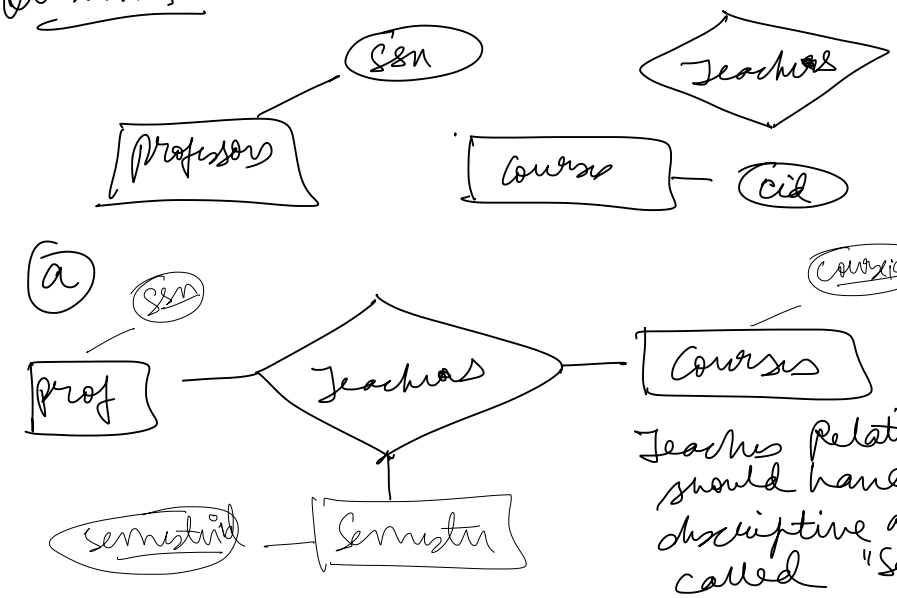
Main exercises

Wednesday, January 31, 2024 5:33 AM

Question 2



(a)

Teaches relationship should have a descriptive attribute called "semesterid" of another entity set called "Semester"

Semester neither describes prof or courses entity set but identifies the Teaches relationship → one has many semesters

So we have another entity set with its own descriptive attributes

2 or more prof can be associated with one course
OR
2 or more courses can be associated with one professor

provided semester number is different

We can't add Semester as an attribute of a course in Courses entity set as a course can be offered in multiple semesters and we keep track of all semesters

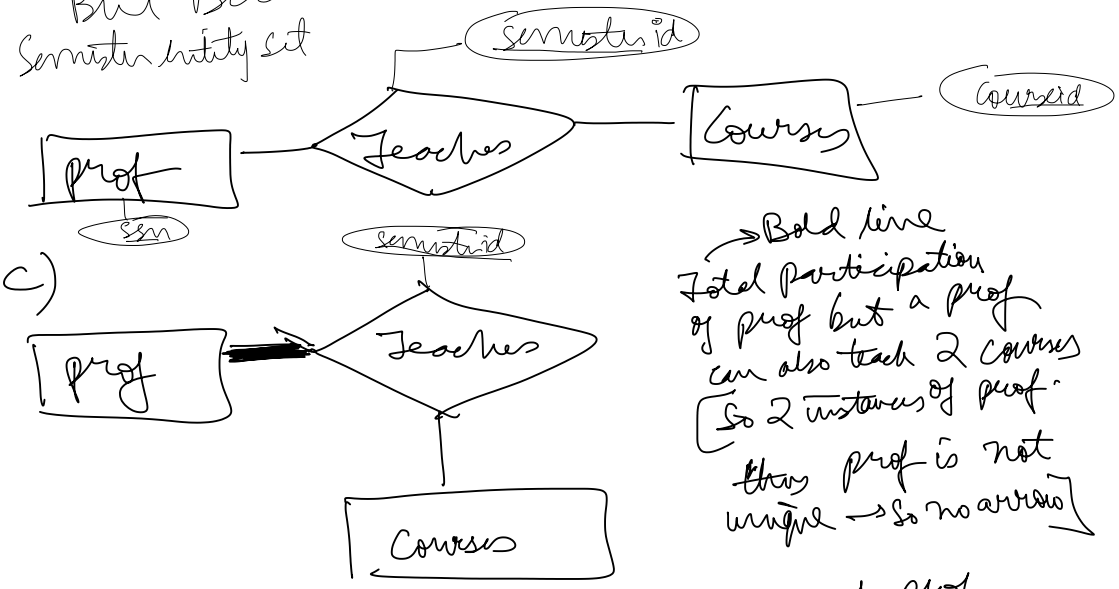
(b) We used to keep track of Semester attribute as only the current / most recent offering is being recorded

now Semester can be an attribute of a course as one course can have only one value

↳ or be an

of Semester (entities) But → all courses will have same value of this attribute, so it is better not to add Semester as an attribute of a course (attribute of the relationship)

But Because we have 1 semester, we can now remove the Semester entity set

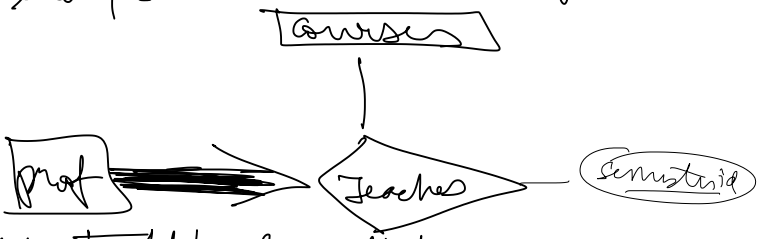


c)

