

OFFICIAL MICROSOFT LEARNING PRODUCT

10990B

Analyzing Data with SQL Server Reporting Services

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The names of manufacturers, products, or URLs are provided for informational purposes only and Microsoft makes no representations and warranties, either expressed, implied, or statutory, regarding these manufacturers or the use of the products with any Microsoft technologies. The inclusion of a manufacturer or product does not imply endorsement of Microsoft of the manufacturer or product. Links may be provided to third party sites. Such sites are not under the control of Microsoft and Microsoft is not responsible for the contents of any linked site or any link contained in a linked site, or any changes or updates to such sites. Microsoft is not responsible for webcasting or any other form of transmission received from any linked site. Microsoft is providing these links to you only as a convenience, and the inclusion of any link does not imply endorsement of Microsoft of the site or the products contained therein.

© 2016 Microsoft Corporation. All rights reserved.

Microsoft and the trademarks listed at <http://www.microsoft.com/about/legal/en/us/IntellectualProperty/Trademarks/EN-US.aspx> are trademarks of the Microsoft group of companies. All other trademarks are property of their respective owners

Product Number: 10990B

Part Number (if applicable):

Released: 10/2016

Module 1

Introduction to Business Intelligence and Data Modeling

Contents:

Lesson 1: Introduction to Business Intelligence	2
Lesson 2: The Microsoft Business Intelligence Platform	4
Lesson 3: Introduction to Reporting Services	7
Module Review and Takeaways	11

Lesson 1

Introduction to Business Intelligence

Contents:

Question and Answers

3

Question and Answers

Question: How does your organization approach BI? Is this a major part of the corporate strategy? What BI solutions does your organization use? What do you think are the major issues with your organization's approach to BI?

Answer: Answers will vary, depending on the students' experience.

Lesson 2

The Microsoft Business Intelligence Platform

Contents:

Question and Answers	5
Demonstration: Reporting Services in SharePoint-integrated Mode	5

Question and Answers

Question: You are setting up a test system for Reporting Services running on SharePoint-integrated mode with SSAS as the data provider. What is the minimum number of servers for this configuration?

- ☐ 1
- ☐ 2
- ☐ 3
- ☐ 4
- ☐ 5

Answer:

- ☒ 1
- ☐ 2
- ☐ 3
- ☐ 4
- ☐ 5

Demonstration: Reporting Services in SharePoint-integrated Mode

Demonstration Steps

View Data in a Transact-SQL Query

1. On the taskbar, click **Microsoft SQL Server Management Studio**.
2. In the **Connect to Server** dialog box, in the **Server name** box, type **MIA-SQL**, in the **Authentication** list, click **Windows Authentication**, and then click **Connect**.
3. On the **File** menu, point to **Open**, and then click **File**.
4. In the **Open File** dialog box, navigate to **D:\Demofiles\Mod01**, click **Sales_Order_Detail_Query.sql**, and then click **Open**.
5. Review the query, and click **Execute**.

View Data in a Reporting Services Report

1. On the taskbar, click **Internet Explorer**.
2. In the address bar, type **mia-sql/Reports_SQL2**, and then press Enter.
3. Click the **AdventureWorks Sample Reports** folder, and then click **Sales_Order_Detail**.
4. In the **First Order ID** box, type **57033**.
5. In the **Last Order ID** box, type **57033**, and then click **View Report**.
6. Compare the output from the query and the report.

Open a Report to SharePoint

1. In Internet Explorer, open a new Tab, and in the address bar, type **mia-sql/sites/adventureworks/Reports**, and then press Enter.
2. Click **Sales_Order_Detail**.
3. In the **Parameters** pane, in the **First Order ID** box, type **57033**.

4. In the **Last Order ID** box, type **57033**, and then click **Apply**.
5. Compare the output.
6. Close Internet Explorer and SQL Server Management Studio, without saving any changes.

Lesson 3

Introduction to Reporting Services

Contents:

Question and Answers	8
Demonstration: Reporting Services and Cloud Data	9

Question and Answers

The following features are either available in Native mode only or in both Native mode and SharePoint Integrated mode. You should place each item into the appropriate category. Indicate your answer by writing the category number to the right of each item.

Items	
1	Updated web portal
2	Paginated reports
3	Mobile reports
4	HTML5 rendering
5	KPIs

Category 1		Category 2
Native mode only		Native mode and SharePoint Integrated mode

Answer:

Category 1		Category 2
Native mode only		Native mode and SharePoint Integrated mode
Updated web portal Mobile reports KPIs		Paginated reports HTML5 rendering

Demonstration: Reporting Services and Cloud Data

Demonstration Steps

Open the Sales_Order_Detail Report in Report Builder

1. On the taskbar, click **Internet Explorer**.
2. In the address bar, type **mia-sql/Reports_SQL2**, and then press Enter.
3. Click the **AdventureWorks Sample Reports** folder.
4. In the top right of the **Sales_Order_Detail** report, click the ellipses, and then click **MANAGE**.
5. In the toolbar, click **Edit in Report Builder**.
6. In the **Internet Explorer** dialog box, clear the **Always ask before opening this type of address** check box, and then click **Allow**.
7. In the **Connect to Report Server** dialog, click **Yes**.

Add an Azure Data Source

1. In the **Sales_Order_Detail - Microsoft SQL Server Report Builder** window, in the **Report Data** pane, right-click **Data Sources**, click **Add Data Source**.
2. In the **Data Source Properties** dialog box, in the **Name** box, type **AdventureWorks_on_Azure**.
3. Click **Use a connection embedded in my report**.
4. In the **Select connection type** list, click **Microsoft Azure SQL Database**.
5. On the taskbar, click **Internet Explorer**.
6. In the address bar, type **portal.azure.com**, and then press Enter.
7. Use your Azure pass credentials to sign in to the portal.
8. On the left, click **SQL databases**.
9. Click the **AdventureWorksLT** database you created in the Preparation step.
10. Copy the **Server name** into the clipboard.
11. On the taskbar, click **Sales_Order_Detail - Microsoft SQL Server Report Builder**, and then click **Build**.
12. In the **Connection Properties** dialog box, in the **Server name** box, paste the copied server name.
13. Click **Use SQL Server Authentication**.
14. In the **User name** box, type **Student**.
15. In the **Password** box, type **Pa\$\$w0rd**, and then select the **Save my password** check box.
16. In the **Select or enter a database name** list, click **AdventureWorksLT**, and then click **OK**.
17. In the **Data Source Properties** dialog box, click **Test Connection**.
18. In the **Test Connection Result** dialog box, click **OK**.
19. In the **Data Source Properties** dialog box, click **OK**.

Add an Azure Dataset and Chart

1. In the **Report Data** pane, right-click **Datasets**, and then click **Add Dataset**.

2. In the **Dataset Properties** dialog box, in the **Name** box, type **Online_Sales**, and then click **Use a dataset embedded in my report**.
3. In the **Data source** list, click **AdventureWorks_on_Azure**.
4. In the **Query** box, type the follow code, and then click **OK**:

```
SELECT
    SalesOrderHeader.SalesOrderNumber
    ,Product.Name
    ,SalesOrderDetail.UnitPrice
    ,SalesOrderDetail.UnitPriceDiscount
    ,SalesOrderHeader.OrderDate
FROM
    SalesLT.SalesOrderHeader AS SalesOrderHeader
    INNER JOIN SalesLT.SalesOrderDetail AS SalesOrderDetail
        ON SalesOrderDetail.SalesOrderID = SalesOrderHeader.SalesOrderID
    INNER JOIN SalesLT.Product AS Product
        ON Product.ProductID = SalesOrderDetail.ProductID
WHERE SalesOrderHeader.SalesOrderNumber BETWEEN 'SO' + @FirstOrderID AND 'SO'+
    @LastOrderID
```

5. On the **Insert** tab, in the **Data Visualizations** group, click **Chart**, and then click **Chart Wizard**.
6. In the **New Chart** dialog box, on the **Choose a dataset** page, click **Online_Sales**, and then click **Next**.
7. On the **Choose a chart type** page, click **Pie**, and then click **Next**.
8. On the **Arrange chart fields** page, drag the **Name** field into the **Categories** box.
9. Drag the **Name** field into the **Values** box, then click the drop-down list for the field, click **Count**, and then click **Next**.
10. On the **Preview** page, click **Finish**.
11. Resize and move the inserted pie chart above the table.
12. Delete the Legend, and then click **Run**.
13. In the **First Order ID** boxes, type **57033**.
14. In the **Last Order ID** boxes, type **57033**, and then click **View Report**.
15. Note that there is no data for the pie chart.
16. In the **First Order ID** boxes, type **71774**.
17. In the **Last Order ID** boxes, type **71774**, and then click **View Report**.
18. In the **First Order ID** boxes, type **71780**.
19. In the **Last Order ID** boxes, type **71780**, and then click **View Report**.
20. Note the chart is querying data on the Azure database, and the table shows data from the on-premises database.
21. Close all open windows, without saving any changes.

Module Review and Takeaways

In this module, you've seen the tools provided by Microsoft to analyze data and produce Business Information reports. You've seen how Reporting Services can be integrated with SharePoint to enable reports to be more easily shared within an organization. Finally, you've seen how easy it is to query data on Azure and combine different data sources inside a single Reporting Services report. The following modules will be going into further detail about the tools used, and also include details on the new Mobile Report Publishing tool made available with SQL Server 2016.

Module 2

Reporting Services Data

Contents:

Lesson 1: Data Sources	2
Lesson 2: Connection Strings	6
Lesson 3: Datasets	9
Lesson 4: Filters and Parameters	12
Module Review and Takeaways	18
Lab Review Questions and Answers	19

Lesson 1

Data Sources

Contents:

Question and Answers	3
Demonstration: Creating a Data Source	3

Question and Answers

Question: You want to create a data source that several reports will use. Which type of data source should you create?

- ☐ A shared data source.
- ☐ An embedded data source.
- ☐ Either a shared data source or an embedded data source; it does not matter which.

Answer:

- ☒ A shared data source.
- ☐ An embedded data source.
- ☐ Either a shared data source or an embedded data source; it does not matter which.

