11ed Chapter 4

Entity-Relationship (ER) Modeling

Discussion Focus

This chapter attempts to present a fairly balanced view of Entity-Relationship Modeling. One the one hand, a pragmatic focus on the aspects of the ERD that directly impact the design and implementation of the database must be considered. On the other hand, properly documenting the business needs that the database must support is equally important. Therefore, we present both Crow's Foot and Chen notation ERDs and discuss the differences between them. Crow's Foot tends to be more pragmatic, while Chen notation has greater semantic content. One exercise that we have found very valuable for our students is after working with both notations is to have the students engage in a discussion of the strengths and weaknesses of the different notations.

Among the points of comparison that we focus on in class are:

- In both notations, entities are modeled the same.
- Attributes are modeled differently – in ovals with Chen notation and in an attribute box with Crow's Foot.
- Identifiers are modeled the same.
- Single-valued attributes are modeled the same.
- Chen notation distinguishes multi-valued attributes with a double-line; however, Crow's Foot notation does not distinguish multi-valued attributes. This can be attributed to the pragmatism of Crow's Foot notation – if you wouldn't implement the design with a multi-valued attribute, then don't draw the design with a multi-valued attribute.
- Chen notation distinguishes derived attributes with a dotted attribute line; however Crow's Foot notation does not distinguish derived attributes.
- Chen notation commonly uses specific minimum and maximum cardinalities. Specific minimum and maximum cardinalities are almost never used in practice with Crow's Foot notations, which is why in practice most people who use Crow's Foot notation refer to connectivity as maximum cardinality and participation as minimum cardinality.
- Both notations typically distinguish weak entities; however, as a peculiarity of the MS Visio tool, weak entities are implied through relationship strength.

Student's tend to catch on to the notations rather quickly; however, giving them some simple modeling tasks where the data model components (entities, attributes, etc.) are already identified that merely require them to apply the notation can be very helpful.

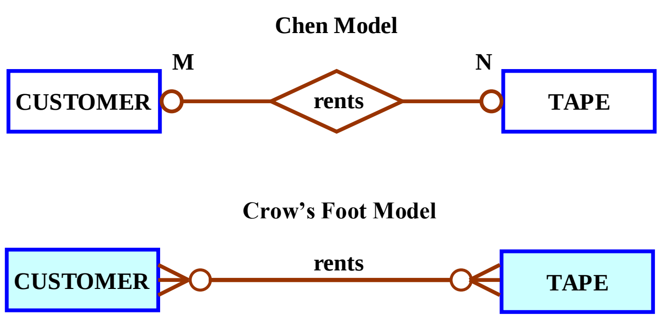
One very important issue that often confuses students is the resolution of M:N relationships into 1:M relationships. We have used Microsoft Visio Professional to create the ERDs in the text and in this manual. Note that ERDs are done at the conceptual level. However, MS Visio has an implementation focus. Therefore, M:N relationships are not directly supported. Instead, the designer is limited to modeling 1:1 and/or 1:M relationships.

Although M:N relationships may properly be viewed in a relational database model at the *conceptual* level, *such relationships should not be implemented*, because their existence creates undesirable redundancies. Therefore, M:N relationships must be decomposed into 1:M relationships to fit into the ER framework. For example, if you were to develop an ER model for a video rental store, you would note that tapes can be rented more than once and that customers can rent more than one tape.

To make the discussion more interesting and to address several design issues at once, explain to the students that it seems reasonable to keep in mind that newly arrived tapes that have just been entered into inventory have not yet been rented and that some tapes may never be rented at all if there is no demand for them. Therefore, CUSTOMER is optional to TAPE. Assuming that the video store only rents videos and that a CUSTOMER entry will not exist unless a person coming into the video store actually rents that first tape, TAPE is mandatory to CUSTOMER. On the other hand, if the store has other services, such as selling movies or games, then a CUSTOMER entry could exist without having rented a video. In which case, a TAPE is optional to CUSTOMER. Note that this discussion includes a very brief description of the video store's operations and some business rules. The relationship between customers and tapes would thus be M:N, as shown in Figure IM4.1.

Figure IM4.1 The M:N Relationship

A customer can rent many tapes.
A tape can be rented by many customers.
Some customers do not rent tapes.
(Such customers might buy tapes or other items.) Therefore, TAPE is optional to CUSTOMER in the rental relationship.
Some tapes are never rented.
Therefore, CUSTOMER is optional to TAPE in the rental relationship.

