**Submitted by:**

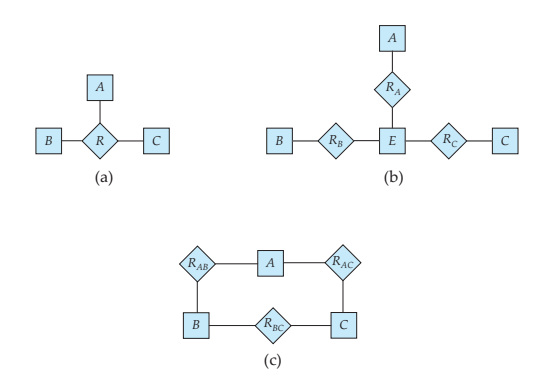
Divyaswor Makai Shrestha

CE(2nd Y/ 2nd Sem)

Roll-48

Assignment 3

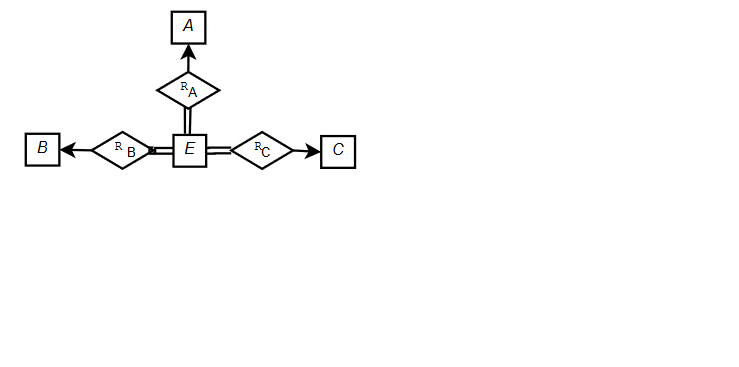
**7.6 Consider the representation of a ternary relationship using binary relationships as described in Section 7.7.3 and illustrated in Figure 7.27b (attributes not shown).**

****

**a. Show a simple instance of E, A, B,C, RA, RB, and RC that cannot correspond to any instance of A, B,C, and R.**

Let E = {e1, e2}, A= {a1, a2}, B = {b1}, C = {c1}, RA = {(e1, a1), (e2, a2)}, RB = {(e1,b1)}, and RC = {(e1, c1)}. We see that because of the tuple (e2, a2), no instance of R exists which corresponds to E, RA, RB and RC.

**b. Modify the E-R diagram of Figure 7.27b to introduce constraints that will guarantee that any instance of E, A, B,C, RA, RB, and RC that satisfies the constraints will correspond to an instance of A, B,C, and R.**

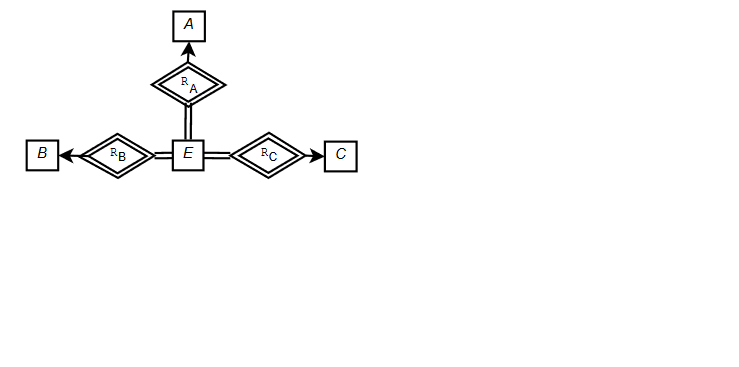
****

**c. Modify the translation above to handle total participation constraints on the ternary relationship.**

Suppose A totally participates in the relationship R, then introduce a totalparticipation constraint between A and RA

**d. The above representation requires that we create a primary-key attribute for E. Show how to treat E as a weak entity set so that a primary-key attribute is not required.**

Consider E as a weak entity set and RA, RB and RC as its identifying relationship sets. Thus the following figure will show E as a weak entity set.



**7.7 A weak entity set can always be made into a strong entity set by adding to its attributes the primary-key attributes of its identifying entity set. Outline what sort of redundancy will result if we do so.**

Weak entity sets are the entity sets with no primary keys. When we convert the weak entity sets to a strong entity set by adding to its attributes the primary key attributes of its identifying entity set the following redundancies are formed:

1. As the weak and strong entity sets are related to each other and can be used by same primary key, the addition of primary key to weak entity set will cause repetition of the same data causing redundant storage of the primary key.
2. In properly managed database systems, the weak entity sets are dependent on the strong entity sets and the weak sets are automatically deleted upon deleting the strong entity sets. So when the weak entity sets are added with primary key to change it to a strong entity set, the deletion of the data process will be hectic.
3. Converting weak entity sets to a strong entity sets can alter the logical relationship between entity sets.

