Course Guide

IBM Cognos Analytics: Author Reports Advanced (v11.0)

Course code B6059 ERC 2.0

IBM Training

April, 2016

NOTICES

This information was developed for products and services offered in the USA.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service. IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive, MD-NC119
Armonk, NY 10504-1785
United States of America

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law: INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM websites are provided for convenience only and do not in any manner serve as an endorsement of those websites. The materials at those websites are not part of the materials for this IBM product and use of those websites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

TRADEMARKS

IBM, the IBM logo, ibm.com and Cognos are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at www.ibm.com/legal/copytrade.shtml.

Adobe, and the Adobe logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

Microsoft, Windows, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Contents

Course overview

Preface overview

This course teaches Professional Report Authors about advanced report building techniques using relational data models, and ways of enhancing, customizing, and managing professional reports. The course builds on topics presented in the Fundamentals course. Attendees will participate in interactive demonstrations and exercises that illustrate key concepts while learning how to use advanced features in the product builds on topics learned in the Fundamentals course.

Intended audience

Report authors

Topics covered

Topics covered in this course include:

- Create query models
- Create reports based on query relationships
- Create advanced dynamic reports
- Design Effective Prompts
- Create additional advanced reports
- Examine the report specification
- Distribute reports through bursting
- Enhance user interaction with HTML
- Explore the Prompt API (Optional)

Course prerequisites

Participants should have:

- knowledge of business requirements
- completed B6088: IBM Cognos Analytics for Consumers (v11.0) - eLearning
- completed B6058: IBM Cognos Analytics: Report Studio: Author Reports Fundamentals (v11.0)

Course Environment

The environment provided in this course requires the following services to be started before you begin performing demonstrations and exercises:

- Apache Directory Server
- DB2 -DB2COPY 1 - DB2
- DB2DAS - DB2DAS00
- IBM Cognos
- Lotus Domino Server (CProgramFilesx86IBMLotusDominodata)
- World Wide Web Publishing Service

To review the services, on the Taskbar of your environment, click the Services icon, and ensure that the above services are running. If you have closed your image and launched it again, it is a best practice to review the status of the services before continuing with your demonstrations and exercises.

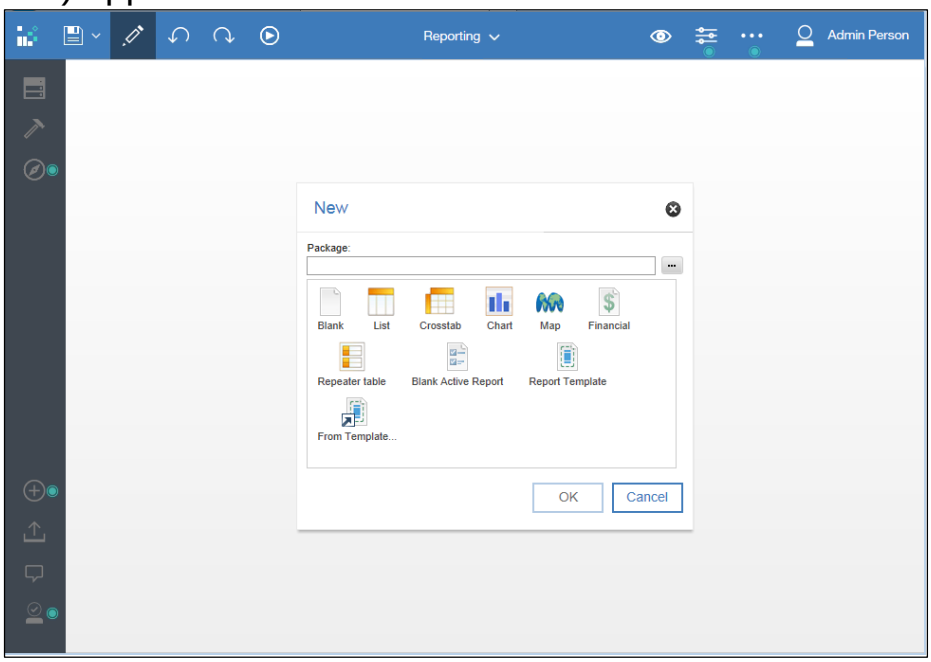
If the Apache Directory Server or DB2 -DB2COPY 1 - DB2 service have stopped, you will need to stop the IBM Cognos service, start the stopped service(s), and then start the IBM Cognos service once the previously stopped service(s) has (have) started successfully. You can start and stop a specific service by double-clicking the service to open the Properties dialog box, and then clicking the Stop or Start buttons.

Note that it may take 15 minutes or more for the IBM Cognos service to start.

Optional configuration for the IBM Cognos Analytics 11.0 course environment

The environment provided has been configured and tested to work with the demonstrations and exercises in this course. For instructors teaching in an ILT or ILO environment, or for students in a Self-paced Virtual Classroom (SPVC) environment, there is an optional configuration that can be performed. This configuration provides access to a wide range of templates that can be used to create reports in the new IBM Cognos Analytics - Reporting environment. For the course environment, this change impacts the IBM Cognos Analytics - Reporting user interface and the initial steps used to create reports.

When creating a report in the current environment (as per the course set up tasks), the user interface and dialog box for choosing a report type (List, Crosstab, Chart, etc.) appears as follows:



The dialog box is a legacy component from the IBM Cognos BI 10.2.2 version of the product. It is available, by default, when performing the "Custom" installation of IBM Cognos Analytics 11.0. The "Custom" installation was chosen for the set up of the course environment. Note: steps in demonstrations and exercises are currently scripted to use this dialog box for creating reports.

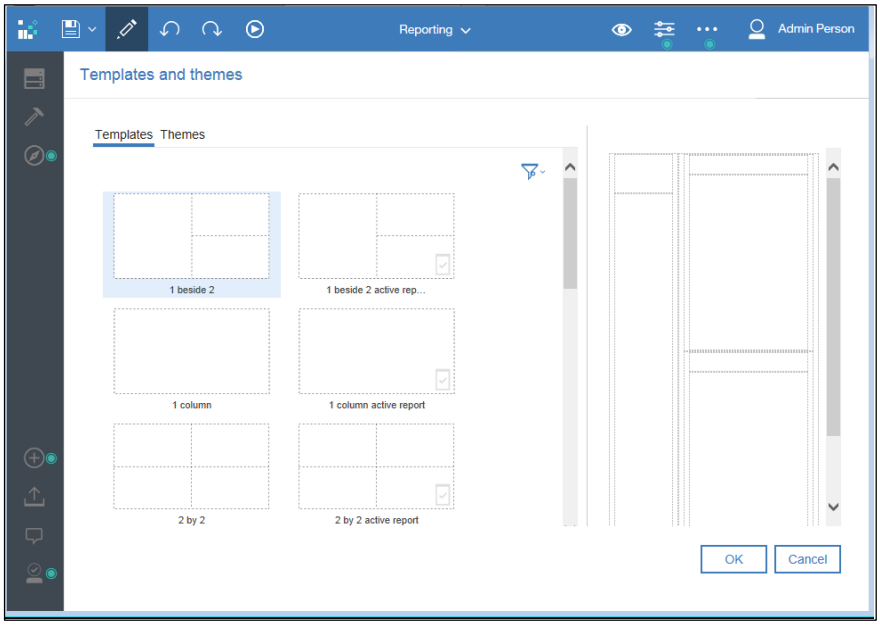
The optional configuration includes the deployment of the Templates.zip deployment archive. This can be performed using the following steps in the course environment:

1. In **Internet Explorer**, log on to **IBM Cognos Analytics** as **admin\Education1**.
2. On the side panel (left pane), click **Manage**, and then click **Administration console**.
3. Click the **Configuration** tab, and then click **Content Administration**.
4. On the toolbar, click **New Import**.
5. On the **Select a deployment archive** page, select the **Templates** archive, and then click **Next**.
6. On the Specify name and description page, click **Next**.
7. On the Public folders, directory and library content page, select the Templates check box, and then click **Next**.
8. On the Specify the general options page, click **Next**.
9. On the Review the summary page, click **Next**.
10. On the **Select an action** page, ensure **Save and run once** is selected, and then click **Finish**.
11. On the **Run with options** page, ensure **Now** is selected, click **Run**, and then click **OK**.
12. Close the **IBM Cognos Administration** tab.

Once this configuration has been performed, use the following steps to create a report.

1. Follow the steps from the demonstration or exercise to log on as a user with the ability to create reports.
2. From the side panel (left pane), click **New**.

3. Click **Report**.
   The user interface now appears as follows:



   You are presented with a list of templates to choose from.
4. Select a template. Note: there are various layouts to choose from. In most instances, you will be successful with current steps using the "1 column" option (1st column, 2nd row). If issues are encountered, the "Blank" option is also available.
5. In the report layout on the right, click +, and then choose the report type (List, Crosstab, Chart), or object (Table, Text item, Block)
6. Continue creating the report as scripted in the demonstration or exercise. Note: some screen captures may appear different than what is presented in the demonstration or exercise.

Document conventions

Conventions used in this guide follow Microsoft Windows application standards, where applicable. As well, the following conventions are observed:

- **Bold**: Bold style is used in demonstration and exercise step-by-step solutions to indicate a user interface element that is actively selected or text that must be typed by the participant.
- *Italic*: Used to reference book titles.
- CAPITALIZATION: All file names, table names, column names, and folder names appear in this guide exactly as they appear in the application.
  To keep capitalization consistent with this guide, type text exactly as shown.

Exercises

Exercise format

Exercises are designed to allow you to work according to your own pace. Content contained in an exercise is not fully scripted out to provide an additional challenge. Refer back to demonstrations if you need assistance with a particular task. The exercises are structured as follows:

The business question section

This section presents a business-type question followed by a series of tasks. These tasks provide additional information to help guide you through the exercise. Within each task, there may be numbered questions relating to the task. Complete the tasks by using the skills you learned in the unit. If you need more assistance, you can refer to the Task and Results section for more detailed instruction.

The task and results section

This section provides a task based set of instructions that presents the question as a series of numbered tasks to be accomplished. The information in the tasks expands on the business case, providing more details on how to accomplish a task. Screen captures are also provided at the end of some tasks and at the end of the exercise to show the expected results.

Additional training resources

- Visit IBM Analytics Product Training and Certification on the IBM website for details on:
  - Instructor-led training in a classroom or online
  - Self-paced training that fits your needs and schedule
  - Comprehensive curricula and training paths that help you identify the courses that are right for you
  - IBM Analytics Certification program
  - Other resources that will enhance your success with IBM Analytics Software
- For the URL relevant to your training requirements outlined above, bookmark:
  - Information Management portfolio: http://www-01.ibm.com/software/data/education/
  - Predictive and BI/Performance Management/Risk portfolio: http://www-01.ibm.com/software/analytics/training-and-certification/

IBM product help

Help type	When to use	Location
Task-oriented	You are working in the product and you need specific task-oriented help.	*IBM Product* - Help link
Books for Printing (.pdf)	You want to use search engines to find information. You can then print out selected pages, a section, or the whole book. Use Step-by-Step online books (.pdf) if you want to know how to complete a task but prefer to read about it in a book. The Step-by-Step online books contain the same information as the online help, but the method of presentation is different.	Start/Programs/*IBM Product*/Documentation
IBM on the Web	You want to access any of the following:	
	• IBM - Training and Certification	• http://www-01.ibm.com/software/analytics/training-and-certification/
	• Online support	• http://www-947.ibm.com/support/entry/portal/Overview/Software
	• IBM Web site	• http://www.ibm.com

IBM Training

Create query models

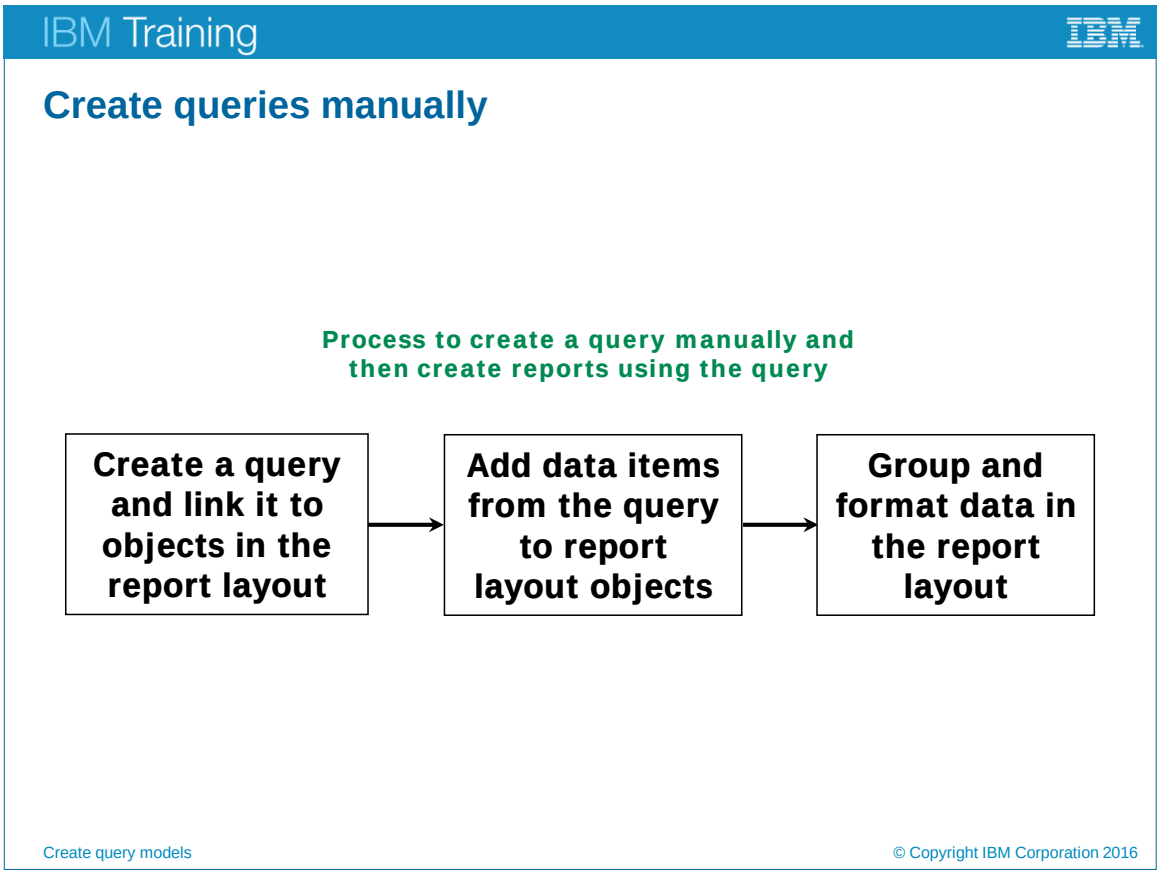
IBM Cognos Analytics (v11.0)

IBM Training

Unit objectives

- Build query models and then connect them to the report layout
- Edit an SQL statement to author custom queries
- Add filters and prompts to a report using the query model

Unit objectives



Create queries manually

When building queries manually, you can:

- add data items to specify the data a query extracts from the data source
- specify data item properties
- add filters and slicers to exclude unnecessary data from the query and specify filter and slicer properties
- specify how a query will retrieve data by customizing its SQL or MDX statement
- combine results from different queries

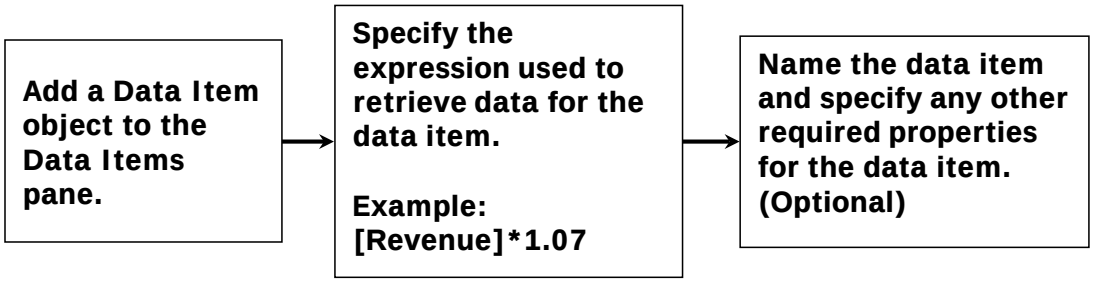
It is a matter of personal preference whether the report author works in Page Explorer to create a report, or if they work in Query Explorer by first creating the query and then later applying the data items and calculations to the report layout.

IBM Training

Add calculated data items to queries

- If the package you are using does not contain the data you require, you can create a calculated data item that will retrieve the required data.

Process to Add a Calculated Data Item to a Query



Create query models

Add calculated data items to queries

When you add a calculated data item using Query Explorer, the item is added directly to the query and you can then choose whether you want to add it to the report layout.

If you are working with many calculated data items, giving the items meaningful names helps you keep track of which data items contain which data.

It is useful to give a meaningful name the calculated data item because if the data item appears in the report layout, its name is used as the title (for example, the column or row title) where the data item appears.

Demonstration and exercise start point information

Before you begin the steps of a demonstration or exercise, you will see information to help you set your environment for the tasks that you will perform. The format appears as follows:

Portal: **http://vclassbase:9300/bi**
User/Password: **brettonf/Education1**
Package: **Team content\Samples\Models\GO data warehouse (query)**
Report Type: **List**
Folder: **Sales and Marketing (query)**
Namespace: **Sales (query)**

This information provides you with a unique starting point for that demonstration or exercise. It tells how you will access the IBM Cognos Analytics portal through a browser, the user ID and password, the package to use, the report type to start with, and within the data source, the folder and namespace for items to be used in your report as you build it. Use this to set your environment before beginning the first task.

If your demonstration provided the start point information above, you would do the following eleven steps before beginning the demonstration tasks (if you perform these steps, you will be ready to start Demonstration 1 of this module, as it uses the same start point provided in this example):

1. From the **Windows** taskbar, click the **Services** icon  , click the **IBM Cognos** service, and then click **Start**.

   It may take 15 or more minutes to start.
2. Once the **IBM Cognos** service has started, close the **Services** window.
3. From the **Start** menu, click **All Programs\Internet Explorer** to launch a browser session.
4. In the **Address** box, type the portal address **http://vclassbase:9300/bi**, and then press **Enter**.

   The Sign in for the IBM Cognos Analytics portal window appears in a browser tab, with a Sign in dialog box prompting for a User ID and Password. You will sign in with the credentials listed in the start point information.

   This section describes in detail how to use the start point information included with the demonstrations and exercises in this course. It is particularly important for students in a self-paced learning environment to review this information before proceeding with the course.

5. In the **User ID** box, type **brettonf**, in the **Password** box, type **Education1**, and then click **OK**.

   The IBM Cognos Analytics portal page displays the applications that your user has permissions for. For example, New  allows the user to launch a new Dashboard, Report, or Data module.

6. Click **New**, and then click **Report**.

   The New dialog box appears. This dialog box allows you to choose a data package and a report template. In the start point information example, you are going to work with the Team content\Samples\Models\GO data warehouse (query) package.

7. In the **New** dialog window, under **Package**, click the ellipsis, navigate to **Team content\Samples\Models\GO data warehouse (query)**, and then click **Open**.

8. From the available report templates, click **Blank**, and then click **OK**.

9. From the **Side panel**, click **Data**  .

10. From the **Source** tab, expand the **Sales and Marketing (query)** folder.

    The expanded folder displays the namespaces that are available to you in this package. The starting point information in this example will work with the Sales (query) namespace.

11. Expand the **Sales (query)** namespace.

    You have completed the start point preparation for the demonstration example provided, and you are now ready to begin the tasks. You will work with the metadata of query subjects, query items, and facts within this namespace unless otherwise mentioned.

    Follow the start point information carefully, as there will be different logins, packages, report types, and namespaces used for each demonstration and exercise in this course.

    If you have performed the steps for the example start point here, then you are ready to begin the tasks in Demonstration 1, as the start point for Demonstration 1 was used in this example.

The environment provided in this course requires the following services to be started before you begin performing demonstrations and exercises:

- Apache Directory Server
- DB2 -DB2COPY 1 - DB2
- DB2DAS - DB2DAS00
- IBM Cognos
- Lotus Domino Server (CProgramFilesx86IBMLotusDominodata)
- World Wide Web Publishing Service

To review the services, on the Taskbar of your environment, click the Services icon, and ensure that the above services are running. If you have closed your image and launched it again, it is a best practice to review the status of the services before continuing with your demonstrations and exercises.

If the Apache Directory Server or DB2 -DB2COPY 1 - DB2 service have stopped, you will need to stop the IBM Cognos service, start the stopped service(s), and then start the IBM Cognos service once the previously stopped service(s) has (have) started successfully. You can start and stop a specific service by double-clicking the service to open the Properties dialog box, and then clicking the Stop or Start buttons.

Note that it may take 15 minutes or more for the IBM Cognos service to start.

IBM Training

Demonstration 1

Build a query and connect it to a report

Product line	Product type	Revenue	Projected Revenue
Camping Equipment	Cooking Gear	272,835,984.18	291,934,503.0726
	Lanterns	126,925,660.64	135,810,456.8848
	Packs	351,880,402.84	376,512,031.0388
	Sleeping Bags	309,172,888.35	330,814,990.5345
	Tents	528,221,728.02	565,197,248.9814
Camping Equipment - Total		1,589,036,664.03	1,700,269,230.5121

Create query models

Demonstration 1: Build a query and connect it to a report

Demonstration 1: Build a query and connect it to a report

Purpose:
Management wants to compare revenue generated by each product line. They also want to examine information about how much revenue each product line will generate if revenue increases by 7%. You will create a query that retrieves this data and then add data from the query to the report layout.

In the interest of time, most demonstrations and exercises in this course will not specifically instruct you to save the report. You are able to save your reports from demonstrations and exercises if you would like. A good place to save reports would be in My content, or in the Team content\B6059 folder of the IBM Cognos Analytics portal.

Portal:	**http://vclassbase:9300/bi**
User/Password:	**brettonf/Education1**
Package:	**Team content\Samples\Models\GO data warehouse (query)**
Report Type:	**Blank**
Folder:	**Sales and Marketing (query)**
Namespace:	**Sales (query)**

Task 1. Clear the Automatic group and summary behavior for lists.

1. On the **Application** bar, click **More**  , and then click **Options**.
2. Click the **Report** tab, and then clear the **Automatic group and summary behavior for lists** check box.
3. Click **OK**.

Task 2. Create a query.

By starting with a blank report, IBM Cognos Analytics - Reporting has not created a query for this report. You will begin by adding a query to this report, and then add data to the query.

1. On the **Side panel**, click **Navigate**  , click the **Query explorer** tab  , and then click **Queries**.
2. On the **Side panel**, click **Toolbox**  , and then drag a **Query** object to the work area.
3. In the work area, double-click **Query1**.
4. On the **Side panel**, click **Data**  .

5. Expand the **Products** query subject.
6. Click **Product line**, Ctrl+click **Product type**, and then drag both the **Data Items** pane.
7. On the **Source** tab, expand **Sales fact**, and then drag **Revenue** to the **Data Items** pane.

   To display projected revenue generated if revenue increases by 7%, you will add a calculated data item to the query that retrieves Revenue multiplied by 1.07.
8. Click **Toolbox**, and then drag a **Data Item** object to the **Data Items** pane.
9. In the **Name** box, replace **Data Item1** with **Projected Revenue**.

   You want to use the Revenue data item that is already in the query, rather than the Revenue item from the source, so you will add Revenue to the expression from the Data Items tab.
10. In the **Data item expression** dialog box, create and validate the following expression:

    [Revenue]*1.07
11. Click **OK** to close the **Data item expression** dialog box.

    You have added the required data items to the query. Before linking this query to the report layout, you want to view the data this query will retrieve.
12. Click **Navigate** on the **Side panel**, right-click **Query1**, and then click **View Tabular Data**.

    You can view the data that the items in the query retrieve.

    A section of the results appear as follows:

Product line	Product type	Revenue	Projected Revenue
Personal Accessories	Binoculars	130,834,653.2	139,993,078.924
Mountaineering Equipment	Climbing Accessories	81,096,582.48	86,773,343.2536
Camping Equipment	Cooking Gear	272,835,984.18	291,934,503.0726
Personal Accessories	Eyewear	867,125,198.48	927,823,962.3736
Outdoor Protection	First Aid	12,429,699.12	13,299,778.0584
Golf Equipment	Golf Accessories	51,514,343.88	55,120,347.9516

13. Close the rendered report tab.

Task 3. Name this query, and then link this query to a list report.

1. In **Query explorer**, click the **Queries** folder, and then in the work area, click **Query1**.
2. From the **Application** bar, click **Show properties**  .

3. In the **Properties** pane, under the **MISCELLANEOUS** section, change the **Name** property from **Query1** to **Projected Revenue for Product Line and Type**.
4. On the **Navigate** tab, click **Page explorer**, and then click **Page1**.
5. From the center of the work area, click **Add**  , click **List**, and then click **OK**.
6. On the **Navigate** tab, click the **Query explorer** tab.

   Notice that the report now contains two queries: **Projected Revenue for Product Line and Type** and **Query1**.

   When you added the list object to this report, IBM Cognos Analytics - Reporting automatically created a second query for this object.

   You will proceed in Page explorer.
7. On the **Navigate** tab, click **Page explorer**  , and then click **Page1**.
8. On the **Side panel**, click **Data**, and then click the **Data items** tab.
9. Under **Projected Revenue for Product Line and Type**, drag **Product line** to the list data object.

   An error message appears explaining that IBM Cognos Analytics - Reporting cannot insert this data item into the list report because the query context "Query1" is not the same as the data items query "Projected Revenue for Product Line and Type".

   Before you can add items from the first query to the list report, you must link the first query to the list object.
10. Click **OK** to close the message box.
11. Click the list data container's **Select List1** button  to select the entire list data container.

    In the Properties pane, under the DATA section, notice that Query1 appears in the Query property. This means that the list is linked to Query1. When you link this list to the first query which you renamed, IBM Cognos Analytics - Reporting will then delete Query1 from the report.
12. In the **Properties** pane, in the **DATA** section, click the **Query** property, and then from the list, select **Projected Revenue for Product Line and Type**.
13. On the **Navigate** tab, click **Query explorer**.

    Observe that the report now contains only the **Projected Revenue for Product Line and Type** query.

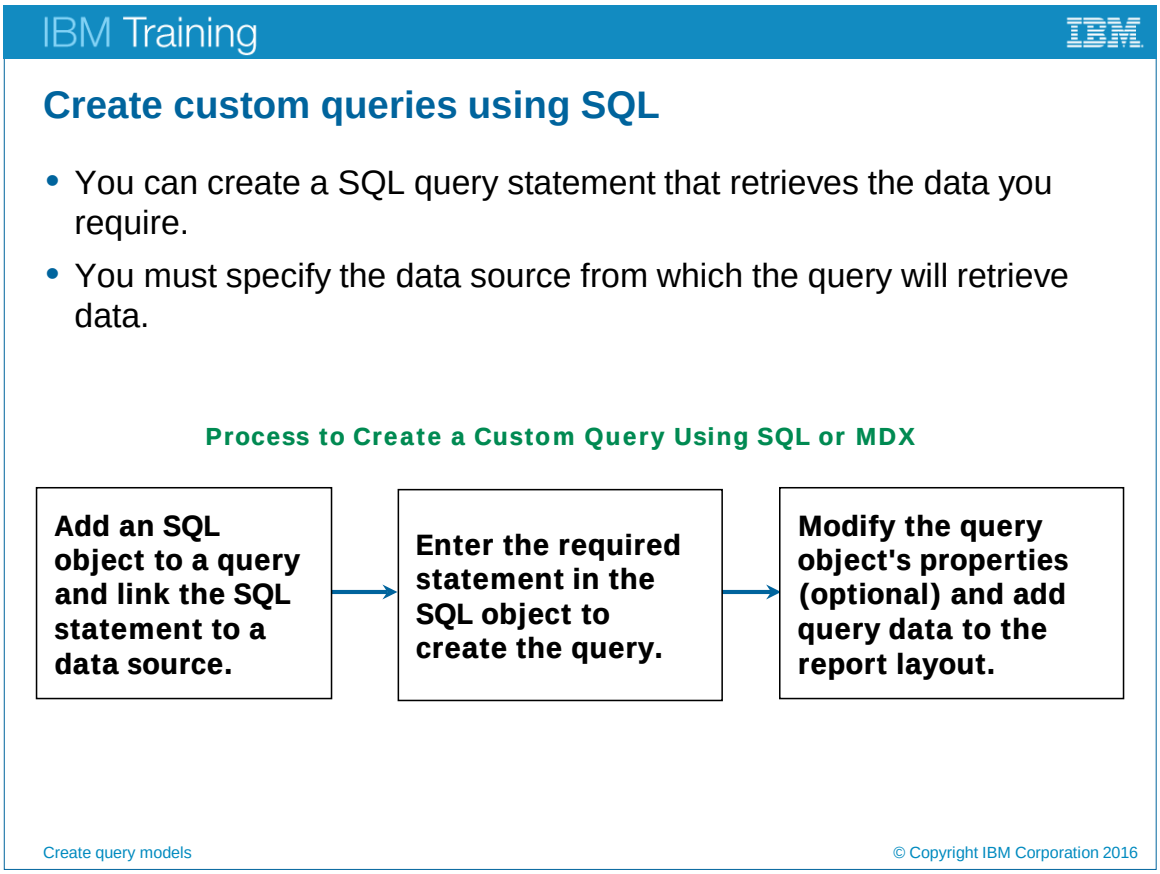
Task 4. Add data to the list report.

1. On the **Data items** tab, click the first item to select it, Shift+click the last item to select all the items in **Projected Revenue for Product Line and Type**, and then drag them all to the list report.
2. In the list data container, click the **<Product line>** list column body, and then from the On-demand toolbar, click **Group / Ungroup**  .
   The list is grouped by Product line.
3. Click the **<Revenue>** list column body, from the On-demand toolbar, click **Summarize**  , and then click **Total**.
4. Click the **<Projected Revenue>** list column body, from the On-demand toolbar, click **Summarize**, and then click **Total**.
   Summary footers are included in the list for the selected columns.
5. On the **Application** bar, click **Run options**  , and then click **Run HTML**.
   A section of the result appears as follows:

Product line	Product type	Revenue	Projected Revenue
Camping Equipment	Cooking Gear	272,835,984.18	291,934,503.0726
	Lanterns	126,925,660.64	135,810,456.8848
	Packs	351,880,402.84	376,512,031.0388
	Sleeping Bags	309,172,888.35	330,814,990.5345
	Tents	528,221,728.02	565,197,248.9814
Camping Equipment - Total		1,589,036,664.03	1,700,269,230.5121

   IBM Cognos Analytics - Reporting retrieves the data you specified in the query model for the Projected Revenue for Product Line and Type query, and displays it in the list report.
6. Close the rendered report tab.
7. On the **Application** bar, to the right of the **New*** report title, click the down arrow to open the currently active report list, and then click **Remove** 
   (to the right of **New***) to close the report.
8. Click **OK** to continue without saving.
9. Leave the **IBM Cognos Analytics** portal open for the next demonstration.

Results:
You built a query containing Product line, Product type, and Revenue items. You added a calculated data item to determine values for a 7% increase in revenue and then linked this query to a list report. You grouped the data in the list report, added aggregate data, and then ran the report.



Create custom queries using SQL

To edit the SQL code used to create a query, first convert the query to SQL, and then edit the query statement.

You can only add new data items to the query by editing the SQL statement to retrieve the item.

If you convert a query to SQL and then modify the query statement, you cannot convert the modified statement back to an IBM Cognos Analytics - Reporting query object.

IBM Training

Demonstration 2

Create a report using a SQL statement

Year: 2010

Product Type	Product Name	Average Forecasted Cost
Binoculars	Opera Vision	51.08
	Ranger Vision	83.09
	Seeker 35	79.72
	Seeker 50	103.37
	Seeker Extreme	81.91
	Seeker Mini	45.46
Binoculars - Average		**74.105**

Demonstration 2: Create a report using a SQL statement

Demonstration 2: Create a report using a SQL statement

Purpose:
You have been asked to create a report using an SQL statement that displays the production cost of each product. The report should also display the average production costs for all products in each product type.

You will convert a query to SQL, and then you will use the SQL in a report to retrieve the necessary data in the report.

As a report author, by default you will not have the capabilities to execute this report from SQL. The administrator must grant capabilities to execute the specification and to execute user-defined SQL. This capability has been configured by the administrator for authors in your environment.

Portal:	**http://vclassbase:9300/bi**
User/Password:	**brettonf/Education1**
Application:	**IBM Cognos Analytics - Reporting**
Package:	**Team content\Samples\Models\GO data warehouse (query)**
Report Type:	**List**
Folder:	**Sales and Marketing (query)**
Namespace:	**Product forecast (query)**

Task 1. Create a query, and then convert the query to SQL.

1. Open a new **List** report template using the **GO data warehouse (query)** package.
2. From the **Data\Source** tab, add the following query items to the list data container from the **Product forecast (query)**:
   - Time: **Year**
   - Products: **Product type**, **Product**
   - Product forecast fact: **Forecast unit cost**

Year	Product type	Product	Forecast unit cost
<Year>	<Product type>	<Product>	<Forecast unit cost>

3. From the **Navigate** tab, right-click **Report**, and then click **Show Generated SQL/MDX**.

   The results are displayed in the Generated SQL dialog box. The default format is Native SQL.
4. Under **Generated SQL/MDX**, change **Native SQL** to **IBM Cognos SQL**.

   Notice the different presentation format of the code. You will work with the Native SQL in this demonstration.
5. In the list, change **IBM Cognos SQL** to **Native SQL**.
6. Click anywhere in the code, press Ctrl+A to select all of the code, and then press Ctrl+C to copy the selection to the clipboard.
7. From the Windows **Start** menu, click **All Programs**, click **Accessories**, and then click **Notepad**.
8. Press Ctrl+V to paste the code into **Notepad**, and then save this file to the **C:\Training\B6059** directory as **Create Query Models SQL.txt**.
9. In **IBM Cognos Analytics - Reporting**, close the **Generated SQL** dialog box.
10. On the **Application** bar, to the right of **New***, click the down arrow, and then click **Remove** to close the report.

    This removes the report from the currently active report list.
11. Click **OK** to continue without saving.

Task 2. Create a list report, add an SQL object, and specify a data source.

1. Open a new **List** report template using the **GO data warehouse (query)** package.
2. On the **Navigate** tab, click the **Query explorer** tab, and then click the **Queries** folder.

   You have the SQL needed to retrieve the data for this report, so you will add an SQL object to Query1 and then paste the SQL statement into the SQL object.
3. From the **Toolbox** tab, drag an **SQL** object to the right of **Query1**.

   The result appears as follows:

   The SQL code that you have saved in Task 1 is native to the data source you will be retrieving data from, so you will leave the SQL Syntax property for the SQL object set to Native.

   You need to specify that you will be retrieving data from the great_outdoors_warehouse database.

4. With the new **SQL** object selected, on the **Application** bar, click **Show properties** to open the **Properties** pane.
5. In the **Properties** pane, double-click the **Data source** property.
6. In the **Data source** dialog box, click **great_outdoors_warehouse**, and then click **OK**.

Task 3. Add an SQL statement, modify it, and then view the data retrieved.

You will copy the SQL statement from Task 1 to the clipboard.

1. Switch to **Notepad**, where the **Create Query Models SQL.txt** code is displayed.
2. From the **Edit** menu, click **Select All**, and then from the **Edit** menu, click **Copy**.
3. In **IBM Cognos Analytics - Reporting**, in the work area, double-click the **SQL** object.
4. Right-click in the work area of the **SQL** dialog box, and then click **Paste**.

   Examine the SQL and observe that near the beginning, this code specifies that the query retrieves four data items aliased as "Year0", "Product_type", "Product", and "Forecast_unit_cost" from the data source.

   You want to modify the way this statement names the data items to make the final report easier to read.

   Adding an SQL object breaks the link to the package. To do this, build a report from a package in IBM Cognos Analytics - Reporting and look at the query object as there will be no SQL object. Package contents are available in the Source tab. From Query properties, open the Generated SQL, click Convert button and add one space at the end of the statement. Although the SQL has not changed, the addition of the space breaks the connection to the package. This also means that you cannot upgraded the report.

   When you add an SQL statement to a report, the report is using custom SQL and it can no longer be supported by IBM Cognos Customer Support.

   Database administrators may be concerned with the performance of standard reports and may provide report authors with the SQL statements to use to create reports if they are concerned about the size or complexity of reports.

   In the Insertable Objects pane, you will not be able to take advantage of any relationships or other modeling data modelers have set up before publishing the package.

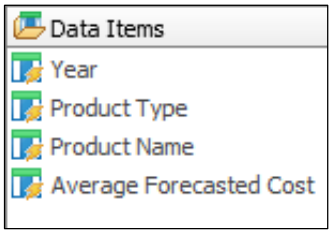
5. In the statement, make the following changes to the alias names (not the data item names):
   - **"Year0"** to **"Year"**
   - **"Product_type"** to **"Product Type"**
   - **"Product"** to **"Product Name"**
   - **"Forecast_unit_cost"** to **"Average Forecasted Cost"**

   A section of the code appears as follows:

```
SELECT
   "GO_TIME_DIM0"."CURRENT_YEAR" AS "Year",
   "Product"."Product_type" AS "Product Type",
   "Product"."Product_name" AS "Product Name",
   AVG(CAST("DIST_PRODUCT_FORECAST_FACT"."UNIT_COST" AS DOUBLE PRECISION)) AS "Average
Forecasted Cost"
```

   These changes specify that the four data items in the query be named Year, Product Type, Product Name, and Average Forecasted Cost.
6. At the end of the code, delete the following text:
   - **FOR FETCH ONLY**
7. In the **SQL** dialog box, click **Validate**, and then once **IBM Cognos Analytics - Reporting** has validated the data, click **OK** to close the **SQL** dialog box.
8. In the work area, double-click **Query1**.

   The four data items retrieved by the data source appear in the Data Items pane for Query1. The data items are named as you specified in the SQL statement.

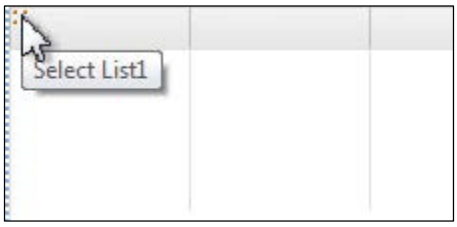


   You will now rename the query.
9. In then **Properties** pane, in the **MISCELLANEOUS** section, change the **Name** property from **Query1** to **Product Type Forecasted Costs**.

   You will now add data from the **Product Type Forecasted Costs** query to the list report.

Task 4. Add data to the report.

1. On the **Navigate** tab, click **Page explorer**, and then click **Page1**.
2. Click the data container's **Select List1** button to select the entire list.



Notice in the Properties pane that this list object is linked to the **Product Type Forecasted Costs** query, which means that you can add data from the query to this report.

3. On the **Data\Source** tab, expand **Sales and Marketing (query)\Product forecast (query)\Products** query subject, and then drag **Product line** query item to the list report.

   A warning message appears. Because the query was sourced from SQL and not from the associated package, you cannot add query items from the Source tab.

4. Click **OK** to close the message, and then click the **Data items** tab.

   The Product Type Forecasted Costs query, and the data items that you added to the query using the SQL statement, appear in the pane.

5. Under **Product Type Forecasted Costs**, click the first data item, Shift+click the last data item to select all of the items, and then drag the selected items to the list.
6. In the list data container, click the **<Year>** list column body, and then from the On-demand toolbar, click **Section / unsection**  .

   You have sectioned the report by year.

7. From the **Application** bar, click **More**, and then click **Locked** to unlock the list data container.
8. In the section header click on **<Year>**, and then from the **Properties** pane, under **DATA**, double-click **Data format**.
9. In the **Format type** list, click **Number** to select it, set the **Use thousands separator** property to **No**, and then click **OK** to close the **Data Format** dialog box.
10. From the **Application** bar, click **More**, and then click **Unlocked** to lock the list data container.

11. In the list, click **<Product Type>**, and then from the On-demand toolbar, click **Group / Ungroup**.

    The list is grouped by Product Type.

Task 5. Modify data item aggregation properties, and then run the report.

You will include data about the average production cost for all products in each product type and the average production cost for all product types.

1. In the list, click the **<Average Forecasted Cost>** list column body, from the On-demand toolbar click **Summarize**, and then click **Average**.

   Notice that only one aggregate row appears at the bottom of the report. You also want the report to include aggregate rows for each product type. To accomplish this, you will need to change the Aggregate Function property for the Product Name data item.

2. From the Application toolbar, click **Undo**  .
3. On the **Navigate** tab, click the **Query explorer** tab, click the **Product Type Forecasted Costs** query, and then in the **Data Items** pane, click **Product Name**.

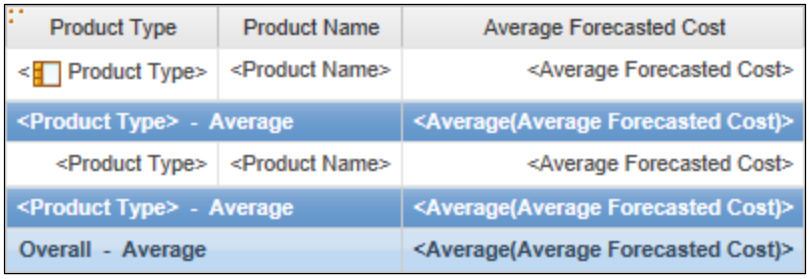
   In the Properties pane, notice that the Detail aggregation property is set to Default. To achieve the results you want in this report, you will set this property to None.

4. In the **Properties** pane, in the **DATA ITEM** section, click the **Detail aggregation** property, and then in the list, click **None**.
5. Click the **Page explorer** tab, and then click **Page1**.

6. In the list, click the **<Average Forecasted Cost>** list column body, from the On-demand toolbar click **Summarize**, and then click **Average**.

   Summary rows appear for each Product Type and a summary row appears at the bottom of the report for all the Product Types.

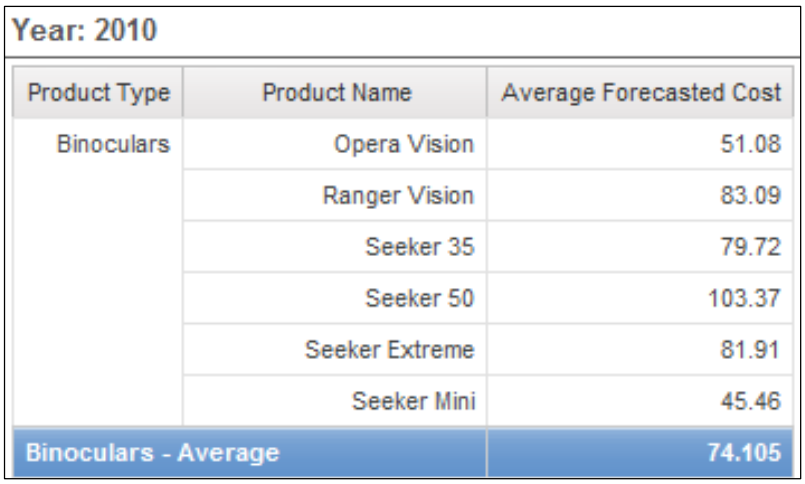
   A section of the result appears as follows:



Product Type	Product Name	Average Forecasted Cost
<Product Type>	<Product Name>	<Average Forecasted Cost>
<Product Type> - Average		<Average(Average Forecasted Cost)>
<Product Type>	<Product Name>	<Average Forecasted Cost>
<Product Type> - Average		<Average(Average Forecasted Cost)>
Overall - Average		<Average(Average Forecasted Cost)>

7. Run the report in **HTML**.

   A section of the result appears as follows:



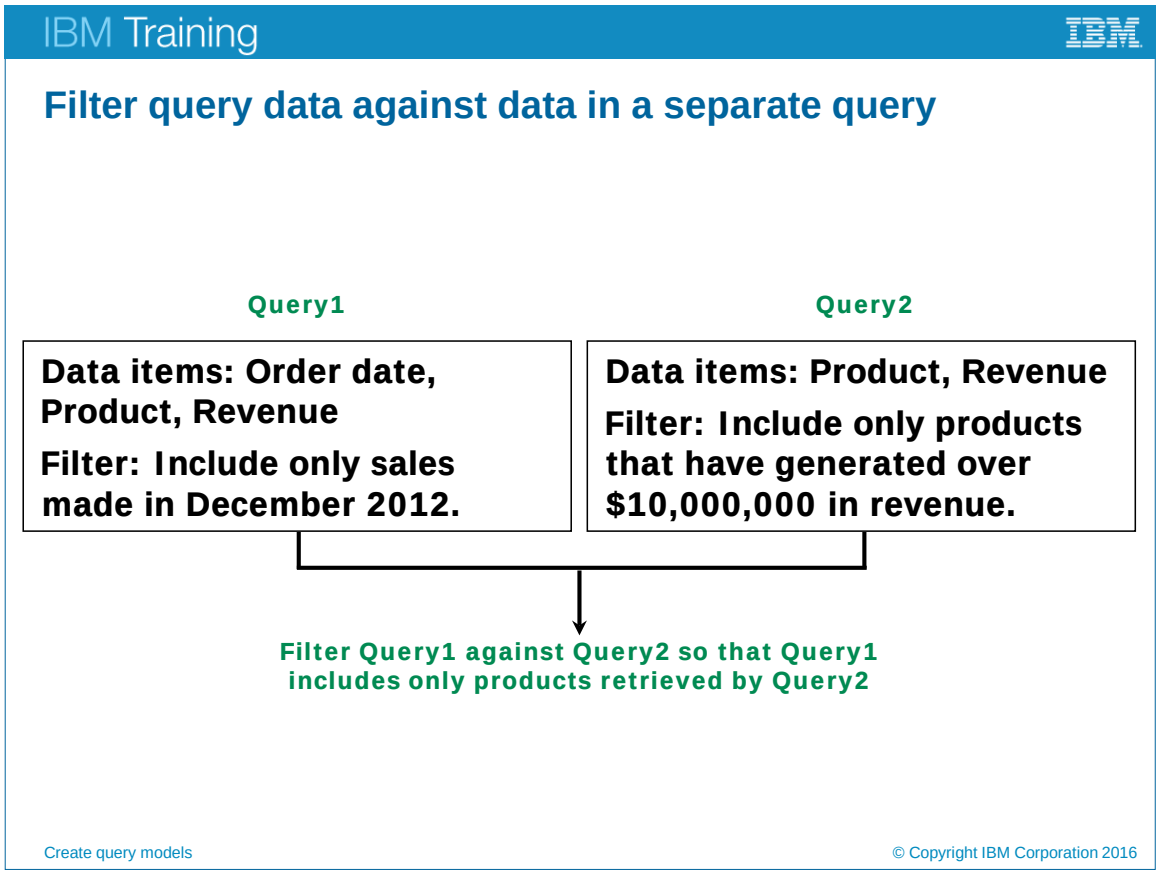
Year: 2010

Product Type	Product Name	Average Forecasted Cost
Binoculars	Opera Vision	51.08
	Ranger Vision	83.09
	Seeker 35	79.72
	Seeker 50	103.37
	Seeker Extreme	81.91
	Seeker Mini	45.46
Binoculars - Average		74.105

   The data items that you added using SQL are displayed, and the grouping and aggregation you specified in the report layout is applied.

8. Close the rendered report tab.
9. Remove the report from the active report list without saving.
10. Leave **IBM Cognos Analytics** open for the next demonstration.

Results:
You used an SQL statement to create a report that displays the production cost of each product. You modified the SQL statement to change the way data items are named and then added query data to the report layout. You sectioned, grouped, and formatted the data using the report layout and added aggregate data to the report to display the average production costs of all products in each product type and the average production cost of all product types.



Filter query data against data in a separate query

When you filter Query1 against data in a second query, each query retrieves a separate set of data.

According to the filter you create, Query1 will only retrieve data retrieved by the second query.

IBM Training

Demonstration 3

Answer a business question by referencing data in a separate query

834905	Zone	755.55	23
834905 - Total		755.55	23
834922	Zone	1,790.1	54
834922 - Total		1,790.1	54
Overall - Total		9,451,826.18	165,443

Create query models

Demonstration 3: Answer a business question by referencing data in a separate query

Demonstration 3: Answer a business question by referencing data in a separate query

Purpose:
For any given month, management wants to know which products were ordered that have lifetime revenues exceeding a specified amount. You will create a report to answer this question.

For example, management could use the report to view data for products ordered during December 2012 that have generated lifetime revenue of at least $150 million.

To create this report, you will create one query containing prompts for the order month and year, and create a second query containing a prompt for lifetime revenue. You will then filter the first query against the data retrieved by the second query.

Portal:	**http://vclassbase:9300/bi**
User/Password:	**brettonf/Education1**
Package:	**Team content\Samples\Models\GO data warehouse (query)**
Report Type:	**List**
Folder:	**Sales and Marketing (query)**
Namespace:	**Sales (query)**

Task 1. Create a list report containing data about orders and add prompts.

1. Open a new **List** report template, using the **GO data warehouse (query)** package.
2. From the **Data\Source** tab, add the following query items to the list data container:
   - Sales order: **Order number**
   - Time: **Year**, **Month (numeric)**
   - Products: **Product**
   - Sales fact: **Revenue**, **Quantity**

Order number	Year	Month (numeric)	Product	Revenue	Quantity
<Order number>	<Year>	<Month (numeric)>	<Product>	<Revenue>	<Quantity>
<Order number>	<Year>	<Month (numeric)>	<Product>	<Revenue>	<Quantity>
<Order number>	<Year>	<Month (numeric)>	<Product>	<Revenue>	<Quantity>

You will create a prompt for users to select a year to view data for.

3. From the On-demand toolbar, click **Filters**, and then click **Edit Filters**.
4. In the **Filters** dialog box, ensure that the **Detail Filters** tab is selected, and then click **Add**.

   You want to create a simple filter condition based on only one data item, so you could use the option of Custom based on data item, and select Year. However, the prompt that is displayed from this option offers the user the opportunity to select more than one year, so you will create an advanced filter to control this.
5. Select **Advanced**, and then click **OK**.
6. Create and validate the following expression:

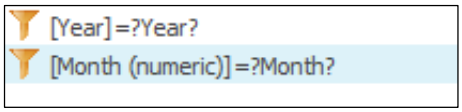
   [Year]=?Year?

   Hint: validate using 2012 when prompted.
7. Click **OK** to close the validation window, and then click **OK** again to close the **Detail filter expression** dialog box.
8. In the **Filters** dialog box, repeat steps 4 through 7, to create another detail filter for **Month (numeric)**:

   [Month (numeric)]=?Month?

   Provide **12** as the value for month number, when validating the expression.

   You now have two filters.



9. Click **OK** to close the **Filters** dialog box.

Task 2. Format and then test the report.

1. In the list data container, click the **<Order number>** list column body, Shift-click the **<Month (numeric)>** list column body, and then click **Group / Ungroup**.
2. In the list data container, click the **<Revenue>** list column body, from the On-demand toolbar click **Summarize**, and then click **Total**.
3. In the list data container, click the **<Quantity>** column body, from the On-demand toolbar click **Summarize**, and then click **Total**.

   A section of the result appears as follows:

Order number	Year	Month (numeric)	Product	Revenue	Quantity
<Order number>	<Year>	<Month (numeric)>	<Product>	<Revenue>	<Quantity>
		<Month (numeric)> - Total		<Total(Revenue)>	<Total(Quantity)>
		<Month (numeric)>	<Product>	<Revenue>	<Quantity>
		<Month (numeric)> - Total		<Total(Revenue)>	<Total(Quantity)>
	<Year> - Total			<Total(Revenue)>	<Total(Quantity)>

   You will test the report by viewing data for December 2012.
4. Run the report in **HTML**.

5. Ensure that the number in the **Month (numeric)** prompt is **12**, and the number in the **Year** prompt is **2012**, and then click **OK**.

   A section of the result appears as follows:

Order number	Year	Month (numeric)	Product	Revenue	Quantity
104579	2012	12	Canyon Mule Weekender Backpack	80,393.6	296
			Firefly 4	7,507.2	255
			Flicker Lantern	7,737.75	225
			Mountain Man Analog	5,891.7	123
			Polar Ice	2,090	19
			Polar Sports	6,296.93	52
			Polar Sun	1,174.96	19
			Star Gazer 2	142,198.1	257
		12 - Total		253,290.24	1,246
	2012 - Total			253,290.24	1,246
104579 - Total				253,290.24	1,246

6. In the report, click **Bottom**, and then scroll to the end of the report.

   You can see the total revenue generated by orders made in December and the total quantity of products sold.

7. Close the rendered report tab.

Task 3. Add a third filter to prompt for a minimum lifetime revenue amount.

You want this report to display only data for orders that have life-to-date revenues of a minimum specified amount, selected using a prompt at run time. You will add a filter to the report to achieve this result.

1. Click anywhere on the report, from the On-demand toolbar click **Filters**, click **Edit Filters**, and then click **Add**.
2. Select **Advanced**, and then click **OK**.
3. Create and validate the following expression:

   [Revenue]>=?Minimum revenue?

   Hint: Drag Revenue from the Data items tab.

   Validate the expression, providing **150000000** (150 million) as the minimum revenue prompt.

4. Click **OK** to close the **Detail filter expression** dialog box.

5. In the **Filters** dialog box, with the Revenue filter still selected, in the **Application** section, click **After auto aggregation**, and then click **OK**.

   You will test this prompt by viewing data for products sold in December 2012 that generated lifetime revenue of at least 150 million dollars.

6. Run the report in **HTML**, ensure that **2012** is the **Year**, **12** is the **Month (numeric)**, **150000000** (150 million) is the **Minimum revenue**, and then click **OK**.

   The report contains no data since all filters combined eliminated all data from the query.

   To retrieve the data you require, you will remove the third filter from Query1 and will create a second query containing this filter. You will then create a query reference from the first query. You will first give Query1 a more meaningful name.

7. Close the rendered report tab.
8. On the **Navigate** tab, click the **Query explorer** tab, and then click **Query1**.
9. Open the **Properties** pane, and then in the **MISCELLANEOUS** section, change the **Name** property to **Selected month**.
10. In the **Detail Filters** pane, click the **[Revenue]>=?Minimum revenue?** filter, and then press **Delete**.

Task 4. Add a second query containing a filter to prompt for a minimum lifetime revenue amount.

1. On the **Query explorer** tab, click **Queries** to view of all queries in the report.
2. On the **Toolbox** tab, drag a **Query** object below the **Selected month** query in the work area, and then double-click the newly added **Query1**.
3. In the **Properties** pane, in the **MISCELLANEOUS** section, change the **Name** property to **Minimum revenue**.

   You will add items to your query.

4. On the **Data\Source** tab, under the **Products** query subject, double-click **Product**, and then under the **Sales fact** query subject, double-click **Revenue** - to add them to the **Data Items** pane.

   You will add a filter to this report to prompt users to select a minimum lifetime revenue for all products retrieved by the query.

5. From the **Data Items** pane, drag **Revenue** into the **Detail Filters** pane.
6. Create and validate the following expression:

   [Revenue]>=?Minimum revenue?

   You want to apply this filter to all revenue generated by a product rather than the revenue generated by individual orders.

7. Click **OK**.
8. With the filter you just created still selected, in the **Properties** pane, under the **GENERAL** section, click the **Application** property, and then from the list click **After Auto Aggregation**.

   You will preview these results to see which products have a life-to-date revenue of at least 150 million.
9. From the **Navigate** tab, right-click the **Minimum revenue** query, click **View Tabular Data**, ensure **150000000** (150 million) appears in the **Minimum revenue** box, and then click **OK**.

   The result appears as follows:

Revenue	Product
168,191,550.48	Star Lite
157,369,344.95	Zone

   You can see that two products, Star Lite and Zone, generated lifetime revenue of at least one hundred and fifty million dollars.
10. Close the rendered report tab.

Task 5. Filter the first query by referencing data in the second query.

You will specify that only products that meet the conditions of the Minimum revenue filter should be included in the Year to date query.

1. On the **Navigate** tab, on the **Query explorer** tab, click the **Selected month** query.

   You will filter the Product name data item in the Selected month query, against the Product name data items retrieved by the Minimum revenue query.
2. From the **Data Items** pane, drag **Product** to the **Detail Filters** pane.
3. At the end of the expression, type **IN (**, and then in the **Available Components** pane, click the **Queries**  tab.
4. From the **Available Components** pane, under **Minimum revenue**, drag **Product** to the end of the expression.
5. At the end of the expression, type **)**.

   The final expression appears as follows:

   [Product] IN ([Minimum revenue].[Product])
6. Validate the expression, and then click **OK** to close the **Detail filter expression** dialog box.
7. Run the report in **HTML**.

8. Ensure that **150000000** (150 million) is the **Minimum revenue**, **2012** is the **Year**, **12** is the **Month (numeric)**, and then click **OK**.

   A section of the result appears as follows:

Order number	Year	Month (numeric)	Product	Revenue	Quantity
104614	2012	12	Star Lite	210,972	600
		12 - Total		210,972	600
	2012 - Total			210,972	600
104614 - Total				210,972	600
104628	2012	12	Star Lite	214,488.2	610

   The report contains data for the orders in December 2012 of the two products that have lifetime revenue of at least 150 million dollars (Star Lite and Zone).

9. Close the rendered report tab.

Task 6. Format the report.

You do not need to include the Year and Month columns in the list report, as the user is filtering on a specific year and month.

1. On the **Navigate** tab, click the **Page explorer** tab, and then click **Page1**.
2. Click the **<Year>** list column body, Ctrl+click the **<Month (numeric)>** column body, and then from the On-demand toolbar, under **More**, click **Cut**.

   You have cut the data items from the report layout, but they are still in the query, as you did not delete them.

3. In the list, delete the **<Month (numeric)>** and **<Year>** summary rows, as they are no longer needed.

   The results appear as follows:

Order number	Product	Revenue	Quantity
<Order number>	<Product>	<Revenue>	<Quantity>
<Order number> - Total		<Total(Revenue)>	<Total(Quantity)>
<Order number>	<Product>	<Revenue>	<Quantity>
<Order number> - Total		<Total(Revenue)>	<Total(Quantity)>
Overall - Total		<Total(Revenue)>	<Total(Quantity)>

4. Run the report in **HTML**, with **150000000** as the **Minimum revenue**, **2012** as the **Order year**, and **12** as the **Month (numeric)**.

5. In rendered report tab, click **Bottom**, and then scroll down.

   A section of the result appears as follows:

834905	Zone	755.55	23
834905 - Total		755.55	23
834922	Zone	1,790.1	54
834922 - Total		1,790.1	54
Overall - Total		9,451,826.18	165,443

   You can see that for December 2012, 165,443 units of these two products were sold and they generated 9,451,826.18 in revenue.

6. Close the rendered report tab.
7. Remove the current report without saving.

   Database administrators may be concerned with the performance of standard reports and may provide report authors with the SQL statements to use to create reports if they are concerned about the size or complexity of reports.

   In the package, you will not be able to take advantage of any relationships or other modeling data modelers have set up before publishing the package.

8. Leave **IBM Cognos Analytics - Reporting** open for the Exercise.

Results:
By filtering one query against another, you created a report that lets users choose to view order data for a specified month for products whose lifetime revenue exceeds a specified amount.

IBM Training

Unit summary

- Build query models and then connect them to the report layout
- Edit an SQL statement to author custom queries
- Add filters and prompts to a report using the query model

Unit summary

IBM Training

Exercise 1

Filter data using the query model

Retailer name	Product	Revenue
Alles für Draußen	Hibernator Lite	24,641.88
	Star Dome	26,309.12
	TrailChef Kettle	10,679.06
Alles für Draußen - Total		**61,630.06**
Baxters Pro Shop	BugShield Extreme	37,065.14
	Glacier GPS	21,836.27
	Seeker 50	14,244.78
Baxters Pro Shop - Total		**73,146.19**
Boot Camp Equipment	Capri	48,989.6

Create query models

Exercise 1: Filter data using the query model

Exercise 1: Filter data using the query model

As Frank Bretton, a report author, you have been asked to create a report that displays revenue data for 2011 for orders by retailer. The report must have the ability to show the retailers that generated revenues less than $100000 for orders that generated revenues more than $10000.

To accomplish this:

- Create a list report using the GO data warehouse (query) package, Sales and Marketing (query) folder, Sales (query) namespace, with the following query items:
  - Retailers: Retailer name
  - Products: Product
  - Sales fact: Revenue
- Group Retailer name and display Revenue totals.
- Add a detail filter for individual orders that generated more than $10,000 in Revenue.
- Add a summary filter to include retailers who generated less than $100,000 in total revenue.
- Add a prompt to let users view data from a particular year.

For more information about where to work and the Exercise results, refer to the Tasks and results section that follows. If you need more information to complete a task, refer to earlier demonstrations for detailed steps.

Exercise 1: Tasks and results

Task 1. Create a list report and add filters to the query.

- **Side panel**: Open a new **List** report template using the **GO data warehouse (query)** package.
- **Source tab**: Navigate to **Sales and Marketing (query)/Sales (query)/Retailers**, and then add **Retailer name** to the list report object.
- **Source tab**: Navigate to **Sales and Marketing (query)/Sales (query)/Products**, and the add **Product** to the list report object.
- **Source tab**: Navigate to **Sales and Marketing (query)/Sales (query)/Sales fact**, and then add **Revenue** to the list report object.

  The results appear as follows:

Retailer name	Product	Revenue
<Retailer name>	<Product>	<Revenue>

- **On-demand toolbar**:
  - Group the **<Retailer name>** list column body.
  - Summarize the **<Revenue>** list column body by **Total**.
- **Navigate** tab: Click the **Query explorer** tab, and then click **Query1**.
- **Data Items pane**:
  - Create and validate a detail filter expression that will filter on revenue greater than 10,000.
  - Create and validate a summary filter expression that will filter revenue totals less than 100,000.
- **Properties pane**: Set the scope property for the summary filter to **Retailer name**.

- **Application** bar: Run the report in **HTML**.

  A section of the report appears as follows:

Retailer name	Product	Revenue
Allt för camping AB	Dante	10,036
	Legend	10,209
	Lux	10,955.6
	Maximus	21,420
	Ranger Vision	10,074
	Retro	22,178.1
	Zone	10,216.35
Allt för camping AB - Total		**95,089.05**

- Close the rendered report tab.

Task 2. Add a prompt that references a data item.

- **Source** tab: Create and validate a detail parameterized filter expression on **Year**. Validate using the year **2011** if prompted.
- **Application** bar: Run the report in **HTML**.
- **Year Prompt**: Type **2011**.

  A section of the results appear as follows:

Retailer name	Product	Revenue
Alles für Draußen	Hibernator Lite	24,641.88
	Star Dome	26,309.12
	TrailChef Kettle	10,679.06
Alles für Draußen - Total		**61,630.06**
Baxters Pro Shop	BugShield Extreme	37,065.14
	Glacier GPS	21,836.27
	Seeker 50	14,244.78
Baxters Pro Shop - Total		**73,146.19**
Boot Camp Equipment	Capri	48,989.6
	Trail Master	34,750
Boot Camp Equipment - Total		**83,739.6**

Notice that the report contains different retailer names than when you ran it for all years. This is because in 2011, different retailers meet the filter conditions that you specified for this report.

You have created a report that displays revenue data for 2011 for orders by retailer. The report shows the retailers that generated revenues less than $100,000 for orders that generated revenues more than $10,000.

- Close the rendered report tab.
- Remove the report without saving.
- Sign out of the **IBM Cognos Analytics** portal.
- Close the web browser.

IBM Training

Create reports based on query relationships

IBM Cognos Analytics (v11.0)

IBM Training

Unit objectives

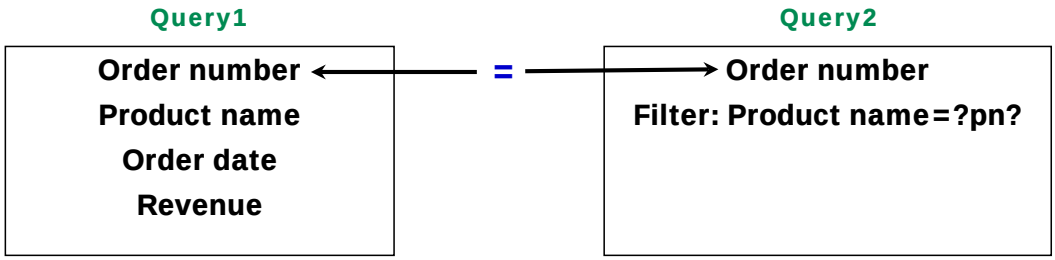
- Create reports by joining queries
- Combine data containers based on relationships from different queries

Unit objectives

IBM Training

Create join relationships between queries (1 of 2)

- Create join relationships between queries when you need to create a relationship between data in separate queries to answer a specific business question.



Create reports based on query relationships

Create join relationships between queries

You can create join relationships when working with relational or dimensionally modeled relational data.

If you want to edit the expression used to create a join relationship, you can convert the join relationship to an expression and then make the required changes.

Join objects let you create both inner joins and outer joins between queries.

An outer join retrieves all rows returned by an inner join (matching rows) as well as all the rows from one table that do not match any row from the other table. Outer joins can be left, right, or full joins.

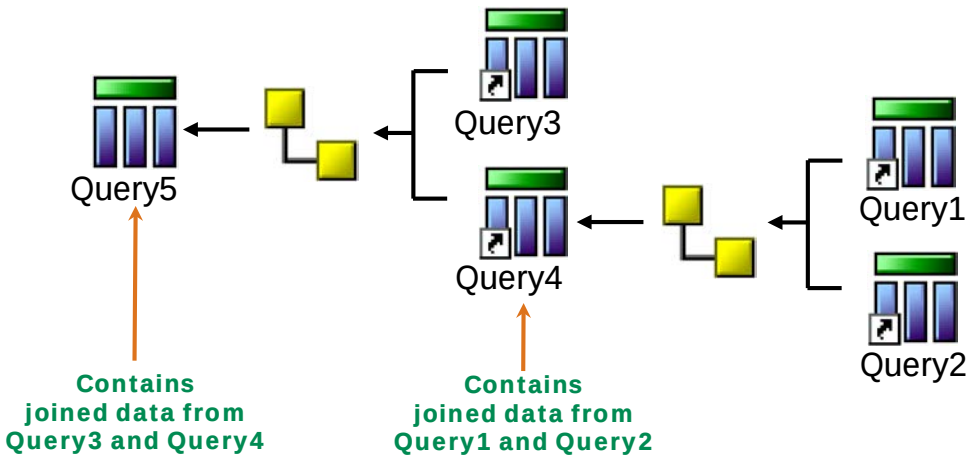
You can create a join relationship if you want to create a list report displaying sales target data for each month, but there is no relationship between the tables in which the Month and Sales Target query items are contained in the model you are using.

After creating the join, you can add both the Month and the Sales target query items to the joined query, and can then link the joined query to a List data container and add the two query items to the list.

IBM Training

Create join relationships between queries (2 of 2)

- To retrieve the data you require, you can join a query or create multiple joined queries.



Create reports based on query relationships

When working with multiple joined queries, it is useful to view the relationships that exist among queries in the report.

IBM Training

Demonstration 1

Analyze product sales by joining two queries

Order number	Date	Product	Quantity	Revenue
100183	Feb 9, 2010	Bear Edge	216	8,402.4
		EverGlow Kerosene	220	6,732
		Star Lite	287	98,791.14
		TrailChef Cook Set	540	28,177.2
		TrailChef Single Flame	477	29,621.7
		TrailChef Utensils	415	7,764.65
100183 - Total			**2,155**	**179,489.09**

Demonstration 1: Analyze product sales by joining two queries

Demonstration 1: Analyze product sales by joining two queries

Purpose:
To help the sales department understand purchasing trends, you have been asked to create a report displaying products that were purchased in orders that included a specific product, selected by users at run time. To achieve this result, you will create a join relationship between two queries.

The environment provided in this course requires the following services to be started before you begin performing demonstrations and exercises:

- Apache Directory Server
- DB2 -DB2COPY 1 - DB2
- DB2DAS - DB2DAS00
- IBM Cognos
- Lotus Domino Server (CProgramFilesx86IBMLotusDominodata)
- World Wide Web Publishing Service

To review the services, on the Taskbar of your environment, click the Services icon, and ensure that the above services are running. If you have closed your image and launched it again, it is a best practice to review the status of the services before continuing with your demonstrations and exercises.

If the Apache Directory Server or DB2 -DB2COPY 1 - DB2 service have stopped, you will need to stop the IBM Cognos service, start the stopped service(s), and then start the IBM Cognos service once the previously stopped service(s) has (have) started successfully. You can start and stop a specific service by double-clicking the service to open the Properties dialog box, and then clicking the Stop or Start buttons.

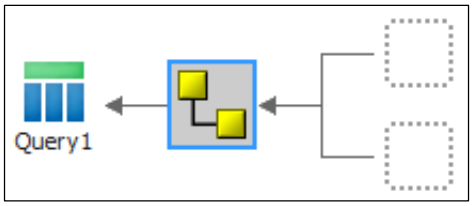
Note that it may take 15 minutes or more for the IBM Cognos service to start.

Portal: **http://vclassbase:9300/bi**
User/Password: **brettonf/Education1**
Package: **Team content\Samples\Models\GO data warehouse (query)**
Report Type: **List**
Folder: **Sales and Marketing (query)**
Namespace: **Sales (query)**

Task 1. Add queries to a list report.

1. Open a new **List** template using the **GO data warehouse (query)** package.
2. On the **Navigate** tab, click the **Query explorer** tab, and then click **Queries**.
3. From the **Toolbox** tab, drag a **Join** object to the right of **Query1**.

   Two drop zones appear to the right of the Join object.



4. From the **Toolbox** tab, drag a **Query** object to each of the two drop zones.

   Query2 and Query3 display in the work area and shortcuts to each query appear in the drop zones to the right of the Join object. You will add data to each of these queries and then join the results.

Task 2. Add data to the queries.

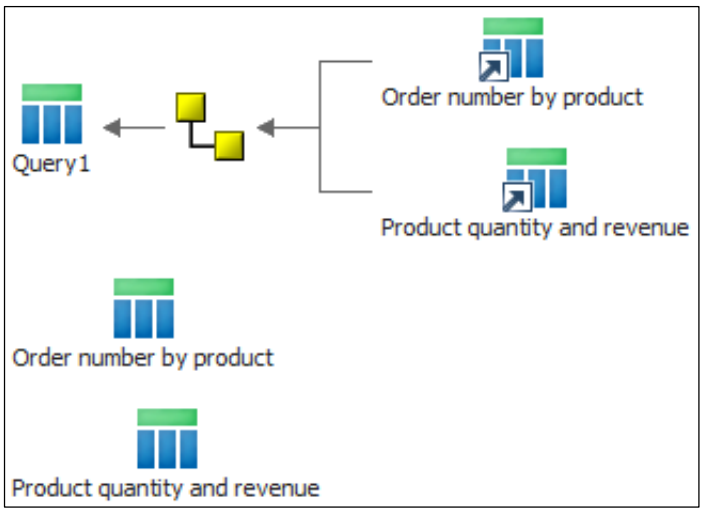
1. In the work area, double-click **Query2** (not the linked shortcut).
2. On the **Data\Source** tab, expand the **Sales and Marketing (query)** folder and the **Sales (query)** namespace.
3. From the **Sales order** query subject, drag **Order number** to the **Data Items** pane.

   You want this query to include only orders that contained a specific product selected at run time.
4. From the **Products** query subject, drag **Product** to the **Detail Filters** pane.
5. Create and validate the following filter expression:

   [Sales (query)].[Products].[Product]= ?Product?
6. Click **OK** to close the **Detail filter expression** dialog box.

7. On the **Application** bar, click **Show properties** to open the **Properties** pane.
8. With the Product detail filter selected, in the **Properties** pane under the **GENERAL** section, change the **Application** property to **After Auto Aggregation**.
9. On the **Navigate** tab, click **Query2**, in the **Properties** pane under the **MISCELLANEOUS** section, rename the query to **Order number by product**.

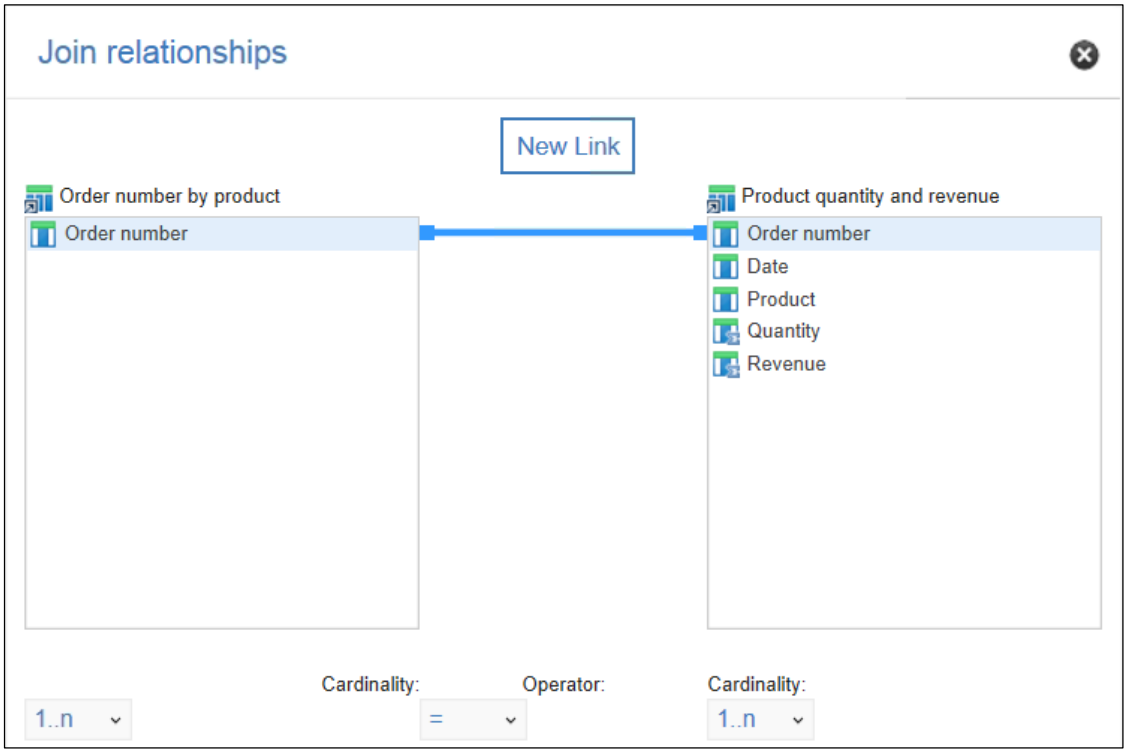
   You will add data to Query3.
10. On the **Navigate** tab, click **Query3**, and then add the following data items from the **Data\Source** tab, to the **Data Items** pane:
    - Sales order: **Order number**
    - Time: **Date**
    - Products: **Product**
    - Sales fact: **Quantity**, **Revenue**
11. On the **Navigate** tab, click **Query3**, in the **Properties** pane, under the **MISCELLANEOUS** section, rename the query as **Product quantity and revenue**.
12. On the **Navigate** tab, click **Queries**.



Task 3. Create a join relationship between the two named queries, and then add data items to Query1.

1. In the work area, double-click the **Join** object.
2. In the **Join relationships** dialog box, click **New Link**.

   You want to specify that the joined query will only contain order numbers that appear in both of the named queries.
3. Ensure that **Order number** is selected in both the **Order number by product** query list and the **Product quantity and revenue** query list, and that the **Cardinality** on the left and the right is **1..n**.



4. Click **OK** to close the **Join relationships** dialog box, and then on the **Query explorer** tab, click **Query1**.
5. From the **Data/Data items** tab, click and drag the **Product quantity and revenue** query to the **Data Items** pane.
6. On the **Navigate** tab, on the **Query explorer** tab, click **Query1**.
7. In the **Properties** pane, in the **MISCELLANEOUS** section, rename the query as **Select product quantity and revenue by order number**.
8. From the **Data/Data Items** tab, in the Data Items pane, click **Product**, and then on the **Properties** pane, change the **Pre-sort** property **to Sort ascending**.

Task 4. Add data to the list, and then run the report.

1. On the **Navigate** tab, click the **Page explorer** tab, and then click **Page1**.
2. On the **Data** tab, click the **Data Items** tab.
3. Under **Select product quantity and revenue by order number**, click the first item, Shift+click the last item to select all of the data items, and then drag them to the list report.

   Do not drag the query to the report data container, as this will create a list inside of the list.
4. In the list data container, click the **<Order number>** list column body, Ctrl+click the **<Date>** list column body, and then from the On-demand toolbar, click **Group / Ungroup**.
5. Click the **<Quantity>** list column body, from the On-demand toolbar, click **Summarize**, and then click **Total**.
6. Click the **<Revenue>** list column body, from the On-demand toolbar, click **Summarize**, and then click **Total**.
7. Run the report in **HTML**.
8. In the **Product** value box, select **Bear Edge**, and then click **OK**.

   A section of the result appears similar to the following:

Order number	Date	Product	Quantity	Revenue
100183	Feb 9, 2010	Bear Edge	216	8,402.4
		EverGlow Kerosene	220	6,732
		Star Lite	287	98,791.14
		TrailChef Cook Set	540	28,177.2
		TrailChef Single Flame	477	29,621.7
		TrailChef Utensils	415	7,764.65
100183 - Total			2,155	179,489.09

9. Click **Page down** and observe that only orders that included the **Bear Edge** product appear in the report.
10. Close the rendered report tab.
11. Remove the report from the active report list, without saving.
12. Leave the **IBM Cognos Analytics** portal open for the next demonstration.

Results:
To help your sales department understand purchasing trends, you created a report displaying products that were purchased in orders that included a specific product, selected by users at run time. You created a join relationship between two queries to ensure that the final report only displayed information for products sold in orders that included the product users selected in the prompt.

IBM Training

Create a report within a report

- You can create a report that delivers information that would otherwise require two or more reports.

Product line	Revenue	Product line Sales Trends
Camping Equipment	$4,578,321	
Mountaineering Equipment	$4,343,692	
Personal Accessories	$3,178,613	

Master list

Detail chart inside a list

Create reports based on query relationships

Create a report within a report

By using master-detail relationships, you ensure that the detail data container only displays information relevant for the row in which it appears in the master data container.

The master and the detail report containers each have their own separate queries. You must create a relationship between the queries based on a data item that is common to both queries.

The common data item used to create the link does not need to appear in both reports but must be included in both of the underlying queries.

IBM Training

Demonstration 2

Create a revenue report using a Master-detail relationship

Country	Planned revenue	Revenue	Revenue by Quarter
Australia	41,383,351.34	38,968,802.62	

Revenue		Q1	Q2	Q3	Q4	2012 Total
Melbourne	Alice Walter	2,019,726.71	2,119,397.51	2,992,620.71	3,374,957.17	**10,506,702.1**
	Dave Smythe	2,186,537.08	1,882,231.16	1,800,068.74	1,047,201.83	**6,916,038.81**
	Donald Ward	2,178,861.5	4,002,379	1,926,622.95	1,293,386.63	**9,401,250.08**
	Jackie Fulford	2,074,084.88	1,559,967.22	2,133,314.29	2,427,298.86	**8,194,665.25**
	Jake Cartel	207,467.54	377,394.22	103,522.89	894,281.45	**1,582,666.1**
	John Sinden		762,028.46	1,208,888.04	396,563.78	**2,367,480.28**

Demonstration 2: Create a revenue report using a Master-detail relationship

Demonstration 2: Create a revenue report using a Master-detail relationship

Purpose:
Management wants a report that shows planned revenue and actual revenue for 2012. They would also like to see the generated revenue broken out by quarters for each sales rep in each city. To do this, you will create a master-detail relationship on Country between a list and a crosstab.

Portal:	**http://vclassbase:9300/bi**
User/Password:	**brettonf/Education1**
Package:	**Team content\Samples\Models\GO data warehouse (query)**
Report Type:	**List**
Folder:	**Sales and Marketing (query)**
Namespace:	**Sales (query)**

Task 1. Create a list report.

1. Open a new **List** template using the **GO data warehouse (query)** package.
2. From the **Data\Source** tab, expand **Sales and Marketing (query)** folder, and then expand **Sales (query)** namespace.
3. Add the following query items to the list data container:
   - Employee by region: **Country**
   - Sales fact: **Planned revenue**, **Revenue**

4. In the list data container, click the **<Country>** list column body, and then from the On-demand toolbar, click **Group / Ungroup**.
5. With **Country** still selected, click **Sort**, and then click **Ascending**.
6. From the On-demand toolbar, click **Filters**, and then click **Edit Filters**.
7. Click **Add**, click **Advanced**, and then click **OK**.

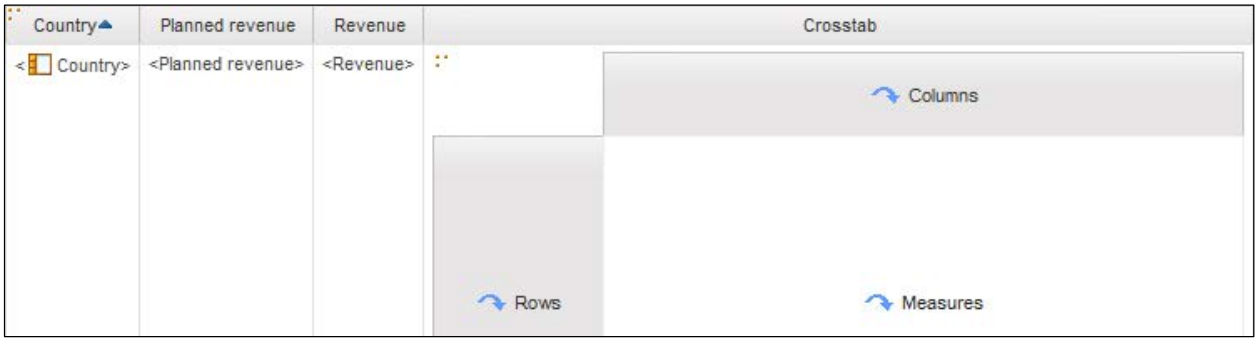
8. In the **Available Components** pane, expand the **Sales and Marketing (query)** folder, the **Sales (query**) namespace, and the **Time** query subject.
9. Drag **Year** to the **Expression Definition** pane.
10. Create and validate the following detail filter expression:

    [Sales (query)].[Time].[Year]=2012
11. Click **OK** to close the **Detail filter expression** dialog box, and then click **OK** to close the **Filters** dialog box.

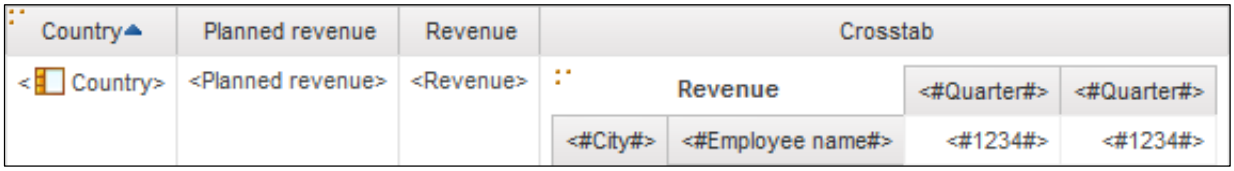
Task 2. Add a crosstab to the list.

1. From the **Toolbox** tab, drag a **Crosstab** object to the last column of the list data container.
2. In the **Query Name** box, replace the **Query2** text with **Employee revenue**, and then click **OK**.

   A section of the result appears as follows:



3. From the **Data\Source** tab, add the following query items to the crosstab:
   - Rows:
     - Employee by region: **City**, **Employee name**(nested as a child of **City**)
   - Columns:
     - Time: **Quarter**
   - Measures:
     - Sales fact: **Revenue**



4. Click the **<#Quarter#>** column header, from the On-demand toolbar click **Sort**, and then click **Ascending**.
5. Click the **<#Employee name#>** row header, from the On-demand toolbar click **Sort**, and then click **Ascending**.
6. On the **Application** bar, click **Show properties** to open the **Properties** pane.
7. Click the **<#Quarter#>** column header, from the On-demand toolbar click **Summarize**, and then click **Total**.
8. Click the **Total** column header, in the **Properties** pane, under the **TEXT SOURCE** section, change the following properties:
   - Source Type: **Text**
   - Text: **2012 Total**
9. From the On-demand toolbar, click **Filters**, and then click **Edit Filters**.
10. Click **Add**, click **Advanced**, and then click **OK**.
11. Create and validate the following detail filter expression:

    [Sales (query)].[Time].[Year]=2012

    Hint: Source tab - Time/Year.

