Database Systems 14th edition Coronel Solutions Manual

SKU: 9780357673034-SOLUTIONS

Solution and Answer Guide

CORONEL AND MORRIS, DATABASE SYSTEMS: DESIGN, IMPLEMENTATION AND MANAGEMENT, ©2023, 9780357673034; CHAPTER 1: DATABASE SYSTEMS

TABLE OF CONTENTS

ANSWERS TO REVIEW QUESTIONS

1. Define each of the following terms:

Answer:

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

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?

   Answer: 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?

   Answer: 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?

   Answer: 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 DBMSs 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 the previous item.
   - 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 minimize 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?

   Answer: 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 among data, information, and a database.

   Answer: 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?

   Answer: 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 1.4 (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.

- *Improved data security*. A DBMS provides a framework for better enforcement of data privacy and security policies.
- *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 (e.g., 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 database 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.

   Answer: 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?

   Answer: 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?

    Answer: Metadata is data about data. That is, metadata defines 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.

    Answer: The focus is on Section 1-4, Why Database Design Is Important. Explain that modern database and applications development software is so easy to use that many people can quickly learn to implement a simple database and develop simple applications within a week or so, without giving design much thought. As data and reporting requirements become more complex, those same people will simply (and quickly!) produce the required add-ons. That's how data redundancies and all their attendant anomalies develop, thus reducing the "database" and its applications to a status worse than useless. Stress these points:

    - Good applications can't overcome bad database designs.
    - The existence of a DBMS does not guarantee good data management, nor does it ensure that the database will be able to generate correct and timely information.
    - Ultimately, the end user and the designer decide what data will be stored in the database.

    A database created without the benefit of a detailed blueprint is unlikely to be satisfactory. Pose this question: would you think it is smart to build a house without the benefit of a blueprint? So why would you want to create a database without a blueprint? (Perhaps it would be OK to build a chicken coop without a blueprint, but would you want your house to be built the same way?)

12. What are the potential costs of implementing a database system?

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

- *Increased acquisition and operating 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.
- *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 database 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 may be reluctant to change database vendors. As a consequence, vendors are less likely to offer pricing point advantages to existing customers and those customers may be limited in their choice of database system components.
- *Frequent upgrade/replacement cycles*. Vendors come up with new features, often included in new versions. Such versions frequently require processing and hardware upgrades. Such upgrades come with additional costs on dollars, personnel time, and downtime.

13. Use examples to compare and contrast unstructured and structured data. Which type is more prevalent in a typical business environment?

Answer: Unstructured data are data that exist in their original (raw) state, that is, in the format in which they were collected. Therefore, unstructured data exist in a format that does not lend itself to the processing that yields information. **Structured data** are the result of taking unstructured data and formatting (structuring) such data to facilitate storage, use, and the generation of information. You apply structure (format) based on the type of processing that you intend to perform on the data. Some data might be not ready (unstructured) for some types of processing, but they might be ready (structured) for other types of processing. For example, the data value 37890 might refer to a zip code, a sales value, or a product code. If this value represents a zip code or a product code and is stored as text, you cannot perform mathematical computations with it. On the other hand, if this value represents a sales transaction, it is necessary to format it as numeric.

If invoices are stored as images for future retrieval and display, you can scan them and save them in a graphic format. On the other hand, if you want to derive information such as monthly totals and average sales, such graphic storage would not be useful. Instead, you could store the invoice data in a (structured) spreadsheet format so that you can perform the requisite computations.

Based on sheer volume, most data are unstructured or semistructured. Data for conducting actual business transactions are usually structured.

14. What are some basic database functions that a spreadsheet cannot perform?

    Answer: Spreadsheets do not support self-documentation through metadata, enforcement of data types or domains to ensure consistency of data within a column, defined relationships among tables, or constraints to ensure consistency of data across related tables. It is important to note that newer versions of MS Office Excel come with new features such as PowerQuery and PowerBI that add more database-like data management functionality to the Excel spreadsheet.

15. What common problems do a collection of spreadsheets created by end users share with the typical file system?

    Answer: A collection of spreadsheets shares several problems with the typical file system. The first problem is that end users create their own, private copies of the data, which creates issues of data ownership. This situation also creates islands of information where changes to one set of data are not reflected in all of the copies of the data. This leads to the second problem—lack of data consistency. Because the data in various spreadsheets may be intended to represent a view of the business environment, a lack of consistency in the data may lead to faulty decision making based on inaccurate data.

16. Explain the significance of the loss of direct, hands-on access to business data that end users experienced with the advent of computerized data repositories.

