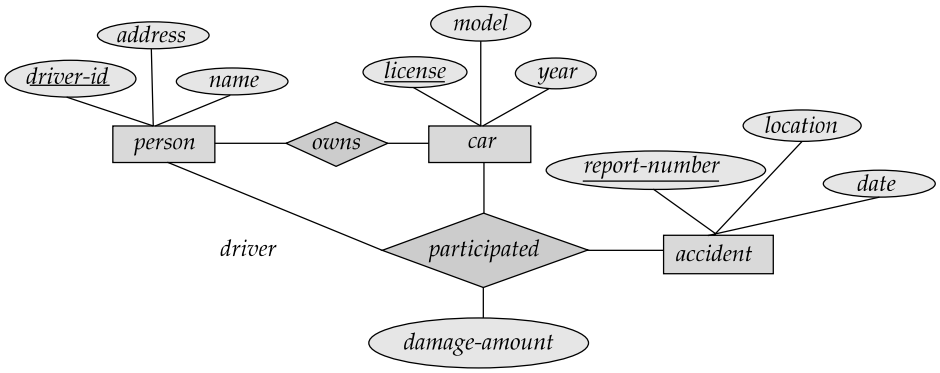

Figure 2.1 E-R diagram for a Car-insurance company.

Exercises

2.1 Explain the distinctions among the terms primary key, candidate key, and superkey.

Answer: A *superkey* is a set of one or more attributes that, taken collectively, allows us to identify uniquely an entity in the entity set. A superkey may contain extraneous attributes. If K is a superkey, then so is any superset of K. A superkey for which no proper subset is also a superkey is called a *candidate key*. It is possible that several distinct sets of attributes could serve as candidate keys. The *primary key* is one of the candidate keys that is chosen by the database designer as the principal means of identifying entities within an entity set.

2.2 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.

Answer: See Figure 2.1

2.3 Construct an E-R diagram for a hospital with a set of patients and a set of medical doctors. Associate with each patient a log of the various tests and examinations conducted.

Answer: See Figure 2.2

2.4 A university registrar's office maintains data about the following entities: (a) courses, including number, title, credits, syllabus, and prerequisites; (b) course offerings, including course number, year, semester, section number, instructor(s), timings, and classroom; (c) students, including student-id, name, and program; and (d) instructors, including identification number, name, department, and title. Further, the enrollment of students in courses and grades awarded to students in each course they are enrolled for must be appropriately modeled.

Construct an E-R diagram for the registrar's office. Document all assumptions that you make about the mapping constraints.

Answer: See Figure 2.3.

In the answer given here, the main entity sets are *student, course, course-offering*,

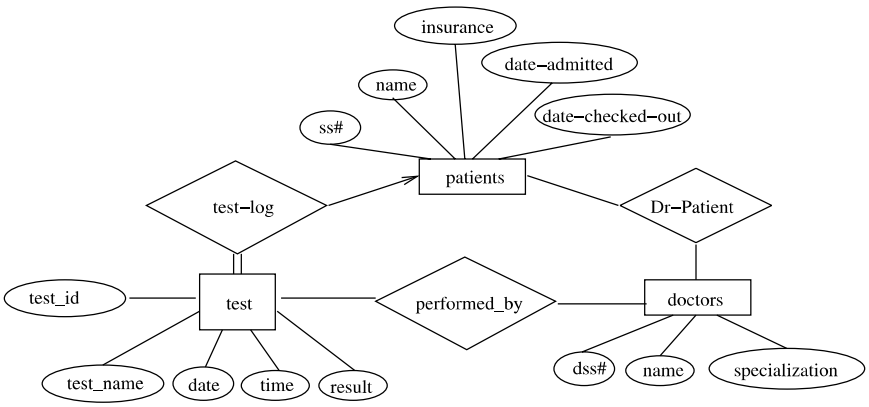

Figure 2.2 E-R diagram for a hospital.

