EXTENDED LEARNING MODULE J

IMPLEMENTING A DATABASE WITH MICROSOFT ACCESS

Student Learning Outcomes

1. IDENTIFY THE STEPS NECESSARY TO IMPLEMENT THE STRUCTURE OF A RELATIONAL DATABASE USING THE DATA DEFINITION LANGUAGE PROVIDED BY MICROSOFT ACCESS.

2. DEMONSTRATE HOW TO USE THE DATA MANIPULATION SUBSYSTEM IN ACCESS TO ENTER AND CHANGE INFORMATION IN A DATABASE AND HOW TO QUERY THAT INFORMATION.

3. EXPLAIN THE USE OF THE APPLICATION GENERATION SUBSYSTEM IN ACCESS TO CREATE REPORTS AND DATA ENTRY SCREENS.

Introduction

In Chapter 3, we discussed the important role that databases play in an organization. We followed that with *Extended Learning Module C,* in which you learned how to design the correct structure of a relational database. That module includes four primary steps. They are:

1. Define entity classes and primary keys.
2. Define relationships among the entity classes.
3. Define information (fields) for each relation (the term *relation* is often used to refer to a file in the context of a database).
4. Use a data definition language to create your database.

In *Extended Learning Module C,* you followed the process through the first three steps above. In this module, we'll take you through the fourth step—using a data definition language to create your database—by exploring the use of Microsoft Access, today's most popular personal database management system package (it comes as part of Microsoft's Office Professional suite).

We'll also show you how to use Microsoft Access's data manipulation subsystem, including how to enter and change information and build queries; and how to use the application generation subsystem to create reports and input forms.

In Figure J.1 (on this page and the next) we've recreated the complete Solomon Enterprises database structure we defined in *Extended Learning Module C.* If it has been a while since you covered that module, we suggest that you review it before creating the database.

Figure J.1

The Database Structure We'll Be Implementing *(continued on the next page)*

CONCRETE TYPE RELATION

Concrete Type	Type Description
1	Home foundation and walkways
2	Commercial foundation and walkways
3	Premier speckled (with smooth gravel aggregate)
4	Premier marble (with crushed marble aggregate)
5	Premier shell (with shell aggregate)

CUSTOMER RELATION

Customer Number	Customer Name	Customer Phone	Customer Primary Contact
1234	Smelding Homes	3333333333	Bill Johnson
2345	Home Builders Superior	3334444444	Marcus Connolly
3456	Mark Akey	3335555555	Mark Akey
4567	Triple A Homes	3336666666	Janielle Smith
5678	Sheryl Williamson	3337777777	Sheryl Williamson
6789	Home Makers	3338888888	John Yu

EMPLOYEE RELATION

Employee ID	Employee Last Name	Employee First Name	Date of Hire
123456789	Johnson	Emilio	2/1/1985
435296657	Evaraz	Antonio	3/3/1992
78934444	Robertson	John	6/1/1999
984568756	Smithson	Allison	4/1/1997

Figure J.1

The Database Structure We'll Be Implementing *(continued)*

SUPPLIER RELATION

Supplier ID	Supplier Name
412	Wesley Enterprises
444	Juniper Sand & Gravel
499	A&J Brothers
999	N/A

TRUCK RELATION

Truck Number	Truck Type	Date of Purchase
111	Ford	6/17/1999
222	Ford	12/24/2001
333	Chevy	1/1/2002

ORDER RELATION

Order Number	Order Date	Customer Number	Delivery Address	Concrete Type	Amount	Truck Number	Driver ID
100000	9/1/2004	1234	55 Smith Lane	1	8	111	123456789
100001	9/1/2004	3456	2122 E. Biscayne	1	3	222	785934444
100002	9/2/2004	1234	55 Smith Lane	5	6	222	435296657
100003	9/3/2004	4567	1333 Burr Ridge	2	4	333	435296657
100004	9/4/2004	4567	1333 Burr Ridge	2	8	222	785934444
100005	9/4/2004	5678	1222 Westminster	1	4	222	785934444
100006	9/5/2004	1234	222 East Hampton	1	4	111	123456789
100007	9/6/2004	2345	9 W. Palm Beach	2	5	333	785934444
100008	9/6/2004	6789	4532 Lane Circle	1	8	222	785934444
100009	9/7/2004	1234	987 Furlong	3	8	111	123456789
100010	9/9/2004	6789	4532 Lane Circle	2	7	222	435296657
100011	9/9/2004	4567	3500 Tomahawk	5	6	222	785934444

RAW MATERIAL RELATION

Raw Material ID	Raw Material Name	QOH	Supplier ID
A	Water	9999	999
B	Cement paste	400	412
C	Sand	1200	444
D	Gravel	200	444
E	Marble	100	499
F	Shell	25	499

BILL OF MATERIAL RELATION

Concrete Type	Raw Material ID	Unit
1	B	1
1	C	2
1	A	1.5
2	B	1
2	C	2
2	A	1
3	B	1
3	C	2
3	A	1.5
3	D	3
4	B	1
4	C	2
4	A	1.5
4	E	2
5	B	1
5	C	2
5	A	1.5
5	F	2.5

IMPLEMENTING THE STRUCTURE OF THE SOLOMON ENTERPRISES DATABASE

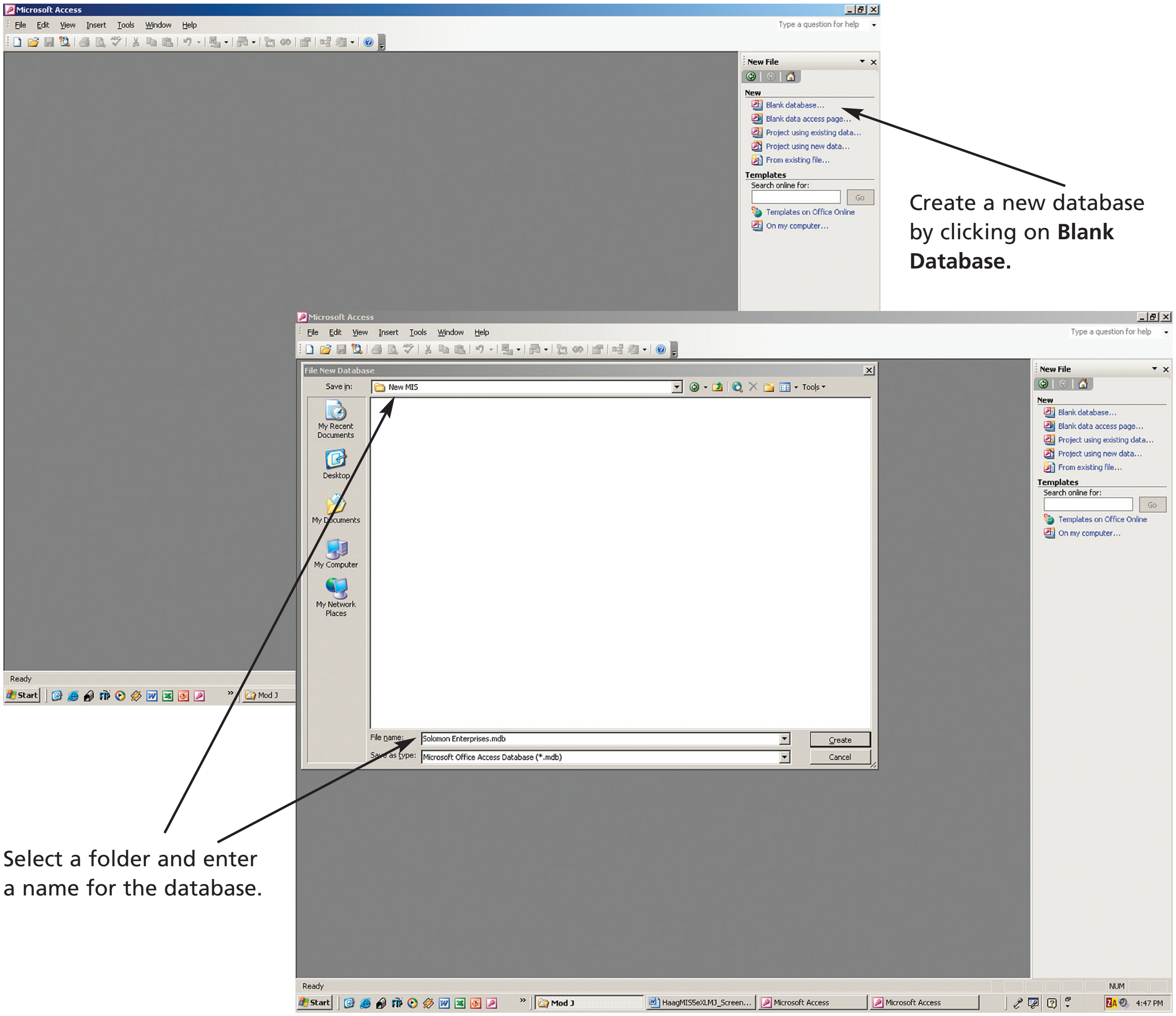
As we said previously, you can't simply start typing information into a database as you can when you create a document with Word or a workbook with Excel. You must first define the correct structure of the database (see *Extended Learning Module C*), and then create the structure of the database by creating its data dictionary before entering any information. The ***data dictionary*** contains the logical structure for the information in a database. It includes a description of each relation (also called a *table* or *file*) and each piece of information in each relation.

To create a database using Microsoft Access, we performed the following steps (see Figure J.2).

1. Start Microsoft Access.
2. Select **Blank Database** on the right side of the screen or select **File** and **New** from the menu and then **Blank Database** on the right side of the screen. (We used **Solomon Enterprises.mdb** as the database name.)
3. Click on **Create.**

Figure J.2

The First Step in Creating a Database



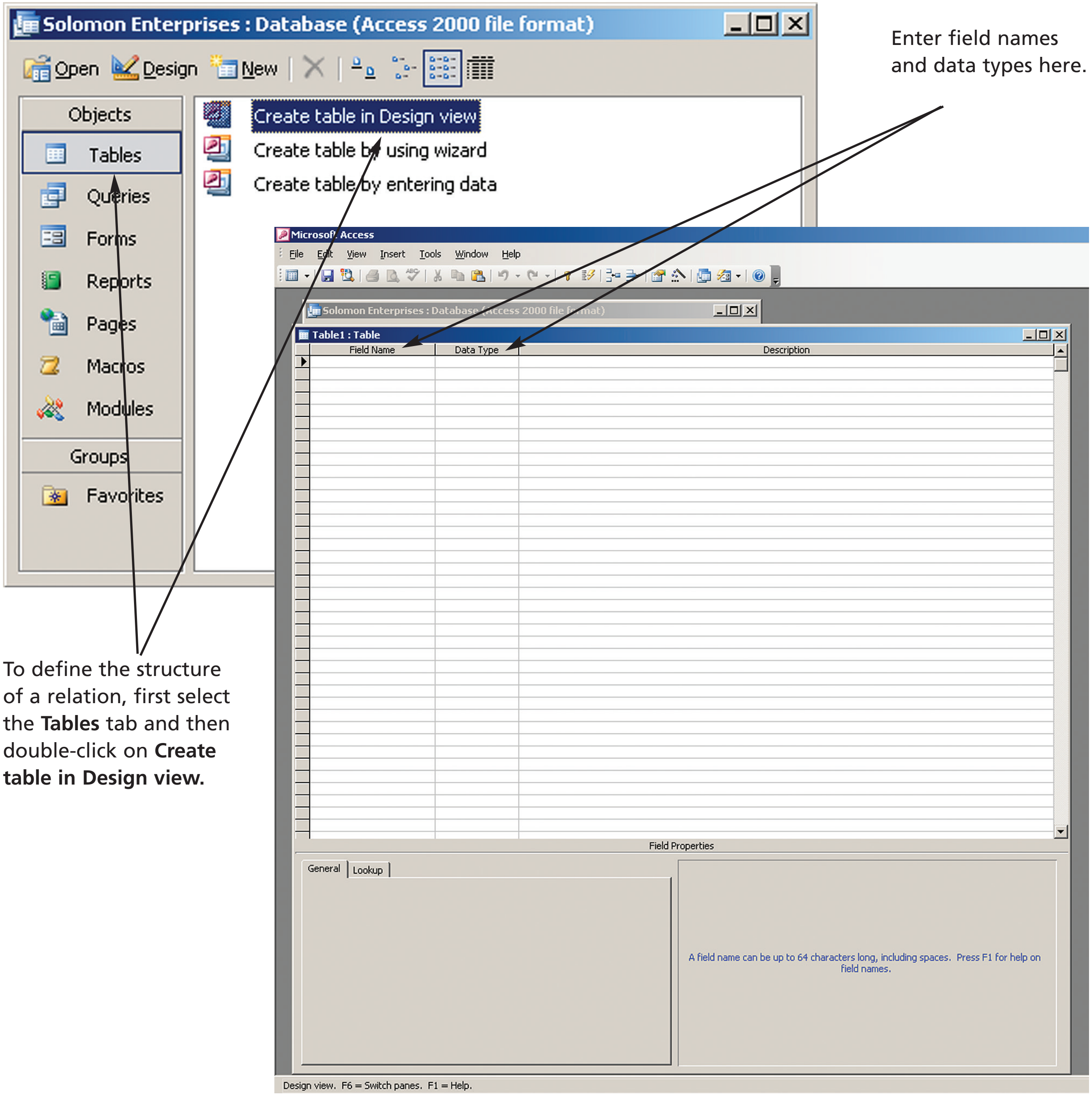
Save in: New MIS

File name: Solomon Enterprises.mdb

Save as type: Microsoft Office Access Database (*.mdb)

Create a new database by clicking on **Blank Database.**

Select a folder and enter a name for the database.



A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Design view. F6 = Switch panes. F1 = Help.

Enter field names and data types here.

To define the structure of a relation, first select the **Tables** tab and then double-click on **Create table in Design view.**

Figure J.3

The First Step in Defining the Structure for Each Relation

After we followed these steps, we got the left screen in Figure J.3. Now that we've created a blank database with the name **Solomon Enterprises.mdb,** we're ready to define the structure and information within each relation.

Take a look at the left screen in Figure J.3. Notice first that you can create relations (the term Microsoft uses for tables) in one of three ways: in Design view; by using the wizard; and by simply entering information. They all achieve the same result. In this module, we'll be creating the relations using the Design view.

Notice also the tabs down the left side of that same screen. They include **Objects, Tables, Queries, Forms, Reports, Pages, Macros,** and **Modules.** To work within each, you must first click on its respective tab. The default is **Tables** and we'll begin there. In this module, we'll also use **Queries, Reports** and **Forms,** leaving **Objects, Pages, Macros,** and **Modules** up to you to explore at a later time.

When creating a relation (table) using the Design view, you have to make sure that the **Tables** tab is selected and then double-click on **Create table in Design view.** You'll then see the screen on the right side of Figure J.3 (on the previous page). In that screen you can see that Microsoft Access expects you enter a field name, data type, and description (the last is optional) for each field in a given relation. Once you've entered all that information (and perhaps some more) for a given relation, you save that structure and repeat the process until you've created the structure for each relation in your database. We'll do that for three of the relations in Solomon's database in the next section.

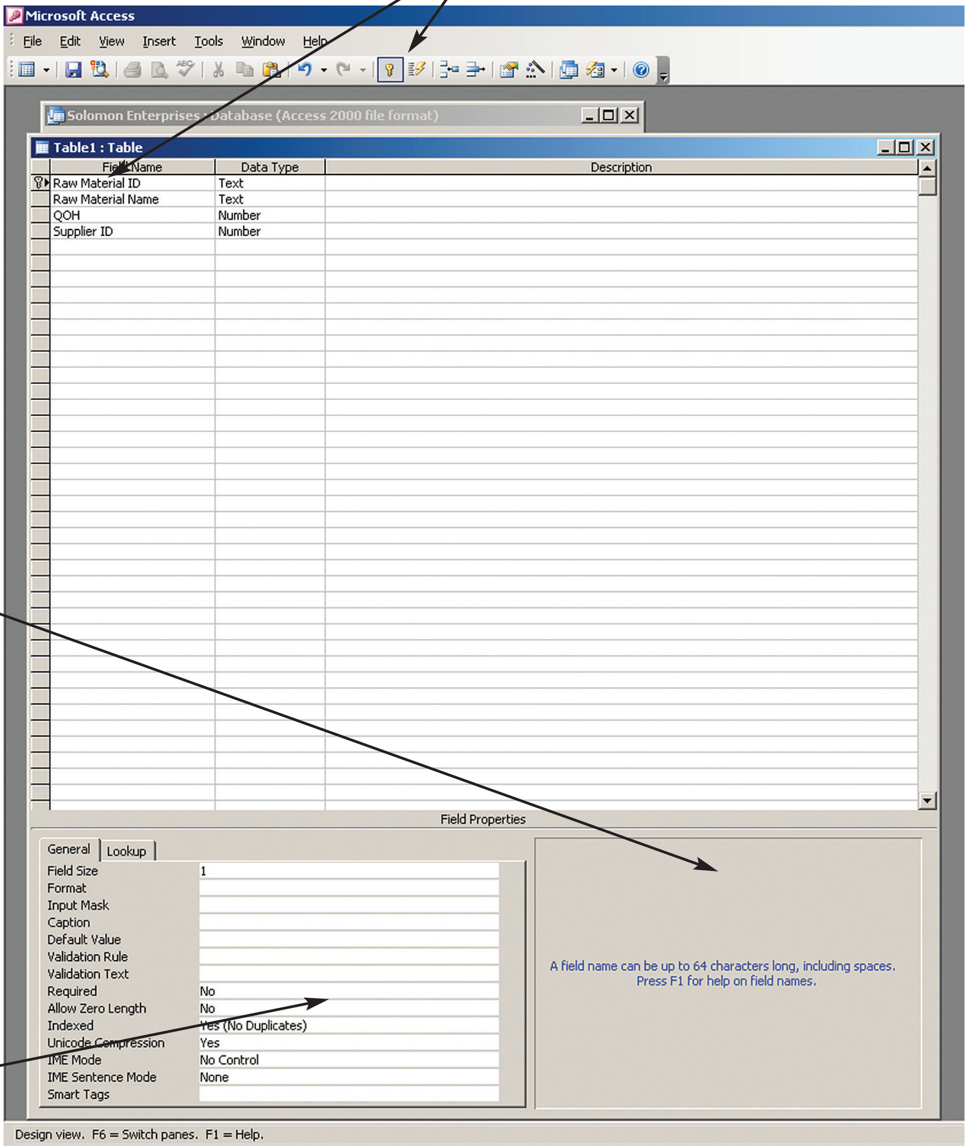
IMPLEMENTING THE *RAW MATERIAL* RELATION STRUCTURE

To start, let's implement the structure for the *Raw Material* relation within **Solomon Enterprises.mdb.** The *Raw Material* relation has four fields: *Raw Material ID, Raw Material Name, QOH,* and *Supplier ID.* As you can see in Figure J.4, we have entered

Figure J.4

Creating the *Raw Material* Relation Structure

To identify a primary key, place your cursor in the appropriate row and click on the key button. The key icon will then appear on the left of the field name.



Field Name	Data Type	Description
Raw Material ID	Text	
Raw Material Name	Text	
QOH	Number	
Supplier ID	Number	

Field Properties

Field Size	1
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	No
Indexed	Yes (No Duplicates)
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

Design view. F6 = Switch panes. F1 = Help.

