

SQL Server Reporting Services 2008

IGATE

Speed. Agility. Imagination

Copyright © 2011 IGATE Corporation. All rights reserved. No part of this publication shall be reproduced in any way, including but not limited to photocopy, photographic, magnetic, or other record, without the prior written permission of IGATE Corporation.

IGATE Corporation considers information included in this document to be Confidential and Proprietary.

## Document History

Date	Course Version No.	Software Version No.	Developer / SME	Change Record Remarks
01-08-2010	1.0	2008	Priya Rane	New course

## Course Goals and Non Goals

### ➤ Course Goals

- This course focuses on developing and organizing basic reports and building advance reports on Sql server 2008. We will be considering Report Designer, Report Builder, Model Designer for reporting.



### ➤ Course Non Goals

- This course will not be focusing on managing server components and installation.

## Pre-requisites

- Basic reports
- Purpose of reports

## Intended Audience

- Business analysts, Developers, Test Engineers, Designers etc.



## Day Wise Schedule

### ➤ Day 1

- Lesson 1: Reporting Life Cycle and Reporting architecture
- Lesson 2: Building reports

### ➤ Day 2

- Lesson 3: Working with Expressions
- Lesson 4: Report model

### ➤ Day 3

- Lesson 5: Report Builder-ad hoc report
- Lesson 7: Working with charts

## Table of Contents

➤ **Lesson 1: Reporting Life Cycle and Reporting architecture**

- 1.1. What is Reporting
- 1.2. Reporting Life Cycle
- 1.3. Reporting architecture and components
- 1.4 Report Designer - Business Intelligence Development Studio

➤ **Lesson 2: Building basic reports**

- 2.1. Server report
- 2.2. Data source
- 2.3. Data set
- 2.4. Data regions – table, matrix, list, charts
- 2.5. Adding existing report
- 2.6. Grouping data, summations on groups
- 2.7. Adjacent groups, textbox, image

## Table of Contents (Contd...)

- 2.8. Subreports
- 2.9. Formatting reports
- 2.10. Page break
- 2.11. Document map
- 2.12. Parameters (Prompt)
- 2.13. Report parameter values
- 2.14. Dataset for Report Parameter values
- 2.15. Query Parameter to Dataset
- 2.16. Built in fields

## Table of Contents (Contd...)

### ➤ Lesson 3: Working with Expressions

- 3.1. Adding calculations
- 3.2. Adding calculated fields in dataset
- 3.3. Text box expression
- 3.4. Report item expression
- 3.5. Built in fields
- 3.6. Aggregate functions
- 3.7. Scope of argument
- 3.8. Aggregate function in data region
- 3.9. Conditional formatting

## Table of Contents (Contd...)

### ➤ Lesson 7: Working with charts

- 4.1. Creating charts
- 4.2. Chart elements
- 4.3. Chart types – Column, line, bar, area, scatter
- 4.4. Creating Column charts
- 4.5. Formatting chart

## Table of Contents (Contd...)

### ➤ Lesson 5: Report Model

- 5.1. Creating model
- 5.2. Reviewing model
- 5.3. Deploying report on server

### ➤ Lesson 6: Building Ad hoc Reports

- 6.1. Creating ad hoc reports using Report Builder
- 6.2. Creating a table report
- 6.3. Creating matrix report
- 6.4. Creating chart reports
- 6.5. Formatting a report
- 6.6. Adding formulae
- 6.7. Filter the report
- 6.8. Sorting data

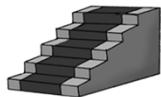
## References

- Microsoft Sql Server Reporting Services 2008



## Next Step Courses (if applicable)

- Administering reporting services



## Other Parallel Technology Areas

- Business Objects
- Cognos

# SQL Server Reporting Services 2008

Lesson 1: Introduction to SSRS 2008 and its Architecture

June 14, 2015

Proprietary and Confidential

- 1 -

**IGATE**  
Speed. Agility. Imagination.

## Lesson Objectives

- **What is Reporting Services (SSRS)?**
- **Reporting Services Architecture**
- **Reporting Life Cycle**



1.1: Reporting

## What is Reporting Services

- **SQL Server 2008 Reporting Services (SSRS)** is a server-based reporting platform that provides comprehensive reporting functionality for a variety of data sources.
- **Reporting Services** includes a complete set of tools for you to create, manage, and deliver reports
- It also provide APIs that enable developers to integrate or extend data and report processing in custom applications

June 14, 2015

Proprietary and Confidential

- 3 -

**IGATE**  
Speed. Agility. Imagination

Microsoft SQL Server 2008 Reporting Services (SSRS) provides a full range of ready-to-use tools and services to help you create, deploy, and manage reports for your organization, as well as programming features that enable you to extend and customize your reporting functionality.

People use reports to communicate information, make decisions, and identify opportunities. SQL Server 2008 Reporting Services (SSRS) is a server-based reporting platform that provides a full range of ready-to-use tools and services to help people throughout your organization create, deploy, manage, and use reports quickly and easily. With SQL Server 2008 Reporting Services (SSRS), you can retrieve data from relational, multidimensional, and XML-based data sources; publish reports that can be viewed in various formats; and centrally manage report security and subscriptions. The reports that you create can be viewed over a Web-based connection or as part of a Microsoft Windows application or SharePoint site.

With Reporting Services, you can create interactive, tabular, graphical, or free-form reports from relational, multidimensional, or XML-based data sources. You can publish reports, schedule report processing, or access reports on-demand. Reporting Services also enables you to create ad hoc reports based on predefined models, and to interactively explore data within the model. You can select from a variety of viewing formats, export reports to other applications, and subscribe to published reports. The reports that you create can be viewed over a Web-based connection or as part of a Microsoft Windows application or SharePoint site. Reporting Services provides the key to your business data.

1.1: Reporting

## What is Reporting Services

- With Reporting Services, you can create interactive, tabular, graphical, or free-form reports from relational, multidimensional, or XML-based data sources.
- You can publish reports, schedule report processing, or access reports on-demand.

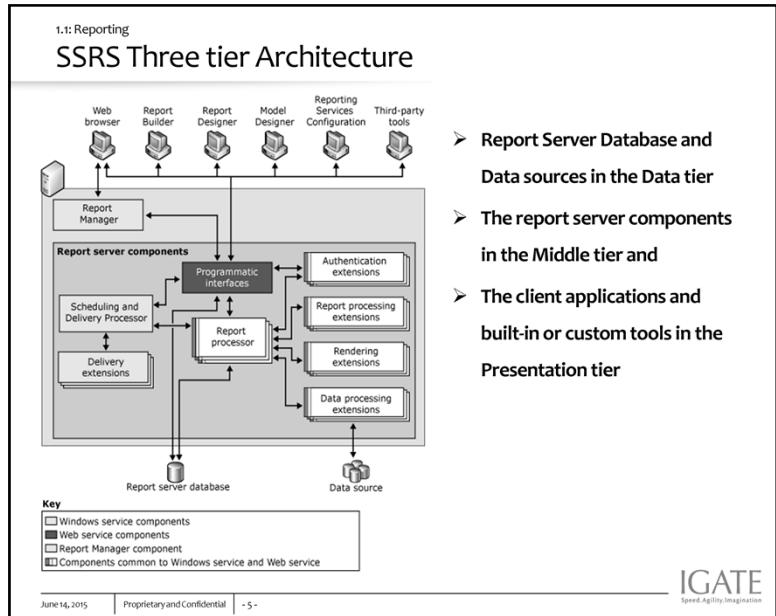
June 14, 2015

Proprietary and Confidential

- 4 -

**IGATE**  
Speed. Agility. Imagination

Microsoft SQL Server 2008 Reporting Services (SSRS) provides a full range of ready-to-use tools and services to help you create, deploy, and manage reports for your organization, as well as programming features that enable you to extend and customize your reporting functionality



In Reporting Services, a report server is implemented as a Windows service that consists of distinct feature areas that run in separate application domains. The service hosts Report Manager, the Report Server Web service, and background processing feature areas.

**The Report Server includes three applications that run in the same Windows service:**

1. Report Manager, which is represented by the Report Manager component in the diagram. Report Manager is a browser application that provides front-end access to the Reporting Services Web service.
2. Reporting Services Web service, which is represented by the Web service components in the diagram. This application handles on-demand, interactive report processing.
3. Background processing application, which is represented by the Windows service components in the diagram. This application processes reports that are triggered from a schedule and delivers reports to target destinations.

**Report Manager**

Report Manager is a zero-footprint client that provides Web front-end access to the Report Server Web service. It is the out-of-the-box tool for viewing and managing report server content and operations. By default, it provides front-end access to the Web service that runs in the same server instance. If the Web service is not enabled in the server instance, you can point Report Manager to a Report Server Web service in a different instance or computer by setting a URL in the configuration files. Report Manager runs within a browser session on the client computer. There are no application files or settings that are stored on the client. Session state is preserved as long as the browser window is open. User-specific settings are saved to the report server database and re-used whenever the user connects to Report Manager. Report Manager can accommodate custom delivery extension settings in the subscription definition pages.

To use Report Manager, you must define a URL to the application. You can effectively disable Report Manager by not creating the URL in the first place. If you installed Reporting Services in the default configuration, the URL is already created and you must delete it if you decide to turn the application off.

If you configure the report server to run in SharePoint integrated mode, Report Manager is turned off. You cannot use Report Manager on a report server that runs in SharePoint integrated mode, even if you previously configured the URL.

**Report Server Web service**

SQL Server Reporting Services provides access to the full functionality of the report server through the Report Server Web service. The Report Server Web service is an XML Web service with a SOAP API. It uses SOAP over HTTP and acts as a communications interface between client programs and the report server. The Web service provides two endpoints - one for report execution and one for report management - with methods that expose the functionality of the report server and enable you to create custom tools for any part of the report life cycle. The Report Server Web service is the core engine for all on-demand report and model processing requests that are initiated by a user or application in real-time, including most requests that are directed to and from Report Manager. The Report Server Web service performs end-to-end processing for reports that run on demand. To support interactive processing, the Web service authenticates the user and checks the authorization rules prior to handing a request. The Web service supports default Windows security extension and custom authentication extensions. The Web service is also the primary programmatic interface for custom applications that integrate with the report server. If you are providing a custom user interface, you can use the Web service without Report Manager.

**Background processing**

Background processing refers to operations that run in the background and are initiated by the report server. Most background processing consists of scheduled report processing and subscription delivery, but it also includes report server database maintenance tasks. Background processing for scheduling, subscription, and delivery is configurable and can be turned off through the Surface Area Configuration for Reporting Services facet of Policy-Based Management in Management Studio. If you turn those operations off, scheduled report or model processing will not be available in the current service instance. Database maintenance is a core task that cannot be turned off as it keeps the server in a working state.

Background processing operations depend on a front-end application or the Web service for definition. Specifically, schedules and subscriptions are created in the application pages of Report Manager or on a SharePoint site if the report server is configured for SharePoint integration, and then forwarded to the Web service, which creates and stores the definitions in the report server database.

**Data Storage:**

The report server is a stateless server that stores all properties, objects, and metadata in a SQL Server database. Stored data includes published reports, report models, and the folder hierarchy that provides the addressing for all items managed by the report server. A report server database can provide internal storage for a single Reporting Services installation or for multiple report servers that are part of a scale-out deployment.

1.1: Reporting

## SSRS Three tier Architecture

June 14, 2015

Proprietary and Confidential

- 6 -

**IGATE**  
Speed. Agility. Imagination**Extensions:-**

The report server supports custom authentication extensions, data processing extensions, report processing extensions, rendering extensions, and delivery extensions. A report server requires at least one authentication extension, data processing extension, and rendering extension. Delivery and custom report processing extensions are optional, but necessary if you want to support report distribution or custom controls.

Security extensions are used to authenticate and authorize users and groups to a report server. The default security extension is based on Windows authentication. You can also create a custom security extension to replace default security if your deployment model requires a different authentication approach.

Data Processing extensions are used to query a data source and return a flattened row set. Reporting Services uses different extensions to interact with different types of data sources. You can use the extensions that are included in Reporting Services, or you can develop your own extensions. Data processing extensions for SQL Server, Analysis Services, Oracle, SAP NetWeaver Business Intelligence, Hyperion Essbase, Teradata, OLE DB, and ODBC data sources are provided. Reporting Services can also use any ADO.NET data provider. Data processing extensions process query requests from the Report Processor component by performing the following tasks:

Rendering extensions transform data and layout information from the Report Processor into a device-specific format. Reporting Services includes seven rendering extensions: HTML, Excel, CSV, XML, Image, PDF, and Microsoft Word.

Report processing extensions can be added to provide custom report processing for report items that are not included with Reporting Services. By default, a report server can process tables, charts, matrices, lists, text boxes, images. If you want to add special features to a report that require custom processing during report execution (for example, if you want to embed a Microsoft MapPoint map), you can create a report processing extension to do so.

The background processing application uses delivery extensions to deliver reports to various locations. Reporting Services includes an e-mail delivery extension and a file share delivery extension. The e-mail delivery extension sends an e-mail message through Simple Mail Transport Protocol (SMTP) that includes either the report itself or a URL link to the report.

1.1: Reporting

## SSRS Reporting Life Cycle

### ➤ Authoring:- Is about creating and developing SSRS reports.

- The Report Designer in BIDS can be used to create different types of enterprise reports such as group, crosstabs, charts, free form, drill down, drill through and interactive reports.
- The Report Builder that can be used by Non developers or end users to create reports.

June 14, 2015

Proprietary and Confidential

- 7 -

**IGATE**  
Speed. Agility. Imagination

## Authoring

As a report author, with RS you have 2 choices for creating reports. The Report Designer and the Report Builder application. For advanced report authoring, the Report Designer will likely be your tool of choice. The Report Builder application is an Office-like application that is used to create on-the-fly reports.

1.1: Reporting

## SSRS Reporting Life Cycle

- **Management:** RS facilitates report management by storing reports and their related items in a central Report Catalog. To deploy and manage a report, you need to upload it to the Report Catalog. When this happens, it becomes a managed report.
- **Delivery:** Reports hosted under RS can be delivered using on-demand (“pulled”) delivery or subscribed (“pushed”) delivery.

June 14, 2015

Proprietary and Confidential

- 8 -

**IGATE**  
Speed. Agility. Imagination

### Managing

What really happens when a report is uploaded to the report catalog. At publishing time, the Report Server parses the report definition (RDL), generates a .NET assembly, and stores the assembly in the Report Configuration Database for the report. The RDL file is never used again. When the report is processed, the assembly is loaded and executed by the Report Server. A report can include other items, such as images and data source-related information. These report-related items are also stored in the report catalog. For example, just as you can organize physical files in folders, RS allows us to organize reports in folders like structure which can be accessed using Report Manager Web Application.

RS offers centralized report management that administrators will appreciate. To simplify the administration of the report catalog, RS comes with a tool called the Report Manager. The Report Manager is implemented as a web-based application, and as such it is easily accessible. This tool empowers you to manage just about any aspect of the report repository, including:

- Report information and metadata, such as the folder structure and report properties
- Data sources from which the report will draw data
- Report parameters (for parameterized reports)
- Security

### Delivery

In Reporting Services, two methods are available for accessing and delivering reports:

1. On Demand Reporting (Pull)
2. Subscribed Reporting (Push)

The more common scenario is on-demand delivery, where the user requests the report explicitly.

On-demand access allows users to select the reports from a report viewing tool. You can use Report Manager, a Microsoft SharePoint Web part, an embedded ReportViewer control, or a browser

The “pushed” delivery option alone can justify implementing RS. This option provides end users with the ability to subscribe to reports, so reports will be sent to them when a certain event is triggered—when a timing event triggers, for instance, report subscriptions based on a schedule. As another example, a financial institution could allow its customers to opt in and subscribe to certain reports of interest, such as a monthly bank statement. Then, at the end of the month, the bank statement report could be generated and sent to users via e-mail.

1.1: Reporting

## Report Management using Report Manager

The screenshot shows the Microsoft Internet Explorer browser displaying the Report Manager. The address bar shows the URL: <http://localhost/Reports/Pages/Folder.aspx?ItemPath=%2fAWReporter&ViewMode=List>. The page title is "Report Management using Report Manager". The left sidebar shows a tree view of report folders: "Corporate Hierarchy", "Employee Performance", "Employee Sales By Territory", "Employee Sales By Territory with Summary", "Employee Sales By Territory with Summary Advanced", "Employee Sales By Territory with Summary Chart", "Employee Sales Freeform", "Employee Sales Freeform Secured", and "Employee Sales Freeform with Chart". The right pane lists reports under "AWC.jpg": "Inventory Levels", "Monthly Order Summary", "Monthly Sales by Product Category", "Products by Category", "Purchase Orders", "Sales By Territory", "Sales By Territory with Grouping Interactive", and "Salesperson Summary". At the bottom, there is a footer with the date "June 14, 2015", the text "Proprietary and Confidential", and a page number "- 9 -". The watermark "IGATE Speed. Agility. Imagination" is visible on the right.

### Managing

What really happens when a report is uploaded to the report catalog. At publishing time, the Report Server parses the report definition (RDL), generates a .NET assembly, and stores the assembly in the Report Configuration Database for the report. The RDL file is never used again. When the report is processed, the assembly is loaded and executed by the Report Server. A report can include other items, such as images and data source-related information. These report-related items are also stored in the report catalog. For example, just as you can organize physical files in folders, RS allows us to organize reports in folders like structure which can be accessed using Report Manager Web Application.

RS offers centralized report management that administrators will appreciate. To simplify the administration of the report catalog, RS comes with a tool called the Report Manager. The Report Manager is implemented as a web-based application, and as such it is easily accessible. This tool empowers you to manage just about any aspect of the report repository, including:

- Report information and metadata, such as the folder structure and report properties
- Data sources from which the report will draw data
- Report parameters (for parameterized reports)
- Security

### Delivery

In Reporting Services, two methods are available for accessing and delivering reports:

1. On Demand Reporting (Pull)
2. Subscribed Reporting (Push)

The more common scenario is on-demand delivery, where the user requests the report explicitly.

On-demand access allows users to select the reports from a report viewing tool. You can use Report Manager, a Microsoft SharePoint Web part, an embedded ReportViewer control, or a browser

The “pushed” delivery option alone can justify implementing RS. This option provides end users with the ability to subscribe to reports, so reports will be sent to them when a certain event is triggered—when a timing event triggers, for instance, report subscriptions based on a schedule. As another example, a financial institution could allow its customers to opt in and subscribe to certain reports of interest, such as a monthly bank statement. Then, at the end of the month, the bank statement report could be generated and sent to users via e-mail.

1.1: Reporting

## Requesting Report (On Demand)

Employee Name:	Tsoufias, Lynn	Product Subcategory	Sales	# Orders
Commission:	1.80 %	Touring Bikes	\$502,322.95	295
Sales YTD:	\$1,758,385.93	Mountain Bikes	\$85,481.50	26
Bonus:	\$5,650.00	Touring Frames	\$64,368.53	73
		Mountain Frames	\$21,512.07	26
		Jerseys	\$9,178.24	60

IGATE Speed. Agility. Imagination

### Managing

What really happens when a report is uploaded to the report catalog. At publishing time, the Report Server parses the report definition (RDL), generates a .NET assembly, and stores the assembly in the Report Configuration Database for the report. The RDL file is never used again. When the report is processed, the assembly is loaded and executed by the Report Server. A report can include other items, such as images and data source-related information. These report-related items are also stored in the report catalog. For example, just as you can organize physical files in folders, RS allows us to organize reports in folders like structure which can be accessed using Report Manager Web Application.

RS offers centralized report management that administrators will appreciate. To simplify the administration of the report catalog, RS comes with a tool called the Report Manager. The Report Manager is implemented as a web-based application, and as such it is easily accessible. This tool empowers you to manage just about any aspect of the report repository, including:

- Report information and metadata, such as the folder structure and report properties
- Data sources from which the report will draw data
- Report parameters (for parameterized reports)
- Security

### Delivery

In Reporting Services, two methods are available for accessing and delivering reports:

1. On Demand Reporting (Pull)
2. Subscribed Reporting (Push)

The more common scenario is on-demand delivery, where the user requests the report explicitly. On-demand access allows users to select the reports from a report viewing tool. You can use Report Manager, a Microsoft SharePoint Web part, an embedded ReportViewer control, or a browser.

The “pushed” delivery option alone can justify implementing RS. This option provides end users with the ability to subscribe to reports, so reports will be sent to them when a certain event is triggered—when a timing event triggers, for instance, report subscriptions based on a schedule. As another example, a financial institution could allow its customers to opt in and subscribe to certain reports of interest, such as a monthly bank statement. Then, at the end of the month, the bank statement report could be generated and sent to users via e-mail.