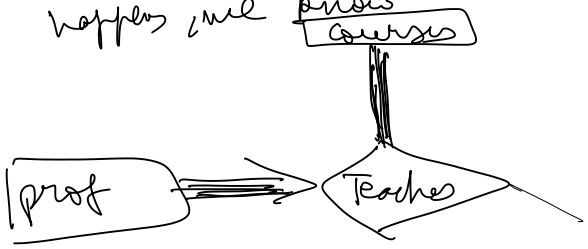
→ Bold line
Total participation of prof but a prof can also teach 2 courses
So 2 instances of prof.
Thus prof is not unique → So no arrow

d) Total participation + ~~all~~ unique instances of prof
But courses is not in Total participation or unique as # courses can be greater than # prof
So some courses might not be taught

If # courses = # prof
then both entity sets have TP + unique



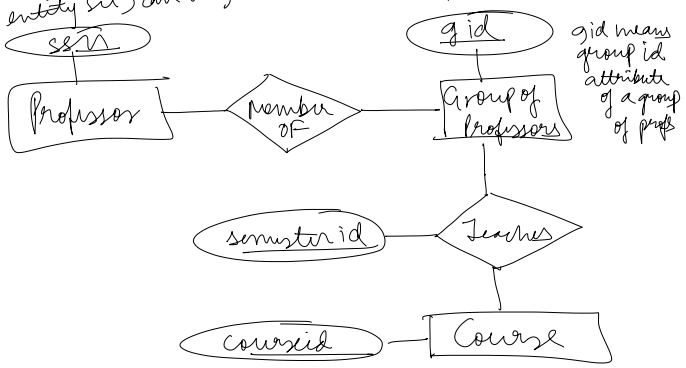
e) every course must be ~~taught~~ by some prof
So Total participation of courses but not unique as 2 prof can teach same course → when this happens (# courses < # prof)



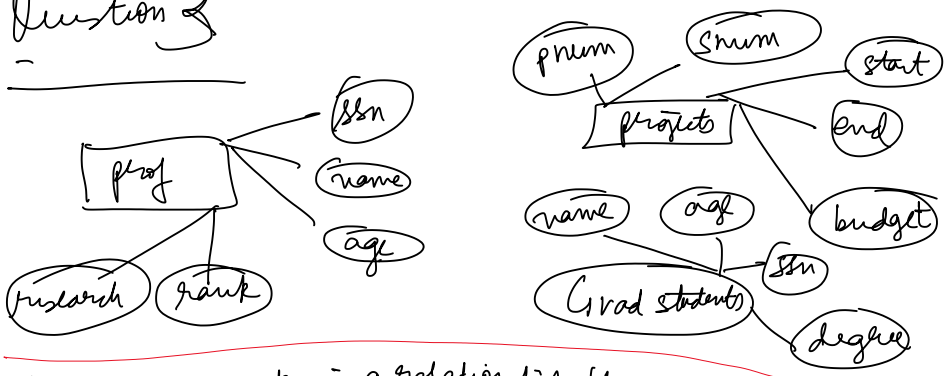
So 2 instances of 1 course for both prof

a) ~~an~~ one to one a course and one professor can be a part of that

f) There is a Teaches relationship between that group and a course (a group now we include a new entity set called Group and a professor now (in its entity set) can only have a relationship with a group and not directly with a course



Question 3



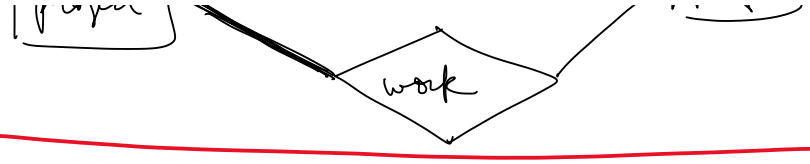
d) project is a key in a relationship b/w prof and project

Total participation of projects because of "each"
no constraint on prof.

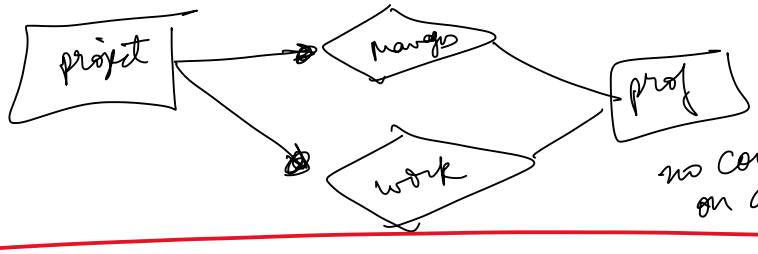
e) Make new works relationship
one or more prof: Project is not a key, no constraints on prof
Only Total participation of projects
not a key as one project can have many professors

Wording wise: work (not manages)
So we can even make a new relationship

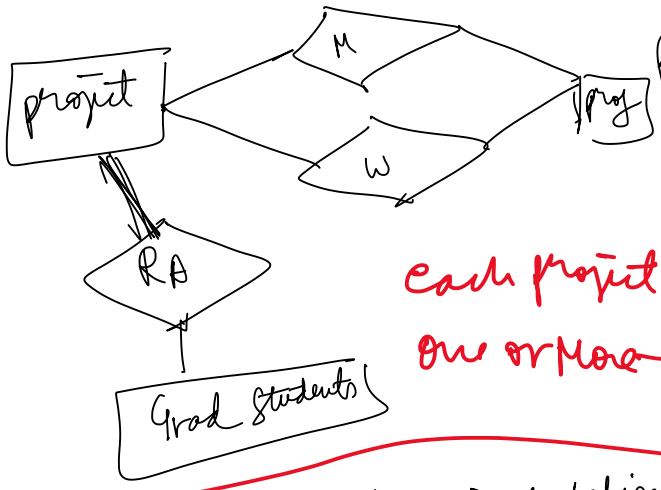
now project is not a key



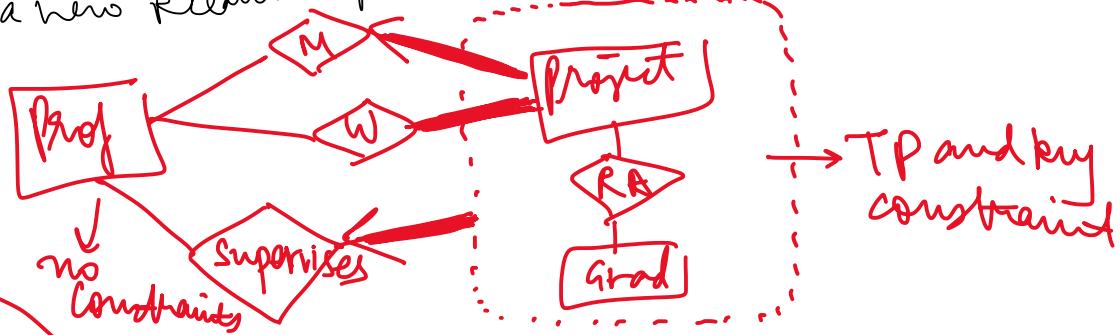
f)