This box contains information about the item where the cursor is sitting. In this case, that's the field name column.

This block shows the field properties for *Raw Material ID,* the primary key. Notice that it is one character long and does not allow for zero length or duplicates.

these four fields for the *Raw Material* relation. For *Raw Material ID,* we chose the data type as **Text,** 1 character long. The data type is to the right of the name of the field and the length of the field is down below in the **Field Properties** box. For *Raw Material Name,* we chose the data type as **Text** once again, but this time we didn't restrict the length to 1 character, but rather chose the default which allows up to 50 characters, although you can't see that in Figure J.4 since you only see the properties of the field where the cursor is. We have to choose a data type for each field. Both *QOH* and *Supplier ID* have data types of **Number,** with **Field Properties** specified as integers (i.e. whole numbers with no decimal places).

As you can see, the **Field Properties** box has more than just the length of the field. For example, *Raw Material ID* is also set to **Yes** for the **Indexed** option and ensures that no duplicates are allowed. This is necessary, of course, for the primary key and prevents the person entering the information from putting in the same *Raw Material ID* for two different raw materials. This is an illustration of how we can enforce integrity constraints. ***Integrity constraints*** are rules that help ensure the quality of the information. You'll see more on that topic a little later.

The *Raw Material ID* field is also set not to allow zero length (i.e. an ID must be entered) and to require this field to be filled when entering information. This ensures that each record in the *Raw Material* relation has a primary key field. You should use this option sparingly and only when necessary. For example, you might want to enter a new type of raw material, but don't yet have a supplier for this item. If you set *Supplier ID* to zero length here, you won't be able to enter the new raw material until you have a supplier lined up.

Finally, notice that we identified *Raw Material ID* as the primary key by placing a key symbol next to its name field. A ***primary key*** is a field (or group of fields in some cases) that uniquely describes each record. To identify a primary key, position the cursor in the line of the primary key field and click on the **Key** button in the button bar. We need to do this for each primary key in each relation.

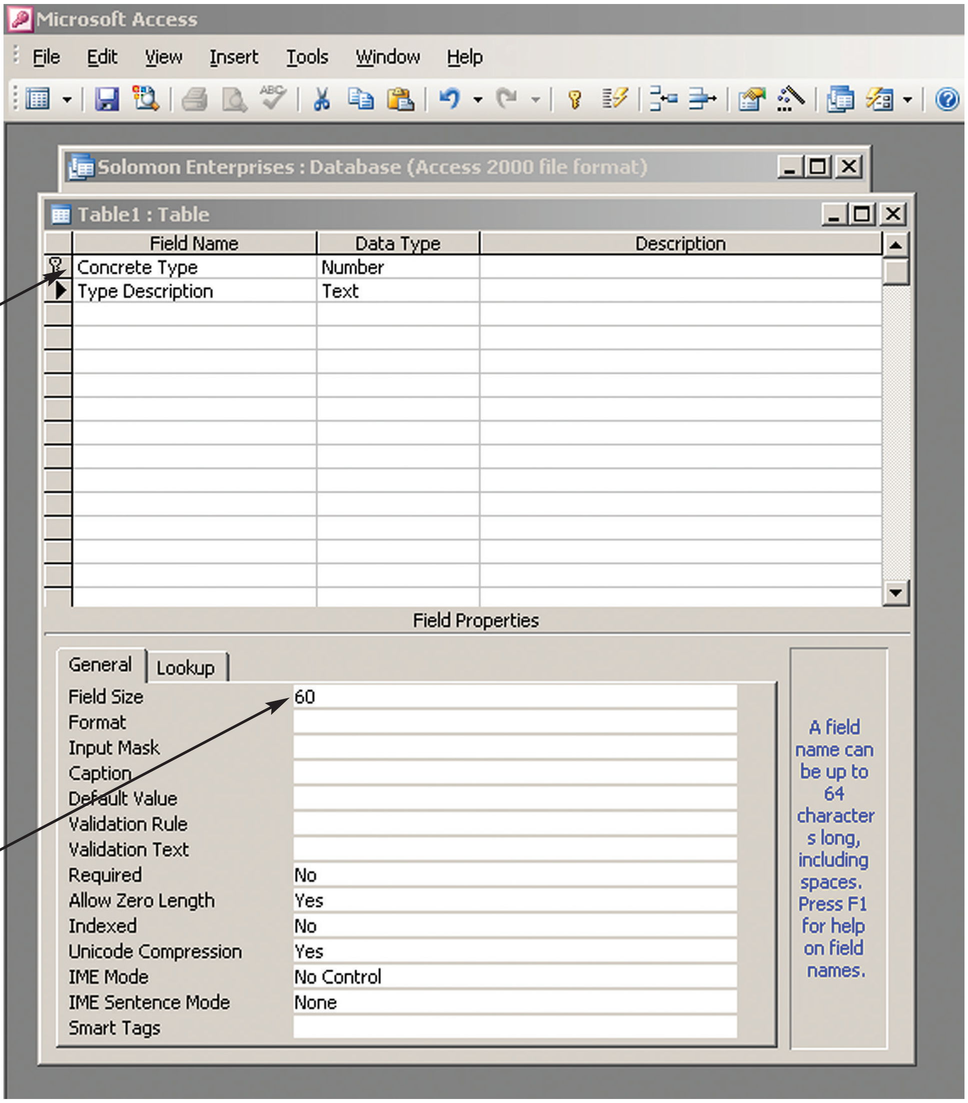
When you have the structure the way you want it, you need to save it. So, we clicked on the **Disk** button (**Save as**), entered *Raw Material* as the table name, and clicked on **OK.** Then we closed the **Table:Raw Material** window. If you forget to save the table structure and try to close the window, Access will ask if you want to save it and give you a chance to do so.

IMPLEMENTING THE *CONCRETE TYPE* RELATION

To develop the structures of the remaining relations, you simply follow the process outlined in the previous section. Let's now create the structure for the *Concrete Type* relation. To create the *Concrete Type* relation using the Design view, make sure the **Tables** tab is selected and then double-click on **Create table in Design view.**

Figure J.5

Creating the *Concrete Type* Relation Structure


Microsoft Access
File Edit View Insert Tools Window Help
Solomon Enterprises : Database (Access 2000 file format)
Table1 : Table
Field Properties
General
Lookup

Field Name	Data Type	Description
Concrete Type	Number	
Type Description	Text	

Field Size	60
Format	
Input Mask	
Caption	
Default Value	
Validation Rule	
Validation Text	
Required	No
Allow Zero Length	Yes
Indexed	No
Unicode Compression	Yes
IME Mode	No Control
IME Sentence Mode	None
Smart Tags	

A field name can be up to 64 character s long, including spaces. Press F1 for help on field names.

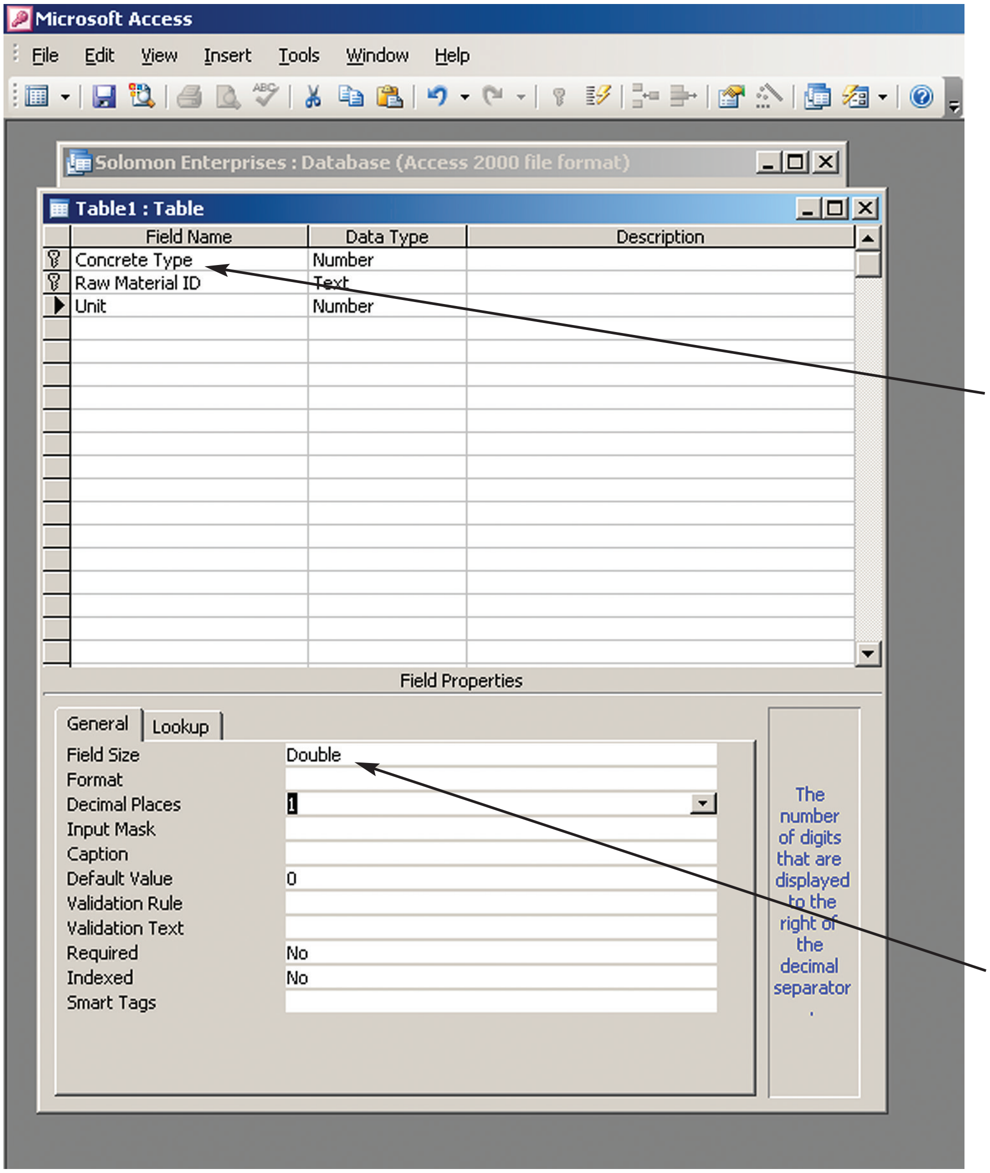
Concrete Type is the primary key.

This box shows the field properties for *Type Description.* We have changed the length of the field (**Field Size**) from its default size of 50 to 60 to make sure there's enough room in case future descriptions are larger. Always make sure text fields are large enough to handle any entry but not so large that you waste space.

In Figure J.5, you can see that we entered the field names for *Concrete Type* and *Type Description. Concrete Type* is also the name of the table, and that's fine if it suits your purposes. *Concrete Type* is a numeric field and *Type Description* is a text field. *Concrete Type* is a primary key field as you can see by the Key icon to the left of the field name.

Let's take a look at the **Field Properties** for *Type Description* (we placed the cursor in the *Type Description* row so that we can see the **Field Properties**). Since this is a text field, the default size (**Field Size**) is 50 characters, which we changed to 60 characters to ensure that there will be enough space for future descriptions. The longest one we have in the database so far is *Premier speckled (with smooth gravel aggregate)* which has 47 characters. There may be longer descriptions in the future, and we want to allow plenty of room. The largest that a **Field Size** can be is 255 characters, but if you make text fields that big when they don't need to be, you'll waste a lot of storage space, especially if you have a lot of records in your database.

To save this relation, we followed the same process as before. That is, we clicked on the **Disk** button (**Save As**), entered *Concrete* as the table name, clicked on **OK,** and closed the box.


Microsoft Access
File Edit View Insert Tools Window Help
Solomon Enterprises : Database (Access 2000 file format)
Table1 : Table
Field Properties
General
Lookup

Field Name	Data Type	Description
Concrete Type	Number	
Raw Material ID	Text	
Unit	Number	

Field Size	Double
Format	
Decimal Places	1
Input Mask	
Caption	
Default Value	0
Validation Rule	
Validation Text	
Required	No
Indexed	No
Smart Tags	

The number of digits that are displayed to the right of the decimal separator.

Figure J.6

Creating the *Bill of Material* Relation Structure

The *Bill of Material* relation is a composite (or intersection) relation, meaning that it has two fields (*Concrete Type* and *Raw Material ID*) making up the primary key. Highlight the appropriate fields and click on the **Key** button.

The type of number is **Double** since the *Units* can have a decimal place. The **Decimal Places** option is set to 1.

IMPLEMENTING THE *BILL OF MATERIAL* RELATION

The last relation that we'll cover here is the *Bill of Material* relation. As you can see in Figure J.6, we've entered all the field names and their types.

This relation is a little different from the rest because it has a composite primary key. A ***composite primary key*** consists of the primary key fields from the two intersecting relations. An ***intersection relation*** (sometimes called a ***composite relation***) is a relation you create to eliminate a many-to-many relationship. In *Extended learning Module C,* we created the *Bill of Material* relation to eliminate the many-to-many relationship that existed between *Concrete Type* and *Raw Material.* So, the *Bill of Material* relation has a primary key composed of two fields: the primary key *Concrete Type* from the *Concrete Type* relation and the primary key *Raw Material ID* from the *Raw Material* relation. The primary key of the *Bill of Material* relation is a composite primary key.

TEAM WORK

CREATE THE OTHER TABLES

You have seen how to create the *Concrete Type, Raw Material,* and *Bill of Material* relations. Use what you've learned so far to create the

- *Employee* relation
- *Order* relation
- *Supplier* relation
- *Truck* relation
- *Customer* relation

When you have created these five tables, create the relationships among them. The relationships we have not demonstrated in the text are the following:

Primary Key	In Relation	Foreign Key	In Relation
Supplier ID	*Supplier*	*Supplier ID*	*Raw Material*
Customer Number	*Customer*	*Customer Number*	*Order*
Concrete Type	*Concrete Type*	*Concrete Type*	*Order*
Truck Number	*Truck*	*Truck Number*	*Order*
Employee ID	*Employee*	*Driver ID*	*Order*

You will have to complete the second step of defining the relationships among the tables after reading the next section.

To identify two fields that aggregately create a primary key, we followed these steps:

1. Define the basic structure of the relation by entering the field names and their properties.
2. Move the pointer to the column immediately to the left of the first field of the composite primary key (the pointer will turn into an arrow pointing to the right).
3. Click on that row and don't unclick.
4. Drag the pointer so that second field in the composite primary key is also highlighted.
5. Unclick.
6. Click on the **Key** button.

The Key icon will appear next to each of the two fields identifying that together they make up a composite primary key.

To save this structure, we followed the same process as before. That is, we clicked on the **Disk** button (**Save As**), entered *Bill of Material* as the table name, clicked on **OK,** and closed the window.

Defining Relationships within the Solomon Enterprises Database

So far, we've created the structure of the relations for our database. In other words, we specified the names and types of the fields in each of the tables. As you can see, the process is very different from creating a word processing document or a workbook. We haven't yet entered any information into our database. We have one last structural issue to take care of, and that is to define how all the relations or files relate to each other.

Recall from our discussions in Chapter 3 and *Extended Learning Module C* that you can create relationships among the various tables by identifying foreign keys. A ***foreign key*** is a primary key of one file (relation) that appears in another file (relation). See Figure J.7 for the logical ties between primary and foreign keys.

Note that all the foreign keys have the same names as the primary keys in the original tables, except for *Employee ID,* which becomes *Driver ID* in the *Order* relation.

It's vitally important that you establish the relationships between primary and foreign keys. That way, the DBMS can enforce integrity constraints and disallow inconsistent information being entered. For example, when we specify that *Supplier ID* is a foreign key (from the *Supplier* relation) in the *Raw Material* relation, the DBMS will not allow us to enter a *Supplier ID* in the *Raw Material* relation that does not appear as a primary key in the *Supplier* relation. This makes business sense: You can't get raw material from a supplier who doesn't exist.

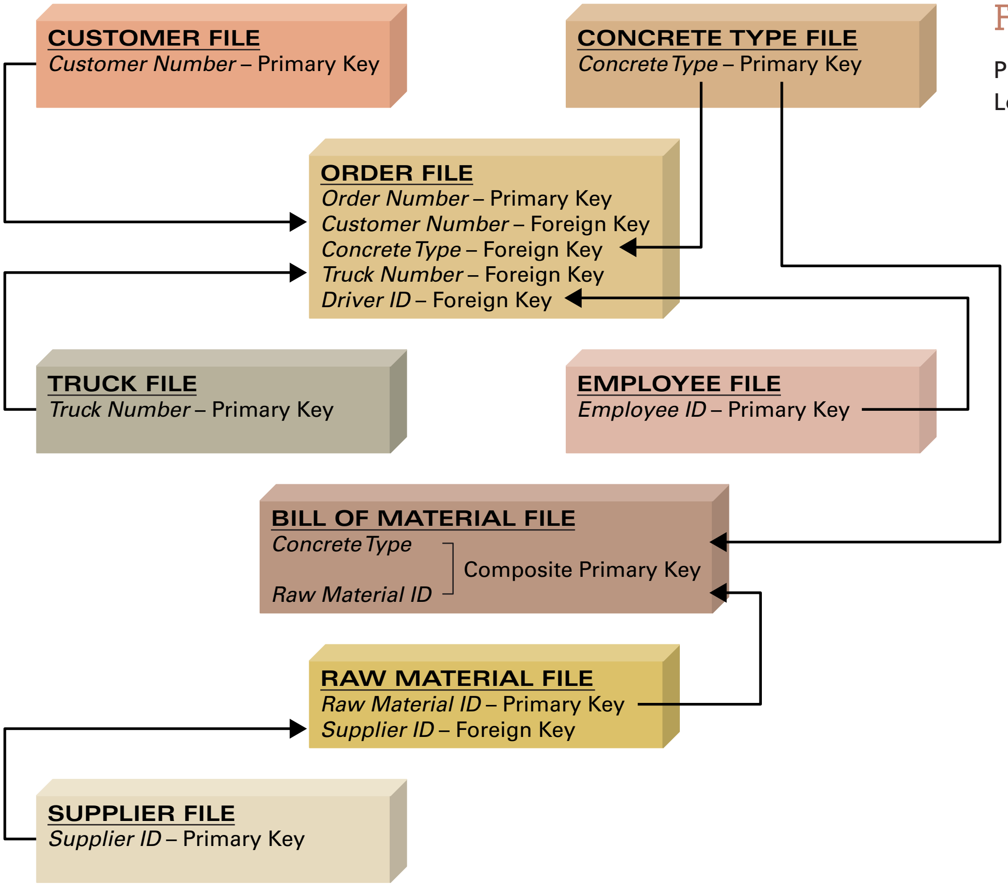

CUSTOMER FILE
Customer Number – Primary Key
CONCRETE TYPE FILE
Concrete Type – Primary Key
ORDER FILE
Order Number – Primary Key
Customer Number – Foreign Key
Concrete Type – Foreign Key
Truck Number – Foreign Key
Driver ID – Foreign Key
TRUCK FILE
Truck Number – Primary Key
EMPLOYEE FILE
Employee ID – Primary Key
BILL OF MATERIAL FILE
Concrete Type
Raw Material ID
Composite Primary Key
RAW MATERIAL FILE
Raw Material ID – Primary Key
Supplier ID – Foreign Key
SUPPLIER FILE
Supplier ID – Primary Key

