CHAPTER

16

Crystal and Data Reports

Crystal Reports

The development of the Invoicing program has so far focussed on controls, tables, databases and the code. What we have not worked on so far is the Reports. The user of our application would definitely want to know a number of details about the state and status of his business. For example, the total number of items sold in a day, week, or month. Or the list of items supplied by a Supplier for a given month. Or the payments due with amount and date. A businessman will need to know a lot more than this. Our application must provide the means for him to get the all the information possible based on the data provided to the application.

We can build this functionality into our application with the help of VBA code or through report generators like Crystal Reports or Data Report.

Let us first work on Crystal Reports. Crystal Reports is a third party product developed by Seagate of Singapore. It has been bundled with various data access tools. It can work with Visual Basic on your PC or with the AS/400 as well.

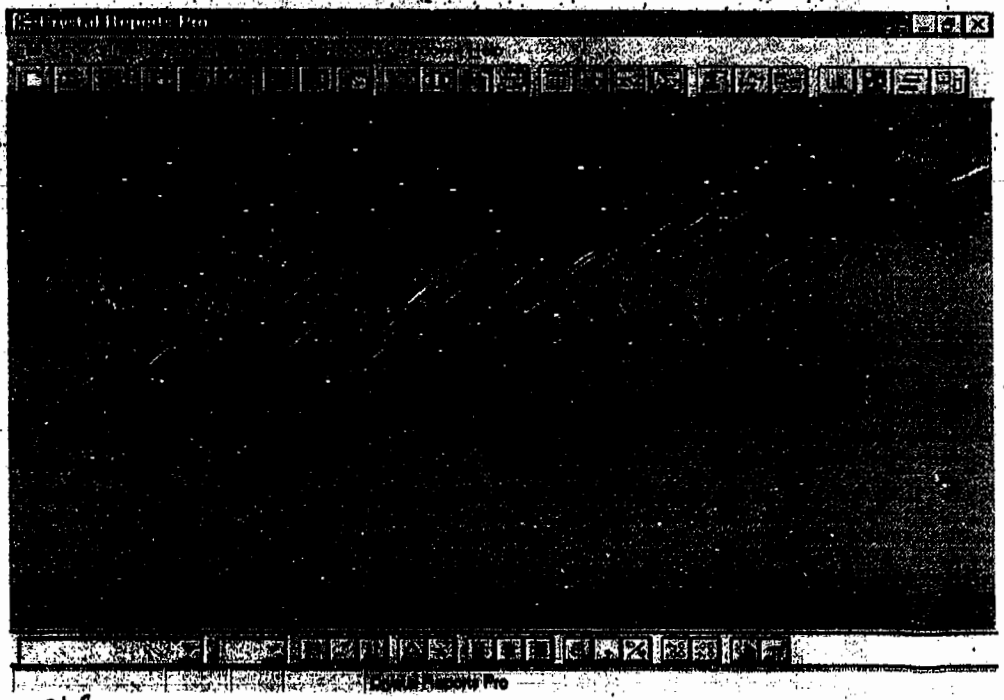
Prerequisites for working with Crystal reports

Hardware A printer must be installed. It need not be physically connected though. This is because Crystal Reports builds the reports based on the properties of the printer.

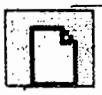
Application You can access Crystal Reports only through the VB IDE. If Crystal reports has not been installed then follow the steps given below. Prepare a pencil copy of the report structure that you want to create.

Installation So, Crystal Reports has been installed and you have a rough 'copy' of the report that you want. Let us get started. Click on Add_Ins. Select Report Designer. (For some reason

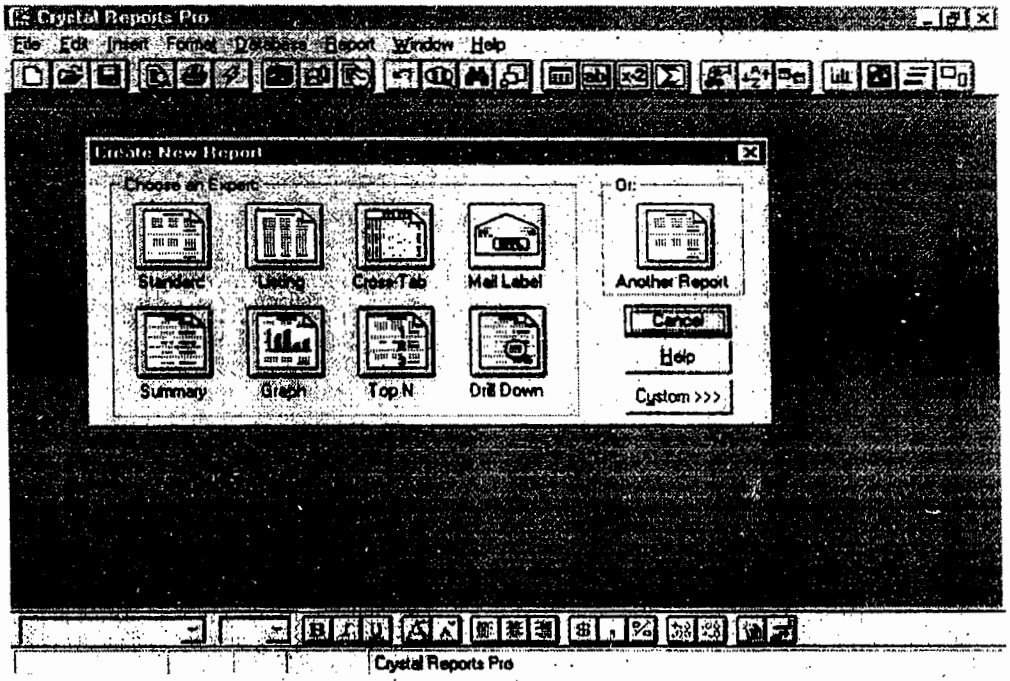
best known to Microsoft, Crystal Reports is not mentioned by its original name.) You will see the following figure. If you have not registered your name so far do so now.



Click on File and Select New. Or click on the icon (in picture) that represents a new report.



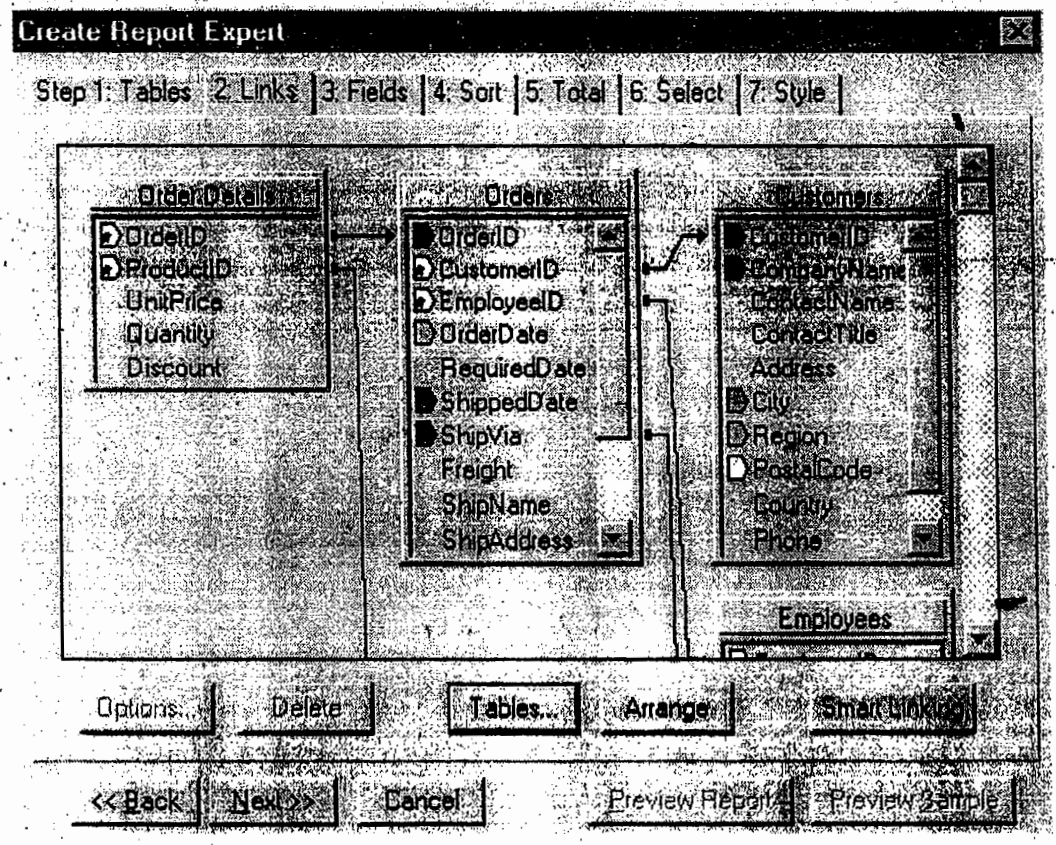
You will see the following figure.



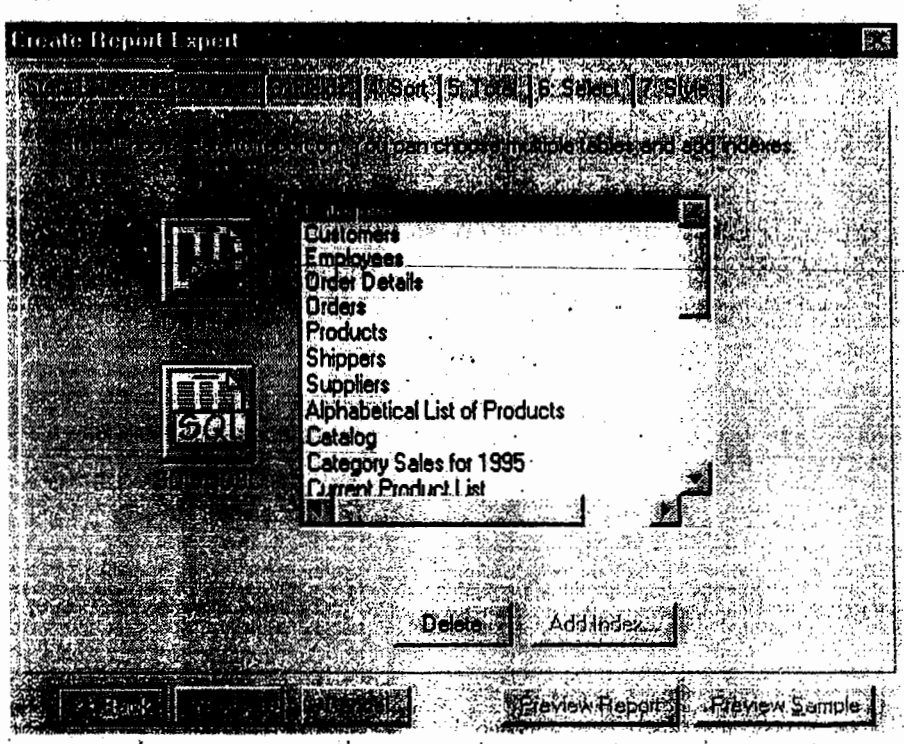
Creating a Report through a Wizard

You are presented with the Create New Report Wizard that will allow you to create new reports. You can choose the report style. For this exercise, click on Standard. The wizard will now take you through a number of steps asking you for details. Supply the details to the best of your knowledge. If even you make an error the reports can be corrected later. Remember that right from the first step you are given an option to preview the report that you are creating. Avoid the temptation to see the report. Once you do that you cannot come back to the wizard. Should you wish to return to the wizard, you will have to start all over again. We will see the various steps required to create a Standard Report and then call the 'Expert' for a particular step.

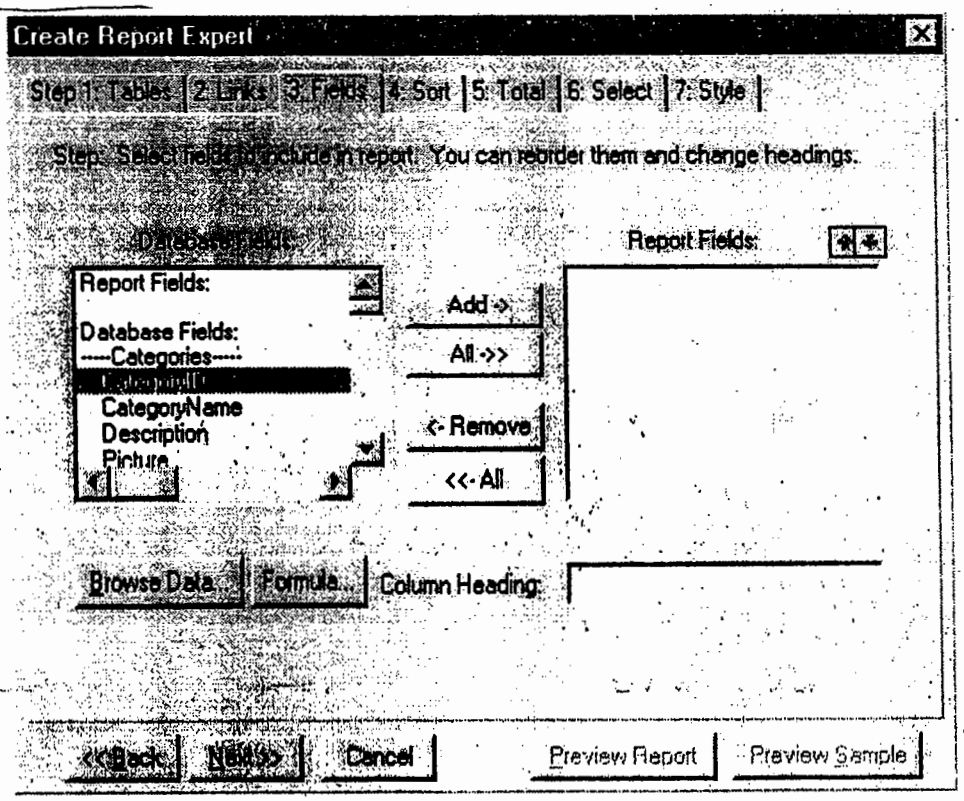
The wizard in the dialog box will ask you to select the database(s) that you will be using to generate the report. Let us select Nwind.mdb for a change. (You will be generating your own reports using the Invoice.mdb). Upon selecting the Nwind.mdb, all the tables and stored queries/view will get added to the ListBox. After you have added all the databases that you want to work on click on 'Done'. You will see the following figure.



