

2.1 Introduction: scenario-based learning

This tutorial package uses a single business problem—order management at a kitchen supply company—to illustrate many different information systems issues. We recognize that it is unlikely that the business problems you face have much in common with wholesaling kitchen gadgets. However, there are important advantages to introducing and sticking with a single scenario.

1. You get to work through a problem from start to finish. You start with the business problem, build a conceptual model, create a physical design that satisfies the requirements, and implement the design in software.
2. By sticking with a single scenario, we do not incur the setup cost of constantly introducing and describing different business environments.
3. Even though the kitchen supply scenario used here is simple, it contains enough complexity to highlight interesting technical and process design issues. Such issues do not generally emerge without looking at a problem in depth.

The purpose of this lesson is to present the business scenario in detail and set the stage for the remainder of the lessons in this tutorial package.

2.2 Business-to-business for little guys

You run a small-but-growing company that imports and distributes kitchen gadgets. Your product line consists of stainless steel utensils, ceramic serving dishes, small electrical appliances, and so on.

Your suppliers are manufacturing firms from around the world and your customers are primarily small retailers such as kitchen specialty stores and hardware stores. In other words, you are a middleman (or middleperson, if you prefer): you aggregate products from a global network of suppliers and market them to other small businesses in your region.

2.2.1 Your “value proposition”

The middleman is generally thought to be an endangered species in the information age. There is a risk that your customers (the mom-and-pop kitchen and hardware stores) will somehow get connected directly to the manufacturers and **disintermediate** you. That



is, they will use the Internet to find your suppliers and submit their orders directly, without the costs and benefits of your participation.

To stay in business, it is important that you provide your customers with something more than simply another layer of cost. Your speciality might be finding unique products. Or you may have nurtured strong relationships with certain suppliers that provide you with preferential access to the newest and hottest products. Either way, one thing is certain: changes in the wholesaling environment mean that you must operate at a very high level of efficiency in order to remain viable.

2.2.2 Scope of the project

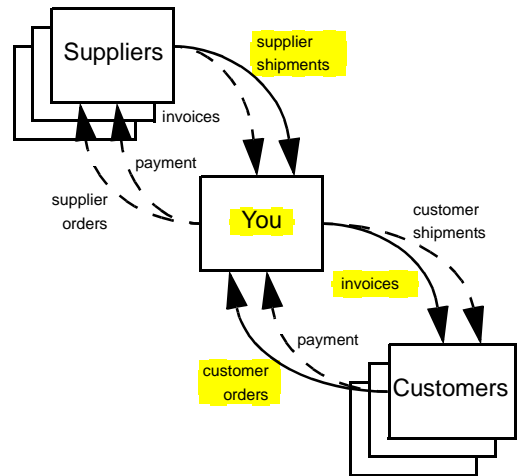
The essential flows of information between you and your suppliers and customers are shown in Figure 2.1. Basically, you buy goods from your suppliers and keep them in inventory until they are ordered by customers.

To keep the scope of the scenario manageable, we are going to focus exclusively on the information flows shown by solid arrows and ignore everything else. The "everything else" includes:

- information flows indicated by dashed arrows;

- physical flows, such as the goods themselves; and,
- information flows to organizations that are not shown in Figure 2.1, such as banks, trucking companies, and so on.

FIGURE 2.1: The primary flows of information in the kitchen supply environment.



As you will hopefully discover, developers of information systems use the same tricks over and over. Thus, if you have the skills and



knowledge to create a simple order entry system, you can use the same skills and knowledge to add additional functionality, such as tracking accounts receivable, managing backorders, reconciling invoices with shipping notices, and so on. The objective here is not to create a realistic order management system; rather, it is to introduce just enough complexity to keep the lessons interesting.

2.2.3 Business processes

In the following sections the critical business process and information flows in [Figure 2.1](#) are discussed in greater detail.

2.2.3.1 Customer orders

From time to time, a customer (or a member of your small sales force) faxes you an order for products. For each item in the order, you check inventory (to determine the quantity you have in stock), decide how many of each item to ship (you cannot ship what you do not have), and verify the price of the product (the person who submitted the order might be using out-of-date prices).

Once you have determined the correct price and quantity to ship for each item in the order, you create an invoice. The people in your warehouse use the invoice to determine what to ship to the customer. The final invoice is

included in the physical shipment to the customer and the customer is expected to pay the amount on the invoice in accordance with your payment terms.

2.2.3.2 Backorders

A backorder occurs whenever a customer orders a product that you currently do not have in stock. At this early stage of the tutorials, we are simply going to ignore backorders. Later on, in [Lesson 21](#), you will add the capability to manage backorders to your application.

2.2.3.3 Supplier shipments

Since you are a middleman, it is natural that the fulfillment processes on the outbound (customer) side are mirrored by similar business processes on the inbound (supplier) side. In order to replenish your own inventory of products, you submit orders to your suppliers, and receive shipments and invoices in response. At this point, we are going to ignore the inbound side of the business and focus on customer orders. In [Lesson 20](#) you will add the capability to track incoming supplier shipments to your application.

2.2.4 Your existing information system

Your current system for managing the information flows in [Figure 2.1](#) is largely based



on spreadsheets, paper records, and manual effort.

