ER-to-Relational Mapping

ER-to-Relational Mapping

- There is almost a one-to-one correspondence between the ER constructs and the relational ones.
- The two major distinctions are:
  I. In a relational schema, relationships are represented *implicitly* through primary and foreign keys of participating entities.
  II. In a relational schema, columns of relations *cannot* be multi-valued or composite.
    - Composite attributes are replaced with their simple component ones, and
    - multi-valued attributes are stored in a separate relation.

Mapping Algorithm

1. For each strong entity type E.
   - Create a new table.
   - Include as its columns, all the simple attributes and simple components of the composite attributes of E.
   - Identify the primary key and the alternate keys.

Mapping Algorithm…Weak Entities

2. For each weak entity W that is associated with only one 1:1 identifying owner relationship.
   - Identify the table T of the owner entity type.
   - Include as columns of T, all the simple attributes and simple components of the composite attributes of W.
3. For each weak entity W that is associated with a 1:N or M:N identifying relationship, or participates in more than one relationship.
   - Create a new table T.
   - Include as its columns, all the simple attributes and simple components of the composite attributes of W.
   - Form its primary key by including as a foreign key in R, the primary key of its owner entity.

Mapping Algorithm…Binary Relationships

4. For each binary 1:1 relationship type R.
   - Identify the tables S and T of the participating entity types.
   - Choose S (preferably the one with total participation).
   - Include as foreign key in S, the primary key of T.
   - Include as Columns of S, all the simple attributes and simple components of the composite attributes of R.
5. For each binary 1:N relationship type R.
   - Identify the table S (at the *N-side*) and T of the participating entities.
   - Include as a foreign key in S, the primary key of T.
   - Include as columns of S, all the simple attributes and simple components of composite attributes of R.

Mapping Algorithm…Any Relationship

6. For each N-ary relationship (including binary N:M relationship) type R.
   - Create a new table T.
   - Include as columns of T, all the simple attributes and simple components of composite attributes of R.
   - Include as foreign keys, the primary keys of the participating (strong or weak) entity types.
   - Specify as the primary key of T, the list of foreign keys.

Mapping Algorithm…Multi-valued Attributes

7. For each multi-valued attribute A.
   - Create a new table T.
   - Include as columns of T, the simple attribute or simple components of the attribute A.
   - Include as foreign key, the primary key of the entity or relationship type that has A.
   - Specify as the primary key of T, the foreign key and the columns corresponding to A.

EER Mapping Algorithm

8. For each specialization with disjoint subclasses.
   - Create a new table T_i for each subclass S_i.
   - Include as columns of T_i, the simple attributes and simple component attributes of the superclass.
   - Include as columns of T_i, the simple attributes and simple component attributes specific to S_i.
   - Identify the primary key.

EER Mapping Algorithm…

9. For each specialization with overlapping subclasses.
   - Create a new table O for the superclass.
   - Include as columns of O, the simple attributes and the simple component attributes of the superclass.
   - Identify its primary key and alternate keys.
   - Create a new table T_i for each subclass S_i.
   - Include as columns of T_i, the simple attributes and simple component attributes specific to S_i.
   - Include as a foreign key in T_i (to be part of the primary key of T_i), the primary key of O.