    Answer: Users lost direct, hands-on access to the business data when computerized data repositories were developed because the IT skills necessary to directly access and manipulate the data were beyond the average user's abilities, and because security precautions restricted access to the shared data. This was significant because it removed users from direct data manipulation and introduced significant time delays for data access. The trade-off of data access versus data security often pays off due to the increasing emphasis in security imposed by the likes of data breaches and data hacks. When users need answers to business questions from the data, necessity often does not give them the luxury of time to wait days, weeks, or even months for the required reports. The desire to return hands-on access to the data to the users, among other drivers, helped to propel the development of database systems. While database systems have greatly improved the ability of users to directly access data, the need to quickly manipulate data for themselves has led to the problem of spreadsheets being used when databases are needed.

17. Explain why the cost of ownership may be lower with a cloud database than with a traditional, company database.

    Answer: Cloud databases reside on the Internet instead of within the organization's own network infrastructure. This can reduce costs because the organization is not required to purchase and maintain the hardware and software necessary to house the database and support the necessary levels of system performance. Companies typically experience savings in hardware, people, and management while investing those savings in increasing data analytics and business intelligence. However, companies must ensure that cloud providers comply with all required data insurance, security, and privacy regulations.

ANSWERS TO PROBLEMS

ONLINE CONTENT

The file structures you see in this problem set are simulated in a Microsoft Access database named Ch01_Problems, available at *www.cengage.com*.

Given the file structure shown in Figure P1.1, answer Problems 1–4.

FIGURE P1.1 The File Structure for Problems 1-4

PROJECT_CODE	PROJECT_MANAGER	MANAGER_PHONE	MANAGER_ADDRESS	PROJECT_BID_PRICE
21-5Z	Holly B. Parker	904-338-3416	3334 Lee Rd., Gainesville, FL 37123	16833460.00
25-2D	Jane D. Grant	615-898-9909	218 Clark Blvd., Nashville, TN 36362	12500000.00
25-5A	George F. Dorts	615-227-1245	124 River Dr., Franklin, TN 29185	32512420.00
25-9T	Holly B. Parker	904-338-3416	3334 Lee Rd., Gainesville, FL 37123	21563234.00
27-4Q	George F. Dorts	615-227-1245	124 River Dr., Franklin, TN 29185	10314545.00
29-2D	Holly B. Parker	904-338-3416	3334 Lee Rd., Gainesville, FL 37123	25559999.00
31-7P	William K. Moor	904-445-2719	216 Morton Rd., Stetson, FL 30155	56850000.00

1. How many records does the file contain? How many fields are there per record?

   Answer: The file contains seven records (21-5Z through 31-7P) and each of the record is composed of five fields (PROJECT_CODE through PROJECT_BID_PRICE).

2. What problem would you encounter if you wanted to produce a listing by city? How would you solve this problem by altering the file structure?

   Answer: The city names are contained within the MANAGER_ADDRESS attribute and decomposing this character (string) field at the application level is cumbersome at best. (Queries become much more difficult to write and take longer to execute when internal string searches must be conducted.) If the ability to produce city listings is important, it is best to store the city name as a separate attribute.

3. If you wanted to produce a listing of the file contents by last name, area code, city, state, or zip code, how would you alter the file structure?

   Answer: The more we divide the address into its component parts, the greater its information capabilities. For example, by dividing MANAGER_ADDRESS into its component parts (MGR_STREET, MGR_CITY, MGR_STATE, and MGR_ZIP), we gain the ability to easily select records on the basis of zip codes, city names, and states. Similarly, by subdividing the MANAGER name into its components MGR_LASTNAME, MGR_FIRSTNAME, and MGR_INITIAL, we gain the ability to produce more efficient searches and listings. For example, creating a phone directory is easy when you can sort by last name, first name, and initial. Finally, separating the area code and the phone number will yield the ability to efficiently group data by area codes. Thus MGR_PHONE might be decomposed into MGR_AREA_CODE and MGR_PHONE. The more you decompose the data into their component parts, the greater the search flexibility. Data that are decomposed into their most basic components are said to be ***atomic***.

4. What data redundancies do you detect? How could those redundancies lead to anomalies?

   Answer: Note that the manager named Holly B. Parker occurs three times, indicating that she manages three projects coded 21-5Z, 25-9T, and 29-2D, respectively. (The occurrences indicate that there is a 1:M relationship between PROJECT and MANAGER: each project is managed by only one manager but, apparently, a manager may manage more than one project.) Ms. Parker's phone number and address also occur three times. If Ms. Parker moves and/or changes her phone number, these changes must be made more than once *and they must all be made correctly … without missing a single occurrence*. If any occurrence is missed during the change, the data are "different" for the same person. After some time, it may become difficult to determine what the correct data are. In addition, multiple occurrences invite misspellings and digit transpositions, thus producing the same anomalies. The same problems exist for the multiple occurrences of George F. Dorts.