g)



h) whole RA/~~Set~~ work-prof relationship becomes an aggregation ~~to~~ a new relationship called "Supervises"

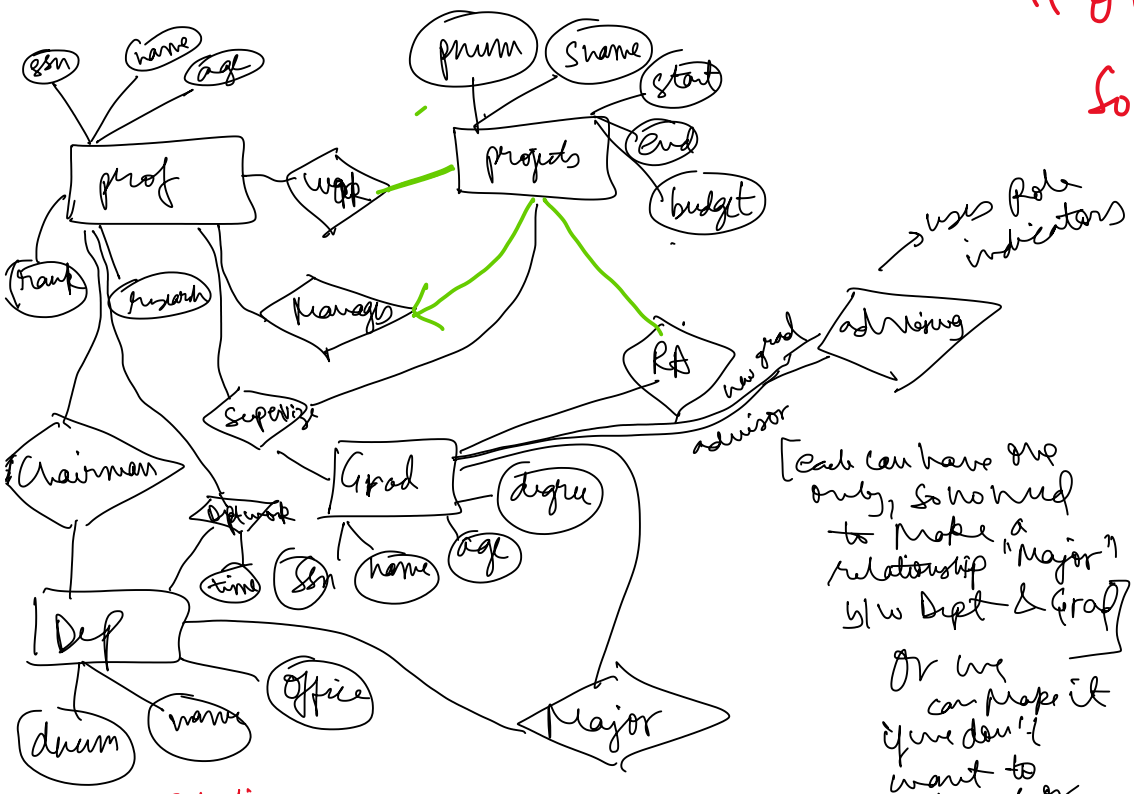


j) and i) and k)

dupe so TP by grad for

"one other grad advisor" means key constraint

prof has TP cou as all of them wi "one or more" ↓ so no key con



Each can have one only, so no need to make a relationship "major" b/w Dept & Grad

Or we can make it if we don't want to change or add attributes to an entity set

LOOK CORR DIF IN OF AN

Correct Solution

prof : primary key is ssn

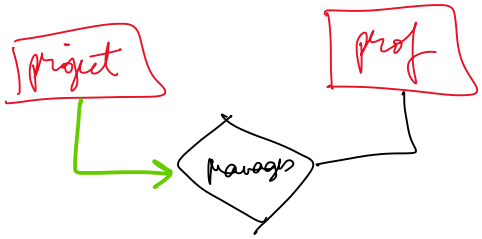
project : " pnum

grad : " ssn

a) each : ~~total participation~~

one project – 1 prof (so project is key constraint)

one prof can manage more than 1 project while some might not manage any project (neither total participation nor key constraints)