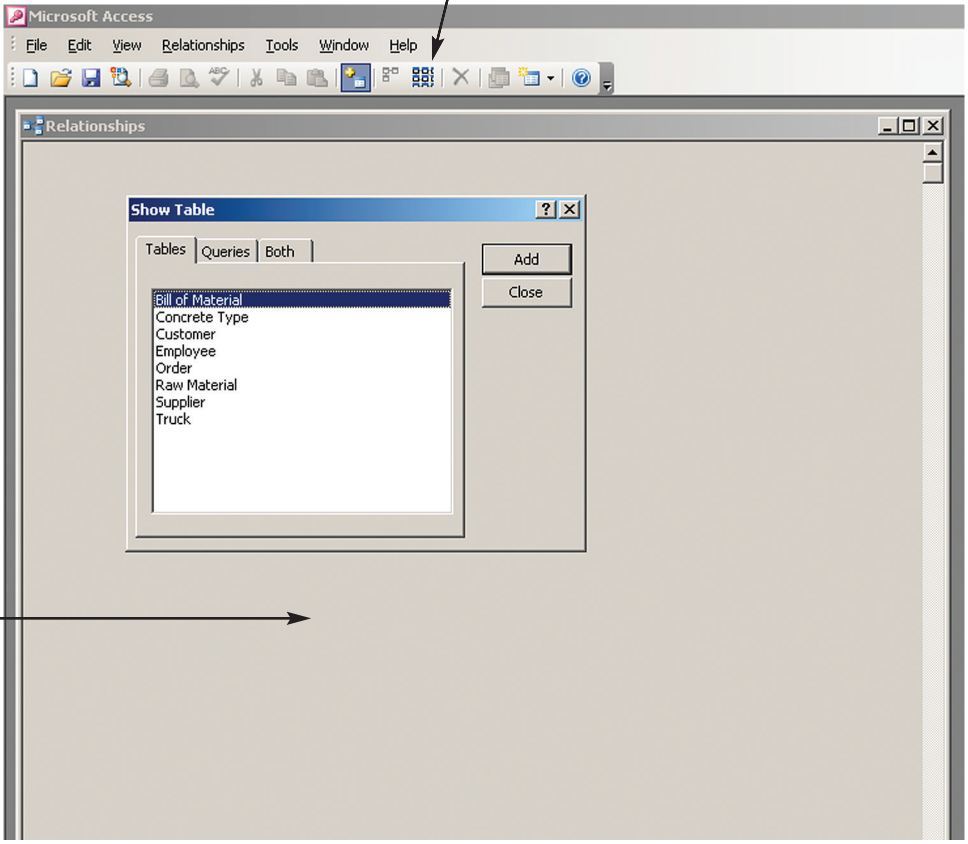
Figure J.7

Primary and Foreign Key Logical Ties

Figure J.8

The First Step in Defining Relationships in a Database

To define relationships in a database, you must first click on the **Relationships** button.



When you first start this process, the **Relationships** palette will be blank. As you highlight each relation and click on the **Add** button in the **Show Table** box, the relations will appear on the palette.

To create these relationships, which means telling Access which fields are foreign keys, you click on the **Relationship** button in the button bar. You'll then see the screen in Figure J.8. Notice that it lists all the relations in our database. When you first start this process, the palette in the background is blank. To identify the relationships, you must make each relation appear on the palette. To do this, simply highlight each relation name and click **Add.** When you have all the tables on the palette, click on the **Close** button to make the **Show Table** box disappear. When all the tables are on the palette, you're ready to identify how all the tables relate to each other.

To do this (and you must follow this process exactly), we clicked on and dragged each primary key to its respective foreign key counterpart and dropped it there. Once you drop the primary key onto its foreign key counterpart, you'll see the **Edit Relationships** box (see Figure J.9 on the next page). In that box, we clicked on **Enforce Referential Integrity** and then **Create.**

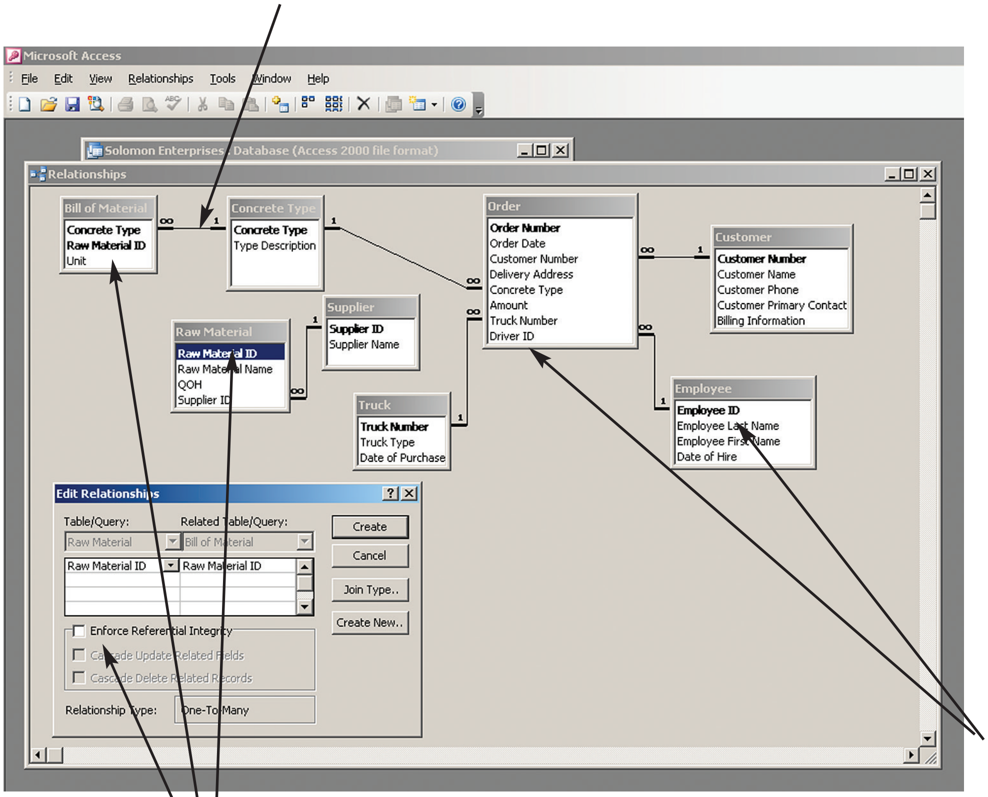
When you drag and drop a primary key onto a foreign key, Microsoft Access assumes that the relationship is one-to-many (1:M). That is, a primary key may appear many times as a foreign key, and a foreign key must appear once and only once as a primary key. If you perform the process in reverse (dragging and dropping the foreign key onto a primary key), the relationship will be reversed (M:1), which is not what you want.

You also need to turn on the enforcement of referential integrity. By doing so, your DBMS will make sure that when you enter a foreign key in a relation it matches one of

Figure J.9

Defining Relationships for the Solomon Enterprises Database

By dragging and dropping *Concrete Type* as a primary key in the *Concrete Type* relation onto *Concrete Type* in the *Bill of Material* relation as a foreign key, you create a 1:M (1-to-many) relationship between *Raw Material* and *Bill of Material.* The "many" is represented with the infinity symbol.



Table/Query: Raw Material

Related Table/Query: Bill of Material

Raw Material ID	Raw Material ID

☐ Enforce Referential Integrity

☐ Cascade Update Related Fields

☐ Cascade Delete Related Records

Relationship Type: One-To-Many

The primary key *Employee ID* in the *Employee* relation changes its name to *Driver ID* in the *Order* relation with both names referring to the same information.

This **Edit Relationship** box is the result of dragging and dropping *Raw Material ID* from the *Raw Material* relation onto the *Bill of Material* relation. As a general rule, you should turn on **Enforce Referential Integrity** to protect your database from inconsistent data. Then, click on the **Create** button and you get the connecting line.

the primary keys in the other relation. Figure J.9 shows the **Edit Relationships** box we got when we dragged and dropped *Raw Material ID* from the *Raw Material* relation onto *Raw Material ID* in the *Bill of Material* relation. We also clicked on **Enforce Referential Integrity** and **Create** and got the connecting lines you see between the other pairs of relations.

In Figure J.9, you can see many instances of a line connecting the primary key in one relationship to the foreign key in another. Let's look first at the relationship between the *Supplier* and *Raw Material* relations where Access shows a connecting line between *Supplier* and *Raw Material* relations with a 1 near the *Supplier* relation and an infinity symbol (∞) near the *Raw Material* relation. So, a *Supplier ID* can appear any number of times (infinity) in the *Raw Material* relation. Likewise, a *Supplier ID* that appears in the *Raw Material* relation can appear only one time in the *Supplier* relation. Also note that the foreign key can have a different name than the primary key, as, for example, *Employee ID* in the *Employee* relation becomes *Driver ID* in the *Order* relation. This is quite permissible, and how you name fields depends on your context.

Once we completed this process, we clicked on the **Disk** button to save the relationships and then closed the **Relationships** box.

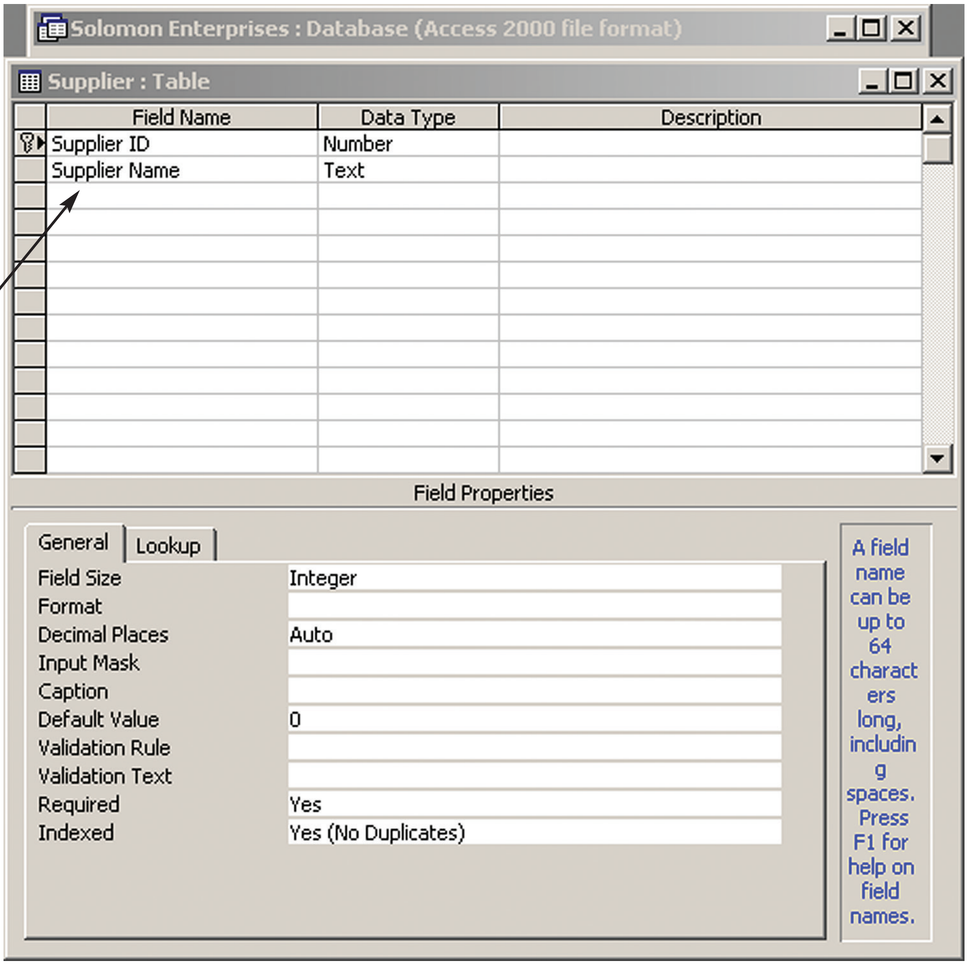
Entering Information into the Solomon Database

Finally, we're ready to begin entering information to populate the tables. We've defined the structure for each relation in the Solomon Enterprises database, and we've defined the relationships among the tables. See Figure J.10 for the design view of the *Supplier* relation that we created in the same manner as the *Concrete Type* relation.

To enter information, you simply highlight the desired relation (with the **Tables** tab selected) and click on **Open** (see Figure J.11). Does it matter which relation you start with when entering information? It does if you've chosen to enforce referential integrity when you created the relationships among the relations. We can't enter information into *Raw Material* relation yet because we need to put in a Supplier ID for each row and we've not yet entered the *Supplier ID* information into the *Supplier* relation. Therefore, if we try to put a *Supplier ID* into the *Raw Material* relation that is not in the *Supplier* relation, referential integrity will be violated and Access will display an error and not allow us to continue.

Figure J.10

Design View of the Structure of the *Supplier* Relation



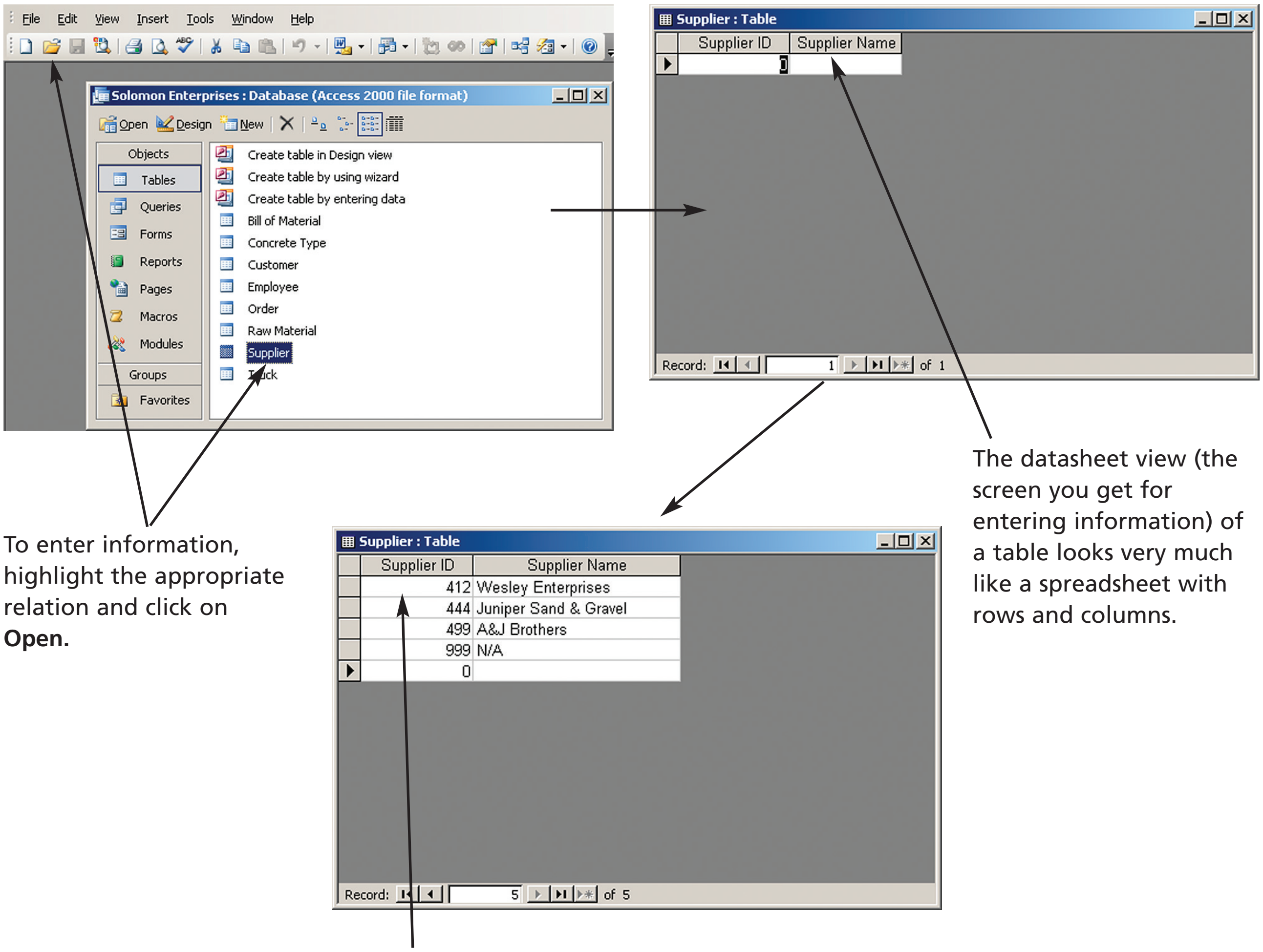
Field Name	Data Type	Description
Supplier ID	Number	
Supplier Name	Text	

General | Lookup

Field Size	Integer
Format	
Decimal Places	Auto
Input Mask	
Caption	
Default Value	0
Validation Rule	
Validation Text	
Required	Yes
Indexed	Yes (No Duplicates)

A field name can be up to 64 characters long, including spaces. Press F1 for help on field names.

The *Supplier* relation has two fields, *Supplier ID* and *Supplier Name.*



Supplier ID	Supplier Name
412	Wesley Enterprises
444	Juniper Sand & Gravel
499	A&J Brothers
999	N/A
0	

To enter information, highlight the appropriate relation and click on **Open.**

The datasheet view (the screen you get for entering information) of a table looks very much like a spreadsheet with rows and columns.

After you enter all the information and close the relation, the information is automatically saved.

Figure J.11

Entering information into the *Supplier* Relation

Before entering information into those relations that have foreign keys, you must first enter information into those tables with primary keys that show up as foreign keys in other tables. So we must complete the information entry for the *Concrete Type, Customer, Employee, Supplier,* and *Truck* relations before populating the *Raw Material, Bill of Material,* and *Order relations.*

In Figure J.11, you can see that we have entered some of the information into the *Supplier* relation. After typing in the appropriate information into each field, we used the **Tab** key to get to the next field or row. Once we entered all the information, and closed the table (relation) box, the information was automatically saved by Access. We can then move on to enter information into the other relations.

ON YOUR OWN

USE EXCEL TO CREATE AND POPULATE A TABLE

If you have your information in an Excel workbook, you can transfer the field names and information directly from the workbook to a database. Follow the steps below and create the *Truck* relation from the information in an Excel workbook.

1. Type the information into an Excel workbook and save it. We used **Truck.xls** as the name of the file.
2. Open the Solomon Enterprises database and click on **File** and then **Get External Data** and then **Import.**
3. Enter the folder and name of the Excel file and click on the **Import** button.
4. Choose the worksheet where the information is stored. We had the information in **Sheet1** which is Access's first choice, so click on **Next>**.
5. Access assumes that the top row of information contains the field names. We accepted that as the default since that is the case, so click on **Next>**.
6. Accept the default, which is to put the information into a new table, so click on **Next>**.
7. Change the **Indexed** options for *Truck Number* to **Yes (no duplicates)**.
8. Click **Choose my own primary key** and then choose **Truck Number** primary key. Click **Next>**.
9. Enter "Truck Number" as the name of the table. Click **Finish.**

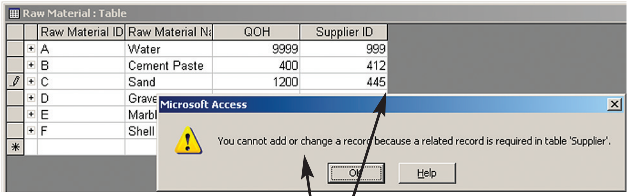
When you open up the *Truck* relation you'll see the finished table complete with information.

