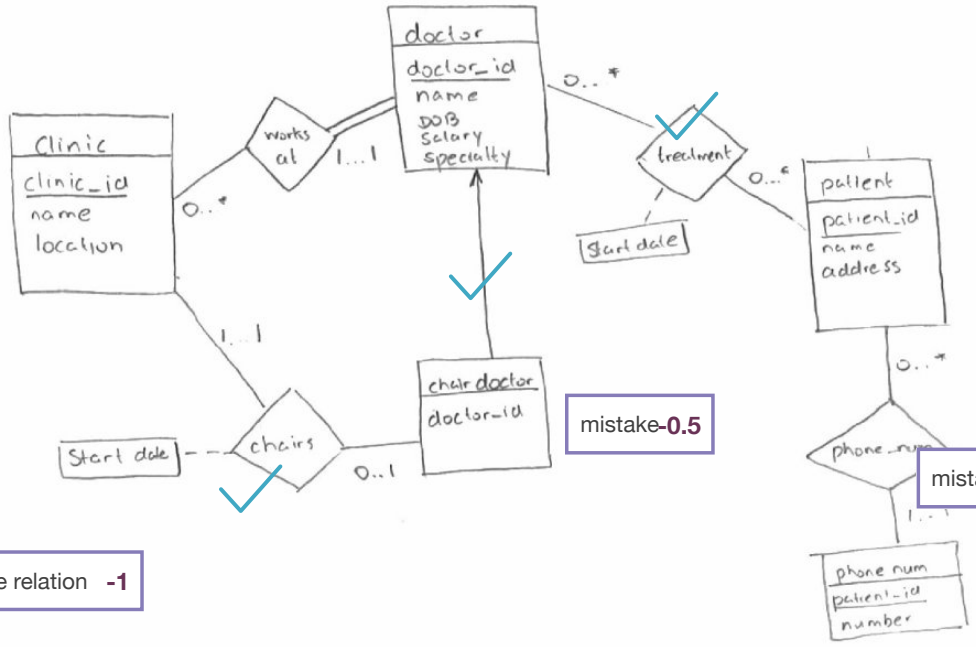
Question 1 [10 points]. Given the following information about a hospital database, draw the corresponding ER diagram. Your diagram should reflect all the requirements listed below as closely as possible.

Q1 8

- The hospital consists of clinics. Each clinic has a clinic_id that uniquely identifies it, a name, and a location.
- Doctors work at the hospital. Each doctor has a doctor_id that uniquely identifies him/her from other doctors, a name, a date_of_birth (DOB), a salary, and a specialty.
- One or more doctors work at each clinic, and each doctor works at exactly one clinic.
- Each clinic is chaired by one and only one doctor called the chair doctor, and a doctor may chair zero or (at most) one clinic. When a doctor starts to chair a clinic, there is a start date for his or her appointment.
- A chair doctor supervises zero or more doctors, and a doctor is supervised by at most one chair doctor.
- Patients receive treatment from doctors. Each patient has a patient_id that uniquely identifies him or her from the other patients, a name, an address, and a phone number.
- A patient may be treated by zero or more doctors, and a doctor may treat zero or more patients. When a doctor starts to treat a patient, a treatment start date must be recorded.



mistake-0.5

mistake-0.5

supervise relation -1

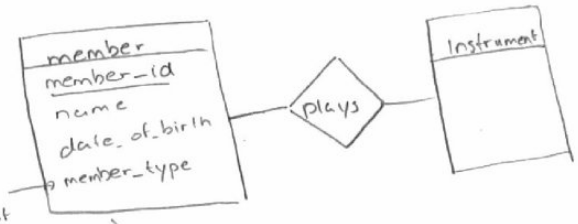
Additional space for your answer to Question 1

Q1page2 0

Q2 4

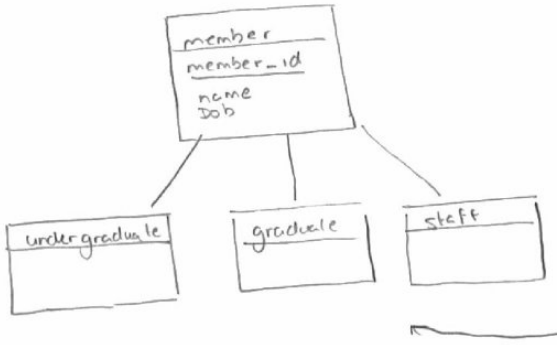
Question 2 [4 points]. The University Jazz Band requires a database to store data about its members, who can be undergraduate students, graduate students, or university staff. For each member, the club wants to record the member_id, which is a primary key, the name, and the date of birth. Also, the entity "member" has a relationship, called "plays", with an "instrument" entity. Assuming that no additional information needs to be captured by the data model, would you consider modelling members as a single entity, or as a specialization, with a "member" parent, and "undergraduate", "graduate" and "staff" children?

Option 1



to hold what type of member they are ie (undergrad, grad etc)

Option 2



Analysis

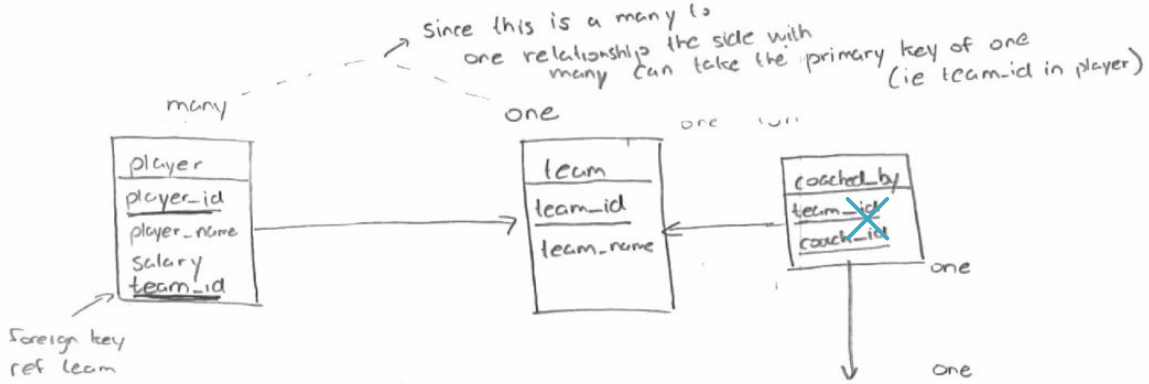
From the two options it is clear that specialization would have a member parent but amongst the children (undergrad, grad, staff) they do not hold other unique attributes.