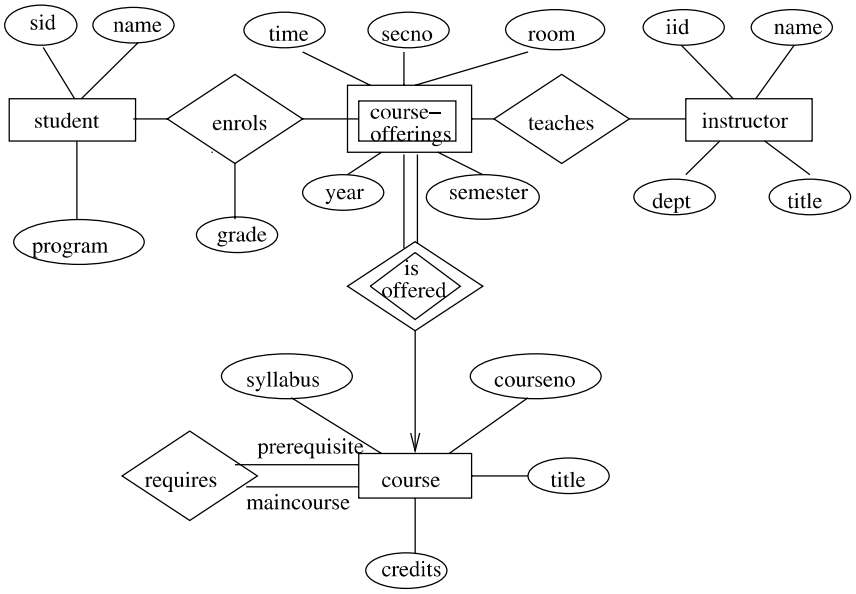

Figure 2.3 E-R diagram for a university.

and *instructor*. The entity set *course-offering* is a weak entity set dependent on *course*. The assumptions made are :

a. a class meets only at one particular place and time. This E-R diagram cannot model a class meeting at different places at different times.

b. There is no guarantee that the database does not have two classes meeting at the same place and time.

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

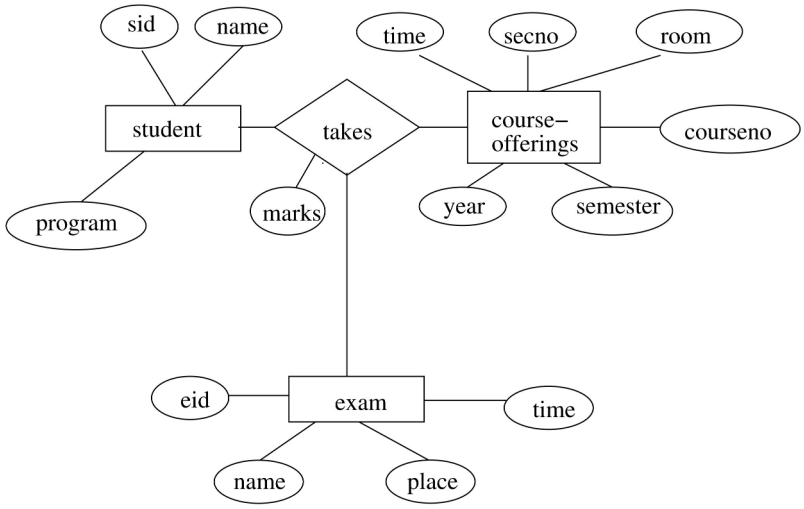

Figure 2.4 E-R diagram for marks database.

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

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

Answer:

a. See Figure 2.4

b. See Figure 2.5

2.6 Construct appropriate tables for each of the E-R diagrams in Exercises 2.2 to 2.4.

Answer:

a. Car insurance tables:

person (driver-id, name, address)
car (license, year, model)
accident (report-number, date, location)
participated(driver-id, license, report-number, damage-amount)

b. Hospital tables:

patients (patient-id, name, insurance, date-admitted, date-checked-out)
doctors (doctor-id, name, specialization)
test (testid, testname, date, time, result)
doctor-patient (patient-id, doctor-id)
test-log (testid, patient-id) performed-by (testid, doctor-id)

