Chapter 8 teaches students the important skill of data modeling. Students learn the underlying system concepts that apply to data models, and then they learn how to construct them. The chapter also teaches the students how to analyze the completed data model to ensure that the final model is simple, non-redundant, flexible, adaptable, and ready for implementation.

Chapter to Course Sequencing

Students are encouraged to read Chapter 5 to provide perspective for data modeling. In addition, data modeling will be easier for most students to perform after a thorough requirements analysis has been completed. Therefore, Chapters 6 and 7 are useful prerequisites. There has always been disagreement concerning the sequencing of the modeling chapters. Classical structured techniques taught process modeling first. Contemporary information engineering techniques suggest that data models should be taught first. We prefer to teach data modeling first because:

- More of the industry now practices information engineering than structured analysis.
- Synchronization of data and process models is simpler if data models are constructed first.
- Data models can usually be constructed faster than process models. This has special benefits to development approaches (routes) that utilize a RAD or prototyping strategy.
- We sincerely believe that analysts gain a deeper and quicker understanding of business terminology and business rules through data modeling.

What’s Different Here and Why?

This chapter did not necessitate many changes from the sixth edition. The following changes have been made to this chapter in the seventh edition:

1. As with all chapters, we have streamlined the SoundStage episode into a quick narrative introduction to the concepts presented the chapter.
2. We added key terms for "Child Entity," "Parent Entity," "Metadata," and "Transitive Dependency."

3. We added a paragraph under Entity Discovery to discuss how use cases can be used to discover entities.

**Lesson Planning Notes for Slides**

The following instructor notes, keyed to slide images from the PowerPoint repository, are intended to help instructors integrate the slides into their individual lesson plans for this chapter.

**Slide 1**

This repository of slides is intended to support the named chapter. The slide repository should be used as follows:

- Copy the file to a unique name for your course and unit.
- Edit the file by deleting those slides you don't want to cover, editing other slides as appropriate to your course, and adding slides as desired.
- Print the slides to produce transparency masters or print directly to film or present the slides using a computer image projector.

Most slides include instructor notes. In recent versions of PowerPoint, notes by default display in a window under the slide. The instructor notes are also reprinted below.

**Slide 2**

No additional notes.

---

Objectives

- Define data modeling and explain its benefits.
- Recognize and understand the basic concepts and constructs of a data model.
- Read and interpret an entity relationship data model.
- Explain how data models are constructed during a project and where the models are stored.
- Discover entities and relationships.
- Construct an entity-relationship context diagram.
- Discover or invent keys for entities and construct a key-based diagram.
- Construct a fully attributed entity relationship diagram and describe data structures and attributes to the repository.
- Normalize a logical data model to remove impurities that can make a database unstable, inflexible, and nonscalable.
- Describe a useful tool for mapping data requirements to business operating locations.
Teaching Notes
This slide shows the how this chapter's content fits with the building blocks framework used throughout the textbook. The emphasis of this chapter is upon the DATA. It also reflects the fact that data modeling may be performed during certain analysis phases and involves not only systems analysts…but owners and users.

Data Modeling

Data modeling – a technique for organizing and documenting a system's data. Sometimes called database modeling.

Entity relationship diagram (ERD) – a data model utilizing several notations to depict data in terms of the entities and relationships described by that data.

Teaching Notes
Be sure to explain that this is merely an example – there are numerous data modeling notations. While they may differ in appearance (symbology) the knowledge that data models are intended to convey the same.
Data Modeling Concepts: Entity

Entity – a class of persons, places, objects, events, or concepts about which we need to capture and store data.

- Named by a singular noun
  - Persons: agency, contractor, customer, department, division, employee, instructor, student, supplier.
  - Places: sales region, building, room, branch, campus.
  - Events: application, award, cancellation, class, flight, invoice, order, registration, renewal, requisition, reservation, sale, trip.
  - Concepts: account, block of time, bond, course, fund, qualification, stock.

Teaching Notes:
Prompt the students for additional examples. Have them classify their example(s). Obtain a data model from a source other than the textbook. Ask the students to classify the entities.

Data Modeling Concepts: Entity

Entity instance – a single occurrence of an entity.

