1

INTRODUCTION TO DATA MODELING

1

DATA MODELING: AN OVERVIEW

CHAPTER OBJECTIVES

- Introduce the process of data modeling
- Present why data modeling is important
- Explain how a data model represents information requirements
- Describe conceptual, logical, and physical data models
- Briefly discuss the steps for building a data model
- Show the role of data modeling in system development
- Provide an initial glimpse of data modeling history and trends

James Watson and Francis Crick, working at Cambridge University, deduced the three-dimensional structure of DNA (deoxyribonucleic acid). In 1953, they published a brief paper describing their now-famous double helix model of DNA. This important milestone of creating a true model of DNA gave a tremendous boost to biology and genetics. For the discovery and creation of the double helix model, Watson and Crick shared the Nobel Prize for Physiology and Medicine in 1962.

Well, what does Watson and Crick's achievement have to do with our current study? Essentially, they built a model. The model is a true representation of the structure of DNA—something we find in the real world. Models are replicas or representations of particular aspects and segments of the real world. Building of models is quite common in many disciplines. When you think about it, the representation "$5 + 4 = 9$" is a mathematical model using symbols and logic. This model represents the fact that if you put five things together with four things of the same kind, you get nine things of the same kind. In physics, we create models to represent physical properties of the world. In economics, we create models of economic trends and forecast economic outcomes.

Data Modeling Fundamentals. By Paulraj Ponniah

Let us get a more vivid picture of what we mean by a model. Let us say that you are interested in buying a new home in one of the upcoming posh developments. You go to the sales office of the real estate developer. They point to a large open site where they plan to build the houses and to complete the development in 3 years. Right now, you cannot see any houses; you cannot observe the layout of roads and houses; all you can notice is a lot of vacant space and numerous trees. How can you get a picture of how the development will look in the future? How can you imagine how the house you want to buy will be structured? While you appear puzzled, the sales staff leads you into a large room. On a big table, in the middle of the room, sits a scale model of the development. They had created the model based on the requirements of the future homeowners in the community. You see roads, houses, swimming pools, tennis courts, and other amenities in the future development. These are not the real roads and houses. These are just components in the model. The model is a true representation of the real estate development. The sales people are able to point to different components in the model and communicate with you clearly and vividly. Now, you are able to understand, and you are happy.

A model serves two primary purposes:

- As a true representation of some aspects of the real world, a model enables clearer communication about those aspects of the real world.
- A model serves as a blueprint to shape and construct the proposed structures in the real world.

DATA MODEL DEFINED

Data modeling is an integral part of the process of designing and developing a data system. While designing and developing a data system for an organization, you take into account all the information that would be needed to support the various business processes of the organization. If you are designing a data system for a banking institution, you have to provide data for the business processes of checking, savings, and loan account operations. If you are creating a data system for a medical center, you have to provide data for inpatient and outpatient services. You start with the analysis and gathering of details about which data elements would be needed for the business. You need to ensure that the results of *requirements definition* are completely implemented as the data content of the database that would support the organization.

You start with planning and requirements definition. Based on these, you have to come up with the proper database system. During this process, you have to keep on communicating with the business stakeholders about the data elements, their structures, relationships among the structures, and the rules governing the structures and relationships. You have to make sure that these data elements are exactly the ones that are needed to support the business. The users must be able to understand clearly what you are designing and give their affirmation.

Making the users understand the information content of the database system being built is one crucial aspect of the development process. The other significant aspect of the development process is your ability to create a database system that meets the information requirements exactly and conforms to what you have presented and described to your users. As database practitioners, what technique can we adopt to achieve these dual goals? How can

we communicate with the users and keep them informed? How can we do what we promise to deliver and meet the information requirements exactly?

What Is a Data Model?

Data modeling provides a method and means for describing the real-world information requirements in a manner understandable to the stakeholders in an organization. In addition, data modeling enables the database practitioners to take these information requirements and implement these as a computer database system to support the business of the organization.

So, what is a data model? A data model is a device that

- helps the users or stakeholders understand clearly the database system that is being implemented based on the information requirements of an organization, and
- enables the database practitioners to implement the database system exactly conforming to the information requirements.

A data model, therefore, serves as a critical tool for communication with the users; it also serves as a blueprint of the database system for the developers. Figure 1-1 illustrates these two significant aspects of a data model. Notice how the data model serves the needs of the two groups: users and developers. Also notice the place of the data model between requirements definition and database system implementation.

Data modeling is a technique for exploring the data structures needed to support an organization. A data model must record and indicate the content, shape, size, and rules of the data elements used throughout the scope of the various business processes of the organization. It would be a conceptual representation or replica of the data structures required in the database system. A data model focuses on what data is required and

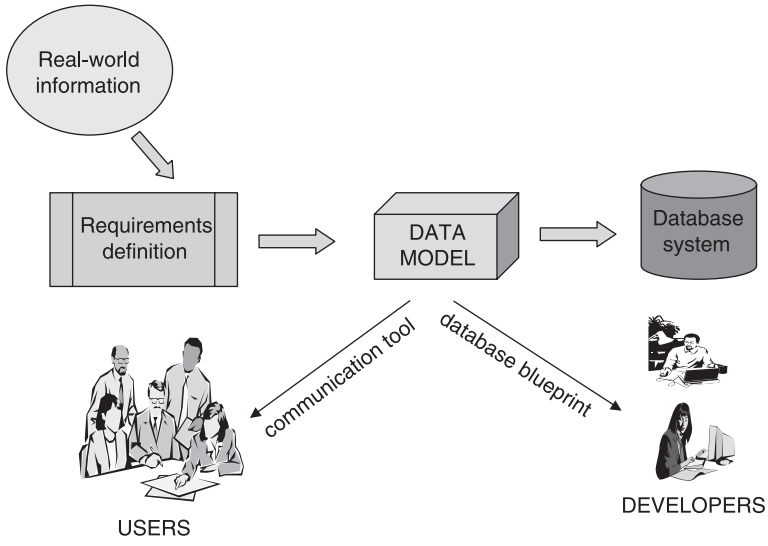


FIGURE 1-1 Data model: communication tool and database blueprint.

how the data should be organized. It does not necessarily reflect the operations expected to be performed on the data.

Data modeling can be applied to representation of the information requirements at various levels. At the highest conceptual level, the data model is independent of any hardware or software constraints. At this level, the data model is generic; it does not vary whether you want to implement an object-relational database, a relational database, a hierarchical database, or a network database. At the next level down, a data model is a logical model relating to the particular type of database—relational, hierarchical, network, and so on. This is because in each of these types, data structures are perceived differently. If you proceed further down, a data model is a physical model relating to the particular database management system (DBMS) you may use to implement the database. We will discuss these levels further.

Why Data Modeling?

You have understood that a data model is created as a representation of the information requirements of an organization. You have also noted that a data model functions as an effective communication tool for discussions with the users; it also serves as a blueprint for the database system. A data model, therefore, acts as a bridge from real-world information to database storing relevant data content.

But, why this bridge? Why not go from real-world information to the database itself? Let us take a simple example. A business sells products to customers. We want to create a database just to support such sale transactions, nothing more. In our database, we need to keep data to support the business. In the real world of this business, data exists about customers, products, and sales of products to customers. Now, look at Figure 1-2, which shows these data elements for the business.

The method for showing the information requirements as indicated in the figure is haphazard and arbitrary. If you are asked to depict the information requirements, you might do

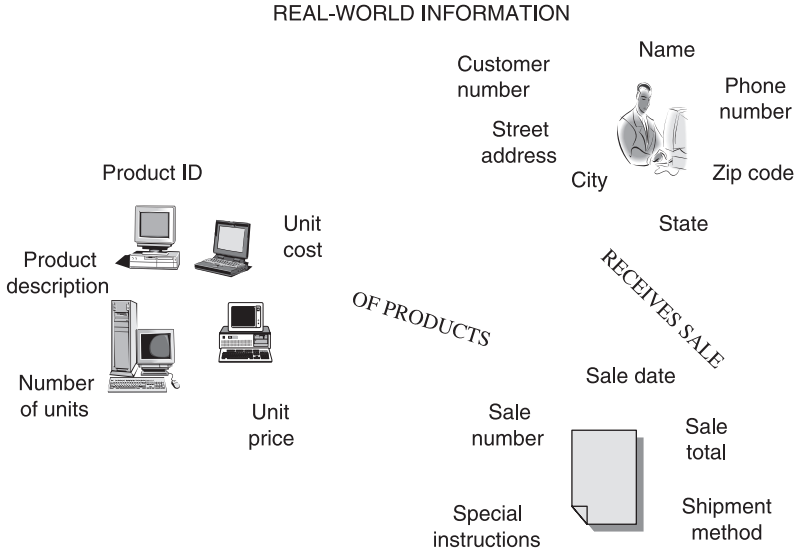


FIGURE 1-2 Sales: real-world information requirements.

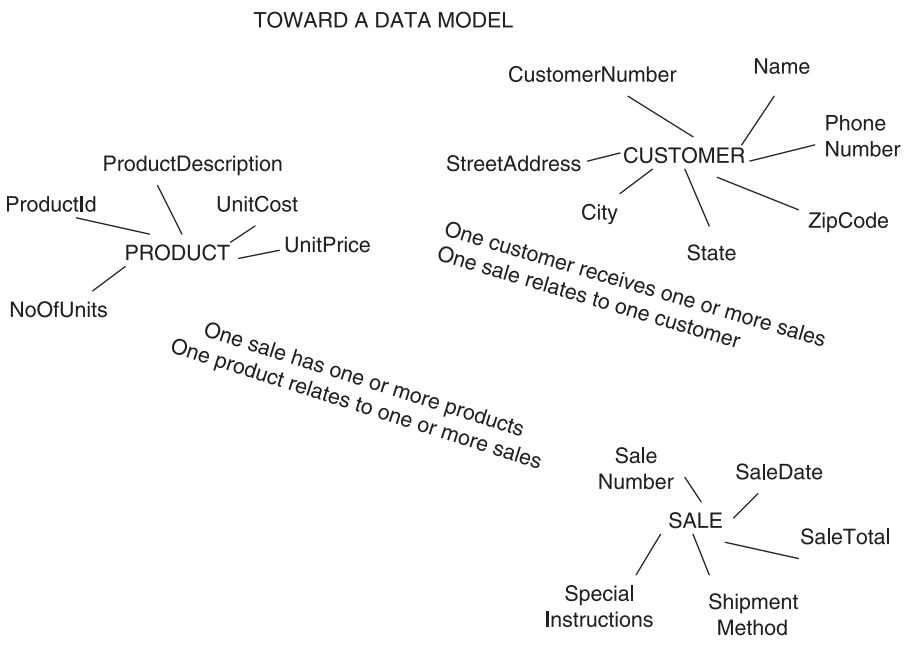


FIGURE 1-3 Sales: a step toward a data model.

it in a different way. If someone else does it, that person might do it in yet a different way. Now consider the database to be built to contain these data elements. Two overall actions need to be performed. First, you have to describe the database system to the users and obtain their confirmation. Then you have to create the database system to provide the necessary information. The depiction of information requirements as shown in the figure falls short of these expectations.

Let us try to improve the situation a bit. Let us try to depict the information requirements in slightly better and more standard manner. See Figure 1-3, where the depiction is somewhat clearer.

This figure presents a picture that could better help us to communicate with the users with a little more clarity and also enable us to proceed with the implementation of the database system. Now you can show the users the business objects of CUSTOMER, PRODUCT, and SALE about which the database system will contain data. You can also point to the various pieces of data about these objects. Further, you can also explain how these objects are related.

Still, this depiction falls somewhat short of the expectations. This figure is an attempt toward a good data model. When we take the depiction a few steps further and create a satisfactory data model—a true representation of the information requirements—we can achieve our goals of user communication and database blueprint. But that is not all. A data model serves useful purposes in the various stages of the data life cycle in an organization. Let us see how.

