IBM Cognos Framework Manager: Design Metadata Models (v10.2):

Student Guide Vol 1

Course Code: B5252

IBM Cognos Framework Manager: Design Metadata Models (v10.2)

B5252

ERC: 1.0

Published October 2012

Contents

Course Overview

IBM Cognos BI Framework Manager: Design Metadata Models (v10.2) is a five-day, instructor-led course that provides participants with introductory to advanced knowledge of metadata modeling concepts, and how to model metadata for predictable reporting and analysis results using Framework Manager. Participants will learn the full scope of the metadata modeling process, from initial project creation, to publishing of metadata to the Web, enabling end users to easily author reports and analyze data.

Intended Audience

- Developers who design metadata models for use in IBM Cognos BI

Topics Covered

Topics covered in this course include:

- Overview of IBM Cognos BI
- Identify common data structures
- Gather requirements
- Create a baseline project
- Prepare reusable metadata
- Model for predictable results
- Create calculations and filters
- Implement a time dimension
- Specify determinants
- Create the presentation view
- Work with different query subject types
- Set security in Framework Manager
- Create analysis objects
- Manage OLAP data sources
- Advanced generated SQL concepts and complex queries
- Use advanced parameterization techniques in Framework Manager

- Model maintenance and extensibility
- Optimize and tune Framework Manager models
- Work in a multi-modeler environment
- Manage packages in Framework Manager

Course Prerequisites

Required:

- Knowledge of common industry standard data structures and design
- Experience with SQL
- Experience gathering requirements and analyzing data

Recommended:

IBM Cognos BI Report Studio: Author Professional Reports Fundamentals (v10.1) or CBT equivalent

Course Structure

This course has been developed so that some modules build on each other. If you skip a module, you may encounter issues as you try to complete other modules.

Note that the course has been tested in sequential order of the modules. Any deviation from this order may produce different results, including screen captures of directory structures and saved reports.

Course Environment

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

- Apache Directory Server
- DB2 -DB2COPY 1 - DB2
- DB2 Remote Command Server(DB2COPY1)
- DB2 Governor (DB2COPY1)
- DB2DAS - DB2FAS00
- IBM Cognos
- IIS Admin
- Lotus Domino Server (D:\Program Files\IBM\LotusDomino\data)
- World Wide Web Publishing

To review the services, in the System tray 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 demos and workshops.

If Apache Directory Server has stopped, be sure to stop the IBM Cognos service, start the Apache Directory Server service, and then start the IBM Cognos service once Apache has 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.

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 demo and workshop step-by-step solutions to indicate either: • actionable items (Point to **Sort**, and then click **Ascending**.) • text to type or keys to press (Type **Sales Report**, and then press **Enter**.) • UI elements that are the focus of attention (In the **Format** pane, click **Data**)
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.

Workshops

Workshop Format

Workshops are designed to allow you to work according to your own pace. The workshops are structured as follows:

The Business Question Section

The first page of each workshop presents a business-type question followed by a series of steps. These steps provide additional information to help guide you through the workshop. Within each step, there may be numbered questions relating to the step. Solve the tasks by using the skills you learned in this module and in previous ones. If you need more assistance, you can refer to the Task Table section that provides more detailed instruction.

The Task Table Section

The second page of the workshop is a Task Table that presents the question as a series of numbered tasks to be accomplished. The first column in the table states the task to be accomplished. The second column, "Where to Work", indicates the area of the product to work in. Finally, the third column provides some hints that may help you complete the workshop.

The Workshop Results Section

This section will contain a screen capture(s) of interim or final results and/or answers to the questions asked in the Business Question section.

Your Feedback

Your feedback is important and valuable. We are interested in your comments or questions.

Please address them to:

Attn: Senior Manager, Customer Education

Email Address: Cognos.Education@ca.ibm.com

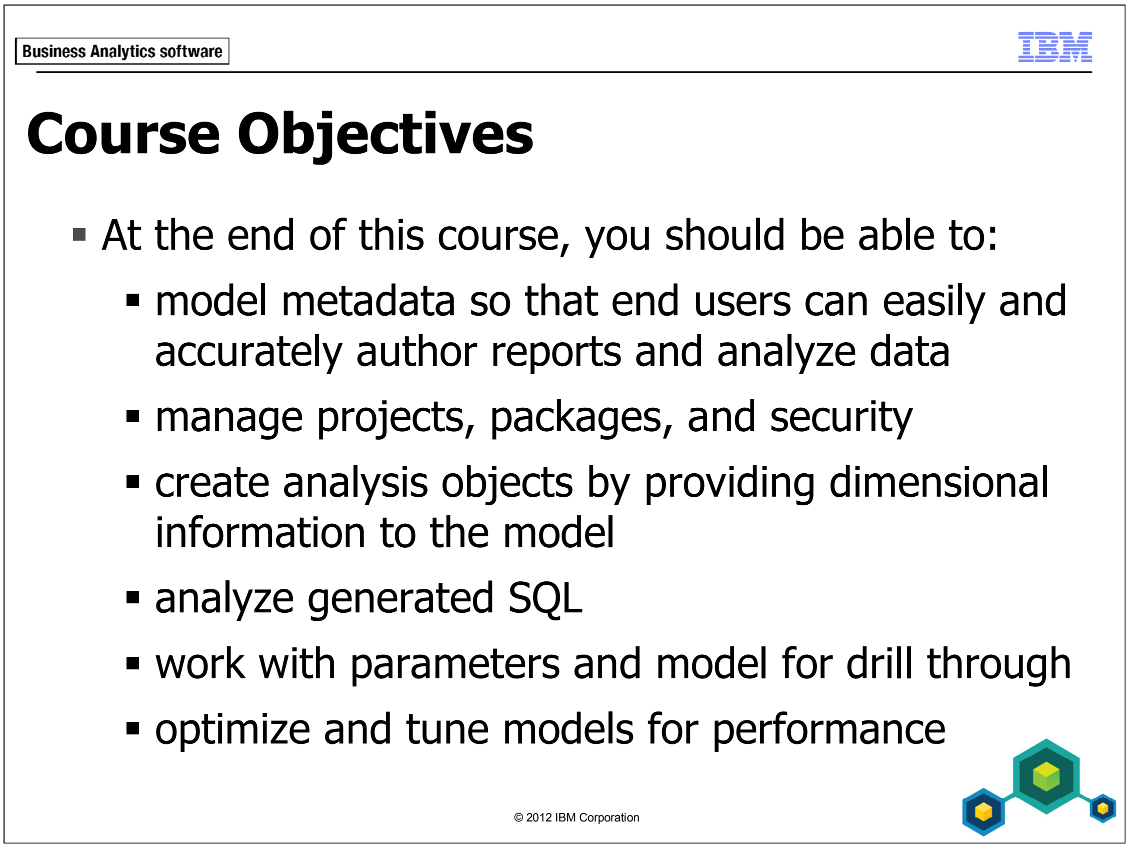
IBM Product Help

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:**	
	• Training and Certification Web site	• http://www-01.ibm.com/software/data/education/cognos.html
	• Online support	• http://www-947.ibm.com/support/entry/portal/Overview/Software/Cognos/Cognos_Business_Intelligence_and_Financial_Performance_Management
	• IBM Web site	• http://www.ibm.com

Introduction

IBM Cognos BI

Business Analytics software



Business Analytics software

Course Objectives

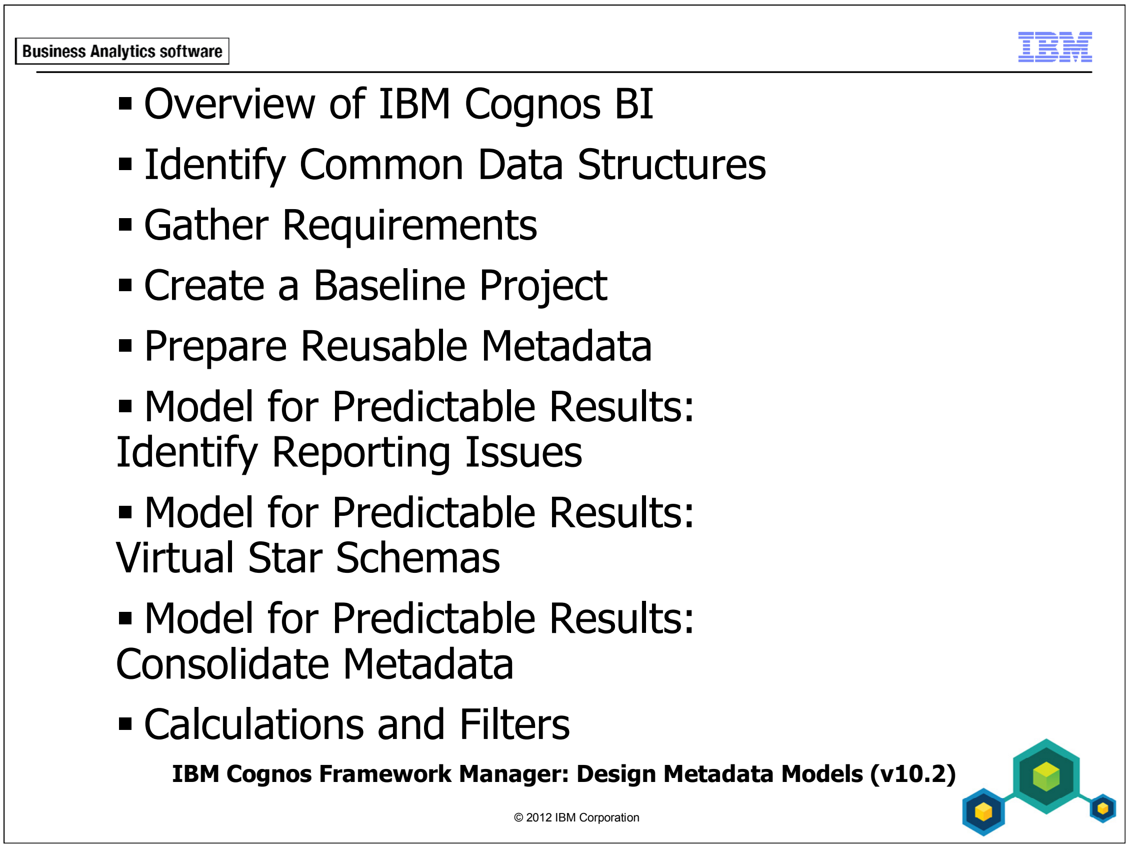
- At the end of this course, you should be able to:
  - model metadata so that end users can easily and accurately author reports and analyze data
  - manage projects, packages, and security
  - create analysis objects by providing dimensional information to the model
  - analyze generated SQL
  - work with parameters and model for drill through
  - optimize and tune models for performance

© 2012 IBM Corporation

IBM Cognos BI Framework Manager: Design Metadata Models (v10.2) is a five-day, instructor-led course that provides participants with introductory to advanced knowledge of metadata modeling concepts, and how to model metadata for predictable reporting and analysis results using Framework Manager. Participants will learn the full scope of the metadata modeling process, from initial project creation, to publishing of metadata to the Web, enabling end users to easily author reports and analyze data.

Intended Audience

- Developers who design metadata models for use in IBM Cognos BI


Business Analytics software

- Overview of IBM Cognos BI
- Identify Common Data Structures
- Gather Requirements
- Create a Baseline Project
- Prepare Reusable Metadata
- Model for Predictable Results: Identify Reporting Issues
- Model for Predictable Results: Virtual Star Schemas
- Model for Predictable Results: Consolidate Metadata
- Calculations and Filters

IBM Cognos Framework Manager: Design Metadata Models (v10.2)

© 2012 IBM Corporation

Throughout this course, Framework Manager concepts and procedures are grouped into modules that are presented in a logical and structured manner. The hands-on demonstrations and workshops provide the knowledge and skills you will require to create and modify reports.

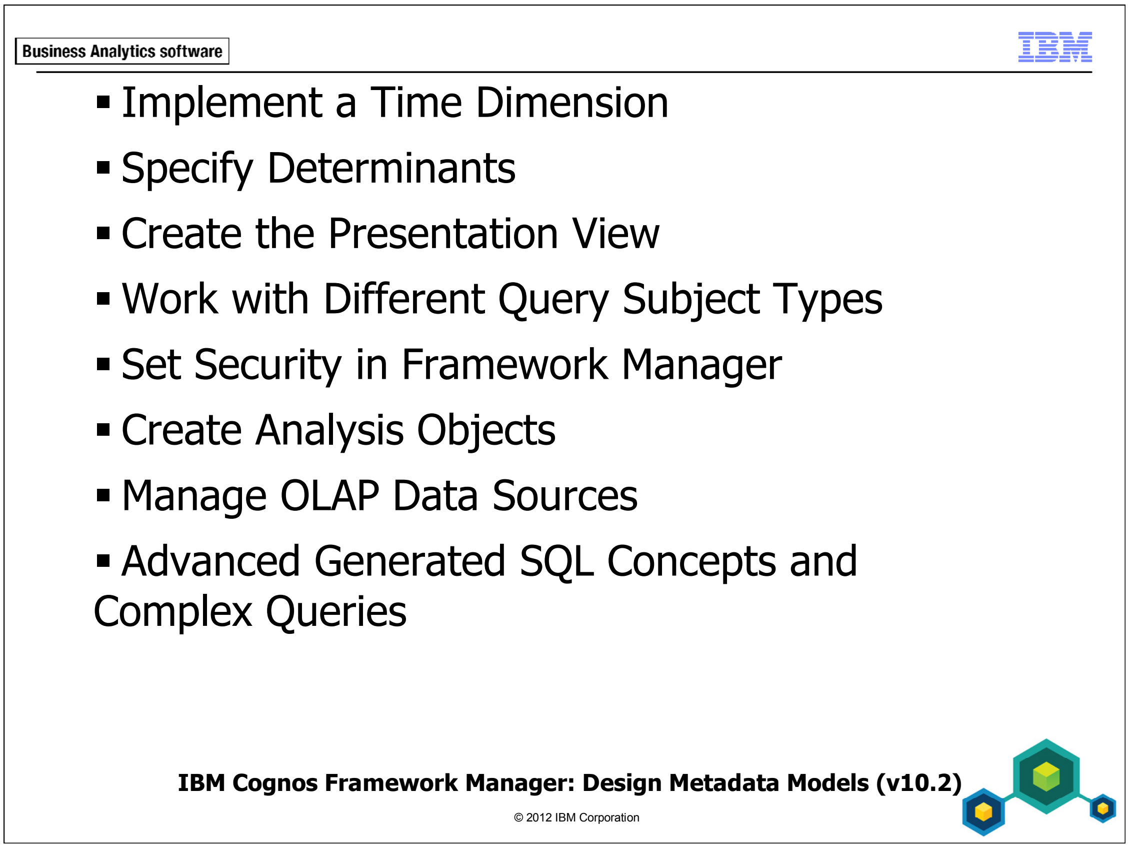
Prerequisites

Required:

- Knowledge of common industry standard data structures and design
- Experience with SQL
- Experience gathering requirements and analyzing data

Recommended:

- IBM Cognos Report Studio: Author Professional Reports Fundamentals (v10.2) or CBT equivalent

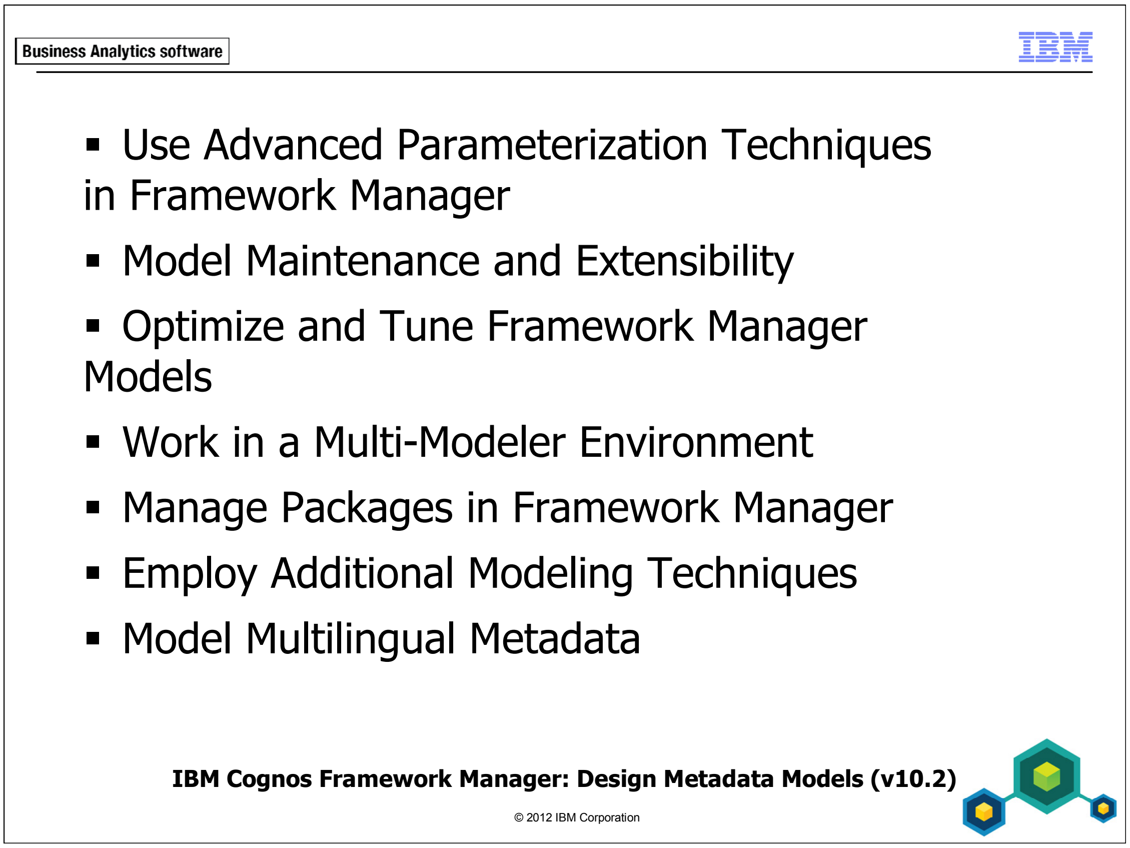


Business Analytics software

- Implement a Time Dimension
- Specify Determinants
- Create the Presentation View
- Work with Different Query Subject Types
- Set Security in Framework Manager
- Create Analysis Objects
- Manage OLAP Data Sources
- Advanced Generated SQL Concepts and Complex Queries

IBM Cognos Framework Manager: Design Metadata Models (v10.2)

© 2012 IBM Corporation

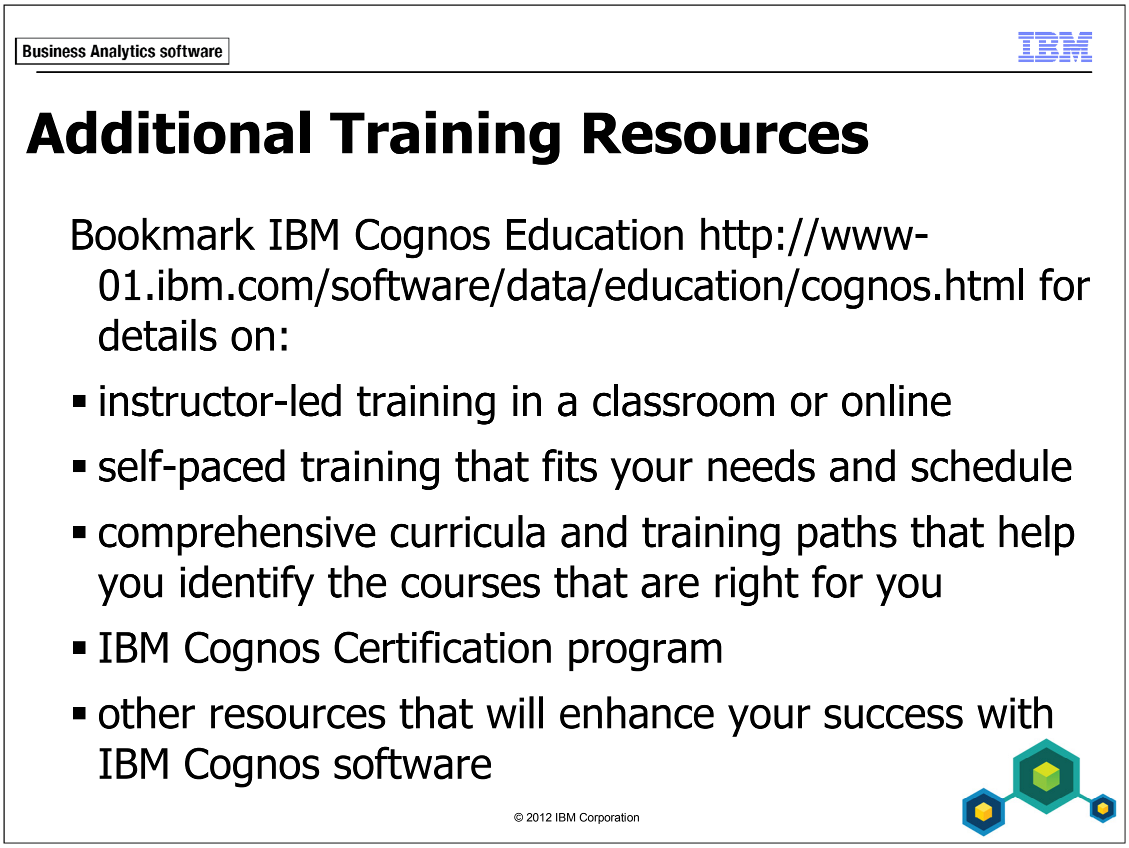


Business Analytics software

- Use Advanced Parameterization Techniques in Framework Manager
- Model Maintenance and Extensibility
- Optimize and Tune Framework Manager Models
- Work in a Multi-Modeler Environment
- Manage Packages in Framework Manager
- Employ Additional Modeling Techniques
- Model Multilingual Metadata

IBM Cognos Framework Manager: Design Metadata Models (v10.2)

© 2012 IBM Corporation



Business Analytics software

Additional Training Resources

Bookmark IBM Cognos Education http://www-01.ibm.com/software/data/education/cognos.html for details on:

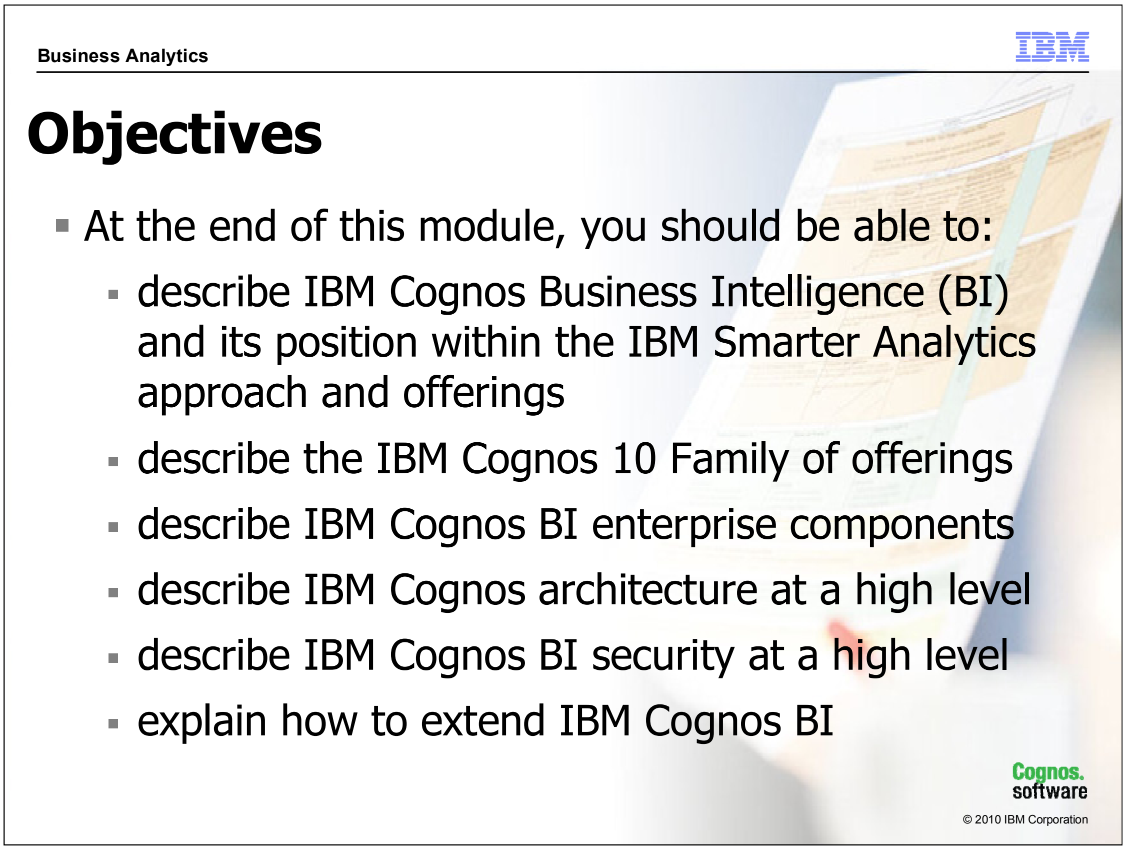
- instructor-led training in a classroom or online
- self-paced training that fits your needs and schedule
- comprehensive curricula and training paths that help you identify the courses that are right for you
- IBM Cognos Certification program
- other resources that will enhance your success with IBM Cognos software

IBM

Overview of IBM Cognos BI

IBM Cognos BI

Business Analytics software



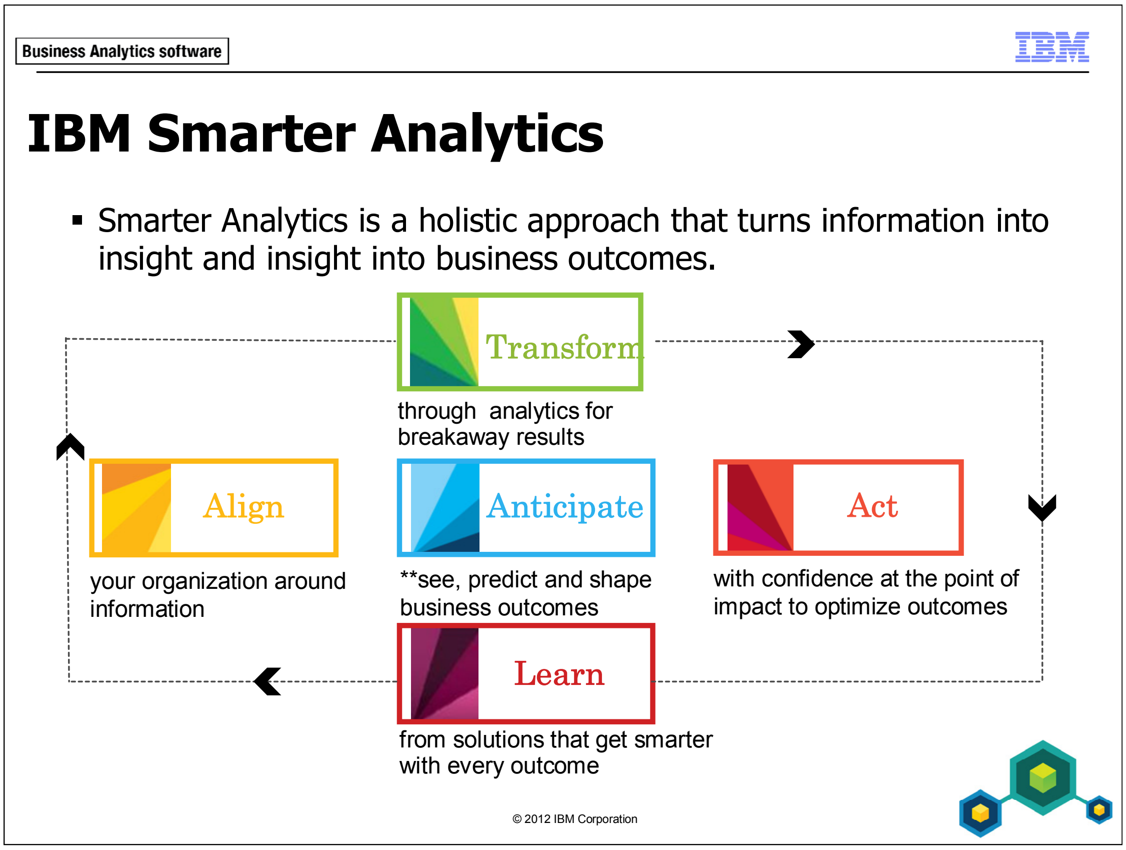
Business Analytics

Objectives

- At the end of this module, you should be able to:
  - describe IBM Cognos Business Intelligence (BI) and its position within the IBM Smarter Analytics approach and offerings
  - describe the IBM Cognos 10 Family of offerings
  - describe IBM Cognos BI enterprise components
  - describe IBM Cognos architecture at a high level
  - describe IBM Cognos BI security at a high level
  - explain how to extend IBM Cognos BI

Cognos. software

© 2010 IBM Corporation



Business Analytics software

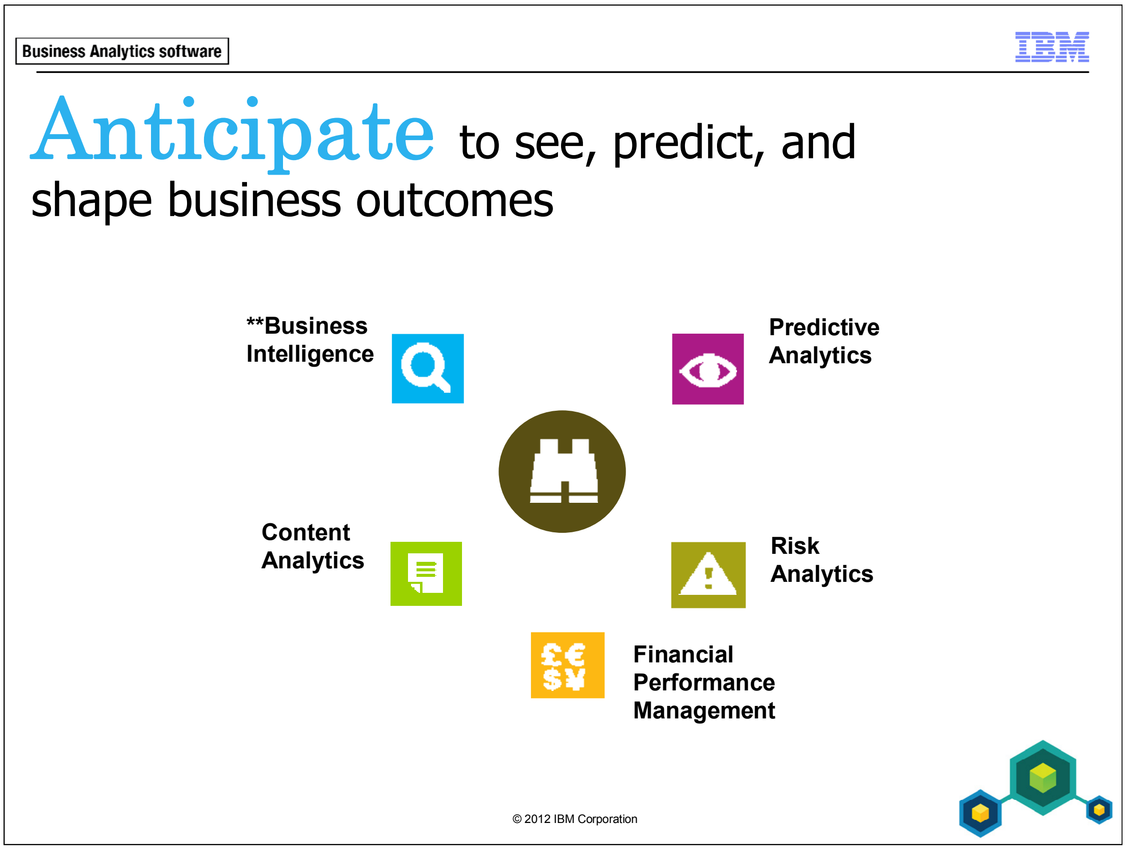
IBM Smarter Analytics

- Smarter Analytics is a holistic approach that turns information into insight and insight into business outcomes.

Transform
through analytics for breakaway results
Align
your organization around information
Anticipate
**see, predict and shape business outcomes
Act
with confidence at the point of impact to optimize outcomes
Learn
from solutions that get smarter with every outcome

© 2012 IBM Corporation

Organizations across industries face tough new challenges created by the information age. A hyper-connected, global community of empowered individuals and consumers is generating an unprecedented amount of big data from billions of diverse sources. Amidst these new complexities, successful organizations are using analytics to acquire, grow and retain customers, transform their financial processes, improve operational efficiency and manage and reduce risk and fraud. Analytics has evolved from a business initiative to a business imperative. Organizations are adopting analytics at a fast rate and those leaders are already transforming entire industries.



Business Analytics software

IBM

Anticipate to see, predict, and shape business outcomes

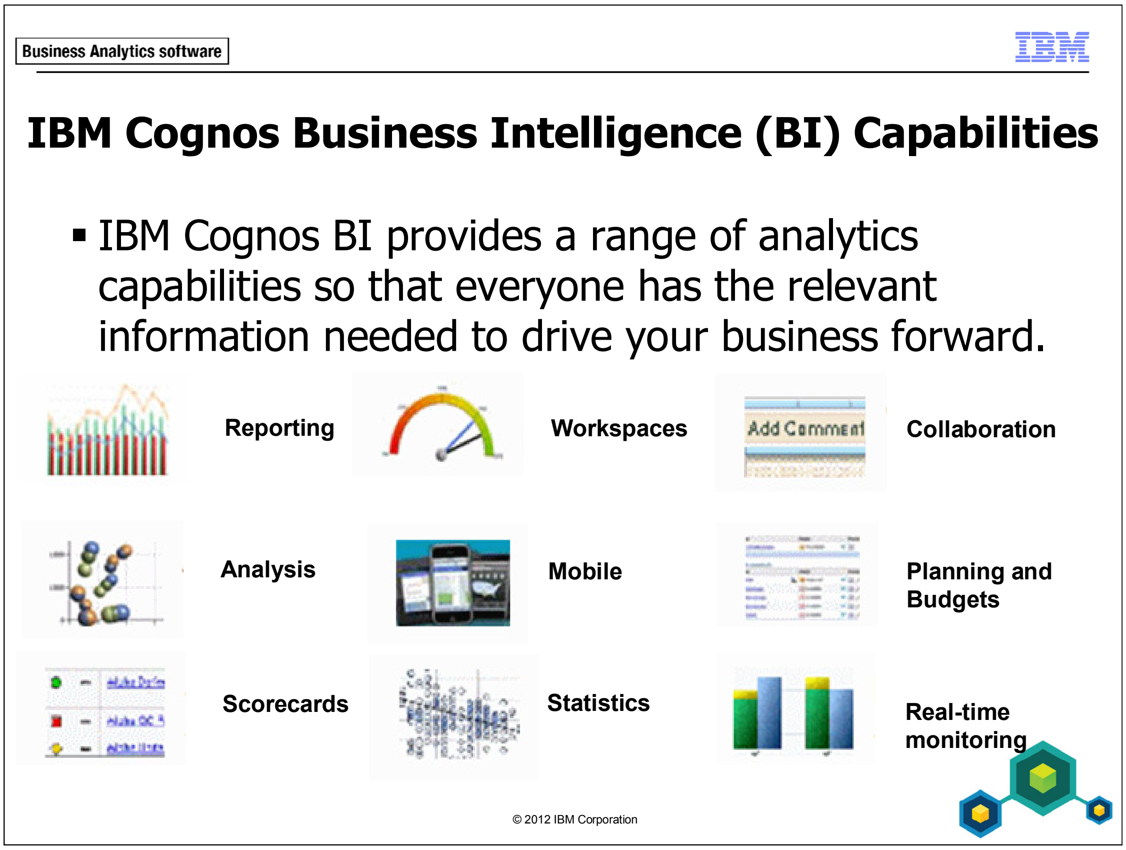
**Business Intelligence
Predictive Analytics
Content Analytics
Risk Analytics
Financial Performance Management

© 2012 IBM Corporation

Leverage business analytics to deliver actionable insights.

- Spot and analyze trends and anomalies
- Predict potential threats and opportunities
- Plan, budget, and forecast resources
- Assess and manage risk
- Compare "what-if" scenarios
- Measure and monitor business performance
- Automate decisions
- Align strategic and operational decisions

Business Intelligence – capabilities offered by IBM Cognos BI suite of products
Predictive Analytics – capabilities offered by IBM SPSS suite of products.
Risk Analytics – capabilities offered by Open Pages and Algorithmics suite of products
Financial Performance Management – capabilities offered by IBM Clarity, IBM Cognos Controller, IBM Cognos Planning, and IBM Cognos TM1
Content Analytics – capabilities offered by IBM Cognos Content Analytics



Business Analytics software

IBM Cognos Business Intelligence (BI) Capabilities

- IBM Cognos BI provides a range of analytics capabilities so that everyone has the relevant information needed to drive your business forward.

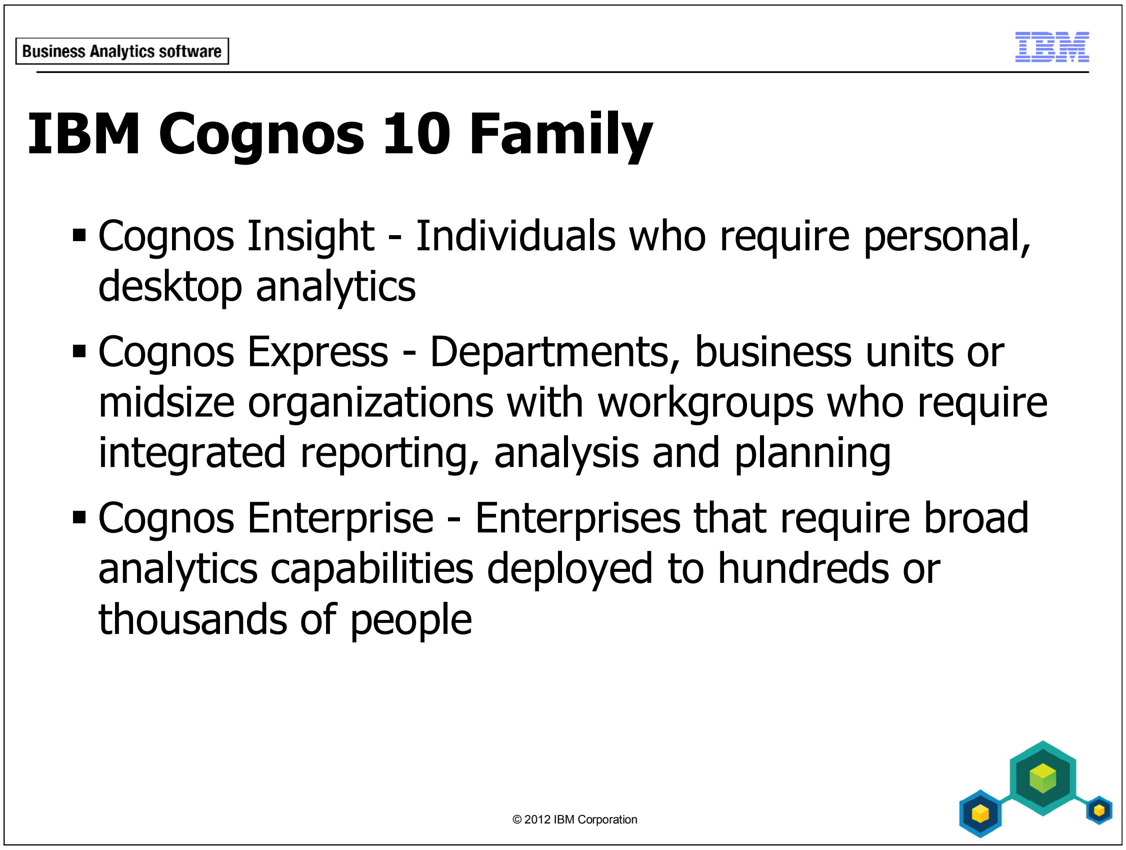
Reporting
Workspaces
Collaboration
Add Comment
Analysis
Mobile
Planning and Budgets
Scorecards
Statistics
Real-time monitoring

© 2012 IBM Corporation

With IBM Cognos BI, users can:

- explore information freely, analyze key facts, collaborate to gain alignment with key stakeholders and make decisions for better business outcomes
- access reports, analysis, dashboards, scorecards, planning and budgets, real-time information, statistics and manage information for more informed decisions.
- integrate the results of "what-if" analysis modeling and predictive analytics into a unified workspace to view possible future outcomes alongside current and historical data.

- work with business intelligence capabilities for the office and desktop, on mobile devices, online and offline.
- work within a highly scalable and extensible solution that can adapt to the changing needs of IT and the business with flexible deployment options that include the cloud, mainframes and data warehousing appliances.



Business Analytics software

IBM

IBM Cognos 10 Family

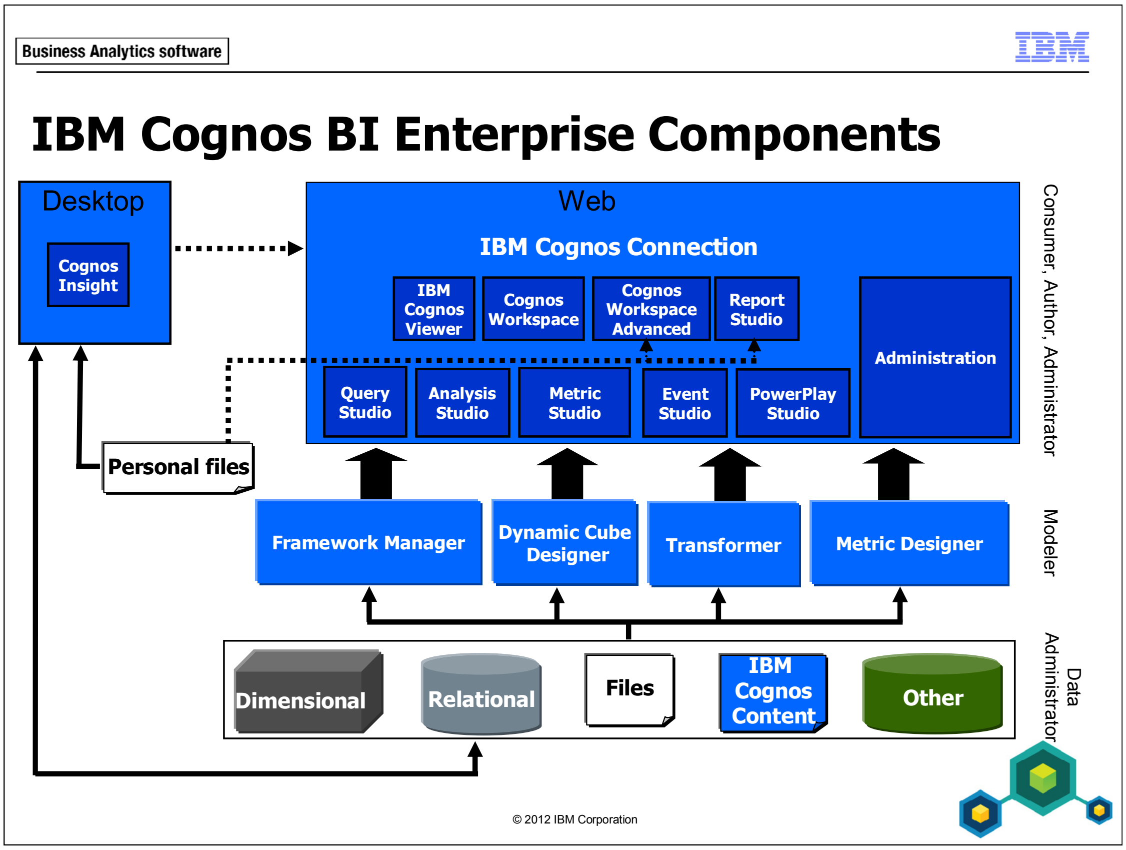
- Cognos Insight - Individuals who require personal, desktop analytics
- Cognos Express - Departments, business units or midsize organizations with workgroups who require integrated reporting, analysis and planning
- Cognos Enterprise - Enterprises that require broad analytics capabilities deployed to hundreds or thousands of people

© 2012 IBM Corporation

The IBM Cognos 10 family of products are right-sized for your organization and integrated together, and offer solutions that meet your current and future needs, whether you want to deploy on a desktop, a single server, a server farm or all three. You can also start small and grow your solution over time. For example:

Start small, using Cognos Insight for data discovery and planning. Add a server to share that insight and create additional reports from larger data sets with Cognos Express. Or combine that insight with real-time and corporate information and place insights on scorecards and interact on mobile devices with Cognos Enterprise.

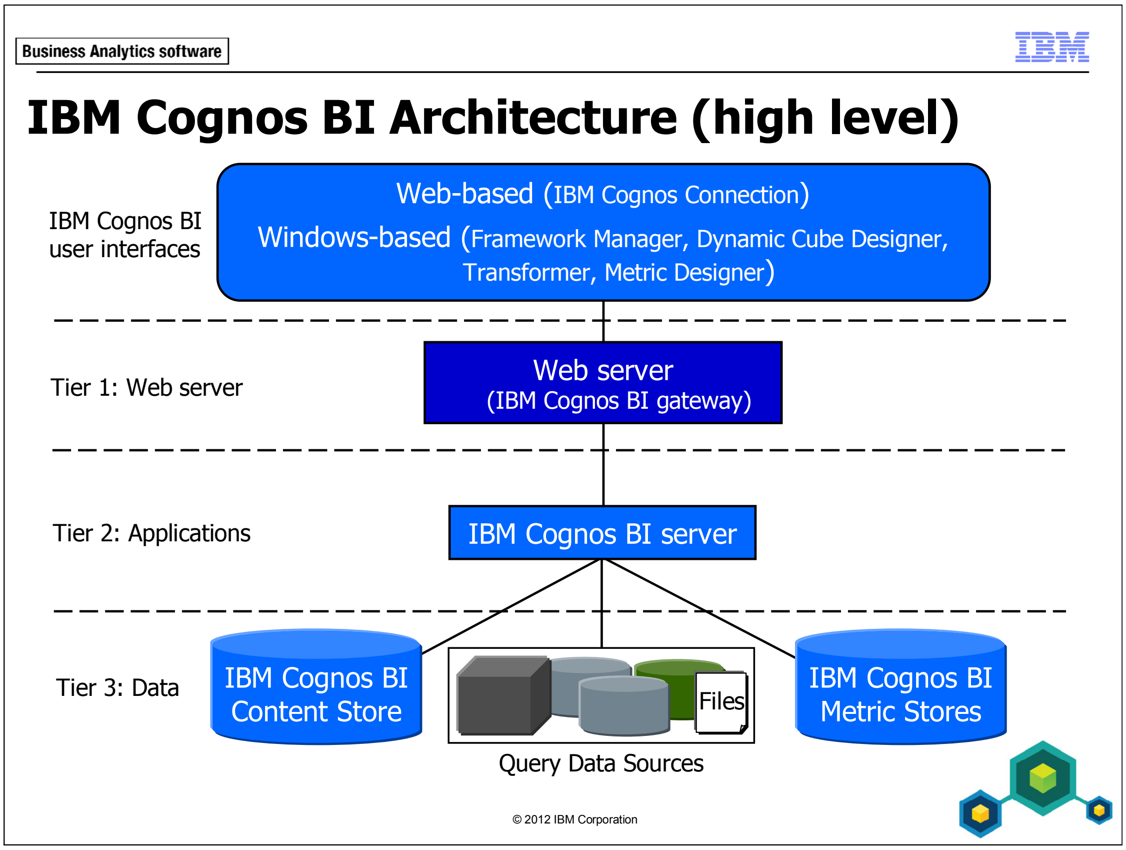
Cognos Express provides a subset of combined tools and functionality from IBM Cognos BI and IBM Cognos TM1 in a solution that provides reporting, analysis, dashboard, scorecard, planning, budgeting and forecasting capabilities for workgroups and midsize companies



IBM Cognos BI capabilities provide reporting, analysis, scorecarding, workspace creation, business event management, and data integration from a wide array of corporate and personal data sources. IBM Cognos BI includes:

- IBM Cognos Connection, which is the Web a portal for BI content presentation, management, and administration.
- Web and desktop reporting and analysis tools to author and analyze corporate data.
- Metadata modeling tools, including Framework Manager, Dynamic Cube Designer, and Transformer.

Use **IBM Cognos Viewer** to view reports
Use **IBM Cognos Workspace** to create personal workspaces
Use **IBM Cognos Workspace Advanced** to perform self-service reporting and analyses of data, including external data files
Use **IBM Cognos Insight** to perform personal analysis in a desktop environment
Use **Query Studio** to perform ad hoc querying and quickly answer a focused question
Use **Analysis Studio** to perform analyses of data to discover trends, risks, and opportunities
Use **Report Studio** to build sophisticated reports, against multiple data sources, including external data files
Use **Event Studio** to create agents which notify users of key operational or performance-related events in their business
Use **Metric Studio** to manage performance by monitoring and analyzing metrics
Use **Framework Manager** to create basic query packages or relationally-based dimensional analysis packages
Use **Dynamic Cube Designer** to create, edit, import, export, and deploy virtual cube models over a relational warehouse schema.
Use **Transformer** to create PowerCubes for dimensional analysis.
Use **Metric Designer** to create scorecard applications for use in Metric Studio.



Business Analytics software

IBM Cognos BI Architecture (high level)

IBM Cognos BI user interfaces
Web-based (IBM Cognos Connection)
Windows-based (Framework Manager, Dynamic Cube Designer, Transformer, Metric Designer)
Tier 1: Web server
Web server (IBM Cognos BI gateway)
Tier 2: Applications
IBM Cognos BI server
Tier 3: Data
IBM Cognos BI Content Store
Files
Query Data Sources
IBM Cognos BI Metric Stores

© 2012 IBM Corporation

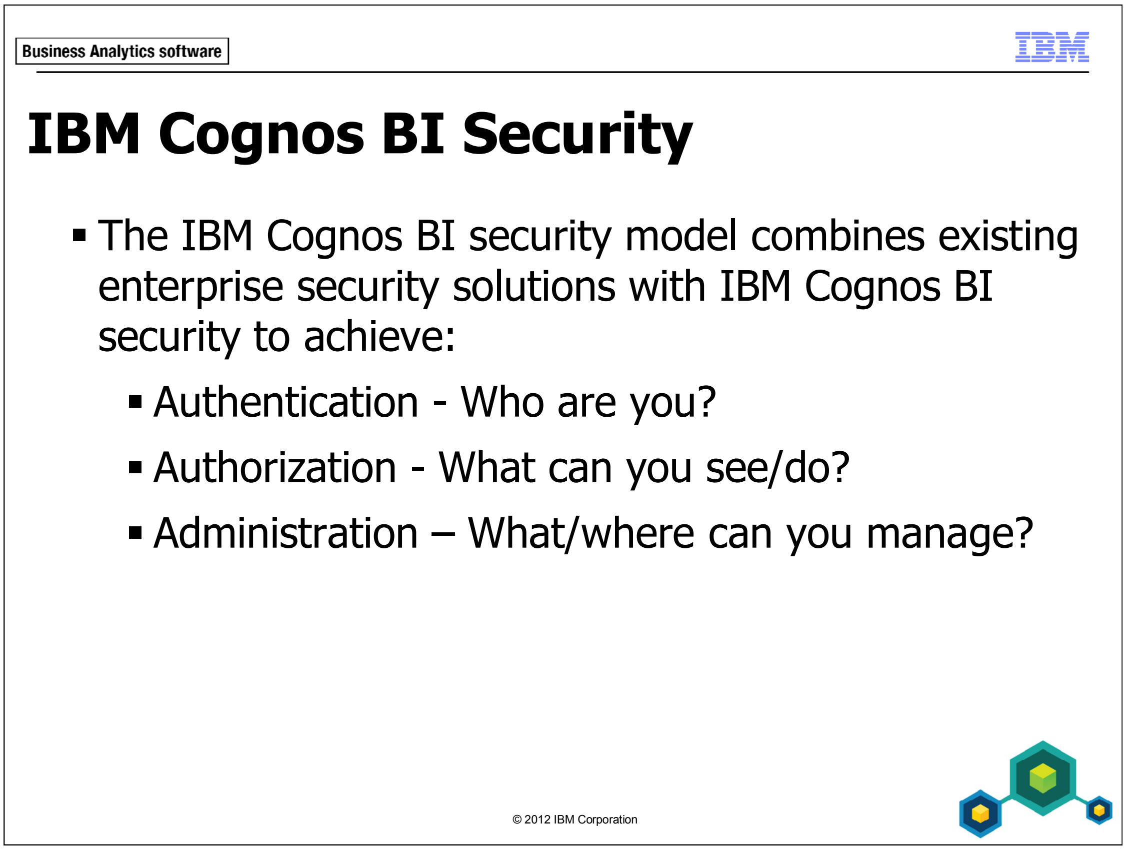
IBM Cognos BI is a Web-based architecture, which is separated into three tiers; Web server, applications, and data.

This architecture is scalable from a software and hardware perspective. For example, you can have several IBM Cognos servers for faster response times and load balancing.

IBM Cognos leverages existing corporate IT resources such as web servers, authentication providers, and application servers, and also supports multiple languages and locales in order to serve a global audience.

IBM Cognos is customizable to adopt your corporate look and feel and can be extended and integrated into other applications through the IBM Cognos SDK.

This is a high level diagram. Detailed information can be found in the IBM Cognos Administration course regarding scalability and flexibility. An example of flexibility is that SDK applications and Framework Manager can bypass the gateway and communicate directly with the IBM Cognos Server.
IBM Cognos BI, IBM Cognos Planning and IBM Cognos Controller all use the IBM Cognos architecture. Planning and Controller portions are not illustrated here.



Business Analytics software

IBM Cognos BI Security

- The IBM Cognos BI security model combines existing enterprise security solutions with IBM Cognos BI security to achieve:
  - Authentication - Who are you?
  - Authorization - What can you see/do?
  - Administration – What/where can you manage?

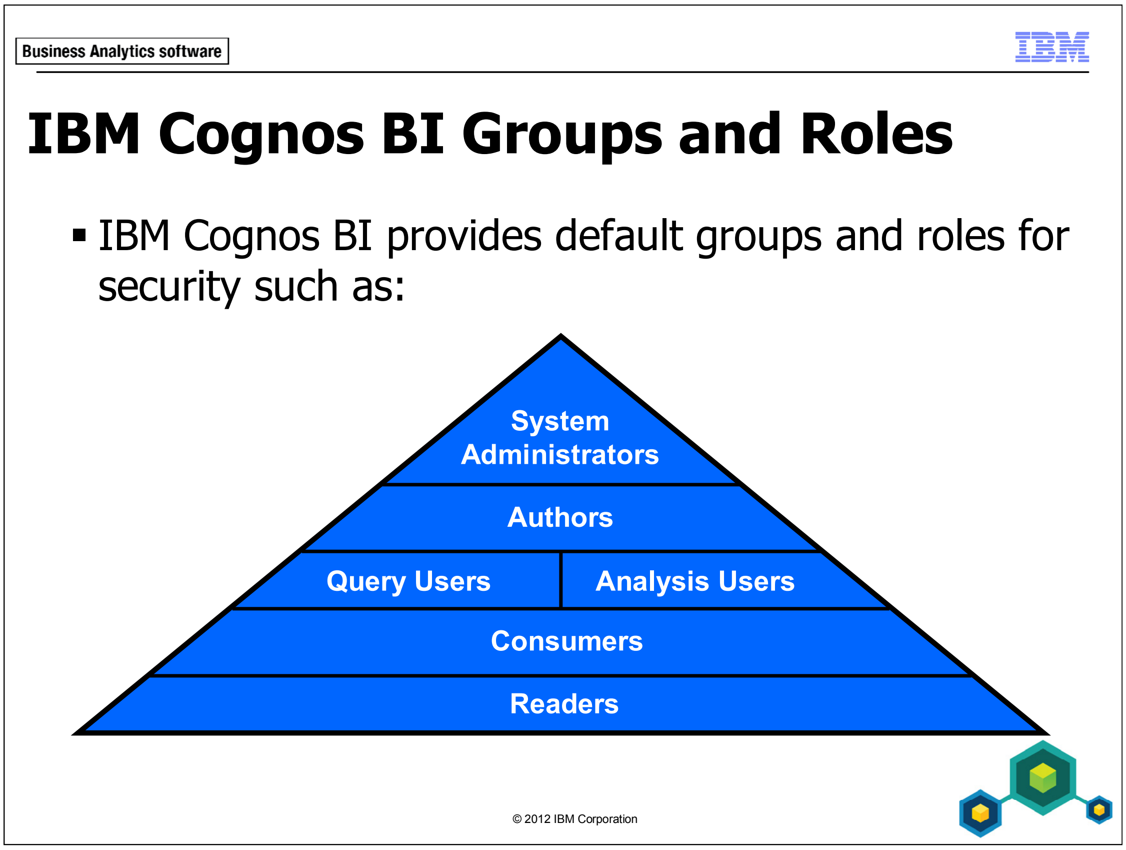
© 2012 IBM Corporation

IBM Cognos BI authentication is based on the use of third party authentication providers. These define users, groups, and roles used for authentication. User names, IDs, passwords, regional settings, and personal preferences are some examples of information stored in the providers.

Authorization is the process of granting or denying access to content, and specifying the actions that can be performed on that content, based on a user identity. Authorization assigns permissions to users, groups, and roles that allow them to perform actions, such as read or write, on objects, such as folders and reports. Permissions can be granted to users, groups, or roles directly from authentication providers or through membership in Cognos namespace groups and roles.

The Cognos namespace is the built-in namespace from IBM Cognos BI. It contains the IBM Cognos objects, such as groups, roles, data sources, distribution lists, and contacts. During the content store initialization, built-in and predefined security entries are created in this namespace, and include default access to functionality.

You can configure and administer IBM Cognos BI security using IBM Cognos Configuration and IBM Cognos Administration.



Business Analytics software

IBM Cognos BI Groups and Roles

- IBM Cognos BI provides default groups and roles for security such as:

System Administrators
Authors
Query Users
Analysis Users
Consumers
Readers

Take advantage of IBM Cognos BI groups and roles from the Cognos namespace to secure your IBM Cognos environment and content. The group or role to which a user belongs determines how much access the user has to the IBM Cognos environment and functionality. For example, if you are a member of only the Consumers role, you cannot access any of the IBM Cognos studios.

Besides the default groups and roles, you can create new groups and roles that are specific to your IBM Cognos needs. Simply add users from your authentication source to specific groups and roles as required.

Not only can you use the groups and roles defined in the IBM Cognos namespace to control access to contents, you can use groups in your authentication provider as well.

Using the IBM Cognos namespace does not require the IT department and creates a more portable environment. There are many different groups and roles the administrator can use to restrict what you can see, what you can do, etc. See the Predefined Entries section of the Administration and Security Guide for detailed information on the predefined groups and roles as well as the anonymous user.

Demo 1: Explore IBM Cognos BI

Purpose:
As an introduction to IBM Cognos BI, you will briefly explore one of the modeling tools, the portal, and various BI studios and desktop application to familiarize yourself with the environment and BI workflow.

Task 1. Explore IBM Cognos Framework Manager.

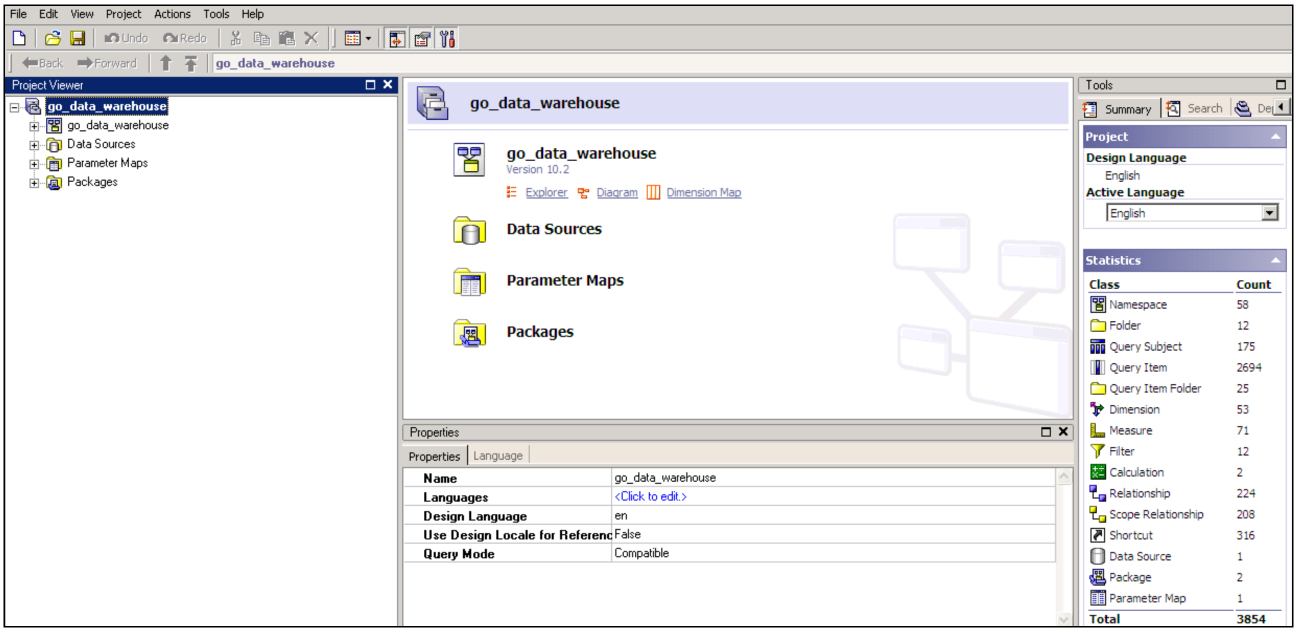
You will examine a model in IBM Cognos Framework Manager.

1. From the **Start** menu, point to **All Programs\IBM Cognos 10**, and then click **IBM Cognos Framework Manager**.
2. Click **Open a project**, navigate to **D:\Program Files\IBM\cognos\c10\webcontent\samples\models\great_outdoors_warehouse**, and then click **great_outdoors_warehouse.cpf**.
3. Click **Open**.

In all tasks, ensure that you are logged in as admin/Education1.

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

   The results appear as follows:



Name	go_data_warehouse
Languages	<Click to edit.>
Design Language	en
Use Design Locale for Referenc	False
Query Mode	Compatible

Project
Design Language: English
Active Language: English

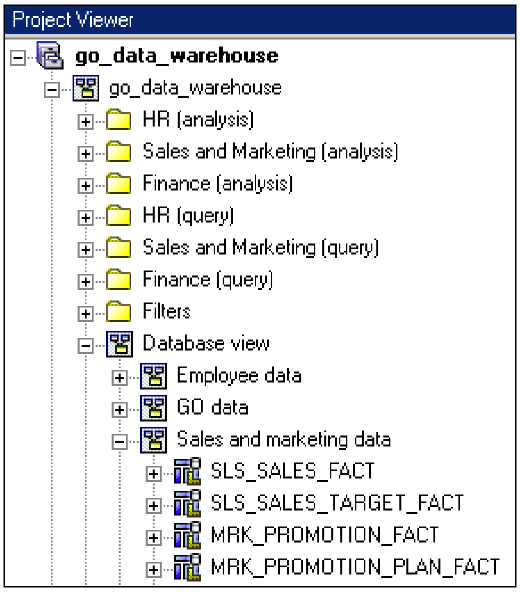
Statistics

Class	Count
Namespace	58
Folder	12
Query Subject	175
Query Item	2694
Query Item Folder	25
Dimension	53
Measure	71
Filter	12
Calculation	2
Relationship	224
Scope Relationship	208
Shortcut	316
Data Source	1
Package	2
Parameter Map	1
Total	3854

IBM Cognos Framework Manager is a modeling tool used to create packages for use in the IBM Cognos studios. Its purpose is to reduce the complexity of the underlying data source by creating user-friendly views of the business for authors and analysts in the IBM Cognos environment. IBM Cognos Framework Manager can also be used to model and publish SAP BW metadata packages as well as publish other Online Analytical Processing (OLAP) packages such as Microsoft Analysis Services cubes.

5. In the **Project Viewer** pane, expand **go_data_warehouse\Database view\Sales and marketing data**.

   The results appear as follows:

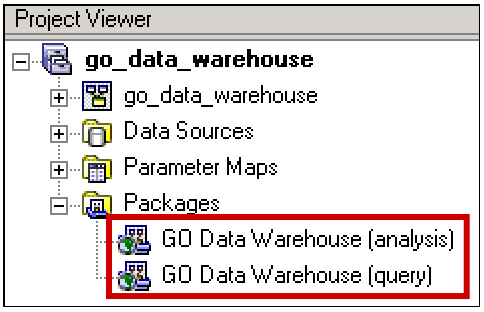


```
go_data_warehouse
  go_data_warehouse
    HR (analysis)
    Sales and Marketing (analysis)
    Finance (analysis)
    HR (query)
    Sales and Marketing (query)
    Finance (query)
    Filters
    Database view
      Employee data
      GO data
      Sales and marketing data
        SLS_SALES_FACT
        SLS_SALES_TARGET_FACT
        MRK_PROMOTION_FACT
        MRK_PROMOTION_PLAN_FACT
```

   Metadata modelers work in a project with several object types to create organized views of the business with any required business logic, and remove any ambiguity that might occur. These views are then placed in packages and published to IBM Cognos Connection.

6. Collapse **go_data_warehouse**, and then expand **Packages**.

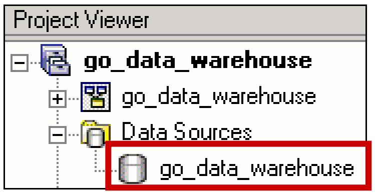
   The results appear as follows:



   Packages can be a subset of the metadata model, or the entire model depending on the requirements. Once published, authors and analysts can choose from the objects contained in the package to create reports. Take note of the package names, as you will see them in IBM Cognos Connection shortly.

7. Expand **Data Sources**.

   The results appear as follows:



   Data Sources contain the information that IBM Cognos uses to connect to and retrieve data from the underlying data sources cited in the model.

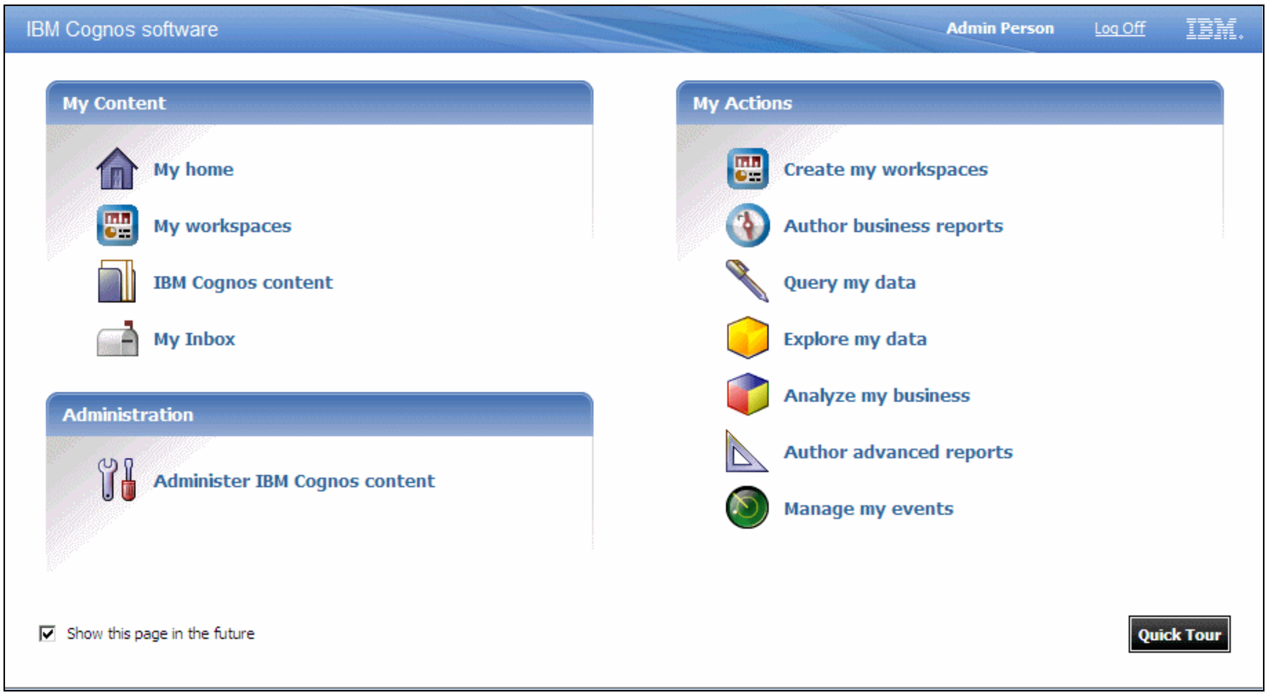
8. Close **IBM Cognos Framework Manager**.

Task 2. Explore IBM Cognos Connection.

You will review the centralized portal and navigate the content structure of packages and reports.

1. Start **Internet Explorer**, and then in the url box, type **http://localhost:88/ibmcognos**.
2. Log on as **admin/Education1**.

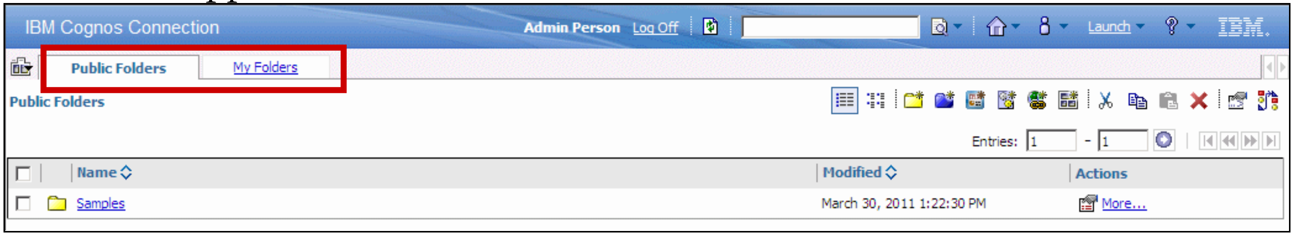
The results appear as follows:



By default, you are presented with a welcome page from which you can choose several activities to perform, such as to query or analyze your data, or perform administrative tasks. Depending on the role to which you belong, you may have more or less capabilities in the IBM Cognos environment.

3. Click **IBM Cognos content**.

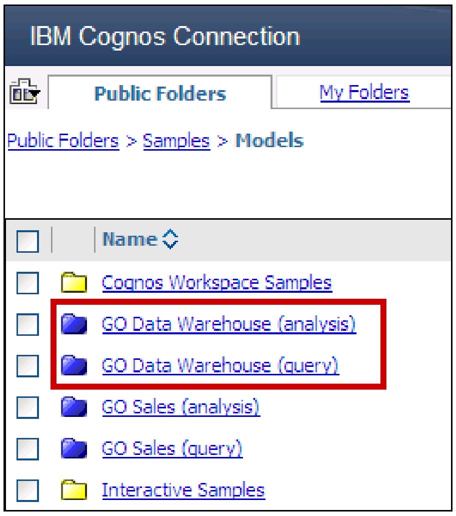
   The results appear as follows:



Here you will find public content, found on the Public Folders tab, or personal content, found on the My Folders tab. You also have the ability to create more personal tabs to suit your needs or to share with others. Content, including reports, analyses, and portal pages, are managed and organized in this area.

4. Click **Samples**, and then click **Models**.

   The results appear as follows:



IBM Cognos Connection

Public Folders | My Folders

Public Folders > Samples > Models

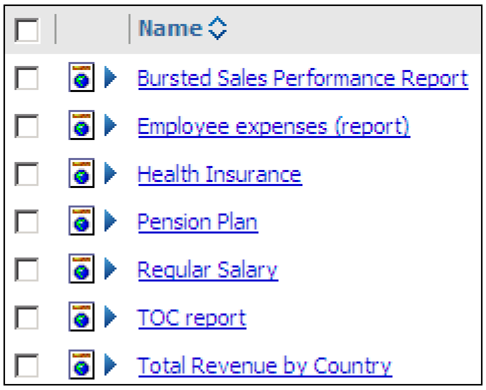
	Name
☐	Cognos Workspace Samples
☐	GO Data Warehouse (analysis)
☐	GO Data Warehouse (query)
☐	GO Sales (analysis)
☐	GO Sales (query)
☐	Interactive Samples

   Notice the packages named GO Data Warehouse (analysis) and GO Data Warehouse (query). These are the packages you saw in the IBM Cognos Framework Manager project earlier that have been published here.

   Packages, by default, are indicated by a blue folder, and regular folders are yellow in color. Both types of folders can contain reports, analyses and other content.

5. Click **GO Data Warehouse (query)**, and then click **Report Studio Report Samples**.

   The results appear as follows:



	Name
☐	Bursted Sales Performance Report
☐	Employee expenses (report)
☐	Health Insurance
☐	Pension Plan
☐	Regular Salary
☐	TOC report
☐	Total Revenue by Country

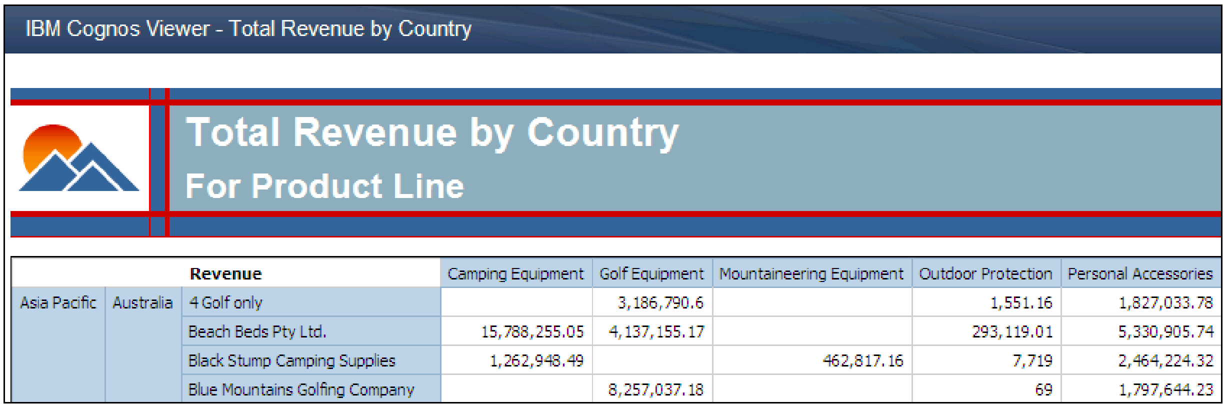
   Here you see several reports based on the GO Data Warehouse (query) package, a model based on a relational data source.

Task 3. View reports in IBM Cognos.

You will open an existing report based on a relational package and a report based on a dimensional data package.

1. Click **Total Revenue by Country**.
2. Click **Select all**, and then click **Finish**.

The results appear as follows:



IBM Cognos Viewer - Total Revenue by Country

Total Revenue by Country
For Product Line

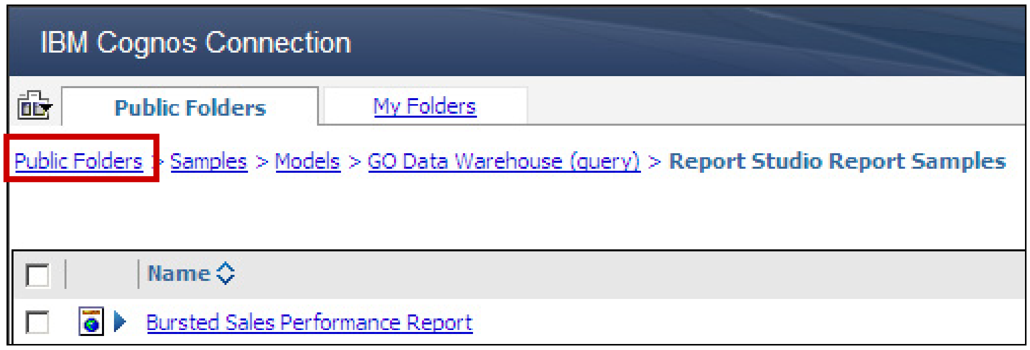
		Revenue	Camping Equipment	Golf Equipment	Mountaineering Equipment	Outdoor Protection	Personal Accessories
Asia Pacific	Australia	4 Golf only		3,186,790.6		1,551.16	1,827,033.78
		Beach Beds Pty Ltd.	15,788,255.05	4,137,155.17		293,119.01	5,330,905.74
		Black Stump Camping Supplies	1,262,948.49		462,817.16	7,719	2,464,224.32
		Blue Mountains Golfing Company		8,257,037.18		69	1,797,644.23

The report is displayed in IBM Cognos Viewer. When the Total Revenue by Country link was clicked, the report was run and the underlying relational data source was queried for data. This query is needed because there was no saved output for the report. If the report had been previously run and saved, the saved version, by default, would appear in IBM Cognos Viewer. The report would display a snapshot of the data at the time it was last run and in the format specified by the person who ran the report. Formats include HTML, PDF, Excel, and XML.

3. In the top right corner, click **Return** ↩.

You will now run a report based on a dimensional package.

4. Click the **Public Folders** link shown below:



IBM Cognos Connection

Public Folders | My Folders

Public Folders > Samples > Models > GO Data Warehouse (query) > Report Studio Report Samples

Name

Bursted Sales Performance Report

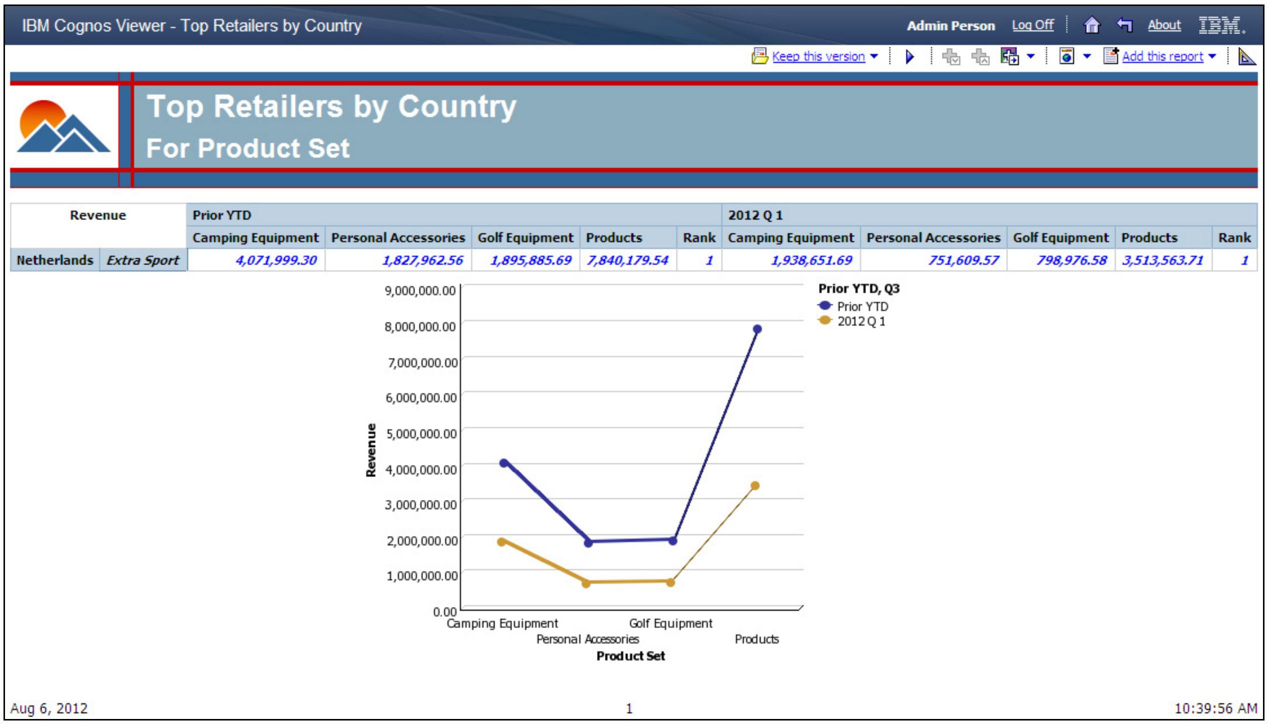
Using the path illustrated here, you can easily navigate the content found in IBM Cognos Connection.

5. Under **Name**, click **Samples_PowerCube** > **Cubes** > **Sales and Marketing (cube)** > **Report Studio Report Samples**.

6. Click **Top Retailers by Country**.

7. In the prompt, click **Extra Sport**, and then click **Finish**.

   The results appear as follows:



Top Retailers by Country
For Product Set

Revenue		Prior YTD					2012 Q 1				
		Camping Equipment	Personal Accessories	Golf Equipment	Products	Rank	Camping Equipment	Personal Accessories	Golf Equipment	Products	Rank
Netherlands	*Extra Sport*	*4,071,999.30*	*1,827,962.56*	*1,895,885.69*	*7,840,179.54*	*1*	*1,938,651.69*	*751,609.57*	*798,976.58*	*3,513,563.71*	*1*

Aug 6, 2012 1 10:39:56 AM

   Again, the report is displayed in IBM Cognos Viewer. This report is based on a package which uses a data source that connects to an IBM Cognos PowerCube. Because the PowerCube is an OLAP dimensional data source, the report can include dimensional functions to perform common OLAP-style queries that compare revenue from selected top retailers for product sets across various time periods.

   Reports and their underlying packages can also be created based on IBM Cognos Dynamic Cubes dimensional data sources. With IBM Cognos Dynamic Cubes, you can create, edit, import, export, and deploy virtual cube models over a relational warehouse schema.

8. Click **Return**.

Task 4. Create a workspace in IBM Cognos Insight.

1. In **Windows Explorer**, navigate to **<IBM Cognos install drive>:\Program Files\IBM\cognos\c10\webcontent\samples\datasources\other\profitability**, and then double-click **profitability.txt**.

   The file opens in Notepad.

   This is a tab-delimited file containing seven columns. The first five columns will be imported as dimensions. The final two columns will be imported as measures.

2. Close the **Profitability.txt** file.
3. From the **Start** menu, click **All Programs** > **IBM Cognos Insight** > **IBM Cognos Insight**.

   IBM Cognos Insight opens to the New Workspace page. Here you can create a new blank workspace, open an existing workspace, or import data into a new workspace.

4. Drag the **Profitability.txt** file from **Windows Explorer** to the **Drag and drop your files** area of **Getting Started** page.

   You can also click Get Data > Quick Import and then browse to select the Profitability.txt file.

   Your data is imported and ready to use.

5. Maximize **IBM Cognos Insight**.

   Notice that by default, the measures dimension is the column dimension.

6. In the top right corner, click **Use the content pane to explore and restructure data** ⊟.

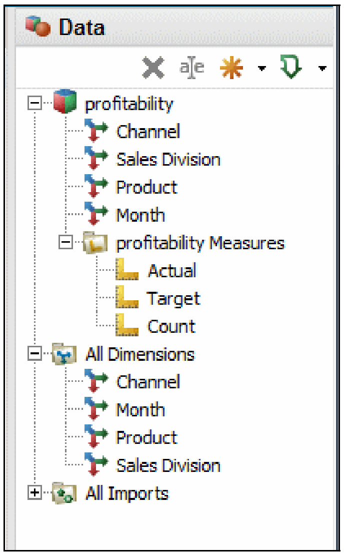
   The Content pane appears containing a cube named after the .txt file, and a folder containing all dimensions in the workspace.

7. Expand the **Profitability** cube and the **All Dimensions** folder.

   The cube contains an extra folder not found in the All Dimensions folder, which contains measures.

8. Expand the **Profitability Measures** folder.

   The results appear as follows:



There are three measures. Actual and Target were generated from the last two columns in the source file that contained numeric data. The count measure is generated automatically when you use the drag and drop or Quick Import methods to import data. You cannot disable this using these methods, but you can delete the Count measure after import.

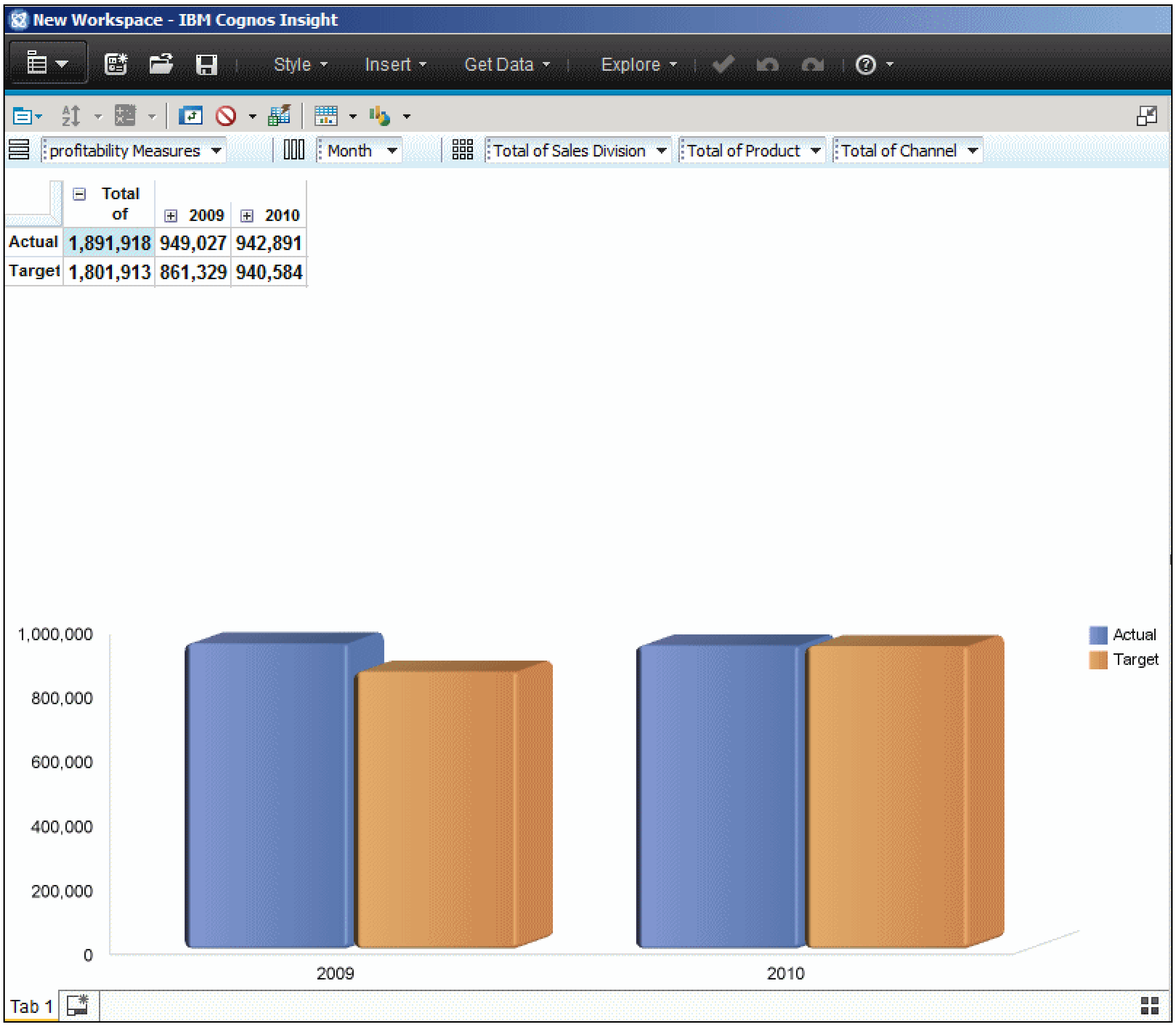
The Count measure is also generated using the Import Data method. Generate a count measure to confirm that your data was imported. The Count column in your crosstab should show the number of rows that you imported. This number will give you a quick indication of any duplicate or missing rows. You can enable or disable the creation of the count measure in the Import Data wizard, by selecting the cube_name Measures dimension in the Target items pane, and selecting/deselecting the Generate count measure check box.

You will remove the Count measure from the cube.

9. In the **Content** pane, right-click the **Count** measure, and then click **Delete**.
10. Click **Yes**.

    Now you want to view the data from a different perspective.
11. In the **Overview** area at the top of the crosstab, drag the **Total of Month** dimension on top of the **Profitability Measures:Actual** dimension on columns.
12. From the **Content** pane, drag the **Profitability Measures:Actual** dimension to the **Overview** area, and drop it on top of the **Channel** dimension on rows.
13. In the columns, collapse **2009** and **2010**.

    The results appear as follows:



	Total of	2009	2010
Actual	1,891,918	949,027	942,891
Target	1,801,913	861,329	940,584

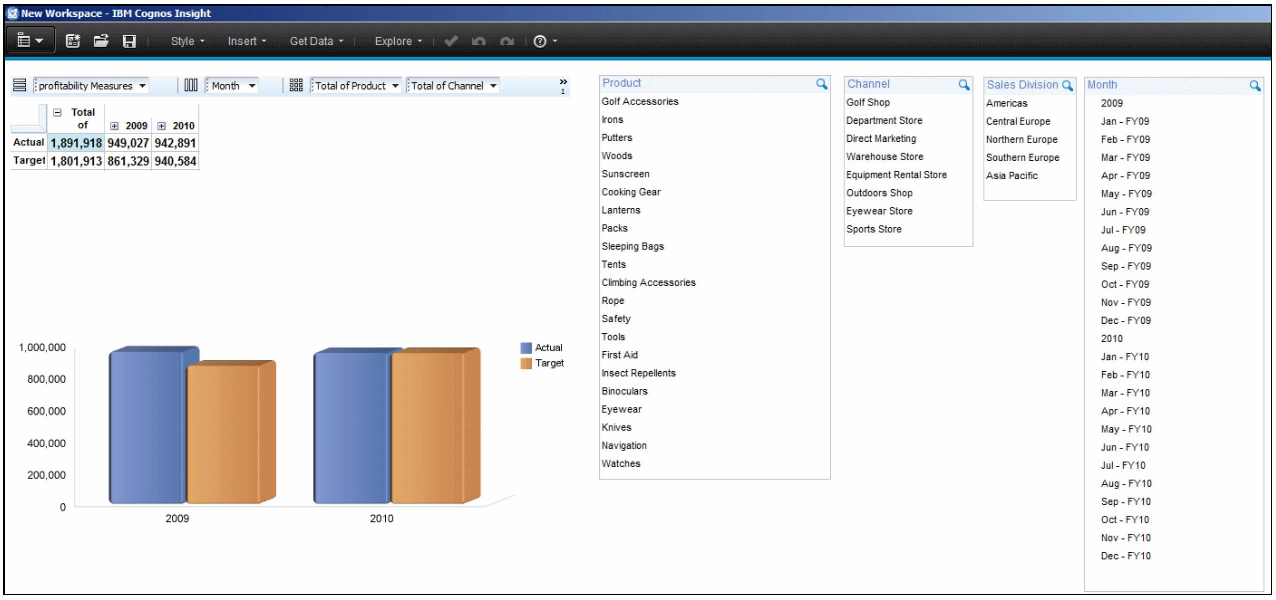
Notice that during import a time-based hierarchy was created automatically based on the relationship identified between the Fiscal Year and Month data in the source file. You can modify this behavior when using the Import Data method.

To aid in your analysis, you will now create Explore Points by dragging dimensions to the workspace. Explore Points are widgets that list all members from a specific dimension. They can be used to display filtered data to view only relevant information and can assist in discovery of associations between members of different dimensions. They are especially useful when viewing sparse data.

First you will resize the crosstab and chart widget so there is room on the workspace.

14. Click **Restore this widget** in the upper right corner of the workspace.
15. Click the widget toolbar, drag it to the upper left corner of the workspace, and then resize the widget so that all dimensions appear in the **Overview** area.
16. From the **Content** pane, drag the **Product** dimension to the workspace to create a **Product** explore point.
17. Resize the **Product** explore point to show all **Products**.
18. Repeat the previous two steps for the:
    - **Channel** dimension
    - **Sales Division** dimension
    - **Month** dimension

The results appear as follows:



	Total of	2009	2010
Actual	1,891,918	949,027	942,891
Target	1,801,913	861,329	940,584

You will now use the Explore Points to filter the data.

19. In the **Product** explore point, click **Binoculars**.

    Notice that the crosstab and chart change to show only Binocular sales. You can see that the Product dimension in the Overview area is filtering on Binoculars. Also notice in the Channel Explore Point that Golf Shop and Equipment Rental are grayed out, indicating that binoculars are not sold through these channels.

20. At the top of the **Product** explore point, click **Clear this explore point** so that no members are selected.

    The crosstab and chart once again change to reflect totals for all products.

21. In the **Channel** explore point, click **Golf Shop**.

    Notice that the crosstab and chart change to show only Golf Shop sales. Also notice in the Product explore point, that all of the products are grayed out except for the five products that are sold in the Golf Shop Channel.

22. Clear the **Channel** explore point, and then in the **Sales Division** explore point, click **Southern Europe**.

    Notice that the crosstab and chart change to show only Southern Europe sales. As well, in the Channel explore point, all of the channels are grayed out, except for the four channels that operate in Southern Europe.

23. In the **Sales Division** explore point, click **Central Europe**.

    When you select another member in the same Explore Point, the first member is no longer selected. You want to view data for Southern and Central Europe.

24. In the **Sales Division** explore point, Ctrl+click **Southern Europe**.

    The crosstab and chart now display data for two sales divisions.

25. Clear the **Sales Division** explore point.

    You can also search for members in each of the Explore Points.

26. In the **Product** explore point, click **Search**, type **Woods**, and then select it in the explore point.

    Notice that the crosstab and chart change to show only Woods sales, and the other explore points are adjusted to reflect only Woods.

27. Clear the **Product** explore point.

    Now you will add an Explore point for measures. You have the option of displaying only those measures you wish to see in the crosstab and chart.

28. From the **Content** pane, drag the **Profitability Measures** dimension to the workspace to create a **Profitability Measures** explore point.

29. In the **Profitability Measures** explore point, click **Actual**.

    The crosstab and chart change to show only Actual sales.

30. In the top left corner, on the menu bar, click the **Actions** menu, and then click **Save As**.
31. Name the file **Demo 1**, and then save the file to your desktop.

    The .cdd file contains both the layout and the data. This is a local copy of the workspace. Workspaces that you create in Cognos Insight can be shared or published to IBM Cognos TM1 or IBM Cognos Business Intelligence for others to work with.

    The following list provides an overview of the ways that you can share your workspaces with others. These options are available from the Actions menu in Cognos Insight.

    - Share - sharing a workspace creates a copy of your workspace in IBM Cognos Connection for other users to download and use on their computers. Shared workspaces in Cognos Connection can be launched from there if users have installed IBM Cognos Insight.

    - Publish - publishing a workspace copies the data in your workspace to the Cognos TM1 server and creates an application in the IBM Cognos Application portal. Other users who have access to the Cognos TM1 server can then view your data or open the workspace from the Cognos Application portal.

    - Publish and distribute - publishing and distributing a workspace publishes your data and workspace as defined in the previous definition, and Cognos Insight also creates a data source connection, a package, and reports in Cognos Connection. Other users on the Cognos TM1 server can access your data in Cognos TM1, work with your workspaces in Cognos Insight, and view the reports in IBM Cognos Workspace Advanced and IBM Cognos Workspace.

    In all of the above cases, you must be granted permissions by your administrator to add files to the appropriate servers.
32. Close **IBM Cognos Insight**.

Task 5. Author a report in Cognos Workspace Advanced using a dimensional source.

You will use Cognos Workspace Advanced to author a report using a dimensional source, include a calculation for analysis of the data, and customize your data by changing measures.

1. In **IBM Cognos Connection**, from the **Launch** menu, click **Cognos Workspace Advanced**.
2. Click **Create new**.
3. Beside the **package** box, click the **ellipsis**.
4. In the **List of all packages**, click **Samples**, click **Models**, and then click **GO Data Warehouse (analysis)**.

   The package on which you have chosen to create your report, is based on a dimensionally modeled relational (DMR) model. This means that the model developer has provided dimensional information to a relational model to allow authors and analysts to perform OLAP-style queries at run time on a relational data source, including the ability to drill up and drill down in the data.
5. Click **OK**.
6. Double-click **Crosstab** to open a new crosstab report.

   Cognos Workspace Advanced opens, with a report layout area on the left, and the data tree displayed on the Source tab in the pane on the right. By default, the tree displays members.
7. On the **Source** tab, expand the **Sales and Marketing (analysis)** folder, and then expand the **Sales** folder.
8. Expand the **Time** hierarchy .

   There are 4 members displayed in Time: 2010, 2011, 2012, and 2013.
9. On the toolbar, above the **Source** tab, click **View Metadata Tree** .

10. On the **Source** tab, expand the **Sales and Marketing (analysis)** folder, expand the **Sales** namespace, and then expand the **Time** dimension.

    The Time hierarchy is displayed. You are now dealing with metadata objects such as dimensions, hierarchies, levels, and facts. This is different from step 5, where only hierarchies and members were displayed.

11. Expand the **Time** hierarchy.

    Under the Time hierarchy, the Members folder is displayed, along with levels such as Year and Quarter.

12. Expand the **Members** folder, and the **Time** member.

    Within the Members folder is the Time member, and the members 2010, 2011, 2012, and 2013.

    With dimensional data, you can quickly select the view that displays the elements you want to work with by using the View Members Tree button or the View Metadata Tree button.

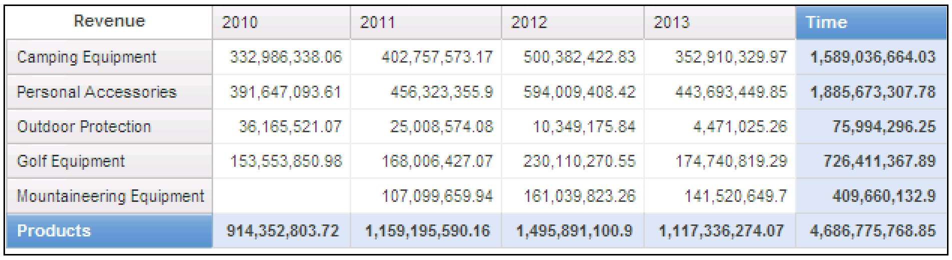
    Now you will enable the report for drilling and populate the report layout area. Enabling drill capability is only required for consumers when they run the report and it is rendered in IBM Cognos Viewer. Drill capability is available by default when viewing live data in Cognos Workspace Advanced.

13. From the **Data** menu, click **Drill Options**, select the **Allow drill-up and drill-down** check box, and then click **OK**.

14. On the toolbar above the **Source** tab, click **View Members Tree**.

15. On the **Source** tab, from **Sales and Marketing (analysis)** > **Sales**, add the following items to the report by dragging each to the report layout:
    - **Columns** drop zone: **Time** hierarchy
    - **Rows** drop zone: **Products** hierarchy
    - **Measures** drop zone: **Sales fact** > **Revenue** measure

    The results appear as follows:

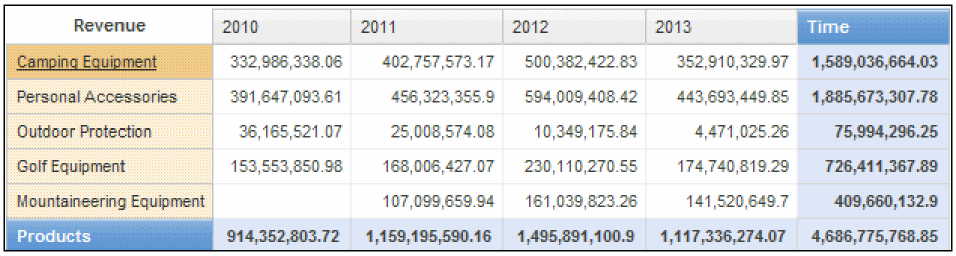


Revenue	2010	2011	2012	2013	Time
Camping Equipment	332,986,338.06	402,757,573.17	500,382,422.83	352,910,329.97	**1,589,036,664.03**
Personal Accessories	391,647,093.61	456,323,355.9	594,009,408.42	443,693,449.85	**1,885,673,307.78**
Outdoor Protection	36,165,521.07	25,008,574.08	10,349,175.84	4,471,025.26	**75,994,296.25**
Golf Equipment	153,553,850.98	168,006,427.07	230,110,270.55	174,740,819.29	**726,411,367.89**
Mountaineering Equipment		107,099,659.94	161,039,823.26	141,520,649.7	**409,660,132.9**
Products	**914,352,803.72**	**1,159,195,590.16**	**1,495,891,100.9**	**1,117,336,274.07**	**4,686,775,768.85**

    The live data values automatically populate the crosstab as you add each element, and there is no need to run the report separately as you develop it.

16. Click **Camping Equipment**.

    The results appear as follows:

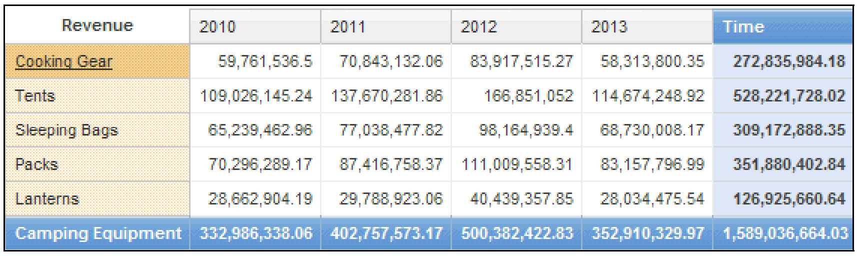


Revenue	2010	2011	2012	2013	Time
Camping Equipment	332,986,338.06	402,757,573.17	500,382,422.83	352,910,329.97	**1,589,036,664.03**
Personal Accessories	391,647,093.61	456,323,355.9	594,009,408.42	443,693,449.85	**1,885,673,307.78**
Outdoor Protection	36,165,521.07	25,008,574.08	10,349,175.84	4,471,025.26	**75,994,296.25**
Golf Equipment	153,553,850.98	168,006,427.07	230,110,270.55	174,740,819.29	**726,411,367.89**
Mountaineering Equipment		107,099,659.94	161,039,823.26	141,520,649.7	**409,660,132.9**
Products	**914,352,803.72**	**1,159,195,590.16**	**1,495,891,100.9**	**1,117,336,274.07**	**4,686,775,768.85**

    Notice that Camping Equipment is now an underlined item, indicating that it is available for drill operations. You can drill down and up on these items to further analyze your data. The same type of behavior applies to OLAP data sources.

17. Click **Camping Equipment** again, and then click **Cooking Gear**.

    The results appear as follows:



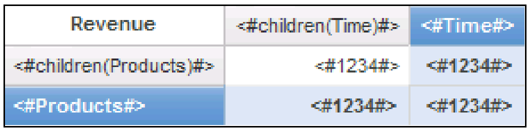
Revenue	2010	2011	2012	2013	Time
Cooking Gear	59,761,536.5	70,843,132.06	83,917,515.27	58,313,800.35	272,835,984.18
Tents	109,026,145.24	137,670,281.86	166,851,052	114,674,248.92	528,221,728.02
Sleeping Bags	65,239,462.96	77,038,477.82	98,164,939.4	68,730,008.17	309,172,888.35
Packs	70,296,289.17	87,416,758.37	111,009,558.31	83,157,796.99	351,880,402.84
Lanterns	28,662,904.19	29,788,923.06	40,439,357.85	28,034,475.54	126,925,660.64
Camping Equipment	332,986,338.06	402,757,573.17	500,382,422.83	352,910,329.97	1,589,036,664.03

    You have now drilled down to a lower level of detail. In this case you are viewing all the product types within the Camping Equipment product line.

    Notice that Cooking Gear is an underlined item. At this point, you could drill down on Cooking Gear to view all the products within the Cooking Gear product type, but instead you will drill back up to the product line level.

18. Right-click **Cooking Gear**, point to **Explore**, and then click **Drill Up**.
19. From the **View** menu, click **Page Design**.

    The results appear as follows:

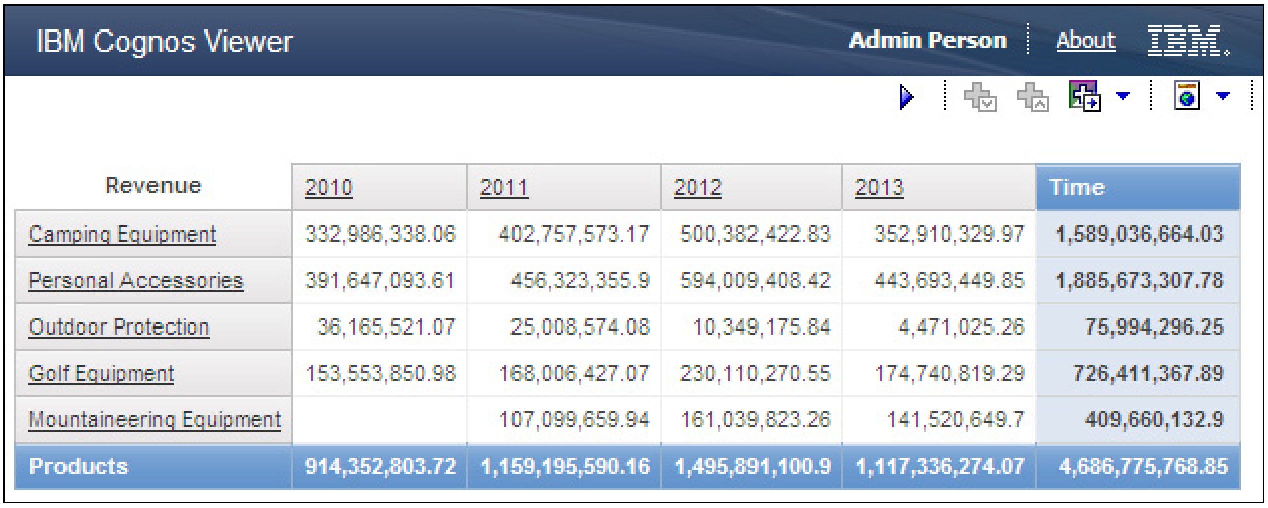


Revenue	<#children(Time)#>	<#Time#>
<#children(Products)#>	<#1234#>	<#1234#>
<#Products#>	<#1234#>	<#1234#>

    The crosstab is now filled with placeholders, instead of the live data. You can see the set of members of Time and Products, in the columns and rows respectively, and the Revenue measure placeholders. To see the data values when working in Page Design mode, you will run the report.

20. On the toolbar, click **Run Report** ▶.

    The results appear as follows:



Revenue	2010	2011	2012	2013	Time
Camping Equipment	332,986,338.06	402,757,573.17	500,382,422.83	352,910,329.97	1,589,036,664.03
Personal Accessories	391,647,093.61	456,323,355.9	594,009,408.42	443,693,449.85	1,885,673,307.78
Outdoor Protection	36,165,521.07	25,008,574.08	10,349,175.84	4,471,025.26	75,994,296.25
Golf Equipment	153,553,850.98	168,006,427.07	230,110,270.55	174,740,819.29	726,411,367.89
Mountaineering Equipment		107,099,659.94	161,039,823.26	141,520,649.7	409,660,132.9
Products	914,352,803.72	1,159,195,590.16	1,495,891,100.9	1,117,336,274.07	4,686,775,768.85

The report opens in IBM Cognos Viewer displaying data values. Notice that all the column and row items in the report are underlined. Consumers can drill down and up on these items to further analyze data.

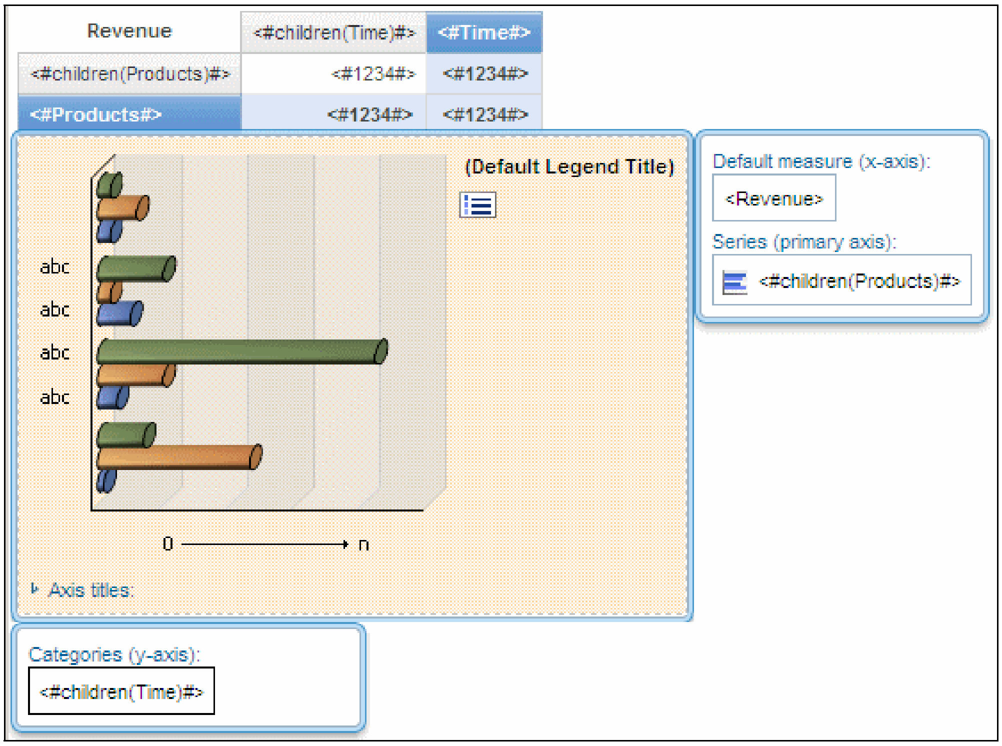
21. Close **IBM Cognos Viewer**.

    You will now add a chart to this report.

22. Click a cell in the crosstab.
23. In the right pane, click the **Toolbox** tab, and then drag a **Chart** object below the crosstab.
24. In the left pane click **Bar**, and then in the right pane click **Clustered Cylinder Bar with 3-D Effects**.

25. Select the **Fill with data** check box, and then click **OK**.

    The bar chart is displayed below the crosstab. The chart is automatically populated with the data items (as placeholders) from the crosstab.



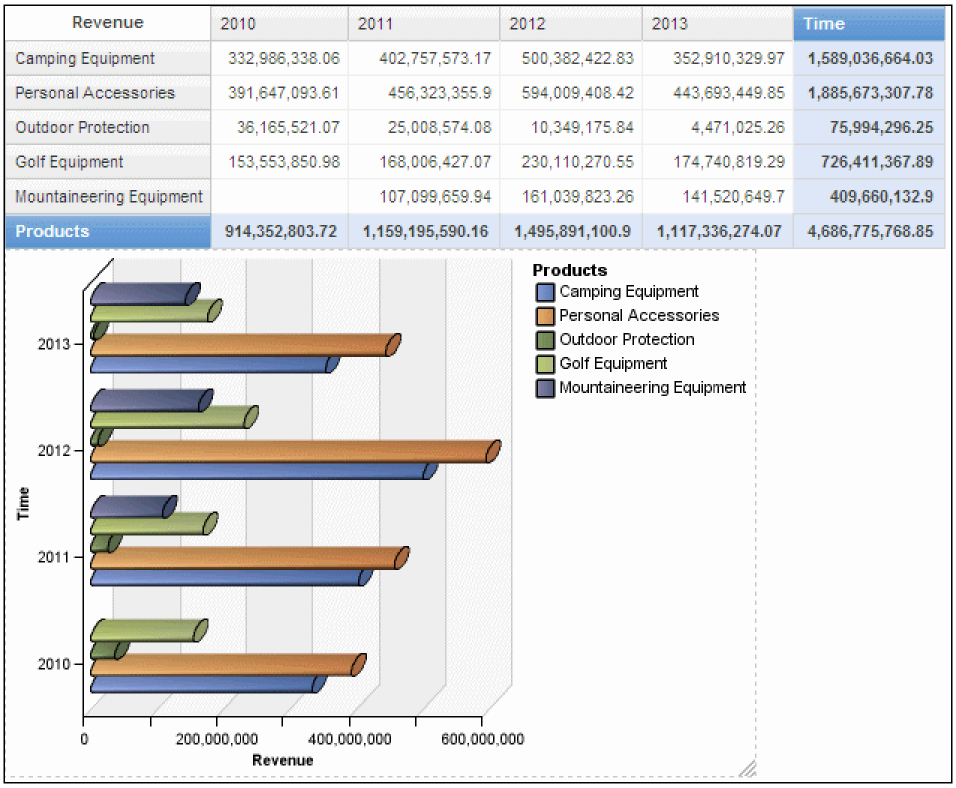
Revenue	<#children(Time)#>	<#Time#>
<#children(Products)#>	<#1234#>	<#1234#>
<#Products#>	<#1234#>	<#1234#>

26. From the **View** menu, click **Page Preview**.

    The crosstab and chart display actual data values.

27. Click the title text to change the focus from the chart.

    The results appear as follows:



Revenue	2010	2011	2012	2013	Time
Camping Equipment	332,986,338.06	402,757,573.17	500,382,422.83	352,910,329.97	**1,589,036,664.03**
Personal Accessories	391,647,093.61	456,323,355.9	594,009,408.42	443,693,449.85	**1,885,673,307.78**
Outdoor Protection	36,165,521.07	25,008,574.08	10,349,175.84	4,471,025.26	**75,994,296.25**
Golf Equipment	153,553,850.98	168,006,427.07	230,110,270.55	174,740,819.29	**726,411,367.89**
Mountaineering Equipment		107,099,659.94	161,039,823.26	141,520,649.7	**409,660,132.9**
Products	**914,352,803.72**	**1,159,195,590.16**	**1,495,891,100.9**	**1,117,336,274.07**	**4,686,775,768.85**

28. In the lower right corner of the chart, drag the resize corner to enlarge the chart to be the same width as the crosstab.

    Charts can easily be resized using this method.

    Now that you have reviewed the revenue for the product lines over the years, you want to change the measure to display quantity.

29. Click the **Source** tab, and then from **Sales fact**, drag **Quantity** to the measures area on the crosstab.

    The crosstab displays the new measure.

30. Right-click the chart, and then click **Update Chart from Crosstab**.

    The chart updates to display the quantity values based on the crosstab.

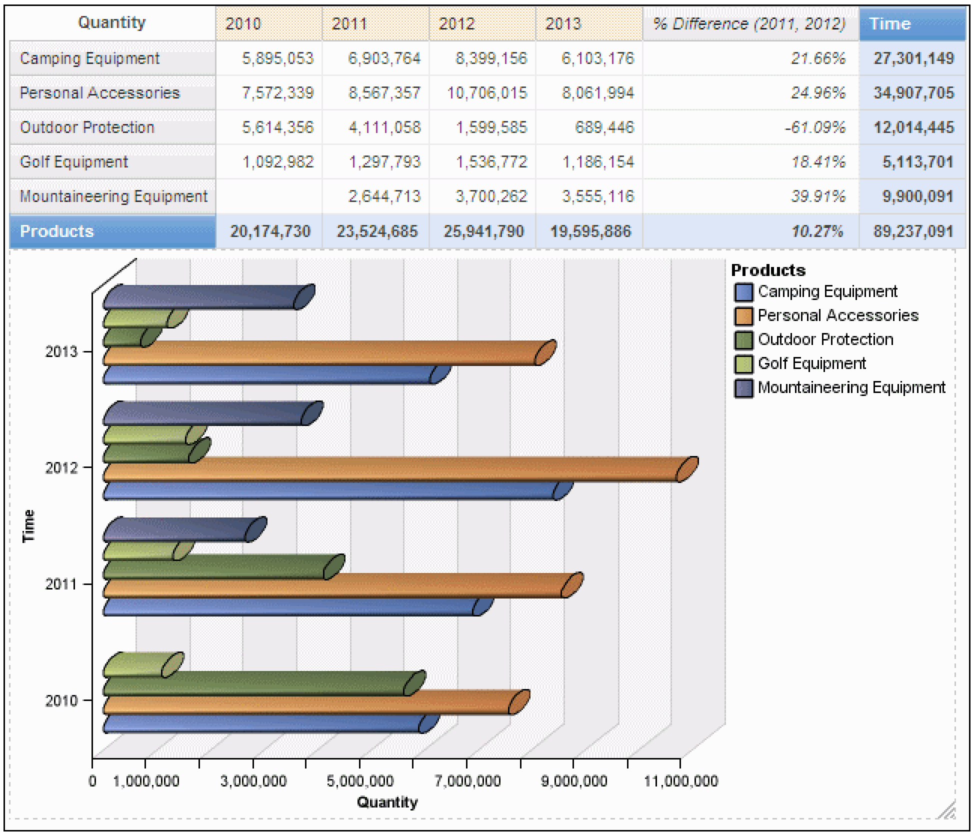
    You want to analyze the previous year-to-year percent difference of quantity for 2011 and 2012. To do this, you will create a variance calculation using a preset calculation in Cognos Workspace Advanced.

31. In the crosstab, click the **2011** column header, and then Shift-click the **2012** column header.

32. Right-click **2012**, point to **Calculate**, and then click **% Difference (2011, 2012)**.

33. Resize the chart if necessary.

    The results appear as follows:



Quantity	2010	2011	2012	2013	% Difference (2011, 2012)	Time
Camping Equipment	5,895,053	6,903,764	8,399,156	6,103,176	21.66%	27,301,149
Personal Accessories	7,572,339	8,567,357	10,706,015	8,061,994	24.96%	34,907,705
Outdoor Protection	5,614,356	4,111,058	1,599,585	689,446	-61.09%	12,014,445
Golf Equipment	1,092,982	1,297,793	1,536,772	1,186,154	18.41%	5,113,701
Mountaineering Equipment		2,644,713	3,700,262	3,555,116	39.91%	9,900,091
Products	20,174,730	23,524,685	25,941,790	19,595,886	10.27%	89,237,091

A new column displays the calculated values for you to review.

Task 6. Author a report in Cognos Workspace Advanced using a relational source.

You will now use Cognos Workspace Advanced to author a List report that is based on a relational source.

1. On the toolbar, click **New**, and then click **No** to saving the report.
2. Beside the **Package** box, click the ellipsis, and then under **Public Folders**, expand **Samples** > **Models**.

3. Click **GO Data Warehouse (query)**, and then click **OK**.

   This is a package based on a relational model. OLAP-style queries are not possible with this package.

