ER-to-Relational Mapping

To convert an Entity-Relationship design to a relational database schema, a procedure which includes the following steps may be followed. (It is taken from Section 7.1 of the textbook; 9.1 in the Third Edition.)

Entity Conversion

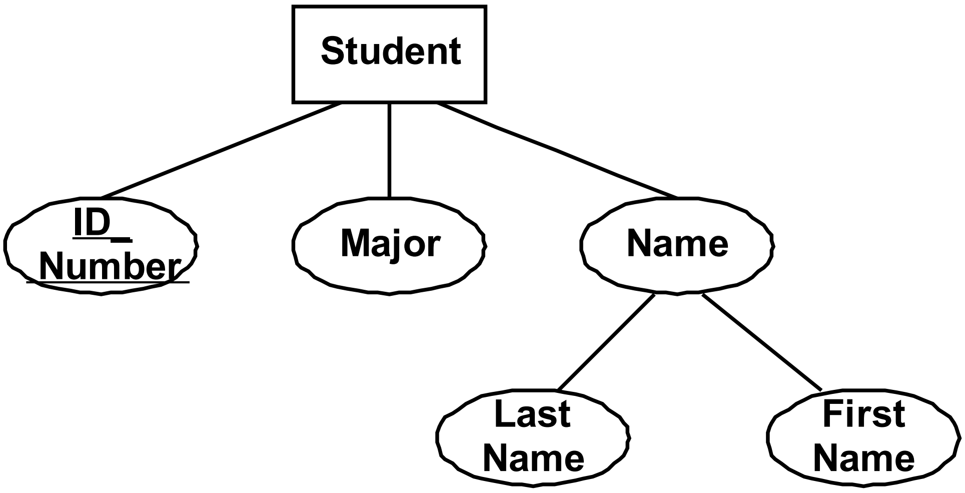
  - Regular Entity Conversion
  - Weak Entity Conversion

- Relationship Conversion
  - Binary Relationship Conversion
    - 1:1 Relationship Conversion
    - 1:Many Relationship Conversion
    - Many:Many Binary Relationship Conversion
  - Non-Binary Relationship Conversion

- Attribute Conversion
  - Multivalued Attribute Conversion

Regular Entity Conversion

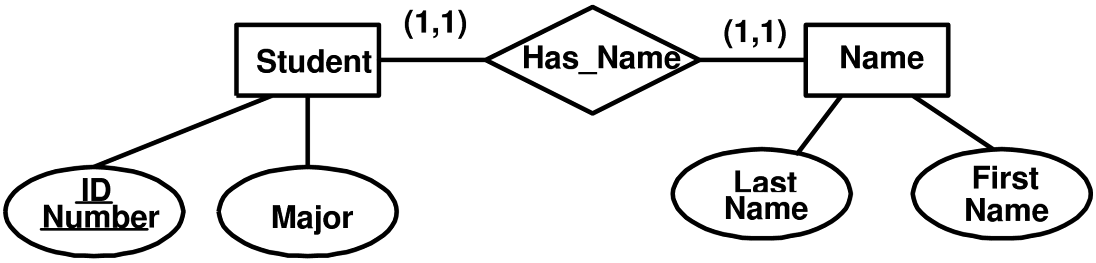
- A regular entity is implemented as a relation which contains as its attributes all of the attributes of the entity.
- Any key attribute (or set of attributes) may be used as a candidate key in the relation.
- All non-atomic attributes are lost. If it is truly desired to retain non-atomic attributes, then alter the ER model so that the compound attribute is an entity. Consider the student example:



The corresponding relational model would be:

Student			
ID_number	Major	Last Name	First Name

- To maintain the existence of the compound attribute Name, re-design the ER schema as follows:



- Then apply the techniques described for binary relationships.

