Printed by: rituadhikari20@yahoo.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

CHAPTER 13

Database Design Case Study #1: Mighty-Mite Motors

Abstract

This chapter presents the first of three major case studies in the book. The emphasis is on reengineering an existing database system, and includes a discussion of the ways in which an organization may need to change along with the introduction of more up-to-date technology.

Keywords

database reengineering
database case study
database design
data flows
ER diagrams
SQL CREATE

It is not unusual for a database designer to be employed to reengineer the information systems of an established corporation. As you will see from the company described in this chapter, information systems in older companies have often grown haphazardly, with almost no planning and integration. The result is a hodgepodge of data repositories that cannot provide the information needed for the corporation to function because they are isolated from one another. In such a situation, it is the job of the database designer to examine the environment as a whole and to focus on the integration of data access across the corporation, as well as the design of one or more databases that will meet individual department needs.

On the bright side, an organization such as Mighty-Mite Motors, which has a history of data processing of some kind, knows quite well what it needs in information systems, even if the employees are unable to articulate those needs immediately. There will almost certainly be a collection of paper forms and reports that the organization uses regularly to provide significant input to the systems design process.

Corporate Overview

Might-Mite Motors, Inc. (MMM) is a closely held corporation, established in 1980, that manufactures and markets miniature rideable motor vehicles for children. Products include several models of cars, trucks, all-terrain vehicles, and trains (see Figure 13.1). Vehicles are powered by car batteries and achieve speed of about 5 mph.

Printed by: rituadhikari20@yahoo.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Mighty-Mite Motors

Product Catalog

Winter Holiday Season 2020

Model #001 **001**

All Terrain Vehicle: Accelerator in the handgrip lets young riders reach speeds of up to 5 mph. Vehicle stops immediately when child removes his or her hand from the handlegrips. Can carry one passenger up to 65 lbs.
Suggested retail price: $124.95

Model #002 **002**

DO NOT COPY
rituadhikari20@yahoo.com

Printed by: rituadhikari20@yahoo.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Model #002

002

4-Wheel Drive Cruiser: Two-pedal drive system lets vehcile move forward at 2 1/2 mph on hard surfaces, plus reverse. Electronic speed reduction for beginners. Includes one 6v battery and one recharger. Ages 3–7 (can carry two passengers up to 40 lbs each). **Suggested retail price: $249.99**

Model #003

003

Classic roadster: Sounds include engine start-up, rev, shifting gears, and idle. Two forward speeds—2 1/2 mph and 5 mph; reverses at 2 1/2 mph. High-speed lockout. On/off power pedal. Power-Lock electric brake. Includes two 6v batteries and recharger. Ages 3–7 (carries two passengers up to 60 lbs each). **Suggested retail price: $189.95**

Model #004

004

Sports car #1: Two-forward speeds, 2 1/2 and 5 mph. Reverses at 2 1/2 mph. High-speed lockout. Power-Lock electric brake. Includes two 6v batteries and one recharger. Ages 3–6 (carries two passengers up to 90 lbs. total).
Suggested retail price: $249.95

Model #005

005

Sports car #2: Phone lets child pretend to talk while he or she drives. Two forward speeds—2 1/2 mph and 5 mph; reverses at 2 1/2 mph. High-

DO NOT COPY
rituadhikari20@yahoo.com

Printed by: rituadhikari20@yahoo.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

speed lockout. Power-Lock electric brake. Includes two 6v batteries and one recharger. Ages 3–6 (carries two passengers up to 90 lbs. total).
Suggested retail price: $249.95

Model #008

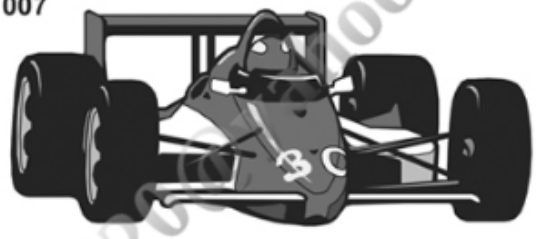
006

Turbo-Injected Porche: Working stick shift—3 mph and 5 mph froward; 3 mph reverse. High-speed lockout. Adjustable seat. Doors, trunk, and hood open. Simulated car phone. Includes one 18v battery and recharger. Ages 3–8 (carries two passengers up to 120 lbs. total)
Suggested retail price: $299.95

Model #007

007

Indy car: Dual motors for cruising on a variety of surfaces, even up hills. Two forward speeds (2 1/2 and 5 mph), plus reverse (2 1/2 mph). Adjustable seat. Includes two 6v batteries and recharger. Ages 3–7 (carries on passenger up to 80 lbs.)
Suggested retail price: $269.95



Model #008

008

2-Ton Pickup: Metallic teal color. Simulated chrome engine covers and headlight with over-sized wheels. 2 1/2 mph forward speed. Includes on 6v battery and recharger. Ages 3–7 (carries one passenger up to 65 lbs).
Suggested retail price: $189.95

DO NOT COPY
rituadhikari20@yahoo.com

Printed by: rituadhikari20@yahoo.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Model #008

Santa Fe Train: Soundly engineered for a little guy or gal. A hand-operated on/off button controls the 6v battery-operated motor. Reaches speeds to 5 mph. Includes a battery powered "whoo whoo" whistle to greet passersby. Ride on 76" x 168" oval track (sold separately) or on carpet or sidewalk, indoors or outdoors. Plastic body and flooboard; steel axles and coupling pins. Brigh red, blue and yellow body featrues a large lift-up seat and trailing car for storage. Includes battery and chargers. Ages 3–6.
Suggested retail price: $159.95

009

Model #010

Oval track: Measures 76" × 168."
Suggested retail price: $39.95

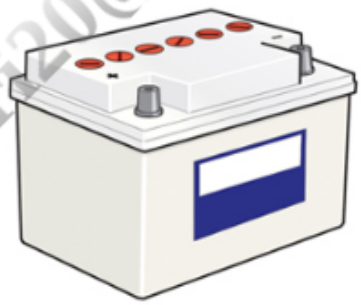
Model #011

6 Pieces Straight track: Six straight track section 19" each (total 109").
Suggested retail price: $19.95

Model #012

Rechargable battery (6v): For use with 6v or 12v vehicles. For 12v vehicles, use two. To charge, use charger included with vehicle.
Suggested retail price: $27.95

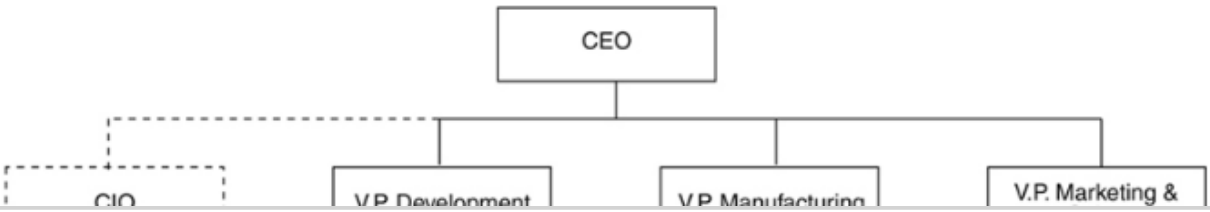
012



DO NOT COPY
rituadhikari20@yahoo.com

FIGURE 13.1 **Might-Mite Motors product catalog.**

At this time, MMM is organized into three divisions: Product Development, Manufacturing, and Marketing and Sales. Each division is headed by a vice president, who reports directly to the CEO. (An organization chart appears in Figure 13.2. The solid lines represent existing positions; dotted lines represent positions that will be added as part of the reengineering project.) All these divisions are housed in a single location that the corporation owns outright.



Printed by: rituadhikari20@yahoo.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

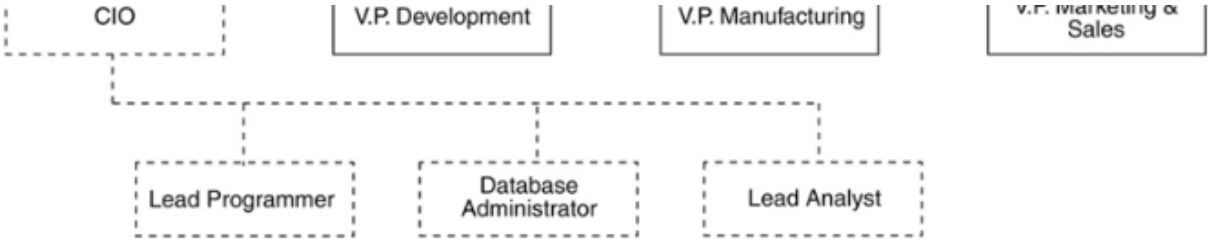


FIGURE 13.2 **Might-Mite Motors organization chart.**

Product Development Division

The Product Development division is responsible for designing and testing new and redesigned products. The division employs design engineers who use computer-aided design (CAD) software to prepare initial designs for new or redesigned vehicles. Once a design is complete, between one and 10 prototypes are built. The prototypes are first tested in-house using robotic drivers/passengers. After refinement, the prototypes are tested by children in a variety of settings. Feedback from the testers is used to refine product designs and to make decisions about which designs should actually be manufactured for mass marketing.

