HQ_1_1

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

✓ zero or more

- Any arbitrary pair of entities in an ER diagram can be unconnected by a relationship. It is also possible for them to be connected by more than 1 distinct relationships.

✗ at most one

✗ exactly one

HQ_1_2

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

✓ False

- 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.

✗ True

HQ_1_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.

1. The hotel chain owns many hotels.
2. Each hotel has the following attributes: a unique building ID (bID), a name that is not necessarily unique, country, state, city and street name.
3. 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.

4. For each room, we associate with it the maximum number of possible occupants.
5. A room can be occupied by hotel guests between a given check-in and check-out date.
6. Each guest has a guest ID and a name that is not necessarily unique.
7. The guest who pays for the room is designated as the payer. This person has an attribute, credit card number, associated with 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 set of attributes that also form a key exists.)

✓ (2, yes)

- It depends on the definition of key. If key is not necessarily minimal, and if “country”, “state”, “street name”, and “name of hotel” is a unique address (as if people use mailing address to deliver mail to specific locations) then these attributes will be a key and bID will be another. So there are two keys, and bID is the unique smallest one.

✗ (3, yes)

✗ (3, no)

✗ (2, no)

✗ (1, no)

✗ (1, yes)

HQ_1_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| = 5, |C3| = 1

- 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.

✗ |C1| = 5, |C2| = 10, |C3| = 10

✗ |C1| = 5, |C2| = 10, |C3| = 1

✗ |C1| = 5, |C2| = 1, |C3| = 10

✗ |C1| = 10, |C2| = 5, |C3| = 10

✗ |C1| = 1, |C2| = 5, |C3| = 10

✗ |C1| = 10, |C2| = 1, |C3| = 5

HQ_1_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.

1. The hotel chain owns many hotels
2. Each hotel has the following attributes: a unique building ID (bID), a name that is not necessarily unique, country, state, city and street name.
3. For each room, we associate with it the maximum number of possible occupants.
4. A room can be occupied by hotel guests between a given check in and check out date.
5. Each guest has a guest ID (gID) and a name that is not necessarily unique.

6. 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 modelled 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)?

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

- 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.

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

- Key of a subclass is the key of the root entity set.

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

- 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.

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

- 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.

✗ 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.

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

✗ 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.

- 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.

HQ_1_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.

1. The hotel chain owns many hotels
2. Each hotel has the following attributes: a unique building ID (bID), a name that is not necessarily unique, country, state, city and street name.
3. 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.
4. For each room, we associate with it the maximum number of possible occupants.
5. A room can be reserved by hotel guests between a given check in and check out date.
6. Each guest has a guest ID and a name that is not necessarily unique.
7. 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?

✓ Add attributes checkInDate and checkOutDate to the relationship **reservedBy**.

- 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.

✗ Add attributes checkInDate and checkOutDate to **room**.

- This means a room can only have one check in and check out date.

✗ Add attributes checkInDate and checkOutDate to **guest**.

- This means the check in 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.

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

- 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.

HQ_1_7

We can associate a key with each relationship in an E/R model. Let R be a relationship among entity sets $E_1,\ldots,E_n E1,\ldots,En$. A key for relationship R is defined as a set K consisting of attributes chosen from the attributes of $E_1,\ldots,E_n E1,\ldots,En$ such that if $(a_1\ldots,a_n)(a1\ldots,an)$ and $(b_1,\ldots,b_n)(b1,\ldots,bn)$ 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?

✓ A5, A6, A9, A10

- A relationship set R can be viewed as containing tuples of entities (e1, e2, e3) where ei is from Ei, i in {1, 2, 3}. A set of attributes K taken from attributes of E1, E2, E3 is a key for R if no two different tuples in R can have the same values for those attributes in K. The contrapositive is if two tuples in R have the same values for the attributes in K then they must be the same. We “expand” a tuple in R to contain the attributes of those entities. 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 have 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}. Also note that any non-key attribute from the entity sets will not be able to constrain the values of any other attributes and hence will not be useful as an attribute in the key of R. This is the only solution.

✗ A1, A2, A3

✗ A1, A2, A3, A4

✗ A5, A6

✗ A5, A9

✗ A9, A10

✗ A1, A5, A9

✗ A1, A2, A3, A5, A6

✗ A1, A2, A3, A9, A10

✗ A1, A2, A3, A5, A6, A9, A10

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

HQ_1_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?

✓ (yes, yes)

- If an entity set has 3 attributes A1, A2, A3 and A1 is a key and A2 is another key, then these two are candidates for the primary key. All other answers are incorrect.

✗ (yes, no)

✗ (no, yes)

✗ (no, no)

HQ_1_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.

✓ Each book can have many writers.

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

✓ Each writer can write at most one book.

- The multiplicity constraint has Writer on the “many” side and Book on the “one” side so a writer can write 0 or 1 book.

✓ Some writers may not have written any books.

- The multiplicity constraint has Writer on the “many” side and Book on the “one” side so a writer can write 0 or 1 book.

✗ Each book can have only one writer.

- A book can have more than one writer base on the multiplicity constraint.

✗ A writer can write more than one book.

- A writer can write 0 or 1 book based on the multiplicity constraint.

✗ All writers have written at least one book.

- A writer can write 0 or 1 book based on the multiplicity constraint. So some of them may not have written any books.

HQ_1_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

- This is part of the property of a subclass.

✕ No