5. Identify and discuss the serious data redundancy problems exhibited by the file structure shown in Figure P1.5.

   Answer:

FIGURE P1.5 The File Structure for Problems 5–8

PROJ_NUM	PROJ_NAME	EMP_NUM	EMP_NAME	JOB_CODE	JOB_CHG_HOUR	PROJ_HOURS	EMP_PHONE
1	Hurricane	101	John D. Newson	EE	85.00	13.3	653-234-3245
1	Hurricane	105	David F. Schwann	CT	60.00	16.2	653-234-1123
1	Hurricane	110	Anne R. Ramoras	CT	60.00	14.3	615-233-5568
2	Coast	101	John D. Newson	EE	85.00	19.8	653-234-3254
2	Coast	108	June H. Sattlemeir	EE	85.00	17.5	905-554-7812
3	Satellite	110	Anne R. Ramoras	CT	62.00	11.6	615-233-5568
3	Satellite	105	David F. Schwann	CT	26.00	23.4	653-234-1123
3	Satelite	123	Mary D. Chen	EE	85.00	19.1	615-233-5432
3	Satellite	112	Allecia R. Smith	BE	85.00	20.7	615-678-6879

NOTE

It is not too early to begin discussing proper structure. For example, you may focus student attention on the fact that, ideally, each row should represent a single entity. Therefore, each row's fields should define the characteristics of one entity, rather than include characteristics of several entities. The file structure shown here includes characteristics of multiple entities. For example, the JOB_CODE is likely to be a characteristic of a JOB entity. PROJ_NUM and PROJ_NAME are clearly characteristics of a PROJECT entity. Also, since (apparently) each project has more than one employee assigned to it, the file structure shown here shows multiple occurrences for each of the projects. (Hurricane occurs three times, Coast occurs twice, and Satellite occurs four times.)

At first glance, the file structure in Figure P1.5 seems appropriate from the reporting point of view. After all, the columns contain a single value, there are no multi-value cells, all data in the columns are the same data type, and each row conveys the needed information (who works in each project, their role, the charge per hour, and the hours worked). However, we need to approach this from the

designer and the data processing point of view. Is the file structure providing info about one or multiple entities? It clearly shows information for multiple entities: project, employees, job roles, hours worked. Therefore, from the processing point of view, this is the ground for data duplication and anomalies. The file's poor structure sets the stage for multiple anomalies. For example, if the charge for JOB_CODE = EE changes from $85.00 to $90.00, that change must be made twice. Also, if employee June H. Sattlemeier is deleted from the file, you also lose information about the existence of her JOB_CODE = EE, its hourly charge of $85.00, and the PROJ_HOURS = 17.5. The loss of the PROJ_HOURS value will ultimately mean that the Coast project costs are not being charged properly, thus causing a loss of PROJ_HOURS*JOB_CHG_HOUR = 17.5 × $85.00 = $1,487.50 to the company.

Incidentally, note that the file contains different JOB_CHG_HOUR values for the same CT job code, thus illustrating the effect of changes in the hourly charge rate over time. The file structure appears to represent transactions that charge project hours to each project. However, the structure of this file makes it difficult to avoid update anomalies and it is not possible to determine whether a charge change is *accurately* reflected in each record. Ideally, a change in the hourly charge rate would be made in only one place and this change would then be passed on to the transaction based on the hourly charge. Such a structural change would ensure the historical accuracy of the transactions.

You might want to emphasize that the recommended changes require a lot of work in a file system.

6. Looking at the EMP_NAME and EMP_PHONE contents in Figure P1.5, what change(s) would you recommend?

   Answer: A good recommendation would be to make the data more *atomic*. That is, break up the data components whenever possible. For example, separate the EMP_NAME into its components EMP_FNAME, EMP_INITIAL, and EMP_LNAME. This change will make it much easier to organize employee data through the employee name component. Similarly, the EMP_PHONE data should be decomposed into EMP_AREACODE and EMP_PHONE. For example, breaking up the phone number 653-234-3245 into the area code 653 and the phone number 234-3245 will make it much easier to organize the phone numbers by area code. (If you want to print an employee phone directory, the more atomic employee name data will make the job much easier.)

7. Identify the various data sources in the file you examined in Problem 5.

   Answer: Given their answers to Problem 5 and some additional scrutiny of Figure P1.5, your students should be able to identify these data sources:

   - Employee data such as names and phone numbers.
   - Project data such as project names. If you start with an EMPLOYEE file, the project names clearly do not belong in that file. (Project names are clearly *not* employee characteristics.)
   - Job data such as the job charge per hour. If you start with an EMPLOYEE file, the job charge per hour clearly does not belong in that file. (Hourly charges are clearly *not* employee characteristics.)
   - The project hours, which are most likely the hours worked by the employee for that project. (Such hours are associated with a work product, not the employee per se.)