<table>
<thead>
<tr>
<th>Student ID</th>
<th>First Name</th>
<th>Last Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>2144</td>
<td>Arnold</td>
<td>Betty</td>
</tr>
<tr>
<td>3122</td>
<td>Taylor</td>
<td>John</td>
</tr>
<tr>
<td>3843</td>
<td>Simmons</td>
<td>Lisa</td>
</tr>
<tr>
<td>9844</td>
<td>Macy</td>
<td>Bill</td>
</tr>
<tr>
<td>2837</td>
<td>Leah</td>
<td>Heather</td>
</tr>
<tr>
<td>2293</td>
<td>Wrench</td>
<td>Tim</td>
</tr>
</tbody>
</table>

Teaching Notes
Substitute the name(s) of one or more of your students. Be sure to explain that these are "instances" and that instances do NOT appear in the names of entity symbols.

Data Modeling Concepts: Attributes

Attribute – a descriptive property or characteristic of an entity. Synonyms include element, property, and field.

- Just as a physical student can have attributes, such as hair color, height, etc., data entity has data attributes.

Compound attribute – an attribute that consists of other attributes. Synonyms in different data modeling languages are numerous: concatenated attribute, composite attribute, and data structure.

Teaching Notes:
Go back to the slide showing the sample ERD (Figure 8-1). Pick an entity and ask the students to list attributes that they feel describe those entities. Show the students a form. Ask the students to identify the attributes. Be sure that the students recognize what items appearing on the form are truly attributes and those that are simply headings or preprinted items. Also, often students accidentally identify attribute values as attributes. For example, they may say that an item that appears as a check box is an attribute when in fact it may be the value of an attribute (ie. Male and female are values, whereas GENDER is the real attribute).
Data Modeling Concepts: Data Type

Data type – a property of an attribute that identifies what type of data can be stored in that attribute.

### Representative Logical Data Types for Attributes

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Logical/Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEXT</td>
<td>Any number, real or integer</td>
</tr>
<tr>
<td>DATE</td>
<td>Any date in any format</td>
</tr>
<tr>
<td>TIME</td>
<td>Any time in any format</td>
</tr>
<tr>
<td>NUMBER</td>
<td>Any number, real or integer</td>
</tr>
<tr>
<td>MEMO</td>
<td>Any picture or image</td>
</tr>
<tr>
<td>VALUE SET</td>
<td>A finite set of values. In most cases, a coding scheme would be established (e.g., FR=Freshman, SO=Sophomore, JR=Junior, SR=Senior)</td>
</tr>
<tr>
<td>YES/NO</td>
<td>Any boolean value (e.g., ON, OFF)</td>
</tr>
<tr>
<td>LOGIC</td>
<td>Logical Business Meaning</td>
</tr>
</tbody>
</table>

Teaching Notes

These are generic data types. If your students have taken a database class they would be familiar with the data types specific to that database.

Data Modeling Concepts: Domains

Domain – a property of an attribute that defines what values an attribute can legitimately take on.

### Representative Logical Domains for Logical Data Types

<table>
<thead>
<tr>
<th>Data Type</th>
<th>Domain</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>TEXT</td>
<td>No limit, specify the range</td>
<td>null</td>
</tr>
<tr>
<td>DATE</td>
<td>format on the MMDDYYYY format, MMDD</td>
<td>null</td>
</tr>
<tr>
<td>TIME</td>
<td>format on the HHMM format, HHMM</td>
<td>null</td>
</tr>
<tr>
<td>NUMBER</td>
<td>{10-99}</td>
<td>0</td>
</tr>
<tr>
<td>VALUE SET</td>
<td>{YES, NO}</td>
<td>NULL</td>
</tr>
</tbody>
</table>

Teaching Notes

No additional notes.

Data Modeling Concepts: Default Value

Default value – the value that will be recorded if a value is not specified by the user.

### Permissible Default Values for Attributes

<table>
<thead>
<tr>
<th>Default Value</th>
<th>Interpretation</th>
<th>Examples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Required</td>
<td>For an instance of the attribute, require that the user enter a legal value from the domain. (This is used when no value in the domain is common enough to be a default but some value must be entered.)</td>
<td>REQUIRED</td>
</tr>
<tr>
<td>Not NULL</td>
<td>For an instance of the attribute, require that the user enter a legal value from the domain. (This is used when no value in the domain is common enough to be a default but some value must be entered.)</td>
<td>NOT NULL</td>
</tr>
<tr>
<td>NONE</td>
<td>For an instance of the attribute, if the user does not specify a value, then use this value</td>
<td>NULL</td>
</tr>
<tr>
<td>DEFAULT</td>
<td>For an instance of the attribute, if the user does not specify a value, then use this value</td>
<td>DEFAULT</td>
</tr>
</tbody>
</table>

Teaching Notes

No additional notes.
Data Modeling Concepts: Identification

**Key** – an attribute, or a group of attributes, that assumes a unique value for each entity instance. It is sometimes called an identifier.

