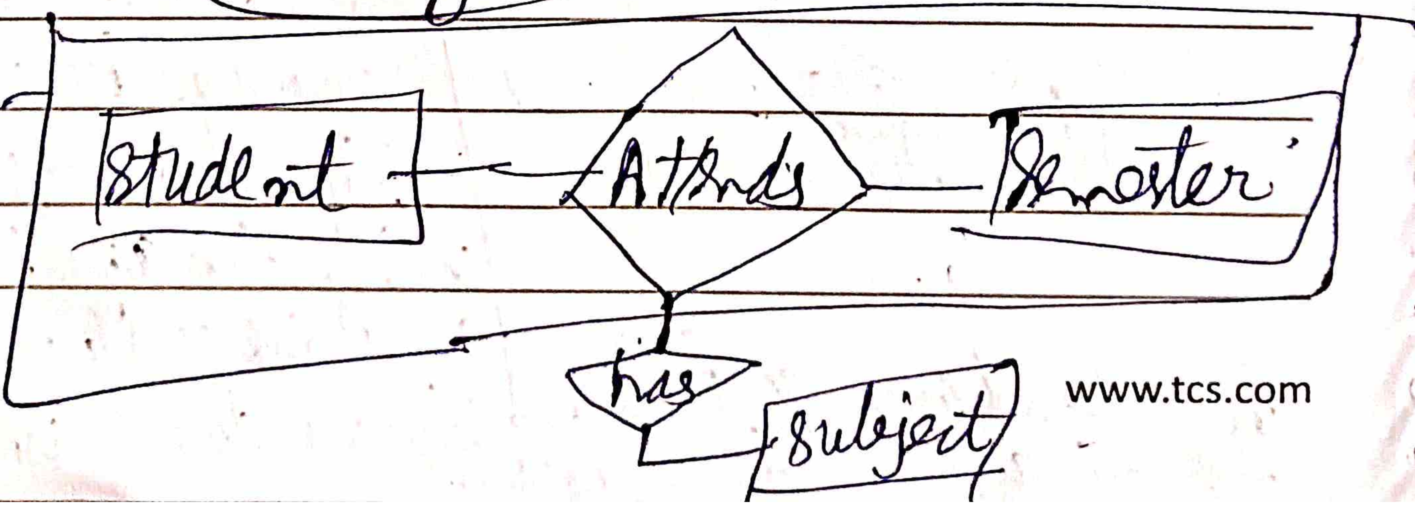
Ab agr aggregation lgateh toh yeah abstraction use krke bahut acha dikhega jaise



ab isko ek awr entity man leteh works nameka awr iss entity ke sath hum add karege ek awr manager entity ko jo iss tino branch employee & job entity ko manage karega.



Eg:



Dec - 5

* Steps to make ER diagram :-

① Identify Entity Sets

② identify attributes & their types

③ identify Relationship & Constraints
   ↳ Mapping
   ↳ Participation

* ER - Model of Banking System :-

① Banking system - Branches Cname
2 Bank → customers
3 Customers → account, & take loan
4 Customer associated with same Banker
5) Bank Employees
6) Accounts — Saving A/C
   Current A/C

7) Loan Originated by branch Loan ∃ Customers
↳ Payment schedules.

① Entity Sets :-

1) Branch 2) Customer 3) Employees
4) Saving A/C 5) Current A/C
6) Loan 7) Payment (Loan)
(weak entity)

② Attributes :-

1) Branch → name, city, assets, liabilities

② Customer → cust-id, name, address, Contact-no, DOB, age.

(3) Employee → emp-id, name, contact-no, dependent name, years of service, start-date (single value) ↓ derived attr.

(4) Saving account → (acc-number, balance), interest-rate, daily withdrawal limit.

(5) Current A/C → (acc-number, Balance), per transaction charges, overdraft amount

(6) Generalized Entity "Account" → acc-no, Balance.

(7) Loan → loan-number (primary key), amount.

(8) Weak Entity Payment → Payment no, date, amount.

