①

Be careful these thing while making P.K
- User flexability
- No of Column
- Data type size

PRIMARY KEY

- Unique
- Not null. either fully or partially
- Primary key use to identify raw uniquely.

FOREIGN KEY (F.K dusre table m b major ho)(PK 2nd hoga)

- Must be a Primary key in another table
- Must match Primary key values
- May be duplicate
- May be null.

• F.K dusre table me P.K honi chahiye

EXAMPLE OF PRIMARY KEY

1.

Roll #	Batch	Deptt	Name	F-Name	GPA
125	2015	CE			
125	2015	CV			
125	2015	CS			

Here is a mixture of three column use as Primary key
P.K should not be partially null.

2.

Roll No	Batch	Deptt	Name	NIC	F-Name	GPA
87	2018	CE				
87	2018	CV				
87	2018	CS				

because every student doesn't remember his roll

EXAMPLE OF FOREIGN KEY:

1. Student_Personal_Info

PK

Roll #	Batch	Dept	Name	F. Name
87	2018	CE		
87	2018	CV		

Parent table

FK

Student_Transcript

PK

Roll #	Batch	Dept	Sem #	GPA
87	2018	CE	1	
87	2018	CE	2	

child table

2. EMPLOYEE

P.K

Emp No	E. Name	Dept No.
1234	Smith	10
1235	John	10
1236	Allen	20

child table

P.K DEPT

Dept	d name	d loc
10	HR	Kar
20	Finance	Lahore
40	Engg	Isb

parent table

Rules Of Foreign Key.

UPDATE:

- Restricted
- Cascaded

Restricted me jb update krain gy Parent table ko to us kuy nh dega jb tk child table me update nh krain gy. (Since no record found error). agr child table me us ka record nh hoga to record update hojay ga (record has been updated)

Cascaded me jb update krain gy parent table ko to wo auto update krdy ga child table me jitny b hongy (Yes; all record in parent as well as in child table is updated).

jb child table me update krain gy to parent me data mojod ho waisa update nh hoga.

DELETE:

Restricted , Cascaded

Restricted me agr record delete krain gy Parents table me se or child me record mojod hy to nh hoga agr child me mojod nh hy to delete hojay ga.

Cascaded me agr delete krain gy Parent table se to us child me se auto delete krdy ga.

Child table me delete hojay ga restricted me be or cascaded no issue

INSERT:

- Insert in Parent table will be possible if primary key will not be dublicate.
- Insert in child table will be possible if data is present table otherwise it is not possible.

ENTITY RELATIONSHIP DIGRAM (ERD)

-ENTITY:

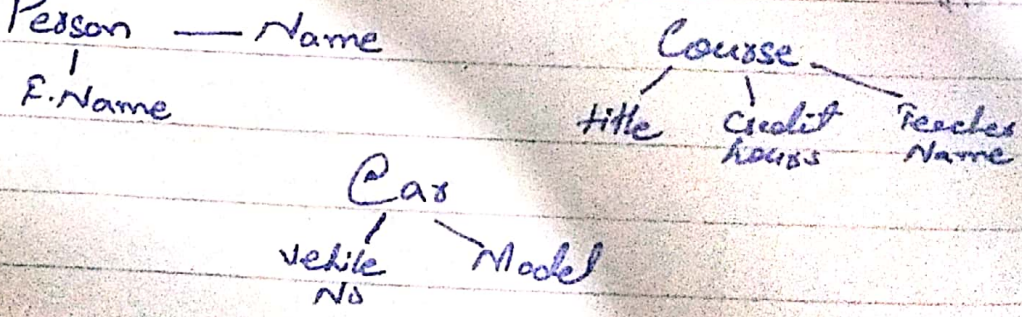
An object which has physical or conceptial existance.

eg: Person, course, car

Entity type: Groups of entites having similar attributes is called Entity type.

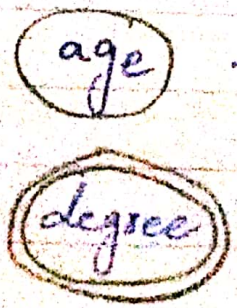
ATTRIBUTES:

Properties of an Entity.



ATTRIBUTES TYPES:

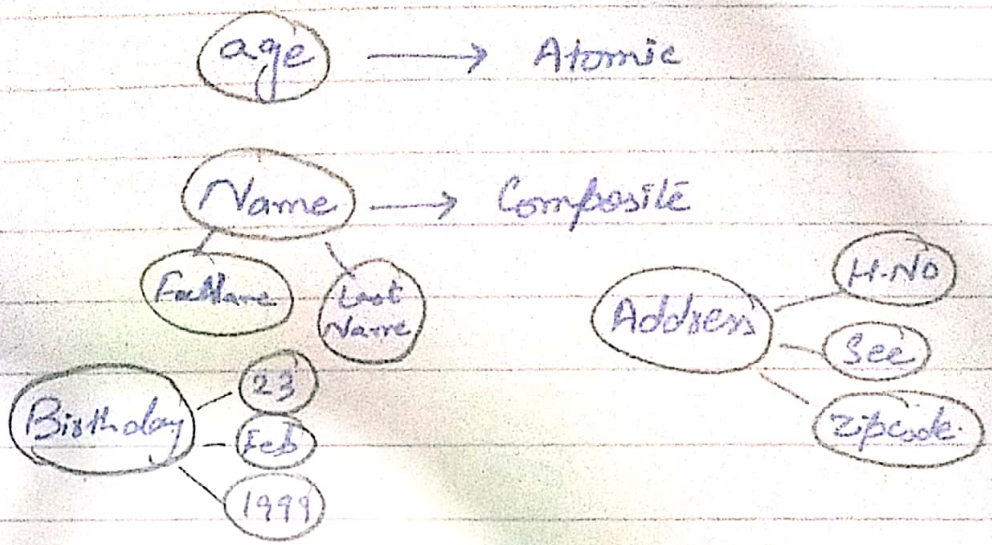
-- Single valued vs Multivalued



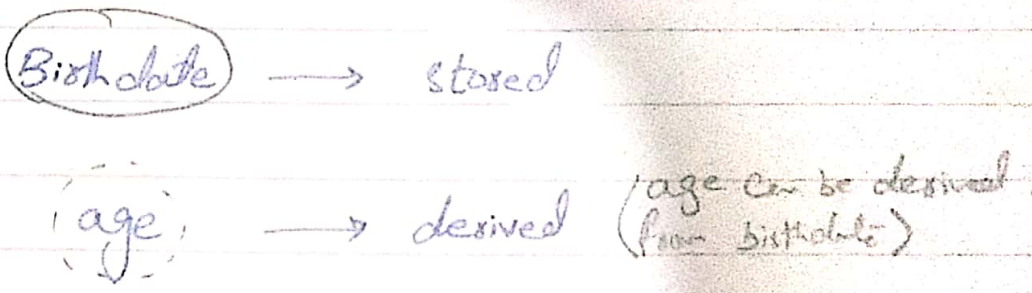
→ single valued

→ multi valued (matric, inter, BS

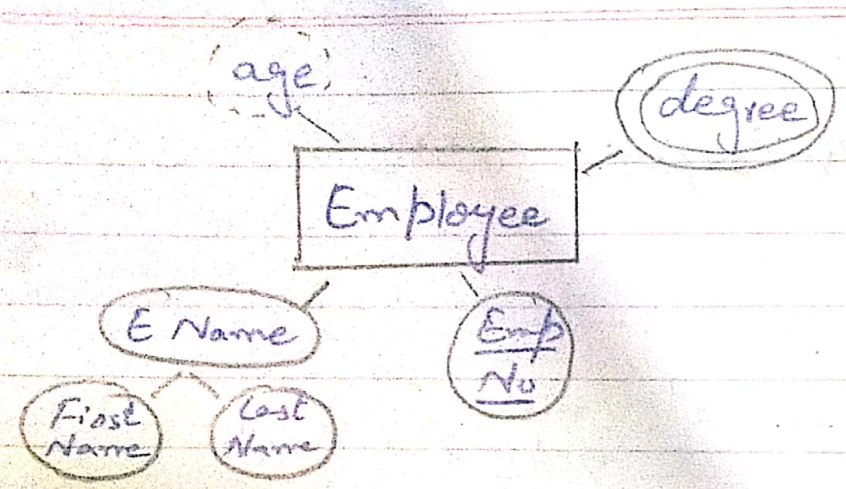
-. Atomic Vs Composite

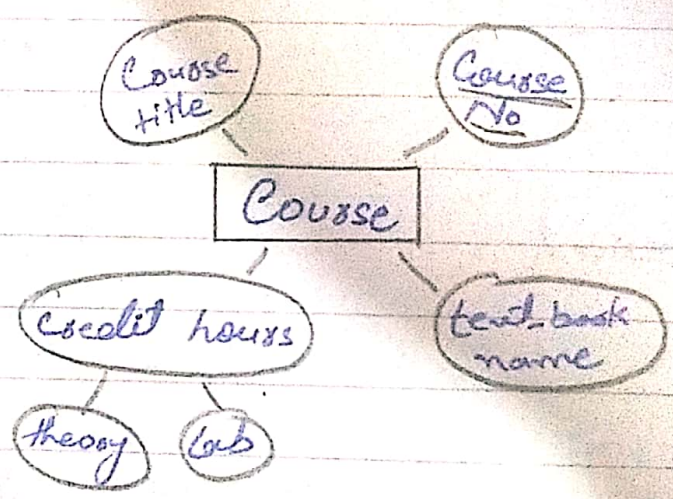


-. Stored vs Derived.



7

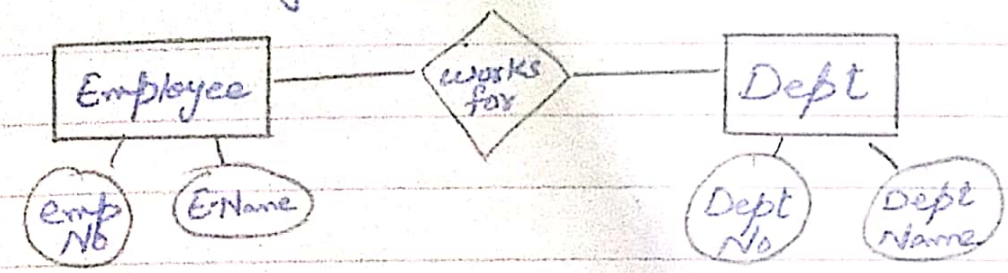


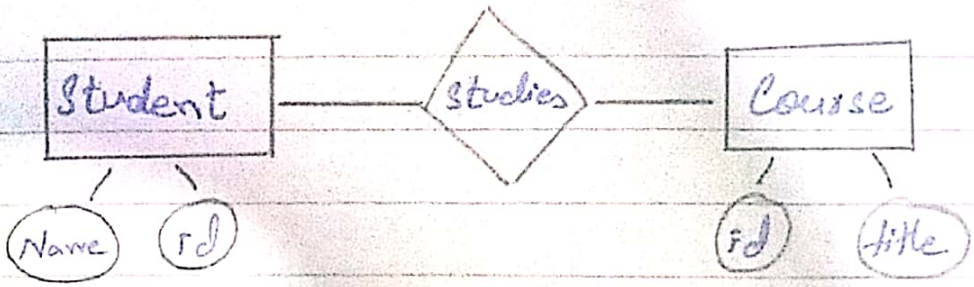


RELATIONSHIP:

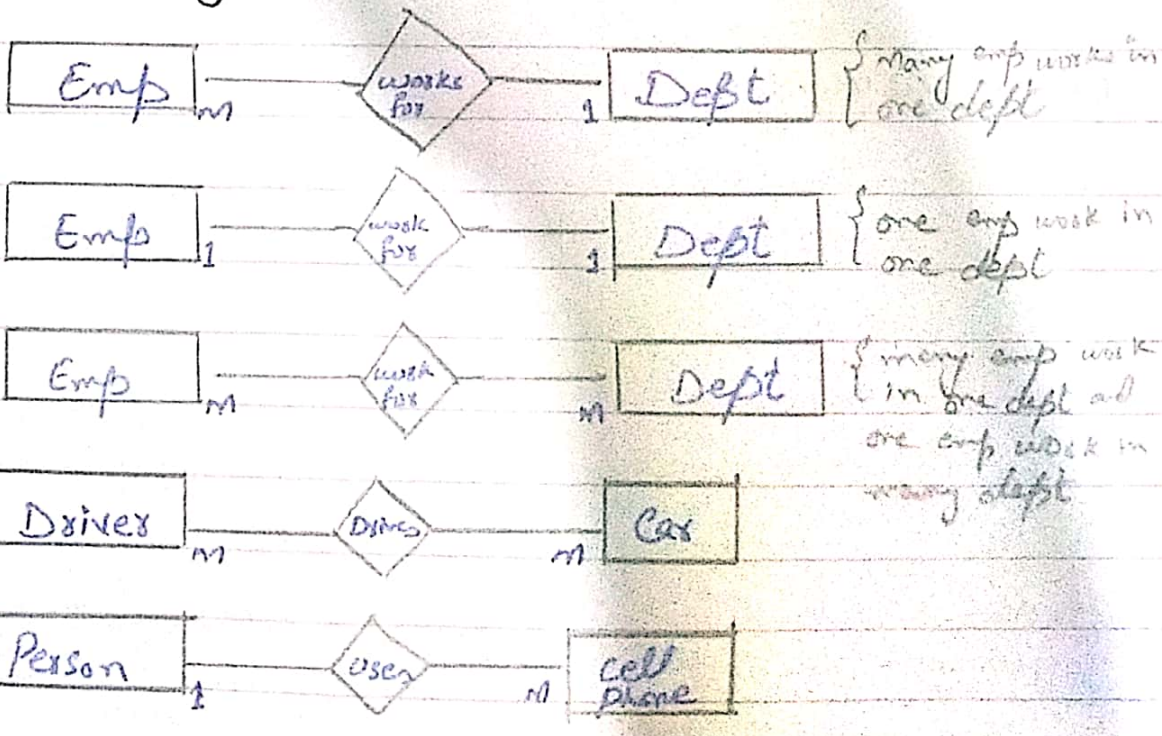
Relationship type

- Employee works for Dept





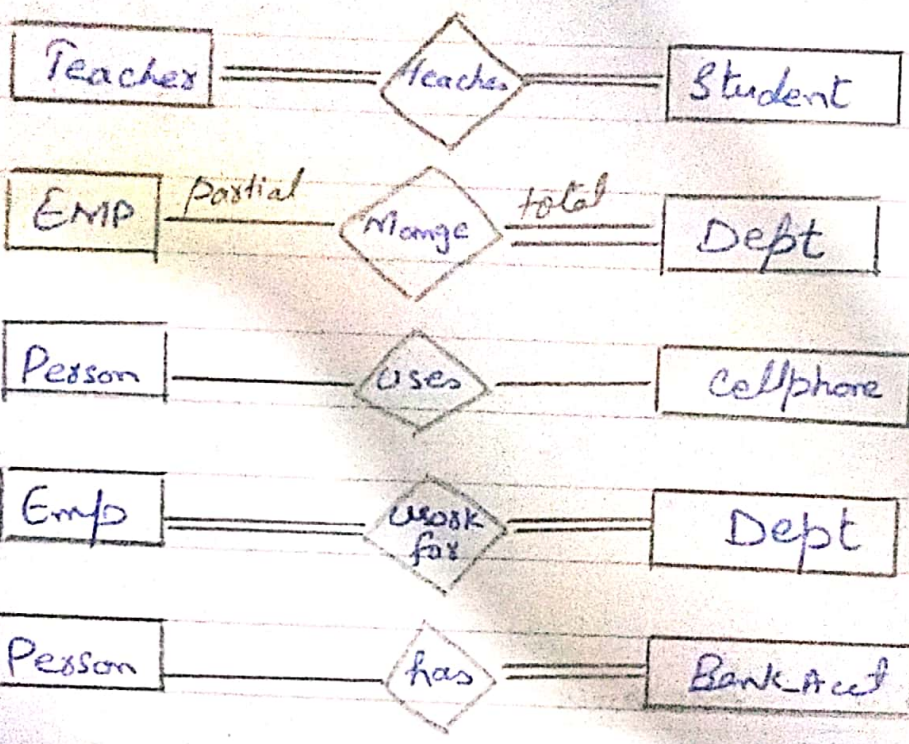
Cardinality Constraints - limitation



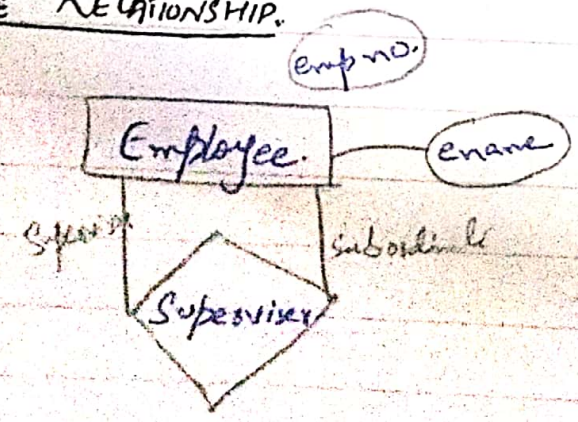
9

PARTICIPATION CONSTRAINTS:

- Total
- Partial

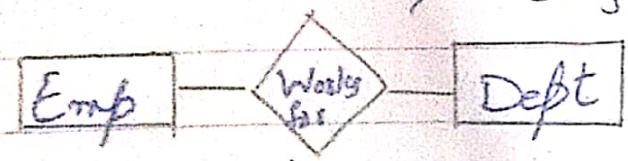


RECURSIVE RELATIONSHIP.



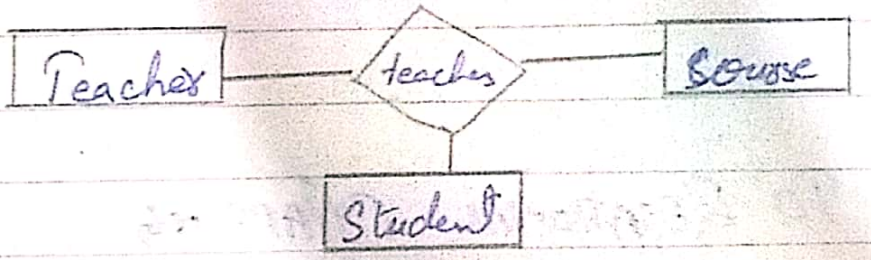
Degree is one
Unary Relationship type
1- ary Relationship type

Relationship Degree.



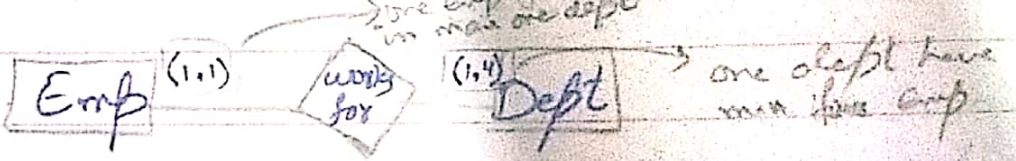
jitni entity hogi
utni degree hogi

Binary Relation type
2-ary Relation type.



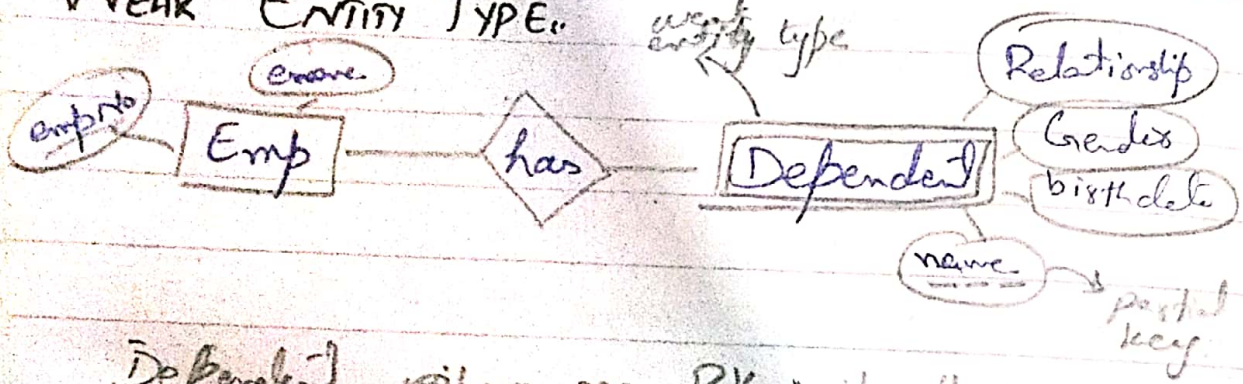
Ternary relationship type
3-ary Relationship type
Degree is 3.

Structural Constraints: (min, max)



ER to Relational Mapping

Weak Entity Type.



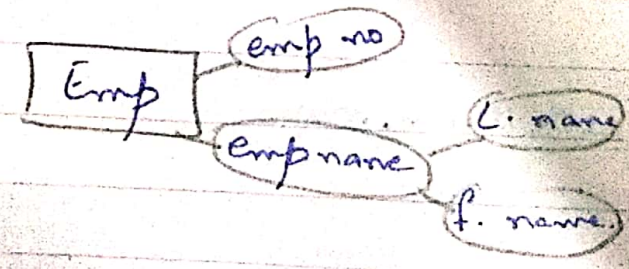
Dependent ~~will~~ have no P.K ∴ it will borrow emp's P.K with dependent partial key

∴ therefore it is weak entity type.

ER To Relational Mapping

ERD Mapping is undeterministic

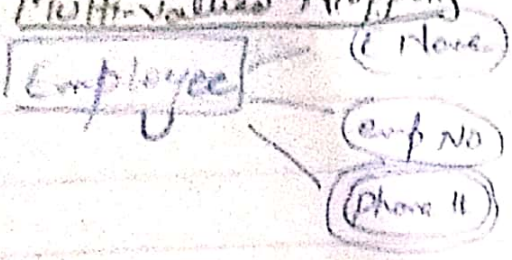
– Mapping of Strong Entity type



Emp

Emp No | Last name | first name

~~Derived~~ Multivalued Mapping



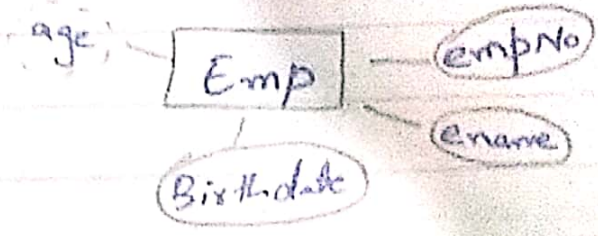
EMPLOYEE

PK EmpNo	E.Name
1234	AU

EMPLOYEE PHONE

FK EmpNo	Phone# PK
1234	021-345..
1234	0322-456..

DERIVED ATTRIBUTE MAPPING



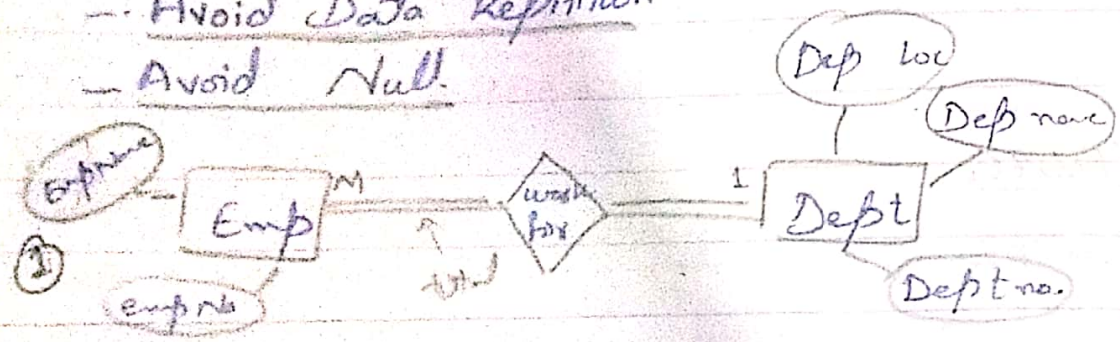
EMP

empno	ename	Birthdate

13

Mapping Of Relationship Type:

- Avoid Data Repitition
- Avoid Null

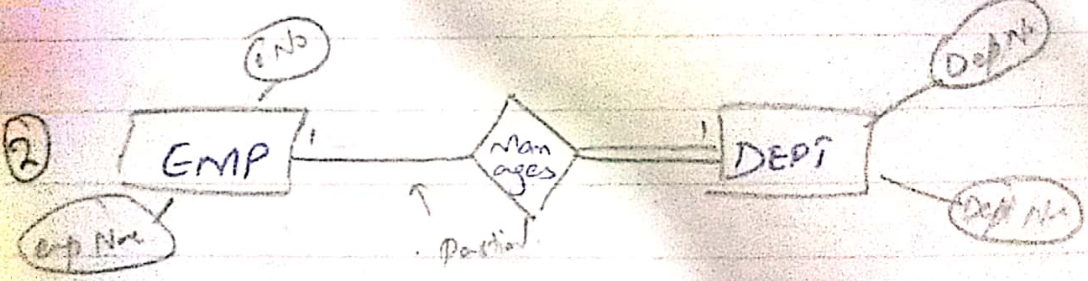


EMP

Emp No | Emp Name | Dept No (F.K)

DEPT

Dept No | Dept Name | Dept Loc



EMP

Emp No | emp Name

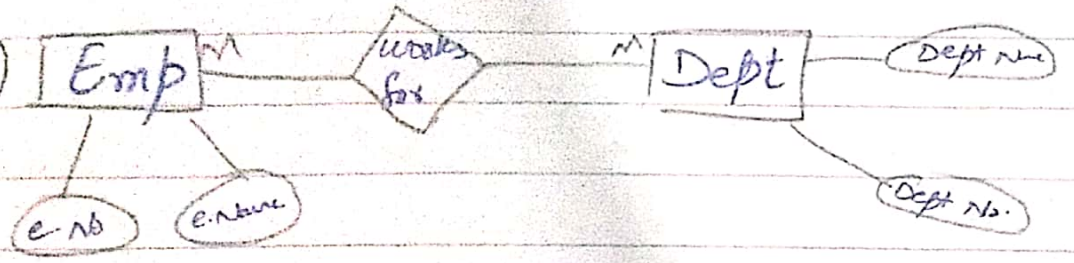
DEPT

Dept No | Dname | EMP No (F.K)

if both side data is existing then draw new table

if both side data is null then draw new table

(3)



EMP P.K

Emp No Emp Name

Dept:

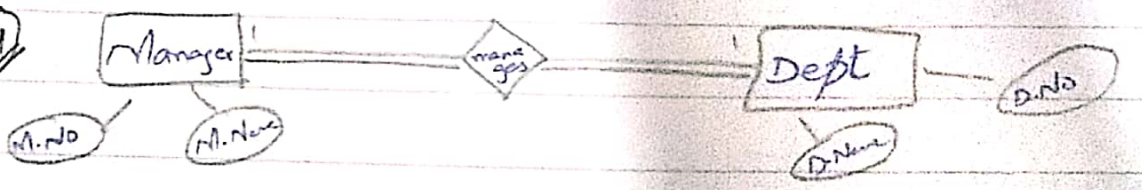
Dept No P.K Dept Name

Now

WORKS_FOR

Emp No Dept No.