Next we'll populate the *Raw Material* relation, which has *Supplier ID* as a foreign key. In the screen on the left of Figure J.12 you can see that we've already entered some of the information; however, as the error message shows, we put in an incorrect *Supplier ID* (445 instead of 444). When we established the relationship between the *Raw Material* and *Supplier* relations, we chose the **Enforce Referential Integrity** option, so Access won't allow us to enter a *Supplier ID* in the *Raw Material* Relation that doesn't already exist in the *Supplier* relation. That's why we got the dialog box informing us of the problem. If we click on **OK,** Access will give us a chance to fix the error. If we don't fix the problem and we try to exit this window (see the screen on the right in Figure J.12), Access will alert us with a new box telling us that the incorrect information will not be saved (although the good information will).

Now that we've created the structure of the relations and entered the necessary information, it's time to move on to creating queries and reports.

Figure J.12

Encountering an Integrity Error While Entering Information



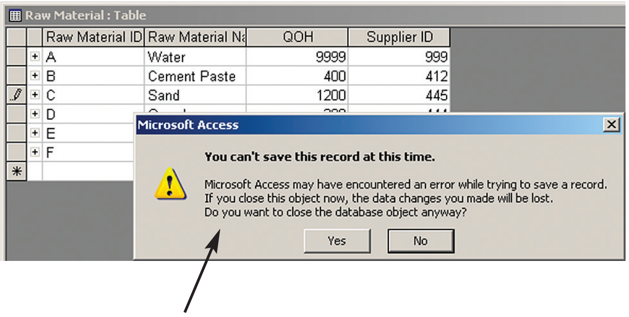
Raw Material : Table

Raw Material ID	Raw Material Na	QOH	Supplier ID
A	Water	9999	999
B	Cement Paste	400	412
C	Sand	1200	445
D	Grave		
E	Marbl		
F	Shell		

Microsoft Access

You cannot add or change a record because a related record is required in table 'Supplier'.

Because we entered a *Supplier ID* (445) that doesn't exist in the *Supplier* relation, Access will not allow us to continue.



Raw Material : Table

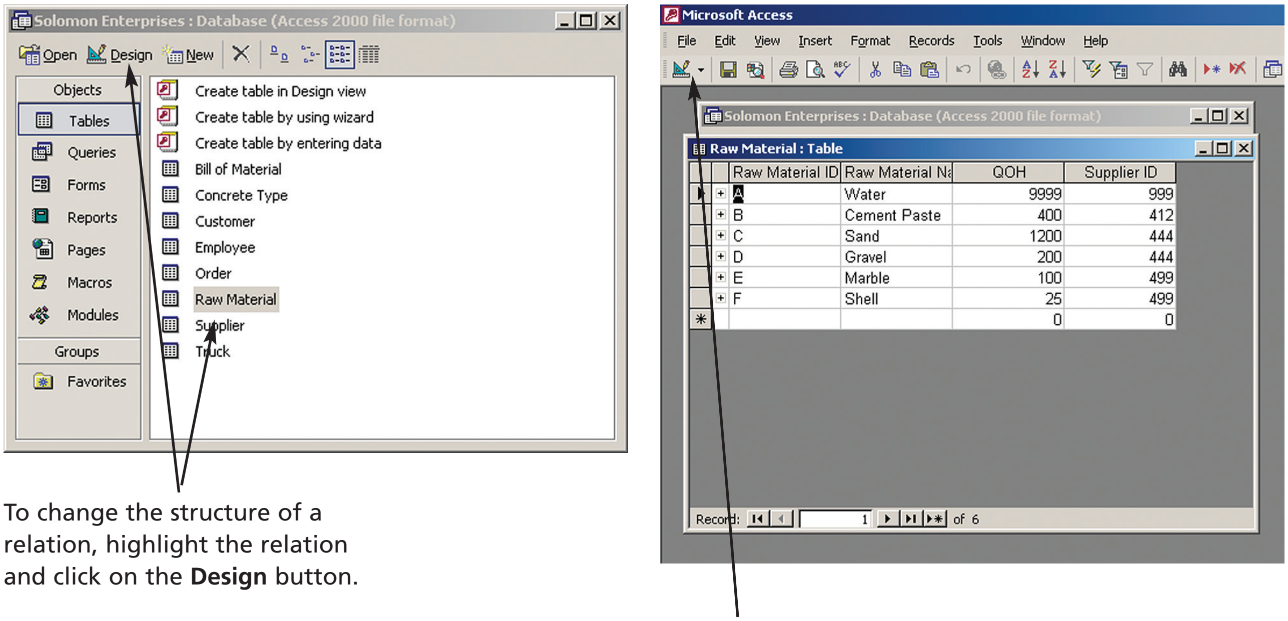
Raw Material ID	Raw Material Na	QOH	Supplier ID
A	Water	9999	999
B	Cement Paste	400	412
C	Sand	1200	445
D			
E			
F			

Microsoft Access

You can't save this record at this time.

Microsoft Access may have encountered an error while trying to save a record.
If you close this object now, the data changes you made will be lost.
Do you want to close the database object anyway?

If you try to close the information entry window, Access will allow you to change the information or save the good information without the bad.



Raw Material ID	Raw Material Na	QOH	Supplier ID
A	Water	9999	999
B	Cement Paste	400	412
C	Sand	1200	444
D	Gravel	200	444
E	Marble	100	499
F	Shell	25	499
		0	0

To change the structure of a relation, highlight the relation and click on the **Design** button.

If you've got the datasheet view (the relation with the information in it), click on the **Design** button.

Figure J.13

Changing the Structure of a Relation

CHANGING THE STRUCTURE OF INFORMATION IN RELATIONS

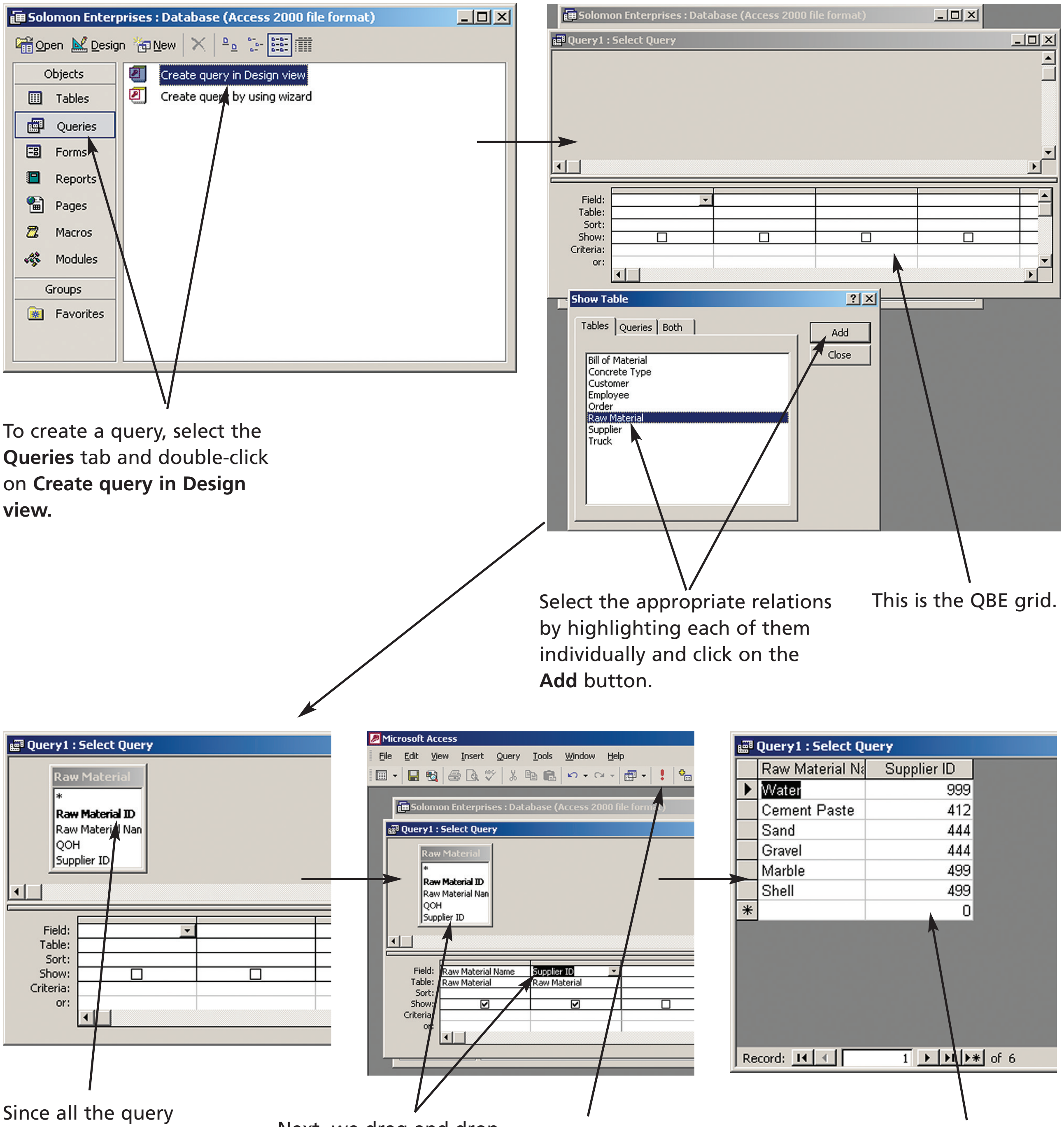
If you want to change the information in any of the relations or the structure of any of the relations, you can do so quite easily. However, you need to be careful about deleting fields that you used to establish relationships between pairs of tables.

The simpler task is to add, change, or delete records in a relation. To add a record, you simply open the relation and add the new information at the bottom. To change information in a field, click on the appropriate field and make the change. To delete a record, highlight the record (row) you want deleted and press the **Delete** key on the keyboard.

To change the structure of a relation, you'll need to open the design view of the relation. You can do so in either one of two ways. If you have the screen on the left in Figure J.13, you would highlight the table whose structure you want to change and click on the triangle in the button bar, and you'll see the data dictionary for that relation. If you have the datasheet view (on the right in Figure J.13) open, you can immediately click on the triangle button, which then becomes the datasheet button allowing you to flip back again.

Creating a Simple Query Using One Relation

The easiest way to create a query is to use a query-by-example tool. A ***query-by-example (QBE) tool*** helps you graphically design the answer to a question. Suppose, for example, that we wanted to see a list of all raw materials, by name, and the ID of the associated supplier. All that information is located in the *Raw Material* relation, making it a relatively simple query requiring the use of only a single relation. Realize, of course, that our Solomon database is small and the query doesn't make much sense since you could easily open the *Raw Material* relation and see the information you want. However, the objective is to demonstrate the use of a QBE tool. In the next section we'll create a more complicated query. But first let's look at a simple query.



Query1 : Select Query (Design view):

Field:	Raw Material Name	Supplier ID
Table:	Raw Material	Raw Material
Show:	☑	☑

Query1 : Select Query (result):

Raw Material Na	Supplier ID
Water	999
Cement Paste	412
Sand	444
Gravel	444
Marble	499
Shell	499
	0

To create a query, select the **Queries** tab and double-click on **Create query in Design view.**

Select the appropriate relations by highlighting each of them individually and click on the **Add** button.

This is the QBE grid.

Since all the query information we need is in the *Raw Material* relation, it's the only relation we need.

Next, we drag and drop *Raw Material Name* and *Supplier ID* from the *Raw Material* relation into the QBE grid.

When all fields are in place, we had to click on the exclamation point (**Run**) to see the result of the query.

After clicking on the exclamation point icon (**Run**), Access provides us with the result of our query.

Figure J.14

Creating a Simple Query Using One Relation

To create a simple query using only the *Raw Material* relation, perform the following steps (see Figure J.14 on the previous page):

1. Select the **Queries** tab.
2. Double-click on **Create query in Design view** (you can also create a query using a wizard; we'll let you explore this feature on your own).
3. In the **Show Table** dialog box, select the appropriate relation name, click on **Add,** and then close the **Show Table** dialog box.
4. Drag and drop the fields that you want in the query result into the QBE grid.
5. Click on the exclamation point (**Run**) in the button bar.

As you can see in Figure J.14, we followed that process by selecting the *Raw Material* relation, and dragging and dropping *Raw Material Name* and *Supplier ID* into the QBE grid. Once we clicked on the exclamation point button, Access returned a list of raw materials along with the ID of the supplier for each item.

If we wanted to be able to use this query at a later time, we could save the query, giving it a unique name such as *Raw Materials and Suppliers.* Then, the next time we need that information, we could simply open up that query in the datasheet view (the view that shows the information) rather than having to start from scratch creating the query.

SIMPLE QUERY WITH A CONDITION (CONDITIONAL QUERY)

Creating a query with only one relation is pretty simple. So, let's add another requirement to our query. Let's say we wanted to see which raw materials from which suppliers we have in quantities of 400 or more units. This is a conditional query because it will return a result based on some condition. This conditional query requires an extra couple of steps in the process we just outlined.

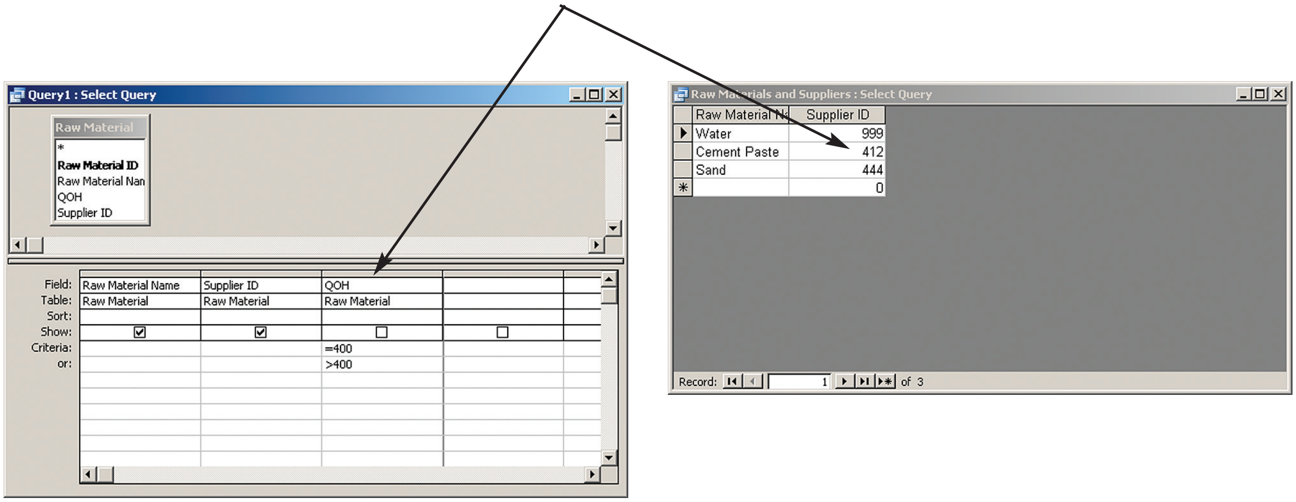
In Figure J.15, you can see that we again only selected the *Raw Material* relation since it contains all the information we need. However, this time we also dragged and dropped *QOH* into the QBE grid. Within the QBE grid, we provided the two important items of information to the query.

The first was to unselect the **Show** parameter. By doing so, we tell Access that we want to use *QOH* as part of the query, but that we don't want it to show in the query result. The second is to enter **"=400"** (without the quotation marks) in the **Criteria**

Figure J.15

A Conditional Query Using One Relation

To create a conditional query, we add *QOH* from the *Raw Material* relation and specify "=400" or ">400" in the **Criteria** fields.



Field:	Raw Material Name	Supplier ID	QOH	
Table:	Raw Material	Raw Material	Raw Material	
Sort:				
Show:	☑	☑	□	□
Criteria:			=400	
or:			>400	

Raw Material N	Supplier ID
Water	999
Cement Paste	412
Sand	444
	0

parameter and then **">400"** (again, without quotes) right below. In doing this, we're telling Access that we want to see only those raw materials that we have in quantities equal to 400 (first **Criteria** line) or greater than 400 (the **or** line). As you can see in the bottom screen of Figure J.15, Access provides information for only three raw materials, all of which Solomon has quantities on hand of 400 or more units. Also note that the *QOH* field does not show.

If you were looking for the suppliers of *Gravel,* you would put "Gravel" (with or without the quotation marks) into the **Criteria** box associated with *Raw Material Name,* but you must be sure to spell it correctly. The capitalization doesn't matter but the right letters (and digits and spaces, if applicable) in the right places do.