Manufacturing Division

The Manufacturing division is responsible for producing product for mass market sales. Manufacturing procures its own raw materials and manages its own operations, including personnel (hiring, firing, scheduling) and assembly line management. Manufacturing maintains the inventory of products ready for sale. It also handles shipping of products to resellers, based on sales information received from Marketing and Sales.

Marketing and Sales Division

MMM sells directly to toy stores and catalog houses; the corporation has never used distributors. Marketing and Sales employs a staff of 25 sales people who make personal contacts with resellers. Salespeople are responsible for distributing catalogs in their territories, visiting and/or calling potential resellers, and taking reseller orders. Order accounting is handled by Marketing and Sales. As noted earlier, Marketing and Sales transmits shipping information to Manufacturing, which takes care of actual product delivery.

Current Information Systems

MMM's information systems are a hodgepodge of computers and applications that have grown up with little corporate planning. The Product Development division relies primarily on stand-alone CAD workstations. In contrast to the sophistication of the CAD machines, testing records are kept and analyzed manually. Product Development employs product designers (some of whom function as project leaders) and clerical support staff, but no information systems personnel. Attempts to have clerical staff develop simple database applications to store data about children who test new and redesigned products and the results of those tests have proved futile. It has become evident that Product Development needs information systems professionals, and although the division is willing to hire IT staff, corporate management has decided to centralize the IT staff, rather than add to a decentralized model.

Manufacturing uses a stand-alone server to track purchases and inventory levels of raw materials, personnel scheduling, manufacturing line scheduling, and finished product inventory. Each of the applications running on the server were custom-written by outside consultants in COBOL, many years ago; the most significant maintenance they have had was when they were ported from the department's original minicomputer to the server, about 15 years ago. The data used by a Manufacturing application are contained in files that do not share information with any of the other applications. Manufacturing employs one COBOL programmer, who will be retiring shortly, and a system administrator. Although the programmer is talented, the most he can do is fix superficial user interface issues and repair corrupted data files; he was not part of the original program development and does not understand the functioning of much of the application code, which was poorly written and even more poorly documented. The applications no longer meet the needs of the Manufacturing division and management has determined that it isn't cost effective to write new applications to access the existing data files.

Marketing and Sales, which wasn't computerized until 1987, has a local area network consisting of one server and 15 workstations. The server provides shared applications, such as word processing and spreadsheets. It also maintains a marketing and sales database that has been developed using a PC-based product. The database suffers from several problems, including a limit of 10 users at one time and concurrency control problems that lead to severe data inconsistencies. The marketing and sales database was developed by the division's two IT employees at the time, both of whom have since left the company. No current staff understands the software. Regardless of the amount of time spent trying to maintain the database, inaccurate data continue to be introduced.

The Marketing and Sales network is not connected to the Internet. Sales people must therefore transmit hard copies of their orders to the central office, where the orders are manually keyed into the existing database. Some of the sales people do have laptop computers, but because the network has no Internet connection, the sales people cannot connect to it when they are out of the office.

Reengineering Project

Because MMM seems to have lost its strategic advantage in the marketplace, the CEO has decided to undertake a major systems reengineering project. The overall thrust of the project is to provide an information system that will support better evaluation of product testing, better analysis of sales patterns, better control of the manufacturing process, and enhanced communications options throughout the corporation. New information systems will be based on a client/server model and include one or more databases running on an Internet-connected network of servers, workstations, and PCs. The ultimate goal is to create a Web site for the company so resellers can place orders online.

New Information Systems Division

Printed by: rituadhikari20@yahoo.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

New Information Systems Division

The first step in the reengineering project is to establish an information technology division. This new division will also be housed in the corporate headquarters, along with the three existing divisions. To accommodate the new division, MMM will be constructing a 10,000 square foot addition to its building.

MMM is in the process of searching for a Chief Information Officer (CIO). This individual, who will report directly to the CEO, will manage the new division and be responsible for overseeing the reengineering of information systems that will handle all of the corporation's operations, as well as the creation, implementation, and maintenance of the company's Web presence.

All current IT personnel (those who work for the Manufacturing, and Marketing and Sales divisions) will be transferred to the new IT division. The division will hire (either internally or externally) three management-level professionals: a Lead Programmer (responsible for overseeing application development), a Database Administrator (responsible for database design and management), and a Lead Analyst (responsible for overseeing systems analysis and design efforts). The company will investigate various solutions for the development of the Web site, with an eye to determining whether it should be done in-house or outsourced to a consulting firm.

Basic System Goals

The CEO has defined the following goals for the reengineering project:

- Develop a corporation-wide data administration plan that includes a requirements document detailing organizational functions that require technology support and the functions that the reengineered system will provide.
- Provide an application roadmap that documents all application programs that will be needed to support corporate operations.
- Investigate alternatives for developing and hosting a Web site that will allow online orders. Conduct a cost-benefit analysis of those alternatives before beginning development.
- Document all databases to be developed for the corporation. This documentation will include ER diagrams and data dictionaries.
- Create a timeline for the development of applications and their supporting databases.
- Specify hardware changes and/or acquisitions that will be necessary to support the reengineered information systems.
- Plan and execute a security strategy for an expanded corporate network that will include both internal and external users.
- Implement the planned systems.

Note: It is important to keep in mind that the implementation of a Web presence for the company is relatively independent of the database design. The Web application will require the same data as all the other types of data entry. Concerns about hosting and security are rarely the job of the database designer.

Current Business Processes

To aid the systems analysts in their assessment of MMM's information systems needs, the CEO of MMM asked all existing division heads to document the way in which information is currently processed. This

DO NOT COPY
rituadhikari20@yahoo.com

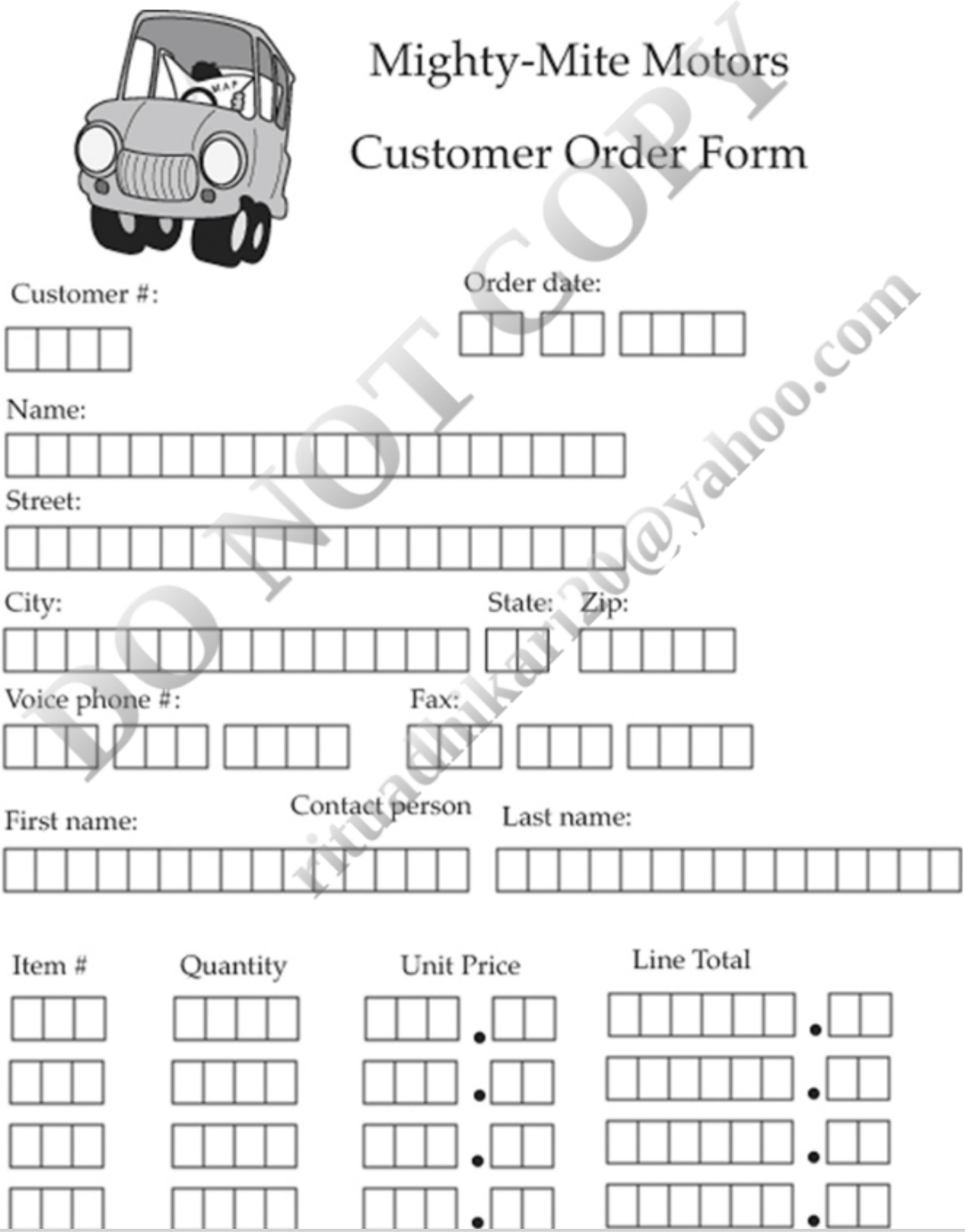
Printed by: rituadhikari20@yahoo.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

