EAR Model → Relational Model

1 Entity Definition

This is rather a straightforward step to move the Entity Definition to Relational Model presentation

If there's any [m:n] relationship in EAR, the breakdown is [1:m and n:1] with an Intermediate Entity. Thus, a corresponding Relation is included in basic model

```
Model AAA
Domains
.....
Relation1 (Entity1Name)
  Entity1_ID
  Attribute1
  Attribute2
  .....
  Primary Key (Entity1_ID)
Relation2 (Entity2Name)
  .....
```

2 Degree of Relationship represented in ER Diagram

1 : m	Add shared attributes to the m: side Relation Define it as Foreign key reference to the 1: side relation
1 : 1	If it is 1:1 relationship, also in the work-like m: side Relation, define the attribute as Alternate Key
m : n	Treat it as 2 x "1 : m" relationship

```
Relation1 (Entity1Name)
  Entity1_ID
  Attribute1
  Attribute2
  .....
  Primary Key (Entity1_ID)
Relation2 (Entity2Name)
  .....
  RefEntity1_ID (shared attribute introduced)
  Foreign Key RefEntity1_ID reference Relation1 (=> 1:m)
  Alternate Key Entity1_ID (=> 1:1)
```

3 Participation Condition of Relationship

m : side

1 : side

● Foreign Key .... **not allowed null**

○ Foreign Key .... **allowed null**

● (Project Relation1 over ReferencedAttribute) DIFFERENCE (Project Relation2 over ReferencingAttribute) is **EMPTY**

○ (Project Relation1 over ReferencedAttribute) DIFFERENCE (Project Relation2 over ReferencingAttribute) is **NOT EMPTY**

```
Relation1 (Entity1Name)
  Entity1_ID
  Attribute1
  Attribute2
  .....
  Primary Key (Entity1_ID)
  (Project Relation1 over Entity1_ID) difference
  (Project Relation2 over RefEntity1_ID) is/is not EMPTY
Relation2 (Entity2Name)
  .....
  RefEntity1_ID
  Foreign Key RefEntity1_ID reference Relation1 allowed/not allowed NULL
  Alternate Key Entity1_ID
```

3 Other Constraints

Use Relational Algebra to implement varies Constraints
e.g. inclusivity and exclusivity
:n entity is exclusively; :n entity is inclusively;
1: entity is inclusively; 1: entity is exclusively

e.g. The following constraint may be defined in the Patient relation, to restrict that any Doctor ocurrence related to a Patient occurrence via the IsResponsibleFor relationship must have a Position attribute with the value Consultant:

```
Constraint (join (select Doctor where Position <> 'Consultant') and Pati
where StaffNo = StaffNo) is empty
```