ER to Relational mapping algorithm

Step 1: Mapping of Regular Entity Types.
  a.) For each regular (strong) entity type E in the ER schema, create a relation R that includes all the simple attributes of E.
    a. Composite attributes should be broken up into their atomic pieces, and those can be added as attributes to your relation
  b.) Choose one of the key attributes of E as the primary key for R.
  c.) If the chosen key of E is composite, the set of simple attributes that form it will together form the primary key of R.

Step 2: Mapping of Weak Entity Types
  a.) For each weak entity type W in the ER schema with owner entity type E, create a relation R & include all simple attributes (or simple components of composite attributes) of W as attributes of R.
  b.) Also, include as foreign key attributes of R, the primary key attribute(s) of the relation(s) that correspond to the owner entity type(s) E.
  c.) The primary key of R is the combination of the primary key(s) of the owner(s) and the partial key of the weak entity type W, if any

Step 3: Mapping of Specializations and Generalizations
  a.) For each Superclass C with m Subclasses {S1 … Sm} add a relation for C according to the rules in step 1.
  b.) For each subclass Si add a relation for Si according to the rules in step one
  c.) For each subclass Si, add the primary key of C as a foreign key attribute. This will be the primary key for the subclass relation.
  d.) If a Subclass Si is a shared subclass, then add a foreign key attribute for each Superclass of Si. The combination of these foreign keys will be the primary key for Si

Step 4: Mapping of Union Types/Categories
  a.) For each Superclass Ci of the Union Type U, add a relation according to the rules in step 1.
  b.) Create an artificial key for U. Add a relation for the U with it's attributes, and the Artificial Key as the Primary key.
  c.) In each Superclass Ci add the primary key of U as a foreign key

Step 5: Mapping of Binary 1:1 Relation Types
  a.) For each binary 1:1 relationship type R in the ER schema, identify the relations S and T that correspond to the entity types participating in R.
  b.) Choose one of the relations-say S-and include a foreign key in S the primary key of T. It is better to choose an entity type with total participation in R in the role of S.

Step 6: Mapping of Binary 1:N Relationship Types.
   a.) For each regular binary 1:N relationship type R, identify the relation S that represent the participating entity type at the N-side of the relationship type.
   b.) Include as foreign key in S the primary key of the relation T that represents the other entity type participating in R.
   c.) Include any simple attributes of the 1:N relation type as attributes of S.

Step 7: Mapping of Binary M:N Relationship Types.
   a.) For each regular binary M:N relationship type R, create a new relation S to represent R. This is a relationship relation.
   b.) Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types; their combination will form the primary key of S.
   c.) Also include any simple attributes of the M:N relationship type (or simple components of composite attributes) as attributes of S.

Step 8: Mapping of Multivalued attributes.
   a.) For each multivalued attribute A, create a new relation R.
   b.) This relation R will include an attribute corresponding to A, plus the primary key attribute K-as a foreign key in R-of the relation that represents the entity type of relationship type that has A as an attribute.
   c.) The primary key of R is the combination of A and K. If the multivalued attribute is composite, we include its simple components.

Step 9: Mapping of N-ary Relationship Types.
   a.) For each n-ary relationship type R, where n>2, create a new relation S to represent R.
   b.) Include as foreign key attributes in S the primary keys of the relations that represent the participating entity types.
   c.) Also include any simple attributes of the n-ary relationship type (or simple components of composite attributes) as attributes of S.