Question #1: Regular entity type

Respondant: Each regular entity type in an ER diagram is transformed into a relation. The name given to the relation is generally the same as the entity type. The identifier of the entity type becomes the primary key of the corresponding action

Respondant: A regular entity type is transformed to relations by giving the same name of the entity type to the relation. The attributes of the entity become attributes of the relation and the identifier becomes the primary key.

Respondant nancyhu: each regular entity type in an ER diagram is transformed into a relation. The name given to the relation is generally the same as the entity type. Each simple attribute of the entity type becomes the primary key of the corresponding relation.The identifier of the entity type becomes the primary key of the corresponding relation.

Respondant attribute of the entity type becomes an attribute of the relation. The identifier of the entity type becomes the primary key of the corresponding relation.

Respondant: Each regular entity in ER diagram is is transfed into a relation The name given to the relation is the same as the Entity type. Echch simple arribute of the entity type becoms an atrribute of relation. The identifier of the entity type became the primery key of the corresponding relation.

Respondant: First of all every entity type in a ER-diagram is transformed into a relation. Name is transferred directly, identifier becomes the primary key and single attributes become the attributes of the relation.

Respondant: The name of the entity is given to the relation and each simple attribute in the entity becomes an attribute in the relation.

Respondant: Each entity type is turned into a relation.
The attributes of the entity type become attributes of the relations: they decribe the relation. The identifier of the identity type is used as the primary key for the relation created. If there are composite or multi-valued attributes see question 5 and 7.

Respondant: The name given to the relation is generally the same as the entity type. Each simple attribute of the entity type becomes an attribute of the relation. The identifier of the entity type becomes the primary key of the corresponding relation.

Respondant: First, the name given to the relation is generally the same as the entity type. Then, each simple attribute of the entity type becomes an attribute of the relation. The identifier of the entity type becomes the primary key of the corresponding relation.

Respondant: The name given to the relation is generally the same as the entity type. Each simple attribute of the entity type becomes an attribute of the relation. The identifier of the entity type becomes the primary key of the corresponding relation.

Respondant: Each simple attribute of the entity type becomes an attribute of the relation. The identifier of the entity type becomes the primary key of the corresponding relation. Regular entity having a composite attribute, only the simple component attributes of the composite attribute are included in the new relation. If regular entity type contains a multivalued attribute, two new relations are created. First relation containing all of the attributes of the entity type except the multivalued attribute. The second relation containing two attributes that form the first relation, which becomes a foreign key in the second relation.

Respondant: The relation's name is usually the same as the entity type. So, each simple attribute of the entity type becomes and attribute of the relation. The identifier of the entity type becomes the primary key of the corresponding relation.

Respondant: Entity types become the header of its relationship block.

Respondant: The given to the relation is the same as the entity type. Represent real-world objects such as persons and products.

Respondant: It usually keeps it's name. The identifier becomes the primary key.

Respondant: Each attribute of the entity tpe becomes an attribute of the relation. The indentifier of the entity type becomes the primary key of the corresponding relation.

Respondant: Each simple attribute of the enity type becomes an attribute of the relation. The identifier of the entity type becomes the primary key of the corresponding relation. Relation name is usually same as enity type.

Respondant: the relation has the same name as the enity type. attributes of the entity type become the attributes of the relation. the identifier of the entity becomes the primary key of the relation.

Respondant: First, each regular entity type in an ER diagram trasnformed into realtion. The name of the relation is the same with the entity type. The indentifier of the intety type become the primary key of the corresponding relation.

Respondant: Each regular entity type is transformed into a relation. The name on the entity becomes the relation and all of the attributes that go with the entity become attributes of the relation. The identifier attribute becomes the primary key.

Respondant: Regular entities are entities that have an independent existence and generally represent real-world objects such as persons and products. Regular entity types are represented by rectangles with a single line.

Respondant: Each entity type is transformed into a relation with each simple attribute of the entity type becomes an attribute of the relation. The identifier becomes the primary key of the corresponding relation.

Respondant: The name given to the relation is the same as the entity type. Each attribute of the entity type becomes an attribute of the relationl. The identifier of the entity type becomes the primary key of the relation.

Respondant: After defining the identifier of the entity as the primary key for the relation, attributes of the entity type is transformed to an attribute for the relation. Only the simple attributes in a composite attribute of an entity is transformed to that of relation

Respondant: The name used for the regular entity type in a relation is the same the one used in the ER diagram. The regular entity then becomes a relation. The attributes in the ER diagram are also directly converted to become attributes of the relation. The identifier of the entity type becomes the primary key of the relation.

Respondant becomes an attribute of the relation and the identifier becomes the primary key of the relation.

Respondant: Each regular entity type in an ER diagram is transformed into a relation. The name given to the relation is generally the same as the entity type.

Question #2: Relationship (1:M)

Respondant: For each binary 1:M relationship, first create a relation for each of the two entity types participating in the relationship. Next, include the primary key attribute of the entity on the one-side of the relationship as a freign key in the realtion that is on the many-side of the relationship.

