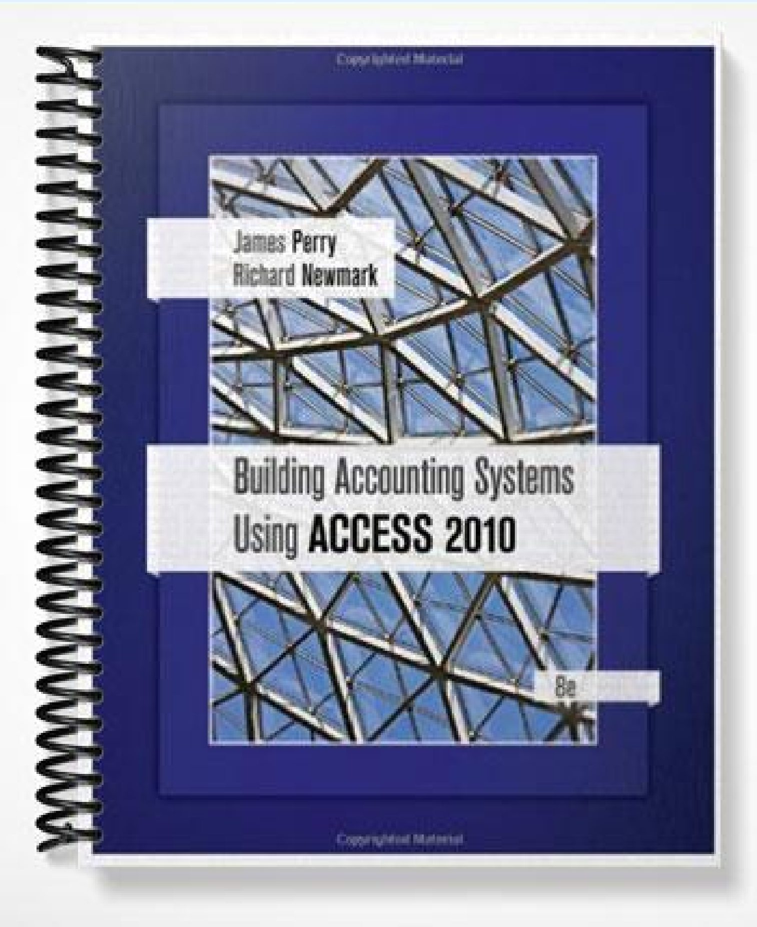
SOLUTIONS MANUAL



Chapter 2

Databases and Accounting Systems

Notes to the Instructor

In this chapter, students learn about database accounting systems and how they differ from double-entry bookkeeping systems. It explains the connections between accounting systems and database systems, why a relational database system is superior to double-entry bookkeeping systems for capturing detailed accounting information. Upon finishing this chapter, each student will understand how to create efficient, optimal database objects to capture accounting information Completed instructor files are available for use in a protected area of the companion Web site.

Before beginning, download the Chapter 2 database, **Ch02.accdb**, from this book's companion Web site. (We assume you can do this part on your own computer.) Your instructor's version of the Chapter 2 database contains answers, where appropriate, for the end of chapter questions. You may *not* want to distribute to students.

Lecture Outline

This chapter is independent of other chapters in the book in that it provides a theoretical foundation that links accounting systems and database systems.

Objectives

- Chapter 2 presents the theoretical foundation for the remainder of the text and contains both practice and theory for both accounting transaction cycles and database management systems
- Important topics covered include:
  - Differences between double-entry bookkeeping and database accounting systems.
  - Advantages and disadvantages of database accounting systems.
  - Business processes.
  - The relationship between accounting systems and database systems.
  - A brief history leading to the development of database management systems.
  - Functions of database managements systems.
  - Theory and application of relational database management systems.
  - The structure of database objects that store accounting events.
  - The importance of normalizing tables.
  - Performing database selections, projections, and joins.
  - The resources-events-agents (REA) model for accounting databases.
- The Coffee Merchant, a fictitious coffee bean and tea wholesaler, demonstrates the classic accounting application involving processing and maintaining invoice data
  - Keeping track of unpaid customer invoices
  - Generating relevant reports based on current and historical customer data

Introduction

- Students usually learn accounting using manual systems with journals and ledgers.
- This chapter explains the differences between database accounting systems and double-entry bookkeeping.
- Database accounting provides advantages, yet also has certain disadvantages.

- The business activity classifications in the chapter provide a three-level taxonomy of complexity that can help system designers decide when to incorporate certain features in their designs.
- Transaction cycles provide accountants and auditors a way to organize economic events into related categories.

Database Accounting Systems

- Chapter 2 introduces database theory and explains how firms can use relational databases as part of their accounting systems.
- Businesses see the advantages of relational databases for all their information processing needs.

Events-Based Theories of Accounting

- Researchers such as William McCarthy and George Sorter developed events approaches to accounting that lend themselves to the use of relational databases to perform accounting tasks.
- Events approaches to accounting argue that accountants ideally should store all relevant attributes of economic events in a readily accessible form.
- This is not a fully attainable objective with currently available technology; however, relational databases do a fairly good job of storing many information attributes and giving users a variety of ways to retrieve those attributes.

Double-Entry Bookkeeping Versus Database Accounting

- Double-entry bookkeeping satisfied accountants' need to capture transaction essentials for many years.
- Five hundred years ago, the costs of gathering and storing information were very high.
- Double-entry bookkeeping let businesspersons capture key attributes of transactions in a highly aggregated form, which minimized the cost of information gathering and storage.
- Debit-credit balancing checks provided important internal control in manual accounting systems.
- A Try it exercise asks students to make a trip to an office supply store in search of two-column accounting paper and predicts that the sales clerk will look up the product's location using the store's inventory database, which will most likely be a relational database.
- Computers now easily capture a wide variety of information about transactions with bar code scanners and other devices.
- A sales transaction provides the basis for comparing double-entry bookkeeping and database accounting. The double-entry bookkeeping journal entry stores five attributes of the transaction.
- Figure 2.1 shows how the database accounting system can efficiently store many more attributes of the sales transaction than can be stored in a traditional double-entry bookkeeping system.
- The chapter refers to the normalization rules discussed in Chapter 2 and presents a Try it exercise that illustrates the problems presented by shipping firm names stored in a non-normalized table. This is a review point and is usually worth some class discussion time.
- We note that the database system directly records only some of the items that the double-entry bookkeeping system records and describe how the database system would use queries

to calculate some information attributes that the double-entry bookkeeping system stores in an aggregate form.

