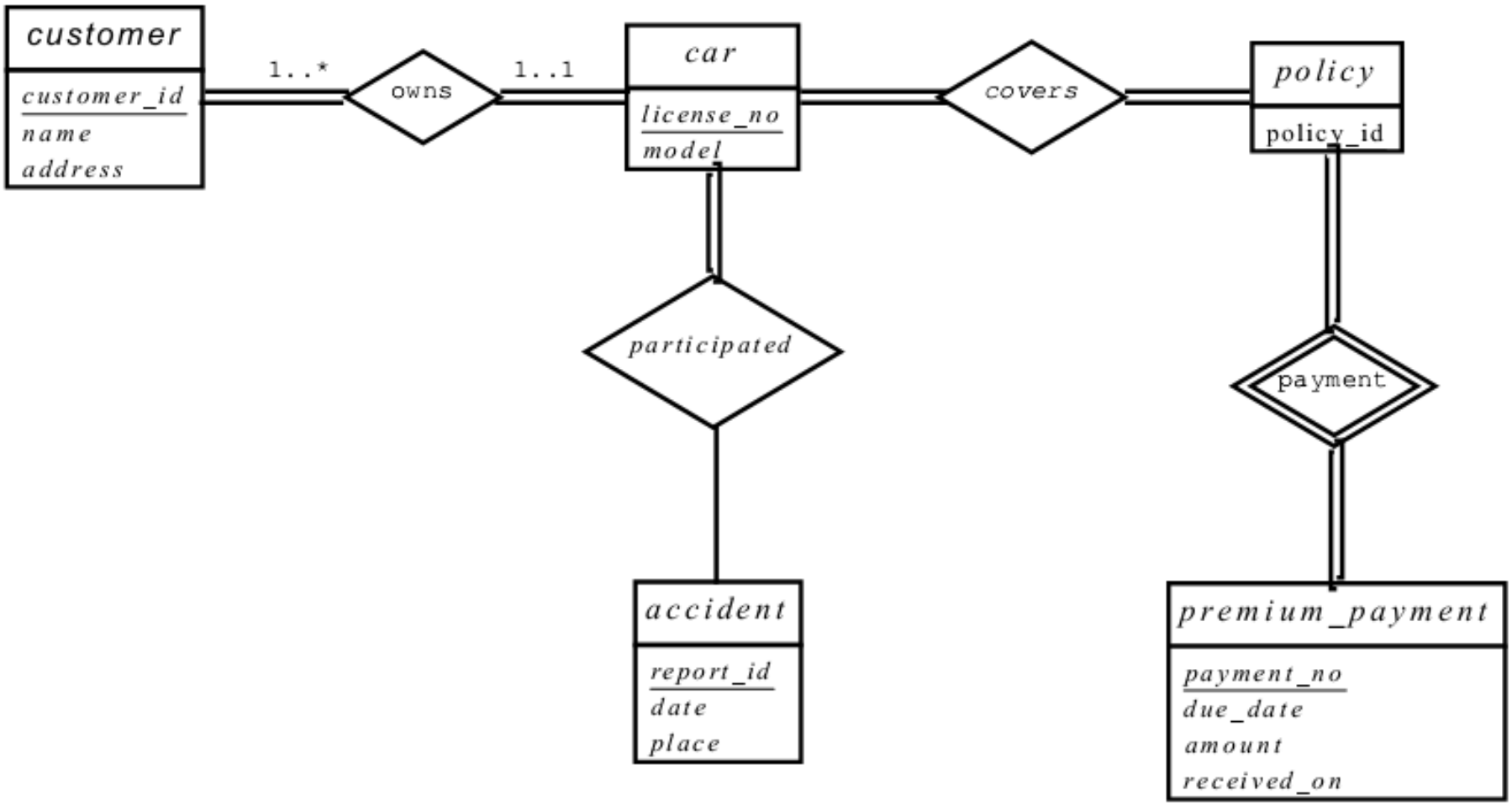
ER Examples

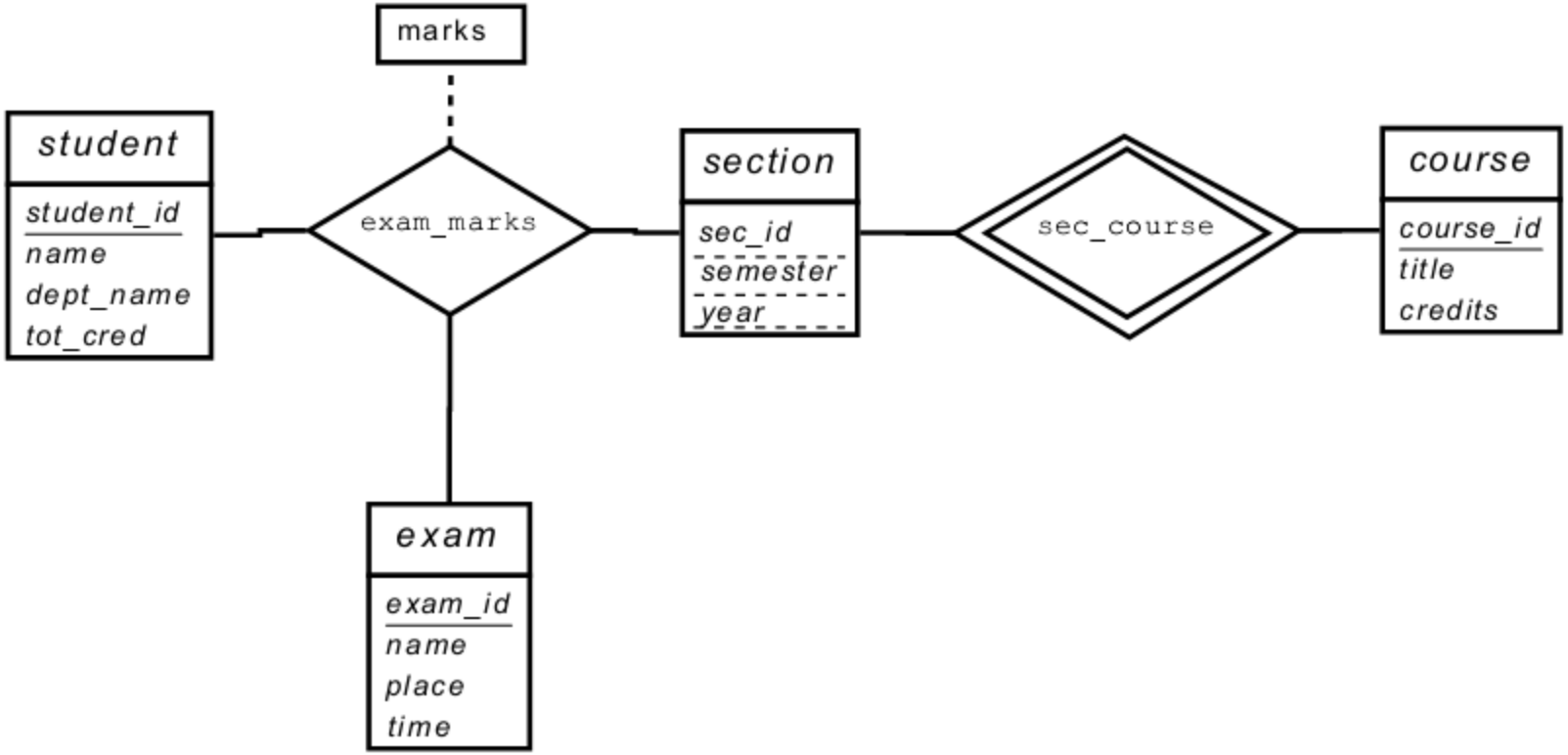
Construct an E-R diagram for a car insurance company whose customers own one or more cars each. Each car has associated with it zero to any number of recorded accidents. Each insurance policy covers one or more cars, and has one or more premium payments associated with it. Each payment is for a particular period of time, and has an associated due date, and the date when the payment was received.

