CHAPTER 7

Database Design and the E-R Model

Practice Exercises

7.1 **Answer:** The E-R diagram is shown in Figure 7.1. Payments are modeled as weak entities since they are related to a specific policy.
Note that the participation of accident in the relationship *participated* is not total, since it is possible that there is an accident report where the participating car is unknown.

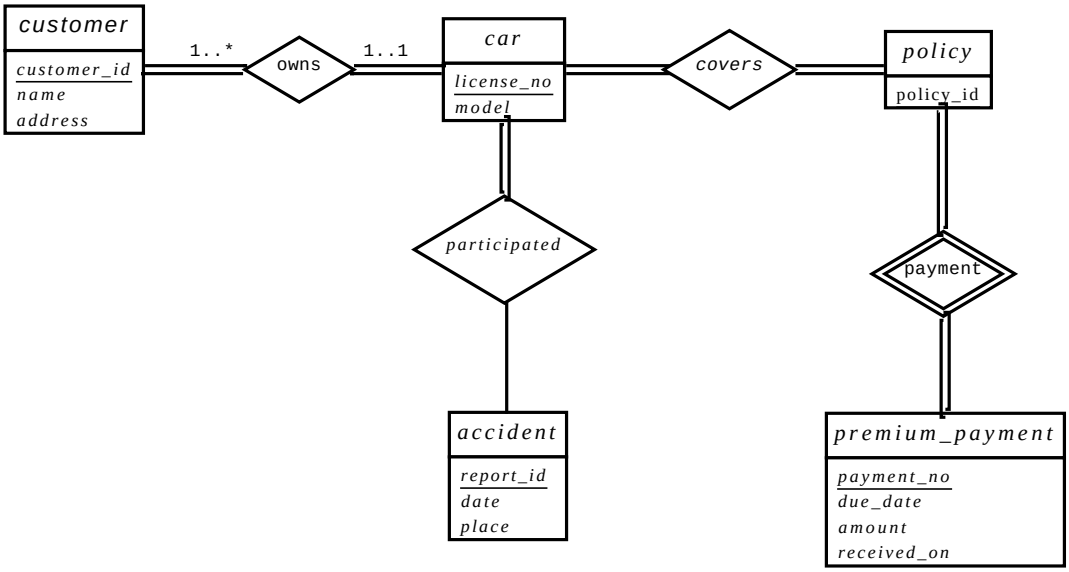

Figure 7.1 E-R diagram for a car insurance company.

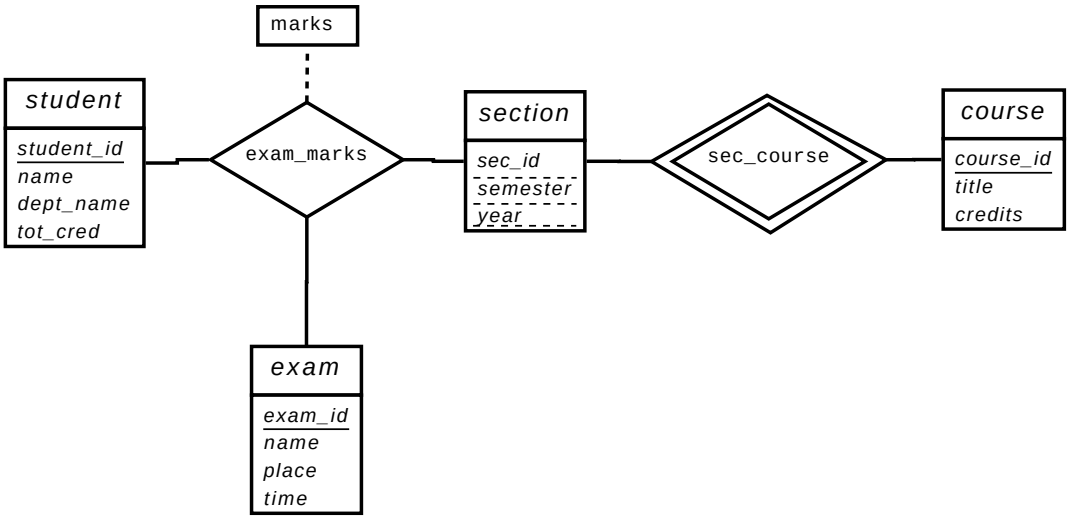

Figure 7.2 E-R diagram for marks database.