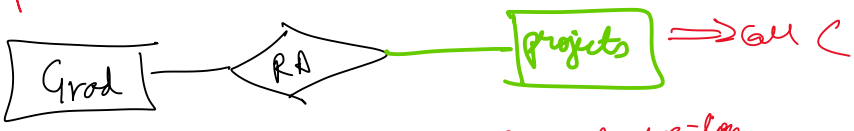


b) each project : TP

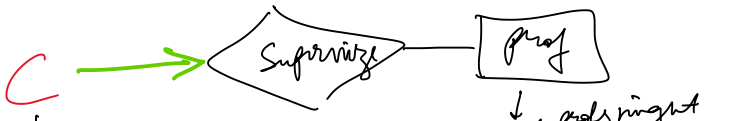
one or more prof → not key anymore

c) no constraint on profs

d) each: TP → not key as more than 1 RA can be there for a project



e) either make a ternary relationship or Aggregation
RA.. why used aggregation



all grad students satisfied
all projects satisfied

some profs might not supervise any project which some might supervise

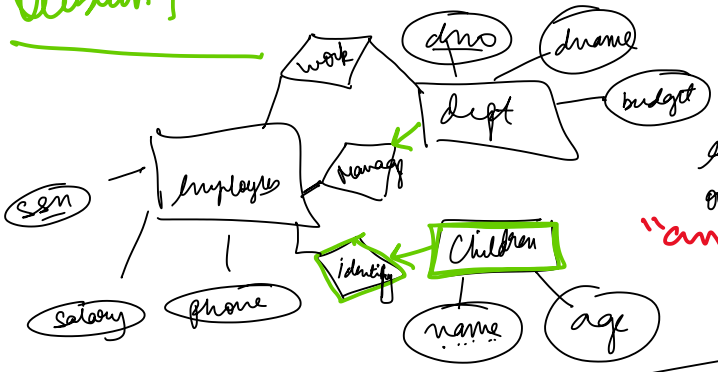
f) g) h)

why TP & Key for dept?

Each department
so TP

why TP b/w prof & work Relationship
b/w (Prof, Dept)

Question 4 — no constraint in ~~Prof~~ work relationship

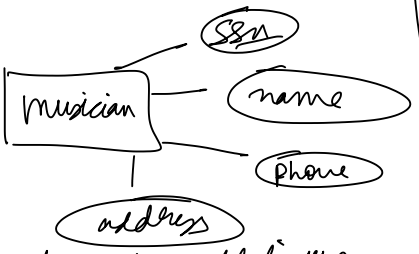


each dept: TP
one dept – one "can employee": dept
is key constraint as it has only 1 manager

Child's name uniquely identifies it when taken together with parent's ssn → child's name is a partial key (Child is a weak entity set)

Child/weak entity set is always made bold with the identifying relationship

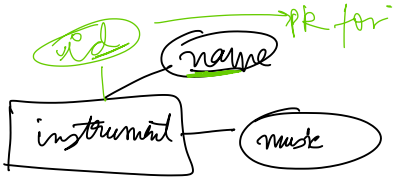
5) a)



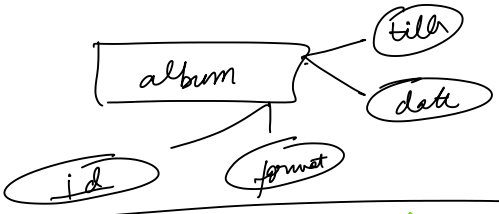
problems: 1) 2 types of address related issues
2) no address with > 1 phone

d)

b)



c)



e) given instrument may be played
no TP constraint for instrument
so no TP for a musician

each: TP (may play → not necessarily, so a music may not play anything)

f) each album has a # of songs → doesn't mean TP for album
↓
it can have 0 songs as well

no song on more than
only TP & PK for song

g)

what if a song is never performed
than it has 0 Musicians for it
↳ so no TP for songs

no constraint
(not key as a musician can do 2 songs)
not TP as a musician may not do any song

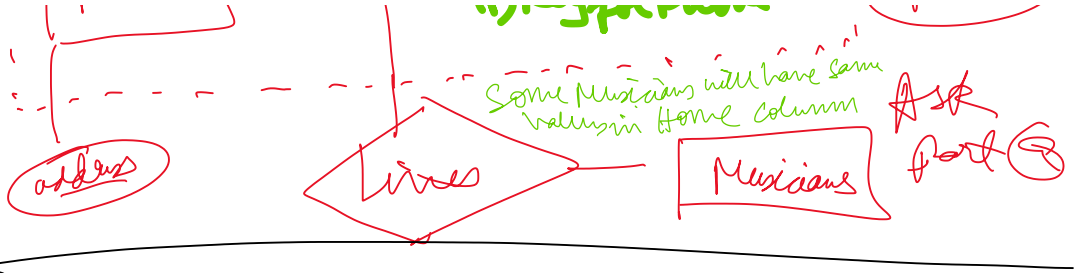
h)

each: TP
one musician: key

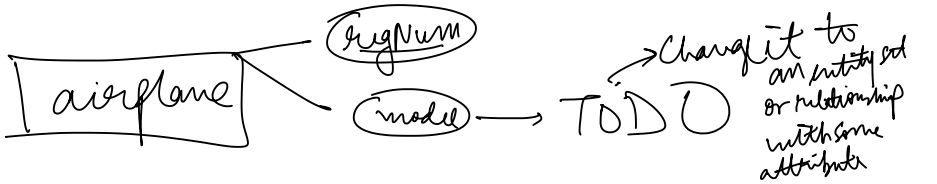
no constraint

part a : each Musician has a phone number and no home has more than 1 phone and each Musician lives in some home
↓
so each home has a phone (only 1 phone)

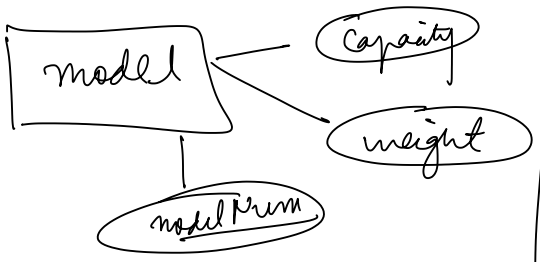
TP: Home & Phone



6 a)

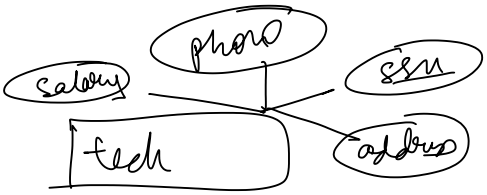


b)



way 1
airplane and model in a relationship

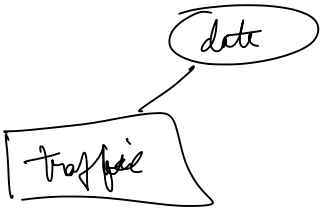
c)



