Chapter 5

Data Modeling with the Entity-Relationship Model

ANSWERS TO REVIEW QUESTIONS

5.1 *Describe the two phases in designing databases that arise from the development of new information systems.*

When developing new information systems, we first create a data model and then transform that data model into a database design.

5.2 *In general terms, explain how a data model could be used to design a database for a small video rental store.*

The two steps of designing a database are:

(1) creating a data model, in which we work out the complexities of the database design, and

(2) transforming the data model into a database design by adding database design features (these features, such as foreign keys, intersection tables for N:M relationships, etc., will be discussed in detail in Chapter 6).

For a small video rental store, we would analyze the business requirements for the store to determine entities (such as EMPLOYEE, CUSTOMER, RENTAL, RENTAL_ITEM, ITEM) and the relationships between them. We would then create a database model. We would validate the model with the users at the store, and modify it until it met the users' needs and perceptions of how the store business should operate.

Once the database model was complete, we would transform it into a database design (described in Chapter 6), and finally create the database itself in the DBMS (Chapter 7).

5.3 Explain how a data model is like a building blueprint. What is the advantage of making changes during the data modeling stage?

Before a building is actually constructed, it is carefully planned and designed. That work is documented in the **building blueprint**. Similarly, before a database is actually created in a DBMS, it needs to be carefully planned and designed. The work of planning and designing a database is documented in a **data model**.

The advantage of making changes during the data modeling stage is that it is easier, simpler, faster, and cheaper to make changes at that stage of database development.

5.4 Who is the author of the entity-relationship data model?

Peter P. Chen, in his paper "The Entity-Relationship Model—Towards a Unified View of Data," ACM Transactions on Database Systems, January 1976, pp. 9–36. For information on Peter Chen see http://en.wikipedia.org/wiki/Peter_Chen, and for a copy of the article see http://osm.cs.byu.edu/CS452/supplements/chen.pdf .

*5.5 Define **entity**. Give an example of an entity (other than one presented in this chapter).*

An entity is something that the users want to track, and is readily identifiable in their environment. We'll use the example of a Real Estate Agency. Example entities are AGENT John Smith, PROPERTY 568 12th Street, and CASH_RECEIPT CR2004001.

5.6 Explain the difference between an entity class and an entity instance.

An entity class is a collection of entities and is described by the structure or format of the entities in that class. An entity instance of an entity class is the representation of a particular entity, such as AGENT John Smith; it is described by the values of attributes of the entity. There are usually many instances of an entity in an entity class.

*5.7 Define **attribute**. Give an example attribute for the entity in your answer to Review Question 5.5.*

Attributes describe the entity's characteristics. For example, in the Real Estate Agency example in question 5.5, attributes for the entity AGENT are FirstName, LastName, DateOfHire, and OfficePhoneNumber.

*5.8 Define **identifier**. Give an example identifier for the entity in your answer to Review Question 5.5.*

Identifiers are attributes that name, specify, locate (or otherwise identify) entity instances. For example, in the Real Estate Agency example in question 5.5, an identifier for the entity AGENT could be AgentID.

5.9 Give an example of a composite identifier.

Identifiers that consist of two or more attributes are called composite identifiers. Examples are {AreaCode, LocalNumber}, {ProjectName, TaskName}, and {City, State}.

*5.10 Define **relationship**. Give an example of a relationship (other than one presented in this chapter). Name your relationship.*

A relationship is an association between two or more entity classes. For example, assume you have an entity class named Student and an entity class named Class. Students enroll in a Class so you would have a relationship named Enrolls In.

Often, a name consists of a verb or verb phrase expressed from the standpoint of the parent in the relationship, followed by a slash, and followed by the verb phrase expressed from the standpoint of the child. Normally, the verb phrase from the child's view is the passive form of the verb phrase from the parent's view.

5.11 Explain the difference between a relationship class and a relationship instance.

Relationship classes are associations among entity classes, and relationship instances are associations among entity instances.

5.12 What is the degree of a relationship? Give an example of a relationship of degree three (other than one presented in this chapter).

The number of entity classes in the relationship is the **degree** of the relationship. For example, in the Real Estate Agency example in question 5.5, there is a relationship of degree three between AGENT, CLIENT and PROPERTY. In this case we are documenting the PROPERTIES that AGENTS showed to their CLIENTS.

5.13 What is a binary relationship?

A relationship between two entity classes.

5.14 Explain the difference between an entity and a table. Why is this difference important?

Formally, an entity is a database design concept while a table is the implementation of that entity in an actual database. However, the main difference is that relationships between entities can be created without specifying the formal mechanism—foreign keys—for implementing that relationship. With tables in a database, the foreign keys must be created to implement the relationship. This is important because it makes it easier to work with entities in a less formal way, which makes database designs easier to create and change as necessary during the design process.

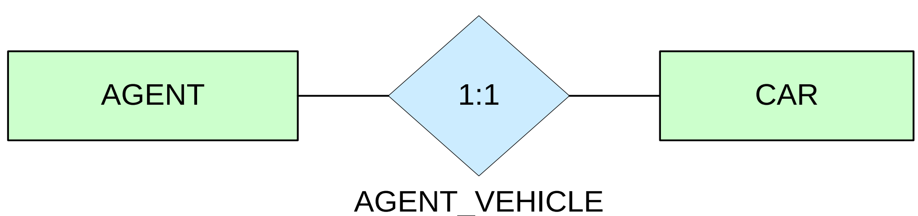
5.15 What does cardinality mean?

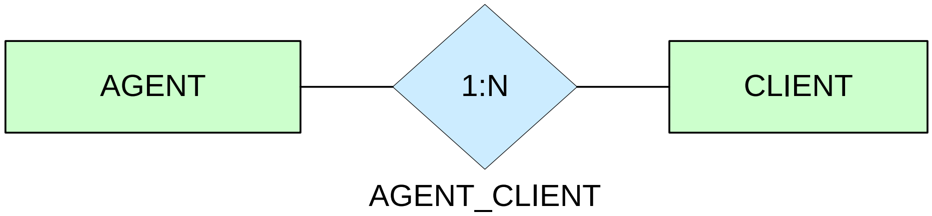
Cardinality means "count." For a relationship, it refers to the number of entity instances in one entity class that are related to entity instances in another class.

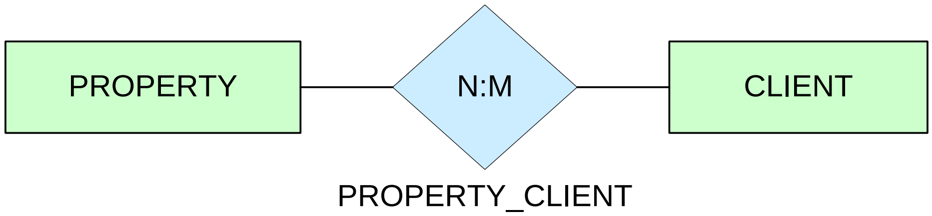
5.16 Define the terms maximum cardinality and minimum cardinality.

Maximum cardinality is the maximum or largest number of entities that can occur on one side of the relationship. **Minimum cardinality** is the minimum or smallest number of entities that must participate in the relationship.

