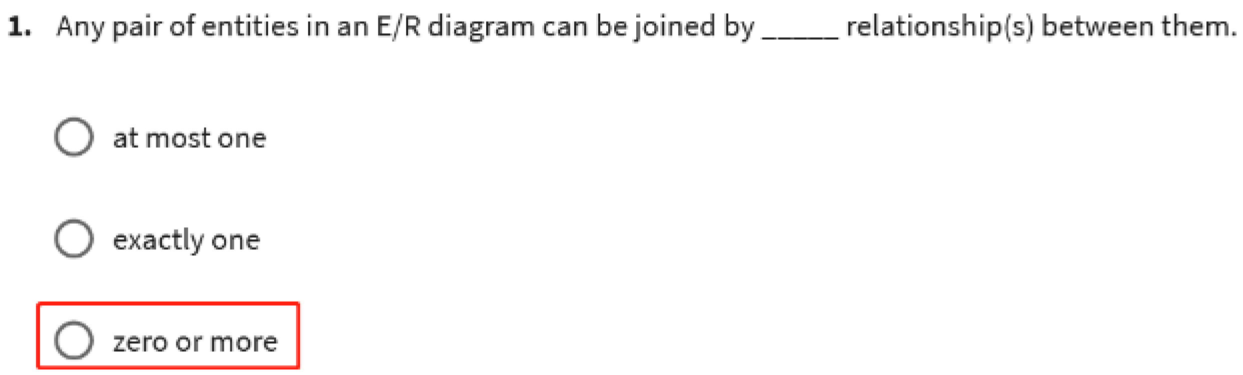
Week 2 Conceptual Data Modeling

Assignment Solutions

Question 1



1. Any pair of entities in an E/R diagram can be joined by _____ relationship(s) between them.

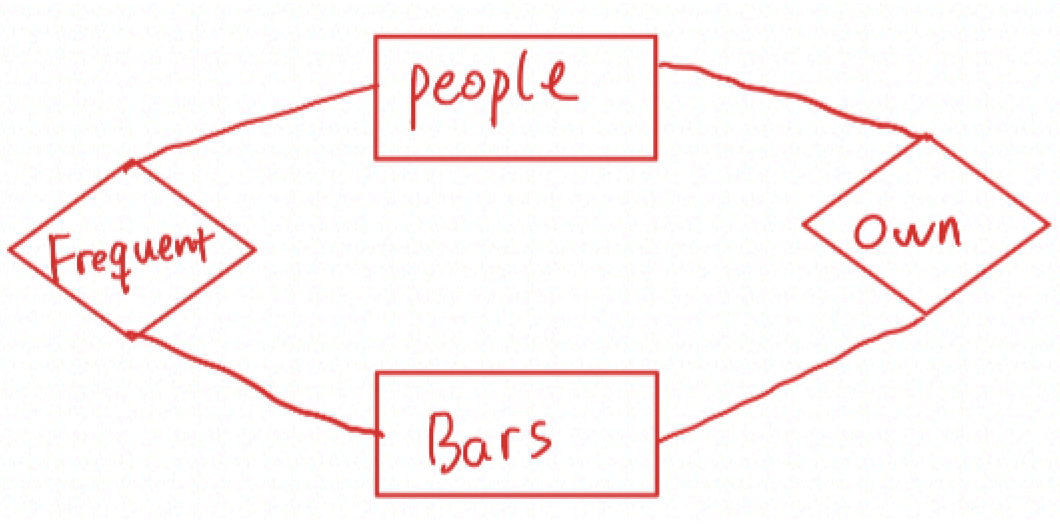
○ at most one

○ exactly one

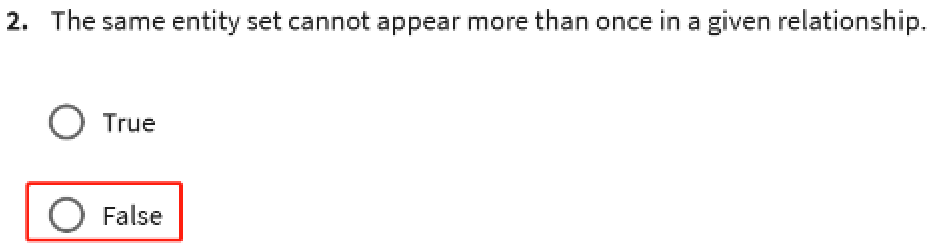
[✓ ○ zero or more]

Solution:

Any arbitrary pair of entities in an ER diagram can have no relationship between them. It is also possible for them to be connected by more than one distinct relationship. Below is a two-relationship example which shows people can frequent certain bars and also own some of the bars.



Question 2



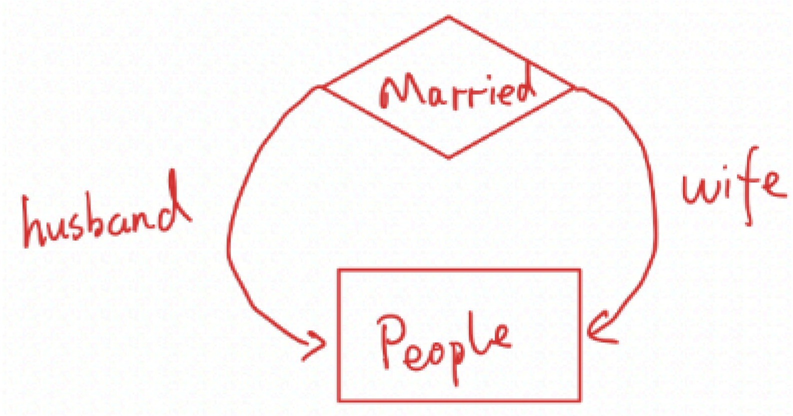
2. The same entity set cannot appear more than once in a given relationship.

