ص ۱

سینا غلامی ۹۵۲۱۳۰۷

درس پایگاه داده - دکتر شهرابی - تمرین #۱ تا ۴

Question 1)

a) Mapping cardinality:

Mapping cardinalites, or cardinality ratios, express the number of entities to which another entity can be associated via a relationship set. Mapping cardinalites are most useful in describing binary relationship sets, although they can contribute to the description of relationship sets that involve more than two entity sets. For a binary relationship set R between entity sets A and B, the mapping cardinality must be one of the following:

① One-to-one = An entity in A is associated with at most one entity in B, and an entity in B is associated with at most one entity in A.
(→)(→)

② One-to-many = An entity in A is associated with any number (zero or more) of entities in B. An entity in B, however, can be associated with at most one entity in A.
(→)(—)

③ Many-to-one = An entity in A is associated with at most one entity in B. But an entity in B can be associated with any number (zero or more) of entities in A
(—)(→)

④ Many-to-Many = An entity in A is associated with any number (zero or more) of entities in B. Similarly, an entity in B is associated with any number (zero or more) of entities in A.
(—)(—)

b) Weak entity set:

A weak entity set is one whose existence is dependant on another entity set, called its identifying entity set. Instead of associating a primary key with a weak entity, we use the primary key of the identifying entity, along with extra attributes, called discriminator attributes to uniquely identify a weak entity. The relationship associating the weak & strong entities is called an identifying relationship.

c) Strong entity set:

An entity set that is not a weak entity set is termed a strong entity set. In other words, a strong entity set is an entity that contains sufficient attributes to uniquely identify all its entities. Simply, strong entity is nothing but an entity set having a primary key attribute or a table which consists of a primary key column.

d) Total participation:

The participation of an entity set E in a relationship set R is said to be total if every entity in E must participate in at least one relationship in R. If it's possible that some entities in E do not participate in relationships in R, the participation of entity set E in relationship R is said to be partial.

Question 2)

Each weak entity set must be a a part of one-to-many relationship set. weak entity set is required for the following reasons:

① To avoid the inconsistencies caused by duplicating the key of the strong entity:

Although weak entity set can be converted into strong entity set by simply adding appropriate attributes, this approach results in the redundant storage of the primary key. Also, the primary key of a weak entity set can be inferred from its relationship with the strong entity set. If we add primary key attributes to the weak entity set, they will be present in both the entity set and the relationship set and they have to be the same. Hence, there will be redundancy in the ER diagram and we lose the concept of dependency.

② Weak entities reflect the logical structure of an entity being dependent on another entity:

A member of a strong entity set is called dominant entity & a member of the weak entity set is called as subordinate entity. A weak entity set does not have a primary key but we need a means of distinguishing among all those entries in the entity set that depend on one particular strong entity set. The descriminator of a weak entity set is a set of attributes that allow this discrimination to be made.

(3) Weak entities can be deleted automatically when their strong entity is deleted:

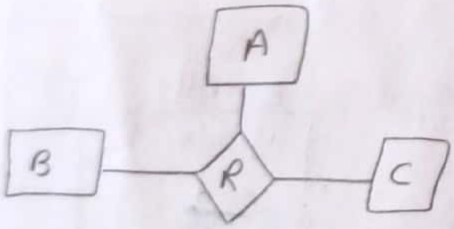
Existence of the weak entity set depends on the existence of the strong entity set. Thus, deletion of the strong entity set results in the deletion of the weak entity set. Therefore, if a database contains an entity set that depends on another entity set that depends on another entity set, it is necessary to mark them as weak entity set. (required!)

(4) weak entities can be stored physically with their strong entities:

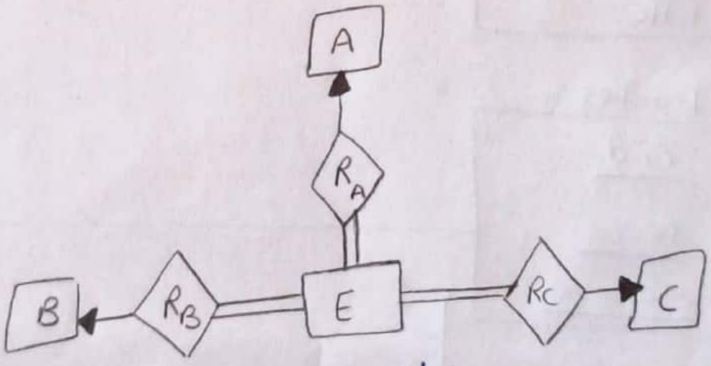
For instance, indexes created automatically with the primary key column will store the values of weak entity columns in the consequtive disk locations. This will lead to faster data retrieval process.

