Database Systems for Management Third edition

James F. Courtney

David B. Paradice

Kristen L. Brewer

Julia C. Graham

For any questions about this text, please email: drexel@uga.edu

The Global Text Project is funded by the Jacobs Foundation, Zurich, Switzerland

This book is licensed under a Creative Commons Attribution 3.0 License

CHAPTER 2

Elements of Database Systems

The first chapter provided an introduction to database systems, concentrating on three areas: user orientation, data as an organizational resource, and the objectives of database systems. In this chapter we discuss the organizational environment in which database systems are planned, designed, and used. As we saw in Chapter 1, the elements of the database environment are people, hardware, software, databases, and techniques for planning, designing, and managing databases.

In this chapter we examine the relationships among three of these elements--people, software, and techniques--and how the database approach may be used to develop systems that effectively support organizational needs. Here, you begin learning how to design and structure effective database systems. Remember that by "effective" system we mean a system that "does the right thing" by providing the information necessary to run the organization. With "structure" we refer to the parts of a system and how these parts interact. The present concern is with blending the five elements into a database system that serves the needs of the organization. The chapter begins with an extended analogy between a database system and the design and layout of a university campus. This provides the background for a discussion of the design and structure of a database system. After the analogy, we present sections on planning and design techniques, software, and people. Even though people are the most important part of a database system, they are discussed last, simply because the tasks performed by various people are easier to describe after the other elements have been discussed. After studying this chapter, you should understand how hardware, software, people, procedures, and plans can be blended to form an effective database system.

HOW A DATABASE SYSTEM IS LIKE A CAMPUS

Whenever a new campus or housing development is planned, a master land-use plan is developed. This plan takes into consideration the special needs of the people who will be using the campus or development and the special characteristics of the site related to those needs. Buildings are generally not constructed in floodplains, and architectural styles and placements are intended to provide convenience and accessibility in a way that is harmonious with the environment.

Land is a valuable commodity, so its use is carefully planned. Data, however, may not be recognized as a valuable resource, and often little planning or coordination is directed toward it. In the next section we discuss the need for careful database planning.

The Need for Planning and Modular Design

It is not uncommon for one unit of an organization to have databases supported by computer systems that are incompatible with those of other organizational units. This can lead to expensive redundancy of computer systems and to difficulty in maintaining organization-wide information.

The analogy with a campus can be used to illustrate this. The buildings of a campus are only loosely coupled. The only links between buildings are sidewalks, roads, and perhaps utility tunnels. Because buildings are highly independent, the detailed design and construction of each one may be done independently. Together, however, they should fit into a grander scheme that blends into the natural environment and serves the needs of the entire university community.

A well-designed database system consists of modules or subsystems that can be compared to the buildings of a campus. Here the term "module" is used to describe a major element of the overall database system, not to a piece of programming code. Database system modules are loosely coupled and may be built independently, but they should fit into a grander scheme described in a **Database System Master Plan.** Like a campus land-use plan, the database system master plan lays out the ultimate structure and contents of the database system--and does so before work begins on the system itself. Some similarities between the buildings of a campus and database system modules are illustrated in Figure 2-1.

The analogy can be carried further. Buildings consist of floors, floors contain rooms, and rooms have attributes such as color, number of windows and doors, and type of floor covering. Just as buildings contain floors and rooms, the subsystems of a database system contain data about entities, such as customers and products. An **entity** is any object, tangible or intangible, about which the organization wishes to store data. Entities have **attributes,** such as name, color, and price. Rooms and their attributes must be accessible through stairways and hallways. In some database systems, entities and their attributes must be accessible via navigation paths through the database. A **navigation path** through a database is like the signs in a hotel showing the direction of rooms. Arrows point left or right to indicate the route to certain rooms. The arrows in databases are **pointers**
indicating where the program should look on the system's storage devices to find certain records.

The Need for Access, Security, and Adaptability

Buildings and database systems are designed to be used by people with different characteristics and needs. It may be difficult for handicapped people to access floors and rooms of poorly designed buildings. Likewise, it may be difficult for "computer-handicapped" people (those who have not been trained to use computers) to access information in a poorly designed database system.

Just as someone may break into a building and steal property of value, a database may be broken into and valuable information may be stolen. Buildings are constructed in a logical sequence. Work does not begin on the third floor, and painters do not paint themselves into a corner.

Once a building is constructed, it may not be very expensive to repaint the walls or to put new carpet on the floors. However, it is probably very costly to modify the structure of a building to take out a wall. The same can be said for database systems. Once they are built, it is usually expensive to modify their structure. Thus the organization should ensure that its buildings and its database system are designed correctly in the first place.

The first thing the organization should do to assure a good database system design is to develop a database system master plan. This plan is discussed in the next section. The similarities between a campus and a database system are summarized in Table 2-1.

THE DATABASE SYSTEM MASTER PLAN

Contents of the database system master plan should include a statement of organizational goals and objectives, an organization chart, entity diagrams showing the major relationships among entities in the enterprise, modules or subsystems to be implemented as part of the database system, the costs and benefits expected to be associated with the system, and a schedule for implementing each module. An outline of the contents of a database system master plan is given in Table 2-2. Each of these elements is discussed in turn in the rest of this section, with illustrations from an educational organization called the Professional Training Institute (PTI).

Organizational Goals and Objectives

Just as a building should fit into its natural environment, a database system should fit into its surroundings. Two aspects of the organizational environment should be considered in the master plan: (1) organizational goals and objectives, and (2) the manner in which the enterprise is structured. Organizational goals and objectives are important to database system design; many critical information needs may be derived from them. When possible, goals and objectives should be stated in numeric terms, so that progress toward those goals can be measured. Instead of saying that the goal is to increase sales, for example, one might say that the goal is to increase sales by 10 percent.

The database system should assist managers of the organization in measuring the level of goal attainment and in steering the organization toward its goals. Thus the database system master plan should begin with a brief statement of organizational goals and objectives and how the database system will measure progress toward goals.

Case Example: Professional Training Institute

PTI's basic mission is to provide high-quality continuing education in a variety of subject areas to practicing professionals in many industries. This mission is attained through the sponsorship of one-day courses at PTI's facilities in a quiet woodland setting. Goals for the upcoming year are to expand course offerings in computer-related topics. Specific objectives are to offer the standard fare of courses taught the previous year and to add new courses in information management and computer networks. It is expected that 50,000 people will attend the standard course offerings and that 8000 will attend the new computer courses. These offerings should generate $10,000,000 and $2,400,000 in gross revenues, respectively. PTI's goals and objectives are summarized in Table 2-3.

Organizational Structure

The second feature of the organizational environment that should be included in the master plan is the organizational structure. It is important to consider the structure of the organization because the database system will surely be required to support the flow of information between various units or divisions of the enterprise. The information flow usually follows organizational channels between superior and subordinate units. For this reason, the database system is said to be "superimposed" on the structure of the organization. It is convenient to include an organizational chart and a brief description of the function of each of the organizational units in the database system master plan.

Case Example: Professional Training Institute

The organizational chart for the Professional Training Institute is shown in Figure 2-2. At the apex of the organization is the Director, who is responsible for the general management of the firm. This responsibility includes determining goals, objectives, and the strategies for achieving them. PTI has two major subdivisions: Marketing and Course Management. Each of these is headed by an Assistant Director who reports to the Director.

The Marketing Department has two major functions: (1) to promote courses already offered by PTI and (2) to develop new marketable courses. To promote courses, Marketing maintains extensive mailing lists of prospective participants ("prospects") for the courses being offered. Brochures describing course offerings are developed by the Marketing staff and mailed to these prospects. This is an extremely vital function, for without paying customers, PTI will be out of business. In the past year over a million brochures were mailed.

The Course Management Department is responsible for organizing and administering courses. Course management involves working with Marketing to promote courses, registering participants, collecting fees, scheduling instructors, ordering textbooks and other materials, administering course evaluations, and maintaining records on participants.

PTI realizes that accurate information on prospects and former participants is vital to its continued success. Satisfied customers are among the most likely prospects for future courses. Thus PTI has developed a database system to support its information management functions. PTI's system is described throughout the remainder of the chapter. One of the first tasks in developing a database system for the organization is the identification of relevant entities, attributes, and relationships between entities. In the next few sections, we discuss this process in more detail.

Entity Charts

The purpose of the database systems master plan is to guide the organization in its quest to manage its information resources effectively. The master plan views the information needs of the organization at a very broad, general level. It should identify the major information users in the organization and their general information needs. It should also specify the entities to be contained within the database and major relationships between entities. Entities and their relationships are initially defined by working with users, who must be able to specify their information needs.

Specified entities and relationships can be illustrated in an entity chart or diagram, such as that of Figure 2-3 for PTI. Entity charts represent the major entities and relationships among them to give an overall view of the database structure. Detailed database design then fleshes out this general structure, as described in the next chapter.