○ True

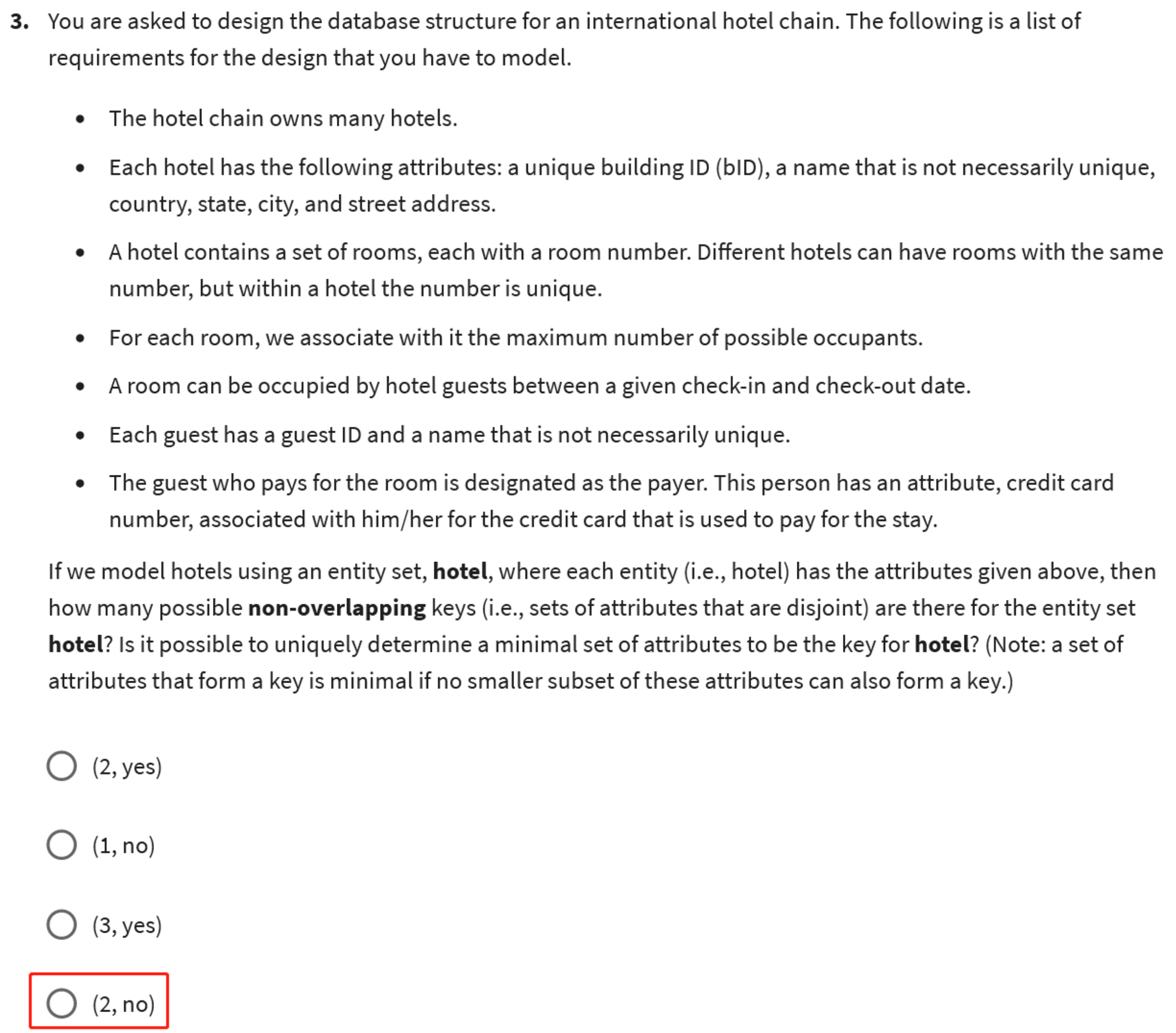
[✓ ○ False]

Solution:

It is possible for an entity set to relate to itself. Usually roles will be added to the edges of the relationship to differentiate the various roles an entity can play in that relationship. For example, “people” can be either the husband or the wife in the “married” relationship.



Question 3



3. You are asked to design the database structure for an international hotel chain. The following is a list of requirements for the design that you have to model.

- The hotel chain owns many hotels.
- Each hotel has the following attributes: a unique building ID (bID), a name that is not necessarily unique, country, state, city, and street address.
- A hotel contains a set of rooms, each with a room number. Different hotels can have rooms with the same number, but within a hotel the number is unique.
- For each room, we associate with it the maximum number of possible occupants.
- A room can be occupied by hotel guests between a given check-in and check-out date.
- Each guest has a guest ID and a name that is not necessarily unique.
- The guest who pays for the room is designated as the payer. This person has an attribute, credit card number, associated with him/her for the credit card that is used to pay for the stay.