Creating an Advanced Query Using More than One Relation

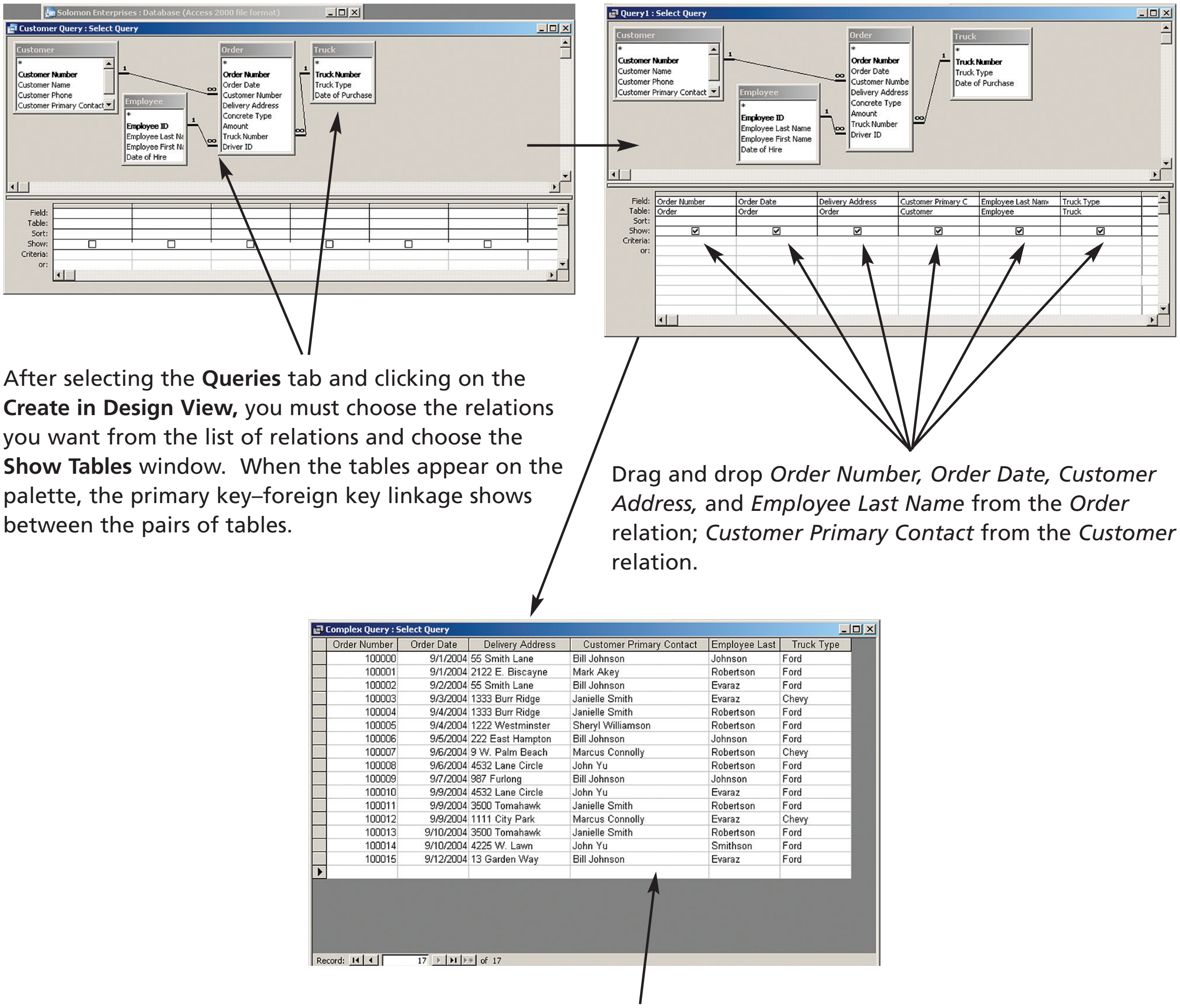
What about queries that require information residing in two or perhaps several different relations? Access can handle those too. We just have to tell it where to get the information and make sure that the tables that need to reference each other have relationships already defined. For example, say we wanted to put the *Supplier Name* instead of the *Supplier ID* into a query that shows *Raw Material ID* and *Raw Material Name* and where Solomon Enterprise gets these items. We would need the *Raw Material* relation and also the *Supplier* relation and would want Access to take the *Supplier ID* from the *Raw Material* relation and match it to the *Supplier ID* in the *Supplier* relation to find the correct *Supplier Name.*

The above query would involve two tables. But Access can handle information from many tables in one query, so let's look at a more complex example. Say the accounts receivable manager at Solomon is trying to sort out a problem with order numbers. So, the manager wants to know the following information:

- All order numbers
- Date of orders
- Where the goods were delivered
- The contact person at the delivery destination
- Which truck was involved in each delivery
- Who drove the delivery truck

We need to start by determining which relations to use. The fields we need are in the *Order, Customer, Employee,* and *Truck* relations. So these are the ones we'll use.

Table	Fields
Order	*Order Number*
	Date of order
	Delivery Address
Customer	*Contact Person*
Employee	*Employee Last Name*
Truck	*Type of truck*



Field:	Order Number	Order Date	Delivery Address	Customer Primary C	Employee Last Nam	Truck Type
Table:	Order	Order	Order	Customer	Employee	Truck
Sort:						
Show:	☑	☑	☑	☑	☑	☑
Criteria:						
or:						

After selecting the **Queries** tab and clicking on the **Create in Design View,** you must choose the relations you want from the list of relations and choose the **Show Tables** window. When the tables appear on the palette, the primary key–foreign key linkage shows between the pairs of tables.

Drag and drop *Order Number, Order Date, Customer Address,* and *Employee Last Name* from the *Order* relation; *Customer Primary Contact* from the *Customer* relation.

Complex Query : Select Query

Order Number	Order Date	Delivery Address	Customer Primary Contact	Employee Last	Truck Type
100000	9/1/2004	55 Smith Lane	Bill Johnson	Johnson	Ford
100001	9/1/2004	2122 E. Biscayne	Mark Akey	Robertson	Ford
100002	9/2/2004	55 Smith Lane	Bill Johnson	Evaraz	Ford
100003	9/3/2004	1333 Burr Ridge	Janielle Smith	Evaraz	Chevy
100004	9/4/2004	1333 Burr Ridge	Janielle Smith	Robertson	Ford
100005	9/4/2004	1222 Westminster	Sheryl Williamson	Robertson	Ford
100006	9/5/2004	222 East Hampton	Bill Johnson	Johnson	Ford
100007	9/6/2004	9 W. Palm Beach	Marcus Connolly	Robertson	Chevy
100008	9/6/2004	4532 Lane Circle	John Yu	Robertson	Ford
100009	9/7/2004	987 Furlong	Bill Johnson	Johnson	Ford
100010	9/9/2004	4532 Lane Circle	John Yu	Evaraz	Ford
100011	9/9/2004	3500 Tomahawk	Janielle Smith	Robertson	Ford
100012	9/9/2004	1111 City Park	Marcus Connolly	Evaraz	Chevy
100013	9/10/2004	3500 Tomahawk	Janielle Smith	Robertson	Ford
100014	9/10/2004	4225 W. Lawn	John Yu	Smithson	Ford
100015	9/12/2004	13 Garden Way	Bill Johnson	Evaraz	Ford

Record: 17 of 17

Click on the exclamation point (**Run**) button to see the results of the query.

Figure J.16

Creating an Advanced Query Using More than One Relation

We'll follow the same set of steps as we did when creating the simple query using one relation and make some modifications along the way to generate a more complex report (see Figure J.16).

1. Select the **Queries** tab.
2. Double-click on **Create query in Design view.**
3. In the **Show Table** dialog box, select the appropriate relation names *(Customer, Order, Employee,* and *Truck),* clicking on **Add** each time, and then close the **Show Table** dialog box. Here, as the tables that are linked by primary and foreign keys appear in the palette, they are joined by lines with the 1 beside the table that has the primary key and the infinity sign (∞) near the table that has the foreign key. These symbols are showing you the 1:M relationship.
4. Drag and drop the fields that you want from the appropriate relation into the QBE grid in the order that you want.
5. Click on the exclamation point (**Run**) in the button bar.

TEAM WORK

FIND ALL THE ORDERS FOR PREMIER SHELL CONCRETE

Create a query that shows the

- Order number
- Date of order
- Customer name
- Amount

for all orders of Premier Shell concrete. One way to do this is to create a query with two tables: *Order* and *Customer.* The fields would be *Order Number, Order Date,* and *Concrete Type* from the *Order* relation. The *Customer Name* will come from the *Customer* relation. The *Concrete Type* field will have the condition that ensures that only those records where *Concrete Type* equals 5 show up. Don't forget to unselect the **Show** feature of *Concrete Type* so that that field does not appear on the query.

And that's it. It's not significantly more difficult than using just one relation. The critical part in using multiple tables in a query is to make sure that the tables are linked correctly. If they are not, your query result will be incorrect, perhaps having multiple entries for each instance of a field entry. We took care of this stipulation when we established the relationships among the tables before we entered the information into them.

If you want to put conditions on any of the fields in this multi-table query, you simply enter those conditions in the **Criteria** line for the appropriate field as we did in the simple conditional table.

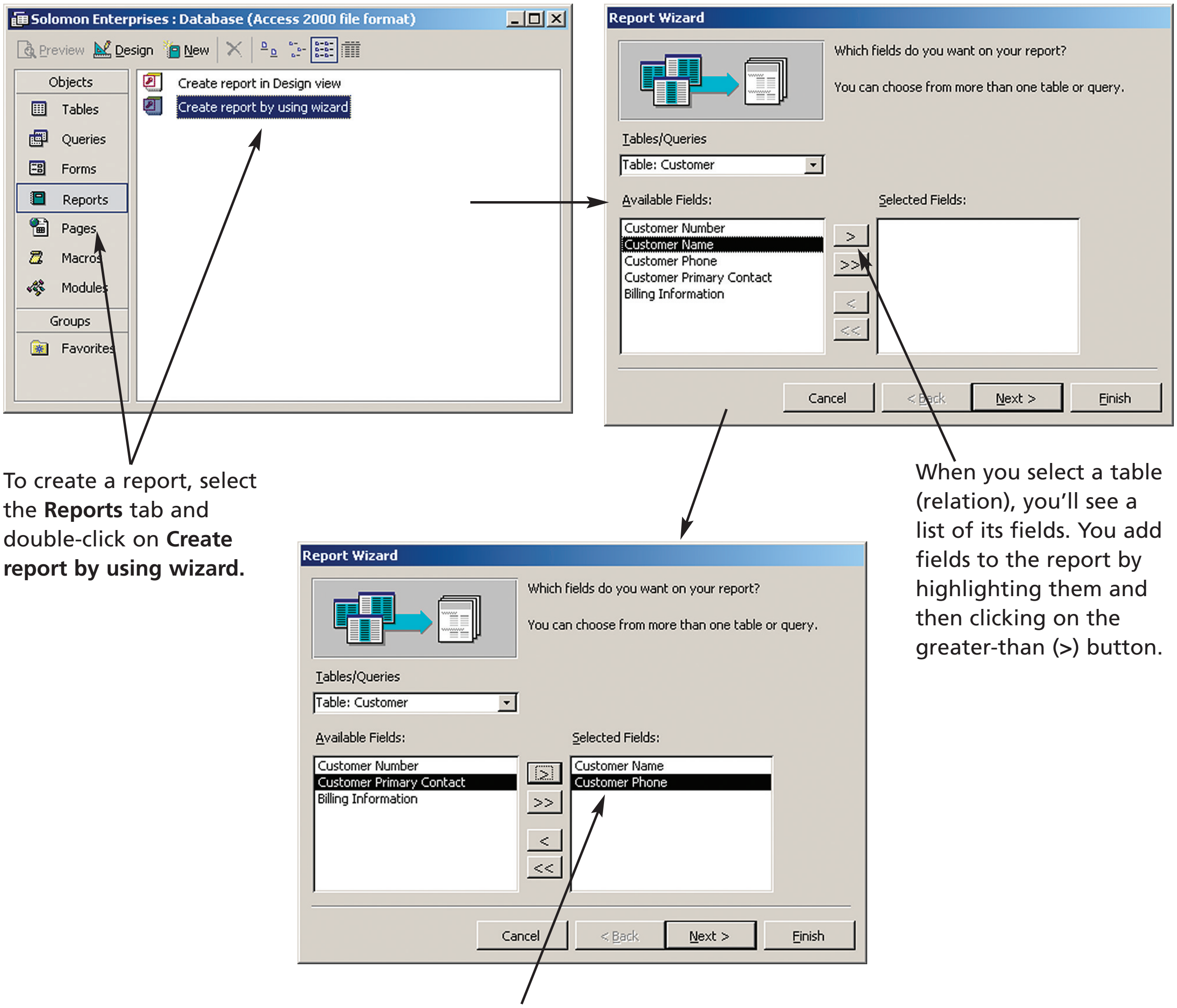
Generating a Simple Report

Now that you know how to create a database and construct queries, let's see about making the output look better. The fundamental difference between tables (or queries) and reports is that a report is designed for human consumption; that is, it has the information arranged so that it looks nice and is easy to read. The output includes headings and footings, and usually includes page numbers and the date of the report.

We'll start with a fairly simple example. As was the case with tables and queries, you can create a report using the design view or the appropriate wizard. We're going to use the report wizard first to create a simple report and then a more complex one that we will modify using the design view.

Let's say we want a report showing all our customers' names and phone numbers. This involves only one table, since both of the fields we want are in the *Customer* relation. To create this report, we followed the steps below (see Figures J.17 and J.18 on pages 23 and 24):

1. Select the **Reports** tab.
2. Double-click on **Create report by using wizard.**
3. Choose tables and/or queries: This screen lets you choose which table or query you want to show in your report. We selected **Table:Customer** in the **Tables/Queries** box.
4. Choose fields: In this screen you can choose the fields you want from the tables and/or queries you chose in the previous step. So, under Available Fields, we selected *Customer Name* and clicked on the greater-than sign (>) to the right. Next, we selected *Customer Phone* and clicked once more on the greater-than sign (>). These actions transferred the two fields to the other side, meaning that



Which fields do you want on your report?

You can choose from more than one table or query.

Tables/Queries: Table: Customer

Available Fields: Customer Number, Customer Name, Customer Phone, Customer Primary Contact, Billing Information

Selected Fields:

Report Wizard

Which fields do you want on your report?

You can choose from more than one table or query.

Tables/Queries: Table: Customer

Available Fields: Customer Number, Customer Primary Contact, Billing Information

Selected Fields: Customer Name, Customer Phone

To create a report, select the **Reports** tab and double-click on **Create report by using wizard.**

When you select a table (relation), you'll see a list of its fields. You add fields to the report by highlighting them and then clicking on the greater-than (>) button.

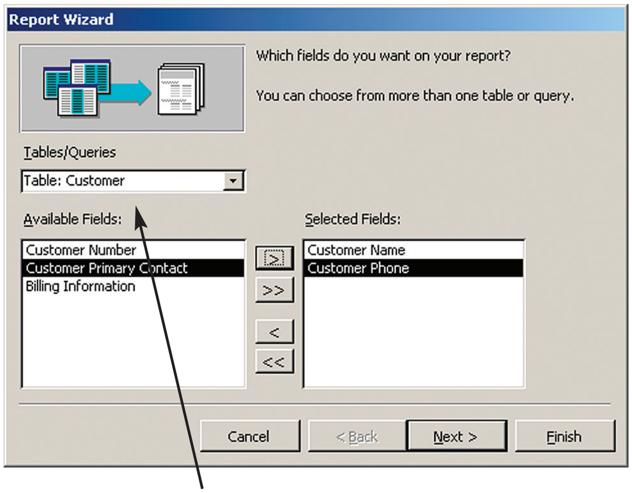
We selected *Customer Name* and *Customer Phone* to go onto the report.

Figure J.17

Creating a Report Using One Relation

we will see these on our report. You can choose all fields in the table by clicking on the double greater-than sign (>>).

5. Grouping: This screen allows you to specify grouping of information. Here again we accepted default by clicking on **Next>**.
6. Sorting: This screen allows you to specify sorting of information. We chose not to sort so we clicked on **Next>**.
7. Layout and orientation of the report: This screen allows you to select layout and page orientation. Again, we accepted the default and clicked on **Next>**.
8. Style of the report: This screen allows you to choose from among predefined report styles. Once more we accepted the default and clicked on **Next>**.
9. Report header: This screen allows us to enter a title for the report. So, we entered "Customer Report" (without the quotation marks) in the title box and clicked on **Finish.**



Which fields do you want on your report?

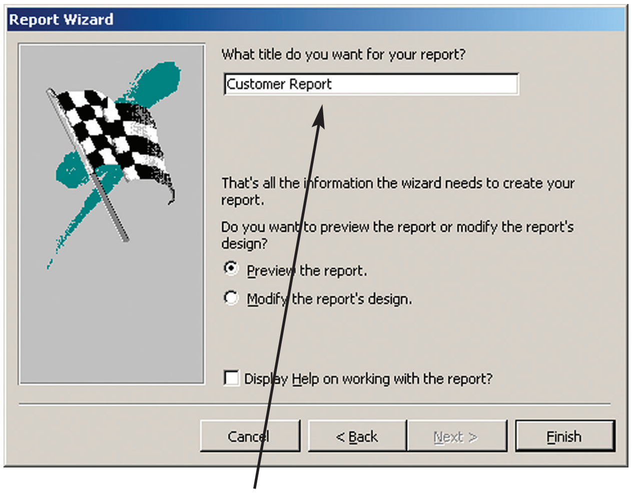
You can choose from more than one table or query.

Tables/Queries: Table: Customer

Available Fields:
- Customer Number
- Customer Primary Contact
- Billing Information

Selected Fields:
- Customer Name
- Customer Phone

Once you select a table, you will see the list of fields in it. You add fields to a report by highlighting them and then clicking on the greater-than sign (>) button.



What title do you want for your report?

Customer Report

That's all the information the wizard needs to create your report.

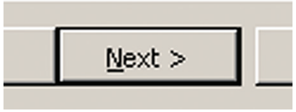
Do you want to preview the report or modify the report's design?

● Preview the report.

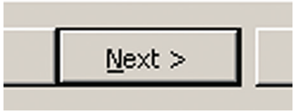
○ Modify the report's design.

□ Display Help on working with the report?

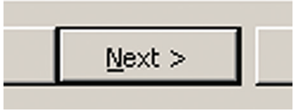
We typed **Customer Report** as the name of the report.



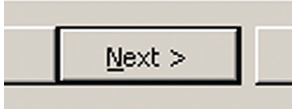
Accept default ordering.



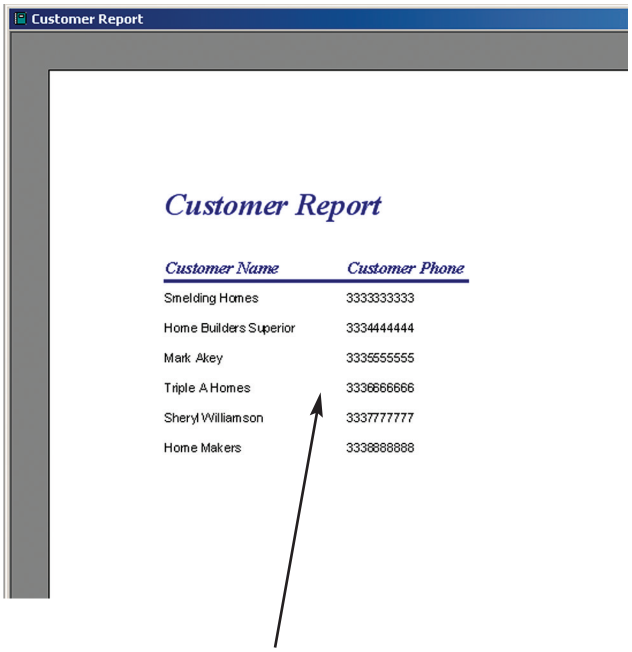
Accept default sorting.



Accept default layout.



Accept default style.



Customer Report

Customer Name	*Customer Phone*
Smelding Homes	3333333333
Home Builders Superior	3334444444
Mark Akey	3335555555
Triple A Homes	3336666666
Sheryl Williamson	3337777777
Home Makers	3338888888

The report shows all customers and their phone numbers.

Figure J.18

Creating a Report Using One Relation, *continued*

What you'll then see is a screen representation of your report. It includes only the two fields of information we requested in the style we chose in a more polished form than you get by simply printing a query with the same two fields.

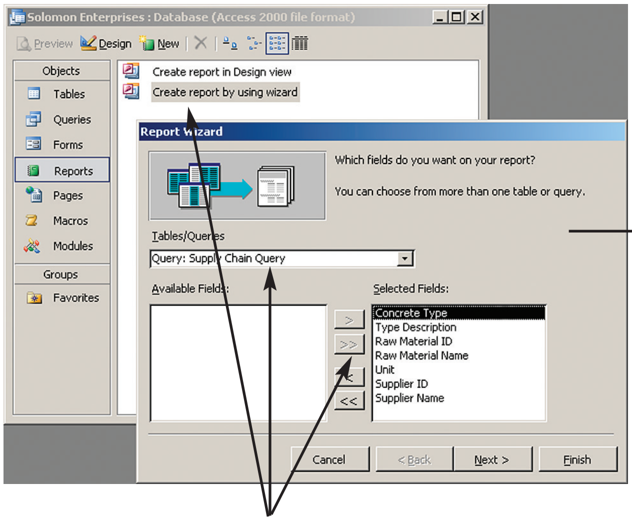
Generating a Report with Grouping, Sorting, and Totals

Now that we know the basic procedure, let's see how we can generate a more complex report. Look again at the *Supply Chain Management* report from *Extended Learning Module C* on page 162. It has grouping of the raw materials that go into each type of concrete along with totals for the number of units for each type of concrete.

