Crystal Reports 2016 Designer 3 Workshop

Vision Harvest, Inc.

888-236-8087
sales@visionharvest.com

Published: August 9, 2016
Part: VISI100102S

This course covers Crystal Reports® 2016

This courseware has been developed by the professional team at Vision Harvest, Inc., www.visionharvest.com. Questions, comments or concerns about courseware content should be sent to sales@visionharvest.com or 888.236.8087.

Table of Contents

Introduction

Introduction Objectives

This manual is written to give you a step-by-step guide for your classroom training and a handy reference for your daily work. In this Introduction, you will learn how to use this training guide effectively. This section covers the following topics:

- An introduction to the Crystal Reports application
- Pre-Qualification Exercise
- Class objectives
- Help with using this training guide
- Information on how to start the program

About Crystal Reports

In today's information intensive environment, every business has a database of some sort. After all, business today is all about information and databases give you a handle on the massive amounts of information you must deal with. Therefore, your business has a database and from that database, you need reports. The problem is, most reporting capabilities that come with database programs are limited. They only report on data from that program. Many users need to report on data from multiple sources, even databases such as Oracle, Microsoft SQL Server, DB2 or Sybase.

Crystal Reports is one of the most powerful reporting programs available with the ability to pull data from all types of data sources. You can use Crystal Reports to generate reports from any of the standard PC database programs; Access, Paradox, or FoxPro, as well as from a mainframe or server database. Crystal also has a powerful web-reporting server that allows you to distribute your reports over the web.

Crystal Reports is bundled with more than 160 other programs including Visual Basic, some medical applications, many accounting packages and several ERP solutions. It makes report generation easy without requiring you to be a programmer or a database expert. If you know how to work in a Windows environment and are familiar with the data you want to use, you can create a Crystal Report that looks professional and makes sense.

Training Philosophy

Studies show that people retain 10% of information they see, 20% of information they hear, 50% of what they see and hear, and 80% of what they see, hear and do. In line with this concept, the class utilizes a hands-on method of training. You will see the effects of new procedures on the screen, hear the instructor explain how and why to use features, and perform the actions yourself as you learn.

In addition, this class focuses on your ability to perform tasks using the most productive techniques. The manual may contain several methods of accomplishing a certain task. However, class time does not allow for practice of all methods for each task. Your instructor will guide you in the most effective method of performing a task, but inform you of other methods that are available.

Questions are encouraged. While we give our best effort to explain new concepts in understandable terms, you may need to hear the concept again or have it explained more thoroughly. Please let the instructor know when you need more information!

Class Objectives

This class is a performance based instructional system. It is geared to provide you with the tools you need to build and distribute reports the quickest, most efficient way. After completing this course, you will be able to perform the following tasks:

- ❖ Create and incorporate custom functions into formulas
- ❖ Build high level advanced formulas with looping statements and case select
- ❖ Use SQL statements to build Crystal Reports SQL Expressions and Commands
- ❖ Apply advanced database concepts with views and stored procedures
- ❖ Performance fine tune reports for shorter run times
- ❖ Build advanced Cross-Tabs using the new Crystal Reports 2008 advanced Cross-Tab features
- ❖ Apply everyday tips and tricks into corporate reports

About This Manual

Each section of this manual contains objectives to provide you with the overall goals for the lesson. Lessons have descriptions of features and concepts followed by systematic directions for completing a specific task. Each section ends with a challenge exercise to help you practice the skills you learned in the lesson. Challenge exercises provide you with tasks to accomplish. Try to complete these exercises on your own.

As you work in this Training Guide, certain conventions are used to identify specific procedures. Use the following table as a guide:

Training Guide Conventions	
Item	**Illustrated As**
Menu Commands	Underlined letters for accessing menu commands are shown: Example: File/Open
Command Buttons	Command Buttons in dialog boxes are shown as buttons: Example: Finish
Categories, Radio Buttons, Text Boxes, Check Boxes	All options within dialog boxes are listed in italicized text: Example: the *Keep Group Together* check box the *Other* radio button
Keystrokes	Keyboard keys are indicated by uppercase text: Example: press ENTER Keyboard combinations are shown in uppercase text with a plus sign (+) between the keys that need to be pressed simultaneously. Example: press CTRL + S to save
Toolbar Buttons	Toolbar buttons are indicated by the button name and a graphic image of the button: Example: click the Print Preview button
Typing or File Selections	Text to be typed or file names to be selected are printed in bold letters: Example: type **Henry** select **grouping.rpt**
Exercises	Step-by-Step exercises in the text are indicated by bold text and the ❖ symbol. For example: ❖ **Exercise - Format Objects**

Tips, Notes, and Warnings

Tips, notes and warnings display with the following icons. Text for these additional comments display in bold and italics as shown below:

This icon indicates a tip or shortcut.

This icon points out a note of additional information.

This icon calls attention to a warning or very important note

NOTES

Lesson 1
Refresher Exercise

Lesson Objectives

After completing this lesson, you will be able to:

- **Plan and Create a report from a set of business requirements**
  Review the techniques needed to plan and create a report
- **Add multiple groups to reports**
  Create groups within groups to subdivide your reports how you want
- **Conditionally format your report**
  Make your report more informative and attractive with formatting
- **Parameterize your report**
  Use parameter fields to provide flexibility to the user as to the data they are viewing
- **Add multiple charts to a report**
  Use more then one chart and conditional formatting to graphically display data for different groups

Review of Planning a Report

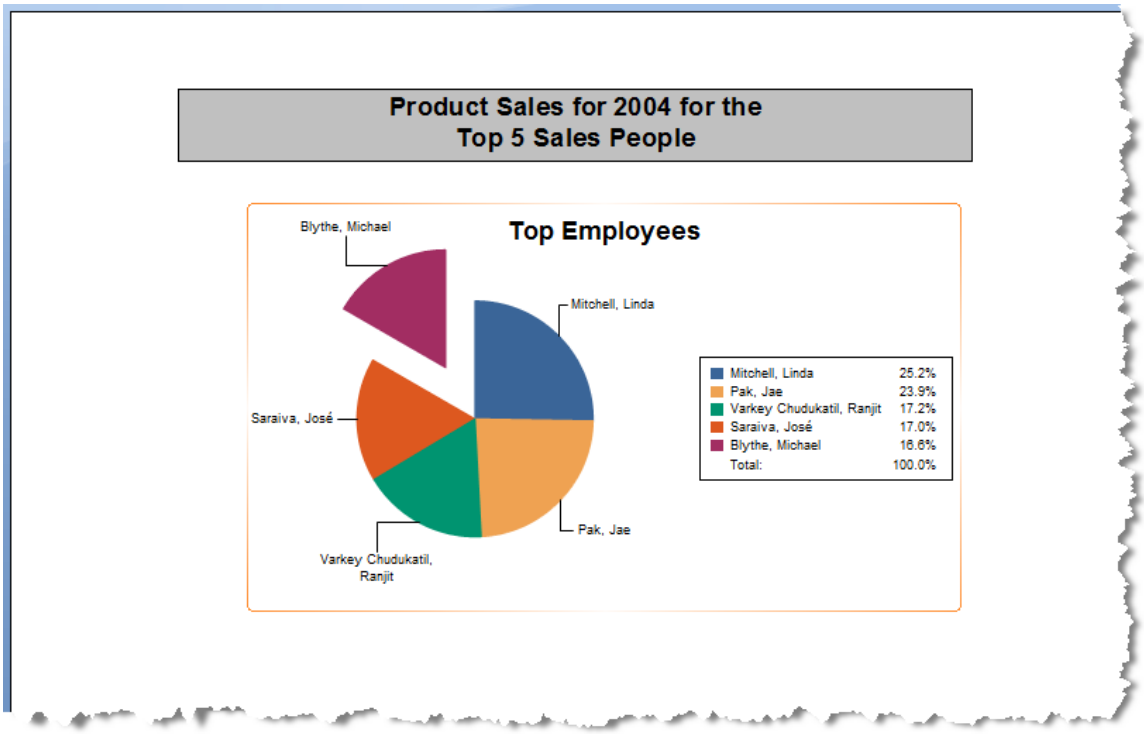
To make the report creation process efficient, you must plan the report. A little work up front before you ever open Crystal Reports can save a lot of time putting the report together. It can also save problems caused by adding more tables to the report that can change the number of records displayed.

You must look at the business requirements for the report and decide what fields you need and the tables in which they are located. It is also helpful to look at things such as groupings, whether you need Cross-Tabs, fields needed for formulas and selection criteria.

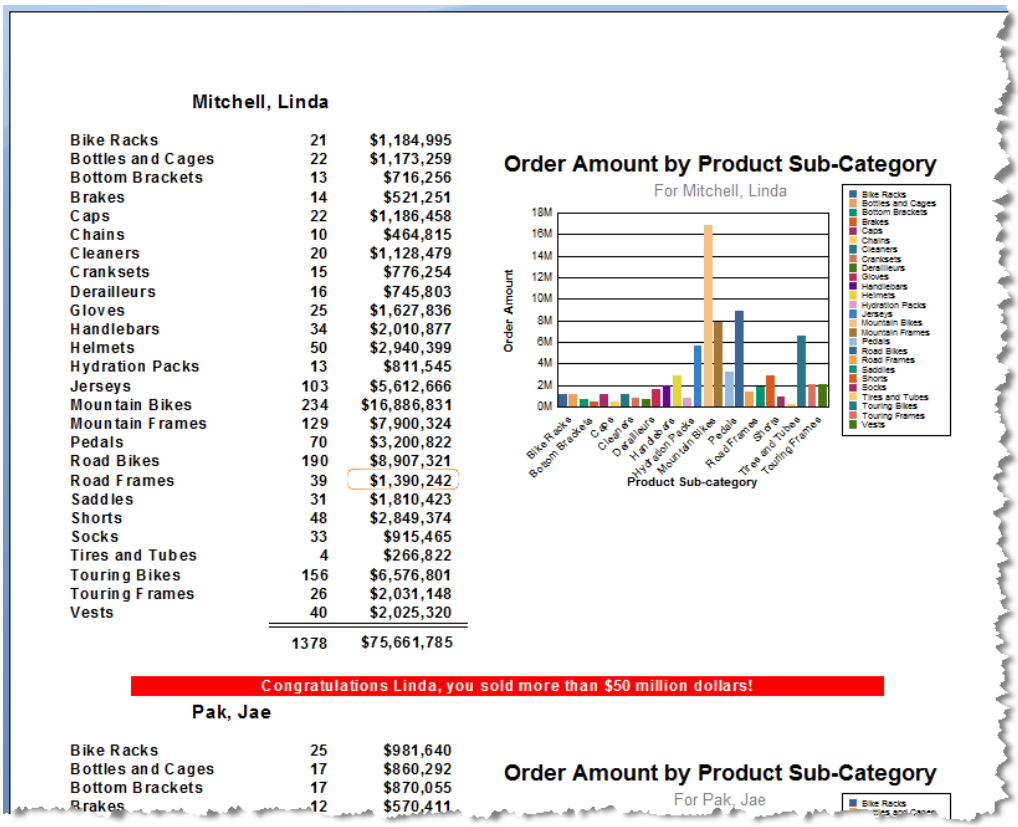
Suppose someone asks you to create a report based on the AdventureWorks database. Below are the business requirements for the report:

- The first page of the report needs to show the top N sales people by product sales; N needs to be a user defined variable.
- The body of the report shows the sales detail for each sales person showing the number of orders and amount by product sub-category. Beside each sales person should be a bar chart showing order amount by product sub-category for that sales person.
- The user needs to be able to choose a year for the report with a default value of 2004.
- Need to be able to see quickly any sales person with over $50,000,000 of sales
- The person requesting the report has supplied you with a printout of what the report should look like, both the first page and detailed pages. See the samples on the next page.

The screen shot below is a view of the first page of the report:



Page 2 of the report should look as follows:



To create this report, first determine which fields you need.

- For the Detail section, you need the SalesOrderID and SubTotal fields
- For grouping, you need the SalesPersonID and the Product Sub-category Name fields
- Finally, you need to select orders based on the OrderDate field

Next, decide which tables you need to add to get the required fields.

- The SalesOrderID, SubTotal and OrderDate fields are in the SalesOrderHeader table
- The SalesPersonID field is in the SalesPerson table; Sales Person Name is in the Contact table via the Employee table
- The Product Sub-category Name field is in the ProductSubcategory table. To get to this table you need the SalesOrderDetail and Product tables

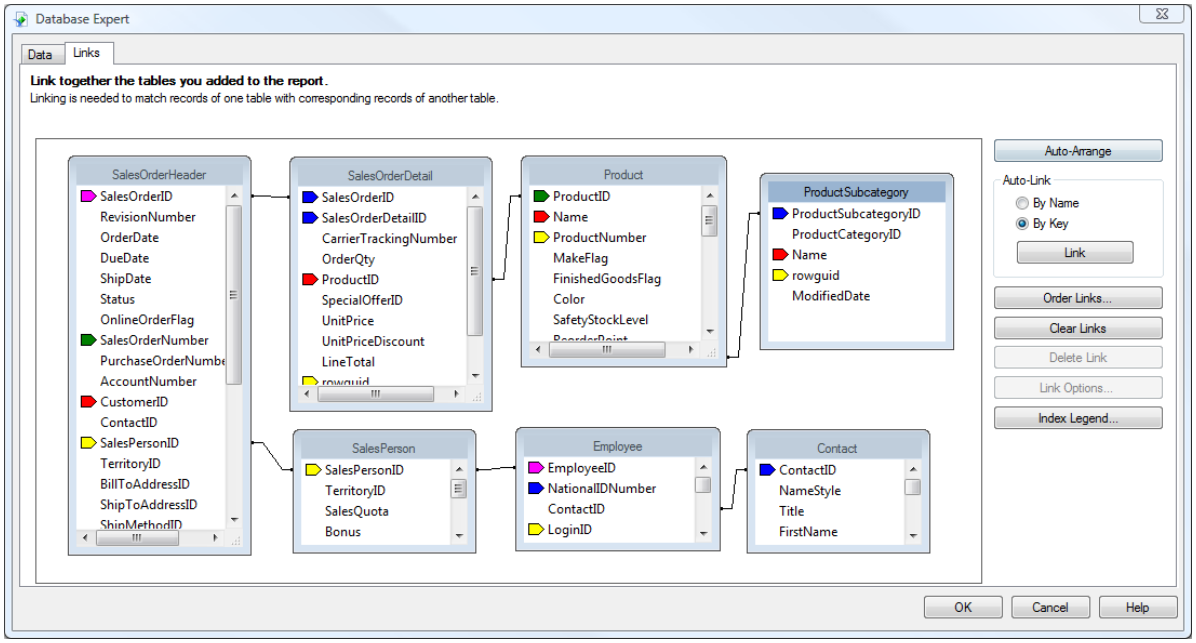
The order in which the tables are linked is usually very important from a performance viewpoint. When using the Smart Linking feature provided by Crystal Reports, the tables will be linked automatically in the most logical performance-driven fashion, but this is only true in Crystal Reports 2008.

Crystal Reports 2008 attempts to link tables together where the "To" table contains a primary key (indexed, no duplicates). The reason is to improve database performance and in most instances is the best choice to reduce database querying time. In some cases this is not the best choice and it is fairly easy to override the automatic links and create your own.

This means you can simply add the tables in any order and let the Smart Linking feature do the rest of the work.

Creating the Report

Since you have given this some thought and determined the required tables, it is time to create the report. Once the tables are added, Smart Linking will create the links (joins) similar to the following example:



NOTE: Remember you can add multiple tables to a report from the Database Expert. You work with links using the Links tab on the Database Expert. The Database Expert is accessed under the Database menu or by selecting the Database Expert button on the Experts toolbar.

Placing Fields on the Report

Remember you have three ways of placing fields on a report. Select the field in the Field Explorer, then:

- Click the Insert button, and then click in the report
- Drag the field to the report
- Right mouse click, choose Insert to Report from the short cut menu, and then click in the report

When you place a field in the Details section, Crystal Reports adds a detail field title in the Page Header section and aligns the field and field title with a guideline. You can drag the guideline marker in the ruler to move the field and its title together.

Now that you have planned the report, you are ready to use Crystal Reports to create the report.

NOTE: Your instructor will provide the database connection entries if they differ from the standard entries in Exercise 1.0.

❖ Exercise 1.0 - Begin the Monthly Sales by Supplier and Category Report and Link the Tables Needed

1. Start a new report as a blank report. Open the **Create New Connection** folder then the **OLE DB (ADO)** data sources; the OLE DB (ADO) dialog will open.

2. In the **OLE DB Provider** section, select **SQL Native Client**, and then click **Next**. The **Connection Information** section appears.

3. Enter **localhost\SQLEXPRESS** in the *Server:* field, **sa** in the *User ID:* field and **cr2008** in the *Password:* field (the password is case sensitive). Select **AdventureWorks** from the *Database:* drop-down field. Be sure that the *Integrated Security:* checkbox is not checked. Click **Finish**.

4. Add the tables indicated by the business requirements. *(SalesOrderHeader, SalesOrderDetail, SalesPerson, Employee, Contact, Product and ProductSubcategory)*

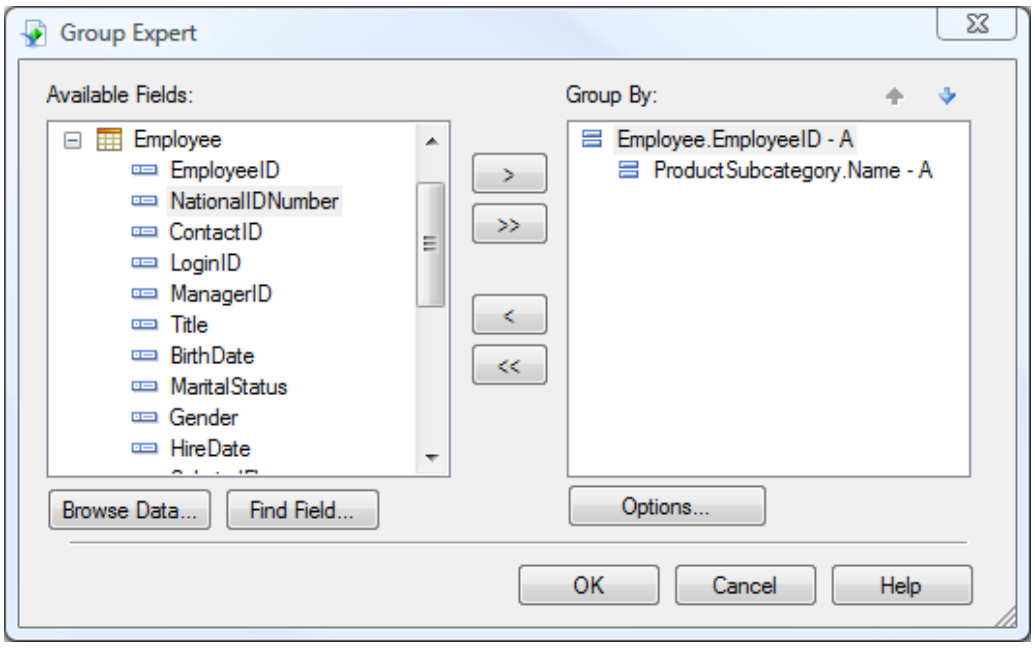
5. The Auto-Link feature should link the tables automatically. Clear the links and do them yourself, but create the links to match the example on the previous page.

