Chapter 5: Data Modeling

# Teaching Tips and Strategies *(from Barbara Wixom)*

The way to present this chapter to students really depends on the way in which data management is taught in the curriculum at your school. The programs within which I have taught have all offered a Data Management course that precedes SAD. Given this, I use Chapter 7 as a review chapter and spend two sessions or less on the topic.

I have the students read the chapter as review and then, instead of presenting material lecture-style, I have them do an in-class modeling exercise (many of the chapter’s end of chapter questions or mini-case would serve well). Then, I go over the exercise, keeping an eye out for concepts that seem to have been forgotten from earlier courses. I then do a bit of lecturing on the fly depending on what areas the students seem weakest. After I feel comfortable that the students have the basics down, I will cover normalization as a way of “validating” a data model.

You can actually make this a fun review day by putting students in groups to work on modeling problems that you assign from the end of the chapter. Then ask tough questions that get at some of the more difficult concepts in the chapter to see if students can work out the answers as a group. (For example, see if they understand the difference between cardinality and modality, or the difference between identifying and non-identifying entities.) Finally, have them normalize their data model to see if they created it correctly.

I always make time to lecture about balance between data modeling and process modeling. At this point, I like to communicate how closely integrated the entire development process really is. It can be a good exercise to show an existing data model and then come up with a list of “client changes” that a pretend client has just asked for. Talk about how both the process and data models will need to change based on the list.

One thing I have tried that works nicely is to actually give students the data management chapters from the SAD textbook in the data management course. This provides for nice supplemental reading and creates a bridge for consistency between the two classes. During the SAD course, you can spend more time reviewing through hands-on exercises rather than a lecture presentation.

If your curriculum does not have a data management course that precedes SAD, I would spend much more time on this chapter. I would present the straightforward topics first (the first half of the chapter) and then assign some simple data models for students to do in groups and individually for homework. Then, I would spend a day doing hands-on class exercises. I would allow at least one more session for covering advanced concepts. I would present normalization as a way of validating the data models that that the students create.

I make the data model an important deliverable for the student’s semester projects, so I can usually get a good feel for how well they have absorbed the data management concepts.

# War Stories *(from Barbara Wixom)*

## Examples of real data models

The following examples come from real companies to give students an idea of the scope of some data models and databases …

1. At Western Digital Corporation (maker of hard drives), one hard drive is associated with 1,000 attributes.
2. 3M had to track in their data warehouse: 500,000 finished goods, 300,000 order related items per day, and over 1 million customer account numbers. The logical data model contains more than 1,000 tables that are pulled in from hundreds of source systems. The data warehouse has 3,000 data elements.
3. The IRS has an accounts receivable table with 1 billion rows.
4. Blue Cross and Blue Shield of North Carolina has a data warehouse. The database is 500GB. There are 200 tables and 1000 fields.
5. Medicaid in New York State answers more than 200 million medical claims each year. A database is set up in the Office of the Attorney General that has 850 million claims taking up 4 Terabytes on the server.

## Modeling Doesn’t Always Start from Scratch

A friend of mine, Bob, works for a consulting firm and told me about how many times logical data models are not created from scratch. In fact, many times when he works on a project, the developers need to work and adapt a model that already exists – even if it’s wrong!

One of Bob’s clients was an energy company that had created a logical data model that resided in Microsoft Access – and the model existed across 80 different Access files! When Bob arrived on the project, the project was already too far along to change the model too drastically because several applications were built using data already in Access. But, the model was terrible.

For example, monthly data existed such that each month of data was a new column in the database. Next year, the developers were going to have to add twelve more columns for the new year’s months. Also, financial drivers were pre-calculated and stored in columns – so if formulas or needs changed, the company couldn’t get the new information.

Part 2 – Normalization

Bob relied on normalization to first figure out where the model was weak, and he had the project team put the model into third normal form using Oracle’s modeling tool. He convinced the company to migrate the data in Access to Oracle into 3rd normal form structures, and eventually the existing applications grabbed data from Oracle instead of Access.

He had convinced the company that the time spent migrating the model to third-normal form, moving from Access to Oracle, and moving applications away from Access would save money in the long term. He was right.

# Answer to Your Turn 5-1: Applying the Concepts of Cardinality and Modality

1. Custom Drone [Order] and Ordered Drone Component:

* A custom drone order includes at least one (modality) and may include many (cardinality drone components
* An ordered drone component must be included in one custom drone order (modality) and may only be included in one custom drone order (cardinality).

These statements tell us that every custom drone order includes at least one drone component and may include more than one. An ordered drone component belongs to exactly on custom drone order.

2. Ordered Drone Component and Drone Component

* A drone component may be included in many ordered drone component instances (cardinality) but is not required to appear in any order drone component instances (modality)
* An ordered Drone Component includes one and only one drone component (modality and cardinality)

These statements tell us that an ordered drone component instance specifies exactly one drone component. A drone component may appear in many instances of ordered drone components, but is not required to appear in an ordered drone component instances.

# Answer to Your Turn 5-2: Evaluate Your CASE Tool

Student answers will vary depending upon the CASE tool selected. The images in the text illustrate Visible Analyst, but your images will vary based on the tool.