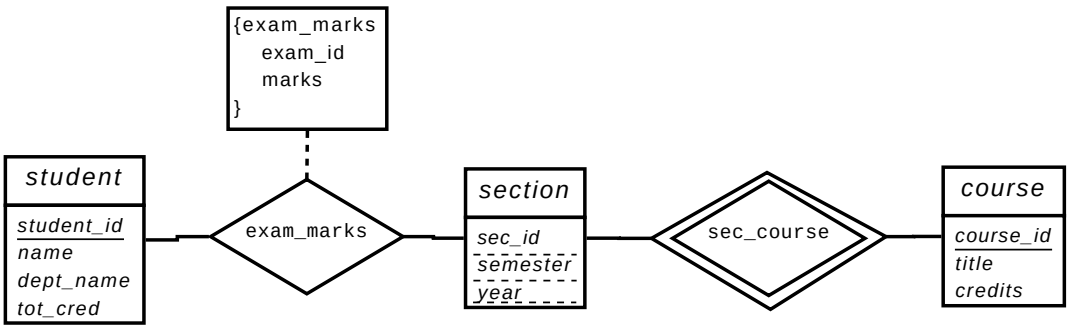

Figure 7.3 Another E-R diagram for marks database.

7.2 **Answer:** Note: the name of the relationship "course offering" needs to be changed to "section".

a. The E-R diagram is shown in Figure 7.2. Note that an alterantive is to model examinations as weak entities related to a section, rather than as a strong entity. The marks relationship would then be a binary relationship between *student* and *exam*, without directly involving *section*.

b. 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.

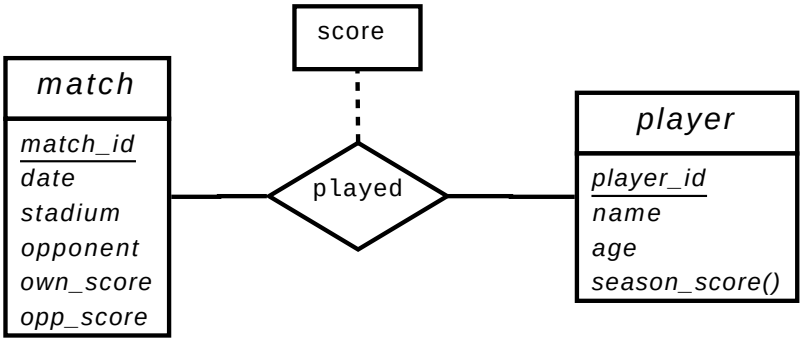

Figure 7.4 E-R diagram for favourite team statistics.

7.3 **Answer:** The diagram is shown in Figure 7.4.

7.4 **Answer:** The different occurrences of an entity may have different sets of attributes, leading to an inconsistent diagram. Instead, the attributes of an entity should be specified only once. All other occurrences of the entity should omit attributes. Since it is not possible to have an entity without any attributes, an occurrence of an entity without attributes clearly indicates that the attributes are specified elsewhere.

7.5 **Answer:**

a. If a pair of entity sets are connected by a path in an E-R diagram, the entity sets are related, though perhaps indirectly. A disconnected graph implies that there are pairs of entity sets that are unrelated to each other. In an enterprise, we can say that the two departments are completely independent of each other. If we split the graph into connected components, we have, in effect, a separate database corresponding to each connected component.

b. As indicated in the answer to the previous part, a path in the graph between a pair of entity sets indicates a (possibly indirect) relationship between the two entity sets. If there is a cycle in the graph then every pair of entity sets on the cycle are related to each other in at least two distinct ways. If the E-R diagram is acyclic then there is a unique path between every pair of entity sets and, thus, a unique relationship between every pair of entity sets.

7.6 **Answer:**

a. 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.

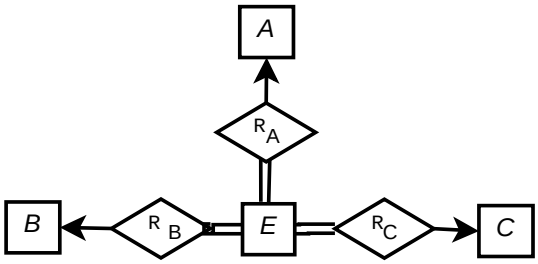

Figure 7.5 E-R diagram for Exercise 7.6b.

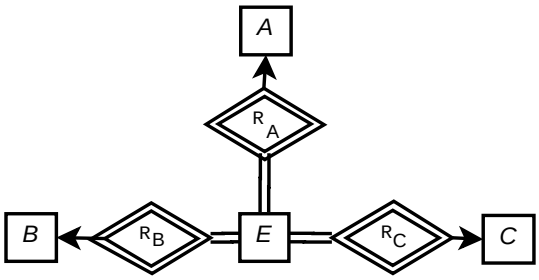

Figure 7.6 E-R diagram for Exercise 7.6d.

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

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

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.

7.8 **Answer:** In this example, the primary key of *section* consists of the attributes (*course_id*, *semester*, *year*), which would also be the primary key of *sec_course*, while *course_id* is a foreign key from *sec_course* referencing *course*. These constraints ensure that a particular *section* can only correspond to one *course*, and thus the many-to-one cardinality constraint is enforced. However, these constraints cannot enforce a total participation constraint, since a course or a section may not participate in the *sec_course* relationship.