- **Concatenated key** - a group of attributes that uniquely identifies an instance. Synonyms: composite key, compound key.
- **Candidate key** – one of a number of keys that may serve as the primary key. Synonym: candidate identifier.
- **Primary key** – a candidate key used to uniquely identify a single entity instance.
- **Alternate key** – a candidate key not selected to become the primary key. Synonym: secondary key.

**Teaching Notes**

Students can generally relate to the following example. Suppose you are working for an hourly wage. The employer has some method of tracking the hours you work. Whether that involves a time clock, an identification badge that it scanned, or a log book, the system records a certain number of hours and some employee identifier that says those hours are yours. Without that identifier, come pay day the employer would not know whose hours were whose. The employer might pay someone else for the hours you worked. That’s how important a primary key or identifier is.

Data Modeling Concepts: Subsetting Criteria

**Subsetting criteria** – an attribute(s) whose finite values divide all entity instances into useful subsets. Sometimes called an inversion entry.

**Teaching Notes**

No additional notes

Data Modeling Concepts: Relationships

**Relationship** – a natural business association that exists between one or more entities.

The relationship may represent an event that links the entities or merely a logical affinity that exists between the entities.

**Teaching Notes**

Explain that there may be more than one relationship between two entities. You may reinforce this by adding additional relationships to the example (such as “transferred from”) to reflect a relationship where students changed from one curriculum to another.)
Data Modeling Concepts: Cardinality

Cardinality – the minimum and maximum number of occurrences of one entity that may be related to a single occurrence of the other entity. Because all relationships are bidirectional, cardinality must be defined in both directions for every relationship.

Slide 15

Cardinality Notations

Although this figure shows five different options, help students see that there are really only two options for minimum cardinality (0 or 1) and two options for maximum cardinality (1 or many).

Slide 16

Data Modeling Concepts: Degree

Degree – the number of entities that participate in the relationship.

A relationship between two entities is called a binary relationship.
A relationship between three entities is called a 3-ary or ternary relationship.
A relationship between different instances of the same entity is called a recursive relationship.

Teaching Notes:
Ask the students to read (or write) declarative sentences to reflect the bidirectional meaning of the relationship between student and curriculum.

Teaching Notes:
Although this figure shows five different options, help students see that there are really only two options for minimum cardinality (0 or 1) and two options for maximum cardinality (1 or many).

Teaching Notes:
Provide the students with an ERD that does not contain relationships. Ask the students to identify possible relationships and indicate a possible degree for that relationship. Emphasize to the students that the degree represents a business rule! Failure to accurately identify and document the degree will result in a system that does not reflect a correct business requirement.
Data Modeling Concepts: Degree

Relationships may exist between more than two entities and are called N-ary relationships. The example ERD depicts a ternary relationship.

Teaching Notes
The example also depicts an associative entity for the first time...as explained on the next slide.

Data Modeling Concepts: Degree

Associative entity - an entity that inherits its primary key from more than one other entity (called parents).

Each part of that concatenated key points to one and only one instance of each of the connecting entities.

No additional notes.

Data Modeling Concepts: Recursive Relationship

Recursive relationship - a relationship that exists between instances of the same entity.

COURSE
Course ID (Primary Key)
Subject Abbreviation
Course Number
Course Title
Course Credit

Teaching Notes
Ask the students to read (or write) declarative sentences to reflect the bidirectional meaning of the relationship. We created a composite key in this example. Be sure to point out the notation. Another classic example of a recursive relationship is in an Employee entity with a Supervisor attribute that holds the identifier of the supervisor’s instance of that same Employee entity.
Data Modeling and Analysis

Slide 21

Data Modeling Concepts: Foreign Keys

Foreign key – a primary key of an entity that is used in another entity to identify instances of a relationship.

- A foreign key is a primary key of one entity that is contributed to (duplicated in) another entity to identify instances of a relationship.
- A foreign key always matches the primary key in the other entity.
- A foreign key may or may not be unique (generally not).
- The entity with the foreign key is called the child.
- The entity with the matching primary key is called the parent.

Teaching Notes
Foreign keys are what make a relational database relational.

Slide 22

Data Modeling Concepts: Parent and Child Entities

Parent entity - a data entity that contributes one or more attributes to another entity, called the child. In a one-to-many relationship the parent is the entity on the "one" side.

