Project 2 - AdventureWorks Report

**November 24, 2018**

**V 1.0**

*Prepared by*

Liliek Sidharta

**A00821393**

# Table of Contents

[Table of Contents 2](#_Toc530847539)

[Executive Summary 1](#_Toc530847540)

[Introduction 2](#_Toc530847541)

[Purpose 2](#_Toc530847542)

[Criteria for Completion 2](#_Toc530847543)

[Audience 2](#_Toc530847544)

[Background 2](#_Toc530847545)

[Requirements/Pre-requisites 3](#_Toc530847546)

[Environment 5](#_Toc530847547)

[System Configuration 5](#_Toc530847548)

[Software used 5](#_Toc530847549)

[Accounts 5](#_Toc530847550)

[Installation/Configuration Steps 6](#_Toc530847551)

[Authoring Reports 6](#_Toc530847552)

[Report 1 - AW Template 6](#_Toc530847553)

[Report 2 - Detail Sales Order 8](#_Toc530847554)

[Report 3 - Territory Sales 17](#_Toc530847555)

[Report 4 – Employee Summary Sales 24](#_Toc530847556)

[Part 2 Managing Reports 44](#_Toc530847557)

[Additional Notes 55](#_Toc530847558)

[References 56](#_Toc530847559)

# Executive Summary

The Reporting Service Project 2 is developed based on AdventureWorks database. There is one report template, detail order report and drill through reports which contain charts to show the sales performance. This project also requires reporting service administration of two levels of role and users with different permissions and rights.

The reports have been built in Visual Studio 2017 Enterprise and SQL Server Data Tools environment.

# Introduction

## Purpose

This document is submitted together with SSRS\_Project2 Visual Studio Reporting Service Project, which is a mandatory final project of part-time course COMP 4679 Business Intelligence with SSRS, at BCIT, Fall 2018 (CRN 35618).

## Criteria for Completion

The criteria of project completion are successfully authoring and deploying the following reports:

* *Report 1 – AW Template*
* *Report 2 – Detail Sales Order*
* *Report 3.- Territory Sales*
* *Report 4 – Employee Summary Sales*

and administering report server through Report Manager to do the following tasks:

* *Create local accounts for John Smith, Peter Doe, Oscar Larsen, Brian Thompson*
* *Create local security groups for AWSales, AWHumanRes, AWAdmin*
* *Create three folders, Datasources, Sales Reports and Employee Reports, in Report Manager*
* *Assign permissions to AWAdmin to have a Site Administrator role*
* *Assign rights to AWHumanRes and AWSales to have Site Browser*
* *Create link reports to Detail Sales Order, Territory Sales and Employee Summary Sales Reports*
* *Export all reports to PDF output*
* *Enable Reports History*

## Audience

The students, the teacher and everyone who is interested.

## Background

This document is submitted as the course requirement and based on the requirements given by the teacher.

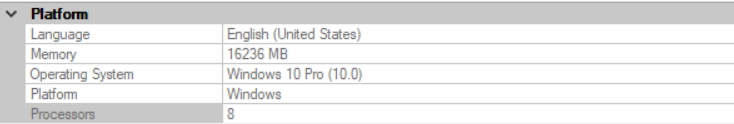
# Requirements/Pre-requisites

* All reports are based on the sample AdventureWorks database
* ***Report 1 – AW Template***
  + Create a template for all reports
  + It should have the AdventureWorks Cycles logo in the header.
  + In to footer it should present
    - report name
    - page number
    - total number of pages
    - date and time when report is generated
* ***Report 2 – Detail Sales Order***
  + This is the detail of an individual Adventure Works Cycles order
  + This report should be accessed as a drillthrough report from the Territory Sales drilldown report.
  + It contains
    - lists,
    - tables,
    - parameters, and
    - expressions, as presented in SalesOrder.Pdf file
* ***Report 3.- Territory Sales***
  + This report drills down through salesperson and order number with drillthrough to individual orders.
  + This report contains table data region, providing drilldown from summary data into detail data by showing and hiding rows. This report **should open Detailed Sales Order report as drillthrough link**.
  + Its final layout should be as TerritorySales.Pdf file

