1. MCTS Self-Paced Training Kit (Exam 70-448): Microsoft SQL Server 2008 - Business Intelligence Development
2. Microsoft® SQL Server® 2012 Reporting Services. Brian Larson 2. Microsoft SQL Server Reporting Services Recipes 2010. Paul Turley, Robert M. Bruckner
3. [http://msdn.microsoft.com/en-us/library/hh965699(v=sql.110).aspx](http://msdn.microsoft.com/en-us/library/hh965699%28v=sql.110%29.aspx)
4. Pro SQL Server 2012 Reporting Services. McDonald B., McGehee Sh., Landrum R.
5. Professional Microsoft SQL Server 2012 Reporting Services. Turley P.

A perfect report solves a specific problem for the client. It’s good looking, informative, processes fast, and correct. Note that the cost of its maintenance should be minimal.

Let us elaborate on this:

“It solves a specific problem” – request a fleshed out user story from the client. If you receive none, then write it yourself. Figure out the details of business logic and request formal or informal approval from the client. This won’t cause the development and testing drag on for too long and will protect it from endless changes.

“Good looking” – use graphics, different controls, colors, groups etc. Find or create a report template that you can use to add client’s reports – either all or some of them. 99% of reports look rather boring, and remember that it’s important for us to make a good impression for our clients, so nice looking templates will always be evaluated and looked at. NOTE: if the report must be printed, then it narrows the opportunities for creativity. It’s important to clarify this point before beginning.

“Informative” – make easy to read reports if possible:

* Don’t overload the with data
* Don’t overload them with controls
* Make navigation if necessary, but don’t overuse it
* Remember that a report is based on many types of data, and the amount of data provided is not normally limited by your supervisors or management. A diagram with a thousand columns is unreadable, and nobody actually reads 100 page reports (if it’s not a bank statement)
* 9 out of 10 times the person who reads a report interprets it as “good” or “bad”, “fast” or “slow” etc. A simple diagram on the first page of the report usually satisfies most requirements of the client. There’s another variant – design and present a separate dashboard, and don’t let the client overload it with data.

“Fast” – whatever happens, the end-user must see the report. Two of the slowest stages of generating a report are obtaining samples from the SQL Server and rendering on the SSRS side. You need to remember that the simplest request can give thousands and even millions of lines, and the physical reading from the HDD is the slowest operation that is poorly amenable to parallelization. Furthermore, rendering of large amounts of data is usually slow, and having lots of controls make rendering go even slower. It can last for hours and be terminated by a timeout. Therefore, before making the report, you must find out the following:

* What parameter values will make report execution longer?
* Consider the growth rate of the database and figure out how big the database will be in a year
* Discover the limit values of parameters
* Discover all the timeout values
* All possible modifications (even sorting) should be performed on the SQL Server side
* Reduce the amount of data sent to the SSRS to an absolute minimum
* Remove all the possible calculations and conversions from RDL
* If this doesn’t help, then recommend to the client to change the report, its design, and the method of its delivery to end users.

a. 1 (Chapter 13 Lesson 1)

b. 5 (Chapter 2)

a. Install and configure Reporting Services

b. SSRS components (web service, report manager, report viewer, SQL Server catalog)

a. SQL Server 2012

b. SQL Server Data Tools

c. Database samples called “Adventure Works for SQL Server 2012” - <http://sqlserversamples.codeplex.com/>

a. Install Reporting Services (native) and SQL Server Data Tools (SSDT) on the virtual machine

b. Configure the SSRS using “Reporting Services Configuration Manager”

c. Enable Remote Errors

d. Connect to the home page using this link - <http://localhost/Reports>

e. Create a “SSRS\_Course” folder.