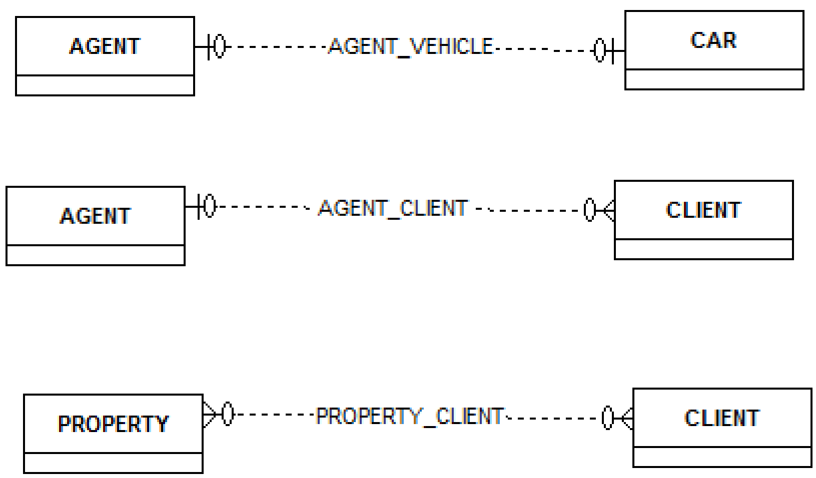
5.17 Give examples of 1:1, 1:N, and N:M relationships (other than those presented in this chapter). Draw two E-R diagrams for each of your examples: one using the traditional diamond notation and one using IE Crow's Foot notation.





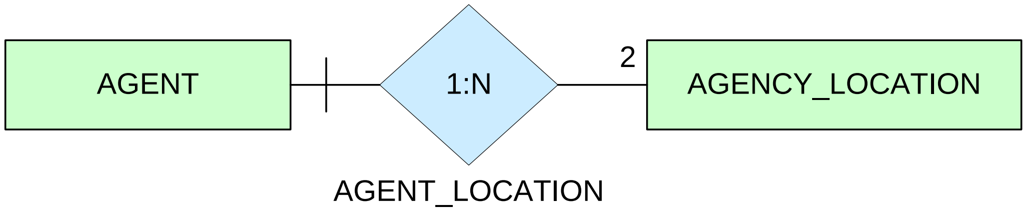


In IE Crow's Foot notation:



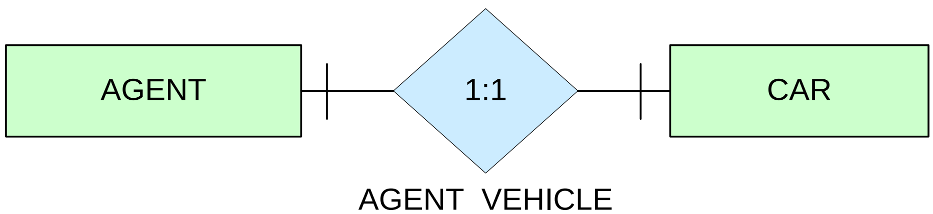
5.18 Give an example for which the maximum cardinality must be an exact number (other than those presented in this chapter).

In the Real Estate Agency example in question 5.5, each AGENT is required to work out of two different AGENCY_LOCATIONs each week. The AGENT always works out of the same two AGENCY_LOCATIONs, so the relationship has an exact maximum cardinality of 2 on AGENCY_LOCATION.

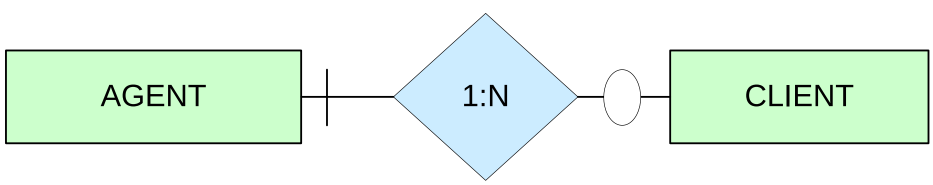


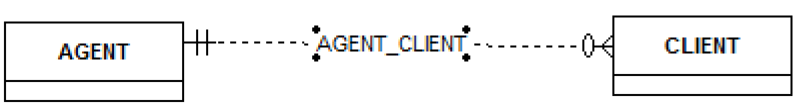
5.19 *Give examples of M-M, M-O, O-M, and O-O relationships (other than those presented in this chapter). Draw two E-R diagrams for each of your examples: one using the traditional diamond notation and one using IE Crow's Foot notation.*

In the Real Estate Agency example in question 5.5, each AGENT must use an agency car when on agency business. Further, to keep costs down the agency keeps exactly enough cars for the agents. Therefore, each AGENT must have a CAR, and each CAR must be assigned to an AGENT. This is an M-M relationship.

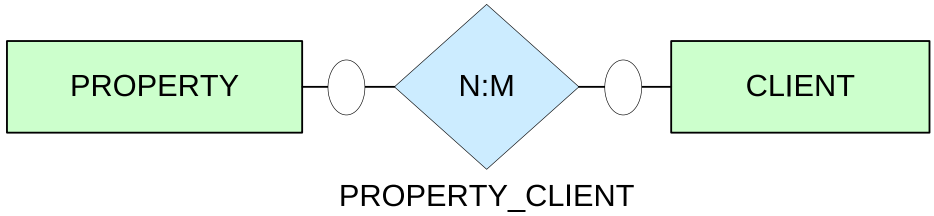


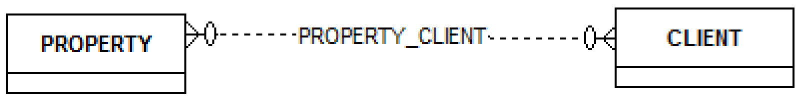
In the Real Estate Agency example in question 5.5, each CLIENT must be assigned to an AGENT, but there may be AGENTs who currently have no CLIENTs. This is an M-O (same as O-M, but seen reversed) relationship.





In the Real Estate Agency example in question 5.5, CLIENTs may be seeing AGENTs without currently being interested in specific PROPERTY, and a PROPERTY may be listed without any CLIENT currently being interested in it. This is an O-O relationship.





5.20 Explain, in general terms, how the traditional E-R model, the IE Crow's Foot version, the IDEF1X version, and the UML version differ. Which version is used primarily in this text?

The traditional E-R model uses the notation shown in RQ 5-17 and 5-19—rectangles for entities, diamonds for relationships, and connected ellipses (the ellipses, which are used for attributes, are not shown in any illustration in the text or in this IM chapter—they are not to be confused with the circles [sometimes drawn as ellipses] used to indicate optional participation) for attributes.

The Information Engineering (IE) model is a variant of the traditional E-R that does not use diamonds, but instead uses connecting lines between entities with specific symbols at the end of the lines. The symbol for "many" as in a "one-to-many: relationship" resembles a crow's foot, and that fact gave the model its nickname.

The IDEF1X model is a national government standard model. Based on the traditional E-R model, it again drops the use of diamonds for relationships and substitutes connecting lines between entities with specific symbols at the ends of the lines. However, these symbols are quite different from the symbols used by the Information Engineering model.