* ***Report 4 – Employee Summary Sales*** 
  + This report includes:
    - Sales Comparison chart
    - Current Month Sales Comparison charts
    - Current Month Order Summary table
  + Contains drillthrough to individual orders, which contains multiple datasets, charts, tables, drillthrough, and dynamic parameters.
  + Order number should be clickable, **and it should open Detail Sales Order report**.
  + Its final layput should follow EmployeeSalesSummary.Pdf
* **Part 2 –Managing Reports**
  + Create local login accounts:
    - John Smith
    - Peter Doe
    - Oscar Larsen
    - Brian Thompson
* Create following local security groups and add users to each of them:
  + AWSales - John Smith and Peter Doe - Browser in Sales Reports directory
  + AWHumanRes - Oscar Larsen – Browser in Employee Reports directory
  + AWAdmin - Brian Thompson - Site Administrator
* Create required data sources in Datasources directory
* Upload all reports from Part 1 to proper directories in Report Manager
* Create Linked Reports as required
* Export all reports to Pdf files and include these files in project documentation
* Enable Report History for all deployed reports

# Environment

## System Configuration



* Microsoft SQL Server 2017 Developer Edition (en\_sql\_server\_2017\_developer\_x64\_dvd\_11296168)
* Microsoft SQL Server Reporting Service for MS SQL Server 2017 (SQLServerReportingServices)
* Microsoft SQL Server Data Tools (SSDT-Setup-ENU) ver 15.8.1
* Microsoft Visual Studio 2017 Enterprise Edition (vs\_enterprise\_\_1921682665.1536936363)

## Software used

Visual Studio 2017 and SSDT

## Accounts

* Windows Authentication
* SQL Server Authentication, System Administrator.