6. Add the appropriate fields to the **Details** section *(SalesOrderID, SubTotal)*

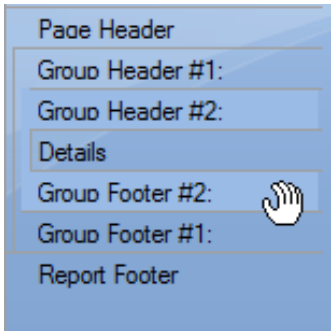
7. Change the Printer Setup to a **Portrait** page layout, if necessary. Change the margins to be **.5"** on all four sides

Advanced Grouping

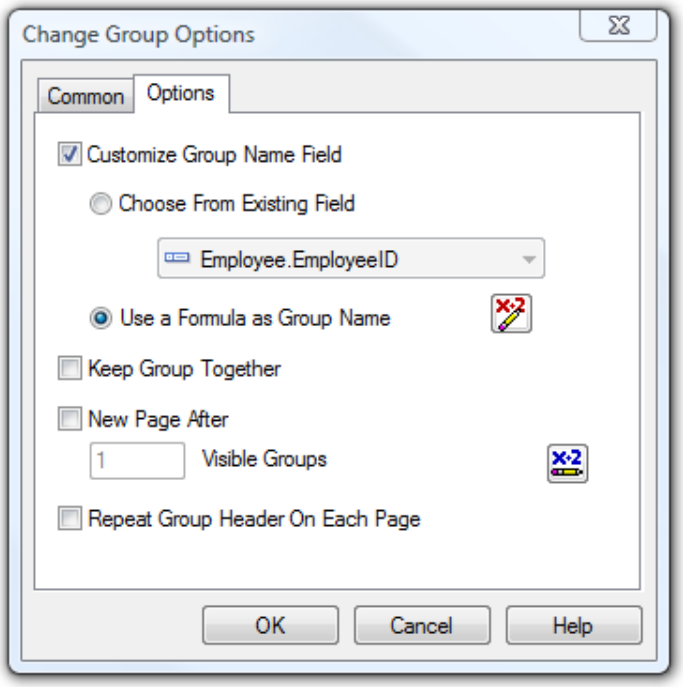
You can group data by using the *Insert/Group…* command or by clicking the Insert Group button on the Insert toolbar or the Group Expert button on the Experts toolbar. This report requires a Product Sub-category Name group nested inside a Contact (Employee) Name group as shown in the Group Expert illustration below:



As you create groups, Crystal Reports nests each group inside the previous one you created. However, you can reorder groups in any order you want. To reorder groups, left-click the Group Header section name at the left of the Design screen for the group you want to move. The group header and footer sections are highlighted. Drag the group to the new location. The mouse pointer changes to a grabbing hand as you drag. In the Group Expert shown above you simply move the group headers using the arrows in the top right of the window:



Remember that you can group on a field but display a formula or another field name as the group name field. When you group on Employee Id using the option shown below (Using a Formula as Group Name) you can display the employee name:



Also remember that you can summarize any field by RIGHT clicking it, then choosing *Insert Summary…* You must specify the type of calculation you want Crystal Reports to perform and the group at which you want the summary to appear.

❖ Exercise 1.1 – Create groups and summarize the SalesOrderID and SubTotal field

1. Open the **Group Expert** window using the Group Expert button. Create a group on the **ProductSubcategory.Name** field.

2. Create a second group on the **Employee.EmployeeID** field. Choose to *Use a formula as Group Name*, displaying the Employee Name as Last Name, First Name (from the **Contact** table).

3. The report requirements call for the report to display information by **EmployeeID** and then **Product Sub-category Name**, so you need to change the order of the groups. Click the Group Expert button. Select the **Product Sub-category Name** field in the *Group By:* list and then click the Down Arrow to reorder the groups
The EmployeeID should be the first group in the Group Expert with the Product Sub-category Name nested underneath.

4. Create a summary for the **SalesOrderID** and **SubTotal** columns. Do this by checking on the *Add to all group levels* option and then choose **OK** from the Insert Summary dialog
All the summary fields should automatically line up on the guideline to which the field is attached.

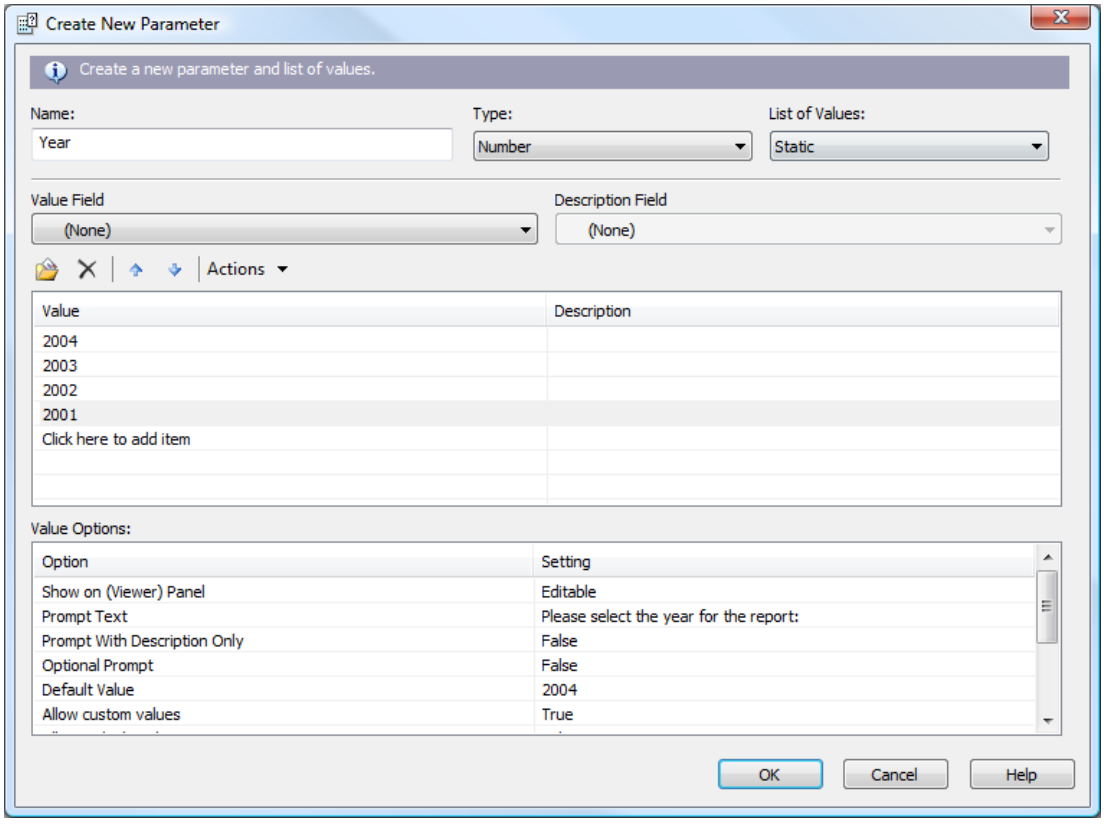
5. Move the summaries in **Group Footer #2** to **Group Header #2**, and then hide the **Details** and **Group Footer #2** sections.

Selecting Certain Records

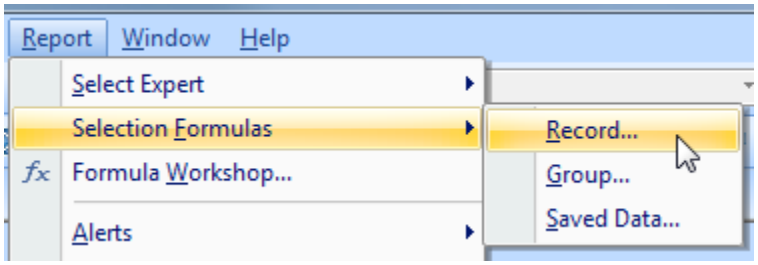
Remember Crystal Reports pulls all records from the database unless you filter the records based on some criteria. The business requirements for this report requests only records from a user selected year. First you need to create a parameter field and then use it in the record selection formula.

❖ Exercise 1.2 – Select Records for a Specific Year

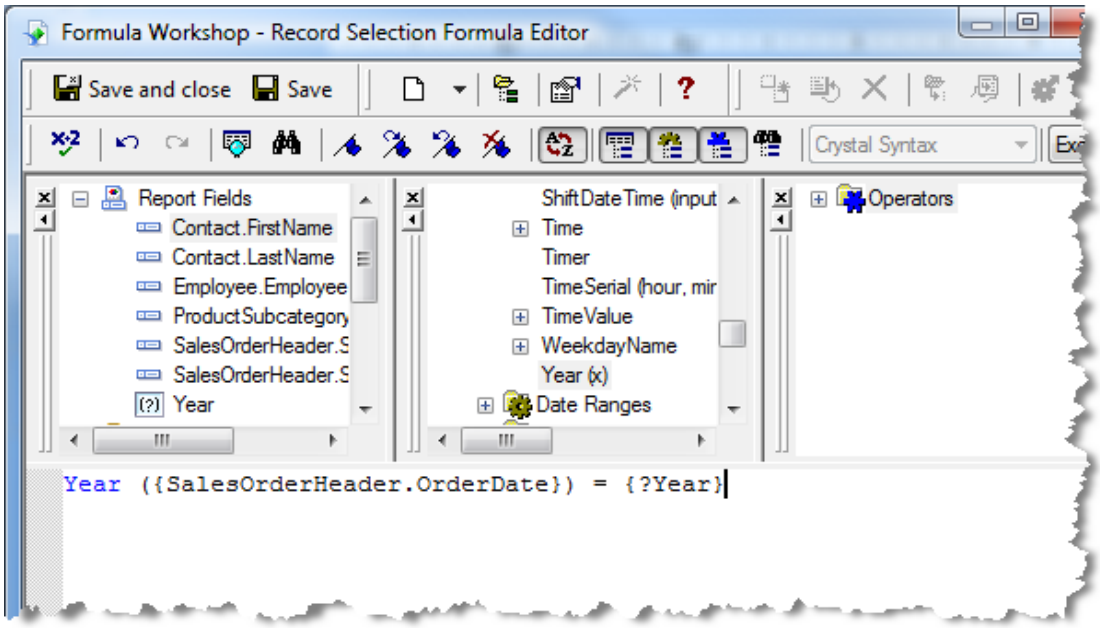
1. Create a parameter field called **Year** with a **Number** data type and a default value of **2004** as shown in the dialog below:



2. Under the **Report Menu** choose to edit the record selection formula as shown in the menu below:



3. In the formula editor you need to enter the following formula:



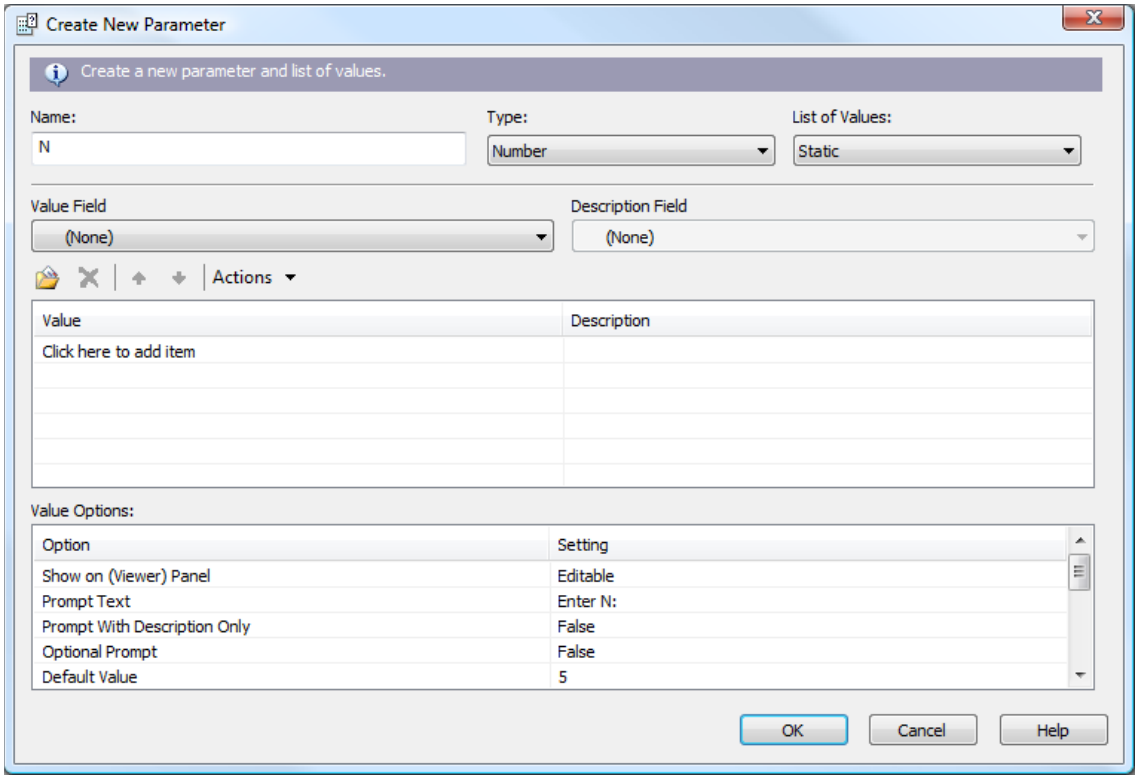
4. At this point save and preview the report. Name the report: **Refresher.rpt**

Group Sorting

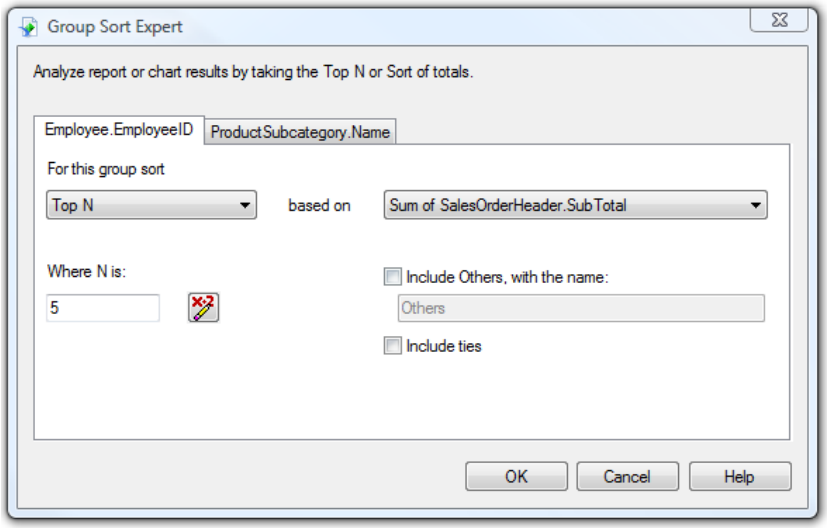
An additional requirement of this report is group sorting. Now you have the group summary fields you can choose how to sort the groups. One of the requirements outlined is the ability to look at the Top N sales people and top is measured by order amount. It is also a requirement to be able to select the value of N at run time.

❖ Exercise 1.3 - Sort Groups

1. Before you start sorting your groups you need to create a parameter field for **N** as the N value needs to be user defined. Create a simple parameter field with a default value of **5** as shown below:



2. Click on the **Group Sort Expert** and you will be able to select the type of sort for your group as shown below. The Group Sort Expert has a tab for each group on the report and within that tab you can choose the summary field on which you want to base your sort. You then decide which type of sort (Top N, Bottom N, All Sort etc) that you would like to do.



3. In this case you want to sort on **EmployeeID**; do a **Top N** sort based on **SubTotal** and exclude *Others*.

4. In order for **N** to be user defined you need to click on the conditional box (next to *Where N is:*). Simply add the parameter field **{?N}** to the formula box.

5. Preview your report and select values for both parameter fields. Save your report.

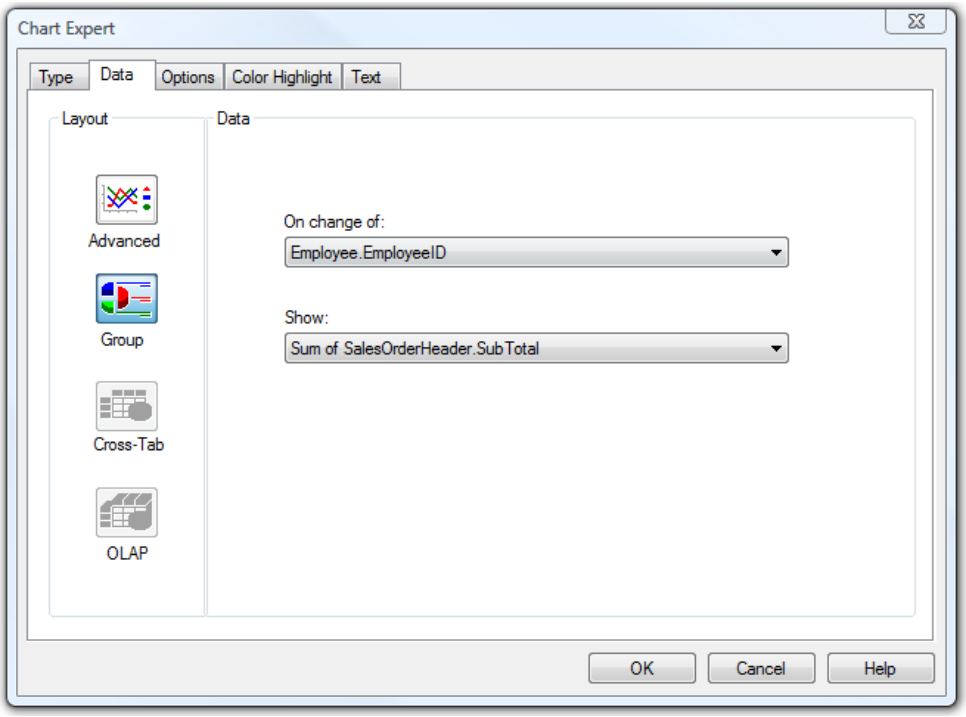
The Report Header

Now the report is providing the correct information before you begin formatting you should think about the business requirements for the report header. The report needs a pie chart on the front page with a text title that contains the parameter values for the report.