③ Relationship & Constraints

① Customer — borrow — loan
M : N

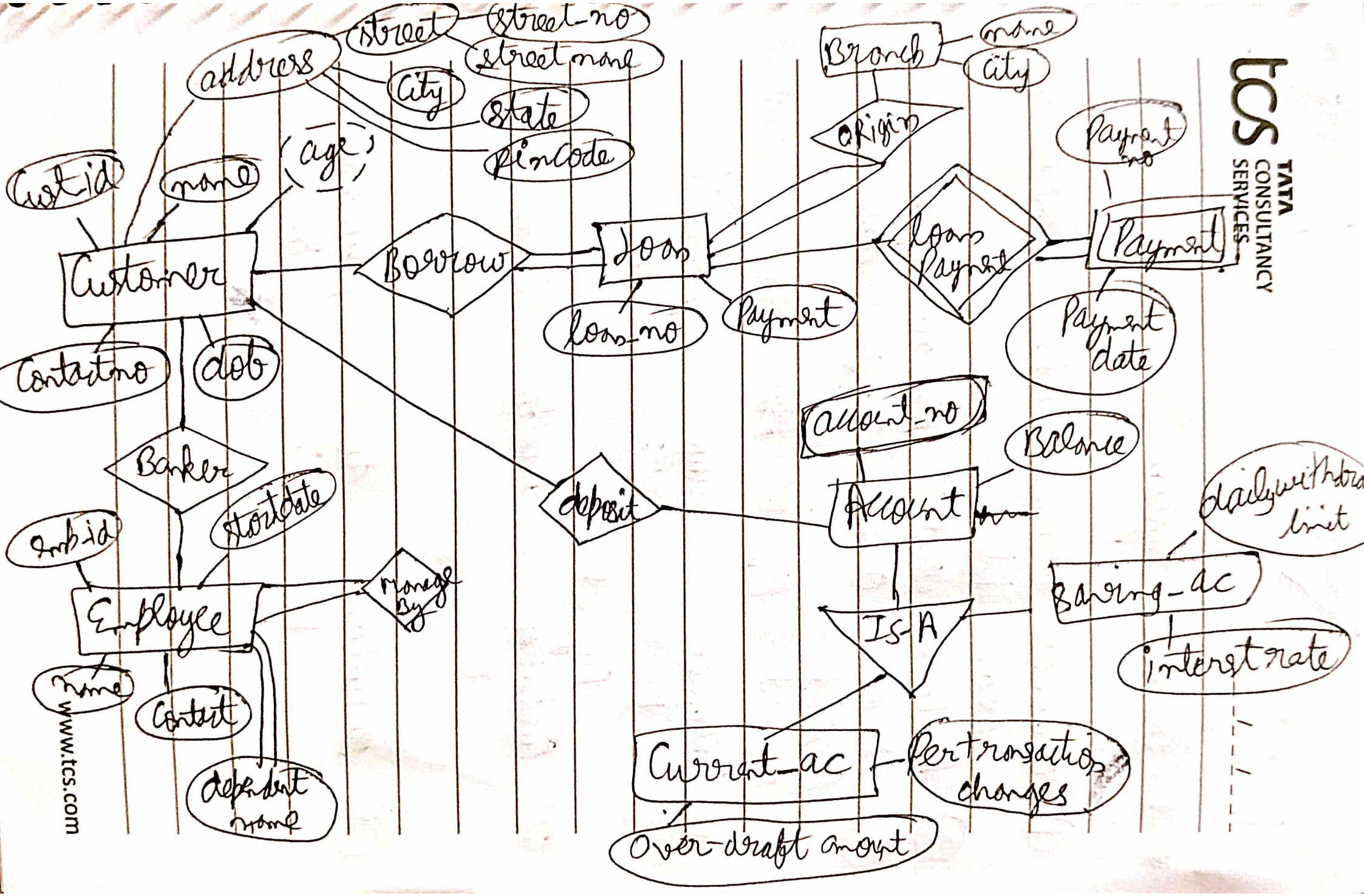
② Loan — originate — by branch
N : 1

③ Loan — loan Payment — Payment
1 : N (weak entity)

④ Customer — deposit — Amount
M : N

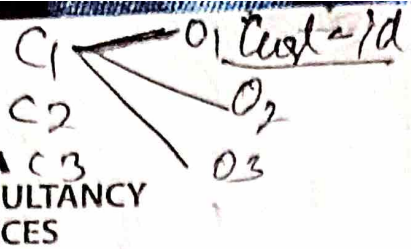
⑤ Customer — Banker — Employee
N : 1

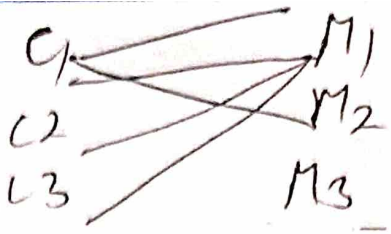
⑥ Employee — managed by — Employee
N : 1

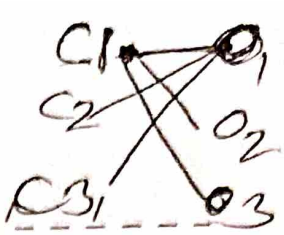


TCS TATA CONSULTANCY SERVICES

www.tcs.com







TATA CONSULTANCY SERVICES

* E-Commerce System :-

① Customer → name, address, contact-no
Customer-id :

② Product → prd-id, prd-name, price, category

③ Order → order-id, order-date, amount, Cust-id

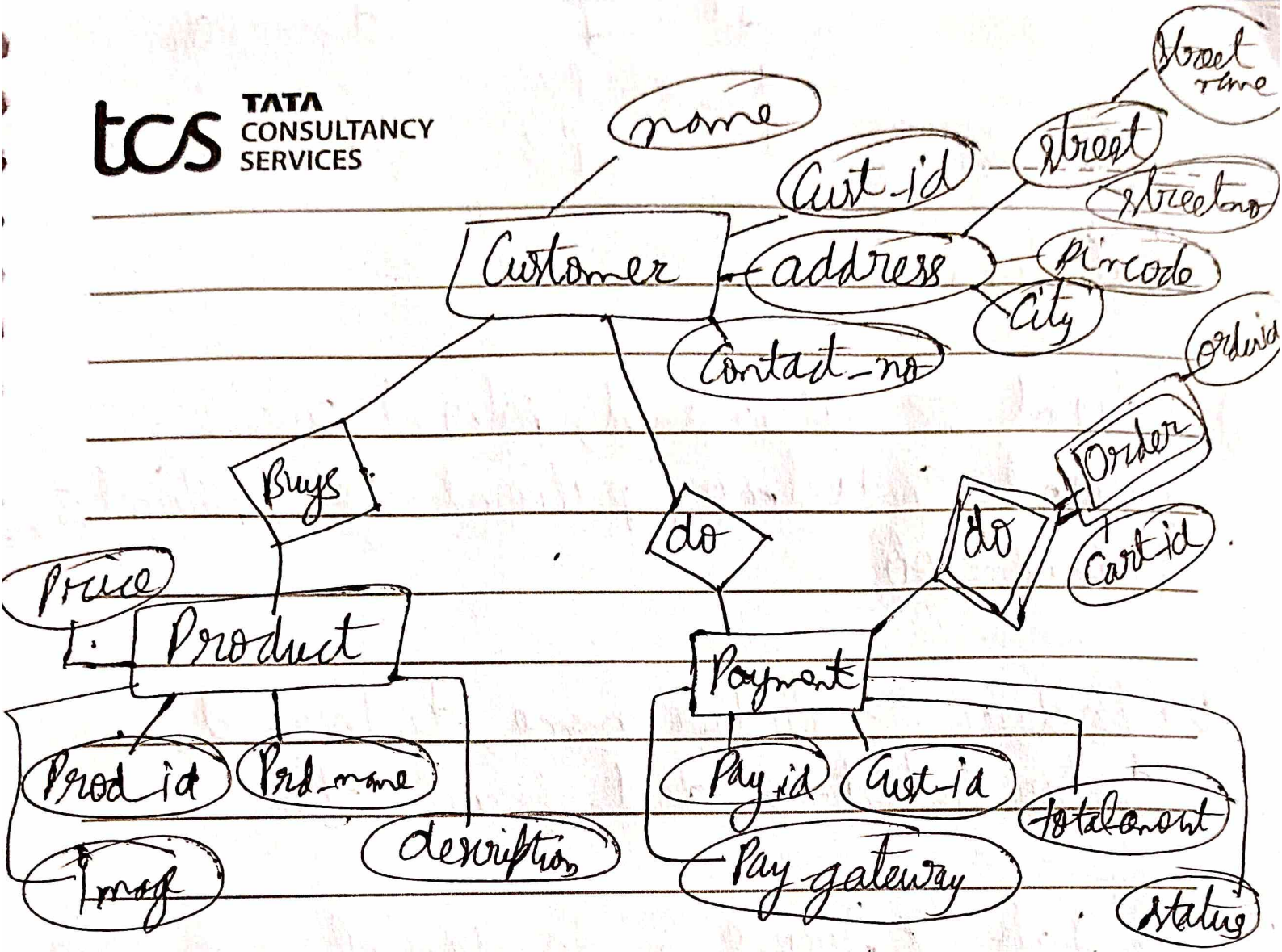
④ Payment → pay-id, Payment-gateway, Cust-id