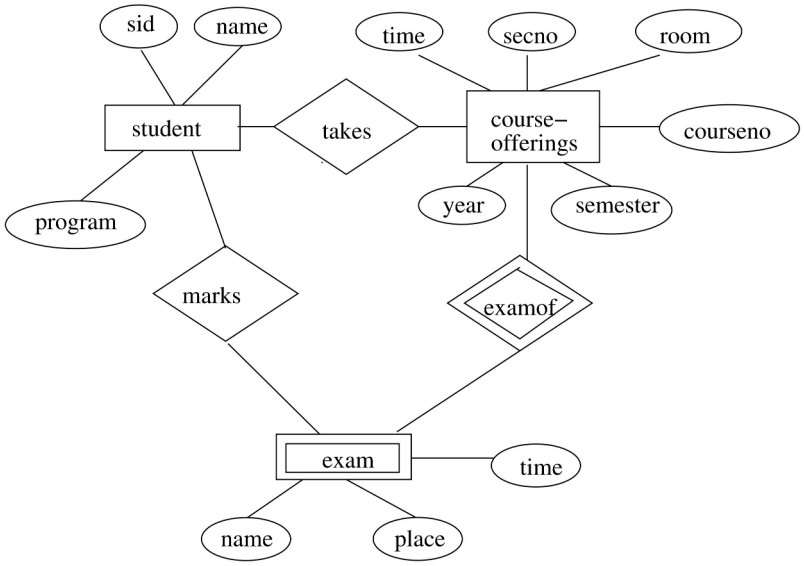

Figure 2.5 Another E-R diagram for marks database.

c. University registrar's tables:

student (<u>student-id</u>, name, program)
course (<u>courseno</u>, title, syllabus, credits)
course-offering (<u>courseno</u>, <u>secno</u>, <u>year</u>, <u>semester</u>, time, room)
instructor (<u>instructor-id</u>, name, dept, title)
enrols (<u>student-id</u>, <u>courseno</u>, <u>secno</u>, <u>semester</u>, <u>year</u>, grade)
teaches (<u>courseno</u>, <u>secno</u>, <u>semester</u>, <u>year</u>, <u>instructor-id</u>)
requires (<u>maincourse</u>, <u>prerequisite</u>)

2.7 Design an E-R diagram for keeping track of the exploits of your favourite 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.
Answer: See Figure 2.6

2.8 Extend the E-R diagram of the previous question to track the same information for all teams in a league.
Answer: See Figure 2.7 Note that a player can stay in only one team during a season.

2.9 Explain the difference between a weak and a strong entity set.
Answer: A strong entity set has a primary key. All tuples in the set are distinguishable by that key. A weak entity set has no primary key unless attributes of the strong entity set on which it depends are included. Tuples in a weak entity set are partitioned according to their relationship with tuples in a strong entity

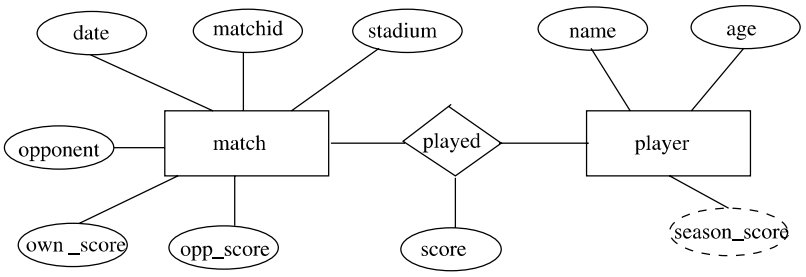

Figure 2.6 E-R diagram for favourite team statistics.

