

# Advanced Crystal Reports



with examples and  
hands-on exercises

---

**WEBUCATOR**

Copyright © 2021 by Webucator. All rights reserved.

No part of this manual may be reproduced or used in any manner without written permission of the copyright owner.

**Version:** 1.1.1

## **The Authors**

### ***Tracy Berry***

Tracy has been a senior graphic designer/programmer, instructor, and consultant since 1993 and has developed hundreds of logos, marketing materials, websites, and multimedia solutions for customers worldwide, including involvement in large corporate software rollouts. She has helped many organizations optimize and streamline data solutions. She teaches both onsite and online courses and has her CTT (Certified Technical Trainer) certification. Tracy specializes in teaching graphics, desktop publishing, web design, reporting/productivity applications, as well as the creation of online courses with software from leading vendors.

### ***Christopher Reed***

Christopher Reed has been programming since he was thirteen and mentoring others almost as long. His expertise includes .NET technology, SQL Server, Crystal Reports, numerous databases, and a variety of programming languages. He has practical experience in academic, Fortune 150, federal government contracting, local government, call center, and software manufacturing settings. Christopher holds current MCT, MCPD, and MCTS certifications from Microsoft. He has been delivering onsite and online training with Webucator since 2010.

## **Class Files**

Download the class files used in this manual at

<https://static.webucator.com/media/public/materials/classfiles/CRY2011.3-1.1.1.zip>.

## **Errata**

Corrections to errors in the manual can be found at

<https://www.webucator.com/books/errata/>.

# Table of Contents

LESSON 1. Crystal Reports Review.....	1
📄 <b>Exercise 1: Review of Basic Report Building.....</b>	<b>2</b>
LESSON 2. Working with SQL.....	21
The SELECT Statement.....	22
The WHERE Clause.....	23
The ORDER BY Clause.....	23
Joins.....	23
SQL Commands.....	24
SQL Expressions.....	28
📄 <b>Exercise 2: Using SQL Expressions in Reports.....</b>	<b>31</b>
LESSON 3. Advanced Functions.....	33
Evaluation Time Functions.....	33
Working with Variables.....	35
For Loops.....	36
While Do Loops.....	36
📄 <b>Exercise 3: Working with Variables and Loops.....</b>	<b>40</b>
LESSON 4. Geographic Mapping.....	45
Geographic Maps.....	45
📄 <b>Exercise 4: Adding a Geographic Map.....</b>	<b>49</b>
LESSON 5. Establishing a Database Connection.....	53
Connecting to the AdventureWorks Database.....	53



# LESSON 1

## Crystal Reports Review

---

### Topics Covered

- Review exercise.

### Introduction

Before we introduce new topics, let's review the topics covered in the introductory/intermediate course by building a practice report that reinforces the following:

- Creating a connection.
- Organizing fields into the report sections.
- Controlling sections using the section expert.
- Creating groups and group summaries.
- Using formulas to create content.
- Creating a chart.
- Using parameters to filter a report.

# Exercise 1: Review of Basic Report Building

 45 to 60 minutes

In this exercise, you will create a new report about the sales people of Adventure Works Cycles.

1. Show the following information for each sales person:
  - A. Full name (using a formula)
  - B. Phone number
  - C. Date of birth
  - D. Last year's sales
2. Total all sales figures.
3. Group sales people by sales territory and region.

**Evaluation Copy**

## Note

There will be two groups based on a conditional formula that represents the regions and then the sales territories. For the US territories, show the specific territory name followed by "United States".

4. Name the report `SalesReport.rpt` and save it to your `ClassFiles\cry2011-advanced-review\Exercises` folder.

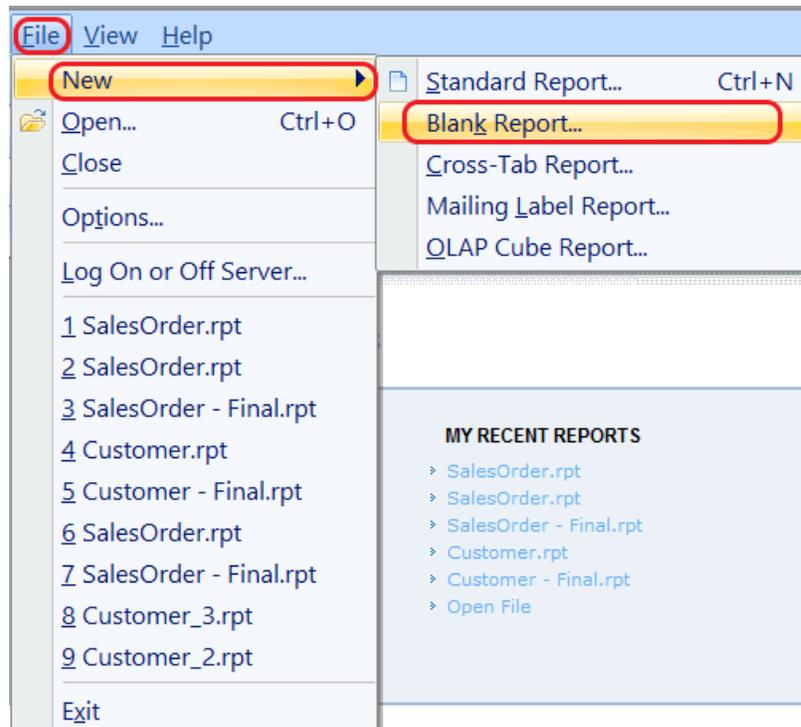


## Solution

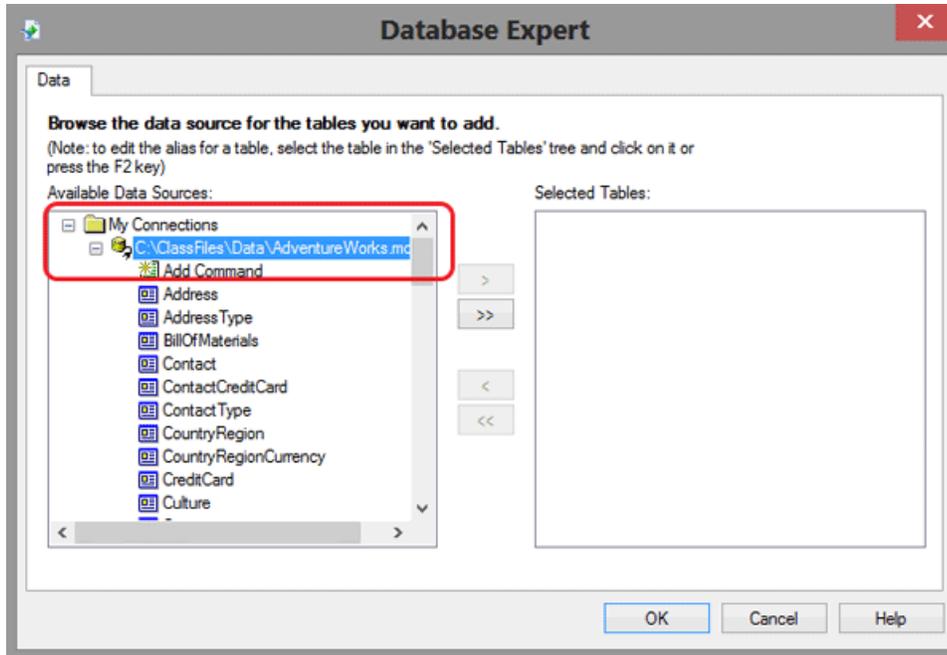
---

The following steps represent one possible solution to the exercise:

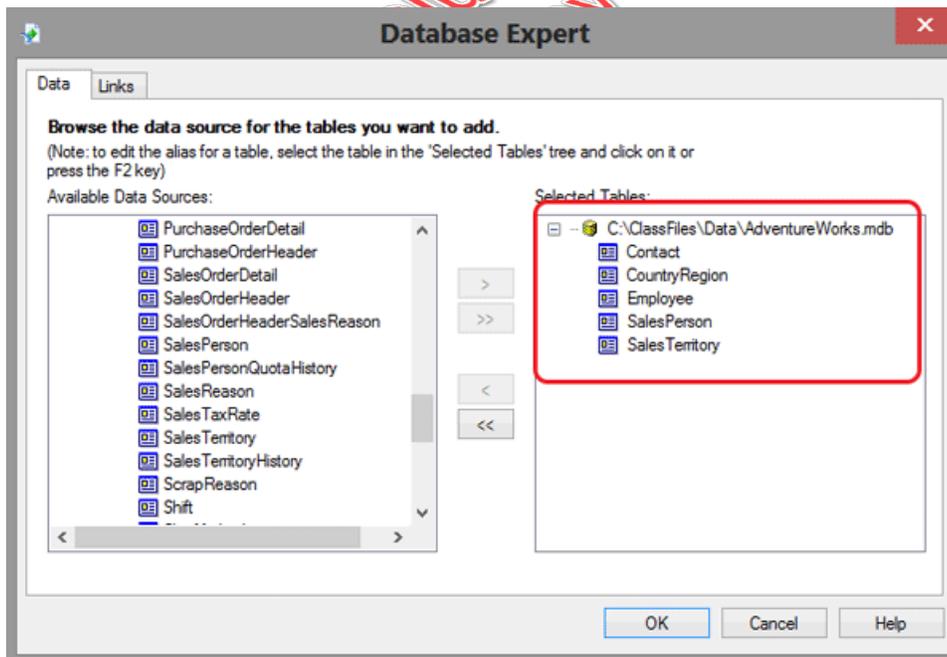
1. From the menu, select **File > New > Blank Report**.



2. In the **Database Expert** dialog box, expand to the **ClassFiles\Data\AdventureWorks.mdb** node. (If you have not already established this connection, see the appendix.)

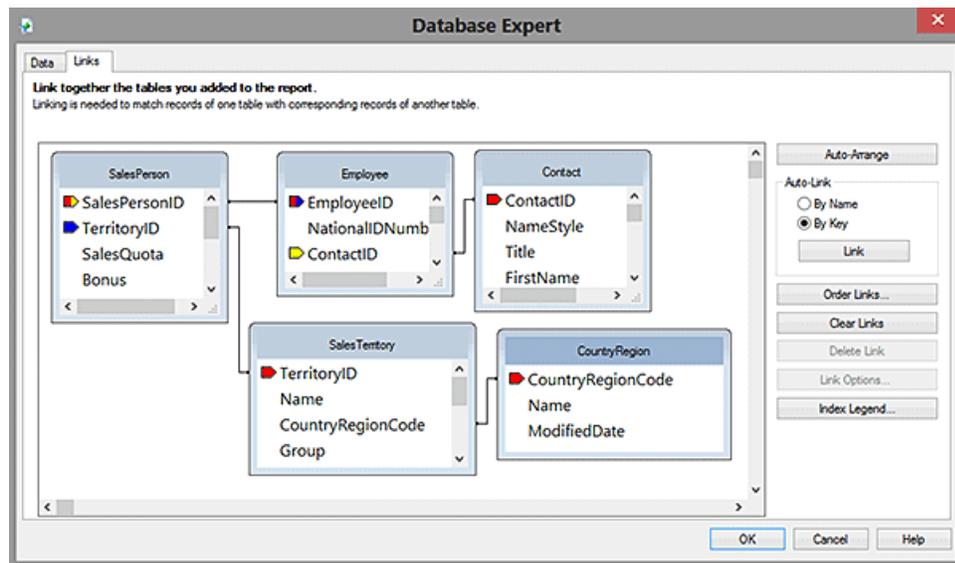


3. Double-click the **Employee**, **Contact**, **CountryRegion**, **SalesPerson**, and **SalesTerritory** nodes.

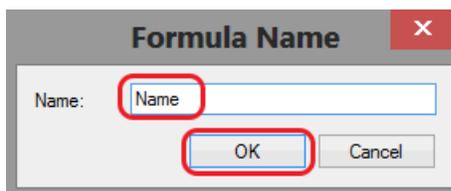


4. Select the **Links** tab.
5. Click the **Clear Links** button. Click **Yes** to confirm that you want to clear the links.

6. Create the following links between the tables:
  - A. Link the **ContactID** field of the **Contact** table to the **ContactID** field of the **Employee** table.
  - B. Link the **EmployeeID** field of the **Employee** table with the **SalesPersonID** field of the **SalesPerson** table.
  - C. Link the **TerritoryID** field of the **SalesPerson** table with the **TerritoryID** field of the **SalesTerritory** table.
  - D. Link the **CountryRegionCode** field of the **SalesTerritory** table with the **CountryRegionCode** field of the **CountryRegion** table.



7. Click **OK**.
8. From the menu, select **View > Field Explorer** to show the **Field Explorer** pane.
9. In the **Field Explorer**, select the **Formula Fields** node and click the **New** icon in the **Field Explorer** toolbar.
10. In the **Formula Name** dialog box, type "Name" and click **OK**.



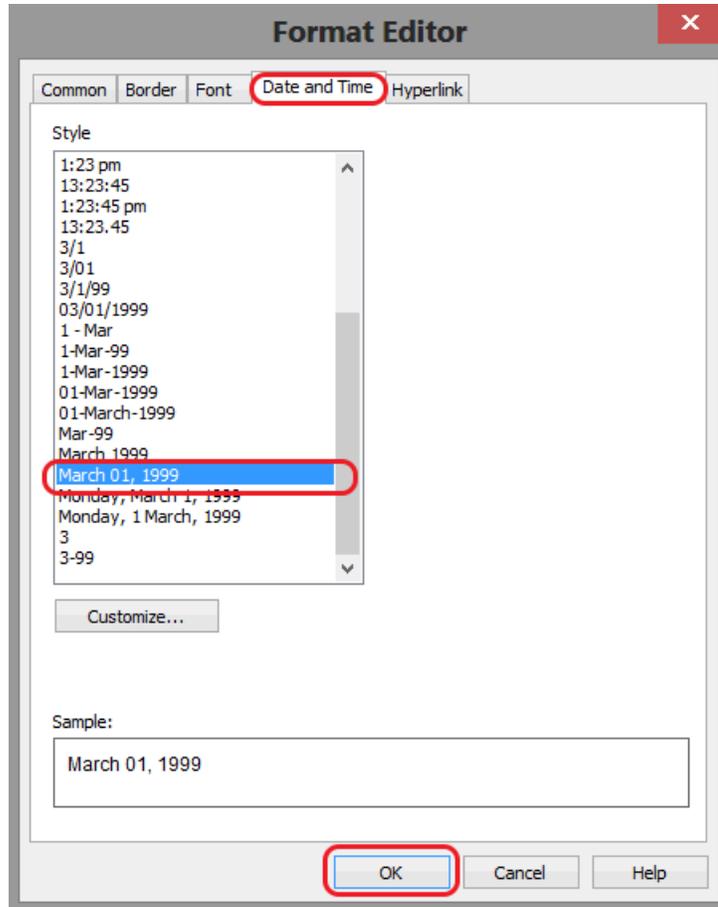
11. In the **Formula Workshop - Formula Editor - Name** dialog box, type the following in the formula editor:

```
{Contact.FirstName} & (If IsNull({Contact.MiddleName}) Then " " Else " " & {Contact.MiddleName}) & " " & {Contact.LastName}
```

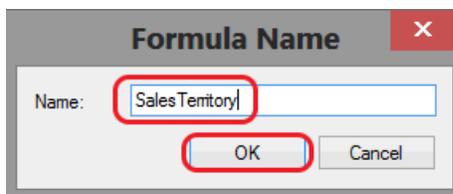
12. Click the **Check** icon to verify the formula.
13. Click **Save and close**.
14. Click and drag the following fields, in the order indicated, from the **Field Explorer** to the **Details** section of the report. Use the text in parentheses as the field title:
  - A. **Name** formula field ("Sales Person")
  - B. **Contact.Phone** ("Phone")
  - C. **Employee.BirthDate** ("Date of Birth")
  - D. **SalesPerson.SalesLastYear** ("Sales Last Year")

	1	2	3	4	5	6	7
RH							
PH							
D	Sales Person	Phone	Date of Birth	Sales Last Year			
RF	@Name	Phone	BirthDate	SalesLastYear			
PF							

15. Right-click the **BirthDate** field and select **Format Field**.
16. In the **Format Editor** dialog box, on the **Date and Time** tab, click **March 01, 1999**.