* 1 Chapter 10 Lesson 2
* 1 Chapter 11 Lesson 3
* 4 Chapter 3, Chapter 8
* Learn how to create data sources, datasets (shared, private)
* Learn how to create a report, add a table to the report and format fields
* Learn how to use expressions
* Learn how to deploy reports on the SSRS server

SQL Data Tools (SSDT)

* Create Report Server Project in SQL Server Data Tools
* Create Shared Datasource – Adventure Works 2012 on localhost
* Create the procedure:

Create procedure ssrsTask01 as

SELECT soh.OrderDate AS DATE

   ,soh.SalesOrderNumber AS [Order]

   ,st.Name AS TerritoryName

   ,ppc.NAME AS ProductCategory

   ,pps.NAME AS ProductSubcategory

   ,pp.NAME AS Product

   ,SUM(sd.OrderQty) AS Qty

   ,SUM(sd.LineTotal) AS LineTotal

FROM Sales.SalesPerson AS sp

INNER JOIN Sales.SalesOrderHeader AS soh ON sp.BusinessEntityID = soh.SalesPersonID

INNER JOIN Sales.SalesOrderDetail AS sd ON sd.SalesOrderID = soh.SalesOrderID

INNER JOIN Production.Product AS pp ON sd.ProductID = pp.ProductID

INNER JOIN Production.ProductSubcategory AS pps ON pp.ProductSubcategoryID = pps.ProductSubcategoryID

INNER JOIN Production.ProductCategory AS ppc ON ppc.ProductCategoryID = pps.ProductCategoryID

INNER JOIN Sales.SalesTerritory AS st ON soh.TerritoryID = st.TerritoryID

WHERE ppc.NAME = 'Clothing'

GROUP BY ppc.NAME

   ,soh.OrderDate

   ,soh.SalesOrderNumber

   ,pps.NAME

   ,pp.NAME

   ,soh.SalesPersonID

   ,st.Name

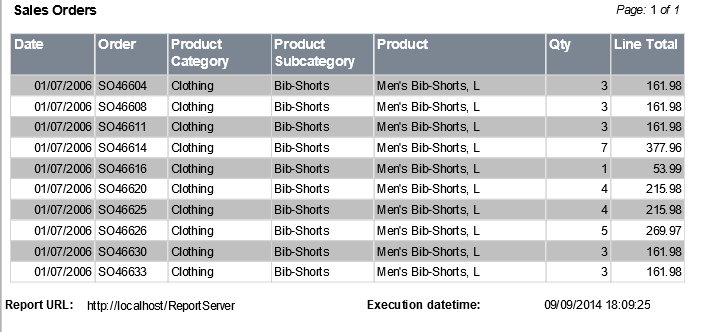
* Create Shared Dataset based on ssrsTask01
* Create the report - Sales Orders Lesson 2

The report must use the Datasource and Dataset created earlier

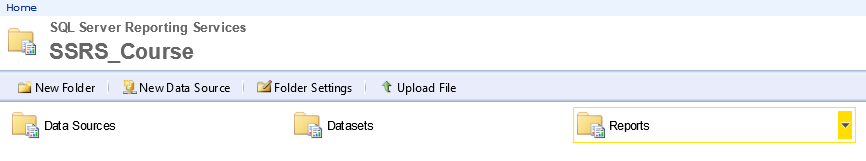
The report must include:

* Header: name of the report, page numbers, a total number of pages
* Footer: Report Url, Execution Time, Execution Date/Time
* A table with the following fields: Date, Order, Product Category, Product Subcategory, Product, Qty, LineTotal
* Lines in the table must be black and white
* The table’s header must be repeated on each page
* The format of “Date” and “Line Total” fields must follow the example below.

Example:



- Deploy Datasource, Dataset, SSRS server-based report in the “SSRS\_Course” folder (lesson 1)



1 Chapter 11 Lesson 1

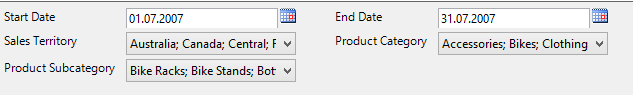
* Learn about report parameters and variables
* Learn about Databound parameters, multi-value parameters
* Learn how to identify variables.