# Installation/Configuration Steps

| **Step** | **Server or Instance** | **Description** | | **Screenshot/Notes** | |
| --- | --- | --- | --- | --- | --- |
| Authoring Reports | | | | | | |
| Report 1 - AW Template | | | | | | |
|  |  | Copy the logo images into the solution folder | | 1. Copy AWLogo and AWSmall.jpg into folder AWTemplate.jpg folder | |
|  |  | Create report | | 1. Find Solution Explorer > SSRS\_Project2 > Reports 2. Right click and select **Add…> New Item** 3. Modify Add NewItem – SSRS\_Project2 dialog box    * Select **Report**    * Name: **AWTemplate** 4. Click **Add** | |
|  |  | Add Page Header and Page Footer | | 1. Find menu Report > **Add Page Header** to add header area 2. Repeat the above step for **Add Page Footer** | |
|  |  | Insert logo into page header | | 1. Drag Image component from Toolbox into Page Header area 2. Modify Image Properties    1. Name: Logo    2. ToolTip: Adventure Works Logo    3. Select the image source: Emnedded    4. Use this image: AWLogo    5. Click OK 3. Add textbox for title | |
|  |  | Create layout for page footer | | 1. Add 3 textboxes into page footer:    1. Textbox 1: [&ReportName]    2. Textbox 2: "Page " & Globals!PageNumber & " of " & Globals!TotalPages    3. Textbox 3: [&ExecutionTime] | |
|  |  | Report layout | | Territ | |
|  |  | Copy the AWTemplate report to common folder | | To be able to be used later as a report template, the report should be stored in a common folder:  C:\Program Files (x86)\Microsoft Visual Studio\2017\Enterprise\Common7\IDE\CommonExtensions\Microsoft\SSRS\ProjectItems\ReportProject | |
| Report 2 - Detail Sales Order  * Detail of an individual Adventure Works Cycles order. * This report will be accessed as a drillthrough report from Territory Sales report. * It contains list, tables, parameters and expression as presented in SalesOrder.pdf file | | | | | | |
|  |  | Create Shared Data Sources | | 1. Find Solution Explorer > SSRS\_Project2 > Shared Data Sources 2. Right click and select **Add New Data Source**      1. Modify the Shared Data Source Properties dialog box:    * Name: AdventureWorks    * Click **Edit…**      1. Modify Connection Properties dialog box:    * Data source: Microsoft SQL Server (SqlClient)    * Server name: DESKTOP-KOPJRLK    * Authentication: Windows Authentication    * Select or enter a database name: AdventureWorks 2. Click **Test Connection** button 3. Click **OK**      1. Click **OK** again | |
|  |  | Create report | | 1. Find Solution Explorer > SSRS\_Project2 > Reports 2. Right click and select **Add…> New Item** 3. Modify Add NewItem – SSRS\_Project2 dialog box    * Select **AWTemplate**    * Name: **DetailSalesOrder** 4. Click **Add** | |
|  |  | Create Datasource AdventureWorks | | 1. Find Report Data window > Data Sources 2. Right click and select **Add Data Source…**      1. Modify Data Source Properties dialog box:  * Name: AdventureWorks * Use shared data source reference: find **AdventureWorks** in the list * Click OK | |
|  |  | Create Dataset **Territory** | | As DetailSalesOrder report needs an input parameter that should be provided in a list, so that we need to create a dataset Territory before creating dataset for DetailSalesOrder.   1. Find Report Data window > Data Sources > AdventureWorks 2. Right click and select **Add Dataset…**      1. Modify Dataset Properties Dialog Box  * Name: **Territory** * Data source**: AdventureWorks** * Query:   SELECT  Sales.SalesTerritory.Name  FROM  Sales.SalesTerritory | |
|  |  | Create Dataset **SalesOrder** | | 1. Find Report Data window > Data Sources > AdventureWorks 2. Right click and select **Add Dataset…** 3. Modify Dataset Properties Dialog Box  * Name: **SalesOrder** * Data source**: AdventureWorks** * Query:   SELECT c.FirstName + ' ' + c.LastName as [Contact]  ,c.Phone  ,h.OrderDate AS [Order Date]  ,SalesOrderNumber  ,cSalesPerson.FirstName AS [SalesPerson FirstName]  ,cSalesPerson.LastName AS [SalesPerson LastName]  ,e.Title AS [SalesPerson Title]  ,st.TerritoryID  ,st.Name AS [TerritoryName]  ,s.Name AS [Bill To / Ship To]  ,billingAdr.AddressLine1 AS [Bill Addr1]  ,billingAdr.AddressLine2 AS [Bill Addr2]  ,billingAdr.City AS [Bill City]  ,billingProv.Name AS [Bill Prov]  ,billingAdr.PostalCode AS [Bill PostCode]  ,billingCountry.Name AS [Bill Ctr]  ,shipmentAdr.AddressLine1 AS [Ship Addr1]  ,shipmentAdr.AddressLine2 AS [Ship Addr2]  ,shipmentAdr.City AS [Ship City]  ,shipmentAdr.PostalCode AS [Ship PostalCode]  ,shipmentProv.Name AS [Ship Prov]  ,shipmentCountry.Name AS [Ship Ctr]  ,h.PurchaseOrderNumber  ,ps.Name AS [Shipment Method]  ,[CarrierTrackingNumber]  ,[OrderQty]  ,p.ProductNumber AS [Item Number]  ,p.Name AS [Description]  ,[UnitPrice]  ,[UnitPriceDiscount]  ,[LineTotal]  FROM [AdventureWorks].