Child entity - a data entity that derives one or more attributes from another entity, called the parent. In a one-to-many relationship the child is the entity on the "many" side.

Teaching Notes
These concepts are illustrated on the next slide.

Slide 23

Data Modeling Concepts: Foreign Keys

Teaching Notes
Have students identify which is the parent entity (Major) and which is the child (Student). Additional examples should be given to test the student’s ability to recognize the parent entity. We suggest you also provide an example of a one-to-one relationship!
Data Modeling Concepts: Nonidentifying Relationships

Nonidentifying relationship – relationship where each participating entity has its own independent primary key
• Primary key attributes are not shared.
• The entities are called strong entities.

No additional notes.

Data Modeling Concepts: Identifying Relationships

Identifying relationship – relationship in which the parent entity’s key is also part of the primary key of the child entity.
• The child entity is called a weak entity.

No additional notes.

Data Modeling Concepts: Sample CASE Tool Notations

Teaching Notes
Add slides to show students how the CASE or modeling tool that will be used in the class differentiates between identifying and nonidentifying relationships and weak and strong entities.
Data Modeling Concepts: Nonspecific Relationships

Nonspecific relationship — relationship where many instances of one entity are associated with many instances of another entity. Also called many-to-many relationship. Nonspecific relationships must be resolved, generally by introducing an associative entity.

Teaching Notes
You may need to refer to the earlier slide that defines an associative entity. In the bottom diagram we see that a student can declare multiple majors and that a curriculum can offer multiple majors. Note that for associative entities the cardinality from child to parent is always one and only one. An instance of MAJOR must correspond to one and only one STUDENT and to one and only one CURRICULUM.

Resolving Nonspecific Relationships

Teaching Notes
Part (c) of Figure 8-9 is on the next slide.

Resolving Nonspecific Relationships (continued)

No additional notes.
Resolving Nonspecific Relationships (continued)

While the above relationship is a many-to-many, the many on the BANK ACCOUNT side is a known maximum of "2." This suggests that the relationship may actually represent multiple relationships... in this case two separate relationships.

Data Modeling Concepts: Generalization

**Generalization** – a concept wherein the attributes that are common to several types of an entity are grouped into their own entity.

**Supertype** – an entity whose instances store attributes that are common to one or more entity subtypes.

**Subtype** – an entity whose instances may inherit common attributes from its entity supertype and then add other attributes unique to the subtype.

Generalization Hierarchy

**Teaching Notes**

This shows that while most nonspecific relationships are resolved by introducing a third entity, some are resolved by introducing separate relationships.

If students have already studied object-oriented modeling, they will be familiar with generalization from that. Ask the students to identify a generalization example. What are some common attributes? What are some unique attributes associated with the subtype(s)? One common example is EMPLOYEE (supertype) with HOURLY EMPLOYEE (subtype) and SALARY EMPLOYEE (subtype).

**Teaching Notes**

Generalization can be multiple levels deep.
**Slide 33**

**Process of Logical Data Modeling**

- Strategic Data Modeling
  - Many organizations select IS development projects based on strategic plans.
  - Includes vision and architecture for information systems
  - Identifies and prioritizes development projects
  - Includes enterprise data model as starting point for projects

- Data Modeling during Systems Analysis
  - Data model for a single information system is called an application data model.

**Teaching Notes**

Data modeling may be performed during various types of projects and in multiple phases of projects. Data models are progressive and should be considered a living document that will change in response to changing business needs.

**Slide 34**

**Logical Model Development Stages**

1. Context Data model
   - Includes only entities and relationships
   - To establish project scope
2. Key-based data model
   - Eliminate nonspecific relationships
   - Add associative entities
   - Include primary and alternate keys
   - Precise cardinalities
3. Fully attributed data model
   - All remaining attributes
   - Subsetting criteria
4. Normalized data model

**Teaching Notes**

These are the steps that are followed in the case studies that accompany the text.

**Slide 35**

**JRP and Interview Questions for Data Modeling**

<table>
<thead>
<tr>
<th>Purpose</th>
<th>Leading Questions (see textbook for a more complete list)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Discover system entities</td>
<td>What are the subjects of the business?</td>
</tr>
<tr>
<td>Discover entities</td>
<td>What unique characteristics (or characteristics that illustrate a category of each subject from other instances of the same subject?)</td>
</tr>
<tr>
<td>Discover entity subsetting</td>
<td>For these any characteristics of a subject that divide all instances of the subject into useful subsets?</td>
</tr>
<tr>
<td>Discover relationships</td>
<td>What are the relationships between entities?</td>
</tr>
<tr>
<td>Discover data timing patterns</td>
<td>What are the timing patterns of the data?</td>
</tr>
<tr>
<td>Discover entity uniqueness</td>
<td>What are the unique characteristics of the subject?</td>
</tr>
<tr>
<td>Discover data security and control</td>
<td>How are the changes to the data controlled and monitored?</td>
</tr>
</tbody>
</table>

**Teaching Notes**

Regardless of whether you use JRP, interviewing, or any other approach for information gathering, these are good questions to ask. Ask students to suggest other questions they could ask. It will help them and you make sure they understand the concepts and their real-world application.