If we model hotels using an entity set, **hotel**, where each entity (i.e., hotel) has the attributes given above, then how many possible **non-overlapping** keys (i.e., sets of attributes that are disjoint) are there for the entity set **hotel**? Is it possible to uniquely determine a minimal set of attributes to be the key for **hotel**? (Note: a set of attributes that form a key is minimal if no smaller subset of these attributes can also form a key.)

○ (2, yes)

○ (1, no)

○ (3, yes)

[✓ ○ (2, no)]

Solution:

“country”, “state”, "city", “street address”, and “name of hotel” can represent a unique address because people can find any specific hotel with the information listed above, hence these attributes combined will be a key. bID will be another key. So, there are two non-overlapping keys.

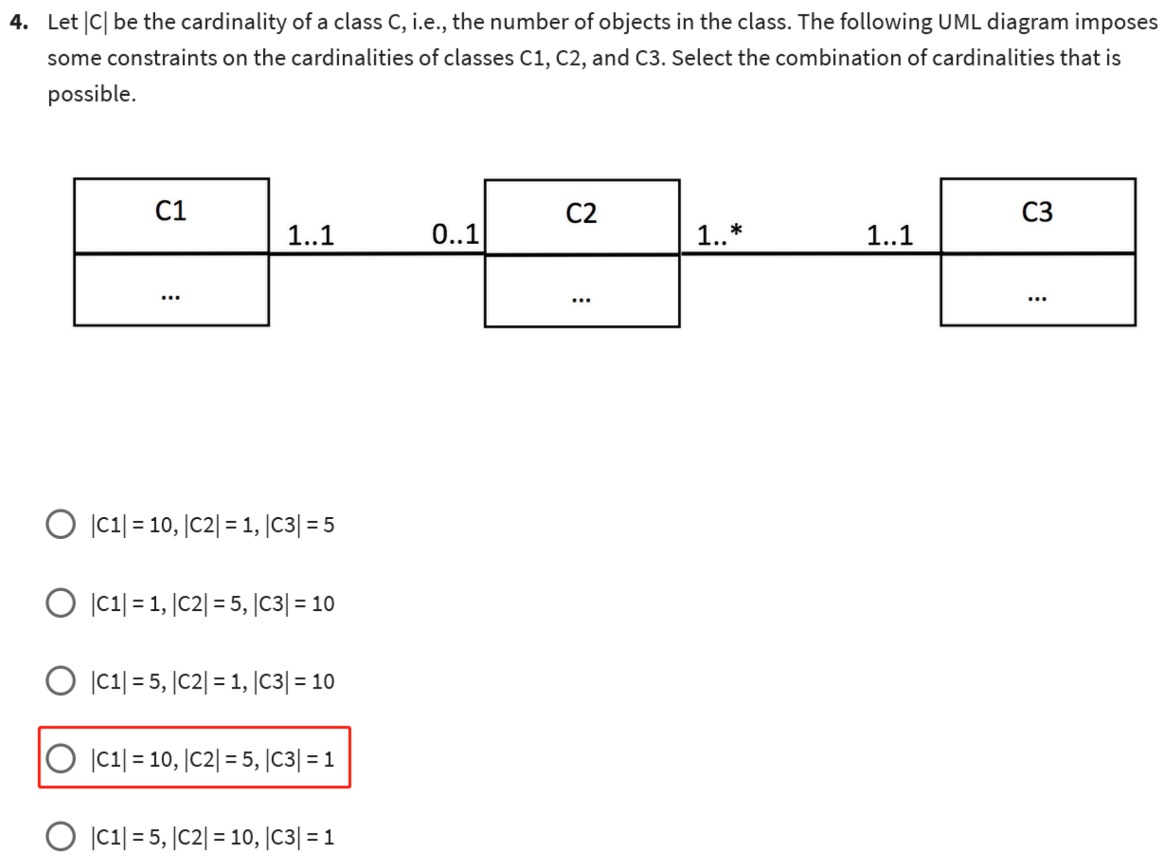
Is it possible to uniquely determine a minimal set of attributes to be the key for hotel?

No.

There is no attribute in the first set of attributes that can be removed such that the remaining attributes can still form a key for hotel. Those attributes have already formed a minimal set of attributes to be the first key. Since there is only one attribute for the second key, it is also a minimal set of attributes for a key.

Now since we have two keys with minimal set of attributes, it is impossible to uniquely determine a minimal set of attributes to be the key for hotel.

Question 4



4. Let |C| be the cardinality of a class C, i.e., the number of objects in the class. The following UML diagram imposes some constraints on the cardinalities of classes C1, C2, and C3. Select the combination of cardinalities that is possible.

○ |C1| = 10, |C2| = 1, |C3| = 5

○ |C1| = 1, |C2| = 5, |C3| = 10

○ |C1| = 5, |C2| = 1, |C3| = 10

○ [✓ |C1| = 10, |C2| = 5, |C3| = 1]

○ |C1| = 5, |C2| = 10, |C3| = 1

Solution:

Each object in C2 must be associated with exactly 1 object in C1.

Each object in C1 can be associated with 0 or exactly 1 object in C2.

This implies |C1| >= |C2| because it's possible for an object in C1 to not be associated with an object in C2.

Similarly, each object in C2 must be associated with exactly 1 object in C3.