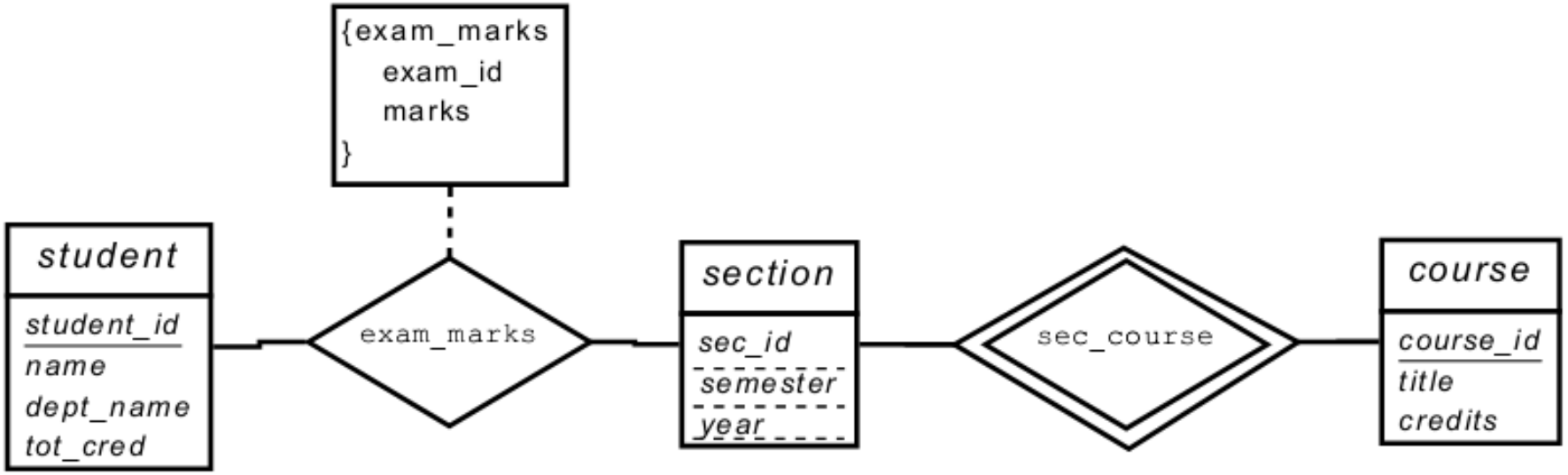

customer
customer_id
name
address
1..*
owns
1..1
car
license_no
model
covers
policy
policy_id
participated
payment
accident
report_id
date
place
premium_payment
payment_no
due_date
amount
received_on

Consider a database used to record the marks that students get in different exams of different course offerings (sections).