17. Click **OK** to close the dialog box.
18. Adjust the widths of the fields as needed.
19. In the **Field Explorer**, select the **Formula Fields** node and click the **New** icon in the **Field Explorer** toolbar.
20. In the **Formula Name** dialog box, type “SalesTerritory” and click **OK**.



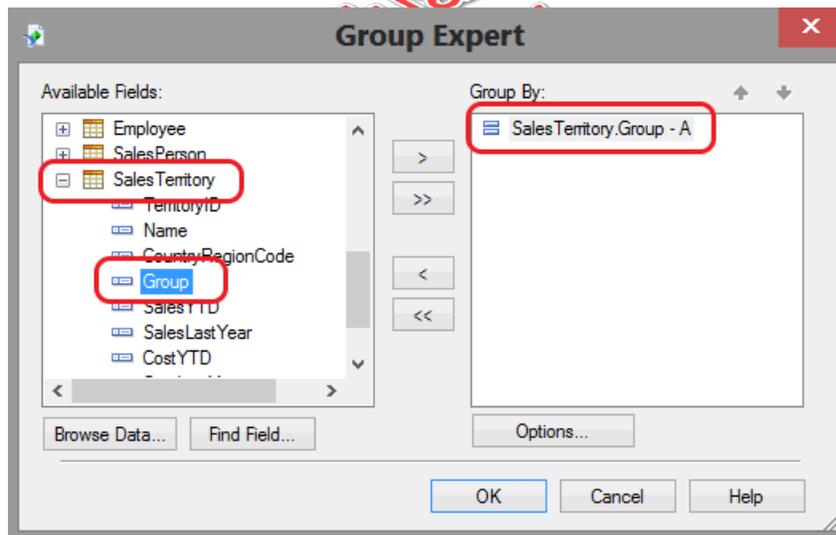
21. In the **Formula Workshop - Formula Editor - Sales Territory** dialog box, type the following in the formula editor:

```
{SalesTerritory.Name} & (If {SalesTerritory.CountryRegionCode} = "US" Then " United States" Else "")
```

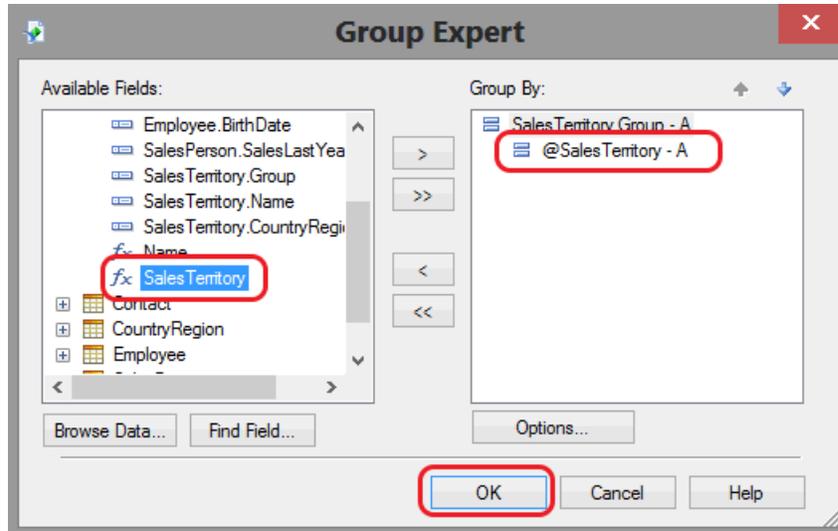
22. Verify the formula and click **Save and close**.
23. In the **Experts** toolbar, click the **Group Expert** icon.



24. In the **Group Expert** dialog box, locate and expand the **SalesTerritory** node and then double-click the **Group** field.

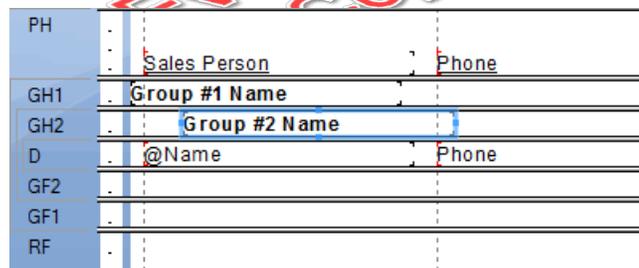


25. Locate and double-click the **Sales Territory** formula field.

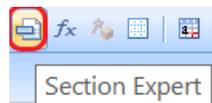


26. Click **OK**.

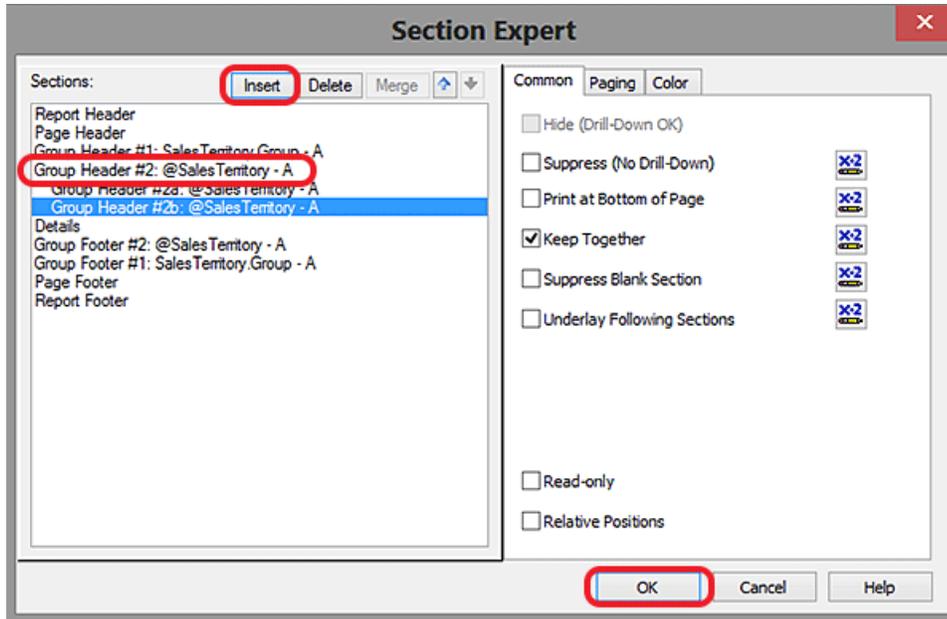
27. Move the **Group #2 Name** over.



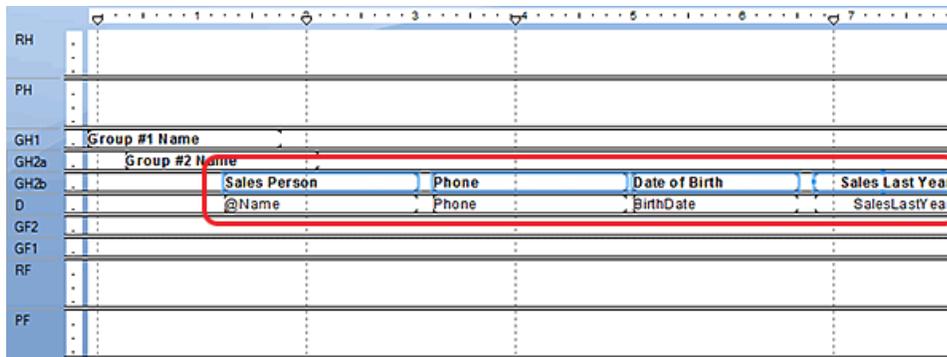
28. Click the **Section Expert** icon.



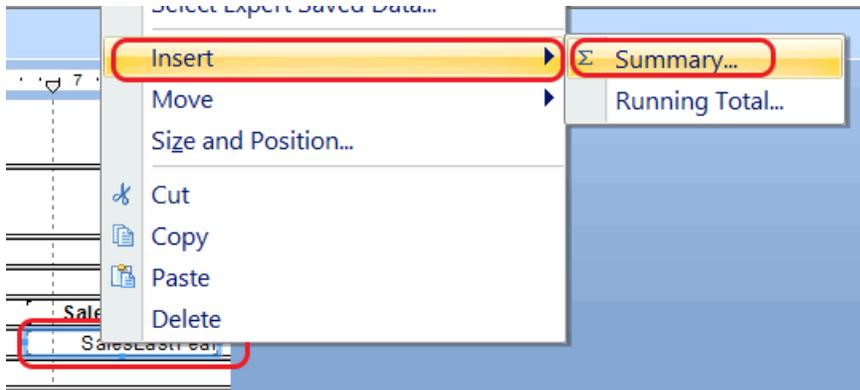
29. In the **Section Expert** dialog box, in the **Sections** list box, select **Group Header #2** and click **Insert**. Click **OK**.



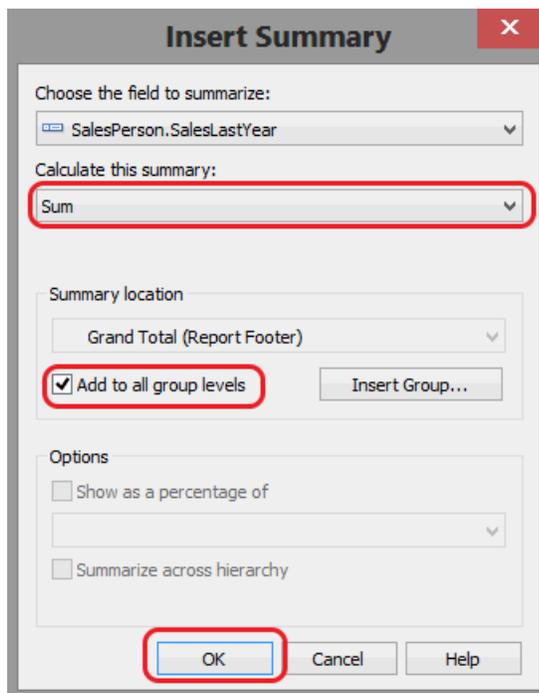
30. Select all field titles, move them to the **Group Header #2b**, and click the **Bold** icon in the **Formatting** toolbar.
31. Adjust the width of any fields along with their corresponding field titles as needed.
32. Move all of the fields to the right, adjusting them accordingly.



33. Right-click the **Sales Last Year** field and select **Insert > Summary**.



34. In the **Insert Summary** dialog box, verify that “Sum” is selected in the **Calculate this summary** drop-down list, and then in the **Summary location** section mark the **Add to all group levels** check box.



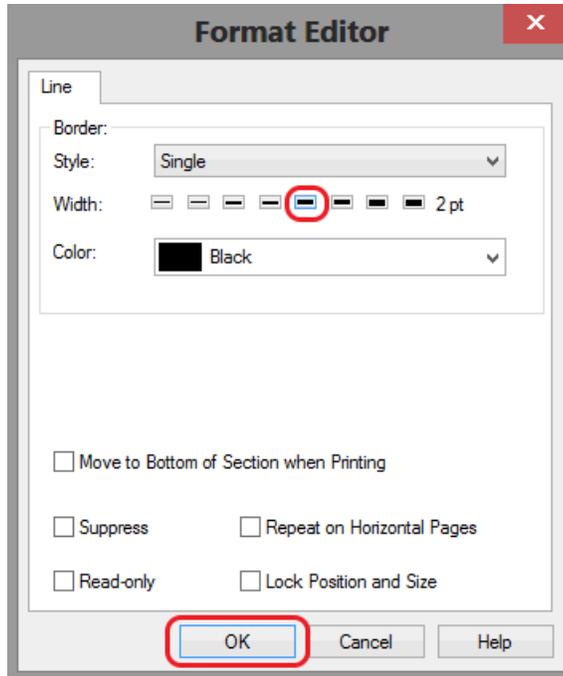
35. Click **OK**.

RH				
PH				
GH1	Group #1 Name			
GH2a	Group #2 Name			
GH2b	Sales Person	Phone	Date of Birth	Sales Last Year
D	@Name	Phone	BirthDate	SalesLastYear
GF2				son.SalesLastYear
GF1				son.SalesLastYear
RF				son.SalesLastYear
PF				

36. Increase the height of **Group Footer #1**, **Group Footer #2**, and the **Report Footer**. Move each summary down to the bottom of the section.

	Date of Birth		Sales Last Year
	BirthDate		SalesLastYear
	of SalesPerson.SalesLastYear		
	of SalesPerson.SalesLastYear		
	of SalesPerson.SalesLastYear		

37. Click the **Insert Line** icon. In the **Group Footer #1** and the **Group Footer #2** section, draw a line just above each summary field.
38. Format each line to suit you by right-clicking the line object and select **Format Line**.
39. In the **Format Editor** dialog box, change the width to 2 pt and click **OK**.