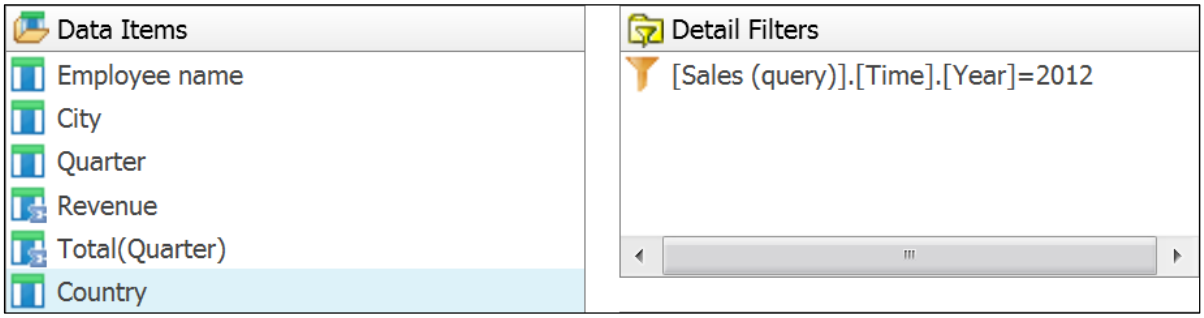
    You have created the same filter as in the list container so that the report only shows data from 2012.
12. Click **OK** to close the **Detail filter expression** dialog box, and then click **OK** to close the **Filters** dialog box.

Task 3. Create a Master-detail relationship, and name the queries.

Although Country is not shown in the crosstab, you need to add it to the query so that a link can be created on Country between both data containers.

1. On the **Navigate** tab, click the **Query explorer** tab, and then click the **Employee revenue** query.
2. From the **Data/Source** tab, under the **Employee by region** query subject, drag **Country** to the **Data Items** pane.

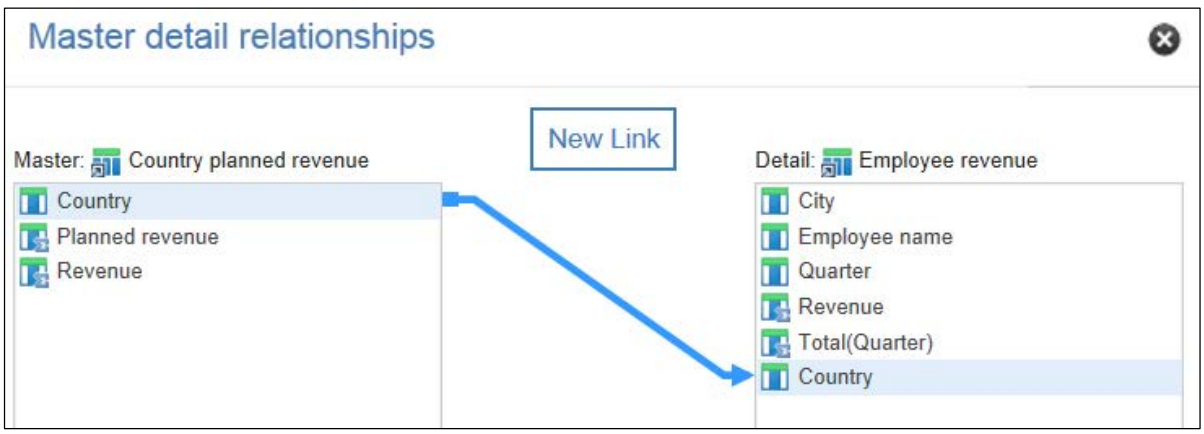
   A section of the result appears as follows:



3. On the **Navigate** tab, on the **Query explorer** tab, click **Query1**, and then in the **Properties** pane under the **MISCELLANEOUS** section, rename the query as **Country planned revenue**.
4. On the **Navigate** tab, click the **Page explorer** tab, and then click **Page1**.
5. Click anywhere in the crosstab, and then from the On-demand toolbar, click **More**, and then click **Master Detail Relationships**.

   The Master Detail Relationships menu option is only active when you have clicked inside the detail report object.
6. Click **New Link**, and then click **Country** in each query list.

   A section of the result appears as follows:



7. Click **OK** to close the **Master detail relationships** dialog box.
8. In the list data container, double-click the **Crosstab** column header, change the text to **Revenue by Quarter**, and then click **OK**.
9. Change the report title to **Revenue by Country in 2012**.
10. Run the report in **HTML**.

    A section of the results appears as follows:

Country	Planned revenue	Revenue	Revenue by Quarter					
Australia	41,383,351.34	38,968,802.62	Revenue		Q1	Q2	Q3	Q4	2012 Total
			Melbourne	Alice Walter	2,019,726.71	2,119,397.51	2,992,620.71	3,374,957.17	**10,506,702.1**
				Dave Smythe	2,186,537.08	1,882,231.16	1,800,068.74	1,047,201.83	**6,916,038.81**
				Donald Ward	2,178,861.5	4,002,379	1,926,622.95	1,293,386.63	**9,401,250.08**
				Jackie Fulford	2,074,084.88	1,559,967.22	2,133,314.29	2,427,298.86	**8,194,665.25**
				Jake Cartel	207,467.54	377,394.22	103,522.89	894,281.45	**1,582,666.1**
				John Sinden		762,028.46	1,208,888.04	396,563.78	**2,367,480.28**

    You can see the details of the revenue generated in each quarter, by each sales rep, for each country.

11. Close the rendered report tab.
12. Remove the report without saving.
13. Leave the **IBM Cognos Analytics** portal open for the exercise.

Results:
You created a revenue report using Master-detail relationships. You linked the master list with the detailed crosstab by using Country, which is a data item common to both queries.

IBM Training

Unit summary

- Create reports by joining queries
- Combine data containers based on relationships from different queries

Unit summary

IBM Training

Exercise 1

Create a report comparing quantity sold in different order years

Order method type	Quantity 2010	% Change (2010 - 2011)	Quantity 2011	% Change (2011 - 2012)	Quantity 2012
E-mail	1,986,395	-51.25%	968,453	-57.76%	409,049
Fax	688,786	-38.15%	426,006	-41.50%	249,234
Mail	488,735	-30.51%	339,635	-64.78%	119,619
Sales visit	2,640,065	-32.62%	1,778,941	-20.66%	1,411,468
Special	340,021	-25.76%	252,429	-94.60%	13,622
Telephone	3,979,898	-43.42%	2,251,898	-69.60%	684,667
Web	10,050,830	74.19%	17,507,323	31.68%	23,054,131

Create reports based on query relationships

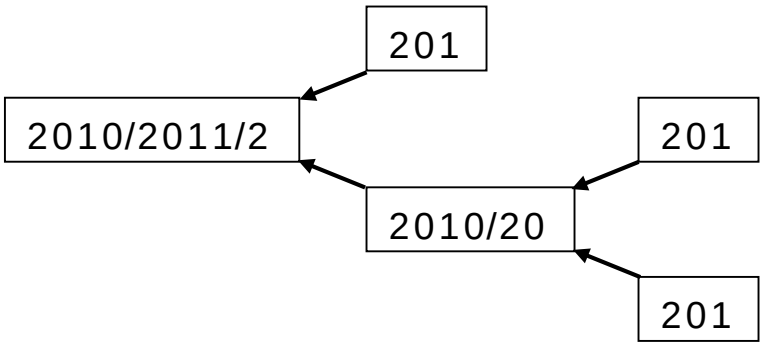
Exercise 1: Create a report comparing quantity sold in different order years

Exercise 1: Create a report comparing quantity sold in different order years

As Frank Bretton, a report author, you have been asked to create a report comparing the percentage change in quantity sold by different order methods between 2010 and 2011, and between 2011 and 2012.
To accomplish this you will:

- Create a list report using the GO data warehouse (query) package, Sales and Marketing (query) folder, Sales (query) namespace.
  - Create three queries, each containing the Order method type data item and a data item for quantity sold in one of the three order years (2010, 2011, and 2012).
  - Filter each query for the appropriate year.
  - Give each Quantity data item a descriptive name and label.
- Use joins to present this data in a single query.
- Add the data from the joined query to the list report.
- Add columns to the report (using query calculations) to display the percentage change in quantity sold from 2010 to 2011, and from 2011 to 2012.
- Format the two query calculation columns to display data as a percentage with two decimal places and then run the report.

Below is a high-level view of the query layout that you will use in this report:



For more information about where to work and the workshop results, refer to the Tasks and results section that follows. If you need more information to complete a task, refer to earlier demonstrations for detailed steps.

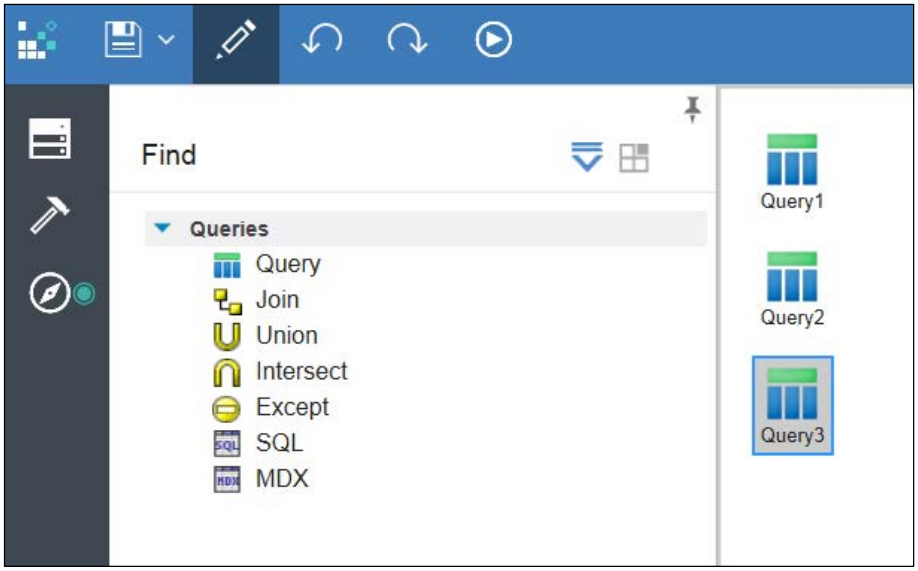
Exercise 1: Tasks and results

Portal: **http://vclassbase:9300/bi**
User/Password: **brettonf/Education1**
Package: **Team content\Samples\Models\GO data warehouse (query)**
Report Type: **List**
Folder: **Sales and Marketing (query)**
Namespace: **Sales (query)**

Task 1. Add data and a filter to three queries to show quantity sold by each order method in 2010, 2011, and 2012.

- **Side bar**: Open a new list report template using the **GO data warehouse (query)** package.
- **Navigate** tab:
  - Click the **Query explorer** tab.
  - Click **Queries**.
- **Toolbox** tab:
  - Drag two **Query** objects to the work area.

The results appear as follows:



- **Work area**: Double-click **Query1**.
- **Data\Source** tab: Add the following query items to the **Data Items** pane:
  - Order method: **Order method type**
  - Sales fact: **Quantity**
- **Detail Filters** pane: Create and validate the following detail filter:

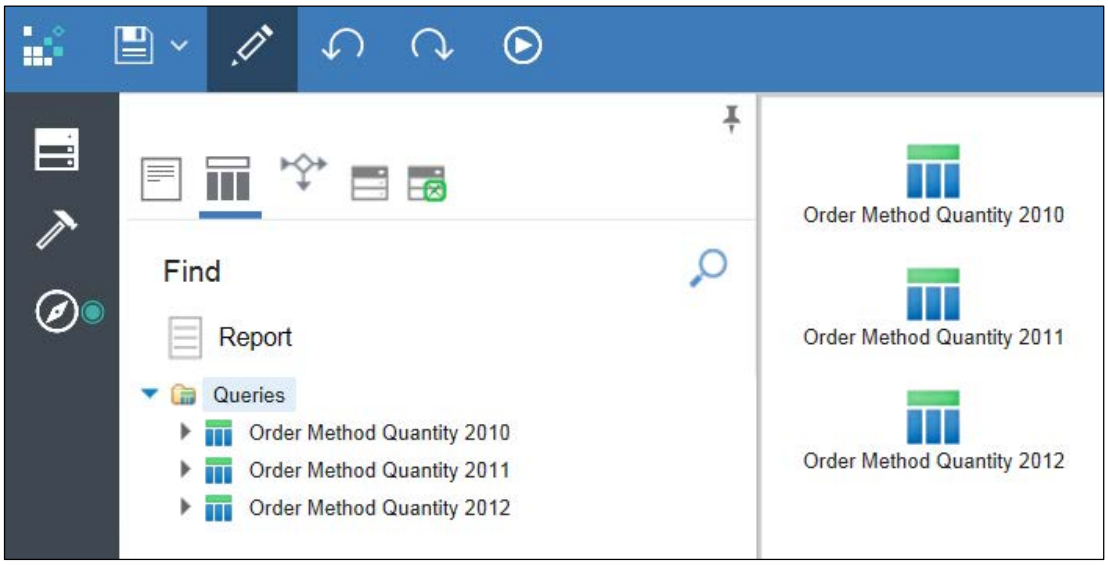
  [Sales (query)].[Time].[Year]=2010
- **Data Items** pane: Change the **Name** and **Label** properties of the **Quantity** data item, to **Quantity 2010**.
- **Navigate** tab: Click **Query2**.
- **Data\Source** tab: Add the following query items to the **Data Items** pane:
  - Order method: **Order method type**
  - Sales fact: **Quantity**
- **Detail Filters** pane: Create and validate the following detail filter :

  [Sales (query)].[Time].[Year]=2011
- **Data Items** pane: Change the **Name** and **Label** properties of **Quantity** to **Quantity 2011**.
- **Navigate** tab: Click **Query3**.
- **Data\Source** tab: Add the following query items to the **Data Items** pane:
  - Order method: **Order method type**
  - Sales fact: **Quantity**
- **Detail Filters** pane: Create and validate the following detail filter :

  [Sales (query)].[Time].[Year]=2012
- **Data Items** pane: Change the **Name** and **Label** properties of **Quantity** to **Quantity 2012**.
- **Navigate** tab.

- **Properties** pane: Rename the queries as follows:
  - Query1: **Order Method Quantity 2010**
  - Query2: **Order Method Quantity 2011**
  - Query3: **Order Method Quantity 2012**
- **Query explorer** tab: Click **Queries**

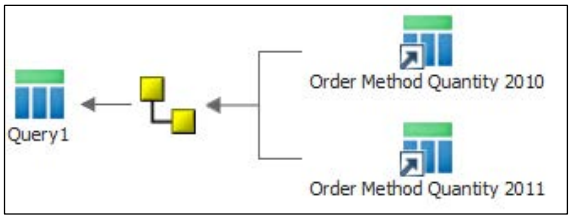
  The results appear as follows:



Task 2. Combine data from the Order Method Quantity 2010 query and the Order Method Quantity 2011 query using a join relationship, and then add data to a new query.

- **Toolbox** tab: Drag a **Join** object to the work area below the three queries.
- **Work area**: Drag the **Order Method Quantity 2010** query and the **Order Method Quantity 2011** query to the two drop zones.

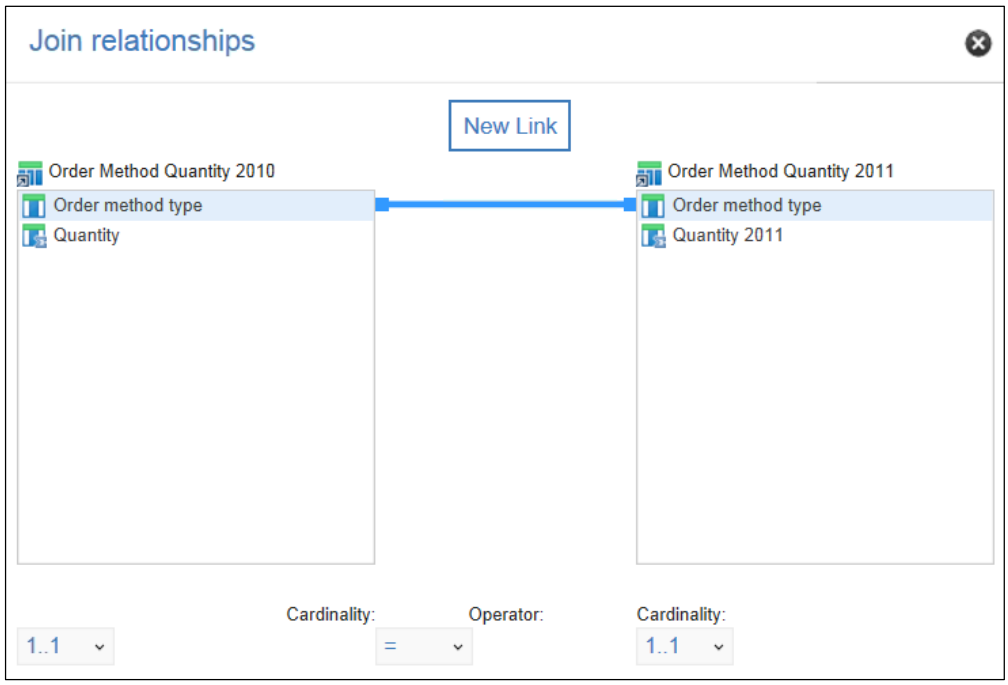
  The results appear as follows:



  - Double-click the **Join** object.

- **Join Relationships** dialog box: Click **New Link**, and then ensure a link is created between the **Order method type** items in both query lists.
  - Set the left and right **Cardinality** lists to **1..1**, and ensure that the **Operator** is set to **=**.

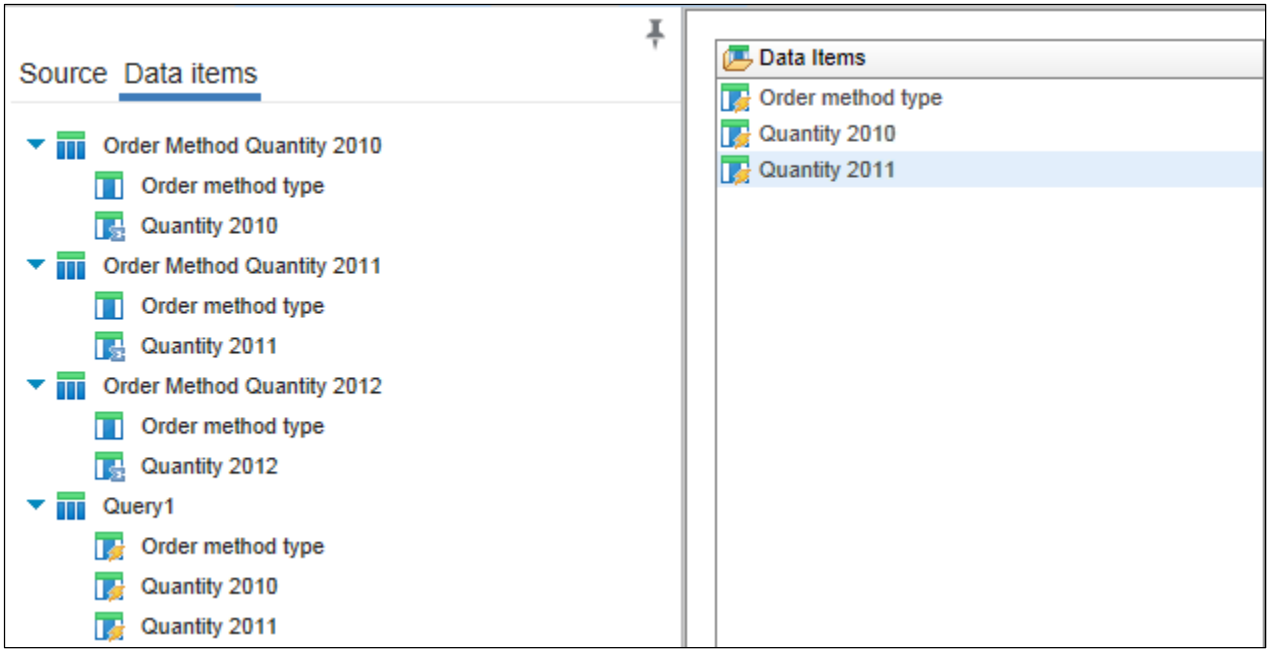
    A section of the result appears as follows:



  - Click **OK** to close the **Join Relationships** dialog box.

- **Work area**: Double-click **Query1**
- **Data\Source** tab: Click the **Data items** tab.
  - From the **Order Method Quantity 2010** query, drag **Order method type** and **Quantity 2010** to the **Data Items** pane.
  - From the **Order Method Quantity 2011** query, drag **Quantity 2011** to the **Data Items** pane.

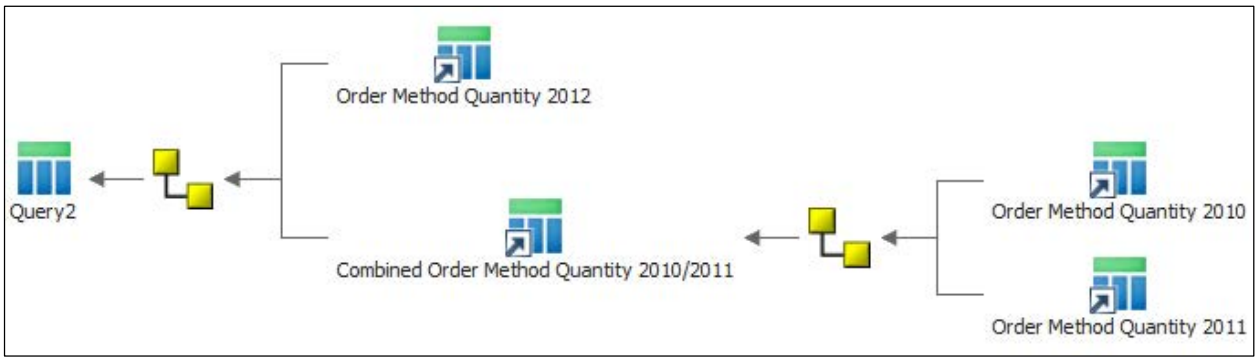
    A section of the result appears as follows:



Task 3. Combine data from the Order Method Quantity 2012 query and the Query1 query using a join relationship, and then add data to Query2.

- **Navigate** tab: Click **Queries**.
- **Toolbox** tab: Drag a **Join** object to the work area below the queries.
- **Properties** pane: Rename **Query1** as **Combined Order Method Quantity 2010/2011**.
- **Work area**: Drag the **Order Method Quantity 2012** query and the **Combined Order Method Quantity 2010/2011** query to the two drop zones.
  - Right-click a blank section of the work area, and then click **Expand References**.

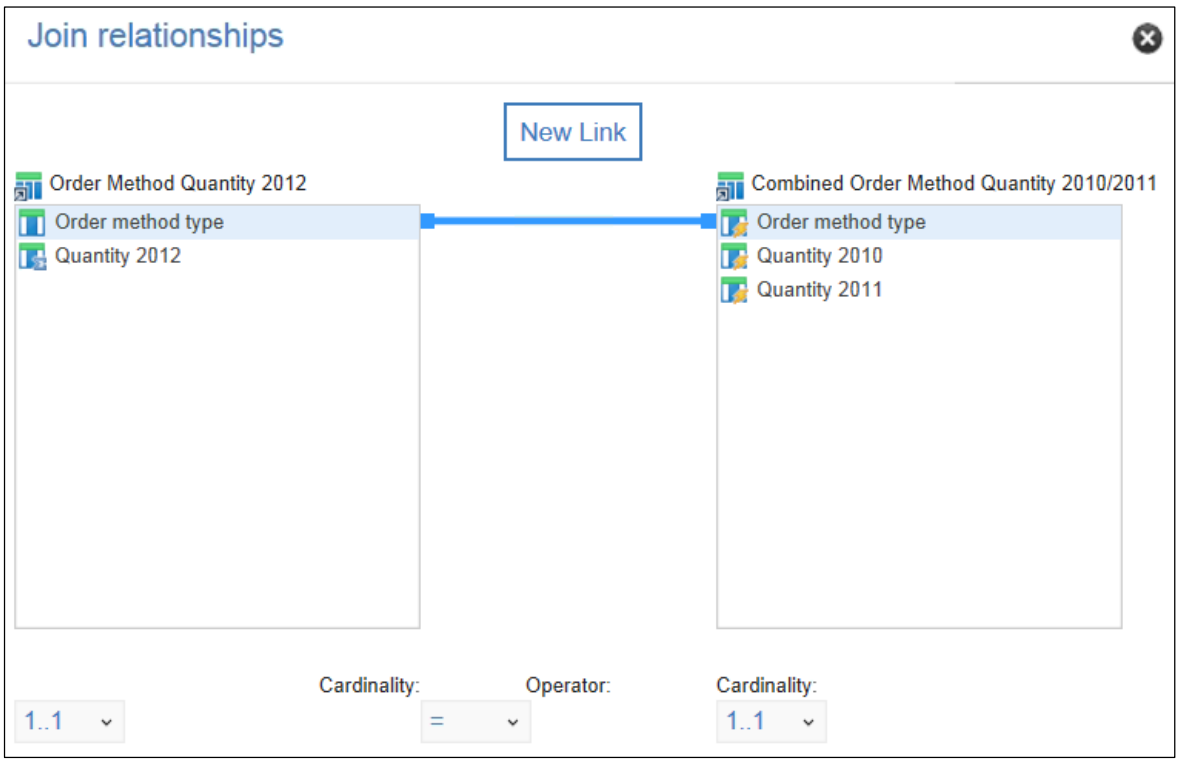
    A section of the result appears as follows:



- Double-click the **Join** object to the right of **Query2**.

- **Join Relationships** dialog box: click **New Link**.
  - Ensure a link is created between the **Order method type** items in both query lists.
  - Set the left and right **Cardinality** lists to **1..1**, and ensure that the **Operator** is set to **=**.

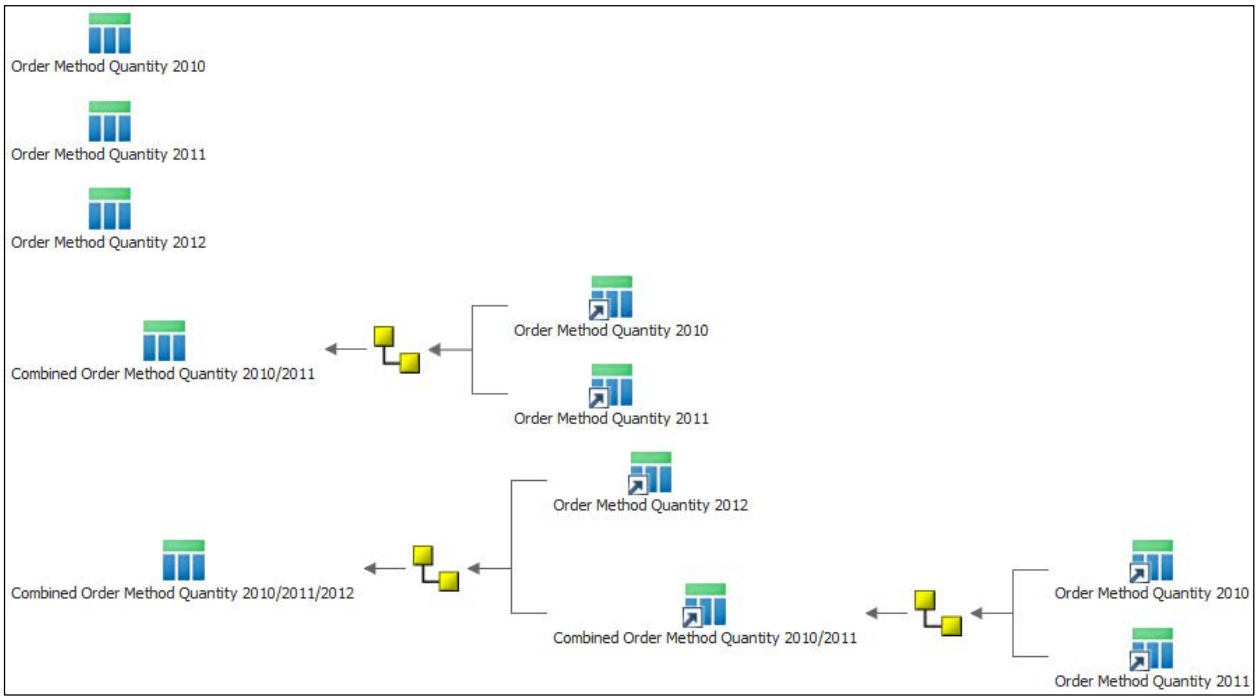
    A section of the result appears as follows:



  - Click **OK** to close the **Join relationships** dialog box.
  - **Work area**: double-click **Query2**.
- **Data/Data items** tab:
  - From the **Combined Order Method Quantity 2010/2011** query, drag the **Order method type**, **Quantity 2010** and **Quantity 2011** items to the **Data Items** pane.
  - From the **Order Method Quantity 2012** query, drag the **Quantity 2012** item to the **Data Items** pane.

- **Navigate** tab: Click **Queries**.
- **Work area**: Rename **Query2** as **Combined Order Method Quantity 2010/2011/2012**.

  The results appear as follows:



Task 4. Add data from the final joined query to the list report.

- **Navigate** tab: Click the **Page explorer** tab.
  - Click **Page1**.
- **List data container**: Click the list **Container Selector** to select the entire list.
- **Properties** pane: From the **DATA** section, change the **Query** property to **Combined Order Method Quantity 2010/2011/2012**.
- **Data/Data Items** tab: From the **Combined Order Method Quantity 2010/2011/2012** query, click the first item, Shift+click the last item, and then drag all items to the list.

  A section of the results appear as follows:

Order method type	Quantity 2010	Quantity 2011	Quantity 2012
<Order method type>	<Quantity 2010>	<Quantity 2011>	<Quantity 2012>
<Order method type>	<Quantity 2010>	<Quantity 2011>	<Quantity 2012>
<Order method type>	<Quantity 2010>	<Quantity 2011>	<Quantity 2012>

Task 5. Add query calculations displaying the percentage change in quantity sold between 2010 and 2011 and between 2011 and 2012.

- **Toolbox** tab: Expand **TEXTUAL**.
- Drag a **Query calculation** object to the right of the **Quantity 2010** column in the list.
- **Data Item expression** dialog box:
  - In the **Name** box, type the name of the calculation as **% Change (2010 - 2011)**.
  - Create and validate the following query calculation expression, using items from the **Source** tab:

    ([Combined Order Method Quantity 2010/2011].[Quantity 2011]-[Combined Order Method Quantity 2010/2011].[Quantity 2010])/[Combined Order Method Quantity 2010/2011].[Quantity 2010]
  - Click **OK**.
- **Toolbox** tab: Drag a **Query Calculation** object to the right of the **Quantity 2011** column in the list.
- **Data Item expression** dialog box:
  - In the **Name** box, type the name of the calculation as **% Change (2011 - 2012)**.
  - Create and validate the following query calculation expression, using items from the **Source** tab:

    ([Order Method Quantity 2012].[Quantity 2012]-[Combined Order Method Quantity 2010/2011].[Quantity 2011])/[Combined Order Method Quantity 2010/2011].[Quantity 2011]

    A section of the result appears as follows:

Order method type	Quantity 2010	% Change (2010 - 2011)	Quantity 2011	% Change (2011 - 2012)	Quantity 2012
<Order method type>	<Quantity 2010>	<% Change (2010 - 2011)>	<Quantity 2011>	<% Change (2011 - 2012)>	<Quantity 2012>

Task 6. Format the two calculated columns to display data as a percentage, and then run the report.

- **Work area**: Click the **<% Change (2010 - 2011)>** list column body, and then Ctrl+click **<% Change (2011 - 2012)>** list column body.
- **Properties** pane:
  - Double-click **Data format**.
  - **Format type**: **Percent**.
  - Change **Number of decimal places** to **2**.
  - Click **OK**.
- Click the **Order method type** list column body, from the On demand toolbar, click **Sort**, and then click **Ascending**.
- Run the report in **HTML**.

  The results appear as follows:

Order method type	Quantity 2010	% Change (2010 - 2011)	Quantity 2011	% Change (2011 - 2012)	Quantity 2012
E-mail	1,986,395	-51.25%	968,453	-57.76%	409,049
Fax	688,786	-38.15%	426,006	-41.50%	249,234
Mail	488,735	-30.51%	339,635	-64.78%	119,619
Sales visit	2,640,065	-32.62%	1,778,941	-20.66%	1,411,468
Special	340,021	-25.76%	252,429	-94.60%	13,622
Telephone	3,979,898	-43.42%	2,251,898	-69.60%	684,667
Web	10,050,830	74.19%	17,507,323	31.68%	23,054,131

  You have created a report comparing the percentage change in quantity sold by different order methods between 2010 and 2011, and between 2011 and 2012.
- Close the rendered report tab.
- Remove the report without saving
- Sign out of the **IBM Cognos Analytics** portal.
- Close the web browser.

IBM Training

Create advanced dynamic reports

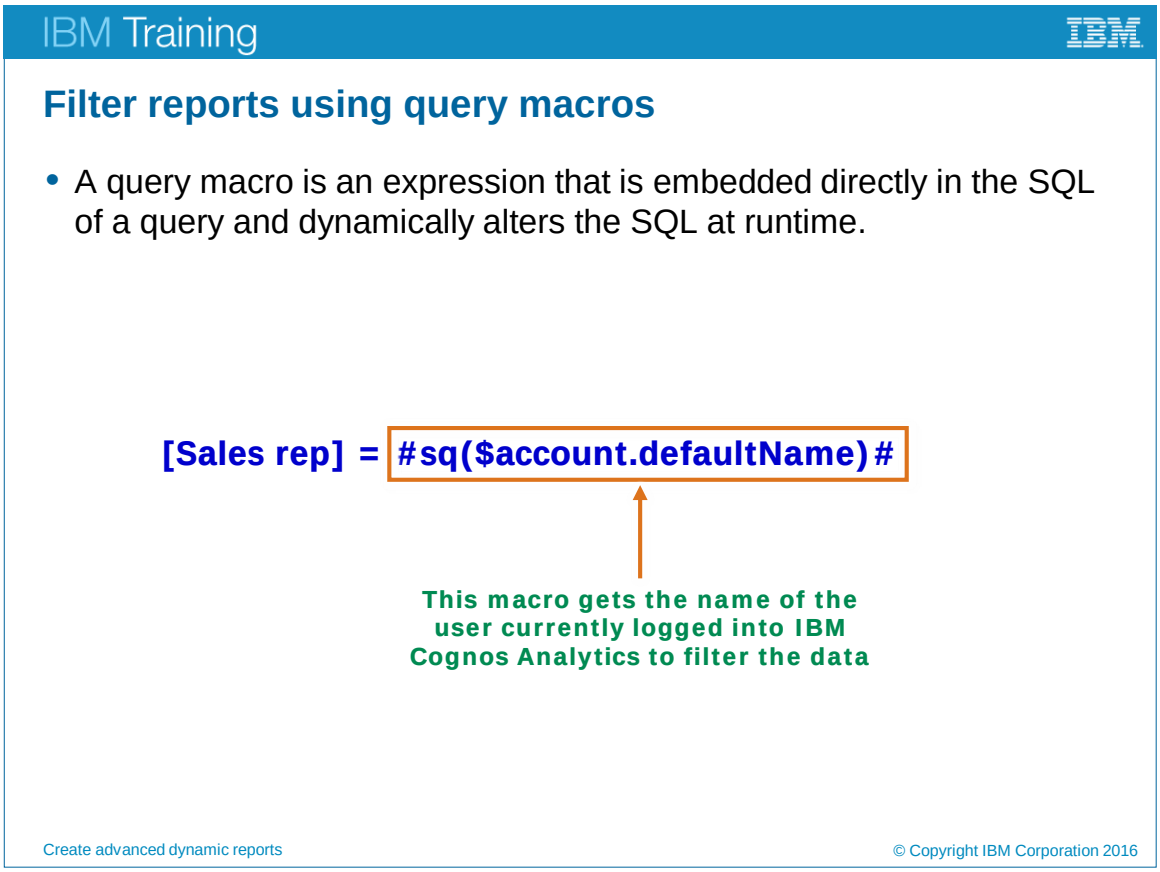
IBM Cognos Analytics (v11.0)

IBM Training

Unit objectives

- Filter reports on session parameter values
- Navigate a briefing book using a table of contents
- Create dynamic headers and titles that reflect report data
- Create a customer invoice report

Unit objectives



Filter reports using query macros

Query macros can be included anywhere in an SQL statement as long as the resulting expression is valid. You can add macros to report objects that pass expressions to the query model, such as filters and calculated query items.

Session parameters are made available to you through your authentication source. Depending on your authentication source, different session parameters are available.

Query macros can be used both in native SQL and Cognos SQL.

Examples of different authentication sources include NTLM, LDAP, or Siteminder.

In this example, you create a filter that takes the name of the person who has logged into IBM Cognos BI, and displays only that person's results to them. The same result as in this example could be achieved through bursting.

While it is better practice to place macros into the Framework Manager model, they are useful in IBM Cognos Analytics - Reporting, especially where the Framework Manager project cannot be modified or the development-time model is not available.

IBM Training

Demonstration 1

Control report output using a query macro

Sales Information for Bart Scott

Product line	Product type	Revenue
Camping Equipment	Cooking Gear	3,096,705.62
	Lanterns	1,597,289.18
	Packs	4,521,701.21
	Sleeping Bags	4,412,665.17
	Tents	6,078,370.27
Camping Equipment - Total		**19,706,731.45**
Golf Equipment	Golf Accessories	753,380.1

Create advanced dynamic reports

Demonstration 1: Control report output using a query macro

Demonstration 1: Control report output using a query macro

Purpose:
Management wants to see a report that outlines the total revenue produced by all product lines for each sales representative. They would like sales representatives in the field to be able to generate this report as needed to view their latest figures, but each rep should be able to see only their own data. You will create a report that includes revenue figures for every sales rep but filters on the current user's identity and displays only data appropriate to them.

Before performing demonstrations in this unit, on the taskbar, click Services, and then ensure that the following services are started:

- Apache Directory Server - default (start this service first, if it is not already started)
- DB2 - DB2COPY1 - DB2
- DB2DAS -DB2DAS00
- Lotus Domino Server (CProgramFilesx86IBMLotusDominodata)

Portal: **http://vclassbase:9300/bi**
User/Password: **brettonf/Education1**
Application: **IBM Cognos Analytics - Reporting**
Package: **Team content\Samples\Models\GO data warehouse (query)**
Report Type: **List**
Folder: **Sales and Marketing (query)**
Namespace: **Sales (query)**

Task 1. Create a report with revenue by sales rep.

1. Open a new **List** template using the **GO data warehouse (query)** package.
2. From the **Data/Source** tab, expand the **Sales and Marketing (query)** folder and the **Sales (query)** namespace.
3. Add the following query items to the list data container:
   - Employee by region: **Employee name**
   - Products: **Product line**, **Product type**
   - Sales fact: **Revenue**

Employee name	Product line	Product type	Revenue
<Employee name>	<Product line>	<Product type>	<Revenue>

4. In the list, click **<Employee name>**, Ctrl+click **<Product line>** list column body, and then from the toolbar click **Group / Ungroup**.

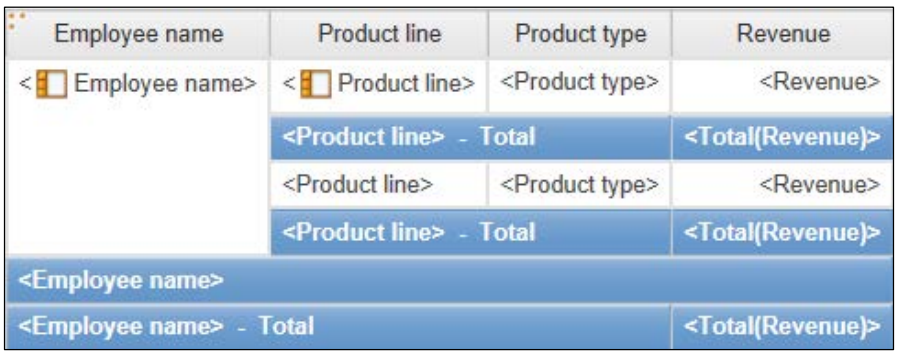
   You can use Ctrl-click or Shift-click methods to select multiple columns.

   You want to include a footer for the employee name in your list, and then summarize the revenue column.

5. Click the **<Employee name>** list column body, on the toolbar click **Headers & footers**, and then click **Create footer**.

6. Click the **<Revenue>** list column body, on the toolbar click **Summarize**, and then click **Total**.

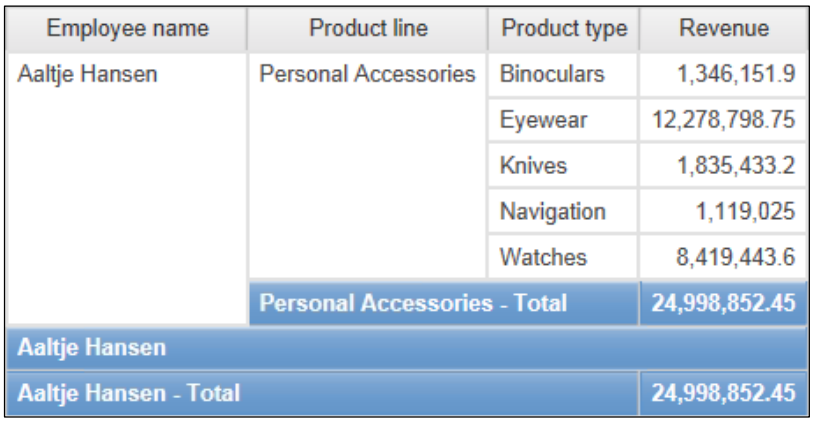
   A section of the result appears as follows:



Employee name	Product line	Product type	Revenue
<Employee name>	<Product line>	<Product type>	<Revenue>
	<Product line> - Total		<Total(Revenue)>
	<Product line>	<Product type>	<Revenue>
	<Product line> - Total		<Total(Revenue)>
<Employee name>			
<Employee name> - Total			<Total(Revenue)>

7. Run the report in **HTML**.

   A section of the result appears as follows:



Employee name	Product line	Product type	Revenue
Aaltje Hansen	Personal Accessories	Binoculars	1,346,151.9
		Eyewear	12,278,798.75
		Knives	1,835,433.2
		Navigation	1,119,025
		Watches	8,419,443.6
	Personal Accessories - Total		24,998,852.45
Aaltje Hansen			
Aaltje Hansen - Total			24,998,852.45

   This report displays revenue figures for every available employee who is a sales rep. You will filter on the Employee name to include only data for the user that is currently logged into IBM Cognos Analytics.

8. Close the rendered report tab.

Task 2. Create a filter that uses a query macro.

1. Click anywhere on the list data container, from the On demand toolbar click **Filters**, and then click **Edit Filters**.
2. Click **Add**, click **Advanced**, and then click **OK**.
3. Create and validate the following expression:

   [Employee name]=#sq($account.defaultName)#.

   Hint: Drag Employee name from the Data Items tab.

   The pound symbols (#) enclose the query macro expression. The variable account.defaultName retrieves the defaultName session parameter from the current IBM Cognos session. The sq expression encloses the variable in single quotes. The expression is case sensitive.

   Notice that there is a Macros tab available, for you to make use of some common macros, but you are using a session parameter that is not available in this list. IBM Cognos Analytics - Reporting does not have access to session parameters. The Framework Manager modeler can provide a list of these parameters.
4. Click **OK** to close each open dialog box.
5. Run the report in **HTML**.

   The report displays no data. You are logged in as a report author (Frank Bretton), not as a sales rep, so you do not see any corresponding sales data. You need to log in to IBM Cognos using one of the sales rep accounts.
6. Close the rendered report tab, and then save the report in the **Team content\B6059** folder as **Sales Rep Revenue**.
7. On the **Application** bar, click **Frank Bretton**, and then click **Sign out**.
8. Sign in using **smythed/Education1**, and then click **Sign in**.
9. On the **IBM Cognos Analytics** portal page, navigate to **Team content/B6059**.

10. Click **Sales Rep Revenue**, to run the report.

    A section of the result appears as follows:

Employee name	Product line	Product type	Revenue
Dave Smythe	Camping Equipment	Cooking Gear	1,742,564.12
		Lanterns	820,939.9
		Packs	2,463,562.46
		Sleeping Bags	1,209,711.63
		Tents	4,745,739.82
	Camping Equipment - Total		10,982,517.93
	Golf Equipment	Golf Accessories	147,287

    The report now only displays sales data for Dave Smythe.

11. Sign out of the **IBM Cognos Analytics** portal.

    Since the report is designed to display data for only one sales rep at a time, there is no need to include the Employee name column. You can now log back in to IBM Cognos Analytics - Reporting as the report author and perform this and other formatting functions.

Task 3. Format the report.

1. On the **Sign in** page, sign in as **brettonf/Education1**.
2. On the **IBM Cognos Analytics** portal page, navigate to **Team content\B6059**, click the **More** button of the **Sales Rep Revenue** report, and then click **Edit report**.
3. In the list data container, click the **<Employee name>** list column body, and then from the On demand toolbar, under **More**, click **Cut**.
4. Click the **<Employee name>** summary row (not the text item), and then click **Cut**.

   A section of the result appears as follows:

Product line	Product type	Revenue
<Product line>	<Product type>	<Revenue>
<Product line> - Total		<Total(Revenue)>
<Product line>	<Product type>	<Revenue>
<Product line> - Total		<Total(Revenue)>
<Employee name> - Total		<Total(Revenue)>

5. Click the list **Container Selector** to select the entire list.
6. From the **Application** bar, click **Show properties** to open the **Properties** pane.

7. From the **Properties** pane, in the **DATA** section, double-click **Properties**, select **Employee name** to make **Employee name** a property of the query, and then click **OK**.
8. Double-click the title text in the page header, change the title to **Sales Information for**, press the space bar, and then click **OK**.
9. Click any column in the list, in the **Properties** pane, click **Select Ancestor**, and then click **Page**.
10. In the **Properties** pane, in the **DATA** section, change the **Query** property to **Query1**.
11. On the **Toolbox** tab, expand **TEXTUAL**, and then drag a **Layout calculation** object to the end of the report title.
12. In the Report expression dialog box, drag **Employee name** to the Expression Definition pane.

    The expression appears as [Query1].[Employee name].
13. Click **Validate**, and then click **OK**.
14. Click the report title, and then on the toolbar, click **Pick up style**  .
15. Click the **<% [Query1].[Em… %>** calculation in the title, and then on the toolbar click **Apply style**  .
16. Click to the left of the title to select the header box, and then from the toolbar, click **Left**  to left align the title.
17. Save the report, and then sign out of the **IBM Cognos Analytics** portal.

Task 4. View the report for various sales reps.

1. On the **Sign in** page, sign in as **smythed/Education1**.
2. On the **IBM Cognos Analytics** portal page, navigate to the **Team content/B6059** folder.

3. Click **Sales Rep Revenue** to run the report.

   A section of the result appears as follows:

Sales Information for Dave Smythe

Product line	Product type	Revenue
Camping Equipment	Cooking Gear	1,742,564.12
	Lanterns	820,939.9
	Packs	2,463,562.46
	Sleeping Bags	1,209,711.63
	Tents	4,745,739.82
Camping Equipment - Total		**10,982,517.93**
Golf Equipment	Golf Accessories	147,287

4. Sign out of **IBM Cognos Analytics**, sign in as **scottb/Education1**, and then run the report again.

   A section of the results appear as follows:

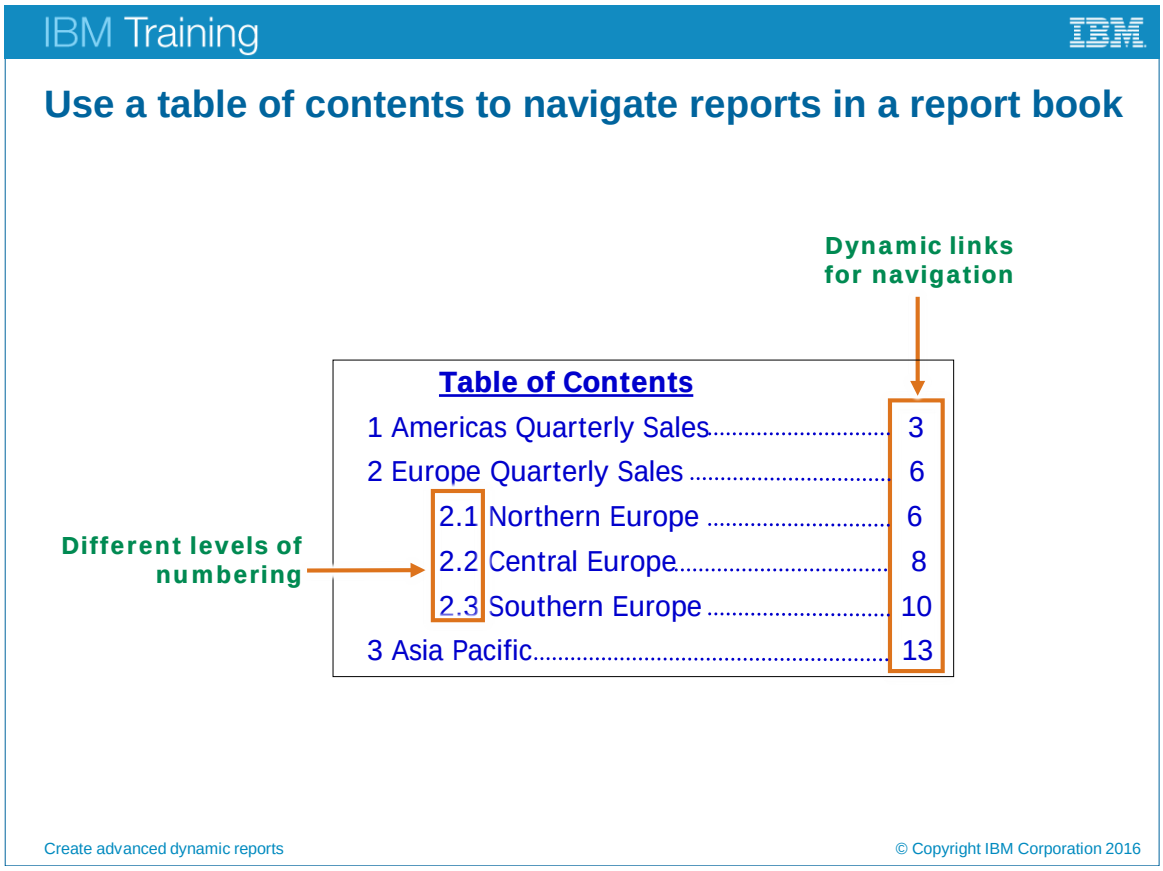
Sales Information for Bart Scott

Product line	Product type	Revenue
Camping Equipment	Cooking Gear	3,096,705.62
	Lanterns	1,597,289.18
	Packs	4,521,701.21
	Sleeping Bags	4,412,665.17
	Tents	6,078,370.27
Camping Equipment - Total		**19,706,731.45**
Golf Equipment	Golf Accessories	753,380.1

   The query changes each time the report is run by a different user.

5. Sign out, and then sign in as **brettonf/Education1**.
6. Leave the **IBM Cognos Analytics** portal open for the next demonstration.

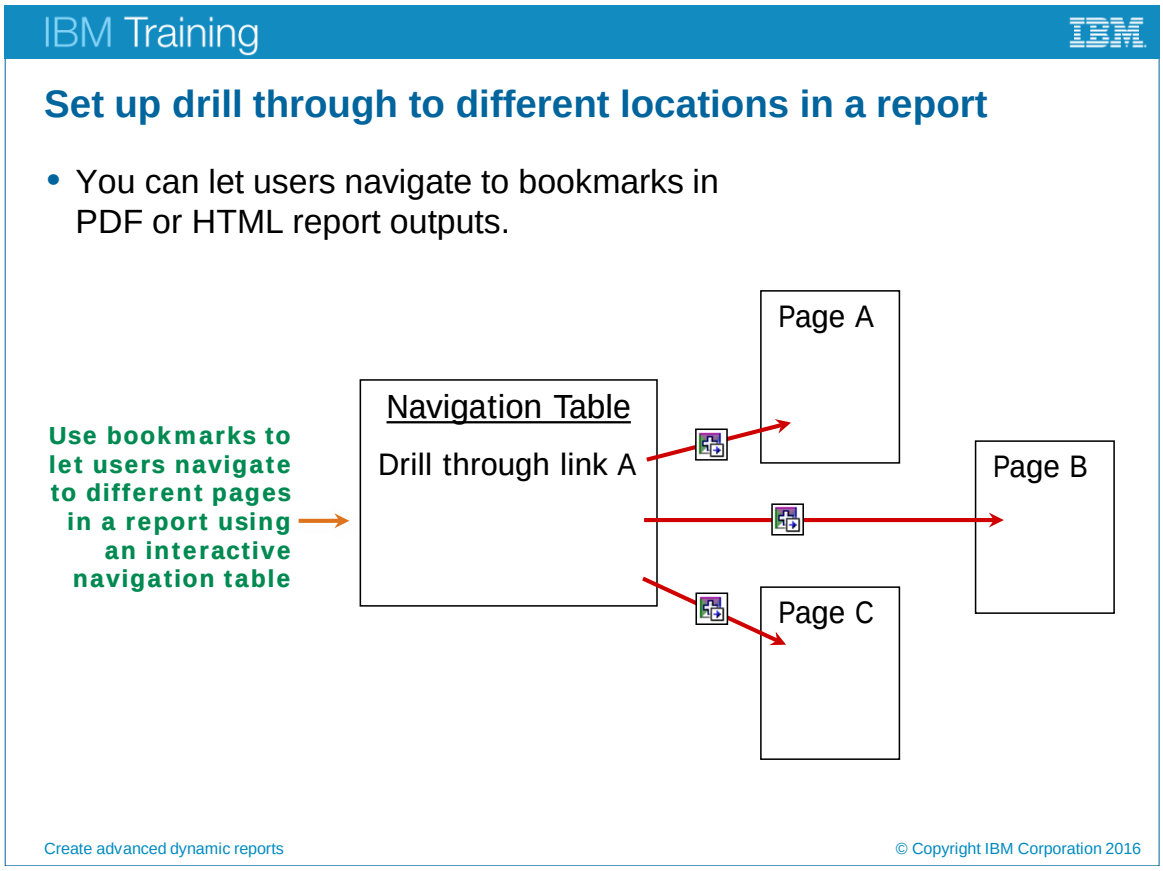
Results:
You created a report that outlines the total revenue produced by all product lines for each sales rep. You then filtered this report using a query macro to display sales data for the user currently signed into IBM Cognos Analytics. After viewing the report while signed in as a particular sales rep, you formatted the report layout to include the name of the sales rep in the header. Finally, you examined this new report from the perspective of two different sales reps.



Use a table of contents to navigate reports in a report book

You can create a table of contents to quickly navigate reports with:

- sectioned items
- grouped items
- multiple pages



Set up drill through to different locations in a report

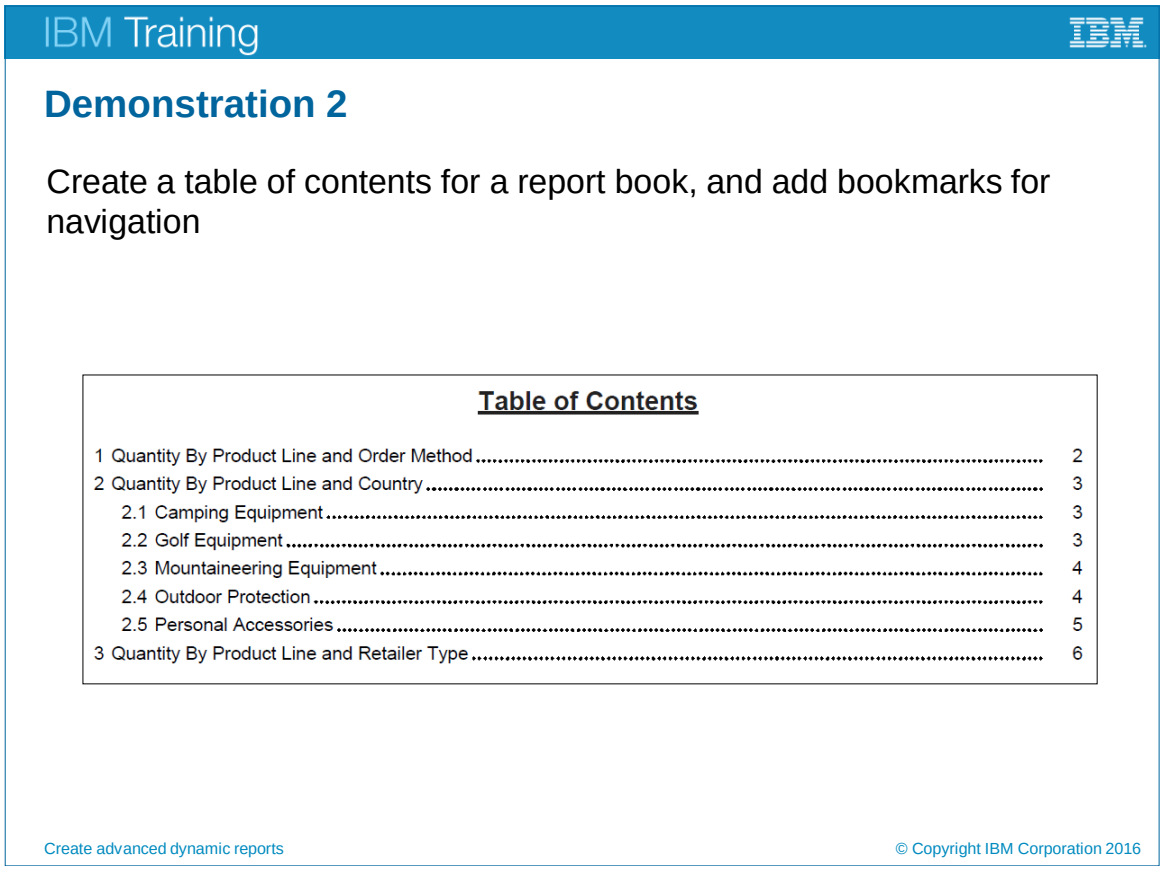
Set up drill-through access to bookmarks to let users navigate:

- to related information within the same report
- from one report to the relevant section of a second report, without filtering the second report

Drill-through access to bookmarks will work only with saved report outputs such as HTML and PDF.

An example of drill-through access to bookmarks is to let users drill through from a list containing images of products to more details about each product contained in another location in the report.

An example of setting up drill-through access for a package to bookmarks in a target report: Your target report displays sales made using different order methods and has a bookmark in the heading where each new order method begins in the report. The source type for this bookmark is data item values. You could set up drill-through access to this report so that data item values are passed to the bookmarks in the target report. If someone drilled through from Fax, for example, they would be taken to the location in the target report where data about Fax orders begins.



Demonstration 2: Create a table of contents for a report book, and add bookmarks for navigation

Demonstration 2: Create a table of contents for a report book, and add bookmarks for navigation

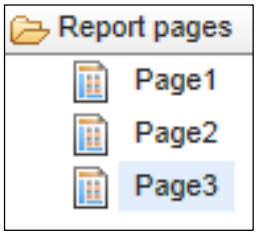
Purpose:
You have been asked to create a report book that contains the following reports: quantity by product line and order method, quantity by product line and country, and quantity by product line and retailer type. This report needs to be distributed in PDF format. You will create a table of contents and add bookmarks to let users navigate quickly to the desired report page and information within the report.

Portal:	**http://vclassbase:9300/bi**
User/Password:	**brettonf/Education1**
Application:	**IBM Cognos Analytics - Reporting**
Package:	**Team content\Samples\Models\GO data warehouse (query)**
Report Type:	**List**
Folder:	**Sales and Marketing (query)**
Namespace:	**Sales (query)**

Task 1. Create three list report pages.

You will copy the first list report page to two other pages so that they will have a header, footer and list already on them. You will then add data and a name to each report page to represent what is on the report.

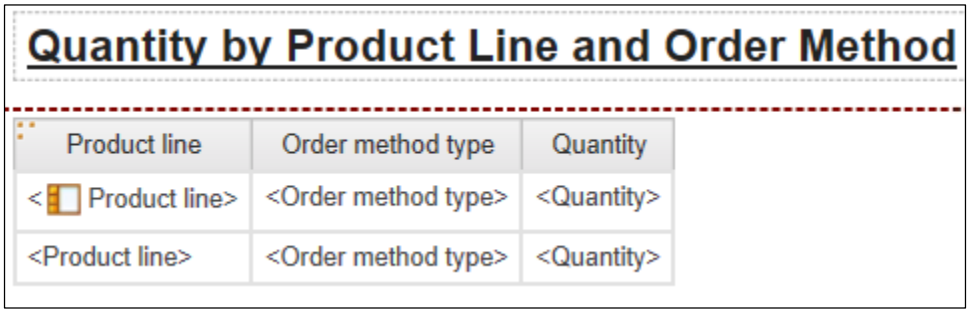
1. Open a new **List** template using the **GO data warehouse (query)** package.
2. On the **Navigate** tab, click **Report pages**.
3. Ctrl+click and drag **Page1** below **Page1** to create a copy.
4. Repeat step **3** so that you have three pages.



5. On the **Application** bar, open the **Properties** pane.
6. For each page, in the **Report pages** pane, click the page to select it, and then in the **Properties** pane, under the **MISCELLANEOUS** section, change the **Name** property as follows:
   - Page1: **Quantity by Product Line and Order Method**
   - Page2: **Quantity by Product Line and Country**
   - Page3: **Quantity by Product Line and Retailer Type**

Task 2. Populate the list report on each page.

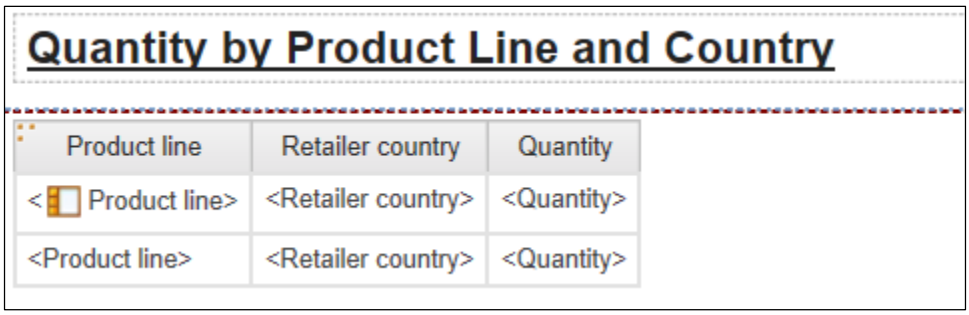
1. Double-click the **Quantity by Product Line and Order Method** page, and then in the report header, double-click the title and change the title to **Quantity by Product Line and Order Method**.
2. Click to the left of the title to select the header box, and then click **Left**.
3. From the **Data/Source** tab, expand **Sales and Marketing (query)** folder and the **Sales (query**) namespace.
4. Drag the following query items to the list:
   - Products: **Product line**
   - Order method: **Order method type**
   - Sales fact: **Quantity**
5. Group the **<Product line>** list column body.
   The results appear as follows:



6. On the **Navigate** tab, click the **Quantity by Product Line and Country** page, and then in the report header, change the title to **Quantity by Product Line and Country**.
7. Left align the title header.

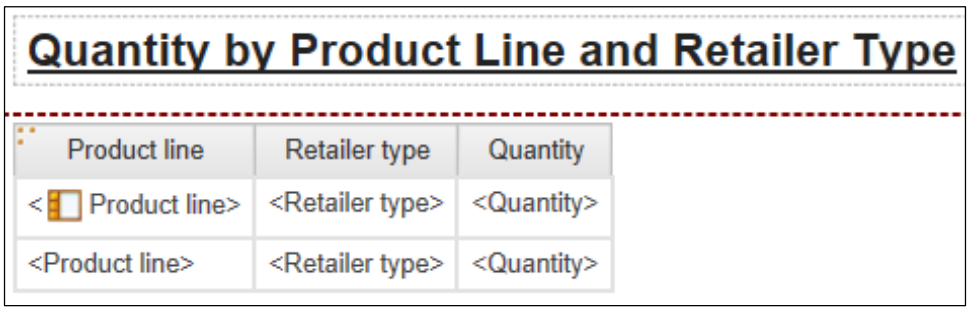
8. On the **Data**/**Data items** tab, drag the following query items to the list:
   - **Product line**
   - **Quantity**
9. On the **Data**/**Source** tab, from the **Retailers** query subject, drag **Retailer country** between the **Product line** column and the **Quantity** column.
10. Group the **<Product line>** list column body.

    The results appear as follows:



11. On the **Navigate** tab, click the **Quantity by Product Line and Retailer Type** page, change the title to **Quantity by Product Line and Retailer Type**, and then left align the title header.
12. On the **Data**/**Data items** tab, drag the following query items to the list:
    - **Product line**
    - **Quantity**
13. On the **Data**/**Source** tab, from the **Retailer type** query subject, drag **Retailer type** between the **Product line** column and the **Quantity** column.
14. Group the **<Product line>** list column body.

    A section of the result appears as follows:

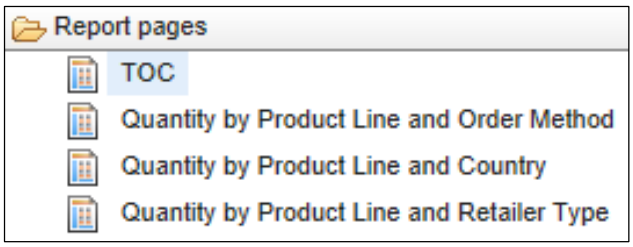


Task 3. Create a table of contents page.

You will create a page for the table of contents by copying one of the existing pages, renaming the page and title, and deleting the list object. You will then create a table of contents on the page.

1. On the **Navigate** tab, click **Report pages**.
2. In the Report pages pane, Ctrl+click and drag **Quantity by Product Line and Order Method** to the top of the list, to create a copy.
3. In the **Properties** pane, under the **MISCELLANEOUS** section, change the **Name** property to **TOC**.

   The Report Pages pane displays the four named pages.

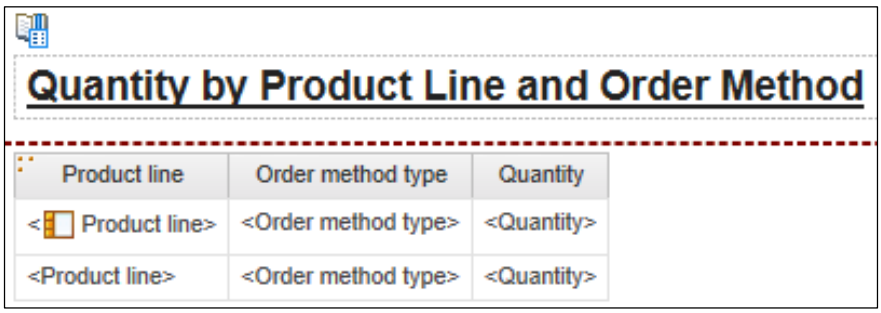


4. On the **Navigate** tab, click **TOC**.
5. Click the list **Container Selector**, and then on the toolbar, under **More**, click **Delete**.
6. Change the title in the header to **Table of Contents**.
7. On the **Toolbox** tab, expand **ADVANCED**, and then drag a **Table of contents** object to the work area.

Task 4. Add a Table of Contents Entry to the report pages.

1. On the **Navigate** tab, click the **Quantity by Product Line and Order Method** page.
2. From the **Toolbox** tab, expand **ADVANCED**, and then drag a **Table of contents entry** object to the top left corner of the page, above the title in the header.

   The results appear as follows:



3. Repeat steps **1** and **2** to include a **Table of Contents Entry** object on the **Quantity by Product Line and Country** and the **Quantity by Product Line and Retailer Type** pages.
4. On the **Navigate** tab, click **TOC**.

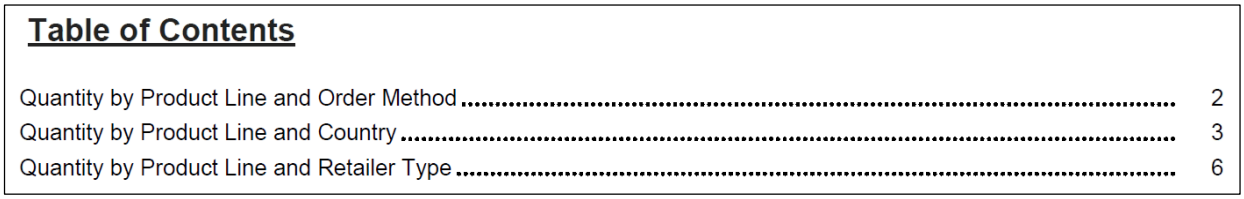
   The TOC itself is constructed of TOC entries, each of which is a 3 cell table constructed with text, dashed lines and page numbers.

   You have the choice of typing the name of the page, or you can change the text to a report expression to include the PageName. If you type the page name, you have to maintain the page name manually if the pages are re-ordered. If you use PageName, then the page name will be updated dynamically even if you re-order the page.

   You will use the PageName method in the next few steps.
5. Click the first text item, and then in the **Properties** pane under **TEXT SOURCE**, change the **Source type** property from **Text** to **Report expression**.
6. In the **Properties** pane, under **TEXT SOURCE**, click the **Report expression** property, and then click the ellipsis.
7. On the **Functions** tab, expand **Report Functions**, double-click **PageName**, and then click **OK** to close the **Report expression** dialog box.
8. Click the **<%PageName()%>** text that you have just created, Ctrl+click and drag a copy of the function to each row of the table just in front of the the text.
9. Delete **Double-click to edit text** from each row.
10. Run the report in **PDF**.

    Table of Contents works only for reports produced in PDF and non-interactive HTML format (when viewing saved report outputs).

    The result appears as follows:



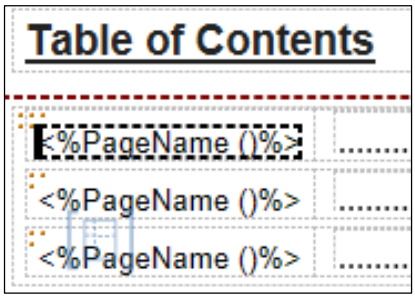
    Three things to note:

    - The PageName now appears in the Table of Contents. This is the page name that you see in Page explorer.
    - The page numbers reflect the actual starting page of the item so as more or less data appears in the report, the page numbers will always be correct.
    - When you move the mouse cursor over each item, it changes shape to indicate a live link. You can click the link to go to that section of the report.
11. Close the rendered report tab.

Task 5. Add numbering to the Table of Contents.

You will format the Table of Contents to have numbers for each entry.

1. On the **TOC** page, from the **Toolbox** tab, expand **TEXTUAL**, and then drag a **Layout calculation** object into the **TOC**, before the first **<%PageName()%>**, as follows:



2. In the **Expression Definition** pane, create and validate the following expression:

   TOCHeadingCount(1)

   Hint: TOCHeadingCount is on the Functions tab, in the Report Functions folder. You may also type the function expression, but be careful of syntax.

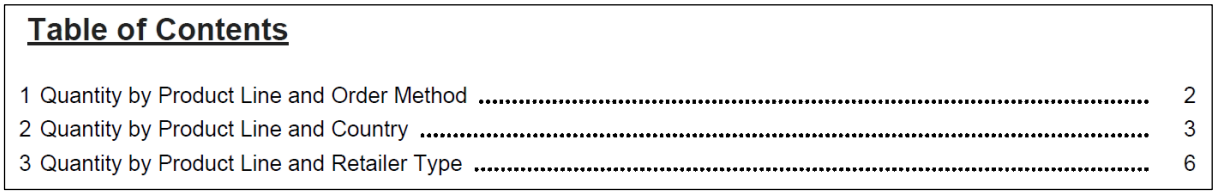
   The value 1 represents the Heading level property found on a Table of contents entry. When you create the Table of contents entry, you can choose which level the entry will be accounted for in the table of contents.

   The functionality of the Heading level property will be more clearly understood after a second level of numbering is added to the TOC.

3. Click **OK** to close the **Report expression** dialog box.
4. Click the new layout expression, and then Ctrl+click and drag a copy into each of the rows.
5. With the first **<%PageName ()%>** item selected, Ctrl+click the other **<%PageName ()%>** items to select multiple items, in the **Properties** pane, under the **BOX** section, change the **Padding** property to **Left padding** of **5 px**, and then click **OK**.

6. Run the report in **PDF**.

   The result appears as follows:



   The TOCHeadingCount () function returns the TOC1 entry count. Since you added three TOC entries, each belonging to Table of Contents 1, there are three entries that display on this page.

7. On the right, click page **3**.

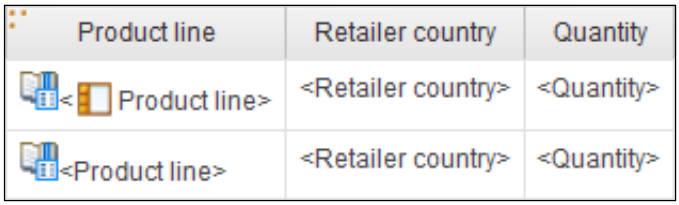
   You are brought to the Quantity by Product Line and Country report on page 3, as indicated in the Table of Contents.

8. Close the rendered report tab.

Task 6. Add data driven TOC entries to the Table of Contents page.

1. On the **Navigate** tab, click the **Quantity by Product Line and Country** page.
2. On the **Application** bar, under **More**, click **Locked** to unlock the report.
3. On the **Toolbox** tab, expand **ADVANCED**, and then drag a **Table of contents entry** object into the list, before **<Product line>**.

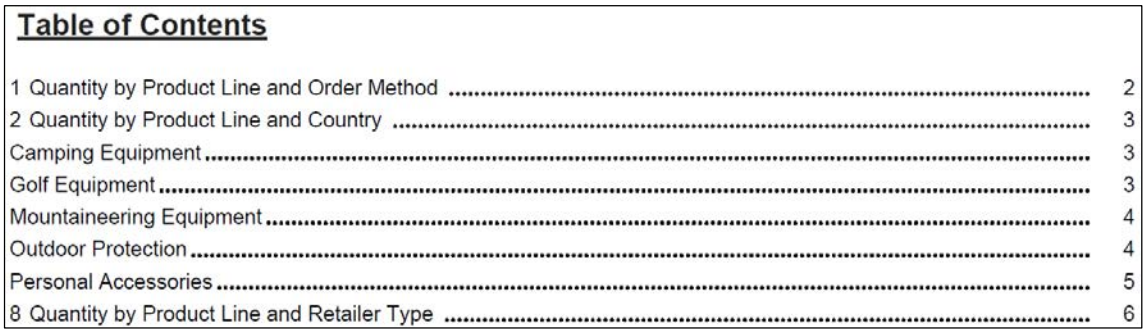
   The result appears as follows:



4. Run the report in **PDF**.

   The table of contents now displays the data values that are in the report.

   The results appear as follows:



   The Product line entries do not yet have an explicit number at the start, because you will add a second level of numbering to the table of contents, and indent the Product line entries.

5. Close the rendered report tab.

Task 7. Add a second level of numbering and indent the product line on the Table of Contents page.

You will build a second level of numbering based on, (<Level1>.<Level2> <Product line>). For example: 2.1 Camping Equipment.

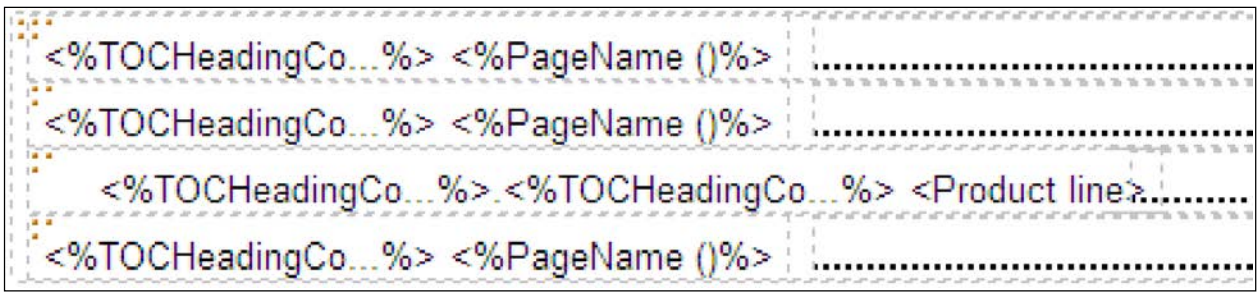
1. With the report still unlocked, click the **Table of contents entry** beside **<Product line>.**
2. In the **Properties** pane, under **GENERAL**, change the **Heading level** property to **2**.
3. On the **Application** bar, under **More**, click **Unlocked** to lock the report.
4. On the **Navigate** tab, click **TOC**.
5. From the **Toolbox** tab, expand **TEXTUAL**, and then insert a **Layout calculation** in front of the **<Product line>** entry.
6. In the **Expression Definition** pane, create and validate the following expression:

   TOCHeadingCount(2)
7. Click **OK**.
8. From the **Toolbox** tab, insert a **Text item** in front of the new layout calculation, type a period (**.**) and then click **OK** to close the **Text** dialog box.
9. Insert a **Layout calculation** in front of the text item you just added.
10. In the **Expression Definition** pane, create and validate the following expression:

    TOCHeadingCount(1)

11. Click **OK**.
12. With this last layout calculation selected (**<%TOCHeadingCo…%>**), in the **Properties** pane, under the **BOX** section, set the **Left padding** property to **20px**, and then click **OK**.
13. Click **<Product line>**, and then set **Left padding** to **5px**.

    A section of the report layout appears as follows:



14. Run the report in **PDF**.

    The result appears as follows:

Table of Contents

    The second level of numbering appears and the entries are indented.
15. Close the rendered report tab.

Task 8. Add a Top link and a bookmark for better navigation.

You want to create a text item at the bottom of the list that will have a drill-through definition for users to return to a bookmark the beginning of the list.

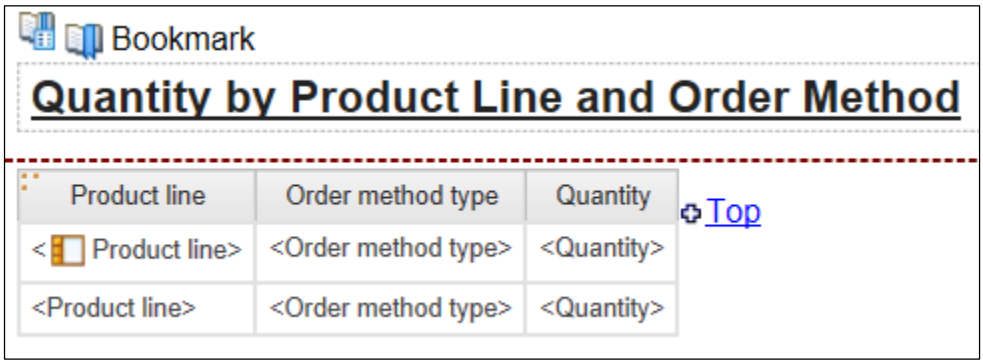
1. On the **Navigate** tab, click **Quantity by Product Line and Order Method**.
2. On the **Toolbox** tab, drag a **Text item** outside of the list, to the right of the list.
3. In the **Text** dialog box, type **Top**, and then click **OK**.
4. Click the **Top** text item, from the On demand toolbar, under **More**, click **Drill-Through Definitions**.
5. Click **New Drill-Through Definition**  , and then click the **Bookmark** tab.
6. In the **Source type** list, select **Text**, and then to the right of the **Text** box, click the ellipsis.

7. Type **Top of Product Line and Order Method**, click **OK** to close the **Text** dialog box, and then click **OK** to close the **Drill-through definitions** dialog box.

   Hint: If you copy the text from the Text dialog box while it is open, you can paste it instead of typing it again at step 9.

   The bookmark that you referenced in the drill-through definition does not yet exist. You will create the bookmark of the top of the report.
8. On the **Toolbox** tab, under **ADVANCED**, drag a **Bookmark** to the right of the **Table of contents entry** object, in the header.
9. With the **Bookmark** item selected, in the **Properties** pane, under the **BOOKMARK SOURCE** section, change the **Label** property to **Top of Product Line and Order Method**.
10. Click **OK**.

    A section of the report layout appears as follows:



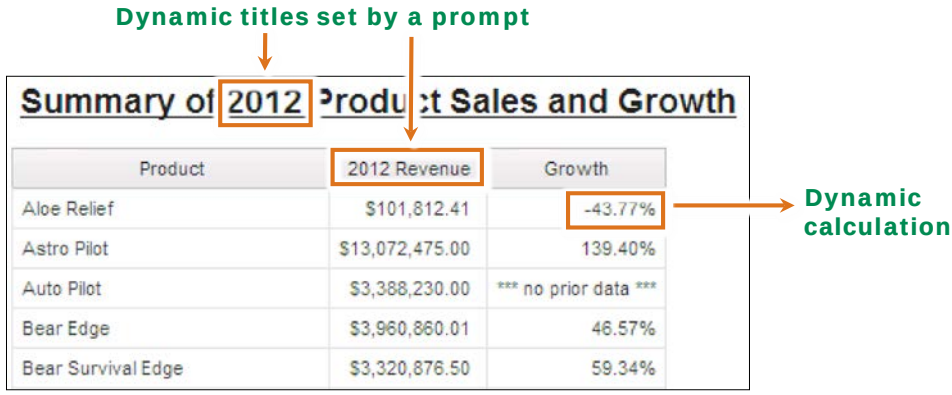
    Optionally, repeat steps 1 through 10 on the pages that you would like to have this Top navigation functionality, changing the bookmark and drill-through definitions to have unique names, such as Top of Product Line and Country and Top of Product Line and Retailer Type.
11. Run the report in **PDF**.
12. Navigate through the report using the table of contents and using bookmark navigation to return to the top of a page.
13. Close the rendered report, then remove the report without saving
14. Leave the **IBM Cognos Analytics** portal open for the next demonstration.

Results:
You created a report book with a table of contents and bookmarks for users to quickly navigate to the desired report. You added numbering and data driven table of content entries that used a second level of numbering.

IBM Training

Create dynamic data items and titles

- Use calculated data items to:
  - create dynamic column titles and report titles
  - display user prompt selections



Create advanced dynamic reports

Create dynamic data items and titles

To create a dynamic column header:

1. Create a calculated data item in the query that will display the title you want.
2. Change the column title Source type property to Data item value.
3. Change the column title Data item value property to the calculated item.

For report titles, follow the same steps, but set the Source type property to Report Expression (not Data Item), and set the value of the report expression to the calculated item.

By default, a List column title displays the name (label) of the data item it is associated with. In the Properties pane, you can change the Content property of the List Column Title to Value, and the column title will display the first data value retrieved by the data item. In crosstabs, column and row titles display the data values returned by the data item associated with it. For both lists and crosstabs, you can associate the column or row title with a different data item (and also change the content to Label or Value) to display different information in the title. You can associate the title with any data item in the query.

IBM Training

Demonstration 3

Create a dynamic revenue growth report

Summary of 2012 Product Sales and Growth

Product	2012 Revenue	Growth
Aloe Relief	$101,812.41	-43.77%
Astro Pilot	$13,072,475.00	139.40%
Auto Pilot	$3,388,230.00	*** no prior data ***
Bear Edge	$3,960,860.01	46.57%
Bear Survival Edge	$3,320,876.50	59.34%

Create advanced dynamic reports

Demonstration 3: Create a dynamic revenue growth report

Demonstration 3: Create a dynamic revenue growth report

Purpose:
Management wants to see a report that outlines the total revenue generated for each product in any given year. The report should also show the percentage growth in revenue between the year of interest and the prior year. To do this, you will use calculated data items to retrieve and derive data based on the year chosen by the user, and to name the column headers and report title accordingly.

Portal:	**http://vclassbase:9300/bi**
User/Password:	**brettonf/Education1**
Application:	**IBM Cognos Analytics - Reporting**
Package:	**Team content\Samples\Models\GO data warehouse (query)**
Report Type:	**List**
Folder:	**Sales and Marketing (query)**
Namespace:	**Sales (query)**

Task 1. Create a query.

You will create a query containing the date you require and a parameter users can use to select an order year at runtime.

1. Open a new **List** template, using the **GO data warehouse (query)** package.
2. On the **Navigate/Query explorer** tab, click **Query1**.
3. On the **Data/Source** tab, expand the **Sales and Marketing (query)** folder, and the **Sales (query)** namespace.
4. Add the following data item to the **Data Items pane**: Products: **Product**.

   You will add a parameter for users to select an order year to view data when they run the report.
5. On the **Toolbox** tab, drag a **Data Item** to the **Data Items** pane.
6. In the **Data item expression** dialog box, update the **Name** field to **Order year_prompt**.
7. In the **Expression Definition** pane, create and validate the following expression:

   ?Order year?

