COMPARISON OF ER MODELING NOTATIONS

Preview

The ER models used in this text are based on the Chen and Crow's Foot notations. However, you should be aware of other ER notations, including the Rein85 and the IDEF1X. Although those ER notations are based on the same modeling concepts, such alternative notations were developed because they fit computer-based modeling tools more easily than the original Chen notation. It is quite likely that you will convert even the finest Chen model into those (or very similar) models when you are using computer-assisted systems engineering (CASE) tools in the database environment.

Data Files Available on www.cengagebrain.com:

Appendix	Available formats				Questions and Problems	Available formats			
	MS Access	Oracle	MS SQL	MySQL		MS Access	Oracle	MS SQL	MySQL
There are no data files for this appendix.									
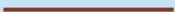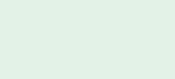
The following list summarizes the major characteristics of each notation:

- The *Chen model* is based on Peter Chen's landmark work "The Entity-Relationship Model: Toward a Unified View of Data," *ACM Transactions on Database Systems* 1(1), March 1971. The Chen model moved conceptual modeling into the practical database design arena by establishing the basic building blocks: entities and relationships. Chen's work was enhanced by T. J. Teorey, D. Yang, and J. P. Fry when they published "A Logical Design Methodology for Relational Databases Using the Extended Entity Relationship Model," *ACM Computing Surveys*, June 1986, pp. 197–222. The Chen model's basic structure, with the enhancements made by Teorey, Yang, and Fry, became a dominant player in the CASE tool market during the late 1980s and early 1990s. (See especially Mike Ricciuti's "Database Vendors Make Their CASE," *Datamation* 38(5), March 1992.) Excelerator, a CASE tool of choice for many database designers in the early 1990s, is perhaps the best "pure" Chen modeling tool. Although the Chen model is no longer the dominant ERD generator, all current ERD tools find their conceptual origins in the Chen model.

- The *Crow's Foot model*, developed by C. W. Bachman, was made popular by the Knowledgeware modeling tool. You used the Crow's Foot model extensively in Chapter 4, Entity Relationship (ER) Modeling, so you are already familiar with its notation. With that in mind, you will find the following comparisons between the various notations easier to understand if you use the Crow's Foot notation as your reference point. You also saw illustrations of the Chen model in Chapter 3, The Relational Database Model. You should remember that the Chen-style connectivity designations 1 and M and cardinality designations such as (0,1), (0,N), (1,1), and (1,N) are replaced by the Crow's Foot's sticklike symbols illustrated in Figure E.1. (The name "Crow's Foot" is a reflection of the symbol used for the connectivity M, which resembles a three-toed bird's foot.) Note that the Crow's Foot model combines connectivity and cardinality information in a single symbol set. Unlike the Chen methodology, the Crow's Foot model cannot detail cardinalities other than 0, 1, or N. For example, the cardinality (5,25) cannot be shown in a Crow's Foot model. However, the commercial modeling tools that use the Crow's Foot—such as Microsoft Visio Professional—let you add those cardinalities to the diagram, using text, and to define the cardinalities in a data dictionary.

- The *Rein85 model* was developed by D. Reiner in 1985. Although the Rein85 model is based on the same modeling conventions as the Crow's Foot, its symbols are quite different. It also operates at a higher level of abstraction than the Crow's Foot. The Rein85 methodology does not recognize cardinalities explicitly, relying on connectivities to lead to logical cardinality conclusions. The Rein85 symbols are displayed in Figure E.1.

- *IDEF1X* is a derivative of the integrated computer-aided manufacturing (ICAM) studies that were conducted in the late 1970s. ICAM became the source of graphical methods for defining the functions, data structures, and dynamics of manufacturing businesses. The integration of those methods became known as IDEF (ICAM DEFinition). The original version of IDEF, developed by Hughes Aircraft, became known as IDEF1. The extended version of IDEF1, known as IDEF1X, became the U.S. Air Force standard and has gained acceptance as a general manufacturing data-modeling tool. As you examine Figure E.1, note that IDEF1X uses fewer symbols than the other modeling methods, thus providing fewer explicit details of the type and extent of the relationships being modeled.

NOTE

Figure E.1 shows Crow's Foot composite and weak entities to reflect the early implementations of that model. However, modern modeling tools, such as Microsoft Visio, do not depict the composite and weak entities. Instead, the existence of weak and composite entities is inferred from the relationship lines, which are solid when the relationship between parent and child entities is strong or identifying. In addition, the nature of the entity can be established by examining the PK/FK depictions. Therefore, the special weak/composite depictions are redundant in the Crow's Foot model.