4. Click **List**, and then click **OK**.

   Cognos Workspace Advanced opens, with a layout area for a List on the left, and the data tree displayed on the Source tab in the pane on the right. Notice that above the Source tab, you no longer have the option to toggle between viewing members and metadata. When working with relational sources, your only option is to work with metadata.

   For the List report you are creating, you will be grouping some of the columns to remove duplicate items within them and make the report easier to read. Along with your grouped columns, you want to include a summary of the data for each level of grouping. When working with a grouped List report, the default behavior in IBM Cognos Workspace is such that a summary is not automatically generated after grouping is applied.

   If you want to create a grouped List report, and have summaries generated automatically when the grouping is applied, you must configure an environment property in IBM Cognos Workspace Advanced. Once applied, every time you create a grouped List report, summaries will be automatically generated. This property has no impact on List reports that were created prior to the property being set. For these reports, you must add summaries manually.

5. From the **Tools** menu, click **Options**.

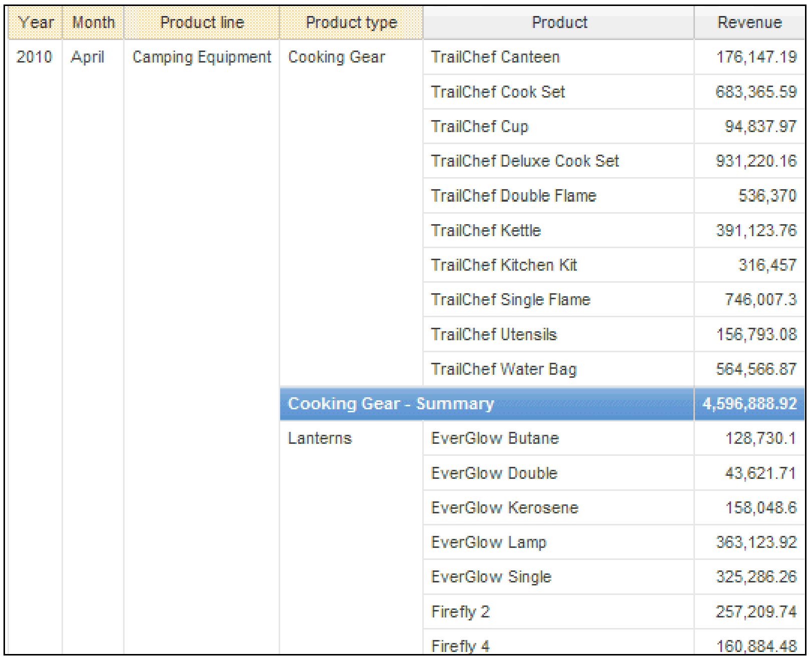
6. Click the **Report** tab, select the **Automatic group and summary behavior for lists** check box, and then click **OK**.

7. On the **Source** tab, expand the **Sales and Marketing (query)** folder, the **Sales (query)** namespace, and the **Time** query subject.

   You are now working with namespaces, query subjects, and query items.

8. Add the following query items to the report by double-clicking them:
   - **Time** query subject: **Year** query item; **Month** query item
   - **Products** query subject: **Product line** query item; **Product type** query item; **Product** query item
   - **Sales fact** query subject: **Revenue** query item
9. Click the **Year** column header, and then Ctrl-click **Month**, **Product line**, and **Product type**.
10. On the toolbar, click **Group**.

    The results appear as follows:

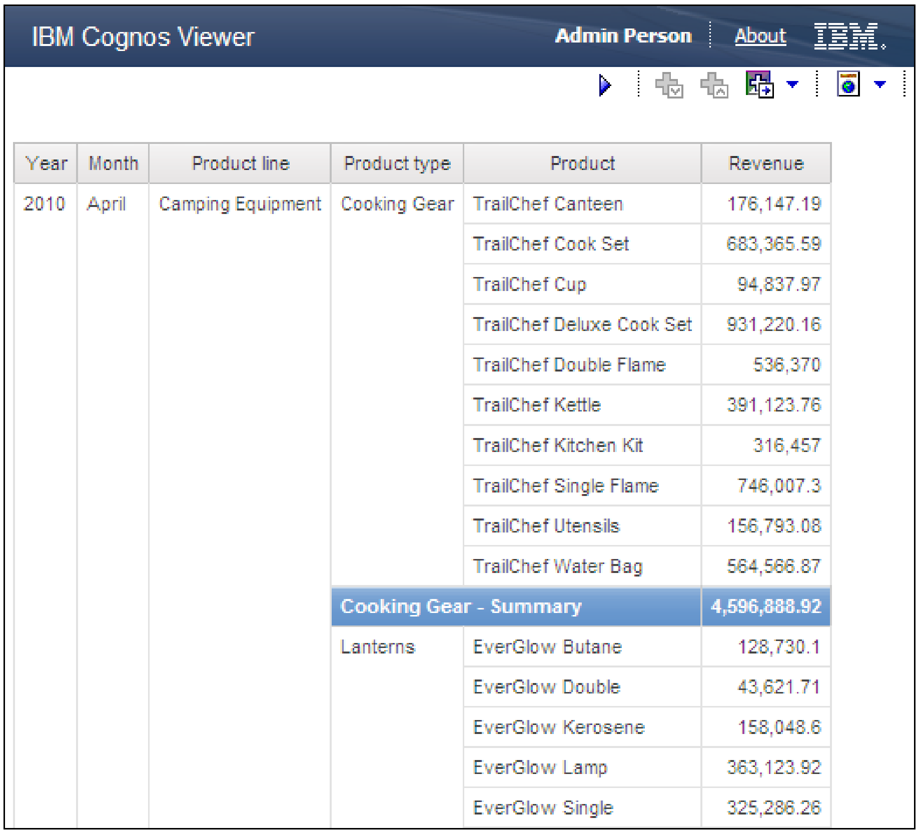


Year	Month	Product line	Product type	Product	Revenue
2010	April	Camping Equipment	Cooking Gear	TrailChef Canteen	176,147.19
				TrailChef Cook Set	683,365.59
				TrailChef Cup	94,837.97
				TrailChef Deluxe Cook Set	931,220.16
				TrailChef Double Flame	536,370
				TrailChef Kettle	391,123.76
				TrailChef Kitchen Kit	316,457
				TrailChef Single Flame	746,007.3
				TrailChef Utensils	156,793.08
				TrailChef Water Bag	564,566.87
			Cooking Gear - Summary		4,596,888.92
			Lanterns	EverGlow Butane	128,730.1
				EverGlow Double	43,621.71
				EverGlow Kerosene	158,048.6
				EverGlow Lamp	363,123.92
				EverGlow Single	325,286.26
				Firefly 2	257,209.74
				Firefly 4	160,884.48

This is a basic list report with some grouping applied to make the report easier to read. Summaries have been automatically generated at each level of grouping.

11. On the toolbar, click **Run Report**.

    The results appear as follows:



Year	Month	Product line	Product type	Product	Revenue
2010	April	Camping Equipment	Cooking Gear	TrailChef Canteen	176,147.19
				TrailChef Cook Set	683,365.59
				TrailChef Cup	94,837.97
				TrailChef Deluxe Cook Set	931,220.16
				TrailChef Double Flame	536,370
				TrailChef Kettle	391,123.76
				TrailChef Kitchen Kit	316,457
				TrailChef Single Flame	746,007.3
				TrailChef Utensils	156,793.08
				TrailChef Water Bag	564,566.87
			Cooking Gear - Summary		4,596,888.92
			Lanterns	EverGlow Butane	128,730.1
				EverGlow Double	43,621.71
				EverGlow Kerosene	158,048.6
				EverGlow Lamp	363,123.92
				EverGlow Single	325,286.26

Notice that this report does not contain underlined items. You cannot drill down on these items to further analyze your data. This is because you are working with a relational source that has no dimensional information applied to it.

You will now save the report and open it in Report Studio to add a corporate graphic and a prompt page.

12. Close **IBM Cognos Viewer**, and then on the toolbar, click **Save** .

    By default the report is saved in the folder of the package on which the report is based. You can save your report to any location that you have access to. You can also copy/cut and paste your report and other objects to organize content in the portal.

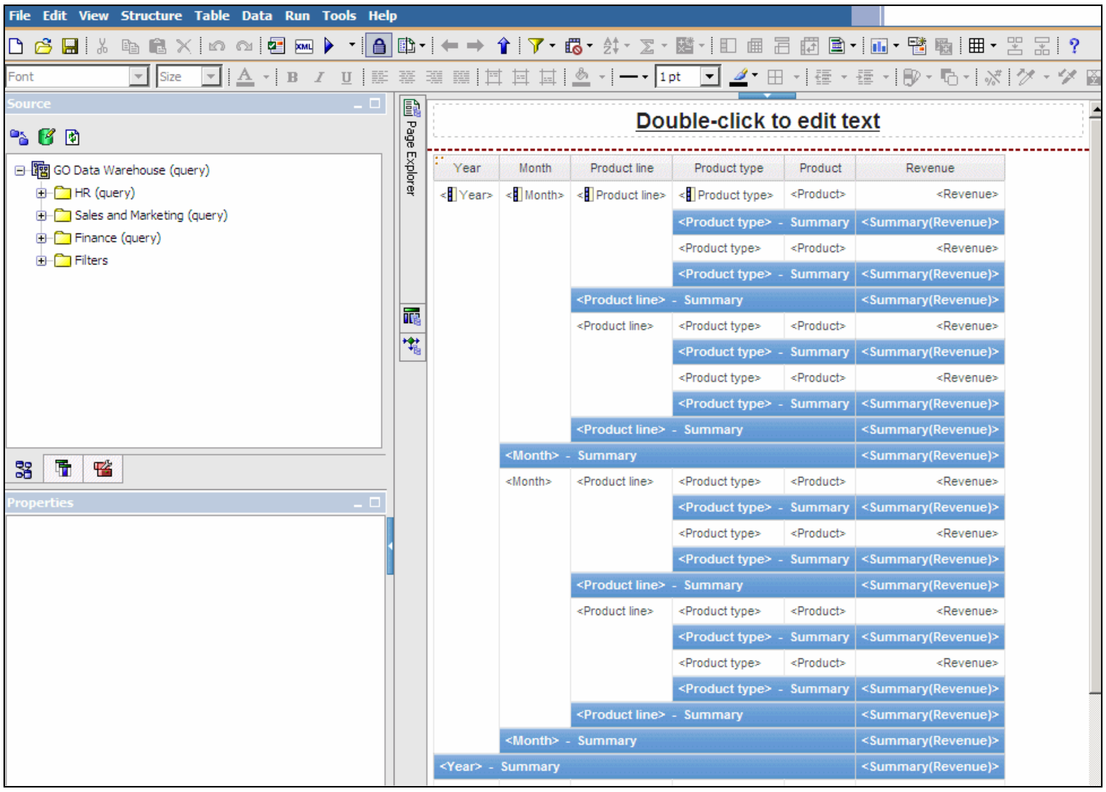
13. Under **Save in**, click **My Folders**, and then in the **Name** box, type **Product Revenue**.
14. Click **Save**, and then close **IBM Cognos Workspace Advanced**.

    You will now open this report in Report Studio and continue to modify it. Because IBM Cognos Workspace and IBM Cognos Report Studio share the same report specification, reports can be authored in one studio and opened in the other.

Task 7. Enhance a report in Report Studio

1. In **IBM Cognos Connection**, click the **My Folders** tab, and then beside the **Product Revenue** report, under the **Actions** column, click **Open with Report Studio – Product Revenue**.

    The results appear as follows:

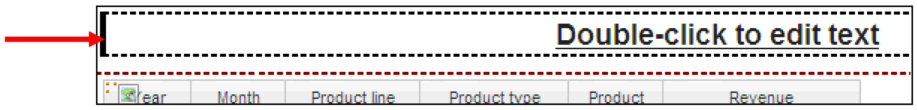

File Edit View Structure Table Data Run Tools Help
Font
Size
1 pt
Source
GO Data Warehouse (query)
HR (query)
Sales and Marketing (query)
Finance (query)
Filters
Properties
Page Explorer
Double-click to edit text

Year	Month	Product line	Product type	Product	Revenue
<Year>	<Month>	<Product line>	<Product type>	<Product>	<Revenue>
			<Product type> -	Summary	<Summary(Revenue)>
			<Product type>	<Product>	<Revenue>
			<Product type> -	Summary	<Summary(Revenue)>
		<Product line> -	Summary		<Summary(Revenue)>
		<Product line>	<Product type>	<Product>	<Revenue>
			<Product type> -	Summary	<Summary(Revenue)>
			<Product type>	<Product>	<Revenue>
			<Product type> -	Summary	<Summary(Revenue)>
		<Product line> -	Summary		<Summary(Revenue)>
	<Month> -	Summary			<Summary(Revenue)>
	<Month>	<Product line>	<Product type>	<Product>	<Revenue>
			<Product type> -	Summary	<Summary(Revenue)>
			<Product type>	<Product>	<Revenue>
			<Product type> -	Summary	<Summary(Revenue)>
		<Product line> -	Summary		<Summary(Revenue)>
		<Product line>	<Product type>	<Product>	<Revenue>
			<Product type> -	Summary	<Summary(Revenue)>
			<Product type>	<Product>	<Revenue>
			<Product type> -	Summary	<Summary(Revenue)>
		<Product line> -	Summary		<Summary(Revenue)>
	<Month> -	Summary			<Summary(Revenue)>
<Year> -	Summary				<Summary(Revenue)>

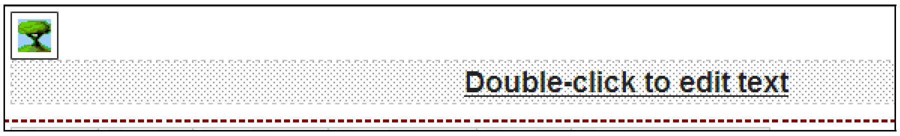
The report, which opens in Report Studio, can now be enhanced with features that are not available in Cognos Workspace Advanced. You will begin by adding a corporate graphic. Note: you are now in a design environment and if you wish to see the data returned by the report, you must run the report.

2. In the left pane, click the **Toolbox** tab.

3. Drag an **Image** object to the left of the **Block** object that contains the **Text Item** (that contains the text **Double-click to edit text**), as shown below:



The results appear as follows:

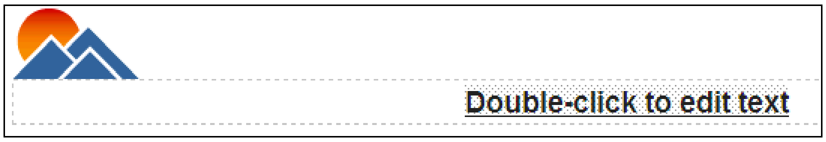


4. Right-click the **Image** object, and then click **Edit Image URL**.

5. Click **Browse**, in the list, select **logo_great_outdoors.gif**, and then click **OK**.

6. Click **OK**, and then click the **Text Item** object (that contains the text **Double-click to edit text**) to remove focus from the image object.

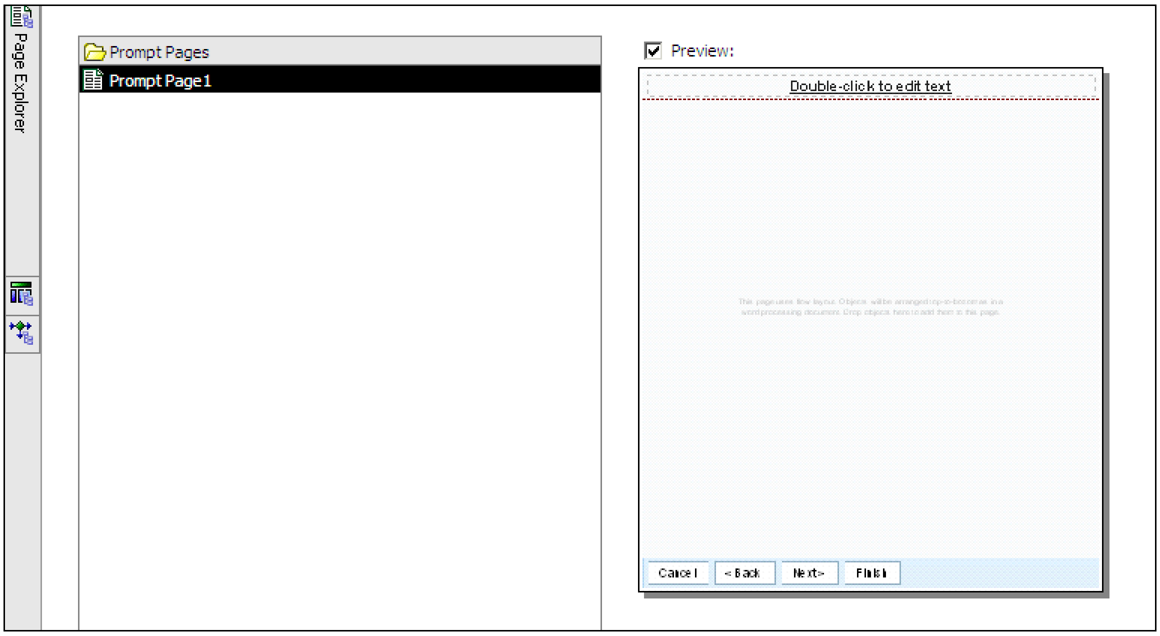
The results appear as follows:



You will now add a prompt page to your report to allow users to filter on a particular year of data. Prompting is also available in Cognos Workspace Advanced. However, in Report Studio you can create individual prompt pages, as well as additional report pages, to help you control the look, feel, and flow of the report.

7. In the middle pane, point to **Page Explorer**, and then click **Prompt Pages**.
8. From the **Toolbox** tab, drag a **Page** object to the **Prompt Pages** pane.

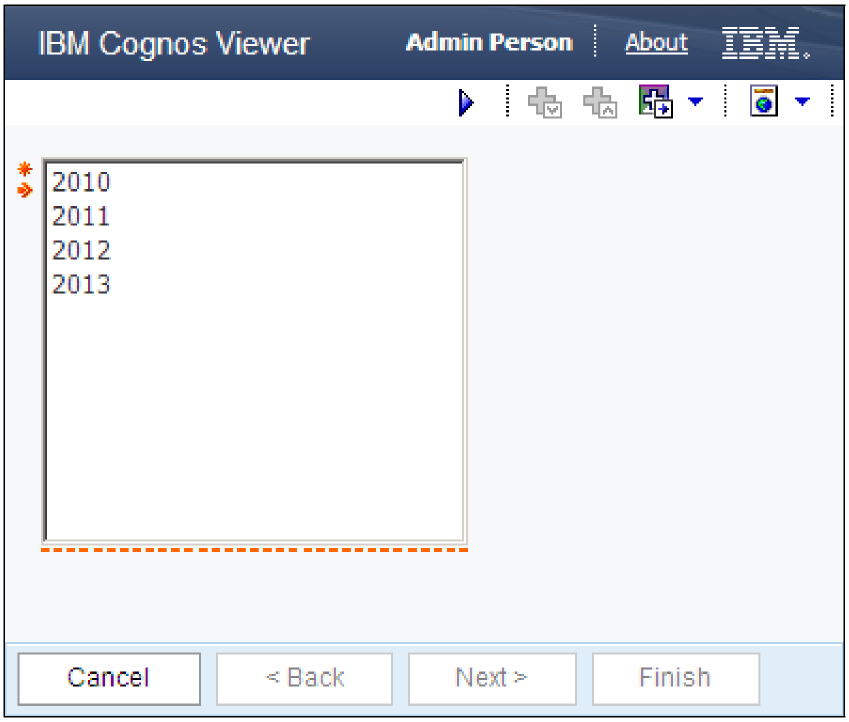
   The results appear as follows:



9. Point to **Page Explorer**, and then click **Prompt Page 1**.
10. In the left pane, click the **Toolbox** tab, and then drag a **Value Prompt** object to the prompt page layout.
11. In the **Create a new parameter** box, type **Year**, and then click **Next**.
12. Beside the **Package item** box, click the ellipsis, and then navigate to **Sales and Marketing (query)** > **Sales (query)** > **Time**.
13. Click **Year**, click **OK**, and then click **Next**.
14. Ensure that **Create a new query** is selected, and then click **Finish**.
15. Click the **Value Prompt** object, and then in the **Properties** pane, under **General**, click the **Multi-Select** property.

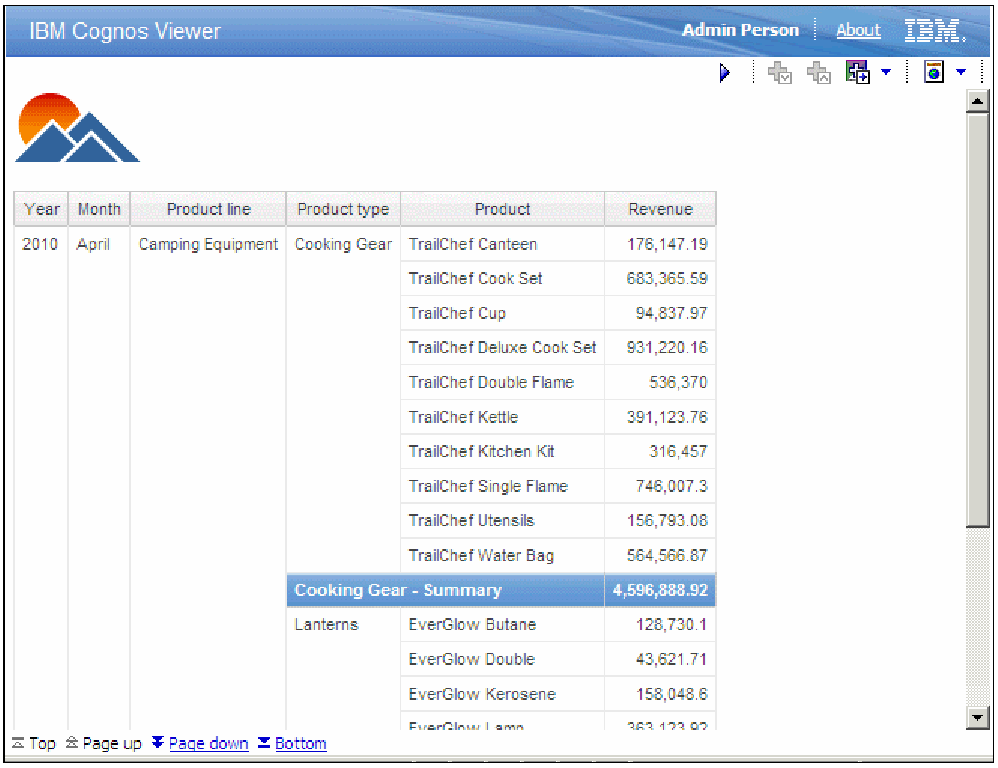
16. In the list, click **Yes**, and then on the toolbar, click **Run Report**.

    The result appears as follows:



17. Click **2010**, and then click **Finish**.

    The results appear as follows:



Year	Month	Product line	Product type	Product	Revenue
2010	April	Camping Equipment	Cooking Gear	TrailChef Canteen	176,147.19
				TrailChef Cook Set	683,365.59
				TrailChef Cup	94,837.97
				TrailChef Deluxe Cook Set	931,220.16
				TrailChef Double Flame	536,370
				TrailChef Kettle	391,123.76
				TrailChef Kitchen Kit	316,457
				TrailChef Single Flame	746,007.3
				TrailChef Utensils	156,793.08
				TrailChef Water Bag	564,566.87
			Cooking Gear - Summary		4,596,888.92
			Lanterns	EverGlow Butane	128,730.1
				EverGlow Double	43,621.71
				EverGlow Kerosene	158,048.6
				EverGlow Lamp	363,123.92

The report displays the image you inserted as well as data for the year 2010.

This is a simple example of enhancing a report with Report Studio. Report Studio is capable of many advanced authoring techniques to create professional, dynamic, and robust reports.

18. Close **IBM Cognos Viewer**, and then in **Report Studio**, from the **File** menu, click **Save As**.
19. Under **Save in**, click **My Folders**, and then in the **Name** box, type **Product Revenue: Select Year**.
20. Click **Save**, and then close **Report Studio**.

Task 8. Customize a workspace in Cognos Workspace.

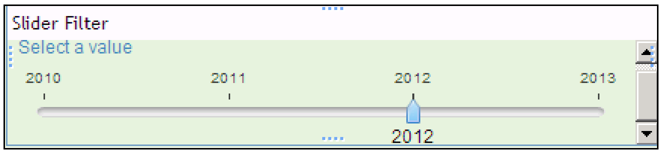
You will use IBM Cognos Workspace to customize and use a workspace. You will modify a filter, delete an element, add a report part, save the workspace, and add an annotation to the report data.

1. In **IBM Cognos Connection**, from the **Launch** menu in the top right corner, click **Cognos Workspace**.
2. On the **Getting Started Page - IBM Cognos Workspace**, click **Open Existing**, navigate to **Samples** > **Models** > **Cognos Workspace Samples** > **Marketing Workspace**, and then click **Open**.

   The workspace opens in Cognos Workspace.
3. From the **Actions** ▼ menu, click **Edit Workspace Style**.
4. Click the **Widgets** tab, select the **Show Titles** check box, and then click **OK**.

   The workspace contains a slider filter named Slider Filter, a select value filter named Campaign name, a select value filter named Product line, and four report widgets.
5. From the **Actions** menu, click **Save As**, and then save the workspace in **My Folders** as **My Marketing Workspace**.
6. In the **Slider Filter**, drag the slider to **2012**.

   The results appear as follows:



The Advertising cost by year report widget, which communicates with the Slider Filter, is refreshed to reflect the modified filter requirements.

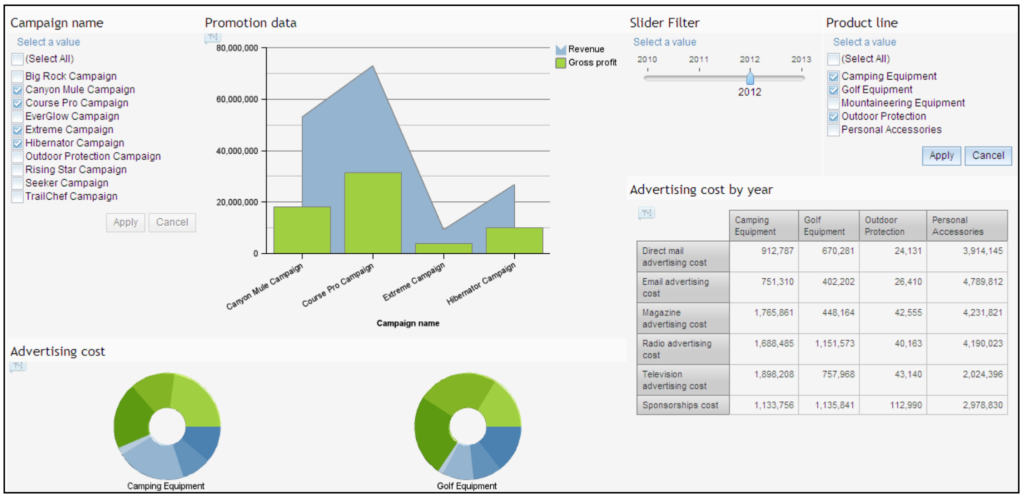
You want to delete one report widget to customize your workspace.

7. Click the **Promotion plan versus promotion revenue** report widget, and then from the widget **Actions** menu, click **Remove from Workspace**, and then click **Remove** to confirm.

   You want to change the layout of the workspace.

8. Click the **Year** slider filter and drag it near the top of the canvas, and then repeat with the **Product line** select value filter and the **Advertising cost by year** report widget.

   A section of the results appear as follows:



Campaign name

Select a value

- [ ] (Select All)
- [ ] Big Rock Campaign
- [✓] Canyon Mule Campaign
- [✓] Course Pro Campaign
- [ ] EverGlow Campaign
- [✓] Extreme Campaign
- [✓] Hibernator Campaign
- [ ] Outdoor Protection Campaign
- [ ] Rising Star Campaign
- [ ] Seeker Campaign
- [ ] TrailChef Campaign

Apply Cancel

Product line

Select a value

- [ ] (Select All)
- [✓] Camping Equipment
- [✓] Golf Equipment
- [ ] Mountaineering Equipment
- [✓] Outdoor Protection
- [ ] Personal Accessories

Apply Cancel

Advertising cost by year

	Camping Equipment	Golf Equipment	Outdoor Protection	Personal Accessories
Direct mail advertising cost	912,787	670,281	24,131	3,914,145
Email advertising cost	751,310	402,202	26,410	4,789,812
Magazine advertising cost	1,765,861	448,164	42,555	4,231,821
Radio advertising cost	1,688,485	1,151,573	40,163	4,190,023
Television advertising cost	1,898,208	757,968	43,140	2,024,396
Sponsorships cost	1,133,756	1,135,841	112,990	2,978,830

9. On the **Cognos Workspace** toolbar, click **Insert**, and then click **Insert Content**.

   On the right side, a content pane displays Content and Toolbox tabs.

10. On the **Content** tab, navigate through the tree by expanding **Public Folders** > **Samples** > **Models** > **GO Data Warehouse (query)** > **Report Studio Report Samples**.

    You can see the reports stored in the folder. Usually, a report would be the end of your navigation capability, however Cognos Workspace allows you to leverage report parts in existing reports.

11. Expand **Total Revenue by Country** > **Page 1**.

    The tree expands to display chart and crosstab report parts that exist in the current report. You want use one of these report parts in your workspace. Also notice that the report includes a prompt page

12. Drag **Crosstab11111** to the canvas, below the **Advertising cost by year** report widget.

13. In the prompt, click **Select all**, and then click **Finish**.

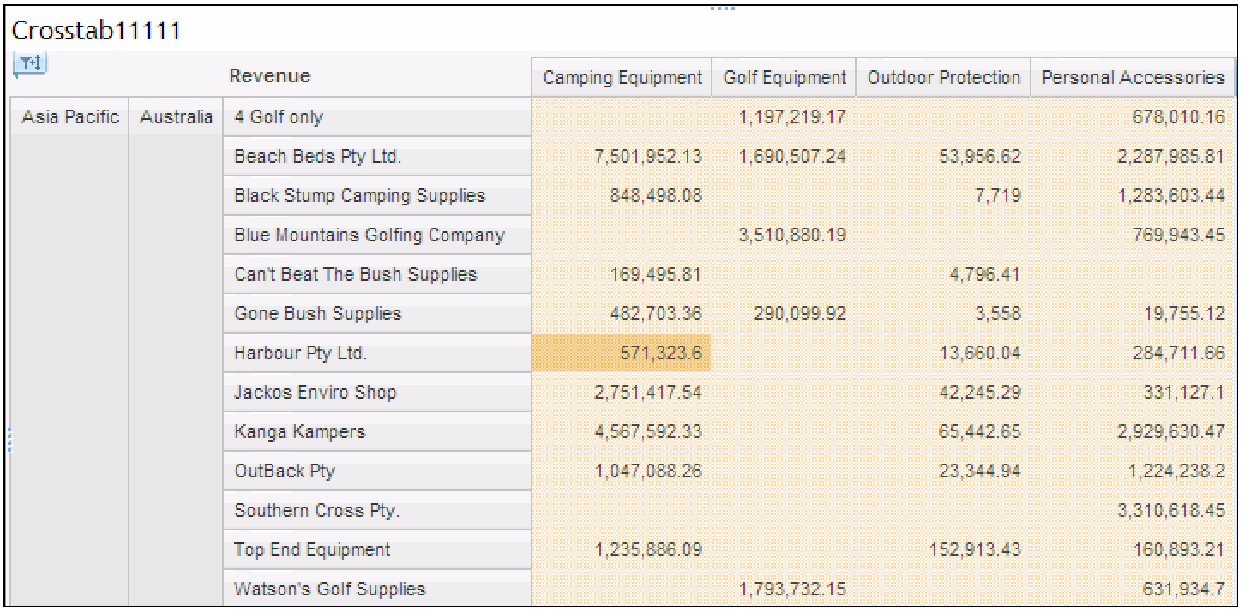
    A new report widget is placed on the canvas, and the crosstab is displayed.

    You want to add a comment about the data displayed in the chart. First, you will save the workspace.

14. From the **Cognos Workspace** toolbar, click **Save**.

15. In the new crosstab report widget, click the intersection of **Harbour Pty Ltd.** and **Camping Equipment**.

    The results appear as follows:



Crosstab11111

		Revenue	Camping Equipment	Golf Equipment	Outdoor Protection	Personal Accessories
Asia Pacific	Australia	4 Golf only		1,197,219.17		678,010.16
		Beach Beds Pty Ltd.	7,501,952.13	1,690,507.24	53,956.62	2,287,985.81
		Black Stump Camping Supplies	848,498.08		7,719	1,283,603.44
		Blue Mountains Golfing Company		3,510,880.19		769,943.45
		Can't Beat The Bush Supplies	169,495.81		4,796.41	
		Gone Bush Supplies	482,703.36	290,099.92	3,558	19,755.12
		Harbour Pty Ltd.	571,323.6		13,660.04	284,711.66
		Jackos Enviro Shop	2,751,417.54		42,245.29	331,127.1
		Kanga Kampers	4,567,592.33		65,442.65	2,929,630.47
		OutBack Pty	1,047,088.26		23,344.94	1,224,238.2
		Southern Cross Pty.				3,310,618.45
		Top End Equipment	1,235,886.09		152,913.43	160,893.21
		Watson's Golf Supplies		1,793,732.15		631,934.7

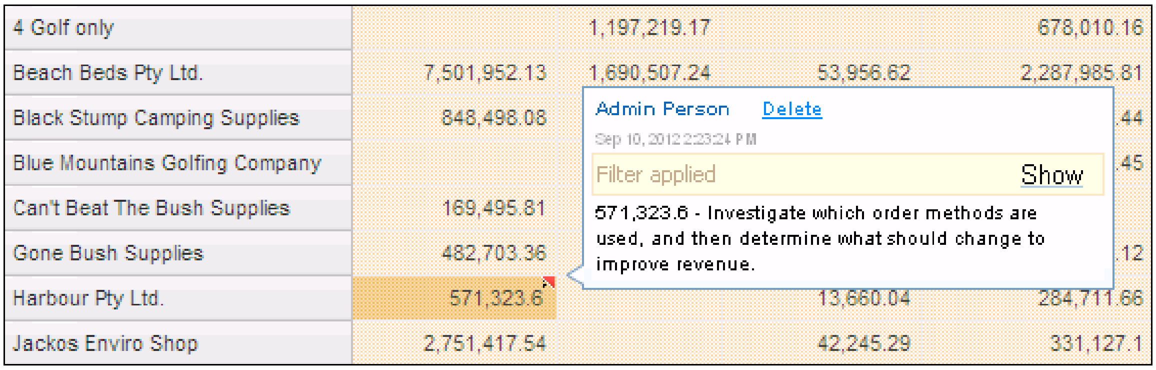
16. On the report widget toolbar, click **Comment**, click **Add – 571,323.6**, and then in the box, type: **Investigate which order methods are used, and then determine what should change to improve revenue.**.

17. Click **Done**.

    A small red triangle at the top right of the cell indicates that a comment is available for reviewers.

18. Place the cursor over the red triangle.

    The results appear as follows:



4 Golf only		1,197,219.17		678,010.16
Beach Beds Pty Ltd.	7,501,952.13	1,690,507.24	53,956.62	2,287,985.81
Black Stump Camping Supplies	848,498.08			.44
Blue Mountains Golfing Company				.45
Can't Beat The Bush Supplies	169,495.81			
Gone Bush Supplies	482,703.36			.12
Harbour Pty Ltd.	571,323.6		13,660.04	284,711.66
Jackos Enviro Shop	2,751,417.54		42,245.29	331,127.1

Admin Person Delete

Sep 10, 2012 2:23:24 PM

Filter applied Show

571,323.6 - Investigate which order methods are used, and then determine what should change to improve revenue.

    The comment displays, with the name of the author and the date and time that the comment was added. Reviewers who see this workspace, can review this comment, and add their own comments.

19. On the **Cognos Workspace** toolbar, click **Save**.

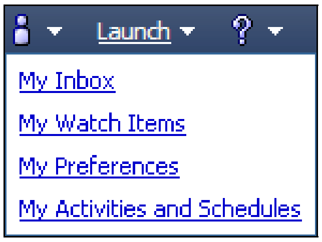
20. From the **Actions** menu, point to **Launch** and then select **IBM Cognos Connection**.

    You are returned to IBM Cognos Connection.

Task 9. Explore preferences.

1. On the **IBM Cognos Connection** toolbar, click **My Area Options** .

   The results appear as follows:

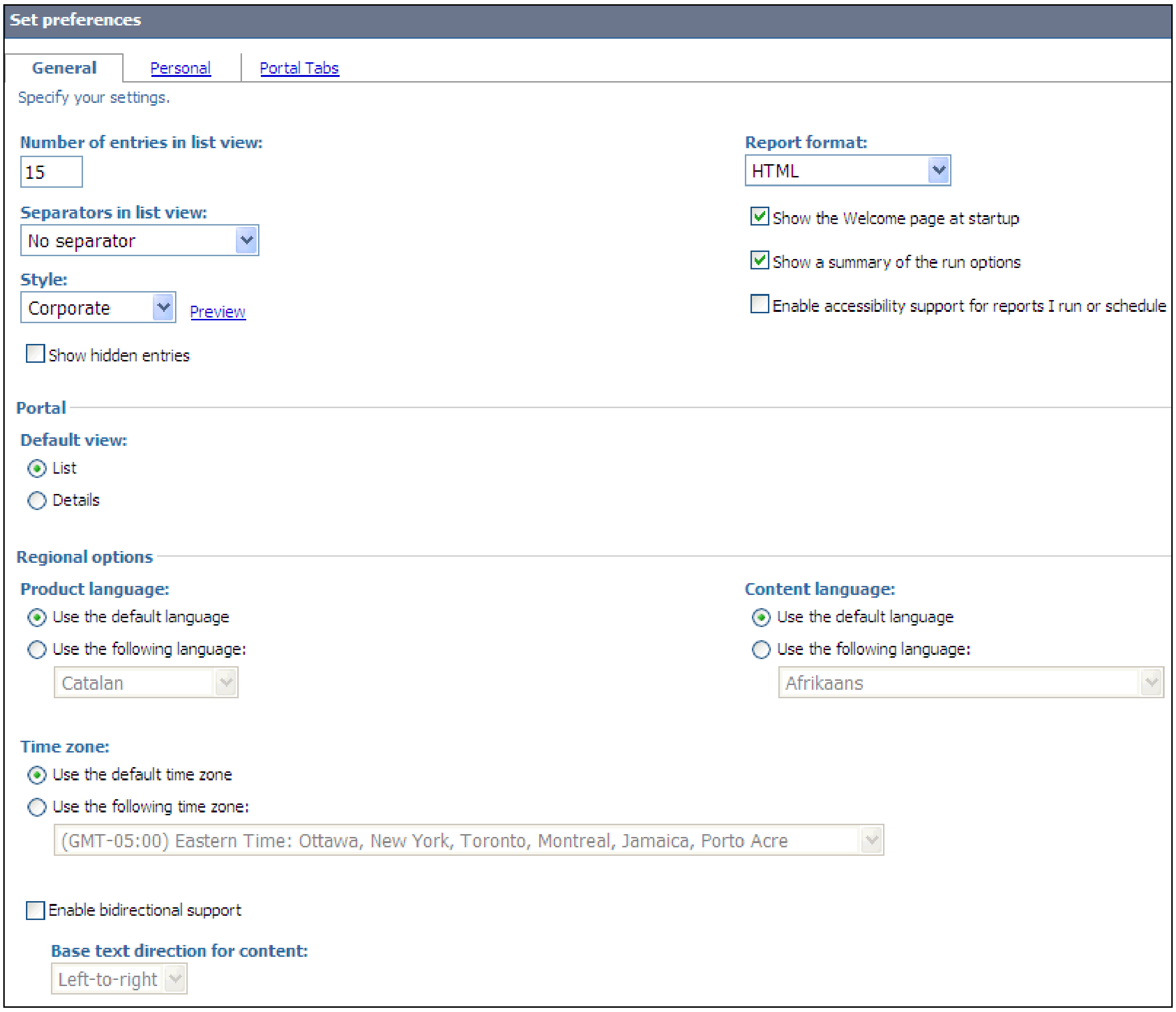
   

   Here you can view and manage:

   - Inbox (notification requests and ad hoc tasks)
   - Watch Items (user-specified rules to receive alerts based on values in a report)
   - Personal Preferences
   - Activities and Schedules (such as reports you are running or have run in the past, as well as reports you have scheduled to run)

2. Click **My Preferences**.

   The results appear as follows:



Set preferences

General | Personal | Portal Tabs

Specify your settings.

Number of entries in list view: 15

Separators in list view: No separator

Style: Corporate Preview

☐ Show hidden entries

Report format: HTML

☑ Show the Welcome page at startup

☑ Show a summary of the run options

☐ Enable accessibility support for reports I run or schedule

Portal

Default view:

◉ List

○ Details

Regional options

Product language:

◉ Use the default language

○ Use the following language: Catalan

Content language:

◉ Use the default language

○ Use the following language: Afrikaans

Time zone:

◉ Use the default time zone

○ Use the following time zone: (GMT-05:00) Eastern Time: Ottawa, New York, Toronto, Montreal, Jamaica, Porto Acre

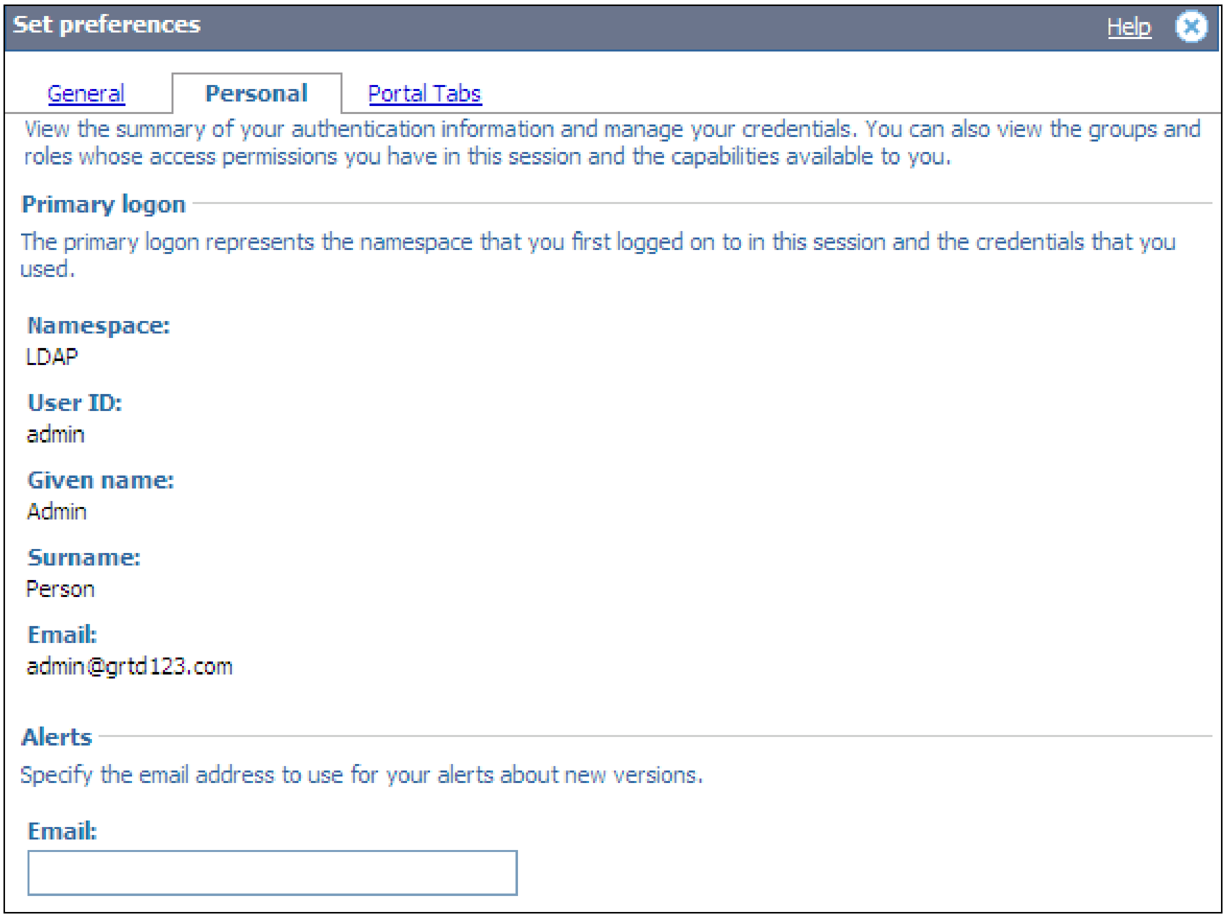
☐ Enable bidirectional support

Base text direction for content: Left-to-right

On the General tab you can configure your personal portal preferences. For example, you can change the UI language and, if your data and metadata support it, you can change your content language as well. You can also enable bidirectional language support, for example for Arabic and Hebrew text. This lets you control the entry and display of bidirectional text strings in IBM Cognos applications. The affected content includes entry names, descriptions, labels and tooltips, and structured text, such as email addressees and file paths, as well as comments in Cognos Viewer.

3. Click the **Personal** tab.

   The results appear as follows:



Set preferences

Help

General | Personal | Portal Tabs

View the summary of your authentication information and manage your credentials. You can also view the groups and roles whose access permissions you have in this session and the capabilities available to you.

Primary logon

The primary logon represents the namespace that you first logged on to in this session and the credentials that you used.

Namespace:
LDAP

User ID:
admin

Given name:
Admin

Surname:
Person

Email:
admin@grtd123.com

Alerts

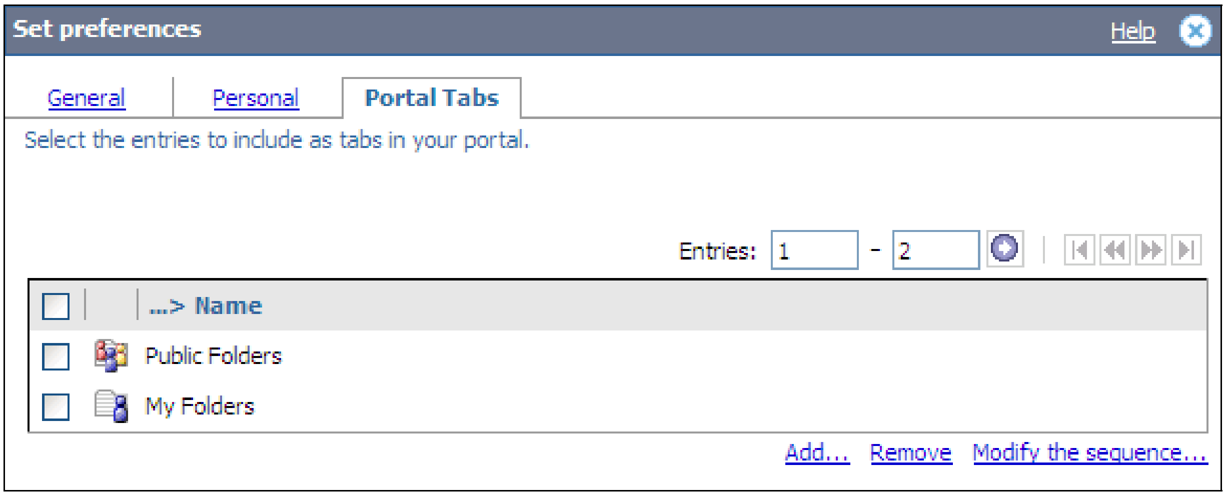
Specify the email address to use for your alerts about new versions.

Email:

On this tab you can see your personal information as well as which groups and roles you belong to and which capabilities you have in the IBM Cognos environment (not all elements are shown in the screen capture). Depending on your authentication provider and capabilities, information on this page will differ.

4. Click the **Portal Tabs** tab.

   The results appear as follows:



Set preferences

Help

General | Personal | Portal Tabs

Select the entries to include as tabs in your portal.

Entries: 1 - 2

☐	...> Name
☐	Public Folders
☐	My Folders

Add... Remove Modify the sequence...

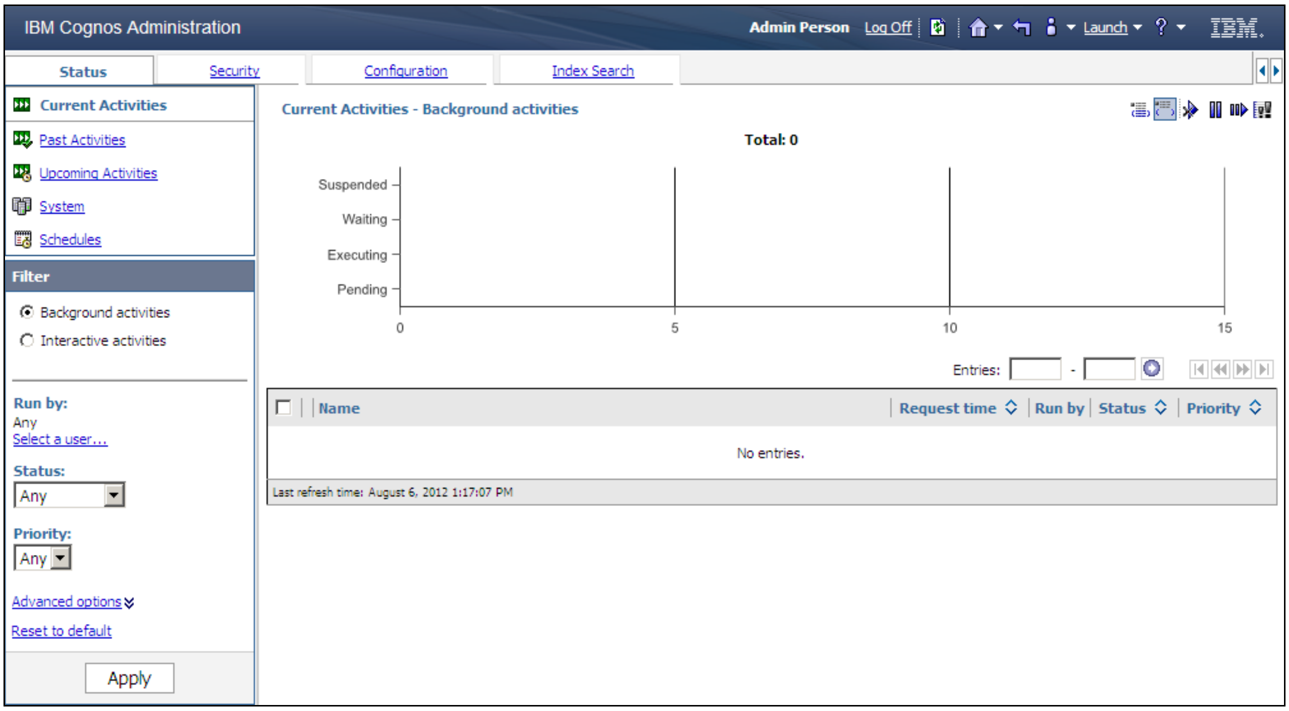
   Here you can manage your portal tabs by adding, removing, or changing the order of the tabs.

5. Click **Cancel**.

Task 10. Explore IBM Cognos Administration.

1. From the **Launch** menu, click **IBM Cognos Administration**.

   The result appears as follows:



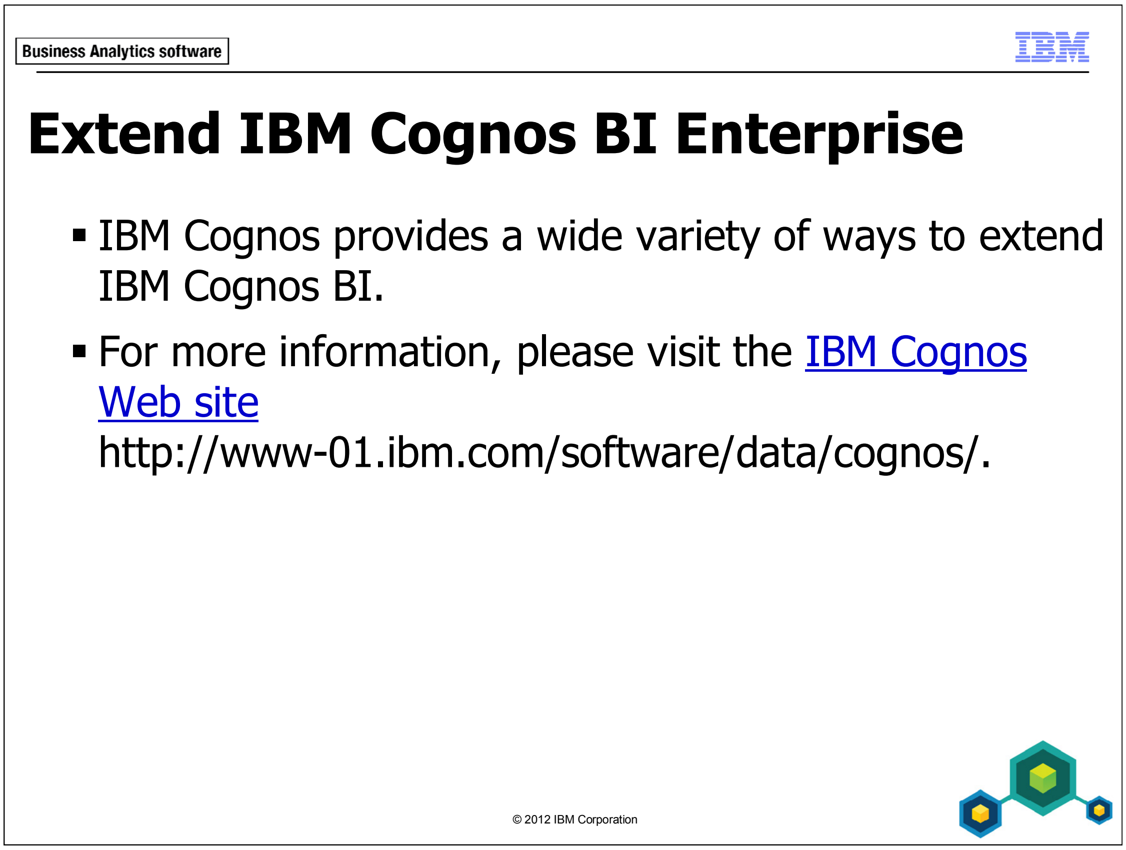
No entries.

Last refresh time: August 6, 2012 1:17:07 PM

   This portion of the portal requires administrative access and allows you to monitor and administer IBM Cognos including servers, security, capabilities, data source connections, and the deployment of content.

2. Click **Log Off**, and then close **Internet Explorer**.

Results:
As an introduction to IBM Cognos BI, you briefly explored one of the modeling tools, IBM Cognos Connection, and various BI studios and desktop applications to familiarize yourself with the environment and BI workflow.



Business Analytics software

Extend IBM Cognos BI Enterprise

- IBM Cognos provides a wide variety of ways to extend IBM Cognos BI.
- For more information, please visit the IBM Cognos Web site http://www-01.ibm.com/software/data/cognos/.

© 2012 IBM Corporation

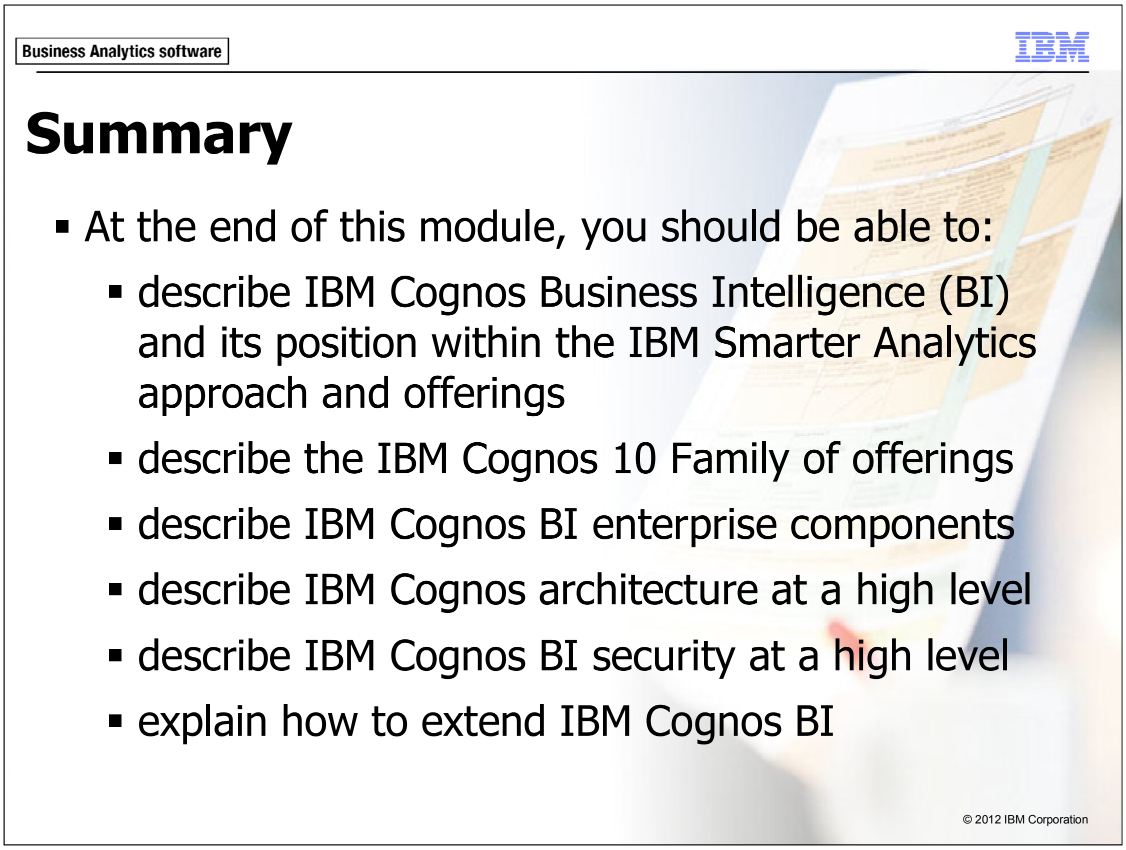
Cognos products that extend IBM Cognos BI include:

- IBM Cognos for Microsoft Office (integrate IBM Cognos content with MS Office)
- IBM Cognos Mobile (IBM Cognos content on mobile devices)
- IBM Cognos Analysis for Microsoft Excel (multidimensional analysis on IBM Cognos BI data in MS Excel spreadsheets)
- IBM Cognos Mashup Service

For developers, there is also Composite, which allows for access to an even wider variety of data sources, and the IBM Cognos SDK for customization and application development.

For those requiring access to realtime monitoring of operational data, IBM Cognos offers IBM Cognos BI Real-time Monitoring, which delivers highly visual, interactive, and self-service dashboards, data integration, analysis, and reports..

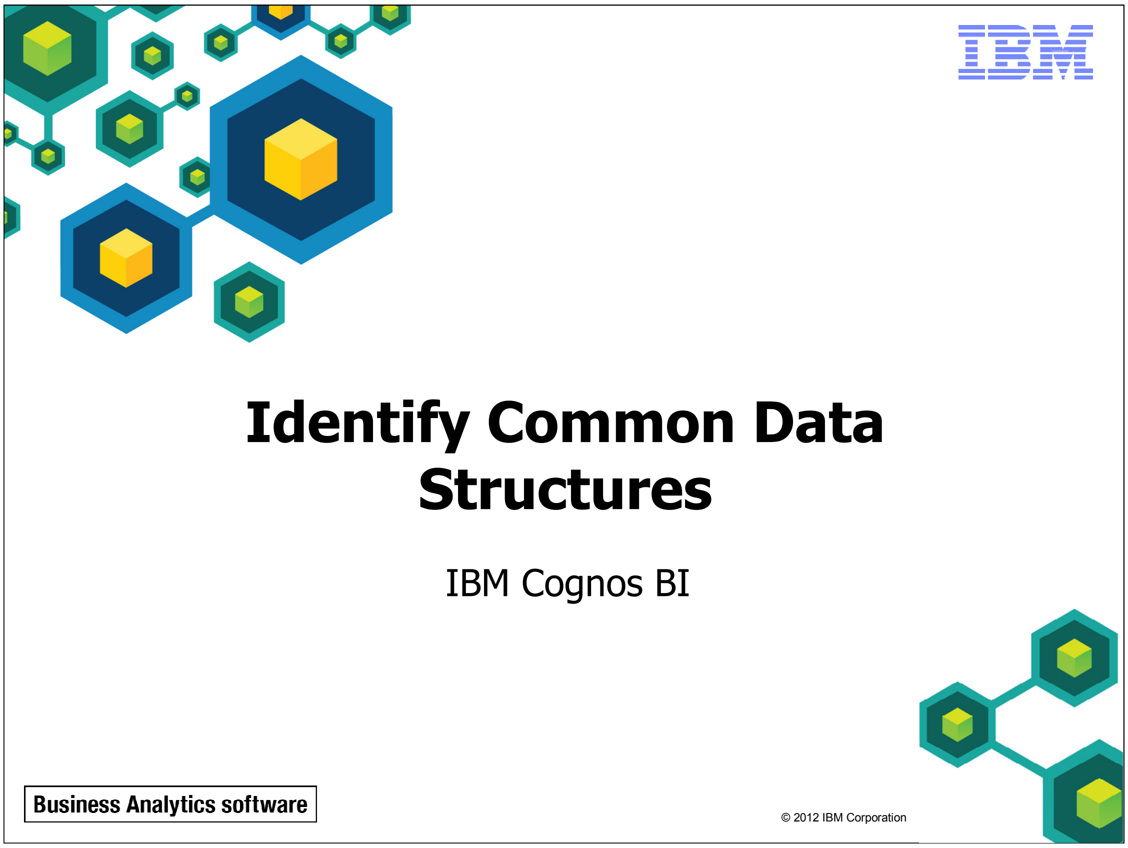
You can incorporate TM1 widgets into IBM Cognos to allow users to interact with financial plans.



Business Analytics software

Summary

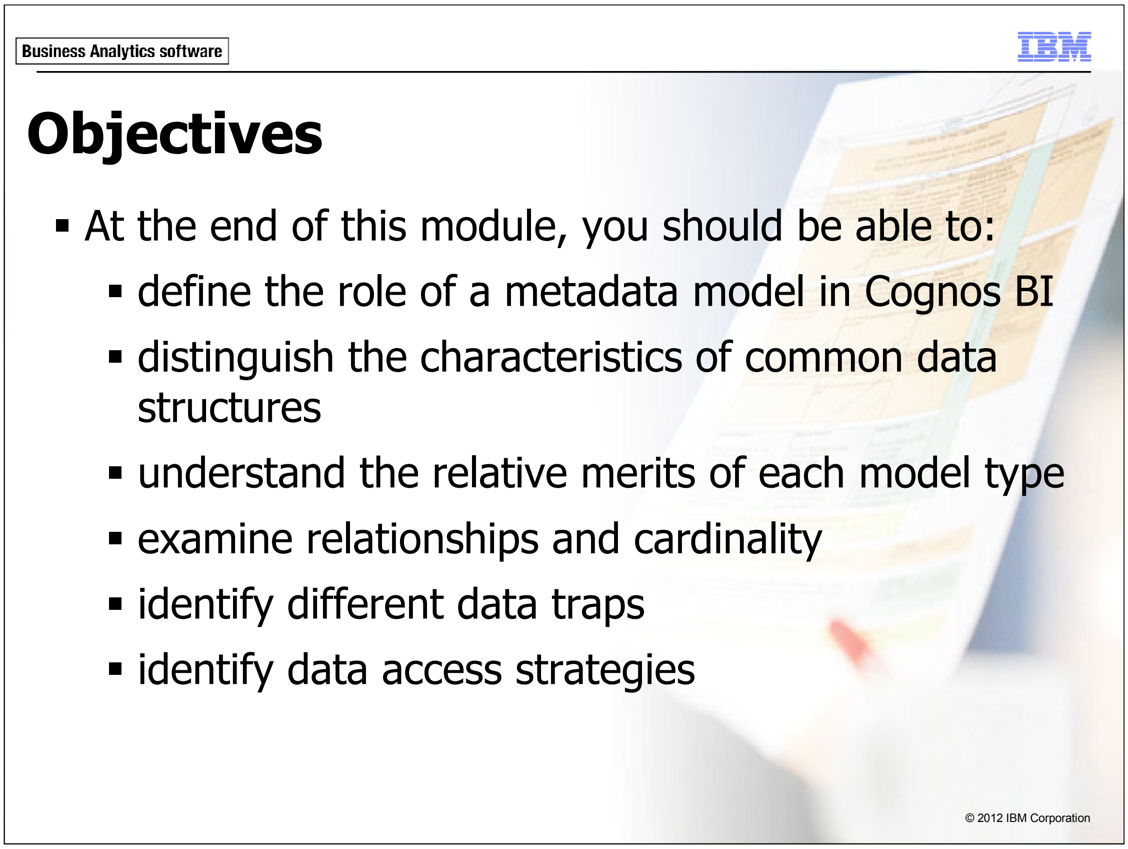
- At the end of this module, you should be able to:
  - describe IBM Cognos Business Intelligence (BI) and its position within the IBM Smarter Analytics approach and offerings
  - describe the IBM Cognos 10 Family of offerings
  - describe IBM Cognos BI enterprise components
  - describe IBM Cognos architecture at a high level
  - describe IBM Cognos BI security at a high level
  - explain how to extend IBM Cognos BI



Identify Common Data Structures

IBM Cognos BI

Business Analytics software



Business Analytics software

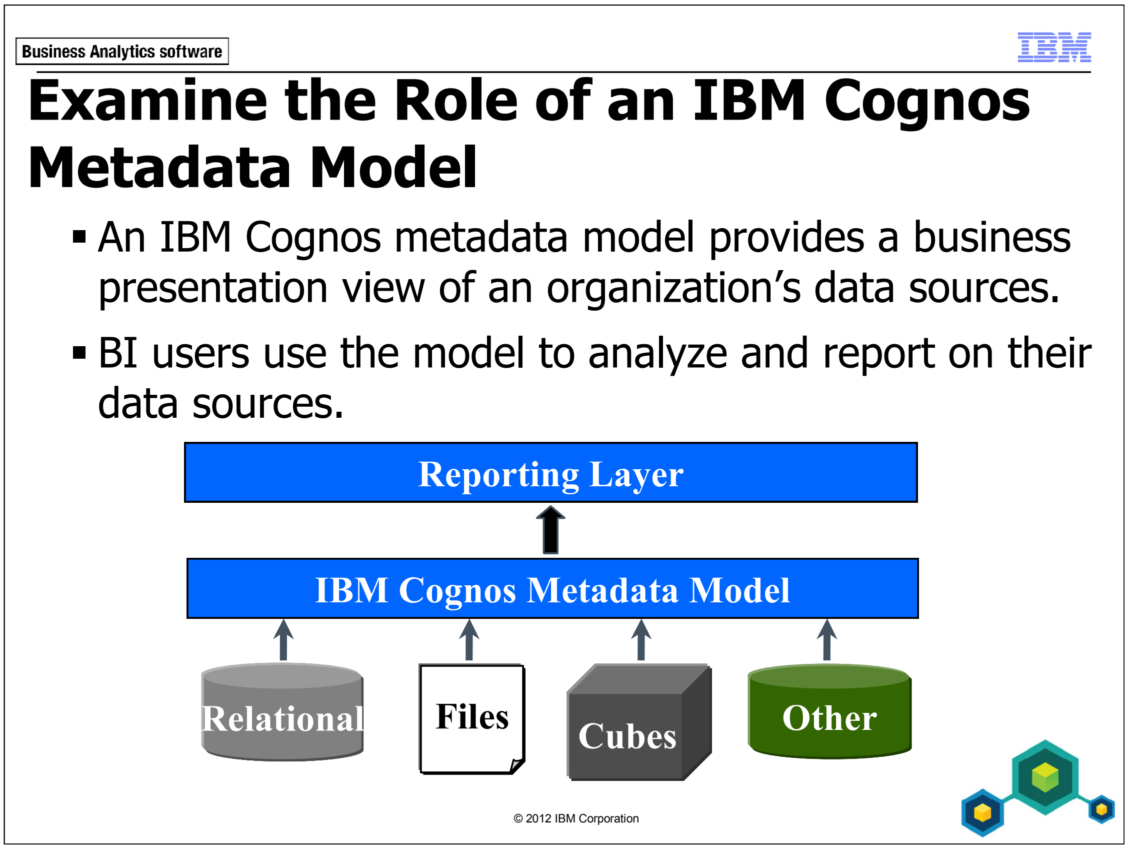
Objectives

- At the end of this module, you should be able to:
  - define the role of a metadata model in Cognos BI
  - distinguish the characteristics of common data structures
  - understand the relative merits of each model type
  - examine relationships and cardinality
  - identify different data traps
  - identify data access strategies

© 2012 IBM Corporation

This module will focus on industry standard data structures. This knowledge is essential before beginning a metadata modeling project. Framework Manager topics will be covered in subsequent modules.

When referring to a modeler in this module, we are referring to a data modeler, not a metadata modeler.



Business Analytics software

Examine the Role of an IBM Cognos Metadata Model

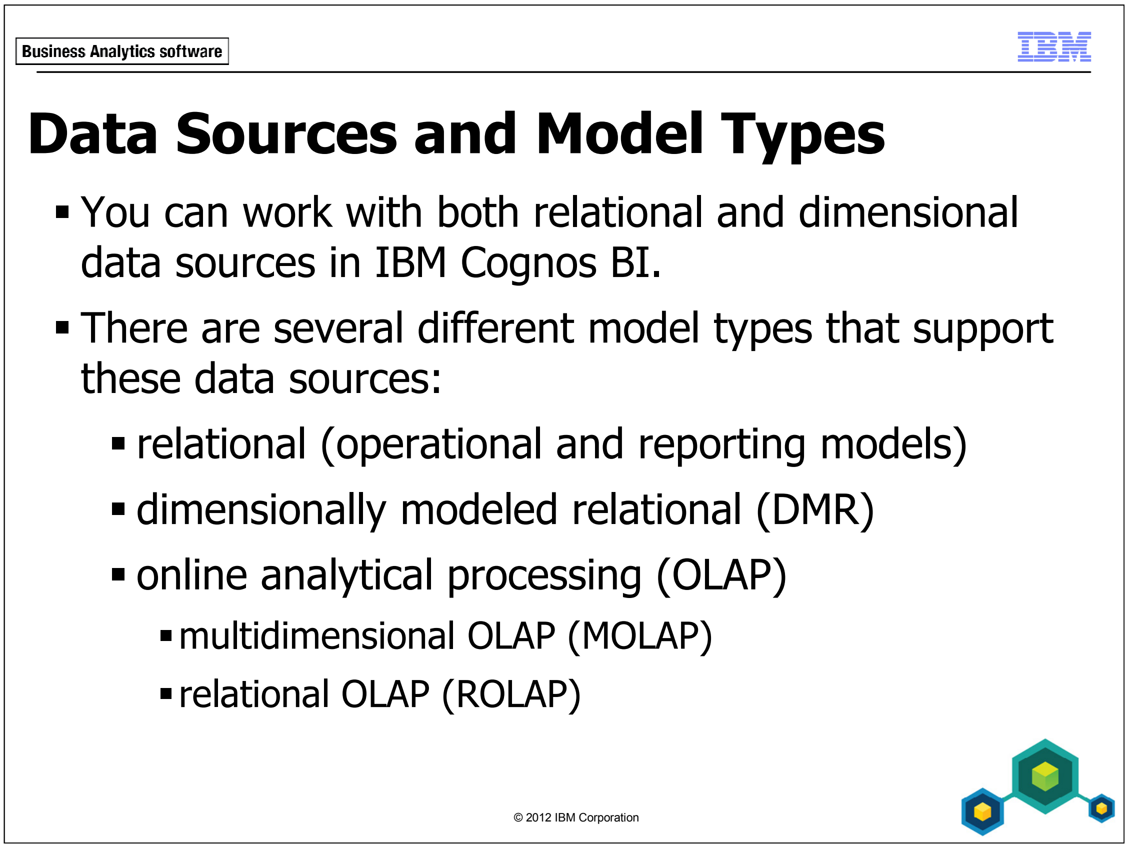
- An IBM Cognos metadata model provides a business presentation view of an organization's data sources.
- BI users use the model to analyze and report on their data sources.

Reporting Layer
IBM Cognos Metadata Model
Relational
Files
Cubes
Other

© 2012 IBM Corporation

A metadata model can hide the structural complexity of your underlying data sources. By creating a metadata model, you have more control over how your data is presented to end users. You can also choose which data to display to your end users and how it will be organized. The overall goal of modeling the metadata is to create a model that provides predictable results as well as an easy-to-use view of the metadata for authors and analysts.

Your underlying data sources may be very diverse. For example, you may have operational or reporting data in one or more relational databases. You may also have legacy data in various file formats, such as text, comma separated values (.csv), and extensible markup language (XML). You may even have online analytical processing (OLAP) sources that include cubes (such as Cognos PowerCubes), as well as other sources such as SAP BW.



Business Analytics software

Data Sources and Model Types

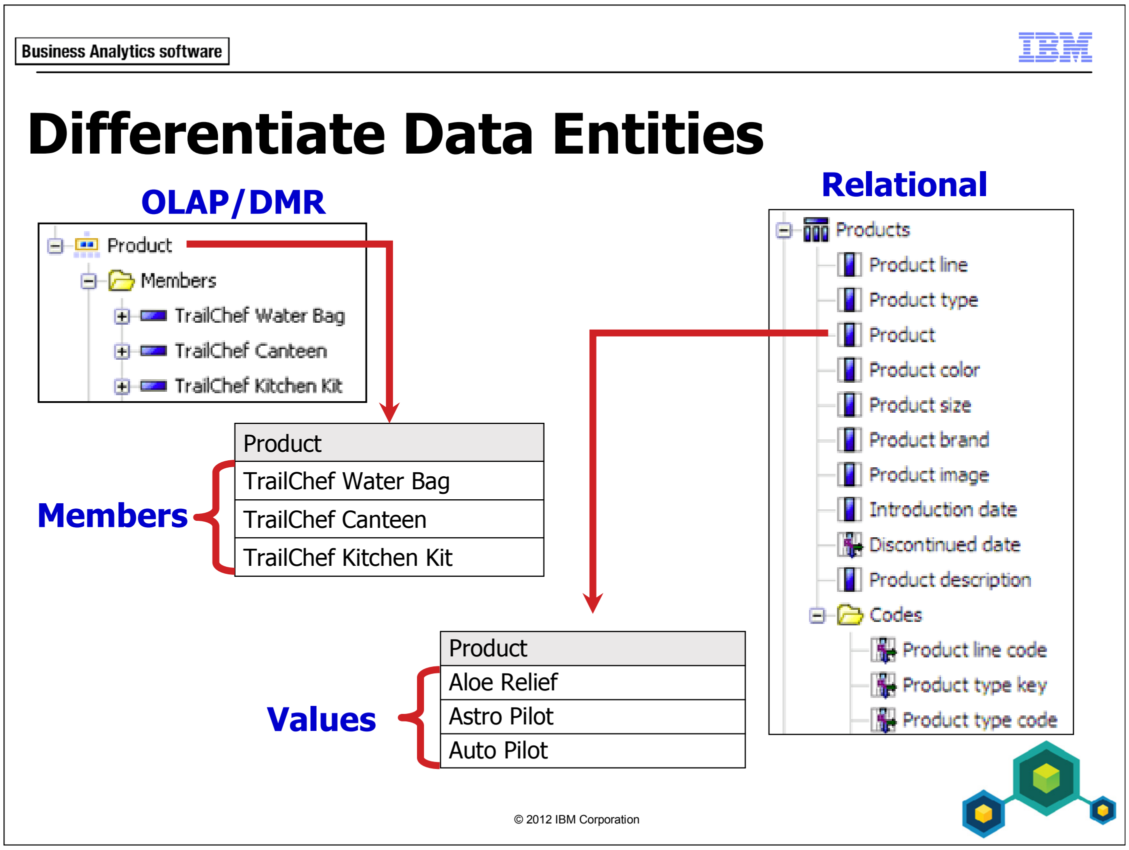
- You can work with both relational and dimensional data sources in IBM Cognos BI.
- There are several different model types that support these data sources:
  - relational (operational and reporting models)
  - dimensionally modeled relational (DMR)
  - online analytical processing (OLAP)
    - multidimensional OLAP (MOLAP)
    - relational OLAP (ROLAP)

© 2012 IBM Corporation

Relational models have a basic metadata structure that looks like tables and columns in a database. Relational models can have either an operational (normalized) or reporting structure (star-schema). You can use Framework Manager to transform metadata from an operational data source into a model that optimizes it for reporting.

Dimensionally Modeled Relational (DMR) models are built from relational data sources, but are modeled with a dimensional structure (like OLAP) consisting of dimensions, hierarchies, and measures. Using Framework Manager, you can add dimensional metadata to model objects, which enables drill-through capabilities in Cognos BI.

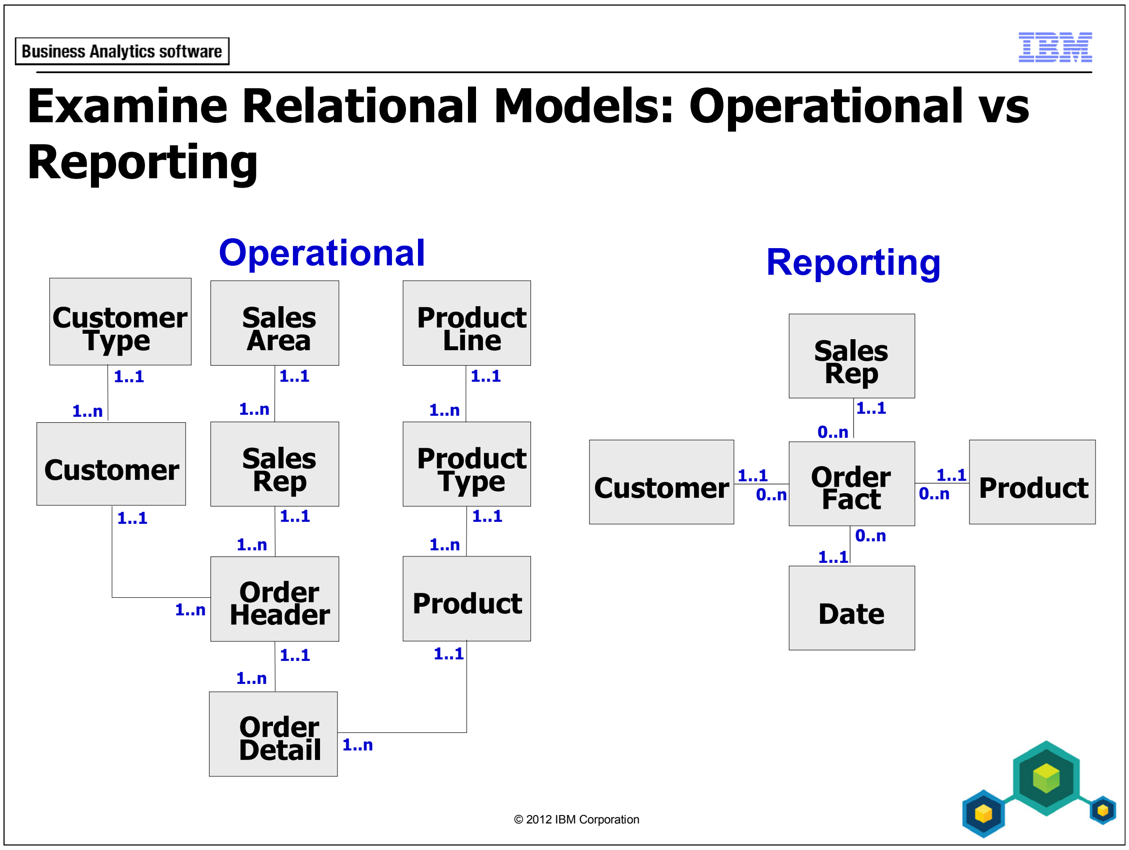
OLAP models point to dimensional data sources, such as IBM Cognos PowerCubes, which are built using their own modeling tools, such as Transformer.



OLAP and DMR data entities are different from relational data entities:

- values (relational)
- members (OLAP and DMR)

When you create a report using a dimensional model, you work with the member. Each member has certain properties such as a member key and member caption. Report authors and consumers see the member caption. Each member is defined and identified by its member unique name (MUN), which describes its position in the dimensional structure.



Business Analytics software

Examine Relational Models: Operational vs Reporting

© 2012 IBM Corporation

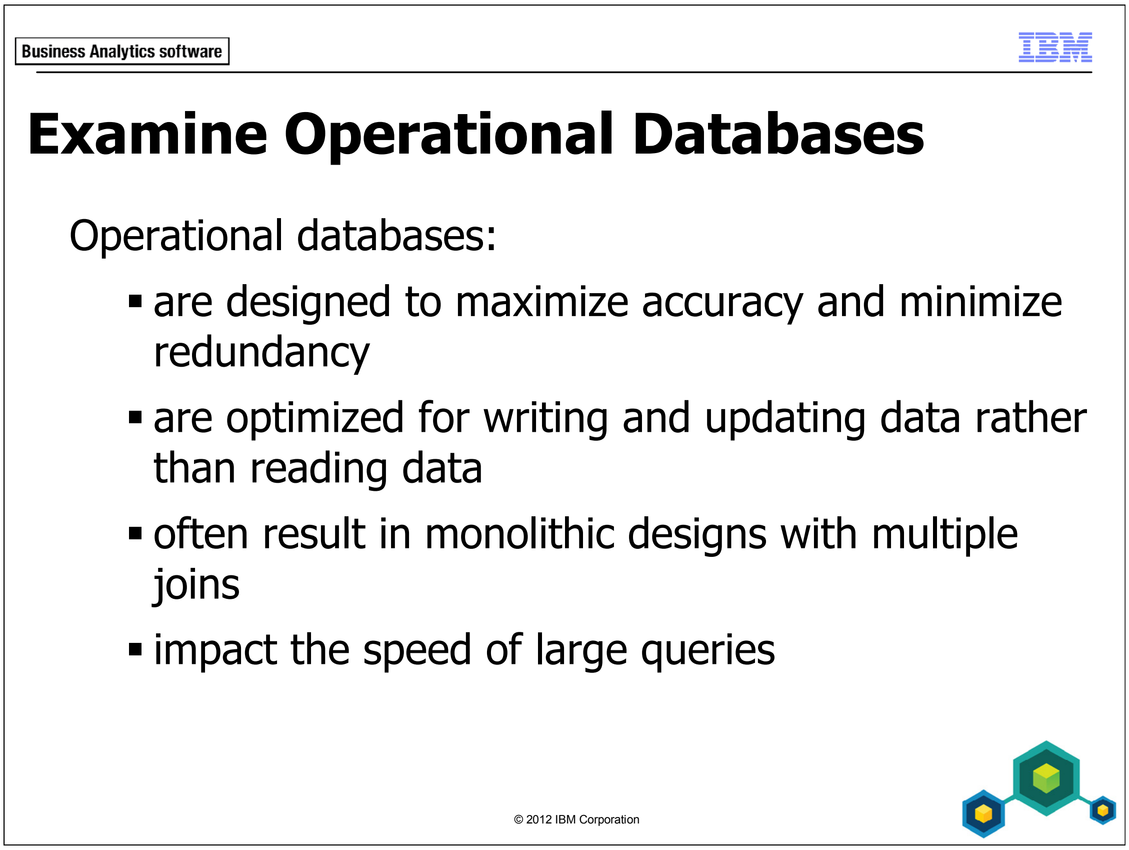
Operational databases are:

- used to track the day-to-day operations of a business
- usually normalized or part of an enterprise resource planning (ERP) vendor package

Reporting databases are:

- typically a copy of the operational data
- structured to make reporting faster and easier
- usually dimensional, taking the form of a star schema design

In general, we recommend that you create a logical model that conforms to star schema concepts. This is a requirement for IBM Cognos Analysis Studio and has also proved to be an effective way to organize data for your users.



Business Analytics software

Examine Operational Databases

Operational databases:

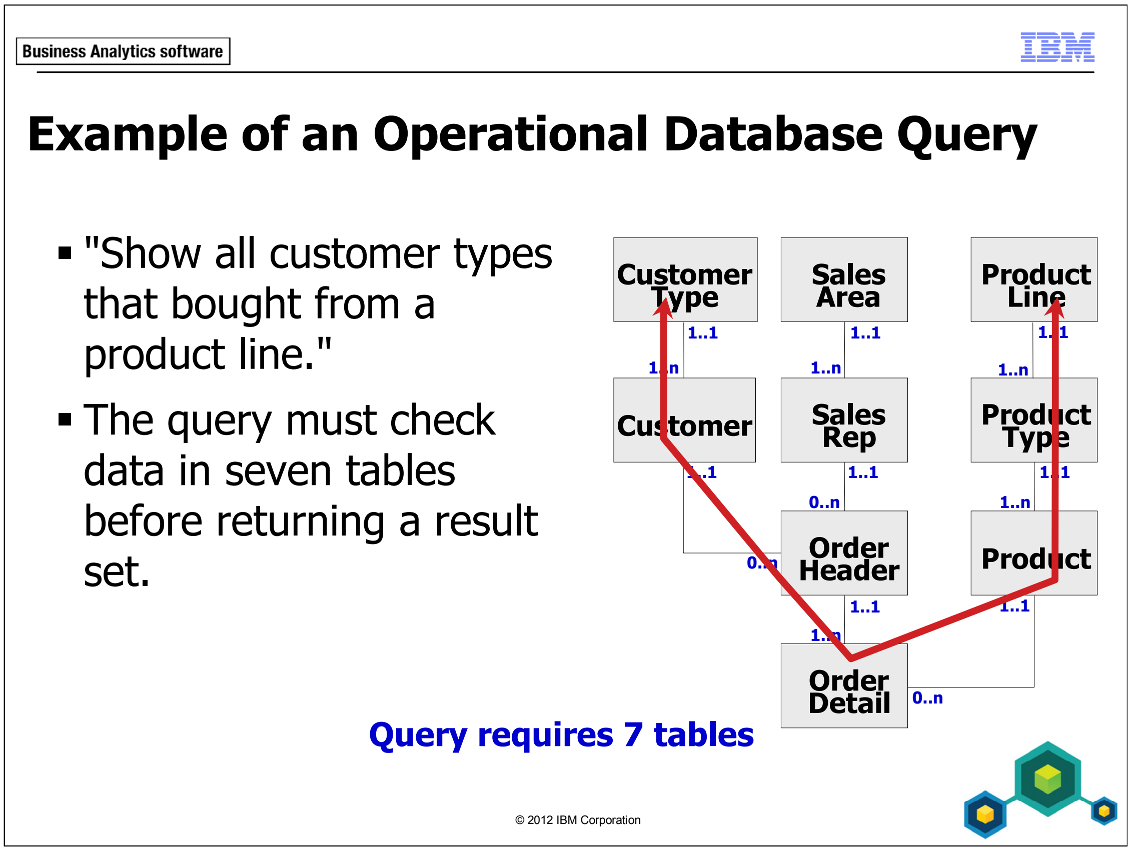
- are designed to maximize accuracy and minimize redundancy
- are optimized for writing and updating data rather than reading data
- often result in monolithic designs with multiple joins
- impact the speed of large queries

© 2012 IBM Corporation

It can be difficult to report from an operational database because:

- reading the data can impact the performance of the system
- the number of tables that must be joined to satisfy a business question can be prohibitively large
- a more complex database requires more metadata modeling efforts to make it consumable by business users

Operational systems are designed with one goal in mind: to get data into the database quickly. These databases are normalized to reduce redundancy. Having little to no redundancy ensures that there is data integrity and that database triggers function properly, so that the right data is captured.



Business Analytics software

Example of an Operational Database Query

- "Show all customer types that bought from a product line."
- The query must check data in seven tables before returning a result set.

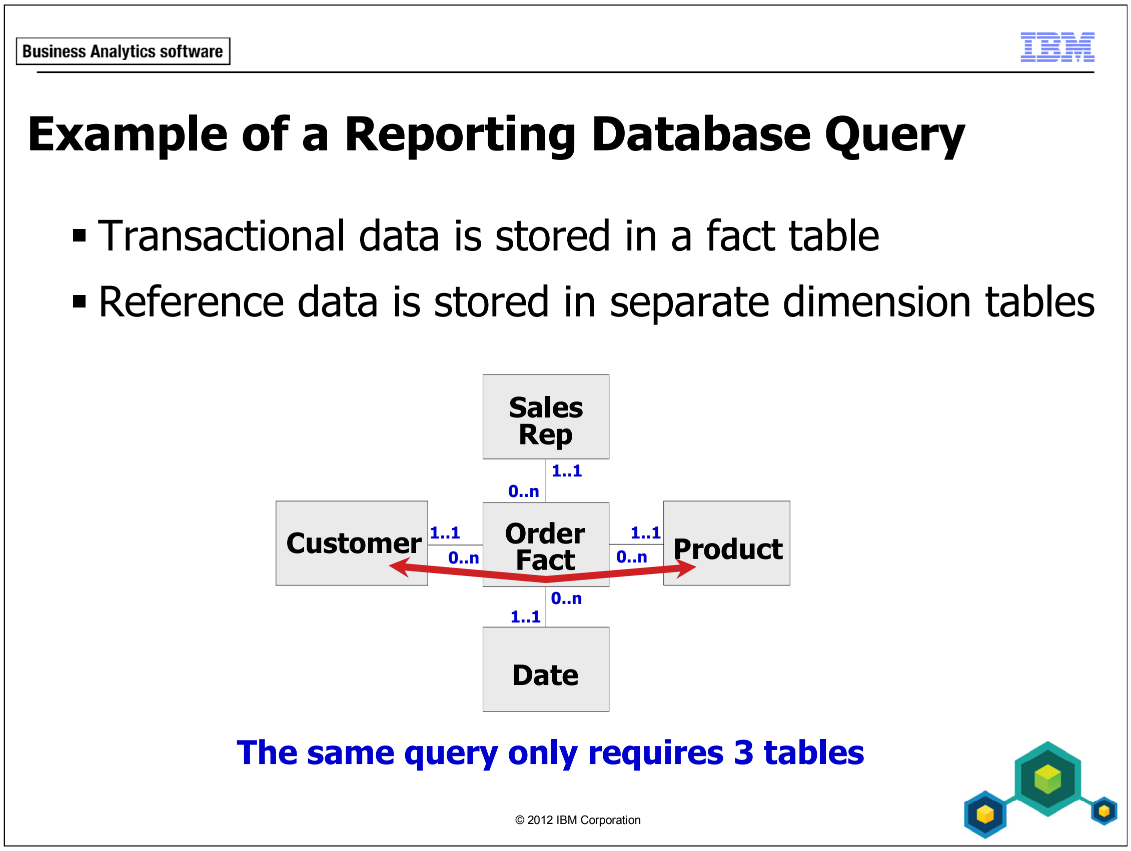
Customer Type
Sales Area
Product Line
1..1
1..1
1..1
1..n
1..n
1..n
Customer
Sales Rep
Product Type
1..1
1..1
1..1
0..n
1..n
Order Header
Product
0..n
1..1
1..1
1..n
Order Detail
0..n

Query requires 7 tables

© 2012 IBM Corporation

Operational databases may be too complex for general reporting. As a result, reports that are generated against these structures may take a long time to run. They may also have unpredictable results.

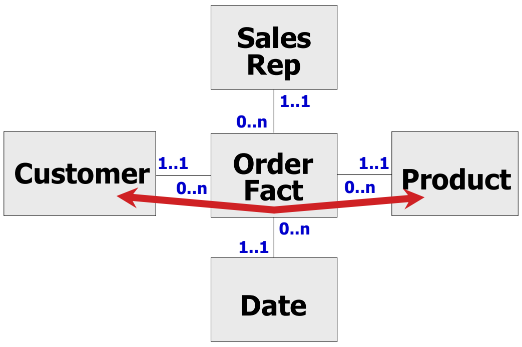
As shown in the slide example, a report may need to access many tables to retrieve all the necessary data.



Business Analytics software

Example of a Reporting Database Query

- Transactional data is stored in a fact table
- Reference data is stored in separate dimension tables



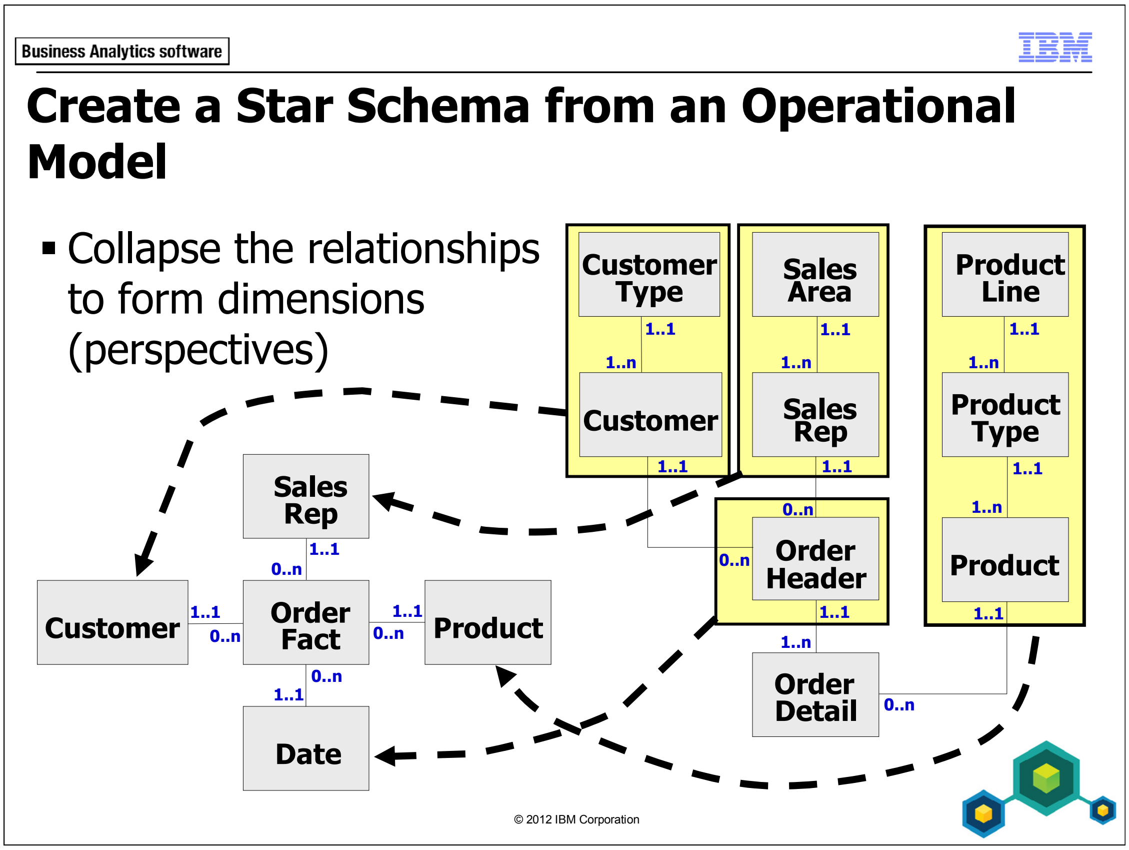
The same query only requires 3 tables

A star schema is a design with two basic components:

- a central table, known as the fact table (typically numeric data)
- satellite tables, known as the dimension tables

Because a star schema database contains fewer tables than a fully normalized database, query performance is much faster.

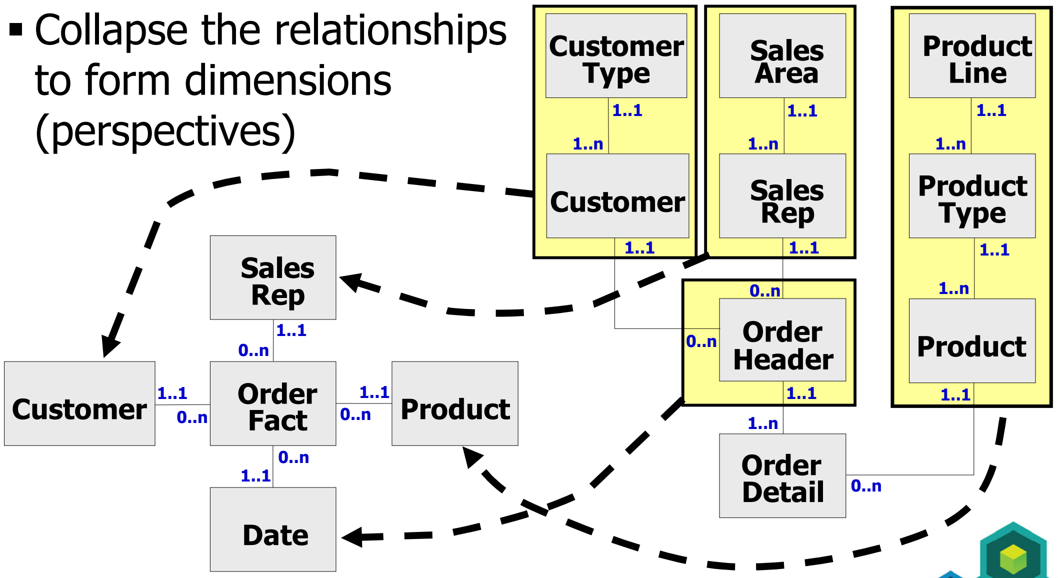
- A typical query against a star schema database focuses on the central fact table and makes integrity checks against the related dimension tables.
- If the query is to retrieve information about a specific subject area only, such as all the products that belong to a particular product line, then the query will be even faster.



Business Analytics software

Create a Star Schema from an Operational Model

- Collapse the relationships to form dimensions (perspectives)



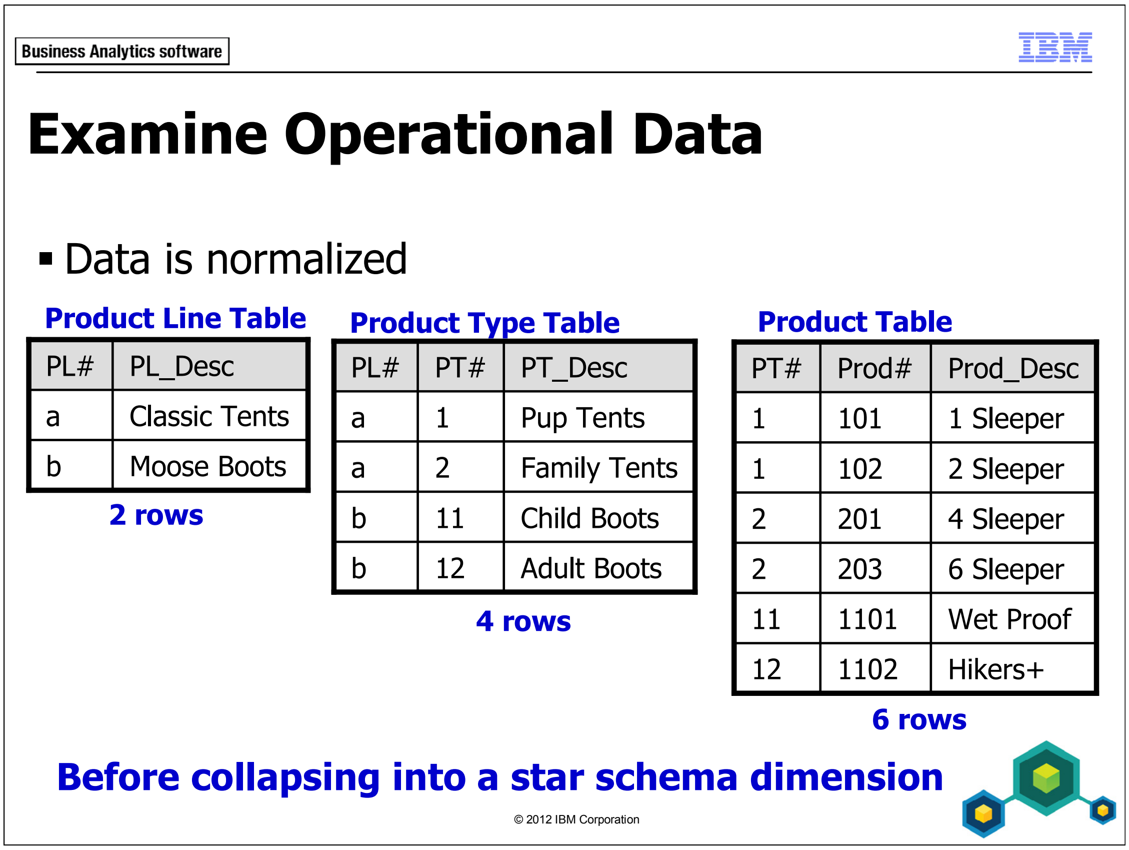
© 2012 IBM Corporation

Each table in a star schema database will contain an expanded set of data.

Extract Transform and Load (ETL) tools can be used to create a star schema data warehouse, or you may use a metadata modeling tool to emulate a star schema structure by generating the appropriate SQL at report design time. The second option will not improve performance, but will yield predictable results.

The Cognos BI ETL tool is called Cognos Data Manager. Framework Manager cannot create a warehouse, but it can emulate a star schema structure by collapsing query subjects to simplify the view and generate the appropriate SQL at run time.

The Data Warehouse Toolkit by Ralph Kimball is a good resource for information on building data ware houses.



Business Analytics software

Examine Operational Data

- Data is normalized

Product Line Table

PL#	PL_Desc
a	Classic Tents
b	Moose Boots

2 rows

Product Type Table

PL#	PT#	PT_Desc
a	1	Pup Tents
a	2	Family Tents
b	11	Child Boots
b	12	Adult Boots

4 rows

Product Table

PT#	Prod#	Prod_Desc
1	101	1 Sleeper
1	102	2 Sleeper
2	201	4 Sleeper
2	203	6 Sleeper
11	1101	Wet Proof
12	1102	Hikers+

6 rows

Before collapsing into a star schema dimension

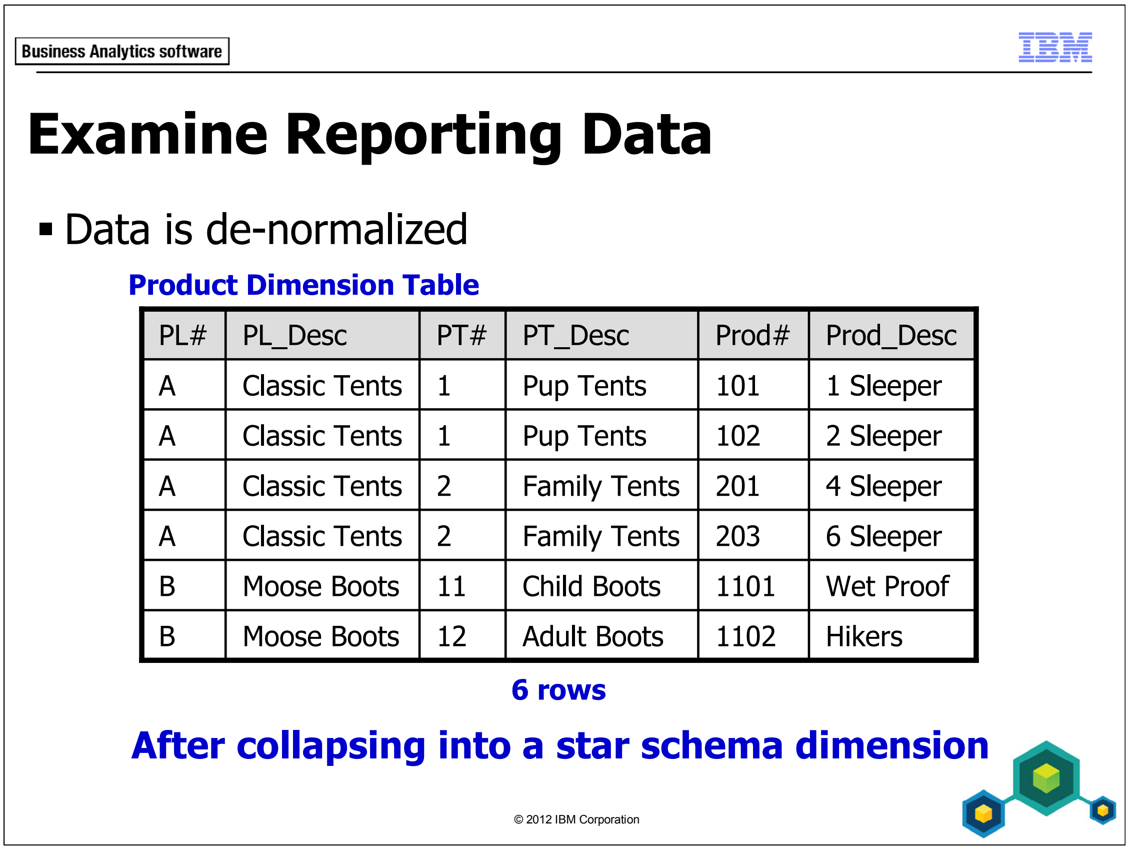
© 2012 IBM Corporation

The slide example shows three normalized tables that represent three hierarchical levels. Products roll up into product types, and product types roll up into product lines.

The Product Line table has two rows that indicate two lines of products sold by the company.

The Product Type table contains four rows to indicate the four types of products that fall under the previous two product lines (two types per product line).

The Product table contains the greatest level of detail. It holds 6 rows to represent the 6 products that fall under the four product types.



Business Analytics software

IBM

Examine Reporting Data

- Data is de-normalized

Product Dimension Table

PL#	PL_Desc	PT#	PT_Desc	Prod#	Prod_Desc
A	Classic Tents	1	Pup Tents	101	1 Sleeper
A	Classic Tents	1	Pup Tents	102	2 Sleeper
A	Classic Tents	2	Family Tents	201	4 Sleeper
A	Classic Tents	2	Family Tents	203	6 Sleeper
B	Moose Boots	11	Child Boots	1101	Wet Proof
B	Moose Boots	12	Adult Boots	1102	Hikers

6 rows

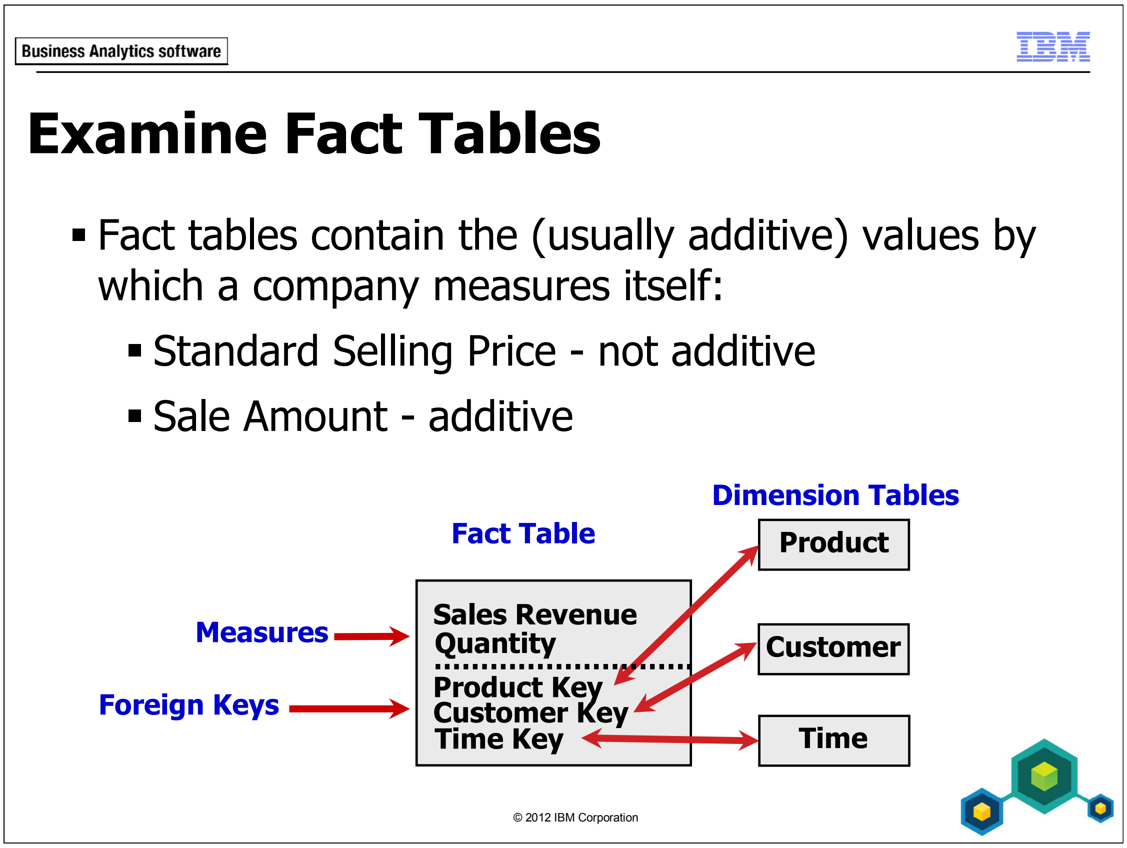
After collapsing into a star schema dimension

The slide example shows a de-normalized dimension table created from the three normalized tables shown on the previous slide.

The Product Line table forms the first two columns of the new dimension table (PL# and PL_Desc), the Product Type table forms the next two columns (PT# and PT_Desc), and the Product table forms the last two columns (Prod# and Prod_Desc).

The main characteristic of this table is its redundancy. Note that each product line (Classic Tents and Moose Boots) is repeated, once for each product that the product line contains. The same applies for product type.

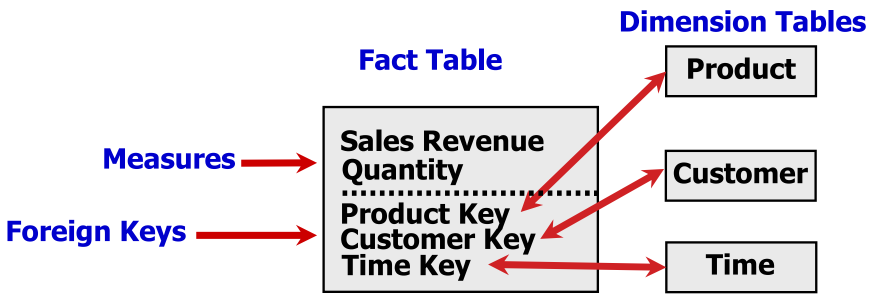
This type of table is unsuitable for a normalized system, but is ideal for a reporting and querying structure.



Business Analytics software

Examine Fact Tables

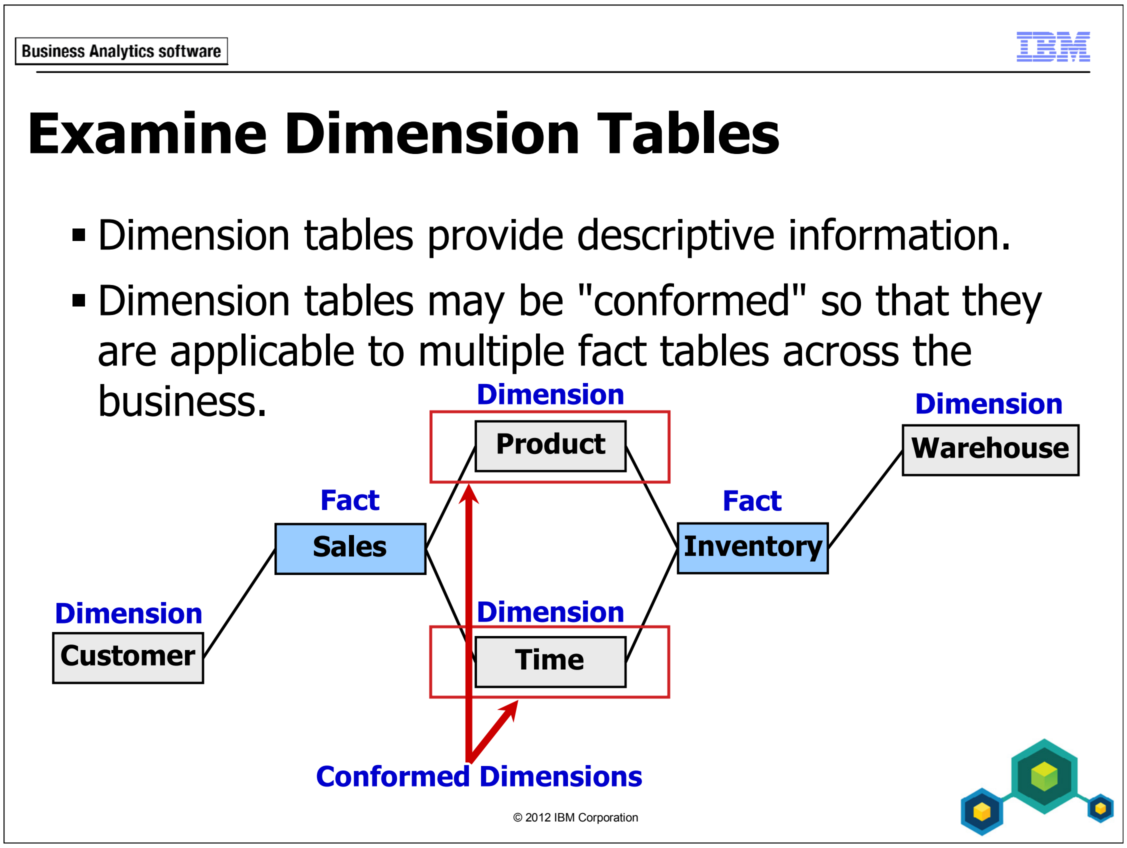
- Fact tables contain the (usually additive) values by which a company measures itself:
  - Standard Selling Price - not additive
  - Sale Amount - additive



© 2012 IBM Corporation

Fact tables are the focal point of any star schema, and typically contain the most rows. There are typically no descriptive attributes in a fact table. Instead, there are foreign keys that relate to the dimension tables, which contain descriptive attributes.

Facts in a fact table are also known as metrics, measures, or key performance indicators. There are cases where you may encounter factless fact tables, in which only foreign keys are found. For example, in the case of a library, you may have a fact table that only contains a book key, a customer key, and a day key, which records which books were checked out by which customer and when.



Business Analytics software

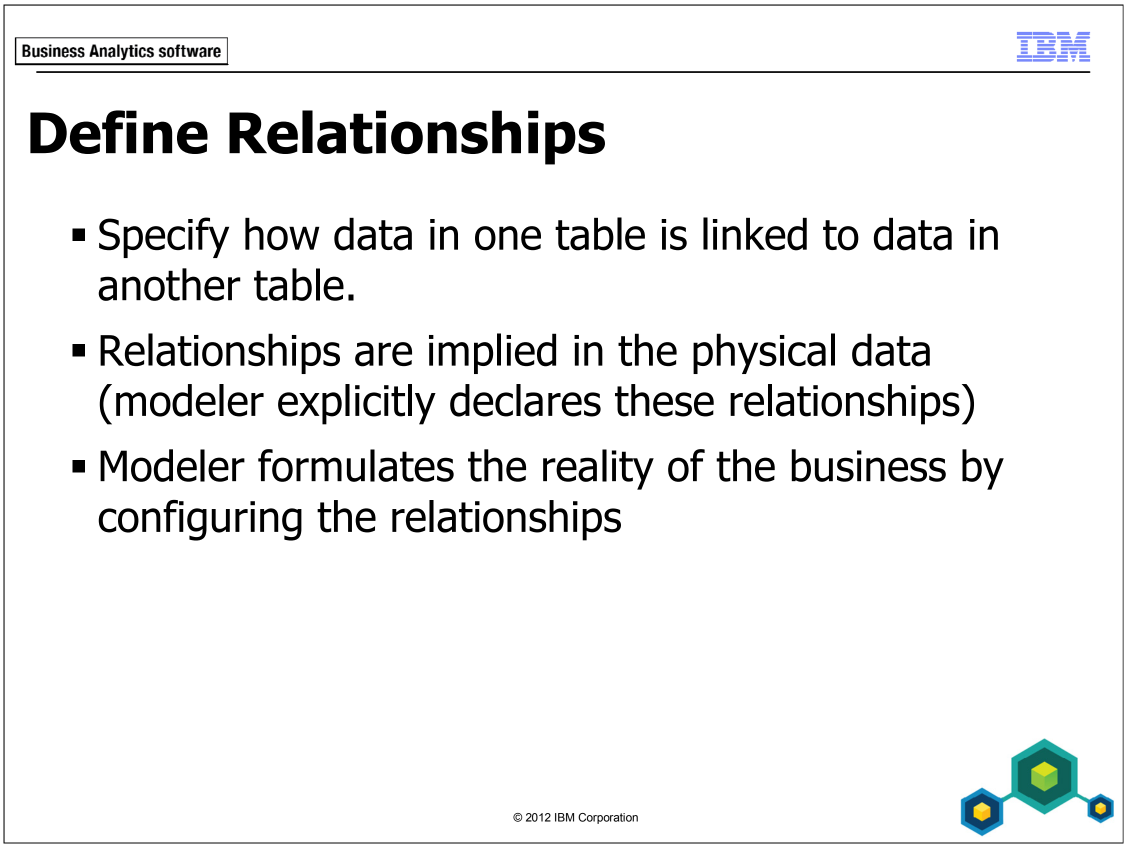
Examine Dimension Tables

- Dimension tables provide descriptive information.
- Dimension tables may be "conformed" so that they are applicable to multiple fact tables across the business.

© 2012 IBM Corporation

A conformed dimension is one which is common to several fact tables; it has the same meaning and content when being referred to from several fact tables. Conformed dimensions prevent "islands of information" by providing context to multiple potential queries.

In the example above, you can query either sales or inventory data through the Product dimension table, the Time dimension table, or both. For example, Product acts as a context when you want to compare quantity sold with stock count.