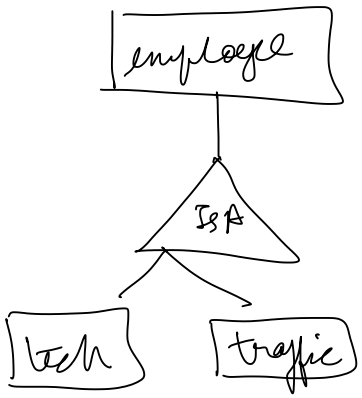
d)

expertise overlap → 2 technicians specializing in same model
So a model has no constraint

e)



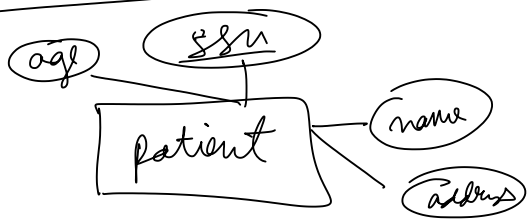
f)



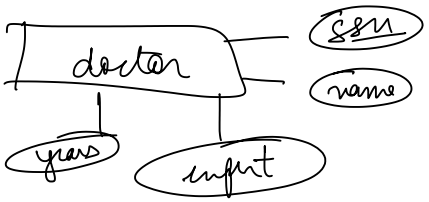
add SSN to traffic-controllers

ER full question

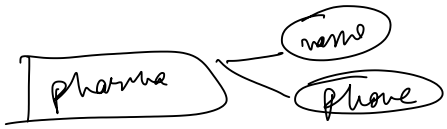
7) a)



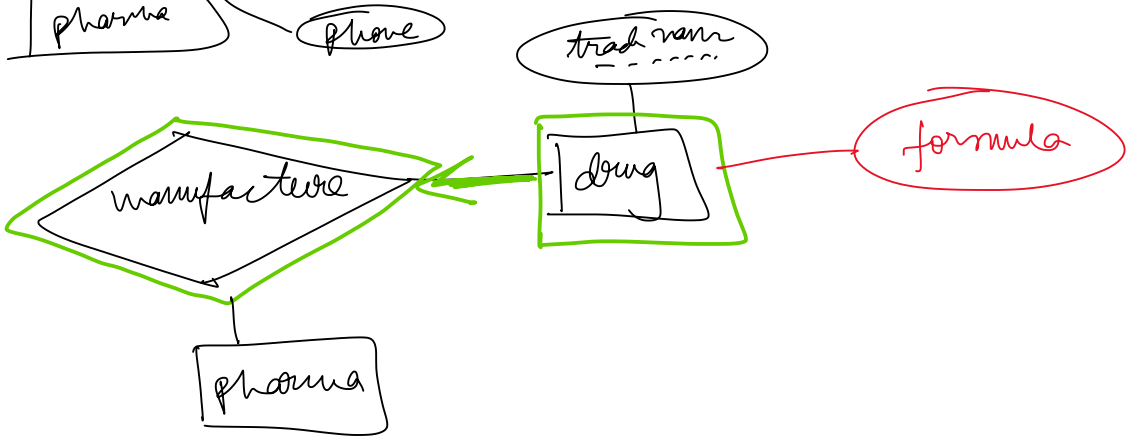
b)



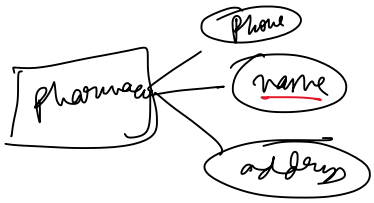
c)



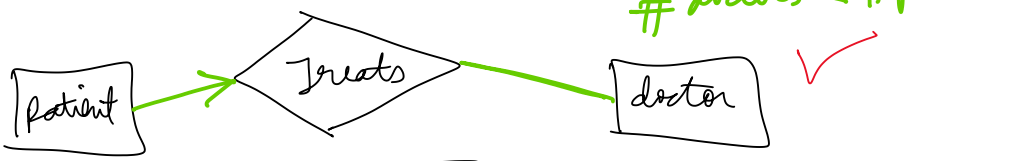
d)



e)

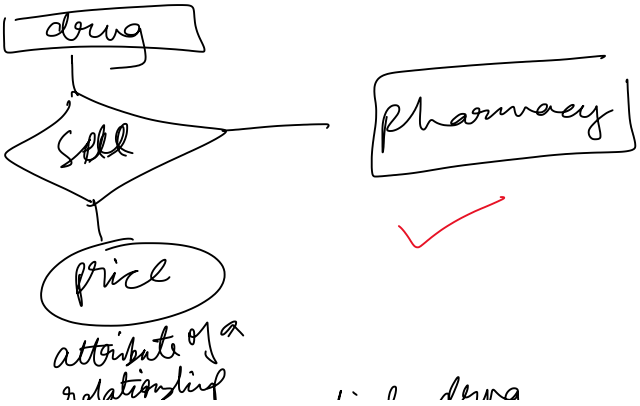


f)



doctors < # patients ✓

g)