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

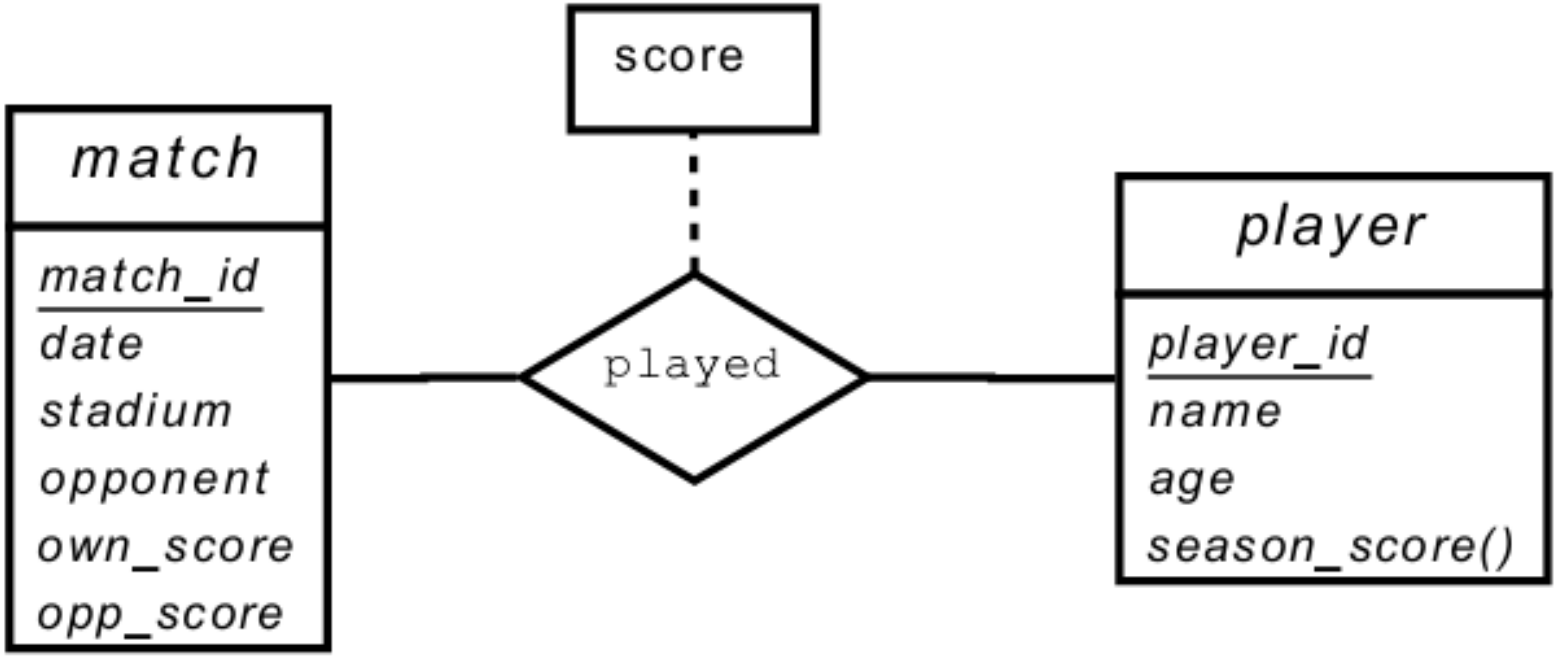
b. Construct an alternative E-R diagram that uses only a binary relationship between *student* and *section*. Make sure that only one relationship exists between a particular *student* and *section* pair, yet you can represent the marks that a student gets in different exams.


marks
student
student_id
name
dept_name
tot_cred
exam_marks
section
sec_id
semester
year
sec_course
course
course_id
title
credits
exam
exam_id
name
place
time

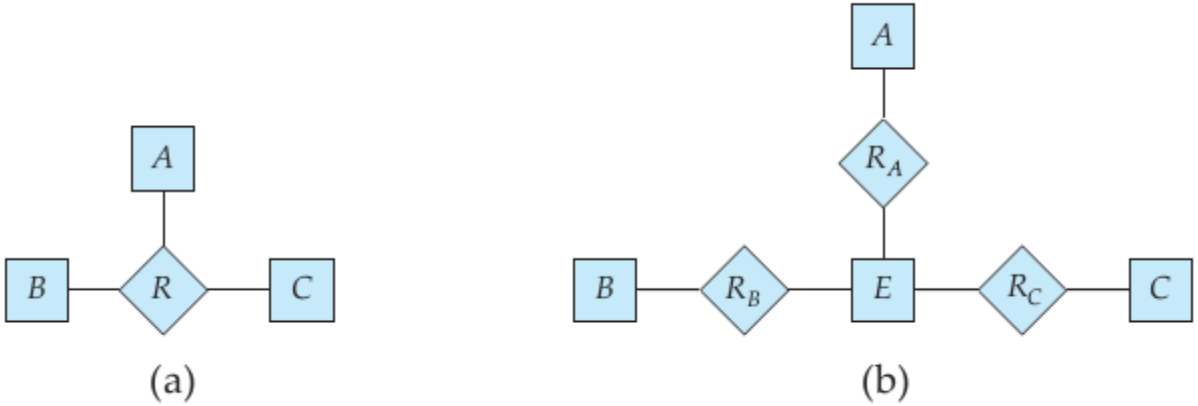


The E-R diagram is shown in Figure 7.3. Note that here we have not modeled the name, place and time of the exam as part of the relationship attributes. Doing so would result in duplication of the information, once per student, and we would not be able to record this information without an associated student. If we wish to represent this information, we would need to retain a separate entity corresponding to each exam.

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.