# Answer to Your Turn 5-3: Understanding the Elements of an ERD

\*ptg\_PaintingID

ptg\_ArtistID

ptg\_PaintingTitle

ptg\_PaintingDate

ptg\_PaintingDesc

PAINTING

\*art\_ArtistID

art\_ArtistLName

art\_ArtistFName

art\_ArtistAddress

art\_ArtistPhone

art\_ArtistEmail

ARTIST

\*res\_ReservationID

res\_PaintingID

res\_ReservationMuseumID

res\_ReservationDateOut

res\_ ReservationDateIn

RESERVATION

\*mus\_MuseumID

mus\_MuseumName

mus\_MuseumAddress

mus\_MuseumPOCName

mus\_MuseumPOCPhone

mus\_MuseumPOCEmail

MUSEUM

creates

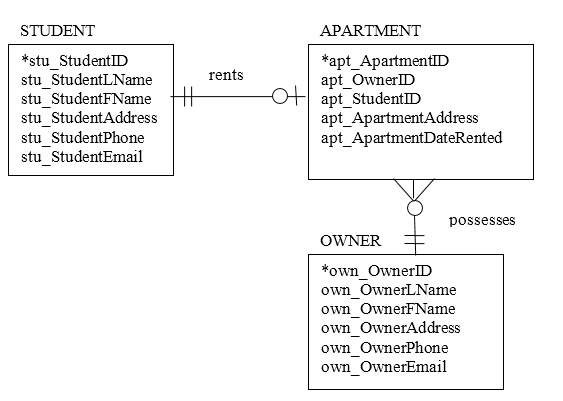
reserved by

requests

Examples of business rules include: One Museum may reserve none, one, or many paintings at a time, one Artist may create none, one, or many paintings.

# Answer to Your Turn 5-4: Campus Housing System

Student answers will vary depending upon previous models. An example would be:



# Answer to Your Turn 5-5: Independent Entities

Independent entities include: Client, Flight Request, Flight Request Notification, Pricing Guideline, Drone, and Pilot. An independent entity on the diagram has its own unique identifier; this is in contrast to the dependent entity Flight Bid that has a concatenated identifier made up of the FRN\_ID and PIL\_ID from surrounding independent entities.

# Answer to Your Turn 5-6: Dependent Entities

Dependent entity is Flight Bid, which uses the primary keys of its parent entities to form a concatenated primary key.

# Answer to Your Turn 5-7: Intersection Entities

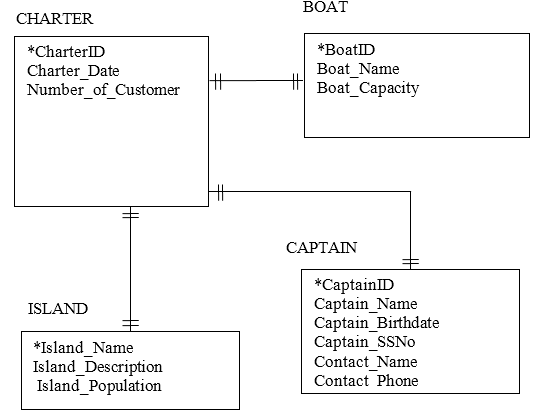
A Pilot owns at least one drone and may own many drones. A drone instance is owned by exactly one pilot.

A Pilot submits possibly many flight requests, but an instance of a pilot is not required to have an instance of a flight request. A flight request instance is submitted by exactly one pilot.

A Drone maybe included in many flight bids, but a drone instance is not required to have a related flight bid instance. A flight bid instance in includes exactly one drone.

# Answer to Your Turn 5-8: Boat Charter Company

Student responses will vary, one example of the Boat Charter Company:



All entities are independent. The model is in third normal form (3NF) in that all repeating groups are removed (1NF), no concatenating identifiers are present (2NF), and all attributes for each entity depend on the identifying attribute, that is, no non-key attribute depends upon any other non-key attribute.

# Answer to Concepts in Action 5-A: The User’s Role in Data Modeling

1. From these two stories, students can make the observation that the user’s role in data modeling can often depend upon the user. In the first story, the user kept herself well-informed about her system, and in the second, the users kept their distance from the modeling, either not needing or not wanting to become involved in that aspect of the project.
2. Users should be involved in data modeling early on in the process. Once an initial ERD has been developed, it can be presented to users for comments, suggestions, or simply confirmation.
3. While developers often have an in-depth understanding of the modeling process, they may not have an in-depth understanding of the business processes they are modeling. That is where users are needed. Having users in on a discussion of the model and the processes depicted will allow for either confirmation of the model or where the model is deficient.

# Answer to Concepts in Action 5-B: Implementing an Enterprise Information Management System

1. There are multiple issues to be addressed in this scenario, the foremost being that there is no centralized management of the data nor standards applied to data definitions. Student answers may vary, but the most important step would be to standardize data definitions among departments. Once that step has been taken, then a comprehensive model of the data should be created. That would allow identification of duplicate data.
2. A model of the existing processes is essential to determine the current state of data in the enterprise. The model would allow the developers to identify where data is duplicated, where data is missing, who needs what data, and when. Without a model in place to use as a reference, it would take significantly more time and effort to develop a centralized management system for the data.

# Solutions to End of Chapter Questions

1. *Provide three different options that are available for selecting an identifier for a student entity. What are the pros and cons of each choice?*

The three options available for selecting an entity identifier: (1) Put together a combination of attributes to serve as the identifier (such as first\_name and last\_name). (2) Sometimes, a single attribute is available that can serve as the identifier (such as social security number) (3) A new attribute can be created to serve as the identifier (such as student\_id).