- A Try it exercise challenges students to identify multiple ways that a user might enter the names of common shipping companies (for example, Federal Express might be entered as *FedEx*, *FEDEX*, *Fed Express*, *FEx*, or *Federal Express*).
- A Try it exercise asks students to identify interesting facts that can be found in or calculated from the database diagrammed in Figure 2.1.
- The database approach can do everything that double-entry bookkeeping can do and more.

Advantages of Database Accounting Systems

- Reduce data storage costs.
- Eliminate data redundancy.
- Eliminate data inconsistencies.
- Avoid duplicate processing.
- Ease add, delete, and update data maintenance tasks.
- Make data independent of applications.
- Centralize data management.
- Centralize data security.
- Database accounting systems offer much greater flexibility in extracting data than flat file double-entry accounting systems. This flexibility leads to other advantages such as:
  - Ease report modifications and updates.
  - Provide ad hoc query capabilities.
  - Facilitate cross-functional data analysis.
  - Permit multiple users simultaneous data access.
- Provide data entry and integrity controls as part of the database management system.

Disadvantages of Database Accounting Systems

- Greater hardware requirements can be more expensive.
- The cost of the database software itself.
- Cost of employing a database administrator (DBA).
- Centralizing management and security control functions creates several drawbacks:
  - System operation becomes critical.
  - Incorrect data entered corrupts many users' work.
  - Territorial disputes over data ownership may arise.
- Accountants tend to distrust any single-entry accounting system, such as a database accounting system, because double-entry bookkeeping is so pervasive in accounting education and practice. Most accountants' first response is to question and fear anything else.

Business Processes

- A value chain is an accounting system viewed as a collection of processes rather than financial statements.
- Transaction cycles are common transactions associated with business processes.
- Business processes include Financing, HR, Conversion, Sales/Collection, and Acquisition.

Sales/Collection Process

- Includes all sales and cash collection activities.
- Transactions recorded include customer orders, sales, and cash receipts.

Acquisition/Payment Process

- In a manufacturing firm, the acquisition and payment processes includes all activities related to ordering materials, receiving materials, and paying for them.
- Service firms record purchases of materials that aid service including office supplies.

Human Resources Process

- HR processes revolve around employees and include calculating employee pay, deductions, and net pay.
- HR reports include checks to employees and vendors, commission reports, timecard reports, and payroll registers.

Financing Process

- Cash receipts and payments related to equity and debt financing fall are part of the financing process.
- Recorded financing transactions involve issuance of stock, receipt of cash for stock, declaration of dividends, and payment of dividends.

Accounting Information Systems and Database Systems

- Historically, accounting information has been captured in journals and posted to ledgers.
  - Strict rules were applied to the process of gathering data and presenting information.
  - Standard reports (e.g., Income Statement or Balance Sheet) provided a specific set of information with a format and level of aggregation that was determined before the reports were generated.
  - Off-the-shelf accounting software provided these standard reports.
- Modern accounting systems are usually built on relational databases and allow management to generate customized reports that present relevant information for decision making.
  - More information about an event (e.g., a sale can include dollar amounts, customer information, inventory information, time of the sale, and so on) can be captured and easily accessed.
  - Advances in inexpensive and easily learned relational database management software and computer hardware support this approach.
- Events account records details about economic events in a database including who was involved, when the event occurred, and what resources were affected.

Database Management Systems

- Database management systems (DBMSs) are valuable to business enterprises because they provide the software to store, retrieve and modify crucial business data.

Pre-DBMS Data Acquisition and Reporting

- Corporate information (e.g., accounts receivable) was kept by department in flat files on computer disks.
- Standard reports (e.g., inventory stock levels) were readily available.

- Custom or unusual reports had to be specially designed and written, often a time consuming and expensive task.
- Duplicate information, data files mirroring master file information, was often kept by different departments and led to data redundancy and inconsistency (e.g., customer address information was kept by marketing and accounts receivable).

Functions of a Database Management System

The capabilities that a database management system provides in development of an information system are:

- Efficient data storage, update, and retrieval.
- User accessible catalog.
- Concurrency control.
- Transaction control.
- Recovery services.
- Security and authorization services.
- Integrity facilities.

Advantages of Database Management Systems

- Subschema: Provide each user with particular or unique view (authorized access) of the database.
- Data independence: system's ability to hide the details of the physical storage of information from the application programs, or the user's views of access techniques and methods.
- Structure changes are transparent.
- Data sharing: centrally stored information is current and consistent and everyone can have access.

Disadvantages of Database Management Systems

- Secondary (disk) storage—DBMS's require more storage space than flat files.
- Additional people—database experts may be needed to operate system.