Case Example: Professional Training Institute

PTI decided that the entities of interest were courses, instructors, participants, course evaluations, prospects, and organizations for which prospects work (many times, an organization sends several of its employees to a course). In the entity chart, a box is made for each entity; the lines between boxes represent relationships of interest.

Relationships in the diagram are of three types: one-to-one, one-to-many, and many-to-many. In a **one-to-one** (abbreviated 1:1) relationship, one instance of an entity of a given type is associated with only one member of another type; for example, a participant in a course fills out one evaluation form for the course. Conversely, a given evaluation form was prepared by one participant. Single-headed arrows are used to represent a "one" relationship in the direction of the arrow. In Figure 2-3, a line is drawn between participants and evaluations with single arrowheads on each end to indicate a one-to-one relationship.

In a **one-to-many** (1:M) relationship, one instance of a given type of entity is related to many instances of another type. For example, one prospect works for (is associated with) one organization, so the line emanating from prospect has a single arrowhead where it terminates with organization. But one organization may have several prospects for a given course. Double-headed arrows are used to represent "many" relationships, so the line emanating from organizations has two arrowheads where it terminates with prospects.

Finally, a **many-to-many** (M:M) relationship is one in which many entities of one type are associated with many instances of another type. For example, an instructor may teach many different courses, and one course may be taught by many different instructors. Thus the line between courses and instructors has double arrowheads at each end.

Note that the entity chart does not show the details of every user view, entity, or piece of data in its scope. If possible, it should reflect a global or organization-wide perspective of information needs. Unfortunately, even with today's sophisticated technology, such a broad view may be unreasonable for massive organizations such as the federal government or multinational corporations. Nevertheless, the master plan and entity chart should be drawn up for the broadest feasible portion of the enterprise. What is feasible depends on the size of the organization itself, the amount of data it generates, and the resources that management is willing to commit to the project.

Entity Groups, Database System Modules, and Schedules

Because database systems are large and complex, they cannot be designed and built all at once. Rather, they are put together piece by piece over a long period of time--perhaps several years. The master plan should describe the major pieces or modules of the system to be developed, how these modules will fit together into an integrated structure, and a schedule for detailed design and development of each module. It may be possible to define these modules based on the number and strength of the relationships in the entity diagram.

Case Example: Professional Training Institute

PTI's entity groups are shown in Figure 2-4. The COURSE Entity Group consists of those entities most closely related to the administration of courses. This entity group defines a module that would be most heavily used by personnel in the Course Management Department.

The PROSPECT Entity Group consists of prospects and organizations and is used most heavily by Marketing to promote courses to individuals and organizations. These modules are not totally independent, of course. Note that two lines cross entity group boundaries (these overlaps are more evident when user views are examined). When PTI decided to develop an integrated database system, it already had workable systems in place for course administration. Because the management of the mailing list of prospects was an important task with a simple data structure but large amounts of data, PTI scheduled this as the first database system module to be developed. The simple structure would give PTI personnel a relatively easy problem to begin its work. The experience gained on the Prospect module would be helpful in tackling the more complex Course module.

Costs and Benefits

Management of the organization should, and probably will, view the development of a database system as an investment that must pay for itself; that is, the benefits derived from using information provided by the system must outweigh the cost of developing it.

Estimating the cost of developing a database system is not an easy task. If the organization does not already own an appropriate database management package, one must be selected and acquired. Selection must be done very carefully and may take six months to a year of personnel time. Database software itself is expensive, as mentioned in Chapter 1. The cost of installing the package and training systems personnel must be estimated. The master plan must be drawn up and detailed plans for each major module must be made before implementation. Applications software for each module must be developed or acquired, tested, and implemented, and users must be trained. Finally, the system must be operated and maintained.

There is a great deal of uncertainty in determining the cost in time and money of these tasks, but such estimates must be made. The organization must have a realistic picture of the anticipated cost to make sure that funds are available to develop and operate the system. It must also ensure that anticipated benefits outweigh these costs.

This brings us to the other side of the coin, the benefits. The value of benefits of database systems is much more difficult to estimate than costs. Many of the benefits of database systems are difficult to measure in dollars.

Case Example: Professional Training Institute

Suppose, for example, that we are developing a database system for the Professional Training Institute. This system is expected to allow Marketing to cut down on postal expenses by eliminating redundancy in current mailing lists and targeting potential course prospects and prospective organizations more effectively. But how much can postal expenses be cut? The system may also reduce the expense of managing courses. But how much will management expenses be reduced?

One way of dealing with such questions is to determine how much expenses would have to be reduced to pay for the system. Is a 5 percent savings in postal and management expenses discounted over each of the next three years enough to pay for developing and operating the database system? Another approach is to estimate how much additional revenue the database system would have to generate to pay for itself. Better targeting of prospects, for example, might lead to larger enrollments for PTI's courses. If enrollments could be increased by 10 percent in each of the next three years, the discounted value might be enough to pay for the system.

Because costs and especially benefits of database systems are difficult to estimate, a great deal of judgment is involved in deciding to institute a new system. This judgment should be based on the best possible estimates from information systems personnel. (For managers, these estimates are a very important form of information concerning the database system, even though they are not derived from the system itself, because it doesn't exist yet.)

To summarize, the database system master plan consists of brief descriptions of organizational goals and objectives, structure, entities, modules, costs and benefits, and an implementation schedule. In the next section, techniques for designing and operating each database system module are discussed.

TECHNIQUES AND PROCEDURES IN A DATABASE ENVIRONMENT

Although there are many different techniques and procedures found within a database environment, only one of the more important ones--database design--is previewed here. Other techniques and procedures are discussed in the chapters on database administration. These are more readily understood after other aspects of database systems have been studied.

Database Design Techniques

Once the master plan has been completed, work begins as scheduled on the detailed design of each module and on the database to support that module. The process of database design consists of three major phases: an information-requirements analysis phase, a logical design phase, and a physical design phase. Each of these phases is described below.

Information-Requirements Analysis and User Views Information-requirements analysis involves working with users to define their information needs in detail. Information-requirements analysis is concerned with information as users see it; that is, information is viewed in terms of the way it appears in documents, on terminal screens, or even in images in the user's mind. As explained in Chapter 1, representations of information and data at this level are referred to as "user views."

Example user views for PTI are illustrated in Figure 2-5. Not all views are shown. Figure 2-5a shows the report the instructor gets after the course is over and grades and evaluations have been reported. Figure 2-5b shows the participant's view before the course begins. The grade has not been reported and the participant has not yet rated the course. Figure 2-5c shows the course promoter's view. Additional information is provided about the instructor for advertising purposes. The course manager's views are shown in Figures 2-5d and 2-5e.

Note that the course manager's final grade report is similar to the instructor's, except that for reasons of privacy the instructor does not see the rating given by individual participants or the fee each paid. Some participants may be given a discount, but they should not be treated differently by the instructor.

Finally, the course manager, who is charged with the financial responsibility for the course, sees the income statement shown in Figure 2-5e.

Note that a great deal of data is shared in these views. The entities, attributes, and relationships contained in user views provide the input necessary for designing a database that supports user needs. Additional information that must be collected in the requirements analysis phase includes specifications for the volume of data within each view, the frequency of each report or query, and requirements for security, integrity, and response time.

Information-requirements analysis is the first and most important phase of the database design process. It is the most important phase because the ultimate effectiveness of the system depends on how accurately the information requirements and user views are specified initially. Specifications of user information requirements feed the entire design process and determine the ultimate form and content of the database system. Poor specifications result in a system that does not satisfy user needs and may lead to costly redesign, or even to cancellation of the project.

Logical Design Once the various views have been defined and specifications developed, they must be coordinated and organized into an integrated, cohesive system. This process is referred to as logical design. Logical design consists of three steps: (1) developing a data model for each user view, (2) integrating the entities, attributes, and relationships into a composite logical schema that describes the database for that module in terms unrelated to the software package being used, and (3) transforming the logical schema into a software schema expressed in the language of the chosen database management package.

Data modeling techniques used to develop a logical schema are described in the next chapter. Software schemas for relational databases and CODASYL databases are described in Chapters 7 and Appendix C??.

Part of a relational software schema written in a data description language similar to that used in the product known as ORACLEtm is shown in Figure 2-6. Even though relational systems view data quite differently at the logical level, the software schema is very similar to that of the CODASYL approach.

Physical Design The last step of the database design process is physical design. Physical design involves converting the software schema into a form implementable with the organization's particular hardware, operating system, and database management system. Physical design involves designing navigation paths, deciding which records to place in nearby portions of the storage devices, determining the size of physical records on storage devices and buffer areas in main memory to hold records, and implementing integrity and security requirements.