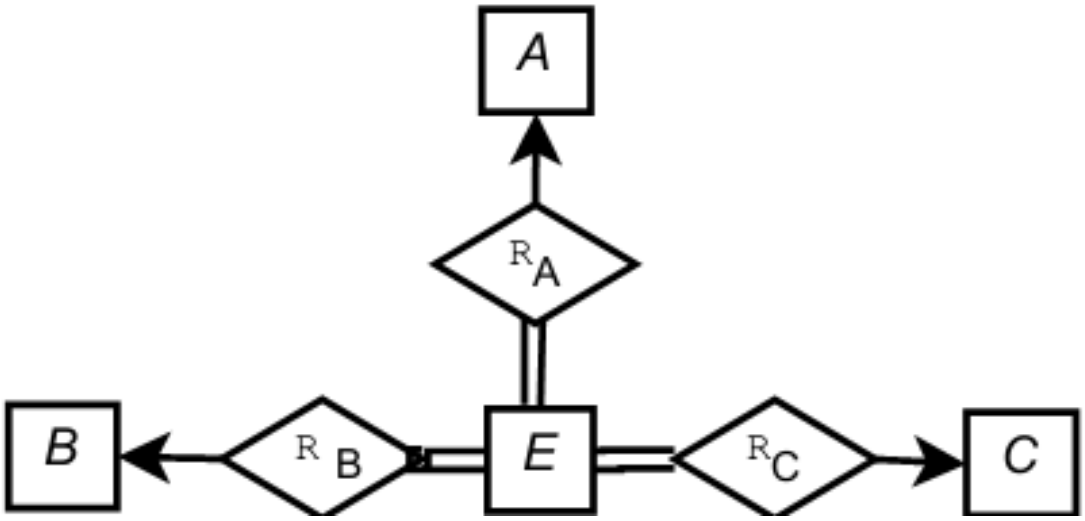


a. Show a simple instance of E, A, B, C, R_A, R_B, and R_C that cannot correspond to any instance of A, B, C, and R.



Let $E = \{e_1, e_2\}$, $A = \{a_1, a_2\}$, $B = \{b_1\}$, $C = \{c_1\}$, $R_A = \{(e_1, a_1), (e_2, a_2)\}$, $R_B = \{(e_1, b_1)\}$, and $R_C = \{(e_1, c_1)\}$. We see that because of the tuple (e_2, a_2), no instance of R exists which corresponds to E, R_A, R_B and R_C.

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.

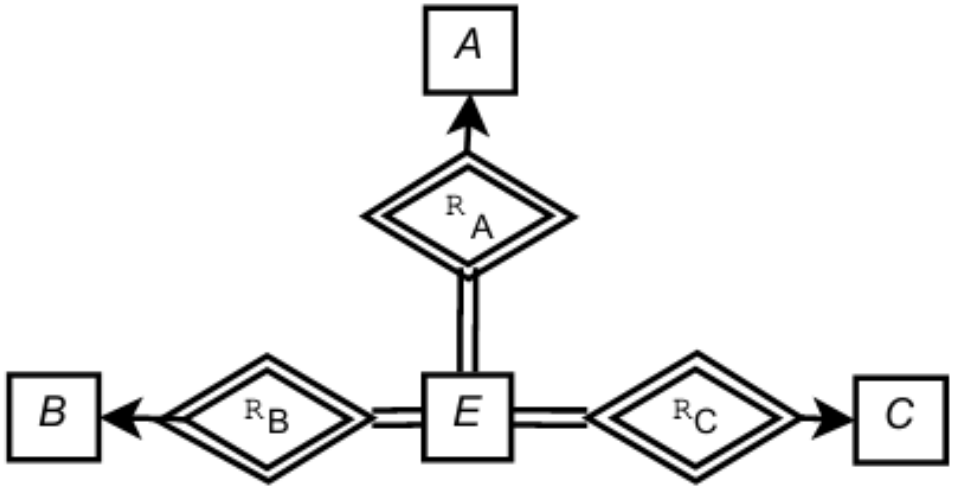


See Figure 7.5. The idea is to introduce total participation constraints between E and the relationships R_A, R_B, R_C so that every tuple in E has a relationship with A, B and C.

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

Suppose A totally participates in the relationhip R, then introduce a total participation constraint between A and R_A.