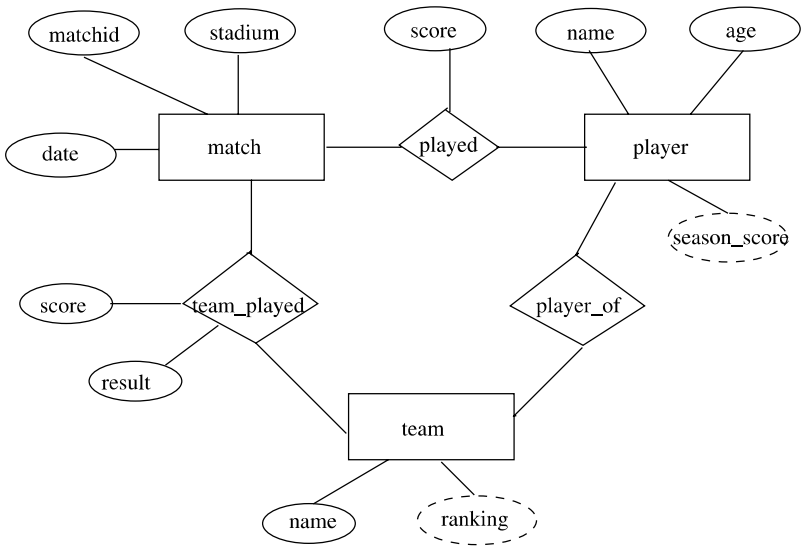

Figure 2.7 E-R diagram for all teams statistics.

set. Tuples within each partition are distinguishable by a discriminator, which is a set of attributes.

2.10 We can convert any weak entity set to a strong entity set by simply adding appropriate attributes. Why, then, do we have weak entity sets?
Answer: We have weak entities for several reasons:

- We want to avoid the data duplication and consequent possible inconsistencies caused by duplicating the key of the strong entity.
- Weak entities reflect the logical structure of an entity being dependent on another entity.
- Weak entities can be deleted automatically when their strong entity is deleted.
- Weak entities can be stored physically with their strong entities.

2.11 Define the concept of aggregation. Give two examples of where this concept is useful.

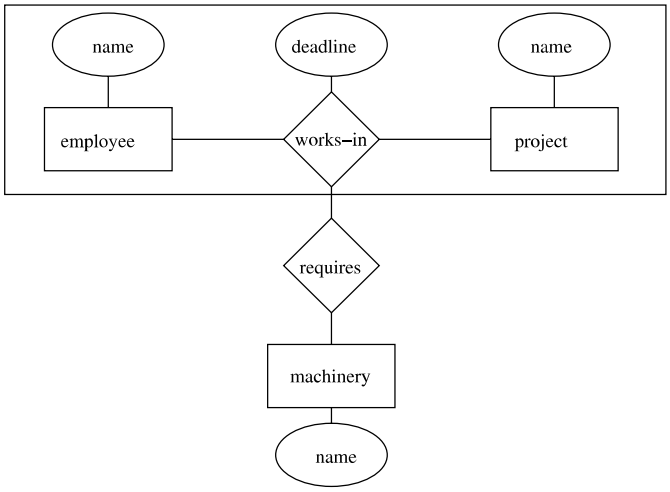

Figure 2.8 E-R diagram Example 1 of aggregation.

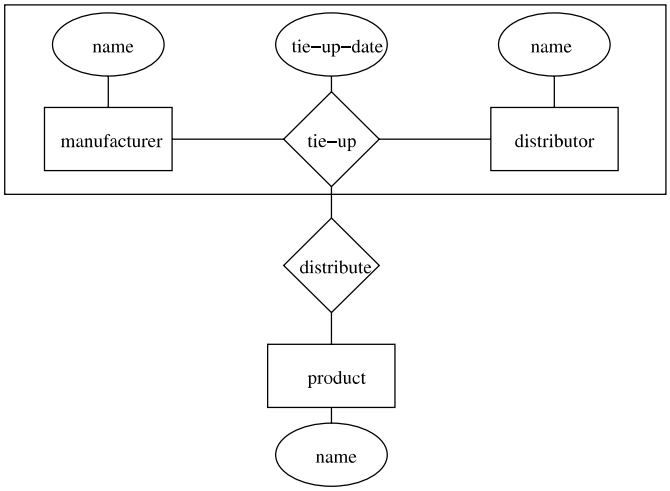

Figure 2.9 E-R diagram Example 2 of aggregation.