documentation, which also includes some information about what an improved system should do, provides a starting point for the redesign of both business and IT processes.

Sales and Ordering Processes

MMM receives orders at its plant in two ways: either by telephone directly from customers or from members of the sales staff who have visited customers in person. Orders from the remote sales staff usually arrive by fax or overnight courier.

Each order is taken on a standard order form (Figure 13.3). If the order arrives by fax, it will already be on the correct form. Telephone orders are written directly onto the form. Several times a day, a clerk enters the orders into the existing database. Unfortunately, if the sales office is particularly busy, order entry may be delayed. This backup has a major impact on production line scheduling and thus on the company's ability to fill orders. The new information system must streamline the order entry process, including online order entry, electronic transmission of order data from the field, and the direct entry of in-house orders.



DO NOT COPY
rituadhikari20@yahoo.com

Printed by: rituadhikari20@yahoo.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

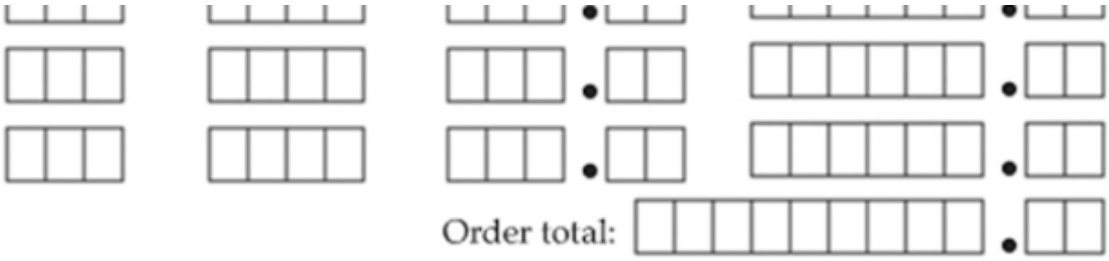


FIGURE 13.3 **Mighty-Mite Motors order form.**

The in-house sales staff has no access to the files that show the current finished-goods inventory. They are therefore unable to tell customers when their orders will ship. They can, however, tell customers how many orders are ahead of theirs to be filled and, based on general manufacturing timetables, come up with an approximation of how long it will take to ship a given order. Therefore, another goal of the information systems reengineering project is to provide better company-wide knowledge of how long it will take to fill customer orders.

Manufacturing, Inventory, and Shipping Processes

The MMM Manufacturing division occupies a large portion of the MMM facility. The division controls the actual manufacturing lines (three assembly lines), a storage area for finished goods, a storage area for raw materials, and several offices for supervisory and clerical staff.

The manufacturing process is triggered when a batch of order forms is received each morning by the manufacturing office. The batch consists of all orders that were entered into the sales database the previous working day. A secretary takes the individual order forms, and completes a report summarizing the number ordered by model (Figure 13.4). This report is then given to the Manufacturing Supervisor, whose responsibility it is to schedule which model will be produced on each manufacturing line, each day.

Mighty-Mite Motors
Order Summary

MM/DD/YYYY

Model #	Quantity Ordered
001	75
002	150
004	80
005	35
008	115
009	25
010	25
011	15

FIGURE 13.4 **Mighty-Mite Motors order summary report format.**

The scheduling process is somewhat complex, because the Manufacturing Supervisor must take into account previously placed orders, which have determined the current manufacturing schedule, and current inventory levels, as well as the new orders. The availability of raw materials and the time it takes to modify a manufacturing line to produce a different model also enter into the scheduling decision. This is one function that MMM's management understands will be almost impossible to automate; there is just too much human expertise involved to translate into an automatic process. However, it is vital that the Manufacturing Supervisor have access to accurate, up-to-date information about orders, inventory, and the current time schedule so that judgments can be made based on as much hard data as possible.

As finished vehicles come off the assembly line, they are packed for shipping, labeled, and sent to finished goods storage. Each shipping carton contains one vehicle, which is marked with its model number, serial number, and date of manufacturing. The Shipping Manager, who oversees finished goods storage and shipping, ensures that newly manufactured items are entered into the shipping inventory files.

The Shipping Manager receives customer order forms after the order report has been completed. (Photocopies of the order forms are kept in the Marketing and Sales office as backup.) The orders are placed in a box in reverse chronological order so that the oldest orders can be filled first. The Shipping Manager checks orders against inventory levels by looking at the inventory level output screen (Figure 13.5). If the manager sees that there is enough inventory to fill an order, then the order is given to a shipping clerk for processing. If there isn't enough inventory, then the order is put back in the box, where it will be checked again the following day. Under this system, no partial orders are filled, because they would be extremely difficult to track. (The reengineered information system should allow handling of partial shipments.)

Current Finished Goods Inventory Levels
MM/DD/YYYY

DO NOT COPY
rituadhikari20@yahoo.com

Printed by: rituadhikari20@yahoo.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

MM/DD/YYYY

Model #	Number on Hand
001	215
002	35
003	180
004	312
005	82
006	5
007	212
008	189
009	37
010	111
011	195
012	22

FIGURE 13.5 **Mite-Mite Motors inventory screen layout.**

Shipping clerks are given orders to fill. They create shipping labels for all vehicles that are part of a shipment. The cartons are labeled and set aside for pickup by UPS. The shipping clerks create UPS manifests, ensure that the items being shipped are removed from the inventory file, and return the filled orders to the Shipping Manager. The orders are then marked as filled and returned to Marketing and Sales. The reengineered information system should automate the generation of pick-lists, packing slips, and updating of finished-goods inventory.

MMM's raw materials inventory is maintained on a just-in-time basis. The Manufacturing Supervisor checks the line schedule (Figure 13.6) and the current raw materials inventory (Figure 13.7) daily to determine what raw materials need to be ordered. This process relies heavily on the Manufacturing Supervisor's knowledge of which materials are needed for which model vehicle. MMM's CEO is very concerned about this process because the Manufacturing Supervisor, while accurate in scheduling the manufacturing line, is nowhere near as accurate in judging raw materials needs. The result is that occasionally manufacturing must stop because raw materials have run out. The CEO would therefore like to see ordering of raw materials triggered automatically. The new information system should keep track of the raw materials needed to produce each model and, based on the line schedule and a reorder point established for each item, generate orders for items when needed.

Line Schedule
MM/DD/YYYY

MM/DD/YYYY		
	Line #1: Model 008	300 units
	Line #2: Model 002	150 units
	Line #3: Model 010	200 units
MM/DD/YYYY		
	Line #1: Model 008	200 units
	Line #2: Model 003	400 units
	Line #3: Model 005	300 units
MM/DD/YYYY		
	Line #1: Model 008	250 units
	Line #2: Model 006	100 units
	Line #3: Model 002	300 units

DO NOT COPY
rituadhikari20@yahoo.com

Printed by: rituadhikari20@yahoo.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

:
:
:

Total production scheduled:

Model 002	450 units
Model 003	400 units
Model 005	300 units
Model 006	100 units
Model 008	750 units
Model 010	200 units

FIGURE 13.6 Mighty-Mite Motors line schedule report format.

Current Raw Materials Inventory Levels
MM/DD/YYYY

Item #	Item	QOH
001	Plastic #3	95 lbs.
002	Red dye 109	25 gals.
003	Wheel 12"	120 each
004	Plastic #4	300 lbs.
005	Yellow dye 110	5 gals.
006	Yellow dye 65	30 gals.
007	Strut 15"	99 each
008	Axle 24"	250 each
009	Blu dye 25	18 gals.
010	Plastic #8	350 lbs.
011	Cotter pin: small	515 each
012	Cotter pin: medium	109 each

Next screen

FIGURE 13.7 Mighty-Mite Motors raw materials inventory screen layout.

