

Alternative Diagrammatic Notations for ER Models

Figure A.1 shows a number of different diagrammatic notations for representing ER and EER model concepts. Unfortunately, there is no standard notation: different database design practitioners prefer different notations. Similarly, various CASE (computer-aided software engineering) tools and OOA (object-oriented analysis) methodologies use various notations. Some notations are associated with models that have additional concepts and constraints beyond those of the ER and EER models described in Chapters 7 through 9, while other models have fewer concepts and constraints. The notation we used in Chapter 7 is quite close to the original notation for ER diagrams, which is still widely used. We discuss some alternate notations here.

Figure A.1(a) shows different notations for displaying entity types/classes, attributes, and relationships. In Chapters 7 through 9, we used the symbols marked (i) in Figure A.1(a)—namely, rectangle, oval, and diamond. Notice that symbol (ii) for entity types/classes, symbol (ii) for attributes, and symbol (ii) for relationships are similar, but they are used by different methodologies to represent three different concepts. The straight line symbol (iii) for representing relationships is used by several tools and methodologies.

Figure A.1(b) shows some notations for attaching attributes to entity types. We used notation (i). Notation (ii) uses the third notation (iii) for attributes from Figure A.1(a). The last two notations in Figure A.1(b)—(iii) and (iv)—are popular in OOA methodologies and in some CASE tools. In particular, the last notation displays both the attributes and the methods of a class, separated by a horizontal line.

23/11/2020 10:12:26 - 665536 - 82.0.96.127

Figure A.1

Alternative notations. (a) Symbols for entity type/class, attribute, and relationship. (b) Displaying attributes. (c) Displaying cardinality ratios. (d) Various (min, max) notations. (e) Notations for displaying specialization/generalization.