1) Customer — buys : — product
M n

Payment

2) ~~Customer~~ — has/do : — Order
1 n

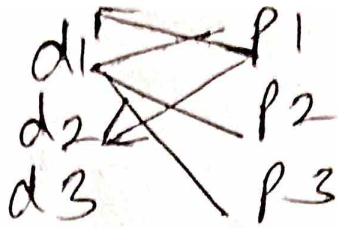
③ Customer — do : — payment
1 n

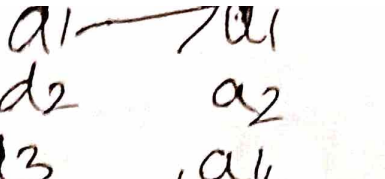


* Hospital Management System :-

Entities

1) Patient
2) Doctor
3) Lab
4) ~~Billing~~ Employees → Billing, cleaning
5) Appointment





TATA CONSULTANCY SERVICES

Attributes :-

1) Patient → Patiend-id, patient-name, patient-Contactno, patient-age, patient-symptom desc, patient-dob

2) Doctor → Doctor-name, doctor-id, doctor-Contactno, doct-specialisation

(3) Lab → lab-id, sample-coll, lab-price

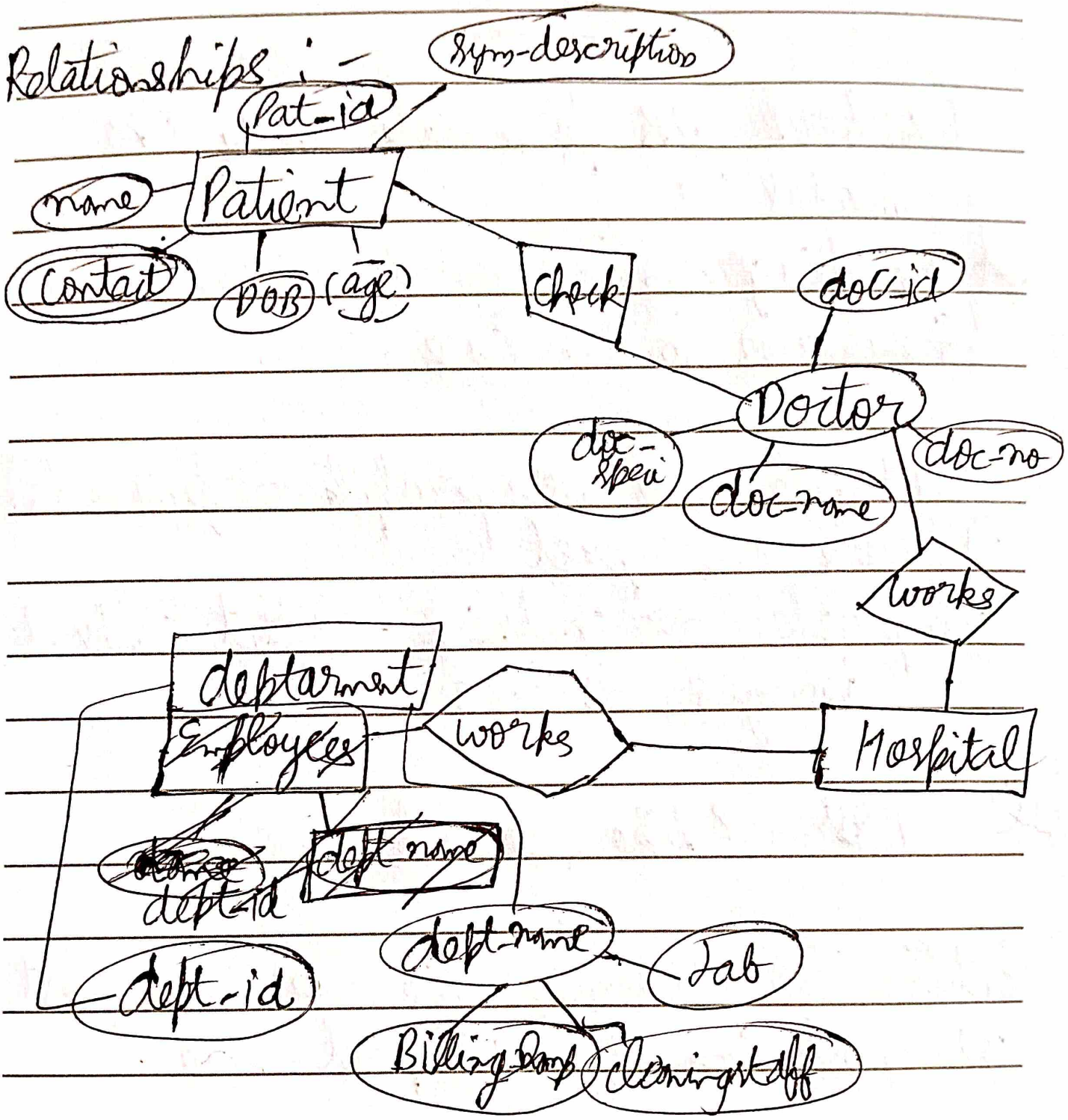
(4) Employees → name, Contact-no, dept, Billing, cleaning

(5) Appointment → app-id, app-date, app-time, patient-id, doctor-id,

→ doctor checks Patient
m : n

→ doctor make appointment

Relationships :-



TATA CONSULTANCY SERVICES

Dec-6.

Facebook - DB formulate using ER model :-

* ER diagram :-

Features & use case :-

① Profile → user profile — friends
② User can post
③ post → contains → text content, images, videos
④ Post → like, coment

Entities :- 1) User-profile 2) User-post
3) post-Coment 4) post-like

Attributes :-

User-profile → name (composite), dob, age (derived), multivalued Contact no, password, gmail

TATA CONSULTANCY SERVICES

② user-post - post-id, post-pic, post-content, Createtime, modytime, text-content

③ Post-coment - coment-id (primary key), coment-time, coment-content

④ Post-like - like-id, time-stamp.

