Working with Formulas

Rishi H. Heerasing
Lecturer

School of Innovative Technologies and Engineering
UTM

Learning Outcomes

- Understanding Crystal Formula Syntax
- Using the Formula Workshop
- Using the Formula Editor
- The Formula Editor Toolbar
- Performing Simple Number Calculations
- Manipulating Dates with Formulas
- Creating Boolean (True/False) Formulas
- Creating String Formulas

Working with Formulas

- Formulas in Crystal Reports can be created, edited, and modified using one of two languages. The Crystal syntax is the most used language, but the Basic syntax is also available.
- Both languages are equal in their functionality, meaning that if something was added to Crystal syntax, it was also added to Basic. The reason you're given a choice is for your comfort; you can use whichever language you are more comfortable with.

Working with Formulas

Types of formulas I:

Report formulas

Report formulas are formulas that you create to stand alone in a report. For example, a formula that calculates the days between the order date and the shipping date is a report formula.

Conditional formatting formulas

Formatting formulas change the layout and design of a report, as well as the appearance of text, database fields, objects, or entire report sections. You format text through the Format Editor. If you need to create a formatting formula, you access the Formula Workshop from the Format Editor.

Selection formulas

Selection formulas specify and limit the records and groups that appear in a report. You can either enter these formulas directly or specify the selection using the Select Expert. Crystal Reports then generates the record selection and group selection formula. You have the option to manually edit these formulas, but you must use Crystal syntax.

Working with Formulas

Types of formulas II:

Search formulas

Search formulas help you locate data in your report. Like selection formulas, you normally do not enter these formulas directly, but instead specify the search criteria using the Search Expert. Crystal Reports generates the formula. You have the option to manually edit these formulas, but you must use Crystal syntax.

Running Total condition formulas

Running Total condition formulas let you define the condition upon which your running total will be evaluated or reset.

Alerting formulas

Alerting formulas help you define conditions and messages for report alerts.

Working with Formulas

- Differences between Crystal and Basic Syntax.

Description	Crystal	Basic
Variable declarations	`StringVar`	`Dim <name> As <type>`
Statement endings	`;`	None required
Comment characters	`//`	`'`
Variable assignment	`:=`	`=`
Formula statement	None required	Required
Formula returns	None required	`Return` statement
Multiline statement indicators	None required	_
`If` statement ending	`;`	`End If`

- Record selection and group selection formulas cannot be written in Basic syntax.
- Report processing is not slowed down by using Basic syntax. Reports using Basic syntax formulas can run on any machine that Crystal Reports runs on.
- Using Basic syntax formulas does not require distributing any additional files with your reports.

Working with Formulas

- Several variations of brackets are used within the formula language. This is a way to recall them phonetically:

  *{ } **F**rench = **Fields***

  For example, {Table.Field} is used to refer to fields, formula fields, or parameter fields in the report definition.

  *[ ] **S**quare = **Selected***

  For example, {Table.Field}[1] returns only the first character of a string field. Square brackets are used for indexes on array types (for example, strings or array data types).

  *( ) **P**arenthesis = **Parameters***

  For example, Function ({Table.Field}) passes the field to the function. Parentheses are used to define which parts of a calculation or formula should be performed first (that is, defines order of precedence for mathematical and non-mathematical operations).