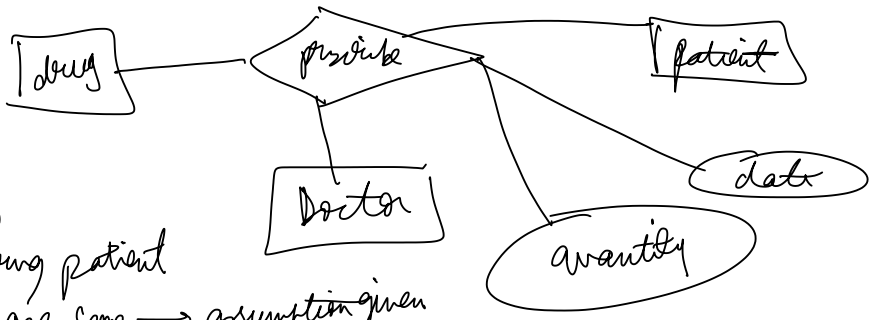


✓

no constraints on any of the entity sets

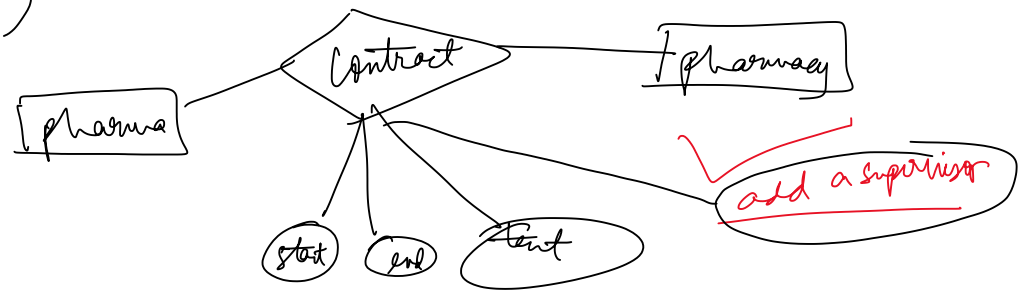
(not specific to a particular or a particular pharmacy)

h)

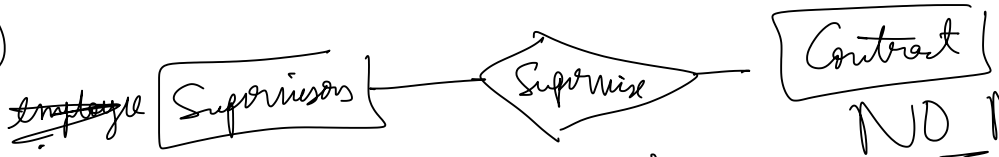


no two
doctor drug patient
records are same → assumption given

i)



(j)



Contract can have 2 supervisors
one supervisor can supervise 2 contracts

NO NEED TO MAKE A SUPERVISE RELATIONSHIP

Part b : Same price for drugs
change can q : now no need of price attribute → still some pharmacies may not sell some drugs which some drugs may be sold by more than 1 pharmacy
so no more changes

Eliminate SELL Relationship → Make PRICE an attribute to the DRUG entity set

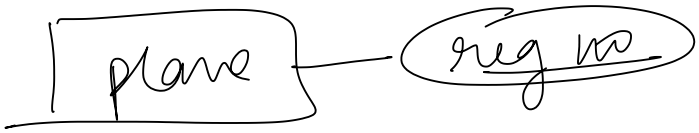
Part c : 2 instances of doctor drug patient
where doctor can change

Now Date attribute can't Remain
So create a new entity set called Date and make
it join the Prescribe Relationship (Now a 4 way relationship)
just add supervisor as an

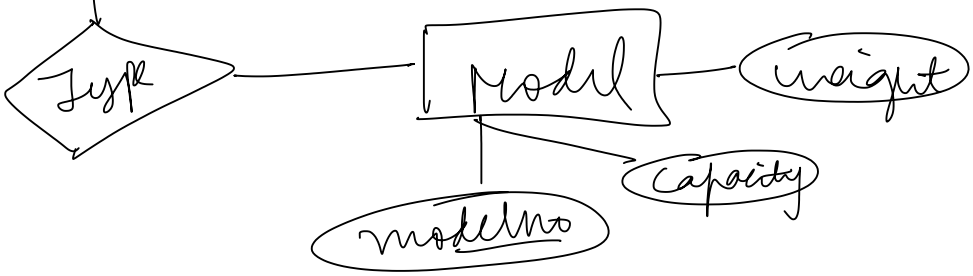
Solution to part j — 1 attribute

Question 6 correct Answer

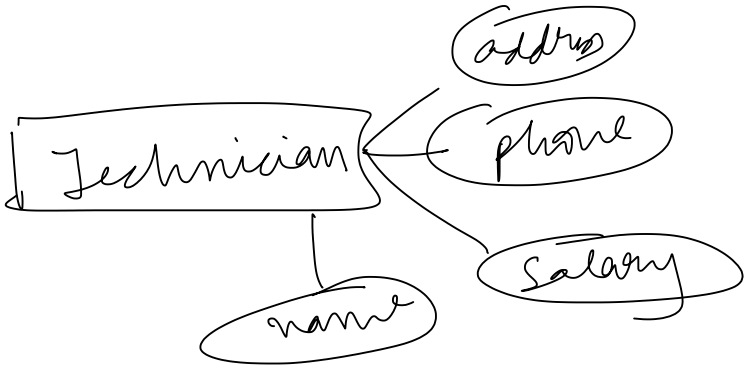
(a)



(b)



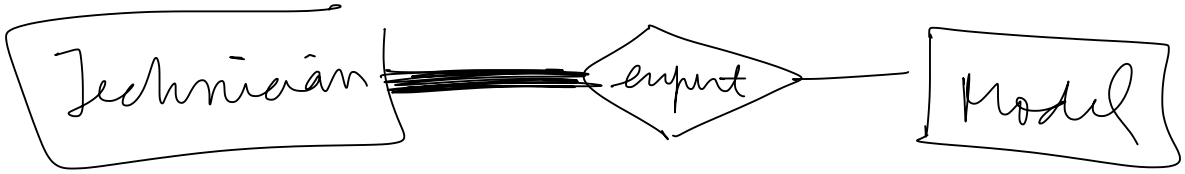
(c)