7.9 **Answer:**

In addition to declaring *s_ID* as primary key for *advisor*, we declare *i_ID* as a super key for *advisor* (this can be done in SQL using the **unique** constraint on *i_ID*).

7.10 **Answer:** The foreign key attribute in R corresponding to primary key of B should be made **not null**. This ensures that no tuple of A which is not related to any entry in B under R can come in R. For example, say **a** is a tuple in A which has no corresponding entry in R. This means when R is combined with A, it would have foreign key attribute corresponding to B as **null** which is not allowed.

7.11 **Answer:**

a. For the many-to-many case, the relationship must be represented as a separate relation which cannot be combined with either participating entity. Now, there is no way in SQL to ensure that a primary key value occurring in an entity $E1$ also occurs in a many-to-many relationship R, since the corresponding attribute in R is not unique; SQL foreign keys can only refer to the primary key or some other unique key.
Similarly, for the one-to-many case, there is no way to ensure that an attribute on the one side appears in the relation corresponding to the many side, for the same reason.

b. Let the relation R be many-to-one from entity A to entity B with a and b as their respective primary keys, repectively. We can put the following check constraints on the "one" side relation B:

constraint *total_part* **check** (b in (**select** b **from** A));
set constraints *total_part* **deferred**;

Note that the constraint should be set to deferred so that it is only checked at the end of the transaction; otherwise if we insert a b value in B before it is inserted in A the above constraint would be violated, and if we insert it in A before we insert it in B, a foreign key violation would occur.

7.12 **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 $Y.name$.

7.13 **Answer:**

a. The E-R diagram is shown in Figure 7.7. The primary key attributes *student_id* and *instructor_id* are assumed to be immutable, that is they are not allowed to change with time. All other attributes are assumed to potentially change with time.
Note that the diagram uses multivalued composite attributes such as *valid_times* or *name*, with sub attributes such as *start_time* or *value*.

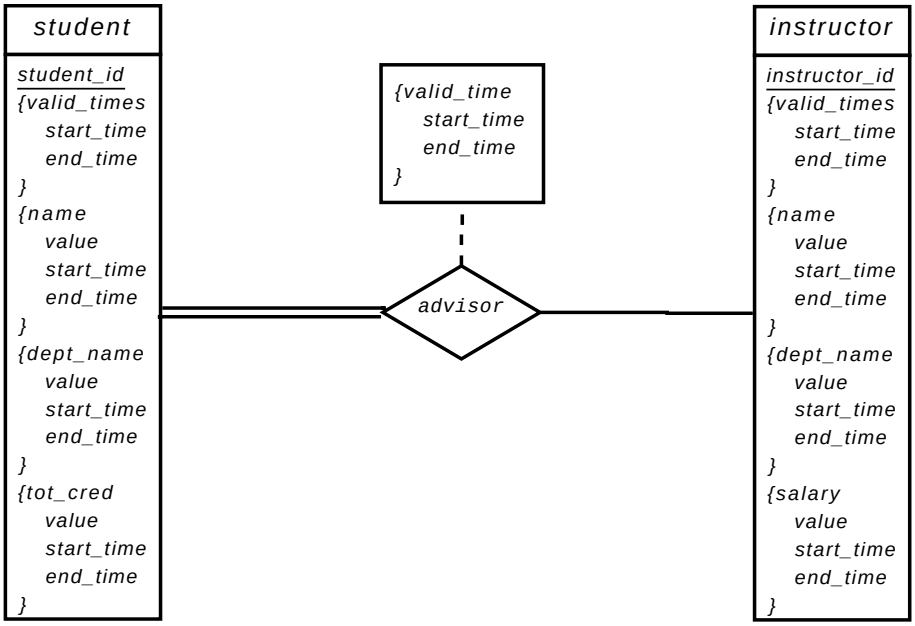

Figure 7.7 E-R diagram for Exercise 7.13

The *value* attribute is a subattribute of several attributes such as *name*, *tot_cred* and *salary*, and refers to the name, total credits or salary during a particular interval of time.

b. The generated relations are as shown below. Each multivalued attribute has turned into a relation, with the relation name consisting of the original relation name concatenated with the name of the multivalued attribute. The relation corresponding to the entity has only the primary key attribute.

student(student_id)
student_valid_times(student_id, start_time, end_time)
student_name(student_id, value, start_time, end_time
student_dept_name(student_id, value, start_time, end_time
student_tot_cred(student_id, value, start_time, end_time
instructor(instructor_id)
instructor_valid_times(instructor_id, start_time, end_time)
instructor_name(instructor_id, value, start_time, end_time
instructor_dept_name(instructor_id, value, start_time, end_time
instructor_salary(instructor_id, value, start_time, end_time
advisor(student_id, intructor_id, start_time, end_time)

The primary keys shown are derived directly from the E-R diagram. If we add the additional constraint that time intervals cannot overlap (or even the weaker condition that one start time cannot have two end times), we can remove the *end_time* from all the above primary keys.