

Setting Up An Invoice System

To fully understand the value of relational databases you need to create a detailed system. In this chapter you will setup an invoicing system for a computer mail order company, PC Direct, which sells computer peripherals through the mail. The invoice system will then be completed in the following three chapters. There are three main sections to the system, customers, products and the sales invoice.

In the creation of any database system you should do some careful planning. In general there are four steps that you should undertake.

- 1 Decide how many tables you think you might need.
- 2 Decide how the tables will be related to one another.
- 3 List the fields in each table trying to avoid having the same field in more than one table. Decide which fields will be the PRIMARY KEY and FOREIGN KEY fields to link the tables.
- 4 Decide what forms and/or reports (or printouts) are required.

In the case of PC Direct an initial TABLE RELATIONSHIP diagram might be:



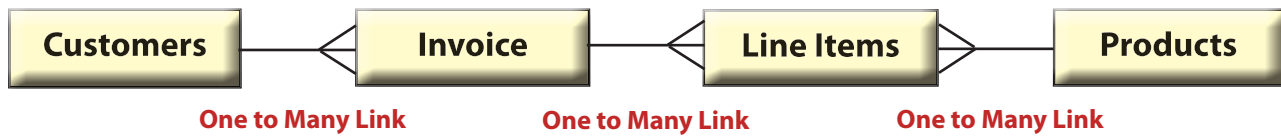
The CUSTOMERS to INVOICE section of this structure forms a ONE TO MANY relationship. The one customer can have MANY invoices sent to them over time, but there will always be ONE customer for each invoice. So the customer table is the ONE side of the relationship and the INVOICE is the MANY side. We can set a relationship to link these two tables.

There is a problem with the link between the INVOICE and the PRODUCTS tables. The one invoice can contain many products and the one product can be included in many invoices. A relational database cannot cater for a MANY TO MANY relationship as you cannot set multiple PRIMARY or FOREIGN KEY fields in the one relationship.

A further problem exists, one invoice might contain a sale of 5 of a particular item, the next invoice might contain a sale of 2 of the same item. The company needs a way of adding these sales so that it knows how many items it has sold. So this initial TABLE RELATIONSHIP will need modification.

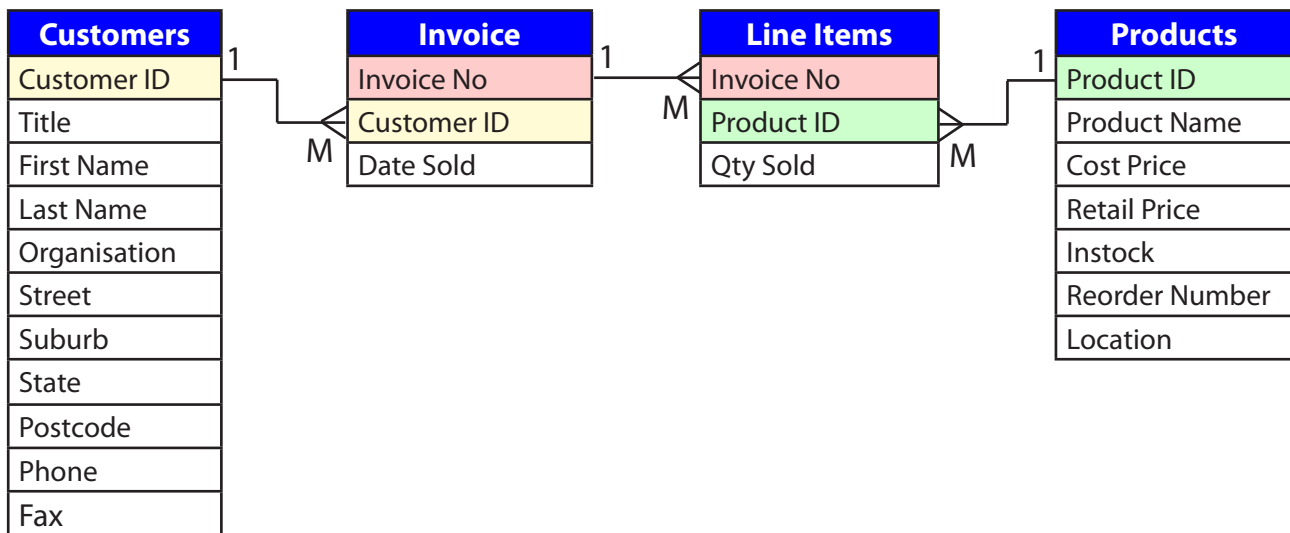
The easiest solution to these problems is to add a table between INVOICE and PRODUCTS. This table can store every item sold by the company as a single record allowing the company to keep track of every item sold. The new table can also provide data to the INVOICE table.