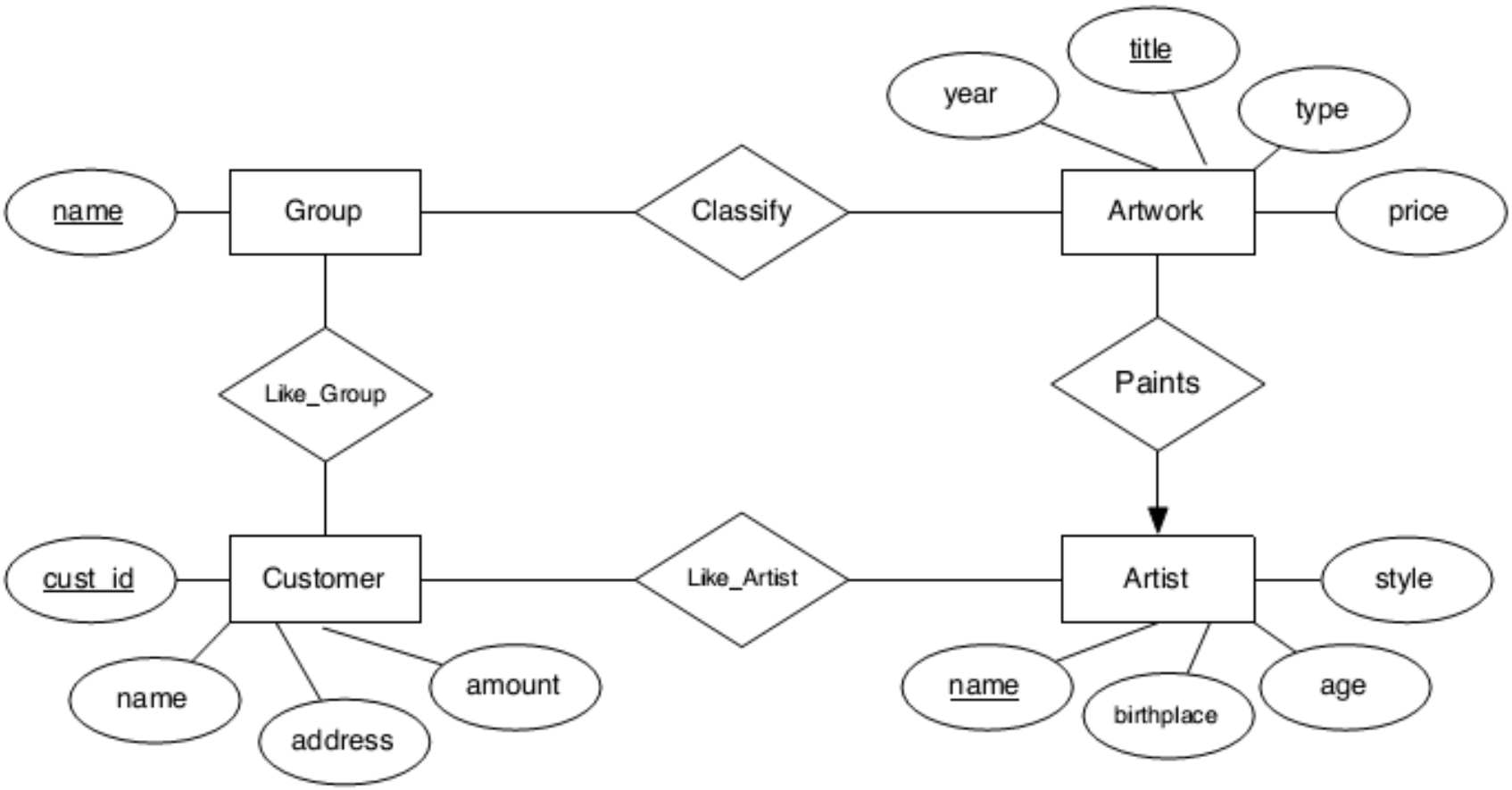
d. The above representation requires that we create a primary-key attribute for E. Show how to treat E as a weak entity set so that a primary-key attribute is not required.



Consider E as a weak entity set and R_A, R_B and R_C as its identifying relationship sets. See Figure 7.6.

- Galleries keep information about artists, their names (which are unique), birthplaces, age, and style of art.

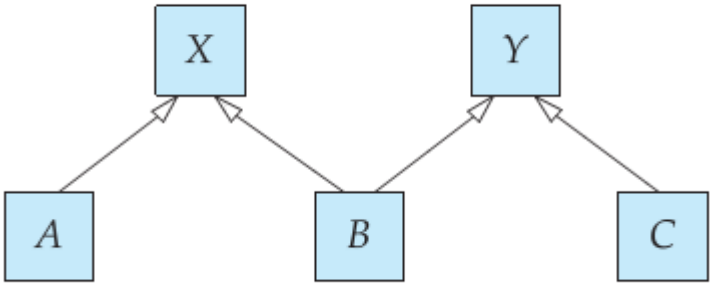
- For each piece of artwork, the artist, the year it was made, its unique title, its type of art (e.g., painting, lithograph, sculpture, photograph), and its price must be stored.
- Pieces of artwork are also classified into groups of various kinds, for example, portraits, still lifes, works by Picasso, or works of the 19th century; a given piece may belong to more than one group.
- Each group is identified by a name (like those just given) that describes the group.
- Finally, galleries keep information about customers. For each customer, galleries keep that persons unique name, address, total amount of dollars spent in the gallery, and the artists and groups of art that the customer tends to like.


title
year
type
name
Group
Classify
Artwork
price
Like_Group
Paints
cust_id
Customer
Like_Artist
Artist
style
name
amount
address
name
birthplace
age

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.

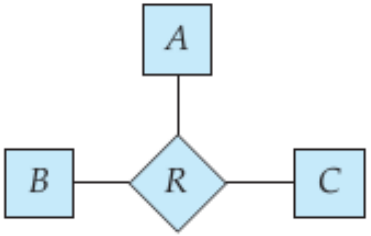
7.7 **Answer:** The primary key of a weak entity set can be inferred from its relationship with the strong entity set. If we add primary key attributes to the weak entity set, they will be present in both the entity set and the relationship set and they have to be the same. Hence there will be redundancy.

Figure 7.28 shows a lattice structure of generalization and specialization (attributes not shown). For entity sets A, B, and C, explain how attributes are inherited from the higher-level entity sets X and Y. Discuss how to handle a case where an attribute of X has the same name as some attribute of Y.

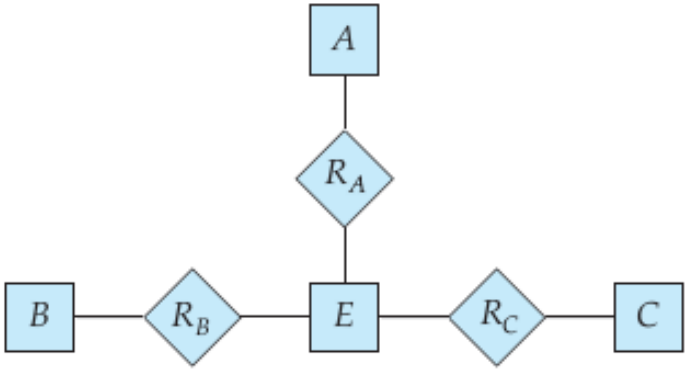


Answer: A inherits all the attributes of X plus it may define its own attributes. Similarly C inherits all the attributes of Y plus its own attributes. B inherits the attributes of both X and Y. If there is some attribute *name* which belongs to both X and Y, it may be referred to in B by the qualified name $X.name$ or $Y.name$.