Respondant: A relation is created for each of the two entity types. The primary key attribute for one entity on oneside of the relationship becomes a foreign key on the relation that is on the many side of the relationship.

Respondant: create a relation for each of the entity types. using the method of the regular entity type's transformation. then include the primary key attribute of the entity on the one-side of the relationship as a foreign key in the relation that is on the many-side of the relationship. the foreign key is indicated with an arrow.

Respondant: For each binary 1:M relationship, first create a relation for each of the two entity types participating in the relationship. Next, include the primary key attribute (or attributes) of the entity on the one-side of the relationshop as a foreign key in the relation that is on the many-side of the relationship (a mnemonic you can use to remember this rule is this: The primary key migrates to the many side).

Respondant: Binary:
Fisrt creat a relation for each of 2 entity types participating in the relationship. Then in clude the primary key attribute of the entityon the one side of the relation as a forein key in the relation that is on the many-side of the relation.

Respondant: First individual relations are created for every entity in the 1:M relationship in same way as in the case of regular entity type. Then the primary key attribute from the one-side of the relationship is used as foreing key in the many-side of the relationship.

Respondant: First a relation must be made for all of the entity types participating in the relationship. The primary key of the one-side of the relationship is placed as a foreign key in the relation on the many-side of the relationship.

Respondant: There are 2 kinds of one-to-many relationships.
-BINARY RELATIONSHIP:
2 relations are created, one for each participating entity. The primary key of the relation which is on the 'one side' of the one-to-may relationship is included in the other relation as a foreign key.
-UNARY RELATIONSHIP:
Only one relation is created. It represents the entity involved in the unary relationship.
The identifier of the entity becomes the primary key of the relation but a second key is added too: it is a foreign key attribute within the same relation and it references to the primary key.

Respondant: First create a relation for each of the two entity types participating in the relationship, using the procedure described for regular entity types. Next, include the primary key attribute (or attributes) of the entity on the one-side of the relationship as a foreign key in the relation that is on many-side of the relationship.

Respondant: For each binary 1:M relationship, first create a relation for each of the two entity types participating in the relationship, using the same procedures described for the regular entity type. Next, include the primary key attribute/attributes of the entity on one-side of the relationship as a foreign key in the relation that is on many-side of the relationship. The foreign key relationship is indicated with an arrow.

Respondant: Create a relation for each of the two entity types particitpating in the relationship, using the procedure described for a regular entity. Next, include the primary key attribute (or attributes) of the entity on the one-side of the relationship as a foreign key in the relation that is on the the many-side of the relationship.

Respondant: First create a relation for each of the two entity types participating in the relationship. Next include the primary key attribute of the entity on the one-side of the relationship as a foreign key in the relation that is on the many-side of the relationship.

Respondant: For each 1:M relationship, we use the steps used in mapping regular entity and create a relation for each of the two entity types. Then, we include the primary key attribute of the entity on the one-side of the relationship as a foreign key in the relation that is on the many-side of the relationship.

Respondant: The primary key attribute is included as a foreign key attribute on the many side relationship and is then linked pointing toward the primary key attribute.

Respondant: Create a relation for each of the 2 entity types in the relationship, then include the primary key attribute of the entity on the one-side of the relationship as a foreign key.

Respondant: Create a relation for each of two entities. Put the primary key for the one into the many.

Respondant: Create a relation for each of the two entity types participating in the relationship. Include the primary key attribute of the entity on the one-side of the relationship as a foreign key in the relationship that is on the many-side of the relationship.

Respondant: First you must create a relation for each of the two entity types participating in the relationship. Then you include the primary key attribute(s) of the entity on the one-side of the relationship as a forein key in the relation the is on the many-side of the relationship.

Respondant: relations are created for each entity types in the relationship. next include the primary key of the entity on the one side of the relationship as a foreign key in the relation on the many side of the relationship.

Respondant: 1. Create a realtion for each of two entities types participating in realtionship.
2. Include the primary attribute of the entity on the one- side of the realtioship as foreign key.

Respondant: Both entities become relations. Then the primary key attribute on the one-side becomes a foreign key in the relation on the many-side.

Respondant: A relationship between entities. ie. A doctore has 1 to many patients.

Respondant: A relation for each of the two entity types participating in the relationship is created. Then include the primary key of entity on one side of relationship as foreign key . The primary key migrates to th many side.

Respondant: For each binary one-to-many relationship, a relation for each of the two entity types participating in the relationship is created. The primary key attribute of the entity on the one-side of the relationship is included as a foreign key in the relation.

Respondant: We create a relation for each entity particpating in one-to-many relationships, using the same way as transforming entities into relations, and we define the primary key of the one-side entity as the foreign key for the many-side of the relation.

Respondant: This mapping involves a one-to-many relationship.
The relation between each of the two entity types participating in the relationship needs to be created first. This is done by simply mapping the regular entities. The next step would be to include the primary key attribute of the entity from the one-side relationship as a foreign key in a relation that is on the many-side relationship.

Respondant: A relation is created for each of the two entity types. The primary key attribute(s) of entity on one side of relationship is used as the foreign key for the "many" side of the relationship.