[Sales].SalesOrderHeader h  INNER JOIN AdventureWorks.Sales.SalesOrderDetail d  ON h.SalesOrderID = d.SalesOrderID  INNER JOIN AdventureWorks.Production.Product p  ON p.ProductID = d.ProductID  INNER JOIN AdventureWorks.Person.Contact c  ON c.ContactID = h.ContactID  INNER JOIN AdventureWorks.Sales.Store s  ON s.CustomerID = h.CustomerID  INNER JOIN AdventureWorks.Purchasing.ShipMethod ps  ON ps.ShipMethodID = h.ShipMethodID  INNER JOIN HumanResources.Employee e  ON h.SalesPersonID = e.EmployeeID  INNER JOIN Person.Contact cSalesPerson  ON cSalesPerson.ContactID = e.ContactID  INNER JOIN Sales.SalesPerson sp  ON h.SalesPersonID = sp.SalesPersonID  INNER JOIN Sales.SalesTerritory st  ON st.TerritoryID = sp.TerritoryID  INNER JOIN Person.Address billingAdr  ON h.BillToAddressID = billingAdr.AddressID  INNER JOIN Person.StateProvince billingProv  ON billingAdr.StateProvinceID = billingProv.StateProvinceID  INNER JOIN Person.CountryRegion billingCountry  ON billingProv.CountryRegionCode = billingCountry.CountryRegionCode  INNER JOIN Person.Address shipmentAdr  ON h.ShipToAddressID = shipmentAdr.AddressID  INNER JOIN Person.StateProvince shipmentProv  ON shipmentAdr.StateProvinceID = shipmentProv.StateProvinceID  INNER JOIN Person.CountryRegion shipmentCountry  ON shipmentProv.CountryRegionCode = shipmentCountry.CountryRegionCode  WHERE SalesOrderNumber = @SalesOrderNumber     1. Click OK   It automatically creates parameter SalesOrderNumber, which is a string data type. Since SalesOrderNumber will be input by user, we don’t need to do any modification on it. | |
|  |  | Create Report Layout:  Modify header | | 1. Update “Adventure Works“ with “Sales Order” 2. Add text box for order number: "Order # " + Parameters!SalesOrderNumber.Value | |
|  |  | Create Report Layout:  Create table **Orderdetail** | | 1. Insert Table 2. Modify List’s properties    1. Name: **OrderDetail**    2. DataSetName: **SalesOrder**  |  |  | | --- | --- | | **Columns** | **Fields** | | Line | =RowNumber("SalesOrder") | | Qty | =Fields!OrderQty.Value | | Item Number | =Fields!Item\_Number.Value | | Description | =Fields!Description.Value | | Tracking # | =Fields!CarrierTrackingNumber.Value | | Unit Price | =Fields!UnitPrice.Value | | Subtotal | =Fields!UnitPrice.Value\*Fields!OrderQty.Value | | Discount | =Fields!UnitPriceDiscount.Value | | Item Total | =Fields!LineTotal.Value | | Total Discount | =Sum(Fields!UnitPriceDiscount.Value) | | Total | =Sum(Fields!LineTotal.Value) | | |
|  |  | Create rectangle and textboxes | | 1. Rectangle Template 2. Insert Textbox for text “Bill To” 3. Insert Textbox for Billing name and address  |  |  | | --- | --- | |  | **Fields** | | Name | Ship\_BillTo | | Address | BillAddrLine1 | |  | BillCity | |  | BillProvCode | |  | BillCountry |  1. Insert Textbox for Shipping name and address  |  |  | | --- | --- | |  | **Fields** | | Name | Ship\_BillTo | | Address | ShipAddrLine1 | |  | ShipCity | |  | ShipProvCode | |  | ShipCountry |  1. Insert Textbox for Contact person  |  |  | | --- | --- | |  | **Fields** | | Name | First(Fields!Contact.Value, "SalesOrder") | | Phone | "Ph: " & First(Fields!Phone.Value, "SalesOrder") |  1. Insert text box for Date generating report   =Globals!ExecutionTime   1. Insert text box for Order Date   =First(Fields!Order\_Date.Value, "SalesOrder")   1. Insert text box for Sales Person   =First(Fields!SalesPerson\_Name.Value, "SalesOrder") + ", " + First(Fields!SalesPerson\_Title.Value, "SalesOrder")   1. Insert text box for Purchase Order   =First(Fields!PurchaseOrderNumber.Value, "SalesOrder")   1. Insert text box for shipment method   =First(Fields!Shipment\_Method.Value, "SalesOrder") | |