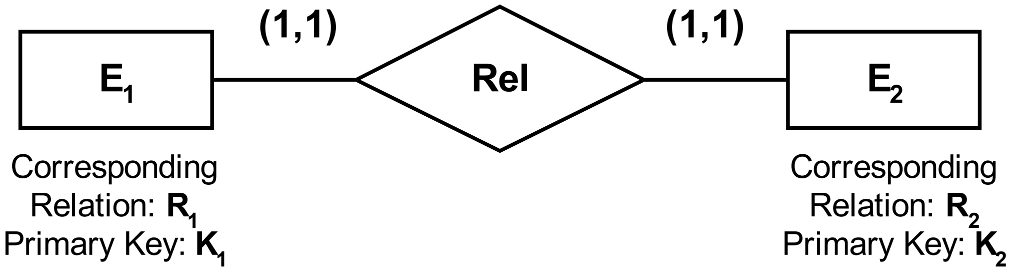
Weak Entity Conversion

The technique is similar to that for a regular entity conversion, except that the primary key of the owner entity must be included, as a foreign key, in the relation for the weak entity.

Relationship Conversion

- For relationship conversion, it is assumed that the entities participating in the relationship have already been converted according to the rules identified above.

One-to-One Binary Relationship Conversion:


(1,1)
(1,1)
E_1
Rel
E_2
Corresponding Relation: $\mathbf{R_1}$
Primary Key: $\mathbf{K_1}$
Corresponding Relation: $\mathbf{R_2}$
Primary Key: $\mathbf{K_2}$

- Choose one of the relations (say R_1) as the controller.

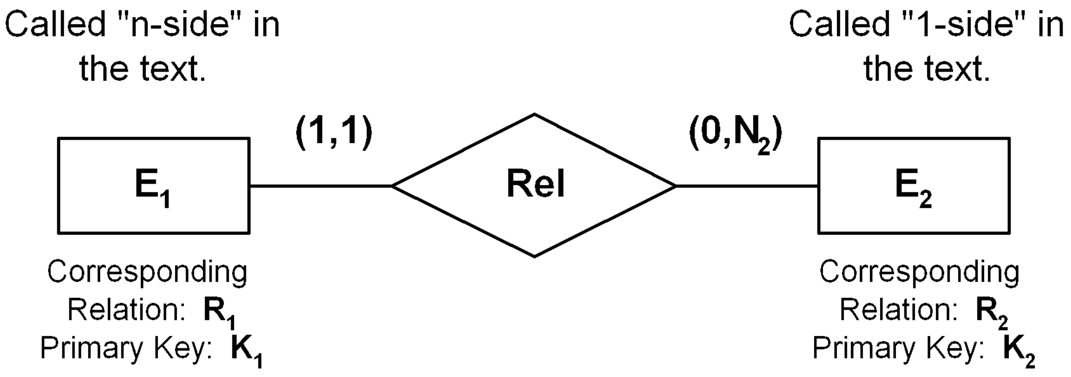
- Add the primary key of the controller to the other relation (R_2 in this case) as a foreign key.

- Also add any attributes associated with Rel to R_1.

- Alternative: Merge the two relations into one. (Only appropriate if the individual relations are not needed in other constructions.)

- If one side is (0,1) and the other (1,1), it is preferable to insert the key from the (0,1) side into the relation of the (1,1) side. This avoids unnecesary use of null values.

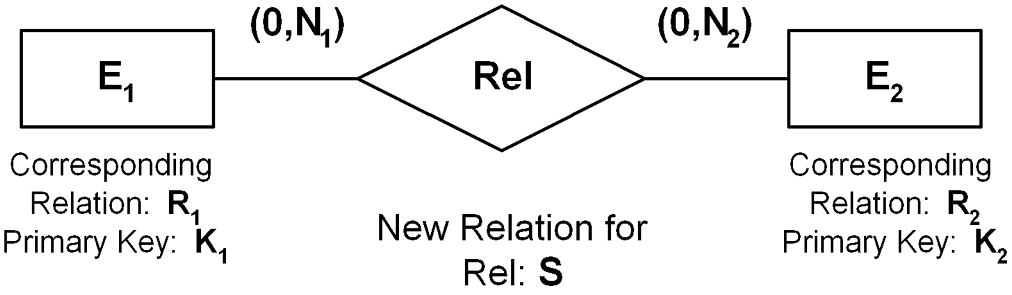
One-to-Many Binary Relationship Conversion:


Called "n-side" in the text.
Called "1-side" in the text.
(1,1)
(0,N2)
E1
Rel
E2
Corresponding Relation: R1
Primary Key: K1
Corresponding Relation: R2
Primary Key: K2

- Insert the primary key K_2 of the “1-side” entity as a foreign key into the relation R_1 of the “n-side” entity.