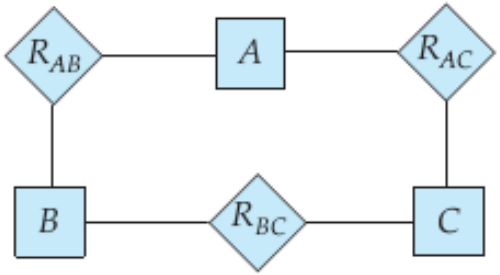
In Section 7.7.3, we represented a ternary relationship (repeated in Figure 7.27a) using binary relationships, as shown in Figure 7.27b. Consider the alternative shown in Figure 7.27c. Discuss the relative merits of these two alternative representations of a ternary relationship by binary relationships.



(a)



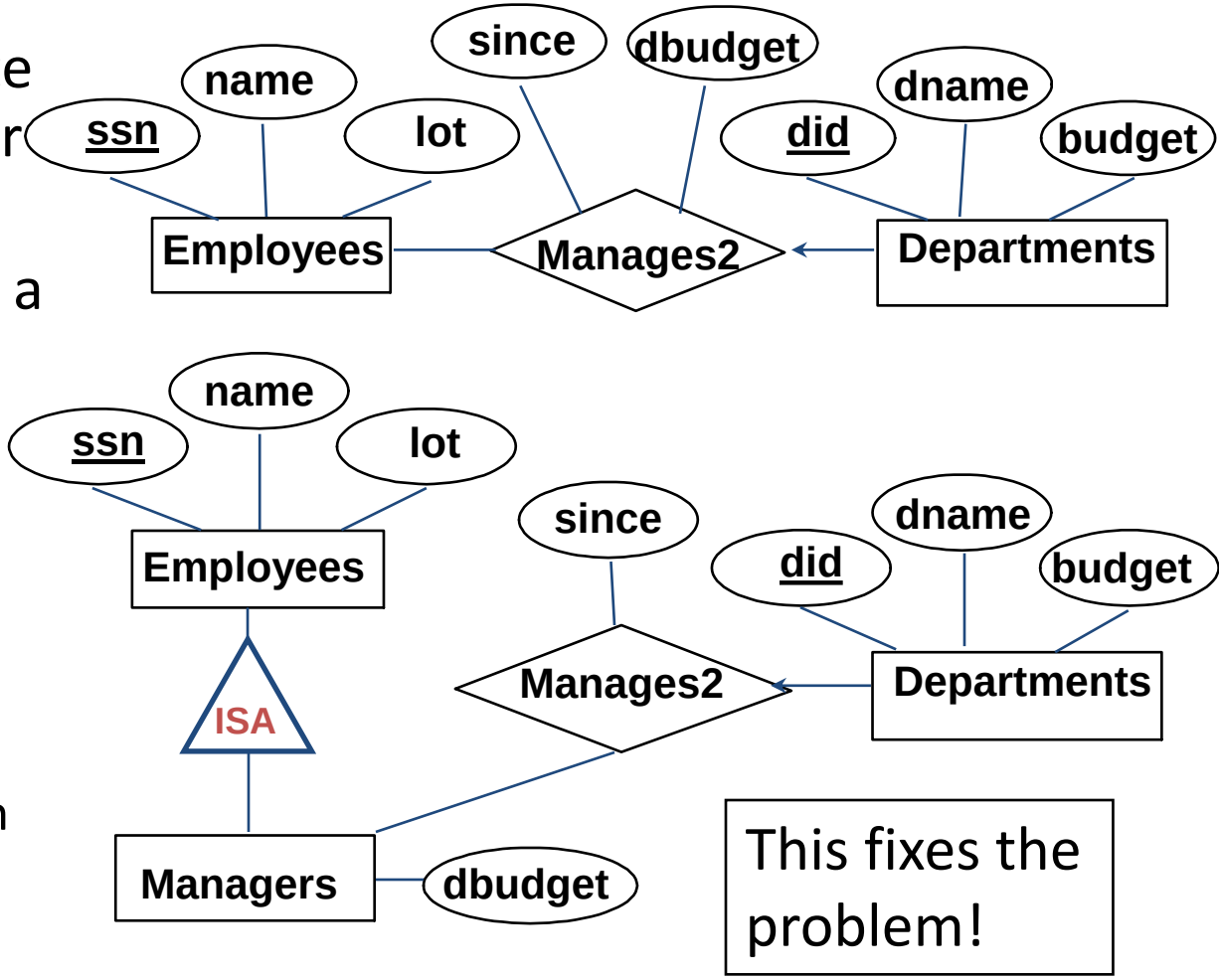
(b)



(c)