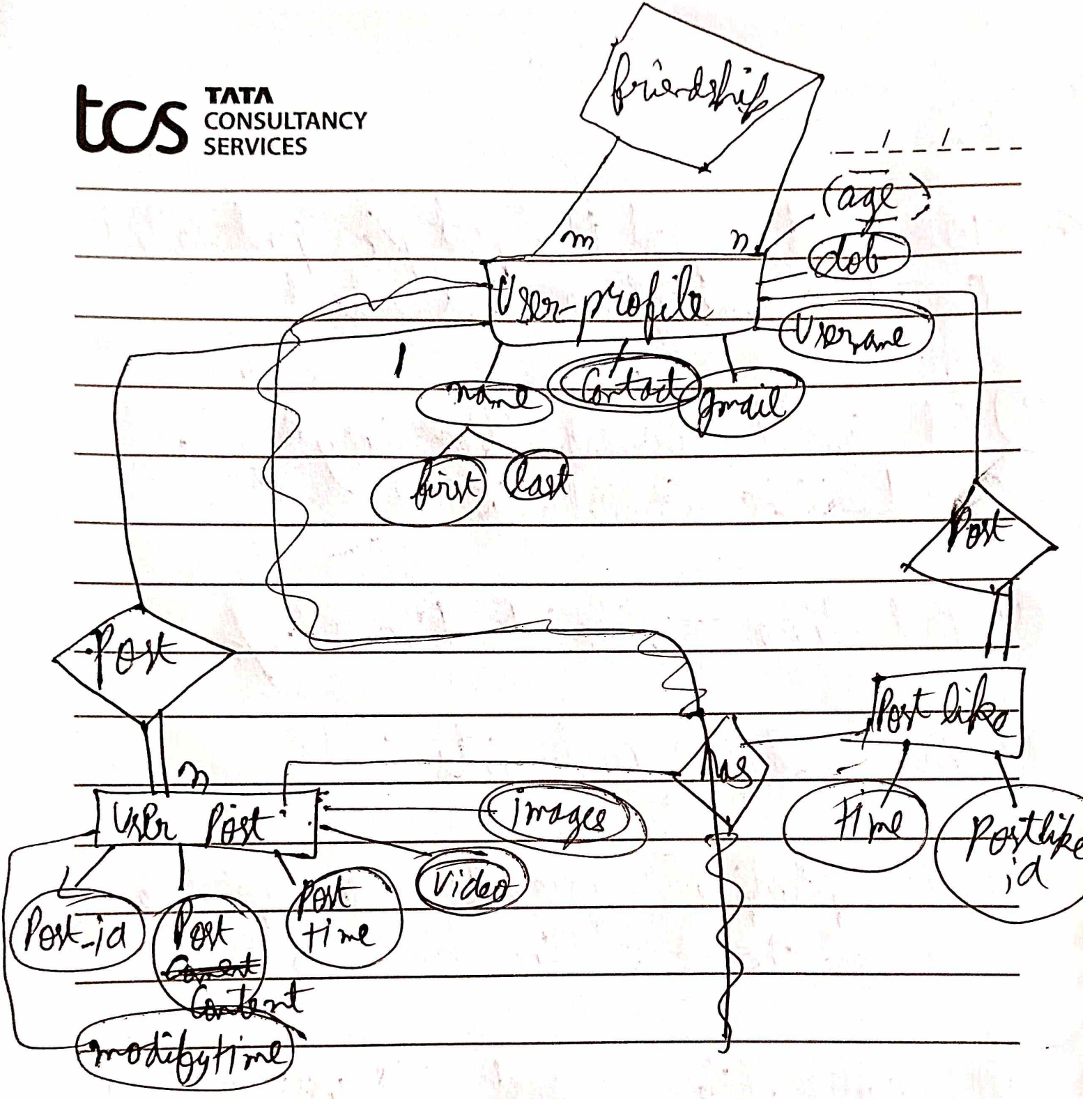
→ ③ Relationship & constraints

1) User-profile — friend — user-profile
m : n

② User-profile — post — user-post
1 : m

③ User-profile — post — post-like
1 : m

④ User-profile — post — coment
1 : m


friendship
m
n
(age)
dob
User profile
Username
1
name
Contact
gmail
first
last
Post
Post
Post like
m
n
User Post
images
Video
Post time
Post_id
Post Content
modifytime
time
Postlike id

TATA CONSULTANCY SERVICES

Dec-7

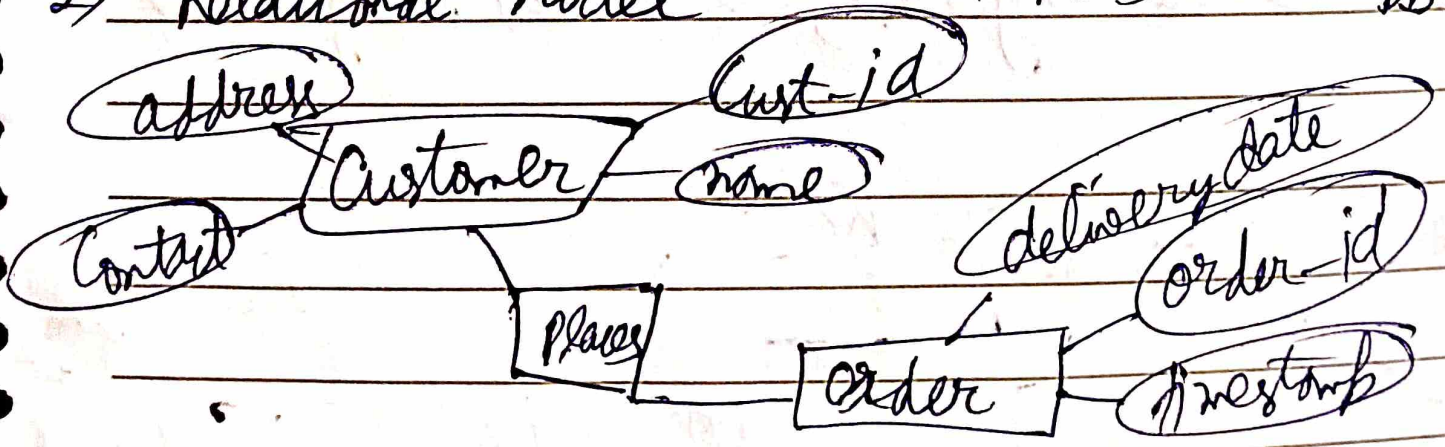
* Relational Model :-

Tables : 7 jo ER model me entity hoti h it become table & attributes become column names like Table :- Customer :-

Cust_id	Cust_name	Contact
1	Aaru	0.0.00.
2	Vip	111.11

→ Degree of tables :- no of attributes = 3
Cardinality no of rows = 2, 2

* When db design it design in 2 steps - 1) first is
1) ER model → ER diagram
2) Relational model → RDBMS - Software DBMS



Now for this ER diagram have to convert it into Relational Model.