So a revised TABLE RELATIONSHIP diagram becomes:



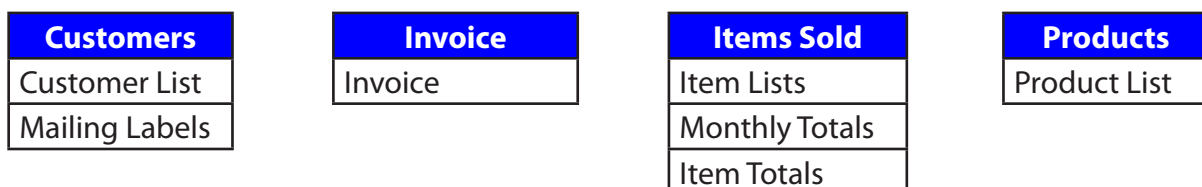
The ONE invoice can have MANY line items sold within it. The ONE product can be sold MANY times. By adding the LINE ITEMS table to the system a series of ONE TO MANY relationships are created and a relational database system can accommodate these.

The next step is to decide which fields will be placed in each table. The following TABLE STRUCTURE diagram shows one possibility. Remember, we do not want to store data more than once (except for PRIMARY KEY or FOREIGN KEY fields).



The PRIMARY KEY and FOREIGN KEY fields need to be considered. In the diagram above you can see that CUSTOMER ID is used to link the CUSTOMERS and INVOICE tables, INVOICE NO is used to link the INVOICE and LINE ITEMS tables and PRODUCT ID is used to link the LINE ITEMS and PRODUCTS table. The INVOICE table is not directly linked to the PRODUCTS table, it will obtain values from the PRODUCTS table via the LINE ITEMS table.

The final consideration in the planning is what forms and reports will be required. The following diagram shows some of the forms and reports that could be made from the various tables. We will create some of these reports in the next chapter.

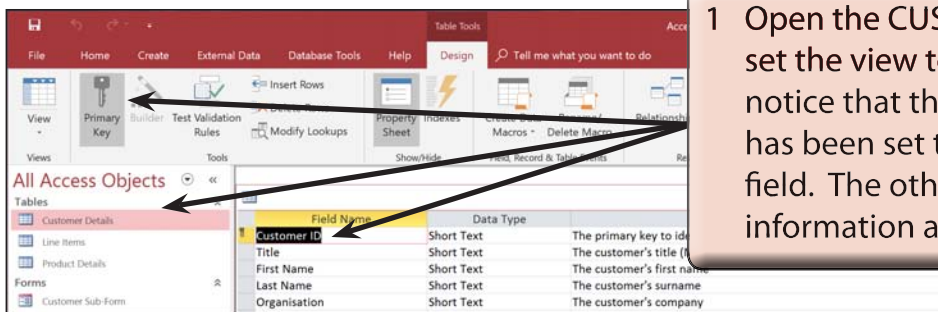


Loading the Sample File

- 1 Load Microsoft Access and click on the OPEN OTHER FILES icon in the START window or close the current file and click on the OPEN icon in the FILE tab.
- 2 BROWSE to the CHAPTER 12 folder of the ACCESS 2019 SUPPORT FILES and open the CHAPTER 12 file as an OPEN READ-ONLY file.
- 3 Click on the SAVE AS icon in the WARNING BAR under the RIBBON.
- 4 Access your STORAGE folder and save the file as CHAPTER 12 COPY.
- 5 Click on the ENABLE CONTENT button so that the data can be viewed.