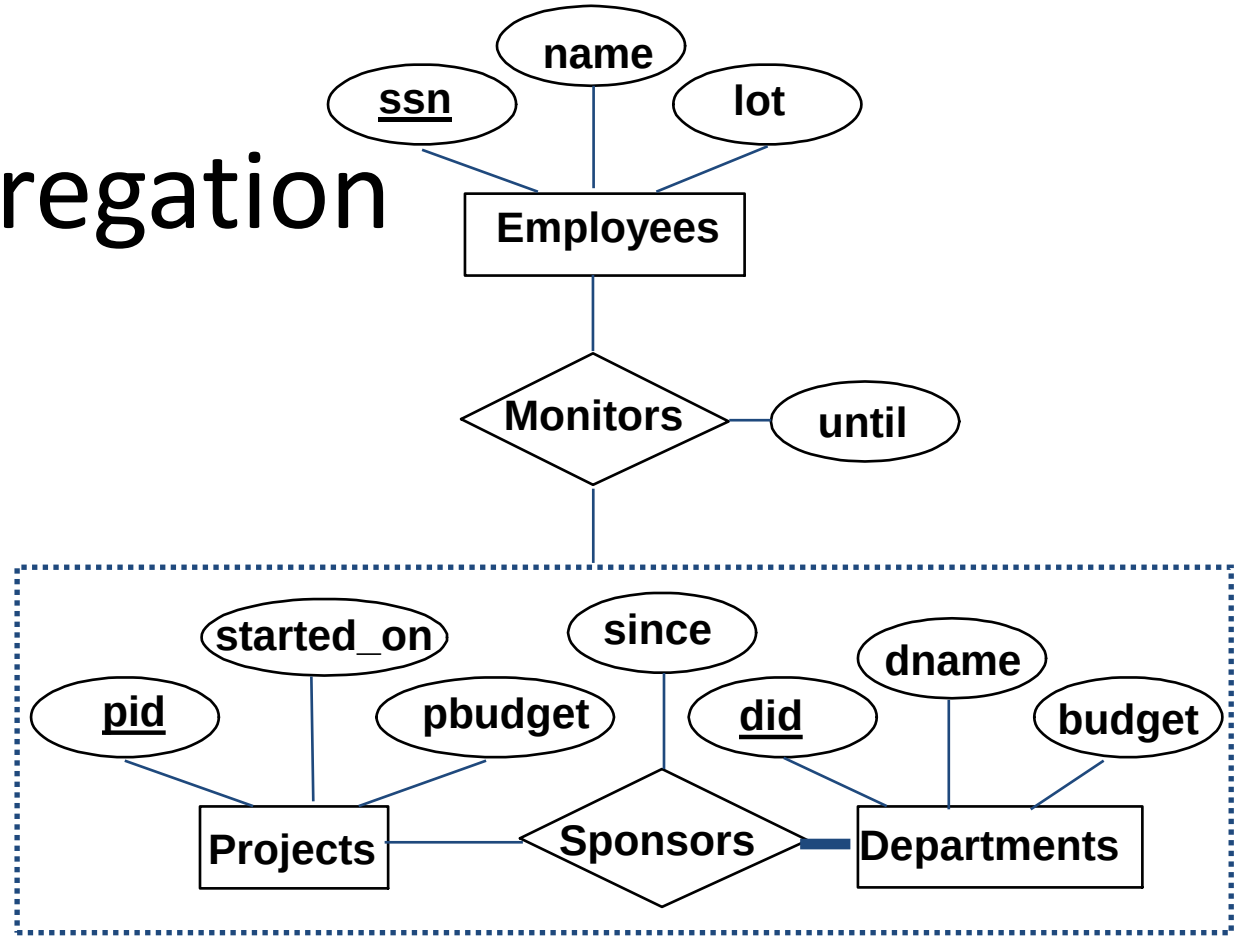
Entity vs. Relationship

- First ER diagram OK if a manager gets a separate discretionary budget for each dept.
- What if a manager gets a discretionary budget that covers *all* managed depts?
  - Redundancy: *dbudget* stored for each dept managed by manager.
  - Misleading: Suggests *dbudget* associated with department-mgr combination.



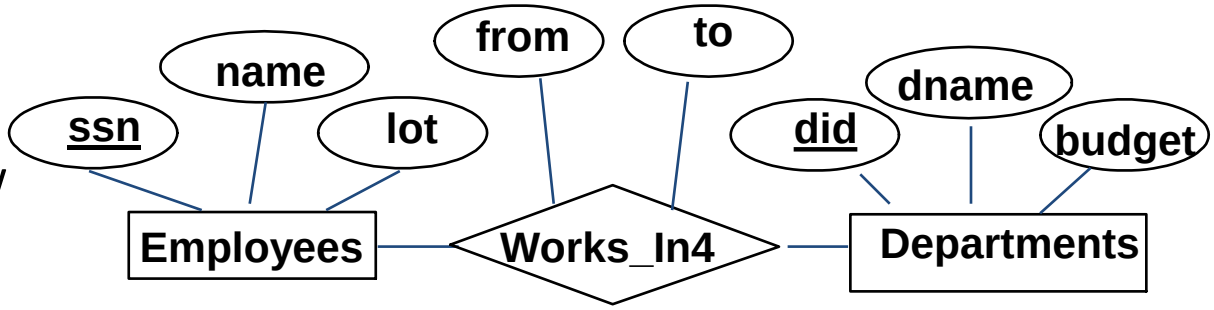
Aggregation

- Used when we have to model a relationship involving (entity sets and) a *relationship set*.
  - *Aggregation* allows us to treat a relationship set as an entity set for purposes of participation in (other) relationships.
  - Employees are assigned to monitor SPONSORSHIPS.