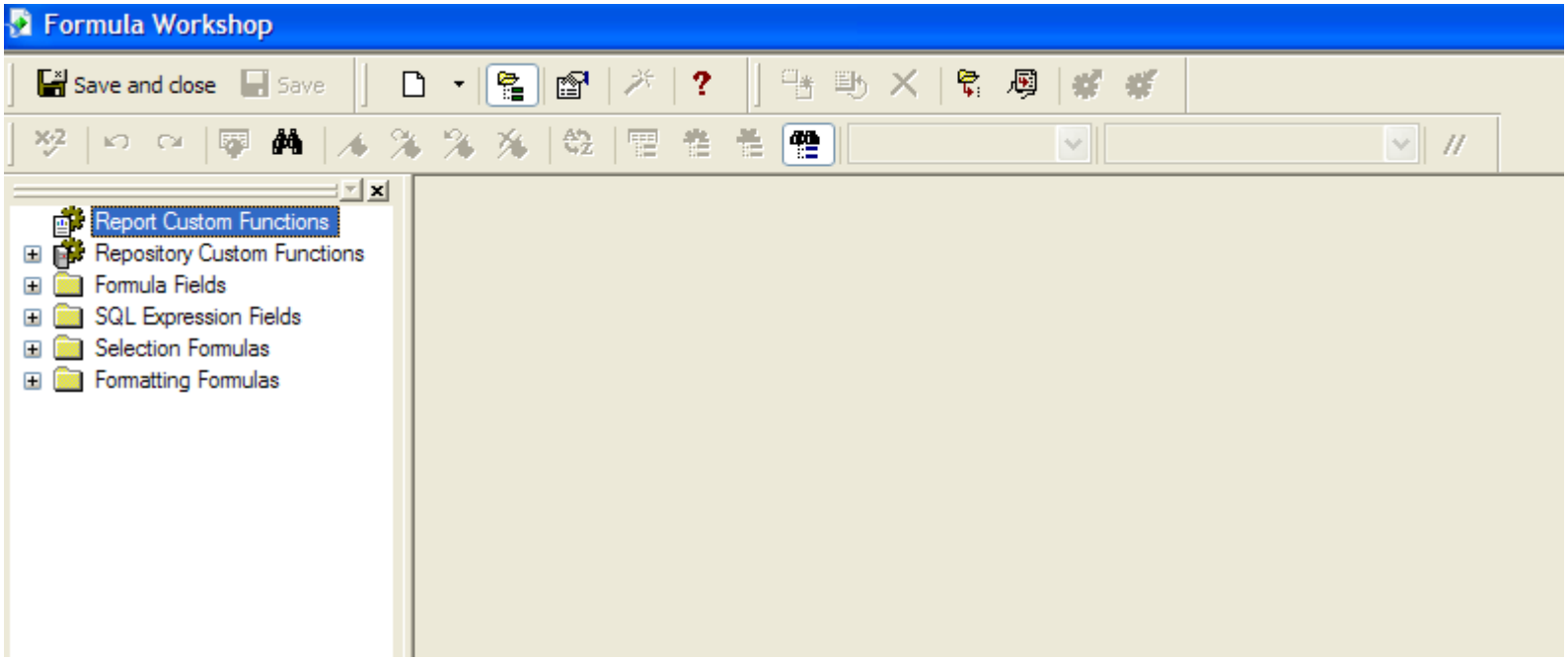
Working with Formulas

- As with brackets, symbols in the formula language (or in the icons) have specific meaning as well

• @ = Formula	{@Formula} is a formula field
• ? = Parameter	{?Param} is a parameter field
• # = Running Total	{#RunTtl} is a running total field
• Σ= Summary	Σ*fieldName* is a summary field on the report
• % = SQL Expression	{%SQL} is a SQL expression field

Working with Formulas

- The Formula Workshop can be invoked anytime by clicking on its icon in the Toolbar.