8. Given your answer to Problem 7, what new files should you create to help eliminate the data redundancies found in the file shown in Figure P1.5?

Answer: The data sources are probably the PROJECT, EMPLOYEE, JOB, and CHARGE. The PROJECT file should contain project characteristics such as the project name, the project manager/coordinator, the project budget, and so on. The EMPLOYEE file might contain the employee names, phone number, address, and so on. The JOB file would contain the billing charge per hour for each of the job types—a database designer, an applications developer, and an accountant would generate different billing charges per hour. The CHARGE file would be used to keep track of the number of hours by job type that will be billed for each employee who worked on the project.

9. Identify and discuss the serious data redundancy problems exhibited by the file structure shown in Figure P1.9. (The file is meant to be used as a teacher class assignment schedule. One of the many problems with data redundancy is the likely occurrence of data inconsistencies—two different initials have been entered for the teacher named Maria Cordoza.)

   Answer:

FIGURE P1.9 The File Structure for Problems 9 and 10

BUILDING_CODE	ROOM_CODE	TEACHER_LNAME	TEACHER_FNAME	TEACHER_INITIAL	DAYS_TIME
KOM	204E	Williston	Horace	G	MWF 8:00-8:50
KOM	123	Cordoza	Maria	L	MWF 8:00-8:50
LDB	504	Patroski	Donald	J	TTh 1:00-2:15
KOM	34	Hawkins	Anne	W	MWF 10:00-10:50
JKP	225B	Risell	James		TTh 9:00-10:15
LDB	301	Robertson	Jeanette	P	TTh 9:00-10:15
KOM	204E	Cordoza	Maria	I	MWF 9:00-9:50
LDB	504	Williston	Horace	G	TTh 1:00-2:15
KOM	34	Cordoza	Maria	L	MWF 11:00-11:50
LDB	504	Patroski	Donald	J	MWF 2:00-2:50

Note that the teacher characteristics occur multiple times in this file. For example, the teacher named Maria Cordoza's first name, last name, and initial occur three times. If changes must be made for any given teacher, those changes must be made multiple times. All it takes is one incorrect entry or one forgotten change to create data inconsistencies. Redundant data are not a luxury you can afford in a data environment.

10. Given the file structure shown in Figure P1.9, what problem(s) might you encounter if building KOM were deleted?

    Answer: You would lose all the time assignment data about teachers Williston, Cordoza, and Hawkins, as well as the KOM rooms 204E, 123, and 34. Furthermore, you will lose all references to Anne Hawkins and Maria Cordoza. Here is yet another good reason for keeping data about specific entities in their own tables! This kind of an anomaly is known as a *deletion anomaly*.

11. Using your school's student information system, print your class schedule. The schedule probably would contain the student identification number, student name, class code, class name, class credit hours, class instructor name, the class meeting days and times, and the class room number. Use Figure P1.11 as a template to complete the following actions.

Answer:

FIGURE P1.11 Student Schedule Data Format

STU_ID	STU_ NAME	CLASS_ CODE	CLASS_ NAME	CLASS_ CREDHRS	INSTR_ NAME	CLASS_ DAYS	CLASS_ TIMES	ROOM

a. Create a spreadsheet using the template shown in Figure P1.11 and enter your current class schedule.
b. Enter the class schedule of two of your classmates into the same spreadsheet.
c. Discuss the redundancies and anomalies caused by this design.

This could be a good “mini-group” problem—groups of three students maximum. Ask them to create their individual class schedules in separate spreadsheets and then, a single spreadsheet containing all their class schedules. This exercise should incentivize “group discussion” and discover data anomalies and brainstorm better ways to store the class schedule data.

Students are likely to use MS Excel or Google Sheets to create a simple tabular spreadsheet containing the data outlined in Figure P1.11. The rows of the spreadsheet(s) will represent each one of the classes they are taking.

Ask the students to generate a roster for students taking the database class. Students are likely to identify the redundancies around the class information since all three schedules (the student’s own schedule plus the schedules of the two classmates) will have at least the database class in common. This easily leads to discussions of separating the data into at least two tables in a database. However, that still leaves the redundancies of student data with each class that they are taking. Astute students might realize that this is analogous to the Employee Skills Certification shown in Figures 1.5 and 1.6, such that a table for student data, a table for class data, and a table to relate the students and classes are appropriate.

For more challenging work, ask them to create a report of the schedule of classes per room. What fields do they need to add to this report, and in what order? Do they have what they need to sort this so the report shows the schedule by day and, within each day, by the time of day?