|  |  | Layout Result | |  | |
|  |  | Report Preview | |  | |
| Report 3 - Territory Sales  * Contains table data region, providing drilldown from summary data into detail data by showing and hiding rows. It drilldown through Sales Person and Order Number. * This report should open Detail Sales Order Report as a drillthrough link | | | | | | |
|  |  | Create report TerritorySales | | 1. Find Solution Explorer > SSRS\_Project2 > Reports 2. Right click and select **Add…> New Item** 3. Modify Add NewItem – SSRS\_Project2 dialog box    * Select **AWTemplate**    * Name: **TerritorySales**   Click **Add** | |
|  |  | Modify the page header | | Update the title to **Territory Sales** | |
|  |  | Create Dataset **Territory** | | As DetailSalesOrder report needs an input parameter that should be provided in a list, so that we need to create a dataset Territory before creating dataset for DetailSalesOrder.   1. Find Report Data window > Data Sources > AdventureWorks 2. Right click and select **Add Dataset…**      1. Modify Dataset Properties Dialog Box  * Name: **Territory** * Data source**: AdventureWorks** * Query:   SELECT  Sales.SalesTerritory.Name  FROM  Sales.SalesTerritory | |
|  |  | Create Dataset **SalesSummary** | | 1. Find Report Data window > Data Sources > AdventureWorks 2. Right click and select **Add Dataset…** 3. Modify Dataset Properties Dialog Box  * Name: **SalesSummary** * Data source**: AdventureWorks** * Query:   SELECT  SalesOrderNumber  ,TotalDue  ,cSalesPerson.FirstName + ' ' + cSalesPerson.LastName AS [SalesPerson Name]  ,st.Name AS [TerritoryName]  FROM [AdventureWorks].[Sales].SalesOrderHeader h  INNER JOIN AdventureWorks.Person.Contact c  ON c.ContactID = h.ContactID  INNER JOIN HumanResources.Employee e  ON h.SalesPersonID = e.EmployeeID  INNER JOIN Person.Contact cSalesPerson  ON cSalesPerson.ContactID = e.ContactID  INNER JOIN Sales.SalesTerritory st  ON st.TerritoryID = h.TerritoryID  WHERE st.Name = @Territory;    It creates parameter @Territory automatically. | |
|  |  | Modify Parameters Territory | | 1. Parameters   Parameter Name: @Territory  Parameter value: @Territory     1. Click OK 2. Find Report Data > Parameters > Territory 3. Modify Report Parameter Properties > Available Values tab  * Select from one of the following options: Get values from a query * Dataset: **Territory** * Value field: **Name** * Label field: **Name**     Click OK | |
|  |  | Create Report Layout | | 1. Insert Table 2. Modify table’s properties    1. DataSetName: **SalesSummary**  |  |  | | --- | --- | | **Columns** | **Fields** | | Total Sales | = Fields!TotalDue.Value | | Order Number | = Fields!SalesOrderNumber.Value |  1. In Row Groups pane, find **Details**, and click the right dropdown. 2. Select **Add Group > Parent Group**      1. Modify Tablix Group    1. Group by: [SalesPerson\_Name]    2. Add group header      1. Click OK 2. At **Details**, click the right dropdown arrow. 3. Select Add Total > Before      1. In Row Groups pane, find **SalesPerson\_Name**, and click the right dropdown. 2. Select **Add Group > Parent Group** 3. Modify Tablix Group    1. Group by: [TerritoryName]    2. Add group header      1. At **SalesPerson\_Name**, click the right dropdown arrow. 2. Select Add Total > Before   Make sure that the rows, columns and totals follow the design.   1. At the SalesOrderNumber textbox, right click and select **Textbox Properties** 2. Select **Action** 3. Modify the Text Box Properties:    1. Enable as an action: **Go to report**    2. Specify a report: **DetailSalesOrder**    3. User these parameters to run the report, click **Add** button       1. Select **SalesOrderNumber** in Name       2. Select [**SalesOrderNumber**] in Value 4. Click OK 5. In the row of **SalesOrderNumber**, modify textbox properties:    1. BackgroundColor: IIf(RowNumber("SalesSummary") Mod 2, "#f7f9f9", "#ffffff") | |
|  |  | Layout Result | |  | |
|  |  | Report preview | | If we click the Sales Order Number, it will display the Detail Sales Order report. | |
| Report 4 – Employee Summary Sales | | | | | | |
|  |  | Create Report | | EmployeeSummarySales.rdl | |