40. Draw a box around the **Report Footer** Summary.
41. Create labels for each summary by doing the following:
  - A. Click the **Insert Text Object** icon. Move the cursor to the **Group Footer #2** section near the middle of the report and click.
  - B. In the text object, type "Sales for".
  - C. In the **Field Explorer**, expand the **Group Name Fields** node, drag the **Group #2 Name** field into the text object, and then type a colon (":") at the end.
  - D. Right-click the summary field in the **Group Footer #2** section and select **Cut**.
  - E. Right-click the text object and select **Edit Text**. Go to the end of the text, add a space, and then right-click and select **Paste**. Click outside of the text object.
  - F. Expand the text object to the far-right side of the **Group Footer #2** section.
  - G. With the text object still selected, click the **Bold** and **Align Right** icons.
  - H. Repeat for each remaining summary.

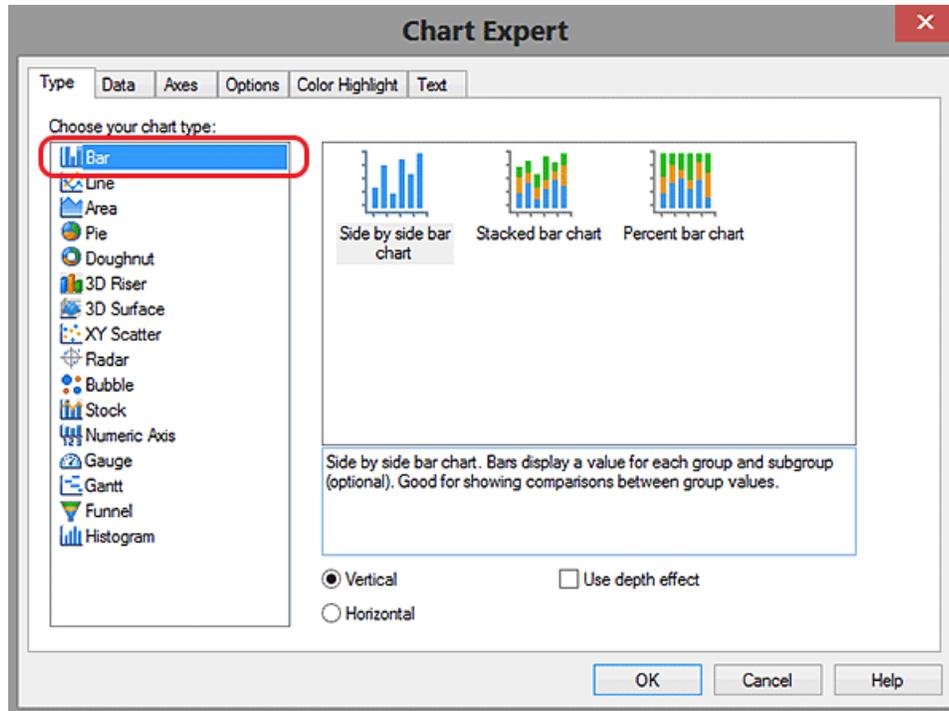
RH				
PH				
GH1	Group #1 Name			
GH2a	Group #2 Name			
GH2b	Sales Person	Phone	Date of Birth	Sales Last Year
D	@Name	Phone	BirthDate	SalesLastYear
GF2	Sales for (Group #2 Name): (Sum of SalesPerson.SalesLastYear)			
GF1	Sales for (Group #1 Name): (Sum of SalesPerson.SalesLastYear)			
RF	Total Sales: (Sum of SalesPers...			
PF				

42. Click the **Insert Text Object** icon. Move the cursor to the top left corner of the **Report Header** section and click.
43. Type "Adventure Works Cycles Sales People Report" inside the text object. Click outside of the text object.
44. Move the text object into the upper-left corner of the **Report Header** section and expand the text object to the other end of the **Report Header** section.
45. Click the **Bold** and **Align Center** icons.
46. From the **Font Size** drop-down list, select "20".
47. Expand the text object vertically to ensure that the text is completely visible. Also adjust the height of the **Report Header** section to ensure that the text object is not too close to the field titles.

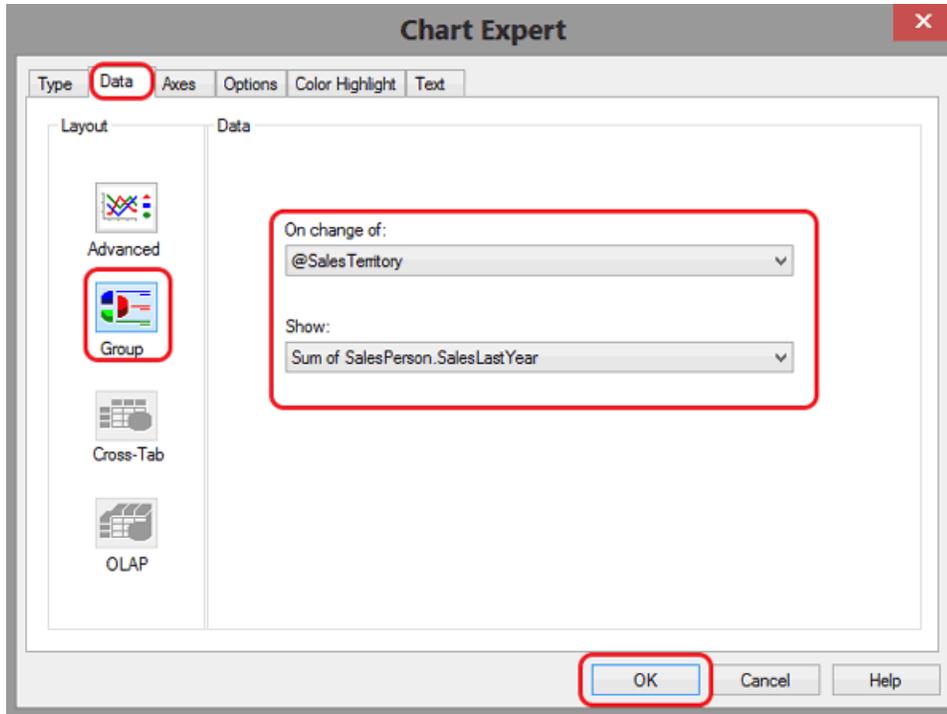
RH	<b>Adventure Works Cycles Sales People Report</b>			
PH				
GH1	Group #1 Name			
GH2a	Group #2 Name			
GH2b	Sales Person	Phone	Date of Birth	Sales Last Year
D	@Name	Phone	BirthDate	SalesLastYear
GF2	Sales for (Group #2 Name): (Sum of SalesPerson.SalesLastYear)			
GF1	Sales for (Group #1 Name): (Sum of SalesPerson.SalesLastYear)			
RF	Total Sales: (Sum of SalesPers...			
PF				

48. Save the report as "SalesReport.rpt".
49. Preview the report.
50. Click the **Insert Chart** icon. Move the mouse cursor to the far-left side of the **Group Header 1** and click.

51. A chart is placed in your **Group Header 1b**, you may need to right-click and choose **Chart Expert** if it does not open on its own.
52. In the **Chart Expert** dialog box, select the **Type** tab. In the **Choose your chart type** list box, select **Bar**.



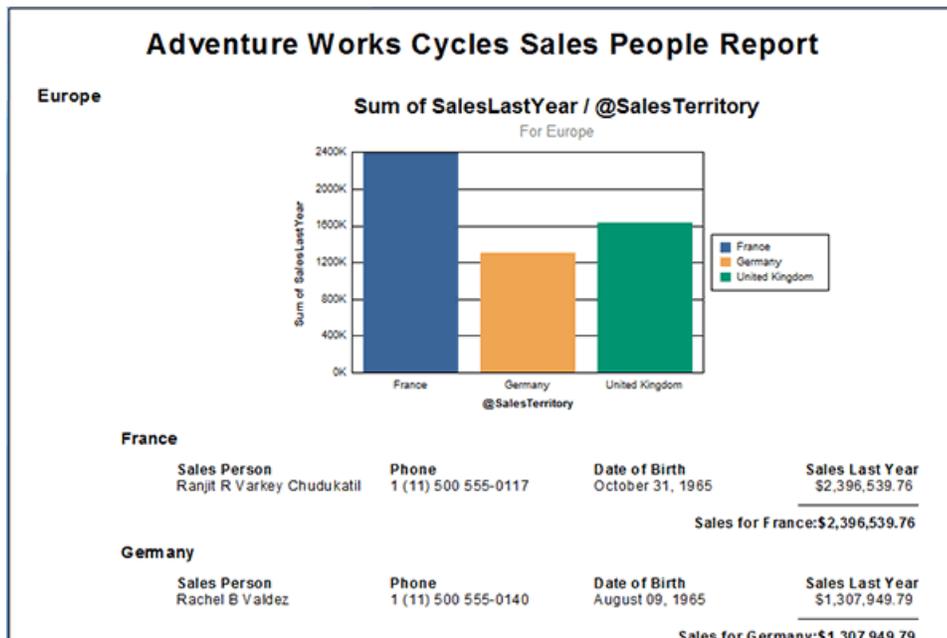
53. Select the **Data** tab.
54. Confirm that the **CountryRegion.Name** field is listed in the **On change of** drop-down and that the **Show** drop-down has **Count of Customer.AccountNumber** is listed.



55. Click **OK**.

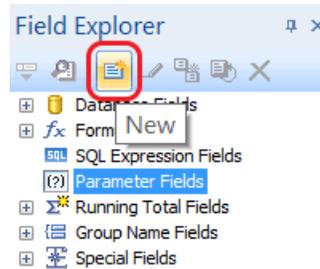
56. Resize the chart object to fit in the subreport.

57. Preview and save the report.



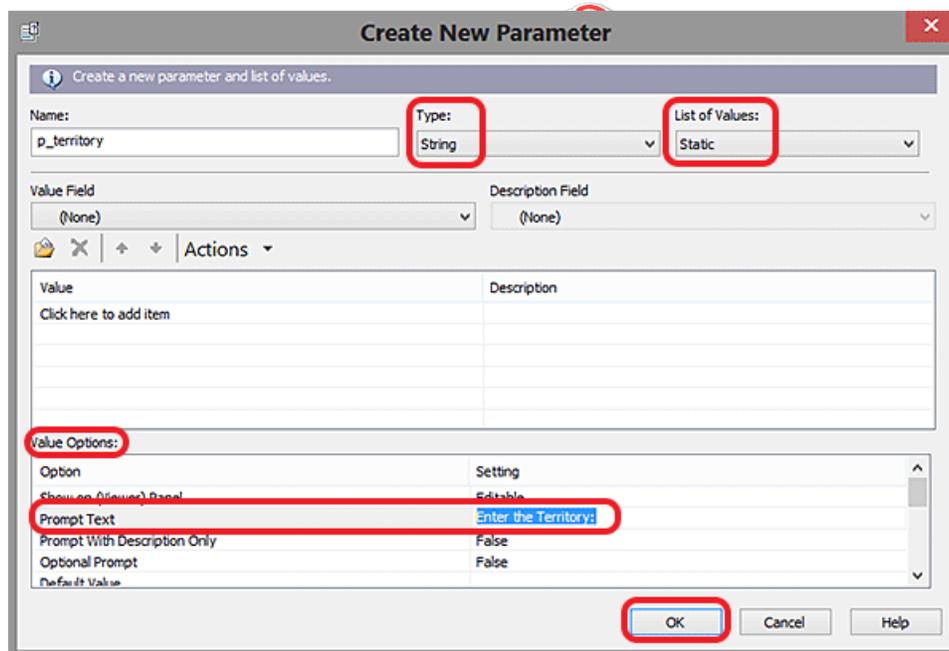
58. Select the **Design** tab.

59. In the **Field Explorer**, select the **Parameter Fields** node and click the **New** icon in the **Field Explorer** toolbar.



60. In the **Create New Parameter** dialog box, replace the text in the **Name** field with "p\_territory".

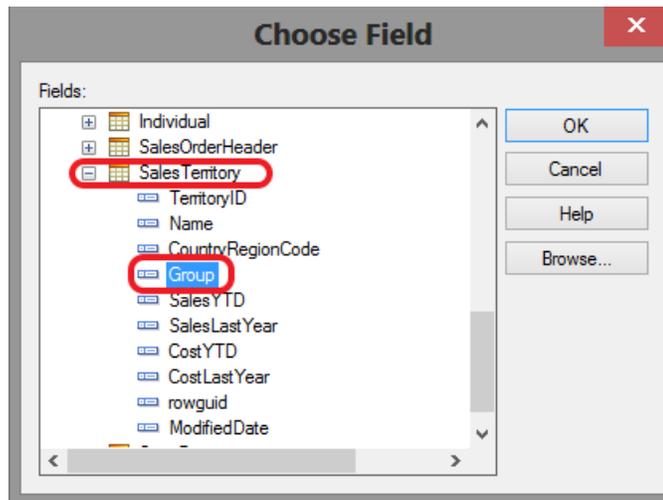
61. Verify that **String** is selected in the **Type** drop-down list and that **Static** is selected in the **List of Values** drop-down list. In the Value Options area, locate the **Prompt Text** and replace it with **"Enter the Territory"**, then click **OK**.



62. Click the **Select Expert** icon.

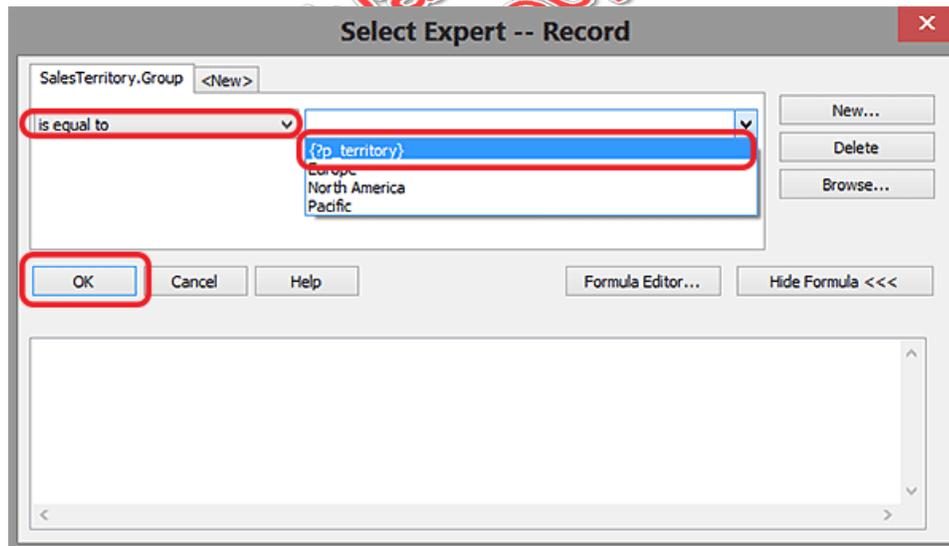


63. In the **Choose Field** dialog box, double-click **SalesTerritory.Group**. Click **OK**.



64. In the **Select Expert -- Record** dialog box, select "is equal to" from the drop-down list of conditions.

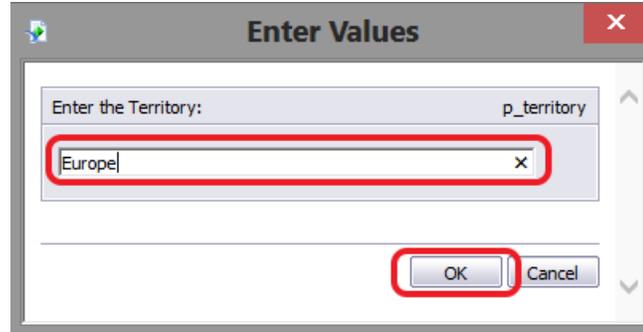
65. In the second drop-down list, select the "{?p\_territory}" option which represents the SalesTerritory.Group parameter field.