Any of these ways of selecting an identifier is acceptable. The important point is to ensure that the chosen identifier uniquely identifies each instance of the entity.

1. *What is the purpose of developing an identifier for an entity?*

One of the aspects of the definition of an entity is the fact that there are multiple occurrences of the entity. If there are not multiple instances of something that is a potential entity, that something is not an entity in the system. Consequently, there must be a way of identifying each individual occurrence of an entity so that it can be picked out from amongst all the other instances of the entity. That is the purpose of having identifiers with unique values.

1. *What type of high-level business rule can be stated by an ERD? Give two examples.*

A business rule is a constraint or guideline to follow during operation of the system. Examples of business rules are: an order belongs to just one customer; a customer cannot cancel an order that has been shipped; a backorder can be created for an out of stock product. Business rules are expressed on ERDs by the kinds of relationships that the entities share.

1. *Define what is meant by an* ***entity*** *in a data model. How should an entity be named? What information about an entity should be stored in the CASE repository?*

The *entity* is the basic building block for a data model. It is a person, place, event, or thing about which data is collected—for example, an employee, an order, or a product. Entities names are nouns. Information stored in a CASE repository regarding an entity includes:

* Name
* Definition
* Special Notes
* User Contact

1. *Define what is meant by an* ***attribute*** *in a data model. How should an attribute be named? What information about an attribute should be stored in the CASE repository?*

An attribute is a characteristic that describes an entity. Attribute names are nouns. Information stored in a CASE repository regarding an attribute includes:

* Name
* Definition
* Alias
* Sample Values
* Acceptable Values
* Format
* Type
* Special Notes

1. *Define what is meant by a* ***relationship*** *in a data model. How should a relationship be named? What information about a relationship should be stored in the CASE repository?*

A relationship describes the association between entities. Relationship names are labeled as active verbs. Information stored in a CASE repository regarding a relationship includes:

* Verb Phrase
* Parent Entity
* Child Entity
* Definition
* Cardinality
* Modality
* Special Notes

1. *A team of developers is considering including ‘warehouse’ as an entity in their data model. The company for whom they are developing the system has just one warehouse location. Should “warehouse” be included? Why or why not?*

Entities represent something for which there exist multiple instances or occurrences. If there is only one instance of a warehouse, then it would not be best represented by an entity. However, if multiple warehouses were planned in the future, then a warehouse entity should be included.

1. *What is meant by a concatenated identifier?*

A concatenated identifier is one in which a combination of attributes serves to uniquely identify an entity. For instance, an appointment entity may have multiple instances for a single date identifier, but the combination of date and time will uniquely identify each instance.

1. *Describe, in terms a businessperson could understand, what are meant by the cardinality and modality of a relationship between two entities.*

These two terms are used to refer to the ‘numerical’ relationship between two entities in a data model. The term, cardinality, refers to the maximum number of times an instance of one entity can be related to instances of the other entity. If the cardinality is one -to-one, then we can infer that one instance of the parent entity can be related to just one instance of the child entity. If the cardinality is more than one, then we know that one instance of the parent entity can be related to more than one instance of the child entity. The determination of cardinality is based upon whatever is appropriate for the business situation being described. The term, modality, refers to whether or not an instance of a child entity can exist without a related instance in the parent entity. Modality values are either null or not null. Using more formal data modeling terminology, modality indicates whether the relationship between an entity instance and an instance of the related entity is “null” (optional) or “not null” (required). If the modality is null, then we can infer that no instances of the child entity are required for an instance of the parent entity. If the modality value is not null, then there must be one instance of the child entity for an instance of the parent entity. Just as in the case of cardinality, the determination of modality is based upon whatever is appropriate for the business situation being described.

1. *What are metadata? Why are they t important to system developers?*

Metadata isquite simply, data about data. Metadata is anything that describes an entity, attribute, or relationship, such as entity names, attribute descriptions, and relationship cardinality, and it is captured to help designers better understand the system that they are building and to help users better understand the system that they will use. Since there are typically several members of the project team, specifying metadata helps ensure that each team member has a consistent understanding of the data model components. Metadata is usually stored in the data dictionary. The metadata information can be used to integrate the different pieces of the analysis phase and can lead to a much better design.

1. *What is an independent entity? What is a dependent entity? How are the two types of entities differentiated on the data model?*

Independent entities are entities that can exist without the presence of another entity. The independent entity does not rely on any other entity in order to exist. On the other hand, dependent entities require the presence of another entity in order to exist. These entities rely on attributes from the parent entity to uniquely identify an instance, and therefore depend on another entity. For example, when an order is placed for a product, an entity that represents a specific product on a specific order is usually created. This entity would not exist without the order and product entities, and in fact gets its identifiers from those entities. So, this ordered\_product entity is a dependent entity. Independent entities are represented by rectangles with a single border line, while dependent entities are usually represented as rectangles with double-border lines.

1. *Explain the distinction between identifying and nonidentifying relationships.*

When relationships have an independent child entity, they are called nonidentifying relationships. When relationships have a dependent child entity, they are called identifying relationships.

1. *What is the purpose of an intersection entity? How do you know whether one is needed in an ERD?*

An intersection entity is created when we need to capture more information about the relationship between two entities. This often occurs when two entities have a many-to-many relationship. One instance of entity A may be related to many instances of entity B, and one instance of entity B can be related to many instances of entity A. The intersection entity is inserted between entities A and B, and is used to capture information about a specific instance of entity A related to a specific instance of entity B.