Respondant: Fore each binary 1:M relationship, first create a relation for each of the two entity types participating in the relationship. Next, include the primary key attribute of the entity on the one-side of the relationship as a foreign key in the relation that is on the many-side of the relationship.

Question #3: Relationship (M:N)

Respondant: For a binary many-to-many relationship between two entity types A and B, we create a new relationship C. Include as foreign key arrtibutes in C the primary key for each of the two participating entity types. These attributes become the primary key of C. Any nonkey attributes that are associated with the M:N relationship are included with the relation C.

Respondant This is a relation between two entity types. A new relation is created which includes the primary key of both of the entity types as a foreign key. These attributes become the primary key of the new relation.

Respondant: create a new relation C. Include as foreign key attributes in C the primary key for each of the participating entity types. These attributes become the primary key of C. Any nonkey attributes that are associated with the M:N relationship are included with the relation C.

Respondant: For M:N relationship, we create a new relation C. Include as foreign key attributes in C the primary key for each of the two participating entity types. These attributes become the primary key of C. Any nonkey attributes that are associated with the M:N relationship are included with the relation C.

Respondant: Binary;Create a new relationC between two entity type of A and B with each primary attributes. which become the primary key of C.

Respondant: First normal relations are created. Second step is to create additional relation, where the primary is is a compination of primary key attributes from both participating entitytype. Any nonkey attributes associated with the M:N relationship are included with the new relation.

Respondant: A new relation must be created when a many to many relationship is needed. The primary keys from both of the original relations are placed in this new relation. And any nonkey attributes important to the many to many relationship are also placed in this new relation.

Respondant: There are 2 kinds of many-to-may relationships.
-BINARY RELATIONSHIP: In addition to the 2 relations for the participating entity, we create a third table for the relationship. The 2 primary keys of the first 2 relations are included as foreign keys. They defined the primary key of the new relation.
- UNARY RELATIONSHIP:
Two relations are created. There is one for the entity involved in the unary relationship. There is also an associative relation to represent the relationship.
In this second one, the primary key is made of two attributes that reference the primary key of the first relation.

Respondant: If there was a binary many-to-many relationship between two entity types, create a new relation. Include as foreign key attributes in the new relation, the primary key for each of the two participating entity types. The attributes become the primary key of the new relation. Any nonkey attributes that are associated with the M:N relationship are included with the new relation.

Respondant: For a binary many-to-many (M:N) relationship between two entity types A and B, we create a new relation C. Included as foreign key attributes in C is the primary key for each of the two participating entity types. These attributes become the primary key of C. Any nonkey attributes that are associated with the M:N relationship are included with the relation C.

Respondant: Create a new relation C. Include as foreign key attributes in C the primary key for each of the wo participating entity types. These attributes become the primary key of C. Any nonkey attributes that are associated with the M:N relationship are included with the relation C.

Respondant: Create a new relation C. Include as foreign key attributes in C the primary key for each of the two participating entity types. These attributes become the primary key of C. Any nonkey attributes that are associated with the M:N relationship are included with relation C.

Respondant: For each M:N relationship, we create a new relation C. Then, we include the foreign key attributes in C the primary key for each of the two participating entity types.

Respondant: Create 3 relations. Have the primary key of one entity be a joint primary key of the newly created relation. In addition, have the primary key of the other entity be the other joint primary key of the newly created relation.

Respondant: First create a new relation C. Include as foreign key attributes in C the primary key for each of the 2 entity types.

Respondant: Create a new relationship with foreign keys for each of the original two relations.

Respondant: Create a new relation C, Include as foreign key attributes in C the primary key for each of the two partcipating entity types.

Respondant: Create a new relation C, include foreign keys as attributesin C the primary key for each of the two participating entity types. These attributes become the primary key C.

Respondant: create a relation with the the primary key of each entity as foreign keys. the attributes of the entity are the primary key of the relation.

Respondant: 1. Create a new relation i.e: C
2. Include as foreign key attribute in C the primary key for each of two participating entities type.

Respondant: Relations are made for both entities and a new relation is made that includes the other relations primary keys as foreign keys. Then these attributes become the primary keys of the new relation and any nonkey attributes associated with the relationship are included in the new relation.

Respondant: Suppose that there is a binary many to many relationship between two entity types A and B. For such a relationship we create a new relation C. Include as foreign key atrributes in C the primary key for each of the two participating entity types. These attributes become the primary key of C. Any nonkey attributes that are associated with the M:N relationship are included with the relation C.

Respondant: Fr binary many to many relationship between two entity types A and B, we create a new relation C. Forign key attributes in C becomes the primary key of C. Nonkey attributes associated with the relationship are included in relation with C.

Respondant: A new relation is created. After that, we include foreign key atrributes in the new relation as the primary key for each of the two participating entity types. These attributes become the primary key of C. Any nonekey attributes taht are associated with the M:N relationship are included with the relation C.

Respondant: We create a new relation, and define the primary keys of each sides of the relation as foreign keys for the new relation.