Business Analytics software

Define Relationships

- Specify how data in one table is linked to data in another table.
- Relationships are implied in the physical data (modeler explicitly declares these relationships)
- Modeler formulates the reality of the business by configuring the relationships

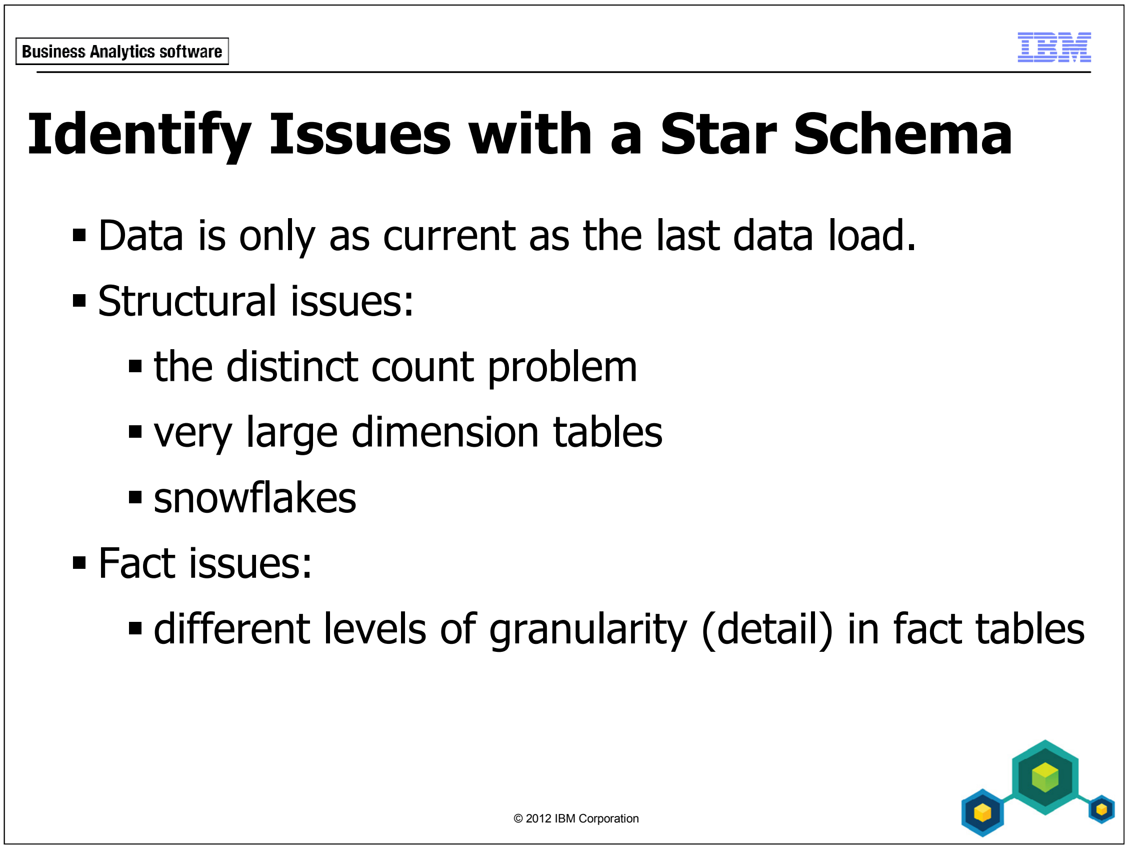
© 2012 IBM Corporation

A relationship states a connection or an operational business rule, such as:

- a sales representative sells a product
- an employee is assigned to a department

Relationships work in both directions. You often have to examine both directions to fully understand the relationship. For example:

- a branch is composed of employees
- an employee may work directly for a branch



Business Analytics software

IBM

Identify Issues with a Star Schema

- Data is only as current as the last data load.
- Structural issues:
  - the distinct count problem
  - very large dimension tables
  - snowflakes
- Fact issues:
  - different levels of granularity (detail) in fact tables

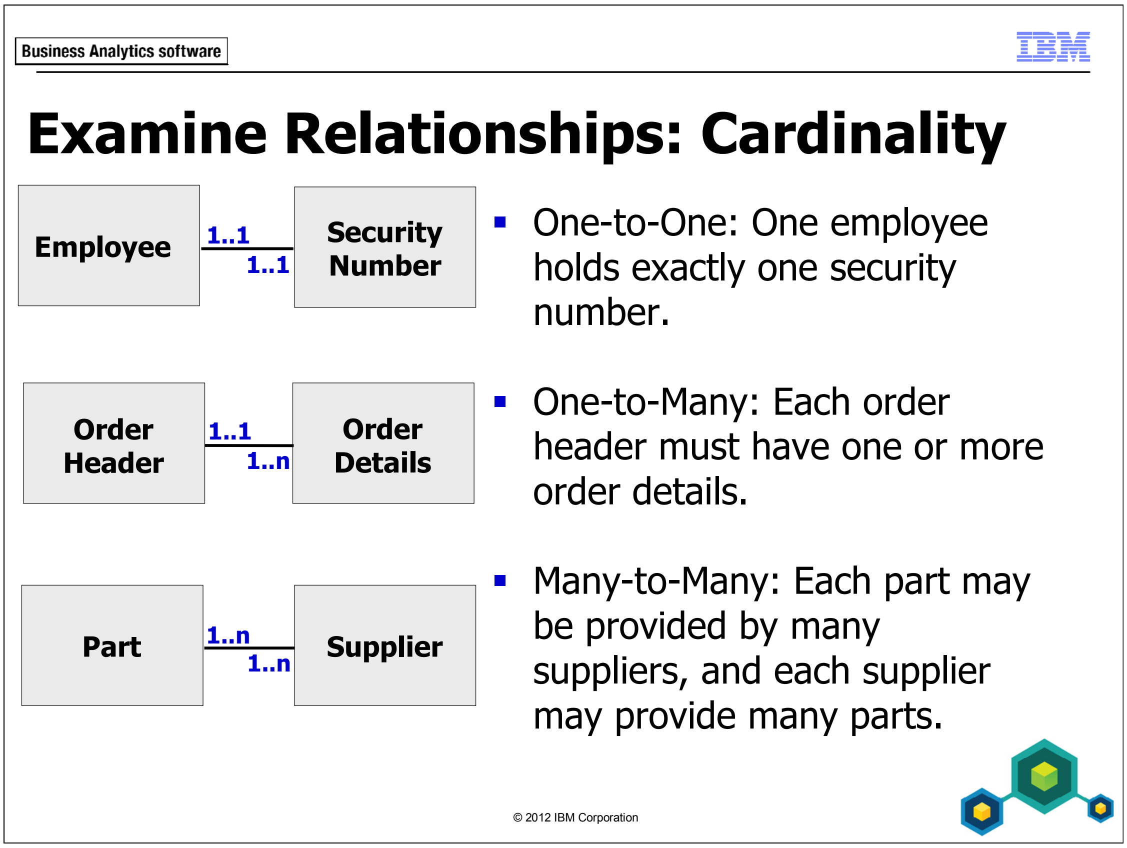
© 2012 IBM Corporation

Unlike an operational database, the data in a reporting database (star schema) is not live. Data is periodically loaded from an operational system into the reporting database (star schema layout). Therefore the data is only as current as the last time it was loaded.

When reporting against a star-schema database, you may encounter difficulties in counting the exact number of items in the dimension, such as separate products.

The star schema may also include large dimension tables that result in reports that run too slowly. These tables may be broken out into smaller tables through a normalization process, which in turn creates snowflake tables.

In some cases, fact tables can have different levels of granularity. For example the data values may either be associated with the day level or the month level. The granularity may change based on the dimensions referenced in the fact table.



Business Analytics software

Examine Relationships: Cardinality

Employee
1..1
1..1
Security Number
Order Header
1..1
1..n
Order Details
Part
1..n
1..n
Supplier

- One-to-One: One employee holds exactly one security number.
- One-to-Many: Each order header must have one or more order details.
- Many-to-Many: Each part may be provided by many suppliers, and each supplier may provide many parts.

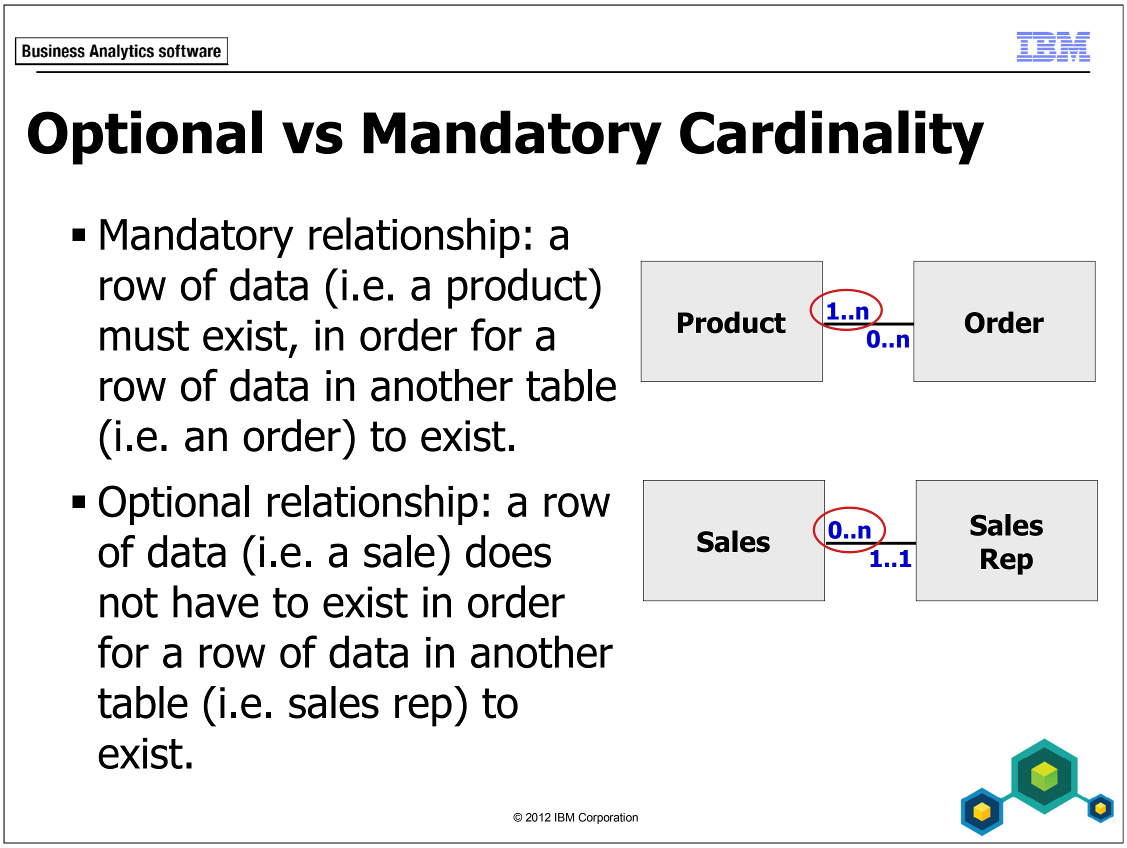
© 2012 IBM Corporation

Cardinality indicates the number of instances of an entity in relation to another entity.

One-to-one relationships occur when one unique row in a table relates to exactly one row in another table. For example, each employee can only have one security number. One security number can only be associated with one employee.

One-to-many relationship example: an order is taken and an Order Header table is populated with data such as date, customer name, and sales staff name. This table is related to an Order Details table that contains data about individual items sold in that one order, such as order detail code, product number, and quantity. Therefore a relationship exists between Order Header and Order Details, whereby each Order Header must contain one or many Order Details, and each Order Detail must appear on one and only one Order Header.

Many-to-many relationships occur when many unique rows in a table relate to many rows in another table. The slide example shows that many suppliers may provide a single part. However, a single supplier may provide many parts.



Business Analytics software

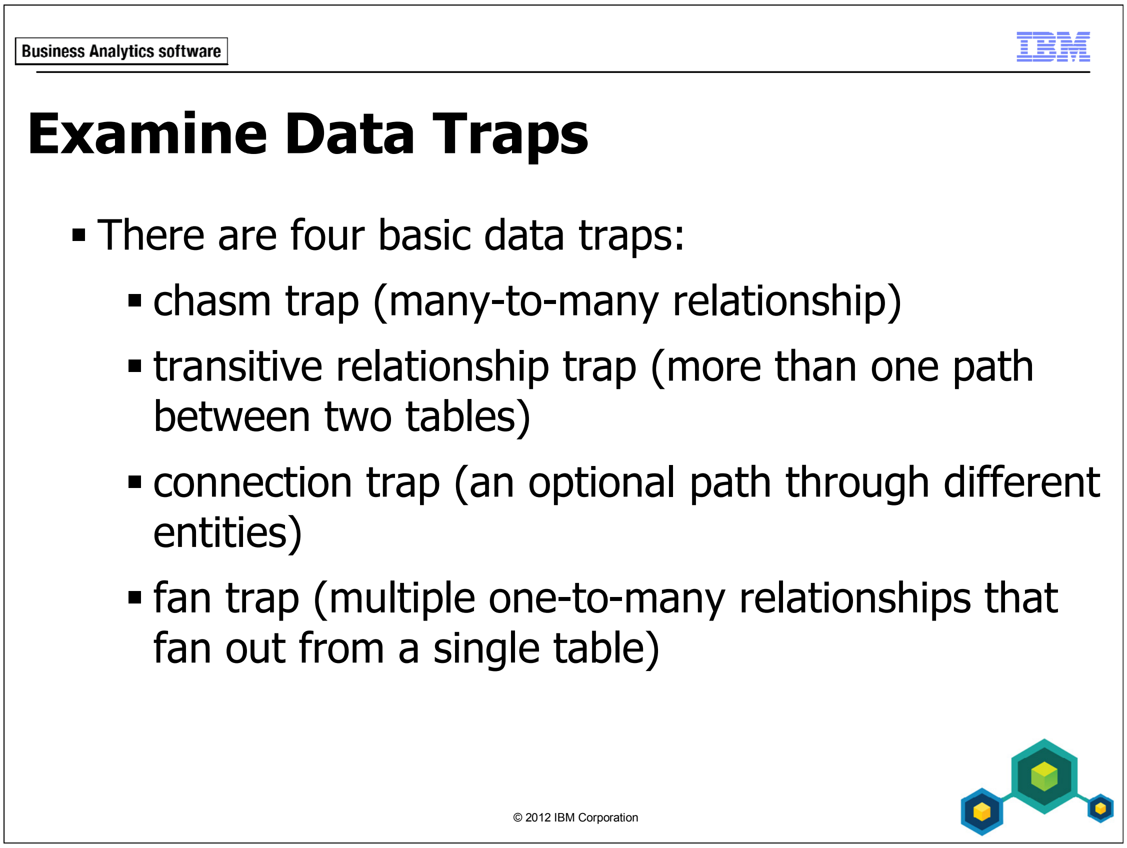
Optional vs Mandatory Cardinality

- Mandatory relationship: a row of data (i.e. a product) must exist, in order for a row of data in another table (i.e. an order) to exist.
- Optional relationship: a row of data (i.e. a sale) does not have to exist in order for a row of data in another table (i.e. sales rep) to exist.

Product
1..n
0..n
Order
Sales
0..n
1..1
Sales Rep

You can specify an optional relationship (minimum cardinality of 0) when you want the query to retain the information on the other side of the relationship in the absence of a match. An optional relationship generates an outer join, which results in a null value when there is no data for one table that matches a row of data in another table.

Make sure that you only define optional relationships when required, as generating outer joins can negatively affect performance.



Business Analytics software

IBM

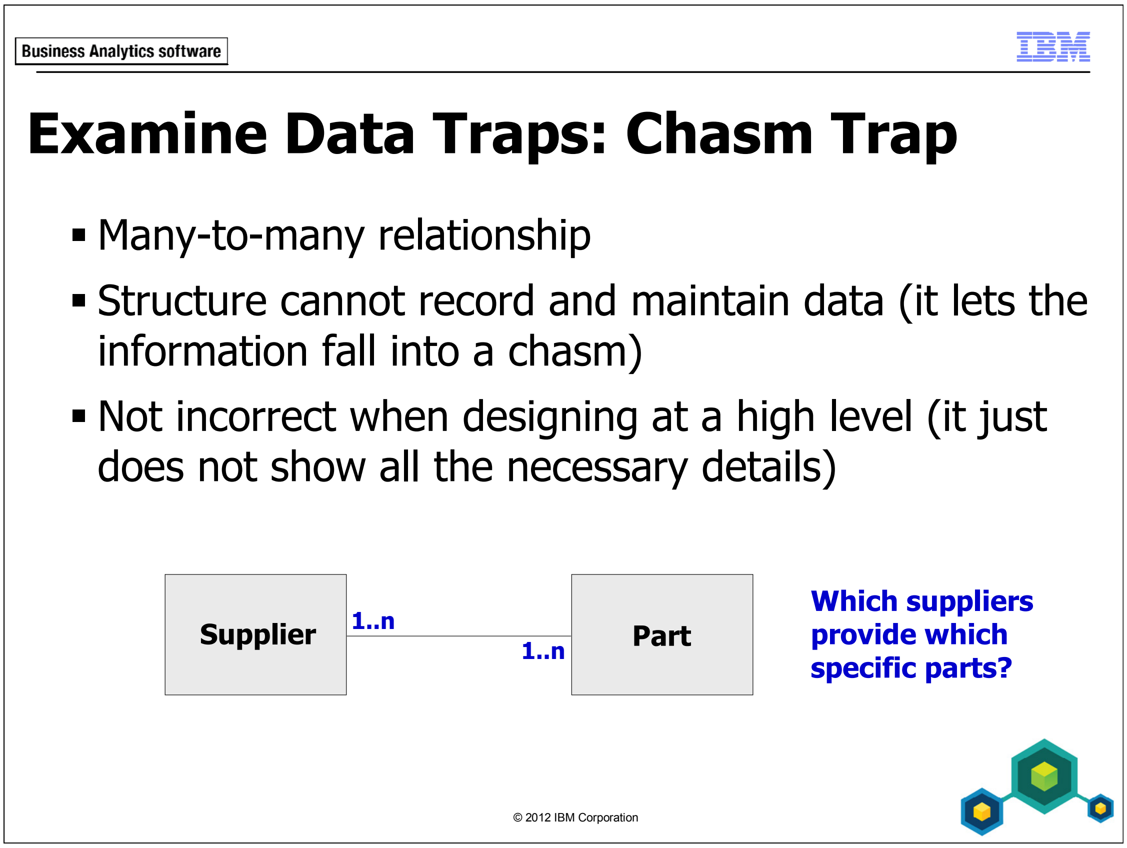
Examine Data Traps

- There are four basic data traps:
  - chasm trap (many-to-many relationship)
  - transitive relationship trap (more than one path between two tables)
  - connection trap (an optional path through different entities)
  - fan trap (multiple one-to-many relationships that fan out from a single table)

© 2012 IBM Corporation

These are data modeling traps, not metadata modeling traps, so you cannot use Framework Manager to fix them in the data source. However it is useful to know of them so that you can make metadata model designs which can handle them and generate the appropriate SQL at run time to provide predictable results. A data trap does not necessarily indicate a problem, only that an area is worth inspection and possible refinement.

The data modeler must carefully examine any area that does not appear to represent the data completely.



Business Analytics software

Examine Data Traps: Chasm Trap

- Many-to-many relationship
- Structure cannot record and maintain data (it lets the information fall into a chasm)
- Not incorrect when designing at a high level (it just does not show all the necessary details)

Supplier
1..n
1..n
Part

Which suppliers provide which specific parts?

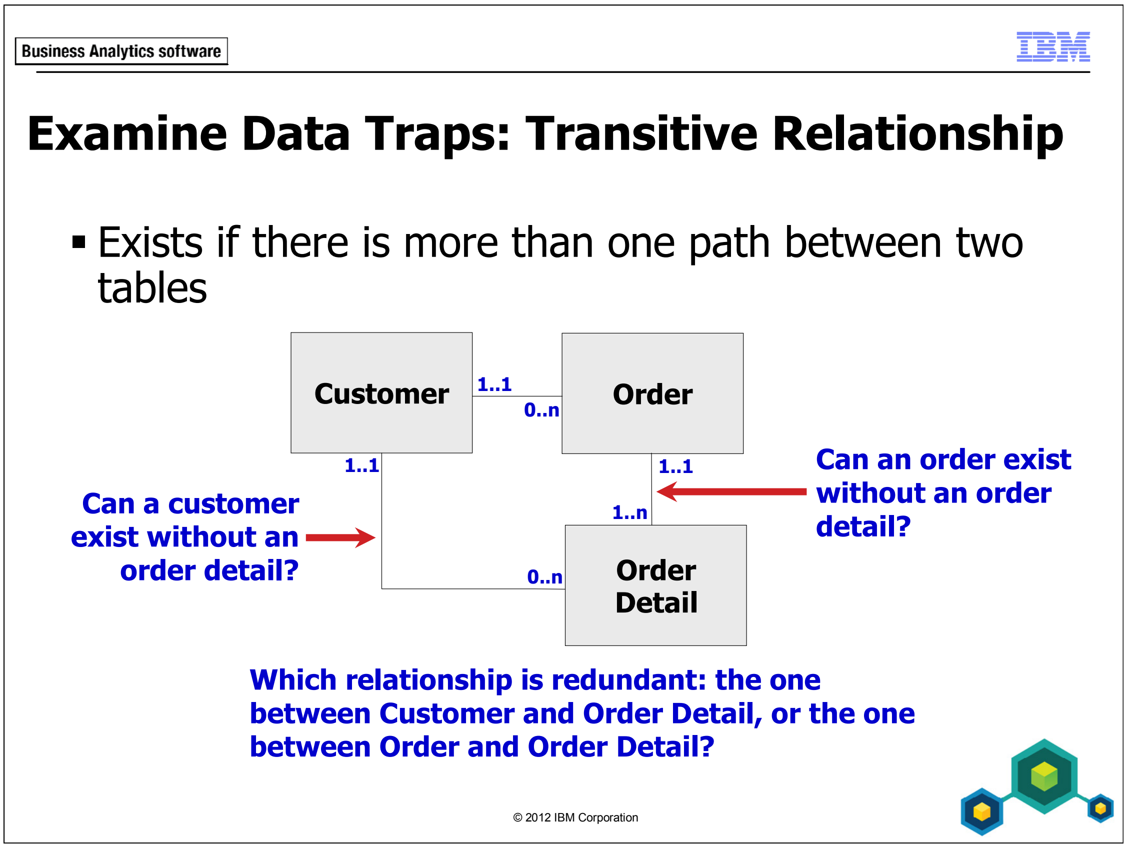
© 2012 IBM Corporation

In a chasm trap, more than one row in a table is related to more than one row in another table.

The slide example shows that a single supplier may provide many parts, and many suppliers may provide a single part.

If each supplier can potentially supply every single part, how do you report on the suppliers that provide specific parts?

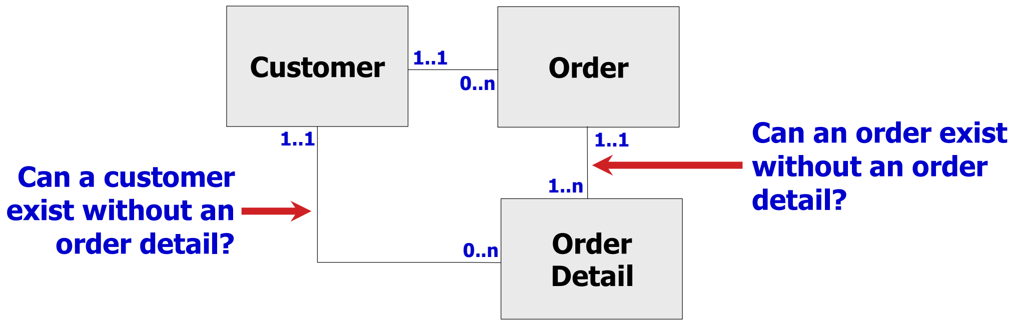
This is typically resolved with a "bridge" table that records the details of the relationship between the two tables.



Business Analytics software

Examine Data Traps: Transitive Relationship

- Exists if there is more than one path between two tables


Customer
Order
Order Detail
1..1
0..n
1..1
1..1
1..n
0..n
Can a customer exist without an order detail?
Can an order exist without an order detail?

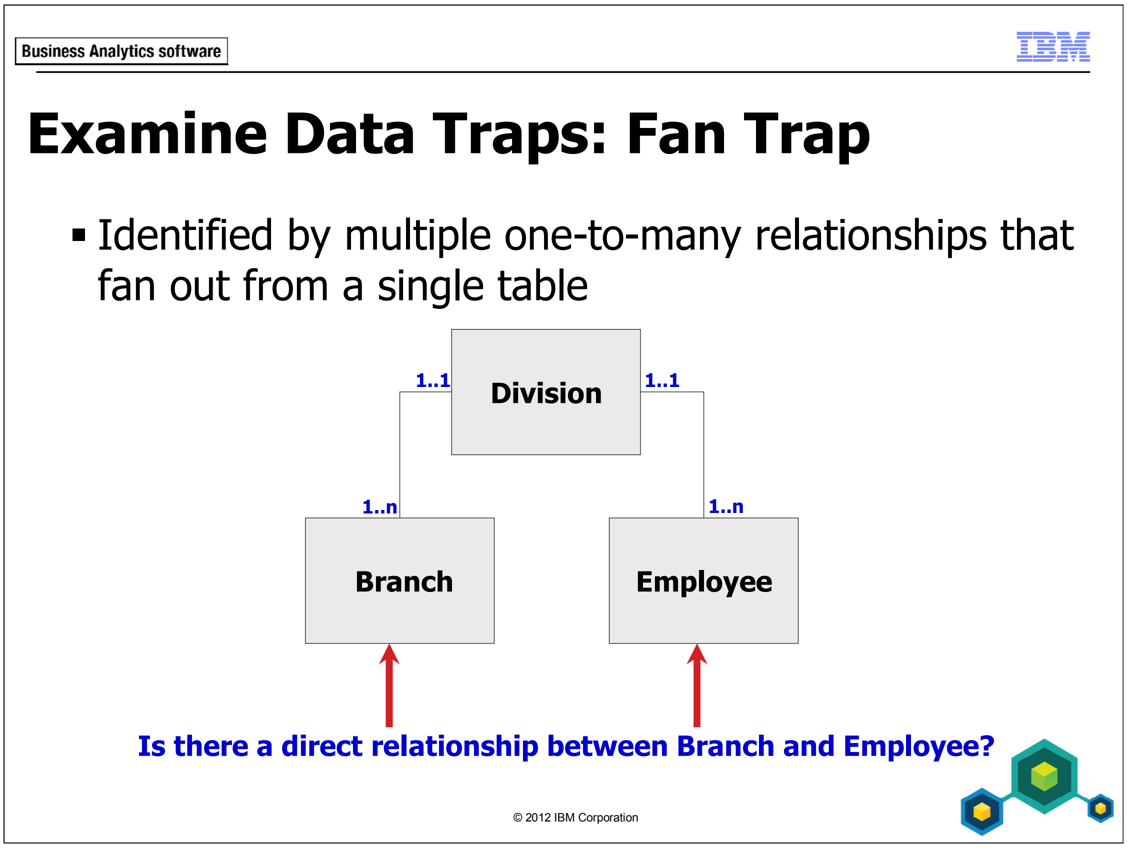
Which relationship is redundant: the one between Customer and Order Detail, or the one between Order and Order Detail?

© 2012 IBM Corporation

A transitive relationship trap resembles a wheel shape.

This kind of trap may make it difficult to write queries that retrieve the appropriate data, because it may not be clear which table columns must be included in the query.

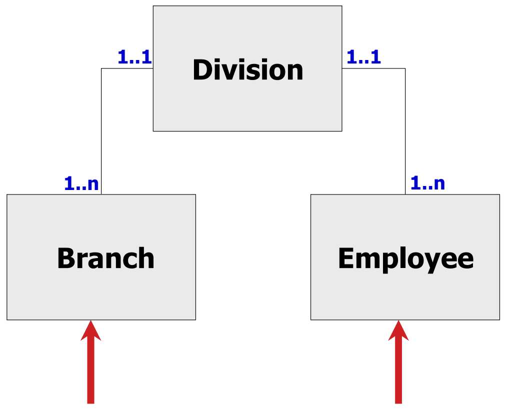
Going through either path produces some sort of result, but which path is the correct or more efficient one for your specific query?



Business Analytics software

Examine Data Traps: Fan Trap

- Identified by multiple one-to-many relationships that fan out from a single table



Is there a direct relationship between Branch and Employee?

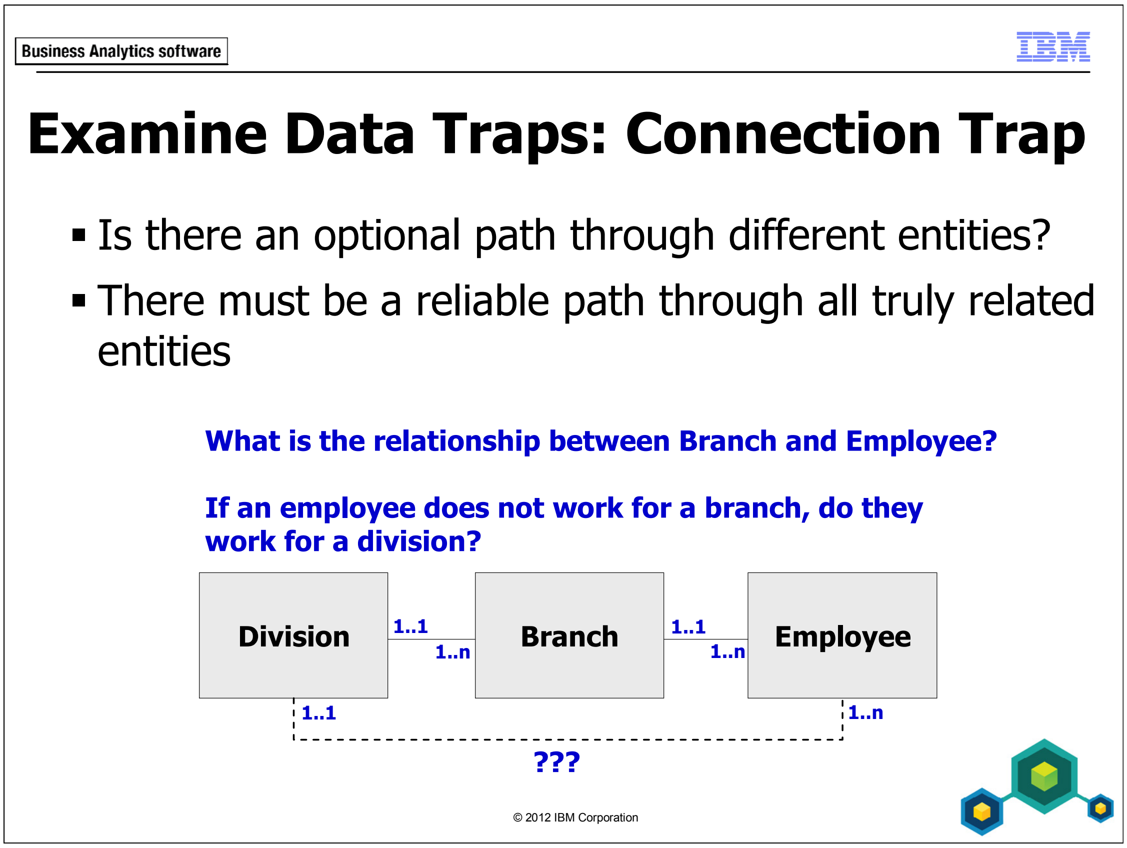
© 2012 IBM Corporation

Fan traps are also known as parallel relationships.

In this kind of trap, a table has two one-to-many relationships spreading out from it, implying that the two other tables have no connection to each other.

The modeler should examine such traps to determine whether a crucial relationship is missing.

Branch may actually have a direct relationship to Employee.

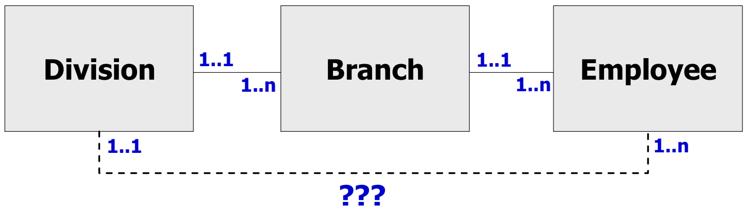

Business Analytics software

Examine Data Traps: Connection Trap

- Is there an optional path through different entities?
- There must be a reliable path through all truly related entities

What is the relationship between Branch and Employee?

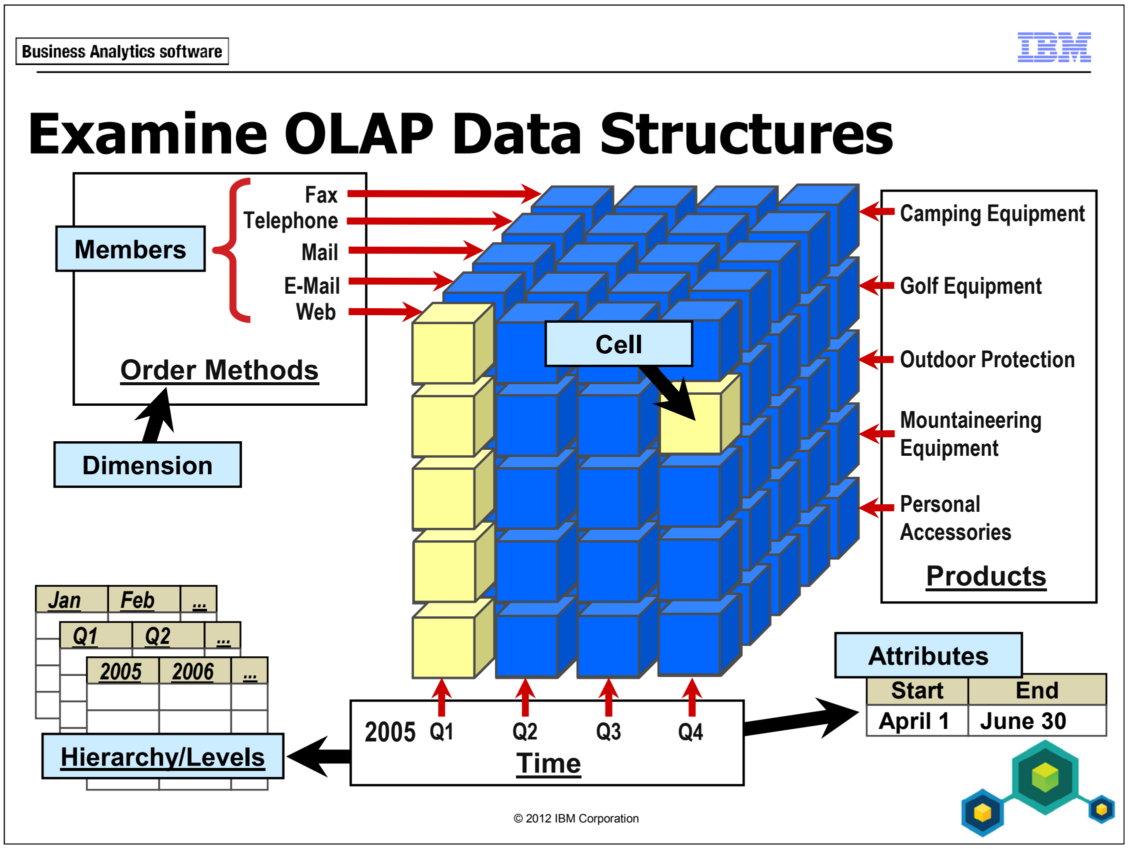
If an employee does not work for a branch, do they work for a division?



© 2012 IBM Corporation

The slide example suggests that there may be a direct relationship between the Employee and Division tables.

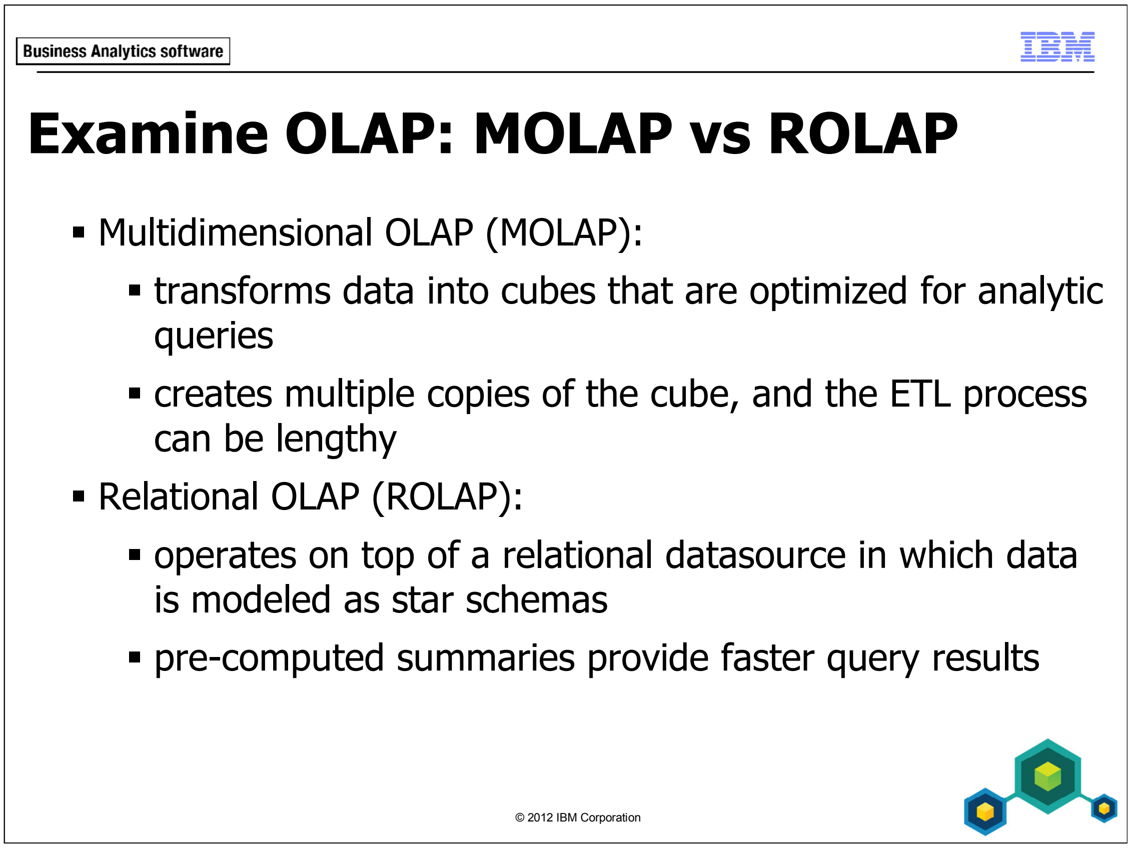
An employee may be able to work directly for a division, instead of working for a branch within a division. For example, the employee may work from a home office.



OLAP is an alternative data access strategy to modeling metadata for Relational models The OLAP structure consists of the following elements:

- **dimensions** - contain members, which may be structured into hierarchies and levels
- **hierarchies** - provide context to the level structures they contain
- **levels** - provide structure for the members of a hierarchy
- **members** - data entities that provide context to cell values
- **attributes** - provide additional information about members
- **cells** - are intersection points containing values (measures) for various members from different dimensions (also referred to as tuples)

Tools that can be used to build an OLAP data structure include Cognos PowerPlay Transformer and SAP BW. The reporting traps just discussed should already have been modeled out in this kind of structure. The data represents a snap shot in time and must be periodically updated.



Business Analytics software

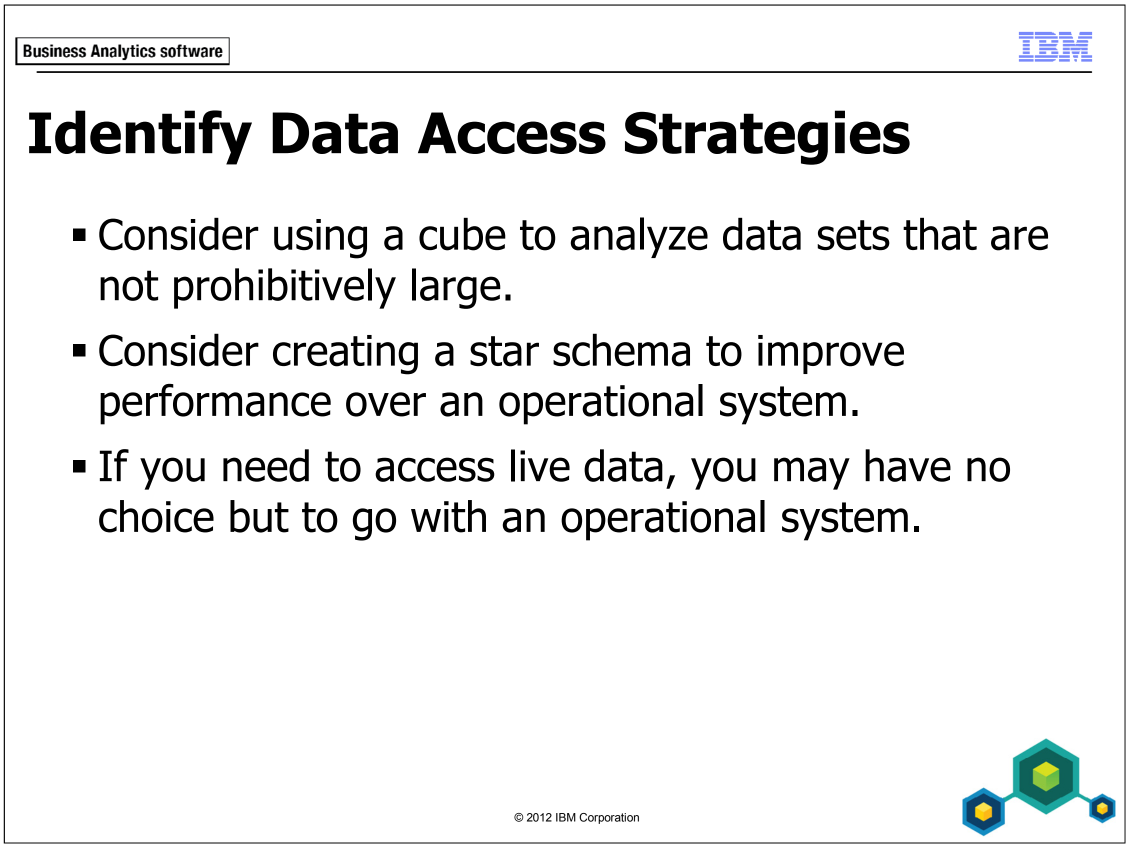
Examine OLAP: MOLAP vs ROLAP

- Multidimensional OLAP (MOLAP):
  - transforms data into cubes that are optimized for analytic queries
  - creates multiple copies of the cube, and the ETL process can be lengthy
- Relational OLAP (ROLAP):
  - operates on top of a relational datasource in which data is modeled as star schemas
  - pre-computed summaries provide faster query results

© 2012 IBM Corporation

ROLAP is optimized to work with large data sets. It uses aggregate tables to improve the speed at which large queries are executed.

You can design a ROLAP cube by using the IBM Cognos Dynamic Cube Designer., which is outside the scope of this course.



Business Analytics software

IBM

Identify Data Access Strategies

- Consider using a cube to analyze data sets that are not prohibitively large.
- Consider creating a star schema to improve performance over an operational system.
- If you need to access live data, you may have no choice but to go with an operational system.

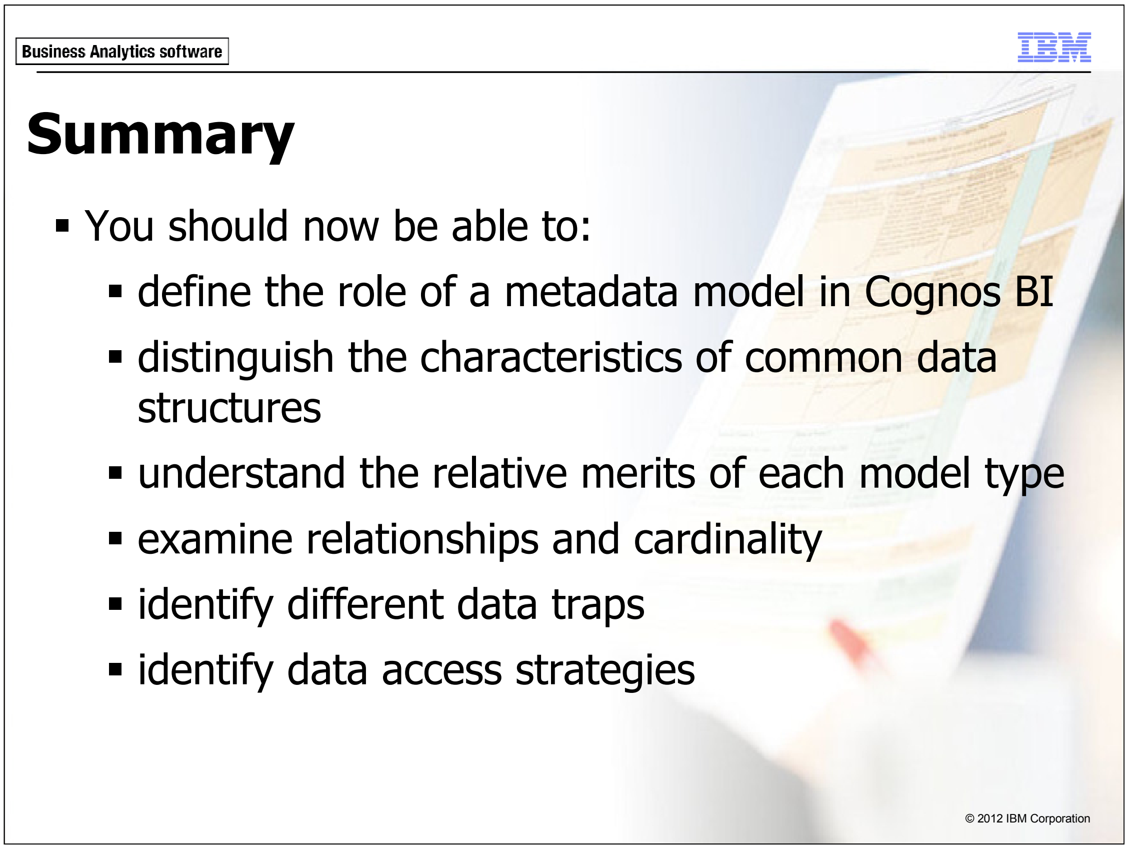
© 2012 IBM Corporation

The methods that you use to access your data for reporting purposes will be determined by your needs and by the structure and content of the data itself.

What are acceptable performance times for your reports? If the information is required quickly (within seconds), consider using cubes. However, if the data set is extremely large, cubes may not be the solution and you may want to consider using a data mart or warehouse with pre-aggregated tables.

If you require live data, you may need to go directly against your operational system. Remember that reporting directly against a normalized operational system may yield poor performance and may slow down the writing of data to the database.

Planning and scope should be given a great deal of attention before embarking on a business intelligence project.



Business Analytics software

Summary

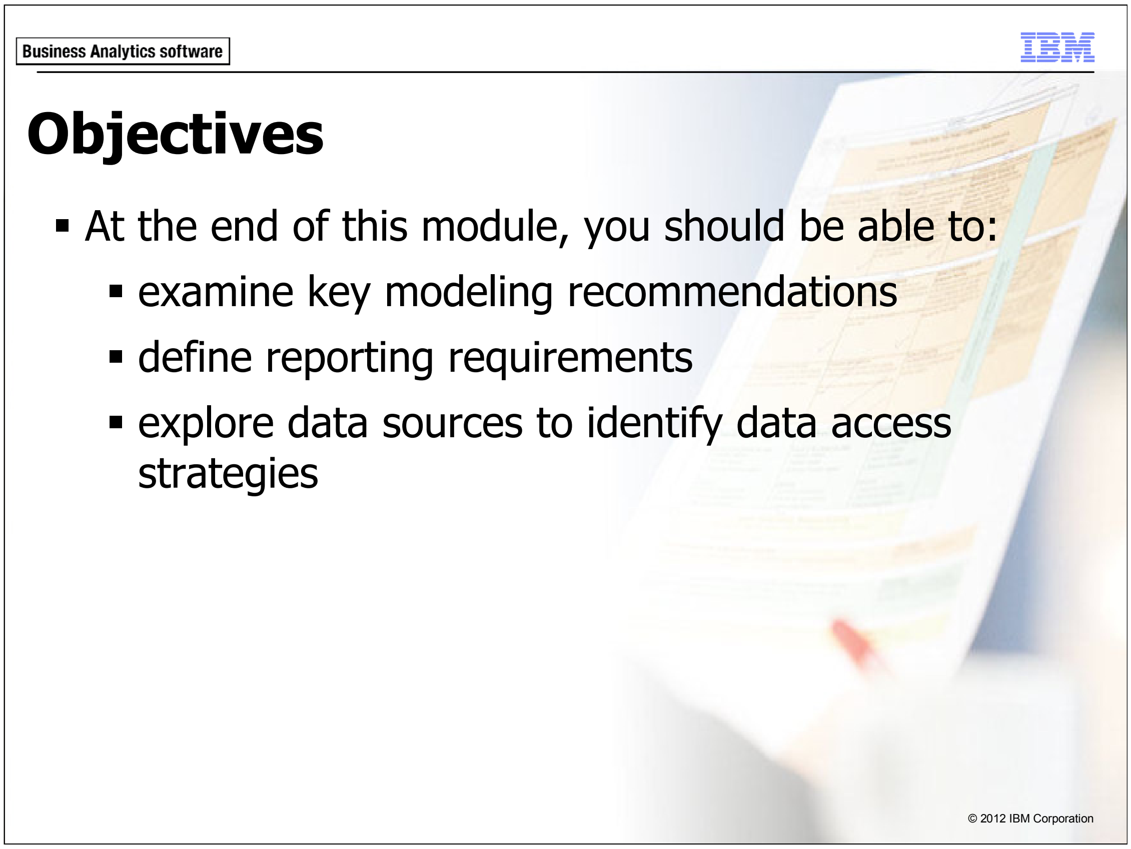
- You should now be able to:
  - define the role of a metadata model in Cognos BI
  - distinguish the characteristics of common data structures
  - understand the relative merits of each model type
  - examine relationships and cardinality
  - identify different data traps
  - identify data access strategies

© 2012 IBM Corporation

Gather Requirements

IBM Cognos BI

Business Analytics software



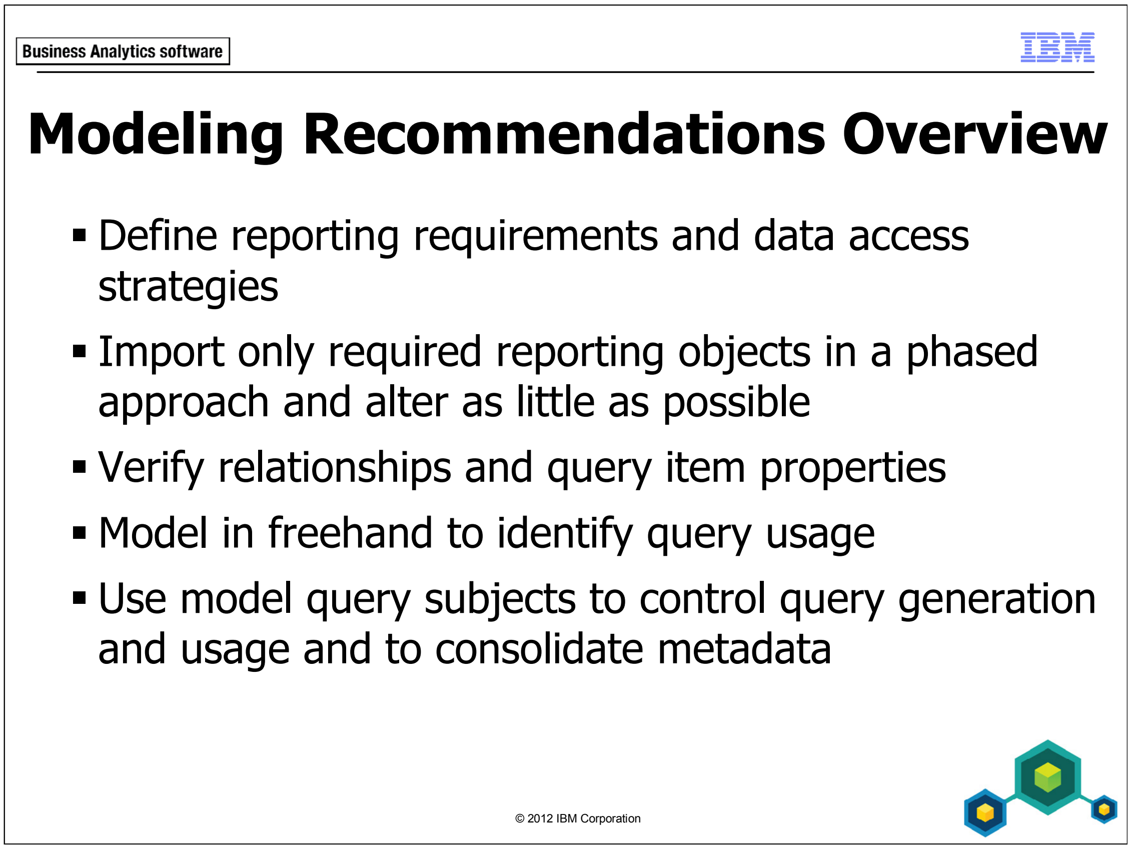
Business Analytics software

Objectives

- At the end of this module, you should be able to:
  - examine key modeling recommendations
  - define reporting requirements
  - explore data sources to identify data access strategies

© 2012 IBM Corporation

Before reviewing this module, you should review the following modules:
• Overview of IBM Cognos
• Identify Common Data Structures



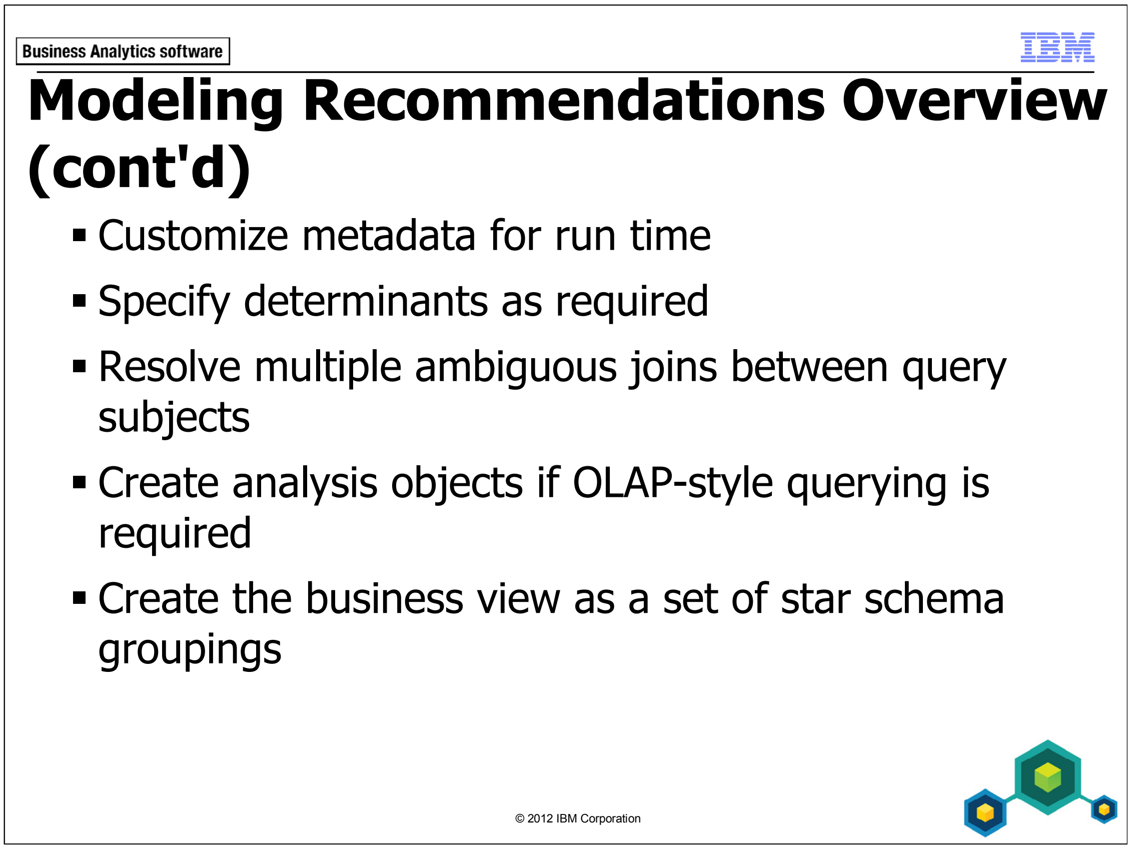
Business Analytics software

Modeling Recommendations Overview

- Define reporting requirements and data access strategies
- Import only required reporting objects in a phased approach and alter as little as possible
- Verify relationships and query item properties
- Model in freehand to identify query usage
- Use model query subjects to control query generation and usage and to consolidate metadata

© 2012 IBM Corporation

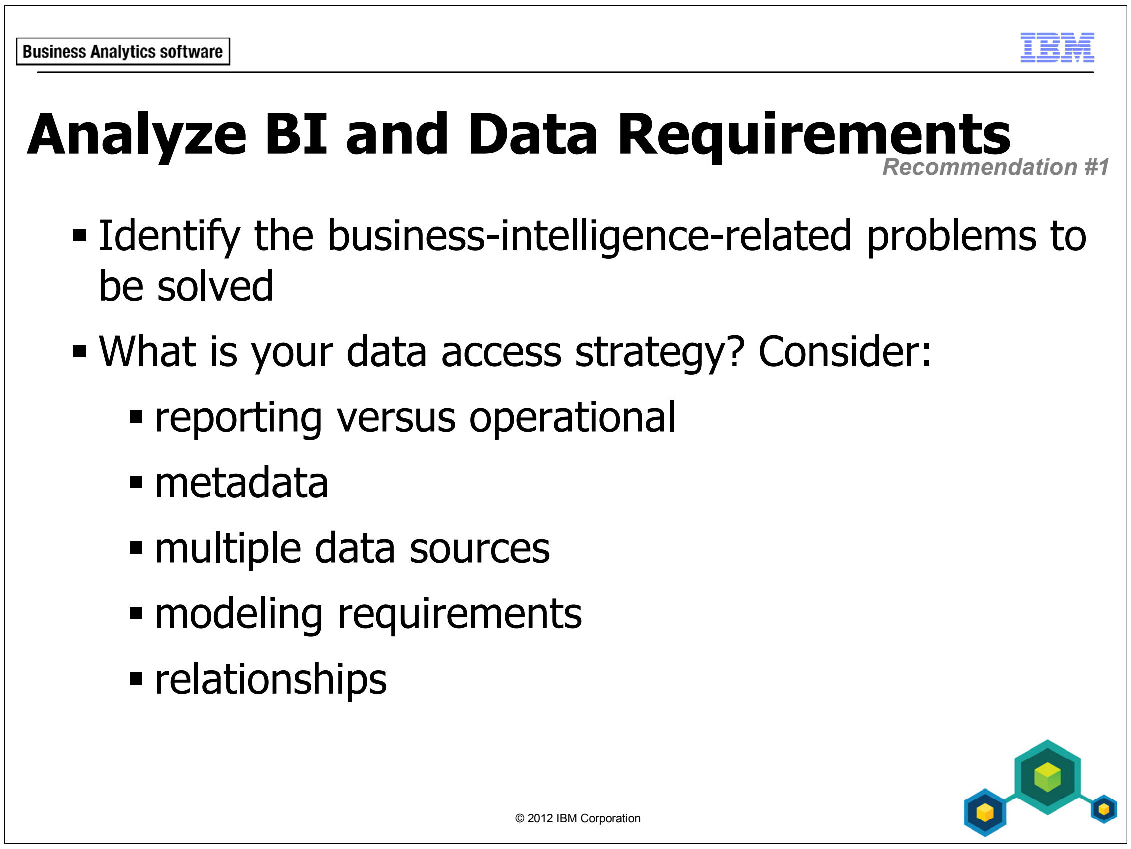
These recommendations will be expanded upon as you progress through the course.



Business Analytics software

Modeling Recommendations Overview (cont'd)

- Customize metadata for run time
- Specify determinants as required
- Resolve multiple ambiguous joins between query subjects
- Create analysis objects if OLAP-style querying is required
- Create the business view as a set of star schema groupings



Business Analytics software

Analyze BI and Data Requirements

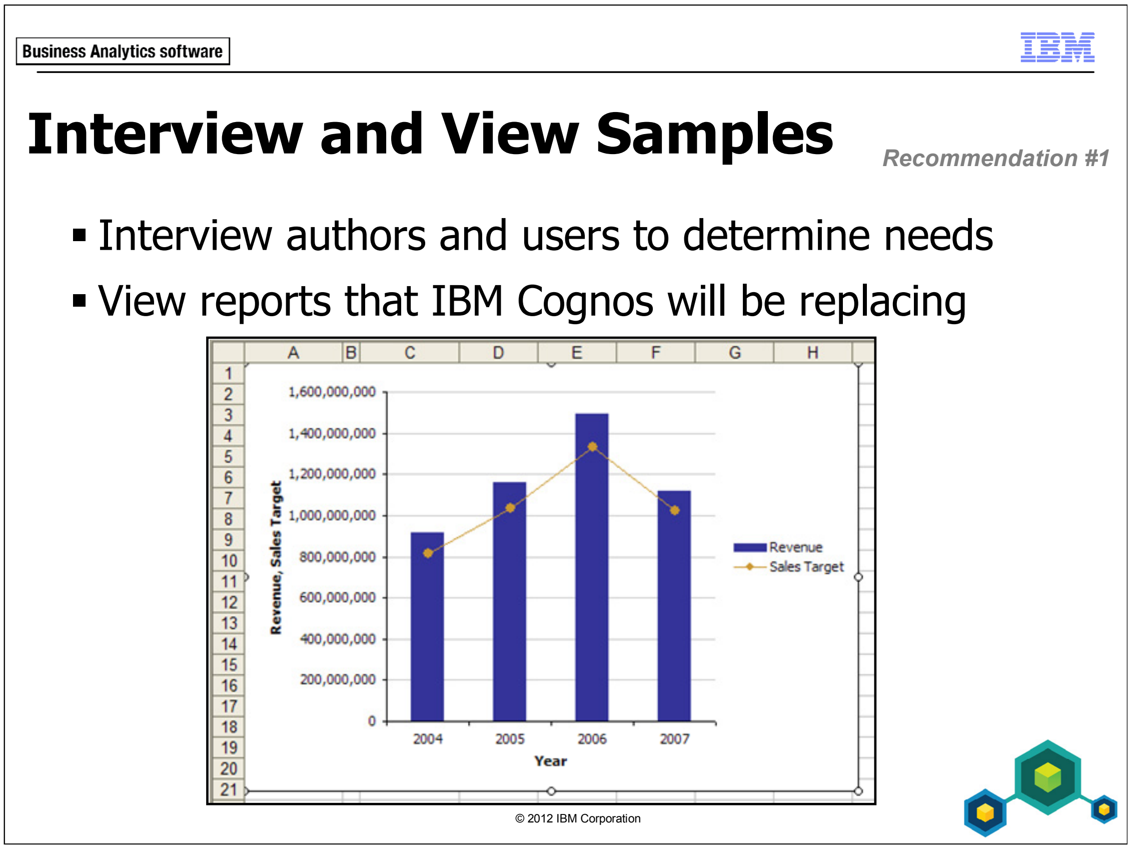
Recommendation #1

- Identify the business-intelligence-related problems to be solved
- What is your data access strategy? Consider:
  - reporting versus operational
  - metadata
  - multiple data sources
  - modeling requirements
  - relationships

© 2012 IBM Corporation

Problems to be solved include setting the scope of your project and setting its methodology, as well as multilingualism, performance, security, and presentation. You should ask questions such as:

- Do you and the IBM Cognos users agree on the model requirements?
- Does the data source contain the data and metadata you need?
- Does the same data exist in more than one source?
- Which data source tables are the fact tables, which are the dimensions, and which are both fact tables and dimensions? What are the keys and attributes of each dimension?
- Do fact tables contain only facts and foreign keys? Do they also contain dimensional attributes that should be in dimension tables?
- What are the required relationships and are there multiple paths between tables?

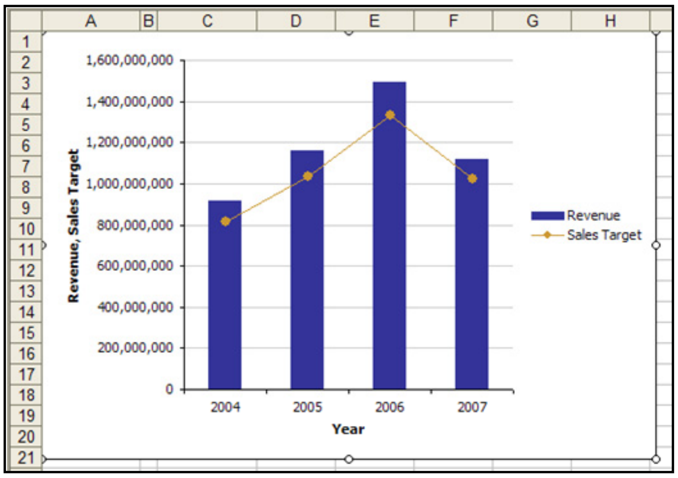


Business Analytics software

Interview and View Samples

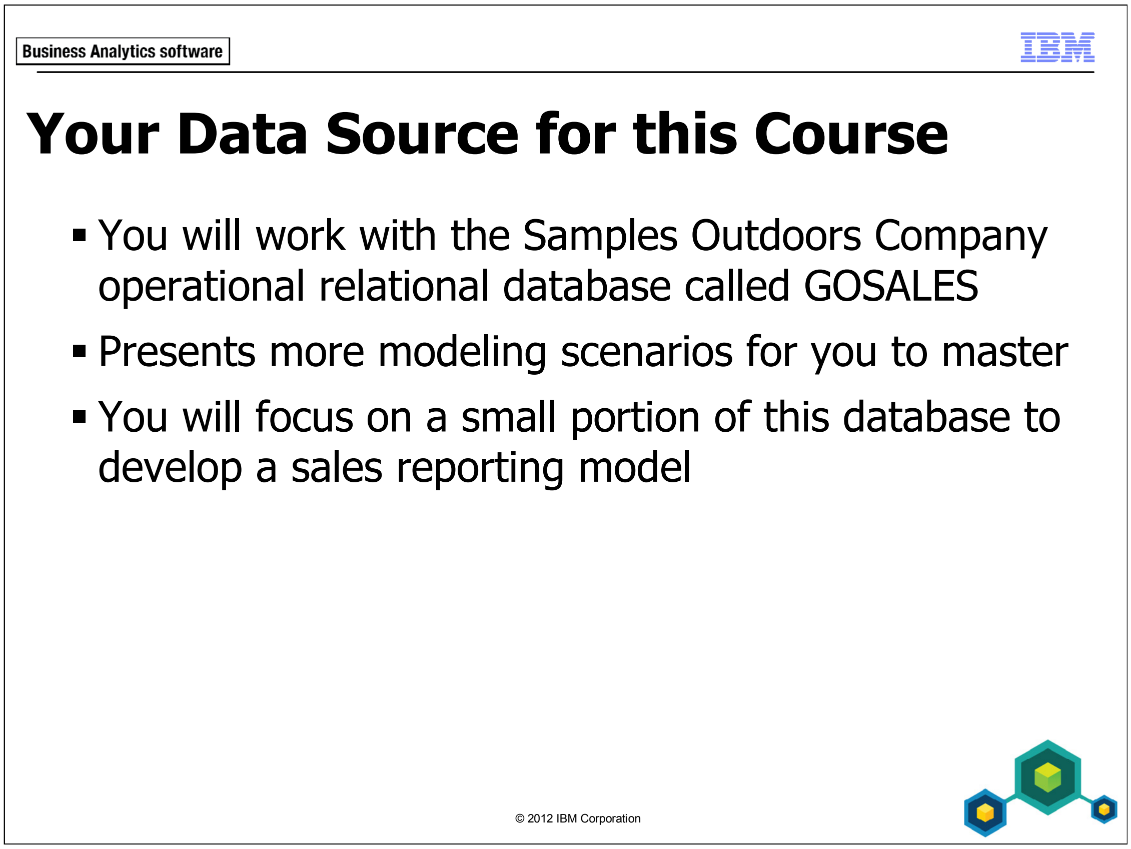
Recommendation #1

- Interview authors and users to determine needs
- View reports that IBM Cognos will be replacing



© 2012 IBM Corporation

Ralph Kimball (author on the subject of data warehousing and business intelligence) recommends an interview-based approach to determine report requirements. The focus is on what information the key business decision makers require to do their jobs. This will result in a framework of key performance indicators (KPI) and business contexts.



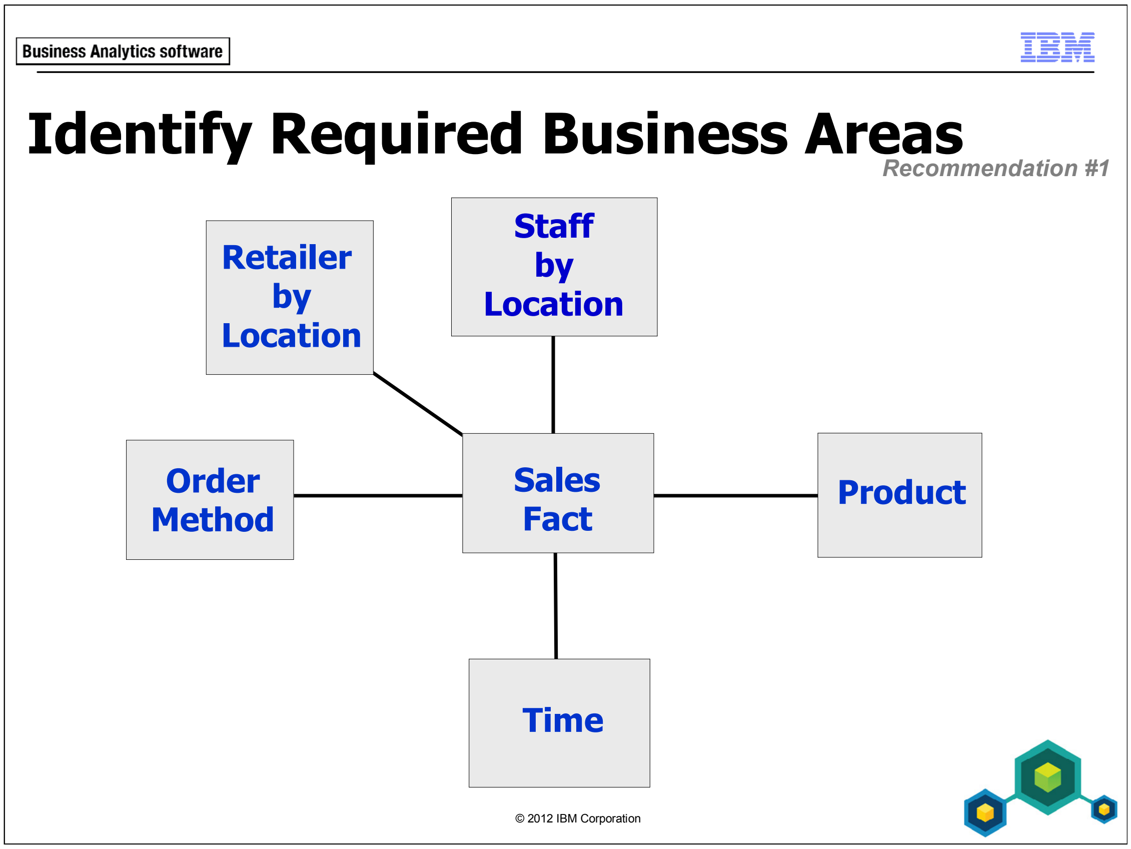
Business Analytics software

Your Data Source for this Course

- You will work with the Samples Outdoors Company operational relational database called GOSALES
- Presents more modeling scenarios for you to master
- You will focus on a small portion of this database to develop a sales reporting model

© 2012 IBM Corporation

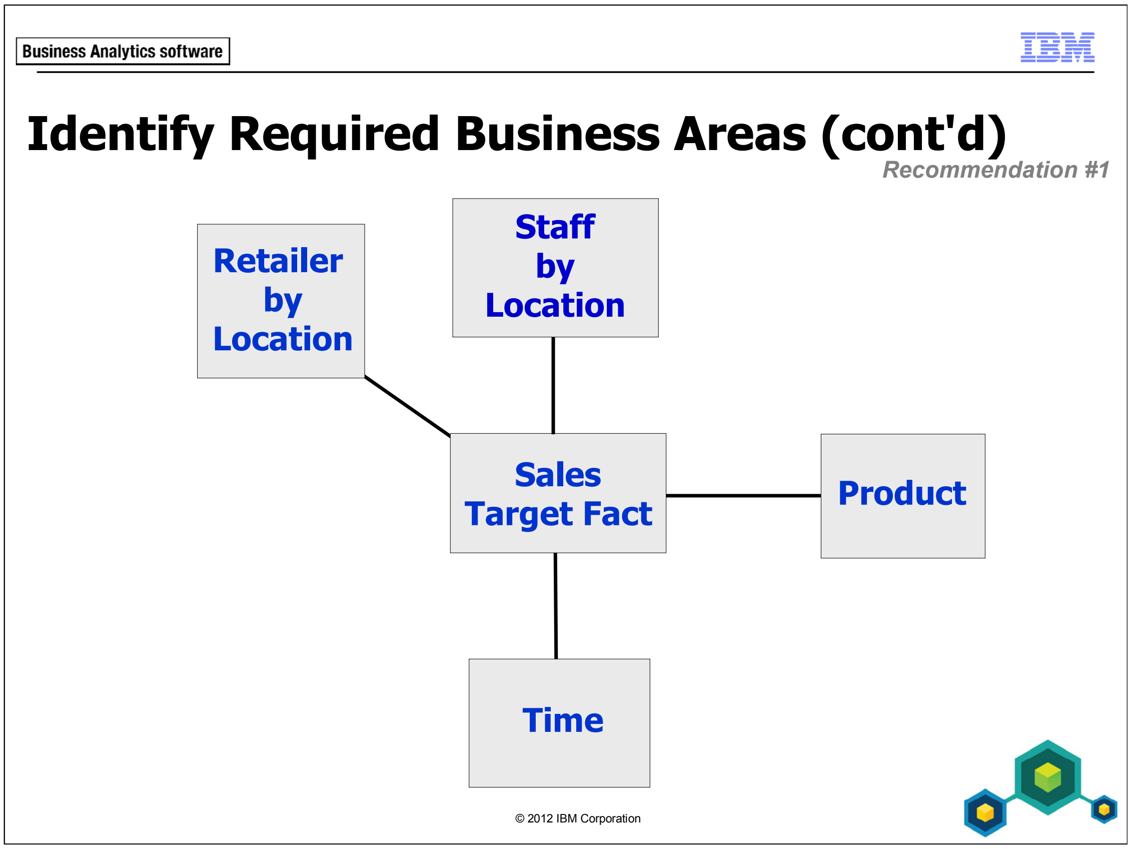
Although a reporting (star schema) database is recommended for performance and ease of use, in this course you will model an operational system to provide you with a wider variety of modeling techniques and prepare you for more data scenarios back at your place of work.



Before creating a Framework Manager project, you must:

- analyze and understand your reporting requirements based on the business process that authors will be reporting on
- understand your data and the structure of your data source(s)

Upon interviewing BI users, you discover that they would like to report on sales data by the dimensions shown in the slide example above.



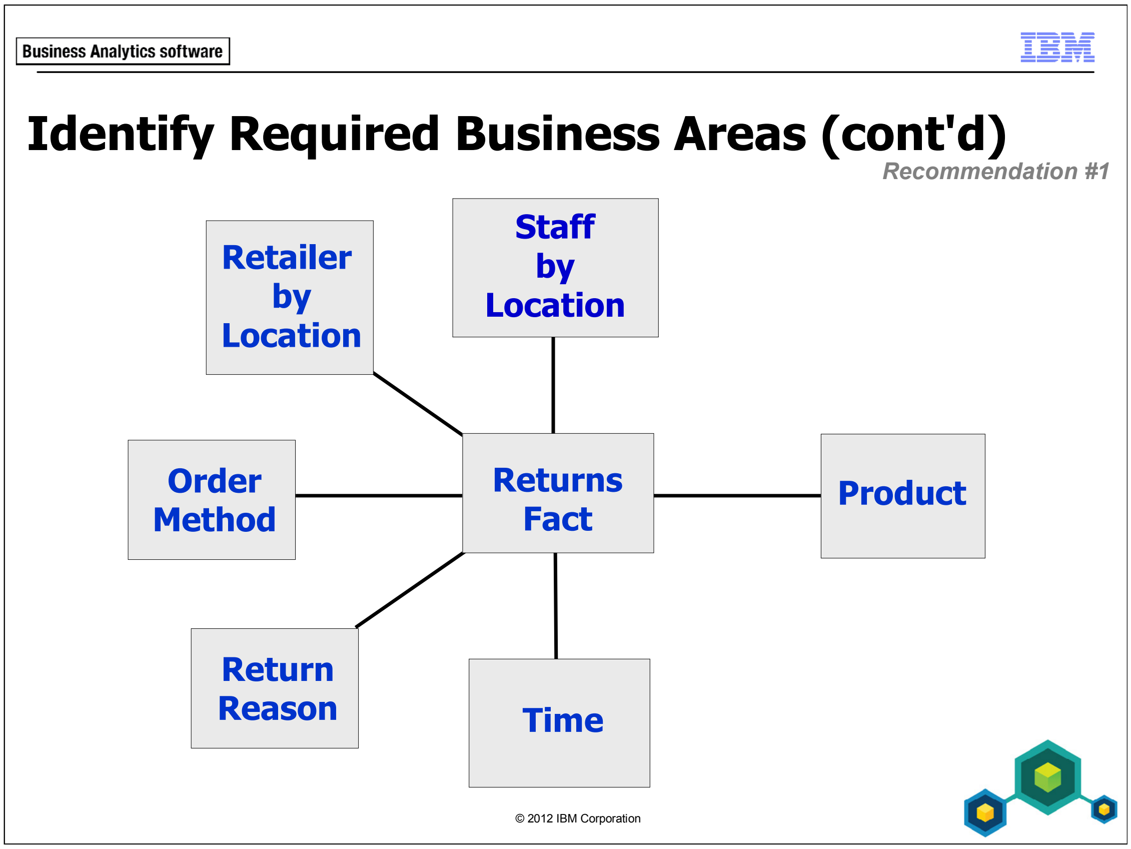
Business Analytics software

Identify Required Business Areas (cont'd)

Recommendation #1

© 2012 IBM Corporation

BI users would also like to report on sales targets by the dimensions shown above.



Business Analytics software

Identify Required Business Areas (cont'd)

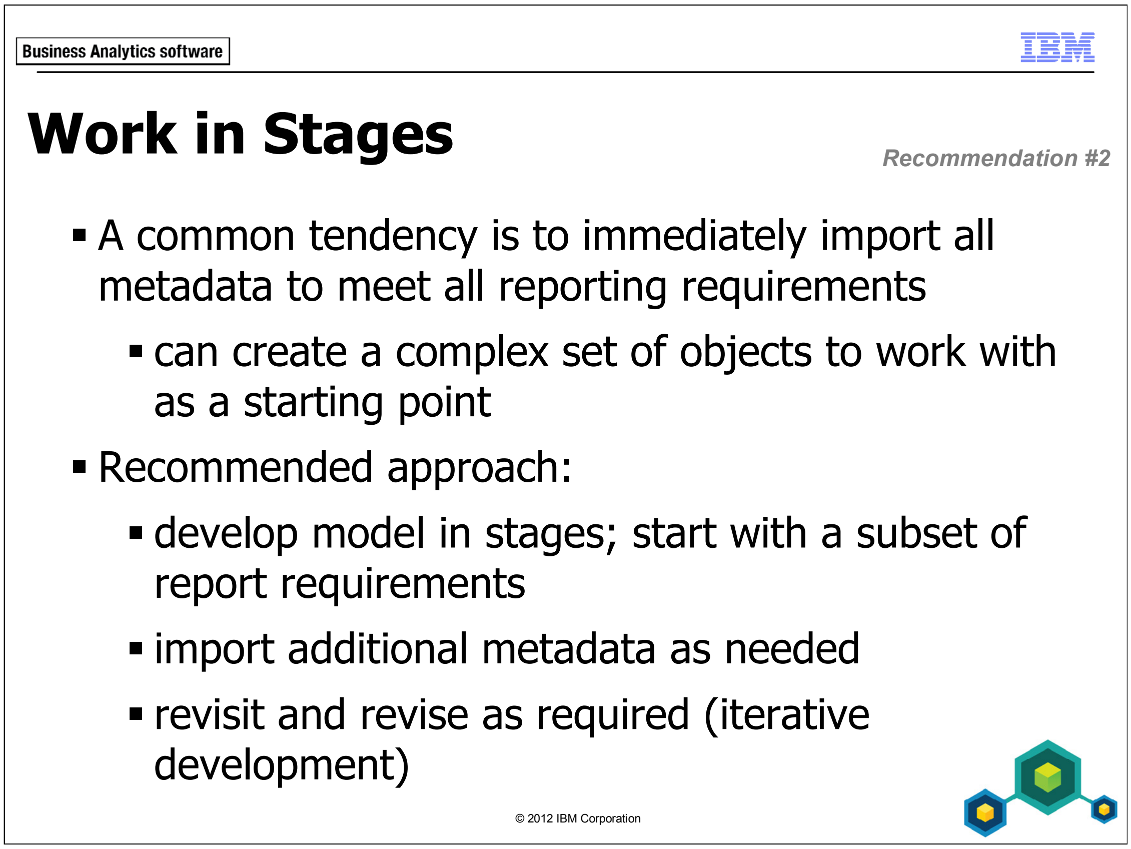
Recommendation #1

© 2012 IBM Corporation

And finally, BI users would like to report on returns by the dimensions shown above.

Knowing this information, you interview the database administrator to find out where this information can be obtained. You are told it is in the GOSALES database.

The B5252-Requirements_Handout.doc file (in C:\Edcognos\B5252\Instructor Files) identifies the dimensions that relate to each fact in the GOSALES database, along with diagrams that give a visual representation of these relationships in a star schema format. This table includes more dimensions than are presented in this module, since additional logical dimensions will be discovered throughout the modeling process. It will be a useful reference as you work their way through the course.



Business Analytics software

Work in Stages

Recommendation #2

- A common tendency is to immediately import all metadata to meet all reporting requirements
  - can create a complex set of objects to work with as a starting point
- Recommended approach:
  - develop model in stages; start with a subset of report requirements
  - import additional metadata as needed
  - revisit and revise as required (iterative development)

© 2012 IBM Corporation

For your course model, the scenario is that you have interviewed functional users and have determined that you will start with a baseline project that allows reporting on sales by product or by order method. This postpones work on sales staff, location, retailer (your customers), sales targets, and returns.

As you progress in your modeling activities, you may discover other requirements or the data itself may present you with other reporting options that were not thought of by the end users.

Demo 1: Identify Data Sources

Purpose:
As the Data Modeler for The Sample Outdoors Company, your task is to develop a model that supports the business requirements of report and ad hoc query authors working with Cognos BI. Before you begin the modeling process, you will locate the data sources identified as necessary to meet those reporting requirements.

Component: **IBM DB2 Control Center**

Task 1. Open DB2 Control Center.

1. From the **Start** menu, point to **All Programs** > **IBM DB2** > **DB2COPY1 (Default)** > **General Administration Tools**, and then click **Control Center**.
2. Click **OK** in the **Control Center View** dialog.
3. Expand **All Databases** > **GS_DB**, and then click the **Tables** folder.

Task 2. Investigate Tables.

The sales reports will be focused on orders, so note the ORDER_DETAILS, ORDER_HEADER, and ORDER_METHOD tables.

1. Click **ORDER_HEADER** to view details for the ORDER_HEADER table in the Details pane.

   You can identify the tables related to the ORDER_HEADER header table by looking for foreign keys. Here you can see several keys: ORDER_NUMBER, RETAILER_SITE_CODE, RETAILER_CONTACT_CODE, SALES_STAFF_CODE, SALES_BRANCH_CODE, and ORDER_METHOD_CODE. You will now examine some of these relationships.

2. Click **ORDER_DETAILS** to view details for the ORDER_DETAILS table in the Details pane.

   You can see the facts needed for many reports: QUANTITY, UNIT_COST, UNIT_PRICE, and UNIT_SALE_PRICE. You also see ORDER_NUMBER, confirming the relationship to ORDER_DETAILS. You also see PRODUCT_NUMBER. You will confirm that this is the link to the Product data.

3. Scroll down the list of tables to the **PRODUCT** table.

   You can see a series of tables whose names start with PRODUCT_, so you will have to view each one to decide if they contain data that meets the reporting requirements.

4. Click **PRODUCT** to view details for the PRODUCT table in the Details pane.

   You see PRODUCT_NUMBER, confirming that this is the link to the sales (ORDER_DETAIL) data. Upon asking the Database Administrator about BASE_PRODUCT_NUMBER, you learn that this is for handling multiple brands of a given product. You see several codes, indicating foreign keys to PRODUCT_TYPE, PRODUCT_COLOR, PRODUCT_SIZE, and PRODUCT_BRAND. You also see the product facts you were looking for to report on: PRODUCTION_COST and GROSS_MARGIN. However you don't see a product name or description field. You will have to keep looking.

5. Click **PRODUCT_TYPE** to view details for the PRODUCT_TYPE table in the Details pane.

   You see PRODUCT_TYPE_CODE, confirming the link to PRODUCT. You also see PRODUCT_LINE_CODE (which you can confirm links to PRODUCT_LINE in a three-level hierarchy; product line>product type>product).

   Note the many PRODUCT_TYPE_xx columns. This is one form of multilingual support, a column for each language. Later in the course, you will see how you can modify the metadata so that authors and consumers can access a single product type column, which will automatically return the appropriate language based on the user's regional settings. Recall that you did not see such a list of columns for the name of the product itself in PRODUCT, so that table must have a different form of multilingual support.

6. Click **PRODUCT_NAME_LOOKUP** to view details for the PRODUCT_NAME_LOOKUP table in the Details pane.

   Now you see PRODUCT_NAME and PRODUCT_DESCRIPTION as well as PRODUCT_LANGUAGE, and you can conclude that this table must have multiple language records for each PRODUCT_NUMBER.

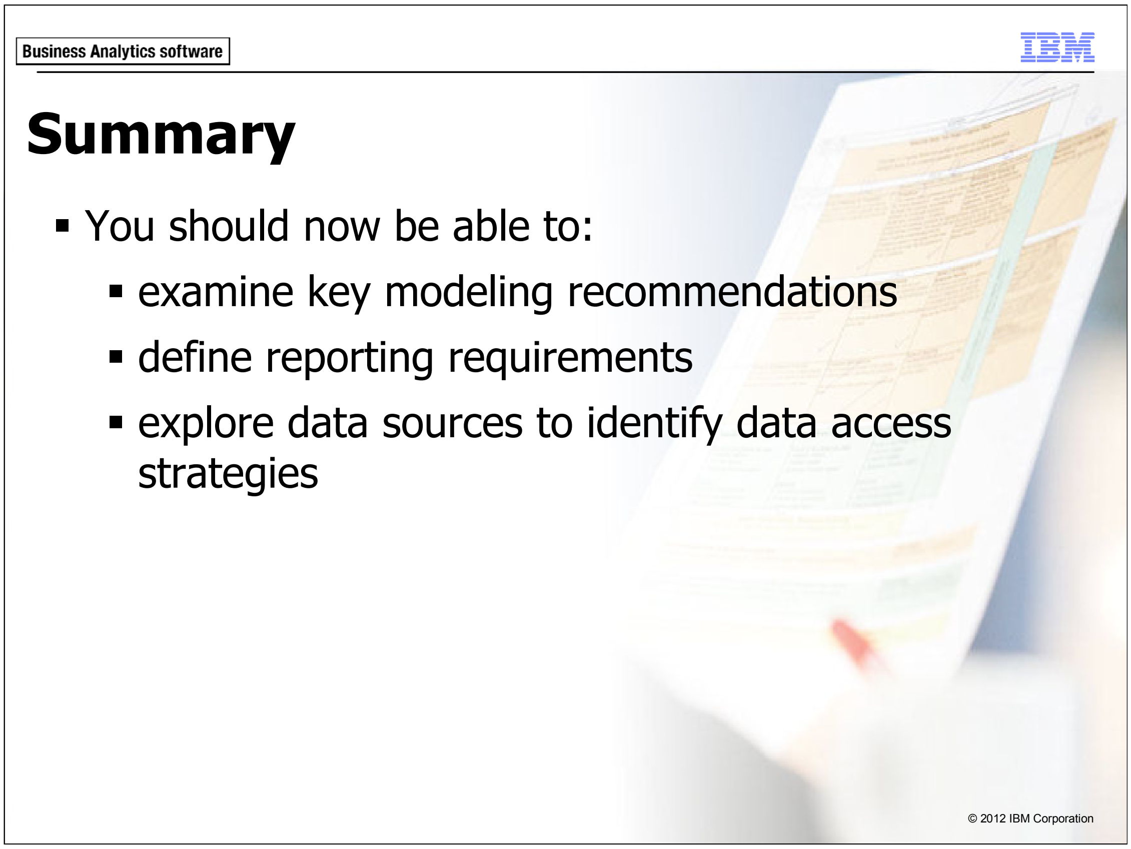
   The only outstanding report item required before beginning a baseline project is an order method. Recall that ORDER_HEADER holds an ORDER_METHOD_CODE column.

7. Click **ORDER_METHOD**.

   You see the order method columns; one for each supported language in the data. You will need to resolve this multilingual challenge in the same manner you will use for PRODUCT_TYPE_xx.

8. Close **Control Center**.

Results:
You have identified the following tables as those needed for the first phase of your model building, a baseline project.
- ORDER_HEADER, ORDER_DETAILS, ORDER_METHOD
- PRODUCT, PRODUCT_TYPE, PRODUCT_LINE, PRODUCT_NAME_LOOKUP



Business Analytics software

IBM

Summary

- You should now be able to:
  - examine key modeling recommendations
  - define reporting requirements
  - explore data sources to identify data access strategies

© 2012 IBM Corporation

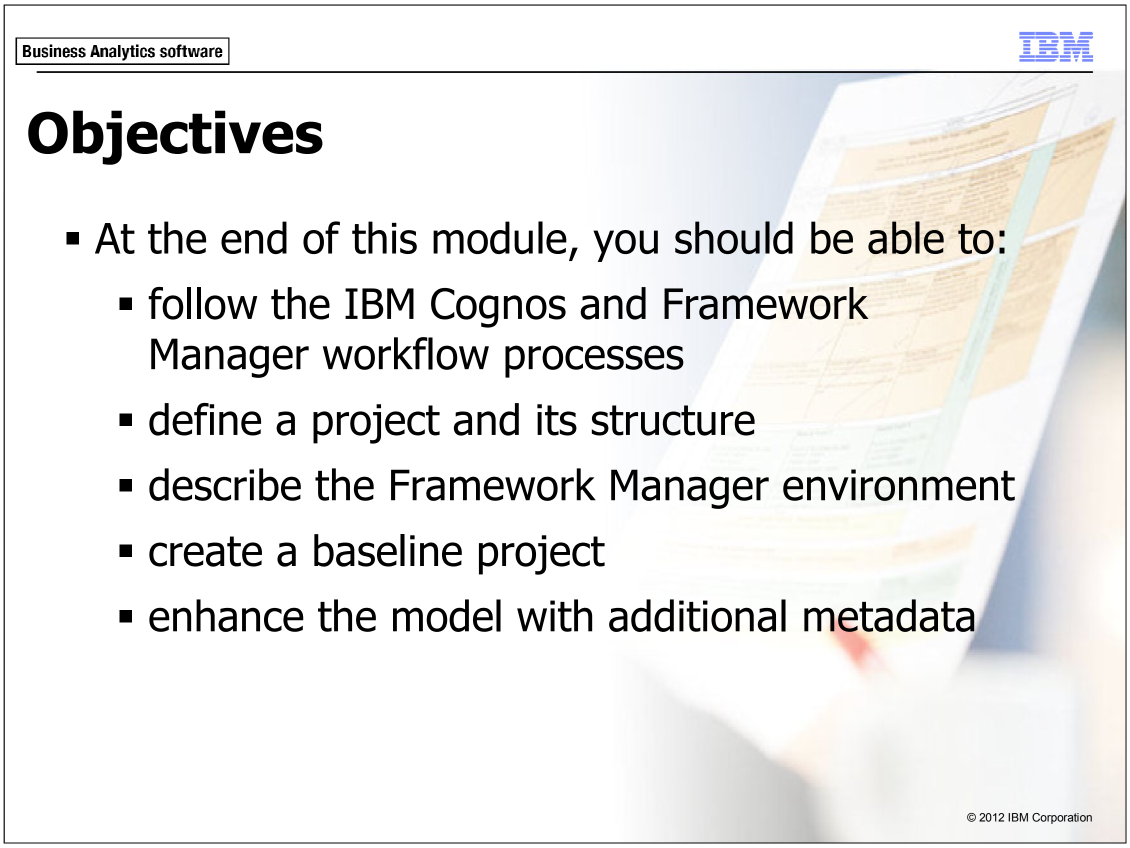
Create a Baseline Project

IBM Cognos BI

Business Analytics software

Unless otherwise specified in demo or workshop steps, you will always log on to IBM Cognos in Local LDAP namespace using the following credentials:

- User ID: admin
- Password: Education1



Business Analytics software

Objectives

- At the end of this module, you should be able to:
  - follow the IBM Cognos and Framework Manager workflow processes
  - define a project and its structure
  - describe the Framework Manager environment
  - create a baseline project
  - enhance the model with additional metadata

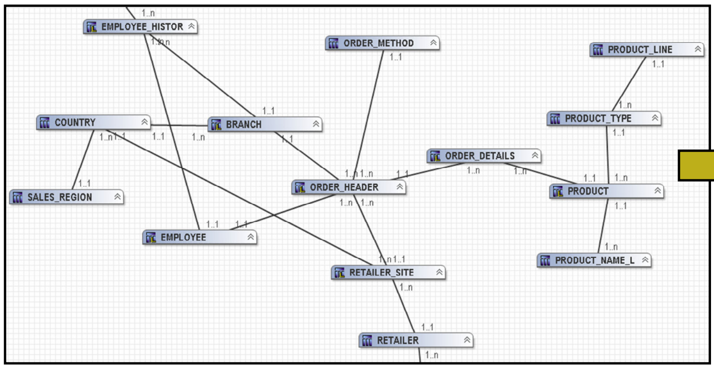
© 2012 IBM Corporation

Before reviewing this module, you should review the following modules:
- Overview of IBM Cognos
- Identify Common Data Structures
- Gather Requirements

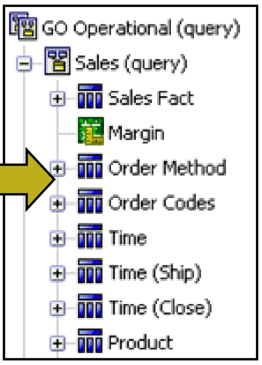
Business Analytics software

What is Framework Manager?

- Provides the metadata model development environment for IBM Cognos
- Used to create simplified business presentations of metadata derived from one or more data sources



Complex Data Source



Simple Business Presentation

© 2012 IBM Corporation

When you work in Framework Manager, you work in a project. A project contains metadata objects (the model) organized for report authors according to the business model and rules. A model in Framework Manager is a business presentation of the structure of the data from one or more data sources. A model defines the metadata objects, structure, and grouping, as well as relationships and security.

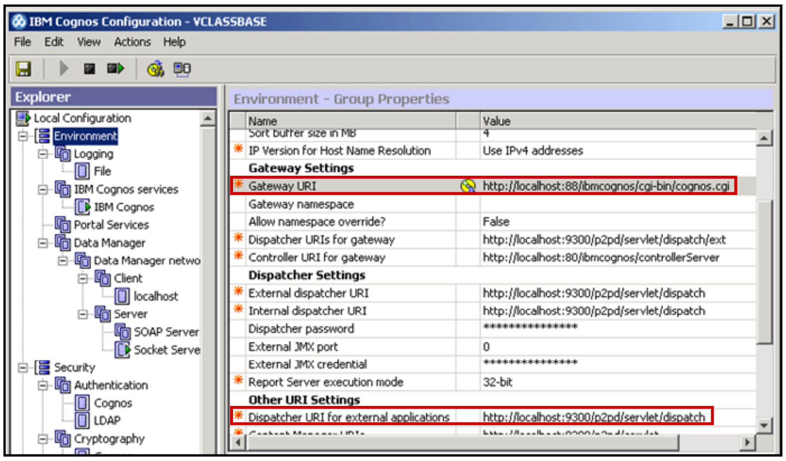
The modeler's job is to take the complexity of the underlying data structures and create simplified presentations for authors and business analysts that provide predictable results. These simplified views are then published as packages to the IBM Cognos Connection for use in the BI studios.

Framework Manager is a Windows-based client application as opposed to Business Workspace, Query Studio, Report Studio, and Cognos Connection, which are Web-based applications.

Business Analytics software

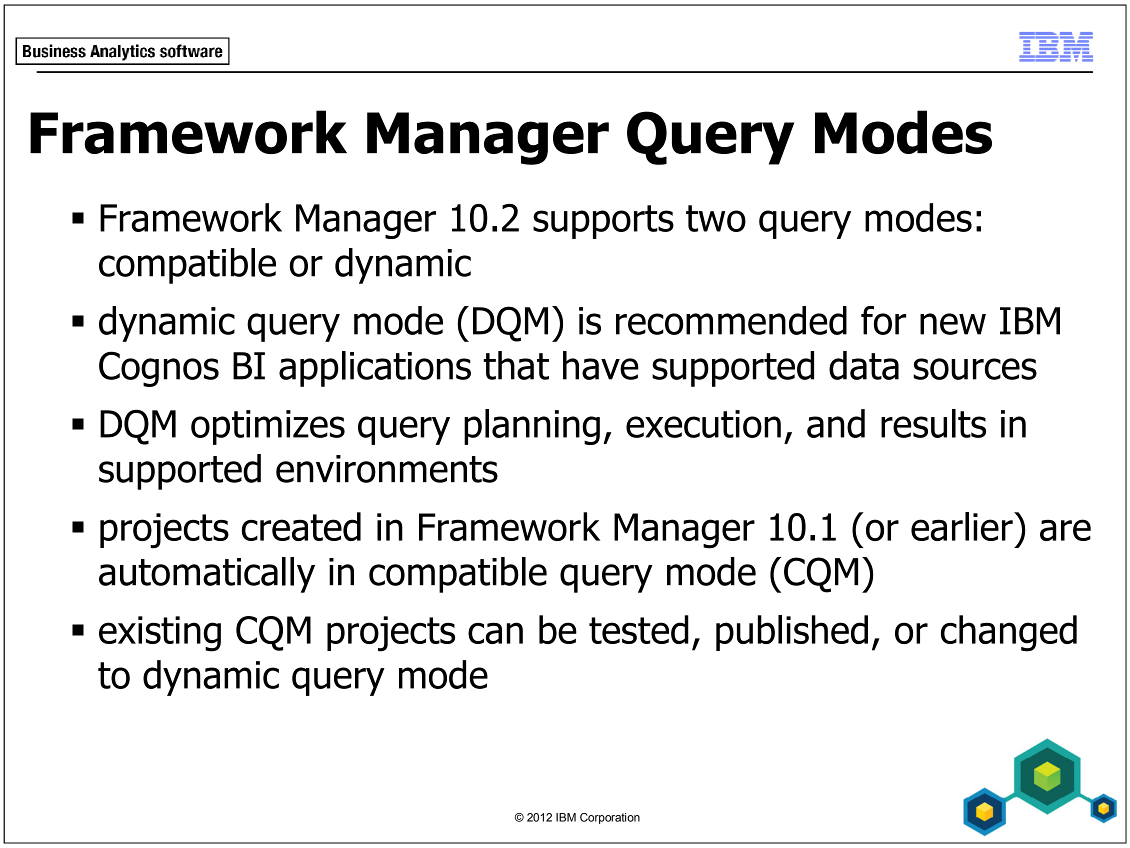
How does Framework Manager Connect to IBM Cognos BI?

- The connection between Framework Manager and the Cognos BI dispatcher is defined in Cognos Configuration



Name	Value
Sort buffer size in MB	4
IP Version for Host Name Resolution	Use IPv4 addresses
Gateway Settings	
Gateway URI	http://localhost:88/ibmcognos/cgi-bin/cognos.cgi
Gateway namespace	
Allow namespace override?	False
Dispatcher URIs for gateway	http://localhost:9300/p2pd/servlet/dispatch/ext
Controller URI for gateway	http://localhost:80/ibmcognos/controllerServer
Dispatcher Settings	
External dispatcher URI	http://localhost:9300/p2pd/servlet/dispatch
Internal dispatcher URI	http://localhost:9300/p2pd/servlet/dispatch
Dispatcher password	***************
External JMX port	0
External JMX credential	***************
Report Server execution mode	32-bit
Other URI Settings	
Dispatcher URI for external applications	http://localhost:9300/p2pd/servlet/dispatch

When installing Framework Manager on a different computer than other IBM Cognos BI components, the installer must define a connection that allows Framework Manager to communicate with the dispatcher, either directly or through a dedicated gateway. This connection allows the modeler to publish packages to the Cognos BI environment, by ensuring that Framework Manager connects to the same URI as the non-modeling components of IBM Cognos BI.



Business Analytics software

Framework Manager Query Modes

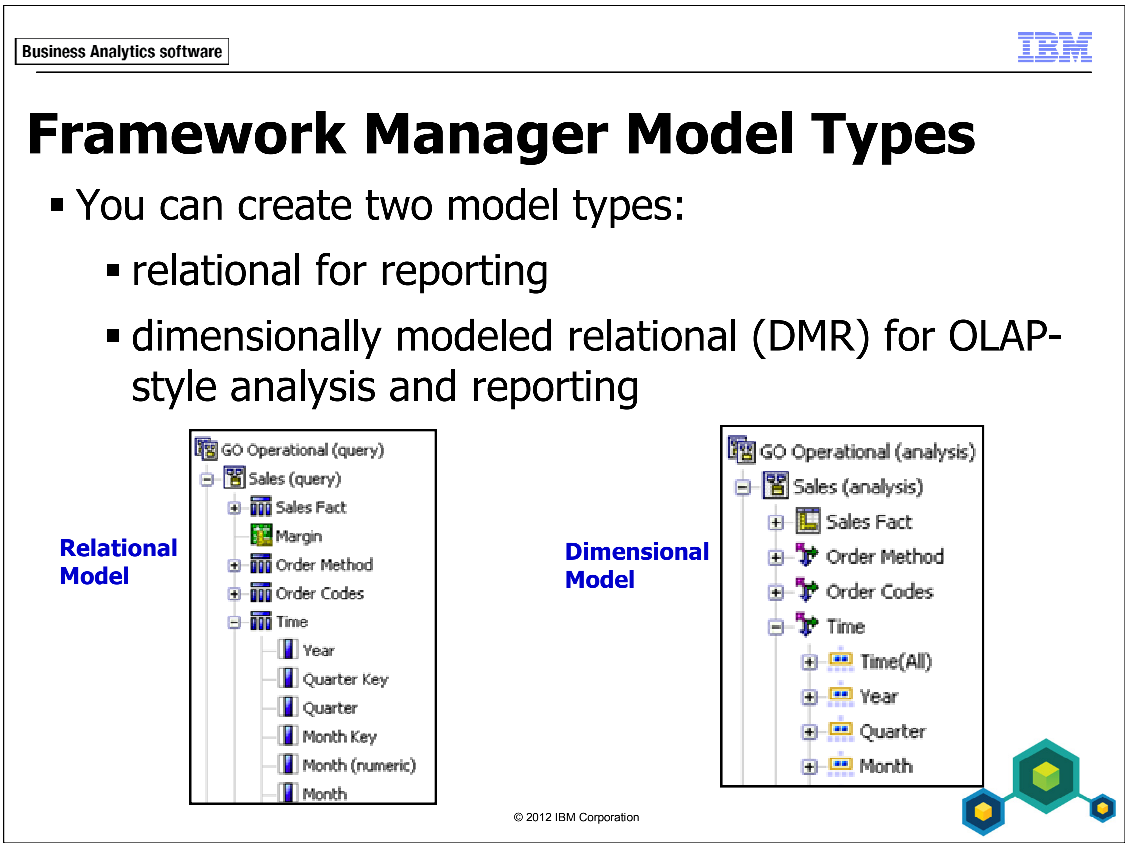
- Framework Manager 10.2 supports two query modes: compatible or dynamic
- dynamic query mode (DQM) is recommended for new IBM Cognos BI applications that have supported data sources
- DQM optimizes query planning, execution, and results in supported environments
- projects created in Framework Manager 10.1 (or earlier) are automatically in compatible query mode (CQM)
- existing CQM projects can be tested, published, or changed to dynamic query mode

© 2012 IBM Corporation

When you create a project in Framework Manager 10.2, you can select which query mode it will use. If your data sources are supported, DQM is the recommended mode, as it improves query performance by reducing query complexity and using in-memory caching to optimize query execution.

Projects created in Framework Manager 10.1 (or earlier versions) are automatically in CQM. However, you can still test model objects and publish packages in DQM, or even change the Query Mode property to dynamic query mode. Note that, before you change the mode in which you publish a package, you should use IBM Lifecycle Manager to identify the impact on any reports that use that package.

This course uses CQM-enabled models, due to the restrictions of the classroom environment.



Business Analytics software

Framework Manager Model Types

- You can create two model types:
  - relational for reporting
  - dimensionally modeled relational (DMR) for OLAP-style analysis and reporting

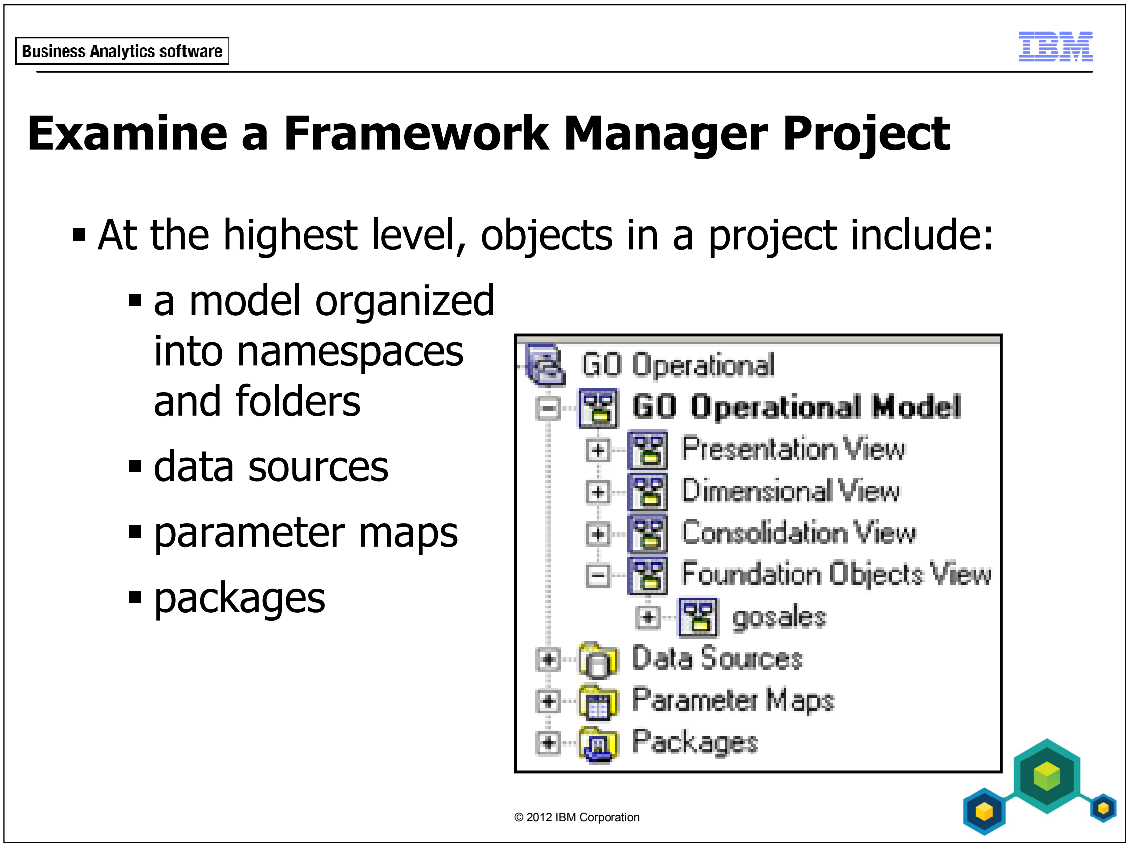
Relational Model
GO Operational (query)
Sales (query)
Sales Fact
Margin
Order Method
Order Codes
Time
Year
Quarter Key
Quarter
Month Key
Month (numeric)
Month
Dimensional Model
GO Operational (analysis)
Sales (analysis)
Sales Fact
Order Method
Order Codes
Time
Time(All)
Year
Quarter
Month

© 2012 IBM Corporation

Relational models consist of query subjects and within those, query items.

DMR models consist of dimensional information provided by the modeler, such as hierarchies and levels, to allow authors to perform OLAP-style queries.

These two model types can be developed in the same project and published as separate packages. In the slide example, the packages are GO Operational (query) and GO Operational (analysis), both published from the same project.



Business Analytics software

IBM

Examine a Framework Manager Project

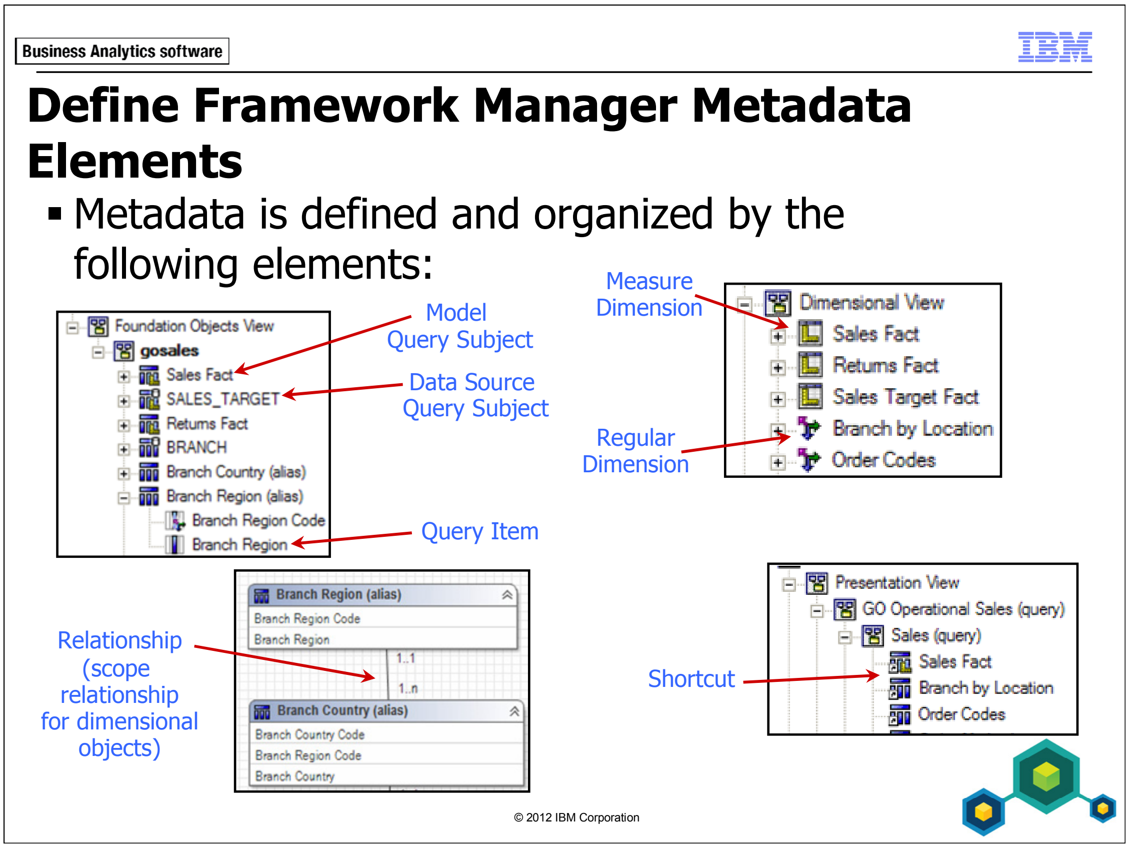
- At the highest level, objects in a project include:
  - a model organized into namespaces and folders
  - data sources
  - parameter maps
  - packages

GO Operational
GO Operational Model
Presentation View
Dimensional View
Consolidation View
Foundation Objects View
gosales
Data Sources
Parameter Maps
Packages

© 2012 IBM Corporation

The system files of a project consist of a folder containing a project file (.cpf) and a set of XML files specific to that project. These files should be backed up or checked into a repository control system regularly to prevent data loss. A project contains the following objects:

- Model - the set of metadata objects organized and modeled for report authors. The root of the model is a namespace that can contain other namespaces and folders, which are containers for organizing objects; however the names of objects in a namespace are qualified with the namespace name to uniquely identify them. This allows you to have objects of the same name in different namespaces.
- Data sources - information IBM Cognos uses to connect to data sources
- parameter maps (optional) - a list allowing the substitution of one value for another at run time
- Package - a subset of the project objects that is published to IBM Cognos Connection for use by authors and business analysts.



Business Analytics software

Define Framework Manager Metadata Elements

- Metadata is defined and organized by the following elements:

© 2012 IBM Corporation

In Framework Manager, you will interact with the following objects:

- **Query subjects** - map to objects in the data source, such as tables, views, synonyms, procedures, or functions. There are three types of query subjects:
  - **Data source**: maps to a corresponding object in the data source and uses a modifiable SQL statement to retrieve the data
  - **Model**: maps to existing metadata in the model
  - **Stored procedure**: executes a database stored procedure to retrieve or update the data

- **Query items** - contained within a query subject and maps to a corresponding object in the data source. Query items are contained in query subjects. For example, a query subject that references an entire table contains query items that represent each column in the table.

  - **Relationships** - represent the connection that explains how the data in one query subject relates to the data in another. When you create a relationship, you define the cardinality of each end of the relationship.

  - **Regular dimensions** - contain descriptive and business key information, and organize the information in a hierarchy from the highest level of granularity to the lowest, allowing for OLAP-style queries. Regular and measure dimensions map to query items. You would work with regular and measure dimensions when modeling dimensionally. You model dimensionally when your report and business analysts require OLAP-style queries, reports, and analyses

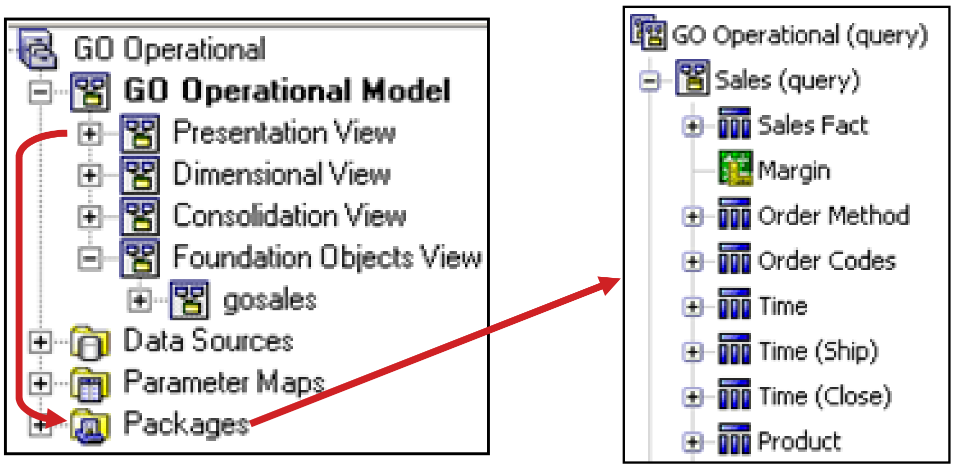
  - **Measure dimensions** - a collection of facts for OLAP-style queries

  - **Scope relationships** - exists between measure dimensions and regular dimensions to define the level at which the measures are available for reporting

  - **Shortcuts** - pointer to an underlying object that can act as an alias or reference. In this course we mainly use shortcuts to create star schema groupings for presentation of logical groupings to the authors.

Business Analytics software

Examine the Model from the Author's Point of View



Within IBM Cognos Connection and the various studios, authors interact with a run-time version of a subset of the Framework Manager model. This subset, which the modeler publishes as a package from Framework Manager to the IBM Cognos server, contains metadata that exists in the development model. It is packaged in a structure reflecting a viewpoint appropriate for report authors and business analysts. The actual data is not published with the package; it is retrieved from the underlying data source at run time.

In Framework Manager, you can create several packages containing different subsets of the model.

Demo 1: Examine the Final Project

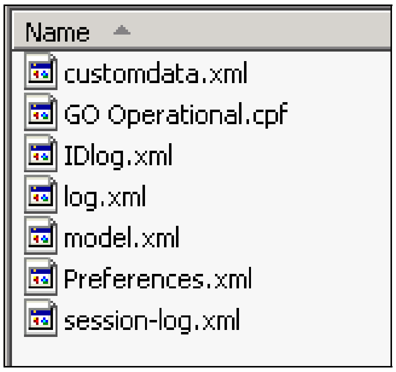
Purpose:
Before you begin the modeling process for your Sample Outdoors model, you will explore the Framework Manager UI and view the final results of the model you will build by the end of this course. This will help bring context to the tasks ahead of you.

Components:	**Framework Manager, IBM Cognos Workspace Advanced**
Project:	**GO Operational**
Package:	**GO Operational (query)**

Task 1. Examine the Framework Manager UI and view the top-level objects.

