


In this section I will be planning my tables and the complete database for the new system. This will include my designs for queries, forms, sub-forms, linked-forms, relationships and macros. Once I have finished this I will be using a test plan to make sure that my system will work the way that I expect it to work. This test plan will ensure that my system is accurate and has accomplished many of the goals.

TABLES

I will create the tables in DESIGN mode as this will allow me to make more customised changes. In DESIGN mode I can enter the field name, the data type and any extra comments, which would not be possible in the WIZARD mode. Also, I can add validation rules and input masks in the field properties section. These are the tables I plan to create:

ITEMS

This table will hold the details of the item and its stock information. This is the record structure of this table:



FIELD NAME	DATA TYPE	EXAMPLE
ICODE 	NUMBER	10304
NAME	TEXT	WEGA T40
SCODE	NUMBER	3100
SUPPLIER	TEXT	Sony
ITEM TYPE	TEXT	TV
PIC	OLE Object	(a picture file)
PRESENT	NUMBER	310
MIN LEVEL	NUMBER	200
MAX LEVEL	NUMBER	330
STATUS	TEXT	OK
COST PRICE	NUMBER	Dhs. 500
SELL PRICE	NUMBER	Dhs. 725

To increase the level of integrity of the data entered, I will be using some simple validation rules to accompany the table. If any of these rules are violated, then an error message is displayed. The error message displays what is entered in VALIDATION TEXT. These are the validation rules and validation texts that I have inserted:

FIELD NAME	VALIDATION RULE	VALIDATION TEXT
NAME	Is Not Null	Please enter the item name
PRESENT	Is Not Null	Enter present level
MIN	Is Not Null	Enter minimum level
MAX	Is Not Null	Enter maximum level
SELL PRICE	Is Not Null	Enter selling price

Sometimes data is required in a certain format. For this reason I will be using the INPUT MASK PROPERTY. An input mask is used in fields to format data and provide some control

over what values can be entered. An input mask consists of literal characters (such as spaces, dots, dashes, and parentheses) that separate blanks to fill in. I will use the input mask property to make the input data formatted in the right way:

FIELD NAME	INPUT MASK	EXAMPLE
ICODE 	99999	54621
SCODE	9999	9980
MIN	9999	130
MAX 	9999	470
COST PRICE	"Dhs. "99999".00"	Dhs. 547.00
SELL PRICE	"Dhs. "99999".00"	Dhs. 1,050.00






TYPES

This table contains the type of items that the company holds, eg. TV, mobile phone etc. This table will be used for queries and for grouping (relationships + sub-forms) the items by item types so that the company can see which companies deal with the concerned product. Since there is only one field in this table I will leave it at that. Hence I will not need any validation rules or input masks for this table. This is the record structure of the file:

FIELD NAME	DATA TYPE	EXAMPLE
ITEM TYPE 	TEXT 	Game console

SUPPLIERS

This table contains all the details of the suppliers that are supplying goods to Elektronikz. This table is used for statistical purposes and to get into contact with the suppliers through the use of letters. This is the record structure of the file:

FIELD NAME	DATA TYPE	EXAMPLE
SCODE 	NUMBER	3100
SUPPLIER	TEXT 	Sony
POBOX 	NUMBER	4988
CITY[*]	TEXT 	Sharjah
TELEPHONE	TEXT 	(074) 583 9240

[*]Here I will use the lookup wizard to make the field display a combo box with the options **Sharjah, Abu Dhabi, Dubai, Fujairah, Al-Ain, RAK and UAQ**. I have explained the lookup wizard below the PURCHASE table design.

Validation Rules...

FIELD NAME	VALIDATION RULE	VALIDATION TEXT
SUPPLIER	Is Not Null	Enter supplier's name
TELEPHONE	Is Not Null	Enter telephone number

Input Masks:

FIELD NAME	INPUT MASK	EXAMPLE
SCODE	9999	6565
TELEPHONE	99"-9999999	04-3367158

PURCHASE

This table holds the details of all the customers who have purchased goods from Elektronik. I will explain some of the fields in this table so that it can be understood fully. This is the record structure of the table:

FIELD NAME	DATA TYPE	EXAMPLE
ID 	NUMBER	10001
FNAME	TEXT	Joseph
LNAME	TEXT	Khan
TELEPHONE	TEXT	06-5773582
ICODE1	NUMBER	17012
NAME1[*]	TEXT	Hoover
SUPPLIER1[**]	TEXT	National
QTY1[***]	NUMBER	2
PRICE1[****]	NUMBER	510.00
TP1[*****]	NUMBER	1020.00
ICODE2	NUMBER	17013
NAME2[*]	TEXT	Hoover
SUPPLIER2[**]	TEXT	National
QTY2[***]	NUMBER	3
PRICE2[****]	NUMBER	511.00
TP2[*****]	NUMBER	1533.00
ICODE3	NUMBER	17014
NAME3[*]	TEXT	Hoover
SUPPLIER3[**]	TEXT	National
QTY3[***]	NUMBER	2
PRICE3[****]	NUMBER	512.00
TP4[*****]	NUMBER	1024.00
ICODE4	NUMBER	17015
NAME4[*]	TEXT	Hoover
SUPPLIER4[**]	TEXT	National
QTY4[***]	NUMBER	3
PRICE4[****]	NUMBER	513.00
TP4[*****]	NUMBER	1539.00
ICODE5	NUMBER	17016
NAME5[*]	TEXT	Hoover
SUPPLIER5[**]	TEXT	National

QTY5[***]	NUMBER	2
PRICE5[****]	NUMBER	514.00
TP5[*****]	NUMBER	1028.00
TOTPRICE[+]	NUMBER	15,050.00
PAYMENT*	TEXT	CASH
CC	NUMBER	65102-85524-12065

[*] and [**]:- I have marked some fields as you can see above. These fields have their values looked up from another source. I will do this by selecting the data type as LOOKUP WIZARD, then follow the wizard instructions to give the fields the capability to lookup from another table or to have specific data entered by means of a combo box. In the wizard I can define whether I should type in the wanted values or refer the values from another table.

- If I choose to refer the values from another table the wizard asks which table contains the field I want to refer to. I choose the required table and field and click next. The wizard then asks whether I want to display the key column as well or just display the concerned data. Then it will ask me to give a name for my field.
- If I choose to type the wanted values the wizard shows a table with a single column asking me to type the wanted values. I will type all these values and then click next. The wizard then asks me for a name for the field.

The marked fields are explained below.

[*] These fields have their data looked up from the NAME field of the ITEMS table. These are the settings in the LOOKUP tab of the FIELD PROPERTIES:

DISPLAY CONTROL : Combo box

ROW SOURCE TYPE : Table/Query

ROW SOURCE : SELECT ITEMS.ICODE, ITEMS.NAME FROM ITEMS;

BOUND COLUMN : 1

COLUMN COUNT : 2

COLUMN HEADS : No

COLUMN WIDTHS : 0";1"

LIST ROWS : 8

LIST WIDTH : 1"

LIMIT TO LIST : Yes

[**] These fields have their data looked up from the SUPPLIER field of the SUPPLIERS table. These are the settings in the LOOKUP tab of the FIELD PROPERTIES:

DISPLAY CONTROL : Combo box

ROW SOURCE TYPE : Table/Query

ROW SOURCE : SELECT ITEMS.ICODE, ITEMS.NAME FROM ITEMS;

BOUND COLUMN : 1

COLUMN COUNT : 2

COLUMN HEADS : No

COLUMN WIDTHS : 0";1"

LIST ROWS : 8

LIST WIDTH : 1"

LIMIT TO LIST : Yes

[***]This field holds the quantity of the item purchased by the customer.

[****]This field holds the unit price of the item.

[*****]This field holds the total price of the item. It calculates the total price of the item by multiplying the quantity field with the unit price field.

Validation rules:


FIELD NAME	VALIDATION RULE	VALIDATION TEXT
FNAME	Is Not Null	Enter first name
LNAME	Is Not Null	Enter last name
NAME1	Is Not Null	Enter item name
SUPPLIER1	Is Not Null	Enter supplier name

Input Masks:

FIELD NAME	INPUT MASK	EXAMPLE
ID	99999	52055
FNAME	>L<??????????	Arun
LNAME	>L<??????????	Dsouza
TELEPHONE	"0"9"-9999999	06-5599873
ICODE1	99999	51235
PRICE1	"Dhs. "99999".00	Dhs. 725.00
ICODE2	99999	65741
PRICE2	"Dhs. "99999".00	Dhs. 150.00
ICODE3	99999	56215
PRICE3	"Dhs. "99999".00	Dhs. 240.00
ICODE4	99999	12345
PRICE4	"Dhs. "99999".00	Dhs. 1,350.00
ICODE5	99999	65654
PRICE5	"Dhs. "99999".00	Dhs. 63,000.00
TOTPRICE	"Dhs. "999999".00"	Dhs. 158,000.00
CHEQUE	99999"-99999"-99999	54141-28781-33581
CC	9999"-9999"-9999"-9999	4541-4565-6321-5598

REPAIRS

This table contains the details of customers who have sent in appliances to be repaired. This table is just required to make sure that the appliance is correctly fixed and if not the customer can give his id and get it properly repaired.