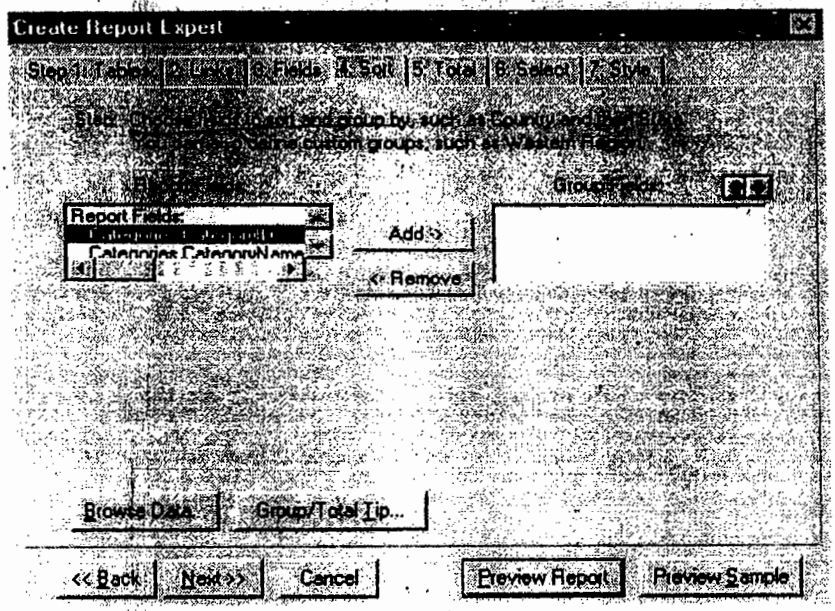
This figure will display the various tables and the relationship between each of them. If you think there are too many tables and views and you do not need all of them, then you can delete some of them. Click on the button 'Back'. You will see the following figure.



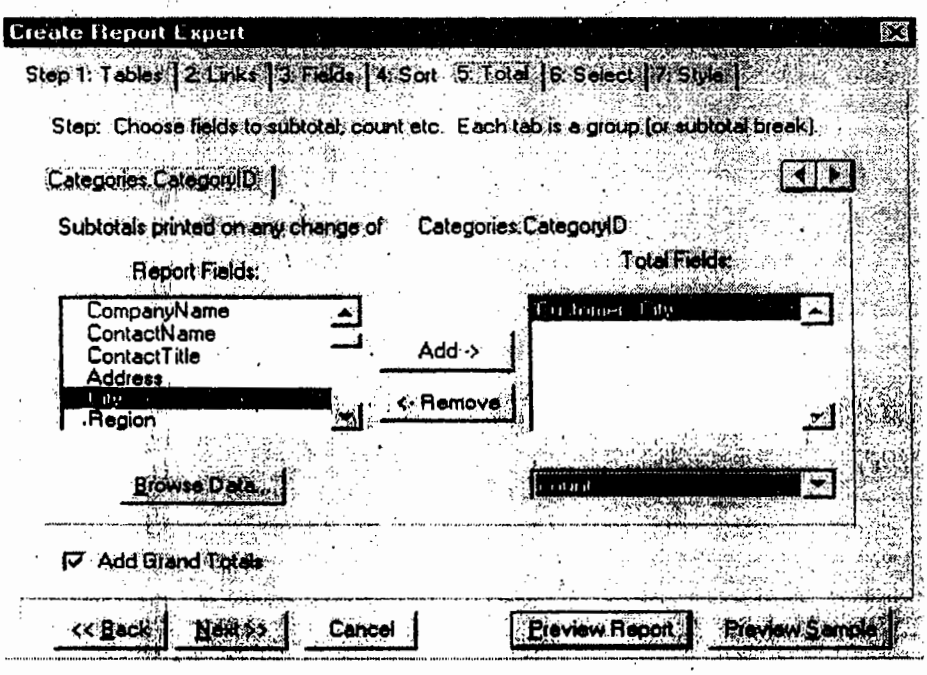
You can select the items that you do not need and click on remove to remove them one by one. When you are sure you have only those tables that you need click Next to continue. You will see the figure with the selected tables and their relationship. Click Next to continue. You will see the following figure.



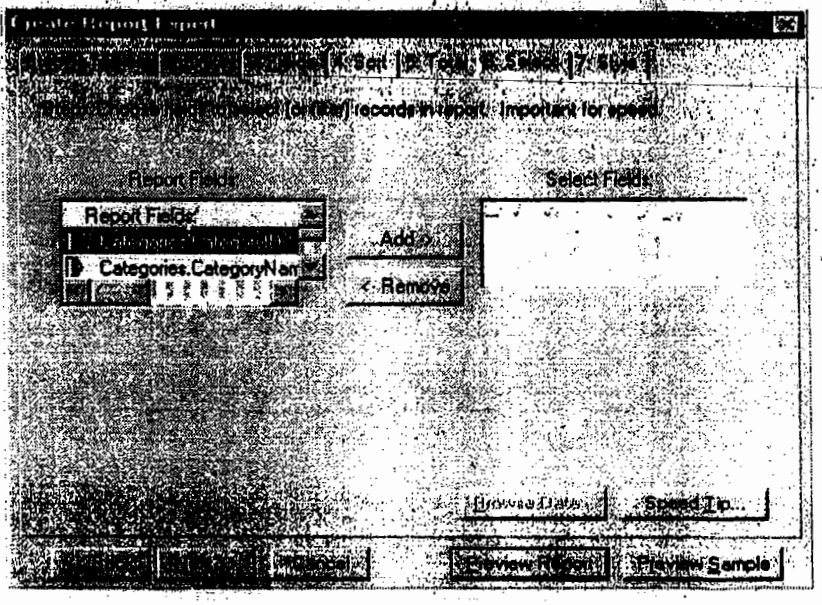
In this dialog box you can add the fields that you wish to include in your report. The fields you select here will appear on the report. However the selection criteria for the selected records need not depend on these fields alone. When you have selected the fields and added them one-by one in the 'Report fields' ListBox. Click on Next to continue. You will see the following figure.



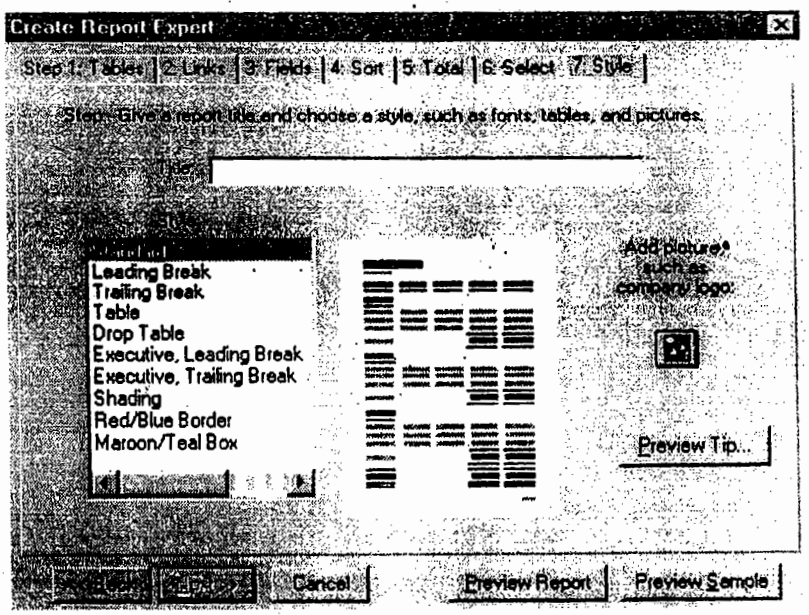
In this dialog box you can choose the fields on which the report is to be sorted out. For example, you can sort all the details based on the City, or the Product that a customer uses, or the Turnover of the company, etc. Select the fields on which the criteria is to be built and then select the sort order. For example, you can sort the details in the ascending order or descending order. When you are through with this click Next to continue. You will see the following figure.



In this dialog box you must select the fields on which you have to perform calculations l group total, sub-total, etc. For example if you want to know the number of customers i particular city, then select Customer_City and add it to the 'Total Fields' ListBox. Here you also choose if you want to total the number of customers for a city or if you want to the add figures for a particular column. For our example choose Count. Then Click Next to conti You will see the following figure.

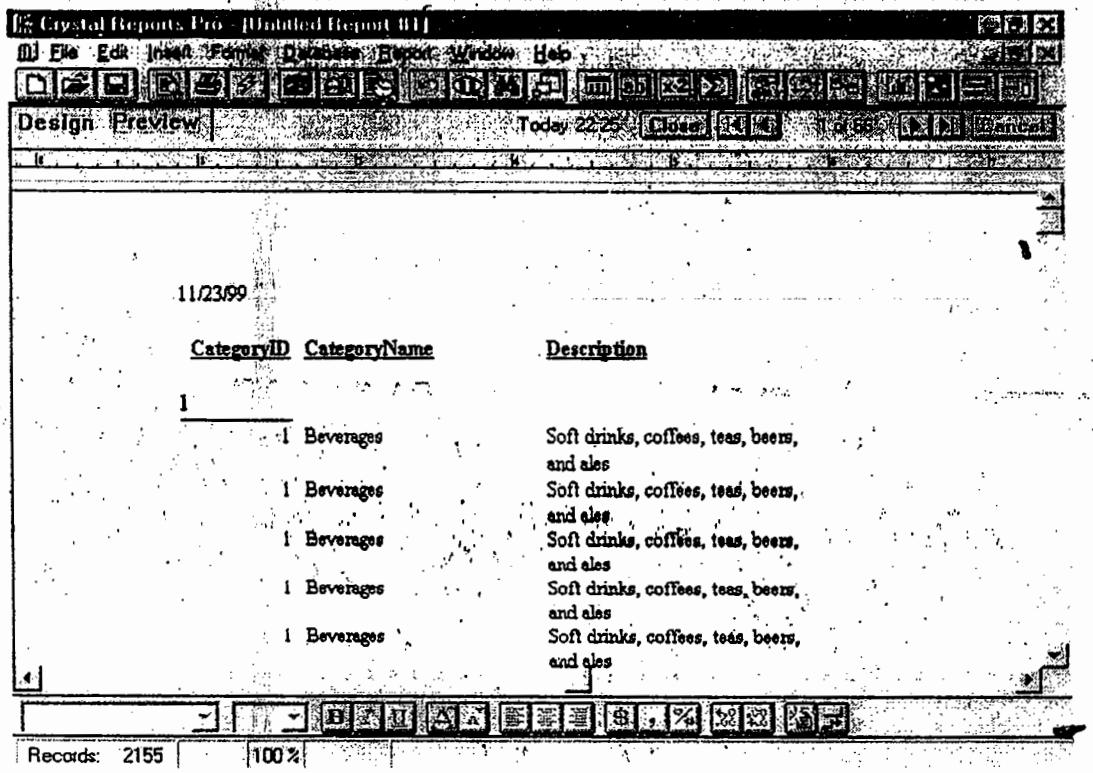


In this dialog box you must choose the fields based on which the records must be selected f the database. In the Report Fields ListBox you are presented with the fields that you have lected for the report. If none of these fields meet your requirements to determine the selec criteria, you can scroll down further and select from the fields that have not been included in report. Build your selection criteria and click Next to continue. You will see the following figu



In this dialog box you can select the layout of the report. Select the report layout style that think suits you best. The selection of the style will depend upon the type of data that you likely to have on the report. For example, if you are going to have the total amount outnding from a customer, and your report will hold the status of customer for a city or area en you can choose Trailing Break Style or the Drop Table style. I suggest that you preview the rious types of reports possible before settling for a particular style.

Next you can preview the report. The preview of the report will look like this



Ok! So we have created a report from scratch with the help of a Wizard or and Expert. Let see how we can create a report without the help of a Wizard. Remember that the steps quired to create a report will be more or less the same.

eating a Report without a Wizard

t us create a sample database file with two tables for this exercise. The name of the .mdb file be any name of your choice (follow the naming conventions). We will call it Test.mdb.

We will work with two tables. The Customer_Data and the Order_Data tables. (In case you not have such a database file with such tables create them for the sake of this exercise).