## Summary

- **What is Reporting Services (SSRS)?**
- **Reporting Services Architecture**
- **Reporting Life Cycle**



June 14, 2015

Proprietary and Confidential

- 11 -

**IGATE**  
Speed. Agility. Imagination

# SQL Server Reporting Services 2008

## Lesson 2: Authoring Reports

June 14, 2015

Proprietary and Confidential

- 1 -

IGATE  
Speed. Agility. Imagination.

## Lesson Objectives

➤ **Authoring various types of reports**

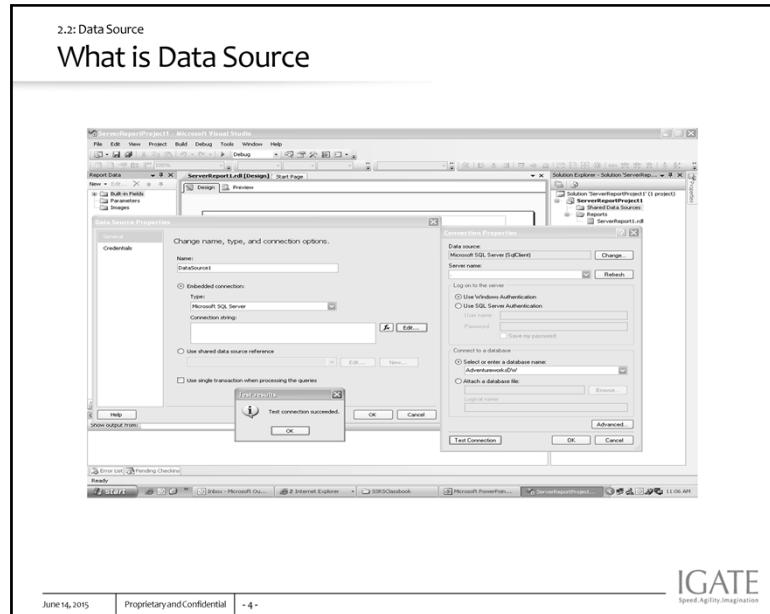
- Data Source
- Data Sets
- Data Regions
- Grouping
- Summation



2.1: Reporting Authoring

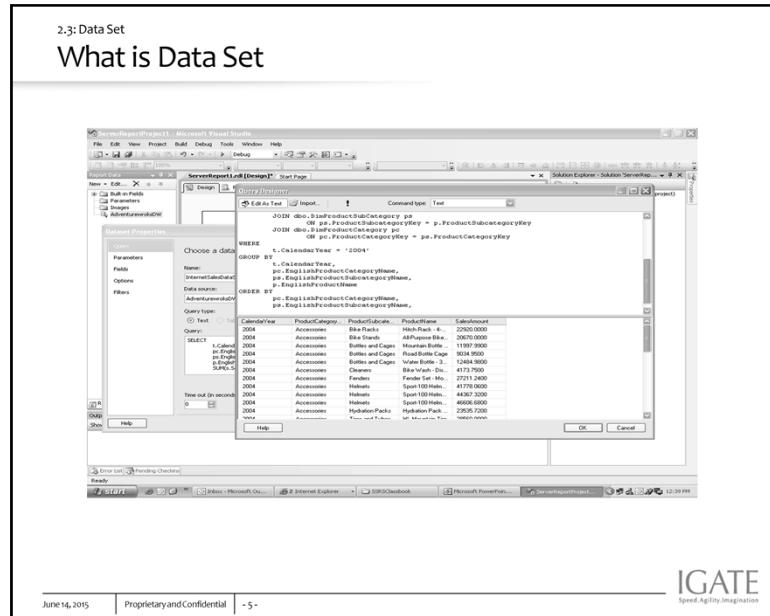
## Report Authoring using Report Designer

- We can create different types of reports using SSRS report designer.
  - Tabular Report
  - Group Report
  - Drill down Report
  - Drill through Report
  - Free Form Report
  - Matrix Report
  - Charts Report
  - Hierarchical Group Report



Data source is the information getting displayed for report. It provides data for executing queries to the report processor like connection string, authentication details, etc. To add new data source in Report project:

1. In Report Data pane, click New, select Data Source.
2. Name data source as **AdventureWorksDS**.
3. Select embedded connection for Microsoft Sql Server.
4. In Connection strings. Select Edit button that opens Connection Properties window.
5. Select server name and select database name.
6. Test the connection. If it gets successfully tested, click on OK.



Once data source is created, we need to add Data set for retrieving data for report. A dataset is a container that provides data values to Report processor.

To add a dataset:

1. In Report Data pane, click New, and select Dataset.
  2. Name the dataset.
  3. Import Query Designer, write query for fetching details. Click on Run button to ensure that query returns values.
  4. Click OK to close Query Designer window.
  5. Dataset appears below Datasource.

2.4: Data regions

## Data regions

- A data region displays report.
- Four data regions: tables, matrixes, lists, charts
- Table: displays data in tabular format
- Matrix: dynamic rows and columns, crosstab or pivot table
- List: flexible layout, combination of text boxes and other matrix
- Charts: graphical representation of data.

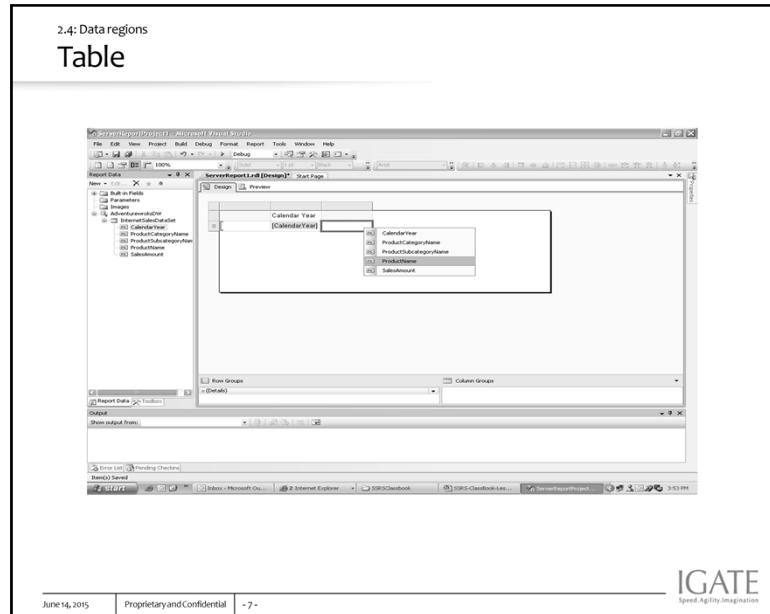
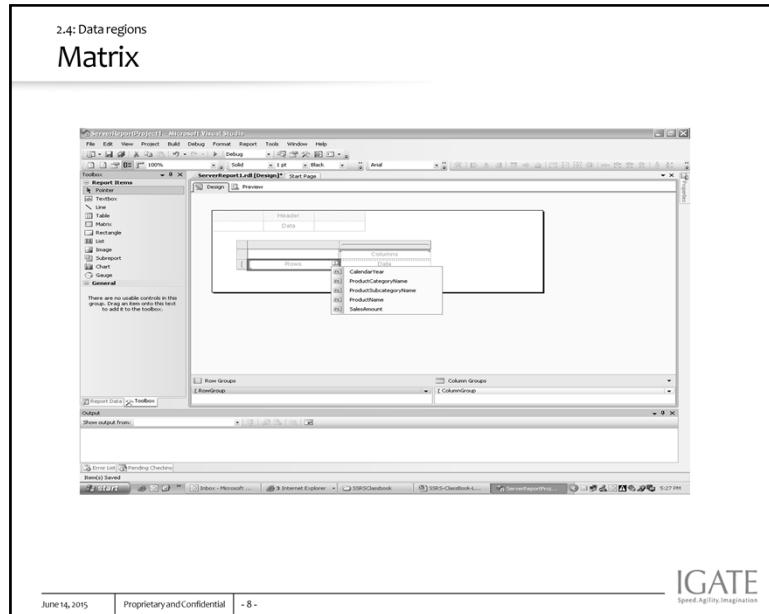


Table has rows and columns combination. It has fixed length depends on rows returned by dataset query. You can specify header for the columns or footer for aggregated values. To add table to report:

1. Go to Toolbox. Double click on Table.

To add fields to table:

1. Open Report Data window.
2. Drag any column field listed into data tab. By adding field value to column we bound dataset to table.
3. Click on Preview tab to view the report.
4. Close the output window.



Unlike tables, matrix data region has dynamic rows and columns.

It displays data like crosstab or pivot table.

To create matrix report:

1. Go to Toolbox. Click on Matrix.
2. Drag and drop data fields from data set in columns.

2.4: Data regions

## Lists

- A list displays report data in more arranged manner.
- It is flexible layout than other layouts.
- List helps to combine different controls and present data in structured way.

June 14, 2015

Proprietary and Confidential

- 9 -

IGATE  
Speed. Agility. Imagination

List is more flexible layout than matrix. You can create layout as a combination of text boxes and other data regions.

To display text box and a matrix in a list:

1. Drag a matrix control on design tab.
2. Associate it with data.
3. Drag List control from Toolbox window on matrix area.
4. Right click in matrix, Click Tablix2 to select matrix. Drag matrix into list.
5. Right click in list, select Tablix3b to select list.
6. In properties window for list, select DatasetName as created dataset.
7. In Grouping pane, right click Details1 group, and click Group Properties. Click Add.
8. In Group On drop-down list, select group criteria and click OK.
9. Drag a text box from Toolbox and place it above matrix. Bind data to it.
10. Save and review the report.

2.4: Data regions

## Charts

- Charts are visual presentation of summarized data.
- It gives graphical or pictorial view of data for analysis
- Charts provide better understanding of data in the form of graphs.
- Reports can be created using only charts or can be merged with statistical reports.
- User can insert chart inside table, matrix or list.

2.5: Existing report

## Adding existing report

- User can add a new report by right clicking the Reports folder, select Add, click New Item.
- User can add a new report by right clicking the Reports folder, select Add, click Existing Item.

2.6: Grouping data, summations on groups

## Grouping data

- Grouping combines data as per specifications
- Group allows to gather information based on certain requirements.
- Grouping of values helps even in performing aggregation on data and plays important role in calculations.
- Data can be grouped on multiple conditions.
- Separate properties for multiple groups are available.

June 14, 2015

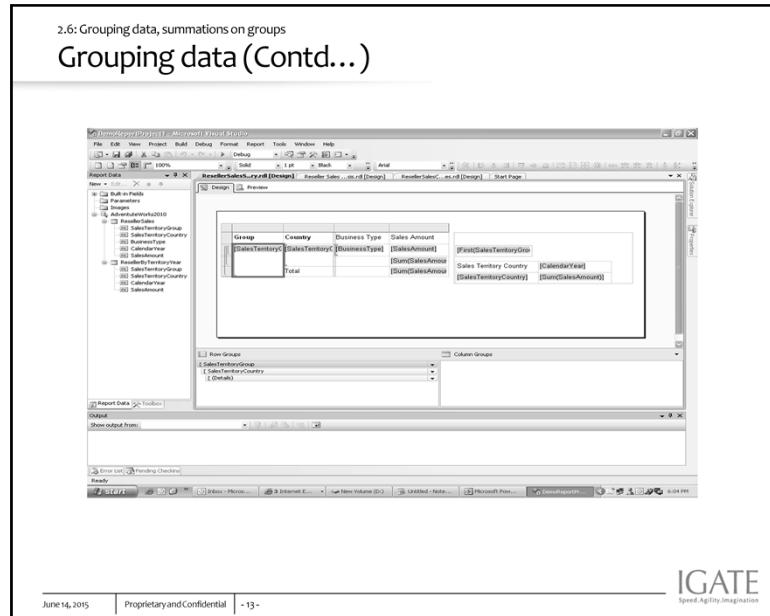
Proprietary and Confidential

- 12 -

IGATE  
Speed. Agility. Imagination

To Group data in table:

1. Click on Design tab.
2. Right click second row of the first column, select Add group.
3. In Row group, select Parent Group. There is Column Group option as well.
4. In the Tablix Group dialog box, select Group criteria for one of the row values.
5. Click OK to close the Tablix Group dialog box.
6. Right click the second row of the first column, select Add Group, in Row Group section, click Child Group.
7. Click the first row of the first column, highlight Group1, name the Group.
8. Click the first row of the second column, highlight Group2, and name the Group.
9. In Grouping pane, right click Group1, click Group Properties.
10. Select Group criteria.
11. Click OK to close Group properties dialog box.
12. In Grouping pane, right click Group2, click Group Properties.
13. Select Group criteria.
14. Click OK to close Group properties dialog box.
15. Save the project
16. Preview the report.



2.7: Adjacent groups, textbox, image

## Defining Adjacent Groups

- Adjacent groups allows to group data for adjacent columns.
- Adjacent grouping allows to group data that is already grouped on other groups.
- Data can be grouped on adjacent left or right column of the selected column.
- User can group data on multiple group criteria like user can display year wise sum of sales revenue for cities as well can adjacently display sum of sales revenue for accessories horizontally as an adjacent column.

June 14, 2015

Proprietary and Confidential

- 14 -

IGATE  
Speed. Agility. Imagination

### Display yearly territory group wise sum of sales amount and adjacent to it accessories wise territory group wise sum of sales amount.

Steps to add adjacent groups:

1. Go to Design tab.
2. Report is getting displayed for yearly territory group wise sum of sales amount.
3. Right click Year column, Add Group, Adjacent right.
4. In Tablix Group dialog box, Group by select Accessories and click Sales Amount.
5. Save and preview the report.
6. Report gets displayed for sum of Sales Amount for Territory Group for two column categories, one for yearly sum and another for category wise sum.

2.7: Adjacent groups, textbox, image

### Defining Adjacent Groups (Contd...)

The screenshot shows the Microsoft Visual Studio interface with the Report Designer open. The report is titled 'Reseller Sales.rdl' and displays a table of sales data. The table has four columns for years (2001, 2002, 2003, 2004) and four rows for product categories (Accessories, Bikes, Clothing, Components). The data is grouped by Sales Territory Group (Europe, North America, Pacific). The Visual Studio interface includes toolbars, menus, and various windows like the Error List and Pending Checkins.

**Display yearly territory group wise sum of sales amount and adjacent to it accessories wise territory group wise sum of sales amount.**

2.6: Grouping data, summations on groups

## Summations on groups

- Tables can be used to add totals for the column.
- It will be added as a new row to display summation for the entire column.
- It is helpful to sum data based on groups.
- Default Add Total command, uses Sum() function.
- We can use other aggregate functions like Average() and Max(), etc.

June 14, 2015

Proprietary and Confidential

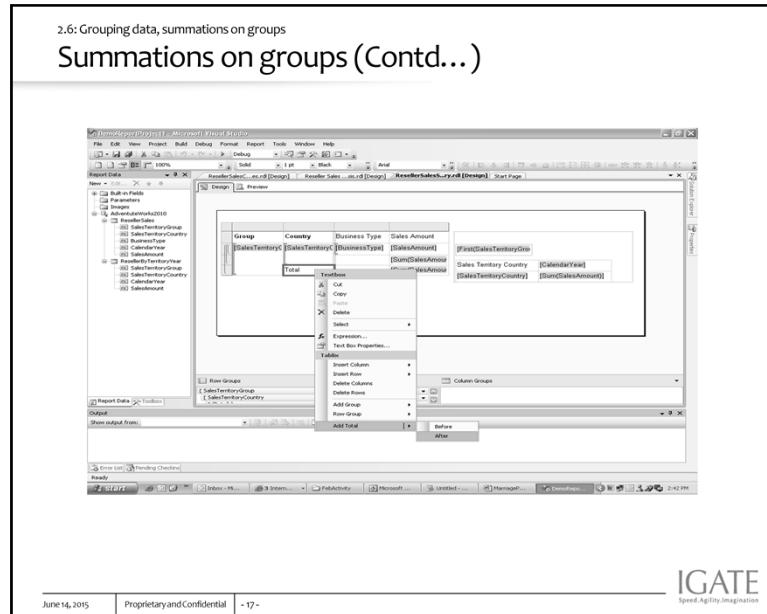
- 16 -

**IGATE**  
Speed. Agility. Imagination

### Display yearly sum of sales Amount

To add total to row Groups:

1. Click on Design tab.
2. Right click text box containing numeric field like SalesAmount and click Add Total.
3. New row appears as a group footer row for the group.
4. Save the report.



The screenshot shows the 'Tabular Report' design view in the Reporting Services Authoring environment. At the top, there's a toolbar with various icons for report navigation and management. Below the toolbar is a header section with the title 'Tabular Report'. The main area contains a table with three columns: 'Product ID', 'Product Name', and 'Category ID'. The data in the table is as follows:

Product ID	Product Name	Category ID
1	Chai	1
2	Chang	1
24	Guaraná Fantastica	1
34	Sasquatch Ale	1
35	Steeleye Stout	1
38	Côte de Blaye	1
39	Chartreuse verte	1
43	Ipoh Coffee	1
67	Laughing Lumberjack Lager	1
70	Outback Lager	1

At the bottom of the report area, there are three status bars: 'June 14, 2015', 'Proprietary and Confidential', and '- i8 -'. In the bottom right corner of the window frame, the IGATE logo is visible.

Usually whenever we design a report, it has fixed number of columns. For example, if we design a report showing a list of all employees, we may choose Employee ID, First Name, Last Name, Email Address, Mobile No. and Gender columns. So here, we know that how many columns we are going to show in a report, and, we fix those columns in the report during design time.

However, the numbers of rows vary and this report could give us one page or more than multiple pages output, but design do not change. It means that columns are always fixed and rows vary depending upon volume of data. This type of report is known as a **Tabular report**.

1.1: Reporting

## Tabular Report with Expressions

**Inventory Products Listing**

Product ID	Product Name	Unit Price	Units In Stock	Stock Value
1	Chai	18.0000	39	702.0000
2	Chang	19.0000	17	323.0000
39	Chartreuse verte	18.0000	69	1242.0000
38	Côte de Blaye	263.5000	17	4479.5000
24	Guaraná Fantástica	4.5000	20	90.0000
43	Ipoh Coffee	46.0000	17	782.0000
76	Lakkalikööri	18.0000	57	1026.0000
67	Laughing Lumberjack Lager	14.0000	52	728.0000
70	Outback Lager	15.0000	15	225.0000
75	Rhönbräu Klosterbier	7.7500	125	968.7500
34	Sasquatch Ale	14.0000	111	1554.0000
35	Steeleye Stout	18.0000	20	360.0000
				Total Stock Value
				12480.2500

=Fields!UnitsInStock.Value  
\* Fields!UnitPrice.Value

=Sum(Fields!UnitsInStock.Value\*Fields!UnitPrice.Value)

June 14, 2015 | Proprietary and Confidential | + 19 -

IGATE  
Speed. Agility. Imagination.

In Reporting Services, expressions are used throughout the report definition to specify or calculate values for parameters, queries, filters, report item properties, group and sort definitions, text box properties, bookmarks, document maps, dynamic page header and footer content, images, and dynamic data source definitions.

Expressions begin with an equal (=) and are written in Microsoft Visual Basic. Expressions can include a combination of constants, operators, and references to built-in values (fields, collections, and functions), and to external or custom code. Expressions can be one of the following two types:

**Simple** An expression that is a single reference to an item in a built-in collection, such as, a dataset field, a parameter, or a built-in field. Simple expressions appear on the design surface and in dialog boxes in brackets, such as [FieldName], which represents the underlying expression =Fields!FieldName.Value. You can type simple expressions directly into a text box on the design surface and the corresponding expression text is set as the value of a placeholder inside the text box. For more information, see [Formatting Text](#) and [Importing HTML](#).

**Complex** An expression that includes more than a simple reference. Complex expressions appear on the design surface as <>Expr<>. You can create complex expressions in the **Expression** dialog box or type them directly into the **Property** pane. For more information about what you can include in an expression

When you write an expression in Reporting Services, you have access to many built-in fields, built-in collections, and functions that you can use alone or combine with other terms. When you create an expression interactively in the **Expression** dialog box, you can explore the categories of references that you can include, and see context-sensitive examples of constants, built-in collections, and functions available for including in your expressions.