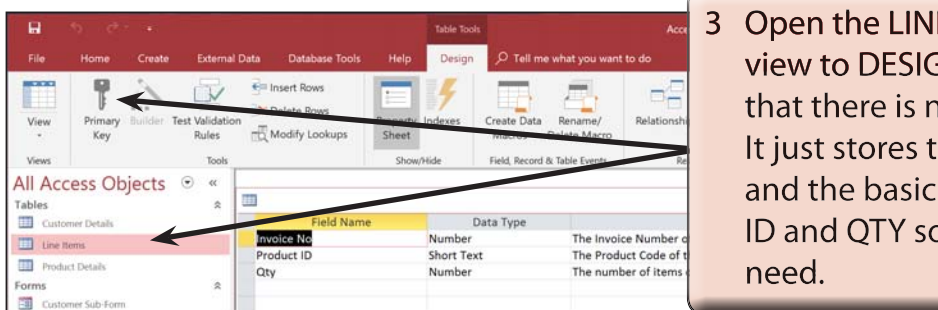
A Looking at the Tables

The database has 3 tables at the moment, CUSTOMER DETAILS which stores data about the company's customers, LINE ITEMS which stores details of each product sold and PRODUCT DETAILS which stores a list of the products that the company provides.



- 1 Open the CUSTOMER DETAILS table, set the view to DESIGN VIEW and notice that the CUSTOMER ID field has been set to the PRIMARY KEY field. The other fields store general information about the customers.

- 2 Close the CUSTOMER DETAILS table.



- 3 Open the LINE ITEMS table, set the view to DESIGN VIEW and notice that there is no PRIMARY KEY field. It just stores the INVOICE NUMBER and the basic details of PRODUCT ID and QTY sold that the invoice will need.

NOTE: The LINE ITEMS table will record each line of the invoice. It will have two main purposes:

- (A) to list each product sold as a separate record so that the company can calculate monthly sales and carry out stock checks.
- (B) to provide product data to the invoice. If you look at the diagram at the centre of page 12-2 you will see that the PRODUCTS table is not directly connected to the INVOICE table so some data, such as Product Name and Retail Price, will need to be linked to the LINE ITEMS table via a relationship to the PRODUCTS table so that the INVOICE can display them.

4 Close the LINE ITEMS table.

5 Open the PRODUCT DETAILS table, set the view to DESIGN VIEW and notice that PRODUCT ID has been set the a PRIMARY KEY field. It will provide data to the LINE ITEMS table via this field.

Field Name	Data Type	Field Properties
Product ID	Short Text	Primary key field
Product Name	Short Text	The name of the prod
Cost Price	Currency	The cost price of the s
Retail Price	Currency	The price the item is s
Instock	Number	The number of the pr

6 Close the PRODUCT DETAILS table.

B Looking at the Forms

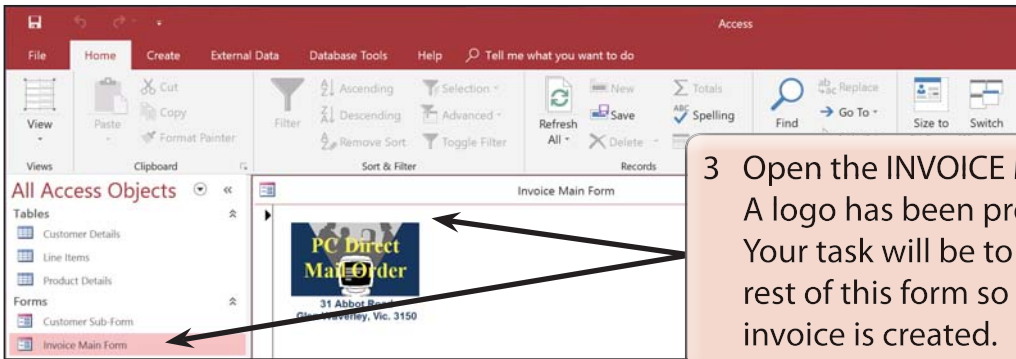
Two forms have been prepared for you. The CUSTOMER SUB-FORM which displays the details from the CUSTOMER DETAILS table and the INVOICE MAIN FORM which you will need to complete.