Raw materials are taken from inventory each morning as each manufacturing line is set up for the day's production run. The inventory files are modified immediately after all raw materials have been removed from storage for a given manufacturing line. There is no way to automate the reduction of inventory; however, the new information system should make it very easy for nontechnical users to update inventory levels.

Product Testing and Support Function

MMM's top management makes decisions about which model vehicles to produce, based on data from three sources: product testing, customer registrations, and problem reports.

Customer registrations are received on cards packaged with sold vehicles (Figure 13.8). Currently, the registration cards are filed by customer name. However, MMM would also like access to these data by model and serial number to make it easier to notify customers if a recall occurs. Management would also like summaries of the data by model purchased, age of primary user, gender of primary user, and who purchased the vehicle for the child.

DO NOT COPY
rituadhikari20@yahoo.com

Printed by: rituadhikari20@yahoo.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Please register your Mighty-Mite Motors vehicle

By registering you receive the following benefits:

- Validation of the warranty on your vehicle, making it easier to obtain warranty service if ever necessary.
- Notification of product updates relating to your vehicle.
- Information mailings about enhancements to your vehicle and other products that may be of interest.

First name

Last name

Street

City State Zip

Phone #:

Model # Serial #

Age of primary user of vehicle: _____

Gender: ☐ Male ☐ Female

Fold Here. Tape closed; do not staple.

Date of purchase:

Place of purchase:

Where did you first learn about Mighty-Mite Motors?

- ☐ Advertisement in a magazine or newspaper
- ☐ Friend's recommendation
- ☐ In-store display
- ☐ Catalog
- ☐ Other

What features of the vehicle prompted your purchase?

- ☐ Size
- ☐ Color
- ☐ Speed
- ☐ Safety features
- ☐ Cost
- ☐ Other

What is the relationship of the purchaser to the primary user?

- ☐ Parent
- ☐ Grandparent
- ☐ Aunt/Uncle
- ☐ Friend
- ☐ Other

FIGURE 13.8 **Mighty-Mite Motors purchase registration form.**

Problem reports (Figure 13.9) are taken by customer support representatives who work within the product testing division. These reports include the serial number and model experiencing problems, along with the date and type of problem. Currently, the problem descriptions are nonstandard, made up of whatever terms the customer support representative happens to use. It is therefore difficult to summarize problem reports to get an accurate picture of which models are experiencing design problems that should be corrected. MMM would therefore like to introduce a standardized method for describing problems, probably through a set of problem codes. The result should be regular reports on the problems reported for each model that can be used to help make decisions about which models to continue, which to discontinue, which to redesign, and which to recall.

DO NOT COPY
rituadhikari20@yahoo.com

Problem Report

Date Time

First name

Last name

Printed by: rituadhikari20@yahoo.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Street

City State Zip

Phone #:

Model # Serial #

Problem Description:

FIGURE 13.9 **Mighty-Mite Motors problem report.**

MMM does not repair its own products. When a problem report is received, the customer is either directed to return the product to the store where it was purchased for an exchange (during the first 30 days after purchase) or directed to an authorized repair center in the customer's area. In the latter case, the problem report is faxed to the repair center, so that it is waiting when the customer arrives. MMM does not plan to change this procedure because it currently provides quick, excellent service to customers and alleviates the need for MMM to stock replacement parts. However, the fax transmissions could be replaced with electronic data sent over the Internet.

Product test results are recorded on paper forms (Figure 13.10). After a testing period is completed, the forms are collated manually to produce a summary of how well a new product performed. MMM would like the test results stored within an information system so that the testing report can be produced automatically, saving time and effort. Such a report will be used to help decide which new models should be placed in production.

Product Test Report

Date Time

DO NOT COPY
rituadhikari20@yahoo.com

Printed by: rituadhikari20@yahoo.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

Location

Model tested:

Test type:

Test description

Test result and comments:

FIGURE 13.10 **Mighty-Mite Motors product test report.**

DO NOT COPY
rituadhikari20@yahoo.com

Designing the Database

The most effective approach to the design of a database (or collection of databases) for an environment as diverse as that presented by Mighty-Mite Motors usually involves breaking the design into components indicated by the organization of the company. As the design evolves, the designer can examine the entities and the relationships to determine where parts of the organization will need to share data. Working on one portion of the design at a time also simplifies dealing with what might at first seem to be an overwhelmingly large database environment. Paying special attention for the needs for shared data helps ensure that shared data are consistent and suitable for all required uses.

A systems analysis indicates that the MMM database environment falls into the following areas:

- Manufacturing (including finished goods inventory and raw materials ordering).
- Sales to toy stores, and shipping of products ordered.
- Purchase registrations.
- Testing.
- Problem handling.

Examining the Data Flows

During the systems analysis, a data flow diagram can be of enormous use in identifying where data are shared by various parts of an organization. The top-level DFD (the *context diagram* in Figure 13.11) actually tells us very little. It indicates that there are three sources outside the company provide data: customers (the stores to which the company sells), purchasers (the individuals who purchase products from the stores), and raw materials suppliers. Somewhere, all those data are used by a general process named "Manufacture and Sell Products" to keep the company in business.

Printed by: rituadhikari20@yahoo.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