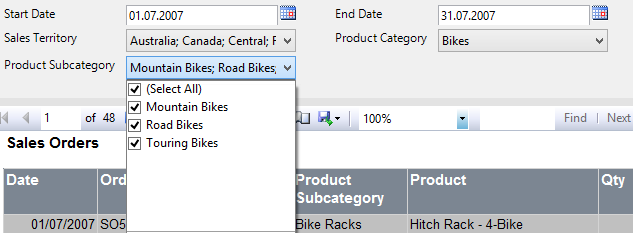
SQL Server Data Tools (SSDT)

* Create a copy of the “Sales Orders Lesson 2.rdl” report and name it “Sales Orders Lesson 3”
* Change Dataset for SalesOrders in a way that it will receive the following parameters - @StartDate, @EndDate, @SalesTerritory, @ProductCategory, @ProductSubcategory, and so it will draw orders for the specific range of dates, selected areas and categories/subcategories. In order to do this, create a new procedure called “ssrsTask02”.
* Add the appropriate parameters to the “Sales Orders Lesson 3” report. All the parameters must have values by default. Parameters such as “Sales Territory”, “Product Category” and “Product Subcategory” must allow for selecting several values. Values of the “Product Subcategory” parameter depend on the selected values of the “Product Category” parameter.
* Display the following message: “The result is empty. Try to refine parameter values”
* Create a copy of the report and name it as “Sales Orders Lesson 3\_1”

     - Add “English Speaking” items in the “Sales Territory” parameter; implement logic in the “ssrsTask02\_1” procedure.

     - If the report receives a blank sample on the given set of parameters, display “No Data Message”

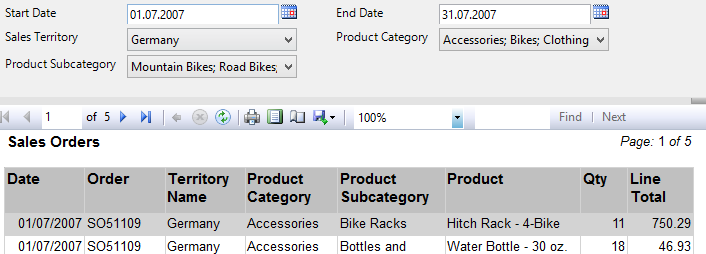




* Implement a method of displaying the report in the following manner:

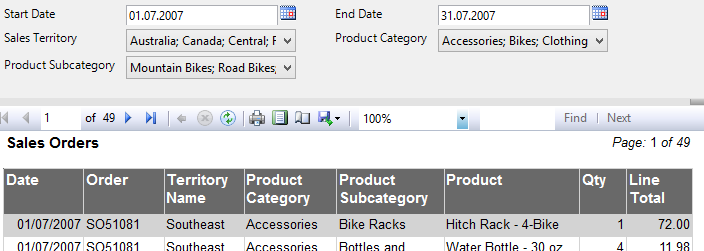
     - If there’s only one “Germany” value selected in the “Sales Territory”, then:

          - The colour of the table header must be “Silver”; the color of its font must be “Black”



     - In other situations:

          - The colour of the table header must be “DimGray”; the color of its font must be “White”