Respondant: To map this many to many relationship, a new relation needs to be created. In this new relation, the primary key from each of the two participating entity types will be included in this new relation as foreign keys. These attributes become the primary key of the new relation. Any other non-key attributes will also be included in this new relation.

Respondant: A new relation is created. The foreign key attributes are the primary keys for each of the two entity types. The attributes become the primary key of the new relation.

Respondant: For a many to many type of relationship a new relation (entity) is created. The attributes contained in this new relation are the foreign keys that "point to" the respective primary keys.

Question #4: Associative Entities

Respondant: Involves essentially the same steps as a M:N relationship above. The first step is to create three relations: one for each of the two participating entity types, and the third for the associative entity. The second step then depends on whether the ER diagram had an identifier assigned to the associative entity. IF not assigned, the default primary key for the associative relation consists of the two primary key attributes from the other 2 relations. If assigned, a new (associative) relation is created to represent the associative entity. However, the primary key for this relation is the edentifier assigned on the ER diagram. The primary keys for the two participating entity types are then included as foreign keys in the associative relation.

Respondant: Three relations are created. One for each of the two participating entity types and the third for the associative entity. If an identifier is not assigned, the primary key of the associative relation consists of the primary key of the other two relations (foreign keys that reference the other two relations). If an identifier is assigned a new relations is still created for the associate entity. The identifier of this associative entity is represented along withe the primary keys of the two participating entity types (foreign keys)

Respondant: a. If an identifier was not assigned, the default primary key for the associative relation consists of the two primary key attributes from the other two relations. These attributes are then foreign keys that reference the other relations.
b. If an identifier was assigned,a new relation is created to represent the associative entity, but the primary key for this relation is the identifier assigned on the E-R diagram. The primary keys for the two participating entity types are then included as foreign keys in the associative relation.

Respondant: Mapping the associative entity involves essentially the same steps as mapping an M:N relationship. The first step is to create three relations: one for each of the two participating entity types, and the third for the associative entity. The second step then depends on whether on the E-R diagram an identifier was assigned to the associative entity.
If an identifier was not assigned, the default primary key for the associative relation consists of the two primary key attributes from the other two relations. These attributes are then foreign keys that reference the other two relations. If an identifier was assigned, the primary key for this relation is the identifier assigned on the E-R diagram. The primary keys for the two participating entity types are then included as foreign keys in the associative relation.

Respondant: Relation formed from the associative entity as the associative relation.
If the identifier was not assigned, the default primery key for the associative relation consists 2 promaryattributes fron the other 2 relationss . Thes attributes are then forein keys that reference the other 2 relationss. If the identifier was assigned in associative entity, a new relationship is created to represent the associative entity. The primary key for thiis relation is the identifier assigned on the E-R diagram. The primary key for the two participating entity types are then in cluded as foreign keys in the associative relation.

Respondant: The steps are basically sames as in the M:N relation.

Respondant: This is modeled almost exactly like a many to many relationship. First two relations are created for each of the entities involved in the relationship and then a third is created for the associative entity. Then if the associative entity does NOT have an identifier the primary keys from the initial two relations become the primary keys of the associative relation. If an identifier IS present in the associative entity then it becomes the primary key of the relation and the primary keys from the initial two are add as foreign keys.

Respondant: When mapping an associtive entity, 3 relations are created:
- one for each entity participating in the relationship
-one for the associative entity
If the associtivr entity has already an identifier, we will use it to define the primary key of the third relation. The 2 identifiers from the participating entity become foreign keys in this third relation.If the associative entity does no thave any identifier, then the default primary key for the third relation (the associative relation) consists of the 2 primary keys from the first 2 relations.

Respondant: Associative entity types involves essentially the same steps as mapping an M:N relationship. The first step is to create three relations: one for each of the two participating entity types, and the third for the associative entity. The relation formed from the associative entity is referred to as the associative relation. The second step then depends on whether on the E-R diagram an identifier was assigned to the associative entity. If an identifier was not assigned, the default primary key for the associative relation consists of the two primary key attributes from the other two relations. These are then the foreign keys that reference the other two relations. If the identifier is assigned, the primary key for this relation is the identifier assigned on the E-R diagram. The primary keys for the two participating entity types are then included as foreign keys in the associative relation.

Respondant: The first step is to create three relations: one for each of the two participating entity types, and the third for the associative entity. The relation formed from the associative entity is known as the associative relation. The second step then depends on whether on the E-R diagram an identifier was assigned to the associative entity. If an identifier was not assigned, the default primary key for the associative relation consists of the two primary key attributes from the other two relations. These attributes are then foreign keys that reference the other two relations. If an identifier was assigned to the associative entity type on the E-R diagram, then as before, a new (associative) relation is created to represent the associative entity. However, the primary key for this relation is the identifier assigned on the E-R diagram (rather than the default key). The primary keys for the two participating entity types are then included as foreign keys in the asssociative relation.

Respondant: The first step is to create three relations: one for each of the two participating entity types, and the third for the associative entit. We refer to the relation formed from the associative entity as the associative relation. The second step then depends on whether on the ER diagram an identifier was assigned to the associative entity.