1.1: Reporting

## Group Report

Inventory Products Listing

Category	Product ID	Product Name	Unit Price	Units In Stock	Stock Value
Beverages	1 Chai		18.0000	39	702.0000
	2 Chang		19.0000	17	323.0000
	39 Chartreuse verte		18.0000	69	1242.0000
	38 Côte de Blaye		283.5000	17	4479.5000
	24 Guarán Fénix tica		4.5000	20	90.0000
	43 Ipan Coffee		46.0000	17	782.0000
	76 Lakritzbröd		18.0000	57	1025.0000
	67 Laughing Lumberjack Lager		14.0000	52	728.0000
	70 Outback Lager		15.0000	15	225.0000
	75 Rhönbräu Klar terbier		7.7500	125	968.7500
	34 Sasquatch Ale		14.0000	111	1554.0000
	35 Steeleeye Stout		18.0000	20	360.0000
					Total Stock Value for Category
Condiments	3 Aniseed Syrup		10.0000	13	130.0000
	4 Chef Anton's Cajun Seasoning		22.0000	63	1166.0000
	5 Chef Anton's Gumbo Mix		21.3500	0	0.0000
	15 Genen Shouyu		15.5000	39	604.5000
	6 Grandma's Boysenberry Spread		25.0000	120	3000.0000

June 14, 2015 | Proprietary and Confidential | - 20 -

IGATE  
Speed. Agility. Imagination.

In Report Designer, we can use groups to organize data on the report or to calculate aggregate summaries

In Report Designer, a group is a named set of data from the report dataset that is bound to a data region. Basically, a group organizes a view of a report dataset. All groups in a data region specify different views of the same report dataset.

A group has a name and a set of group expressions that you specify. The set of group expressions can be a single dataset field reference or a combination of multiple expressions. At runtime, Report Designer combines and applies group expressions to data in a group.

Groups can be nested within one another to have parent child relationship. We can think of the parent/child groups as a tree structure.

1.1: Reporting

## Nested Row Group Report

Orders Placed by Customers between 1/1/1996 and 1/1/1998

Country	City	Customer	Order ID	Order Date	Freight
Argentina	Buenos Aires	Cactus Comidas para llevar	10521	29 Apr 1997	17.2200
			10782	17 Dec 1997	1.1000
				Total Orders 2	
		Océano Atlántico Ltda.	10531	08 May 1997	8.1200
	10409		09 Jan 1997	29.8300	
			Total Orders 2		
	Rancho grande	10448	17 Feb 1997	38.8200	
		10716	24 Oct 1997	22.5700	
			Total Orders 2		
		Total	6		<b>117.6600</b>
			6		117.6600
Austria	Graz	Ernst Handel	10698	09 Oct 1997	272.4700
			10771	10 Dec 1997	11.1900

June 14, 2015 | Proprietary and Confidential | - 21 -

IGATE  
Speed. Agility. Imagination.

### Groups in a Tablix Data Region

**Details Group** The Details group consists of all data from a report dataset after Report Designer applies dataset and data region filters. Thus, the Details group is the only group that has no group expression.

Basically, the details group specifies the data that you would see when you run a dataset query in a query designer. For example, you have a query that retrieves all columns from a sales order table. Thus, the data in this detail group includes all the values for every row for all the columns in the table. The data in this detail group also includes values for any calculated dataset fields that you have created.

By default, when you add a table or list to your report, Report Designer automatically creates the Details group for you, and adds a row to display the detail data. When you view the data region, the details row repeats once for every value in the result set.

**Row groups and column groups** You can organize data into groups by rows or columns. Row groups expand vertically on a page. Column groups expand horizontally on a page. Groups can be nested, for example, group first by [Year], then by [Quarter], then by [Month]. Groups can also be adjacent, for example, group on [Territory] and independently on [ProductCategory].

**Recursive hierarchy groups** A recursive hierarchy group organizes data from a single report dataset that includes multiple levels. For example, a recursive hierarchy group could display an organization hierarchy, for example, [Employee] that reports to [Employee]. Reporting Services provides group properties and built-in functions to enable you to create groups for this kind of report data.

1.1: Reporting

## Drill Down Group Report

Select Start Order Date:- 01-01-1996  Select End Order Date 01-01-1998

1 of 1

Orders Placed by Customers between 1/1/1996 and 1/1/1998

Country	City	Customer	Order ID	Order Date	Freight
<input checked="" type="checkbox"/> Argentina	Total		6		117.6600
<input checked="" type="checkbox"/> Austria	Total		29		5000.7400
<input checked="" type="checkbox"/> Belgium	Total		9		518.0500
<input checked="" type="checkbox"/> Brazil	<input checked="" type="checkbox"/> Campinas	Total	7		309.0600
	<input checked="" type="checkbox"/> Resende	Total	6		131.7400
	<input checked="" type="checkbox"/> Rio de Janeiro	Total	22		945.1100
	<input checked="" type="checkbox"/> Sao Paulo	Total	21		2068.8600
		Total	56		3454.7700
<input checked="" type="checkbox"/> Canada	Total		22		1828.4000

## Hiding Report Items Conditionally

You can control whether a report item initially displays or is hidden when a user views a report. By providing a toggle on a text box, you can enable users to hide and display items interactively. For a table or matrix, you can show or hide static rows and columns, or rows and columns that are associated with groups.

The main reason for hiding items is to provide a report that shows summary data but enables a user to drill down into detail data.

For example, you can initially hide all the rows except the outer group summary row for a table with row groups. For each inner group (including the details group), add a toggle to the grouping cell of the containing group. When the report is rendered, the user can click the text box to expand and collapse the detail data.

## Drillthrough Reports

A Drillthrough report enables a user to click a link for a summary value and open a separate, related report to show detail data. The detail data is only retrieved when the detail report runs. Drillthrough reports typically require fewer resources than drilldown reports.

For example, a sales order summary report might list all the sales orders for a sales person, and when each sales order number might link to a report that shows the details of that order.

If the data for the main report and the detail report must be retrieved at the same time, consider using a drilldown report or a subreport.

Drillthrough reports typically have report parameters that specify which report data to display. For example, when you click a sales order number in a main report, a drillthrough report opens, which accepts the sales order number as a parameter, and then displays all the data for that sales order. When you create the link in the main report, you must specify values to pass as parameters to the drillthrough report.

1.1: Reporting

## Matrix Report

Find | Next

	Argentina	Austria	Belgium	Brazil
Beverages	1798.0000	26452.0500	5864.4000	40400.5000
Condiments	907.0000	16802.4000	2714.7000	12139.0000
Confections	2135.1000	14653.3500	7711.1800	12164.7300
Dairy Products	1143.5000	30342.9000	8825.0000	16894.5000
Grains/Cereals	390.0000	14854.2500	3226.0000	6638.0000
Meat/Poultry		12001.4800	2258.5000	8008.1200
Produce	1139.0000	13755.9500	3223.2000	5385.1500
Seafood	606.5000	10634.2500	1312.0000	13338.4800

June 14, 2015 Proprietary and Confidential - 23 -

IGATE Speed.Agency.Imagination

## Matrix Reports

We can use a Matrix Report to display grouped data and summary information. You can group data by multiple fields or expressions in row and column groups. Matrices provide functionality similar to crosstabs and pivot tables. At run time, as the report data and data regions are combined, a matrix grows horizontally and vertically on the page. Values in matrix cells display aggregate values scoped to the intersection of the row and column groups to which the cell belongs. You can format the rows and columns to highlight the data you want to emphasize. You can also include drilldown toggles that initially hide detail data; the user can then click the toggles to display more or less detail as needed.

1.1: Reporting

## Matrix Report

1 of 1 | Find | Next

	Argentina	Austria	
	Buenos Aires	Graz	Salzburg
Beverages	1798.0000	13704.5500	12747.5000
Condiments	907.0000	15584.7000	1217.7000
Confections	2135.1000	13934.7000	718.6500
Dairy Products	1143.5000	26629.3000	3713.6000
Grains/Cereals	390.0000	12898.0000	1956.2500
Meat/Poultry		9271.4800	2730.0000
Produce	1139.0000	13104.9500	651.0000
Seafood	606.5000	8109.0000	2525.2500

IGATE Speed.Agency.Imagination

June 14, 2015 | Proprietary and Confidential | - 24 -

The screenshot displays the 'Groups and Grouping Pane' in the SQL Server Reporting Services 2008 Authoring Reports interface. The pane is titled 'Groups and Grouping Pane' and includes a header '1.1: Reporting'. The main area shows a 'Sales by' report structure with a grid. The grid has columns for 'Geography', 'CountryRegion', 'Year', and 'Total'. Row groups are defined by 'Area' and 'Year'. A specific row group 'Subcat' is highlighted with a red box. The 'Row Groups' pane on the left shows 'Category' and 'Subcat' as children of the main row group. The 'Column Groups' pane on the right shows 'Geography', 'CountryRegion', and 'Year' as children of the main column group. The bottom of the pane shows the date 'June 14, 2015', the status 'Proprietary and Confidential', and page number '-25-'.

## Grouping Pane

The Grouping pane displays the row groups and column groups for the currently selected Tablix data region. The Grouping pane is not available for the Chart or Gauge data regions. The Grouping pane is comprised of a Row Groups pane and a Column Groups pane. The Row Groups pane and the Column Groups pane display a hierarchical view for all parent groups, child groups, and adjacent groups. A child group appears indented under its parent group. An adjacent group appears at the same indent level as its sibling groups. The following figure shows a Tablix data region with nested row groups and nested and adjacent column groups.

1.1: Reporting

## Creating Sub Report

Category ID      Category Name      Description  
[CategoryID]      [CategoryName]      [Description]

ProductsList

June 14, 2015 | Proprietary and Confidential | - 26 -

IGATE  
Speed. Agility. Imagination.

### Sub Reports

A sub report is a report item that displays another report inside the body of a main report. Conceptually, a sub report is similar to a frame in a Web page. It is used to embed a report within a report. Any report can be used as a sub report.

You can design the parent report to pass parameters to the sub report. A sub report can be repeated within data regions, using a parameter to filter data in each instance of the sub report.

You can place a sub report in the main body of the report, or in a data region. If you place a sub report in a data region, the sub report will repeat with each instance of the group or row in the data region. To pass a value from the group or row to the subreport, in the sub report value property, use a field expression for the field containing the value you want to pass to the sub report parameter.

To pass parameters from the parent report to the subreport, define a report parameter in the report that you use as the subreport. When you place the sub report in the parent report, you can select the report parameter and a value to pass from the parent report to the report parameter in the subreport.

1.1: Reporting

## Sub Report

[Navigation icons] 1 of 2 [Report View Options]

Category ID	Category Name	Description
1	Beverages	Soft drinks, coffees, teas, beers, and ales

Product ID	Product Name	Unit Price
1	Chai	18.0000
2	Chang	19.0000
39	Chartreuse verte	18.0000
38	Côte de Blaye	283.5000
24	Guaraná Fáctica	4.5000
43	Ish Coffee	48.0000
76	Lakkaliköori	18.0000
87	Laughing Lumberjack Lager	14.0000
70	Oubak Lager	15.0000
75	Rhönbräu Klosterbier	7.7500
34	Sasquatch Ale	14.0000
35	Steeleye Stout	18.0000

June 14, 2015 | Proprietary and Confidential | - 27 -

IGATE  
Speed. Agility. Imagination.

In Report Designer, if you preview a report that contains subreports, and then change the subreport, the preview may not be updated. To see the changes, click the **Refresh** button.

1.1: Reporting

## Hierarchical Group Report

Reporting Hierarchy

Employees
Fuller Andrew
Davolio Nancy
Leverling Janet
Peacock Margaret
Buchanan Steven
Suyama Michael
King Robert
Dodsworth Anne
Callahan Laura

June 14, 2015 | Proprietary and Confidential | -28-

IGATE  
Speed. Agility. Imagination.

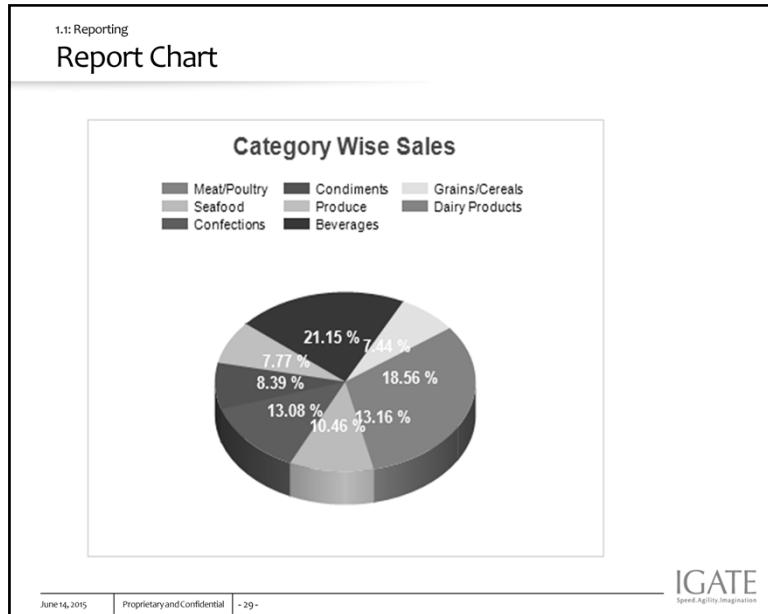
### Creating Recursive Hierarchies

To display recursive data where the relationship between parent and child is represented by fields in the dataset, you can set the data region group expression based on the child field and set the **Parent** property based on the parent field.

Displaying hierarchical data is a common use for recursive hierarchy groups, for example, employees in an organizational chart. The dataset includes a list of employees and the managers, where the manager names also appear in the list of employees.

To build a recursive hierarchy in a Tablix data region, you must set the group expression to the field that specifies the child data and the **Parent** property of the group to the field that specifies the parent data. For example, for a dataset that includes fields for employee ID and manager ID where employees report to managers, set the group expression to employee ID and the **Parent** property to manager ID.

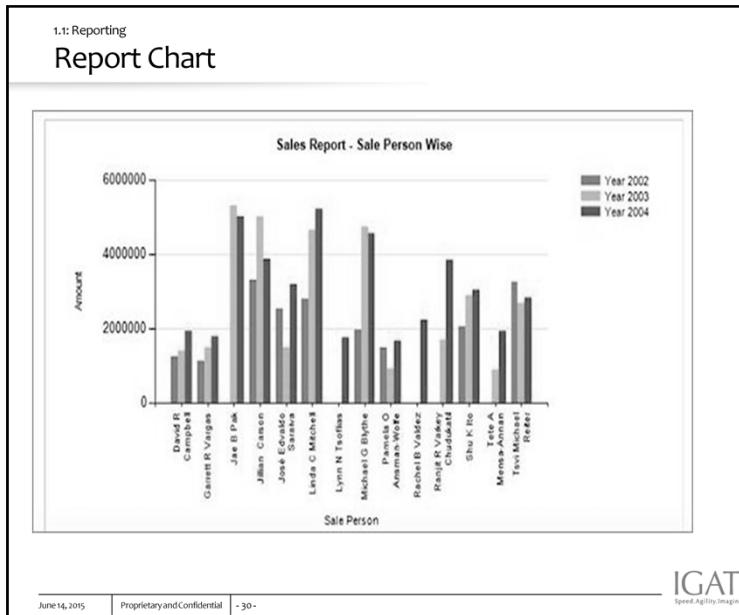
A group that is defined as a recursive hierarchy (that is, a group that uses the **Parent** property) can have only one group expression. You can use the **Level** function in text box padding to indent employee names based on their level in the hierarchy.



### Report Charts

One of the greatest rewards of developing any type of report is to transform the vast amounts of business data into useful information that can support commercial decision-making; producing such reports as the performance of an internal process, percentage of an employees' contribution to overall product sales, or a department's budget compared to other departments. It can become a real challenge for the developer of a report to present this vast amount of information properly, because the correct interpretation of data by the user is as important as the data itself.

One solution is to present the data in an aggregated format so that business users can more easily and quickly digest this information. The chart is a tool designed specifically for the presentation of aggregated data. If it is done properly, it is possible for the user to quickly grasp the information available as they dash off to their next meeting, without needing to scroll through, and assimilate, a huge list of tabular results.



1.1: Reporting

## Free Form Report

Bikes

Product	Sales Total
Mountain Bikes	\$25,998,934.56
Road Bikes	\$28,760,967.50
Touring Bikes	\$9,920,485.64

Geography	Sales Totals
Europe	
North	Preview of List with tables, chart
Pacific	

IGATE  
Speed. Agility. Imagination.

June 14, 2015 | Proprietary and Confidential | - 31 -

## Summary

### ➤ Authoring various types of reports

- Data Source
- Data Sets
- Data Regions
- Grouping
- Summation



# SQL Server Reporting Services 2008

Lesson 3: Working with Report Parameters, Expressions  
and Functions

June 14, 2015

Proprietary and Confidential

- 1 -

**IGATE**  
Speed. Agility. Imagination.

## Lesson Objectives

- **Working With**
  - Parameters
  - Expressions
  - Functions
  - Built In Collections



## Using Parameters in Reporting Services

- The most common use of parameters is to vary report data retrieved by dataset queries.
- Users are prompted for a value or values when they run the report, and the dataset query retrieves only the data that is requested.
- The dataset query includes query parameters and Reporting Services automatically creates corresponding report parameters that are indirectly linked to the query parameters.

June 14, 2015

Proprietary and Confidential

- 3 -



In Reporting Services, parameters are used to specify the data to use in a report. Report parameters typically filter report data that is retrieved from a data source. Filtering data at the data source can improve performance for processing and viewing a report.

After a report parameter is created, you can modify the default values and other parameter properties, such as visibility. For example, report parameters can use built-in variables like UserID, so you might want to create a report with a hidden parameter that selects data specific to the user running the report. Parameters can contain single values or multiple values, use a static or query-based valid values list, and accept null or blank values. Multivalue parameters enable users to select more than one value at run time.

### Dependent or Cascading Parameters

When you create a query that uses multiple query parameters, you can create a set of cascading parameters. Cascading parameters provide a way of filtering a very large number of parameter values down to a manageable number of values. For example, suppose a query includes the parameters @Category, @Subcategory, and @Product, where the list of subcategories is dependent on @Category, and the list of products is dependent on the @Subcategory. When a user chooses a value for the report parameter Category, the values for Subcategory are limited to valid values for the chosen category. After the user selects a value for Subcategory, the choices for Product have already been filtered by the choice for category and subcategory. Using this technique, you can reduce the valid choices for a parameter down to a reasonable number of values.

**To design cascading parameters, you must include the following items in your report:**  
A. The main dataset query, which has multiple related query parameters.

- B. separate dataset for each report parameter that supplies its available values. It is important to use the same case-sensitive spelling for each query parameter so that the query parameters and report parameters are linked properly. The query for each set of available values for each report parameter must provide only values that make sense in the context of the main query.

## Using Parameters to Connect to Other Reports

- We can use parameters to relate main reports to drill through reports, to sub reports, and to linked reports.
- To pass parameters from the parent report to the sub report, define a report parameter in the report that you use as the sub report.
- When you place the sub report in the parent report, you can select the report parameter and a value to pass from the parent report to the report parameter in the sub report.

For dividing the report requirement, the sub reports could be used. A existing report also can be re used across other report designs.

Sub report could be linked with a parameter or be unlinked.

## Working with Expressions

- Expressions are widely used throughout a report definition to retrieve, calculate, display, group, sort, filter, parameterize, and format the data in a report.
- Because many report item properties can be set to an expression, you have great flexibility to control the content, design, and interactive nature of your report by using expressions.
- Expressions are written in Visual Basic, saved in the report definition, and evaluated by the report processor when you run the report.

On the report design surface, expressions appear as simple or complex expressions. Simple expressions contain a reference to a single dataset field, parameter, or built-in field. Simple expressions appear on the design surface and in dialog boxes in brackets; for example, a dataset field appears as [ProductID]. Simple expressions are created for you automatically (for example, when you drag a field from a dataset onto a text box), or you can type them directly into a data region cell, or text box on the design surface or in a dialog box. Complex expressions can contain multiple built-in references, operators, and function calls, and appear on the design surface as <>Expr<>. To see or change the expression text, you must open the **Expression** dialog box.

## Working with Expressions

Item	Display text example	Expression text example
Dataset fields	[Sales]	=Fields!Sales.Value
	[SUM(Sales)]	=Sum(Fields!Sales.Value)
	[FIRST(Store)]	=First(Fields!Store.Value)
Report parameters	[@Param]	=Parameters!Param.Value
Built-in fields	[vReportName]	=Globals!ReportName.Value
Literal characters used for display text	\{Sales\}	[Sales]
Complex expressions	<<Expr>>	="Page " & Globals!PageNumber & " of " & Globals!TotalPages

## Using Functions

### ➤ Aggregate Functions

- Count
- CountDistinct
- Max
- Min
- Sum
- Avg

June 14, 2015

Proprietary and Confidential

- 7 -

**IGATE**  
Speed. Agility. Imagination.

Functions are predefined formulas that perform calculations by using specific values, called arguments, in a particular order. Functions can be used to perform simple or complex calculations. The structure of a function begins with the function name, followed by an opening parenthesis, the arguments for the function separated by commas, and a closing parenthesis.