FIELD NAME	DATA TYPE	EXAMPLE
ID 	NUMBER	3411
FNAME	TEXT	Salem
LNAME	NUMBER	Khan
ICODE	NUMBER	WEGA T40
NAME	NUMBER	3100
SCODE	TEXT	Sony
SUPPLIER	NUMBER	3840
ITEM TYPE	TEXT	TV
SUBMIT	TEXT	3/14/2003
PROBLEM	NUMBER	No picture, sound distortion
TECHNICIAN	TEXT	Ravi
COLLECT	TEXT	3/20/2003
COST	NUMBER	Dhs. 600

Validation Rules:

FIELD NAME	VALIDATION RULE	VALIDATION TEXT
FNAME	Is Not Null	Enter first name
LNAME	Is Not Null	Enter last name
NAME	Is Not Null	Enter item name
SUPPLIER	Is Not Null	Enter supplier name
PROBLEM	Is Not Null	Describe problem
TECHNICIAN	Is Not Null	Enter tech's name
COLLECT	Is Not Null	Enter collection date

Input Masks:

FIELD NAME	INPUT MASK	EXAMPLE
ID	9999	4740
ICODE	99999	47714
SCODE	9999	3698
SUBMIT	99"-99"-99	15-03-03
COLLECT	99"-99"-99	17-05-02

QUERIES

As I had mentioned in ANALYSE, queries form an important part of filtering data. Through the use of queries the system will be able to distinguish certain data from the whole file. Thus this makes the feature very essential in my system as this feature will allow the users to search for data very quickly. It will also make it easier to produce reports for certain data (which I have mentioned later on). I will design these queries in two forms - direct (the query searches for a particular item only) and prompted (where the user is asked what he wants to search for). I am going to make the queries in design mode as it has more options i.e. I can control the fields to be added, criteria etc. These are the queries I plan to use for my system:

QUERY NAME	CRITERIA	FIELDS SELECTED FOR QUERY
<100	SELL PRICE = <100	ICODE, NAME, SCODE, SUPPLIER, ITEMTYPE, SELL PRICE
>15000	SELL PRICE = >15000	ICODE, NAME, SCODE, SUPPLIER, ITEMTYPE, SELL PRICE
COLLECTION	COLLECT = [Enter date]	COLLECT, ID, LNAME, NAME, ITEMTYPE, TECHNICIAN, COST
REP TECH N SUB	TECHNICIAN = [Enter tech's name], SUBMIT = [Enter submission date]	ID, LNAME, ICODE, NAME, SUPPLIER, ITEMTYPE, TECHNICIAN, SUBMIT
REPANDY	TECHNICIAN = "Andy"	ID, LNAME, ICODE, SCODE, ITEMTYPE, TECHNICIAN, COLLECT, COST
REPARVINDA	TECHNICIAN = "Arvinda"	ID, LNAME, ICODE, SCODE, ITEMTYPE, TECHNICIAN, COLLECT, COST
REPBARRY	TECHNICIAN = "Barry"	ID, LNAME, ICODE, SCODE, ITEMTYPE, TECHNICIAN, COLLECT, COST
REPICODE	ICODE = [Enter ICODE]	ID, FNAME, LNAME, ICODE, NAME, SCODE, SUPPLIER, TECHNICIAN, COST
REPJOSE	TECHNICIAN = "Jose"	ID, LNAME, ICODE, SCODE, ITEMTYPE, TECHNICIAN, COLLECT, COST
RESTOCK	STATUS = "Restock"	ICODE, NAME, SCODE, SUPPLIER, PRESENT, SELL PRICE, STATUS

FORMS

Forms are also important in my system. This will make the system more user friendly as I can customise the fields to be in different positions. This enables me to group specific fields together so that the database is easier to understand. The next few sheets will display my designs of my forms. I will create the forms using the WIZARD. Here I will choose the required fields, the type of view (columnar), the colour scheme and name of the form. After I make the form in the wizard I will tamper around with the design of the form in design mode. My forms will be in the COLUMNAR format as it looks more professional and user friendly. These are the designs of my forms:

In the **MAX** field, I will insert the following code in the **CODE BUILDER**:

```
Private Sub MAX_LostFocus() 'also MAX_onChange() and MAX_AfterUpdate()  
01   Dim maxz  
02   Dim minz  
03   Dim pres  
04   Dim subtr  
05   Dim subtr2  
06   Dim perc  
07   maxz = MAX.Value  
08   minz = MIN.Value  
09   pres = PRESENT. Value  
10   subtr = pres - minz  
11   subtr2 = maxz - minz  
12   perc = Int((subtr / subtr2) * 100)  
13   If (perc <= 2) Then  
14     MsgBox "WARNING! STOCK VERY LOW!!!"  
15     STATUS.Value = "WARNING!!!"  
16   ElseIf (perc <= 25) Then  
17     STATUS.Value = "LOW"  
18   Else  
19     STATUS.Value = "OK"  
20   End If  
End Sub
```

To explain the code, I have numbered each statement. This is the explanation:

- 01 Creates a variable named **maxz**
- 02 Creates a variable named **minz**
- 03 Creates a variable named **pres**
- 04 Creates a variable named **subtr**
- 05 Creates a variable named **subtr2**
- 06 Creates a variable named **perc**
- 07 Stores the value of the **MAX** field in the **maxz** variable
- 08 Stores the value of the **MIN** field in the **minz** variable
- 09 Stores the value of the **PRESENT** field in the **pres** variable
- 10 Subtracts the value of **minz** from **pres** and stores it in the **subtr** variable
- 11 Subtracts the value of **maxz** from **minz**
- 12 Turns **subtr** into a percentage, takes the integer value of it and stores it in the **perc** variable
- 13 Opens an **IF** condition. Checks if **perc** is lesser than or equal to 2. If the condition is true then lines 13 and 14 are executed and it skips to line 20, otherwise it skips to line 16
- 14 Displays a dialog box with the message "WARNING! STOCK VERY LOW!!!"
- 15 Sets the value of the **STATUS** field to "WARNING!!!"
- 16 Checks if **perc** is lesser than or equal to 25. If the condition is true then line 17 is executed and it skips to line 20, otherwise it skips to line 18.
- 17 Sets the value of the **STATUS** field to "LOW".
- 18 This line is executed if the other two conditions are false.
- 19 Sets the value of the **STATUS** field to "OK".
- 20 Closes the **IF** condition

This code is part of my automation of stock control. Should stock start running out, it checks the percentage of stocks left. If this percentage is below 25, the computer considers the stock amount as low. If the percentage is below 2, the computer issues a warning to the user so that he is aware that an item requires restocking. Otherwise the computer considers the stock amount acceptable.

I will add a logo to the forms to make it look professional. Then my next plan is to make a good colour scheme for my forms. This will improve the GUI and make it more presentable. Of course, CODEBUILDER is an important feature as it will help me to program my fields to do some special jobs which could not be done by MS Access alone. Such a case would be storing an addition into both the form and the table, making harder validation rules etc.

RELATIONSHIPS

Relationships between tables are necessary so as to avoid the duplication of data. So I will create relationships between relevant tables. This is a diagram of how my relationship window will look after I am done with creating them:

As you can see, I have 5 instances of the ITEMS table. As you saw in the beginning of this section, I have 5 items that can be stored in the PURCHASE field. I wanted to create a relationship between the ITEMS table and each of the items in the PURCHASE field. In order to carry that out I created 5 instances of the ITEMS table and created a relationship between the ICODE of the ITEMS table and the ICODE(no) of the purchase table.

REPORTS

Reports are a very important feature of MS Access. This feature creates a summary of the data, something like the pivot table feature in MS Excel. I will create my reports using the WIZARD. These are the reports I plan to design:

NO	NAME OF REPORT	EXPLANATION
1	ANDY	Lists all the repairs that Andy will have to carry out with collection dates.
2	ARVINDA	Lists all the repairs that Arvinda will have to carry out with collection dates.
3	BARRY	Lists all the repairs that Barry will have to carry out with collection dates.
4	ITEMS	Lists all the items in the company for statistical purposes and stock control.
5	ITEMS BY ITEM TYPE	Lists all the items in the company by its type. Used for checking suppliers.
6	ITEMS BY SUPPLIERS	Lists all the items by suppliers.
7	JOSE	Lists all the repairs that Jose will have to carry out with collection dates.
8	REPAIR DAY SCHEDULE	Lists the schedule of repairs for a user-defined date.
9	REPAIRS BY COLLECTION	Lists the repairs by the collection date
10	REPAIRS BY TECH	Lists the repairs by the technician
11	SUPPLIERS	Lists all the suppliers dealing with Elektronik(sorted by city)