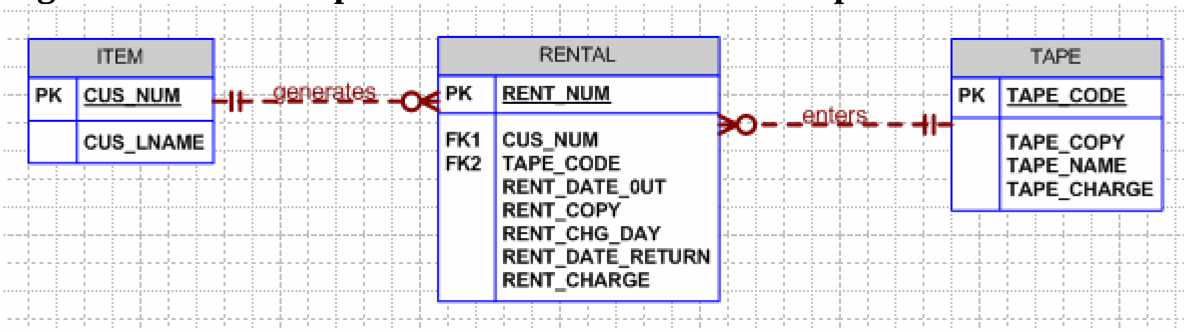


As you discuss the presentation in Figure IM4.1, note that the ERD reflects two business rules:

1. A CUSTOMER may rent many TAPEs.
2. A TAPE can be rented by many CUSTOMERs.

The M:N relationship depicted in Figure IM4.1 must be broken up into two 1:M relationships through the use of a bridge entity, also known as a composite entity. The composite entity, named RENTAL in the example shown in Figure IM4.2, must include *at least* the primary key components (CUS_NUM and TAPE_CODE) of the two entities it bridges, because the RENTAL entity's foreign keys must point to the primary keys of the two entities CUSTOMER and TAPE.

Figure IM4.2 Decomposition of the M:N Relationship



Several points about Figure IM4.2 are worth emphasizing:

- The RENTAL entity's PK could have been the combination TAPE_CODE + CUS_NUM. This composite PK would have created strong relationships between RENTAL and CUSTOMER and between RENTAL and TAPE. Because this composite PK was not used, it is a candidate key.
- In this case, the designer made the decision to use a single-attribute PK rather than a composite PK. Note that the RENTAL entity uses the PK RENT_NUM. It is useful to point out that single-attribute PKs are usually more desirable than composite PKs – especially when relationships must be established between the RENTAL and some – as yet unidentified – entity. (You cannot use a composite PK as a foreign key in a related entity!) In addition, a composite PK makes queries less efficient – a point that will become clearer in Chapter 11, "Database Performance Tuning and Query Optimization."
- Note the placement of the optional symbols. *Because a tape that is never rented will never show up in the RENTAL entity, RENTAL has become optional to TAPE*. That's why the optional symbol has migrated from CUSTOMER to the *opposite* side of RENTAL. Also, note the addition of a few attributes in each of the three entities to make it easier to see what is being tracked.
- Because the M:N relationship has now been decomposed into two 1:M relationships, the ER model shown in Figure IM4.2 is can be implemented. However, it may be useful to remind your students that "implementable" is not necessarily synonymous with "practical" or "useful." (We'll modify the ERD in Figure IM4.2 after some additional discussion.)
- ***Remind the students that the relationships are read from the 1 side to the M side.*** Therefore, the relationships between CUSTOMER and RENTAL and between TAPE and RENTAL are read as

  CUSTOMER generates RENTAL
  TAPE enters RENTAL

  (Compare these relationships to those generated by the two business rules above Figure IM4.1A.)
- The dashed relationship lines indicate weak relationships. In this case, the RENTAL entity's primary key is RENT_NUM – and this PK did not use any attribute from the CUSTOMER and TAPE entities.

The (implied) cardinalities in Figure IM4.2 reflect the rental transactions. Each rental transaction, i.e., each record in the RENTAL table, will reference one and only one customer and one an only one tape. The (simplified!) implementation of this model may thus yield the sample data shown in the database in

Figure IM4.3. The database's relational diagram is shown in Figure IM4.4.