Finally, Unified Modeling Language (UML) again dropped the diamonds (and ellipses) and added connecting lines with its own set of symbols. However, the UML model also added in notation for recording object-oriented programming (OOP) concepts.

The IE Crow's Foot model is used primarily in this text.

5.21 Explain how the notations shown in Figure 5-7 differ.

The notations in Figure 5-7 show the difference between the original E-R model and the IE Crow's Foot notation used in the text for a 1:N O-M relationship.

5.22 Explain how the notations shown in Figure 5-9 differ.

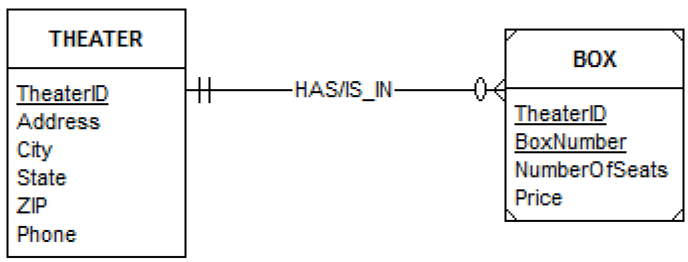
The notations in Figure 5-9 show the difference between the original E-R model and the IE Crow's Foot notation used in the text for an N:M O-M relationship.

5.23 What is an ID-dependent entity? Give an example of an ID-dependent entity (other than one presented in this chapter).

An ID-dependent entity is one in which the identifier of one entity includes the identifier of another entity.

In the Real Estate Agency example in question 5.5, we would find that some PROPERTYs are apartment houses, and each APARTMENT has its own identifying number. This number by itself is meaningless (101, 102, etc.) and must be combined with the identifier of the PROPERTY to truly identify each APARTMENT. Alas, this is too similar to the BUILDING – APARTMENT example in the text.

So, another example—a THEATER has BOXes, which are special seating sections. The number of each BOX is meaningless— "Here's your ticket to Box 303" —without knowing which THEATER the BOX is in. Note that not all THEATERs have BOXes, therefore the relationship is M:O.



5.24 Explain how to determine the minimum cardinality of both sides of an ID-dependent relationship.

The ID-dependent entity (the "child") cannot exist without the entity upon which it is dependent (the "parent"). Therefore, the minimum cardinality from the ID-dependent entity to the parent is always one (1).

On the other hand, a parent entity may be able to exist without any children. For example, not all PROPERTYs have APARTMENTs (or UNITs), and not all THEATERs have BOXes. Therefore the minimum cardinality from the parent to the ID-dependent entity depends upon database application requirements.

5.25 What rules exist when creating an instance of an ID-dependent entity? What rules exist when deleting the parent of an ID-dependent entity?

In order to create an instance of an ID-dependent entity, the parent entity upon which it depends must have already been created. If the parent of an ID-dependent entity is deleted, all associated instances of the ID-dependent entity must be deleted as well.

5.26 What is an identifying relationship? How is it used?

An identifying relationship is a special type of relationship. It is used to represent ID-dependence. Most data modeling products use a solid line to represent an identifying relationship and a dashed line to represent a nonidentifying relationship.

*5.27 Explain why the relationship between **BUILDING** and **APARTMENT** discussed on page 206 is an identifying relationship.*

The relationship between BUILDING and APARTMENT is an identifying relationship because APARTMENT is ID-dependent on BUILDING.

5.28 What is a weak entity? How do weak entities relate to ID-dependent entities?

A **weak entity** is an entity whose existence depends upon the existence of another entity. All ID-dependent entities are weak entities, but not all weak entities are ID-dependent.

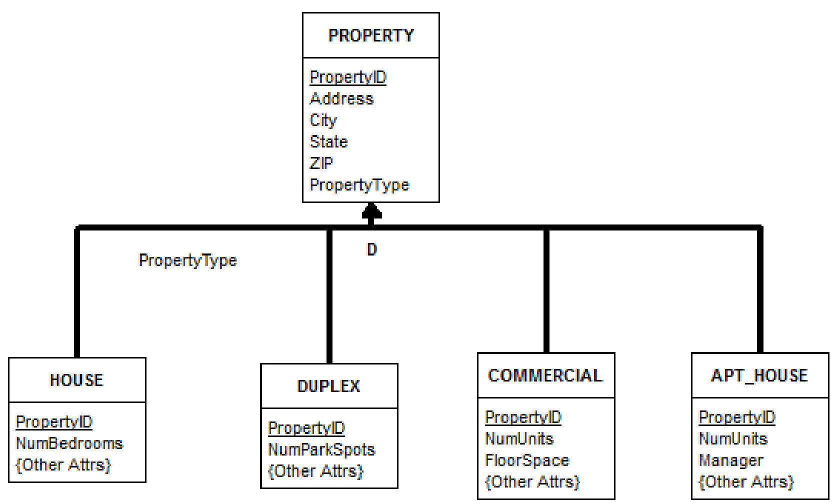
5.29 What distinguishes a weak entity from a strong entity that has a required relationship to another entity?

A strong entity that has a required relationship with another entity can and will exist outside the database without the presence of the other, strong entity. A weak entity cannot and does not exist without the presence of the other, strong entity.

*5.30 Define **subtype** and **supertype**. Give an example of a subtype–supertype relationship (other than one presented in this chapter).*

A supertype is an entity class that contains a set of attributes common to what would otherwise be modeled as several entity classes. The subtypes (there can be more than one) are entity classes that contain the specialized attributes not common to the other subtype entities.

In the Real Estate Agency example from question 5.5, we would have PROPERTY as a supertype, and HOUSE, DUPLEX, APARTMENT_HOUSE, and COMMERCIAL as subtypes. This would be shown in ER-Assistant as:



5.31 Explain the difference between exclusive subtypes and inclusive subtypes. Give an example of each.

A group of subtypes may be considered as either a set of exclusive subtypes or inclusive subtypes. In a group of exclusive subtypes, a supertype entity instance is associated with at most one subtype entity instance. An example of this is the Real Estate Agency example shown in the answer to review question 5.30 above. In a group of inclusive subtypes, the supertype entity instance can be associated with one or more of the subtypes. An example of this for the Real Estate Agency database is that a CLIENT may be included in more than one of the subtype sets—HOME_BUYER, RENTER, or COMMERCIAL_BUYER.

5.32 What is a discriminator?

A discriminator is an attribute of the supertype entity that identifies the associated subtype entity. An example of this is the Real Estate Agency example shown in the answer to review question 5.30 above; PropertyType is the discriminator.

5.33 Explain the difference between IS-A and HAS-A relationships.

The relationship between a supertype and its subtypes is sometimes called an IS-A relationship. Entities with an IS-A relationship should have the same identifier because they represent different aspects of the same thing. Entities with HAS-A relationships represent aspects of different things and thus have different identifiers. These relationships do not involve subtypes.

5.34 What is the most important reason for using subtypes in a data model?

The most important reason for using subtypes in a data model is to avoid value-inappropriate null values. In the Real Estate Agency example shown in the answer to review question 5.30 above, HOUSEs do not have TotalFloorSpace or NumberOfUnits, and COMMERCIAL does not have NumberOfBedrooms. If all the attributes in the subtypes appeared in the supertype, there would be null values in such columns.