66. Click **OK**.

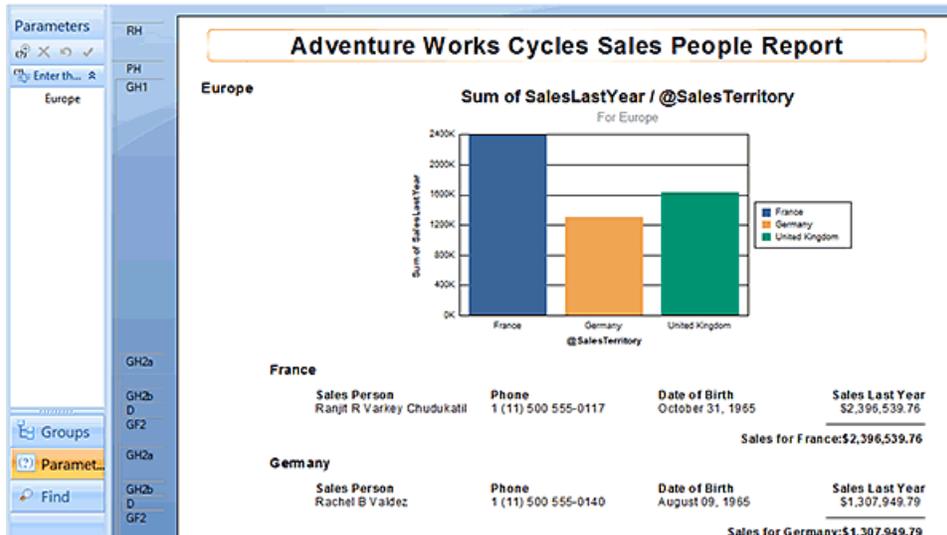
67. Preview the report. The **Enter Values** dialog box prompts you for a territory name.

68. In the **Enter p\_territory** field, type "Europe" and then click **OK**.



69. Review the report. Notice that only customers in Europe appear on the report.

After some formatting adjustments, the report should look similar to the following:



## Conclusion

In this lesson, you have learned

- To complete a introductory/intermediate Crystal Reports review exercise.

# LESSON 2

## Working with SQL

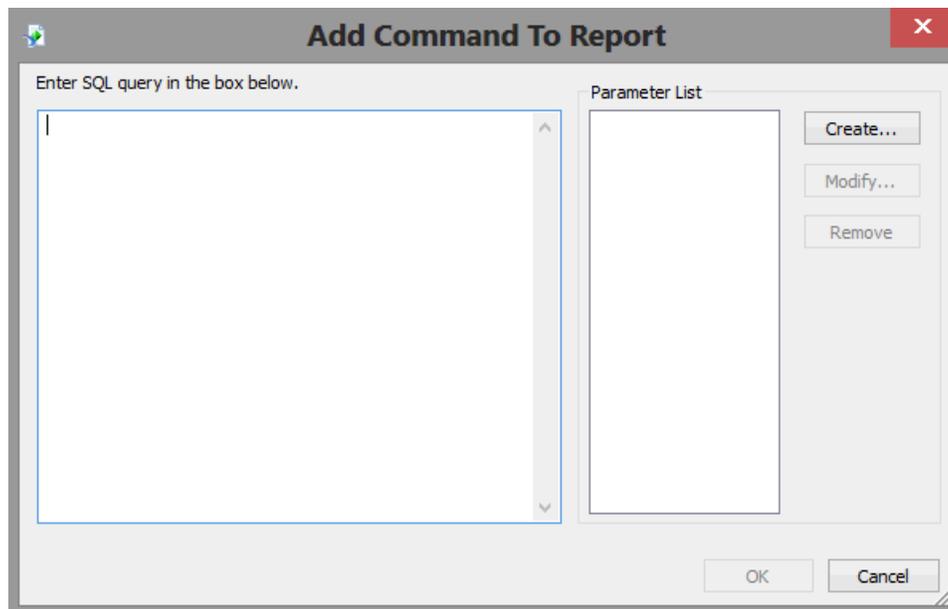
---

### Topics Covered

- The SQL SELECT statement.
- Joining tables.
- Adding a SQL command to a report.
- Using SQL Expressions in a report.

### Introduction

Throughout this course we have been using a relational database as our data source. Though there are many other data repositories available, relational databases are the most often used. The language we use to extract and update information in a relational database is referred to as Structured Query Language, or SQL. With SQL, we can use the Database Expert (Add Command to Report) to build a database command that we call a *query*.



In this lesson, we will explore SQL and see how it works within a database system. Additionally, we will see how we can use SQL directly in Crystal Reports.



## 2.1. The SELECT Statement

Data is only helpful if we are able to see it and use it. To acquire data from a relational database, we use the SELECT statement from SQL. The SELECT statement is the basic command for acquiring information from a relational database.

Since we are using an Access database for our reports, we'll use the Database Expert (Add Command to Report) to build a database command that will contain the SQL.

### ❖ 2.1.1. Working with SELECT Statements

Selects all of the fields from a Contact table.

```
SELECT * FROM Contact
```

**Evaluation  
Copy**

#### Note

The database may be case sensitive, so you may need to use the exact name for the schema, tables, and fields.

Selects only certain fields from the Contact table.

```
SELECT ContactID, Title, FirstName, MiddleName, LastName, EmailAddress, Phone  
FROM Contact
```



## 2.2. The WHERE Clause

During this exploration of SQL, we will look at some of the features that we are familiar with in Crystal Reports and see how they work in SQL. The first feature we will look at is filtering. To create a filter within a SQL statement, you need to use a WHERE clause.

### ❖ 2.2.1. Filtering with a WHERE Clause

Selects certain fields from Contact table who have the last name of Brown.

```
SELECT ContactID, Title, FirstName, MiddleName, LastName, EmailAddress, Phone
FROM Contact
WHERE LastName = 'Brown'
```



## 2.3. The ORDER BY Clause

To sort report data using SQL, you need the ORDER BY clause.

### ❖ 2.3.1. Using ORDER BY Clauses

Selects certain fields from Contact table ordered by last name, first name, and contact ID in descending order.

```
SELECT ContactID, Title, FirstName, MiddleName, LastName, EmailAddress, Phone
FROM Contact
ORDER BY LastName, FirstName, ContactID
DESC
```



## 2.4. Joins

As we have seen in Crystal Reports, most reports require the use of more than one table from the data source. While not all data sources are relational databases, the main reason we use relational databases as a primary repository for data is that we can effectively limit the amount of data we store by eliminating some of the redundancy. To accomplish this, we use multiple tables in the database. Since these tables belong to a *relational* database,

we need to be able to link them together. The link we create is called a *join* and we create a join by linking two tables with a common field.

In Crystal Reports, we have been handling joins by linking tables together on the **Links** tab of the **Database Expert**. Behind the scenes, Crystal Reports turns these links into SQL JOINS.

### ❖ 2.4.1. Joining Tables Together

We are seeking individual customer information that contains the customer's name and account number along with sales order information including the order number, the sales subtotal, the tax amount for the sale, and the freight charges.

```
SELECT AccountNumber, FirstName, MiddleName, LastName, SalesOrderNumber, SubTotal, TaxAmt, Freight
FROM Customer INNER JOIN Individual ON CustomerID = CustomerID INNER JOIN Contact ON ContactID = ContactID INNER JOIN SalesOrderHeader ON ContactID = ContactID
```

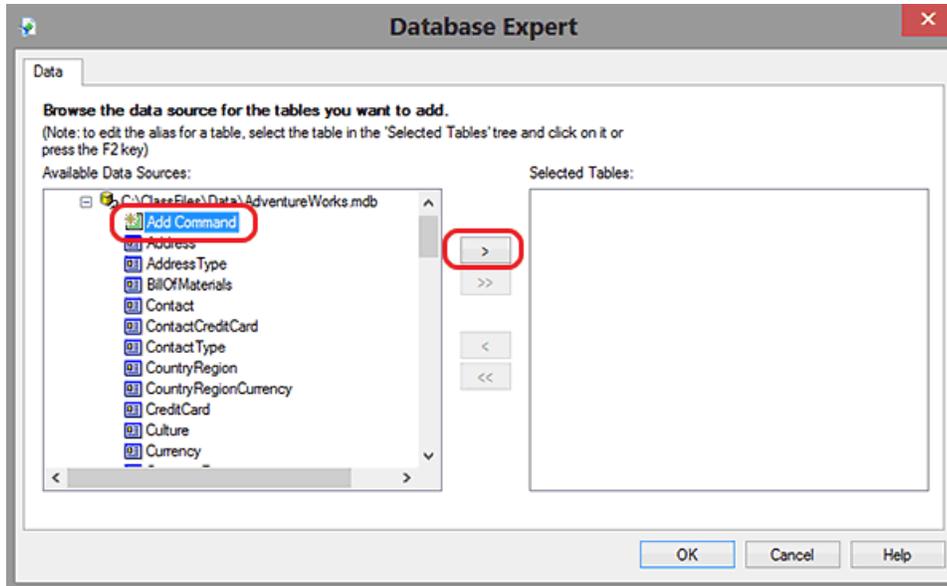
## 2.5. SQL Commands

While there is a great deal of flexibility built into Crystal Reports with respect to querying data, there are some cases when building and using your own query may be better than using the SELECT statements built by Crystal Reports. First and foremost, a report may be faster with a well-crafted SELECT statement. Additionally, there are many specialized queries that cannot be built using Crystal Reports either because of the complexity of the query itself or the database-specific functionality that is needed for the query. Thus, Crystal Reports provides us with the means to create a report based on a SELECT statement rather than adding tables with relationships through the **Database Expert**.

### ❖ 2.5.1. Using a SQL Command

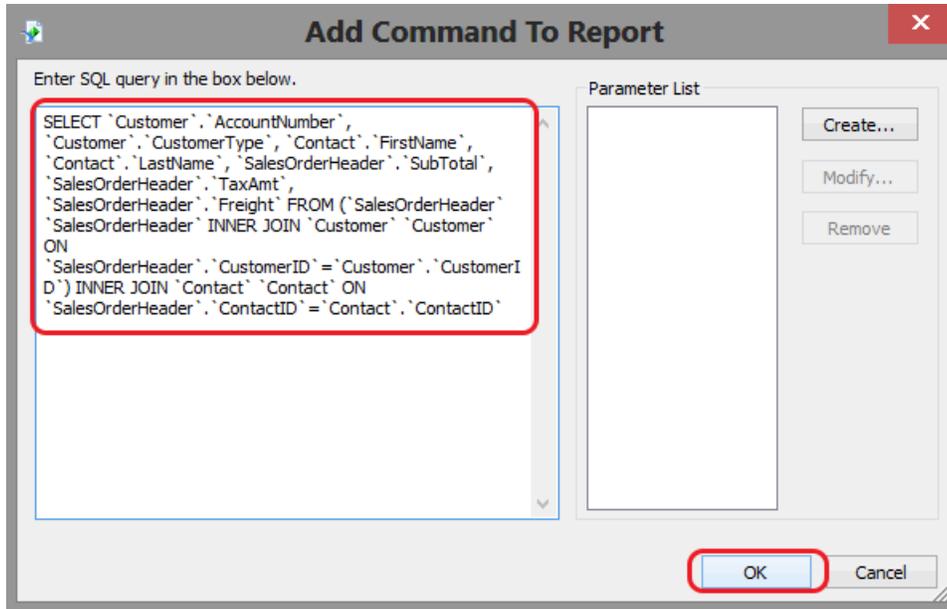
We will create a new report using the SELECT statement that we already discussed.

1. From the menu, select **File > New > Blank Report**.
2. In the **Database Expert**, in the **Available Data Sources** tree, expand the **C:\ClassFiles\Data\AdventuerWorks.mdb** node and double-click **Add Command**.