8. Click **OK** to close the **Data item expression** dialog box.

   You will add a data item that can be used to return only data for the year that the user selects in the prompt.
9. From the **Toolbox** tab, drag a **Data Item** to the **Data Items** pane.
10. In the **Data item expression** dialog box, update the **Name** field to **Y2 Revenue**.
11. In the **Expression Definition** pane, create and validate the following expression:

    if ([Sales (query)].[Time].[Year]=[Order year_prompt]) then ([Sales (query)].[Sales fact].[Revenue]) else (0)

    Hints:

    - **Source** tab: **Time**: **Year**
    - **Data items** tab: **Order year_prompt**
    - **Source** tab: **Sales fact**: **Revenue**

    This data item will retrieve every record from the Time dimension query subject and examine the year of each one. If the year is equal to that specified in the prompt, then the Revenue data is retrieved and aggregated, otherwise the number 0 is added to the running Revenue total.
12. Click **OK** to close the **Data item expression** dialog box.

Task 2. Add additional data to the query.

You will add a second data item to retrieve data for revenue of the previous year.

1. In the **Data Items** pane, double-click the **Y2 Revenue** data item, copy the expression in the **Expression Definition** pane, and then click **OK**.
2. From the **Toolbox** tab drag a **Data Item** to the **Data Items** pane.
3. In the **Data item expression** dialog box, in the **Name** field, type **Y1 Revenue**.
4. In the **Expression Definition** pane, create and validate the following expression:

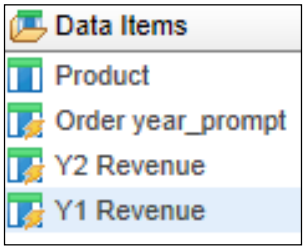
   if ([Sales (query)].[Time].[Year]=[Order year_prompt]-1) then ([Sales (query)].[Sales fact].[Revenue]) else (0)

   Hint: At the end of **[Order year_prompt]**, type **-1**.

   This expression is the same as the one you created earlier except it will retrieve revenue for the previous order year.

5. Click **OK** to close the **Data item expression** dialog box.

   Four data items are currently in the query.



Task 3. Add calculated items and a filter to the query.

You will create a calculated item for the percentage growth over the previous year.

1. From the **Toolbox** tab, drag a **Data Item** object to the **Data Items** pane.
2. In the **Data item expression** dialog box, in the **Name** field, type **Growth**.
3. In the **Expression Definition** pane, create and validate the following expression:

   ([Y2 Revenue]-[Y1 Revenue])/[Y1 Revenue]

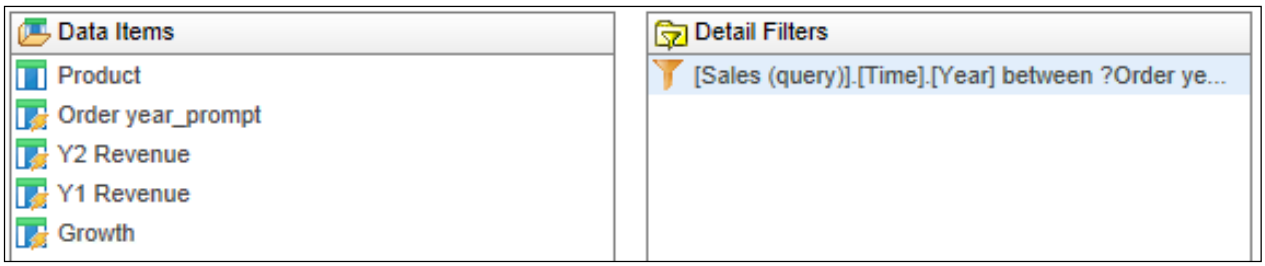
   Hint: Use the Data items tab for the data items in the expression.

   This data item uses the aggregated Revenue totals calculated by the query to determine the percentage change in total revenue between the year specified in the prompt and the previous year.
4. Click **OK** to close the **Data item expression** dialog box.
5. From the **Data/Source** tab, from the **Time** query subject, drag **Year** to the **Detail Filters** pane, and then create the following expression:

   [Sales (query)].[Time].[Year] between ?Order year?-1 and ?Order year?

   This filter is not necessary for this report to function properly, but it makes the query more efficient. By having the filter, the IF-THEN-ELSE statements on Y1 Revenue and Y2 Revenue will not have to evaluate all four years in the database. Instead, it will only have to evaluate one year.
6. Click **Validate**, and then click **OK**.

   The work area appears as follows:



Task 4. Build the report layout.

1. On the **Navigate** tab, click the **Page explorer** tab, and then click **Page1**.
2. On the **Data/Data items** tab, click **Product**, Ctrl+click **Y2 Revenue** and **Growth**, and then drag the selected items to the list.
3. In the list data container, click the **<Product>** list column body, from the On demand toolbar click **Sort**, and then click **Ascending**.
4. On the **Application** bar, open **Properties**.
5. Click the **<Growth>** list column body, and then in the **Properties** pane, under the **DATA** section, double-click the **Data format** property.

   The Data format dialog box appears.
6. In the **Format type** list, select **Percent**, and then under **Properties**, set **Number of Decimal Places** to **2**.
7. Scroll down, change the **Missing value characters** property to ***** no prior data *****, and then click **OK** to close the **Data format** dialog box.
8. Click the **<Y2 Revenue>** list column body, and then in the **Properties** pane, under the **DATA** section, double-click the **Data format** property.
9. In the **Format type** list, select **Currency**, and then in the **Properties** pane, change the **Currency** property to **$ (USD) - United States of America, dollar**.
10. Click **OK**, and then run the report in **HTML**.
11. In the **Year** prompt box, type **2012**, and then click **OK**.

    A section of the result appears as follows:

Product	Y2 Revenue	Growth
Aloe Relief	$101,812.41	-43.77%
Astro Pilot	$13,072,475.00	139.40%
Auto Pilot	$3,388,230.00	*** no prior data ***
Bear Edge	$3,960,860.01	46.57%
Bear Survival Edge	$3,320,876.50	59.34%

    Since the report displays data for only the year chosen in the prompt, you will add a dynamic column header and report title that will always reflect the data being displayed.
12. Close the rendered report tab.

Task 5. Create a dynamic column header.

1. In the list data container, click the **Y2 Revenue** column header, and then in the **Properties** pane, under the **TEXT SOURCE** section, change the **Source type** property to **Report expression**.
2. In the **Properties** pane, under the **TEXT SOURCE** section, double-click the **Report expression** property, and create and validate the following expression:

   ParamDisplayValue('Order year')

   Hint: Drag Order year for the Parameters tab, and then validate with 2012.
3. Click **OK** to close the **Report Expression** dialog box.

   In the Properties pane, the TEXT SOURCE section appears as follows:

TEXT SOURCE	
Source type	Report expression
Report expression	ParamDisplayValue('Order year')

4. On the **Application** bar, under **More**, click **Locked** to unlock the report.
5. From the **Toolbox** tab, drag a **Text item** to the end of the **Y2 Revenue** column header, press the spacebar, type **Revenue**, and then click **OK**.
6. On the **Application** bar, under **More**, click **Unlocked** to lock the report
7. Run the report in **HTML**. In the **Year** prompt box type **2012**, and then click **OK**.

   A section of the report appears as follows:

Product	2012 Revenue	Growth
Aloe Relief	$101,812.41	-43.77%
Astro Pilot	$13,072,475.00	139.40%
Auto Pilot	$3,388,230.00	*** no prior data ***

   The title for the revenue column now reflects the data in the report. You will now create a dynamic title for the report that also references the selected order year.
8. Close the rendered report tab.

Task 6. Create a dynamic report title.

1. Double-click the report title text, type **Summary of**, press the spacebar, and then click **OK** to close the **Text** dialog box.
2. From the **Toolbox** tab, under **TEXTUAL**, drag a **Layout calculation** object to the end of the report title.
3. In the **Available Components** pane, on the **Parameters** tab, drag the **Order year** parameter to the **Expression Definition** pane.

   The expression appears as follows:

   ParamDisplayValue('Order year')
4. Click **OK** to close the **Report expression** dialog box.
5. From the **Toolbox** tab, drag a **Text item** object to the end of the report title, press the spacebar, type **Product Sales and Growth**, and then click **OK**.
6. In the title, click **Summary of**, and then on the toolbar, click **Pick up Style**.
7. Click the layout calculation, and then from the On demand toolbar, click **Apply Style.**
8. Click **Product Sales and Growth**, and then from the On demand toolbar, click **Apply Style**.
9. Click the title text block to select it, and then on the toolbar, click **Left**.
10. Run the report in **HTML**, when prompted for a year, type **2012**, and then click **OK**.

    A section of the report appears as follows:

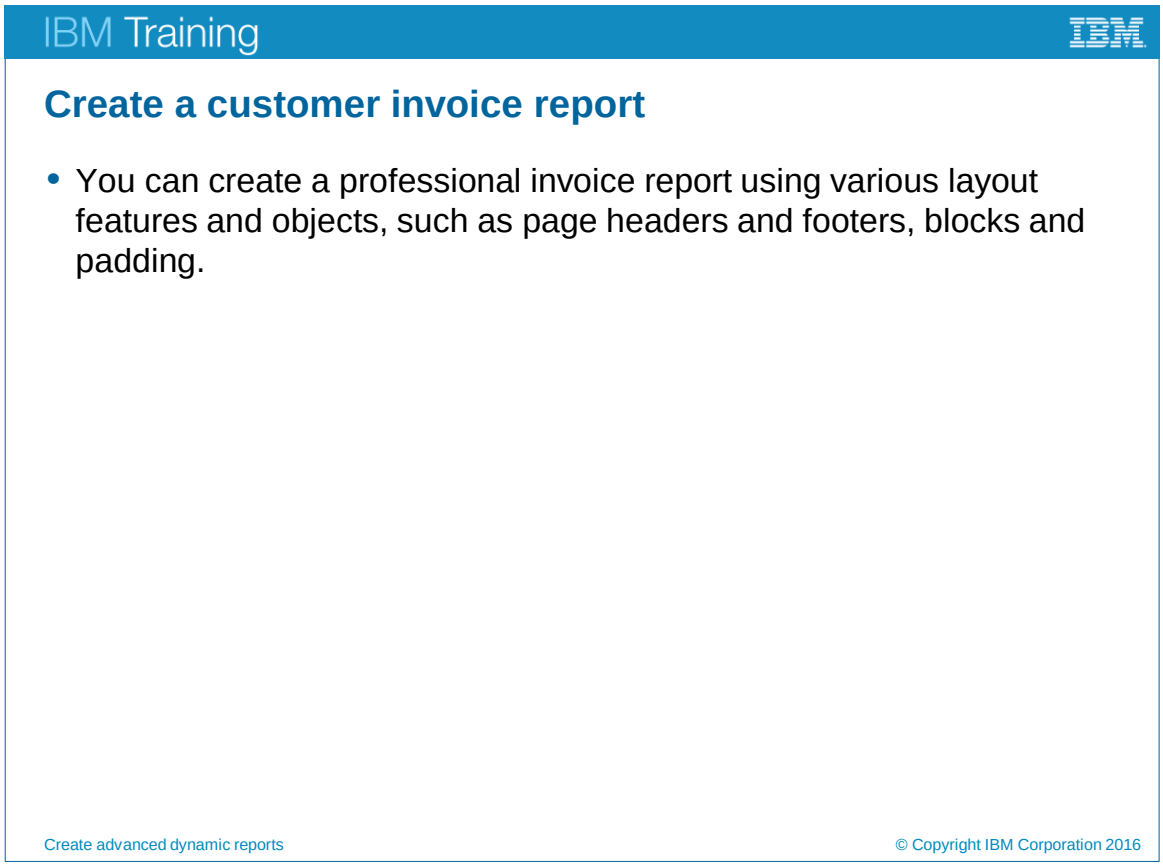
Summary of 2012 Product Sales and Growth

Product	2012 Revenue	Growth
Aloe Relief	$101,812.41	-43.77%
Astro Pilot	$13,072,475.00	139.40%
Auto Pilot	$3,388,230.00	*** no prior data ***
Bear Edge	$3,960,860.01	46.57%
Bear Survival Edge	$3,320,876.50	59.34%

Now both the revenue column header and the report title reflect the data in the report, based on the year specified in the prompt.

11. Close the rendered report tab.
12. Remove the report without saving it.
13. Leave **IBM Cognos Analytics** portal open for the exercise.

Results:
You created a revenue and growth report that uses calculated data items to retrieve data based on the year selected in the prompt. You then formatted a column title and the report title using data items that incorporated the order year specified by the user.



Create a customer invoice report

The important part of laying out a complex report is performing the translation of a desired layout to the layout objects (for example, blocks or lists) that IBM Cognos Analytics - Reporting provides for you.

Plan the layout objects according to page header, body, and footer. Think about horizontal bands and vertical bands of information. The process is iterative and will be refined over time.

IBM Training

Unit summary

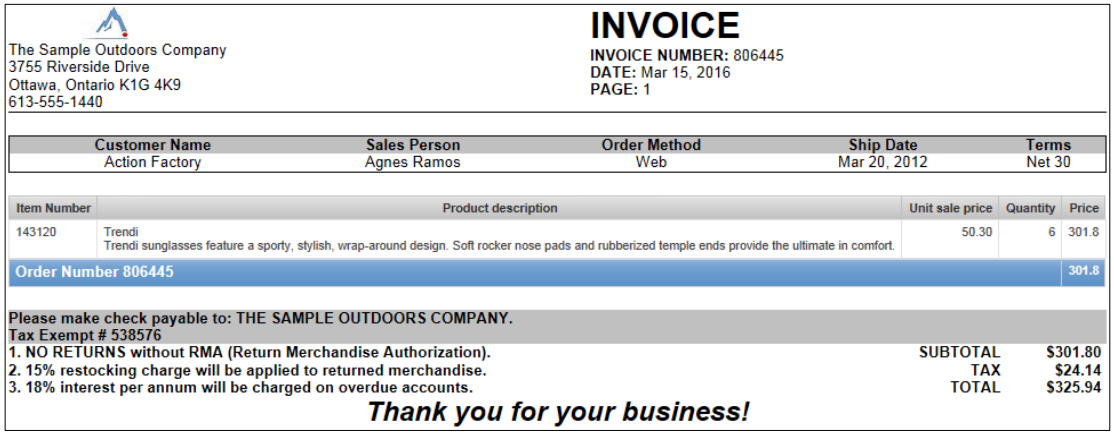
- Filter reports on session parameter values
- Navigate a briefing book using a table of contents
- Create dynamic headers and titles that reflect report data
- Create a customer invoice report

Unit summary

IBM Training

Exercise 1

Create a customer invoice report (Optional)



The Sample Outdoors Company
3755 Riverside Drive
Ottawa, Ontario K1G 4K9
613-555-1440

INVOICE
INVOICE NUMBER: 806445
DATE: Mar 15, 2016
PAGE: 1

Customer Name	Sales Person	Order Method	Ship Date	Terms
Action Factory	Agnes Ramos	Web	Mar 20, 2012	Net 30

Item Number	Product description	Unit sale price	Quantity	Price
143120	Trendi Trendi sunglasses feature a sporty, stylish, wrap-around design. Soft rocker nose pads and rubberized temple ends provide the ultimate in comfort.	50.30	6	301.8
Order Number 806445				301.8

Please make check payable to: THE SAMPLE OUTDOORS COMPANY.
Tax Exempt # 538576
1. NO RETURNS without RMA (Return Merchandise Authorization).
2. 15% restocking charge will be applied to returned merchandise.
3. 18% interest per annum will be charged on overdue accounts.

SUBTOTAL $301.80
TAX $24.14
TOTAL $325.94

Thank you for your business!

Create advanced dynamic reports

Exercise 1: Create a customer invoice report (Optional)

Exercise 1: Create a customer invoice report (Optional)

As a report author for The Sample Outdoors Company, you have been asked to create a standardized invoice that can be sent to customers whenever an order is processed. You will use the query items from the GO data warehouse (query) package\Sales and Marketing (query) folder\Sales (query) namespace.

To build this report you will:

- start with a blank report, and build a query, named OrderNumber, based on the Order number data item
- create a prompt page with two value prompts; one to select the customer (Retailer name from Retailers) and one to select sales representative (Employee name from Employee by region)
- Rename the queries as follows:
  - Query 1: Retailer Name Query
  - Query2: Sales Rep query
- specify that the data for each order should be on a new page by defining a page set associated with the OrderNumber query, grouping by order number, and defining a detail page associated with the OrderNumber query
- create a page header with the company logo (logo.jpg) and address:
  - The Sample Outdoors Company
    3755 Riverside Drive
    Ottawa, Ontario K1G 4K9
    613-555-1440
- add invoice information to the header, for example: Invoice number, Date, and Page
- create the top portion of the page body with general information about the order, for example: Customer Name, Sales Person, Order Method, Ship Date and Terms
- add a list report to the page body with the following columns: Order number, Product number, Product, Description, Unit Sale Price, Quantity and Revenue
- combine the product and description into a single column and change the title of the product number column to Item Number
- using a block and a table, add the order number group footer, tax calculation, and closing information

For more information about where to work and the workshop results, refer to the Tasks and results section that follows. If you need more information to complete a task, refer to earlier demonstrations for detailed steps.

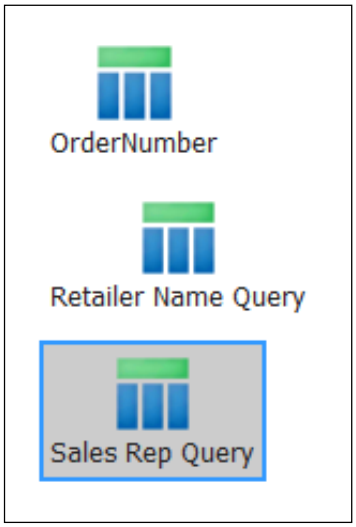
Exercise 1: Tasks and results

Task 1. Add data and prompts to the query.

- **IBM Cognos Analytics portal**: Open a new **Blank** report template, using the **GO data warehouse (query)** package.
- **Navigate/Query explorer** tab**:** Click **Queries**.
- **Toolbox tab**: Add a **Query** object to the work area.
- **Properties pane**: Change the new query's **Name** property to **OrderNumber**.
- **Work area**: Open the **OrderNumber** query.
- **Data/Source tab**: Navigate to **Sales and Marketing (query)/Sales (query)/ Sales order**.
  - Add the **Order number** query item to the **Data Items** pane.
- **Navigate** tab: Navigate to the **Page explorer** tab, **Prompt pages**.
- **Toolbox** tab: Add a **Page** object to the **Prompt pages** work area.
- **Work area**: Open **Prompt page1**.
- **Toolbox** tab: Under **PROMPTING**, add a **Value prompt** to the work area.
- **Prompt Wizard**: Create a new parameter named **Retailer name**, click **Next**.
  - For the Package item, add **Retailer name** from **Sales and Marketing (query)/Sales (query)/Retailers**.
  - Verify the values in the **Populate control** pane, and then click **Finish**.
- **Toolbox** tab: Add another **Value prompt** to the work area.
- **Prompt Wizard**: Create a new parameter named **SalesRep**, based on the following query item: **Sales and Marketing (query)/ Sales (query)/Employee by region/Employee name**.
  - Verify that the **OrderNumber** query is selected in the **Apply filter** pane, and then click **Next**.
  - Verify the values in the **Populate control** pane, and then click **Finish**.
- **Navigate** tab**:** Click the **Query explorer** tab, and then click **Queries**.

- **Properties pane**: Rename the queries.
  - **Query1** to **Retailer Name Query**.
  - **Query2** to **Sales Rep Query**.

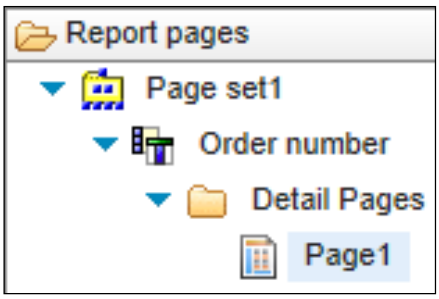
  The results appear as follows:



Task 2. Specify that data for each order will begin on a new page.

- **Navigate** tab: Click **Page explorer**.
  - Click **Report pages**.
- **Toolbox** tab: Drag a **Page set** object to the **Report pages** pane.
- **Properties** pane: Change the **Query** property for **Page set1** to **OrderNumber**.
  - From the **Grouping & sorting** property, add **Order number** to the **Groups** folder.
- **Report pages** pane: Drag **Page1** onto the **Detail Pages** folder.
- **Properties pane**: Change the **Query** property for **Page 1** to **OrderNumber**.

  The results appear as follows:



Task 3. Create a page header with the company logo and address.

- **Navigate** tab: Click **Page1**.
- **Work area**: Add a **Table** object onto the work area using the default settings.
  - Add an additional **Table** object into the left table cell with **1** column and **5** rows.
- **Toolbox** tab: Under **LAYOUT**, drag an **Image** object into the top table cell.
- **Image object**: Double-click the image, and then click **Browse**.

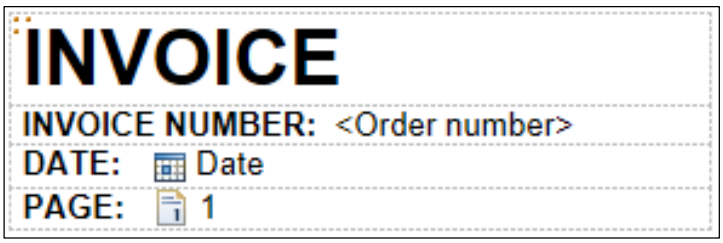
  (Set the image **Server URL** to **http://vclassbase:88/images** if necessary)
  - Double-click **logo.jpg.**
- **Toolbox tab**: Add a **Text item** object to the table cell below the logo table cell.
- **Text box**: Type **The Sample Outdoors Company**.
- **Toolbox tab**: Add a **Text Item** object to the table cell below the previous table cell.
- **Text box**: Type **3755 Riverside Drive**.
- **Toolbox tab**: Add a **Text Item** object to the table cell below the previous table cell.
- **Text box**: Type **Ottawa, Ontario K1G 4K9**.
- **Toolbox tab**: Add a **Text Item** object to the table cell below the previous table cell.
- **Text box**: Type **613-555-1440**.

  The results appear as follows:

Task 4. Add invoice information to the header.

- **Toolbox tab**: Add another **Table** object into the right table cell with **1** column and **4** rows.
- **Toolbox tab**: Add a **Text Item** object to the top table cell of the table that you have just added.
- **Text box**: Type **INVOICE**.
- **Toolbar**: Format the **INVOICE** text as **Arial**, **24 pt**, **Bold**.
- **Toolbox tab**: Add a **Text item** object to the table cell below the previous table cell.
- **Text box**: Type: **INVOICE NUMBER:** and then press the spacebar.
- **Toolbar**: Format the **INVOICE NUMBER:** text as **Arial**, **10 pt**, **Bold**.
- **Toolbox tab**: Add a **Text item** object to the table cell below the previous table cell.
- **Text box**: Type **DATE:** and then press the spacebar.
- **Toolbar**: Format the **DATE:** text as **Arial**, **10 pt**, **Bold**.
- **Toolbox tab**: Add a **Text item** object to the table cell below the previous table cell.
- **Text box**: Type **PAGE:** and then press the spacebar.
- **Toolbar**: Format the **PAGE:** text as **Arial**, **10 pt**, **Bold**.
- **Toolbox tab**: Under **TEXTUAL**, add a **Date** object to the right of the **DATE:** text item.
  - Add a **Page number** object to the right of the **PAGE:** text item.
- **Data/Data items** tab**:** Add the **Order number** data item to the right of the **INVOICE NUMBER:** text item.

  The results appear as follows:

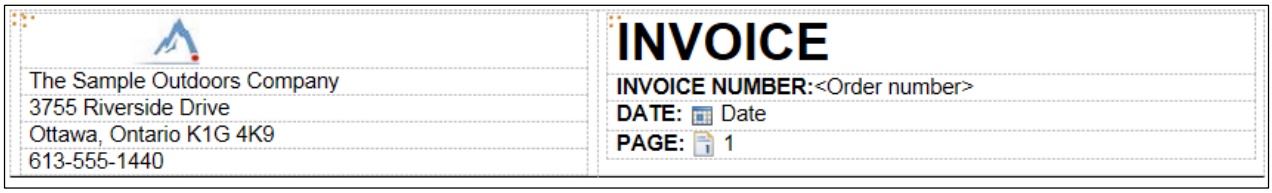


Task 5. Add a border and padding to the page header.

- **Properties pane**: Changer the **Border** property for the outer table as follows:

  Solid line, **1 pt**, and then click **Apply Bottom Border**  .
- **Work area**: Click the white space directly under the **PAGE:** cell to select the outer table cell.
- **Properties pane**: Change the **Bottom padding** property to **10**.

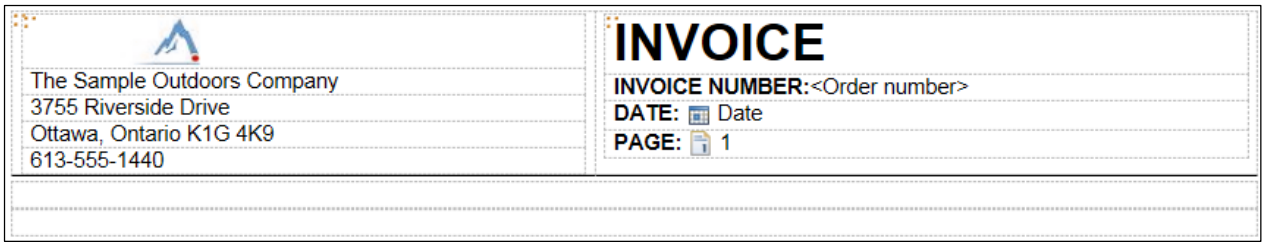
  The results appear as follows:



Task 6. Create the top portion of the page body.

- **Toolbox tab**: Add two **Block** objects to the body of the report, under the page header.

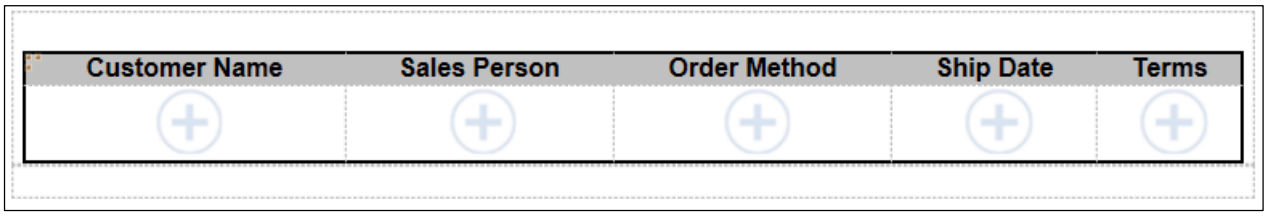
  The results appear as follows:



  - Add a **Table** into the first (upper) empty block with **5** columns and **2** rows.
- **Properties pane**: For the new table block, change the **Padding** property to **20px** for the **Top padding**.
- **Toolbox tab**: Create the following headers in the top row of cells:
  Customer Name, **Sales Person**, **Order Method**, **Ship Date**, and **Terms**.
- **Properties pane**: **Apply All Borders**, with **1 pt**, **solid line** border to entire table.

- **Table**: Ctrl+click each cell and center them.
  - For the top table row, set the font **Weight** to **Bold**, and the **Background Color** to **Silver**.

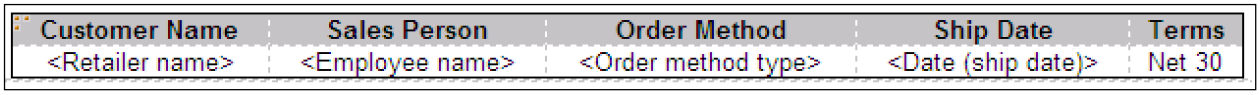
  The results appear as follows:



Task 7. Add data to the page body.

- **Data/Source** tab: Populate the first four cells of the second row of table cells as follows: **Retailer name**, **Employee name**, **Order method type**, **Date (ship date)**.
- **Toolbox tab**: Add the following text item to the **Terms** column, **Net 30**.

  The results appear as follows:



Customer Name	Sales Person	Order Method	Ship Date	Terms
<Retailer name>	<Employee name>	<Order method type>	<Date (ship date)>	Net 30

Task 8. Add details to the report and format the list.

- **Properties pane**: For the empty block, change the **Padding** property to **20px** of **Top padding**.
- **Toolbox** tab: Add a **List** data container to the empty block.
- **Properties pane**: For the **List** data container, change the **Query** property to **OrderNumber**.
- **Data/Data items** tab: Add **Order number** to the list report object.
- **Data/Source** tab: Navigate to **Sales and Marketing (query)/Sales (query)/Products/Codes**.
  - add the **Product number** query item to the list report object:
- **Source** tab: Navigate to **Sales and Marketing (query)/Sales (query)/Products**.
  - Add the **Product** and **Product description** query items to the list report object.

- **Source** tab: Navigate to **Sales and Marketing (query)/Sales (query)/Sales fact**.
  - Add **Unit sale price**, **Quantity**, and **Revenue** facts to the list report object.

  A section of the results appear as follows:

Order number	Product number	Product	Product description	Unit sale price	Quantity	Revenue
<Order number>	<Product number>	<Product>	<Product description>	<Unit sale price>	<Quantity>	<Revenue>

- **Toolbar**: Group the **<Order number>** list column body.
- **Properties** pane: Click the **List columns title ttyle** from the **Select Ancestor** list for any column header in the current list report object.
- **Toolbar**: Set the **Background Color** to **Silver**,
  - Change the font **Weight** to **Bold**.

  A section of the result appears as follows:

Order number	Product number	Product	Product description	Unit sale price	Quantity	Revenue
<Order number>	<Product number>	<Product>	<Product description>	<Unit sale price>	<Quantity>	<Revenue>
<Order number>	<Product number>	<Product>	<Product description>	<Unit sale price>	<Quantity>	<Revenue>

Task 9. Combine columns and change column titles.

- **Application** bar: Unlock the report.
- **Toolbox** tab: Add two **Block** objects to the left of **<Product description>** (not the heading).

  A section of the result appears as follows:

- **List data container**: Drag the **<Product>** data item from the **Product** column into the first empty block in the **Product description** column.
  - Drag the **<Product description>** data item from the **Product description** column into the second empty block.
- **Application** bar: Lock the report.
  - Delete the empty **Product** column.

- **Properties** pane: For the **Product number** list column title (header cell), change the **Source type** property to **Text**.
  - Change the **Text** property to **Item Number**.
  - For the **Revenue** list column title, change the **Source type** property to **Text**.
  - Change the **Text** property to **Price**.
- **Toolbar**: Summarize the **<Revenue>** list column body by **Total**.
  - Clear the **Overall footer** check box from **List headers & footers**.
- **Properties** pane: For the list data container, under the **DATA** section, double-click the **Properties** option, and then select the **Order number** properties check box.
- **On demand toolbar**: Cut the **<Order number>** list column body.

  The results appear as follows:

Item Number	Product description	Unit sale price	Quantity	Price
<Product number>	<Product> <Product description>	<Unit sale price>	<Quantity>	<Revenue>
<Order number> - Total				<Total(Revenue)>
<Product number>	<Product> <Product description>	<Unit sale price>	<Quantity>	<Revenue>
<Order number> - Total				<Total(Revenue)>

Task 10. Add text and data to the report.

- **Toolbox** tab: Drag a **Block** object below the list data container, within the existing block.
  - Drag a **Table** object into the new block with **3** columns and **3** rows.
- **Properties** pane: With the new block selected, add **Top** padding of **20px**.
- **Table**: Add the following text to the each row of the first column:
  - **1. NO RETURNS without RMA (Return Merchandise Authorization).**
  - **2. 15% restocking charge will be applied to returned merchandise.**
  - **3. 18% interest per annum will be charged on overdue accounts.**

- **Table**: Add the following text to each row of the second column:
  - **SUBTOTAL**
  - **TAX**
  - **TOTAL**
- **Table**: Right-justify the **SUBTOTAL**, **TAX**, and **TOTAL** cells.
- **Properties** pane: Ctrl+click all of the text items that you have just added.
  - Change the font to **Arial**, **10pt**, and **Bold**.
- **On demand toolbar**: With the first cell of the first row of the table selected, under **More**, click **Insert**.
  - Click **Rows above**.
  - Change the number of rows to **2** and then click **OK**.
- **On demand toolbar**: With each of the new first row cells selected, click **Merge Cells**.
  - Merge all of the cells in the second row.
- **Toolbox** tab: Add the following Text items to the new rows:
  - **Please make check payable to: THE SAMPLE OUTDOORS COMPANY.**
  - **Tax Exempt # 5386576**
- **On demand toolbar**: Ctrl+click the two text items that you have just added.
  - Change the font to **Arial**, **10pt**, and **Bold**.
  - Set the background color of each row to **Silver**.
- **On demand toolbar**: With the first cell of the last row of the table selected, under **More**, click **Insert**.
  - Click **Rows below**, and then click **OK** to accept the default value.
- **On demand toolbar**: With each of the new row cells selected, click **Merge Cells**.
- **Toolbox** tab: Add the following **Text** Item to the new row:
  - **Thank you for your business!**

- **Toolbar**: Format the new text, **Arial**, **18 pt**, **Bold**, **Italic**, and **Centered**.

  The results appear as follows:

Please make check payable to: THE SAMPLE OUTDOORS COMPANY. Tax Exempt # 5386576		
1. NO RETURNS without RMA (Return Merchandise Authorization).	SUBTOTAL	
2. 15% restocking charge will be applied to returned merchandise.	TAX	
3. 18% interest per annum will be charged on overdue accounts.	TOTAL	
Thank you for your business!		

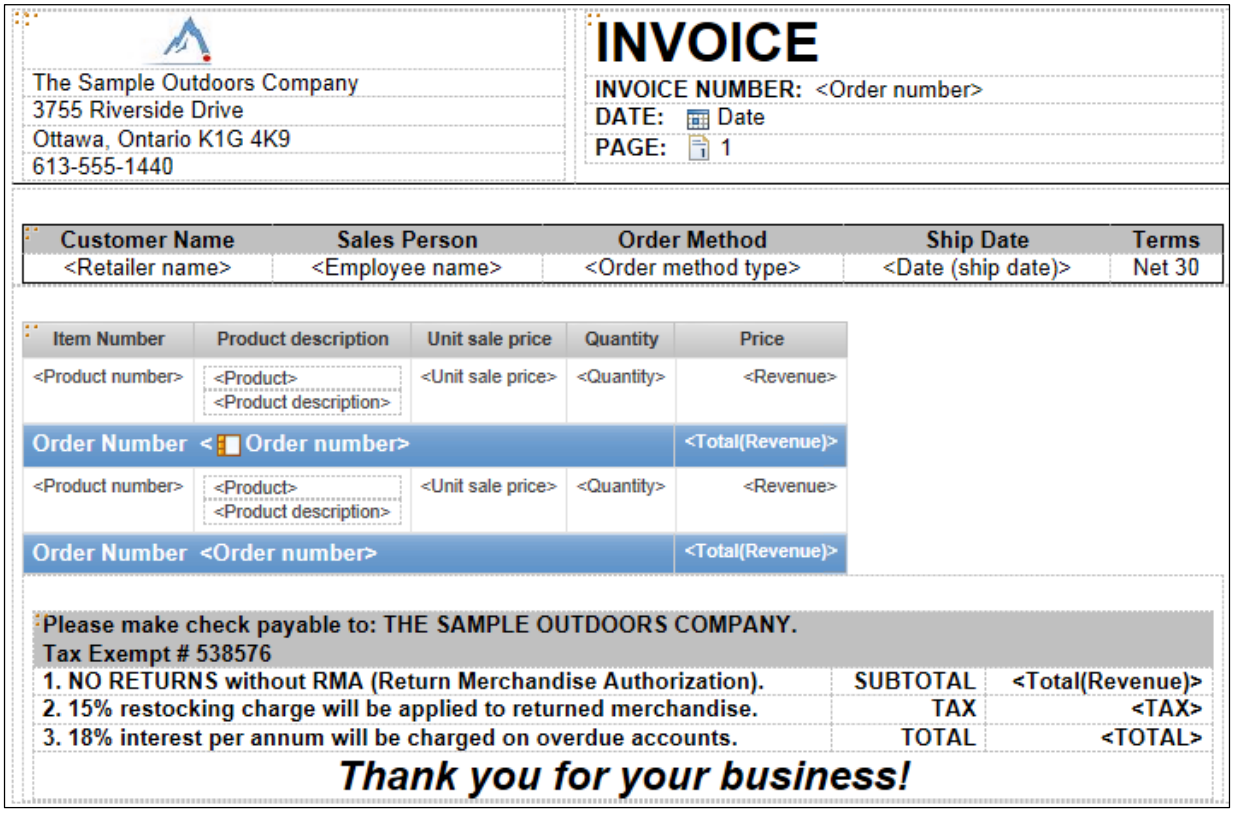
Task 11. Add additional data, and tax and total calculations.

- **Data/Data items**: Drag **Total(Revenue)** to the empty cell to the right of **SUBTOTAL**.
- **Toolbox tab**: Under **TEXTUAL**, create and validate the following **Query calculation** expression, named **TAX**, for the empty cell to the right of the **TAX** cell: **[Sales (query)].[Sales fact].[Revenue]*.08**
  - Validate with **Action Factory** and **Agnes Ramos**, if required.
  - Create and validate the following **Query Calculation** expression, named **TOTAL**, for the empty cell to the right of the **TOTAL** cell: **[Total(Revenue)]+[TAX]**
- **Toolbar**: Set the **<Total(Revenue)>**, **<TAX>**, and **<TOTAL>** calculations font **Weight** to **Bold**, and then right justify.

Task 12. Add additional formatting, and then run the report.

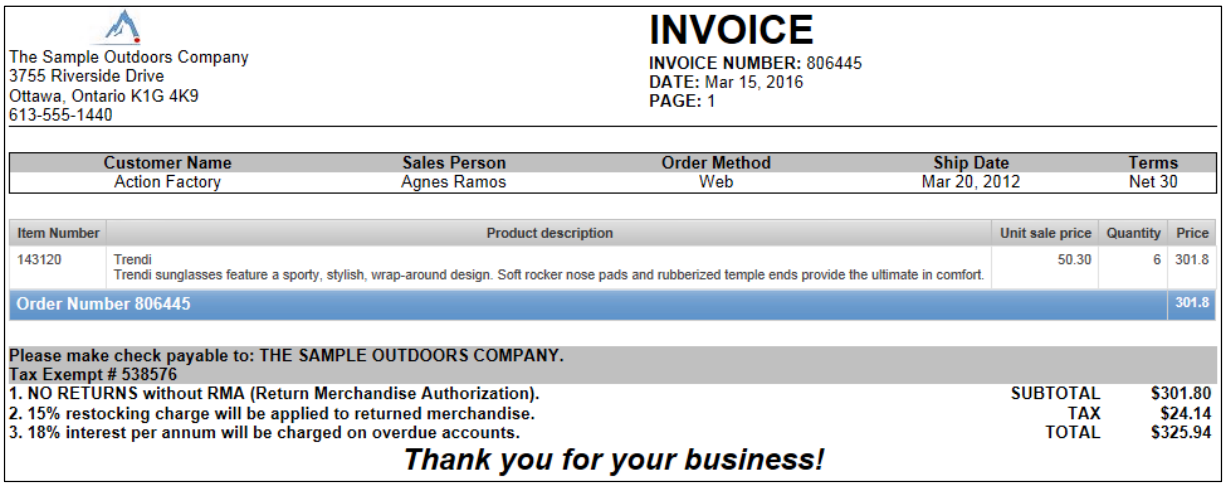
- **Properties** pane: With the **<Total(Revenue)>**, **<TAX>**, and **<TOTAL>** calculations selected, change the **Data format** property to **Currency**, **$ (USD) - United States of America, dollar**, with **2** decimal places.
- **Application** bar: Unlock the report.
  - Delete the "**-** "and "**Total"** text items in the same summary row cell to the right of **<Order number>**.
- **Toolbox** tab: Drag a **Text item** object to the left of **<Order number>**.
  - Type **Order Number**, press the spacebar, and then click **OK**.
- **On demand toolbar**: Ctrl+click both the **Order Number** text item and the **<Order Number>** data item, and then set the format to **Arial**, **10pt**. and **Bold**.
- **Application** bar: Lock the report.

  The results appear as follows:



- Run the report in **HTML**, using **Action Factory** for the **Retailer name** prompt, and **Agnes Ramos** for the **Employee name** prompt.

  The results appear as follows:



The Sample Outdoors Company
3755 Riverside Drive
Ottawa, Ontario K1G 4K9
613-555-1440

INVOICE

INVOICE NUMBER: 806445
DATE: Mar 15, 2016
PAGE: 1

Customer Name	Sales Person	Order Method	Ship Date	Terms
Action Factory	Agnes Ramos	Web	Mar 20, 2012	Net 30

Item Number	Product description	Unit sale price	Quantity	Price
143120	Trendi Trendi sunglasses feature a sporty, stylish, wrap-around design. Soft rocker nose pads and rubberized temple ends provide the ultimate in comfort.	50.30	6	301.8
Order Number 806445				301.8

Please make check payable to: THE SAMPLE OUTDOORS COMPANY.
Tax Exempt # 538576

1. NO RETURNS without RMA (Return Merchandise Authorization).
2. 15% restocking charge will be applied to returned merchandise.
3. 18% interest per annum will be charged on overdue accounts.

SUBTOTAL	$301.80
TAX	$24.14
TOTAL	$325.94

Thank you for your business!

  Spacing may differ due to screen resolution.

  As a report author for The Sample Outdoors Company, you have created a standardized invoice that can be sent to customers whenever an order is processed.

- Close the rendered report tab.
- Sign out of **IBM Cognos Analytics - Reporting**.
- Close the **IBM Cognos Analytics** portal.

IBM Training

Design effective prompts

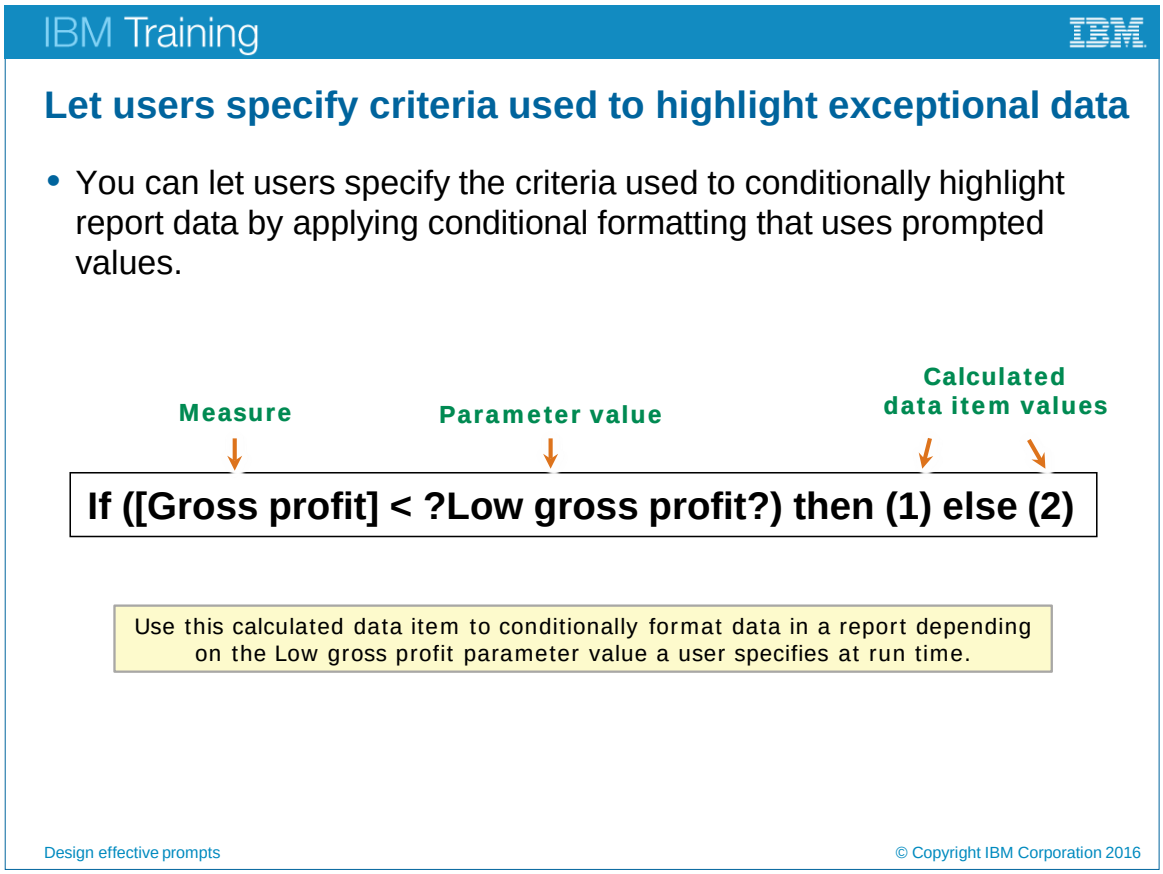
IBM Cognos Analytics (v11.0)

IBM Training

Unit objectives

- Control report displays using prompts
- Specify conditional formatting values using prompts
- Specify conditional rendering of objects based on prompt selection
- Create sorted and filtered reports based on prompt selection

Unit objectives



Let users specify criteria used to highlight exceptional data

By setting up conditional formatting using prompted values, you can create a report that does not need to be modified when company benchmark values change.

If you do not want the calculated data item to appear in the report, you can either:

- add the calculated data item to the report to test conditional formatting, and then cut the calculated data item from the report layout, or
- add the calculated data item to the query in Query explorer but not to the report layout

When applying conditional formatting, you cannot use parameters directly in a variable expression, but you can reference data items that use parameters.

IBM Training

Demonstration 1

Create a prompt that lets users select conditional formatting values

Country	Product line	Product type	Revenue	Gross profit	LowMedHigh
Australia	Camping Equipment	Cooking Gear	7,464,381.59	2,938,400.28	2
		Lanterns	3,416,036.14	1,495,901.15	2
		Packs	9,063,575.38	3,590,975.57	3
		Sleeping Bags	8,027,884.16	3,217,019.11	3
		Tents	13,964,054.92	4,481,882.33	3
	Golf Equipment	Golf Accessories	1,318,528.37	805,825.44	2
		Irons	6,839,374.25	3,270,992.55	3
		Putters	2,401,092.95	1,150,708.83	2
		Woods	8,520,560.86	4,181,044.42	3
	Mountaineering Equipment	Climbing Accessories	2,646,717.89	1,356,587.33	2
		Rope	4,039,366.36	1,274,262.53	2
		Safety	2,637,772.5	983,979.46	2
		Tools	4,609,263.31	1,930,313.75	2
	Outdoor Protection	First Aid	169,911.38	91,347.2	1
		Insect Repellents	559,867.46	371,574.87	1
		Sunscreen	410,564.72	245,984.02	1
	Personal Accessories	Binoculars	2,214,888.28	756,112.83	2
		Eyewear	2,132,100.87	1,130,134.9	2
		Knives	2,760,632.22	893,852.34	2
		Navigation	3,118,074.7	1,230,319.6	2

Design effective prompts

Demonstration 1: Create a prompt that lets users select conditional formatting values

Demonstration 1: Create a prompt that lets users select conditional formatting values

Purpose:
Management has requested a report that lets them identify the total revenue and gross profit generated in every country by each product type. They want to be able to customize this report by highlighting high and low gross profit values based on changing benchmark levels. You will create a report that formats gross profit to appear in different colors if gross profit is exceptionally low or exceptionally high. This formatting will be based on high and low gross profit values users select using prompts at run time.

Portal:	**http://vclassbase:9300/bi**
User/Password:	**brettonf/Education1**
Application:	**IBM Cognos Analytics - Reporting**
Package:	**Team content\Samples\Models\GO data warehouse (query)**
Report Type:	**List**
Folder:	**Sales and Marketing (query)**
Namespace:	**Sales (query)**

Task 1. Create a report with product sales information.

1. Open a new **List** template using the **GO data warehouse (query)** package.
2. From the **Data/Source** tab, add the following query items to the new list data container.
   - Employee by region: **Country**
   - Products: **Product line**, **Product type**
   - Sales fact: **Revenue**, **Gross profit**

Country	Product line	Product type	Revenue	Gross profit
<Country>	<Product line>	<Product type>	<Revenue>	<Gross profit>
<Country>	<Product line>	<Product type>	<Revenue>	<Gross profit>
<Country>	<Product line>	<Product type>	<Revenue>	<Gross profit>

3. In the list data container, click the **<Country>** list column body, Ctrl+click the **<Product line>** list column body, and then on the toolbar click **Group / Ungroup**.
4. Click the **<Product type>** list column body, on the toolbar click **Sort**, and then click **Ascending**.

Task 2. Add a calculated data item with an expression that uses parameter values, and then run the report.

You will add a calculated data item that compares the gross profit values to those selected in two prompts: Low gross profit and High gross profit. This calculated data item will then assign a value: 1 for gross profit less than the Low gross profit value, 2 for gross profit values greater than the Low gross profit value but less than the High gross profit value, and 3 for gross profit greater than the High gross profit value.

1. From the **Toolbox** tab, expand **TEXTUAL**, and then drag a **Query calculation** object to the end of the list.
2. In the **Data item expression** dialog box, in the **Name** field, type **LowMedHigh**.
3. Create and validate the following expression:

   if ([Gross profit] < ?Low gross profit?) then (1) else if ([Gross profit] < ?High gross profit?) then (2) else (3)

   Hint: Data Items tab: Gross profit.

   Validate using:

   - Low gross profit: **600000** (600,000)
   - High gross profit: **3000000** (3 million)

   This calculation retrieves the prompt values the user specifies at run time and assigns a value of 1, 2, or 3 to each row depending on their gross profit values.
4. Click **OK** to close any open dialog box.
5. Run the report in **HTML**.
6. When prompted for values, ensure that **Low gross profit** is **600000**, **High gross profit** is **3000000**, and then click **OK**.

   The LowMedHigh column displays a value for each row based on the values that you supplied using the prompts. Product types generating below $600,000 have a value of 1, product types generating between $600,000 and $3 million have a value of 2, and product types generating over $3 million in gross profit have a value of 3.

   A section of the result appears as follows:

Country	Product line	Product type	Revenue	Gross profit	LowMedHigh
Australia	Camping Equipment	Cooking Gear	7,464,381.59	2,938,400.28	2
		Lanterns	3,416,036.14	1,495,901.15	2
		Packs	9,063,575.38	3,590,975.57	3
		Sleeping Bags	8,027,884.16	3,217,019.11	3
		Tents	13,964,054.92	4,481,882.33	3

7. Close the rendered report tab.

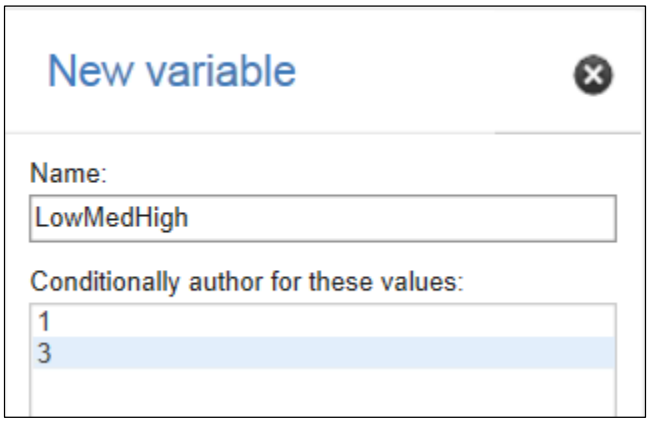
Task 3. Create a variable based on the calculation.

You will create a string variable that you can use to conditionally format the Gross profit column of the report. You will use the LowMedHigh query calculation to specify the conditions for this variable.

1. On the **Application** bar, open the **Properties** pane.
2. In the list data container, click the **<Gross profit>** list column body, in the **Properties** pane, under the **CONDITIONAL** section, double-click the **Style variable** property, and then in the **Variable** list, select **<New string variable>**.
3. In the **Name** box, type **LowMedHigh**, and then click **Add**.

   Since you want to apply conditional formatting only to high and low gross profit values, you will add two values (1 and 3) for which you will be able to apply conditional formatting.
4. In the **Add** box, type **1**, and then click **OK**.
5. Repeat the previous step to add **3**, and then click **OK**.

   A section of the result appears as follows:



6. Click **OK** to close the **New variable** dialog box.
7. In the **Report expression** dialog box, on the **Queries** tab, drag the **LowMedHigh** item to the **Expression Definition** pane.
8. Click **Validate**, and then after the expression is validated without any errors, click **OK** to close any open dialog box.

Task 4. Apply conditional formatting based on the LowMedHigh variable, and then run the report.

1. With the **<Gross profit>** list column body still selected, on the **Navigate** tab, click the **Condition explorer**  tab, and then click **1**.

   You want low gross profit values to have a red background, white foreground, and bold weight font.
2. From the On demand toolbar, under **More**, point to **Style**, and then click **Background color**.
3. Click **Red**, and then click **OK**.
4. On the **Conditional explorer** pane, verify that **1** is still selected.
5. From the On demand toolbar, under **More**, point to **Style**, and then click **Font**.
6. Click **Foreground color**, click **White**, and then click **OK**.
7. Under **Weight**, click **Bold**, and then click **OK**.

   You want high gross profit values to have a green background, of a different color tone value than the red, in case the report is printed on a non-color printer.
8. With the **<Gross profit>** list column body still selected, on the **Condition explorer**, click **3**.
9. From the On demand toolbar, under **More**, point to **Style**, and then click **Background color**.
10. Click the **Custom color** tab, type **77FF00** into the **#RGB** box, and then click **OK**.
11. On the **Conditional explorer** pane, verify that **3** is still selected.
12. On the **Properties** pane, under the **FONT & TEXT** section, double-click the **Font** property, under **Weight** click **Bold**, and then click **OK**.
13. On the **Conditional explorer** tab, click **LowMedHigh** to clear the conditional tool.

14. Run the report in **HTML**.
15. Ensure that **High gross profit** prompt value is **3000000**, **Low gross profit** prompt value is **600000**, and then click **OK**.

    The results appear as follows:

Country	Product line	Product type	Revenue	Gross profit	LowMedHigh
Australia	Camping Equipment	Cooking Gear	7,464,381.59	2,938,400.28	2
		Lanterns	3,416,036.14	1,495,901.15	2
		Packs	9,063,575.38	3,590,975.57	3
		Sleeping Bags	8,027,884.16	3,217,019.11	3
		Tents	13,964,054.92	4,481,882.33	3
	Golf Equipment	Golf Accessories	1,318,528.37	805,825.44	2
		Irons	6,839,374.25	3,270,992.55	3
		Putters	2,401,092.95	1,150,708.83	2
		Woods	8,520,560.86	4,181,044.42	3
	Mountaineering Equipment	Climbing Accessories	2,646,717.89	1,356,587.33	2
		Rope	4,039,366.36	1,274,262.53	2
		Safety	2,637,772.5	983,979.46	2
		Tools	4,609,263.31	1,930,313.75	2
	Outdoor Protection	First Aid	169,911.38	91,347.2	1
		Insect Repellents	559,867.46	371,574.87	1
		Sunscreen	410,564.72	245,984.02	1
	Personal Accessories	Binoculars	2,214,888.28	756,112.83	2
		Eyewear	2,132,100.87	1,130,134.9	2
		Knives	2,760,632.22	893,852.34	2
		Navigation	3,118,074.7	1,230,319.6	2

    In the Gross profit column, values of less than six-hundred thousand dollars are highlighted in red, while values of greater than three million dollars are highlighted in green.
16. Close the rendered report tab.

Task 5. Cut the calculated column, and then run the report.

You do not want to see the LowMedHigh column in the list report, as the conditional formatting has been applied in another column.

1. On the **Navigate** tab, click the **Page explorer** tab, and then click **Page1**.
2. In the list report, click the **<LowMedHigh>** column body, then from the On demand toolbar, under **More**, click **Cut**.

   To let the string variable use the LowMedHigh data item values for each row in the report, you need to make the LowMedHigh data item a property of the List object.
3. Click the list **Container Selector** to select the entire list.
4. In the **Properties** pane, under the **DATA** section, double-click the **Properties** option, click the **LowMedHigh** check box to select it, and then click **OK**.
5. Run the report in **HTML**.
6. Ensure that the **Low gross profit** prompt value is **600000**, the **High gross profit** prompt value is **3000000**, and then click **OK**.

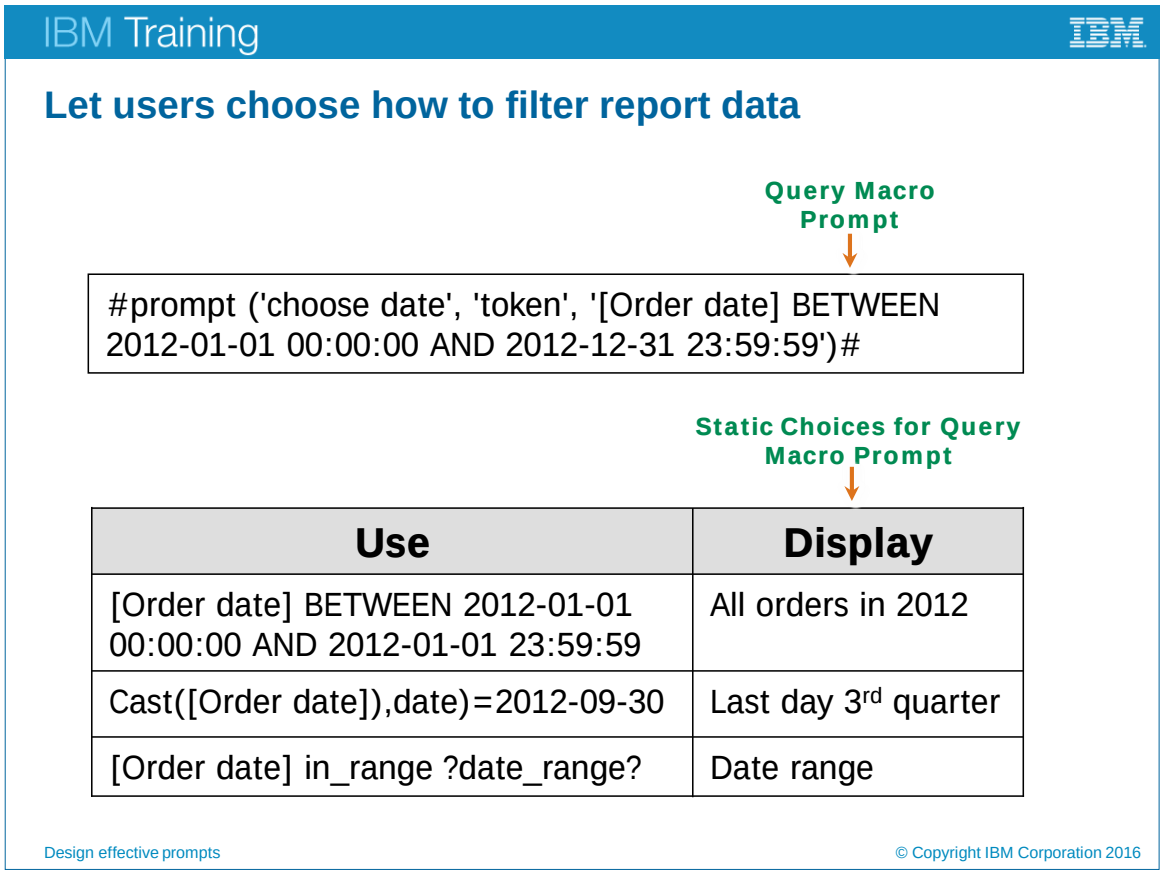
   A section of the result appears as follows:

Country	Product line	Product type	Revenue	Gross profit
Australia	Camping Equipment	Cooking Gear	7,464,381.59	2,938,400.28
		Lanterns	3,416,036.14	1,495,901.15
		Packs	9,063,575.38	3,590,975.57
		Sleeping Bags	8,027,884.16	3,217,019.11
		Tents	13,964,054.92	4,481,882.33
	Golf Equipment	Golf Accessories	1,318,528.37	805,825.44
		Irons	6,839,374.25	3,270,992.55
		Putters	2,401,092.95	1,150,708.83
		Woods	8,520,560.86	4,181,044.42
	Mountaineering Equipment	Climbing Accessories	2,646,717.89	1,356,587.33
		Rope	4,039,366.36	1,274,262.53
		Safety	2,637,772.5	983,979.46
		Tools	4,609,263.31	1,930,313.75
	Outdoor Protection	First Aid	169,911.38	91,347.2
		Insect Repellents	559,867.46	371,574.87
		Sunscreen	410,564.72	245,984.02
	Personal Accessories	Binoculars	2,214,888.28	756,112.83
		Eyewear	2,132,100.87	1,130,134.9
		Knives	2,760,632.22	893,852.34
		Navigation	3,118,074.7	1,230,319.6

The conditional formatting is now applied to the Gross profit column, even though the LowMedHigh column no longer appears in the report.

7. Close the rendered report tab.
8. Remove the report without saving it.
9. Leave the **IBM Cognos Analytics** portal open for the next demonstration.

Results:
You created a report that identifies the total revenue and gross profit generated in every country by each product type. You added prompts to let users customize this report by selecting high and low gross profit values to highlight data.



Let users choose how to filter report data

A macro is a fragment of code that you can insert into the Select statement of a query or into an expression. Query macro prompts return different values to a query depending on the prompt option users select.

In query macro prompts, everything between the # signs is replaced with the Use value returned when a user selects a prompt option. You can use query macro prompts when working with relational data or dimensionally-modeled relational data.

Static choices present the user with prompt options that are not found in the data source. Refer to *IBM Cognos Business Intelligence Version 11.0.0 IBM Cognos Analytics - Reporting User Guide*: *Using Query Macros*, for more information on using query macro prompts.

It is important to know your database when creating the Use values. In the Let users choose how to filter report data slide example, the second static choice casts Order date as a 'date' data type because in the database, the Order date item is stored as a Date & Time data type. If you do not cast it as 'date', you must specify the exact date and time of the order in the static choice.

The Name parameter, which is mandatory, specifies the name of the query macro prompt.

Be sure to use straight quotes, you may need to turn off Microsoft Word's Smart quotes.

IBM Training

Demonstration 2

Create a prompt to let users choose between different filters

Date	Order number	Retailer name	Revenue
May 8, 2012	103576	Great Adventures	185,785.19
May 8, 2012	103581	Todo para el Golf, S.A. de C.V.	318,307.36
May 8, 2012	103601	The Marketplace	237,845.26
May 8, 2012	103605	Tamarack Outfitter Rentals	3,859.1
May 8, 2012	103607	Edward's Department Store	27,064.78
May 8, 2012	103610	I-wear Direct	45,934.56
May 8, 2012	103619	Nature Voyageurs	61,421.94
May 8, 2012	103625	The Sport Pros	254,232.43
May 8, 2012	103627	Extreme Outdoors	361,564.54

Demonstration 2: Create a prompt to let users choose between different filters

Demonstration 2: Create a prompt to let users choose between different filters

Purpose:
You have been asked to create a report showing the order date, order number, and revenue for sales to all retailers. The report should let users choose entire year of 2012, a specific date of high retail activity in 2012, or any date range of interest to the user. To provide users with these options, you will add a query macro prompt so users can choose which filter to apply to the report.

Portal: **http://vclassbase:9300/bi**
User/Password: **brettonf/Education1**
Application: **IBM Cognos Analytics - Reporting**
Package: **Team content\Samples\Models\GO data warehouse (query)**
Report Type: **List**
Folder: **Sales and Marketing (query)**
Namespace: **Sales (query)**

Task 1. Add items and a filter containing a query macro prompt.

1. Open a new **List** template using **GO data warehouse (query)** package.
2. From the **Data/Source** tab, add the following query items to the new list data container:
   - Time: **Date**
   - Sales order: **Order number**
   - Retailers: **Retailer name**
   - Sales fact: **Revenue**

Date	Order number	Retailer name	Revenue
<Date>	<Order number>	<Retailer name>	<Revenue>

3. In the list data container, click the **<Date>** list column body, on the toolbar click **Sort**, and then click **Ascending**.
4. On the **Navigate** tab, click **Query explorer**, and then click **Query1**.
5. On the **Application** tab, open the **Properties** pane.
6. In the **Properties** pane, under the **MISCELLANEOUS** section, change the **Name** property to **DateQuery**.

7. From the **Toolbox** tab, drag a **Filter** object to the **Detail Filters** pane.

   You will create a query macro prompt that includes the name of the parameter (choose date), the data type (token), and a default prompt value ([Order date] between 2012-01-01 00:00:00 and 2012-12-31 23:59:59).

   You can add parameter maps, session parameters, and macro functions to expressions. In the expression editor, to add a macro, on the Macros tab double-click the parameter maps, session parameters, or macro functions that you want to add to the macro expression.

   You can use the Insert macro block to get you started with creating the macro expression. Insert macro block inserts number signs in the expression. Ensure that the macro expression you create is between the number signs.
8. In the **Expression Definition** pane, create and validate the following expression:

   #prompt ('choose date', 'token', '[Date] between 2012-01-01 00:00:00 and 2012-12-31 23:59:59')#

   Hints:

   - Macros tab: Insert macro block, Macro Functions folder\prompt.
   - Data Items tab: Date.
9. Click **OK** to close the **Detail filter expression** dialog box.

   You will create a prompt page with a value prompt that uses the 'choose date' parameter you just created. You will create static choices for the prompt where the Use values are the SQL statements that are passed to the parameter.

Task 2. Create a prompt with static choices.

1. On the **Navigate** tab, click the **Page explorer** tab, click **Prompt pages**, and then from the **Toolbox** tab, drag a **Page** object to the **Prompt pages** pane.
2. Double-click **Prompt page1**, from the **Toolbox** tab, expand **PROMPTING**, and then drag a **Value prompt** object to the work area.
3. In the **Prompt Wizard**, click **Use existing parameter**, from the list select **choose date**, and then click **Finish**.
4. On the prompt page layout, click the **Value prompt** object to select it, and then in the **Properties** pane, under the **DATA** section, double-click the **Static choices** property.
5. In the **Static choices** dialog box, click **Add**.
6. In the **Use** box type **[Date] between 2012-01-01 00:00:00 and 2012-12-31 23:59:59**, in the **Display** box type **All Orders in 2012**, and then click **OK**.

7. Repeat steps **5** and **6** to add the following two static choices for the prompt:

Use	Display
Cast([Date],date)=2012-12-15	**December 15, 2012**
[Date] in_range ?date_range?	**Date Range**

In the second option, you cast [Date] as a 'date' data type because in the GO data warehouse database, the Date information is stored as 'date and time'. By converting this to 'date', you will retrieve every order made on December 15th, 2012 - regardless of what time the order was placed.

Use	Display
[Date] between 2012-01-01...	All Orders in 2012
Cast([Date],date)=2012-12-15	December 15 2012
[Date] in_range ?date_range?	Date Range

8. Click **OK** to close the **Static choices** dialog box.

Task 3. Add a Date & time prompt and test the prompt.

You will add a Date & time prompt object to the prompt page, so that users can enter specific dates if they choose the Date range option in the value prompt.

1. From the **Toolbox** tab, drag a **Date & time prompt** object to the right of the **Value prompt** object in the work area.
2. In the **Create a new parameter** box, type **date_range**, and then click **Finish**.
3. On the prompt page, click the **Date & time prompt** object to select it, and then in the **Properties** pane, under the **GENERAL** section, change the **Range** property to **Yes**.

   You will test the prompt to see whether the default date range you specified for this prompt (between 2012-01-01 00:00:00 and 2012-12-31 23:59:59) is used if you do not select a prompt value.
4. Run the report in **HTML**.
5. On the prompt page, in the **choose date** prompt list, select **All Orders in 2012**, and then click **Finish**.

   The report runs. The first dates displayed are early January 2012.
6. In the report, click **Bottom**.

   The last dates contained in the report are near the end of December 2012. The query prompt macro is using the default prompt value that you specified.

   You will now test the prompt by selecting one of the other prompt options.
7. From the rendered report tab, click **Run Report** to run the report again.

8. In the value prompt list, click **December 15, 2012**, and then click **Finish**.

   The report runs and appears on the rendered report tab. The report displays data for December 15, 2012, as expected.

   You will test the prompt by selecting a different prompt option.
9. On the rendered report tab, click **Run Report**, to run the report again.
10. In the **choose date** prompt list, click **Date Range**.
11. In the **From** prompt, select **May 1, 2012**, and then in the **To** prompt, select **May 31, 2012**.
12. Click **Finish**.

    You may get a second prompting for a date range. This is expected.

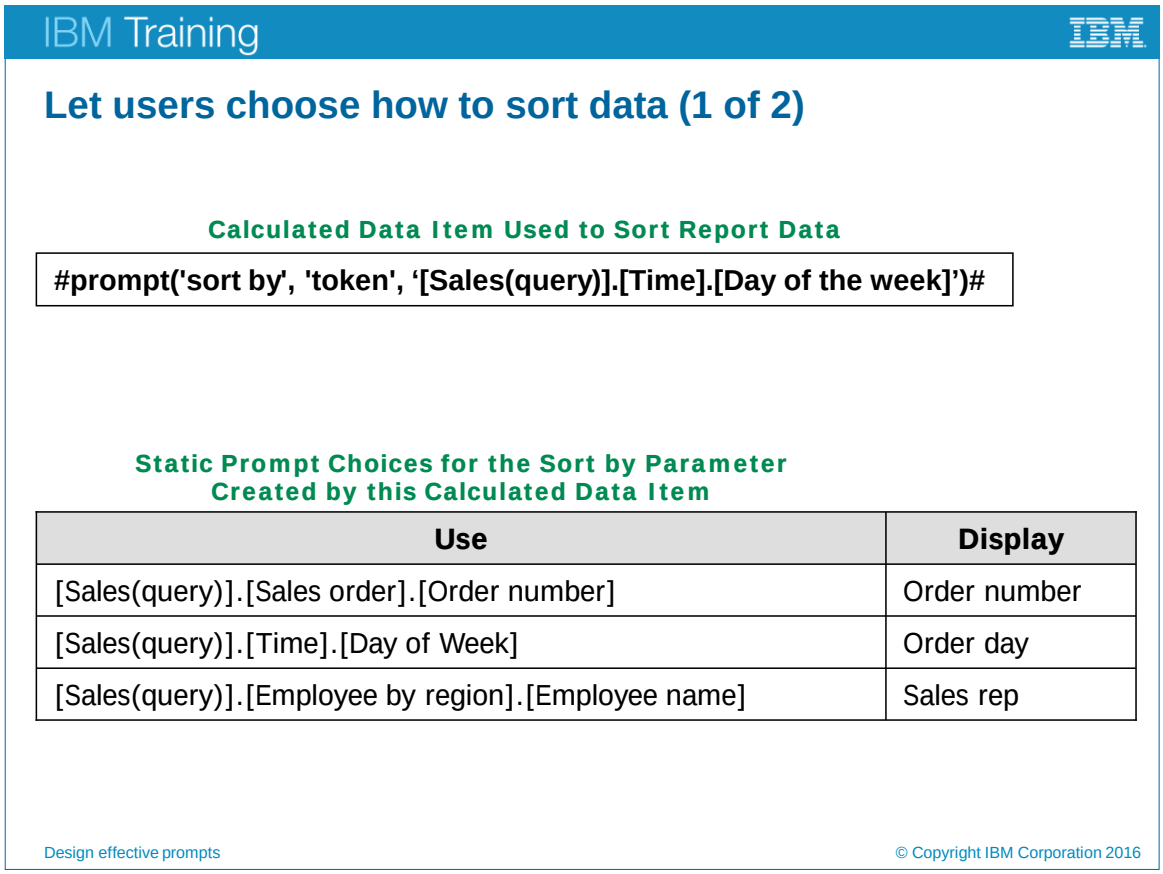
    A section of the results appear similar to the following:

Date	Order number	Retailer name	Revenue
May 8, 2012	103576	Great Adventures	185,785.19
May 8, 2012	103581	Todo para el Golf, S.A. de C.V.	318,307.36
May 8, 2012	103601	The Marketplace	237,845.26

    The beginning of the report contains data for early in May 2012.
    (The rows may not be in the same order as shown)
13. On the report, click **Bottom**.

    The end of the report displays data for the end of May 2012.
14. Close the rendered report tab.
15. Remove the report without saving.
16. Leave the **IBM Cognos Analytics** portal open for the next demonstration.

Results:
You created a report displaying the order date, order number, and revenue for sales to all retailers. You used a query macro prompt to let users choose to view orders for the entire year of 2012, for December 15th, 2012, or for any date range of interest to the user.

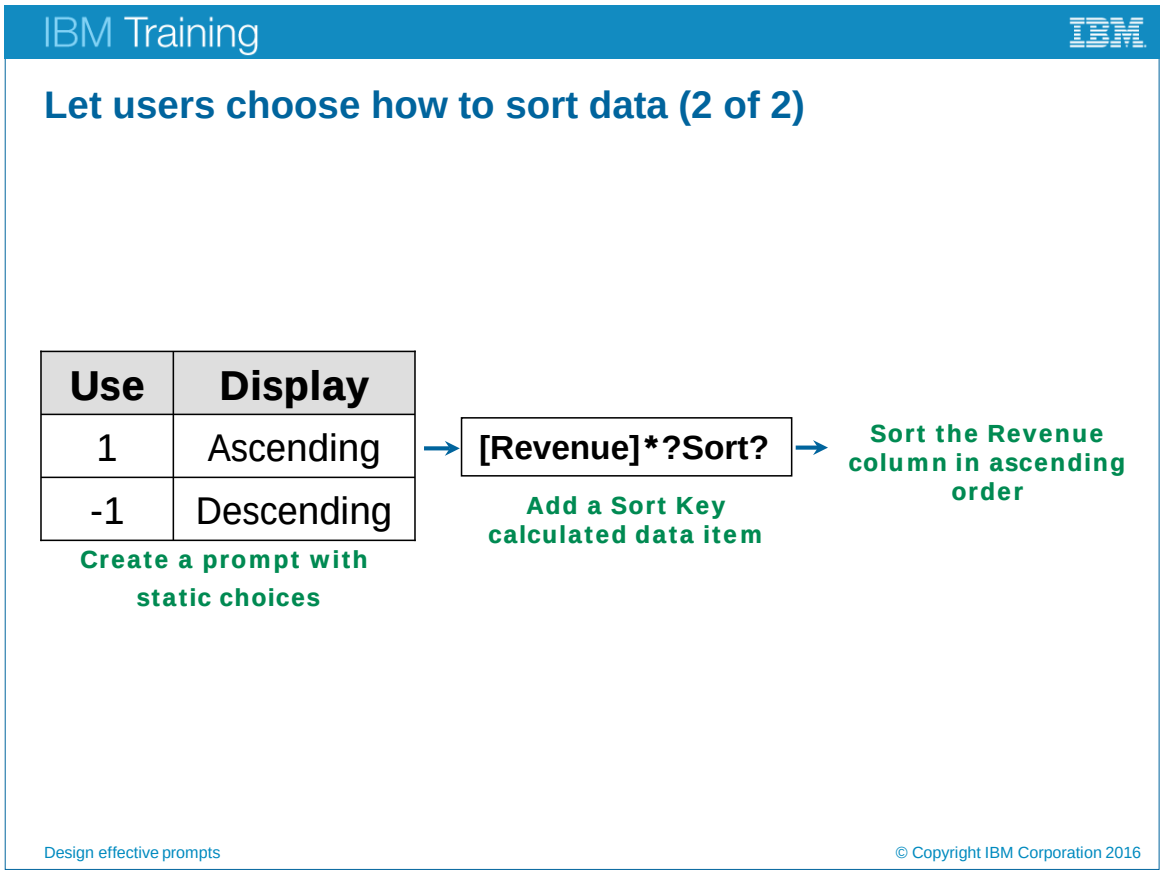


Let users choose how to sort data

You can create prompts that let users choose which data item to use to sort a report. If you want to let users choose how to sort data, ensure none of the columns are sorted in the report layout.

You can use the fully qualified path name of the data item as the Use Value property in the prompt, or if the item is associated with the report, you can use an alias.

If users can select only one data item to sort data with, create one calculated data item. For multiple data items, create a calculated data item for each option.



To create a prompt that lets users choose whether numerical values are sorted in ascending or descending order, add a prompt to the report with two static choices: 1 and -1.

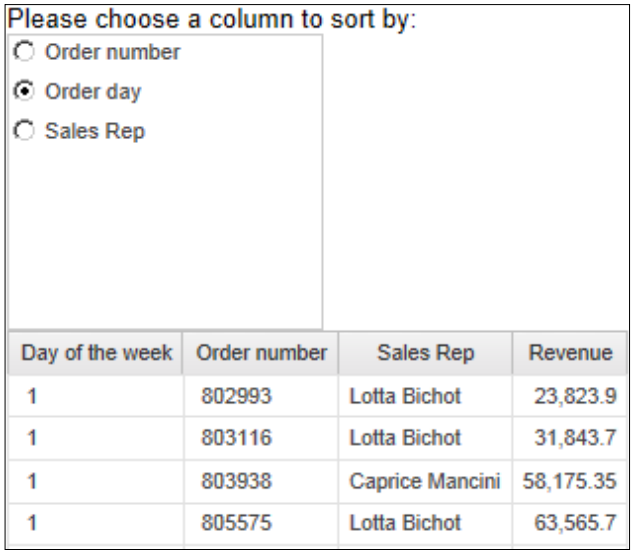
Add a calculated data item that multiplies the numerical values in the column you want to sort by either 1 or -1, depending on the prompt option users select.

Sort this calculated data item in ascending order.

IBM Training

Demonstration 3

Create a prompt to let users choose how to sort list columns



Design effective prompts

Demonstration 3: Create a prompt to let users choose how to sort list columns

Demonstration 3: Create a prompt to let users choose how to sort list columns

Purpose:
A manager wants to view information about individual sales orders. The manager wants to easily locate sales based on either order number, order date, or the sales representative who made the sale. You will create a list report with a prompt on the report page so that the manager can choose the column to sort the report with.

Portal: **http://vclassbase:9300/bi**
User/Password: **brettonf/Education1**
Application: **IBM Cognos Analytics - Reporting**
Package: **Team content\Samples\Models\GO data warehouse (query)**
Report Type: **List**
Folder: **Sales and Marketing (query)**
Namespace: **Sales (query)**

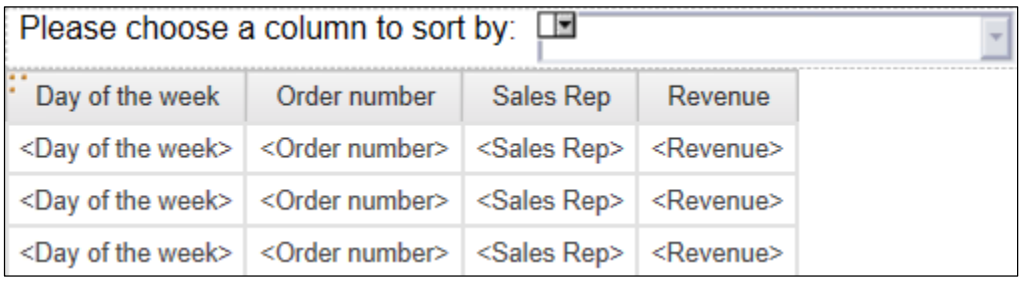
Task 1. Create a list with a prompt on the report page.

1. Open a new **List** template using the **GO data warehouse (query)** package.
2. From the **Data/Source** tab, add the following query items to the list report object:
   - Time: **Day of the week**
   - Sales order: **Order number**
   - Employee by region: **Employee name**
   - Sales fact: **Revenue**

Day of the week	Order number	Employee name	Revenue
<Day of the week>	<Order number>	<Employee name>	<Revenue>
<Day of the week>	<Order number>	<Employee name>	<Revenue>
<Day of the week>	<Order number>	<Employee name>	<Revenue>

3. On the **Application** bar, open the **Properties** pane.
4. In the list data container, click the **<Employee name>** list column body, in the **Properties** pane, under the **DATA ITEM** section, change both the **Name** property and the **Label** property to **Sales Rep**.
5. From the **Toolbox** tab, drag a **Block** object to the left of the list, to add a block object above the list object.
6. From the **Toolbox** tab, under **PROMPTING**, drag a **Value prompt** object into the block that you just added.
7. In the **Prompt Wizard**, in the **Create a new parameter** box, type **Sort by**, and then click **Finish**.
8. From the **Toolbox** tab, drag a **Text item** object to the left of the value prompt, within the block.
9. Type **Please choose a column to sort by:**, press the spacebar, and then click **OK**.

   A section of the report layout appears as follows:



Task 2. Add choices to the prompt.

You will add static choices for the Sort by prompt to let users choose to sort report data by order number, order date, or sales representative name.

1. In the work area, click the **Value prompt** object, and then in the **Properties** pane, under the **DATA** section, double-click the **Static choices** property.
2. In the **Static choices** dialog box, click **Add**, in the **Use** box type **[Order number]**, and then in the **Display** box type **Order number**.

   You could also have used the fully qualified path ([Sales (query)].[Sales order].[Order number]) to the employee name as the Use value property, however in this report, all of the columns are members of the report, so you can use the alias as the Use value property here.
3. Click **OK** to close the **Edit** dialog box.

4. Repeat steps **2** and **3** to create the following two options:

Use	Display
[Day of the week]	Order day
[Sales Rep]	Sales Rep

A section of the result appears as follows:

Use	Display
[Order number]	Order number
[Day of the week]	Order day
[Sales Rep]	Sales Rep

5. Click **OK** to close the **Static choices** dialog box.
6. With the Value prompt object still selected, in the **Properties** pane, under the **GENERAL** section, double-click the **Default selections** property.
7. Click the **Add** button, type **[Day of the week]**, and then click **OK** to close any open dialog boxes.

   You want to make this prompt optional.
8. With the Value prompt object still selected, set the following properties, under the **GENERAL** section:
   - Required: **No**
   - Auto-submit: **Yes**
   - Select UI: **Radio button group**

   Once a user selects a prompt option you want the prompt to automatically submit the parameter value.

   You will add a calculated data item to the query to retrieve and sort data based on the sort option that a user selects.

Task 3. Add a calculated data item to the query.

1. On the **Navigate** tab, click the **Query explorer** tab, and then click **Query1**.
2. In the **Properties** pane, under the **MISCELLANEOUS** section, rename **Query1** to **Sales Rep Order number Query**.
3. From the **Toolbox** tab, drag a **Data Item** object to the **Data Items** pane.
4. In the **Data item expression** dialog box, in the **Name** field, type **Sort Key**.

5. In the **Expression Definition** pane, create and validate the following expression:

   #prompt('Sort by', 'token', '[Day of the week]')#

   You could have used the fully qualified path for the data item, rather than the alias.

   Day of the week is the default column to sort on.

   Hints:

   - Macros tab: Report Parameters folder\Sort by
   - Data Items tab: Day of the week

6. Click **OK** to close the dialog box.

Task 4. Name and sort the calculated data item, add the data item as a property of the list, and then test the prompt.

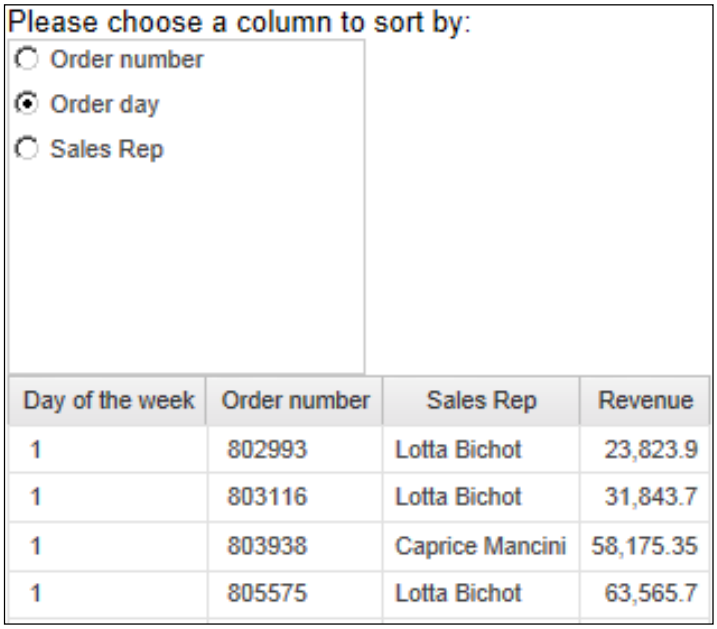
1. With the **Sort Key** data item selected, in the **Properties** pane, under the **DATA ITEM** section, change the **Pre-sort** property to **Sort ascending**.