|  |  | Add Data Sources in Report Data window | | AdventureWorks | |
|  |  | Add Dataset | | YearlySales  With Query String:  SELECT soh.OrderDate AS [OrderDate],  MONTH(soh.OrderDate) AS [Month],  YEAR(soh.OrderDate) AS [Year],  c.FirstName + ' ' + c.LastName AS [Name],  sod.LineTotal AS [LineTotal]  FROM [AdventureWorks].[Sales].SalesOrderHeader soh  INNER JOIN [Sales].[SalesPerson] sp  ON sp.SalesPersonID = soh.SalesPersonID  INNER JOIN [HumanResources].[Employee] e  ON e.EmployeeID = sp.SalesPersonID  INNER JOIN [Person].[Contact] c  ON c.ContactID = e.ContactID  INNER JOIN [Sales].[SalesOrderDetail] sod  ON sod.SalesOrderID = soh.SalesOrderID  WHERE YEAR(soh.OrderDate) = @year AND (c.FirstName + ' ' + c.LastName) =@name | |
|  |  | Add Dataset Year | | This dataset provides a selection list for parameter @year.  The query string is:  SELECT DISTINCT YEAR(OrderDate) AS [Year]  FROM Sales.SalesOrderHeader  ORDER BY YEAR(OrderDate) | |
|  |  | Add Dataset SalesPersonName | | This dataset provides a selection list for parameter @name.  The query is:  SELECT DISTINCT c.FirstName + ' ' + c.LastName AS [Name]  FROM [AdventureWorks].[Sales].SalesOrderHeader soh  INNER JOIN [Sales].[SalesPerson] sp  ON sp.SalesPersonID = soh.SalesPersonID  INNER JOIN [HumanResources].[Employee] e  ON e.EmployeeID = sp.SalesPersonID  INNER JOIN [Person].[Contact] c  ON c.ContactID = e.ContactID | |
|  |  | Set Report Parameter Properties for @year and @name | | For @year:        For @name: | |
|  |  | Set Chart Data | | **Month property**:  Label =MonthName(Fields!Month.Value, False)  **LineTotal property**:  Color: Green  BorderWidth: 2pt;  **Chart Area Property**:  BackgroundColor: Silver  BorderStyle: solid  **Chart Gridlines Property**:  LineStyle: solid  **Chart Title Property**:  Color: Black  Font: Arial, 14 pt, default, bold  Position: TopCenter  Delete Legend.  Vertical Axis Properties:    Horizontal Axis Properties:    Line Series Properties:    Legend = =Parameters!year.Value + "-Sales"  Chart Legend Properties:  BorderStyle: Solid | |
|  |  | Chart Preview | |  | |
| Monthly Sales Comparison | | | | | | |
|  |  | Create Dataset **Month** for month selection. It creates a list of 12 month’s names, regardless there is order or not. | | | WITH months(MonthNumber) AS  (  SELECT 0  UNION ALL  SELECT MonthNumber+1  FROM months  WHERE MonthNumber < 11  )  select DATENAME(MONTH, DATEADD(MONTH, MonthNumber, '2000-01-01')) AS [Month]  from months; |
|  |  | Create Dataset **MonthlySales** | | | SELECT DATENAME(MONTH, OrderDate) AS [Month],  YEAR(soh.OrderDate) AS [Year],  c.FirstName + ' ' + c.LastName AS [Name],  sod.LineTotal AS [LineTotal],  pc.Name AS [ProductCategory]  FROM [AdventureWorks].[Sales].SalesOrderHeader soh  INNER JOIN [Sales].[SalesPerson] sp  ON sp.SalesPersonID = soh.SalesPersonID  INNER JOIN [HumanResources].[Employee] e  ON e.EmployeeID = sp.SalesPersonID  INNER JOIN [Person].[Contact] c  ON c.ContactID = e.ContactID  INNER JOIN [Sales].[SalesOrderDetail] sod  ON sod.SalesOrderID = soh.SalesOrderID  INNER JOIN [Production].[Product] p  ON p.ProductID = sod.ProductID  INNER JOIN [Production].[ProductSubcategory] ps  ON ps.ProductSubcategoryID = p.ProductSubcategoryID  INNER JOIN [Production].[ProductCategory] pc  ON pc.ProductCategoryID = ps.ProductCategoryID WHERE YEAR(soh.OrderDate) = @year AND DATENAME(MONTH, OrderDate) = @month AND (c.FirstName + ' ' + c.LastName) = @name |
|  |  | Set Chart DatasetName Properties | | |  |
|  |  | Set Chart Properties | | | **Chart Series property**:  Color: DarkOliveGreen  Legend > Text: =Parameters!month.Value + " " + Parameters!year.Value + "-Sales"  Position: BottomCenter  Marker>BorderColor: Gray  Marker>BorderStyle: solid  Marker>BorderWidth: 1pt  Marker>Color: DimGray  **Chart Area Properties**:  BackgroundColor: FloralWhite  BorderColor: Gray  BorderStyle: solid  BorderWidth: 1pt  **Chart Gridlines Properties**:  LineColor: DarkKhaki  LineStyle: Default  LineWidth: 0.5pt  **Chart Legend Properties**:  Layout: =Parameters!month.Value + " " + Parameters!year.Value + "-Sales"  Position: BottomCenter  Font: Arial, 8pt, Default, Default  **Chart Axis Properties**:  LabelsFormat: '$'#,0.00;('$'#,0.00)  LabelsColor: DimGray  LabelsFont: Arial, 8pt, default, default  Margin: Enable  **Chart Title Properties**:  Caption: =Parameters!month.Value + " " + Parameters!year.Value + " Sales Comparison"  Font: Arial, 16pt, Default, Bold  Position: TopCenter  **Chart Properties:**  Size: 15.5cm, 8cm  BorderSTyle: Solid  BorderColor: LightGray |