Arguments can be field references, numbers, text, and logical values such as TRUE or FALSE. Arguments can also be constants, formulas, or other functions. The arguments that you enter must produce a valid value for that argument. For example, if the formula is multiplying two integers, the result cannot be a text string.

## Using Functions

### ➤ Conditional Functions

- IF
  - IF(Sub Total >= 1000, "Discount", "No Discount")
- IN
  - IN(State Province, {"California", "Oregon", "Washington", "Alaska"})
- Switch
  - SWITCH(Customer Type = "I", "Individual", Customer Type = "S", "Shop")

June 14, 2015

Proprietary and Confidential

- 8 -

**IGATE**  
Speed. Agility. Imagination.

On the report design surface, expressions appear as simple or complex expressions. Simple expressions contain a reference to a single dataset field, parameter, or built-in field. Simple expressions appear on the design surface and in dialog boxes in brackets; for example, a dataset field appears as [ProductID]. Simple expressions are created for you automatically (for example, when you drag a field from a dataset onto a text box), or you can type them directly into a data region cell, or text box on the design surface or in a dialog box. Complex expressions can contain multiple built-in references, operators, and function calls, and appear on the design surface as <>Expr<>. To see or change the expression text, you must open the **Expression** dialog box.

## Using Functions

### ➤ DateTime Functions

- Date and Today
- DateTime and Now
- Day
- Month
- Quarter
- Year

June 14, 2015

Proprietary and Confidential

- 9 -

**IGATE**  
Speed. Agility. Imagination.

On the report design surface, expressions appear as simple or complex expressions. Simple expressions contain a reference to a single dataset field, parameter, or built-in field. Simple expressions appear on the design surface and in dialog boxes in brackets; for example, a dataset field appears as [ProductID]. Simple expressions are created for you automatically (for example, when you drag a field from a dataset onto a text box), or you can type them directly into a data region cell, or text box on the design surface or in a dialog box. Complex expressions can contain multiple built-in references, operators, and function calls, and appear on the design surface as <>Expr<>. To see or change the expression text, you must open the **Expression** dialog box.

## Using Functions

### ➤ Text Functions

- Find
  - FIND("A light yet stiff aluminum bar for long distance riding.", "aluminum bar")
- Length
- Ltrim and Rtrim
- Substring
  - SUBSTRING("lavender", 3, 5)
  - Returns vende
- Replace

June 14, 2015

Proprietary and Confidential

- 10 -

**IGATE**  
Speed. Agility. Imagination.

On the report design surface, expressions appear as simple or complex expressions. Simple expressions contain a reference to a single dataset field, parameter, or built-in field. Simple expressions appear on the design surface and in dialog boxes in brackets; for example, a dataset field appears as [ProductID]. Simple expressions are created for you automatically (for example, when you drag a field from a dataset onto a text box), or you can type them directly into a data region cell, or text box on the design surface or in a dialog box. Complex expressions can contain multiple built-in references, operators, and function calls, and appear on the design surface as <>Expr<>. To see or change the expression text, you must open the **Expression** dialog box.

## Using Built In Collections

- **Globals**:- Represents global variables useful for reports, such as the report name or page number.
- **User**:- Represents a collection of data about the user running the report.
- **Parameters**:- Represents the collection of report parameters, each of which can be single-value or multivalue.
- **Fields**:- Represents the collection of fields of the dataset that are available to the report.
- **DataSets**:- Represents the collection of datasets referenced from the body of a report definition.

Reporting Services provides the following built-in collections that you can reference from an expression: ReportItems, Parameters, Fields, DataSets, DataSources, Variables, and built-in fields for global information such as the report name. Not all collections appear in the **Expression** dialog box. The DataSets and DataSources collections are available only at run-time for published reports on a report server. The ReportItems collection is collection of text boxes in a report region, for example, the text boxes on a page or in a page header.

## Using Built In Collections

- **DataSources**:- Represents the collection of data sources referenced from within the body of a report. Does not include data sources used only in page headers or page footers.
- **Variables**:- Represents the collection of report variables and group variables.
- **ReportItems**:- Represents the collection of text boxes for a report item. This collection can be used to summarize items on the page for including in a page header or page footer.

## Using Global Collections

Member	Type	Description
ExecutionTime	<b>DateTime</b>	The date and time that the report began to run.
PageNumber	<b>Integer</b>	The current page number that can be used only in page header and footer.
ReportFolder	<b>String</b>	The full path to the folder containing the report. This does not include the report server URL.
ReportName	<b>String</b>	The name of the report as it is stored in the report server database.
ReportServerUrl	<b>String</b>	The URL of the report server on which the report is being run.
TotalPages	<b>Integer</b>	The total number of pages in the report that can be used only in page header and footer.

The Built-in fields collection, which includes both the **Globals** and the **User** collections, represent global values provided by Reporting Services when a report is processed. The **Globals** collection provides values such as the name of the report, the time when report processing began, and current page numbers for the report header or footer.

## Summary

### ➤ Working With

- Parameters
- Expressions
- Functions
- Built In Collections



June 14, 2015

Proprietary and Confidential

- 14 -

**IGATE**  
Speed. Agility. Imagination.

# SQL Server Reporting Services 2008

Lesson 4: Working with Report Builder and Report Model

June 14, 2015

Proprietary and Confidential

- 1 -

**IGATE**  
Speed. Agility. Imagination.

## Lesson Objectives

- Working With Report Model
- Working With Report Builder



## Working with Report Model

- In order to create reports using Report Builder, you must first build a report model based on your business database.
- Model Designer provides the most flexibility for designing models based on SQL Server or Oracle databases.
- In order to build a model, you must first create a report model project.
- A report model project is a container for the model and consists of one or more data source (.ds) files, one or more data source view (.dsv) files, and one or more report model (.smdl) files.

June 14, 2015

Proprietary and Confidential

- 3 -

**IGATE**  
Speed. Agility. Imagination

A report model is a metadata description of a data source and its relationships. Report models provide familiar business names for database fields and tables, logically grouped model items, and predefined relationships between items within the data source. The report model helps Report Builder users to explore and select the data that they want to use from the underlying data source. The report model definitions are used by Report Server to automatically generate a query for retrieving the requested data.

To build your model, you need to run the Model Designer wizard. When running the Model Designer wizard, you can create a data source, a data source view, and generate the model. If you want, you can continue to refine your model using the Model Design window. You can refine your model by rearranging the model items within the model and by adding additional model items, such as entities, folders, or perspectives, to the model.

When you have finished designing a model, you can publish it to a report server running in either native mode or SharePoint integration mode. After the model is published, individuals with the appropriate permissions can use the model to create reports in Report Builder or Report Designer.

To use a published model, the data source used in the model must be available on the report server. You can publish data sources for the model from Model Designer at the same time you publish a model, or you can create a data source independently on the report server and then bind it to the published model.

## Creating and Customizing Report Model

- When you create a report model using the Report Model wizard, a collection of entities is automatically generated for you based on items in your database as described in your data source view (.dsv) file.
- For example, a Products entity might contain information, such as product name, product number, and list price, related to each of the products that your company manufactures.
- Once you create the report model, you might want to customize the model to make it even easier for your users to use.

June 14, 2015

Proprietary and Confidential

- 4 -



Once you create the report model, you might want to refine the model to make it even easier for your users to use. By default, entities and folders are listed in alphabetical order, but you can rearrange them according to how you think your users will use the data in them to build reports. For example, if you think most of your Report Builder users will use the Products entity frequently you might want to move it to the top of the model items list.

We can organize entities, perspectives and other folders into folders within your report models. This allows you to expose frequently used model items, group related items, and archive inactive or infrequently used items to facilitate navigation for your users.

Perspectives also can be defined. A perspective is a subset of a model. Model subsetting can be helpful when the size of your model is very large, or if you want to partition the model so that specific user groups can access only the data they need. Perspectives exist within a model and can contain entities, folders, roles, source fields, and expressions. They are not generated by the Model Designer wizard. Instead, manually create them and then select the model items that you want to include in the perspective by using the Edit Perspective dialog box.

Entities contain source fields (attributes), roles, folders, and expressions. When you run Model Designer, roles and source fields are automatically created within entities for you. You can add additional fields, roles, folders, or expressions to the entities once the model is created.

A role indicates that two entities are related to each other. The role can be a 1-to-1 relationship, a 1-to-many relationship or a many-to-many relationship. For example, the Store Location entity has a 1-to-1 relationship with the Address entity; therefore, for every store location, there is a single address. The Customer entity has a 1-to-many relationship with the Sales Orders entity; this means that for every customer there are multiple sales orders.

### Working with Clickthrough Reports

In Report Builder, a clickthrough report is a report that provides detailed information about the data contained within the main report. A clickthrough report is displayed when the Report Builder user clicks the interactive data that appears in the main report. These reports are automatically generated by the report server. You, as the model designer, determine what is displayed in clickthrough reports by setting the DefaultDetailAttribute and DefaultAggregateAttribute properties that you assign to an entity in the report model.

## Working With Report Builder

- Report Builder allows business users to create their own reports based on a user-friendly report model created in Model Designer.
- Fully integrated with Microsoft SQL Server Reporting Services, Report Builder brings ad hoc reporting to all users.
- Report Builder is a client-side application that information workers can use to create and design ad hoc reports. Using Report Builder, users can select data and design reports without having to understand how and where the data is stored. Nor do they need to know any complex programming languages in order to create reports.

June 14, 2015 | Proprietary and Confidential | - 5 -

**IGATE**  
Speed. Agility. Imagination.

Information workers throughout an organization need access to business data for timely and effective decision making. With a wide range of skills and expertise, these business users need an intuitive yet powerful tool to explore and find answers to critical questions without deep technical knowledge of corporate data sources.

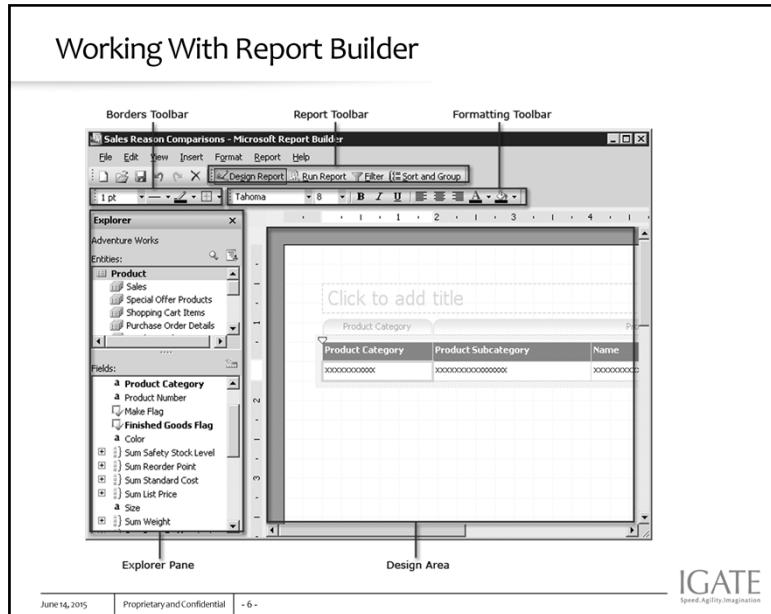
Users create reports with the Report Builder tool. The Report Builder interface is built on top of familiar Microsoft Office paradigms such as Excel and PowerPoint. Users start with report layout templates containing pre-defined data regions to build combinations of tables, matrices and charts. They navigate the reporting model to select report items and set constraints to filter the report data. The reporting model contains all of the necessary information for the Report Builder to automatically generate the source query and retrieve the requested data.

### The Report Builder also allows users to:

1. Add text and formatting to reports.
2. Create new fields and calculations defined against the reporting model.
3. Preview, print and publish reports.
4. Explore data related to the content of their report.

### Interactive Data Exploration

1. Report Builder reports enable users to interactively drill and explore data related to the content of their report.
2. Automatic generation of new drill down reports allows users to explore data along model navigation paths.
3. Queries for navigation reports are generated on the fly, passing the context of the user's current "location."
4. Users can keep drilling as long as there is a relationship to follow from the current item



Using Report Builder, users can create table, matrix, or chart reports. To get started, select a report layout template and then simply drag and drop the fields into the design area. The data can be manipulated by filtering, grouping and sorting, or working with formulas. You can specify parameters, such as the prompt parameter in filters. In addition, you can add images to your report layout and format your reports to use colors, fonts, lines, and different formats. Save reports to the report server, and then they can be managed just like any other report on the server. Or, reports can be exported to the local computer as a different file type, such as a TIFF, PDF, Microsoft Office Excel, or HTML file.

Report Builder builds reports based on report models, referred to as data sources in Report Builder, provided by the model designer who is typically an analyst, database administrator, or database developer. A data source contains information about the data available in the database and the relationships between that data. When fields are dragged onto the design area, users are laying out the report and providing the information needed to retrieve the data at the same time.

Report Builder is a ClickOnce Windows Forms application that is accessed from the report server for easy centralized management. Report Builder reports are published using Report Definition Language (RDL), which allows users to take advantage of Reporting Service's full capabilities. Because Report Builder reports are saved as RDL, they can be opened and modified using the advanced programming capabilities in Report Designer.

## Summary

- Working With Report Model
- Working With Report Builder



June 14, 2015

Proprietary and Confidential

- 7 -

**IGATE**  
Speed. Agility. Imagination.

# Sql Server Reporting Services

Lesson 5: Building Report Model

## Lesson Objectives

- **Creating Report model**
  - Adding data source to Report Model
  - Adding data source view
  - Adding Report Model through wizard
- **Reviewing Report Model**
- **Deploying report on server**



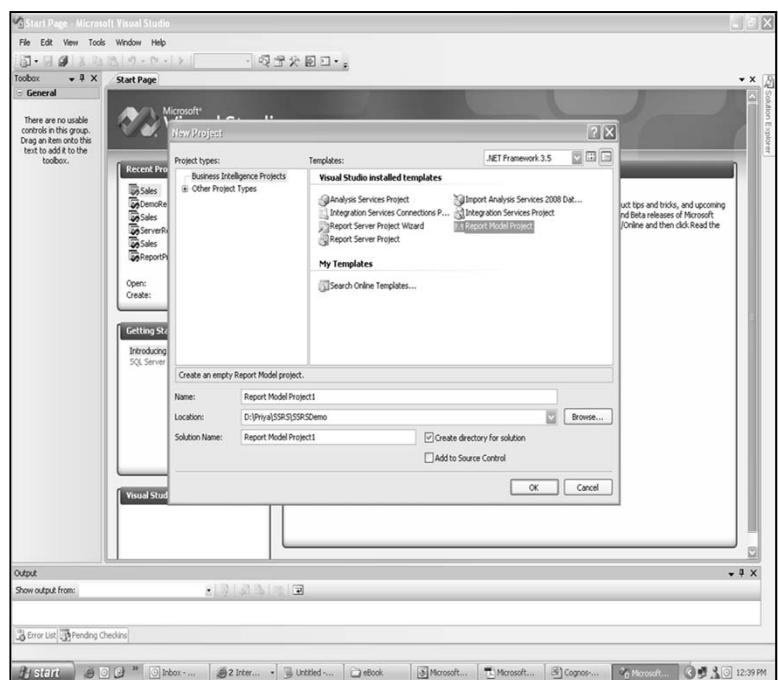
5.1: Creating Report model

## Creating Report model

- Report models are used to deploy ad hoc reports to report server.
- It is an intermediate layer between data source and business users.
- With this layer set in place, user does not need to know any programming language to query a database for building reports on that.
- To create report model, data source and data source view (DSV) is required for connection information and description of data structure respectively.

June 14, 2015 Proprietary and Confidential - 3 -

IGATE Speed. Agility. Imagination



5.1: Creating Report model  
Adding Data Source to Report Model

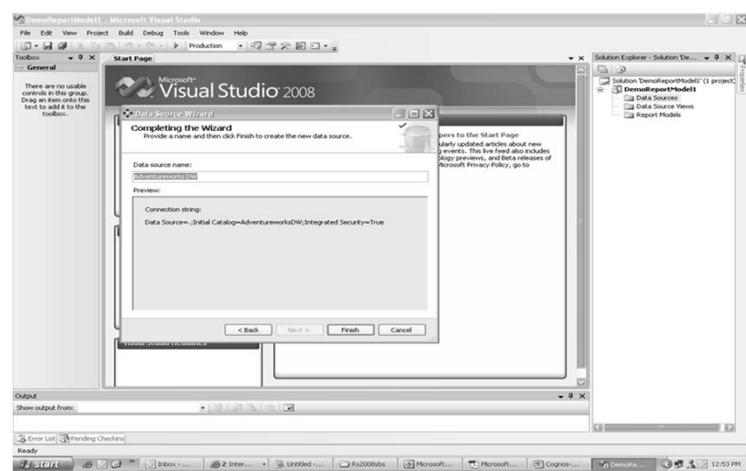
- Adding data source gives database connection and authentication information for accessing database.
- Only one data source is used for each report model.

June 14, 2015 | Proprietary and Confidential | - 4 -

IGATE Speed. Agility. Imagination

To add data source to Report Model:

1. Create a new project as Report Model Project in Microsoft Sql Server 2008 BI Development Studio, name it as ResellerSales.
2. Right click on Data Sources folder and select Add New Data Source and name it as Ds\_AdventureworksDW.
3. Select server name as Localhost and database as AdventureworksDW.
4. Test the connection and click on Finish in the wizard.



5.1: Creating Report model

## Adding Data Source to Report Model

- To define structure of data, we need to add data source view to report model.
- You can add only tables and views to DSV.
- Multiple tables can be added as named query like joins.
- Changes made in DSV do not affect source data.
- Its just a logical structure of actual data.

June 14, 2015

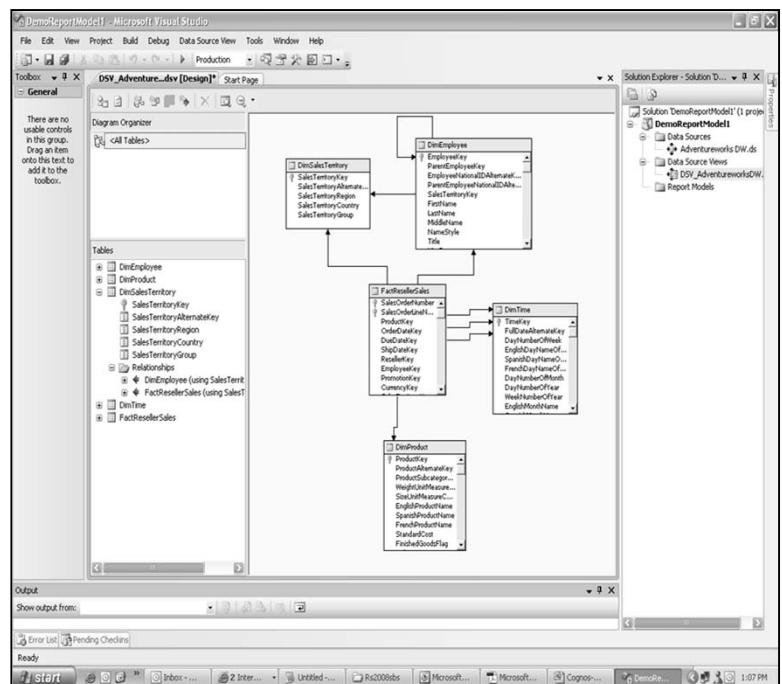
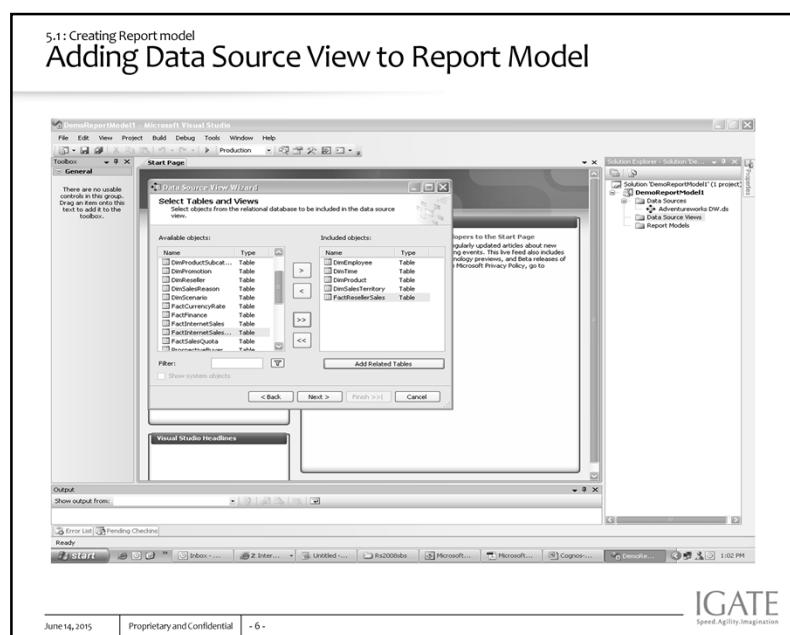
Proprietary and Confidential

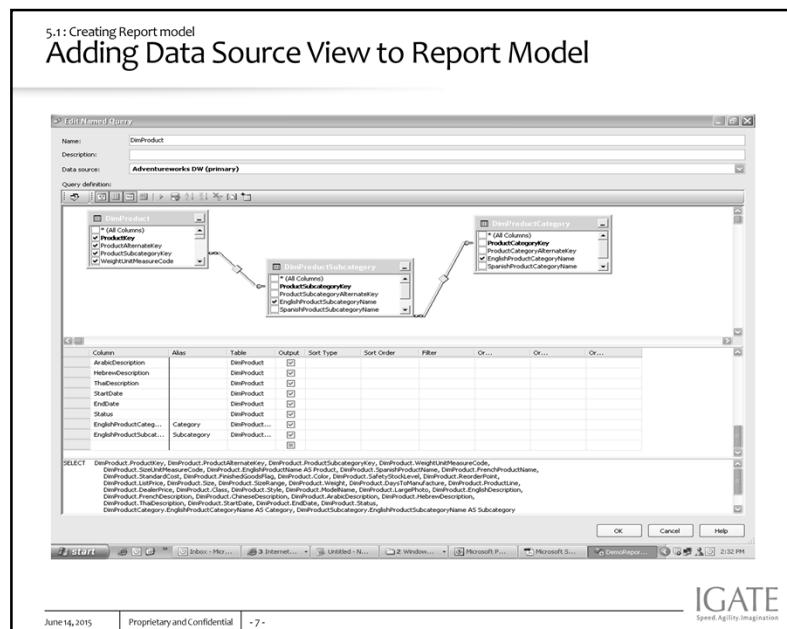
- 5 -

IGATE  
Speed. Agility. Imagination

To create DSV:

1. Open Solution Explorer, right click on Data Source View folder, select Add New Data Source View.
2. Select data source as Ds\_AdventureworksDW and select tables like Employee, Time, Product, SalesTerritory, FactResellerSales.
3. Click Next and then click Finish.
4. Double click on DSV file to view relationship amongst tables.
5. Select Product table and create new named query.
6. Add tables namely DimProductCategory and DimProductSubCategory.
7. Click Add and then Close.
8. In the Diagram pane, assign alias name of column EnglishProductName to **Product**, EnglishProductSubCategoryName to **SubCategory** and EnglishProductCategoryName to **Category**.
9. Create new named query and click OK.