Respondant: Create three relations: one for each of the two participating entity types, and the third for the associative entity. The second step depends whether on the E-R diagram an identifier was assigned to the associative entity.

Respondant: For associative entities, we first create three relations: one for each of the two participating entity types, and the third for the associative entity. Then, we decide if an identifier was assigned to the associative entity.

Respondant: Requires the same steps as a M:N relationship.

Respondant:

Respondant: Create three relations, one for each of two participating entity types and a third for the associative entity.

Respondant: First create three relations: one for each of the two participating entity types, and the third for the associative entity. Then determine whether the identifier was assigned to the associative entity.

Respondant: First create three relations: one for each of the two entities types, and the third for the associative entity. This is called an associative relation. The second step then depends on whether on the E-R diagram an identifier was assigned to the associative entity. If an identifier was not assigned the default primary key for the associative relationship consists of two primary key attributes from the other two relations. These attributes are then foreign keys that reference the other two relations. An identifier may be assigned for one of the two following reasons: 1.The associative entity type has a natural identifier that is familiar to the end user. 2.The default identifier may not uniquely identify instances of the associative entity.

Respondant: create a relation for each of the participating entities and the asscoiative entity. if the e-r diagram identifier was not assigned then the primary key for the asscoiative relation is the two primary key attributes of the other two relations. these are the foreign keys that regerence the other two relations. if the identifier is assigned, the primary key for this relation is the identifier assigned on the E-R diagram. the primary keys for the two participating entity types are then included as foreign keys in the associative relation.

Respondant: 1. Create three relations:one for each of two participating entity types. and the third for assosiative type.

2. Next step depends on whether on the E-R diagram an identifier was assigned to the assosiative entity.

Respondant: Associative entities are treated just like M:N relationships with 1 exception. If the associative entity has an identifier, that becomes the cooresponding relations primary key, and the other relations primary keys are foreign keys. If the associative entity doesn't have an identifier, then the other two primary keys become the 3rd relations primary keys as with M:N relations.

Respondant: Associative entities are formed from many to many relationships between other entity types. Associative entites are represented by a rectangle with a single line that encloses the diamond relationship model.

Respondant: To map associate entity, you nee to create three relations: one for each of the two participating entity types, and the third for associative entity. Then if identifier is not assigned, default primary key for associative relation consists of the two primary key from other two relations. If identifier is assigned, the a new relation is crated to represent the associative eneity, and the primary key is the identifier assigned on the ER diagram. The primary keys for the two participating eneity types becomes foreign keys.

Respondant: If an identifier was not assigned, the default primary key for the associative relation consists of the two primary key attributes fromt he other tow relations. These attributes are then foreign keys that reference the other two relations.
If identifier assigned, a new associative relation is created and the primary key for this relation is the identifier assigned on the E-R diagram. The primary keys for the two participating entity types are then included as foreign keys in the associative relation.

Respondant: We create three relations, one for each of the entities, one for the associative. if no identifier is assigned, then the primary key for the associative entity becomes the combination of the primary keys of the other two.

Respondant: Three relations need to be created for this many to many mapping: one for each of the two participating entities, and other associative entity. The next step depends of whether there is an identifier in the associative entity. If there isn't, then simply use those primary keys from each of the two entity types to become the foreign keys in the associative relation. Also, any non-key attributes would also be added. If there is an identifier, than that identifier would be used as the foreign key, in addition from the other two entity types. Again, the other non-key attributes are also added.

Respondant: Create three relations: 1 for each entity and a third for the associative entity. The latter relation is called the associative relation.

Respondant: Create three relations: one for each of the two participating entity type, and the third for the associative entity.

Question #5: Multi-valued attribute

Respondant: When the regular entity type contains a multivalued attribute, two new relations are created. The first relation contains all of the attributes of the entity type except the mulitvalued attrubute. The second relation contains two attributes that form the primary key of the second relation. The first of these attributes is the primary key function from the first relation, which becoms a foreign key in the second relation The second is the multivalued attribute.

Respondant: If an entity has a multivalued attribute, two new relations are created. The first relation contains all the attributes of the entity type except the multivalued attribute. The second relation contains two attributes (The primary key of the first relation, and the multivalues attribute). The name of the second relation should reflect the mulivalued attribute

Respondant: creat two relations. the first contains all of the attributes of the entity type except the multivalued attribute. the second contains two attributes that form the primary key of the second relation. the first of these attributes is the primary key from the first relation, which becomes a foreign key in the second relation. the second is the multivalued attribute. the name of the second relation should capture the meaning of the multivalued attribute.

Respondant: When the regular entity type contains a multivalued attribute, two new relations (rather than one) are created. The first relation contains all of the attributes of the entity type except the multivalued attribute. The second relation contains two attributes that form the primary key of the second relation. The first of these attributes is the primary key from the first relation, which becomes a foreign key in the second relation. The second is the multivalued attribute. The name of the second relation should capture the meaning of the multivalued.

Respondant: When the regular entity type contains a multivalued attribute, rwo new relationa rae created. the first relation contains all of the atributes of entity typse except the multivalued attribute. The second is the multivvalued atrribute. The name of the second relation should capture the meaning of the multivalued atrribute.