   To use the Sort Key data item to determine how data is sorted in the report layout, you must make the Sort Key data item a property of the list object.

2. On the **Navigate** tab, click the **Page explorer** tab, and then click **Page1**.
3. Click the list **Container Selector** to select the entire list.
4. In the **Properties** pane, under the **DATA** section, double-click the **Properties** property, select the **Sort Key** check box, and then click **OK**.

5. Run the report in **HTML**.

   A section of the result appears as follows:



Day of the week	Order number	Sales Rep	Revenue
1	802993	Lotta Bichot	23,823.9
1	803116	Lotta Bichot	31,843.7
1	803938	Caprice Mancini	58,175.35
1	805575	Lotta Bichot	63,565.7

   The Day of the week column is sorted in ascending order because this is specified in the Sort Key expression as the default column on which to sort.

6. Click on the **Sales Rep** radio button, and examine the results; click the **Order number** radio button, and examine the results.

   The report is sorted by Sales Rep and Order number respectively.

7. Close the rendered report tab.
8. Remove the report without saving.
9. Leave the **IBM Cognos Analytics** portal open for the next demonstration.

Results:
You created a list report with a prompt on the report page that lets users choose the column that they want to use to sort the report.

IBM Training

Let users choose which objects appear in reports

Prompt

Display Options
Crosstab
Chart

Static Choices

Use	Display
Crosstab	Crosstab
Chart	Chart

Boolean variable applied to the conditional block:
?Display Options?='Crosstab'

When block value = 'Yes', drag Crosstab to conditional block

When block value = 'No', drag Chart to conditional block

Design effective prompts

Let users choose which objects appear in reports

Create a list of custom options that will appear at runtime by creating static choices for the prompt.

In this example, the user is prompted to choose Crosstab or Chart to view the report. Static choices represent a list of prompt options presented to the user. A conditional block is created with values that depend on the static choices the user selects in the prompt. Static Choices only apply to Search & Select and Value prompts.

When the report is being created, the block value is set to Yes and the crosstab object is dragged to the conditional block. The block value is then set to No and the chart object is dragged to the conditional block.

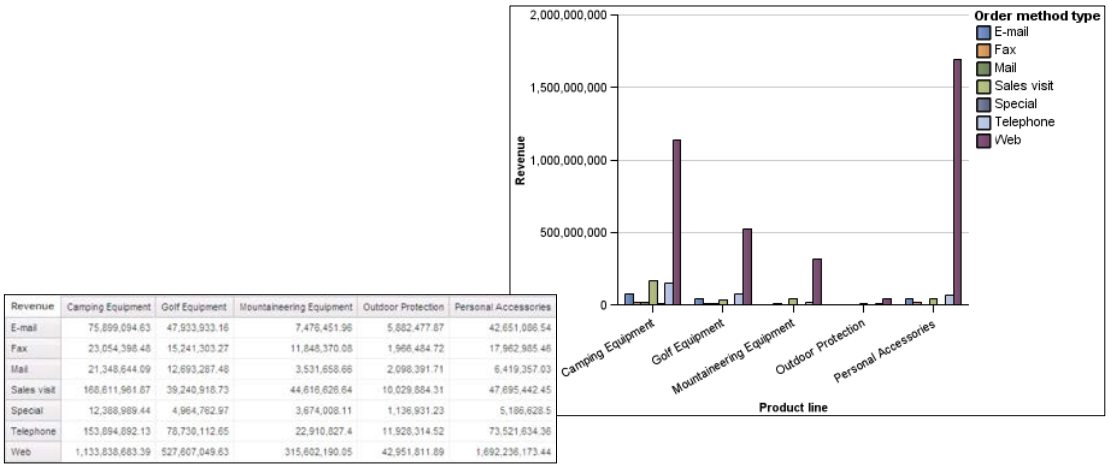
If you want to give users the option to select one of two choices, use a Boolean variable. If you want to give users more than two choices, use a string variable.

After creating a report with a conditional block, to view the objects contained in the conditional block for different values, view the page structure for the report page. To do this, from the Application bar, click Page views, and then click Page structure.

IBM Training

Demonstration 4

Create a prompt to let users select a display type



Revenue	Camping Equipment	Golf Equipment	Mountaineering Equipment	Outdoor Protection	Personal Accessories
E-mail	75,899,094.63	47,933,933.16	7,476,451.96	5,882,477.87	42,651,086.54
Fax	23,054,398.48	15,241,303.27	11,848,370.08	1,966,484.72	17,962,985.46
Mail	21,348,644.09	12,693,287.48	3,531,658.66	2,098,391.71	6,419,357.03
Sales visit	168,611,961.87	39,240,918.73	44,616,626.64	10,029,884.31	47,695,442.45
Special	12,388,989.44	4,964,762.97	3,674,008.11	1,136,931.23	5,186,628.5
Telephone	153,894,892.13	78,736,112.65	22,910,827.4	11,928,314.52	73,521,634.36
Web	1,133,838,683.39	527,607,049.63	315,602,190.05	42,951,811.89	1,692,236,173.44

Design effective prompts

Demonstration 4: Create a prompt to let users select a display type

Demonstration 4: Create a prompt to let users select a display type

Purpose:
The Marketing department has requested a report that shows the revenue generated for each product line by each order method. Some members of the department prefer a visual representation of the data, and some have requested the data in crosstab format. You will create a report that lets users choose the format that they prefer.

Portal:	**http://vclassbase:9300/bi**
User/Password:	**brettonf/Education1**
Application:	**IBM Cognos Analytics - Reporting**
Package:	**Team content\Samples\Models\GO data warehouse (query)**
Report Type:	**Crosstab**
Folder:	**Sales and Marketing (query)**
Namespace:	**Sales (query)**

Task 1. Add a crosstab and a chart to the report.

1. Open a new **Crosstab** template using the **GO data warehouse (query)** package.
2. From the **Data/Source** tab, add the following query items to the crosstab data container:
   - **Rows**:
     - Order method: **Order method type**
   - **Columns**:
     - Products: **Product line**
   - **Measures**:
     - Sales fact: **Revenue**

Revenue	<#Product line#>	<#Product line#>
<#Order method type#>	<#1234#>	<#1234#>
<#Order method type#>	<#1234#>	<#1234#>

3. From the **Toolbox** tab, drag a **Chart** object to the right of the crosstab, and then click the **Clustered Column** chart option.

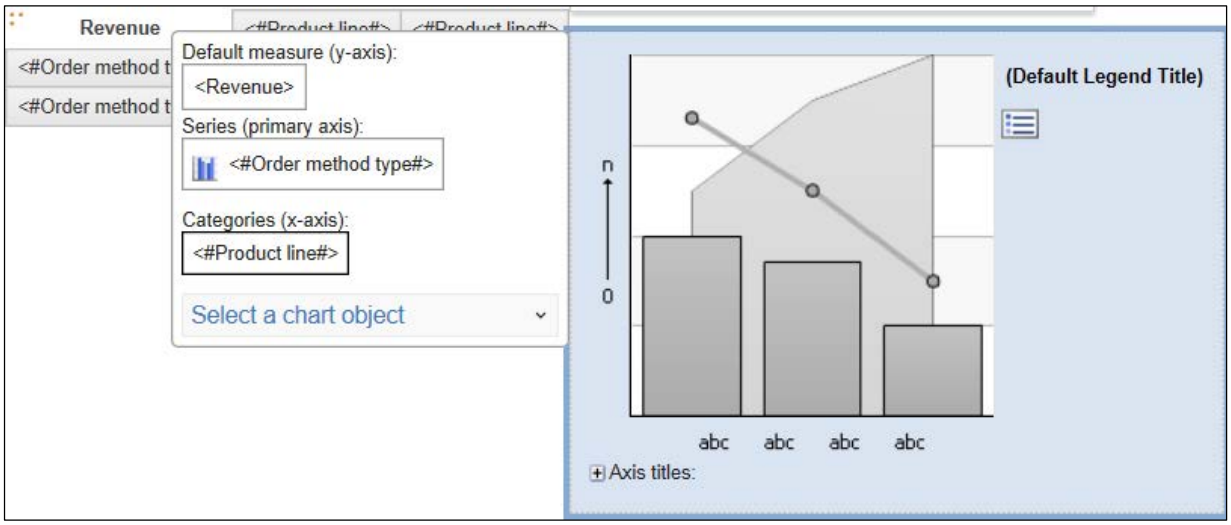
   This should be the first chart option displayed in the **Column** chart options. You want to add the same data that is in the crosstab to the chart.

4. With the **Insert chart** dialog box still open, click the **Fill with data from** radio button to select it, ensure that **Crosstab1** (the current object in the report layout) is selected in the list, and then click **OK**.

   You could also have manually added the chart, modified the chart object to use the Query1 association, and then added the data items from Query1 into the chart as follows:

   - **Default measure (y-axis)**: **Revenue**
   - **Series (primary axis)**: **Order method type**
   - **Categories (x-axis)**: **Product line**

   The results appear as follows:



Task 2. Add a prompt to the report and add static choices for the prompt.

At run time, you want users to choose either a crosstab or a chart, so you will create a value prompt that gives users two display options.

1. On the **Navigate** tab, click **Prompt pages**.
2. From the **Toolbox** tab, drag a **Page** object to the **Prompt pages** pane.
3. Double-click **Prompt page1**, and then from the **Toolbox** tab, under **PROMPTING**, drag a **Value prompt** object onto the prompt page.

   The Prompt Wizard opens and loads any parameters contained in the report. In this case, the report has no parameters. You will create a parameter for this prompt.
4. In the **Create a new parameter** box, type **Display Options**, and then click **Finish**.

   You will add two static choices for the prompt (Crosstab and Chart) to let users select a format in which to view data.

5. On the **Application** bar, open the **Properties** pane.
6. In the work area, click the value prompt object to select it, and then in the **Properties** pane, under the **DATA** section, double-click the **Static choices** property.
7. In the **Static choices** dialog box, click **Add**.
8. In the **Use** and **Display** boxes, type **Crosstab**, and then click **OK**.
9. Repeat steps **7** and **8** to add a second static choice, with **Chart** for the **Use** and **Display** values.
10. Click **OK** to close the **Static choices** dialog box.
11. On the **Properties** pane, under **GENERAL**, change **Auto-submit** to **Yes**.

Task 3. Add a conditional block and create a Boolean variable.

You will add a Conditional blocks object and apply conditional formatting to this object to determine whether it will contain the chart or the crosstab when the report is run.

1. On the **Navigate** tab, click **Page1**.
2. From the **Toolbox** tab, under **ADVANCED**, drag a **Conditional blocks** object to the right of the chart to add it below the chart.
3. Click the **Conditional blocks** object (that you just added) to select it, and then in the **Properties** pane, under the **CONDITIONAL** section, double-click the **Block variable** property.

   You will create a Boolean variable that uses the Display Options parameter so that the block displays differently depending on whether the user selects the Crosstab or the Chart prompt option.
4. In the **Variable** list, select **<New boolean variable>**, in the **Name** box type **DisplayVariable**, and then click **OK**.

   You will create an expression that specifies that when users select the Crosstab prompt option, the Yes condition from the Boolean variable is applied to the conditional block.
5. Create and validate the following expression:

   ParamDisplayValue('Display Options')='Crosstab'

   Hint: Parameters tab: Display Options.

6. Click **OK** to close any open dialog box.

   You can conditionally render many objects, including Blocks, Text Items, Lists, List columns, Crosstabs, Charts, Tables, Repeaters, Repeater Tables, Repeater Table Cells, and images. For a complete list, see the IBM Cognos Analytics - Reporting Object and Property Reference section of the *IBM Cognos Business Intelligence Version 11.0.0 IBM Cognos Analytics - Reporting User Guide*.

Task 4. Specify how the conditional block values will display.

1. Ensure that the **Conditional blocks** object is selected, in the **Properties** pane, under the **CONDITIONAL** section, change the **Current block** property to **Yes**.

   In task 3, you specified that when users select the Crosstab prompt option, the Yes value is applied to the Conditional blocks object. Therefore, with the Yes value selected, you will drag the Crosstab object to the Conditional block.
2. Click the crosstab **Container Selector** to select the entire crosstab.
3. Drag the crosstab to the Conditional block object.
4. Click the **Conditional blocks** object to select it, in the **Properties** pane, under the **CONDITIONAL** section, change the **Current block** property to **No**.

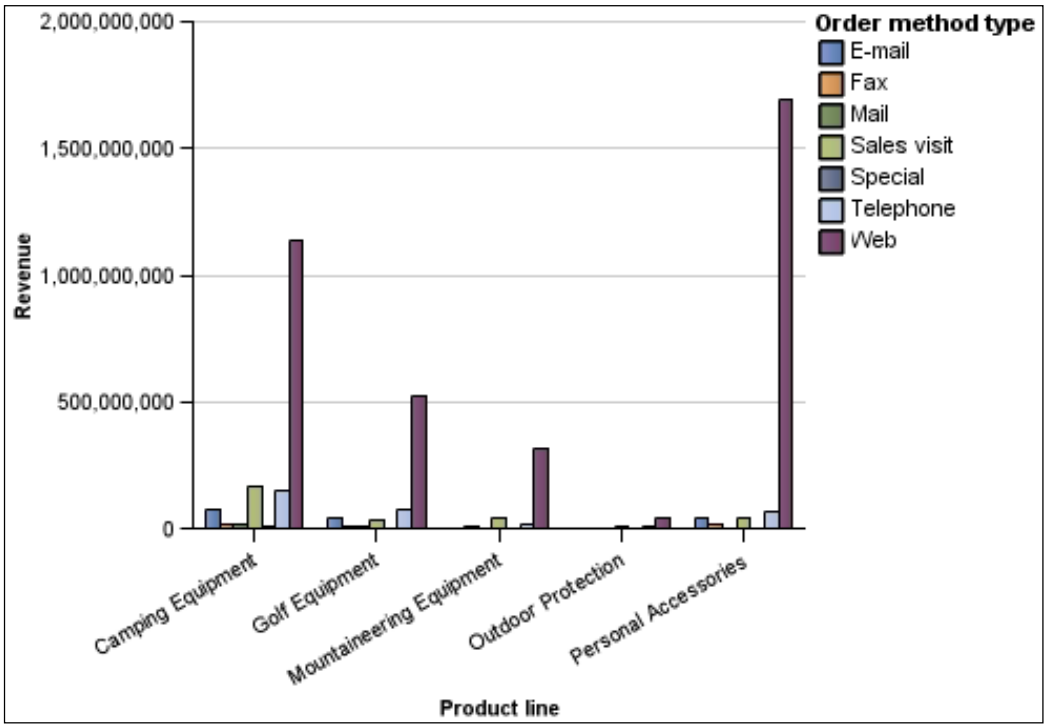
   In task 3, you specified that when users select the Chart prompt option, the No value is applied to the Conditional Block object. Therefore, with the No value selected, you will drag the Chart object to the Conditional Block.
5. Click the chart background to select the **Combination chart** object, and then drag the chart in to the **Conditional blocks** object.
6. Run the report in **HTML**.
7. In the **Display Options** prompt, select **Crosstab**.

   The results appear as follows:

Revenue	Camping Equipment	Golf Equipment	Mountaineering Equipment	Outdoor Protection	Personal Accessories
E-mail	75,899,094.63	47,933,933.16	7,476,451.96	5,882,477.87	42,651,086.54
Fax	23,054,398.48	15,241,303.27	11,848,370.08	1,966,484.72	17,962,985.46
Mail	21,348,644.09	12,693,287.48	3,531,658.66	2,098,391.71	6,419,357.03
Sales visit	168,611,961.87	39,240,918.73	44,616,626.64	10,029,884.31	47,695,442.45
Special	12,388,989.44	4,964,762.97	3,674,008.11	1,136,931.23	5,186,628.5
Telephone	153,894,892.13	78,730,112.65	22,910,827.4	11,928,314.52	73,521,634.36
Web	1,133,838,683.39	527,607,049.63	315,602,190.05	42,951,811.89	1,692,236,173.44

   Only the crosstab appears in the report.

8. On the rendered report tab, click **Run Report**.
9. In the **Display Options** list, select **Chart**.
   The results appear as follows:



10. Close the rendered report tab.
11. Remove the report without saving.
12. Leave the **IBM Cognos Analytics** portal open for the exercise.

Results:
You created a report that lets users choose whether to view data in chart or crosstab format. To create this report, you added a prompt to the report, added static choices for this prompt, and then added a Conditional block object to the report. You created a variable for the Conditional block object with values that depend on the static prompt choice that users select. You specified that different report objects display in the Conditional block object depending on the value applied.

IBM Training

Unit summary

- Control report displays using prompts
- Specify conditional formatting values using prompts
- Specify conditional rendering of objects based on prompt selection
- Create sorted and filtered reports based on prompt selection

Unit summary

IBM Training

Exercise 1

Create a prompt to let users choose the revenue sort order

Retailer name	Revenue Descending
Grand choix	$72,616,954.84
Chen Yu Enterprise Co.,	$71,970,453.88
VIP Department Stores	$50,813,492.98
Artículos de Campismo El Aquila, S.A. de C.V.	$46,093,027.97
Leisure Land	$45,381,496.34
Extreme Outdoors	$43,545,914.33
Naranco de Bulnes	$41,273,432.20

Design effective prompts

Exercise 1: Create a prompt to let users choose the revenue sort order

Exercise 1: Create a prompt to let users choose the revenue sort order

As report author, Frank Bretton, you have been requested to create a report by the marketing manager. The report needs to show the sales revenue generated for each retailer. The marketing manager wants to be able to quickly display either the retailers who generate the most or the least revenue so that she can create specific marketing campaigns for each. You will create a list report that contains a prompt that lets users choose to sort revenue values in either ascending or descending order.

To accomplish this:

- Create a list report using the GO data warehouse (query) package, Sales and Marketing (query), Sales (query) with retailer and revenue displayed.
- Add a value prompt to the Revenue column heading and name the parameter Sort.
- Create static choices for the value prompt for Ascending (use: 1) and Descending (use: -1).
- Change prompt properties to not Required, Auto-submit and Hide adornments. Add a Default selection of 1, for Ascending.
- Add a calculated Data item named sort, with Pre-sort set to Ascending. Add the sort item as a property of the list object.
- Run the report, and then view the default ascending results.
- Run the report, select Descending, and then view the descending results.

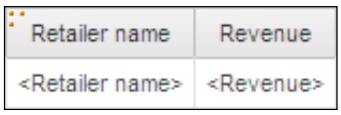
For more information about where to work and the exercise results, refer to the Tasks and results section that follows. If you need more information to complete a task, refer to earlier demonstrations for detailed steps.

Exercise 1: Tasks and results

Task 1. Create a list with a prompt on the report page.

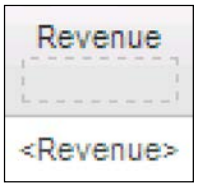
- **Side panel**: Open a new **List** template using the **GO data warehouse (query)** package.
- **Source** tab: Add the following query item to the **List** data container:
  - **Sales and Marketing (Query)/Sales (Query)/Retailers/Retailer name**
  - Add the following query item to the list data container:
    Sales and Marketing (Query)/Sales (Query)/Sales fact: Revenue

  The results appear as follows:



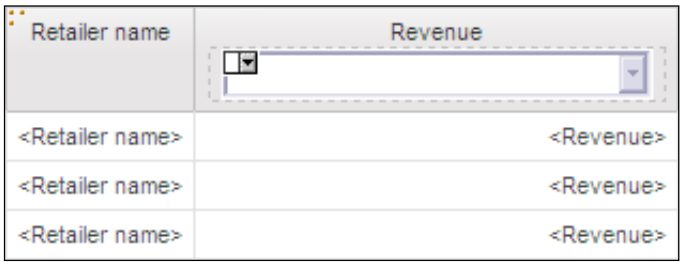
- **Application** bar**/More**: Unlock the report.
- **Toolbox** tab: Add a **Block** object to the **Revenue** list column title cell to the right of the **Revenue** text.

  The results appear as follows:



  - Add a **Value prompt** object, named **Sort**, to the block object that you just added, click **Next**, and then click **Finish**.

  The results appear as follows:



Task 2. Create static choices for the prompt.

- **Properties** pane: Add the following static choices for the value prompt:

Use	Display
1	Ascending
-1	Descending

  - Change the **Required** property to **No**; change the **Auto-submit** property to **Yes**, and the **Hide adornments** property to **Yes**.
  - Add a **Default selections** property of **1**.
- **Application** bar/**More**: Lock the report.

Task 3. Add a calculated data item to the query, and then add this item as a property of the list object.

- **Navigate** tab: Navigate to **Query1**.
- **Properties** pane: Rename **Query1** to **Retailer Revenue Query**.
- **Toolbox** tab: Add a **Data Item** object to the **Data Items** pane.
- **Expression Definition** pane: Rename **Data Item 1** to **Sort Key**.
  - Create and validate the following expression:

    [Revenue]*?Sort?
- **Properties pane**:
  - Change the **Pre-sort** property to **Sort ascending**.
- **Navigate** tab: Navigate to **Page1**.
- **Properties pane**: Add the **Sort Key** property to the list.
- **Properties pane**: Format the **<Revenue>** list column body to **$ (USD) - United States of America, dollar**.
  - Change the **Number of decimal places** property to **2**.

Task 4. Test the prompt.

- **Application** bar: Run the report in **HTML**.
- **Sort Prompt**: Select **Descending**.

  A section of the results appear as follows:

Retailer name	Revenue Descending
Grand choix	$72,616,954.84
Chen Yu Enterprise Co.,	$71,970,453.88
VIP Department Stores	$50,813,492.98
Artículos de Campismo El Aquila, S.A. de C.V.	$46,093,027.97
Leisure Land	$45,381,496.34
Extreme Outdoors	$43,545,914.33

- Close the rendered report tab.
- Close the report, without saving changes.
- Sign out of the **IBM Cognos Analytics** portal.
- Close the web browser.

You have created a report that shows the sales revenue generated for each retailer. You included a prompt that lets users choose to sort revenue values in either ascending or descending order.

IBM Training

Create additional advanced reports

IBM Cognos Analytics (v11.0)

IBM Training

Unit objectives

- Create a report that displays summarized data before detailed data
- Highlight alternate rows in a list report
- Create a report using an external data file
- Use single data items to summarize report information

Unit objectives

IBM Training

Display summary data before detailed data

- To let users quickly locate key data, you can create a report that displays summarized information before detailed data.

Product	Quantity	Revenue
Camping Equipment(Orders: 51,715)	27,301,149	1,589,036,664.03
Cooking Gear(Orders: 12,329)	13,400,351	272,835,984.18
TrailChef Canteen	965,723	11,333,518.65
TrailChef Cook Set	813,780	41,184,274.9
TrailChef Cup	1,812,123	5,702,502.7
TrailChef Deluxe Cook Set	442,136	53,195,154.45
TrailChef Double Flame	245,559	34,311,174.84

These rows display summarized data about sales of all products in this product line and product type

Create additional advanced reports

Display summary data before detailed data

You can display summary data before the details when it would be useful to know the summarized values before reviewing the detailed values.

In the slide example, to display the number of orders in the header row title cells, the author added a calculated column that counts the number of individual Quantity entries retrieved from the data source. Since each record in the data source represents one order and each order contains a Quantity value, by counting the individual instances of Quantity the query retrieves, the calculation indicates the number of orders made. Any other data item that would be part of each order record (such as Order number, Revenue, and so on) could be used instead of Quantity.

IBM Training

Demonstration 1

Create a report that displays summary data before detailed data

Product Orders

Total Quantity Sold: 89,237,091

Total Revenue: $4,686,775,768.85

Camping Equipment (Orders: 51,715) Total Quantity: 27,301,149 Total Revenue: 1,589,036,664.03

Cooking Gear (Orders: 12,329) Total Quantity: 13,400,351 Total Revenue: 272,835,984.18

Product	Quantity	Revenue
TrailChef Canteen	965,723	11,333,518.65
TrailChef Cook Set	813,780	41,184,274.9
TrailChef Cup	1,812,123	5,702,502.7
TrailChef Deluxe Cook Set	442,136	53,195,154.45
TrailChef Double Flame	245,559	34,311,174.84
TrailChef Kettle	2,336,950	25,368,496.06
TrailChef Kitchen Kit	866,669	19,535,825.83
TrailChef Single Flame	686,493	43,189,819.56
TrailChef Utensils	922,090	15,958,075.73
TrailChef Water Bag	4,308,828	23,057,141.46

Lanterns (Orders: 14,649) Total Quantity: 4,826,755 Total Revenue: 126,925,660.64

Product	Quantity	Revenue
EverGlow Butane	117,948	7,558,900.7

Create additional advanced reports

Demonstration 1: Create a report that displays summary data before detailed data

Demonstration 1: Create a report that displays summary data before detailed data

Purpose:
Management wants a report that lets them focus on summary data about total orders made, total quantity sold, and total revenue generated for each product line and product type. You will create a report where totals appear before the details. This report will have group headers that display summary data. To add additional context, below the report title, you will display the total quantity sold and total revenue generated by all products included in the report.

Portal:	**http://vclassbase:9300/bi**
User/Password:	**brettonf/Education1**
Application:	**IBM Cognos Analytics - Reporting**
Package:	**Team content\Samples\Models\GO data warehouse (query)**
Report Type:	**List**
Folder:	**Sales and Marketing (query)**
Namespace:	**Sales (query)**

Task 1. Create a list report and add a Count column.

1. Open a new **List** template using the **GO data warehouse (query)** package.
2. From the **Data/Source** tab, add the following query items to the list data container:
   - Products: **Product line**, **Product type**, **Product**
   - Sales fact: **Quantity**, **Revenue**

Product line	Product type	Product	Quantity	Revenue
<Product line>	<Product type>	<Product>	<Quantity>	<Revenue>
<Product line>	<Product type>	<Product>	<Quantity>	<Revenue>
<Product line>	<Product type>	<Product>	<Quantity>	<Revenue>

You want the Product line and Product type headers to display the total number of orders made for each Product line and Product type.

To obtain this data, you will add a calculated column that counts the number of individual Quantity entries retrieved from the data source. Since each record in the data source represents one order and each order contains a Quantity value, by counting the individual instances of Quantity the query retrieves, you can determine how many orders were made.

3. From the **Toolbox** tab, under **TEXTUAL**, drag a **Query calculation** object to the end of the list.
4. In the **Data item expression** dialog box, in the **Name** field, type **Count Orders**.
5. In the **Expression Definition** pane, create and validate the following expression:

   count([Quantity])

   Hints:

   - Data Items tab: Quantity.
   - You can get the count() function from the Functions tab, in the Summaries folder, if you prefer to not type the function.
6. Click **OK**.

   Although you chose to use the Quantity measure in the calculation in this task, this was not the only possible choice. For example, you could have used the Revenue measure in the calculation and obtained the same result.

   You will now name the query.
7. Click the **<Product>** list column body, from the On demand toolbar, under **More**, click **Go to query**.
8. On the **Application** bar, open the **Properties** pane.
9. In the **Properties** pane, under the **MISCELLANEOUS** section, change the **Name** property from **Query1** to **Product Quantity and Revenue**.
10. From the **Navigate** tab, click **Page1**.

Task 2. Group and summarize items and add headers.

1. In the list data container, group the <**Product line>** and <**Product type>** columns.
2. Click the **<Quantity>** list column body, and then summarize by **Total**.
3. Repeat step **2** for **<Revenue>** and **<Count Orders>**.
4. Click the **<Product line>** list column body, from the On demand toolbar click **Headers & footers**, and then click **Create header**.
5. With the **<Product line>** list column body still selected, from the On demand toolbar, under **More**, click **Delete** to remove the redundant column from the list data container.

   The header for <Product line> remains in the report.

6. Repeat steps **4** and **5** to create a **<Product type>** header and remove the **<Product type>** column.

   A section of the result appears as follows:

Product	Quantity	Revenue	Count Orders
<Product line>			
<Product type>			
<Product>	<Quantity>	<Revenue>	<Count Orders>
<Product type> - Total	<Total(Quantity)>	<Total(Revenue)>	<Total(Count Orders)>
<Product type>			
<Product>	<Quantity>	<Revenue>	<Count Orders>
<Product type> - Total	<Total(Quantity)>	<Total(Revenue)>	<Total(Count Orders)>
<Product line> - Total	<Total(Quantity)>	<Total(Revenue)>	<Total(Count Orders)>

Task 3. Move summary data from footers to header tables.

You want to add summary data to table cells in the header rows, so you will need to unlock the report.

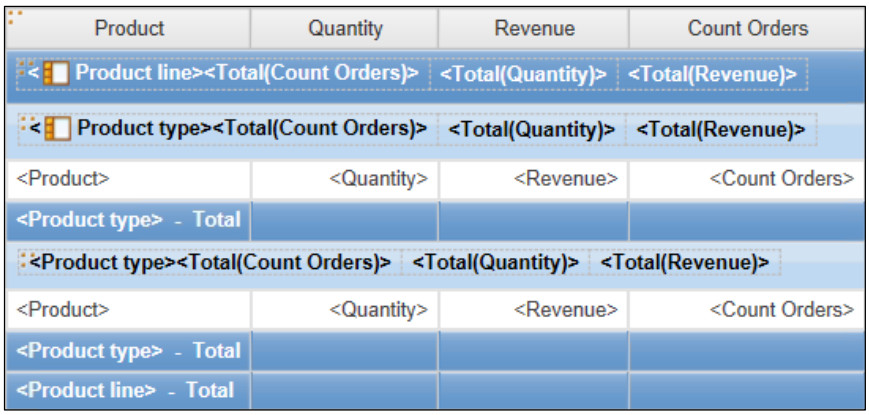
1. On the **Application** bar, under **More**, click **Locked** to unlock the report.
2. From the **Toolbox** tab, drag a **Table** object to the right of the **<Product line>** header.
3. Set the **Table** to **3 columns** and **1 row** and then clear the **Maximize width** check box.
4. Click the **Select Table** button in the top left corner of the table, to select the entire table.
5. On the **Properties** pane, set the table font to **Arial**, **8pt**, **Bold,** and a **Foreground color** of **White**.
6. Click **OK** to close all open dialog boxes.
7. Ctrl+click and then drag a copy of the table to the right of the **<Product type>** header.
8. Ensure the new table is the only one selected, and then change the **Foreground color** to **Black**.
9. Click and drag the <**Product line>** header into the first cell of the table right next to it.
10. From the **<Product line> - Total** footer row at the bottom of the list, click and drag the **<Total(Quantity)>** item to the empty cell to the right of the **<Product line>** header cell at the top of the list.
11. Click and drag the **<Total(Revenue)>** item to the empty cell to the right of the **<Total(Quantity)>** cell.

12. Click and drag the **<Total(Count Orders)>** item to the right of **<Product line>** in the first cell.

    A section of the results appear as follows:

13. Repeat steps **9** through **12** to move the **<Product type>** header, and then the **<Total(Quantity)>**, **<Total(Revenue)**>, and **<Total(Count Orders)>** items from the **<Product type> - Total foote**r row to the **<Product type>** header.
14. From the **Application** bar, under **More**, click **Unlocked** to lock the report.

    A section of the result appears as follows:



    You no longer require the empty footers, so you will delete them from this report.

15. Click the **<Product**> list column body in the list data container, from the On demand toolbar click **Headers & footers**, click **List headers & footers**, clear the **Product type (footer)**, **Product line (footer)**, and **Overall footer** check boxes, and then click **OK**.

16. Run the report in **HTML**.

    A section of the result appears as follows:

Product	Quantity	Revenue	Count Orders
Camping Equipment51,715 27,301,149 1,589,036,664.03			
Cooking Gear12,329 13,400,351 272,835,984.18			
TrailChef Canteen	965,723	11,333,518.65	1,180
TrailChef Cook Set	813,780	41,184,274.9	1,394

    The total quantity sold, revenue generated, and the total number of orders for each product line and product type appears in the headers.

17. Close the rendered report tab.

Task 4. Format the report.

You will now format the headers to be more readable and informative.

1. From the **Application** bar, under **More**, click **Locked** to unlock the report.
2. From **Toolbox** tab, drag a **Text item** object between **<Product line>** and **<Total(Count Orders)>**, in the **Product line** header.
3. In the **Text** dialog box, press the spacebar, type **(Orders:**, press the spacebar again, and then click **OK**.
4. From the **Toolbox** tab, drag a **Text item** object to the right of the **<Total(Count Orders)>** item (in the header), in the **Text** dialog box type **)**, and then click **OK**.
5. From **Toolbox** tab, drag a **Text item** object to the left of the **<Total(Quantity)>** item (in the header), in the **Text** dialog box press the space bar 3 times, type **Total Quantity:**, press the space again, and then click **OK**.
6. Repeat step **5** to create a text item, **Total Revenue:**, for **<Total(Revenue)>**.
7. Repeat steps **2** through **6** to add text items to the **<Product type>** header row.

   A section of the result appears as follows:

Product	Quantity	Revenue	Count Orders
<Product line> (Orders: <Total(Count Orders)>)		Total Quantity: <Total(Quantity)>	Total Revenue: <Total(Revenue)>
<Product type> (Orders: <Total(Count Orders)>		Total Quantity: <Total(Quantity)>	Total Revenue: <Total(Revenue)>

8. Click **<Product>**, and on the **Application** bar, under **More**, click **Unlocked**.

   You no longer require the Count Orders column in the report layout.

9. Click the **Count Orders** column header, and then from the On demand toolbar, under **More**, click **Delete**.

   To make this report easier to read, you want to indent the <Product type> and the <Product> items in the Product column body.

10. In the list data container, click the **<Product type>** header cell, and then press the **Tab** key.
11. With the cell still selected, in the **Properties** pane, under the **BOX** section, double-click the **Padding** property.

    Notice that because you pressed the Tab key, the Left padding is now set to 20 pixels.
12. Click **OK** to close the **Padding** dialog box.
13. In the report, click a **<Product>** cell, and then press the **Tab** key twice.

    In the Properties pane, under the BOX section, observe that the Padding property for the Left padding has changed to 40 pixels.

Task 5. Add a report title and add quantity and revenue summary labels below the report title.

You will add data to the report title displaying the total revenue generated and quantity sold by all product lines included in the report.

1. In the report, change the report title text to **Product Orders**.
2. From the **Toolbox** tab, drag a **Table** object (**1** column, **2** rows) to the right of the report title block.
3. From the **Toolbox** tab, drag a **Text item** object to the top row of the table that you just added.
4. In the **Text** dialog box, type **Total Quantity Sold:**, press the spacebar, and then click **OK**.
5. From the **Toolbox** tab, drag a **Text Item** object to the bottom row of the table.
6. In the **Text** dialog box, type **Total Revenue:**, press the spacebar, and then click **OK**.

   A section of the result appears as follows:

Task 6. Add quantity and revenue summary data using a singleton.

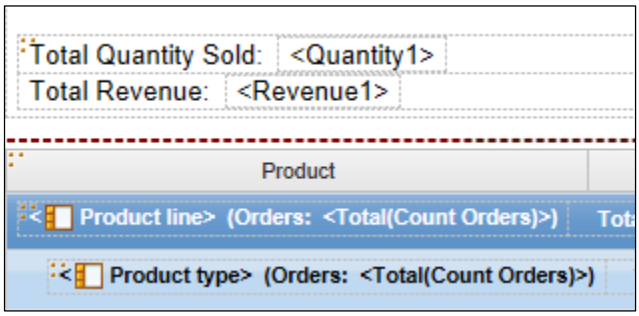
You need to add a query calculation to display the total quantity sold by all records retrieved by the query. To do this, you need to set the Product Quantity and Revenue query as a property of the Page object.

1. From the **Toolbox** tab, under **DATA CONTAINER**, drag a **Singleton** object to the right of the **Total Quantity Sold:** text in the report header.

   Singletons will be discussed in more detail, later in this unit.
2. From the **Object and query name** dialog box, in the **Query Name** list, select **Product Quantity and Revenue**, and then click **OK**.
3. With the **Singleton** selected, from the **Data/Source** tab, drag **Quantity** onto the singleton object in the report.
4. With the **Singleton** still selected, from the On demand toolbar, under **More**, click **Go to query**.
5. In the **Data Items** pane, double-click **Quantity1**.
6. Update and validate the existing expression as follows:

   total([Sales (query)].[Sales fact].[Quantity]for report)
7. Click **OK**.
8. Navigate to **Page1**, and then repeat steps **1** through **7** to create the summary total for **Total Revenue:**, using **Revenue**.
9. Navigate to **Page1**.

   A section of the report layout appears as follows:



Task 7. Format the quantity and revenue summary information.

1. In the table of the report title header, click the top row, Ctrl+click the second row, and then in the **Properties** pane, under the **MISCELLANEOUS** section, double-click the **Classes** property.
2. In the **Global classes** list, double-click **Report title area** to add it to the **Selected classes** pane, and then click **OK**.

   You applied the formatting of the Report title area, but want to change the font size and weight for these items.
3. With the two rows still selected, on the **Properties** pane, change the **Font Size** to **10 pt**, and then click **OK**.
4. Click the **<Revenue1>** singleton, and then in the **Properties** pane, under the **DATA** section, change the **Data format** property to **Format type** of **Currency**, the **Currency** property to **$ (USD) - United States of America, dollar**, and then click **OK** to close the **Data format** dialog box.
5. Click anywhere on the page below the list data container to select the page body, then from the On demand toolbar, click **Center**.
6. Click the list **Container Selector** to select the entire list, and then in the **Properties** pane, under **GENERAL**, change the **Column titles** property to **At start of details**.
7. Run the report in **HTML**.

   A section of the result appears as follows:

Product Orders

Total Quantity Sold: 89,237,091

Total Revenue: $4,686,775,768.85

Camping Equipment (Orders: 51,715) Total Quantity: 27,301,149 Total Revenue: 1,589,036,664.03		
Cooking Gear (Orders: 12,329) Total Quantity: 13,400,351 Total Revenue: 272,835,984.18		
Product	Quantity	Revenue
TrailChef Canteen	965,723	11,333,518.65
TrailChef Cook Set	813,780	41,184,274.9
TrailChef Cup	1,812,123	5,702,502.7
TrailChef Deluxe Cook Set	442,136	53,195,154.45
TrailChef Double Flame	245,559	34,311,174.84
TrailChef Kettle	2,336,950	25,368,496.06
TrailChef Kitchen Kit	866,669	19,535,825.83
TrailChef Single Flame	686,493	43,189,819.56
TrailChef Utensils	922,090	15,958,075.73
TrailChef Water Bag	4,308,828	23,057,141.46
Lanterns (Orders: 14,649) Total Quantity: 4,826,755 Total Revenue: 126,925,660.64		
Product	Quantity	Revenue
EverGlow Butane	117,948	7,558,900.7

The text under the report title displays the total quantity sold and total revenue generated by all product lines.

8. Close the rendered report tab.
9. Remove the report without saving it.
10. Leave **IBM Cognos Analytics - Reporting** open for the next demonstration.

Results:
You created a report that lets users quickly locate total quantity sold, total revenue generated, and total orders made for each product type and product line. To give additional context, you added report summary totals for quantity and revenue below the report title.

IBM Training

Highlight alternate rows

Product name	Revenue	Quantity	Calc.
TrailChef Cup	$218,416	51,834	1
TrailChef Kettle	$118,595	9,900	0
Star Gazer 2	$9,123,447	18,262	1
Star Peg	$85,937	44,686	0
TrailChef Utensils	$268, 084	15,928	1
Firefly Lite	$179,749	13,558	0

Use an expression to control the conditional formatting of alternating rows

Create additional advanced reports

Highlight alternate rows

To make a report easier to read, you can format a list so that rows appear in two alternating colors. To highlight alternate rows in a report, create a calculated data item that returns 0 for even rows and 1 for odd rows. Then, apply conditional formatting based on the report data.

You may want to leave the calculated column in the report for testing purposes. The calculated column need not appear in the final report, although the calculated data item must be a property of the List object.

The mod operator, used in an expression such as in the Highlight Alternate Rows slide example, works as follows:

- If you ask the mod function to return the remainder of 1 divided by 2, since the remainder would not be a whole number, the mod function will return the numerator (1).
- 2/2 will have a remainder of 0
- 3/2 will have a remainder of 1
- 4/2 will have remainder of 0, etc.

Because you use the running-count summary function to assign a sequential number to every row, this technique can only be used on list reports, and is not recommended for crosstabs.

A quick way to apply identical conditional formatting to all columns in a list report is to apply it to the List Columns Body Style. This saves time, and will also apply the formatting to any future columns you may add to the report.

IBM Training

Demonstration 2

Highlight alternate rows in a list report

Employee name	Salary	Bonus value
Aaghie Heiman	60,500	$6,576.92
Aaltje Hansen	31,730.77	$0.00
Abel Antunes	63,461.54	$6,969.23
Abram Ruiz	46,653.85	$0.00
Ada Morales	29,190.77	$3,228.40
Adara Cruz	63,461.54	$7,056.92
Adda Heijman	42,576.92	$0.00
Adelaide Wiesinger	27,943.08	$6,406.89
Adeline Arnaud	25,076.92	$1,883.08
Adelma Ortiz	41,153.85	$4,753.85
Adriaantje Haanraads	42,576.92	$0.00

Demonstration 2: Highlight alternate rows in a list report

Demonstration 2: Highlight alternate rows in a list report

Purpose:
The Human Resources department has requested a report showing bonus and salary data for each member of the sales staff. To make the report easier to read, every other row must be highlighted in blue. You will create a calculated item that returns 0 for even rows and 1 for odd rows, and will then conditionally format every other row to be highlighted in blue.

Portal: **http://vclassbase:9300/bi**
User/Password: **brettonf/Education1**
Application: **IBM Cognos Analytics - Reporting**
Package: **Team content\Samples\Models\GO data warehouse (query)**
Report Type: **List**
Folder: **HR (query)**
Namespace: **Employee summary (query)**

Task 1. Create a list report.

1. Open a new **List** template using the **GO data warehouse (query)** package.
2. From the **HR (query)** folder**Employee summary (query)** namespace, add the following items to the list data container, and then sort the **Employee name** column in ascending order:
   - Employee by region: **Employee name**
   - Employee summary fact: **Salary**, **Bonus value**

Employee name	Salary	Bonus value
<Employee name>	<Salary>	<Bonus value>

3. Run the Report in **HTML**.
   A section of the result appears as follows:

Employee name	Salary	Bonus value
Aaghie Heiman	60,500	$6,576.92
Aaltje Hansen	31,730.77	$0.00

   To make this report easier to read, you want to highlight every other row in blue. You will create a calculated data item that will return zeros for even rows and ones for odd rows. You can then apply conditional formatting to the report based on this calculated item.
4. Close the rendered report tab.

Task 2. Add a calculated data item.

1. On the **Navigate** tab, click the **Query explorer** tab, and then click **Query1**.
2. From the **Toolbox** tab, drag a **Data Item** object to the **Data Items** pane.

   You will create a calculation that counts each row in the report that contains a Salary data item value. The calculation will then divide the row number by two and return the remainder. The remainder for even rows will be 0 and the remainder for odd rows will be 1.
3. In the **Data item expression** dialog, in the **Name** field, type **EvenOdd**.
4. Create and validate the following expression:

   mod(running-count([Employee name]),2)

   Hints:

   - Function tab: mod() is in the Common Functions folder\M-Q folder; running-count() is in the Summaries folder.
   - Data Items tab: Employee name.

   You are using the Employee name because of the possibility that Salary and Bonus may have $0.00 rows that will not be counted. This would cause a duplication of a highlighted row. Since the Employee name row will always have a value, its rows will always have alternate highlighting.
5. Click **OK**.
6. On the **Application** bar, open the **Properties** pane.
7. On the **Navigate** tab, click **Query1**, and then, on the **Properties** pane, rename the query to **Employee Salary Information**.

Task 3. Add the calculated item to the report and then create a conditional variable.

You will add the EvenOdd data item to the list to see the values returned in the report.

1. Navigate to **Page1**.

2. From the **Data/Data items** tab, drag the **EvenOdd** item to the end of the list, and then run the report in **HTML**.

   A section of the result appears as follows:

Employee name	Salary	Bonus value	EvenOdd
Aaghie Heiman	60,500	$6,576.92	1
Aaltje Hansen	31,730.77	$0.00	0
Abel Antunes	63,461.54	$6,969.23	1
Abram Ruiz	46,653.85	$0.00	0

   In even rows, the EvenOdd value is 0 and in odd rows the value is 1. These values will be used to apply conditional formatting to alternating rows in the report.

   You do not want to display the calculated data item in the report layout, however, you need to make this calculated item a property of the list object.

3. Close the rendered report tab.
4. In the list, click the **<EvenOdd>** list column body, on the **Properties** pane, click **Select Ancestor**, and then click **List column**.
5. In the **Properties** pane, under the **CONDITIONAL** section, change the **Render** property to **No**.

   This maintains the EvenOdd column as a property of the list but does not render it at run time.

   You will create a Boolean variable that will let you apply formatting to alternate rows depending on whether the EvenOdd calculation is value is 1 or 0.

6. On the **Navigate** tab, click the **Condition explorer** tab, click **Variables**, and then on the **Toolbox** tab, double-click **Boolean Variable**.
7. In the **Report expression** dialog box, create and validate the following expression:

   [Employee Salary Information].[EvenOdd]=0

   Hint: Drag EvenOdd from the Available Components pane.

8. Click **OK** to close the **Report expression** dialog box.

   You created a Boolean variable because there are only two possible choices: the EvenOdd value can be 1 or 0. You can now apply this variable to the report so if the value is Yes (because the row is even and the EvenOdd value equals 0), the row will be highlighted in blue.

9. With the variable still selected, in the **Properties** pane, under the **MISCELLANEOUS** section, change the **Name** property to **EvenRows**.
10. Navigate to **Page1**.

Task 4. Apply conditional formatting and run the report.

1. In the list data container, click any list column body in the list, in the **Properties** pane, click **Select Ancestor**, and then click **List columns body style**.
2. In the **Properties** pane, under the **CONDITIONAL** section, double-click the **Style variable** property.
3. From the Variable list, select **EvenRows**, and then click **OK**.

   You will now specify that even rows be highlighted in blue.
4. On the **Navigate** tab, click the **Condition explorer** tab, and then from under **EvenRows**, click **Yes**.

   A green message box indicates that conditional formatting is turned on and that the EvenRows variable is set to 1.
5. On the **Properties** pane, under **COLOR & BACKGROUND**, double-click **Background color**, and then click the **Color swatch** tab.
6. Click the blue (**#99CCFF**) box, located five rows from the bottom and seven columns in from the right, and then click **OK**.

   The list columns are highlighted in blue.
7. On the **Conditional explorer** tab, click the **EvenRows** variable to turn off conditional formatting mode.
8. Run the report in **HTML**.

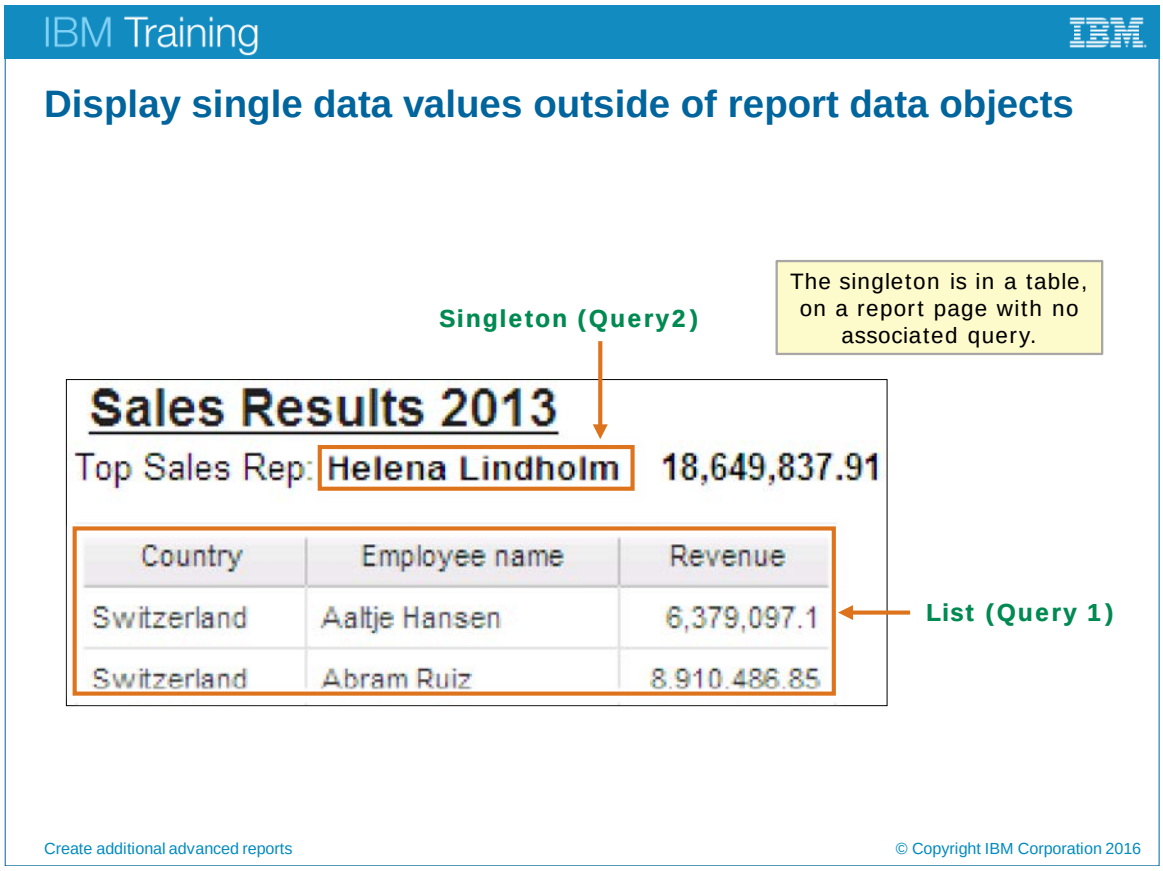
   A section of the result appears as follows:

Employee name	Salary	Bonus value
Aaghie Heiman	60,500	$6,576.92
Aaltje Hansen	31,730.77	$0.00
Abel Antunes	63,461.54	$6,969.23
Abram Ruiz	46,653.85	$0.00
Ada Morales	29,190.77	$3,228.40
Adara Cruz	63,461.54	$7,056.92
Adda Heijman	42,576.92	$0.00
Adelaide Wiesinger	27,943.08	$6,406.89
Adeline Arnaud	25,076.92	$1,883.08
Adelma Ortiz	41,153.85	$4,753.85
Adriaantje Haanraads	42,576.92	$0.00

   Even rows in the list report are highlighted in blue.

9. Close the rendered report tab.
10. Remove the report without saving.
11. Leave the **IBM Cognos Analytics** portal open for the next demonstration.

Results:
You created a report in which every other row is highlighted in blue. To do this, you created a calculated item that returned 0 for even rows and 1 for odd rows. You then applied conditional formatting to the list columns so that every other row was highlighted in blue.



Display single data values outside of report data objects

Singletons allow you to display single data values in a report without having to set the page-query association or join unrelated queries. You cannot sort singleton queries, since only one row is returned. You also cannot set 'no data' options on a singleton item.

When you drop a singleton item into a report, it creates a query that retrieves only the first row value.

Singleton queries are not supported when producing report output in CSV format.

IBM Training

Demonstration 3

Use singletons to summarize information in a report

Sales Results 2013

Top Sales Rep: **Helena Lindholm** **18,649,837.91**

Country	Employee name	Revenue
Switzerland	Aaltje Hansen	6,379,097.1
Switzerland	Abram Ruiz	8,910,486.85
Switzerland	Adda Heijman	8,131,762.35
Switzerland	Adriaantje Haanraads	7,578,816.55

Demonstration 3: Use singletons to summarize information in a report

Demonstration 3: Use singletons to summarize information in a report

Purpose:
You want to create a list report showing the revenue generated by each sales rep for each year. You then want to create a table using singletons to show the top sales representative for the year 2013. This table will be in the header, so as to be seen on every page, to make it easier for managers to compare results of their sales representatives to the top sales representative of 2013.

Portal: **http://vclassbase:9300/bi**
User/Password: **brettonf/Education1**
Application: **IBM Cognos Analytics - Reporting**
Package: **Team content\Samples\Models\GO data warehouse (query)**
Report Type: **List**
Folder: **Sales and Marketing (query)**
Namespace: **Sales (query)**

Task 1. Create a list report with a title and table in the header.

1. Open a new **List** template using the **GO data warehouse (query)** package.
2. On the **Data/Source** tab, add the following items to the list data container:
   - Time: **Year**
   - Employee by region: **Country**, **Employee name**
   - Sales fact: **Revenue**

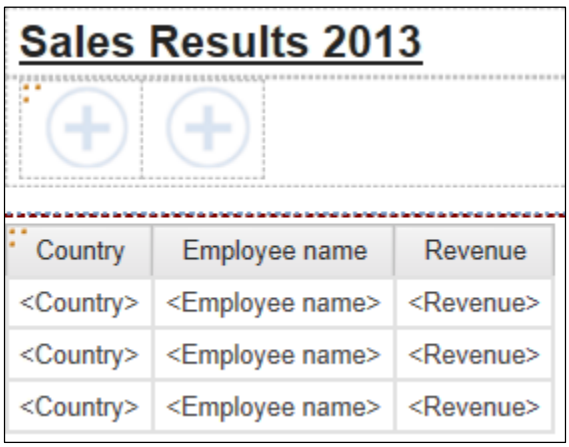
Year	Country	Employee name	Revenue
<Year>	<Country>	<Employee name>	<Revenue>
<Year>	<Country>	<Employee name>	<Revenue>
<Year>	<Country>	<Employee name>	<Revenue>

3. In the list, click **Year**, from the On demand toolbar, under **More**, click **Go to query**.
4. From the **Data Items** pane, drag **Year** to the **Detail Filters** pane, and then create and validate the following expression:
   [Year]=2013
5. Click **OK** to close the **Detail filter expression** dialog box.

6. Navigate to **Page1**, click the **<Year>** list column body to select it, and then from the On demand toolbar, under **More**, click **Cut**.

   You have removed Year from the report layout, but it remains in the query.
7. Double-click the report title text, change the title to **Sales Results 2013**, and then left align the title block.
8. From the **Toolbox** tab, drag a **Block** object below the title text, and then drag a **Table** object into the new block.
9. Accept the default number of columns (**2**) and rows (**1**), clear the **Maximize width** check box, and then click **OK**.

   A section of the result appears as follows:



Task 2. Add a singleton to a table.

1. From the **Toolbox** tab, drag a **Text item** into the left table cell, type **Top Sales Rep:**, press the spacebar, and then click **OK**.
2. On the **Data/Source** tab, from **Employee by region**, drag **Employee name** to the right of the new text item in the left table cell.
3. Click **OK** to accept the new singleton.
4. On the **Application** bar, open the **Properties** pane.
5. From the **Navigate/Query explorer** tab, click **Queries**.
6. Click **Query2**, and then change the query name to **Top Sales Rep**.
7. Run the report in **HTML** to view the table.

   A section of the results appear as follows:

   Top Sales Rep: Aaghie Heiman

   This is not the actual top sales rep that you want in the table. When you drop a singleton item into a report, it creates a query that retrieves only the first row value. You need to customize the query to display the correct value.
8. Close the rendered report tab.

Task 3. Customize the query.

1. Navigate back to **Page1**.
2. In the work area, click the **<Employee name>** singleton, and then from the On demand toolbar, under **More**, click **Go to query**.
3. On the **Data**/**Source** tab, from **Sales fact**, drag **Revenue** to the **Data Items** pane.
4. Double-click **Revenue** to open the **Data item expression** dialog box.
5. Edit and validate the expression as follows:
   total([Sales (query)].[Sales fact].[Revenue] for [Employee name])
   Hint: Data Items tab: Employee name.
6. Click **OK** to close the **Data item expression** dialog box.
7. From the **Toolbox** tab, drag a **Data Item** into the **Data Items** pane.
8. In the **Name** field, type **Maximum Sales Rep Revenue**.
9. Create and validate the following expression:
   maximum ([Revenue] for report)
   Hint: Data Items tab: Revenue.
10. Click **OK**.
11. From the **Data Items** pane, drag **Revenue** into the **Detail Filters** pane.
12. Create and validate the following expression:
    [Revenue]=[Maximum Sales Rep Revenue]
    Hint: Data Items tab: Maximum Sales Rep Revenue.
13. Click **OK**.
14. On the **Data/ Source** tab, from the **Time** query subject, drag **Year** into the **Detail Filters** pane.
15. Create and validate the following expression:
    [Sales (query)].[Time].[Year]=2013
16. Click **OK** to close the dialog box.

Task 4. Add a singleton to the table and format the table items.

1. Navigate to **Page1**, and then from the **Data** tab, click the **Data items** tab.

   You now have new items available to include in your table. You want to include the new Maximum Sales Rep Revenue item in your header table.
2. From the **Data items** tab, drag **Maximum Sales Rep Revenue** to the right table cell, and then click **OK** to accept the singleton.
3. Click the **<Maximum Sales Rep Revenue>** singleton in the table, in the **Properties** pane, click **Select Ancestor**, and then click **Table cell**.
4. In the **Properties** pane, under the **BOX** section, change the **Padding** property to apply **15** pixels of left padding.
5. In the table, click **<Employee name>**, Ctrl-click **<Maximum Sales Rep Revenue>**, and then from the On demand toolbar, change the font **Weight** to **Bold**.
6. Run the report in **HTML**.

   A section of the result appears as follows:

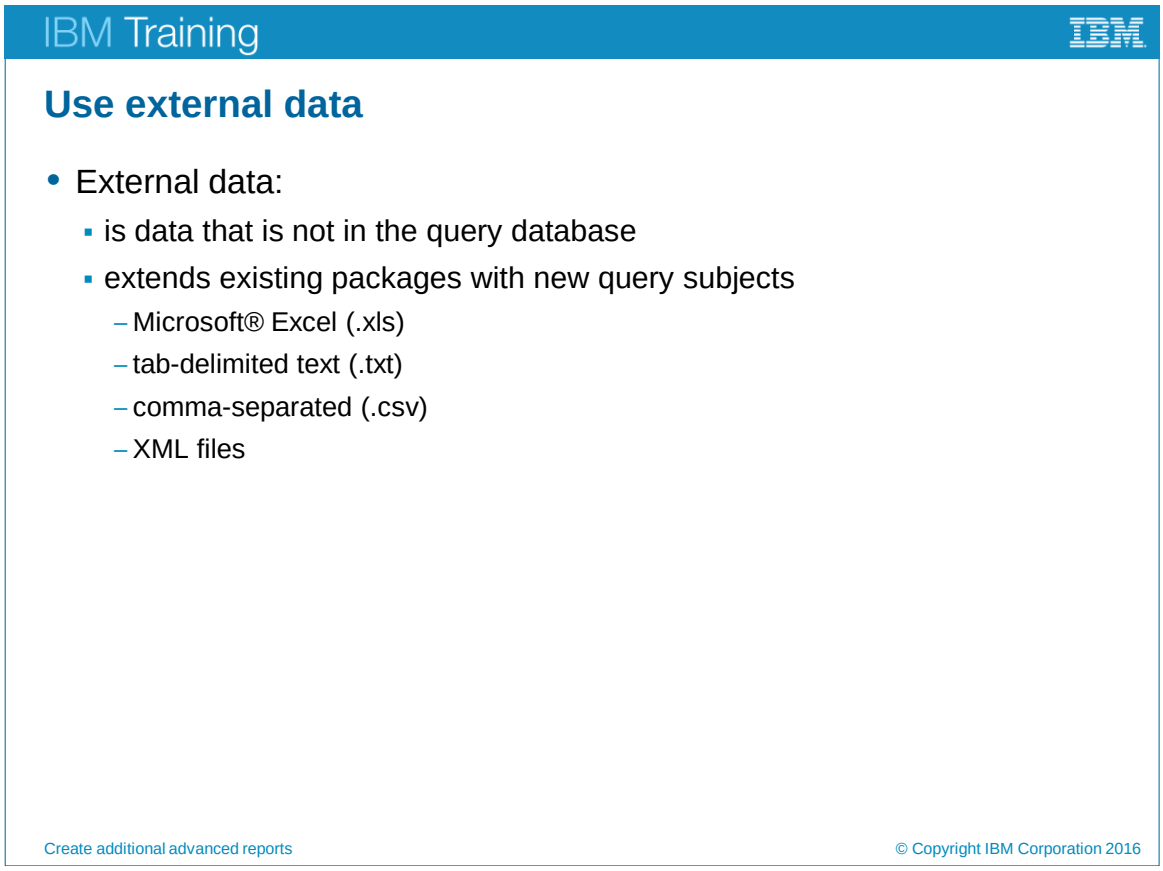
Sales Results 2013

Top Sales Rep: **Helena Lindholm** **18,649,837.91**

Country	Employee name	Revenue
Switzerland	Aaltje Hansen	6,379,097.1
Switzerland	Abram Ruiz	8,910,486.85
Switzerland	Adda Heijman	8,131,762.35
Switzerland	Adriaantje Haanraads	7,578,816.55

   As managers review the pages of the report, they can easily compare their sales rep results with the results of the top sales rep of 2013.
7. Close the rendered report tab.
8. Remove the report without saving.
9. Leave the **IBM Cognos Analytics** portal open for the next demonstration.

Results:
You created a list report showing the revenue generated by each sales rep for each year. You then created a table in the page header to display the top sales representative for the year 2013.



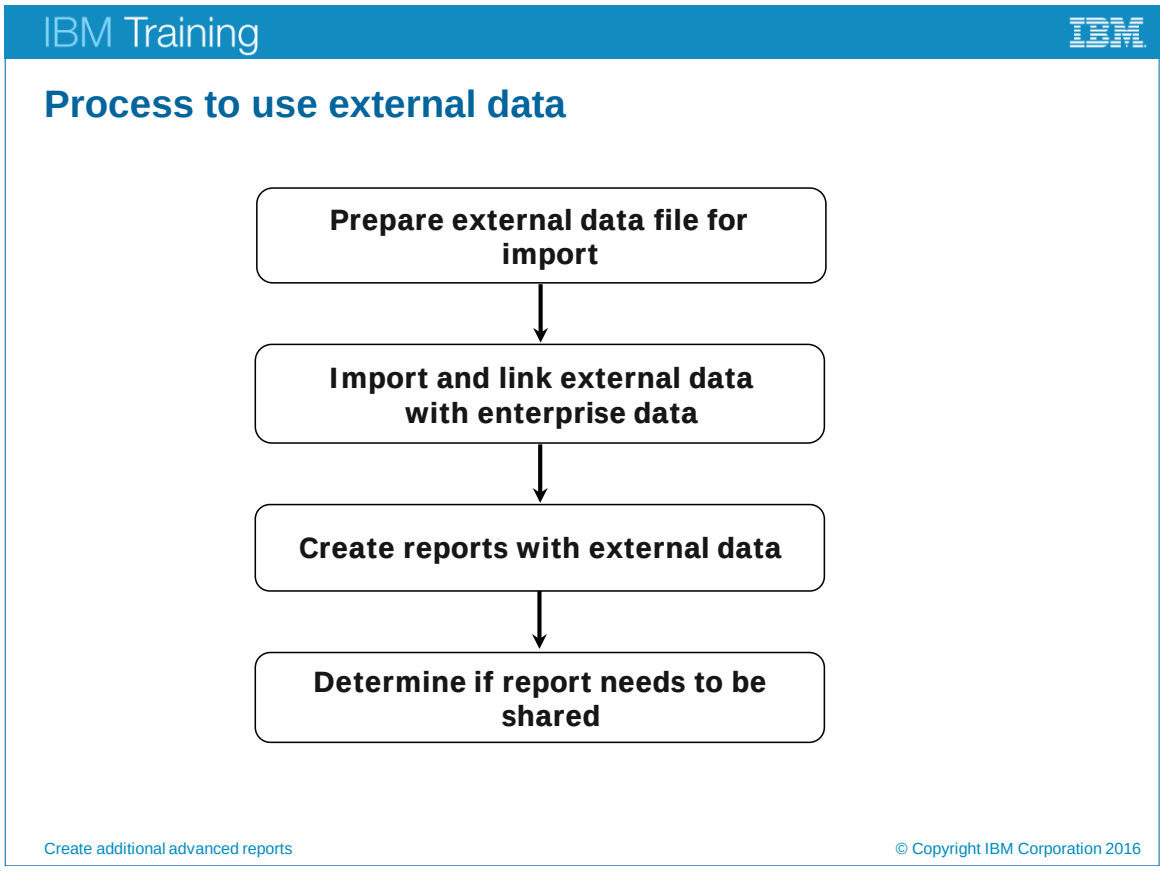
Use external data

Many times report authors need to add external data to IBM Cognos content to meet report requirements. In many cases, the authors are unable to merge the data from external data sources without intervention from IT. The objective of the External Data functionality is to provide this capability to the report author. This allows the author to do basic data modeling, including creating relationships between the data items and setting cardinality.

External data extends an existing package definition with new query subjects that are in the external data file. The definition involves two steps:

1. Identifying columns from the external data file to become new query subjects.
2. Defining how the External Data file will be related to the existing IBM Cognos content.

At least one query subject in the report must be mapped to a data item from their data. The mapping creates joins (or relationships) between external data and enterprise data. This will ensure that the data is integrated smoothly.



Process to use external data

External data extends an existing package definition by adding new query subjects from the external data source. As you prepare to import, identify which columns of the external data will become new query subjects, and then define how the external data will relate to the existing IBM Cognos content.

To work with external data:

1. Ensure the external data file matches your enterprise data. It is necessary to link at least one column of external data with your enterprise data source.
2. Import your external data file using the wizard.
3. Create reports. After imported and linked, the data appears in the data tree under the Source tab.
4. Determine if you want to share the reports that use your external data.

Before external data can be imported by a user, the IBM Cognos administrator must grant permission for the Allow External Data capability.

IBM Training

Demonstration 4

Create a report and add external data

COUNTRY_EN	CITY	Staff name	Sales staff code	Sales target
Australia	Melbourne	Dave Smythe	10090	$15,084,300.00
		Jake Cartel	10092	$3,786,500.00
		John Sinden	10093	$4,462,400.00
		Alice Walter	10089	$16,834,700.00
		Jackie Fulford	10527	$17,965,800.00
		Donald Neely	10526	$997,200.00
		Donald Ward	10091	$18,036,200.00
		Jonathan Farrel	10773	$2,073,100.00
	Melbourne - Total			**$79,240,200.00**
Australia - Total				**$79,240,200.00**

Create additional advanced reports

Demonstration 4: Create a report and add external data

Demonstration 4: Create a report and add external data

Purpose:
You have received a .csv file from the Human Resources manager, which includes updated city and country locations for all employees. You want to create a report that will include the employee details (name, employee code, and sales target numbers) from the corporate data, with the updated employee location information in the external file. To do this, you will create a report using the enterprise data, and then add the external data to the package. This will allow you to create a report that displays the sales targets and current employee location information.

Portal: **http://vclassbase:9300/bi**
User/Password: **brettonf/Education1**
Application: **IBM Cognos Analytics - Reporting**
Package: **Team content\Samples\Models\GO sales (query)**
Report Type: **List**
Namespace: **Sales target (query)**

Task 1. Create a list report.

1. Open a new **List** template using the **GO sales (query)** package.
2. From the **Data/Source** tab, add the following query items to the list data container:
   - Sales staff: **Staff name**, **Sales staff code**
   - Sales target: **Sales target**

Staff name	Sales staff code	Sales target
<Staff name>	<Sales staff code>	<Sales target>

3. Run the report in **HTML**.
   A section of the results appear similar to the following:

Staff name	Sales staff code	Sales target
Ennio Ricci	10233	$8,824,700.00
Alphonse Sauvage	10800	$37,298,800.00
Brendon Pike	10052	$22,148,600.00
Fiorenza Giordano	10346	$64,458,900.00

4. Close the rendered report tab, and then save the report to **Team content\B6059**, as **Staff Sales Targets**.

Task 2. Identify the external data file and columns for the report.

1. Click **Manage external data**  , above the Source data list.

   The External data wizard opens, to guide you through the process of identifying and mapping the external data. The process begins with selecting the external data.
2. In the **External data file** section, click **Browse**, navigate to **C:\Program Files (x86)\IBM\cognos\c10\webcontent\samples\datasources\cubes\PowerCubes\EN\great_outdoors_sales_en**, click **sales_reg.csv**, and then click **Open**.
3. Scroll through the list of selected items to review the file.

   You will bring in all columns at this time. You could clear check boxes for any columns that you do not require.
4. Select the **Allow the server to automatically load the file** check box.

   If you share this report, this option allows other users to run your report using their own version of the external data file. If they use their own version, the file must contain the same columns as the original external data file that you used to import the data and create the report. If you always want the report to run using your version of the external data file, then you should select the Allow the server to automatically load the file check box.

   Notice the Namespace for the external data box displays the default namespace that will be assigned. The namespace provides a unique name to associate with the data items that you import. The namespace appears in the data tree of the Source tab and is used to organize the data items. By default, the namespace is the imported file name without the extension. If you change the default name for the namespace, you are prompted to select the external data file each time you run the report. To avoid this, select the Allow the server to automatically load the file check box.
5. Click **Next** to proceed to the **Data Mapping** page.

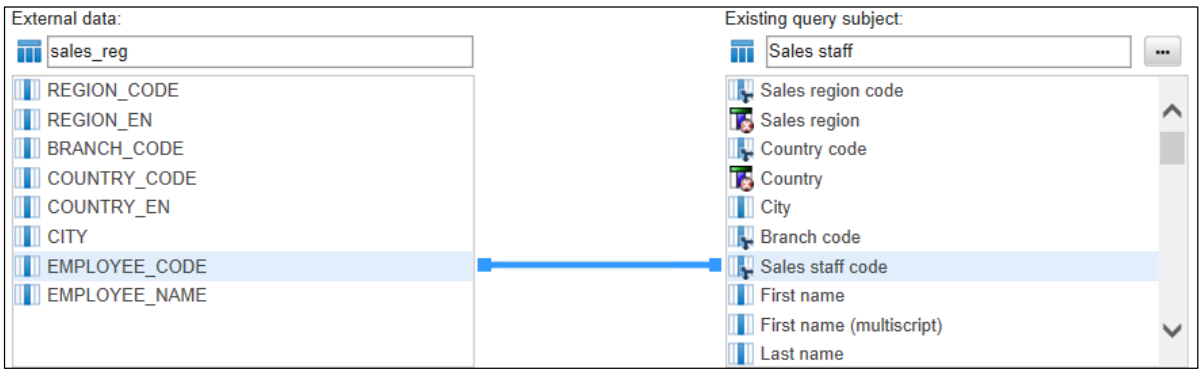
Task 3. Map the external data file with the existing data.

1. On the **Data Mapping** page, click the ellipsis under **Existing query subject / report**, and then click **Choose Query Subject**.
2. Under the **Sales target (query)** namespace, click the **Sales staff** query subject, and then click **OK**.

   The Existing query subject pane is populated with the available query items. You will now define the relationships between the query items in the external data and the query items in the corporate data of the existing query subject.
3. Click **New Link**.

   A line appears between the panes, defaulting to the first item in each list. You will define the relationship by selecting the items for the new link.
4. In the **External data** list click **EMPLOYEE_CODE**, and then in the **Existing query subject** list click **Sales staff code**.

   A section of the result appears as follows:



   You have defined a link.
5. Click **New Link** again, then in the **External data** list click **COUNTRY_CODE**, and then in the **Existing query subject** list click **Country code**.

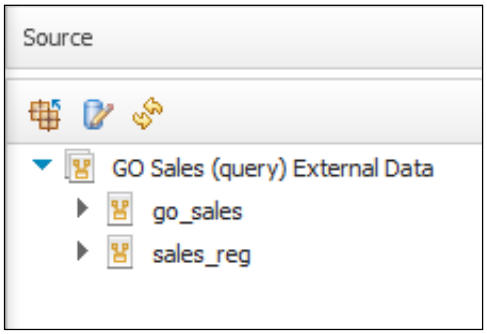
   You have identified joins between the two sources of data.
6. Click **Next** to define the attributes of the external data.

   Data item attributes can be modified on this page, if needed. You do not need to modify any attributes for this demonstration.
7. Click **Next** to define your mappings.

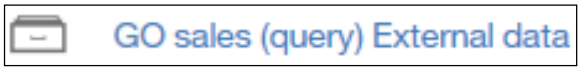
   A summary of the items is displayed. You will accept the default handling of values in the data source and in the report results.
8. Click **Finish**.

   The External Data wizard closes and the Manage external data dialog box displays the new package name. The default name uses the original package name, and appends External data.

9. Once you are satisfied with the package name, click **Publish**.
   The new package will be automatically saved in My content.
10. Click **Yes** if prompted to replace the existing package.
    The system updates the model metadata, and validates the currently open report against the new package.
11. Click **OK** after the report is validated.
    Notice that the report itself has not changed, but notice that the Source tab has updated to the newly created package that includes the new external data.
    The results appear as follows:



12. On the middle of the **Application** bar, click the down arrow to the right of the **Staff Sales Targets**, to show the currently active list, and then click **Welcome**.
13. Click the **My content** tab to see the package.



Task 4. Use the external data file in a report.

1. On the **Side panel**, click **New**, and then click **Report**.
2. Under **Package**, click the ellipsis, navigate to **My content**, click **GO sales (query) External data,** and then click **Open**.
3. Double-click **List**.
4. On the **Data/Source** tab, expand the **sales_reg** namespace.
5. Expand the **sales_reg** query subject, click **COUNTRY_EN**, Ctrl+click **CITY**, and then drag them to the list.
6. Expand the **go_sales** namespace, expand the **Sales targets (query)**, and then from **Sales staff**, drag **Staff name** and **Sales staff code** to the list data container.
7. From **Sales target**, drag **Sales target** to the end of the list.

8. In the list data container, group the **<COUNTRY_EN>** list column body and the **<CITY>** list column body.
9. On the **Application** bar, open the **Properties** pane.
10. Click the **<Sales target>** list column body, and then on the **Properties** pane, under the **DATA** section, double-click **Data format**.
11. In the **Format type** list, select **Currency**, and then set the following properties:
    - Currency: **$ (USD) - United States of America, dollar**
    - Use Thousands Separator: **Yes**
12. Click **OK** to close the **Data format** dialog box.
13. With the **<Sales target>** list column body still selected, from the On demand toolbar, click **Summarize**, and then click **Total**.

    A section of the report layout appears as follows:

COUNTRY_EN	CITY	Staff name	Sales staff code	Sales target
<COUNTRY_EN>	<CITY>	<Staff name>	<Sales staff code>	<Sales target>
	<CITY> - Total			<Total(Sales target)>
	<CITY>	<Staff name>	<Sales staff code>	<Sales target>
	<CITY> - Total			<Total(Sales target)>
<COUNTRY_EN> - Total				<Total(Sales target)>

14. Run the report in **HTML**.

    A section of the result appears similar to the following:

COUNTRY_EN	CITY	Staff name	Sales staff code	Sales target
Australia	Melbourne	Dave Smythe	10090	$15,084,300.00
		Jake Cartel	10092	$3,786,500.00
		John Sinden	10093	$4,462,400.00
		Alice Walter	10089	$16,834,700.00
		Jackie Fulford	10527	$17,965,800.00
		Donald Neely	10526	$997,200.00
		Donald Ward	10091	$18,036,200.00
		Jonathan Farrel	10773	$2,073,100.00
	Melbourne - Total			$79,240,200.00
Australia - Total				$79,240,200.00

15. Close the rendered report tab.
16. Close the report without saving.
17. Leave the **IBM Cognos Analytics** portal open for the exercise.

Results:
You received a .csv file from the Human Resources manager, which included updated city and country locations for all employees. You created a report with employee details using the updated employee location information in the external file.

IBM Training

Unit summary

- Create a report that displays summarized data before detailed data
- Highlight alternate rows in a list report
- Create a report using an external data file
- Use single data items to summarize report information

Unit summary

IBM Training

Exercise 1

Create a report with alternating sections of color

First name	Last name	Work phone	Extension	Fax	Email
Aaghie	Heiman	+(41) 17 12 13 11	9676	+(41) 17 12 61 82	AHeiman@grtd123.com
Aaltje	Hansen	+(41) 17 12 13 11	9640	+(41) 17 12 61 82	AHansen@grtd123.com
Abel	Antunes	+55 (11) 344-4444	2605	+55 (11) 333-2223	AAntunes@grtd123.com
Abram	Ruiz	+(41) 17 12 13 11	9762	+(41) 17 12 61 82	ARuiz@grtd123.com
Ada	Morales	+35 94 322 3540	7693	+35 91 548 1637	AMorales@grtd123.com
Adara	Cruz	+(39) 02 79 53 780	7704	+(39) 02 79 53 477	ACruz@grtd123.com
Adda	Heijman	+(41) 17 12 13 11	9639	+(41) 17 12 61 82	AHeijman@grtd123.com
Adelaide	Wiesinger	+(41) 17 12 13 11	9662	+(41) 17 12 61 82	AWiesinger@grtd123.com
Adeline	Arnaud	+33 1 68 94 52 20	3260	+33 1 68 94 56 60	AArnaud@grtd123.com
Adelma	Ortiz	+(39) 02 79 53 780	7705	+(39) 02 79 53 477	AOrtiz@grtd123.com
Adriaantje	Haanraads	+(41) 17 12 13 11	9638	+(41) 17 12 61 82	AHaanraads@grtd123.com
Adriana	Iacobucci	+(41) 17 12 13 11	9691	+(41) 17 12 61 82	AIacobucci@grtd123.com
Adrien	Martin	+32 16 20.73.21	3354	+32 16 20.73.32	AMartin1@grtd123.com
Adrienne	Roche	+33 1 68 94 52 20	3724	+33 1 68 94 56 60	ARoche@grtd123.com
Aert	Haak	+(41) 17 12 13 11	9787	+(41) 17 12 61 82	AHaak@grtd123.com
Aert	Meyer	+31 (0)20 692 93 94	6587	+31 (0)20 692 93 06	AMeyer@grtd123.com
Agatha	Reyes	+35 94 322 3540	7692	+35 91 548 1637	AReyes@grtd123.com
Agathe	Roque	+32 16 20.73.21	3715	+32 16 20.73.32	ARoque@grtd123.com
Agnelo	Chavez	+(41) 17 12 13 11	9761	+(41) 17 12 61 82	AChavez@grtd123.com

Create additional advanced reports

Exercise 1: Create a report with alternating sections of color

Exercise 1: Create a report with alternating section of color

The Human Resources department wants Frank Bretton, report author, to develop a report containing contact information for each employee, found in the Team content\Samples\Models\GO data warehouse (query) package using the Sales and Marketing (query)\Sales (query) namespace. To make this report easier to read, the department wants the report to contain alternating sections of green and white rows in the resulting list.

To accomplish this:

- Populate a new list report with contact information for employees by region, including first name, last name, work phone, extension, fax, and email.
- Add a calculated column to the report that lets you conditionally format alternating groups of rows to appear with a green background.
- Name the calculated item Green or White, add the calculated item to the report, and run the report to see the data generated in the calculated column.
- Conditionally format the list columns so every other five rows are highlighted in light green.
- Modify the report so the Green or White column does not appear in the report.
- Run the report.

For more information about where to work and the exercise results, refer to the Tasks and results section that follows. If you need more information to complete a task, refer to earlier demonstrations for detailed steps.

Exercise 1: Tasks and results

Task 1. Create a list report with a calculated column.

- **Side panel**: Open a new **List** template using the **GO data warehouse (query)** package.
- **Source tab**: Add the following query items to the list data container:
  - **Sales and Marketing (query)/Sales (query)**:
    - Employee by region: **First name**, **Last name**, **Work phone**, **Extension**, **Fax**, and **Email**

First name	Last name	Work phone	Extension	Fax	Email
<First name>	<Last name>	<Work phone>	<Extension>	<Fax>	<Email>

- **Navigate** tab: Navigate to **Query1**.
- **Properties** pane: Rename **Query1** to **Contact Information**.
- **Toolbox** tab: Add a **Data Item** object to the **Data Items** pane.
- **Data item expression - Data Item1** dialog box:
  - In the **Name** field, type **Green or White**.
  - Create and validate the following expression:

    mod(running-count([First name]),10)
- **Navigate** tab: Navigate to **Page1**.
- **Data/Data items** tab: Add the **Green or White** data item to the end of the list data container.
- **Application** bar: Run the report in **HTML**.

  A section of the results appear as follows:

First name	Last name	Work phone	Extension	Fax	Email	Green or White
Aaghie	Heiman	+(41) 17 12 13 11	9676	+(41) 17 12 61 82	AHeiman@grtd123.com	1
Aaltje	Hansen	+(41) 17 12 13 11	9640	+(41) 17 12 61 82	AHansen@grtd123.com	2
Abel	Antunes	+55 (11) 344-4444	2605	+55 (11) 333-2223	AAntunes@grtd123.com	3
Abram	Ruiz	+(41) 17 12 13 11	9762	+(41) 17 12 61 82	ARuiz@grtd123.com	4
Ada	Morales	+35 94 322 3540	7693	+35 91 548 1637	AMorales@grtd123.com	5

- Close the rendered report tab.

- **Data item expression - Green or White** dialog box: Modify and validate the expression for **<Green or White>** as follows:

  if (mod(running-count([First name]),10)<=4) then ('Green') else ('White')

- **Application** bar: Run the report in **HTML**.

  A section of the results appear as follows:

First name	Last name	Work phone	Extension	Fax	Email	Green or White
Aaghie	Heiman	+(41) 17 12 13 11	9676	+(41) 17 12 61 82	AHeiman@grtd123.com	Green
Aaltje	Hansen	+(41) 17 12 13 11	9640	+(41) 17 12 61 82	AHansen@grtd123.com	Green
Abel	Antunes	+55 (11) 344-4444	2605	+55 (11) 333-2223	AAntunes@grtd123.com	Green
Abram	Ruiz	+(41) 17 12 13 11	9762	+(41) 17 12 61 82	ARuiz@grtd123.com	Green
Ada	Morales	+35 94 322 3540	7693	+35 91 548 1637	AMorales@grtd123.com	White
Adara	Cruz	+(39) 02 79 53 780	7704	+(39) 02 79 53 477	ACruz@grtd123.com	White
Adda	Heijman	+(41) 17 12 13 11	9639	+(41) 17 12 61 82	AHeijman@grtd123.com	White
Adelaide	Wiesinger	+(41) 17 12 13 11	9662	+(41) 17 12 61 82	AWiesinger@grtd123.com	White

- Close the rendered report tab.

Task 2. Create a conditional variable.

- **Navigate** tab: Navigate to **Page1**.
- **List data container**: Click **First name** list column body.
- **Properties** pane: For the **List columns body**, open the **Style variable** property dialog.
- **Style variable** dialog box: Add a new string variable named **RowColor**.
- **Conditionally author for these values** box: Add a new value named **Green**.
- **Expression Definition** dialog: Create and validate the following expression: **[Contact Information].[Green or White]**.

Task 3. Apply conditional formatting, and then run the report.

- **Properties** pane: From the **Select Ancestor** button click **List columns body style.**
  - From the **CONDITIONAL** section, double-click **Style variable**.
  - Under **Variable**, select **RowColor** from the list.
- Condition **explorer**: Click the **Green** value.
- **Properties** pane: Set the **Background color** to **#CCFFCC** (Color Swatch tab, three rows up, in the second column from the right).
- **Navigate** tab: Navigate to **Page1**.

- **On demand toolbar/More**: Cut the **<Green or White>** list column body.
- Click the list **Container selector** to select the entire list.
- **Properties** pane: Add the **Green or White** data item to the **DATA/Properties** property.
- **Application** bar: Run the report in **HTML**.