5.35 Describe the relationship between the structure of forms and reports and the data model.

The structure of forms and reports determines the structure of the data model. The reverse is also true, for the structure of the data model will determine the structure of the forms and reports that can be based on it.

5.36 Explain two ways forms and reports are used for data modeling.

Forms and reports are used to:

(1) Determine the structure of the data model, and

(2) Validate the data model.

5.37 Explain why the form and report in Figure 5-15 indicate that the underlying relationship is 1:1.

The form shows that each MEMBER is associated with just one LOCKER. The report shows that each LOCKER is associated with just one MEMBER. This indicates a 1:1 relationship.

5.38 Why is it not possible to infer minimum cardinality from the form and report in Figure 5-15?

The form and report in Figure 5-15 show specific instances, but not the full range of possible relationship values between MEMBER and LOCKER. For example, unassigned LOCKERs may not show up on the report.

5.39 Describe two tests for determining if an entity is a strong entity.

The two tests are:

(1) Does the entity have an identifier of its own?

(2) Does the entity seem logically different and separate from other entities?

5.40 Why does the form in Figure 5-17 not indicate that the underlying relationship is 1:N? What additional information is required to make that assertion?

The form in Figure 5-17 only shows that the relationship from MEMBER to UNIFORM is 1:N. There is no form or report to show whether the relationship from UNIFORM to MEMBER is 1:1 or 1:N. If it is 1:1, the overall relationship between MEMBER and UNIFORM is 1:N, but if it is 1:N, then the overall relationship between MEMBER and UNIFORM is N:M.

We need to (directly) ask the users or (indirectly) consider the typical structure clubs to find out if one UNIFORM can be associated with more than one MEMBER.

5.41 Explain why two forms or reports are usually needed to infer maximum cardinality.

Each form or report only shows the maximum cardinality in one direction between the entities. Therefore, to know the cardinalities in both directions requires two forms or reports.

5.42 How can you assess minimum cardinality for the entities in the form in Figure 5-17?

You **cannot** assess minimum cardinality from the form in Figure 5-17. In general, you cannot determine minimum cardinality from forms and reports.

5.43 Explain why the form and report in Figure 5-19 indicate that the underlying relationship is N:M.

In Figure 5-19, the form shows that the relationship from SUPPLIER to PART is 1:N, while the report shows that the relationship from PART to SUPPLIER is 1:N. Therefore, the overall relationship is N:M.

5.44 Name three patterns that use ID-dependent relationships.

Three patterns that use ID-dependent relationships are (1) the **association pattern**, (2) the **multivalued attribute pattern**, and (3) the **archetype/instance pattern**.

5.45 *Explain how the association pattern differs from the N:M strong entity pattern. What characteristic of the report in Figure 5-21 indicates that an association pattern is needed?*

The association pattern differs from the N:M strong entity pattern in that a new, third entity is added to hold additional attributes not associated with the original two entities. In the report in Figure 5-21, the Price column is the indicator that an association pattern is needed because Price is an attribute of neither COMPANY nor PART.

5.46 *In general terms, explain how to differentiate an N:M strong entity pattern from an association pattern.*

In general, if there are one or more additional attributes associated with the relationship between two strong entities in an otherwise N:M strong entity pattern, then an association pattern is needed. In the data model, this will be shown as a third, weak entity that is ID-dependent on both of the other entities.

5.47 *Explain why two entities are needed to model multivalued attributes.*

In the E-R model, all attributes must have a single value. Therefore, multivalued attributes must be modeled with a second table to hold the multiple values of the attribute.

NOTE: This is the multivalued dependency problem from Chapter 4. The additional entity or table is needed to put the relations into 4NF.

5.48 *How do the forms in Figures 5-26 and 5-28 differ? How does this difference affect the data model?*

In Figure 5-26, CONTACT and PHONE are independent—any phone number can be used to reach any contact. In Figure 5-28, CONTACT and PHONE are dependent—one phone number is used to reach one specific contact. This difference determines whether the data model will be created with (1) two entities—one for CONTACT and one for PHONE, or (2) one entity—PHONE_CONTACT that holds both CONTACT and the associated PHONE number.

5.49 *Describe, in general terms, the archetype/instance pattern. Why is an ID-dependent relationship needed for this pattern? Use the CLASS/SECTION example shown in Figure 5-30 in your answer.*

The archetype/instance pattern generally has one entity that is a manifestation (or "instance") of another (logical abstraction or "archetype") entity. The archetype is usually an abstract concept that is actually seen in the real world as the instance. The CLASS and SECTION example illustrates this. In reality, CLASS is an abstract concept—the conceptualization of what the class is about, what texts will be used, and in what order the topics will be presented. However, the only time that a CLASS is seen in the real world is as a SECTION of the CLASS. The SECTION has an instructor and students, real text books, and real lectures and exams. However, since a SECTION is based on the conceptual CLASS, it cannot exist apart from that CLASS, and is therefore ID-dependent.

5.50 *Explain what caused the entities in Figure 5-31 to change from ID-dependent entities.*

The entities in Figure 5-31 changed from ID-dependent entities to simply weak entities when a separate identifier was added. The entities now no longer have to be dependent on the identifier of the parent entity in order to be recognized.

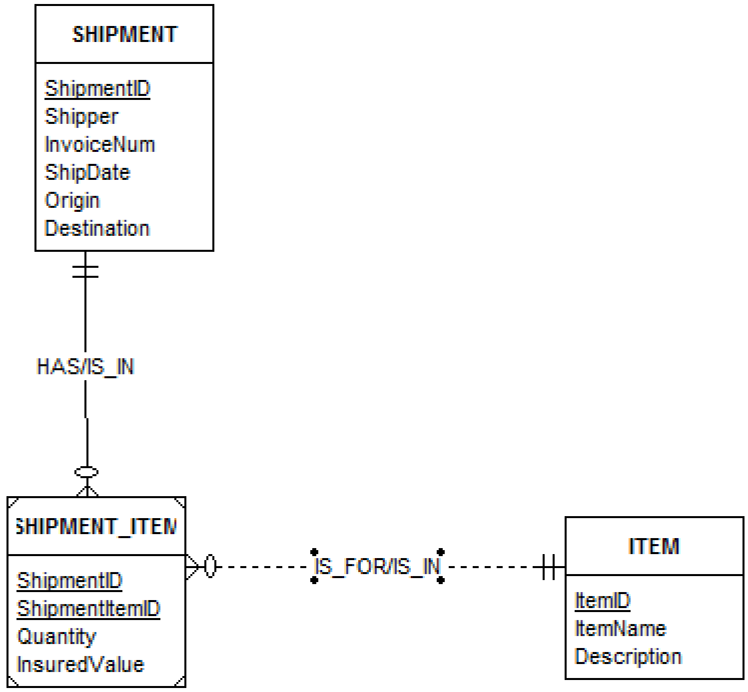
5.51 *Summarize the two sides in the argument about the importance of weak but not ID-dependent entities.*

Side one – Weak, non-ID-dependent entities are NOT important: Although they exist, they are not necessary. The same result can be achieved by making a strong entity a child entity with a required relationship to the parent entity.