|  |  | Chart Preview | | |  |
| **Order Summary** | | | | | | |
|  |  | Add table and set its dataname | DataSetName: OrderSummary  WITH cte\_OrderSummary AS  (  SELECT soh.SalesPersonID,  soh.SalesOrderNumber AS [OrderNumber],  soh.OrderDate,  pc.Name AS [ProductCategory],  SUM(sod.LineTotal) AS [Sales]  FROM [AdventureWorks].[Sales].SalesOrderHeader soh  INNER JOIN [Sales].[SalesOrderDetail] sod  ON sod.SalesOrderID = soh.SalesOrderID  INNER JOIN [Production].[Product] p  ON p.ProductID = sod.ProductID  INNER JOIN [Production].[ProductSubcategory] ps  ON ps.ProductSubcategoryID = p.ProductSubcategoryID  INNER JOIN [Production].[ProductCategory] pc  ON pc.ProductCategoryID = ps.ProductCategoryID  GROUP BY  soh.SalesOrderNumber,  soh.OrderDate,  pc.Name,  soh.SalesPersonID  )  SELECT c.FirstName + ' ' + c.LastName AS [Name],  DATENAME(MONTH, cte.OrderDate) AS [Month],  YEAR(cte.OrderDate) AS [Year],  cte.OrderNumber,  cte.ProductCategory,  cte.Sales  FROM [cte\_OrderSummary] cte  INNER JOIN [Sales].[SalesPerson] sp  ON sp.SalesPersonID = cte.SalesPersonID  INNER JOIN [HumanResources].[Employee] e  ON e.EmployeeID = sp.SalesPersonID  INNER JOIN [Person].[Contact] c  ON c.ContactID = e.ContactID WHERE YEAR(cte.OrderDate) = @year AND DATENAME(MONTH, cte.OrderDate) = @month AND (c.FirstName + ' ' + c.LastName) =@name | | |
|  |  | Add a Table to the report’s canvas | Set DataSetName to OrderSummary.  Set 3 columns, each for Sales, Product Category and Order Number.  In the Row Groups:  On Details, Add Total > After  On Details, Add Group > Parent Group > Group by [OrderNumber]  On OrderNumber, Add Total > After | | |
|  |  | Table Preview |  | | |
|  |  | Add Link to Order Number | 1. At the SalesOrderNumber textbox, right click and select **Textbox Properties** 2. Select **Action** 3. Modify the Text Box Properties:    1. Enable as an action: **Go to report**    2. Specify a report: **DetailSalesOrder**    3. User these parameters to run the report, click **Add** button       1. Select **SalesOrderNumber** in Name       2. Select [**SalesOrderNumber**] in Value 4. Click OK | | |
|  |  | Report Layout |  | | |
|  |  | Report Preview | If the Order Number link such as SO48732 is clicked, the browser will show us the Detail Order of that order number: | | |
| Part 2 Managing Reports | | | | | | |
| 1. **1** |  | Create local login accounts | | 1. Open Control Panel\All Control Panel Items\Administrative Tools 2. Open **Computer Management** 3. Find **Local Users and Groups** under Computer management (local)      1. On Users, right click and select **New User…** to open New User dialog box 2. Modify Dialog Box:    1. User name: JohnSmith    2. Full name: John Smith    3. Password: Win2k16    4. Confirm password: Win2k16 3. Click **Create** 4. Repeat step 4 to 6 for Peter Dow, Oscar Larson and Brian Thompson | |
| 1. **2** |  | Create local security groups | | 1. On Groups, right click and select **New Group…** to open New Group dialog box 2. Modify Dialog Box:   Group name: AWSales     1. Click button **Add** to open **Select Users** dialog box 2. In the text area under Enter the object names to select, type JohnSmith; PeterDoe;      1. Click **Check** Names button      1. Click **OK** 2. Repeat step 1-6 for AWHumanRes and AWAdmin; and for users Oscar Larson and Brian Thompson. | |
|  |  | Open **SQL Server Reporting Services Server** | | 1. Open Report Manager in this URL: <http://desktop-kopjrlk/Reports/>      1. Click the **Settings** icon, and select **Site Settings** to open Site settings page     The most important part of Site Settings is **Security** tab.    At the moment this capture is captured, the System Administrator of the Reporting Server is default **Administrators**. | |