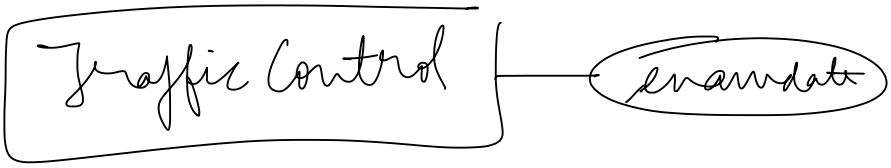
(d) each Technician: TP

overlap → many technicians can be experts of one Model

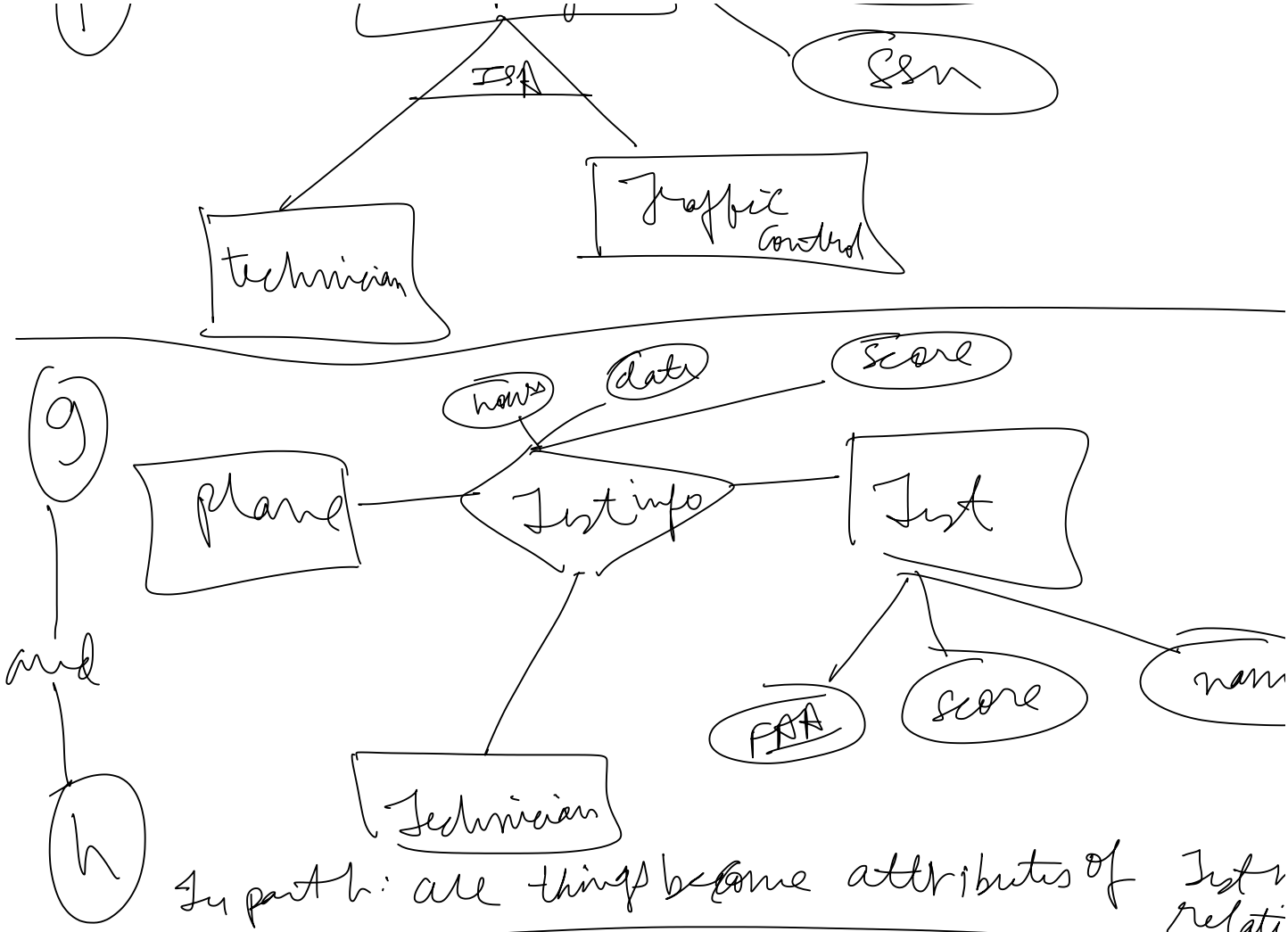
Since co
on a p



(e)



(f)



