7. **ER to Relational Data Model Mapping**

One technique for database design is to first design a conceptual schema using a high-level data model, and then map it to a conceptual schema in the DBMS data model for the chosen DBMS.

- Here we see a way to do this mapping from the ER to the relational data model.

- We follow the 7 step process given in E/N 6.8 (or 12.1.1 in 1st Edition), augmented with a step for specialised entities.
Step 1: For each regular (not weak) entity type E, create a relation R with

Attributes: All simple attributes (and simple components of composite attributes) of E.

Key: Choose one of the keys of E as the primary key for the relation.
**Step 1a** : For each specialised entity type E, with parent entity type P, create a relation R with

- **Attributes** : The attributes of the key of P, plus the simple attributes of E.
- **Key** : The key of P.
**Step 2**: For each weak entity type \( W \), with owner entity type \( E \), create a relation \( R \) with

**Attributes**: All simple attributes (and simple components of composite attributes) of \( W \), and include as a foreign key the prime attributes of the relation derived from \( E \).

**Key**: The foreign key plus the partial key of \( W \).
Step 3: For each 1:1 relationship type B, let E and F be the participating entity types. Let S and T be the corresponding relations.

- Choose one of S and T (prefer one that participates totally), say S.
- Add the attributes of the primary key of T to S as a foreign key.
- Add the simple attributes (and simple components of composite attributes) of B as attributes of S.

(Alternative: merge the two entity types and the relationship into a single relation, especially if both participate totally and do not participate in other relationships).
Step 4: For each regular 1:N relationship type B. Let E and F be the participating entity types.

Let E by the entity type on the 1 side, F the one on the N side.

Let S and T be the corresponding relations.

Add the attributes of the primary key of S to T as a foreign key.

Add to T any simple attributes (or simple components of composite attributes) of the relationship.

(Notice that this doesn’t add any new tuples, just attributes.)
**Step 5** : For each N:M relationship type B. Create a new relation R. Let E and F be the participating entity types. Let S and T be the corresponding relations.

**Attributes** : The key of S and the key of T as foreign keys, plus the simple attributes (and simple components of composite attributes) of B.

**Key** : The key of S and the key of T.
Step 6: For each multivalued attribute A. Create a new relation R.
Let A be an attribute of E.

Attributes:
1. A (if A is a simple attribute) together with the key of E as a foreign key.
2. The simple components of A (if A is a composite attribute), together with the key of E as a foreign key.

Key: All attributes.
**Step 7** : For each n-ary relationship type \( n > 2 \). Create a new relation with

- **Attributes** : as for Step 5.
- **Key** : as for Step 5, except that if one of the participating entity types has participation ratio 1, its key can be used as a key for the new relation.
See E/N Fig 3.2, Fig 6.5 and Fig 6.19 (or Fig 12.1, Fig 12.2 and Fig 12.3 in 1st Edition) for examples.