For example, you currently keep track of all your customers, orders, and inventory using different spreadsheets. Although this approach works reasonably well, there are at least three major problems:

1. You create all your invoices by cutting and pasting from your orders spreadsheet into a specialized invoice spreadsheet. This is error prone and tedious.
2. You and your employees sometimes make errors entering data and thus the values in your inventory spreadsheet are suspect. You spend a great deal of time verifying inventory levels by physically counting items.
3. You find it very difficult to make business sense of all the data in your spreadsheets. For example, calculating the sales-per-customer for your hottest selling products typically takes you most of an afternoon. And you have to repeat the whole process every quarter.
4. Words that are *not* currently in your vocabulary include: backup, transaction atomicity, auditability, and referential integrity. In short, you have a nagging suspicion that your current system is less

sophisticated than it should be given the important role it plays in your business.

2.2.5 A wish list

You are interested in building a small transaction processing system that will provide the following core functionality:

- allow orders to be entered into the computer as they are received from customers and salespeople,
- track inventory levels of each SKU¹ as product is shipped to your customers
- automatically generate customer invoices.

There are other functions—such as recording shipments from your suppliers, managing back orders, and flagging items to reorder—that you want to add later. In addition, you are investigating the possibility of providing direct ordering via the Internet for certain customers. However, you prefer to move slowly since you do not have a great deal of experience with database software or information systems generally.

¹ Firms typically assign a number to each product they keep in stock and thus you may hear the term SKU (pronounced “skew”) to refer the “stock keeping units.” In the scenario used here, you are dealing with a very small number of SKUs.



You have decided to use MICROSOFT ACCESS to develop your application. Although you examined other packages, you chose ACCESS because it was reasonably priced (it was bundled in an office suite) and because the reviews you read were favorable.

Ideally, selection of a software package would have come after an in-depth requirements analysis. However, at this point, your objective is to build a prototype system to learn more about the benefits of a database system. Down the road, your plan is either to hire a professional developer to clean-up and extend your prototype application or buy an off-the-shelf package. You are certain that whichever path you follow, the experience you gain implementing the prototype system will enable you to make much better decisions regarding the use of technology to support the order management process in your company.

2.3 Project package

To help you build a prototype system, you are provided with the following information and documents:

1. **A complete inventory** – The current quantity on hand information for each product you carry has been stored in a plain text file. The file can be found in the project package as [package\inventor.txt](#).

2. **Sales orders** – Your customers and sales representatives fax you sales orders every couple of days (orders are normally filled and invoiced in the sequence that they arrive). Electronic copies of these orders available in project package as [package\orders.pdf](#).
3. **Current backorders and shipments** – The project package also contains a list of backordered products and details of a small number of shipments from your suppliers. You will not require these files until you extend the functionality of your system in [Lesson 20](#) and [Lesson 21](#).

2.4 Discussion: automate and informate

A distinction is sometimes made between the use of technology to **automate** and the use of technology to **informate**.¹ The distinction between the two concepts can be applied to the kitchen supply scenario to refine our thinking about the role of a database in the context of the business problem.

¹ The automate versus informate distinction was introduced by Shoshana Zuboff in her 1988 book: *In the age of the smart machine: the future of work and power*.



2.4.1 Automating the business

Looking up inventory levels and generating invoices are both menial tasks that should be automated. Your value to your customers lies in your ability to maintain relationships with your suppliers and find interesting products that people want to buy. Every minute that you spend cobbling together an invoice or tracking down a backorder is a lost opportunity to do what you are supposed to be doing.

The concept of automating a process using a computer is straightforward enough. The difficulty lies in implementation—making the automated process work the way it is supposed to work. As discussed in [Lesson 3](#), we are going to start with a data-centric view of the operation and add automation incrementally. A useful way to think about automation is:

1. What needs to be done? What are the steps in the process?
2. What data is required? For example, to know how much of a particular product to ship to a customer, you need to know how much the customer has ordered and how much you have on hand. Other processes have more complex data requirements. A simple rule of thumb is: if you do not have the data, you cannot automate the process.
3. When should the automated process occur? Is the process triggered automatically or

manually? Does the process occur in response to the passage of time (such as a monthly closing) or in response to some business event (such as a stockout)?

The critical issues in automation are not which programming language you use or what type of computer you are running. Instead, what is important is an understanding of what needs to be done and when from a business perspective.

2.4.2 Informating the business process

Two things happen when you process a customer's order:

1. you execute the transaction requested by the customer, and
2. you learn something about what the customer wants.

Many organizations focus on execution—that is, on automation of business processes. However, it is important to recognize that every time you process a transaction in your day-to-day operations, you create information about your business.

For example, if you process a large number of customer orders and store the information about the orders in an appropriate manner, it is possible to learn new and important things from this data: Who buys what? When do they buy it? And so on. This information can enable you to



operate more efficiently and provide a better level of customer service.

The problem is that although many people talk about the benefits of informing the organization, it is not always clear how one should go about achieving these benefits. This is especially true in small organizations.

Throughout the lessons that follow, you should step back from the nitty-gritty details of implementation from time to time and situate your progress with respect to the underlying business problem. Are you automating a tedious task? Are you storing data that can be used to support decision-making?

2.5 Application to the project

- ➔ Ensure you understand the business scenario described in this lesson and the flows of information shown in [Figure 2.1](#).