FIGURE E.1 A comparison of ER modeling symbols

Cengage Learning © 2015

To illustrate the use of the four methods, let's examine the invoicing example discussed in Chapter 3, Section 3.7. The invoicing example is based on the following business rules:

- A CUSTOMER may generate zero INVOICEs, one INVOICE, or many INVOICEs. An INVOICE is generated by one CUSTOMER.

- An INVOICE refers to many PRODUCTs—for example, you can sell many hammers over some period of time, and a PRODUCT may or may not be referred to in many INVOICEs. (Products that are stocked are not necessarily sold.) You should remember from Chapter 4 that the M:N relationship between INVOICE and PRODUCT is implemented through LINE, in order to decompose the M:N relationship into two 1:M relationships. Therefore, LINE becomes optional to PRODUCT because an unsold product will never appear in an invoice line.

Based on the preceding business rules, the four ERD notations are illustrated in Figures E.2 through E.5.

FIGURE E.2 **The Chen ERD for the invoicing problem**

This model may be read as follows:

each CUSTOMER *may* generate one or more INVOICEs
each INVOICE is generated by one CUSTOMER

each INVOICE contains one or more invoice LINEs
each invoice LINE is contained in an INVOICE

each invoice LINE references one PRODUCT
each PRODUCT *may* be referenced in one or more invoice LINEs

Cengage Learning © 2015

FIGURE E.3 **The Crow's Foot ERD for the invoicing problem**

This model may be read as follows:

each CUSTOMER *may* generate one or more INVOICEs
each INVOICE is generated by one CUSTOMER

each INVOICE contains one or more invoice LINEs
each invoice LINE is contained in one INVOICE

each invoice LINE references one PRODUCT
each PRODUCT *may* be referenced in one or more invoice LINEs
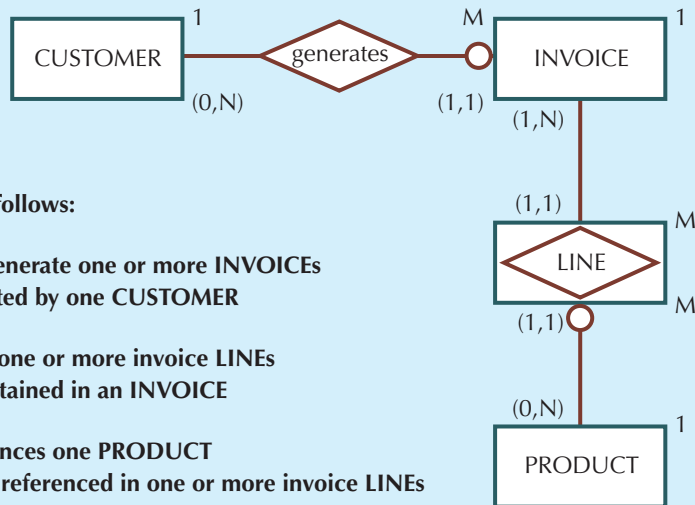
Cengage Learning © 2015

FIGURE E.4 The Rein85 ERD for the invoicing problem

This model may be read as follows:

each CUSTOMER *may* generate one or more INVOICEs
each INVOICE is generated by one CUSTOMER

each INVOICE contains one or more invoice LINEs
each invoice LINE is contained in an INVOICE

each invoice LINE references one PRODUCT
each PRODUCT *may* be referenced in one or more invoice LINEs
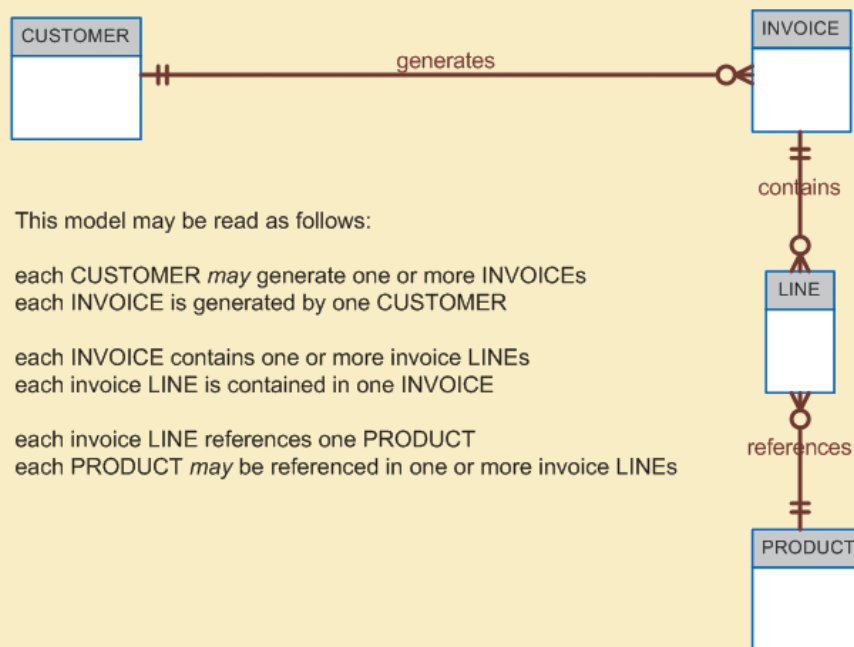
Cengage Learning © 2015