Relational Database Management Systems

- Implementation of RDBMS requires a data model—an abstract representation of a database system providing a description of the data and methods for accessing the data managed by the database.
- Three models have been used during the history of databases:
  - ➢ Hierarchical—(1960s through early 1970s).
  - ➢ Network—(1960s through early 1970s).
  - ➢ Relational—(1970s) overwhelming choice today and has many advantages:
    - o Logical and physical characteristics are distinct and provide the user a more intuitive view of data.
    - o Requires little training.
    - o More powerful retrieval and update operators available.
    - o Powerful tools to warn when a database has design flaws.
- Relational or RDBMS model is easily the most popular of the three models, and it will be discussed from this point on

Database Objects

- RDBMS model defines the conceptual view that the user has of all the objects contained by the database system.
  - The RDBMS model represents both data and relationships between them.
  - All data, including the database table definitions and object information, exists in tables.
- Relational Database is a collection of relations.
  - A table is a relation that consists of rows (tuples) and columns (attributes).
  - Files, records, and fields are the same as relations, tuples, and attributes (see Figure 2.3).
- Properties of relations:
  - The entries in each column of any row are single valued.
  - Each attribute of a given relation has a distinct name, called the attribute name.
  - Every value in a column contains values for that column only, and the values are of like data type.
  - The order of the rows is unimportant.
  - The order (position) of the columns in relation to each other is unimportant.
  - Each row is unique from all other rows in the relation.
- Customer table—Coffee merchant (Figure 2.4)
  - Rows represent customers—each row must be unique
  - Columns represent attributes of each customer, schema for table includes: CustomerID (primary key), CompanyName, PhoneNumber, and Contact
  - Attribute values are within columns

Primary and Foreign Key Attributes

- Primary key is a column (or group of columns) that uniquely identifies a given row—the CustID column, for example (Figure 2.5).
- Data Dictionary is a collection of tables containing the definition, characteristics, structure, and description of all data maintained by the RDBMS.
- Foreign key is an attribute in one table that must match the primary key in another table—the CustomerID column of the tblInvoice table, for example, shown in Figure 2.5.
- tblInvoice and tblCustomer tables can be joined on the CustID/CustomerID fields.
- Each column of a table is an attribute.
- Columns, including the primary key column and any foreign key columns, can appear in any order.
- Microsoft Access does not require that each table have primary key, but each table *should* have one.

Schema of a Relation

- A schema is a representation of a table that lists all its attributes and identifies the primary key and, optionally, foreign keys
- Schema for the Customer table, tblCustomer, with a reduced number of attributes is:

  Customer(<u>CustID</u>, CustAddress, CustCity, CustState)

  where CustID is the primary key and other attributes are listed and separated by commas

Data Dictionary

- Every relational database system has a data dictionary.

- Data Dictionary is a collection of tables containing the definition, characteristics, structure, and description of all data maintained by the RDBMS.
- Information about tables and their attributes is self contained in the database.
- When a user changes a structure element of a table or other database object, the RDBMS automatically changes the object's definition in the data dictionary to keep it constantly current.
- The data dictionary contains the names of all tables, queries, forms, reports, and other objects as individual rows.
- Microsoft Access does not include a built-in facility for creating and managing a data dictionary. Many other RDBMSs that are used in large database installations (such as Oracle) do include a data dictionary. In Access, the user must create a data dictionary, usually in a separate Access or Microsoft Word file.

The Coffee Merchant Tables

- Figure 2.6 shows the schema of all the tables that make up the Coffee Merchant invoicing system.
- The students should study the primary-key-to-foreign-key relationships in Figures 2.7 through 2.9.

Normalization

- Normalization is the process of determining the correct location for each attribute to meet the relational database rules.
- Unnormalized databases lead to redundant, inconsistent, and anomalous information being stored in tables.
- Properly normalized tables necessary to create an invoice—Coffee Merchant
  - ➢ **tblCustomer**—customer information; primary key CustID.
  - ➢ **tblInvoice**—unpaid invoices; primary key InvoiceID.
  - ➢ **tblInvoiceLine**—quantity information; InvoiceID plus InventoryID combine to form composite primary key.
  - ➢ **tblInventory**—catalogs all items available; primary key InventoryID.
- Note: Extended price, subtotal, etc., are calculated when the invoice is printed.
- Other secondary tables involved in the invoicing subsystem include these:
  - ➢ **tblEmployee**—contains sales employee information; primary key (pk) is EmpID.
  - ➢ **tblCountry**—contains country names for the World and coffee/tea export data; pk is CountryID.

First Normal Form

- First Normal Form (1NF) requires that repeating groups (unnormalized table Figure 2.10) be eliminated
- To attain 1NF status, a new table must be created with a composite primary key—for example, Invoice Line [InvoiceID, ItemID, Quantity].
- Least restrictive, this table can be placed into a RDBMS, but is not as good as higher order forms.
- Tables in 1NF (Figure 2.11) pose several problems:

  - New customer cannot be added without an invoice being generated—known as an insertion anomaly—invoice attributes (e.g., Invoice) cannot be null according to relational database rules.
  - Invoice cannot be deleted without deleting customer information—known as a deletion anomaly.
  - Customer table contains a great deal of redundant information (e.g., Company is repeated for each new invoice).
- Problems can be eliminated by altering the table structure and changing it into Second Normal Form.

Second Normal Form

- Second Normal Form (2NF) requires that each non-key attribute be fully dependent on the entire primary key, not just part of it (e.g., Figure 2.11—Total is functionally dependent on Invoice because a value of Invoice determines a single value of Total, not the primary key, CustID).
- Figure 2.16 shows functional dependencies in the Customer table.
- To attain 2NF status, break Customer table into two tables (Customer and Invoice) and add an additional attribute—a foreign key—linking both tables on their keys CustID and CustomerID, respectively.

Third Normal Form

- Third Normal Form (3NF) requires that a table be in 2NF with all transitive dependencies eliminated (that is, all attributes functionally dependent on the single-attribute primary key).
  - Transitive Dependency occurs when, for example, attribute B determines attribute C, and C determines D (Figure 2.14).
  - To attain 3NF status, create a new table containing at least the determinant attribute (For example, Customer [CustID, CustName]).
- 3NF enforces an informal rule stating that a table should store one fact and one fact only.