Respondant: In the case of multivalued attribute, two relations are created. Firs is a regular entity relation and the second one is formed from the primary key of the regular relation and from the multivalue attribute. The multivalued attribute is the primary key for this relation and the primary key from the regular relation is foreing key in this relation.

Respondant: In the case of any Multivalued Attributes, a relation is made specifically to address this attribute. The entity original identifier of the relation and the multi-valued attribute become the relations primary keys.

Respondant: If there is a multivalued attribute,first we have to create a relation for the entity described by this attribute. Then, we create a second relation including this multivalued attribute and the primary key of the first relation described above. This primary key becomes a foreign key in the second relation referencing to the key of the first relation.
These two attributes form the primary key of the second relation.

Respondant: When the regular entity type contains a multivalued attribute, two new relations are created. The first relation contains all of the attributes of the entity type except the multivalued attribute. The second relation contains two attributes that form the primary key of the second relation. The first of these attributes is the primary key from the first relation, which becomes a foreign key in the second relation. The second is the multivalued attribute. The name of the second relation should capture the meaning of the multivalued attribute.

Respondant: When the regular entity type contains a multivalued attribute, two new relations (rather than one) are created. The first relation contains all of the attibutes of the entity type except the multivalued attribute. The second relation contains two attributes that form the primary key of the first relation. The first of these attributes is the primary key from the first relation, which becomes a foreign key in the second relation. The second is the multivalued attribute. The name of the second relation should capture the meaning of the multivalued attribute. The relationship between foreign key and primary keys is indicated by an arrow.

Respondant: The first relation contains all of the attributes of the entity type except the multivalued attribute. The second relation contains two attributes that form the primary key of the second relation. The foreign key in the second relation. The second is the multivalued attribute.

Respondant: Two new relations are created for Multi-valued attribute. First relation contains all of the attributes of the entity type except the multivalued attribute. Second relation contains two attributes that form the primary key of the first relation. The first of these attributes is the primary key from the first relation, which becomes a foreign key in the second relation.

Respondant: Two new relations are created for multivalued attributes. The first relation contains all of the attributes of the entity type except the multivalued attribute. The second relation contains two attributes that form the primary key of the second relation. The first of these attributes is the primary key from the first relation, which becomes a foreign key in second relation. The second is the multivalued attribute.

Respondant: Two new relations are created. The first contains all relations of the entity type except multivalued ones. The second relation contains two attributes that form the primary key of the second relation. The relation of the multivalued attribute contains no nonkey attributes.

Respondant: The first relation contains all of the attributes of the attributes of the entity type except the multivalue attribute. The second relation contains 2 attributes that fom the primary key of the second relation. The name of the second relation should capture the meaning of the multivalued attribute.

Respondant: two new relations are created.

Respondant: Two relations are created. First relation contains all the attributes of the entity type except the multivalued attribute. Second relation contains two attributes that form the primary key of the second relation.

Respondant: When the regular entity type contains a multivalued attribute, two new relations (rather than one) are created. The first relationship contains all of the attributes of the entity type except the multivalued attribute. The second relation contains two attributes that form the primary key from the first relation, which becomes the foreign key in the second relations. The second is the multivalued attribute.

Respondant: When the regular entity type contains a multivalued attribute, two new relations are created. The first relation contains all of the attributes of the entity type except the multivalued attribute. The second relation contains two attributes that form the primary key of the second relation. The first of these attributes is the primary key from the first relation, which becomes a foreign key in the second relation. The second is the multivalued attribute. The name of the second relation should capture the meaning of the multivalued attribute.

Respondant: 1. Create a new realtion when the regular entity type contains a multivalue attribute.
2. The fisrt realtion contains of all the attibutes except the one that has multivalue attibute.
3. The second relation contains two attributes that form the primary key of the second relation.

Respondant: Two relations are made for 1 entity. One relation has all of the attributes except the multivalued attribute. The 2nd relation includes the primary key of the previous relation and the multivalued attribute, which also becomes a primary key.

Respondant: When the regular entity type contains a multivalued attribute, two new relations are created. The first relation contains all of the attributes of the entity type except the multivalued attribute. The second relation contains two attributes that form the primary key of the second relation, which becomes a foreign key in the second relation. The second is the multivalued attribute. The name of the second relation should capture the meaning of the multivalued attribute.

Respondant: Two new relations are created. First relation contains attributes of the entity type excpet the multivalued attribute. Teh second relation contains two attributes that forms the primary key of the first relation. The scond is the multivalued attibute.

Respondant: Two new relations are created. The first relation contains all of the attributes of the entity type except the multivalued attribute. The second relation contains two attributes that form the primary key of the second relation. The first of these attributes is the primary key from the first relation, which becomes a foreign key in the second relation. The second is the multivalued attribute. The name of the second relaiton should capture the meaning of the multivalued attribute.

Respondant: Rather than one relation, we create two relations, first one containing the attributes other than the multi- ones, the other with two attributes. First of these two is the primary key becoming the foreign key, and the second being the multi-attribute.