Data Life Cycle. Follow the stages that data goes through in an organization. First, a need for data arises to perform the various business processes of an organization. Then

a determination is made about exactly what data is needed. Gathering of the data takes place. Then the data gets stored in the database system. In the next stage, data is manipulated by reading it from storage, combining it in various desired ways, and changing it. After a while some of the data gets archived and stored elsewhere. After some of the data completes its usefulness, the corresponding data elements get deleted from the database system. Figure 1-4 presents the stages in the data life cycle of an organization and also the interaction with the data model at the different stages.

Now let us walk through the various stages of the data life cycle. At each stage, we will note how a data model is helpful and serves useful purposes.

Needing Data. In this earliest stage, an organization recognizes the need for data for performing the various business processes. For example, to perform the process of taking orders, you need data about products and inventory. For producing invoices, you need data about orders and shipments. Thus, this stage in the data life cycle recognizes the need for data in the organization. At this stage, a high-level conceptual data model is useful to point to the various business processes and the data created or used in these processes.

Determining Needed Data. Once you recognize the need for data, you have to determine which data elements are needed for performing business processes. At this stage, you will come up with the various types of data, which data is really needed and which data would be superfluous, and how much of each type of data is needed. At this stage,

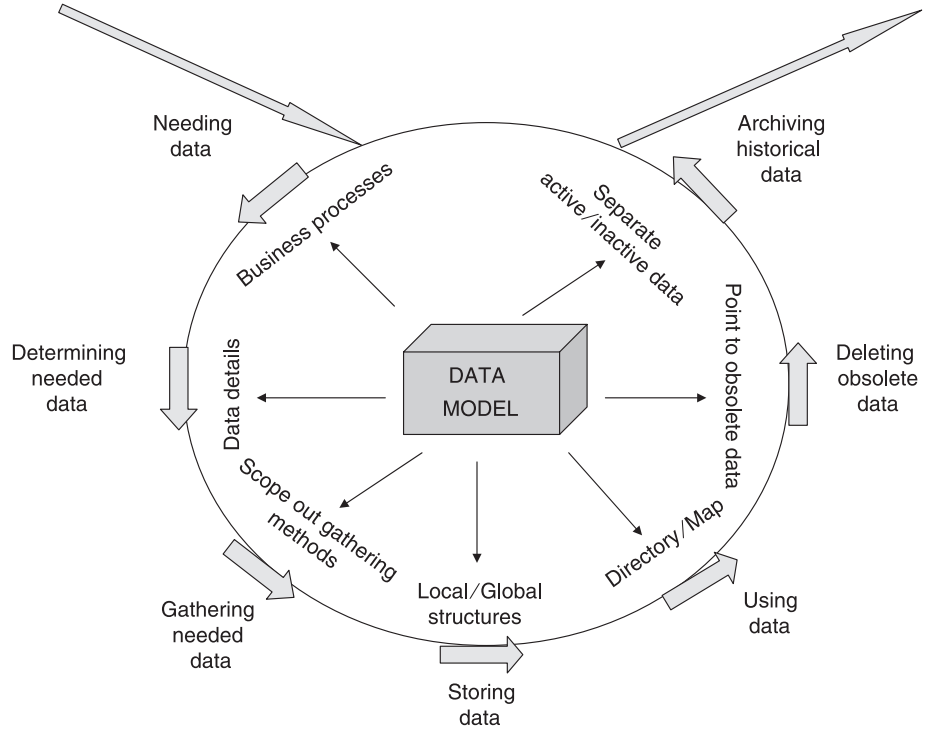


FIGURE 1-4 Organization's data life cycle.

all the required details of the needed data elements are discovered and documented in the data model.

Gathering Needed Data. After the determination of which data is needed, collection of data takes place. Here you apply a sort of filter to gather only the data that is needed and ignore the irrelevant data that is not necessary for any of your business processes. You will apply different methods of data creation and data gathering in this stage. The data gathering trials and methodologies are scoped out with the aid of the data model.

Storing Data. The collected data must be stored in the database using appropriate methods of storage. You will decide on the storage medium and consider the optimal storage method to suit the needs of users for accessing and using data. The data model in this stage enables you to assemble the components of the global data repository. Each part of the data model determines a specific local data structure, and the conglomeration of all the parts produces the global structure for data storage.

Using Data. Data, collected and stored, is meant for usage. That is the ultimate goal in the data life cycle. At this stage, you will combine various data elements, retrieve data elements for usage, modify and store modified data, and add new data created during the business processes. At this stage, the data model acts as a directory and map to direct the ways of combining and using data.

Deleting Obsolete Data. After a while, a particular data element in storage may become stale and obsolete. After a period of time, the data element may no longer be useful and, therefore, not accessed in any transactions at all. For example, orders that have been fulfilled and invoiced need not remain in the database indefinitely beyond the statutory time of legal and tax reporting purposes. An organization may decide that such orders may be deleted from the database after a period of 10 years. Deleting obsolete data becomes an ongoing operation. A particular data element may fall into the category qualifying for deletion. At this stage, the data model is used to examine the various data elements that can be safely deleted after specified periods.

Archiving Historical Data. However, some data elements may still be useful even long after any activity on those data elements had ceased. Data relating to customer purchases can be useful to forecast future trends. Historical data is useful in the organization's data warehouse. Any such useful data elements are removed from the current database and archived into a separate historical repository. The data model in this stage provides the ability to point to the original and final spots of data storage and trace the movement from active to archived repositories.

Who Performs Data Modeling?

In a database project, depending on the size and complexity of the database system, one or more persons are entrusted with the responsibility of creating the data models. Data models at various levels call for different skills and training. Creating a conceptual data model involves capturing the overall information requirements at a high level. A logical data model is different and is meant for different purposes. A physical data model, on the other hand, pictures the information at the lowest level of hardware and physical

storage. So, who performs data modeling? Data modeling specialists with appropriate training, knowledge, and skills do the work of data modeling.

However, the recent trend is not to employ persons having data modeling skills alone. This is an age of generalizing specialists. Data modeling is usually an additional set of skills acquired by certain persons on the database project. These generalists are trained in the principles and practice of data modeling and assigned the responsibility of creating the data models.

Who Are the Data Modelers? This is another way of asking the same question. In an organization, who are these folks? What functions do they perform? How can we think of the various tasks performed by the data modelers? Are they like architects? Are they like librarians? Are they like document specialists?

The primary responsibility of data modelers is to model and describe that part of the real world that is of interest to the organization to achieve its goals and purposes. In doing so, a data modeler may be thought of performing the following functions.

Scanning Current Details. The data modeler scans and captures details of the current state of the data system of the enterprise. New models are built by looking at the current data structures.

Designing the Architecture. The data modeler is an architect designing the new data model. He or she puts together all the pieces of the architecture.

Documenting and Maintaining Meta-Data. The data modeler is like a librarian and custodian of the data about the data of the organization. The data modeler is also a tremendous source of information about the data structures and elements, current and proposed.

Providing Advice and Consultation. With in-depth knowledge about the composition of the data system of an organization, the data modeler is the expert for consultation.

INFORMATION LEVELS

By now, it is clear to you that a data model is a representation of the information requirements of an organization. A data model must truly reflect the data requirements of an enterprise. Every aspect of the data for the company's business operations must be indicated clearly and precisely in the data model. As we defined a data model, we also considered the two major purposes of a data model. A data model serves as a means for communication with the users or domain experts. It is also a blueprint for the proposed database system for the organization.

Let us examine the first purpose. A data model is a tool for communication with the users. You will use the data model, review its components, describe the various parts, explain the different connections, and make the users understand the ultimate data system that is being built for them. The data model, therefore, must be at a level that can be easily understood by the users. For this purpose, the data model must be devoid of any complexities. Any complexity in terms of the data structures must be hidden from the users. In the data model, there can be no indication of any physical storage considerations. Any reference to how data structures are laid out or perceived by analysts and

programmers must be absent from the model. The data model must just be a conceptual portrayal of the information requirements in human terms. The data model must be a representation using a high level of ideas. The primary purpose here is clear communication with the domain experts.

Now let us go to the second major purpose of a data model. The data model has to serve as a blueprint for building the database system. In this case, the database practitioners must be able take the data model, step through the components, one by one, and use the model to design and create the database system. If so, a data model as a representation at a high level of ideas is not good enough as a blueprint. To serve as a blueprint, the data model must include details of the data structures. It should indicate the relationships. It should represent how data is viewed by analysts and programmers. It should bear connections to how database vendors view data and design their database products.

In order to build the database system and determine how data will be stored on physical storage and how data will be accessed and used, more intricate and complex details must be present in the data model. This is even more detailed than how data is viewed by programmers and analysts.

So, we see that a data model must be at a high and general level that can be easily understood by the users. This will help the communication with the users. At the same time, we understand that the data model must also be detailed enough to serve as a blueprint. How can the data model serve these two purposes? At one level, the data model needs to be general; at another level, it has to be detailed. What this means is that representation of information must be done at different levels. The data model must fit into different information levels. In practice, data models are created at different information levels to represent information requirements.

Classification of Information Levels

Essentially, four information levels exist, and data models are created at each of these four levels. Let us briefly examine and describe these levels. Figure 1-5 indicates the information levels and their characteristics.

Conceptual Level. This is the highest level consisting of general ideas about the information content. At this level, you have the description of application domain in terms of human concepts. This is the level at which the users are able to understand the data system. This is a stable information level.

At this level, the data model portrays the base type business objects, constraints on the objects, their characteristics, and any derivation rules. The data model is independent of all physical considerations. The model hides all complexities about the data structures from the users through levels of abstraction. At this level, the data model serves as an excellent tool for communication with the domain experts or users.

External Level. At the conceptual level, the data model represents the information requirements for the entire set of user groups in the organization. The data model is comprehensive and complete. Every piece of information required for every department and every user group is depicted by the comprehensive conceptual model. However, when you consider a particular user group, that group is not likely to be interested in the entire conceptual model. For example, the accounting user group may be interested in just customer information, order information, and information about invoices and

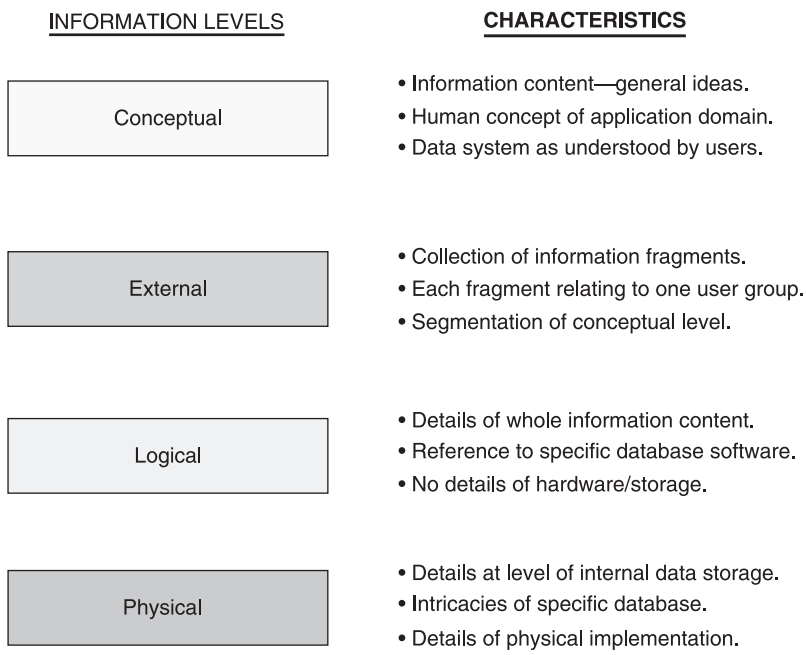


FIGURE 1-5 Information levels for data modeling.

payments. On the other hand, the inventory user group may be interested in only the product and stock information. For each user group, looking at the conceptual model from an external viewpoint, only a portion of the entire conceptual model is relevant. This is the external level of information—external to the data system. At the external level, portions of the entire conceptual model are relevant. Each user group relates to a portion of the conceptual model.