Question 3)

Yes. It is always possible to replace a nonbinary (n-ary, for $n>2$) relationship set by a number of distinct binary relationship sets. In the following ternary relationship set R, (relating entity sets A, B, C) we replace the relationship set R with an entity set E, and we create three relationship sets as in (b):



(a)



(b)

- R_A: A many-to-one relationship set from E to A
- R_B: " " " " E to B
- R_C: " " " " E to C

E is required to have total participation in each of R_A, R_B, and R_C. If the relationship set R had any attributes, these are assigned to entity set E. Furthermore, a special identifying attribute is created for E (since it must be possible to distinguish different entities in an entity set on the basis of their attribute values). For each relationship (a_i, b_i, c_i) in the relationship set R, we create a new entity e_i in the entity set E.

Then, in each of the three new relationp sets, we insert a relationship as follows:

$$\begin{cases} (e_i, a_i) \text{ in } R_A \\ (e_i, b_i) \text{ in } R_B \\ (e_i, c_i) \text{ in } R_C \end{cases}$$

we can generalize this process in a straightforward manner to n-ary relationship sets. Thus, conceptually, we can restrict the ER model to include only binary relationship sets. However, this restriction is not always desirable, such as:

(1) An identifying attribute may have to be created for the entity set created to represent the relationship set. This attribute, along with the extra relationship sets required, increase the complexity of the design and overal storage requirements.

(2) An n-ary relationship set shows more clearly that several entities participate in a single relationship.

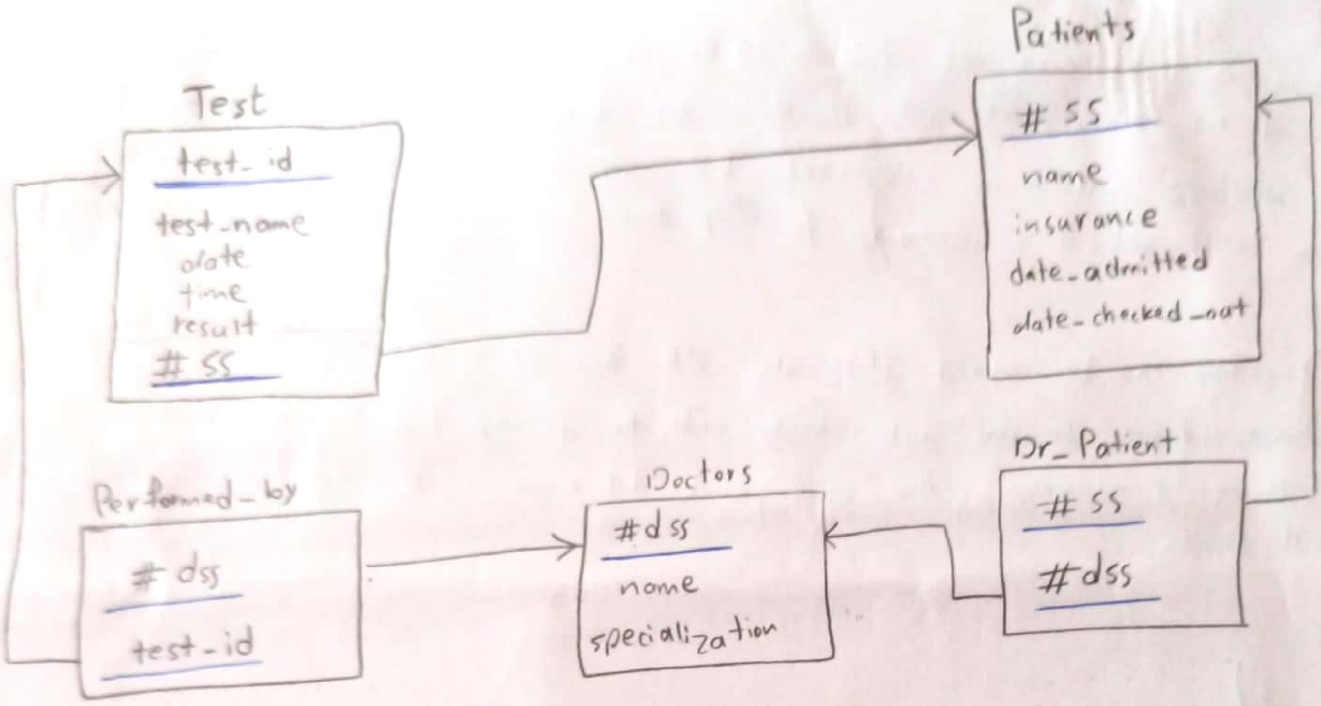
(3) There may not be a way to translate constraints on the ternary relationship into constraints on the binary relationships. For example, consider a constraint that says that R is many-to-one from A, B to C; that is, each pair of entities from A and B is associated with at most one C entity. This constraint cannot be expressed by using cardinality constraints on the relationship sets R_A, R_B, and R_C.