Side two – Weak, non-ID-dependent entities ARE important: Weak entities exist based on a logical necessity that reflects reality. This exists regardless of a business rule that would require a strong entity as the child entity with a required relationship to the parent entity.

5.52 *Give an example of the line-item pattern as it could be used to describe the contents of a shipment. Assume that the shipment includes the names and quantities of various items as well as each item's insured value. Place the insurance value per item in an ITEM entity.*

This question actually looks ahead to the Morgan Importing project, where exactly such a situation occurs, and is a variant of it. Here is the E-R diagram:

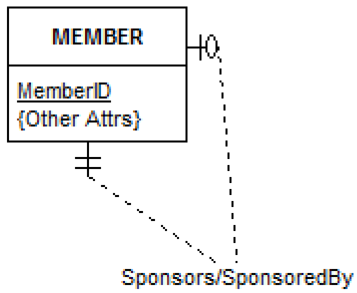


5.53 *What entity type should come to mind when you see the words "For use by" in a form?*

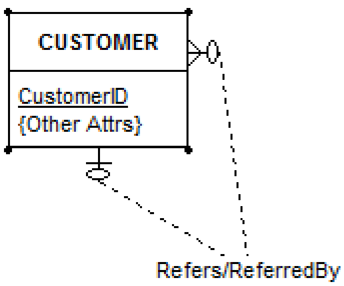
The words "For use by" indicate a supertype/subtype entity pattern.

5.54 Give examples of 1:1, 1:N, and N:M recursive relationships (other than those presented in this chapter).

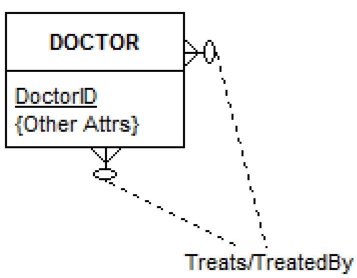
1:1 Recursive: To be a member of the New City Club, you must have a sponsor, who can sponsor at most one other member. Therefore each member is associated with exactly one sponsor, who is also a member with a sponsor.



1:N Recursive: Customers of the All Stuff Warehouse may refer one or more other customers to the business. Therefore, there are many referred customers with exactly one referring customer, who may also have been referred by another customer. However, customers may not have been referred, and customers do not have to make referrals.



N:M Recursive: In the medical community in New Town, doctors treat each other. Given the number of medical specialties, it is no wonder that any one doctor is treated by many other doctors, while at the same time that doctor has treated many other doctors him- or herself. However, a doctor does not have to be treated by another doctor in the community, nor does the doctor have to be treating another community doctor.



5.55 Explain why the data modeling process must be iterative. Use the Highline University example.

The data modeling process must be iterative because each new step may reveal new entities that need to be added to the model, new attributes that should be added to existing entities, existing attributes that should be moved from one entity to another, and other needed changes.

Two examples (of several) in the Highline University example are:

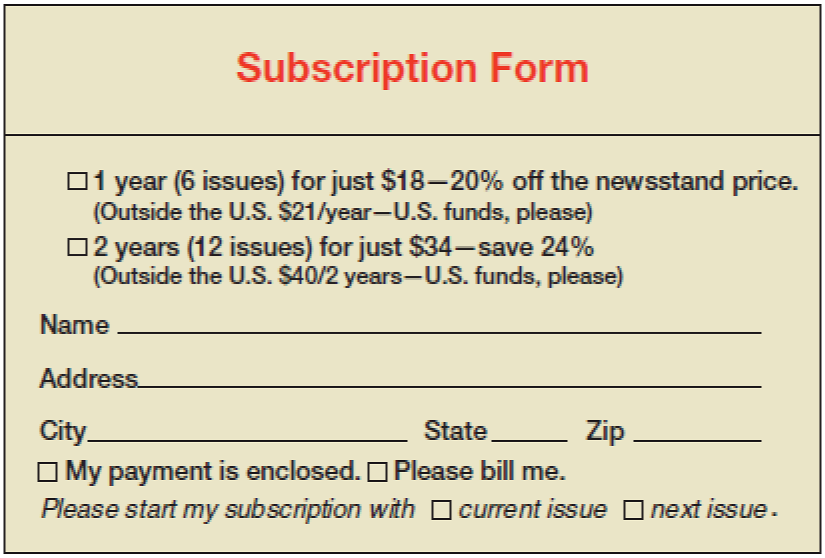
(1) The Department Report in Figure 5-46 revealed a Campus Address consisting of Building and RoomNumber that needed to be added to the existing DEPARTMENT entity.

(2) The assumed discovery of a PROFESSORs appointment titles and term resulted in the conversion of an N:M relationship between DEPARTMENT and PROFESSOR into an association pattern using the new entity APPOINTMENT.

❖ ANSWERS TO PROJECT QUESTIONS

Answer the following questions using IE Crow's Foot notation.

5.56 Examine the subscription form shown in Figure 5-53. Using the structure of this form, do the following:



Subscription Form

☐ 1 year (6 issues) for just $18—20% off the newsstand price.
(Outside the U.S. $21/year—U.S. funds, please)
☐ 2 years (12 issues) for just $34—save 24%
(Outside the U.S. $40/2 years—U.S. funds, please)

Name ______________________________

Address ______________________________

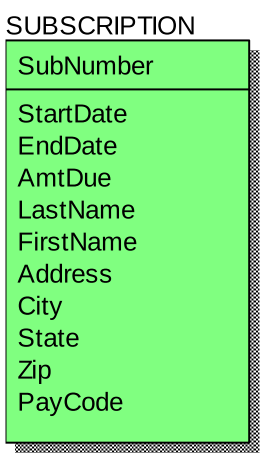
City ____________ State ______ Zip ________

☐ My payment is enclosed. ☐ Please bill me.

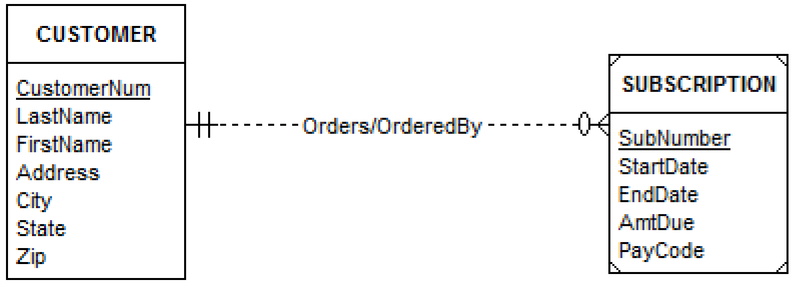
Please start my subscription with ☐ *current issue* ☐ *next issue.*

Figure 5-53 – Subscription Form

A Create a model with one entity. Specify the identifier and attributes.



B Create a model with two entities, one for customer and a second for subscription. Specify identifiers, attributes, relationship name, type, and cardinalities.



Note that subscription is a weak (indicated by the corners) but NOT ID-dependent (indicated by the dotted line) entity. The E-R Crow's Foot model above is based on the following data:

RELATIONSHIP			CARDINALITY	
PARENT	**CHILD**	**TYPE**	**MAX**	**MIN**
CUSTOMER	SUBSCRIPTION	Strong	1:N	M-O

C Under what conditions do you prefer the model in A to that in B?