  The results appear as follows:

First name	Last name	Work phone	Extension	Fax	Email
Aaghie	Heiman	+(41) 17 12 13 11	9676	+(41) 17 12 61 82	AHeiman@grtd123.com
Aaltje	Hansen	+(41) 17 12 13 11	9640	+(41) 17 12 61 82	AHansen@grtd123.com
Abel	Antunes	+55 (11) 344-4444	2605	+55 (11) 333-2223	AAntunes@grtd123.com
Abram	Ruiz	+(41) 17 12 13 11	9762	+(41) 17 12 61 82	ARuiz@grtd123.com
Ada	Morales	+35 94 322 3540	7693	+35 91 548 1637	AMorales@grtd123.com
Adara	Cruz	+(39) 02 79 53 780	7704	+(39) 02 79 53 477	ACruz@grtd123.com
Adda	Heijman	+(41) 17 12 13 11	9639	+(41) 17 12 61 82	AHeijman@grtd123.com
Adelaide	Wiesinger	+(41) 17 12 13 11	9662	+(41) 17 12 61 82	AWiesinger@grtd123.com
Adeline	Arnaud	+33 1 68 94 52 20	3260	+33 1 68 94 56 60	AArnaud@grtd123.com
Adelma	Ortiz	+(39) 02 79 53 780	7705	+(39) 02 79 53 477	AOrtiz@grtd123.com
Adriaantje	Haanraads	+(41) 17 12 13 11	9638	+(41) 17 12 61 82	AHaanraads@grtd123.com
Adriana	Iacobucci	+(41) 17 12 13 11	9691	+(41) 17 12 61 82	AIacobucci@grtd123.com
Adrien	Martin	+32 16 20.73.21	3354	+32 16 20.73.32	AMartin1@grtd123.com
Adrienne	Roche	+33 1 68 94 52 20	3724	+33 1 68 94 56 60	ARoche@grtd123.com
Aert	Haak	+(41) 17 12 13 11	9787	+(41) 17 12 61 82	AHaak@grtd123.com
Aert	Meyer	+31 (0)20 692 93 94	6587	+31 (0)20 692 93 06	AMeyer@grtd123.com
Agatha	Reyes	+35 94 322 3540	7692	+35 91 548 1637	AReyes@grtd123.com
Agathe	Roque	+32 16 20.73.21	3715	+32 16 20.73.32	ARoque@grtd123.com
Agnelo	Chavez	+(41) 17 12 13 11	9761	+(41) 17 12 61 82	AChavez@grtd123.com

  You have developed a report containing contact information for each employee. You accomplished this by alternating sections of green and white rows in the resulting list.

- Close the rendered report tab.
- Remove the report without saving it.
- Sign out of the **IBM Cognos Analytics** portal.
- Close the web browser.

IBM Training

Examine the report specification

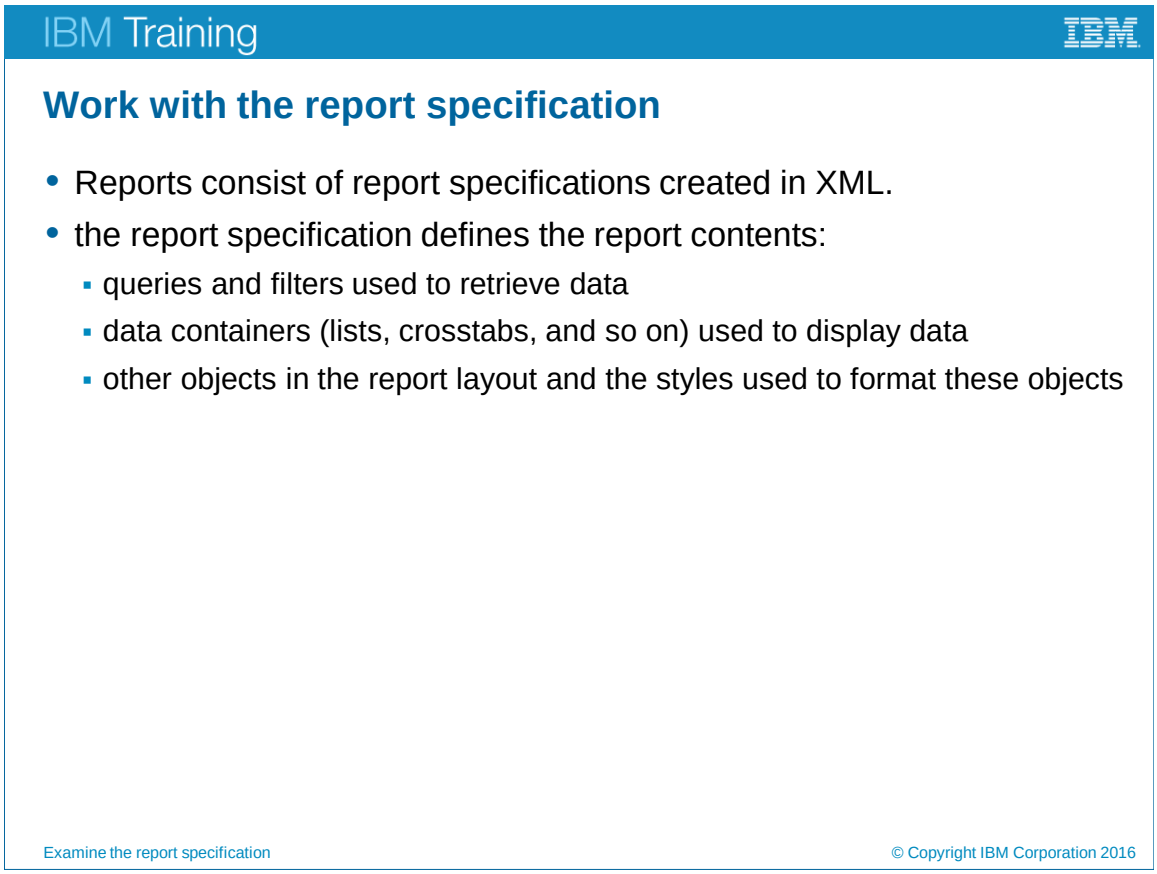
IBM Cognos Analytics (v11.0)

IBM Training

Unit objectives

- Examine the report specification structure
- Modify a report specification
- Add custom toolbox objects and custom template options

Unit objectives



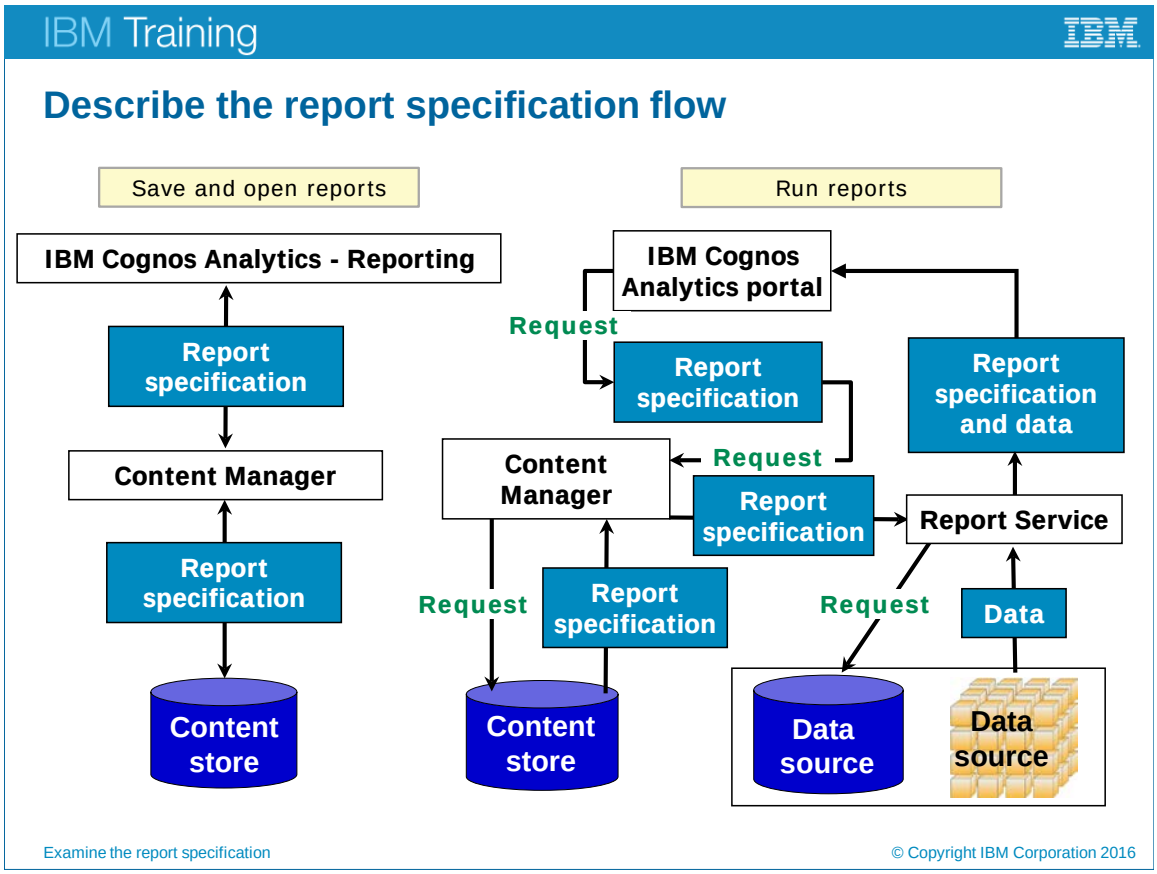
Work with the report specification

When you create a report, you are actually creating a report specification. The report specification defines the queries and prompts that are used to retrieve data and the layouts and styles used to present the data.

You can use the specification for a report to:

- view the XML code to better understand report contents
- modify a report by editing the XML code in the report specification
- share report specifications with report authors working in different environments
- copy the specification code for a specific report object such as an image or a page header and then reuse this code to add this object to other specification files

You can create a report specification entirely in XML, independent of the user interface and then open the report in IBM Cognos Analytics - Reporting. While the report specification can appear complex, It is possible to work with and modify the code with a basic understanding of how it works.



Describe the report specification flow

When you create a report, the specification is created on the local client machine. When you save a report, IBM Cognos Analytics - Reporting sends the specification to Content Manager. Content Manager then stores the specification in the content store.

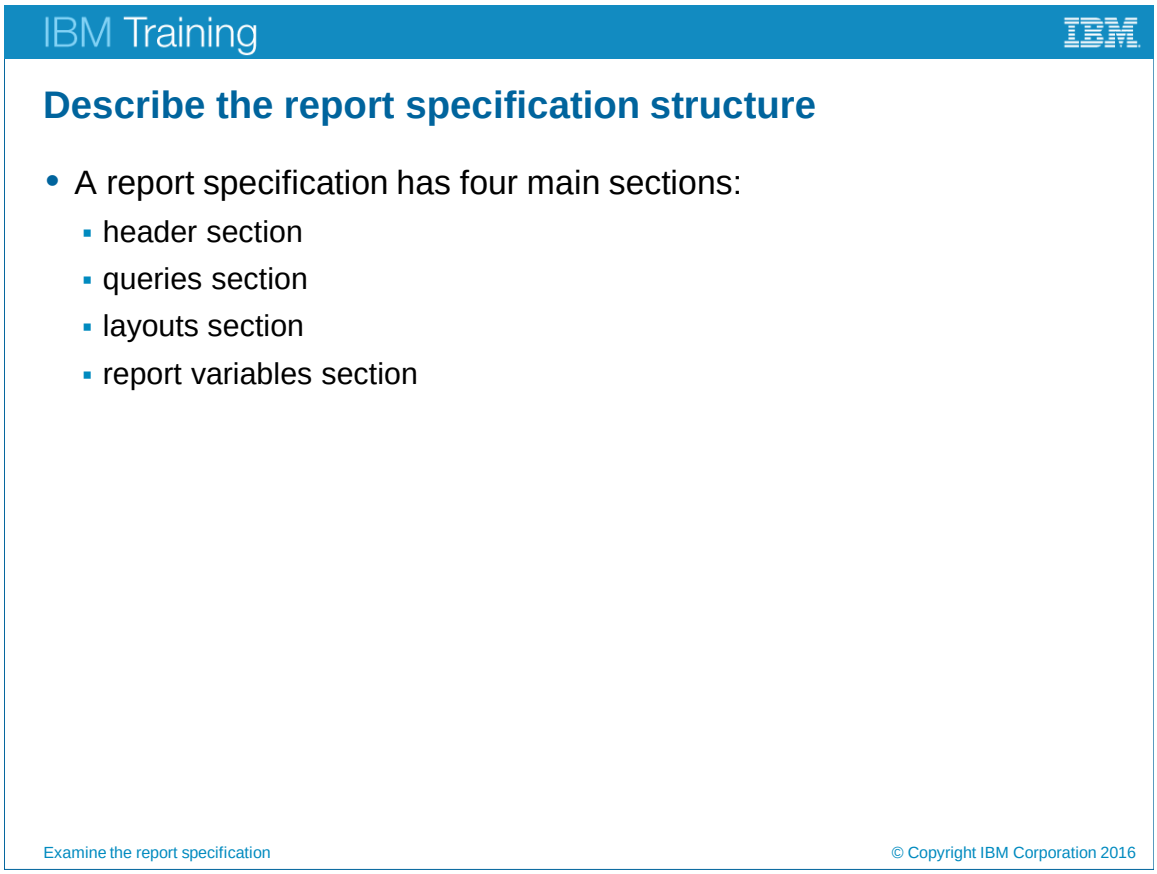
When you run a report, the request is sent in two parts:

1. To the Content Manager to retrieve the report specification from the content store.
2. To the database to retrieve the report data.

Once data is returned from the database, and the report specification (or report spec) is returned from Content Manager, the two are displayed in IBM Cognos Viewer by the report service.

If you edit a previously-saved report, changes to the report specification are applied only on your client machine until you save the report. Once saved, the specification is updated in the content store.

When you run a report in IBM Cognos Analytics - Reporting, Content Manager does not retrieve the specification. Because the specification is already open, IBM Cognos Analytics - Reporting can send the specification.



Describe the report specification structure

Depending on the report, the report specification may not include all of these sections.

IBM Training

Demonstration 1

Examine a report specification and copy it to the clipboard

Examine the report specification

Demonstration 1: Examine a report specification and copy it to the clipboard

Demonstration 1: Examine a report specification and copy it to the clipboard

Purpose:
To learn more about report specifications, you want to create a simple list report and view the XML specification. You will copy this report specification to the clipboard, and then re-open the specification in IBM Cognos Analytics - Reporting.

Portal: **http://localhost:88/ibmcognos**
User/Password: **brettonf/Education1**
Application: **IBM Cognos Analytics - Reporting**
Package: **Team content\Samples\Models\GO data warehouse (query)**
Report Type: **List**
Folder: **Sales and Marketing (query)**
Namespace: **Sales (query)**

Task 1. Create a list report.

1. Open a new **List** template using the **GO data warehouse (query)** package.
2. From the **Data/Source** tab, add the following query items to the list data container:
   - Order method: **Order method type**
   - Sales fact: **Revenue**

Order method type	Revenue
<Order method type>	<Revenue>

3. Run the Report in **HTML**.
   The result appears as follows:

Order method type	Revenue
E-mail	179,843,044.16
Fax	70,073,542.01
Mail	46,091,338.97
Sales visit	310,194,834
Special	27,351,320.25
Telephone	340,985,781.06
Web	3,712,235,908.4

4. Close the rendered report tab.

Task 2. View the report specification.

1. On the **Navigate** tab, right-click **Report**, and then click **Show Specification**.

   The Report Specification for the report displays in XML. The browser window includes controls to open and close the XML elements. IBM Cognos Analytics - Reporting does not let you copy the code displayed in this window.

   This report specification contains multiple sections: the specification header section, the queries section, the layouts section, and the XMLAttributes section.

   The header section specifies the namespace the report uses as well as the language and package.

```
- <report xmlns="http://developer.cognos.com/schemas/report/13.0/" useStyleVersion="10"
expressionLocale="en-us">
    <modelPath type="FM">/content/folder[@name='Samples']/folder
  [@name='Models']/package[@name='GO data warehouse (query)']/model
  [@name='model']</modelPath>
    <drillBehavior modelBasedDrillThru="true"/>
```

   The queries section contains information about the query used to retrieve data for the list report (as seen in the <queries> section).

   The layouts section describes the objects that appear on Page1 of the report (as seen in the <layouts> section).

   The XMLAttributes section provides additional information about the XML elements used in the report specification.

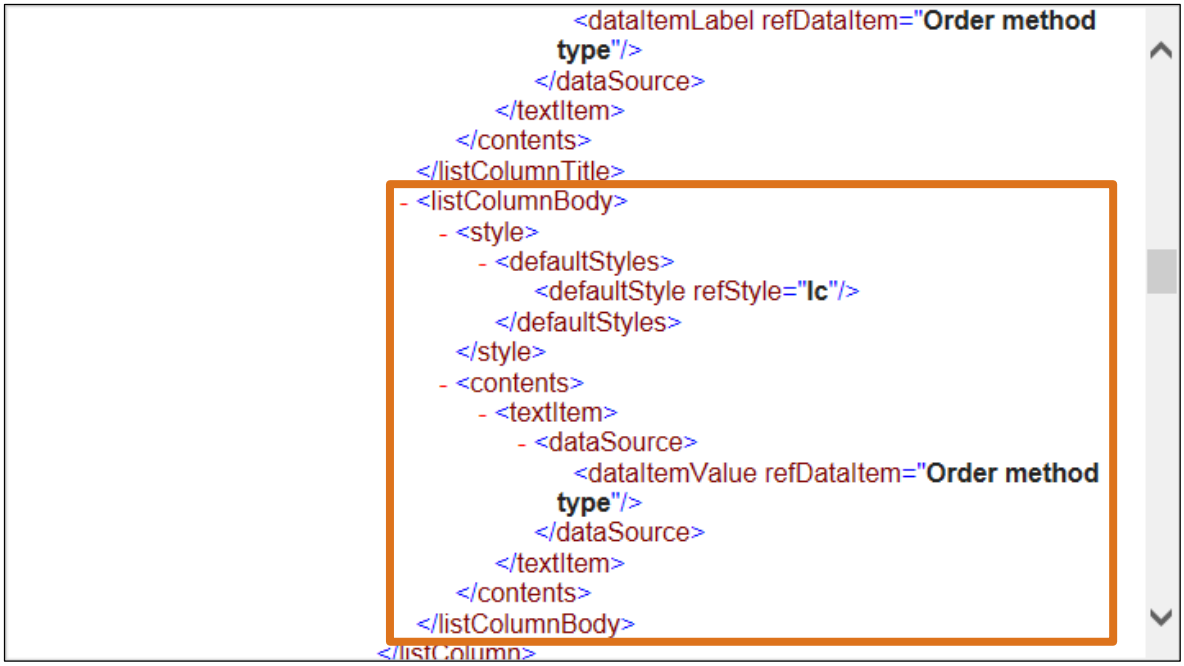
2. In the specification, click the **-** to the left of **<queries>**, to collapse the **<queries>** tag.

   The window within the dialog does not allow direct copy and paste of its content, however, the dialog box does allows you to copy the entire content via the Copy to clipboard button  , found in the bottom left corner.

3. Scroll down to the **<listColumnBody>** section.

   The specification for the Order method type column body displays.

   The details of the Order method type column body specify that this object is formatted using the "lc" style (list column) and that the data source for this column body is the Order method type data item.



4. Close the Report specification XML window.

Task 3. Copy the report specification and open the report from the clipboard.

1. From the **Page explorer** tab, right-click **Report**, and then click **Copy Report to Clipboard**.
2. From the **Side panel**, click **New**, and then click **Report**.
3. Double-click **Crosstab**, and then click **OK** without identifying a report package.

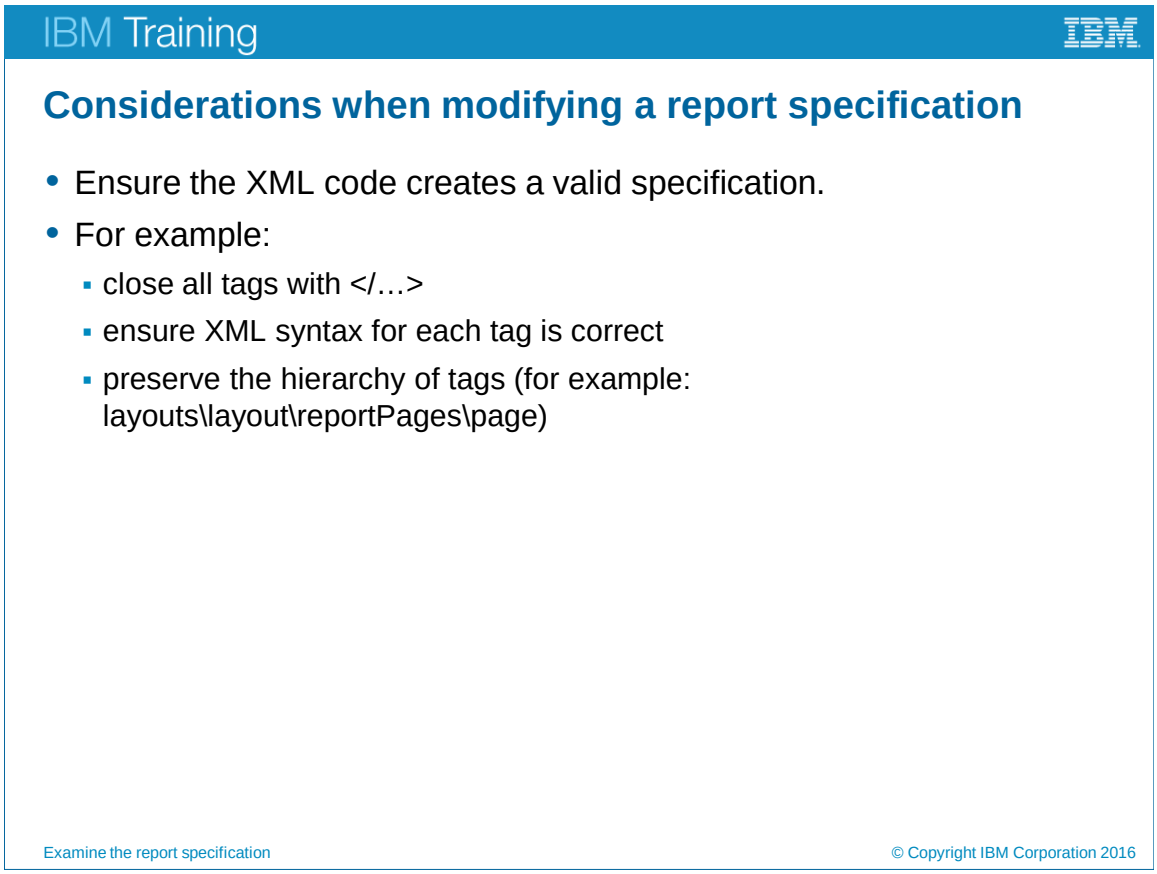
4. From the **Page explorer** tab, right-click **Report**, and then click **Open Report from Clipboard**.

   IBM Cognos Analytics - Reporting uses the specification you copied to the clipboard, to open the report you created earlier in this demonstration.

   A section of the result appears as follows:

5. Remove any reports created during this demonstration, without saving.
6. Leave the **IBM Cognos Analytics** portal open for the next demonstration.

Results:
You created a simple list report and viewed its XML specification. You then copied the report specification to the clipboard and re-opened it in IBM Cognos Analytics - Reporting.



Considerations when modifying a report specification

Because some XML editors, such as Notepad, do not validate the report specification, it is important to use the correct syntax. Otherwise, when you edit a report specification and then run the report in IBM Cognos Analytics - Reporting, you will receive an error message.

The error message describes the problems that were identified and need to be corrected.

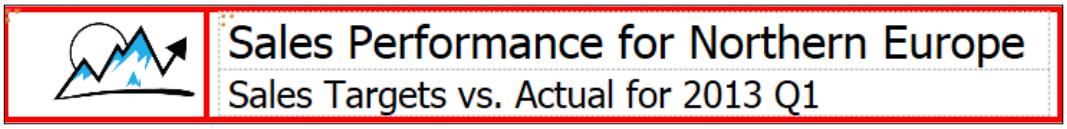
If the report specification is written in a language that is not ANSI-supported, you should save the specification file with UTF-8 encoding.

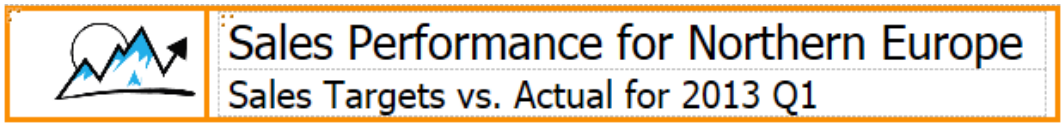
To edit a report in an environment that lets you validate report syntax and structure, edit the report in IBM Cognos Analytics - Reporting instead of working directly with the report specification.

IBM Training

Demonstration 2

Modify a report specification





Examine the report specification

Demonstration 2: Modify a report specification

Demonstration 2: Modify a report specification

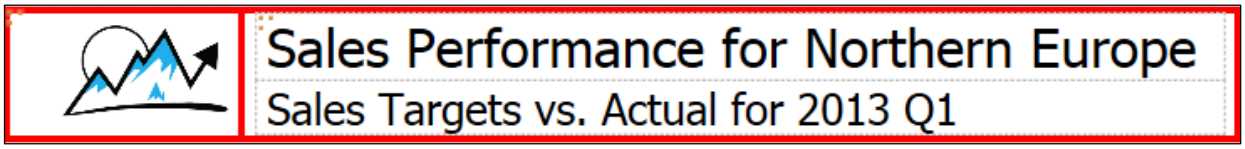
Purpose:
You want to update a report header to reflect new branding changes. To do this most efficiently, you will modify the report specification.

Portal: **http://localhost:88/ibmcognos**
User/Password: **brettonf/Education1**
Application: **IBM Cognos Analytics - Reporting**

Task 1. Open a report and copy the report specification.

1. From the **IBM Cognos Analytics** portal, navigate to **Team content\ Legacy Samples\Samples\Models\GO Data Warehouse (query)\Report Studio Report Samples**.
2. Point to the **Bursted Sales Performance Report** to expose the **More** button.
3. Click the **More** button, and then click **Edit report**.
4. On the **Application** bar, open the **Properties** pane.
5. Click on the far right space of the title table, from the **Properties** pane, click **Select Ancestor**, and then click **Table**.
6. On the **Properties** pane, under **BOX**, double-click **Border**.
7. Set the **Width** to **2¼**, the **Color** to **Red**, click **Apply All Borders**, and then click **OK**.
8. Repeat steps **6** and **7** to apply a border to the logo cell.

   The results appear as follows:



Management has decided that the orange found in the list is more in line with corporate colors and would like you to change the border accordingly. You could select each object change the color manually, but we will explore how to change both objects at one time by modifying the report specification with this change. You will copy the report specification to the clipboard, and paste it into a text editor to make the changes.

9. From the **Navigate** tab, right-click **Report**, and then click **Copy Report to Clipboard**.
10. From the Windows **Start** menu, launch **Notepad**.

    If Notepad is not available directly from the Start menu, click All Programs\Accessories, and then click Notepad.
11. From the **Edit** menu, click **Paste**, to paste the report specification from your clipboard.
12. Press Ctrl+Home to place the cursor at the beginning of the specification.

Task 2. Modify the report specification.

You want to replace the red with orange (F78E06). There is no other red in the report, so you can replace all occurrences of red in your specification. If there were some instances of red to remain in the report, you could do a search for each instance and decide what to change, based on the element in the report specification.

1. From the **Edit** menu, click **Replace**.
2. In the **Find what** box, type **red**, in the **Replace with** box type **#F78E06**, and then click **Replace All**.

   #F78E06 is the hex code for the color used in the report.
3. Close the **Replace** dialog box.

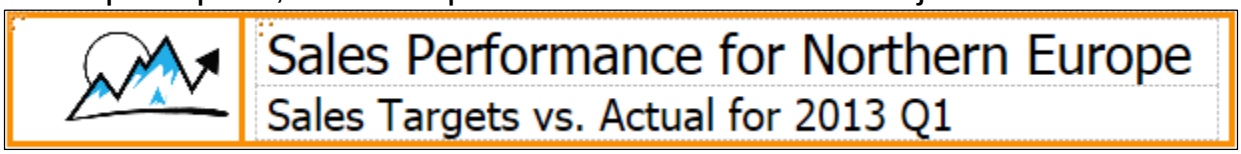
   If you had further modifications to the report specification, you could do them now, such as changing the header text or other elements. Ensure that the replacement syntax that you use is valid. At this time, you will incorporate only the change from red borders to orange borders.
4. From the **Edit** menu, click **Select All**, and then with the specification selected, from the **Edit** menu, click **Copy**.

   The modified specification is copied to your clipboard.

Task 3. Open the updated specification from the clipboard.

1. From the **Page explorer** tab, right-click **Report**, and then click **Open Report from Clipboard**.
2. Do not save changes to the currently open report if prompted.

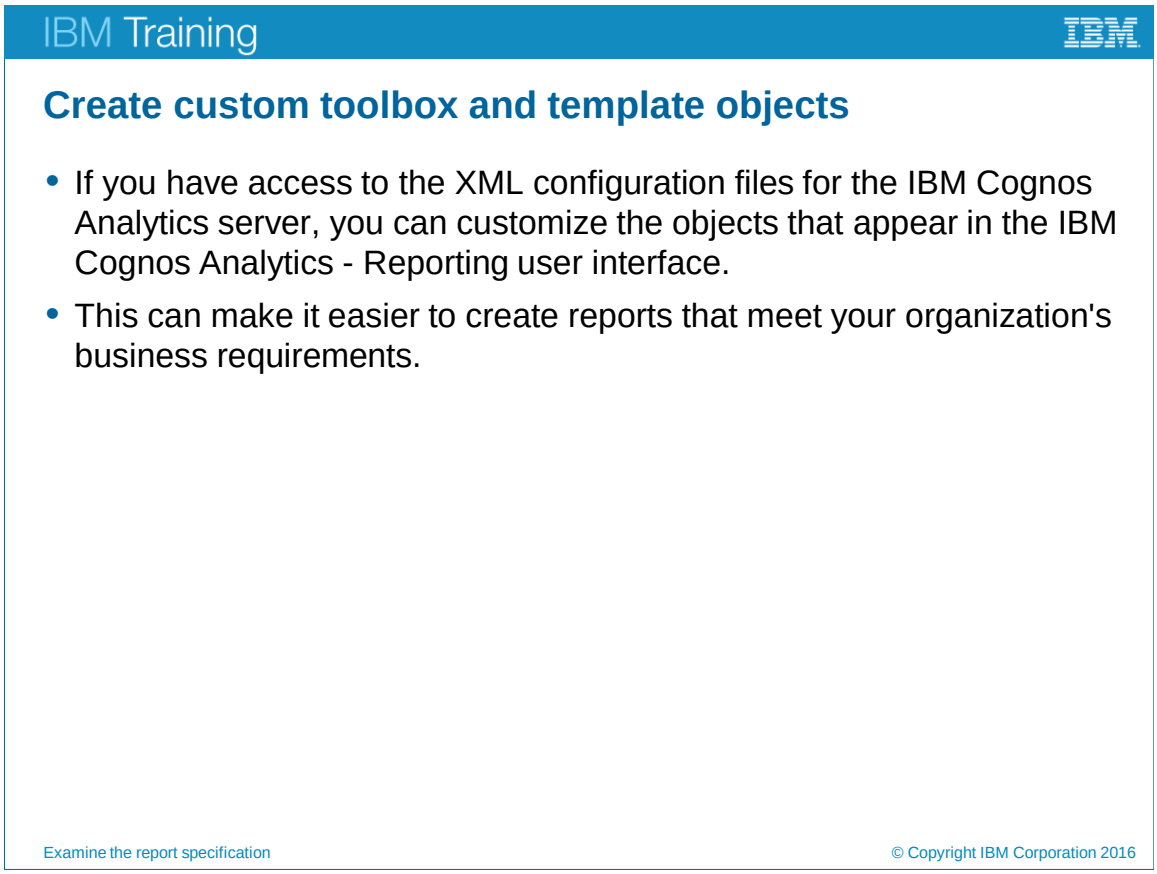
   The report opens, with the updated border color on the objects in the header.



You modified the report specification to efficiently update the report.

3. Close **Notepad** without saving changes.
4. Remove the report without saving it.
5. Leave the **IBM Cognos Analytics** portal open for the exercise.

Results:
You updated a report header to reflect new branding changes by modifying the report specification.



Create custom toolbox and template objects

Once added to the IBM Cognos Analytics portal, custom templates and toolbox objects are available to all users with access to IBM Cognos Analytics - Reporting.

Authors might not have been granted access to the server-side configuration XML files used to create custom objects, and therefore may need to collaborate with the server administrator to implement this functionality.

Because you add custom objects by manually editing server-side configuration files, your changes may not be preserved during maintenance or version upgrades.

It is recommended that you back up these configuration files before and after you edit them.

Examples of customization:

- if the IBM Cognos Analytics - Reporting Toolbox tab does not contain the objects you require, you can add or customize the objects available on this tab
- if the existing IBM Cognos Analytics - Reporting template options do not meet your requirements, you can add or customize the template options available when creating new reports

IBM Training

Specify how a custom toolbox object will appear in a report

- To add a new toolbox object, you must first add XML code to the ToolboxControls.xml file.
- This code specifies how the object will appear and behave once it is added to a report from the Toolbox tab.

Code specifying the appearance and behavior of the List and Image objects

```
<!-- List -->
<list id="list" horizontalPagination="true">
        <listColumnBodyStyle/>
        <listColumnTitleStyle/>
        <listColumnStyle/>
        <style>
                <CSS value="border-collapse:collapse"/>
                <defaultStyles>
                        <defaultStyle refStyle="ls"/>
                </defaultStyles>
        </style>
```

```
<!-- Image -->
<image id="image">
        <dataSource>
                <staticValue/>
        </dataSource>
</image>
```

Examine the report specification

Specify how a custom toolbox object will appear in a report

When you create the object in the ToolboxControls.xml file, specify object properties such as the CSS styles used to format the object and data sources used to create the object.

To obtain the XML code used to create an object, you can create the object in IBM Cognos Analytics - Reporting and then copy the specification for this object to the clipboard.

In the slide example:

- the class attribute in the list tag uses list styles from the GlobalReportStyles.css file (<IBM Cognos Analytics Installation Location>\Program Files\IBM\cognos\Analytics\bin) to format this object
- the specification for the Image object indicates that a static value specified by the user will act as a data source for the image
- The XML files are found at <IBM Cognos Analytics Installation Location>\Program Files\IBM\cognos\analytics\webcontent\pat\res

IBM Training

Specify how custom objects will appear on the toolbox tab

- To inform IBM Cognos Analytics - Reporting where to find the definition for the new toolbox object, you must add XML code to the Toolbox.xml file.
- This code will point to the object definition you added to the ToolboxControls.xml file.

Reference to the Text Item object defined in the ToolboxControls.xml file

```
        <listView id="Toolbox_PagesView" classPrefix="clsListItem_tb">
                <listItems>
                        <listItem controlRef="page" idsLabel="IDS_EL_page"
idsTooltip="IDS_EL_page" smallIcon="page.gif"/>
                        <listItem controlRef="pageSet" idsLabel="IDS_EL_pageSet"
idsTooltip="IDS_EL_pageSet" smallIcon="pageSet.gif"/>
                </listItems>
        </listView>
```

Examine the report specification

Specify how custom objects will appear on the toolbox tab

The Toolbox.xml file references the various toolbox objects. It also specifies how the object will appear on the toolbox tab, through properties such as the label, the tooltip, and the icon image.

The Toolbox.xml file has different sections. Each section specifies the toolbox objects that appear for a specific toolbox view in IBM Cognos Analytics - Reporting.

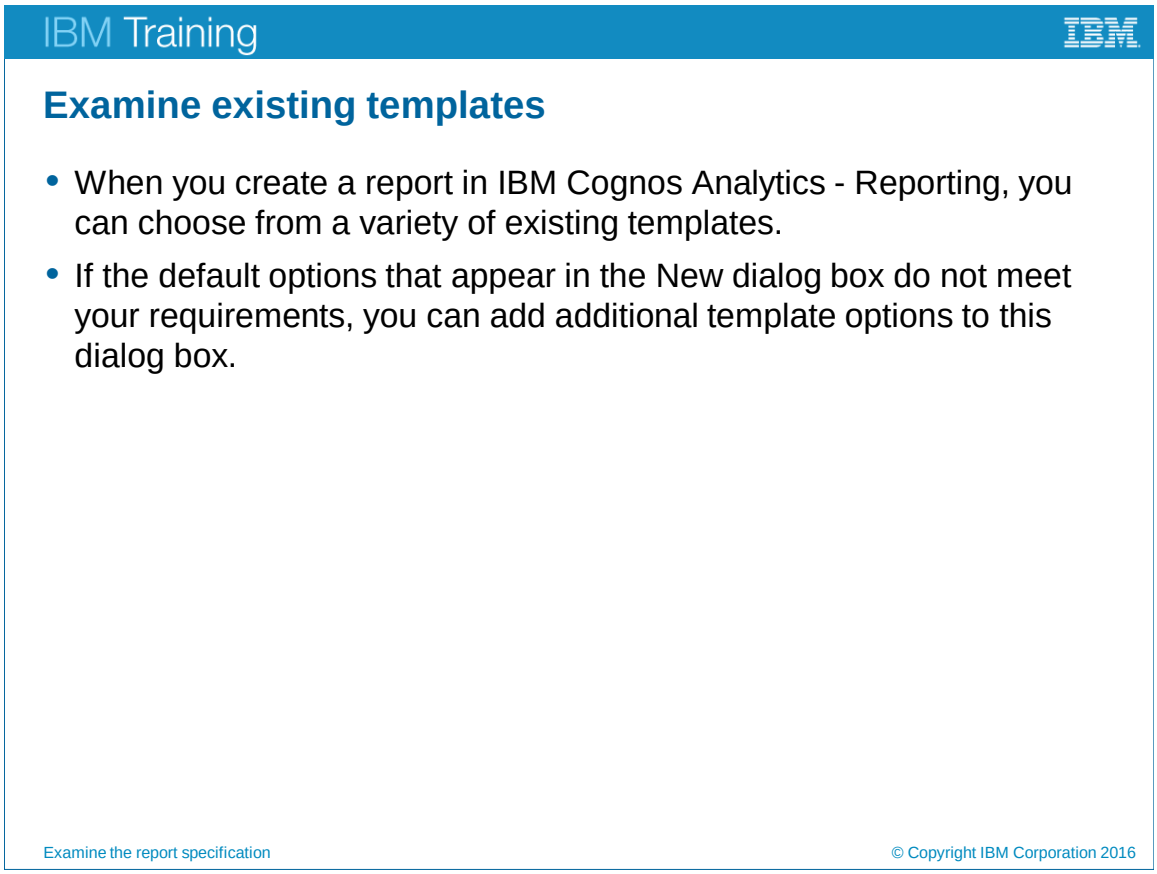
In the Toolbox.xml file, use a controlRef attribute to point to the object definition you added to the ToolboxControls.xml file; the attribute must be named the same in both files.

The id attributes for each Toolbox.xml view are predetermined. You cannot add your own view id attributes.

It is important to add this code to the appropriate section of the Toolbox file so that the object appears in the appropriate toolbox view.

For example, the objects listed in the LayoutView section of the Toolbox.xml file will appear on the Toolbox tab only when report authors are working on the layout of report pages.

Re-sorting the Toolbox and Template icons can be done by manually changing the order in the respective XML files.



Examine existing templates

To add a new template option to the dialog box:

- specify the contents of the template
- point to this specification
- add a graphic that will represent this template option
- add a label for the template

IBM Training

Specify the custom template contents

- To add a template:
  - add XML code defining the template to the templates.xml file
  - add code to the Resources.xml file to identify the template to IBM Cognos Analytics - Reporting

```
<template name="Blank">
        <report expressionLocale="en-us">
                <modelPath/>
                <drillBehavior modelBasedDrillThru="true"/>
                <layouts>
                  <layout>
                     <reportPages>
                        <page>
                          <style>
                            <defaultStyles>
                                <defaultStyle refStyle="pg"/>
                            </defaultStyles>
                          </style>
```

Code used to define the Blank Template in the templates.xml file

Examine the report specification

Specify the custom template contents

The templates defined in the templates.xml file are report specifications.

For best results, create the template you require in IBM Cognos Analytics - Reporting, and then copy the XML specification for the report and paste it into the templates.xml file.

IBM Training

Unit summary

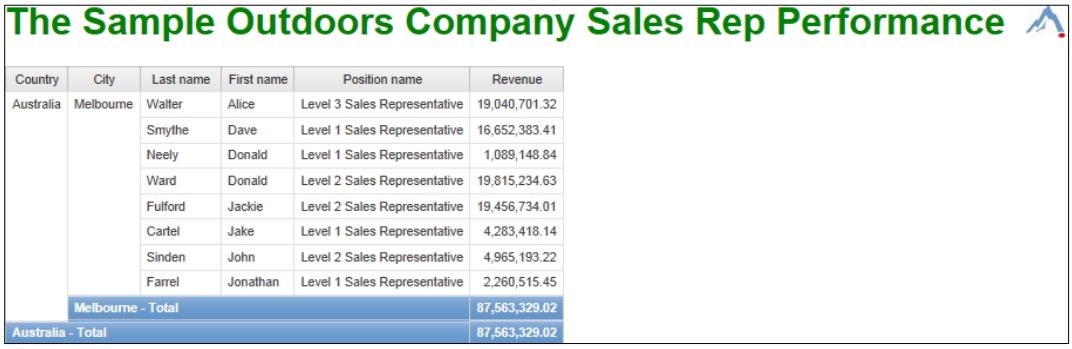
- Examine the report specification structure
- Modify a report specification
- Add custom toolbox objects and custom template options

Unit summary

IBM Training

Exercise 1

Update a report layout by importing new specification code



The Sample Outdoors Company Sales Rep Performance

Country	City	Last name	First name	Position name	Revenue
Australia	Melbourne	Walter	Alice	Level 3 Sales Representative	19,040,701.32
		Smythe	Dave	Level 1 Sales Representative	16,652,383.41
		Neely	Donald	Level 1 Sales Representative	1,089,148.84
		Ward	Donald	Level 2 Sales Representative	19,815,234.63
		Fulford	Jackie	Level 2 Sales Representative	19,456,734.01
		Cartel	Jake	Level 1 Sales Representative	4,283,418.14
		Sinden	John	Level 2 Sales Representative	4,965,193.22
		Farrel	Jonathan	Level 1 Sales Representative	2,260,515.45
	Melbourne - Total				87,563,329.02
Australia - Total					87,563,329.02

Examine the report specification

Exercise 1: Update a report layout by importing new specification code

Exercise 1: Update a report layout by importing new specification code

Note: Use the Edit menu when copying or pasting in Notepad.

As Frank Bretton, a report author, you have been asked to create a new basic report template for The Sample Outdoors Company that authors can use to standardize specific report headers.
In the report specification, the sections of code that describe the page header and footer are grouped together. Therefore, authors can copy the necessary block of code from the new standard template specification and replace the old header and/or footer code in the report specification to update their own report layouts.
You will test the new template on a report you are creating for the sales manager on sales rep performance, by introducing the header from the template report specification into the sales report specification.
To accomplish this:

- Create a blank report using the GO data warehouse (query) package, with a header that has a two column, 1 row table. Insert a company graphic (go_logo_small.jpg) in the right cell, aligned to the right with a height of 50 pixels. Include a text block in the left table cell with the company name The Samples Outdoor Company in size 24 pt. font that is bold and green. Review the report specification.
- Save the report as a template in the portal in the B6059 folder, so that other authors can access it. Review the report specification again to ensure that template has been included in the code.
- Create a new list report called Sales Rep Performance with the following items:
  - Employee by region: Country, City, Last name, First name, Position name
  - Sales fact: Revenue
- Group the Country and City columns and display Revenue totals. Save the report as Sales Rep Performance in the B6059 folder, and then run the report to view the results. Copy the report specification to the clipboard, and paste it in a text editor, and then review the code, specifically the header and footer sections.
- In IBM Cognos Analytics - Reporting, open the corporate template, and copy the report to the clipboard.
- Open a new instance of a text editor (note: use the Edit menu when copying or pasting in Notepad.), and paste the contents into it. Select the code for the header, copy it to the clipboard, and then close this instance of the text editor without saving changes.

- Open the first instance of the text editor, and replace the header section with the header section in the clipboard to add the corporate template header into your report specification. Modify the title in the header to include the report title, Sales Rep Performance, after the company header text. Select the entire report specification, and then copy it to the clipboard.
- In IBM Cognos Analytics - Reporting, open the report from the clipboard, and review the results.

For more information about where to work and the exercise results, refer to the Tasks and results section that follows. If you need more information to complete a task, refer to earlier demonstrations for detailed steps.

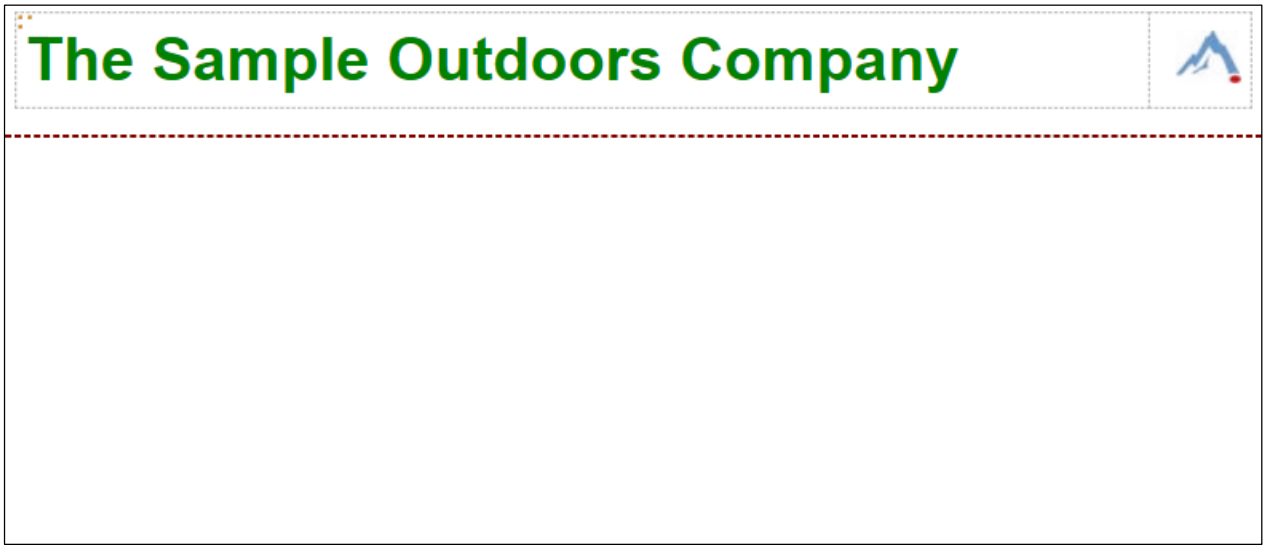
Exercise 1: Tasks and results

Task 1. Create a report and review the report specification.

- **Side panel**: Open a new **List** report template using the package. **GO data warehouse (query)**
- **Page Header**: Click the **Header** block and then delete.
- **Work area**: Select the entire list data container and then delete it.
- **Page Header**: Click the **Add** button and then click **Table**.
  - Click **OK** to add a table with the current defaults.
- **Toolbox** tab: Under **LAYOUT**, add an **Image** into the right table cell, using **go_logo_small.jpg**.

  Set the URL address to http://vclassbase:88/images/ if necessary.
- **Properties** pane: For the right cell, set the horizontal alignment to **Right** and the **Size & vverflow/Height** to **50px**.
- **Toolbox tab**: Add the following **Text item** object to the left table cell: **The Sample Outdoors Company**.
- **On demand toolbar**: Apply the following formatting to the text:
  - Font size: **24 pt.**
  - Weight**: Bold**
  - Foreground color: **Green**

  The results appear as follows:



- **Page explorer** tab: Show the report specification.

  A section of the result appears as follows.

```
- <report xmlns="http://developer.cognos.com/schemas/report/13.0/" useStyleVersion="10"
expressionLocale="en-us">
    <modelPath type="FM">/content/folder[@name='Samples']/folder
   [@name='Models']/package[@name='GO data warehouse (query)']/model
   [@name='model']</modelPath>
    <drillBehavior modelBasedDrillThru="true"/>
```

- Close the **Report Specification** dialog box.

Task 2. Save the report as a template.

- **Application** bar**\Save** menu: Save the report as a **Query Studio template**.
- **Page explorer** tab: Show the report specification XML.

  A section of the result appears as follows. Note the addition of the **template="true"**.

```
- <report xmlns="http://developer.cognos.com/schemas/report/13.0/" useStyleVersion="10"
expressionLocale="en-us" template="true">
    <drillBehavior modelBasedDrillThru="true"/>
```

- Close the **Report specification XML** dialog box.
- **Save** menu**:** Save the template in **Team content\B6059** as **Corporate Template - Green**.

Task 3. Create a report, and review the report specification.

- **Side panel**: Open a new **List** report template using the **GO data warehouse (query)** package.
- **Source** tab: Navigate to **Sales and Marketing (query)/Sales (query)** and then add the following query items to the list data container:
  - Employee by region: **Country**, **City**, **Last name**, **First name**, and **Position name**.
  - Sales fact: **Revenue**.
- **On demand toolbar**: Group the **<Country>** and **<City>** list column bodies.
  - Summarize the**< Revenue>** list column body by **Total**.
- **Work area**: Create the following report title: **Sales Rep Performance**

- **On demand toolbar**: Left justify the title block object.
  - Save the report in **Team content\B6059** as **Sales Rep Performance**.
  - Run the report in **HTML**.

  A section of the report appears as follows:

Sales Rep Performance

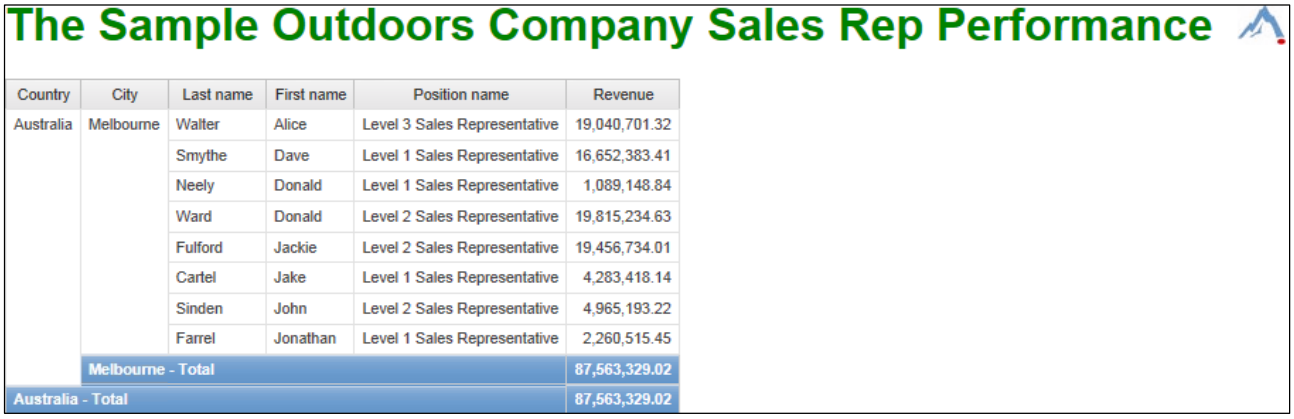
Country	City	Last name	First name	Position name	Revenue
Australia	Melbourne	Walter	Alice	Level 3 Sales Representative	19,040,701.32
		Smythe	Dave	Level 1 Sales Representative	16,652,383.41
		Neely	Donald	Level 1 Sales Representative	1,089,148.84
		Ward	Donald	Level 2 Sales Representative	19,815,234.63
		Fulford	Jackie	Level 2 Sales Representative	19,456,734.01
		Cartel	Jake	Level 1 Sales Representative	4,283,418.14
		Sinden	John	Level 2 Sales Representative	4,965,193.22
		Farrel	Jonathan	Level 1 Sales Representative	2,260,515.45
	Melbourne - Total				87,563,329.02
Australia - Total					87,563,329.02

- Close the rendered report tab.
- **Navigate** tab: Copy the report to the clipboard.
- **Desktop**: Launch **Notepad**.
- **Notepad**: Paste the contents of the clipboard to the work area, and then examine the report specification.

Task 4. Replace the header in the report specification.

- **Application** tab: Click **Edit**.
- **Side panel**: click **Team content/B6059**.
- **Corporate Template - Green**: Under **More**, click **Edit report**
- **Navigate** tab: Copy the report to the clipboard.
- **Desktop**: Launch another session of **Notepad**.
- **Notepad/Edit** menu: Paste the contents of the clipboard to the work area, and then examine the report specification.
- **Notepad - Corporate Template Green/Edit** menu: Select and copy the text from **<pageHeader>** to **</pageHeader>** (including these two tags).
- Close this instance of **Notepad** without saving changes.
- **Notepad/Edit** menu: Select the text from **<pageHeader>** to **</pageHeader>** (including these two tags), and then paste from the clipboard.
  - In the **pageHeader** code, after **The Sample Outdoors Company** text, add a space, and then type: **Sales Rep Performance**.
- **Notepad/Edit** menu: Click **Select All**.
  - Click **Copy**.
- **IBM Cognos Analytics - Reporting Navigate** tab: Open the report from the clipboard.
- **Application** bar: Run the report in **HTML**.

  A section of the results appear as follows:



The Sample Outdoors Company Sales Rep Performance

Country	City	Last name	First name	Position name	Revenue
Australia	Melbourne	Walter	Alice	Level 3 Sales Representative	19,040,701.32
		Smythe	Dave	Level 1 Sales Representative	16,652,383.41
		Neely	Donald	Level 1 Sales Representative	1,089,148.84
		Ward	Donald	Level 2 Sales Representative	19,815,234.63
		Fulford	Jackie	Level 2 Sales Representative	19,456,734.01
		Cartel	Jake	Level 1 Sales Representative	4,283,418.14
		Sinden	John	Level 2 Sales Representative	4,965,193.22
		Farrel	Jonathan	Level 1 Sales Representative	2,260,515.45
	Melbourne - Total				87,563,329.02
Australia - Total					87,563,329.02

- Close the rendered report tab
- Close any reports without saving.
- Sign out of the **IBM Cognos Analytics** portal.
- Close the web browser.
- Close all open **Notepad** instances without saving.

IBM Training

Distribute reports through bursting

IBM Cognos Analytics (v11.0)

IBM Training

Unit objectives

- Distribute reports using bursting
- Create burst keys
- Identify report recipients and data items using burst tables
- Distribute reports using email and the IBM Cognos Analytics portal

Unit objectives

IBM Training

Burst reports

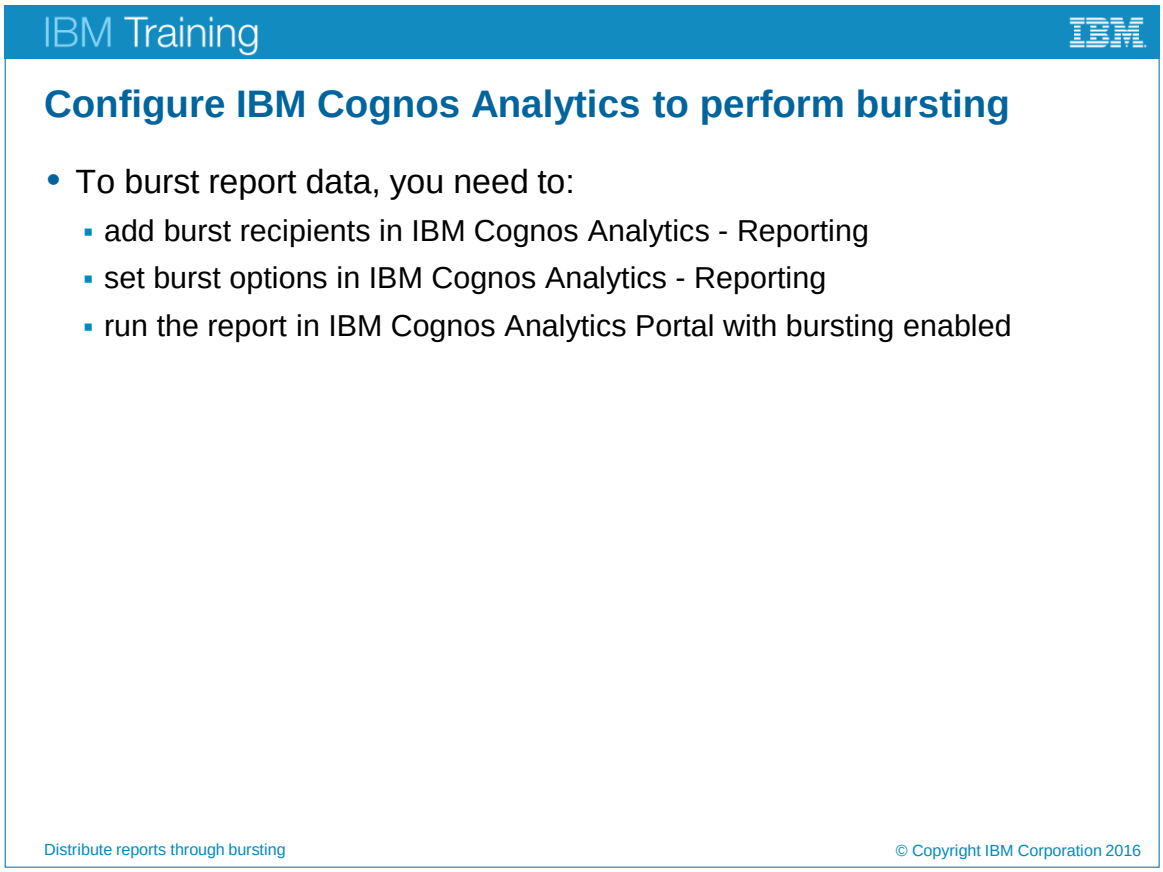
- When you burst a report, you run a report once and divide the results for distribution to multiple recipients.
- Each report recipient will only see the subset of data to which they have access to.

Distribute reports through bursting

Burst reports

While authoring a report, you can choose to burst to email addresses or to a directory (in the Burst options). IBM Cognos Analytics leverages the existing security infrastructure in the namespace to dynamically obtain email addresses. Your company may have sales people in different regions who require the sales target for each country; you can use burst reports to send each salesperson the information that they need.

Just as with regular reports, burst reports can be distributed in multiple formats (for example, PDF, XML, XLS and CSV).

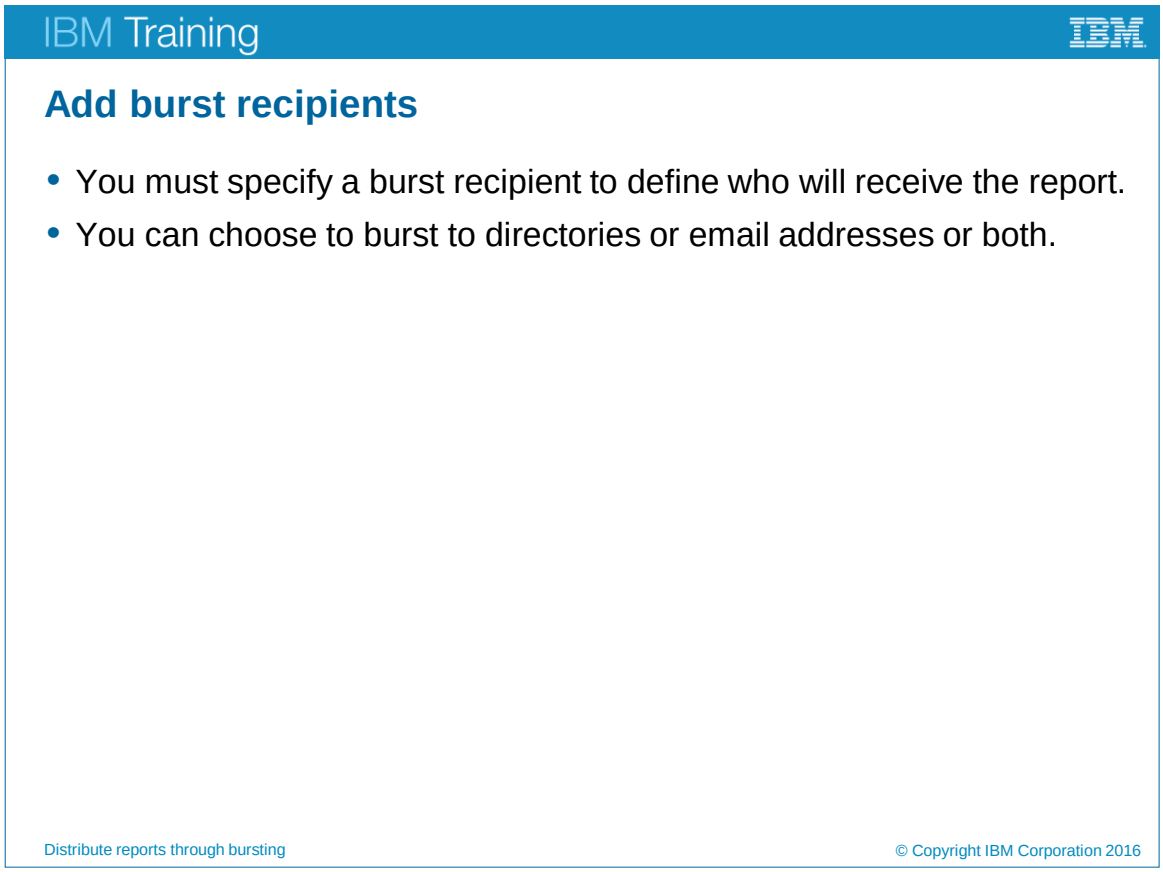


Configure IBM Cognos Analytics to perform bursting

The burst recipient is a data item in your report that determines who will receive the burst reports.

The data source can be an existing employee table or a custom burst table that you create.

In order to burst a report, you must add a data item to the report that tells IBM Cognos Analytics who the intended recipients are and how you intend to distribute the reports. A burst table maps recipients or groups of recipients to specific data that they are allowed to see and can be a separate table in your data source.



Add burst recipients

Burst recipients can be users, groups, roles, email contacts, or email distribution lists.

When you burst a report to e-mail, you can:

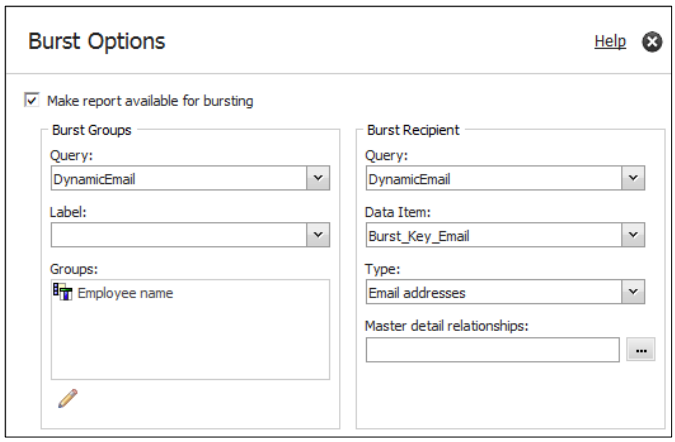
- reference the e-mail address in the authentication source
- use the e-mail address as it is stored in your data source
- supply the e-mail address in the report itself by creating a calculated field
- reference the appropriate burst table column

If the authentication provider does not contain email information, the email address can be manually entered in the IBM Cognos Analytics portal in the user's personal information. Privileges are set differently at each organization. By default, a report author does not have the necessary rights to make changes to the personal information of other users, in which case your administrator may need to do this for you.

IBM Training

Set burst options in IBM Cognos Analytics - Reporting

- Group the report on the data item to burst on
- Burst groups define how the data should be separated
- Burst recipient determines who receives the subset of data
- Burst key defines how the report is burst to its recipients



Distribute reports through bursting

Set burst options in IBM Cognos Analytics - Reporting

If you burst on a query item that is not a part of the layout, the same report will be distributed to all the burst recipients. This allows you to use the burst functionality to limit who receives the report, even when everyone who has access to the report sees the same thing. This is similar to drilling through with no parameters in the target.

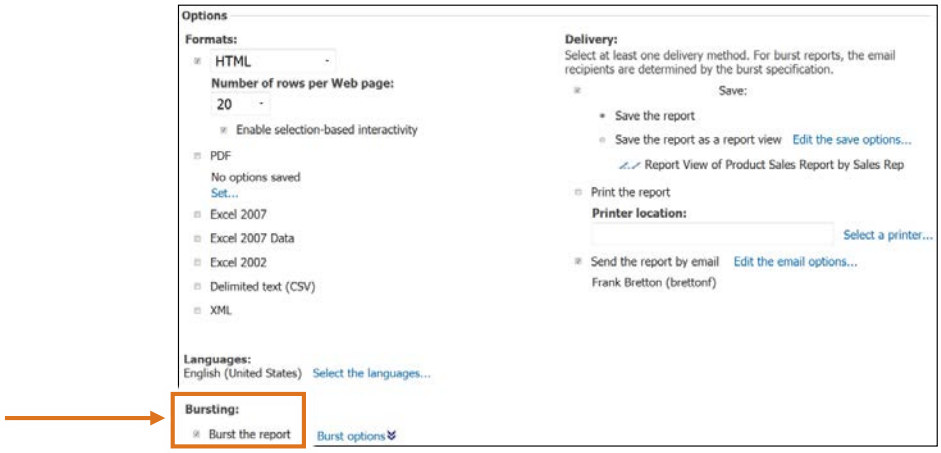
If you only want to e-mail the report to various recipients and not burst to the portal, your burst recipient must be an email address. If you want the report to be emailed as well as burst to the portal, you must specify your Burst Recipient Type to be Directory Entries.

The Label field in the Burst options dialog box determines how the reports are labeled in the IBM Cognos Analytics portal. For example, if you use a country data item as the label, the report will be named after the appropriate country. If you choose the burst key, the report will be named after the burst key value, such as CAMID(":Australia"). If no label is set, the default label is the data item that is being grouped.

IBM Training

Enable burst reports in IBM Cognos Analytics portal

- You can burst the report by running it from the IBM Cognos Analytics portal.



Enable burst reports in IBM Cognos Analytics portal

You can burst the report immediately or schedule it to run at a later date and time. If you intend to email the reports, you must select the Send the report by email option as well. When emailing the report, you have the option of sending the report as an attachment, sending a link to the report, or both.

To burst the report, run the report with options in IBM Cognos Analytics portal. In the advanced options, select the Burst the report check box. This box is only available if the report has the check box selected for Make available for bursting.

IBM Cognos lets administrators disable the ability to send reports as email attachments to ensure that only approved users receive reports. If this is the case, you may still send links to reports. Because these links are accessed through the IBM Cognos system, users must log on to IBM Cognos Analytics portal to access a report.

IBM Training

Demonstration 1

Burst a report to email recipients using a data item

Product line	Product type	Product	Quantity	Revenue
John Sinden				
Camping Equipment	Cooking Gear	TrailChef Canteen	1,351	16,414.65
		TrailChef Cook Set	1,845	97,748.85
		TrailChef Cup	1,788	6,347.4
		TrailChef Double Flame	578	83,336.04
		TrailChef Kettle	7,903	82,524.23
		TrailChef Kitchen Kit	1,759	41,019.88
		TrailChef Single Flame	816	51,758.88
		TrailChef Water Bag	8,197	51,313.22
	Cooking Gear - Total			430,463.15

Distribute reports through bursting

Demonstration 1: Burst a report to email recipients using a data item

Demonstration 1: Burst a report to email recipients using a data item

Purpose:
Three sales representatives at The Sample Outdoors Company need to know the total number of items that they have sold of each product line and type. However, employees need to see only data pertaining to their own sales. You will burst a product sales report so that each sales representative will see only his or her sales totals.

Note: In this demonstration, you will dynamically build an email burst key rather than reference a burst table.

Portal: **http://vclassbase:9300/bi**
User/Password: **brettonf/Education1**
Application: **IBM Cognos Analytics - Reporting**
Package: **Team content\Samples\Models\GO data warehouse (query)**
Report Type: **List**
Folder: **Sales and Marketing (query)**
Namespace: **Sales (query)**

Task 1. Create a report with product sales information.

1. Open a new **List** template using the **GO data warehouse (query)** package.
2. From the **Data/Source** tab, add the following query items to the list data container:
   - Employee by region: **Employee name**
   - Products: **Product line**, **Product type**, **Product**
   - Sales fact: **Quantity**, **Revenue**

Employee name	Product line	Product type	Product	Quantity	Revenue
<Employee name>	<Product line>	<Product type>	<Product>	<Quantity>	<Revenue>

3. Select the **<Employee name>**, **<Product line>** and **<Product type>** list column bodies, and then from the On demand toolbar, click **Group / Ungroup**.
   You can use Ctrl+click or Shift+click methods to select multiple columns.
4. Click the **<Employee name>** list column body, from the On demand toolbar click **Headers & footers**, and then click **Create header**.
5. With the **<Employee name>** column body still selected in the list, from the On demand toolbar, under **More**, click **Delete** to remove the redundant column.

6. Click the **<Revenue>** list column body, from the On demand toolbar, click **Summarize**, and then click **Total**.

   A section of the result appears as follows:

Product line	Product type	Product	Quantity	Revenue
<Employee name>				
<Product line>	<Product type>	<Product>	<Quantity>	<Revenue>
	<Product type> - Total			<Total(Revenue)>
	<Product type>	<Product>	<Quantity>	<Revenue>
	<Product type> - Total			<Total(Revenue)>
<Product line> - Total				<Total(Revenue)>
<Product line>	<Product type>	<Product>	<Quantity>	<Revenue>
	<Product type> - Total			<Total(Revenue)>
	<Product type>	<Product>	<Quantity>	<Revenue>
	<Product type> - Total			<Total(Revenue)>
<Product line> - Total				<Total(Revenue)>
<Employee name> - Total				<Total(Revenue)>

   To include only the data for the three sales reps that need to know what they have sold, you will filter the report.

7. From the On demand toolbar, click **Filters**, **Edit Filters**, and then click **Add**.
8. Click **Advanced**, and then click **OK**.
9. Create and validate the following expression:

   [Sales (query)].[Employee by region].[Last name] in ('Scott', 'Sinden', 'Torta')

   Hint: Source tab: Employee by region: Last name.

10. Click **OK** to close each open dialog box, and then run the report in **HTML**.
    A section of the result appears as follows:

Product line	Product type	Product	Quantity	Revenue
Alessandra Torta				
Camping Equipment	Cooking Gear	TrailChef Canteen	4,539	55,042.06
		TrailChef Cook Set	8,516	432,851.68
		TrailChef Cup	14,887	48,060.15
		TrailChef Deluxe Cook Set	5,260	634,770.66
		TrailChef Double Flame	2,257	306,593.1
		TrailChef Kettle	8,060	101,092.81
		TrailChef Kitchen Kit	6,234	142,467.02
		TrailChef Single Flame	8,164	512,200.86
		TrailChef Utensils	7,506	140,241.37
		TrailChef Water Bag	24,990	144,791.08
	Cooking Gear - Total			**2,518,110.79**

11. Click **Page down** to scroll through the report, and view the other reps.
    The report is filtered to only include sales made by Alessandra Torta, Bart Scott, and John Sinden.
12. Close the rendered report tab, and save the report in **Team content\B6059** as **Product Sales Report by Sales Rep**.
    You will now create a burst key data item that will let you distribute this report to the sales reps using their email addresses.

Task 2. Create a burst key.

You will create a dynamic email data item that combines the sales representative's first and last name and the domain name. The email data item will be used as a burst key for burst recipients.

In this example participants could use the Email query item in the model. However, a company can change its domain name, before having the email addresses updated yet. To get these reports delivered right away, you can create your own burst key - to send these reports to the new accounts. The steps in this task show how to create the field if it is not a part of the data source.

1. On the **Navigate** tab, click the **Query explorer** tab, and then click **Query1**.
2. On the **Application** bar, open the **Properties** pane.
3. In the **Properties** pane, under the **MISCELLANEOUS** section, rename the query to **DynamicEmail**.

4. From the **Toolbox** tab, drag a **Data Item** to the **Data Items** pane.
5. In the **Data item expression** dialog box, change the **Name** field to **Burst_Key_Email**.
6. In the **Expression Definition** pane, create and validate the following expression:

   lower(substring([Sales (query)].[Employee by region].[First name],1,1) + [Sales (query)].[Employee by region].[Last name]) + '@grtd123.com'

   Hint: Source tab: Employee by region: First name, Last name.

   The lower function returns the string with all text in lowercase. The substring function returns the first initial of the first word in the [First name]. If a person's name is Jean Marc, the substring function will return J.
7. Click **OK** to close the dialog box.
8. On the **Navigate** tab, click the **Page explorer** tab, click **Page1**, and then select the entire list.
9. From the **Properties** pane, under the **DATA** section, double-click **Properties** then select the **Burst_Key_Email** check box, and then click **OK**.

   You are making this data item a part of the list.

Task 3. Set the Burst options.

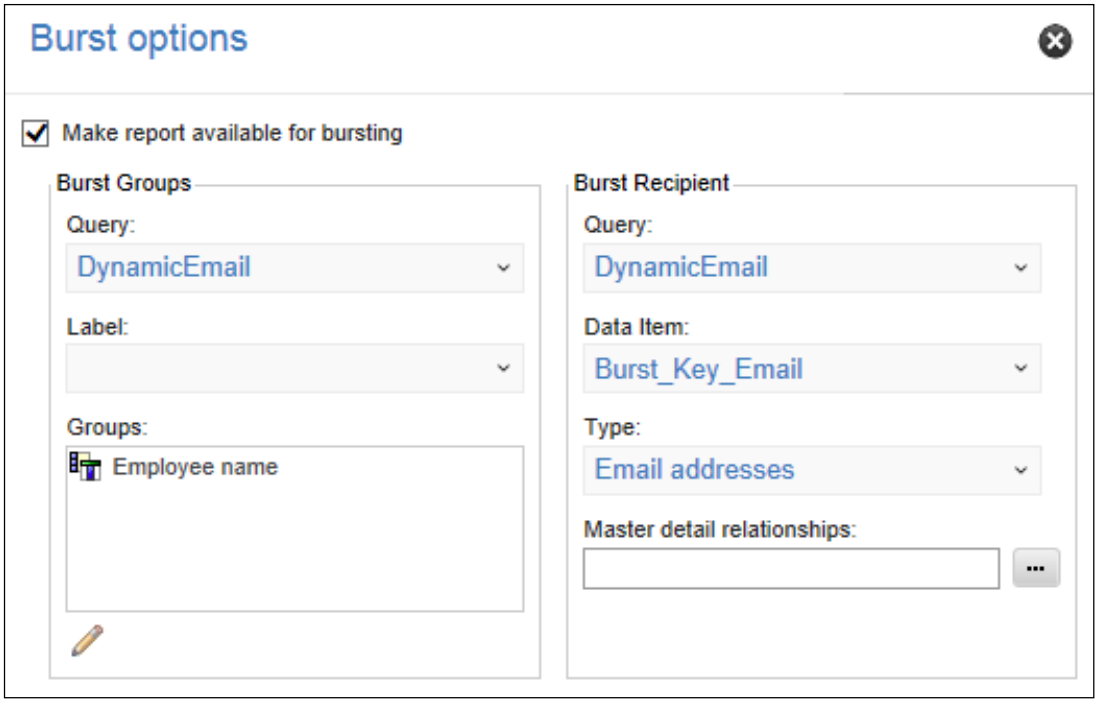
1. From the **Properties** pane, on the **Select Ancestor** list, click **Report**.
2. On the **Properties** pane, under **RUNNING & VALIDATING**, double-click **Burst options**.
3. Click the **Make report available for bursting** check box to select it.

   This check box sets a flag for IBM Cognos Analytics portal to allow bursting.
4. In the **Burst Groups** section, from the **Query** list, select **DynamicEmail**.

   You will not enter a value for the Label, because you will use the default label, Employee name.
5. Below **Groups**, click **Edit**, double-click **Employee name** to add it to the **Groups** folder, and then click **OK** to close the **Grouping & sorting** dialog box.

6. In the **Burst Recipient** section, from the **Query** list, select **DynamicEmail**, from the **Data Item** list, select **Burst_Key_Email**, and then from the **Type** list, select **Email addresses**.

   A section of the result appears as follows:



   With these options, the report will be distributed by email (Type) to the email address specified in Burst Key. The report is grouped by Employee name, and each employee only receives the report with their data.

   By selecting Email addresses (in the Burst options dialog box), you are indicating that the email address is available in the Burst_Key_Email data item.

7. Click **OK**.
8. On the **Application** bar, click **Save**.
9. On the **Application** bar, from the currently active report list, select **Welcome** to return to the **IBM Cognos Analytics** portal.

Task 4. Burst the report and view the results.

The mail server must be started before continuing with this task.

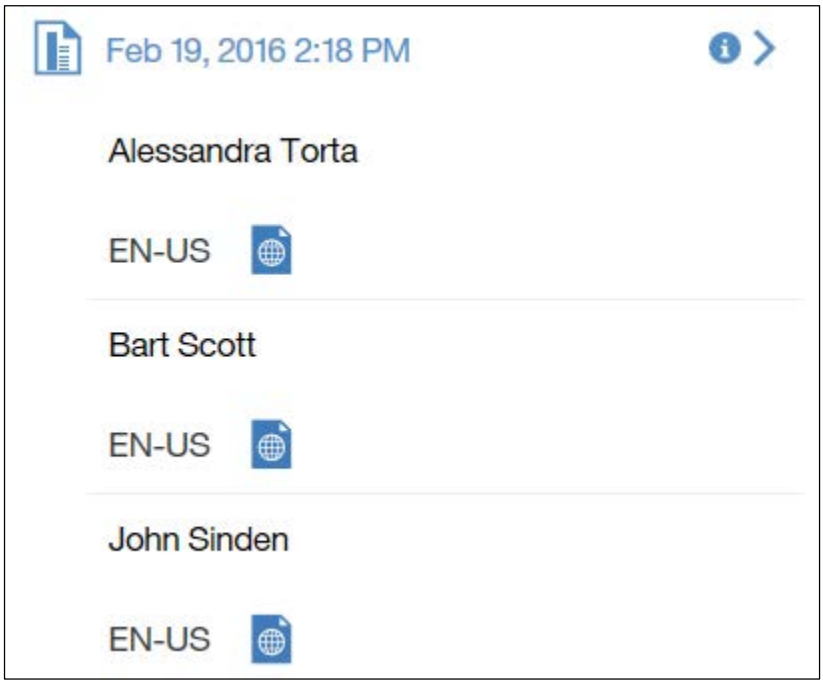
1. From **Internet Explorer**, open a new **Internet Explorer** tab.
2. In the address window type the following address **http://localhost/mail/aperson.nsf** and then press **Enter**.

   You are now acting as an administrator, Admin Person, to access the mail server and the administrator email account.

3. If prompted, log in as **Admin Person/Education1**.

4. In the left pane, click **Mail**  .
5. In **Internet Explorer**, return to the **IBM Cognos Analytics** portal tab, and navigate to **Team content\B6059**.
6. For the **Product Sales Report by Sales Rep** report, click **More**, and then click **Run as**.
7. Click the **Run in background** button to enable it (should show a checkmark on the toggle switch), and then expand **Advanced**.
8. Click the **Burst the report** check box to select it.
9. Expand the **Delivery** section, click the **Send this report by email** check box to select it, and then click **Done**.
10. Click **Run**.
11. After a couple of minutes (to allow the report versions to render), click the **Team content\B6059** page, click the **More** button of the **Product Sales Report by Sales Rep** report, and then click **View versions**.
12. In the **Versions** flyout, click the most recent timestamp.

    The output versions appear similar to the following:



13. On the **Side panel**, click **Team content** to close the open panels.
14. In **Internet Explorer**, click the **Inbox** tab, type the following address **http://localhost/mail/atorta.nsf**, and then press **Enter**.
15. Click **Mail**.

    If prompted for credentials, use Alessandra Torta/Education1.

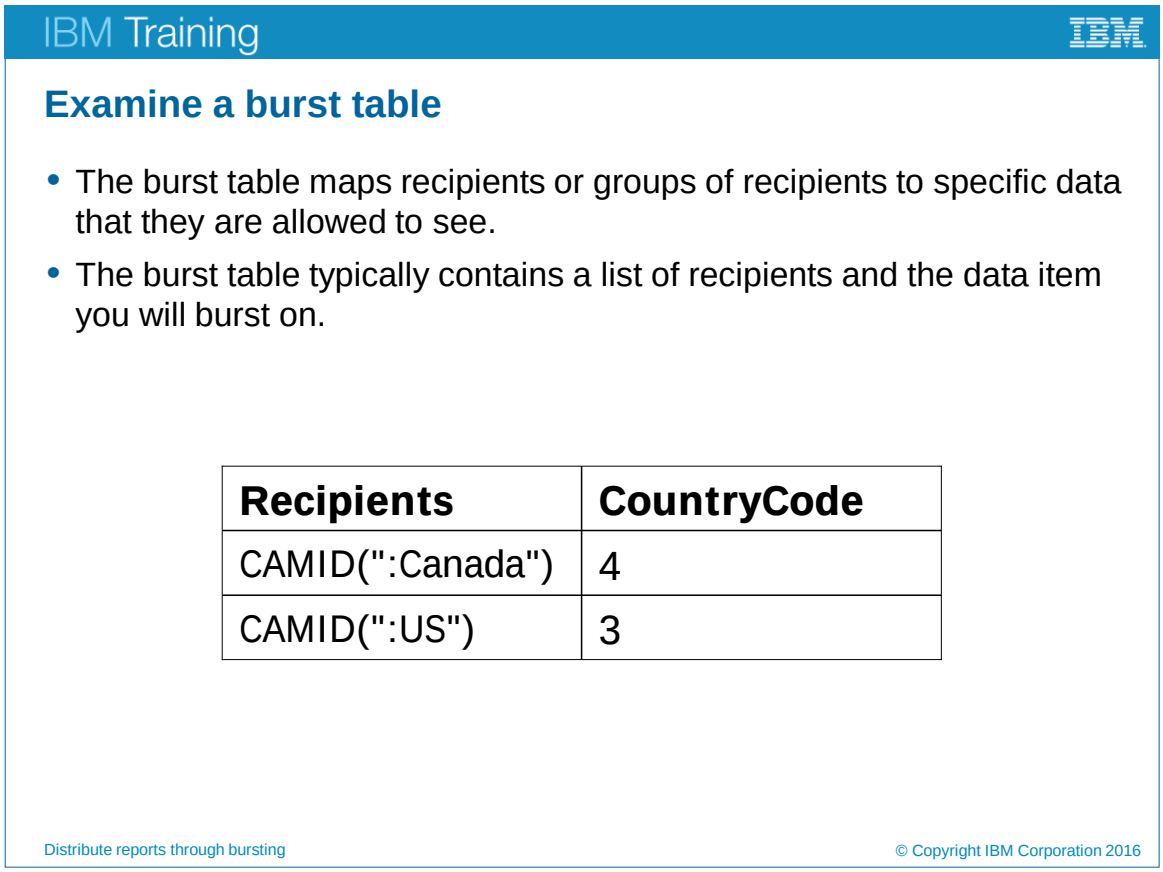
    The burst report appears in the Inbox.
16. Open the email from **FBretton** (Frank Bretton, the report author), and then click the report link.

    The report renders, and a section of the data to Alessandra Torta appears as follows:

Product line	Product type	Product	Quantity	Revenue
Alessandra Torta				
Camping Equipment	Cooking Gear	TrailChef Canteen	4,539	55,042.06
		TrailChef Cook Set	8,516	432,851.68
		TrailChef Cup	14,887	48,060.15
		TrailChef Deluxe Cook Set	5,260	634,770.66
		TrailChef Double Flame	2,257	306,593.1
		TrailChef Kettle	8,060	101,092.81
		TrailChef Kitchen Kit	6,234	142,467.02
		TrailChef Single Flame	8,164	512,200.86
		TrailChef Utensils	7,506	140,241.37
		TrailChef Water Bag	24,990	144,791.08
	Cooking Gear - Total			**2,518,110.79**

17. Scroll to the end of the report to see total revenue generated.
18. Close the rendered report tab.
19. Repeat steps **14** through **18** for:
    - Bart Scott, http://localhost/mail/bscott.nsf
    - John Sinden, http://localhost/mail/jsinden.nsf
20. Return to **IBM Cognos Analytics** portal, and leave it open for the next demonstration.

Results:
By creating a burst key and setting the appropriate Burst options in IBM Cognos Analytics - Reporting, you were able to burst product sales report to sales reps. Each sales rep received only the data specific to them.



Examine a burst table

CAMID (Cognos Access Manager ID) is an internal path to users, groups, roles, contacts or distribution lists.

The list of recipients can be either a CAMID, or it could be the alternate search path syntax.

The burst table must also include the data item you intend to burst on. In the slide example, these values are in the CountryCode column. This column will be used to create the appropriate relationship in Framework Manager.

Burst tables are typically created by the Database Administrator (DBA), and the recipient values are provided to the DBA by the report author.

After the DBA creates the burst table in the source database, the metadata modeler must import the table's metadata into Framework Manager. The modeler must create a relationship between the burst table query subject and the query subject containing the item that you, the report author, are bursting on.