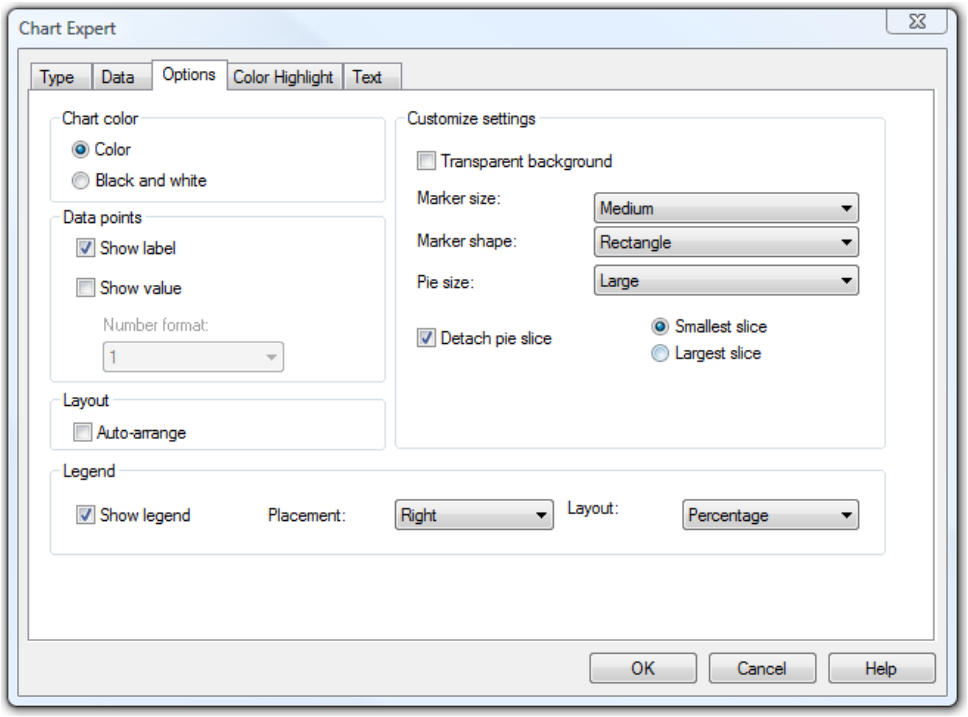
❖ Exercise 1.4 – Report Header

1. Insert a text object into the report header with an appropriate title for the report. Make sure you insert both parameter fields so their values at runtime are displayed. Format the title to stand out.

2. Insert a pie chart into the **Report Header** by clicking on **Insert Chart**; Using the **Chart Expert**, choose a pie chart and then make sure the correct data is selected as shown below:



3. Under **Options** tab you should choose to **show a label** and also **detach the largest piece of the pie** as shown below. Also uncheck the *Auto-arrange* option.



4. Add the title **Top Employees** to the chart, and format as you would like.

5. Finally, in the **Section Expert** choose to format the **Report Header** with a new page after (on the **Paging** tab) to make sure the **Report Header** is the front page of the report.

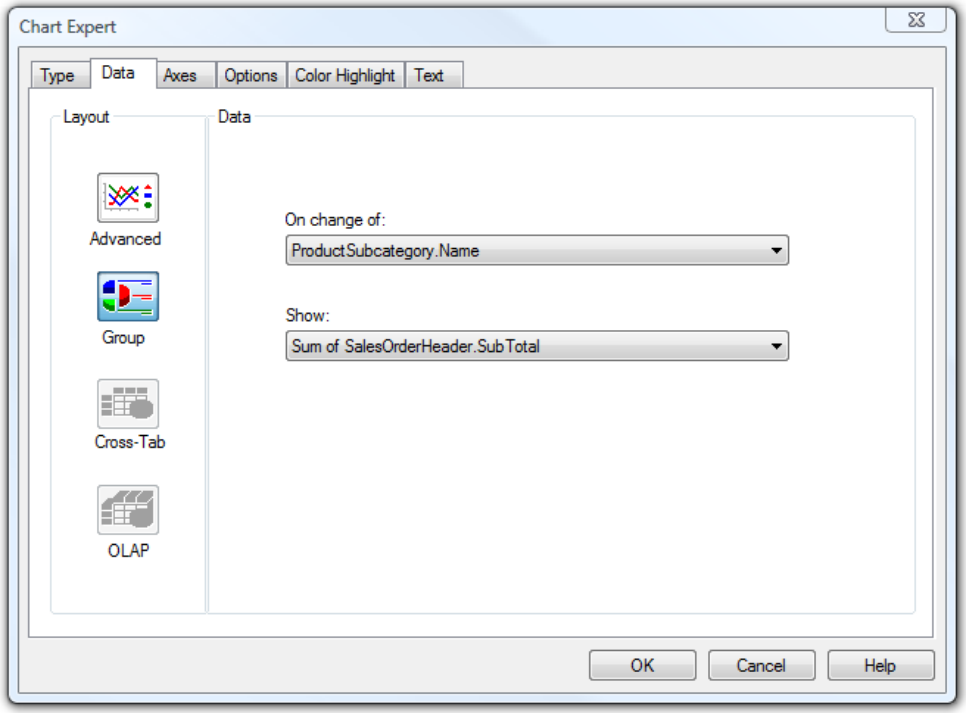
6. Save the report.

The Group Chart

As well as the chart in the report header there is also a chart in the main report for each employee. This chart is showing the product sub-categories sold for an employee.

❖ Exercise 1.5 – Group Chart

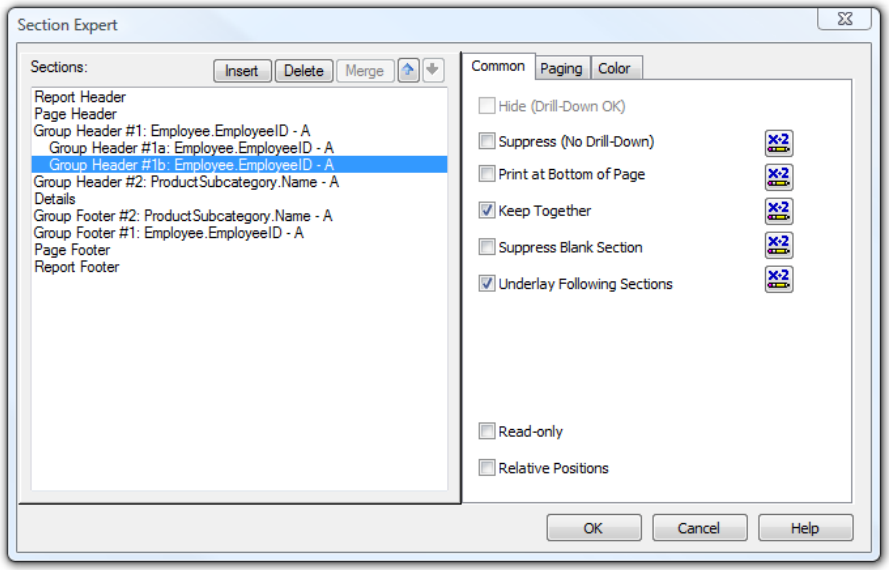
1. Click on the **Insert Chart** button and place the chart into **Group Header 1**. Open the **Chart Expert** and choose a bar chart; click on the **Data** tab to choose *SubTotal by ProductSubcategory.Name* as shown below:



An additional requirement is to have the chart beside the detailed data for the sales person. To do this, use will use the Underlay option in the section expert. However, you need to put the chart in its own section to prevent the group title from being underlayed as well.

2. Insert a new group header section for **Group 1**. Move the chart to **Group Header 1b**. Reformat the **Group Header 1a** to remove the excess white space.
Hint: use Fit Section.

3. In the Section Expert select **Group Header 1b** and choose to underlay the following section:



4. Save the report.

Your report is almost complete but needs further formatting to ensure any sales person who sold over $50,000,000 in a year needs to be recognized. You can achieve this with conditional formatting on the employee group footer adding a message when the sales are over $50,000,000.

❖ Exercise 1.6 - Conditionally Format the Report

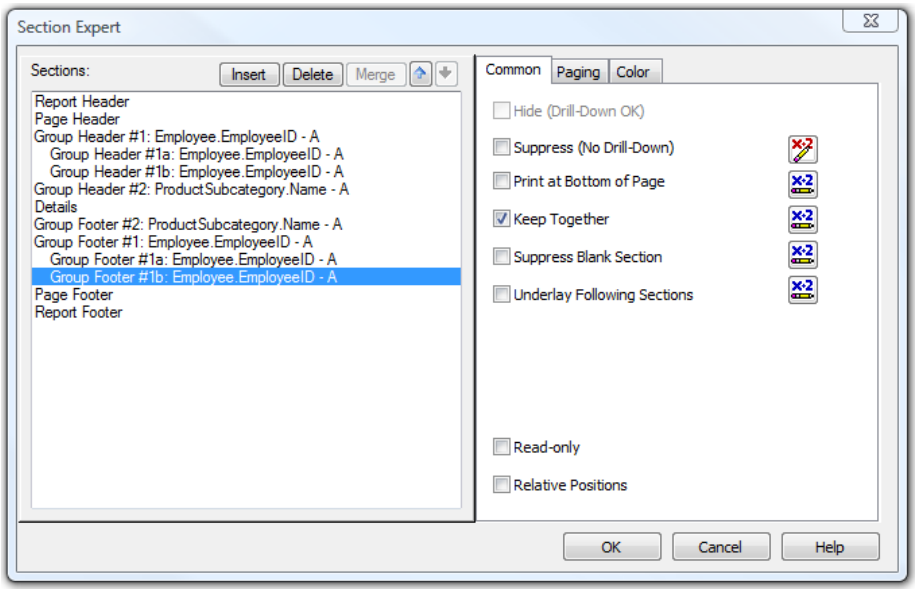
1. Return to the **Design** view. It is easier to add and format objects from Design view.

2. Insert an additional group footer section for **Group Footer 1** and add a text object to it with the sales person's first name embedded The text object should look similar to the following:

Congratulations {FirstName}, you sold more than $50 million dollars!

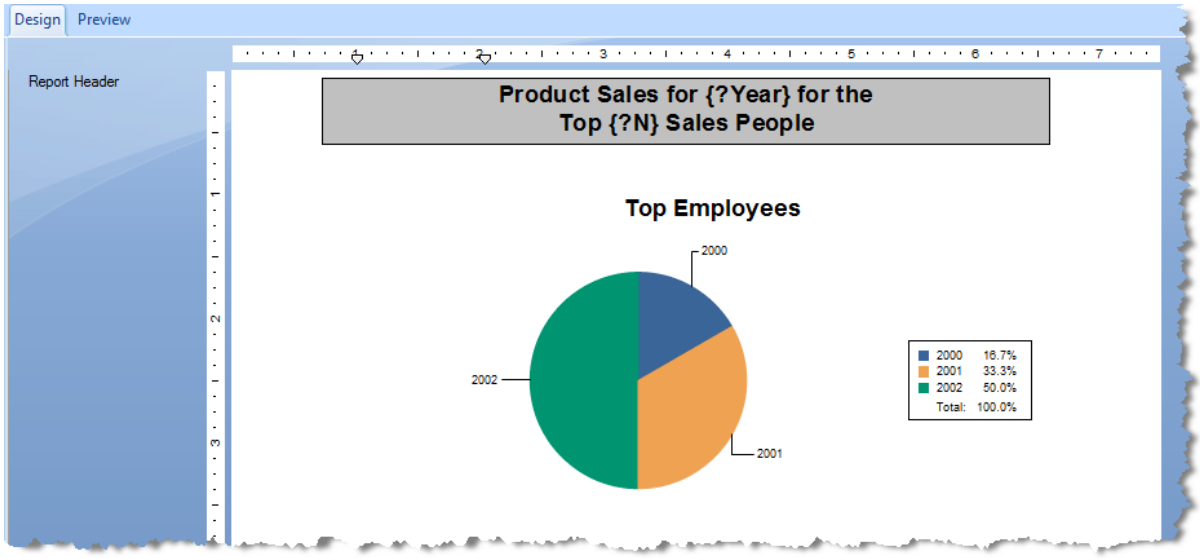
3. If you preview the report now you see that this text object displays for every sales person regardless of what they have sold. You now need to conditionally suppress the section. You need the section to suppress if sum of sales by Employee ID is less than $50,000,000. In the **Section Expert** add the following formula to the conditional box next to the *Suppress* option for **Group Footer #1b**:

   Sum({SalesOrderHeader.SubTotal},{Employee.EmployeeID}) < 50000000



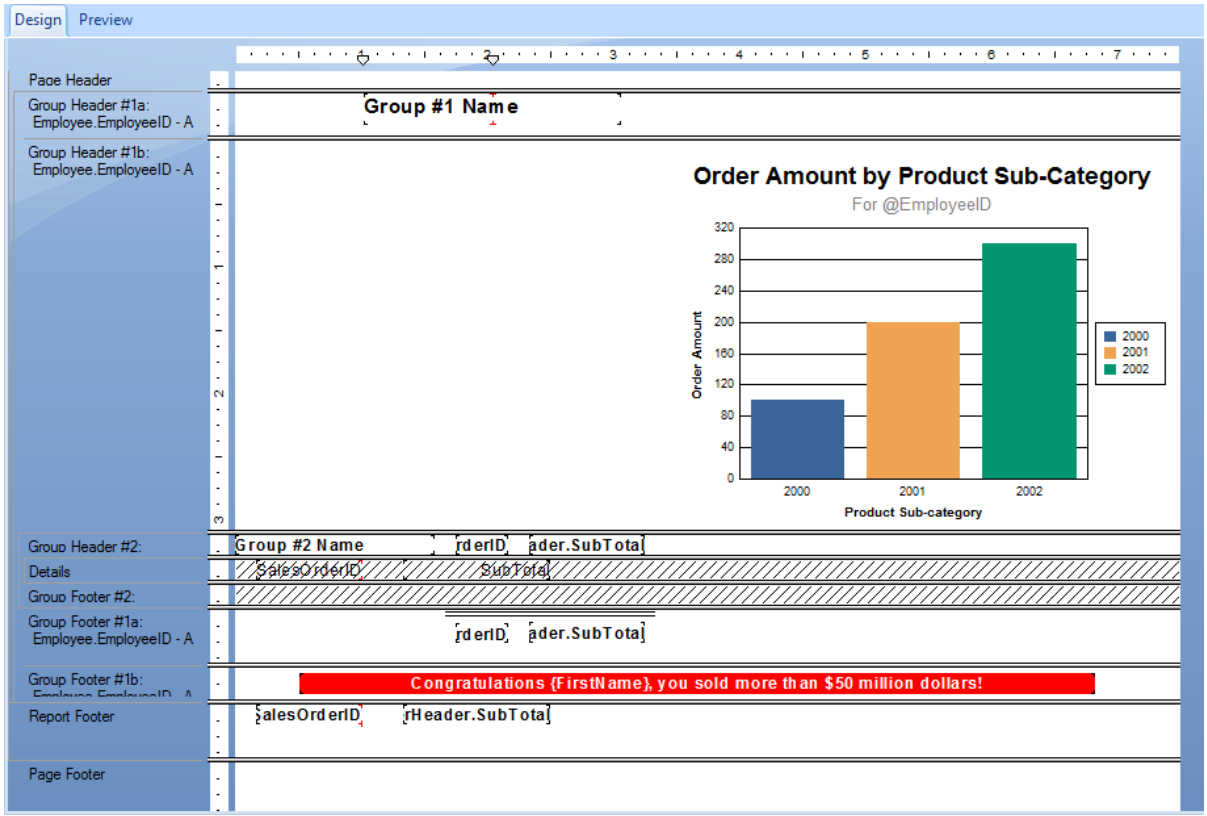
4. **Save** your report and run it with different values for N and the year. Make sure it meets the business specifications.

The Design view of your report should look as follows:



This is the Design View of the Report Header – note the chart is a place holder and does not show the data.

The Design view for the rest of the report should look as follows:



NOTES

Lesson 2
Advanced Database Concepts

Lesson Objectives

After completing this lesson, you will be able to:

- **Understand Database Access**
  Understand the different ways Crystal accesses the database
- **Understand more about Linking**
  Spend some time learning about the options in the Links tab of the Database Expert
- **Adding a table to a report multiple times**
  What happens when you need to add a table multiple times and why would you may need to
- **Database Location and Field Mapping**
  Learn how to change a database location, verify a database and map fields when field names change
- **Stored Procedures and Views**
  Writing a report against a stored procedure or view
- **Create reports from SQL Commands**
  Use Crystal Reports to generate reports from SQL commands

What is SQL?

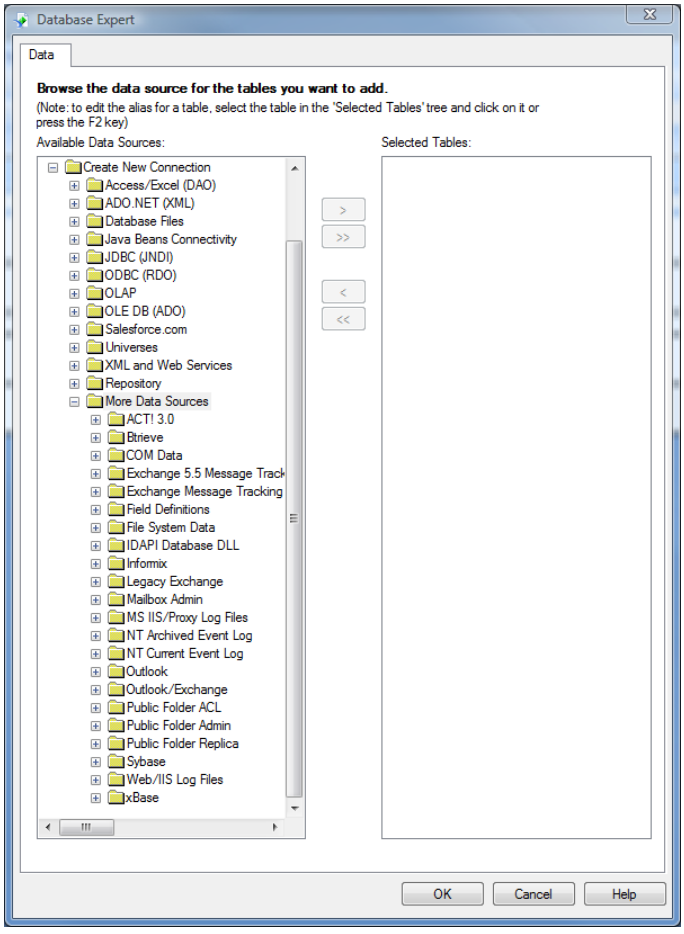
The letters SQL stand for Structured Query Language. SQL is the language with which you speak to the database. Although you may not know the language, Crystal Reports does, and converts many of the commands you execute to SQL language to communicate with the database. There are many client/server databases, the most common being Microsoft SQL Server, Oracle and IBM DB2.