Model A would be the best model if a Customer can only have one subscription.

D Under what conditions do you prefer the model in B to that in A?

Model B would be the best model if a Customer can have one or more (i.e., multiple) subscriptions.

5.57 *Examine the list of e-mail messages in Figure 5-54. Using the structure and example data items in this list, do the following:*

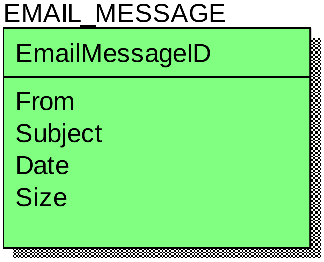
From	Subject	Date	Size
SailorBob256@somewhere.com	Big Wind	5/13/2015	3 KB
SailorBob256@somewhere.com	Update	5/12/2015	4 KB
SailorBob256@somewhere.com	Re: Saturday Am	5/11/2015	4 KB
SailorBob256@somewhere.com	Re: Weather window!	5/10/2015	4 KB
SailorBob256@somewhere.com	Re: Howdy!	5/10/2015	3 KB
SailorBob256@somewhere.com	Still here	5/9/2015	3 KB
SailorBob256@somewhere.com	Re: Turle Bay	5/8/2015	4 KB
SailorBob256@somewhere.com	Turle Bay	5/8/2015	4 KB
SailorBob256@somewhere.com	Re: Hi	5/6/2015	3 KB
SailorBob256@somewhere.com	Sunday, Santa Maria	5/5/2015	3 KB
MotorboatBobby314@elsewhere.com	Cabo, Thurs. Noon	5/2/2015	2 KB
SailorBob256@somewhere.com	turbo	5/1/2015	3 KB
SailorBob256@somewhere.com	on our way	4/28/2015	3 KB
TugboatAmanda756@anotherwhere.com	RE: Hola!	4/26/2015	3 KB
TugboatAmanda756@anotherwhere.com	RE: Hola!	4/24/2015	2 KB
TugboatAmanda756@anotherwhere.com	RE: Hola!	4/23/2015	3 KB

Figure 5-54 – E-mail List

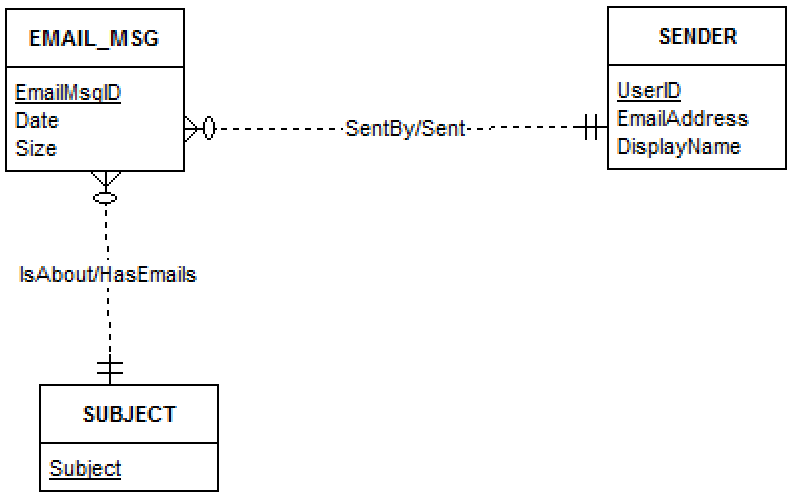
A *Create a single-entity data model for this list. Specify the identifier and attributes.*

No attribute is unique, even a combination of all the shown attributes is not necessarily unique. Therefore, a surrogate identifier of EmailMessageID was created.

EMAIL_MESSAGE



B Modify your answer to A to include entities SENDER and SUBJECT. Specify the identifiers and attributes of entities and the types and cardinalities of the relationships. Explain which cardinalities can be inferred from Figure 5-54 and which need to be checked out with users.



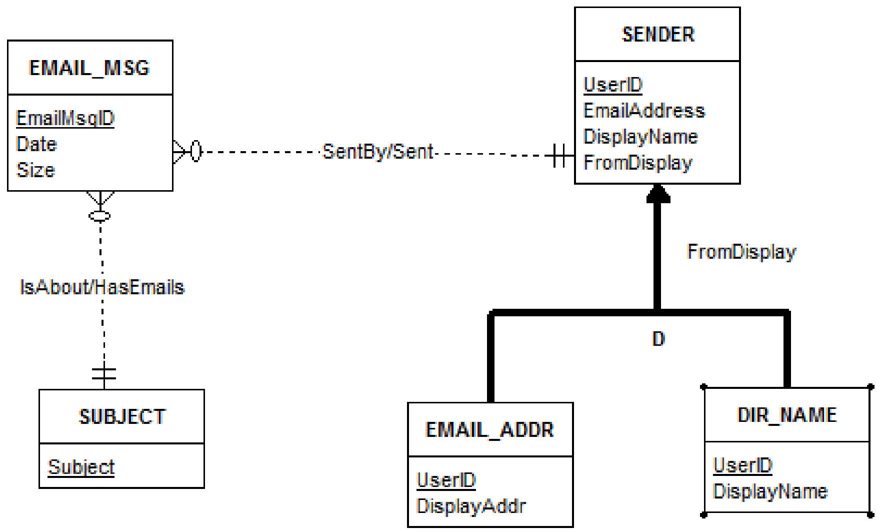
The E-R Crow's Foot model above is based on the following data:

RELATIONSHIP			CARDINALITY [Blue = Inferable]	
PARENT	**CHILD**	**TYPE**	**MAX**	**MIN**
SENDER	EMAIL_MESSAGE	Strong	1:N	M-O
SUBJECT	EMAIL_MESSAGE	Strong	1:N	M-O

We can infer that:

- the one-to-many relationship between EMAIL_MESSAGE and SUBJECT can be implied because the subject repeats. For example, there are three messages with the Subject RE:Hotel.
- the one-to-many relationship between EMAIL_MESSAGE and SENDER can also be implied because the sender repeats. For example, there are three messages from Tom Cooper.

C *The e-mail address in the From column in Figure 5-54 is in two different styles. One style has the true e-mail address; the second style (e.g., Tom Cooper) is the name of an entry in the user's e-mail directory. Create two categories of SENDER based on these two styles. Specify identifiers and attributes.*



The E-R Crow's Foot model is identical to the design in part B except for the addition of two subtypes EMAIL_ADDRESS and DIRECTORY_NAME to the entity SENDER. An additional attribute, FromDisplay, is added to SENDER as a discriminator.

The information about the supertype/subtype relationships is in the following table:

RELATIONSHIP			CARDINALITY [Blue = Inferable]	
PARENT	**CHILD**	**TYPE**	**MAX**	**MIN**
SENDER	EMAIL_ADDRESS	Subtype	1:N	M-O
SENDER	DIRECTORY_NAME	Subtype	1:N	M-O

We can infer that:

- the one-to-many relationships between (SENDER and EMAIL_ADDRESS) and (SENDER and DISPLAY_NAME) are implied because of the supertype/subtype relationship.
- the M-O relationships between (SENDER and EMAIL_ADDRESS) and (SENDER and DISPLAY_NAME) are implied because of the supertype/subtype relationship.

