DATABASE SYSTEMS ASSIGNMENT 4

Name-Vinayak Raghupathy

NYUID-N11568565

NETID-VR840

Q 1) Discuss the options for mapping EER model constructs to relations.

Ans 1)

Step \*

We discuss the mapping of EER model constructs to relations by extending the

ER-to-relational mapping algorithm-

**Options for Mapping Specialization or Generalization**.

Convert each specialization with m subclasses {S1, S2, ..., Sm} and (generalized) superclass C, where the attributes of C are {k, a1, ...an} and k is the (primary) key, into relation schemas using one of the following options:

**Option A: Multiple relations-Superclass and subclasses**

Create a relation L for C with attributes Attrs(L) = {k, a1, ..., an} and PK(L) = k. Create a relation Li for each subclass Si, 1 ≤ i ≤ m, with the attributes Attrs(Li) = {k} ∪ {attributes of Si} and PK(Li) = k. This option works for any specialization.

**Option B: Multiple relations-Subclass relations only**

Create a relation Li for each subclass Si, 1 ≤ i ≤ m, with the attributes Attrs(Li) = {attributes of Si} ∪ {k, a1, ..., an} and PK(Li) = k. This option only works for a specialization whose subclasses are total (every entity in the superclass must belong to (atleast) one of the subclasses). Additionally, it is only recommended if the specialization has the disjointedness constraint (see Section 8.3.1).If the specialization is overlapping, the same entity may be duplicated in several relations.

**Option C: Single relation with one type attribute**

Create a single relation L with attributes Attrs(L) = {k, a1, ..., an} ∪ {attributes of S1} ∪ ... ∪ {attributes of Sm} ∪ {t} and PK(L) = k. The attribute t is called a type (or discriminating) attribute whose value indicates the subclass to which each tuple belongs, if any. This option works only for a specialization whose subclasses are disjoint, and has the potential for generating many NULL values if many specific attributes exist in the subclasses.

**Option D: Single relation with multiple type attributes**

Create a single relation schema L with attributes Attrs(L) = {k, a1, ..., an} ∪ {attributes of S1} ∪ ... ∪ {attributes of Sm} ∪ {t1, t2, ..., tm} and PK(L) = k. Each ti, 1 ≤ i ≤ m, is a Boolean type attribute indicating whether a tuple belongs to subclass Si. This option is used for a specialization whose subclasses are overlapping (but will also work for a disjoint specialization)

**Mapping of Shared Subclasses (Multiple Inheritance)**

A shared subclass, such as ENGINEERING\_MANAGER is a subclass of several superclasses, indicating multiple inheritance. These classes must all have the same key attribute; otherwise, the shared subclass would be modeled as a category (union type). We can apply any of the options discussed in step \* to a shared subclass, subject to the restrictions discussed in step \* of the mapping algorithm.

**Mapping of Categories (Union Types)**

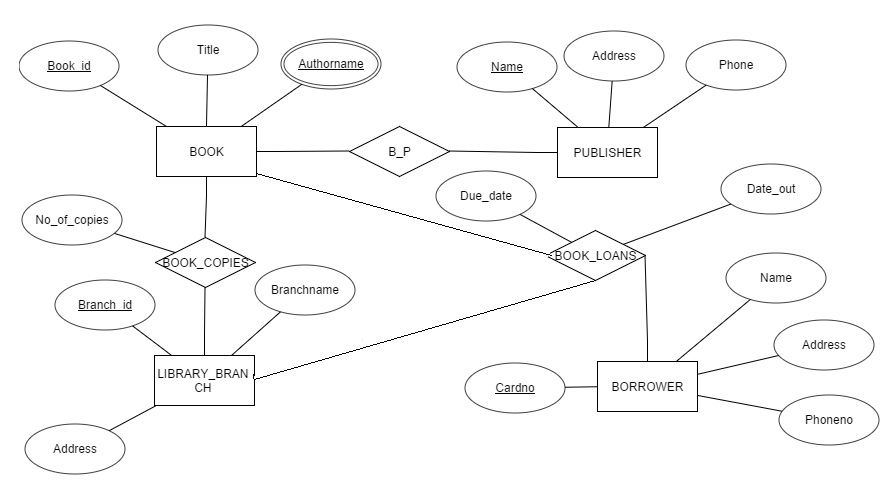
We add another step to the mapping procedure—step +—to handle categories. A category (or union type) is a subclass of the union of two or more superclasses that can have different keys because they can be of different entity types .An example is the OWNER category, which is a subset of the union of three entity types PERSON, BANK, and COMPANY. The other category REGISTERED\_VEHICLE, has two superclasses that have the same key attribute.

Step +

For mapping a category whose defining superclasses have different keys, it is customary to specify a new key attribute, called a **surrogate key**, when creating a relation to correspond to the category.The keys of the defining classes are different, so we cannot use any one of them exclusively to identify all entities in the category. In our example , we create a relation OWNER to correspond to the OWNER category and include any attributes of the category in this relation. The primary key of the OWNER relation is the surrogate key, which we called Owner\_id.We also include the surrogate key attribute Owner\_id as foreign key in each relation corresponding to a superclass of the category, to specify the correspondence in values between the surrogate key and the key of each superclass. Notice that if a particular PERSON (or BANK or COMPANY) entity is not a member of OWNER, it would have a NULL value for its Owner\_id attribute in its corresponding tuple in the PERSON (or BANK or COMPANY) relation, and it would not have a tuple in the OWNER relation. It is also recommended to add a type attribute to the OWNER relation to indicate the particular entity type to which each tuple belongs (PERSON or BANK or COMPANY).

Q2) Try to map the relational schema in Figure 6.14 into an ER schema. This is part of a process known as reverse engineering, where a conceptual schema is created for an existing implemented database. State any assumptions you make.

Ans 2)



Q 3)Map the EER diagrams in Figures 8.9 and 8.12 into relational schemas.Justify your choice of mapping options

Q 4) Is it possible to successfully map a binary M:N relationship type without requiring a new relation? Why or why not?

Q 5) Consider the EER diagram in Figure 9.9 for a car dealer.

Map the EER schema into a set of relations. For the VEHICLE to

CAR/TRUCK/SUV generalization, consider the four options presented in

Section 9.2.1 and show the relational schema design under each of those

options.

Q 6) Using the attributes you provided for the EER diagram in Exercise 8.27, map

the complete schema into a set of relations. Choose an appropriate option

out of 8A thru 8D from Section 9.2.1 in doing the mapping of generalizations

and defend your choice.