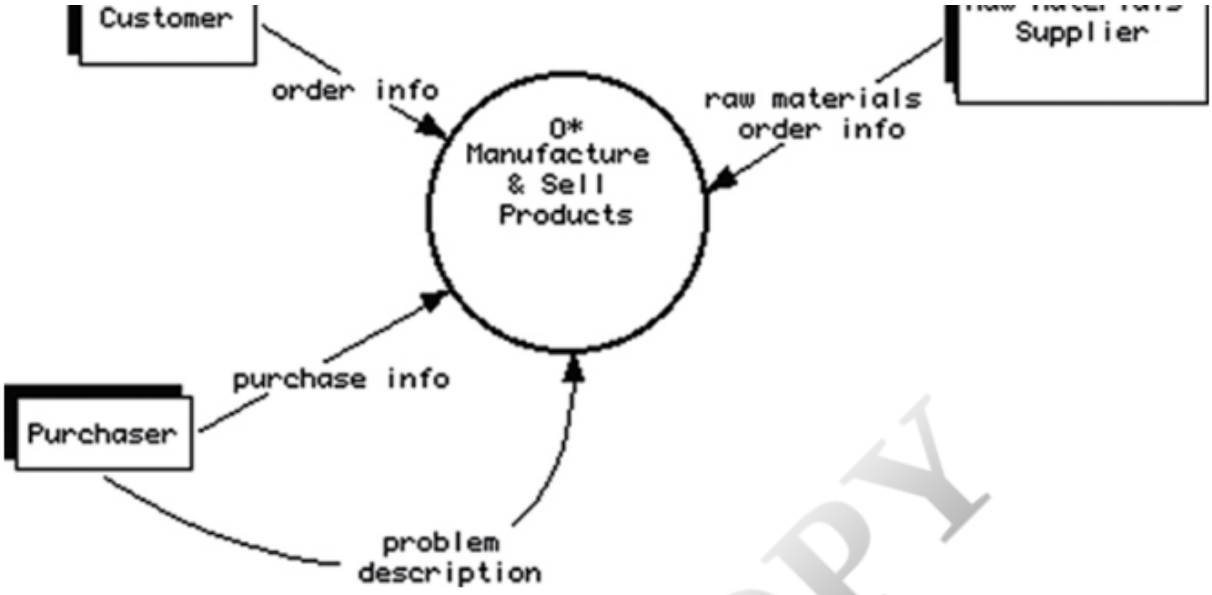


FIGURE 13.11 **Context DFD for Mighty-Mite Motors.**

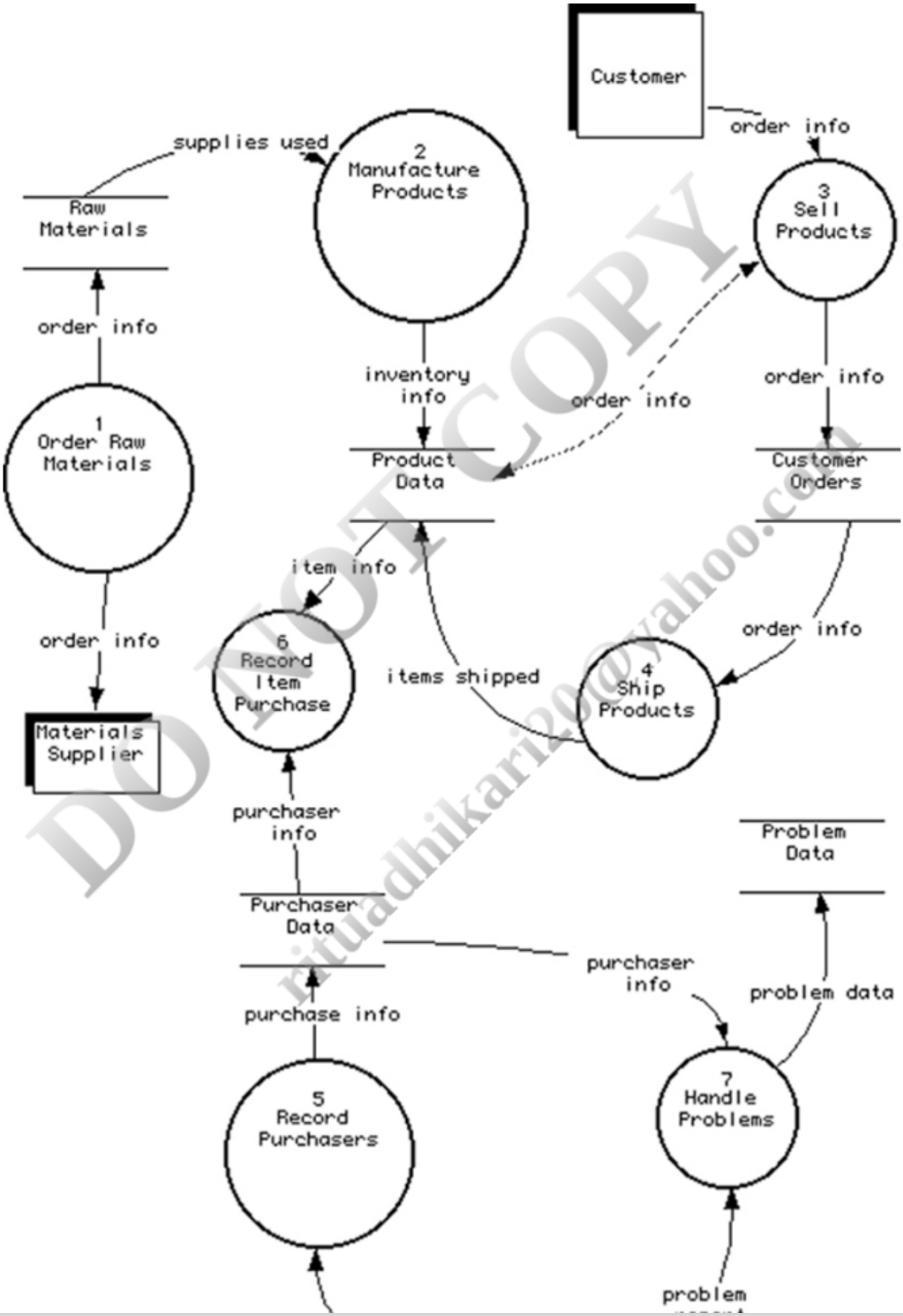
However, the level 1 DFD (Figure 13.12) is much more telling. As the data handling processes are broken down, five data stores emerge:

- Raw materials: This data store holds both the raw materials inventory and the orders for raw materials.
- Product data: The product data store contains data about the products being manufactured, product testing results, and the finished goods inventory.
- Customer orders: This data store contains customer information, as well as order data.

DO NOT COPY
rituadhikari20@yahoo.com

Printed by: rituadhikari20@yahoo.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

- Purchaser data: The purchaser data store contains information about the individuals who purchase products and the products they have purchased.
- Problem data: This final data store contains problem reports.



DO NOT COPY
rituadhikari20@yahoo.com

Printed by: rituadhikari20@yahoo.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

FIGURE 13.12 **Level 1 DFD for Mighty-Mite Motors.**

As you examine the processes that interact with these five data stores, you will find a number of processes that manipulate data in more than one data store, as well as data stores that are used by more than one process:

- The raw materials data store is used by the raw materials ordering and the manufacturing processes.
- Product data are used by manufacturing, sales, shipping, and product registration.
- Customer order data are used by sales and shipping.
- The purchases data store is used by purchaser registration and problem handling.
- The problem data store, used only by problem handling, is the only data store not shared by multiple processes.

The raw materials ordering process is the only process that uses only a single data store. Nonetheless, the level 1 DFD makes it very clear that there is no instance in which a single process uses a single data store without interaction with other data stores and processes. Given that each process in the DFD probably represents all or part of an application program, this suggests that the database designer should consider either a single database or a set of small databases, along with software to facilitate the interchange of data.

The DFD makes it very clear that the need for the integration of the various data stores is very strong. In addition, Mighty-Mite Motors is a relatively small business and therefore a single database that manages all needed aspects of the company will not grow unreasonably large. It will also be more cost effective and perform better than multiple databases that use some type of middleware to exchange data. Ultimately, the database designer may decide to distribute the database onto multiple servers, placing portions of it that are used most frequently in the division where that use occurs. The database design, however, will be the same, regardless of whether the final implementation is centralized or distributed. The essential decision is to create a single database rather than several smaller, interrelated databases that must exchange data.

The ER Diagram

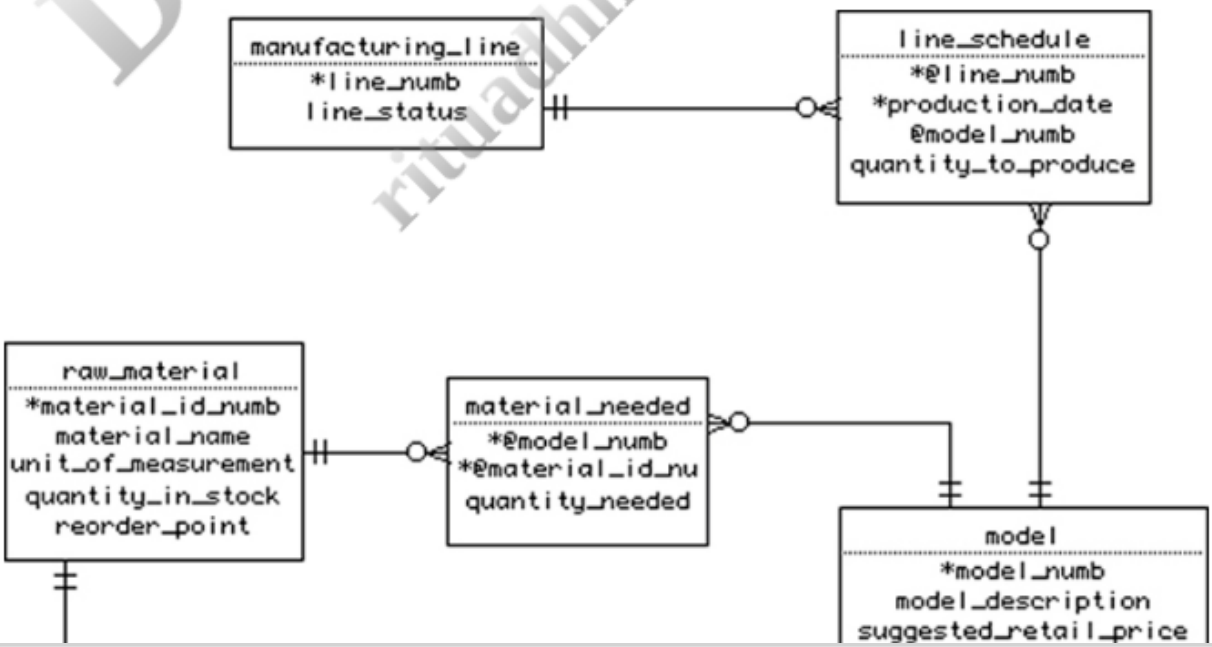
The systems analyst preparing the requirements document for the Mighty-Mite Motors reengineering project has had two very good sources of information about exactly what needs to be stored in the database: the employees of the company and the paper documents that the company has been using. The document that is given to the database designer is therefore quite complete.

The design needs to capture all the information on the paper documents. Some documents are used only for input (for example, the product registration form or the order form). Others represent reports that an application program must be able to generate (for example, the line schedule report). Although the current documents do not necessarily represent all the outputs application programs running against the database will eventually prepare, they do provide a good starting place for the design. Whenever the designer has questions, he or she can then turn to Might-Mite's employees for clarification.

Working from the requirements document prepared by the systems analyst, along with the paper input and output documents, the database designer puts together the ER diagram. Because there are so many entities, all of which interconnect, the diagram is very wide. It has therefore been split into three pieces so you can see it. As you look at each piece, keep in mind that entities that appear on more than one piece represent the connection between the three illustrations.