1 Open the CUSTOMER SUB-FORM form and notice that fields have been arranged in a normal address format.

Customer Details

Mr Colin Norris
Ballarat Electrical Services
24 Main Street BALLARAT VIC 3350

2 Close the CUSTOMER SUB-FORM form.

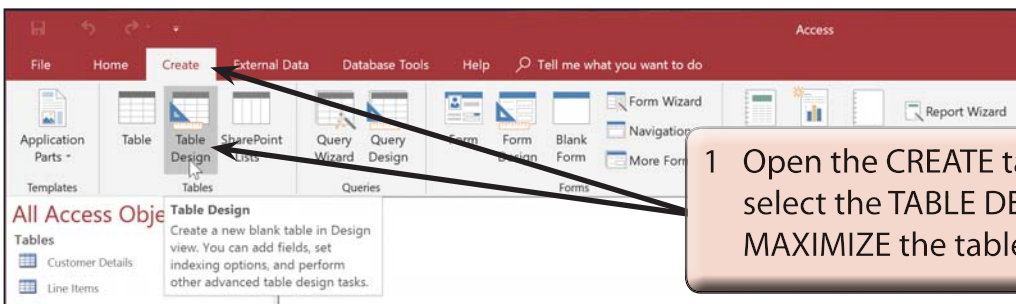


3 Open the INVOICE MAIN FORM form. A logo has been prepared for you. Your task will be to complete the rest of this form so that a functional invoice is created.

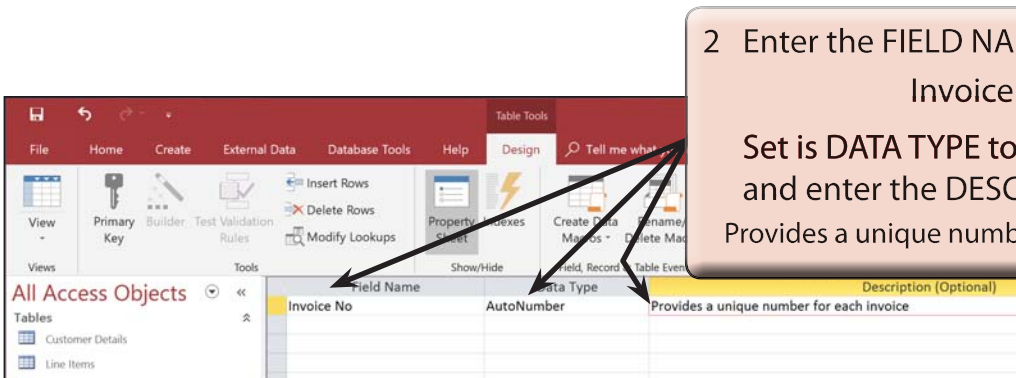
4 Close the INVOICE MAIN FORM form.

Creating the Invoice Table

An invoice table is required to store details every time an invoice is sent. This table simply records the INVOICE NUMBER, the CUSTOMER ID and the DATE of the purchase. The links that we make to the other tables will display other details such as product name and retail price.



1 Open the CREATE tab of the RIBBON, select the TABLE DESIGN icon and MAXIMIZE the table window.



2 Enter the FIELD NAME:
Invoice No
Set is DATA TYPE to AUTONUMBER and enter the DESCRIPTION:
Provides a unique number for each invoice.

NOTE: The AUTONUMBER field type will tell the program to provide a new number every time an invoice is created so that no two invoices will ever have the same number.

3 Click on the PRIMARY KEY button in the DESIGN tab to set the INVOICE NO field as the PRIMARY KEY field. A key symbol is added next to the field.

4 In the FIELD PROPERTIES pane set the FORMAT box to 00000.

Field Name	Data Type
Invoice No	AutoNumber

Property	Value
Field Size	Long Integer
New Values	Increment
Format	00000
Caption	

NOTE:

- i The PRIMARY KEY field will index the table so that it is listed in INVOICE NO order.
- ii The format (00000) will set each number to five digits, for example, 00001, 00002, etc.

5 Click in the second FIELD NAME cell and enter the field name:
Customer ID

6 Set DATA TYPE to SHORT TEXT, with a FIELD SIZE of 10 and a DESCRIPTION:
Identity code for each customer

Field Name	Data Type	Description (Optional)
Invoice No	AutoNumber	Provides a unique number for each invoice
Customer ID	Short Text	Identity code for each customer

Property	Value
Field Size	10
Format	