

Chapter 1

Database Systems

Discussion Focus

How often have your students heard that “you have only one chance to make a good first impression?” That’s why it’s so important to sell the importance of databases and the desirability of good database design during the first class session.

Start by showing your students that they interact with databases on a daily basis. For example, how many of them have bought anything using a credit card during the past day, week, month, or year? None of those transactions would be possible without a database. How many have shipped a document or a package via an overnight service or via certified or registered mail? How many have checked course catalogs and class schedules online? And surely all of your students registered for your class? Did anybody use a web search engine to look for – and find – information about almost anything? This point is easy to make: Databases are important because we depend on their existence to perform countless transactions and to provide information.

If you are teaching in a classroom equipped with computers, give some “live” performances. For example, you can use the web to look up a few insurance quotes or compare car prices and models. Incidentally, this is a good place to make the very important distinction between data and information. In short, spend some time discussing the points made in Section 1.1, "Why Databases?" and Section 1.2 “Data vs. Information.”

After demonstrating that modern daily life is almost inconceivable without the ever-present databases, discuss how important it is that the (database) transactions are made successfully, accurately, and quickly. That part of the discussion points to the importance of database design, which is at the heart of this book. If you want to have the keys to the information kingdom, you’ll want to know about database design and implementation. And, of course, databases don’t manage themselves ... and that point leads to the importance of the database administration (DBA) function. There is a world of exciting database employment opportunities out there.

After discussing why databases, database design, and database administration are important, you can move through the remainder of the chapter to develop the necessary vocabulary and concepts. The review questions help you do that ... and the problems provide the chance to test the newfound knowledge.

Answers to Review Questions

1. Define each of the following terms:

a. data

Raw facts from which the required information is derived. Data have little meaning unless they are grouped in a logical manner.

b. field

A character or a group of characters (numeric or alphanumeric) that describes a specific characteristic. A field may define a telephone number, a date, or other specific characteristics that the end user wants to keep track of.

c. record

A logically connected set of one or more fields that describes a person, place, event, or thing. For example, a CUSTOMER record may be composed of the fields CUST_NUMBER, CUST_LNAME, CUST_FNAME, CUST_INITIAL, CUST_ADDRESS, CUST_CITY, CUST_STATE, CUST_ZIPCODE, CUST_AREACODE, and CUST_PHONE.

d. file

Historically, a collection of file folders, properly tagged and kept in a filing cabinet. Although such manual files still exist, we more commonly think of a (computer) file as a *collection of related records* that contain information of interest to the end user. For example, a sales organization is likely to keep a file containing customer data. Keep in mind that the phrase *related records* reflects a relationship based on function. For example, customer data are kept in a file named CUSTOMER. The records in this customer file are related by the fact that they all pertain to customers. Similarly, a file named PRODUCT would contain records that describe products – the records in this file are all related by the fact that they all pertain to products. You would not expect to find customer data in a product file, or vice versa.

NOTE

Note: Field, record, and file are computer terms, created to help describe how data are stored in secondary memory. Emphasize that computer file data storage does not match the human perception of such data storage.

2. What is data redundancy, and which characteristics of the file system can lead to it?

Data redundancy exists when unnecessarily duplicated data are found in the database. For example, a customer's telephone number may be found in the customer file, in the sales agent file, and in the invoice file. Data redundancy is symptomatic of a (computer) file system, given its inability to represent and manage data relationships. Data redundancy may also be the result of poorly-designed databases that allow the same data to be kept in different locations. (Here's another opportunity to emphasize the need for good database design!)

3. What is data independence, and why is it lacking in file systems?

Data independence is a condition in which the programs that access data are not dependent on the data storage characteristics of the data. Systems that lack data independence are said to exhibit data dependence. File systems exhibit data dependence because file access is dependent on a file's data characteristics. Therefore, any time the file data characteristics are changed, the programs that access the data within those files must be modified.

Data independence exists when changes in the data characteristics don't require changes in the programs that access those data. File systems lack data independence because all data access programs are subject to change when any of the file system's data storage characteristics – such as changing a data type -- change.

4. What is a DBMS, and what are its functions?

A DBMS is best described as a collection of programs that manage the database structure and that control shared access to the data in the database. Current DBMSes also store the relationships between the database components; they also take care of defining the required access paths to those components. The functions of a current-generation DBMS may be summarized as follows:

- The DBMS stores the definitions of data and their relationships (metadata) in a data dictionary; any changes made are automatically recorded in the data dictionary.
- The DBMS creates the complex structures required for data storage.
- The DBMS transforms entered data to conform to the data structures in item 2.
- The DBMS creates a security system and enforces security within that system.
- The DBMS creates complex structures that allow multiple-user access to the data.
- The DBMS performs backup and data recovery procedures to ensure data safety.
- The DBMS promotes and enforces integrity rules to eliminate data integrity problems.
- The DBMS provides access to the data via utility programs and from programming languages interfaces.
- The DBMS provides end-user access to data within a computer network environment.

5. What is structural independence, and why is it important?

Structural independence exists when data access programs are not subject to change when the file's structural characteristics, such as the number or order of the columns in a table, change. Structural independence is important because it substantially decreases programming effort and program maintenance costs.

6. Explain the differences between data, information, and a database

Data are raw facts. Information is processed data to reveal the meaning behind the facts. Let's summarize some key points:

- Data constitute the building blocks of information.
- Information is produced by processing data.
- Information is used to reveal the meaning of data.
- Good, relevant, and timely information is the key to good decision making.
- Good decision making is the key to organizational survival in a global environment.

A database is a computer structure for storing data in a shared, integrated fashion so that the data can be transformed into information as needed.