Generally the database exists on a database server, which is designed to process SQL queries efficiently. For the best report performance, your goal as a report designer is to have the database server do as much work as possible. We'll look at different ways of achieving this during this lesson.

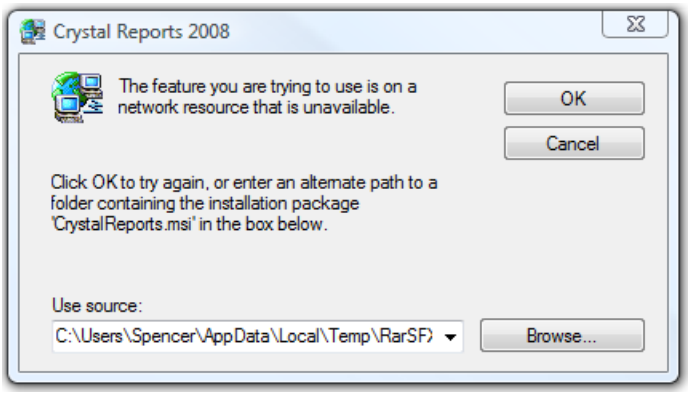
When designing a report the first thing you do is choose the SQL database. You have three choices on how you want to access the data. You can use a native database driver; an ODBC driver or an OLE DB driver. Let's look at these 3 choices.

Native Database Driver

When you expand the Create New Connection folder in the Database Expert and then expand the More Data Sources folder you will see a long list of databases that you can connect to directly. These are called Native Database Drivers and they use native communication to access the database; this will typically require database software to be installed locally. There may be a performance advantage to accessing the database directly. See the list of database drivers listed in the Database Expert below:



NOTE: You may need to install the native database drivers when you select a database for reporting. In this case you will need the Install CD or Microsoft Installer application (.msi) when you select the database type. For example if we select Informix, we see the following message (assuming you have not installed the Informix drivers earlier). In earlier versions, all drivers were installed by default but that is not the case in versions XI and above.



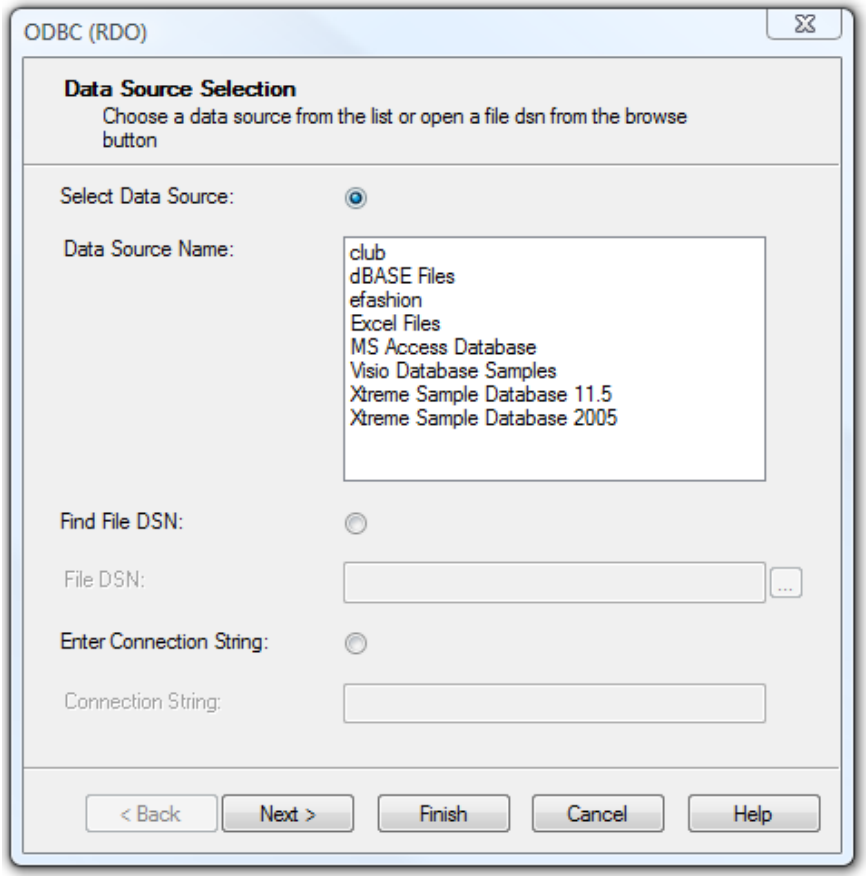
ODBC

Microsoft Corporation developed a standard method to connect to different types of data sources. This standard connectivity is known as Open Database Connectivity (ODBC). The benefit of utilizing ODBC is that any application needing to access a data source only needs one set of files through which to communicate. With hundreds of desktop applications currently in the marketplace, it would be a tedious process if there was not a standard method in which to extract data. The industry has accepted ODBC as the standard method for database connectivity.

With this in mind, companies that develop applications with data sources generally provide items referred to as ODBC drivers to allow external applications, such as Crystal Reports, to easily connect to the data source for data extraction. Crystal Reports has also generated some of the ODBC drivers for standard databases, such as SQL Server, Access, Oracle and Sybase. These drivers are shipped with the application and can be accessed when a new connection is established in the Database Expert. ODBC allows Crystal Reports to access data from more than one type of database management system. Together the SQL language and ODBC interface allow Crystal Reports a tremendous amount of flexibility in reporting from multiple data sources.

Before you can use Crystal Reports to access a database through ODBC you have to set up an ODBC data source. When you access data through an ODBC data source, the first step in the process is to logon to the ODBC data source. You MUST have all the appropriate Windows components (database DLLs and Windows Client Software for that database, etc.), network protocol drivers, and database drivers in place or the logon will fail and you will not be able to access the data. The server logon process may also require you to enter a user ID and password.

When you choose to create a new report you will see an ODBC (RDO) folder in the Database Expert. When you click on that you are taken to the following dialog to choose your data source:



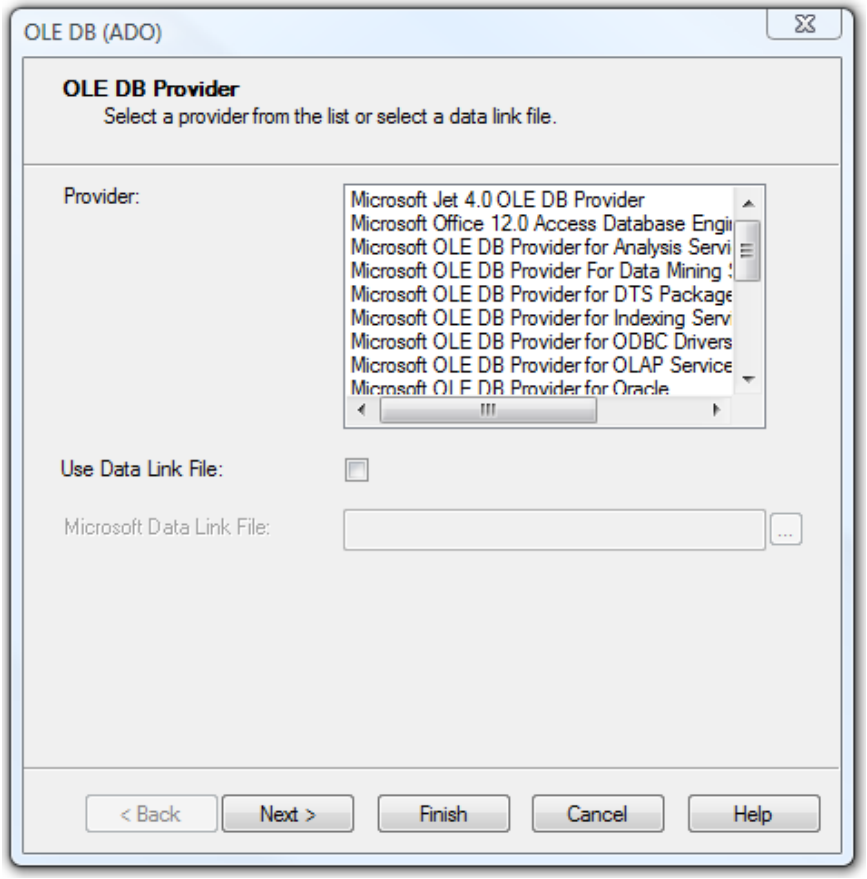
All the data sources here have been set up in the Windows ODBC Data Source Administrator application. If you do not see your source here then use the ODBC Data Source Administrator to add a source.

OLE DB

OLE DB was designed to build on the success of ODBC by providing another standard for accessing data. Whereas ODBC was created to access relational databases, OLE DB interfaces are designed to communicate with any data source including relational and non-relational data, such as Microsoft Excel spreadsheets, as well as e-mail, and text files. Using OLE DB it is possible to query an Excel spreadsheet in the same way you can query a database table.

Another difference from ODBC is that OLE DB does not require a data source to be defined on the client PC. All the needed information is stored as a connection string, and is saved as part of the application itself.

When you choose to create a new report you will see an OLE DB folder. When you click to expand that folder you are taken to the OLE DB dialog shown below:



NOTE: You may notice the abbreviations RDO and ADO after the ODBC and OLE DB folder names. RDO stands for Remote Data Objects and ADO stands for ActiveX Data Objects. These simply refer to the internal method that Crystal uses to connect to the ODBC and OLE DB APIs.

ODBC, OLE DB or Native

In general, OLE DB is the better choice. Microsoft strongly encourages the use of one of the native OLE DB Providers instead of the ODBC Provider. Native OLE DB Providers provide better application stability and performance. In addition, OLE DB provides more portability of a report by not requiring client-side configuration.

Native database drivers may be appropriate in certain situations for performance reasons, but require software to be installed on each client.

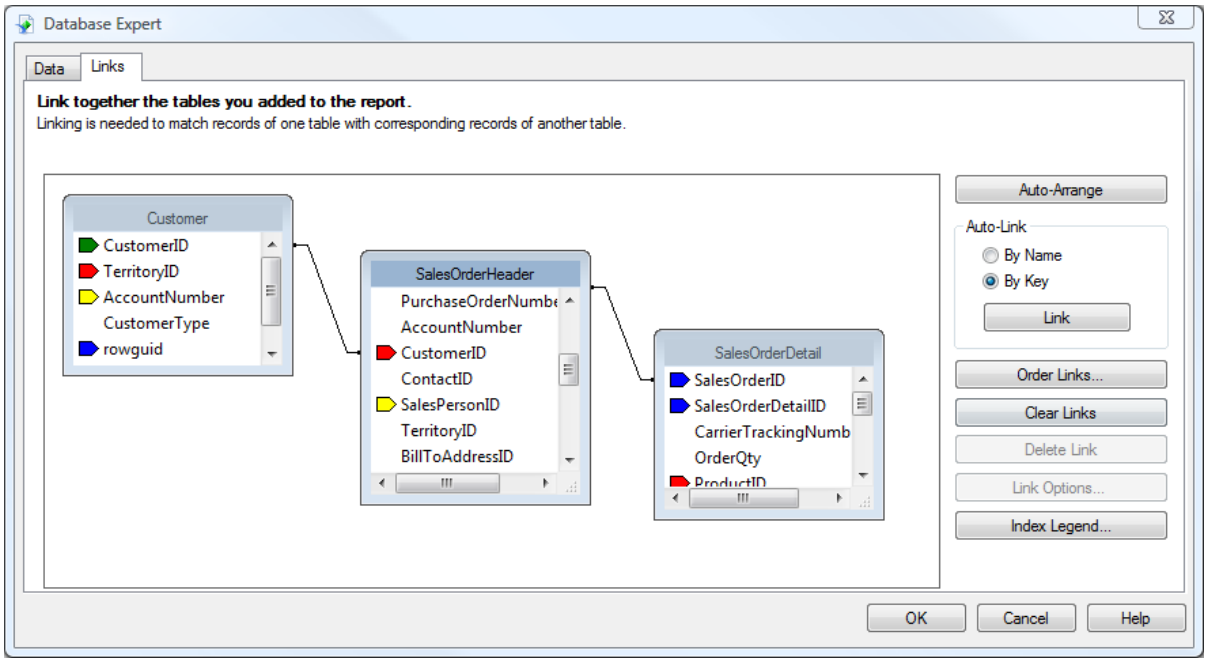
Using the Database Expert for Linking

In the previous courses you have learned that you can pull data from more than one table by linking the tables together on a common field. You link database tables so that records from one table match related records from another table. The Links tab of the Database Expert makes it easy to link tables together.

When you link two tables, you link from one table to another table. The from table is used as a primary table, while the to table acts as a lookup table, where records are looked up by the primary table. In a simple link, the database examines the first record in the primary table and finds all matching records in the lookup table. Once all matches have been found in the lookup table for the first record in the primary table, all matches in the lookup table for the next record in the primary table are found.

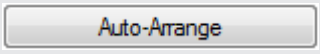
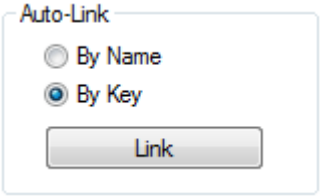
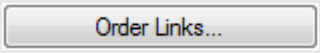
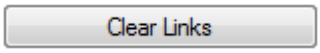
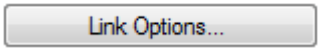
Crystal automatically displays the Link tab of the Database Expert when you select more than one table. If you choose to go into the Database Expert and add a table, Crystal will display the Links tab. Click on the Database Expert button to view the Links for a report, or select Database Expert under the Database menu.

The links tab of the Database Expert graphically shows the tables in the report and their links.



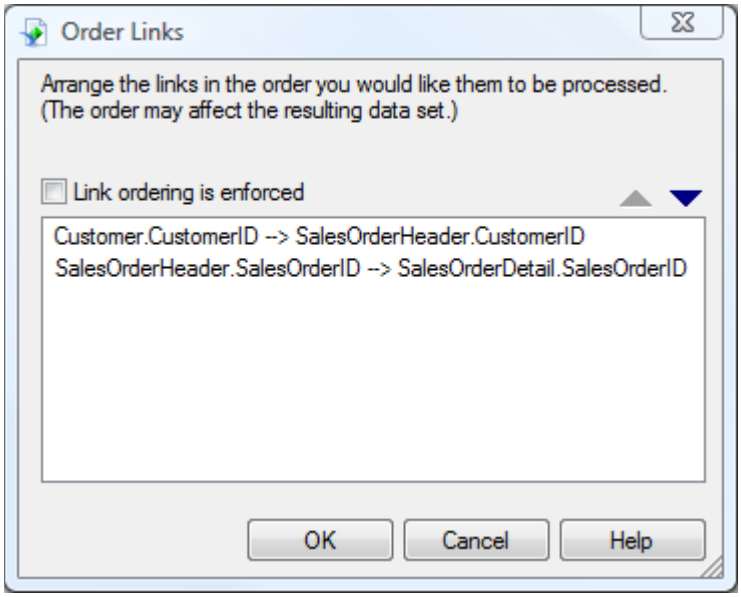
You can use the scroll bars at the right and bottom of the screen to view tables that may not be visible. You can also move any table by dragging its title bar and resize any table by dragging its border. This graphical image and Windows functionality lets you view link information so it is most helpful to you.

The Database Expert also has buttons at the right side of the dialog box and on the bottom that let you work with the links. The following table defines the options:

VISUAL LINKING EXPERT BUTTONS	
Button	**Description**
	Arranges the tables to fit in the most efficient manner inside the Links display area
	Use this area to choose a method of linking tables. Click By Name to link tables by their names or By Key to use foreign key information. Click the Auto-Link button to link tables automatically (by name or foreign key, as selected). If you choose the By Name option, Crystal Reports looks for fields that share a common name and data type to establish the link. If you choose By Key, Crystal uses external keys defined in the database to establish the links
	To activate this button, you must have more than two tables available for linking. When you click the button, the Order Links Dialog is shown – see below. Establish the link processing order you want for the available linked tables
	Deletes ALL the links between the tables in the Database Expert
	Removes the selected link. Select the link line, and then click the button. WARNING! There is no undo for this action. This button is grayed out until you have a link selected
	To activate this button, you must have a link line selected. Once you have a link selected, click this button to open the Link Options dialog where you can choose the join and link type for the selected link. This is explained further below
	Click this button to open the Index Legend dialog so you can see a key to the index indicators used in the Links display area. This dialog is explained further below
	Indicates a field that is indexed. See more about indexed fields and this legend below

Order Links Dialog

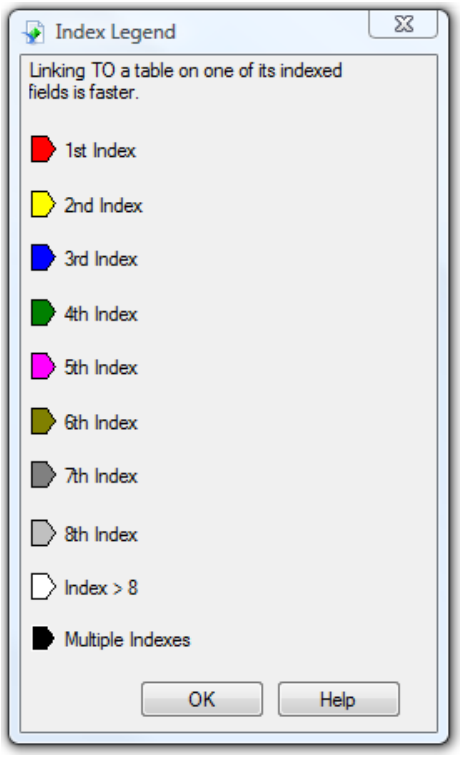
The Order Links dialog box appears when you click the Order Links button on the Links tab of the Database Expert. This button is only available when you have more than one link. Use this dialog box to specify the order in which Crystal Reports should process the table links. By default, the processing order matches the order in which the links appear on the Links tab. Use the arrow buttons in the Links Order dialog box to change the default order.



