

# **MAPPING ENTITY RELATIONSHIP SCHEMAS INTO THE RELATIONAL MODEL**

## **Steps of the Algorithm (Chapter 9 – pages 290-295)**

- **STEP 1: Map Entity Types**
- **STEP 2: Map Weak Entity Types- draw identifier from parent entity type into weak entity type**
- **Map Relationship Types (STEPS 3 to 5):**
  - 1:1 options for setting up one, two or three relations**
  - 1:N the many side provides a key to the one side**
  - M:N need to set up a separate relation for the relationship**
- **STEP 6: Map multivalued attributes – set up a new relation**
- **Step 7: Map higher order relationships (ternary, 4- way etc.) – they become separate relations.**
- **STEP 8: Mapping of generalization hierarchies and set-subset relationships – possibility of collapsing into one relation vs. as many as the distinct classes.**

# MAPPING Extended Entity Relationship (EER) SCHEMAS INTO THE RELATIONAL MODEL

## (Section 9.2, Pages 295-299)

Attrs ( R ) = attributes of relation R

PK ( R ) = Primary Key of relation R

Assume there is a specialization of class C into subclasses  $S_1, S_2, S_3, \dots, S_m$ .  
That is, there are m subclasses of a superclass C.

STEP 8 in the mapping mentions the following options:

Option 8A : (1) One relation for C with all attributes and primary key.  
(2) m additional relations : One relation per subclass  $S_i$  with its own attributes Attrs ( $S_i$ ) plus the PK ( C ).  
The primary key of each subclass relation is the same as PK ( C ).

Option 8B : NO Relation corresponding to the superclass.  
M subclass relations, one per subclass. Each relation for subclass  $S_i$  includes PK ( C ), Attrs ( C ) and the Attrs ( $S_i$ ).  
The primary key of each subclass relation is the same as PK ( C ).

Option 8C : Create a single FAT relation containing all attributes from C, the key PK ( C), the union of all attributes from all subclasses, and one type attribute t that determines to which (one or more) subclasses an entity belongs. This works well for disjoint subclasses. However, the type attribute can be so designed that it can represent all possible valid combinations of subclasses that might exist.

Option 8D: Create a single FAT relation containing all attributes from C, the key PK ( C), the union of all attributes from all subclasses, and m boolean type attributes  $t_1, t_2, t_3, \dots, t_m$ .  
The attributes to which an entity belongs are set to one. This can work for Disjoint as well as overlapping subclasses.

## Mapping of a Category

The category may stand for entities coming from different types with different keys. A SEPARATE relation is set up for a category with its own SURROGATE KEY and includes the collection of its own attributes. It also contains a type attribute to denote the source entity type involved. Each of the relations for the source entities then includes the surrogate id as cross reference.

E.g., OWNER as a category defined over COMPANY and PERSON may be set up as a relation:  
OWNER (Owner\_id, Owner-type)

Then each relation PERSON and COMPANY has an added attribute Ownerid (with possible null values for those that are not owners).

PERSON (SS#, Driver\_license\_no, Name, Address, Owner\_id)  
COMPANY ( Company\_name, Address, Owner\_id)