**7.26 Design generalization-specialization hierarchy for a motor vehicle sales company. The company sells motorcycles, passenger cars, vans, buses. Justify your placement of attributes at each level of the hierarchy. Explain why they should not be placed at a higher or lower level.**

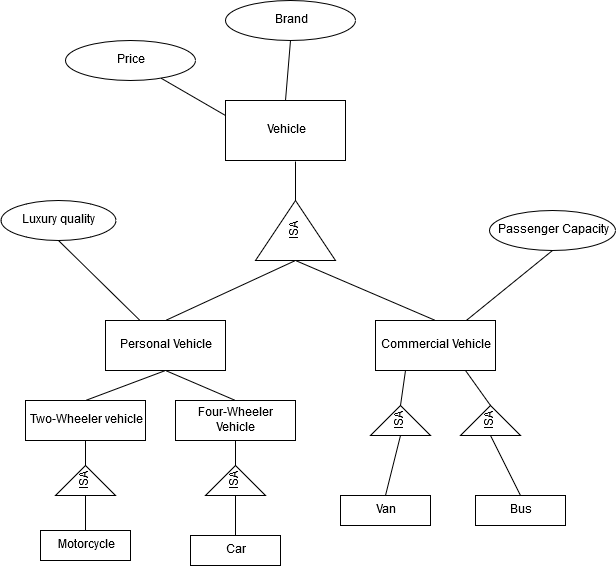


Fig: General-specialization hierarchy

The above figure shows a generalization-specialization hierarchy for a motor vehicle sales company. The vehicles it sells are motorcycles, cars, vans and buses. Vehicle is the higher-level entity set with attributes such as brand, price etc. as these attributes will be common to any of the vehicles. The next level of entity set has been taken as Personal and Commercial. Personal vehicles are compared based upon the luxury status of the vehicle while commercial vehicles are purchased on the basis of the passenger capacity. The attribute on each of these subclasses are distinct from each other. So the attributes placed in the superclass would mean unnecessary use of storage. Thus, attributes that are common to the subclasses should be included in the superclass whereas specific attributes to any subclass should to attribute to the subclass only.

**7.27 Explain the distinction between condition-defined and user-defined constraints. Which of these constraints can the system check automatically? Explain your answer.**

a. Condition-defined constraints:

In condition defined lower level entity sets, membership is evaluated on the basis of whether or not an entity set satisfies an explicit condition or predicate. When lower-level entity sets of a higher-level entity sets are to be created then constraints are applied to the entities of the lower-level sets so that they can be classified. For example, if people is the higher-level entity set which has attributes such as name, role. The attributes in roles can be used as condition to divide the entities to lower-level sets. The lower-level sets may be students, teachers, staff etc. Entities satisfying role as student will be included in the student (lower-level entity set).

b. User-defined constraints:

User-defined lower-level entity sets are not constrained by a membership condition but rather are assigned entities to a given entity set. The entities are assigned certain values by the database user so that they are classified into lower-level entity sets. Let us take an example of student as the higher-level entity set. The value of graduate is set to true only when the student has completed his course. Here lower-level entity sets may be graduate and undergraduate. When the student completes his course a database user sets the attribute of the student as true and thus the student is classified into the graduate entity set.

The system can check the condition-defined constraints automatically. The entity entered into the entity set can be classified into lower-level entity sets automatically as the system can check the predicate value entered with the data. Whereas the user-defined constraints aren’t necessarily fulfilled at times and system cannot easily classify the entities.

**7.28 Explain the distinction between disjoint and overlapping constraints.**

a. Disjoint Constraints:

A disjoint constraint is the condition where the entity will belong only to one lower-level entity set. For example a student can be graduate or undergraduate but not both at once.

b. Overlapping Constraints:

Overlapping constraints allow an entity to belong to more than one lower-level entity sets within a single generalization. For example, a person can be both customer as well as an employee to a bank. The person can have their account in the bank and can be employed under the bank as well. Thus the generalization is overlapping.

**7.29 Explain the distinction between total and partial constraints.**

The completeness constraints on generalization specifies whether or not an entity set in higher-level entity set must belong to lower-level entity sets within the generalization.

1. Total design constraints:

Each higher-level entity must belong to a lower-level entity set. This is similar to total participation of entity sets. All the students under the university must be either a graduate or undergraduate. Here every higher-level entity set is related to a lower-level entity set.

1. Partial design constraints:

Some higher-level entities may not belong to any lower-level entity sets. For example if a worker is placed in a team only after s/he works for 3 months, then the higher-level entity(workers who haven’t worked for 3 months) may not belong to the team(lower-level entity set). Thus the higher-level entity set is not related to the lower-level entities.