Respondant: When mapping multi-valued attribute, two new relations are created. The first relation would contain all the attributes that are not multi-valued. The second relation would have one primary key from the first relation, and one attribute which is of multi-valued.

Respondant: If the regular entity type contains a multivalued attribute, two new relations are created. The first relation contains all of the attributes of the entity type except the multivalued attribute. The second contains 2 attributes that form the primary key of the second relation.

Respondant: Two new relations rather than one are created with a multivalued attribute. The first relation contains all of the attributes of the entity type except the multivalued attribute. The second relation contains two attributes that form the primary key of the second relation. The first of these attributes is the primary key from the first relation, which becomes a foreign key in the second relation.

Question #6: Weak entity

Respondant: For each weak entity type, create a new relation and include all of the simple attributes as attributes of this relation. Then include the primary key of hte owner relation as a foreign key attribute in this new relation. The primary key of the new relation is the combination of this primary key of the owner and the partial identifier of the weak entity type.

Respondant: An relation for the owner entity is first created. For each weak entity a new relation is created including all of the simple attributes. The primary key of the owner relation becomes a foreign key of the new relation. The primary key of the new relation is the primary key of the owner and the partial identifier of the weak entity type.

Respondant: for each weak entity type, create a new relation and include all of the simple attributes as attributes of this relation. then include the primary key of the owner relation as a foreign key attribute in this new relation. the primary key of the new relation is the combination ot thies primary key of the owner and the partial identifier of the weak entity type.

Respondant: For each weak entity type, create a new relation and include all of the simple attributes (or simple components of composite attributes) as attributes of this relation. Then include the primary key of the owner relation as a foreign key attribute in this new relation. The primary key of the new relation is the combination of this primary key of the owner and the partial identifier of the weak entity type. (Partial identifier distinguishes the various occurrences of the weak entity for each owner entity instance.)

Respondant: For each weak entity type, create a new relation and include all of the simple attributes as the arributes of the relation. Then include the primery key of owner relation as a forien key attribute in this new relation. The primary key of the new relation is the combination of this primery key of the owner and the partial identifier of the weak entity type.

Respondant: First normal relation is created with all single attributes of the weak entity. Then a foreing key is included. The foreing key is the key from the 'parent' entity. The primary key of the weak entity relation is combination of the foreing key and the key of the weak entity.

Respondant: Assuming a relation has already been created for the owner entity. . .MAke a new relation for the weak entity. Then include all simple attributes in this relation along with the primary key from the owner relation. The primary key for the weak entity then becomes a combination of partial identifiers from the weak entity and the primary key from the owner relation.

Respondant: A relation is first of all createdfor the owner entity associated to the weak entity.
In addition, a second relation is created for the weak entity. All its attributes are included. The primary of the first relation is a foreign key is this second relation. The primary key of the second relation consists of this foreign key and the partial identifier of the weak entity type defined in the ER diagram.

Respondant: For each weak entity type, create a new relation and include all of the simple attributes (or simple components of composite attributes) as attributes of this relation. Then include the primary key of the owner relation as a foreign key attribute in this new relation. The primary key of the new relation is the combination of this primary key of the owner and the partial identifier of the weak entity type.

Respondant: For each weak entity type, create a new relation and include all of the simple attributes (or simple components of composite attributes) as attributes of this relation. Then include the primary key of the owner relation as a foreign key attribute in this new relation. The primary key of the new relation is the combination of this primary key of the owner and the partial identifier of the weak entity type. The foreign key relationship with its primary key is indicated by an arrow.

Respondant: For each weak entity type, create a new relation and include all of the simple attributes as attributes of this relation. Then include the primary key of the owner relation as a foreign key attribute in this newnew

relation. The primary key of the new relation is the combination of this primary key of the owner and the partial identifier of the weak entity type.

Respondant: Because weak entity type can not have an independent existence, a relation corresponding to the owner entity type must be created. For each weak entity type, a new relation and all of the simple attributes are created. Then include the primary key of the owner relation as a foreign key attribute in this new relation. The primary key of the new relation is the combination of this primary key of the owner and the partial identifier of the weak entity type.

Respondant: First, we create a relation like the regular entity. Then, for each weak entity, we create a new relation and include all of the simple attribute as attribute of this relation. Next, we include the primary key of the owner relation as a foreign key attribute in this new relation. The primary key of the new relation is the combination of this primary key of the owner and the partial identifier of the weaker entity type.

Respondant: The primary key of the owner relation is placed into the foreign key attribute and is then linked.

Respondant: Create a new relation and include all of the simple attributes as attributes of relation.

Respondant: Has an attribute and partial identifier.

Respondant: For each weak entity, create a new relation and include all of the simple attributes as attributes of this relation. Include the primary key of the owner relation as a foreign key attribute in this new relation.

Respondant: A weak entity type does not have a complete identifier, but must have an attribute called a partial identifier that permits distinguishing the various occurrences of the weak entity for each owner entity instances. First create a relation corresponding to the owner entity type. For each weak entity type, create a new relation and include all of the simple attributes as attributes of this relation. Then include the primary key of the owner relation as a foreign key attribute in this new relation. The primary key of the new relation is the combination of this primary key of the owner and the partial identifier of the weak entity type.