5.1: Creating Report model  
Using Report Model wizard

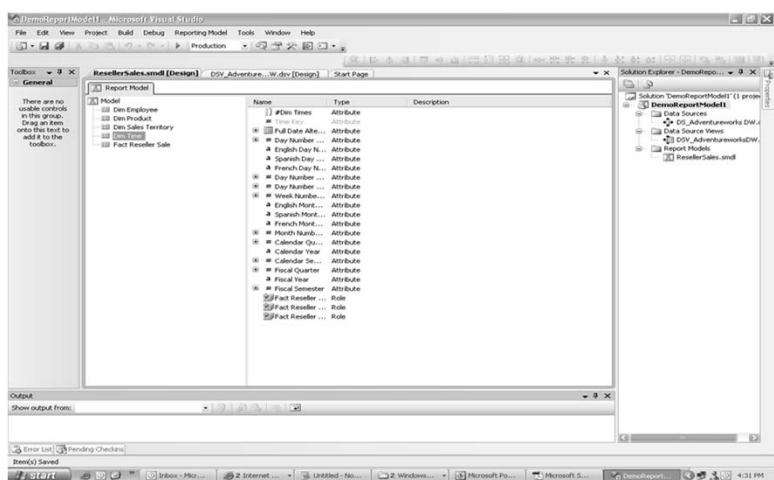
- Report Model wizard allows to select DSV to show cardinality information and model statistical data.
- It also stores numbers of instances of columns.

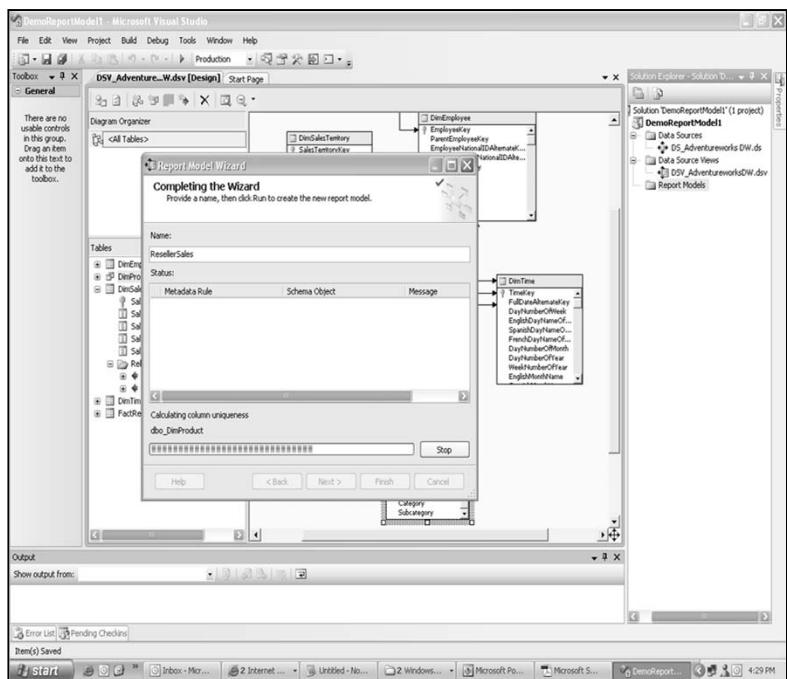
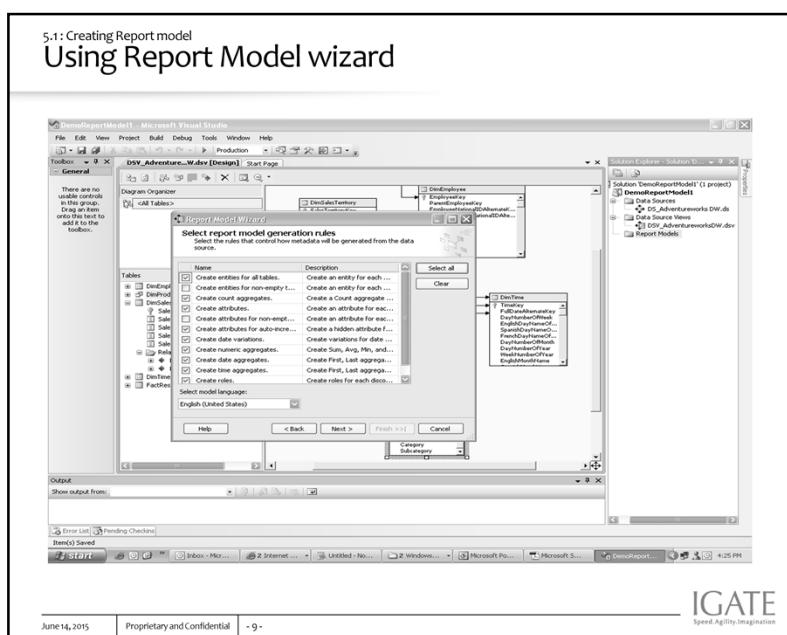
June 14, 2015 Proprietary and Confidential - 8 -

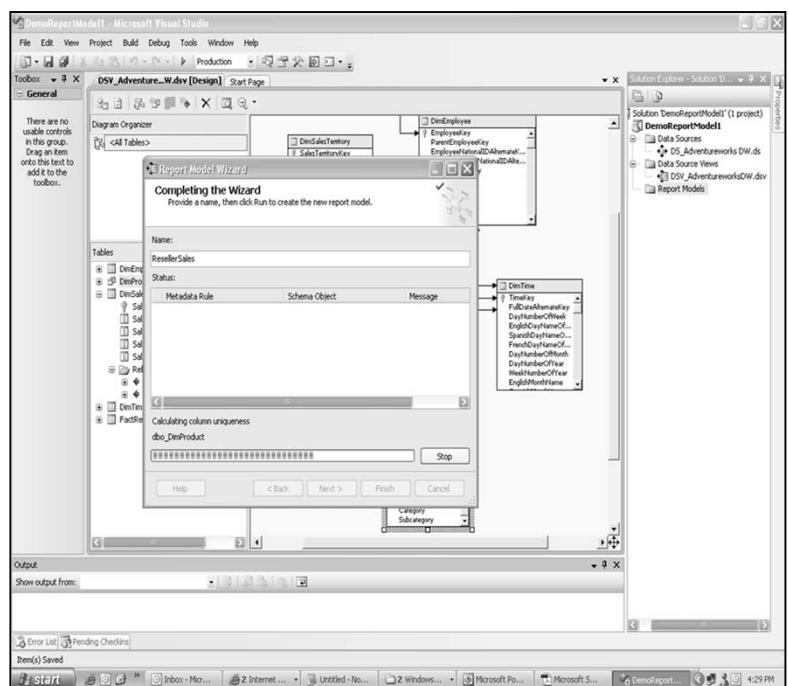
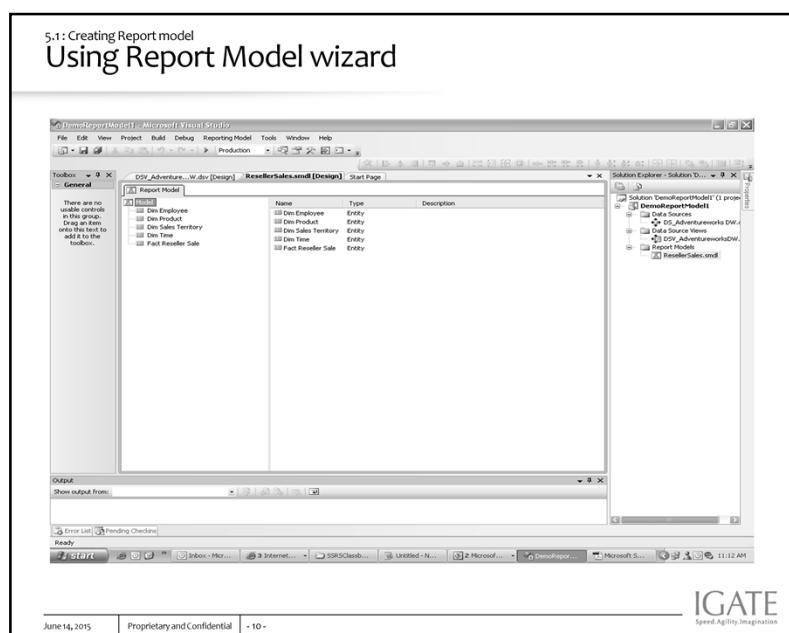
**IGATE**  
Speed. Agility. Imagination

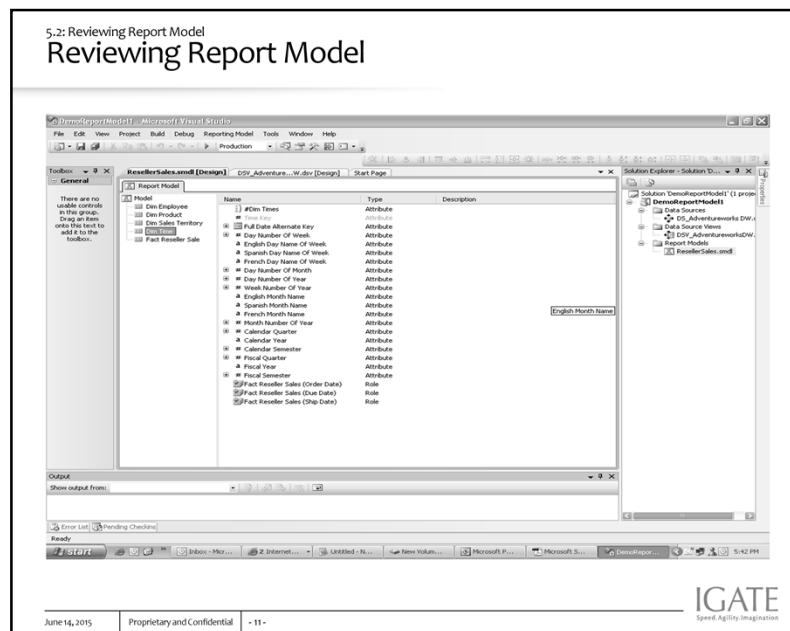
To run Report Model Wizard:

1. In Solution explorer, right click Report Model folder and click on Add New Report Model.
2. Select DSV file.
3. Select default generation rules and update statistics are selected on Model Statistics.
4. Name the model as ResellerSales and run the wizard.
5. Click on Finish to complete wizard.



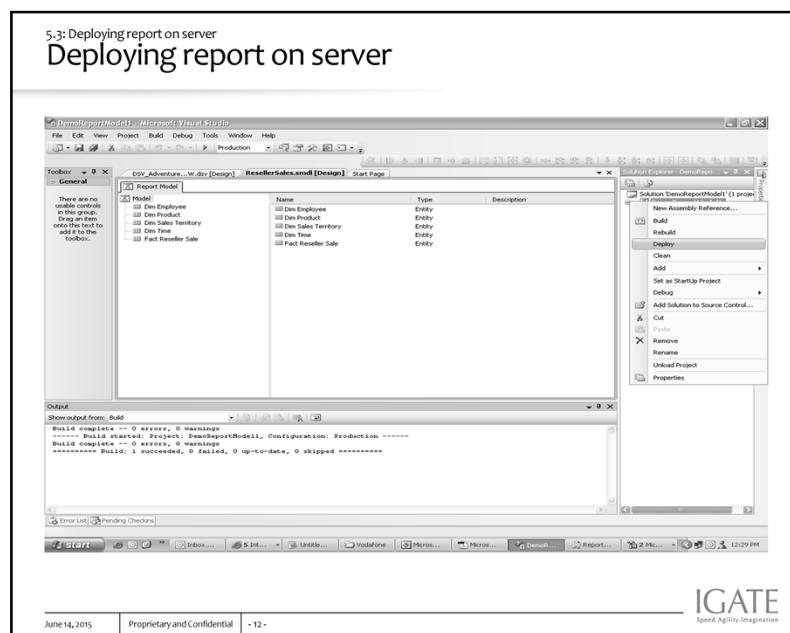






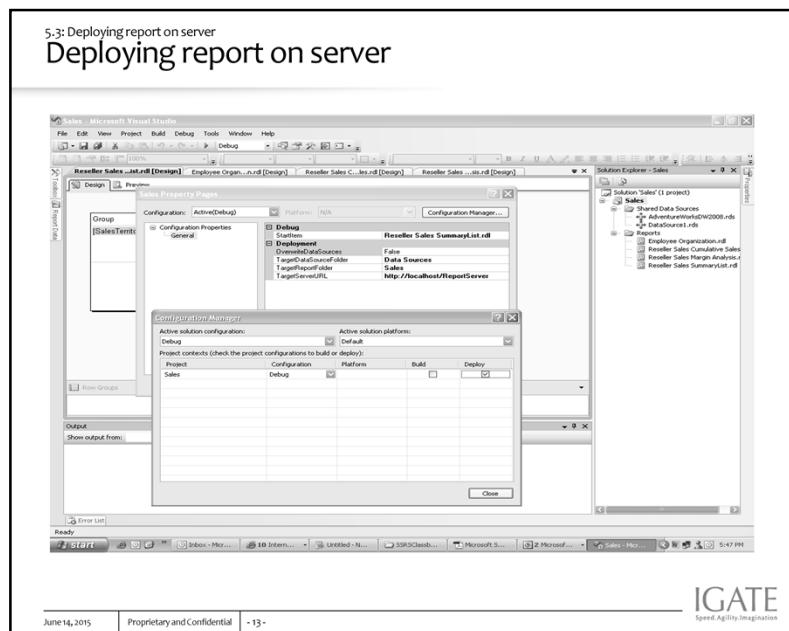
To review report model:

1. Open layout of report model designer.
2. Click on tree view, select table Time.
3. In detail view, expand Full Date Alternate Key attribute.



To deploy report model:

1. In Solution explorer, right click on report model, select Properties.
2. Click on Deploy.



To deploy report model:

1. Right click Project & select properties.
  - Assign following parameter values as mentioned.
  - TargetDataSourceFolder = Data Sources
  - TargetReportFolder = Sales
  - TargetServerURL = http://atrgoracle/ReportServer
2. Open the link <http://atrgoracle/Reports> to view all the deployed reports with data sources.

## Summary

- Report model defines a complete structure for a report according to specifications.
- Report model acts as a template for developing new reports from it.



# Sql Server Reporting Services

Lesson 6: Building Ad hoc Reports

June14,2015

Proprietary and Confidential

- 1 -

IGATE  
Speed. Agility. Imagination.

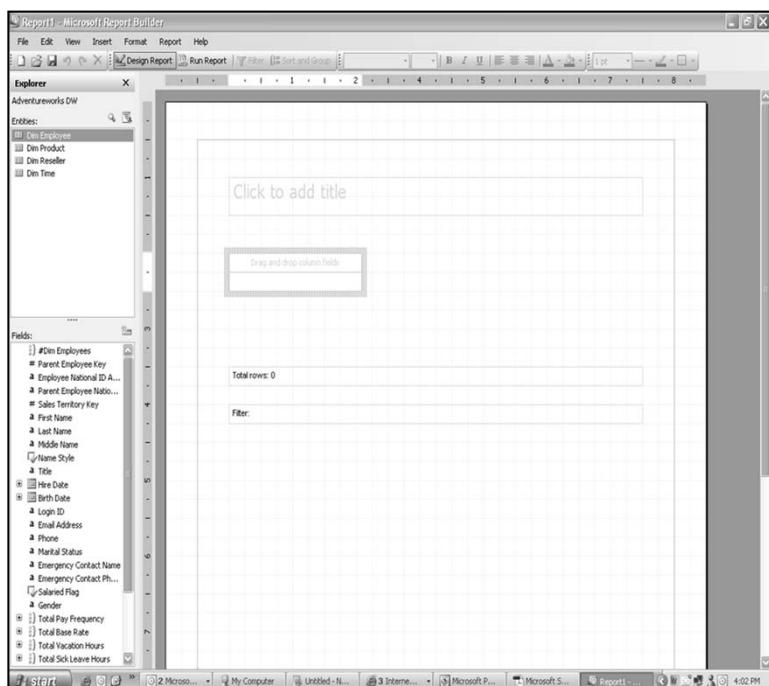
## Lesson Objectives

- Creating ad hoc reports using Report Builder
- Creating a table report
- Creating matrix report
- Creating chart reports
- Formatting a report
- Adding formulae
- Filter the report
- Sorting data



June14,2015 Proprietary and Confidential - 2 -

IGATE Speed. Agility. Imagination



The screenshot shows the Microsoft Report Builder interface. The top menu bar includes File, Edit, View, Insert, Format, Report, and Help. The toolbar below has icons for Design Report, Run Report, Filter, Sort and Group, and various text and table tools. The left sidebar is titled 'Explorer' and shows 'Adventureworks DW' with a tree view of tables: Dim Employee, Dim Product, Dim Reseller, and Dim Time. The main workspace is a grid with a title placeholder 'Click to add title' and a column placeholder 'Drag and drop column fields'. Below the grid are sections for 'Total rows: 0' and 'Filter:'. The bottom left pane is titled 'Fields' and lists numerous fields from the Dim Employee table, such as EmployeeID, FirstName, LastName, MiddleName, Title, HireDate, BirthDate, LoginID, EmailAddress, Phone, MaritalStatus, EmergencyContactName, EmergencyContactPhone, SaluredFlag, Gender, TotalPayFrequency, TotalBaseRate, TotalVacationHours, and TotalSickLeaveHours. The bottom of the screen shows the Windows taskbar with various open applications.

6.1: SSRS

## Creating ad hoc reports using Report Builder

- Report Builder is a tool for business users to create reports more easily.
- Non technical users can easily draw reports on model designed.
- Report can be previewed while designing it.