The Customer_Data table has these fields

Customer_Code
Customer_Name
Customer_City

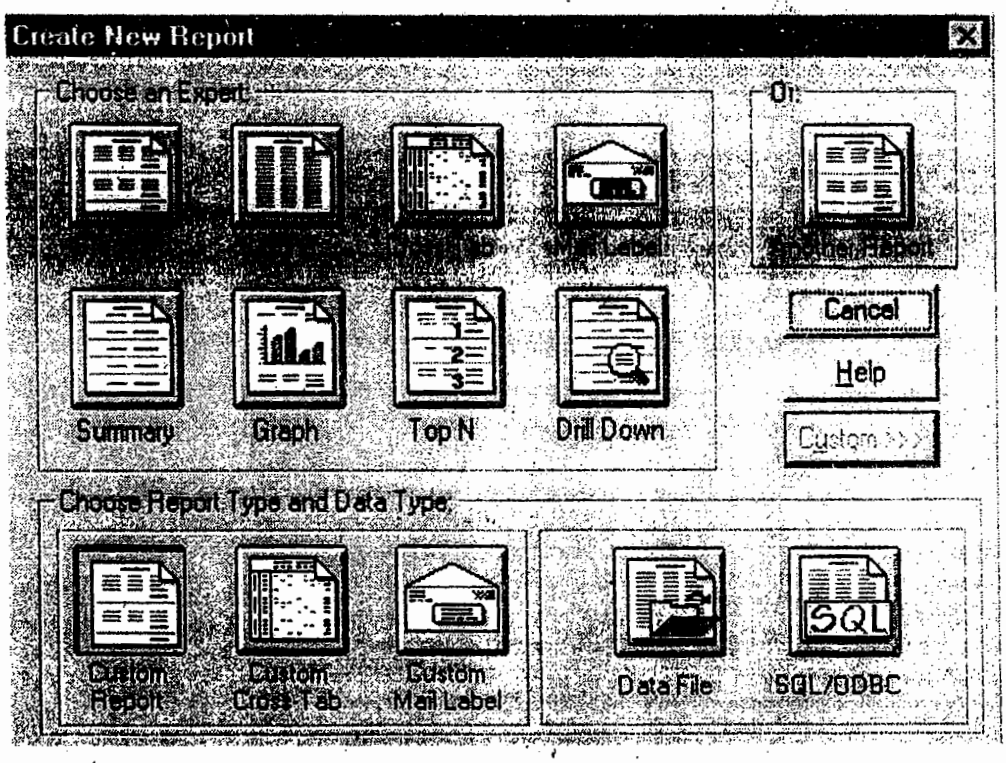
The Customer_Data table is indexed on Customer_Code as the primary index.

The Order_Data table has these fields

Order_ID
Customer_ID
Order_Value
Order_Date

The Order_Data table is indexed on Order_ID as the primary index and Customer_Code the secondary index.

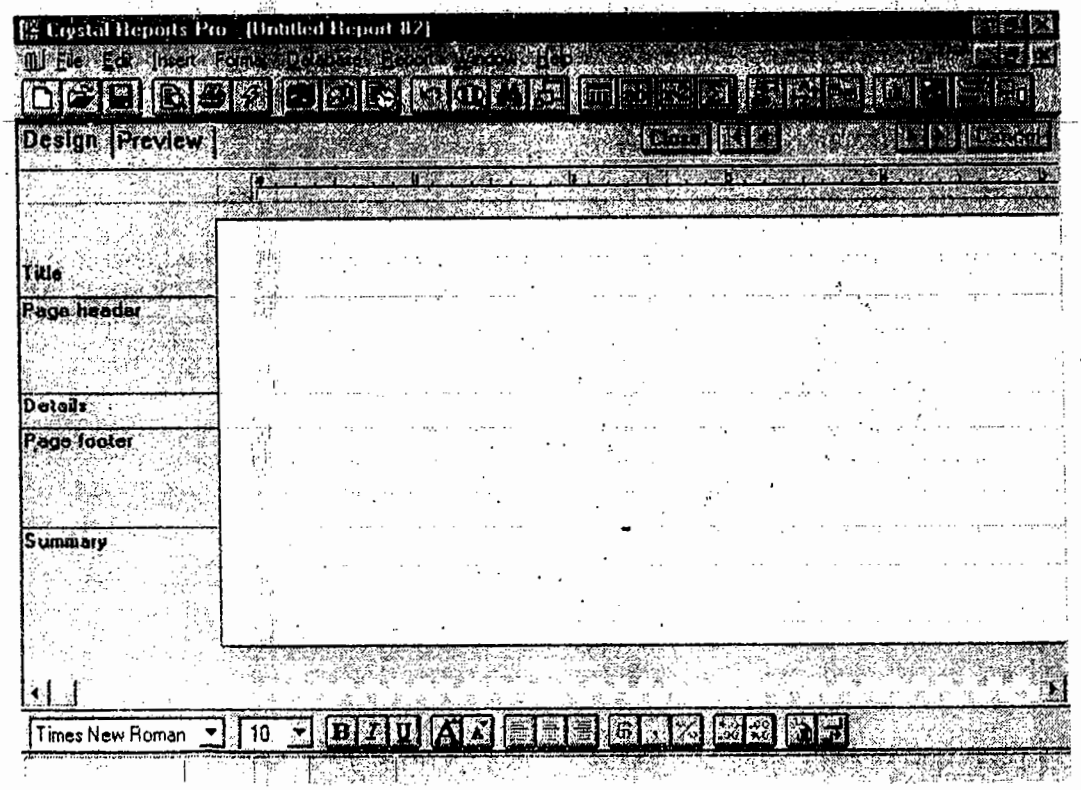
Click on the New File icon or select New from the File Menu. You will see the figure that saw in the last session. Instead of selecting Standard, click on 'Custom'. An extended set buttons will be displayed.



From the extended set of buttons displayed select Custom and then click on the Data button.

You will be asked to select the database file. Select Test.mdb in order to complete this ample. (In case you do not have such a database file with such tables create them for the sa this exercise).

Next you will see a figure like one as shown in the next page.



This is called the Design/Preview window. Here you can design and view the report as you go on adding fields to it.

The Design/Preview Window

This is the window where you design your reports. The default tab is the Design tab. This window is divided into two parts. The Large white area is where you actually insert the fields and the gray area on the left side with several rows. The rows will be titled as Title, Page Header, Details, Page Footer and Summary. Each of these titles is separated by a line that extends into the large white area. This is to help you correctly insert and correct the data that should appear on the report.

If you click on the Preview tab, a preview of the report will be displayed. You can zoom to get better view of the report. The button to zoom is 

In the figure above, the Insert Data Field dialog box displayed over the design window consists of the list of fields that you can display on the report. Apart from the list box there are three buttons "Insert", "Done" and "Browse Field Data". To insert a field select it and push the Insert button. Your mouse pointer will suddenly acquire a square tail. This is tells you that you have selected a field and you can insert it at a location on the report.

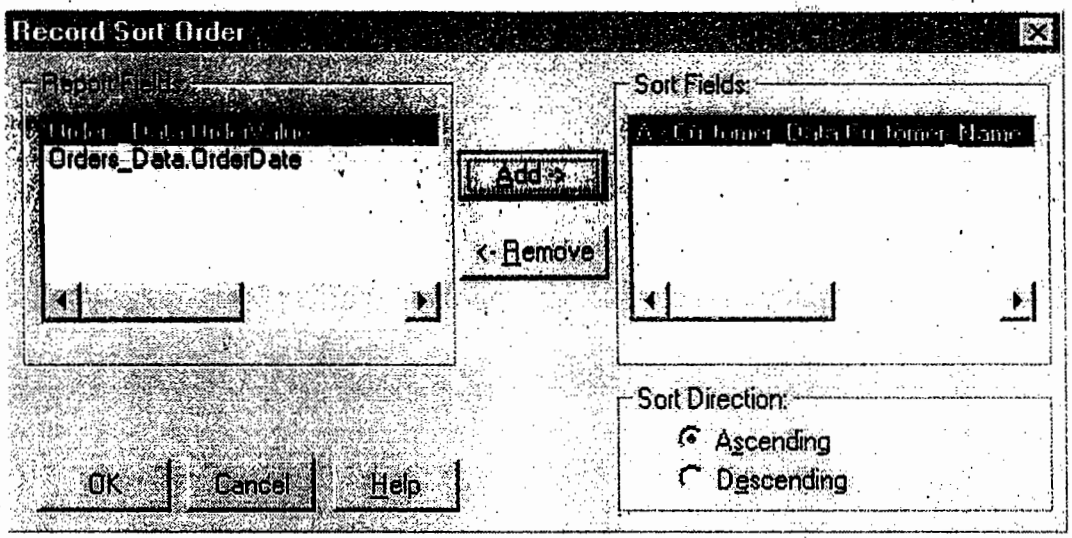
Move the pointer to the location where you wish to display the selected field and click on the mouse. The field will get inserted at that location.

Let us get back to the Insert Data Field Dialog box. Select Customer_Name from the list of fields displayed in the ListBox. Click on the 'Insert' button. Notice that your mouse pointer has acquired a 'tail'. Move the mouse pointer to the section 'Details' on the 'White Area' and click the left mouse button. The Customer Name field gets placed there along with the heading. You can change the column heading later if you do not like the current heading. Similarly add two more columns namely Order_Date and Order_Value to the report.

Click on the Preview tab to view the report. You will see a list of the customers with the Order Date and Order Value. Some of the problems with this report are:

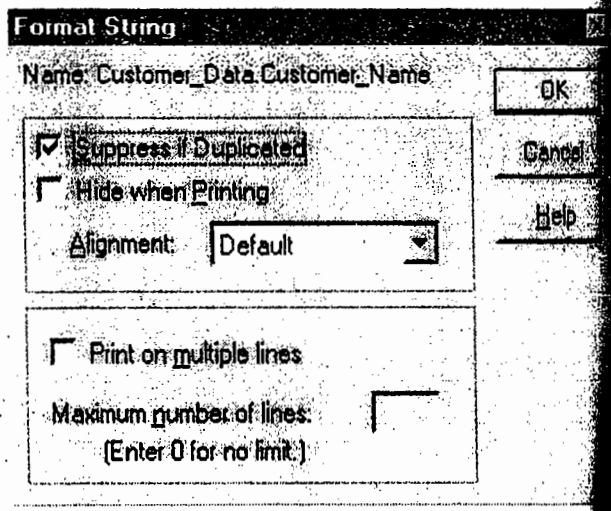
1. The customer names are repeated.
2. There are no sub-totals for individual customers.
3. There is no grand total of the Order Value.

Let us see how we can correct them. We are at liberty to call on the 'experts' for help. We will first sort the listing on Customer_Name. Click on the 'Reports' Menu Option. From the menu items select 'Sort Records'. You will see the following dialog box.

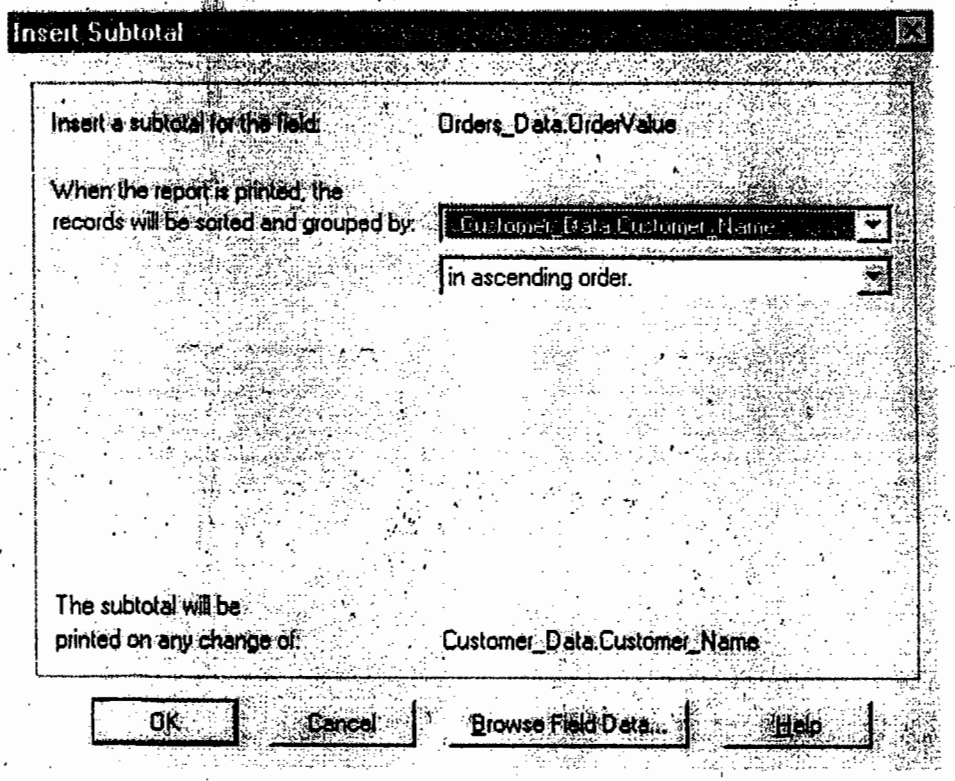


From the list of items in the Report Fields ListBox on the left add the Customer_Name to the Sort Fields ListBox on the right. Select the Ascending (default) order for sorting the records. Now Click the Preview tab to see the result.

To avoid repeating the Customer_Name (or for that matter any field) for every occurrence, click on Format. From the menu click on 'Fields'. You will see the following dialog box.



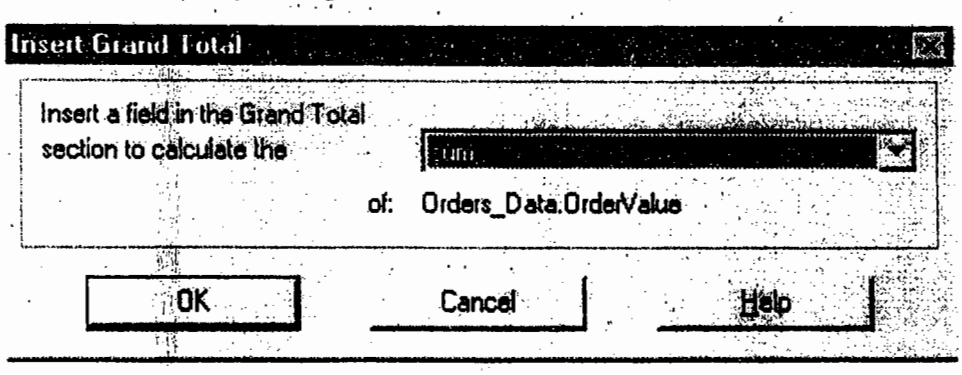
Click on the 'Suppress if Repeated' option button. To include Sub-Totals for the Order_Value each customer, right click the mouse on the Order_Value in the Details section. From the up menu select 'Insert Sub-Totals'. You will see the following dialog box.