Table Relationships

- Process of normalizing creates additional tables with relationships maintained by foreign key to primary key links.
- Three fundamental relationships exist between related tables:
  - One-to-one (1–1)—e.g., a master table contains customer information—a related table contains occasional notes about a few customers.
  - One-to-many (1–M)—e.g., a master table contains customer information—a related table has unpaid invoices for each customer.
  - Many-to-many (M–M)—e.g., a master table contains invoice information—a related table contains inventory information. A relationship table is created to represent the M–M relationship in a convenient way that avoids anomalies. This table makes the connection between two M-M tables so that two 1-M relationships are created among the three tables.
- General Rules Governing Relationships Among Tables:
  - Primary keys must not be null.
  - Create a foreign key from the primary key on the one side of the 1–M relationship.

  - ➢ Many-to-many relationships are handled by creating an additional table—the relationship table—that consists entirely of the parent tables' primary keys. (The relationship table can contain other columns as well.)
  - ➢ Most one-to-one relationships indicate unnecessary tables in the database design. Normally, you should merge the two tables. Exceptions occur when there are just too many columns in one table, or one group of fields is used far more frequently than another group. Then, you can consider separating them into different tables for efficiency's sake.

Fundamental Relational Database Operations

Select

- Select operator chooses a set of rows from a table (Figure 2.15)—the query selects rows in which the hire date later than 1/1/2008 and delivers answers in table form.

Project

- Project operator returns a subset of columns from one or more tables (Figure 2.16)—the projection indicates which columns, EmpLName and EmpGender, are retrieved in the answer.

Join

- Join provides the ability to pull together data from disparate but associated tables into a single, virtual table based on a common attribute (Figure 2.17)—EmpDivisionID is foreign key in tblEmployee table and primary key in tblEmployeeDivision table. *Instructors note the following*: The tblEmployeeDivision table has been omitted from this edition of the textbook for simplification reasons.
- Equijoin—rows from the two tables are linked or chained together on matching join column values and the join column appears only once in the result
- Outer join—combines rows from two or more tables on the join column, but rows that do not match on the join column are included in the result

Introduction to Database Design

- An important aspect of database design is the care in choosing the tuples and attributes that each table comprises. This activity is called modeling

Developing Entity-Relationship Models

- Another way to model database objects, called Entity-Relationship (E-R) modeling, to represent a business is a graphical approach using three terms to describe a company's information.
  - ➢ Entities (nouns) are the objects found in the company (e.g., invoices, purchase orders, etc.)
  - ➢ Relationships are the way in which the distinct entities interact or are related to one another.
  - ➢ Attributes (adjectives) describe the entities and relationships.
- Entities and relationships are represented by diagrams (Figure 2.18).
  - ➢ Rectangles represent entities.
  - ➢ Diamonds represent relationships.

  - Lines represent the connections between the two.
  - Digit or letter above line indicates degree of the relationship: 1–1, 1–M, M–M.
- Completed E-R diagrams can be combined into a system E-R diagram—a process called view integration.
  - Process started by placing the most often used entity in the center of the diagram.
  - Lines connect the related entities.
  - Identify and create tables from entities.
  - Relationships between entities are maintained by foreign/primary key linkages between tables.
  - Third table, a relationship table, is created for M–M relationship—it contains the primary keys of both tables.
  - Last step is to normalize the individual tables.

Resources, Events, Agents (REA) Modeling

- In the late 1970s, Professor William McCarthy drew upon the principles of relational database theory developed by E. F. Codd and the entity-relationship modeling principles of Peter Chen to
  - create a modeling approach specifically designed for accounting systems.
  - develop an entity classification system that would allow accountants to use relational databases handle accounting information.
- REA modeling provides categories of entities that accountants can use to classify the entities that appear in accounting systems.
  - Resources are assets such as cash, inventory, and fixed assets.
  - Events are transactions or other occurrences that have accounting effects and include things such as purchases and sales.
  - Events can also be more subtle occurrences, such as the passage of time that causes interest to accrue on a loan or depreciation of a fixed asset to occur.
  - Agents are humans or organizations that interact with resources and events, such as customers, suppliers, and employees
- A number of researchers have proposed extensions to McCarthy's basic model
  - Eric Denna devised a REAL model, which added an entity classification for location
  - Most accounting researchers agree that the REA model has considerable value as an approach to the design of accounting information systems
  - Figure 2.19 shows a list of entities for a purchase cycle accounting system. The entities are classified using the REA model.

Answers to Review Exercises

Multiple Choice Questions

1.	b	6.	c
2.	d	7.	a
3.	b	8.	b
4.	a	9.	d
5.	c	10.	b

Discussion Questions

The solutions presented here come from the textbook discussion. Your students may include other insight that is relevant but not presented in these solutions.

1. Double-entry bookkeeping provided an excellent way of recording transactions for many years. It helped accountants capture the essence of each transaction in a safe, effective manner. The double-entry nature of the approach also provided an automatic internal control feature in manual accounting systems. Since recording transactions with pen and paper was a time-consuming task, double-entry bookkeeping gave accountants a valuable tool that quickly identified essential elements of transactions. Today’s computerized transaction processing systems have freed accountants from the need to capture only essential transaction elements and store them in a highly aggregated form. We can capture a wide variety of information about each transaction quite easily using bar code scanners and other electronic data entry devices. The costs of information storage have also dropped dramatically.

2. In manual systems, the dual nature of the accounting debit and credit model provides a built-in error correction mechanism. In automated systems, this same duality is inefficient and serves no control purpose. Manual double-entry bookkeeping systems can be very efficient; however, most computer implementations of double-entry bookkeeping use a flat file processing design instead of a relational database model. Database accounting systems only store data once and avoid the redundancies of flat-file systems. Database accounting systems use normalized tables to avoid anomalies. Anomalies can become very troublesome as the accounting system grows to a size that requires computers to handle transaction-processing volume.