1. *Describe the three-step process of creating an intersection entity.*
2. Remove the M:N relationship line and insert a new entity between the two existing ones.
3. Add two 1:N relationships to the model.
4. Name the intersection entity.
5. *Is an intersection entity dependent or independent? Explain your answer.*

If the intersection entity has a logical identifier that can uniquely identify instances within, then it would be considered an independent entity. If, however, the intersection entity requires the identifiers from its parent entities to be uniquely identified, then it is a dependent entity.

1. *What is the purpose of normalization?*

Normalization is a process whereby a series of rules are applied to a logical data model or a file to determine how well formed it is. Normalization helps analysts identify entities that are not represented correctly in a logical data model, or entities that can be broken out from a file. The rules of normalization help assure that the data is stored as efficiently as possible.

1. *Describe the analysis that is applied to a data model in order to place it in first normal form (1NF).*

A logical data model is in first normal form (1NF) if it does not contain attributes that have repeating values for a single instance of an entity.

1. *Describe the analysis that is applied to a data model in order to place it in second normal form (2NF).*

Second normal form requires that the data model be in 1NF, and that all partial dependencies have been removed.

1. *Describe the analysis that is applied to a data model in order to place it in third normal form (3NF).*

*Third normal form* (*3NF*) occurs when a model is in both 1NF and 2NF and when, in the resulting entities, none of the attributes is dependent on a nonidentifier attribute (i.e., *transitive dependency*).

1. *Describe how the data model and process model should be balanced against each other.*

The key to balancing DFDs and ERDs is to recognize that all system data must be accounted for on each type of diagram. The DFD data components need to correspond with the ERD’s data stores (i.e., entities) and the data elements that comprise the data fl ows (i.e., attributes) depicted on the data model. That is one aspect of balancing. In addition, the attributes that are a part of the data model should be used somewhere in the flows and stores of the process models.

1. *What is a CRUD matrix? How does it relate to process models and data models?*

The CRUD (create, read, update, delete) matrix shows is a table that depicts how

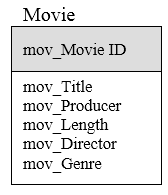
the system’s processes use the data within the system. It is helpful to develop the CRUD matrix on the basis of the logical process and data models and then revise it later in the design phase. The matrix also provides important information for program specifications, because it shows exactly how data are created and used by the major processes in the system.

# Solutions to End of Chapter Exercises

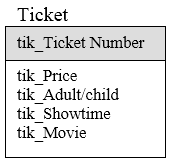
1. *Draw data models for the following entities:*

*Note: Primary Keys are shown at top of entity.*

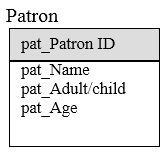
Movie (title, producer, length, director, genre)



Ticket (price, adult or child, showtime, movie)

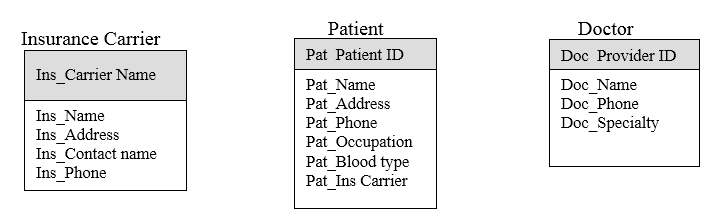


Patron (name, adult or child, age)



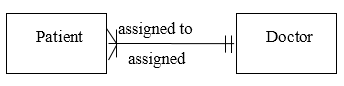
1. Draw a data model for the following entities, considering the entities as representing a system for a patient billing system and including only the attributes that would be appropriate for this *context:*

* *Patient (age, name, hobbies, blood type, occupation, insurance carrier, address, phone)*
* *Insurance carrier (name, number of patients on plan, address, contact name, phone)*
* *Doctor (specialty, provider identification number, golf handicap, age, phone, name)*

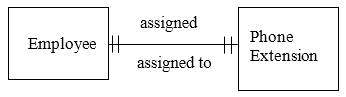


1. *Draw an entity relationship diagram (ERD) for the following situations. Would the relationships be identifying or nonidentifying? Why?*

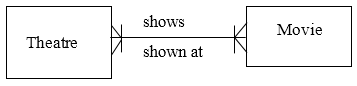
*1. Doctor's office - as the Doctor entity is an independent entity, (can exist without the Patient entity), this is a non-identifying relationship.*



*2. Phone - as the Phone Extension entity is an independent entity, (can exist without the Employee entity), this is a non-identifying relationship.*