Demonstration: Creating a Data Source

Demonstration Steps

Use the SSRS Portal to Create a Shared Data Source

1. Ensure that the MSL-TMG1, 10990B-MIA-DC, and 10990B-MIA-SQL virtual machines are running, and then log on to 10990B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa\$\$w0rd**.
2. In the **D:\Demofiles\Mod02** folder, run **Setup.cmd** as administrator.
3. In the **User Account Control** dialog box, click **Yes**.
4. In Windows® Internet Explorer®, browse to **mia-sql/Reports_SQL2**, and then click **DataSources**.
5. On the toolbar, click **New**, and then click **Data Source**.
6. On the **New Data Source** page, in the **Name** box, type **AdventureWorksDW-portal**.
7. In the **Type** box, verify that **Microsoft SQL Server** is selected.
8. In the **Connection string** box, type the following code:

```
data source="(local)";initial catalog=AdventureWorksDW
```

9. Click **Test connection**. Note that the test succeeds, and then click **Create**.
10. Leave Internet Explorer open for the next demonstration.

Use Report Designer to Define a Shared Data Source

1. Start Microsoft Visual Studio®, and on the **File** menu, point to **Open**, and then click **Project/Solution**.
2. In the **Open Project** dialog box, in **D:\Demofiles\Mod02\Demo**, click **Demo.sln**, and then click **Open**. This opens a new, empty SSRS project.
3. In Solution Explorer, right-click **Shared Data Sources**, and then click **Add New Data Source**.
4. In the **Shared Data Source Properties** dialog box, in the **Name** box, type **AdventureWorksDW-SSDT**.
5. In the **Type** box, verify that **Microsoft SQL Server** is selected, and then click **Edit**.
6. In the **Connection Properties** dialog box, in the **Server name** box, type **MIA-SQL**.
7. In the **Select or enter a database name** box, select **AdventureWorksDW**, and then click **Test Connection**.

8. In the **Test results** dialog box, click **OK**.
9. In the **Connection Properties** dialog box, click **OK**.
10. In the **Shared Data Source Properties** dialog box, on the **Credentials** page, point out that you can amend the credentials that are used for the data source. Verify that **Use Windows Authentication** is selected, and then click **OK**.
11. In Solution Explorer, expand **Shared Data Sources**, and then verify that the definition for the data source has been created. Point out that the data source definition is a file with an **.rds** extension.
12. In Internet Explorer, on the **SQL Server Reporting Services** page, click **Home**, and then click **Designer**, notice that the **AdventureWorksDW-SSDT** data source does not yet exist on the server; it must be deployed.
13. In Visual Studio, in Solution Explorer, right-click **AdventureWorksDW-SSDT.rds**, and then click **Deploy**. Wait for the deployment to succeed.
14. In Internet Explorer, on the **SQL Server Reporting Services** page, click **Refresh**.
15. Click **Designer**, and verify that the **AdventureWorksDW-SSDT** data source has now been created.
16. Leave Visual Studio and Internet Explorer open for the next demonstration.

Use Report Builder to Define an Embedded Data Source

1. In Internet Explorer, click **Home**.
2. On the toolbar, click **New**, and then click **Paginated Report**.
3. In the **Internet Explorer** dialog box, click **Allow**.
4. In the **Connect to Report Server** dialog box, click **Yes**.
5. In the **New Report or Dataset** dialog box, click **Blank Report**.
6. In the **Report Data** pane, right-click **Data Sources**, and then click **Add Data Source**.
7. In the **Data Source Properties** dialog box, on the **General** page, in the **Name** box, type **AdventureWorksDW_embedded**, and then click **Use a connection embedded in my report**.
8. In the **Select connection type** box, verify that **Microsoft SQL Server** is selected, and then click **Build**.
9. In the **Connection Properties** dialog box, in the **Server name** box, type **MIA-SQL**.
10. In the **Select or enter a database name** list, click **AdventureWorksDW**, and then click **Test Connection**.
11. In the **Test results** dialog box, click **OK**.
12. In the **Connection Properties** dialog box, click **OK**.
13. In the **Data Source Properties** dialog box, on the **Credentials** page, point out that you can amend the credentials that you use for the data. Note the warning message: **This information is only stored when you save the report to a report server**. Verify that **Use current Windows user** is selected, and then click **OK**.
14. On the **File** menu, click **Save**.
15. In the **Save As Report** dialog box, double-click **DataSources**.
16. In the **Name** box, type **report_builder.rdl**, and then click **Save**.

17. In Internet Explorer, close the **We're opening Report Builder** window, and then click **DataSources**. Note that, although the report definition has been saved to the folder, there is not an independent data source definition for the data source that is used in the report.
18. In the **report_builder** object, in the top-right corner, click the ellipsis (...) button, and then click **MANAGE**.
19. On the **Edit report_builder** page, click **Data sources**, verify that you can amend the properties of an embedded connection after the report is published. When you have finished examining the report properties, click **DataSources**.
20. Leave Internet Explorer and Report Builder open for the next demonstration.

Lesson 2

Connection Strings

Contents:

Question and Answers	7
Demonstration: Working with Azure SQL Database Connection Strings	7

Question and Answers

Question: Which of the following connection strings will configure a SQL Server connection to connect to the AWSales database on the Sales database engine instance on a server called SALES2?

- () data source=SALES2;initial catalog=Sales\AWSales
- () data source=SALES2\Sales;initial catalog=AWSales
- () data source=SALES2\Sales;initial catalog=AdminDB
- () data source=SALES2;initial catalog=AWSales
- () None of the above

Answer:

- () data source=SALES2;initial catalog=Sales\AWSales
- (v) data source=SALES2\Sales;initial catalog=AWSales
- () data source=SALES2\Sales;initial catalog=AdminDB
- () data source=SALES2;initial catalog=AWSales
- () None of the above

Demonstration: Working with Azure SQL Database Connection Strings

Demonstration Steps

Connect to Azure SQL Database from Report Builder

1. In Report Builder, in the **Report Data** pane, right-click **Data Sources**, and then click **Add Data Source**.
2. In the **Data Source Properties** dialog box, on the **General** page, in the **Name** box, type **AdventureWorksLT**, and then click **Use a connection embedded in my report**.
3. In the **Select connection type** list, click **Microsoft Azure SQL Database**. Notice that two optional properties are added to the connection string, and then click **Build**.
4. In the **Connection Properties** dialog box, in the **Server name** box, type the name of your Azure SQL Database server; this needs to be the fully qualified name, ending in **.database.windows.net**.
5. Verify that **Use SQL Server Authentication** is selected, in the **User name** box, type **Student**, in the **Password** box, type **Pa\$\$w0rd**.
6. In the **Select or enter a database name** box, type **AdventureWorksLT**, and then click **Test Connection**.
7. In the **Test results** dialog box, click **OK**.
8. In the **Connection Properties** dialog box, click **OK**.
9. In the **Data Source Properties** dialog box, click **OK**.
10. On the **File** menu, click **Save**.
11. Leave Report Builder open for the next demonstration.

Connect to Azure SQL Database from Report Designer

1. In Visual Studio, in Solution Explorer, right-click **Shared Data Sources**, and then click **Add New Data Source**.
2. In the **Shared Data Source Properties** dialog box, in the **Name** box, type **AdventureWorksLT**.

3. In the **Type** list, click **Microsoft Azure SQL Database**, and then click **Edit**.
4. In the **Connection Properties** dialog box, in the **Server name** box, type the name of your Azure SQL Database server; this needs to be the fully qualified name, ending in **.database.windows.net**.
5. In the **Authentication** box, verify that **SQL Server Authentication** is selected, in the **User name** box, type **Student**, in the **Password** box, type **Pa\$\$w0rd**.
6. Select the **Save my password** check box.
7. In the **Select or enter a database name** list, click **AdventureWorksLT**, and then click **Test Connection**.
8. In the **Test results** dialog box, click **OK**.
9. In the **Connection Properties** dialog box, click **OK**.
10. In the **Shared Data Source Properties** dialog box, click **OK**.
11. Leave Visual Studio open for the next demonstration.

Lesson 3

Datasets

Contents:

Question and Answers	10
Demonstration: Creating a Shared Dataset	10

Question and Answers

Question: True or false? A dataset definition includes a connection string.

() True

() False

Answer:

() True

(√) False

Demonstration: Creating a Shared Dataset

Demonstration Steps

Use Report Designer to Create a Shared Dataset

1. In Visual Studio, in Solution Explorer, right-click **Shared Datasets**, and then click **Add New Dataset**.
2. In the **Shared Dataset Properties** dialog box, in the **Name** box, type **AssociatedOrders-SSDT**.
3. In the **Data source** list, click **AdventureWorksDW-SSDT**, and then click **Query Designer**.
4. In the **Query Designer** dialog box, click **Add Table**.
5. In the **Add Table** dialog box, on the **Views** tab, click **vAssocSeqOrders**, click **Add**, and then click **Close**.
6. In the Diagram pane, in the **vAssocSeqOrders** object, select **OrderNumber**, **CustomerKey**, **Region**, and **IncomeGroup**.
7. Click the **Run** button (!) to preview the results of the query. Briefly review the data in the Result pane, click **OK**.
8. In the **Shared Dataset Properties** dialog box, click **OK**.
9. In Solution Explorer, expand **Shared Datasets**, right-click **AssociatedOrders-SSDT.rsd**, and then click **Deploy**.



Note: There is currently an issue with SSDT. When deploying you may get this error: **The shared dataset definition is not valid. Details: The required attribute 'Name' is missing.**

To resolve it edit the D:\Demofiles\Mod02\Demo\Demo \AssociatedOrders-SSDT.rsd file in notepad and change:

<DataSet>

To:

<DataSet Name="">

Save and you should be able to deploy.

10. In Internet Explorer, on the **SQL Server Reporting Services** page, click **Home**, and then click **Designer**. Verify that the **AssociatedOrders-SSDT** shared data source has been created. If the shared data source does not appear, refresh the page.
11. Leave Visual Studio and Internet Explorer open for the next demonstration.