IBM Training

Examine a mixed recipient list

- You can mix the recipient types in your burst table for greater flexibility in your distribution methods.

Recipients	CountryCode
CAMID(":Canada")	4
CAMID(":"/contact[@name='John Sinden'])	3
CAMID(":"/distributionList[@name="European Partners)	1
CAMID("Local NT ID:u=S-1-5-21-1434109735-2681017343-4103935507-1037")	1
fbretton@grtd123.com	4

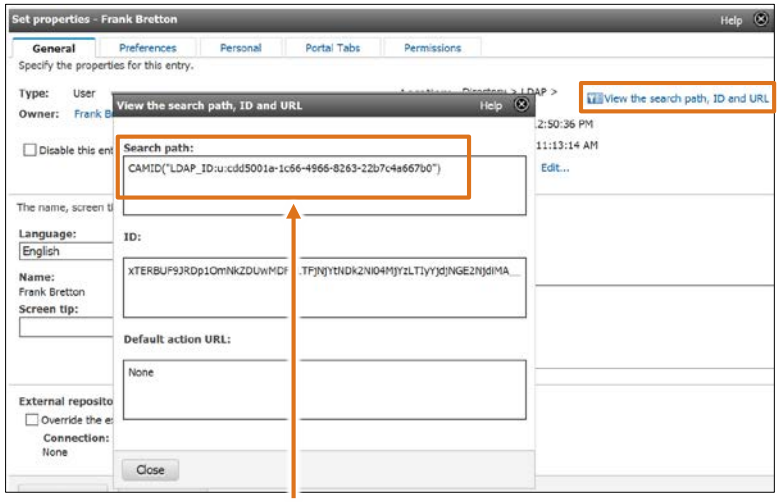
Distribute reports through bursting

Examine a mixed recipient list

In a mixed recipient list scenario, you should not mix email address recipients and alternate search path recipients. If you are using the alternate search path syntax, you must set this option to Directory entries. This is because the syntax contains the @ symbol which will be mistaken for an e-mail address.

IBM Training

Obtain burst recipient values for LDAP namespaces



CAMID("LDAP_ID:u:cdd5001a-1c66-4966-8263-22b7c4a667b0")

Alternate Search Path Syntax:
/directory/namespace[@name="LDAP_ID"]//account[@userName="brettonf"]

Obtain burst recipient values for LDAP namespaces

You can obtain values for burst recipients by examining their properties in IBM Cognos Administration.

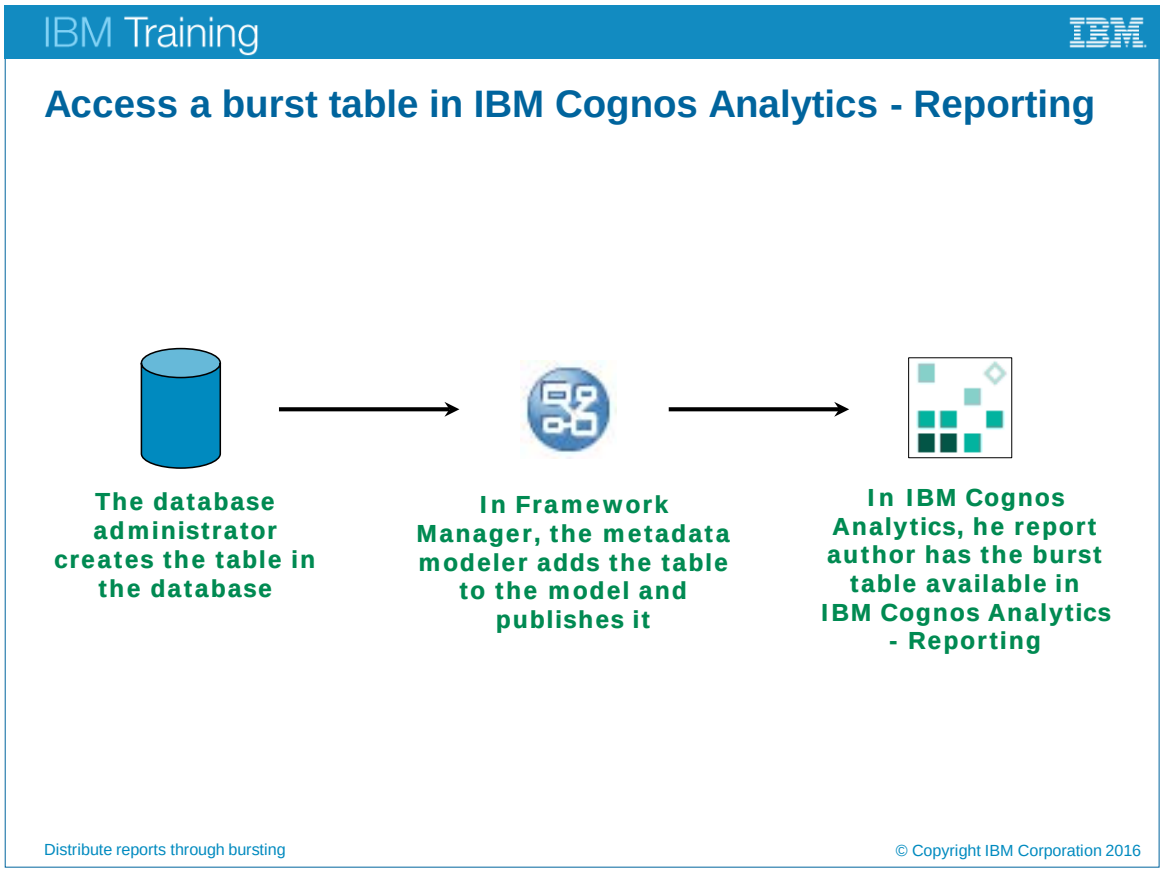
You can copy the CAMID, or use an alternate search path syntax that supports user-friendly IDs, into the burst recipient expression.

You can burst to both directory and the user's email address if your authentication provider contains the users' email address, or if you enter it manually in IBM Cognos Analytics portal for each user.

The CAMID is accessed from the Properties page of any object. The CAMID is similar to the Member Unique Name (MUN), in that it uniquely identifies objects. The double forward slash preceding the account element indicates that you are searching all accounts under the specified namespace.

If you are using an NTLM or a Series 7 namespace, the user IDs are alphanumeric or numeric. You can create an alternative search path that dynamically generates user IDs based on users' names and not alphanumeric or numeric IDs. Using this CAMID syntax, you can substitute a static user ID (such as brettonf in the slide example) with a dynamically generated user ID when creating your burst key.

The search path for NTLM namespace is of the form: CAMID("Local NT ID: u=S-1-5-21-1004336348-688789844-682003330-1134").



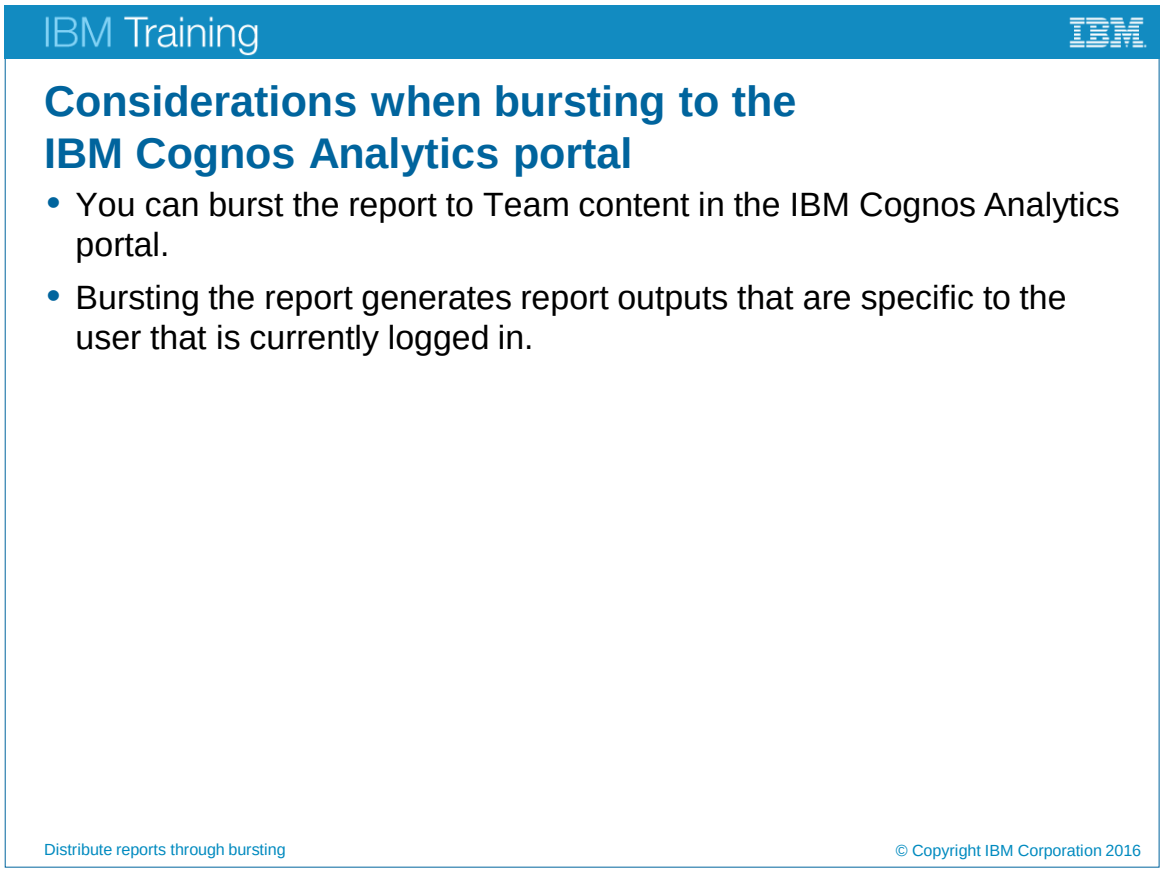
Access a burst table in IBM Cognos Analytics - Reporting

The metadata modeler can set access permissions so that only specific report authors can see the burst table.

Burst tables can be built independently of IBM Cognos Analytics - Reporting or IBM Cognos Analytics using user account information from external authentication providers. This information is extracted from the namespace, arranged to fit the structure of CAMIDs, and then added to a table in a data source.

Burst tables will often be created programmatically since typing each individual record into a database table would be very time consuming.

Most popular programming languages can incorporate libraries that allow them to interact with security software in this way. For example, LDAP security information can be accessed using the Java Naming and Directory Interface (JNDI) libraries.



Considerations when bursting to the IBM Cognos Analytics portal

To ensure that a report is burst to the Team content in IBM Cognos Analytics portal, you must specify that you want to burst to Directory entries in the Burst options dialog box in IBM Cognos Analytics - Reporting.

If the Send the report by email option is also selected on the Run with advanced options page, and the email addresses for the recipients are accessible by IBM Cognos Analytics, recipients will also receive the report by email.

After a report is burst, each user will see the data subset specific to them.

To view output versions for a report, you can click the View output versions icon in Team content. Administrators have permission to view all output versions of the report in the Schedule Management tool of the IBM Cognos Analytics portal.

IBM Training

Demonstration 2

Burst a crosstab report to the IBM Cognos Analytics portal using a burst table

Demonstration 2: Burst a crosstab report to the IBM Cognos Analytics portal using a burst table

Demonstration 2: Burst a crosstab report to the IBM Cognos Analytics portal using a burst table

Purpose:
You want sales reps to see product revenue that was generated by their country. For example, sales reps from Italy should only see report outputs that contain Italian sales data. To accomplish this, you will use the data in a burst table to burst a report to Team content.

Portal: **http://vclassbase:9300/bi**
User/Password: **brettonf/Education1**
Application: **IBM Cognos Analytics - Reporting**
Package: **Team content\Samples\Models\GO data warehouse (query)**
Report Type: **List**
Folder: **Sales and Marketing (query)**
Namespace: **Sales (query)**

Task 1. Create a report that displays product revenue by country.

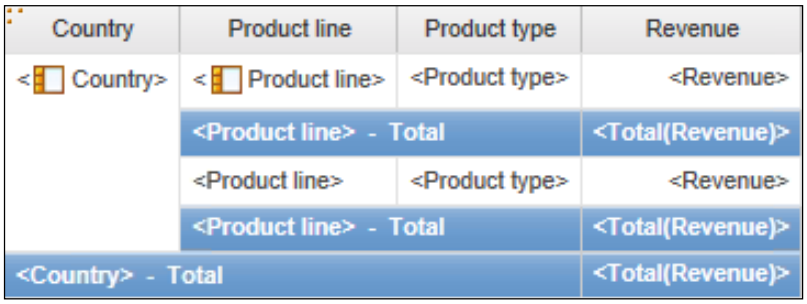
1. Open a new **List** template using the **GO data warehouse (query)** package.
2. From the **Data/Source** tab, add the following query items to the list report object:
   - Employee by region: **Country**
   - Products: **Product line**, **Product type**
   - Sales fact: **Revenue**

Country	Product line	Product type	Revenue
<Country>	<Product line>	<Product type>	<Revenue>

3. Select the **<Country>** and **<Product line>** list column bodies, and then from the On demand toolbar, click **Group / Ungroup**.

4. Click the **<Revenue>** list column body, from the On demand toolbar, click **Summarize**, and then click **Total**.

   A section of the result appears as follows:



Country	Product line	Product type	Revenue
<Country>	<Product line>	<Product type>	<Revenue>
	<Product line> - Total		<Total(Revenue)>
	<Product line>	<Product type>	<Revenue>
	<Product line> - Total		<Total(Revenue)>
<Country> - Total			<Total(Revenue)>

5. On the **Navigate** tab, click the **Query explorer** tab, and then click **Query1**.
6. On the **Application** bar, open the **Properties** pane.
7. In the **Properties** pane, under the **MISCELLANEOUS** section, rename the query to **Product Revenue by Country**.
8. On the **Data/Source** tab, expand the **Burst table by country** query subject, and then drag **Recipients** to the **Data Items** pane.

   The Burst table by country table includes the Country CAMID Recipients and the Country Code for each user.
9. Navigate to **Page1**, and then select the entire list.
10. From the **Properties** pane, under **DATA**, double-click **Properties**, and then select the **Recipients** data item check box to make it a property of the list.
11. Click **OK**.

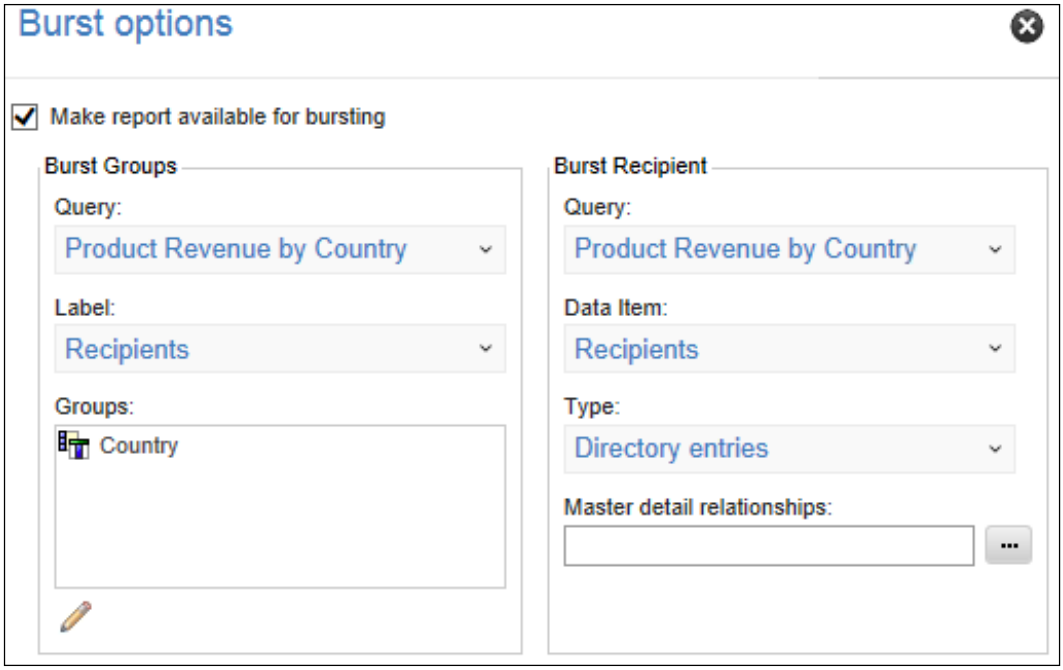
Task 2. Set Burst options.

1. Click the **Select Ancestor** button, and then click **Report**.
2. On the **Properties** pane, under **RUNNING & VALIDATING**, double-click **Burst options**, and then select **Make report available for bursting**.
3. In the **Burst Groups** section, in the **Query** list, select **Product Revenue by Country**, and then in the **Label** list, select **Recipients**.

   The label identifies the text that appears in the report name when burst to the Web.
4. Below **Groups**, click **Edit**, double-click **Country** to add it to the **Groups** folder, and then click **OK** to close the **Grouping & sorting** dialog box.

5. In the **Burst Recipient** section, in the **Query** list, select **Product Revenue by Country**, in the **Data Item** list, select **Recipients**, and then in the **Type** list, select **Directory entries**.

   A section of the result appears as follows:



   A report will be generated for each country, as indicated by the Groups, and will be distributed to each burst recipient - in this case, country. Each report will be labeled by the Recipients, which is the CAMID of a country.

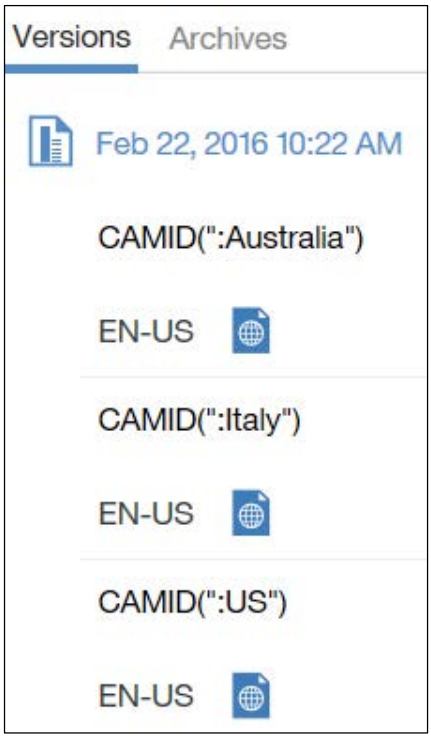
6. Click **OK** to close the **Burst options** dialog box.
7. Save the report in **Team content\B6059** as **Sales by Country**, and then remove the **IBM Cognos Analytics - Reporting** report.

Task 3. Burst the report.

1. From **IBM Cognos Analytics** portal, navigate to **Team content\B6059**.
2. Under **More** for the **Sales by Country** entry, click **Run as**.
3. Enable **Run in background**, and then expand **Advanced.**
4. Click the **Burst the report** check box to select it, and then click **Run**.

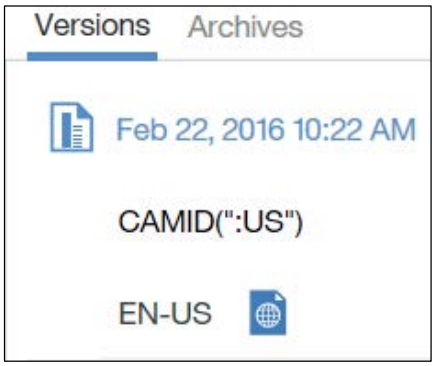
5. Navigate to **Team content\B6059**, and then under **More** for the **Sales by Country** report, click **View versions**.
6. Click the most recent timestamp item to expand.

   A section of the result appears similar to the following:



   Three output versions were created for this report: one each for Australia, Italy, and United States. Because you are logged in as Frank Bretton, a report author, you can see all three output versions.
7. On the **Application** bar, click **Frank Bretton**, and then click **Sign out**.
8. Sign in as **scottb\Education1**, and then click **OK**.
9. Navigate to **Team content\B6059**, and then, under **More** for the **Sales by Country** report, click **View versions**.
10. Click the most recent timestamp to expand.

    A section of the result appears similar to the following:



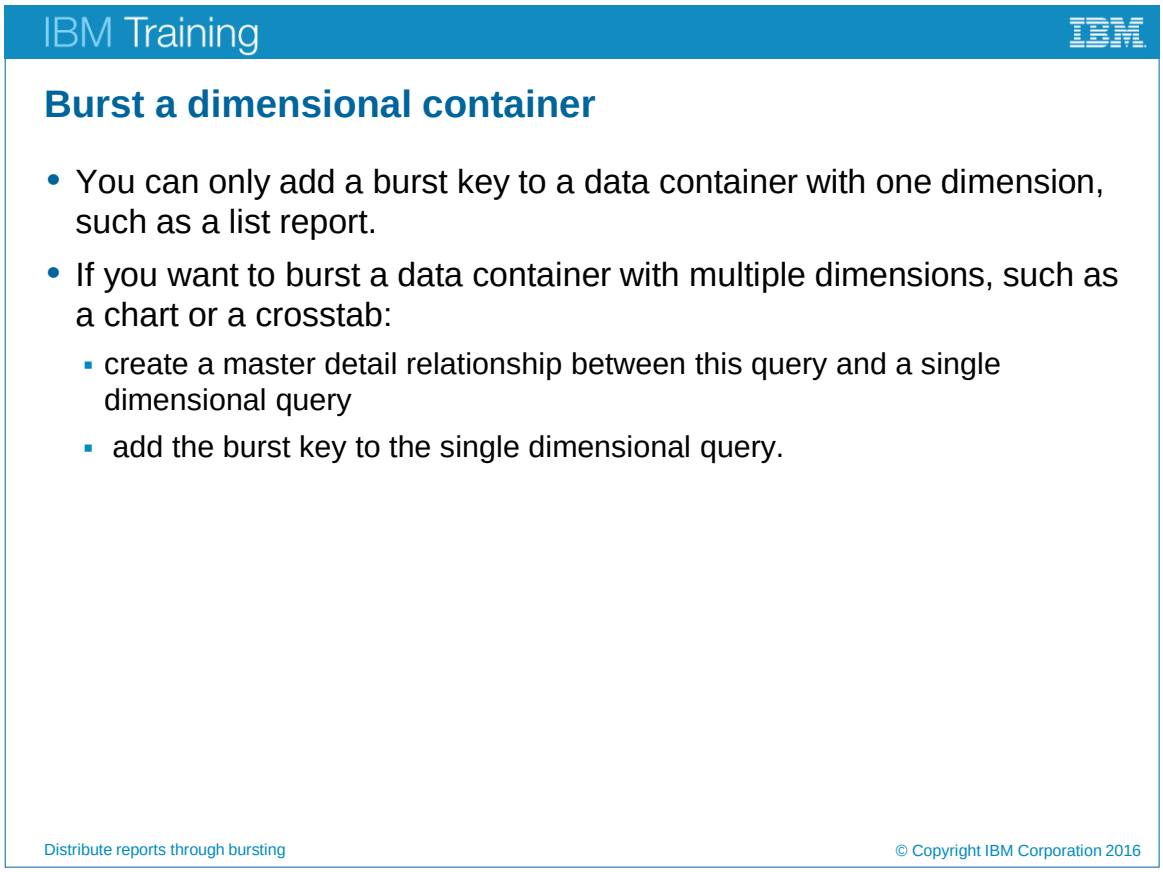
    Only one output version is shown, because Bart Scott belongs to the US group.

11. Click the **HTML**  document.
    A section of the report appears as follows:

Country	Product line	Product type	Revenue
United States	Camping Equipment	Cooking Gear	43,152,191
		Lanterns	21,597,711.83
		Packs	57,520,074.37
		Sleeping Bags	50,338,776.21
		Tents	90,663,037.65
	Camping Equipment - Total		**263,271,791.06**

12. Sign out, and then sign in as **brettonf\Education1**.
13. Leave the **IBM Cognos Analytics** portal open for the next demonstration.

Results:
The report that you created shows how much revenue each country generated by product type. You added a burst key to the report, set the Burst options, and burst the report to the Web.



Burst a dimensional container

An easy way to create a master detail relationship for a dimensional data container is to:

1. Create a list report with the item that you want bursting to group on as your first column. This will be the master list.
2. Insert your chart or crosstab as the second column of the list. This automatically creates a second query, which will be the detail query.
3. Add the item that you will group on to your chart or crosstab.

In the list, create a section with the item that you want to group.

IBM Training

Demonstration 3

Burst a crosstab report to the IBM Cognos Analytics portal using a burst table and a master detail Relationship

Country: Australia

Revenue	2011	2012	2013
Camping Equipment	9,752,591.01	19,175,957.2	13,007,383.98
Golf Equipment	4,094,643.54	8,482,438.67	6,502,474.22
Mountaineering Equipment	2,691,279.15	5,861,253.12	5,380,587.79
Outdoor Protection	600,956.77	367,636.38	171,750.41
Personal Accessories	2,131,381.68	5,081,517.25	4,261,477.85

Demonstration 3: Burst a crosstab report to the IBM Cognos Analytics portal using a burst table and a master detail relationship

Demonstration 3: Burst a crosstab report to the IBM Cognos Analytics portal using a burst table and a master detail relationship

Purpose:
You want sales reps to view a crosstab report of the revenue generated by their country by product line and year. For example, sales reps from Italy will see a report that contains Italian sales data by product line and year. To burst reports to Team content in IBM Cognos Analytics portal, you will use the data in a burst table and create a master detail relationship.

Portal:	**http://vclassbase:9300/bi**
User/Password:	**brettonf/Education1**
Application:	**IBM Cognos Analytics - Reporting**
Package:	**Team content\Samples\Models\GO data warehouse (query)**
Report Type:	**List**
Folder:	**Sales and Marketing (query)**
Namespace:	**Sales (query)**

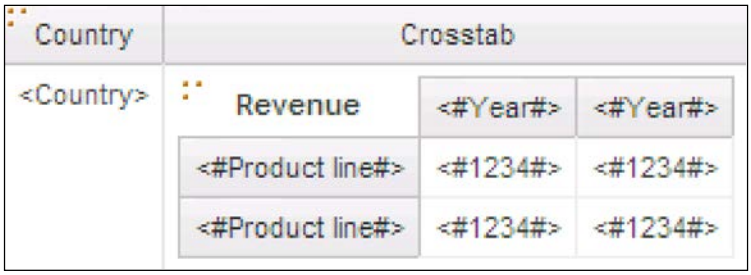
Note: It is not possible to burst a crosstab directly. In order to burst a crosstab, you need to create a master detail relationship between the crosstab and a single dimensional query.

Task 1. Create a crosstab within a list report.

1. Open a new **List** template using the **GO data warehouse (query)** package.
2. From the **Data/Source** tab, add the **Country** query item, from the **Employee by region** query subject, to the list data container.
3. From the **Toolbox** tab, drag a **Crosstab** to the last column of the list data container. (inside the list data container).
4. From the **Object and query name** dialog box, rename **Query2** to **Product line Revenue Query**, and then click **OK**.

5. From the **Data/Source** tab, populate the crosstab with the following query items:
   - **Rows**:
     - Products: **Product line**
   - **Columns**:
     - Time: **Year**
   - **Measures**:
     - Sales fact: **Revenue**

   The result appears as follows:



6. Click the **Country** list column body, and then from the On demand toolbar, click **Section / unsection**.
7. On the **Application** bar, open the **Properties** pane.
8. Click the **Crosstab** list column title, and then in the **Properties** pane, under the **BOX** section, change the **Box type** property to **None**.
9. In the **Crosstab**, sort **<#Product line#>** and **<#Year#>** in **Ascending** order.

10. Run the report in **HTML**.

    A section of the result appears as follows:

Country: Australia

Revenue	2010	2011	2012	2013
Camping Equipment	332,986,338.06	402,757,573.17	500,382,422.83	352,910,329.97
Golf Equipment	153,553,850.98	168,006,427.07	230,110,270.55	174,740,819.29
Mountaineering Equipment		107,099,659.94	161,039,823.26	141,520,649.7
Outdoor Protection	36,165,521.07	25,008,574.08	10,349,175.84	4,471,025.26
Personal Accessories	391,647,093.61	456,323,355.9	594,009,408.42	443,693,449.85

Country: Austria

Revenue	2010	2011	2012	2013
Camping Equipment	332,986,338.06	402,757,573.17	500,382,422.83	352,910,329.97
Golf Equipment	153,553,850.98	168,006,427.07	230,110,270.55	174,740,819.29

    Notice that each of the countries has identical data. This is because there is no relationship between the list and the crosstab, so you see data for all countries. You will add query items to both queries and then create a master detail relationship to link the queries, which will make the results more meaningful.

11. Close the rendered report tab.

Task 2. Add a query item to the queries.

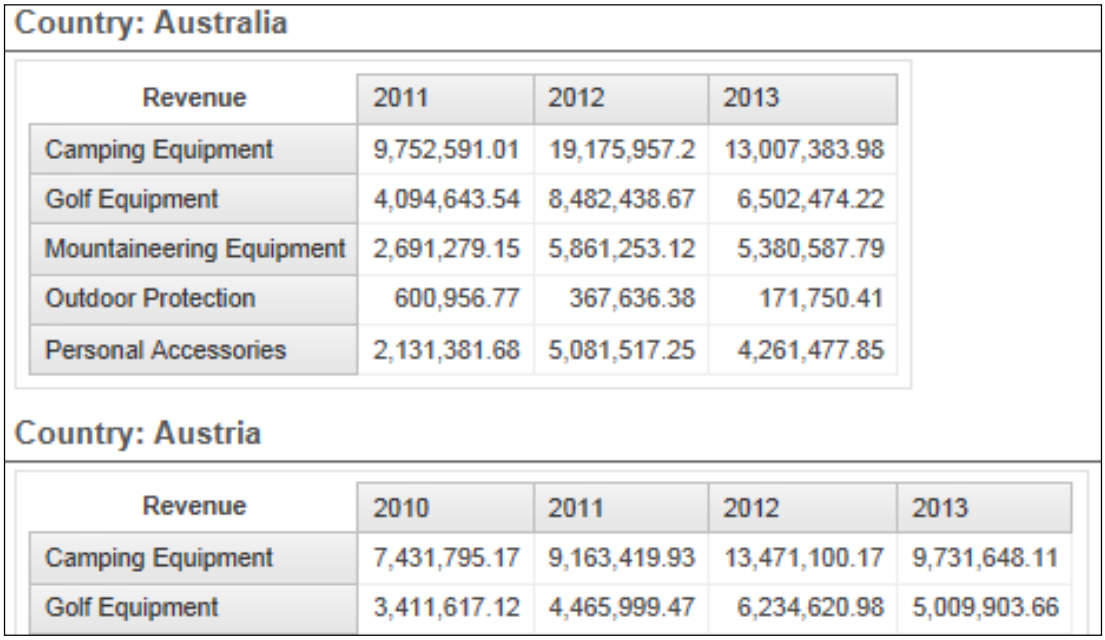
1. On the **Navigate** tab, click **Query explorer**, and then click **Query1**.
2. From the **Properties** pane, under **MISCELLANEOUS**, rename **Query1** to **Country Query**.
3. From the **Data/Source** tab, add the following items to the **Data Items** pane:
   - Employee by region: **Codes**: **Country code**
   - Burst table by country: **Recipients**
4. On the **Navigate** tab, click **Product Line Revenue Query**.
5. From the **Data/Source** tab, add **Country code** from **Employee by region\Codes.**
6. On the **Navigate** tab, click the **Page explorer** tab, and then click **Page1**.

Task 3. Create a master detail relationship and set burst options.

1. Click any cell in the crosstab, and then from the On demand toolbar, under **More**, click **Master Detail Relationships**.
2. In the **Master detail relationships** dialog box, click **New Link**.

   IBM Cognos Analytics - Reporting creates a link between the first items; the Country data item in Country Query and the Product line data item in Product Line Revenue Query. You need to link Country code in Country Query to Country Code in Product Line Revenue Query. You will create the link on Country code instead of Country because Country code is indexed in the database and the queries will run faster.
3. In the **Master: Country Query** pane, click **Country code.**
4. In the **Detail: Product Line Revenue Query** pane, click **Country code.**
5. Click **OK** to close the **Master detail relationships** dialog box.

6. Run the report in **HTML**.
   Note that each crosstab is different now, due to the master detail relationship.



Country: Australia

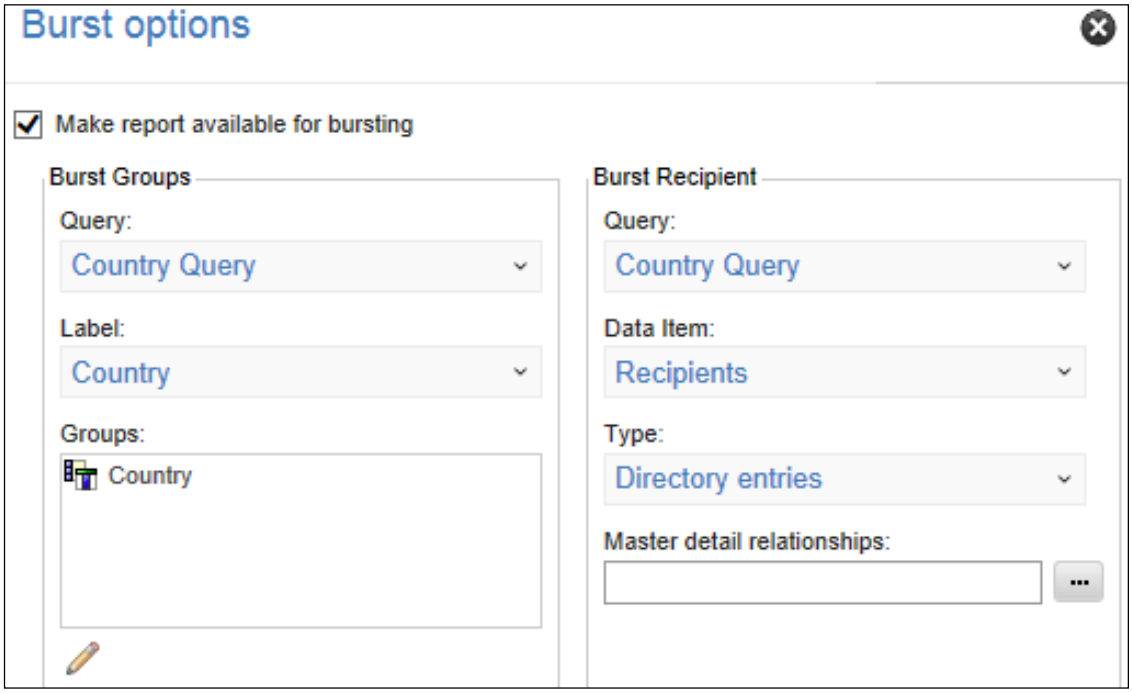
Revenue	2011	2012	2013
Camping Equipment	9,752,591.01	19,175,957.2	13,007,383.98
Golf Equipment	4,094,643.54	8,482,438.67	6,502,474.22
Mountaineering Equipment	2,691,279.15	5,861,253.12	5,380,587.79
Outdoor Protection	600,956.77	367,636.38	171,750.41
Personal Accessories	2,131,381.68	5,081,517.25	4,261,477.85

Country: Austria

Revenue	2010	2011	2012	2013
Camping Equipment	7,431,795.17	9,163,419.93	13,471,100.17	9,731,648.11
Golf Equipment	3,411,617.12	4,465,999.47	6,234,620.98	5,009,903.66

7. Close the rendered report tab.
8. In the **Properties** pane, click the **Select Ancestor** button, and then click **Report**.
9. Under **RUNNING & VALIDATING**, double-click **Burst options**.
10. Click the **Make report available for bursting** check box to select it.
11. In the **Burst Groups** section, in the **Query** list, select **Country Query**, and then in the **Label** list, select **Country**.
    Country Query references the list with only one dimension.
12. Below **Groups**, click **Edit**, double-click **Country** to add it to the **Groups** folder, and then click **OK** to close the **Grouping & sorting** dialog box.
13. In the **Burst Recipient** section, in the **Query** list, select **Country Query**.
14. In the **Data Item** list, select **Recipients**.

15. In the **Type** list, select **Directory entries**.
    A section of the result appears as follows:

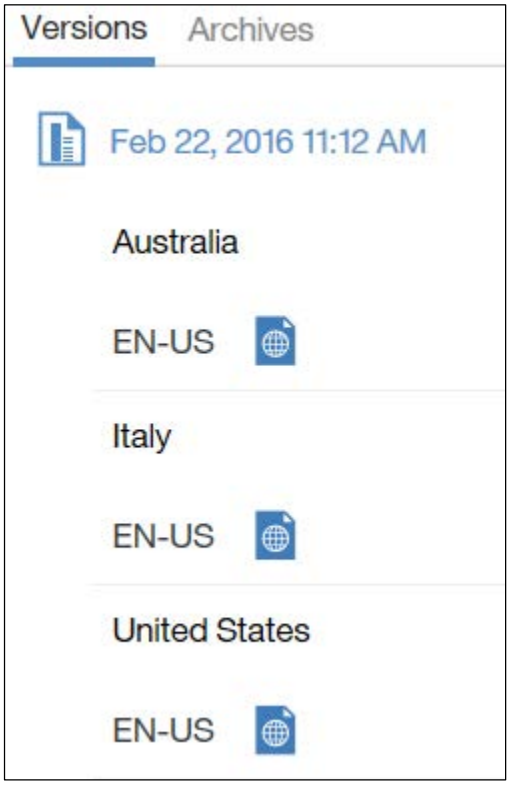


16. Click **OK**, and then save the report as **Country Product line Revenue Bursted Report** in **Team content\B6059**.
    You are ready to burst the report.

Task 4. Burst the report and view the results.

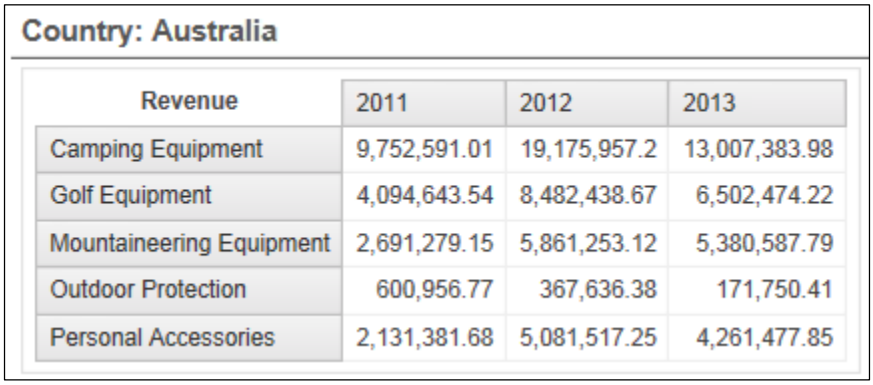
1. Remove the report, and then in the **IBM Cognos Analytics** portal, navigate to **Team content\B6059**.
2. Under **More**, for the **Country Product Line Revenue Bursted Report** entry, click **Run as**.
3. Enable **Run in background**, and then expand **Advanced**.
4. Click the **Burst the report** check box to select it, and then click **Run**.
5. Navigate **to Team content/B6059**, and then under **More** for the **Country Product Line Revenue Bursted Report** entry, click **View versions**.

6. Click the most recent timestamp to open it.
   A section of the result appears similar to the following:



   Three report output versions were created: one each for Australia, Italy, and United States. Because you are logged in as Frank Bretton, the report author, you can see all three output versions. You will now log on as a different user.
7. On the **Application** bar, sign out, and then sign in as **sindenj/Education1**.
8. On the **IBM Cognos Analytics** portal navigate to **Team content\B6059**, and then under **More** for the **Country Product Line Revenue Bursted Report** entry, click **View versions**.
9. Click the most current timestamp to open it.
   Only one output version, for Australia, is displayed.

10. Open the **HTML** report.
    The results appear as follows:



Country: Australia

Revenue	2011	2012	2013
Camping Equipment	9,752,591.01	19,175,957.2	13,007,383.98
Golf Equipment	4,094,643.54	8,482,438.67	6,502,474.22
Mountaineering Equipment	2,691,279.15	5,861,253.12	5,380,587.79
Outdoor Protection	600,956.77	367,636.38	171,750.41
Personal Accessories	2,131,381.68	5,081,517.25	4,261,477.85

    This output version contains only Australian data, as you specified in the Burst options.
11. Sign out of **IBM Cognos Analytics**, and then sign in as **brettonf/Education1**
12. Leave the **IBM Cognos Analytics** portal open for the exercise.

Results:
Sales reps can view a crosstab report of the revenue generated in their country by product line and year. To accomplish this, you used the data in a burst table and a master-detail relationship to burst reports to Team content.

IBM Training

Unit summary

- Distribute reports using bursting
- Create burst keys
- Identify report recipients and data items using burst tables
- Distribute reports using email and IBM Cognos Analytics Portal

Unit summary

IBM Training

Exercise 1

Create and burst a sales report to email and IBM Cognos Analytics portal

A new version of Product line by Country Bursted Report is available
FBretton@grtd123.com
To: BScott@grtd123.com

United States HTML

A new version of Product line by Country Bursted Report is available
FBretton@grtd123.com
To: ATorta@grtd123.com

Italy HTML

Exercise 1: Create and burst a sales report to email and IBM Cognos Analytics portal

Exercise 1: Create and burst a sales report to email and IBM Cognos Analytics portal

Some country-based sales reps at the Sample Outdoors Company want to be aware of how the product lines are selling.
Your assignment as Frank Bretton, report author, is to create a report that outlines sales, and then burst it to various country-based sales reps through the web, in Team content, and through email.
Once you burst the report, you will see product line reports for the following country-based sales reps:

- John Sinden, Australia
- Bart Scott, United States
- Alessandra Torta, Italy

To accomplish this:

- Sign in as **Frank Bretton**, and then using the **GO data warehouse (query)** package, **Sales and Marketing (query)** folder, **Sales (query)** namespace.
- Create a list report with **Country** (from Employee by region), **Product line**, **Product type**, **Product** (from Products), and **Revenue** (from Sales fact).
- Group on **Country**, **Product line**, and **Product type**, and summarize **Revenue**.
- Section on **Country**.
- From the **Burst table by country** query subject, add **Recipients** to the query as a data item.
- From the **Employee by region\Codes** query subject, add **Country code** to the query as a data item.
- Set **Recipients** as a property of the list object. \
- Set **Burst options**, labeling the report by **Country**, and then save the report in the **Team content\B6059** folder as **Product Line by Country Revenue Bursted Report**.

- Burst the report and then view the report outputs in IBM Cognos Analytics portal, logged on as brettonf, sindenj, tortaa, and scottb (all with password Education1).
- Open the resulting emails in Lotus iNotes, as each of the following users:
  - (**http://localhost/mail/bscott.nsf**
  - **http://localhost/mail/jsinden.nsf**
  - **http://localhost/mail/atorta.nsf**)

  Note: Email addresses for the users exist in the IBM Cognos Analytics properties for each user.

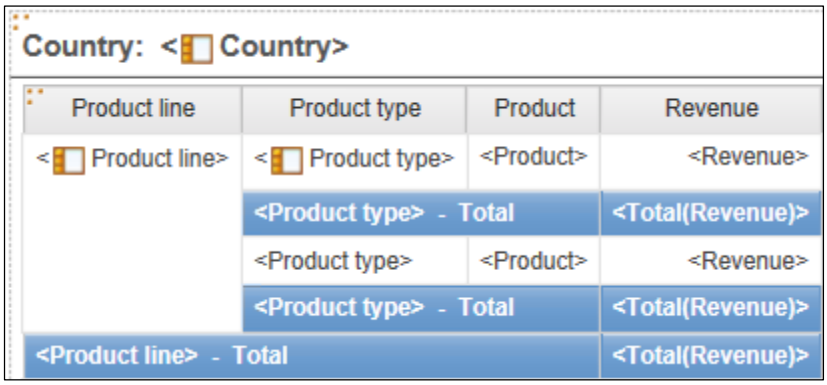
For more information about where to work and the exercise results, refer to the Tasks and results section that follows. If you need more information to complete a task, refer to earlier demonstrations for detailed steps.

Exercise 1: Tasks and results

Task 1. Create a list report.

- **Side panel**: Open a new **List** template using the **GO data warehouse (query)** package.
- **Data/Source** tab: Navigate to **Sales and Marketing (query)/Sales (query)/Employee by region**, and then add **Country** to the list data container:
  - From **Products**, add the following query items to the list data container: **Product line**, **Product type**, **Product**.
  - From **Sales fact**, add **Revenue** to the list report object
- **On demand toolbar**: Group the **<Country>**, **<Product line>**, and **<Product type>** list column bodies.
  - Summarize the **<Revenue>** list column body by **Total**.
  - Section the **< Country>** list column body.

  A section of the results appear as follows:

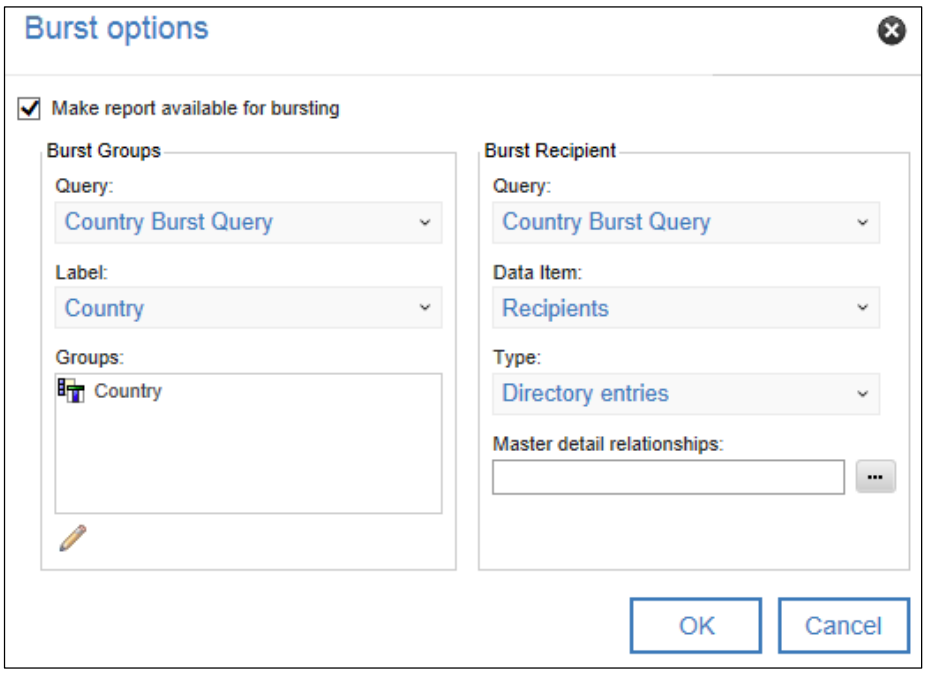


Task 2. Add query items to the query.

- **Navigate** tab: Navigate to **Query1**.
- **Properties pane**: Rename **Query1** to **Country Burst Query**.
- **Source tab**:
  - Navigate to **Employee by region/Codes**, and then add **Country code** to the **Data Items** pane.
  - Navigate to **Burst table by country**, and then add **Recipients** to the **Data Items** pane.
- **Navigate** tab: Navigate to **Page1**.
- **Work area**: Select the entire list data container.

- **Properties** pane: From **Select Ancestors**, click **Report**.
- **Burst options**: Set the following Burst options:
  - Select **Make report available for bursting**.
  - In the **Burst Groups** section, use **Country Burst Query**.
  - Label it **Country**.
  - Group it on **Country**.
  - In the **Burst Recipient** section, use **Country Burst Query**.
  - Use **Recipients** to identify the **Data Item**.
  - Set the **Type** list to **Directory entries**.

  The results appear as follows:

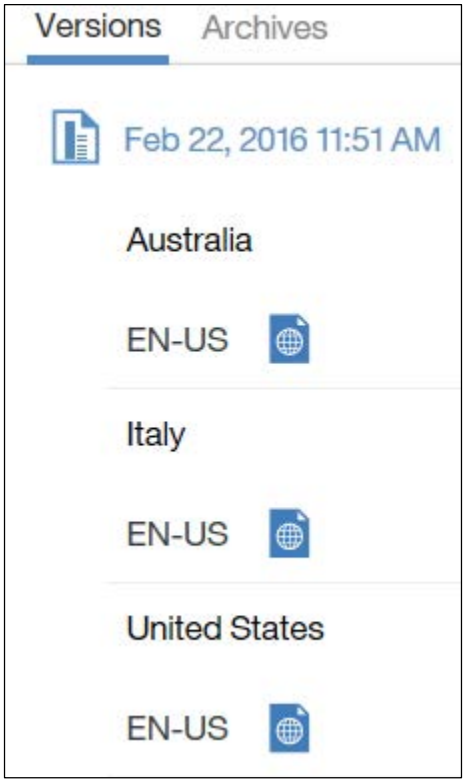


- **Application** bar: Save the report as **Product line by Country Bursted Report** in **Team content\B6059**.
- Remove the report.

Task 3. Burst the report and view the results.

- Navigate to **Team content/B6059**.
- **Product Line by Country Bursted Report**: Click **More**.
- **More**: click **Run as**.
- **Run as**: enable **Run in background**.
  - Expand **Advanced**.
- **Advanced**: Burst the report, and send the report by email.
  - Run the report.
- **IBM Cognos Analytics** portal: View the versions for this report.

  The results appear similar to the following:

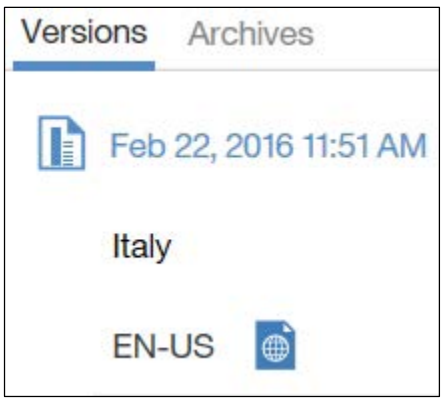


- **Sign out** as Frank Bretton.

Task 4. Log on as different users to review the results.

- **Sign in** window: Log on as **sindenj/Education1**.
- **IBM Cognos Analytics** portal: Navigate to **Team content\B6059**.
- **Product Line by Country Bursted Report**: View the versions for the report.
  The results appear as follows:

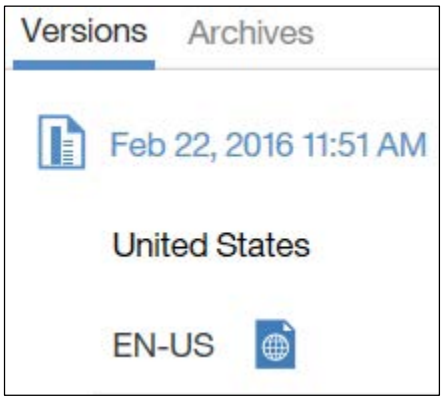
- **Versions** pane: Click **HTML** to run the report.
- Sign out as **John Sinden**.
- **Sign in** window: Log on as **tortaa/Education1**.
- **IBM Cognos Analytics** portal: Navigate to **Team content\B6059**.
- **Product Line by Country Bursted Report**: View the versions for the report.
  The results appear as follows:



- **Versions** pane: Click **HTML** to run the report.
- Sign out as **Alessandra Torta**.

- **Sign on** window: Sign on as **Bart Scott** using the following credentials: **scottb\Education1**.
- **IBM Cognos Analytics** portal: Navigate to **Team content\B6059**.
- **Product Line by Country Bursted Report**: View the versions for the report.

  The results appear as follows:



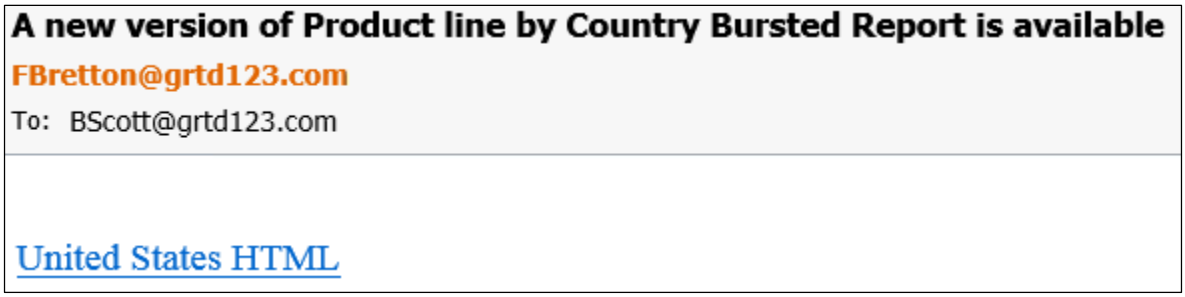
- **Versions** pane: Click the **HTML** to run the report.
- Sign out as **Bart Scott**.

Task 5. Log on to the mail server as different users to review the results.

- **Internet Explorer**: Open a new tab and navigate to **http://localhost/mail/bscott.nsf**

  If prompted to login, use Admin Person\Education1 credentials.
- **IBM Lotus iNotes**: Open **Mail**.
- **Mail-inbox window**: Open **Product Line by Country Revenue Bursted Report** email.

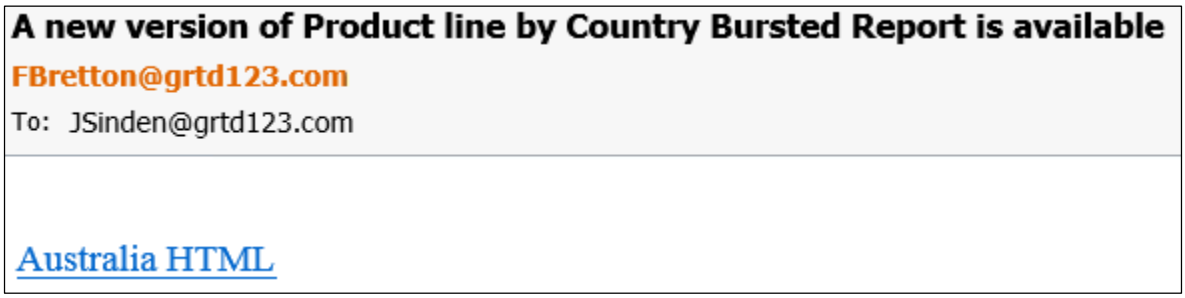
  A section of the results appear as follows:



  You see the link for the HTML report.

- **Internet Explorer**: Change the address to: **http://localhost/mail/jsinden.nsf**
- **IBM Lotus iNotes**: Navigate to the mail-inbox window.
- **Mail-inbox window**: Open **Product Line by Country Revenue Bursted Report** email.

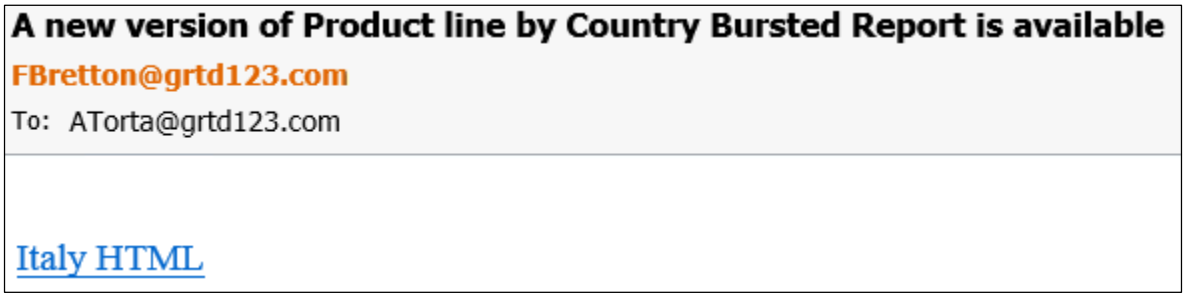
  A section of the results appear as follows:



  You see the link for the HTML report.

- **Internet Explorer**: Change the address to: **http://localhost/mail/atorta.nsf**
- **IBM Lotus iNotes**: Navigate to the mail-inbox window.
- **Mail-inbox window**: Open **Product Line by Country Revenue Bursted Report** email.

  A section of the results appear as follows:



  You see the link for the HTML report.

- Close the **IBM Lotus iNotes** tab.
- Close the web browser

IBM Training

Enhance user interaction with HTML

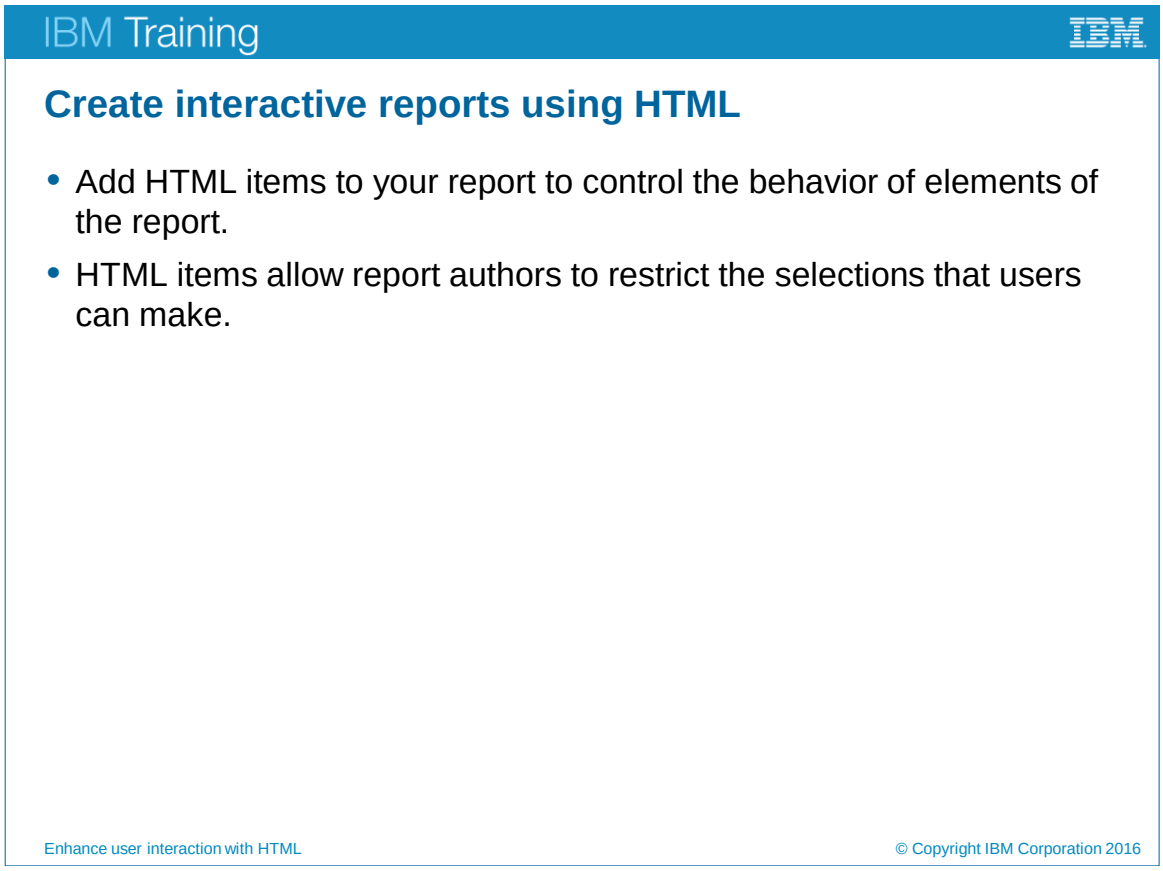
IBM Cognos Analytics (v11.0)

IBM Training

Unit objectives

- Create tooltips that clarify report data
- Send emails using links in a report

Unit objectives



Create interactive reports using HTML

An HTML item adds a container in which you can insert HTML code, such as a link to a multimedia file, or JavaScript.

It is important to consider your audience and how the report will be rendered, as HTML items will only work when the report is rendered in HTML.

If you want to use HTML items with other report outputs (such as PDF), you can use a Rich text Item.

Functionality for the HTML item can be controlled using Capabilities\Report Studio\HTML Items in Report found on the Administration console.

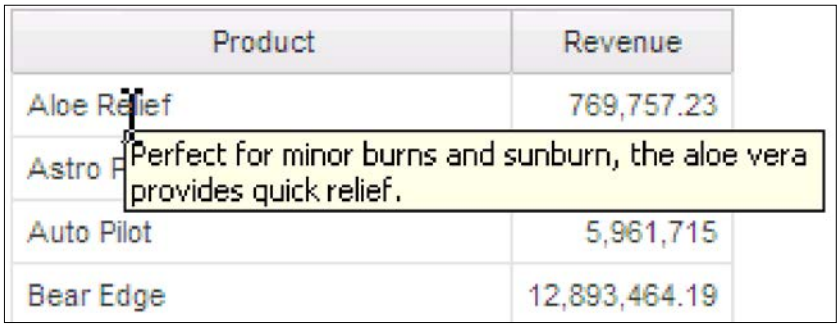
An HTML layout object can be anything your browser will execute, such as links, images, multimedia or JavaScript.

The Appendix A - Explore the Prompt API optional unit demonstrates how you can expand report functionality using HTML items and JavaScript.

IBM Training

Include additional information with tooltips

- You can embed HTML code in your report that will create tooltips in selected areas of your layout.
- Tooltips can link to query data and display information not included in the layout.



Enhance user interaction with HTML

Include additional information with tooltips

You can create a tooltip when you want users to have access to detail that you do not want to display directly in the report layout.

You can create a generic tooltip creation function that accepts a string of text as a parameter and uses it to create the tooltip. This lets you create HTML code out of report expressions and embed tooltips in the report data.

A tooltip is a small window of text that appears beside the mouse pointer when it is held over a specific area of the screen.

It is also possible to create tooltips using the tags in HTML, though it is not possible to apply any formatting to the tooltip window.

<Span> tags are also supported with Rich text items.

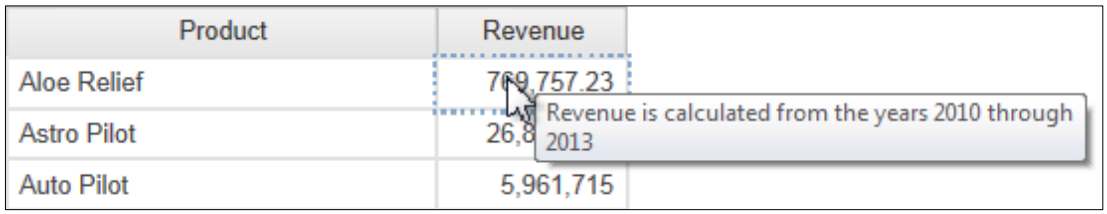
Tooltips are already available on charts by setting the Tooltips property to Yes, however, the technique illustrated here, would be suitable for list and crosstab reports.

Depending on the content of your HTML item, set the appropriate source type. For example, for a static HTML item, use Text.

IBM Training

Demonstration 1

Add tooltips to a report



Enhance user interaction with HTML

Demonstration 1: Add tooltips to a report

Demonstration 1: Add tooltips to a report

Purpose:
Management would like a product revenue report to include additional information such as product descriptions and the years included in the revenue value displayed. This information is only supplementary, however, and should not be added directly to the report layout. You will achieve this by including this information in tooltips, which will display at the appropriate points in the report.

Portal: **http://localhost:9300/bi**
User/Password: **brettonf/Education1**
Application: **IBM Cognos Analytics - Reporting**
Package: **Team content\Samples\Models\GO data warehouse (query)**
Folder: **Sales and Marketing (query)**
Namespace: **Sales (query)**

Task 1. Create the report.

1. Open a new **List** template using the **GO data warehouse (query)** package.
2. From the **Data/Source** tab, add the following query items to the list data container:
   - Products: **Product**
   - Sales fact: **Revenue**
   - Time: **Year**
3. Group the **<Product>** column, and then cut the **<Year>** column from the list.
   You do not want the year displayed in the rows for each product, but you want to keep Year in the query.

4. Run the report in **HTML**.
   Note that as you hover your mouse cursor over the **Product** and the **Revenue** headers and cells, there is no additional information provided to the user.
5. Close the rendered report tab.

Task 2. Create a static text tooltip.

You will add a tooltip to the revenue header, to inform users that the revenue displayed is for all years, otherwise they may assume that it is the current year revenue.

1. On the **Application** bar, under **More**, click **Locked** to unlock the cells of the report.
2. From the **Toolbox** tab, under **ADVANCED**, drag an **HTML item** to the left of the **Revenue** column header title, and then drag another **HTML tem** to the right of the **Revenue** column header title.

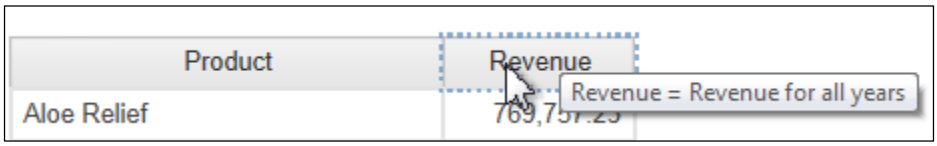
   The Revenue column header title appears between two < HTML item > objects.

3. Double-click the left **< HTML item >**, and then type the following HTML code in the text box:

   ****
4. Click **OK**, and then double-click the right **< HTML item >**.
5. In the **HTML** dialog box, type ****, and then click **OK**.
6. Run the report in **HTML**.

   Note that as you hover your mouse cursor over the **Revenue** column header, that the tooltip which you created appears beside the mouse cursor, as follows:

7. Close the rendered report tab.

Task 3. Create a tooltip from a report expression.

You will create a tooltip that displays the appropriate product description when the mouse cursor hovers over a product. You need to add the Description query item to the list, and then create an HTML Item that uses a report expression to reference the required data from the query.

1. On the **Navigate** tab, click the **Query explorer** tab, and then click **Query1**.
2. On the **Application** bar, open the **Properties** pane.
3. In the **Properties** pane, under the **MISCELLANEOUS** section, rename the query as **Product Revenue Query**.
4. On the **Data/Source** tab, from the **Products** query subject, drag **Product description** to the **Data Items** pane.
5. On the **Navigate** tab, click the **Page explorer** tab, and then click **Page1**.
6. Click the list **Container selector** to select the entire list.

7. In the **Properties** pane, under the **DATA** section, update the **Properties** option to include **Product description**.
8. From the **Toolbox** tab, under **ADVANCED**, drag an **HTML item** to the left of an item in the **<Product>** column body, and then drag an **HTML item** to the right of an item in the **<Product>** column body.

   An < HTML Item > object appears on each side of each data item.

9. Click an **< HTML item >** object to the left of a **<Product>** item, to select it.
10. In the **Properties** pane, under the **HTML SOURCE** section, change the **Source type** property to **Report expression**, and then double-click the **Report expression** property.
11. In the **Expression Definition** box, create and validate the following code:

    ''

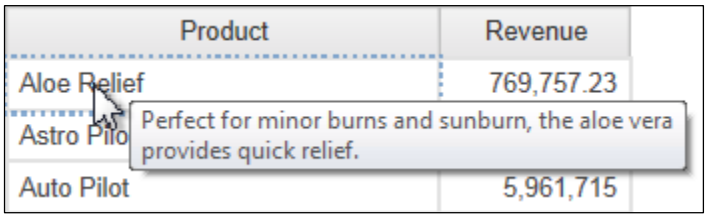
    The first '" is a double quote followed by a single quote, and the last "' is a single quote followed by a double quote.
12. Click **OK** to close the **Report expression** dialog box.
13. Double-click an **< HTML item >** object to the right of a **<Product>** item, in the **HTML** dialog box, type ****, and then click **OK**.

    HTML items are not case sensitive.

Task 4. Test the tooltip.

1. Run the report in **HTML**, and point to the Aloe Relief product to see the tooltip that you created.

   A section of the results appear as follows:



2. Point to another product.

   The tooltip now displays the description for this product. When IBM Cognos Analytics - Reporting created the HTML version of this report, it created a copy of the tooltip code for every row in the Product column. Each instance of the code passes the Description query item related to the Product to the generic tooltip function.
3. Close the rendered report tab.

Task 5. Create a tooltip from a data item.

While the Revenue column header tooltip indicates that the values in that column are based on revenue for all years, not all years contain the same products. Some products are discontinued, and new products are introduced in different years. You will create another tooltip in the Revenue column to display the years that the revenue is based on

1. On the **Navigate** tab, click the **Query explorer** tab, and then click **Product Revenue Query**.
2. From the **Toolbox** tab, drag a **Data Item** into the **Data Items** pane.
3. In the **Data item expression** dialog box, in the **Name** field, type **Revenue ToolTip**.
4. the **Expression Definition** pane, create and validate the following HTML code:

   ' '

   Hints:

   - Data Items tab: [Year].
   - Functions tab:
     - Common Functions: A-C: cast
     - Summaries: minimum, maximum

   This data item references the year data, and for all the years with revenue data for a product, it will determine what the earliest year is, and what the latest year is.

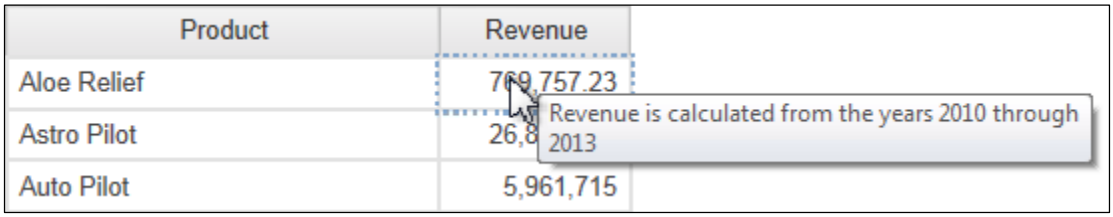
   The minimum() function will determine the minimum year, maximum() function determines the maximum year, and the cast() function coverts the data expression to a string of 4 characters.
5. Click **OK** to close the **Data item expression** dialog box.
6. On the **Navigate** tab, click the **Page explorer** tab, and then click **Page1**.
7. From the **Toolbox** tab, under **ADVANCED**, drag an **HTML item** to the left of an item in the **<Revenue>** column body, and then drag an **HTML item** to the right of an item in the **<Revenue>** column body.

8. Click the **< HTML item >** to the left of a **<Revenue>** list column body item, in the **Properties** pane in the **HTML SOURCE** section, change the **Source type** property to **Data item value**, and then change the **Data item value** property to **Revenue ToolTip**.
9. Double-click the **< HTML item >** to the right of a **<Revenue>** column body item, in the **HTML** dialog box, type ****, and then click **OK**.
10. On the **Application** bar, under **More**, click **Unlocked** to lock the cells of the report.

Task 6. Test the tooltip.

1. Run the report in **HTML**, and then hover the mouse cursor over the revenue for **Aloe Relief** in the first row of the **Revenue** column.

   A custom tooltip is displayed to indicate the revenue is derived from sales from 2010 through 2013.



2. Hover the mouse cursor over the revenue for **Astro Pilot**.

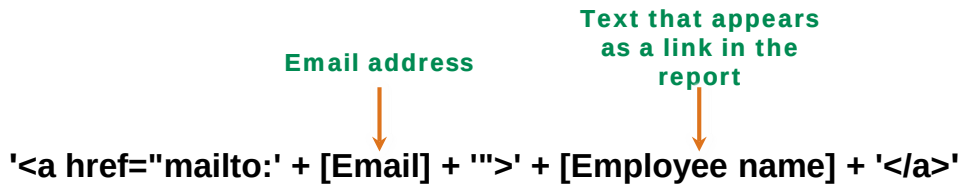
   The tooltip now displays the years used in this revenue value for this specific product. When IBM Cognos Analytics - Reporting created the HTML version of this report, it created a copy of the tooltip code for every row in the Revenue column. Each instance of the code referenced the Revenue ToolTip data item and passed the content of this item to the generic tooltip function.
3. Hover the mouse cursor over **Auto Pilot** to see the years included in the revenue value.
4. Close the rendered report tab.
5. Remove the report without saving it.
6. Leave the **IBM Cognos Analytics** portal open for the exercise.

Results:
You created tooltips in your report using HTML items. Your tooltips described the revenue column, both in terms of the general statement for the revenue column that was displayed, details of the years included in the revenue values for each product, as well as a tooltip that displayed descriptions for each product.

IBM Training

Send emails using links in a report

- You can replace a data item with an HTML item in order to create a hyperlink that opens an email when you click the data item in the report.



Enhance user interaction with HTML

Send emails using links in a report

The query must include the item you want to appear as a link in the report (such as Employee name), and the query item containing the email addresses of the recipients.

Unlock the report, cut the Employee name query item out of the column, and then add an HTML Item to the column instead.

Modify the Report Expression of the HTML to create a "mailto" link Item to open an email window, specify what to display in the To box of the email, and what item will appear as a link in the report.

IBM Training

Unit summary

- Create tooltips that clarify report data
- Send emails using links in a report

Unit summary

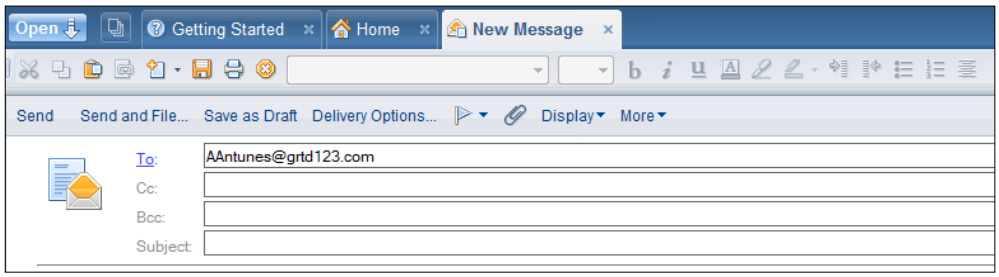
IBM Training

Exercise 1

Send emails using links in a report

Employee Contact Information

Employee name	Position name	Work phone	Extension
Aaghie Heiman	Information Technology Manager	+(41) 17 12 13 11	9676
Aaghie Heiman	Software Engineer	+(41) 17 12 13 11	9676
Aaltje Hansen	Level 1 Sales Representative	+(41) 17 12 13 11	9640
Abel Antunes	Product Manager	+55 (11) 344-4444	2605
Abram Ruiz	Level 2 Sales Representative	+(41) 17 12 13 11	9762



Enhance user interaction with HTML

Exercise 1: Send emails using links in a report

Exercise 1: Send emails using links in a report

Management has requested an employee contact list. They would like to be able to contact employees by email easily. To facilitate this, as report author Frank Bretton, you will create a list report where the employee names are hyperlinks that will launch a new blank email populated with the address of that employee.

To accomplish this:

- Create a list report with using the GO data warehouse (query) package, Sales and Marketing (query), Sales (query) with Employee name, Position name, Work phone, Extension, and Email displayed.
- Make the Email and Employee name data items properties of the report.
- Cut the Email column and the Employee name data item (not the column title) from the report.
- Rename the query as Employee Contact Information Query.
- Drag an HTML Item to the top cell of the Employee name column, and make it a report expression.
- Type the report expression as follows: '' + [Employee Contact Information Query].[Employee name] + ''.
- Type the title of the report: Employee Contact Information.

Run the report and click Abel Antunes. If prompted to log in to email, accept the default account and use the password Education1.

For more information about where to work and the IBM Cognos Analytics - Reporting results, refer to the Tasks and Results section that follows. If you need more information to complete a task, refer to previous content and demonstrations for detailed steps.

Exercise 1: Tasks and Results

Task 1. Create a list report containing HTML item objects.

- **Side panel**: Open a new **List** template using the **GO data warehouse (query)** package.
- **Source tab**: Navigate to **Sales and Marketing (query)/Sales (query)/Employee by region**, and add the following query items to the list data container:
  - **Employee name**
  - **Position name**
  - **Work phone**
  - **Extension**
  - **Email**.

  The results appear as follows:

Employee name	Position name	Work phone	Extension	Email
<Employee name>	<Position name>	<Work phone>	<Extension>	<Email>

- **Properties** pane: For the list, add the **Employee name** and **Email** to the **Properties** option.
- **On demand toolbar**: Cut the **<Email>** list column body.
- **Navigate** tab: Navigate to **Query1**.
- **Properties** pane: Rename the query as **Employee Contact Information Query**.
- **Navigate** tab: Navigate to **Page1**.
- **Application** bar: Unlock the report.
- **On demand toolbar**: Cut the **<Employee name>** list column body.

  The results appear as follows:

Employee name	Position name	Work phone	Extension
	<Position name>	<Work phone>	<Extension>

- **Toolbox** tab: Add an **HTML item** into the top, empty cell of the **Employee name** list column body.

Task 2. Define the HTML item and format the report.

- **Properties** pane: For the **< HTML item** >, change the **Source type** property to **Report expression**.
  - Open the **Report expression** property, and then create and validate the following expression:

    '' + [Employee Contact Information Query].[Employee name] + ''
- **Work area**: Add the following title to the header block:
  Employee Contact Information.
- **Toolbar**: Align the header block left.
- **Application** bar: Lock the report.
- **Navigate** tab: Navigate to the **Employee Contact Information Query**.
- **Properties** pane: For the **Employee name** data item, change the **Pre-sort** property to **Sort ascending**.
- **Application** bar: Run the report in **HTML**.

  A section of the report appears as follows:

Employee Contact Information

Employee name	Position name	Work phone	Extension
Aaghie Heiman	Information Technology Manager	+(41) 17 12 13 11	9676
Aaghie Heiman	Software Engineer	+(41) 17 12 13 11	9676
Aaltje Hansen	Level 1 Sales Representative	+(41) 17 12 13 11	9640
Abel Antunes	Product Manager	+55 (11) 344-4444	2605
Abram Ruiz	Level 2 Sales Representative	+(41) 17 12 13 11	9762

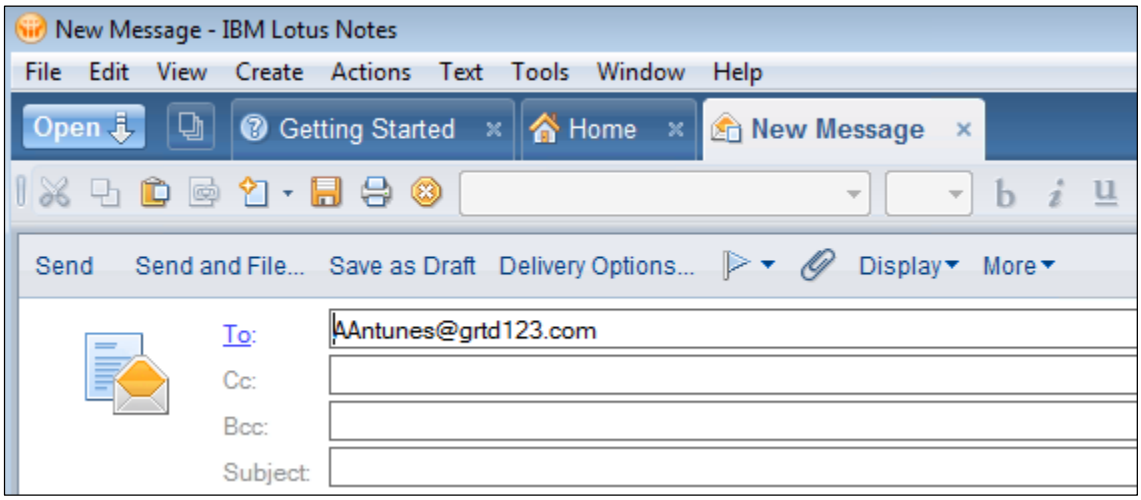
- **Employee Contact Information** report: Click **Abel Atunes**.
- **Lotus Notes** authentication window: accept the default **User name**, type the password **Education1**, and then click **Log In**.

  You may need to maximize IBM Lotus Notes from the taskbar, and then click the New Message tab.

- **New Message tab**: Review the generated email.

  The e-mail is addressed to AAntunes@grtd123.com, and will be sent by your currently logged-in account.

  The results appear as follows:

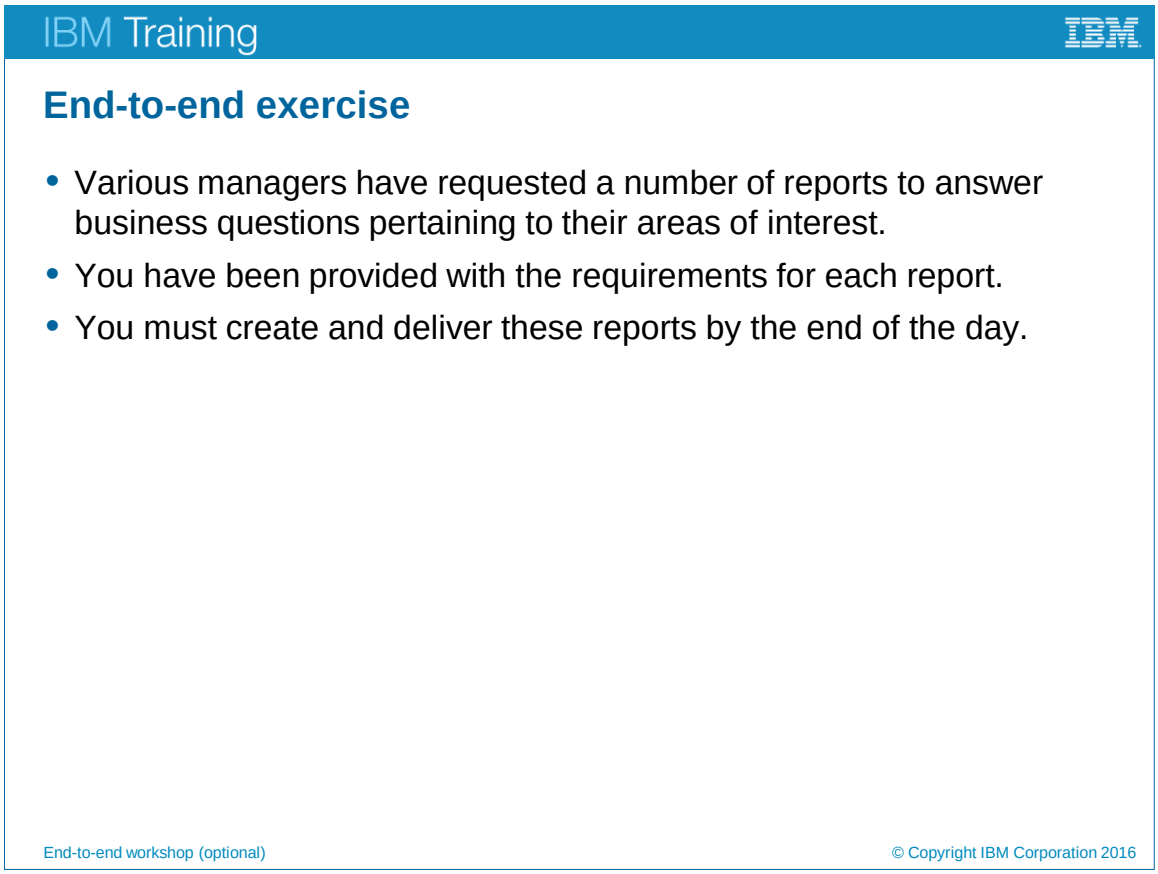


- Close the e-mail without sending or saving it.
- Close **IBM Lotus Notes**.
- Sign out of the **IBM Cognos Analytics** portal.
- Close the web browser.

IBM Training

End-to-end exercise (optional)

IBM Cognos Analytics (v11.0)



End-to-end exercise

In this exercise, you will:

- create reports to meet various business requirements
- reinforce concepts learned throughout the course

Report 1: Burst a sales report by country

User\Password: **brettonf\Education1**

Optionally, you may name the queries.

Sales managers want to review details about sales in 2012. They want to view a list displaying revenue generated and quantity sold for sales for each retailer type. Within this list, they want to view a pie chart showing revenue generated by each product line for each retailer type. All of this information has to be displayed for each country, and then burst to the Web so that users can only see data for their country. For example, if a sales rep from Australia logs into IBM Cognos Analytics portal, they must be able to view a report containing only Australian data.

To accomplish this:

- Create a list report using the GO data warehouse (query) package, Sales and Marketing (query) folder.
- Add the following query items from the Sales (query) namespace:
  - Employee by region\Country
  - Retailers\Retailer type
  - Sales fact\Quantity and Revenue
- Group Country, and then make Country a header instead of a column. Add totals for Quantity and Revenue.
- Filter the list to only include data for orders placed in 2012.
- Verify that legacy chart authoring is turned off, add a pie chart with 3-D Effects and Rounded Bevel chart to the list, and have it display Revenue by Product line. Filter the chart to only show data for orders placed in 2012.
- Add the recipients from the Burst table by country query subject and country code from the Employee by region query subject to Query1, and then make these data items properties of the list.
- Add the retailer type from the Retailers query subject and country code from Employee by region query subject to Query2.
- Create a master-detail relationship with two links between the two queries, so that the country code in the list matches country code in the chart and retailer type matches retailer type.
- Change the report title to "2012 Sales Report by Retailer Type and Product line", and make the Country item in the list header 12 pt, bold, and italic.
- Set up the burst options to burst on country and to distribute the report to recipients as directory entries.