As an example for the University dataset in the book, the relationship set "proj-guide" (relating "instructor", "student" and "project") cannot be directly split into binary relationships between "instructor" and "project" & between "instructor" and "student". If we did so, we would be able to record that instructor "katz" works on project A & B with student Shankar & Zhang. However, we would not be able to record that katz works on project A with student Shankar and also works on project B with student Zhang, but does not work on project A with Zhang or on project B with Shankar. Instead, the relationship set "proj-guide" can be split into binary relationships by creating a new entity set as described above. However, still, doing so would not be very natural!

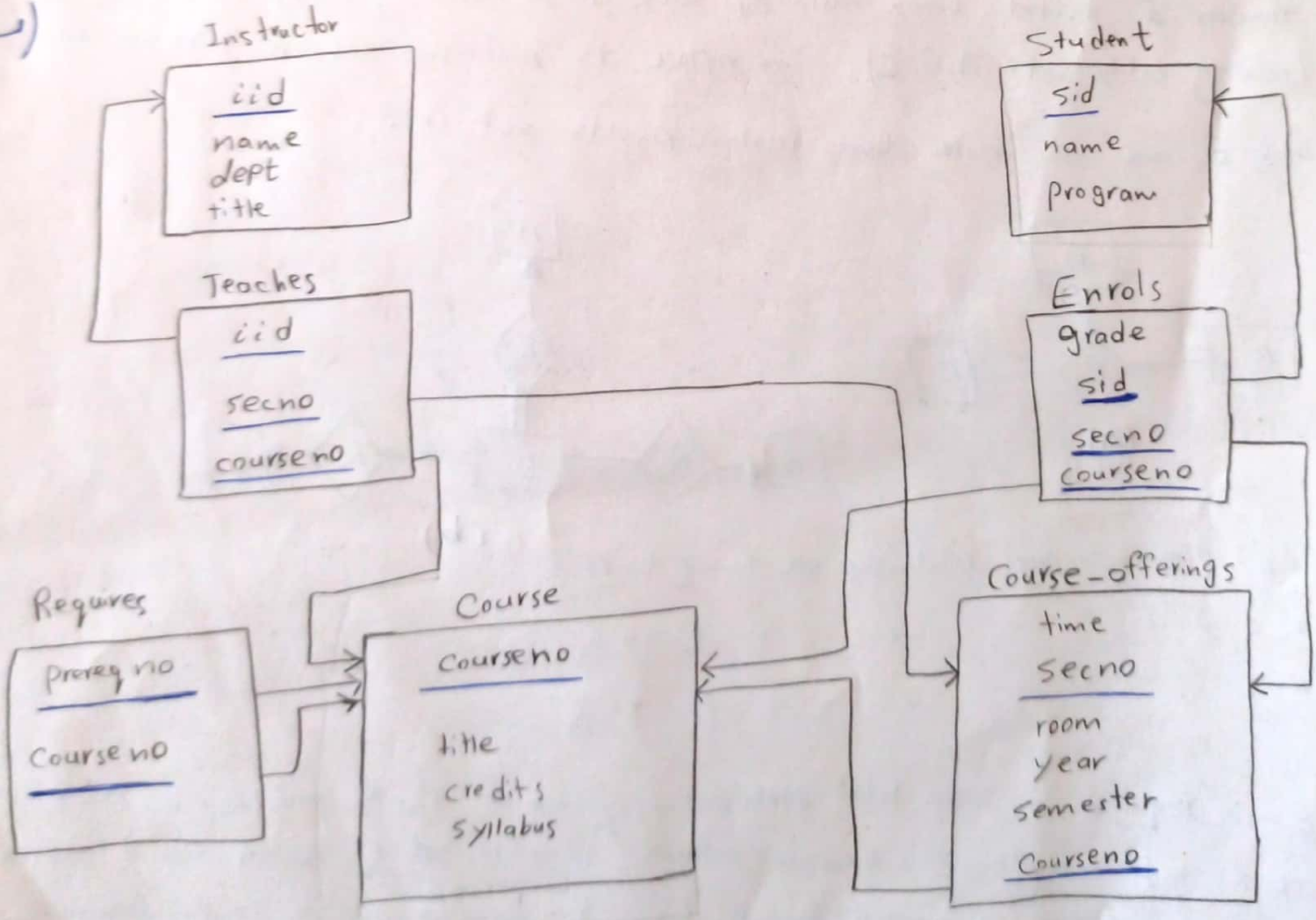
for the Quaternary relationships, the same idea holds true as descussed.