Other Database Techniques and Procedures

Many other procedures and techniques in addition to those for database design exist in a database environment. These include procedures for analyzing, implementing, and operating systems, designing and documenting programs, and training users. Among the tasks involving these procedures are collecting data to be entered into the system, performing manual edit checks to ensure the integrity of the data before it is entered, actually entering data, running programs necessary to update the database and generate reports from it, logging changes to the databases, making backup copies of the database so that it can be reconstructed if accidentally damaged or destroyed, and assuring that security procedures are followed. Most of these tasks are the responsibility of the database administrator. The role of the database administrator is briefly discussed later in this chapter, and in detail in Chapters 10 and 11.

DATABASES

As defined in Chapter 1, databases are integrated, shared collections of files. Databases are stored physically on the storage devices of the organization's computer system. The way the data is organized and stored on these devices is referred to as its **physical organization.**

People think of data in terms of its **logical organization.** There are many different ways that databases can be organized. Several of these are discussed in the next chapter. Many, perhaps most, database management packages use the relational approach to represent the logical organization of data. The **relational** approach describes databases using tables with special characteristics. These tables are called **relations.** Chapter 3 and all of Part Two are concerned with databases, so they are not discussed further here.

SOFTWARE

There are three different kinds of software involved in most database systems: database management packages, operating systems, and applications programs. The relationships among these three types of software were illustrated in Chapter 1. As Figure 1-2 illustrates, applications programs gain access to the database through the database management system. Most database management packages use the basic access methods of the host operating system to perform the necessary input-output operations on the data files themselves (some packages support their own access methods). In this section, database management packages are described, then application programming using a DBMS is briefly discussed. Refer to the Appendix for a discussion of access methods and operating systems.

Database Management Systems

The database management packages of interest in this book are complex and expensive. They are complex because many interrelated components are required to support the objectives discussed in Chapter 1. The minimum components of a package that could truly be called a database management system are a data dictionary for cataloging all the data items in the system, a data definition language for further describing the software schema and user views, one or more query languages suitable for end-users as well as programmers, a data manipulation language for use by programmers, a report generator language for efficient programming of management reports, and features to support data integrity, security, privacy, and recovery. Most packages support other functions as well. Some of these features are discussed elsewhere in this book.

Data Dictionary The data dictionary is used to maintain standard definitions of all data items within the scope of the database system. The purpose of the data dictionary is to enforce standard definitions of the data resources of the organization. Definitions include a precise description of the meaning of a data item name and a precise description of how values of each data item are stored in the computer system.

Some example data item descriptions are illustrated in Figure 2-8. Descriptions from this figure have been extracted from an actual data dictionary for a database built on Digital Equipment Corporation's (DEC) package called the Common Data Dictionary (CDD). The Common Data Dictionary can be used in conjunction with DEC database software, such as their Relational Data Base (RDB) package, or it can be used as a stand-alone system to simply catalog and standardize data definitions. Other data dictionary packages have interfaces to database systems of several vendors.

Some of the functions of a good data dictionary include (1) maintaining standard user definitions of the precise meaning of data items and standard computer definitions giving formats and data types, (2) maintaining cross-reference lists of data items used by applications programs and applications programs using a given data item, and (3) providing standard definitions of data items to applications programs when requested to do so. As for the latter, some data dictionary systems will generate appropriate Data Division statements describing specific data items or groups of data items on issuance of required commands by the applications program.

Data dictionaries are said to be either "active" or "passive." Passive dictionaries simply maintain user and computer definitions of data items. Active dictionaries include an interface to applications programs to support features such as generating record and field (data item) descriptions for application programs. Other properties of active and passive dictionaries are discussed in the chapters on database administration. Data dictionaries are discussed more fully in Chapter 10.

Data Definition Languages The data definition language is used to translate the logical schema into a software schema for the database management package in use. A standard data definition language (DDL) has been defined by CODASYL committees. Actually, two languages were defined--the schema DDL for specifying the composite schema, and the subschema DDL for specifying user views or data subsets of the schema. The Structured Query Language (SQL) has been adopted as the industry standard for relational languages. Many relational packages use SQL, which has features for describing the relations of a relational database. Relational data definition languages are discussed further in Chapter 7. A relational definition language was illustrated in Figure 2-7.

Query Languages Query languages such as SQL, used in most relational packages, are designed to give end-users direct access to the database. Query languages are one type of **fourth-generation language (4GL).** The first generation is machine (binary) language; the second, assembly language; and the third, programming languages such as C++, SQL, JAVA, or VISUALBASIC. Fourth-generation languages are much easier to learn and use than programming languages. Many developers of such languages claim that non-programmers can be doing useful work with such languages in a matter of hours and can be proficient in the language in a few days or weeks. However, fourth-generation languages lack the power and versatility of third-generation languages. For example, it would be difficult to write a program to simulate a rocket launch in SQL. Fourth-generation simulation languages are available for this, but it would be difficult to obtain flexible access to a database with a simulation language. A third-generation language could be used to do either job, but not as easily as with languages tailored to the problem.

Report Generators Report generators are languages designed for creating management reports. They generally include features for defining report titles, column headings, and row contents. They usually support "control breaks" that facilitate summarizing columns based on a data item such as a department name. Such languages are often simple enough that users are willing to learn them to write their own report programs. They may save many hours compared to what would be required to write the program in a standard programming language.

Security, Integrity, and Recovery Functions Database management systems should also include facilities for helping to maintain data integrity and security. Features for security are generally of two types--passwords or procedures for controlling access to the database, and data encryption, which makes the data unusable if it is obtained by an unauthorized user.

One way database management systems help ensure integrity is by providing the ability to specify legitimate values that data items may take. Once such ranges of values are input, the system checks values at the time of data entry to assure that they are acceptable. Of course, this does not totally eliminate incorrect data, but it helps to reduce the rate of error.

In the complex, multi-user mainframe environments of today it is almost impossible to totally protect a database system from accidental or intentional damage. Because so many people in the organization depend on shared systems, it is imperative that methods be provided for rapid **recovery** from damage to the database. Database management systems must provide features to assist in the recovery process. Techniques for doing so are described in Chapter 10.

Database Applications Software

The major difference between database applications software and conventional applications software is the use of special statements for data insertion, removal, retrieval, and modification in database systems. These data manipulation commands are usually embedded in a host language such as JAVA or C++. Many systems have precompilers that convert these programs with embedded data manipulation language (DML) statements into a program with only standard statements of the host language. The revised program is then compiled and executed.

Other differences in database programming and conventional programming in high-level languages include different ways of describing data and data relationships, and the use of report generators. In many cases, these features make the programming much easier than with the conventional file approach.

HARDWARE

The primary hardware components of interest in a database environment are the data storage devices and the central processing unit through which the database is accessed and controlled. Disks remain the most common storage medium for most database systems. Tape is the most common medium for backup. Some systems use faster devices, perhaps with semiconductor memory, to store relatively small amounts of data that is frequently accessed.

Some organizations are beginning to develop distributed database systems that consist of networks of computers, each of which maintains a portion of the total database. This approach may give users better access to and control of local information. It does, however, complicate the operating environment, because networking hardware and software are required. Security procedures become more complex, because the database is scattered among many different machines, which may be at different sites. If so, each machine must have its own security system.

During the early 1980s a special type of computer began to appear. These computers are called database machines, because they are specifically designed to perform some of the special operations required in a database environment. Database machines are discussed in Chapter 12. Storage devices such as tapes and disks are discussed in the Appendix.

PEOPLE: THE MOST IMPORTANT ELEMENT

Database systems exist to serve the information needs of people, both within and outside the organization. The types of people who use database systems are many and varied. Managers, for example, need information to make decisions. Clerical personnel must enter data from source documents into the system. Customers need information about bills or orders. Government agencies require financial and employment reports. Investors want information about the financial health of companies, and auditors are responsible for expressing an opinion as to whether the firm's financial statements are fairly presented.

The Primary Concerns of Users

Users are concerned with obtaining the information needed to get their jobs done. These jobs are as varied as the users themselves. Some tasks are directly concerned with the system, such as entering data into the database or retrieving data for personal use or for use by others. Even people who do not interact directly with the computer system are important users if the system must satisfy their information requirements. Such is the case for customers of a business or managers whose assistants actually retrieve information.

If the system is to serve end-users effectively and efficiently, it must be easy to learn and easy to use. It must also supply accurate, current data in a format suitable to users.

User Views of Data and Information

It should be noted that in Figures 2-5a through 2-5e no single user requires access to all the data needed by the entire organization. As is usually the case, each user or user group has a limited view of data and information in the system. The course promoter, for example, needs each prospect's name and address to mail out brochures. The instructor needs only the data on participants. The course manager needs information on course income and expenses, which is of no direct concern to the instructor or the promoter (although Marketing should know what courses are doing well). All this data can be stored in one integrated database, but no one user needs all the data.

We are taught as children that sharing is a nice thing to do, and this adage applies to data. Unfortunately, just as children may fight over toys or the largest cup of lemonade, users may fight over the rights to data. Users may derive power or prestige through their ability to control data. Logical attempts to share data may fail because of a failure to consider political battles over rights to data.