We'll use the same process as we did with the *Customer Report,* but we'll do more than just accept default options as we proceed through the steps. The first thing to note about our *Supply Chain Management* report is that it requires more than one table. In fact, from the *Concrete* relation, we need *Concrete Type* and *Type Description;* from the *Raw Material* relation we need *Raw Material ID* and *Raw Material Name;* from the *Bill of Material* relation we need the *Unit* field; and from the *Supplier* relation we need *Supplier ID* and *Supplier Name.* We could choose each table in turn and then select the fields from each one that we need. Instead, we'll first construct a query using all these relations then transform the result of the query into a report. The process is the same one we used in the *Creating an Advanced Query Using More than One Relation* section on page 200. We named the new query *Supply Chain Query*.

Once we have our query ready, we're ready to generate the report. Following are the steps (see Figures J.19 and J.20 on pages 26 and 27).

1. Select the **Reports** tab.
2. Double-click on **Create report by using wizard.**
3. <u>Choose tables and/or queries:</u> In the **Tables/Queries** box, select **Query: Supply Chain Query.**
4. <u>Choose fields:</u> Under **Available Fields,** select all fields in the query by clicking on the double greater-than sign (>>).
5. <u>Top-level grouping:</u> The next screen allows us to choose the ordering of information for presentation, also known as "grouping information." You'll notice in Figure J.19, in the top right-hand screen, that Access has already preselected a grouping for us. As it happens, Access has done the grouping we want (by *Concrete Type* and *Type Description*), so we accepted the default and clicked on **Next>**.
6. <u>Further grouping:</u> The next screen lets us specify groups within the top grouping of *Concrete Type.* Since we don't want any subgrouping we clicked on **Next>**.
7. <u>Sorting:</u> Next we have a chance to sort our information. Here we'll specify that the raw material information appear in alphabetical order based on *Raw Material ID.* Since water is the least significant of the raw materials, in the sense that it's freely available, we chose to put it last in the list. Therefore, we clicked in box 1, and used the arrow button to bring *Raw Material ID* into the box. Then we clicked on the **Ascending** key to change it to **Descending.**
8. <u>Totaling:</u> The sorting screen that we saw above also has a **Summary Options** button. We clicked on that and chose to **Sum** *Units* and to show **Detail and Summary.** Then we clicked on **OK** and **Next>**.



Solomon Enterprises : Database (Access 2000 file format)

Create report in Design view

Create report by using wizard

Report Wizard

Which fields do you want on your report?

You can choose from more than one table or query.

Tables/Queries

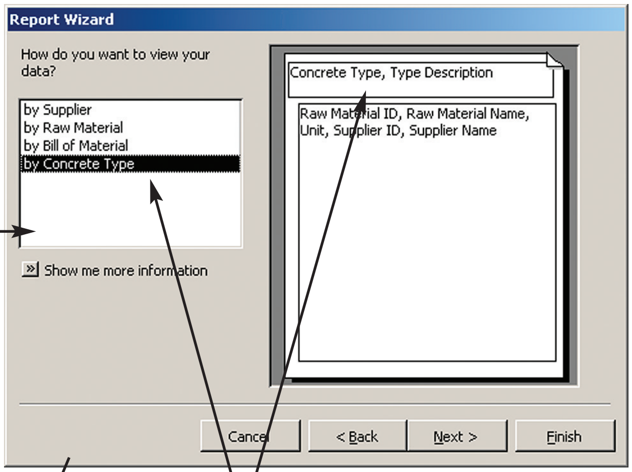
Query: Supply Chain Query

Available Fields:

Selected Fields:

Concrete Type
Type Description
Raw Material ID
Raw Material Name
Unit
Supplier ID
Supplier Name

We selected the **Create report by using wizard.** Then we chose the **Query: Supply Chain Query.** Next we selected all fields to appear on the report by clicking on the double greater-than sign (>>).



Report Wizard

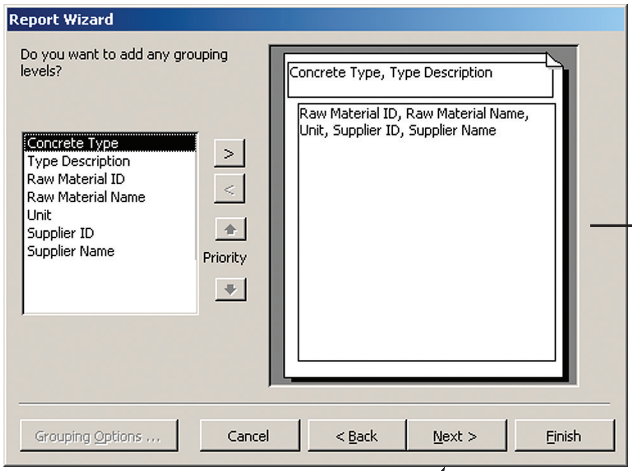
How do you want to view your data?

by Supplier
by Raw Material
by Bill of Material
by Concrete Type

Concrete Type, Type Description

Raw Material ID, Raw Material Name, Unit, Supplier ID, Supplier Name

We chose the default top-level grouping of *Concrete ID* and *Type Description.*



Report Wizard

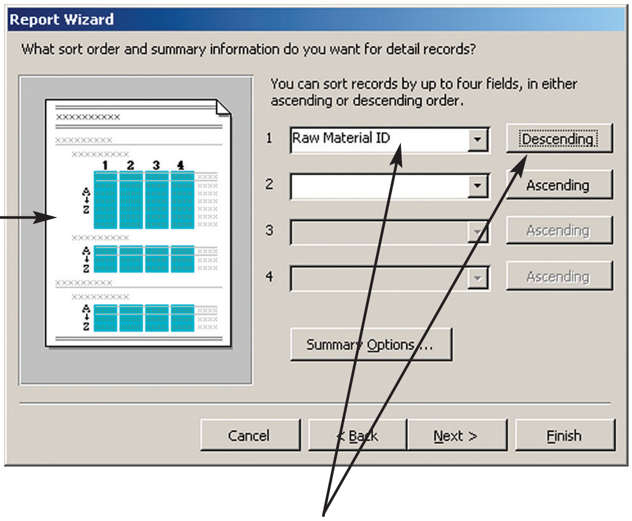
Do you want to add any grouping levels?

Concrete Type
Type Description
Raw Material ID
Raw Material Name
Unit
Supplier ID
Supplier Name

Concrete Type, Type Description

Raw Material ID, Raw Material Name, Unit, Supplier ID, Supplier Name

We don't want any further grouping of information, so we click **Next>.**



Report Wizard

What sort order and summary information do you want for detail records?

You can sort records by up to four fields, in either ascending or descending order.

1 Raw Material ID Descending

2 Ascending

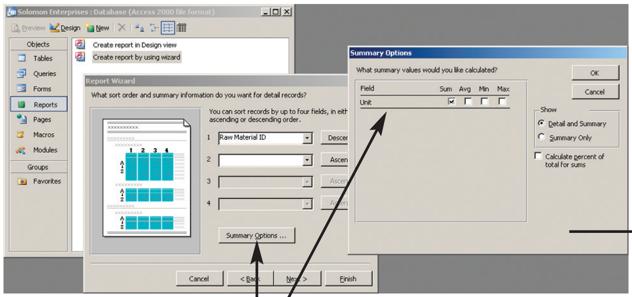
3 Ascending

4 Ascending

We chose to sort the information within the group on *Raw Material ID* in descending order so that water (with *Raw Material ID* = C) would always be last in the list.

Figure J.19

Creating a Report with Grouping, Sorting, and Totals



Solomon Enterprises : Database (Access 2000 file format)

Create report in Design view

Create report by using wizard

Report Wizard

What sort order and summary information do you want for detail records?

You can sort records by up to four fields, in eith ascending or descending order.

1 Raw Material ID

Summary Options

What summary values would you like calculated?

Field	Sum	Avg	Min	Max
Unit	☑	□	□	□

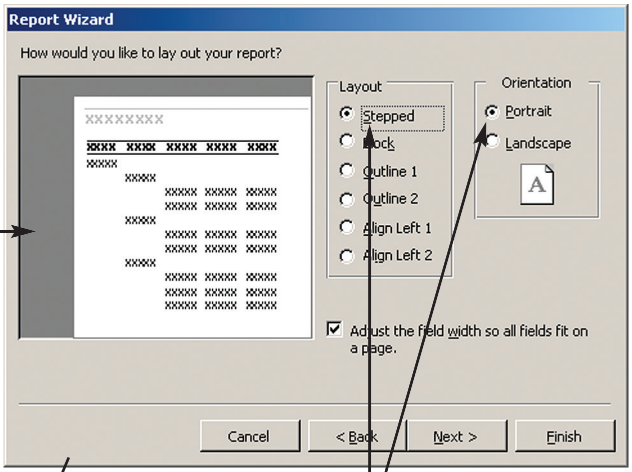
Show

● Detail and Summary

○ Summary Only

□ Calculate percent of total for sums

We then click on **Summary Options** so that we can total the units of raw material that each type of concrete would use. When we check the **Sum** box, we click **OK.**



Report Wizard

How would you like to lay out your report?

Layout

● Stepped

○ Block

○ Outline 1

○ Outline 2

○ Align Left 1

○ Align Left 2

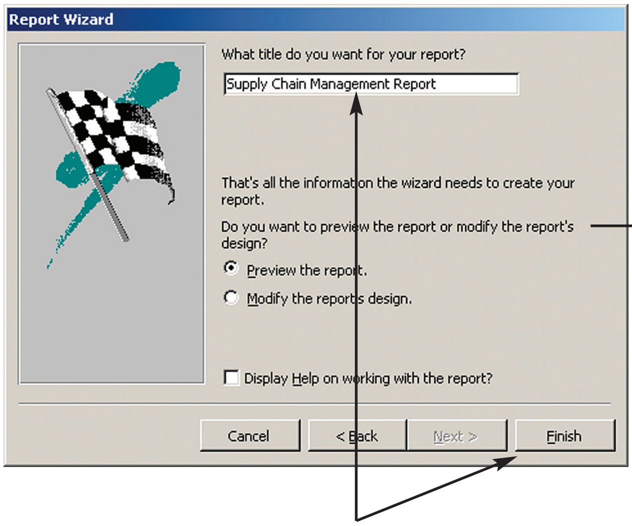
Orientation

● Portrait

○ Landscape

☑ Adjust the field width so all fields fit on a page.

We choose the type of report we want. **Stepped** is the **Layout** and **Portrait** is the page **Orientation** of the *Supply Chain Management Report* in *Module C.* Then, in the next dialog box, we chose **Corporate** as the style.



Report Wizard

What title do you want for your report?

Supply Chain Management Report

That's all the information the wizard needs to create your report.

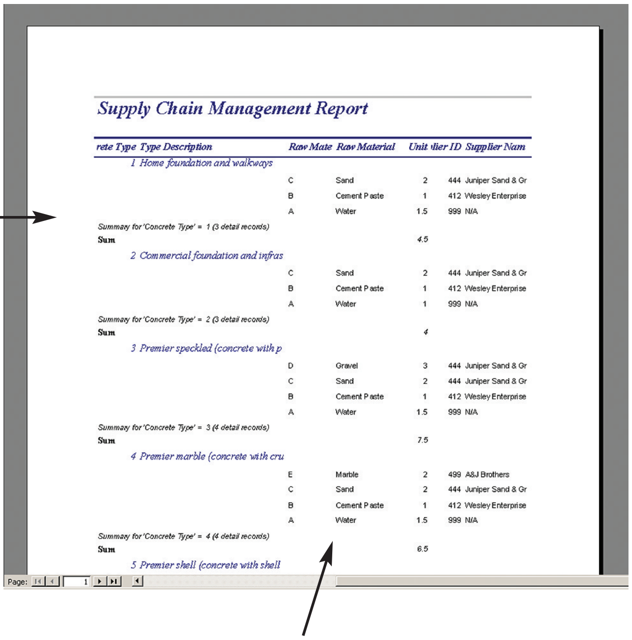
Do you want to preview the report or modify the report's design?

● Preview the report.

○ Modify the report's design.

□ Display Help on working with the report?

Type in **Supply Chain Management Report** as the title of the report. Then click on **Finish** to see what the report looks like.



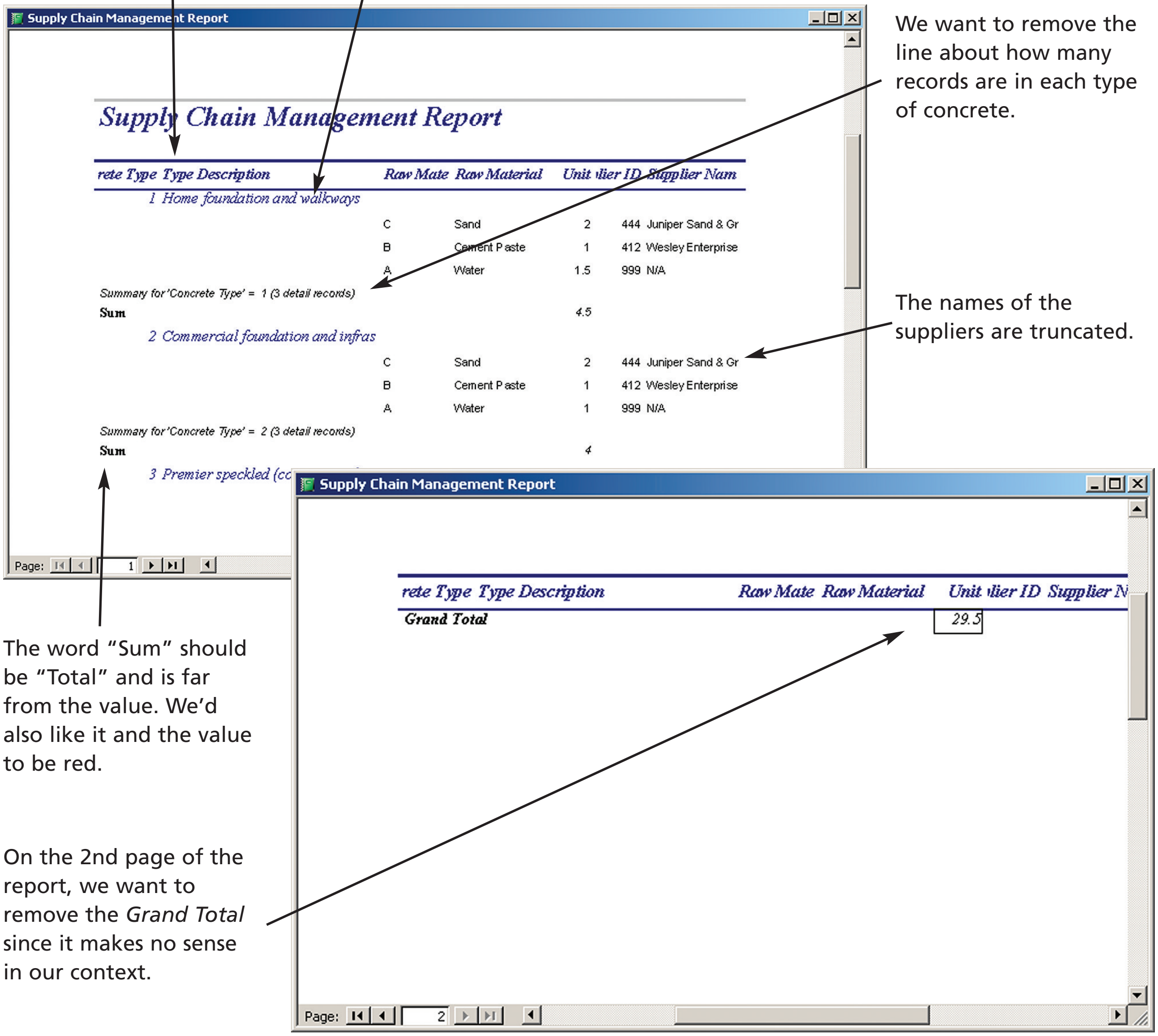
```
Supply Chain Management Report

rete Type Type Description                  Raw Mate Raw Material  Unit lier ID Supplier Nam
          1 Home foundation and walkways
                                            C        Sand           2   444 Juniper Sand & Gr
                                            B        Cement Paste   1   412 Wesley Enterprise
                                            A        Water        1.5   999 N/A
Summary for 'Concrete Type' = 1 (3 detail records)
Sum                                                               4.5
          2 Commercial foundation and infras
                                            C        Sand           2   444 Juniper Sand & Gr
                                            B        Cement Paste   1   412 Wesley Enterprise
                                            A        Water          1   999 N/A
Summary for 'Concrete Type' = 2 (3 detail records)
Sum                                                                 4
          3 Premier speckled (concrete with p
                                            D        Gravel         3   444 Juniper Sand & Gr
                                            C        Sand           2   444 Juniper Sand & Gr
                                            B        Cement Paste   1   412 Wesley Enterprise
                                            A        Water        1.5   999 N/A
Summary for 'Concrete Type' = 3 (4 detail records)
Sum                                                               7.5
          4 Premier marble (concrete with cru
                                            E        Marble         2   499 A&J Brothers
                                            C        Sand           2   444 Juniper Sand & Gr
                                            B        Cement Paste   1   412 Wesley Enterprise
                                            A        Water        1.5   999 N/A
Summary for 'Concrete Type' = 4 (4 detail records)
Sum                                                               6.5
          5 Premier shell (concrete with shell
```

The report shows all the information on the *Supply Chain Management Report* in *Module C.* But if we want to alter anything in the report we have to do so in the **Design view.**

Figure J.20

Creating a Report with Grouping, Sorting, and Totals, *continued*



```
Supply Chain Management Report

Supply Chain Management Report

rete Type  Type Description          Raw Mate  Raw Material    Unit  lier ID  Supplier Nam
     1  Home foundation and walkways
                                      C         Sand             2    444  Juniper Sand & Gr
                                      B         Cement Paste     1    412  Wesley Enterprise
                                      A         Water          1.5    999  N/A
Summary for 'Concrete Type' =  1 (3 detail records)
Sum                                                            4.5
     2  Commercial foundation and infras
                                      C         Sand             2    444  Juniper Sand & Gr
                                      B         Cement Paste     1    412  Wesley Enterprise
                                      A         Water            1    999  N/A
Summary for 'Concrete Type' =  2 (3 detail records)
Sum                                                              4
     3  Premier speckled (co

Page: 1
```

```
Supply Chain Management Report

rete Type  Type Description          Raw Mate  Raw Material    Unit  lier ID  Supplier N
Grand Total                                                   29.5

Page: 2
```

The column titles are truncated.

The concrete *Type Description* is not complete.

We want to remove the line about how many records are in each type of concrete.

The names of the suppliers are truncated.

The word "Sum" should be "Total" and is far from the value. We'd also like it and the value to be red.