The first part (found in Figure 13.13) contains the entities for raw materials and manufacturing. This portion of the data model is dealing with three many-to-many relationships:

- *material_order* to *raw_material* (resolved by the composite entity *material_order_line*),
- *raw_material* to *model* (resolved by the composite entity *material_needed*),
- *manufacturing_line* to *model* (resolved by the composite entity *line_schedule*).



Printed by: rituadhikari20@yahoo.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

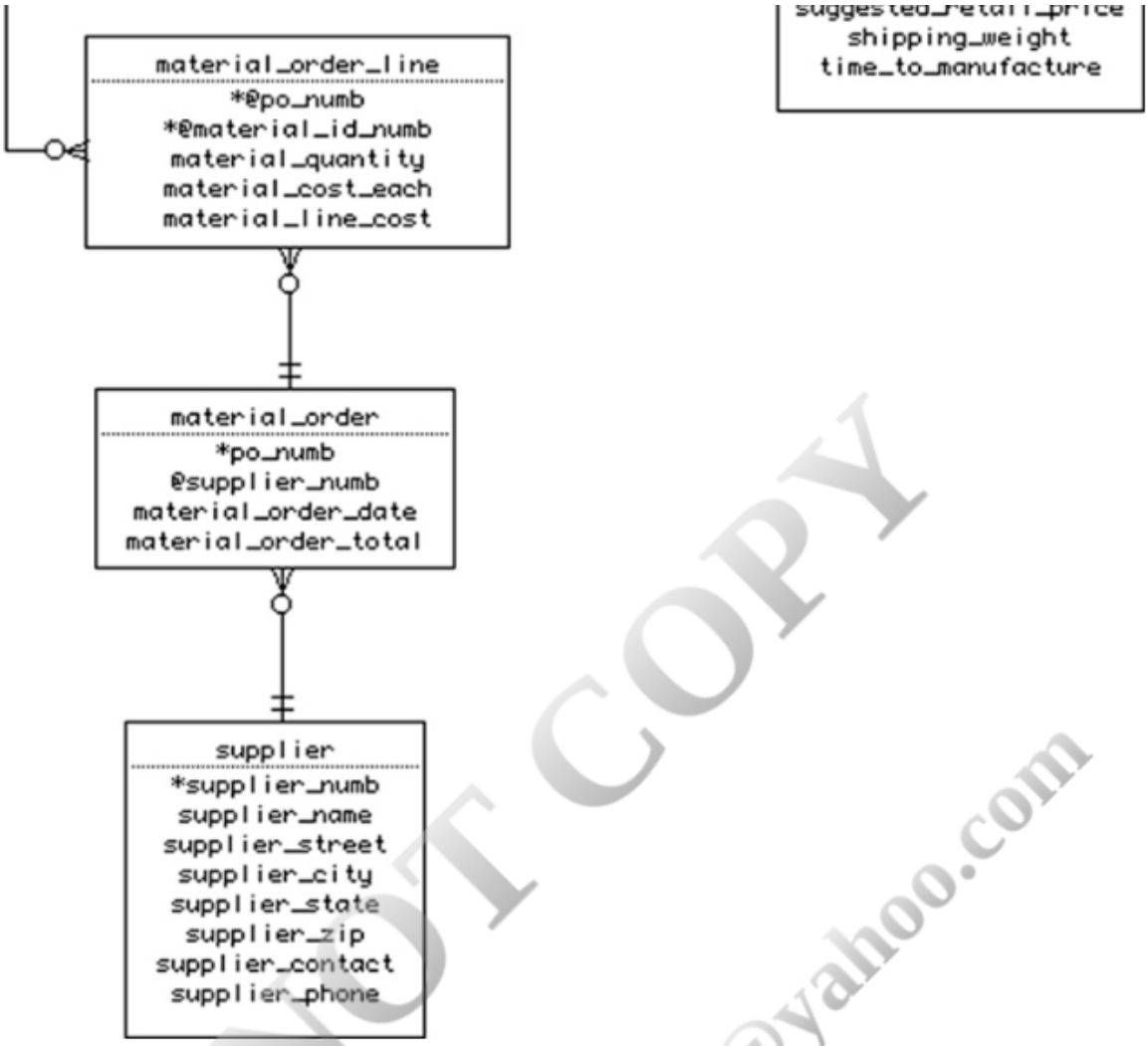
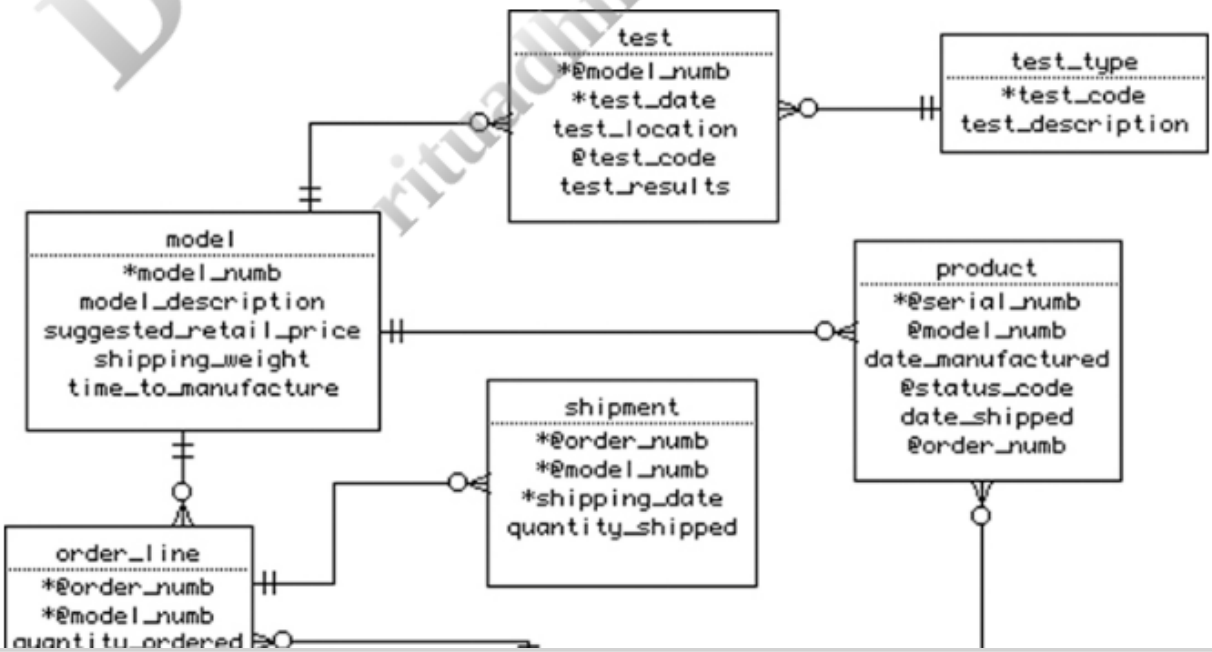


FIGURE 13.13 **Might-Mite Motors ERD (part 1).**

The second portion of the ERD (Figure 13.14) contains entities for product testing and sales. (Remember that in this instance, the customers are toy stores rather than individual purchasers.) There are two many-to-many relationships:

- *test_type* to *model* (resolved by the *test* entity),
- *order* to *model* (resolved by the *order_line* composite entity).



Printed by: rituadhikari20@yahoo.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