It may not be possible for the systems development staff to resolve such conflicts. Often they must be resolved at the highest levels in the organization. Many more database projects have run into serious difficulty over organizational issues than technical problems.

The Role of Top Management

In the development of organization-wide database systems, top management must become actively involved to gain impetus for the project and to resolve conflicts between users, or between users and the systems development staff. Top management must give visible support to the project and provide the resources necessary for successful project completion. Comprehensive database systems are expensive and require a heavy commitment of funds. Top management must be willing to commit the funds necessary for successful completion of the project. Top management may also have to step into frays between users and make unpopular decisions for the benefit of the organization.

The Systems Staff and Its Role

The job of the systems staff is to implement the elements of the database system master plan. This involves working with users to design and develop database system modules that support and integrate multiple end-user views of shared data. Typically, data in the various views is dispersed throughout the organization. Usually there has been little or no effort to organize this data in any comprehensive fashion. Data in this form is not suitable for computerization, nor can it be effectively managed as an organizational resource.

One major task in database system development is organizing and integrating the data in various views into a logical schema and implementing this schema with the organization's database management system. This is a complex task that may take months and must be carefully managed. Various personnel roles associated with logical database design, development, and operation are discussed next.

The Manager of Information Resources The person responsible for the information resources of the organization and for the development of its database systems must have a high-level position in the organizational chart. This position should be just beneath the vice-president in charge of all information processing in the enterprise. Titles such as "Manager of Information Resources" and "Chief Information Officer (CIO)" are beginning to appear for this position.

The primary responsibility of the manager of information resources is to work with a steering committee to develop the database system master plan and to oversee the implementation of that plan. The steering committee should be composed of the information resources manager, selected members of the systems staff, and key personnel from user areas. These users should be senior managers who understand the business of the organization and can provide the input necessary for the development of an effective master plan. Systems staff members on the committee must be user oriented and capable of communicating effectively with users.

Information Analysts The job of the information analyst is to work with end-users of the intended system to implement specific modules and applications of the overall database system. The skills needed by the information analyst include the ability to understand the business of the organization and to communicate effectively with users to determine user views, the information contained within those views, and the data required to support the views. The data must then be logically organized into a logical schema independent of any database management system.

Database Analysts The task of the database analyst is to transform the information analyst's logical schema into an efficient software schema and physical design for the particular database management system, operating system, and hardware in use. The database analyst must know how to get good performance from the organization's hardware and software. Database processing can be excessively slow with a poor physical design. This job is more technical than that of the information analyst in that it also encompasses physical design of the database required to support the module. The database analyst must be well versed in the particular database management system, operating system, and computer hardware in use.

Database Application Programmers Database application programmers (or database programmers) write programs in high-level languages such as VISUAL BASIC or C++ to perform the data processing activities required to support the module. These programs use the data manipulation language, the application generator, or the report generator to access the database, so database programmers must be familiar with
the database management system and database processing. We touch on aspects of database programming in this book, but we do not give detailed instruction in database programming.

Database Administrators The jobs of the systems staff described so far involve managing the overall operation or designing or programming part of the database system. Someone must also be responsible for managing the development of specific databases, as well as managing their operation and maintenance once they are put into production.

Database administrators are assigned the responsibility of managing specific databases in the overall database system. Database administration involves such tasks as providing documentation and training, developing security procedures, backup and recovery procedures, performance evaluation, and database reorganization when necessary. The database administrator must be familiar with the organization's DBMS and computer system and must be capable of working with users during the actual operation of the database system. Because this is such a critical role in successful implementation of database systems in organizations, a section of the chapter will discuss database administration issues, after a brief note on role sharing.

A Comment on Sharing Roles All the roles described in this section must be filled by someone. In many organizations, one person fills more than one role. The same person may, for example, be responsible for information analysis, logical database design, and database programming. Many other combinations of roles are also possible, depending on the size and philosophy of the organization and the management of data processing. Many organizations cannot afford, or are not large enough to require, different people for each role.

KEY ISSUES IN DATABASE ADMINISTRATION

Once a database system has been implemented in an organization, the task of meeting the daily operational challenges of maintaining the system follows. The maintenance function is an extremely complex activity. The essence of database administration is really the ability to adequately address the issues involved in the routine use of the database system without compromising future uses.

The key to effective database administration is twofold. The first step is to be prepared for potentially disastrous events that can affect the database system (discussed in Chapter 10). These are the events that cause the database system to be inoperative for some period of time. The second step is to anticipate and adequately prepare for requests that require changes in either the database system or the database system procedures. Because an organization is dynamic and the needs of users of the database system can be expected to change, requests for changes in the way the database system operates, or in what it produces, are inevitable.

Providing Documentation

The database administrator is the target of continual requests for information regarding effective and efficient use of the database system. In most cases, the requests are straightforward and can be satisfied by directing the person making the request to documentation, <u>if it exists.</u> The key is to have documentation available.

What type of documentation should be provided? Documentation can be divided into three types. First, is the database system master plan, which gives users a general context in which to place their specific needs and requirements. The master plan can also educate users who do not understand how their job relates to the operations of the entire organization.

Second, nontechnical documentation can support users who are concerned, not with how the database system operates, but with how to get the information they need to support the decision-making activities of their job. An example of this type of documentation could be an explanation of Personal Computer database system interfaces which would explain how to efficiently down-load data for analysis.

Finally, technical documentation gives detailed explanations of how the database system does what it does. Although this documentation is used predominantly by the database administration staff, it can also be of great use to other organizational database system users who have technical training or are computer literate. Access to some technical documentation may be controlled for reasons of security.

If possible, a central location for documentation should be provided. Even if no more than a bookcase in a particular room, there should be a specified place for documentation. In this way, users can take the first step on their own, without interrupting the database administrator or waiting for the database administration support staff to become available.

Providing Training and Overcoming Resistance

Documentation, however well written, is not enough to entirely support the needs of the organization. The database administrator must also provide periodic training sessions. Although this is an extremely vital function, even initial training is often neglected.

Initial training may be required because new users may actually oppose the database system. These users may see the new system as threatening, or they may be unwilling to learn new procedures for a job they have performed for years. In these cases, the database administrator must address these behavioral issues in order to successfully integrate the database approach into the organization.

This resistance, sometimes referred to as "social inertia," may come about simply because change is stressful and often resisted, but many factors are involved: (1) formal information is only a small component of the organizational decision process; (2) human information processing tends to be simple, experiential, and nonanalytical; (3) organizations are inherently complex, making large changes difficult to implement and institutionalize; (4) many individuals and groups within an organization tend to get their influence and autonomy from their control over information; (5) database systems may restrict lower management prerogative; (6) database systems may tend to increase a superior's ability to evaluate personnel; (7) information systems and computerization may be expected to lower the employment level in general and threaten job security in particular; and (8) database systems are complex and may be perceived as incomprehensible.

The thread tying these factors together is the observation that database systems are agents of organizational change, with far-reaching effects on many aspects of organizational functioning. Informal information, negotiations, and habits are important factors that are practically impossible to implement in a formal database system. When these factors are overlooked completely or threatened, social inertia may occur because the database system is perceived to have little relevance or importance concerning the "real" information flows in the organization.

The complexity of organizations leads to processes of change that are incremental and evolutionary. Large steps are inevitably avoided and resisted. In the case of database systems, this implies that implementation must be phased. Compromise and cooperation are essential.

Many units in organizations get their influence and autonomy from their control over information, so they do not want to lose control of it. When a database system designer is working toward the goal of integrating information controlled by different departments into a system of common databases, groups and individuals naturally view the system as a direct threat and respond accordingly.

Database systems may restrict a manager's prerogative, especially at lower levels of the organization. The degree of standardization and formalization of the role of supervisor may result in a reduction in supervisors' ability to shape their own jobs.

On a more personal note, resistant users may also see the system as a criticism of their own decision processes. In some cases, the integrated database system provides a user's superior with data that makes any mistake highly visible. There is a widespread view that advancing computerization will have a negative effect on job security.

Due to the complexity of these issues and of the database system itself, user training, education and preparation for change are critical in ensuring successful database implementation. Database administrators must deal effectively with social inertia. One technique for doing so is based on the Lewin-Schien model of organization change. In this model, organizational change is viewed as a process with three stages: **unfreezing, changing,** and **refreezing.**

Unfreezing entails the development of a felt need for change in the organization by encouraging users to recognize shortcomings of the existing system and benefits of a new system. Users then become open to trying something new. Instituting the change is the next stage. In this phase, the new database system is introduced and training is initiated. The third stage, refreezing, involves institutionalizing the change by integrating new attitudes and behaviors into existing organizational patterns and relationships.

Education is necessary at all levels of the organization. Top management must have a basic understanding of database concepts to be able to support the information systems department when necessary. Data processing personnel must understand that operating under a database environment is substantially different from conventional operation. And, of course, the end-users of the system at all levels must be trained to prepare them for the transition and to provide a basis for support in the wake of redistributed information flows.