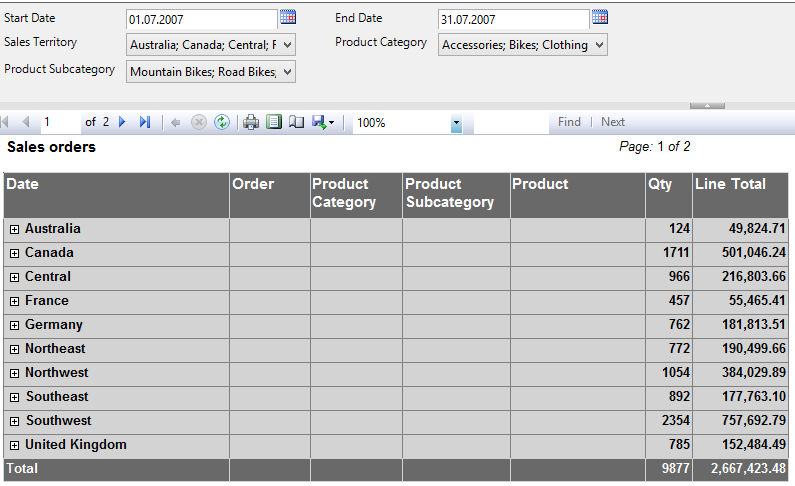


You can use other colors

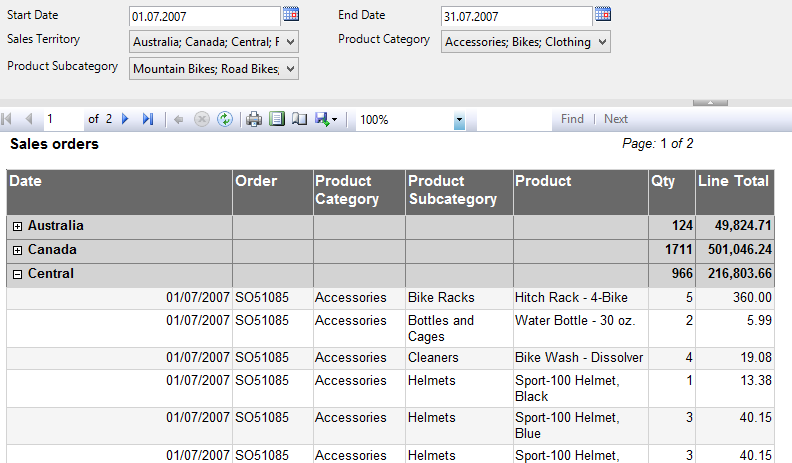
* 1 Chapter 10 Lesson 3
* 2 Chapter 3
* Learn about groups and data sorting
* Learn how to filter data by parameters and expressions
* Learn how to use aggregations

SQL Server Data Tools (SSDT)

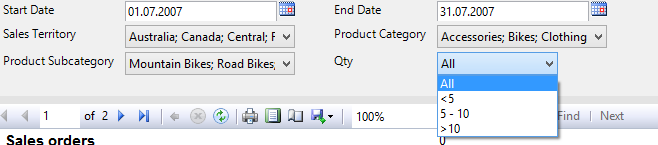
* + Create a copy of the “Sales Orders Lesson 3.rdl” report and name it “Sales Orders Lesson 4”
  + Implement grouping and sorting by “Sales Territory” and add the results
  + Add the “English Speaking” item to the “Sales Territory” parameter in order that only English speaking countries will be displayed



* + Implement drilldown (hide details and display them by clicking on “+”)



* Add the “Qty” filter. Values - “All”, “< 5”, “5 – 10”, “> 10”. Implement a filter for the table



[https://msdn.microsoft.com/en-us/library/dd207141(v=sql.110).aspx](https://msdn.microsoft.com/en-us/library/dd207141%28v=sql.110%29.aspx)

<https://msdn.microsoft.com/en-us/library/dd207131.aspx>

Learn about decomposition of reports and reusable parts

* Create a copy of the “Sales Orders Lesson 4.rdl” and rename it as “Sales Orders Lesson 5”
* Implement the report with the following elements:

    - A table with the columns Date, TerritoryName, ProductCategory, ProductSubCategory and Total. The date must be in the following format – “year, month” (e.g. 2010, April); group by Date and TerritoryName.

    - If Total >= 120, then its value must be displayed as a hyperlink. By clicking on it, there will be a Drillthrough on the report that displays the appropriate Order, Product, Qty and LineTotal. The report must display a line like “Detailed Clothing (Socks) Sales Report for France in April, 2007” in the header. If Total < 120, the cell must be displayed as plain text.

    - All the parameters and its panel must be hidden.

5 Chapter 8

Learn about charts, sparklines, gauges, indicators

Implement the “Sales Orders Lesson 6.rdl” report with the following elements:

Filter: Sales Territory

* Table

     - Product Category

     - Sparklines; must display the sum of “LineTotal” by years

          - When you point the mouse on the “LineTotal” value, a hint must appear and be displayed in the following manner: “day: sum LineTotal”

          - Implement “Drillthrough”: when clicking on the graph, it must redirect you to the “Sales Orders” report. The year data must be displayed as parameters (Start Date – beginning of the year, End Date – end of the year, Sales Territory, Product Category)

     - The last value of LineTotal

     - Status indicator for LineTotal (determine Thresholds)