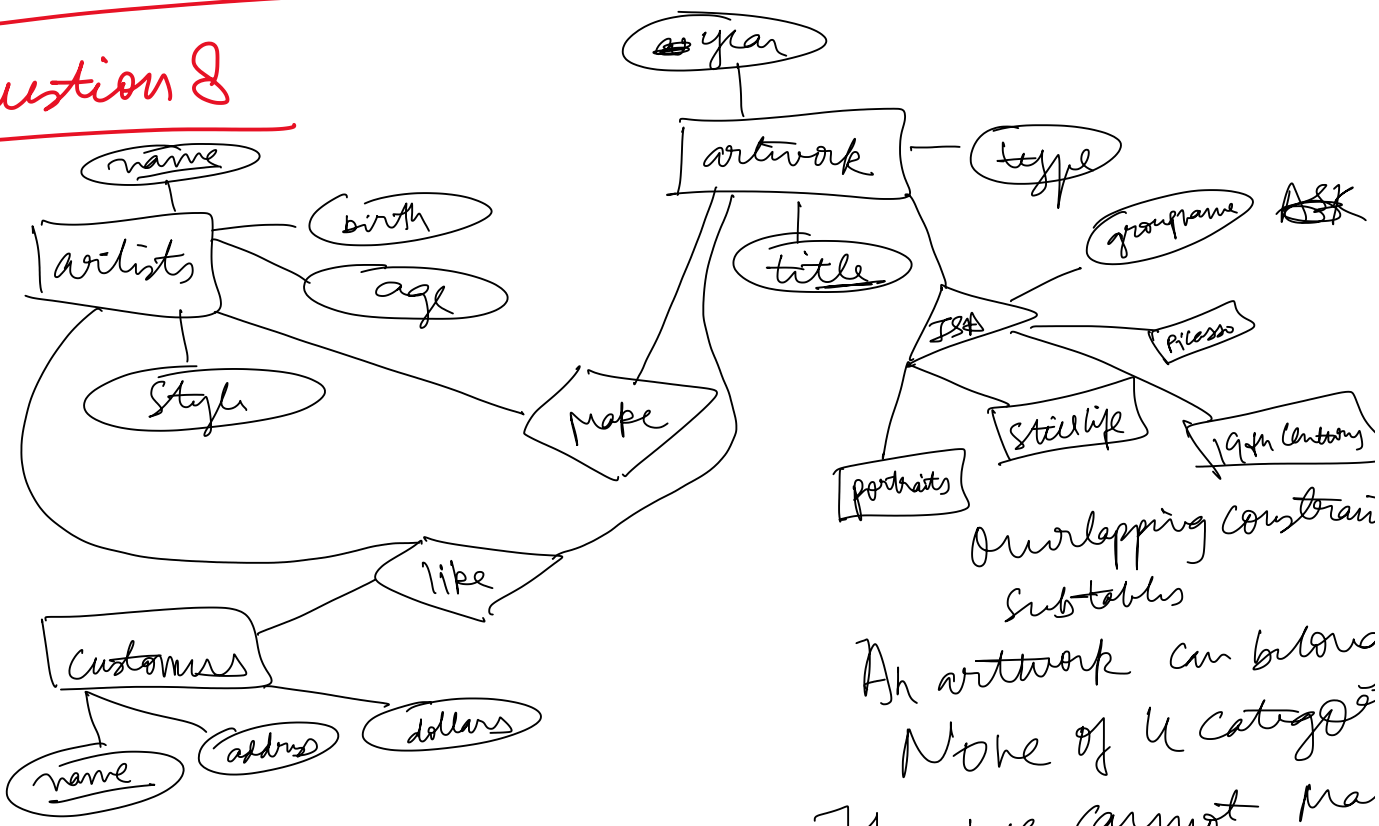
In path: all things become attributes of flight relati

extra condition

we cannot express this constraint as it is
Yes/No check condition → we can only
relationships b/w entities in an ER diagr
while this question first asks us to ch
condition on the relationship b/w a Techn
and his expertise on a model and then
if the plane he wants to operate on
the same model or not → Thus 2 ch
condition C

can't happen as ~~Date~~ ...
in ER diagrams

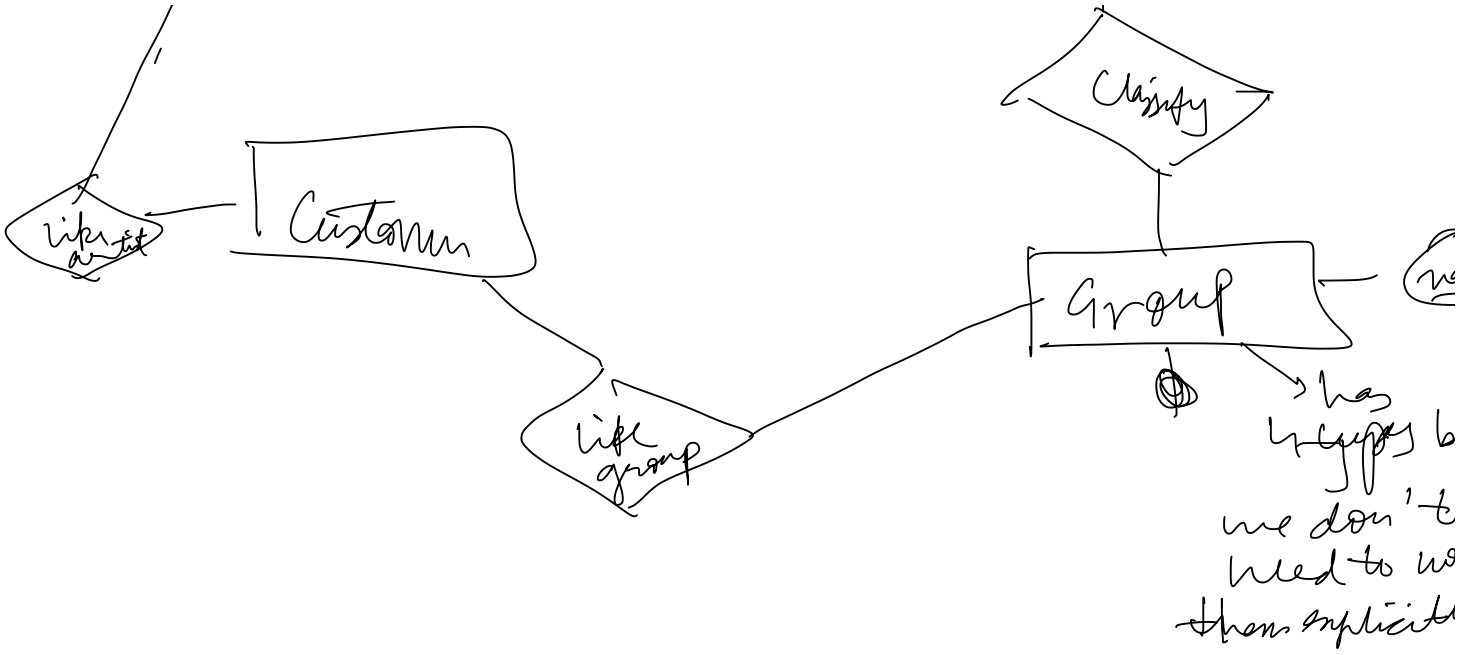
Question 8



Overlapping constrai
subtables
An artwork can belon
None of 4 catego
Thus we cannot ma
groupname an attribu
of artwork entity set

Method 2

each artwork : TP
Only one artist creates ~~one~~ an artwork by Common Sense
So key constraint on Artwork



QUESTION 1 AND 9 ARE FROM BOOK TEXT