Even experienced users may benefit from additional training. The benefits of advanced training (or even occasional retraining or refresher courses) are not obvious, and hence this type of training is often neglected. Users frequently learn just enough to do what they desire, with little regard for the efficiency of their actions. They may turn to other users for advice and help regarding use of unfamiliar database subsystems. This tendency results in a slow but steady propagation of inefficient techniques throughout the organization. Periodic advanced training of the database system users can therefore help improve the overall efficiency of the database system.

Arbitrating Requested Changes in the Database System

Database administration frequently requires trade-offs among people in the organization. Everyone wants the fastest response time they can have; regardless of the application, users always indicate that they would be able to do a better job if they could get their information faster. Unfortunately, some users must accept slower response times so that other users may have faster response.

A good example is an airline reservation database system. A user in the accounting department may be required to accept slower response times than a ticketing agent. This situation arises because a ticket agent's request for information is probably initiated to handle a prospective sale, whereas the accounting department's request involves a sale that has already been made. The airline could reasonably place higher value on fast response in handling a customer's order than on handling information regarding a sale that has been "sealed."

Another example is computer-based university registration processes. During registration, higher priority is given to requests that service the registration process than to normal student computing services. This situation arises because the administration gives higher value to expediting the registration-related activity during this particular time of the school year.

In any organization, there are some users who must accept slower response times than others for the good of the organization as a whole. In many cases, the database administrator must decide which users receive the faster response time.

Response time is not the only issue about which such decisions must be made. The amount of secondary storage area that users may have, the number of off-line peripherals (printers, plotters, tape drives) that users may allocate to their applications, the amount of primary memory that a user's application may command, the time of day that particular applications may run, and of course the specific data items in the database system that a user may access--these are all resource-related matters for decision.

Users from different parts of the organization frequently propose conflicting requests for changes to the database system. Several requests for increases in primary memory allocation, for example, can add up to too great a decrease in the response time of the database system for all users in the organization. A second common conflict occurs when the database administration function is overloaded with requests requiring the development of new applications within the database system. This happens as users become more aware of what information is available in the database system and how it can be used to help them perform more effectively. Because implementing all requested changes to the database system is typically impossible, some requests must be chosen over others. The database administrator must evaluate the requests and determine how best to resolve conflicts. The following list gives some of the major considerations involved in such decisions:

How important is this request relative to the organization's purpose and goals?

Are the user's claims accurate?

What other alternatives are available for producing the desired result?

Can the modification requested be completed in time to be useful?

Can a short-term solution be implemented while a detailed study is executed?

Would improvement in manual procedures solve the problem?

If work on this request is begun, what other requests will necessarily have to wait?

Is this request technologically feasible? Is it feasible given the organization's resources?

Above all, the database administrator must keep the objectives of the organization foremost and try to resolve disputes in a way that maintains the integrity of the overall database system.

Developing New Applications

The development of new applications of the database system should proceed according to standards specified by the database administrator, or the chief information officer, if one exists. A representative of the information administration function should be included in the design of any new application. This would normally be the person who will ultimately become the database administrator for that application.

The standards for application development should include (1) guidelines for the structure of any new program code and data dictionary entries, (2) the format of supplemental documentation, (3) a full range of tests to determine the reliability of the new application, and (4) a comprehensive review process by members of the organization who have not been involved in the actual development of the application. Access to all parts of the database system at the source code level should be limited to only one or two persons in the organization whose responsibility is to maintain the integrity of the database system at this level. Organizational policy should prevent integration of any new applications programs until all standards have been met.

Controlling Redundancy

Data redundancy leads to inconsistencies in a database system. As we saw in Chapter 1 (at the Conventional National Bank), inconsistencies lead to processing problems. Normalization complicates this issue because it results in an increase in redundant data items. Redundancy is also proliferated through inadequate standardization of the data items in a database, as when a data item occurs more than once because different database users have different names for it.

The database administrator must strive to avoid problems of redundancy. A well-developed data dictionary can help, since it should completely specify the characteristics of a data item, facilitating identification of similar items. However, periodic data dictionary reviews are still good policy; knowledge of the applications to the database system is necessary to identify the subtler redundancies.

THE USER'S VIEW

Users are the most important element of an information management system. Without users there is no need for a database system to exist. Information systems exist to serve the needs of users, so a database system is a success if it satisfies the information needs of its intended users. Many database system projects in the past have met with little success. Martin (1983) lists over 50 reasons for database system failures and corollary requisites for success; most of these relate to management or organizational issues, not to technical problems.

If users are not heavily involved in the planning and design of a database system, there is little chance of its success. Users have their own, limited view of information in the system. As consumers of information, they have their own tastes and preferences, just as consumers of any other commodity do. Their preferences relate to what information is consumed when, and how it is packaged. The only way the information analyst can know the views and preferences of users is to work closely with them during the design process.

Communication between user and analyst should be effective and straightforward. Neither user nor analyst should have hidden agendas or axes to grind--but this is not always the case. Before the database system is considered, users may have control over their own information. They may be hesitant to yield control, especially if the information relates to performance criteria or other sensitive data.

Similarly, people may have developed methods of hiding or masking poor performance data. If so, they surely will not want this data put into a widely accessible database where it might be used against them.

Users are also often concerned about the ownership of data in a shared database. Ownership and the privilege to access and use data may be a source of power, influence, and prestige. If this is the case, it may be necessary for the systems personnel to act as custodians of the data while ownership and other privileges remain in the hands of users.

If the analyst is charged with developing a database system in such an environment, then top management must become involved or the cooperation of users may never be gained. Analysts cannot resolve these issues alone. Only management can make decisions regarding what is best for the organization as a whole.

Metropolitan National Bank

The Board of Directors has given you the mandate to bring MetNat to the forefront of information technology use for competitive advantage. In response, you have indicated that you would like to begin a complete and thorough review of all of the information processing systems in the bank: manual and computer-based. You expect that a major reorganization of these systems is necessary, since MetNat does not currently operate with an integrated database system.

As a prelude to your investigation of the bank's information processing systems, you have been reviewing the bank's annual reports for the past few years. Excerpts are provided below:

> "Metropolitan National is committed to a complete integration of computing resources into every aspect of the organization's operations. This integration is to entail almost completely automated customer service, account management, human resources management, facilities management, and subsystems for supporting the strategic thrusts of the bank into new areas of competitive advantage, integrated as needed to maintain an accurate description of the entire enterprise."
>
> "We have several goals in each of these areas. We want to significantly reduce any amount of time associated with attending to the needs of our customers. Two-to-three day turnaround on customer inquiries is not sufficient. Our customers deserve better."
>
> "Account information should be available via telecommunications technology. Checks must be cleared within twenty-four hours, making funds available for our customers as soon as possible."
>
> "We cannot expect our employees to take good care of our customers if we neglect our employees. We intend, therefore, to invest in developing computer-based human resources management systems so that our employees' needs are well-maintained."

> "We will be able to offer competitive services only if we manage our costs well. Therefore we will look for ways to automate the management of our facilities and physical surroundings. We will investigate means for reducing our inefficiencies in dealing with our suppliers. We expect computer-based systems will provide new methods for creating new efficiencies and for making our management in these areas more effective."
>
> "We must begin to put the vast data resources of the bank to work. The data maintained by the bank provides a barometer by which to measure the Lincoln area. We must begin to build computer-based information systems that will help us continue to identify areas where the bank may gain strategic competitive advantage, thus allowing us to provide still better services to our customers."

The bank is functionally organized as follows:

The President of the bank has the ultimate responsibility for ensuring growth in the bank's assets. He reports to the Board of Directors, five business leaders who controlled 70% of the bank's original asset base. The President is assisted by an Executive Vice President. The Executive Vice President manages the bank's internal operations. Each functional area in the bank is controlled by a Vice President who reports to the Executive Vice President. The President and the Executive Vice President each have an Executive Assistant to support their daily activities.

One functional area of the bank provides administrative functions for managing the accounts of typical customers. This department is called the Accounts Administration Department. The head of the department is the Vice President of Accounts Administration. Several account managers oversee the duties associated with opening and closing accounts, monitoring account use for overdrafts, authorizing unusually large withdrawals, wire transfers, account consolidations. Tellers are employees of this department.

A second functional area of the bank is the Loans Administration Department. This department has two areas: one for business and commercial loans; one for personal loans. The Vice President of Loans Administration is the head of this department. There are two Personal Loan Managers and two Commercial Loan Managers. There are five Commercial Loan Clerks and three Personal Loan Clerks working in these areas.

Approximately 4% of the loans made by MetNat end up in default. These accounts are turned over to an external collection agency which has had almost a 50% success rate in collecting the loan. The agency receives 33% of all loans it ultimately collects.

A third functional area of the bank is called the Special Services Department. Several small areas are housed in the Special Services Department, which has a Vice President of Special Services. The bank maintains a small marketing staff in this department. The customer service area is also a component of this department, as is a financial planning group.