(4)



DEPT

P.K D.Name D.No M.No

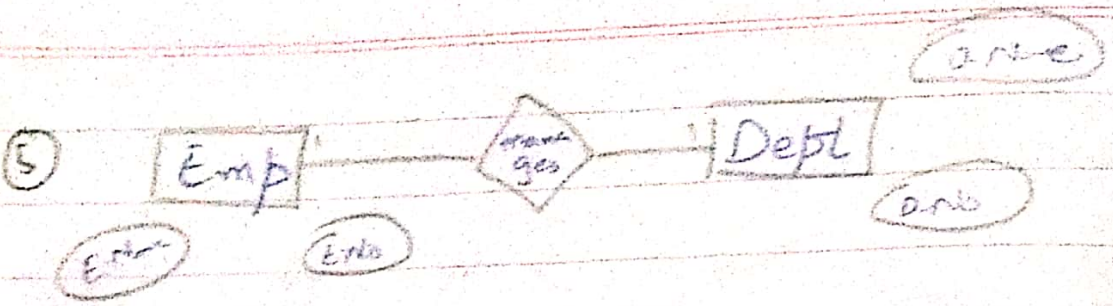
F.K

MANAGER

P.K M.No M.Name

15

5



EMP

P.K

e.no e.name

Manages

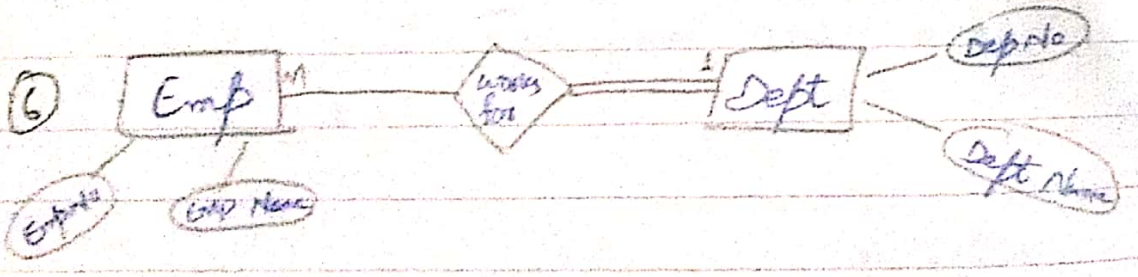
e.no deptNo.

DEPT

P.K

D.No D.Name

6



EMP

emp no ename

Works for

Emp no Dept no.

Dept

dept no dept name

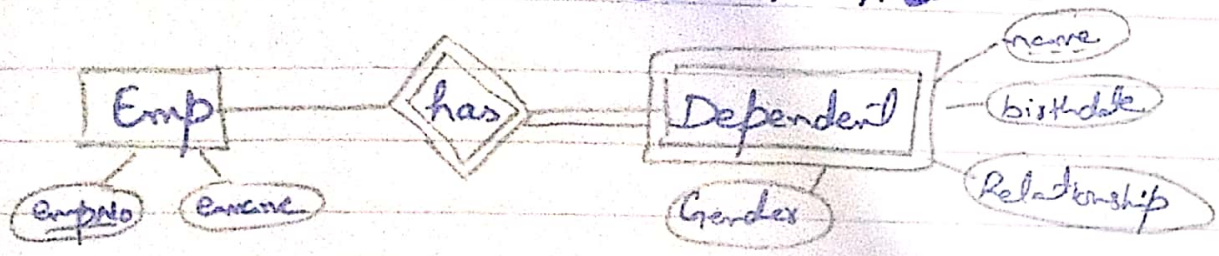
Conversion:

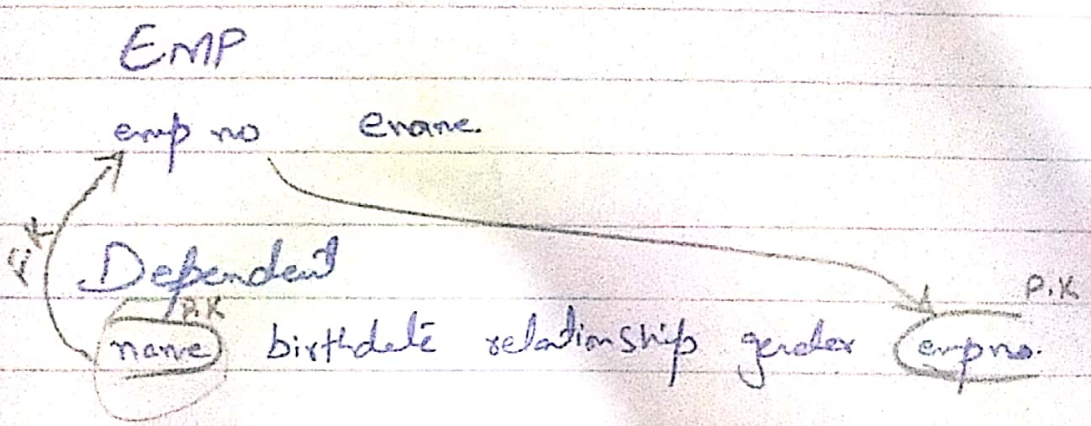
M:M => new table

Partial-Partial => new table

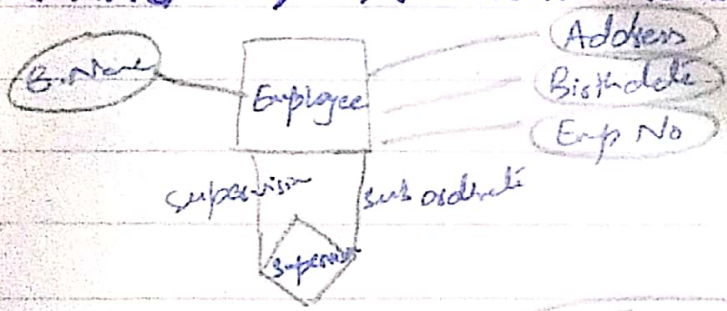
weak entity ki p.k nh hoti

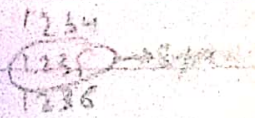
MAPPING OF WEAK ENTITY TYPE:





MAPPING OF RECURSIVE RELATIONSHIP TYPE:



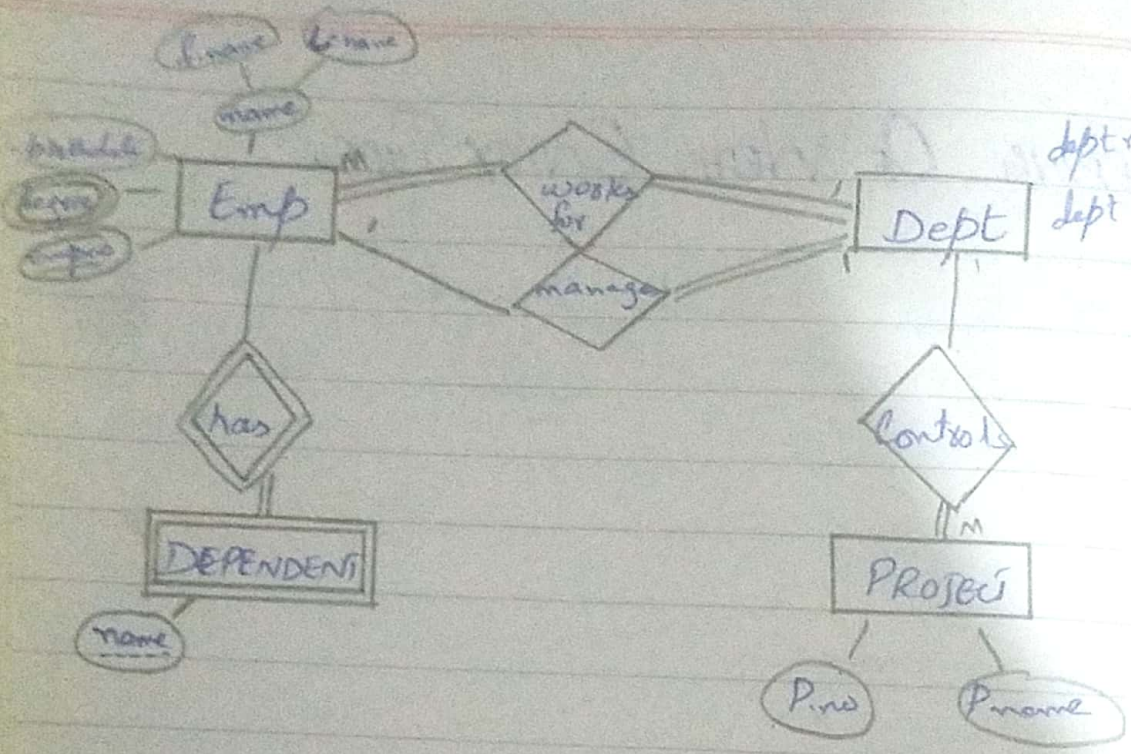


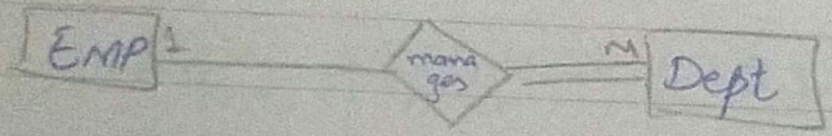
EMPLOYEE:

F.K

P.K emp no	ename	birthdate	address	supercomp no.
1234	Smith	3/0/xxx		
1235	John	6/7/2001		
1236	Allen	6/9/xxxx		1235

17



① 

1 emp May Dept manage

EMP

Emp no Ename

F.K

DEPT

Dept no Dname Emp No

② EMP manages DEPT

EMP

Emp no Ename

Manages

EMP No Dept No

DEPT

Dept no Dname

F.K

partial => null ce skte hy
Many => Data replication

* Many wali me one wale jy ge

ER to Relational Mapping

Summary:

- Mapping of strong Entity
  - Single vs multivalued
  - Atomic vs composite
  - Stored vs Derived

- Mapping of Relationship type
  - 1 : 1
  - 1 : M
  - M : M

- Mapping of weak Entity type

- **1 : 1** (new table)
  - total : total → kese b side move krwa skte gy
  - total : partial → jis taraf null nh hogi us taraf P.K chali jay gi (total ki taraf jay maybe)
  - partial : partial ← new table