Each Entity become a table &
every attr. of entity become colum

Relational model of Customer ER diagram is
Table :-

① Customer (Cust-id, name, address, contact_no)

② ORDER (Order_id, timestamp, delivery date)

ER diagram ← Steps of db design

↓

Relational model

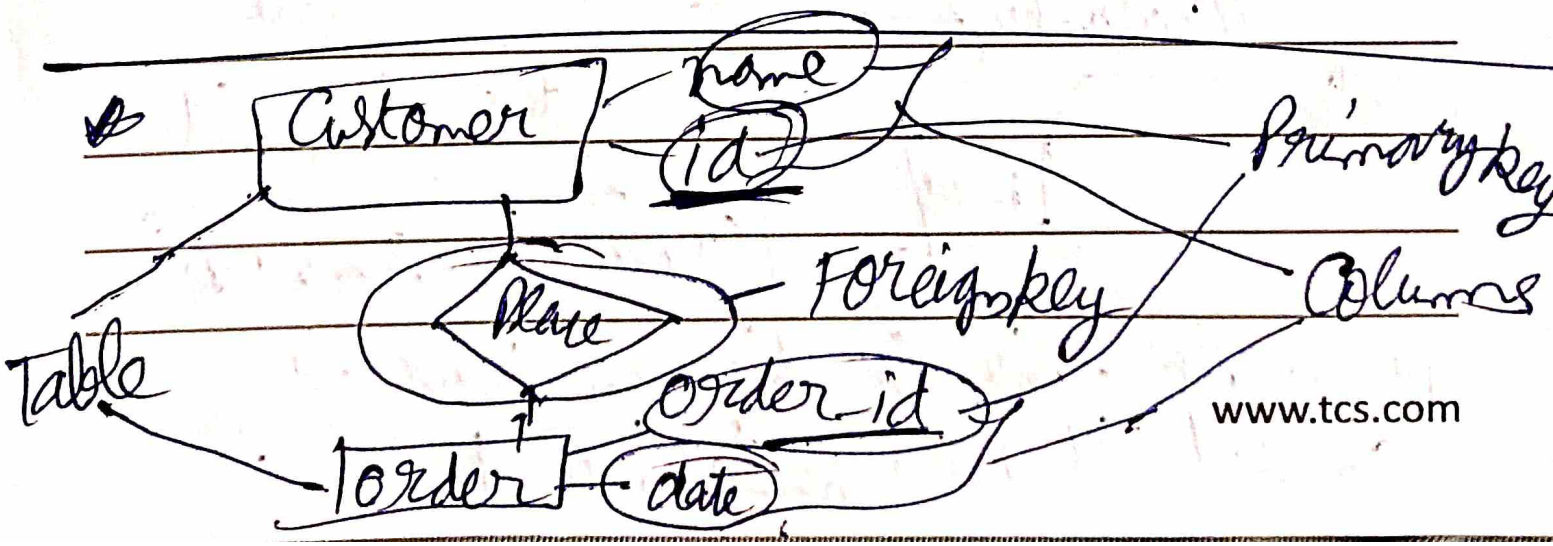
↓

RDBMS → Software DBMS

↓

Software implementation of Relational model

(Software - MySQL, Oracle, PostgreSQL)



* Keys (Relations Keys)

1) Super key → Any permutation and combination of column or attr. Then it will give me uniquely identify combination for ex. cust-id / cust-name / contact / gmail combinations :- (cust-id, gmail), (cust-id, contact), (name, gmail) ... They will give me uniquely combination

2) Candidate Key → is which any redundant value such as name of column ones we remove that redundant key. for ex in Customer table, none we remove.

candidate key :- contact, cust-id, gmail.

CK should n't be null but Super key can be null.

3) Primary key is that selected candidate key which has limited no. of attributes like customer-id.

4) From Candidate key we choose one column as primary key but all other remaining candidate key instead of primary key is known as alternate key
Alternate key = Candidate key - Primary key

5) Foreign key :- it make establish relation b/w 2 tables.

like customer (cust-id, name, address)
order (order-id, delivery date, cust-id (FK))

6) Composite key - PK form of 2 attr atleast.
7) Compound key - PK form using 2 FK
8) Surrogate key - Suppose there are 2 school table.

Reg no	Name
101	Arru
102	VIP

School A Table

Reg no	name
AB101	Koby
AB102	Panda

B Table

Merge

if we merge both the table A & school B table then both primary key of school table A & B now not be pk anymore

Merge

Surrogate_no (Synthetic Primarykey (int value / autoincrement	Reg_no	Name
1	101	Aarti
2	102	Vip
3	AB101	Scooby
4	AB102	Panda

M.Imp

Referential Constraint :- Creates introduce FK

i) insert constraint :- don't insert any tuple / Row in child table unless it get stored in customer / parent table.

2) Delete Constraint - Value can't be deleted in parent table unless data get deleted in child table. (cannot)
→ ON delete Cascade - When applied it delete both parent & child row.

Q) Can FK have Null value?
ON delete Null → Put corresponding FK as null. (when child key have null value) - jb hum parent table me se vo record delete kr diya h pura aur ab hum is child table me se sirf FK ko hi null krna chahte h, puri row ko delete nhi krna chahte tb hum ON Delete Null use karege.