The marketing manager is assisted by three market analysts. The marketing department handles all of the bank's television, radio, and newspaper advertisements. This department conducts market surveys to determine how to best position the bank's services. It develops promotional brochures, and because of this expertise also handles production of literature that specifies the parameters of bank services. In other words, this department produces the brochures that explicitly outline the bank's responsibilities to customers. The marketing department oversees any promotional campaigns that the bank has, such as contests that occur to increase the number of accounts at the bank.

The customer service area handles inquiries and suggestions that are registered by customers. Inquiries come primarily in the form of questions regarding customers' monthly statements. Prospective customers are directed to the customer service representatives. These representatives determine the needs of the customer and determine which of the bank's services best fill those needs. The customer service area also administers the safe deposit boxes.

The financial planning group provides advice in personal financial services. There are three financial planners in the group. Recently, their primary role has been to establish individual retirement plans for customers. They also provide advice on matters of estate planning, tax deferred savings plans, and college savings plans. The financial planning group also provides assistance to customers that have exceeded their credit limits or have found themselves over-extended on credit. In these cases, the financial planning group will arrange a loan through the Loans Administration Department to consolidate the debt, and will work out a plan for repaying the loan.

Another major department at MetNat is the Computer Services Department. This is your department, and your title of Chief Information Officer is considered equivalent to the Vice President level. You have a manager of computer operations that oversees the actual day-to-day operation of the computer machinery. He has two employees whose titles are Systems Programmers. These three employees install and maintain all operating system and purchased software. You also have a manager of applications programming. She has a staff of three Programmer Analysts who maintain the applications that support the bank's operations.

Finally, a small human resources staff exists at MetNat. There is a Vice President of Human Resources. The manager of human resources has one assistant whose title is Human Resources Clerk.

Employees may arrange to have their paychecks deposited directly into their accounts. Employees have free checking, automatic overdraft protection, and pay no annual fee on credit cards issued by the bank.

MetNat's main office building is located in a relatively new office park development in the northwest part of the city. There are five branch offices, one located roughly in each of the north, south, east, and west suburban areas, and one in the downtown business district.

As you have been gathering your information about MetNat's goals and organization, you have learned that some employees are worried about your mandate. For example, the marketing department manager has indicated that her staff cannot understand why a computer-based system might be necessary. The general attitude in marketing is that the manual systems that have been used for years have been quite adequate. Also, these employees really have no computer skills, and they feel that using a computer-based system could cause them to "lose touch" with their market.

The customer service manager has echoed the marketing department's misgivings. In the past, customer service answered complaints from telephone or "walk-in" customers. A form was completed noting the customer's complaint, and the form was forwarded to the proper department. For example, if a customer had a complaint about her monthly statement, the form was sent to the demand accounts department. If the problem concerned a loan, the form went to the loan department. The department provided the information requested on the form, usually in 2-3 working days, and the form was returned to customer service. The customer service department then notified the customer, either by telephone or mail, depending on the customer's choice. Everyone seemed happy with this system, why change it?

The customer service staff has also raised concerns about being replaced by a computer. After all, they note, when you need a telephone number, it is provided by a computerized voice. What will keep MetNat from replacing them?

The bank tellers were also worried about their jobs. Rumors were spreading that MetNat intended to purchase more automated tellers and terminate some of the employees. Some wondered whether the entire teller operation might become automated. Stories were beginning to spread of banks controlled by robots.

You have decided to prepare and distribute a memorandum addressing these issues. The memorandum will be in the form of a master plan. These "problems" can be solved by providing the correct information to the employees at MetNat, but the information must also provide the "big picture" to the employees.

At this stage, your estimates of costs and benefits will be qualitative at best. Your implementation schedule must also be tentative, since you are not yet familiar with the computing resources. Try, however, to list as many major components as possible.

CHAPTER SUMMARY

The relationships among the five major elements of database systems may be summarized by referring to the ANSI/SPARC architecture in Figure 1-10. Scanning from the top down, the figure depicts an external (end-user) level, a conceptual level, and an internal level. Each of these levels deals with information and data at a different degree of abstraction.

The viewpoint of information users is reflected at the highest level of abstraction, that farthest from computerization. At the other extreme is the physical level, the representation of data in the storage devices of the computer system. The level in between--the conceptual level--serves as a bridge from the user to the hardware. Procedures at each successive level of abstraction make the representation more concrete and hence more suitable for use by computers.

When converting to a database approach, an organization should first develop a database system master plan. The master plan should state the goals and objectives of the organization, describe its structure, list its major entities, entity groups, and relationships between entities, lay out schedules for implementation, and estimate expected costs and benefits to be derived from the database system.

The individual modules and databases themselves must be carefully designed. Information requirements must be specified correctly, for they drive the entire design process. User views are merged into an integrated schema, which is converted to a physical design.

People are the most important element in a database environment. Users need information to perform their jobs. Database administrators and managers must create a climate in which analysts and users can work together to achieve the goals of database projects.

QUESTIONS AND EXERCISES

1. Name and briefly describe the five elements of a database system. How do they fit together into an integrated system?

2. Which of the five elements is most important? Why?

3. Why are careful planning and design crucial to the success of a database system?

4. Briefly describe the basic elements of database management software.

5. For whom are data manipulation languages intended? For whom are query languages intended?

6. What are the four generations of software?

7. Describe the basic elements of a database system master plan. Why is this an important document to the

organization?

8. Why is it important to design database systems in a modular fashion?

9. What is an entity chart and how is it useful in defining database system modules?

10. What are the three phases of the database design process? Which is the most important? Why?

11. What is meant by "view integration"? What is the output of the view integration process?

12. What is a software schema? How does it differ from a logical schema?

13. What is a data structure diagram? How are data structure diagrams related to logical schemas?

14. What are the primary concerns of users? Why do different users have different views of the same database?

15. What is the role of top management in a database system project? Why is the role of top management important?

16. What is the major role of the information analyst in a database system project? How does this differ from the role of the database analyst?

17. Describe the role of the database administrator and explain how it differs from that of the database analyst.

18. How might you go about treating the estimation of benefits in the following database applications?

> a. A private university developing a database on students expressing an interest in attending the university. What if the university were publicly supported?
> b. A database on donors to a public or a private university.
> c. Profiles on expenditure habits, income, preferences, and so on of customers of a large department store.
> d. A reservation system for an airline company.
> e. A reservation system for a hotel chain.

FOR FURTHER READING

Barrett, Stephanie. "Strategic Alternatives and Interorganizational Systems." Journal of Management Information Systems, Vol. 3, No. 3, Winter 1986-1987, pp.5-16. Discusses strategies some companies are using to link suppliers and customers directly into their database systems by putting terminals in the offices of other organizations. This is a new and effective information strategy.

Batiste, John, and John Jung. "Requirements, Needs, and Priorities: A Structured Approach for Determining MIS Project Definition." MIS Quarterly, Vol. 8, No. 4, 1984, pp.215-228. Describes a technique for requirements analysis for MIS projects in general. This technique is useful in database projects as well.

Benjamin, Robert, Charles Dickinson, and John Rockart. "The Changing Role of the Corporate Information Systems Officer." MIS Quarterly, Vol. 9, No. 3, September 1985, pp.177-188. Describes the position of corporate or chief information officer and how this role relates to database systems in organizations.

Doll, William J. "Avenues for Top Management Involvement in Successful MIS Development." MIS Quarterly, Vol. 9, No. 1, March 1985, pp.17-38. Discusses the need for top management involvement in large-scale MIS projects, such as database systems.

Drury, D.H. "An Evaluation of Data Processing Steering Committees." MIS Quarterly, Vol. 8, No. 4, December 1984, pp.257-266. Evaluates steering committees and gives guidelines on how to organize and operate them.

Jackson, Ivan. Corporate Information Management. Prentice-Hall, Englewood Cliffs, New Jersey, 1986. Thorough discussion of planning and the organizational environment in Chapters 1-4.

Martin, James. Managing the Data-Base Environment. Prentice-Hall, Englewood Cliffs, New Jersey, 1983. An excellent treatment of normalization and view integration with bubble charts in Chapters 10-14.

Tom, Paul. Managing Information as a Corporate Resource. Scott, Foresman and Company, Glenview, Illinois, 1987. Information strategic planning is discussed in Chapter 2, and the relationship of databases to the strategic plan is discussed in Chapter 3.

FIGURE 2-1
Analogy: Campus and Database System

FIGURE 2-2
Organizational Chart for the Professional Training Institute

FIGURE 2-3
Entity Chart for the Professional Training Institute

FIGURE 2-4
Entity Groups for the Professional Training Institute

FIGURE 2-5
PTI Database: (a) instructor's view; (b) participants' view; (c) course promoter's view; (d) course manager's view (showing course rating and fee for each participant); (e) course manager's view of PTI income statement (**figure continued on pp. 54-57**).