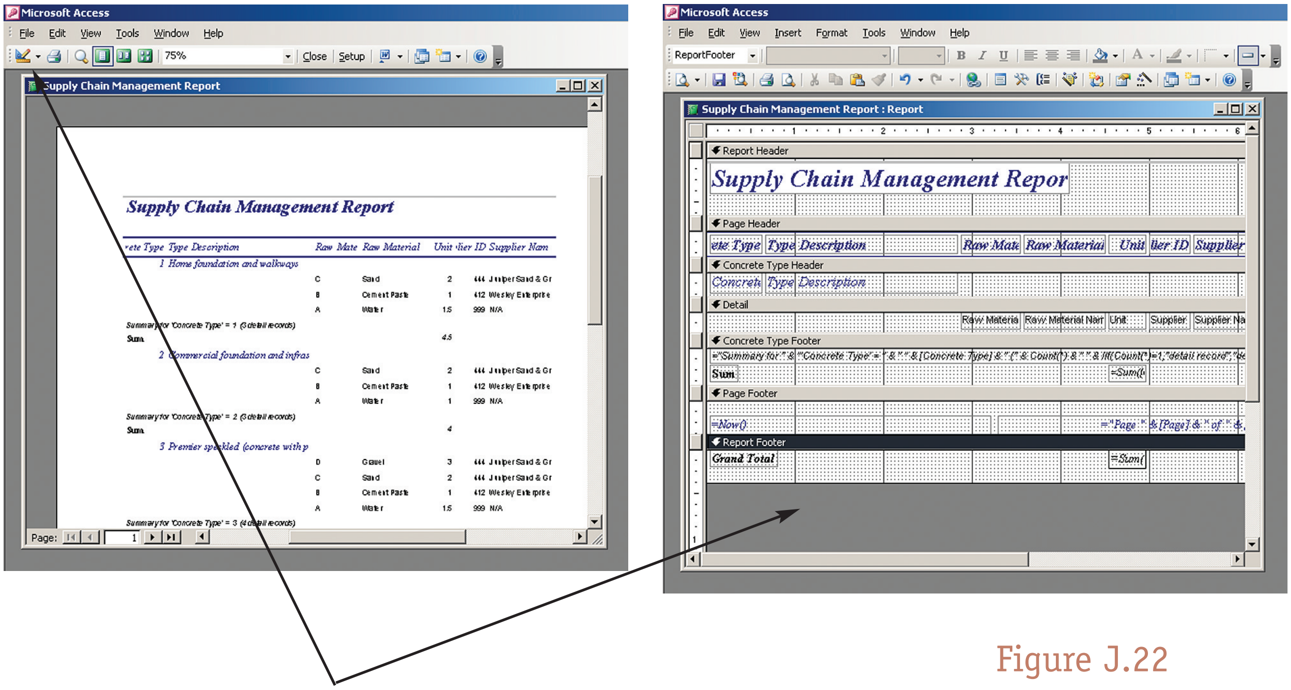
On the 2nd page of the report, we want to remove the *Grand Total* since it makes no sense in our context.

Figure J.21

The Supply Chain Management Report Generated by the Report Wizard

9. Overall structure of report: Here we accepted **Stepped,** the default **Layout** and **Portrait** as the page **Orientation.** Lastly, we clicked on **Next>**.
10. Style of report: Here we can choose from various report styles. We chose **Corporate** and then clicked on **Next>**.
11. Report heading: Here we entered "Supply Chain Management Report" (without the quotation marks) for the heading and clicked on **Finish.**

Look at the final screen of Figure J.20. You can see that all the information we wanted is there, grouped and sorted as we specified. The problem is that the presentation isn't aesthetically pleasing. See Figure J.21 for a closer look. Note the column headings. They



```
Supply Chain Management Report

·ete Type  Type Description          Raw Mate  Raw Material   Unit ·lier ID Supplier Nam
        1  Home foundation and walkways
                                     C         Sand           2    444  JuniperSand & Gr
                                     B         Cement Paste   1    412  Wesley Enterprise
                                     A         Water          1.5  999  N/A
Summary for 'Concrete Type' = 1 (3 detail records)
Sum                                                           4.5
        2  Commercial foundation and infras
                                     C         Sand           2    444  JuniperSand & Gr
                                     B         Cement Paste   1    412  Wesley Enterprise
                                     A         Water          1    999  N/A
Summary for 'Concrete Type' = 2 (3 detail records)
Sum                                                           4
        3  Premier speckled (concrete with p
                                     D         Gravel         3    444  JuniperSand & Gr
                                     C         Sand           2    444  JuniperSand & Gr
                                     B         Cement Paste   1    412  Wesley Enterprise
                                     A         Water          1.5  999  N/A
Summary for 'Concrete Type' = 3 (4 detail records)
```

Microsoft Access
File Edit View Tools Window Help
75%
Close Setup
Supply Chain Management Report
Page: 1
Microsoft Access
File Edit View Insert Format Tools Window Help
ReportFooter
Supply Chain Management Report : Report
Report Header
Supply Chain Management Repor
Page Header
ete Type Type Description Raw Mate Raw Materia Unit lier ID Supplier
Concrete Type Header
Concrete Type Description
Detail
Raw Materia Raw Material Nan Unit Supplier Supplier Na
Concrete Type Footer
="Summary for " & "'Concrete Type' = " & " " & [Concrete Type] & " (" & Count(*) & " " & IIf(Count(*)=1,"detail record","de
Sum =Sum(
Page Footer
=Now() ="Page " & [Page] & " of " &
Report Footer
Grand Total =Sum(

Click on the protractor button to get the report in Design View.

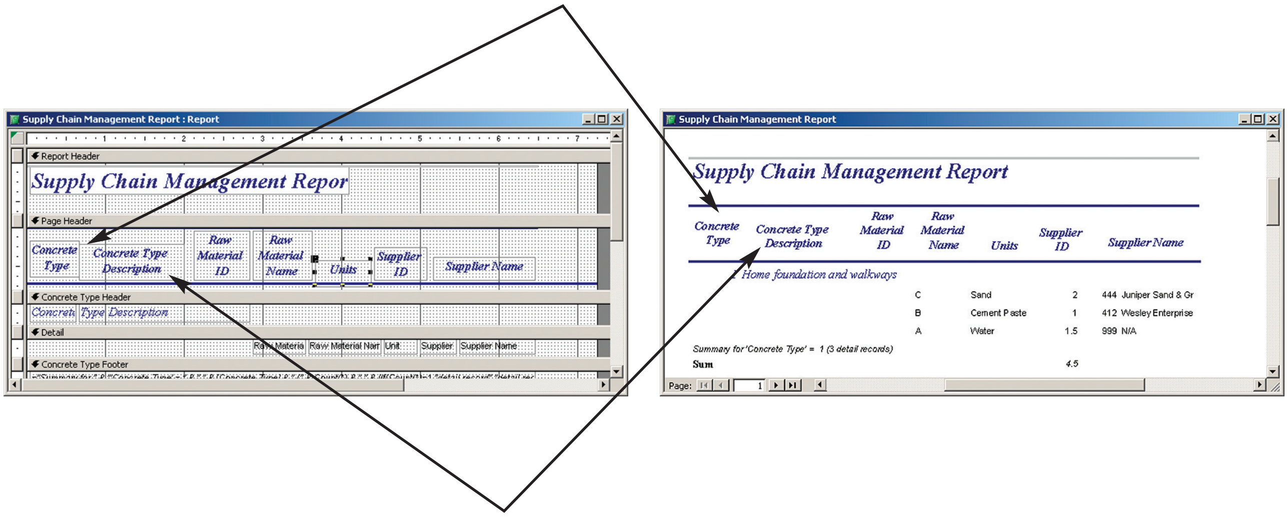
Figure J.22

The Supply Chain Management Report in Design View

appear incomplete and seem to be overlapping each other. The concrete *Type Description* entries are truncated. The word "Sum" is far away from the number. Some of the names of the suppliers are truncated. There is a second page to the report that has a *Grand Total* which we don't need since it makes no sense in this context.

We can fix these things and implement other presentation enhancements by using the Design view of the report. So we opened the *Supply Chain Management Report* and clicked on the triangle button in the top left-hand corner of the Access screen (see Figure J.22). The Design view screen divides the report into the following sections: *Page Header, Concrete Type Header* (as specified in step 8 of the report generation process), *Detail, Concrete Type Footer* (as specified in step 7), *Page Footer,* and *Report Footer.* By clicking on the boxes within these dividers, we can change their text, font, color, size, and position, etc.

We enlarged the *Concrete Type* box and centered the words.



```
Supply Chain Management Report

Supply Chain Management Report

Concrete   Concrete Type   Raw Material   Raw Material   Units   Supplier   Supplier Name
Type       Description     ID             Name                   ID
   1 Home foundation and walkways
                                   C              Sand             2      444  Juniper Sand & Gr
                                   B              Cement Paste     1      412  Wesley Enterprise
                                   A              Water            1.5    999  N/A
Summary for 'Concrete Type' =  1 (3 detail records)
Sum                                                                4.5
Page: 1
```

We added the word "Concrete" to the *Type Description* title box, made it longer, and moved it down.

Figure J.23

Changing the *Page Header* Section of the Report

PAGE HEADER

In Figure J.23, you see that we made the title box for the *Concrete Type* larger and centered the text; moved the title box for *Type Description* down a little; changed it to read "Concrete Type Description"; and made it longer. We also moved the horizontal line under the *Page Header* boxes so that it's below them. Similarly, we rearranged the other *Page Header* boxes so that they look nice.

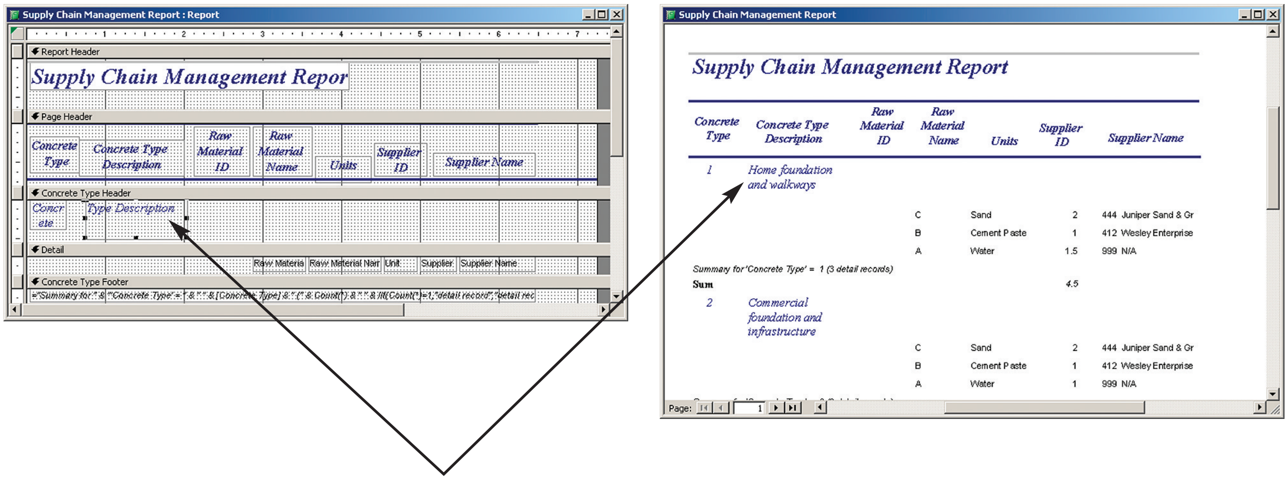
To see the effect of our changes on the report, we clicked on the datasheet view button, which is located in the top left-hand corner of the Access screen (in the spot where the triangle was).

CONCRETE TYPE HEADER

Next we'll tackle the *Concrete Type Header.* In Figure J.24 you can see that we dragged the *Type Description* box down to make it larger so that the whole description is visible. We also shrank the *Concrete Type* box so that the header is further to the left.

Figure J.24

Changing the *Concrete Type Header* Section of the Report

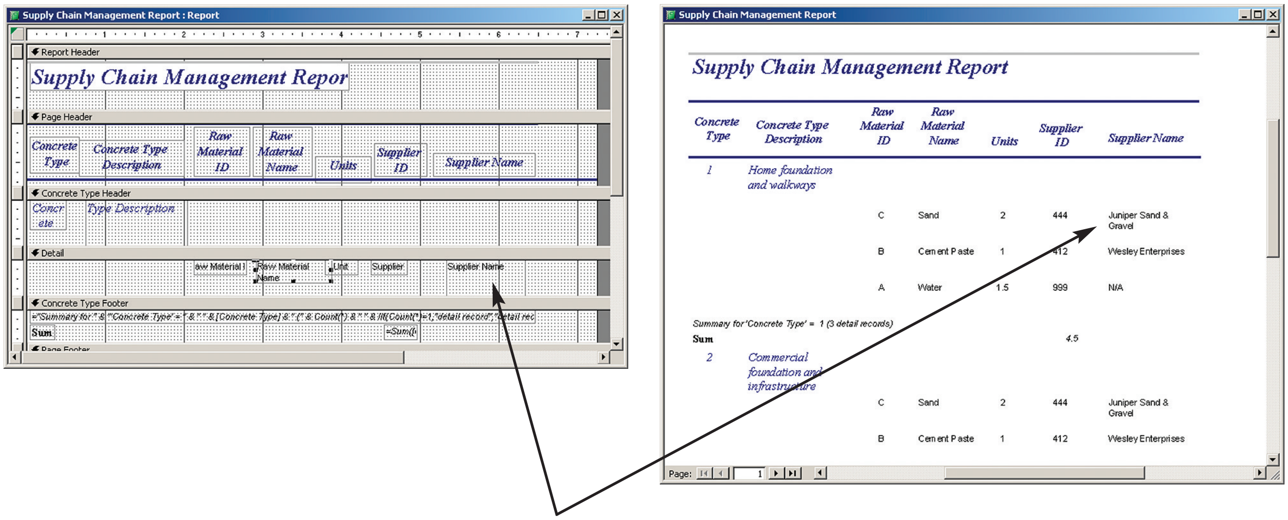


```
Supply Chain Management Report

Supply Chain Management Report

Concrete   Concrete Type   Raw Material   Raw Material   Units   Supplier   Supplier Name
Type       Description     ID             Name                   ID
   1       Home foundation
           and walkways
                                   C              Sand             2      444  Juniper Sand & Gr
                                   B              Cement Paste     1      412  Wesley Enterprise
                                   A              Water            1.5    999  N/A
Summary for 'Concrete Type' =  1 (3 detail records)
Sum                                                                4.5
   2       Commercial
           foundation and
           infrastructure
                                   C              Sand             2      444  Juniper Sand & Gr
                                   B              Cement Paste     1      412  Wesley Enterprise
                                   A              Water            1      999  N/A
Page: 1
```

Make the *Concrete Type Description* larger so that the entire description is visible.


Supply Chain Management Report : Report
Report Header
Supply Chain Management Repor
Page Header
Concrete Type
Concrete Type Description
Raw Material ID
Raw Material Name
Units
Supplier ID
Supplier Name
Concrete Type Header
Concr ete
Type Description
Detail
aw Material I
Raw Material Name
Unit
Supplier
Supplier Name
Concrete Type Footer
="Summary for " & "'Concrete Type' = " & " " & [Concrete Type] & " (" & Count(*) & " " & IIf(Count(*)=1,"detail record","detail rec
Sum
=Sum([
Page Footer
Supply Chain Management Report
Page: 1

Supply Chain Management Report

Concrete Type	Concrete Type Description	Raw Material ID	Raw Material Name	Units	Supplier ID	Supplier Name
1	Home foundation and walkways					
		C	Sand	2	444	Juniper Sand & Gravel
		B	Cement Paste	1	412	Wesley Enterprises
		A	Water	1.5	999	N/A
Summary for 'Concrete Type' = 1 (3 detail records)						
Sum					4.5	
2	Commercial foundation and infrastructure					
		C	Sand	2	444	Juniper Sand & Gravel
		B	Cement Paste	1	412	Wesley Enterprises

We enlarged *Supplier Name* so that there would be room for the whole name in every case and moved the boxes to line up with their corresponding headings.

Figure J.25

Changing the *Detail* Section of the Report

DETAIL

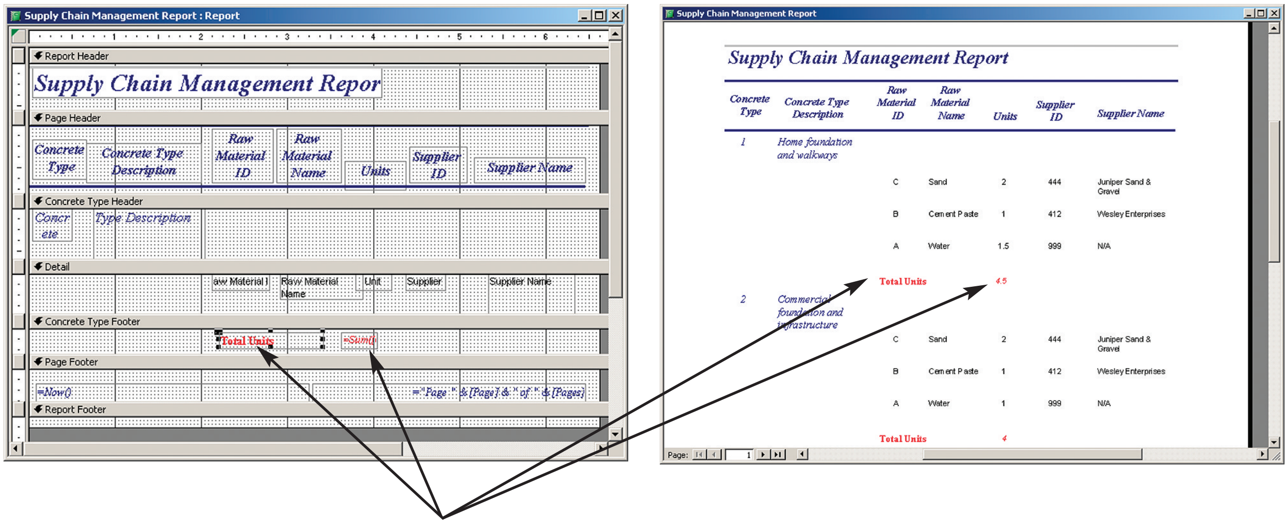
In the *Detail* section, the *Supplier Name* field was truncated on the original report. So, in the Design view we enlarged that *Supplier Name* box so that long names appear on two lines in their entirety (see Figure J.25).

CONCRETE TYPE FOOTER

The top box in the *Concrete Type Footer* section of the report in Design view shows details of the summary lines on the report that shows the *Concrete Type* and the number of records for each type of concrete. We removed that by highlighting the box and pressing the **Delete** key on the keyboard and you can see the result in Figure J.26. We changed the contents of the **Sum** label box to "Total Units" written in red font, and then moved it closer to where the total value appears. We also changed the color of the font in the box with the total value—the one with "=Sum([Unit])". We moved both these boxes so they look better on the report.