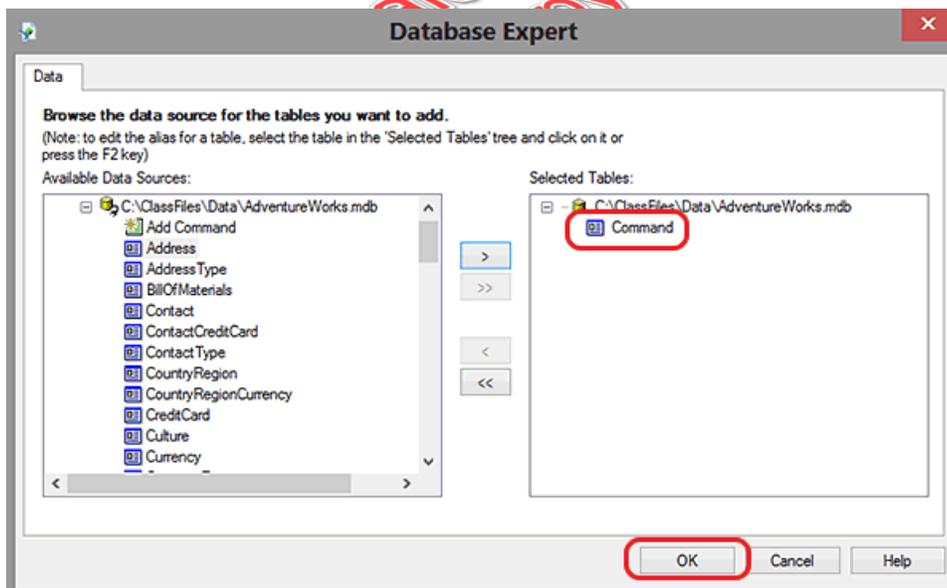


3. In the **Add Command To Report** dialog box, add the following query in the **Enter SQL query in the box below** list box:

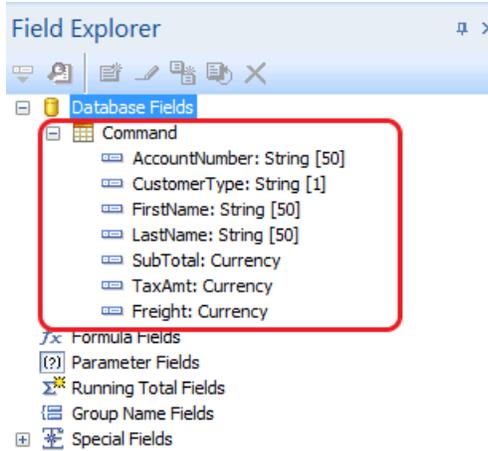
```
SELECT `Customer`.`AccountNumber`, `Customer`.`CustomerType`, `Contact`.`First Name`, `Contact`.`LastName`, `SalesOrderHeader`.`SubTotal`, `SalesOrderHeader`.`TaxAmt`, `SalesOrderHeader`.`Freight`
FROM(`SalesOrderHeader` `SalesOrderHeader` INNER JOIN `Customer` `Customer` ON `SalesOrderHeader`.`CustomerID`=`Customer`.`CustomerID`) INNER JOIN `Contact` `Contact` ON `SalesOrderHeader`.`ContactID`=`Contact`.`ContactID`
```



4. Click **OK**. Note that the **Selected Tables** list box contains an entry called "Command".



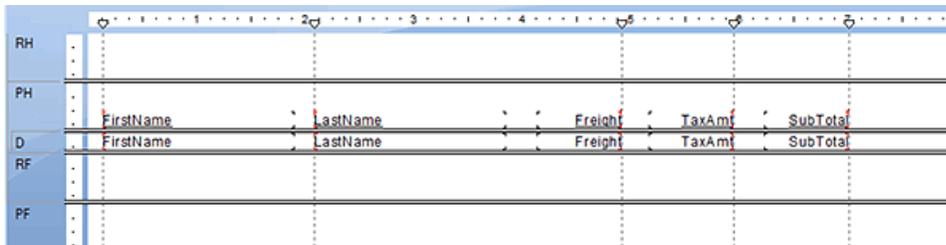
5. Click **OK**.
6. In the **Field Explorer**, expand the **Database Fields > Command** node.



7. Drag the following fields, in order, to the **Details** section of the report:

- **FirstName**
- **LastName**
- **Freight**
- **TaxAmt**
- **SubTotal**

Evaluation Copy



8. Preview the report. Save the report with the name "Customer2.rpt".

<u>FirstName</u>	<u>LastName</u>	<u>Freight</u>	<u>TaxAmt</u>	<u>SubTotal</u>
Gustavo	Achong	\$79.56	\$254.59	\$3,182.39
Gustavo	Achong	\$122.40	\$391.68	\$4,895.99
Gustavo	Achong	\$725.76	\$2,322.44	\$29,030.54
Gustavo	Achong	\$821.96	\$2,630.28	\$32,878.50
Gustavo	Achong	\$1,127.75	\$3,608.81	\$45,110.13
Gustavo	Achong	\$977.71	\$3,128.68	\$39,108.55
Gustavo	Achong	\$121.50	\$388.80	\$4,859.99
Catherine	Abel	\$658.18	\$2,106.17	\$26,327.11
Catherine	Abel	\$953.01	\$3,049.64	\$38,120.52
Catherine	Abel	\$994.63	\$3,182.83	\$39,785.33
Catherine	Abel	\$844.44	\$2,702.22	\$33,777.76
Kim	Abercrombie	\$1,648.73	\$5,275.94	\$65,949.25
Kim	Abercrombie	\$839.35	\$2,685.93	\$33,574.15
Kim	Abercrombie	\$1,416.85	\$4,533.92	\$56,674.04
Kim	Abercrombie	\$1,627.61	\$5,208.35	\$65,104.34
Kim	Abercrombie	\$1,856.97	\$5,942.30	\$74,278.77
Kim	Abercrombie	\$1,054.90	\$3,375.68	\$42,195.99
Kim	Abercrombie	\$500.21	\$1,600.69	\$20,008.58
Kim	Abercrombie	\$2,022.84	\$6,473.10	\$80,913.69
Kim	Abercrombie	\$698.91	\$2,236.52	\$27,956.44
Kim	Abercrombie	\$904.74	\$2,895.16	\$36,189.47
Kim	Abercrombie	\$1,804.15	\$5,773.28	\$72,166.02
Kim	Abercrombie	\$1,228.96	\$3,932.68	\$49,158.55
Humberto	Acevedo	\$528.74	\$1,691.96	\$21,149.47
Humberto	Acevedo	\$74.57	\$238.63	\$2,982.84



## 2.6. SQL Expressions

When we use formulas in Crystal Reports, they are evaluated by the report. In some cases, it is more efficient to have the formula evaluated by the database server. To accomplish this, the formulas need to be handled by the database system using its particular version of SQL. Fortunately, we have the means to do this in Crystal Reports using SQL Expressions.

SQL Expressions can be used instead of a formula field to achieve the same result; however, a SQL Expression is evaluated by the database server and the result is passed back to the report on the client. The expression used, of course, must be compatible with the database system on which the data source is based; therefore, the SQL Expression you write for a SQL Server database may differ from the equivalent expression for an Oracle database. Still, in those cases where a complex formula is necessary, using a SQL Expression may be the optimal solution.

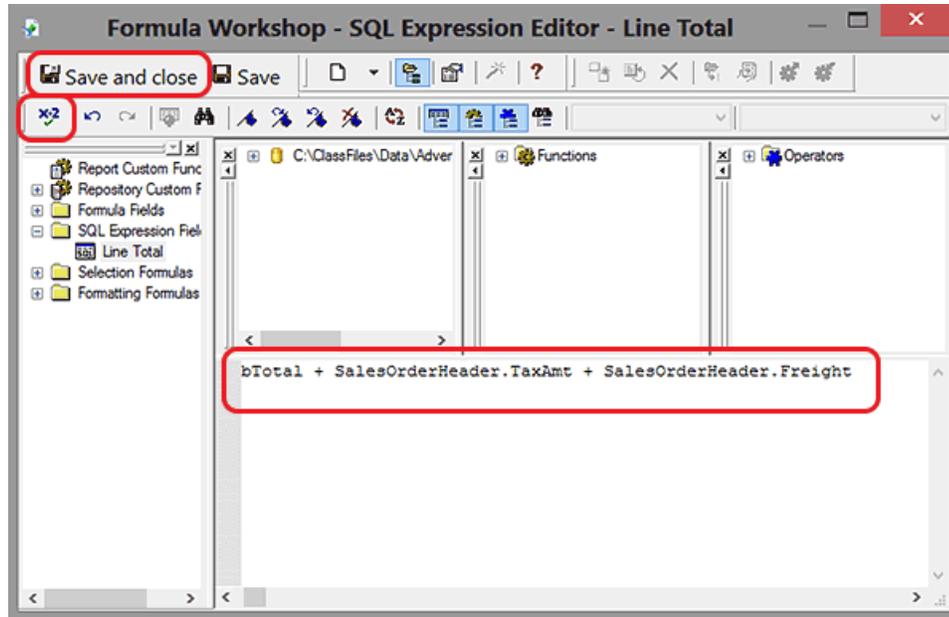
### ❖ 2.6.1. Using a SQL Expression

We will replace the current **Line Total** formula field in the Customer report with a SQL Expression field.

1. Open C:\ClassFiles\cry2011-working-with-SQL\Demos\Customer.rpt and select the **Design** tab.

2. In the **Field Explorer**, right-click the **SQL Expression Fields** node and select **New**.
3. In the **SQL Expression Name** dialog box, type “Line Total” and click **OK**.
4. In the **Formula Workshop - SQL Expression Editor - Line Total** dialog box, type the following in the expression editor:

```
SalesOrderHeader.SubTotal + SalesOrderHeader.TaxAmt + SalesOrderHeader.Freight
```



5. Verify the expression and then click **Save and close**.
6. In the report, remove the current **Line Total** formula field and replace it with the new **Line Total** SQL Expression field.
7. Move the new field title to the **Group Header #2b** section.
8. With the field title selected, click the **Bold** and **Underline** icons.

Customer	Customer Count	Total
@RegionDescription	Count of Customer.Account	Sum of @LineTotal
Account #	Customer Name	Line Total
{AccountNumber}-(CustomerTyp	{FirstName} {LastName}	%Line Total

salesReport

9. Preview and save the report.

Customer	Customer Count	Total
Region - Alabama	2	\$41.21
Account #	Customer Name	Line Total
AW00011533-I-1	Ebony Gill	\$2.53
AW00011533-I-2	Ebony Gill	\$38.68

FirstName	LastName	Name
Pamela	Ansman-W olfe	United States
David	Campbell	United States
Tete	Mensa-Annan	United States
Michael	Blythe	United States
Jillian	Carson	United States
Linda	Mitchell	United States
Shu	Ito	United States
Tsvi	Reiter	United States

Region - Alberta      15      \$24,826.92

# Exercise 2: Using SQL Expressions in Reports

 15 to 30 minutes

---

In this exercise, we will use a **SQL Expression** field to help improve the performance of the Sales Order report.

1. Create a new **Line Total** field using a SQL Expression and replace the existing **Line Total** formula field.

Evaluation  
Copy

## Conclusion

In this lesson, you learned:

- About the SQL SELECT statement and its different parts.
- How to make a JOIN within a SQL SELECT statement.
- How to add a SQL command to a report.
- How to use SQL Expressions in a report.



# LESSON 3

## Advanced Functions

---

### Topics Covered

- Report processing.
- Evaluation time functions.
- Variables.
- for loops.
- while do loops.

### Introduction

In this lesson, you will learn about report processing and how the evaluation time functions work, about variables, about for loops, and about while do loops.

Evaluation  
Copy

---

## 3.1. Evaluation Time Functions

When we preview or print a report from Crystal Reports, we do not see the work required to produce the output. In Crystal Reports, it is possible for the report processing to pass over the data up to three times. In fact, there are actually five potential steps involved in processing a report:

- Pre-Pass #1 -- Before passing through any data, the report will evaluate all formulas that are not based on field values. This allows these constant formulas to be evaluated once and the results used throughout the reporting process.
- Pass #1 -- This is the first time that the report reads the records from your data source. In this step, most of the basic operations such as retrieving data and grouping and sorting are performed.
- Pre-Pass #2 -- When special ordering and grouping is needed for a report, it is handled in this phase after the regular sorting and grouping have already been performed.

- Pass #2 -- This is the step of the process in which report formatting is actually performed. In addition, objects that are dependent on elements that were processed during Pass #1 are processed. In particular, this is the step when running totals, subreports, and group selection formulas are evaluated. Note that this step is performed on demand (if Pass #3 is not required), so the individual pages are not completely processed until they are viewed or printed.
- Pass #3 -- The total page count is determined during this step. This step will not run unless a Special Field based on the total page count is used in the report.

To help us take advantage of these steps, Crystal Reports has several built-in functions that allow us to change when items are evaluated during report processing. These functions are more like instructions to the formula about when it should be processed. Note, however, that sometimes we cannot use these functions to evaluate an item before it would normally be evaluated since it would not be ready to be processed and would result in an error message.

The four evaluation time functions are as follows:

Function	Purpose
BeforeReadingRecords	This function is used during the Pre-Pass #1 step and forces the formula to be evaluated before reading any records. The function is primarily used to ensure that a constant formula is processed prior to the Pass #1 step.
WhileReadingRecords	This function is used during the Pass #1 step and forces the formula to be evaluated when reading through the data. This can be used to prevent a constant formula from being processed during Pre-Pass #1 as well as ensuring that a formula has been processed prior to the Pre-Pass #2 step.
WhilePrintingRecords	This function is used during the Pass #2 step to force a formula to be evaluated while formatting the pages of the report. This can be used to prevent constant formulas from being processed during Pre-Pass #1, to prevent formulas from processing in the Pass #1 step, and to wait for Pass #2 to run.
EvaluateAfter(x)	This function is used to ensure that a formula is evaluated after another formula. This can be useful when one formula is dependent on the result of another formula.



## 3.2. Working with Variables

Variables act as placeholders for real data that is yet unknown, and when a value is entered or calculated, the variable will then temporarily hold that value for use later. Once a variable has passed the value to its proper destination, we often use code to reset it back to a null or 0 value, making it ready for use again.

Variable Scope referred to the reach the variable and where it may be used. Crystal Reports has the following variable scopes.

- Local - Limited to a single formula and they may not share the value with another.

```
//local variable
Local NumberVar x;
x := 5;
```

- Global - Limited to the main report and may be used throughout the report, but not used by the subreport.

```
//global variable
Global NumberVar x;
x := 5;
```

**Evaluation  
Copy**

- Shared - Limited to the main report and may be by the subreport.

```
//shared variable
Shared NumberVar x;
x := 5;
```

Basic rules must be followed in Crystal Reports when working with variables.

- Variables must be declared before we are allowed to use them.
- They must be given a unique name that does not match any existing name used in Crystal Reports to the data source.
- There are seven basic data types that we may use to declare a new variable.
  - NumberVar
  - CurrencyVar
  - StringVar
  - BooleanVar

- DateVar
  - TimeVar
  - DateTimeVar
- Variable can only hold one data type. Once they have been assigned a type, they may not be altered later to hold a different value type.

```
Local StringVar z := "Hey World";
```



### 3.3. For Loops

There are many ways to evaluate data in Crystal Reports. When you have a preset number of times something should happen, a For Loop is a good choice.

The following example would give each person with children the number of children at home times 3.

```
Local numberVar i;  
Local numberVar kidAmt := {Individual.NumberChildrenAtHome};  
Local numberVar tickets := 0;  
for i := 1 to kidAmt do tickets := tickets + 3;  
tickets;
```



### 3.4. While Do Loops

When we know the number times something should happen, the For Loop is a good choice. What happens when we don't know the number of times? In this case, a While Do Loop could benefit us by using a condition to test; if that test is true, then the loop occurs. Once the condition is false, the loop will complete that cycle, and then cease to run.