Question 4)

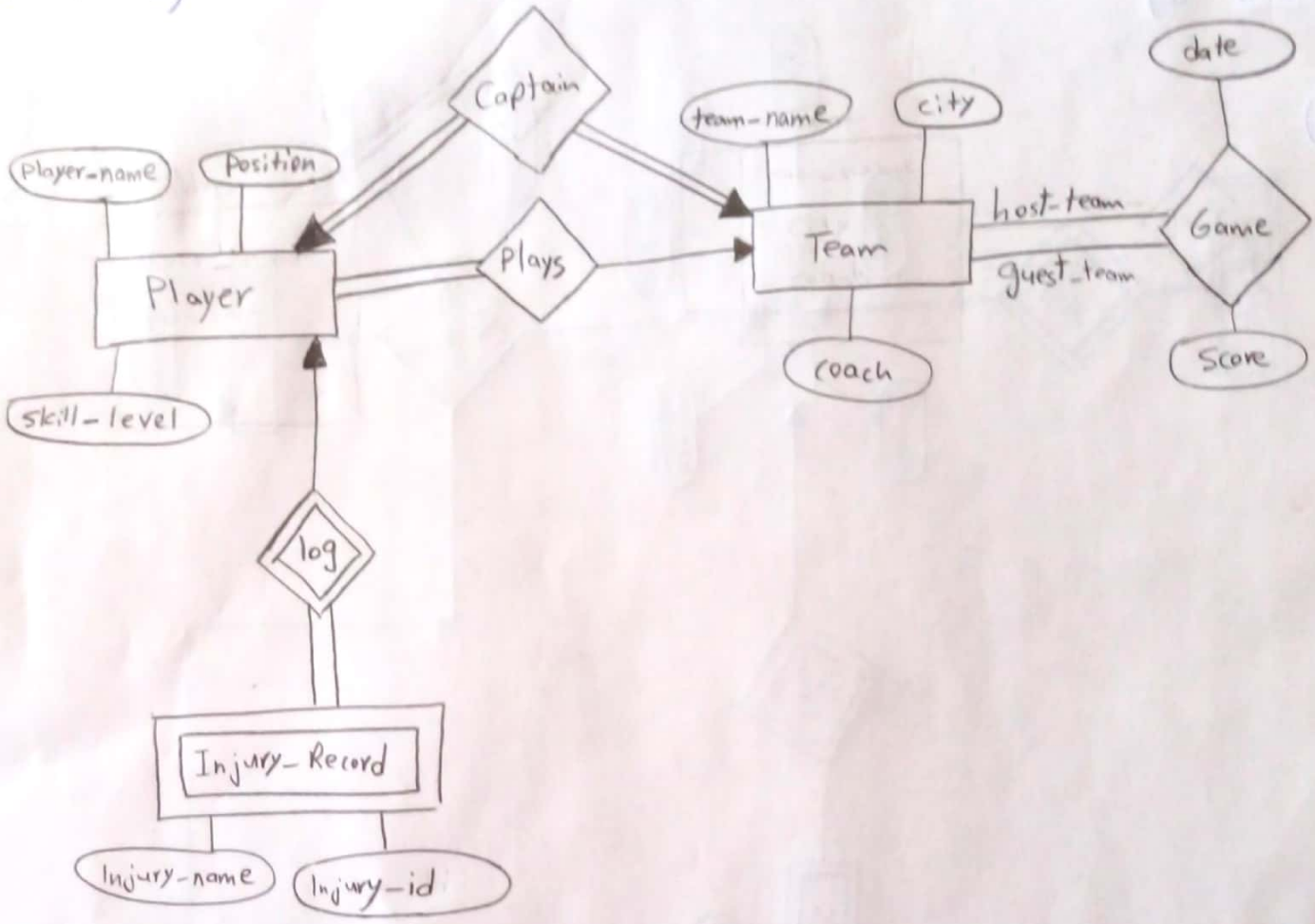
ا)