A data model at the external level consists of fragments of the entire conceptual model. In a way, each fragment is a miniconceptual model. If you consider an external data model, it contains representation of a particular segment of information requirements applicable to only one user group. Thus, if you create all the external data models for all the user groups and aggregate all the external data models, then you will arrive at the comprehensive conceptual model for the entire organization. External data model enables the database practitioners to separate out the conceptual data model by individual user groups and thus allocate data access authorizations appropriately.

Logical Level. At this level, the domain concepts and their relationships are explored further. This level accommodates more details about the information content. Still, storage and physical considerations are not part of this level. Not even considerations of a specific DBMS find a place at this level. However, representation is made based on the type of database implementation—relational, hierarchical, network, and so on.

If you are designing and implementing a relational database, the data model at this level will depict the information content in terms of how data is perceived in a relational model. In the relational model, data is perceived to be in the form of two-dimensional tables. So, a logical data model for a relational database will consist of tables and their relationships.

Data in the tables will be represented as rows and columns. The data model at the logical level will be used in the ultimate construction of the database system.

Internal or Physical Level. This information level deals with the implementation of the database on secondary storage. Considerations of storage management, access management, and database performance apply at this level. Here intricate and complex details of the particular database are relevant. The intricacies of the particular DBMS are taken into account at the physical level.

The physical data model represents the details of implementation. The data model at this level is primarily intended as a blueprint for implementation. It cannot be used as a means for communication with the users. The data model represents the information requirements in terms of files, data blocks, data records, index records, file organizations, and so on.

Data Models at Information Levels

When we began our discussion on data models, it appeared as if a data model is a single type of representation of information requirements for an organization. When we analyzed the purposes of a data model, it became clear that a single type of representation is not sufficient to satisfy the two major purposes. The type of representation that is conducive for communication with users does not have the lower level details needed for the model to serve as a blueprint. On the other hand, the type of representation with details about the data structure is necessary in a construction blueprint; but such a representation is not easy to be used as a communication tool with the users.

This has led to the need to create data models at different information models. We have understood the necessity for different types of representations for the different purposes. These are the data models at the various levels of information—conceptual data model, external data model, logical data model, and physical data model. Figure 1-6 shows the data models at the different information levels. Note the nature of the data model at each level and also notice the transition from one level to the next. The figure also indicates the purpose of the data model at each level.

Earlier we had developed an initial data model consisting of three business objects, namely, CUSTOMER, PRODUCT, and SALES. Let us use these three objects to illustrate data models at different levels. In the section of our real world, all the information we need is only about these three objects. For the purpose of illustrating the different data models, let us make this restrictive assumption and proceed. Also, we will assume that our ultimate database will be a relational database.

External Data Model. The external data model is a depiction of the database system from the viewpoints of individual user groups. This model may be used for communication with individual groups of users. Each individual user group is interested in a set of data items for performing its specific business functions. The set of data items relevant for a specific user group forms part of the external data model for this particular user group.

For the purpose of our example, let us consider three user groups: accounting, marketing, and inventory control. Try to figure out the data items each user group would be interested in. For the sake of simplicity, let us consider a minimum set of data items.

Figure 1-7 shows the set of data items each of these groups is interested in. This figure illustrates the formation of an external data model.

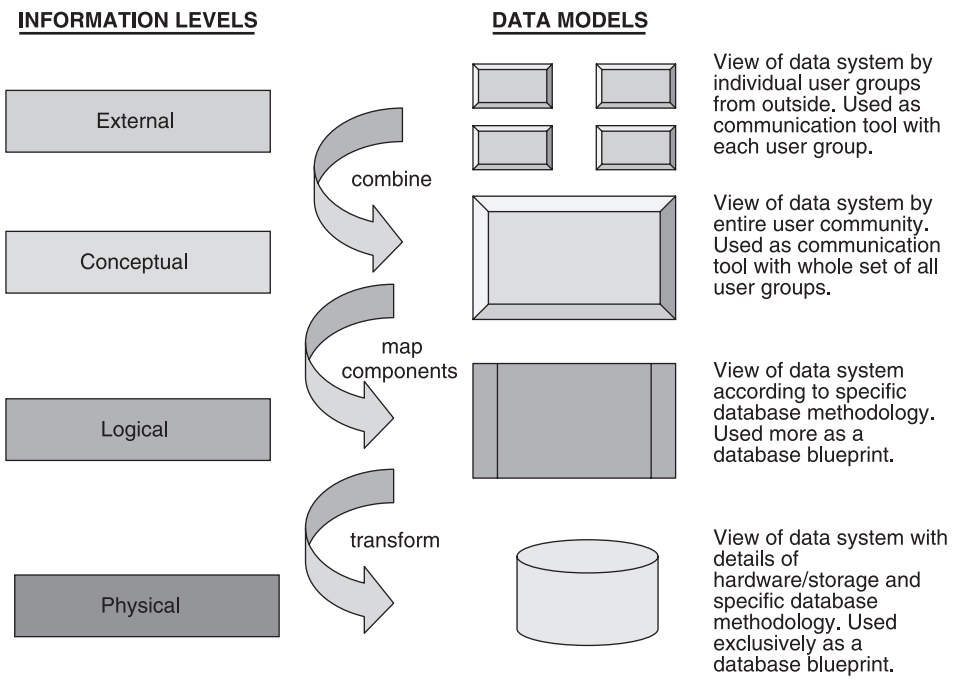


FIGURE 1-6 Data models at different information levels.

Conceptual Data Model. The conceptual data model is at a high and general level, intended mainly as a communication tool with the user community. In the model, there is no room for details of data structure or for any considerations of hardware and database software. This model does not even address whether the final database system is going to be implemented as a relational database system or any other type of database system. However, the model should be complete and include sufficient components so that it would be a true representation of the information requirements of the organization.

Figure 1-8 illustrates the idea of a conceptual data model. The information requirements we are considering relate to the data items for the user groups of accounting, marketing, and inventory control. That was the external data model shown in Figure 1-7. You see that the conceptual data model has representations for the three business objects of CUSTOMER, PRODUCT, and SALES. You can easily see the connection between the external data model and the conceptual data model. The figure also shows the intrinsic characteristics of these business objects—the data about these objects. Further, the conceptual model also indicates the relationships among the business objects. In the real world, business objects in an organization do not exist as separate entities; they are related with one another and interact with one another. For example, customer orders product, and products are sold to customers.

By looking at the figure, you would have noticed that for the conceptual data model to serve as a communication tool with the users, there must be some easily understood notations or symbols to represent components of the model. Some accepted symbol must indicate a business object; some notation must indicate the characteristics or attributes of a

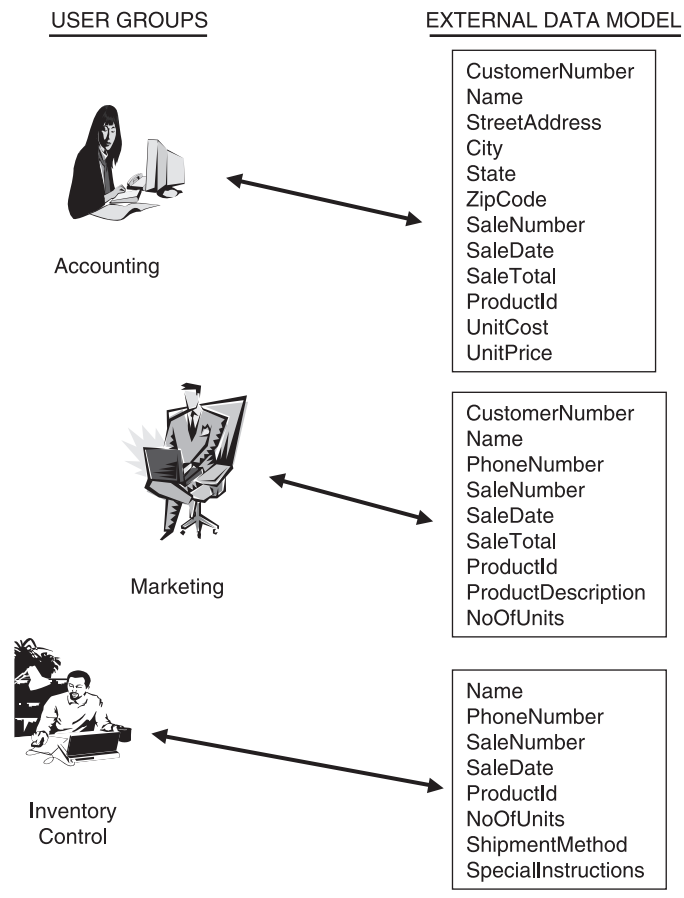


FIGURE 1-7 External data model.

business object; some representation must be made to show the relationship between any two objects. Over time, several useful techniques have evolved to make these representations. We will introduce some of the techniques at the end of this chapter. Further, Chapter 2 is totally dedicated to a discussion of data modeling methods, techniques, and symbols.

Logical Data Model. In a sense, the logical data model for an organization is the aggregation of all the parts of the external data model. In the above external data model, three user groups are shown. We assume that there are only three user groups in the organization. Therefore, the complete logical model must represent all the combined information requirements of these three user groups.

For the relational type of database system, the logical model represents the information requirements in the form of two-dimensional tables with rows and columns. Refer to Figure 1-9 for an example of the logical data model. At this stage, the figure just gives you an indication of the logical data model. We will discuss this concept a lot more elaborately in subsequent chapters.

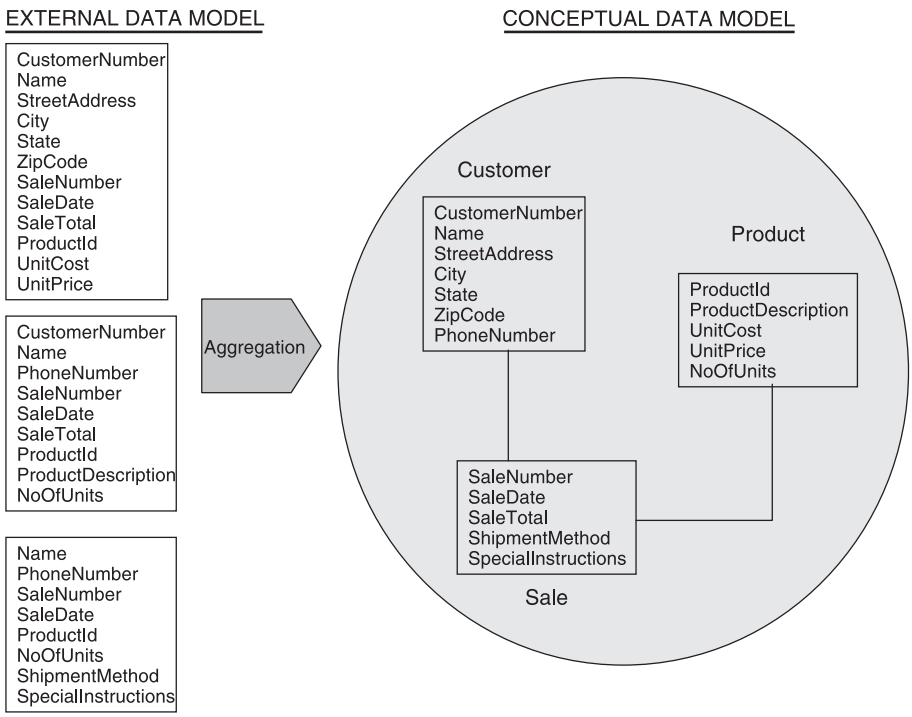


FIGURE 1-8 Conceptual data model.