Use Report Builder to Create a Shared Dataset

1. In Report Builder, on the **File** menu, click **New**.
2. In the **New Report or Dataset** dialog box, click **New Dataset**, and then click **Browse other data sources**.
3. In the **Select Data Source** dialog box, double-click **DataSources**, click **AdventureWorksDW-portal**, and then click **Open**.
4. In the **New Report or Dataset** dialog box, click **Create**. Report Builder will switch to the Shared Dataset Design view.
5. In the **Database view** pane, expand **Views**, select **vAssocSeqOrders**, and then click **Edit as Text** to switch to the Query Designer view.
6. Click the **Run** button (!) to preview the output of the query. Briefly examine the query results, and then on the **File** menu, click **Save**.
7. In the **Save As Dataset** dialog box, verify that the value of the **Look in** box is **http://mia-sql/ReportServer_SQL2/DataSources**, in the **Name** box, type **AssociatedOrders-builder**, and then click **OK**.
8. In Internet Explorer, on the **SQL Server Reporting Services** page, click **Home**, and then click **DataSources**. Verify that the **AssociatedOrders-builder** shared dataset appears on the report server portal.
9. Leave Report Builder and Internet Explorer open for the next demonstration.

Lesson 4

Filters and Parameters

Contents:

Question and Answers	13
Demonstration: Using Filters	13
Demonstration: Using Parameters	15

Question and Answers

Question: A dataset that is linked to a SQL Server data source has the query text that is shown in the following code:

```
SELECT CurrencyName FROM dbo.DimCurrency WHERE CurrencyKey = @CurrencyKey;
```

Does the dataset use a filter or a parameter to restrict the rows that are returned from the **DimCurrency** table?

Answer: The dataset uses a parameter.

Demonstration: Using Filters

Demonstration Steps

Add a Filter to a Shared Dataset in Report Builder

1. In Report Builder, click **Set Options**.
2. In the **Shared Dataset Properties** dialog box, on the **Filters** page, click **Add**.
3. In the **Expression** list, click **[Region]**. Verify that **Text** is selected in the data type box, and that the **Operator** box has the value **=**.
4. In the **Value** box, type **Pacific**, and then click **OK**.
5. Click **Run**. Notice that in the Query results pane, the filter is not applied; rows with values other than **Pacific** in the **Region** column are returned.
6. On the **File** menu, click **Save**.
7. In Internet Explorer, on the **DataSources** page, on the **AssociatedOrders-builder** shared dataset, click the ellipsis (...) button, and then click **MANAGE**.
8. On the **Edit AssociatedOrders-builder** page, click **Data Preview**, and then click **Load data**. Observe that only rows with a **Region** of **Pacific** are returned. Notice that the values in the **IncomeGroup** column include **Low**, **Moderate**, and **High**.
9. Click **DataSources** to return to the DataSources folder.

Add a Filter to a Report Dataset in Report Builder

1. In Report Builder, on the **File** menu, click **Open**.
2. In the **Microsoft SQL Server Report Builder** dialog box, click **No**.
3. In the **Open Report** dialog box, double-click **DataSources**, click **report_builder**, and then click **Open**.
4. In the **Report Data** pane, right-click **Datasets**, and then click **Add Dataset**.
5. In the **Dataset Properties** dialog box, on the **Query** page, in the **Name** box, type **AssociatedOrders_filtered**. Verify that **Use a shared dataset** is selected, and then click **Browse**.
6. In the **Select Dataset** dialog box, double-click **DataSources**, click **AssociatedOrders-builder**, and then click **Open**.
7. On the **Filters** page, click **Add**.
8. In the **Expression** list, click **[IncomeGroup]**. Verify that **Text** is selected in the data type box, and in the **Operator** list, click **<>**.
9. In the **Value** box, type **High**, and then click **OK**.
10. On the **Insert** tab, click **Table**, and then click **Table Wizard**.

11. In the **New Table or Matrix** wizard, on the **Choose a dataset** page, click **AssociatedOrders_filtered**, and then click **Next**.
12. On the **Arrange fields** page, drag **OrderNumber**, **CustomerKey**, **Region**, and **IncomeGroup** from the **Available fields** list to the **Values** list, and then click **Next**.
13. On the **Choose the layout** page, click **Next**.
14. On the **Preview** page, click **Finish**.
15. Click **Run** to preview the report.
16. In the **Enter Data Source Credentials** dialog box, in the **User name** box, type **Student**, and in the **Password** box, type **Pa\$\$w0rd**, and then click **OK**. Notice that the report includes only rows from the **Pacific** region (because of the filter on the shared dataset) with an income group of **Low** or **Moderate** (because of the filter on the report dataset).
17. Click **Design**, and then leave Report Builder open for the next demonstration.

Add a Filter to a Shared Dataset in Report Designer

1. In Visual Studio, in Solution Explorer, under **Shared Datasets**, right-click **AssociatedOrders-SSDT.rsd**, and then click **Open**.
2. In the **Shared Dataset Properties** dialog box, on the **Filters** page, click **Add**.
3. In the **Expression** list, click **[Region]**. Verify that **Text** is selected in the data type box, and that the **Operator** box has the value **=**.
4. In the **Value** box, type **Pacific**, and then click **OK**.
5. Right-click **AssociatedOrders-SSDT.rsd**, and then click **Deploy**.
6. In Internet Explorer, on the **SQL Server Reporting Services** page, click **Home**, and then click **Designer**.
7. On the **AssociatedOrders-SSDT** shared dataset, click the ellipsis (...) button, and then click **MANAGE**.
8. Click **Data Preview**, and then click **Load data**. Observe that only rows with a **Region** of **Pacific** are returned. Notice that the values in the **IncomeGroup** column include **Low**, **Moderate**, and **High**.
9. Close Internet Explorer.

Add a Filter to a Report Dataset in Report Designer

1. In Visual Studio, in Solution Explorer, right-click **Reports**, point to **Add**, and then click **New Item**.
2. In the **Add New Item - Demo** dialog box, click **Report**, and then click **Add**.
3. In the **Report Data** pane, right-click **Datasets**, and then click **Add Dataset**.
4. In the **Dataset Properties** dialog box, on the **Query** page, in the **Name** box, type **AssociatedOrders_filtered**. Verify that **Use a shared dataset** is selected, and then click **AssociatedOrders-SSDT**.
5. On the **Filters** page, click **Add**.
6. In the **Expression** list, click **[IncomeGroup]**. Verify that **Text** is selected in the data type box, and in the **Operator** list, click **<>**.
7. In the **Value** box, type **High**, and then click **OK**.
8. In the **Toolbox** pane, drag **Table** to the report design section of the Report1.rdl pane.
9. In the **Report Data** pane, drag **OrderNumber** to the first column of the **Data** row of the new table.

10. In the **Report Data** pane, drag **Region** to the second column of the **Data** row of the new table.
11. In the **Report Data** pane, drag **IncomeGroup** to the third column of the **Data** row of the new table.
12. In the **Report Data** pane, drag **CustomerKey** to the right side of the **IncomeGroup** column—when a blue marker appears.
13. Click the **Preview** tab to preview the report. Notice that the report includes only rows from the **Pacific** region (because of the filter on the shared dataset) with an income group of **Low** or **Moderate** (because of the filter on the report dataset).
14. Click the **Design** tab, and then leave Visual Studio open for the next demonstration.

Demonstration: Using Parameters

Demonstration Steps

Define a Parameter for a Shared Dataset in Report Builder

1. In Report Builder, on the **File** menu, click **Open**.
2. In the **Microsoft SQL Server Report Builder** dialog box, asking to save changes, click **Yes**.
3. In the **Open Report** dialog box, in the **Items of type** list, click **Datasets (*.rsd)**, click **AssociatedOrders-builder**, and then click **Open**.
4. Click **Edit as Text**.
5. In the **Query Designer** dialog box, click **Yes**.
6. In the **Database view** pane, expand **Views**, and then select the **vAssocSeqOrders** check box.
7. In the **Applied filters** pane, click **Add Filter**.
8. In the **Field name** column, click **OrderNumber**, and then click **CustomerKey**.
9. In the **Operator** column, click **is**, and then click **is more than or equal to**.
10. In the **Parameter** column, select the check box.
11. Click **Edit as Text** to view the query text. Notice that a WHERE clause has been added to the query, referencing a parameter named **@CustomerKey**, and then click **Set Options**.
12. In the **Shared Dataset Properties** dialog box, on the **Parameters** page, notice that an entry appears for the **@CustomerKey** parameter, click **Cancel**, and then click the **Run** button (!).
13. In the **Define Query Parameters** dialog box, in the **Parameter Value** column, in the first row, type **29000**, and then click **OK**. Notice that the parameter value is passed to the WHERE clause, and the result set is limited to rows where **CustomerKey** is greater than or equal to 29,000.

Set the Source Value for a Parameter in Report Builder

1. On the **File** menu, click **Open**.
2. In the **Microsoft SQL Server Report Builder** dialog box, asking to save changes, click **Yes**.
3. In the **Open Report** dialog box, click **report_builder**, and then click **Open**.
4. In the **Report Data** pane, expand **Datasets**, right-click **AssociatedOrders_filtered**, and then click **Dataset Properties**.
5. In the **Dataset Properties** dialog box, on the **Parameters** page, notice that the parameter is not listed.

6. On the **Query** page, click **Refresh Fields**, and then return to the **Parameters** page; an entry for the parameter is now shown. Notice that fewer options are available in this view of the **Parameters** page than were shown in the Shared Dataset Properties dialog box when the parameter was created, click **OK**.
7. To preview the report, click **Run**.
8. In the **Enter Data Source Credentials** dialog box, in the **User name** box, type **Student**, and in the **Password** box, type **Pa\$\$w0rd**, and then click **OK**. Notice that the report is not rendered immediately; instead, a report header is shown where you can provide a value for the *Customer Key* parameter.
9. In the **Customer Key** box, type **29000**, and then click **View Report**.
10. Notice that the values that are shown in the report are limited both by the parameter value and by the filters that you applied in the previous demonstration, click **Design**.
11. In the **Report Data** pane, right-click **AssociatedOrders_filtered**, and then click **Dataset Properties**.
12. In the **Dataset Properties** dialog box, on the **Parameters** page, notice that the **Parameter Value** box has been automatically populated with the name of a report parameter—**[@CustomerKey]**.
13. In the **Parameter Value** box, type **29000**, and then click **OK**. This breaks the link between the report parameter and the dataset parameter.
14. Click **Run**. Notice that, even though the link between the dataset parameter and the report parameter has been broken, the report parameter has not been deleted from the report definition.
15. In the **Customer Key** box, type **50000**, and then click **View Report**. There are no rows in the dataset where **CustomerKey** is greater than or equal to **50000**.
16. Notice that the values in the report are returned, based on the value that was hard-coded in the dataset properties window. The value of the report parameter is ignored.
17. Close Report Builder without saving changes.