ب)

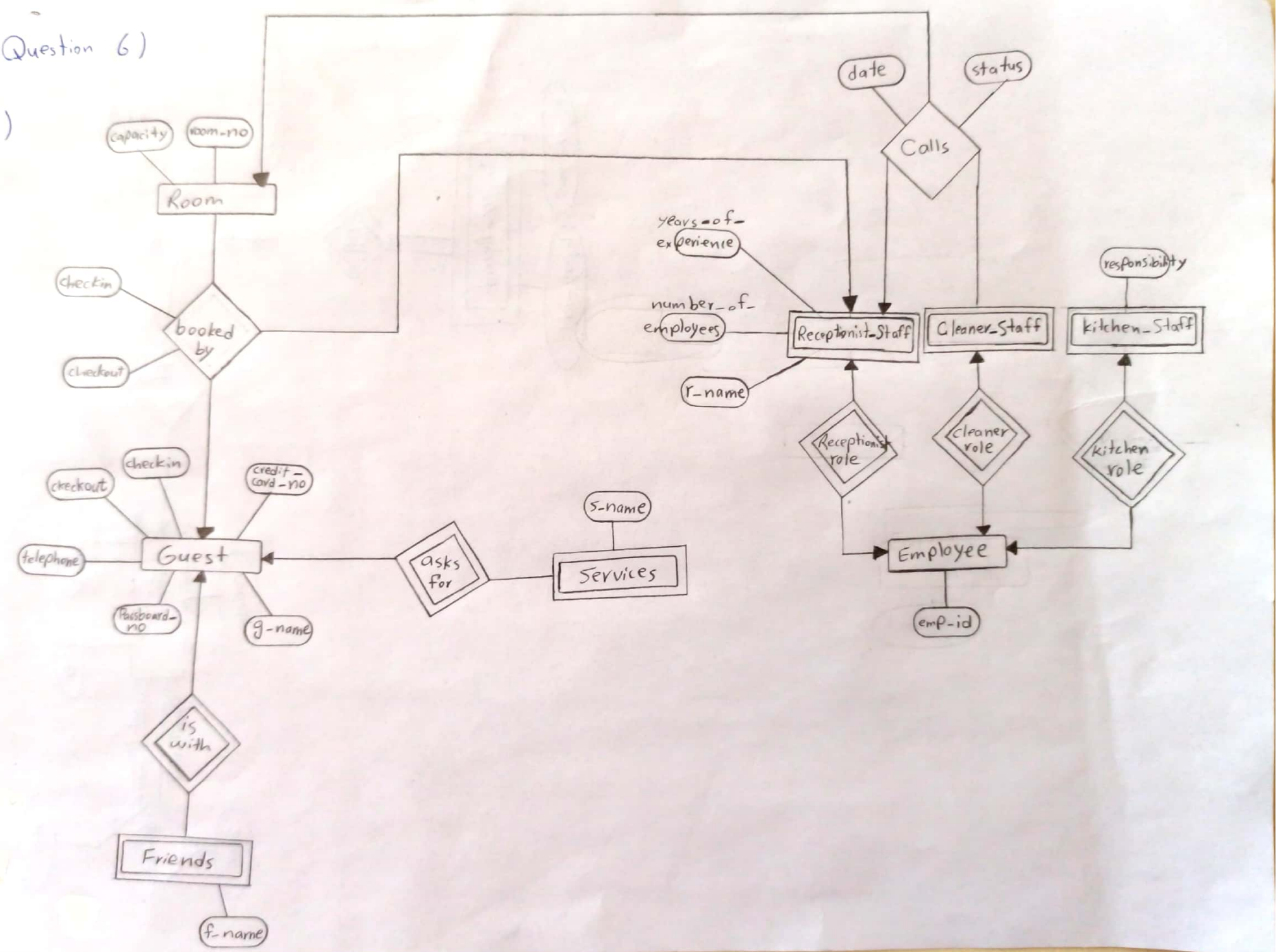


Question 5)



Question 6)

الف)



ب)

As can be seen in the related ER-Diagram, the weak and strong entities are:

weak Entity	strong Entity
Kitchen_Staff	Employee
Cleaner_Staff	Employee
Receptionist_Staff	Employee
Friends	Guest