3. Problems of storing data in two or more places include the creation of outdated and redundant data. Years ago, it was common for departments and individuals to create and maintain their own computer files that duplicated information kept in the master files. Duplicate files were maintained in order to access and examine the data with their own programs quickly, rather than incurring the long delays from the already overworked data processing department. Duplicate data files—files mirroring the master file data—leads to inevitable data redundancy and, soon, data inconsistency. For example, Marketing kept its own files, sorted by zip code, of its larger customers so that they could send them advertising pieces. The marketing department hired a bright young programmer to keep current the “best customer” file, and she would write programs to produce mailing labels from the customer file. Problems occurred when the independently maintained customer list fell out of date.

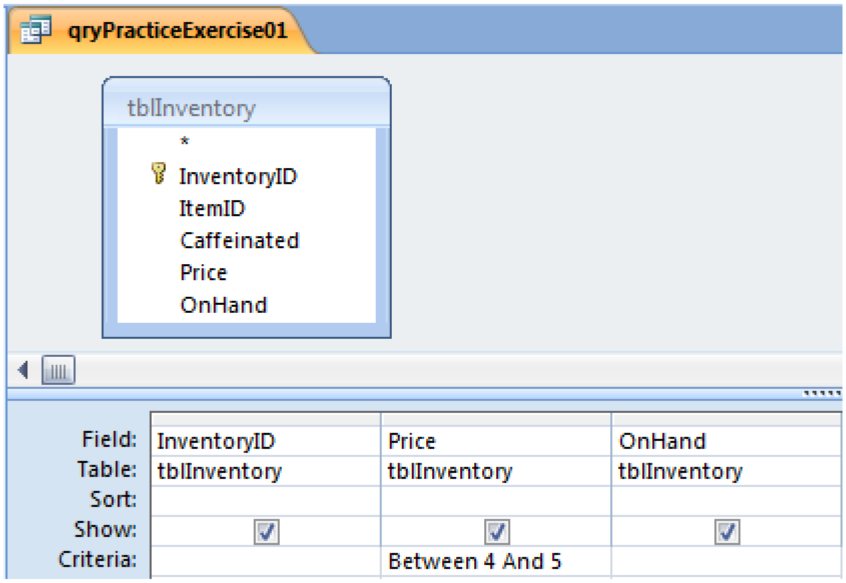
While the master list of customers was kept current by the data processing department from each month's purchase orders, the marketing department did not have access to that data. Their customer list became so out of date—customers moved, and new customers came on board—that it was practically useless.

4. The primary key consists of one or more fields in each table that provides a unique identifier for each row in the table. Every table in a relational database (a table in a relational database is also called a relation) must have a primary key that is unique and that exists for each row in the table. The primary key can include one or more fields (columns). When the primary key includes more than one column it is called a composite primary key. The individual column values in a composite primary key need not be unique, but the combined column values must be unique. The primary key is important because it allows all of the information that has been broken apart to be stored in the individual tables to be pulled back together when a user needs the information. The primary keys (and the foreign keys) provide the connections among all of the tables in a database.

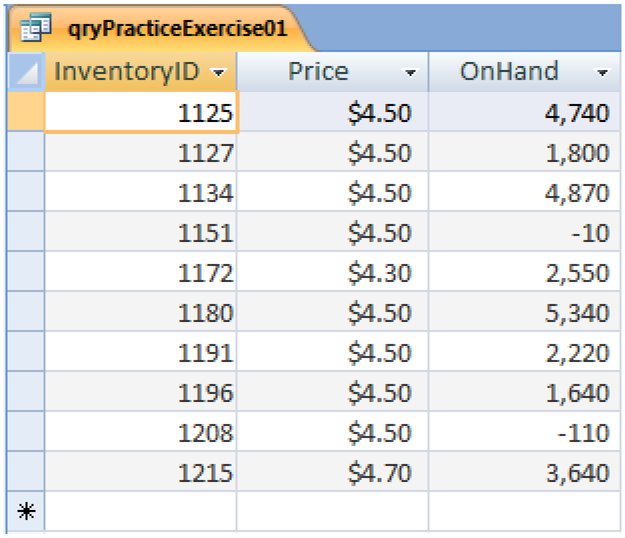
Practice Exercises

The solutions are included in the Ch02-PracticeExercises.accdb file on the Instructor web site.

1. One possible solution is shown below in Design view and Datasheet view.



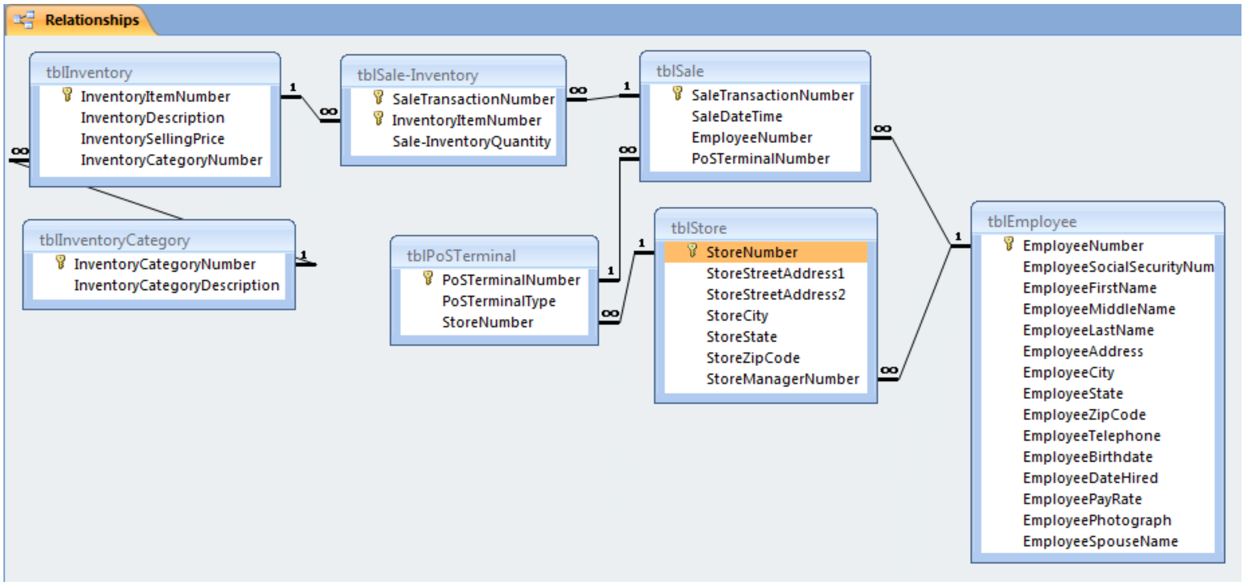
Field:	InventoryID	Price	OnHand
Table:	tblInventory	tblInventory	tblInventory
Sort:			
Show:	☑	☑	☑
Criteria:		Between 4 And 5	