Define a Parameter for a Shared Dataset in Report Designer

1. In Visual Studio, in Solution Explorer, under **Shared Datasets**, right-click **AssociatedOrders-SSDT.rsd**, and then click **Open**.
2. In the **Shared Dataset Properties** dialog box, on the **Query** page, edit the text of the **Query** box so that it matches the following code:

```
SELECT      OrderNumber, CustomerKey, Region, IncomeGroup
FROM        vAssocSeqOrders
WHERE      CustomerKey >= @CustomerKey
```

3. On the **Parameters** page, notice that the parameter has been detected automatically and added to the parameter list.
4. On the **Query** page, click **Query Designer**.
5. In the **Query Designer** dialog box, in the **Grid** pane, in the **CustomerKey** row, in the **Filter** column, notice the parameter definition, and then click the **Run** button (!).
6. In the **Query Parameters** dialog box, in the **Value** column, in the first row, type **29000**, and then click **OK**.
7. In the **Results** pane, scroll down to verify that the parameter value has restricted the rows that are returned to those where **CustomerKey** is greater than or equal to **29000**, and then click **OK**.
8. In the **Shared Dataset Properties** dialog box, click **OK**.

Set the Source Value for a Parameter in Report Designer

1. In the **Report Data** pane, under **Datasets**, right-click **AssociatedOrders_filtered**, and then click **Dataset Properties**.
2. In the **Dataset Properties** dialog box, on the **Parameters** page, notice that the addition of the parameter to the shared dataset is not reflected.
3. On the **Query** page, click **Refresh Fields**.
4. On the **Parameters** page, notice that an entry now exists in the parameters list for the *@CustomerKey* parameter, and then click **OK**.
5. In the **Report1.rdl** pane, click **Preview**. Notice that the report is not rendered immediately; instead, a report header is shown where you can provide a value for the *Customer Key* parameter.
6. In the **Customer Key** box, type **29000**, and then click **View Report**.
7. Notice that the values that are shown in the report are limited both by the parameter value and by the filters that you applied in the previous demonstration, and then click **Design**.
8. In the **Report Data** pane, right-click **AssociatedOrders_filtered**, and then click **Dataset Properties**.
9. In the **Dataset Properties** dialog box, on the **Parameters** page, notice that the **Parameter Value** box has been automatically populated with the name of a report parameter—*[@CustomerKey]*.
10. In the **Parameter Value** box, type **29000**, and then click **OK**. This breaks the link between the report parameter and the dataset parameter.
11. In the **Report1.rdl** pane, click **Preview**. Notice that, even though the link between the dataset parameter and the report parameter has been broken, the report parameter has not been deleted from the report definition.
12. In the **Customer Key** box, type **50000**, and then click **View Report**. There are no rows in the dataset where **CustomerKey** is greater than or equal to **50000**.
13. Notice that the values in the report are returned, based on the value that was hard-coded in the dataset properties dialog box. The value of the report parameter is ignored.
14. Close Visual Studio without saving any changes.

Module Review and Takeaways

Best Practice

To simplify the management and administration of your SSRS servers, use shared data sources whenever possible.

Review Question(s)

Question: Which SSRS data access object would you typically create first when you configure data access to a source system?

- ☐ An embedded dataset
- ☐ A shared dataset
- ☐ A parameter
- ☐ A data source
- ☐ A filter

Answer:

- ☐ An embedded dataset
- ☐ A shared dataset
- ☐ A parameter
- ☒ A data source
- ☐ A filter

Lab Review Questions and Answers

Lab A: Configuring Data Access with Report Builder

Question and Answers

Lab Review

Question: True or false? In this exercise, you could have used a parameter instead of a filter to restrict the values that the **FilteredTerritories** dataset returned (based on the **dbo.usp_SalesTerritory** stored procedure).

Assume that you cannot make changes to the **dbo.usp_SalesTerritory** stored procedure when you select your answer.

☐ True

☐ False

Answer:

☐ True

☒ False

Lab B: Configuring Data Access with Report Designer

Question and Answers

Question: True or false? In this exercise, you could have used a parameter instead of a filter to restrict the values that the **FilteredTerritories** dataset returned (based on the **dbo.usp_SalesTerritory** stored procedure).

Assume that you cannot make changes to the **dbo.usp_SalesTerritory** stored procedure when you select your answer.

☐ True

☐ False

Answer:

☐ True

☒ False

Module 3

Implementing Reports

Contents:

Lesson 1: Creating a Report with the Report Wizard	2
Lesson 2: Creating a Report	5
Lesson 3: Show Data Graphically in a Report	9
Module Review and Takeaways	15

Lesson 1

Creating a Report with the Report Wizard

Contents:

Question and Answers	3
Demonstration: Using the Report Wizard to Build a Report	3

Question and Answers

Put the following steps in order by numbering each to indicate the correct order.

	Steps
	Select or create a data source.
	Design a query to define the dataset.
	Specify the report type.
	Add fields to the report.
	Give the report a name.
	Preview the report.

Answer:

	Steps
1	Select or create a data source.
2	Design a query to define the dataset.
3	Specify the report type.
4	Add fields to the report.
5	Give the report a name.
6	Preview the report.

Demonstration: Using the Report Wizard to Build a Report

Demonstration Steps

1. Ensure that both 10990B-MIA-DC and 10990B-MIA-SQL virtual machines are running, and then log on to 10990B -MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa\$\$w0rd**.
2. In the **D:\Demofiles\Mod03** folder, run **Setup.cmd** as Administrator.
3. Click **Yes** when prompted to confirm that you want to run the command file, and wait for the script to finish. If a Visual Studio Just-In-Time Debugger dialog box shows an error relating to w3wp.exe, click **No, cancel debugging**.
4. Start Visual Studio.
5. On the **File** menu, point to **Open**, and then click **Project/Solution**. In the **Open Project** dialog box, go to the **D:\Demofiles\Mod03** folder, click **Reports Demo.sln**, and then click **Open**.
6. In Solution Explorer, right-click **Reports**, and then click **Add New Report**.
7. On the **Welcome to the Report Wizard** page, click **Next**.
8. On the **Select the Data Source** page, click **Next**.
9. On the **Design the Query** page, click **Query Builder**.
10. In the **Query Designer** dialog box, on the toolbar, click **Add Table**.

11. In the **Add Table** dialog box, click **DimGeography**, hold down the CTRL key, click **DimReseller**, click **FactResellerSales**, click **Add**, and then click **Close**.
12. In the **Query Designer** dialog box, in the upper pane, select the following columns:
 - Dim Geography.EnglishCountryRegionName
 - DimGeography.StateProvinceName
 - DimGeography.City
 - DimReseller.ResellerName
 - FactResellerSales.SalesOrderNumber
 - FactResellerSales.OrderDate
 - FaceResellerSales.SalesAmount
13. Add the following values to the **Alias** column:
 - EnglishCountryRegionName: **Country-Region**
 - StateProvincename: **State**
 - ResellerName: **Reseller**
14. Click **OK**.
 - On the **Design the Query** page, click **Next**.
 - On the **Select the Report Type** page, ensure that **Tabular** is selected, and then click **Next**.
 - On the **Design the Table** page, add all fields to the **Details** section then click **Next**.
15. On the **Completing the Wizard** page, in the **Report name** box, type **Reseller Sales**, and then click **Finish**.
16. Click the **Preview** tab to view the report.
17. Leave Visual Studio open for the next demonstration.

Lesson 2

Creating a Report

Contents:

Question and Answers	6
Demonstration: Creating a Report	6
Demonstration: Publishing a Report	8

Question and Answers

Question: True or false? For every page break added into a report, the Excel rendering extension will generate a new worksheet.

- () True
() False

Answer:

- (√) True
() False

Demonstration: Creating a Report

Demonstration Steps

Create a Report in Report Designer

1. In Visual Studio, in Solution Explorer, right-click **Reports**, point to **Add**, and then click **New Item**.
2. In the **Add New Item – Reports Demo** dialog box, click **Report**, in the **Name** box, type **Reseller Sales 2** and then click **Add**.
3. In Solution Explorer, right-click **Shared Datasets**, and then click **Add New Dataset**.
4. In the **Shared Dataset Properties** dialog box, in the **Name** box, type **MyDataset**, in the **Data source** list, click **AdventureWorksDW**, in the **Query** box, type the following code, and then click **OK**:

```
SELECT      DimGeography.EnglishCountryRegionName, DimGeography.StateProvinceName,
DimGeography.City, DimReseller.ResellerName, FactResellerSales.SalesOrderNumber,
FactResellerSales.OrderDate, FactResellerSales.SalesAmount
FROM
DimGeography INNER JOIN
DimReseller ON DimGeography.GeographyKey = DimReseller.GeographyKey INNER JOIN
FactResellerSales ON DimReseller.ResellerKey = FactResellerSales.ResellerKey
```

5. From the Toolbox, drag a Table item to the report design surface.
6. In the **Dataset Properties** dialog box, in the **Name** box, type **MyDataset**, and then in the list of datasets, click **MyDataset**.
7. On the **Fields** page, change the Field Name values as shown in the following table, and then click **OK**:

Field Name	Field Source
Country_Region	EnglishCountryRegionName
State	StateProvinceName
Reseller	ResellerName

8. In the **Report Data** pane, expand **Datasets**, expand **MyDataset**, and then drag each data item in turn to the table header. If the available cells are full, position the item on the right-side border of the table and a new cell will automatically appear.
9. From the **Toolbox**, drag a text box to the report design surface, aligning it with the top of the table.
10. In the text box, type the text **Reseller Sales**.
11. Click the **Preview** tab to view the report.
12. Leave Visual Studio open for the next demonstration.