FIGURE 2-6 (Note: this is the old Figure 2-11)
Partial Relational Software Schema for PTI's Database

FIGURE 2-7 (Note: this is the old Figure 2-12)
Example Data Dictionary Entry for an Employee Record Description

Figure 2-1 Analogy: Campus and Database System

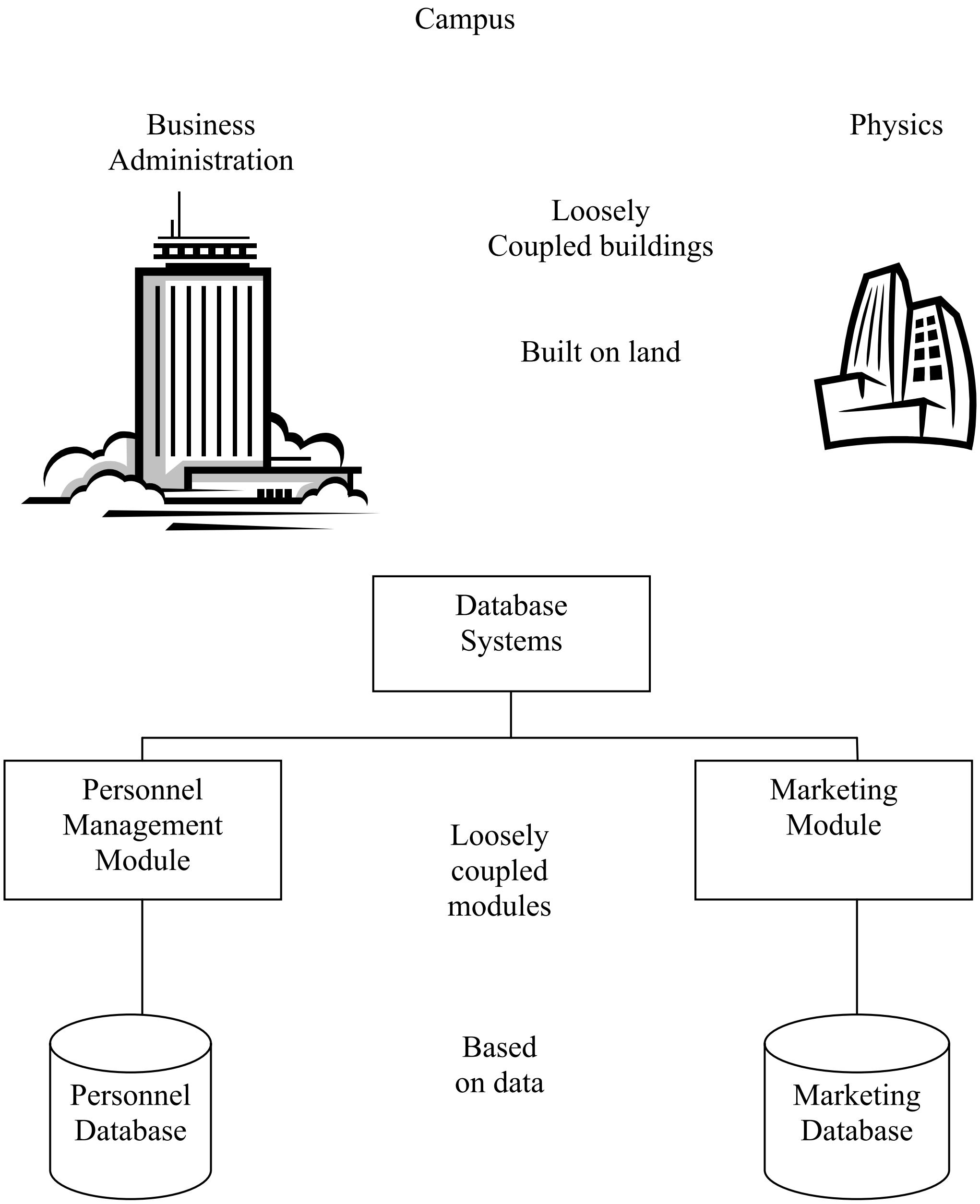

Table 2-1 Similarities between a Database System and a University Campus

Land-Use Plan	Database System Master Plan
Loosely coupled buildings	Loosely coupled modules
Buildings based on land	Modules based on databases
Shared by different people with different needs	Shared by different people with different views
Buildings designed, then built from the ground up	Top-down design, built from the database up
Structure expensive to modify	Structure expensive to modify
Some features easy to modify	Some features easy to modify
Stairways and halls provide route to rooms	Access paths provide route to data

Table 2-2 Suggested Outline for the Database System Master Plan

I. Organizational Environment
   A. Goals and Objectives
   B. Organizational Structure and Function
II. Elements of the Database System
   A. Major Entities and Relationships
   B. Entity Groups
   C. Modules
   D. User Groups
   E. Costs and Benefits
   F. Implementation Schedule
III. Summary

Table 2-3 Goals and Objectives for PTI

I. Continue offering currently viable courses
   A. Attract 50,000 participants
   B. Generate $10 million in gross revenue
II. Expand course offerings in computer topics
   A. Attract 8,000 participants
   B. Generate $2.4 million in gross revenue