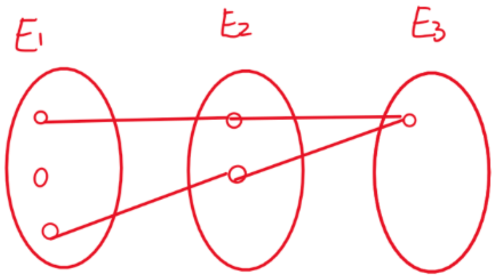
Each object in C3 must be associated with 1 or more object in C2.

This implies |C2| >= |C3|.

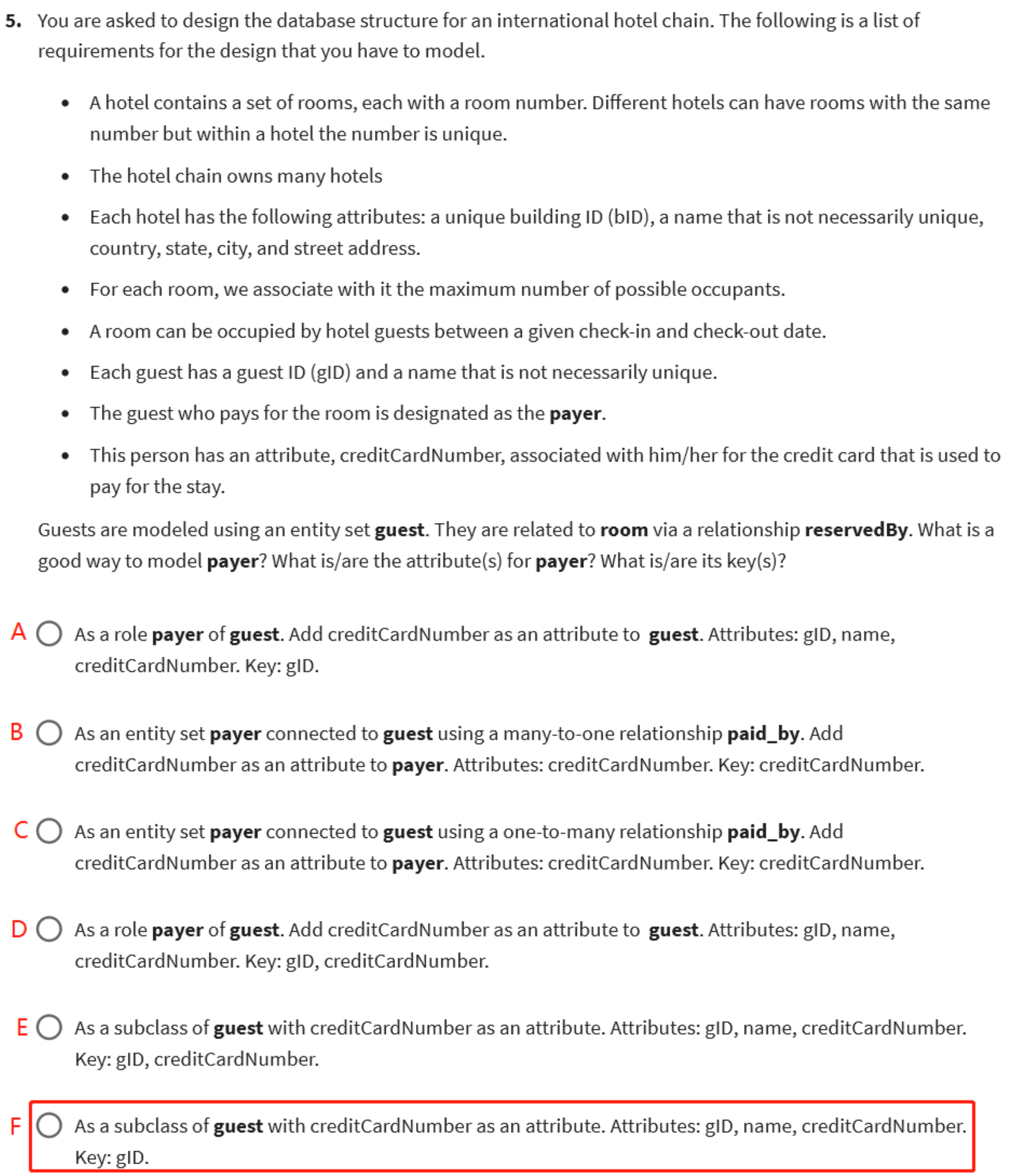
Together we have |C1| >= |C2| >= |C3|

All other options violate this condition.

One possible example:


E1
E2
E3

Question 5



5. You are asked to design the database structure for an international hotel chain. The following is a list of requirements for the design that you have to model.

- A hotel contains a set of rooms, each with a room number. Different hotels can have rooms with the same number but within a hotel the number is unique.
- The hotel chain owns many hotels
- Each hotel has the following attributes: a unique building ID (bID), a name that is not necessarily unique, country, state, city, and street address.
- For each room, we associate with it the maximum number of possible occupants.
- A room can be occupied by hotel guests between a given check-in and check-out date.
- Each guest has a guest ID (gID) and a name that is not necessarily unique.
- The guest who pays for the room is designated as the **payer**.
- This person has an attribute, creditCardNumber, associated with him/her for the credit card that is used to pay for the stay.