Instructor Manual Chapter 1

CORONEL AND MORRIS, DATABASE SYSTEMS: DESIGN, IMPLEMENTATION AND MANAGEMENT, 14e, ©2023, 9780357673034; CHAPTER 1: DATABASE SYSTEMS

TABLE OF CONTENTS

PURPOSE AND PERSPECTIVE OF THE CHAPTER

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. Students can use the internet to look for COVID data, for example https://covid.cdc.gov/covid-data-tracker/#datatracker-home and think about how that data is collected, stored, aggregated and how that raw data is used to generate all the different information about COVID. 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.

[return to top]

List of Student Downloads

Students should download the following items from the Student Companion Center to complete the activities and assignments related to this chapter:

- Data Files (to support chapter Problems, in MS Access, Oracle, MS SQL, and MySQL):
  - Ch01_Text
  - Ch01_Problems

[return to top]

Chapter Objectives

The following objectives are addressed in this chapter:

1-1: Define the difference between data and information

1-2: Describe what a database is, the various types of databases, and why they are valuable assets for decision making

1-3: Explain the importance of database design

1-4: Outline how modern databases evolved from file systems

1-5: Identify flaws in file system data management

1-6: Outline the main components of the database system

1-7: Describe the main functions of a database management system (DBMS)

[return to top]

WHAT'S NEW IN THIS CHAPTER

Read about updates and improvements in this chapter:

- Streamlined coverage of historical evolution from file processing to database solutions for business needs. Clarified and streamlined coverage of APIs as a function of the DBMS.
- Updated multiple figures.

[return to top]

CHAPTER OUTLINE

I. WHY DATABASES?

II. DATA VERSUS INFORMATION

- **Data** consists of raw facts
  - The facts have not yet processed to reveal meaning to the end user
- **Information** is the result of processing raw data to reveal the meaning of data
  - To reveal meaning, information requires context
- Data is the foundation of information, which is the bedrock of **knowledge**
  - Knowledge implies familiarity, awareness, and understanding of information
- Accurate, relevant, and timely information is the key to good decision making
- **Data management** is a discipline that focuses on the proper generation, storage, and retrieval of data

III. INTRODUCING THE DATABASE

- A **database** is a shared, integrated computer structure that stores a collection of the following:
  - End-user data – that is, raw facts of interest to the end user
  - **Metadata**, or data about data, through which the end-user data is integrated and managed
    - Metadata describes the data characteristics and the set of relationships that links the data found within the database
- 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

A. Role and Advantages of the DBMS

- The DBMS presents the end user with a single, integrated view of the data in the database
- A DBMS provides the following advantages:
  - Improved data sharing
  - Improved data security
  - Better data integration
  - Minimized data inconsistency
  - Improved data access
  - Improved decision making
- Increased end-user productivity

B. Types of Databases

- A **single-user database** supports one user at a time
  - A **desktop database** is single-user database on a personal computer
- A **multiuser database** supports multiple users at the same time
  - A **workgroup database** supports a small number of users or a specific department
  - An **enterprise database** supports many users across many departments
- Classification by location
  - A **centralized database** supports data located at a single site
  - A **distributed database** supports data distributed across different sites
  - A **cloud database** is created and maintained using cloud data services
- Classification by data type
  - **General-purpose databases** contain a wide variety of data used in multiple disciplines
  - **Discipline-specific databases** contain data focused on specific subject areas
  - An **operational database** is designed to support a company's day-to-day operations
  - An **analytical database** stores historical data and business metrics used exclusively for tactical or strategic decision making and is comprised of two main components:
  - The **data warehouse** stores data in a format optimized for decision support
  - **Online analytical processing (OLAP)** is a set of tools for retrieving, processing, and modeling data from the data warehouse
- **Business intelligence** describes a comprehensive approach to capture and process business data to generate information that support decision making

- Databases can be classified to reflect the degree to which the data is structured
  - **Unstructured data** exists in its original (raw) state
  - **Structured data** is the result of formatting unstructured data to facilitate storage and use
  - **Semistructured** data has already been processed to some extent
- **Extensible Markup Language (XML)** is a language used to represent data elements in textual format
  - An **XML database** supports the storage and management of unstructured XML data
- **Social media** refers to web and mobile technologies that enable "anywhere, anytime, always on" human interaction
  - Data is captured about end users and consumers that requires the use of specialized database systems
- The term **NoSQL** (Not only SQL) is a new generation of DBMS that is not based on the traditional relational database model
  - These databases are designed to handle an unprecedented volume of data, variety of data types and structures, and velocity of data operations of new business requirements

IV. Why Database Design Is Important

- **Database design** refers to the activities that focus on the design of the database structure that will be used to store and manage end-user data
- Designing appropriate data repositories of integrated information using the two-dimensional table structures found in most databases is a process of decomposition
  - The integrated data must be decomposed properly into its constituent parts
- A well-designed database facilitates data management and generates accurate and valuable information
- A poorly designed database causes difficult-to-trace errors that may lead to poor decision making