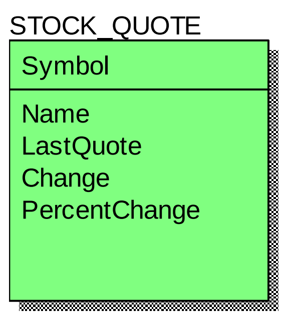
We should ask the users whether subject lines are required for all messages.

5.58 *Examine the list of stock quotes in Figure 5-55. Using the structure and example data items in this list, do the following:*

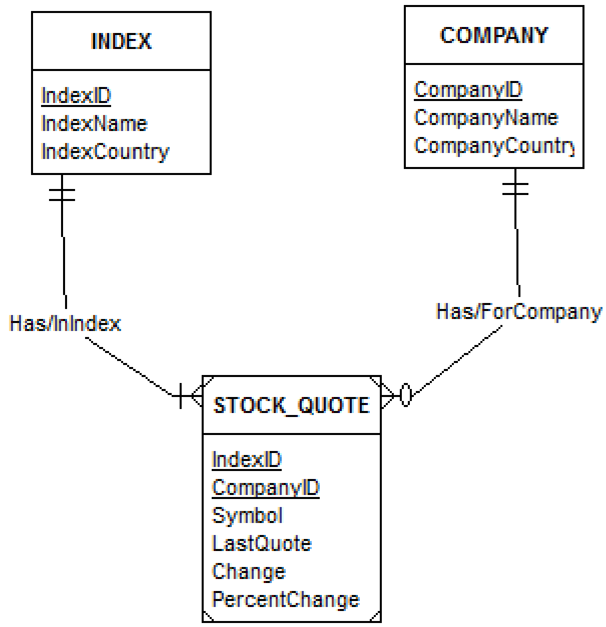
Symbol	Name	Last	Change	% Chg
$COMPX	Nasdaq Combined Composite Index	1,400.74 ▼	-4.87	-0.35%
$INDU	Dow Jones Industrial Average Index	9,255.10 ▼	-19.80	-0.21%
$INX	S&P 500 INDEX	971.14 ▼	-5.84	-0.60%
ALTR	Altera Corporation	13.45 ▼	-0.450	-3.24%
AMZN	Amazon.com, Inc.	15.62 ▲	+0.680	+4.55%
CSCO	Cisco Systems, Inc.	13.39 ▼	-0.280	-2.05%
DELL	Dell Computer Corporation	24.58 ▼	-0.170	-0.69%
ENGCX	Enterprise Growth C	14.60 ▼	-0.210	-1.42%
INTC	Intel Corporation	18.12 ▼	-0.380	-2.05%
JNJ	Johnson & Johnson	53.29 ▼	-0.290	-0.54%
KO	Coca-Cola Company	56.70 ▼	-0.580	-1.01%
MSFT	Microsoft Corporation	53.96 ▲	+1.040	+1.97%
NKE	NIKE, Inc.	57.34 ▲	+0.580	+1.02%

Figure 5-55 – Stock Quotations

A Create a single-entity data model for this list. Specify the identifier and attributes.



B *Modify your answer to A to include the entities COMPANY and INDEX. Specify the identifier and attributes of the entities and the types and cardinalities of the relationships. Explain which cardinalities can be inferred from Figure 5-55 and which need to be checked out with users.*



The E-R Crow's Foot model above is based on the following data:

RELATIONSHIP			CARDINALITY [Blue = Inferable]	
PARENT	**CHILD**	**TYPE**	**MAX**	**MIN**
INDEX	STOCK_QUOTE	ID-Dependent	1:N	M-M
COMPANY	STOCK_QUOTE	ID-Dependent	1:N	M-O

We can infer that:

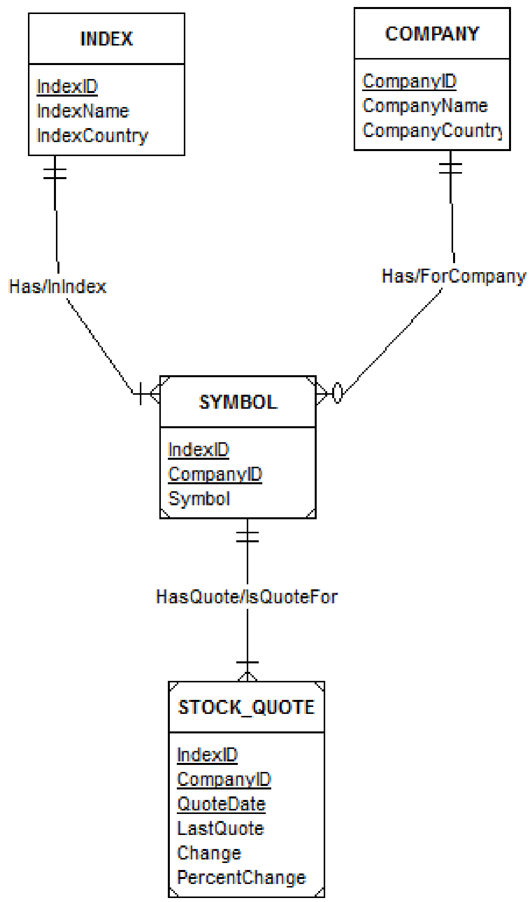
- the one-to-many relationship between INDEX and STOCK_QUOTE can be implied because there are multiple quotes in the single INDEX shown in the figure.

We need to determine if:

- the one-to-many relationship between COMPANY and STOCK_QUOTE is correct. It is only correct if a COMPANY can be listed on multiple INDEXes.

- the M-M relationship between INDEX and STOCK_QUOTE is correct. We have assumed that we are working only with stocks listed on an INDEX.
- the M-O relationship between COMPANY and STOCK_QUOTE is correct. We can infer that if a COMPANY has a STOCK_QUOTE it must be in the COMPANY table, but we have assumed that a COMPANY can be included without us having obtained a STOCK_QUOTE for it yet.

C The list in Figure 5-55 is for a quote on a particular day at a particular time of day. Suppose that the list were changed to show closing daily prices for each of these stocks and that it includes a new column: QuoteDate. Modify your model in B to reflect this change.



The E-R Crow's Foot model above is based on the data in the table on the next page.

Note that both the SYMBOL and STOCK_QUOTE entities are necessary. This is because a symbol now has multiple quotes, so the "LastQuote" information is now multivalued. We could also explain this in term of functional dependencies as follows: This is because (IndexID, CompanyID) → Symbol, and if we had just added QuoteDate to STOCK_QUOTE in question A, we would have had Symbol functionally dependent on part of the composite key (QuoteDate, IndexID, CompanyID). This violates BCNF, and we must break out the (IndexID, CompanyID) → Symbol dependency into its own table.