Students can ask themselves these questions as they walk through data modeling for class assignments.
Slide 36

**Automated Tools for Data Modeling**

*Teaching Notes*
You could substitute a screen shot from the modeling tool you use in your class. If your classroom has computer projection capabilities, you could demo the modeling tool.

---

Slide 37

**Entity Discovery**

- In interviews or JRP sessions, pay attention to key words (i.e. "we need to keep track of ...").
- In interviews or JRP sessions, ask users to identify things about which they would like to capture, store, and produce information.
- Study existing forms, files, and reports.
- Scan use case narratives for nouns.
- Some CASE tools can reverse engineer existing files and databases.

*Teaching Notes*
You cannot model data that you do not understand. Therefore, some kind of entity definition is crucial. Generally students learn data modeling only by practice. It is very useful to walk through the SoundStage example on this and the following slides. We recommend you then add other cases for in-class practice. The Projects and Research at the end of the chapter has several example cases. Or you can add one or more of your own.

---

Slide 38

**The Context Data Model**

*Teaching Notes*
The purpose of this slide is not just to show what a context data model looks like. Take the time to walk through the entities and relationships using information from the textbook as a guide.
Teaching Notes
Depending on your room conditions, you may want to break this diagram into two or more parts and resize each to insure readability. Discuss the issues regarding keys and codes from the textbook.

Teaching Notes
Depending on your room conditions, you may want to break this diagram into two or more parts and resize each to insure readability. Discuss the need for the generalization.

Teaching Notes
Depending on your room conditions, you may want to break this diagram into two or more parts and resize each to insure readability. Discuss attribute naming conventions, attribute discovery from forms, and other attribute issues. Use the information in the textbook as a guide.
What is a Good Data Model?

- A good data model is simple.
  - Data attributes that describe any given entity should describe only that entity.
  - Each attribute of an entity instance can have only one value.
- A good data model is essentially nonredundant.
  - Each data attribute, other than foreign keys, describes at most one entity.
  - Look for the same attribute recorded more than once under different names.
- A good data model should be flexible and adaptable to future needs.

Data Analysis & Normalization

Data analysis — a technique used to improve a data model for implementation as a database.
  - Goal is a simple, nonredundant, flexible, and adaptable database.

Normalization — a data analysis technique that organizes data into groups to form nonredundant, stable, flexible, and adaptive entities.

Normalization: 1NF, 2NF, 3NF

First normal form (1NF) — entity whose attributes have no more than one value for a single instance of that entity.
  - Any attributes that can have multiple values actually describe a separate entity, possibly an entity and relationship.

Second normal form (2NF) — entity whose nonprimary-key attributes are dependent on the full primary key.
  - Any nonkey attributes dependent on only part of the primary key should be moved to entity where that partial key is the full key. May require creating a new entity and relationship on the model.

Third normal form (3NF) — entity whose nonprimary-key attributes are not dependent on any other nonprimary key attributes.
  - Any nonkey attributes that are dependent on other nonkey attributes must be moved or deleted. Again, new entities and relationships may have to be added to the data model.
First Normal Form Example 1

Teaching Notes
The repeating attributes are moved to a separate entity that has a one-to-many relationship to the original entity.

First Normal Form Example 2

Teaching Notes
The repeating attributes are moved to a separate entity that has a one-to-many relationship to the original entity.

Second Normal Form Example 1

Teaching Notes
The attributes ordered-product-description and ordered-product-title do not describe the primary key (order-number) of Member Ordered Product. They describe Merchandise and Title instead.
Second Normal Form Example 2

Teaching Notes
This could be a good time to bring out the old saw, "The key, the whole key, and nothing but the key."

Third Normal Form Example 1

Derived attribute – an attribute whose value can be calculated from other attributes or derived from the values of other attributes.