V. Evolution of File System Data Processing

A. Manual File Systems

Manual file systems are accomplished through a system of file folders and filing cabinets

B. Computerized File Systems

With computerized file systems, a **data processing (DP) specialist** created a computer-based system to track data and produce required reports

C. File System Redux: Modern End-User Productivity Tools

- Business users widely use spreadsheet programs such as Microsoft Excel to enter data in a series of rows and columns to manipulate data
- A common misuse of spreadsheets is as a substitute for a database

VI. Problems with File System Data Processing

- The following problems with file systems challenge the types of information that can be created from data as well as information accuracy:
- Lengthy development times
- Difficulty of getting quick answers
- Complex system administration
- Lack of security and limited data sharing
- Extensive programming

A. Structural and Data Dependence

- A file system exhibits **structural dependence**, which means that access to a file is dependent on its own structure
  - All file system programs are modified to conform to a new file structure
- **Structural independence** exists when you change the file structure without affecting the application's ability to access the data
- **Data dependence** exists when all data access programs are subject to change when data storage characteristics change
- **Data independence** exists when data storage characteristics are changed without affecting the program's ability to access the data
- The practical significance of data dependence is the difference between the **logical data format** (how humans view the data) and the **physical data format** (how the computer must work with the data)
- Any program that accesses a file system's file must tell the computer not only what to do but also how to do it
- Data dependence makes the file system cumbersome from the point of view of a programmer and database manager

B. Data Redundancy

- **Data redundancy** exists when the same data is stored unnecessarily at different places
  - Database professionals use the term **islands of information** for scattered data locations
  - Increases the probability of having different versions of the same data

- Possible results of uncontrolled data redundancy
  - Poor data security
  - Data inconsistency
  - Data-entry errors
- Data integrity problems

C. Data Anomalies

- A **data anomaly** develops when not all of the required changes in the redundant data are made successfully
- The following are three types of anomalies:
  - Update anomalies
  - Insertion anomalies
  - Deletion anomalies

VII. Database Systems

- The **database system** consists of logically related data stored in a single logical data repository
  - However, the data might be physically distributed among multiple storage facilities
- The database's DBMS eliminates most of the file system's data inconsistency, data anomaly, data dependence, and structural dependence problems
- The current generation of DBMS software stores data structures, relationships between those structures, and access paths to those structures
  - The current generation also defines, stores, and manages all access paths and components

A. The Database System Environment

- The term **database system** refers to an organization of components that define and regulate the collection, storage, management, and use of data within a database environment
- The database system is composed of the following five components:
  - **Hardware**
  - **Software**
  - **People**
  - **Procedures**
  - **Data**
- Database solutions must be cost-effective as well as tactically and strategically effective

B. DBMS Functions

- A DBMS performs the following important functions:
  - Data dictionary management – The DBMS stores definitions of data elements and their relationships in a **data dictionary**
  - Data storage management – The DBMS creates and manages the structures required for data storage
    - **Performance tuning** ensures efficient performance
  - Data transformation and presentation – The DBMS transforms entered data to conform to required data structures
    - Data is formatted to conform to the user's logical expectations
  - Security management – The DBMS creates a system that enforces user security and data privacy
  - Multiuser access control – The DBMS uses sophisticated algorithms to ensure that multiple users can access the database concurrently without compromising its integrity
  - Backup and recovery management – The DBMS provides backup and data recovery to ensure data safety and integrity
    - Recovery management deals with the recovery of the database after a failure
  - Data integrity management – The DBMS promotes and enforces integrity rules, thus minimizing redundancy and maximizing data consistency
  - Database access languages and application programming interfaces – The DBMS provides data access through a query language
    - A **query language** lets the user specify what must be done without having to specify how
    - **Structured Query Language (SQL)** is the de facto query language and data access standard supported by the majority of DBMS vendors
  - Database communication interfaces – A DBMS accepts end-user requests via multiple communication interfaces

C. Managing the Database System: A Shift in Focus

- Disadvantages of database systems include the following:
  - Increased costs
  - Management complexity
  - Maintaining currency
  - Vendor dependence
  - Frequent upgrade/replacement cycles

VIII. Preparing for Your Database Professional Career

[return to top]

Complete List of Chapter Activities and Assessments

These tables lists all of the activities and assessments available in this chapter that will help you enhance your students' learning experience as they move through the course.