Guests are modeled using an entity set **guest**. They are related to **room** via a relationship **reservedBy**. What is a good way to model **payer**? What is/are the attribute(s) for **payer**? What is/are its key(s)?

A ○ As a role **payer** of **guest**. Add creditCardNumber as an attribute to **guest**. Attributes: gID, name, creditCardNumber. Key: gID.

B ○ As an entity set **payer** connected to **guest** using a many-to-one relationship **paid_by**. Add creditCardNumber as an attribute to **payer**. Attributes: creditCardNumber. Key: creditCardNumber.

C ○ As an entity set **payer** connected to **guest** using a one-to-many relationship **paid_by**. Add creditCardNumber as an attribute to **payer**. Attributes: creditCardNumber. Key: creditCardNumber.

D ○ As a role **payer** of **guest**. Add creditCardNumber as an attribute to **guest**. Attributes: gID, name, creditCardNumber. Key: gID, creditCardNumber.

E ○ As a subclass of **guest** with creditCardNumber as an attribute. Attributes: gID, name, creditCardNumber. Key: gID, creditCardNumber.

F [✓ ○ As a subclass of **guest** with creditCardNumber as an attribute. Attributes: gID, name, creditCardNumber. Key: gID.]

Solution:

A payer in this case is one of the guest as specified in the list of requirements. He/she has additional information - creditCardNumber. So we can model payer as a subclass of guest with this additional attribute. Key of a subclass is the key of the root entity set.

Adding attribute to guest means every guest will have this attribute but we wish to model only a subset of them - the ones who pay - to have it. – A is wrong.

This option does not make sense as it says that a guest's stay is paid by many payers, which does not match the requirements of the problem. – B is wrong.

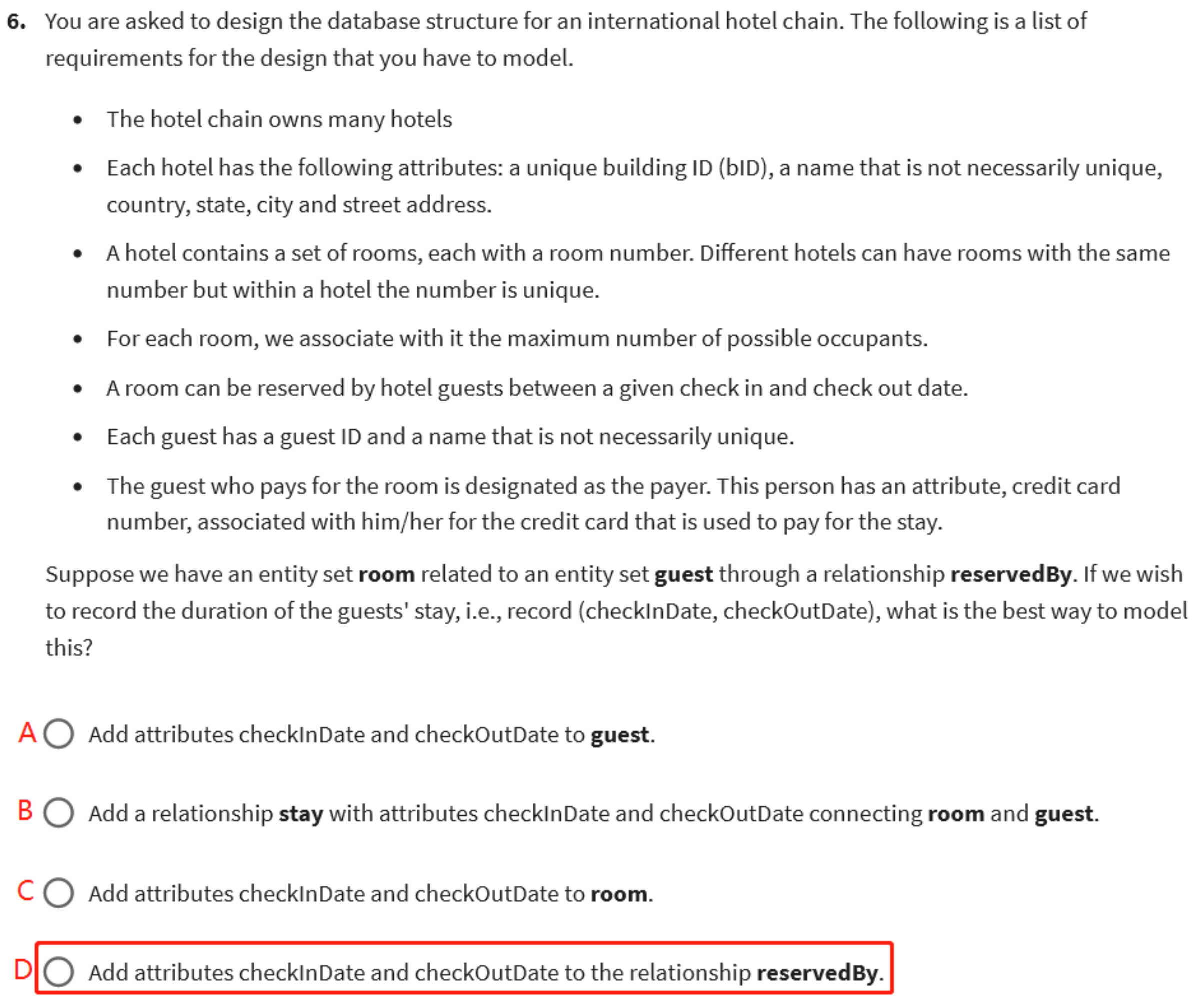
The requirement states the guest who pays for the room is designated as the payer. That is, a payer is one of the guests. – C is wrong.