qryPracticeExercise01

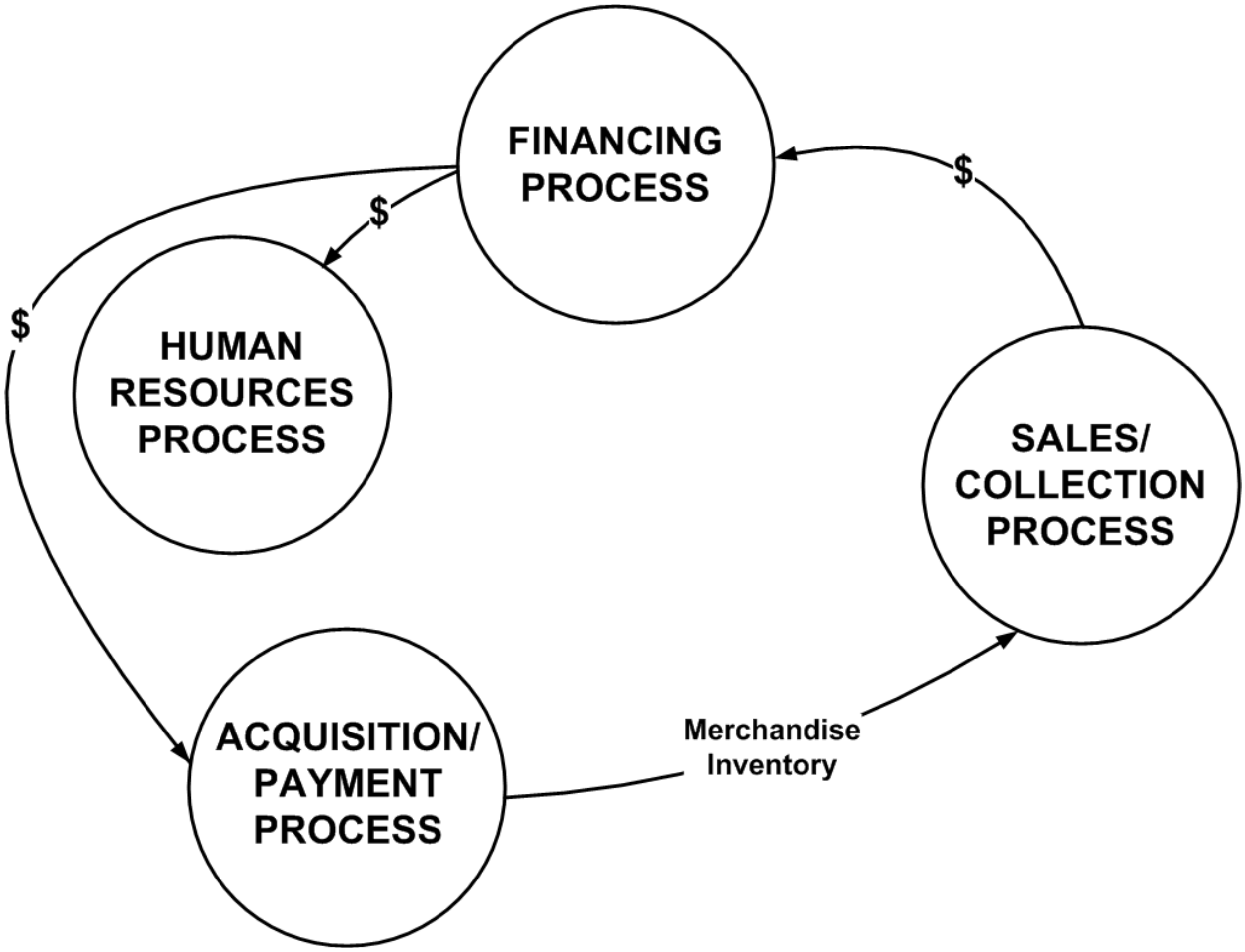
InventoryID	Price	OnHand
1125	$4.50	4,740
1127	$4.50	1,800
1134	$4.50	4,870
1151	$4.50	-10
1172	$4.30	2,550
1180	$4.50	5,340
1191	$4.50	2,220
1196	$4.50	1,640
1208	$4.50	-110
1215	$4.70	3,640
*		

2. One possible solution to this exercise is the database shown in the following figure, which shows a screen capture of an Access Relationships window. In creating this file, we assumed that the grocery store does not track its customers' names or addresses and that it only accepts cash. The tblPoSTerminal stores information about each point-of-sale (PoS) terminal in each store. This database design is a useful example because it illustrates that not all businesses include customer tables in their sales/collection processes.



3. The additional entities would be best modeled in two separate tables, an Office table and a Department table. The primary keys of these two tables would each appear as a foreign key in the Time Worked table.