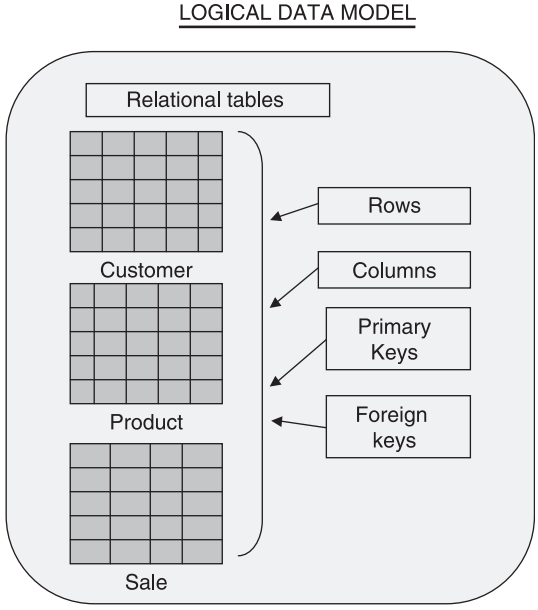


FIGURE 1-9 Logical data model.

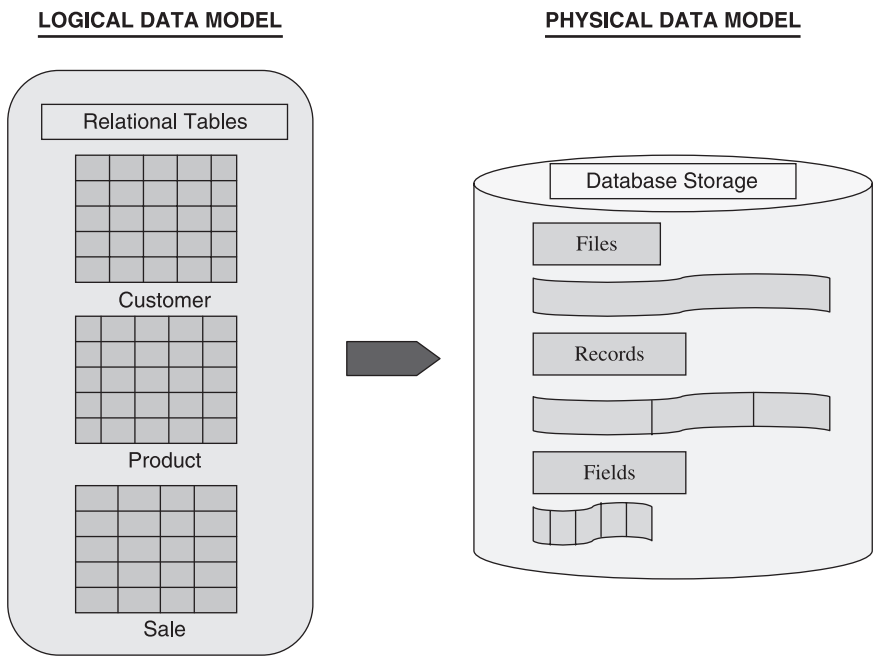


FIGURE 1-10 Physical data model.

As can be seen from the figure, the logical data model may serve both purposes—communication tool and database blueprint. In this case, it will serve as a blueprint for a relational database system along with the physical data model.

Physical Data Model. A physical data model has little use as a means of communication with the users. Its primary purpose is to act as a blueprint for the implementation of the database system. The details contained in a physical data model are beyond the normal comprehension of the users. The model expresses too many intricate details. It includes considerations of the particular DBMS and the hardware environment in which the database system gets implemented.

See Figure 1-10 for an example of the physical data model. Notice how the model represents the information requirements in terms of files, data blocks, records, fields, and so on. The model is a representation at the lowest level of abstraction with a lot of complex details.

CONCEPTUAL DATA MODELING

Having considered the different types of data models and their purposes, we are now ready to ponder the question how exactly is a data model created. What are the major steps? What are the various components that make up a data model? Let us get an initial introduction to the terminology, components, and the steps for creating a data model. Here we want to be brief and just introduce the topics. Part II covers the topics in elaborate detail with a comprehensive case study. So, let us now confine ourselves to getting a quick glimpse of the components and the steps.

For our purposes here, let us take a simple example to identify the components and the steps. We will deal with the conceptual data model because that is generic and is usually the first data model that is created. As you know, the conceptual data model has no considerations about the type of database system being implemented, no reference to the particular DBMS, and absolutely no concern about the storage and hardware environment where the ultimate data system will reside and perform. These are details that are deliberately kept out of the conceptual model. Later on in the following chapters, discussions will cover the logical and physical data models.

Again, as mentioned earlier, several standard techniques and notations exist for creating a data model. We will be discussing those in Chapter 2. For now, we will not get bogged down with specific techniques or symbols. We can use some meaningful and easily understood symbols for now. Remember the main goals at this stage: identification of the major components of a data model and overview of the major steps in the modeling process.

Data Model Components

Before proceeding further, let us introduce some terminology and identify the primary data model components. Our simple introductory data model will contain these components. At this early stage, we will not represent the components using any standard technique. That will come later. Let us just use very basic symbols to represent the components for now.

For the purpose of our simple data model, we will consider four basic components and define them. Most of the conceptual data models we come across in practice consist of these basic components. Remember, the data model reflects and represents the information requirements of an organization. What are the pieces of information a company or business is interested in? What are the parts of the information a company requires to run its business? Data model components are derived from such business considerations. Let us move on to identify the components.

Objects or Entities. When you analyze the information requirements of a company, you will notice that the company needs information about the business objects that are of interest to it. The company needs data about business objects. The organization needs to know how these business objects are related and the implications of such relationships.

What are such business objects? For example, a bank is interested in data about its customers and about checking, savings, and loan accounts. So, for a bank, CUSTOMER, CHECKING-ACCOUNT, SAVINGS-ACCOUNT, and LOAN-ACCOUNT would be examples of business objects. Similarly, for a hospital, PATIENT, PHYSICIAN, PROCEDURE, and HOSPITAL-VISIT would be examples of business objects. Each organization has its own set of business objects. A particular institution needs data about its own set of business objects. Data about the business objects or entities must be present in the data system for the organization to function.

Attributes or Characteristics. Consider a specific business object. Let us take the business object called CUSTOMER. What would this business object represent in a data model? The object will represent the set of all the customers of the organization. How can all the customers be grouped and represented by a single object called CUSTOMER? This is because all these customers mostly possess the same characteristics.

Each customer has an intrinsic characteristic known as *Customer Name*. Every customer has a specific name. Every customer has other inherent of intrinsic characteristics such as *Customer Address*, *Customer Phone Number*, *Customer Balance*, and so on. These intrinsic characteristics are known as attributes. The business object CUSTOMER has attributes of CustomerName, CustomerAddress, CustomerPhoneNumber, Customer Balance, and so on. Each individual customer will have distinct values for these attributes. Thus, attributes of an object or entity are representations of its inherent or intrinsic characteristics.

Identifiers. Let us get back to the definition of an object or entity. The object CUSTOMER represents all the customers of the organization. Each customer has a distinct set of values for the attributes of the object CUSTOMER. If the CUSTOMER object represents all and every customer, then how can we distinguish one customer from another represented through the object called CUSTOMER? Why do we need to make this distinction? Frequently in business situations, we need to find information about a single customer. Where does the customer live so that we may send invoices to the customer? What are the recent orders placed by that customer?

How can we distinguish one customer from another and indicate this distinction in the data model? We can use values of a particular attribute to make the distinction. You can say that you need information about that particular customer for whom the value of the CustomerNumber is 1234. In this case, CustomerNumber would be an attribute that can be used to distinguish one customer from another. Provided values of customer numbers are not duplicated in the data system, values of the attribute CustomerNumber can be used to make the distinction between customers. In such a case, the attribute CustomerNumber is an identifying attribute or an identifier for the object CUSTOMER. In practice, one or more attributes whose values would uniquely determine specific instances of an object are chosen as the identifier for that object.

Relationships. Let us get back to the example of business objects for a bank. Consider the objects CUSTOMER and CHECKING-ACCOUNT. These are two separate objects with their own attributes. For the purpose of running the banking business, the users in the bank need data about customers and checking accounts. They need the data about these two objects separately, but even more so about the relationship between customers and their checking accounts. For a particular customer, what is the status of the customer's checking account?

In a company's environment, business objects do not exist in isolation. They are related to one another. What is the relationship between CUSTOMER and CHECKING-ACCOUNT? Customers operate checking accounts. That is the relationship. For a data model to be a true representation of the real-world situation for a bank, the model must reflect the relationships among the business objects.

Simple Symbols. We have identified the primary components of a data model. We have introduced business objects, attributes, identifiers, and relationships. Let us assign some simple symbols to these components. Make a square box to represent a business object. Small circles connected to a square box may represent attributes. Two concentric circles may be used to represent identifiers. When two objects are directly related, let us indicate the relationship by joining the two square boxes by a straight line. For our initial discussion of the data modeling steps, these simple symbols are sufficient.

Figure 1-11 shows examples of these data model components.

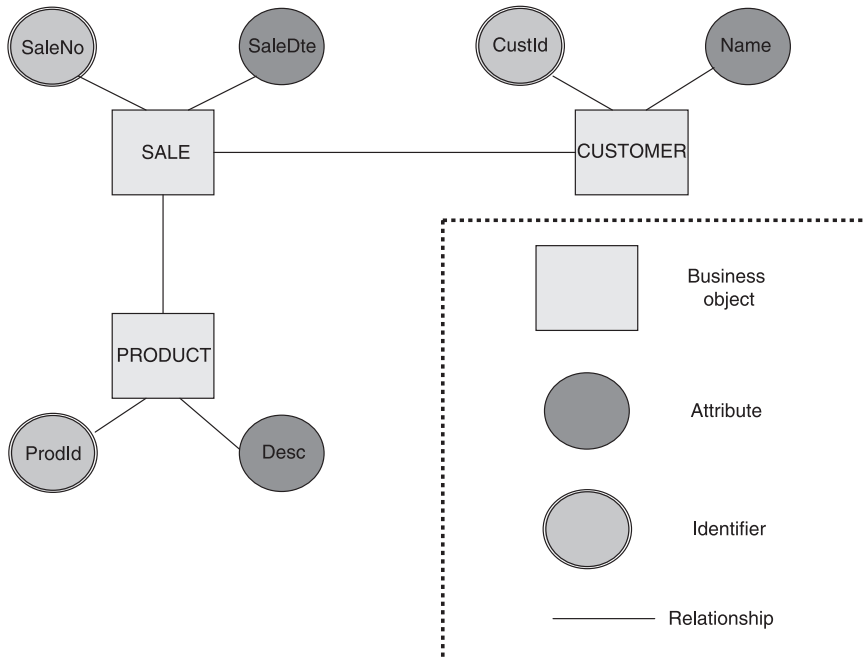


FIGURE 1-11 Data model components: simple representation.

Data Modeling Steps

Armed with the definition and description of the major data model components, let us quickly walk through the process of creating a conceptual data model using these components. A simple business example will illustrate the steps. Let us consider a rather easy and uncomplicated business.

The company is called Raritan Catering started by two sisters, Mary and Jane. They offer catering projects for birthday parties, anniversaries, student reunions, small business luncheons, and so on. They have four permanent employees and also use temporary help whenever necessary. For each project, the clients specify the type of food to be served and also the number of people to be served. Raritan Catering charges the client on a per-plate basis. Permanent and temporary employees working on a project are compensated based on the proceeds from the project. Twenty-five percent of the revenues from each project is shared among the employees working on that project.

In order to run the catering business, the two sisters perform various business processes. For the purpose of illustrating the data modeling steps, let us concentrate on the following major business processes:

- Plan for projects
- Assign employees to projects
- Bill clients
- Receive payments
- Compensate employees

With this information about the business processes and operations, let us walk through the process of creating a data model.

Identify Business Objects. In this step, the data modeler identifies those business objects about which data would be needed. How does he or she do this? The data modeler examines and studies the business processes that are necessary to run the business. In the case of Raritan Catering, we have noted the major business processes to run the business. The business needs information to perform these business processes.