Print Only

Note: All of the activities listed are also available in the MindTap

Activity/Assessment	Module Objective	Duration	Source
Review Questions 1-17	All of the LOs in the chapter.	20 min	Book
Problems 1-11	All of the LOs in the chapter.	10 min	Book and MindTap

MindTap Only

Activity/Assessment	Module Objective	Duration
Watch: Chapter 01 Video – SQL	All of the LOs in the chapter.	03:05
Database for Life: Chapter 01: Database Concepts	All of the LOs in the chapter.	30 min
PowerPoints	All of the LOs in the chapter.	Times will vary
Flashcards	All of the LOs in the chapter.	Times will vary
Quiz: Chapter 01	All of the LOs in the chapter.	20 min
Chapter 01 Use It	All of the LOs in the chapter.	20 min
Chapter 01 Reflection	All of the LOs in the chapter	30 min

Instructor Resource/Supplements

Activity/Assessment	Duration	Source: Book, MindTap, or Instructor Resources
Test Bank 75 Multiple Choice, True/False, and Essay Questions	Times will vary	Instructor Resource Site, MindTap
Knowledge Check Activity 1-1	Times will vary	PPT Deck: Instructor Resources, MindTap
Knowledge Check Activity 1-2	Times will vary	PPT Deck: Instructor Resources, MindTap
Knowledge Check Activity 1-3	Times will vary	PPT Deck: Instructor Resources, MindTap

[return to top]

KEY TERMS

ad hoc query: A "spur-of-the-moment" question.

analytical database: A database focused primarily on storing historical data and business metrics used for tactical or strategic decision making.

application programming interface (API): Software through which applications interact with each other transmitting data, messages, status, etc. Also specifically software through which programmers interact with middleware. An API allows the use of generic SQL code, thereby allowing client processes to be database server-independent.

business intelligence: A comprehensive, cohesive, and integrated set of tools and processes used to capture, collect, integrate, store, and nalyze data with the purpose of generating and presenting information to support business decision making.

centralized database: A database located at a single site.

cloud database: A database that is created and maintained using cloud services, such as Microsoft Azure or Amazon AWS.

data: Raw facts, or facts that have not yet been processed to reveal their meaning to the end user.

data anomaly: A data abnormality in which inconsistent changes have been made to a database. For example, an employee moves, but the address change is not corrected in all files in the database.

data dependence: A data condition in which data representation and manipulation are dependent on the physical data storage characteristics.

data dictionary: DBMS component that stores metadata—data about data. Thus, the data dictionary contains the data definition as well as their characteristics and relationships. A data dictionary may also include data that are external to the DBMS. Also known as an *information resource dictionary*. See also *active data dictionary*, *metadata*, and *passive data dictionary*.

data inconsistency: A condition in which different versions of the same data yield different (inconsistent) results.

data independence: A condition in which data access is unaffected by changes in the physical data storage characteristics.

data integrity: In a relational database, a condition in which the data in the database complies with all entity and referential integrity constraints.

data management: A process that focuses on data collection, storage, and retrieval. Common data management functions include addition, deletion, modification, and listing.

data processing (DP) specialist: The person responsible for developing and managing a computerized file processing system.

data quality: A comprehensive approach to ensuring the accuracy, validity, and timeliness of data.

data redundancy: Exists when the same data is stored unnecessarily at different places.

data type: Defines the kind of values that can be used or stored. Also, used in programming languages and database systems to determine the operations that can be applied to such data.

data warehouse: An integrated, subject-oriented, time-variant, nonvolatile collection of data that provides support for decision making.

database: A shared, integrated computer structure that houses a collection of related data. A database contains two types of data: end-user data (raw facts) and metadata.

database design: The process that yields the description of the database structure and determines the database components. The second phase of the database life cycle.

database management system (DBMS): The collection of programs that manages the database structure and controls access to the data stored in the database.

database system: An organization of components that defines and regulates the collection, storage, management, and use of data in a database environment.

desktop database: A single-user database that runs on a personal computer.

discipline-specific database: A database that contains data focused on a specific subject area.

distributed database: A logically related database that is stored in two or more physically independent sites.

enterprise database: The overall company data representation, which provides support for present and expected future needs.

Extensible Markup Language (XML): A meta-language used to represent and manipulate data elements. Unlike other markup languages, XML permits the manipulation of a document's data elements. XML facilitates the exchange of structured documents such as orders and invoices over the Internet.

field: A character or group of characters (alphabetic or numeric) that has a specific meaning. A field is used to define and store data.

file: A collection of related records. For example, a file might contain data about the students currently enrolled at Gigantic University.

general-purpose database: A database that contains a wide variety of data used in multiple disciplines.

information: The result of processing raw data to reveal its meaning. Information consists of transformed data and facilitates decision making.

islands of information: In the old file system environment, pools of independent, often duplicated, and inconsistent data created and managed by different departments.

knowledge: The body of information and facts about a specific subject. Knowledge implies familiarity, awareness, and understanding of information as it applies to an environment. A key characteristic is that new knowledge can be derived from old knowledge.

logical data format: The way a person views data within the context of a problem domain.

metadata: Data about data; that is, data about data characteristics and relationships. See also *data dictionary*.

multiuser database: A database that supports multiple concurrent users.