The message displayed will be "When the report is printed the records will be sorted and uped by". Next to this message is a drop-down combo box. Click on the down arrow to view list of fields that you can group by. Select Customer_Data Customer Name. A message will ear at the bottom of the dialog box. "The subtotals will be printed on any change of tomer_Data.Customer_Name".

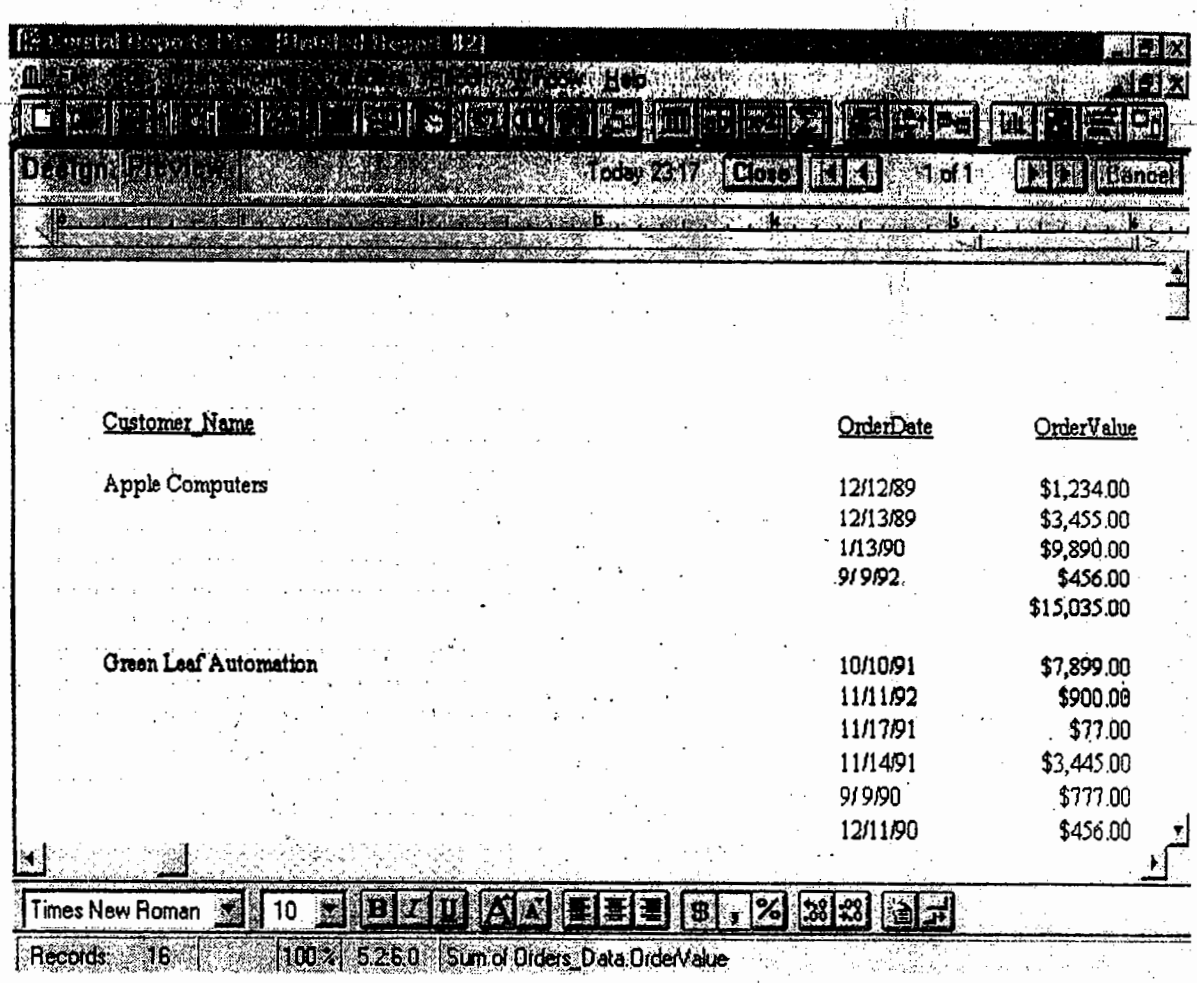
Click OK to accept the changes.

To add the Grand Total to represent the total of all Order_Values, once again right click the se on the Order_Value in the Details section. From the pop-up menu select 'Insert Grand l'. You will see the following dialog box.



Select 'Sum' from the drop-down ComboBox. You have a wide choice of options to choo from for the Grand Totals column.

Now preview your report. Your report should look like this.



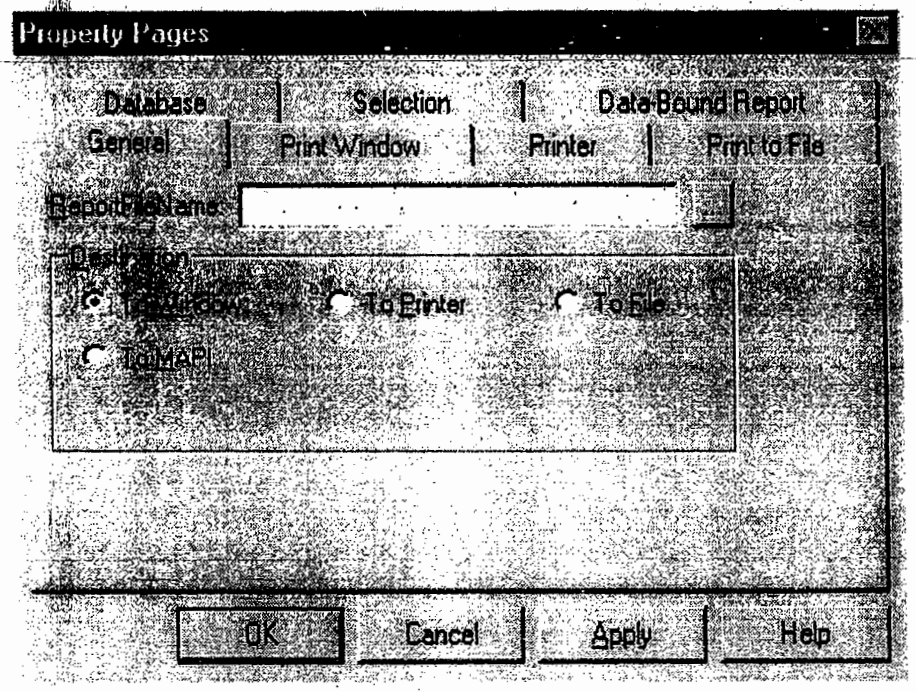
Save the report in your directory. It will have an extension name ".rpt".

Calling Crystal reports from Visual Basic.

So you have learnt how to create a report using Crystal Reports. We need to be able to the report that we have created from our application. In order to be able to call Crystal Repo from our program we must first add the Crystal Reports control to our program.

Start a new project. The Crystal Reports control should be added to the form from wh you intend to call the report. You will not find the control on your toolbox. To add it to y toolbox right click the mouse on the toolbox and select Components from the pop-up me From the list of components select Crystal Reports Control 4.6.

Add an instance of the Crystal Reports control on your form. Since this control is not vis during runtime it will appear as an icon at design time. Right Click on the icon and se Crystal Properties from the menu. The property pages will appear as follows.



The most important property to set here is the ReportFileName. This is set in the General Tab. Set it to the filename under which you saved your report. Under this tab you can also specify if your report should go to the printer or to the window or to a file.

Add a CommandButton to your form. In the click event of the CommandButton add the following line of code

```
CrystalReport1.Action = 1
```

If you are going to use more than one report file in your program then you need to set the ReportFileName before you run the report.

You see the ReportFileName as follows

```
CrystalReport1.ReportFileName = "C:\azam\vb-exersices\crys1.rpt"
CrystalReport1.Action = 1
```

Most of the properties that you see in the Property pages are available at runtime and can be modified when needed. You can for example use a selection formula to select the records that appear on the report.

If you want the report to display selected records, for example you want to see details of the customer whose Customer_Code is C455 then add the following line in the SelectionFormula ListBox under the Selection Tab.

```
{Customer_Data.Customer_Code } = "C455"
```

Run the program and you will see the details of customer C455 only.

Data Report

In the last session we saw how to use Crystal Reports to create reports. In the following session we will take a look at Data Report. Data Report is the new offering from Microsoft perhaps with a view to replacing Crystal Reports in the long run.

Data Report as it stands today is meant for programmers. A general user of computers will not be able to get around it. Let us take a look at what Data report has to offer and how we go about using this tool

In order to use Data Report you need to use ADO or Data Environment. Since we have covered ADO in the previous chapter we will work using ADO with Data Report.

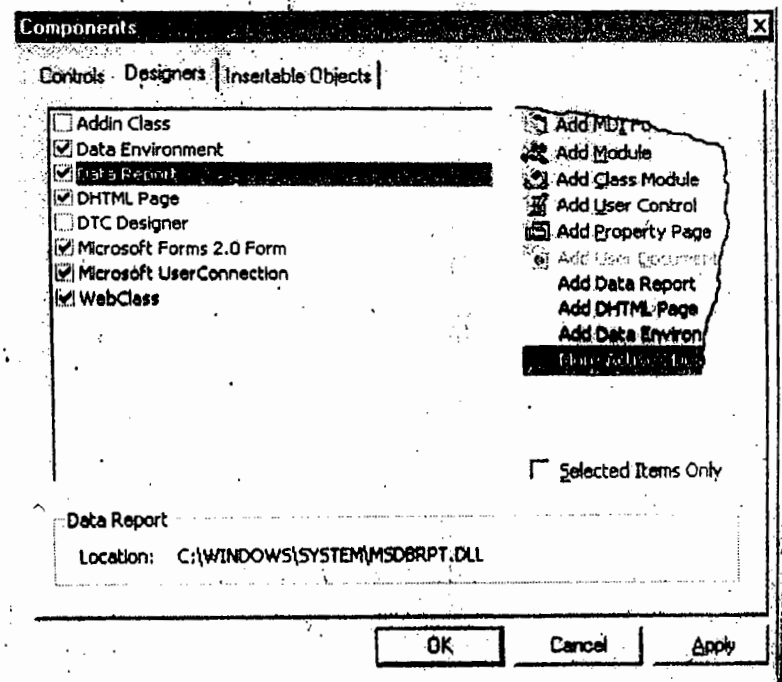
You need to follow the following steps to generate a report using the Data Report.

1. Create a data source using ADO.
2. Add the Data Report object to your project.
3. Place texboxes representing the various fields that you want on the DataReport object.
4. Link the Textboxes to the various fields of the data source
5. Display the report using the Show method.

This sounds like quite a bit of work. Actually it is quite easy if you have already worked with the Data Control or the ADO Data Control. The approach is very similar and you will find it fairly simple unless you are a die-hard fan of Crystal Reports.

Getting acquainted with the Data Report Designer

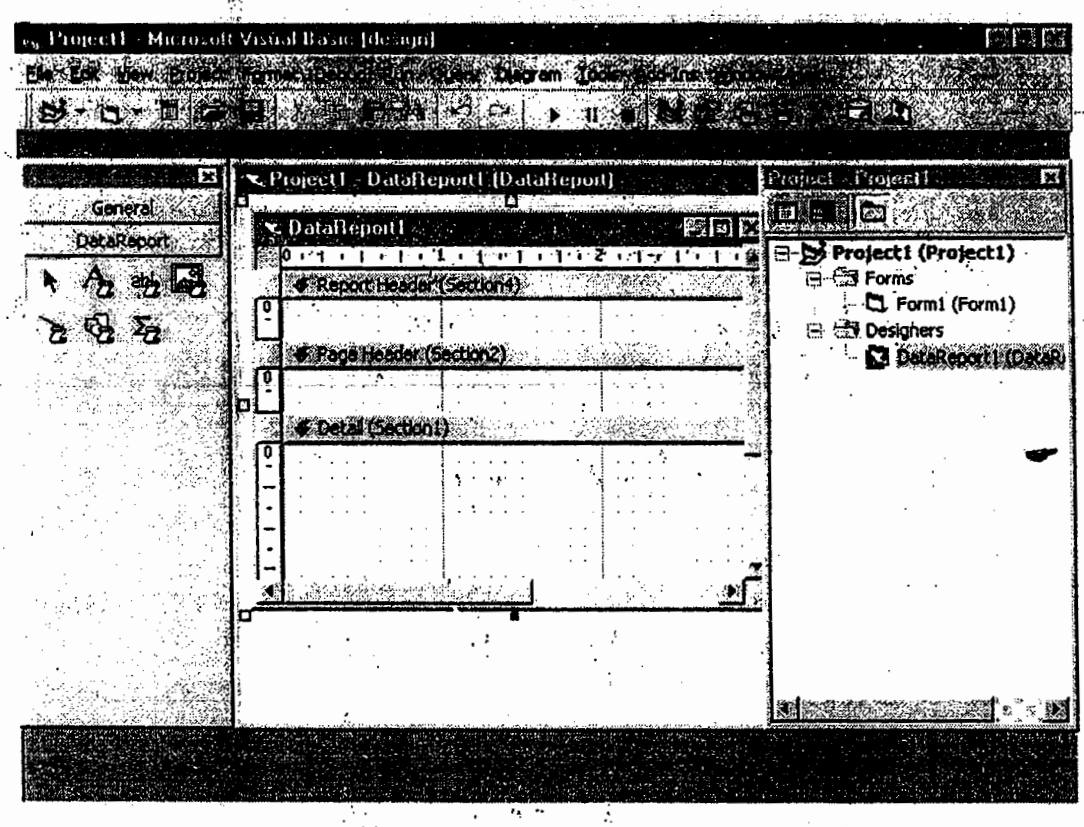
The Data Report Designer is not part of your toolbox. To add it to your toolbox, right click on the toolbox and select Components from the pop-up menu. On the Component dialog box, click on the 'Designers' tab and select 'Data Report'. Close this dialog box.