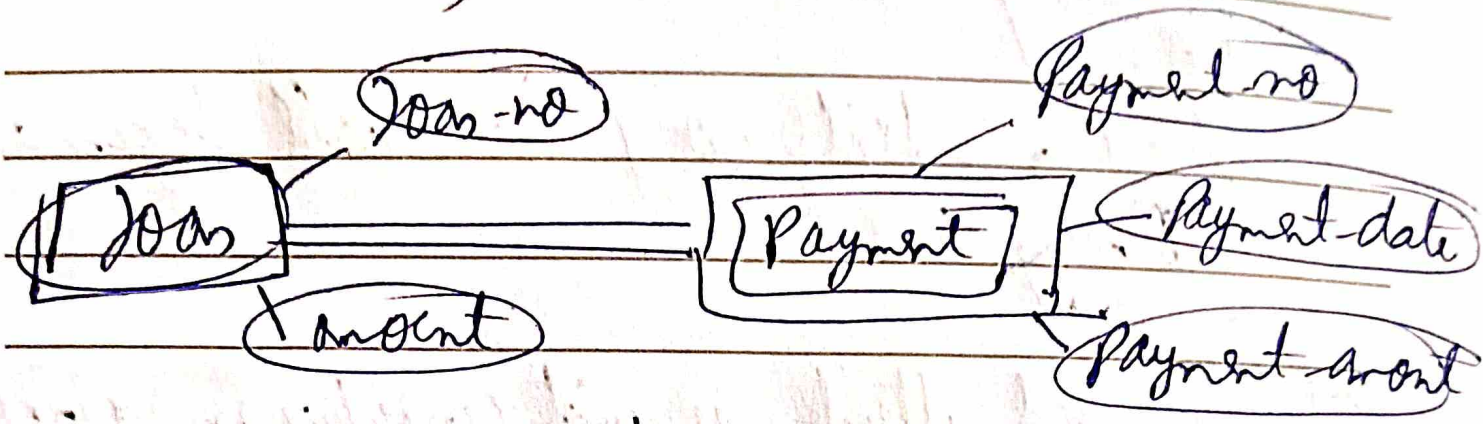
* key Constraints :-
1) NOT NULL

Dec-8

TATA CONSULTANCY SERVICES

1) Strong Entities → Make individual table, attributes becomes colums of Relation. Loan Table:- Loan-no / Amount

2) Weak Entity.



Weak entity Payment Table:-

Payment-no	Payment-date	Payment-amount	Loan-no (FK)

PK of Payment Table: Together they become

3) Composite attributes like location / address → separate attribute for each component

Eg Customer table

cust-name	address-city	add-state	pincode	street-name

(4) Multivalued attributes :- like Contact no
↳ for attr. multivalue make new relation / table. eg make new table of contact no
emp-id / contact no
FK

(emp-id, contact no) Both will become Pkey.

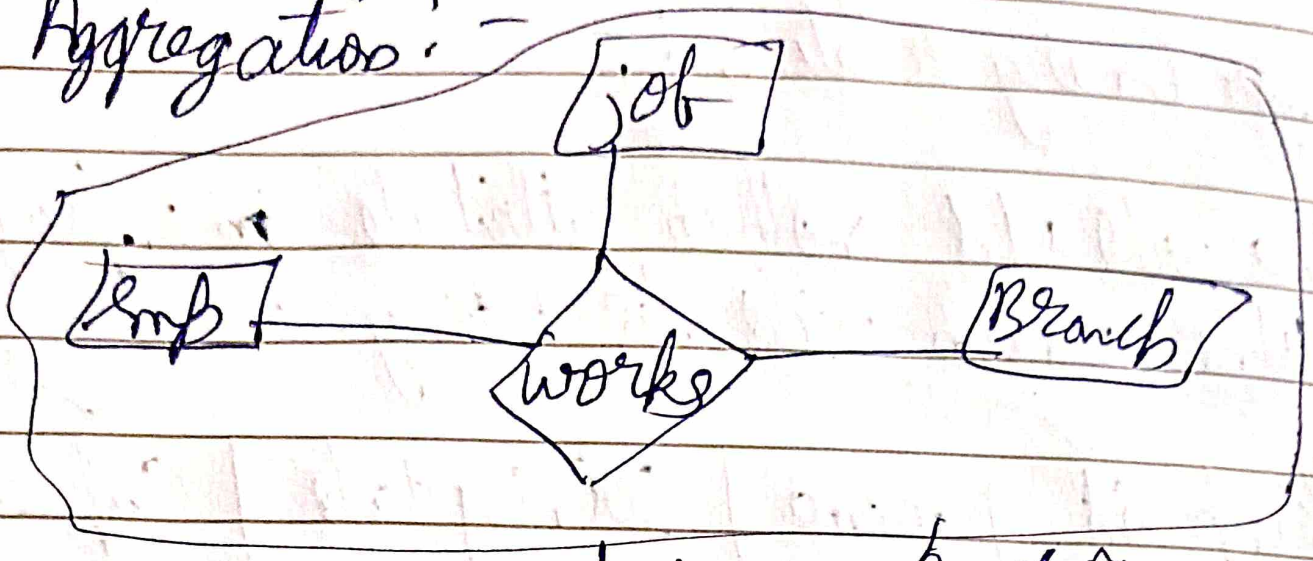
(6) Derived attributes not consider in tables.

* How to show Generilisation from ER diagram into Relation Table :

1) Generalisation do types se hoti h method 1 & method 2 ; in method 1, eg account table Banegi then aag se uski IS-A relation vali table Saving & current account ki aag se table Banegi &
Second method me sirf saving & current table Banegi lekin iske kuch drawbacks rahbi Rahar aaten.