For example, if you have chosen the Customer, SalesOrderHeader, and SalesOrderDetail tables from the AdventureWorks database, In this case, the links will be processed first between the Customer/ SalesOrderHeader tables, then between the SalesOrderDetail/SalesOrderHeader tables. Different link orders may result in different data sets returned for use in your report. Click the Link ordering is enforced check box to ensure that Crystal Reports re-orders the link statements in the SQL query being generated.

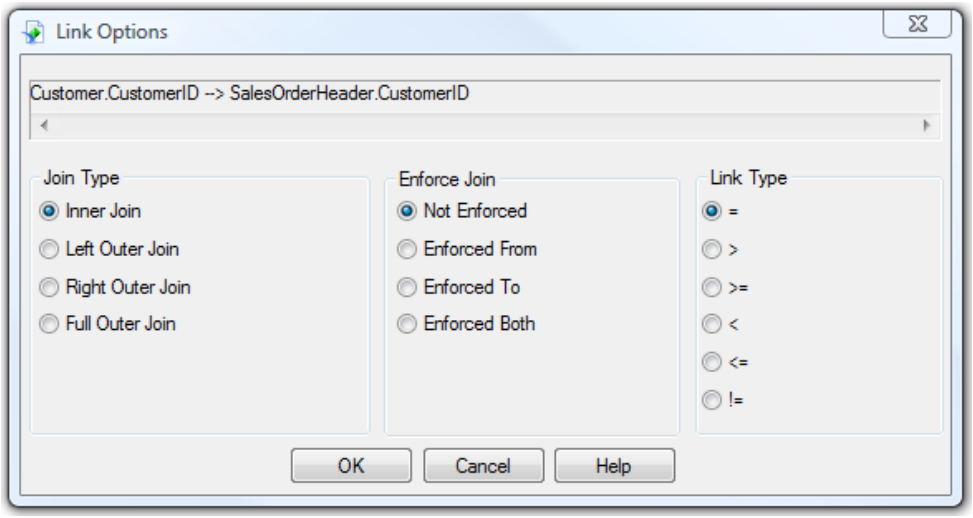
Index Legend Dialog

In earlier versions of Crystal Reports you would only see index legends when you were creating a report against a PC-style database or a database accessed using a native driver. If you saw the index legend you had to link to one of the indexed fields. In version XI and above you will see indexed legends even when using ODBC to access your database. Although you do not have to link to an index field it is often advisable to do so for performance. The index legend dialog shows you the colors used for your legends and what that means in terms of indexes.



Link Options Dialog

To see the link options for a link you need to have it selected and then click the link options button. In the Links Options dialog box you can specify the type of join and type of link you want to use when linking tables. These options determine how linked fields in two tables are compared when records are read. The default is an inner join with a link type of equal as we see below:



TIP: If your report is not returning the expected number of records reviewing the link type is a good place to start.

To work with links in the Database Expert, you are going to create a report with a number of tables, then add an additional table.

❖ Exercise 2.0 –Work with Links

1. Create a report against the **SalesOrderHeader** table and list **SalesOrderID**. You should see **31,465** records with no **SalesOrderID** duplication.

2. To add another table, go into the **Database Expert** under the **Database** menu or by clicking the **Database Expert** button.

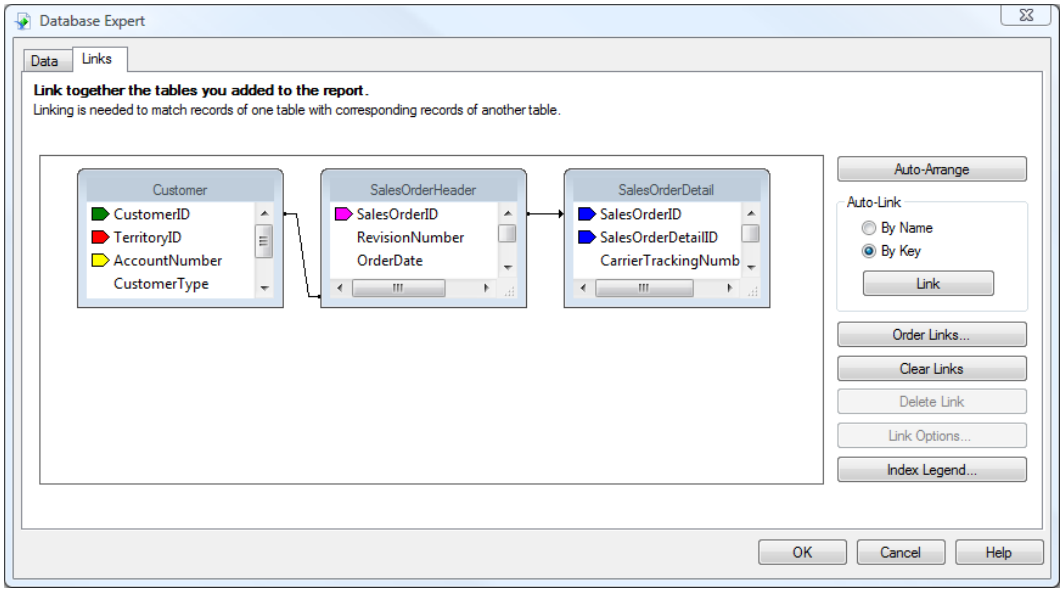
3. Select the **SalesOrderDetail** table and click the right arrow button > or double click the **SalesOrderDetail** table.

4. Click on the links tab and right click on the link, select *Link Options…*, and make the link a **left outer join enforced both ways**.

5. Notice now that if you refresh the data you have **121,317** records even though you have not added data from the **SalesOrderDetail** table.

6. Go to the **Database Expert** and double click the **Customer** table.
You do not really need the Customer table. You are adding it now so you can see how to order links.

7. When finished adding the **Customer** table, click the **Links** tab. **Reverse** the link between the **Customer** and **SalesOrderHeader** tables and between **SalesOrderHeader** and **SalesOrderDetail**; **Auto-Arrange** the tables so that your links look as follows:



8. Click on the **Index Legend…** button and compare to the tables. Click **OK** to close the Index Legend dialog.

9. View the SQL query (*Database|Show SQL Query…*)

10. Click on the **Order Links…** button, then change the order so that the **SalesOrderHeader/Customer** link is on top. Be sure the *Link ordering is enforced* box is **checked**.

11. **View** the SQL query again. Notice that the query has changed.

12. Close the report **without** saving.

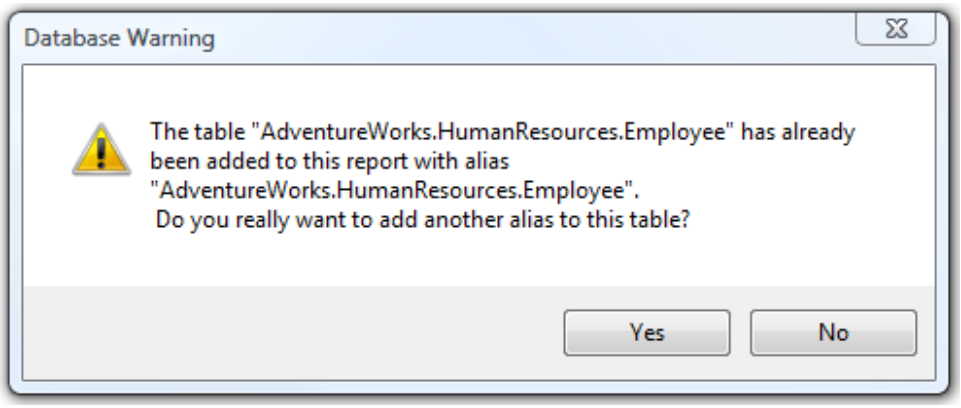
Adding Tables to a Report Multiple Times

Sometimes the way a database is designed, and the information you need to see in the report, requires that you add a table to the report more than one time. This can happen if you have code or ID fields that refer to text-based descriptions throughout the database. One example of this situation is available in the AdventureWorks database in the Employee table.

Suppose you wanted to see a report with employee information. You need to see the employee name, position, salary, and supervisor name. All this information is available in the Employee and Contact tables. The problem arises when you try to add the manager name. The ManagerID field displays a number. You want to see the name of the manager, not the ID. However, the name of every employee, including managers is in the Employee and Contact tables themselves.

You have a problem here. You need to link from the ManagerID field to the EmployeeID field so you get the correct name for the manager, but you cannot create a link from a table back into the same table. To solve the problem, you have to add the Employee and Contact tables to the report twice.

Crystal lets you add a table more than one time quite easily, but you cannot add it with the same name. You have to create an alias for the table so Crystal can distinguish it from the original. Crystal automatically warns you that a table already exists in the report as soon as you try to add the table the second time.



If you click the Yes button, Crystal adds the table again with a name Employee_1. If you were to add it again it would provide the alias Employee_2 for the table. To change the alias name simply right click on the Table name and choose Rename. You can type a new name such as Manager in this case or right click and choose to rename the table.

Once you have added the table with the alias, you must be very careful which table you pull fields from. The fields in the aliases table always refer to the link you used to get their data. For example, if you add the Employee table twice and create a link from the Supervisor ID to the Employee ID. Any fields from the second Employee table always refer to the supervisor, not the original employee.

❖ Exercise 2.1 – Add a Table Twice to a Report and Create an Alias

1. Start a new report as a *Blank report*, and then add the **Employee** and **Contact** tables, linking on the **ContactID** field.

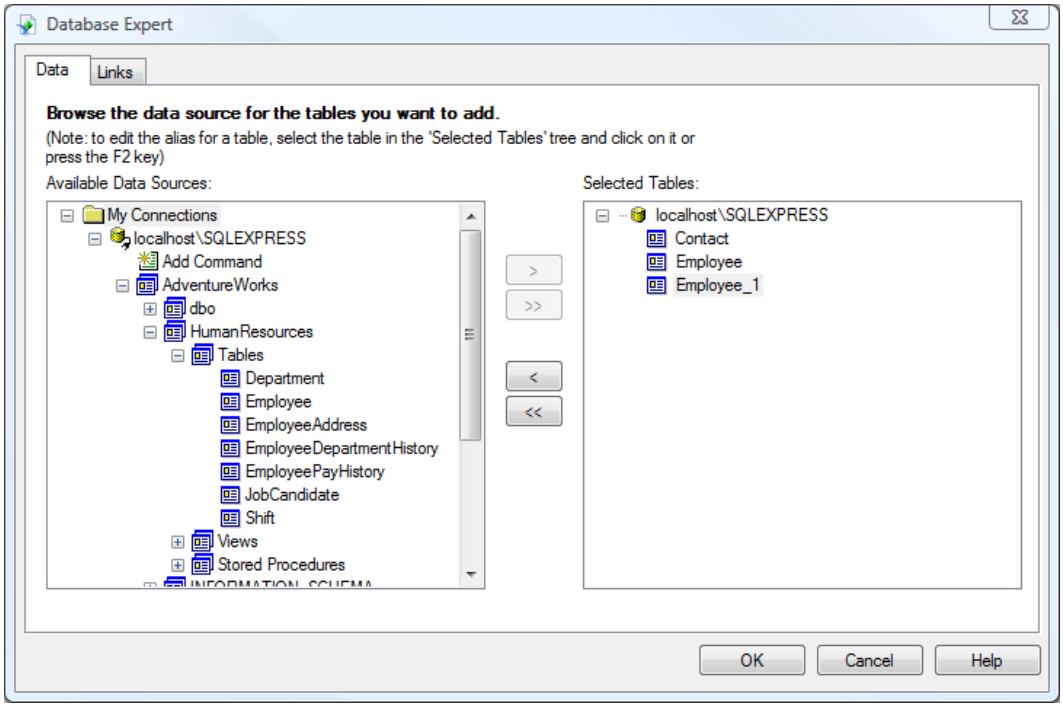
2. Add the **EmployeeID** field to the **Details** section.

3. Create a text object that has **Contact.LastName** followed by a **comma** and **space** and then **Contact.FirstName** . Add to the **Details** section and resize the field to be about 1.5" long. Add a column title of **Employee Name**.

4. Add the **Title**, **HireDate** and **ManagerID** fields (from the **Employee** table) to the **Details** section. Format the **HireDate** to display the date only (03/01/1999). Resize the **Title** field to be 1.5" long. Set the Page orientation to **Landscape**.

5. Save the report as **Employees and Managers.rpt**. Preview the report.
Notice that you get 290 employees and that the ManagerID field is a number, not a name. You need to add the Employee and Contact tables again and link on the ManagerID field to get the manager name.

6. Go to the **Database Expert** and double click on the **Employee** table on the left side. Click **Yes** when you see the warning. Notice you now have a table in the right hand column named **Employee_1**.



7. Change the alias from **Employee_1** to **Manager**. Right click on its name in the **Database Expert**, select the **Rename** option and then type the new name.

8. Double click on the **Contact** table on the left side. Click **OK** when you see the warning. Notice you now have a table in the right hand size column named **Contact_1**.

9. Change the alias from **Contact_1** to **Manager_Contact**.

10. Go into the **Links** tab and notice Crystal created 5 links from **Employee** to **Manager**, and 2 links from **Employee** to **Manager_Contact** using Auto Linking. Select a link between the **Employee** and **Manager** tables, RIGHT-click, then select **Remove All Links** from the menu.

11. Repeat with a link between the **Employee** and **Manager_Contact** tables.

12. Drag a link from the **ContactID** field in the **Manager** table to the **ContactID** field in the **Manager_Contact** table. Ensure that the join type is **Inner**.

13. Drag a link from the **ManagerID** field in the **Employee** table to the **EmployeeID** field in the **Manager** table. Ensure that the join type is **Inner**. Click **OK** to exit the Database Expert.

14. Create a new formula called **Manager Name**. Concatenate the **FirstName** field from the **Manager_Contact** table with a **space**, then concatenate with the **LastName**. The formula should look like this:

    { Manager.FirstName} + " " + { Manager.LastName}

15. Add the **Manager Name** formula to the **Details** section after the **ManagerID** field. Then save (**Employees and Managers.rpt**) and preview the report.
    N*otice that you only see 289 records; one record is not appearing in the report.*

Changing the Join Type for a Link

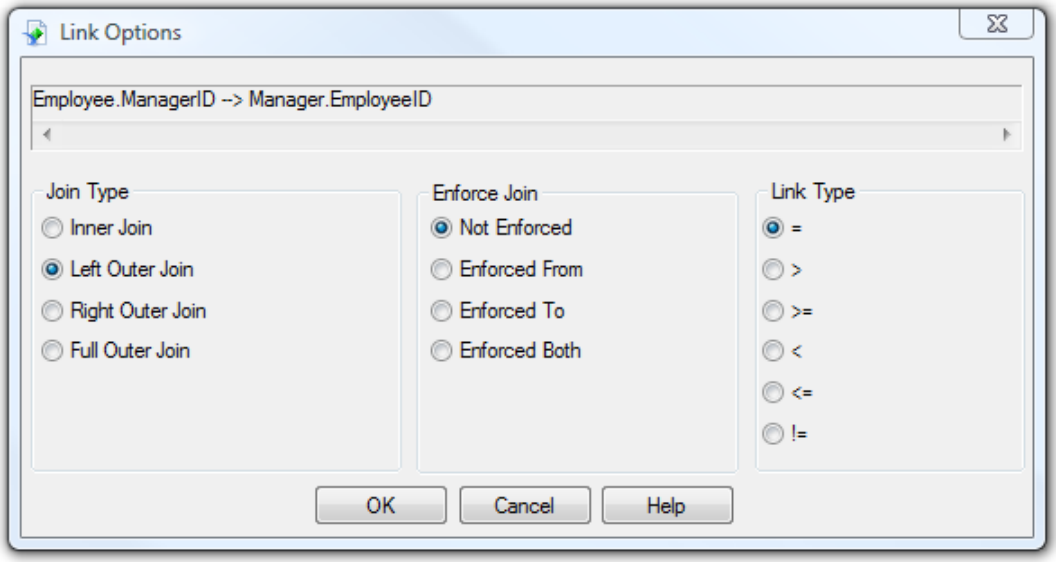
Why are there only 289 records? You should have noticed that there are not as many records as were in the report before you did the link. The reason for this is that the two tables are being linked with an Inner Join.

Remember that an Inner Join only pulls records that have matching data in both tables. Also, remember that if there is a blank in the linking field, it does not match anything. Therefore, any records that did not contain a Manager ID will not show up in the report using the Inner Join

In this case Crystal pulls from both tables ONLY if there is matching data and the result is a 289 record dataset.

The way to retrieve all the Employee records is to use a Left Outer Join instead of an Inner Join. A Left Outer Join pulls all records from the left table in the link, but only matching records from the right table in the link. Since you are linking from the Employee table (it is the left table), a Left Outer Join will display all employees whether they have a manager or not. Alternatively you can enforce the equal join.

To change the join type on a link, open the Database Expert, then RIGHT click the link and choose Link Options. Alternatively, you can select the link line, and then click the Link Options button.



NOTE: The Join Type is very important and will impact the records you see from the database as illustrated above.

❖ Exercise 2.2 – Change the Join Type

1. Open **Employees and Managers.rpt**.
2. Click the **Database Expert** button, then select the **Link** tab.
3. Select the link between the **Employee** table and the **Manager** table.
4. Click the Link Options... button.
5. Click the **Left Outer Join** check box and click **OK**.
6. Select the link between the **Manager** table and the **Manager_Contact** table, and change to a **Left Outer Join**.
7. Close the **Database Expert**; click **OK** again to refresh the data.
   Notice that you now have 290 records. Ken Sanchez, the CEO, does not have a manager (Employee ID 109 on page 3). Both the Manager ID field and the Manager Name formula are blank.
8. Save (**Employees and Managers.rpt**) and close the report.

Database Changes

It may be that the database you designed your report against has moved or changed. For example you may be writing reports against a test database and then want to have them run against the production database – in this case the database location has changed. You do not need to rewrite your report, but can tell Crystal Reports to point to a different database.