- **M : M**
  - total : total
  - total : partial
  - partial : partial

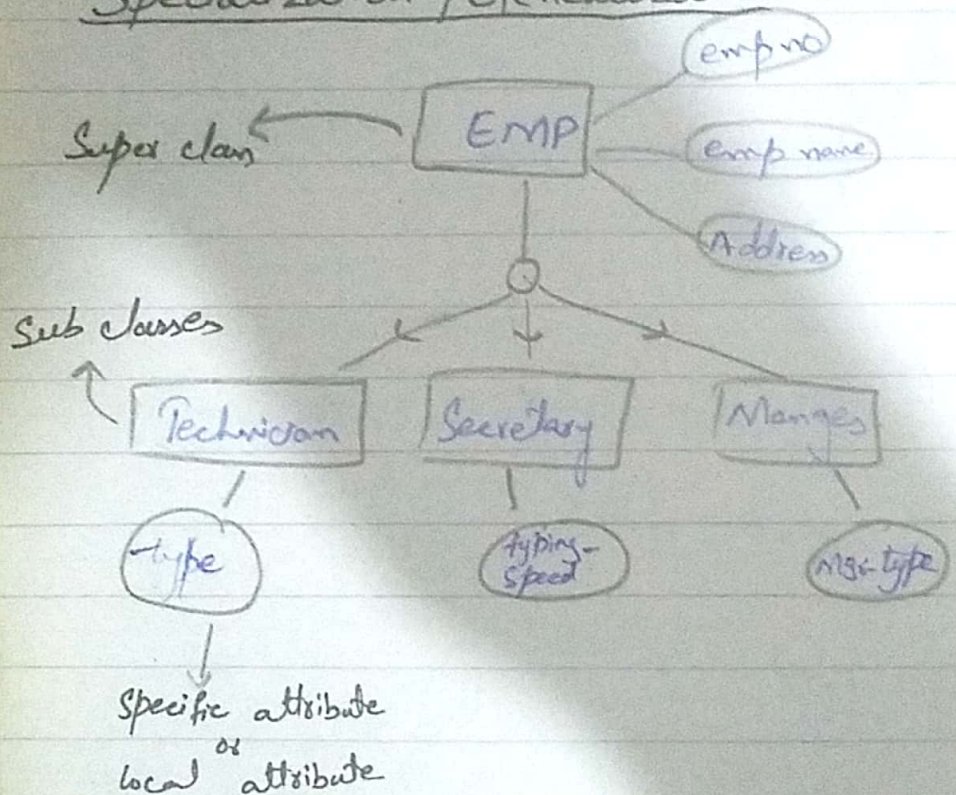
  } => new table

- **1 : M** (new table)
  - total : total → null nh hoskta, jaha pey many hoga waha jay data refeed hoga 1 wale many waly table me jay ge
  - total : partial
  - partial : partial ← new table → bht ziada sod k

Extended Entity Relationship Diagram (EERD)

top to bottom ↑

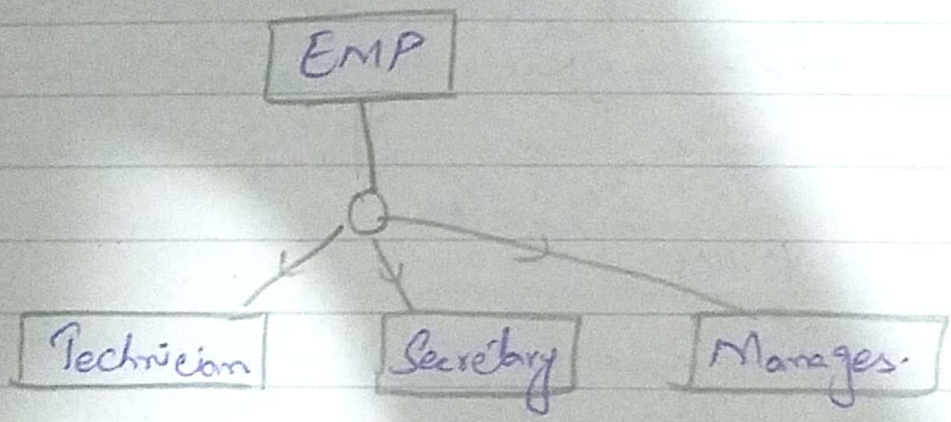
Specialization / Generalization → bottom to top



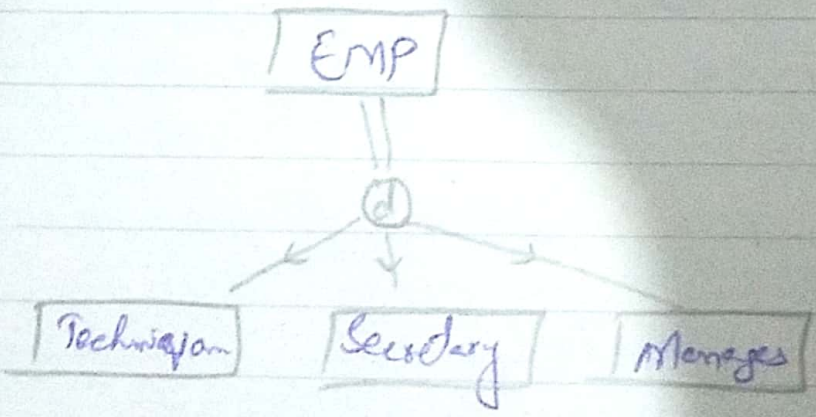
DISJOINTNESS CONSTRAINTS.

d: disjoint => mein 1 sub class join kre ga

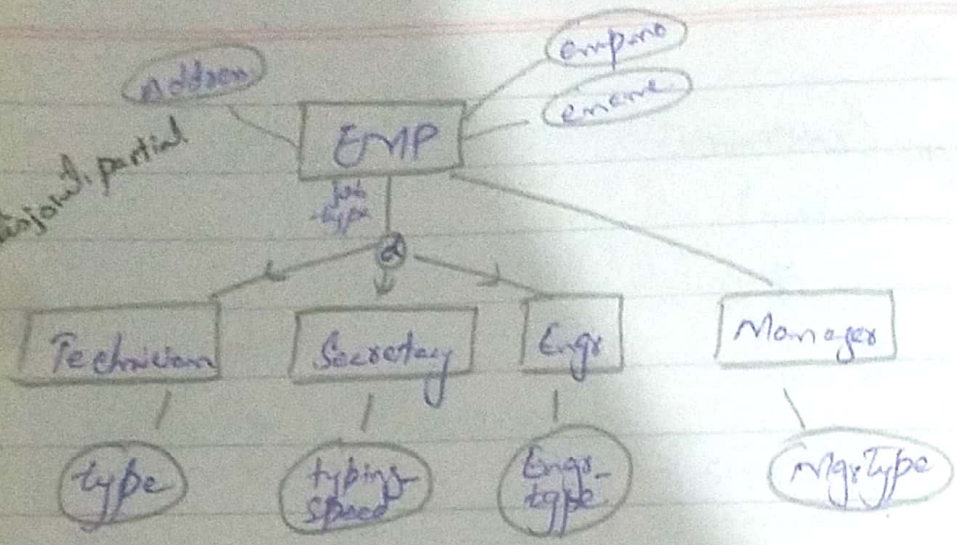
O: overlapping. => 1 se zicde sub class join krsktay

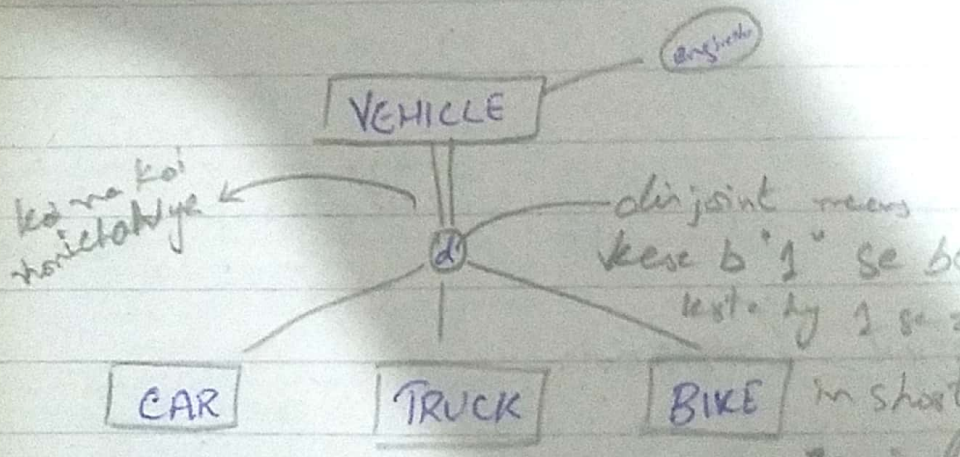


PARTICIPATION CONSTRAINTS.



21

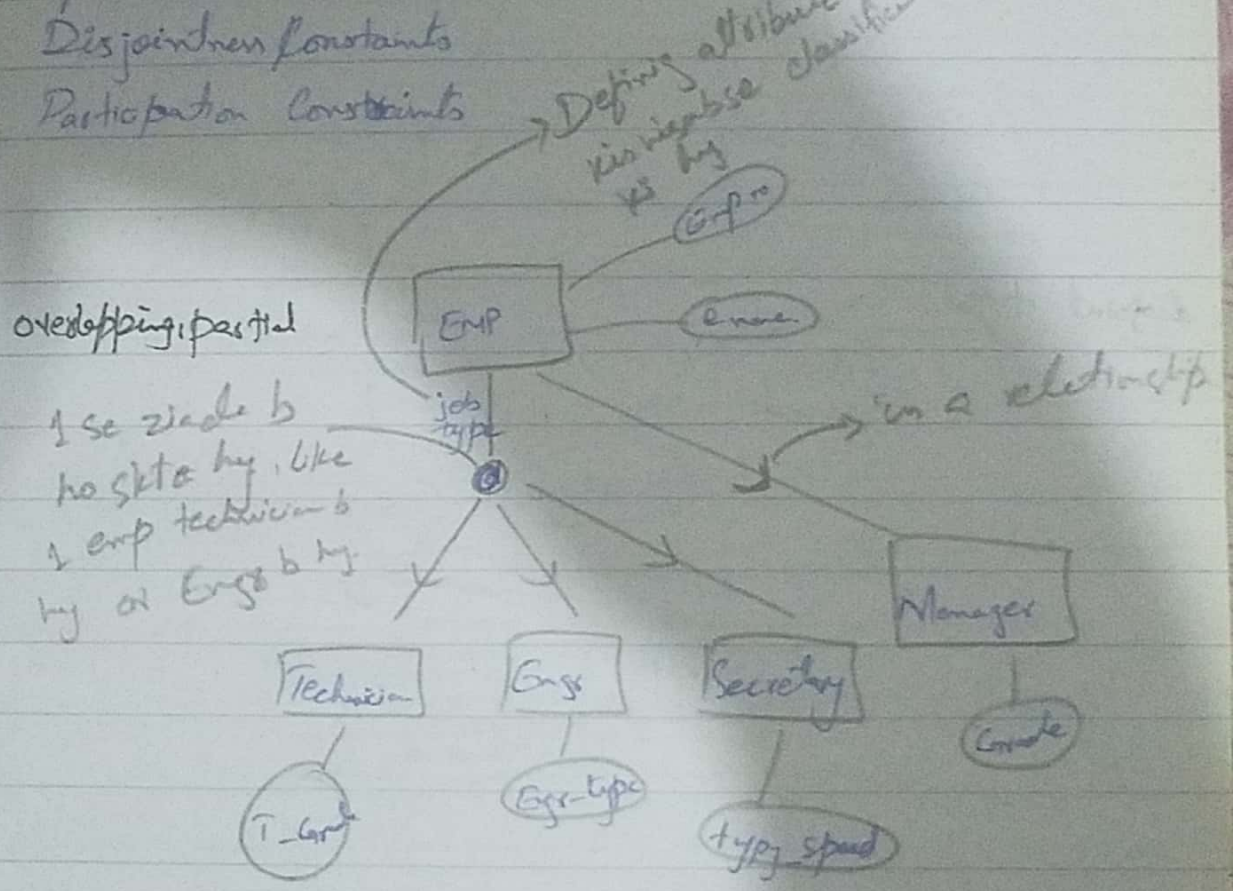




Scanned with CamScanner

Enhanced Entity Relationship OR Extended Entity Relationship Diagram

Super class / Sub class
Specialization / Generalization
Disjointness constraints
Participation Constraints