Figure 2-2 Organizational Chart for the Professional Training Institute

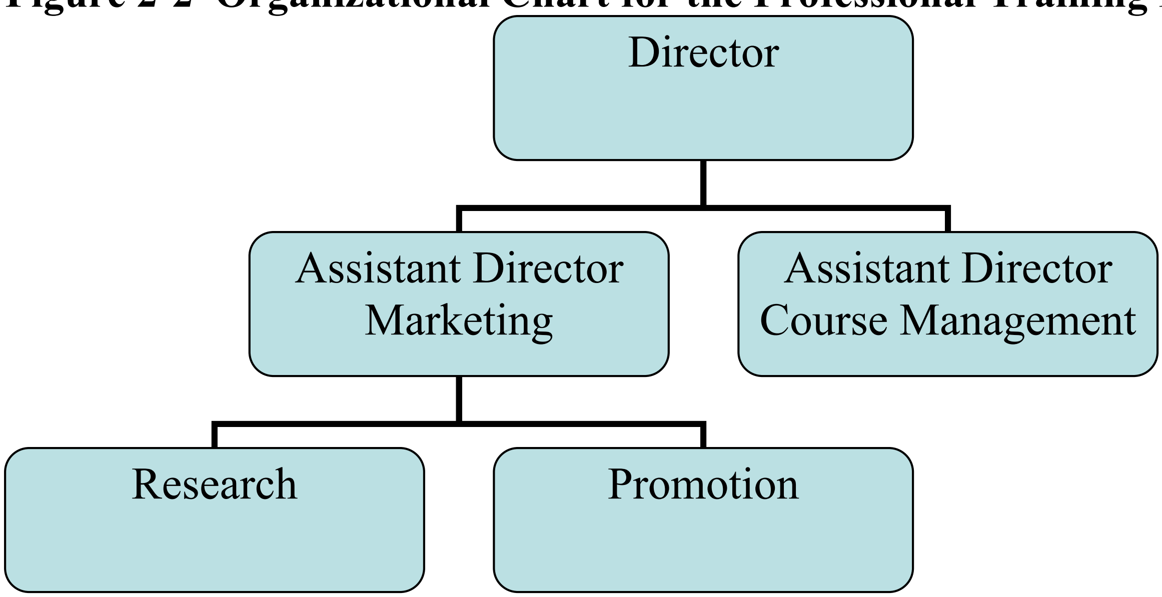

Figure 2-3 Entity Chart for the Professional Training Institute

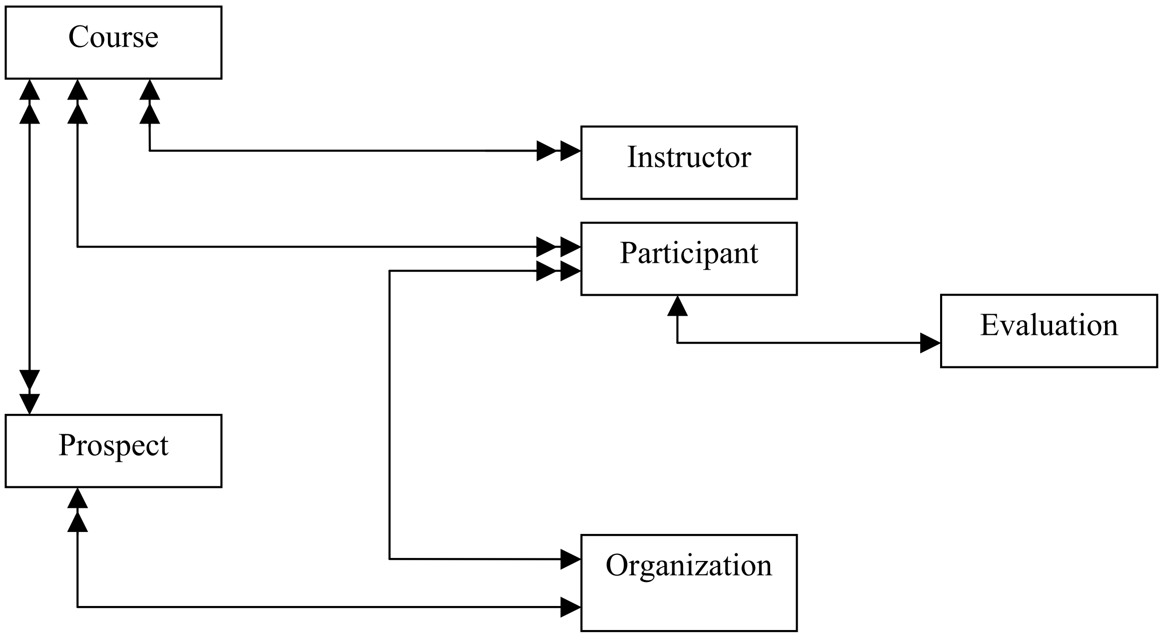

Figure 2-4 Entity Groups for the Professional Training Institute

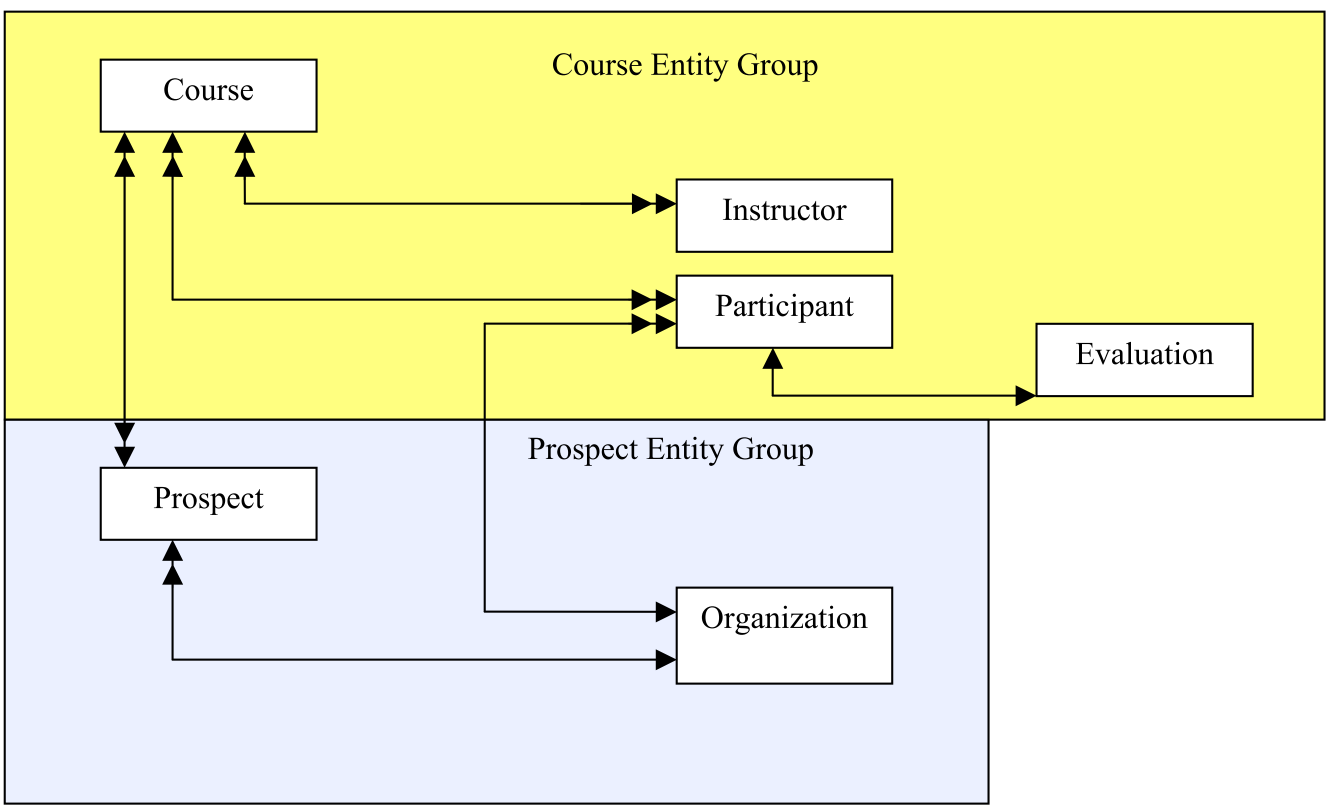

Figure 2-5a PTI Database: Instructor's View

FINAL GRADE ROLL AND EVALUATION REPORT

Course Advanced Database Design

Date : February 20 , 2010

Instructor : Daniel Wong

Textbook : Advanced Database Design
By Lilly Baldwin

Overall Rating : 4.5

Participant	Title	Organization	Grade
Cobb, Ted	Analyst	Sally's Bedding	Pass
Greise, Fred	Coach	Flatlands High	Pass
Kaufman, Julie	Manager	Hurts Trucks	Pass
Solinski, Bill	Chef	Henri's	Pass

Figure 2-5b PTI Database: Participants' View

Participant : Vianelli, Marc

Professor of Physics

Flatland U

9999 Spaceland Avenue

Flatland, California 90001

Course: Advanced Database Design

Instructor : Daniel Wong

Date : February 20 , 2010

Textbook : Advanced Database Design
By Lilly Baldwin

Fee : $950

Grade :

Rate the Course (Scale of 1 to 5):

Figure 2-5c PTI Database: Course Promoter's View

```
Course:  Advanced Database Design

Instructor :  Daniel Wong
              Ph. D.  University of Oklahoma, 1984

Date :  February 20 ,  2010

Textbook :  Advanced Database Design
             By Lilly Baldwin

Standard Fee :  $950
```

PROSPECT MAILING LIST

Mr. Alfred Adams
System Analyst
American Business Machines
One Computer Plaza
San Jose, Michigan 87905

Dr. Marc Vianelli
Professor of Physics
Flatland U
9999 Spaceland Avenue
Flatland, California 90001

Ms. Julie Kaufman
Manager
Hurts trucks
1212 Detroit Street
Acuff, Texas 78777

Figure 2-5d PTI Database: Course Manager's View

FINAL GRADE ROLL AND EVALUATION REPORT

Course: Advanced Database Design

Instructor : Daniel Wong

Date : February 20 , 2010

Textbook : Advanced Database Design
By Lilly Baldwin

Standard Fee : $950

Overall Rating : 4.5

Participant	Title	Organization	Grade	Rating	Fee
Cobb, Ted	Analyst	Sally's Bedding	Pass	4	950
Greise, Fred	Coach	Flatlands High	Pass	5	950
Kaufman, Julie	Manager	Hurts Trucks	Pass	5	950
Solinski, Bill	Chef	Henri's	Pass	5	950

Figure 2-5e PTI Database: Course Manager's View of the Income Statement

INCOME STATEMENT

Course: Advanced Database Design

Date : February 20 , 2010

Instructor : Daniel Wong

Textbook : Advanced Database Design
By Lilly Baldwin

Overall Rating : 4.5

Standard Fee : $950

Income from Participant Fees		$2850.00
Expenses		
Instructor's Fee	$800.00	

Textbooks	200.00	
Promotion	500.00	
Miscellaneous	150.00	
Total Expenses		$1650.00
Net Income		$1200.00

Figure 2-6 Partial Relational Software Schema for PTI's Database

```
CREATE TABLE INSTRUCTOR
INSTRUCTOR NAME (CHAR [20] NONULL)
HIGHEST-DEGREE (CHAR [5])
UNIVERSITY (CHAR [20])
YEAR (NUMBER [4])
CREATE TABLE COURSE
COURSE NAME (CHAR [20] NONULL)
DATE-OFFERED (CHAR [8])
TEXT (CHAR [20])
RATING (NUMBER [9 9])
FEE (NUMBER [999 ])
```

Figure 2-8 Example Data Dictionary Entry for an Employee Record Description

```
DEFINE RECORD COD$TOP CORPORATE EMPLOYEE_LIST DESCRIPTON IS
/*This record contains the master list of all employees */
EPLOYEE STRUCTURE
SS_NUMBER       DATATYPE IS UNSIGNED NUMBERIC SIZE IS 9 DIGITS.
NAME STRUCTURE
LAST_NAME       DATATYPE IS TEXT SIZE IS 15 CHARACTERS.
FIRST_NAME      DATATYPE IS TEXT SIZE IS 10 CHARACTERS.
MIDDLE_NAME     DATATYPE IS TEXT SIZE IS 1 CHARACTER
END NAME STRUCTURE
ADDRESS         COPY FROM CCD$TOP CORPORATE ADDRESS_RECORD
DEPT_CODE       DATATYPE IS UNSIGNED NUMBERIC SIZE IS 3 DIGITS.
END EMPLOYEE STRUCTURE
END EMPLOYEE_LIST RECORD
```