Answer: Aggregation is an abstraction through which relationships are treated as higher-level entities. Thus the relationship between entities A and B is treated as if it were an entity C. Some examples of this are:

a. Employees work for projects. An employee working for a particular project uses various machinery. See Figure 2.8

b. Manufacturers have tie-ups with distributors to distribute products. Each tie-up has specified for it the set of products which are to be distributed. See Figure 2.9

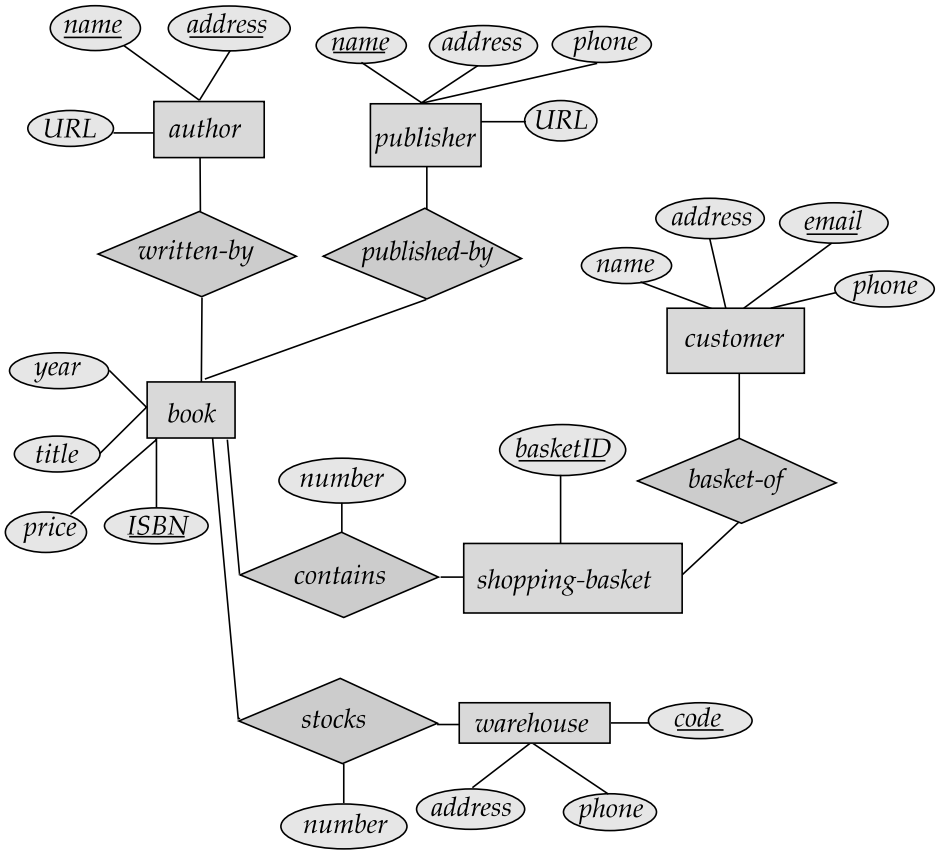

Figure 2.10 E-R diagram for Exercise 2.12.

2.12 Consider the E-R diagram in Figure 2.10, which models an online bookstore.

- **a.** List the entity sets and their primary keys.
- **b.** Suppose the bookstore adds music cassettes and compact disks to its collection. The same music item may be present in cassette or compact disk format, with differing prices. Extend the E-R diagram to model this addition, ignoring the effect on shopping baskets.
- **c.** Now extend the E-R diagram, using generalization, to model the case where a shopping basket may contain any combination of books, music cassettes, or compact disks.

Answer:

2.13 Consider an E-R diagram in which the same entity set appears several times. Why is allowing this redundancy a bad practice that one should avoid whenever possible?

Answer: By using one entity set many times we are missing relationships in

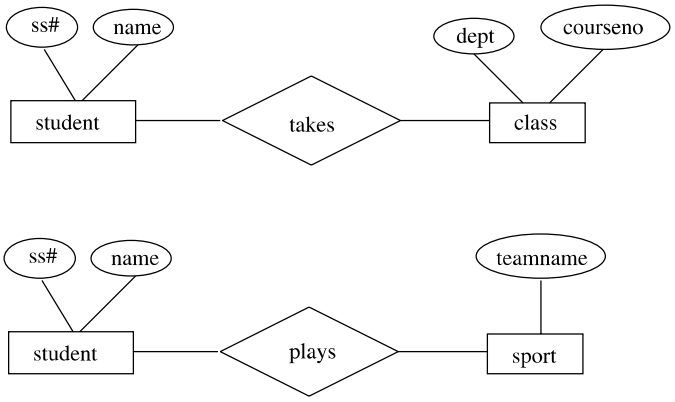

Figure 2.11 E-R diagram with entity duplication.

the model. For example, in the E-R diagram in Figure 2.11: the students taking classes are the same students who are athletes, but this model will not show that.