- Also add any attributes associated with Rel to R_1.

- Many-to-Many Binary Relationship Conversion:



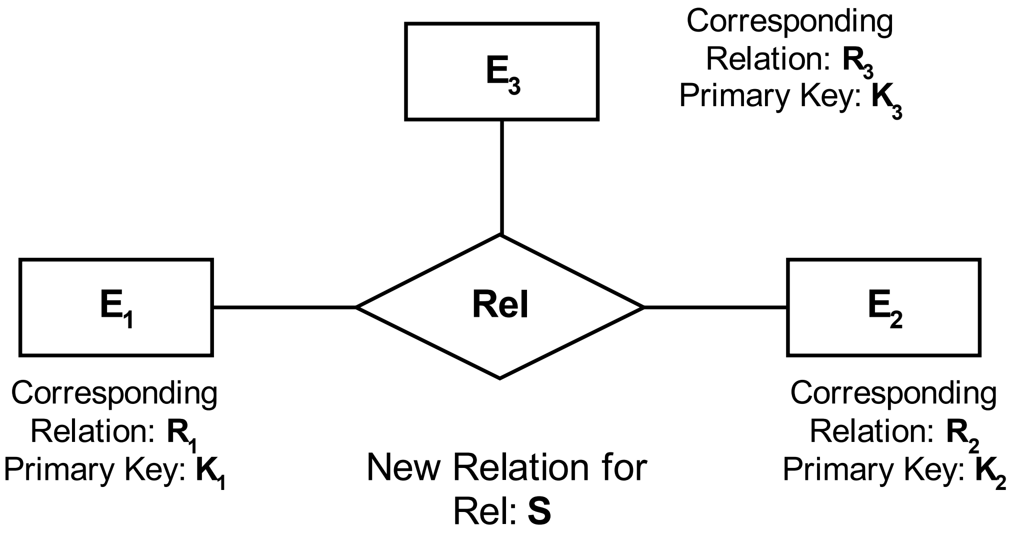
Corresponding Relation: **R$_1$** Primary Key: **K$_1$**

New Relation for Rel: **S**

Corresponding Relation: **R$_2$** Primary Key: **K$_2$**

- The new relation S has, as foreign keys, the primary key of each of E$_1$ and E$_2$.
- Include primary key pairs which “match” in the relationship Rel.
- Also include any attributes directly associated with Rel.

Non-Binary Relationship Conversion:



E_3

Corresponding Relation: $\mathbf{R_3}$
Primary Key: $\mathbf{K_3}$

E_1

Rel

E_2

Corresponding Relation: $\mathbf{R_1}$
Primary Key: $\mathbf{K_1}$

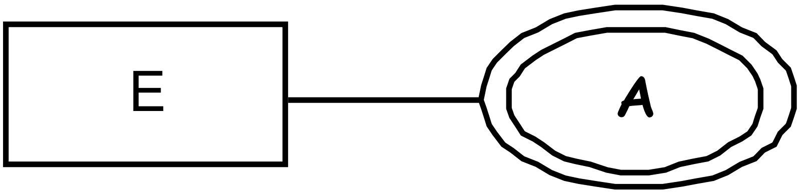
New Relation for Rel: **S**

Corresponding Relation: $\mathbf{R_2}$
Primary Key: $\mathbf{K_2}$

- Create a new relation S which contains as attributes each of the primary keys of the participating entities. These will be foreign keys.

- Also include any attributes directly associated with Rel.

- Multivalue Attribute Conversion:



Corresponding
Relation: **R**

- Create a new relation R_A which contains the attribute *A*, in addition to the primary key of R.

- There is one tuple for each
(Primary Key, Attribute value)
pair.