- Save the report as Report1.
- Burst the Report1 report to generate three outputs: one each for United States, Australia, and Italy.
- Sign out of the IBM Cognos Analytics portal, sign on as scottb\Education1 from the United States, and view the data in the report.
- Sign out of the IBM Cognos Analytics portal, sign on as sindenj\Education1 from Australia, and view the data in the report.
- Sign out of the IBM Cognos Analytics portal.

For more information about where to work and the exercise results, refer to the Tasks and results section that follows. If you need more information to complete a task, refer to earlier demonstrations for detailed steps.

Report 1: Tasks and results

Portal: **http://vclassbase:9300/bi**
User/Password: **brettonf/Education1**
Application: **IBM Cognos Analytics - Reporting**
Package: **Team content\Samples\Models\GO data warehouse (query)**
Folder: **Sales and Marketing (query)**
Namespace: **Sales (query)**

Task 1. Create a list report that displays country sales by retailer.

- **Side panel**: Open a new **List** template, using the **GO data warehouse (query)** package.
- **Source tab**: Navigate to **Sales and Marketing (query) / Sales (query)**
  - **Employee by region**: Add the **Country** query item to the list.
  - **Retailers**: Add the **Retailer type** query item to the list.
  - **Sales fact**: Add the **Quantity** and **Revenue** query items to the list.

  The results appear as follows:

Country	Retailer type	Quantity	Revenue
<Country>	<Retailer type>	<Quantity>	<Revenue>

- **Toolbar**: Group the **<Country>** list column body.
  - Create a header for **<Country>**.
  - Delete the redundant **<Country>** list column body.
  - Summarize **<Quantity>** and **<Revenue>** for **Total**.
  - Create and validate the following advanced detail filter: **[Sales (query)].[Time].[Year]=2012**.

  The results appear as follows:

Retailer type	Quantity	Revenue
<Country>		
<Retailer type>	<Quantity>	<Revenue>
<Country> - Total	<Total(Quantity)>	<Total(Revenue)>
<Country>		
<Retailer type>	<Quantity>	<Revenue>
<Country> - Total	<Total(Quantity)>	<Total(Revenue)>
Overall - Total	<Total(Quantity)>	<Total(Revenue)>

  - Run the report in **HTML**.

  A section of the results appear as follows:

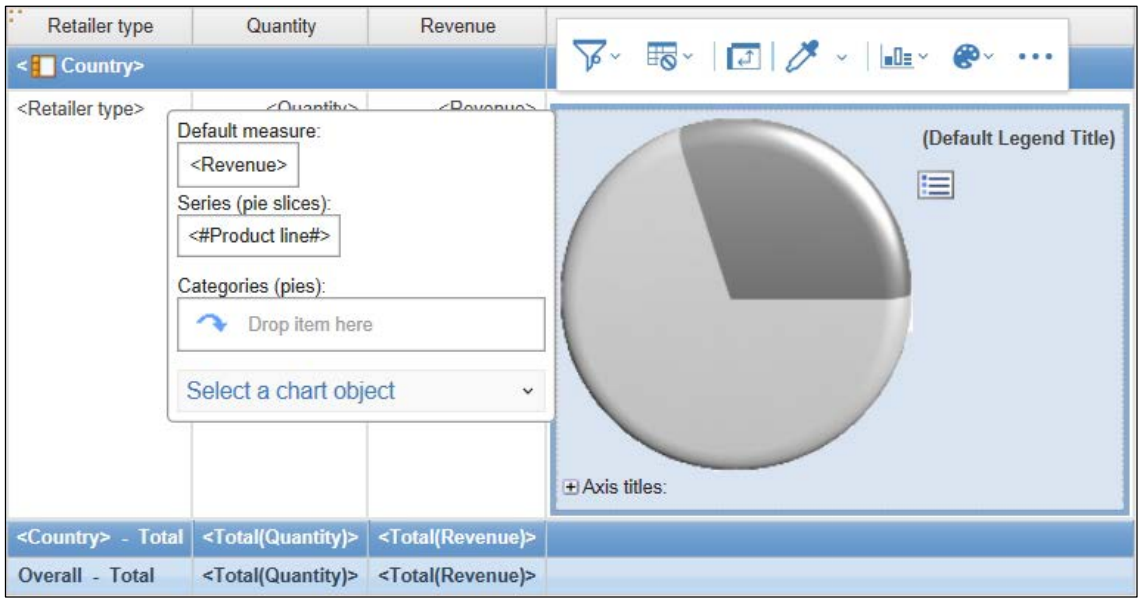
Retailer type	Quantity	Revenue
Australia		
Department Store	207,589	10,249,363.28
Direct Marketing	70,429	1,723,984.95
Equipment Rental Store	18,679	998,328.88
Eyewear Store	22,024	1,880,527.78
Golf Shop	43,591	6,574,888.17
Outdoors Shop	265,525	13,877,293.83
Warehouse Store	54,792	3,664,415.73
Australia - Total	682,629	38,968,802.62

- Close the rendered report tab.

Task 2. Add a pie chart to the list.

- **Application** bar/**More/Options**: Verify that the **Use legacy chart authoring** check box is cleared.
- **Toolbox** tab: Create a **Pie chart with 3-D Effects and Rounded Bevel**, based on **Query2**, and place it to the right of the **<Revenue>** list column body in the list data container.
- **Source** tab:
  - From **Sales fact**, add **Revenue** as the **Default measure**.
  - From **Products**, add **Product line** to the **Series (pie slices)** drop zone.
- **On demand toolbar**: Create and validate the following advanced detail filter for **Query2**: **[Sales (query)].[Time].[Year]=2012**

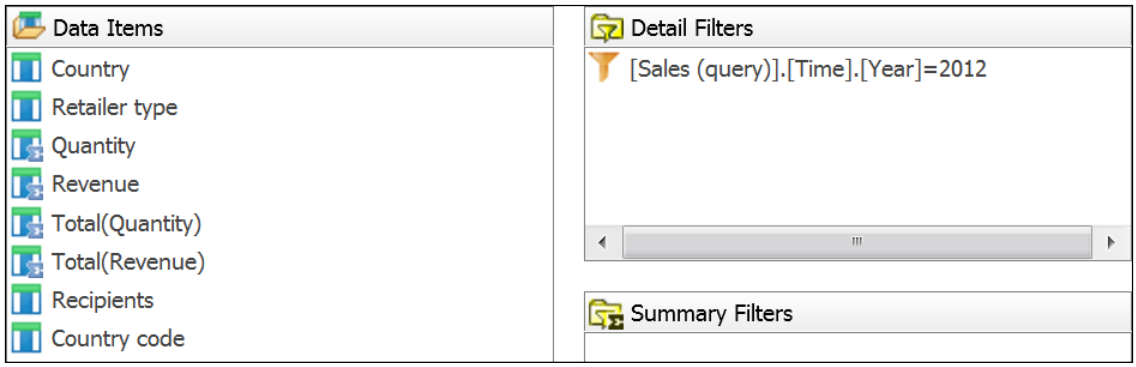
  The results appear as follows:



Task 3. Add Recipients and Country code to Query1.

- **Navigate** tab: Navigate to **Query1**.
- **Source tab**:
  - Navigate to **Sales and Marketing (query)/Sales (query)/Burst table by country**, and then add the **Recipients** query item to the **Data Items** pane.
  - Navigate to **Sales and Marketing (query)/Sales (query)/Employee by region/Codes**, and then add the **Country code** query item to the **Data Items** pane.

    The results appear as follows:

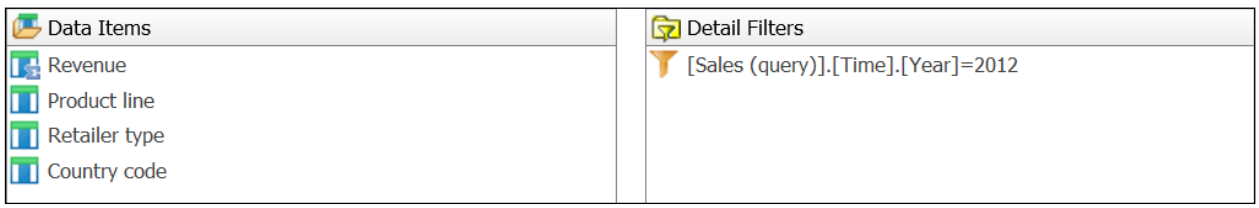


- **Navigate** tab: Navigate to **Page1**.
- **Properties** pane: Make the **Recipients** and **Country code** data items properties of the list.

Task 4. Add Retailer type and Country code to Query2.

- **Navigate** tab: Navigate to **Query2**.
- **Source tab**:
  - Navigate to **Sales and Marketing (query)/Sales (query)/Retailers**, and then add the **Retailer type** query item to the **Data Items** pane.
  - Navigate to **Sales and Marketing (query)/Sales (query)/Employee by region/Codes**, and then add the **Country code** query item to the **Data Items** pane.

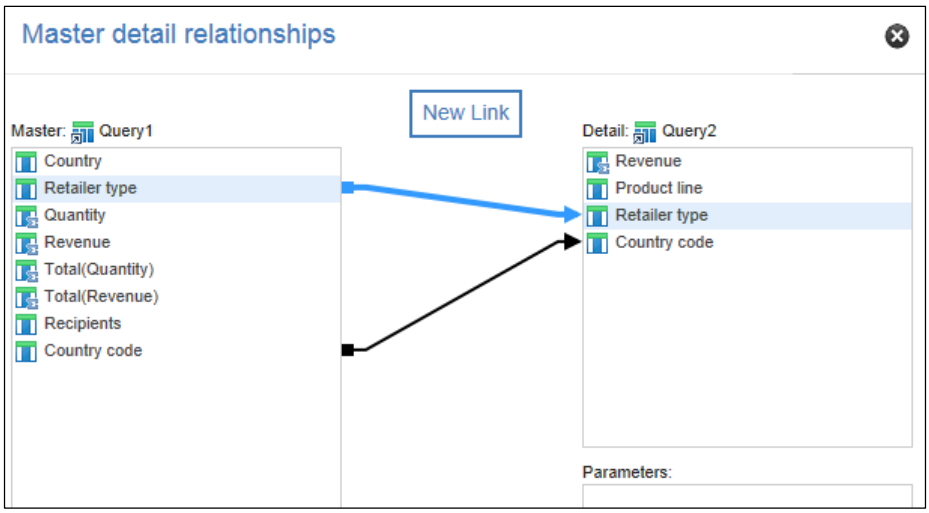
    The results appear as follows:



Task 5. Create a master detail relationship between the queries, add a title, and format the header.

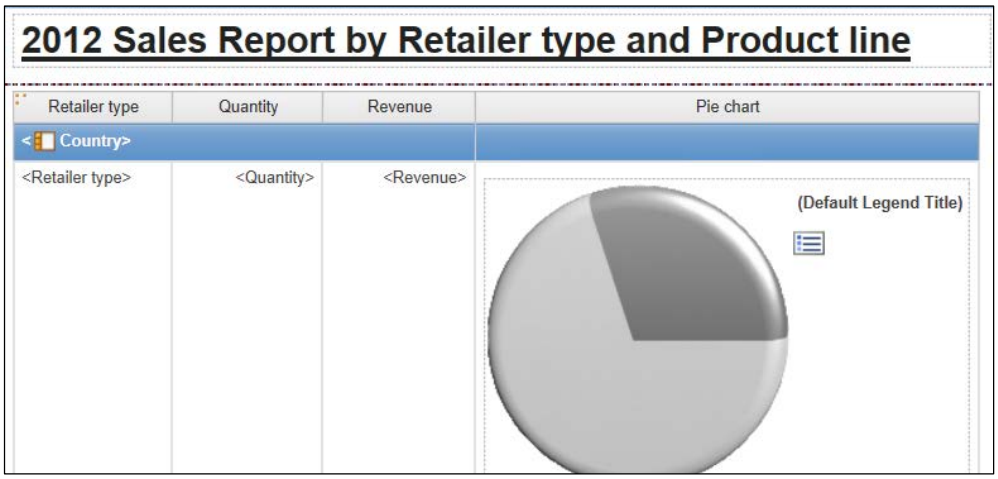
- **Navigate** tab: Navigate to **Page1**.
- Select the **Pie chart**.
- **On demand toolbar/More**:
  - Create a new master detail relationship link between the **Country code** data items in each query.
  - Create a new link between the **Retailer type** data items in each query.

    The results appear as follows:



- **Report Header**: Add the following title:
  2012 Sales Report by Retailer type and Product line.
- **On demand toolbar**:
  - Set the title font to **18 pt**, **Bold**.
  - Left-justify the title block.

    A section of the results appears as follows:



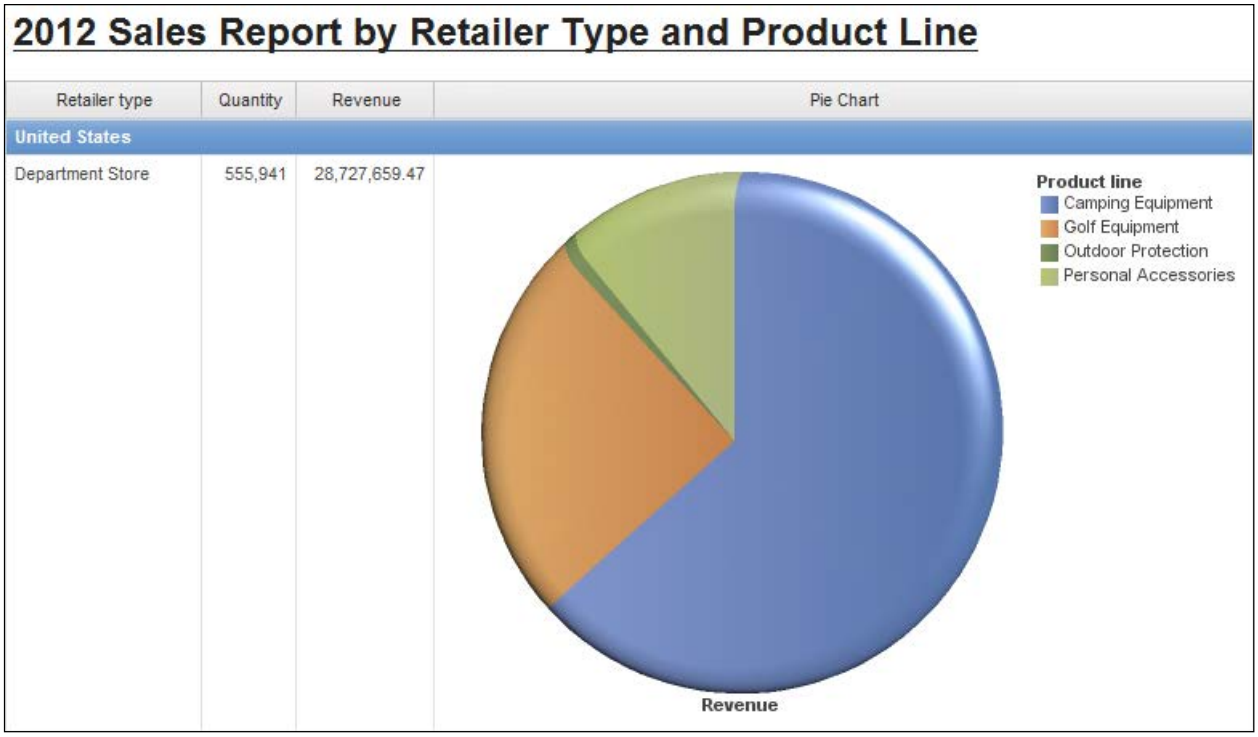
Task 6. Set the burst options for the report.

- **Properties** pane/**Report**: With the list selected, set the following burst options:
  - **Burst options**: Make the report available for bursting.
  - **Burst Groups**:
    - **Query**: Query1
    - **Label**: Country
    - **Groups**: Country
  - **Burst Recipient**:
    - **Query**: Query1
    - **Data Item**: Recipients
    - **Type**: Directory entries
- **Application** bar: Save the report in the **Team content**/**B6059** folder as **Report1**.
- Remove the **IBM Cognos Analytics - Reporting** report.

Task 7. Set the burst options for the report and test.

- **IBM Cognos Analytics** portal: Navigate to **Team content/B6059**.
- **Report1/More**: Click **Run as**, set the following advanced options**:**
  - Run in background.
  - Burst the report.
- **Application** bar: Sign out.
- **Sign in** window: Sign in as: **scottb\Education1**
- **IBM Cognos Analytics** portal: Navigate to the **B6059** folder.
  - Run the **Report1** report.

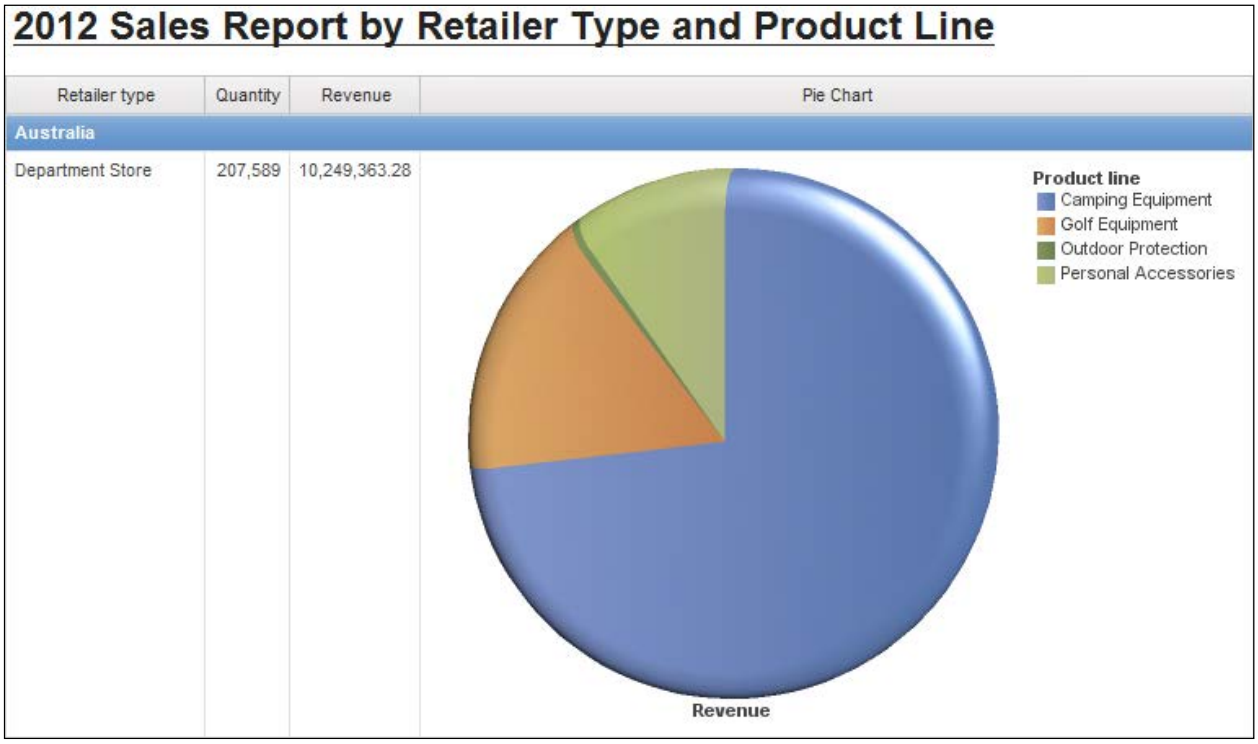
  A section of the results appears as follows:



- **Application** bar: Sign out as **Bart Scott**.

- **Sign in** window: Sign in as **sindenj\Education1**.
- **IBM Cognos Analytics** portal: Navigate to the **B6059** folder.
  - Run the **Report1** report.

  A section of the results appear as follows:



- **Application** bar: Sign out.
- Sign in as **brettonf/Education1**
- Leave the **IBM Cognos Analytics** portal open for the next report.

Report 2: Create a report letting users choose top and bottom 'N' product types

User\Password: **brettonf\Education1**

Optionally, you may name the queries.

Marketing wants to know how the company's product types are performing based on gross profit. You have been asked to create a report that shows the top-performing and bottom-performing product types. At runtime, the managers want to be able to select the number of top-performing product types and the number of bottom-performing product types that will appear in the report.
To accomplish this:

- Create a list report using the GO data warehouse (query) package, Sales and Marketing (query) folder, Sales (query) namespace that includes Product line, Product type, and Gross profit.
- Add a calculated column called Rank1 (Label: Rank) that ranks the gross profit generated by each product type.
- Add a table with one column and two rows, move the existing list report to the top cell, and then add a new list to the bottom cell.
- Add Product line, Product type, and Gross profit to the bottom list.
- Add a calculated data item called NegGP to the query used to create the second list. This data item multiplies the gross profit generated by each product type by negative one.
- Add a calculated column called Rank2 (Label: Rank) to the bottom list that ranks the values in the NegGP column.
- Filter the top list to only show product types whose gross profit rank is less than or equal to the rank number provided by the user, and then do the same for the bottom list.
- Add a dynamic title that changes depending on the value entered using the prompt. The title appears as follows: <prompt value displayed here> Top-performing and Bottom-performing Product Types.
- Format the entire title so all the text looks the same.
- Add text before the top list and before the bottom list describing their contents. Make this text dynamic so that it reflects the number of products users select using the prompt.
- Make all explanatory text above the list reports bold.

- Create space between the two lists by adding 20 pixels of padding to the top of the cell holding the bottom list report.
- Sort the Rank columns in both lists in ascending order.
- Create a prompt page that lets users provide the number of top and bottom product types to view. Include instructions above the prompt asking for the number of top and bottom product types to view.
- Run the report and enter 5 as the prompt value.

For more information about where to work and the exercise results, refer to the Tasks and results section that follows. If you need more information to complete a task, refer to earlier demonstrations for detailed steps.

Report 2: Tasks and results

Portal: **http://vclassbase:9300/bi**
User/Password: **brettonf/Education1**
Application: **IBM Cognos Analytics - Reporting**
Package: **Team content\Samples\Models\GO data warehouse (query)**
Folder: **Sales and Marketing (query)**
Namespace: **Sales (query)**

Task 1. Create a list report that ranks product types by how much gross profit they generate.

- **IBM Cognos Analytics** portal**:** Open a new **List** template using the **GO data warehouse (query)** package.
- **Source** tab: Navigate to **Sales and Marketing (query)/Sales (query)**
  - **Products**: Add **Product line** and **Product type** query items to the list report.
  - **Sales fact**: Add the **Gross profit** measure to the list report.
- **Toolbox** tab/**TEXTUAL**: Add a query calculation to the end of the list data container called **Rank1**, and create and validate the following expression:

  rank([Gross profit])
- **Properties** pane: Rename the column label to **Rank**.

  The results appear as follows:

Product line	Product type	Gross profit	Rank
<Product line>	<Product type>	<Gross profit>	<Rank1>

Task 2. Add a table and a second list for bottom-performing product types.

- **Toolbox** tab: Add a table object to the report layout below the list, with **1** column and **2** rows.
- **Work area**: Move the list to the top cell of the table.
- **Toolbox** tab: Add a second list report object to the bottom cell.
- **Source** tab: Navigate **Sales and Marketing (query)/Sales (query)**
  - **Products**: Add **Product line** and **Product type** query items to the new list report object.
  - **Sales fact**: Add the **Gross profit** measure to the list report object.
- **Navigate** tab: Navigate to **Query2**.
- **Toolbox** tab: Add a query calculation called **NegGP**, and then create and validate the following expression:

  [Gross profit] * -1
- **Navigate** tab: Navigate to **Page1**.
- **Properties** pane: Make **NegGP** a property of the **Query2** list.
- Create and validate an additional query calculation, placed at the end of the new list report object, called **Rank2**, using the following expression: **rank([NegGP])**.
- **Properties** pane: Rename the column label to **Rank**.

  The results appear as follows:

Product line	Product type	Gross profit	Rank
<Product line>	<Product type>	<Gross profit>	<Rank1>
<Product line>	<Product type>	<Gross profit>	<Rank1>
<Product line>	<Product type>	<Gross profit>	<Rank1>
Product line	Product type	Gross profit	Rank
<Product line>	<Product type>	<Gross profit>	<Rank2>
<Product line>	<Product type>	<Gross profit>	<Rank2>
<Product line>	<Product type>	<Gross profit>	<Rank2>

Task 3. Filter both lists so that users can choose the number of top-performing and bottom-performing product types to view.

- **On demand toolbar**: Create and validate an advanced detail filter, for the top list, using the following expression**:**

  [Rank1]<=?RankNumber?. (Validate using the number 1 if required.)
- **Filters - Query1** pane: Set filter application to **After auto aggregation**.
- **On demand toolbar**: Create and validate an advanced detail filter, for the bottom list, using the following expression**: [Rank2]<=?RankNumber?**.
- **Filters - Query2** pane: Set filter application to **After auto aggregation**.

Task 4. Add an explanatory report title.

- **Report header block**: Create the following report title: press the spacebar, and then type: **Top-Performing and Bottom-Performing Product types**.
- **Toolbox** tab: Add a **Layout calculation** to the beginning of the report title text using the **RankNumber** parameter as the report expression.
- **Header:** Format the layout calculation text with same format as the title.
  - Left-justify the header block.

    The results appear as follows:

<%ParamDisplay...%> Top-Performing and Bottom-Performing Product types

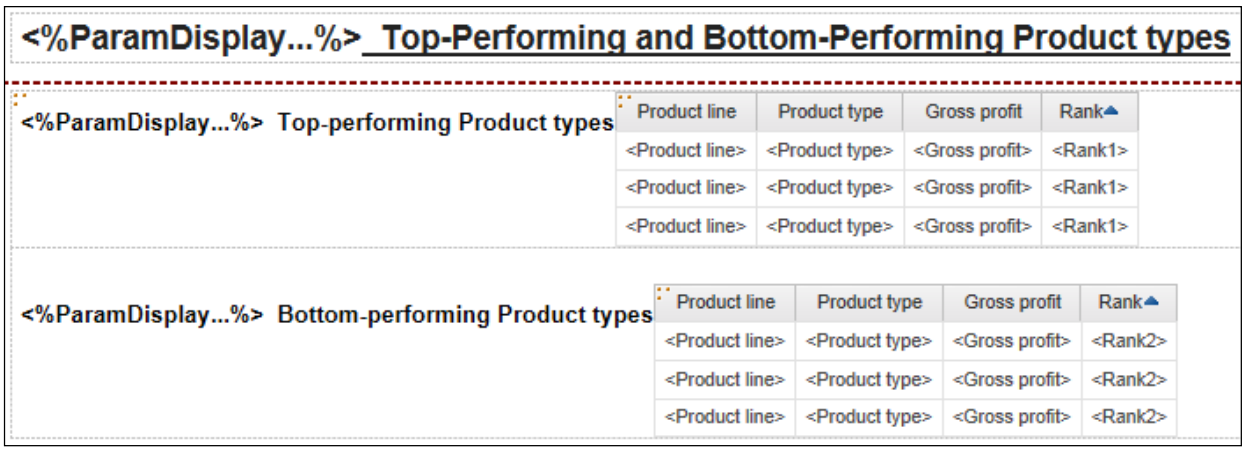
Product line	Product type	Gross profit	Rank
<Product line>	<Product type>	<Gross profit>	<Rank1>
<Product line>	<Product type>	<Gross profit>	<Rank1>
<Product line>	<Product type>	<Gross profit>	<Rank1>

Product line	Product type	Gross profit	Rank
<Product line>	<Product type>	<Gross profit>	<Rank2>
<Product line>	<Product type>	<Gross profit>	<Rank2>
<Product line>	<Product type>	<Gross profit>	<Rank2>

Task 5. Format the report and add explanatory text above the reports.

- **Toolbox** tab: Add the following **Text item** object to the left of the top list data container, inside the top table cell: Press the spacebar and then type **Top-performing Product types**.
  - Add a **Layout calculation** to the left of the text item that you have just added.
- **Report expression**: Use the **RankNumber** parameter for the report expression.
- **Toolbox** tab: Add the following **Text item** object to the left of the bottom list report object, inside the top table cell: Press the spacebar and then type **Bottom-performing Product types**.
  - Add a **Layout calculation** to the left of the text item that you have just added.
- **Report expression**: Use the **RankNumber** parameter for the report expression.
- **Toolbar**: Format both of text items and layout calculations to **Bold**.
- **Properties** pane: Add padding (**20px**) to the top of the bottom table cell.
- **On demand toolbar**:
  - Sort the **<Rank1>** list column body in **Ascending** order.
  - Sort the **<Rank2>** list column body in **Ascending** order.

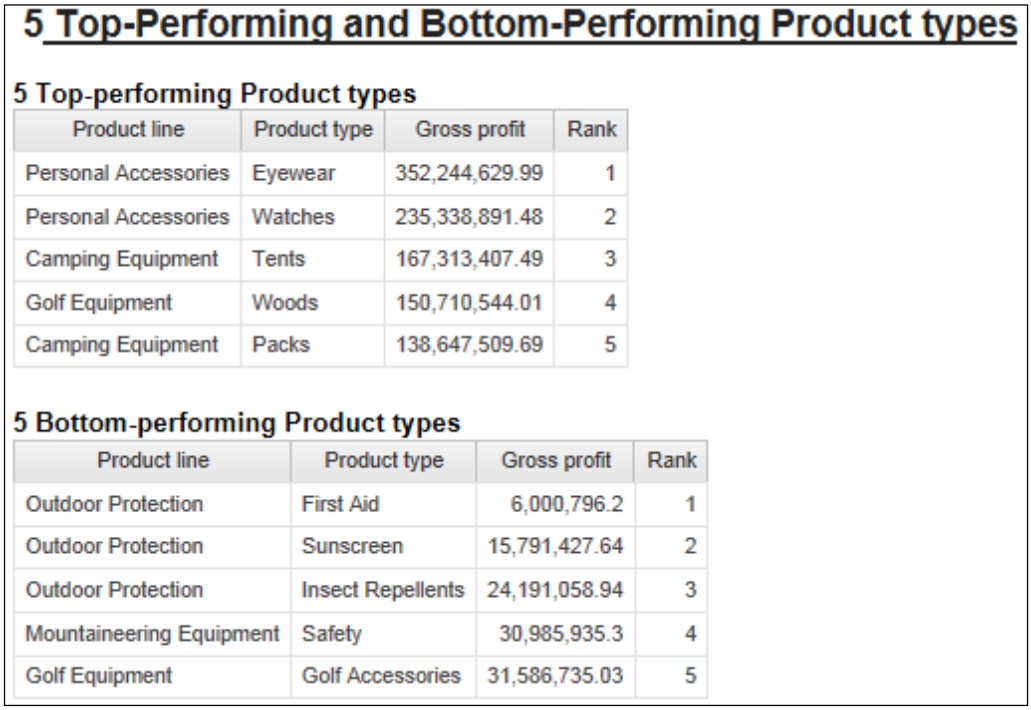
The results appear similar to the following:



Task 6. Add a prompt page

- **Navigate** tab: Navigate to **Prompt pages**.
- **Toolbox** tab: Add a **Page** object to the **Prompt pages** pane.
- **Prompt pages**: Open **Prompt page1**.
- **Toolbox** tab: Add a **Table** object to the **Work area** with **1** column **2** rows.
  - Add a **Text box prompt** to the bottom cell using the **RankNumber** parameter.
  - Add the following **Text Item** to the top cell: **Select the number of top-performing and bottom-performing product types to view:**
- **On demand toolbar**: Make the newly added text **Bold**.
  - Run the report in **HTML**.
- **Prompt**: Set the prompt value to **5**.

  The results appear as follows:



5 Top-Performing and Bottom-Performing Product types

5 Top-performing Product types

Product line	Product type	Gross profit	Rank
Personal Accessories	Eyewear	352,244,629.99	1
Personal Accessories	Watches	235,338,891.48	2
Camping Equipment	Tents	167,313,407.49	3
Golf Equipment	Woods	150,710,544.01	4
Camping Equipment	Packs	138,647,509.69	5

5 Bottom-performing Product types

Product line	Product type	Gross profit	Rank
Outdoor Protection	First Aid	6,000,796.2	1
Outdoor Protection	Sunscreen	15,791,427.64	2
Outdoor Protection	Insect Repellents	24,191,058.94	3
Mountaineering Equipment	Safety	30,985,935.3	4
Golf Equipment	Golf Accessories	31,586,735.03	5

- Close the rendered report tab.
- Remove the report without saving it.
- Leave the **IBM Cognos Analytics** portal open for the next report.

Report 3:
Let users choose a chart type to display with a crosstab

User\Password: **brettonf\Education1**

Optionally, you may name the queries.

For an upcoming presentation, management has requested a report comparing the volume of sales of each product line in each sales territory. The report must show data in crosstab format and must show the same data in a graphical display. Depending on their preferences, users should be able to view the data graphically in a column, line, or gauge chart.
To accomplish this:

- Create a column chart, using the GO data warehouse (query) package, Sales and Marketing (query) folder, Sales (query) namespace, that displays the quantity of each product line sold in each sales territory.
- Add a clustered line with markers chart to the report using the same query, and add the same items as you did for the column chart.
- Add a gauge chart with bevelled border to the report using the same query, and add the same items you did for the column chart.
- Hide the horizontal axis titles on all of the charts.
- Add a prompt page with a value prompt and create a new parameter for the prompt called "DisplayOptions".
- Add three static choices for the prompt that let users choose to view a column chart, line chart, or gauge chart, and add explanatory text to let users know how to use the prompt.
- Add a Conditional blocks object to the report page below the three charts. Use a string variable to conditionally format this block to display the chart type that users select from the prompt at run time, and set the property of the conditional block object to associate each chart with the conditional block, with the gauge chart set as (Other).
- Add a crosstab to the bottom of the report and populate it using the same query and data that you added to the charts.
- Run the report for each of the prompt options.
- Save the report in the B6059 folder as Report3.

For more information about where to work and the exercise results, refer to the Tasks and Results section that follows. If you need more information to complete a task, refer to earlier demonstrations for detailed steps.

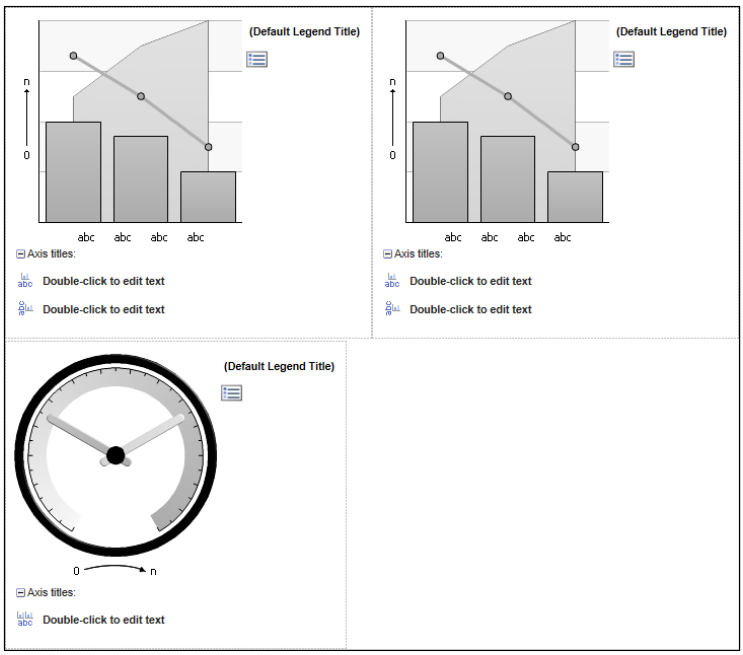
Report 3: Tasks and results

Portal: **http://vclassbase:9300/bi**
User/Password: **brettonf/Education1**
Application: **IBM Cognos Analytics - Reporting**
Package: **Team content\Samples\Models\GO data warehouse (query)**
Folder: **Sales and Marketing (query)**
Namespace: **Sales (query)**

Task 1. Create a chart report containing three display types.

- **Side panel**: Open a **Clustered Column** chart report using the **GO data warehouse (query)** package.
- **Source** tab: Navigate to **Sales and Marketing (query)/Sales (query)**
  - Sales fact: **Quantity** to **Default measure (y-axis)** drop zone.
  - Products: **Product line** to **Series (primary axis)** drop zone.
  - Retailers: **Region** to **Categories (x-axis)** drop zone.
- **Toolbox** tab:
  - Add a new **Clustered Line with Markers** chart object, using the same query and data as the first chart.
  - Add a new **Gauge chart with bevelled border** chart object, using the same query and data.
- **Properties** pane/**GENERAL**: Change the **Axis titles/Default titles** values to **No** for the column, line, and gauge charts.

  The results appear similar to the following:



Task 2. Add a prompt with static choices.

- **Navigate** tab: Navigate to **Prompt Pages**.
- **Toolbox** tab: Add a new **Page** object.
- **Prompt pages** pane: Open **Prompt page1**.
- **Toolbox** tab/**PROMPTING**: Add a **Value prompt** object to the **Work area**.
- **Prompt Wizard**: Create a new parameter called **DisplayOptions**.
- **Properties** pane: Create the following static choices for the value prompt parameter:

Use	Display
Column	Column
Line	Line
Gauge	Gauge

- **Toolbox** tab: Add a **Table** object with **1** column and **2** rows.
  - Add the following **Text item** object to top table cell:
    Please select a chart type:.
  - Drag the **Value prompt** to the bottom table cell.
- **On demand toolbar**: Make the text bold.

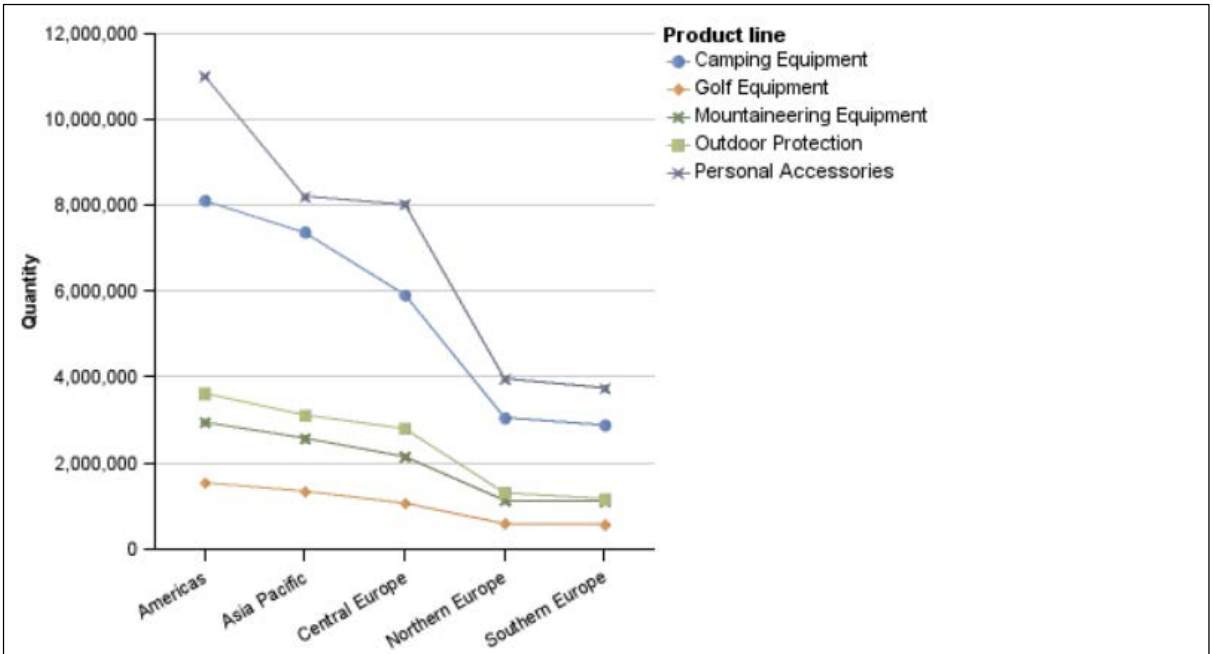
Task 3. Add a conditional block and apply conditional formatting to this conditional block.

- **Navigate** tab: Navigate to **Page1**.
- **Toolbox** tab/**ADVANCED**: Add a **Conditional blocks** object to the **Work area**, below the charts.
- **Properties** pane: Create a new **Block variable (string)** called **Chart**.
- **New Variable** pane: Add the following two variables: **Column**, and **Line**.
- **Expression Definition** pane: Create and validate the following expression: **ParamDisplayValue('DisplayOptions')**.
- **Properties** pane: For the **Conditional block** object, set the **Current block** property to **Column**.
- **Work area**: Drag the **Column** chart into the **Conditional block** object.
- **Properties** pane: For the **Conditional Block object**, set the **Current block** property to **Line**.
- **Work area**: Drag the **Line** chart into the **Conditional block** object.
- **Properties** pane: For the **Conditional block object**, set the **Current block** property to **(Other)**.
- **Work area**: Drag the **Gauge** chart into the **Conditional block** object.
- **Properties** pane: Validate by changing the current block to the different options.

Task 4. Add a crosstab to the report, and then run the report.

- **Toolbox** tab: Add a **Crosstab** object below the **Conditional block** object.
- **Object and query name** dialog box: Click **Query1** from the **Query Name** field list.
- **Data/Data items/Query1**:
  - Add the **Region** data item to the **Rows**.
  - Add the **Product line** data item to the **Columns**.
  - Add the **Quantity** data item to the **Measures** area.
- **Application** bar: Run the report in **HTML**.
- **Prompt**: Choose **Line**.

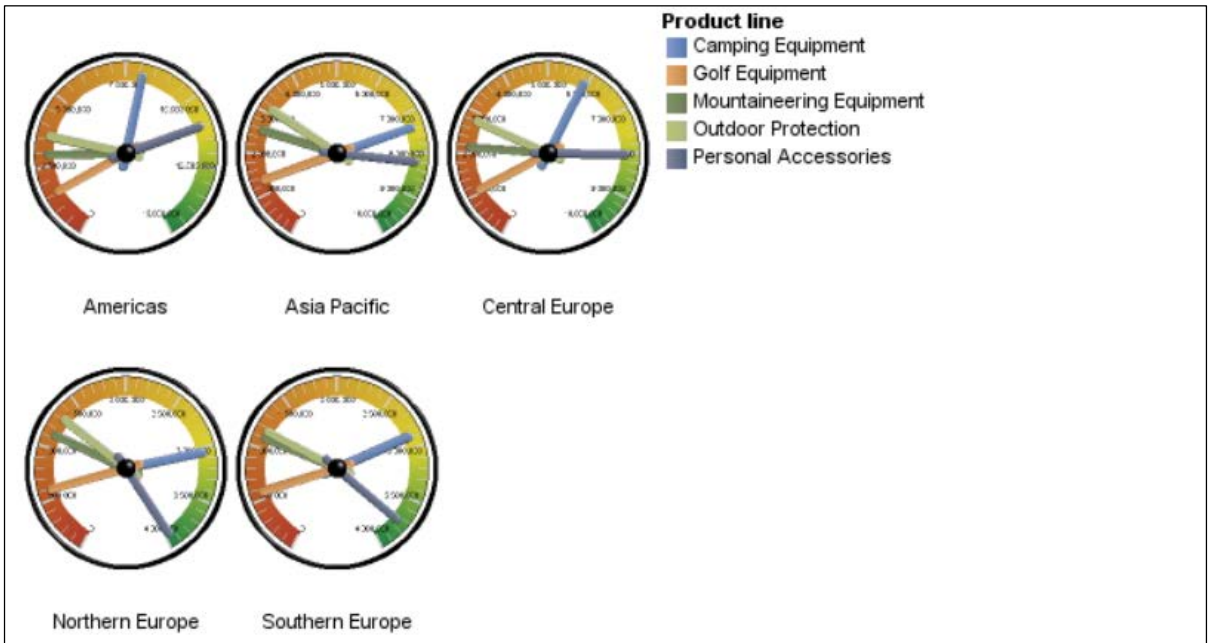
  The results appear as follows:



Quantity	Camping Equipment	Golf Equipment	Mountaineering Equipment	Outdoor Protection	Personal Accessories
Americas	8,101,682	1,544,411	2,948,533	3,619,457	10,998,468
Asia Pacific	7,366,131	1,338,406	2,571,299	3,114,960	8,198,744
Central Europe	5,904,428	1,071,235	2,146,207	2,800,923	8,016,774
Northern Europe	3,046,563	592,168	1,131,215	1,310,804	3,954,449
Southern Europe	2,882,345	567,481	1,102,837	1,168,301	3,739,270

- **Rendered report** tab: Run the report.
  - **Prompt**: Choose **Gauge**.

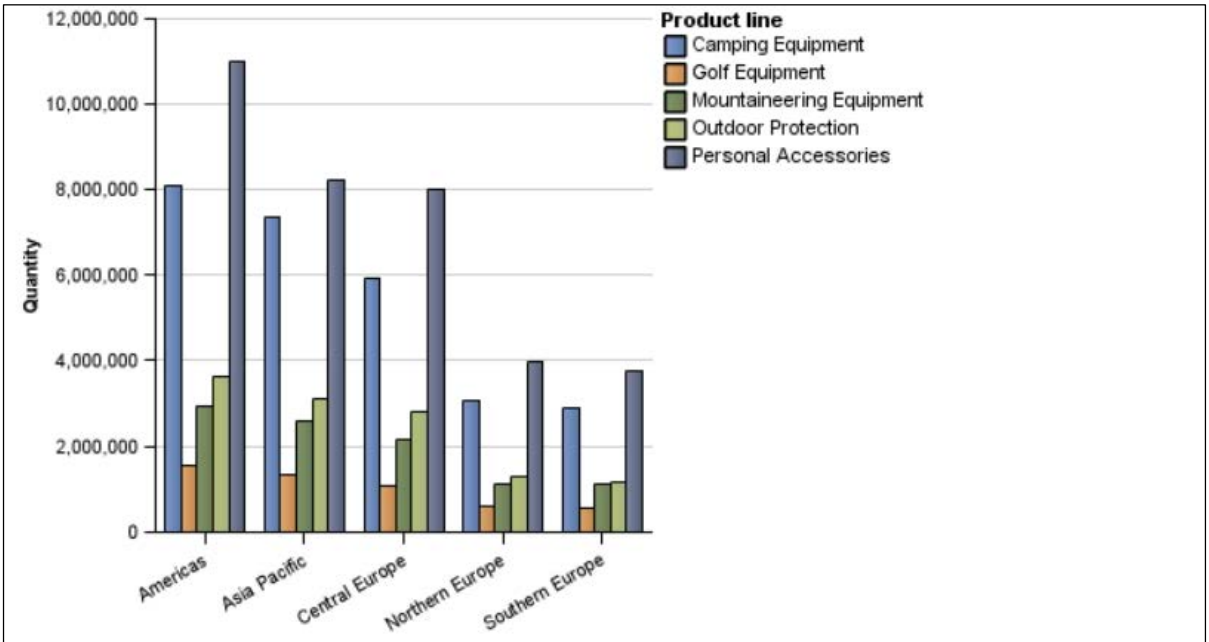
  The results appear as follows:



Quantity	Camping Equipment	Golf Equipment	Mountaineering Equipment	Outdoor Protection	Personal Accessories
Americas	8,101,682	1,544,411	2,948,533	3,619,457	10,998,468
Asia Pacific	7,366,131	1,338,406	2,571,299	3,114,960	8,198,744
Central Europe	5,904,428	1,071,235	2,146,207	2,800,923	8,016,774
Northern Europe	3,046,563	592,168	1,131,215	1,310,804	3,954,449
Southern Europe	2,882,345	567,481	1,102,837	1,168,301	3,739,270

- **Rendered report** tab: Run the report.
- **Prompt**: Choose **Column**.

  The results appear as follows:



Quantity	Camping Equipment	Golf Equipment	Mountaineering Equipment	Outdoor Protection	Personal Accessories
Americas	8,101,682	1,544,411	2,948,533	3,619,457	10,998,468
Asia Pacific	7,366,131	1,338,406	2,571,299	3,114,960	8,198,744
Central Europe	5,904,428	1,071,235	2,146,207	2,800,923	8,016,774
Northern Europe	3,046,563	592,168	1,131,215	1,310,804	3,954,449
Southern Europe	2,882,345	567,481	1,102,837	1,168,301	3,739,270

- Close the rendered report tab.
- Close the report without saving it..
- Sign out of the **IBM Cognos Analytics** portal.
- Close the web browser.

IBM Training

Explore the Prompt API

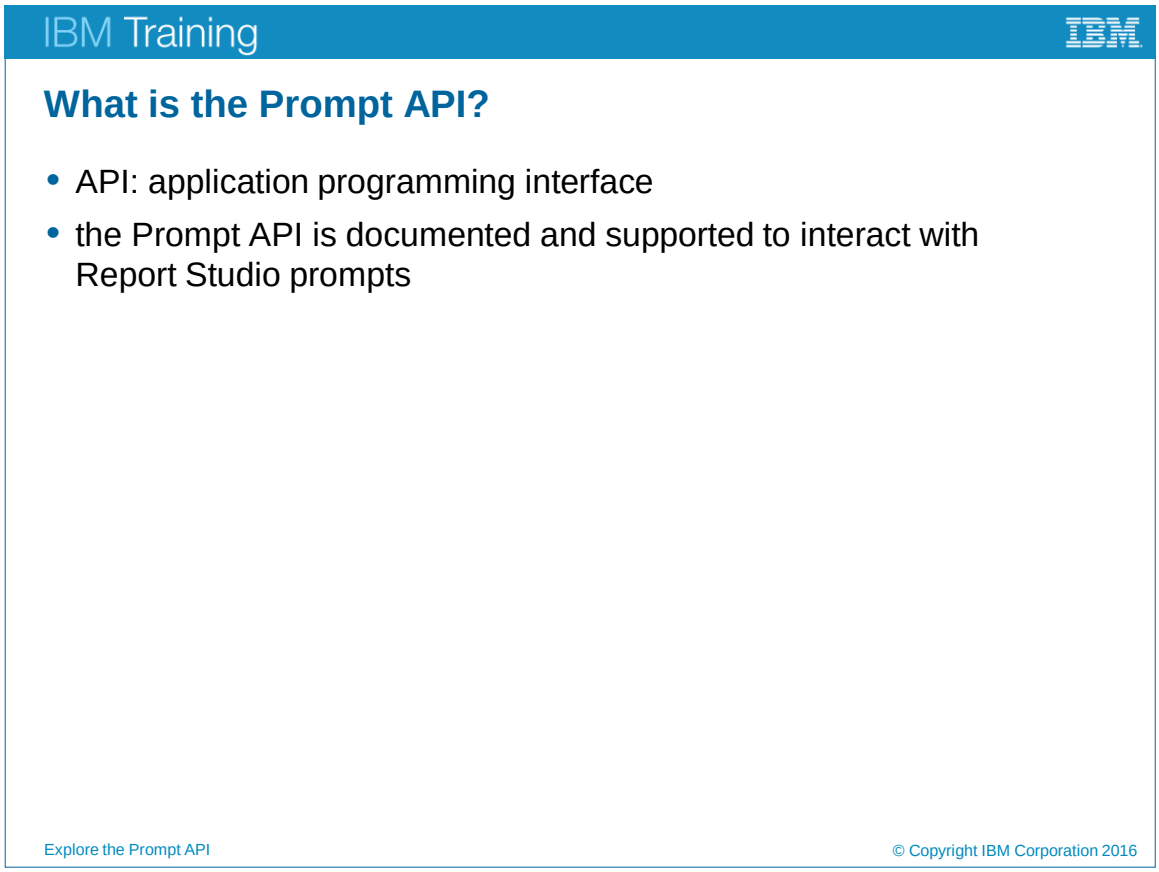
IBM Cognos Analysis (v11.0)

IBM Training

Unit objectives

- Describe the Prompt API
- Describe the uses of the Prompt API
- Describe the primary objects, methods and properties used in the Prompt API
- Review prompt values
- Explore the enhancement of prompts in reports with the Prompt API

Unit objectives

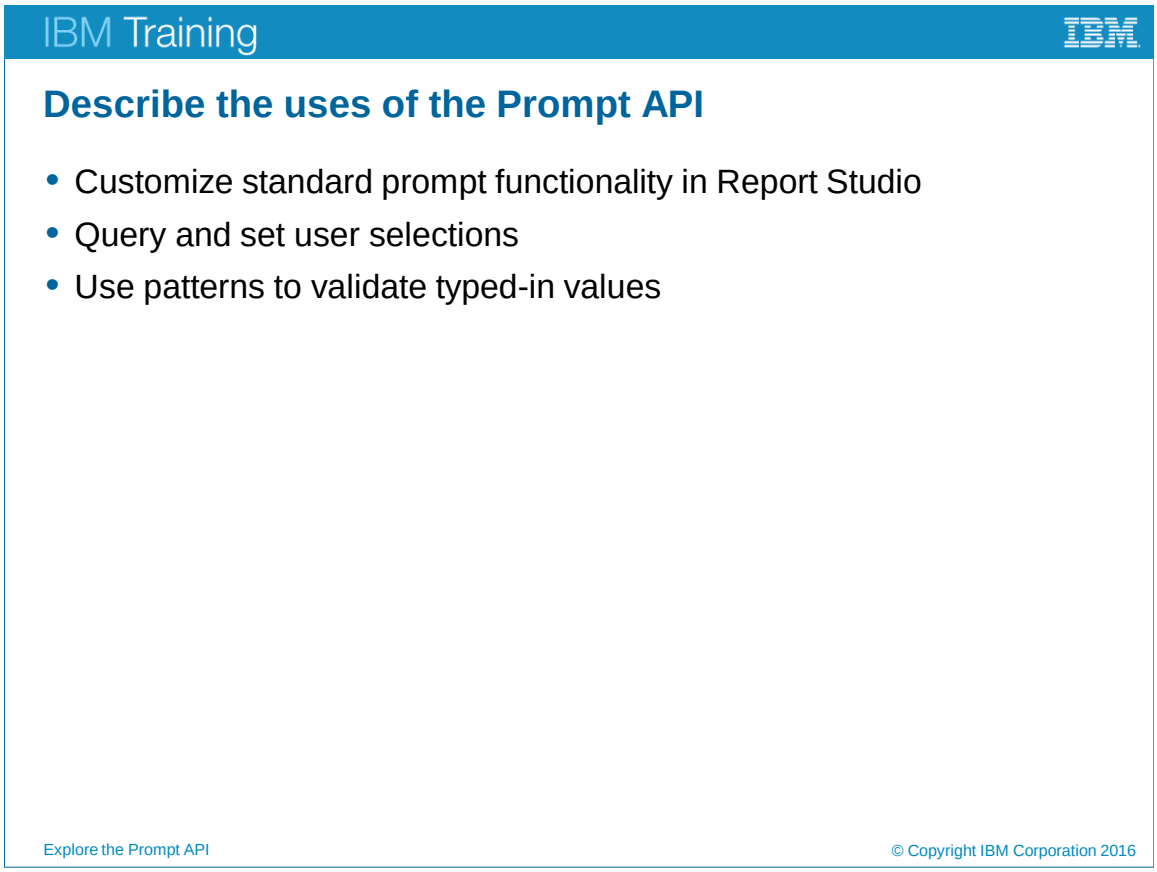


What is the Prompt API?

An API is an interface that allows an application program that is written in a high-level language to use specific data or functions of the operating system or another program.

Using the Prompt API, you can extend or replace prompts in IBM Cognos Analytics - Reporting. This gives you the flexibility to enhance your prompts without requesting formal product enhancements.

The Prompt API is supported in IBM Cognos Analytics - Reporting for authoring and IBM Cognos Viewer for consumption.



Describe the uses of the Prompt API

For the prompts in your report, you can use the Prompt API to set a prompt value, read a prompt value, or delete a prompt value.

You can also validate prompt values that are typed in by a user, to ensure that the value matches the required pattern for the prompt. For example, you may require that the user submit a value with specific alpha-numeric ordering, such as a catalog number format of AA1234.

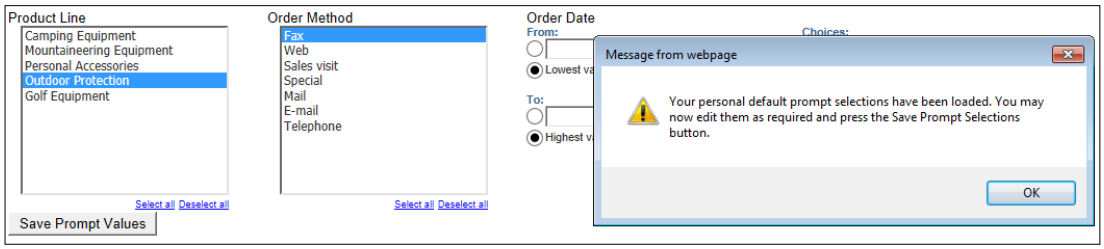
With the Prompt API you can also set default values using expressions or query data. For example, you can ensure if a user selects a prompt value of a specific tent model, that the selection of tent colors selection to the user in another prompt will only include the colors applicable to the selected tent model.

Sample reports exist to demonstrate various usage scenarios of the Prompt API.

IBM Training

Demonstration 1

Run two Prompt API samples to introduce enhanced prompt functionality



Explore the Prompt API

Demonstration 1: Run two Prompt API samples to introduce enhanced prompt functionality

Demonstration 1: Run two Prompt API samples to introduce enhanced prompt functionality

Purpose:
You want to run a couple of Prompt API sample reports to see how prompt functionality can be enhanced with the Prompt API in IBM Cognos Analytics - Reporting.

This demonstration is optional, as it is to merely run a couple of reports and show one example of Prompt API functionality. This demonstration does not go in to detail about how the specific JavaScript code would be used, but instead, is more of a quick peek before diving in to the Prompt API detail.

Portal:	**http://vclassbase:9300/bi**
User/Password:	**brettonf/Education1**
Application:	**IBM Cognos Analytics - Reporting**
Package:	**Team content\Samples_Prompt_API\GO Sales (query)**
Report location:	**Team content\Samples_Prompt_API\GO Sales (query)**
Report:	**Personal Default Prompt Selections - Set Selections**

Task 1. Run the sample report.

1. Sign in to the **IBM Cognos Analytics** portal with **brettonf/Education1** credentials.
2. From the **IBM Cognos Analytics** portal, navigate to **Team content\ Samples_Prompt_API\GO Sales (query)**.

   A list of sample reports that use the Prompt API are available for your reference.
3. Click the **Personal Default Prompt Selections - Set Selections** report to run it.

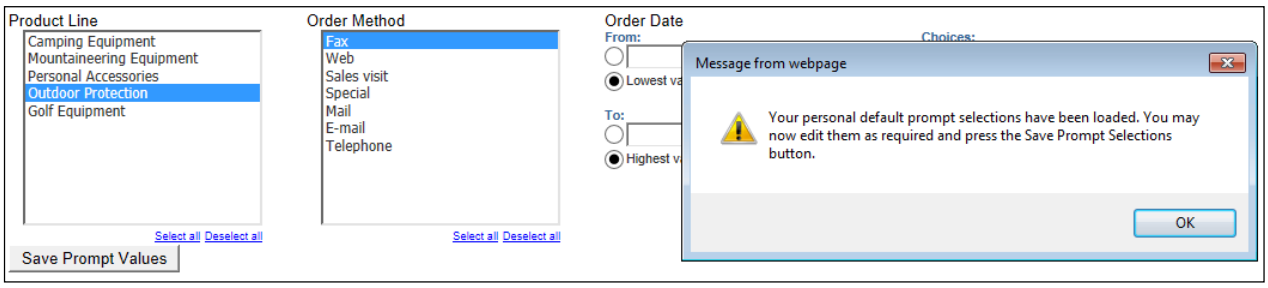
Task 2. Review the contents of the report in IBM Cognos Viewer.

The report is commented with instructions to help you understand how it works. The instructions are to help select prompt values that you (as a report consumer) would use most often, and then the option to save your selections using the Save Prompt Values button. The values will be saved to your browser cookies. You will follow the steps to make some prompt selections, and then you will open another Prompt API sample report which uses this functionality, to ensure that your prompt selections were saved.

1. In the **Product Line** prompt on the left, click **Outdoor Protection**.
2. In the **Order Method** prompt, click **Fax**.
3. Click the **Save Prompt Values** button.

   A message notifies you that your personal default prompt selections have been saved.
4. Click **OK** to close the message, and then on the **Application** bar, remove the report from the currently active list (if you don't see the arrow to drop down, click the report name - the option to Remove will appear).
5. From the **IBM Cognos Analytics** portal, navigate to **Team content\ Samples_Prompt_API\GO Sales (query)**.
6. Click the **Personal Default Prompt Selections - Use Selections** to run a different report.

   A prompt page is displayed with a message that informs you that your personal default prompt selections have been loaded. A section of the prompt page appears as follows:



This report displays the prompts that you selected in the first report. The prompts in this report use the Prompt API enhancement to populate the prompt values from the cookie settings in the browser.

7. Click **OK** to the alert message box.
8. On the **Application** bar, remove the report from the currently active list.

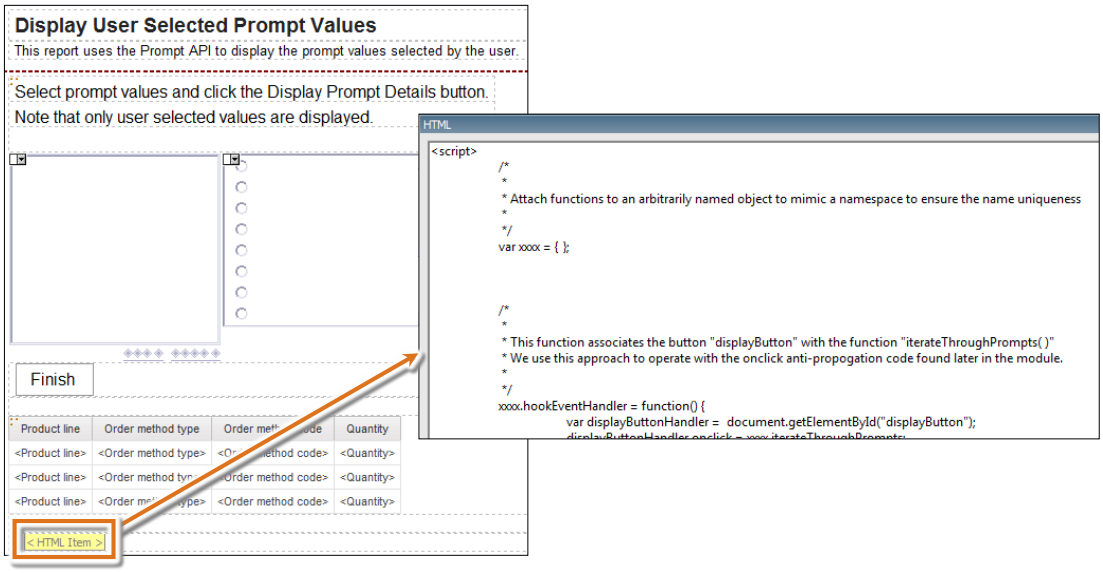
   In this Unit, you will be learning in greater detail how the Prompt API can be used in reports. This demonstration was intended to introduce you to some Prompt API functionality. When you have completed this Unit, you are encouraged to look again at these two reports and the JavaScript code in them, for implementation into your own reports.

Results:
You ran a couple of Prompt API sample reports to see how prompt functionality can be enhanced with the Prompt API in IBM Cognos Analytics - Reporting.

IBM Training

Who uses the Prompt API?

- Report authors



Explore the Prompt API

Who uses the Prompt API?

With the Prompt API, report authors can write JavaScript applications to customize the standard prompt functionality available in IBM Cognos IBM Cognos Analytics - Reporting. To do this, you can use the IBM Cognos Analytics - Reporting HTML object to embed HTML/JavaScript into the report page or prompt page. JavaScript interacts with prompt objects through the Prompt API.

Note: The user developing and running these reports must have the assigned capability to use the 'HTML Items in Report' layout element.

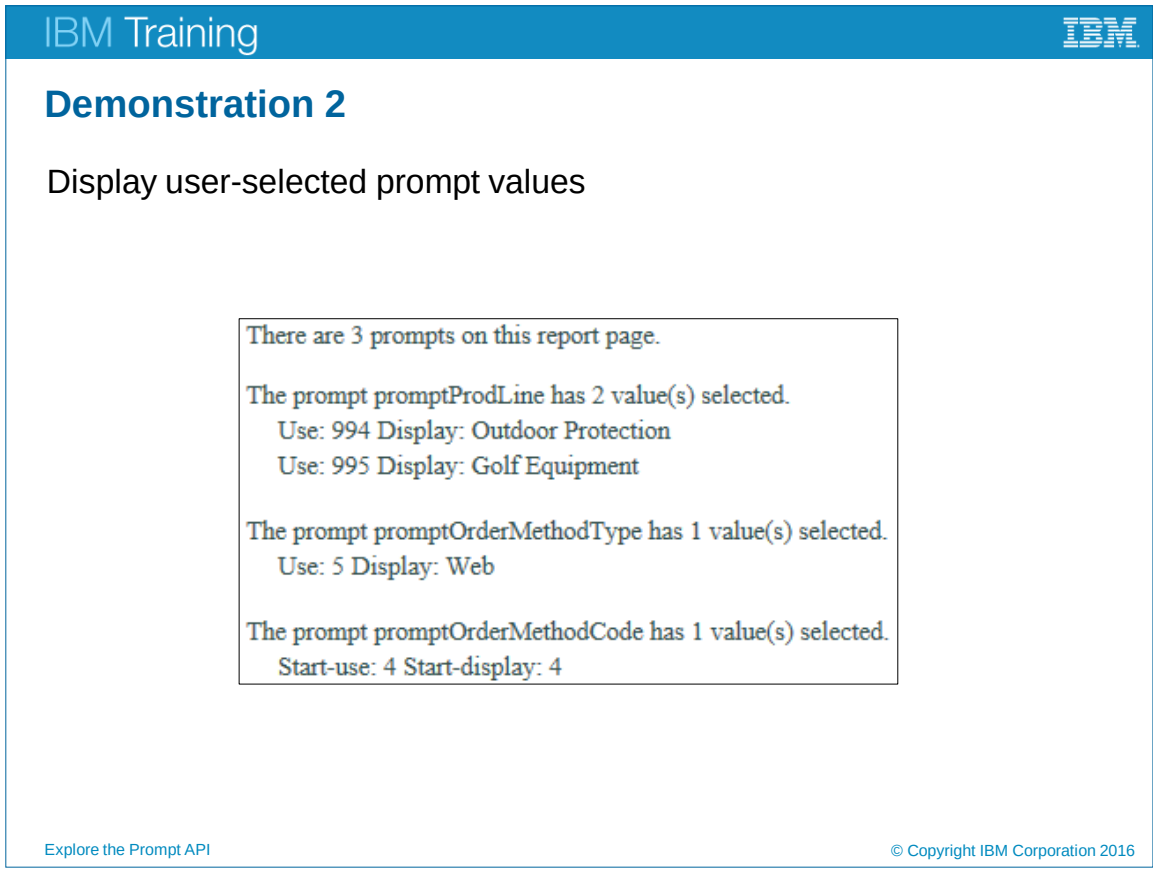
What is JavaScript?

Using JavaScript, in a browser that supports it, can enhance the functionality of Web pages. The reports that you create in IBM Cognos IBM Cognos Analytics - Reporting, when they are run as HTML, are Web pages. Therefore, you can add JavaScript to the HTML coding of your reports using <script></script> tags and JavaScript syntax.

JavaScript runs locally, so user interaction to the scripted application is very fast. Use JavaScript to customize standard prompt functionality.

Be aware that some browsers may not support JavaScript. This module is intended to introduce you to JavaScript in the Prompt API in IBM Cognos Analytics - Reporting. It is not intended that this module will teach JavaScript.

Many people confuse Java with JavaScript. Although there are similarities, there are significant differences. One difference is that Java is compiled by bytecode ahead of being executed, whereas JavaScript is executed in real-time.



Demonstration 2: Display user-selected prompt values

Demonstration 2: Display user-selected prompt values

Purpose:
You want to explore a sample report that demonstrates how to display user-selected prompt values. You will do this to learn about the Prompt API and how you can use JavaScript in IBM Cognos Analytics - Reporting.

Portal:	**http://vclassbase:9300/bi**
User/Password:	**brettonf/Education1**
Application:	**IBM Cognos Analytics - Reporting**
Package:	**Team content\Samples_Prompt_API\GO Sales (query)**
Report location:	**Team content\Samples_Prompt_API\GO Sales (query)**
Report:	**Display User Selected Prompt Values**

Task 1. Run the sample report.

1. In the **IBM Cognos Analytics** portal, navigate to **Team content\ Samples_Prompt_API\GO Sales (query)**.

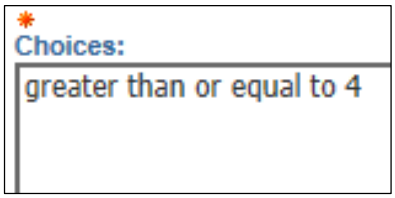
   A list of sample reports that use the Prompt API are available for your reference.
2. Click **Display User Selected Prompt Values** to run the report.

Task 2. Review the contents of the report.

The report is commented with instructions to help you understand how it works. The instructions are to select prompt values and then click the Display Prompt Details button.

1. In the **Product line** prompt, on the left, click **Outdoor Protection**, and then Ctrl+click **Golf Equipment**.
2. In the **Order method type** prompt, click **Web**.
3. In the **From** prompt, click inside the text box, and then type **4**.
4. Click the **Insert** button.
5. In the **Choices** prompt box, click **between 1 and 3**, and then click the **Remove** button.

   The Choices box appears as follows:



6. Click the **Finish** button.

   Below the Finish button, the list is populated with data that satisfies the user prompt selections.

Product line	Order method type	Order method code	Quantity
Golf Equipment	Web	5	3,693,439
Outdoor Protection	Web	5	6,848,660

7. Click the **Display Prompt Details** button.

   A summary of the prompt values that were selected by the user are displayed. The product line prompt had two values selected (Outdoor Protection, Golf Equipment), the order method type prompt had one value (Web) selected, and the order method code prompt had one value selected (greater than or equal to 4). All prompts had use and display properties defined. The result is as follows:

```
There are 3 prompts on this report page.
The prompt promptProdLine has 2 value(s) selected.
    Use: 994 Display: Outdoor Protection
    Use: 995 Display: Golf Equipment
The prompt promptOrderMethodType has 1 value(s) selected.
    Use: 5 Display: Web
The prompt promptOrderMethodCode has 1 value(s) selected.
    Start-use: 4 Start-display: 4
```

   You will look at the report layout to explore the code that was used.

Task 3. Edit the report in IBM Cognos Analytics - Reporting.

1. On the **Application** bar, click **Edit**, and then open the **Properties** pane.
2. Click the first value prompt, and then in the **Properties** pane, review the properties of the prompt, taking notice of properties such as:
   Use value, **Display value**, **Multi-select**, and **Auto-submit**.
3. Review the properties for each of the prompts in the report layout.

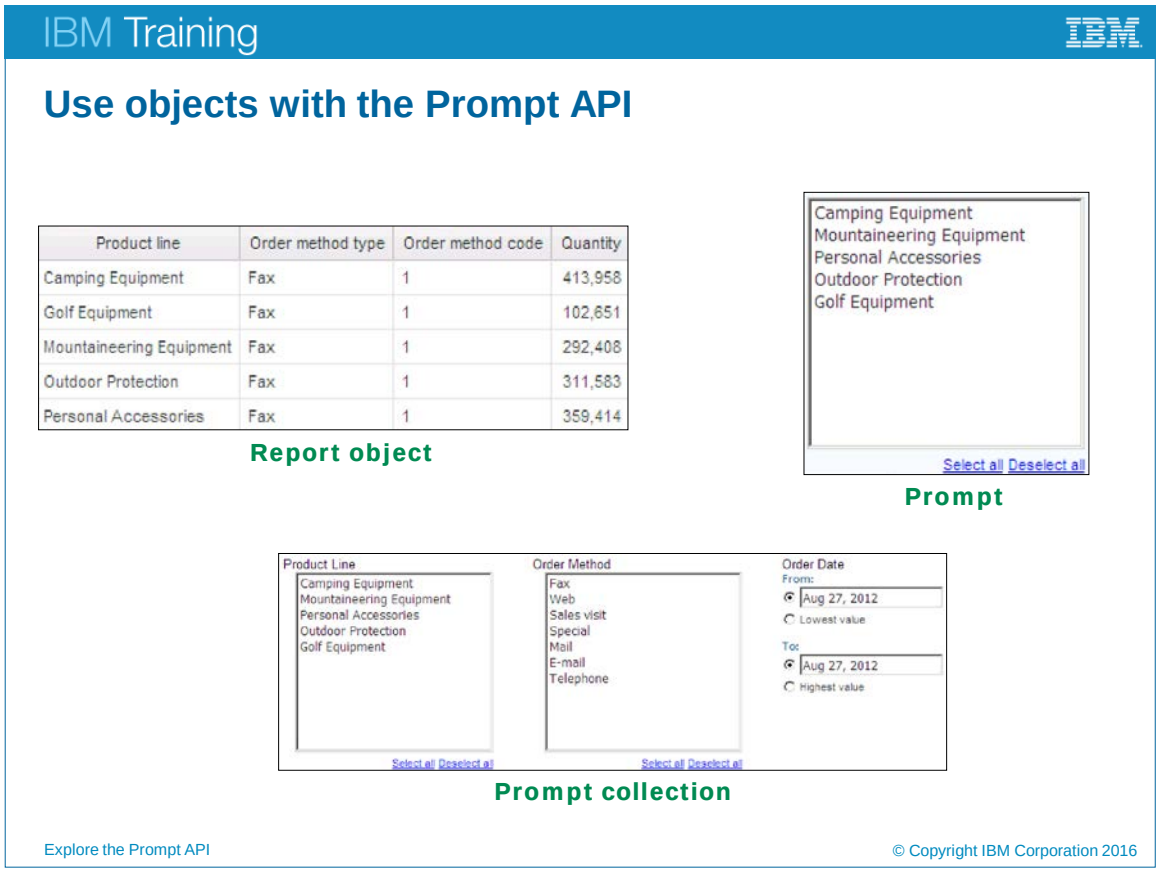
4. In the report layout, double-click the **< HTML item >** object.

   The code reveals JavaScript coding between the <script> and </script> tags. The following actions are coded:

   - var xxxx = { }; is to attach the functions to an object to mimic a namespace and ensure the name uniqueness. JavaScript does not support namespaces, so this is a mechanism to mimic one. There are other techniques that you could potentially use; this one is a simple technique. As you develop your own JavaScript code, you could try creating an object with an unlikely name and attach functions to it; it is the object name that provides the uniqueness.
   - xxxx.hookEventHandler = function() associates the displayButton button with the iterateThroughPrompts() function.
   - iterateThroughPrompts() function loops through the prompt objects, displaying the selected values; notice that the code will handle both range and non-range values.
   - xxxx.cancelBubble = function(eventObject) helps to avoid odd behavior in some browsers with the onClick event.

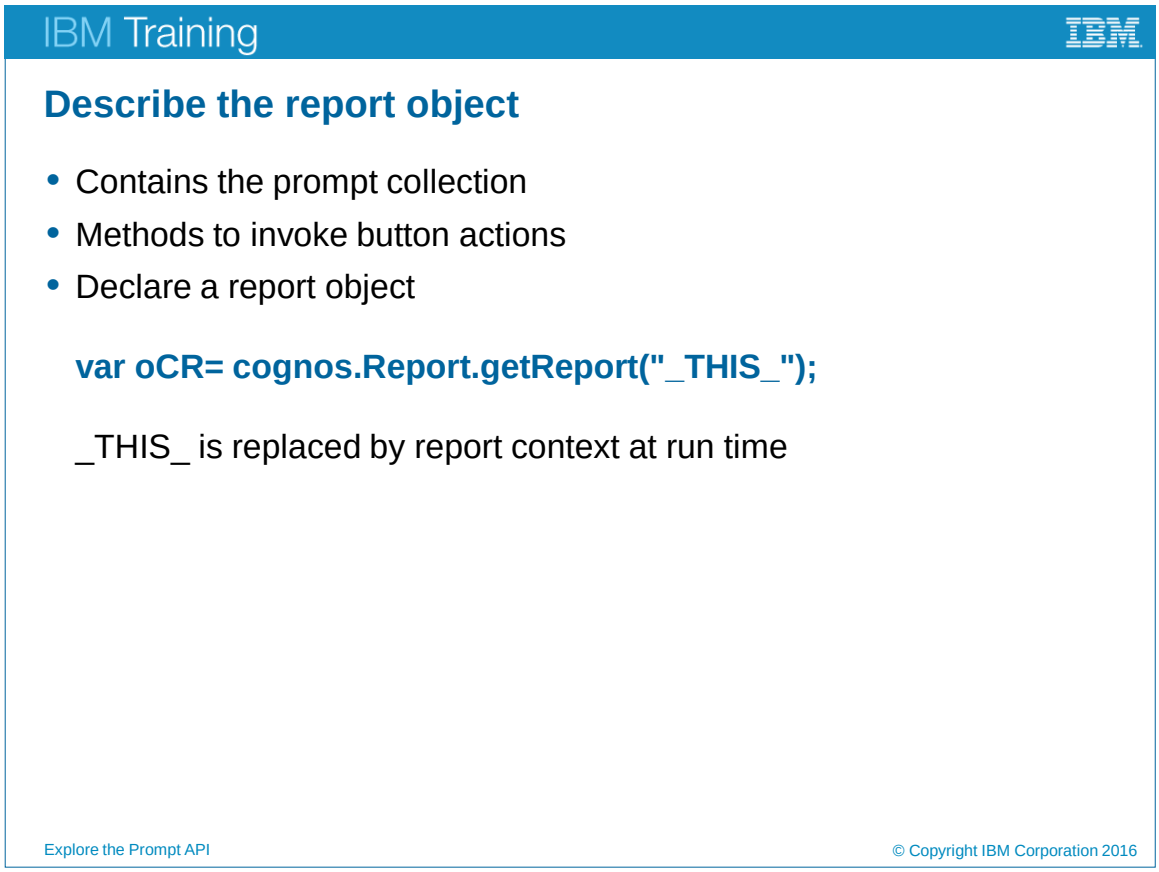
   You can use snippets of this code to enhance your reports.
5. In the **HTML** dialog box, click **Cancel**.
6. Remove the report from the currently active list.
7. Leave the **IBM Cognos Analytics** portal open for the next demonstration.

Results:
You explored a sample report that displayed user-selected prompt values, to learn about how you can use JavaScript in IBM Cognos Analytics - Reporting.



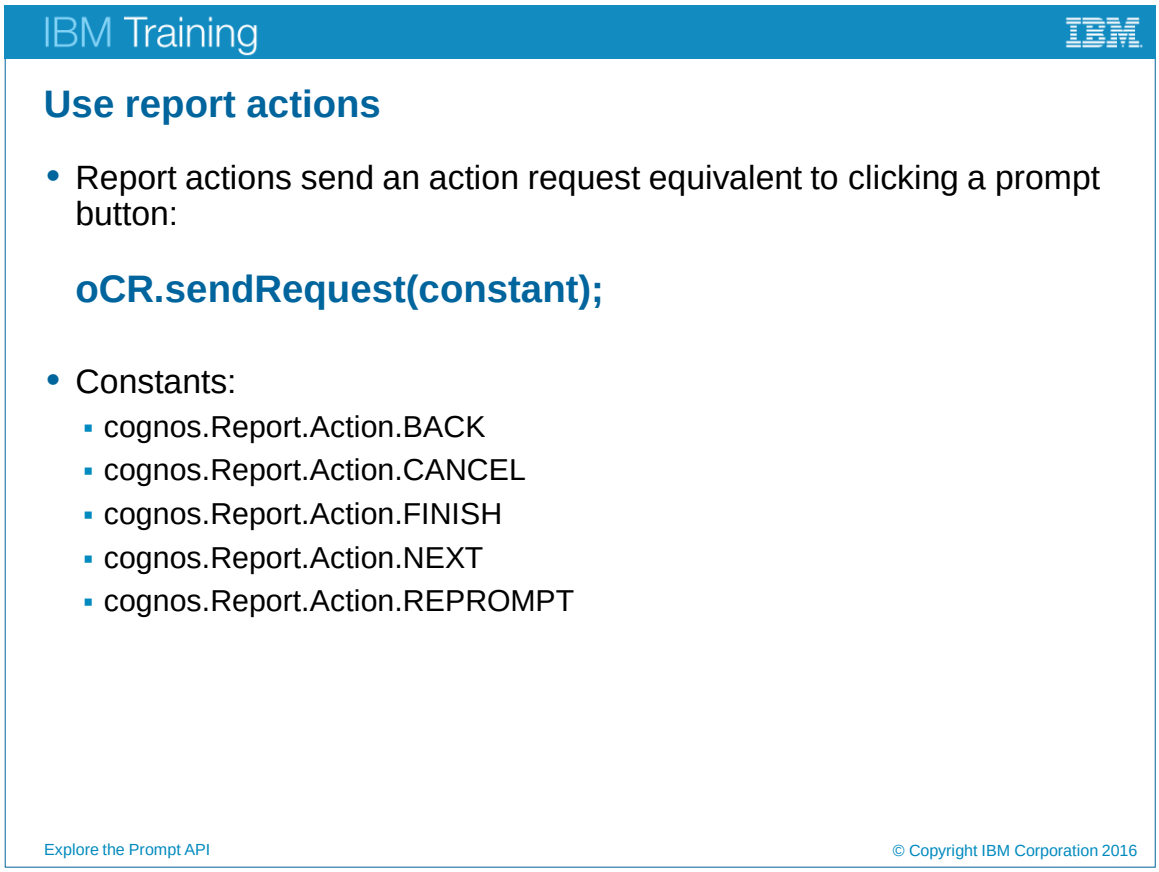
Use objects with the Prompt API

Objects that you can use with the Prompt API include report object, prompt collection and prompt.



Describe the report object

The report object contains the prompt collection and the methods for button actions. The slide provides an example of the code to declare a report object.