Examining each of these processes closely, we can come up with the follow data elements that are necessary for each process. The data elements noted below comprise the basic information requirements for the processes.

Plan for Projects. For performing this business process, the business needs data about clients, their scheduled projects, and the type of food items to be served at the projects. Let us list these data items.

ClientName
ClientPhone
DateOfProject
NumberOfGuests
EstimatedCost
TypeOfFood

Assign Employees to Projects. For doing this function and assigning a particular employee to a specific project, the business requires some data about client, project, and employee. The following date elements are needed.

ClientName
DateOfProject
EmployeeName

Bill Clients. For performing this business process, the business needs to send invoices to the clients for the projects served. The following data items are required.

ClientName
ClientAddress
ClientCity
ClientState
ClientZip
DateOfProject
NumberOfGuests
AmoutCharged

Receive Payments. For this business process, the organization needs data about the payment received and from whom it is received. The following data items are essential.

ClientName
DateOfProject

PaymentDate
TypeOfPayment
AmountPaid

Compensate Employees. For this business process, the company must have information about each project and the employees who worked on the project. Here is the list of data items.

DateOfProject
AmountCharged

EmployeeName
EmployeeSSNo

After collecting the list of data items that would be required for all the business processes, the data modeler examines the list and groups together data items that are about the same thing. This process of aggregation results in the ability to identify the business objects. Let us aggregate the data items listed above for the five business processes. When you look at the data items and group similar data items together, it becomes clear to you that the data items relate to the following business objects:

CLIENT
PROJECT
EMPLOYEE
FOOD TYPE
PAYMENT

Identify Direct Relationships. In an organization, the business objects are interrelated. For Raritan Catering, clients order projects. So the business objects CLIENT and PROJECT are connected through this relationship. Similarly, other pairs of objects are likely to be connected by such direct relationships.

Let us examine each pair of potentially related objects. We are looking for potential relationships. The following direct relationships become apparent:

Employees are assigned to projects.
Projects use food types.
Clients order projects.
Client makes payments for projects.

Thus we can identify the following direct relationships between the pairs of business objects noted below:

EMPLOYEE —— assigned to —— PROJECT
PROJECT —— uses —— FOOD TYPE
CLIENT —— orders —— PROJECT
CLIENT —— makes —— PAYMENT

We have identified the direct relationships among the business objects. Well and good. But what about some compelling questions about each of these relationships? Let us take the relationship between EMPLOYEE and PROJECT. We see that information in our data system must tell us which employees were assigned to which projects. Right away, we observe that in our data system, the following restrictions would apply:

One employee may be assigned to one or more projects.

One project can have one or more employees.

That means the two objects EMPLOYEE and PROJECT are in a many-to-many relationship. See Figure 1-12, which illustrates the many-to-many relationship. Note also how the many-to-many relationship is indicated with asterisks (*).

Let us examine one more relationship. Take the relationship between CLIENT and PROJECT. We see that we need information in our data system about which clients ordered which projects. In this case, we note the following restrictions that apply to the relationship:

One client can order one or more projects.

However, one project can have only one client.

This is a one-to-one relationship. Relationships between business objects can be one-to-one or one-to-many or many-to-many. This aspect of a relationship is known as the cardinality of the relationship. The cardinality indicator such as "*" or "1" indicates how many instances of one object may be related to how many instances of the other object. Our data model must not only represent the relationships but their cardinalities as well.

Let us pause at this point and summarize what we have done so far. We have identified the business objects. We have noted the relationships among the objects and recorded the cardinalities of each relationship. Let us draw an initial data model diagram to represent the steps up to now. Figure 1-13 shows the initial data model diagram for Raritan Catering.

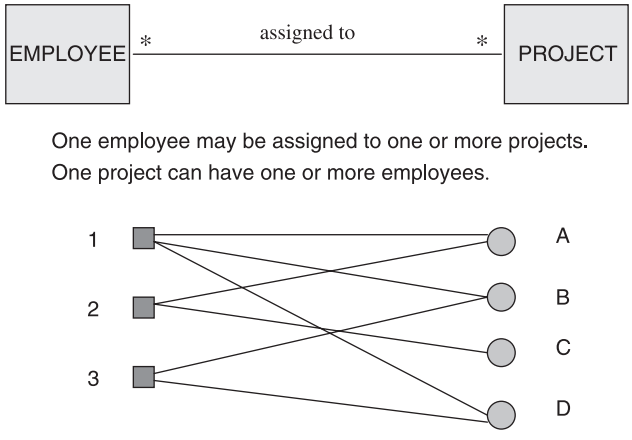


FIGURE 1-12 Many-to-many relationship.

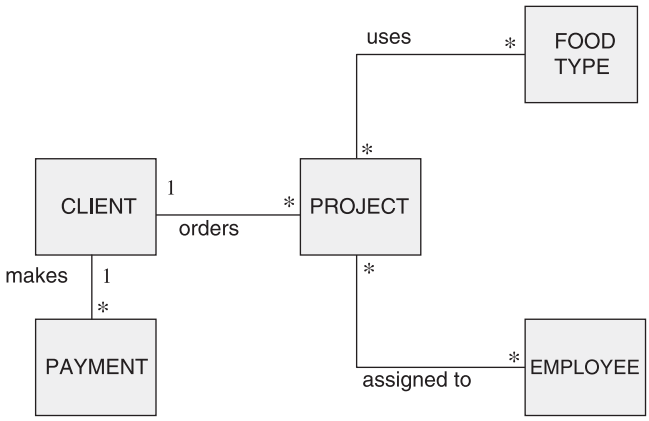


FIGURE 1-13 Raritan Catering: initial data model diagram.

Add Attributes. Attributes describe the business objects. Therefore, attributes are necessary components in a data model. After identifying the business objects and determining the relationships among the objects, the next step is to include the attributes and make the data model diagram even more complete.

Initially, when we examined the business processes and noted the data elements needed for each business process, we had made a list. This list of data elements will form the set of attributes to be included now. Let us collect the data items and form the set of attributes as follows:

CLIENT
- Name
- Address
- City
- State
- Zip
- Phone

PROJECT
- Date
- NoOfGuests
- EstCost
- AmtCharged

EMPLOYEE
- SSNumber
- EmpName

FOOD TYPE
- FoodName

PAYMENT
- Date

PymntType
Amount

Assign Identifiers. An identifier for a business object is one or more attributes whose values uniquely identify individual instances of the object. In this step, we examine the set of attributes for each object and determine which ones are candidates to be the identifier. If none of the known attributes of an object qualifies to be an identifier, then we have to introduce one or more new attributes to form the identifier. Let us examine the list of attributes.

For the object EMPLOYEE, one of its attributes, namely, SSNumber qualifies to be the identifier. The values of Social Security number are not duplicated. Therefore, the values of this attribute can be used to identify individual employees.

Scrutinizing the attributes of the other objects, we note that in each case there are no candidates for identifiers. So, for these objects, we have to add new attributes to serve as identifiers. After doing that, we come up with the following list of identifiers:

CLIENT
ClientNo

PROJECT
ProjectNo

EMPLOYEE
SSNumber

FOOD TYPE
TypeId

PAYMENT
PymntSeq

Let us return to the data model diagram and add the attributes and identifiers. Figure 1-14 shows the revised and more complete data model diagram.

At this point, the data model diagram is generally complete.

Incorporate Business Rules. Already when we marked the cardinality indicators for the relationships, we have indicated some business rules in the data model. For example, a rule guiding this business is that a particular food type can relate to several projects and that many different food types may be served at a single project. We have incorporated this business rule by marking the relationship between PROJECT and FOOD TYPE as many-to-many.

For a different business, there could business rules governing how invoices may be paid. Partial payments may be allowed against a single invoice; on the other hand, one payment may cover several invoices. This is a business rule, and the data model must reflect this rule. Business rules govern objects, relationships, and even attributes. Business rules must be incorporated into the data model so that the model could truly represent the real-world business situation. We will elaborate on business rules later in Part II.

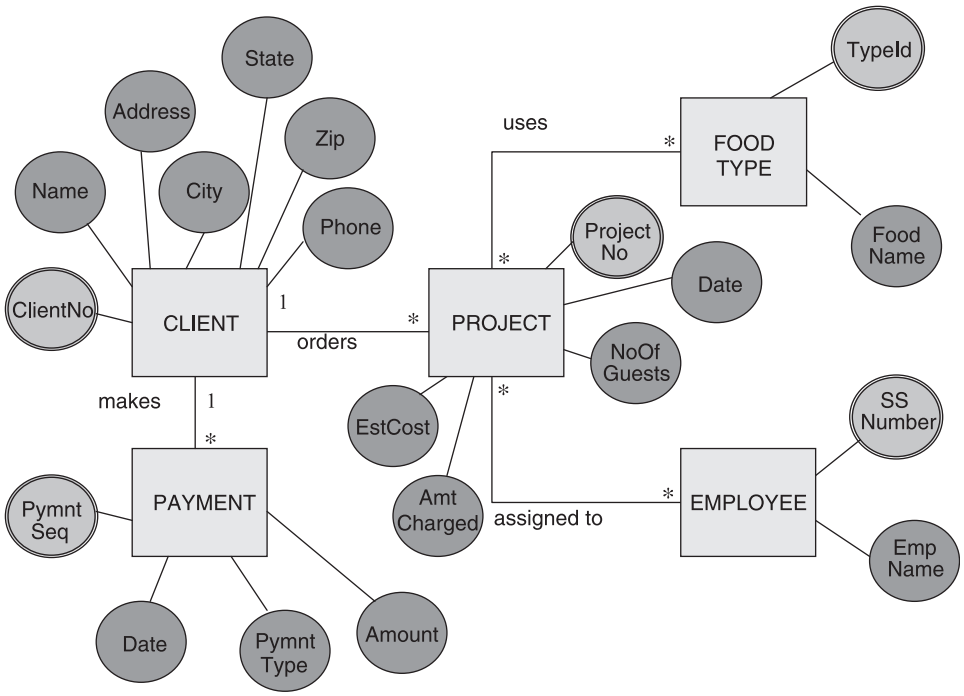


FIGURE 1-14 Raritan Catering: revised data model diagram.

Validate the Data Model. This step is really a review and validation step. We want to make sure the conceptual data model we have created satisfies two criteria. Does the data model truly reflect and represent the information requirements of the organization? Is the data model complete and simple enough to be used as a communication tool with the users? If so, we have succeeded in creating an effective conceptual data model.

In this step, you will review the models for completeness of the number of business objects. Have you missed any object that must be in the data model? Then you will review the relationships and ensure that the cardinalities are correctly specified. Again, you will go over the list of attributes and make sure no attributes are missing. Finally, you will verify the appropriateness of the identifiers. After this validation process, the data modeling steps are completed and the diagram represents the information requirements of the organization.

In real life, data modeling is not as simple as this example tends to make us believe. However, the overall steps for creating a conceptual data model are more or less the same for all situations. Part II pursues the topics in great depth. Using a comprehensive case study, you will go through the data modeling process in elaborate detail. What we have done in this section now is just a preliminary introduction to the modeling process.

DATA MODEL QUALITY

When we walked through the steps for creating a conceptual data model, we validated the model in the final step. We indicated a few of the tasks for validating the model. These

tasks ensure that the model is of high quality. But ensuring data quality involves a lot more than indicated in that final step. The importance of high quality in a data model cannot be overemphasized. Further phases follow the data modeling phase for implementing a data system for an organization. If the model is inadequate and of poor quality, then the inadequacy will be propagated to all the phases that follow data modeling.

Chapter 10 is dedicated completely to data model quality. There we will discuss the topic in elaborate detail. We will examine the reasons for the need for high quality. We will explore quality dimensions as they relate to data models. We will study how to recognize a high-quality data model. Also, we will review the methods for ensuring high quality in a data model. In this section, we just want to introduce the concept of quality in a data model and catch a glimpse of the relevant topics.