Let us take a look at the DataReport object. This will give us an idea about the approach to be taken for generating a report.

The Data Report Designer is the form on which you design the layout of the report. The DataReport object is the programmable object that represents the Data Report Designer.

Click on 'Projects' In this menu you will see a new item 'Add Data Report'. Select this item to add a Data Report Designer to your IDE. Please remember that this designer will not be placed on your form. The Data Report Designer is a separate form by itself. Open the Project Explorer and you will see another item called DataReport1 along with the Form1. Also notice that your toolbox has acquired a tab called Data Report with its own set of tools.



Parts of the Data Report

The Data Report consists of three main components.

Data Report object

Section object

Data Report Controls

You need a form or the area where you arrange the various fields that you want to display. This is the Visual Designer component of the DataReport object. The designer component can be controlled programatically using the DataReport object.

The Data Report Designer consists of a number of Sections like the header, footer, and details sections. Each of these Section objects can be configured at design time or controlled through code at run time. Each section has a set of properties that can be manipulated.

Finally you have the Data Report controls, which are special control that you can create on the Data Report designer. These tools are placed under a separate tab on your toolbox.

The default Data Report designer contains these Sections

- **Report Header:** You give the title of the report in this section. If the first page of the report should contain only the title, then set its ForcePageBreak property to rptPageBreakAfter.
- **Page Header:** You give the page heading here.
- **Group Header/Footer:** you give the heading for every group here. For example your report can contain details of Customer for each city here. So the Group Header can be the name of the city. A group header must also have a Group Footer.
- **Details:** This section contains the actual data. The records are displayed in this section.
- **Page Footer:** You give the page footer here.This can be the page number or any relevant text like the date of report, etc.
- **Report Footer:** You give the summary for the report in this section. This can contain the address, the bibliography, contact address, etc. The Report Footer appears between the last Page Header and Page Footer.

Data Report Controls

Following are the new set of controls that are placed under the Data report tab on your toolbox.

- TextBox Control (RptTextBox)— To display text or other formatted data.
- Label Control (RptLabel)—To display the labels on the report to identify fields or sections.
- Image Control (RptImage)—To display pictures on the report. This control cannot be tied to a data field.
- Line Control (RptLine)—To draw lines on the report.
- Shape Control (RptShape)— To draw rectangles, circles, etc on the report.
- Function Control (RptFunction)—This is a special text box that calculates values as the report is generated.

Extracting the Data

We saw earlier in this session that we must first create a data source using ADO. Let us do that now. We will use our Invoice.mdb. Let us create a list that consists of CustomerName, Customer_City and Order_Value. We need to work with two tables

```
Dim adocon As ADODB.Connection
Dim adors As ADODB.Recordset
```

In the above code fragment we have declared an ADO connection and an ADO Recordset object.

In the Form load event add code to create the connection and then to create a recordset. The following lines of code will do the trick

```
Set adocon = New ADODB.Connection
adocon.Open "DSN=Invoice"
```

In the form load event itself you can populate the recordset. But this is not such a good idea. If you are not going to view/display the report then the recorset is unncecessarily taking up memory. In order to avoid this add a button to your form. Let the caption be "Display". In the code window of this CommandButton, enter the following code.

Set adors = adocon.Execute("Select distinctrow Customer_data.customer_name, Customer_data.address2, orders_data.ordervalue from Customer_data, orders_data where Customer_data.customer_code = orders_data.customercode")

The above code segment will populate the Recordset.

We now have data readily available. The Recordset will have all the fields from both the tables. Next we have to display the fields that the user is supposed to view.

Working with the Data Report In the details section of the Data Report designer, add three of the RptTextBox controls. Notice that it is just like adding ordinary textbox controls. Also observe that these textboxes contain a caption called 'Unbound'. This means that these RptTextBox controls are not bound to any data source or data field.

Binding the RptTextBox to a data field: Bring up the Properties window of the RptTextBox by pressing F4. Enter the name of the field that you want to display against the Data Field property.

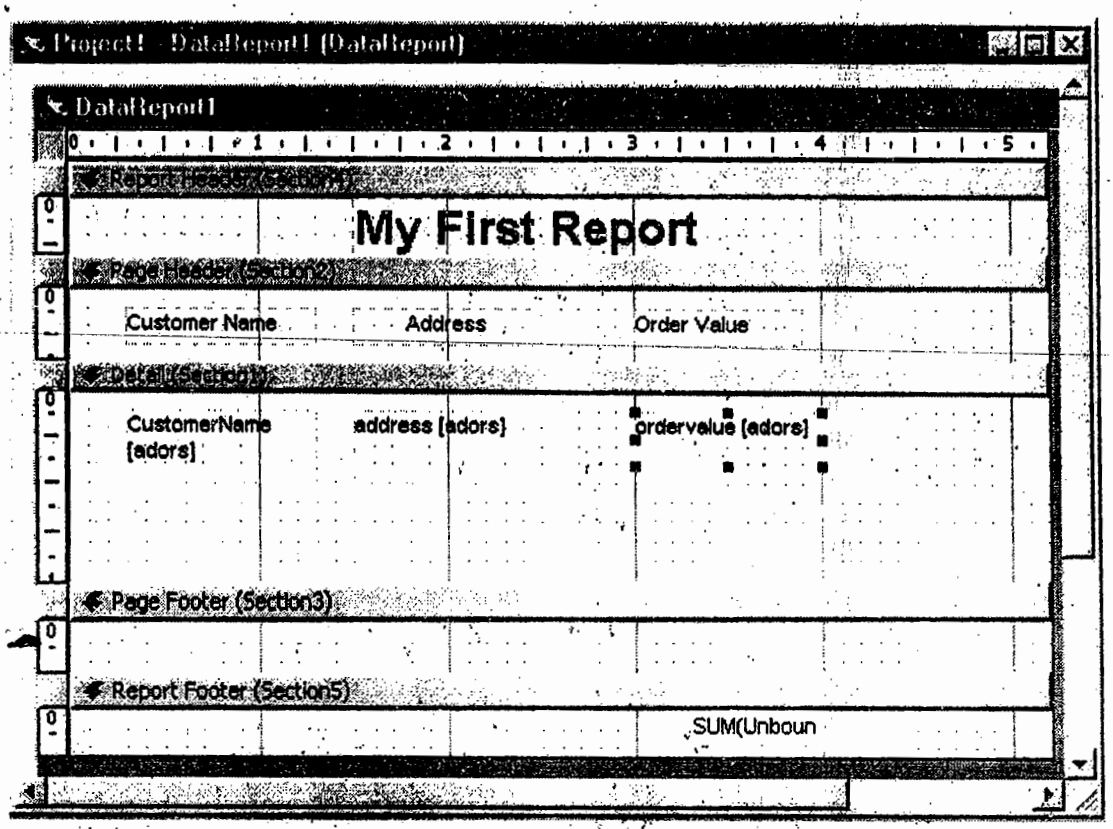
For our example, the three RptTextBox controls will display the "Customer Name", "Customer City", "OrderValue". Enter these field names as they are in the database. If they are wrongly spelt you will get an error message.

Your Data Report designer will look like the figure in the next page.

Displaying the report We are now ready to display the data. We have created the recordset. We have assigned the fields in the Data Report Designer. We need to link the recordsource to the Data Report. Then we must call the Show method of the Data Report. The following lines have to be added to the "Display" command button.

```
Set DataReport1.DataSource = adors
DataReport1.Show
```

Run the program now.



Exercise

Add headers to the report using the RptLableBox controls so that the report looks meaningful.

Creating Multiple Reports

The users of your application are not going to be happy with just one report. They will want the data presented in many and every combination. We can cater to this as well. There are two ways of handling the 'multiple reports' situation. For every report that the user wants, we can create a DataReport or we can display different data using only one or a few DataReport forms. Both the approaches have their merits and demerits. We have already created one report using one DataReport. Creating more reports using the same method should not be very difficult. However displaying more than one report using only one DataReport involves a little work.

Details like Caption, Page Headers, Footers, etc for each of the reports must be determined. The heading for the data must also be determined. The data and the source of the data must also be worked out. Depending upon the number of reports that you may need to display on a form, you have to work out if it is feasible to create a recordset or a number of recordsets for all the reports. Creating a recordset every time the user asks for a report may not be a good idea especially in a multi-user environment. At the same time creating a large number of recordsets and locking up resources will not be the right thing to do.

Let us use the DataReport1 to create another report. In this report we will display the Customer Name, Customer Address and the Order Value. Add another button to the form. You can give an appropriate caption. We suggest you give a caption like "Report2". In the code window of this button, add the following code.

```
' Creating the Recordset.
Set adors = adocon.Execute("Select distinctrow Customer_data.customer_name,
Customer_data.address2, orders_data.ordervalue from Customer_data, orders_data where
Customer_data.customer_code = orders_data.customerid")
    Set DataReport1.DataSource = adors
    ' Assigning the captions for the labelbox controls.These will
    ' be the column headers.
    DataReport1.Sections(2).Controls("Label2").Caption = "Customer Name"
    DataReport1.Sections(2).Controls("Label3").Caption = "Address"
    DataReport1.Sections(2).Controls("Label4").Caption = "Order Value"
    ' Assigning the values to the textbox controsl in the details
    section.
    DataReport1.Sections(3).Controls("Text3").DataField = "ordervalue"
    DataReport1.Sections(3).Controls("Text2").DataField = "address2"
    DataReport1.Sections(3).Controls("Text1").DataField = "customer_name"
    ' The Function1 will add the total of ordervalue.
    DataReport1.Sections(5).Controls("Function1").DataField = "ordervalue"
    DataReport1.Show
```

Observe that we are creating a fresh recordset here. The value for the column headers are given during runtime. Similarly the data that has to be displayed is also specified at runtime. In this way you can create multiple reports using one DataReport designer.

Caution: Make sure you know the captions and the fields that you are going to display in each report. If during design time you have created a column header and do not use the same during runtime then you will have a column and no data. At the same time if you have a textbox on the Datareport designer you must assign a valid data field to it. You can however set its visible property to False so that it will not be displayed on the report. In order to reposition the various data fields and their captions you have carefully work out the Left and Width values of each of the fields. You can also work on changing the sequence of display of the data fields. You might be tempted to design a report from scratch.

With this we will wind up our discussion on Data Report.

Exercise

Create a report using the Invoice.mdb. The report must list the total order value for each city.

In this chapter we took a look at Crystal Reports and the Data Report.

There is much more that you can do with Crystal Reports and Data Report. We will however stop our discussion with this.

Summary

In this chapter we saw how to

Create a report using the Crystal Reports Wizard.

Create a report from scratch.

Make changes to the report to suit our requirements.

Add formulae to display selected records.

Use the Crystal reports control in our Visual Basic program.

How to use Data Report.

Using a single Data Report to create multiple reports.

Creating Reports

Visual Basic 6.0 gives two different ways to generate the reports :

1. Data Report Designer
2. Crystal Reports

Difference between Data Report and Crystal Report

S.No.	Data Report	Crystal Report
1.	Data Environment is a built-in utility with Visual Basic 6.0	Crystal Report is an Add-on feature whose version 3.0 is coming bundled with 3rd CD of enterprise edition. For higher versions Crystal Report s/w has to be procured.
2.	It is easy way to create simple reports.	It is more powerful than Data Report.
3.	Data fields can be dragged from the Data Environment created.	Since it is an external tool so Data Environment can't be used.

How to create reports using Data Report Designer

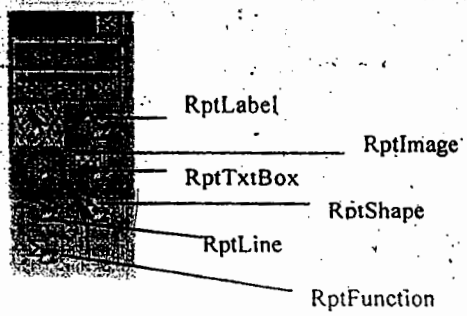
Step 1 :

- Start a new project
- Add a Data Environment and set the connection to point to the employee database created earlier.
- Add a command to the connection created in above step and edit the properties. Set the Database Object to Table, and for the Object Name pick the table emp.