6.1: SSRS  
Creating a table report

Reseller Sales

Sales Territory Group	Sales Territory Country	Sales Territory Region	Total Sales Amount
000000000000	000000000000	000000000000	0.0
Total	Total	Total	0.0
Total Dim Sales Territories: 0			

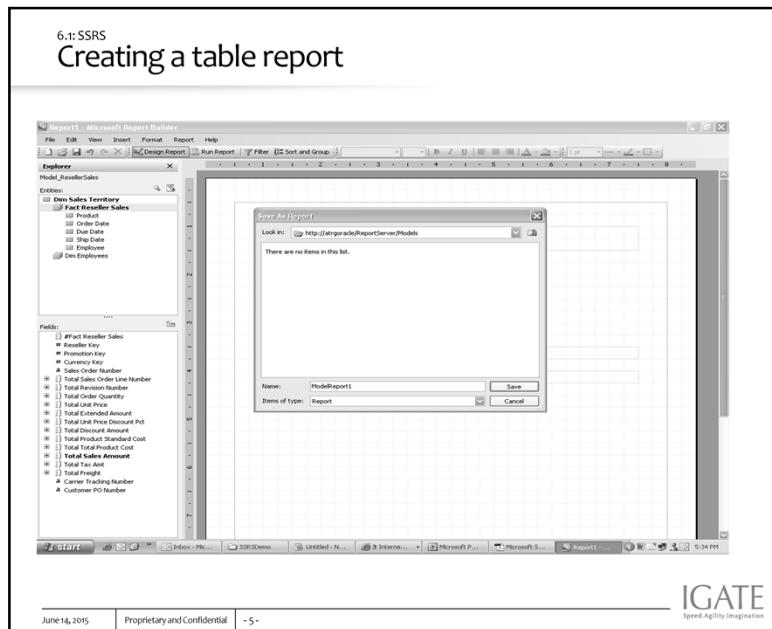
Filer: All Dim Sales Territories

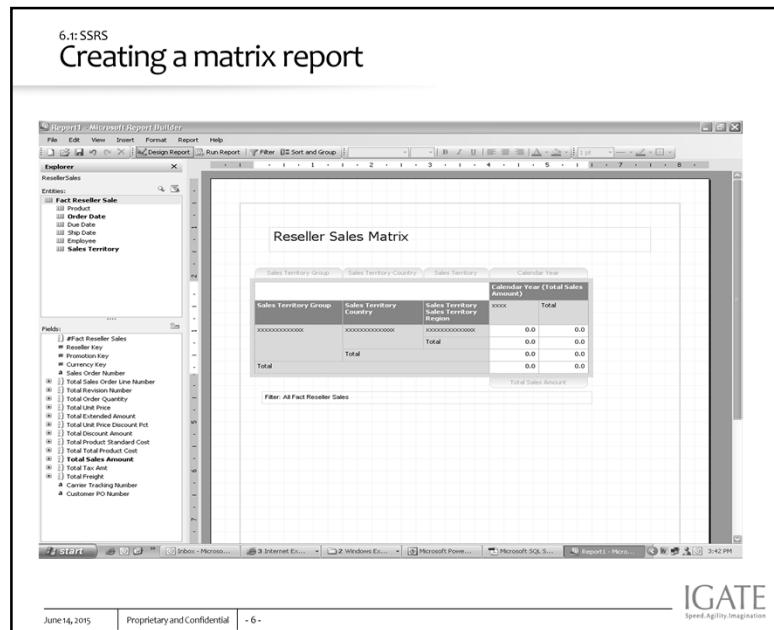
IGATE  
Speed Agility Inspiration

June14,2015 | Proprietary and Confidential | - 4 -

To create table report:

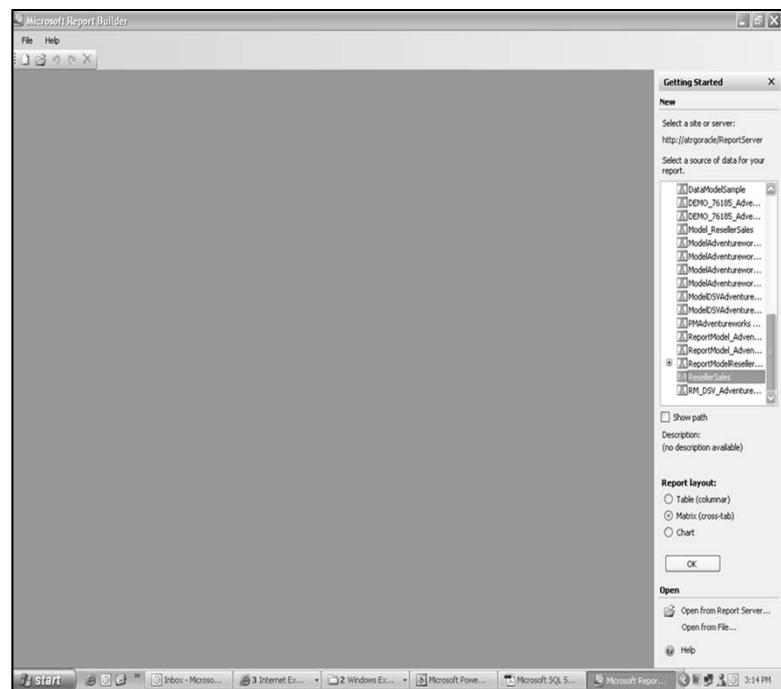
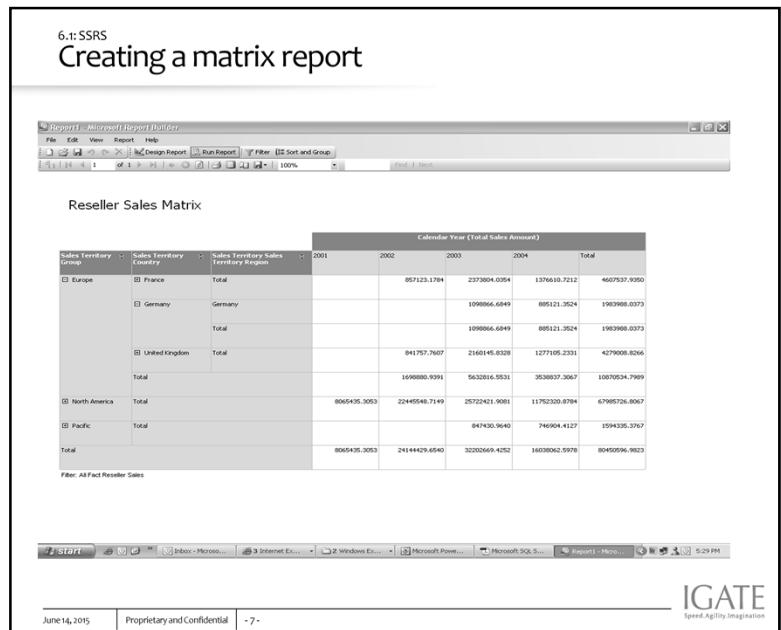
1. Select Report builder and choose for table report as the radio option button.
2. In the Entities list, select Reseller Sales.
3. Drag & drop Sales Territory Group, Sales Territory Country & Sales Region from the Fields list on report.
4. In the Fields list, expand Total Sales Amount to select related fields.
5. Drag Avg Sales Amount from Fields list & drop it on the report.
6. Add report title as 'Reseller Sales'.
7. Save & run the report.

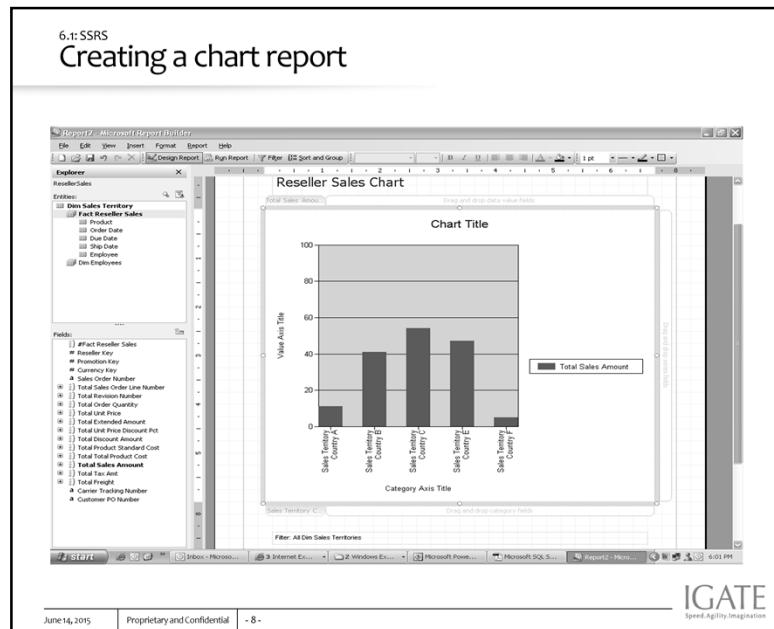




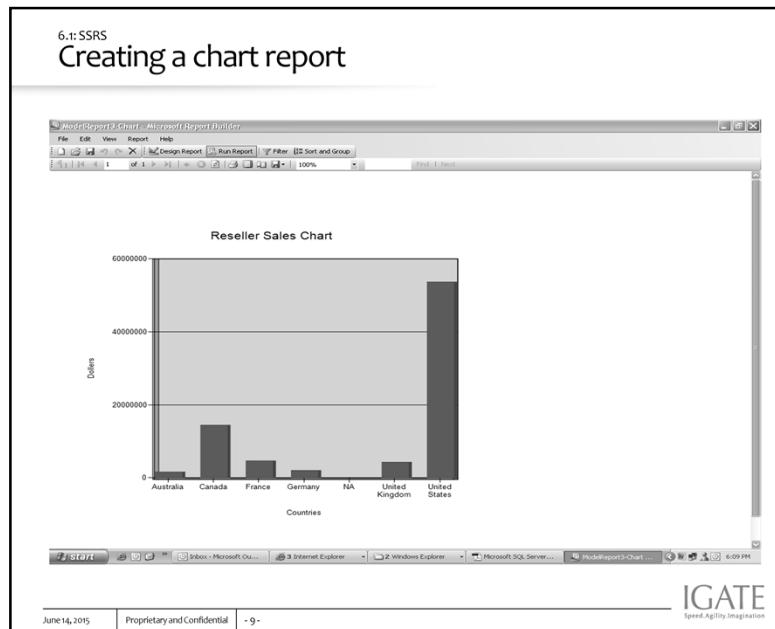
To create matrix report:

1. Create report as matrix layout.
2. Name the report as Matrix Report.
3. In Entities list, select Sales Territory, drag & drop Sales Territory Group, Sales Territory Country & Sales Region from the Fields list in Row Groups.
4. Select Order Date from Entities list, select Calendar Year from Fields list & drop it in Column Groups.
5. In the Fields list, expand Total Sales Amount to select related fields.
6. Select Total Sales Amount from Fact reseller sales entities list & drop it in values tab.
7. Save & Run the report.





1. To create chart report:
2. Click on File, New. Select Report Layout as Chart & click OK.
3. Title the report as 'Reseller Sales Chart'.
4. Click on chart to display data regions to drop fields on it.
5. From Entities list, select Sales Territory, drag & drop Sales Territory Country to Category field.
6. Drag & drop Total Sales Amount from Fields list to Data Value Fields.
7. Save & Run the report.



1. To format a graph with 3-D effect:
2. Open the existing chart report.
3. Select chart area, right click, select Chart Options.
4. Select Title tab, provide the chart title. Change Category title of X axis to Countries & Value title of Y axis to Dollars.
5. Select Legend tab, uncheck show legend check box.
6. For 3-D effects, go to 3-D effect tab, check the check box for Display chart with 3-D effect.
7. Assign following values to respective properties viz, Horizontal – 5 degree, Vertical – 0, Perspective – 0%, Wall thickness – 10%.
8. Click OK.

6.1: SSRS  
Formatting a report

Sales Territory Group	Sales Territory	Sales Territory Region	Total Sales Amount	Avg Sales Amount
Europe	France	France	460757.956	1305
		<b>Total</b>	460757.956	1305
	Germany	Germany	1099998.293	1099
		<b>Total</b>	1099998.293	1099
United Kingdom	United Kingdom	United Kingdom	4279008.826	1216
		<b>Total</b>	4279008.826	1216
	<b>Total</b>	1067054.799	1223	
		NA	NA	
North America	Canada	Canada	1437925.596	1256
		<b>Total</b>	1437925.596	1256
	United States	Central	7990008.1777	1360
		Northeast	6932942.0148	1193
Pacific	Australia	Northeast	12495076.0001	980
		Southeast	7867416.2251	1325
	<b>Total</b>	1594335.3767	931	
		Australia	1594335.3767	931
<b>Total</b>	6945956.9823	1322		

Total Dim Sales Territories: 11  
Filter: All Dim Sales Territories

June14,2015 | Proprietary and Confidential | -10-

IGATE  
Speed. Agility. Innovation.

1. To format a chart:
2. Open the existing tabular report.
3. Add Avg Sales Amount to the report.
4. Select Total cell & select format.
5. Change text alignment property for Horizontal to Center & Vertical to Middle.
6. Change font size to 12 & font type to Bold.
7. Select Border tab, for presets, choose outline option.
8. For Fill tab, select option as 'Light ocean'.
9. Select Avg Sales Amount, right click & select format, select Number tab.
10. Select Defined Format as 1234.56 & change Decimal Places to 0.
11. Click OK.
12. Save & Run the report.

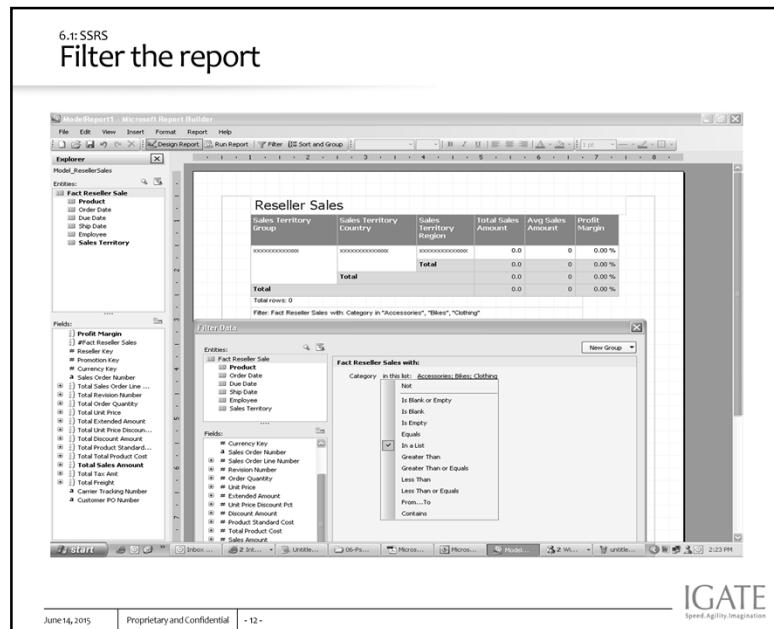
6.1: SSRS  
Adding formulae

Sales Territory Group	Sales Territory	Sales Region	Total Sales Amount	Avg Sales Amount	Profit Margin	
Europe	France	France	460752.950	1305	-0.81 %	
		<b>Total</b>	460752.950	1305	-0.81 %	
	Germany	Germany	198909.073	1079	-4.41 %	
	<b>Total</b>	198909.073	1079	-4.41 %		
United Kingdom	United Kingdom	United Kingdom	427900.826	1216	0.15 %	
		<b>Total</b>	427900.826	1216	0.15 %	
		<b>Total</b>	1081604.769	1223	-1.71 %	
NA	NA	NA			NA	
		<b>Total</b>			NA	
		<b>Total</b>			NA	
North America	Canada	Canada	1437765.595	1256	1.24 %	
			<b>Total</b>	1437765.595	1256	1.24 %
	United States	Central	790000.177	1360	1.78 %	
			<b>Total</b>	790000.177	1360	1.78 %
		Northeast	659200.0148	1193	1.11 %	
			<b>Total</b>	659200.0148	1193	1.11 %
		Northwest	1249500.0001	1590	1.78 %	
			<b>Total</b>	1249500.0001	1590	1.78 %
		Southeast	796716.2051	1325	1.53 %	
			<b>Total</b>	796716.2051	1325	1.53 %
Southwest	1084640.7925	1380	-0.08 %			
	<b>Total</b>	1084640.7925	1380	-0.08 %		
	<b>Total</b>	5306700.216	1381	1.01 %		
	<b>Total</b>	6795752.807	1353	1.06 %		
Pacific	Australia	Australia	1594325.3767	931	-4.82 %	
			<b>Total</b>	1594325.3767	931	-4.82 %
		<b>Total</b>	1594325.3767	931	-4.82 %	
	<b>Total</b>	80405094.9623	1322	0.58 %		
<b>Total On Sales Territories: 11</b>						
Filter: All On Sales Territories						

June14,2015 | Proprietary and Confidential | - 11 -

IGATE  
Speed Agility Inspiration

1. To add formula fields:
2. Open existing report.
3. Click Design Report.
4. Click on New Field Button.
5. In Define Formula dialog box, name the text box as ‘Profit Margin’.
6. Select the formula for fields as  $\text{sum}(\text{SalesAmount} - \text{TotalProductCost}) / \text{Sum}(\text{SalesAmount})$
7. Click on OK.
8. Drag Total Product Cost, Avg Sales Amount & Profit margin to the report.
9. Change the format of Profit Margin to 1,234.56% & click OK.
10. Save & Run the report.



1. To add filter on report:
2. Open the report.
3. Click on Design Report.
4. Select Filter option on toolbar.
5. Filter dialog box appears. Select Fact Reseller Sales from Entities.
6. Select Product from Entities & select Category from Fields list.
7. Dialog box appears for Filter criteria. Select operator as 'In List' and choose values from available list of Category.
8. Click OK.
9. Save & Run the report.

6.1: SSRS  
Sorting data

The screenshot shows the Microsoft Report Builder interface. On the left, the 'Fields' pane lists various data items such as Profit Margin, Fact Reseller Sales, Reseller Key, Product Key, Order Date, Order Type, Ship Date, Employee, and Sales Territory. The main area displays a table titled 'Reseller Sales' with columns: Sales Territory Group, Sales Territory Country, Sales Territory Region, Total Sales Amount, Avg Sales Amount, and Profit Margin. The table has three rows: a header row, a 'Total' row, and another 'Total' row at the bottom. A 'Sort' dialog box is open over the report, showing 'Sales Territory Group' selected as the primary sort item with 'Ascending' selected. It also shows 'Avg Sales Amount' and 'Profit Margin' as secondary sort items, both with 'Descending' selected. The dialog also includes options for 'Then by' and 'Page break at'.

1. To sort data on report:
2. Open an existing report.
3. In Design report, click on Sort and Group tab.
4. In Sort dialog box, select Group as Sales Territory Group, select Ascending order in radio button.
5. Multiple sorts can be applied using next options for Then by & select the sorting column from the list.
6. Click OK.
7. Save & Run the report.

## Summary

- Business users can easily create ad hoc reports using Report Builder.
- Ad hoc reports are available in table, matrix and chart layout.
- Filtering, Sorting, formula can be applied on reports.
- Default drill through option is available with report builder.



# SQL Server Reporting Services 2008

## Lesson 7: Working with Charts

June 14, 2015

Proprietary and Confidential

- 1 -

IGATE  
Speed. Agility. Imagination.

## Lesson Objectives

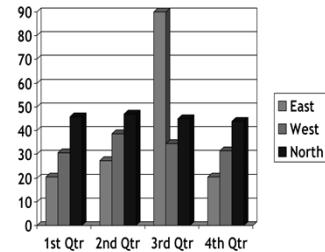
- Creating charts
- Chart elements
- Chart types – Column, line, bar, area, scatter
- Creating column charts
- Formatting chart



7.1: Creating Charts

## Creating Charts

- Charts are better way for visualizing summarized data.
- There are different types of charts available that gives data in pictorial view specially for numerical data.
- Report can be generated only as chart report or can be combined with other data regions like table or so.



7.2: Chart elements

## Chart elements

➤ There are various chart elements like

- Chart title
- Value axis
- Category axis
- Series
- Legend

June 14, 2015 | Proprietary and Confidential | - 4 -

IGATE Speed.Aggility.Imagination

Chart elements are essential part of charts.

1. Chart title – This is default chart element, can be removed from chart. Its moveable, its properties for look and feel can be changed. Actions can be associated with chart title.
2. Value axis – It is known as Y-axis, the horizontal scale in column chart. This is used to display numeric values of chart. Title can be added or removed. Appearance of this axis can be changed.
3. Category axis – This is known as X-axis, vertical scale on chart. It is used for grouping numerical data values. Look and feel can be customized.
4. Series – Its an optional element of chart. This facilitates another level of grouping. When you have to group multiple numerical data values, series is used. Filter can be applied to series.
5. Legend – This gives information about different axis values on chart. This can be also customized.

7.3: Chart types

## Chart types

➤ **Chart type defines different ways to represent data as per requirement.**

- Column chart
- Line chart
- Bar chart
- Area chart
- Scatter chart
- Pie chart
- Pyramid chart
- Range chart

June 14, 2015 | Proprietary and Confidential | - 5 -

IGATE Speed.Agility.Imagination

Different types of charts are available to display data in different look and feel.