Significance of Data Model Quality

Two basic concepts of quality are completeness and correctness. For a data model to be of high quality, it must be both complete and correct. Let us briefly examine these two concepts and see how they relate to the significance of data model quality.

Data Model Completeness. When you scrutinize a data model for completeness, let us suppose you find that representations of some of the business objects are missing in the model. Consequently, you will also find that any direct relationships among these objects will also be missing. What is the result of lack of completeness?

To that extent, the data model will not truly represent the information requirements of the organization. Therefore, the final data system implemented based on the defective data model will not be able support the business of the company. Business processes that depend on the data about the missing objects and relationships cannot be performed.

Data Model Correctness. Similarly, let us suppose that the attributes of an object shown in the data model are wrong. Also, assume that two of the relationships are shown in the data model with erroneous cardinality indicators.

To that extent, the data model represents the information requirements incorrectly. These errors will filter through to the final data system and will affect the corresponding business processes.

Data Model Characteristics

What makes a data model to be of high quality? When can we say that a data model is good and adequate? Can we specify any general characteristics for a high-quality data model? Let us explore some of these features.

Involves Users. Unless the relevant users are completely involved during the process of data modeling, the resulting model cannot be good and valuable. The domain experts need to provide continuous input. While reviewing business operations for the purpose of identifying the right business objects, the involvement of the users with appropriate expertise in the particular business domain is absolutely necessary. Also, the right stakeholders must participate in the process.

At every iteration in the modeling process, the data model will be used as a means of communication with the domain experts and stakeholders. The input from these users will

enable the data modeler to refine the model as it is being created. With this kind of close participation, the data model is expected to be of high data quality.

Covers the Proper Enterprise Segments. If the goal is to represent the information requirements of the entire enterprise, then your data model must be comprehensive to include all the business processes of the whole enterprise. In this case, the final data system built based on the comprehensive model will be of use for all the users.

In practice, however, unless the enterprise is of small to medium size, all information requirements will not come within the scope of the data model. The data model will be created to cover only those enterprise segments of immediate interest. In a large company, it is possible to start with a data system to support the functions of only a few divisions such as marketing and finance. Then the data model will represent the information requirements to support only the business processes of marketing and finance. Here, the emphasis is on knowing what to include and what not to include so that the data model will be correct as well as complete.

Uses Accepted Standard Rules and Conventions. In the previous section when we reviewed the components of a data model and walked through the steps for creating a conceptual data model, we improvised and used our own simple set of symbols. For the purpose of introducing the data modeling process, these symbols and conventions were sufficient. However, if you showed the data model diagram to someone else, that person may not understand the representations. This is because the symbols and conventions are not an accepted standard. To this extent, our data model is not of high quality.

A good data model must be governed by standard rules and diagramming conventions. Only if you use industry-accepted standards can your data model be good and universal. We will introduce some modeling techniques toward the end of this chapter. Chapter 2 is completely dedicated to accepted standard modeling techniques.

Produces High-Quality Design. One of the primary goals of data modeling is to produce a good blueprint for the final database system of the organization. The completeness and correctness of the blueprint are essential for a successful implementation. A poor data model cannot serve as an effective blueprint.

For a data model to be considered a high-quality model, you must be able to complete the design phase effectively and produce an excellent end product. Otherwise, the model lacks quality.

Ensuring Data Model Quality

The importance of data model quality necessitates measures to ensure the quality. A poor-quality data model results in a poor-quality data system. Quality control must be given a high priority in the whole modeling process. Let us just mention how quality considerations must be approached and also a few quality control methods.

Approach to Data Model Quality. At every step of the data modeling process, you must review and ensure that the completed data model will truly serve each of its two major purposes. Is the data model clear, complete, and accurate to serve as an effective communication tool? Can the data model be used as a good working blueprint for the

data system? The data model must be reviewed for clarity, completeness, and accuracy at every stage of its creation.

Quality control comprises three distinct tasks: review, detection, and fixing. Every step of the way, the model must be reviewed for quality control. There must be techniques and tools for detecting problems with quality. Once quality problems are detected, they must be fixed forthwith. The data modeling team must develop and use proper methods to fix the quality problems.

Quality Control Methods. Quality control methods include the three tasks of review, detection, and repair. Usually, these tasks are performed in two ways. First, the tasks are performed continuously at every modeling step. In this way, less problems are likely to surface at the end of the modeling process. Second, when the modeling is complete, the overall model is again reviewed for any residual quality problems. When any residual problems are detected at this stage, they are fixed to assure high quality for the complete data model.

Who performs the quality control functions? A good approach is to share these functions. In the review and detection tasks, the data modeling team and the users must work cooperatively. Fixing of quality problems is generally the responsibility of the data modelers.

DATA SYSTEM DEVELOPMENT

As an IT professional, you are familiar with how a database system for an organization is developed and implemented. You have fairly good ideas of the various phases and tasks of the development process. Perhaps you are also knowledgeable of who does what in the process. In this section, we want to consolidate your ideas with particular emphasis on data modeling. Where does data modeling fit in the whole process? What purposes does it serve and what is its significance?

Data modeling forms an integral part of the design phase. It plays the role of the link between the requirements definition phase and the actual implementation of the data system. We have already reviewed data models at the different information levels. We have discussed the external, conceptual, logical, and physical data models that, in their specific ways, represent the information requirements. Where do these types of data models fit in the development process? How are they used?

Data System Development Life Cycle

Modern organizations depend upon the effectiveness of their data systems for their success. The information content of the data system of an organization is a key asset for the enterprise. The data system provides information critical for achieving the organizational goals. The data system enables the fulfillment of the organization's core business and drives the various business processes. The importance of the data system cannot be overstated.

Because the data system is a precious asset, each organization develops the data system with utmost care utilizing the best available resources. The design and development of the data system must be substantially significant. If so, how should an organization go about establishing its data system? The organization needs to do sufficient planning for the data

system project. The development of a data system calls for a coordinated systematic approach with distinct and purposeful phases. Organizations adopt a systematic life cycle approach. A life cycle approach addresses all the phases from beginning to end in an organized and methodical manner. Let us discuss a few major aspects of the data system development life cycle (DDLC).

Starting the Process. After all the preliminary administrative functions are performed, the following are a few major factors in starting the project.

Data-Oriented Approach. At the outset, you must realize that this project requires a data-oriented approach instead of a function-oriented approach. This means emphasis on the data remains throughout the development and implementation phases.

Development Framework. Create and work with a structured framework for the development of the data system. The following components of a framework may be adapted to suit your individual organization: scope of the data system, goals and objectives, expectations, justification, current and future requirements, implementation strategy, time constraints, and development tools and techniques.

Initiation Report. Initiate the project with a report whose standard contents would include the following: scope, goals and values, key business objects, core and primary business processes, tentative schedule, project authorization.

Planning. Do sufficient initial planning to get the project started. The planning for the data system should include the interpretation of the organization's long-term plan and application to the data system.

Feasibility Study. This is assessment of the organization's readiness for the implementation. Assess the resource requirements, estimate costs, and determine tangible and intangible benefits.

Requirements Definition. Business analysts and data analysts review the various business processes and study the information requirements to support the processes. The study would include one-on-one and group interviews with the users. Existing documentation must be reviewed. The analysts would watch and analyze how each business process is performed and what data is generated or used in each process.

Requirement definition comprises the following major tasks:

- Study overall business operations
- Observe business processes
- Understand business needs
- Interview users
- Determine information requirements
- Identify data to be collected and stored
- Establish data access patterns
- Estimate data volumes

When the requirements definition gets completed, an appropriate definition document will be issued. This document will be reviewed with the users and confirmed for correctness and completeness.

Design. Data modeling forms an integral part of the design effort. You design the data system based on the data models created at the different information levels. Conceptual design is based on the conceptual data model; logical design results from the logical model. The physical implementation works on the basis of the physical data model.

Figure 1-15 illustrates the design process. Note the different types of data models and how they are related. Also, notice how each part of the design effort rests on the particular data model.

Implementation. When the design phase is completed, the data system is ready for implementation. Completion of the physical data model and using it for implementation are responsibilities of the database administrator. He or she defines the data structures, relationships, business rule constraints, storage areas, performance improvement techniques, and completes the physical data model. This is at the physical level of hardware and storage.

Using the facilities of the selected DBMS, the database administrator establishes the data system. Once the structures and relationships are defined, the database is ready for initial data. Typically, organizations make the transition from earlier file systems to the database environment. Programmers extract data from the earlier systems and use the data to populate the new database. Special utility programs that are usually part of the DBMS enable the data loading with considerable ease.

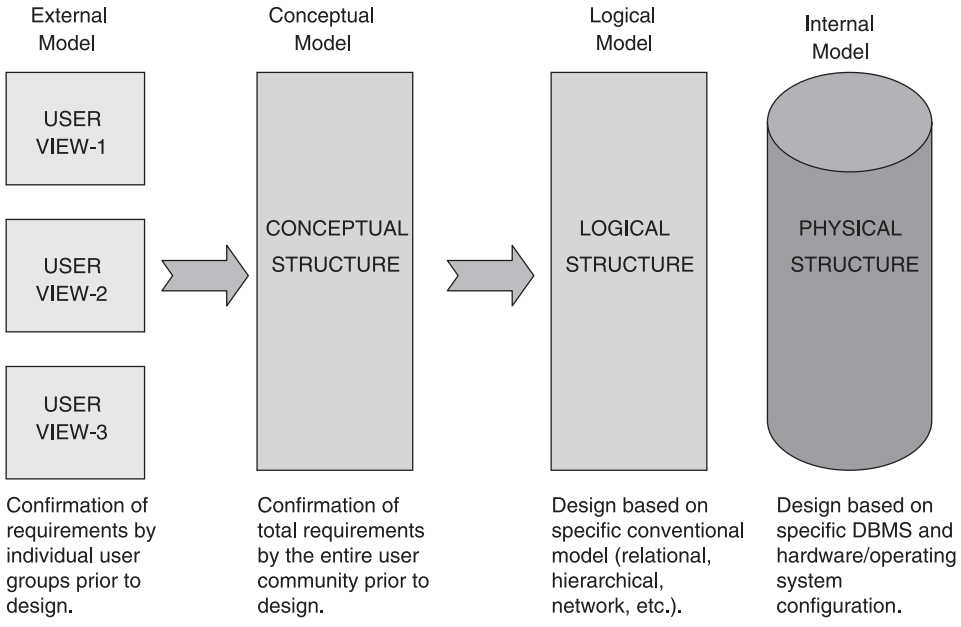


FIGURE 1-15 Data modeling in the design function.

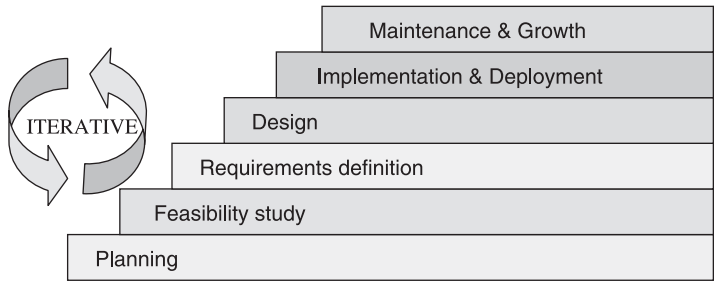


FIGURE 1-16 DDLC: major phases.

Phases and Tasks. The life cycle approach comprises systematic and well-defined phases or steps to complete the design and development of a data system. Each phase consists of specific major activities; each activity contains individual tasks. Although the project progresses from phase to phase, the people working on the project do not necessarily complete one phase and then move on to the next. Parts of the phases may be performed in parallel. Sometimes it becomes essential to repeat and refine some of the phases in an iterative manner.

Figure 1-16 shows the major phases of the DDLC. Note the sequence of the phases from bottom to top. Notice how the figure illustrates that the phases may be performed in an iterative fashion. Although some aspects of requirements definition remain to be completed, the design phase may commence. When you bring the design phase to partial completion, you may go back to the requirements phase and fine-tune some aspects there.