1. Launch **Windows Explorer**, and then navigate to **C:\Edcognos\B5252\CBIFM-Files\Final**.

   The results appear as follows:

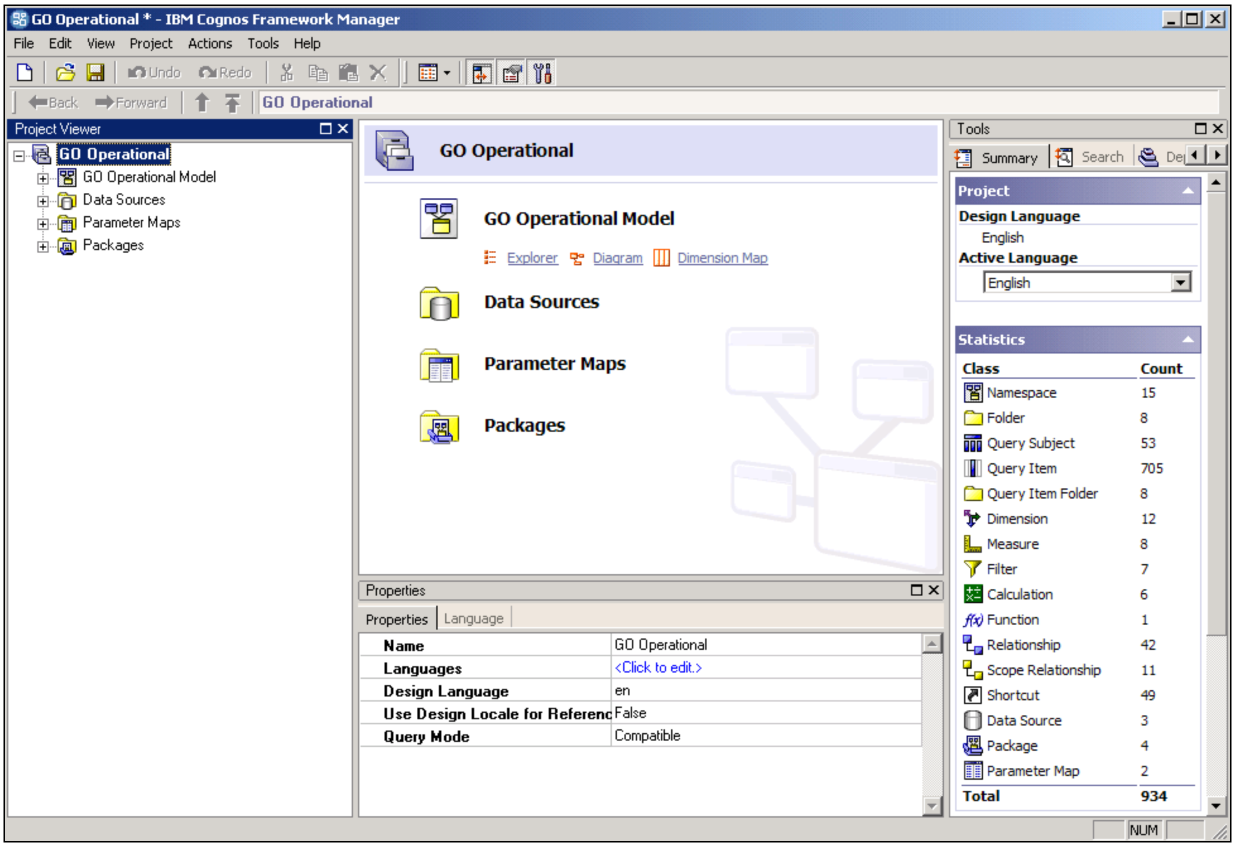
   Name
   - customdata.xml
   - GO Operational.cpf
   - IDlog.xml
   - log.xml
   - model.xml
   - Preferences.xml
   - session-log.xml

   Here you see the project files for the final Framework Manager model you will be building throughout this course. The cpf file is the main project file. The xml files store model information, preferences, session logging, action logging (each modeling action you make), and so on.

Task 1: This demo is just a brief one to show the Framework Manager UI and project structure and view our goal for the demos and workshops over the length of the course.
The training development process differs from the methods used in this course to create models. Therefore, log files have been removed from the training environment.

2. From the **Start** menu, point to **All Programs>IBM Cognos 10**, and then click **IBM Cognos Framework Manager**.
3. Click **Open a project**, navigate to **C:\Edcognos\B5252\CBIFM-Files\Final**, and then double-click **GO Operational.cpf**.
4. Log in as **admin/Education1**.

The results appear as follows:



Design Language
English
Active Language
English

Name	GO Operational
Languages	<Click to edit.>
Design Language	en
Use Design Locale for Referenc	False
Query Mode	Compatible

Statistics

Class	Count
Namespace	15
Folder	8
Query Subject	53
Query Item	705
Query Item Folder	8
Dimension	12
Measure	8
Filter	7
Calculation	6
Function	1
Relationship	42
Scope Relationship	11
Shortcut	49
Data Source	3
Package	4
Parameter Map	2
Total	934

Take the time to examine the different UI elements.

Project Viewer pane:	By default this pane is on the left and provides an easy way to access all your project's objects in a tree format.
Project Info pane:	This is the center pane and also provides access to the project's objects through various methods. Three tabs (Explorer, Diagram, and Dimension Map) allow you to create, edit, configure, or delete objects. You will use each of these throughout the course.
Properties pane:	This pane allows you to configure various properties for any of the project's objects.
Tools pane:	This pane provides several useful tools. You can use it to quickly switch the project language in this pane as well as view project statistics, and quickly perform common tasks for selected objects. This pane also provides a search utility (second tab) and an object dependencies utility (third tab). Simply drag an object (and its children if it has any) to the top panel, select the object or one of its children in the top panel and view the dependant objects in the bottom panel. This is very useful when you want to change an object and assess the impact that the change will have on other objects in the model.

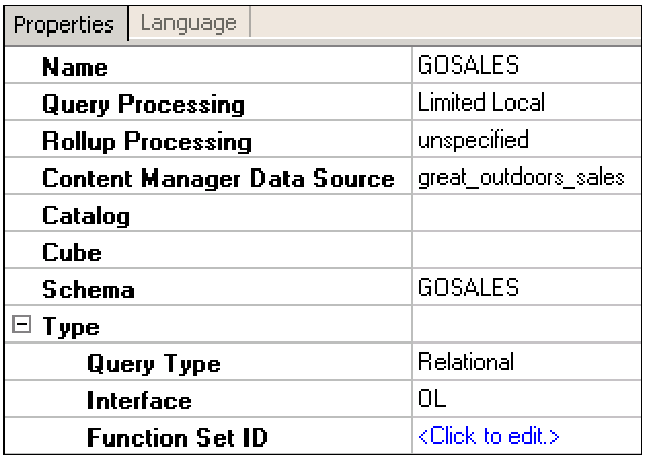
All panes can be hidden except the Project Info pane, which is the main work area. To bring them back, simply select them from the View menu or use the toggles on the toolbar. You can also detach and rearrange the Project Viewer, Properties and Tools panes.

Now that you have examined the main elements of the Framework Manager UI, take the time to examine the project.

In the project viewer examine the top level-objects. They are a GO Operational Model namespace (contains model objects), a Data Sources folder (contains the data sources used in the project), a Parameter Maps folder (contains parameter maps that allow for data substitution at run time), and a Packages folder (contains packages that are published to IBM Cognos to make model metadata available to authors).

5. Expand **Data Sources**, click **GOSALES**, and then expand **Type** in the **Properties** pane.

   The results appear as follows:



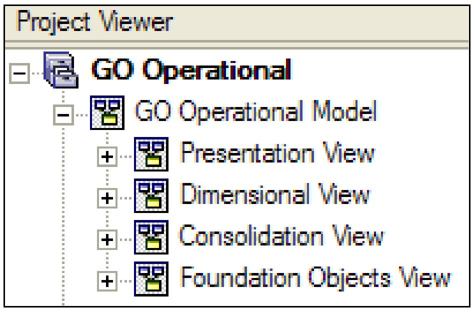
Properties | Language

Name	GOSALES
Query Processing	Limited Local
Rollup Processing	unspecified
Content Manager Data Source	great_outdoors_sales
Catalog	
Cube	
Schema	GOSALES
⊟ Type	
Query Type	Relational
Interface	OL
Function Set ID	<Click to edit.>

   Notice the Query Type property indicates that these are relational sources.

6. Expand **GO Operational Model**.

   The results appear as follows:



Project Viewer

```
⊟ GO Operational
   ⊟ GO Operational Model
      ⊞ Presentation View
      ⊞ Dimensional View
      ⊞ Consolidation View
      ⊞ Foundation Objects View
```

   Notice the four namespaces for logical separation of objects by function.

Consolidation View is an isolation layer. It is made up of model query subjects that are used as containers to consolidate multiple query subjects into one (such as Product, Product Type, and Product Line). It also simplifies query subjects by hiding codes or organizing them in subfolders. This layer insulates reports from potential changes in the underlying data source.

Dimensional View contains multidimensional objects based on model query subjects in the Consolidation View. These are used for OLAP-style queries and are based on objects in the Consolidation View.

Presentation View contains groupings of shortcuts that present the underlying objects in a logical presentation.

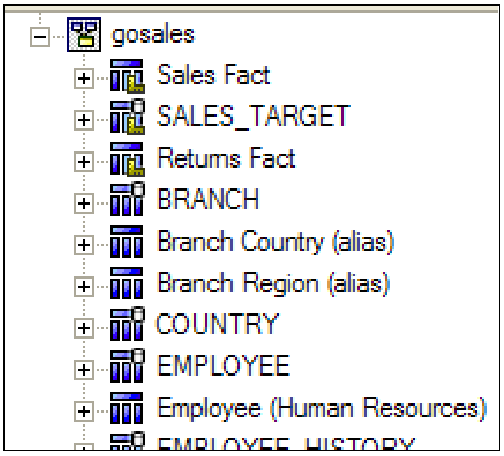
These are the naming conventions used in this course. You can use whatever naming convention and object organization you are comfortable with.

Foundation Objects View contains the "base" query subjects; data source query subjects as well as any model query subjects used to resolve reporting issues.

Task 2. View the detail objects.

1. Expand **Foundation Objects View** > **gosales**.

   The results appear as follows:

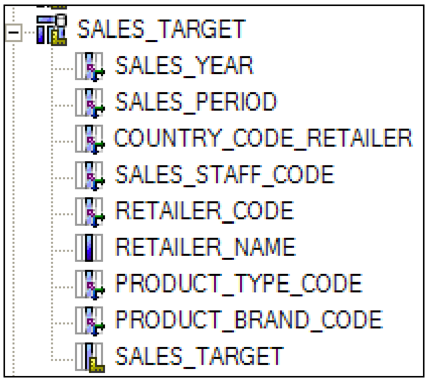
   

   The gosales namespace contains all the data source query subjects (represented with a small database icon) created during the import process from the GOSALES database as well as model query subjects that are used to alias original data source query subjects or to create views (merge query subjects together to create "As View" behavior) of original data source query subjects.

   The small yellow angle ruler icon indicates the query subject contains measures as seen on Sales Fact, SALES_TARGET, and Returns Fact.

2. Expand **SALES_TARGET**.

   The results appear as follows:

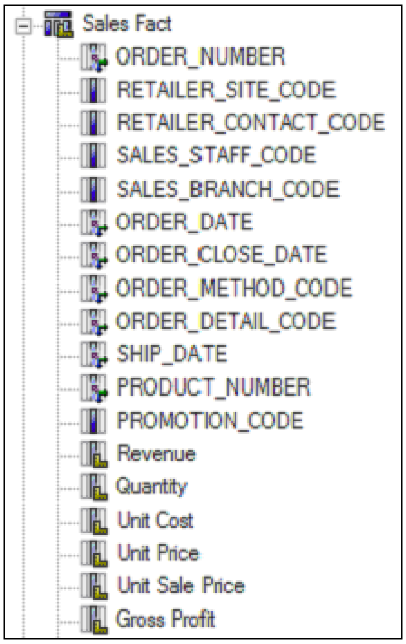


SALES_TARGET
SALES_YEAR
SALES_PERIOD
COUNTRY_CODE_RETAILER
SALES_STAFF_CODE
RETAILER_CODE
RETAILER_NAME
PRODUCT_TYPE_CODE
PRODUCT_BRAND_CODE
SALES_TARGET

   Notice that, with the exception of a few property changes, these data source query subjects are untouched after being imported. This is recommended to ease maintenance.

3. Expand **Sales Fact**.

   The results appear as follows:



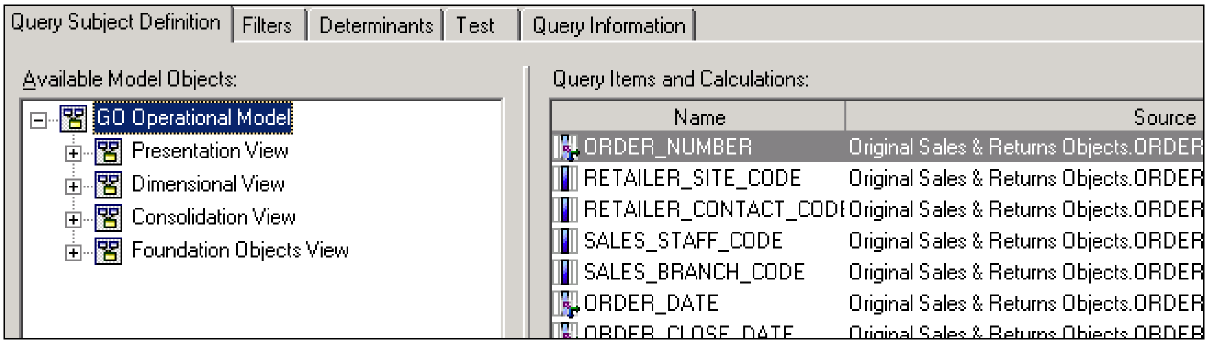
Sales Fact
ORDER_NUMBER
RETAILER_SITE_CODE
RETAILER_CONTACT_CODE
SALES_STAFF_CODE
SALES_BRANCH_CODE
ORDER_DATE
ORDER_CLOSE_DATE
ORDER_METHOD_CODE
ORDER_DETAIL_CODE
SHIP_DATE
PRODUCT_NUMBER
PROMOTION_CODE
Revenue
Quantity
Unit Cost
Unit Price
Unit Sale Price
Gross Profit

This is a model query subject that is the result of merging two data source query subjects together to create "As View" behavior.

Query items to be used by authors have been given user-friendly names. This is not needed for query items that will not be included in the final Presentation View. Calculations have also been implemented in this object.

4. Double-click **Sales Fact** to open its definition dialog.

The results appear as follows:

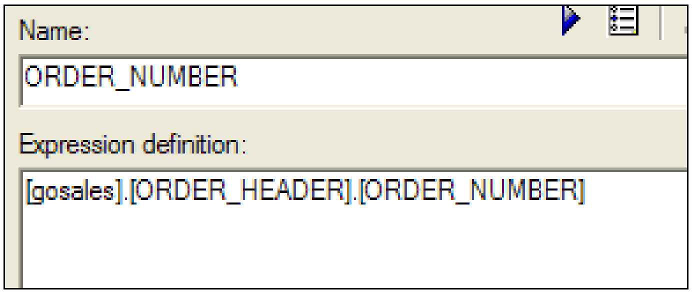


Name	Source
ORDER_NUMBER	Original Sales & Returns Objects.ORDER
RETAILER_SITE_CODE	Original Sales & Returns Objects.ORDER
RETAILER_CONTACT_CODE	Original Sales & Returns Objects.ORDER
SALES_STAFF_CODE	Original Sales & Returns Objects.ORDER
SALES_BRANCH_CODE	Original Sales & Returns Objects.ORDER
ORDER_DATE	Original Sales & Returns Objects.ORDER
ORDER_CLOSE_DATE	Original Sales & Returns Objects.ORDER

Here you can see all the query items that make up the model query subject as well as any filters or determinants that might be specified. The source column tells you how the query item is derived. You can edit the definition of each query item in this dialog by clicking the ellipses in the Source column. You can also go directly to the query item definition from the Project Viewer tree. You will do this now.

5. Click **Cancel**, and then double-click **ORDER_NUMBER**.

   The results appear as follows:

   Name:

   ORDER_NUMBER

   Expression definition:

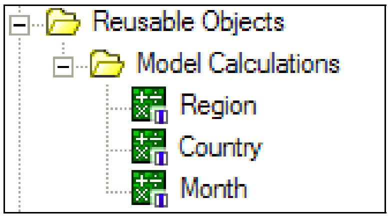
   [gosales].[ORDER_HEADER].[ORDER_NUMBER]

   Here you can see and edit the definition of the query item. The query item can be a direct reference to another item, or a calculation. In this case it is a direct reference that is broken down as follows:

   [Namespace].[Query Subject].[Query Item]

6. Click **Cancel**, and then expand **Reusable Objects** > **Model Calculations**.

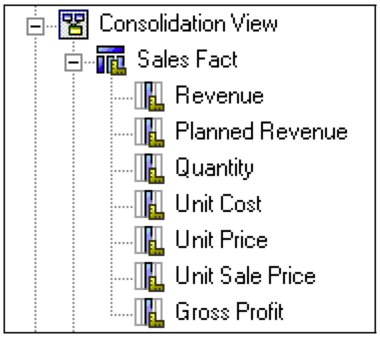
   The results appear as follows:

   This folder contains several commonly requested calculations (in this case, the calculation retrieves a specific language value from the data source) to allow maintenance from a single source.

7. Expand **Consolidation View** and **Sales Fact**.

   The results appear as follows:

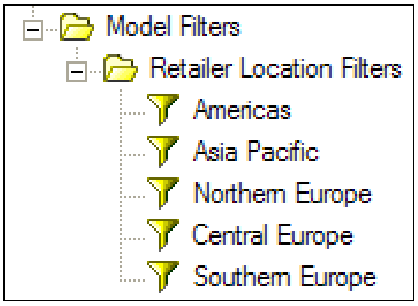
   

   Consolidation View
   - Sales Fact
     - Revenue
     - Planned Revenue
     - Quantity
     - Unit Cost
     - Unit Price
     - Unit Sale Price
     - Gross Profit

   This namespace is a simplified version of Sales Fact from the Foundation Objects View. It only contains the measures (facts), hiding the complex keys. This layer is a consistent presentation where all objects simply act as containers to consolidate information from the underlying Foundation Objects View. This layer also contains calculations, filters, and appropriate naming conventions. These are the objects authors will work with directly or indirectly.

8. Expand **Model Filters** > **Retailer Location Filters**.

   The results appear as follows:

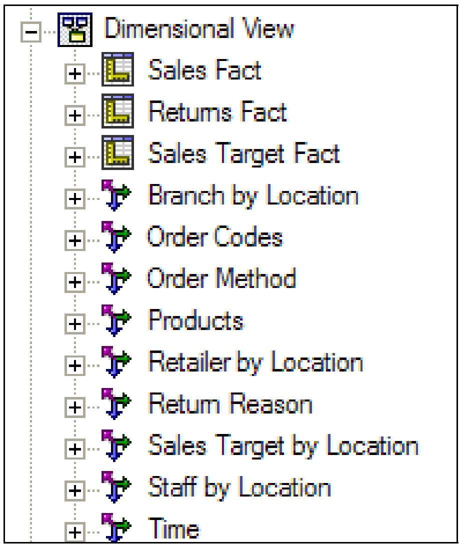
   

   Model Filters
   - Retailer Location Filters
     - Americas
     - Asia Pacific
     - Northern Europe
     - Central Europe
     - Southern Europe

   These folders contain several filters that authors can pick from to simplify report filtering.

9. Expand **Dimensional View**.

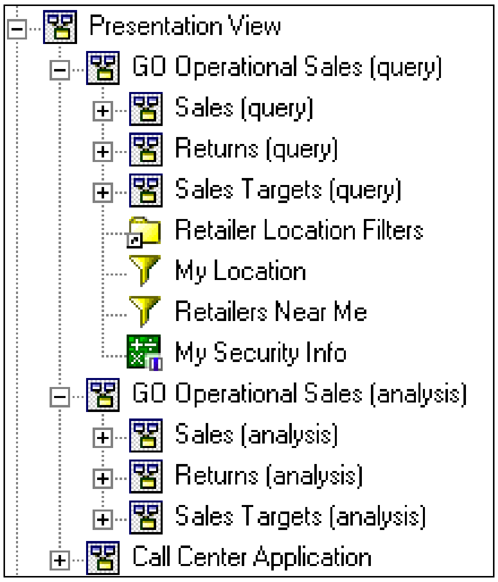
   The results appear as follows:



   Here you can see dimensions and measures based on objects in the Consolidation View. Dimensional information (hierarchies, levels, attributes, and so on) is provided in this view. These objects are used for OLAP-style queries in Cognos Workspace, Analysis Studio, Query Studio, and Report Studio.

10. Expand **Presentation View**, and then expand **GO Operational Sales (query)** and **GO Operational Sales (analysis)**.

    The results appear as follows:

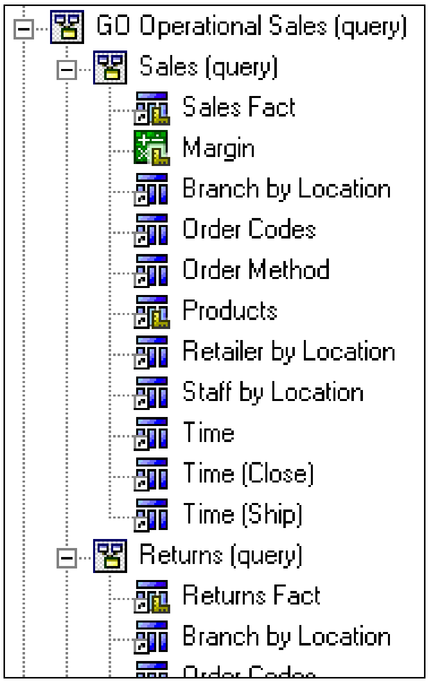
    

    ```
    Presentation View
        GO Operational Sales (query)
            Sales (query)
            Returns (query)
            Sales Targets (query)
            Retailer Location Filters
            My Location
            Retailers Near Me
            My Security Info
        GO Operational Sales (analysis)
            Sales (analysis)
            Returns (analysis)
            Sales Targets (analysis)
        Call Center Application
    ```

    Here you can see two namespaces, one containing objects for queries and reports, and the other for analysis. There is also a Call Center Application namespace that is specific to a particular application to quickly retrieve order information.

11. In **GO Operational Sales (query)**, expand **Sales (query)**, and **Returns (query)**.

    The results appear as follows:

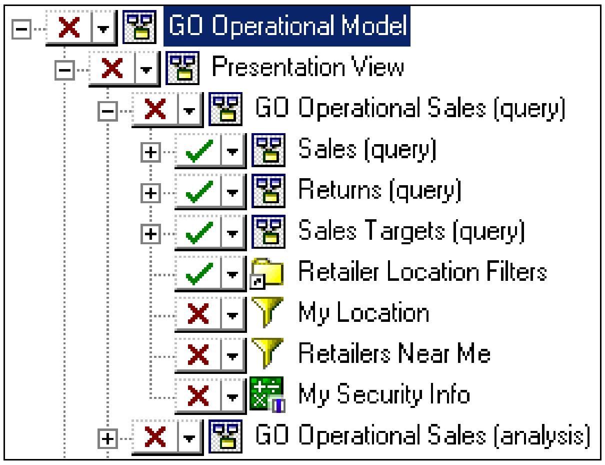
    
    GO Operational Sales (query)
    Sales (query)
    Sales Fact
    Margin
    Branch by Location
    Order Codes
    Order Method
    Products
    Retailer by Location
    Staff by Location
    Time
    Time (Close)
    Time (Ship)
    Returns (query)
    Returns Fact
    Branch by Location

    These two namespaces contain shortcuts that logically group the Consolidation View objects. Facts are grouped with their related dimensions. The namespaces have some overlapping object names (conformed dimensions). To create multi-fact queries across multiple areas of the business, include at least one conformed dimension.

Task 3. View the package as a report author.

1. Expand **Packages**, and then double-click **GO Operational (query)**.

   The results appear as follows:

   GO Operational Model
   Presentation View
   GO Operational Sales (query)
   Sales (query)
   Returns (query)
   Sales Targets (query)
   Retailer Location Filters
   My Location
   Retailers Near Me
   My Security Info
   GO Operational Sales (analysis)

   The package contains a subset of the project specific to objects to be used for relational queries and reports.

2. Click **Cancel**, right-click **GO Operational (query)**, and then click **Publish Packages**.
3. Click **Next**, click **Next**, deselect **Verify the package before publishing**, and then click **Publish**.
4. Click **Finish**.
5. Open your Internet browser and navigate to the following page: **http://localhost:88/ibmcognos**.
6. Log on as **admin/Education1**.

   This will log you in to IBM Cognos Connection.

7. Click **Author business reports**, and then click **GO Operational (query)**.

   This is the package you just published.

8. Click **Create new**, select **List** and then click **OK**.

9. In the **Insertable Objects** pane, ensure the **Source** tab is selected, and then expand **Sales (query)** and **Returns (query)**.

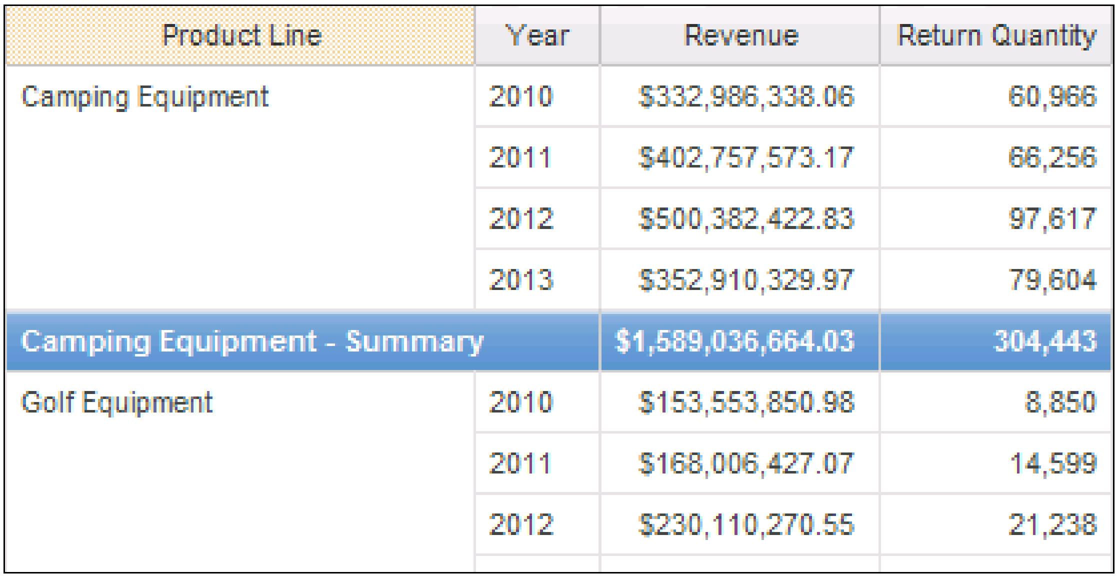
   Notice that this is the same structure and grouping seen in the Presentation View in the previous task.

10. Add the following items to the report:

Query Subject	Query Item
Sales (query)>Products	**Product Line**
Sales (query)>Time	**Year**
Sales (query)>Sales Fact	**Revenue**
Returns (query)>Returns Fact	**Return Quantity**

11. Click the **Product Line** column and click the **Group** icon to group the report on **Product Line**.

    The results appear as follows:

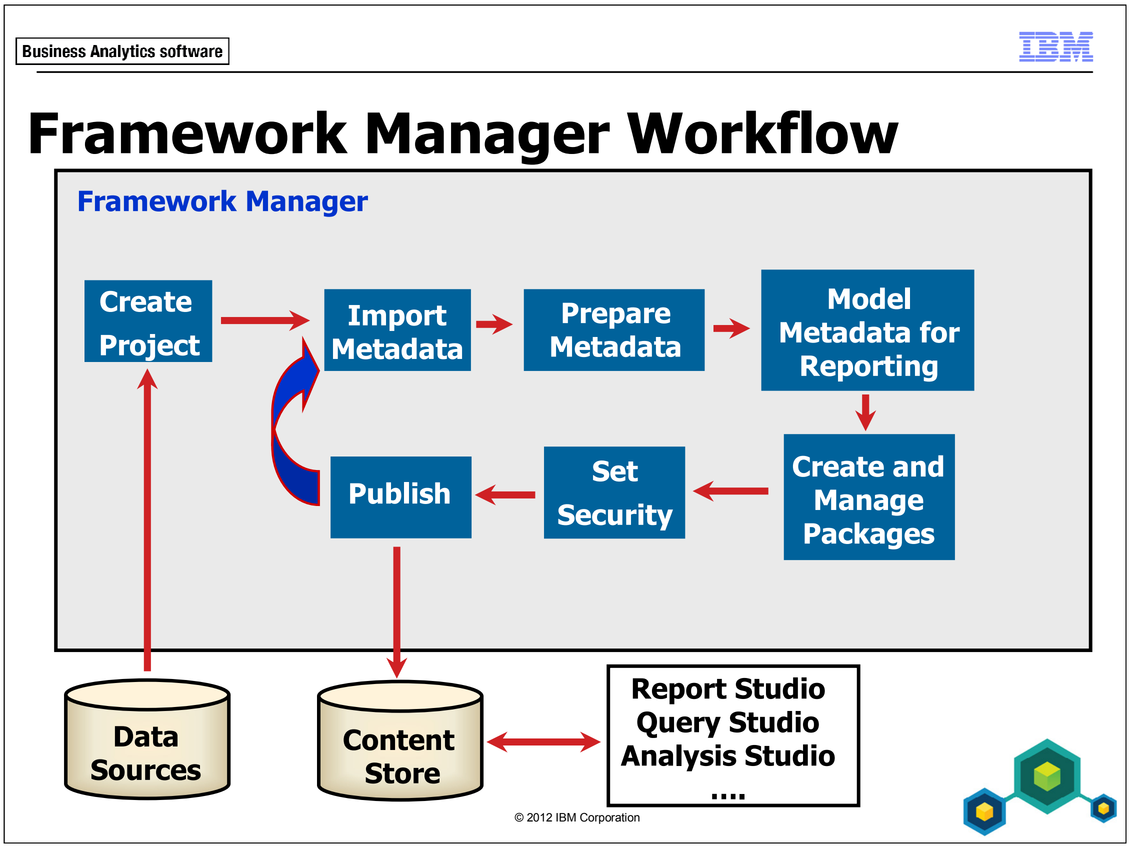


Product Line	Year	Revenue	Return Quantity
Camping Equipment	2010	$332,986,338.06	60,966
	2011	$402,757,573.17	66,256
	2012	$500,382,422.83	97,617
	2013	$352,910,329.97	79,604
Camping Equipment - Summary		$1,589,036,664.03	304,443
Golf Equipment	2010	$153,553,850.98	8,850
	2011	$168,006,427.07	14,599
	2012	$230,110,270.55	21,238

Using this intuitive view of the metadata, you can create queries quickly and easily.

12. Close the browser without saving the report.
13. In **Framework Manager**, from the **File** menu, click **Close** to close the project without saving changes.

Results:
You explored the Framework Manager UI and saw some of the final results in the model that you will build throughout this course. You also saw the benefits of creating a simple-to-use view of the metadata for the business.

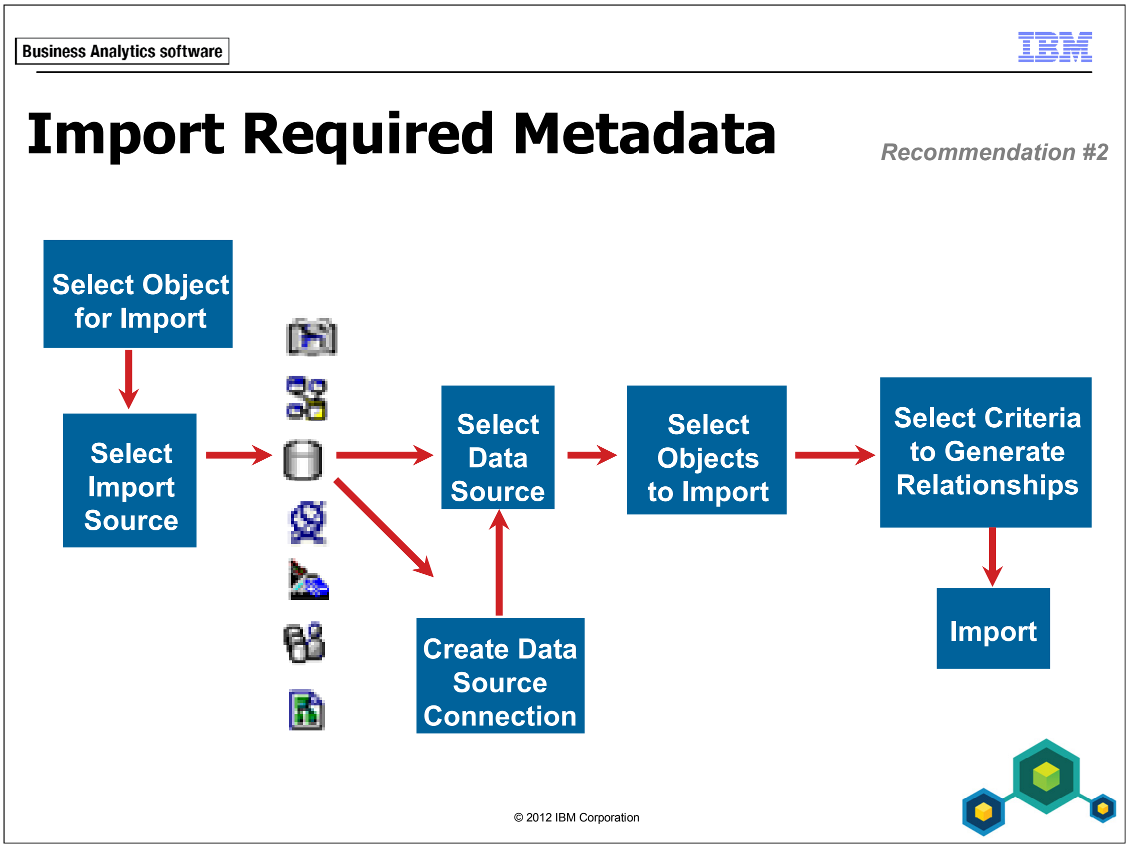

Business Analytics software

Framework Manager Workflow

© 2012 IBM Corporation

- **Import Metadata** - import objects such as tables, views, and procedures
- **Prepare Metadata** - examine and modify properties and relationships
- **Model Metadata for Reporting** - add business value specific to reporting requirements and model for predictable results
- **Create and Manage Packages** - identify subsets of the metadata to be published
- **Set Security** - apply security at various levels to restrict access
- **Publish** - publish packages to the IBM Cognos servers for use by report authors and business analysts

The blue arrow in the above diagram emphasizes that modeling is a reiterative process. Managing the project is another important activity, not shown on the diagram because it is to be performed throughout the modeling process. This includes activities such as implementing multi-user modeling, sharing and reusing information, action logging and synchronizing, and validating the project.



Business Analytics software

Import Required Metadata

Recommendation #2

© 2012 IBM Corporation

Import only the required metadata in a phased approach to minimize clutter in the model.

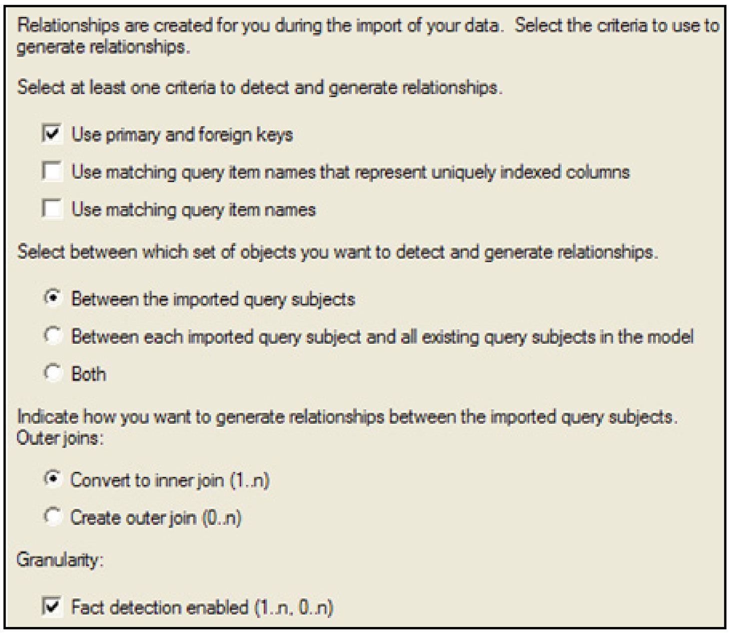
For relational data sources, choose the model object into which you will import, and then do the following:

1. Select Data Sources as the import source.
2. Select the data source to be used (usually a relational database), or create a data source connection, and then select it.

   Note: Creating a data source connection is usually done by an Administrator, using IBM Cognos Administration. Depending on the roles and responsibilities in your organization, you may need to request that your Administrator perform this task for you.
3. Select the required objects to import; for example, tables, views, or procedures.
4. Select the criteria by which Framework Manager will generate relationships.

Business Analytics software

Generate Relationship Criteria


Relationships are created for you during the import of your data. Select the criteria to use to generate relationships.

Select at least one criteria to detect and generate relationships.

- [✓] Use primary and foreign keys
- [ ] Use matching query item names that represent uniquely indexed columns
- [ ] Use matching query item names

Select between which set of objects you want to detect and generate relationships.

- ● Between the imported query subjects
- ○ Between each imported query subject and all existing query subjects in the model
- ○ Both

Indicate how you want to generate relationships between the imported query subjects.
Outer joins:

- ● Convert to inner join (1..n)
- ○ Create outer join (0..n)

Granularity:

- [✓] Fact detection enabled (1..n, 0..n)

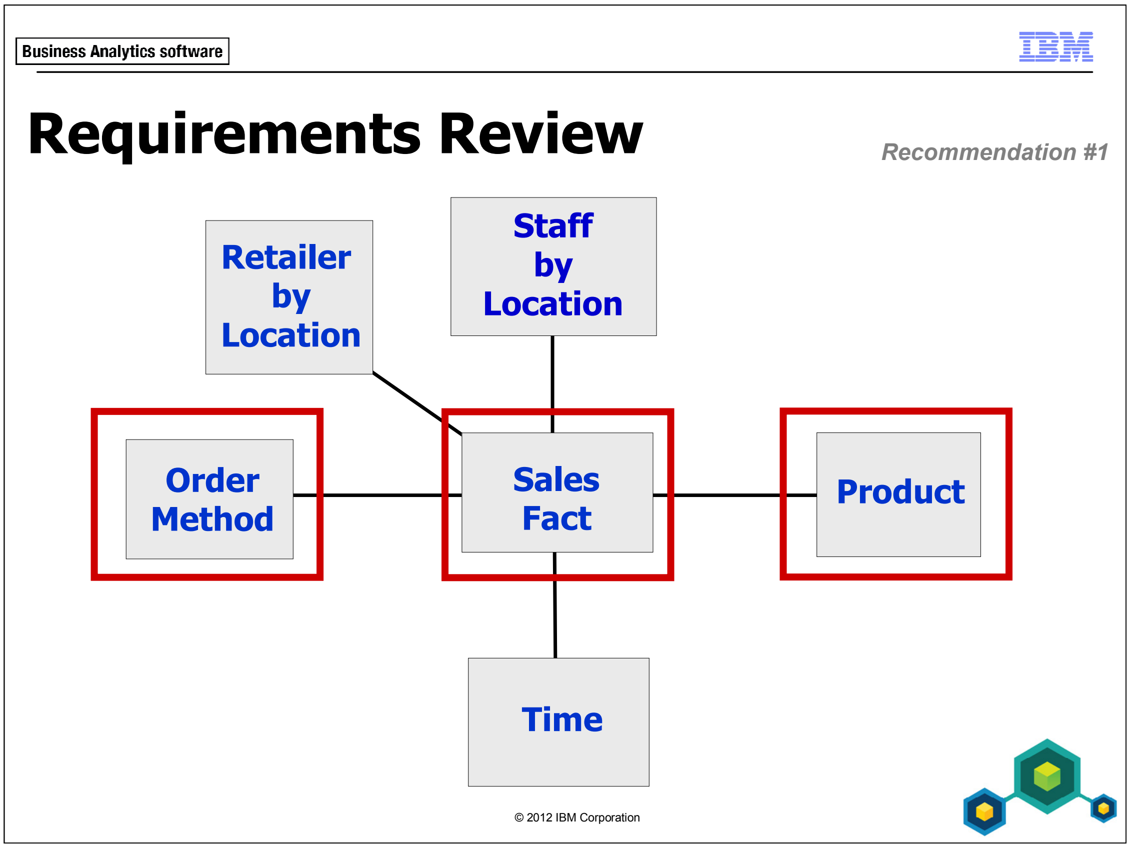
This dialog box is part of the Import Metadata Wizard.

In general, use the first check box for relational data from the same database. The other two check boxes are often used when importing from a different database.

You can request to create relationships among either the objects being imported, or the objects being imported and the existing objects in the model, or both.

By default, an import converts outer joins to inner joins. This is done for performance reasons. You can choose to generate outer joins if that meets your business needs or you can edit specific relationships after import to meet your needs.

Another option is to enable or disable fact detection. If this option is disabled, all relationships will be 1..1 to 1..1.



Business Analytics software

IBM

Requirements Review

Recommendation #1

© 2012 IBM Corporation

In the next demo, you will import metadata to satisfy the highlighted reporting requirements above. The database tables used to satisfy each requirement are as follows:

Reporting Requirement	Database Tables
Sales Fact	ORDER_DETAILS ORDER_HEADER
Order Method	ORDER_METHOD
Product	PRODUCT PRODUCT_LINE PRODUCT_NAME_LOOKUP PRODUCT_TYPE

Demo 2: Create a Baseline Project

Purpose:
Your first step as a data modeler for The Sample Outdoors Company is to create a new project, set up the data source connection it will use, and start the metadata modeling process.

Component:	**Framework Manager**
Project:	**GO Operational**

Task 1. Create a project.

1. Launch **Framework Manager**, and under **Projects**, click **Create a new project**.
2. In the **Location** box, click **Browse**, navigate to **C:\Edcognos\B5252** and create a new folder called **Course_Project**.
3. Select the **Course_Project** folder, click **OK**, and then in the **Project name** box, type **GO Operational**.

   The GO Operational folder is created by default and appears in the Location box.
4. Deselect the **Use Dynamic Query Mode** check box, and then click **OK**.
5. Log in as User ID **admin**, and Password **Education1**, if prompted.

   The Select Languages dialog box appears. You will initially set the default and design language for this project as English.
6. Ensure that **English** is selected, and then click **OK**.

   The Metadata Wizard appears. You will first organize your project and then import metadata into the appropriate location in a later task.
7. Click **Cancel**.

   The project opens in Framework Manager. Notice in the Project Viewer that the project appears with the name you provided and a default namespace called Model.

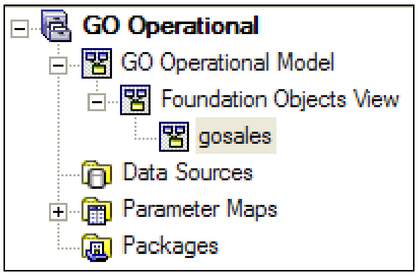
Task 2. Create and organize objects.

1. Rename the **Model** namespace to **GO Operational Model**.

   Tip: You can press F2 to edit the label.

2. Right-click the **GO Operational Model** namespace, point to **Create**, and then click **Namespace**.
3. Rename **New Namespace** to **Foundation Objects View**.
4. Right-click the **Foundation Objects View** namespace, point to **Create**, and then click **Namespace**.
5. Rename **New Namespace** to **gosales**.

   The model appears as shown below:

   Creating the gosales namespace makes the Foundation Objects View extensible in the event you want to import metadata from another data source. At the time of the import of the second data source, you would create another namespace to contain those data source query subjects.

   You will now begin the process for importing the metadata into the gosales namespace. You will start by creating a data source and testing the connection to it.

Task 3. Create a data source and test the connection.

1. Right-click the **gosales** namespace, and then click **Run Metadata Wizard**.

   The Metadata Wizard appears. From here you can select the source from which you will import the metadata. You will import data from a relational database.

2. Ensure that **Data Sources** is selected, and then click **Next**.

   You are now prompted to select a data source from the list of available data sources. The data source you need to access is currently not available. You must create a connection to the database from which you are importing metadata. You are connecting to the GOSALES database, which is a DB2 database.

3. Click **New**.

   The Welcome page of the New Data Source Wizard appears. This is the same wizard that is used to create data sources through the Directory tool in the IBM Cognos Connection administration interface.

4. Click **Next**, in the **Name** box, type **GOSALES**, and then click **Next**.

5. In the **Type** list, click **IBM DB2**, deselect the **Configure JDBC connection** checkbox, and then click **Next**.

   When you select the Configure JDBC connection checkbox, you are prompted to enter additional connection details that support Dynamic Query Mode. This feature is discussed in detail in Optimize and Tune Framework Manager Models.

6. In the **DB2 database name** box, type **GS_DB**.

7. Under **Signons**, select the **Password** check box, in the **User ID** box, type **GOSALES**, and then in the **Password** and **Confirm password** boxes, type **Education1**.

8. Click **Test the connection**, and then click **Test**.

   The View the results - Test the connection page appears indicating that the test succeeded.

9. Click **Close**, and then click **Close** again.

10. Click **Finish**, and then click **Close**.

    A new data source called GOSALES, through which you will import metadata, now appears in the list.

    You will now take a moment to view this data source in IBM Cognos Connection.

11. Launch **IBM Cognos Connection**, log on, click on **Administer IBM Cognos content**, and then click the **Configuration** tab.

    The GOSALES data source appears in the list. You could have also created the data source here rather than through Framework Manager. If you need to edit the data source connection, you will need to do it in this location.

12. Close **IBM Cognos Connection**.

Task 4. Import metadata.

1. In Framework Manager, ensure the newly created **GOSALES** data source is selected, and then click **Next**.

2. In the list of objects, expand **GOSALES** > **Tables**.

3. Select the following initial tables:

   ORDER_DETAILS
   ORDER_HEADER
   ORDER_METHOD
   PRODUCT
   PRODUCT_LINE
   PRODUCT_NAME_LOOKUP
   PRODUCT_TYPE

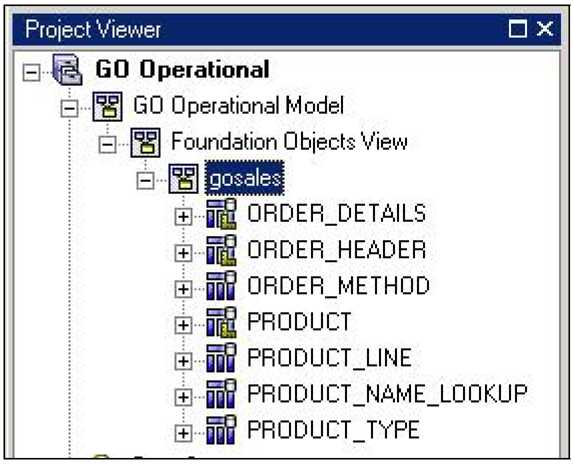
4. Click **Next**, leave the defaults for the **Generate Relationship** criteria, and then click **Import**.

   The import process begins, and then a message appears summarizing the seven query subjects and six relationships that were imported.

5. Click **Finish**.

6. In the **Project Viewer** pane, expand the **gosales** namespace.

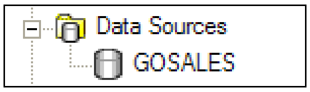
   The results appear as follows:



   The namespace now contains a list of data source query subjects, which represent each of the tables that were imported from the relational database.

7. Expand the **Data Sources** folder.

   The results appear as follows:



Data Sources
GOSALES

   This folder now contains the GOSALES data source, which you specified during the import. This data source is now associated with the objects you just imported.

8. Select the **GOSALES** data source.

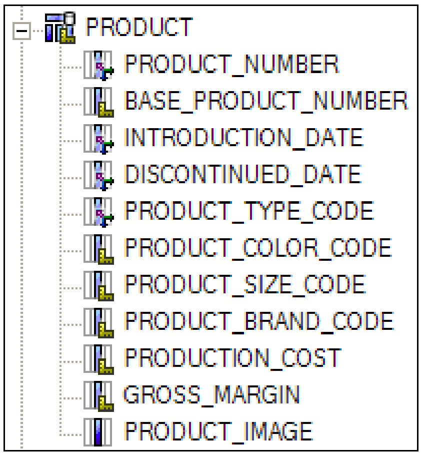
9. In the **Properties** pane, ensure the **Query Processing** property is set to **Limited Local**.

   This will allow IBM Cognos to process operations locally that are not supported by the data source.

Task 5. Test basic data access.

1. In the **Project Viewer**, expand **PRODUCT**.

   The results appear as follows:



PRODUCT
PRODUCT_NUMBER
BASE_PRODUCT_NUMBER
INTRODUCTION_DATE
DISCONTINUED_DATE
PRODUCT_TYPE_CODE
PRODUCT_COLOR_CODE
PRODUCT_SIZE_CODE
PRODUCT_BRAND_CODE
PRODUCTION_COST
GROSS_MARGIN
PRODUCT_IMAGE

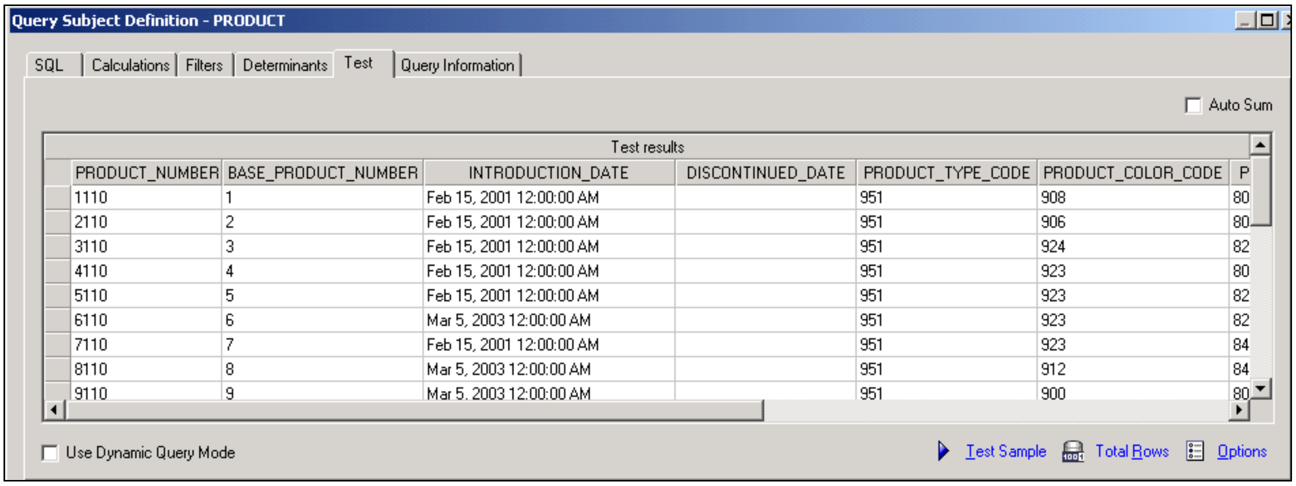
   Here you see all the query items for this query subject. These query items represent the columns found in the data source table.

2. Double-click **PRODUCT**.

   The SQL tab shows "Select * from [GOSALES].PRODUCT". This SQL statement defines the scope of the PRODUCT query subject. It is used to generate your query item list as well as the run time SQL when you create a query using the query items. The [GOSALES] portion is the name of the data source the query subject is associated with.

3. Click the **Test** tab, and then click **Test Sample** in the bottom right corner.

   The results appear as follows:



Test results

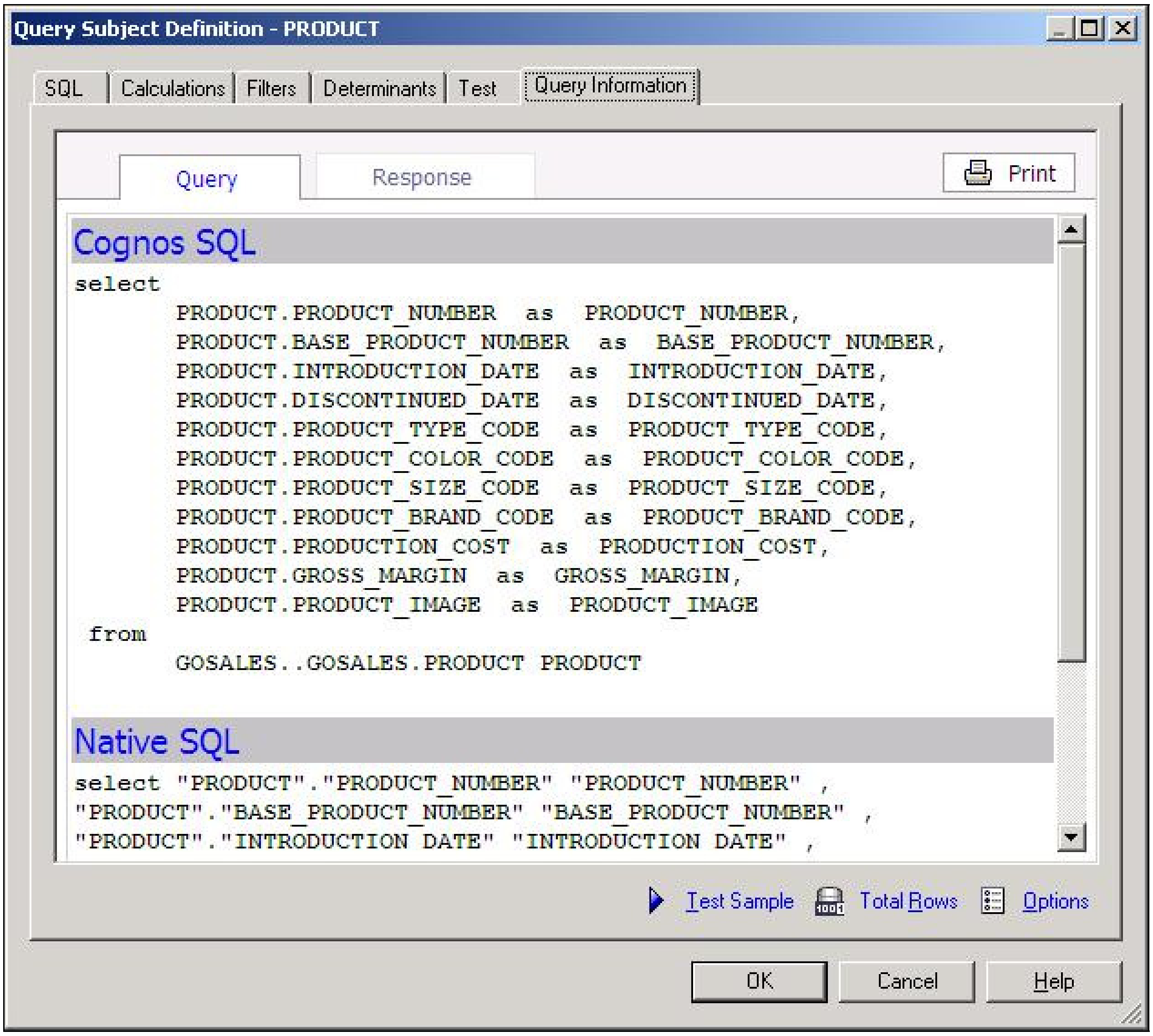
PRODUCT_NUMBER	BASE_PRODUCT_NUMBER	INTRODUCTION_DATE	DISCONTINUED_DATE	PRODUCT_TYPE_CODE	PRODUCT_COLOR_CODE	P
1110	1	Feb 15, 2001 12:00:00 AM		951	908	80
2110	2	Feb 15, 2001 12:00:00 AM		951	906	80
3110	3	Feb 15, 2001 12:00:00 AM		951	924	82
4110	4	Feb 15, 2001 12:00:00 AM		951	923	80
5110	5	Feb 15, 2001 12:00:00 AM		951	923	82
6110	6	Mar 5, 2003 12:00:00 AM		951	923	82
7110	7	Feb 15, 2001 12:00:00 AM		951	923	84
8110	8	Mar 5, 2003 12:00:00 AM		951	912	84
9110	9	Mar 5, 2003 12:00:00 AM		951	900	80

The test sample contains the first 25 records, confirming your access to the data. You can increase or decrease the amount of rows retrieved in the Options located in the lower right corner. If you want to know how many rows of data there are in the data source table, click Total Rows.

As for the SAP BW Time Dependent Hierarchies, this is an example of something that will now be imported properly into Framework Manager where in previous version adjustments had to be made. As of v10.1, there is no modeling to do on these data items once imported.
Time-dependant hierarchies now automatically reflect hierarchy or structure changes. When a structure is imported into Framework Manager, each SAP BW time hierarchy is depicted as an individual level. Report Studio users can use these structures to report on and compare levels that are valid for a specific time period.

4. Click the **Query Information** tab.

   The results appear as follows:



```
select
       PRODUCT.PRODUCT_NUMBER  as  PRODUCT_NUMBER,
       PRODUCT.BASE_PRODUCT_NUMBER  as  BASE_PRODUCT_NUMBER,
       PRODUCT.INTRODUCTION_DATE  as  INTRODUCTION_DATE,
       PRODUCT.DISCONTINUED_DATE  as  DISCONTINUED_DATE,
       PRODUCT.PRODUCT_TYPE_CODE  as  PRODUCT_TYPE_CODE,
       PRODUCT.PRODUCT_COLOR_CODE  as  PRODUCT_COLOR_CODE,
       PRODUCT.PRODUCT_SIZE_CODE  as  PRODUCT_SIZE_CODE,
       PRODUCT.PRODUCT_BRAND_CODE  as  PRODUCT_BRAND_CODE,
       PRODUCT.PRODUCTION_COST  as  PRODUCTION_COST,
       PRODUCT.GROSS_MARGIN  as  GROSS_MARGIN,
       PRODUCT.PRODUCT_IMAGE  as  PRODUCT_IMAGE
 from
       GOSALES..GOSALES.PRODUCT PRODUCT
```

```
select "PRODUCT"."PRODUCT_NUMBER" "PRODUCT_NUMBER" ,
"PRODUCT"."BASE_PRODUCT_NUMBER" "BASE_PRODUCT_NUMBER" ,
"PRODUCT"."INTRODUCTION_DATE" "INTRODUCTION_DATE" ,
```

   You see the full SQL in two formats: the generic Cognos syntax (Cognos SQL) and the syntax specific to the data source (Native SQL). This is the SQL generated at run time and passed to the data source. The SQL generated here is based on the SQL defined on the SQL tab of the data source query subject.

5. Click **OK** to close the **Query Subject Definition** dialog box.

Task 6. Change SQL generation settings and specify a function list.

1. From the **Project** menu, click **Edit Governors**.

   These properties govern the queries that are generated based on the model. These will be described in more detail later in the course.

2. Clear the **Use WITH clause when generating SQL** checkbox, and then click **OK**.

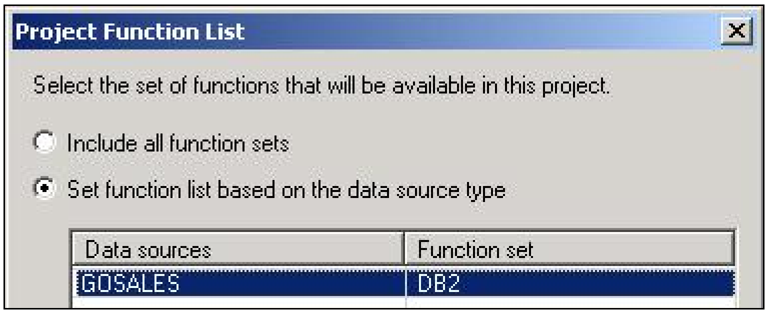
   You will now specify a function list specific to the data source you are using in this model. These functions are specific to the database vendor and allow authors and modelers to perform advanced tasks with the data sources such as date and string manipulations.

   Specifying a function set per data source will limit the function sets that are published to IBM Cognos with a package. By default, all supported function sets are published with a package. Publishing only the required functions sets can reduce the package size and prevent confusion where authors may inadvertently use functions from another data source.

3. From the **Project** menu, click **Project Function List**.

4. Select **Set function list based on the data source type**, and then in the **Function set** column beside **GOSALES**, select **DB2**.

   The results appear as follows:

   Project Function List

   Select the set of functions that will be available in this project.

   Include all function sets

   [✓ Set function list based on the data source type]

Data sources	Function set
GOSALES	DB2

   Now, by default, only DB2 functions will be published with a package in this project.

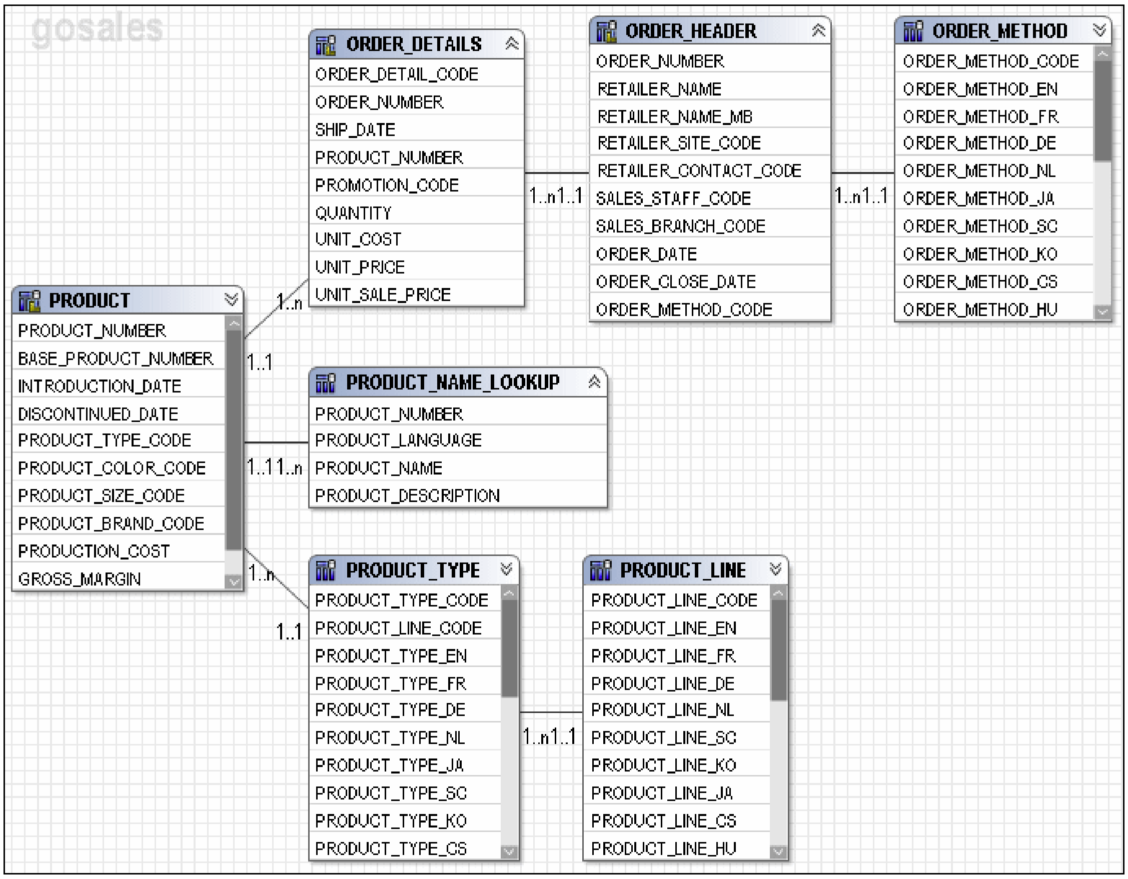
5. Click **OK**.

Task 7. Test multiple query subject data access.

1. If necessary, in the center pane click **Diagram**.
2. Close the **Properties** and **Tools** panes to view more of the center pane diagram, and then double-click the **gosales** box to focus on the **gosales** namespace.

3. Click **Fit All** .

The results appear as follows:



Note: Here you can see all the relationships generated by Framework Manager during import. These relationships allow IBM Cognos to generate the appropriate SQL at run time to join query items from the different query subjects. You can adjust the settings of the diagram from the Diagram menu. There are several options. For example, you can choose to view cardinalities in Merise notation (as seen in the screen capture) or Crowsfeet notation (provides a pictorial representation of the relationship). Take a moment to explore the options.

4. In the **diagram**, Ctrl-click to select the following query items:

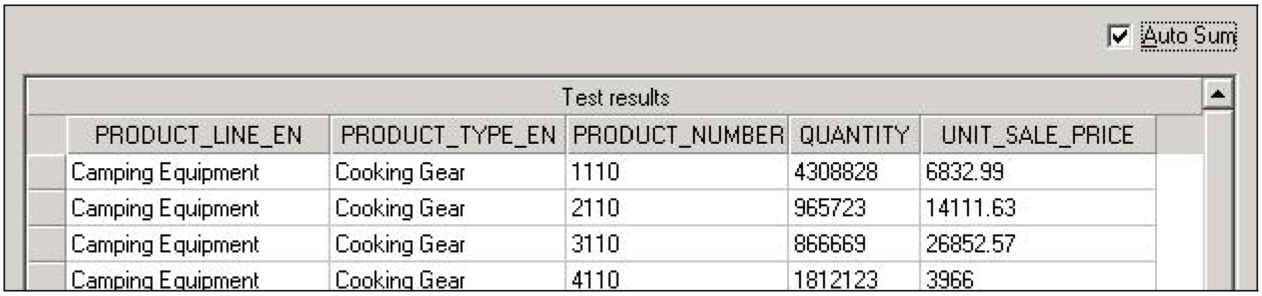
Query Subject	Query Item
PRODUCT_LINE	**PRODUCT_LINE_EN**
PRODUCT_TYPE	**PRODUCT_TYPE_EN**
PRODUCT	**PRODUCT_NUMBER**
ORDER_DETAILS	**QUANTITY** **UNIT_SALE_PRICE**

5. Right-click one of the selected items, click **Test**, and then click **Test Sample**.

   A sample 25-row report confirms that you can access the data. Note the values for quantity and unit sale price.

6. In the top right corner of the window, click **Auto Sum**, and then click **Test Sample** again.

   The results appear as follows:



Test results

PRODUCT_LINE_EN	PRODUCT_TYPE_EN	PRODUCT_NUMBER	QUANTITY	UNIT_SALE_PRICE
Camping Equipment	Cooking Gear	1110	4308828	6832.99
Camping Equipment	Cooking Gear	2110	965723	14111.63
Camping Equipment	Cooking Gear	3110	866669	26852.57
Camping Equipment	Cooking Gear	4110	1812123	3966

   This time, you see that QUANTITY has been summed. UNIT_SALE_PRICE has also been summed, which we will correct later.

7. Close the **Test Results** dialog box, and then save the project.

Results:
You started to develop the model that will support the business requirements of report and ad hoc query authors. You created a project, structured it, and imported baseline metadata.

Demo 3: Publish a Package

Purpose:
Now that you have a baseline project, you will verify its reporting capabilities by publishing a package and using it to view data in IBM Cognos Workspace Advanced.

Components:	**Framework Manager**, **IBM Cognos Workspace Advanced**
Project:	**GO Operational**
Package:	**GO Operational**

Task 1. Create and publish a package.

1. In the **Project Viewer** pane, right-click **Packages**, point to **Create**, and then click **Package**.

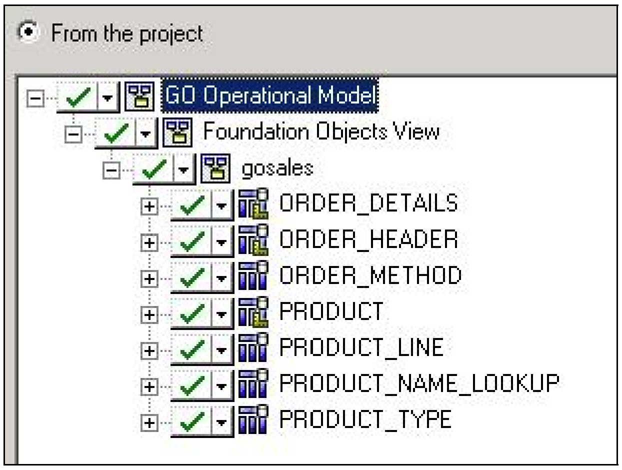
   The Create Package Wizard appears. Since the package will include the full contents of the project, you will give it the same name as the project.

   Note: To be able to publish packages, a user must belong to the Report Administrators role, or have write privileges to the Public Folders area of Cognos Connection.

2. In the **Name** box, type **GO Operational**, and then click **Next**.

3. Expand **Foundation Objects View** and **gosales**.

   The results appear as follows:



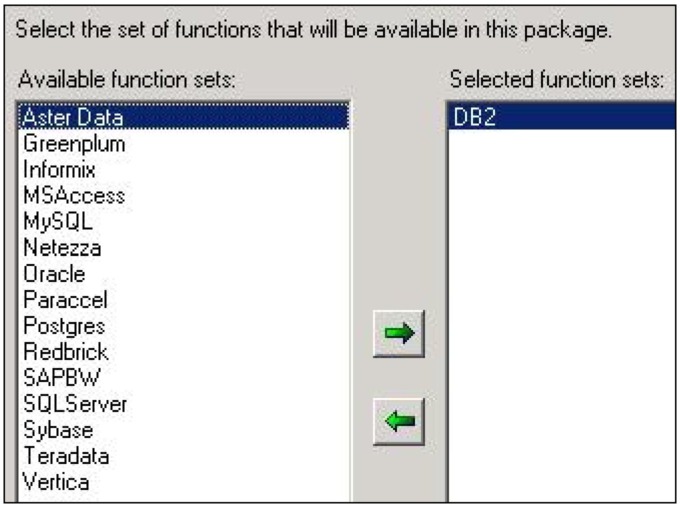
From the project

- GO Operational Model
  - Foundation Objects View
    - gosales
      - ORDER_DETAILS
      - ORDER_HEADER
      - ORDER_METHOD
      - PRODUCT
      - PRODUCT_LINE
      - PRODUCT_NAME_LOOKUP
      - PRODUCT_TYPE

   This is where you can restrict the package contents to a subset of your model.

4. Click **Next**.

   The results appear as follows:



Select the set of functions that will be available in this package.

Available function sets:

Aster Data
Greenplum
Informix
MSAccess
MySQL
Netezza
Oracle
Paraccel
Postgres
Redbrick
SAPBW
SQLServer
Sybase
Teradata
Vertica

Selected function sets:

DB2

   Here you have the option to specify which function sets get published with the package. Since you associated DB2 functions with the GOSALES data source, it is by default, the only function set currently selected.

5. Click **Finish**.

   A dialog box appears indicating that the package was created successfully, and prompts you to open the Publish Package wizard. Publishing places a package in the IBM Cognos Content Store, where it can be accessed from IBM Cognos Connection and the IBM Cognos Studios.

6. Click **Yes**.

   IBM Cognos allows you to maintain multiple versions of a package. This is done to prevent breaking reports when re-publishing a package with changes that affect the reports. You will not use this feature for most of this course.

7. Clear the **Enable model versioning** check box, accept the remaining defaults, and then click **Next**.

8. On the **Add Security** page, click **Next**, and then click **Publish**.

   The package is verified by default before the package is published.

   Once verified and published, a message appears within the wizard indicating that the package was successfully published.

9. Click **Finish** to close the wizard, and then save the project.

Task 2. View the package in IBM Cognos Workspace Advanced.

1. Open **Internet Explorer**, in the **Address** box, type **http://localhost:88/ibmcognos**, and then press **Enter**.

   The Log on page appears.

2. Log on as **admin/Education1**.

   The Welcome page of IBM Cognos Connection appears.

3. Under **My Actions**, click **Author business reports**.

   The Select a package page appears, and prompts you to choose a package to use in creating the report. Note that the GO Operational model you just published is available.

4. Click **GO Operational**.
5. In the **Welcome** dialog, click **Create new**.
6. In the **New** dialog, click **List** and then click **OK**.

   IBM Cognos Workspace Advanced appears in the browser. Under Insertable Objects on the right side of the page, you can see that the GO Operational root namespace and the Foundation Objects View namespace appear. These were the objects you specified in the Publish Package wizard in Framework Manager.
7. Expand **Foundation Objects View**, and then **gosales**.

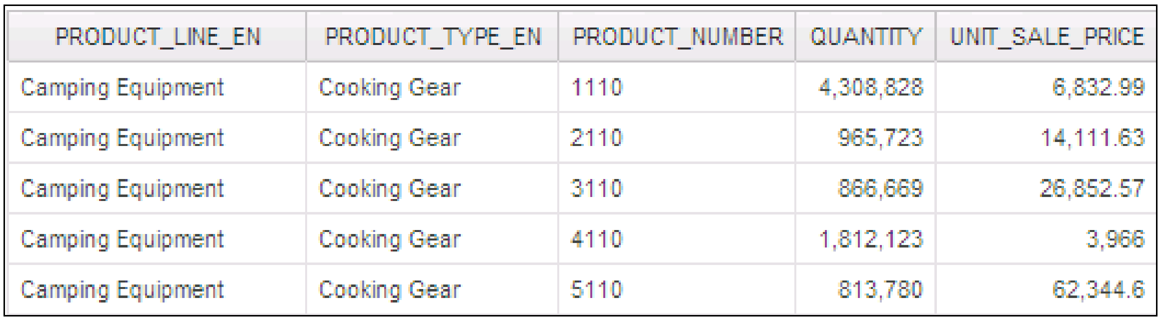
   The query subjects appear and are available to an ad hoc query author for creating a report.

Task 3. Create a report.

1. In the **gosales** namespace, expand the **PRODUCT_LINE** query subject, and then double-click the **PRODUCT_LINE_EN** query item.
2. Repeat step 1 to add the following items:

Query Subject	Query Item
PRODUCT_TYPE	**PRODUCT_TYPE_EN**
PRODUCT	**PRODUCT_NUMBER**
ORDER_DETAILS	**QUANTITY** **UNIT_SALE_PRICE**

The results appear as follows:

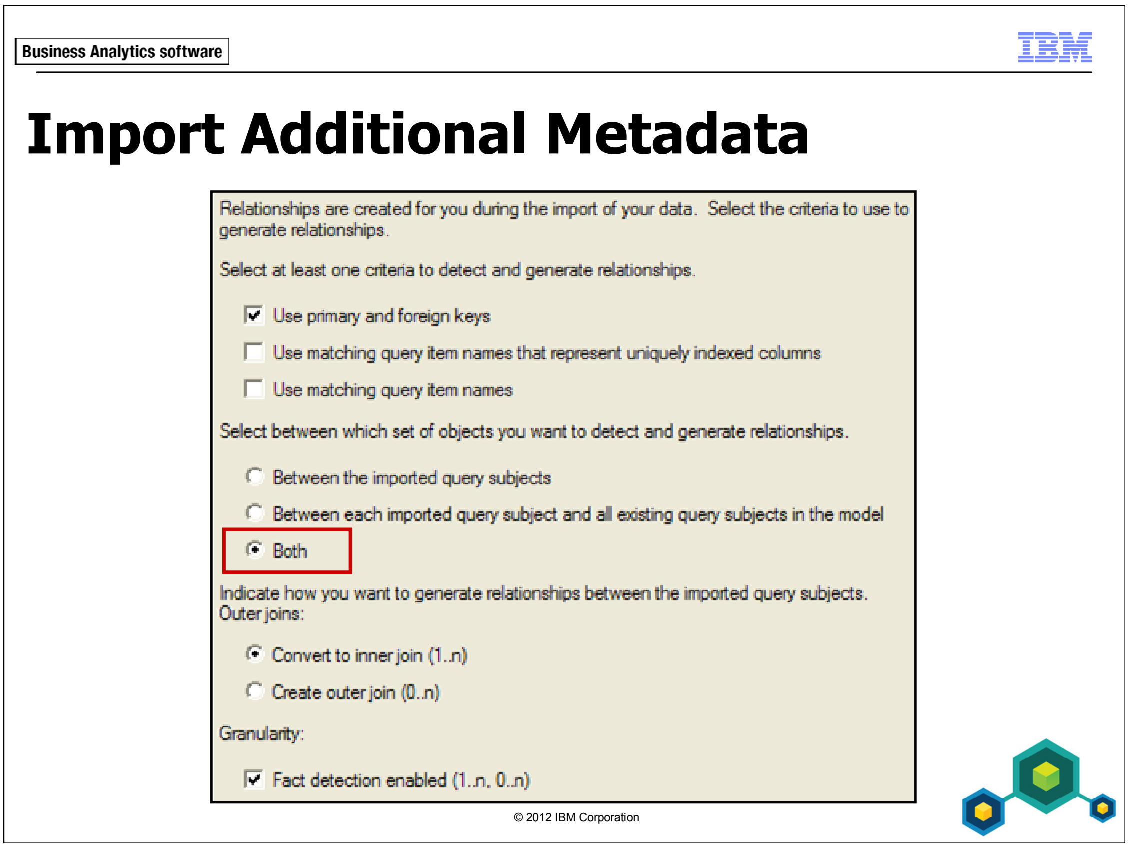


PRODUCT_LINE_EN	PRODUCT_TYPE_EN	PRODUCT_NUMBER	QUANTITY	UNIT_SALE_PRICE
Camping Equipment	Cooking Gear	1110	4,308,828	6,832.99
Camping Equipment	Cooking Gear	2110	965,723	14,111.63
Camping Equipment	Cooking Gear	3110	866,669	26,852.57
Camping Equipment	Cooking Gear	4110	1,812,123	3,966
Camping Equipment	Cooking Gear	5110	813,780	62,344.6

You have successfully created the same ad hoc query you tested in Framework Manager earlier. Here the values are automatically sorted and summarized.

3. Close **Internet Explorer**, do not save changes, and leave **Framework Manager** open.

Results:
You have verified the reporting capabilities of your new project by publishing a package from it and using the package to view data in IBM Cognos Workspace Advanced.



Business Analytics software

Import Additional Metadata

Relationships are created for you during the import of your data. Select the criteria to use to generate relationships.

Select at least one criteria to detect and generate relationships.

- [✓] Use primary and foreign keys
- [ ] Use matching query item names that represent uniquely indexed columns
- [ ] Use matching query item names

Select between which set of objects you want to detect and generate relationships.

- ○ Between the imported query subjects
- ○ Between each imported query subject and all existing query subjects in the model
- ● Both

Indicate how you want to generate relationships between the imported query subjects.
Outer joins:

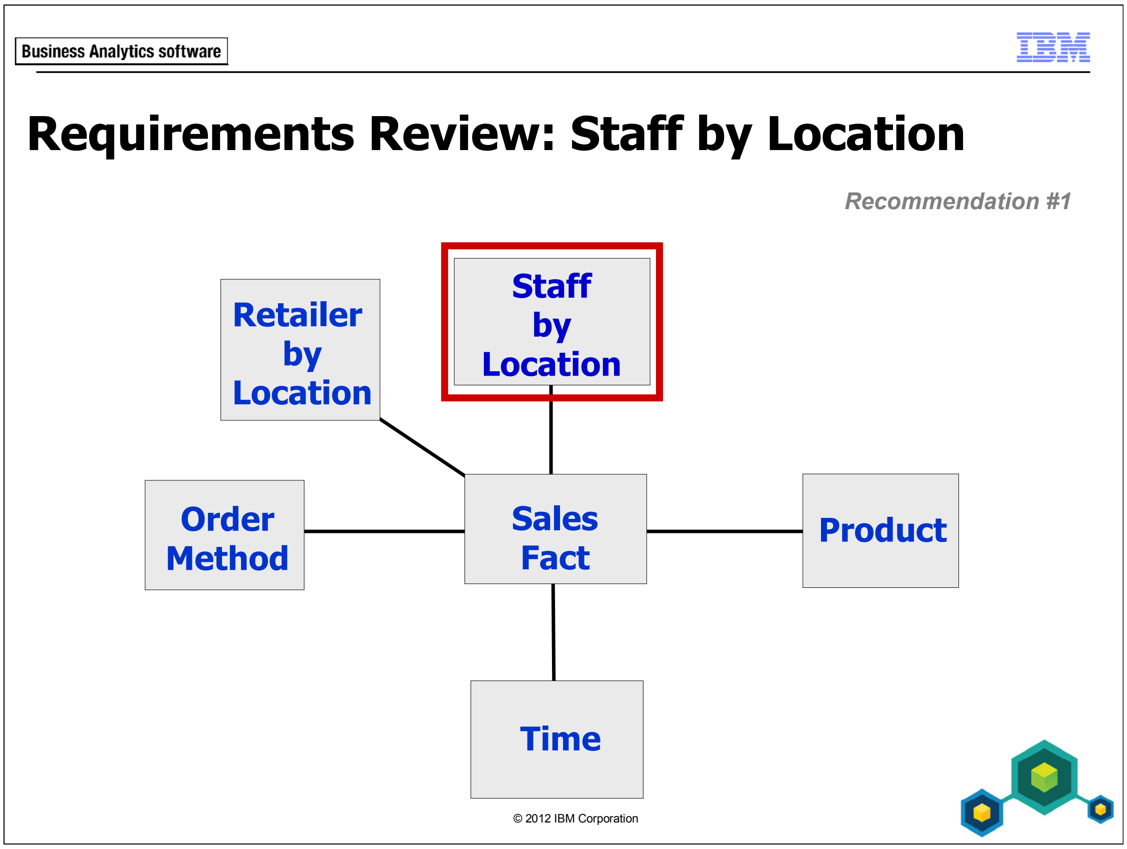
- ● Convert to inner join (1..n)
- ○ Create outer join (0..n)

Granularity:

- [✓] Fact detection enabled (1..n, 0..n)

© 2012 IBM Corporation

When importing additional metadata from the same data source or another compatible data source (primary and foreign keys match), select Both in the wizard. This generates relationships among the objects being imported as well as the existing objects in the model. By selecting this option, you will not need to create the relationships manually after import.



Business Analytics software

Requirements Review: Staff by Location

Recommendation #1

© 2012 IBM Corporation

In the next demo, you will import metadata to satisfy the highlighted reporting requirement above. The database tables used to satisfy this requirement are as follows:

Reporting Requirement	**Database Tables**
Staff by Location	BRANCH COUNTRY SALES_REGION EMPLOYEE EMPLOYEE_HISTORY POSITION_LOOKUP

Demo 4: Extend the Model to Add Staff Location Metadata

Purpose:
Now that you have an understanding of the Framework Manager modeling environment and process, you can extend your model to support additional business requirements from report and ad hoc query authors, namely reporting sales by sales staff location.

Component:	**Framework Manager**
Project:	**GO Operational**

Task 1. Import additional metadata.

1. In the **Project Viewer**, in **Foundation Object view**, right-click the **gosales** namespace, and then click **Run Metadata Wizard**.

   You will import more items from the GOSALES database.
2. Ensure that **Data Sources** is selected, and then click **Next**.
3. Ensure that **GOSALES** is selected, and then click **Next**.
4. In the list of objects, expand **GOSALES** and **Tables**, and then select the following tables:

   BRANCH

   COUNTRY

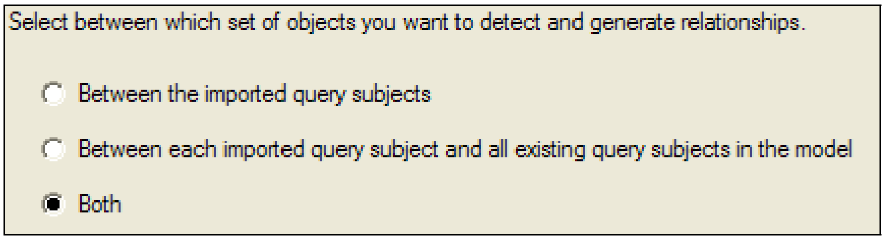
   SALES_REGION
5. Expand **GOSALESHR** and **Tables**, and then select the following tables:

   EMPLOYEE

   EMPLOYEE_HISTORY

   POSITION_LOOKUP.
6. Click **Next**.

7. On the **Generate Relationships** page, click the **Both** radio button as shown below.


Select between which set of objects you want to detect and generate relationships.

- ○ Between the imported query subjects
- ○ Between each imported query subject and all existing query subjects in the model
- ● Both

   You want to generate relationships among the new query subjects as well as between the new and existing query subjects. This will prevent you from having to manually create the relationships after the import.

8. Click **Import**.

   The import process begins, and then a message appears summarizing the count of objects that were imported.

9. Click **Finish**.

   The gosales namespace now contains a list of query subjects, which represent each of the tables that were imported from the relational database.

10. From the **View** menu, click **Properties**.
11. If necessary, expand the **Data Sources** folder, and then click **GOSALES1**.

   The second data source, GOSALES1, was created for the query subjects you selected from the second schema, gosaleshr.

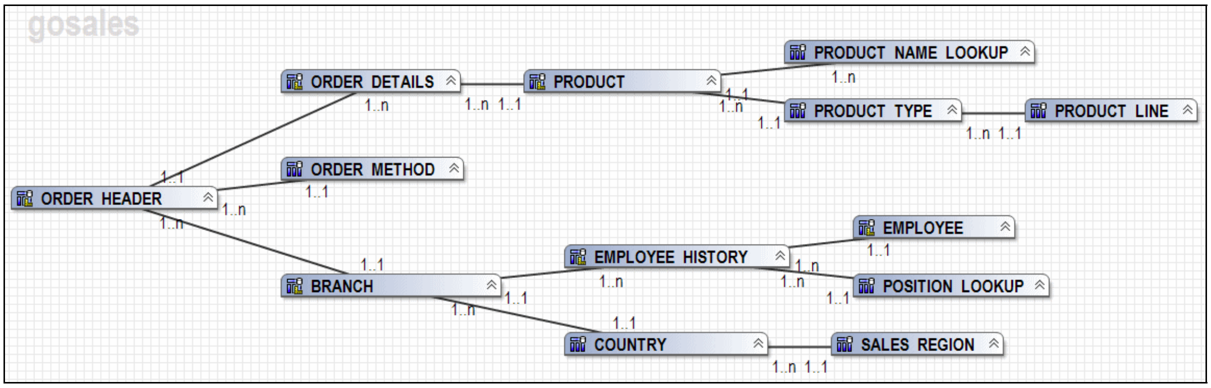
Task 2. Rearrange the diagram.

1. From the **Diagram** menu, click **Diagram settings**.
2. Under **Level of Details**, deselect **Query Items**, and then click **OK**.
3. Close the **Properties** pane, ensure **Diagram** is selected and then, from the toolbar, click **Auto Layout** .

Task 2, Step 1: In the next module, once we have several more query subjects, we'll demo the Context Explorer as a more focused alternative.

4. Beside **Layout Style**, ensure that **Standard** is selected, and then set both **Horizontal** and **Vertical** distances to **30**, click **Apply**, and then click **Close**.

   The results appear as follows:



   You can now view all the items imported into your project in a simplified way by reducing the detail and adjusting the space between the objects.

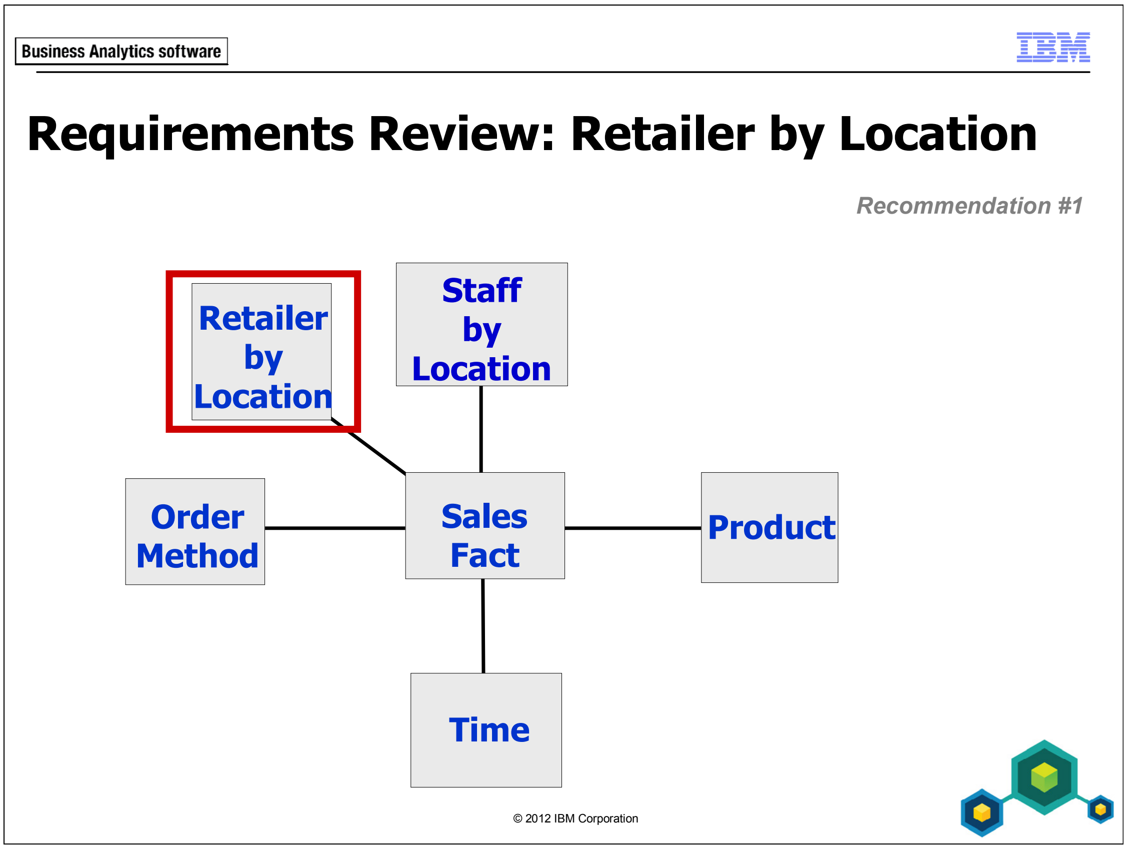
   If you were importing a star schema structure, you would select the Star option under Layout Style to set fact query subjects as focal points with their related dimensions surrounding them.

   You will identify and correct any missing relationships in the next module.

Results:
You extended the model to support additional business requirements for report and ad hoc query authors by importing metadata to support sales staff location queries.

Task 2: An alternative is to manually rearrange this diagram, or do some manual adjustment after Auto Layout. You may also wish to review the results of both the Standard and Star dropdown settings.

You can also change the Scale manually to a different percentage such as 75% to make the diagram more readable.


Business Analytics software

Requirements Review: Retailer by Location

Recommendation #1

© 2012 IBM Corporation

In the next workshop, you will import metadata to satisfy the highlighted reporting requirement above. The database tables used to satisfy this requirement are as follows:

Reporting Requirement	Database Tables
Retailer	RETAILER RETAILER_SITE RETAILER_TYPE

Workshop 1: Enhance the Model to Add Retailer Data

Business analysts at The Sample Outdoors Company have additional reporting requirements that need to be met. They want to report on sales by retailer.

Retailer information is supported by the following tables in the data source:

RETAILER

RETAILER_SITE

RETAILER_TYPE

Import these query subjects, and then rearrange the diagram to show all query subjects.

Use the Test tool in Framework Manager to verify that you can report on the new data.

For more detailed information outlined as tasks, see the Task Table section on the next page.

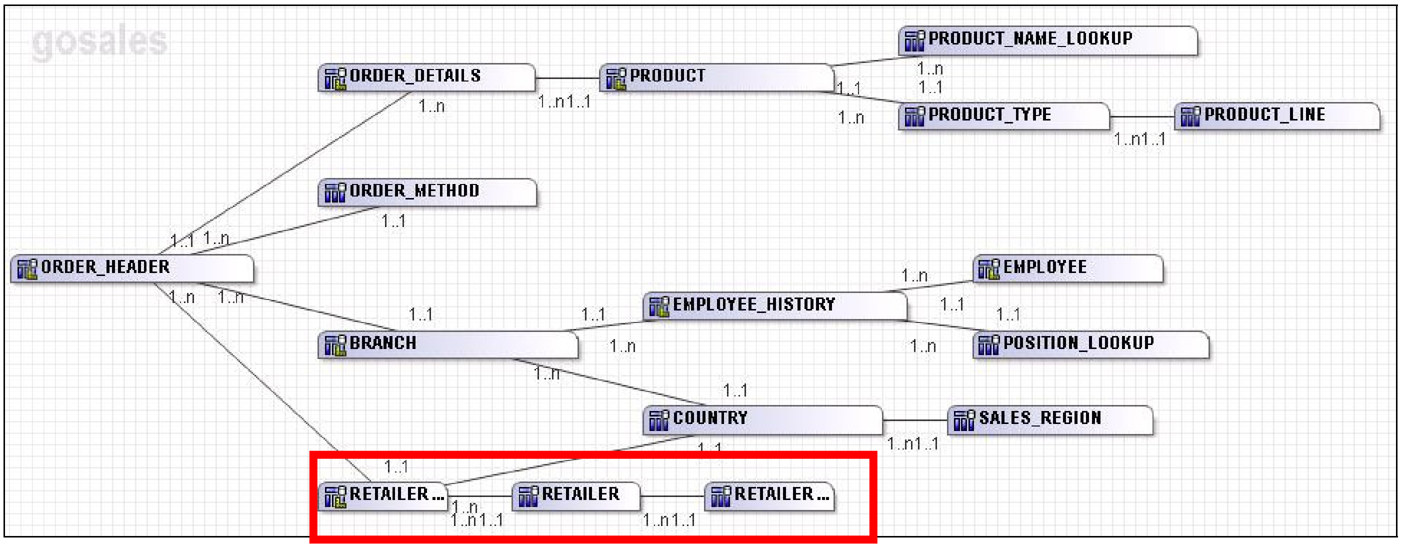
For the final query results, see the Workshop Results section that follows the Task Table section.

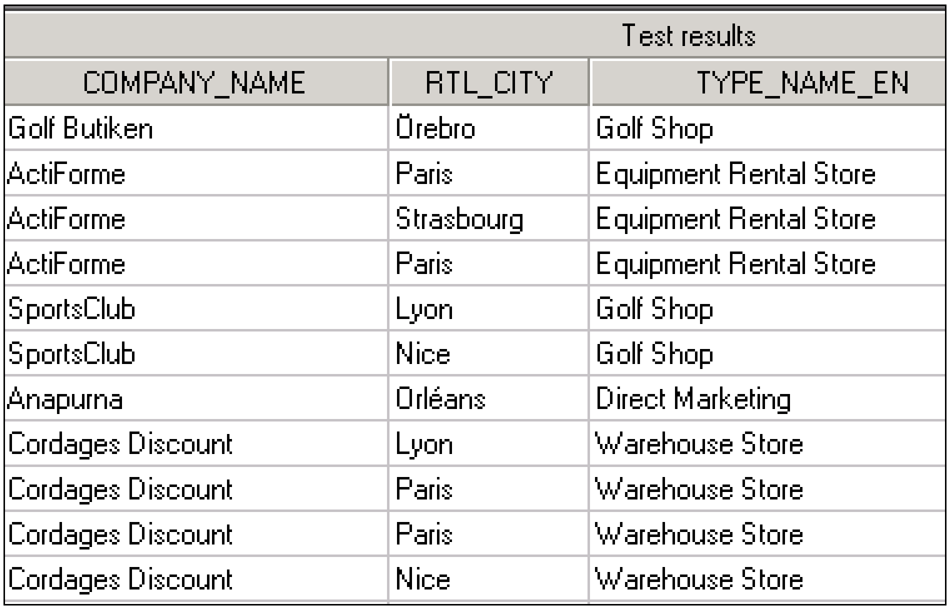
Workshop 1: Task Table

Task	Where to Work	Hints
1. Import metadata.	From gosales namespace, Run Metadata Wizard	• Data Source = GOSALES • Schema = GOSALESRT • Tables = RETAILER, RETAILER_SITE, RETAILER_TYPE • Generate relationships: Both
2. Organize the entity relationship diagram.	Center pane, Diagram setting	• Use Auto Layout with distances of 30
3. Test that you have access to the new data.	Project Viewer	• Select RETAILER.COMPANY_NAME and RETAILER_SITE.RTL_CITY, RETAILER_TYPE.TYPE_NAME_EN • Right-click one of the selected items, click Test, and then click Test Sample
4. Save and close the project.	File menu	• Save • Close Framework Manager

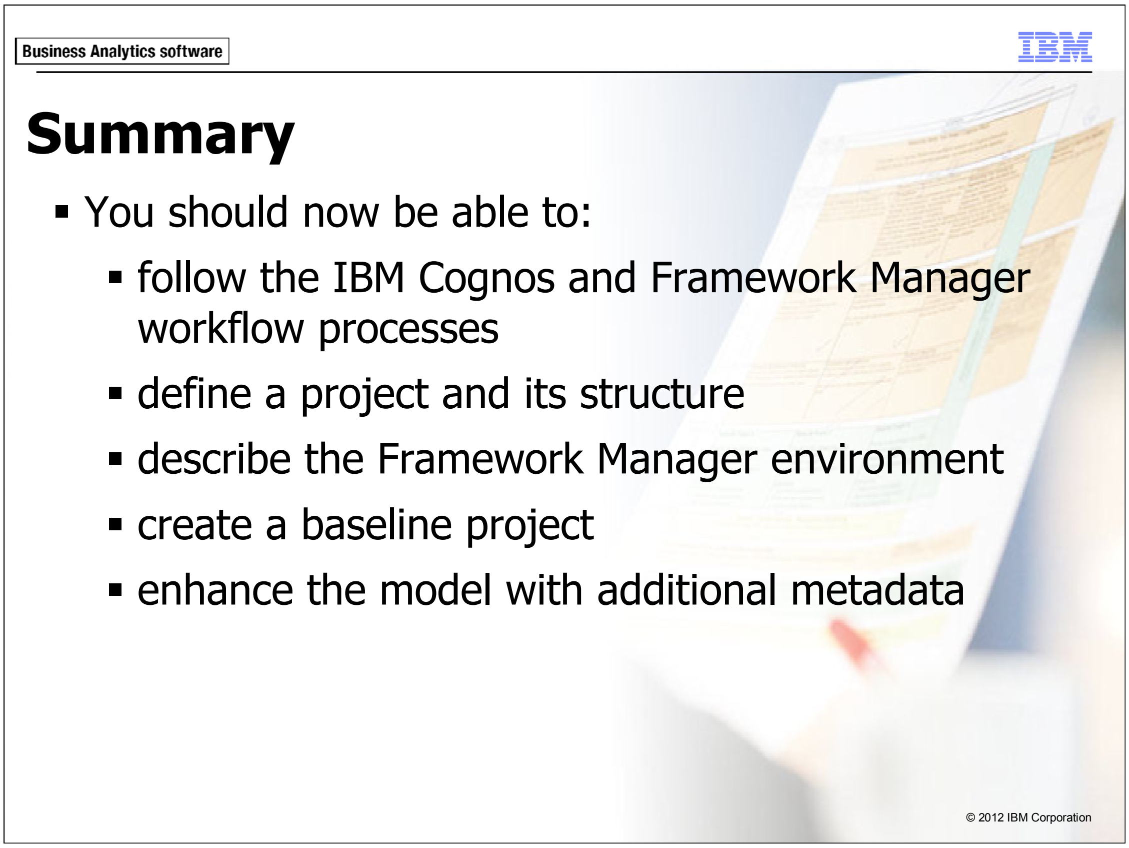
Workshop 1: Workshop Results

The resulting diagram should have the same query subjects and relationships as the following:





Test results		
COMPANY_NAME	RTL_CITY	TYPE_NAME_EN
Golf Butiken	Örebro	Golf Shop
ActiForme	Paris	Equipment Rental Store
ActiForme	Strasbourg	Equipment Rental Store
ActiForme	Paris	Equipment Rental Store
SportsClub	Lyon	Golf Shop
SportsClub	Nice	Golf Shop
Anapurna	Orléans	Direct Marketing
Cordages Discount	Lyon	Warehouse Store
Cordages Discount	Paris	Warehouse Store
Cordages Discount	Paris	Warehouse Store
Cordages Discount	Nice	Warehouse Store

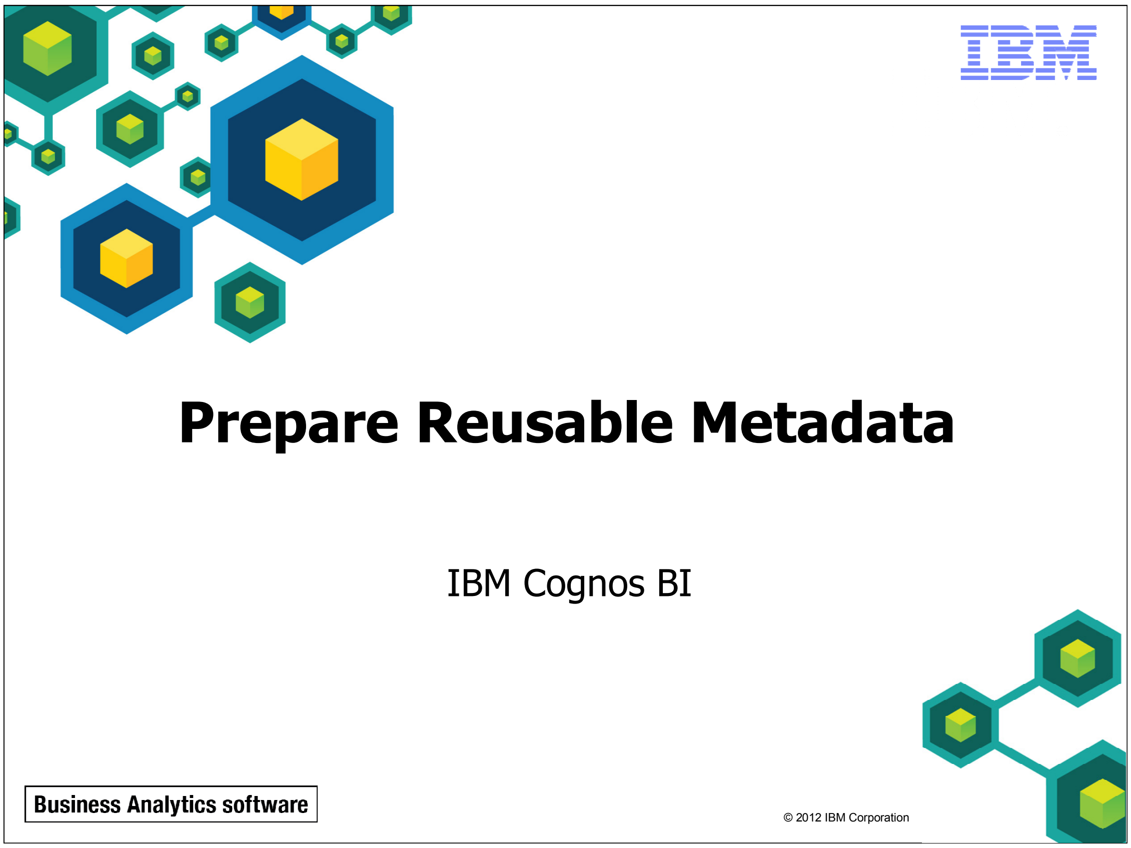


Business Analytics software

Summary

- You should now be able to:
  - follow the IBM Cognos and Framework Manager workflow processes
  - define a project and its structure
  - describe the Framework Manager environment
  - create a baseline project
  - enhance the model with additional metadata

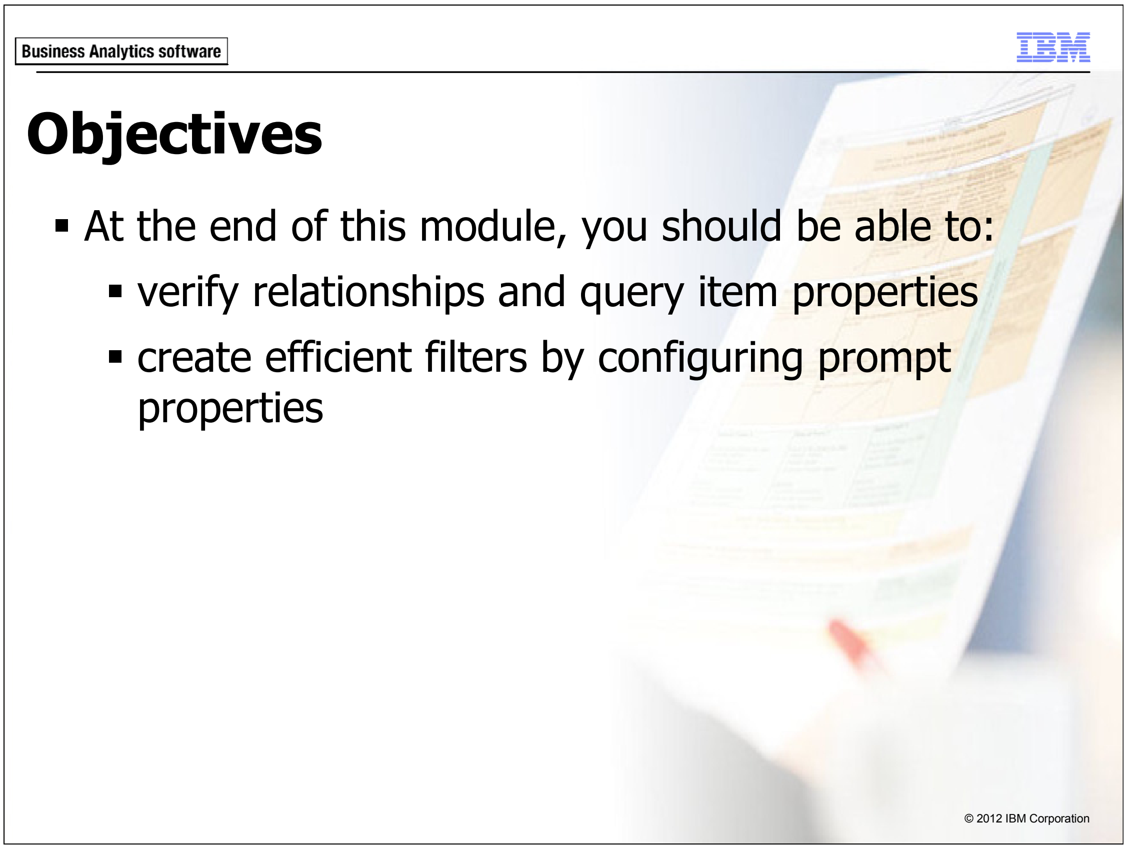
© 2012 IBM Corporation



Prepare Reusable Metadata

IBM Cognos BI

Business Analytics software


Business Analytics software

IBM

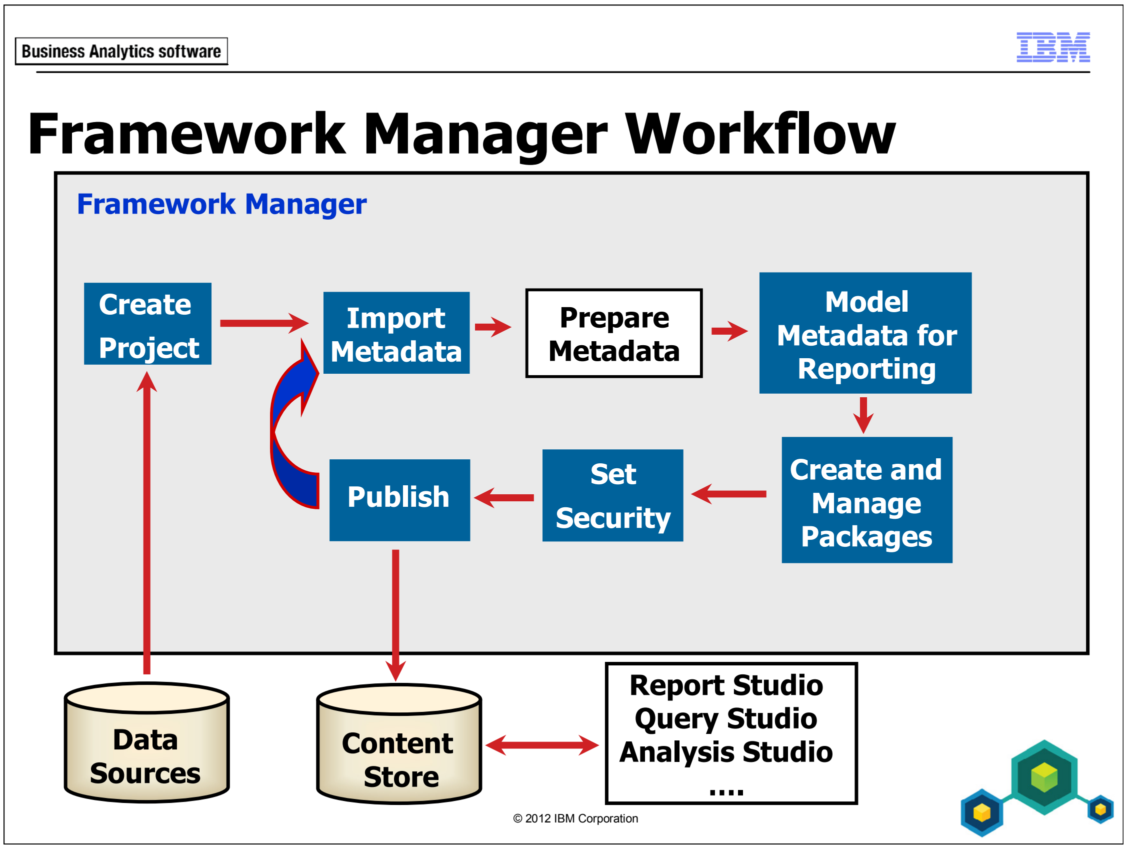
Objectives

- At the end of this module, you should be able to:
  - verify relationships and query item properties
  - create efficient filters by configuring prompt properties

© 2012 IBM Corporation

The demos and workshops in this module represent the early phases of model development. You have gathered requirements and will now begin to design the metadata models that meet those requirements. The solutions in this module are not necessarily meant to be the final view seen by report authors. You test the results of the metadata as you design the model.

Before reviewing this module, you should review the following modules:
- Overview of IBM Cognos
- Identify Common Data Structures
- Gather Requirements
- Create a Baseline Project



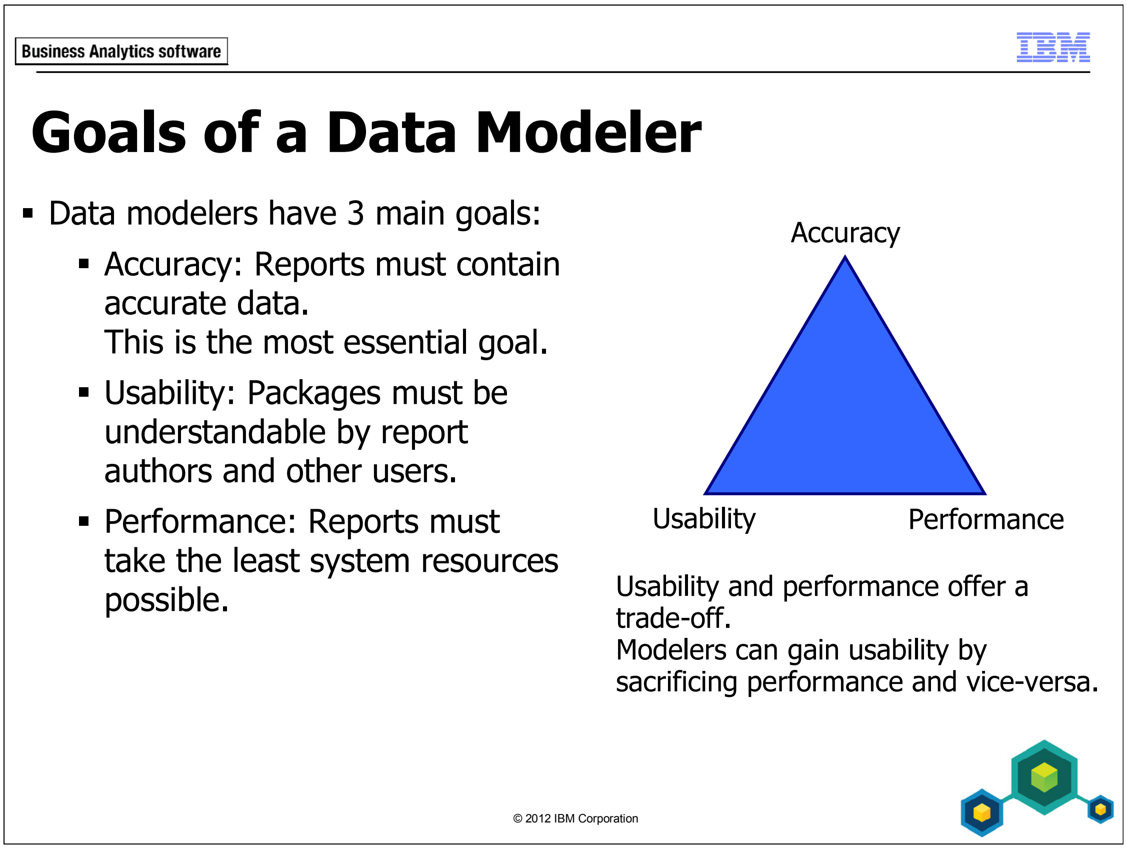
Business Analytics software

Framework Manager Workflow

© 2012 IBM Corporation

The Prepare Metadata phase involves verifying and modifying relationships and object properties and customizing metadata for run time.

These changes can be made either to the original data source query subjects (in the Foundation Objects View), or to the model query subjects (such as those you will create later in the course). Changes to the original data source query subjects are reusable. These query subjects may later be used in several different model query subjects, each of which will inherit the same changes. However, if you do not intend for a modification to be universal, then it should be made in the specific model query subjects that require that change.



Business Analytics software

Goals of a Data Modeler

- Data modelers have 3 main goals:
  - Accuracy: Reports must contain accurate data.
    This is the most essential goal.
  - Usability: Packages must be understandable by report authors and other users.
  - Performance: Reports must take the least system resources possible.

Accuracy
Usability
Performance

Usability and performance offer a trade-off.
Modelers can gain usability by sacrificing performance and vice-versa.

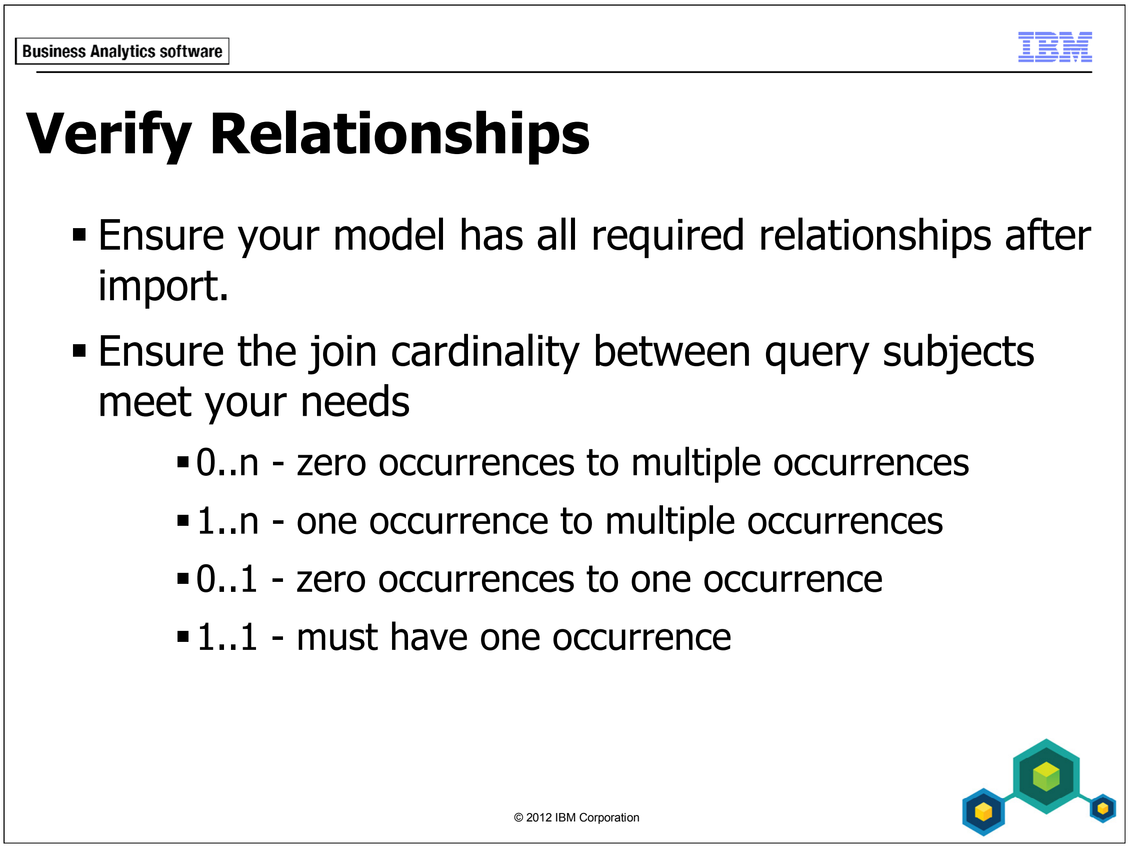
© 2012 IBM Corporation

The goals of the data modeling process can be summarized in three categories:

- **Accuracy:** If the packages created from a model do not produce reports with accurate data, the model is inherently flawed and must be repaired.
- **Usability:** The packages produced from a model must be understandable by those who will use the packages to create reports and perform analysis.
- **Performance:** Data should be retrievable in a reasonable amount of time.

Usability concerns can vary depending on the database understanding and skill level of your authors and analysts. Usability will not always affect performance, but it can. Several modeling techniques can slow down data retrieval. As a modeler, you must determine whether usability or performance is a higher priority for your authors and analysts. Remember, accuracy is not negotiable.

We will examine various methods you can use to improve the performance of your IBM Cognos data retrieval later in this course.