Step 2:

Add Data Report by :

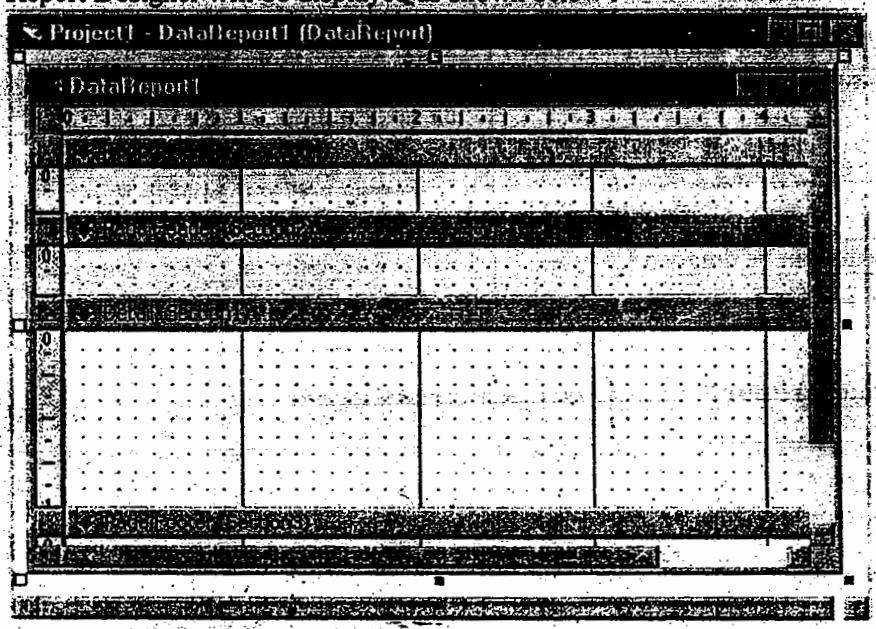
- Clicking on **Project >> Add Data Report.**
- One more control group will appear on the toolbox as shown below :



Note : If this option is not visible then

- *Click Project>>Components.*
- *Click Designers tab.*
- *Check the Data Report check box.*
- *Click OK.*

Report Designer will be displayed as shown below :



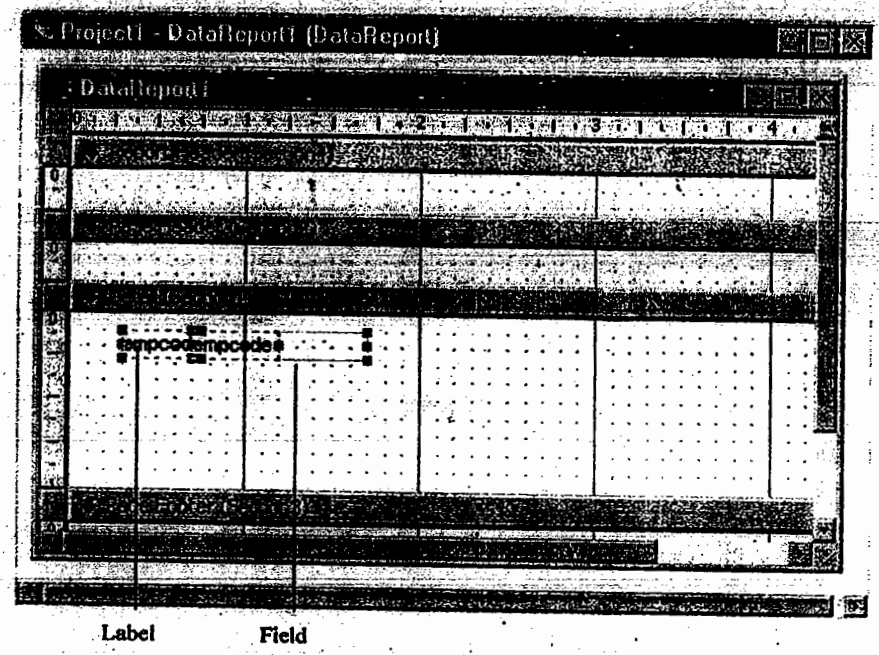
- Set the following report propereties as given below :
  DataSource field to: *DataEnvironment1*
  DataMember field to: *Command1*

Following table shows various report sections and its descriptions :

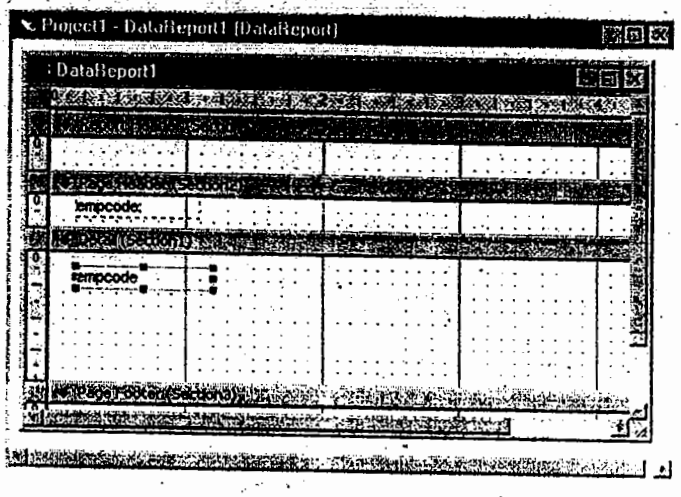
Data Report Section	Description
Report Header	Appears at the Beginning of the Report.
Report Footer	Appears at the End of the Report.
Page Header	Appears at the top of every page.
Page Footer	Appears at the end of every page.
Group Header	A group is associated with a Command in the Data Environment. You can use multiple groups to show data from multiple Commands. Each group begins a new section based on a command.
Group Footer	Ends a group section.
Detail	This is where all the action happens. This is repeated over and over for each record in the Database. so if you have 100 records there will be 100 copies of this detail.

- In the Data Environment window expand Command1 so that you can see each of the fields.

- Now drag the required fields onto the Data Report window and drop it in the Detail section. For example drag and drop empcode.
- It adds two objects to the report as shown in figure below :



- If you click on the left most one you will notice it is a Label, which will display the field name and the rightmost one is a field (name:ReportTextBox) which will contain the data from that field.
- Drag the **label** in the **page header** section so that it should appear once per page and in details section only the data will be displayed. It should appear as shown below :

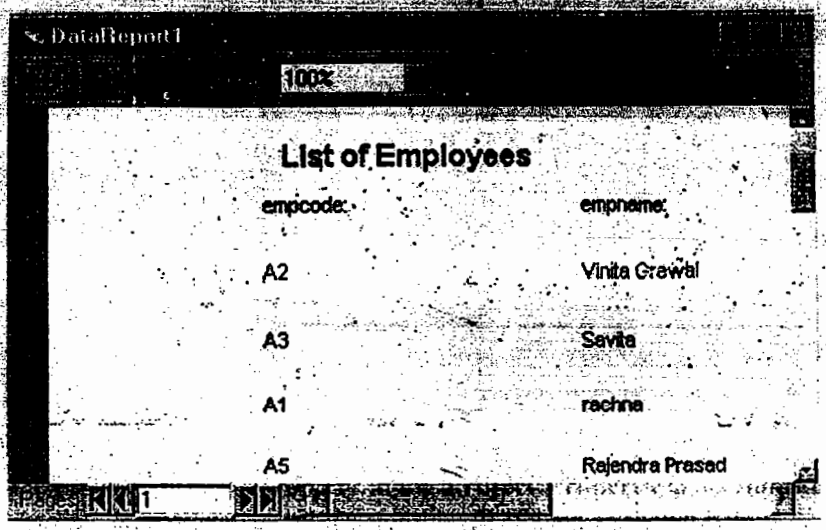


- Similarily drag other fields and adjust them.
- To create a header

- Click RptLabel control from the Control box and drop it in the Report Header section.
- Set the labels caption property to *"List of Employees"*.
- Set the font *Arial, 12, Bold.*

Preview the Report

- Set the **Start Up** property in the **Project Properties..** as *DataReport1.*
- Run the project. The result will be displayed as shown below :



How to call a report from a form

- Open the Form called *Form1.*
- Add a CommandButton to the form, and change the Caption to be "Preview". Now double-click the CommandButton and place this code:

```
Private Sub Command1_Click()
   DataReport1.Show
End Sub
```

- Set the **Start Up** property in the **Project Properties..** as *Form1.*
- Run the project and click on the **Preview** button to see the report.

Printing the Report

- Add another CommandButton to the form, and change the Caption to be "Print". Now double-click the CommandButton and place this code to print first 4 pages of the report:

```
Private Sub Command1_Click()
    DataReport1.PrintReport True, rptRangeFromTo, 1, 4
End Sub
```

Exporting the Report

This is probably the best feature about the Data Report Designer. It has the ability to export reports to text files and HTML files.

- Add **CommandButton** to the form.
- Set the Caption to be ***"Export"***.
- Double-click the CommandButton and place this code to print first 4 pages of the report:

```
Private Sub Command1_Click()
    DataReport1.ExportReport rptKeyHTML[7], "C:\MyReport.htm", True, True,
    rptRangeFromTo, 1, 4
End Sub
```

- Run the project and click the "Export" button.
- Select a place to save the file and then press ***OK***.

Add calculated field in the Report

A calculated field is a field whose value is calculated as the report is generated.

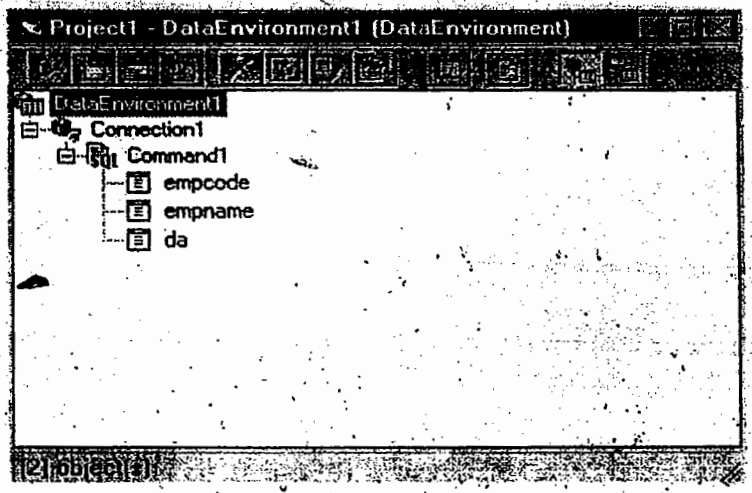
In the Data Environment designer

- **Right-click the Command1 command.**
- **Click Properties to display the Command1 Properties dialog box.**
- On the **General** tab, click the **SQL Statement** button, and add the following statement to the box:

```
Select empcode, empname, (basic * .42) As da from emp[8]
```

The SQL statement multiplies the basic value by the .42 value to create the da value—the calculated field.

- **Click OK to close the dialog box.**
- **Data Environment Designer will look, as given below:**



- Drag the fields from the data environment designer to the detail section of the report.
- Arrange the objects on the report as per requirement.
- Run the report.

---

[7] In case of text file write RptKeyText.

[8] If the name of the table contains a space, then it should be enclosed by square brackets.

Adding Date, Time, Page Number, and Title

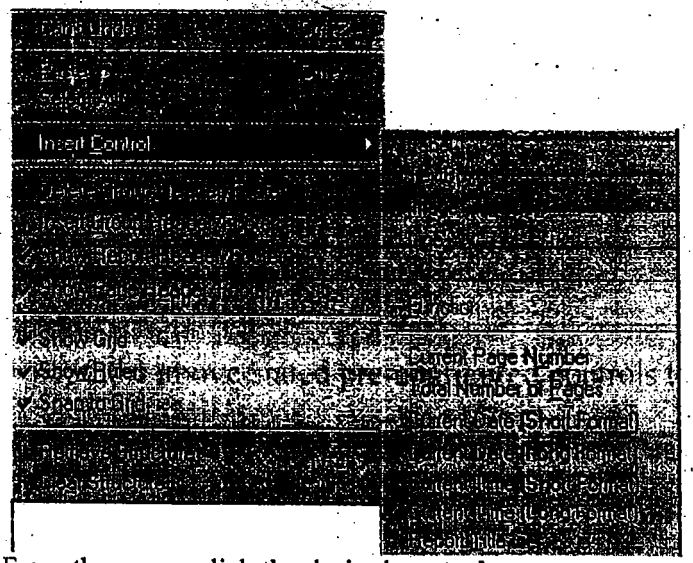
The Data Report designer contains several pre-configured controls that can be used to quickly add the date, time, page number, or report title to any section of the report. These controls are simply Label controls containing codes that display the following information:

Function	Code
Current Page Number	%p
Total Number of Pages	%P
Current Date (Short Format)	%d
Current Date (Long Format)	%D
Current Time (Short Format)	%t
Current Time (Long Format)	%T
Report Title	%I

Note : To display a percent sign, use %%.

To add the above stated pre-configured controls to the page header

- Right-click the Data Report designer and click **Show Page Header/Footer,** *if it is not checked.*
- Right-click the appropriate section and click **Insert Control.** Following menu will appear:



- From the menu, click the desired **control.**.

To add more function to the Report

To add the number of products at the bottom of report, or add a bunch of numbers. Well this is where the RptFunction control comes in.
List of functions supported by **RptFunction** is given in the following table :

Name	Description
rptFuncSum	Sums the values of a field.
rptFuncMin	Displays the minimum value of a field.
rptFuncMax	Displays the maximum value of a field.
rptFuncAve	Displays the average of values.
rptFuncSDEV	Displays the standard deviation for a column of figures.
rptFuncSERR	Displays the standard error for a column of figures.
rptFuncVCnt	Displays the number of fields containing non-null values.
rptFuncRCnt	Displays the number of rows in a section of the report.