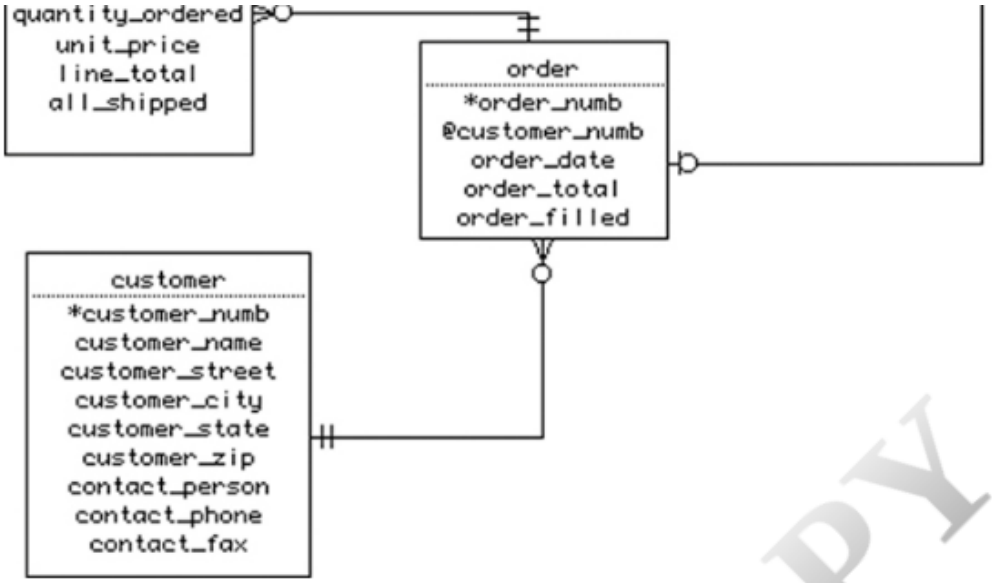


FIGURE 13.14 **Might-Mite Motors ERD (part II).**

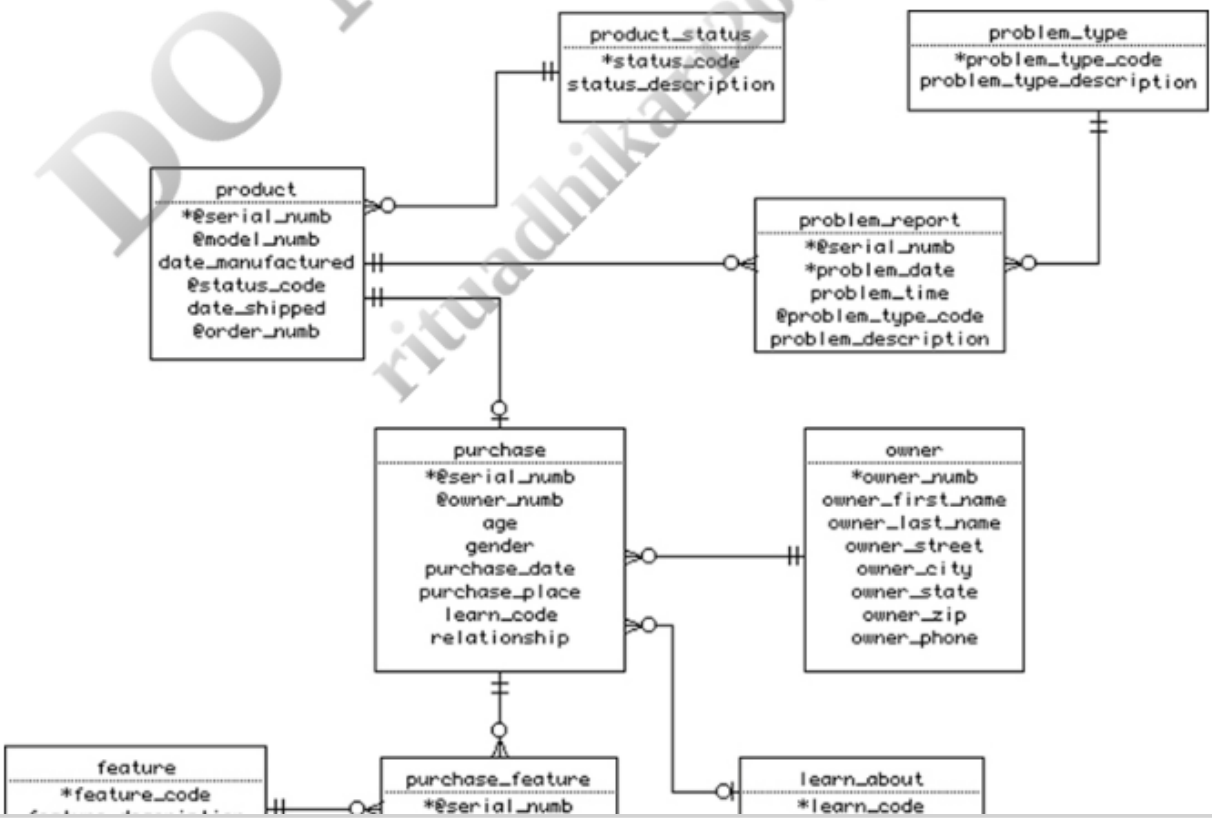
The test entity is somewhat unusual for a composite entity. It is an activity that someone performs and, as such, has an existence outside the database. It is not an entity created just to resolve a many-to-many relationship.

At this point, the diagrams become a bit unusual because of the need to keep track of individual products rather than simply groups of products of the same model. The *model* entity, which you first saw in Figure 13.13, represents a type of vehicle manufactured by Mighty-Mite Motors. However, the *product* entity, which first appears in Figure 13.14, represents a single vehicle that is uniquely identified by a serial number. This means that the relationships between an order, the line items on an order, and the models and products are more complex than for most other sales database designs.

The *order* and *line_item* entities are fairly typical. They indicate how many of a given model are required to fill a given order. The *shipment* entity then indicates how many of a specific model are shipped on a specific date. However, the database must also track the order in which individual products are shipped. As a result, there is a direct relationship between the *product* entity and the *order* entity in addition to the relationships between *order_line* and *model*. In this way, Mighty-Mite Motors will know exactly where each product has gone. At the same time, the company will be able to track the status of orders (in particular, how many units of each model have yet to ship).

The final portion of the ERD (Figure 13.15) deals with the purchasers and problem reports. There are two many-to-many relationships:

- *problem_type* to *product* (resolved with the entity *problem_report*),
- *purchase* to *feature* (resolved with the composite entity *purchase_feature*).



Printed by: rituadhikari20@yahoo.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

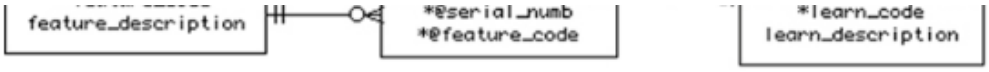


FIGURE 13.15 **Mighty-Mite Motors ERD (part III).**

As with the test entity that you saw earlier, the *problem_report* entity acts like a composite entity to resolve a many-to-many relationship, but is really a simple entity. It is an entity that has an existence outside the database and was not created simply to take care of the M:N relationship.

Note: Calling an entity "problem_report" can be a bit misleading. In this case, the word "report" does not refer to a piece of paper, but to the action of reporting a problem. A "problem_report" is therefore an activity rather than a document. In fact, the printed documentation of a problem report will probably include data from several entities, including the product, problem_report, purchase, and owner entities.

If you look closely at the diagram, you'll notice that there is a one-to-one relationship between the product and purchase entities. The handling of the data supplied by a purchaser on the product registration card presents an interesting dilemma for a database designer. Each product will be registered by only one purchaser. (Even if the product is later sold or given to someone else, the new owner will not have a registration card to send in.) There will be only one set of registration data for each product, at first thought suggesting that all the registration data should be part of the *product* entity.

However, there is a lot of registration data—including one repeating group (the features for which the purchaser chose the product, represented by the *feature* and *purchase_feature* entities)—and the product is involved in a number of relationships that have nothing to do with product registration. If the DBMS has to retrieve the registration data along with the rest of the product data, database performance will suffer. It therefore makes sense in this case to keep the purchase data separate and to retrieve it only when absolutely needed.

Note: One common mistake made by novice database designers is to create an entity called "registration card." It is important to remember that the card itself is merely an input document. What is crucial is the data the card contains and the entity that the data describe, rather than the medium on which the data are supplied.

Creating the Tables

The tables for the Mighty-Mite Motors database can come directly from the ER diagram. They are as follows:

```
model (model_numb, model_description, suggested_retail_price,
     shipping_weight, time_to_manufacture)

test (model_numb, test_date, test_location, test_code,
     test_results)

test_types (test_code, test_description)

customers (customer_numb, customer_name, customer_street,
     customer_city, customer_state, customer_zip,
     contact_person, contact_phone, contact_fax)

orders (order_numb, customer_numb, order_date, order_total,
     order_filled)

order_line (order_numbzu, model_numb, quantity_ordered,
     unit_price, line_total, all_shipped)

shipments (order_numb, model_numb, quantity_shipped)

product (serial_numb, model_numb, date_manufactured,
     status_code, order_numb, date_shipped)

raw_material (material_id_numb, material_name,
     unit_of_measurement, quantity_in_stock, reorder_point)

supplier (supplier_numb, supplier_name, supplier_street,
     supplier_city, supplier_state, supplier_zip,
     supplier_contact, supplier_phone)

material order (po numb, supplier numb, material order date,
```

DO NOT COPY
rituadhikari20@yahoo.com

Printed by: rituadhikari20@yahoo.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

```
material_order (po_numb, supplier_numb, material_order_date,
     material_order_total)

material_order_line (po_numb, material_id_numb,
     material_quantity, material_cost_each, material_line_cost)

manufacturing_line (line_numb, line_status)

line_schedule (line_numb, production_date, model_numb,
     quantity_to_produce)

owner (owner_numb, owner_first_name, owner_last_name,
     owner_street, owner_city, owner_state, owner_zip,
     owner_phone)

purchase (serial_numb, owner_numb, age, gender, purchase_date,
     purchase_price, learn_code, relationship)

purchase_feature (serial_numb, feature_code)

learn_about (learn_code, learn_description)

feature (feature_code, feature_description)

problem_report (serial_numb, problem_date, problem_time,
     problem_type_code, problem_details)

problem_type (problem_type_code, problem_type_description)
```

Generating the SQL

Assuming that the designers of the Mighty-Mite Motors database are working with a CASE tool, then generating SQL statements to create the database can be automated. For example, in Figure 13.16 you will find the SQL generated by Mac A&D from the ER diagram you saw earlier in this chapter.