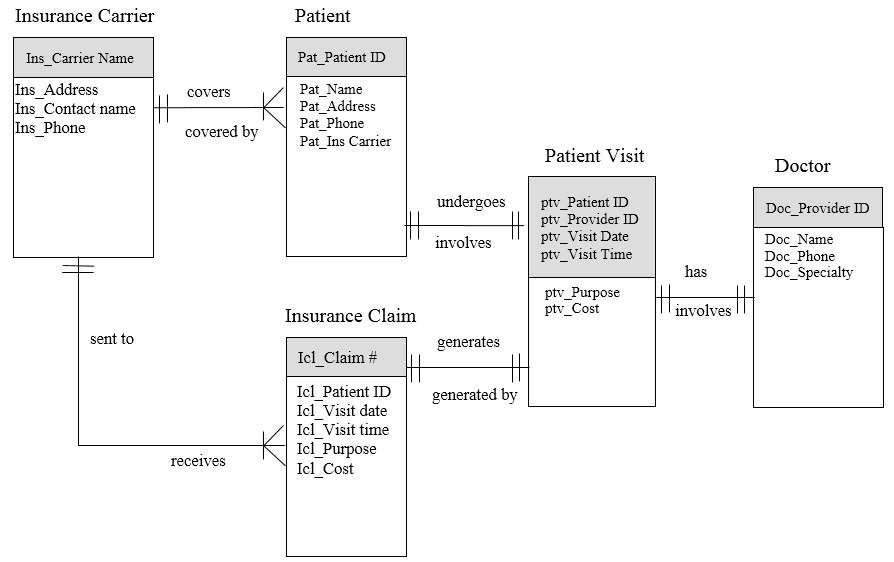


*3. Movie theatre - as the Movie entity is an independent entity, (can exist without the Theatre entity), this is a non-identifying relationship.*

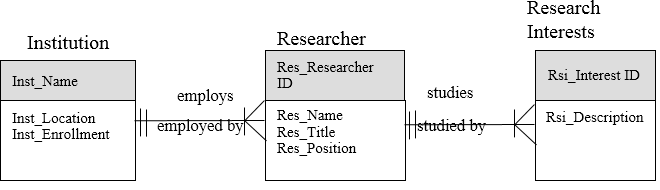


1. Draw an entity relationship diagram (ERD) for the following situations:

*1. Doctor's office*



*2. Researcher database*



*3. Bridal registry*

Registered

Product

Customer

Products

includes

Prd\_Product ID

Rpd\_Customer ID

Rpd\_Product ID

cus\_Customer ID

cus\_Bride name

cus\_Groom name

cus\_Phone

cus\_Wedding date

registered for

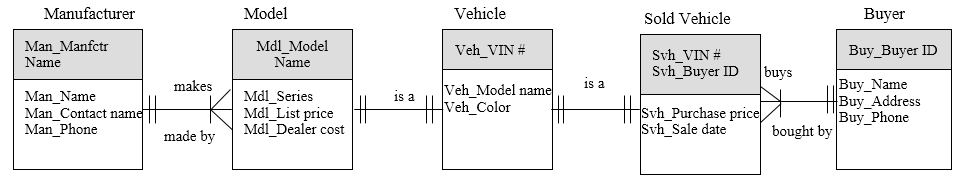
Is part of

registered by

Prd\_Description

Prd\_Price

*4. Car dealership*



1. *Examine the data models that you created for Exercise D. How would the respective models change (if at all) on the basis of these corresponding new assumptions?*
2. *Two patients have the same first and last names.*

No change. The Patient ID takes care of uniquely identifying instances of patient.

1. *Researchers can be associated with more than one institution.*

Would need a new intersection entity whose attributes link a specific researcher to a specific institution. This would be a dependent entity.

1. *The store would like to keep track of purchased items*

Would need a new entity to describe purchased items for members of the registry.

1. *Many buyers have purchased multiple cars from Jim over time because he is such a good dealer.*

Need to add the Sales Date to the primary key of the Sold Vehicle.

1. *Visit a Web site that allows customers to order a product over the Web…*

Student answers will vary.

1. *Create metadata entries for the following data model components…*

Entity:

|  |  |
| --- | --- |
| **Entity Name:** | Product |
| **Definition:** | The product entity refers to the items that are produced by the company |

Attribute:

|  |  |
| --- | --- |
| **Attribute Name:** | Product Number |
| **Aliases:** | Product ID; Product Num |
| **Entity:** | Product |
| **Description:** | A seven-digit code that uniquely identifies the product. The code consist of three alpha characters and four numeric digits |
| **Data Type:** | Text |
| **Identifier** | Yes |
| **Notes:** |  |

Attribute:

|  |  |
| --- | --- |
| **Attribute Name:** | Product Type |
| **Aliases:** | Type |
| **Entity:** | Product |
| **Description:** | A three-digit code that categorizes the product. |
| **Data Type:** | Text |
| **Identifier** | No |
| **Notes:** | The coding scheme for the product type categories should be entered here when it is determined |

Relationship:

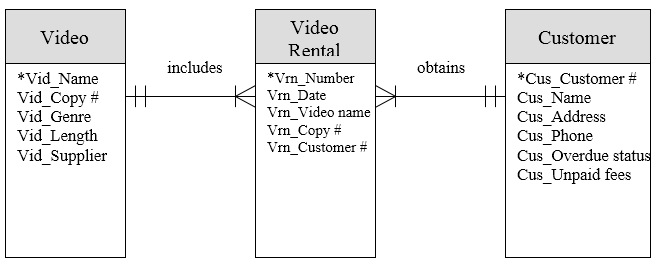
|  |  |
| --- | --- |
| **Parent Entity:** | Company |
| **Child Entity:** | Product |
| **Relationship:** | Makes / made by |
| **Cardinality:** | 1:N |
| **Modality:** | Not null |
| **Notes:** |  |

1. *Describe the assumptions that are implied from the data model shown.*

* A cinema contains many theatres; a theatre belongs to just one cinema.
* A theatre offers many showings; a showing is offered by just one theatre.
* A movie may be viewed at zero or more showings; a showing is for just one movie.