What if field names change or fields are added? Crystal Reports provides functionality to help work through those issues as well. You can simply map fields rather than have to change field names throughout the report and remember which formulas used those fields. If Crystal Reports detects changes in the field names for example it will automatically invoke the field mapping function. There are 3 functions to help you manage database changes:

- Verify Database
- Set Datasource Location
- Field Mapping

Database Verification

Some changes in the database will not be noticed by Crystal Reports even when you choose to refresh the report. The only time Crystal Reports will notice a change is if a field that is used in the report has been renamed or deleted. You may know as a Report Designer that there is a new field in the database that you would like to use. In order to see this new field you need to verify the database.

Choose Verify Database from the Database menu, then if the database has more fields now than it had when the report was first created, Crystal Reports attempts to identify and use the correct fields from the new database. The aim is to print an unchanged version of the report even though the underlying database has changed. If the database has fewer fields now than it had when the report was first created, it uses those fields that are still available when it prints the report and ignores those that are no longer available. When working with the database verification process, Crystal will either update the database automatically or display the Map Fields dialog box if user intervention is required.

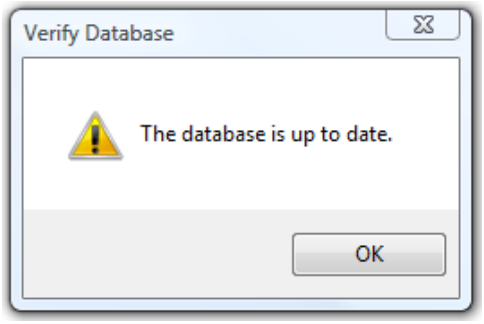
Crystal automatically updates the database if the following changes are detected:

- New fields have been added to the database
- If fields that were in the database have been deleted and these fields have not been used in the report
- Order of the fields have changed in the database
- Fields in the database have new data types

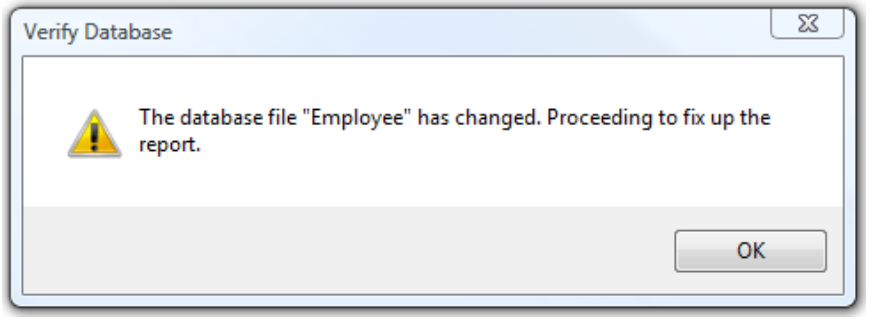
Crystal will need intervention from you through the Map Fields dialog box if one of the following situations occurs:

- A field used in the report has had a name change in the database
- The database has changed to a more robust data source, such as upgrading from a PC data source to a SQL data source

Verifying a database is simple. Choose Database|Verify Database. You will then either see the following if the database is up to date



or the following if the database has changed:

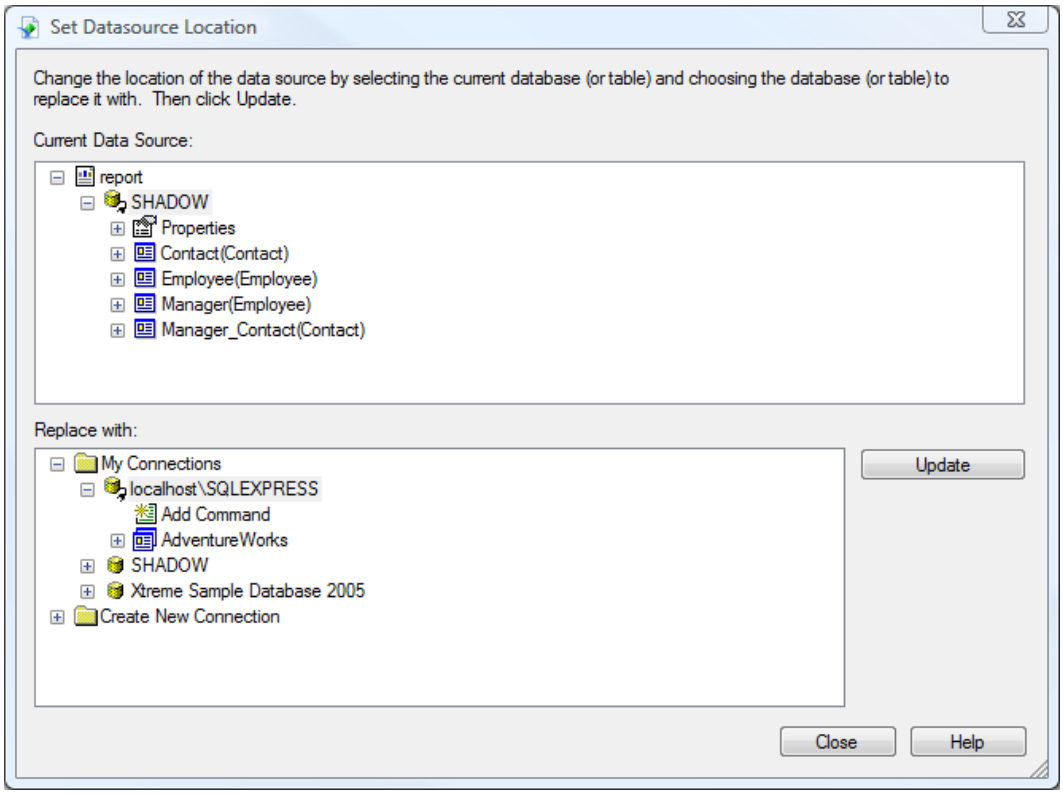


Click OK and Crystal Reports will read the database structure and make any changes. After clicking OK you will see the changes such as a new field in the Field Explorer.

If you are working with a database that changes frequently you will need to make sure you verify database often. You can set an option for the Report to Verify the Database on First Refresh. This means that Crystal Reports verifies the database the first time you refresh it in a session. By setting this under Report Options… under the File Menu then you are setting it for this report. If you would like to set this for all reports then set the option on the Database tab under File|Options... The options available are to verify on first refresh (which is on by default) and verifies the database once per session when you first refresh the data. You can also choose to verify when the database driver is upgraded.

Set Datasource Location

If you need to change the connection to a database (database driver), change the physical location of a database or table or change the database to be used, then you need to use the Set Datasource Location under the Database menu. Clicking this option shows the following dialog:



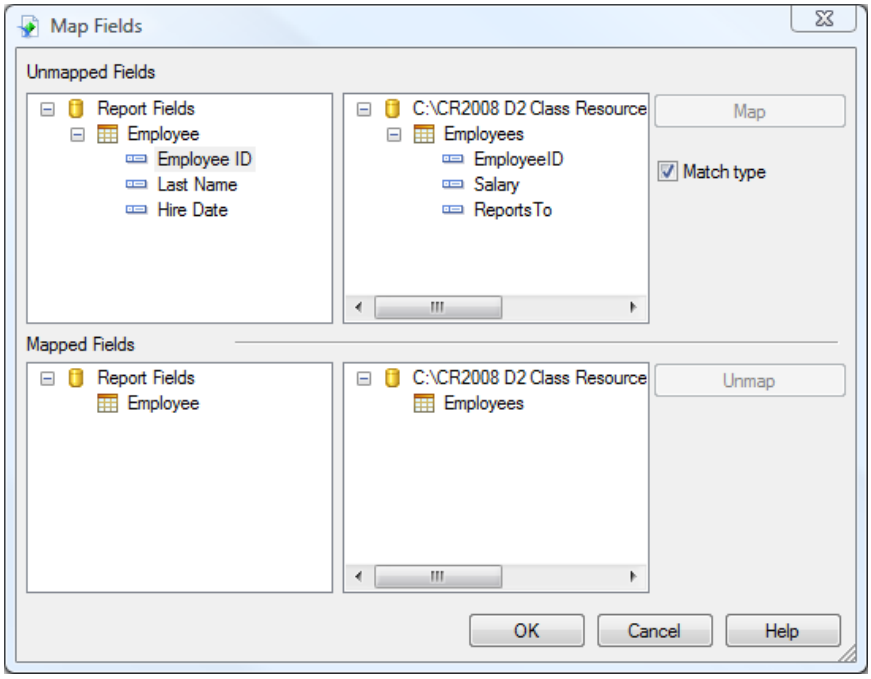
At the top we have the Current Data Source: which lists the database and tables currently used in the report. The bottom gives us options for replacing either a database or a table. Highlight the table in the report and then highlight the table name for the replacement. Click Update and the changes will be made. If you do not see the Database in the Replace with: box then you may need to create a connection. If the field names are not different then this is all you have to do.

In any of the above cases if there are field name changes then you need to remap database fields as discussed in the next section.

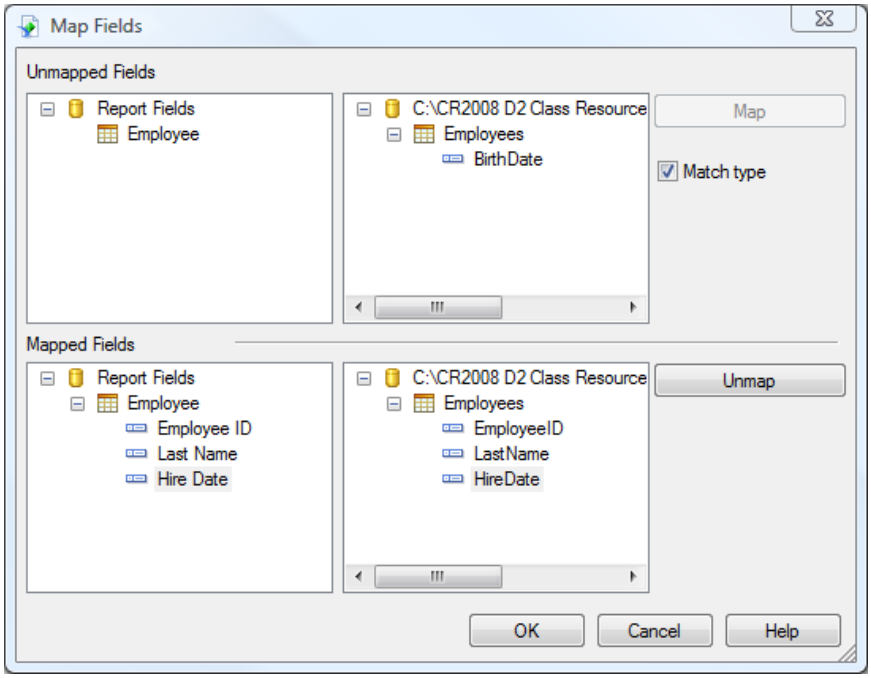
Re-mapping Database Fields

You can use the Map Fields dialog box to link report fields to their corresponding database fields when you have made changes to the structure of the database. This feature is also useful if you create a report based on one database from a template of another report based on some other database that has the same table and field structure.

When you first create a report, the report draws its fields from the existing database. If you change the structure of the database after you create the report, then Crystal needs to adapt the report to the new structure. If field names have changed, then the Field Mapping function is used to map the new names for Crystal. You cannot choose to map fields but the dialog is shown when Crystal detects a need for field mapping as shown on the next page:



You notice two upper lists that show the fields in the report that are unmapped. The bottom box shows the mapped fields as determined by Crystal Reports. If these are not correct then you can simply unmap them. In this case Employee ID, Last Name and Hire Date are unmapped. Highlight the report field and then highlight the field name to map. Above we are mapping Last Name to LastName (no space). Each time you click Map the fields are moved into the mapped fields dialog. The Match Type check box means that the only fields shown in the map to list are those of the same data type. The two lower boxes display the mapped fields in the report. Once we have mapped all the fields then the dialog will look as shown below.



NOTE: On the Design tab of the report, you should notice that the field is still labeled with its original name. The label does not change until you refresh the data, preview the report, or select the field. You must manually change the column titles.

Views

Up until now you have used Crystal Reports against database tables linking them so you can report against more than one table. However you may also create reports against Views. A view is a query against the underlying database. For example a view may simply join together a few tables saving the report designer from having to join tables. The AdventureWorks database has several views; we will use the view vSalesPerson in several exercises. The SQL for this view is shown below:

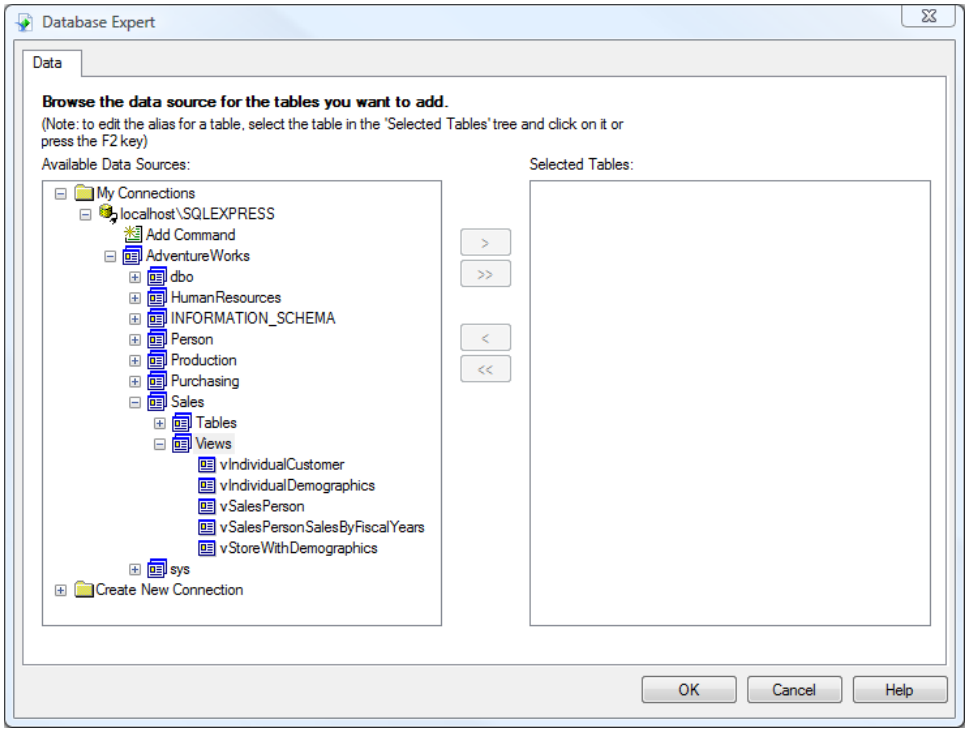
vSalesPerson

```
SELECT    s.SalesPersonID, c.Title, c.FirstName, c.MiddleName, c.LastName, c.Suffix,
          e.Title AS JobTitle,  c.Phone, c.EmailAddress, c.EmailPromotion, a.AddressLine1,
          a.AddressLine2, a.City, sp.Name AS StateProvinceName,
          a.PostalCode, cr.Name AS CountryRegionName, st.Name AS TerritoryName, st.Group
          AS TerritoryGroup, s.SalesQuota, s.SalesYTD, s.SalesLastYear
FROM      Sales.SalesPerson AS s INNER JOIN
          HumanResources.Employee AS e ON e.EmployeeID = s.SalesPersonID
          LEFT OUTER JOIN Sales.SalesTerritory AS st ON st.TerritoryID = s.TerritoryID
          INNER JOIN Person.Contact AS c ON c.ContactID = e.ContactID
          INNER JOIN HumanResources.EmployeeAddress AS ea
                ON e.EmployeeID = ea.EmployeeID
          INNER JOIN Person.Address AS a ON ea.AddressID = a.AddressID
          INNER JOIN Person.StateProvince AS sp
                ON sp.StateProvinceID = a.StateProvinceID
          INNER JOIN Person.CountryRegion AS cr
                ON cr.CountryRegionCode = sp.CountryRegionCode
```

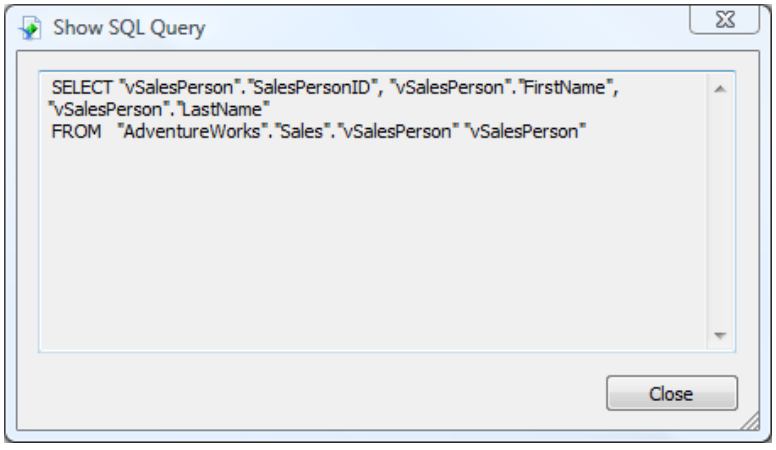
Views are a dynamic, virtual table of the data unlike stored procedures that return a dataset on which you can report; a view is not part of the physical database. The view is an SQL query that is executed when you refresh the report.

Views are often created to provide pre-linked tables for commonly used tables (such as the vSalesPerson view) to speed report creation, or created to provide more complex queries, such as SQL UNIONs or complicated groupings, which can make report development much simpler.

When you start a new report you expand the Views folder and will get a list of views which can be used as the report source as shown below.



As with a table you click on the button to move the view into Selected Tables column. If you choose to view the SQL you will see the view treated as a table.