DO NOT COPY
rituadhikari20@yahoo.com

Printed by: rituadhikari20@yahoo.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

```
CREATE TABLE model
(
      model_numb INTEGER,
      model_description VARCHAR (40),
      suggested_retail_price NUMBER (6,2),
      shipping_weight NUMBER(6,2),
      time_to_manufacture TIME,
      PRIMARY KEY (model_numb)
);

CREATE TABLE test_type
(
      test_code INTEGER,
      test_description VARCHAR (40),
      PRIMARY KEY (test_code)
);

CREATE TABLE test
(
      test_date DATE,
      test_location VARCHAR (40),
      test_code INTEGER,
      test_results VARCHAR (256),
      PRIMARY KEY (model_numb, test_date),
      FOREIGN KEY (model_numb) REFERENCES model,
      FOREIGN KEY (test_code) REFERENCES test_type
);

CREATE TABLE customer
(
      customer_numb INTEGER,
      customer_name VARCHAR (40),
      customer_street VARCHAR (50),
      customer_city VARCHAR (50),
      customer_state CHAR (2),
      customer_zip CHAR (10),
      contact_person VARCHAR (30),
      contact_phone CHAR (12),
      contact_fax CHAR (12),
      PRIMARY KEY (customer_numb)
);
```

```
CREATE TABLE order
(
      order_numb INTEGER,
```

DO NOT COPY
rituadhikari20@yahoo.com

Printed by: rituadhikari20@yahoo.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

```
      order_numb INTEGER,
      customer_numb INTEGER,
      order_date DATE,
      order_total NUMBER (8,2),
      order_filled BOOLEAN,
      PRIMARY KEY (order_numb),
      FOREIGN KEY (customer_numb) REFERENCES customer
);

CREATE TABLE order_line
(
      order_numb INTEGER,
      model_numb INTEGER,
      quantity_ordered INTEGER,
      unit_price NUMBER (6,2),
      line_total NUMBER (8,2),
      all_shipped BOOLEAN,
      PRIMARY KEY (order_numb, model_numb),
      FOREIGN KEY (order_numb) REFERENCES order,
      FOREIGN KEY (model_numb) REFERENCES model
);

CREATE TABLE shipment
(
      order_numb INTEGER,
      model_numb INTEGER,
      shipping_date DATE,
      quantity_shipped INTEGER,
      PRIMARY KEY (order_numb, model_numb, shipping_date),
      FOREIGN KEY (order_numb, model_numb) REFERENCES order_line
);

CREATE TABLE product
(
      serial_numb INTEGER,
      model_numb INTEGER,
      date_manufactured DATE,
      status_code INTEGER,
      date_shipped DATE,
      order_numb INTEGER,
      PRIMARY KEY (serial_numb),
      FOREIGN KEY (model_numb) REFERENCES model,
      FOREIGN KEY (status_code) REFERENCES product_status,
      FOREIGN KEY (order_numb) REFERENCES order
);

CREATE TABLE product_status
(
      status_code INTEGER,
```

DO NOT COPY
rituadhikari20@yahoo.com

Printed by: rituadhikari20@yahoo.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

```
      status_description VARCHAR (40),
      PRIMARY KEY (status_code)
);
```

```
CREATE TABLE raw_material
(
      material_id_numb INTEGER,
      material_name VARCHAR (40),
      unit_of_measurement CHAR (12),
      quantity_in_stock INTEGER,
      reorder_point INTEGER,
      PRIMARY KEY (material_id_numb)
);

CREATE TABLE material_needed
(
      model_numb INTEGER,
      material_id_numb INTEGER,
      quantity_needed INTEGER,
      PRIMARY KEY (model_numb, material_id_numb),
      FOREIGN KEY (model_numb) REFERENCES model,
      FOREIGN KEY (material_id_numb) REFERENCES raw_material
};

CREATE TABLE supplier
(
      supplier_numb INTEGER,
      supplier_name VARCHAR (40),
      supplier_street VARCHAR (50),
      supplier_city VARCHAR (50),
      supplier_state CHAR (2),
      supplier_zip CHAR (10),
      supplier_phone CHAR (12),
      PRIMARY KEY (supplier_numb)
};

CREATE TABLE material_order
(
      po_numb INTEGER,
      supplier_numb INTEGER,
      material_order_date DATE,
      material_order_total NUMBER (8,2),
      PRIMARY KEY (po_numb),
      FOREIGN KEY (supplier_numb) REFERENCES supplier
);

CREATE TABLE material_order_line
(
      po_numb INTEGER,
```

DO NOT COPY
rituadhikari20@yahoo.com

Printed by: rituadhikari20@yahoo.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

```
      material_id_numb INTEGER,
      material_quantity INTEGER,
      material_cost_each NUMBER (6,2),
      material_line_cost NUMBER (8,2),
      PRIMARY KEY (po_numb, material_id_numb),
      FOREIGN KEY (po_numb) REFERENCES material_order,
      FOREIGN KEY (material_id_numb) REFERENCES raw_material
}
```

```
CREATE TABLE manufacturing_line
(
      line_numb INTEGER,
      line_status CHAR (12),
      PRIMARY KEY (line_numb)
);

CREATE TABLE line_schedule
(
      line_numb INTEGER,
      production_date DATE,
      model_numb INTEGER,
      quantity_to_product INTEGER
      PRIMARY KEY (line_numb, production_date),
      FOREIGN KEY (lne_numb) REFERENCES manufacturing_line,
      FOREIGN KEY (model_numb) REFERENCES model
);

CREATE TABLE owner
(
      owner_numb INTEGER,
      owner_street VARCHAR (50),
      owner_city VARCHAR (50),
      owner_state CHAR (2),
      owner_zip CHAR (10),
      owner_phone CHAR (10),
      PRIMARY KEY (owner_numb)
};

CREATE TABLE purchase
(
      serial_numb INTEGER,
      owner_numb INTEGER,
      age INTEGER,
      gender CHAR (1),
      purchase_date DATE,
      purchase_place VARCHAR (50)
```

DO NOT COPY
rituadhikari20@yahoo.com

Printed by: rituadhikari20@yahoo.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

```
        purchase_place VARCHAR (50),
        learn_code INTEGER,
        relationship CHAR (10),
        PRIMARY KEY (serial_numb),
        FOREIGN KEY (serial_numb) REFERENCES product,
        FOREIGN KEY (owner_numb) REFERENCES owner
        FOREIGN KEY (learn_code) REFERENCES learn_about
);

CREATE TABLE feature
(
        feature_code INTEGER,
        feature_description VARCHAR (40),
        PRIMARY KEY (feature_code)
);
```

```
CREATE TABLE purchase_feature
(
        serial_numb INTEGER,
        feature_code INTEGER,
        PRIMARY KEY (serial_numb, feature_code),
        FOREIGN KEY (serial_numb) REFERENCES product,
        FOREIGN KEY (feature_code) REFERENCES feature
};

CREATE TABLE learn_about
(
        learn_code INTEGER,
        learn_description VARCHAR (50),
        PRIMARY KEY (learn_code)
};

CREATE TABLE problem_type
(
        problem_type_code INTEGER,
        problem_type_description VARCHAR (50),
        PRIMARY KEY (problem_type_code)
);

CREATE TABLE problem_report
(
        serial_numb INTEGER,
        problem_date DATE,
        problem_time TIME,
        problem_type_code INTEGER,
        problem_details VARCHAR (50)
```

DO NOT COPY
rituadhikari20@yahoo.com

Printed by: rituadhikari20@yahoo.com. Printing is for personal, private use only. No part of this book may be reproduced or transmitted without publisher's prior permission. Violators will be prosecuted.

```
        problem_details VARCHAR (50),
        PRIMARY KEY (serial_numb, problem_date),
        FOREIGN KEY (serial_numb) REFERENCES product,
        FOREIGN KEY (product_type_code) REFERENCES problem_type
};
```

FIGURE 13.16 **SQL statements needed to create the Mighty-Mite Motors database.**

DO NOT COPY
rituadhikari20@yahoo.com