Business Analytics software

Verify Relationships

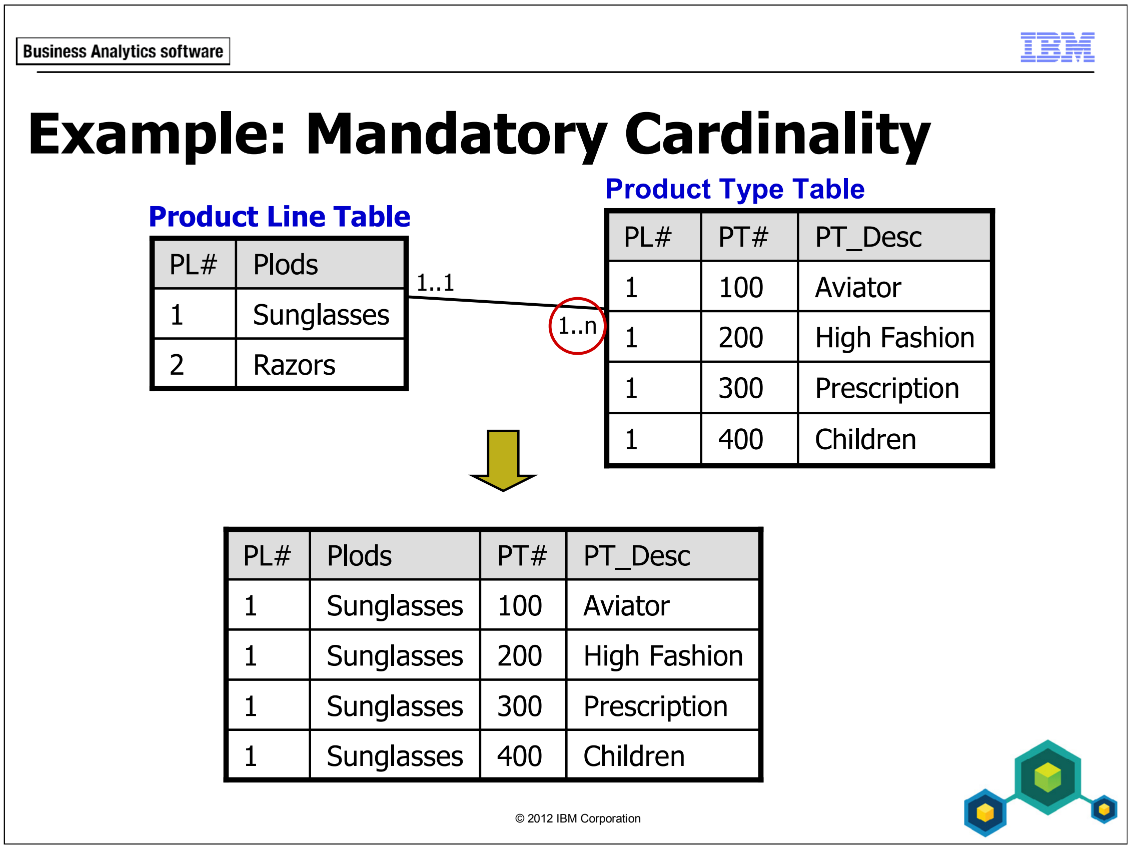
- Ensure your model has all required relationships after import.
- Ensure the join cardinality between query subjects meet your needs
  - 0..n - zero occurrences to multiple occurrences
  - 1..n - one occurrence to multiple occurrences
  - 0..1 - zero occurrences to one occurrence
  - 1..1 - must have one occurrence

© 2012 IBM Corporation

Relationships are maintained in the Object Diagram or Context Explorer.

When verifying your relationships, you must ensure that the appropriate relationships exist to meet your reporting needs and you must decide if you require optional or mandatory cardinalities. Optional cardinalities require more processing, but may be needed to return the desired results.

Optional cardinality is represented by a 0 as seen in the 0..n and 0..1 examples.



Business Analytics software

Example: Mandatory Cardinality

Product Line Table

PL#	Plods
1	Sunglasses
2	Razors

1..1

1..n

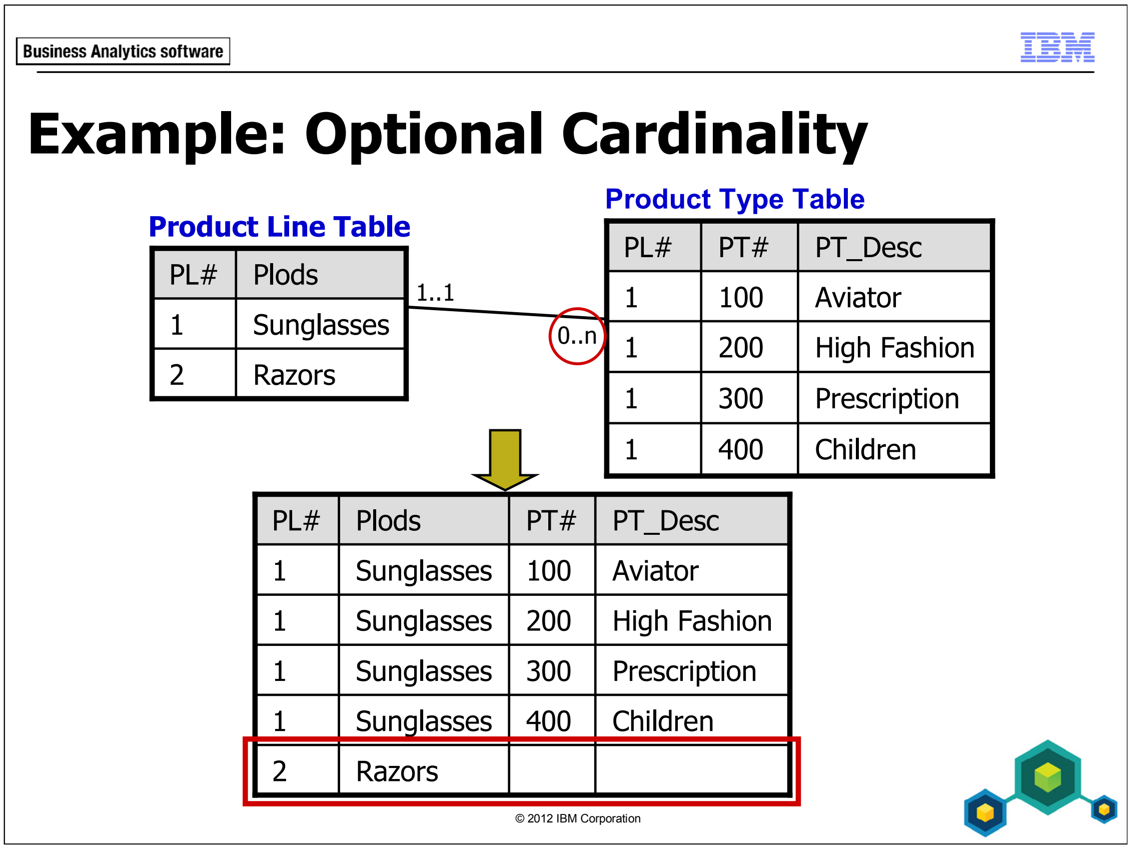
Product Type Table

PL#	PT#	PT_Desc
1	100	Aviator
1	200	High Fashion
1	300	Prescription
1	400	Children

PL#	Plods	PT#	PT_Desc
1	Sunglasses	100	Aviator
1	Sunglasses	200	High Fashion
1	Sunglasses	300	Prescription
1	Sunglasses	400	Children

© 2012 IBM Corporation

In the slide example, only rows in which the join condition is met are returned (where the product line number exists in both tables). The Razors product line is omitted from the record set.



Business Analytics software

Example: Optional Cardinality

Product Line Table

PL#	Plods
1	Sunglasses
2	Razors

Product Type Table

PL#	PT#	PT_Desc
1	100	Aviator
1	200	High Fashion
1	300	Prescription
1	400	Children

PL#	Plods	PT#	PT_Desc
1	Sunglasses	100	Aviator
1	Sunglasses	200	High Fashion
1	Sunglasses	300	Prescription
1	Sunglasses	400	Children
2	Razors		

Optional cardinality is represented by a 0 as seen in the 0..n in the above example.

With the cardinality set to 0..n on the Product Type table, all rows from the Product Line table are returned since a mandatory match in the Product Line table is not required. The Razors product line now appears in the record set.

In this course, we primarily use mandatory cardinalities (inner joins), as they provide better performance. However, if you require optional cardinality (outer join for reporting purposes, you can implement it, and optimize performance in other ways, such as adding filters, to offset the performance impact of outer joins.

Relationships are maintained in the Object Diagram or Context Explorer.

When verifying your relationships, you must ensure that the appropriate relationships exist to meet your reporting needs and you must decide if you require optional or mandatory cardinalities. Optional cardinalities require more processing, but may be needed to return the desired results.

Demo 1: Verify Relationships

Purpose:
To meet reporting requirements, you will verify your relationships to ensure none are missing and that the cardinality is set accordingly.

Component:	**Framework Manager**
Project:	**GO Operational**

Task 1. Examine relationships.

1. In **Framework Manager**, open the **GO Operational** project located at **C:\Edcognos\B5252\CBIFM-Start Files\ Module 5\GO Operational**.

   This is the baseline project created in the previous module.

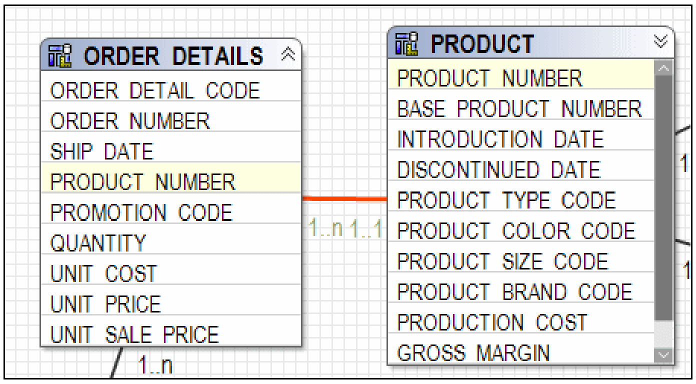
2. Log in as User ID **admin**, and Password **Education1**, if prompted.
3. In the middle pane, click **Diagram**, and then double-click the **gosales** box to give it focus in the diagram.

   You will increase the level of detail in the diagram to include query items.

4. From the **Diagram** menu, click **Diagram settings**, select the **Query Items** checkbox, and then click **OK**.
5. Using the **Auto Layout** dialog box, set the horizontal and vertical distance to **25** to increase the distance between objects.

6. Click the relationship between **ORDER_DETAILS** and **PRODUCT**.

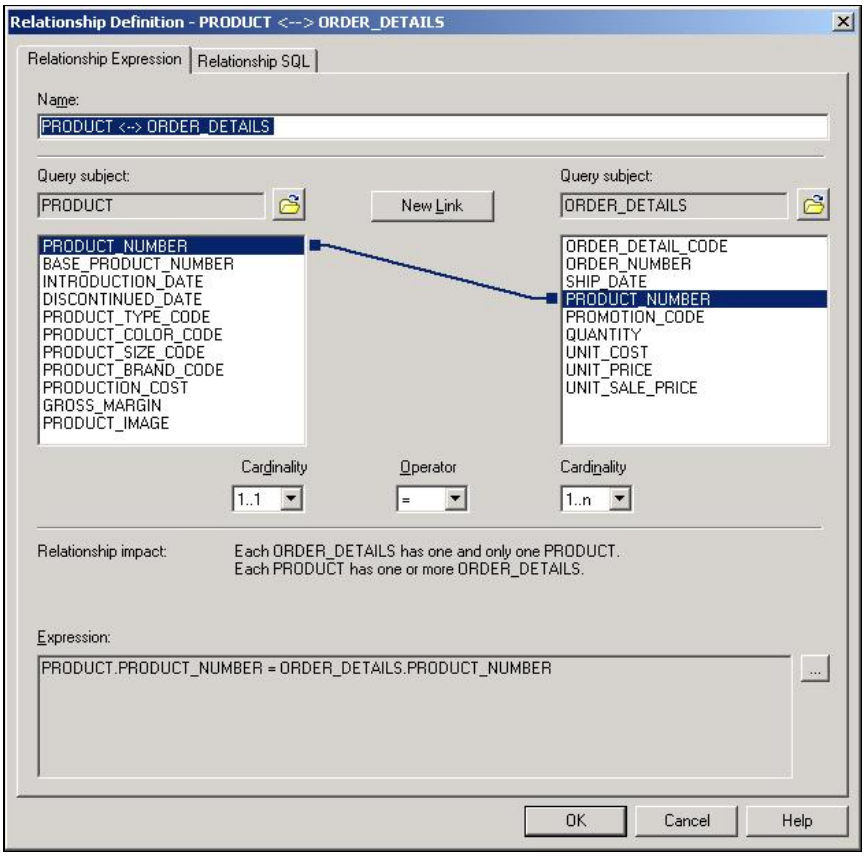
   The results appear as follows:



   The relationship line turns red and the query items used in the join are highlighted, in this case PRODUCT_NUMBER.

7. Double-click the relationship.

   The results appear as follows:



Name: PRODUCT <--> ORDER_DETAILS

Query subject: PRODUCT

Query subject: ORDER_DETAILS

PRODUCT_NUMBER
BASE_PRODUCT_NUMBER
INTRODUCTION_DATE
DISCONTINUED_DATE
PRODUCT_TYPE_CODE
PRODUCT_COLOR_CODE
PRODUCT_SIZE_CODE
PRODUCT_BRAND_CODE
PRODUCTION_COST
GROSS_MARGIN
PRODUCT_IMAGE

ORDER_DETAIL_CODE
ORDER_NUMBER
SHIP_DATE
PRODUCT_NUMBER
PROMOTION_CODE
QUANTITY
UNIT_COST
UNIT_PRICE
UNIT_SALE_PRICE

Cardinality: 1..1
Operator: =
Cardinality: 1..n

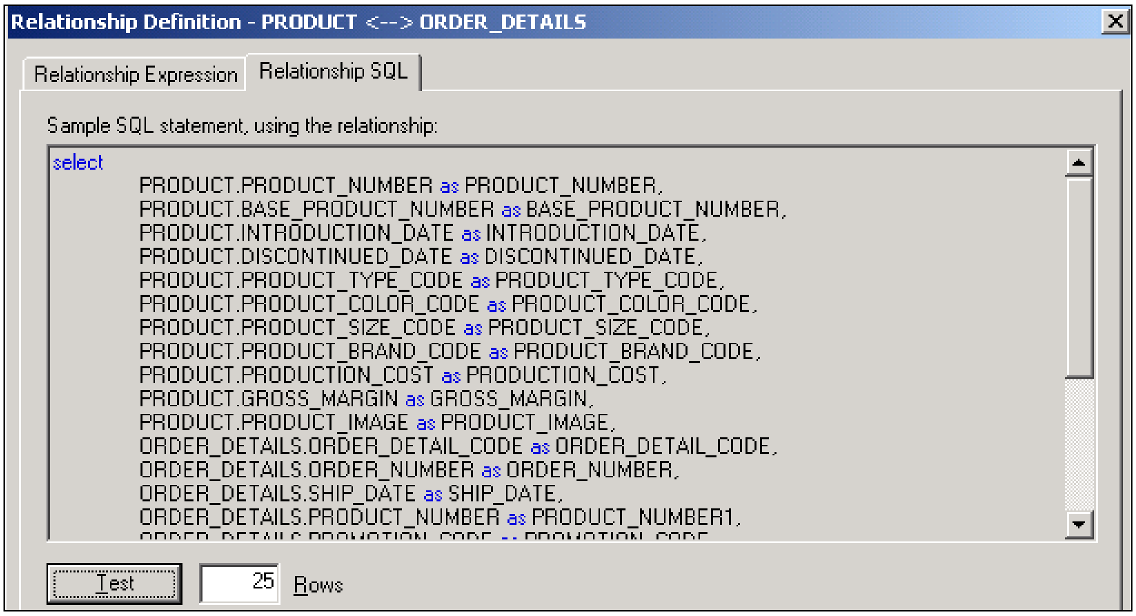
Relationship impact: Each ORDER_DETAILS has one and only one PRODUCT.
Each PRODUCT has one or more ORDER_DETAILS.

Expression:
PRODUCT.PRODUCT_NUMBER = ORDER_DETAILS.PRODUCT_NUMBER

You can change the query items used in the join, the cardinality settings, the operator, or edit the expression to create a more complex join.

8. Click the **Relationship SQL** tab.

   The results appear as follows:



Relationship Definition - PRODUCT <--> ORDER_DETAILS

Relationship Expression | Relationship SQL

Sample SQL statement, using the relationship:

```
select
            PRODUCT.PRODUCT_NUMBER as PRODUCT_NUMBER,
            PRODUCT.BASE_PRODUCT_NUMBER as BASE_PRODUCT_NUMBER,
            PRODUCT.INTRODUCTION_DATE as INTRODUCTION_DATE,
            PRODUCT.DISCONTINUED_DATE as DISCONTINUED_DATE,
            PRODUCT.PRODUCT_TYPE_CODE as PRODUCT_TYPE_CODE,
            PRODUCT.PRODUCT_COLOR_CODE as PRODUCT_COLOR_CODE,
            PRODUCT.PRODUCT_SIZE_CODE as PRODUCT_SIZE_CODE,
            PRODUCT.PRODUCT_BRAND_CODE as PRODUCT_BRAND_CODE,
            PRODUCT.PRODUCTION_COST as PRODUCTION_COST,
            PRODUCT.GROSS_MARGIN as GROSS_MARGIN,
            PRODUCT.PRODUCT_IMAGE as PRODUCT_IMAGE,
            ORDER_DETAILS.ORDER_DETAIL_CODE as ORDER_DETAIL_CODE,
            ORDER_DETAILS.ORDER_NUMBER as ORDER_NUMBER,
            ORDER_DETAILS.SHIP_DATE as SHIP_DATE,
            ORDER_DETAILS.PRODUCT_NUMBER as PRODUCT_NUMBER1,
```

Test | 25 Rows

   Here you can view the SQL that is generated to join these two query subjects together and can test the items returned by the join.

9. Click **Cancel**.

10. In the bottom right corner of the diagram, click **Diagram Overview** .

    Tip: The Diagram Overview icon only appears when there is more data, both vertically and horizontally, than the window can display.

11. Drag the red viewing window to view all parts of the diagram.

    This feature is useful when dealing with larger models.

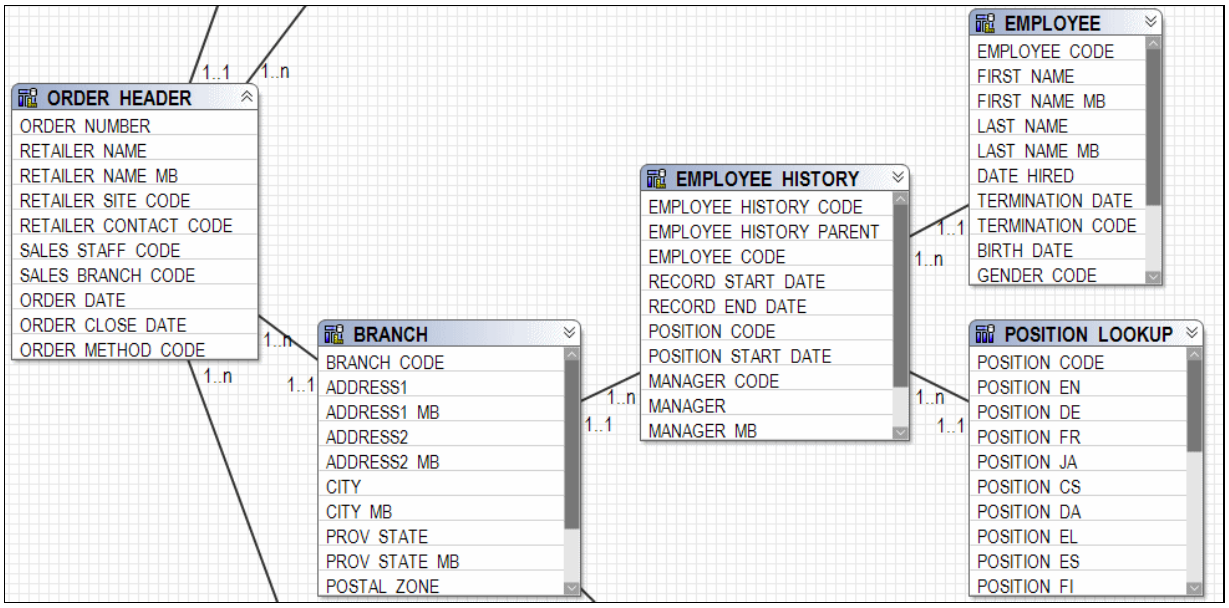
    You can see that all query subjects are linked to at least one other query subject, and that most relationships are one-to-many. You will examine one relationship in detail.

12. Close the **Diagram Navigation** window.

    It is recommended that you examine each relationship to verify that it is the relationship required for reporting purposes (with the appropriate cardinality) and make note of any missing relationships, where issues might occur, or items may need to be revisited.

Task 2. Identify missing relationships.

1. Examine the portion of the diagram below:



EMPLOYEE is joined to EMPLOYEE_HISTORY in a one-to-many relationship, allowing you to report on past positions. You will test this in the next task. BRANCH is also joined to EMPLOYEE_HISTORY in a one-to-many relationship. In this scenario EMPLOYEE_HISTORY is acting as a "bridge" table in a many-to-many relationship between BRANCH and EMPLOYEE. An employee may have worked in several branches and a branch may have several employees.

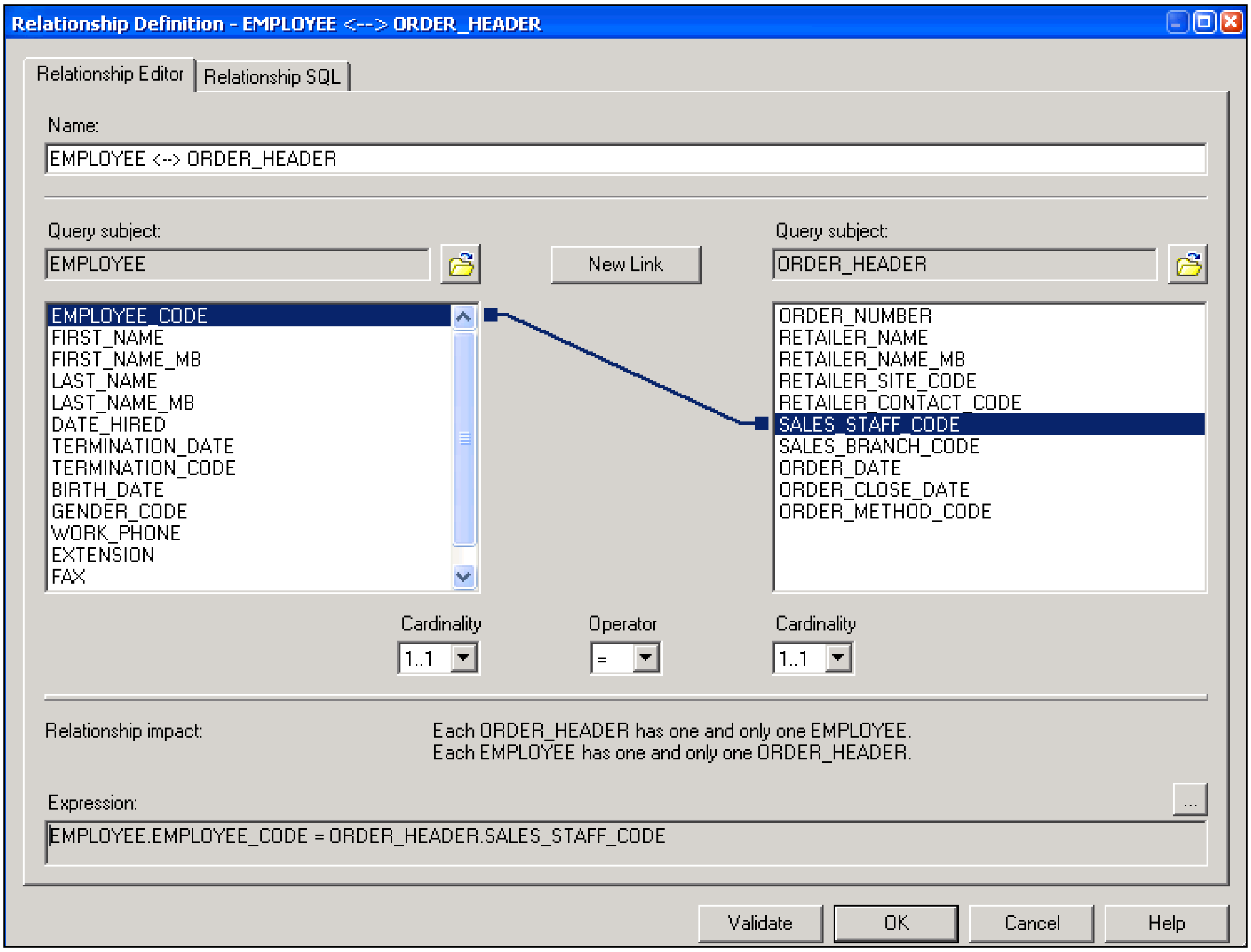
BRANCH has a relationship to ORDER_HEADER that indicates which branch made a sale. EMPLOYEE does not have a relationship to ORDER_HEADER. In this state, you cannot report on sales by staff, which is one of the requirements. EMPLOYEE needs a relationship to ORDER_HEADER.

In speaking with the database modeler, you discover that the relationship between the two objects should be EMPLOYEE.EMPLOYEE_CODE to ORDER_HEADER.SALES_STAFF_CODE. The names are different, but they do in fact represent the same information.

2. In the diagram, Ctrl+click **EMPLOYEE.EMPLOYEE_CODE** and **ORDER_HEADER.SALES_STAFF_CODE**.

3. Right-click one of the selected items, point to **Create**, and then click **Relationship**.

   The results appear as follows:



Relationship Definition - EMPLOYEE <--> ORDER_HEADER

Name: EMPLOYEE <--> ORDER_HEADER

Query subject: EMPLOYEE

EMPLOYEE_CODE
FIRST_NAME
FIRST_NAME_MB
LAST_NAME
LAST_NAME_MB
DATE_HIRED
TERMINATION_DATE
TERMINATION_CODE
BIRTH_DATE
GENDER_CODE
WORK_PHONE
EXTENSION
FAX

Query subject: ORDER_HEADER

ORDER_NUMBER
RETAILER_NAME
RETAILER_NAME_MB
RETAILER_SITE_CODE
RETAILER_CONTACT_CODE
SALES_STAFF_CODE
SALES_BRANCH_CODE
ORDER_DATE
ORDER_CLOSE_DATE
ORDER_METHOD_CODE

Cardinality: 1..1

Operator: =

Cardinality: 1..1

Relationship impact: Each ORDER_HEADER has one and only one EMPLOYEE.
Each EMPLOYEE has one and only one ORDER_HEADER.

Expression:

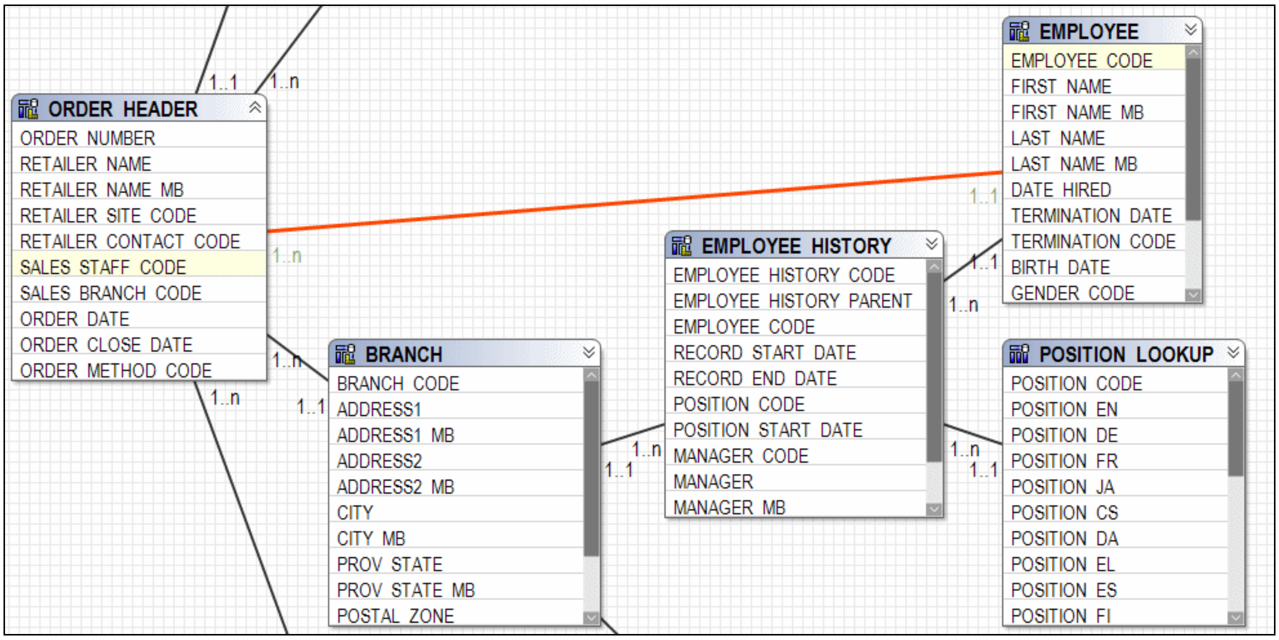
```
EMPLOYEE.EMPLOYEE_CODE = ORDER_HEADER.SALES_STAFF_CODE
```

   A relationship with the selected query items is created.

4. Under **ORDER_HEADER**, set the **Cardinality** to **1..n**.

5. Click **OK**.

   The results appear as follows:



   At this point, you are just ensuring that all required relationships are in place. Refining the model for presentation and predictable results will occur later in the modeling process.

Task 4. Test relationships.

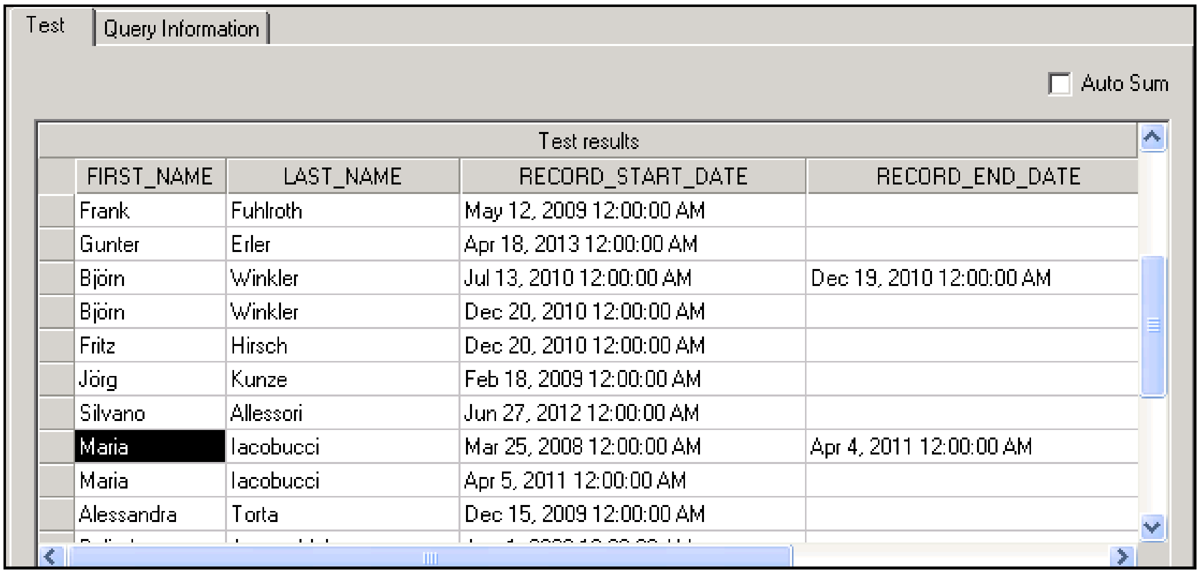
You will now test one relationship to help you understand the data as well as test the new relationship you just created.

1. In the diagram, select the following query items:

Query Subject	Query Item
EMPLOYEE	**FIRST_NAME** **LAST_NAME**
EMPLOYEE_HISTORY	**RECORD_START_DATE** **RECORD_END_DATE** **MANAGER** **BRANCH_CODE**
POSITION_LOOKUP	**POSITION_EN**

2. Right-click one of the selected items, click **Test**, and then click **Test Sample**.
3. Scroll down until you see the manager named **Maria Iacobucci**.

   The results appear as follows:



Test results

FIRST_NAME	LAST_NAME	RECORD_START_DATE	RECORD_END_DATE
Frank	Fuhlroth	May 12, 2009 12:00:00 AM	
Gunter	Erler	Apr 18, 2013 12:00:00 AM	
Björn	Winkler	Jul 13, 2010 12:00:00 AM	Dec 19, 2010 12:00:00 AM
Björn	Winkler	Dec 20, 2010 12:00:00 AM	
Fritz	Hirsch	Dec 20, 2010 12:00:00 AM	
Jörg	Kunze	Feb 18, 2009 12:00:00 AM	
Silvano	Allessori	Jun 27, 2012 12:00:00 AM	
Maria	Iacobucci	Mar 25, 2008 12:00:00 AM	Apr 4, 2011 12:00:00 AM
Maria	Iacobucci	Apr 5, 2011 12:00:00 AM	
Alessandra	Torta	Dec 15, 2009 12:00:00 AM	

   You can see that Maria changed managers (perhaps moving to a different part of the organization, but within the same branch).

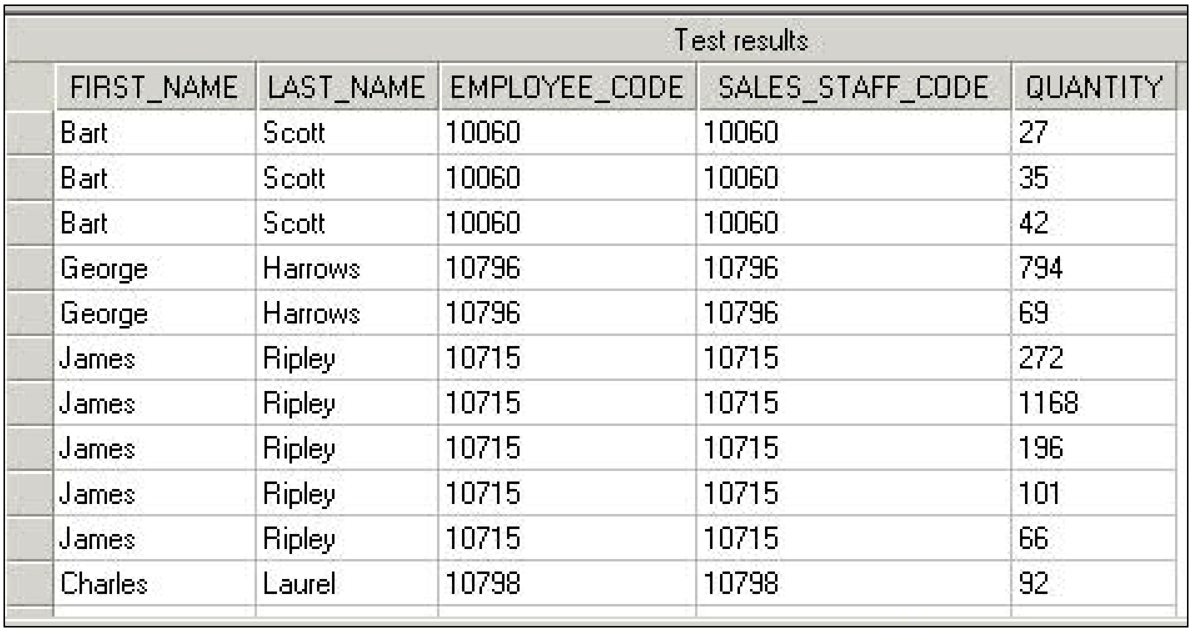
4. Click **Close**.

   You will now test the new relationship between EMPLOYEE and ORDER_HEADER.

5. In the diagram, test the following query items:

Query Subject	Query Item
EMPLOYEE	**FIRST_NAME** **LAST_NAME** **EMPLOYEE_CODE**
ORDER_HEADER	**SALES_STAFF_CODE**
ORDER_DETAILS	**QUANTITY**

The results are similar to the following:



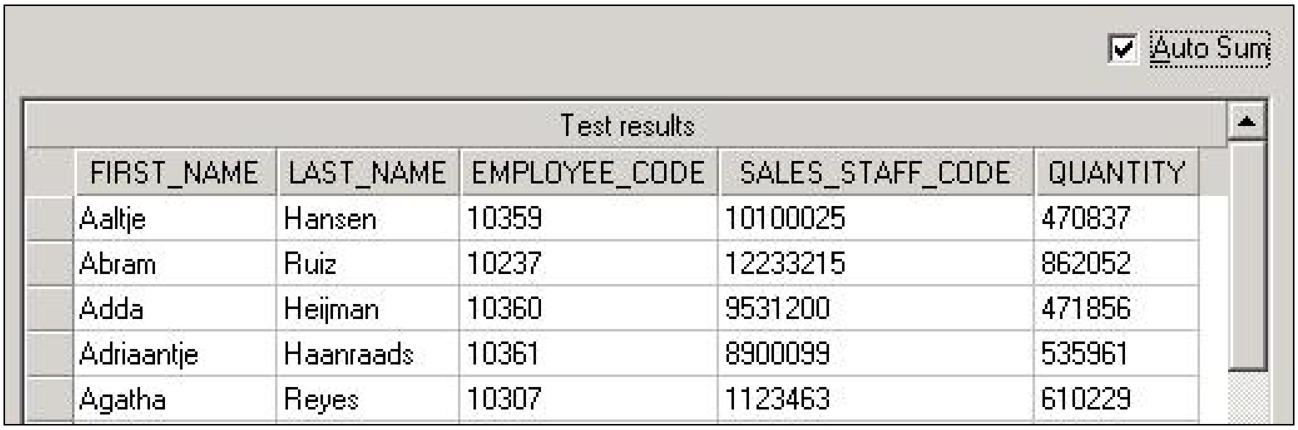
Test results

FIRST_NAME	LAST_NAME	EMPLOYEE_CODE	SALES_STAFF_CODE	QUANTITY
Bart	Scott	10060	10060	27
Bart	Scott	10060	10060	35
Bart	Scott	10060	10060	42
George	Harrows	10796	10796	794
George	Harrows	10796	10796	69
James	Ripley	10715	10715	272
James	Ripley	10715	10715	1168
James	Ripley	10715	10715	196
James	Ripley	10715	10715	101
James	Ripley	10715	10715	66
Charles	Laurel	10798	10798	92

You can see that the EMPLOYEE_CODE values and SALES_STAFF_CODE values do match. If you would like to see how many items an employee sold overall, you can test with Auto Sum enabled.

6. Select **Auto Sum**, and then click **Test Sample** again.

   The results appear as follows:



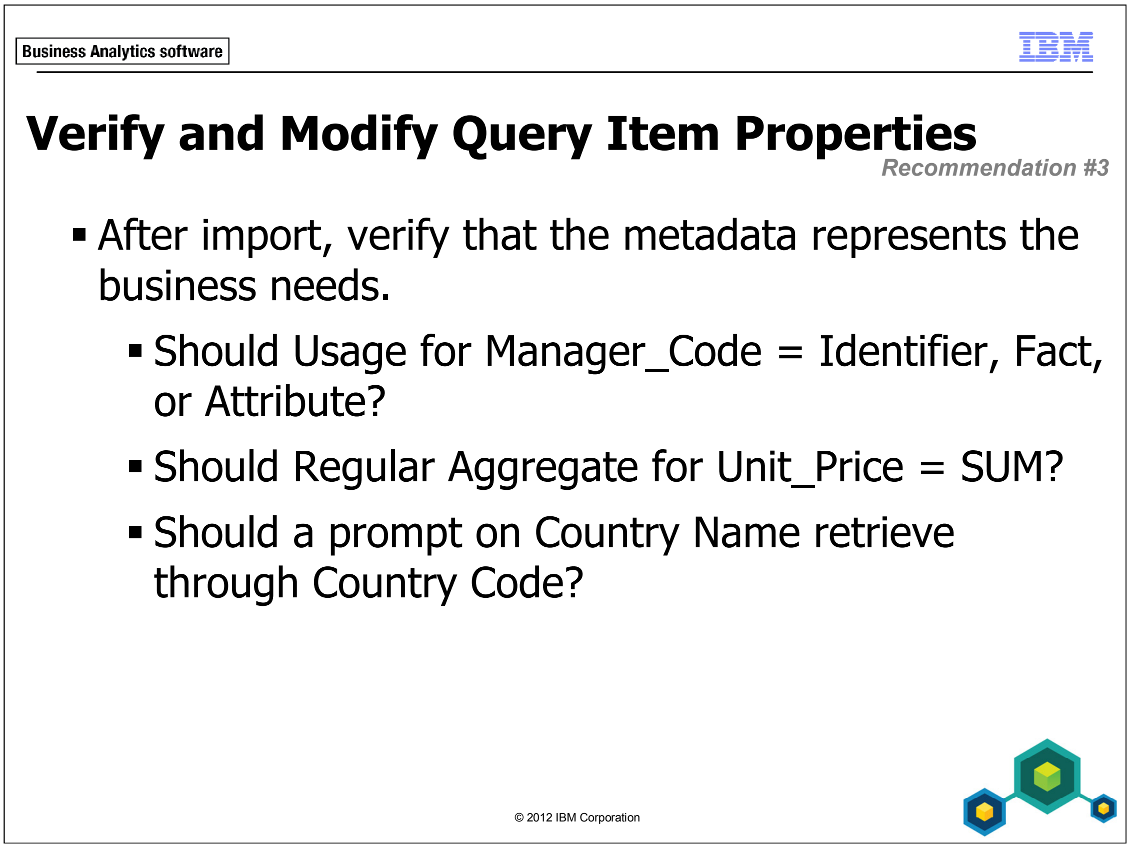
Test results

FIRST_NAME	LAST_NAME	EMPLOYEE_CODE	SALES_STAFF_CODE	QUANTITY
Aaltje	Hansen	10359	10100025	470837
Abram	Ruiz	10237	12233215	862052
Adda	Heijman	10360	9531200	471856
Adriaantje	Haanraads	10361	8900099	535961
Agatha	Reyes	10307	1123463	610229

   Quantity is now summed for each employee. The relationship is working as expected. However, notice the SALES_STAFF_CODE value. It seems to have been summed as well. This is because it was set as an integer during import. You will examine and fix this issue in the next demo.

7. Click **Close**, click **Save**, and keep **Framework Manager** open for the next demo.

Results:
To meet reporting requirements, you verified your relationships and added a missing relationship.



Business Analytics software

Verify and Modify Query Item Properties

Recommendation #3

- After import, verify that the metadata represents the business needs.
  - Should Usage for Manager_Code = Identifier, Fact, or Attribute?
  - Should Regular Aggregate for Unit_Price = SUM?
  - Should a prompt on Country Name retrieve through Country Code?

© 2012 IBM Corporation

Framework Manager populates object properties, such as Name, Usage, and Description, during import When you see a property value that seems questionable, you should ask the database administrator how the item is configured in the database.

Facts are numeric or time-interval, non-indexed columns. Identifiers are key, index, date, datetime, or any indexed columns. Attributes are typically strings.

The Regular Aggregate property for numeric facts (measures) defaults to SUM, which is correct for most measures (such as REVENUE and QUANTITY). However a report identifying four tents sold to a customer would probably not display the sum of their prices, but instead the average.

Take advantage of Prompt properties to improve performance by causing automatic retrieval through indexes.

Demo 2: Verify and Modify Query Item Properties

Purpose:
To meet reporting requirements, you will verify that important query item properties, such as Usage and Regular Aggregate, are correct. You will also ensure efficient queries using filters by configuring query items to use an indexed key value rather than a non-indexed string value in the filter.

Components:	**Framework Manager, Query Studio, Report Studio**
Project:	**GO Operational**
Package:	**GO Operational**

Task 1. Verify Usage property settings.

1. If necessary, from the **View** menu, click **Properties** to view the **Properties** pane.
2. In the **Project Viewer**, expand **Foundation Object View** > **gosales** > **ORDER_HEADER**, and then click **ORDER_NUMBER**.

   Both the query item icon and the Usage property field tell you that ORDER_NUMBER is an identifier, and were identified as such during import since this is the primary key in the database table.
3. Click **SALES_STAFF_CODE**.

   In comparison, SALES_STAFF_CODE has a measure icon (also known as a Fact) and its Usage property is set to Fact. This was specified during the import because it is a non-indexed numeric field in the database. You know that this query item is not a measure. It is used for associating several employees to an order. You will set this query item's Usage property to Attribute. If you know that any item would be used in relationships or that report authors would filter on an item regularly, you would ask the database administrator to index the field in the database. At that point you would set the Usage property to Identifier. For now, you will set SALES_STAFF_CODE as Attribute to indicate to yourself and other modelers that this integer field is not indexed in the database.

4. In the **Properties** pane, click the **Usage** property, and then select **Attribute** from the list.

   You will now change multiple properties at once to correct other items that Framework Manager incorrectly set as facts.

5. Under **ORDER_HEADER**, Ctlt+click to select the following items:
   - **RETAILER_SITE_CODE**
   - **RETAILER_CONTACT_CODE**
   - **SALES_BRANCH_CODE**

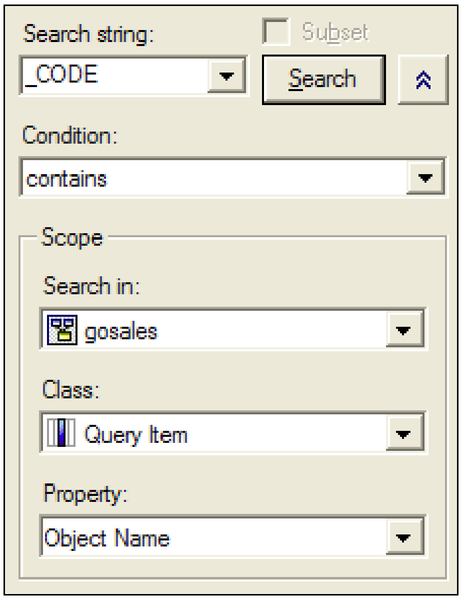
6. In the **Properties** pane, in the **RETAILER_SITE_CODE** row, scroll to the right, click **Fact** below the **Usage** heading, and then select **Attribute** from the list.

7. Drag the small black arrow beneath the value downward to change the values to **Attribute** for the other two query items.

Task 2. Use Bulk Replace to change property settings.

Rather than go through each query subject and manually change the Usage setting for codes from Fact to Attribute, you can use the Search tool to perform bulk changes.

1. If necessary, from the **View** menu, click **Tools**.
2. In the **Tools** pane, click the **Search** tab, and then in the **Search string** box, type **_CODE**.
3. Beside the **Search** button, click ☐ to see additional search options.
4. Select the following configuration:

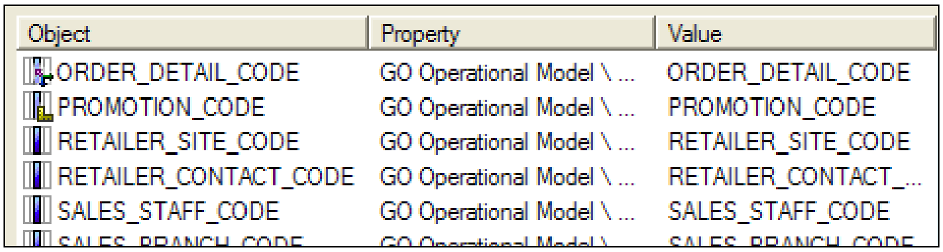


Search string: _CODE
Subset
Search
Condition: contains
Scope
Search in: gosales
Class: Query Item
Property: Object Name

This will confine your search to the gosales namespace, only look at query items, and only look at name properties containing the string _CODE.

5. Click **Search**, and then, if necessary, resize the Tools pane to view the results.

   The results appear as follows:

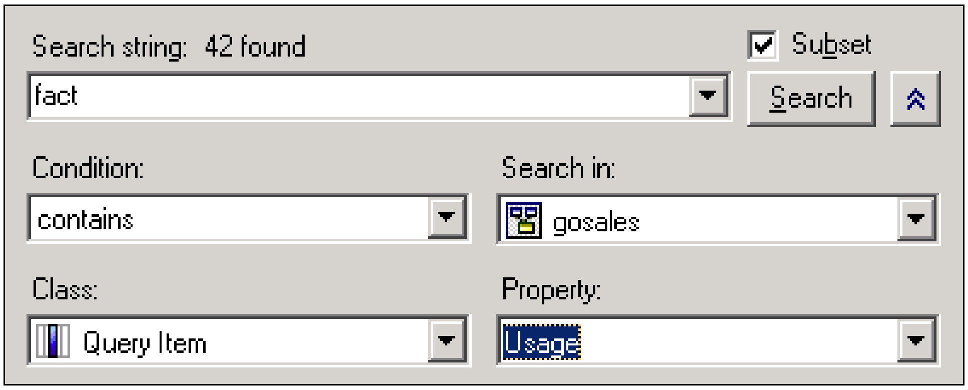


Object	Property	Value
ORDER_DETAIL_CODE	GO Operational Model \ ...	ORDER_DETAIL_CODE
PROMOTION_CODE	GO Operational Model \ ...	PROMOTION_CODE
RETAILER_SITE_CODE	GO Operational Model \ ...	RETAILER_SITE_CODE
RETAILER_CONTACT_CODE	GO Operational Model \ ...	RETAILER_CONTACT_...
SALES_STAFF_CODE	GO Operational Model \ ...	SALES_STAFF_CODE
SALES_BRANCH_CODE	GO Operational Model \	SALES_BRANCH_CODE

   You will search a subset of this result set to display only items whose Usage property is set to Fact.

6. Select the **Subset** check box, in the **Search string** box, type **fact**, and then change the **Property** selection to **Usage**.

   The settings appear as follows:



Search string: 42 found

fact

☑ Subset

Condition: contains

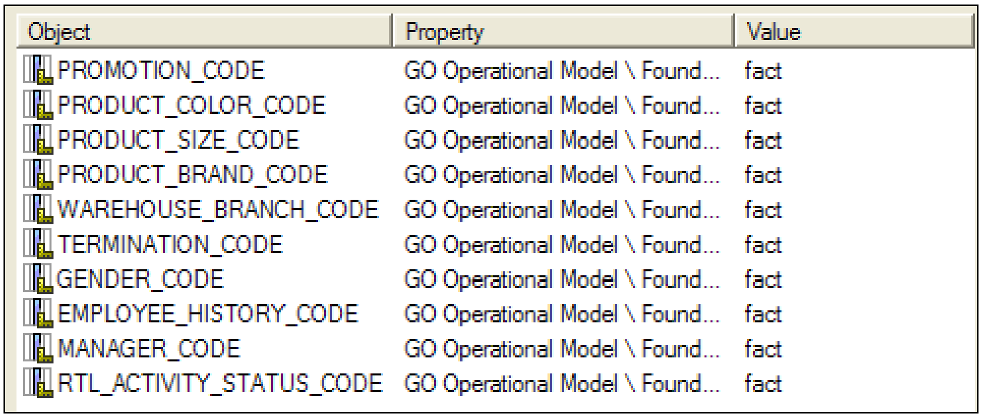
Search in: gosales

Class: Query Item

Property: Usage

7. Click **Search**.

   The results appear as follows:

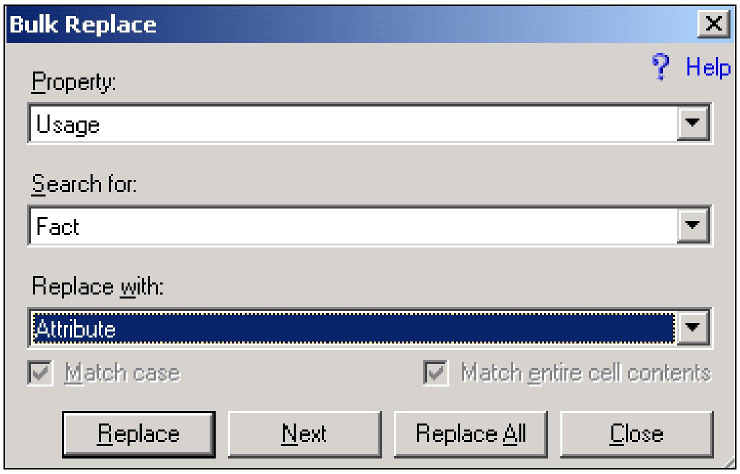


Object	Property	Value
PROMOTION_CODE	GO Operational Model \ Found...	fact
PRODUCT_COLOR_CODE	GO Operational Model \ Found...	fact
PRODUCT_SIZE_CODE	GO Operational Model \ Found...	fact
PRODUCT_BRAND_CODE	GO Operational Model \ Found...	fact
WAREHOUSE_BRANCH_CODE	GO Operational Model \ Found...	fact
TERMINATION_CODE	GO Operational Model \ Found...	fact
GENDER_CODE	GO Operational Model \ Found...	fact
EMPLOYEE_HISTORY_CODE	GO Operational Model \ Found...	fact
MANAGER_CODE	GO Operational Model \ Found...	fact
RTL_ACTIVITY_STATUS_CODE	GO Operational Model \ Found...	fact

   You will now use the Bulk Replace feature to change the Usage property for these items to Attribute.

8. In the bottom right corner of the **Tools** pane, click **Bulk Replace**.

   Modify the settings as shown here:



Bulk Replace

Property: Usage

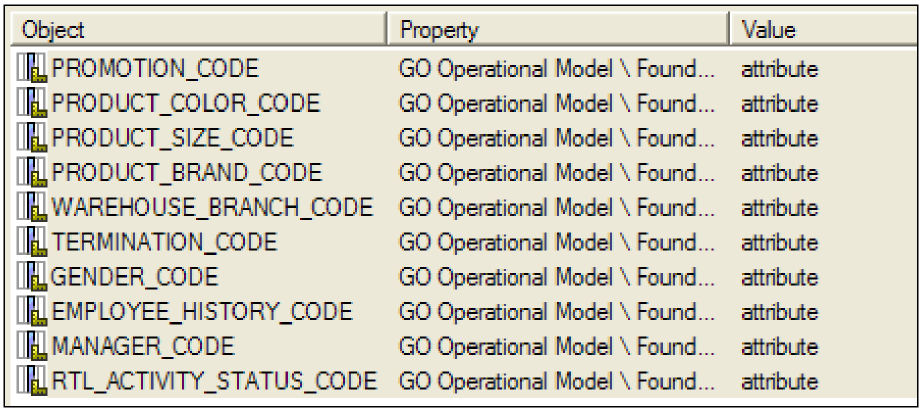
Search for: Fact

Replace with: Attribute

☑ Match case ☑ Match entire cell contents

9. Click **Replace All**, and then click **Yes**.

   The results appear as follows:



Object	Property	Value
PROMOTION_CODE	GO Operational Model \ Found...	attribute
PRODUCT_COLOR_CODE	GO Operational Model \ Found...	attribute
PRODUCT_SIZE_CODE	GO Operational Model \ Found...	attribute
PRODUCT_BRAND_CODE	GO Operational Model \ Found...	attribute
WAREHOUSE_BRANCH_CODE	GO Operational Model \ Found...	attribute
TERMINATION_CODE	GO Operational Model \ Found...	attribute
GENDER_CODE	GO Operational Model \ Found...	attribute
EMPLOYEE_HISTORY_CODE	GO Operational Model \ Found...	attribute
MANAGER_CODE	GO Operational Model \ Found...	attribute
RTL_ACTIVITY_STATUS_CODE	GO Operational Model \ Found...	attribute

   All Usage properties are now changed to attribute.

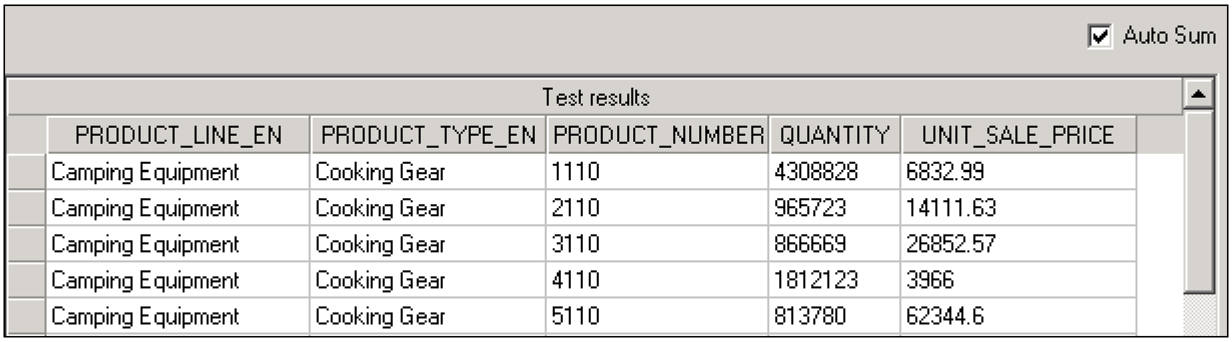
   Using the Search tool can save a lot of time when performing these types of tasks. However, you should still examine all query items to ensure the settings are correct. For example, PRODUCT_BASE_NUMBER from PRODUCT and EMPLOYEE_HISTORY_PARENT from EMPLOYEE_HISTORY are still identified as facts. The search did not catch these since they did not contain the string _CODE.

Task 3. Verify Regular Aggregate property settings.

1. Select and test the following query items with **Auto Sum** enabled:

Query Subject	Query Item
PRODUCT_LINE	**PRODUCT_LINE_EN**
PRODUCT_TYPE	**PRODUCT_TYPE_EN**
PRODUCT	**PRODUCT_NUMBER**
ORDER_DETAILS	**QUANTITY** **UNIT_SALE_PRICE**

The results appear as follows:



☑ Auto Sum

Test results

PRODUCT_LINE_EN	PRODUCT_TYPE_EN	PRODUCT_NUMBER	QUANTITY	UNIT_SALE_PRICE
Camping Equipment	Cooking Gear	1110	4308828	6832.99
Camping Equipment	Cooking Gear	2110	965723	14111.63
Camping Equipment	Cooking Gear	3110	866669	26852.57
Camping Equipment	Cooking Gear	4110	1812123	3966
Camping Equipment	Cooking Gear	5110	813780	62344.6

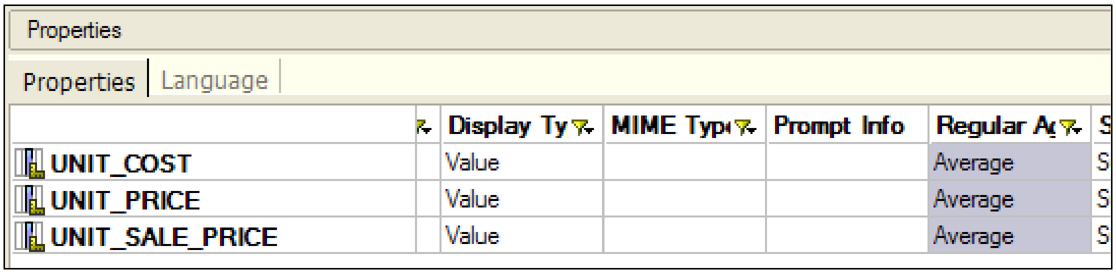
Notice UNIT_SALE_PRICE is summed. This does not meet the reporting requirements. In a report on ORDER_DETAILS rolled up for each product you would expect the QUANTITY to be summed, but it would make no sense to sum up the UNIT_SALE_PRICE (the sale price for one unit of each product in the order), since you may have sold several units in one order detail. You will change this aggregation to default to something more useful. In this case, Average. The same situation applies to UNIT_COST and UNIT_PRICE. (Later in the course, you will create a Revenue calculation, QUANTITY * UNIT_SALE_PRICE.)

2. Click **Close**, and then in the **Project Viewer**, under **ORDER_DETAILS**, select the following items:

   UNIT_COST
   UNIT_PRICE
   UNIT_SALE_PRICE

3. Using the method shown in Task 1, change the **Regular Aggregate** property value to **Average** for all the items.

   The results appear as follows:



Properties

Properties | Language

	Display Ty	MIME Typ	Prompt Info	Regular A	S
UNIT_COST	Value			Average	S
UNIT_PRICE	Value			Average	S
UNIT_SALE_PRICE	Value			Average	S

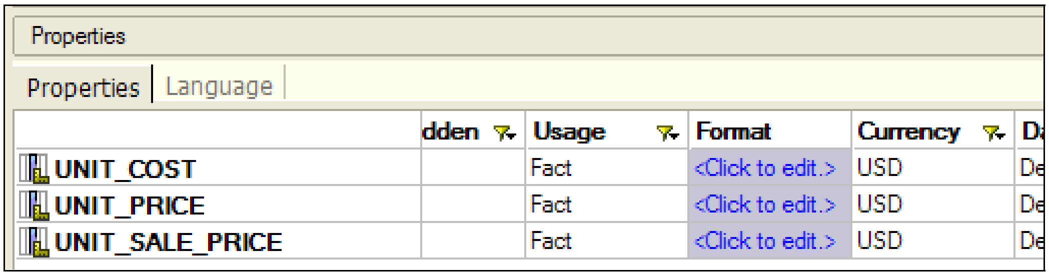
   You will change the format for these objects to $USD Currency.

4. In the **UNIT_COST** row, under the **Format** heading, click on **<Click to edit.>**.

5. Under **Format type**, select **Currency**, and then in the **Properties** pane, set **Currency** to **$(USD) - United States of America, dollar**.

6. Click **OK**, drag the small **black arrow** down to change the property setting for the other two items, and then click **OK**.

   The results appear as follows:



Properties

Properties | Language

	dden	Usage	Format	Currency	D
UNIT_COST		Fact	<Click to edit.>	USD	De
UNIT_PRICE		Fact	<Click to edit.>	USD	De
UNIT_SALE_PRICE		Fact	<Click to edit.>	USD	De

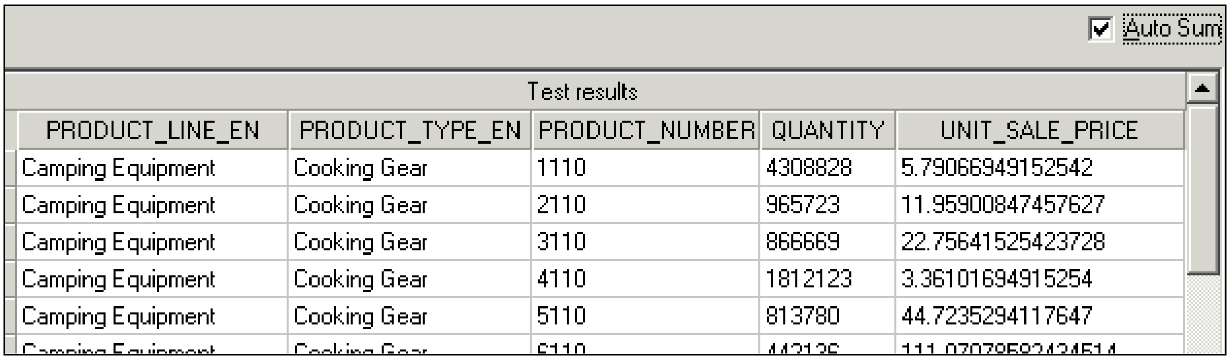
7. Save the project.

Task 4. Test property changes in Framework Manager.

1. Select and test the following query items with **Auto Sum** enabled:

Query Subject	Query Item
PRODUCT_LINE	**PRODUCT_LINE_EN**
PRODUCT_TYPE	**PRODUCT_TYPE_EN**
PRODUCT	**PRODUCT_NUMBER**
ORDER_DETAILS	**QUANTITY** **UNIT_SALE_PRICE**

The results appear as follows:



PRODUCT_LINE_EN	PRODUCT_TYPE_EN	PRODUCT_NUMBER	QUANTITY	UNIT_SALE_PRICE
Camping Equipment	Cooking Gear	1110	4308828	5.79066949152542
Camping Equipment	Cooking Gear	2110	965723	11.95900847457627
Camping Equipment	Cooking Gear	3110	866669	22.75641525423728
Camping Equipment	Cooking Gear	4110	1812123	3.36101694915254
Camping Equipment	Cooking Gear	5110	813780	44.7235294117647

QUANTITY has been summed while UNIT_SALE_PRICE has been averaged. This is based on the property setting you specified earlier.

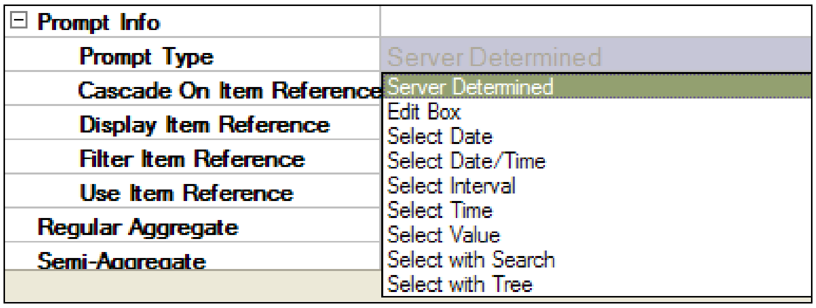
2. Click **Close**.

Task 5. Adjust Prompt properties.

Prompts can be either manually created by the report author, or generated by IBM Cognos 10 when the report author filters an item, or includes a prompt value (i.e. ?Product Line?) in an expression. The prompt type for each query item is defined in Framework Manager.

1. In the **Project Viewer**, under **PRODUCT_LINE** select **PRODUCT_LINE_EN**.
2. In the **Properties** pane, scroll down and expand **Prompt Info**.
3. View the options in the **Prompt Type** list.

   The results appear as follows:



⊟ Prompt Info	
Prompt Type	Server Determined
Cascade On Item Reference	Server Determined
Display Item Reference	Edit Box
Filter Item Reference	Select Date
Use Item Reference	Select Date/Time
Regular Aggregate	Select Interval
Semi-Aggregate	Select Time
	Select Value
	Select with Search
	Select with Tree

   You can specify the type of prompt that you want generated in Query Studio or the default prompt type for Report Studio. The default is for the server to determine the prompt type based on data type.

   Here is a brief description of the other properties:

   Cascade on Item Reference is for cascading prompts in Report Studio, for example, where the list of Product Type choices is restricted to those within a selected Product Line.

Display Item Reference identifies the default value that a manually created Report Studio prompt will display for a particular query item. For example, to see a list of PRODUCT_LINE_EN names when using PRODUCT_LINE_CODE in a prompt, set the Display Item Reference property for PRODUCT_LINE_CODE to PRODUCT_LINE_EN. If this field is left blank, it will default to the values returned by query item to which it belongs.

Use Item Reference identifies the default value that a manually created Report Studio prompt will use in the query's filter. For example, for a Report Studio generated prompt on PRODUCT_LINE_EN to display a list of PRODUCT_LINE_EN names but retrieve data through the PRODUCT_LINE_CODE, set the Use Item Reference property in PRODUCT_LINE_EN to PRODUCT_LINE_CODE.

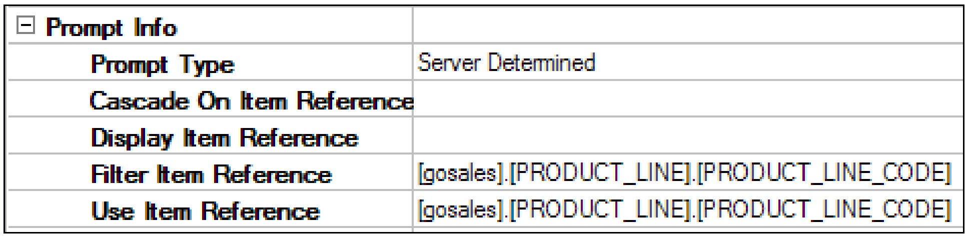
Filter Item Reference identifies the value that an IBM Cognos 10 generated prompt will use to filter your query. For example, for Query Studio to display PRODUCT_LINE_EN values but use PRODUCT_LINE_CODE in the query's filter, set the Filter Item Reference property to PRODUCT_LINE_CODE.

You will set prompt properties on PRODUCT_LINE_EN and PRODUCT_TYPE_EN so that they use appropriate indexes in any prompt situation to retrieve values from the database. To achieve this, you will set the Use Item Reference property to ensure user generated prompts in Report Studio are efficient, and you will set the Filter Item Reference property to ensure any generic IBM Cognos generated prompt is efficient. You will also configure PRODUCT_TYPE_EN to cascade off of PRODUCT_LINE_EN.

4. In the cell to the right of **Use Item Reference**, click the **ellipsis** (**…**), expand **Foundation Objects View**>**gosales**> **PRODUCT_LINE**, click **PRODUCT_LINE_CODE**, and then click **OK**.

5. In the cell to the right of **Filter Item Reference**, click the **ellipsis** (…), expand **Foundation Objects View>gosales>PRODUCT_LINE**, click **PRODUCT_LINE_CODE**, and then click **OK**.

   The results appear as follows:



⊟ Prompt Info	
Prompt Type	Server Determined
Cascade On Item Reference	
Display Item Reference	
Filter Item Reference	[gosales].[PRODUCT_LINE].[PRODUCT_LINE_CODE]
Use Item Reference	[gosales].[PRODUCT_LINE].[PRODUCT_LINE_CODE]

6. Save the project.

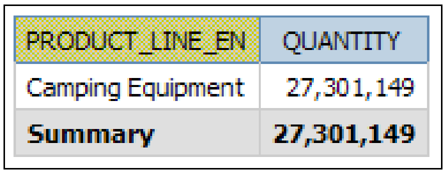
Task 6. Test the prompt property change in Query Studio and Report Studio.

1. Publish the **GO Operational** package again, overwriting the earlier version.
2. Launch **IBM Cognos Connection**, log on (if not already open), and then launch **Query Studio** selecting the **GO Operational** package.
3. Add the following items to the report:

Query Subject	Query Item
PRODUCT_LINE	**PRODUCT_LINE_EN**
ORDER_DETAILS	**QUANTITY**

4. Select the **PRODUCT_LINE_EN** column, and then click **Filter** 🝖.
5. Select **Camping Equipment** and then click **OK**.

   The query retrieves the expected row.



PRODUCT_LINE_EN	QUANTITY
Camping Equipment	27,301,149
Summary	**27,301,149**

   To view the underlying SQL, you will convert the Query Studio report to a Report Studio report.

6. Under **Menu**, click **Manage File**, and then click **Open in Report Studio**.
7. In **Report Studio**, from the **Tools** menu, click **Show generated SQL/MDX**, and then, in the **Generated SQL/MDX** list, select **IBM Cognos SQL**.

   Note that the Where clause uses (PRODUCT_LINE.PRODUCT_LINE_CODE in ('991')). This shows that you are retrieving data based on indexed codes rather than the less-efficient and non-indexed PRODUCT_LINE_EN strings.
8. Click **Close**, and then close **Report Studio** and **Query Studio** without saving the reports.

Results:
To meet reporting requirements, you verified that important query item properties such as Usage and Regular Aggregate were correct. Also, to make filtering more efficient, you configured PRODUCT_LINE _EN to use PRODUCT_LINE _CODE when filtering on PRODUCT_LINE _EN.

Workshop 1: Verify and Modify Query Item Properties

After every import of metadata, you need to verify that query item properties meet the needs of the business analysts creating reports for the Sample Outdoors. You should:

- Verify query item properties. Change the Usage property from Fact to Attribute where numeric query items are neither facts nor indexes.
- In the PRODUCT query subject, ensure that PRODUCTION_COST and GROSS_MARGIN are averaged when their values are summarized.
- In the COUNTRY query subject, change COUNTRY_EN so that when a prompt is manually created or automatically generated it will cause retrieval through the COUNTRY_CODE key.
- Publish the package and verify your prompt works using Query Studio and Report Studio.
- In Framework Manager, save the project, close Framework Manager
- Close all browser windows without saving the report

For more detailed information outlined as tasks, see the Task Table on the next page.

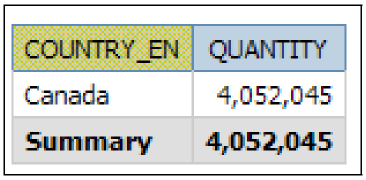
Workshop 1: Task Table

Task	Where to Work	Hints
1. Change query item Usage properties.	Project Viewer and Property panes	• Ctrl+click all desired query items to make the same change to multiple items at once. • Change Usage to Attribute for numeric query items that are not facts for the following items: • **PRODUCT> BASE_PRODUCT_NUMBER** • **EMPLOYEE_HISTORY> EMPLOYEE_HISTORY_ PARENT**
2. Change Regular Aggregate properties.	Project Viewer and Property panes	• Ctrl+click all desired query items to make the same change to multiple items at once. • Set the Regular Aggregate property to Average for the following items: • **PRODUCT>PRODUCTION_ COST** • **PRODUCT>GROSS_MARGIN**
3. Change prompt properties.	Project Viewer and Property panes	• Under Prompt Info, change the Filter Item Reference and Use Item Reference properties for **COUNTRY>COUNTRY_EN** to use **COUNTRY>COUNTRY_CODE**

4. Publish the package and verify your prompt works using Query and Report Studio.	Query Studio, Report Studio	• Publish GO Operational package. • In Query Studio, create a query using the following items: • **COUNTRY>COUNTRY_EN** • **ORDER_DETAILS> QUANTITY** • Filter COUNTRY_EN on Canada. • Open the report in Report Studio and use Tools>Show Generated SQL/MDX to see if retrieval is through COUNTRY_CODE. • In Framework Manager, save the project, close Framework Manager • Close all browser windows without saving the report

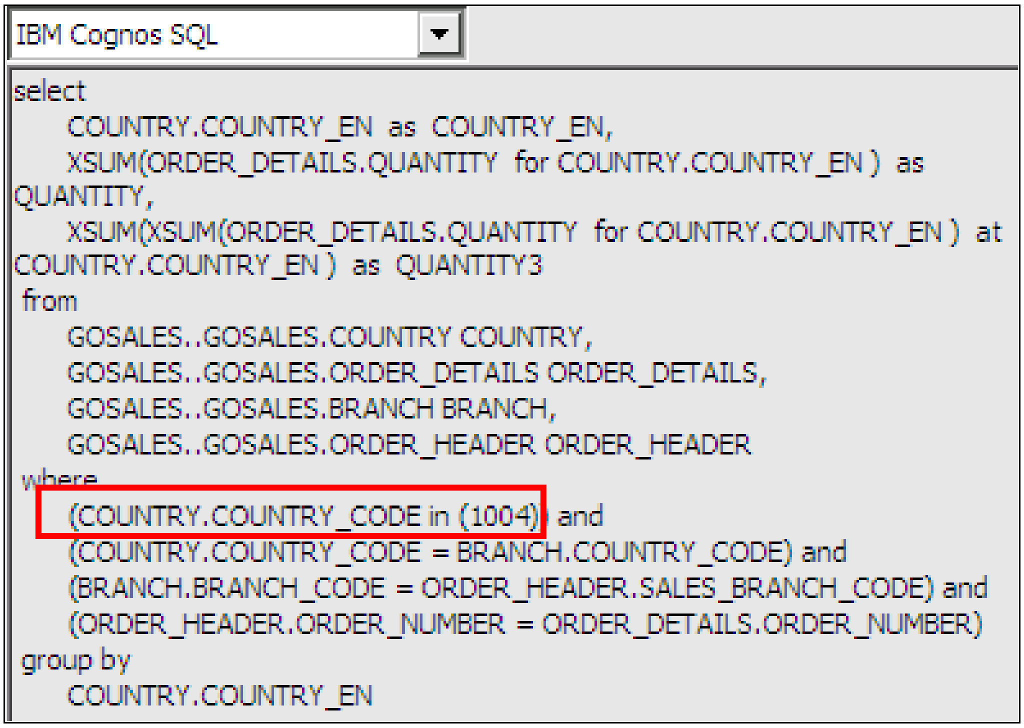
Workshop 1: Results

After filtering the report in Query Studio, the results appear as follows:



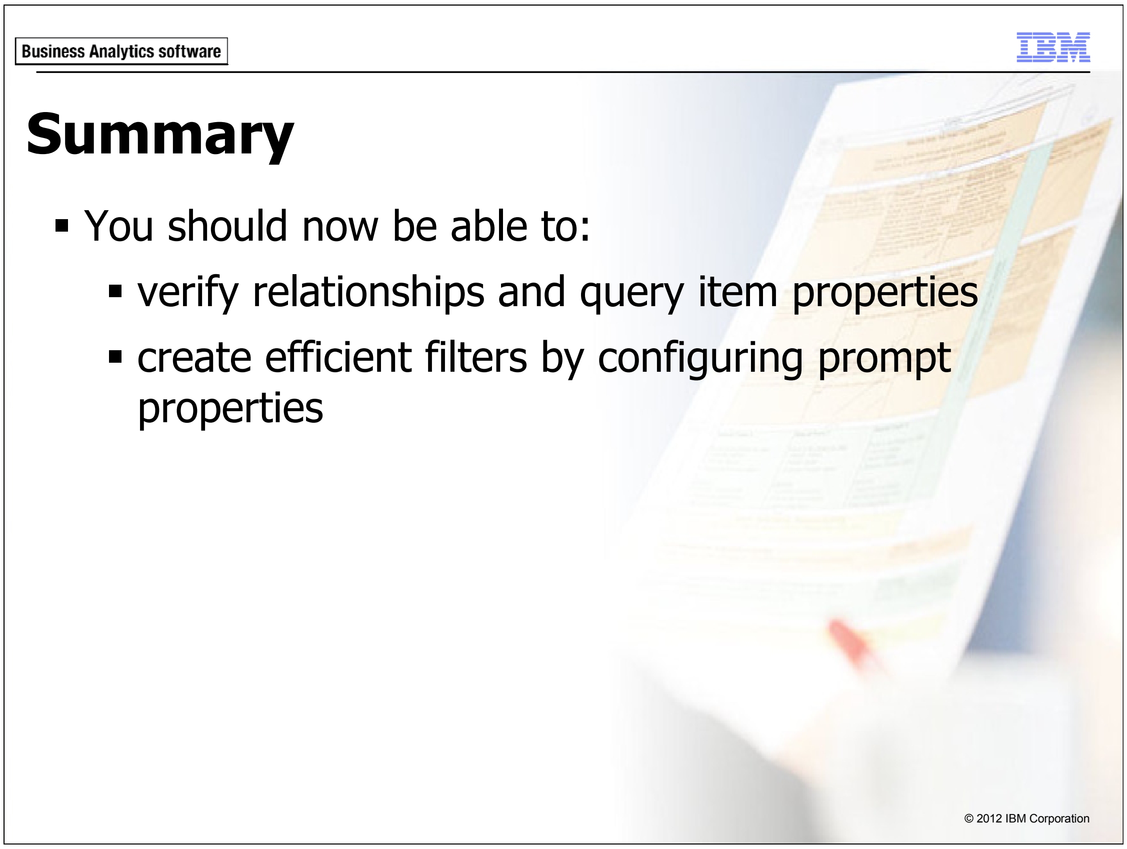
COUNTRY_EN	QUANTITY
Canada	4,052,045
Summary	**4,052,045**

After viewing the IBM Cognos SQL in Report Studio, the results appear as follows:



```
select
    COUNTRY.COUNTRY_EN  as  COUNTRY_EN,
    XSUM(ORDER_DETAILS.QUANTITY  for COUNTRY.COUNTRY_EN )  as
QUANTITY,
    XSUM(XSUM(ORDER_DETAILS.QUANTITY  for COUNTRY.COUNTRY_EN )  at
COUNTRY.COUNTRY_EN )  as  QUANTITY3
 from
    GOSALES..GOSALES.COUNTRY COUNTRY,
    GOSALES..GOSALES.ORDER_DETAILS ORDER_DETAILS,
    GOSALES..GOSALES.BRANCH BRANCH,
    GOSALES..GOSALES.ORDER_HEADER ORDER_HEADER
 where
    (COUNTRY.COUNTRY_CODE in (1004)) and
    (COUNTRY.COUNTRY_CODE = BRANCH.COUNTRY_CODE) and
    (BRANCH.BRANCH_CODE = ORDER_HEADER.SALES_BRANCH_CODE) and
    (ORDER_HEADER.ORDER_NUMBER = ORDER_DETAILS.ORDER_NUMBER)
 group by
    COUNTRY.COUNTRY_EN
```

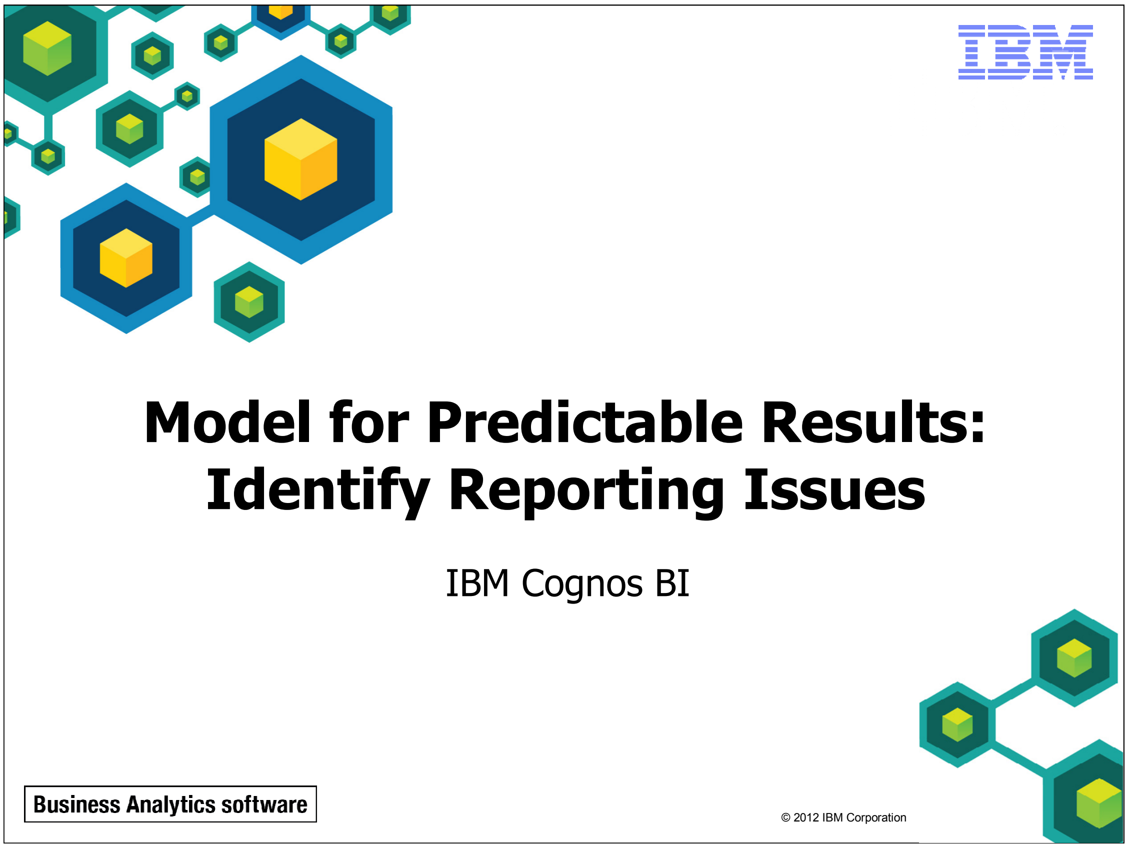
The IBM Cognos SQL generated in Report Studio for the Query Studio query shows that you are accessing the data through COUNTRY_CODE.



Business Analytics software

Summary

- You should now be able to:
  - verify relationships and query item properties
  - create efficient filters by configuring prompt properties



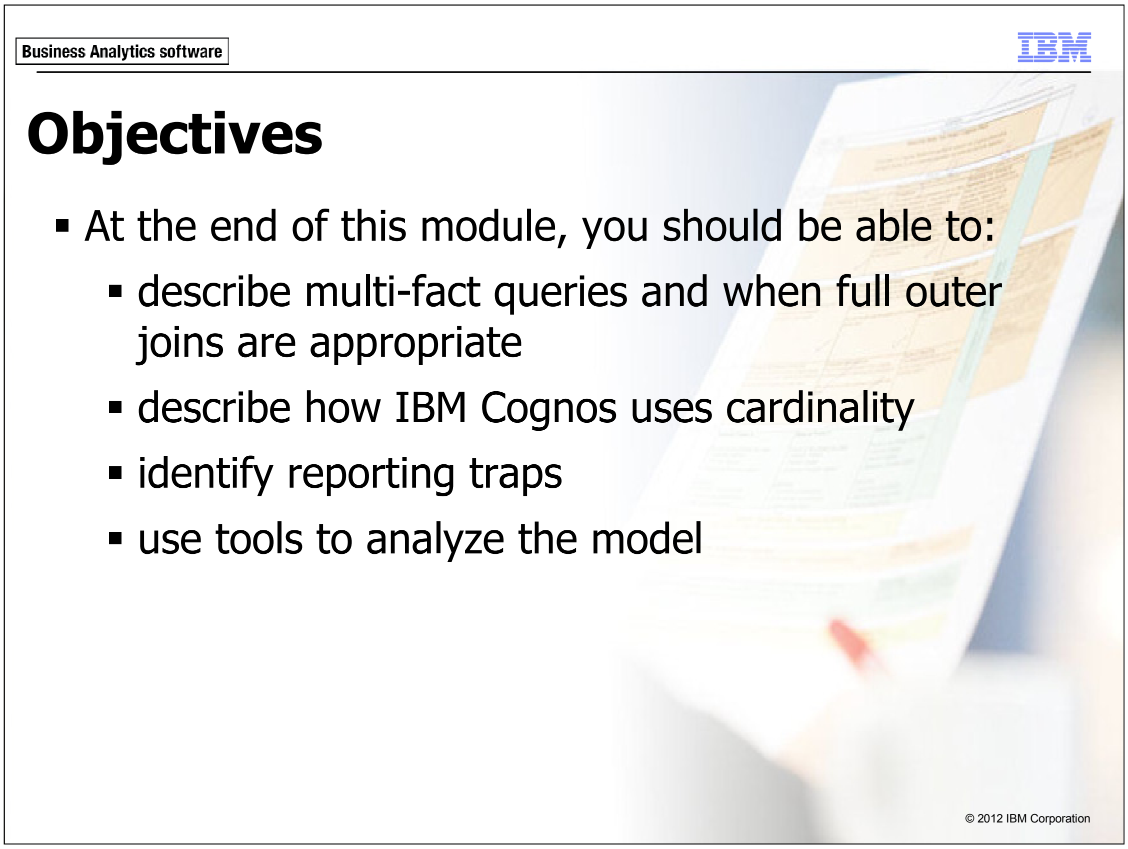
Model for Predictable Results: Identify Reporting Issues

IBM Cognos BI

Business Analytics software

Unless otherwise specified in demo or workshop steps, you will always log on to IBM Cognos BI in the Local LDAP namespace using the following credentials:

- User ID: admin
- Password: Education1



Business Analytics software

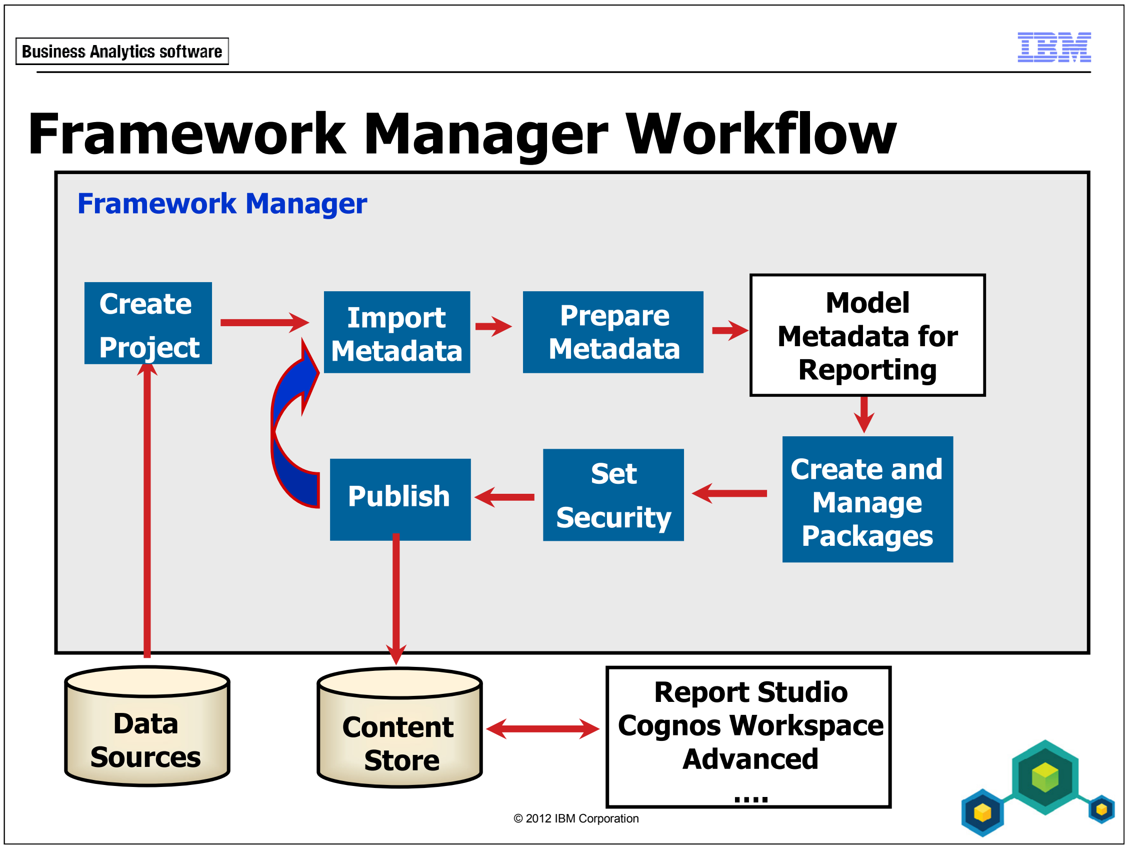
Objectives

- At the end of this module, you should be able to:
  - describe multi-fact queries and when full outer joins are appropriate
  - describe how IBM Cognos uses cardinality
  - identify reporting traps
  - use tools to analyze the model

© 2012 IBM Corporation

The concepts and techniques covered in this module are fundamental with respect to creating models for predictable results. Although some of the concepts appear advanced or complex, they are truly fundamental to successful models.

Before reviewing this module, you should be familiar with IBM Cognos, IBM Cognos Connection, Query Studio and Report Studio. Suggested modules to reference:
• Overview of IBM Cognos
• Identify Common Data Structures
• Gather Requirements
• Create a Baseline Project
• Prepare Reusable Metadata

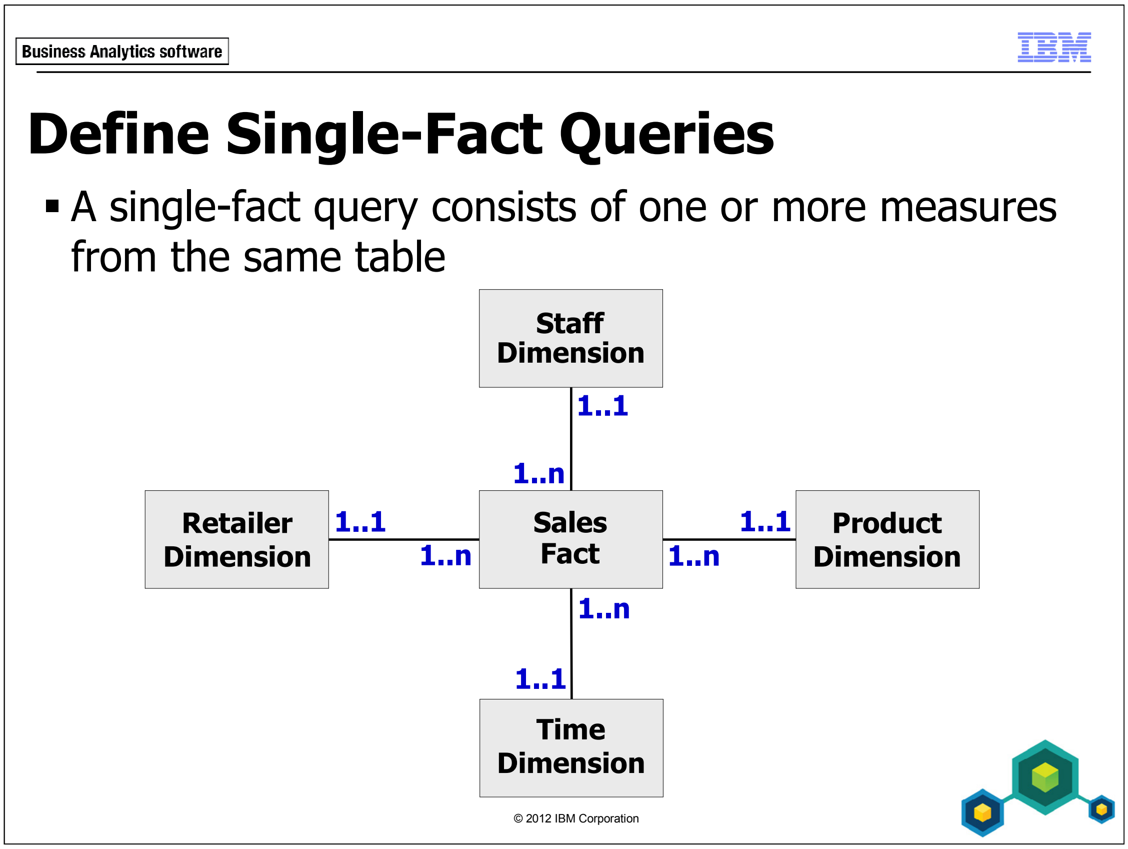

Business Analytics software

IBM

Framework Manager Workflow

© 2012 IBM Corporation

This module deals with identifying potential reporting issues before beginning to model the metadata. Solutions to the issues found in this module will be applied in later modules.



Business Analytics software

Define Single-Fact Queries

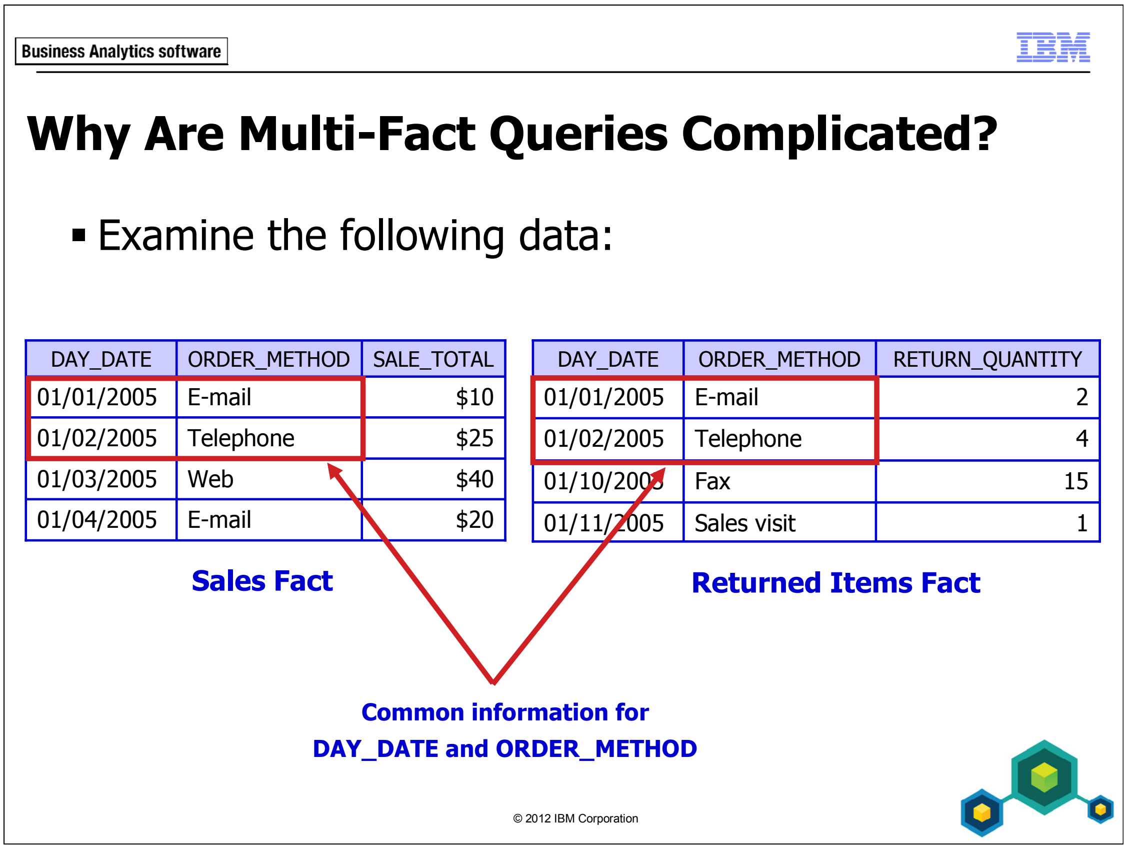
- A single-fact query consists of one or more measures from the same table

© 2012 IBM Corporation

In the above example, if you only queried items from the dimensions, such as Product and Retailer, you would go through the fact table. Performance may be an issue if the fact table is very large. This is something to be aware of when viewing the generated SQL and considering performance for those types of queries.

If we use optional cardinalities, the full outer joins ensure that all data is returned, not just where matches occur. For example, if we wanted to see all sales, regardless of whether a sales rep made the sale, and see all sales reps, regardless of whether they made sales or not, we would set optional cardinality on both Staff Dimension and Sales Fact. This would generate a full outer join, ensuring that all rows are returned from each table. These types of scenarios should be configured with caution as they have the potential to be resource intensive. The measures can be constricted by dimensional attributes taken from related dimension query subjects at the 1..1 end of 1..1 to 1..n relationships.

In most cases, single-fact queries are not a problem for reporting and usually generate simple and expected SQL.



Business Analytics software

Why Are Multi-Fact Queries Complicated?

- Examine the following data:

DAY_DATE	ORDER_METHOD	SALE_TOTAL
01/01/2005	E-mail	$10
01/02/2005	Telephone	$25
01/03/2005	Web	$40
01/04/2005	E-mail	$20

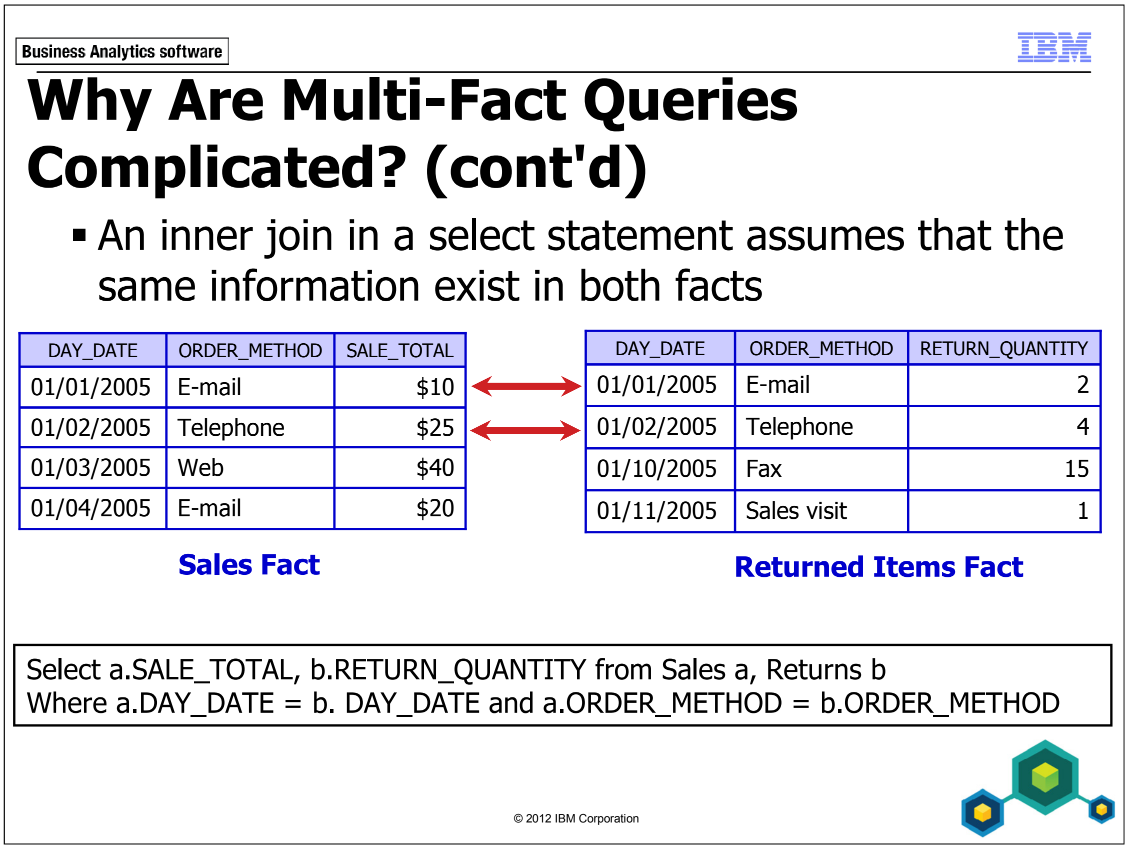
Sales Fact

DAY_DATE	ORDER_METHOD	RETURN_QUANTITY
01/01/2005	E-mail	2
01/02/2005	Telephone	4
01/10/2005	Fax	15
01/11/2005	Sales visit	1

Returned Items Fact

Common information for DAY_DATE and ORDER_METHOD

While the first two rows have identical DAY_DATE and ORDER_METHOD information, rows three and four do not.



Business Analytics software

Why Are Multi-Fact Queries Complicated? (cont'd)

- An inner join in a select statement assumes that the same information exist in both facts

DAY_DATE	ORDER_METHOD	SALE_TOTAL
01/01/2005	E-mail	$10
01/02/2005	Telephone	$25
01/03/2005	Web	$40
01/04/2005	E-mail	$20

Sales Fact

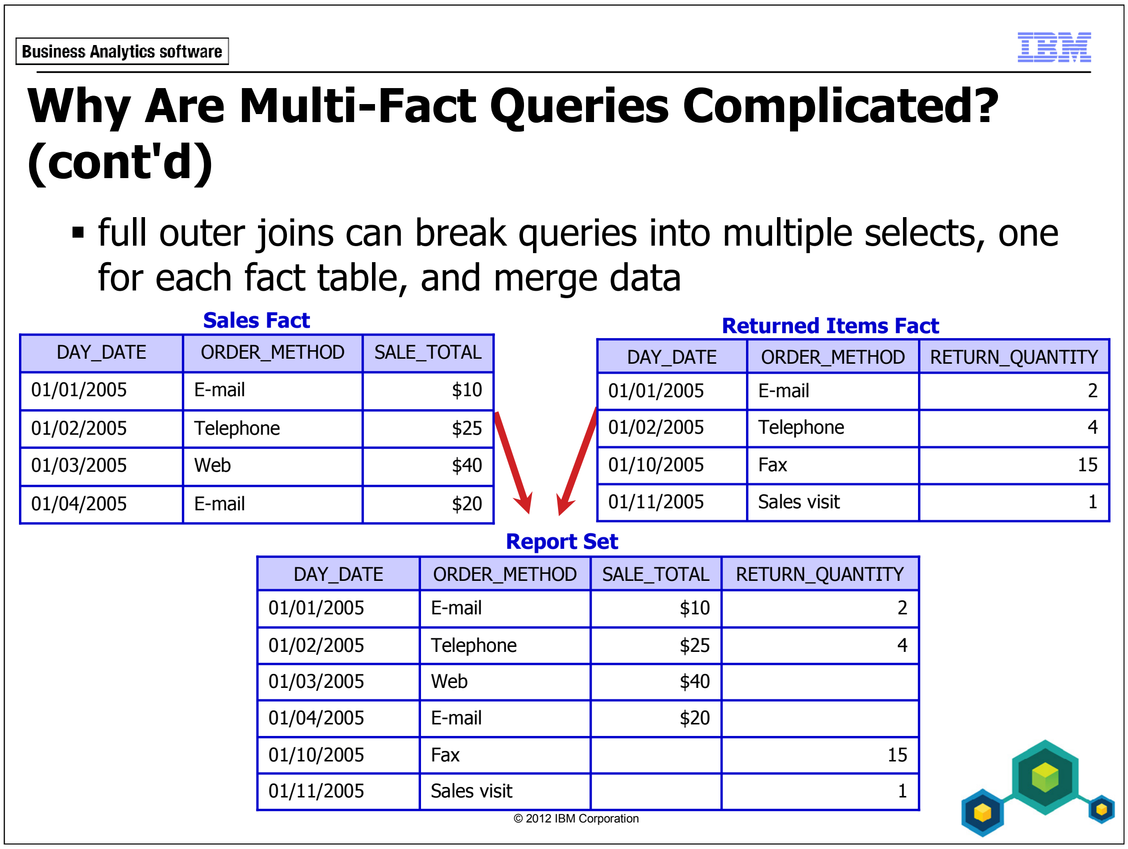
DAY_DATE	ORDER_METHOD	RETURN_QUANTITY
01/01/2005	E-mail	2
01/02/2005	Telephone	4
01/10/2005	Fax	15
01/11/2005	Sales visit	1

Returned Items Fact

```
Select a.SALE_TOTAL, b.RETURN_QUANTITY from Sales a, Returns b
Where a.DAY_DATE = b. DAY_DATE and a.ORDER_METHOD = b.ORDER_METHOD
```

An inner join would only return the first two records. If a filter ORDER_METHOD = 'Web' was added, 0 records would be returned.

Data is lost in this scenario.



Business Analytics software

Why Are Multi-Fact Queries Complicated? (cont'd)

- full outer joins can break queries into multiple selects, one for each fact table, and merge data

Sales Fact

DAY_DATE	ORDER_METHOD	SALE_TOTAL
01/01/2005	E-mail	$10
01/02/2005	Telephone	$25
01/03/2005	Web	$40
01/04/2005	E-mail	$20

Returned Items Fact

DAY_DATE	ORDER_METHOD	RETURN_QUANTITY
01/01/2005	E-mail	2
01/02/2005	Telephone	4
01/10/2005	Fax	15
01/11/2005	Sales visit	1

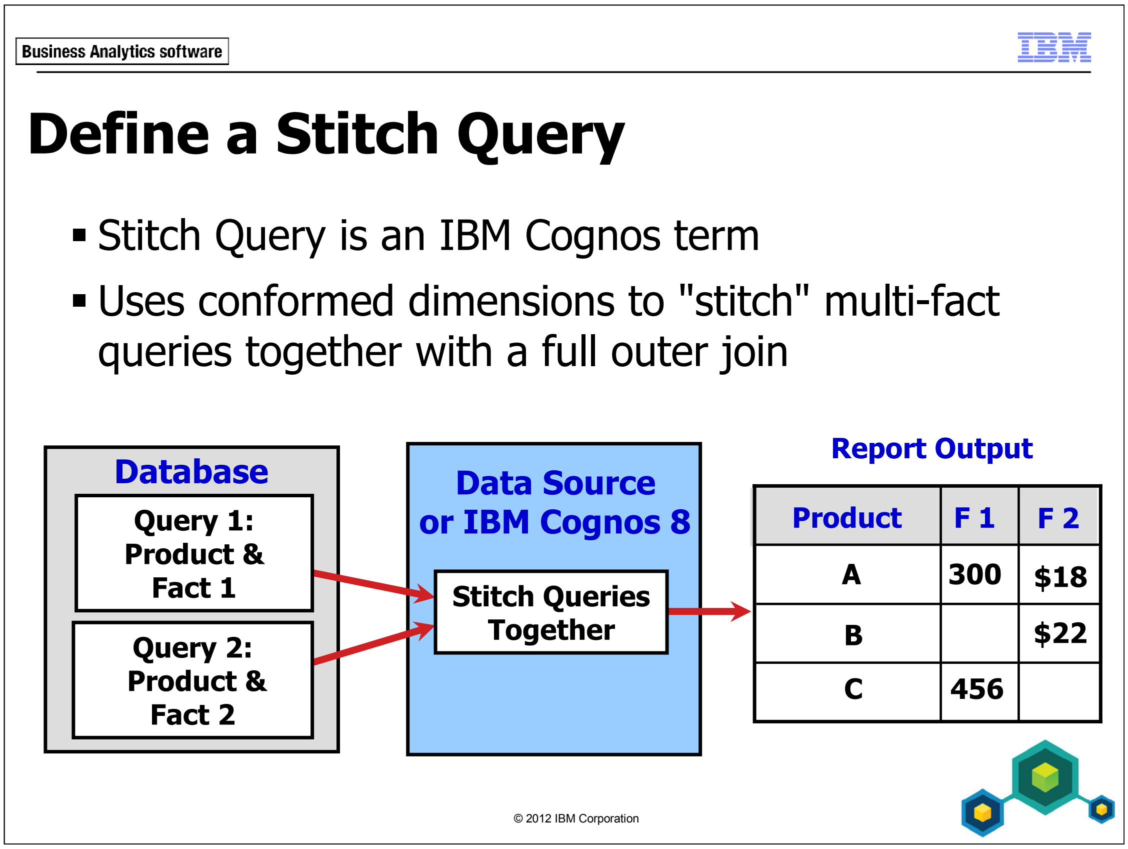
Report Set

DAY_DATE	ORDER_METHOD	SALE_TOTAL	RETURN_QUANTITY
01/01/2005	E-mail	$10	2
01/02/2005	Telephone	$25	4
01/03/2005	Web	$40	
01/04/2005	E-mail	$20	
01/10/2005	Fax		15
01/11/2005	Sales visit		1

In the above example, no data is lost. This is what IBM Cognos refers to as a stitch query.

This type of data pattern is not exclusive to star schema data sources, it is also found in operational systems. For example, an employee has skills and also has billings. Employee is a conformed dimension that relates to a Skills table as well as a Billings table. To avoid losing any skills or billings related to an employee, the two result sets would be stitched together.

This type of query could be further complicated if optional cardinalities (outer joins) were specified. For example, you could have a left outer join on the sales fact from the order method dimension. This would return order methods that had sales as well as those that did not, which means it would be possible to see null values in both fact fields of the report.



Business Analytics software

Define a Stitch Query

- Stitch Query is an IBM Cognos term
- Uses conformed dimensions to "stitch" multi-fact queries together with a full outer join

Database
Query 1: Product & Fact 1
Query 2: Product & Fact 2
Data Source or IBM Cognos 8
Stitch Queries Together
Report Output

Product	F 1	F 2
A	300	$18
B		$22
C	456	

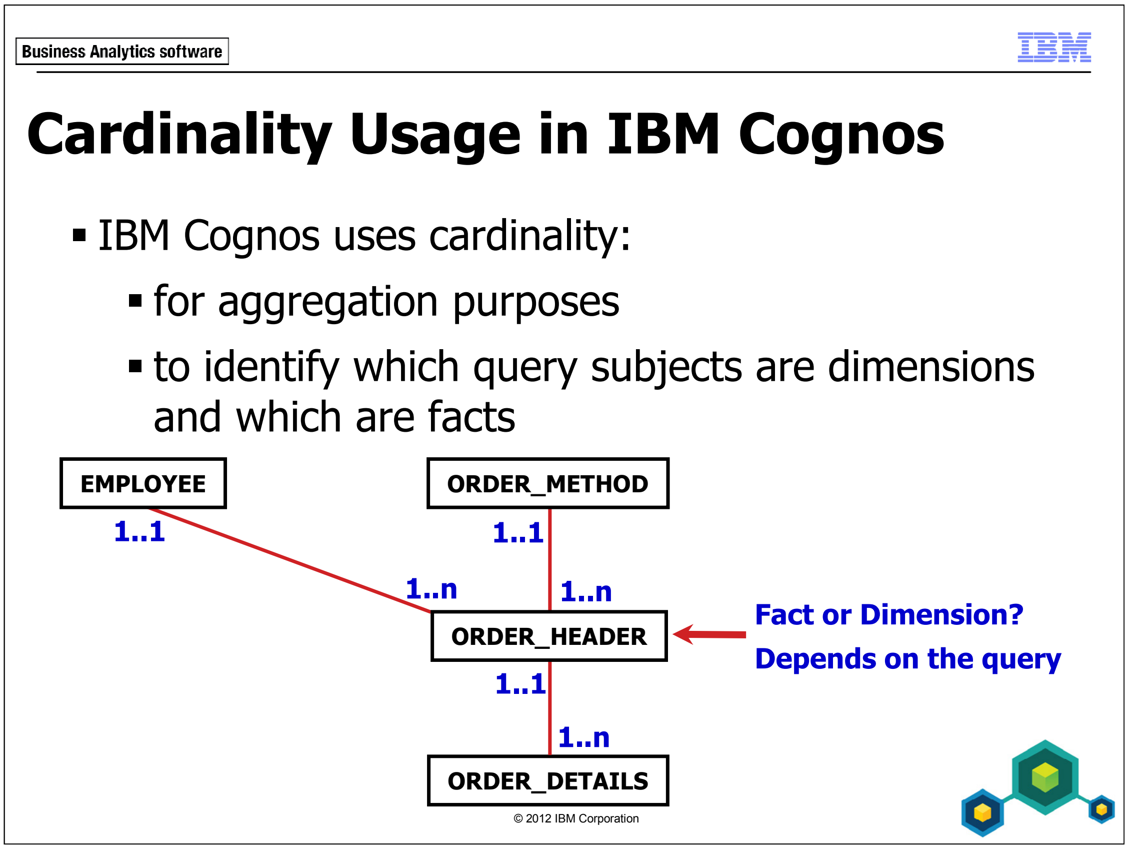
© 2012 IBM Corporation

Conformed dimensions contain the descriptive attributes and corresponding names, meanings, and values that have been agreed to across the enterprise.

Conformed dimensions can be used to merge two independent fact queries together, as seen in the slide example above, without loss of data. In this case, the Product dimension is used to stitch the results of query 1 and query 2 together on the common value selected from the Product dimension. Again, as seen on the previous page, no values are lost on either side of the query. Where the facts have the same data in common, both fact cells contain values. Where a fact does not share common data, the fact cells are null.

If the data source supports this type of query, the processing will be done at the data source level. If not, the processing will be done on the IBM Cognos servers.

We will examine the anatomy of a stitch query later in the course.



Business Analytics software

Cardinality Usage in IBM Cognos

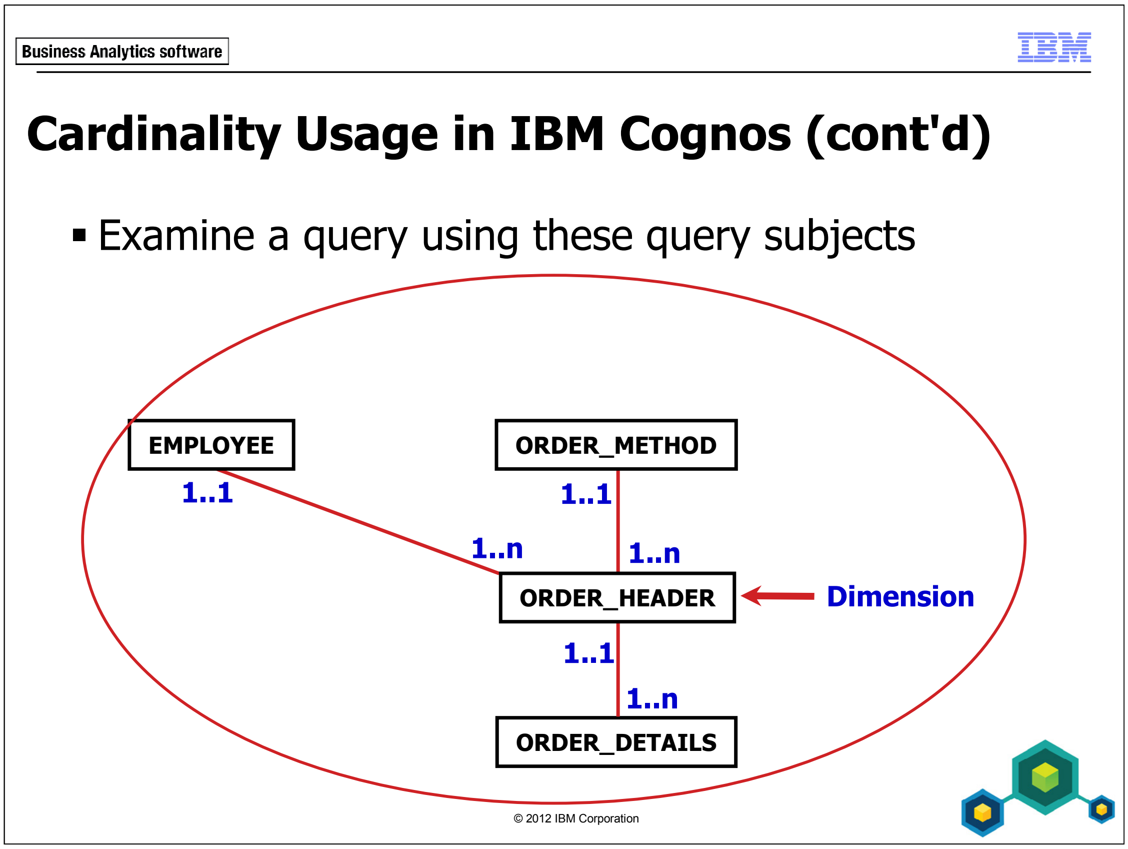
- IBM Cognos uses cardinality:
  - for aggregation purposes
  - to identify which query subjects are dimensions and which are facts

© 2012 IBM Corporation

Typically, facts are found on the 1..n side of a query. The facts found in these tables can then be aggregated if requested by the user.

For every generated query, IBM Cognos identifies each query subject as either a "fact" or a "dimension" based on cardinality. Query subjects with only 0..n or 1..n cardinalities attached are identified as facts and query subjects with 0..1 or 1..1 cardinalities attached are identified as dimensions. Query subjects with both types of cardinality have the potential to be ambiguous. This is the case with ORDER_HEADER.

Note that, although query subjects may contain no measures, such as ORDER_HEADER, IBM Cognos BI will view them as fact tables based on cardinality. Treating the query subjects on the 1..n side on the cardinality as facts is a logical choice when trying to apply aggregation rules since measures are usually on this end of the cardinality. If they were on the 1..1 side, there would not be very much to aggregate.