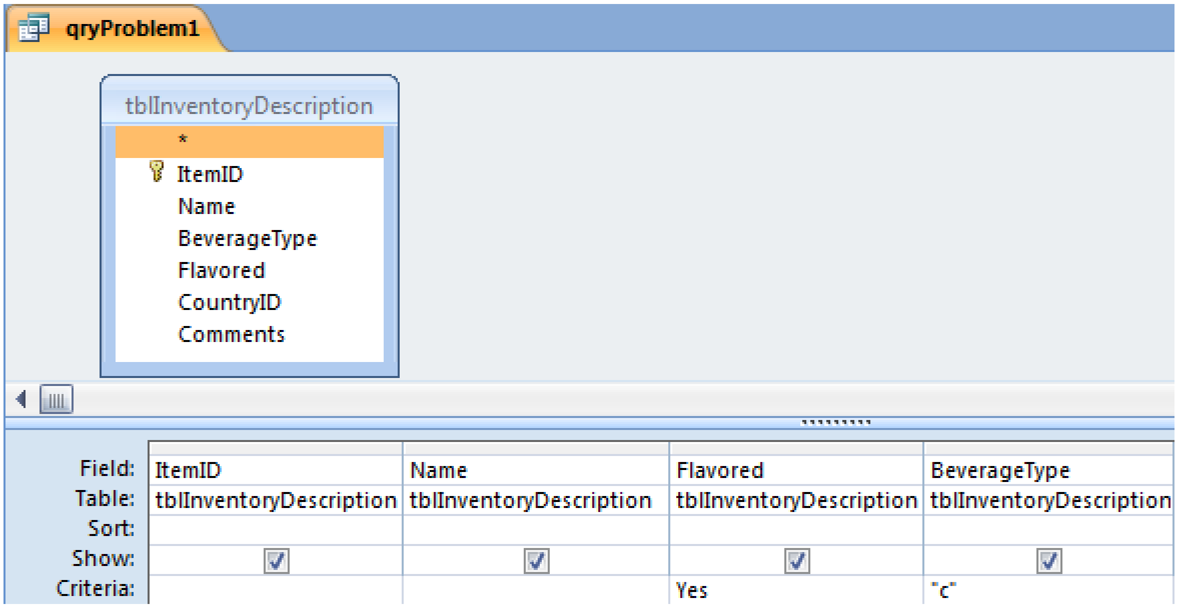
4. Omit the conversion process bubble and the related resource flow arrows from Figure 2.2. Also, add a resource flow line for merchandise inventory from the acquisition/payment process to the sales/collection process to yield the correct answer. See below.



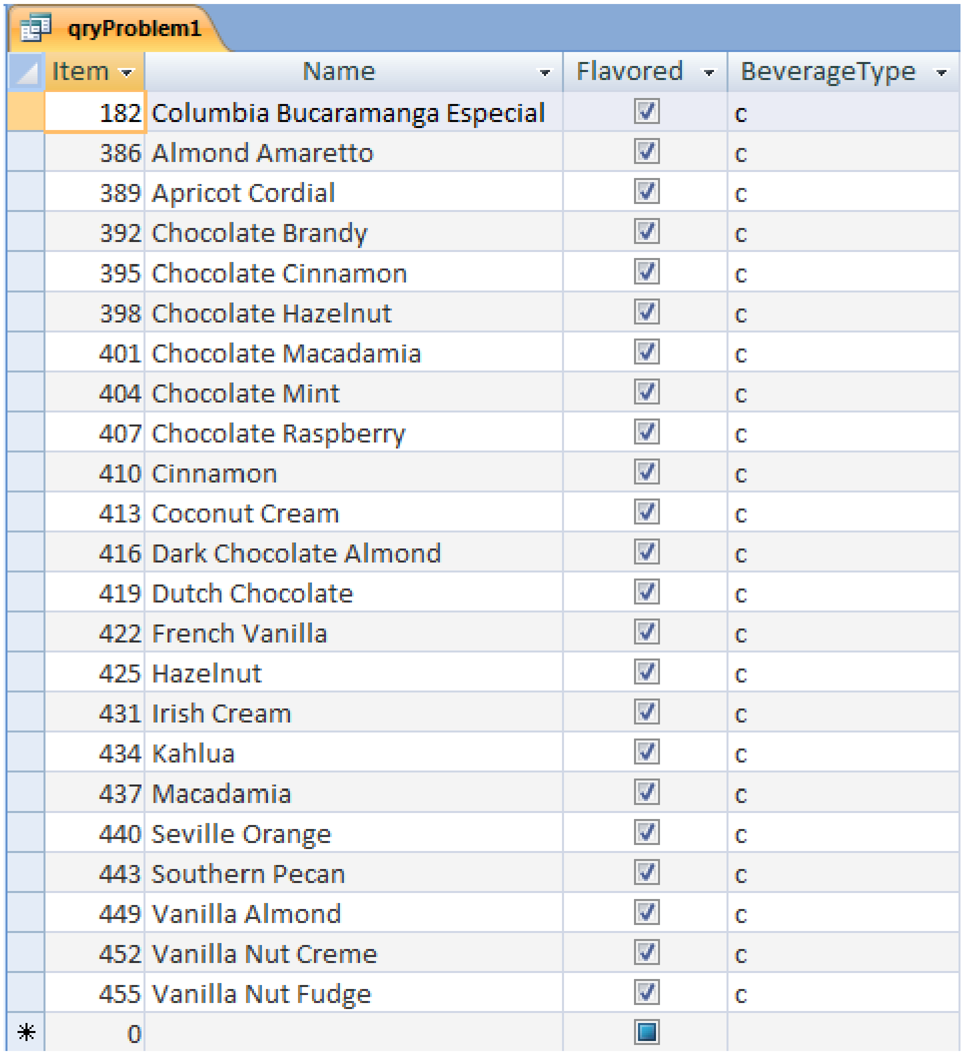
5. The problem is that age is a value that changes each year. The solution is to store a person's date of birth. That never changes. Follow the general rule "never store any value in a table that can be derived from other values in the table or the external environment." Age is such a value—it can be calculated from date of birth and the current date, which is stored in the computer's clock.

Problems

1. One possible solution is the object qryProblem01, which is included in the Ch02Problem01.mdb database on the Instructor's CD. The figures below show that query in Design view and in Datasheet view.