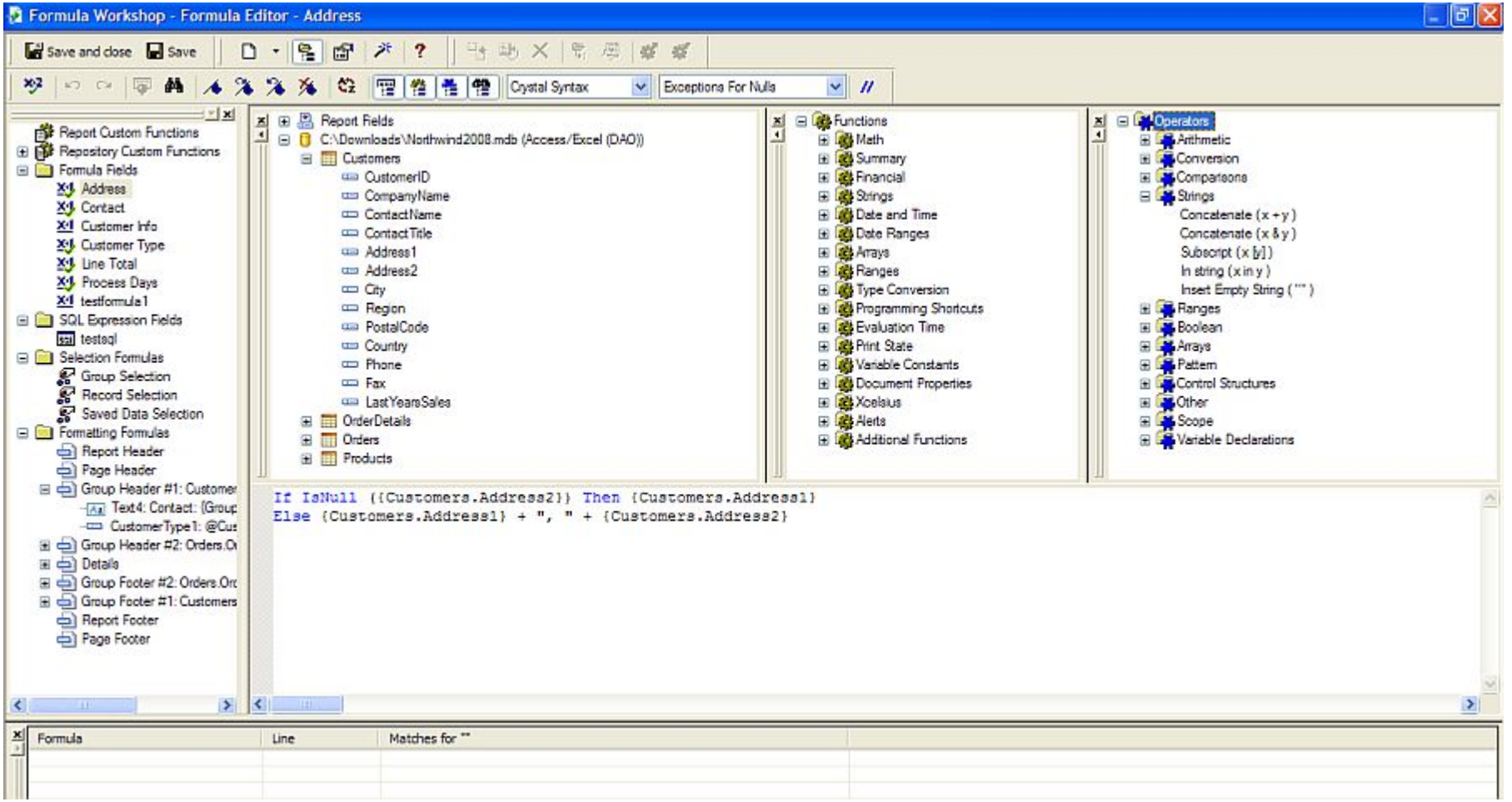


Working with Formulas

- The upper Toolbar contains options to Save, Allow New types of formulas to be created, Using Formula Expert, Rename and other various viewing Options.
- The lower Toolbar contains a formula checker, Browse Data, Find and Search and Bookmarking facilities, Results, Syntax Chooser, Toggling vertical panes, Quick commenting.

Working with Formulas

- The Formula Editor is the main component of the Formula Workshop.



Working with Formulas

- The Formula Editor consists of four main windows.

Window	Description of contents
Report Fields	Report fields contain all database fields accessible for your report. They also contain any formulas or groups already created for the report.
Functions	Functions are prebuilt procedures that return values. They perform calculations such as average, sum, count, sin, trim, and uppercase. Custom functions are also listed in this window.
Operators	Operators are the "action verbs" you use in formulas. They describe an operation or an action to take place between two or more values. Examples of operators: add, subtract, less than, and greater than.
Formula text window	Area where you create a formula.