The scope of our discussion here does not call for detailed description of each phase. Nevertheless, let us highlight the objectives in each phase.

Planning. Review the organization's long-term business plan; plan specifically for the data system.

Feasibility Study. Study the state of readiness: estimate costs and explore benefits.

Requirements Definition. Define the business objects and relationships; document data requirements.

Design. Complete data modeling; design at conceptual, logical, and physical levels.

Implementation and Deployment. Complete physical design and define data structures and relationships using DBMS; populate data system; get data system ready for applications.

Maintenance and Growth. Perform ongoing maintenance; plan and manage growth of data system.

Roles and Responsibilities

Although we are mainly interested in data modeling within the DDLC, we here just want to identify who plays which roles in the entire process. Here is an indication of the participation by users and practitioners.

Planning: Senior management
Feasibility study: Business analysts
Requirements definition: Systems analysts, data analysts, user representatives
Design: Data modelers, database designers
Implementation and deployment: Systems analysts, programmers, database administrators
Maintenance and growth: database administrators

Modeling the Information Requirements

Let us now turn our attention to data modeling within the design phase. Let us discuss how data models are created to represent the information requirements. We will take a simple specific example. Let us consider the information requirements for an insurance company. Each of the user groups performs specific business processes. While performing business processes, each user group either uses relevant stored data or creates and stores data for later use. In either case, each user group is interested in a set of data elements.

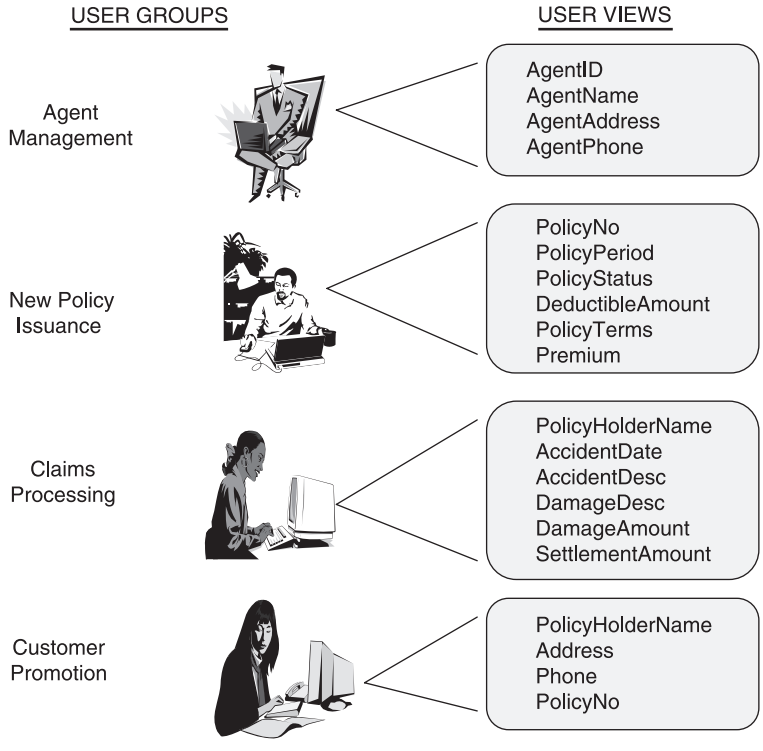


FIGURE 1-17 User groups and user views of data system.

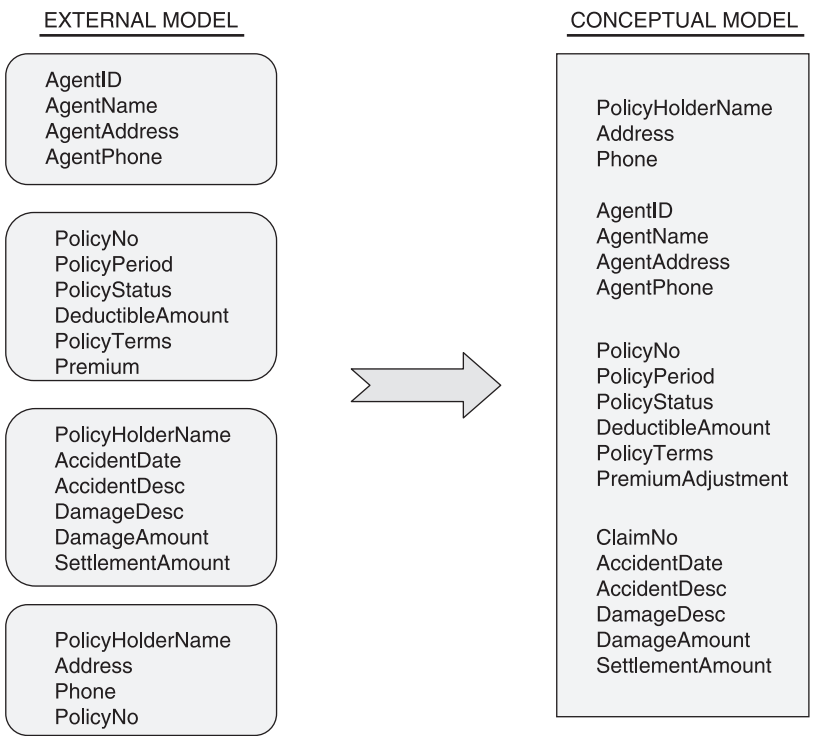


FIGURE 1-18 Insurance company: external and conceptual data models.

For the sake of simplicity, assume four user groups for the insurance company: agent management, new policy issuance, claims processing, and customer promotion. Each user group uses or requires a set of data elements. These sets of data elements form the user views of the data system. Figure 1-17 indicates the user views for these four user groups.

The complete list of user views comprises the external data model. The external data model is simply the various sets of data elements the different user groups are interested in. If you take into account all the user groups in the organization for which the data system is being implemented, then the aggregation of all the user views produces the conceptual data model. See Figure 1-18 for the external and conceptual data models for the insurance company.

The next data modeling task produces the physical data model from the conceptual. Every business object, attribute, and relationship represented in the conceptual data model gets transformed in the physical data model. Figure 1-19 gives an indication of the physical data model.

Applying Agile Modeling Principles

In recent years, system developers and data modelers have been adopting agile development principles. This is because agile development principles enable professionals to be proactive and produce results. The literature on agile system development and agile project management continues to grow. See the bibliography at the end of the book for some leading authors on these subjects. Throughout the subsequent chapters, we will be

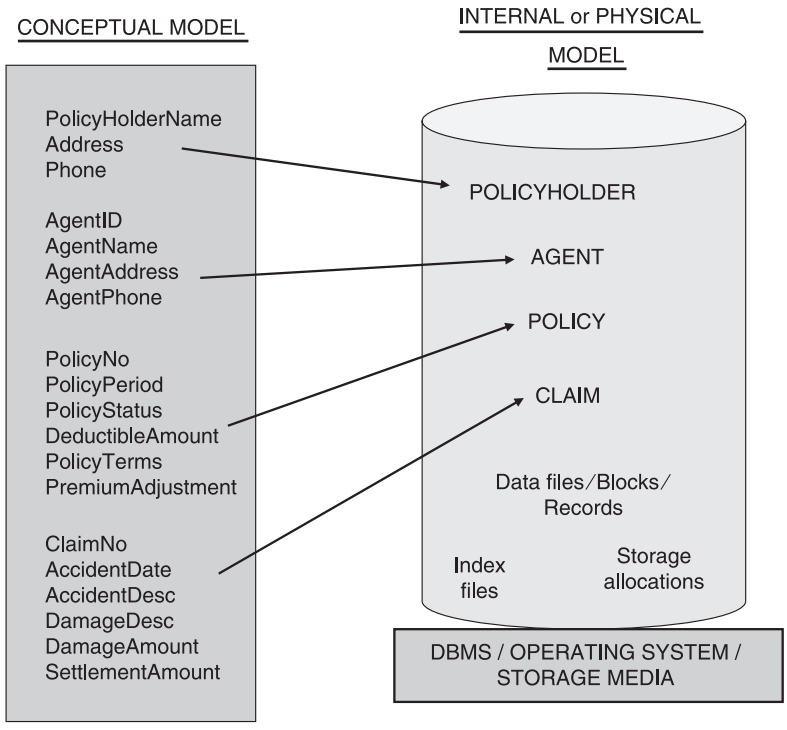


FIGURE 1-19 Insurance company: conceptual and physical data models.

mentioning agile modeling principles as they are applicable to our study. Chapter 11 is totally dedicated to agile modeling as it is practiced.

Agile modeling is not a complete software development process. It is more a set of valuable principles than a methodology for data modeling. The principles act as catalysts to any chosen modeling technique. Agile modeling enables putting values and principles into practice for effective, easy modeling.

DATA MODELING APPROACHES AND TRENDS

Thus far we have reviewed the basic concepts of data modeling. We discussed how and why information perceived at various information levels in an organization must be modeled. At each level, the data model serves specific purposes. We concentrated more on the conceptual data model at this stage and reviewed its components. We walked through the phases of the data system development life cycle. More explicitly, we covered the steps that are taken to create a data model—particularly, conceptual data model. We also touched upon the importance of data model quality.

Let us now conclude this introductory chapter with a historical note and review the evolution of data modeling over time. The need for data modeling was realized from the earliest times when database systems began to appear on the scene. Joint participation of users and information technology (IT) professionals in system development ensued. As organizations expanded in size and complexity, translating information requirements

directly into database systems became almost impossible. Users had to understand and confirm the understanding of IT professionals. Information technology professionals also needed an intermediary representation of information requirements that could act as a blueprint for the database system. Hence, data modeling evolved as a distinct effort in the process of system development.

Data Modeling Approaches

Earlier when we created a data model diagram, remember we used a set of symbols. A few basic rules guided us in the formation of a data model. We used square boxes, circles, and straight lines. This was a method of abstraction for representing information requirements in a somewhat understandable way. Every modeling method, therefore, has a set of notations or symbols. Each symbol means something and represents some aspect of the real world we are trying to represent. A data modeling method also has a set of rules or procedures for using the symbols.

Data modeling, especially conceptual modeling, is a collaborative effort between data modelers and users who are domain experts. The real-world information requirements must somehow be made clear in the data model through natural language, easily understood and intuitive diagrams, and also through data examples. These examples are the examples of the type of data that the data model is expected to portray. Sometimes, these are referred to as data use cases because they are the data used by the system.

Data modelers adopt three major data modeling approaches: entity-relationship data modeling, fact-oriented data modeling, and object-oriented data modeling. Chapter 2 is fully dedicated to the discussion of data modeling methods. Therefore, in this section, we will just introduce these approaches and consider some major points.

Entity-Relationship Modeling. This approach, introduced by Peter Chen in 1976, is still the most popular and widely used technique. Vendors have produced several computer-aided software engineering (CASE) tools to support this method. This method perceives and portrays the information requirements of an organization as a set of entities with attributes participating in relationships. Based on earlier discussions and examples, we are already somewhat familiar with this method. Over the years, newer versions of entity-relationship modeling came on the scene. The newer versions included improvements and enhancements. Symbols used in entity-relationship modeling are not fully standardized although if you know one convention, it would be easy to guess the meaning of similar symbols in another convention.

Entity-relationship modeling portrays the information domain of an organization in a way that is free from any considerations of database software or hardware. Because of this independence, this method is well suited for conceptual data modeling. It does not burden the domain experts with unnecessary details. However, an entity-relationship (E-R) data model diagram has its shortcomings. The diagram does not clearly indicate constraints in the relationships. Does every instance of one entity always relate to instances of the related entity? How are constraints on mandatory and optional relationships indicated? Those practitioners who tried to remove the defects have attempted some enhancements to E-R modeling. But these attempts are not fully satisfactory.