Respondant: create a relation with all the simple attributes for each weak entity type. the primary key of the owner relation is a foreign key attribute of the new relation. the primary key of the new relation is the combination of the primary key of the owner and the partial identifier of the weak entity

Respondant: 1. Create relation for entity type.
2. For each entity type, create a new relation and include all of the simple attributes as attibute for this relation.

Respondant: Each weak entity becomes a relation including all the simple attributes as attributes of the relation. The primary key of the owner relation becomes a foreign primary key to the weak entities relation. The other primary keys for this relation will be the entities partial identifiers.

Respondant: Weak entites are entities that cannot exist except with an identifying relationship with an owner entity type. Weak entities are identified by a rectangle with a double line.

Respondant: Weak entity type exists only through an identifying relationship with another "owner" entity type. For each weak entity type, create a new relation and include all of the simple attributs as atributes of relation. Then included the primary key of the owner relation as a foreign key attribute. The primary key is the combination of primary key of the owner and the partial identifer of the weak entity type.

Respondant: The new relation that is included all of the simple attributes is created. Then include the primary key of the owner relation as a foreign key attribute in this new relation. The primary key of the new relation is the combination of this primary key of the owner and the partial identifier of the weak entity type.

Respondant: We include all attributes from the entity, and include the primary key defining the entity as foreign key for the relation.

Respondant: Two relations are also created for this one. THe first relation is the one that contains the attribute from the entity type, while the second relation would list all the attributes of the ewak entity type, plus, the primary key of the first entity type.

Respondant: For each weak entity a new relation is created and all the simple attributes are included as attributes of this relation. The primary key of the "parent" relation is the foreign key attribute in the "dependent" relation. The primary key in the dependent relation is the combination of the partial identifier and the foreign key.

Respondant: For each weak entity type, a new relation is created with all its attributes included as attributes of this relation. The the primary key of the owner relation as a foreign key attribute in this new relation.

Question #7: Composite attribute

Respondant: When a regular entity type has a composite attribute, only the simpel component attributes of the composite attribute are included in the new relation.

Respondant: Only the simple components of the composite attribute are included in the new relation.

Respondant: only the simple component attributes of the composite attribute are included in the new relation, the other procedures are the same as the regular entities mapping.

Respondant: When a regular entity type has a composite attribute, only the simple component attributes of the composite attribute are included in the new relation.

Respondant: When the regular entity has a composite atribute, only the simple componenet attributes of composite attribute are included in the new relation.

Respondant: Only single attributes of the composite attribute are included in the new relation.

Respondant: In the case of Composite Attributes, only the simple ones end up as part of the relation. The entity contains a composite attribute. So the corresponding relation will contain the only the simple attributes that make up the composite attribute. It will also include any other simple attributes of course.

Respondant: If there is a composite attribute in the ER diagram, only one relation is created. This relation 'represents' the entity described by this composite attribute. (creation as explained in question 1). The simple components of the composite attribute are included as attributes of this relation.

Respondant: When a regular entity type has a composite attribute, only the simple component attributes of the composite attribute are included in the new relation.

Respondant: When a regular entity type has a composite attribute, only the simple component attributes of the composite attribute are included in the new relation.

Respondant: The simple component attributes of the composite attribute are included as attributes of the entity.

Respondant: Only the simple component attributes of the composite attribute are included in the new relation.

Respondant: We include only the simple component attributes of the composite attribute in the new relation.

Respondant: Only the simple components of the composite attribute are added to the relation.

Respondant: The simple component attributes are included in the new relation.

Respondant: only the simple component attributes of the composite attribute are included.

Respondant: When a regular entity type has a composite attribute, only the simple component attributes of the composite attribute are included in the new relation.

Respondant: A regular entity type has a composite attribute type when only the simple component attributes of the composite attribute are included in the new relation.

Respondant: When a regular entity type has a composite attribute, only the simple component attributes of the composite attribute are included in the new relation.

Respondant: First, each regular entity type in an ER diagram trasnformed into realtion. The name of the relation is the same with the entity type. The indentifier of the intety type become the primary key of the corresponding relation.

Include only simple component attributes of this composite attribute.

Respondant: For an entity with a composite attribute, only the simple attributes of the composite attribute are included in the relation.

Respondant: When a regular entity type has a composit attribute, only the simple component attributes of the composite attribute are included in the new relation.

Respondant: Only the simple components attributes of the composite attrubute are inclued in the new relation. For example, address is mapped with customer reations that contains the address attribute

Respondant: Only the simple component attribues of the composite attribute are included in the new relation.

Respondant: Answered in question 1.

Respondant: With the composite attribute, there is only one relation which lists all of the simple attributes, including the primary key. However, the composite attribute itself will not be included in this relation.

Respondant: Only the simple component attributes are included in the new relation.

Respondant: Only the simple component attributes of the composite attribute are included in the new relation.