Teaching Notes
Some students also might see Purchased-Unit-Price as a derived attribute since it can be derived from Suggested-Retail-Price of the PRODUCT entity. This is a useful concept to discuss in class.
The reason why Purchased-Unit-Price must exist in MEMBER ORDERED PRODUCT is that while it can initially be derived from the PRODUCT entity, one would not be able to derive it at a later time if there were a price change. So these two attributes have subtly different definitions. One is the current price, which may change. The other is the price use for that particular order, which should not change.

Third Normal Form Example 2

Transitive dependency – when the value of a nonkey attribute is dependent on the value of another nonkey attribute other than by derivation.

Teaching Notes
Member Name and Member Address are dependent on Member Number, which is a foreign key to the Member entity. So they are moved to the Member entity.
SoundStage 3NF Data Model

Teaching Notes
Edit this slide as needed to ensure readability or refer students to the textbox.

Data-to-Location-CRUD Matrix

No additional notes
**Answers to End of Chapter Questions and Exercises**

**Review Questions**

1. Logical shows what a system is, or does. It is implementation-independent. Physical shows what the system is, or does, but also shows how it is physically or technically implemented. Physical models are thus implementation-dependent.

2. An implementation-independent model of a system allows the design of the system to remain stable and usable even when the technology supporting the system changes.

3. An implementation-dependent view of a system depicts available technology choices, and the limitations of those choices.

4. Entity: It is a class of persons, places, objects, events, or concepts about which we need to capture and store data. (I.e. think of it as a noun, or a thing such as a ‘student’ or ‘bird.’)

5. A descriptive property or characteristic of an entity.

6. A student is taught by a teacher, or conversely a teacher teaches a student. While the relationship itself is not affected by the number of classes taught or taken, the cardinality is affected by the number of classes a student can take (e.g. If a student may take only one class, the cardinality of the relationship is different than if the student can take many classes.)

7. Cardinality is the minimum and maximum number of occurrences of one entity that can be related to a single occurrence of another entity. An example would be: A student can have one or many teachers.

**Problems and Exercise**

1. Text (12)

   Note: Allowing more than 12 characters means that the database will catch the full last names of students with more than 12 characters in length. But, you must weigh the length of the average last name against the need to capture all of every last name. That is, increasing the domain size to say, 20 characters will allow students with names 19 characters long to be entered in their entirety. But, if the majority of last names have 5 characters, then there will be a lot of wasted space taken in the database (as the database must reserve that character length whether it is used, or not).
2. ‘required’

Note: No value in the last name domain is common enough to set as the default. Since this attribute would usually be required in a database, set the default to ‘required.’

3. ‘none given’

Note: This one is tricky. As a system designer, you must be sensitive to issues for which you may not be familiar. In the case of gender, a simple M or F is not permissible. A person may choose not to reveal their gender, or may have gender-identity issues that preclude them from answering. Thus, ‘none given’ or a similar default is best.

4. Table will contain: Student_ID, First_Name, Last_Name, Home_Phone, Work_Phone, Emergency_Phone, and Major.

Note: Name and Phone_Number were not atomic. Thus, they needed to be reduced to meet the requirements of 1st NF.

5. Movie_Identifying_Number, Title, Release_Date, Rating, Lead_Actor, Lead_Actress, Producer

Note: Above are examples. Can you think of others?

6. Many to many relationships are resolved using entities rather than associative entity, when the many to many relationship is caused by a forgotten entity in the data model.

7. See page 305 for an example of resolving involving entities and associative entities. Remember to use the associative entity when all business entities have been correctly included in the diagram. Otherwise, use an entity. Recall that some many to many relationships occur because an important entity has been forgotten in the model.

8. 1st NF: All attributes must be atomic (no multivalued or composite attributes)

Example: Student(Name(First_Name, Last_Name), {phone number})

Becomes: Student(First_Name, Last_Name, Home_Phone, Work_Phone, Cell_Phone)

2nd NF: 1st + all attributes must be fully functionally dependent

Consider a table where there is a joint primary key of Student_Id and Class_Id for a student’s identifying number and the identifying number of a
class at a college. An appropriate attribute in the table would be: Student_Grade: i.e. a student’s grade in a particular class. Note that the attribute is both dependent upon the Student_Id and the particular class, or Class_Id. An attribute for Teacher_Name would not be appropriate, since the teacher of a particular class is dependent only upon the Class_Id and not the Student_Id.

3rd NF: 2nd + no transitive dependencies

Example: A table with Student_Id as the primary key has an attribute of Date (for entrance into the program). Also included in the table is an attribute for Weather. Since weather is dependent on the Date, and Date is dependent on the primary key, there is a transitive dependency. Move Weather to another table with a primary key of Date.