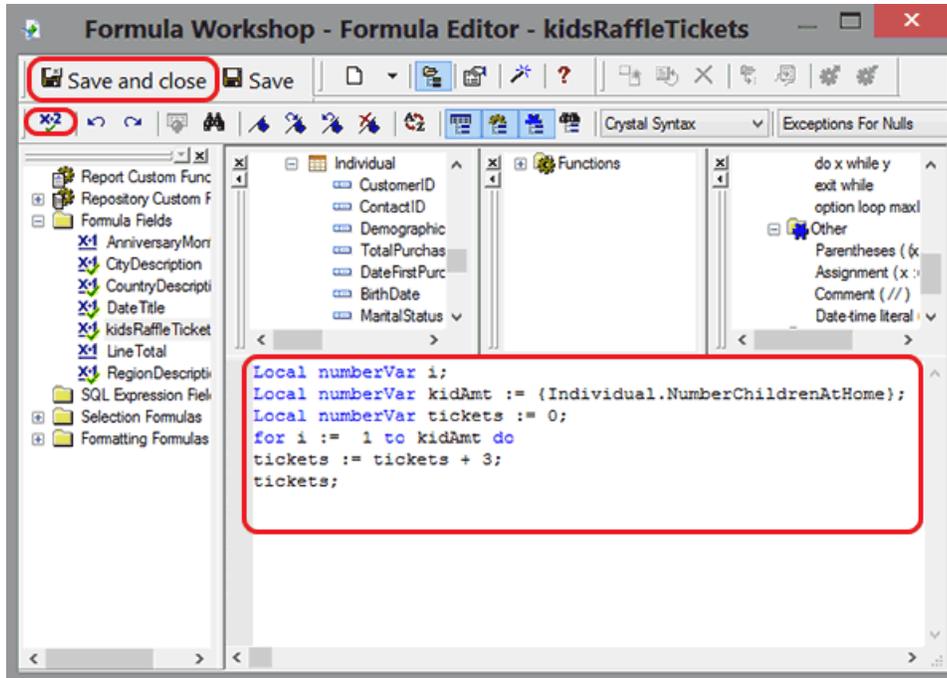
The following example would render the result of "7" as the number is in the seventh place.

```
Local StringVar inPhrase := "Where 2 Go";
Local NumberVar strLen := Length (inPhrase);
Local NumberVar result := -1;
Local NumberVar i := 1;
While i <= strLen And result = -1 Do(Local StringVar c := inPhrase [i];
If NumericText (c) Then result := i;
i := i + 1;);
result
```

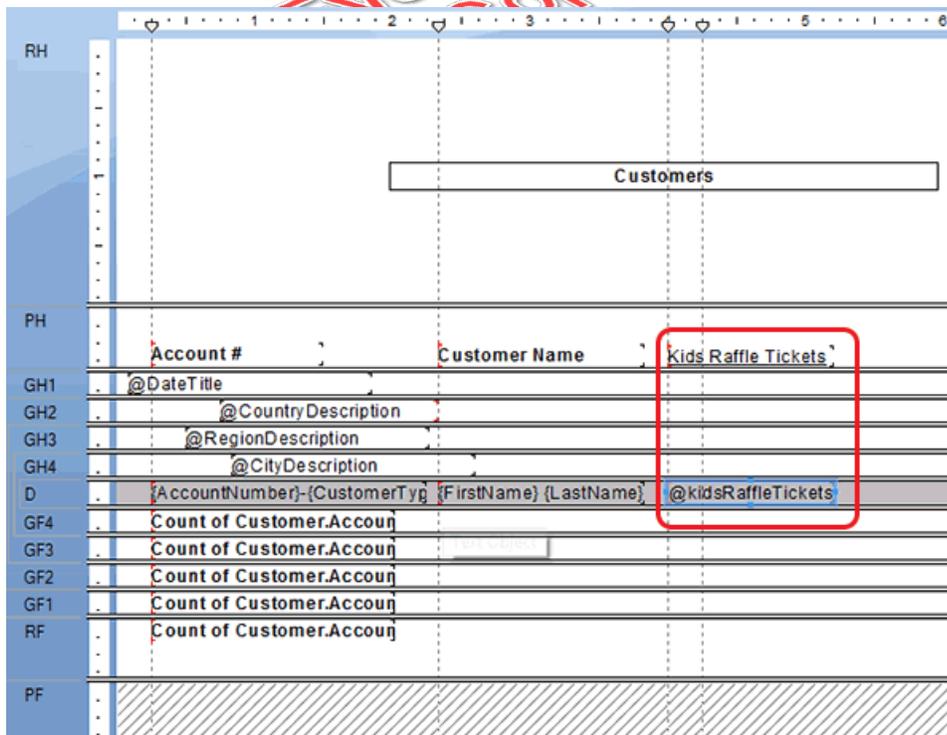
### ❖ 3.4.1. Working with Variables and Loops

1. Open cry2011-advanced-functions\Demos\Customer.rpt and select the **Design** tab.
2. In the **Field Explorer**, right-click the **Formula Fields** node and select **New**.
3. In the **Formula Name** dialog box, type "kidsRaffleTickets" and click **OK**.
4. In the **Formula Workshop - Formula Editor - kidsRaffleTickets** dialog box, type the following in the expression editor:

```
Local numberVar i;
Local numberVar kidAmt := {Individual.NumberChildrenAtHome};
Local numberVar tickets := 0;
for i := 1 to kidAmt do
tickets := tickets + 3;
tickets;
```



5. Verify the expression and then click **Save and close**.
6. Move the new field to the **Details** section.



- With the field title selected, click the **Bold** and **Underline** icons.
- Preview the report and scan through the pages for individuals with kids.

Account #	Customer Name	Kids Raffle Tickets
AW00016620-I-77	Cameron Thompson	0
AW00016662-I-78	Stacey Gao	0
AW00017247-I-79	Michael Moore	0
<b>5</b>		
City - Wollongong		
AW00016345-I-80	Francis Ramos	0
AW00017259-I-81	Trisha Wang	15
AW00016658-I-82	Ashlee Rale	0
AW00011004-I-83	Elizabeth Johnson	15
AW00011039-I-84	Marc Martin	0
AW00016517-I-85	Katrina Raji	3
AW00016742-I-86	Mitchell Pal	0
<b>7</b>		
<b>86</b>		
Region - Queensland		
City - Brisbane		
AW00016931-I-87	Dalton Lee	0
AW00017051-I-88	Jessie Ortega	6
AW00016607-I-89	Logan Gonzales	0
<b>3</b>		
City - Caloundra		
AW00017326-I-90	Paul Shakespear	0
AW00016693-I-91	Ethan Coleman	0
AW00017047-I-92	Kendra Rubio	0
AW00016522-I-93	Casey Pal	3
AW00025556-I-94	Joan Martin	0
<b>5</b>		
City - East Brisbane		
AW00011010-I-95	Jacquelyn Suarez	0

- Save and close the report.

# Exercise 3: Working with Variables and Loops

 15 to 25 minutes

---

In this exercise, you will use Variable and Loops to calculate the delay encountered if a shortage occurs. If a shortage does occur, the regular days to manufacture is multiplied by 2.

1. Open `cry2011-advanced-functions\Exercises\SalesOrder.rpt` and select the **Design** tab.
2. Create a formula to handle the forecast of a times 2 delay in time for product manufacturing.
3. Move the new field to the **Details** section.
4. Format to fit.
5. Preview the report.
6. Save and close the report.

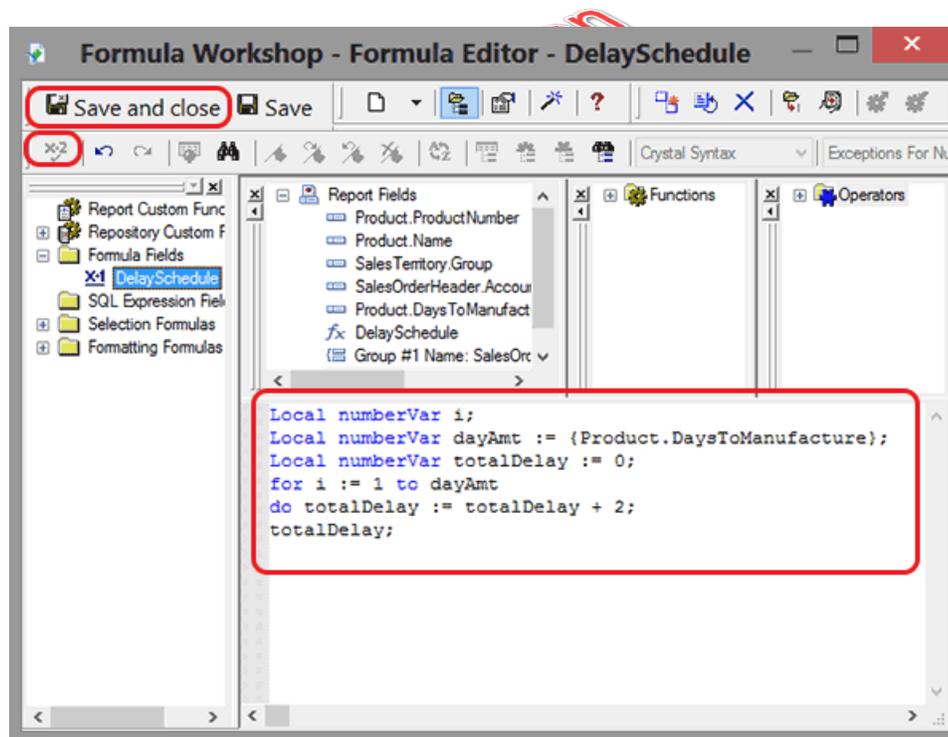


## Solution

The following describes one possible solution to the exercise.

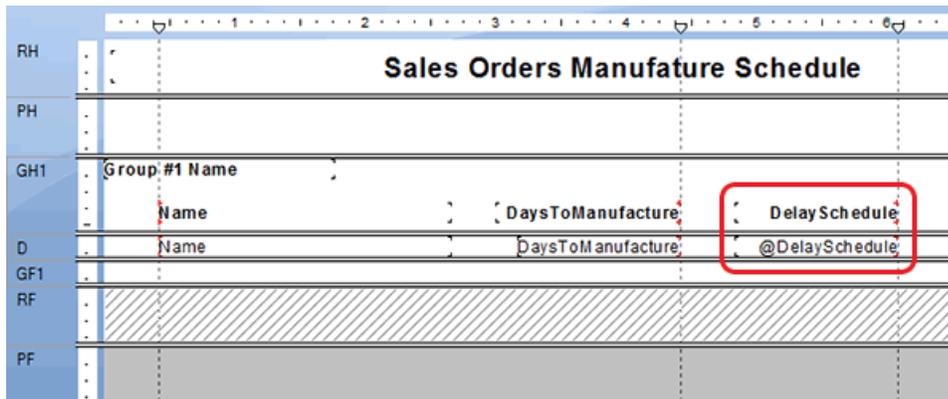
1. Open cry2011-advanced-functions\Exercises\SalesOrder.rpt and select the **Design** tab.
2. In the **Field Explorer**, right-click the **Formula Fields** node and select **New**.
3. In the **Formula Name** dialog box, type “DelaySchedule” and click **OK**.
4. In the **Formula Workshop - Formula Editor - DelaySchedule** dialog box, type the following in the expression editor:

```
Local numberVar i;  
Local numberVar dayAmt := {Product.DaysToManufacture};  
Local numberVar totalDelay := 0;  
for i := 1 to dayAmt  
do totalDelay := totalDelay + 2;  
totalDelay;
```



5. Verify the expression and then click **Save and close**.
6. Move the new field to the **Details** section.

- With the field title selected, click the **Bold** and **Underline** icons.



- Preview the report.

**Sales Orders Manufacture Schedule**

10-4020-000014

Name	DaysToManufacture	Delay Schedule
Water Bottle - 30 oz.	0	0
Classic Vest, S	0	0
Classic Vest, S	0	0
Classic Vest, S	0	0
Classic Vest, M	0	0
Women's Tights, S	0	0
Women's Tights, L	0	0
Short-Sleeve Classic Jersey, XL	0	0
Short-Sleeve Classic Jersey, XL	0	0
Short-Sleeve Classic Jersey, XL	0	0
Short-Sleeve Classic Jersey, L	0	0
Women's Mountain Shorts, S	0	0
Women's Mountain Shorts, S	0	0
Women's Mountain Shorts, S	0	0
Women's Mountain Shorts, M	0	0
Women's Mountain Shorts, M	0	0
Women's Mountain Shorts, M	0	0
Women's Mountain Shorts, L	0	0
Women's Mountain Shorts, L	0	0
Women's Mountain Shorts, L	0	0
Women's Mountain Shorts, L	0	0
HL Mountain Seat/Saddle	1	2
Men's Bib-Shorts, M	0	0
ML Mountain Rear Wheel	1	2
Hitch Rack - 4-Bike	0	0

- Save and close the report.

## Conclusion

In this lesson, you have learned

- Report processing and how the evaluation time functions work.
- About variables.
- About for loops.
- About while do loops.

Evaluation  
Copy

# LESSON 4

## Geographic Mapping

---

### Topics Covered

- Geographic map based on a data source.

### Introduction

Crystal Reports has the ability to extract data from a source and display it based on geographic location. This data can be displayed using several map models included. Each time the report runs, the data is updated.



### 4.1. Geographic Maps

#### ❖ 4.1.1. Map basics

Evaluation  
Copy

Only a basic address, some numeric data to summarize, or count of records is needed to create the geographic map. The numeric data is used to set the frequency as displayed in the map itself.

There are five map types available:

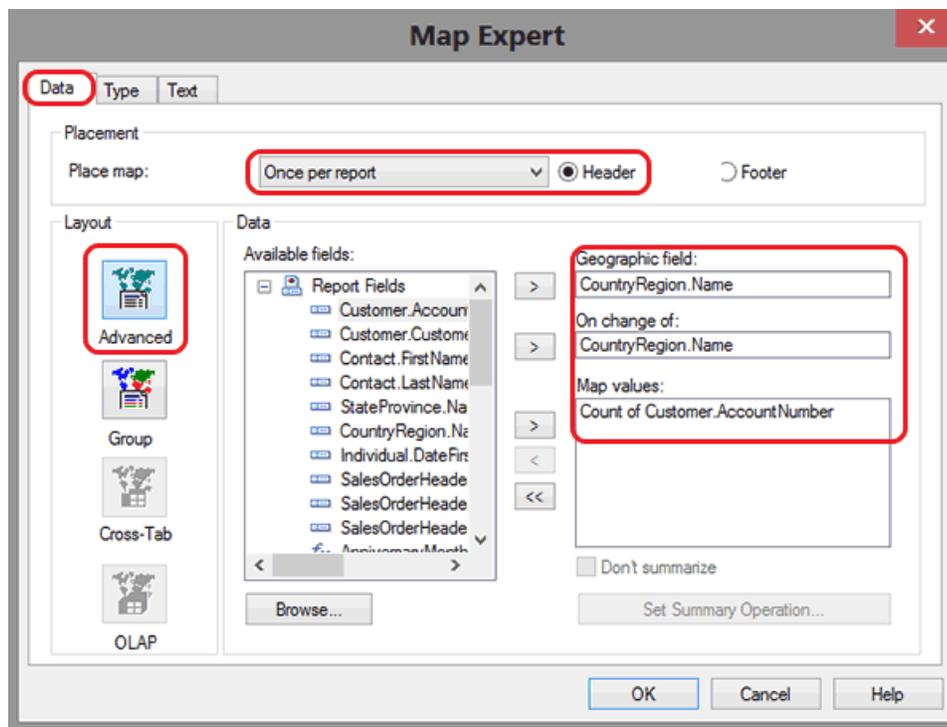
- Ranged: Map intervals are assigned a specific color by range value.
- Equal Count: Map intervals are assigned to be as close to equal as possible based on the numeric quantity of the summary values.
- Equal Range: Map intervals are assigned to be equal based on the individual regions and then summary values.
- Natural Break: Map intervals are calculated based on difference between summary of values and the average of values per interval.
- Standard Deviation: Map intervals break at the average of the data values.