2.14 Consider a university database for the scheduling of classrooms for final exams. This database could be modeled as the single entity set *exam*, with attributes *course-name*, *section-number*, *room-number*, and *time*. Alternatively, one or more additional entity sets could be defined, along with relationship sets to replace some of the attributes of the *exam* entity set, as

- *course* with attributes *name*, *department*, and *c-number*
- *section* with attributes *s-number* and *enrollment*, and dependent as a weak entity set on *course*
- *room* with attributes *r-number*, *capacity*, and *building*

a. Show an E-R diagram illustrating the use of all three additional entity sets listed.

b. Explain what application characteristics would influence a decision to include or not to include each of the additional entity sets.

Answer:

a. See Figure 2.12

b. The additional entity sets are useful if we wish to store their attributes as part of the database. For the *course* entity set, we have chosen to include three attributes. If only the primary key (*c-number*) were included, and if courses have only one section, then it would be appropriate to replace the *course* (and *section*) entity sets by an attribute (*c-number*) of *exam*. The reason it is undesirable to have multiple attributes of *course* as attributes of *exam* is that it would then be difficult to maintain data on the courses, particularly if a course has no exam or several exams. Similar remarks apply to the *room* entity set.

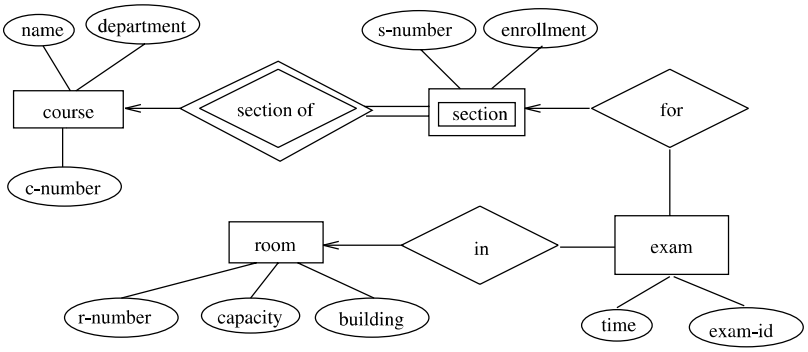

Figure 2.12 E-R diagram for exam scheduling.

2.15 When designing an E-R diagram for a particular enterprise, you have several alternatives from which to choose.

a. What criteria should you consider in making the appropriate choice?

b. Design three alternative E-R diagrams to represent the university registrar's office of Exercise 2.4. List the merits of each. Argue in favor of one of the alternatives.

Answer:

a. The criteria to use are intuitive design, accurate expression of the real-world concept and efficiency. A model which clearly outlines the objects and relationships in an intuitive manner is better than one which does not, because it is easier to use and easier to change. Deciding between an attribute and an entity set to represent an object, and deciding between an entity set and relationship set, influence the accuracy with which the real-world concept is expressed. If the right design choice is not made, inconsistency and/or loss of information will result. A model which can be implemented in an efficient manner is to be preferred for obvious reasons.

b. Consider three different alternatives for the problem in Exercise 2.4.

- See Figure 2.13
- See Figure 2.14
- See Figure 2.15

Each alternative has merits, depending on the intended use of the database. Scheme 2.13 has been seen earlier. Scheme 2.15 does not require a separate entity for *prerequisites*. However, it will be difficult to store all the prerequisites(being a multi-valued attribute). Scheme 2.14 treats prerequisites as well as classrooms as separate entities, making it useful for gathering data about prerequisites and room usage. Scheme 2.13 is in between the others, in that it treats prerequisites as separate entities but not classrooms. Since a registrar's office probably has to answer general questions about the number of classes a student is taking or what are all the prerequisites of a course, or where a specific class meets, scheme 2.14 is probably the best choice.