*Note : The RptFunction Control can only be used the **Group Footer or Report Footer** sections.*

- Click on **RptFunction** control and drop it in your **Report Footer.**
- Set the **FunctionType** property to ***rptFuncRCnt.***
- Set the DataMember to ***Command1.***

This will count the number of record's in the table

Creating Grouped Reports

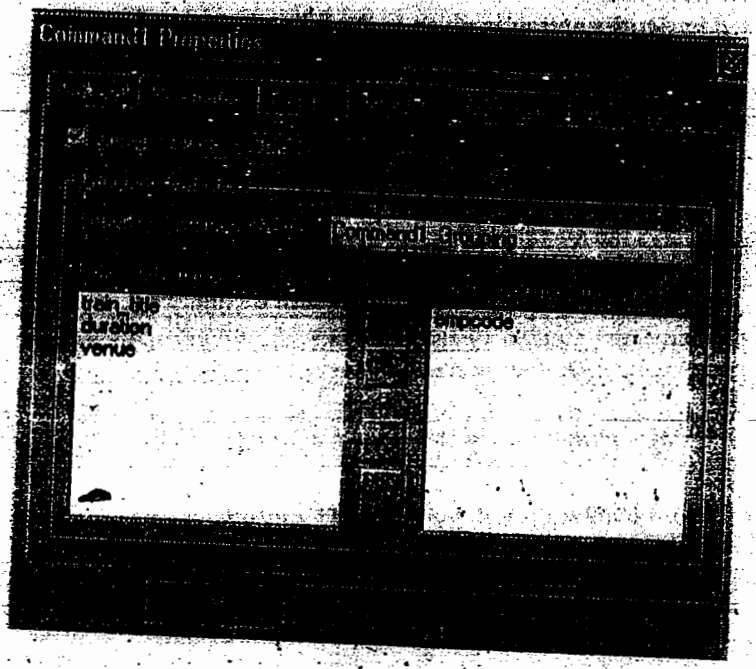
The Data Report designer gives you the ability to group data according to any field in a table. The grouping field provided by the data environment's Command object differs from the grouping already achieved by creating group headers and footers. Instead of using the table of a database as the basis for grouping, the Grouping feature of the Data Environment designer allows you to select a particular field in the table as the grouping field without having to create a new Command object.
The steps to adding a grouping field include:

1. Creating a group field in the Data Environment designer.
2. Adding a Group Header/Footer to the Data Report designer to correspond to the new command.
3. Resetting the DataMember property of the data report to the new Grouping Command object created in the data environment.
4. Dragging the group field from the data environment to the data report.

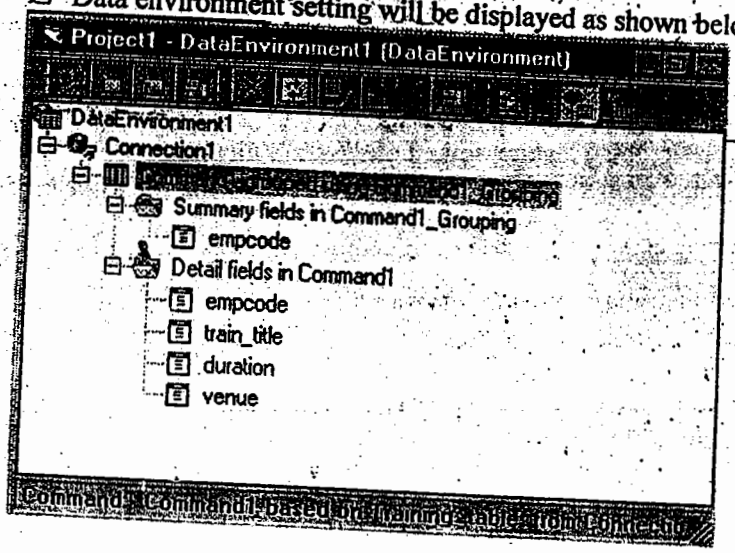
To add a grouping field to the Data Report designer

- Create a Data Environment designer, having connection pointing to employee database and a command connecting to the training table.
- Right click on the command and click Properties to show the **Command Properties** dialog box.
- Click the **Grouping** tab.

- Click **Group Command Object.**
- In the **Fields in Command** box, double-click **empcode**. Settings will be as shown below:



- Click **OK** to close the dialog box.
- Data environment setting will be displayed as shown below :



- Right-click the Data Report designer, and then click **Insert Group Header/Footer.**
- Click the Data Report designer's title bar to select the entire data report. On the **Properties** window, click **DataMember** and change the property from Customers to **Command1_Grouping.**
- In the Data Environment designer, open **Summary Fields in Command1_Grouping**. Drag the **empcode** field into the new section on the Data Report designer.

- Delete the **Label** control that accompanies the Country field.
- Drag the fields from **Command1_grouping** in the details section.
- Save and run the project.

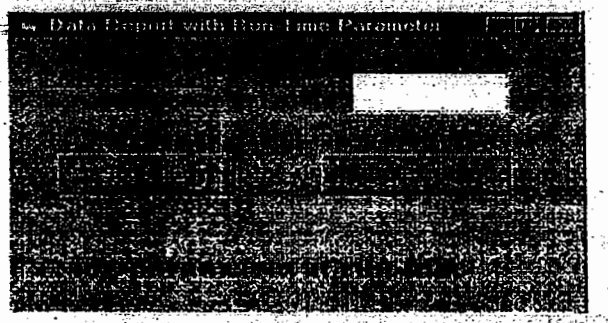
Working with Parent-Child Relationship in the Reports

- Create a command in Data Environment.
- Add a child command to the command having relation with some field
- In the Relationship tab of the properties window of child command , set the relationship.
- Drag the fields from the command in the group header and child command in the detaile section of the reports.

Passing parameters at run-time

How to create a data report by passing parameter at the run-time is illustrated as below by taking the *employee* database and *emp* table.
Following example will display the list of the employee along with their code and name whose basic is greater than the value entered at the run-time.

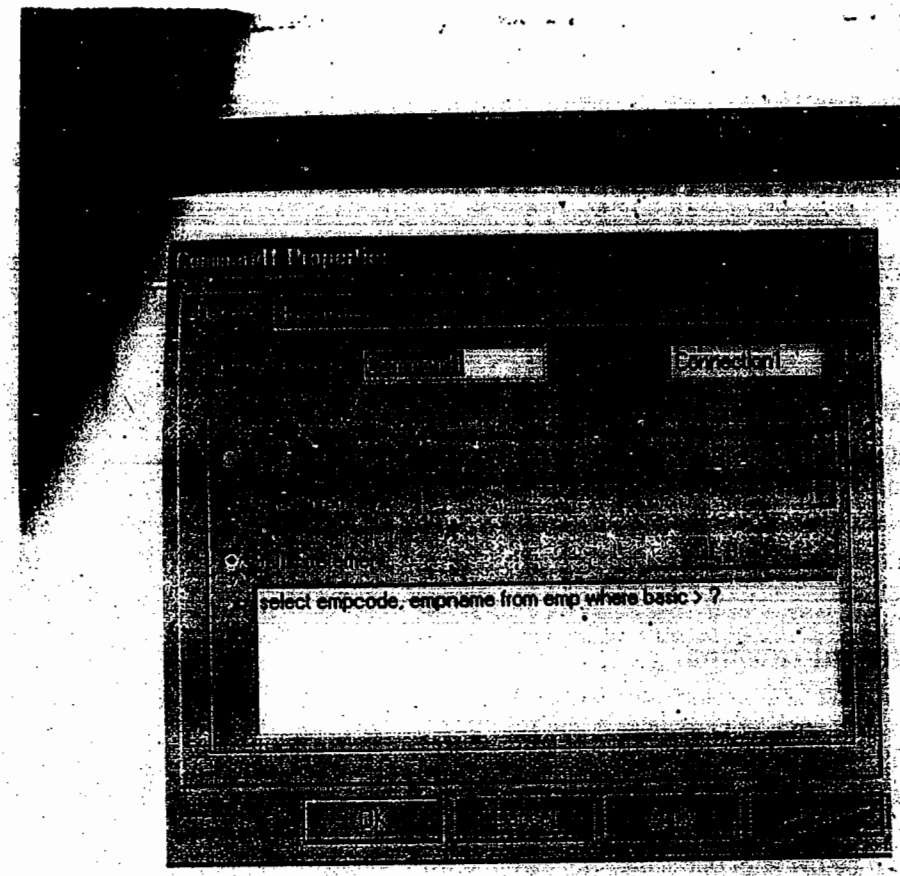
- Create a form as shown below with one label, one text box , two command buttons with the property settings given in the table below the figure :



- Property settings of the objects on the form :

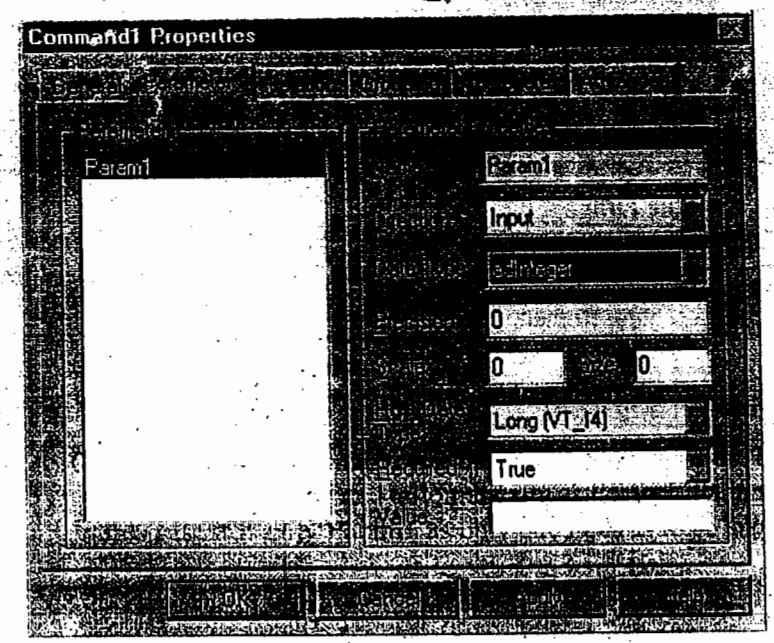
Object	Property Settings
Label	Caption : ***Enter the Basic Pay*** Font : Times New Roman, 11
Text Box	Text : ***Blank***
Command Button 1	Caption : ***Show List***
Command Button 2	Caption : ***Exit***

- Create a **DataEnvironment** with the connection to the **employee** database and add **command** to the connection.
- Right click the command **Command1** and then click **properties.**
- Do the settings as given in the following figure :

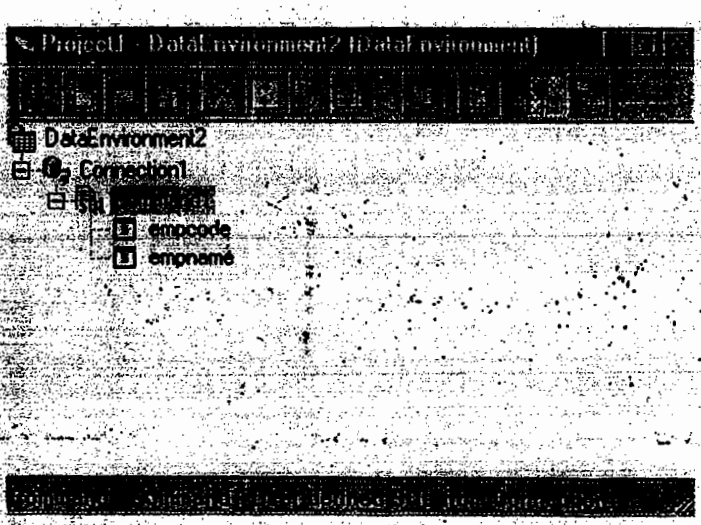


"?" is used for the input to be given during run-time.

- Click the **Parameters** tab to set the Parameter properties. Depending on the number of "?", parameters will be created automatically as Param1, Param2 and son on.
  In this case it is only Param1 as only one input is to be supplied through keyboard.



- Click **OK**.
- Data Environment will be displayed as given in the following figure :



- Set the **DataSource** property to ***DataEnvironment2*** and **DataMember** property to ***Command1.***
- Drag the fields on to the report and adjusted as per requirement as already told in the above sections.
- Write the following code on Command button 1 and Command button 2 of the form already created :

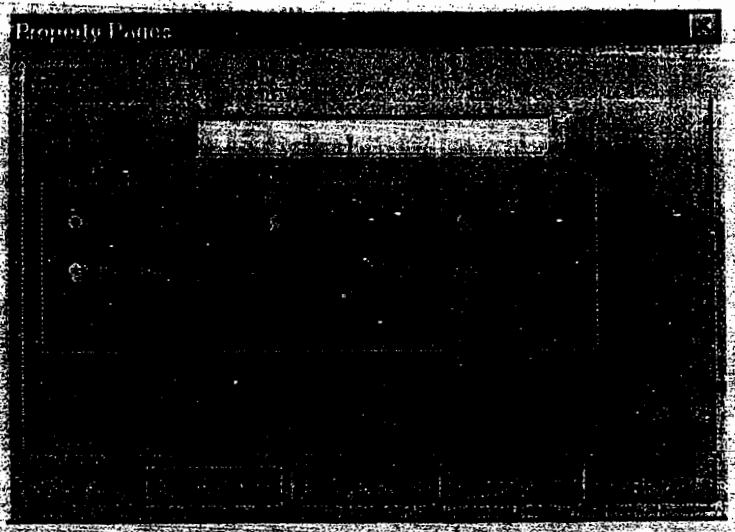
```
Private Sub Command1_Click()
    If DataEnvironment2.rsCommand1.State = adStateOpen Then
        DataEnvironment2.rsCommand1.Close
    End If
      DataEnvironment2.Command1 Text1
      Load DataReport1
      DataReport1.Show
End Sub

Private Sub Command2_Click()
    End
End Sub
```

Using ADO programming in Data Reports

- Create a data Report with textboxes wrt the fields in the table.
- Specify the DataField property to the field name in the table.
- Write the following code on the event from where the report is to be called

```
Private Sub mnuado_Click()
  Dim p As String  'variable to store recordset
  Dim conn As New ADODB.Connection
  Dim rs As New ADODB.Recordset
  conn.Open _
"Provider=Microsoft.Jet.OLEDB.4.0;DataSource=d:\vb_tot\demos\reports\employee.mdb"
'connection  to the database
```



- Select the **report file name** already created by clicking on to the **browse** button
- Select the desired destination from the available options.
- Click OK.
- Create a command button on the form.
- Write the following code on the click event of the button :

```
Private Sub Command1_Click()
 CrystalReport1.PrintReport
End Sub
```

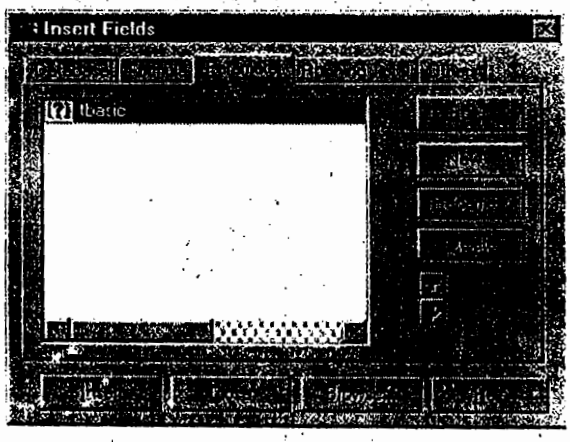
- Run the form and click on the button to view the report.