Based on the section that we add the map, the scope of data and number of maps displayed throughout the report will match the scope of the section.

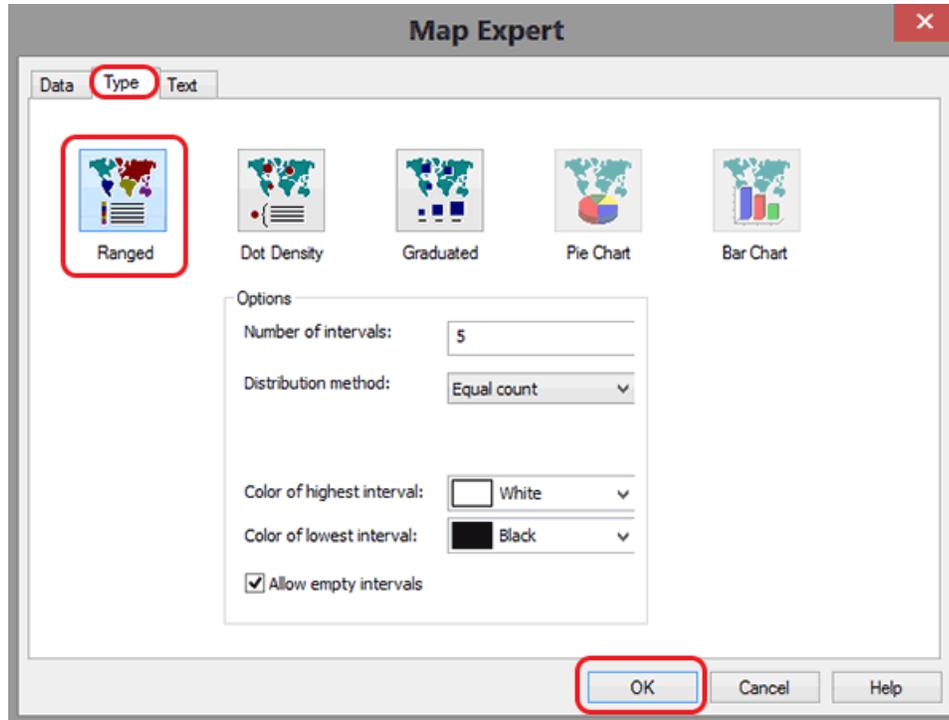
## ❖ 4.1.2. Adding a Geographic Chart

We will add a map chart based on the the number of customer in each region to the Customer report.

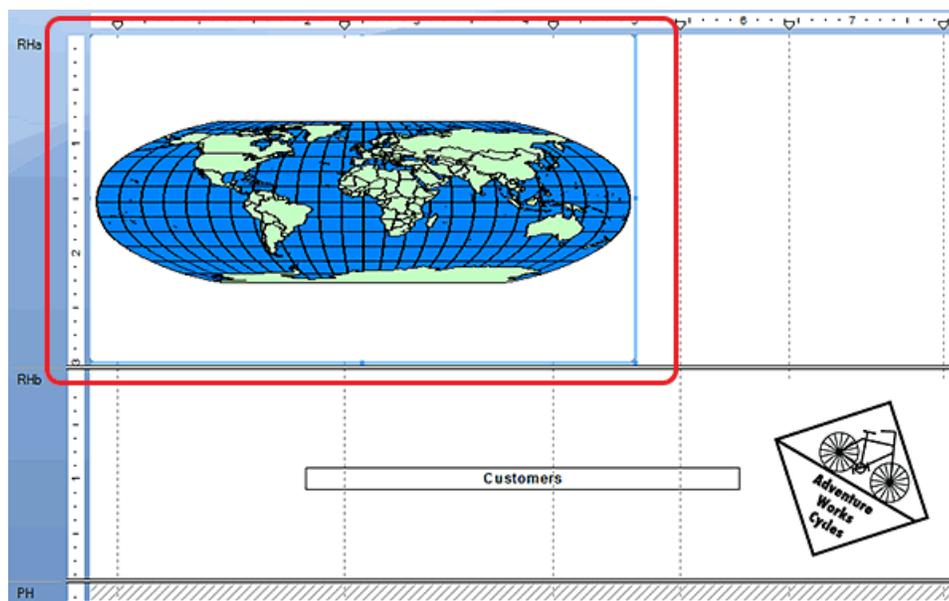
1. Open cry2011-geographic-mapping\Demos\Customer.rpt and select the **Design** tab.
2. Click the  **Insert Map** icon.
3. In the **Map Expert** dialog box, select the **Data** tab. In the **Layout** area, select **Advanced**. Choose **CountryRegion.Name** for the **Geographic field** and the **On change of** area. Choose **Customer.AccountNumber** for the Map values area.



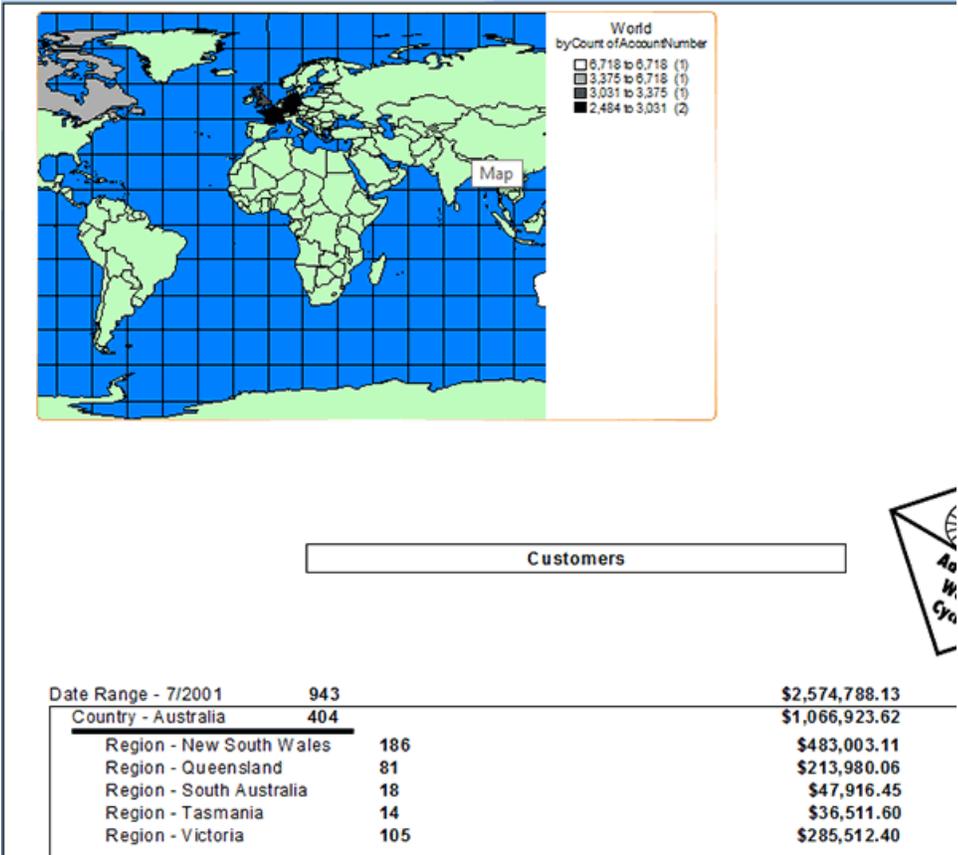
4. Select the **Type** tab, choose **Ranged**. Click **OK**.



5. A chart is placed in your **Report Header**, you may need to right-click and choose **Chart Expert** if you need to make changes.
6. Format if needed to make room for everything.



7. Preview report.



8. Save and close.

## Exercise 4: Adding a Geographic Map

 15 to 25 minutes

---

In this exercise, you will add a geographic map to a report based on number of products sold in each city.

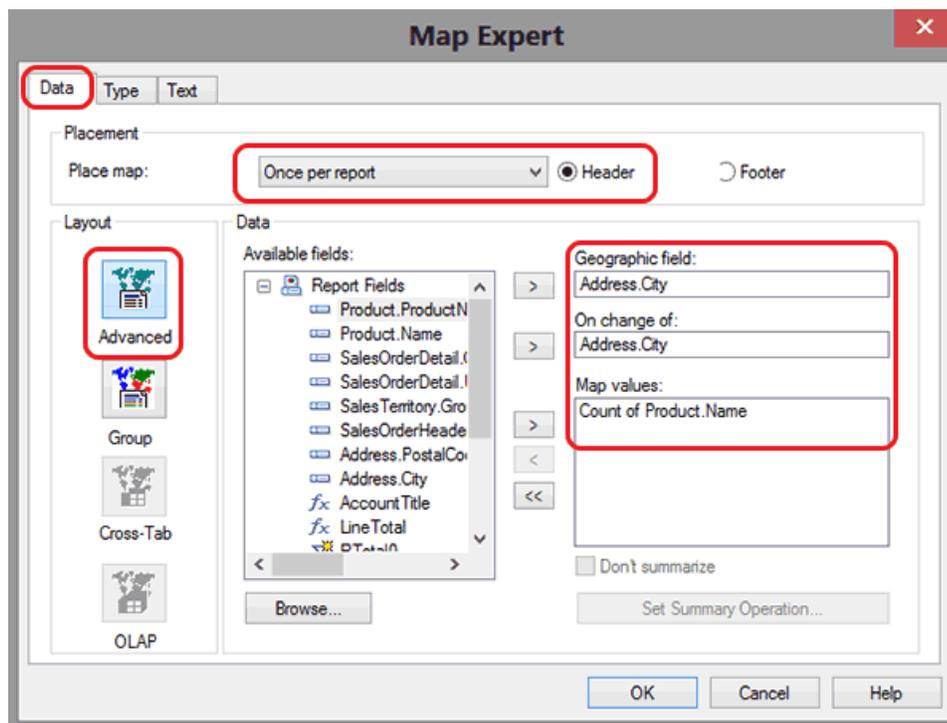
1. Open `cry2011-geographic-mapping\Exercises\SalesOrder.rpt` and select the **Design** tab.
2. Place a chart in your **Report Header** area based on **Address.City** for the **Geographic field** and the **On change of** area. Choose **Product.Name** for the Map values area.
3. You may need to right-click and choose **Chart Expert** if you need to make changes.
4. Format if needed to make room for everything.
5. Preview report.
6. Save and close.

## Solution

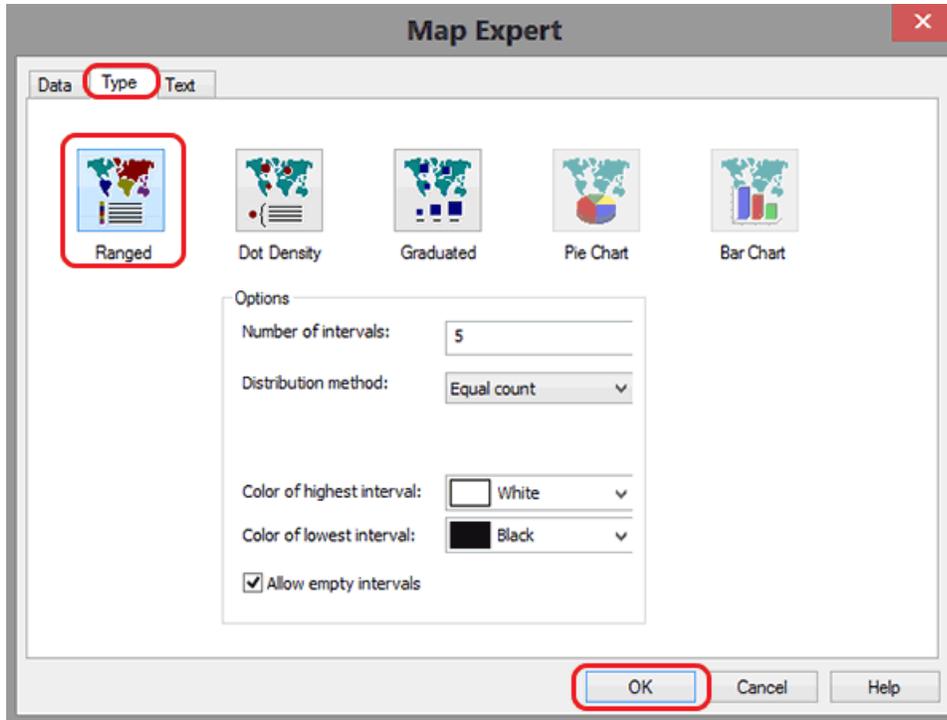
---

The following describes one possible solution to the exercise.