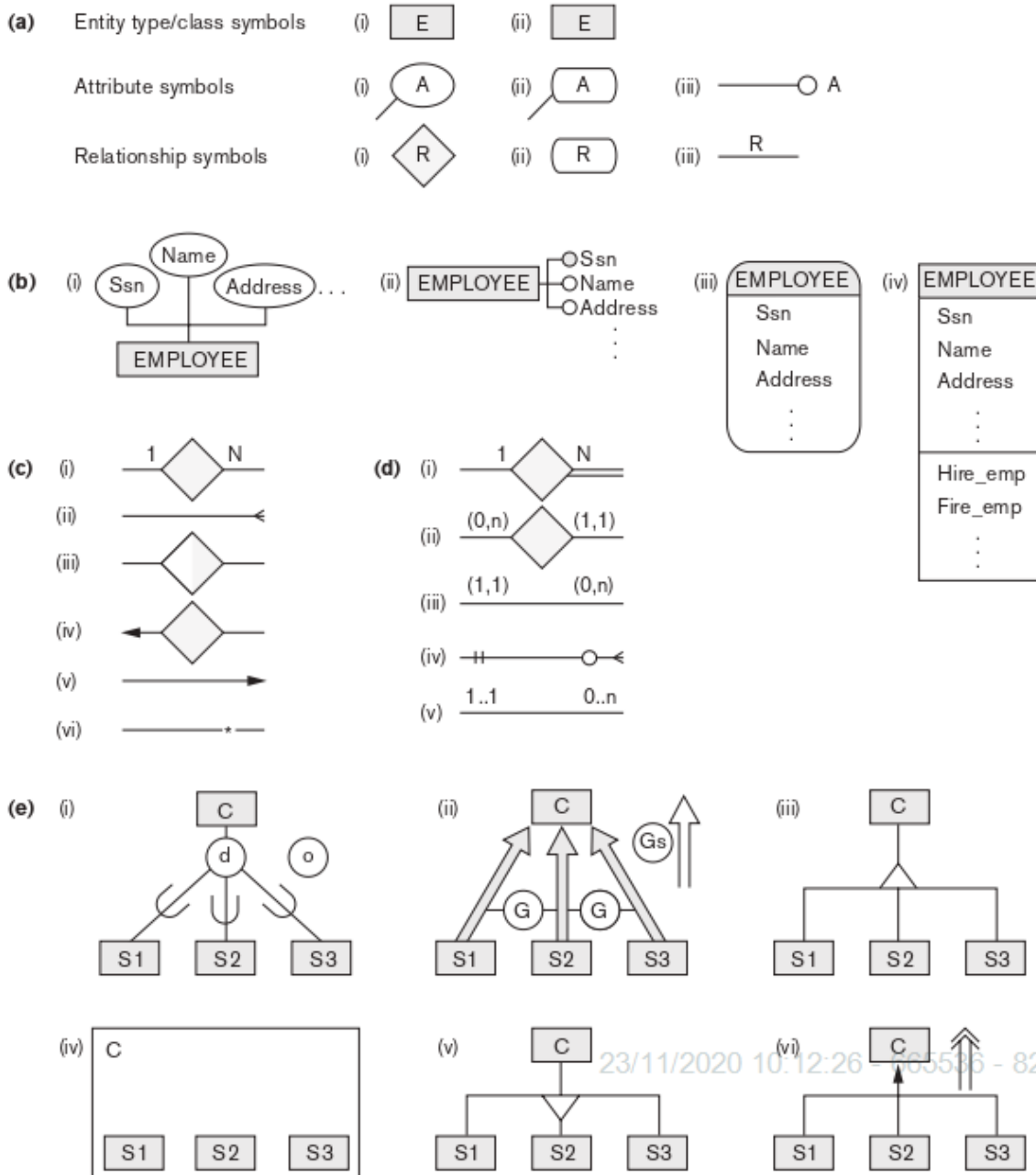


Figure A.1(c) shows various notations for representing the cardinality ratio of binary relationships. We used notation (i) in Chapters 7 through 9. Notation (ii)—known as the *chicken feet* notation—is quite popular. Notation (iv) uses the arrow as a functional reference (from the N to the 1 side) and resembles our notation for foreign keys in the relational model (see Figure 9.2); notation (v)—used in *Bachman diagrams* and the network data model—uses the arrow in the *reverse direction* (from the 1 to the N side). For a 1:1 relationship, (ii) uses a straight line without any chicken feet; (iii) makes both halves of the diamond white; and (iv) places arrowheads on both sides. For an M:N relationship, (ii) uses chicken feet at both ends of the line; (iii) makes both halves of the diamond black; and (iv) does not display any arrowheads.

Figure A.1(d) shows several variations for displaying (min, max) constraints, which are used to display both cardinality ratio and total/partial participation. We mostly used notation (i). Notation (ii) is the alternative notation we used in Figure 7.15 and discussed in Section 7.7.4. Recall that our notation specifies the constraint that each entity must participate in at least min and at most max relationship instances. Hence, for a 1:1 relationship, both max values are 1; for M:N, both max values are n. A min value greater than 0 (zero) specifies total participation (existence dependency). In methodologies that use the straight line for displaying relationships, it is common to *reverse the positioning* of the (min, max) constraints, as shown in (iii); a variation common in some tools (and in UML notation) is shown in (v). Another popular technique—which follows the same positioning as (iii)—is to display the *min* as o (“oh” or circle, which stands for zero) or as | (vertical dash, which stands for 1), and to display the *max* as | (vertical dash, which stands for 1) or as chicken feet (which stands for n), as shown in (iv).

Figure A.1(e) shows some notations for displaying specialization/generalization. We used notation (i) in Chapter 8, where a d in the circle specifies that the subclasses (S1, S2, and S3) are disjoint and an o in the circle specifies overlapping subclasses. Notation (ii) uses G (for generalization) to specify disjoint, and Gs to specify overlapping; some notations use the solid arrow, while others use the empty arrow (shown at the side). Notation (iii) uses a triangle pointing toward the superclass, and notation (v) uses a triangle pointing toward the subclasses; it is also possible to use both notations in the same methodology, with (iii) indicating generalization and (v) indicating specialization. Notation (iv) places the boxes representing subclasses within the box representing the superclass. Of the notations based on (vi), some use a single-lined arrow, and others use a double-lined arrow (shown at the side).

The notations shown in Figure A.1 show only some of the diagrammatic symbols that have been used or suggested for displaying database conceptual schemes. Other notations, as well as various combinations of the preceding, have also been used. It would be useful to establish a standard that everyone would adhere to, in order to prevent misunderstandings and reduce confusion.