1. *Create a data model for one of the processes in the end-of-chapter Exercises for Chapter 4. Explain how you would balance the data model and process model.*

Data model for Exercise 4-F, Video Store System:

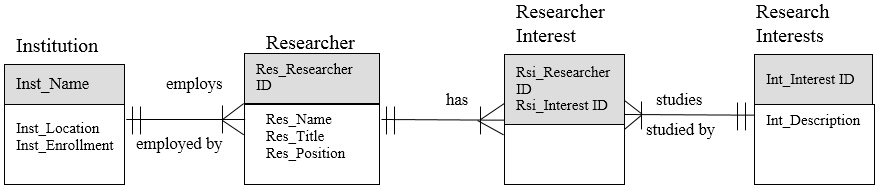


Generally, all of the data entities shown on the ERDs will correspond to data stores on the DFDs. That is one aspect of balancing. In addition, the attributes that are a part of the data model should be used somewhere in the flows and stores of the process models.

1. *Apply the steps of normalization to validate the models you drew in Exercise D.*

Problem 1 is in 3rd normal form

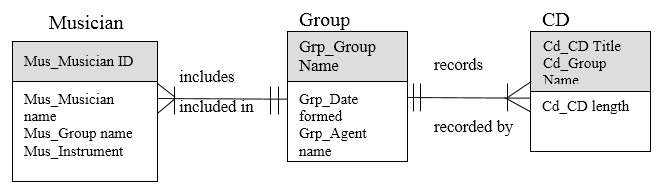
Problem 2: the many-to-many relationship between researcher and research interest should be resolved:



Problem 3 is in 3rd normal form

Problem 4 is in 3rd normal form

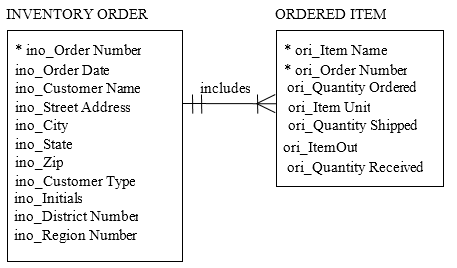
1. *Create a logical data model in third normal form for the CD information given:*



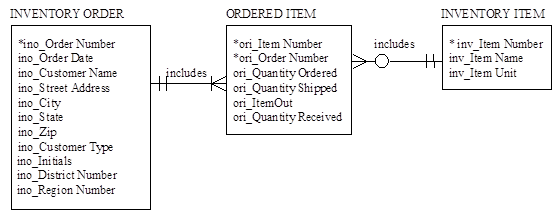
# Answers to Textbook Minicases

1. a. Data modeling is another tool that is used by systems developers to help represent the key elements of a system. While process models help us understand how data is created and used in the system, we also need a way to model and represent just the data component of the system.
2. The only way we can possibly create a valuable new system is to understand clearly the data component of that system. Data models help us focus strictly on the data, and so are necessary parts of the work performed by the project team.
3. We develop data models of the current system for several reasons. First, it is easiest for the users to discuss the data that is currently being used in the existing system, and to understand the business rules that apply to this data. We will learn specialized terminology and jargon by talking to the users about the existing data, and that will enhance our understanding of the overall system. Second, the data component of a system tends to be pretty stable. In many projects, the data component of the new system is not very different than the data component of the existing system. The changes made in the data model when going from the existing system to the new system are not nearly as extensive as the changes that are made to the process model. Therefore, it is valuable to develop an As-Is data model, since it will probably represent the data foundation of the new system.
4. Developing the process model of the existing system is helpful in developing the data model of the As-Is system. Review of the scenarios that were developed, data flow diagrams, and the process descriptions that were written will help the team identify data entities and the entity attributes that are required by the business system. This will enable the team to get a good start on the As-Is system data model.

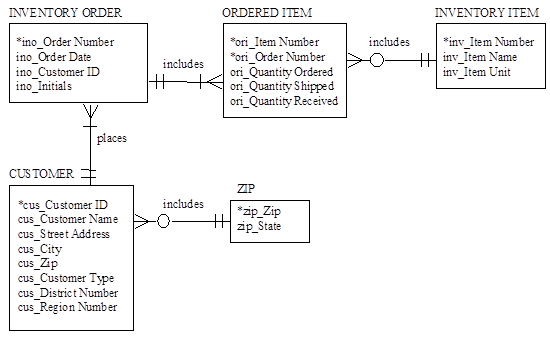
*2. a. The rule for first normal form is to remove any attributes or groups of attributes that occur more than once for a single occurrence of the entity. The repeating group of attributes that describe each item included in the order is separated out into the Ordered Item entity.*



*2. b. The rule for second normal form is to evaluate all entities with concatenated keys to verify that all non-key attributes depend on the full concatenated key, not just part of the key. The Ordered Item entity has a concatenated key, and so must be placed in second normal form. Note that a new attribute has been added to serve as primary key for the Item entity. The attributes Item Name, and Item Unit depend only on the Item Number, not on the full key of Item Number and Order Number. Therefore, they have been separated out into the Item entity.*