These are my plans of the designs of the reports:

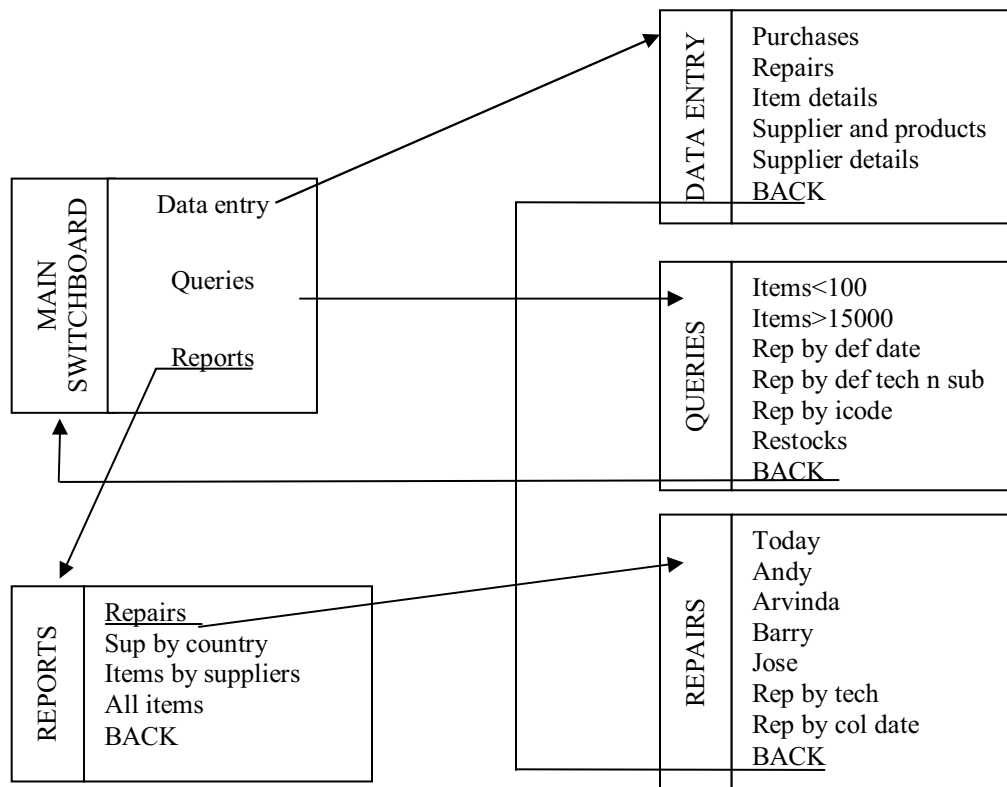
MACROS

I will require macros to open queries through the switchboard. I plan to create the macros in design view because, just like the tables, the macros have extra parameters which can be easily configured in the design mode. These are the macros I intend to create:

NO	NAME	QUERY OPENED
1	<100	<100
2	>15000	>15000
3	COLLECTION	COLLECTION
4	REP TECH N SUB	REP TECH N SUB
5	REPCODE	REPCODE
6	RESTOCK	RESTOCK

SWITCHBOARD MANAGER

The switchboard manager is a very important feature of MS Access. This feature is the starting point of my interface – it will be used as a menu. I am going to create the switchboards using the switchboard manager built in within MS Access. The switchboard manager can be accessed by clicking Tools > Database Utilities > Switchboard Manager. The window which pops up shows all the switchboards made. Here I will click New and another pop-up will open asking for a name for the switchboard. After naming it I will choose the switchboard and click Edit. The window that pops-up displays all the items within the switchboard. I can add new items by clicking New and then choosing the options. This is the layout structure that I plan to design:



The following page shows the basic format of my switchboard:

TEST PLAN

Now that my design plans for the system are complete, I will devise a test plan to make sure that my system works the way I expect it to. Thus I will create a TEST PLAN to ensure that my system will not make any errors which are in my control. The following table is the test plan that I will carry out:

NO	FIELD	TABLE	DATA INPUT	EXPECTED RESULTS
1	NAME	ITEMS	Hoover	ACCEPTED
2	NAME	ITEMS	-	REJECTED : Please enter the item name
3	PRESENT	ITEMS	854	ACCEPTED
4	PRESENT	ITEMS	-	REJECTED : Enter present level
5	MIN	ITEMS	130	ACCEPTED
6	MIN	ITEMS	-	REJECTED : Enter minimum level
7	MAX	ITEMS	450	ACCEPTED
8	MAX	ITEMS	-	REJECTED : Enter maximum level
9	SELL PRICE	ITEMS	1,500.00	ACCEPTED
10	SELL PRICE	ITEMS	-	REJECTED : Enter selling price
11	FNAME	PURCHASE	Salem	ACCEPTED
12	FNAME	PURCHASE	-	REJECTED : Enter first name
13	LNAME	PURCHASE	Khan	ACCEPTED
14	LNAME	PURCHASE	-	REJECTED : Enter last name
15	NAME1	PURCHASE	Hoover	ACCEPTED
16	NAME1	PURCHASE	-	REJECTED : Enter item name
17	SUPPLIER1	PURCHASE	National	ACCEPTED
18	SUPPLIER1	PURCHASE	-	REJECTED : Enter supplier name
19	QTY1	PURCHASE	2	ACCEPTED
20	QTY1	PURCHASE	-	REJECTED : Enter quantity
21	PRICE1	PURCHASE	1,300.00	ACCEPTED
22	PRICE2	PURCHASE	-	REJECTED : Enter rate
23	FNAME	REPAIRS	Salem	ACCEPTED
24	FNAME	REPAIRS	-	REJECTED : Enter first name
25	LNAME	REPAIRS	Khan	ACCEPTED
26	LNAME	REPAIRS	-	REJECTED : Enter last name
27	NAME	REPAIRS	Hoover	ACCEPTED
28	NAME	REPAIRS	-	REJECTED : Enter item name
29	SUPPLIER	REPAIRS	National	ACCEPTED
30	SUPPLIER	REPAIRS	-	REJECTED : Enter supplier name
31	PROBLEM	REPAIRS	No sound,...	ACCEPTED
32	PROBLEM	REPAIRS	-	REJECTED : Describe problem
33	TECHNICIAN	REPAIRS	Andy	ACCEPTED
34	TECHNICIAN	REPAIRS	-	REJECTED : Enter tech's name
35	COLLECT	REPAIRS	15-05-03	ACCEPTED
36	COLLECT	REPAIRS	-	REJECTED : Enter collection date
37	SUPPLIER	SUPPLIERS	National	ACCEPTED
38	SUPPLIER	SUPPLIERS	-	REJECTED : Enter supplier name
39	TELEPHONE	SUPPLIERS	(098) 684 5841	ACCEPTED
40	TELEPHONE	SUPPLIERS	-	REJECTED : Enter telephone number

USER FEEDBACK

Now my design plans are complete. Before I began my practical work, I decided to have the manager evaluate my progress, so that I would not make a mistake and regret it later. A few days later I went to the manager and asked his opinion after showing him all the designs that I proposed for the system. The manager said it was "well done" but one important thing was missing - the mail merge feature. Also, the manager wanted me to change the structure of the PURCHASE form so that it would look like a receipt with details. So then I started working on the problems. To send letters to the suppliers I need information from both the SUPPLIERS table and the ITEM PROPERTIES TABLE. At first I thought of making a table but after some time it struck me that I could create a query that would link the two tables together. So now I plan to create the query in the following way:

QUERY NAME: MAILMERGE

CRITERIA: (none)

FIELDS SELECTED: ICODE, NAME, ITEM TYPE, MAX, STATUS, SUPPLIER (ITEMS table), SCODE, ADDRESS, STATE, COUNTRY, P/ZCODE, SUPPLIER (SUPPLIERS table).

Then this will be my format for my standard letter:

To:

The manager,

<<POBOX>>

Dear Sir,

Elektronikz has reached its stocking date and is now currently restocking all goods from various countries. In order to replenish our stock levels, we require the normal level (<<MAX>>) of <<NAME>> which is supplied by your company, hopefully reached to us latest by the 7th of April 2003. Please ensure that the goods delivered are in good condition and payment will be made on the 10th of April personally by the manager.

Thank you,

Mr. Nathan Fernandes,
GENERAL MANAGER OF ELEKTRONIX

Now that the mail merge has been handled, I will create the new design of the PURCHASE form. The following is my new planning, and as shown, it looks like a receipt:

Now that my designs have completely been prepared, I will start implementing the new system.