a. Construct an E-R diagram that models exams as entities, and uses a ternary relationship, for the database.

b. Construct an alternative E-R diagram that uses only a binary relationship between *student* and *section*. Make sure that only one relationship exists between a particular *student* and *section* pair, yet you can represent the marks that a student gets in different exams.

7.3 Design an E-R diagram for keeping track of the exploits of your favorite sports team. You should store the matches played, the scores in each match, the players in each match, and individual player statistics for each match. Summary statistics should be modeled as derived attributes.

7.4 Consider an E-R diagram in which the same entity set appears several times, with its attributes repeated in more than one occurrence. Why is allowing this redundancy a bad practice that one should avoid?

7.5 An E-R diagram can be viewed as a graph. What do the following mean in terms of the structure of an enterprise schema?

a. The graph is disconnected.

b. The graph has a cycle.

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).

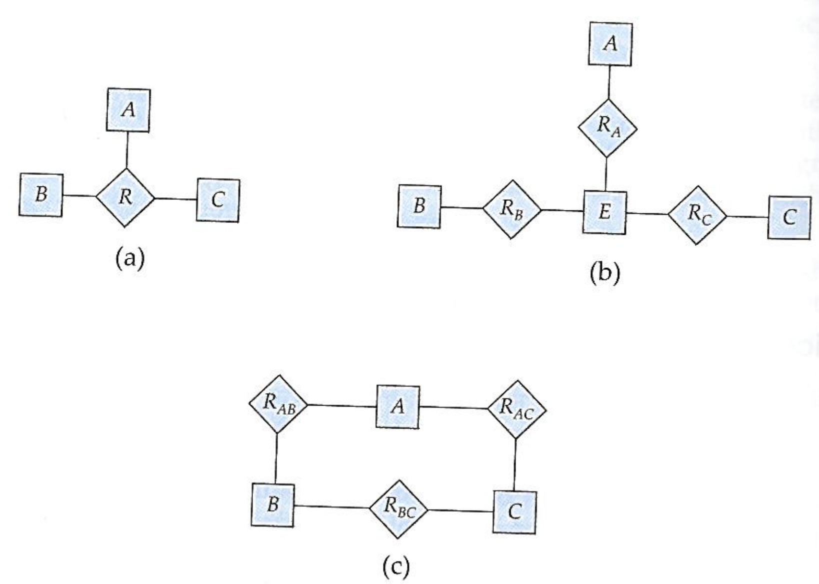

Figure 7.27 E-R diagram for Practice Exercise 7.6 and Exercise 7.24.

a. Show a simple instance of $E, A, B, C, R_A, R_B,$ and R_C that cannot correspond to any instance of $A, B, C,$ and R.

b. Modify the E-R diagram of Figure 7.27b to introduce constraints that will guarantee that any instance of $E, A, B, C, R_A, R_B,$ and R_C 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.

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.

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.

7.8 Consider a relation such as *sec_course*, generated from a many-to-one relationship *sec_course*. Do the primary and foreign key constraints created on the relation enforce the many-to-one cardinality constraint? Explain why.

7.9 Suppose the *advisor* relationship were one-to-one. What extra constraints are required on the relation *advisor*to ensure that the one-to-one cardinality constraint is enforced?

7.10 Consider a many-to-one relationship R between entity sets A and B. Suppose the relation created from R is combined with the relation created from A. In SQL, attributes participating in a foreign key constraint can be null. Explain how a constraint on total participation of A in R can be enforced using **not null** constraints in SQL.

7.11 In SQL, foreign key constraints can only reference the primary key attributes of the referenced relation, or other attributes declared to be a super key using the **unique** constraint. As a result, total participation constraints on a many-to-many relationship (or on the "one" side of a one-to-many relationship) cannot be enforced on the relations created from the relationship, using primary key, foreign key and not null constraints on the relations.

a. Explain why.

b. Explain how to enforce total participation constraints using complex check constraints or assertions (see Section 4.4.7). (Unfortunately, these features are not supported on any widely used database currently.)

7.12 Figure 7.28 shows a lattice structure of generalization and specialization (attributes not shown). For entity sets A, B, and C, explain how attributes