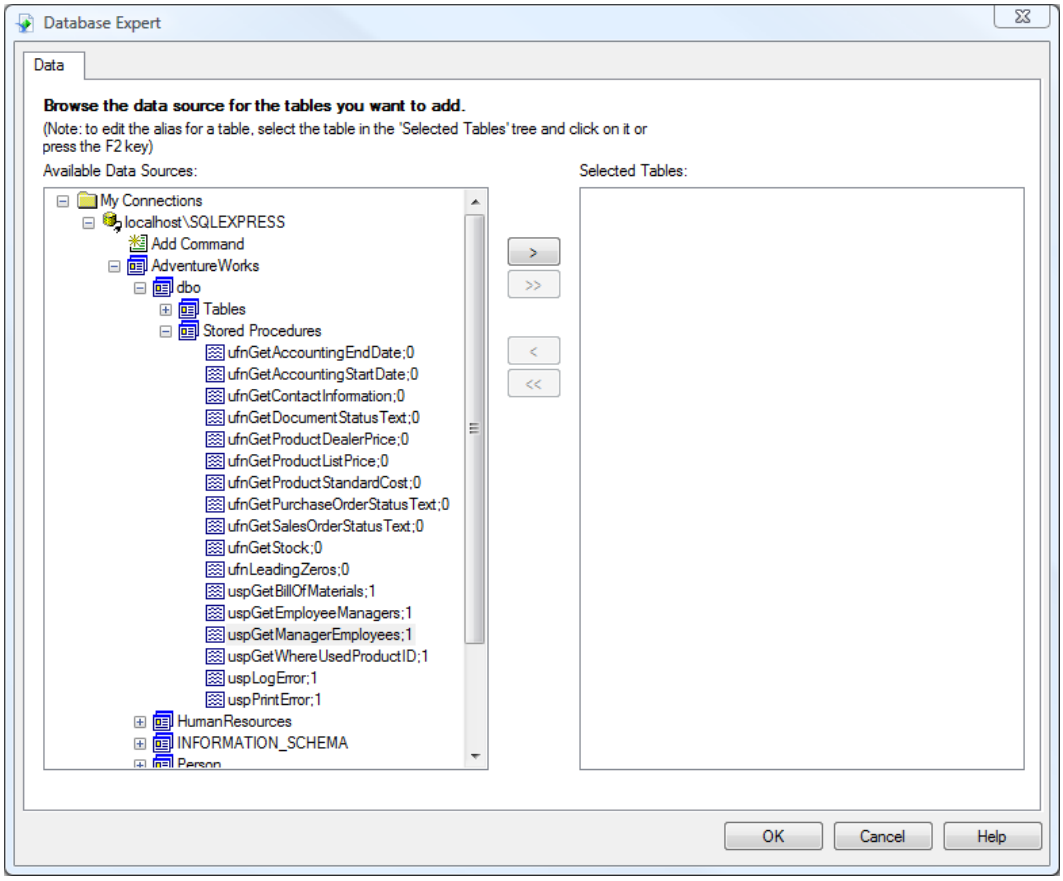
Stored Procedures

In addition to the common relational database attributes (tables, fields, records, and so on) many SQL DBMS systems support stored procedures. A stored procedure is a compiled SQL program consisting of one or more SQL statements. A stored procedure can be used to define an SQL query that you can use over and over again. Furthermore, variables, conditional expressions, and variable arguments can be defined in the stored procedure so that you are prompted to provide information before the procedure is executed.

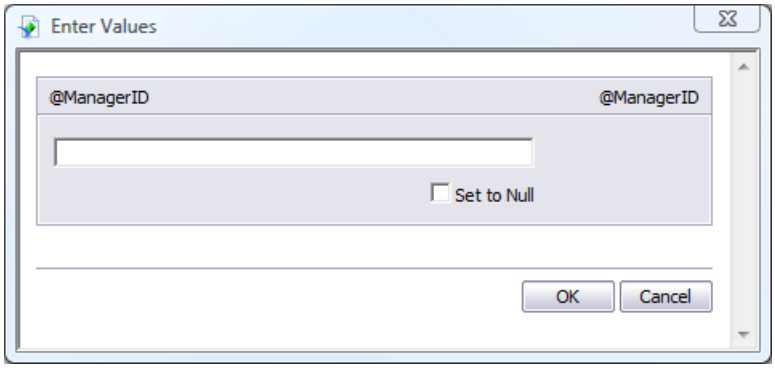
Since stored procedures can return a result set, they can provide a specific set of data when executed. In fact, Crystal Reports allows you to execute a stored procedure on an SQL database and use the returned data to design a report. If the stored procedure is designed to prompt a user for information to base its query on, Crystal Reports will prompt you for that information when you select the stored procedure for your report.

The AdventureWorks database has a stored procedure that you will report on in the following exercise; it is called uspGetManagerEmployees and returns a recursive listing of managers and their employees, starting with the Manager ID passed in as a parameter:

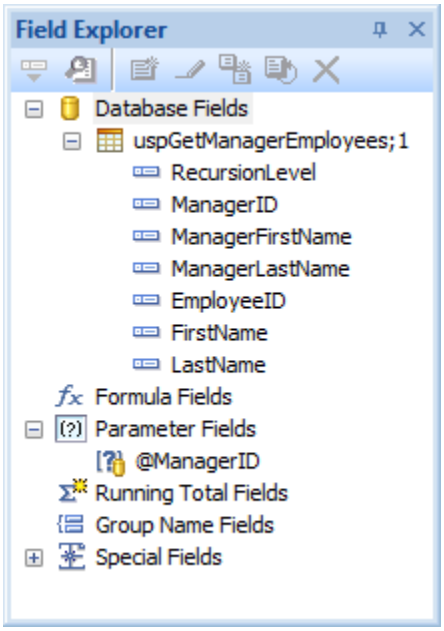
When you connect to the database you will see a folder of stored procedures like you do for tables and views as shown below.



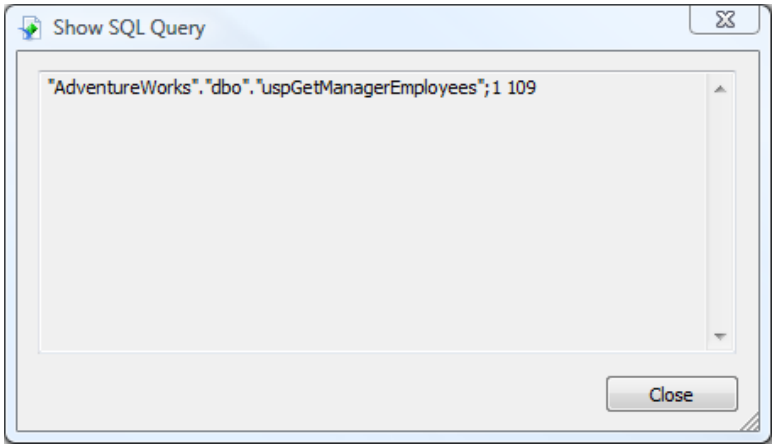
You select a stored procedure just like you select a table. However, if the stored procedure is parameterized, as it is in this case, you will be prompted for a value before you even begin designing your report. The dialog is the same as if you had a parameter in your report and were being prompted for it.



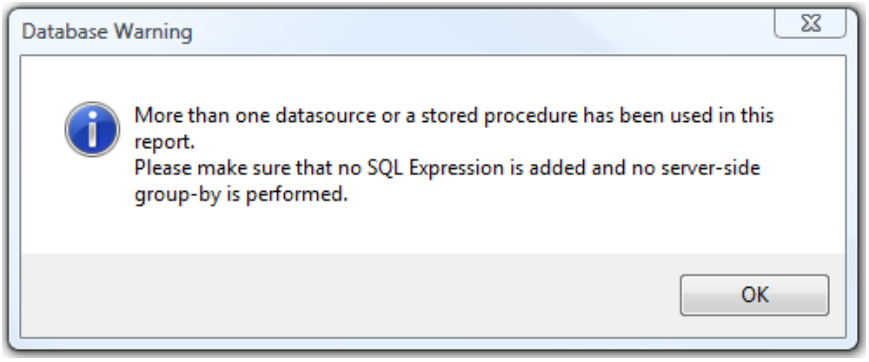
In the Crystal Reports designer the fields available for your report are those in the Stored Procedure above. You also see the parameter field in the Field Explorer.



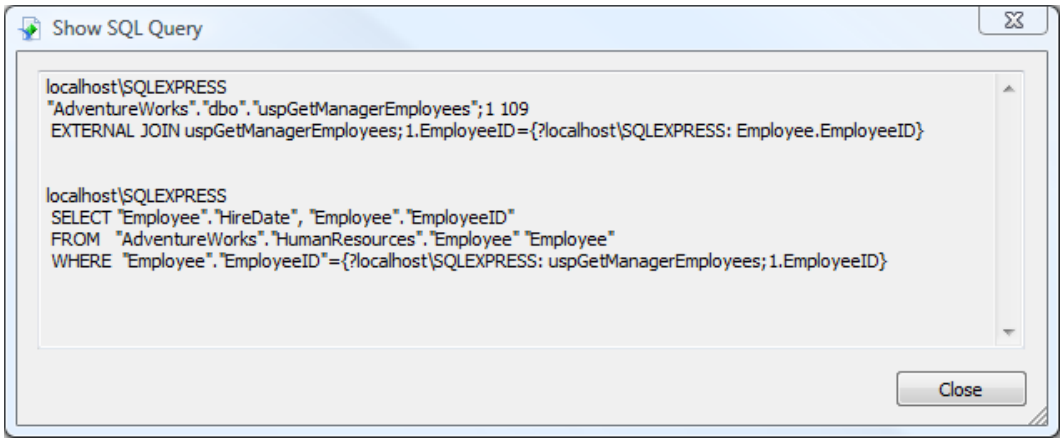
Having placed some fields on the report you should look at the SQL – it is simply a call for the stored procedure with the parameter value. In order to see the SQL behind a stored procedure you need to go into the database.



You can add additional tables to the report if needed. When you first add tables to a report with a stored procedure you will get a warning as shown below:

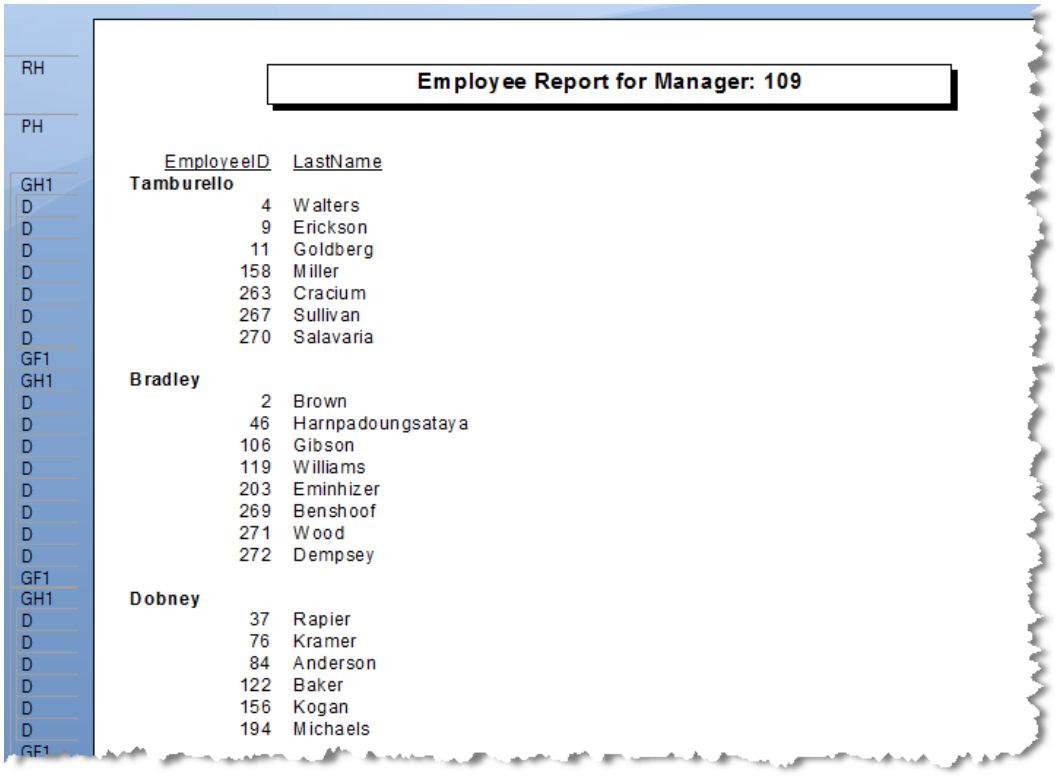


As long as you do not use SQL expressions or server side grouping you will be fine. The SQL below shows the SELECT statement when we add tables to the report.

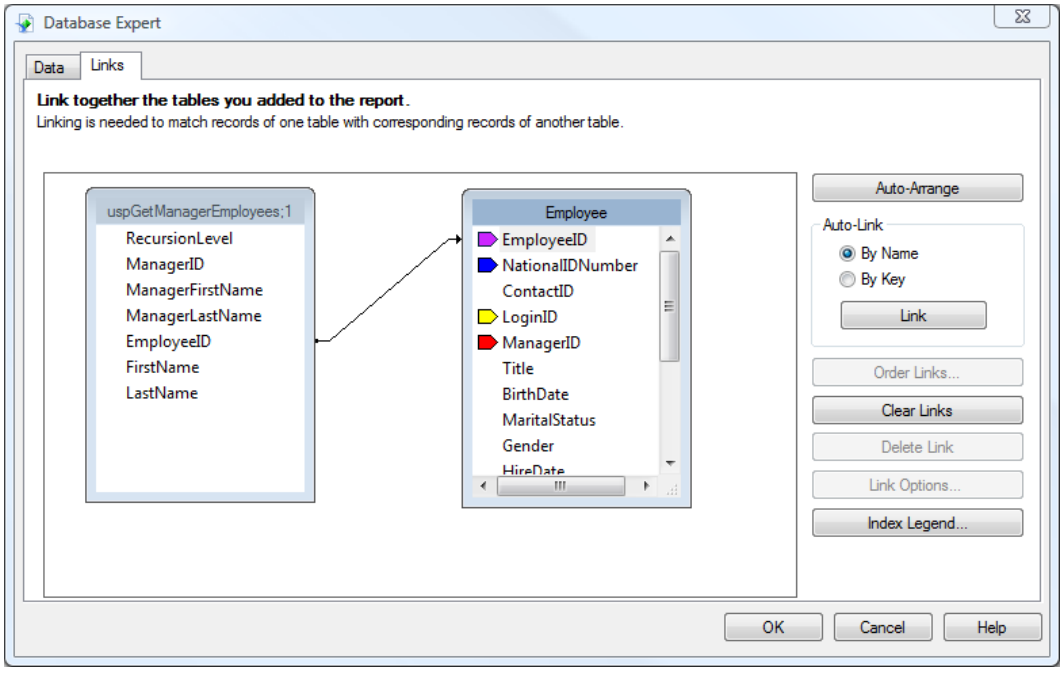


❖ Exercise 2.3 –Creating a Report on a Stored Procedure

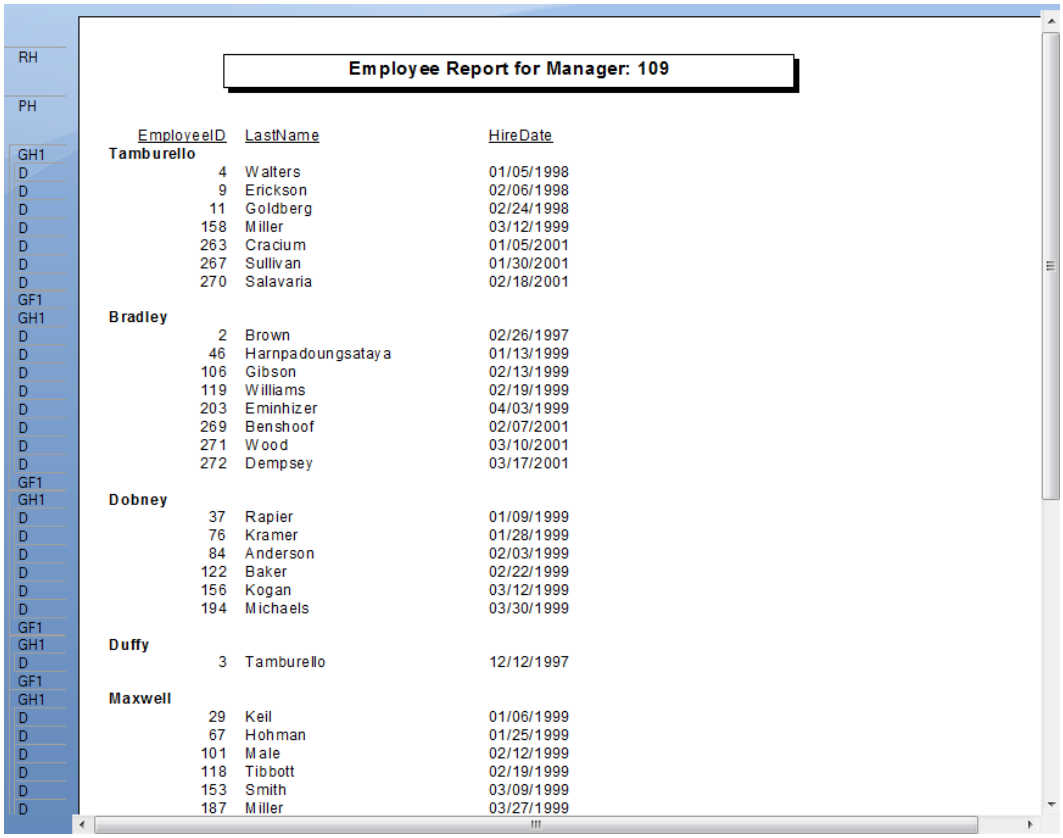
1. Start a *Blank report* and choose the **uspGetManagerEmployees** (under the **dbo** section) stored procedure for the report.
2. Select a **ManagerID** of **109** (this is the CEO).
3. Group by **ManagerID** showing **Manager.LastName** as the **Group Name** and add **EmployeeID** and **LastName** to the **Details** section.
4. Create a title showing the **Manager ID** chosen.
5. Refresh the report and notice you are prompted for a new parameter field value. Your report should look as follows:



6. Add the **Employee** table to the report. Clear the links and link the tables on **EmployeeID** as shown below.



7. Put Employee HireDate on the report. Your report should look as follows:



8. Save the report as **Report from Proc.rpt**.

SQL Commands

Report Designers used to be able to directly edit parts of the SQL generated by Crystal or use the Crystal SQL Designer to create an SQL query which could be used to create a report. Starting with version 10, the functionality of the Crystal SQL Designer is no longer available.