Create a Report in Report Builder

1. Start Internet Explorer and browse to **mia-sql/Reports_SQL2**.
2. On the **Home** page, in the toolbar, click **New**, then click **Paginated Report**.
3. In the **Internet Explorer** dialog box, click **Allow**.
4. If the **Connect to Report Server** dialogue appears, click **Yes**.
5. In the **New Report or Dataset** dialog box, click **Blank Report**.
6. In Report Data, right-click **Datasets**, and then click **Add Dataset**.
7. In the **Dataset Properties** dialog box, in the **Name** box, type **MyDataset**, select **Use a dataset embedded in my report**, and then click **New**.
8. In the **Data Source Properties** dialog box, in the **Name** box, type **AdventureWorksDW**, and then click **Use a connection embedded in my report**.
9. In the **Connection string** text box, type the following:


```
Data Source=MIA-sql;Initial Catalog=AdventureWorksDW
```
10. In the Credentials pane, click **Use current Windows user. Kerberos delegation might be required**. Then click **OK**.
11. In the **Dataset Properties** dialog box, in the **Query** box, type the following:


```
SELECT      DimGeography.EnglishCountryRegionName, DimGeography.StateProvinceName,
DimGeography.City, DimReseller.ResellerName, FactResellerSales.SalesOrderNumber,
FactResellerSales.OrderDate, FactResellerSales.SalesAmount,
DimDate.CalendarYear AS SalesYear
FROM
DimGeography INNER JOIN
DimReseller ON DimGeography.GeographyKey = DimReseller.GeographyKey INNER JOIN
FactResellerSales ON DimReseller.ResellerKey = FactResellerSales.ResellerKey INNER
JOIN
DimDate ON [DateKey] = FactResellerSales.OrderDateKey
```
12. Click **Refresh Fields**.
13. On the **Fields** page, change the **Field Name** values to match the following table, and then click **OK**.

Field Name	Field Source
Country_Region	EnglishCountryRegionName
State	StateProvinceName
Reseller	ResellerName
14. On the **Insert** menu, click **Table**, and then click **Insert Table**.
15. Click anywhere on the report design surface to insert a new table.
16. In **Report Data**, expand **Datasets**, expand **MyDataset**, and then drag each data item in turn to the table header. If the available cells are full, position the item on the right-side border of the table and a new cell will automatically appear.
17. Click the **Click to add title** text box.
18. In the text box, type **Reseller Sales**.
19. On the **Home** menu, click **Run**.

20. Leave Report Builder open for the next demonstration.

Demonstration: Publishing a Report

Demonstration Steps

Publishing a Report from Report Designer

1. In Visual Studio, in Solution Explorer, right-click **Reports Demo**, and then click **Properties**.
2. In the **Reports Demo Property Pages** dialog box, ensure the **TargetServerURL** property is set as **http://mia-sql/ReportServer_SQL2** and the **TargetReportFolder** property is set as **Designer**, and then click **OK**.
3. In Solution Explorer, right-click **Reseller Sales.rdl**, and click **Deploy**. Wait for the deployment to complete.
4. Start Internet Explorer and go to **mia-sql/Reports_SQL2**.
5. Click **Designer**, and then click **Reseller Sales**. Check that the report has been correctly published.
6. Close Internet Explorer.

Publishing a Report from Report Builder

1. In Report Builder, on the **File** menu, click **Save As**.
2. In the **Look in** drop-down list, ensure that **http://mia-sql/ReportServer_SQL2** is selected.
3. Double-click **Builder** and then in the **Name** box, type **Reseller Sales**, and then click **Save**. If the **Save As Report** window appears, click **Yes**.
4. Start Internet Explorer and go to the address **mia-sql/Reports_SQL2**.
5. Click **Builder**, and then click **Reseller Sales**. Check that the report has been correctly published.
6. Close Internet Explorer.
7. Leave Report Builder open for the next demonstration.

Lesson 3

Show Data Graphically in a Report

Contents:

Question and Answers	10
Demonstration: Adding Images and Charts	10
Demonstration: Adding Data Bars, Sparklines, and Indicators to a Report	12
Demonstration: Adding Maps to a Report	13

Question and Answers

Question: Which object can you use to visually display aggregated data on a report?

- () Image
- () Chart
- () Data bar
- () Sparkline
- () Indicator

Answer:

- () Image
- () Chart
- () Data bar
- (√) Sparkline
- () Indicator

Demonstration: Adding Images and Charts

Demonstration Steps

Add an Image to a Report in Report Designer

1. In Visual Studio, click **Design** to ensure you are in Design mode.
2. If the **Toolbox** is not visible, on the **View** menu, click **Toolbox**, and in the **Toolbox**, drag an **Image** to the design surface.
3. Name the Image **AdventureWorksLogo**, and then click **Import**.
4. Go to the folder **D:\Demofiles\Mod03**, select the **AdventureWorks.jpg** image, click **Open**, and then click **OK**.
5. Resize the image to an appropriate size and reposition it in the top right corner of the report above the table.

Add a Chart to a Report in Report Designer

1. In Visual Studio, click **Design** to ensure you are in Design mode.
2. Click the tablix data region so that the gray row and column headers appear, and click the gray box where the row and column headers intersect at the top left to select the data region. Then drag the multidirectional arrow handle to move the data region down about 10 centimeters.
3. If the **Toolbox** is not visible, on the **View** menu, click **Toolbox**, and in the **Toolbox**, drag a chart to the blank area you just created above the tablix data region. Then in the **Select Chart Type** dialog box, in the **Shape** section, select the third chart style (**3-D Pie**) and click **OK**.

Specify Chart Data in Report Designer

1. Click the chart to display the Chart Data pane.
2. In the Chart Data pane, in the **Values** section, add the **SalesAmount** field.
3. In the Chart Data pane, in the **Category Groups** section, add the **Country_Region** field.

Format a Chart in Report Designer

1. On the **Design** tab, click the **Chart Title** label, and press the F4 key. In the Properties pane, change the **Caption** property to **Reseller Sales by Geography**.
2. Click a blank area just below the pie chart. Right-click the move icon (four arrows) and then click **Chart Properties**.
3. In the **Chart Properties** dialog box, in the **Color palette** drop-down list, select any color palette, and click **OK**.
4. Preview the report and view the formatted chart.
5. Click the **Export** button, and then click **Excel**. When prompted, save the report as **Reseller Sales.xlsx** in the **D:\Demofiles\Mod03** folder, replacing any existing files.

Add an Image to a Report in Report Builder

1. In Report Builder, click **Design**. On the **Insert** tab, click **Image**, and then click the top right corner of the design surface, above the tablix.
2. Name the image **AdventureWorksLogo**, and then click **Import**.
3. Go to the folder **D:\Demofiles\Mod03**, select the **AdventureWorks.jpg** image, click **Open**, and then click **OK**.
4. Resize the image to an appropriate size and reposition it in the top right corner of the report above the tablix.

Add a Chart to a Report in Report

1. In Report Builder, click **Design**.
2. Click the tablix data region so that the gray row and column headers appear, and click the gray box where the row and column headers intersect at the top left to select the data region. Then drag the multidirectional arrow handle to move the data region down about 10 centimeters. You may have to resize the page and footer to achieve this.
3. On the **Insert** tab, click **Chart**, click **Insert Chart**, and then click anywhere in the space above the tablix.
4. In the **Select Chart Type** dialog, click **3-D Pie**, and then click **OK**.

Specify Chart Data in Report Builder

1. Double-click the chart to display the Chart Data pane.
2. In the Chart Data pane, in the **Values** section, add the **SalesAmount** field.
3. In the Chart Data pane, in the **Category Groups** section, add the **Country_Region** field.

Format a Chart in Report Builder

1. Click **Design**, double-click the **Chart Title** label, and change the chart title to **Reseller Sales by Geography**.
2. Click a blank area on the report design surface, and then right-click the chart and click **Chart Properties**.
3. In the **Chart Properties** dialog box, in the **Color palette** drop-down list, select any color palette, and click **OK**.
4. Click **Run** to preview the report and view the formatted chart.
5. Click **Export**, and then click **Excel**. When prompted, save the report as **Reseller Sales Builder.xlsx** in the **D:\Demofiles\Mod03** folder.

6. Minimize Report Builder, and then open the **Reseller Sales.xlsx** Excel workbook and view the exported report, noting that the first worksheet contains the chart as an image. Then close Excel.

Demonstration: Adding Data Bars, Sparklines, and Indicators to a Report

Demonstration Steps

Add a Data Bar in Report Designer

1. Switch to Visual Studio. In Solution Explorer, expand **Reports** and then double-click the **USA Sales by State** report.
2. Click **Preview** and observe that the report displays Sales values by state and year.
3. Click **Design**, right-click the **[SalesYear]** column on the tablix, point to **Insert Column** and then click **Outside Group – Left**.
4. Double-click the header row of the new column and type **Sales Volume**.
5. From the **Toolbox**, drag a **Data bar** onto the report, in the detail cell under **Sales Volume**.
6. In the **Select Data Bar Type** dialog, select **Bar**, and then click **OK**.
7. Double-click the newly added Data bar and in the **Chart Data** dialog, click the plus symbol next to **Values**, and then click **SalesAmount**.

Add a Sparkline in Report Designer

1. Click **Design**, right-click the **[SalesYear]** column on the tablix, point to **Insert Column**, and then click **Outside Group – Right**.
2. Double-click the header row of the new column and type **Trend**.
3. From the **Toolbox**, drag a **Sparkline** onto the report, in the detail cell under **Trend**.
4. In the **Select Sparkline Type** dialog box, click **Line**, and then click **OK**.
5. Double-click the newly added sparkline, click the plus symbol next to **Values**, and then click **SalesAmount**.
6. Click the plus symbol next to **Category Groups**, and then click **SalesYear**.
7. Click **Preview** to show the report with the newly added trend sparkline.