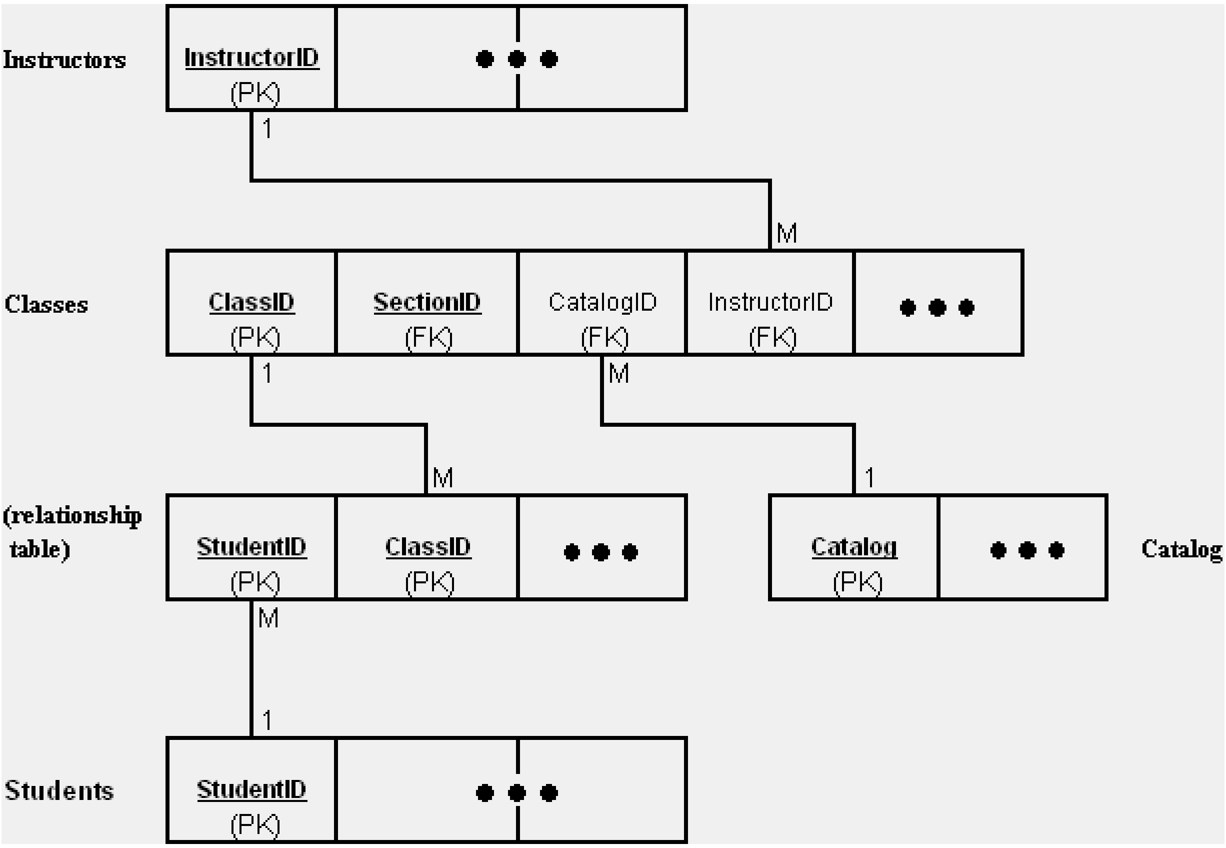
Field:	ItemID	Name	Flavored	BeverageType
Table:	tblInventoryDescription	tblInventoryDescription	tblInventoryDescription	tblInventoryDescription
Sort:				
Show:	☑	☑	☑	☑
Criteria:			Yes	"c"



	Item	Name	Flavored	BeverageType
	182	Columbia Bucaramanga Especial	☑	c
	386	Almond Amaretto	☑	c
	389	Apricot Cordial	☑	c
	392	Chocolate Brandy	☑	c
	395	Chocolate Cinnamon	☑	c
	398	Chocolate Hazelnut	☑	c
	401	Chocolate Macadamia	☑	c
	404	Chocolate Mint	☑	c
	407	Chocolate Raspberry	☑	c
	410	Cinnamon	☑	c
	413	Coconut Cream	☑	c
	416	Dark Chocolate Almond	☑	c
	419	Dutch Chocolate	☑	c
	422	French Vanilla	☑	c
	425	Hazelnut	☑	c
	431	Irish Cream	☑	c
	434	Kahlua	☑	c
	437	Macadamia	☑	c
	440	Seville Orange	☑	c
	443	Southern Pecan	☑	c
	449	Vanilla Almond	☑	c
	452	Vanilla Nut Creme	☑	c
	455	Vanilla Nut Fudge	☑	c
✳	0		■	

2. The Invoice Line table is a relationship table. All relationship tables have a composite primary key that includes two or more individual fields (the primary key fields of the entity tables that the relationship table joins). In this case, tblInvoiceLine joins the Invoice table and the Inventory table. Thus, the primary key of tblInvoiceLine includes the InvoiceID field and the InventoryID field. The extension is not included in the table because it is a calculated field. A printed invoice would be produced by creating a report based on a query. That query would calculate the extension.

3. The relationship between the Instructor table and a Classes table is one-to-many (1—M). Each instructor teaches many classes (unless he or she is a researcher and teaches only one class per semester—atypical of most instructors). The relationship between Students table and the Classes table is many-to-many (M—M). That is, a given student can take several classes. Any particular class can have several students enrolled in it. The following shows a typical relationship between the Catalog, Classes, Instructors, and Students. Notice a relationship table exists when necessary to convert M—M relationships to two 1—M relationships.


Instructors
InstructorID (PK)
• • •
1
M
Classes
ClassID (PK)
SectionID (FK)
CatalogID (FK)
InstructorID (FK)
• • •
1
M
M
1
(relationship table)
StudentID (PK)
ClassID (PK)
• • •
Catalog (PK)
• • •
Catalog
M
1
Students
StudentID (PK)
• • •