7. What is the role of a DBMS, and what are its advantages? What are its disadvantages?

A **database management system (DBMS)** is a collection of programs that manages the database structure and controls access to the data stored in the database. Figure 1.2 (shown in the text) illustrates that the DBMS serves as the intermediary between the user and the database. The DBMS receives all application requests and translates them into the complex operations required to fulfill those requests. The DBMS hides much of the database's internal complexity from the application programs and users. The application program might be written by a programmer using a programming language such as COBOL, Visual Basic, or C++, or it might be created through a DBMS utility program.

Having a DBMS between the end user's applications and the database offers some important advantages. First, the DBMS enables the data in the database *to be shared* among multiple applications or users. Second, the DBMS *integrates* the many different users' views of the data into a single all-encompassing data repository.

Because data are the crucial raw material from which information is derived, you must have a good way of managing such data. As you will discover in this book, the DBMS helps make data management more efficient and effective. In particular, a DBMS provides advantages such as:

- *Improved data sharing.* The DBMS helps create an environment in which end users have better access to more and better-managed data. Such access makes it possible for end users to respond quickly to changes in their environment.
- *Better data integration.* Wider access to well-managed data promotes an integrated view of the organization's operations and a clearer view of the big picture. It becomes much easier to see how actions in one segment of the company affect other segments.
- *Minimized data inconsistency.* **Data inconsistency** exists when different versions of the same data appear in different places. For example, data inconsistency exists when a company's sales department stores a sales representative's name as "Bill Brown" and the company's personnel department stores that same person's name as "William G. Brown" or when the company's regional sales office shows the price of product "X" as \$45.95 and its national sales office shows the same product's price as \$43.95. The probability of data inconsistency is greatly reduced in a properly designed database.
- *Improved data access.* The DBMS makes it possible to produce quick answers to ad hoc queries. From a database perspective, a **query** is a specific request for data manipulation (for

example, to read or update the data) issued to the DBMS. Simply put, a query is a question and an **ad hoc query** is a spur-of-the-moment question. The DBMS sends back an answer (called the **query result set**) to the application. For example, end users, when dealing with large amounts of sales data, might want quick answers to questions (ad hoc queries) such as:

- What was the dollar volume of sales by product during the past six months?
 - What is the sales bonus figure for each of our salespeople during the past three months?
 - How many of our customers have credit balances of \$3,000 or more?
- *Improved decision making.* Better-managed data and improved data access make it possible to generate better quality information, on which better decisions are based.
 - *Increased end-user productivity.* The availability of data, combined with the tools that transform data into usable information, empowers end users to make quick, informed decisions that can make the difference between success and failure in the global economy.

The advantages of using a DBMS are not limited to the few just listed. In fact, you will discover many more advantages as you learn more about the technical details of databases and their proper design.

Although the database system yields considerable advantages over previous data management approaches, database systems do carry significant disadvantages. For example:

- *Increased costs.* Database systems require sophisticated hardware and software and highly skilled personnel. The cost of maintaining the hardware, software, and personnel required to operate and manage a database system can be substantial. Training, licensing, and regulation compliance costs are often overlooked when database systems are implemented.
- *Management complexity.* Database systems interface with many different technologies and have a significant impact on a company's resources and culture. The changes introduced by the adoption of a database system must be properly managed to ensure that they help advance the company's objectives. Given the fact that databases systems hold crucial company data that are accessed from multiple sources, security issues must be assessed constantly.
- *Maintaining currency.* To maximize the efficiency of the database system, you must keep your system current. Therefore, you must perform frequent updates and apply the latest patches and security measures to all components. Because database technology advances rapidly, personnel training costs tend to be significant.
- *Vendor dependence.* Given the heavy investment in technology and personnel training, companies might be reluctant to change database vendors. As a consequence, vendors are less likely to offer pricing point advantages to existing customers, and those customers might be limited in their choice of database system components.
- *Frequent upgrade/replacement cycles.* DBMS vendors frequently upgrade their products by adding new functionality. Such new features often come bundled in new upgrade versions of the software. Some of these versions require hardware upgrades. Not only do the upgrades themselves cost money, but it also costs money to train database users and administrators to properly use and manage the new features.

8. List and describe the different types of databases.

The focus is on Section 1-3b, TYPES OF DATABASES. Organize the discussion around the number of users, database site location, and data use:

- Number of users
 - Single-user
 - Multiuser
 - Workgroup
 - Enterprise
- Database site location
 - Centralized
 - Distributed
 - Cloud-based
- Type of data
 - General-purpose
 - Discipline-specific
- Database use
 - Transactional (production) database (OLTP)
 - Data warehouse database (OLAP)
- Degree of data structure
 - Unstructured data
 - Structured data

For a description of each type of database, please see section 1-3b.

9. What are the main components of a database system?

The basis of this discussion is Section 1-7a, THE DATABASE SYSTEM ENVIRONMENT. Figure 1.10 provides a good bird's eye view of the components. Note that the system's components are hardware, software, people, procedures, and data.

10. What is metadata?

Metadata is data about data. That is, metadata define the data characteristics such as the data type (such as character or numeric) *and the relationships that link the data*. Relationships are an important component of database design. What makes relationships especially interesting is that they are often defined by their environment. For instance, the relationship between EMPLOYEE and JOB is likely to depend on the organization's definition of the work environment. For example, in some organizations an employee can have multiple job assignments, while in other organizations – or even in other divisions within the same organization – an employee can have only one job assignment.

The details of relationship types and the roles played by those relationships in data models are defined and described in Chapter 2, Data Models.”. Relationships will play a key role in subsequent chapters. You cannot effectively deal with database design issues unless you address relationships.

11. Explain why database design is important.