Add a Data Bar in Report Builder

1. In Report Builder, click **Design**, click **File**, and then click **Open**.
2. In the **Microsoft SQL Server Report Builder** dialog box, click **Yes**.
3. Go to the **Builder** folder on the server and open the **USA Sales by State** report.
4. Click **Run** and observe that the report shows yearly sales values for each USA state.
5. Click **Design**, right-click the **[SalesYear]** header row, point to **Insert Column**, and then click **Outside Group – Left**.
6. Double-click the header row of the new column and type **Sales Volume**.
7. Select the cell under Sales Volume, and on the **Insert** tab, double-click **Data bar**.
8. In the **Select Data Bar Type** dialog box, click **Bar**, and then click **OK**.
9. Double-click the newly added **Data bar** and in the **Chart Data** dialog box, click the plus symbol next to **Values**, and then click **SalesAmount**.

Add a Sparkline in Report Builder

1. In Report Builder, click **Design**, right-click the **[SalesYear]** column on the tablix, point to **Insert Column**, and then click **Outside Group – Right**.
2. Double-click the header row of the new column and type **Trend**.
3. Select the cell under Trend and on the **Insert** tab, double-click **Sparkline**.
4. In the **Select Sparkline Type** dialog box, click **Line**, and then click **OK**.
5. Double-click the newly added sparkline, click the plus symbol next to **Values**, and then click **SalesAmount**.
6. Click the plus symbol next to Category Groups, and then click **SalesYear**.
7. Click **Run** to show the revised report.
8. Leave Report Builder open for the next demonstration.

Demonstration: Adding Maps to a Report

Demonstration Steps

Add a Map to a Report in Report Designer

1. Switch to Visual Studio, click **Design**, click the tablix data region so that the gray row and column headers appear, and click the gray box where the row and column headers intersect at the top left to select the data region. Then drag the multidirectional arrow handle to move the data region down about 10 centimeters.
2. From the **Toolbox**, drag a **Map** item onto the report design surface above the tablix.
3. In the **New Map Layer** dialog box, click **USA by State Exploded**, and then click **Next**.
4. In the **Choose spatial data and map view options** dialog box, click **Next**.
5. In the **Choose map visualization** dialog box, click **Bubble Map**, and then click **Next**.
6. In the **Choose the analytical dataset** dialog box, click **MyDataset**, and then click **Next**.
7. In the **Specify the match fields for spatial and analytical data** dialog box, click the check box in the **Match Fields** column next to STATENAME, in the **Select a field** drop-down list, select **State**, and then click **Next**.
8. In the **Choose color theme and data visualization** dialog box, clear **Use bubble sizes to visualize data**, check the **Use polygon colors to visualize data**. In the Data field, select **[Sum(SalesAmount)]**, in the Color rule, select **Red-Yellow-Green** and then click **Finish**.
9. Rearrange the map and tablix on the design surface so the map is directly above the tablix.
10. Double-click **Map Title** and then, in the Properties pane set the (Text) property to **Sales by State**.
11. Double-click the map legend and then, in the Properties pane, change the **(Hidden)** property to **True**.
12. Double-click the **Map Distance Scale** and then, in the Properties pane, change the **(Hidden)** property to **True**.
13. Double-click the **Map Color Scale** and reposition in the bottom right of the map.
14. Click **Preview** to view the completed report.
15. Close Visual Studio, saving changes if you are prompted.

Add Maps to a Report in Report Builder

1. Switch to Report Builder, then click **Design** to return to Design mode.
2. Click the tablix data region so that the gray row and column headers appear, and click the gray box where the row and column headers intersect at the top left to select the data region. Then drag the multidirectional arrow handle to move the data region down about 10 centimeters.
3. On the **Insert** tab, click **Map**, and then click **Map Wizard**.
4. In the **Choose a source of spatial data** dialog box, click **USA by State Exploded**, and then click **Next**.
5. In the **Choose spatial data and map view options** dialog box, click **Next**.
6. In the **Choose map visualization** dialog box, click **Bubble Map** and then click **Next**.
7. In the **Choose the analytical dataset** dialog box, click **SalesByRegion** and then click **Next**.
8. In the **Specify the match fields for spatial and analytical data** dialog box, click the check box in the **Match Fields** column next to STATENAME; in the **Select a field** drop-down list, select **State**, and then click **Next**.
9. In the **Choose color theme and data visualization** dialog box, clear **Use bubble sizes to visualize data**, check the **Use polygon colors to visualize data**; in the Data field select **[Sum(SalesAmount)]**, in the Color rule, select **Red-Yellow-Green**, and then click **Finish**.
10. Rearrange the map and tablix on the design surface so that the map is directly above the tablix.
11. Double-click **Map Title**, delete the text Map Title, and type **Sales by State**.
12. Select the map legend, and then press Delete.
13. Select the **Map Distance Scale**, and then press Delete.
14. Double-click the **Map Color Scale** and reposition in the bottom right of the map.
15. Click **Run** to preview the report.
16. Close Report Builder.

Module Review and Takeaways

Review Question(s)

Question: What considerations can you think of for including graphical elements in a report?

Answer: Answers will vary, depending on the students' experiences.

Module 4

Configuring Reports

Contents:

Lesson 1: Implementing Filters and Parameters	2
Lesson 2: Implementing Sorting and Grouping	5
Lesson 3: Publishing a Report	9
Module Review and Takeaways	11

Lesson 1

Implementing Filters and Parameters

Contents:

Question and Answers	3
Demonstration: Report Parameters	3

Question and Answers

Question: You want to filter report data. Where is the most efficient place to specify the filter?

- () On a row in a tablix.
- () On a detail group in a tablix.
- () On a category group in a chart.
- () On a data region.
- () On a dataset where clause.

Answer:

- () On a row in a tablix.
- () On a detail group in a tablix.
- () On a category group in a chart.
- () On a data region.
- (v) On a dataset where clause.

Demonstration: Report Parameters

Demonstration Steps

Add a Parameter Automatically

1. On the toolbar, click **Internet Explorer**.
2. In the address bar, type **mia-sql/Reports_SQL2**, and then press Enter.
3. Click the **Builder** folder.
4. On the **USA Sales by State** report, click the ellipsis (...) button, and then click **MANAGE**.
5. Click **Edit in Report Builder**.
6. In the **Internet Explorer** dialog, click **Allow**.
7. In the USA Sales by State - Microsoft SQL Server Report Builder window, in the **Report Data** pane, expand **Datasets**.
8. Right-click **SalesByRegion**, then click **Query**.
9. In the **Query Designer** dialog box, type the following query:

```
SELECT      DimGeography.EnglishCountryRegionName, DimGeography.StateProvinceName,
DimGeography.City, DimReseller.ResellerName, FactResellerSales.SalesOrderNumber,
FactResellerSales.OrderDate, FactResellerSales.SalesAmount,
DimDate.CalendarYear AS SalesYear
FROM
DimGeography INNER JOIN
DimReseller ON DimGeography.GeographyKey = DimReseller.GeographyKey INNER JOIN
FactResellerSales ON DimReseller.ResellerKey = FactResellerSales.ResellerKey INNER
JOIN
DimDate ON [DateKey] = FactResellerSales.OrderDateKey
WHERE DimDate.CalendarYear = @Year
```

10. Click **OK**.
11. In the Report Data pane, expand **Parameters**. Note that there is now a **Year** parameter.
12. On the **Home** tab, in the **Views** section, click **Run**.

13. In the **Year** box, type **2010**, and then click **View Report**.

Add Available Values to a Parameter

1. On the **Run** tab, in the **Views** section, click **Design**.
2. In the Report Data pane, right-click **Datasets**, then click **Add Dataset**.
3. In the **Name** box, type **AvailableYears**, then click **Use a dataset embedded in my report**.
4. In the **Data source** drop-down select **DataSource1**.
5. In the **Query** box, type:

```
SELECT Year(FactResellerSales.OrderDate) AS Year
FROM FactResellerSales
GROUP BY Year(FactResellerSales.OrderDate)
ORDER BY Year(FactResellerSales.OrderDate) DESC;
```

6. Click **OK**.
7. In the Report Data pane, in **Parameters**, right-click **Year**, then click **Parameter Properties**.
8. In the **Report Parameter Properties** dialog box, on the **Available Values** page, click **Get values from a query**.
9. In the **Dataset: (Warning: Possible performance impact)** drop-down, select **AvailableYears**.
10. In the **Value field** drop-down, select **Year**.
11. In the **Label field** drop-down, select **Year**.
12. On the **Default Values** page, click **Get values from a query**.
13. In the **Dataset: (Warning: Possible performance impact)** drop-down, select **AvailableYears**.
14. In the **Value field** drop-down, select **Year**, and then click **OK**.
15. On the **Home** tab, in the **Views** section, click **Run**.
16. Note that the report runs automatically and shows the figures for **2013**.

Lesson 2

Implementing Sorting and Grouping

Contents:

Question and Answers	6
Demonstration: Sorting and Grouping	6
Demonstration: Implementing Drillthrough Actions	7

Question and Answers

Question: You will mostly want to display data by country but, occasionally, you will want to expand a country and display the data for regions within a country. What is the most straightforward way to implement this?

- () Create a drillthrough action for each country that you might want to expand.
- () Create groupings by country and add a drilldown for the country column.
- () Sort by country and then by region.
- () Configure a page break for each country.

Answer:

- () Create a drillthrough action for each country that you might want to expand.
- (✓) Create groupings by country and add a drilldown for the country column.
- () Sort by country and then by region.
- () Configure a page break for each country.

Demonstration: Sorting and Grouping

Demonstration Steps

Add Groups to an Existing Report

1. On the taskbar, click **File Explorer**.
2. Navigate to **D:\Demofiles\Mod04**.
3. Double click **Reports Demo.sln**.
4. In the **Sorting and Grouping.rdl[Design]** design surface, click the **Preview** tab.
5. Note the table contains a row for each order in the **InternetSales** dataset.
6. In the **Sorting and Grouping.rdl[Design]** design surface, click the **Design** tab.
7. In the Row Groups pane at the bottom of the screen, right-click (**Details**), click **Add Group**, then click **Parent Group**.
8. In the **Tablix group** dialog box, in the **Group by** drop-down, select **[EnglishProductName]**.
9. Click **OK**.
10. In the Row Groups pane at the bottom of the screen, right-click **[EnglishProductName]**, click **Add Group**, then click **Parent Group**.
11. In the **Tablix group** dialog box, in the **Group by** drop-down, select **[EnglishProductSubcategoryName]**.
12. Click **OK**.
13. In the Row Groups pane at the bottom of the screen, right-click **[EnglishProductSubcategoryName]**, click **Add Group**, then click **Parent Group**.
14. In the **Tablix group** dialog box, in the **Group by** drop-down, select **[EnglishProductCategoryName]**.
15. Click **OK**.
16. In the **Sorting and Grouping.rdl[Design]** design surface, click the **Preview** tab.