Business Analytics software

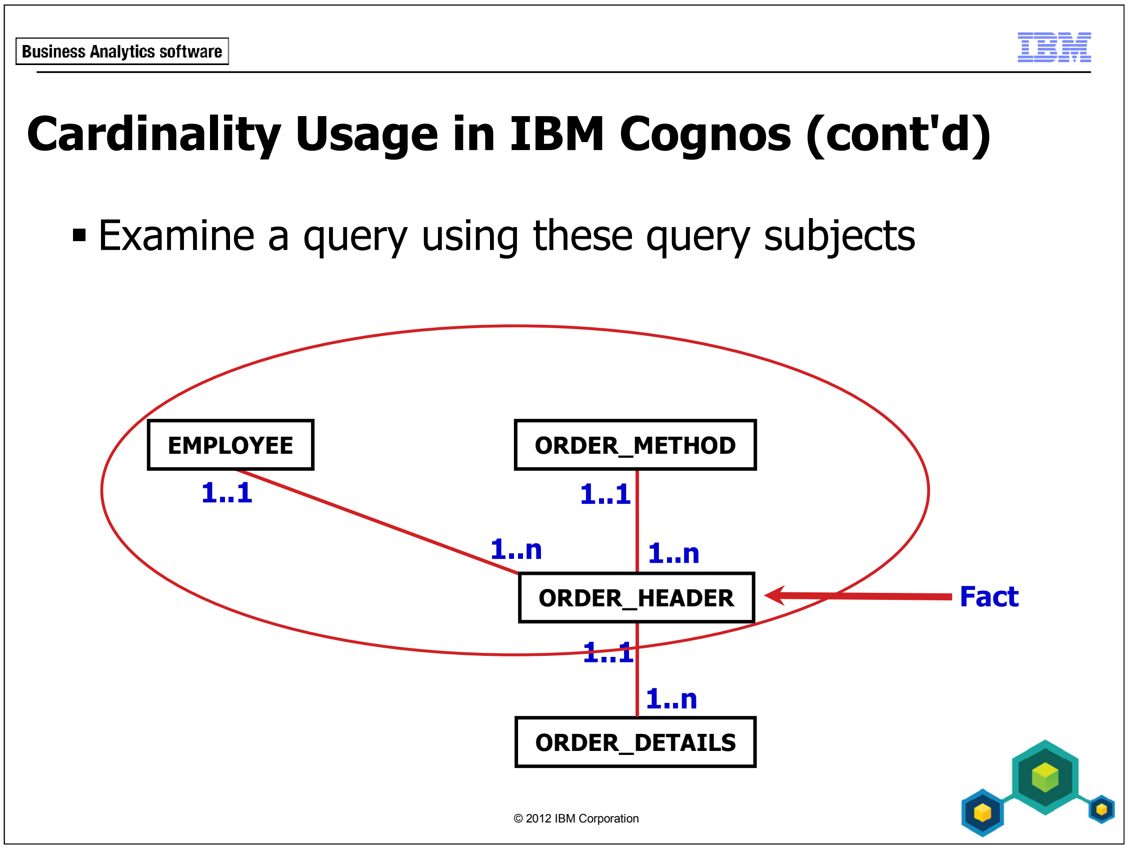
Cardinality Usage in IBM Cognos (cont'd)

- Examine a query using these query subjects

© 2012 IBM Corporation

Again, query usage is applied based on which query subjects are used in a query.

In the slide example, ORDER_HEADER has both 1..1 and 1..n cardinalities attached. Since fact query subjects must have only 1..n or 0..n cardinalities attached, ORDER_HEADER will be treated as a dimension in this scenario.



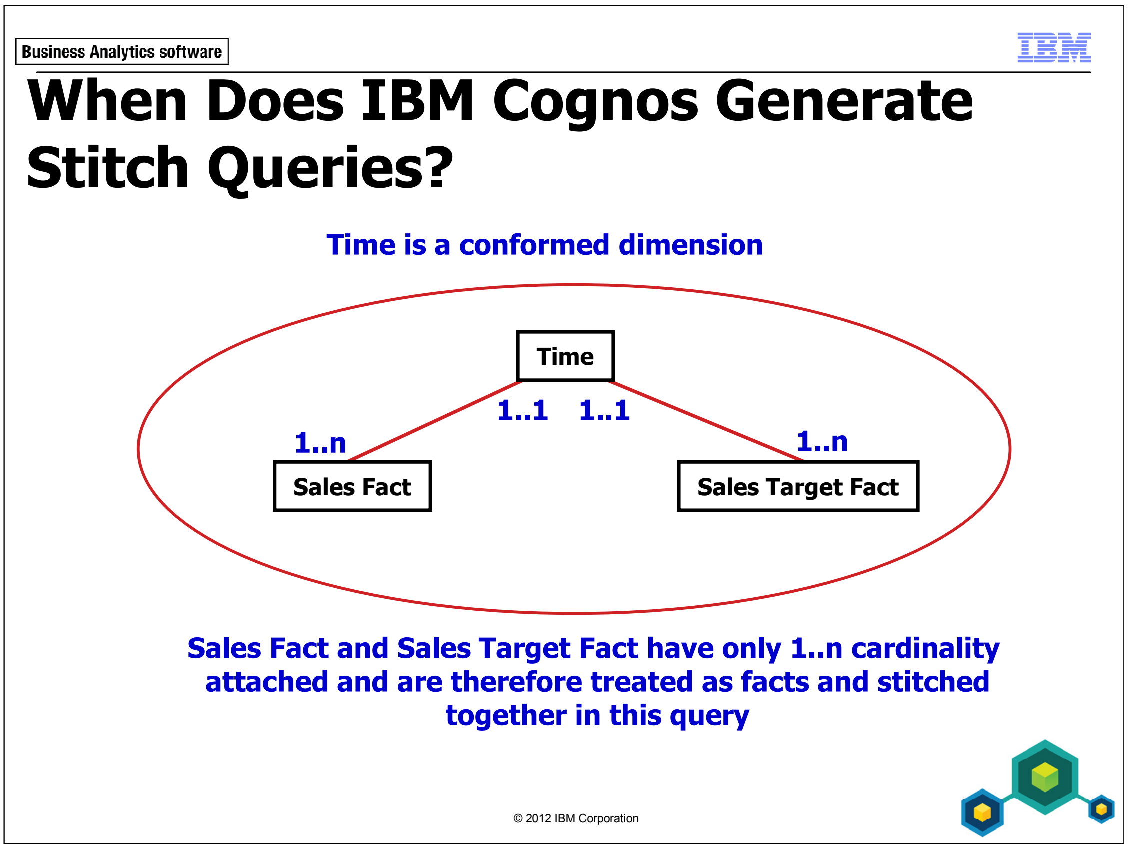
Business Analytics software

Cardinality Usage in IBM Cognos (cont'd)

- Examine a query using these query subjects

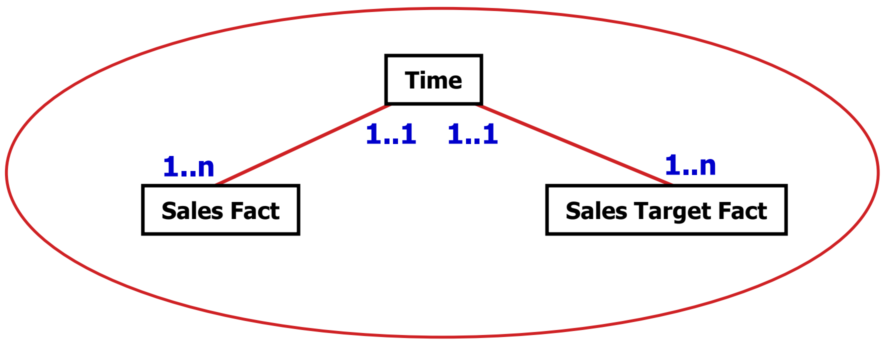
© 2012 IBM Corporation

In this query scenario, ORDER_HEADER has only 1..n cardinalities attached. Therefore it will be treated as a fact.



Business Analytics software

When Does IBM Cognos Generate Stitch Queries?

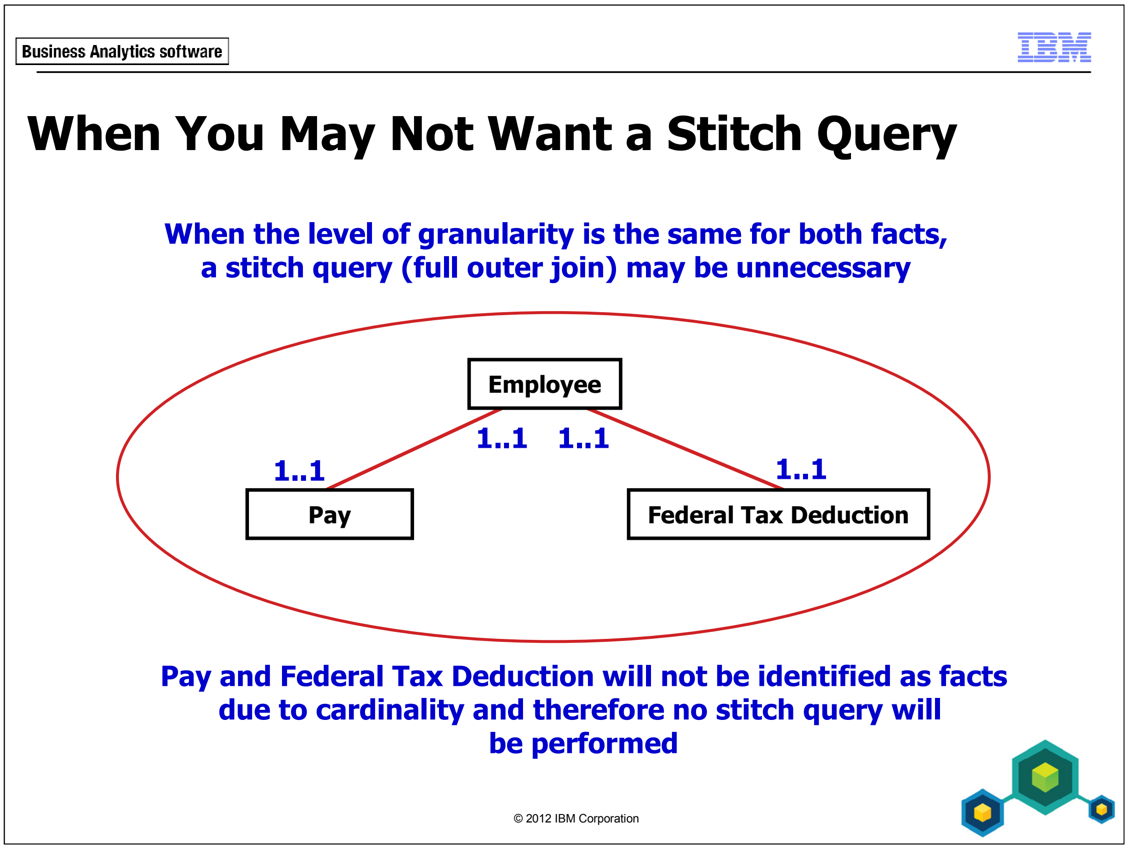


Sales Fact and Sales Target Fact have only 1..n cardinality attached and are therefore treated as facts and stitched together in this query

© 2012 IBM Corporation

When one or more conformed dimensions are used in a multi-fact query (where query subjects are identified as facts by the IBM Cognos query engine), a stitch query (full outer join) will occur.

A stitch query is required when you have different levels of granularity between the facts and do not want to lose uncommon rows of data.



Business Analytics software

When You May Not Want a Stitch Query

When the level of granularity is the same for both facts, a stitch query (full outer join) may be unnecessary

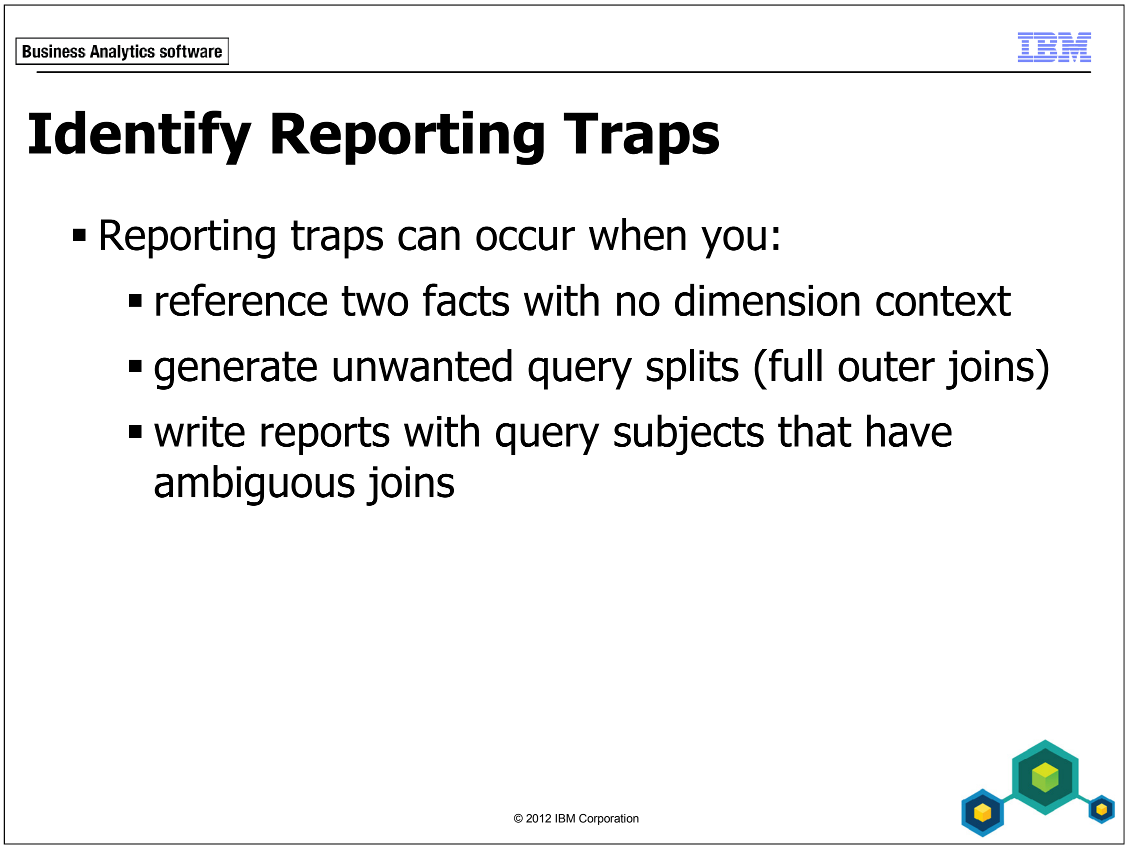
Employee
1..1
1..1
1..1
1..1
Pay
Federal Tax Deduction

Pay and Federal Tax Deduction will not be identified as facts due to cardinality and therefore no stitch query will be performed

© 2012 IBM Corporation

In cases where the facts are at the same level of granularity, you may want to avoid the extra processing of a full outer join and simply configure 1..1 to 1..1 relationships between your dimension and fact query subjects.

You must be extremely familiar with your data and the expected results when implementing this type of configuration. Use this approach with caution as there is potential to prevent required stitch queries as the model scales and involves dimensions with different levels of granularity.


Business Analytics software

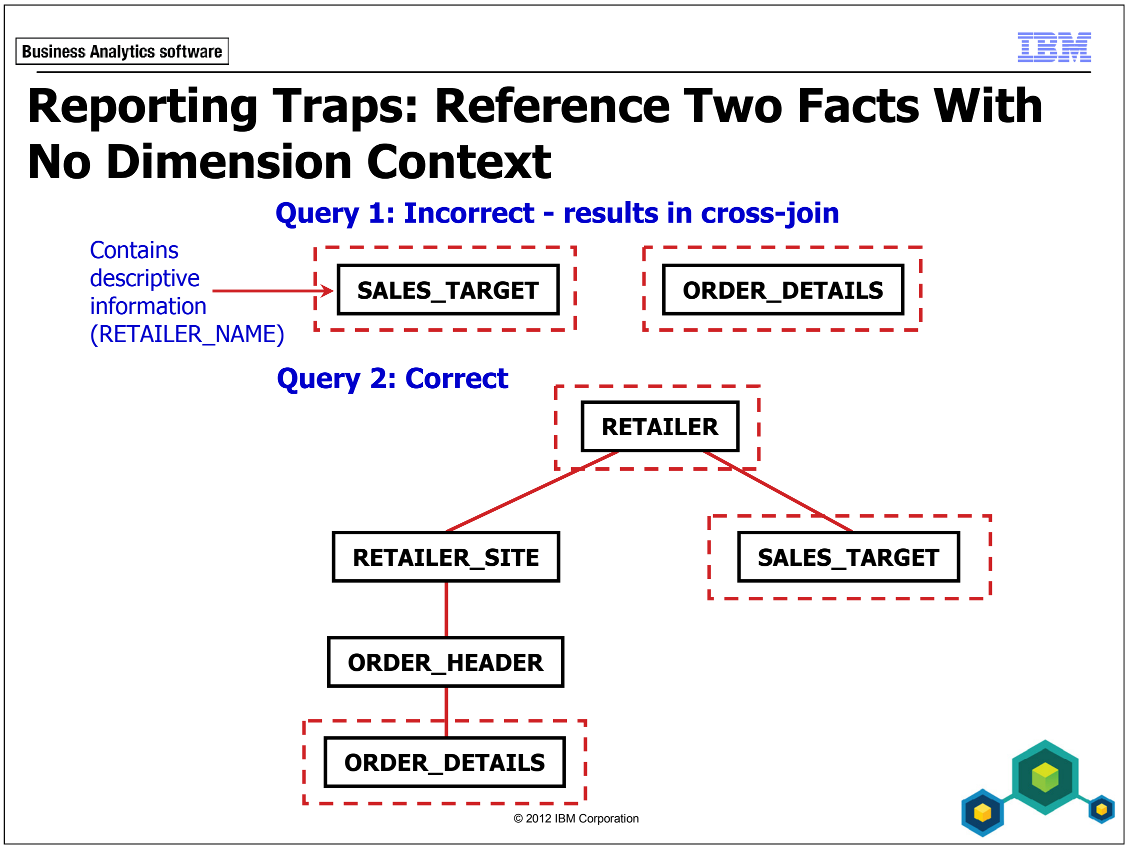
Identify Reporting Traps

- Reporting traps can occur when you:
  - reference two facts with no dimension context
  - generate unwanted query splits (full outer joins)
  - write reports with query subjects that have ambiguous joins

© 2012 IBM Corporation

When beginning to model for predictable results, you will want to initially identify any reporting traps that exist within your model.

We will cover these reporting traps throughout this module.



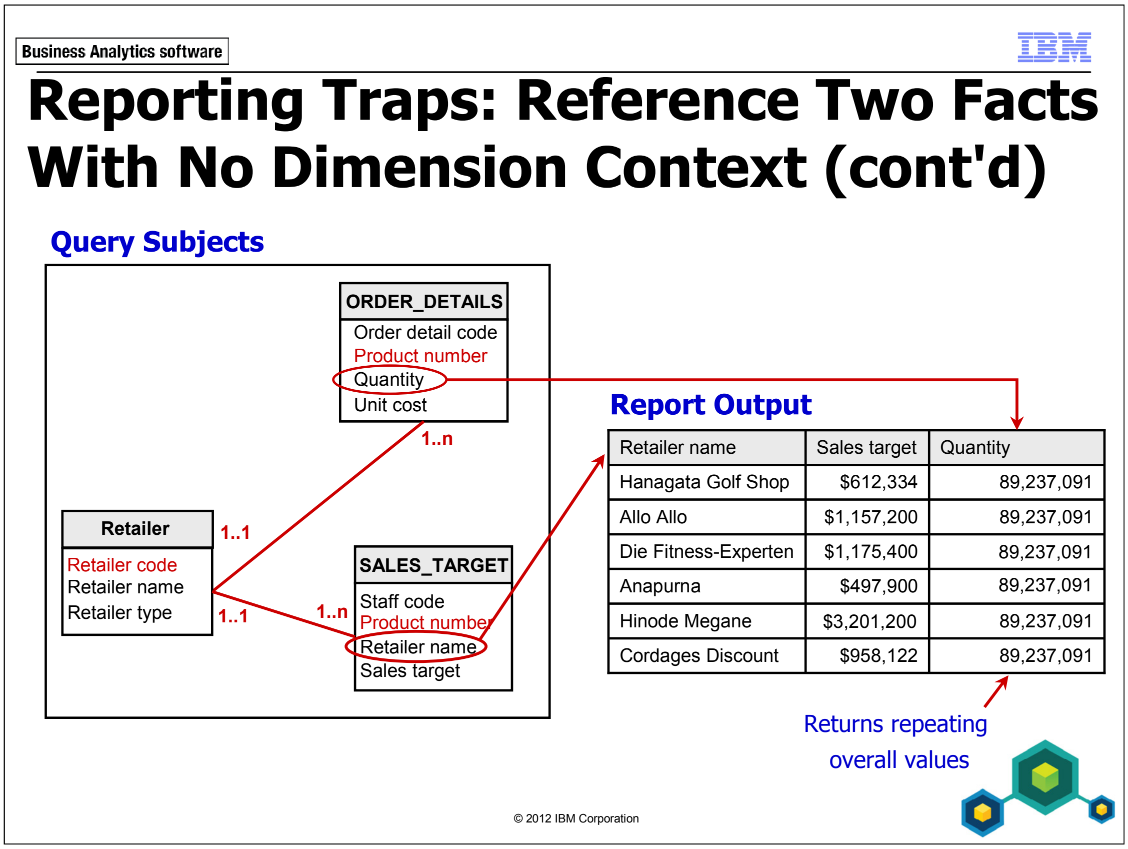
Business Analytics software

Reporting Traps: Reference Two Facts With No Dimension Context

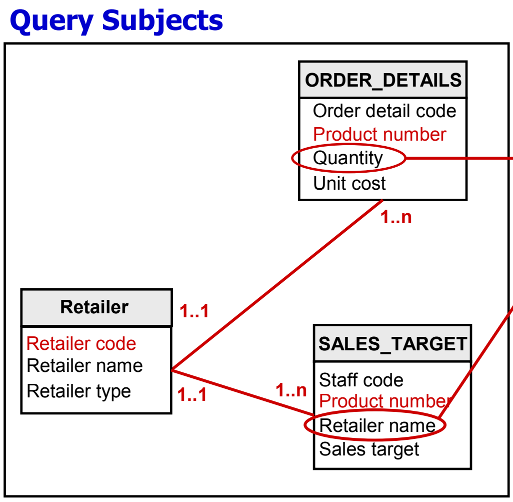
© 2012 IBM Corporation

A query on two facts with no dimension context is treated as a cross-product join. This brings duplicate data from the two datasets together in one query. Authors can accidentally author these types of queries if you have left textual (dimensional information) items in a fact query subject. For example, if Retailer name was a query item in SALES_TARGET, authors might choose this item thinking it also applies to ORDER_DETAILS.

This can be avoided if textual items are removed from the fact query subjects. Doing so will force authors to use at least one common dimension in the query to join the facts together. In Query 2, retailer information is taken from RETAILER.


Business Analytics software
IBM

Reporting Traps: Reference Two Facts With No Dimension Context (cont'd)



Report Output

Retailer name	Sales target	Quantity
Hanagata Golf Shop	$612,334	89,237,091
Allo Allo	$1,157,200	89,237,091
Die Fitness-Experten	$1,175,400	89,237,091
Anapurna	$497,900	89,237,091
Hinode Megane	$3,201,200	89,237,091
Cordages Discount	$958,122	89,237,091

Returns repeating overall values

© 2012 IBM Corporation

The query above returns the distinct overall total of Quantity and applies it to each Retailer name row.

Again, in this scenario, authors can accidentally misuse Retailer name from the SALES_TARGET fact query subject and cause unexpected results. Taking Retailer name from the conformed dimension would generate a stitch query and provide expected results.

Demo 1: Reference Two Facts with No Dimension Context

Purpose:
A common requirement for the business analysts will be to generate a report that compares sales targets against actual sales. You will test such a report to identify possible reporting traps.

Components:	**Framework Manager**, **IBM Cognos Report Studio**
Project:	**GO Operational**
Package:	**GO Operational**

Task 1. Try to create a Targets vs Actual Revenue report.

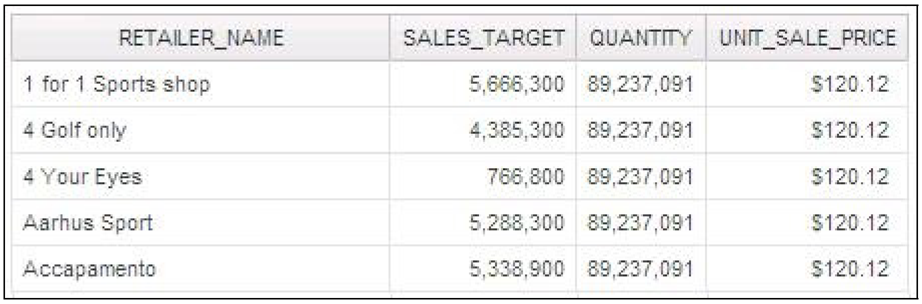
1. In **Framework Manager**, open the **GO Operational** project located at **C:\Edcognos\B5252\ CBIFM-Start Files\Module 6\GO Operational**.
2. Log in as User ID **admin**, Password **Education1**, if prompted.
3. Publish the **GO Operational** package
4. Launch **IBM Cognos Connection**, log in, and launch **IBM Cognos Report Studio**, selecting the **GO Operational** package for a **List** report.
5. Drag the following query items to the report:

Query Subject	Query Item
SALES_TARGET	**RETAILER_NAME** **SALES_TARGET**
ORDER_DETAILS	**QUANTITY** **UNIT_SALE_PRICE**

You could have taken COMPANY_NAME from RETAILER, but to identify potential areas of unpredictability for authors you will take it from SALES_TARGET.

6. Click the arrow beside the **Run** button and select **Run Report – HTML**.

   The results appear as follows:



RETAILER_NAME	SALES_TARGET	QUANTITY	UNIT_SALE_PRICE
1 for 1 Sports shop	5,666,300	89,237,091	$120.12
4 Golf only	4,385,300	89,237,091	$120.12
4 Your Eyes	766,800	89,237,091	$120.12
Aarhus Sport	5,288,300	89,237,091	$120.12
Accapamento	5,338,900	89,237,091	$120.12

   The resulting report has the same QUANTITY and UNIT_SALE_PRICE for every row, which is unexpected. To complete the report, you would create a calculation of quantity times unit sale price to get actual revenue values, but there is no need to continue since an issue has already been found.

7. Close your Internet browser.

Task 2. Analyze the query in Framework Manager.

1. In **Framework Manager**, in the **Diagram**, locate **SALES_TARGET** and **ORDER_DETAILS**.

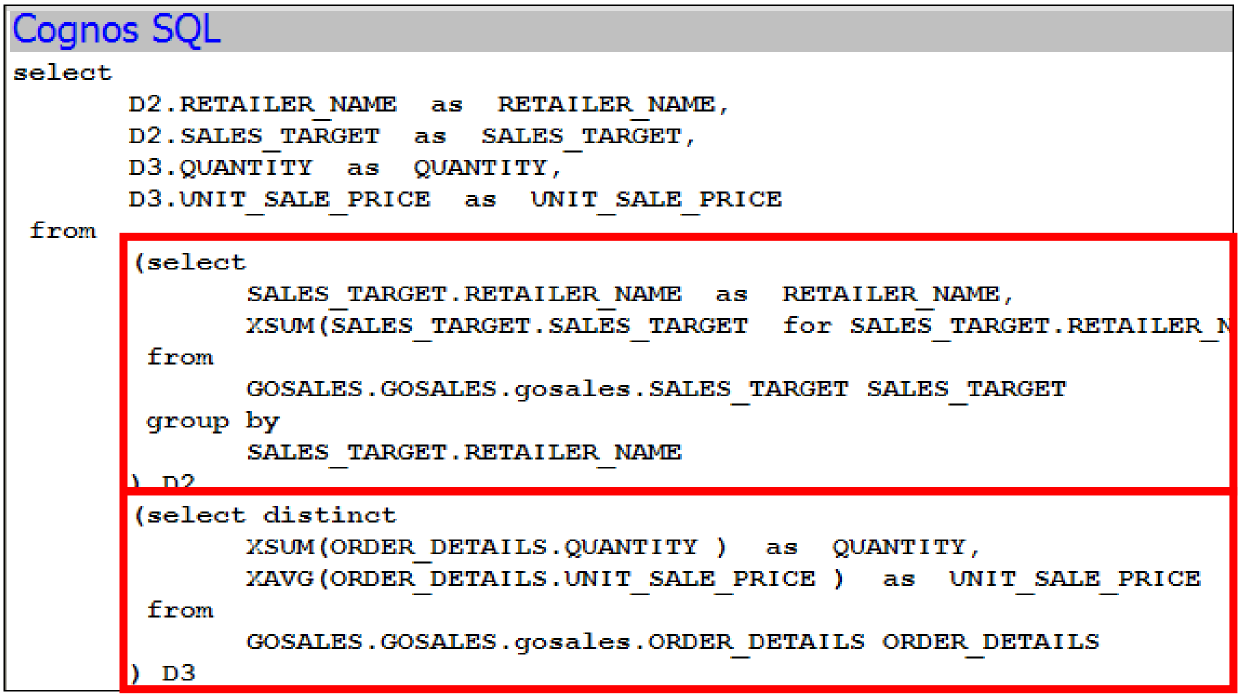
   There are no direct relationships between these two query subjects. To see how they are joined in a query, you will recreate the same query in Framework Manager and view the generated SQL.

2. In the **Project Viewer**, select and test the following query items with **Auto Sum** enabled:

Query Subject	Query Item
SALES_TARGET	**RETAILER_NAME** **SALES_TARGET**
ORDER_DETAILS	**QUANTITY** **UNIT_SALE_PRICE**

3. Click the **Query Information** tab.

   The results appear as follows:



Cognos SQL

```
select
       D2.RETAILER_NAME  as  RETAILER_NAME,
       D2.SALES_TARGET  as  SALES_TARGET,
       D3.QUANTITY  as  QUANTITY,
       D3.UNIT_SALE_PRICE  as  UNIT_SALE_PRICE
 from
       (select
              SALES_TARGET.RETAILER_NAME  as  RETAILER_NAME,
              XSUM(SALES_TARGET.SALES_TARGET  for SALES_TARGET.RETAILER_N
        from
              GOSALES.GOSALES.gosales.SALES_TARGET SALES_TARGET
        group by
              SALES_TARGET.RETAILER_NAME
       ) D2
       (select distinct
              XSUM(ORDER_DETAILS.QUANTITY )  as  QUANTITY,
              XAVG(ORDER_DETAILS.UNIT_SALE_PRICE )  as  UNIT_SALE_PRICE
        from
              GOSALES.GOSALES.gosales.ORDER_DETAILS ORDER_DETAILS
       ) D3
```

The query has two subqueries (aliased as D2 and D3) in the **from** clause that are not joined. The two queries are simply separated by a comma with no join conditions. This is known as a cross-product join. In this case, one query is on all sales targets grouped by retailer name, and the other is on a distinct value for all order quantities and unit sale prices with no grouping. This explains why all the quantities and prices are the same. IBM Cognos does not join the two query structures because they are both facts and there is no conformed (shared) dimension joining them.

You will need an alternate source for RETAILER_NAME, so you will try the only other instance of RETAILER_NAME in the project, the one in ORDER_HEADER.

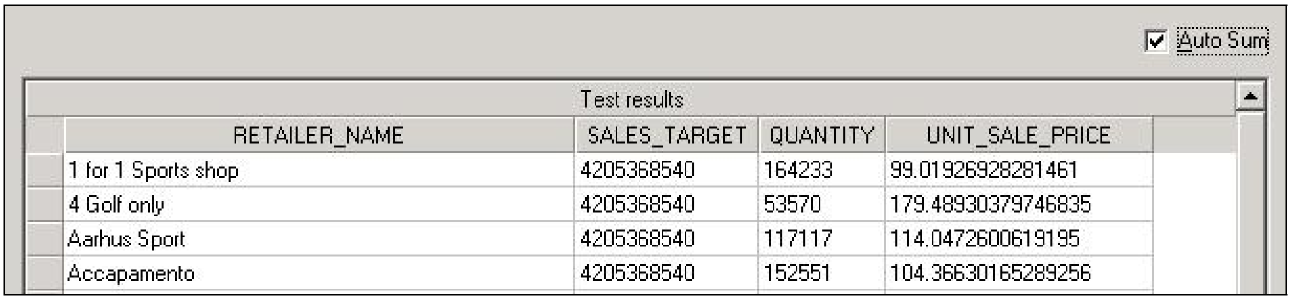
4. Click **Close**.

Task 3. Attempt a resolution.

1. Select and test the following query items with **Auto Sum** enabled:

Query Subject	Query Item
ORDER_HEADER	**RETAILER_NAME**
SALES_TARGET	**SALES_TARGET**
ORDER_DETAILS	**QUANTITY** **UNIT_SALE_PRICE**

The results appear as follows:

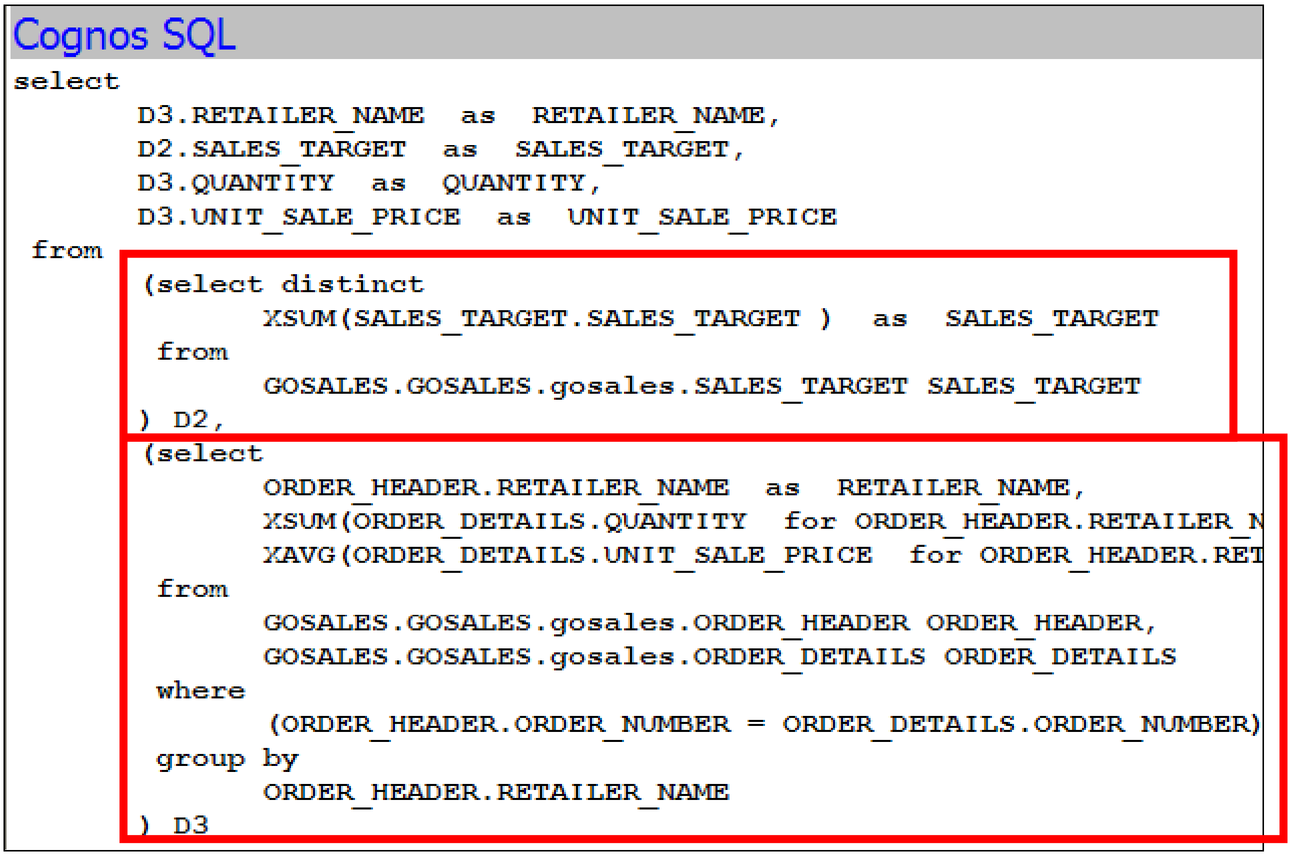


RETAILER_NAME	SALES_TARGET	QUANTITY	UNIT_SALE_PRICE
1 for 1 Sports shop	4205368540	164233	99.01926928281461
4 Golf only	4205368540	53570	179.48930379746835
Aarhus Sport	4205368540	117117	114.0472600619195
Accapamento	4205368540	152551	104.36630165289256

You can now see separate quantities and unit sale prices for each retailer, but the sales targets are now all identical.

2. Click the **Query Information** tab.

   The results appear as follows:



Cognos SQL

```
select
       D3.RETAILER_NAME  as  RETAILER_NAME,
       D2.SALES_TARGET  as  SALES_TARGET,
       D3.QUANTITY  as  QUANTITY,
       D3.UNIT_SALE_PRICE  as  UNIT_SALE_PRICE
 from
       (select distinct
              XSUM(SALES_TARGET.SALES_TARGET )  as  SALES_TARGET
        from
              GOSALES.GOSALES.gosales.SALES_TARGET SALES_TARGET
       ) D2,
       (select
              ORDER_HEADER.RETAILER_NAME  as  RETAILER_NAME,
              XSUM(ORDER_DETAILS.QUANTITY  for ORDER_HEADER.RETAILER_N
              XAVG(ORDER_DETAILS.UNIT_SALE_PRICE  for ORDER_HEADER.RET
        from
              GOSALES.GOSALES.gosales.ORDER_HEADER ORDER_HEADER,
              GOSALES.GOSALES.gosales.ORDER_DETAILS ORDER_DETAILS
        where
              (ORDER_HEADER.ORDER_NUMBER = ORDER_DETAILS.ORDER_NUMBER)
        group by
              ORDER_HEADER.RETAILER_NAME
       ) D3
```

   Once again you have two subqueries that are not joined. However, this time it is sales targets that are not grouped by retailer name. IBM Cognos still does not join the two query subjects because you have not provided a conformed dimension.

   If you speak with the database administrator, you will find out that COMPANY_NAME in the RETAILER query subject holds the retailer name that can be used between the two fact query subjects. You will try this in the next task.

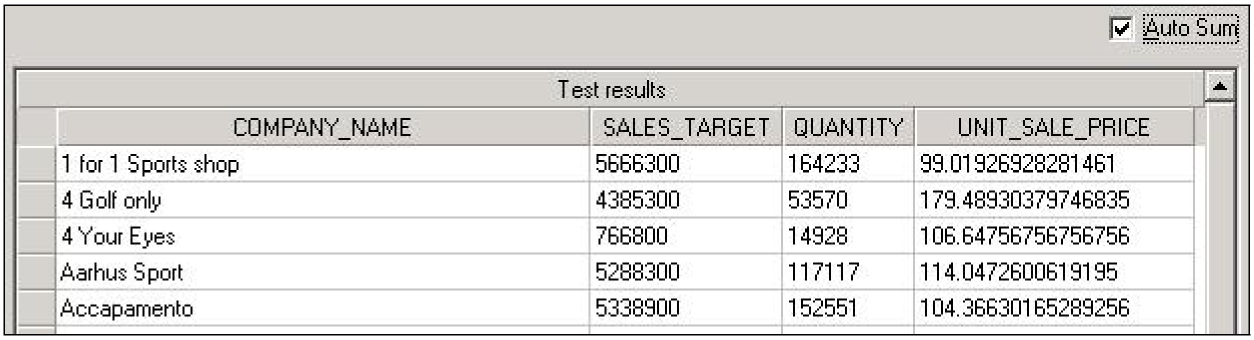
3. Click **Close**.

Task 4. Correct the query.

1. Select and test the following query items with **Auto Sum** enabled:

Query Subject	Query Item
RETAILER	**COMPANY_NAME**
SALES_TARGET	**SALES_TARGET**
ORDER_DETAILS	**QUANTITY** **UNIT_SALE_PRICE**

The results appear as follows:



☑ Auto Sum

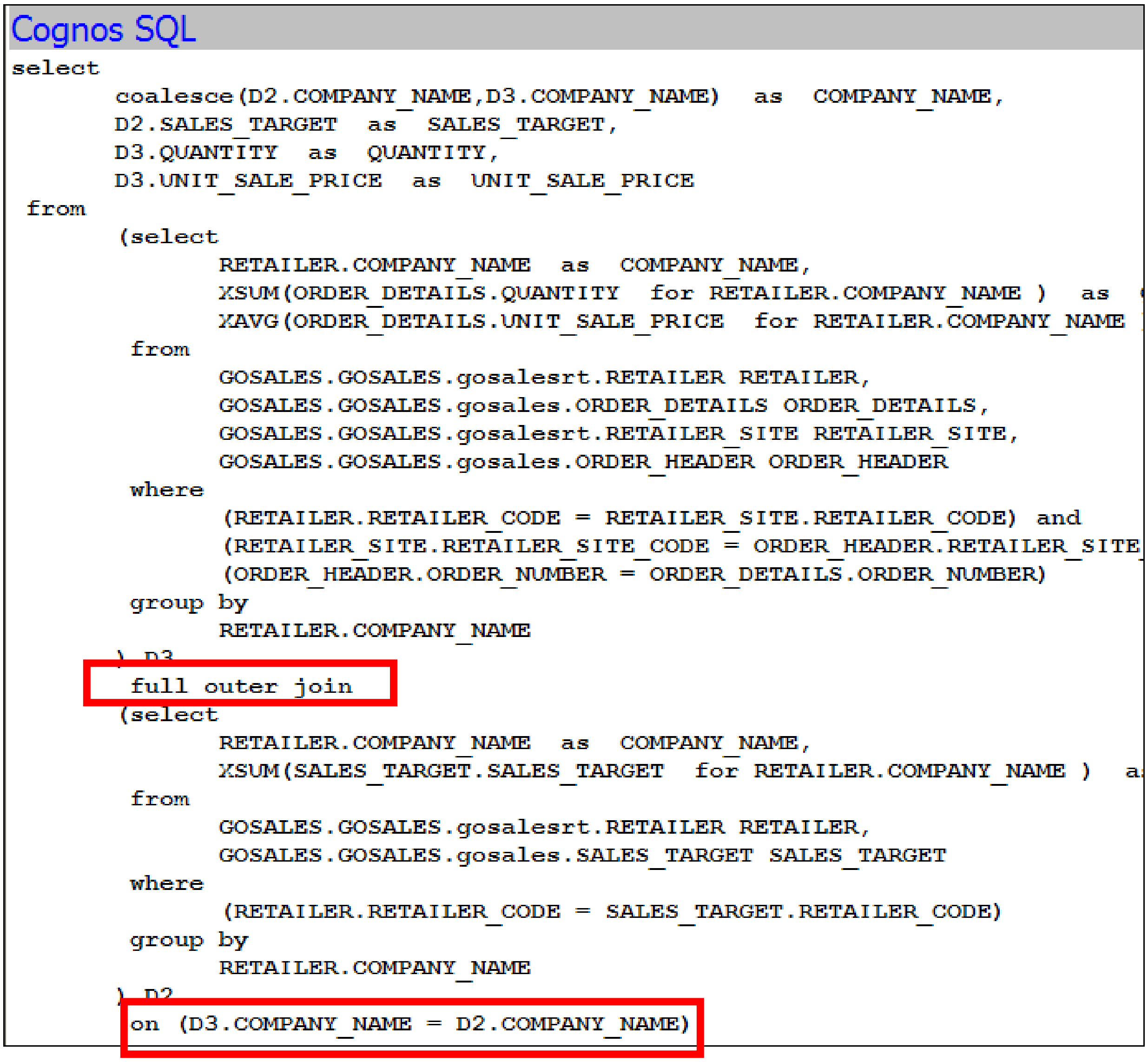
Test results

COMPANY_NAME	SALES_TARGET	QUANTITY	UNIT_SALE_PRICE
1 for 1 Sports shop	5666300	164233	99.01926928281461
4 Golf only	4385300	53570	179.48930379746835
4 Your Eyes	766800	14928	106.64756756756756
Aarhus Sport	5288300	117117	114.0472600619195
Accapamento	5338900	152551	104.36630165289256

You can now see separate targets, quantities, and prices for each retailer. If you were to create this report in Query Studio, you would now calculate Actual Revenue to compare against the targets. (You will not do this now in order to continue investigating modeling issues and will create a Revenue calculation in the model later in the course.)

2. Click the **Query Information** tab.

   The results appear as follows:



Cognos SQL

```
select
       coalesce(D2.COMPANY_NAME,D3.COMPANY_NAME)  as  COMPANY_NAME,
       D2.SALES_TARGET  as  SALES_TARGET,
       D3.QUANTITY  as  QUANTITY,
       D3.UNIT_SALE_PRICE  as  UNIT_SALE_PRICE
 from
       (select
              RETAILER.COMPANY_NAME  as  COMPANY_NAME,
              XSUM(ORDER_DETAILS.QUANTITY  for RETAILER.COMPANY_NAME )  as
              XAVG(ORDER_DETAILS.UNIT_SALE_PRICE  for RETAILER.COMPANY_NAME
        from
              GOSALES.GOSALES.gosalesrt.RETAILER RETAILER,
              GOSALES.GOSALES.gosales.ORDER_DETAILS ORDER_DETAILS,
              GOSALES.GOSALES.gosalesrt.RETAILER_SITE RETAILER_SITE,
              GOSALES.GOSALES.gosales.ORDER_HEADER ORDER_HEADER
        where
              (RETAILER.RETAILER_CODE = RETAILER_SITE.RETAILER_CODE) and
              (RETAILER_SITE.RETAILER_SITE_CODE = ORDER_HEADER.RETAILER_SITE
              (ORDER_HEADER.ORDER_NUMBER = ORDER_DETAILS.ORDER_NUMBER)
        group by
              RETAILER.COMPANY_NAME
       ) D3
        full outer join
       (select
              RETAILER.COMPANY_NAME  as  COMPANY_NAME,
              XSUM(SALES_TARGET.SALES_TARGET  for RETAILER.COMPANY_NAME )  a
        from
              GOSALES.GOSALES.gosalesrt.RETAILER RETAILER,
              GOSALES.GOSALES.gosales.SALES_TARGET SALES_TARGET
        where
              (RETAILER.RETAILER_CODE = SALES_TARGET.RETAILER_CODE)
        group by
              RETAILER.COMPANY_NAME
       ) D2
        on (D3.COMPANY_NAME = D2.COMPANY_NAME)
```

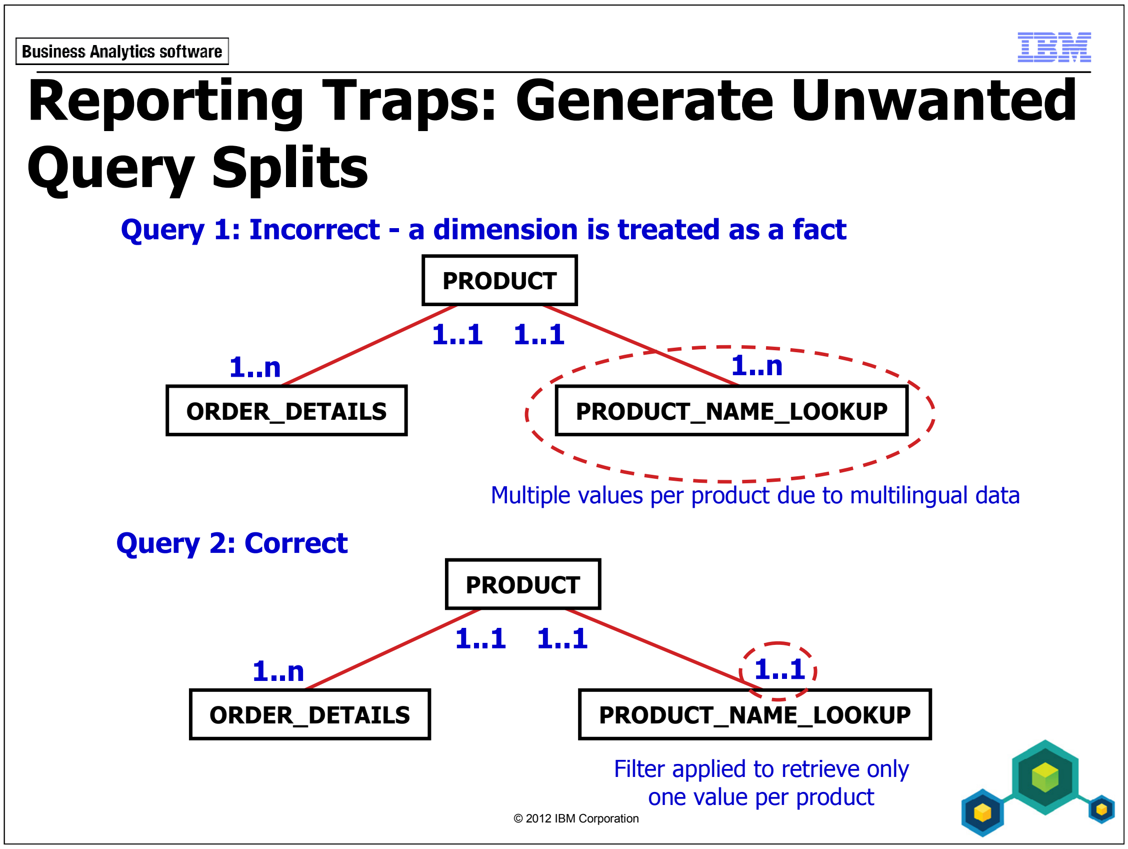
There are two subqueries joined by COMPANY_NAME (retailer name) through a full outer join. Again, in this case, a full outer join is expected since you are stitching two separate fact queries together.

The query scenarios you just went through support the concept of ensuring descriptive data is retrieved from dimensions and not facts.

3. Click **Close**, and leave Framework Manager open for the next demo.

Results:
A common requirement for the business analysts will be to compare sales targets against actual sales. You tested such a query, identified a reporting trap, and identified a proper query to return expected results.

In the Framework Manager diagram, review the join paths for this query. RETAILER goes through RETAILER_SITE and ORDER_HEADER to get to ORDER_DETAILS and has a direct relationship with SALES_TARGET_FACT.



Business Analytics software

Reporting Traps: Generate Unwanted Query Splits

Query 1: Incorrect - a dimension is treated as a fact
PRODUCT
1..1
1..1
1..n
1..n
ORDER_DETAILS
PRODUCT_NAME_LOOKUP
Multiple values per product due to multilingual data
Query 2: Correct
PRODUCT
1..1
1..1
1..n
1..1
ORDER_DETAILS
PRODUCT_NAME_LOOKUP
Filter applied to retrieve only one value per product

© 2012 IBM Corporation

Unwanted query splits (full outer joins) occur when query subjects (such as PRODUCT_NAME_LOOKUP in Query 1) are incorrectly identified as facts.

In this case PRODUCT_NAME_LOOKUP contains several values for each product in the PRODUCT table, one for each supported language in this multilingual table. The intent is for PRODUCT_NAME_LOOKUP to behave as a lookup table and return one value per product. Instead it acts as a fact based on cardinality.

In Query 2, the cardinality for PRODUCT_NAME_LOOKUP has been changed to 1..1 after a filter is put in place to ensure only one value is returned. In this scenario, it will always act as a dimension.

This issue will be resolved in the Create Calculations and Filters module.

Demo 2: Generate Unwanted Query Splits

Purpose:
Authors will want to report on the overall quantity of products sold. You will test such a query to identify a possible reporting trap.

Component:	**Framework Manager**
Project:	**GO Operational**

Task 1. Create a query to view total quantity sold by product.

1. In **Framework Manager**, select and test the following query items with **Auto Sum** enabled:

Query Subject	Query Item
PRODUCT_NAME_LOOKUP	**PRODUCT_NAME**
PRODUCT	**PRODUCT_NUMBER**
ORDER_DETAILS	**QUANTITY**

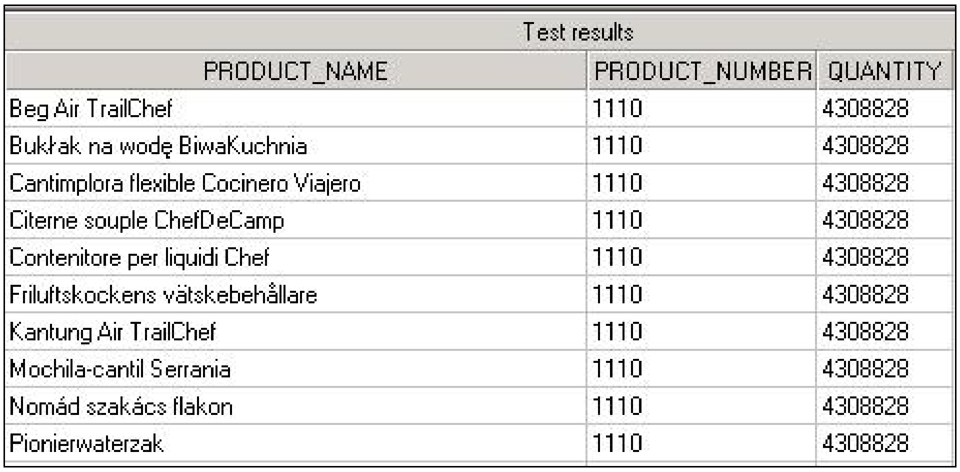
In order to return a larger result set, you will alter the options.

2. Click **Options** in the lower right corner, under **Number of results**, change the value from **25** to **250**, and then click **OK**.

Other options will be covered later in the course.

3. Click **Test Sample**.

   The results are similar to the following:



Test results

PRODUCT_NAME	PRODUCT_NUMBER	QUANTITY
Beg Air TrailChef	1110	4308828
Bukłak na wodę BiwaKuchnia	1110	4308828
Cantimplora flexible Cocinero Viajero	1110	4308828
Citerne souple ChefDeCamp	1110	4308828
Contenitore per liquidi Chef	1110	4308828
Friluftskockens vätskebehållare	1110	4308828
Kantung Air TrailChef	1110	4308828
Mochila-cantil Serrania	1110	4308828
Nomád szakács flakon	1110	4308828
Pionierwaterzak	1110	4308828

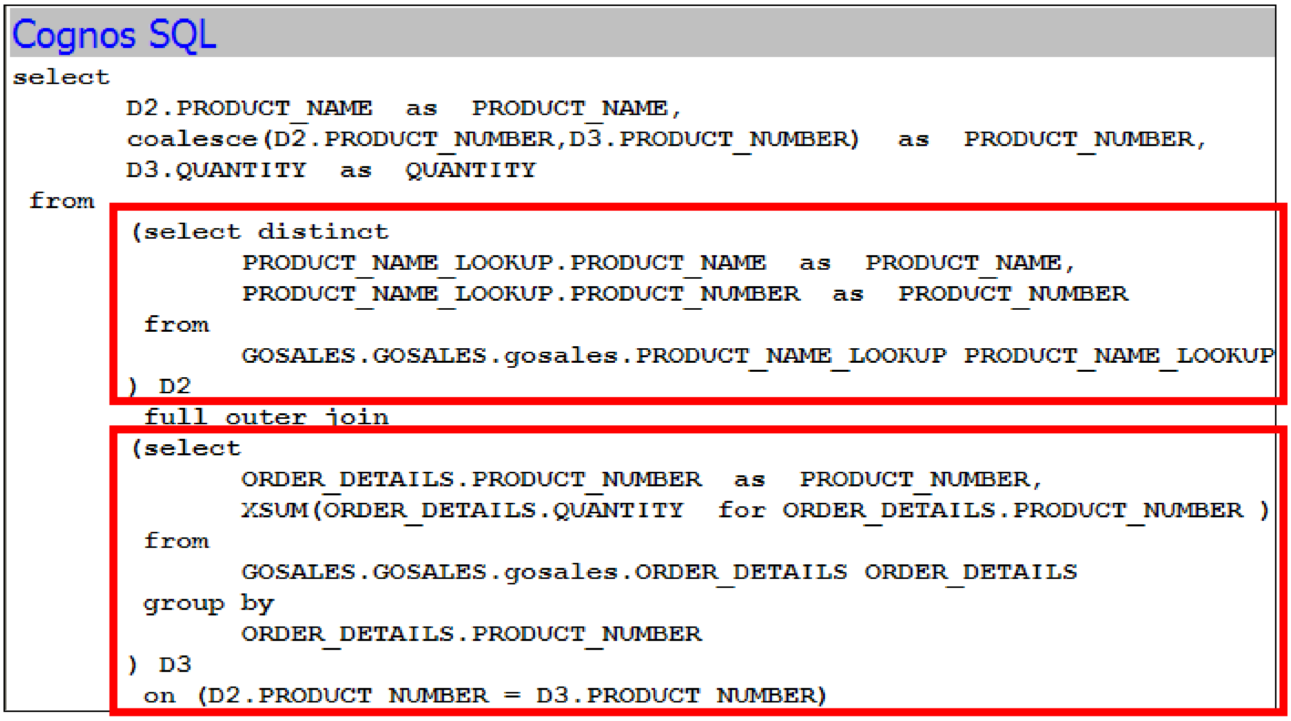
   Notice the different language name but the same quantity value for the same product number.

   PRODUCT_NAME_LOOKUP is a table in the database that contains multiple rows for each product, one for each supported language. Because there are multiple rows, the relationship is currently correct. 1..1 from PRODUCT to 1..n to PRODUCT_NAME_LOOKUP. It is not that this query is incorrect, but it may not be what the author expected. Likely you would want to see one product, in one language, with one summarized QUANTITY value.

   You will look at the SQL to see what occurred.

4. Click the **Query Information** tab.

   The results appear as follows:



Cognos SQL

```
select
       D2.PRODUCT_NAME  as  PRODUCT_NAME,
       coalesce(D2.PRODUCT_NUMBER,D3.PRODUCT_NUMBER)  as  PRODUCT_NUMBER,
       D3.QUANTITY  as  QUANTITY
 from
       (select distinct
              PRODUCT_NAME_LOOKUP.PRODUCT_NAME  as  PRODUCT_NAME,
              PRODUCT_NAME_LOOKUP.PRODUCT_NUMBER  as  PRODUCT_NUMBER
        from
              GOSALES.GOSALES.gosales.PRODUCT_NAME_LOOKUP PRODUCT_NAME_LOOKUP
       ) D2
        full outer join
       (select
              ORDER_DETAILS.PRODUCT_NUMBER  as  PRODUCT_NUMBER,
              XSUM(ORDER_DETAILS.QUANTITY  for ORDER_DETAILS.PRODUCT_NUMBER )
        from
              GOSALES.GOSALES.gosales.ORDER_DETAILS ORDER_DETAILS
        group by
              ORDER_DETAILS.PRODUCT_NUMBER
       ) D3
        on (D2.PRODUCT_NUMBER = D3.PRODUCT_NUMBER)
```

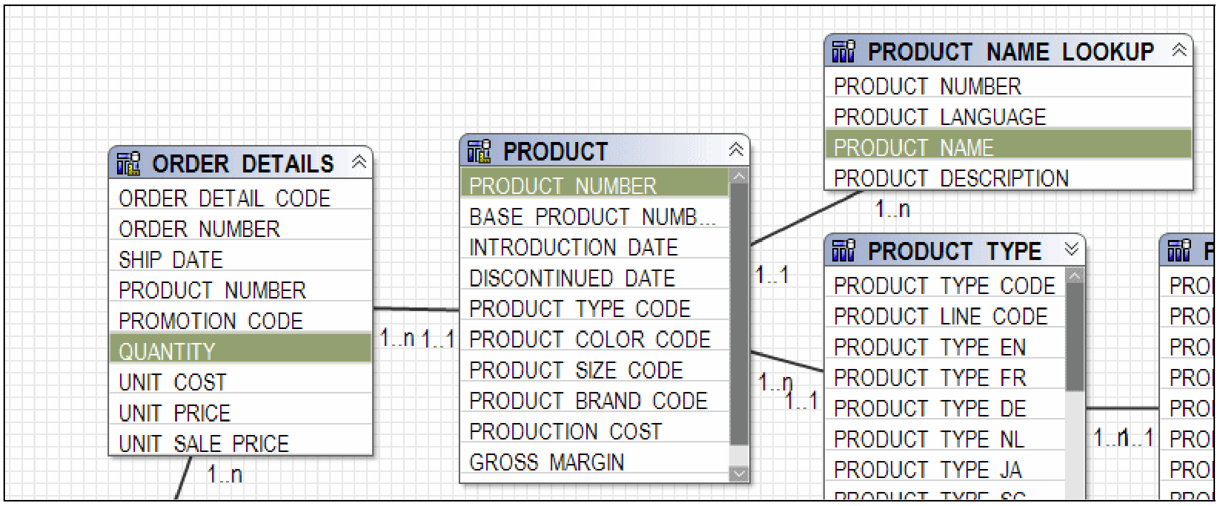
   Notice the two subqueries that are being joined with a full outer join. One retrieves all the multilingual product names for each product number, and the other, summarized order quantities for each product number. The two queries are then stitched together. This is an unwanted query split, since the intent in this query is to return one product name per product and show the quantities sold for that product.

5. Click **Close**.

Task 2. Examine the objects and their cardinality.

1. In the middle pane, click **Diagram**, and then double-click the **gosales** box to give it focus in the diagram.
   You will increase the level of detail in the diagram to include query items.
2. From the **Diagram** menu, click **Diagram settings**, select **Query Items**, and then click **OK**.
3. Using the **Auto Layout** dialog box, set the **horizontal** and **vertical distance** to **25** to increase the distance between objects.
4. In the diagram, locate **PRODUCT_NAME_LOOKUP**, **PRODUCT**, and **ORDER_DETAILS**.

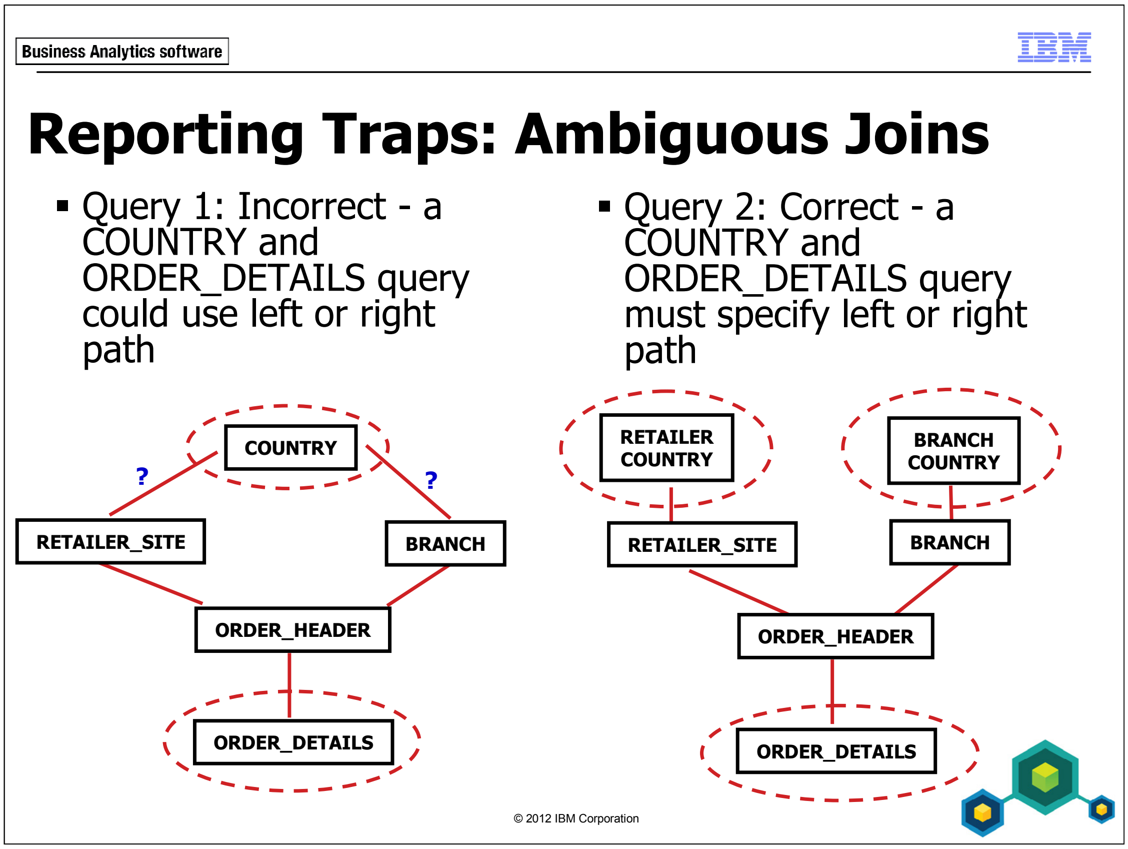
   The results appear similar to as follows:



In this scenario, PRODUCT is treated as a dimension (only 1..1 cardinalities attached), and ORDER_DETAILS and PRODUCT_NAME_LOOKUP are treated as facts (only 1..n cardinalities attached).

You will resolve the PRODUCT_NAME_LOOKUP issue later in the course by applying a filter and changing the cardinality.

Results:
You tested a query on total quantity sold by product to identify a possible reporting trap and found an unwanted query split.



Business Analytics software

Reporting Traps: Ambiguous Joins

- Query 1: Incorrect - a COUNTRY and ORDER_DETAILS query could use left or right path
- Query 2: Correct - a COUNTRY and ORDER_DETAILS query must specify left or right path

© 2012 IBM Corporation

Remove the risk of taking the wrong query path by removing the ambiguity: Creating aliases of COUNTRY with the appropriate relationships allows report authors to choose the desired path, in this case order details by retailer country or branch country.

The GO Operational model would actually use four versions of COUNTRY (and of SALES_REGION above it). The third version is for the hierarchy of SALES_REGION to COUNTRY to BRANCH to EMPLOYEE_HISTORY to EMPLOYEE. And the fourth version is for the hierarchy of SALES_REGION to COUNTRY to SALES_TARGET. We have just shown two of the four hierarchies in the slide to keep the diagram simple.

Demo 3: Identify Ambiguous Joins

Purpose:
Another common requirement for the business analysts will be to report on sales figures by retailer country. You will test such a query to identify a possible reporting trap.

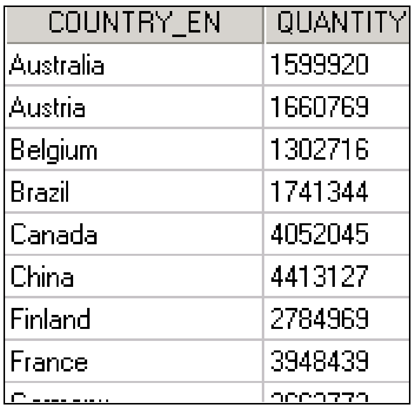
Component:	**Framework Manager**
Project:	**GO Operational**

Task 1. Create the report.

1. In **Framework Manager**, select and test the following query items with **Auto Sum** enabled:

Query Subject	Query Item
COUNTRY	**COUNTRY_EN**
ORDER_DETAILS	**QUANTITY**

The results appear as follows:

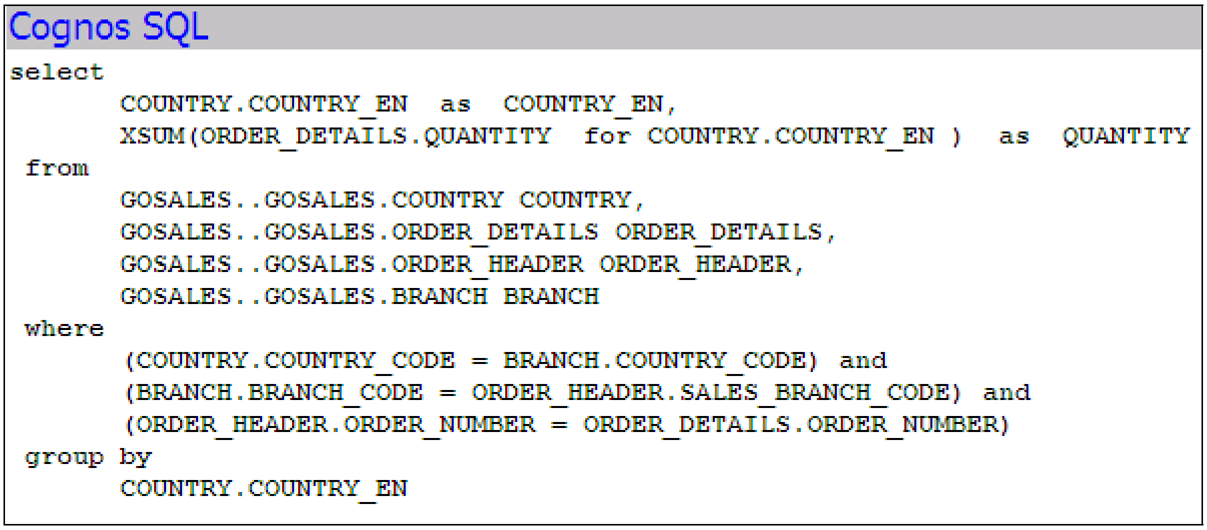


COUNTRY_EN	QUANTITY
Australia	1599920
Austria	1660769
Belgium	1302716
Brazil	1741344
Canada	4052045
China	4413127
Finland	2784969
France	3948439

The goal is to create a query for the number of units sold in each retailer's country. These results seem like they might be correct. But to be sure, you will look at the SQL generated for this query.

2. View the SQL on the **Query Information** tab.

   The results appear as follows:



Cognos SQL

```
select
       COUNTRY.COUNTRY_EN  as  COUNTRY_EN,
       XSUM(ORDER_DETAILS.QUANTITY  for COUNTRY.COUNTRY_EN )  as  QUANTITY
 from
       GOSALES..GOSALES.COUNTRY COUNTRY,
       GOSALES..GOSALES.ORDER_DETAILS ORDER_DETAILS,
       GOSALES..GOSALES.ORDER_HEADER ORDER_HEADER,
       GOSALES..GOSALES.BRANCH BRANCH
 where
       (COUNTRY.COUNTRY_CODE = BRANCH.COUNTRY_CODE) and
       (BRANCH.BRANCH_CODE = ORDER_HEADER.SALES_BRANCH_CODE) and
       (ORDER_HEADER.ORDER_NUMBER = ORDER_DETAILS.ORDER_NUMBER)
 group by
       COUNTRY.COUNTRY_EN
```

   The query did not take the path you wanted (through RETAILER_SITE). Instead, it took the path through BRANCH to ORDER_HEADER to ORDER_DETAILS. So the query set returned the number of units sold in each sales branch country.

3. Click **Close**.

Task 2. Examine the objects and their relationships.

You will use a feature called Context Explorer that helps you isolate modeling objects for closer examination and testing. It is a focused subset of the overall diagram. You will first change the default behavior of the Context Explorer to only show objects you select and not all related objects of the items you select.

1. From the **Diagram** menu, click **Diagram settings**.
2. Click the **Context Diagram** tab, select **Show only the selected objects**, and then click **OK**.

3. In the **Project Viewer**, select the following query subjects:

   COUNTRY

   BRANCH

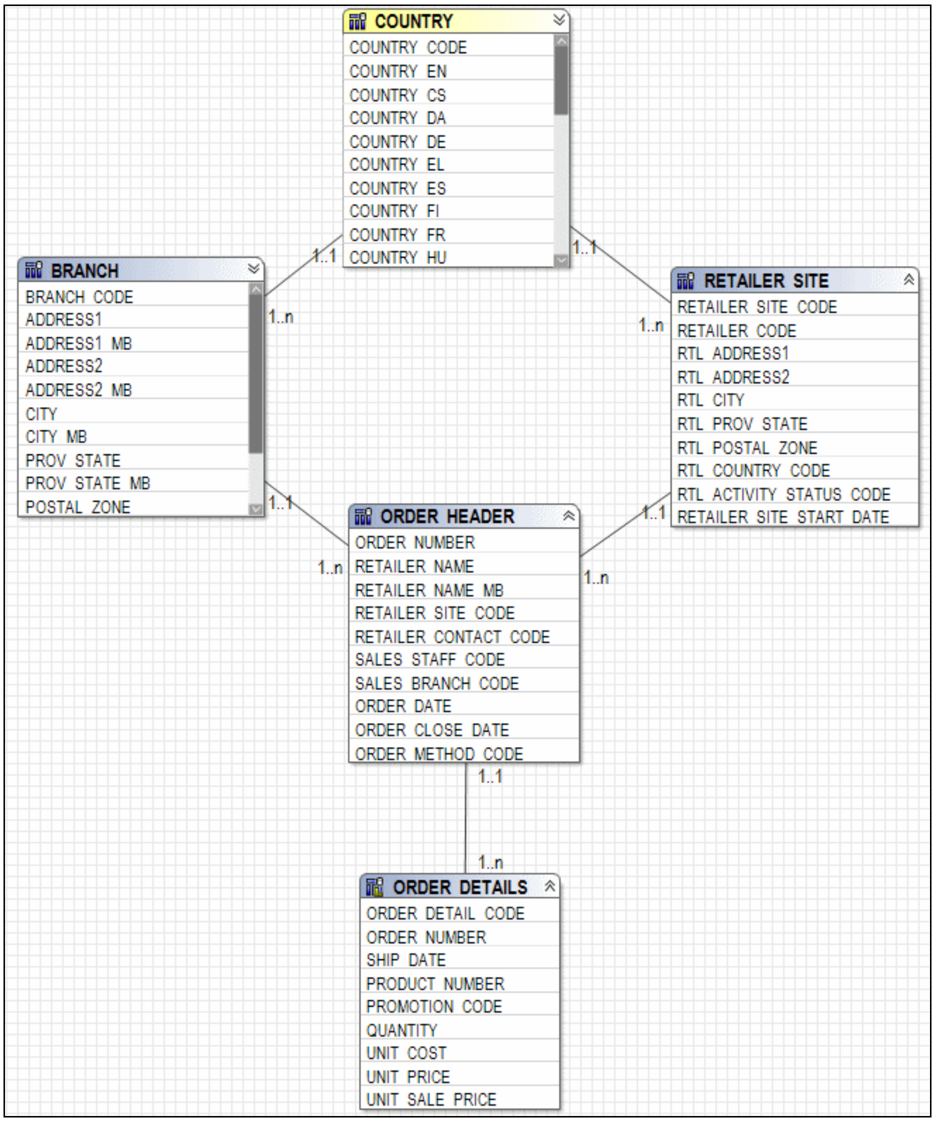
   RETAILER_SITE

   ORDER_HEADER

   ORDER_DETAILS

4. Right-click one of the selected objects, and then click **Launch Context Explorer**.

5. Arrange the diagram as shown below:

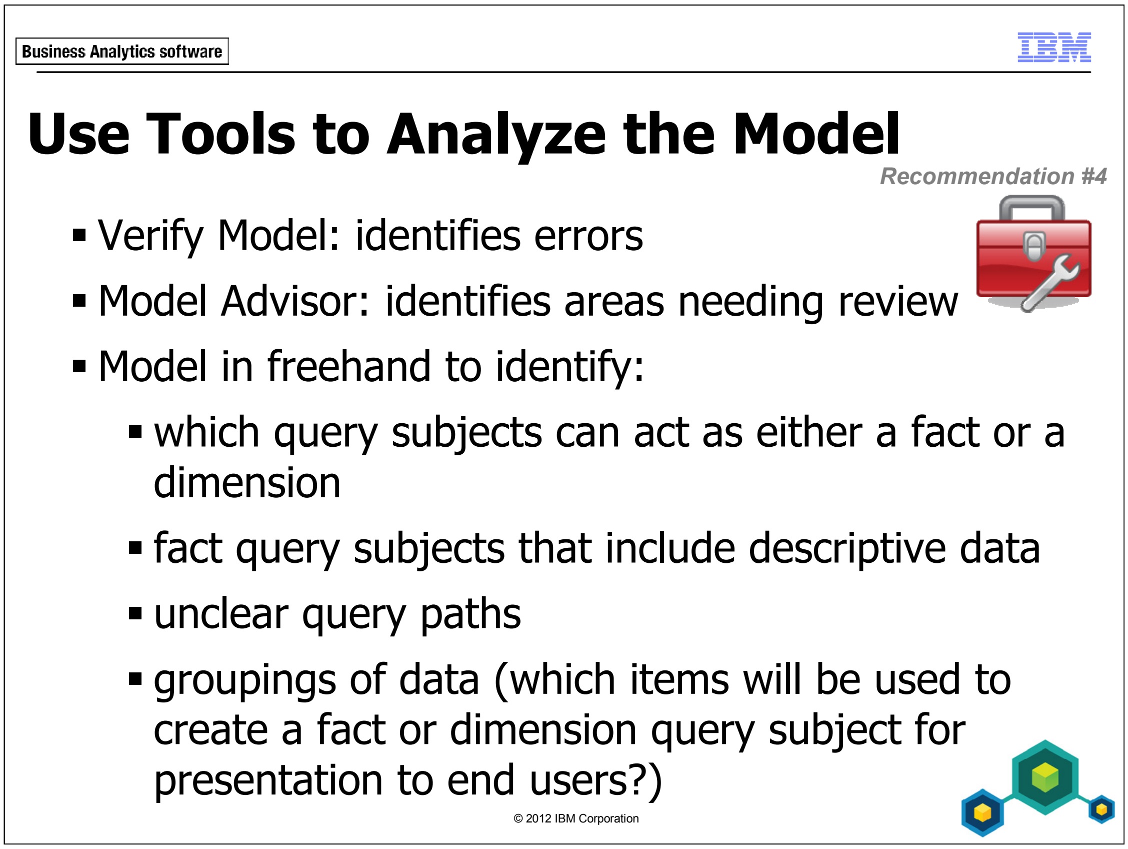


When a query can take more than one path of equal length and all query subjects are in a single namespace or folder, IBM Cognos chooses the relationship that comes first alphabetically. Using modeling techniques such as alias query subjects, you can avoid all ambiguity.

You will apply a solution in the next module.

6. Close the **Context Explorer**, and leave Framework Manager open for the next demo.

Results:
You tested a query on sales by country, intending the countries to be for retailers. Because there were multiple query paths that could be used, you discovered the potential for unpredictable results.



Business Analytics software

Use Tools to Analyze the Model

Recommendation #4

- Verify Model: identifies errors
- Model Advisor: identifies areas needing review
- Model in freehand to identify:
  - which query subjects can act as either a fact or a dimension
  - fact query subjects that include descriptive data
  - unclear query paths
  - groupings of data (which items will be used to create a fact or dimension query subject for presentation to end users?)

© 2012 IBM Corporation

- Verify Model: After you select the conditions and severity for which you want to test, Verify Model identifies the errors and offers automated or manual repairs.

- Model Advisor: After you select the conditions for which you want to test, Model Advisor gives a clear, graphical presentation that summarizes the design issues uncovered and offers links to documentation for evaluating and resolving the issues.

- Modeling in freehand: Use a diagram printout to identify areas for concern and how you will build your final model. Concentrate on modeling dimensions first, since they are usually easy to identify, more problematic, and can resolve most of the modeling design issues.

When modeling in freehand, it is important to have an understanding of the business processes that created the facts. You cannot create a correct model without interaction with functional users and data modelers. Your requirements and understanding of the data should be clear.

Demo 4: Use Tools to Analyze the Model

Purpose:
Prior to modeling the metadata, you will analyze the model for errors and design issues.

Component:	**Framework Manager**
Project:	**GO Operational**

Task 1. Analyze for errors.

1. In the **Project Viewer**, right-click the **gosales** namespace, and then click **Verify Selected Objects**.

   The Options tab lets you control two aspects of the verification process: the severity and the categories of verification. You will accept the defaults.

2. Click **Verify Model**.

   No issues have been detected in this simple model. Note the Repair Selected tool, and the ability to group reported issues by object id, severity, or message description.

3. Click **Close**.

   You can perform this verification process on any level under the root namespace of the project or verify the entire model from the Project menu.

   Verification is also performed by default when publishing a package. All contents of the package are verified unless you deselect the option in the publish wizard.

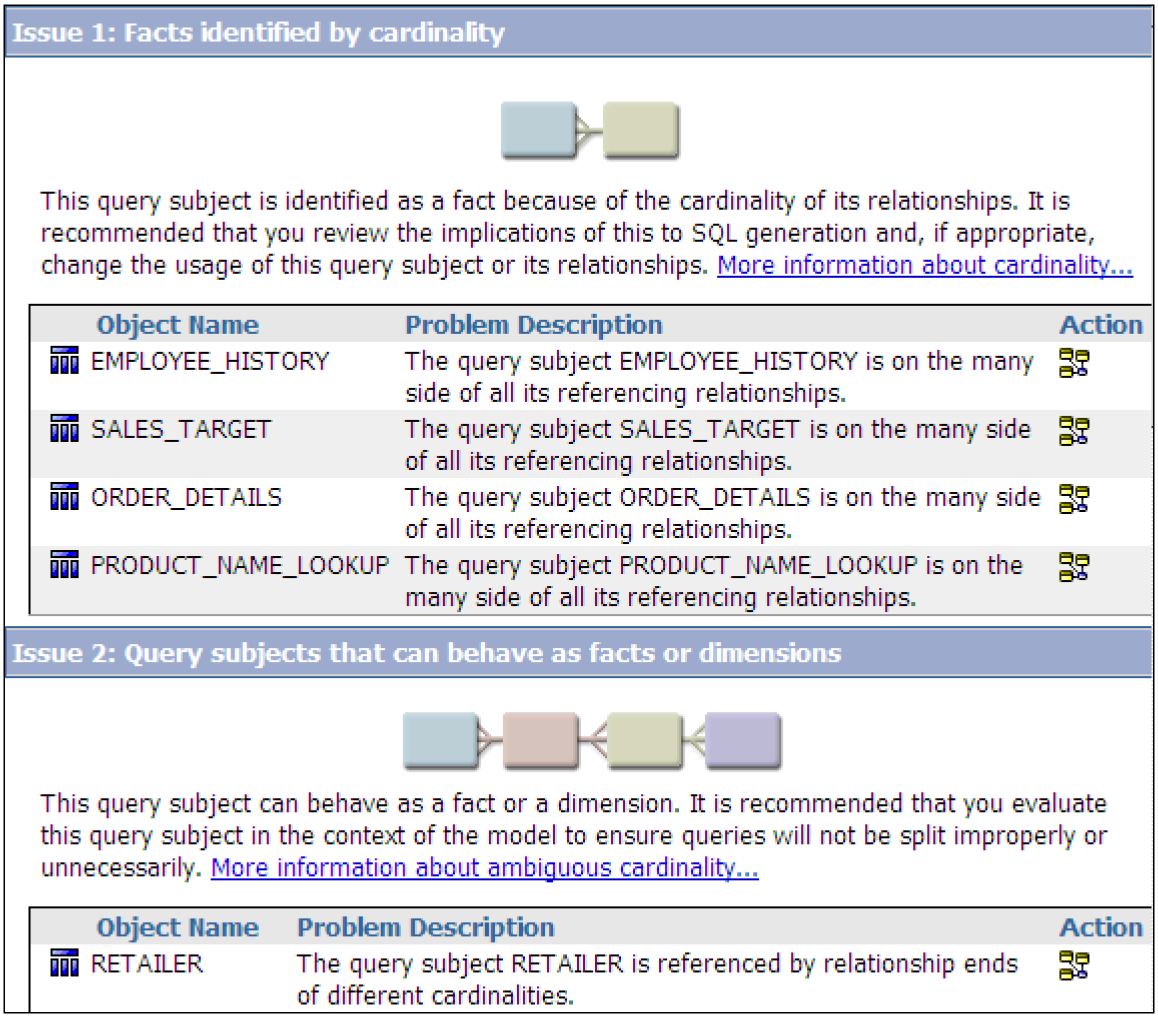
Task 2. Analyze for design issues.

1. In **Project Viewer**, right-click **gosales** and then click **Run Model Advisor**.

   The Options tab lets you control three forms of analysis: relationships, determinants, and miscellaneous checks. You will accept the defaults.

2. Click **Analyze**.

   The results appear similar to as follows:



Issue 1: Facts identified by cardinality

This query subject is identified as a fact because of the cardinality of its relationships. It is recommended that you review the implications of this to SQL generation and, if appropriate, change the usage of this query subject or its relationships. More information about cardinality...

Object Name	Problem Description	Action
EMPLOYEE_HISTORY	The query subject EMPLOYEE_HISTORY is on the many side of all its referencing relationships.	
SALES_TARGET	The query subject SALES_TARGET is on the many side of all its referencing relationships.	
ORDER_DETAILS	The query subject ORDER_DETAILS is on the many side of all its referencing relationships.	
PRODUCT_NAME_LOOKUP	The query subject PRODUCT_NAME_LOOKUP is on the many side of all its referencing relationships.	

Issue 2: Query subjects that can behave as facts or dimensions

This query subject can behave as a fact or a dimension. It is recommended that you evaluate this query subject in the context of the model to ensure queries will not be split improperly or unnecessarily. More information about ambiguous cardinality...

Object Name	Problem Description	Action
RETAILER	The query subject RETAILER is referenced by relationship ends of different cardinalities.	

Task 2, Step 2: You may wish to use the "More information" links for any of these issues to show the students the wealth of helpful information available to them.

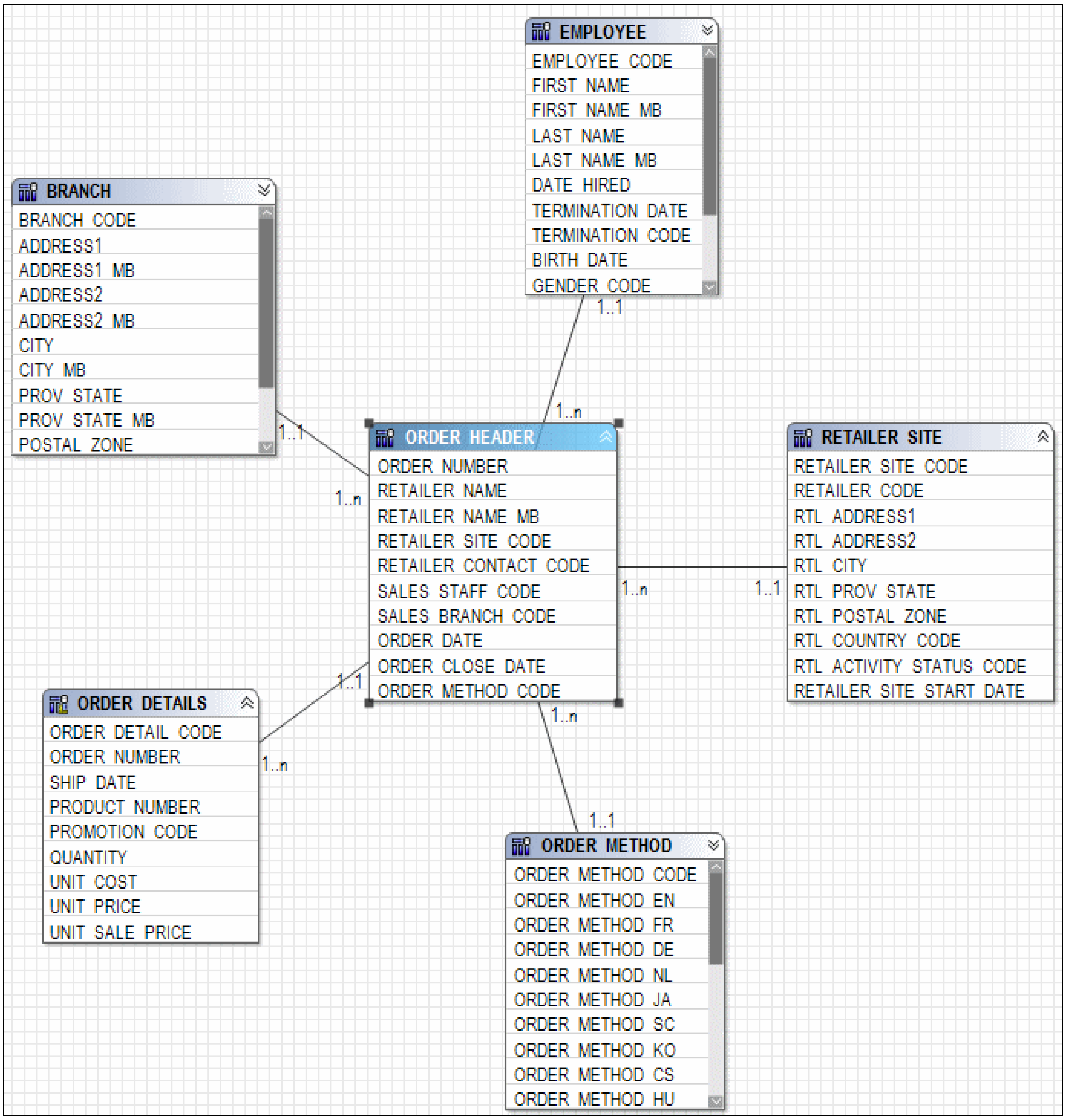
Issue 1 is "Facts identified by cardinality." You should confirm that the query subjects identified here are indeed facts. Both ORDER_DETAILS and SALES_TARGET are. But PRODUCT_NAME_LOOKUP is not a fact as defined by the data. EMPLOYEE_HISTORY is a factless fact, but is this how you want to use it for the reporting requirements, or should it be used as a lookup table for EMPLOYEE to return an employee's current job status? You'll look at these query subjects in more detail later in the course.

Issue 2 is "Query subjects that can behave as facts or dimensions." These items are on the 1..1 side of at least one relationship as well as being on the 1..n side of at least one relationship. These are also known as ambiguous query subjects. Before placing these in a package for reporting, you should investigate these objects to ensure that they do not have the potential of being used in an unpredictable manner. You have already seen that ORDER_HEADER may be used as a dimension in one query scenario and as a fact in another.

3. Under **Issue 2**, to the right of **ORDER_HEADER**, click **Launch Context Explorer** .

4. If necessary, to view all query subjects, maximize the window and click **Fit All**.

   The results appear similar as follows:



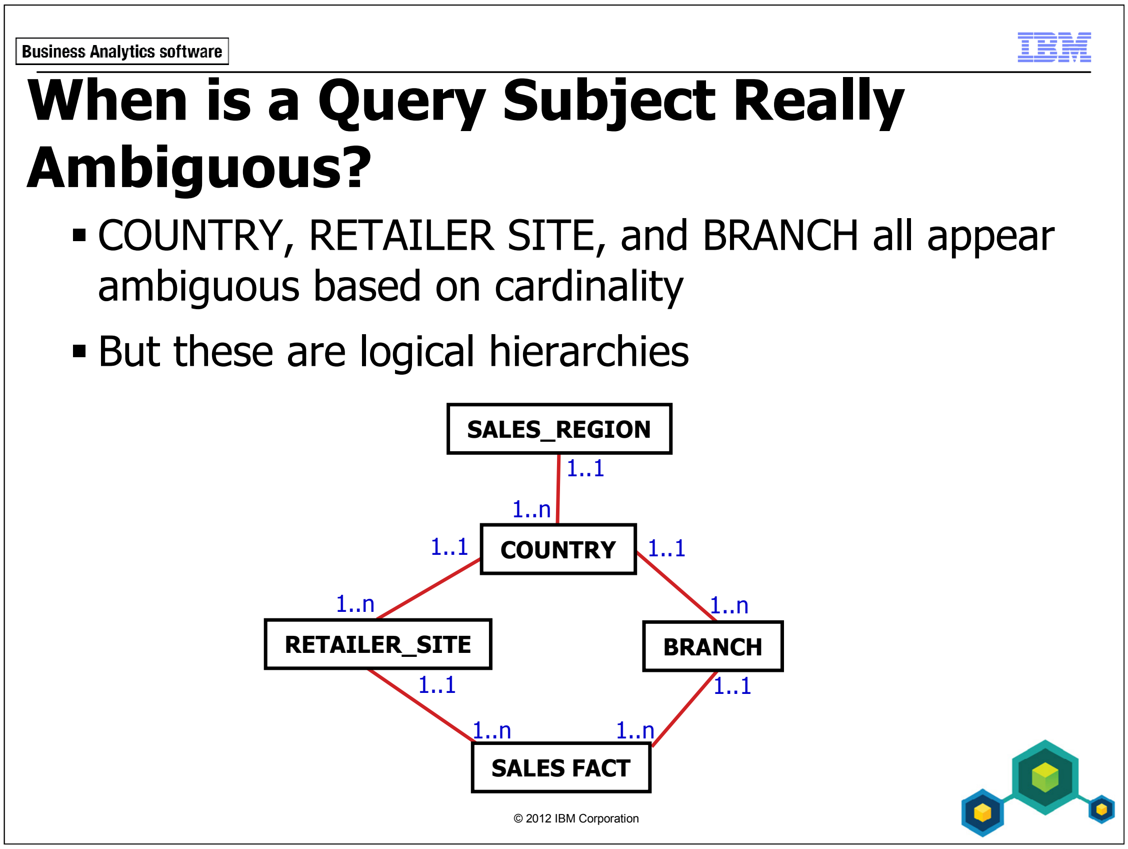
Here you can quickly see how ORDER_HEADER relates to other query subjects and where potential problems might occur.

5. Close the **Context Explorer** and continue with the Model Advisor issues.

   Issue 3 is "Query subjects with multiple relationships." This section identifies query subjects that may cause ambiguous query paths. Ambiguity will be removed throughout the modeling process.

6. Close the **Model Advisor** dialog box, and leave Framework Manager open for the workshop.

Results:
Prior to modeling the metadata, you analyzed the model for errors and design issues.



Business Analytics software

When is a Query Subject Really Ambiguous?

- COUNTRY, RETAILER SITE, and BRANCH all appear ambiguous based on cardinality
- But these are logical hierarchies

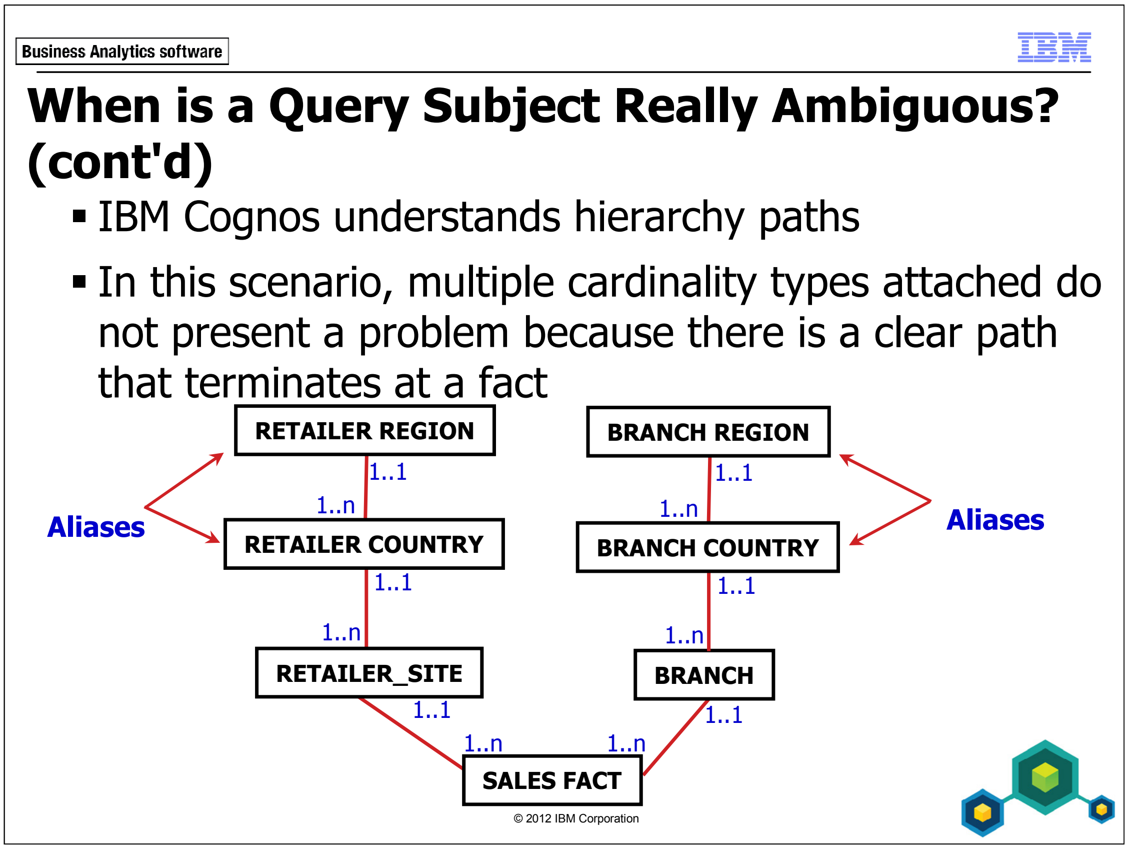
SALES_REGION
1..1
1..n
COUNTRY
1..1
1..1
1..n
1..n
RETAILER_SITE
BRANCH
1..1
1..1
1..n
1..n
SALES FACT

© 2012 IBM Corporation

It is important to understand when query subjects are really ambiguous and how to solve them.

There are logical hierarchies from SALES_REGION all the way to SALES FACT, but there is an ambiguous query path.

In this scenario, multiple cardinality types attached to the query subjects is not an issue, but a clear query path is.



Business Analytics software

When is a Query Subject Really Ambiguous? (cont'd)

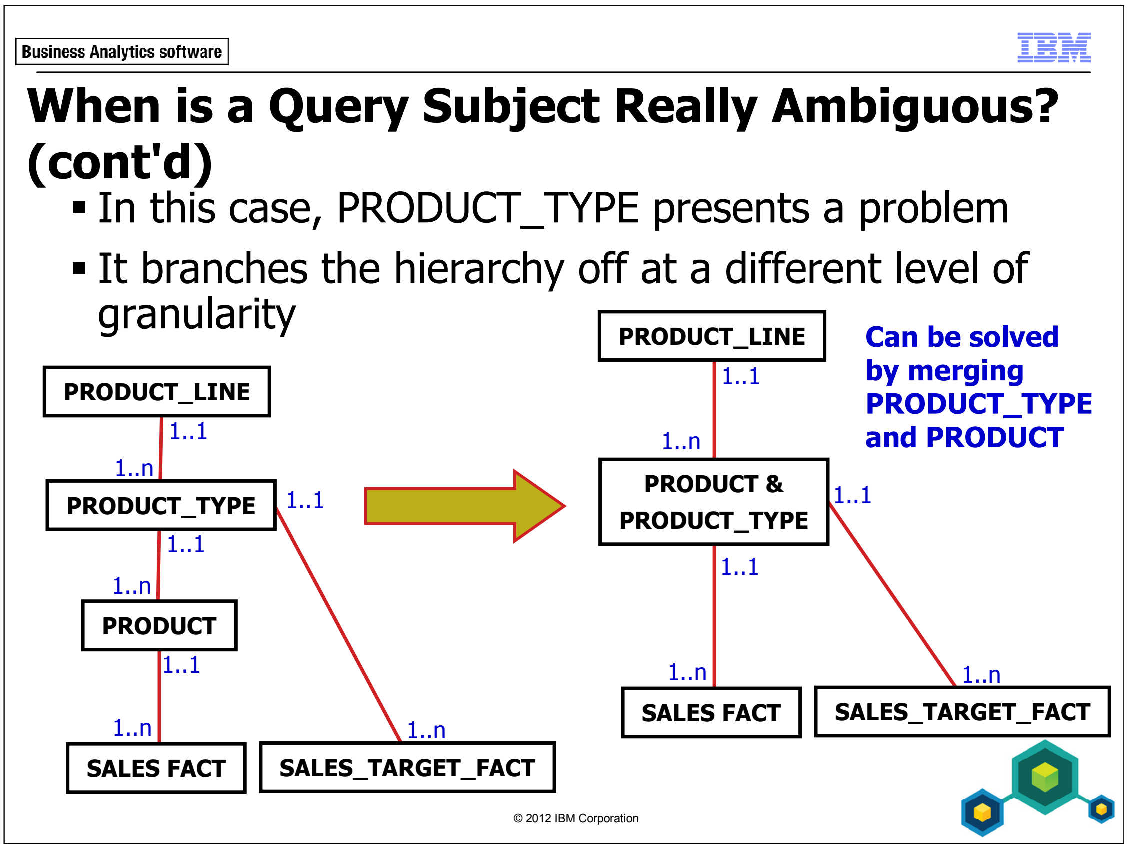
- IBM Cognos understands hierarchy paths
- In this scenario, multiple cardinality types attached do not present a problem because there is a clear path that terminates at a fact

© 2012 IBM Corporation

Using aliases you can remove the ambiguous query paths.

But the query subjects still have multiple cardinalities attached. This is OK because IBM Cognos can traverse the hierarchies and terminate at the actual fact, in this case SALES FACT.

This only becomes an issue if the hierarchy can branch off to different facts at different levels. You will see this example on the next page.



Business Analytics software

When is a Query Subject Really Ambiguous? (cont'd)

- In this case, PRODUCT_TYPE presents a problem
- It branches the hierarchy off at a different level of granularity

Can be solved by merging PRODUCT_TYPE and PRODUCT

© 2012 IBM Corporation

Remember querying PRODUCT_TYPE, PRODUCT, and SALES_TARGET_FACT caused PRODUCT to be treated as a fact query subject and generated an unwanted query split resulting in a full outer join? This is because IBM Cognos determined PRODUCT and SALES_TARGET_FACT were fact termination points and that PRODUCT_TYPE was a shared dimension between the two. Based on cardinality, this is the logical choice. But is it what you want?

After merging PRODUCT and PRODUCT_TYPE, there is a clear hierarchy path that terminates at the real fact query subjects, in this case, SALES FACT and SALES_TARGET_FACT.

Workshop 1: Model in Freehand to Identify Query Usage

This workshop is based on modeling Recommendation 4: Model in freehand to identify query usage.

You have just been asked to analyze a Framework Manager model based on The Sample Outdoors Company operational database, and to make recommendations on how to enhance the model so that authors will achieve predictable results.

To do so, analyze the following diagram to identify:

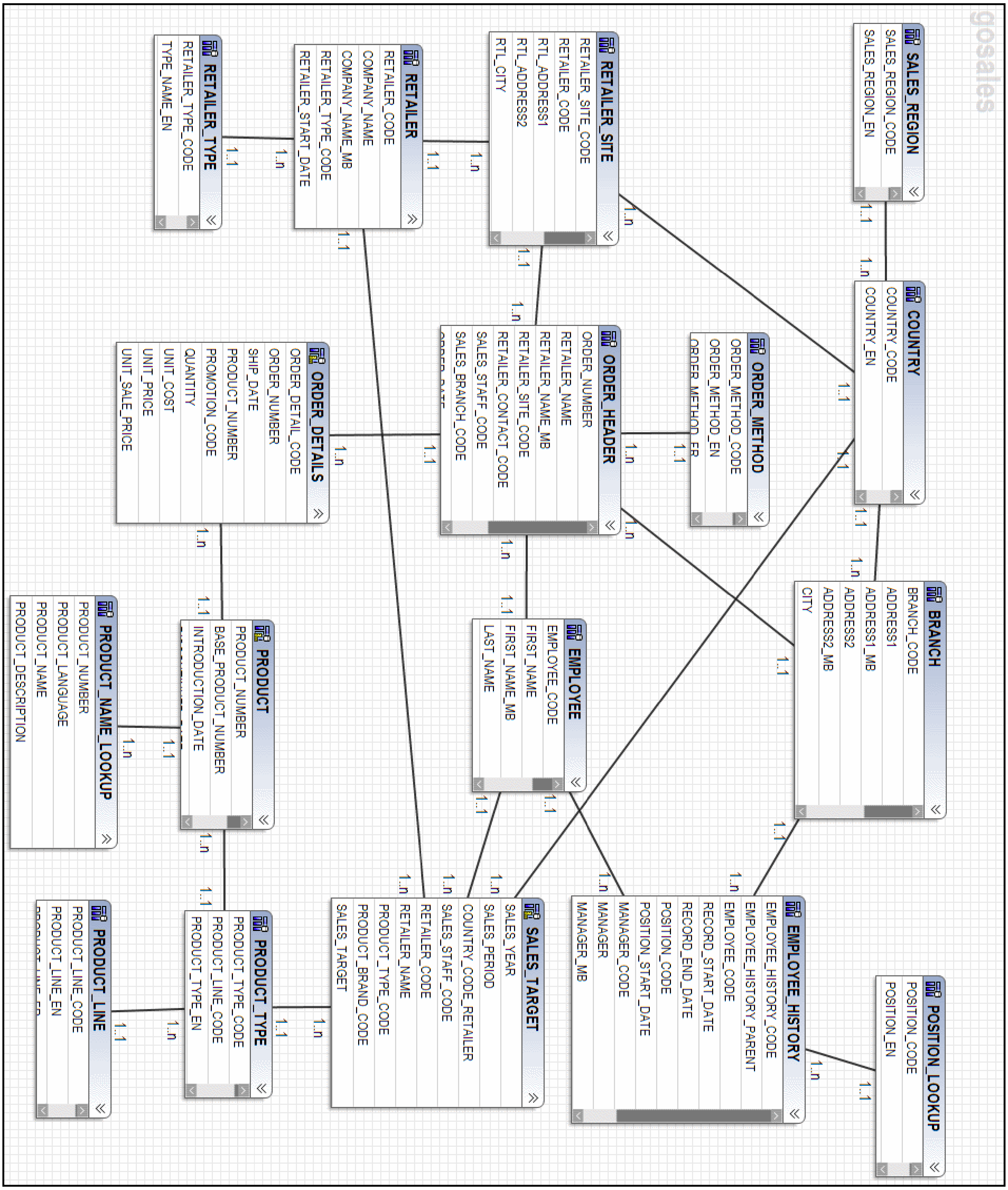
- All groupings of related query structures (for example, PRODUCT_LINE, PRODUCT_TYPE, and PRODUCT). These can be consolidated into a single author-friendly model query subject for presentation
- All facts (query subjects whose relationships are all on the 1..n side)
- Any ambiguous query subjects that can behave as either facts or dimensions (these can be resolved through a process of creating aliases to remove ambiguous query paths or by merging query subjects).

Use the three diagrams on the following pages to identify the items in each task above.

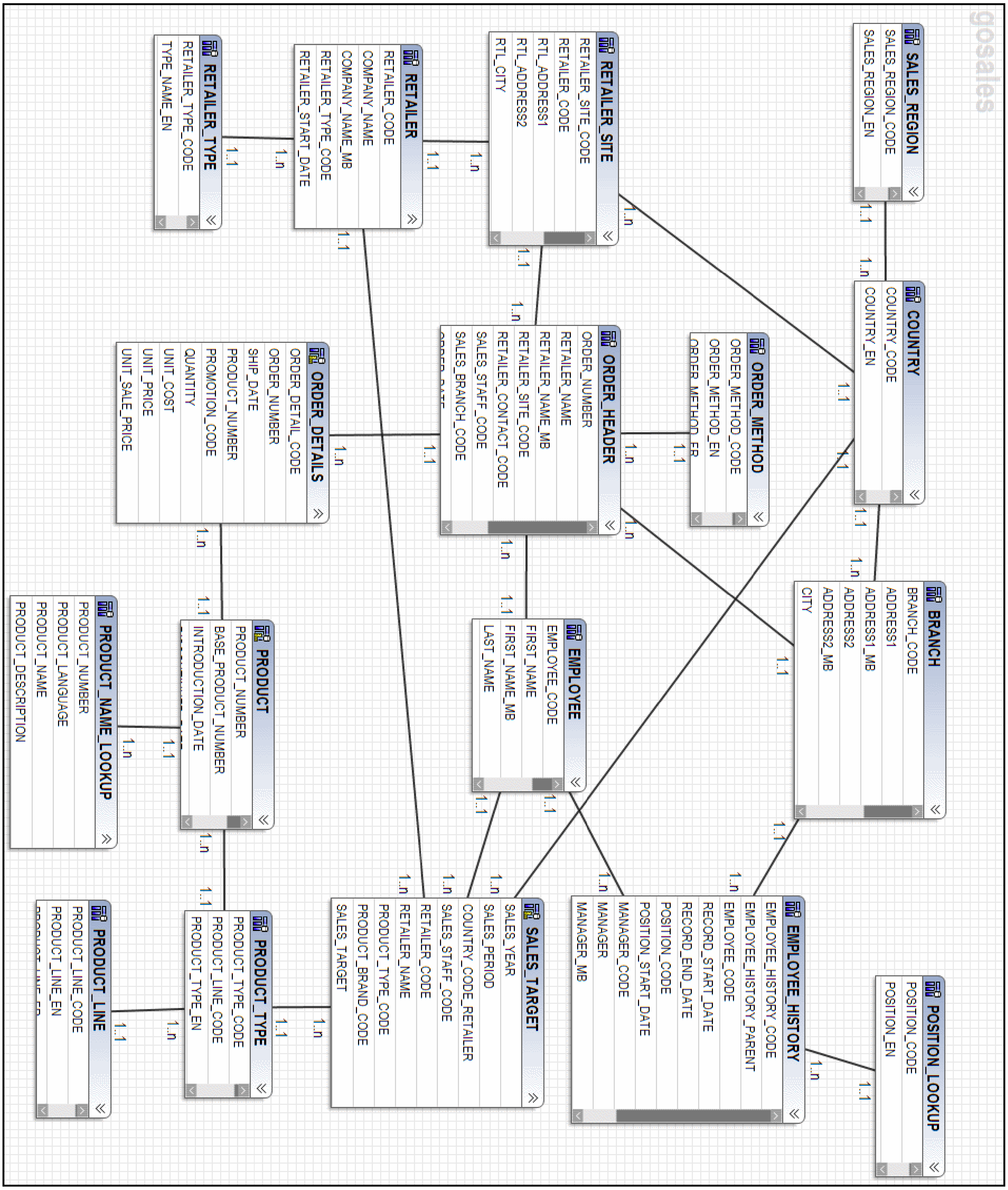
When done, close Framework Manager.

This is an extremely important workshop as it lays the foundation of what will be modeled throughout this course.

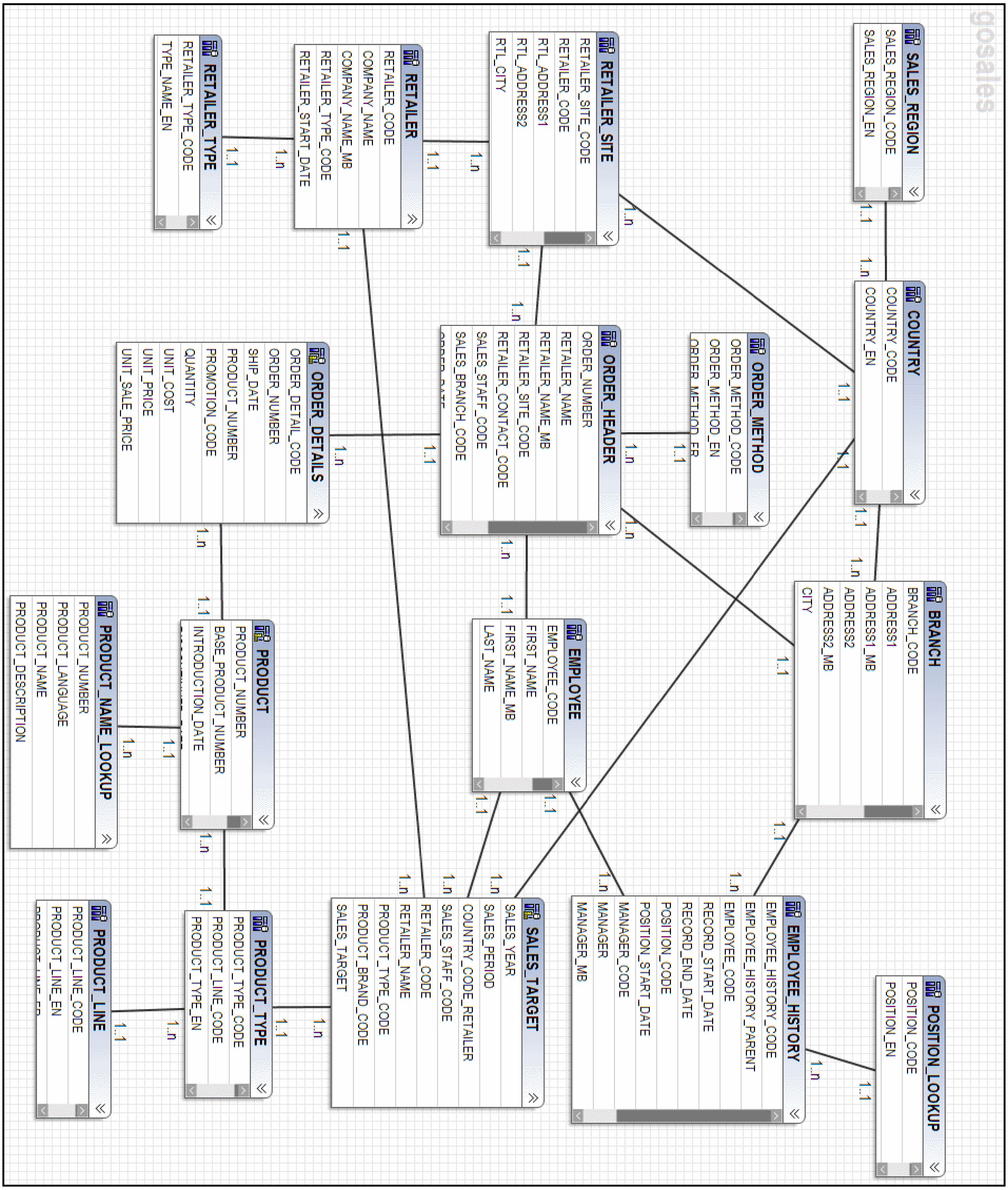
Workshop 1: Identify Groupings



Workshop 1: Identify Facts



Workshop 1: Identify Ambiguous Query Subjects



Workshop 1: Solution

These solutions will be explained in further detail and implemented throughout the next few modules.

Groupings:

The Identify Groupings page illustrates the first step in a star schema modeling solution, logically grouping related query structures:

- SALES_REGION, COUNTRY, BRANCH, EMPLOYEE_HISTORY, EMPLOYEE, and POSITION_LOOKUP to make up a **Staff by Location** dimension
- SALES_REGION, COUNTRY, RETAILER_SITE, RETAILER, and RETAILER_TYPE to make a **Retailer** dimension
- PRODUCT, PRODUCT_TYPE, PRODUCT_LINE and PRODUCT_NAME_LOOKUP to create a **Product** dimension

ORDER_METHOD is a stand-alone dimension and presents no issues.

There are also two other groupings, although not in the initial requirements, that may make sense. Sometimes during the modeling process the data reveals other alternatives that you can choose to take advantage. The ones found here are:

- SALES_REGION, COUNTRY and BRANCH to create a **Branch by Location** Dimension, which can then link to the grouping of ORDER_HEADER and ORDER_DETAILS in order to identify which branch a sale was made from
- SALES_REGION and COUNTRY to create a **Sales Target by Location** dimension

All of the above groupings will be used as dimensions to meet reporting requirements.

Facts:

ORDER_DETAILS is a fact as it only has 1..n cardinalities attached.

SALES_TARGET is a fact as it only has 1..n cardinalities attached.

EMPLOYEE_HISTORY is also a fact even though it contains no measures. However the requirements for this application are to report on employees' current positions. Later in the modeling process, a filter will be used to filter on the current position and therefore change the nature of the EMPLOYEE to EMPLOYEE_HISTORY relationship to 1..1 to 1..1.

PRODUCT_NAME_LOOKUP is identified as a fact as it only has 1..n cardinalities attached. In the data, this is not a fact and will be corrected with a filter that will change the cardinality to 1..1.

Ambiguous Query Subjects:

COUNTRY is used in multiple query paths and is a good candidate for aliasing.

BRANCH is used in two query paths and is a good candidate for aliasing.

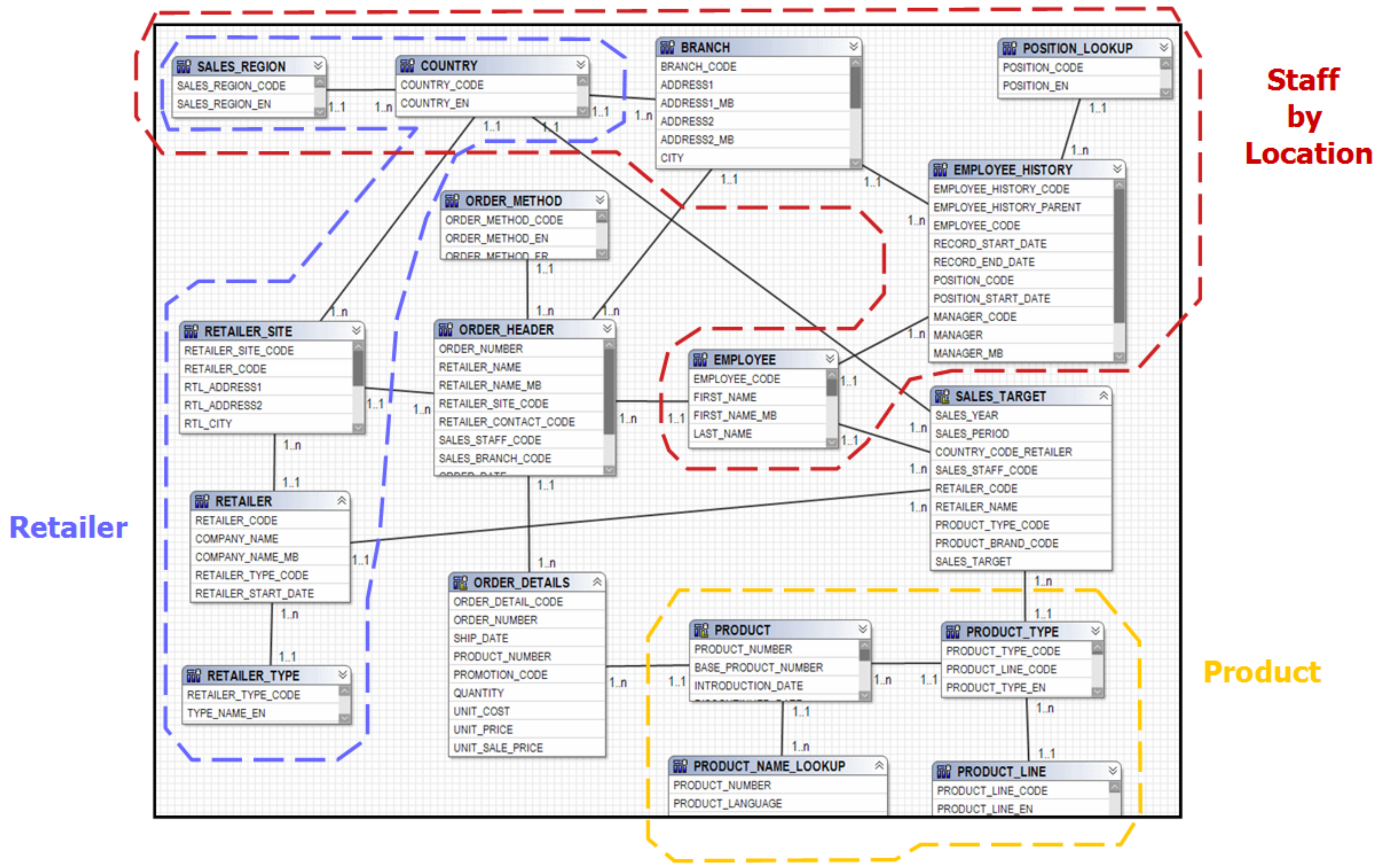
ORDER_HEADER is an ambiguous query subject because it has multiple cardinality types attached with multiple paths. It is not part of a hierarchy with one path.