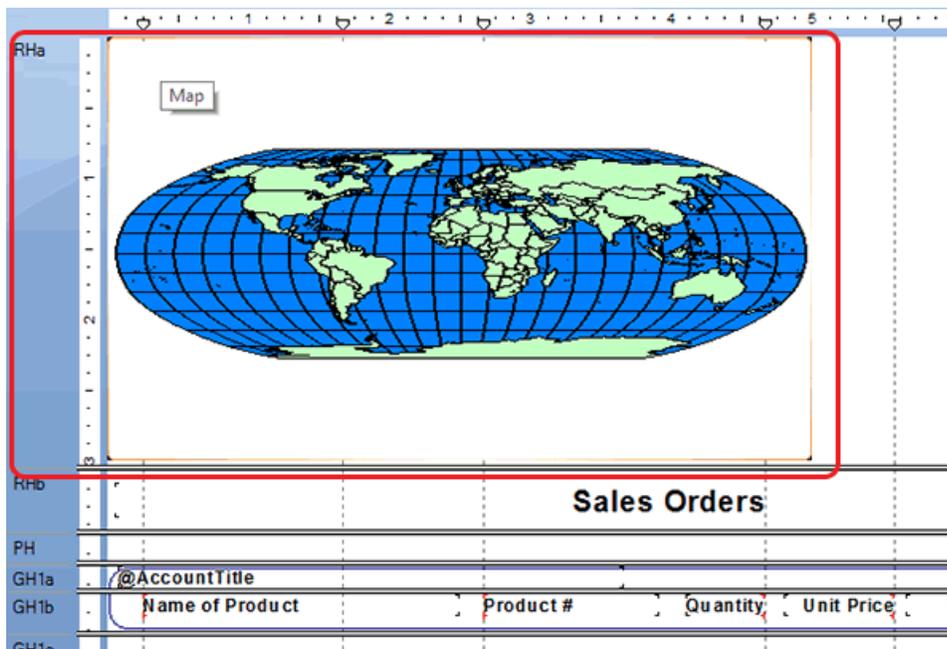
1. Open cry2011-geographic-mapping\Exercises\SalesOrder.rpt and select the **Design** tab.
2. Click the  **Insert Map** icon.
3. In the **Map Expert** dialog box, select the **Data** tab. In the **Layout** area, select **Advanced**. Choose **Address.City** for the **Geographic field** and the **On change of** area. Choose **Product.Name** for the Map values area.



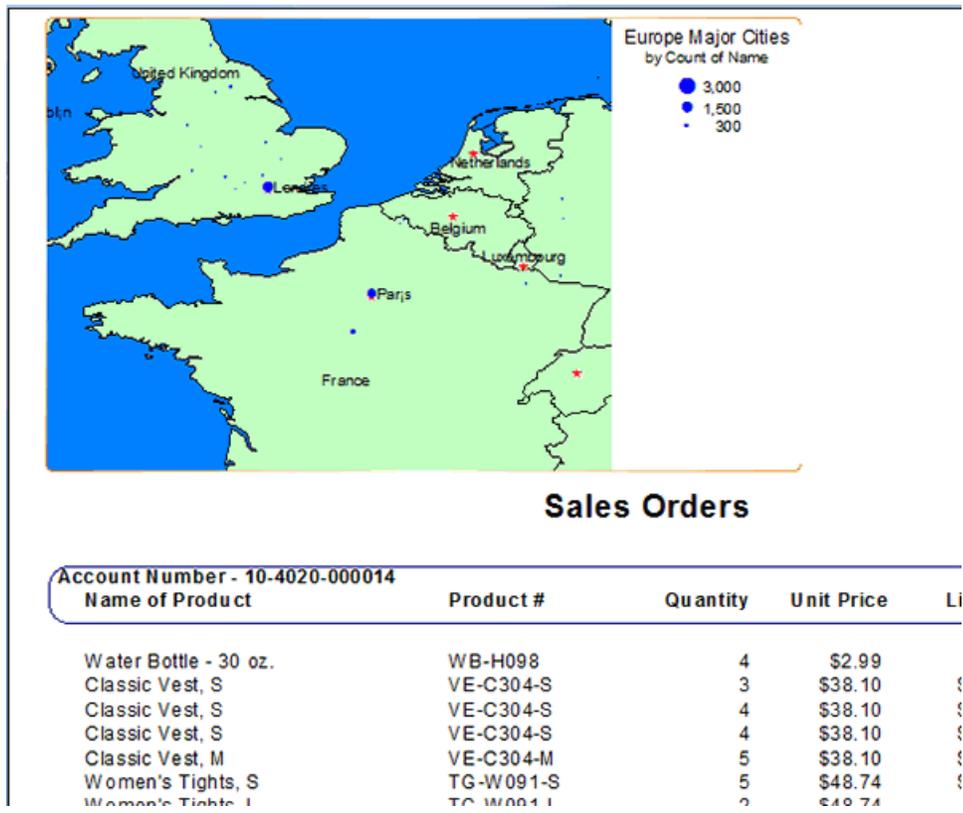
4. Select the **Type** tab, choose **Ranged**. Click **OK**.



5. A chart is placed in your **Report Header a**, you may need to right-click and choose **Chart Expert** if you need to make changes.
6. Format if needed to make room for everything.



7. Preview report.



8. Save and close.

## Conclusion

In this lesson, you have learned

- To create a geographic map based on a data source.

# LESSON 5

## Establishing a Database Connection

---

### Topics Covered

- Creating a new database connection.

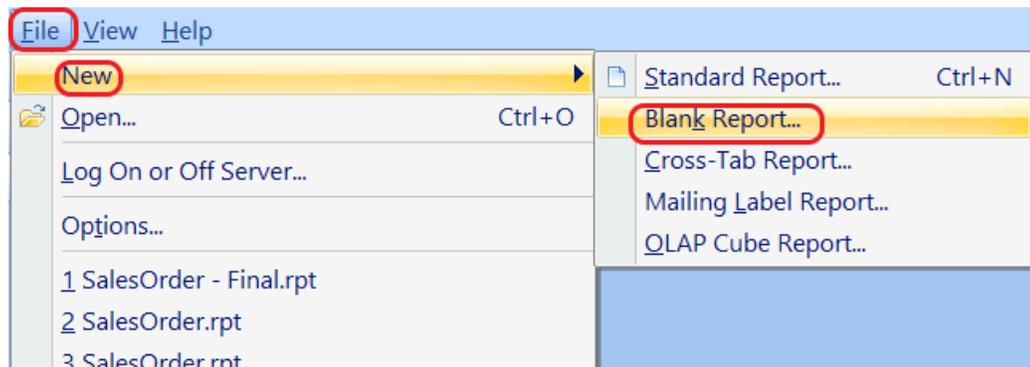
### Introduction

In this lesson we will learn to create a new database connection to the AdventureWorks database.

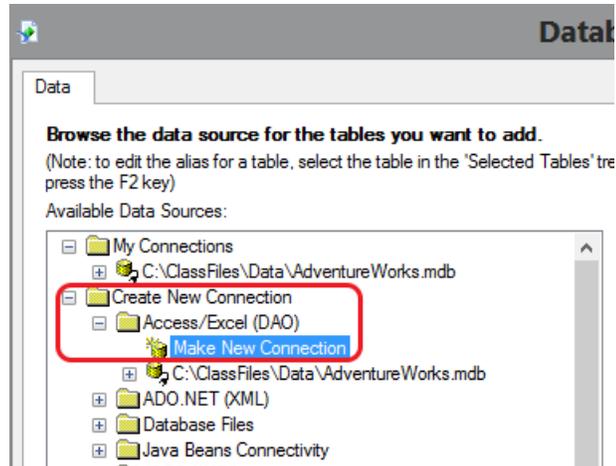
\*

### 5.1. Connecting to the AdventureWorks Database

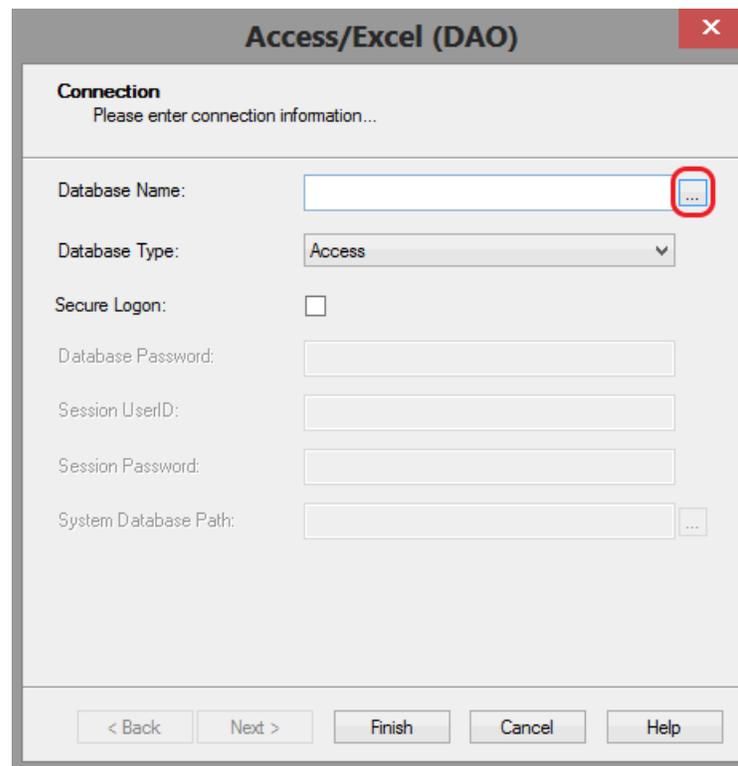
1. From the menu, select **File > New > Blank Report**.



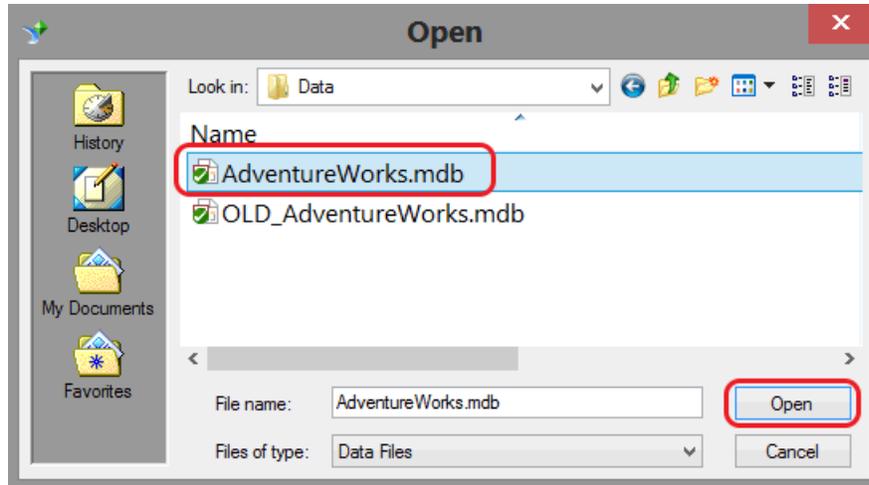
2. In the **Database Expert** dialog box, expand the **Create New Connection > Access/Excel (DAO)** node.



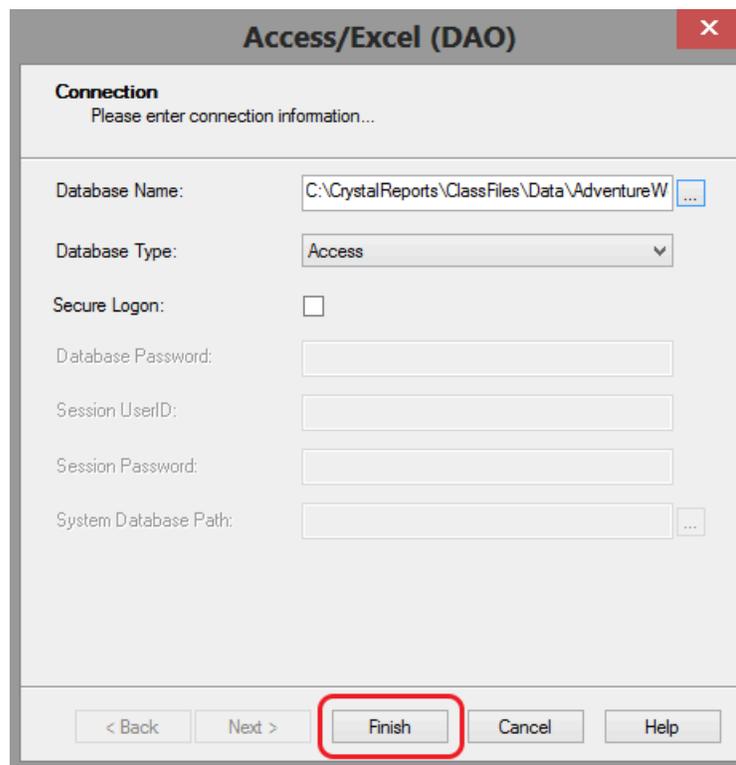
3. Select the **Make New Connection** node. The **Access/Excel (DAO)** wizard starts.



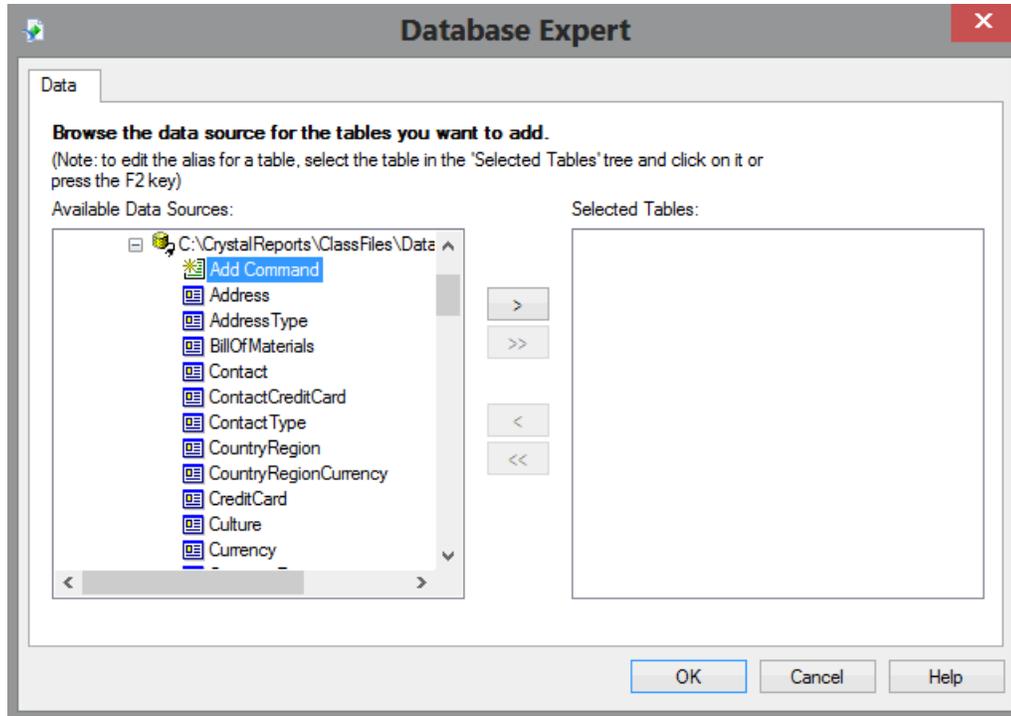
4. Click the  **Open** icon.
5. Locate and select **C:\ClassFiles\Data\AdventureWorks.mdb**.



6. Click **Open**.
7. Click **Finish**.



8. Choose the tables you need and proceed.



## Conclusion

In this lesson, you learned how to create a new database connection.