17. Note the grouping that has been applied, and the extra fields that are no longer required.
18. In the **Sorting and Grouping.rdl[Design]** design surface, click the **Design** tab.
19. On the report, in the table, click the cell containing **Category**. Right-click the gray button above the **Category** cell.
20. Click **Delete Columns**.
21. Click the cell containing **Subcategory**. Right-click the gray button above the **Subcategory** cell.
22. Click **Delete Columns**.
23. Click the cell containing **Product**. Right-click the gray button above the **Product** cell.
24. Click **Delete Columns**.
25. In the **Sorting and Grouping.rdl[Design]** design surface, click the **Preview** tab.
26. Page through the report to show the grouped data.

Add Sorting to a Report

1. In the **Sorting and Grouping.rdl[Design]** design surface, click the **Design** tab.
2. In the Row Groups pane at the bottom of the screen, right-click **(Details)**, click **Group Properties**.
3. In the **Group Properties** dialog box, click **Sorting** in the left pane.
4. Click **Add**.
5. In the **Sort by** drop-down, select **[OrderDate]**. Click **OK**.
6. In the **Sorting and Grouping.rdl[Design]** design surface, right-click the top left cell containing **English Product Category Name**. Click **Text Box Properties**.
7. In the **Text Box Properties** dialog box, click **Interactive Sorting** in the left pane.
8. Click **Enable interactive sorting on this text box**.
9. Click **Groups**, and in the drop-down select **EnglishProductCategoryName**.
10. In the **Sort by** drop-down, select **[EnglishProductCategoryName]**. Click **OK**.
11. In the **Sorting and Grouping.rdl[Design]** design surface, click the **Preview** tab.
12. Note that the orders are sorted by Order Date, and that interactive sorting is available on product category.
13. Leave Visual Studio open for the next demonstration.

Demonstration: Implementing Drillthrough Actions

Demonstration Steps

Add a Drillthrough to a Summary Report

1. In the Reports Demo - Microsoft Visual Studio window, in the Solution Explorer pane on the right, double-click **Adding drillthrough actions.rdl**.
2. Right-click the bottom left cell in the table, and then click **Text Box Properties**.
3. In the **Text Box Properties** dialog box, in the left pane, click **Action**.
4. Click **Go to report**.
5. In the **Specify a report** field, in the drop-down, select **Drilldown report**.

6. Click **Add**.
7. In the **Name** drop-down, select **ReportParameter1**.
8. In the **Value** drop-down, select **[EnglishProductCategoryName]**.
9. Click **OK**.
10. In the **Adding drillthrough actions.rdl** design surface, click the **Preview** tab.
11. In the report, click **Accessories**, and note that the report drills down to a summary report for accessories.
12. In the **Adding drillthrough actions.rdl** design surface, click the **Design** tab.
13. On the design surface, right-click the cell containing **[SalesOrderNumber]**, and then click **Text Box Properties**.
14. In the **Text Box Properties** dialog box, in the left pane, click **Action**.
15. Click **Go to report**.
16. In the **Specify a report** field, in the drop-down, select **Order Detail**.
17. Click **Add**.
18. In the **Name** drop-down, select **SalesOrderNumber**.
19. In the **Value** drop-down, select **[SalesOrderNumber]**.
20. Click **OK**.
21. In the **Adding drillthrough actions.rdl** design surface, click the **Preview** tab.
22. Click **SO51380**, and note that an order detail report is shown for the clicked order.

Lesson 3

Publishing a Report

Contents:

Question and Answers	10
Demonstration: Publishing a Report to a SharePoint Site	10

Question and Answers

Question: When you are publishing a report to a SharePoint site with Report Builder, what needs to be specified?

- () Report server location
- () TargetReportFolder
- () TargetDataSourceFolder
- () TargetDatasetFolder

Answer:

- (√) Report server location
- () TargetReportFolder
- () TargetDataSourceFolder
- () TargetDatasetFolder

Demonstration: Publishing a Report to a SharePoint Site

Demonstration Steps

Publishing with Report Designer

1. On the taskbar, click **File Explorer**.
2. Navigate to **D:\Demofiles\Mod04**.
3. Double click **Reports Demo.sln**.
4. In the Reports Demo - Microsoft Visual Studio window, in the Solution Explorer pane on the right, right-click **Reports Demo**, then click **Properties**.
5. Ensure the **TargetDatasetFolder** is set to **http://mia-sql/sites/adventureworks/Reports/Datasets**.
6. Ensure the **TargetDataSourceFolder** is set to **http://mia-sql/sites/adventureworks/Reports/DataSources**.
7. Ensure the **TargetReportFolder** is set to **http://mia-sql/sites/adventureworks/Reports**.
8. Ensure the **TargetReportPartFolder** is set to **http://mia-sql/sites/adventureworks/Reports/ReportParts**.
9. Ensure the **TargetServerURL** is set to **http://mia-sql/sites/adventureworks**.
10. Click **OK**.
11. In the Reports Demo - Microsoft Visual Studio window, in the Solution Explorer pane on the right, right-click **Reports Demo**, then click **Deploy**.
12. Wait until the Deploy process has completed.
13. On the toolbar, click **Internet Explorer**.
14. In the address bar type **http://mia-sql/sites/adventureworks/Reports**.
15. Note that all the reports in the project have been deployed.

Module Review and Takeaways

It is very common for the requirements of businesses to continually change regarding the information they need, and how they want it to be presented. On viewing a detailed report, senior management will likely request higher level, summarized, and filtered versions of the same report.

Report Builder and Report Designer support these scenarios via filtering, sorting, drilldowns, grouping and the parameterizing of reports.

In this module, you have seen how it is possible to update existing reports to make them more dynamic and useful for business users.

Module 5

Create Mobile Reports

Contents:

Lesson 1: Overview of SQL Server Mobile Reports	2
Lesson 2: Preparing Data for Mobile Reports	5
Lesson 3: SQL Server Mobile Report Publisher	7
Module Review and Takeaways	13
Lab Review Questions and Answers	14

Lesson 1

Overview of SQL Server Mobile Reports

Contents:

Question and Answers	3
Demonstration: Mobile Reports in Action	3

Question and Answers

Question: True or false? When you follow the design-first approach to mobile report development, you configure your data sources before designing your report.

() True

() False

Answer:

() True

(v) False

Demonstration: Mobile Reports in Action

Demonstration Steps

1. Ensure that the 10990B-MIA-DC and 10990B-MIA-SQL virtual machines are both running, and then log on to 10990B-MIA-SQL as **ADVENTUREWORKS\Student** with the password **Pa\$\$w0rd**.
2. In the **D:\Demofiles\Mod05** folder, run **Setup.cmd** as Administrator.
3. If a **Visual Studio Just-In-Time Debugger** dialog box shows an error relating to w3wp.exe, click **No, cancel debugging**.
4. When setup steps are complete, open Internet Explorer, navigate to **mia-sql/Reports_SQL2**, click **MobileReports**, and notice the two KPIs in the folder.
5. In the **Sales Total KPI**, click the ellipsis (...), and then click **MANAGE**. Notice that the KPI is configured from a mixture of dataset data and manual data.
6. Click the browser **Back** button.
7. Click **Gauge Example**.

The report shows an example of the same data presented on various different gauge styles supported by mobile reporting. The subtitle of each gauge shows the gauge type.

8. Click the title bar of the **Total Sales KPI Linear gauge** report element. Notice that the element expands to fill the whole view. Click the title bar of the **Total Sales KPI Linear gauge** report element again to return to the default view. This facility makes it easier to view detailed charts on small screens.
9. Click the browser **Back** button.
10. Click **Simulated Example**.

The report shows various different report elements linked by a scorecard grid. The report uses simulated data.

11. In the **Scorecard grid 1** region of the report, click **Simulated Item A2**. Notice that the values of all the other reporting elements on the page change to reflect the subtotals for the simulated data item.
12. In the **Scorecard grid 1** region of the report, click **Simulated Item B1**. Notice that the other reporting elements again change to reflect your selection.
13. In the **Scorecard grid 1** region of the report, click **All** to return to the overall summary view.
14. Click the browser **Back** button.
15. Click **Time Navigator Example**.

The report shows a bar graph and a map linked to a time navigator element. The report will take a few seconds to open because it is based on data retrieved from the **AdventureWorks** and **AdventureWorksDW** databases.

16. Notice that the **All** section of the time navigator (the unlabeled element in the top left of the report) is selected. Notice that the other bar graph shows data by year.
17. In the time navigator click **2013**. Notice that the bar graph changes to show values by month. Also notice that the values on the **Sales by State** map update; the map now shows aggregated data for the previous year.
18. Click the browser **Back** button.
19. Click **Tree Map Example**.

The report shows a tree map filtered by a selection list. The report is based on a dataset embedded in the report. The size of each box in the tree map is determined by the value of the **Amount** field, relative to the other **Amount** values. The color of each box is determined by a comparison of **Amount** to **Delta** values.

20. On the left, click **Media** and demonstrate that the tree map adjusts to show data related to **Media**.
21. Close Internet Explorer.

Lesson 2

Preparing Data for Mobile Reports

Contents:

Question and Answers

6

Question and Answers

Question: How many datasets should you have per Excel worksheet, for use as a data source for a mobile report?

- ☐ One
- ☐ Two
- ☐ Any number

Answer:

- ☒ One
- ☐ Two
- ☐ Any number

Lesson 3

SQL Server Mobile Report Publisher

Contents:

Question and Answers	8
Demonstration: Adding a Dataset	8
Demonstration: Designing a Report	9
Demonstration: Defining Phone and Tablet Views	10
Demonstration: Publishing a Mobile Report	11

Question and Answers

Question: True or false? You use SSMRP to write queries that define datasets for a mobile report.