Adding attribute to guest means every guest will have this attribute but we wish to model only a subset of them - the ones who pay - to have it. Also, if payer is a role of guest, then key should be that of guest, and so should just be gID. – D is wrong.

Key of a subclass is the key of the root entity set. – E is wrong.

F is correct.

Question 6



6. You are asked to design the database structure for an international hotel chain. The following is a list of requirements for the design that you have to model.

- The hotel chain owns many hotels
- Each hotel has the following attributes: a unique building ID (bID), a name that is not necessarily unique, country, state, city and street address.
- A hotel contains a set of rooms, each with a room number. Different hotels can have rooms with the same number but within a hotel the number is unique.
- For each room, we associate with it the maximum number of possible occupants.
- A room can be reserved by hotel guests between a given check in and check out date.
- Each guest has a guest ID and a name that is not necessarily unique.
- The guest who pays for the room is designated as the payer. This person has an attribute, credit card number, associated with him/her for the credit card that is used to pay for the stay.

Suppose we have an entity set **room** related to an entity set **guest** through a relationship **reservedBy**. If we wish to record the duration of the guests' stay, i.e., record (checkInDate, checkOutDate), what is the best way to model this?

A ○ Add attributes checkInDate and checkOutDate to **guest**.

B ○ Add a relationship **stay** with attributes checkInDate and checkOutDate connecting **room** and **guest**.

C ○ Add attributes checkInDate and checkOutDate to **room**.

[✓ D ○ Add attributes checkInDate and checkOutDate to the relationship **reservedBy**.]

Solutions:

It is given in requirement item 5 that a room can be reserved by guests between a given check in and check out date. These dates are part of the information associated with each reservation.

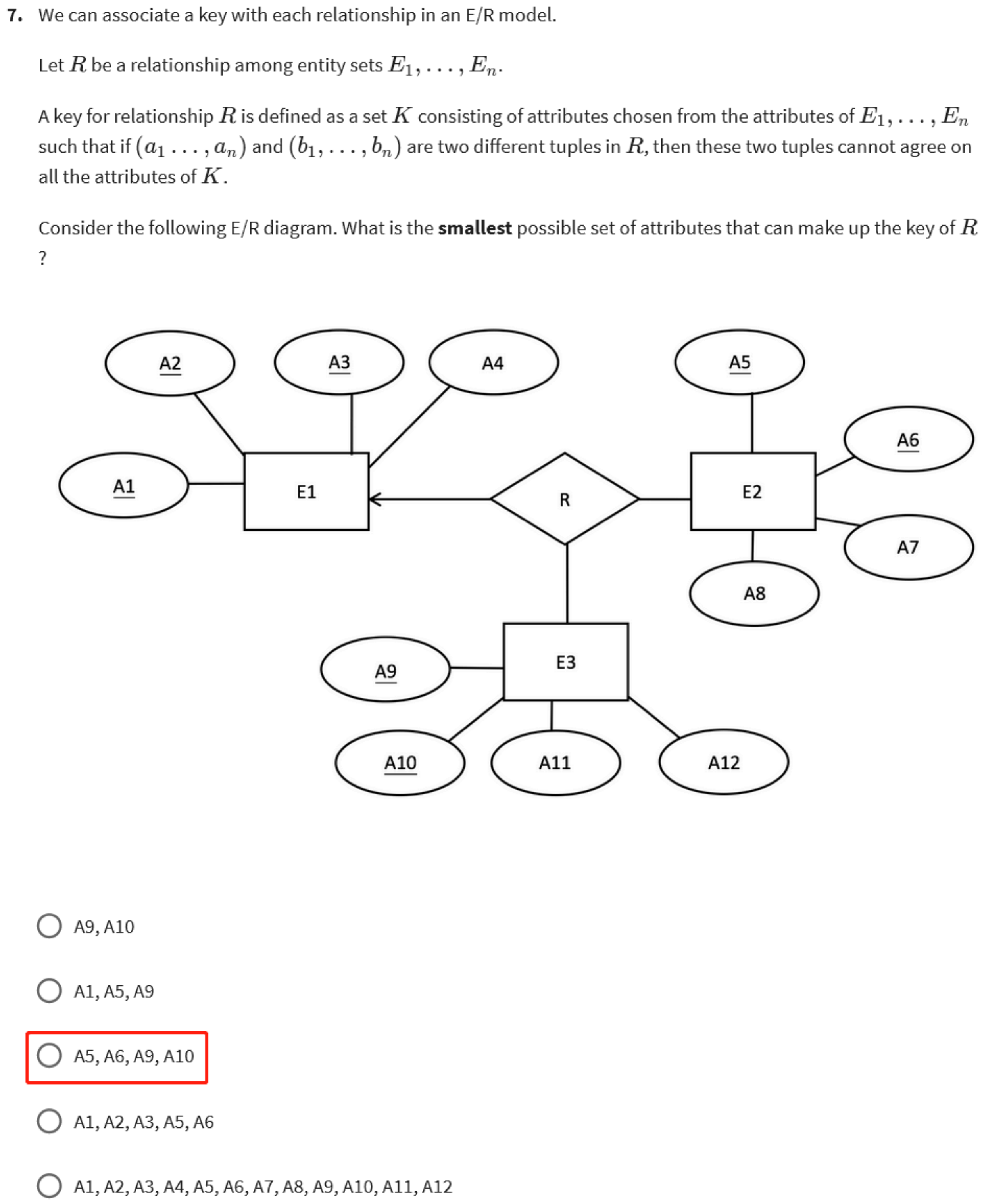
A means the check in and check out information is associated with each guest and not with the reservation of a room. It cannot model the case where a guests reserve more than one room with different check in and check out dates. The idea is that the check in check out information is part of the reservation as given in the requirement. – A is wrong.