TATA CONSULTANCY SERVICES

* Aggregation :-



↓ iss pure ko ek Manager manage karega

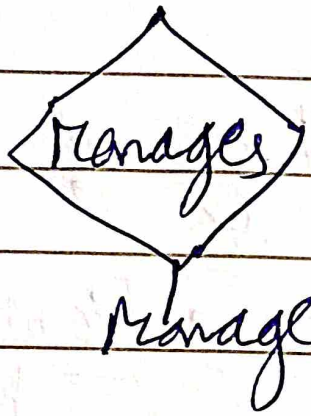
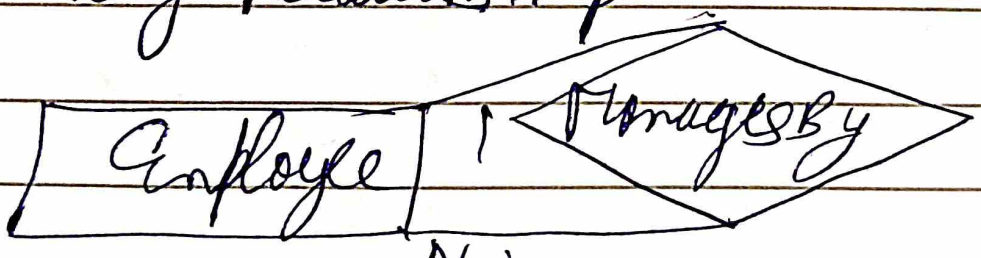


Table manager (mgr-id, branch-id, emp-id, job-id) - iss sbko ki primary key bna dega. mila ke sb ko

* Unary Relationship



__/__/____

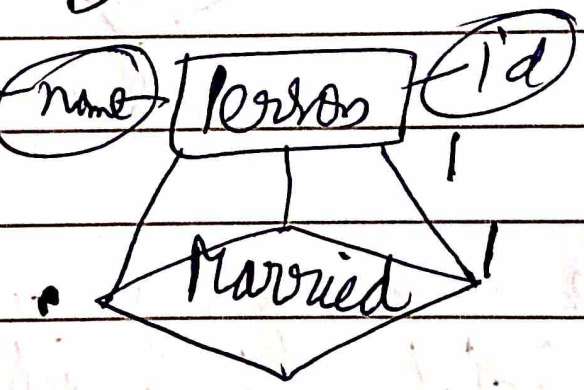
for Unary relation :-

We will add another attribute in employee table ; which will be F key.

P key				F key
Emp-id	name	joining date	...	Emp_mgr_id
201	asru	.....		205

Emp_mgr_id is F key of Emp-id itself.

Eg :- One to One :



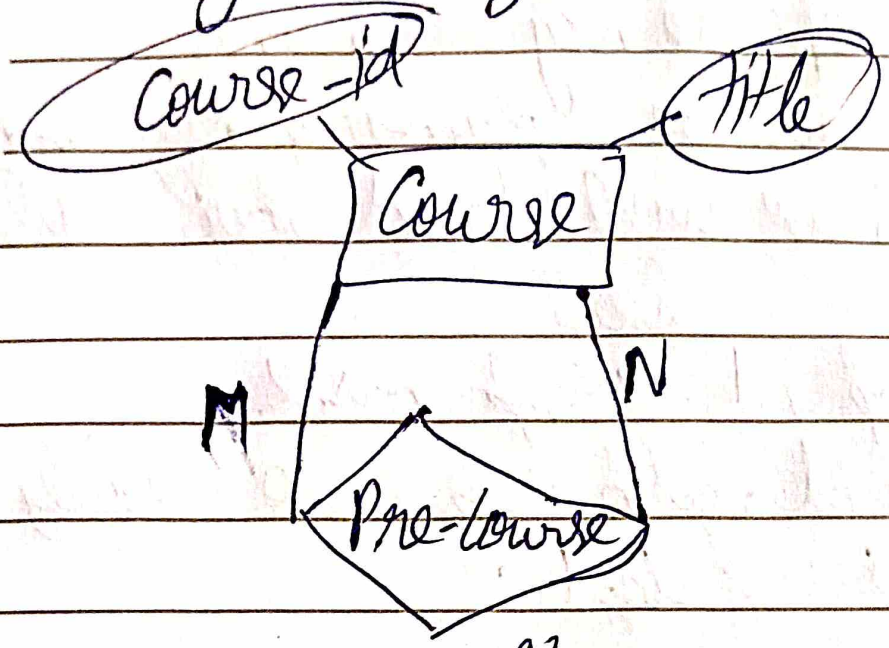
Table

Person (id, Name, spouse-id) → attributes

P key, S key

TATA CONSULTANCY SERVICES

Many to Many.



1) Course Table (id, title)
PKey

2) pre-course Table (id, pre-course-id)
↓ FKey ↓ FKey

By Combining (id + pre-course-id) in pre-course Table
↓ primary key.

TATA CONSULTANCY SERVICES

Dec-11 ;

To Remove Redundant data

Normalisation

functional dependency :- means that there will be primary key ~~who~~ from which we will extract other details

fetch

FD :- Emp id → Emp-name ;

from emp id, fetch emp name solely

emp_id → dept.

primary key (determinant) → other keys / Columns (dependent)

① Trivial FD :- (A → B), B is subset of A

eg :- (emp id, Name) → emp id & Name

name find krte h & vice versa

means A is → B.

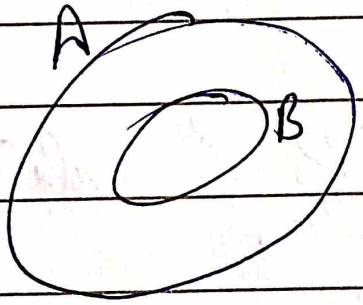
② Non-trivial FD :- {Emp-id, name} → {emp_address}

if (emp-id & name) gives (determinant) then value of (emp-address (dependent) value can be fetched. so $A \to B$, B is no subset of A.

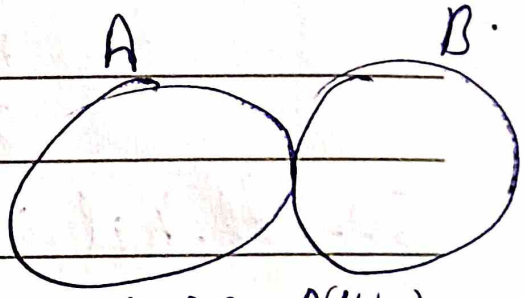
$A \to B$, $B \not\subset A$, $A \cap B = NULL$

eg :- Emp-id → Emp-dept so it is NTSD

Venn diagram TSD
Trivial



Non Trivial



$A \cap B = NULL$.

* - Rules :- 1) Reflexive :-

$X = \{a, b, c, d, e\}$.

Subset $Y = \{a, b, c\}$

Y is subset of X, $X \to Y$

2) Augmentation :- $X \to Y$.

$(XZ) \to (YZ)$.

R(X Y Z)
~ 3 attributes

if we add one attri such as z with X & Y then it will also able to find value of YZ

3) Transitivity :-

$A \rightarrow B$ & $B \rightarrow C$, $C \rightarrow A$

(Ques) R(A B C D E) tell valid or not.

all attri.

Points :- $A \rightarrow B$

$A \rightarrow C$

$CD \rightarrow E$

$B \rightarrow D$

$E \rightarrow A$.