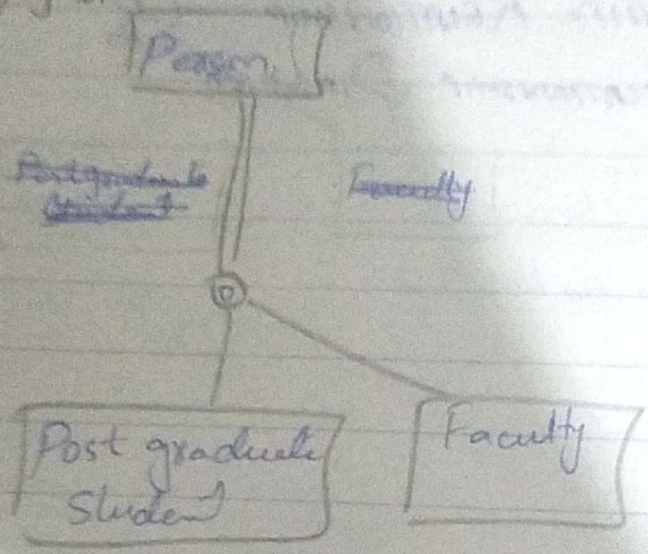
overlapping / partial

1 se zyada b ho skta hy, like 1 emp technician b hy or Engr b hy

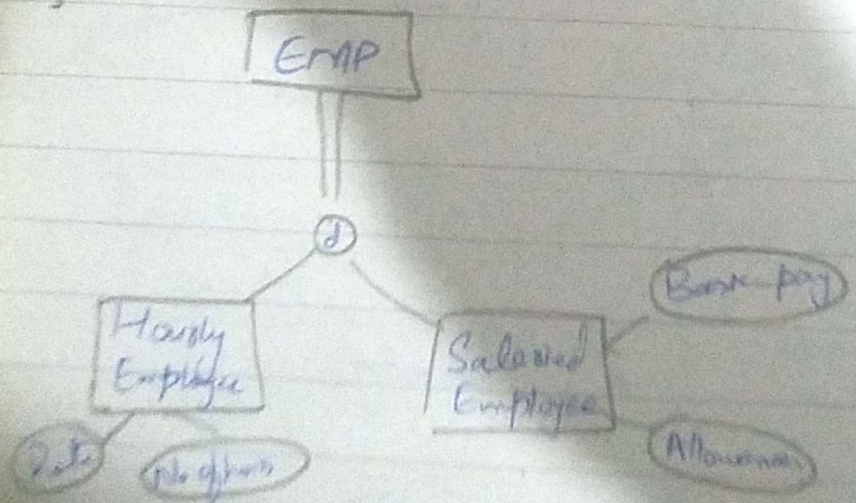
is a relationship

Emp is technician, Engr, Secretary ⇒ Specialization
technician, Engr, Secretary → Emp ⇒ Generalization

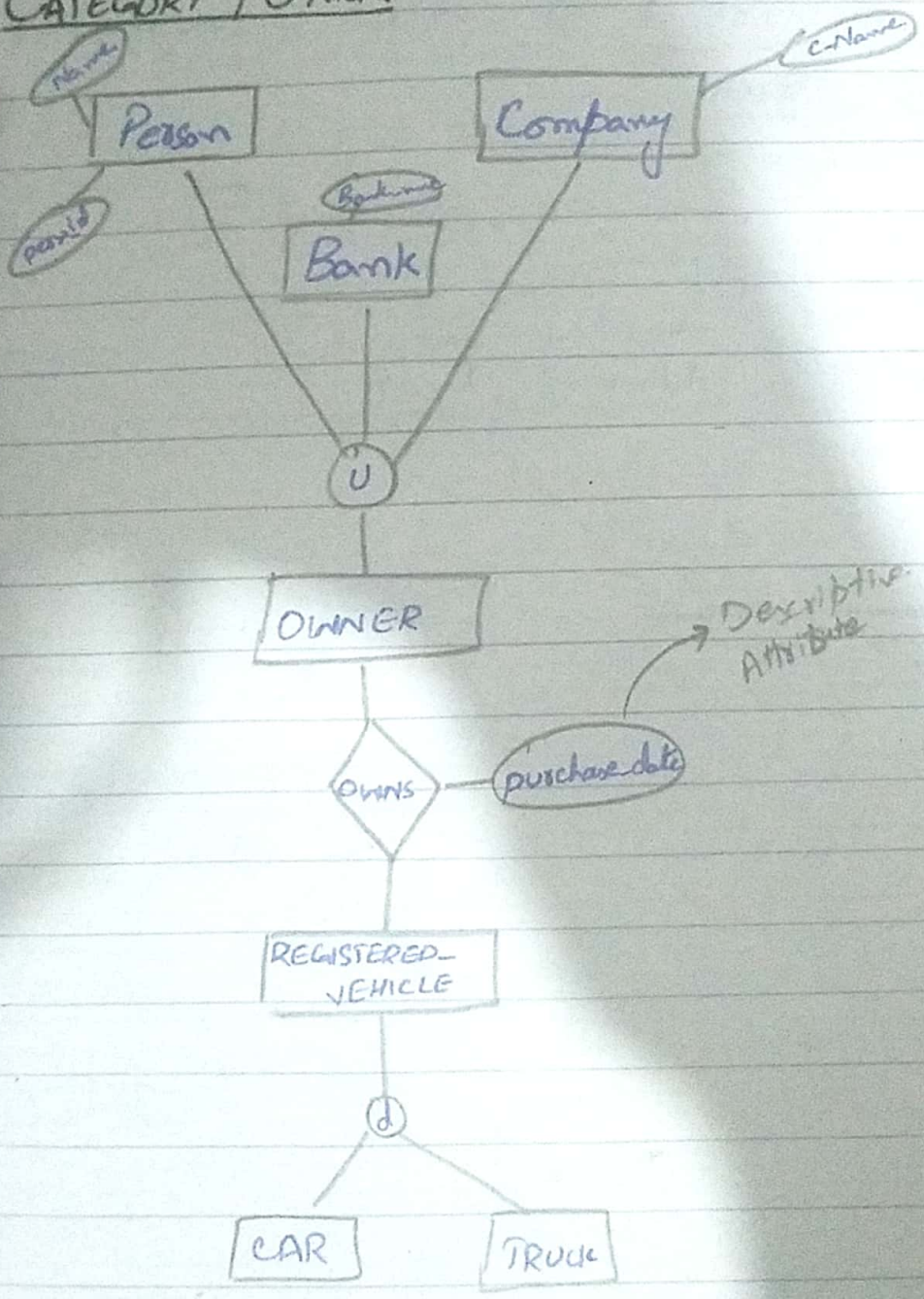
overlapping and



disjoint total.



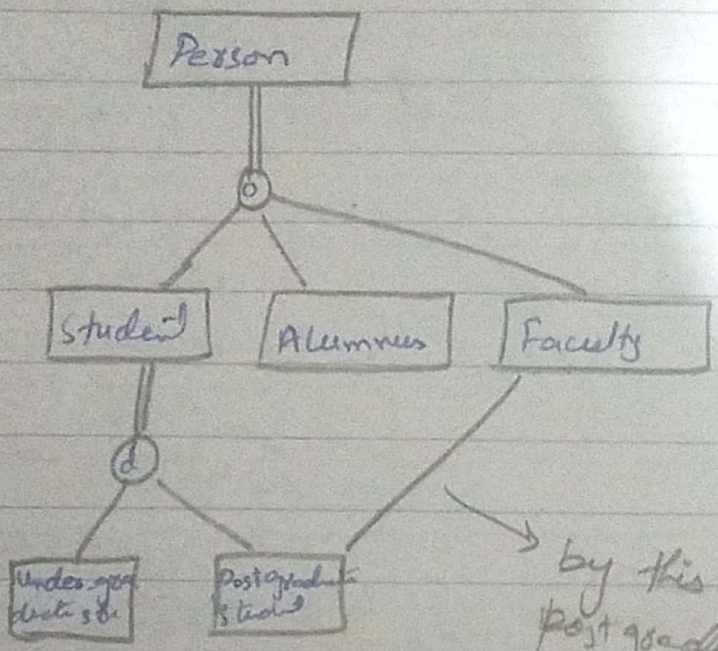
CATEGORY / UNION.



25

HIERARCHY / LATTICE.

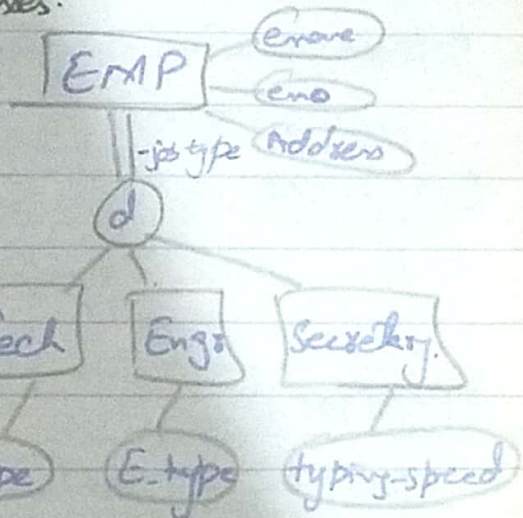
HIERARCHY: → no shared subclass should exist.



by this line postgraduate student become shared subclass so it become lattice

EER To Relational Mapping.

Option #1. Multiple Relations: (disjoint or overlapping / total or partial) both
Super class & subclasses.



EMP

Emp no Emp Name Address.

TECHNICIAN

Emp No T_Type

ENGINEER

Emp No E_Type

SECRETARY

Emp No Typing_Speed

Option #2: Multiple Relations. When Subclass are total (not partial)
- Subclasses Relation only.
Super class k sare attribute
Sub class re kr dain gy

TECHNICIAN.

Emp no Ename Address T_type

ENGINEER.

Emp no Ename Addoess E-type

SECRETARY.

Emp no Ename Addres Typing-speed

Option #3: Single Relation. {Subclass are disjoint}
with one type attribute.
it geverates null values.

EMPLOYEE

empNo ename Address Jobtype T_type E_type Typing-speed

P.K File
Concealed
Restricted

Option #4: Single Relation with Multiple type boolean Attribute. {Sub classes are overlapping as well as disjoint}

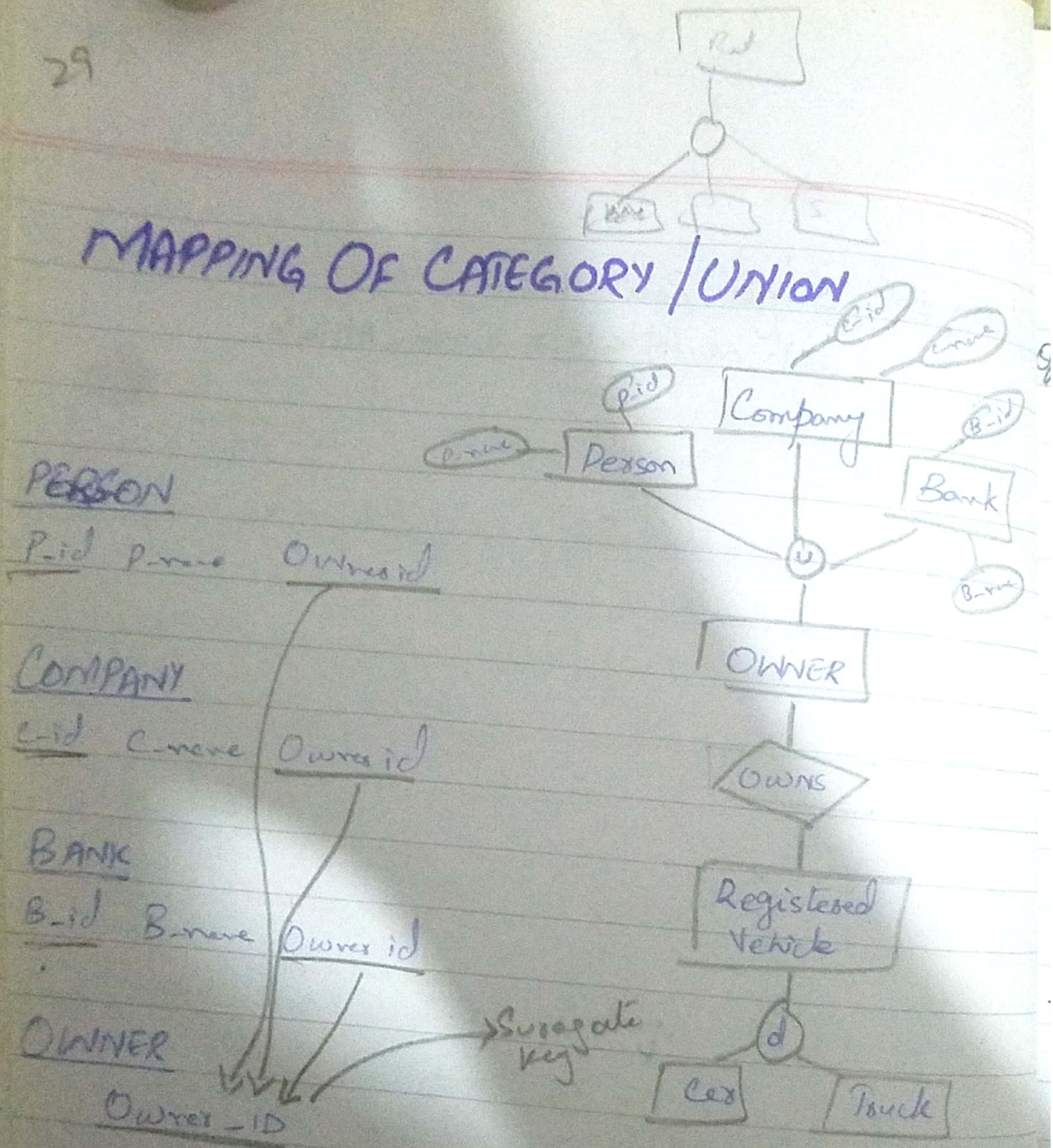
{it also generates null values}

EMP

empNo empName Address T_Flag E_Flag S_Flag T_type E_type Typing Type

Agr 1 bnda teachr ho hy or Engineer ho to T_Flag or E_Flag on hr k entry kr dein gy.