1. Column chart – This is one of the very common chart types to present data values in vertical scale against multiple value series.
2. Line chart – Line charts basically used for business reports. It gives values on X-axis for relevant data horizontally. Line chart can be combined with column chart, but not with other charts.
3. Bar chart – It is similar to column chart, displays like bars. It displays horizontal bars. Perpendicular axis is category axis and horizontal axis is the value axis.
4. Area chart – This chart displays the series as a data points. If report contains any blank data, area charts are suitable.
5. Scatter chart – it takes ratio values to plot a chart.
6. Pie chart – It is seen as circular diagram.
7. Pyramid chart – It is also called as funnel chart. It is used to display percentage wise whole summary for each category. This chart gives best good summary if the data is sorted.
8. Range chart – Range chart considers high and low values for each category.

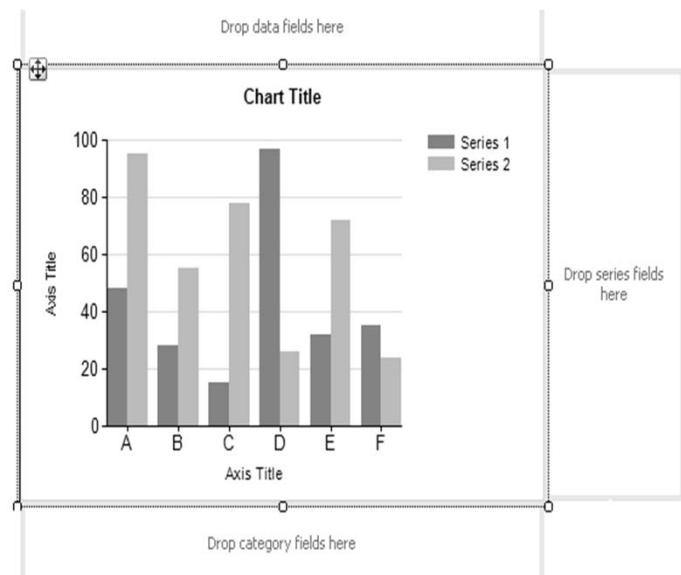
7.4: Creating Charts

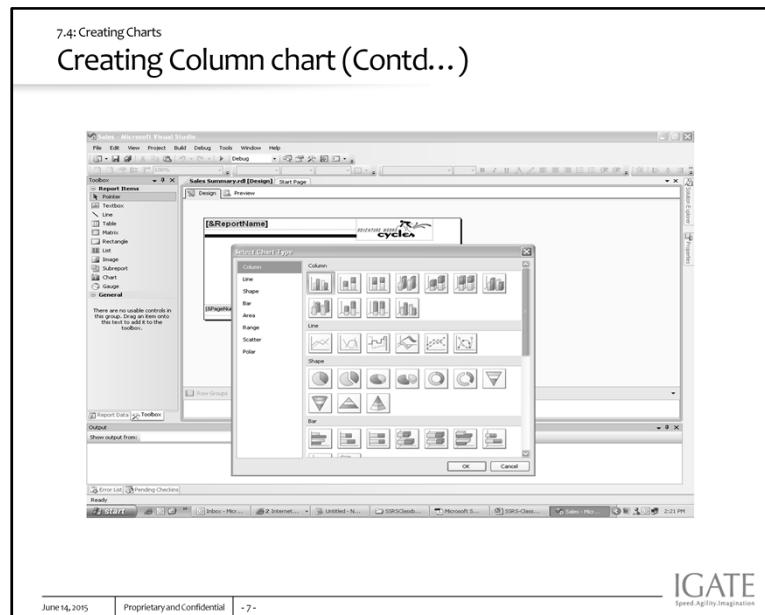
## Creating Column chart

- Column chart is very simple and commonly used chart.
- Simple column chart can be created by dragging chart control from toolbox and dropping it on the report.
- There are 3 sections on chart to divide data to be displayed viz., data fields, category fields and series fields.
- Category section is used to group data horizontally across the chart.
- Optional Series section appears on right side of the report to add next level of grouping.

June 14, 2015 | Proprietary and Confidential | - 6 -

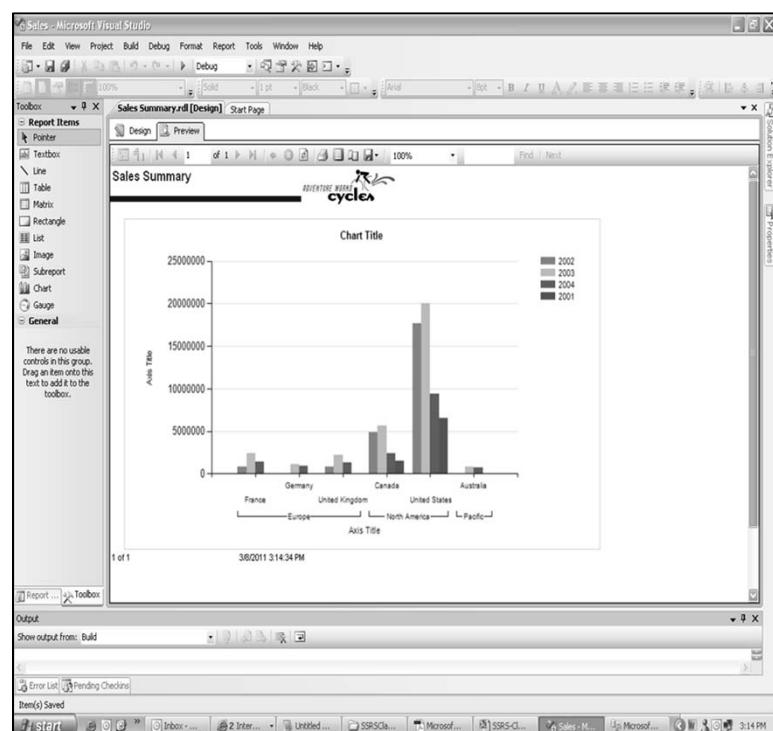
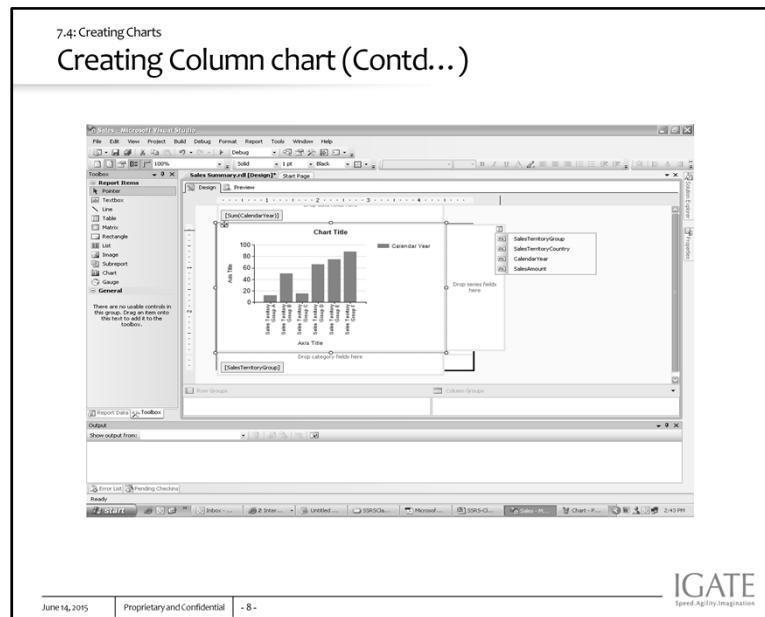
IGATE Speed.Aggility.Imagination

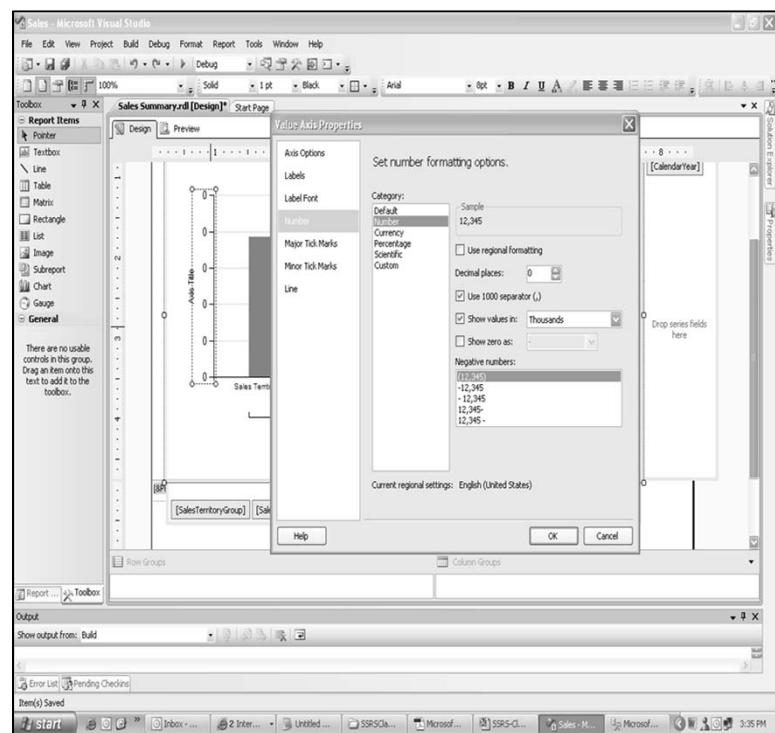
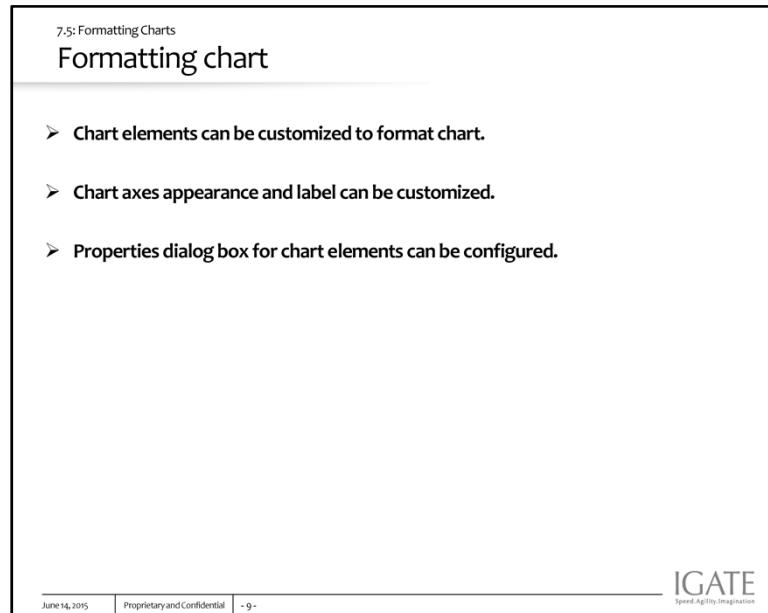


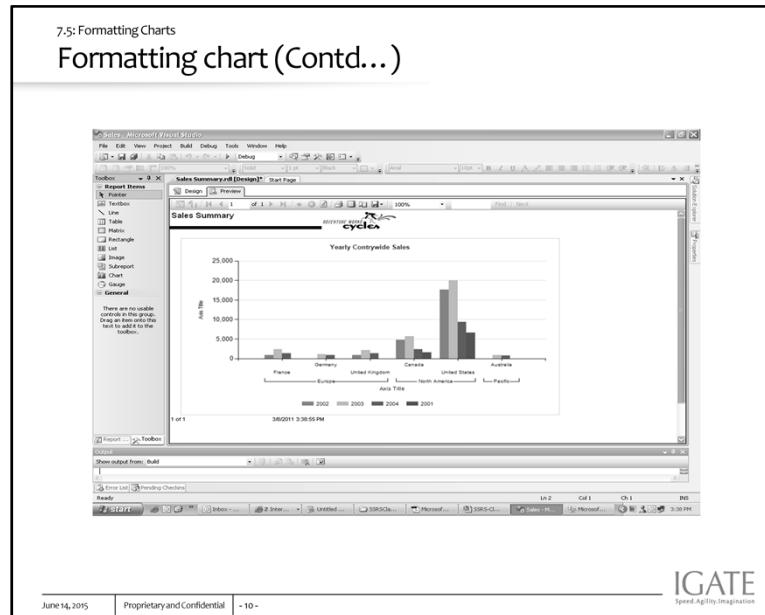


To create column chart:

1. Create a report.
2. Click on Design tab.
3. From toolbox, drag and drop chart control on report.
4. Select Chart Type dialog box, click OK.
5. select default chart type as column chart.
6. Select Category field as Year,







To format chart elements:

1. Select chart on report.
2. In Properties window, select Size category & configure width to 6in and height to 7in.
3. Right click legend and select legend properties.
4. Select Legend Position to bottom center.
5. Configure legend properties for visibility, font, border, shadow.
6. Change Chart Title name to Sales in Thousand. Change Font size and color.

## Summary

- Charts gives data in pictorial way.
- Charts simplify complex data in a report.





## **SQL Server Reporting Services 2008 Lab Book**

---

Copyright © 2011 iGATE Corporation. All rights reserved. No part of this publication shall be reproduced in any way, including but not limited to photocopy, photographic, magnetic, or other record, without the prior written permission of iGATE Corporation.

iGATE Corporation considers information included in this document to be Confidential and Proprietary.

## Document Revision History

---

Date	Revision No.	Author	Summary of Changes
25-Jan-2013	1	Ajit Jog	Content Creation

---

**Table of Contents**

---

Creating a Tabular Product Listing Report .....	4
Creating a drill down group report.....	7
Using Subreport .....	10
Creating a Report using Chart .....	13
Creating a Hierarchical Report.....	15
Customizing Report Model and using Report Builder .....	17

<b>Lab 1</b>	<b>Creating a Tabular Product Listing Report</b>
<b>Description</b>	We will create a report which will show product details grouped according product category and add category wise subtotals/
<b>Objective</b>	To learn <ul style="list-style-type: none"> <li>1. How to create Shared Data Source and Local Data Source</li> <li>2. Define Report DataSet</li> <li>3. Use Data Set and Table report item to create tabular report</li> <li>4. Define a row group</li> <li>5. Add Sub totals to group</li> <li>6. Add Expressions using Data Set fields</li> </ul>

1. Open BIDS IDE, Start New SSRS Project
  - a. File -> New Project , select Business Intelligence Project, Select Report Server Project template, Give Project Name IgateSSRSReportsDemo, click ok
2. Create Shared Data Source
  - a. In Solution Explorer , right click Shared Data Sources, click Add New Data Source
    - i. Type: MS SQL
    - ii. For Connection String, click edit.. specify the credentials and other details.
3. Right click Reports folder in Soln Explorer, click Add -> New Item.
  - a. Select report item template
  - b. Give ProductsReport as report name
4. In Report Data Window,
  - a. Click New Data Source and create a local report level data source based on Shared Data Source of step 2
  - b. Click New Data Set
    - i. Create a dataset on local data source created in step 4.a
    - ii. The Select Query for DataSet is:-

```

SELECT      Products.ProductID, Products.ProductName,
Products.UnitPrice, Products.UnitsInStock, Products.ReorderLevel,
Categories.CategoryName,
          Categories.CategoryID
FROM        Products INNER JOIN
          Categories ON Products.CategoryID = Categories.CategoryID
ORDER BY    Categories.CategoryName, Products.ProductName

```

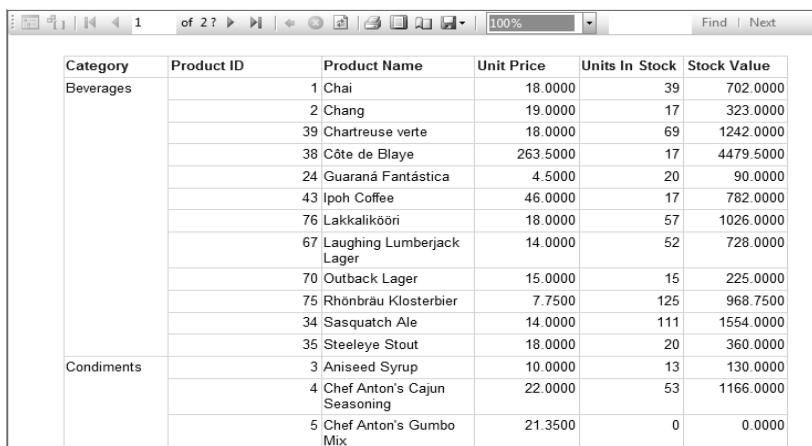
    - iii. Click ok.
5. From toolbox, drag a Table report item onto report designer surface.
6. Right click the column header of 3<sup>rd</sup> column in table report item and insert a new column.
7. Drag ProductId, ProductName, UnitPrice, UnitsInStock dataset fields onto the table columns. Set dark blue color, bold 10pt font size for column headers and for fields set fontsize 9pt

DataSource1

- DataSetProducts
  - ProductID
  - ProductName
  - UnitPrice
  - UnitsInStock
  - ReorderLevel
  - CategoryName
  - CategoryID

	Product ID	Product Name	Unit Price	Units In Stock
	[ProductID]	[ProductName]	[UnitPrice]	[UnitsInStock]

8. Right click the column header of 4<sup>th</sup> column in table report item and insert a new 5<sup>th</sup> column.
  9. Type Stock Value in the 5<sup>th</sup> column header. Right click the textbox cell below the 5<sup>th</sup> column header, select expression  and in the expression dialog box type following expression and click ok
- =Fields!UnitsInStock.Value\*Fields!UnitPrice.Value
10. Right click the ProductId field textbox, select Add Group, under Row Group – select Parent Group.
  11. In the Group dialog box
    - a. give group name as Category
    - b. select [CategoryName] from the drop down. Click ok
  12. Click Preview Tab the output should be as below:



The screenshot shows a report preview window with a toolbar at the top. The main area displays a table with the following data:

Category	Product ID	Product Name	Unit Price	Units In Stock	Stock Value
Beverages	1	Chai	18.0000	39	702.0000
	2	Chang	19.0000	17	323.0000
	39	Chartreuse verte	18.0000	69	1242.0000
	38	Côte de Blaye	263.5000	17	4479.5000
	24	Guaraná Fantástica	4.5000	20	90.0000
	43	Ipooh Coffee	46.0000	17	782.0000
	76	Lakkalikööri	18.0000	57	1026.0000
	67	Laughing Lumberjack Lager	14.0000	52	728.0000
	70	Outback Lager	15.0000	15	225.0000
	75	Rhönbräu Klosterbier	7.7500	125	968.7500
Condiments	34	Sasquatch Ale	14.0000	111	1554.0000
	35	Steeleye Stout	18.0000	20	360.0000
	3	Aniseed Syrup	10.0000	13	130.0000
	4	Chef Anton's Cajun Seasoning	22.0000	53	1166.0000
	5	Chef Anton's Gumbo Mix	21.3500	0	0.0000

13. Right Click CategoryName field textbox in the report table, Insert Row -> Inside Group- Below. Add such total 3 blank rows.
  14. In the middle row under stock value , right click the cell textbox and give following expression
- =Sum(Fields!UnitsInStock.Value\*Fields!UnitPrice.Value)
15. Select the 2 adjacent cells and right click and merge and then type the following:

### Total Stock Value for Category

16. Drag a TextBox from toolbox on the top of table and give “Inventory Products Listing” as report heading and format it.
17. Click Preview tab to see the result

Inventory Products Listing

Category	Product ID	Product Name	Unit Price	Units In Stock	Stock Value
Beverages	1	Chai	18.0000	39	702.0000
	2	Chang	19.0000	17	323.0000
	39	Chartreuse verte	18.0000	89	1242.0000
	38	Côte de Blaye	263.5000	17	4479.5000
	24	Guaraná Fantástica	4.5000	20	90.0000
	43	Ipoh Coffee	46.0000	17	782.0000
	76	Lak Lakiköri	18.0000	57	1026.0000
	67	Laughing Lumberjack Lager	14.0000	52	728.0000
	70	Outback Lager	15.0000	15	225.0000
	75	Rhönbräu Klar terbier	7.7500	125	968.7500
	34	Sasquatch Ale	14.0000	111	1554.0000
	35	Steeleye Stout	18.0000	20	360.0000
					Total Stock Value for Category
					12480.2500
Condiments	3	Aniseed Syrup	10.0000	13	130.0000
	4	Chef Anton's Cajun Seasoning	22.0000	53	1166.0000
	5	Chef Anton's Gumbo Mix	21.3500	0	0.0000
	15	Genen Shouyu	15.6000	39	604.5000
	6	Grandma's Boysenberry Spread	25.0000	120	3000.0000

<b>Lab 2</b>	<b>Creating a drill down group report</b>
<b>Description</b>	We will create a report which will show the orders placed by customers from various countries and it's cities between specific dates
<b>Objective</b>	To learn <ol style="list-style-type: none"> <li>1. How to create multiple nested groups.</li> <li>2. How to add drill down feature.</li> <li>3. Use Parameters for filtering.</li> <li>4. Dynamic Report heading through expression</li> </ol>

1. Add a new blank report
  - a. Right click Reports folder in Soln Explorer, click Add -> New Item.
  - i. Select report item template
  - ii. Give DrillDownReportDemo as report name
2. Create a local data source based on shared data source and define a data set using following select statement:

```

SELECT Customers.CustomerID, Customers.CompanyName, Customers.Country,
Customers.City, Orders.OrderID, Orders.OrderDate, Orders.Freight
FROM Customers INNER JOIN
Orders ON Customers.CustomerID = Orders.CustomerID
ORDER BY Customers.Country, Customers.City, Customers.CompanyName
  
```

3. Drag a table report item.
4. Drag Orderid, OrderDate and Freight fields on to table columns.
5. Right click OrderId field , Add Group , Parent Group under Row Group
  - a. Select companyname field in drop down
6. Similarly Right click CompanyName field and a parent row group using field City.
7. Again Right click City field and a parent row group using field Country
8. Now in Row Groups pane below double click every group and rename the groups as grpcompanyname, grpcity and grpcountry
9. Similarly change the column header for those group fields in the table.
10. Add Subtotal to OrderId at company level.
  - a. Right click orderid field, Select Add Total
  - b. Aggregation will be added for orderid and freight.
  - c. Right click and edit the aggregate expresion for OrderId and change it to:

= "Total Orders " + cstr(Count(Fields!OrderID.Value))

11. Similarly right click companyname and city add subtotals for it. and change sum(orderid) expression to Count(Orderid)
12. Format the table as below:-

Country	City	Customer	Order ID	Order Date	Freight
[Country]	[City]	[CompanyName]	[OrderID]	[OrderDate]	[Freight]
		«Expr»			
		Total	[Count(Orde]		[Sum(Freight)]
	Total		[Count(Orde		[Sum(Freight)]

13. Select the City and Country Field textboxes and note the textbox names.
14. Double click grpcompanyname group in row groups pane, goto visibility tab, select hide, check Display can be toggled checkbox and select City Field textbox name from the drop down. Click.
15. Similarly for grpcity group set the Country Field Textbox name for toggle.
16. Preview the report.