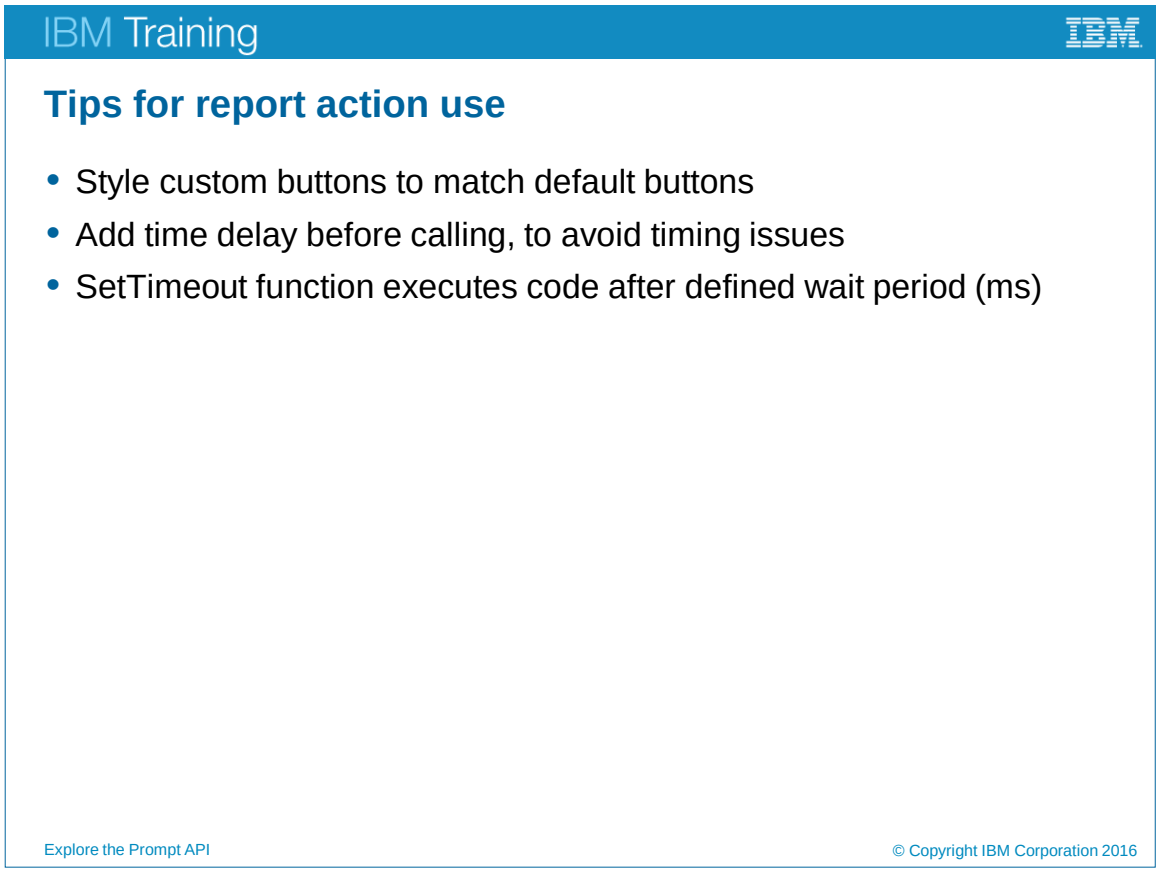
Use report actions

An example of an action request would be:

```
oCR.sendRequest(cognos.Report.Action.FINISH);
```

Use a report action when replacing a standard prompt button with a custom button. For example, replace a standard button that allows validation when the Finish button is clicked, with a custom Finish button that will validate prompt values and then call the Finish action. This example of code could include the following:

```
<button type="button" onClick="validateSelection() ">Finish</button>
validateSelection() {
   var oCR = cognos.Report.getReport('_THIS_');
   …
   oCR.sendRequest(cognos.Report.Action.FINISH);
}
```

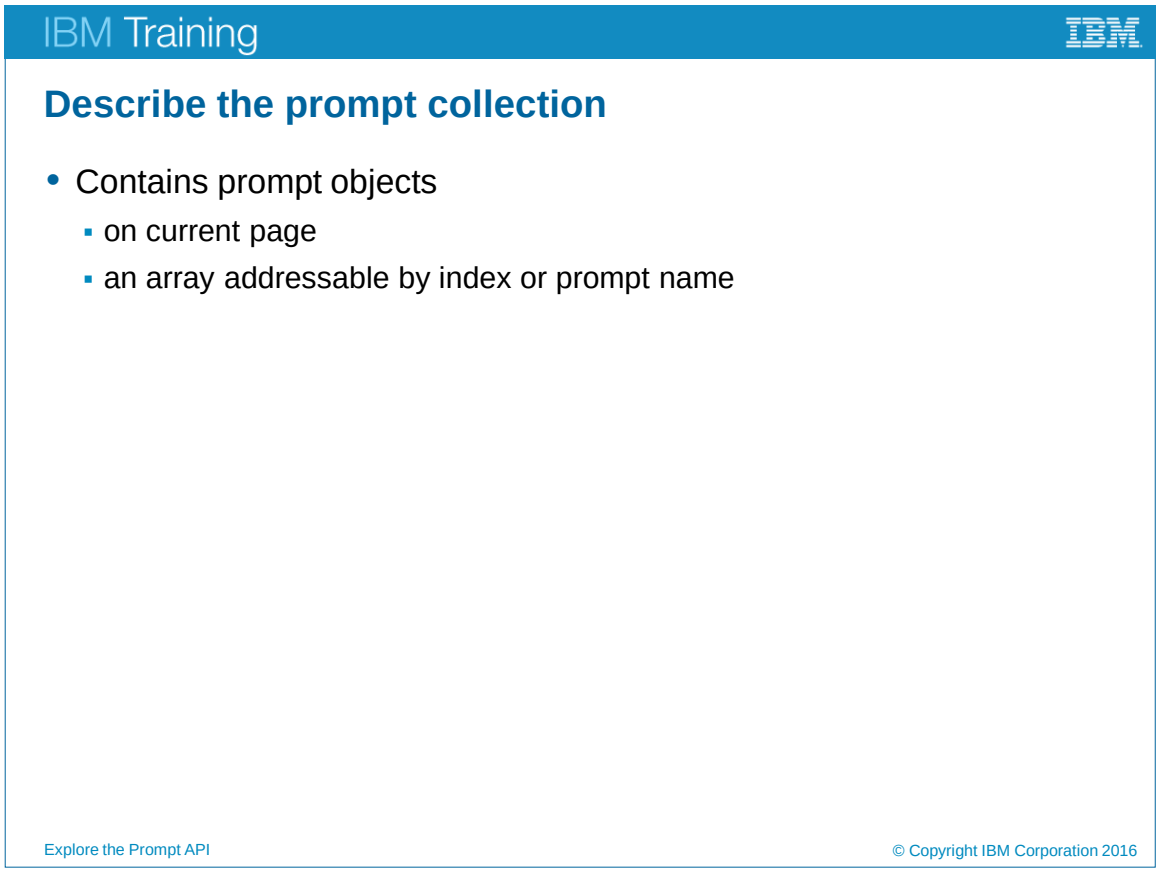


Tips for report action use

For a consistent appearance in your reports, it is recommended that you style your custom buttons to match default buttons elsewhere in the report. Refer to the .buttonStyle properties of default buttons to maintain padding, color, border, and other properties consistently in your reports.

You may consider adding a time delay, to avoid timing issues. An example of the setTimeout function:

```
validateSelection() {
        …
        setTimeout(sendFinishAction, 100);
}
   sendFinishAction = function() {
        var oCR = cognos.Report.getReport('_THIS_');
        oCR.sendRequest(cognos.Report.Action.FINISH);
   }
```



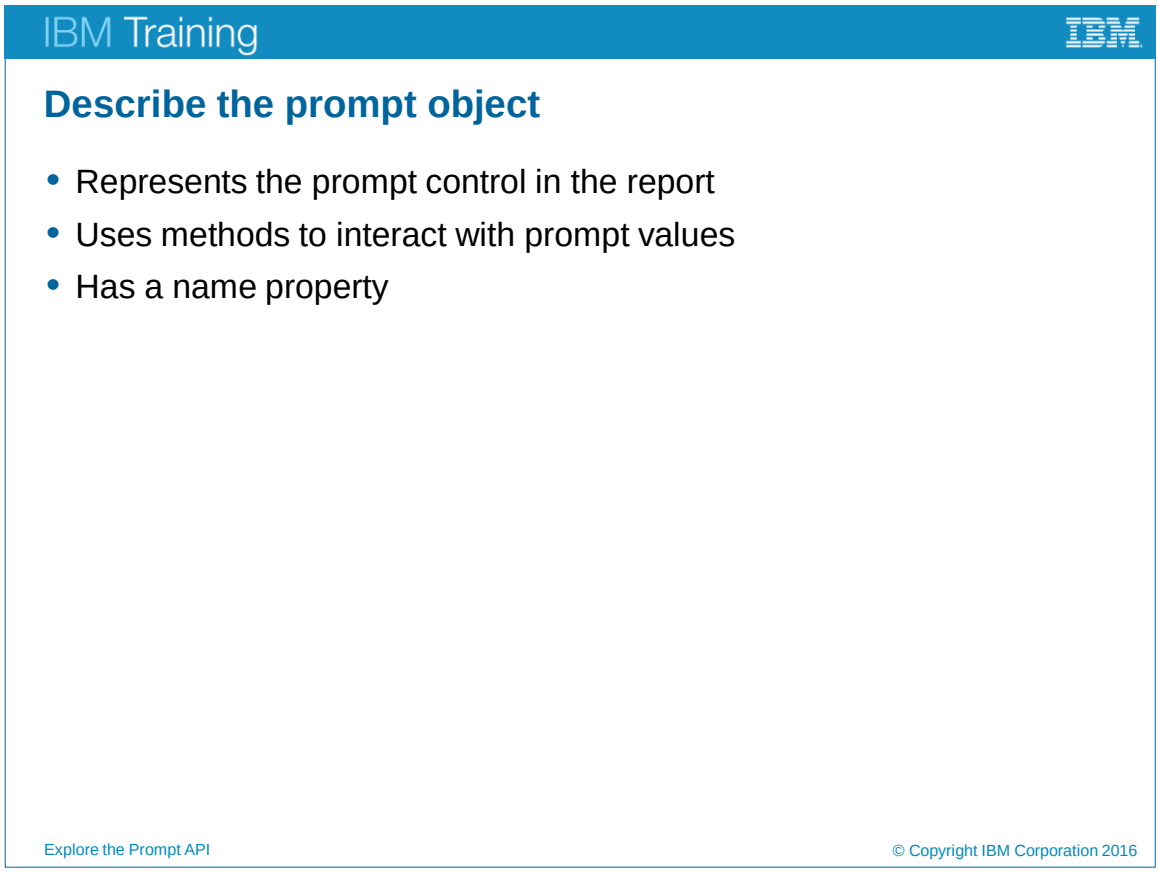
Describe the prompt collection

To get a specific prompt object by name:

```
var myPrompt = oCR.prompt.getControlByName("promptName");
```

To get an array of prompt controls:

```
var myPromptArray oCR.prompt.getControls( );

for (var i = 0; i < myPromptArray.length; i++) {
… myPromptArray[i]…
}
```



Describe the prompt object

Prompt object methods include a method to delete or deselect all prompt values with the .clearValues() function.

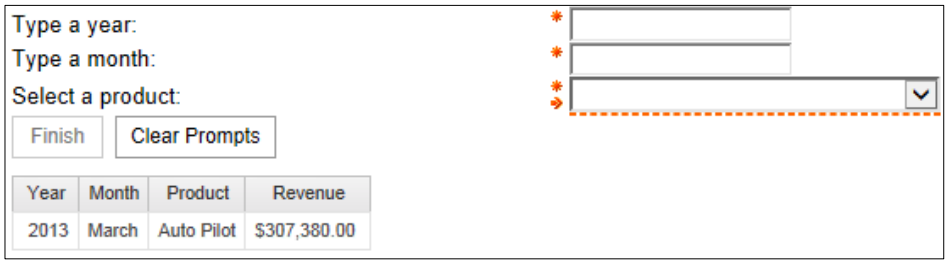
An example of this method is as follows:

```
aPromptControls[i].clearValues();
```

IBM Training

Demonstration 3

- Clear prompt selections



Explore the Prompt API

Demonstration 3: Clear prompt selections

Demonstration 3: Clear prompt selections

Purpose:
You want to enhance a report to include a button that will clear multiple prompt selections. You will use existing JavaScript code from a sample report, and incorporate it into your report. Then you will customize the script to provide a more descriptive label and match formatting with the existing prompt button.

Portal:	**http://vclassbase:9300/bi**
User/Password:	**brettonf/Education1**
Application:	**IBM Cognos Analytics - Reporting**
Package:	**Team content\Samples_Prompt_API\GO Sales (query)**
Report location:	**Team content\Samples_Prompt_API\GO Sales (query)**
Report:	**Clear Prompt Selections**

Task 1. Review the sample report and then run it.

1. In the **IBM Cognos Analytics** portal, navigate to **Team content\Samples_Prompt_API\GO Sales (query)**, and then click **Clear Prompt Selections**.

   The report layout displays two value prompts and a date prompt.

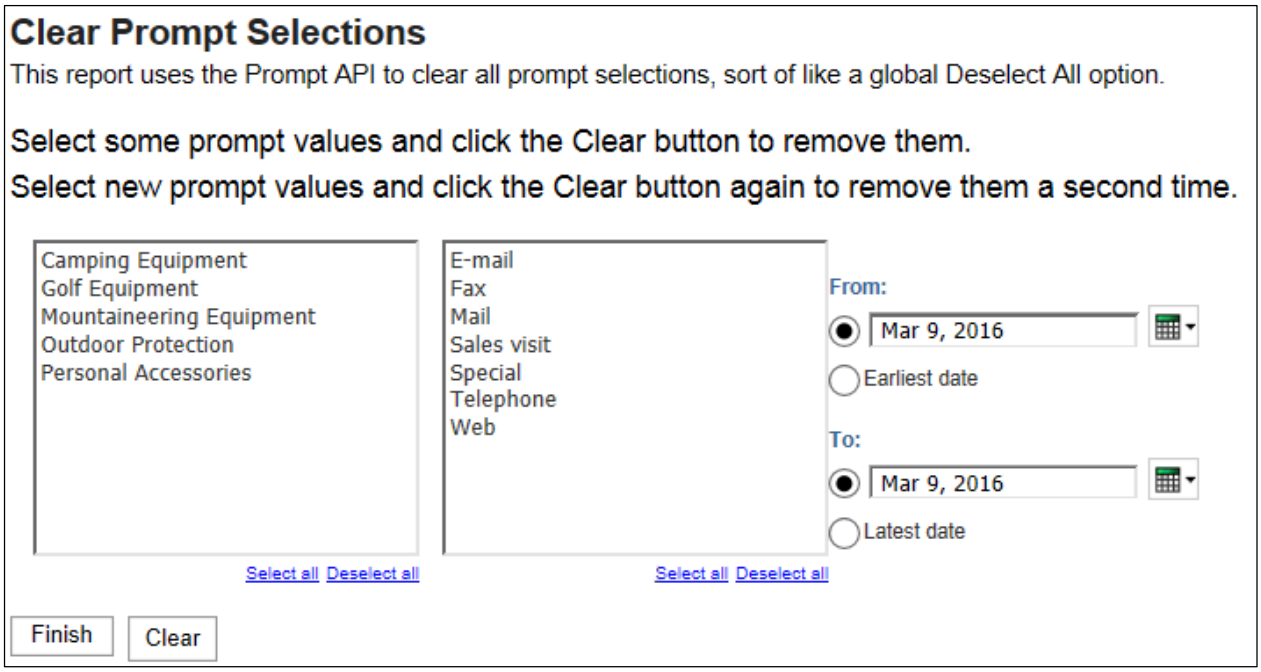
   The description explains the benefit of the enhancement by offering a method to clear all prompt selections with the click of the Clear button.
2. In the **Product line** prompt, click **Camping Equipment**, and then in the **Order method type** prompt, click **Sales visit**.
3. In the **From** prompt select **Jan 1, 2012**, in the **To** prompt select **Aug 31, 2012**, and then click **Finish**.

   The list is populated with data from this range, for the selected product line and order method.

4. Click the **Clear** button.

   All prompts return to their initial state. This is convenient for the user, instead of clearing each prompt separately.

   The results appear similar to the following:



5. On the **Application** bar, click **Edit**.

Task 2. Review the code.

1. In the **Clear Prompt Selections** report, double-click the **< HTML item >** object in the report layout.

   The item is defined as the clearPromptsBtn, which is associated with the clearValues() function. The clearValues() function creates a report object, creates an array of prompts controls, then loops through the prompt controls on the page, clearing selections from each prompt.

```
xfdf.clearValues = function(eventObject) {
    // Create the report object
    var oCR= cognos.Report.getReport("_THIS_");

    // Create an array of prompts controls
    var aPromptControls = oCR.prompt.getControls( );

    // Loop through the prompt controls on this page
    for(var  i=0; i<aPromptControls.length; i++) {
         // Create a prompt control object
         var p =aPromptControls[i];

         // Clear the user selections
         p.clearValues();
    }
```

   You will copy the code to use it in another report, to allow users to clear their prompt selections.

2. Click into the **HTML** dialog box code area, press Ctrl+A to select all of the code, press Ctrl+C to copy it to the clipboard, and then paste it into a **Notepad** document.
3. In the report, click **Cancel** to close the **HTML** dialog box.

   Leave Notepad open, but minimize it, to use in a later task.

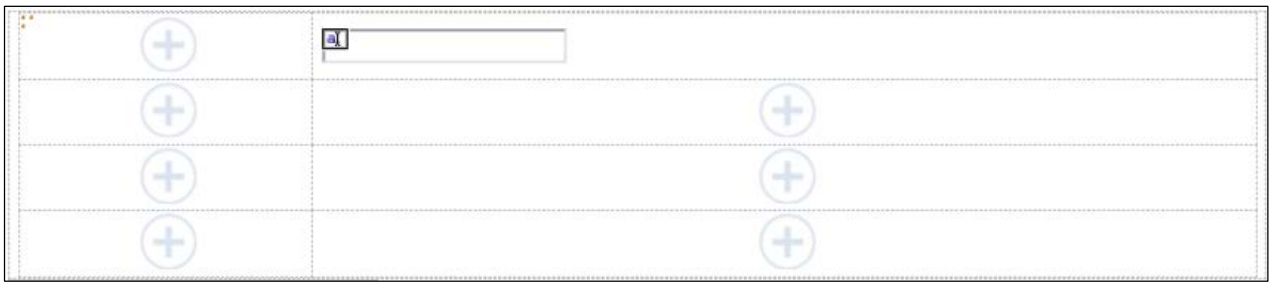
Task 3. Create a report with prompts.

1. On the **Side panel**, open a new **List** template, using the **GO Sales (query**) package from **Team content/Sample_Prompt_API**.
2. Populate the list data container from the **Data/Source** tab with the following items from **GO Sales (query)\Sales (query)**:
   - Time: **Year**, **Month**
   - Products: **Product**
   - Sales: **Revenue**

Year	Month	Product	Revenue
<Year>	<Month>	<Product>	<Revenue>
<Year>	<Month>	<Product>	<Revenue>
<Year>	<Month>	<Product>	<Revenue>

   You will build several prompts, organized in a table, and contained in a block above the list.
3. From the **Toolbox** tab, drag a **Block** object above the list, and then drag a **Table** to the block with **2** columns and **4** rows.
4. From the **Toolbox** tab, expand **PROMPTING**, and then drag a **Text box prompt** to the second column, first cell.
5. In the **Create a new parameter** box, type **p_year**, and then click **Next**.
6. Click the **Package item** ellipsis, expand **Sales (query)\Time**, click **Year**, click **OK**, and then click **Finish**.
7. On the **Application** bar, open the **Properties** pane.
8. With the **Text box prompt** selected, in the **Properties** pane, under the **GENERAL** section, add **2013** to the **Default selections** property.

   The results appear as follows:



9. Repeat steps 4 to 8 to create two more prompt objects in the cell below the previous, using the following information:

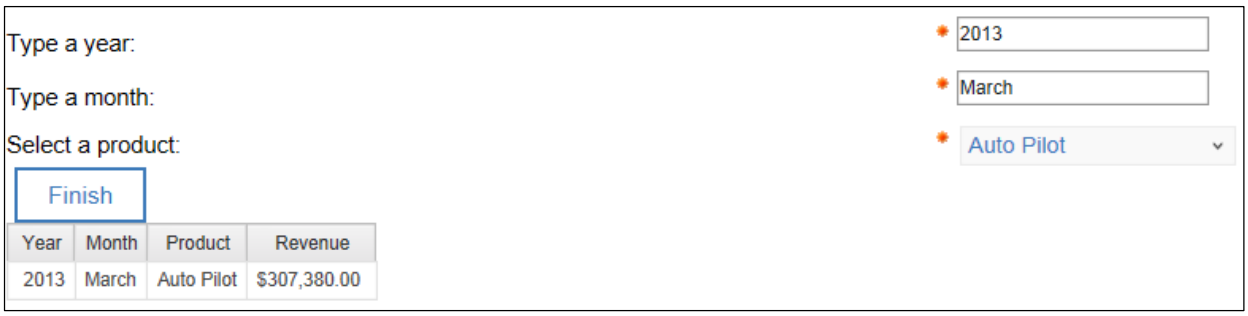
Prompt Type	Text box prompt	Value prompt
Location	Right column, second row	Right column, third row
Parameter name	p_month	p_product
Package item	Time/Month	Products/Product (on the **Create Filte**r page, click **Next**, and then **Finish**)
Default selections	March	Auto Pilot

10. From the **Toolbox** tab, drag a **Prompt button** to the left column, fourth row.
11. Select the prompt button, and then in the **Properties** pane in the **GENERAL** section, change the **Type** property to **Finish**.
12. From the **Toolbox** tab, drag a **Text item** into the first, second, and third rows of the left column of the table, populating the text boxes as follows:

You could apply formatting to size the columns and define fonts and styles, but in the interest of time for this demonstration, this will not be performed.

13. Run the report in **HTML**.

The result appears as follows:



You can see the prompts with the resulting filtered list.

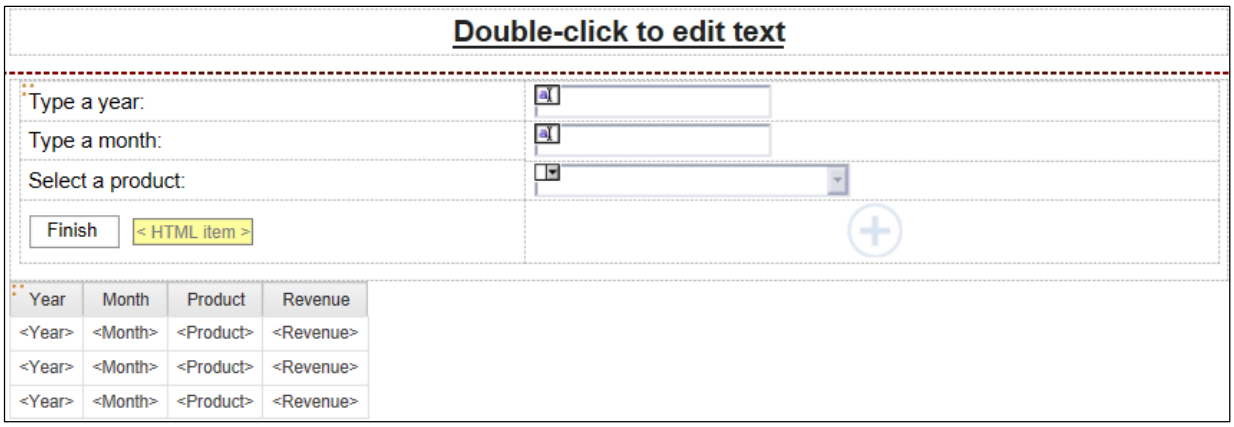
14. Close the rendered report tab, and then save the report into **Team content\B6059** as **Product Revenue (Prompted)**.

Task 4. Use the code from the sample report, in the new report.

You want the ability to clear all prompts on the page. To do this, you will use the JavaScript code from the sample report, and implement it into your report.

1. From the **Toolbox** tab, expand **ADVANCED**, and then drag an **HTML item** object to the right of the **Finish** button object.
2. In the report layout, double-click the **< HTML item >** object.
3. Switch to the **Notepad** session that you left open in Task 2.
4. From the **Edit** menu, click **Select All**, and then click **Copy** to copy the text to the clipboard.
5. Switch to the **IBM Cognos Analytics - Reporting** report, paste the clipboard contents into the **HTML** dialog box pane, and then click **OK** to close the **HTML** dialog box.
6. Click on the block object that contains the table, and then, on the **Properties** pane, under **BOX**, change the bottom padding to **10px**

   A section of the report layout appears as follows:



7. Click the list data container, then on the **Properties** pane, from the **Select Ancestor** button, click **Report**, and then change the **Run with full interactivity** setting to **No**.

   This is very important, the copied HTML code will not function in full interactivity mode.

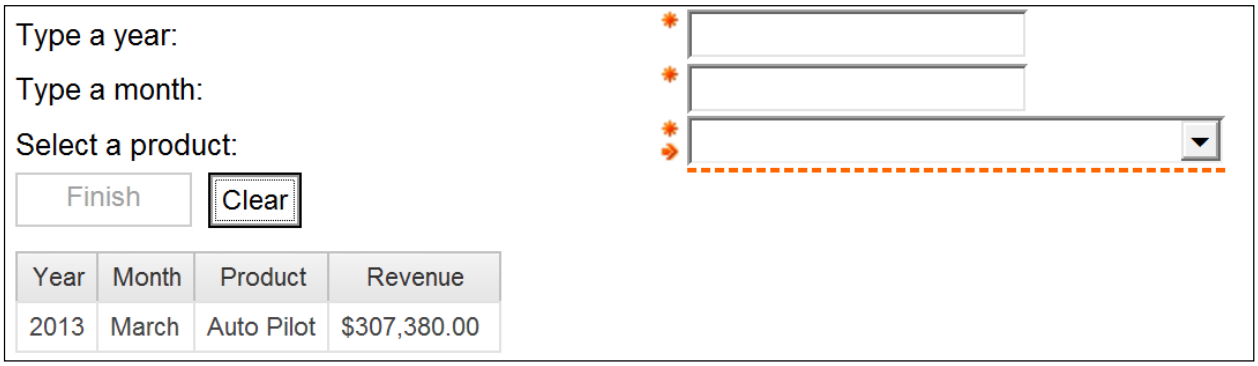
8. Run the report in **HTML**.

   The report opens with the default options displayed in the prompts (2013, March, Auto Pilot).

   Tip: If you were prompted to enter a product before the full report is displayed, then you have not created the p_Product prompt correctly. Return to Task 3 and recreate the prompt as per the table details in step 9.

9. To test your **Clear** button functionality, change the prompt values to **2010**, **May**, and **Bella**, and then click the **Clear** button.

   All prompt values are cleared.



   The list report displays data for the initially selected default values when the report was run, and will only be updated when new values are submitted.

Task 5. Customize the JavaScript code and run the report.

You would like to customize the button that you have added, so that the label is more descriptive, and the button format matches the current Finish button in your report. You will review the properties of the Finish button, and update the JavaScript code to include this. You will also change the button label to Clear Prompts.

1. Close the rendered report tab.
2. On the **Navigate** tab, click **Page1**.
3. Click the **Finish** button, and then review the properties.

   Notice that the Finish button is assigned to the Prompt button class, which defines the formatting of the object when the report is run.
4. On the **Properties** pane, under **POSITIONING**, change the **Size & overflow** property to a height of **25px**.

   The font and text in your custom button appear the same but the border and the height are slightly different.
5. Double-click the **< HTML item >** object.

   Resize it to view the entire first line.
6. In the **HTML** dialog box, modify the top line to reflect the following:

   <button id="clearPromptsBtn" class="icdUI clsPromptButton" style="font-size: 90%; height: **25**px**; border: 1px solid #999999**">Clear **Prompts**</button>

   (The text that is bolded in the above step needs to be added or modified).
7. Click **OK** to close the **HTML** dialog box.

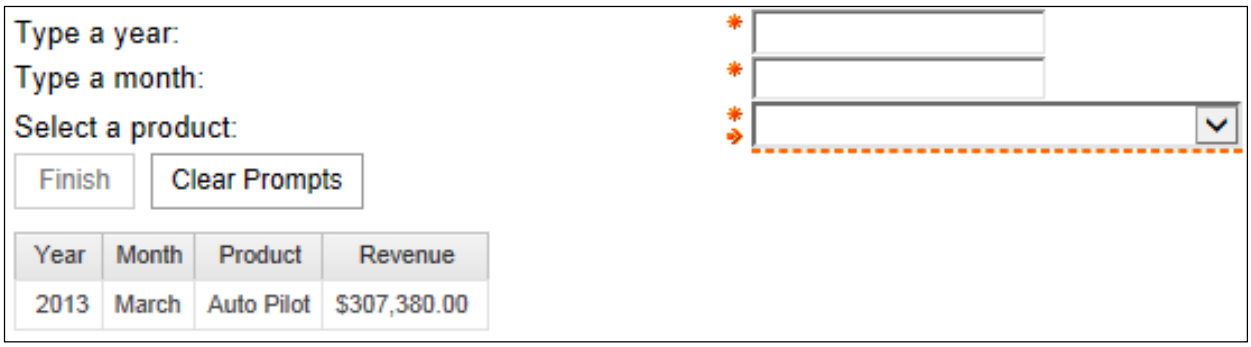
   You have modified the label on your custom button.

8. Run the report in **HTML**.

   The button labels have been customized to have a similar appearance, and the Clear button now has a more descriptive label.

9. Click **Clear Prompts**.

   The prompt values are cleared.

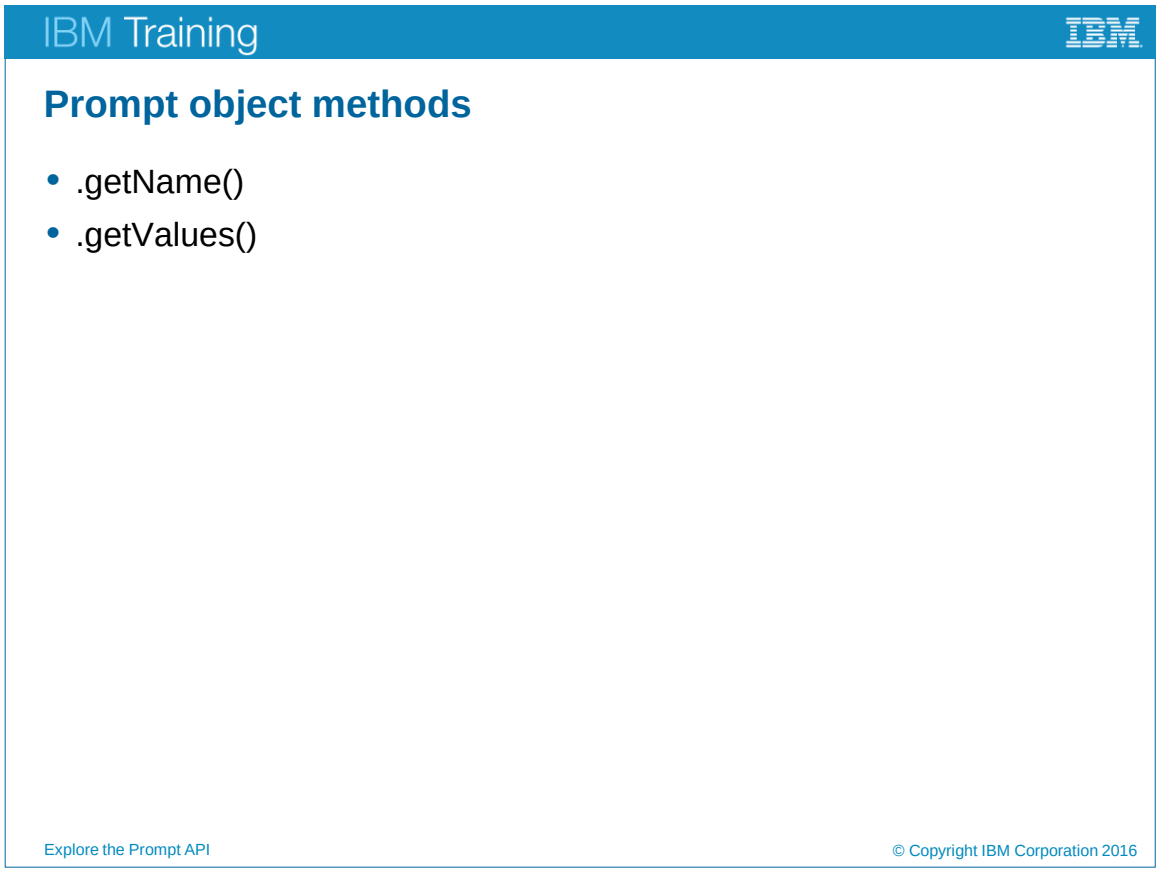


10. Close the rendered report tab.
11. On the toolbar, click **Save**.

    You will use this report in the next demonstration.

12. Close **Notepad** without saving.

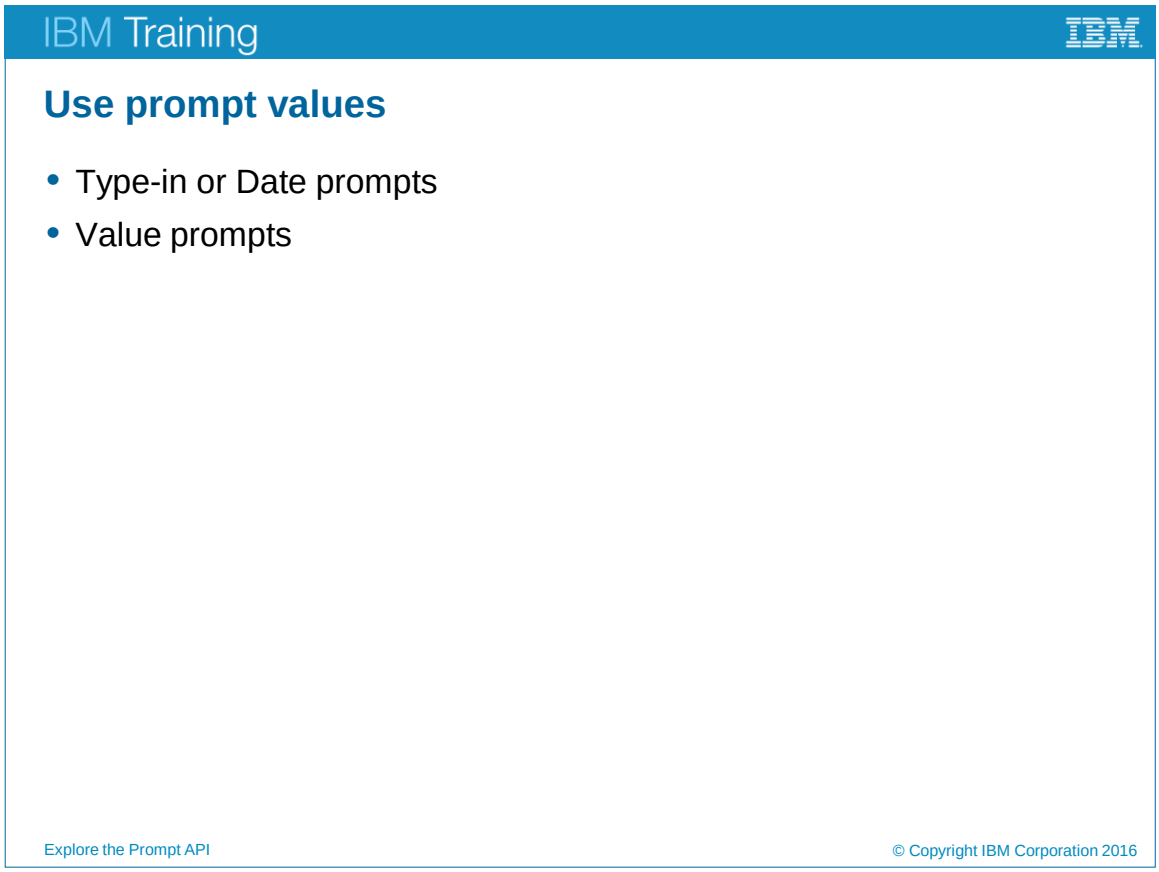
Results:
You enhanced a report using JavaScript code, and customized the script to provide a more descriptive label and match formatting with an existing prompt button.



Prompt object methods

getName() returns the prompt name, and if one was not created by the author, then an arbitrary name will be assigned at run-time.

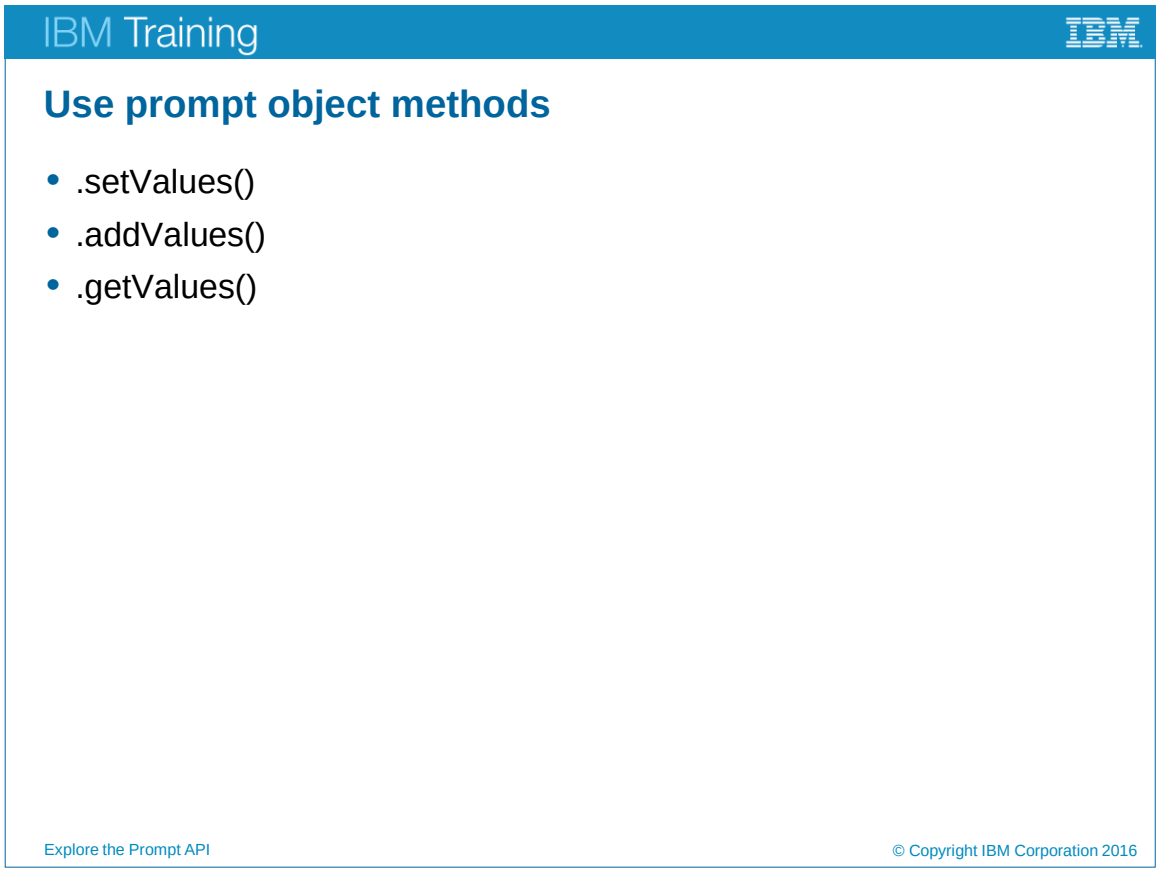
getValues() returns an array of prompt values. (Sample reports…)



Use prompt values

Type-in or date prompts allow the user to enter the value(s), and can be single value or can include multiple values.

Value prompts allow the user to select pre-defined values.

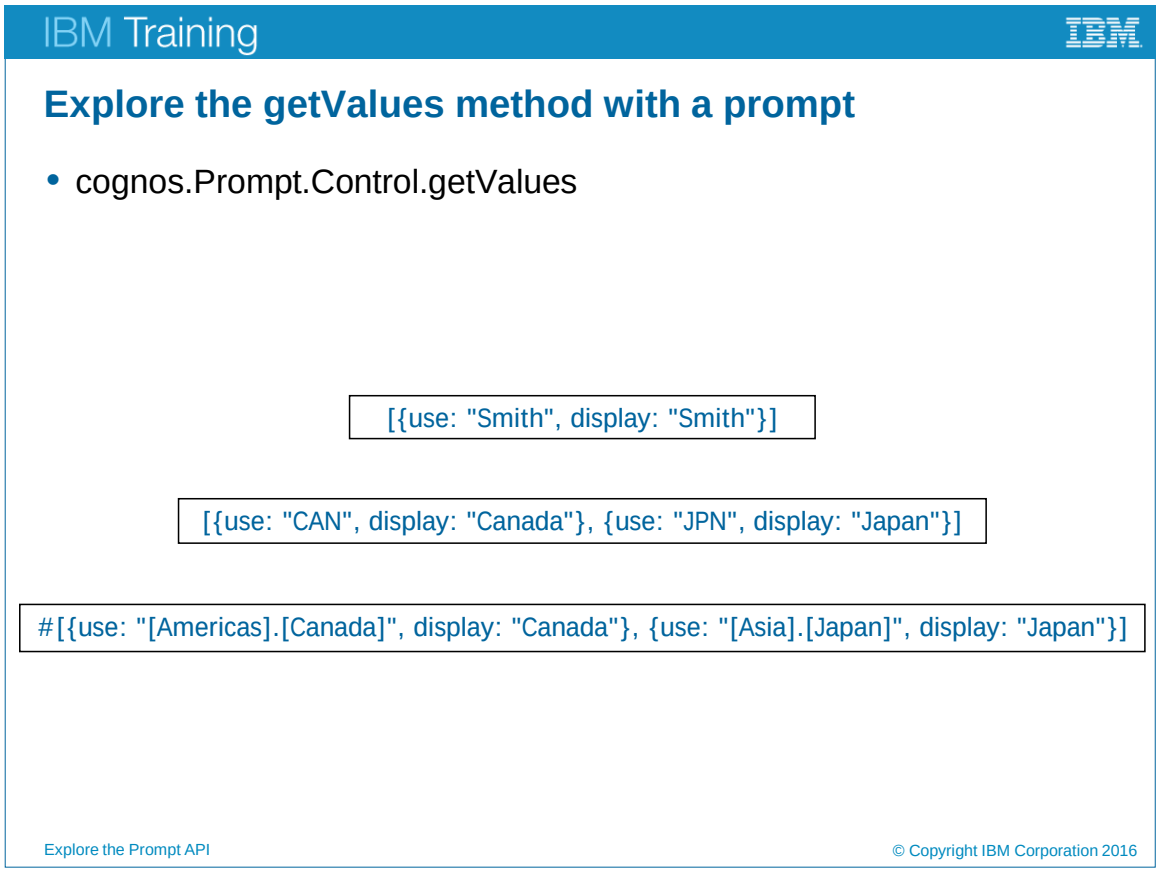


Use prompt object methods

Use .setValues() to add or select prompt values, or delete existing values. This method uses JSON encoded values. JSON is JavaScript Object Notation, a lightweight data-interchange format that is based on the object-literal notation of JavaScript. JSON is programming-language neutral but uses conventions from languages that include C, C++, C#, Java, JavaScript, Perl, Python.

Use .addValues() to add or select prompt values without affecting existing values. This method also uses JSON encoded values.

Use .getValues() to return an array of values. The array may be no value (empty length array), a single value, multiple values, ranges, multiple ranges, or any combination of these.

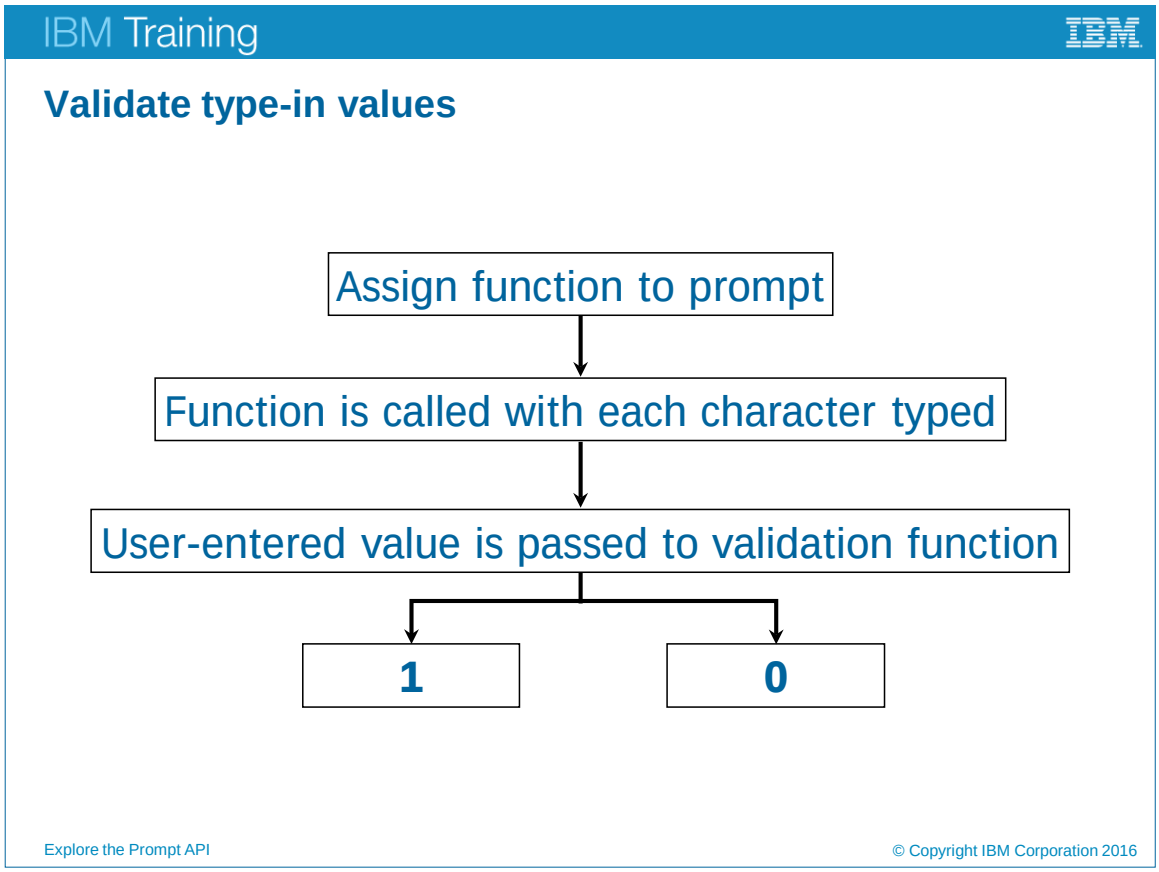


Explore the getValues method with a prompt

Using the cognos.Prompt.Control.getValues method will return an array, even if only one value is selected. For example, using the method on a text box prompt with a single value, an array of one item, with the use value and the display value presented, is returned.

Using the method on a prompt where multiple values can be selected will return an array of all items that the user selected at the prompt, and will include the use value and the display value of all items.

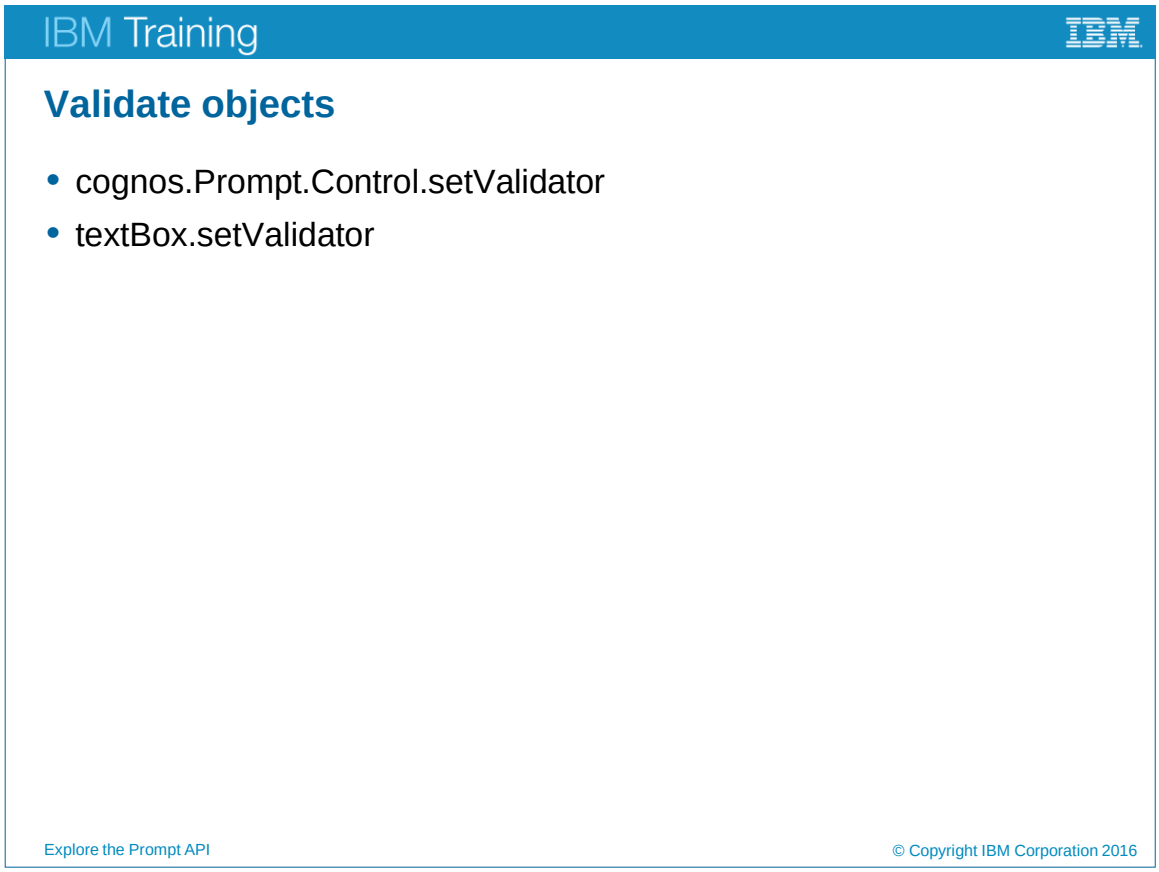
When the method is used on a tree prompt, where multiple items can be selected, the array returns an expanded use value (such as [Americas].[Canada]) and the basic display value of each item.



Validate type-in values

When validating type-in values, you will assign (bind) a function such as promptControl.setValidator(validatorFunction); to the prompt. The function is called with each character that the user types, unless a character repeat such as holding down a key on the keyboard occurs. This user entered values is passed to the validation function and is returned as an array of one or zero individual or range values.

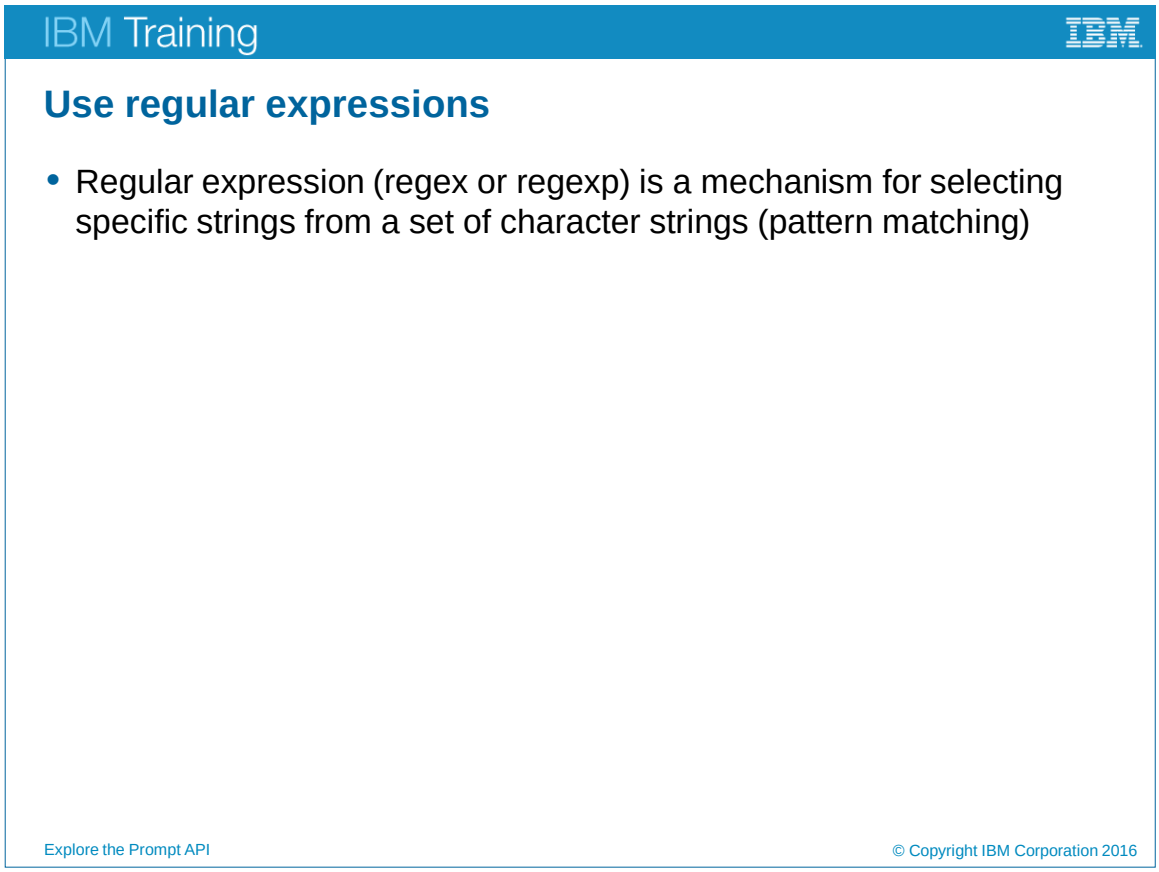
One is returned if there is user input (promptValue.length == 1). Zero value is returned if the user has deleted all the characters in the prompt, or if the value fails the prompt validation. For example, if a non-numeric character is typed in a prompt expecting numbers only, this would return promptValue.length == 0.



Validate objects

Using the cognos.Prompt.Control.setValidator method will allow the default validations function of a control to be defined by the user. When the function returns a value of false, the UI element associated with the control indicates that a validation error occurred. If this method is used in a multi-select control for example, the Insert button is disabled.

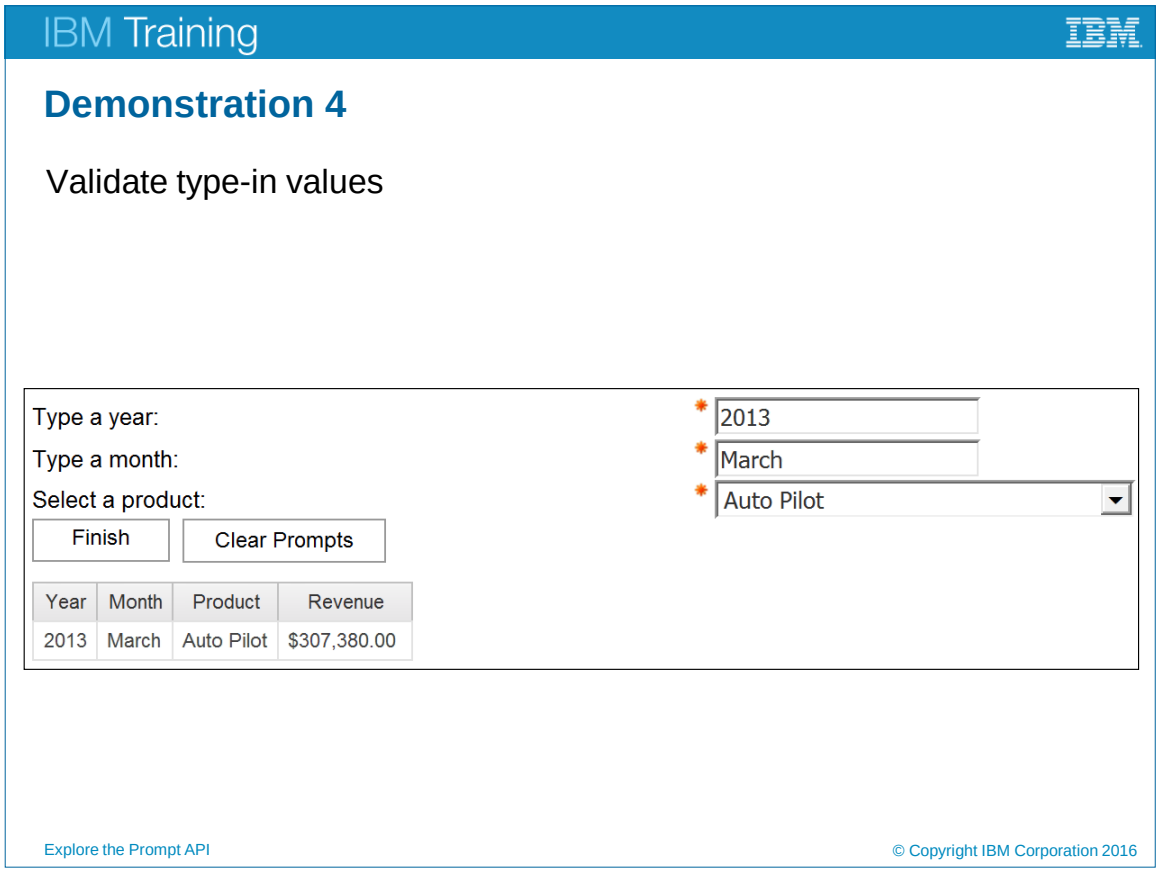
This method can also be used with the text box object, such as when a user must type an entry. The setValidator method can be used to ensure a valid postal code is provided in the form A1A 1A1 for example.



Use regular expressions

More information on using regular expressions is available on the Web at the time of this publication:

- Overview: http://en.wikipedia.org/wiki/Regular_expression
- Tutorials and Syntax: http://www.regular-expressions.info/tutorial.html
- Use of in JavaScript: http://www.javascriptkit.com/javatutors/re.shtml
- Expression Tester: http://www.regular-expressions.info/javascriptexample.html



Demonstration 4: Validate type-in values

Demonstration 4: Validate type-in values

Before beginning this demonstration, ensure that you have performed Demonstration 3: Clear prompt selections, as the report created in that demonstration will be used in this demonstration.

Purpose:
You want to ensure that typed-in prompt values are valid before being passed to the query. To do this you will use JavaScript code to validate the values as numeric, and ensure that the numbers are in an acceptable range.

Portal:	**http://vclassbase:9300/bi**
User/Password:	**brettonf/Education1**
Application:	**IBM Cognos Analytics - Reporting**
Package:	**Team content\Samples_Prompt_API\GO Sales (query)**
Report location:	**Team content\Samples_Prompt_API\GO Sales (query)**
Report:	**Validate Type-In ProductLine Code**

Task 1. Review the sample report and then run it.

1. On the **Application** bar, open the currently active reports list, and then click **Welcome**.
2. On the **Side panel**, navigate to **Team content\Samples_Prompt_API\GO Sales (query)**,
3. Point to **Validate Type-In ProductLine Code** to expose the **More** button, click the **More** button, and then click **Edit report**.
4. Read the descriptive text which explains how the code works.

   As the user types values in the prompt, the values must be within a number range. You can use this code for your year prompt. You will run the report to see how the validation works.
5. Run the report in **HTML**.
6. In the prompt box, type **1**.

   The Finish button is disabled, as the first character of the value must be a 9. All product line codes start with a 9, as indicated in the Product line code column.

7. Delete the **1**, and type **990**.

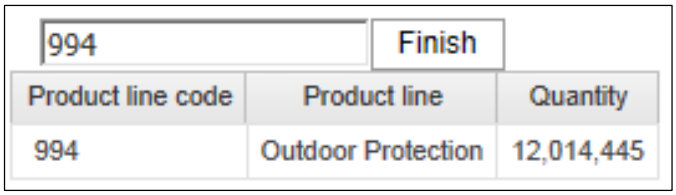
   This invalid entry also disables the Finish button. 990 is not within the range of acceptable product line codes.

8. Delete the **0** and type **4**, to enter a value of **994**.

   The Finish button is enabled.

9. Click **Finish**.

   The list displays data for product line code 994. A section of the result appears as follows:



10. Close the rendered report tab.

Task 2. Review the code.

1. In the **Validate Type-In Product Line Code** report, double-click the **< HTML item >** object in the report layout.

   Review the validation function. Notice the following line:

   ```
   * This function will associate a validation function with the prompt control called promptProductLineCode
   ```

   You will have to customize this code to reflect a valid prompt name in your report.

   Notice the following comment:

   ```
   // Make sure we have a valid Product Line Code of 3 numeric digits
   ```

   Your prompt value criteria will be slightly different.

   Read through the description of the function describing that it is set to numeric values only, and the description below that stating that the filter in the example report will only accept numeric values. You will have to do the same in your report.

   You will copy the code to use it in another report, and modify to suit your business requirements.

2. Click the mouse cursor in the **HTML** dialog box pane, press **Ctrl**+**A** to select all of the code, press **Ctrl**+**C** to copy it to the clipboard, and then paste it into a new **Notepad** document.

   Leave Notepad open, but minimize it, to use in a later task.

3. In **IBM Cognos Analytics - Reporting**, click **Cancel** to close the **HTML** dialog box.

4. On the **Application** bar, open the **Properties** pane.
5. In the report layout, click the text box prompt object to select it, and then in the **Properties** pane, under the **GENERAL** section, notice that the **Numbers only** property is set to **Yes**.
6. Under the **MISCELLANEOUS** section, notice that the name of the object is **promptProductLineCode**.

   This name is referred to in the JavaScript code that you copied.

Task 3. Open a report with prompts.

1. On the **Application** bar, open the currently active reports list, and then click **Product Revenue (Prompted)**.

   This report was saved in Demonstration 3 of this Unit. You must complete that demonstration before you can continue with the remainder of this demonstration.
2. Click the **Year** prompt to select it.
3. In the **Properties** pane, under the **GENERAL** section, change the **Numbers only** property to **Yes**.
4. Under the **MISCELLANEOUS** section, change the **Name** property to **promptYear**.
5. On the **Toolbox** tab, under **ADVANCED**, drag an **HTML item** to the right of the list data container in the report layout, and then double-click this **< HTML item >** object.

Task 4. Customize the JavaScript code.

1. Switch to the **Notepad** session that you left open in Task 2, Step 2.

   You will replace the sample prompt reference to your prompt reference.
2. Change the following lines of code (see all bolded items for the changes):

   - * This function will associate a validation function with the prompt control called promptProductLineCode
     changes to:
     *** This function will associate a validation function with the prompt control called promptYear**
   - asdf.promptControlProdLineCode = oCR.prompt.getControlByName("promptProductLineCode");
     changes to:
     asdf.promptControlyear = oCR.prompt.getControlByName("promptYear");

- // Make sure we have a valid Product Line Code of 3 numeric digits
  changes to:
  // Make sure we have a valid Year of 4 numeric digits
- asdf.promptControlProdLineCode.setValidator(asdf.prodLineCodeValidator);
  change to:
  asdf.promptControlyear.setValidator(asdf.yearValidator);
- * value is 3 digits long - between 100 and 999.
  changes to:
  *** value is 4 digits long - between 2010 and 2013.**
- asdf.prodLineCodeValidator = function(promptValue)
  changes to:
  asdf.yearValidator = function(promptValue)
- // Is value between 991 and 995??
  changes to:
  // Is value between 2010 and 2013??
- if (promptValue[0].use >=991 && promptValue[0].use <=995)
  changes to
  if (promptValue[0].use >=2010 && promptValue[0].use <=2013)

These changes define a range of years acceptable in the prompts.

3. Copy all lines of code to the clipboard, switch to **IBM Cognos Analytics - Reporting**, paste the clipboard contents into the **HTML** dialog box pane, and then click **OK** to close the **HTML** dialog box.

Task 5. Test the code.

1. Run the report in **HTML**.
2. In the year prompt, type **6**.
3. The **Finish** button is immediately disabled, as the validation process has determined that the first digit in a year cannot be anything but 2.
4. In the **year prompt**, delete the **6** and then type **2012**.
5. Upon typing the final digit, the validation process accepts the entry and the **Finish** button is enabled.

6. Ensure that the month is **March**, select the **Bella** product, and then click **Finish**.

   A section of the report appears as follows:

7. Close the rendered report tab.
8. **Save** the report.

Task 6. Review a different validation sample report for comparison. (Optional)

1. On the **Application** bar, open the currently active reports list, and then click **Welcome**.
2. Navigate to **Team content\Samples_Prompt_API\GO Sales (query)**.
3. Point to the **Validate Type-In Postal Code Value** report, and then under **More**, click **Edit report**.
4. Read the descriptive text which explains how the code works.

   As the user types values in the prompt, the values must match a specific alpha-numeric pattern such as M2M 4K8. This is an example of a pattern that is used in Canada for postal codes. You will run the report to see how the validation works.
5. Run the report in **HTML**.
6. In the prompt box, type **1**.

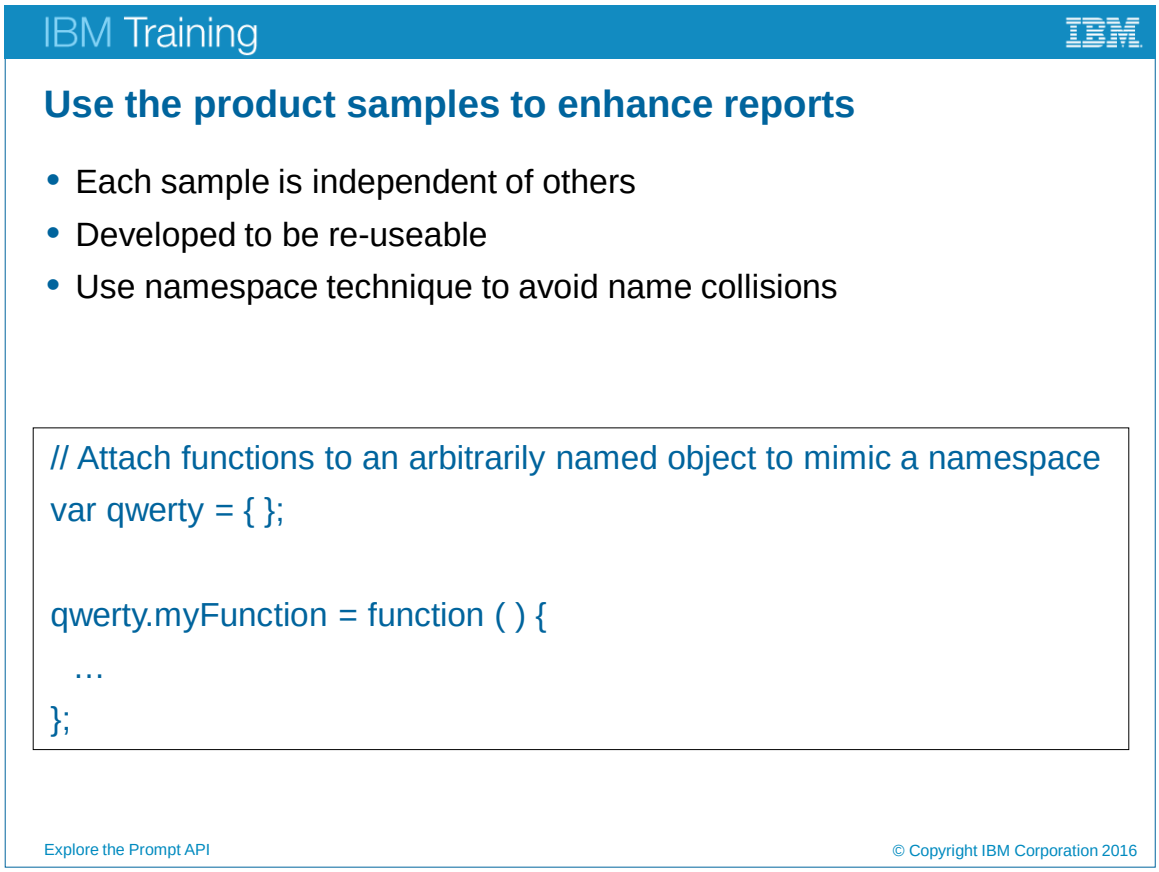
   Descriptive text is displayed at the top of the report to indicate examples of valid postal codes. All postal codes must start with an alphabetic character (non-numeric). The Finish button is disabled, as the first character of the value must not be a number.
7. Typing any combination of letters and numbers that will match the pattern **A#A #A#** will not necessarily result in an enabled Finish button.

   For example, if you type T2P 3Z0, it is a valid code - and the Finish button is enabled. Unless you completely type either of the example postal codes shown in the list below, there will be no data returned, so you cannot click Finish.

   You will review the code to learn how the pattern matching for the text box prompt was achieved.
8. Close the rendered report tab, and then in the report layout, double-click the **< HTML item >** object.

9. Review the code.
10. Click **Cancel** to close the **HTML** dialog box.
11. Remove all **IBM Cognos Analytics - Reporting** reports, without saving.
12. Close **Notepad** without saving.
13. Leave the **IBM Cognos Analytics** portal open for the exercise.

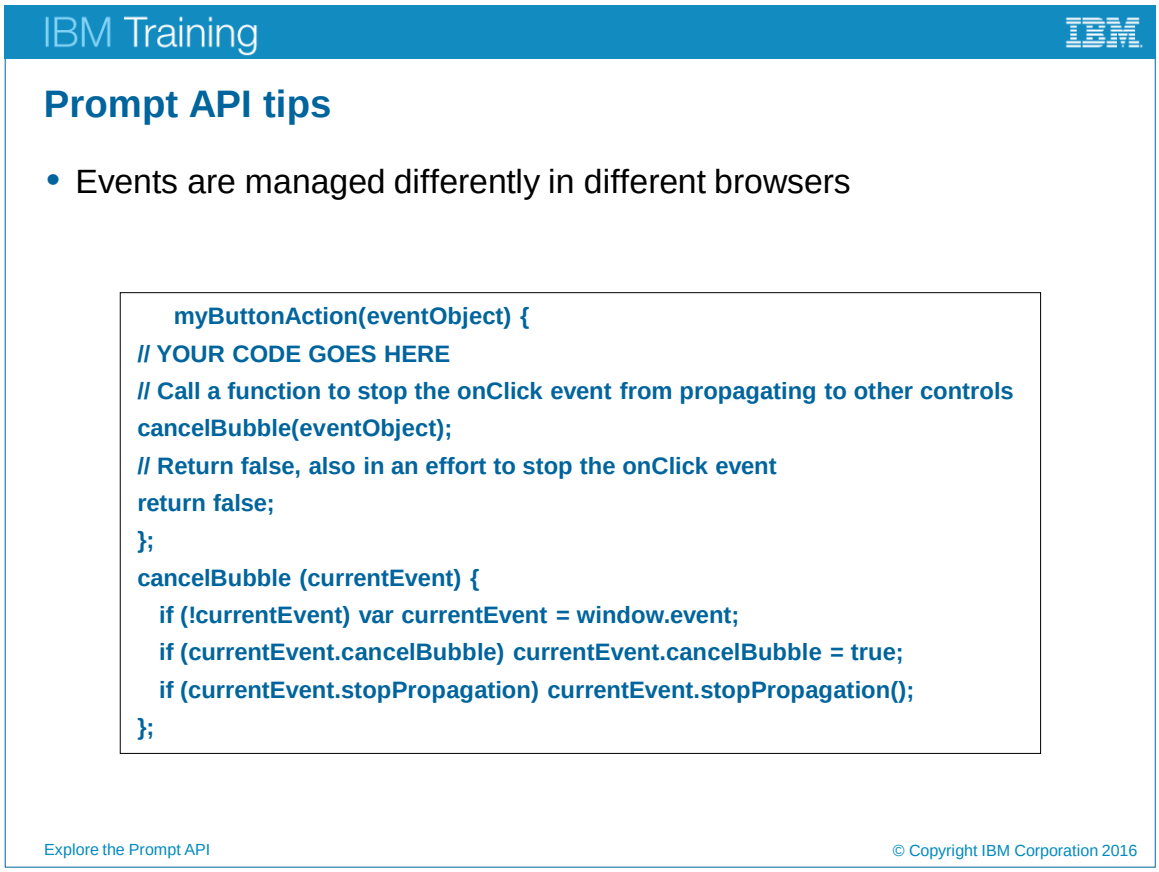
Results:
You used JavaScript to ensure that typed-in prompt values are valid before being passed to the query.



Use the product samples to enhance reports

Prompt API samples are available, to allow you to explore the code to use in your own reports.

Each time you re-use the code, rename the object and all references, to ensure that there are no name collisions.



Prompt API tips

With bubbling or cascading events especially, you may find that code may work in one browser, yet fail in another browser. One example is with onClick cascade.

The samples that are provided use a standard technique to avoid issues. In the slide is an example of a technique that you may use to stop the onClick event from propagating to other controls.

Refer to product documentation for browsers that have been tested with the Prompt API.

IBM Training

Unit summary

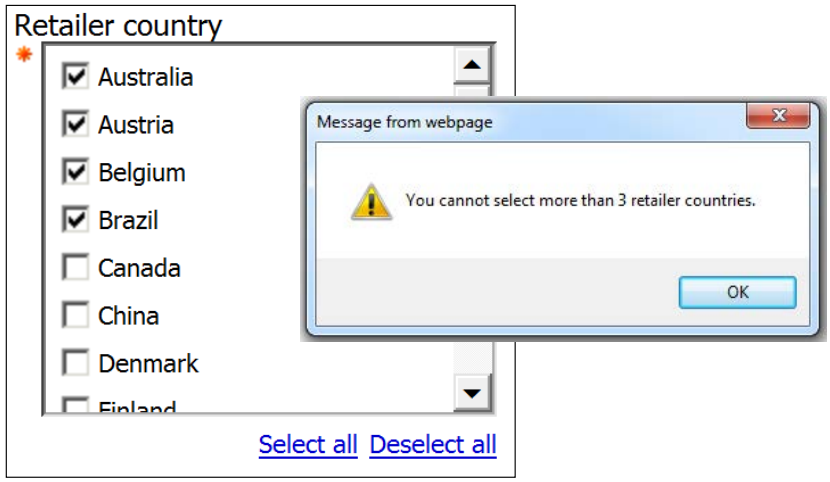
- Describe the Prompt API
- Describe the uses of the Prompt API
- Describe the primary objects, methods and properties used in the Prompt API
- Review prompt values
- Explore the enhancement of prompts in reports with the Prompt API

Unit summary

IBM Training

Exercise 1

Limit user selection to three items



Explore the Prompt API

Exercise 1: Limit user selection to three items

Exercise 1: Limit user selection to three items

As report author Frank Bretton, you have been asked to create a report with a prompt that limits the user selections to three retailer countries.
To accomplish this you will:

- Open a sample Prompt API report (Team content\Samples_Prompt_API\ GO Sales (query)\Limit User Selection to Two Items) that limits the number of user selections, review the descriptive text, and then run it to test the functionality. You will review the contents of the < HTML item > object which contains the sample JavaScript code, and copy this code into a text editor. You will then review the properties of the value prompt code in the report.
- Open an existing report (Team content\Samples\Models\Go data warehouse (query)\IBM Cognos Analytics - Reporting Report Samples\Total Revenue by Country) in IBM Cognos Analytics - Reporting and then run the report, selecting three items in the list. Save this report as Team content\B6059\Total Revenue by Country (Select 3).
- Review the prompt page of the report, and the value prompt properties. Change the Select UI property to check boxes, to make multiple selections easier for the user. Change the object name to something appropriate for Retailer countries. Drag an HTML item into the layout.
- Customize the JavaScript code in the text editor to use the correct prompt references and number of items to be selected. Copy the contents to the clipboard, and then paste into the HTML item object in the prompt layout. Delete the unnecessary Finish button.
- Run the report and test the prompt to ensure that it runs successfully.

For more information about where to work and the exercise results, refer to the Tasks and results section that follows. If you need more information to complete a task, refer to earlier demonstrations for detailed steps.

Exercise 1: Tasks and results

Task 1. Review the sample report and then run it.

- **IBM Cognos Analytics** portal: Navigate to **Team content\Samples_Prompt_API\ GO Sales (query)**
  - Point to the **Limit User Selection to Two Items** report, and then click **Edit report**.
- **Work area**: Read descriptive text.
- **Application** bar: Run the report in **HTML**.
- **Rendered report** tab: Click **Finish**.
  - Close the message.
  - Select check boxes for only two items, and then click **Finish**.
- Close the rendered report tab.

Task 2. Review the code.

- **Navigate** tab: Navigate to **Prompt Page1**.
- **Work area footer**: Double-click the **< HTML item >** object.
- **HTML dialog box**: Review the **validateSelection()** function.
  - Copy the entire code to the clipboard.
- Close the **HTML** dialog box.
- **Desktop**: Open **Notepad**.
- **Notepad**: Paste the code from the clipboard.
- **Prompt Page1 - Properties** pane: Review the properties of the value prompt.

Task 3. Open a report with a prompt.

- **Application** bar: Open the currently active reports list.
  - Click **Welcome**.
- **IBM Cognos Analytics** portal: Navigate to the following report: **Team content\Legacy Samples\Samples\Models\Go Data Warehouse (query)\Report Studio Report Samples**.
  - Point to the **Total Revenue by Country** report to expose the **More** button.
  - Click the **More** button.
  - Click **Edit** report.
- **Application** bar: Run the report in **HTML**.
- **Prompt screen**: Select three items in the list, and then click **Finish**.
- Close the rendered report tab.
- **Application** bar: Save the report as **Team content\B6059** as **Total Revenue by Country (Select 3)**.

Task 4. Update the value prompt and add an HTML object.

- **Navigate** tab: Navigate to **Prompt Page1**.
- **Work area**: Select the value prompt object.
- **Properties** pane: Change the **GENERAL / Select UI** property to **Check box group**.
  - Change the **Name** property to **RC**.
- **Toolbox** tab: Add an **HTML item** object to the right of the **Finish** prompt button, at the bottom of the page.
- Double-click the **< HTML item >** object.

Task 5. Customize the JavaScript code and run the report.

- **Notepad**: Customize the **JavaScrip**t code for this report as follows:
  - Click **Ctrl**+**Home** to return to the top of the code.
  - Replace: **PL**; Replace with: **RC** (do not replace all instances, as there are some valid PL instances; only replace instances of PL that refer to product line instances. For example, do not replace "Sample" to be "SamRCe".) Always start the search at the top of the code.
  - Replace: **productLine**; Replace with: **RC**
  - Replace: **2 Product lines**; Replace with: **3 retailer countries**
  - Replace: **two**; Replace with: **three**
  - Replace: **length > 2**; Replace with: **length > 3**

The results appear as follows:

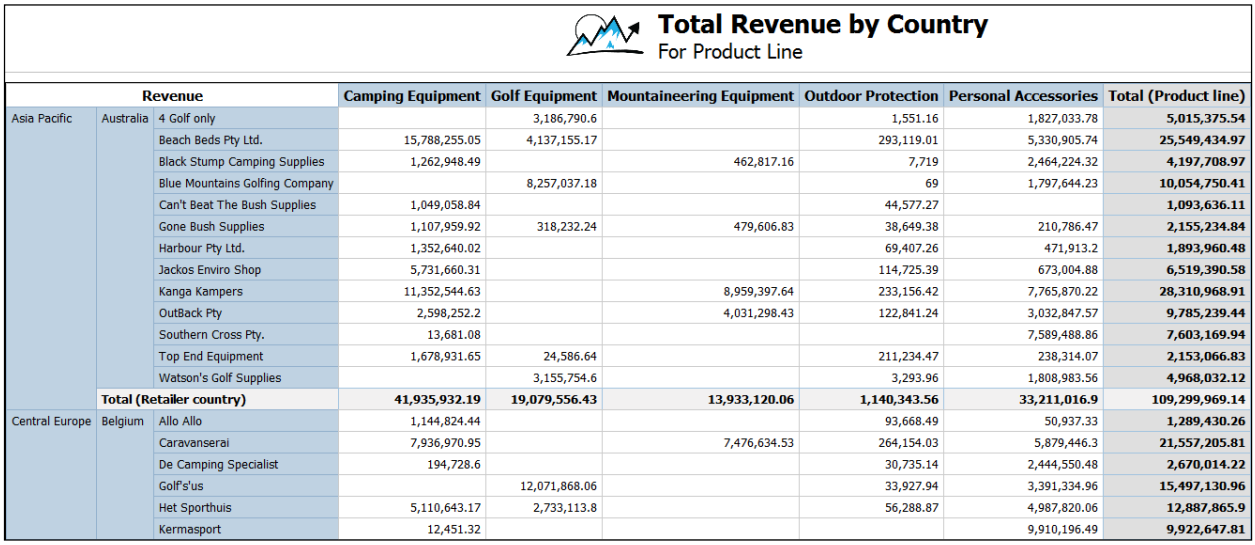
```
<style type="text/css">
.buttonStyle {
     font-size: 90%;
     text-align: center;
     padding-left: 10px;
     padding-right: 10px;
     padding-bottom: 3px;
     padding-top: 2px;
     margin-right: 7px;
     margin-bottom: 7px;
     background-color: #ffffff;
     color: #000000;
     font-weight: normal;
     border: 1px solid #92AFC2;
     background-image: url(../reportstyles/images/button_bg.gif);
     background-position: left bottom;
     background-repeat: repeat-x;
}

</style>

<!--  This button replaces the standard prompt Finish button. It has been
styled to appear like a standard prompt button.  --!>

<button type="button" class="buttonStyle" onClick="vvv.validateSelection()
">Finish</button>

<script>
     /*
      *
```

```
      * Attach functions to an arbitrarily named object to mimic a
namespace to ensure the name uniqueness
      *
      */
     var vvv = { };

     /*
      *
      * This function is called when the user presses the Finish button.
      * It will ensure that the user has not selected more than three
items.
      *
      */
     vvv.validateSelection = function() {
          // Create a report object
          var oCR = cognos.Report.getReport("_THIS_");

          // Create a prompt object
          RCPrompt = oCR.prompt.getControlByName('RC');

          // Create an array of prompt values
          RCValues = RCPrompt.getValues();

          // If the array holds more than three items (prompt values) then
raise an error and return from this function
          if (RCValues && RCValues.length > 3) {
               alert("You cannot select more than 3 retailer countries.");
               return false;
          }
          else {
               // If we have not returned, invoke the Finish action
               setTimeout(vvv.sendFinishAction, 100);
          }
     };

     /*
      *
      * This function is called when the user presses the Finish button.
      * It will ensure that the user has not selected more than three
items.
      *
      */
     vvv.sendFinishAction = function() {
          var oCR = cognos.Report.getReport('_THIS_');
          oCR.sendRequest(cognos.Report.Action.FINISH);
     };

</script>
```

```
<!--
Notice: Source Components and Sample Materials

The Program may include some components in source code form ("Source
Components") and other materials identified as Sample Materials.
Licensee may copy and modify Source Components and Sample Materials for
internal use only provided such use is within the limits of the license
rights
under this Agreement, provided however that Licensee may not alter or
delete any copyright information or notices contained in the Source
Components or Sample Materials. IBM provides the Source Components and
Sample Materials without obligation of support and "AS IS", WITH NO
WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING THE WARRANTY OF
TITLE, NON-INFRINGEMENT OR NON-INTERFERENCE AND THE IMPLIED WARRANTIES AND
CONDITIONS OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.
--!>
```

- **Edit menu**: Select all of the text.
  - Copy the text to the clipboard.
- **HTML dialog box**: Paste the clipboard contents into the **HTML** dialog box.
- **On demand toolbar**: Delete the **Finish** button.
- **Properties** pane: With the prompt block selected, click **Report** form the **Select Ancestor** button.
  - Verify that the **Run with full interactivity** property is set to **No**.
- **Application** bar: Run the report in **HTML**.
- **Rendered report** tab: Test **HTML** object by selecting four items in the list, and then click **Finish**.
  - Close the message.
  - Select three items in the list, and then click **Finish**.

A section of the results appear similar to the following:



Total Revenue by Country
For Product Line

Revenue			Camping Equipment	Golf Equipment	Mountaineering Equipment	Outdoor Protection	Personal Accessories	Total (Product line)
Asia Pacific	Australia	4 Golf only		3,186,790.6		1,551.16	1,827,033.78	**5,015,375.54**
		Beach Beds Pty Ltd.	15,788,255.05	4,137,155.17		293,119.01	5,330,905.74	**25,549,434.97**
		Black Stump Camping Supplies	1,262,948.49		462,817.16	7,719	2,464,224.32	**4,197,708.97**
		Blue Mountains Golfing Company		8,257,037.18		69	1,797,644.23	**10,054,750.41**
		Can't Beat The Bush Supplies	1,049,058.84			44,577.27		**1,093,636.11**
		Gone Bush Supplies	1,107,959.92	318,232.24	479,606.83	38,649.38	210,786.47	**2,155,234.84**
		Harbour Pty Ltd.	1,352,640.02			69,407.26	471,913.2	**1,893,960.48**
		Jackos Enviro Shop	5,731,660.31			114,725.39	673,004.88	**6,519,390.58**
		Kanga Kampers	11,352,544.63		8,959,397.64	233,156.42	7,765,870.22	**28,310,968.91**
		OutBack Pty	2,598,252.2		4,031,298.43	122,841.24	3,032,847.57	**9,785,239.44**
		Southern Cross Pty.	13,681.08				7,589,488.86	**7,603,169.94**
		Top End Equipment	1,678,931.65	24,586.64		211,234.47	238,314.07	**2,153,066.83**
		Watson's Golf Supplies		3,155,754.6		3,293.96	1,808,983.56	**4,968,032.12**
	Total (Retailer country)		**41,935,932.19**	**19,079,556.43**	**13,933,120.06**	**1,140,343.56**	**33,211,016.9**	**109,299,969.14**
Central Europe	Belgium	Allo Allo	1,144,824.44			93,668.49	50,937.33	**1,289,430.26**
		Caravanserai	7,936,970.95		7,476,634.53	264,154.03	5,879,446.3	**21,557,205.81**
		De Camping Specialist	194,728.6			30,735.14	2,444,550.48	**2,670,014.22**
		Golf's'us		12,071,868.06		33,927.94	3,391,334.96	**15,497,130.96**
		Het Sporthuis	5,110,643.17	2,733,113.8		56,288.87	4,987,820.06	**12,887,865.9**
		Kermasport	12,451.32				9,910,196.49	**9,922,647.81**

- Close the rendered report tab.
- Sign out as **Frank Bretton** without saving the report.
- Close the web browser.
- Close **Notepad** without saving changes.

IBM Training