FIGURE E.5 The IDEF1X ERD for the invoicing problem

This model may be read as follows:

each CUSTOMER *may* generate one or more INVOICEs
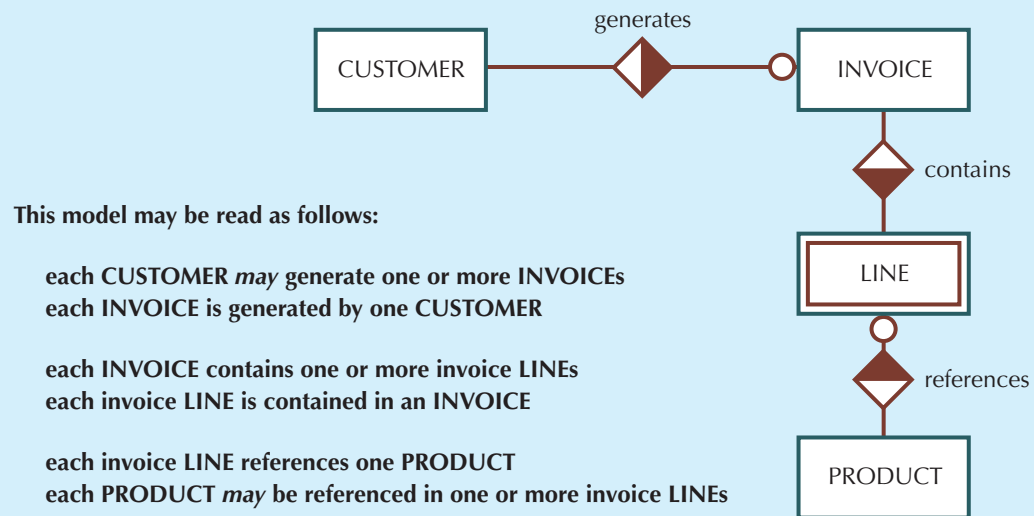each INVOICE is generated by one CUSTOMER

each INVOICE contains one or more invoice LINEs
each invoice LINE is contained in an INVOICE

each invoice LINE references one PRODUCT
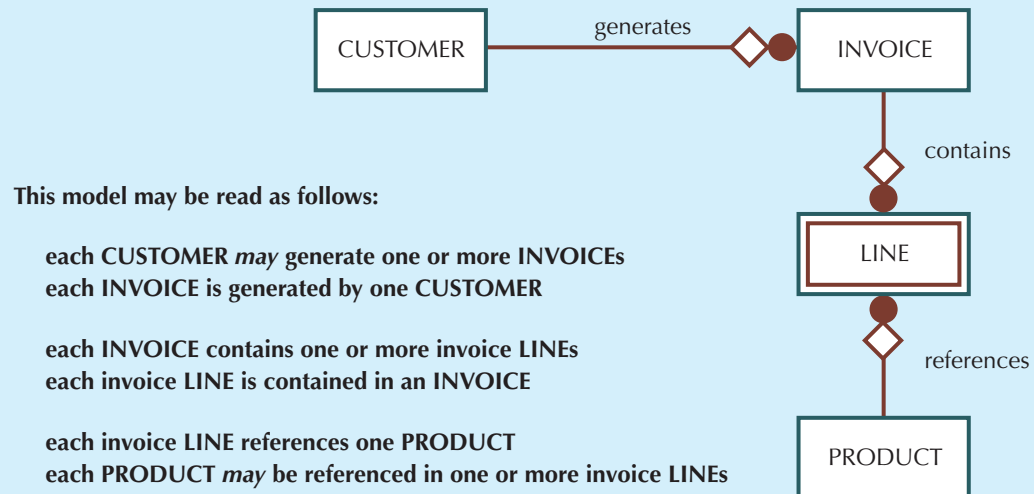each PRODUCT *may* be referenced in one or more invoice LINEs

Cengage Learning © 2015

Not For Sale