() True

() False

Answer:

() True

(√) False

Demonstration: Adding a Dataset

Demonstration Steps

Format Excel Data for Use in a Mobile Report

1. Using File Explorer, navigate to **D:\Demofiles\Mod05**, and then double-click **report_data.xlsx** to open the workbook in Excel. To make the data in the **Sheet1** worksheet suitable for use in mobile reports, each dataset be moved to a separate worksheet.
2. On the **Home** tab, in the **Cells** group, click the **Insert** drop-down arrow, and then click **Insert Sheet**. A new sheet called **Sheet2** is added to the workbook.
3. In **Sheet1**, select the cell range **F4:G11**, right-click the highlighted area, and then click **Copy**.
4. In **Sheet2**, click cell **A1**, right-click the highlighted area, and then click **Paste**.
5. Repeat steps 2 and 4 to copy the data in **Sheet1!A1:D36** to a new worksheet called **Sheet3**.
6. Save the workbook and close Excel.

Add a Dataset from Excel

1. On the Start page, type **Microsoft SQL Server Mobile Report Publisher**, and then press Enter. The application opens with an empty report.
2. On the **Data** tab, click **Add data**, and then click **Excel**.
3. In the **Open** dialog box, navigate to **D:\Demofiles\Mod05**, and then double-click **report_data.xlsx**.
4. In the Add data window, select the **Sheet1**, **Sheet2**, and **Sheet3** check boxes, and then click **Import**. The data from the three sheets is imported to SSMRP.
5. Click **Sheet1** and scroll right to demonstrate that all the data from the different regions on the worksheet has been imported, making it unsuitable for use in a mobile report.
6. On the **Sheet1** tab, click the **cog** icon, and then click **Remove**.

Update a Dataset from Excel

1. In File Explorer, navigate to **D:\Demofiles\Mod05**, and the double-click **report_data.xlsx**.
2. On **Sheet2**, click cell **B7**, type **Rebecca Green**, and then press Enter.
3. Save the workbook and close Excel.
4. In SSMRP, click **Sheet2**, then click **Refresh all data**. Notice that the dataset updates to reflect the changes you made in the Excel file.

Add a Dataset from an SSRS Shared Dataset

1. Click **Add data**, and then click **Report server**.
2. If the **Connect to a server** dialog box appears, in the **Server address** box, type **mia-sql/Reports_SQL2**.
3. Clear the **Use secure connection** check box, and then click **Connect**.
4. In the **Add data from server** dialog box, click **mia-sql/Reports_SQL2**, click **Builder**, and then click **SalesByRegionAndYear**. The result set will be imported.
5. Leave SSMRP open for the next demonstration.

Demonstration: Designing a Report

Demonstration Steps

1. In SSMRP, click **Settings**, in the **Report title** box, type **Regional Sales**.
2. On the **Layout** tab, click and drag a **Selection List** navigator from the gallery onto the top left cell of the design grid.
3. Grab the sizer and expand the selection list to fill two grids across and three down.
4. In the **Visual properties** region, in the **Title** box, type **Region Selector**.
5. On the **Data** tab, under **Report elements**, click **Region Selector**.
6. In the **Data properties** region, in the **Keys** list, click **Sheet2**.
7. In the **Filter these details when a selection is made** region, select the **Sheet2** and **Sheet3** check boxes.
8. In the **Sheet3** list, click **Region**.
9. On the **Layout** tab, notice that the contents of the selection list now reflect the values from the dataset.
10. Click and drag a **Simple data grid** from the gallery onto the first empty cell at the top left of the design grid to the right of the selection list.
11. Grab the sizer and expand the Data Grid to fill three grids across and three down.
12. In the **Visual properties** region, in the **Title** box, type **Sales Person**.
13. In the **Row numbers** list, click **Hide**.
14. On the **Data** tab, click **Sales Person**.
15. In the **Data properties** region, in the **Data for the grid view** list, click **Sheet2**.
16. On the **Preview** tab, in the **Region Selector** list, click on each region name and notice that the content of the **Sales Person** grid does not change, and then click the back button.
17. On the **Data** tab, click **Sales Person**.
18. In the **Data properties** region, click **Options**, select the **Region Selector** check box, and then click **Done**.
19. On the **Preview** tab, in the **Region Selector** list, click on each region name and notice that the content of the **Sales Person** changes according to the selected region, and then click the back button.

20. On the **Layout** tab, click and drag a **Radial gauge** from the gallery onto the first empty cell under the selection list in the design grid.
21. Grab the sizer and expand the gauge to fill two grids across and two down.
22. In the **Visual properties** region, in the **Title** box, type **Total Value**.
23. In the **Visual properties** region, click **Set ranges**.
24. In the **Neutral end** box, type **110.00 %**.
25. In the **Neutral start** box, type **90.00 %**, and then click **Done**.
26. On the **Data** tab, click **Total Value**.
27. In the **Data properties** region, in the **Main value** list, click **Sheet3**, and in the adjacent list, click **Actual**.
28. In the **Comparison value** list, click **Sheet3**, and in the adjacent list, click **Estimated**. Notice that the comparison value does not have to come from the same dataset as the main value.
29. On the **Layout** tab, on the design grid, click the **Total Value** gauge, click the cog icon at the top right of the gauge, and then click **Copy**.
30. Click the top left empty cell beneath the **Sales Person** data grid, click the cog icon at the top right of the cell, and then click **Paste**.
31. In the **Visual properties** region, in the **Title** box, type **Regional Value**.
32. On the **Data** tab, click **Regional Value**.
33. In the **Data properties** region, next to the **Main value** lists, click **Options**, ensure that **Region Selector** is selected, and then click **Done**.
34. Next to the **Comparison value** lists, click **Options**, ensure that **Region Selector** is selected, and then click **Done**.
35. On the **Preview** tab, click on each region name in the **Region Selector** list, and notice that the **Total Value** gauge always displays the same value (because it is not filtered by the selection list). By contrast, the **Regional Value** gauge changes to reflect the currently selected region, and then click the back button.
36. Leave SSMRP open for the next demonstration.

Demonstration: Defining Phone and Tablet Views

Demonstration Steps

Define Tablet View

1. In SSMRP, demonstrate that the view in which you have been working is the master view.
2. On the **Layout** tab, in the top right of the window, in the view drop-down list, click **Tablet**. Notice that the design grid is empty, and the element gallery has been replaced with the Control Instances gallery.
3. Click and drag the **Total Value** gauge from the gallery onto the top left cell of the design grid. Grab the sizer and expand the gauge to fill three grids across and three down.
4. Click and drag the **Regional Value** gauge from the gallery onto the first empty cell to the top right of the **Total Value** gauge. Grab the sizer and expand the gauge to fill three grids across and three down.

5. Click and drag the **Region Selector** list from the gallery onto the top left empty cell below the **Total Value** gauge. Grab the sizer and expand the selection list to fill six grids across and five down.
6. On the **Preview** tab, click the **Regional Value** text to expand the **Regional Value** gauge to fill the screen.
7. Click the **Regional Value** text to return to tablet view, and then click the back button to return to the **Layout** tab.

Define Phone View

1. On the **Layout** tab, in the top right of the window, in the view drop-down list, click **Phone**.
2. Customize the phone view by changing the value of the **Grid rows** slider to **5**.
3. Click and drag the **Region Selector** list from the gallery onto the top left cell of the design grid. Grab the sizer and expand the selection list to fill four grids across and one down.
4. Click and drag the **Total Value** gauge from the gallery onto the top left empty cell below the **Region Selector** list. Grab the sizer and expand the gauge to fill two grids across and two down.
5. Click and drag the **Regional Value** gauge from the gallery onto the first empty cell to the top right of the **Total Value** gauge. Grab the sizer and expand the gauge to fill two grids across and two down.
6. Click and drag the **Sales Person** DataGrid into the top left empty cell. Grab the sizer and expand the grid to fill four grids across and two down.
7. On the **Preview** tab, notice that the selection list is rendered as a drop-down box, and then click the back button to return to the **Layout** tab.
8. Leave SSMRP open for the next demonstration.

Demonstration: Publishing a Mobile Report

Demonstration Steps

Set Palette and Publish Report

1. On SSMRP, on the **Layout** tab, click the color palette selector, click the **Green** palette, and then click **Done**.
2. On the menu bar, click **Save mobile report as**.
3. In the **Save mobile report as** dialog box, click **Save to file system**.
4. In the **Save As** dialog box, in the **File name** box, type **D:\Demofiles\Mod05\Regional Sales.rsmobile**, and then click **Save**.
5. On the menu bar, click **Save mobile report as**.
6. In the **Save mobile report as** dialog box, click **Save to Server**, in the **Location** box, type **/MobileReports**, and then click **Save**.
7. Close SSMRP.

View a Published Report

1. In Internet Explorer, navigate to **http://mia-sql/Reports_SQL2**, click **MobileReports**, and then click **Regional Sales**. This is the master view of the report.
2. In File Explorer, navigate to **D:\Demofiles\Mod05**, right-click **browser_size.html**, point to **Open with** then click **Internet Explorer**.
3. In the message bar, click **Allow blocked content**.

4. In Internet Explorer, click **520x730 - tablet**. This starts a new browser window in a tablet aspect ratio.
5. In the new browser window, click **Regional Sales**. This is the tablet view of the report. Close the tablet view window.
6. Click **400x640 - phone**. This starts a new browser window in a phone aspect ratio. In the new browser window, click **Regional Sales**. This is the phone view of the report. Close all Internet Explorer windows.

Module Review and Takeaways

Question: Is mobile reporting important to your organization? How will SQL Server mobile reports change your approach to publishing reports?

Answer: Answers will vary.

Lab Review Questions and Answers

Lab: Working with Mobile Reports

Question and Answers

Lab Review

Question: Which type of visual component of mobile reports do you use to filter data?

- ☐ Gauges
- ☐ Navigators
- ☐ Graphs
- ☐ DataGrids

Answer:

- ☐ Gauges
- ☒ Navigators
- ☐ Graphs
- ☐ DataGrids