- ORDER_HEADER is a query subject used as a bridge between most dimensions and the sales facts in ORDER_DETAILS. It contains keys allowing other dimensions to query order details.
- ORDER_HEADER is a good candidate for merging with ORDER_DETAILS to create a new fact query subject. Once these two query subjects are merged, the new query subject will have only 1..n cardinalities attached ensuring it will always be treated as a fact.

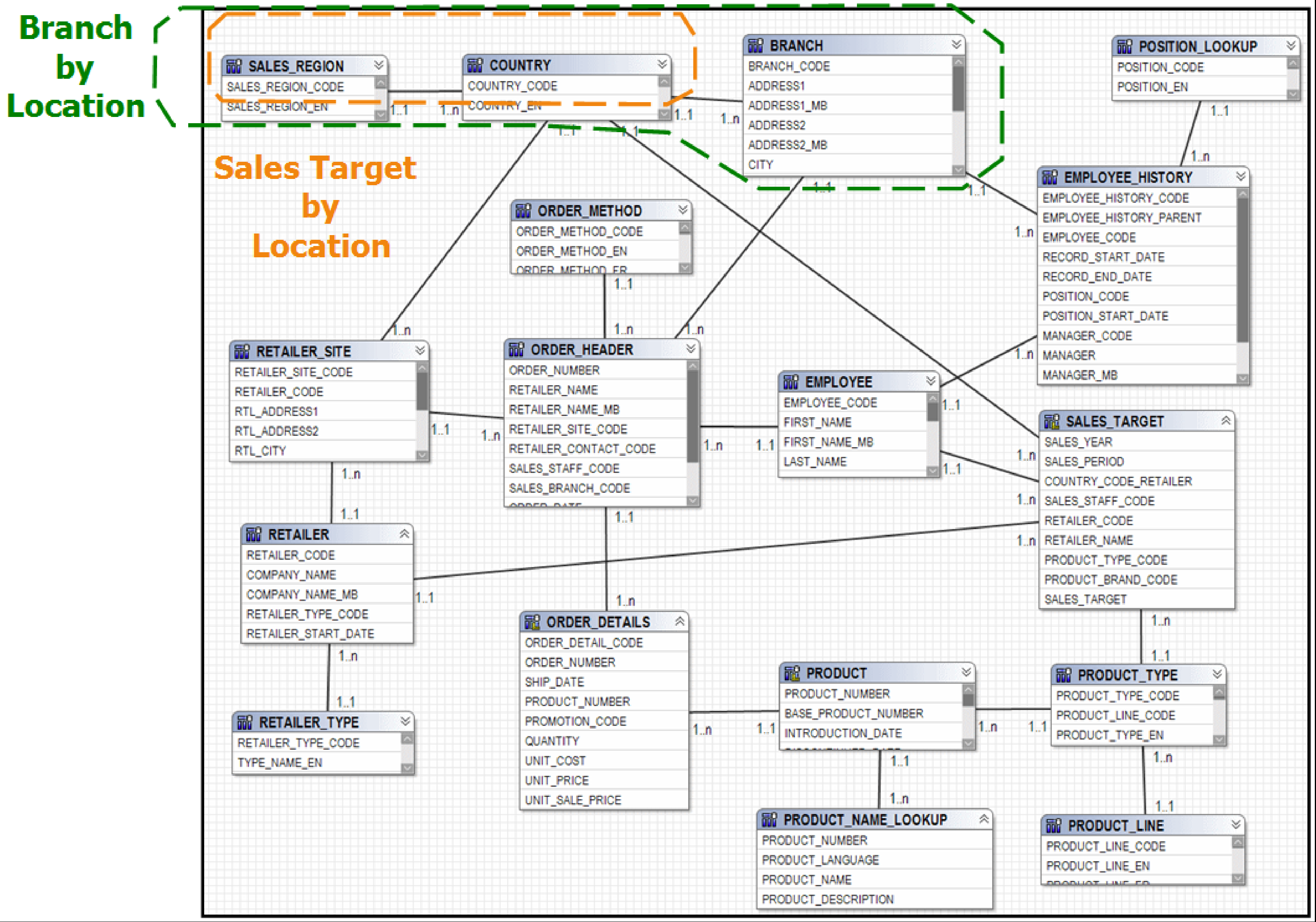
RETAILER has multiple cardinality types attached and branches the hierarchy off at a higher level of granularity. It is a good candidate for merging with RETAILER_SITE.

PRODUCT_TYPE has multiple cardinalities attached and branches off at a higher level of granularity. It is also a good candidate for merging with PRODUCT.

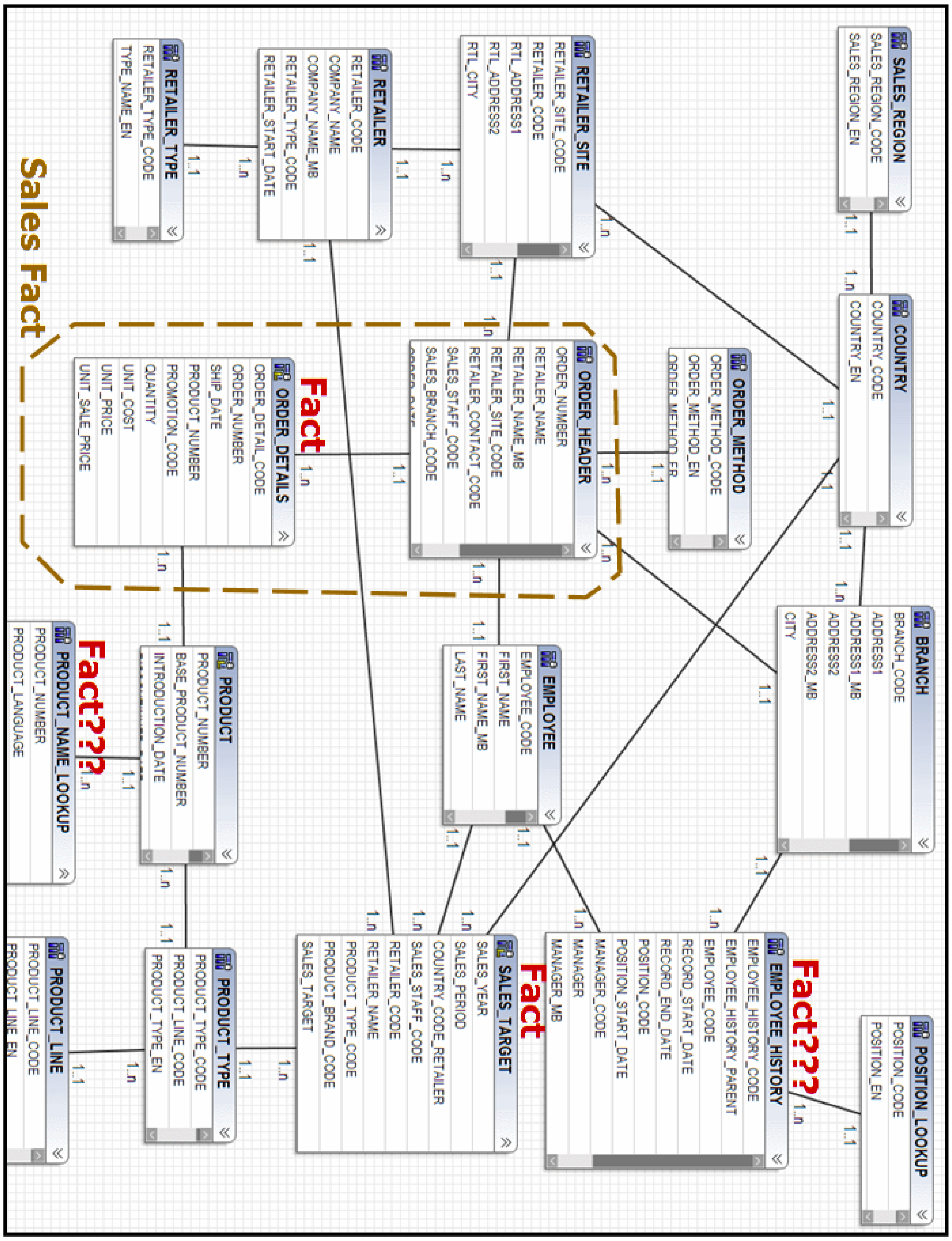
Workshop 1: Main Groupings



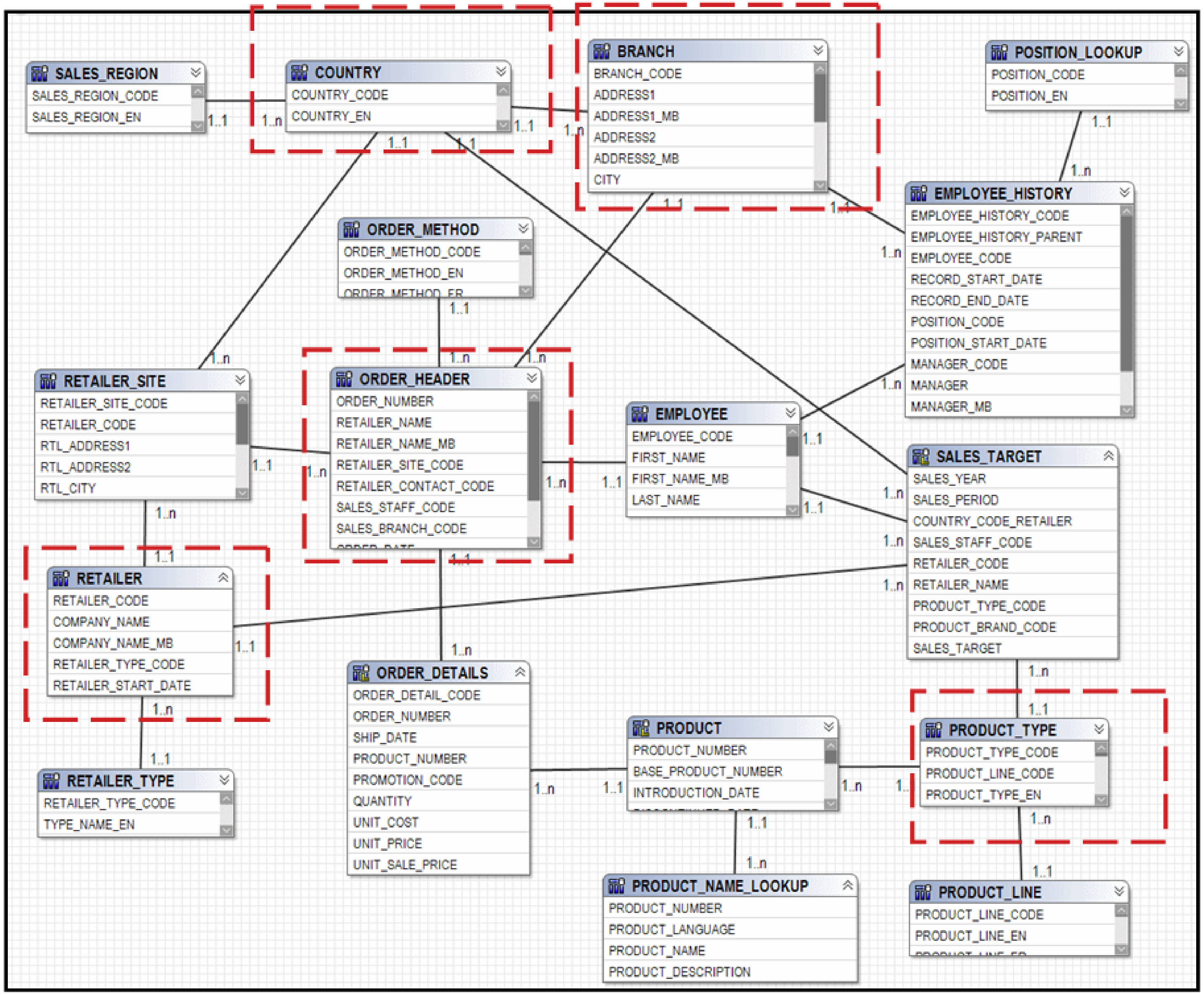
Workshop 1: Other Groupings

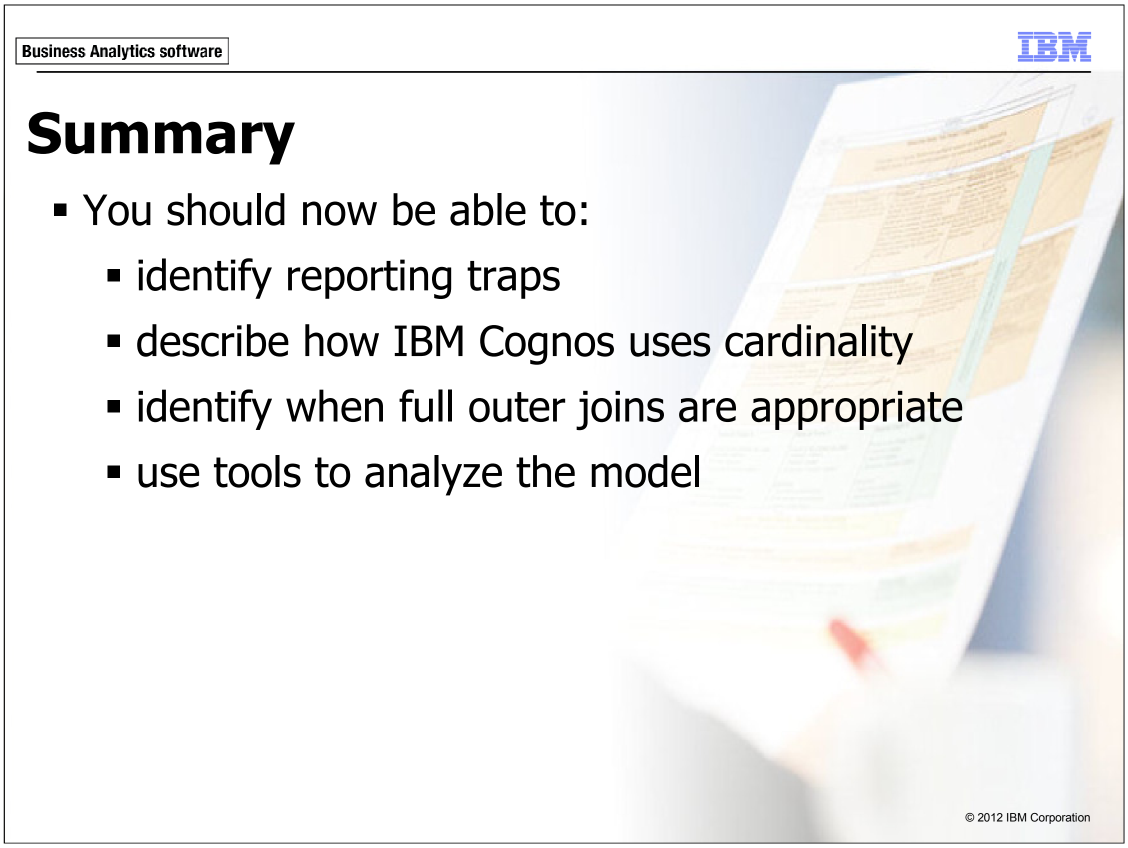


Workshop 1: Facts



Workshop 1: Ambiguous Query Subjects



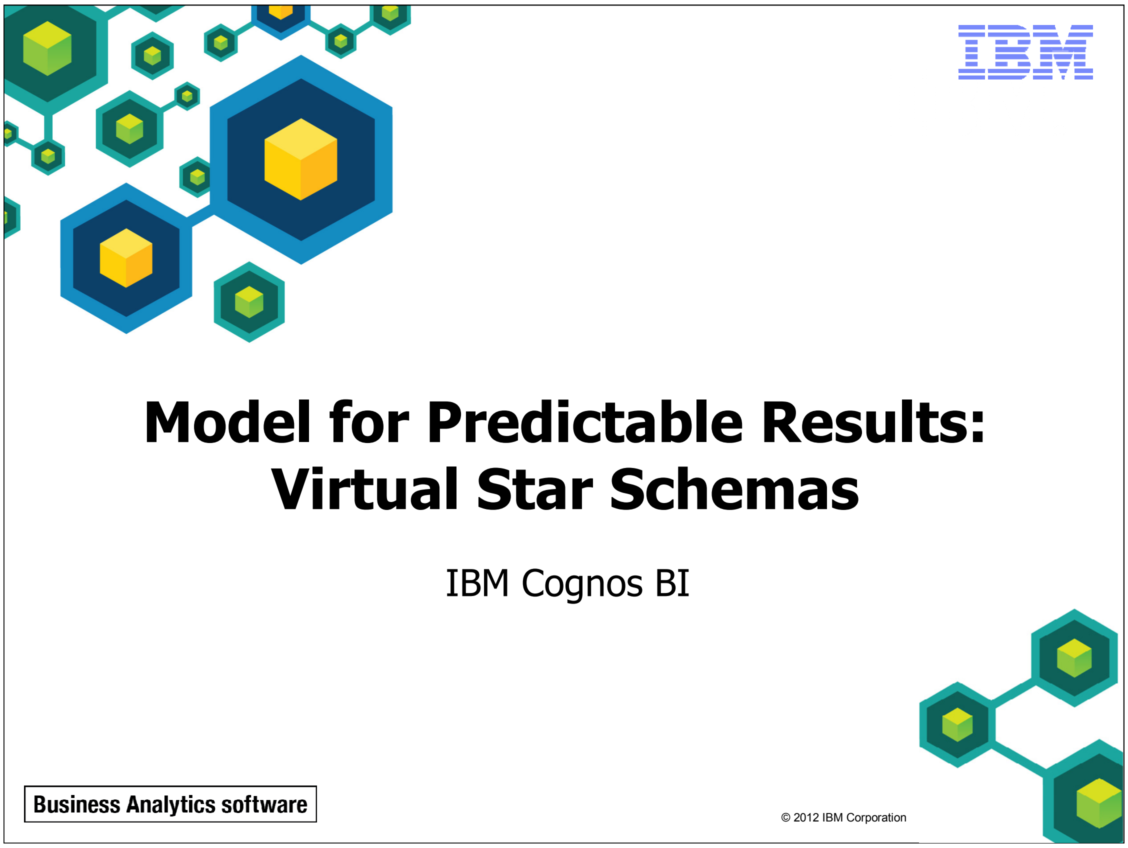


Business Analytics software

Summary

- You should now be able to:
  - identify reporting traps
  - describe how IBM Cognos uses cardinality
  - identify when full outer joins are appropriate
  - use tools to analyze the model

© 2012 IBM Corporation



IBM

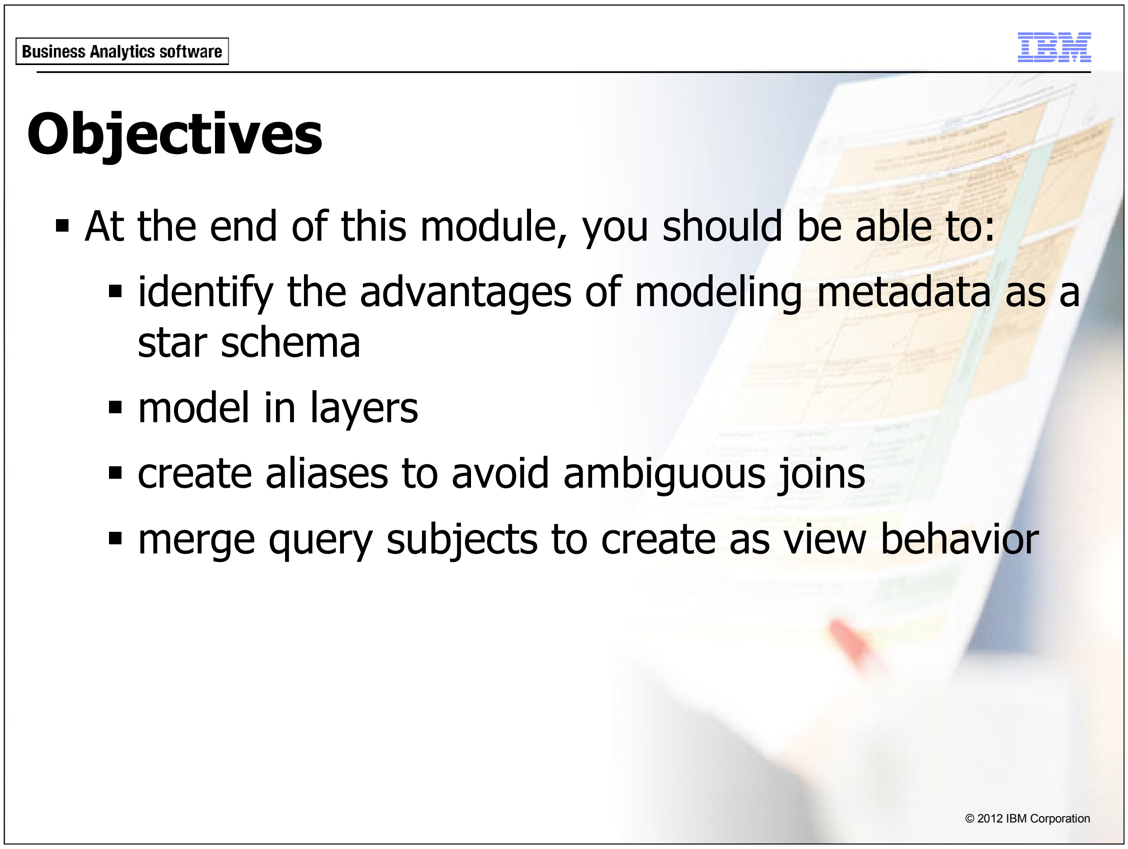
Model for Predictable Results: Virtual Star Schemas

IBM Cognos BI

Business Analytics software

Unless otherwise specified in demo or workshop steps, you will always log on to IBM Cognos BI in the Local LDAP namespace using the following credentials:

- User ID: admin
- Password: Education1



Business Analytics software

Objectives

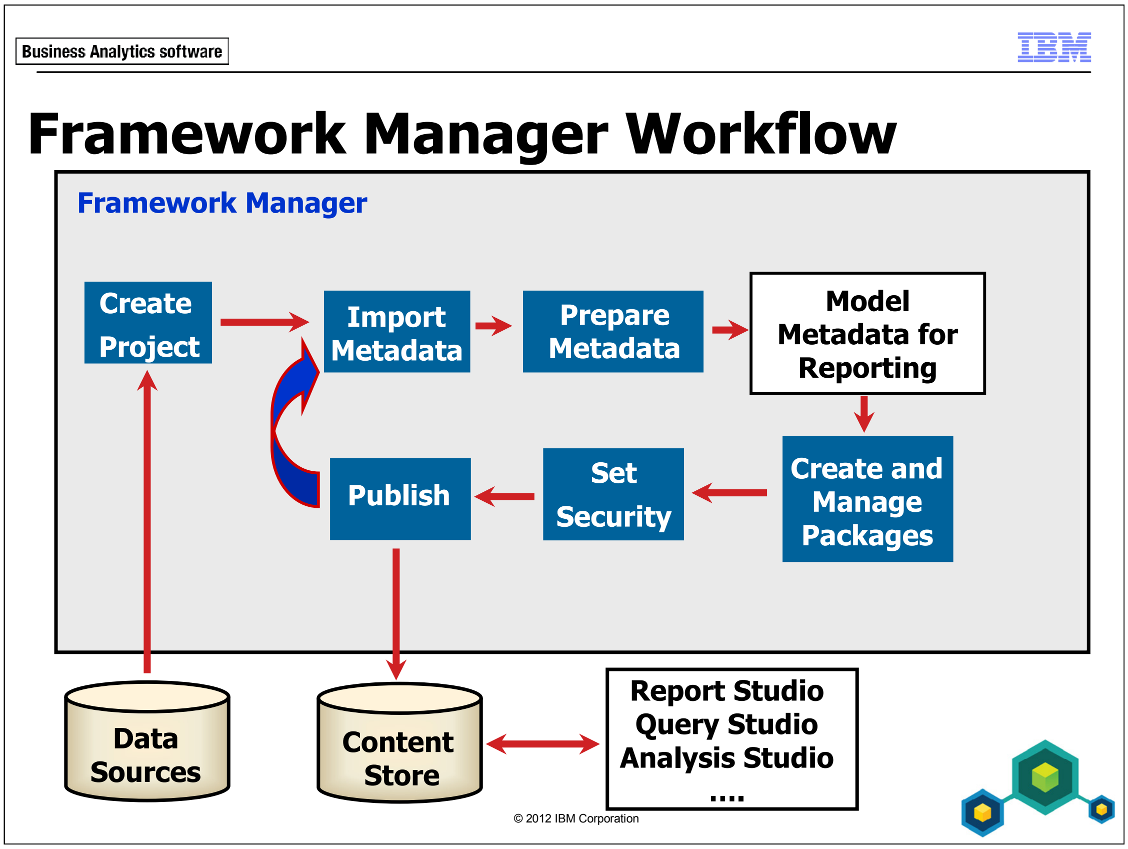
- At the end of this module, you should be able to:
  - identify the advantages of modeling metadata as a star schema
  - model in layers
  - create aliases to avoid ambiguous joins
  - merge query subjects to create as view behavior

© 2012 IBM Corporation

The techniques covered in this module are considered to be fundamental with respect to creating models for predictable results. Although some of the techniques appear advanced or complex, they are truly fundamental to successful models that are geared towards a broad audience.

There are instances were these techniques are not required and the modeling recommendations do not meet the needs of the modeler. Some applications may be very specific. Again, the modeling recommendations are geared towards providing predictable results to a broad audience.

Before reviewing this module, you should be familiar with IBM Cognos, IBM Cognos Connection, Query Studio and Report Studio. Suggested modules to reference:
• Overview of IBM Cognos
• Identify Common Data Structures
• Gather Requirements
• Create a Baseline Project
• Prepare Reusable Metadata
• Model for Predictable Results: Reporting Issues

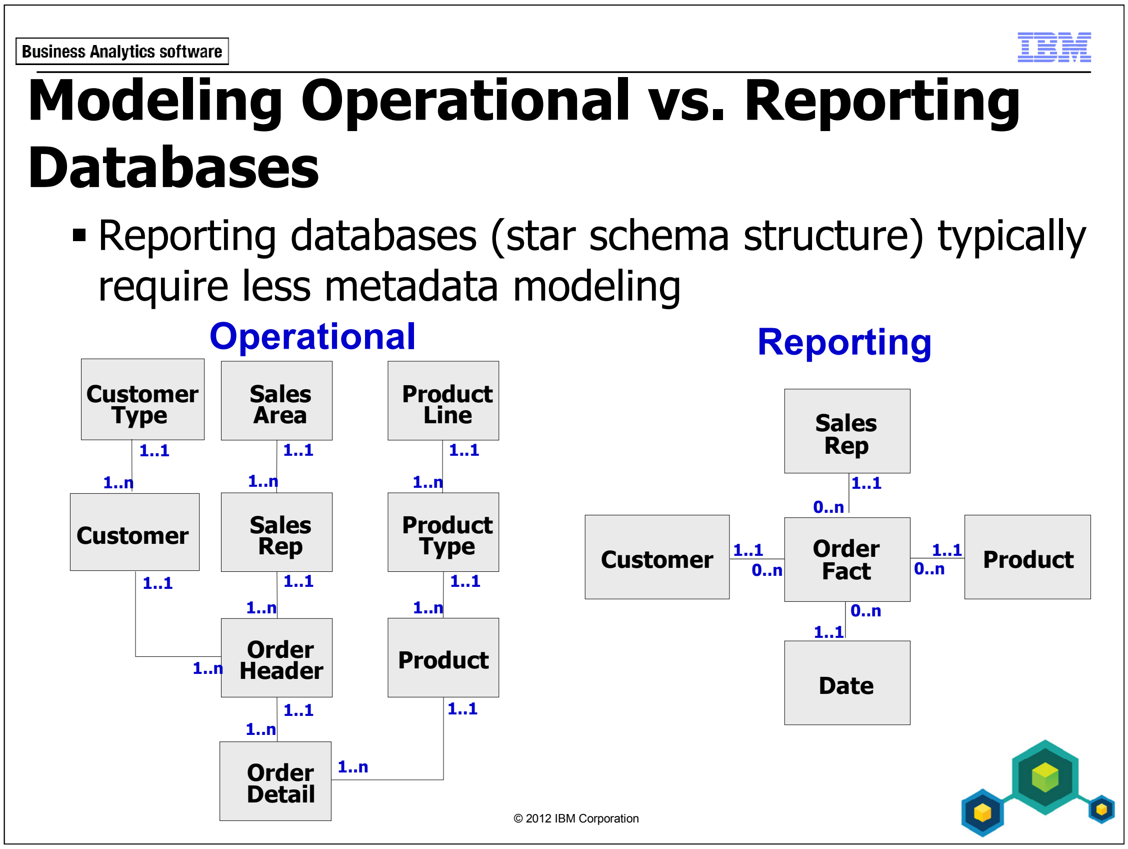


Business Analytics software

Framework Manager Workflow

© 2012 IBM Corporation

This module deals with modeling as a virtual star schema in order to ensure predictable results.



Business Analytics software

Modeling Operational vs. Reporting Databases

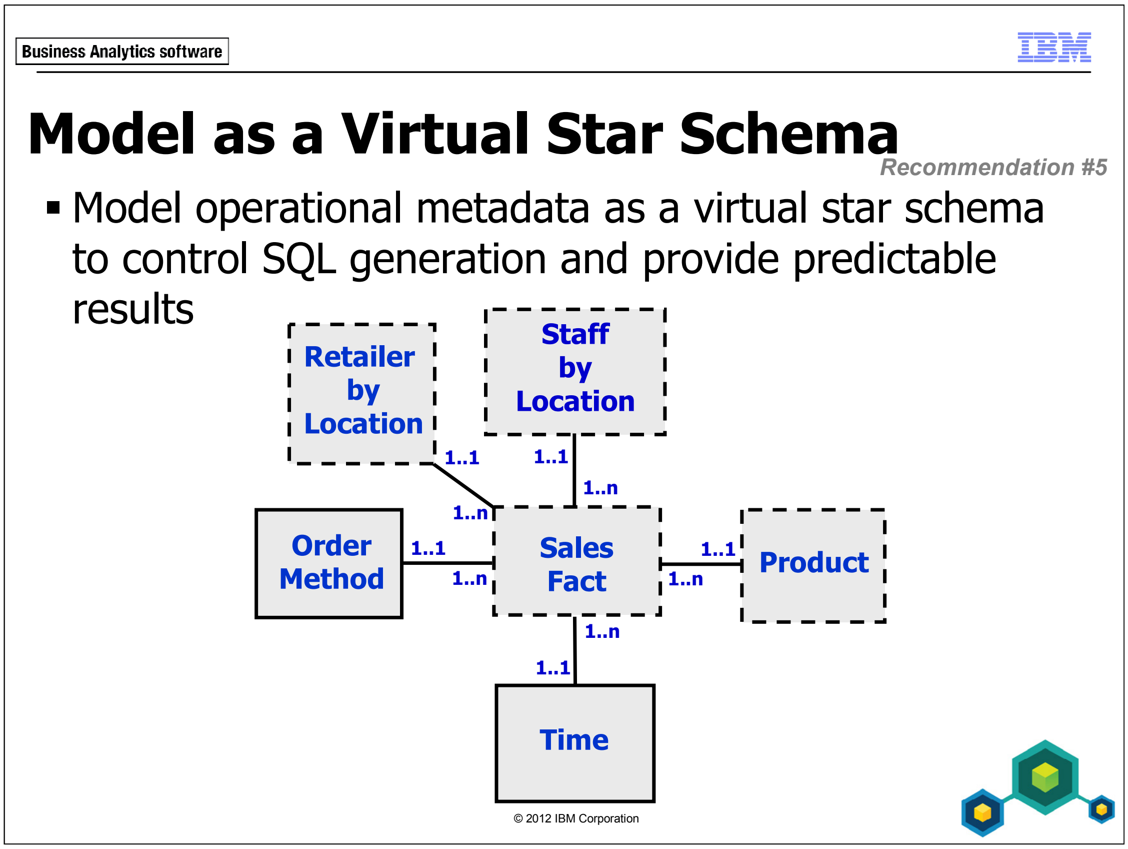
- Reporting databases (star schema structure) typically require less metadata modeling

© 2012 IBM Corporation

Operational databases typically require more of a metadata modeling effort in Framework Manager since they are confusing for users to understand and can present several reporting traps.

This course focuses on modeling an operational source because it illustrates more modeling scenarios that may apply to any database, both operational and reporting.

Star schema databases will likely require much less modeling since they are already designed for reporting. However, not all reporting database designs are immune to design issues and may require some of the very same techniques found in this course.

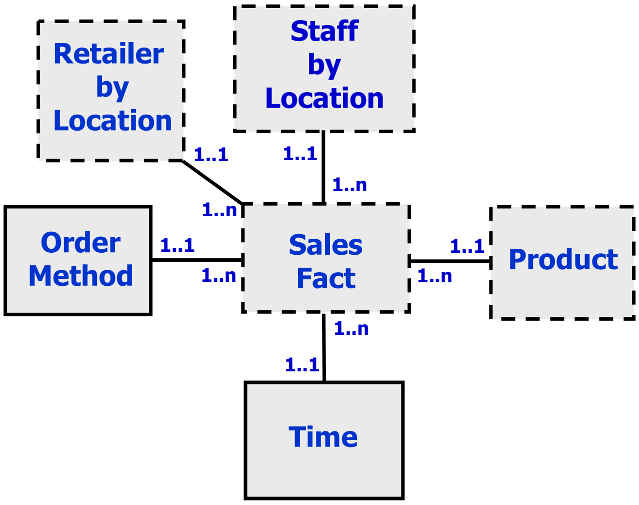


Business Analytics software

Model as a Virtual Star Schema

Recommendation #5

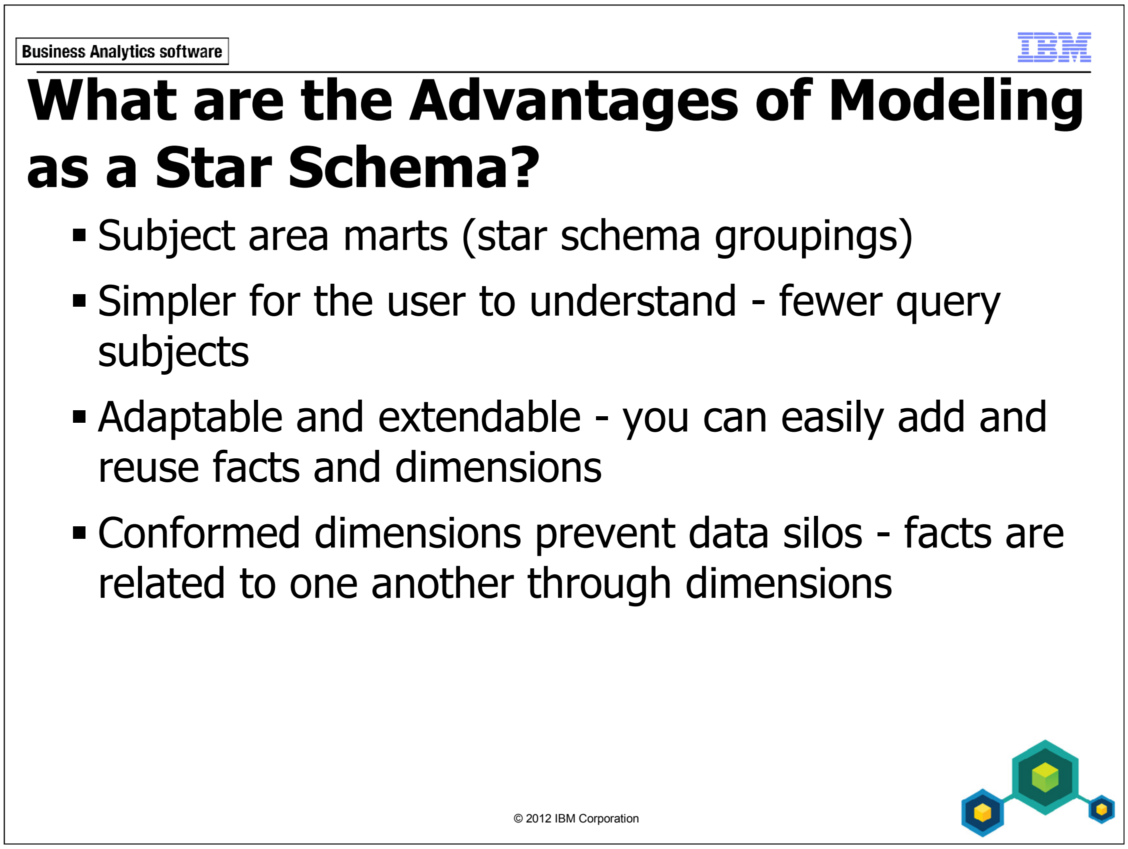
- Model operational metadata as a virtual star schema to control SQL generation and provide predictable results



© 2012 IBM Corporation

In the slide example, query subjects with dashed borders represent model query subjects that have been used to create views of the data at run time. These views organize and/or control the SQL generation, which helps to provide an easy to use model and predictable results.

When you model an operational or reporting data source as a star schema, you are not changing the underlying, data structure. You are creating a virtual star schema that allows IBM Cognos to generate the correct SQL for predictable results. Another benefit of virtual star schema models is that they are more flexible, allowing authors to answer a wider range of queries.



Business Analytics software

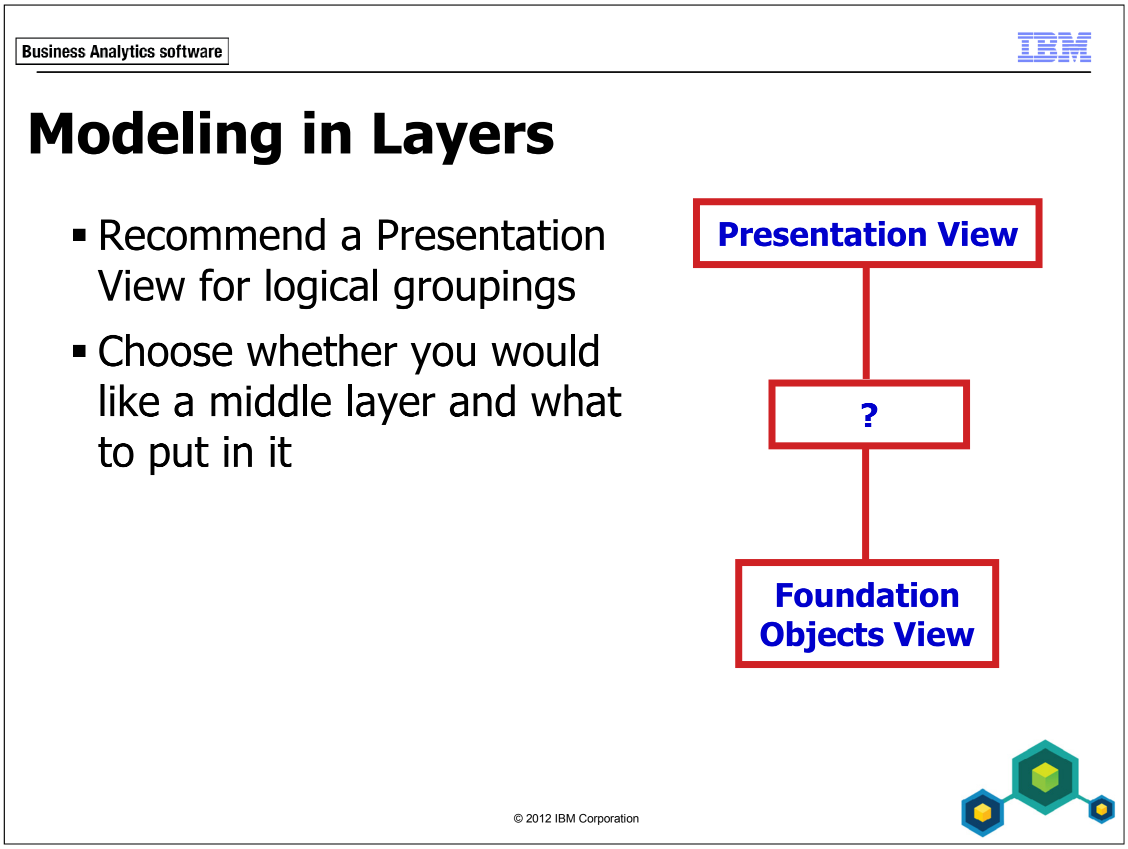
What are the Advantages of Modeling as a Star Schema?

- Subject area marts (star schema groupings)
- Simpler for the user to understand - fewer query subjects
- Adaptable and extendable - you can easily add and reuse facts and dimensions
- Conformed dimensions prevent data silos - facts are related to one another through dimensions

© 2012 IBM Corporation

The advantage of modeling dimensionally is that the end result is a business view that is organized by subject area. By presenting each subject area in a namespace with the relevant objects, it is easier for a business user to select the appropriate dimensions to go with a particular fact. Also, adding more facts and relevant dimensions can extend the model without affecting the existing metadata presentation. This allows you to easily add another subject area to the model.

Note that some people, with a relational database design background, challenge the idea of modeling operational data as a star schema because it generates lots of SQL at the reporting end. This is true, but it is an accepted trade-off to present authors with a structure that ensures that they retrieve the correct data. If you want simpler SQL in the reports without the risk of queries retrieving incorrect data (a risk encountered when you report directly on operational data structures), your only other solution is to physically store the data in a star schema.



Business Analytics software

Modeling in Layers

- Recommend a Presentation View for logical groupings
- Choose whether you would like a middle layer and what to put in it

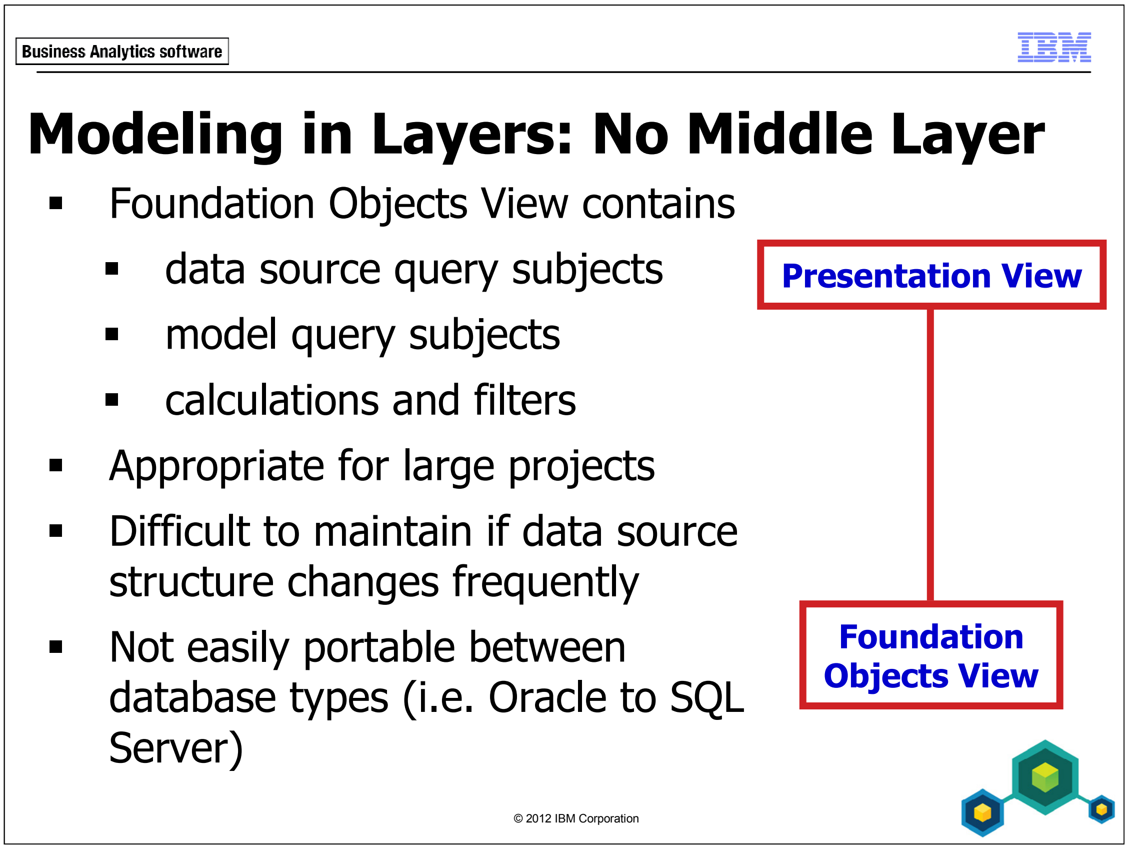
Presentation View
?
Foundation Objects View

© 2012 IBM Corporation

A Presentation View contains only the star schema groupings. This logically groups objects appropriate for the business and easily allows you to create separate packages for different reporting needs.

The decision to have another layer between the Foundation Objects View and the Presentation View involves several factors. For example, what is the size of the model? Do you need to reduce development time rather than ensure ease of maintenance later in the modeling cycle?

The next three slides investigate some approaches.



Business Analytics software

Modeling in Layers: No Middle Layer

- Foundation Objects View contains
  - data source query subjects
  - model query subjects
  - calculations and filters
- Appropriate for large projects
- Difficult to maintain if data source structure changes frequently
- Not easily portable between database types (i.e. Oracle to SQL Server)

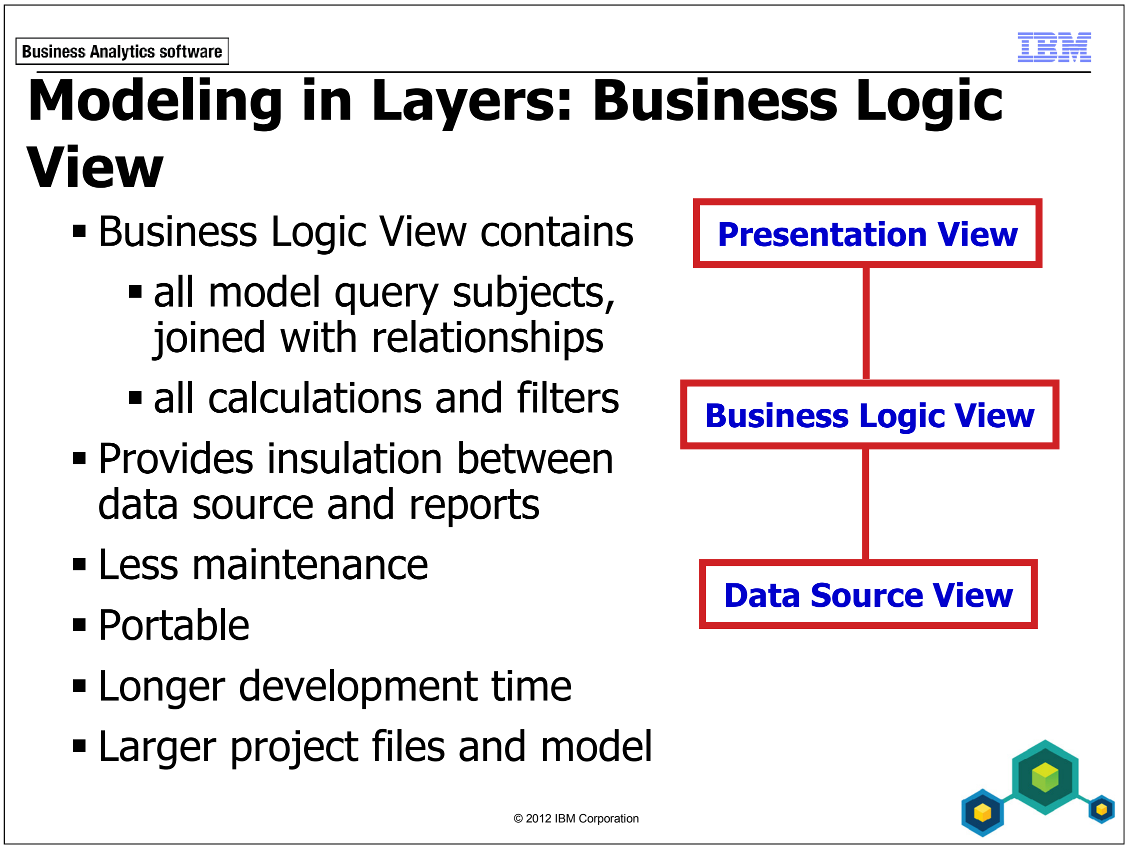
Presentation View
Foundation Objects View

This method requires the least duplication of query subjects, keeping the physical size of the project files to a minimum. It is best suited to large implementations or situations where a data warehouse has already been set up to accommodate the majority of the specialized business logic for reporting.

While it requires less development time, this method can require more maintenance when in production, since you will need to remodel to reflect any changes to the underlying data source objects, since the published objects are simply shortcuts to the data source query subjects. Therefore, if you expect ongoing changes to the underlying data structure, this may not be a suitable option.

You can view this approach in the GO Operational - (No Middle Layer) model located at C:\Edcognos\B5252\CBIFM-Files\Alternate_Models.

The model.xml file should not be larger than 50 MB. If the model is larger than this you may not be able to publish it.



Business Analytics software

Modeling in Layers: Business Logic View

- Business Logic View contains
  - all model query subjects, joined with relationships
  - all calculations and filters
- Provides insulation between data source and reports
- Less maintenance
- Portable
- Longer development time
- Larger project files and model

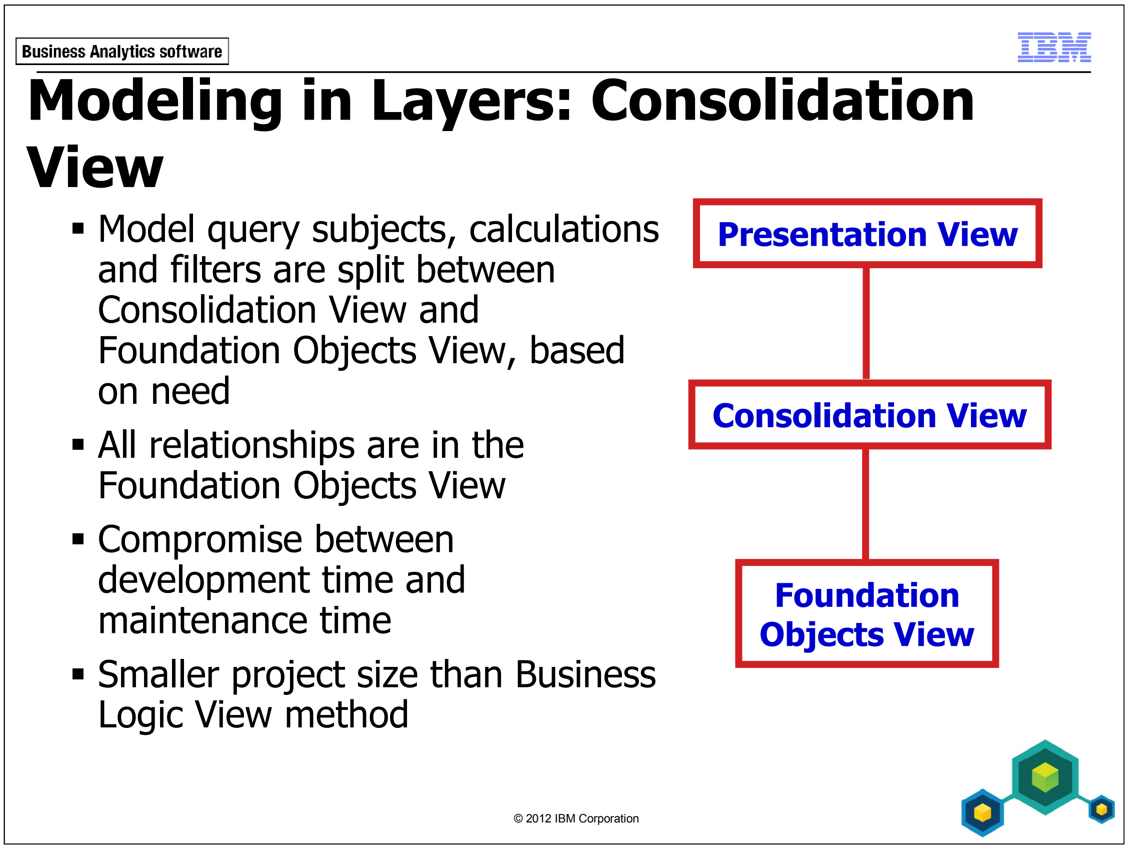
Presentation View
Business Logic View
Data Source View

© 2012 IBM Corporation

Using a business logic layer lets you set up the complex queries and reuse foundation layer objects in multiple locations. This provides insulation from the underlying data source for the reports. No work is required in the Data Source View since it is all done in the Business Logic View.

Creating the model query subjects (and rebuilding all their relationships) in the Business Logic View takes extra work, but it provides a layered structure for improved model maintenance, readability and portability. For example, to move the application from one database vendor to another, all you have to do is re-map the model query subjects in the Business Logic View to the tables in the new data source. There is no need to rebuild the reports, or remodel the metadata.

You can view this approach with the GO Operational - (Business Logic View) model located at C:\Edcognos\B5252\CBIFM-Files\Alternate_Models folder. Model query subjects in this view are also used to consolidate hierarchy items for presentation (no relationships). This can be seen in the folders of the Business Logic View (example: Product in the Product Info folder).



Business Analytics software

Modeling in Layers: Consolidation View

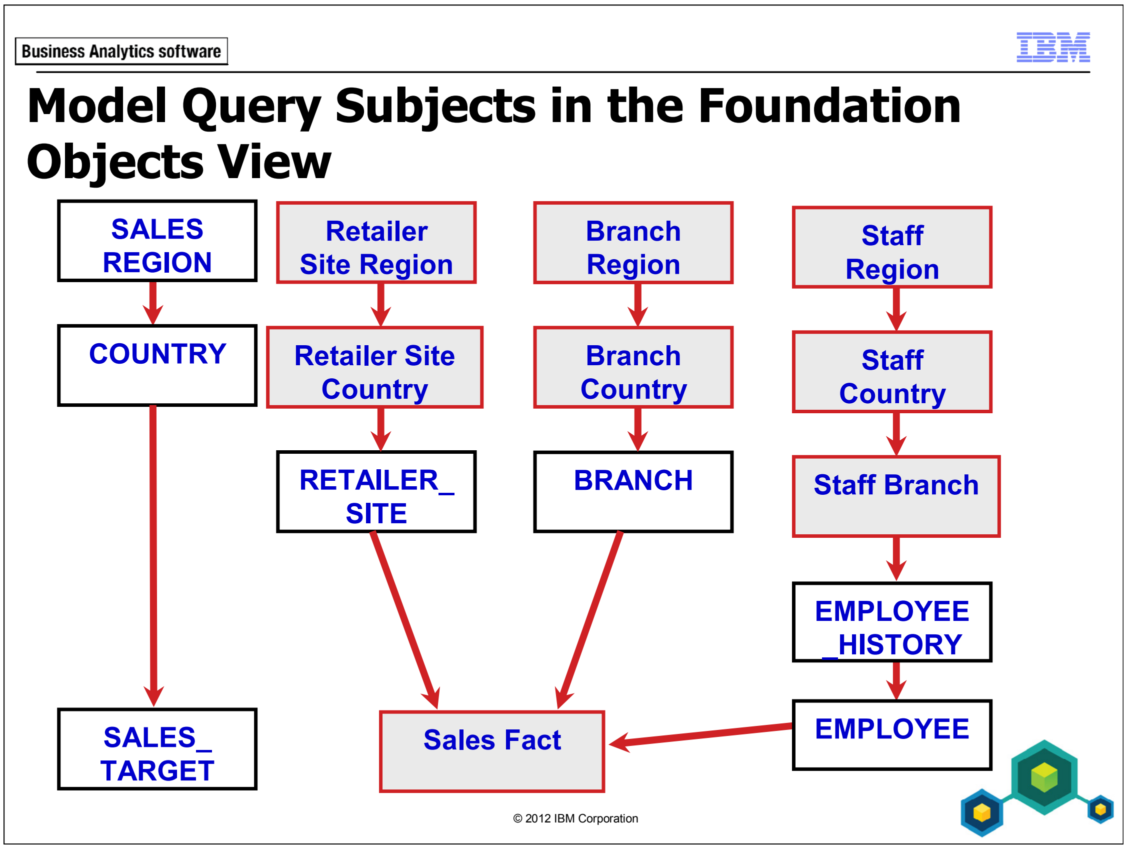
- Model query subjects, calculations and filters are split between Consolidation View and Foundation Objects View, based on need
- All relationships are in the Foundation Objects View
- Compromise between development time and maintenance time
- Smaller project size than Business Logic View method

© 2012 IBM Corporation

In this compromise between the previous two methods, you create model query subjects and their relationships in the lowest layer. The middle layer acts as a consolidation layer with some business logic (where required). For example, this view is where snowflake dimensions are consolidated. It also acts as an insulation layer between reports and the data source.

Calculations may appear in either the lowest or the middle layer, depending on where the related model query subject is created. A model query subject that is created to resolve a reporting issue will be created in the lowest layer. Therefore, any calculations required for that object will be defined in the lowest layer.

The Business Logic View and Consolidation View methods create additional model query subjects even when data source query subjects present no issue and could technically be used in the Presentation View. This can affect physical file size noticeably. The Consolidation View method is the middle ground and is the approach that this course will use.


Business Analytics software

Model Query Subjects in the Foundation Objects View

© 2012 IBM Corporation

This course will use the Consolidation View approach in the demos and workshops. You can view the results of this approach in the CBIFM-Final model located at C:\Edcognos\B5252\CBIFM-Files\Final.

In this module, you will focus on the early stages of the Foundation Objects View layer. However, after the Foundation Objects View layer is completed, it will contain:

- all data source query subjects
- any model query subjects required to resolve reporting issues
- some calculations and filters where appropriate

The above diagram identifies, in the grey boxes (with mixed-case text), the model query subjects used to resolve reporting issues. Sales Fact is used to create a view of ORDER_HEADER and ORDER_DETAILS, thereby controlling the SQL generation, and all others are used to create aliases to control query paths.

Demo 1: Create Alias Model Query Subjects to Avoid Ambiguous Query Paths

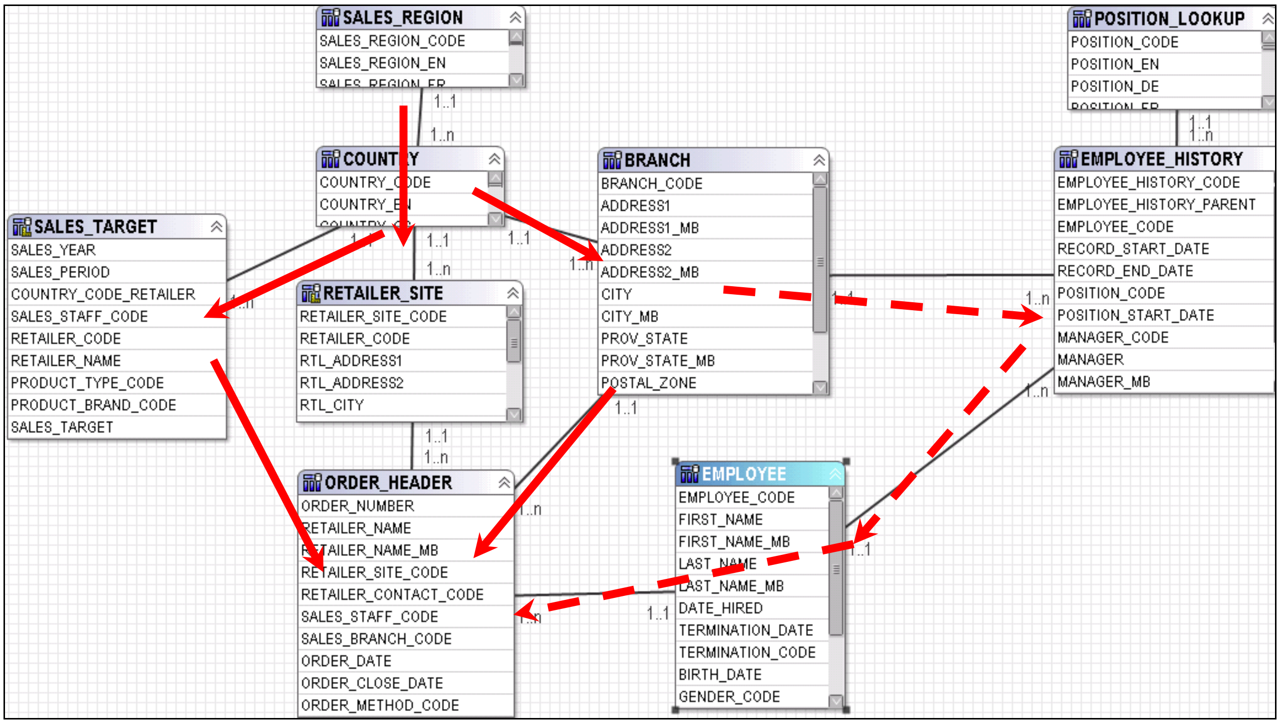
Purpose:
In an earlier exercise, you identified a situation where there was an ambiguous query path between query subjects. SALES_REGION is above COUNTRY in this hierarchy and therefore has the same ambiguous path to ORDER_ DETAILS. You will resolve the problem by creating aliases for SALES_REGION and COUNTRY.

Component:	**Framework Manager**
Project:	**GO Operational**

Task 1. Review the 'ambiguous join' reporting trap.

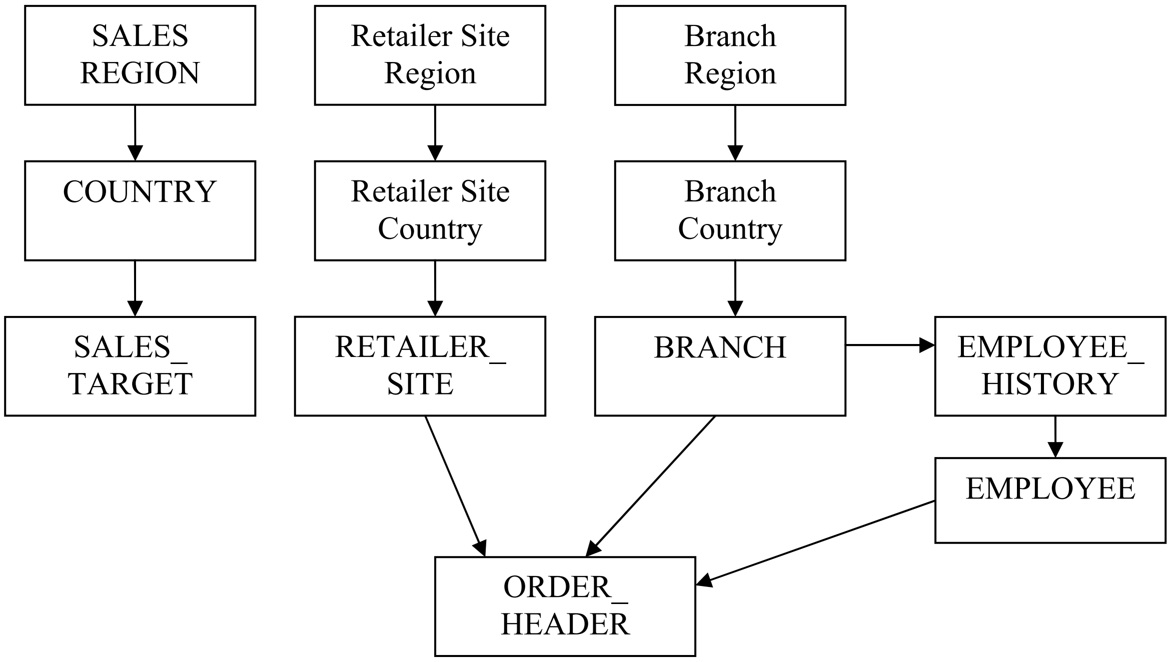
1. In **Framework Manager**, open the **GO Operational** model located at **C:\Edcognos\B5252\CBIFM-Start Files\Module 7\GO Operational**.
2. If prompted, log in as User ID **admin**, and Password **Education1**.
3. In the **Project Viewer** pane, in the **gosales** namespace, right-click **COUNTRY**, and then click **Launch Context Explorer**.
4. If only the **COUNTRY** table displays in **Context Explorer**, click **Show Related Objects** .
5. Click **BRANCH**, and then click **Show Related Objects**.
6. Click **EMPLOYEE_HISTORY**, and then click **Show Related Objects**.

7. If you wish, arrange the diagram as shown below to better see the hierarchy structure:



SALES_REGION and COUNTRY have two separate paths that lead to ORDER_HEADER (or three if you include the route through EMPLOYEE_HISTORY and EMPLOYEE). You want to resolve ambiguity by creating three separate aliases of SALES_REGION and COUNTRY (so that each can be linked to just one of the paths to ORDER_HEADER and ORDER_DETAIL). Keep the original SALES_REGION and COUNTRY to link to SALES_TARGET.

You will achieve the following design at the end of this demo:



8. Close the **Context Explorer**.

Task 2. Create Region aliases as model query subjects.

1. In the **Project Viewer** pane, right-click **gosales**, point to **Create**, and then click **Query Subject**.
2. Name the query subject **Retailer Site Region (alias)**, and then, with **Model** selected, click **OK**.
3. Expand **Foundation Objects View>gosales>SALES_REGION**.

Task 2 Step 1: You want to create model query subjects, not create copies of data source query subjects or shortcuts. Model query subjects can reduce maintenance and allow more flexibility with naming conventions and allow the modeler to override settings

4. Add the following query items by double-clicking them.

   SALES_REGION_CODE
   SALES_REGION_EN

   This alias could also have been created by expanding SALES_REGION in the Project Viewer, selecting the two query items, right-clicking one of them, and clicking Merge in New Query Subject. Both methods have the same results.

   You will include only the columns your report authors will require. Later in the course, you will replace SALES_REGION_EN with a calculation that dynamically picks up the appropriate language column based on the language setting of the user's computer.

   You should rename the items to be more user-friendly before you make copies.

5. Click **OK**, and then, in the **Project Viewer** pane, rename the query items to:

   - **Retailer Site Region Code**
   - **Retailer Site Region**

6. Copy and paste **Retailer Site Region (alias)** in the **gosales** namespace, and then rename the copy, and its two query items, as follows:

   - **Branch Region (alias)**
     - **Branch Region Code**
     - **Branch Region**

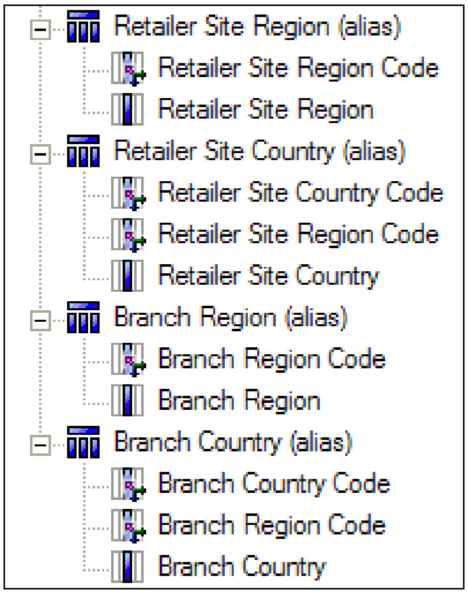
Task 3. Create Country aliases as model query subjects.

1. Repeat Task 2 to create two aliases of **COUNTRY** as follows:
   - **Retailer Site Country (alias)**
     - **Retailer Site Country Code** (from COUNTRY_CODE)
     - **Retailer Site Region Code** (from SALES_REGION_CODE)
     - **Retailer Site Country** (from COUNTRY_EN)

   and

   - **Branch Country (alias)**
     - **Branch Country Code**
     - **Branch Region Code**
     - **Branch Country**
2. Drag **Retailer Site Country (alias)** below **Retailer Site Region (alias)**.

   The results appear as follows:



```
Retailer Site Region (alias)
    Retailer Site Region Code
    Retailer Site Region
Retailer Site Country (alias)
    Retailer Site Country Code
    Retailer Site Region Code
    Retailer Site Country
Branch Region (alias)
    Branch Region Code
    Branch Region
Branch Country (alias)
    Branch Country Code
    Branch Region Code
    Branch Country
```

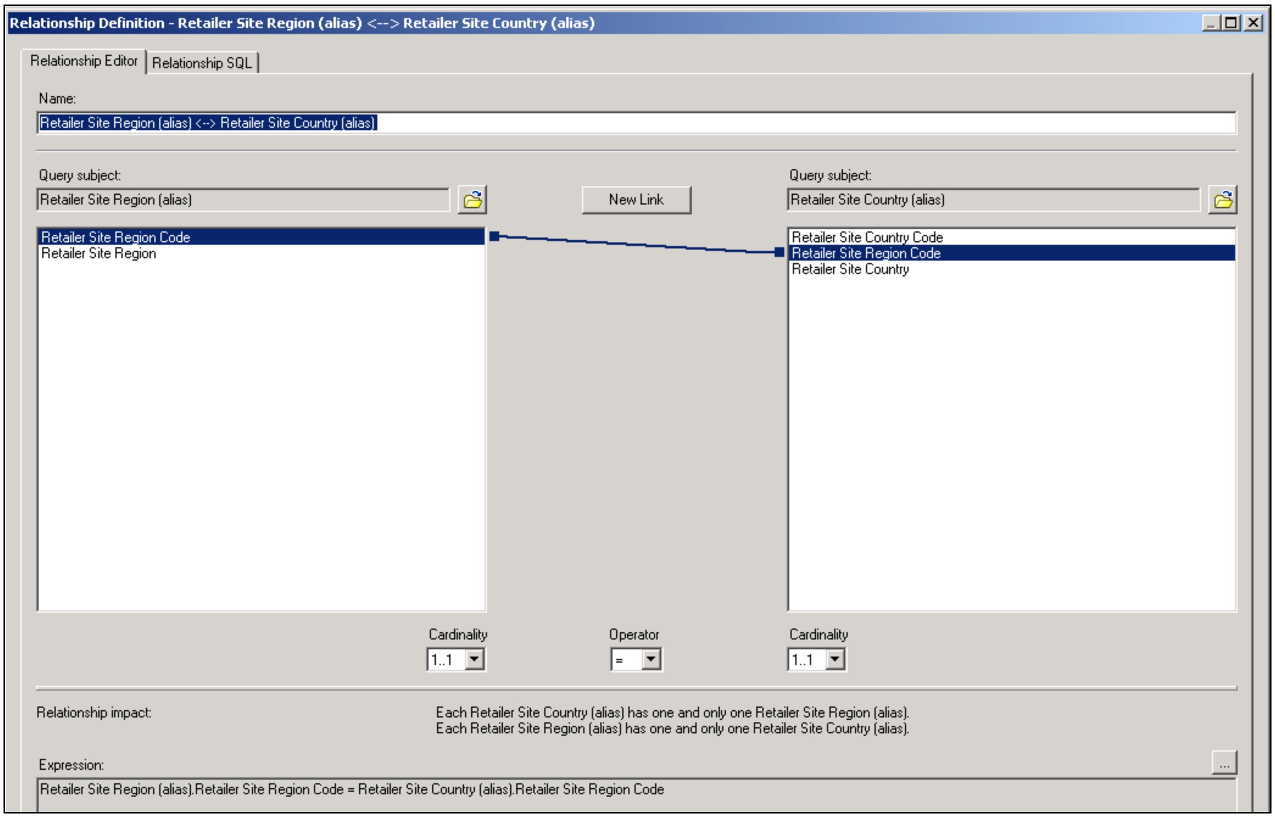
   You now have aliases for two of the three relationships between Region/Country and ORDER_HEADER/ORDER_DETAILS. You will create the third one (for employees) in a workshop.

Task 3 Step 2: The original SALES_REGION and COUNTRY query subjects will be used for relationships to SALES_TARGET later in the course. You will not rename them now because they are original data source query subjects, not model query subjects.

Task 4. Create relationships.

1. In the **Project Viewer** pane, select the following items:
   - **Retailer Site Region (alias)>Retailer Site Region Code**
   - **Retailer Site Country (alias)>Retailer Site Region Code**
2. Right-click one of the selected items, point to **Create** and then click **Relationship**.

   The results appear as follows:


Relationship Definition - Retailer Site Region (alias) <--> Retailer Site Country (alias)
Relationship Editor
Relationship SQL
New Link

Name: Retailer Site Region (alias) <--> Retailer Site Country (alias)

Query subject: Retailer Site Region (alias)

Retailer Site Region Code
Retailer Site Region

Query subject: Retailer Site Country (alias)

Retailer Site Country Code
Retailer Site Region Code
Retailer Site Country

Cardinality: 1..1
Operator: =
Cardinality: 1..1

Relationship impact: Each Retailer Site Country (alias) has one and only one Retailer Site Region (alias).
Each Retailer Site Region (alias) has one and only one Retailer Site Country (alias).

Expression:
Retailer Site Region (alias).Retailer Site Region Code = Retailer Site Country (alias).Retailer Site Region Code

   You now have a relationship between the two query subjects on a common key with the correct cardinality. Again, using this method, the first object selected will be on the 1..1 side of the relationship. You can modify the cardinality after the fact if it does not meet your needs.

3. Under **Retailer Site Country (alias)**, set the **Cardinality** to **1..n**.

4. Click **OK**.
5. Click **No**, when prompted about existing underlying relationships.
6. Repeat the above steps to create the relationship from **Branch Region (alias)** (1..1) to **Branch Country (alias)** (1..n) on **Branch Region Code**.

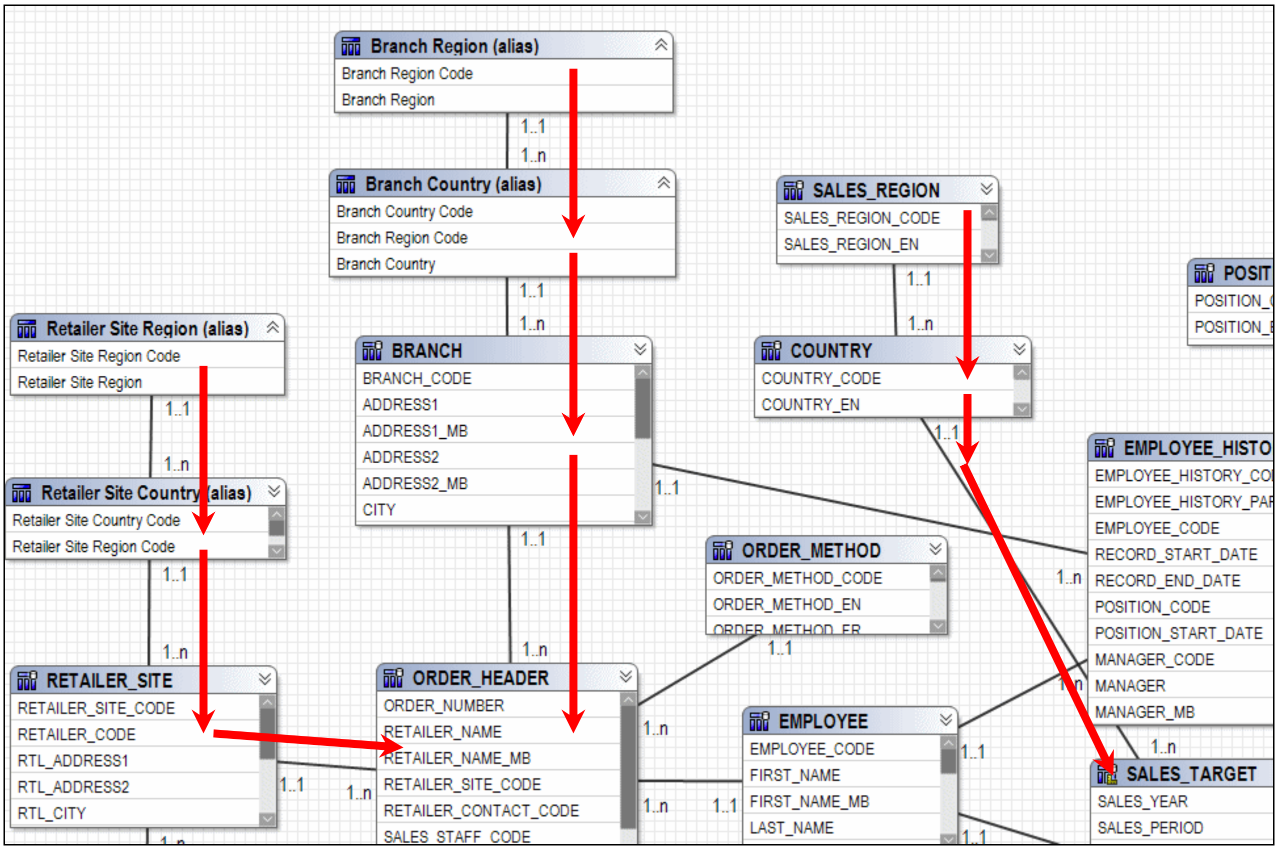
   As shown in the diagram in Task 1, you also want to create relationships between your COUNTRY aliases and the query subjects that lead them to ORDER_HEADER, RETAILER_SITE and BRANCH respectively.
7. Repeat steps **1** to **4** to create the following relationships:
   - **Retailer Site Country (alias)** (Retailer Site Country Code, 1..1) to **RETAILER_SITE** (RTL_COUNTRY_CODE, 1..n)
   - **Branch Country (alias)** (Branch Country Code, 1..1) to **BRANCH** (COUNTRY_CODE, 1..n)

Task 5. Remove old relationships.

1. Right-click **COUNTRY**, click **Launch Context Explorer**, and then click **Show Related Objects**.

   COUNTRY should no longer link to RETAILER_SITE or BRANCH, now that Retailer Site Country (alias) links to RETAILER_SITE and Branch Country (alias) links to BRANCH.
2. Delete the relationship between the following items:
   - **COUNTRY** and **RETAILER_SITE**
   - **COUNTRY** and **BRANCH**
3. Close **Context Explorer**.

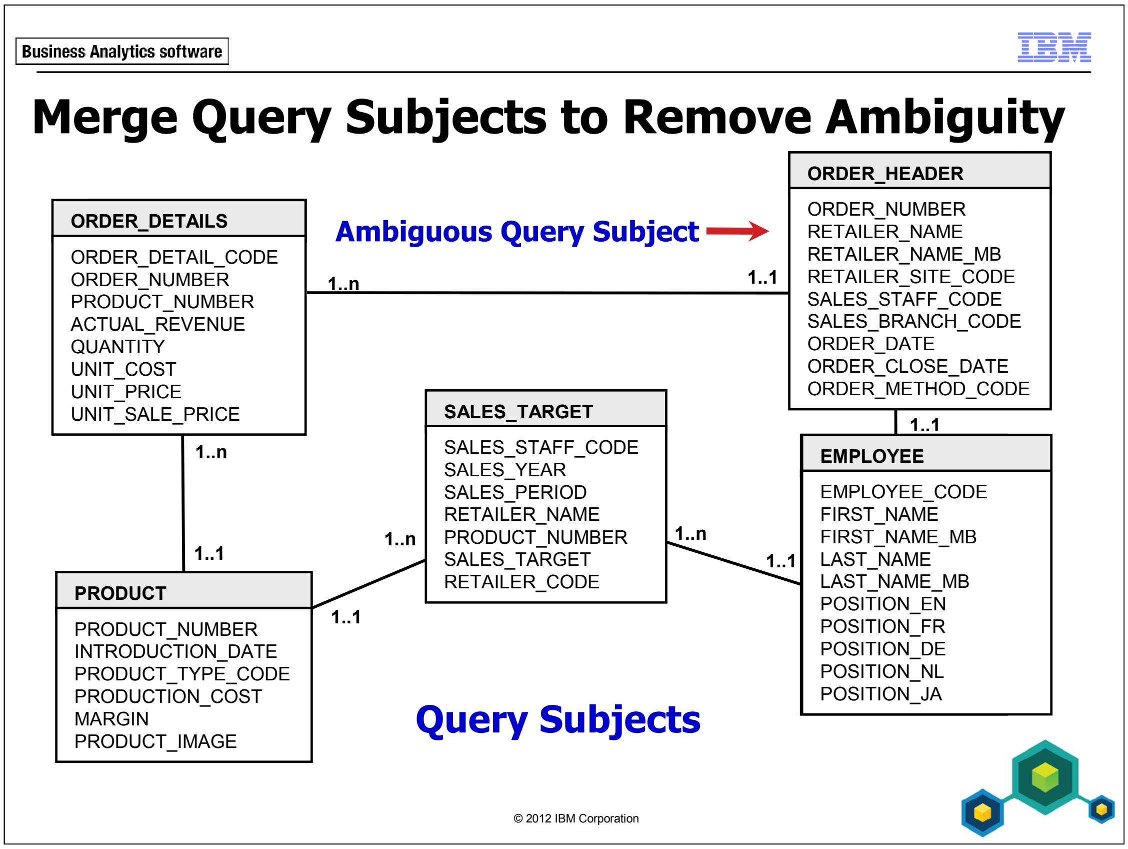
4. Click **Diagram** on the center side of the screen.
5. In the **Diagram**, arrange the objects as shown below:



SALES_REGION and COUNTRY no longer have ambiguous query paths. There is now a clear path for retailer queries, branch queries and sales target queries.

6. Save the project and leave Framework Manager open for the next demo.

Results:
You have taken the first steps to resolving a situation where the proper relationship between SALES_REGION/COUNTRY and ORDER_HEADER/ORDER_DETAILS could be ambiguous. You have created multiple copies of the ambiguous query subjects and relationships among them.


Business Analytics software

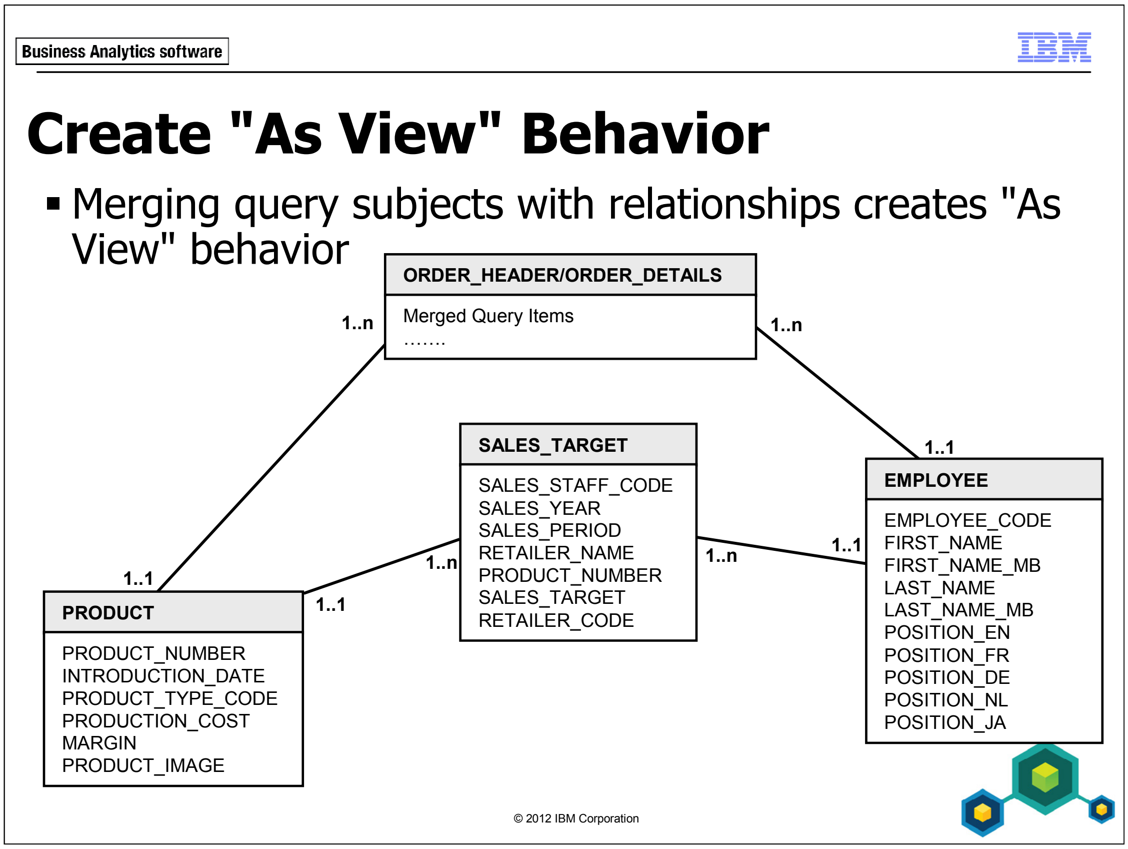
Merge Query Subjects to Remove Ambiguity

© 2012 IBM Corporation

In the example above, the IBM Cognos query engine can identify ORDER_HEADER as a fact or as a dimension depending on the query in which it is used. This is because it has a mix of cardinalities attached to it and contains descriptive information that could be used inappropriately.

ORDER_HEADER is closely related to ORDER_DETAILS and is part of the path for related dimensions, such as EMPLOYEE, in the majority of queries using ORDER_DETAILS. You can merge ORDER_HEADER and ORDER_DETAILS, keeping only required query items, to remove this ambiguity.

ORDER_HEADER is ambiguous because of the attached cardinalities. It is not a candidate for aliasing since it does not play more than one role like COUNTRY or BRANCH does. Those scenarios cause ambiguous query paths. This scenario presents an ambiguous query subject, in particular if you allow authors to choose an item from it such as ORDER_DATE or RETAILER_NAME.



Business Analytics software

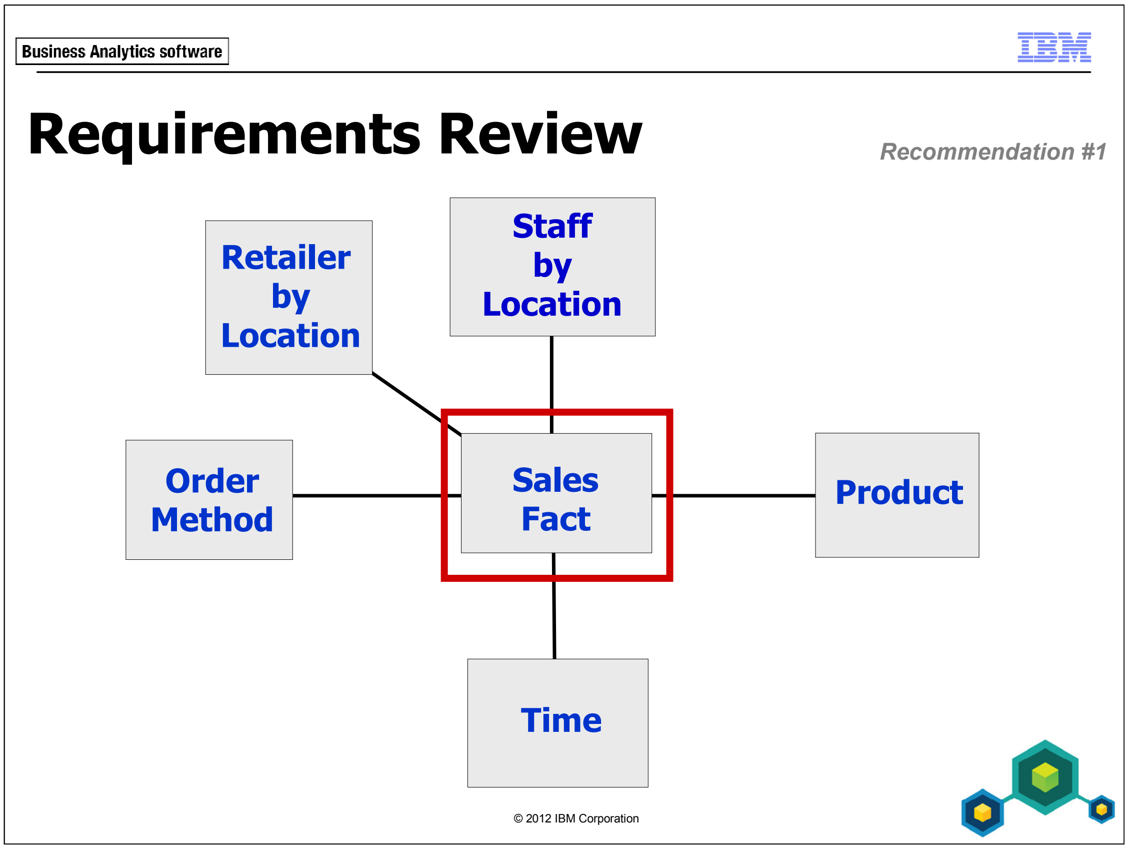
Create "As View" Behavior

- Merging query subjects with relationships creates "As View" behavior

© 2012 IBM Corporation

By merging ORDER_HEADER and ORDER_DETAILS and maintaining existing relationships, you create "As View" behavior in which the underlying join between ORDER_HEADER and ORDER_DETAILS is always honored. In this way, these two query subjects will always be treated as one, and in this case, always as a fact.

Another scenario for forcing joins could be the need to apply a security table to specific data to govern who can see what data. You could merge the security table with the table that needs to be secured in a model query subject which would enforce the join between the two tables.



Business Analytics software

Requirements Review

Recommendation #1

© 2012 IBM Corporation

In the next demo, you will create a model query subject to meet your Sales Fact requirement.

Demo 2: Create a Sales Fact Model Query Subject

Purpose:
You have resolved the ambiguous relationship between SALES_REGION/COUNTRY and ORDER_DETAILS by creating aliases of SALES_REGION and COUNTRY as model query subjects. To further simplify the model and remove ambiguity, you will now create a Sales Fact model query subject that merges ORDER_HEADER and ORDER_DETAILS into one model query subject. This is done to ensure that the query path through ORDER_HEADER is maintained when reporting on sales figures and to remove the ambiguity that ORDER_HEADER presents.

Component:	**Framework Manager**
Project:	**GO Operational**

Task 1. Create a Sales Fact model query subject.

1. In the **Project Viewer** pane, select **ORDER_HEADER** and **ORDER_DETAILS**, right-click one of the selected items, and then click **Merge in New Query Subject**.

   You are asked if you would like to recreate existing relationships.

2. Click **Yes**.

   You want all the relationships recreated because this is a fact query subject that connects to many dimensional query subjects. You did not want to recreate relationships when creating aliases earlier, because you were specifically creating new relationships to avoid ambiguous query paths.

   By merging these two objects and maintaining relationships, the query subject will behave as a view and always generate the underlying join between the two query subjects. This will be examined in more detail later in the course when exploring SQL generation.

3. Rename the new model query subject to **Sales Fact**.
4. Double-click **Sales Fact** to open its **Query Subject Definition** dialog box.
5. Delete the following query items:

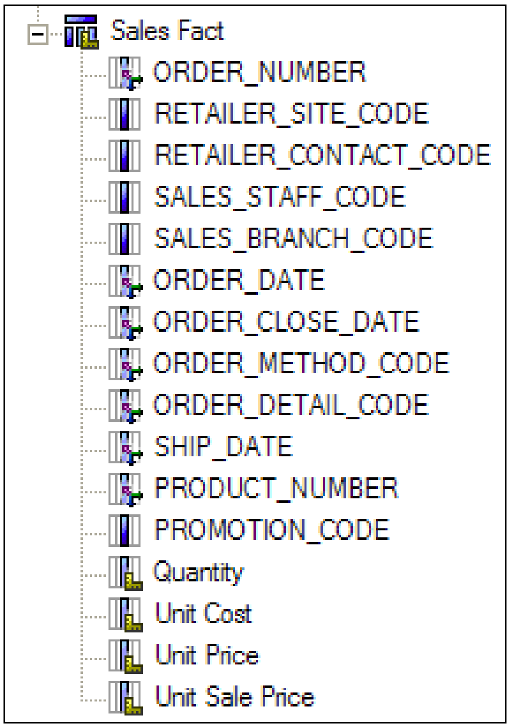
   RETAILER_NAME
   RETAILER_NAME_MB
   ORDER_NUMBER1

   By deleting the retailer names, you are preventing authors from accessing the incorrect version of the retailer name and inadvertently taking descriptive information from a fact query subject.

   ORDER_NUMBER1 is a redundant query item merged in from ORDER_DETAILS. You will use ORDER_NUMBER from ORDER_HEADER.

6. Click **OK**, and then rename the measure query items as follows:
   - **Quantity**
   - **Unit Cost**
   - **Unit Price**
   - **Unit Sale Price**

   The results appear as follows:



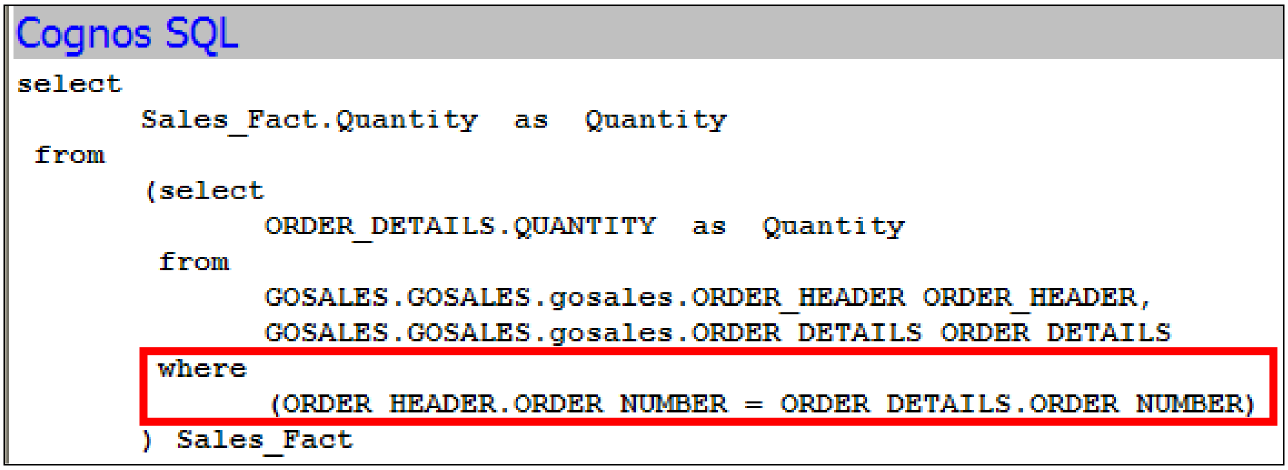
```
Sales Fact
    ORDER_NUMBER
    RETAILER_SITE_CODE
    RETAILER_CONTACT_CODE
    SALES_STAFF_CODE
    SALES_BRANCH_CODE
    ORDER_DATE
    ORDER_CLOSE_DATE
    ORDER_METHOD_CODE
    ORDER_DETAIL_CODE
    SHIP_DATE
    PRODUCT_NUMBER
    PROMOTION_CODE
    Quantity
    Unit Cost
    Unit Price
    Unit Sale Price
```

   These are the query items that will eventually be used in the Presentation View which your authors will see.

Task 2. Test "As View" behavior.

1. Test the **Quantity** query item, and then view the **Query Information** tab.

   The results appear as follows:



Cognos SQL

```
select
       Sales_Fact.Quantity  as  Quantity
 from
       (select
              ORDER_DETAILS.QUANTITY  as  Quantity
        from
              GOSALES.GOSALES.gosales.ORDER_HEADER ORDER_HEADER,
              GOSALES.GOSALES.gosales.ORDER_DETAILS ORDER_DETAILS
       where
              (ORDER_HEADER.ORDER_NUMBER = ORDER_DETAILS.ORDER_NUMBER)
       ) Sales_Fact
```

   Although you selected an item only associated with ORDER_DETAILS, a join with ORDER_HEADER is honored as seen in the where clause. The two underlying query subjects are now treated as one.

2. Click **Close**.

   You will now quickly test the behavior of a merged query with no relationships attached.

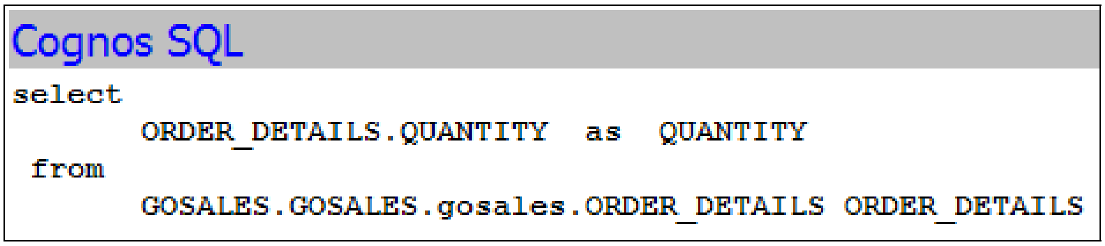
3. In the **Project Viewer** pane, select **ORDER_HEADER** and **ORDER_DETAILS**, right-click one of the selected items, and then click **Merge in New Query Subject**.

   You are asked if you would like to recreate existing relationships.

4. Click **No**.

5. Expand **ORDER_HEADER_ORDER_DETAILS**, test **QUANTITY**, and then click the **Query Information** tab.

   The results appear as follows:



Cognos SQL

```
select
       ORDER_DETAILS.QUANTITY  as  QUANTITY
 from
       GOSALES.GOSALES.gosales.ORDER_DETAILS ORDER_DETAILS
```

   There is no join with ORDER_HEADER. This new query subject is simply acting as a container for the two underlying query subjects. Since you only queried an item from ORDER_DETAILS, the SQL is minimized and only requests information from that one table in the database.

6. Click **Close**, and then click **Undo** on the toolbar to remove the new query subject.

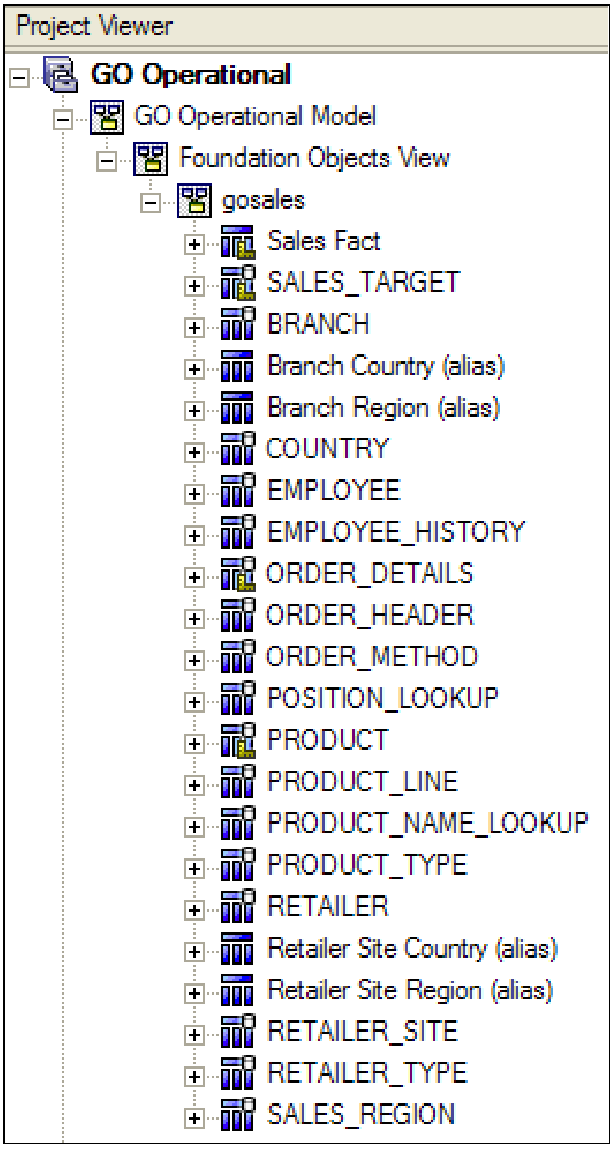
Task 3. Organize gosales namespace and reduce model clutter.

As a convention, you will sort your project objects alphabetically, but place all facts before all dimensions.

1. In the **Project Viewer**, right-click the **gosales** namespace, and then click **Reorder**.
2. Ensure **Ascending** is selected, and then click **OK**.

3. Drag **Sales Fact** to the top in the **gosales** namespace, and then drag **SALES_TARGET** below **Sales Fact**.

   The results appear as follows:



Project Viewer

- GO Operational
  - GO Operational Model
    - Foundation Objects View
      - gosales
        - Sales Fact
        - SALES_TARGET
        - BRANCH
        - Branch Country (alias)
        - Branch Region (alias)
        - COUNTRY
        - EMPLOYEE
        - EMPLOYEE_HISTORY
        - ORDER_DETAILS
        - ORDER_HEADER
        - ORDER_METHOD
        - POSITION_LOOKUP
        - PRODUCT
        - PRODUCT_LINE
        - PRODUCT_NAME_LOOKUP
        - PRODUCT_TYPE
        - RETAILER
        - Retailer Site Country (alias)
        - Retailer Site Region (alias)
        - RETAILER_SITE
        - RETAILER_TYPE
        - SALES_REGION

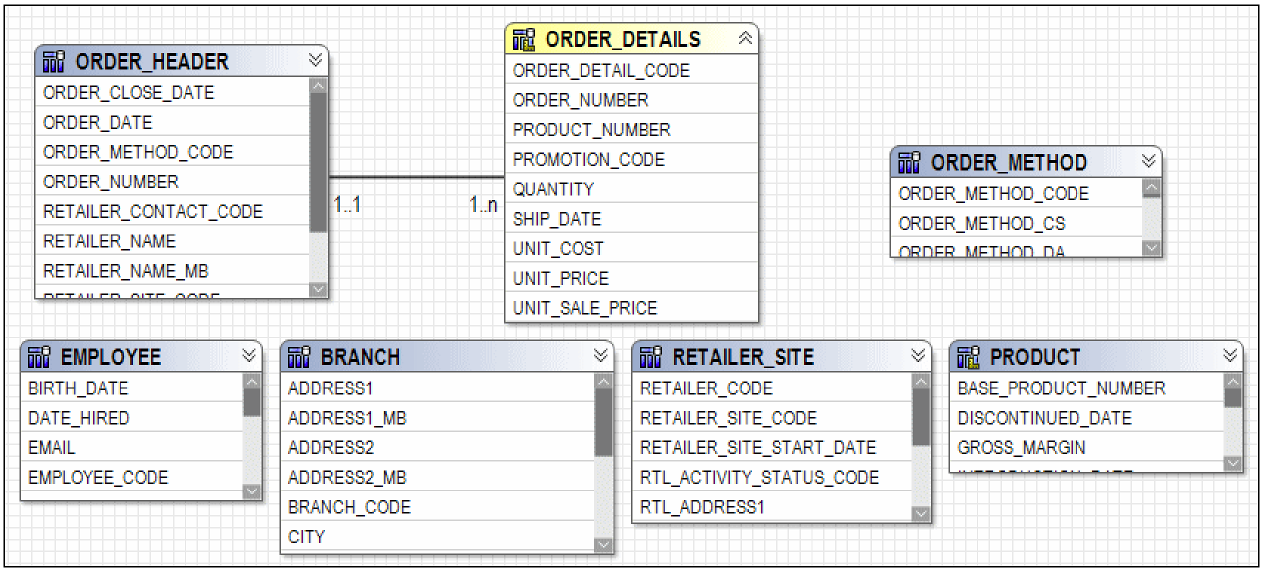
Original data source query subjects that have been replaced by model query subjects (in order to prevent reporting traps) will be placed in a separate folder and non-required relationships will be deleted in order to reduce model clutter.

You will do this now for ORDER_HEADER and ORDER_DETAILS.

4. Right click the **gosales** namespace and choose **Create** > **Folder**, and name it **Original Sales Objects**, and then click **Next**.
5. Expand **Foundation Objects View** > **gosales**, select the **X** next to **ORDER_DETAILS** and **ORDER_HEADER**.

   They will change to green checks.
6. Click **Finish**.
7. Expand **Original Sales Objects**, select **ORDER_HEADER** and **ORDER_DETAILS**, right-click one of the selected items, and then click **Launch Context Explorer**.
8. Click **Show Related Objects**, and then delete all relationships except the one between **ORDER_HEADER** and **ORDER_DETAILS**.

   The results appear as follows:



9. Close the **Context Explorer**.
10. Save your project.

Task 4. Test your new virtual star schema query subjects (optional).

Recall the Ambiguous Relationships demo in the Identify Reporting Issues module. An author was asked to create a query with COUNTRY_EN and QUANTITY, but there was more than one path between the data source query objects and so there was potential for unexpected results.

Now authors must explicitly take the country name from either Retailer Country (down one path) or Branch Country (down another path), depending on whether they want the countries of the clients (retailers) or of the sales offices (branches). The results are always expected and predictable.

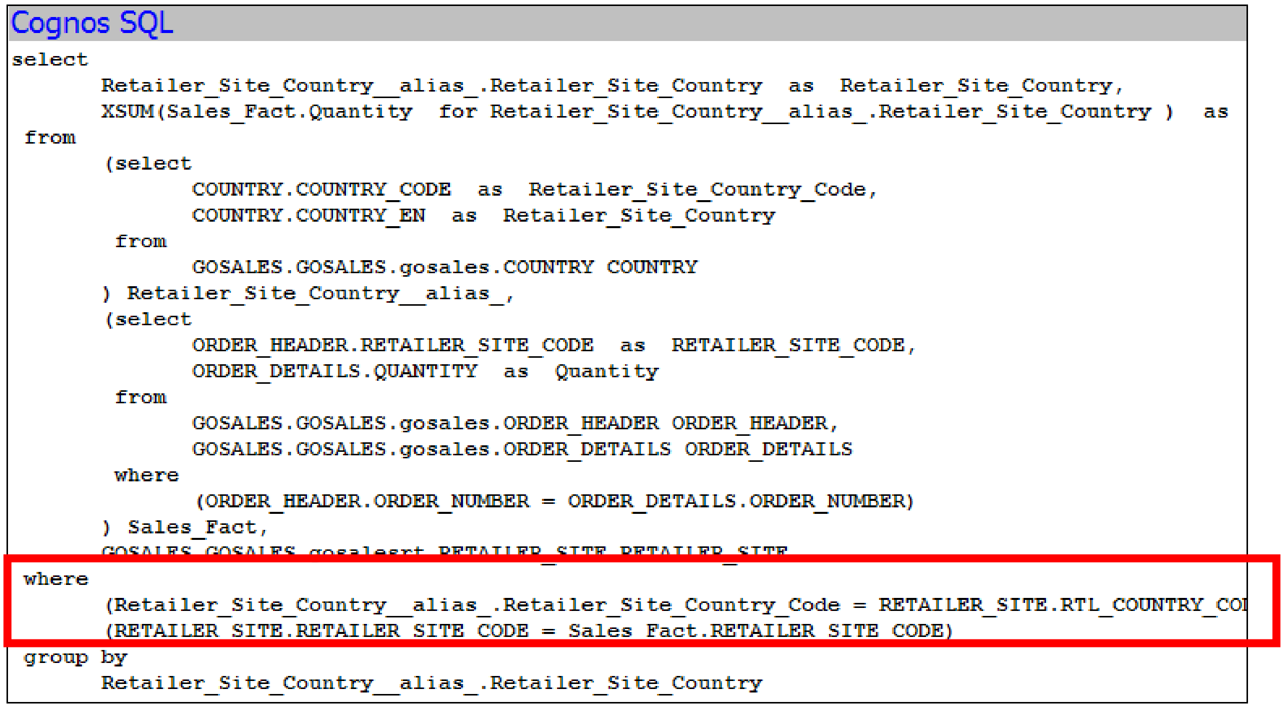
1. Select and test the following query items with **Auto Sum** enabled:

Query Subject	**Query Item**
Retailer Site Country (alias)	**Retailer Site Country**
Sales Fact	**Quantity**

The goal is to report on the number of units sold to each retailer's country.

2. Click the **Query Information** tab.

   The results appear as follows:



Cognos SQL

```
select
       Retailer_Site_Country__alias_.Retailer_Site_Country  as  Retailer_Site_Country,
       XSUM(Sales_Fact.Quantity  for Retailer_Site_Country__alias_.Retailer_Site_Country )  as
 from
       (select
              COUNTRY.COUNTRY_CODE  as  Retailer_Site_Country_Code,
              COUNTRY.COUNTRY_EN  as  Retailer_Site_Country
        from
              GOSALES.GOSALES.gosales.COUNTRY COUNTRY
       ) Retailer_Site_Country__alias_,
       (select
              ORDER_HEADER.RETAILER_SITE_CODE  as  RETAILER_SITE_CODE,
              ORDER_DETAILS.QUANTITY  as  Quantity
        from
              GOSALES.GOSALES.gosales.ORDER_HEADER ORDER_HEADER,
              GOSALES.GOSALES.gosales.ORDER_DETAILS ORDER_DETAILS
        where
              (ORDER_HEADER.ORDER_NUMBER = ORDER_DETAILS.ORDER_NUMBER)
       ) Sales_Fact,
       GOSALES.GOSALES.gosalesrt.RETAILER_SITE RETAILER_SITE
 where
       (Retailer_Site_Country__alias_.Retailer_Site_Country_Code = RETAILER_SITE.RTL_COUNTRY_CO
       (RETAILER_SITE.RETAILER_SITE_CODE = Sales_Fact.RETAILER_SITE_CODE)
 group by
       Retailer_Site_Country__alias_.Retailer_Site_Country
```

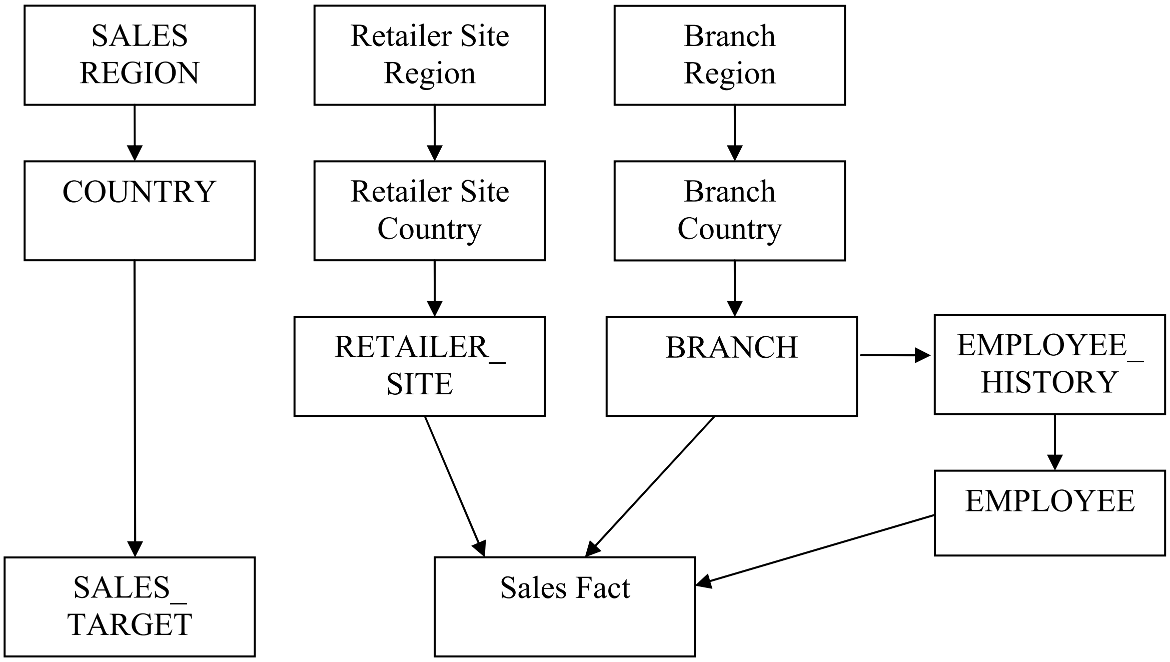
   Notice the final where clause. The query takes the path you expected, Retailer Country (alias) to RETAILER_SITE to Sales Fact. This query works because the model has no alternative query paths. You have removed any ambiguity. The same concept would apply if you tested Branch Country (alias) and Sales Fact.

3. Click **Close**, save the project, and leave Framework Manager open for the Workshop.

Results:
To remove ambiguity that may occur with the ORDER_HEADER data source query subject, you merged ORDER_HEADER with ORDER_DETAILS. You then organized your model and reduced clutter and tested your new alias query subjects with the new Sales Fact query subject to ensure ambiguity has been removed.

Workshop 1: Create Staff-Related Model Query Subjects

In the last two demos, you wanted report authors to avoid accidentally choosing the wrong path between SALES_REGION/COUNTRY and ORDER_HEADER/ORDER_DETAILS, so you created two aliases of SALES_REGION and COUNTRY, one for retailer sites and one for branches. You also, merged ORDER_HEADER and ORDER_DETAILS into a Sales Fact model query subject.



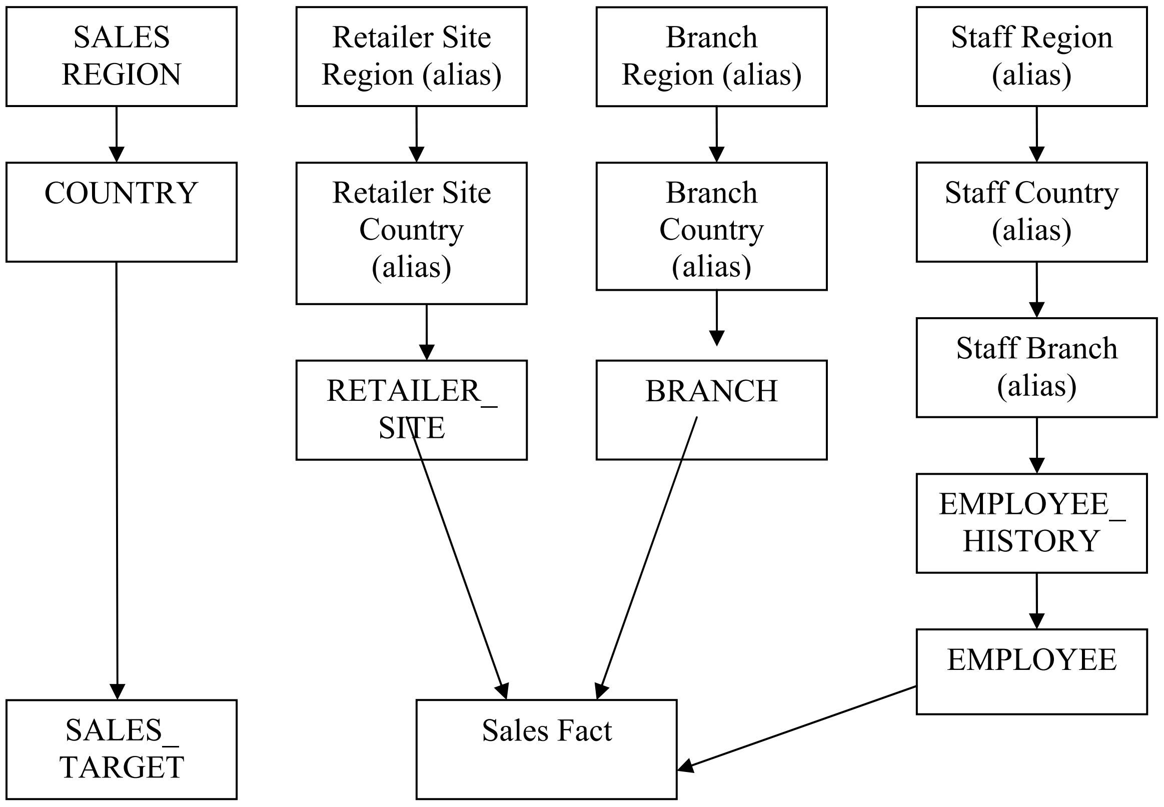
However, BRANCH links to Sales Fact in two paths; directly, as well as through EMPLOYEE_HISTORY and EMPLOYEE.

1. Create a third alias for SALES_REGION and COUNTRY as follows:
   - **Staff Region (alias)**
     - o **Staff Region Code** (from SALES_REGION_CODE)
     - o **Staff Region** (from SALES_REGION_EN)
   - **Staff Country (alias)**
     - o **Staff Country Code** (from COUNTRY _CODE)
     - o **Staff Region Code** (from SALES_REGION_CODE)
     - o **Staff Country** (from COUNTRY_EN)
2. Create a relationship between Staff Region (alias) (**Staff Region Code**, **1..1**) and Staff Country (alias) (**Staff Region Code**, **1..n**).

   You also need a relationship from Staff Country (alias) to BRANCH, but Branch Country (alias) already has a join to BRANCH. To avoid ambiguity, you first need to create an alias for BRANCH.
3. Create a new model query subject called **Staff Branch (alias)**, with the following query items (taken from BRANCH and renamed):
   - o Staff Country Code
   - o Staff Branch Code
   - o Staff Address1
   - o Staff Address2
   - o Staff City
   - o Staff Prov/State
   - o Staff Postal Zone
4. Create a relationship between Staff Country (alias) (**Staff Country Code**, **1..1**) and Staff Branch (alias) (**Staff Country Code**, **1..n**).

5. Create a relationship between Staff Branch (alias) (**Staff Branch Code**, **1..1**) and EMPLOYEE_HISTORY **(BRANCH_CODE**, **1..n)**.
6. Delete the relationship between the original **BRANCH** and **EMPLOYEE_HISTORY**.

   The results appear as follows:



Remember to manually sort your new objects alphabetically and arrange your diagram to show the new staff query path from region through to EMPLOYEE_HISTORY.

For more detailed information outlined as tasks, see the Task Table on the next page. For the workshop results, see the Workshop Results section that follows the Task Table.

Renaming of the BRANCH alias to Staff Branch (alias) helps to qualify its purpose.

What we have are four snowflake dimensions that branch out from a fact, be it Sales Target Fact or Sales Fact. Your next step will be to consolidate each of these four hierarchies into a single model query subject for ease of reporting.