9.

10. See pg. 279 for example of Identifying and Non-Identifying Relationships
And see pg. 277 for an example of a Ternary Relationship

11. Data modeling is much like drawing a picture of a database. A creative person will create a model that incorporates entities and attributes that are advantageous to the company as well as non-intrusive to the customer. A good modeler will balance the need for information with the realities of constraints on getting, keeping and maintaining that information. All of this takes creativity.

12. Absolutely, yes. A well designed database offers management strategic information about such things as product usage and marketing possibilities, as well as supply and inventory issues – this cuts costs as well as enables management to exploit new opportunities. For employees that interact directly with customers, a well designed database can enable the employee to offer superior customer service, individualized attention and enable them to resolve problems more easily. This increase in customer service will help the company keep returning customers.

Copyright © 2007 by McGraw-Hill Companies, Inc.
**Projects and Research**

1. Expect students to find that their student ID, name, birth date, major, past due fees, current checked out books, and probably a recent history of books checked out. There may also be a home phone number and address kept in the database for the collection of overdue books. Most probably, students will not come across anything unusual or surprising.

   Note: Allow and encourage students to question what they find. Would they do anything differently? Is there something missing that they think should be kept in the database? Is there something kept that does not seem necessary? Etc.

2. Student results will vary by school, and expect that answers will vary slightly between students. This is OK. Expect entities such as were outlined in problem 1. Remember when grading:
   - 1st NF: All attributes must be atomic (no multi-valued or composite attributes)
   - 2nd NF: 1st + all attributes must be fully functionally dependent
   - 3rd NF: 2nd + no transitive dependencies

3. Expect students to find common entities such as customer information, address/zip code, and transaction information such as items purchased, quantity of each item, time and date of the transaction, payment information, cashier information, and the use of any coupons. Consider this to be the bare minimum of a good information system for this case. Encourage students to interview professionals in marketing, accounting (accounts receivables, payroll, etc), business, and grocery store managers to find out other entities that would enable a grocery store company to "leap ahead" of the competition.

   Consider some follow-up questions to pose (in class) to your students:

   - What is the target market (audience) of the particular store they chose? This will influence not only the specific need for information, but also what is done with it. Does the store market to high end customers who are willing to pay a higher price for products, but want gourmet or custom orders? What about stores that cater to a more price-conscious crowd? How can the information system be used to cut costs? Encourage students to be innovative in their responses and to not be constrained by the "way things are usually done" in a grocery store.

4. Expect students to find some difference between their answers. This illustrates the uniqueness of the individual and the different approaches that can be taken to address the same problem. Do, however, expect a reasonable commonality among answers to include basic information such as: cus-
customer information, address/zipcode, and transaction information such as items purchased, quantity of each item, and the use of any coupons, etc. Encourage students to share with each other why they chose to keep information that they did: for example a student who is a marketing major may think of things that an MIS student or an Accounting student might not, and vice versa.

5. Examples:
   - California SB 1386 – organizations must provide notice to data subject if there has been a breach of personally identifiable information
   - For grocery stores with pharmacies: Health Insurance Portability and Accountability Act (HIPAA) – patient has the right to access and control of health record usage and the burden of privacy protection is placed on the provider
   - Gramm-Leach-Billey Act – disclosure of privacy policies in financial settings (include if grocery stores handle financial information)

6. There is no correct answer for this problem, and no incorrect answer. It is an exercise in collecting information, analyzing it, and thinking creatively and analytically to find a solution. Expect students to need some guidance along the process. Don’t be discouraging, but also don’t supply answers. In this case, the process is as important as the result.

   The students may have to get input from users or customers, as well as management. Expect them to determine whether they need this, and act accordingly. They should consider different methods for questioning people – interviews, questionnaires, etc. Also, quiz them on how they dressed and behaved when they went to this business. How does dress and behavior impact the information they can gleam out of people?

   Students should research what the competitors have done. Is there something that the competitors have done that might be useful? Something that they have not done, which presents a business weakness? Expect students to have done a SWOT analysis (strength, weakness, opportunity, and threat).

**Mini-Cases**

1. Step 1: Students should report that they will need to research competitors, do interviews, questionnaires, and surveys, as well as collect forms and reports that the company generates. They should also report that they will need to look at historical information as relevant.