* The “Product Category” diagram with labels indicating the percentage of sales for each category
* The “Total Sales by Years” diagram that displays the total number of sales for the “Bikes” product category and other categories united under the “Other” group.
* Implement “Custom” colors in diagrams
* Copy the first report, then add any chart before the table which can give an impression about what products generate the biggest profit. Make sure that the report retains its informative value after being printed in black and white (hint: often TOP N in charts have to be limited in order for them to be readable).
* <https://msdn.microsoft.com/en-us/library/ms156028.aspx>
* <http://msdn.microsoft.com/en-us/library/ms154682.aspx>
* <http://msdn.microsoft.com/en-us/library/ms156281.aspx>

Learn how to write functions inside a report

* Create a copy of the “Sales Orders Lesson 3\_1” and name it “Sales Orders Lesson 6”. Hide every parameter from a user except Start Date, End Date and Product Category.
* Create the following procedure in SSMS:

if (object\_id('ssrsTask\_bigTricky', 'P') is not null)

     drop procedure ssrsTask\_big

go

create procedure ssrsTask\_bigTricky

     @ProductSubCategory nvarchar(100) = null,

     @StartDate date = null,

     @EndDate date = null

as

create table #t(i int)

insert into #t (i) values (1),(2),(3),(4),(5),(6),(7),(8),(9),(10)

SELECT dateadd(year, 8, dateadd(dd, #t.i, soh.OrderDate)) AS DATE

     ,soh.SalesOrderNumber + '0' + ltrim(str(#t.i)) AS [Order]

     ,st.Name AS TerritoryName

     ,ppc.NAME AS ProductCategory

     ,pps.NAME AS ProductSubcategory

     ,pp.NAME AS Product ,sd.OrderQty AS Qty

     ,sd.LineTotal AS LineTotal

FROM Sales.SalesPerson AS sp

INNER JOIN Sales.SalesOrderHeader AS soh ON sp.BusinessEntityID = soh.SalesPersonID

INNER JOIN Sales.SalesOrderDetail AS sd ON sd.SalesOrderID = soh.SalesOrderID

INNER JOIN Production.Product AS pp ON sd.ProductID = pp.ProductID

INNER JOIN Production.ProductSubcategory AS pps ON pp.ProductSubcategoryID = pps.ProductSubcategoryID

INNER JOIN Production.ProductCategory AS ppc ON ppc.ProductCategoryID = pps.ProductCategoryID

INNER JOIN Sales.SalesTerritory AS st ON soh.TerritoryID = st.TerritoryID

cross join #t

where (@ProductSubCategory is null or pps.Name = @ProductSubCategory)

and (dateadd(year, 8, dateadd(dd, #t.i, soh.OrderDate)) between isnull(@StartDate, '2000-01-01') and isnull(@EndDate, getdate()))

drop table #t

go

* Edit the dataset by changing the procedure to ssrsTask\_bigTricky (don’t forget to create Refresh Fields)
* Add a “Data Range” parameter that can have one of the following values: Today, Yesterday, Last Week, Last Month, This Year, Date Interval. The default value must be “Last Week”. The logic must be as follows: If “Data Interval” is selected, then send the values of the “Start Date” and “End Date” parameters to the procedure. If other is selected, then send the appropriate dates to the procedure and ignore “Start Date” and “End Date”. Set the beginning of the week in “Control Panel”. You can also open RDL in text mode and write <CODE> inside a tag. Don’t forget to change “>” to “&gt” etc.
* Display a table of used parameters in the header. If a parameter hasn’t been used in the execution of the report, then it must not be displayed
* Copy the report to “Sales Orders Lesson 6\_1”. Change it in the following manner: don’t send any parameters to Dataset, filter in the table (Properties -> Filters). Compare performance.

[https://msdn.microsoft.com/en-us/library/dd239307(v=sql.110).aspx](https://msdn.microsoft.com/en-us/library/dd239307%28v=sql.110%29.aspx)

End users often convert reports to PDF for archives or for emailing them, and to CSV and Excel in order to import them to their CRM etc. or for further analysis and reporting to Excel. You need to learn how to present them such an opportunity.