MAPPING OF CATEGORY / UNION



PERSON

P_id P_name Owner id

COMPANY

C_id C_name Owner id

BANK

B_id B_name Owner id

OWNER

Owner_ID

Surrogate key

The company is organized into departments. Each department has a unique name, a unique number, and a particular employee who manages the department. We keep track of the start date when that employee began managing the department. A department may have several locations. A department control ~~several~~ number project, each of which has a unique name, a unique number and a single location. We store each employee's name, Social Security number, address, salary, Gender and birthdate. An employee is assigned to one department, but may work on several projects, which are not necessarily controlled by the same department. We keep track of the current number of hours per week that an employee works on each project. We also keep track of the direct supervisor of each employee (who is another employee). We want to keep track of the dependents of each employee for insurance purpose. We keep each dependent's first name, Gender, birth date, and relationship to the employee.

DEPT

EMP manages DEPT

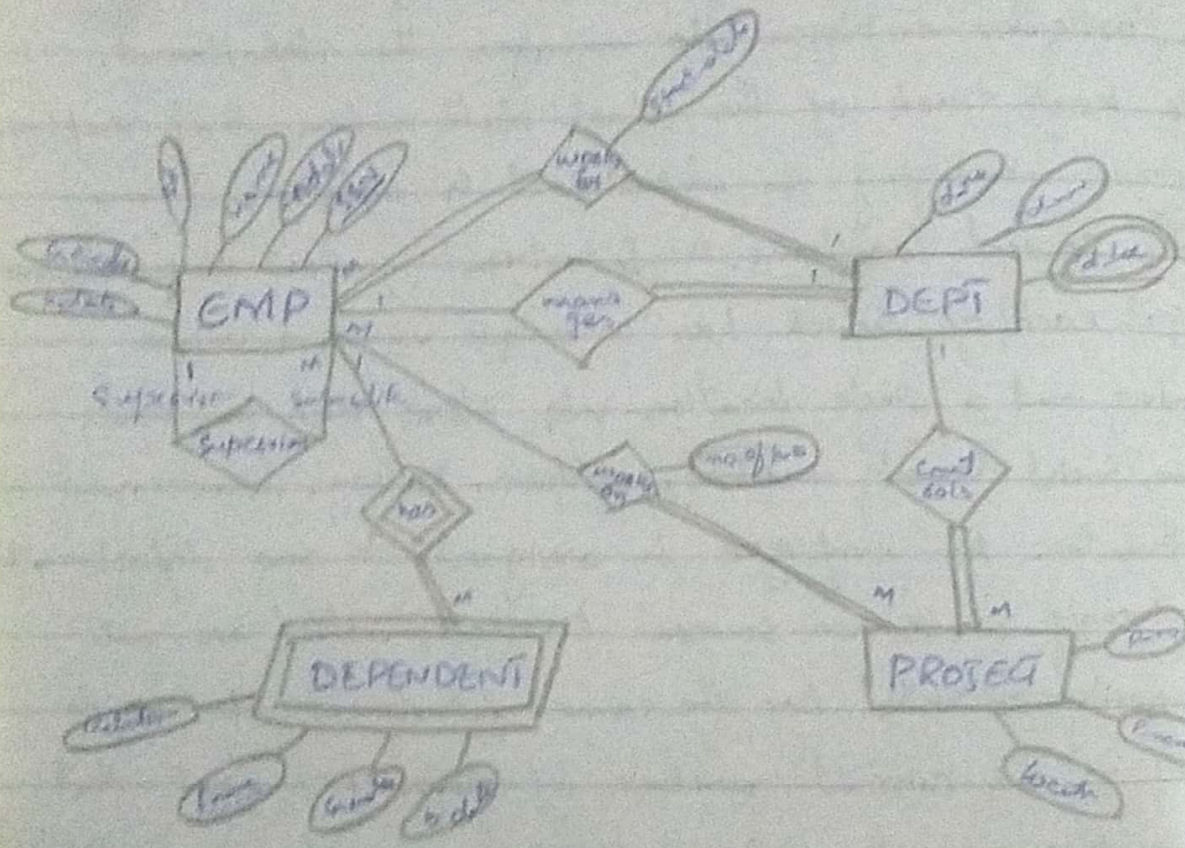
DEPT Control PROJECT

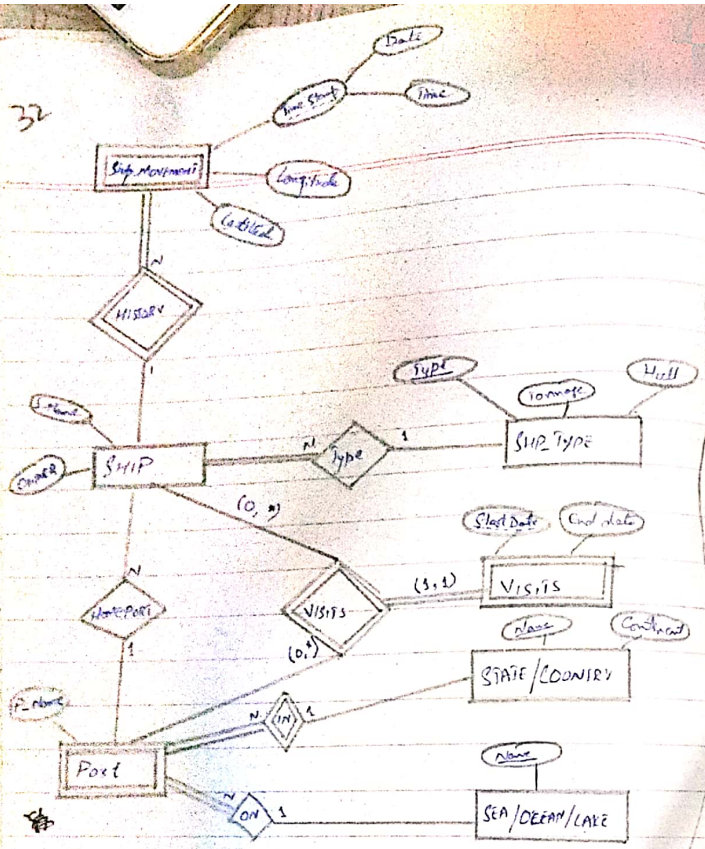
EMP works on PROJECT

EMP assigned to DEPT

EMP has DEPENDENT

EMP supervises (supervises)





VISITS

SName Start_Date End_Date

HOME_PORT

SName Name PName

SHIP

SName OWNER Type

SHIP_Type

Type Tonnage Hull

STATE/COUNTRY

Name Continent

SEA/OCEAN/LAKE

SOL_Name

SHIP_MOVEMENT

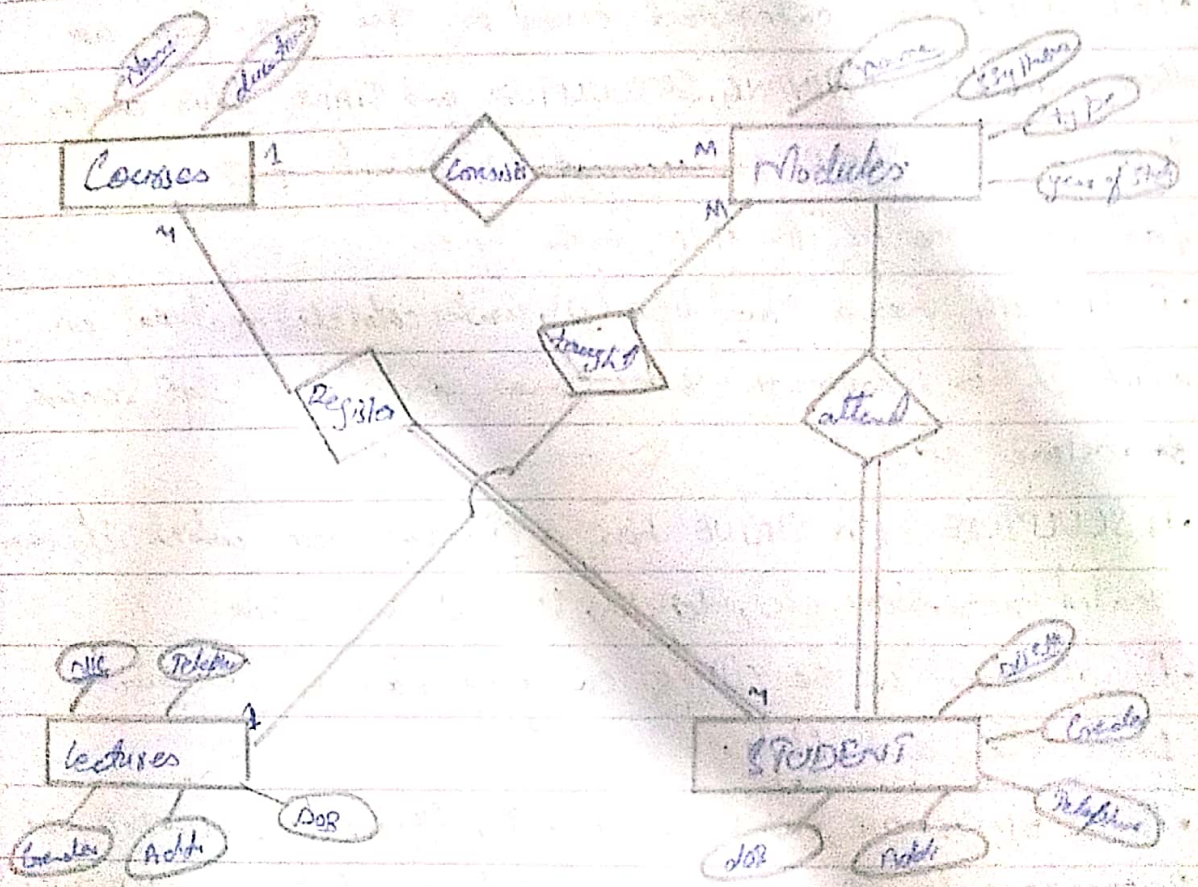
Date time Longitude Latitude SName

PORT

Name Pname SOL_Name

Q) The Department of Computing and Mathematical Science of Karachi University of has a set of courses provided for the undergraduate level. For each course a well-defined set of admission criteria is specified, for instance, for the "Computer Science". course, 3-A-levels are required minimum B for each, as A level in Mathematics being compulsory, Each course has a name and durati certain courses can allow industrial placements. Each course consists of a set of modules, some compulsory and some optional. A certain modules is described by its name, a short syllabus, type (lectured, seminar-like, project) and year of study for which it was intended Each students enrolled with the departments is registere for one particular course. The students must attend the compulsory modules. For every year of study, from the set of optional modules, the student must attend a certain number which is specified in the course's descriptio Each module is taught by a single lecturer. However, there might be modules, from the optional ones, that are not taught at all (for instance, because no student have actually registered for them). There are also can exist lecturers who doesn't teach at all, being only involved in research. Contact information (address, telepho

plus personal details (date of birth, male/female, etc) is needed for both lectures and student.



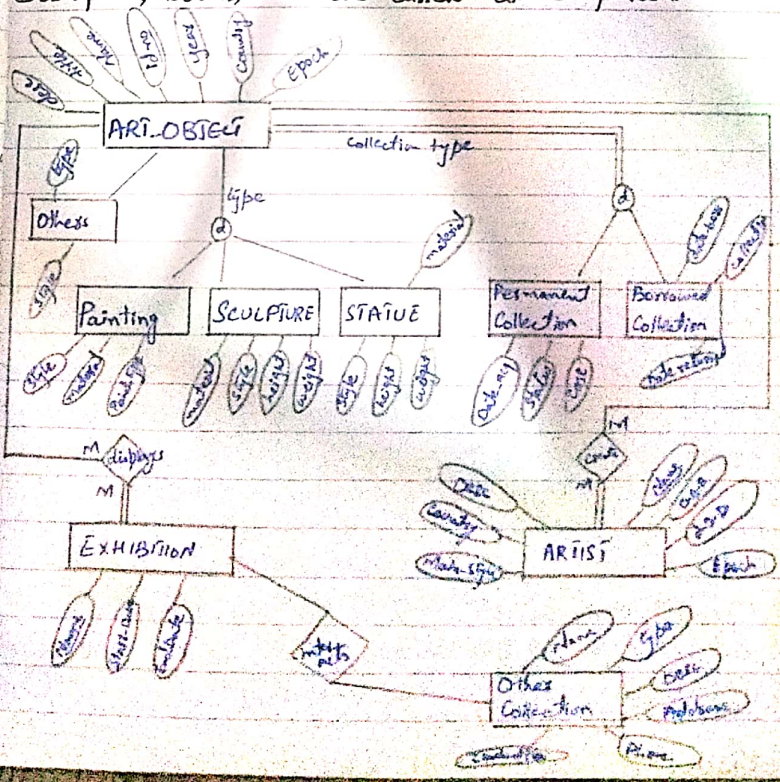
EERD MAPPING DIAGRAM

Design a database to keep track of information for an art museum. Assume that the following requirement were collected:

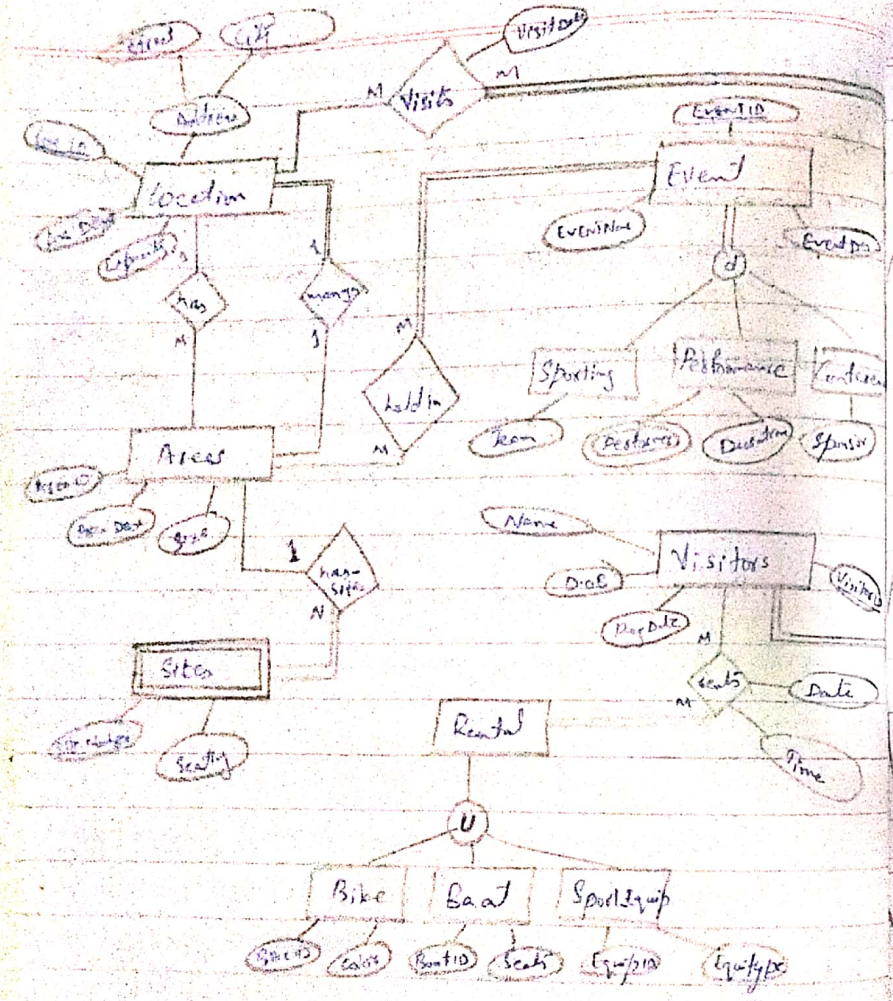
- The museum has a collection of ART_OBJECT. Each ART_OBJECT has a unique Id_no, an Artist (if known), a Year (when it was created, if known), a Title and a Description. The art objects are categorized in several ways, as discussed below.
- ART_OBJECT are categorized based on their type. There are three main types: PAINTING, SCULPTURE and STATUE, plus another type called OTHER to accumodate objects that don't fall into one of the three main types.
- A PAINTING has a Paint_type (oil, water color etc) material on which it is Drawn_on (Paper, Canvas, wood etc) & style (modern & abstract etc).
- A SCULPTURE or a STATUE has a Material from which it was created (wood, stone, etc), Height, Weight and Style.
- An art object in the OTHER category has a Type (paint, photo, etc) and Style.
- ART_OBJECTS are categorized as either PERMANENT_COLLECTION and BORROWED. Information captured about object in the PERMANENT COLLECTION includes Date_acquired, Status (on display, on loan, or stored), and Cost. Information captured about BORROWED objects includes the Collection from which it was borrowed, Date_borrowed, and Date_returned.
- Information describing the country or culture of Origin (Italian, Egyptian & so forth) and Epoch (Renaissance, Modern & so forth) is

captured for each ART_OBJECT.

- The museum keeps track of ARTISTS information, if known: Name, D.O.B (if known), Date_Died (if not living), Country_of_origin, Epoch, Main_Style, & Description. The Name is assumed to be unique.
- Different EXHIBITIONS occurs, each having a Name, Start_date, and End_date. EXHIBITIONS are related to all the art objects that were on display during the exhibition.
- Information is kept on other COLLECTIONS with which the museum interacts, including Name (unique), Type (museum, personal etc), Description, Address, Phone and current Contact_person.



EERD MAPPING



LOCATION

LocID locDesc Capacity Street city AreaID

AREAS

AreaID AreaDesc Size

SITES

SiteNumber Selling AreaID

VISITORS

VisitorsID Name DOB RegDate

VISITS

LocID VisitorsID VisitDate

HELD IN

AreaID EventID

SPORTING

EventID EventName EventDate Team

PERFORMANCE

EventID EventName EventDate Duration

PERFORMER
Event_id Performer

CONFERENCE
Event_ID Event Name Event Desc Sponsor

BIKE
Bike id Color Rental_id

BOAT
Boat id Seat Rental_id

SPORTS EQUIP
Equip id Equip type Rental_id

RENTALS
Rental_id

RENTS
Visitor id Rental_id Date time

POST MID-TERM

Functional Dependencies (FD)

Q Why we use FD?
FD is used to identify following in relation.

- Data Repitition (Data Redundance).
- Insertion Anomalies Problem
- Updation Anomalies
- Deletion Anomalies

Example

M:M, Composite PK.

Student_Course PK

Stu id	Stud name	Stu addr	Course id	CourseName	Instructor
S1015	ABC	XYZ	CE 301	DBMS	CEF

- Updation will be [illegible] done due to these Problem.
- Data repitition in course id, course Name, Instructor
- Insertion Anomalies b/c we cannot insert student data until course data [illegible] course data
- If a course taken by only 1 student and you want to delete only course not student so you are not allowed to do. but if one course is allowed by 2 student so one student record will be delete, we [illegible]

TEACHER

Course id	Teacherid	TName	TOfficeAddr
CE 301	1015	ABC	XYZ
CE 401	1015	ABC	XYZ
CE 402	1025	DEF	

Data is repeating.

- Insertion not possibility jb tk wo koi course na pdhae
- Delete. agr 1 teacher 1 hi course pdha raha hy or hm course drop krna chah rahy hain to teacher ka data waste hojayga

DEFINITION:

Example 1

A	B
A_1	B_1
A_2	B_2
A_3	B_3

Example 2

A	B
A_1	B_1
A_1	B_1
A_2	B_2
A_3	B_1

Example 3

A	B
A_1	B_1
A_1	B_2
A_2	B_3
A_3	B_4

Agr 1 column unique hy to 2nd column determine krrha hy
$A \to B \to$ (A determine to B)
Determinant of column

Agr 1st column repeat ho raha hy or B bhi same repeat hy to
$(A \to B)$

Agr 1st column repeat ho raha hy or 2nd repeat nhi horaha
$(A \not\to B)$

Stuid Studname Courseid (M:M)

Stuid → Studname (Example 2)
Studname ↛ Courseid (Example 3)
Courseid ↛ Stuid. (Example 3)

Stuid Studname Course_id.

Stuid, Course_id → Studname.
Composed
Compound

Coursetitle textbook_name Course_Instruction

NK

Course_title → text_name
Course_title → Course_Instruction
textbook_name → Course_title
Course_Instructor ↛ Course title — If instructor teaches many course

ARMSTRONG AXIOMS RULES OR INFERENCE RULES

1. **REFLEXIVITY.** (Image)

Every determinant determinates itself or its subset

ex

A → A

AB → A

AB → B

AB → AB

2. **AUGMENTATION OR ADDITION RULE.** (Addition)

If A → B

then AC → BC.

3. **Transitivity Rule**

If A → B & B → C

then A → C.

Roll → Courseid

Courseid → Cname

DERIVED RULE

1. Decomposition Rule

If $A \rightarrow BC$

$A \rightarrow B$

$A \rightarrow C$

2. UNION Rule

If $A \rightarrow B$ & $A \rightarrow C$

then $A \rightarrow BC$.

3. Pseudo Transitive Rule

If $A \rightarrow B$ and $BC \rightarrow D$

then $AC \rightarrow D$.

A B C D

Stuid Studname Coursetitle Course name

Studid → Studname

Coursetitle, Studname → Coursename

Stud, Course id → course_name.

(6)

EXAMPLE:

$F = \{A \to B$

$B \to F$

$F \to C$

$FE \to G$

$CD \to E\}$

Derived additional dependences using Armstrong Rules

Using transitive Rule

Since $A \to B$ & $B \to F$

Hence $A \to F$

$A \to B$ and $B \to F$ and $F \to C$

Therefore $A \to C$

Hence $B \to C$.

Using Pseudo transitive Rule

$A \to B$ & $B \to F$ $A \to F$ $FC \to G$

Since $B \to F$ & $FE \to G$ $AE \to G$

Hence $BE \to G$

Since $F \to C$ & $CD \to E$

Therefore $FD \to E$

Compound Determinant

Courseid Studid Studname

Courseid, Studid → Studname

↙

Compound Determine

Full Functional Dependency.

If AB → C

then if A → C

& B → C

Batch, Roll, Dept → Name.

So no subset.

Partial Functional Dependency

If AB → C

then if A → C

or B → C

Courseid, Studid → Studname

So Studid → Studname

(8)

Using Algorithm to find the closure of F (F^+)

$F = \{A \to B, \quad A \to C$
$B \to C\}$

Closure of F

$F^+ = \{A \to B,$
$B \to C,$
$A \to C\}$

$A^+ \rightleftharpoons ABFCD$

$A \to A$
$A \to B$
$A \to F$
$A \to C$
$A \to D$

$F = \{A \to B;$
$B \to F;$
$H \to I;$
$C \to D;$
$E \to H;$
$F \to C;\}$

Find closes of A

$A^+ = ABFCD$

$A \to A$
$A \to B$
$A \to F$
$A \to C$
$A \to D$

Find whether $A \to D$ emit or not

$F \to C, \; C \to D$

So $F \to D$, $B \to F$ & $F \to D$

So $B \to D$, $A \to B$, $B \to D$

So $A \to D$

Uses Of Algorithm.

1. To find whether any dependency exit or not

Example $F = \{A \rightarrow B;$
$BC \rightarrow D;$
$D \rightarrow C\}$

Find wether $A \rightarrow D$ exit or not

Ans) $A^+ = AB.$

$A \rightarrow A$

$A \rightarrow B.$

$A \rightarrow D$ doesnot exit.

2. ~~CAN~~ SUPER KEY.

~~Super key~~ is unique.

Any of its ~~type~~ subset is also unique.

Student

Superkey

Roll No	Batch	Dept	NIC#	Name ...
P.K 87	2018	CE		

10

CANDIDATE KEY

- Unique
- None of its subset can be unique

Student

Roll #	Batch	Dept	NID	Name	...
P.K			A.K		

Candidate key.

Periodic Table.

Atomic #	Name	Symbol.
P.K	A.K	A.K

Candidate key

2. To Find Candidate key or Superkey

Example: $F = \{ A \to B;$
$B \to C;$
$EC \to D;$
$C \to E;$
$D \to I \}$

Single Set Left

$A^+ = ABCEDI$

1 column se sary attributes aajain gy to wo Candidate key hojay gy

Superkey:

Now $(AB)^+ = ABCDEI$

Super — $A^+ = ABCDEI$

Ca — $B^+ = BCEDI$

AB me koman, koi b subset me sary aajain gy to wo superkey hi hoy gy

∞ AB me komen aagy or koi b subset me sary na aain to wo Candidate key hogi

$F = \{ A \to B;$ $CD \to I;$
$B \to F;$ $F \to C;$
$ACD \to E,$ $C \to D; \}$

$A^+ = ABFCDEI$

$(ACD)^+ = ABCDEFI$

12

A^+ =

B^+ = ~~BFEDF~~ D

C^+ = CDI

Ac^+ =

AD^+ =

CD^+ =

3. To find closer of any attribute

IRREDUCIBLE SET OR MINIMAL SET OR CANONICAL COVER

$F = \{ A \rightarrow BC;$
$B \rightarrow C;$
$A \rightarrow B;$
$AB \rightarrow C;$
$AC \rightarrow D \}$

STEP #1

Convert set into singleton set.

$F = \{ A \rightarrow B;$
$A \rightarrow C;$
$B \rightarrow C;$
$A \rightarrow B;$
$AB \rightarrow C;$
$AC \rightarrow D \}$

Now

$F = \{ A \rightarrow C;$
$B \rightarrow C;$
$A \rightarrow B;$
$AB \rightarrow C;$
$AC \rightarrow D \}$

Step#2. Remove extra attribute from L.H.S (if any)

F = { A → C;
B → C;
A → B;
A → C;
AC → D }

Here AB → C
agr A → C aaycha hoga to B ko remove krdain gy or B → C aaycha hoga to A ko remove agr 2ono gy th aajyge to remove nh krain gy or 2ono nh aayge to here ko b remove krdain gy

Now

F = { A → C;
B → C;
A → B;
AC → D; }

$A^+ = ACBD$

So

F = { A → C;
B → C;
A → B;
A → D }

$A^+ = ACDB$

Step #3. Remove those dependencies that can be derived from other dependencies.