So tell that - i) $BC \rightarrow CD$ =? Valid

argumentation

(ii) $EC \rightarrow AC$ -? - Valid

(iii) $BD \rightarrow CD$ =? - Not Valid becoz in points its not given as $B \rightarrow C$; so not Valid

Imp

→ Why to remove Redundancy?

Ans → Becoz Redundancy introduce / causes 3 ~~anomalies~~ anomalies (abnormalities).

So 3 abnormalities are :-

1) Insertion Anomaly :- Eg :- When 2 independent Tables we merge then the ~~one~~ attribute can't get set to another attr like

Table student :-

id	name	age	~~cou~~ Branch code	name
1	Aaru	14	1	CS
2	Vip	19	1	CS
3.	X	18	2	IT.
NULL	NULL	NULL	3	Bio

if have to enter another branch detail but we can't enter name of student to that corresponding branch, so then again after sometime have to come & update it then it all as insertion anolomy.

TATA CONSULTANCY SERVICES

deletion of data cause some unintended loss of data.

(2) Deletion Anamoly :- if deletion happen as
eg - if we delete student detail who study in CS if delete that stud detail then with that detail its Branch also get deleted (which is not correct).

Name	id	age	Branch-code	name
Arun	1	18	1	CS

→ if we delete it then cost deletion anomaly

(3) Updation Anamoly :-
Suppose that HOD get change of CS dept as Q then branch-hod column have to go & update multiple changes in whole db.

Name	id	age	branch-name	Branch-hod
arun	1	18	CS	X - Q change
vip	2	18	CS	X - update to Q

→ When update of single data value require multiple rows to be updated

* What we do is normalisation :-
Table → decompose → into multiple tables
↓
decompose it until it achieve SRP (single responsibility principle)

TCS TATA CONSULTANCY SERVICES

Type of Normalisation :-

⇒ 1NF : 1) Every cell have (atomic value) means can't be further divide like Contact no can't be further get divide.

Eg :- Table Employee.

id	name	Contact no.
1	Aaru	888.
2	Vip	(12, 17)

doesn't contain multiple values so its not in 1NF.

↓ Convert it into 1NF.

id	name	Contact no
1	aaru	888
2	vip	12
2	vip	17

1NF says that values should be atomic, it doesn't say much about repetition, here repetition happens but 1NF doesn't bother it.

⇒ 2NF :- 1) Relation must be in 1NF and not be and it must not be in partial dependency.

⇒ partial dependency means that we have table table R & attr [A B C D.] now, [A.B] — Both together is primary key.

A, B → prime attr.

CD → non-prime attr.

2NF - fully dependent on primary key

So in partial dependency it says as that C can be determined from part of prime attribute which means that C can be alone determined from part of primary key i.e B.

(B → C) this known as partial dependency.

But problem of partial dependency is that Consider table values :-

A	B
1	Null
2	3
Null	4
~~Null~~	~~Null~~

if A & B both together is primary key then A & B can't be null, only case mapped as [1, null], ~~Null Null~~ X this case doesn't exist.

[2, 3], [Null, 4], But if case [1, null] that is B is null then from null B case ; C can't be determined from null value any other value can't be fetched that is why in 2NF values

TATA CONSULTANCY SERVICES

not be in partial dependency

So if we want to convert R(A,B,C,D) table into 2NF form then it would be.

R_1(A B D) - in this from AB → D D can be determined.

R_2(B C) - from B, C can be determined B → C

⇒ 2NF real life example :- Table (Student project)

Stud-id	Stud-name	proj-id	proj-name
1	A	101	IoT
2	B	102	BIO
3	X	103	CSS
4	Y	104	HTML

this ↑ table is in 1NF form that is all values are atomic but not in 2 nf form bcoz this [Stud-id, proj-id] ⇒ both are pkey.

~~Functional~~ partial dependency is ⇒ Stud-id → Stud-name
proj-id → proj-name

but in 2NF form partial dependency is not valid

tcs TATA CONSULTANCY SERVICES

Convert table student project in 2NF form
↓ Break it into 2 tables

→ Student Table.

stud-id (pkey)	stud name	proj-id (Fkey)
1	A	101
2	B	102

→ Project Table

proj-id (pkey)	proj-name
101	IOT
102	CSS

from 1 these tables stud-id determined → stud name (full dependency)
from proj-id, proj name can be determined
proj-id → proj-name (full dependency)

3NF :- Relation must be in 2NF & no transitivity dependency. (non-prime attr. should not find non-prime attr)

TATA CONSULTANCY SERVICES

Suppose Relation R (A B C)

PKey [A]

Functional dependency :- A → B, B → C

it is in 2NF form

A	B	C
a	1	x
b	1	x
c	1	x
d	2	z
e	2	z

Repetitive data

A → prime, pkey. from A, B can be determine it is valid, A → B ✓ but from B (non-prime attr) why C (non-prime attr) determine like B → C is not valid, in above table we see redundancy (repetitive data). So this ↑ table is not in 3NF :-

→ So to change it in 3NF :- decompose it

R1 (A B) R2 (B C)

R1 table

A	B
a	1
b	1
c	1
d	2
e	2

R2

B	C
1	x
2	z

* BCNF (Boyce - Codd normal form)
- it is also known as 3NF strong version
↳ Relation is in 3NF and Functional dependency $A \rightarrow B$, A be superkey.

Suppose :-

Stud-id	Subject	Professor
101	Java	JP
102	C++	CP1
102	C++	CP2
103	CSS	CS

→ So in ↑ table One student enroll in multiple subjects, for each subject, one professor is assigned to a student.
→ multiple professor can teach single subject
→ One professor teach only one subject.
Pkey = [Stud-id, Subject] means
① from [stud-id, subject] → Professor table can be determined

BCNF says that on the right side like professor → subject (This right side will not be prime, attr).

② from professor column → subject can be determine but here subject is primary key subset i.e prime attr so its no in BCNF form. So to make it in BCNF form :-

↓

BCNF form :-

Student Table :-

	Stud-id	Professor-id → fkey
pkey	101	1
	102	2
	102	3

Professor Table :-

pkey	Professor-id	Prof-name	Subject
	1	JP	Java
	2	JP1	Java
	3	C++	CP1

Here not any non prime attr will find prime attr, reverse is possible prime will find non prime attr.