   People/Departments that they should be interacting with: customers, stakeholders in management, management head and users in: shipping, accounting, marketing, and customer service, as well as existing IT personnel as relevant.
Step 2: Students should administer at least two of the three possibilities of surveys, questionnaires and forms. Students should also have included both closed and open-ended questions and enable anonymous answers to the closed questions on the surveys. Among the standard questions querying the users’ needs, students should also ask about likes and dislikes in computer software, interface design, and such. This will give the students an idea not just what the employees need from the system, but also an idea what they want from it.

Step 3: Confirm that students have reviewed their work and revised as necessary.

Step 4: Compare the responses to the surveys, etc. and the forms with the list that the students have submitted. Confirm that all entities that should be included are, and no irrelevant entities have been included. Then, review the attributes. Between groups, expect a bit of differentiation about what entities are included, and even more difference in attributes. This is normal and consistent with the students’ individuality, and difference in education and life experiences. Do not require identical answers – but do encourage the students to quiz each other as to why they chose the attributes they did.

Step 5: The students should report that they chose an implementation non-specific data model. Recall that the problem statement said that the CIO had not decided what hardware or software she wanted to utilize.

Step 6: Remember when grading:
  1st NF: All attributes must be atomic (no multi-valued or composite attributes)
  2nd NF: 1st + all attributes must be fully functionally dependent
  3rd NF: 2nd + no transitive dependencies

And: Many to many relationships are resolved using entities rather than associative entity, when the many to many relationship is caused by a forgotten entity in the data model.

2. This is a different problem statement, but the same as problem #1 in approach. Implicitly, expect the students to follow each of the steps in problem #1. Try to give them as much freedom as you can in exploring how to accomplish the task at hand. On the same narrative, encourage them to “strike out on their own” first and then come to you with what they are finding or discovering. Recall, creativity is a messy process. The end result should be a report/narrative that is thoughtful, and clearly researched. Students should display that they have sought out information from literature, historical data, users and management to ascertain the information that they needed to develop their data model.
They should also be sure to include departments such as: accounting, marketing, customer service, and service repair departments, as well as inventory management such as purchasing and selling as needed for the new and used cars.

Entities should include customers, cars, employees, and students should consider car tracking through the rental process, as well as wear and tear from renters, stolen cars, selling of used cars, customer service issues for marketing (e.g. what kind of car does this person prefer? Any equipment needs? Etc.), as well as customer tracking and payment information.

3. This problem is included to capture the students’ attention. Insert “mafia” instead of organized crime occasionally and sprinkle in discussions of the “Sopranos.” The students should analyze security measures for database contents, as well as determining entities and attributes. At what layer should a database be encrypted? What sort of issues arise when information from the database needs to be sent to another user?

   Note: Have them pretend to be a “Boss.” How would they keep track of their business and keep the feds off their tail? Usually, there will be a comedian in the class (look to your ‘back row hecklers’) who can do a good impersonation. Encourage them to run with it. I.e. turn this problem into a light-hearted game.

4. Legal and Privacy Examples:
   • California SB 1386 – organizations must provide notice to data subject if there has been a breach of personally identifiable information
   • If the database uses health information or interacts with pharmacies or hospitals: Health Insurance Portability and Accountability Act (HIPAA) – patient has the right to access and control of health record usage and the burden of privacy protection is placed on the provider
   • Gramm-Leach-Billey Act – disclosure of privacy policies in financial settings (include if the database handles financial information)
   • Safe Harbor Act: extends privacy protection for members of the European Union
   • Council of Europe Convention on Cybercrime – adopted to ensure congruency among laws among joining countries
   • G-8 Subgroup of High-Tech Crime – no criminal safe havens
   • National Institute of Standards and Technology (NIST) – recommendations for documentation of company wireless policies and information collected and stored

Ethical examples:
   • Stockholder, Stakeholder and Social Welfare standpoints
   • Slippery Slope
   • Golden Rule
Team and Individual Exercises

1. Step one: Research cultural, communications, religious, and time zone issues associated with a team makeup from the countries you have chosen.

   Step two: Each member takes on the attributes of a person from his or her “country.” This includes “workdays” timelines, labor rules, potential political issues, etc.

   Step three: Complete a simple design and program development for a cash register. Remember, all team members must work as if they were working within their “country.”

   Advice: Utilize email as much as possible, as members can check it during their normal “work day.” Phone conference calls will probably require that at least one person has to “get up” in the middle of the night – take turns!

   Step four: What did you learn? Prepare thoughts for a roundtable class discussion, and a brief written summary to hand in with your program.

2. There is no answer to this question.

3. There is no answer to this question.