there is nothing unique attribute to differentiate something like undergrad from grad vs. staff.

∴ it is better to use a single entity with a member type attribute

Q3 4

Question 3 [6 points]. A basketball database contains information about players, teams and coaches. A player is identified by a player_id, and also has a player_name and a salary. A team can be identified by a team_id, and also team_name. A coach is identified by a coach_id, and also has a coach_name. Every player plays for exactly one and every team has exactly one coach. Draw the schema diagram corresponding to this information so as to capture as many of the constraints as possible. If you cannot capture some constraint, explain why.



Constraints not captured

- relation between player and coach unknown
- assuming relationship between player and team has no relationship attributes (ex. starting date) otherwise a relation entity would be required

Q4 7

Question 4 [8 points; 2 for each part]. Let T be a table with n records. Assume T has three integer columns: a, b and c. Assume (a,b) is the primary key of T, and that there are no blanks or NULLs in any column. Without making any assumptions, give the **minimum** and the **maximum** number of records that would be returned by the following relational algebra expressions and SQL queries.

a) SELECT DISTINCT(a) FROM T
b) SELECT * FROM T WHERE c=10
c) SELECT * FROM T WHERE a=1 and b=2
d) T ∪ T

* T table with n records (tuples)
* 3 columns (attributes)

a) Distinct (a) from T

minimum: 1 → only one unique a value

max: n → all a values unique

b) * from T where c=10

min: 0 → no tuples with c=10

max: n → all tuples with c=10

c) * from T where a=1 & b=2

min: 0 → no tuples where a=1 & b=2

max: n → all tuples have a=1 & b=2

d) T ∪ T → union

min: n

max: n

Q5 8

Question 5 [8 points; 4 each]. Consider a supply chain database with three tables: Suppliers, Parts, and Catalog (indicating which suppliers supply which parts, and at what price).

Suppliers(sno, sname, address) Parts(pno, pname, colour) Catalog(sno, pno, price)

Note: Catalog.sno is a foreign key referencing Suppliers.sno, and Catalog.pno is a foreign key referencing Parts.pno.

Write the following queries in SQL

a) Print the sno of each supplier who supplies at least one part. Remove duplicates from the output.

```
Select distinct sno
From Suppliers S, Catalog C
where S.sno = C.sno
```

b) Print the colours of parts supplied by suppliers named Bob. Remove duplicates from the output.

```
Select distinct colour
from Suppliers S, Parts P, Catalog C
where S.sno = C.sno And C.pno = P.pno
```

Question 6 [16 points; 4 each]. Recall the supply chain schema from Question 5, repeated below for convenience.

Q6ab 6

Suppliers(sno, sname, address) Parts(pno, pname, colour) Catalog(sno, pno, price)

Write the following queries in relational algebra

a) Print the names of suppliers who supply at least one green part that costs more than 10 dollars and at least one blue part that costs more than 20 dollars.

Green part costs more than $10 : $T_1 \leftarrow \sigma_{colour = "green"}(Parts) \bowtie \sigma_{price > 10}(catalog)$

blue parts more than $20 : $T_2 \leftarrow \sigma_{colour = "blue"}(Parts) \bowtie \sigma_{price > 20}(catalogue)$

$$\Pi_{Sname}(Suppliers) \bowtie T_1 \bowtie T_2$$

b) Print the sno and sname of each supplier who supplies at least one yellow part but does not supply any red parts.

$T_1 \leftarrow \Pi_{pno}(catalog) \bowtie \Pi_{pno}(catalog)$

$T_2 \leftarrow T_1(\sigma_{= yellow \wedge color \neq red})$

must have atleast one yellow part but no red part logic: yellow-red -2

Question 6, continued: Suppliers(sno, sname, address) Parts(pno, pname, colour) Catalog(sno, pno, price)

Q6cd 1

rint the pno of each part that is supplied by exactly two different suppliers.

Π_{pno} ((parts ⋈ catalog ⋈ ($\sigma_{a.sno \neq b.sno}$ (suppliers)

(ρ_a (supplies)) (ρ_b (suppliers))

Also find out for atleast two suppliers, atleast 3 suppliers c1<- catalog 1 c2<- catalog 2 c3<-catalog 3 final ans atleast 2-atleast 3 **-3**

d) Print the sno and sname of each supplier who supplies every part.

$\Pi_{sno, sname}$ (suppliers ⋈ parts ⋈ catalog)

atleast one part,Missing Part,all possible parts,supply all parts, doesn't supply all parts,supply all parts not mentioned **-4**

Question 7 [8 points; 4 each]. Recall the GET and POST methods discussed in Lab 3.

Q7 7

Explain the purpose of these methods. What functionality do they enable?

- Get and post allow query data to be displayed/used in html ✗
- acts as a web server for accessing data

b) Explain the difference between GET and POST.

Get

- displays trace inside the URL HTTP request
- saved and cached
- does nothing ?

Post

- does not display data within the URL (more secure)
- not saved not cached
- data is resubmitted