Working with Formulas

- Creating a Formula is straightforward as long as you know what you want to achieve. Let us create a simple formula that displays the age of employees.
- Select Northwind2008 and import *Employees* table. This table contains a *BirthDate* field which we are going to use to calculate the age.
- Right-Click on Formula Field in the Field Explorer tab, and select New... Name the field ***Age***. You are then automatically brought in Formula Workshop.

Working with Formulas

- The next step is to write the formula to calculate the age. We can calculate the age of someone by calculating the difference between the present and his/her birth date. We will search if such a predefined function exist.
- Functions are usually categorized under generic nodes. So a good place to start is to search under Date and Time header. Expanding the node, we find a sub-node called DateDiff. Expanding the sub-node, give us two overloaded functions: one with 3 parameter and another with four. We double-click on the first one.

Working with Formulas

- The function is automatically inserted in the Formula Window.
- Now the first Parameter is Interval Type. To get more info about the function. Use the help file. Click on **?** Icon and do a search for DateDiff.
- When you get there, it will provide you with all the info, you will need and even illustrate some examples.
- We need to find the difference in years, so we will have to insert '*yyyy*' as IntervalType.

Working with Formulas

- According to the Help file, the EndDate must be bigger than the StartDate. So, StartDate must be the BirthDate of the employees. Drag the appropriate field from the Field Window into the Formula Window. Now the last parameter is EndDate which must be now. There exist a similar command called CurrentDate which can be used. No need to choose CurrentDateTime as Time is not needed here.
- **Note**: You can display the list of available functions or keyword auto-complete anytime by pressing CRTL-SPACE.

Working with Formulas

- Check the formula and Save and Close.
- Now you can place the Age formula Field on your report. Note that all formula fields have an "@" prefixed to it.
- Now let us also add a brief EmploymentHistory of each employee. A sample output should be.

  "Rishi joined Northwind on the July 17, 2002.

  This was a Wednesday."
- We could use a Text field but we'll use a formula instead.

Working with Formulas

- Just like before create a Formula field called EmploymentHistory.
- Now in the Formula Window, we will need to add the FirstName field. Text are added by simply enclosing in double quotes and concatenated using the '+' sign.
- Then we need to convert the HireDate in the appropriate format.
- If we look in Strings under Functions; we find a ToText() function. Using the help file once again will allow us to format a TimeDate Field.

Working with Formulas

- By rummaging through the help file, we find that the correct syntax is:

  ToText({Employees.HireDate}, "MMMM d, yyyy.")

- Next we need to enter a newline or a line feed. Here we have to make use of a special function called Chrw(x) where x is a ASCII/Unicode value of the character. We enter 13 as it is the code for Line Feed.
- This function is useful for displaying special characters as well as non-printable characters.

Working with Formulas

- Finally, to display the last line we use the same function as before except that we only want to return the WeekDay:

  ToText({Employees.HireDate}, "dddd.")

- Next, suppose now we also want to display two more fields: 20th Anniversary (the date the employee will have 20 years service) and Model Employee (if the employee is still in service for more than 6 years (i.e. 2190 days). The last field should return either True or False (i.e. Boolean)

Working with Formulas

Task 7a

- Create a report as faithfully as this one.
- Database is Northwind2008.mdb.
- Table is Employees.
- Save your report as 7a.rpt

Working with Formulas

Task 7b

- Create a report as faithfully as this one.
- Database is Northwind2008.mdb.
- Tables are *Order*, *Order Details*, *Products*.
- Percentage Discount = (RetailPrice-UnitPrice) / RetailPrice * 100
- VAT is at 15%
- Display only Top 5 Groups with highest Line Totals
- Save your report as 7a.rpt

Working with Formulas

- We can also use formulas to perform simple numerical operations.
- Select Northwind2008 and import *Employees* table. This table contains a *BirthDate* field which we are going to use to calculate the age.
- Right-Click on Formula Field in the Field Explorer tab, and select New... Name the field ***Age***. You are then automatically brought in Formula Workshop.