Passing parameters from visual basic to Crystal report

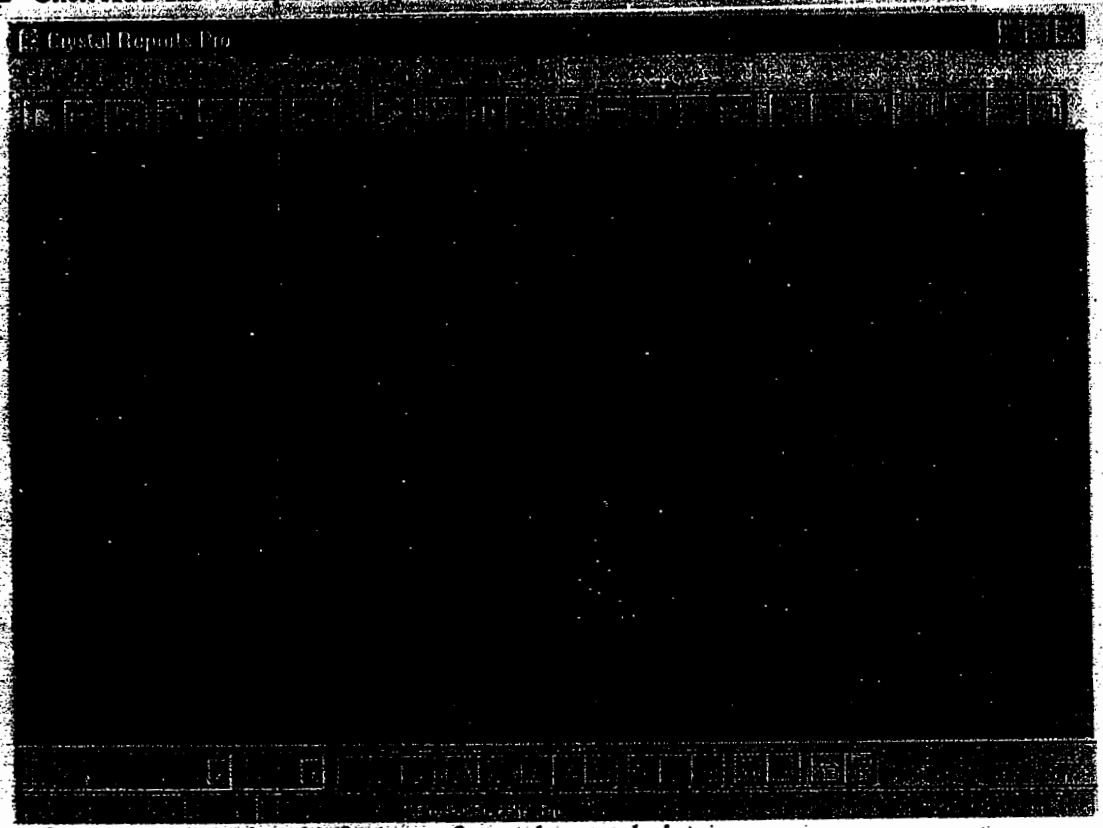
At the Crystal report end :

- Create a crystal report.
- Click **Insert>>Parameter Fields**.

Following screen will appear :

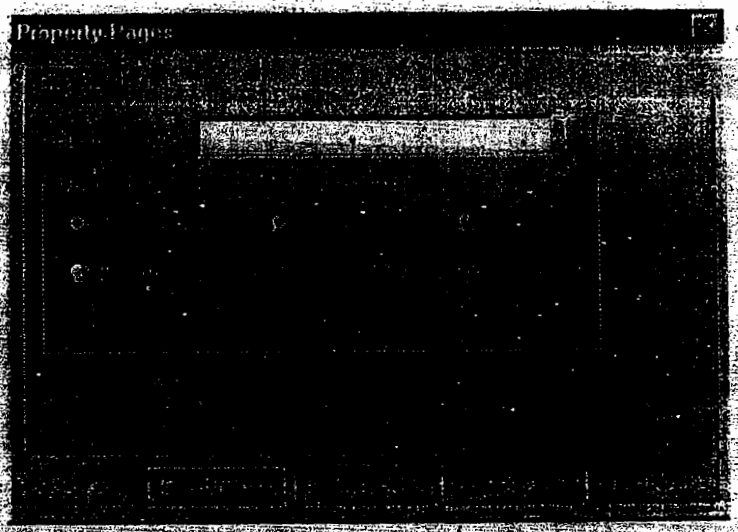


- Click **Add-Ins>>Report Designer**. Following screen will appear :



- Create a report using the features of crystal report designer.
- Save the report. Its extension will be ***rpt.***
- Add **Crystal Report control** to the toolbox by selecting **Crystal Report Control 4.6** from the ***Project>>Components.*** Following icon will be added on the toolbox:

- Put this control on the form. Its default name will be ***CrystalReport1.***
- Click on then **ReportFileName** property of **Crystal Report Control .** Folllowing property page will be dispalyed :



- Select the report file name already created by clicking on to the browse button
- Select the desired destination from the available options.
- Click OK.
- Create a command button on the form.
- Write the following code on the click event of the button :

```
Private Sub Command1_Click()
 CrystalReport1.PrintReport
End Sub
```

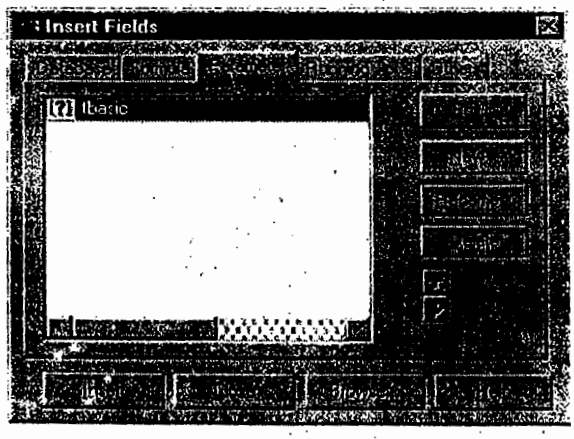
- Run the form and click on the button to view the report.

Passing parameters from visual basic to Crystal report

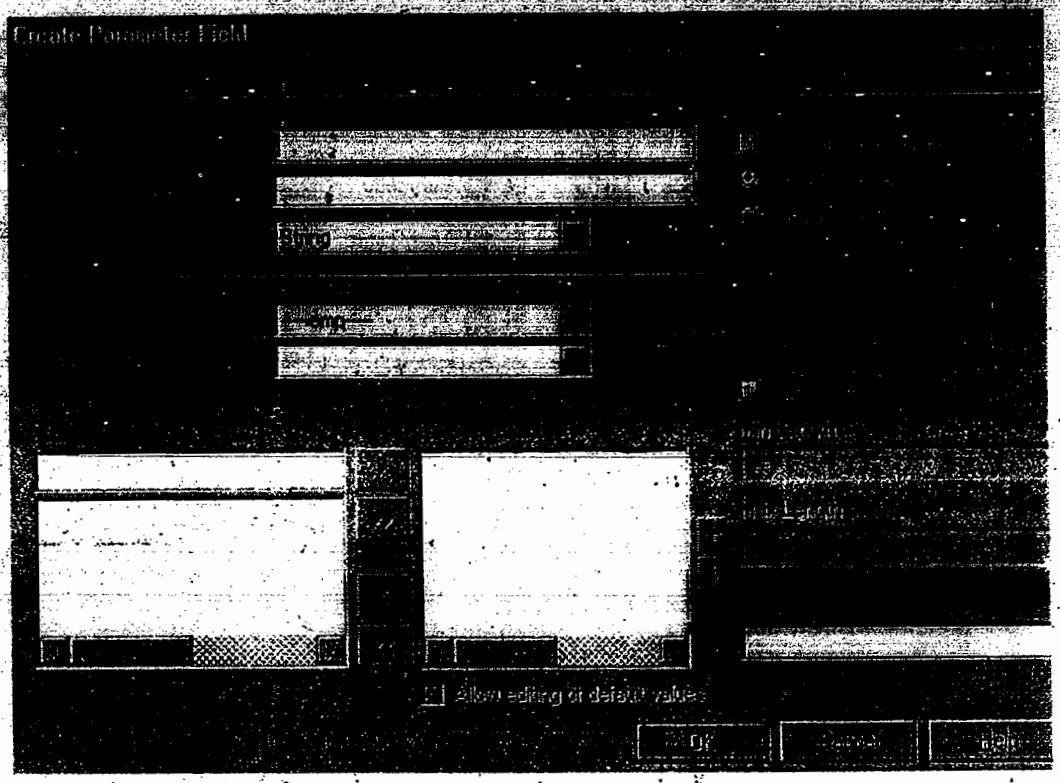
At the Crystal report end :

- Create a crystal report.
- Click **Insert>>Parameter Fields.**

Following screen will appear :



- Click **New.**

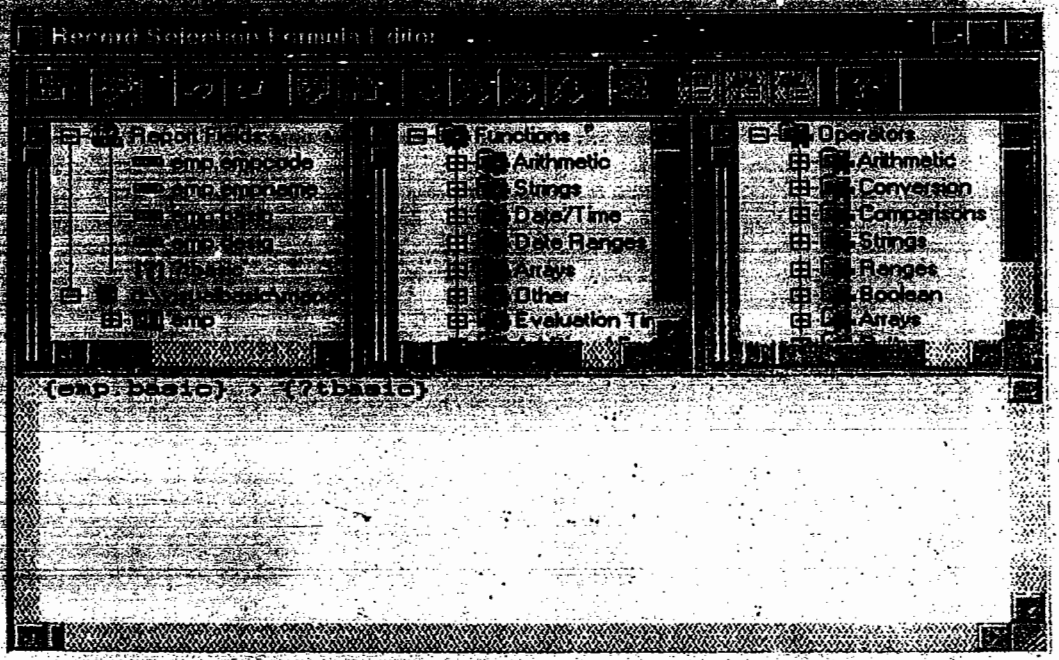


- Specify the name and the datatype in the appropriate boxes.
- Click **OK.**

Repeat the above three steps to incorporate more parameter fields.

- Click **Close**
- Click **Report>>Edit Selection formula>>Record.**

Specify the formula by selection from the following screen



- Close the window.

At the Visual Basic end :

- Open a VB project and create a form from which input will passed to the Crystal Report. Values can be passed either through text boxes, combo boxes or any other input method.
- Add **Crystal Report control (crystl32.ocx)** to the toolbox by selecting **Crystal Report Control 4.6** from the ***Project>>Components***. Following icon will be added on the toolbox:

- Put this control on the form. Its default name will be *CrystalReport1*.

Write the following piece of code on the click event of the object

```
CrystalReport1.ReportFileName = "<Reportfile name with path>"
CrystalReport1.ParameterFields(0) = "<crystal report parameter field name>;" & _
Text1.Text & ";TRUE"
CrystalReport1.Destination = crptToWindow
CrystalReport1.PrintReport
```