$F = \{ B \to C;$ $A \to B$ and $B \to C$ So $A \to C$
$A \to B;$ So we remove $A \to C$
$A \to D \}$

Minimal set.

Cover & Equivalence.

Example 1

$F1 = \{A \rightarrow B,\ A \rightarrow C\}$

$F_2 = \{A \rightarrow B,\ B \rightarrow C\}$ $A \rightarrow C$

F_1 is covered by F_2.. F_2 ki dependencies sary F_2 me mojud hain.

F_2 is not covered by F_1 F_2 ki dependencies sary F_1 me mojud nhi hain

Example 2.

$F_3 = \{A \rightarrow C;\ C \rightarrow B;\ B \rightarrow C\}$ $A \rightarrow B$

$F_4 = \{A \rightarrow B;\ B \rightarrow C;\ C \rightarrow B\}$ $A \rightarrow C$

F_3 is covered by F_4

F_4 is also covered by F_3

therefore F_3 and F_4 are both equivalent

opposite of Analysis

SYNTHESIS OF RELATIONS:

If A → B and B → A then A and B have one to one attribute relationship

(A) Sid	NIC# (B)
A → B , B → A	
one to one	

If A → B and B ↛ A then A and B have a many to one attribute relationship

(A) Sid	Stude (B)
A → B.	
B ↛ A	one to many

If A ↛ B and B ↛ A then A and B have a many to many attribute relationships.

(A) Sid	comic id (B)
	A ↛ B
	B ↛ A
	many to many

TRIVIAL DEPENDENCIES

- A functional dependency is trivial if and only if R.H.S is a subset of L.H.S.

In general $X \rightarrow Y$ if $Y \subseteq X$ (Y subset of X)

$AB \rightarrow A$

$AB \rightarrow AB$

$AB \rightarrow B$

Trivial: B's subset of A's

$AB \rightarrow A$

Non trivial: At least one of B's not part of A's

$AB \rightarrow BC$

Completely non trivial: Non of the B's part of A's

$AB \rightarrow CD$

$A \to B$
$B \to C$
$\Downarrow$
$\times\ A \to C$

$AL \to B$
$BC \to D$
$\Downarrow$
$AC \to D$ ✗

(14)

Minimal set

Example 2.

$F = \{ AB \to C$
$ACD \to B$
$CG \to BD$
$C \to A$
$D \to EG$
$CE \to AG$
$BC \to D$
$BE \to C \}$

Step 1: convert set into singleton

$\{AB \to C$
$ACD \to B$
$CG \to B$
$CG \to D$
$C \to A$
$D \to EG$
$CE \to A$
$CE \to G$
$BC \to D$
$BE \to C \}$

Step 2.

$(CD^+) = CDAB$ ∴ remove A (ACD)

$CE \to A : C \to A$ ∴ remove E

$\{AB \to C$
$CD \to B$
$CG \to B$
$CE \to D$
$C \to A$
$D \to E$
$C \to A$
$CE \to G$
$D \to G$
$BC \to D$
$BE \to C : \}$

$AB \to C$

20

$CG \rightarrow B$

$BC \rightarrow D$

$\Downarrow$

$CG \rightarrow D$ ✗

$\{ AB \rightarrow C,$

$CD \rightarrow B;$

$CG \rightarrow B;$

$C \rightarrow A;$

$D \rightarrow E;$

$D \rightarrow G;$

$CE \rightarrow G$

$BC \rightarrow D$

$BE \rightarrow C; \}$

Ex:1) $F = \{ AB \to C;$
$A \to D;$
$BD \to C;$
$D \to BG;$
$AE \to F \}$

Step 1.

$F = \{ AB \to C$
$A \to D$
$BD \to C$
$D \to B$
$D \to G$
$AE \to F \}$

Step 2

$A^+ : ADBC$ So $AB \to C$ (remove)
$D^+ : DBGC$ So $BD \to C$

$F = \{ A \to C$
$A \to D$
$D \to C$
$D \to B$
$D \to G$
$AE \to F \}$

Step 3:

$A \to D, D \to C$
$\Downarrow$
$A \to C$ X

$F = \{ A \to D;$
$D \to C$
$D \to B$
$D \to G$
$AE \to F \}$

NORMALIZATION.

Normalization is used to resolve following four problems

- Data Repitition
- Deletion Anomalies
- Updation Anomalies
- Insertion Anomalies

Example

PROJECT_EMPLOYEE

atomic | non-atomic also repeat

Projid	Proj_name	Projmgrid	Empid	Empname	Empdept	deptheaddate
234	XYZ	45	1234, 1255, 1256	A, B, C	MIS, MIS, Fin	4-4-50

1NF Data must be atomic or
there shall be no repeating groups.

Sol 1 create 3. colum for each column
Avoid due to null value.

Sol 2 Enter data in single raw for each employee
Data Repitition increased.

Sol 3, Break the table

empid Proj_id Proj_name Proj_mgr_id total_hours

FK

Empid Emp_name Empdate dept_hourly_rate

if AB→C then A→C or B→C

2NF: There should be no partial dependencies

EMPid Projid Projname ~~Proj_name~~ Proj_mgr_id total_hrs

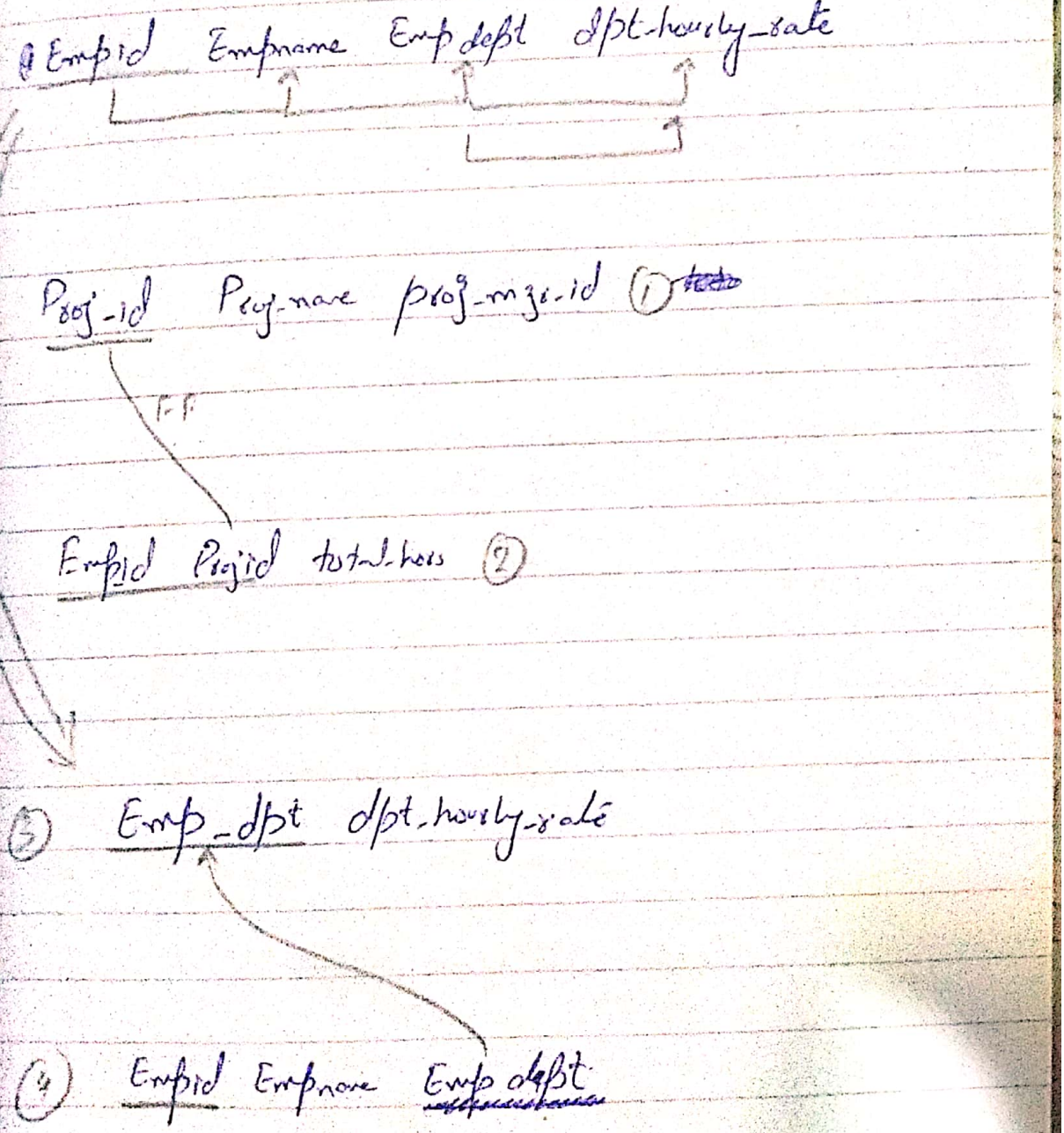
(1) Empid Empname Empdpt dept_hourly_rate

(2) Projid Proj_name Proj_mgr_id.

F.K

(3) Empid Projid total_hrs

3NF: There should be not transitivity.



7b

Q Question

Sales Order

Fictom Company

Main, Karachi

Customer Number : 1001 Sales Order #: 405

Customer Name : ABC Company Sales Order Date : 2/1/2020

Customer Address : Gulshan, Karachi Clerk # : 210

Clerk Name : John

Item #	Description	Quantity	Unit_price	total
800	Sharpner	10	10	100
801	Pencil	20	10	200
805	Erasers	30	10	300
		Order total		600

Solution

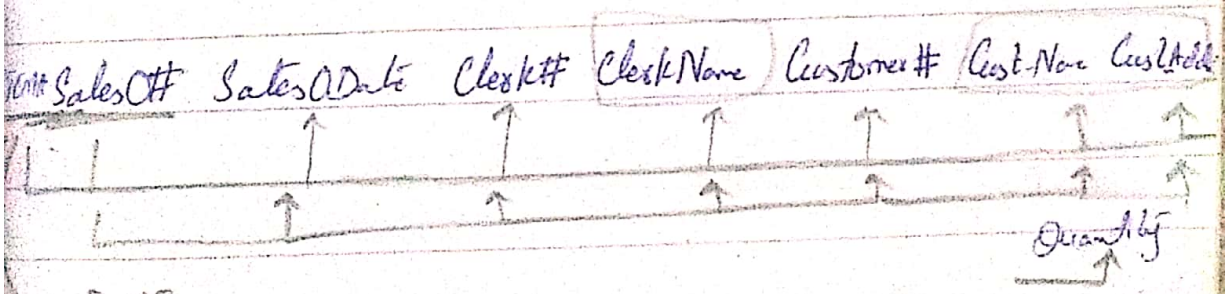
1NF

SalesO# SalesODate Clerk# ClerkName Customer# Cust_Name —

— Cust_addr Item # Desc Quantity Unit Price

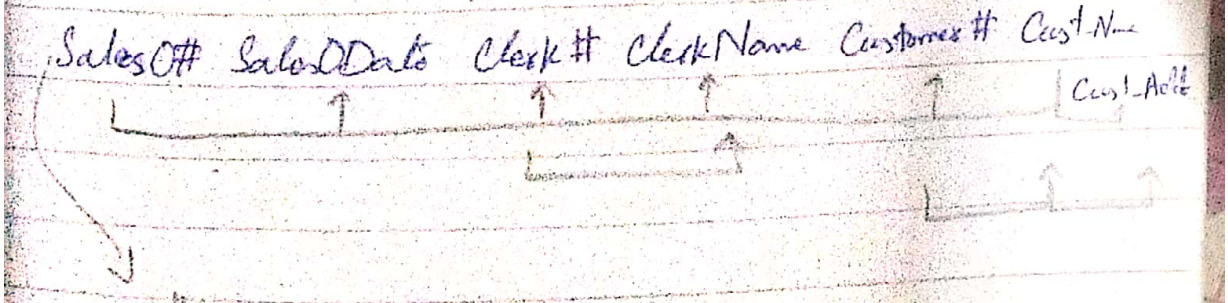
⇓

Item# Desc ~~Quantity~~ UnitPrice (1)



2NF

Item # SalesO# Quantity (2)



2D

Clerk # Clerk Name (3)

Cust # Cust name Cust address (4)

Sales Order # Sales order Date Cust # Clerk # (5)

Example

#

Dept# Deptname Loc MgrName MgrIDNo Telephone Ext Customer# CustomerName

Date of Complaint Nature of Complaint