RELATIONSHIP			CARDINALITY [Blue = Inferable]	
PARENT	**CHILD**	**TYPE**	**MAX**	**MIN**
INDEX	SYMBOL	ID-Dependent	1:N	M-M
COMPANY	SYMBOL	ID-Dependent	1:N	M-O
SYMBOL	STOCK_QUOTE	ID-Dependent	1:N	M-M

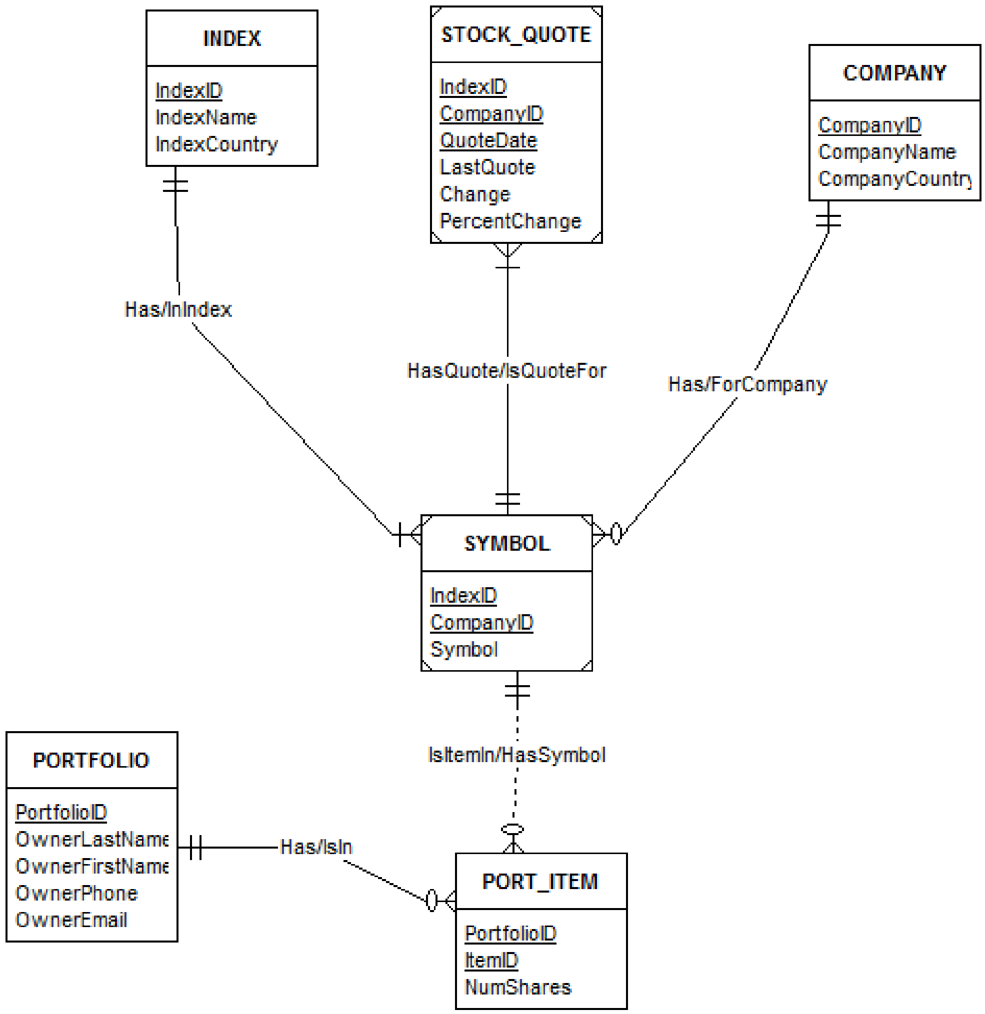
We can infer that:

- the one-to-many relationship between INDEX and STOCK_QUOTE can be implied because there are multiple quotes in the single INDEX shown in the figure.
- the one-to-many relationship between SYMBOL and STOCK_QUOTE can be implied because there are multiple Quote Dates.

We need to determine if:

- the one-to-many relationship between COMPANY and STOCK_QUOTE is correct. It is only correct if a COMPANY can be listed on multiple INDEXes.
- the M-M relationship between INDEX and SYMBOL is correct. We have assumed that we are only recording stocks included in some stock INDEX.
- the M-O relationship between COMPANY and STOCK_QUOTE is correct. We can infer that if a COMPANY has a STOCK_QUOTE it must be in the COMPANY table, but we have assumed that a COMPANY can be included without us having obtained a STOCK_QUOTE for it yet.
- the M-M relationship between SYMBOL and STOCK_QUOTE is correct. We have assumed that we only work with SYMBOLs for stocks for which we want and have obtained a STOCK_QUOTE.

D Change your model in C to include the tracking of a portfolio. Assume the portfolio has an owner name, a phone number, an e-mail address, and a list of stocks held. The list includes the identity of the stock and the number of shares held. Specify all additional entities, their identifiers and attributes, and the type and cardinality of each relationship.



The E-R Crow's Foot model above is based on the data in the table on the next page.

RELATIONSHIP			CARDINALITY [Blue = Inferable]	
PARENT	**CHILD**	**TYPE**	**MAX**	**MIN**
INDEX	SYMBOL	ID-Dependent	1:N	M-M
COMPANY	SYMBOL	ID-Dependent	1:N	M-O
SYMBOL	STOCK_QUOTE	ID-Dependent	1:N	M-M
PORTFOLIO	PORT_ITEM	ID-Dependent	1:N	M-O
SYMBOL	PORT_ITEM	Non-ID-Dependent	1:N	M-O

We can infer that:

- the one-to-many relationship between INDEX and STOCK_QUOTE can be implied because there are multiple quotes in the single INDEX shown in the figure.
- the one-to-many relationship between STOCK_QUOTE and SYMBOL can be implied because there are multiple Quote Dates.
- the one-to-many relationship between PORTFOLIO and PORT_ITEM can be implied because PORTFOLIOs by definition are intended to hold many PORT_ITEMS.
- the one-to-many relationship between SYMBOL and PORT_ITEM can be implied because a single stock can be an item in many different portfolios.

We need to determine if:

- the one-to-many relationship between COMPANY and STOCK_QUOTE is correct. It is only correct if a COMPANY can be listed on multiple INDEXes.
- the M-M relationship between INDEX and SYMBOL is correct. We have assumed that we are only with stocks listed on an INDEX.
- the M-O relationship between COMPANY and STOCK_QUOTE is correct. We can infer that if a COMPANY has a STOCK_QUOTE it must be in the COMPANY table, but we have assumed that a COMPANY can be included without us having obtained a STOCK_QUOTE for it yet.
- the M-M relationship between SYMBOL and STOCK_QUOTE is correct. We have assumed that we only work with SYMBOLs for stocks for which we want and have obtained a STOCK_QUOTE.

- the M-O relationship between PORTFOLIO and PORT_ITEM is correct. We have assumed that a PORTFOLIO can be created before any PORT_ITEMS are added.
- the M-O relationship between SYMBOL and PORT_ITEM is correct. We have assumed that just because we have a SYMBOL and an associated STOCK_QUOTE, this does not mean that we have actually purchased any shares of the stock itself.

E Change your answer to question D to keep track of portfolio stock purchases and sales in a portfolio. Specify entities, their identifiers and attributes, and the type and