Figure J.26

Changing the *Concrete Type Footer* Section of the Report



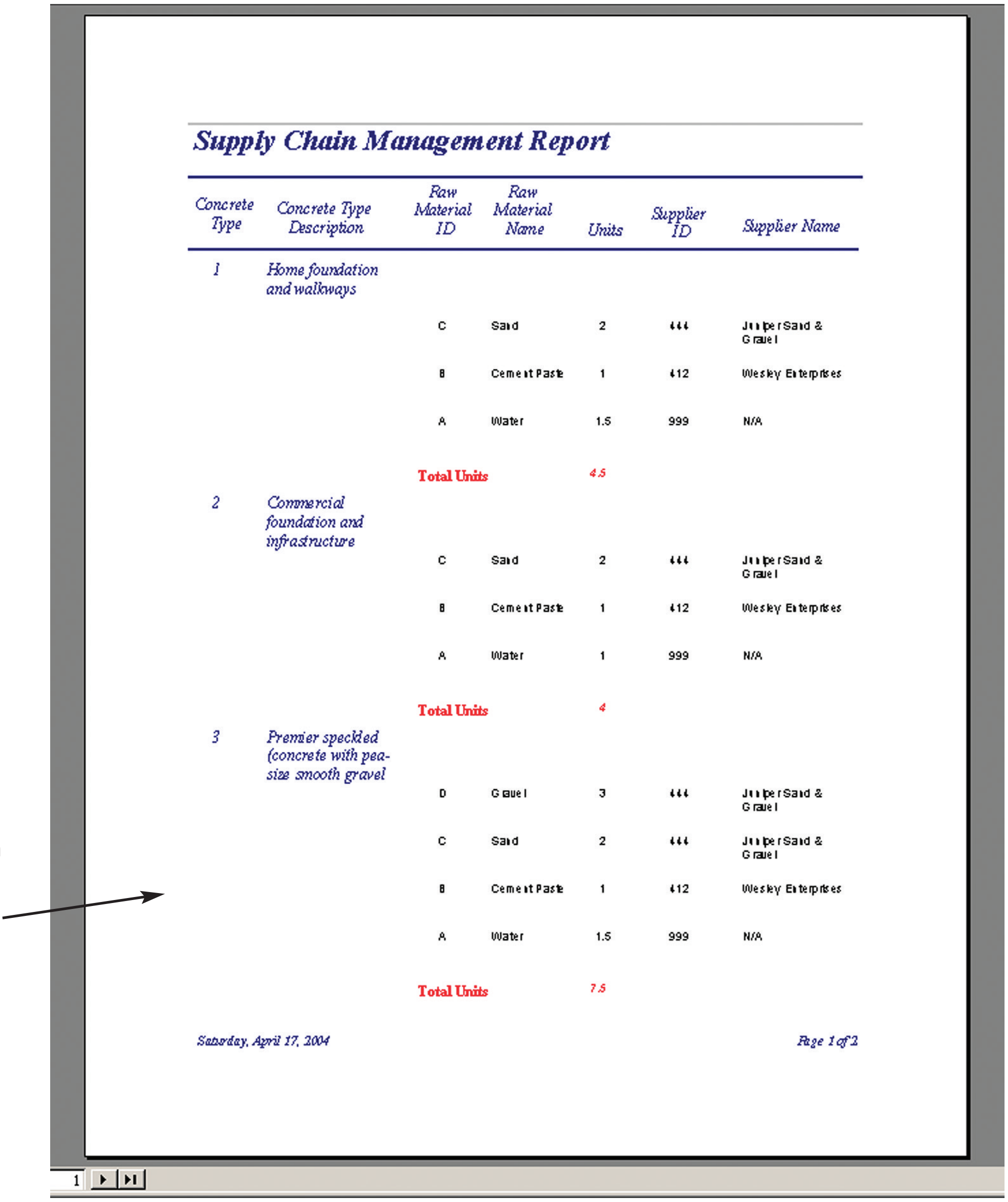
Supply Chain Management Report

Concrete Type	Concrete Type Description	Raw Material ID	Raw Material Name	Units	Supplier ID	Supplier Name
1	Home foundation and walkways					
		C	Sand	2	444	Juniper Sand & Gravel
		B	Cement Paste	1	412	Wesley Enterprises
		A	Water	1.5	999	N/A
			Total Units	4.5		
2	Commercial foundation and infrastructure					
		C	Sand	2	444	Juniper Sand & Gravel
		B	Cement Paste	1	412	Wesley Enterprises
		A	Water	1	999	N/A
			Total Units	4		

We deleted the summary lines, changed the wording in the **"Sum"** label, and moved it closer to the value box. We also changed the font color of the label "Total Units" and the font color of the total units value box to red.

Figure J.27

The Completed Report



Supply Chain Management Report

Concrete Type	Concrete Type Description	Raw Material ID	Raw Material Name	Units	Supplier ID	Supplier Name
1	Home foundation and walkways					
		C	Sand	2	444	Juniper Sand & Gravel
		B	Cement Paste	1	412	Wesley Enterprises
		A	Water	1.5	999	N/A
		Total Units		4.5		
2	Commercial foundation and infrastructure					
		C	Sand	2	444	Juniper Sand & Gravel
		B	Cement Paste	1	412	Wesley Enterprises
		A	Water	1	999	N/A
		Total Units		4		
3	Premier speckled (concrete with pea-size smooth gravel					
		D	Gravel	3	444	Juniper Sand & Gravel
		C	Sand	2	444	Juniper Sand & Gravel
		B	Cement Paste	1	412	Wesley Enterprises
		A	Water	1.5	999	N/A
		Total Units		7.5		

Saturday, April 17, 2004

Page 1 of 2

With that change and the previous ones, the report is now tidier and looks better. You can't see here that the **Grand Total** label and value have been deleted from the *Report Footer Section.* That's on the second page of the report along with the rest of the information.

PAGE AND REPORT FOOTERS

In the *Page Footer* section, the box with "=Now()" is the command that places the date at the bottom of each page of our report. The box to the right keeps track of the current page and the total number of pages.

The first box in the *Report Footer* section puts the label "Grand Total" at the end of the report, and the box to the right places the grand total of all the units in the report at the end of the report. This doesn't make any sense in our context, so we can delete both of these grand total boxes.

Figure J.27 shows the revised *Supply Chain Management Report.* It's much tidier, more informative, and more pleasing to the eye.

You can do much more with reports in the Design view. You can put in totals, averages, and lots of other things. We'll leave this for you to investigate on your own.

ON YOUR OWN

CREATE AN ACCOUNTS RECEIVABLE REPORT

An accounts receivable report tells you how much money your customers owe you.

Add a column (field) to the *Customer* relation called *Billing Total.* You'll need to go into the design view and modify the original structure. Don't forget to specify that the new field has a data type of **Currency.** Remember to save the new structure.

Enter the following *Billing Total* information into that field:

Customer Number	*Billing Total*
1234	$3,546.00
2345	$8,664.00
3456	$4,786.00
4567	$6,540.00
6789	$1,243.00

Create a report (you'll need only one table) that prints

- *Customer Number*
- *Customer Names* (in alphabetical order)
- *Billing Total* for each customer
- Total of accounts due (page total)

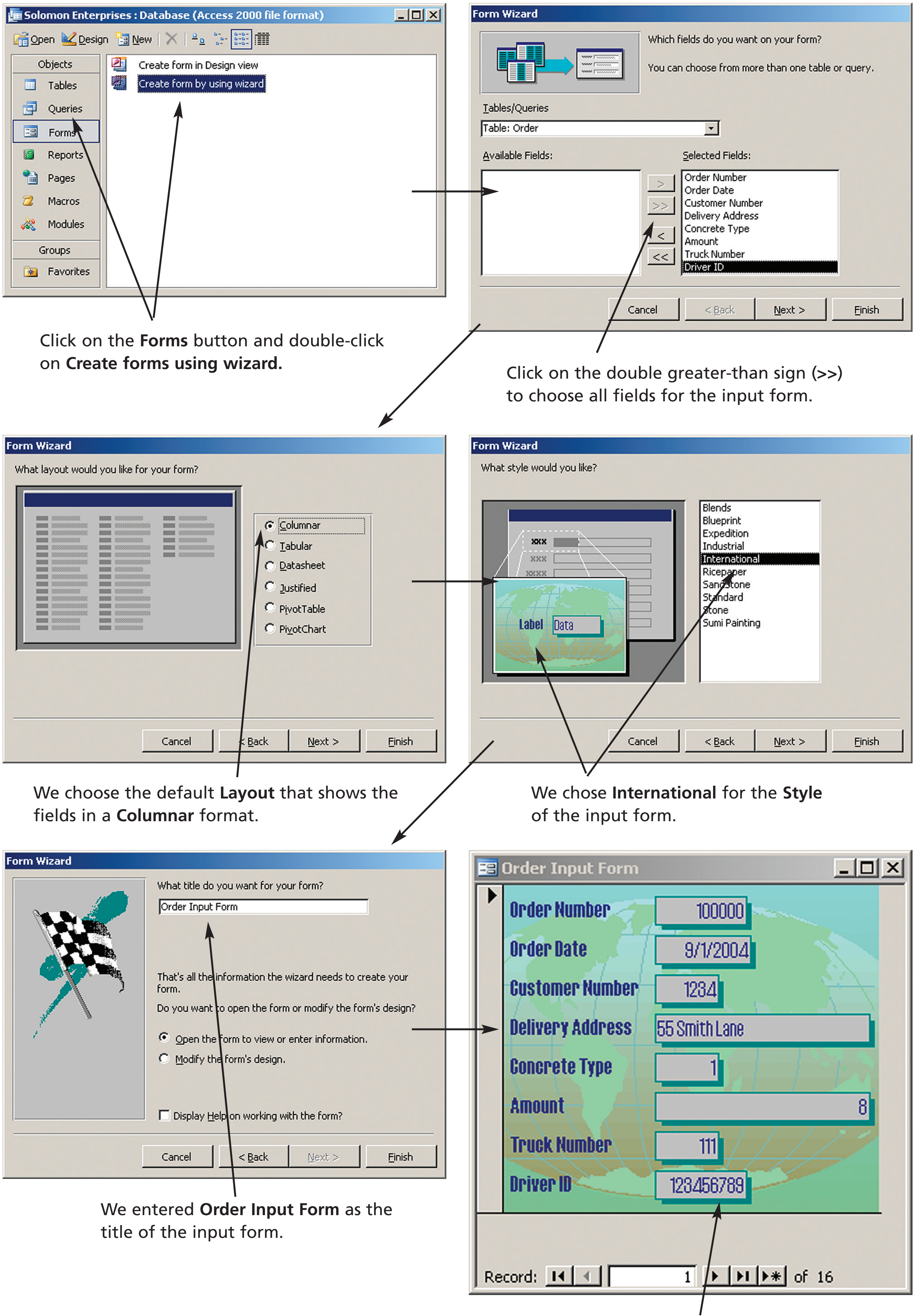
Creating a Data Input Form

Our last task in this database is to design an input form to simplify the task of entering new information. Let's create an input form for new orders. Here, too, we used the wizard. It's actually quite simple. Here are the steps (see Figure J.28 on the next page):

1. Click on **Forms** and then on **Create form by using wizard.**
2. Choose **Table: Order.**
3. Click on the double greater-than symbol (**>>**) to select all the fields of the order table. Then click **Next>.**
4. You can choose a layout for your input form. We chose the default (**Columnar**). Click **Next>.**
5. This screen allows you to choose a style. We chose **International.** Then click **Next>.**
6. Enter a title for the input Form. We chose "Order Input Form" (without the quotes). Then click **Finish.**

When you complete these steps, the finished form appears. You can move through the records with the arrow keys. When you reach the end of the list of records, you'll get a blank input form (see Figure J.28). Using this screen, you can enter a new order.

You can change the layout of the input form after you create it with the wizard, just as we did with the report.



Solomon Enterprises : Database (Access 2000 file format)

Create form in Design view

Create form by using wizard

Click on the **Forms** button and double-click on **Create forms using wizard.**

Form Wizard

Which fields do you want on your form?

You can choose from more than one table or query.

Tables/Queries: Table: Order

Available Fields:

Selected Fields:

- Order Number
- Order Date
- Customer Number
- Delivery Address
- Concrete Type
- Amount
- Truck Number
- Driver ID

Click on the double greater-than sign (>>) to choose all fields for the input form.

Form Wizard

What layout would you like for your form?

- ● Columnar
- ○ Tabular
- ○ Datasheet
- ○ Justified
- ○ PivotTable
- ○ PivotChart

We choose the default **Layout** that shows the fields in a **Columnar** format.

Form Wizard

What style would you like?

- Blends
- Blueprint
- Expedition
- Industrial
- International
- Ricepaper
- Sandstone
- Standard
- Stone
- Sumi Painting

We chose **International** for the **Style** of the input form.

Form Wizard

What title do you want for your form?

Order Input Form

That's all the information the wizard needs to create your form.

Do you want to open the form or modify the form's design?

- ● Open the form to view or enter information.
- ○ Modify the form's design.

□ Display Help on working with the form?

We entered **Order Input Form** as the title of the input form.

Order Input Form

Order Number	100000
Order Date	9/1/2004
Customer Number	1234
Delivery Address	55 Smith Lane
Concrete Type	1
Amount	8
Truck Number	111
Driver ID	123456789

Record: 1 of 16

Figure J.28

Creating an Input Form

The completed input form allows you to scroll through the arrows using the arrow buttons at the bottom. The last entry will be blank and there you can add a new **Order** record.

That ends our brief tour of Microsoft Access. It's not really a difficult package to learn. The most difficult part is correctly defining the structure of your database. Actually implementing your design in Access is comparatively simple after that. We recommend once again that you reread *Extended Learning Module C* if you intend to become a designer as well as a user of database applications. "Using" is the easy part; "designing" is the challenging part.

Summary: Student Learning Outcomes Revisited

1. **Identify the steps necessary to implement the structure of a relational database using the data definition language provided by Microsoft Access.** Using a data definition language to implement a database is the fourth and last step in designing a database. It first requires that you use the data dictionary to create the structure of each table, assigning each one a primary key, and defining what type of information the fields will hold, how large they will be, and many other properties. The ***data dictionary*** contains the logical structure for the information in a database. A ***primary key*** is a field (or group of fields in some cases) that uniquely defines each record. A ***composite primary key*** consists of the primary key fields from two intersecting relations. An ***intersection relation*** (sometimes called a ***composite*** relation) is a relation you create to eliminate a many-to-many relationship.

   You must also define the relationships among the tables, including their foreign keys. A ***foreign key*** is a primary key of one file (relation) that appears in another file (relation). You must also consider ***integrity constraints***—rules that help ensure the quality of the information. For this you use the *Enforce Referential Integrity* feature of Access to stipulate that no information may be entered as a foreign key unless that information already exists as a primary key.
2. **Demonstrate how to use the data manipulation subsystem in Access to enter and change information in a database and how to query that information.** When the tables and their fields have been defined and the relationships between pairs of tables have been established, you can enter information into the tables. Then you can construct queries to view the information in multiple ways, i.e., you can ask questions of the information. The simplest way to do this is to use Access's QBE tool. A ***query-by-example (QBE) tool*** helps you graphically design the answer to a question.
3. **Explain the use of the application generation subsystem in Access to create reports and data entry screens.** To create professional-looking reports you must use the report generator in the application generation subsystem of Access. It allows you to present information with page headers and footers, and grouping, sorting, and totaling of information. Similarly, to create an easy-to-use method of entering information into tables, you create input forms. This is also part of the application generation subsystem of Access.

Key Terms and Concepts

Composite primary key, 9
Data dictionary, 4
Foreign key, 11
Integrity constraint, 7
Intersection (composite) relation, 9
Primary key, 7
Query-by-example (QBE) tool, 17

Short-Answer Questions

1. What is the difference between a report and a query?
2. How many characters can a text field have in Access?
3. What is the difference between a data manipulation language and a data definition language?
4. Would it be a good idea to make every field as big as it can be to ensure that information will always fit? Why or why not?
5. What is a QBE tool?
6. Why is it important to establish relationships between pairs of tables and to link primary keys to foreign keys?
7. What is a foreign key?
8. Is it possible to change the structure of a table (relation) when you have already entered information into that table?
9. What is the purpose of a data dictionary?
10. A query looks very much like a table at first glance. How do they differ?
11. What function does the Field Properties part of the data dictionary have?
12. What is a primary key?
13. What is a composite primary key?
14. What is an intersection relation?
15. What are integrity constraints?
16. What is the purpose of referential integrity? Where would you use that feature when building a database with Access?
17. When using a QBE tool, where do you enter the condition information?
18. How does a report differ from the datasheet view of a query?
19. How would you include totals or averages on your report without creating the report in Design view?
20. What is a Form in Access?
21. How do you define a composite primary key in Access?

Assignments and Exercises

1. **EXCEL VERSUS ACCESS** An Access relation (table) has rows and columns and so does an Excel worksheet, so how do workbooks and databases differ? Discuss both the creation and use of each.
2. **ENTER NEW EMPLOYEE INFORMATION** It's likely that Solomon Enterprises would need to enter the information for a new employee. Create a new input form to enter *Employee ID, Employee Last Name, Employee First Name,* and *Date of Hire.* Design the form so that the information appears in tabular form and has Sandstone background.
3. **INVESTIGATE SQL** In this module we created our queries using the query-by-example tool in the data manipulation subsystem of Access. In Chapter 3, you learned about structured query language (SQL), which does the same thing as the query-by-example, but instead of drag and drop, you write actual code. You can see an example of SQL by looking at any query and asking Access to show you the corresponding SQL code. To do this, open up any query in either the Datasheet or Design view, click on the **View** menu and choose **SQL view.** Here you'll see how we would have written our query if we hadn't used QBE. Which would you prefer to use? Can you imagine why QBE was invented as an alternative to SQL?
4. **WHAT ARE THE INGREDIENTS FOR PREMIER MARBLE CONCRETE?** Write a query to show how many units of each raw material are in concrete type 4. Print out the name of the concrete type, its ID, the name of the raw material (not its ID), and the number of units of each of the raw materials.
5. **INVENTORY REPORT** Create a report that shows how many units Solomon Enterprises has of each of the raw materials. Don't include water (hint: you want all *Raw Material* fields that do not equal *water*). Choose your own layout and page orientation.

6. **SORT QUERY INFORMATION** The Datasheet view of a query allows you to sort the information in that query. Try this out with the *Order* relation. Download the Solomon Enterprises database from the Web site that supports this text: www.mhhe.com/haag. (Select XLM/J. The name of the file is **XLMJ_Solomon_Enterprises.mdb.**) Click on the **Queries** tab and open the *Order* relation in Datasheet view. Sort the table alphabetically on *Employee Last Name.* Place your cursor anywhere in the column you want to sort by (in this case *Employee Last Name*), and click on the sort-ascending button. It has an "A" above a "Z" with an arrow pointing downwards.
7. **FILTER QUERY INFORMATION** You can request Access to show you any other occurrences of a data item that is in the same column. Use the same file you needed for question 6 above. You can download it from the Web site for this text: www.mhhe.com/haag. (Select XLM/J. The name of the file is **XLMJ_Solomon_Enterprises.mdb.**) Filter the information so that only those records in which the truck is a Ford appear. To do this click in the *Truck Type* column on any one of the occurrences of Ford. Then, click on the **Filter by Selection** button (that's the button in the button bar with the funnel and a lightning strike). You will instantly see only the three records where the truck is a Ford. To return the data to its previous state, click on **Remove Filter** button (that's the button that has the funnel without any other symbol). If you click on this button again, it will reapply the filter.