Domain experts want to relate data use cases to the model diagram. They want to know how each use of a set of data is specified in the model. This link of data use case to the model is not obvious. Not all domain experts are comfortable with the notations in the

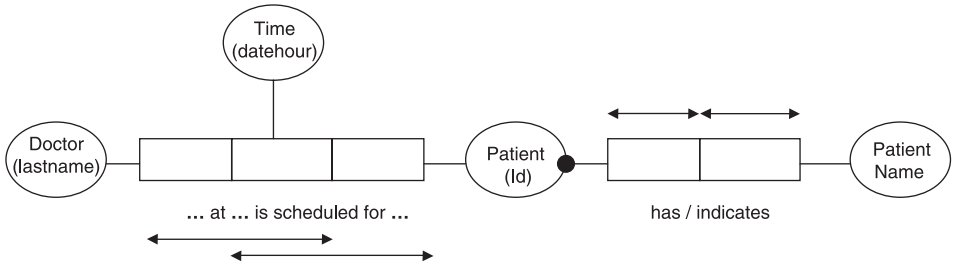
E-R model and find some of the notations, especially those for relationships, incomplete and imprecise. The fact-oriented data modeling approach attempts to overcome some of the deficiencies of the E-R approach.

Fact-Oriented Modeling. In the 1970s, an approach to data modeling arose by viewing the information domain in terms of objects playing roles. What are roles? A role is the part played by one object in a relationship. Object-role modeling (ORM) is such a fact-oriented modeling approach. This is perhaps the only major fact-oriented modeling technique with fairly wide industry support.

Let us try to understand ORM with an example. Consider the data use case for scheduling the time of doctors for patient visits. The domain expert who is familiar with this simple use case is able to verbalize the information content of this use case. The data modeler must be able to transform this verbalization into a data model to discuss easily and to validate the model. In this data modeling technique, the data modeler verbalizes sample data as instances of facts and then abstracts them into fact types. Constraints and rules of relationships and derivation get added to the model to make it complete.

Figure 1-20 shows an ORM diagram for doctor scheduling. Named ellipses indicate entity types and have a method for referencing them. The reference (last name) shown in parentheses within the ellipse for Doctor means that doctors are referred to by last names. The sample data form the fact instances, and these get abstracted into the fact types. The ORM diagram indicates the structure consisting of the fact types of Doctor, Time, Patient, and PatientName. A named sequence of one or more role boxes depicts a relationship. In this case, we have a three-role or ternary relationship: Doctor at Time is scheduled for Patient. We also have a binary or two-role association between Patient and PatientName. The model diagram also includes some counterdata to indicate appropriate constraints.

An ORM diagram presents all facts in terms of entities or values. Unlike as in E-R, attributes are not specified in the base ORM models. Object-role modeling allows for relationships with multiple roles. Object-role modeling diagrams tend to be large; nevertheless, an attribute-free model is simpler, more stable, and easier for validation. The arrow-tipped lines indicate uniqueness constraints showing which roles or combination of roles must



11	Mon, 9 a.m.	223
12	Wed, 2 p.m.	254
15	Thu, 3 p.m.	276
…	…	…
11	Mon, 9 a.m.	276 ?
15	Thu, 3 p.m.	223 ?

221	Hastings
223	Wallace
254	Rivera
276	Truman
…	…
221	Goldman?
292	Wallace?

FIGURE 1-20 ORM diagram: doctor scheduling.

have unique entries. A black dot on Patient refers to a mandatory role constraint. The counterexamples of data with question marks (?) provide means to test such constraints while validating the model with a domain expert.

Compared with ORM, E-R has the following shortcomings:

- It is not closer to natural language for validation with domain experts.
- E-R techniques generally support only two-way relationships. Although *n*-way relationships in E-R are broken down into two-way relationships by introducing intersection identities, these intersection identifies seem arbitrary and not understood by domain experts.

Object-Oriented Modeling. In this approach, both data and behavior are encapsulated within objects. Thus, object-oriented modeling was primarily devised for designing code of object-oriented programs. However, this modeling approach can be adapted for conceptual modeling and eventually for database design.

By far, the most popular and widely used object-oriented approach is the Unified Modeling Language (UML). The Unified Modeling Language has an array of diagram types, and class diagrams form one important type. Class diagrams can represent data structures and may be considered as extensions of the E-R technique. Apart from this brief mention of UML here, we will postpone our detailed discussion of UML until the end of Chapter 2.

Modeling for Data Warehouse

As an IT professional, one must have worked on computer applications as an analyst, programmer, designer, or project manager. One must have been involved in the design, implementation, or maintenance of systems that support the day-to-day business operations of an organization. Examples of such systems are order processing, inventory control, human resources, payroll, insurance claims, and so on. These applications that support the running of the business operations are sometimes known as OLTP (online teleprocessing) systems. Although OLTP systems provide information and support for running the day-to-day business, they are not designed for analysis and spotting trends.

In the 1990s, as businesses grew more complex, corporations spread globally, and competition became fiercer, business executives became desperate for information to stay competitive and improve the bottom line. They wanted to know which product lines to expand, which markets to strengthen, which new stores and industrial warehouses would be worthwhile. They became hungry for information to make strategic decisions. Although companies had accumulated vast quantities of data in their OLTP systems, these systems themselves could not support intricate queries and analysis for providing strategic information.

Data warehousing is a recent paradigm specifically intended to provide vital strategic information. It is a decision-support system. In a data warehouse, data has to be viewed and represented differently. Data modeling appropriate for building OLTP database systems becomes ineffective for data warehouse systems. Techniques such as entity-relationship data modeling do not meet the requirements. Let us consider a simple example to illustrate why a different modeling technique becomes desirable.

Suppose we take all the sales data accumulated in the OLTP systems and want to use it for analysis. We want to ask a question such as: What are the sales of Product A for the

current year, broken down by regions, compared with prior year and targets, sorting the sales in ascending sequence by region? After viewing the sales data, suppose we want to zero in on the region with the lowest performance. We would then want to ask a follow-up question such as: For the region with the lowest performance, what is the breakdown by sales representatives, districts, and shipment methods. This approach of querying and analysis is likely to lead us to the reasons for the low performance so that we can make strategic decisions to rectify the situation. Therefore, a data warehouse must contain data extracted from OLTP systems—data that can be viewed and modeled for querying and analysis.

In this example, we want a data model that enables analysis of sales by year, region, sales representative, and shipment method. We need a model that supports analysis of sales data or facts about sales by combinations of the business dimensions of year, region, sales representative, and shipment method. The model should depict the information content in a data warehouse using dimensional modeling. We will discuss dimensional modeling technique in great detail in Chapter 9. That chapter describes data modeling for decision-support systems extensively.

Other Modeling Trends

Some recent data modeling and related trends include very-high-level languages for querying information systems, enhanced schema or model abstraction methods, newer techniques for creating external models, and extended modeling languages such as extensible markup language (XML). In the remaining chapters, we will include discussions on these as and when necessary.

Let us now conclude with brief discussions on a few other trends.

Postrelational Databases. In our earlier discussions, we have noted that a relational data model views data in the form of two-dimensional tables. We have also seen that a conceptual model may be mapped to a relational model. Prior to the advent of the relational model on the database scene, other models such as the hierarchical and network models preceded the relational model. In these models, data is viewed differently. The hierarchical model presents data as hierarchical segments in parent–child relationships. On the other hand, the network model consists of data nodes arranged as a network. Still, a conceptual model while being transformed into a logical model may take any one of these forms.

The relational model is still the most popular and widely used model; most data system implementations are relational databases. However, the recent uses of data and the generation of newer types of data pose problems for the relational model. Now we have to deal with data in the form of images, sounds, and spatial elements. More complex data objects have become common in industry. A relational model does not seem to be adequate for representing recent trends in data usage. Organizations adopt postrelational models to address the recent requirements. Data modelers need to adapt their modeling techniques to accommodate postrelational models.

Let us briefly highlight a few of these postrelational approaches.

Object-Oriented Databases. In addition to features of relational databases, these databases possess additional features such as support for complex objects, encapsulation involving bundling of operations and data together, and user-defined data types.

Deductive Databases. These databases offer powerful and elegant methods for declaring and managing complex data. For data that has to be derived by the use of a series of recursions, this approach proves very effective.

Spatial Databases. These databases provide support for spatial data types (points, lines, multisided figures, etc.) and spatial operators (intersect, overlap, contain, etc.). Because of these facilities, spatial databases efficiently manage spatial data such as maps (land, counties, states or provinces, etc.) and two- and three-dimensional designs such town plans, flight paths, and so on.

Process Modeling. Data modeling primarily confines itself to creating conceptual, external, logical, and physical models. If you examine each of these models, they represent the data content, perhaps, in a static manner. However, other aspects of working with the data exist in an organization. Somehow these other aspects such as business processes that use the data content must also be modeled. Many types of model diagrams depict these other aspects. The Unified Modeling Language includes such diagrams: use-case diagrams, sequential diagrams, collaboration diagrams, state charts, and activity diagrams. Other modeling techniques include data flow diagrams, process flow charts, and function trees.

As our primary emphasis rests on data modeling, we do not intend to discuss the process modeling techniques in detail. However, we will list the functions of the important ones.

Use Case Diagrams. These provide comprehensive overviews of the processes in a system.

Activity Diagrams. These show activities of each process. Activity diagrams at successive levels refine the activities as you proceed to the lower levels. Domain experts understand activity diagrams intuitively and find them very valuable.

Function Trees. These decompose major functions into subfunctions at several levels.

Data Flow Diagrams. These display flow of information between processes.

Meta-Meodeling. Data modeling creates models of data content in a database system. Process modeling deals with modeling the activities and functions surrounding the data system. Modeling involves creating models of the application domain. Meta-modeling involves creating models of the models themselves. A meta-model represents the contents of the model itself.

You can create meta-models for data models at various levels. A meta-model at a particular level describes the data model at that level. Thus, a conceptual meta-model contains the representation of the components of a conceptual data model. A conceptual data model describes business objects and their relationships. A conceptual meta-model expresses the components of a conceptual model, that is, the representations themselves as meta-objects and meta-relationships. A meta-model may be used to verify and validate the corresponding data model.

CHAPTER SUMMARY

- A data model is a representation of the information content in the real world.
- Essentially, a data model serves two purposes: as a communication tool and as a blueprint for constructing the database.
- A data model plays a significant role in the data life cycle of an organization.
- In an organization, four information levels are discerned; data models are created at these levels.
- A conceptual data model depicts the information requirements of an organization at the highest general level. Logical and physical data models are representations at the next two lower levels with more intricate details.
- A conceptual data model has the following major components or building blocks: objects, attributes, identifiers, and relationships.
- The process of data modeling consists of specific modeling steps or activities.
- Quality of a data model is of paramount importance.
- Data system development life cycle (DDLC) phases: planning, feasibility study, requirements definition, design, implementation and deployment.
- Data modeling techniques have evolved and been refined during the past decade.

REVIEW QUESTIONS

1. Match the column entries:

1. Data model	A. Closest to the users
2. Storing data	B. Intrinsic characteristic
3. External model	C. Popular modeling technique
4. Attribute	D. Representation of real world
5. Patient	E. Completeness and correctness
6. Conceptual model	F. Data life cycle stage
7. Model quality parameters	G. Blueprint for database
8. Feasibility study	H. Fragment of logical model
9. Physical model	I. Determine state of readiness
10. Entity-relationship modeling	J. Business object for a hospital

2. Describe the two primary purposes served by a data model.
3. What are the various stages of the data life cycle in an organization? How can a data model be useful in these stages?
4. Name some major functions performed by a data modeler. What types of skills are needed to perform these functions?
5. Describe the notion of information levels in an organization. What are the typical levels and why are they important?
6. Name the different types of data models and relate them to the information levels. What are the essential purposes of these types of data models at these levels?

7. Name and briefly describe the components of a conceptual data model. Give examples.
8. Briefly describe the general data modeling steps. Indicate the activities at each step.
9. List the major phases of DDLC. What are the objectives of each phase?
10. Name any three data modeling approaches. Describe one of the approaches.