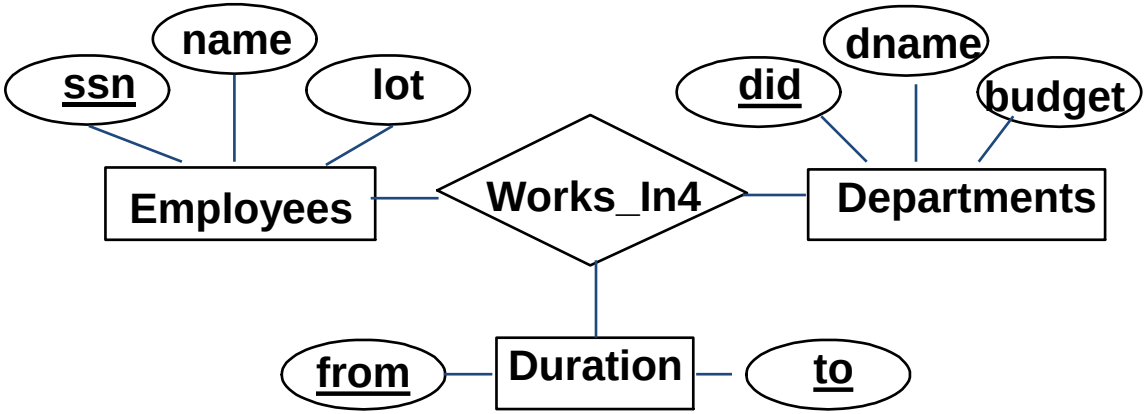


- *Aggregation vs. ternary relationship*:
  - Monitors and Sponsors are distinct relationships, with descriptive attributes of their own.
  - Also, can say that each sponsorship is monitored by at most one employee (which we cannot do with a ternary relationship).

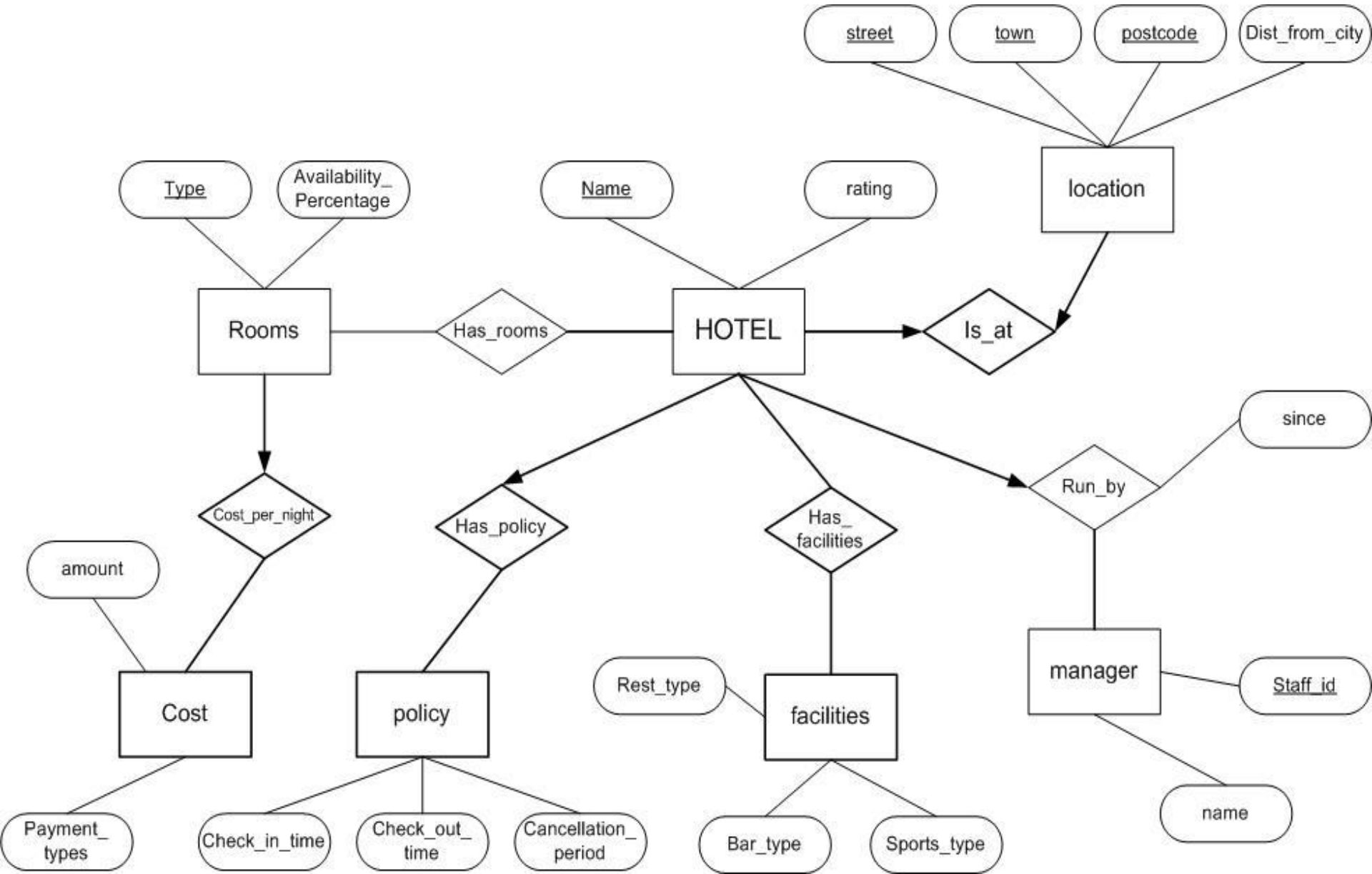
Entity vs. Attribute (Contd.)



- Works_In4 does not allow an employee to work in a department for two or more periods (a relationship is identified by participating entities).
- Similar to the problem of wanting to record several addresses for an employee: We want to record *several values of the descriptive attributes for each instance of this relationship.* Accomplished by introducing new entity set, Duration.



ERD (More Examples)



ERD (More Examples)