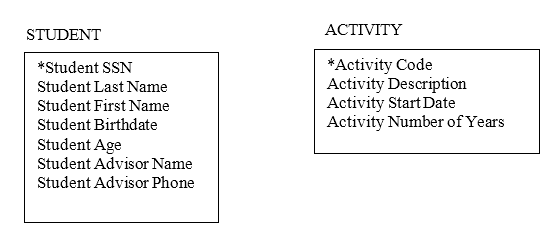
*2. c. To place an entity in third normal for, the analyst must verify that the values of any non-key attribute do not depend on any other non-key attributes. For the Inventory Order entity, the Address, City, State, Zip, Customer Type, District Number, and Region Number all depend upon the Customer Name. These attributes should be moved to an entity that just describes the customer. Note that a Customer ID has been added as the primary key of that entity. Within the customer’s address information, the State attribute depends on the Zip attribute. A separate Zip entity has been created. Also, in the Ordered Item entity, the Item Out attribute is a field that simply flags a situation when the Quantity Ordered and Quantity Shipped are different. Therefore, Item Out can be determined through comparing Quantity Ordered and Quantity Shipped, and should not be stored. It has been removed from the Ordered Item entity.*



*2. d. One should balance the DFDs and ERDs to ensure that all system data is accounted for on each type of diagram. If the data in the data model is not created, stored and used in the process model, then the analyst should check to see if the data is really needed, or if it has been inadvertently omitted from the process model. Similarly, the data created, stored and used in the process model should be associated with an appropriate entity in the data model. The analyst should investigate any discrepancies.*

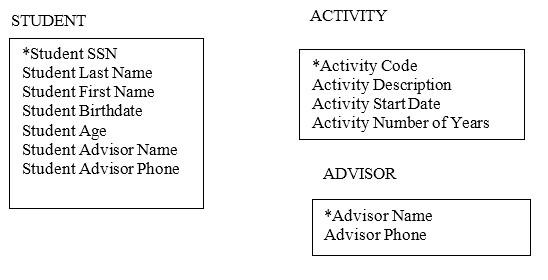
# Answer to Your Turn 5A-1: Normalizing a Student Activity File

The rule for first normal form is to remove any attributes or groups of attributes that occur more than once for a single occurrence of the entity. The repeating group of attributes that describe each item included in the activity is separated out into the Activity entity.



The rule for second normal form is to evaluate all entities with concatenated keys to verify that all non-key attributes depend on the full concatenated key, not just part of the key. In this case there are no concatenated keys, so the model is considered to be in 2NF.

To place an entity in third normal for, the analyst must verify that the values of any non-key attribute do not depend on any other non-key attributes. For the Student entity, the Advisor Phone depends upon the Advisor Name. These attributes should be moved to an entity that just describes the advisor.



# Supplemental Minicases

1. The project team at Williams Specialty Company (see Chapter 5 Minicases) reached the point of completing the process model and data model for the to-be system. The four staff members who are most familiar with the business have been heavily involved in defining needed improvements in the system and sharing their ideas for the way they’d like to see things work in the new system. The users role played the new logical process model and were pleased with the way the new system’s processes were outlined.

As project manager, you are anxious to keep the project moving into the design phase. You know, however, that your team should be sure and check to see that the process model and data model are balanced before proceeding with the design phase work. Your newest team member did a good job in balancing the data flow diagrams. You believe it would be a valuable learning experience for him to work through balancing the process and data models now.

Explain to your new team member why balancing the DFDs and ERDs is important. Explain exactly what he should be looking for as he reviews the diagrams.

*Answer: The key to balancing DFDs and ERDs is to recognize that all system data must be accounted for on each type of diagram. The ERD shows the system data ‘at rest,’ while the DFD shows the flow and use of data in the system. Generally, all of the data entities shown on the ERDs will correspond to data stores on the DFDs. This is the first issue that the analyst should check out when reviewing the diagrams.*

*Another aspect to be investigated is the attributes that are a part of the data model. These attributes should be used somewhere in the process models. If the data in the data model is not created, stored and used in the process model, then the analyst should check to see if the data is really needed, or if it has been inadvertently omitted from the process model. Similarly, the data created, stored and used in the process model should be associated with an appropriate entity in the data model. The analyst should investigate any discrepancies. The analyst should work with the metadata that documents the data entities and data attributes in the data models and the data flow and data store entries in the process model, to verify that the data component of system has been correctly represented in the DFDs and ERDs.*

1. Thomas is a systems analyst for Top Flight Systems Consulting Co., Inc. Thomas has been with Top Flight Systems for three years and has so far worked primarily as a programmer on several large implementation engagements. At long last, Thomas has been assigned to a project team as a systems analyst, and he has worked hard to apply the concepts he learned in college to his work.

Thomas’ manager, Jeff, is a hard-driving perfectionist who has Thomas more than a little intimidated. Jeff’s demanding standards and blunt criticisms did a lot to motivate Thomas to do well on the development of the process model for their current project. Thomas spent long hours carefully balancing the data flow diagrams, making sure that the terminology was consistent, and decomposing the diagrams down to the same level of detail. Thomas was relieved when Jeff’s review of his work resulted in only a few minor changes to the models.

Now it’s time to begin development of the system’s data model, and Thomas is more nervous about this task. Data modeling was never his strength in college. It always seemed so mysterious. Thomas began his first draft of the data model by going through the process model descriptions and identifying data entities and attributes that seemed important to the business system. He then sketched in some relationships between entities, and tried to determine the relationships’ cardinality and modality. He always got confused about those two concepts. He knew he’d need to review some of his college textbooks at home tonight to try and make sure he’d done things right.