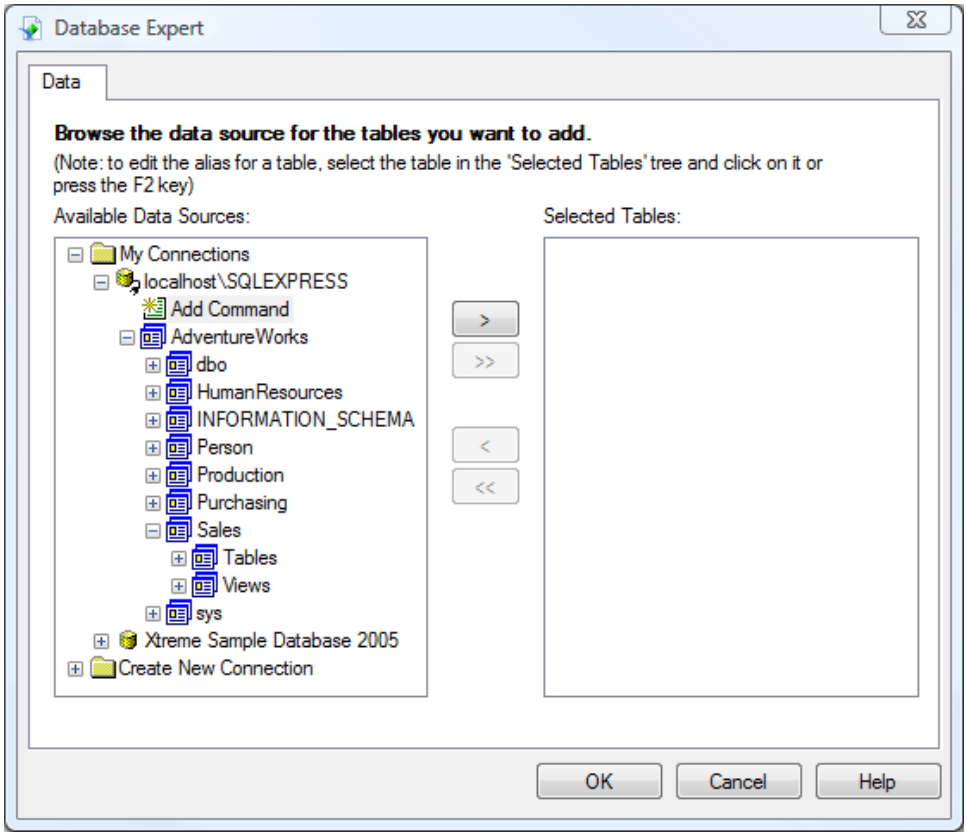
In Crystal Reports XI and above you can write an SQL command, which is able to perform any function within SQL. Once you create an SQL Command you can write a report against it – it looks like a table in the Database Expert. You can also add an SQL Command to the Enterprise Repository so that other report designers can use it to base their reports on.

Some of the situations that can be solved by SQL Commands:

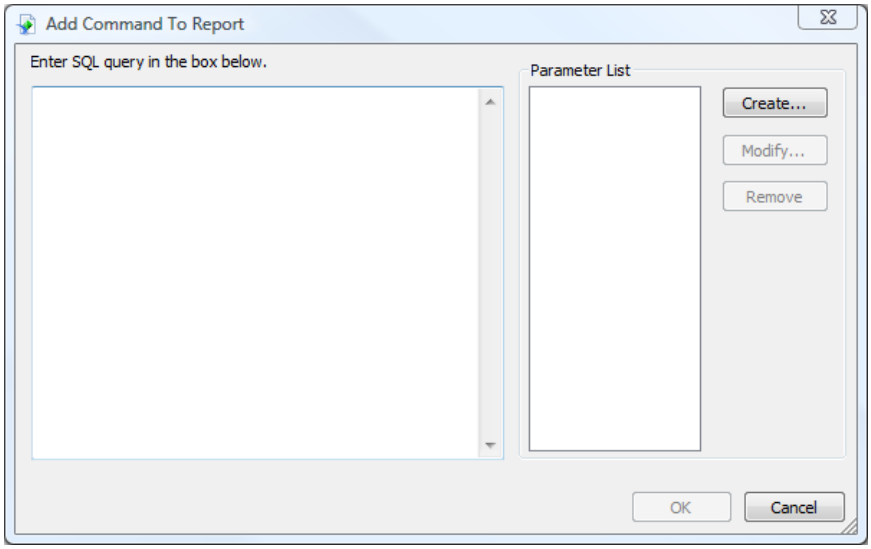
- Creating UNION queries
- Filtering on an inner join
- Allowing users to change tables, fields and join types by parameter
- Create "views" when you don't have rights to create database views

Creating a SQL Command

You create or edit an SQL command from the Database Expert either in the first stages of creating a report once you have chosen a database or from the Database Expert within the report. Once you have logged onto the database you will notice the Add Command option as shown below:



Once you select the Add Command option you will see the following dialog:



This dialog box is where you type the SQL for the command. There is no syntax checking of the SQL, so it is easy to make mistakes. Notice you can add parameters to the command as well as adding the command to the repository from this dialog. When you click OK to add the command, the command is submitted to the database and the syntax checked. At this time you will get an error message if there is a problem with the SQL command. The command is added to the tables list with a title of "Command" – press F2 and hold your mouse button over the name until it is in edit mode at which time you may change the name of the SQL Command.

To view or edit an SQL Command right click on the name and choose view or edit. To delete an SQL Command, simply left arrow click and remove it from the list of tables. If the edit option is not available then the command is connected to the repository (part of Business Objects Enterprise XI). Login to Enterprise and disconnect the command from the repository and then choose edit.

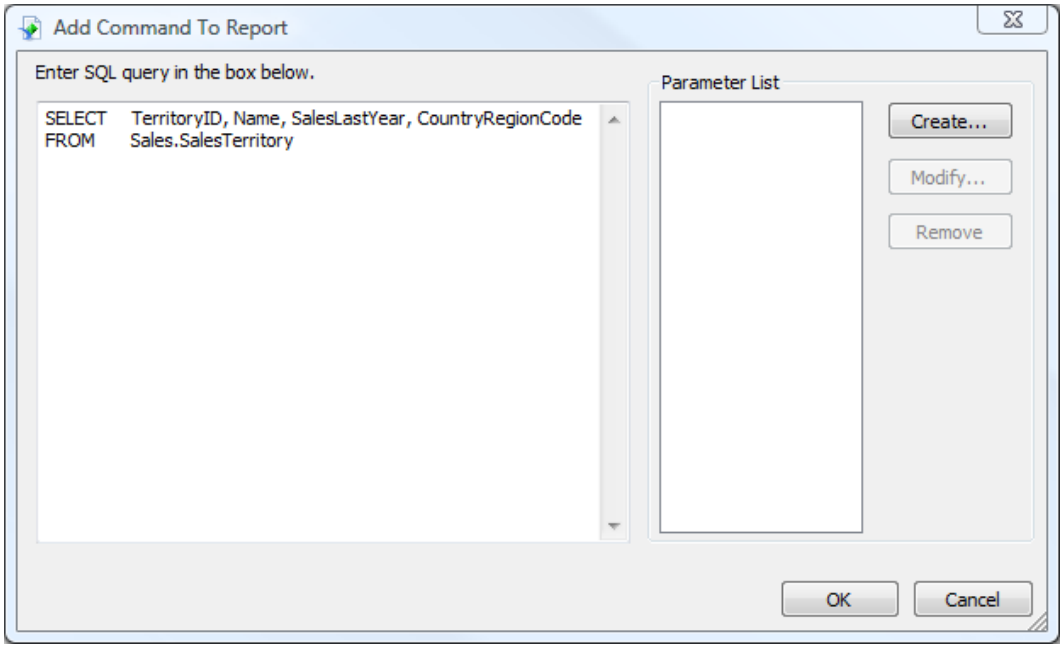
WARNING: Once a SQL Command is deleted, the SQL cannot be recovered; it will not be available under the list of tables and views on the left side of the Data tab.

TIP: It is highly recommended that you develop the SQL in another tool provided by the database vendor where you can have the syntax checked and then copy it into this dialog.

❖ Exercise 2.4 - Creating a SQL Command

1. Start a new report using a *Blank report*. Open the **AdventureWorks** data source and then DOUBLE-click on the **Add Command** item.
   The Add Command to Report dialog box will now open.

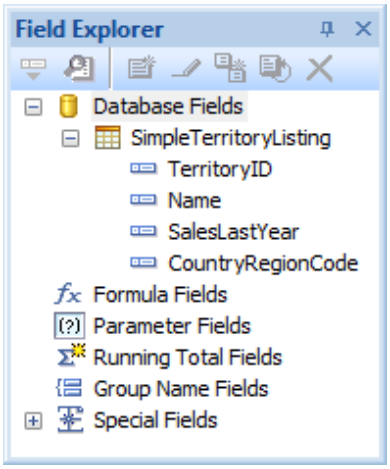
2. Write a SQL command that selects **TerritoryID**, **Name**, **SalesLastYear**, and **CountryRegionCode** from the **SalesTerritory** table (this is a simple SQL command and probably too simple to make using it beneficial). Click **OK** after entering your SQL statement. Your SQL Command dialog should look as follows:



3. Notice the command is listed in the tables listing as **Command**. Press **F2** and change the name to **SimpleTerritoryListing** and click **OK**.

   NOTE: No spaces are allowed in the Command name, underscores '_' are OK.

4. In your **Field Explorer** you will see the command as a table with the selected fields listed underneath as shown below.



From here you design your report exactly as you would against a database table. Insert **Name** and **SalesLastYear** on the report and group by **CountryRegionCode**.

5. Save the report as **SQL Command.rpt**.

Adding a Parameter to the Command

Having created an SQL command it will probably be most useful if it is parameterized. An SQL command with parameters will prompt the user for input, such as department or date range, when the command is added to the report or the report is refreshed. To add a parameter you need to do two things:

1. Create the Parameter in the Add Command to Report or Modify Command dialog box

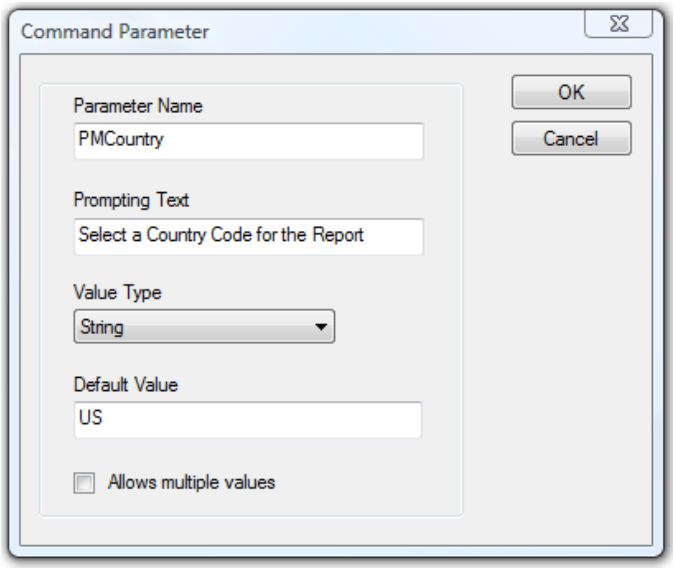
2. Place the parameter in the SQL statement.

At the very least you need to choose a name for the parameter and a value type. The value type needs to match the data type of your selection i.e. if you are creating a parameter field that selects a date range then the data type of the parameter needs to be date. You can also enter prompting text, which is always useful, and a default value. You do not have the same choices for parameter fields such as multi value or a pick list but we can change this in the report itself.

❖ Exercise 2.5 - Parameterize your SQL Command

1. Open the **SQL Command.rpt** report that we created. Open the **Database Expert,** right click on the command and choose **Edit Command**.
We are going to create a parameter field so that the user can select a country for the report.

2. Create a parameter field for country with a default of **US**. Click the **Create…** button and then complete the following dialog options:

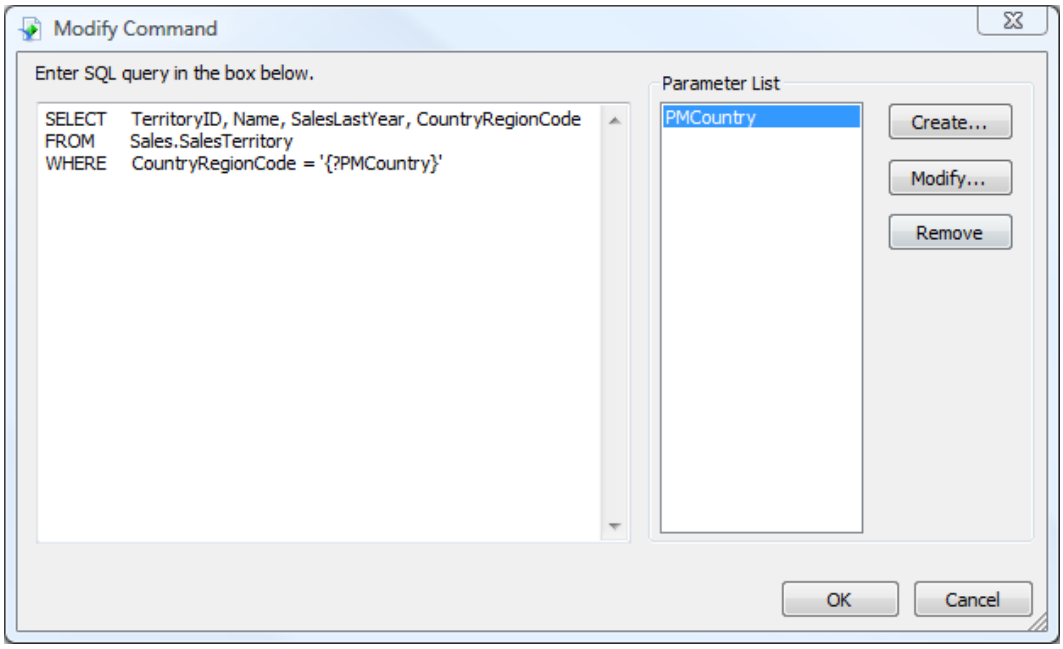


Click **OK** to close the Command Parameter dialog box when finished.

3. We now need to add the parameter field to our SQL query. In the **Modify Command** dialog box, press ENTER at the end of the last line and type

   WHERE CountryRegionCode =

   and then DOUBLE-click the newly created parameter, **PMCountry**. (Don't forget to add the single quotes around the parameter: **'{?PMCountry}'**) Your SQL command dialog should look as follows:



4. Click **OK**

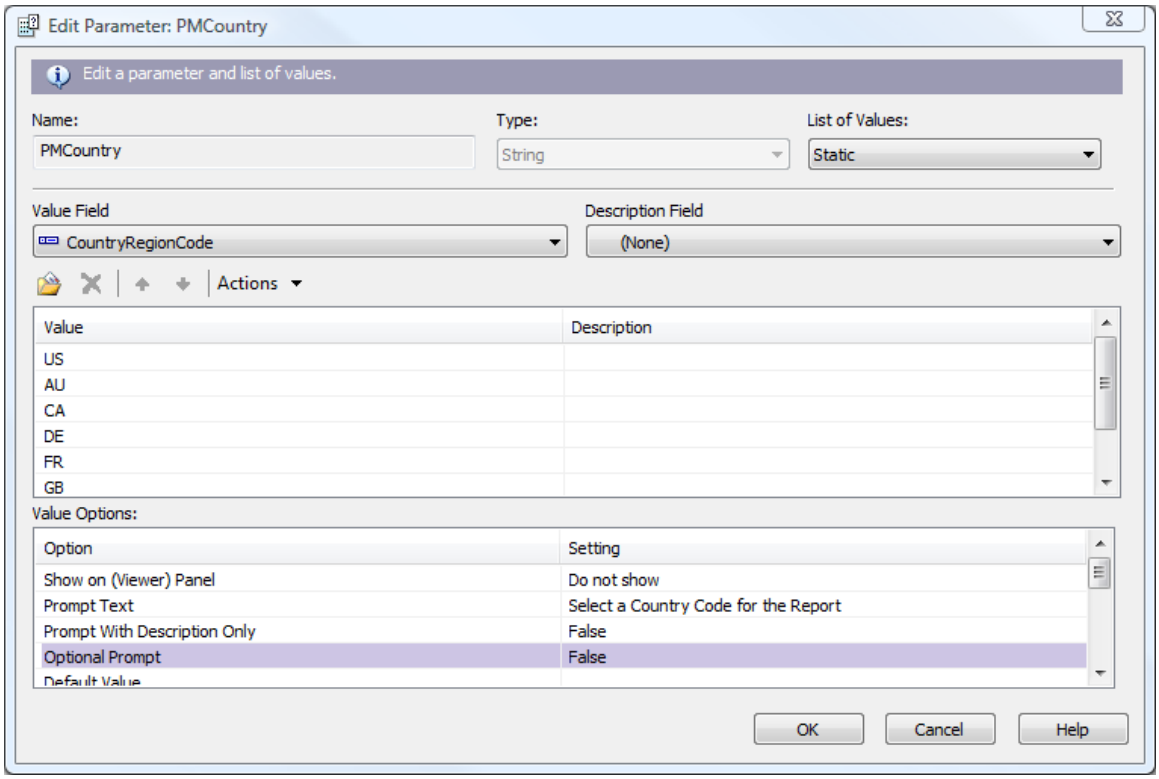
5. The **Enter Parameters Value** dialog box opens prompting you to respond to the parameter. Since we have already supplied a default value, select **US**, and click **OK** and **OK** again to close the Database Expert dialog box. Preview and save the report (**SQL Command.rpt**).

Now let's modify the parameter field for the query to be a little more user friendly. You will notice you have a parameter field in your Field Explorer called PMCountry. You can edit this in the same way as we discussed in the parameter fields lesson. We are going to add more values to our default value list. If you select the SimpleTerritoryListing table and the field CountryRegion, the only value you can add to the default value list is that value you have already selected. Instead of typing all the values let's add the SalesTerritory table to our report letting Crystal link it however it likes as we are going to remove it later. We can then edit the parameter field and select the CountryRegion field values from the SalesTerritory table as shown below.

6. Open the **Database Expert** and add the **SalesTerritory** table to the report. Accept any linking shown in the **Linking** tab. Click **OK** once the table is added.
   Read, but ignore the warning and cancel the Refresh Data request since it is of no value to you.

7. From the **Field Explorer**, select the **PMCountry** parameter and open it for editing.

8. Under *Value Field* select the **SalesTerritory** table and **CountryRegionCode** field. Click the **Actions** button and choose **Append all database values**. Choose **OK**.



9. Remove the **SalesTerritory** table from the report.

10. Refresh the report and select another country, such as **CA** (Canada) to test the changes. *Notice only one Country can be chosen at a time.*

11. Save the report as **SQL Command with Parameter.rpt**.

NOTE: Make sure you remove the SalesTerritory table from the report in the Database Expert.

NOTES

NOTES