* Copy the “Sales Orders Lesson 6” report and name it “Sales Orders Lesson 7”
* Instead of the “Date” column make two separate columns: “Year” and “Month” (use Expression)
* Upload the report to SSRS. Execute the report and export the results to CSV. Open the CSV.
* Change the report so that the columns in CSV would bear the same names as ones in the report
* In the “Textbox Order” properties write “DataElementOutput=NoOutput”. Select the “Territory Name” column, Column Visibility -> Hide.
* Execute the report. Export it in CSV, Excel, and PDF. Compare the columns in UI, CSV, Excel and PDF.
* Look though the PDF file. Does it look correct? Offer your variants of solving the problem for printable and non-printable reports.

<http://www.databasejournal.com/features/mssql/creating-a-custom-report-template-in-ssrs.html>

Reduce the efforts required to create a set of similar reports.

* Copy the “Sales Orders Lesson 2” report and name it “Sales Orders Lesson 8 Template”
* Delete information about the procedure and fields from Dataset
* Delete columns related to Dataset from the table
* Create a template from it that is available for selection during the creation of a new report.

What questions must you answer before you start developing reports? Write at least 5 of them.

1. The client requests one table to be able to receive data from different databases. How will you do it?
2. The client requests one table to be able to receive data from different databases so that the names of servers and databases may change during the usage. How will you do it?
3. There are several dozen frequently used reports. You need to create new reports and periodically refine older ones. The client proposes to use SQL queries directly from the report, speaking in favor of the existing practice and organizational challenges of creating stored procedure. What will you say to them?
4. What happens if you start a report where the parameters have default values? Please write about one scenario where this is helpful and another one where it’s not.
5. The client requests some parameters to be optional. How should you use a procedure in this case?
6. The client request good looking and creative reports that their customers will like. What can you offer them? What clarifying questions you need to ask the client?
7. A report contains many columns and does not fit in the width of the page or the screen. What solution can you offer the client?
8. The client liked a pie chart in the test database that shows quarterly sales of 12 products over one and a half years. The client’s database contains data on the customers they gathered for at least 8 years for 10 000 products. How must you implement the chart for it not to disappoint the client with real data?
9. What controls mustn’t be used if a report is to be sent to print or PDF forms?
10. The client has a chain of stores in USA, Canada, Germany and other countries. Business users view reports in regional offices. The main base has the “SaleDate DateTime” field. How to display the date according to the following formats?

     a. In the US format (mm/dd/yyyy) for everyone

     b. In a regional format for each office.

1. The client views reports from the application using the “ReportViewer” control (a standard control). The report that the client requested have, among other things, such parameters as “StartDate” and “EndDate”. The client’s database is rather large, there is a lot of historical data, the report has an average level of difficulty. Which answers must you obtain in order to write the SQL part?
2. Major customers of our client complain about the slow performance (or timeouts) of a legacy report. At a first glance, the report looks like a summary report and is half a page long. It is said that the report has been running for a long time, but the problems started occurring about a year ago. Since then they have had to periodically increase the timeouts. What would you do?
3. The client views reports from the application using the “ReportViewer” control (a standard control). The report is presented in the form of operations statement for a certain period and can consist of a thousand pages. The report’s footer has “TotalPages”, and “Totals” and “Subtotals” are calculated in the report itself. The client requests to add a “DocumentMap” for a more convenient navigation inside the large report. How will you do it?
4. The client has 200 reports. They request to add a footer (with a number of the current page) to every report. How will you do it?

a. <http://bhushan.extreme-advice.com/create-matrix-report-in-ssrs/>

b. <https://www.simple-talk.com/sql/reporting-services/advanced-matrix-reporting-techniques/>

Learn about Matrix

* Copy “Sales Orders Lesson 4” to “Sales Orders Lesson 9”
* Change the report so that instead of one “Line Total” column there would be “Total” columns with years: Line Total 2005, Line Total 2006 etc. The amount of these columns must be adequate to a number of years in the database.

<http://www.codeproject.com/Articles/195017/SSRS-Series-Part-II-Working-with-Subreports-DrillD#4>

Learn about subreports.

* Create a new report based on “Sales Orders Lesson 8 Template” and name it “Sales Orders Lesson 10”
* Delete the table in it, add a Subreport, insert “Sales Orders Lesson 3” and implement the parameters from “Sales Orders Lesson 10”.