Suddenly, Jeff appeared at his desk and snatched up the ERDs he’d just drafted. After glancing through them, Jeff threw them down on Thomas’ desk with disgust. “Thomas,” said Jeff, “is this the best you can do? In just two minutes I can see at least three relationships where you’ve messed up the cardinality and modality. Plus, I’ve never seen such a bunch of Many-to-Many relationships on one diagram. Haven’t you ever heard of an intersection entity? This kind of work won’t cut it here at Top Flight, Thomas. I expect to see a data model that makes sense tomorrow!” As Jeff turned on his heel and walked away, Thomas felt about two inches tall.

Help Thomas out. Prepare a concise review of cardinality and modality that will help him understand the concepts and apply them correctly to his model. Also, develop an explanation of intersection entities and how they can be used to remove many-to-many relationships from a data model.

*Answer: The term, cardinality, refers to the maximum number of times an instance in one entity can be related to instances of the other entity. If the cardinality is one, then we can infer that one instance of the entity can be related to just one instance of the other entity. If the cardinality is more than one, then we know that one instance of the entity can be related to more than one instance of the other entity. The determination of cardinality is based upon whatever is appropriate for the business situation being described. The term, modality, refers to the minimum number of times that an instance of one entity can be related to instances of the other entity. Modality values are either zero or one. If the modality is zero, then we can infer that no instances of the other entity are required for an instance of the entity (i.e., an instance of the related entity is not required). If the modality value is one, then there must be one instance of the related entity for an instance of the entity (i.e., an instance of the related entity is required.). Just as in the case of cardinality, the determination of modality is based upon whatever is appropriate for the business situation being described.*

*An intersection entity is created when there is a need to capture more information about the relationship between two entities. This often occurs when two entities have a many-to-many relationship. In this case, one instance of entity A may be related to many instances of entity B, and one instance of entity B can be related to many instances of entity A. The intersection entity is inserted between entities A and B, and is used to capture information about a specific instance of entity A as it relates to a specific instance of entity B. After discovering a many-to-many relationship, it is common for the analyst to recognize that the entity relationship itself needs to be described with data. In this case, the intersection entity should be used to enable the data model to show this information.*

1. Jeff, a project manager at Top Flight Systems Consulting Co., Inc., has frequently encountered junior systems analysts on his project teams who had great difficulty developing correct data models (see Supplemental Minicase #2). Frankly, Jeff was a little tired of always finding glaring errors in his team’s data models, then trying to be patient as the team member struggled to get things right. The time pressure on the consulting engagements always weighed heavily on Jeff’s shoulders.

Since data models seemed to be a particular sticking point for the junior analysts, Jeff decided to develop a set of rules and guidelines for his junior analysts to use in developing data models. Jeff hoped that by providing a concise summary of data modeling tips, he can improve the junior analysts’ performance on this important systems development technique.

Develop Jeff’s list of data modeling tips and suggestions. Include all recommendations that you would find helpful as a junior systems analyst just starting out as a data modeler.

*Answer: Guidelines that may be helpful to a junior systems analyst regarding data modeling:*

1. *An entity is a person, place, thing, object, or event about which we wish to capture data in our system.*
2. *It is helpful to remember that there must be more than one occurrence of an entity. If there is just one occurrence of the entity in the system, we will not capture data about it.*
3. *Attributes should be identified for all characteristics of the entity that are important to the business system. If an attribute is not relevant to the business system, omit it. On the other hand, if the attribute is relevant to the business system, be sure and include it.*
4. *Include attributes that have been expressed at the appropriate level of detail. For example, if we want to identify a customer by first name, middle initial, and last name, then include all three as attributes.*
5. *Label all components of the data model. Use the correct business terminology for the labels.*
6. *Data models should reflect the business rules as expressed by the users. The users have the most knowledge about how data entities are related to each other. Assumptions that are made about the business relationships should be explicitly stated.*
7. *Define relationship cardinality and modality correctly.*
8. *When many-to-many relationships occur, determine if an intersection entity is appropriate. These entities are generally called for when specific data needs to be captured to relate a specific entity instance to another specific entity instance.*

# Experiential Exercises

1. Purpose: To discover the importance of making assumptions explicit when developing entity relationship diagrams.

Working in pairs, have students develop an entity relationship diagram that represents the student registration system at your university. When the diagrams are finished, have the class discuss and compare the entities that were chosen, the attributes that were identified, and the relationships that were formulated. Focus the discussion on the different results that were obtained by each group. Did the students make their assumptions explicit? Who would they need to talk to in order to validate their assumptions?

1. Purpose: To learn how entity relationship diagrams are used in actual practice.

Invite a systems analyst or database administrator as a class guest. Ask the guest to bring samples of entity relationship diagrams that have been prepared for a system in his/her organization. Some interesting topics to have the guest discuss include: the diagramming standards of his/her organization; the use of CASE tools in his/her organization (or not); the role of iteration in developing entity relationship diagrams; how successful the entity relationship diagrams are in conveying the meaning of the system data to the end users.

1. Purpose: To improve understanding of entity relationships.

Working in pairs, have students generate a list of entities that are likely to be associated with information systems in hospitals. Then have the students formulate relationships between the entities and assign the cardinality and modality of the relationship. Have the students discuss the factors that would cause them to change the relationships and/or the cardinality/modality of the relationships.

1. Purpose: To understand the need to balance ERDs and DFDs.

Working with the Patient Registration system example in later chapters of the textbook, have the students trace the common data elements between the two models. Have the students identify the data entities/data stores in both models. Discuss the potential problems of not balancing the two models.