|  |  | Create folders Datasources, Sales Reports and Employee Reports | | 1. In the toolbar, find **+** icon, click on it, and select **Folder** to open Create a new folder in Home dialog box.      1. In **Name** type **Datasources**, and click Create      1. Repeat step 1 and 2 to create Sales Reports and Employee Reports. | |
|  |  | Assign **Site Administrator permission** for **AWAdmin** group | | 1. Click the **Settings** icon, and select **Site Settings** 2. Click **Security** 3. Click **Add group or user** 4. Add Group or user: **AWAdmin** 5. Check System Administrator      1. Click OK | |
|  |  | Grant **Content Manager** rights to AWAdmins group in all **Employee Reports, Sales Reports** | | 1. Find **Employee Reports** folder, click **...**, and select **Manage** to open Manage Employee Reports page.      1. Click **Security** 2. Click **Customize Security** 3. Click **ok** to confirm modification on security settings. 4. Click **Add group or user** 5. In Group or user: **AWAdmin** 6. Check **Content Manager**      1. Click **OK**      1. Repeat steps 1 to 8 to add **AWAdmin** to **Sales Reports** folder. | |
|  |  | Grant **Browser** rights to **AWSales and AWHumanRes groups** in Sales Reports folder | | 1. Find **Sales Reports** folder, click **...**, and select **Manage** to open Manage Employee Reports page. 2. Click **Security** 3. Click **Add group or user** 4. In Group or user type **AWSales** 5. Check **Browser**      1. Click **OK** 2. Add group **AWHumanRes** as Browser to Sales Reports as well.      1. Now Sales Reports has security profile as follows. | |
|  |  | Grant Browser rights to AWHumanRes group in Employee Reports folder | | 1. Find **Employee Reports** folder, click **...**, and select **Manage** to open Manage Employee Reports page. 2. Click **Security** 3. Click **Add group or user** 4. In Group or user type **AWHumanRes** 5. Check **Browser**      1. Click **OK** 2. Now Employee Reports has security profile as follows. | |
| 1. **9-10** |  | Modify Visual Studio project and deploy to **upload all reports** to **Sales Reports directory** and datasource to **Datasources directory** | | 1. At Solution Explorer > SSRS\_Project2 project, right click and select **Properties** 2. Modify several values in the dialog box SSRS\_Project2 Property Pages:    1. TargetDataSourceFolder: **Datasources**    2. TargetReportFolder: **Sales Reports**    3. TargetServerURL: [**http://desktop-kopjrlk/ReportServer**](http://desktop-kopjrlk/ReportServer) 3. Click **OK** 4. Deploy     Project deployment generate reports in **Sales Reports** folder:   * AWTemplate * DetailSalesOrder * EmployeeSummarySales * TerritorySales * AWSmall.jpg | |
|  |  | Create Linked Reports for EmployeeSummarySales report | | 1. Find **EmployeeSummarySales**, click **…** and select **Manage** 2. Click **Parameters** 3. Modify the page    1. **Check** Use default    2. Default value: **January** 4. Click **Apply**      1. Go back to Manage > Properties page 2. Find Create linked report and click it. 3. Provide a link name: **Employee Sales Report** 4. Click **Create** | |
|  |  | Create Linked Reports for Territory Sales | | 1. Find **Territory Sales**, click **…** and select **Manage** 2. Click **Parameters** 3. Modify the page    1. **Check** Use default    2. Default value: **Australia** 4. Click **Apply**      1. Go back to **Manages** page 2. Find Create linked report and click it. 3. Provide a link name: **Territory Sales Report** 4. Click **Create** | |
|  |  |  | | Linked Reports are:   * Employee Sales Report * Territory Sales Report * Detail Sales Report   Hide all reports, but the linked reports | |
|  |  | Export all reports to PDF files | | 1. Open report Detail Sales Report 2. Click icon **Floppy Disk** and select **PDF** | |
|  |  | Enable Report History for all deployed reports | | 1. In Report Manager, click **Site Settings** 2. Select **Retain all history snapshots** for I want to keep all report history. 3. Click **Apply** | |

Additional Notes

Attachments to this document:

* COMP4679\_Project2\_LiliekSidharta.sql
* COMP4679\_Project2\_LiliekSidharta.zip
* DetailSalesOrder.pdf
* EmployeeSalesSummary.pdf
* TerritorySales.pdf

References

|  |
| --- |
| **Reference Title** |
| **Brian Larson, MS SQL 2016 Reporting Service, Fifth Edition, McGrawHill** |