NoSQL: A new generation of database management systems that is not based on the traditional relational database model.

online analytical processing (OLAP): A set of decision support system (DSS) tools that use multidimensional data analysis techniques. OLAP creates an advanced data analysis environment that supports decision making, business modeling, and operations research.

online transaction processing (OLTP) database: database *See operational database.*

operational database: database designed primarily to support a company's day-today operations. Also known as a *transactional database, OLTP database*, or *production database*.

performance tuning: Activities that make a database perform more efficiently in terms of storage and access speed.

physical data format: The way a computer "sees" (stores) data.

production database: *See operational database.*

query: A question or task asked by an end user of a database in the form of SQL code. A specific request for data manipulation issued by the end user or the application to the DBMS.

query language: A nonprocedural language that is used by a DBMS to manipulate its data. An example of a query language is SQL.

query result set: The collection of data rows returned by a query.

record: A logically connected set of one or more fields that describes a person, place, or thing.

semistructured data: Data that has already been processed to some extent.

single-user database: A database that supports only one user at a time.

social media: Web and mobile technologies that enable "anywhere, anytime, always on" human interactions.

structural dependence: A data characteristic in which a change in the database schema affects data access, thus requiring changes in all access programs.

structural independence: A data characteristic in which changes in the database schema do not affect data access.

structured data: Data that has been formatted to facilitate storage, use, and information generation in a predefined data model.

Structured Query Language (SQL): A powerful and flexible relational database language composed of commands that enable users to create database and table structures, perform various types of data manipulation and data administration, and query the database to extract useful information.

transactional database: *See operational database.*

unstructured data: Data that exists in its original, raw state; that is, in the format in which it was collected. Usually, unstructured data does not conform to a predefined data model.

workgroup database: A multiuser database that usually supports fewer than 50 users or is used for a specific department in an organization.

XML database: A database system that stores and manages semistructured XML data.

[return to top]

APPENDIX

GENERIC RUBRICS

Providing students with rubrics helps them understand expectations and components of assignments. Rubrics help students become more aware of their learning process and progress, and they improve students' work through timely and detailed feedback.

Customize these rubric templates as you wish. The writing rubric indicates 40 points and the discussion rubric indicates 30 points.

STANDARD WRITING RUBRIC

Criteria	Meets Requirements	Needs Improvement	Incomplete
Content	The assignment clearly and comprehensively addresses all questions in the assignment. 15 points	The assignment partially addresses some or all questions in the assignment. 8 points	The assignment does not address the questions in the assignment. 0 points
Organization and Clarity	The assignment presents ideas in a clear manner and with strong organizational structure. The assignment includes an appropriate introduction, content, and conclusion. Coverage of facts, arguments, and conclusions are logically related and consistent. 10 points	The assignment presents ideas in a mostly clear manner and with a mostly strong organizational structure. The assignment includes an appropriate introduction, content, and conclusion. Coverage of facts, arguments, and conclusions are mostly logically related and consistent. 7 points	The assignment does not present ideas in a clear manner and with strong organizational structure. The assignment includes an introduction, content, and conclusion, but coverage of facts, arguments, and conclusions are not logically related and consistent. 0 points
Research	The assignment is based upon appropriate and adequate academic literature, including peer reviewed journals and other scholarly work. 5 points	The assignment is based upon adequate academic literature but does not include peer reviewed journals and other scholarly work. 3 points	The assignment is not based upon appropriate and adequate academic literature and does not include peer reviewed journals and other scholarly work. 0 points

Research	The assignment follows the required citation guidelines. 5 points	The assignment follows some of the required citation guidelines. 3 points	The assignment does not follow the required citation guidelines. 0 points
Grammar and Spelling	The assignment has two or fewer grammatical and spelling errors. 5 points	The assignment has three to five grammatical and spelling errors. 3 points	The assignment is incomplete or unintelligible. 0 points

STANDARD DISCUSSION RUBRIC

Criteria	Meets Requirements	Needs Improvement	Incomplete
Participation	Submits or participates in discussion by the posted deadlines. Follows all assignment. instructions for initial post and responses. 5 points	Does not participate or submit discussion by the posted deadlines. Does not follow instructions for initial post and responses. 3 points	Does not participate in discussion. 0 points
Contribution Quality	Comments stay on task. Comments add value to discussion topic. Comments motivate other students to respond. 20 points	Comments may not stay on task. Comments may not add value to discussion topic. Comments may not motivate other students to respond. 10 points	Does not participate in discussion. 0 points
Etiquette	Maintains appropriate language. Offers criticism in a constructive manner. Provides both positive and negative feedback. 5 points	Does not always maintain appropriate language. Offers criticism in an offensive manner. Provides only negative feedback. 3 points	Does not participate in discussion. 0 points

[return to top]