Figure IM4.3 The Ch04_Rental Database Tables

Database name: Ch04_Rental

Table name: CUSTOMER

CUS_NUM	CUS_LNAME
1001	Smith
1002	Cordoza
1003	Bianco
1004	Jones

Table name: TAPE

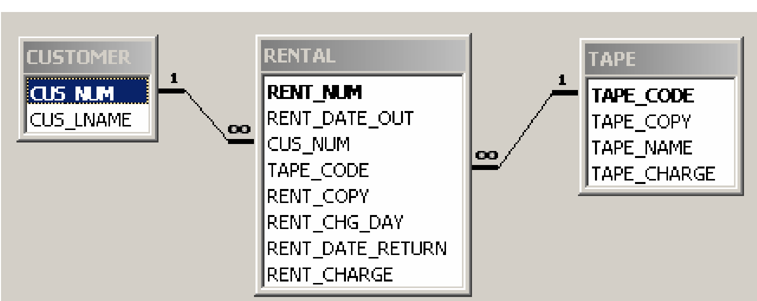
TAPE_CODE	TAPE_COPY	TAPE_NAME	TAPE_CHARGE
M3456-1	1	Ramblin' Tulip	2.50
R2345-1	1	Once Upon a Midnight Breezy	3.99
R2345-2	2	Once Upon a Midnight Breezy	3.99
S4567-1	1	Tulips and Threelips	2.50
W1234-1	1	Bright Stars and Doodle Berries	3.00
W1234-2	2	Bright Stars and Doodle Berries	3.00
W1234-3	3	Bright Stars and Doodle Berries	3.00

Table name: RENTAL

RENT_NUM	RENT_DATE_OUT	CUS_NUM	TAPE_CODE	RENT_COPY	RENT_CHG_DAY	RENT_DATE_RETURN	RENT_CHARGE
10050	10-Jan-2014	1002	R2345-1	1	3.99	11-Jan-2014	3.99
10051	10-Jan-2014	1003	R2345-2	2	3.99	11-Jan-2014	3.99
10052	10-Jan-2014	1001	W1234-2	2	3.00	12-Jan-2014	6.00
10053	10-Jan-2014	1004	M3456-1	1	2.50	11-Jan-2014	2.50
10054	12-Jan-2014	1004	S4567-1	1	2.50	13-Jan-2014	2.50
10055	12-Jan-2014	1001	R2345-1	1	3.99	13-Jan-2014	3.99
10056	13-Jan-2014	1003	W1234-3	3	3.00	13-Jan-2014	3.00
10057	13-Jan-2014	1002	W1234-1	1	3.00	14-Jan-2014	3.00
10058	13-Jan-2014	1001	S4567-1	1	2.50	14-Jan-2014	2.50
10059	14-Jan-2014	1004	M3456-1	1	2.50	15-Jan-2014	2.50
10060	14-Jan-2014	1002	R2345-2	2	3.99	15-Jan-2014	3.99
10061	14-Jan-2014	1003	W1234-3	3	3.00	16-Jan-2014	6.00
10062	15-Jan-2014	1001	S4567-1	1	2.50		
10063	15-Jan-2014	1004	R2345-1	1	3.99	16-Jan-2014	3.99
10064	15-Jan-2014	1004	R2345-2	2	3.99		

The relational diagram that corresponds to the design in Figure IM4.2 is shown in Figure IM 4.4.

Figure IM4.4 The Ch04_Rental Database Relational Diagram



The **Ch04_Rental** database's TAPE and RENTAL tables contain some attributes that merit additional discussion.

- The TAPE_CODE attribute values include a "trailer" after the dash. For example, note that the third record in the TAPE table has a PK value of R2345-2. The "trailer" indicates the tape copy. For example, the "-2" trailer in the PK value R2345-2 indicates the second copy of the :Once Upon a Midnight Breezy" tape. So why include a separate TAPE_COPY attribute? This decision was made to make it easier to generate queries that make use of the tape copy value. (It's much more difficult to use a right-string function to "strip" the tape copy value than simply using the TAPE_COPY value. And "simple" usually translates into "fast" in a query environment – "fast" is a good thing!
- The RENTAL table uses two dates: RENT_DATE_OUT and RENT_DATE_RETURN. This decision leaves the RENT_DATE_RETURN value null until the tape is returned. Naturally, such nulls can be avoided by creating an additional table in which the return date is not a named attribute. Note the following few check-in and check-out transactions:

RENT_NUM	TRANS_TYPE	TRANS_DATE
10050	Checked-out	10-Jan-2014
10050	Returned	11-Jan-2014
10051	Checked-out	10-Jan-2014
10051	Returned	11-Jan-2014
10052	Checked-out	11-Jan-2014
10053	Returned	10-Jan-2014
…..	……………	……………
…..	……………	……………

The decision to leave the RENT_DATE_RETURN date in the RENTAL table – and leaving its value null until the tape is returned is –again – up to the designer, who evaluates the design according to often competing goals: simplicity, elegance, reporting capability, query speed, index management, and so on. (Remind your students that they, too, should be able to evaluate such decisions as they gain database design knowledge.)