Country	City	Customer	Order ID	Order Date	Freight
Argentina	Total		16		598.5800
Austria	Total		40		7391.5000
Belgium	Total		19		1280.1400
Brazil	Total		83		4880.1900
Canada	Total		30		2198.0900
Denmark	Total		18		1396.1900

17. Switch to design tab, right click data set select Query and change the existing query to:- (note the WHERE clause)

```

SELECT Customers.CustomerID, Customers.CompanyName, Customers.Country,
Customers.City, Orders.OrderID, Orders.OrderDate, Orders.Freight
FROM Customers INNER JOIN
Orders ON Customers.CustomerID = Orders.CustomerID
WHERE (Orders.OrderDate BETWEEN @fromdate AND @todate)
ORDER BY Customers.Country, Customers.City, Customers.CompanyName

```

18. Note in report data window 2 parameters will be created.
19. Double click @fromdate parameter
  - a. Give prompt as Select Start Order Date:-
  - b. Set data type to datetime
  - c. Goto Default Values tab, select specify values
  - d. Click add Type 01/01/1996, click ok
20. Similarly configure @todate parameter
  - a. Prompt: Select End Order Date:-
  - b. Default value 01/01/1998
21. From the toolbox drag a textbox on top of table. Set arial, 14 pt bold size and dark maroon color. Right click the textbox and set following expression :  
`= "Orders Placed by Customers between " +  
cstr(Parameters!fromdate.Value) + " and " +  
cstr(Parameters!todate.Value)`

22. Click preview tab, You can expand the country and the city to get the further sales details as shown below:-

Select Start Order Date:-	<input type="text" value="01-01-1996"/> 	Select End Order Date	<input type="text" value="01-01-1998"/> 
                                               <img			

### Orders Placed by Customers between 1/1/1996 and 1/1/1998

Country	City	Customer	Order ID	Order Date	Freight
✉ Argentina	Total		6		117.6600
✉ Austria	Total		29		5000.7400
✉ Belgium	Total		9		518.0500
✉ Brazil	✉ Campinas	Total	7		309.0600
	✉ Resende	Total	6		131.7400
	✉ Rio de Janeiro	Total	22		945.1100
	✉ Sao Paulo	Total	21		2068.8600
	Total		56		3454.7700
✉ Canada	Total		22		1828.4000

<b>Lab 3</b>	<b>Using Subreport</b>
<b>Description</b>	We will create 2 reports one which will list product categories and other will display products. The products report will be embedded or added as a subreport in the product categories report.
<b>Objective</b>	To learn <ol style="list-style-type: none"> <li>1. How to add another report as a subreport</li> <li>2. How to link subreport to parent report via parameters</li> </ol>

1. Create a report named ProductCategoriesListing which displays the category details in a table as shown below:

Category ID	Category Name	Description
[CategoryID]	[CategoryName]	[Description]

2. The DataSet Query is as below:

```

SELECT CategoryID, CategoryName, Description
FROM Categories
ORDER BY CategoryName
  
```

3. Preview the report  
 4. Create another report named ProductsListing which will show product details in a table as shown below:-  
 Note: Set the dimensions of report body so as it matches the dimensions of table report item.

Product ID	Product Name	Unit Price
[ProductID]	[ProductName]	[UnitPrice]

5. The Data Set Query of this report:-

```

SELECT ProductID, ProductName, UnitPrice,
       UnitsInStock, CategoryID
FROM Products
WHERE (CategoryID = @catid)
ORDER BY ProductName
  
```

6. In the report data window goto @catid parameter properties  
 a. Set data type: integer  
 b. Select parameter visibility to hidden  
 c. Goto Default Values tab, and specify default value as 1  
 7. Preview the report to ensure that it always displays category id 1 products.  
 8. Open ProductCategoriesListing report.  
 a. Right click the field row in left grey area, and add a row below.Add such 2 blank rows.

Category ID	Category Name	Description
[CategoryID]	[CategoryName]	[Description]
Insert Row	Above	
Delete Rows	Below	

- b. Select the cells of 1<sup>st</sup> blank row and merge the cells.
- c. Select the cells of 2<sup>nd</sup> blank row and merge the cells.
- d. In the 2<sup>nd</sup> row drag Subreport report item from toolbox.
- e. Right Click the subreport and in the subreport properties
  - i. Select ProductsList as subreport.
  - ii. Goto parameters tab, click add and set as shown below:

Use these parameters to run the subreport

Add	Delete	Up	Down
Name	Value		
catid	[CategoryID] <input type="button" value="fx"/>		

- 9. Right click column header row and add a blank row above
- 10. Right click subreport row and add a blank row below.
- 11. Report design is as below:-

Category ID	Category Name	Description
[CategoryID]	[CategoryName]	[Description]
ProductsList		

- 12. Preview the report

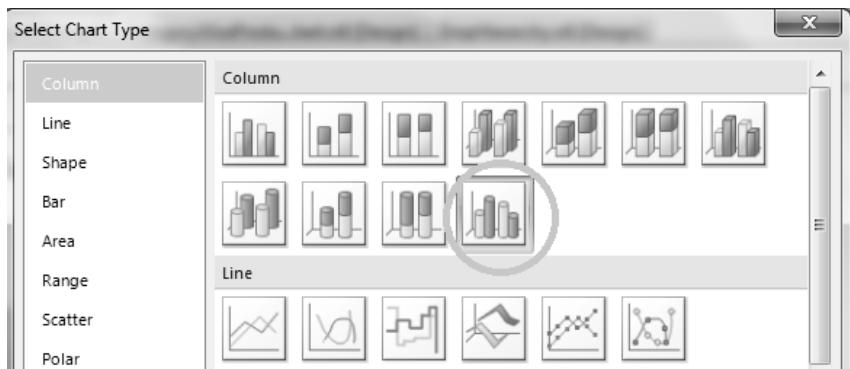
Category ID	Category Name	Description
1	Beverages	Soft drinks, coffees, teas, beers, and ales
Product ID	Product Name	Unit Price
1	Chai	18.0000
2	Chang	19.0000
39	Chartreuse verte	18.0000
38	Côte de Blaye	263.5000
24	Guaraná Fantástica	4.5000
43	Ipoh Coffee	46.0000
76	Lakka likööri	18.0000
67	Laughing Lumberjack Lager	14.0000
70	Oubak Lager	15.0000
75	Rhône Valley Ristorante	7.7500
34	Sasquatch Ale	14.0000
35	Steeleye Stout	18.0000

<b>Lab 4</b>	<b>Creating a Report using Chart</b>
<b>Description</b>	We will create a report which will have a Column Chart which will display Region wise customer distribution
<b>Objective</b>	To learn 1. How to use Chart to display the Information

1. Add a new blank report
  - a. Right click Reports folder in Soln Explorer, click Add -> New Item.
    - i. Select report item template
    - ii. Give RegionWiseCustomerDistribution as report name
2. Create a local data source based on shared data source and define a data set using following select statement:

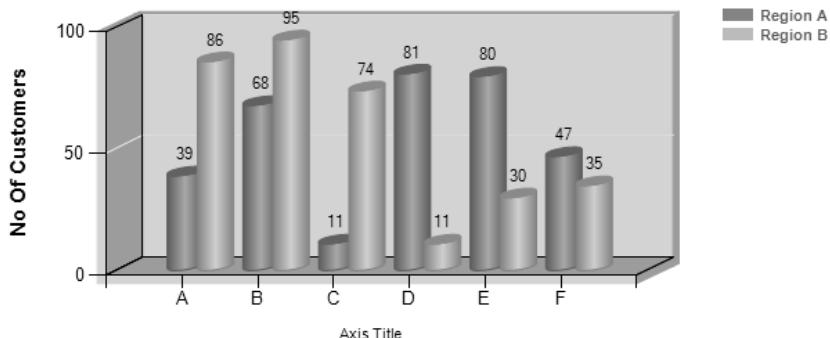
```
SELECT      Region, COUNT(*) AS NOOFCUSTOMERS
FROM        Customers
WHERE       Region is not null
GROUP BY    Region
```

3. Drag a chart and select 3D chart type.



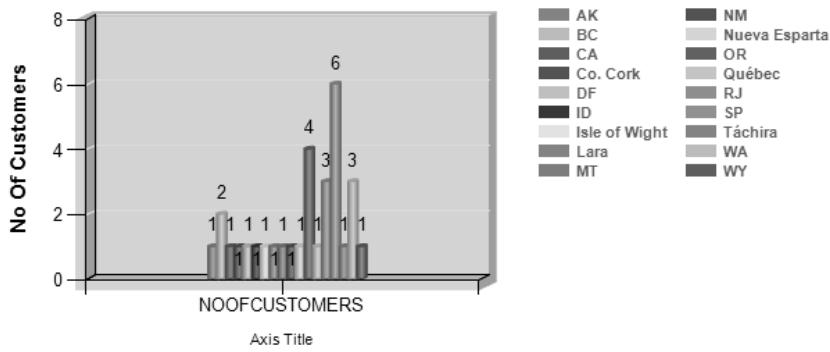
4. Drop “NOOFCUSTOMERS” field on Data Field section and Region Field on Series section
5. Rt click chart in the middle and select show data labels.
6. Change the Chart Title and Axis title as below

### Region Wise Customer Distribution



- Run the report.

### Region Wise Customer Distribution



<b>Lab 5</b>	<b>Creating a Hierarchical Report</b>
<b>Description</b>	We will create a report which will show employees listed according to their reporting hierarchy
<b>Objective</b>	To learn 1. How to create a hierarchical report.

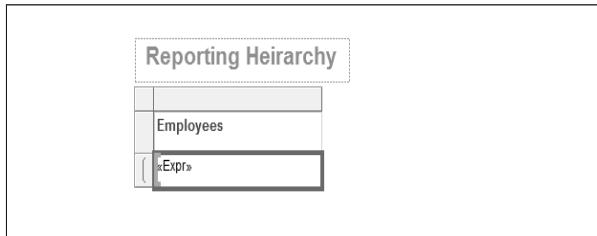
1. Add a new blank report
  - a. Right click Reports folder in Soln Explorer, click Add -> New Item.
    - i. Select report item template
    - ii. Give EmpHierarchyReport as report name
2. Create a local data source based on shared data source and define a data set using following select statement:

```
SELECT EmployeeID, LastName, FirstName, ReportsTo
FROM Employees
```

3. Drag a table report item. Delete all columns except one
4. Right click the only cell in the Details Row and write following expression

**=trim(Fields!LastName.Value) + " " +  
Fields!FirstName.Value**

5. Give Employees as column heading and drag texbox and provide report header as below



6. Rt Click the expression TextBox above, goto Alignment TAB, and write following expr for Left padding option  
**=CStr(5 + Level()\*16) + "pt"**
7. Double click the details pane in Row Groups.
8. Add a group on EmployeeId field

Name:

Group expressions:

Group on: [EmployeeID]

9. Goto Advanced TAB, and set recursive parent to ReportsTo field.
10. Run the report.



## Reporting Hierarchy

Employees
Fuller Andrew
Davolio Nancy
Leverling Janet
Peacock Margaret
Buchanan Steven
Suyama Michael
King Robert
Dodsworth Anne
Callahan Laura

<b>Lab 6</b>	<b>Creating and Customizing Report Model and using Report Builder</b>
<b>Description</b>	We will create a Report Model based on AdventureWorks Database tables and use it in Report Builder to create a report.
<b>Objective</b>	To learn <ul style="list-style-type: none"> <li>1. How to Create and Customize Report Model</li> <li>2. How to create report using Report Builder</li> </ul>

1. In the Microsoft Visual Studio window, click File, point to New, and then click Project.
2. Click Report Model Project. In the Name box, type Adventure Works Model. Click OK.
3. In Solution Explorer, right-click Data Sources, and then select Add New Data Source connecting AdventureWorks database
4. In Solution Explorer, right-click the Data Source Views folder and select Add New Data Source View.
5. On the Select Tables and Views page, select all the tables in the Available objects area, and then click the arrow button.
6. In Solution Explorer, right-click Report Models and then select Add New Report Model.
7. The Report Model Wizard opens.
8. On the Welcome to the Report Model Wizard page, click Next.
9. On the Select Data Source View page, verify that the AdventureWorks data source view you created in the previous procedure is selected.
10. Click Next.
11. In the Select model language drop-down list of the Select report model generation rules page, select a language for the model.
12. The language you select tells Model Designer which language-specific code it should use when generating user-friendly names for the tables and columns in your database. After you complete the wizard, you can set the Culture property on the model.
13. Accept the default rules.
14. Click Next.
15. On the Collect Model Statistics page, verify that the Update model statistics before generating option is selected, and then click Next.

16. On the Completing the Wizard page, verify that AdventureWorks appears in the Name box, and then click Run.
17. The report model is created.
18. When the wizard is finished, click Finish.
19. The AdventureWorks.smdl file appears in Solution Explorer. You have successfully created a report model. In the next procedure, you will explore the model.
20.  To explore the report model
21. In the Model pane, select the Customer entity.
22. View the center pane. The contents of the Customer entity appear. The attributes include: #Customers, #CustomerID, Account Number, Customer Type, and Modified Date.
23. In the List view, select the #Customers attribute, and then view the contents of the Properties window.
24. In the List view, select the Customer Type attribute, and then view the contents of the Properties window.
25. In the List view, select the Sales Order Headers role, and then view the contents of the Properties window.
26. On the File menu, click Save All.
27. Next, you will deploy the report model to the report server.
28.  To deploy the report model
29. On the Project menu, click AdventureWorks Model Properties.
30. After you confirm the deployment properties that you want to use, click OK.
31. On the Build menu, click Deploy AdventureWorks Model.
32. If the deployment is successful, a Deploy succeeded message is displayed in the Output window. To view the Output window, on the View menu, click Output. If the deployment fails, you need to troubleshoot the reported error.
33. In the next procedure, you will test the report model in Report Builder.
34.  To use the model in Report Builder
35. Open your browser.
36. In the Address bar, type <http://localhost/reports>.
37. The Home page for SQL Server Reporting Services opens.
38. On the Report Manager toolbar, click Report Builder.

39. In the Getting Started pane, select AdventureWorks and then click OK.
40. In the Entities list, select Sales Order Header.
41. From the Entities list, drag the Sales Person entity to the design area.
42. Notice that the Sales Person National ID Number and Commission Pct fields are added to the design area.
43. In the Fields list, double-click Total Total Due.
44. Notice that the Total Total Due field is not formatted as a monetary value.
45. To see the results of your query, click Run Report.
46. On the File menu, click Exit
47. To add a description to a report model
48. Return to the AdventureWorks report model project open in the Business Intelligence Development Studio window.
49. In the Tree view, select the Model node.
50. In the Properties window, locate the Description property.
51. In the Description box, type the following: This report model is for the AdventureWorks database.
52. In the Tree view, select the Sales Person entity, and then type the following in the Description box: All sales people are employees.
53. In the List view, select the Sales Quota attribute, and then type the following in the Description box: Sales quotas are updated quarterly.
54. On the File menu, click Save All.
55. To add a folder to the report model
56. In the Tree view, right-click Model, point to New, and then click Folder.
57. The new folder is added at the bottom of the list of entities.
58. Right-click the new folder, and select Rename.
59. Type Product Details.
60. You have successfully added a new folder. Next, to organize the contents of your model, you will add model items to the folder.
61.  To add entities to the folder
62. In the Tree view, verify that Model is selected.

63. In the List view, select all the entities that begin with the word Product, except for the Product entity itself.
64. Drag the selected entities to the Product Details folder in the Tree view.
65. You have successfully organized the Product entities within the model.
66. To view the contents of Product Details, click the folder.
67. To rename model items
68. In the Tree view, right-click the Sales Order Header entity, and then click Rename.
69. Type the following: Sales Order.
70. With the Sales Order entity selected, locate the Sales Order Details role in the List view.
71. Right-click the Sales Order Details role, and then click Rename.
72. Type the following: Sales.
73. In the List view, expand the Total Due attribute.
74. Right-click Total Total Due, and then click Rename.
75. Type the following: Sum Total Due.
76. In the Tree view, select the Sales Order Detail entity, and rename the entity Sales.
77. In the Tree view, select the Sales Person entity.
78. Notice that there is a role called Sales Person within the Sales Person entity. The name of this role is misleading because this role really links to the Employee entity.
79. Rename the Sales Person role to Employee.
80. On the File menu, click Save All.
81. To hide a model item
82. In the Tree view, select the Sales Order entity.
83. In the List view, select the Rowguid attribute.
84. In the Properties window, select the Hidden property.
85. In the properties drop-down list, select True.
86. On the File menu, select Save All.
87. In the Tree view, select the Sales Person entity.
88. In the Properties window, select DefaultDetailAttributes.

89. To edit the property, click the ellipsis (...) button.
90. In the Members list of the AttributeReference Collection Editor dialog box, select Commission Pct, and then click Remove.
91. Click Add.
92. The Default Detail Attributes dialog box appears.
93. In the Entities list, select the Employee entity and then select the Contact entity.
94. In the Fields list, select First Name, Middle Name, and Last Name, and then click OK.
95. Click OK again.
96. In the Properties window, note that the DefaultDetailAttributes property indicates that there are four attributes. When the user adds the Sales Person entity to their report, the sales person's National ID Number, and first, middle and last names are added.
97. On the File menu, click Save All.
98.  To deploy the report model
99. On the Build menu, click Deploy AdventureWorks Model.
100.  To view report model changes in Report Builder
101. Launch Report Builder.
102. In the Getting Started pane, select the AdventureWorks model, and then click OK.
103. To see the Product Details folder, scroll down the list of entities.
104. This folder was created in Lesson 3.
105. To see the contents of the Product Details folder, click Product Details.
106. The entities in this folder were added to the folder in Lesson 3.
107. In the Entities list, find the Sales Orders and Sales entities.
108. These entities were renamed in Lesson 4.
109. Select the Sales Orders entity, and then drag the Sum Total Due field to the design area.
110. This field was renamed in Lesson 4 and formatted in Lesson 7.
111. Drag the Sales Person entity to left of the Sum Total Due field.
112. Notice that the National ID Number, First Name, Middle Name, and Last Name fields are added to the report. These are the default detail attributes that you added to the

entity in this lesson. This is the same report that you created earlier except that now your new model changes are displayed.

113. To view the results of your report, click Run Report.

114. To return to the design area, click Design Report.