It is given in the question that there is already a reserveBy relationship between guest and room. We do not need to introduce a new relationship as we can associate the check in check out information with reserveBy. – B is wrong.

C means a room can only have one check in and check out date. – C is wrong.

D is correct.

Question 7



7. We can associate a key with each relationship in an E/R model.

Let R be a relationship among entity sets $E_1, \ldots, E_n$.

A key for relationship R is defined as a set K consisting of attributes chosen from the attributes of $E_1, \ldots, E_n$ such that if $(a_1 \ldots, a_n)$ and $(b_1, \ldots, b_n)$ are two different tuples in R, then these two tuples cannot agree on all the attributes of K.

Consider the following E/R diagram. What is the **smallest** possible set of attributes that can make up the key of R ?

○ A9, A10

○ A1, A5, A9

○ [✓ A5, A6, A9, A10]

○ A1, A2, A3, A5, A6

○ A1, A2, A3, A4, A5, A6, A7, A8, A9, A10, A11, A12

Solution:

Notice the constraint here – It’s a ‘0 or 1 to many to many’ relationship. E2, E3 are on the many side.

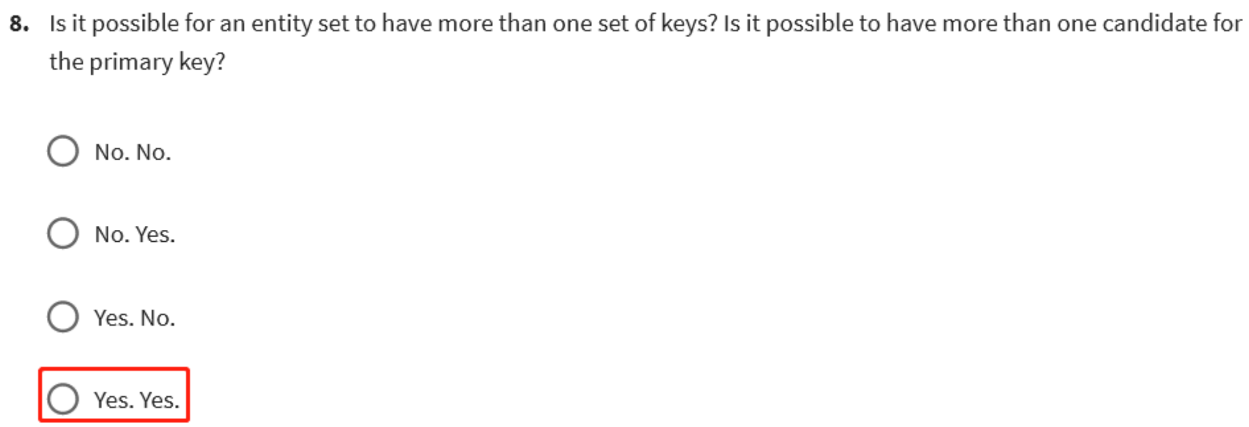
A5 & A6 form the key for E2 and A9&A10 form the key for E3.

If two tuples have the same values for attributes A5, A6, then they must have the same value for A7, A8 (because A5, A6 are keys of E2). Similar situation for A9, A10 for E3.

Therefore, if two tuples have the same values for A5, A6, A9, A10, they must also have the same values for attributes A7, A8 and A11, A12. This means their corresponding entities drawn from E2 and E3 are the same.

Lastly, from the multiplicity constraints given in the diagram, we know that each pair of entities from E2 and E3 can only be associated with 0 or 1 entity in E1. This means once we fixed values for A5, ..., A12, there can only be one (or 0) set of values for A1, ..., A4. So, if two tuples have the values for A5, ..., A12, their values for A1, ..., A4 must also be the same, which means they are the same tuples. All this from knowing that their values for A5, A6, A9, A10 are the same. You can reason it out that it’s not possible to pick any smaller set of attributes to “force” the remaining attributes to take on the same values. Hence smallest key is {A5, A6, A9, A10}.

Question 8



8. Is it possible for an entity set to have more than one set of keys? Is it possible to have more than one candidate for the primary key?

○ No. No.

○ No. Yes.

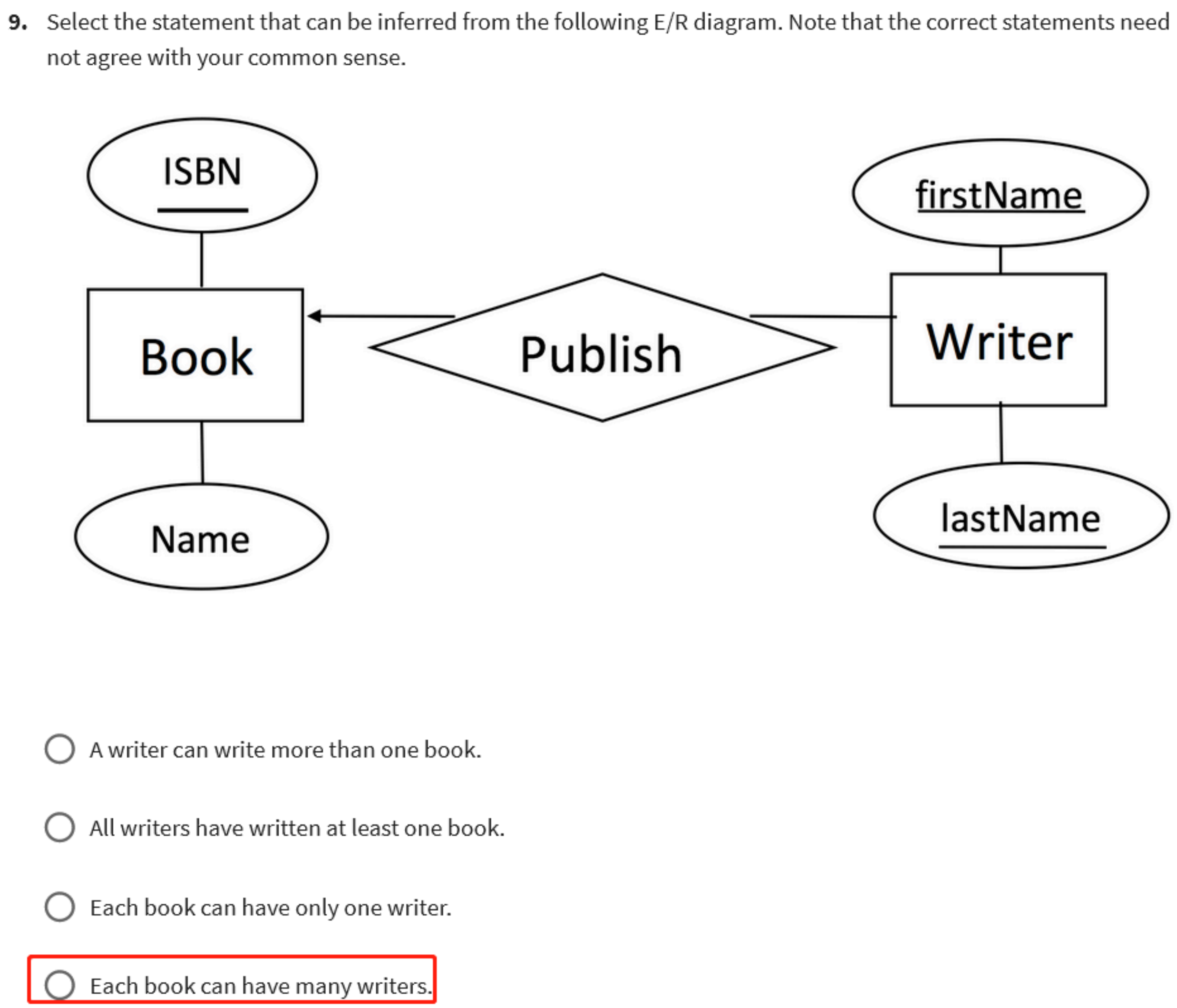
○ Yes. No.

○ [✓ Yes. Yes.]

Solution:

Suppose we have a “Student” entity set. The attributes can be ‘University ID’ and “Network ID”. We have two candidates for the primary key – either ‘University ID’ or ‘Network ID’ can be the primary key for this entity set.

Question 9



9. Select the statement that can be inferred from the following E/R diagram. Note that the correct statements need not agree with your common sense.

○ A writer can write more than one book.

○ All writers have written at least one book.

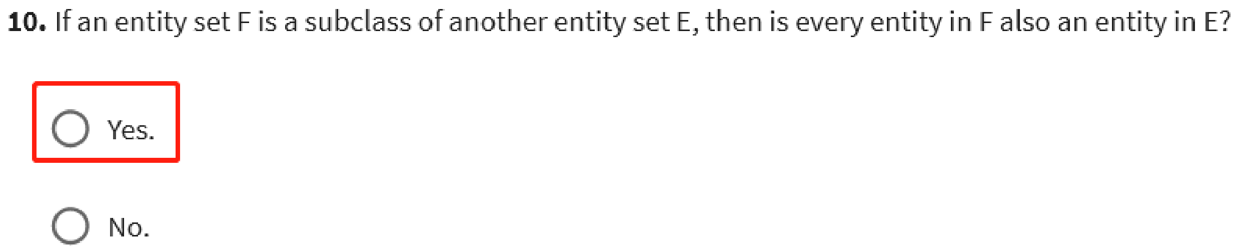
○ Each book can have only one writer.

[✓ ○ Each book can have many writers.]

Solution:

The multiplicity constraint has Writer on the “many” side and Book on the “one” side so a book can have more than one writer.

Question 10



10. If an entity set F is a subclass of another entity set E, then is every entity in F also an entity in E?

[✓ ○ Yes.]

○ No.

Solution:

This is part of the property of a subclass.