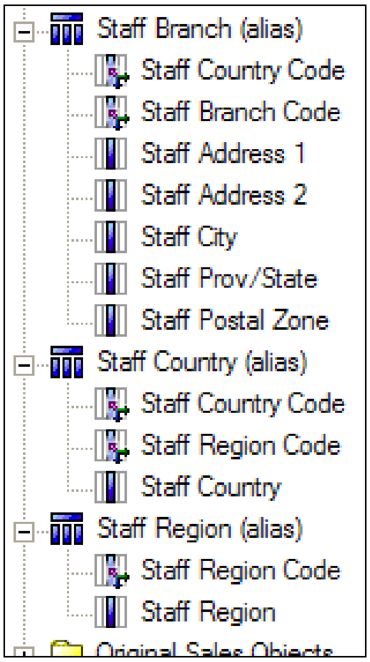
Workshop 1: Task Table

Task	Where to Work	Hints
1. Create Staff Region (alias) and Staff Country (alias).	Project Viewer	• Create the new model query subjects named Branch Region (alias) and Branch Country (alias) and rename objects as follows: Staff Region (alias) containing Staff Region Code and Staff Region Staff Country (alias) containing Staff Country Code and Staff Region Code and Staff Country
2. Create relationship	Project Viewer, Relationship Definition dialog box	• Staff Region (alias) (1..1) to Staff Country (alias) (1..n) on Staff Region Code. Note: Do not replicate underlying relationships.
3. Create Staff Branch (alias) model query subject.	Project Viewer, Query Subject Definition dialog box	• Create a new model query subject called Staff Branch (alias) in the gosales namespace. • Contents (after renaming), from BRANCH: Staff Country Code Staff Branch Code Staff Address1 Staff Address2 Staff City Staff Prov/State Staff Postal Zone

Task	Where to Work	Hints
4. Create branch relationships.	Project Viewer, Relationship Definition dialog box	Note: Do not replicate underlying relationships. • Staff Country (alias) (1..1) to Staff Branch (alias) (1..n) on Staff Country Code. • Staff Branch (alias) (1..1) to EMPLOYEE_HISTORY (1..n) on Staff Branch Code = BRANCH_CODE
5. Delete unwanted relationship.	Project Viewer, Context Explorer	• Delete the relationship between the original BRANCH and EMPLOYEE_HISTORY
6. Organize your project.	Project Viewer, Diagram	• Organize the new model query subjects manually so that they are listed alphabetically in the gosales namespace. • Arrange your diagram to clearly show the new path created for staff information. • Save and leave the project open for the next workshop.

Workshop 1: Workshop Results

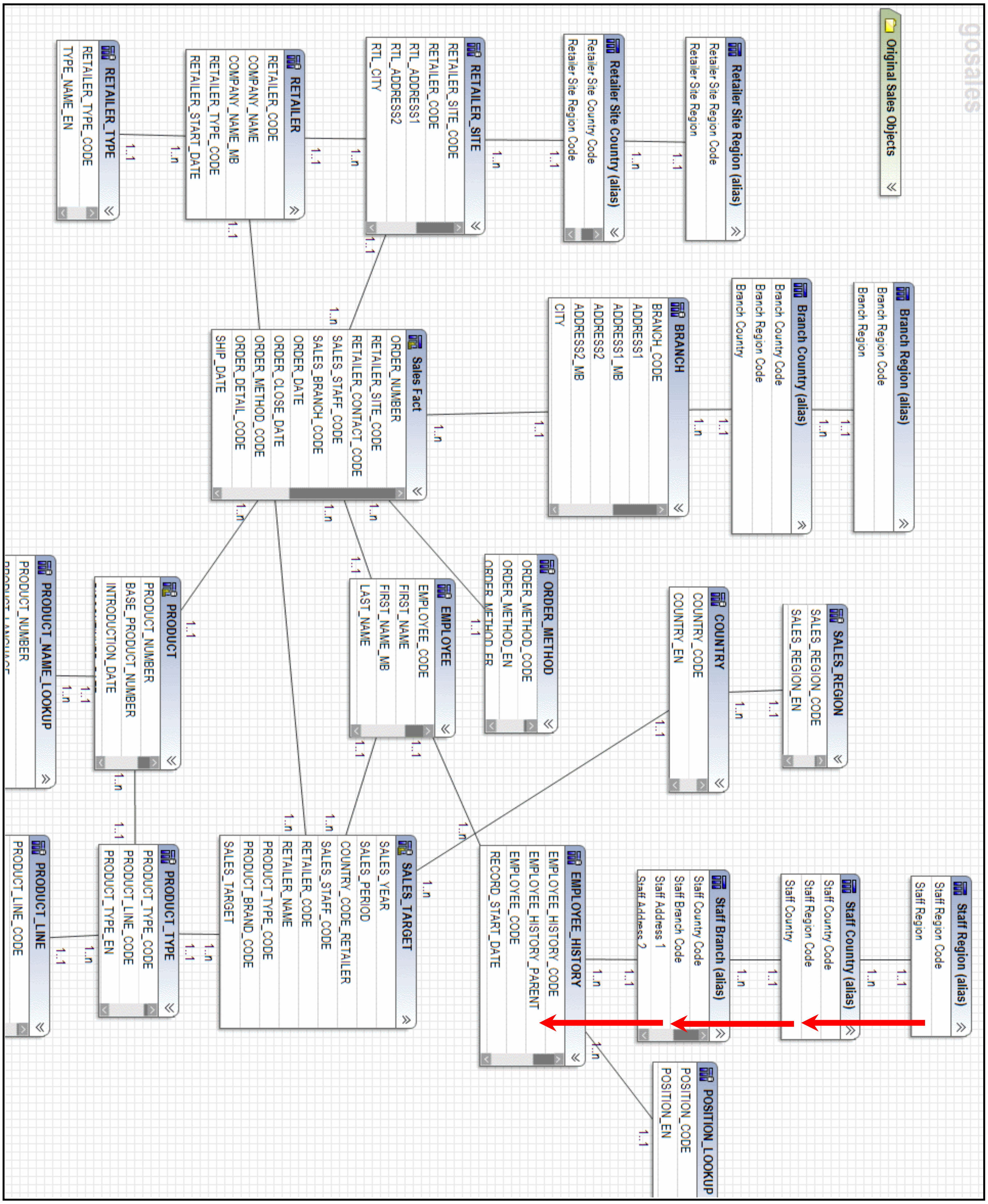
Your Foundation Objects View should include the new objects shown below:



- Staff Branch (alias)
  - Staff Country Code
  - Staff Branch Code
  - Staff Address 1
  - Staff Address 2
  - Staff City
  - Staff Prov/State
  - Staff Postal Zone
- Staff Country (alias)
  - Staff Country Code
  - Staff Region Code
  - Staff Country
- Staff Region (alias)
  - Staff Region Code
  - Staff Region
- Original Sales Objects

Workshop 1: Workshop Results

Your Diagram should be organized to show the new staff query path.



Workshop 2: Merge Query Subjects to Remove Ambiguity

Based on the modeling in the freehand exercise, it was determined that both PRODUCT_TYPE and RETAILER had the potential to be ambiguous query subjects.

To remove their ambiguity, you will merge them with other query subjects.

- PRODUCT_TYPE will be merged with PRODUCT, say yes to recreating relationships, rename the new query subject to Product Type & Product, and remove any redundant query items
    - Move original query subjects into a new folder called Original Product Objects and delete all relationships to them except the one between each other
- RETAILER will be merged with RETAILER_SITE, say yes to recreating relationships, rename the new query subject to Retailer & Retailer Site, and remove any redundant query items
    - Move original query subjects into a new folder called Original Retailer Objects and delete all relationships to them except the one between each other

Arrange the new model query subjects alphabetically.

Save and close Framework Manager.

For more detailed information outlined as tasks, see the Task Table on the next page.

For the workshop results, see the Workshop Results section that follows the Task Table.

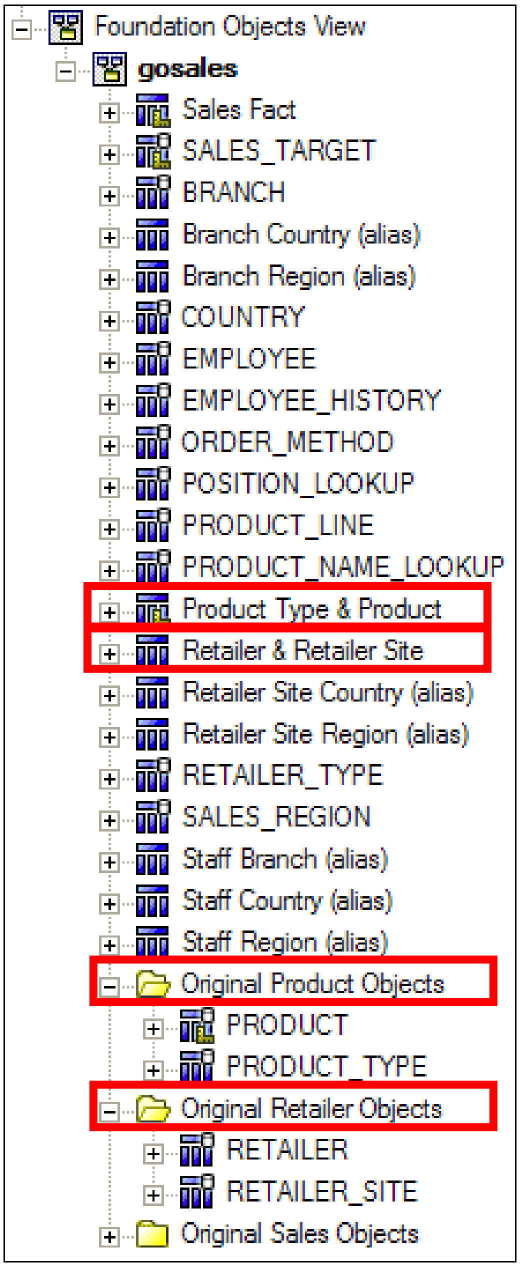
Workshop 2: Task Table

Task	Where to Work	Hints
1. Merge product query subjects.	Project Viewer, Diagram or Context Explorer	• Merge PRODUCT_TYPE and PRODUCT and say yes to recreating relationships • Rename to Product Type & Product • Delete PRODUCT_TYPE_CODE1 • Create a new folder in Foundation Objects View>gosales named called Original Product Objects, and move the PRODUCT_TYPE and PRODUCT query subjects into that folder. • Delete all relationships to PRODUCT_TYPE and PRODUCT except the relationship between PRODUCT_TYPE and PRODUCT • Arrange new model query subject alphabetically

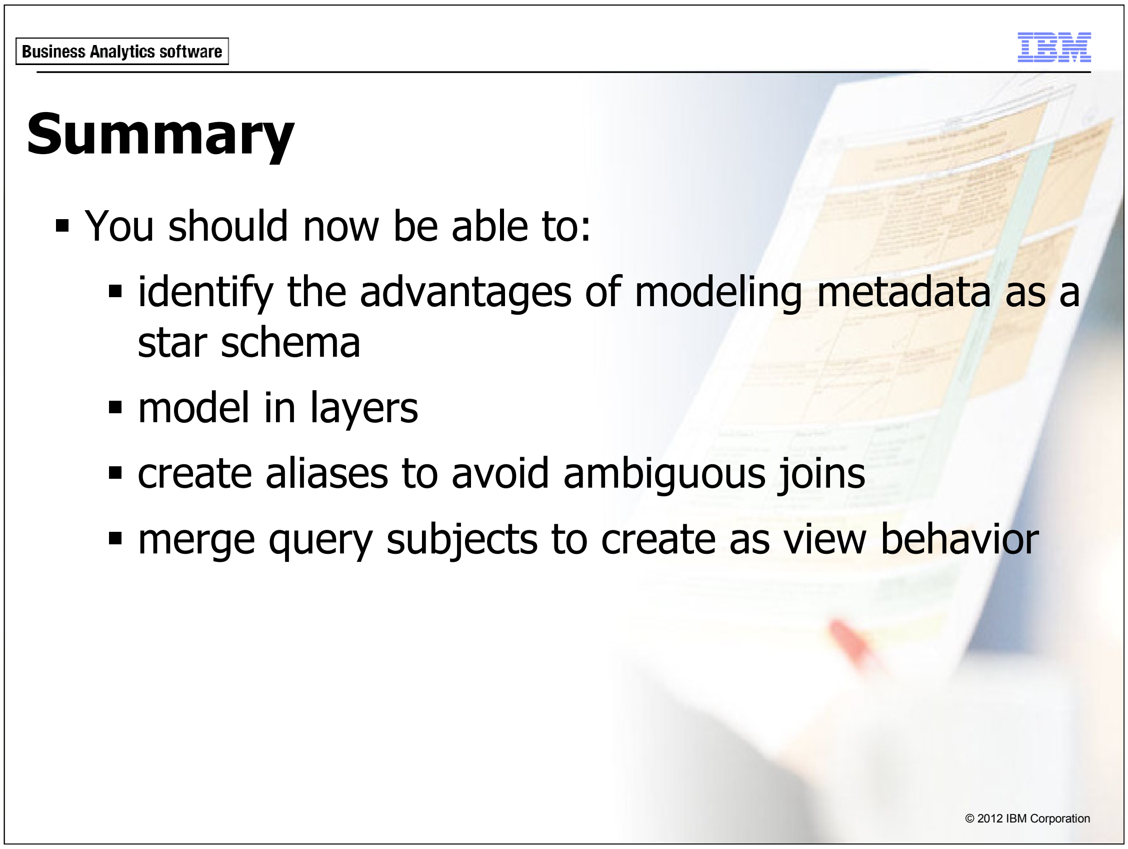
Task	Where to Work	Hints
2. Merge retailer query subjects.	Project Viewer, Diagram or Context Explorer	• Merge RETAILER and RETAILER_SITE and say yes to recreating relationships • Rename to Retailer & Retailer Site • Delete RETAILER_CODE1 • Move RETAILER and RETAILER_SITE to a new folder called Original Retailer Objects in the Foundation Objects View/gosales • Delete all relationships to RETAILER and RETAILER_SITE except the relationship between RETAILER and RETAILER_SITE • Manually arrange new model query subject alphabetically • Save and close Framework Manager

Workshop 2: Workshop Results

Your Foundation Objects View should include the new objects and folders shown below:



```
Foundation Objects View
  gosales
    Sales Fact
    SALES_TARGET
    BRANCH
    Branch Country (alias)
    Branch Region (alias)
    COUNTRY
    EMPLOYEE
    EMPLOYEE_HISTORY
    ORDER_METHOD
    POSITION_LOOKUP
    PRODUCT_LINE
    PRODUCT_NAME_LOOKUP
    Product Type & Product
    Retailer & Retailer Site
    Retailer Site Country (alias)
    Retailer Site Region (alias)
    RETAILER_TYPE
    SALES_REGION
    Staff Branch (alias)
    Staff Country (alias)
    Staff Region (alias)
    Original Product Objects
      PRODUCT
      PRODUCT_TYPE
    Original Retailer Objects
      RETAILER
      RETAILER_SITE
    Original Sales Objects
```

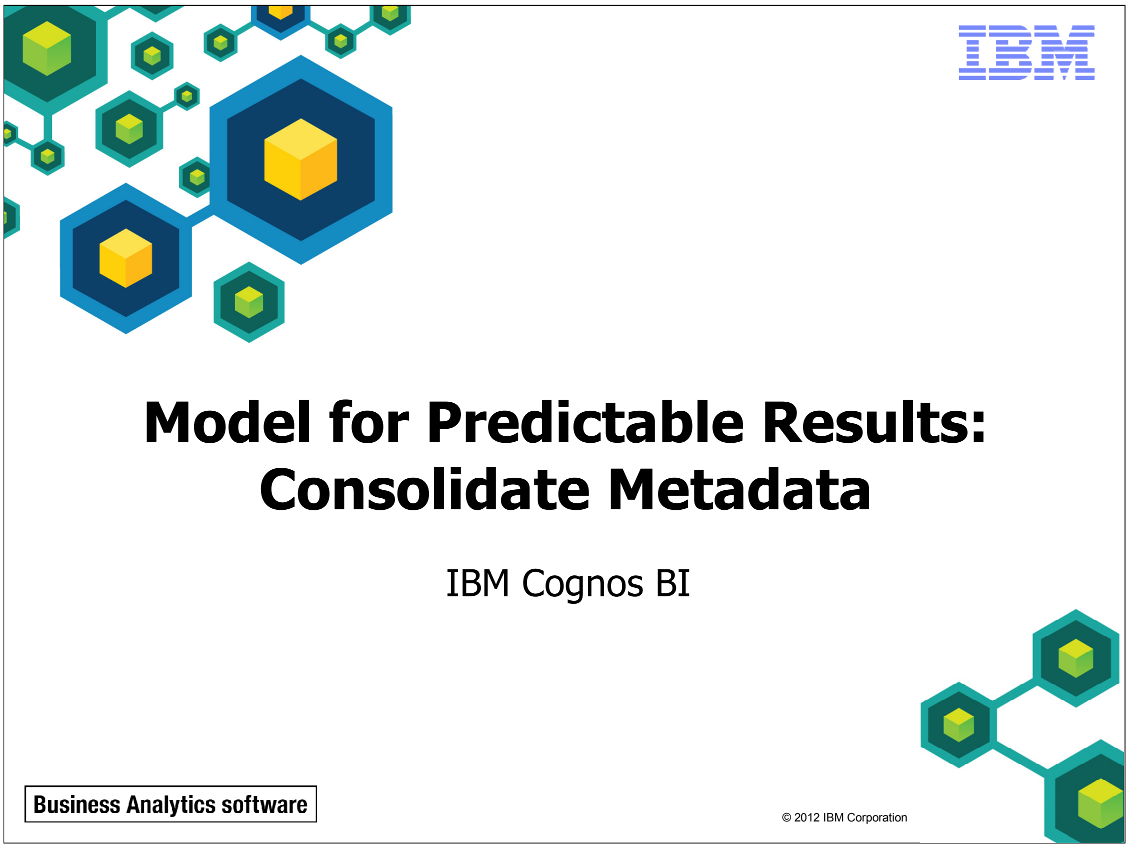


Business Analytics software

Summary

- You should now be able to:
  - identify the advantages of modeling metadata as a star schema
  - model in layers
  - create aliases to avoid ambiguous joins
  - merge query subjects to create as view behavior

© 2012 IBM Corporation



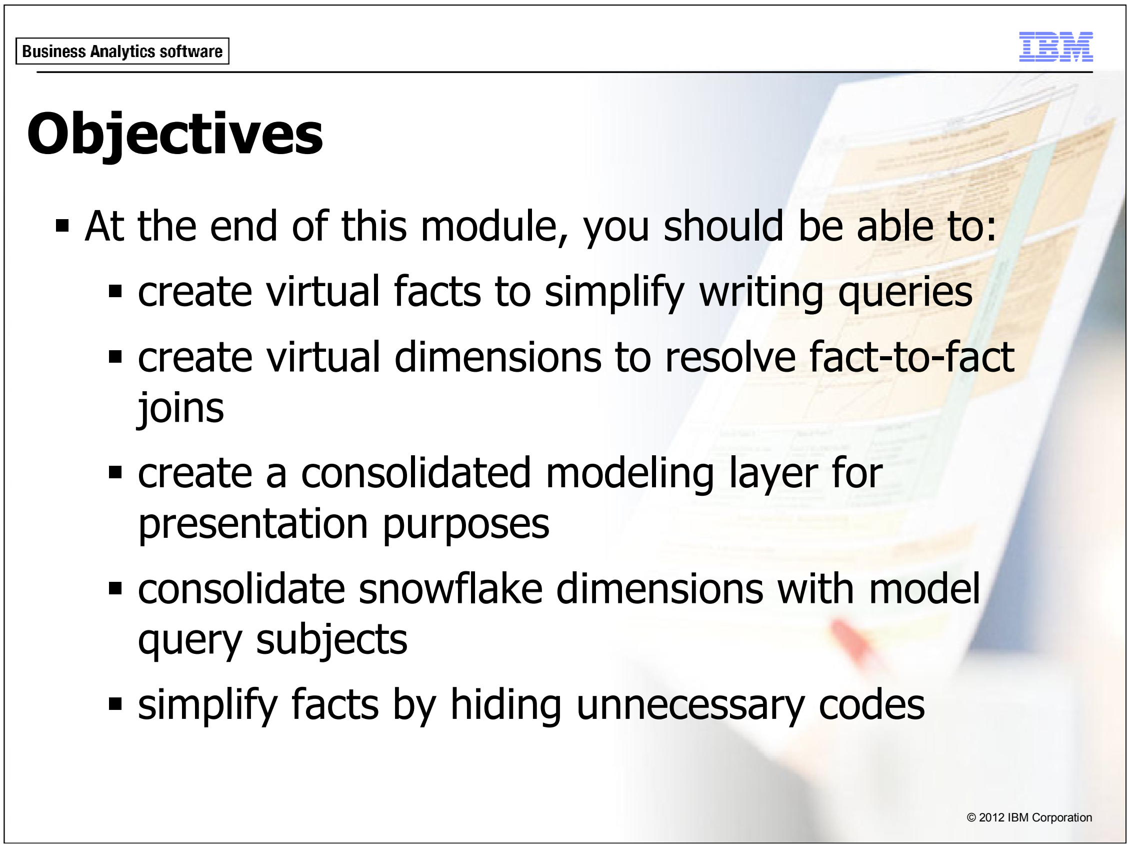
Model for Predictable Results: Consolidate Metadata

IBM Cognos BI

Business Analytics software

Unless otherwise specified in demo or workshop steps, you will always log on to IBM Cognos in Local LDAP namespace using the following credentials:

- User ID: admin
- Password: Education1



Business Analytics software

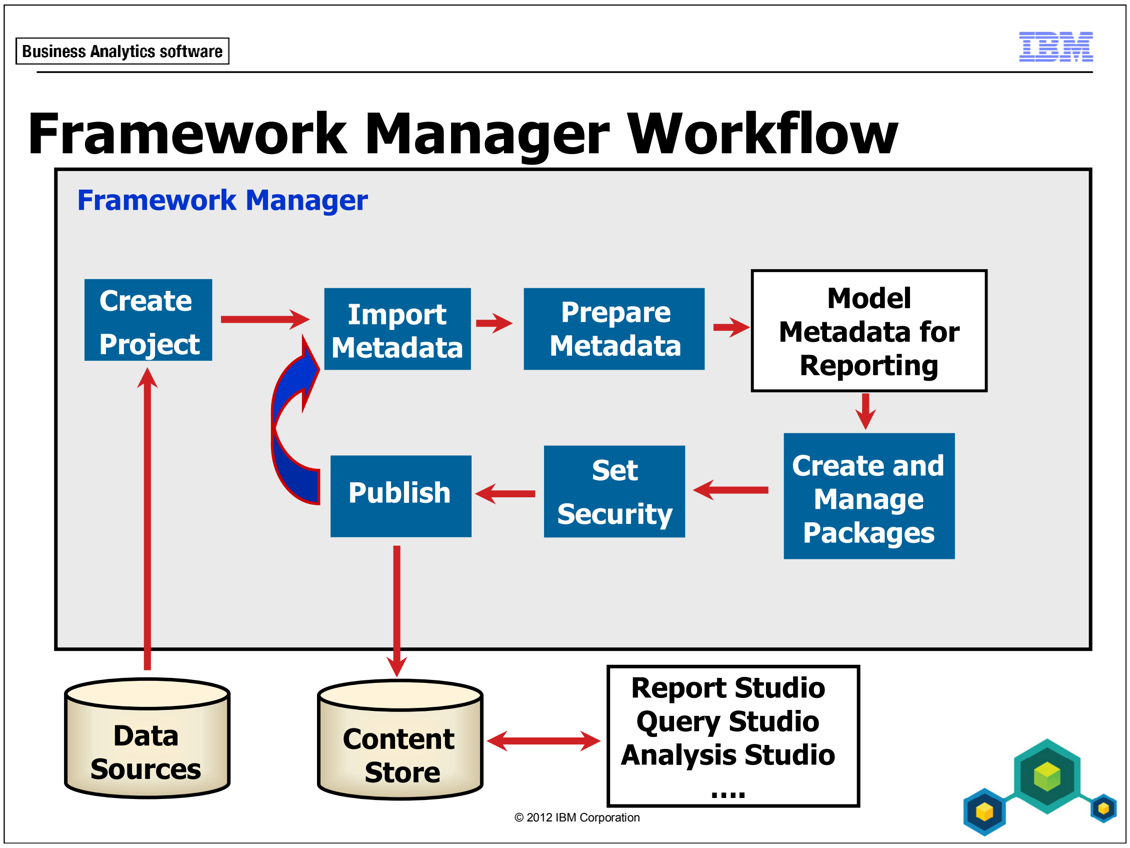
Objectives

- At the end of this module, you should be able to:
  - create virtual facts to simplify writing queries
  - create virtual dimensions to resolve fact-to-fact joins
  - create a consolidated modeling layer for presentation purposes
  - consolidate snowflake dimensions with model query subjects
  - simplify facts by hiding unnecessary codes

© 2012 IBM Corporation

Before reviewing this module, you should be familiar with IBM Cognos, IBM Cognos Connection, Query Studio and Report Studio. Suggested modules to reference:

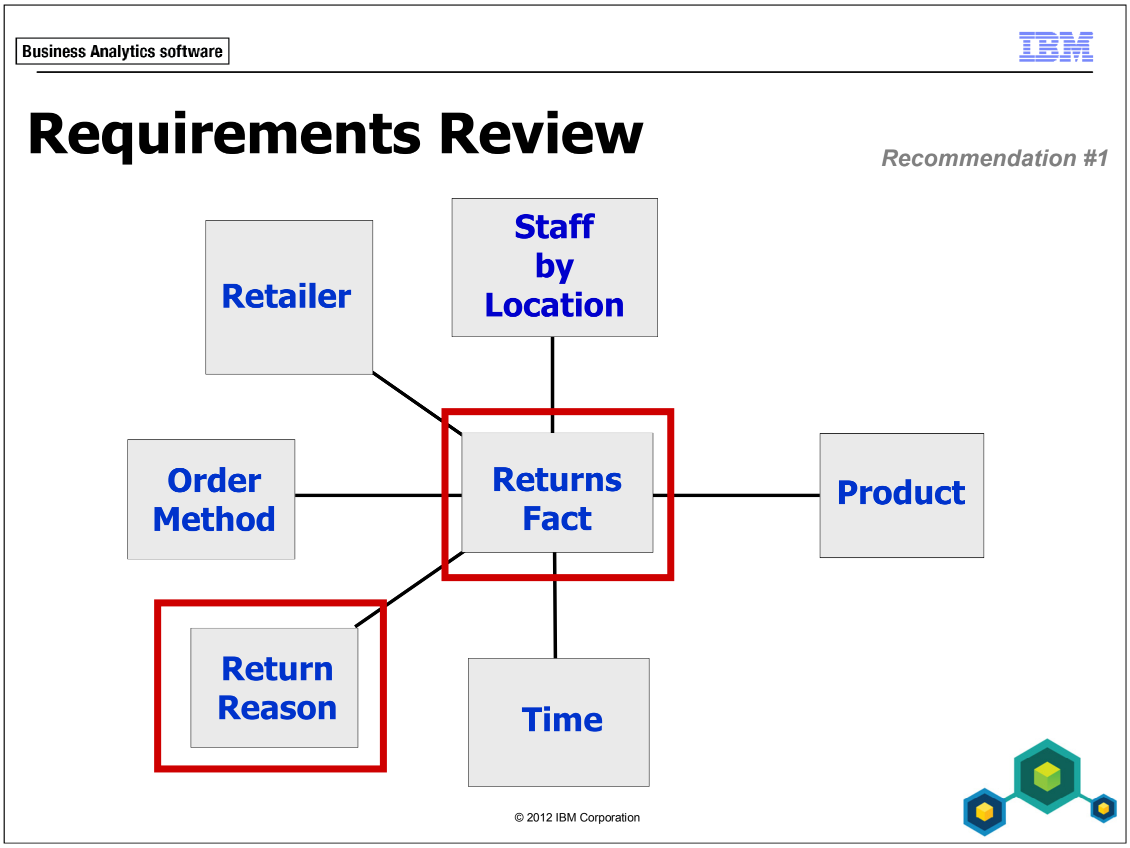
• Overview of IBM Cognos
• Identify Common Data Structures
• Gather Requirements
• Create a Baseline Project
• Prepare Reusable Metadata
• Model for Predictable Results: Reporting Issues
• Model for Predictable Results: Virtual Star Schemas


Business Analytics software

Framework Manager Workflow

© 2012 IBM Corporation

This module deals with creating a Consolidation View in which model query subjects are used to create a layer of consolidated metadata to be presented to authors. This layer will insulate reports from changes to the underlying data source.



Business Analytics software

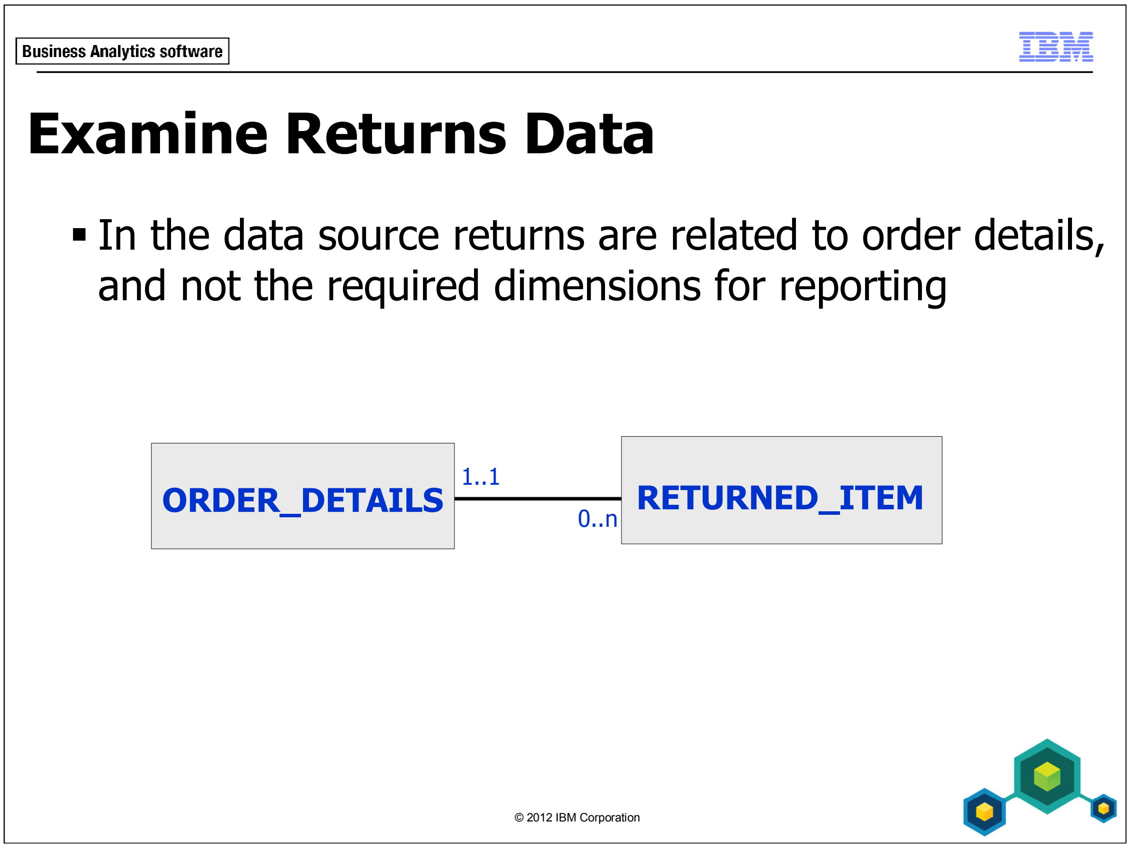
IBM

Requirements Review

Recommendation #1

© 2012 IBM Corporation

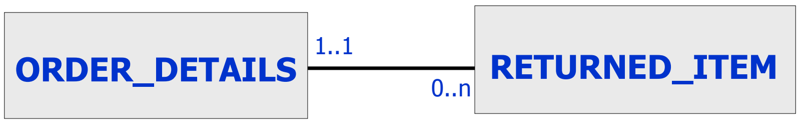
In the next demo, you will import Returns metadata and model it, so that the report author can create a report that examines Returns data for specific dimensions.



Business Analytics software

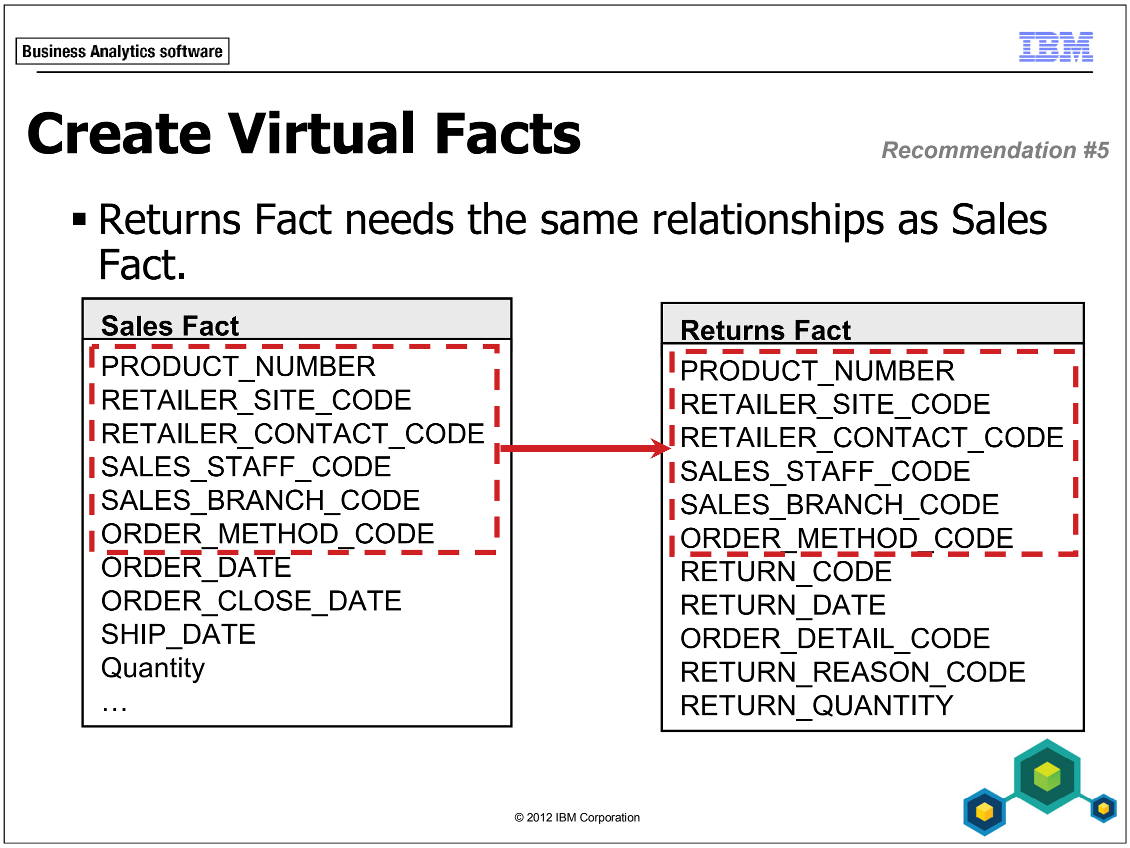
Examine Returns Data

- In the data source returns are related to order details, and not the required dimensions for reporting



The nature of the data does not currently let you directly report returns by staff, order method, or product.

You will need to use modeling techniques to make returns fact data appear directly related to the required dimensions.



Business Analytics software

Create Virtual Facts

Recommendation #5

- Returns Fact needs the same relationships as Sales Fact.

© 2012 IBM Corporation

Because Returns are so closely related to Sales, they should have the same relationships to other dimensions, such as Products, that Sales have. This give authors the ability to write logical reports, such as "which products were returned", without having to include Sales information to get results for Returns.

You can add the missing context by creating a model query subject that includes the query items (keys) that you need in order to create relationships between Returns Fact and all of the required dimensions. The new model query subject behaves like a virtual fact table in a star schema data warehouse. You are simply creating a view to meet your needs and then creating relationships. Keys are hidden so that only facts are visible to authors.

Demo 1: Create a Virtual Fact

Purpose:
Report authors would like to create reports on returns (such as how many returns are on a particular product) without having to explicitly include Sales Fact to get the correct results. This can be done through a Returns Fact, which will also fit the goal of modeling as a virtual star schema. You can create relationships between this fact and other dimensional model query subjects.

Component:	**Framework Manager**
Project:	**GO Operational**

Task 1. Import returns metadata.

1. In **Framework Manager**, open the **GO Operational** model located at **C:\Edcognos\B5252\CBIFM-Start Files\Module 8\GO Operational**.
2. If prompted, log in as User ID **admin**, and Password **Education1**.
3. In the **Project Viewer** pane, right-click the **gosales** namespace, and then click **Run Metadata Wizard**.
4. Ensure that **Data Sources** is selected, and then click **Next**.
5. Ensure **GOSALES** is selected, and then click **Next**.
6. In the list of objects, expand **GOSALES>Tables**, and then select **RETURNED_ITEM** and **RETURN_REASON**.
7. Click **Next**.
8. On the **Generate Relationships** page, select **Both**.

   Note: If you do not select Both, you will get a cross-join error when creating the Model Query Subject.

9. Click **Import**, and then click **Finish**.
10. Expand the two new data source query subjects to view their contents.

    RETURNED_ITEM will be used to create Returns Fact and RETURN_REASON will be used as a dimension to provide context to returns.

    You are now ready to create a new virtual fact by combining query items from several data source query subjects into one model query subject.

Task 2. Build the Returns Fact.

1. In the **gosales** namespace, create a new model query subject called **Returns Fact**, and then click **OK**.
2. In the **Query Subject Definition**, expand **Foundation Objects View**> **gosales**, and **RETURNED_ITEM**, and then add the following query items to the definition:

Query Subject	Query Item
RETURNED_ITEM	**RETURN_CODE** **RETURN_DATE** **ORDER_DETAIL_CODE** **RETURN_REASON_CODE**
ORDER_HEADER (in Original Sales Objects folder)	**RETAILER_SITE_CODE** **SALES_STAFF_CODE** **SALES_BRANCH_CODE** **ORDER_METHOD_CODE**
ORDER_DETAILS (in Original Sales Objects folder)	**PRODUCT_NUMBER**
RETURNED_ITEM	**RETURN_QUANTITY**

Task 2: You could just move the three RETURNED_ITEM query items into Sales Fact, instead of creating Returns Fact, providing you also add a determinant to Sales Fact to handle multiple levels of granularity. However, doing so would cause all authored reports and queries on Sales Fact to do an outer join to RETURNED_ITEM. Since most Sales Fact queries are not about returns, this would be a lot of unnecessary processing.

3. Click **OK**, expand **Returns Fact**, and then rename **RETURN_QUANTITY** to **Return Quantity**.
4. Drag **Returns Fact** to just below **SALES_TARGET**, and then drag **RETURN_REASON** below **RETAILER_TYPE**.

   You now have three query subjects for facts (Sales Fact, SALES_TARGET, and Returns Fact), followed by all the query subjects for dimensions.
5. Drag **RETURNED_ITEM** to the **Original Sales Objects** folder, and then rename the folder to **Original Sales & Returns Objects**.

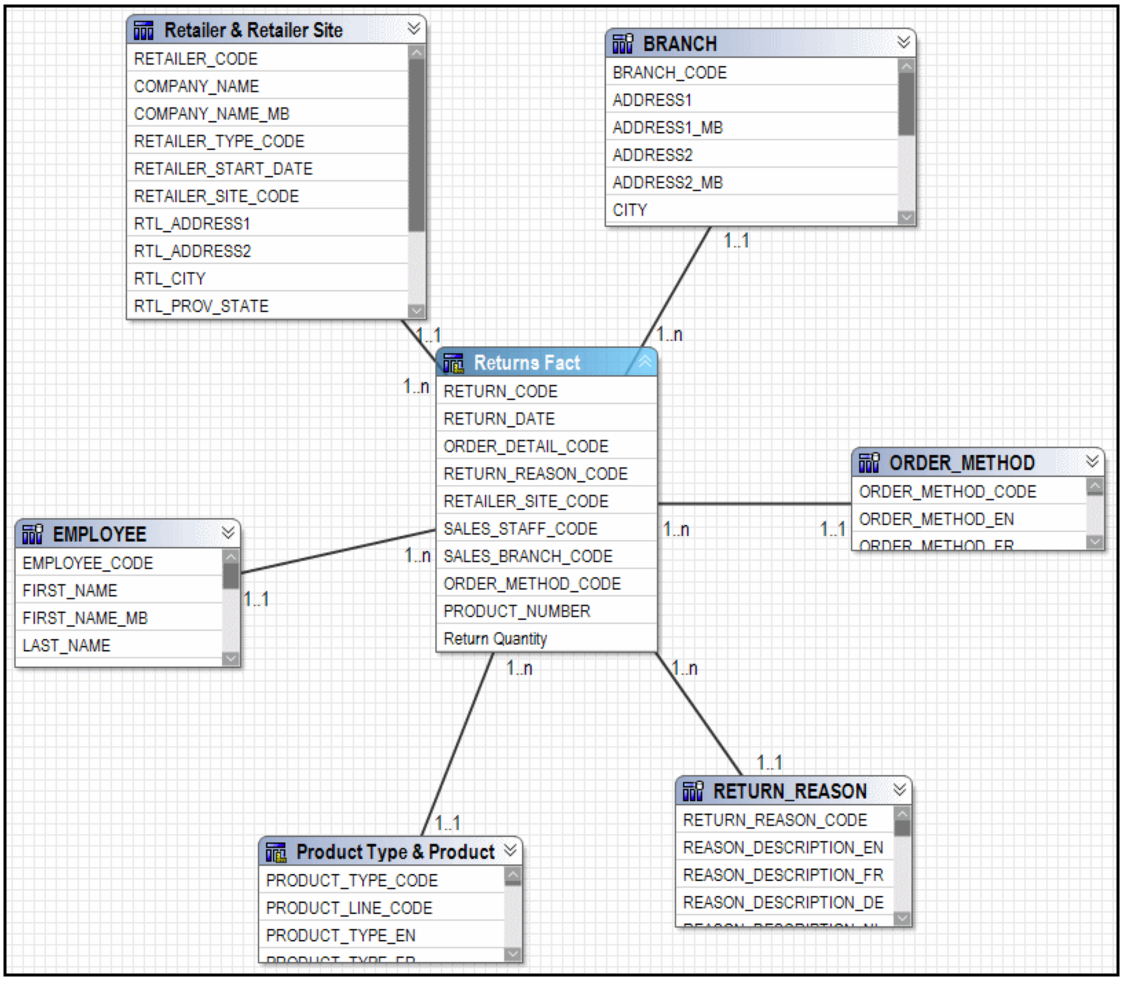
Task 3. Build relationships.

1. Create the following relationships, clicking **No** if asked to replicate the existing underlying relationships:
   - **RETURN_REASON** (RETURN_REASON_CODE, 1..1) to **Returns Fact** (RETURN_REASON_CODE, 1..n)
   - **ORDER_METHOD** (ORDER_METHOD_CODE, 1..1) to **Returns Fact** (ORDER_METHOD_CODE, 1..n)
   - **Product Type & Product** (PRODUCT_NUMBER, 1..1) to **Returns Fact** (PRODUCT_NUMBER, 1..n)
   - **EMPLOYEE** (EMPLOYEE_CODE, 1..1) to **Returns Fact** (SALES_STAFF_CODE, 1..n)
   - **Retailer & Retailer Site** (RETAILER_SITE_CODE, 1..1) to **Returns Fact** (RETAILER_SITE_CODE, 1..n)
   - **BRANCH** (BRANCH_CODE, 1..1) to **Returns Fact** (SALES_BRANCH_CODE, 1..n)

Task 3: These relationships are the same ones that Sales Fact has (with the exception of the one to RETURN_REASON). Optionally, launch Context Explorer on Sales Fact see this as well.
Task 3: In a bigger picture, it can be argued that these relationships should be 0..n instead of 1..n. For example, you may wish to report on the products, staff, or retailers with no returns. The scope of our reporting package, however, does not include those situations. You can change the relationships to 0..n if the business case requires it, but if it doesn't, you should keep them at 1..n for the better performance.

2. Launch **Context Explorer** from **Returns Fact**, and then click **Show Related Objects**.
3. Click **Auto Layout**, ensure **Layout Style** is set to **Star**, and then click **Apply**.
4. Click **Close**.

   The results appear as follows:



   Notice that Returns Fact appears in a star schema in which the fact query subject is surrounded by its related dimensions.

   Note: The placement of the related dimensions may display in a different order.

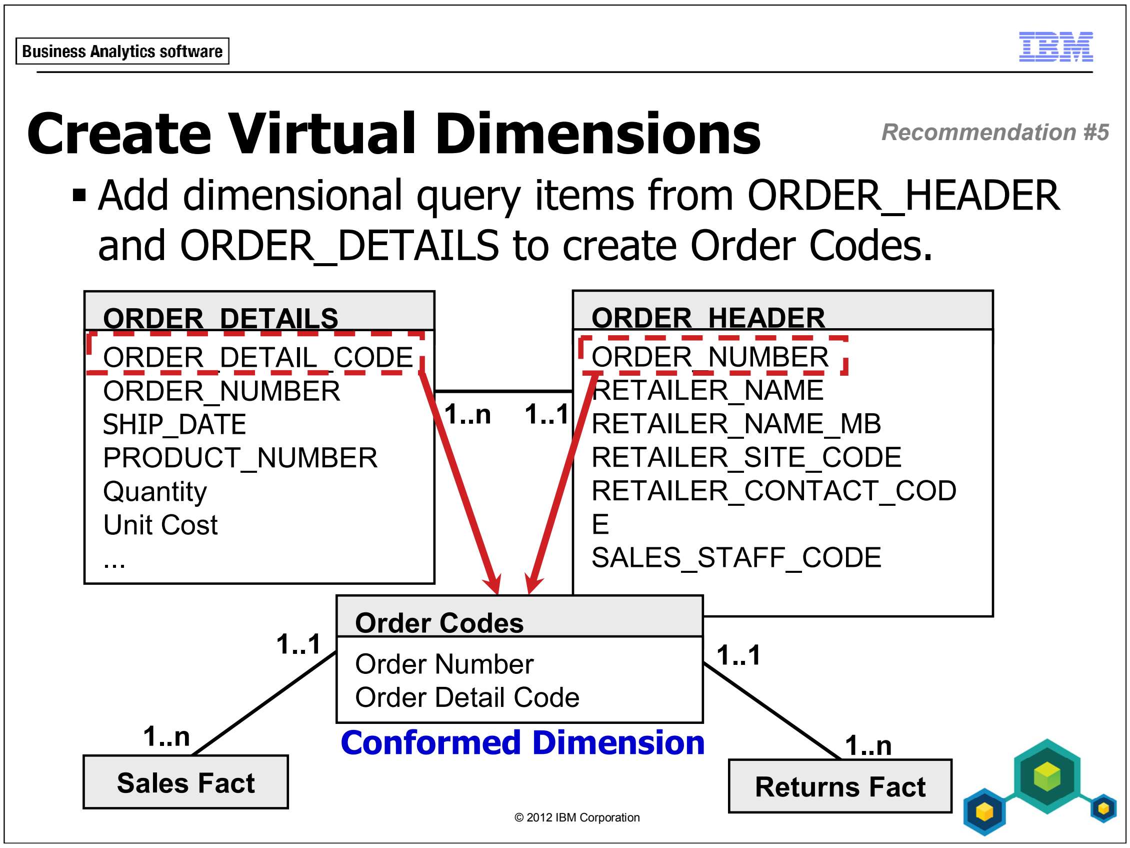
5. Close **Context Explorer**.

Task 4. Delete redundant relationship.

The original RETURNED_ITEM has a relationship to RETURN_REASON. This relationship is no longer required since the new Returns Fact model query subject now has a relationship to RETURN_REASON.

1. Right-click **RETURN_REASON**, and then click **Launch Context Explorer**.
2. Click **Show Related Objects**, and then delete the relationship between **RETURN_REASON** and **RETURN_ITEM**.
3. Close **Context Explorer**, and then save the project.

Results:
Report authors would like to create reports on returns (such as how many returns are for a particular product) without having to explicitly include Sales Fact to get the correct results. This was done by creating a Returns Fact, as well as relationships between this fact and other dimension model query subjects.



Business Analytics software

IBM

Create Virtual Dimensions

Recommendation #5

- Add dimensional query items from ORDER_HEADER and ORDER_DETAILS to create Order Codes.

Certain non-fact query items from fact tables can be placed in a model query subject to ensure that they are viewed as coming from a dimension by IBM Cognos.

For example, ORDER_NUMBER and ORDER_DETAIL_CODE are required for reporting purposes and are located in ORDER_HEADER and ORDER_DETAILS respectively. By placing these query items in a new model query subject called Order Codes, you create a query subject that acts like a virtual dimension table as seen in a star schema data warehouse. This ensures that the dimensional query items are not missing from the final model presentation.

In the underlying data source, the only way to query returns is through ORDER_DETAILS, which is a fact-to-fact join. Once created, the Order Codes query subject can act as a conformed dimension between Sales Fact and Returns Fact.

Workshop 1: Create a Virtual Dimension

Analyzing the model design for flexibility, you see two reasons to create an Order Codes dimensional model query subject. First, it creates a virtual dimension that ensures that dimensional query items in ORDER_DETAILS and ORDER_HEADER are available in a dimension. Second, it allows report authors to stitch together queries on Sales Fact and Returns Fact in a single report, by using it as a conformed dimension.

Create this virtual dimension as follows:

- Create an Order Codes model query subject containing ORDER_DETAIL_CODE from ORDER_DETAILS and ORDER_NUMBER and from ORDER_HEADER.
- Form relationships from Order Codes to both Sales Fact and Returns Fact on the order detail code.
- Test the use of Order Codes as a conformed dimension. First, do a Test on Sales Fact>ORDER_NUMBER, Sales Fact>Quantity, and Returns Fact>Return Quantity. Then compare the results to a Test on Order Codes>Order Number, Sales Fact>Quantity, and Returns Fact>Return Quantity.
- Compare the SQL between the two tests.

For more detailed information outlined as tasks, see the Task Table on the next page.

To see the desired queries and model query subject contents, see the Workshop Results section that follows the Task Table.

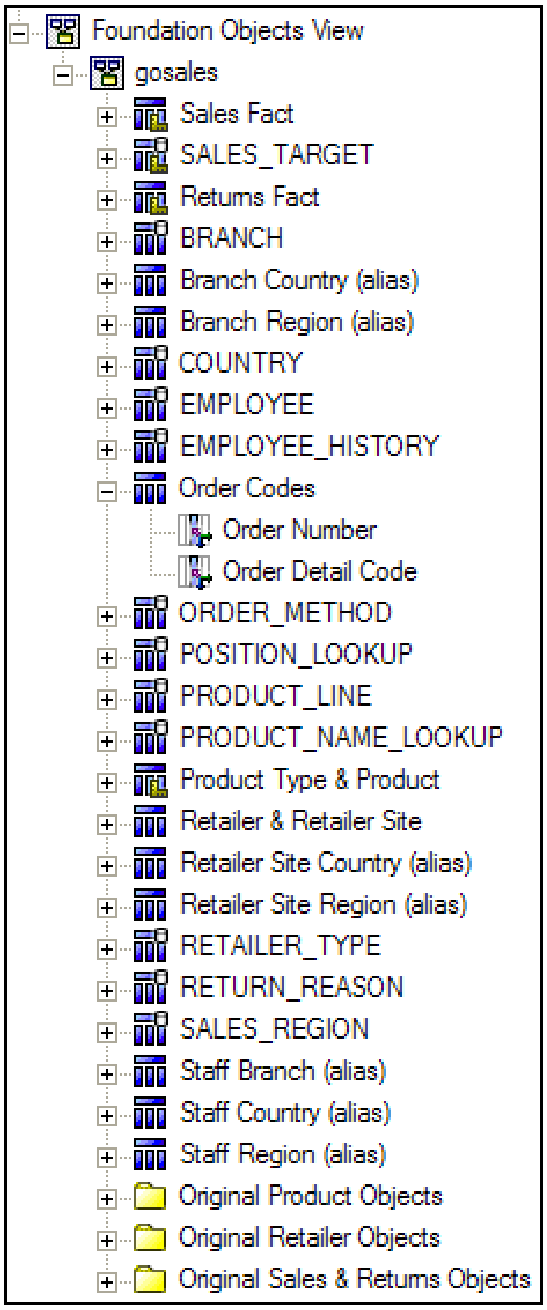
Workshop 1: Task Table

Task	Where to Work	Hints
1. Create a new model query subject.	Project Viewer	• Right-click gosales, Create > Query Subject • Name it Order Codes • From the Original Sales & Returns Objects folder, add: ORDER_HEADER> ORDER_NUMBER, and ORDER_DETAILS> ORDER_DETAIL_CODE. • Rename the query items to Order Number and Order Detail Code. • Drag Order Codes to just below EMPLOYEE_HISTORY to keep alphabetical order
2. Create relationships for Order Codes.	Project Viewer, Relationship Definition dialog box	• Reply No if asked to replicate the existing underlying relationships • Order Codes (Order Detail Code, 1..1) to Sales Fact (ORDER_DETAIL_CODE, 1..n) • Order Codes (Order Detail Code, 1..1) to Returns Fact (ORDER_DETAIL_CODE, 1..n)

3. Test Order Codes as a conformed dimension.	Project Viewer, Test Results	• Do a Test, with Auto Sum, on: Sales Fact >ORDER_NUMBER Sales Fact >Quantity, and Returns Fact >Return Quantity • View the SQL • Do a Test, with Auto Sum, on: Order Codes>Order Number Sales Fact >Quantity, and Returns Fact>Return Quantity • View the SQL

Workshop 1: Workshop Results

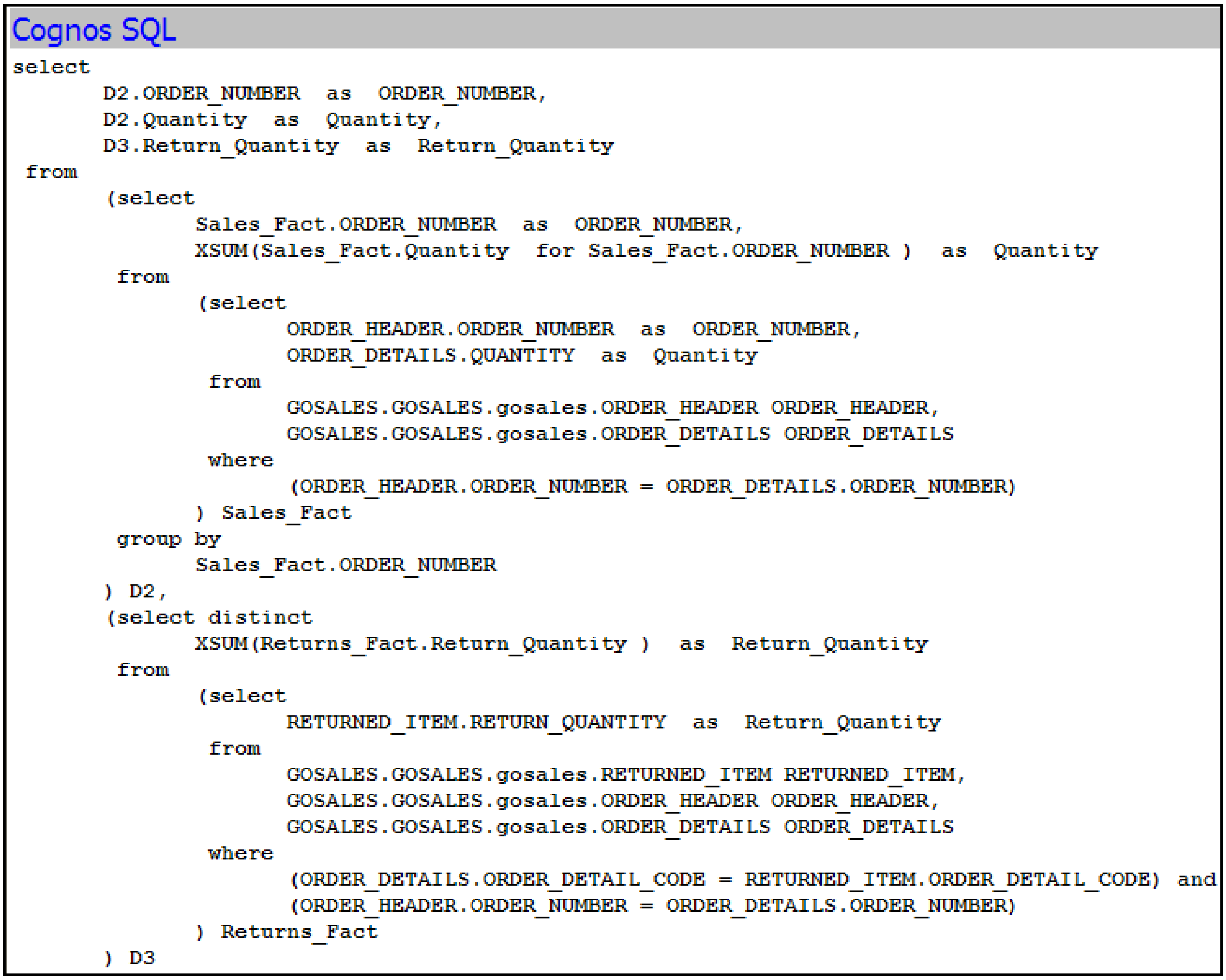
Your Foundation Objects View should appear as shown below:



Foundation Objects View
gosales
Sales Fact
SALES_TARGET
Returns Fact
BRANCH
Branch Country (alias)
Branch Region (alias)
COUNTRY
EMPLOYEE
EMPLOYEE_HISTORY
Order Codes
Order Number
Order Detail Code
ORDER_METHOD
POSITION_LOOKUP
PRODUCT_LINE
PRODUCT_NAME_LOOKUP
Product Type & Product
Retailer & Retailer Site
Retailer Site Country (alias)
Retailer Site Region (alias)
RETAILER_TYPE
RETURN_REASON
SALES_REGION
Staff Branch (alias)
Staff Country (alias)
Staff Region (alias)
Original Product Objects
Original Retailer Objects
Original Sales & Returns Objects

Workshop 1: Workshop Results

SQL for the first test without Order Codes appears as shown below:



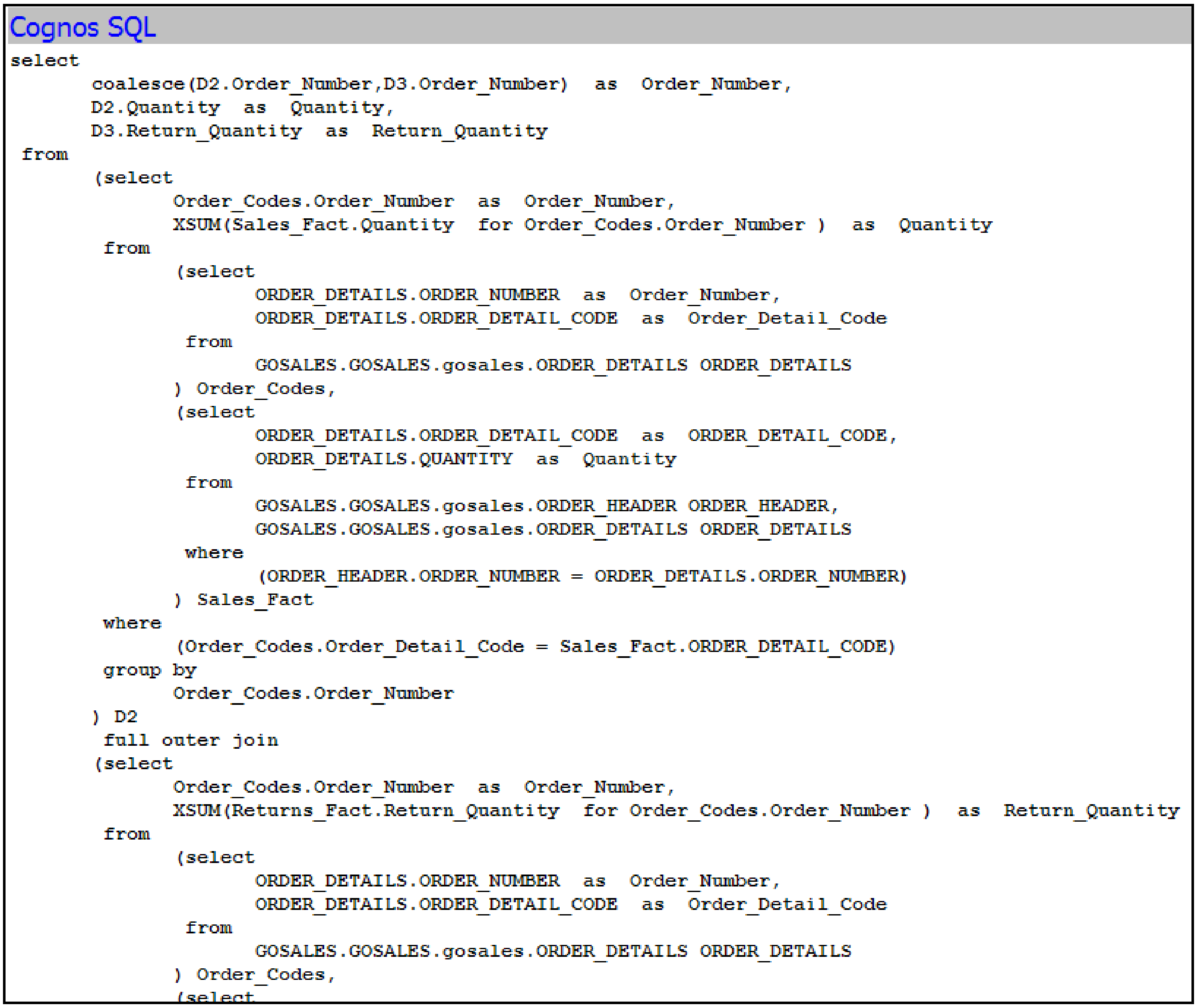
Cognos SQL

```
select
       D2.ORDER_NUMBER  as  ORDER_NUMBER,
       D2.Quantity  as  Quantity,
       D3.Return_Quantity  as  Return_Quantity
 from
       (select
              Sales_Fact.ORDER_NUMBER  as  ORDER_NUMBER,
              XSUM(Sales_Fact.Quantity  for Sales_Fact.ORDER_NUMBER )  as  Quantity
        from
              (select
                     ORDER_HEADER.ORDER_NUMBER  as  ORDER_NUMBER,
                     ORDER_DETAILS.QUANTITY  as  Quantity
               from
                     GOSALES.GOSALES.gosales.ORDER_HEADER ORDER_HEADER,
                     GOSALES.GOSALES.gosales.ORDER_DETAILS ORDER_DETAILS
               where
                     (ORDER_HEADER.ORDER_NUMBER = ORDER_DETAILS.ORDER_NUMBER)
              ) Sales_Fact
        group by
              Sales_Fact.ORDER_NUMBER
       ) D2,
       (select distinct
              XSUM(Returns_Fact.Return_Quantity )  as  Return_Quantity
        from
              (select
                     RETURNED_ITEM.RETURN_QUANTITY  as  Return_Quantity
               from
                     GOSALES.GOSALES.gosales.RETURNED_ITEM RETURNED_ITEM,
                     GOSALES.GOSALES.gosales.ORDER_HEADER ORDER_HEADER,
                     GOSALES.GOSALES.gosales.ORDER_DETAILS ORDER_DETAILS
               where
                     (ORDER_DETAILS.ORDER_DETAIL_CODE = RETURNED_ITEM.ORDER_DETAIL_CODE) and
                     (ORDER_HEADER.ORDER_NUMBER = ORDER_DETAILS.ORDER_NUMBER)
              ) Returns_Fact
       ) D3
```

No stitch query is performed and Return Quantity returns only one overall value for each record.

Workshop 1: Workshop Results

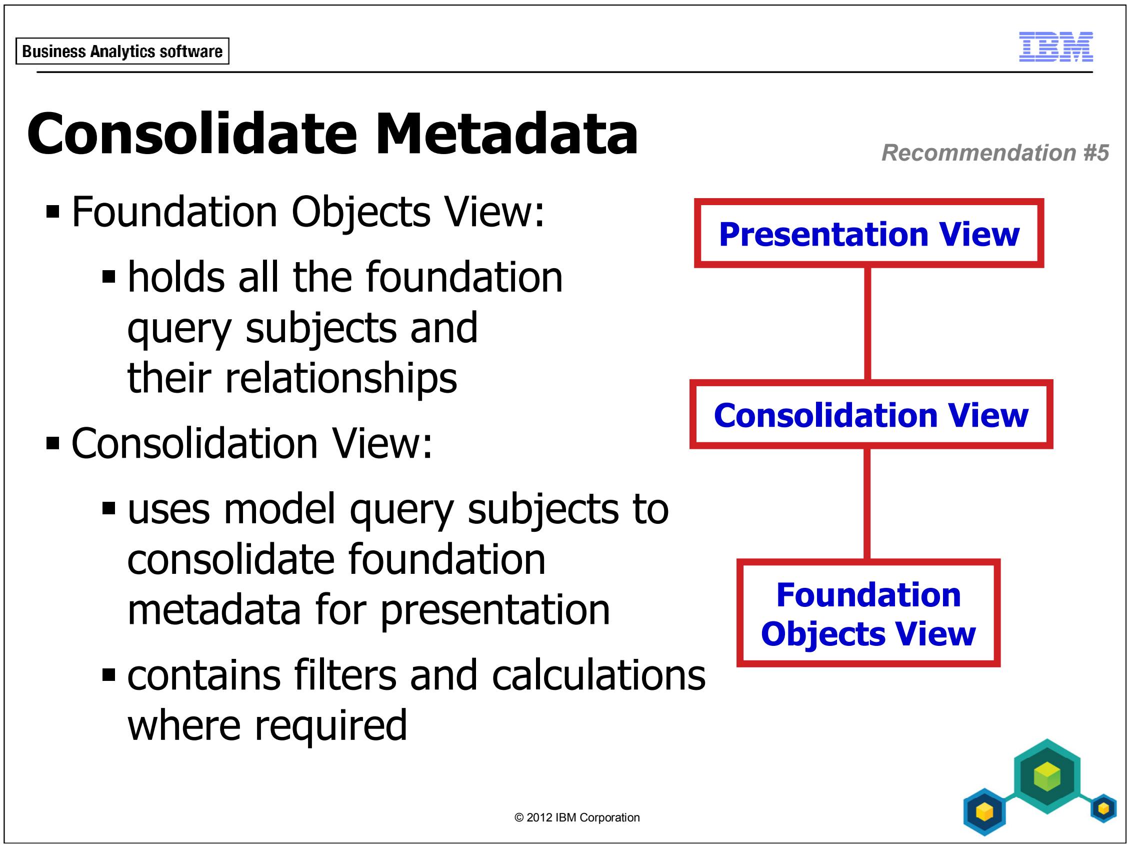
SQL for the second test with Order Codes appears as shown below:



Cognos SQL

```
select
       coalesce(D2.Order_Number,D3.Order_Number)  as  Order_Number,
       D2.Quantity  as  Quantity,
       D3.Return_Quantity  as  Return_Quantity
 from
       (select
              Order_Codes.Order_Number  as  Order_Number,
              XSUM(Sales_Fact.Quantity  for Order_Codes.Order_Number )  as  Quantity
        from
              (select
                     ORDER_DETAILS.ORDER_NUMBER  as  Order_Number,
                     ORDER_DETAILS.ORDER_DETAIL_CODE  as  Order_Detail_Code
               from
                     GOSALES.GOSALES.gosales.ORDER_DETAILS ORDER_DETAILS
              ) Order_Codes,
              (select
                     ORDER_DETAILS.ORDER_DETAIL_CODE  as  ORDER_DETAIL_CODE,
                     ORDER_DETAILS.QUANTITY  as  Quantity
               from
                     GOSALES.GOSALES.gosales.ORDER_HEADER ORDER_HEADER,
                     GOSALES.GOSALES.gosales.ORDER_DETAILS ORDER_DETAILS
               where
                     (ORDER_HEADER.ORDER_NUMBER = ORDER_DETAILS.ORDER_NUMBER)
              ) Sales_Fact
        where
              (Order_Codes.Order_Detail_Code = Sales_Fact.ORDER_DETAIL_CODE)
        group by
              Order_Codes.Order_Number
       ) D2
        full outer join
       (select
              Order_Codes.Order_Number  as  Order_Number,
              XSUM(Returns_Fact.Return_Quantity  for Order_Codes.Order_Number )  as  Return_Quantity
        from
              (select
                     ORDER_DETAILS.ORDER_NUMBER  as  Order_Number,
                     ORDER_DETAILS.ORDER_DETAIL_CODE  as  Order_Detail_Code
               from
                     GOSALES.GOSALES.gosales.ORDER_DETAILS ORDER_DETAILS
              ) Order_Codes,
              (select
```

The correct data is returned and the SQL shows a stitch query is performed to merge the two fact result sets together.



Business Analytics software

Consolidate Metadata

Recommendation #5

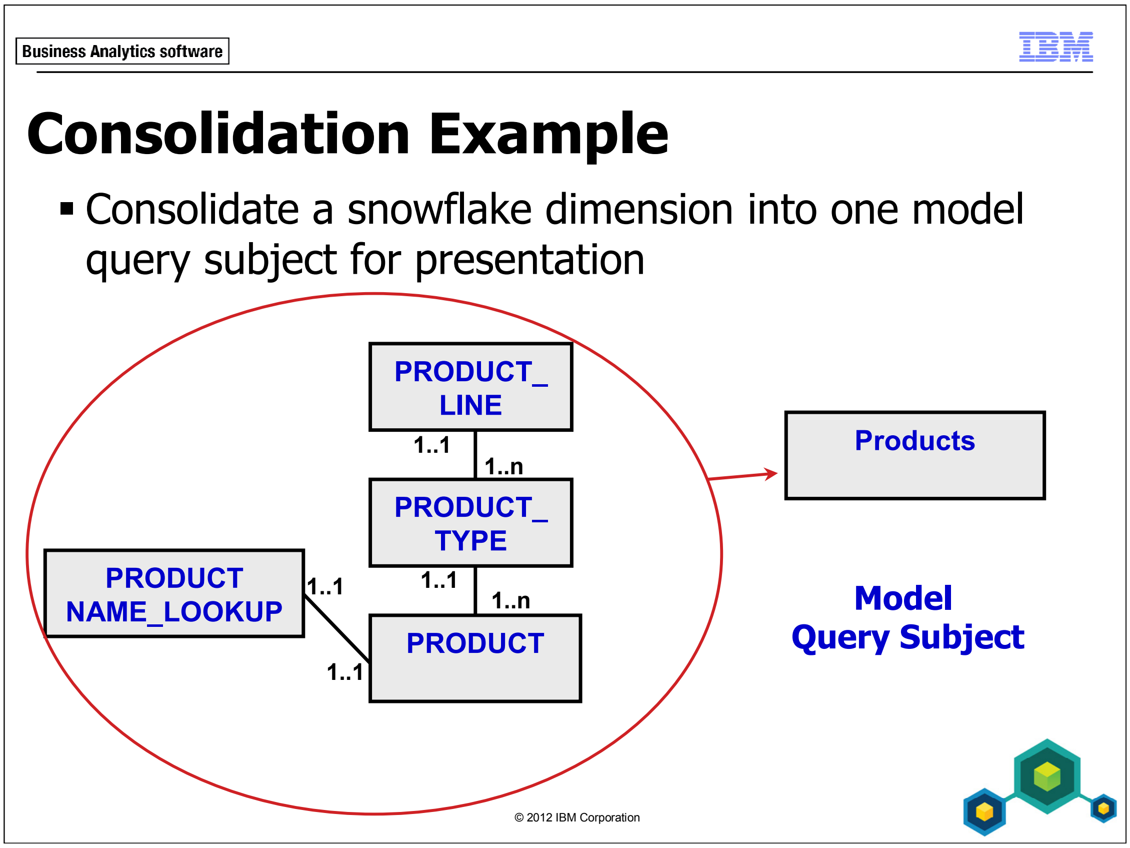
- Foundation Objects View:
  - holds all the foundation query subjects and their relationships
- Consolidation View:
  - uses model query subjects to consolidate foundation metadata for presentation
  - contains filters and calculations where required

Presentation View
Consolidation View
Foundation Objects View

© 2012 IBM Corporation

The Consolidation View provides a layer of metadata that is separate from the foundation layer that contains all the relationships. To understand query paths, view the Foundation Objects View. To view how metadata will be presented to authors, view objects in the Consolidation View.

Consolidating the metadata involves tasks such as consolidating snowflake dimensions into one model query subject, or hiding codes found in fact query subjects, which are used for relationships.



Business Analytics software

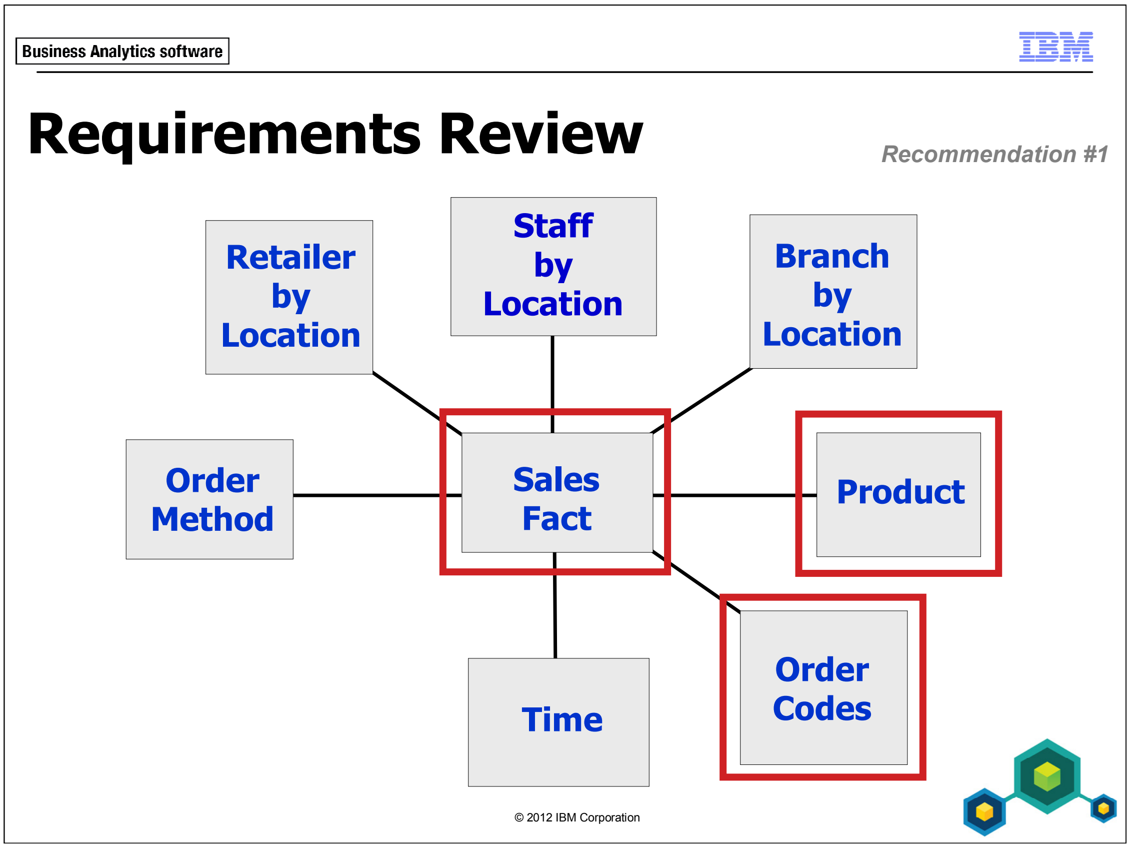
Consolidation Example

- Consolidate a snowflake dimension into one model query subject for presentation

© 2012 IBM Corporation

An example of consolidating snowflake dimensions is placing query items from PRODUCT_LINE, PRODUCT_TYPE, PRODUCT, and PRODUCT_NAME_LOOKUP into a model query subject called Products.

You can also use the Consolidation View to neatly present your facts. For example you can create a fact query subject that only contains measures. The codes used in relationships found in the underlying Foundation Objects View are not visible to report authors.


Business Analytics software

Requirements Review

Recommendation #1

© 2012 IBM Corporation

In the next demo, you will consolidate product information into one model query subject to create a simplified Product dimension that will be presented to users.

You will also create a simplified model query subject for Sales Fact and the new conformed dimension you created in the last workshop called Order Codes. The Order Codes dimension is shared between Sales Fact and Returns Fact.

Demo 2: Create the Consolidation View

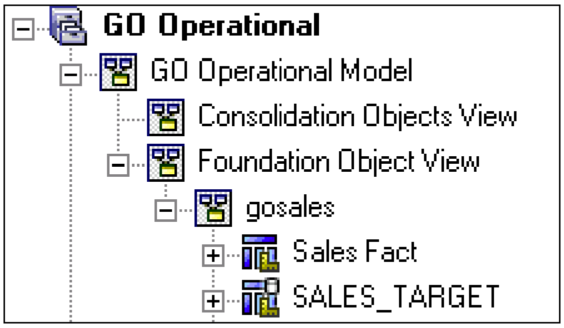
Purpose:
To simplify the metadata view for authors, you will consolidate related query items into single query subjects. Authors would like access to codes but to reduce clutter you will organize them into query item folders.

Components:	**Framework Manager, Cognos Workspace Advanced**
Project:	**GO Operational**
Package:	**Consolidation View Test**

Task 1. Create a new namespace to contain consolidated objects.

1. In the **Project Viewer** pane, in the **GO Operational Model** namespace, create a namespace called **Consolidation View**.
2. Drag the new namespace above **Foundation Objects View**.

   The results appear as follows:



Task 2. Consolidate a snowflake dimension.

1. In the **Consolidation View** namespace, create a new model query subject called **Products**, and then click **OK**.
2. In the **Query Subject Definition**, expand **Foundation Objects View>gosales**, and then add the following query items to the definition:

Query Subject	Query Item
PRODUCT_LINE	**PRODUCT_LINE_CODE** **PRODUCT_LINE_EN**
Product Type & Product	**PRODUCT_TYPE_CODE** **PRODUCT_TYPE_EN**
PRODUCT_NAME_LOOKUP	**PRODUCT_NAME** **PRODUCT_DESCRIPTION**
Product Type & Product	**PRODUCT_NUMBER** **PRODUCT_IMAGE** **INTRODUCTION_DATE** **DISCONTINUED_DATE** **PRODUCTION_COST** **GROSS_MARGIN**

   Note: You can also create this model query subject by selecting the individual query subjects (or the query items within them) and then merging them into a new model query subject.

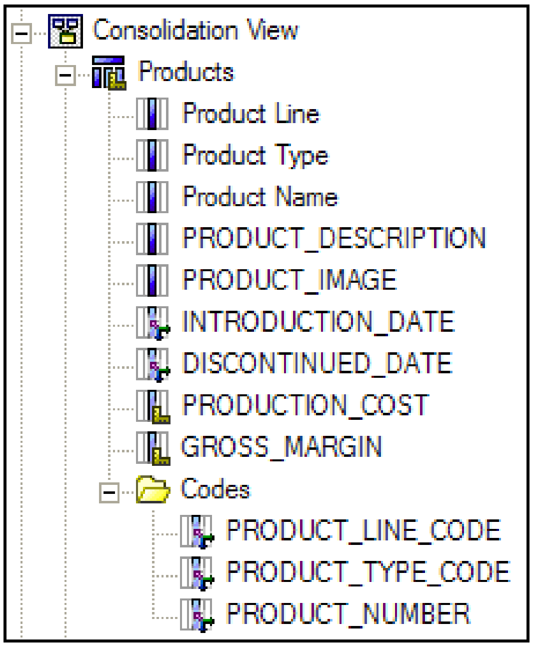
3. Click **OK**, right-click **Products**, point to **Create**, and then click **Query Item Folder**.
4. Rename the folder to **Codes**.

5. Drag the following items into the folder:
   - **PRODUCT_LINE_CODE**
   - **PRODUCT_TYPE_CODE**
   - **PRODUCT_NUMBER**

   Authors still need these query items for activities like filtering and drill-through scenarios, so moving them into a subfolder is a good way to centralize such codes. It also organizes the metadata and makes the main dimensional query items easier to locate.

6. Rename the first three query items in **Products** as follows:
   - **Product Line**
   - **Product Type**
   - **Product Name**

   The results appear as follows:



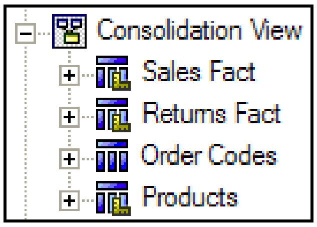
```
Consolidation View
  Products
    Product Line
    Product Type
    Product Name
    PRODUCT_DESCRIPTION
    PRODUCT_IMAGE
    INTRODUCTION_DATE
    DISCONTINUED_DATE
    PRODUCTION_COST
    GROSS_MARGIN
    Codes
      PRODUCT_LINE_CODE
      PRODUCT_TYPE_CODE
      PRODUCT_NUMBER
```

   You would normally rename all items, but in order to save time items will be renamed for you in the starting point project at the beginning of the next module.

Task 3. Create simplified query subjects for presentation.

1. In the **Consolidation View** namespace, create a new model query subject called **Sales Fact** containing the following query items from **Sales Fact** in the **Foundation Objects View**:
   - **Quantity**
   - **Unit Cost**
   - **Unit Price**
   - **Unit Sale Price**
2. Create a new model query subject called **Returns Fact** containing the following query item from **Returns Fact** in the **Foundation Objects View**:
   - **Return Quantity**
3. Create a new model query subject called **Order Codes** containing the following query items from **Order Codes** in the **Foundation Objects View**:
   - **Order Number**
   - **Order Detail Code**
4. In the **Consolidation View** namespace, drag **Products** below **Order Codes**.

   The results appear as follows:



5. Save the project.

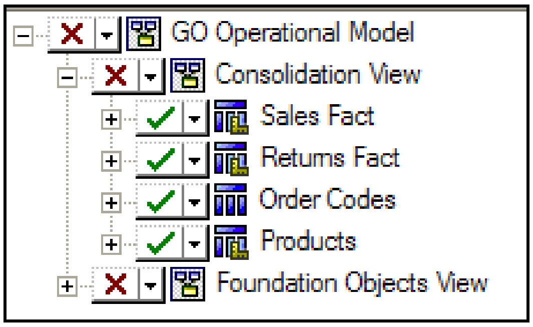
   You now have a consolidated view of the data that authors can report from without having to know the underlying relationships or table complexity.

Task 3 Step 3: You create a new model query subject for Order Codes instead of simply creating a shortcut to the Order Codes in Foundation Objects View. This is partly to give us a consistent view in this layer, but it is also because shortcuts cannot be used to create star schema groupings, which you will later need to do to populate the Presentation View.

Task 4. Publish a package of the Consolidation View.

1. Right-click **Packages**, point to **Create**, and click **Package**.
2. Name the package **Consolidation View Test**, and then click **Next**.
3. Clear **GO Operational Model**, expand **Consolidation View**, and select the four model query subjects.

   The results appear as follows:

4. Click **Finish**, and then click **Yes** to open the **Publish Package** wizard.
5. Clear **Enable Model Versioning**, and then click **Next** twice.
6. Click **Publish**, and then click **Finish**.

   You are presented with an information list indicating that the objects you published refer to objects not included in the package. In order for the package to work, these items will be included in the package but hidden from the users.

7. Click **Close**.

Task 5. Test the Package.

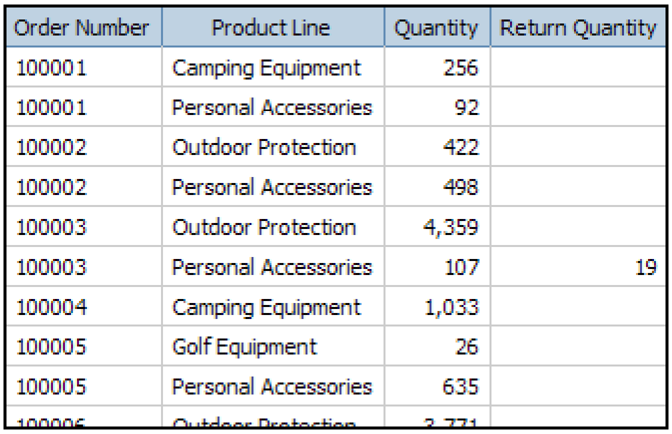
1. Launch **IBM Cognos Connection**, log in, and open **Cognos Workspace Advanced** selecting the **Consolidation View Test** package for a **List** report.

   Note that only the selected query subjects appear in the data tree.

2. Create a query with the following query items:

Query Subject	Query Item
Order Codes	Order Number
Products	Product Line
Sales Fact	Quantity
Returns Fact	Return Quantity

The results appear as follows:



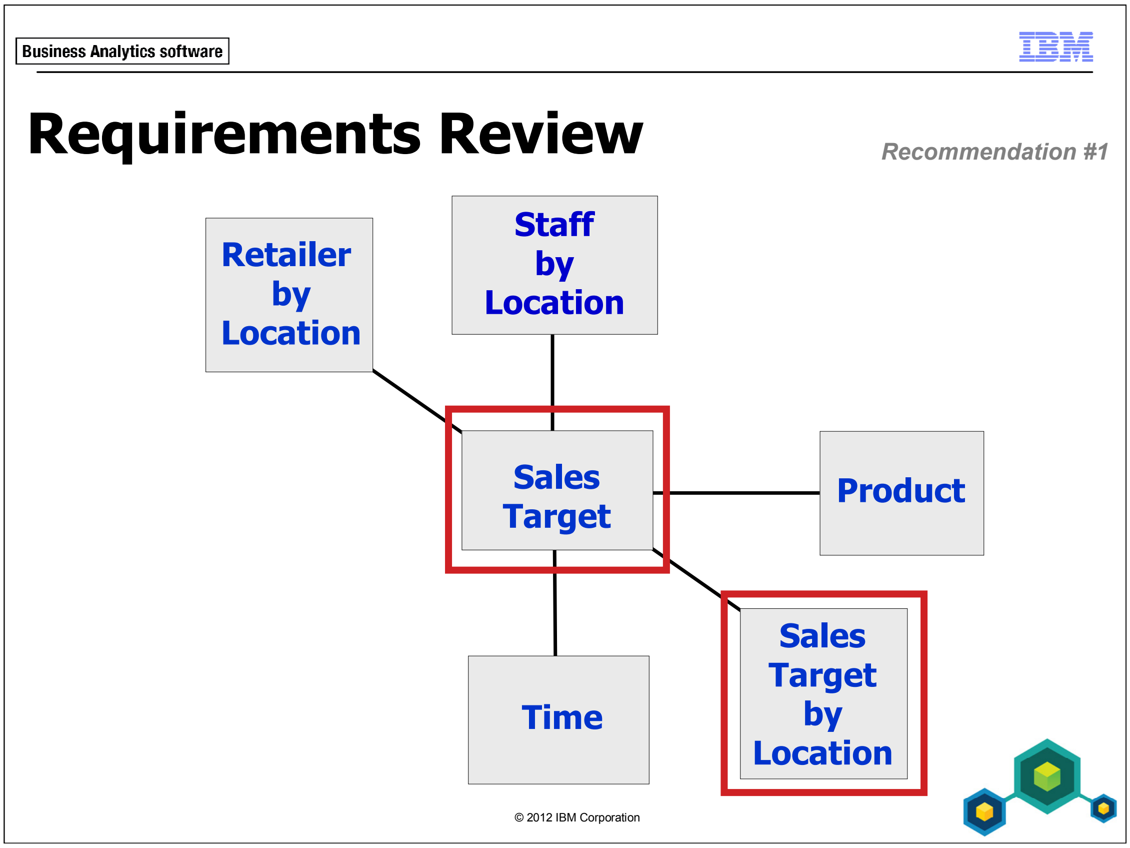
Order Number	Product Line	Quantity	Return Quantity
100001	Camping Equipment	256	
100001	Personal Accessories	92	
100002	Outdoor Protection	422	
100002	Personal Accessories	498	
100003	Outdoor Protection	4,359	
100003	Personal Accessories	107	19
100004	Camping Equipment	1,033	
100005	Golf Equipment	26	
100005	Personal Accessories	635	
100006	Outdoor Protection	3,771	

You have successfully created a report from a simplified and author-friendly model that hides the underlying complexity of the metadata.

3. Close the browser without saving the query.

Results:
You have created consolidated and simplified query subjects to meet reporting needs, and you have tested the resulting package in Query Studio.

Product Name does not yet have a filter that only retrieves the local session's language. Therefore adding it to the report would return unexpected results.



Business Analytics software

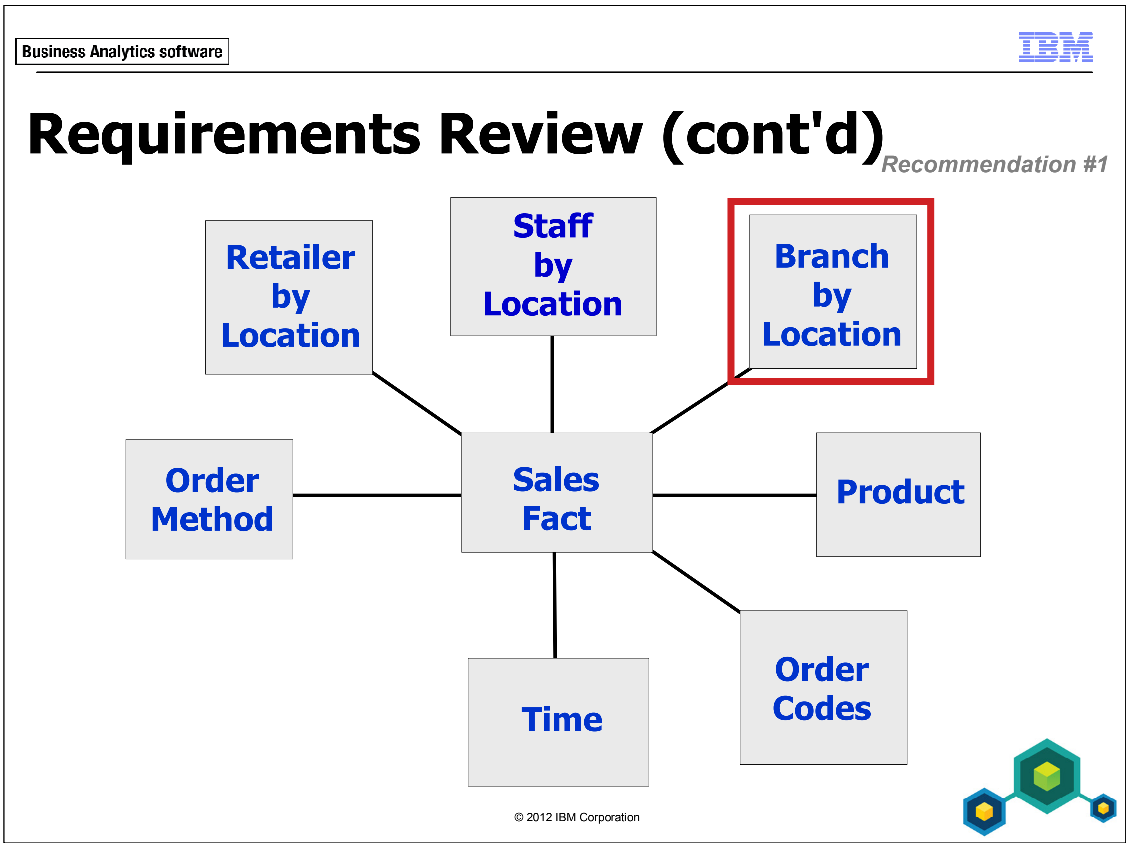
Requirements Review

Recommendation #1

© 2012 IBM Corporation

During the modeling process, you discovered another important dimension that can be used to report against Sales Target. In the next workshop, you will create a consolidated model query subject to allow users to report on sales targets by location.

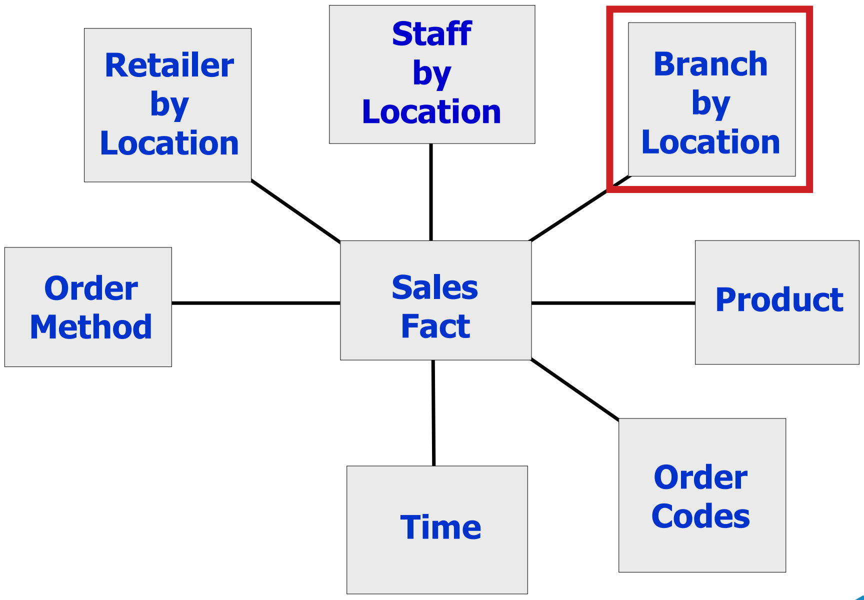
You will also create a simplified view of Sales Target.



Business Analytics software

Requirements Review (cont'd)

Recommendation #1



© 2012 IBM Corporation

During the modeling process, you also discovered an alternate and important dimension that can be used to report against Sales Fact. In the next workshop, you will create a consolidated model query subject called Branch by Location.

Workshop 2: Consolidate and Simplify the Model for Presentation

You will continue to consolidate related query items into single query subjects to simplify the presentation of metadata for authors.

Create consolidated and simplified query subjects in the Consolidation View to meet these needs, as follows:

- Create a consolidated Sales Target by Location dimension from SALES_REGION and COUNTRY as shown in the Workshop Results section that follows the Task Table.
- Create a consolidated Branch by Location dimension from Branch Region (alias), Branch Country (alias), and BRANCH as shown in the Workshop Results section that follows the Task Table.
- Create a simplified Sales Target Fact containing just gosales.SALES_TARGET> SALES_TARGET and renamed to mixed case.
- Save the project and close Framework Manager when you are finished.

For more detailed information outlined as tasks, see the Task Table on the next page.

To see the desired queries and model query subject contents, see the Workshop Results section that follows the Task Table.

Workshop 2: Task Table

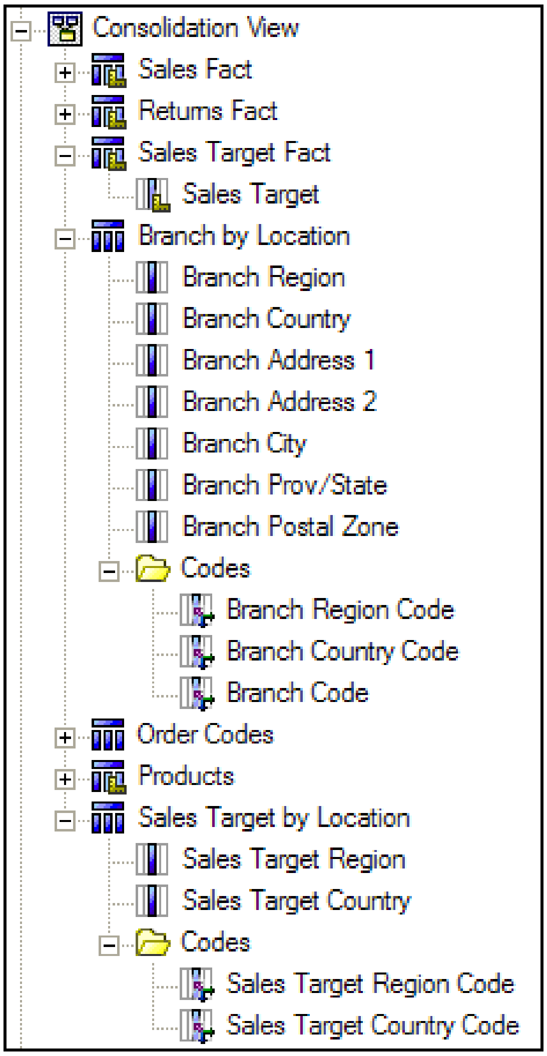
Task	Where to Work	Hints
1. Create the consolidated Sales Target by Location query subject.	Project Viewer, Query Subject Definition dialog box.	• SALES_REGION: SALES_REGION_CODE SALES_REGION_EN • COUNTRY: COUNTRY _CODE COUNTRY _EN • Rename query items to Sales Target Region Code, Sales Target Region, Sales Target Country Code, and Sales Target Country • Place Sales Target Region Code and Sales Target Country Code in a new Codes query item folder

Task	Where to Work	Hints
2. Create the consolidated Branch by Location query subject	Project Viewer, Query Subject Definition dialog box.	• Branch Region (alias): Branch Region Code Branch Region • Branch Country (alias): Branch Country Code Branch Country • BRANCH: BRANCH_CODE ADDRESS1 ADDRESS2 CITY PROV_STATE POSTAL_ZONE • Place the three codes in a new Codes query item folder. • Rename all query items from BRANCH to mixed case with a 'Branch' prefix (for example, Branch Address1)

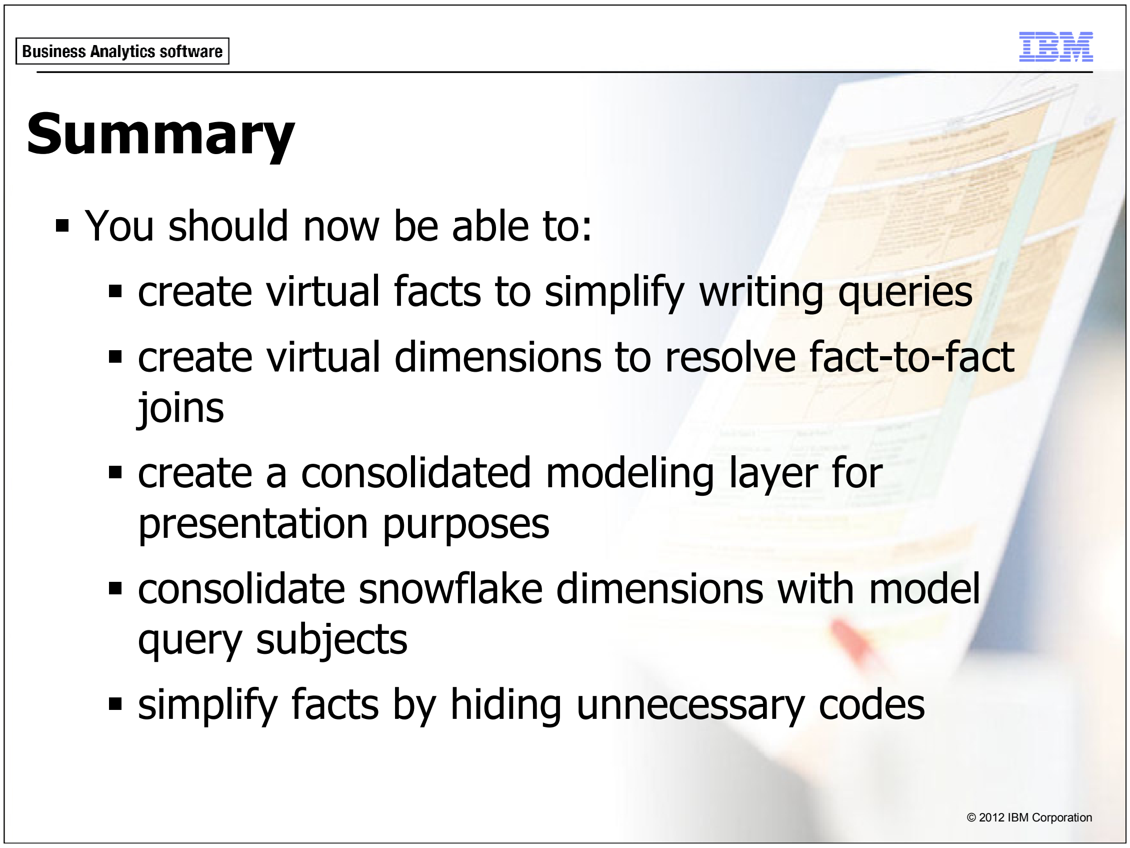
3. Create a simplified Sales Target Fact.	Project Viewer, Query Subject Definition dialog box.	• SALES_TARGET > SALES_TARGET • Rename query item to Sales Target • Drag Sales Target Fact to just below Returns Fact • Save and close Framework Manager

Workshop 2: Workshop Results

Your Consolidation View should appear as shown below:



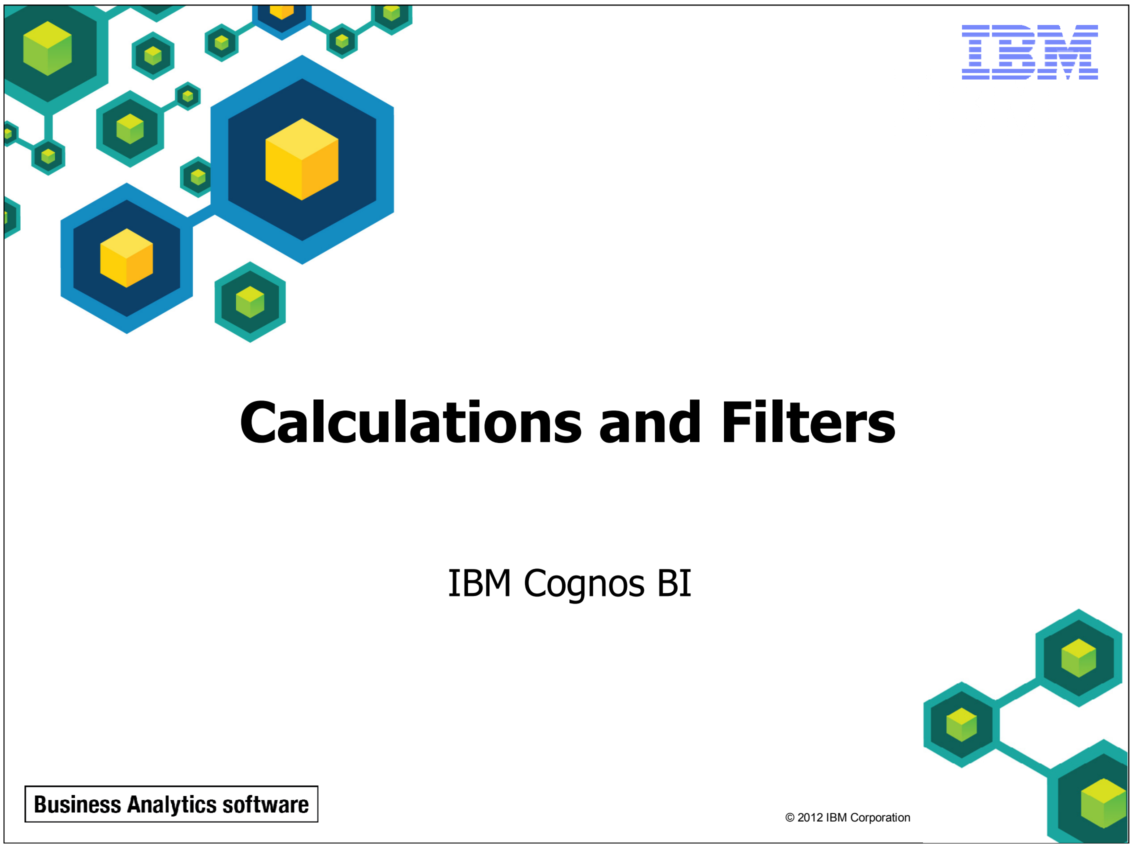
```
Consolidation View
    Sales Fact
    Returns Fact
    Sales Target Fact
        Sales Target
    Branch by Location
        Branch Region
        Branch Country
        Branch Address 1
        Branch Address 2
        Branch City
        Branch Prov/State
        Branch Postal Zone
        Codes
            Branch Region Code
            Branch Country Code
            Branch Code
    Order Codes
    Products
    Sales Target by Location
        Sales Target Region
        Sales Target Country
        Codes
            Sales Target Region Code
            Sales Target Country Code
```



Business Analytics software

Summary

- You should now be able to:
  - create virtual facts to simplify writing queries
  - create virtual dimensions to resolve fact-to-fact joins
  - create a consolidated modeling layer for presentation purposes
  - consolidate snowflake dimensions with model query subjects
  - simplify facts by hiding unnecessary codes



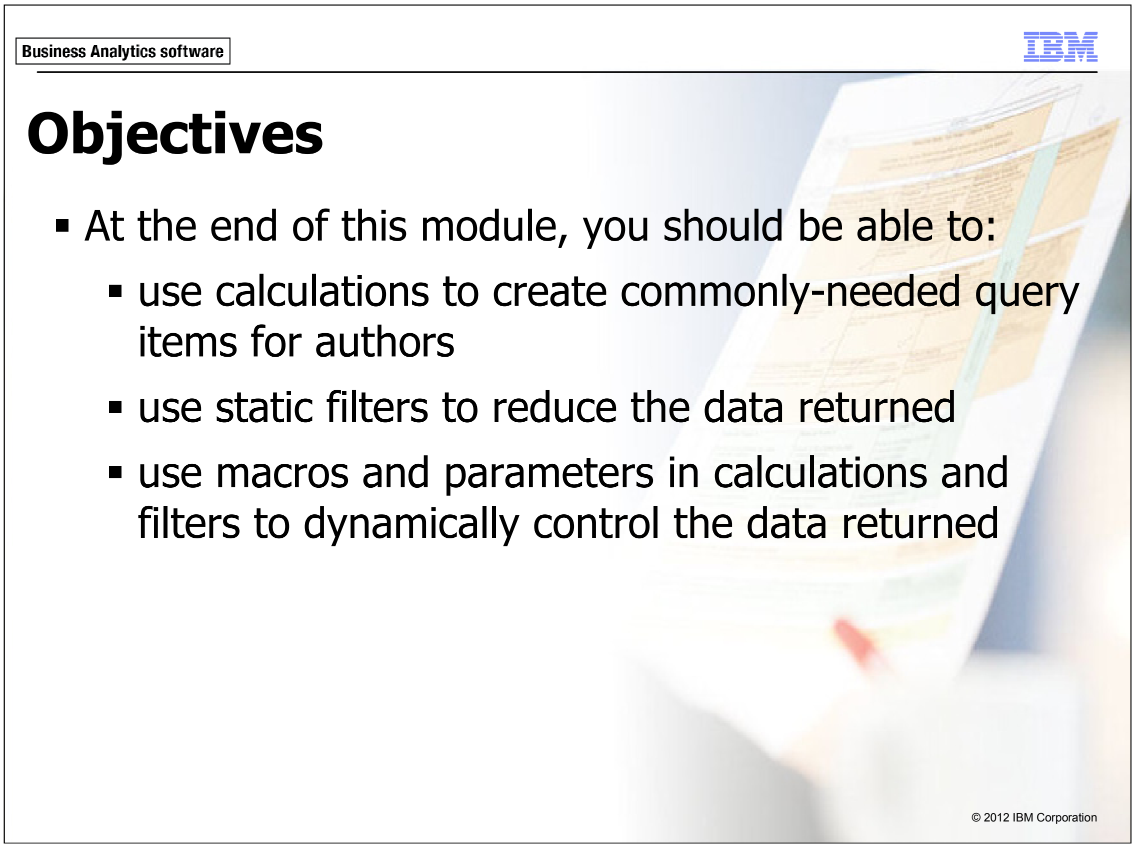
Calculations and Filters

IBM Cognos BI

Business Analytics software

Unless otherwise specified in demo or workshop steps, you will always log on to IBM Cognos in the Local LDAP namespace using the following credentials:

• User ID: admin
• Password: Education1



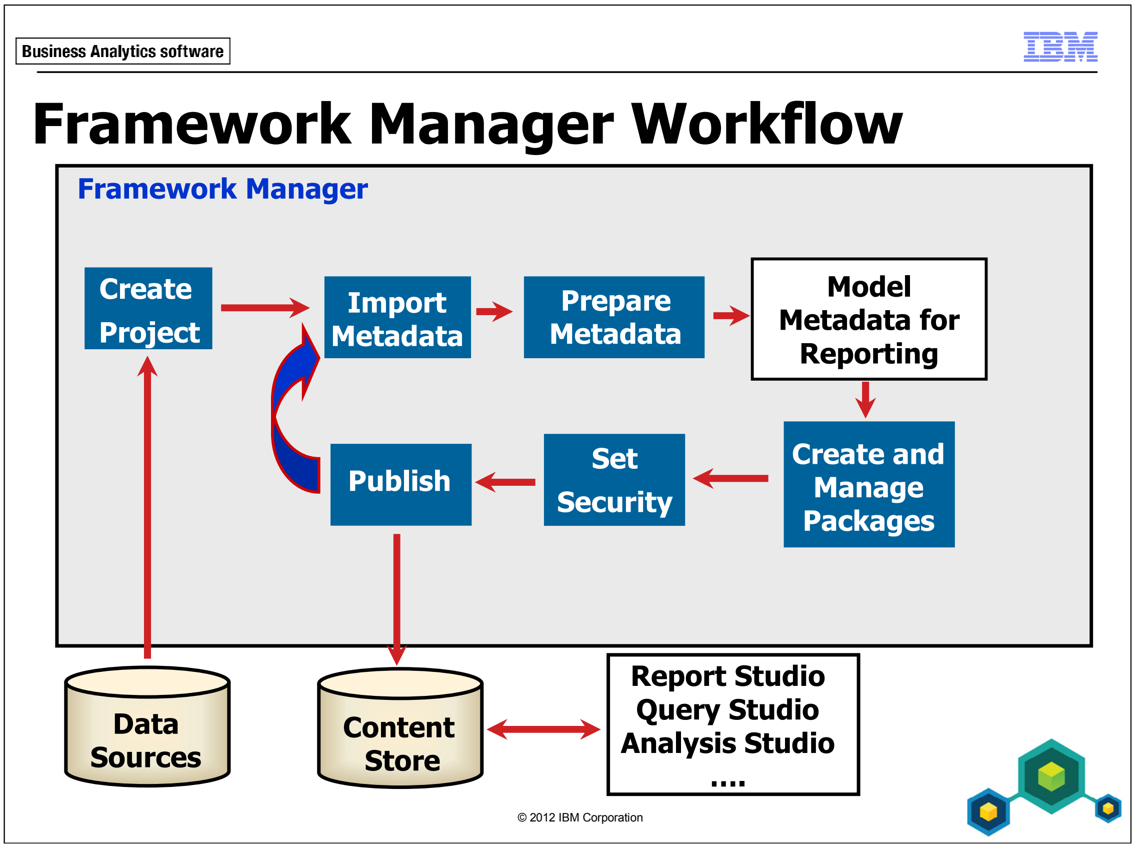
Business Analytics software

Objectives

- At the end of this module, you should be able to:
  - use calculations to create commonly-needed query items for authors
  - use static filters to reduce the data returned
  - use macros and parameters in calculations and filters to dynamically control the data returned

© 2012 IBM Corporation

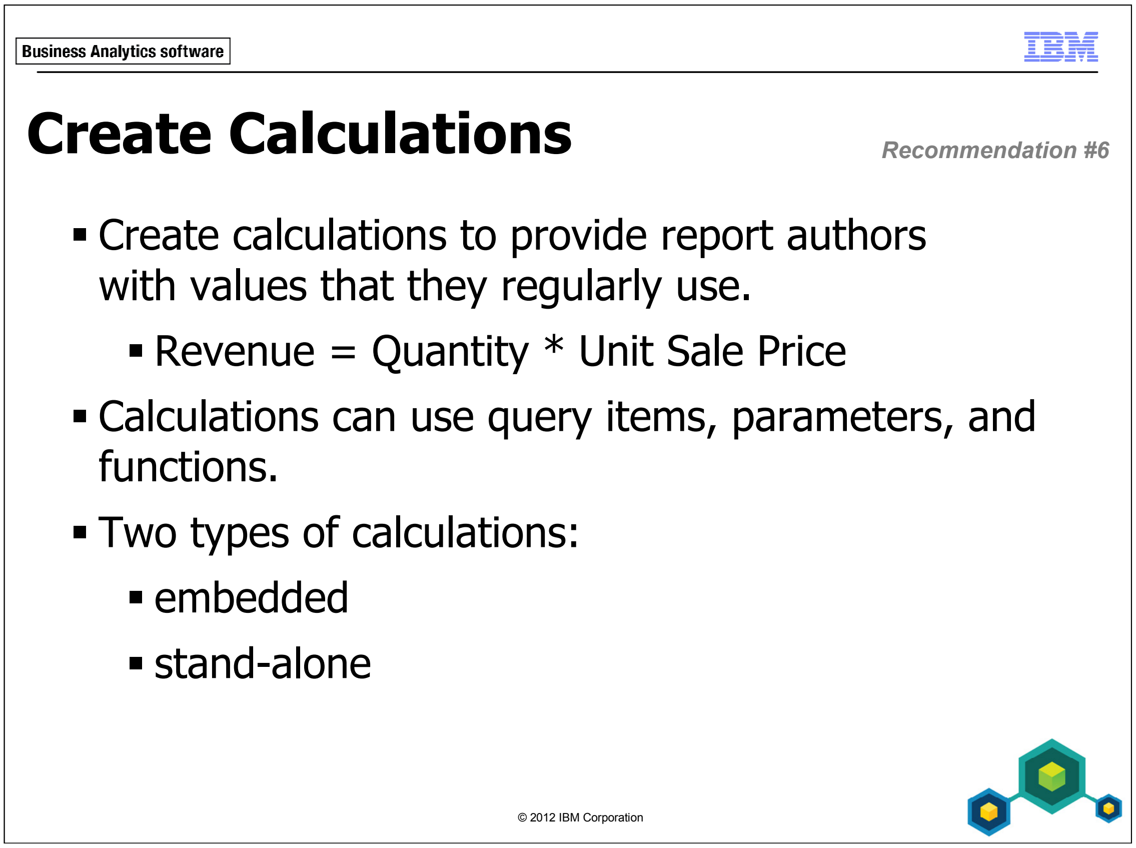
Before reviewing this module, you should be familiar with IBM Cognos, IBM Cognos Connection, Query Studio and Report Studio. Suggested modules to reference:
• Overview of IBM Cognos
• Identify Common Data Structures
• Gather Requirements
• Create a Baseline Project
• Prepare Reusable Metadata
• Model for Predictable Results: Reporting Issues
• Model for Predictable Results: Virtual Star Schemas
• Model for Predictable Results: Consolidate Metadata


Business Analytics software

Framework Manager Workflow

© 2012 IBM Corporation

This module deals with creating calculations and filters to add business logic to the model.



Business Analytics software

IBM

Create Calculations

Recommendation #6

- Create calculations to provide report authors with values that they regularly use.
  - Revenue = Quantity * Unit Sale Price
- Calculations can use query items, parameters, and functions.
- Two types of calculations:
  - embedded
  - stand-alone

© 2012 IBM Corporation

If you want to create a calculation specifically for one query subject or dimension, you can embed the calculation directly in that object. For query subjects, this calculation can be done for either data source query subjects or model query subjects. However, it is recommended that you apply calculations in model query subjects wherever possible. This allows for better maintenance and change management.

Create a stand-alone calculation when you want to apply the calculation to more than one query subject or dimension. Stand-alone calculations are also valuable if you need to do aggregation before performing the calculation (as shown in Appendix A). You can also create stand-alone calculations as an alternate way to present information rather than in a query subject or dimension. However, stand-alone calculations are not visible in Analysis Studio.

If you start with an embedded calculation, you can later convert it into a stand-alone calculation that you can apply to other query subjects.

Demo 1: Create Embedded Calculations

Purpose:
Report authors want to include the revenue, gross profit, and margin of each order in their reports. Therefore, you will create three embedded calculations in the Sales Fact query subject. This will produce three new query items.

Component:	**Framework Manager**
Project:	**GO Operational**

Task 1. Create the calculations.

1. In **Framework Manager**, open the **GO Operational** model located at **C:\Edcognos\B5252\CBIFM-Start Files\Module 9\GO Operational**.
2. Expand **Foundation Objects View>gosales**, and then double-click **Sales Fact**.

   The calculations you are about to create could also be placed in the Consolidation View Sales Fact model query subject. However, placing them in the lower level increases opportunities for reusability.
3. Click **Add** in the bottom right corner.
4. In the **Name** box, type **Revenue**, and then, in the **Available Components** pane, double-click **Quantity**.
5. Click the **Functions** tab, expand **Operators**, and then double-click the multiplication operator (*****).

   Notice there is a description of the function in the Tips pane.

6. Under **Available Components**, click the **Model** tab, and then double-click **Unit Sale Price**.

   The results appear as follows:

   [gosales].[Sales Fact].[Quantity] * [gosales].[Sales Fact].[Unit Sale Price]

7. Click **Test Sample** ▶ to verify that the calculation works, and then click **OK**.

8. Create a **Gross Profit** calculation with the following expression:

   [gosales].[Sales Fact].[Revenue] - ([gosales].[Sales Fact].[Quantity] * [gosales].[Sales Fact].[Unit Cost])

   Where Revenue is the calculation you just created.

9. Create a **Margin** calculation with the following expression:

   [gosales].[Sales Fact].[Gross Profit] /[gosales].[Sales Fact].[Revenue]

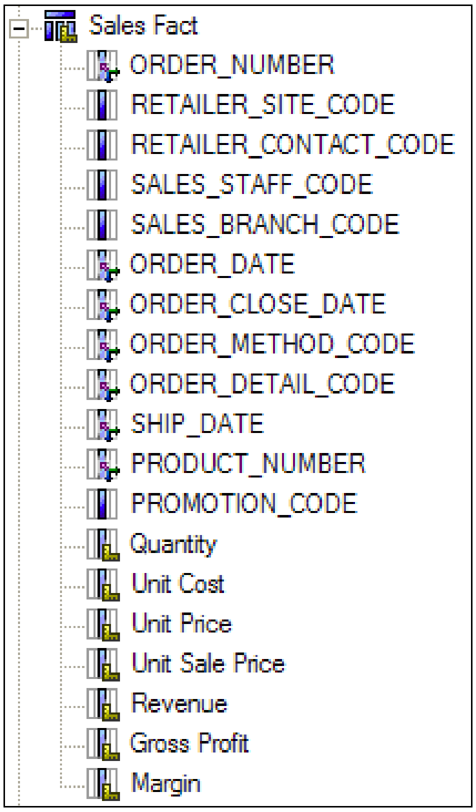
   Where Gross Profit and Revenue are the calculations you just created.

   Currently the new calculations appear as attributes because they have not been evaluated yet.

10. Click **Validate**, and then click **OK.**

    The query subject and its new calculations are evaluated.

    The results appear as follows:



```
Sales Fact
    ORDER_NUMBER
    RETAILER_SITE_CODE
    RETAILER_CONTACT_CODE
    SALES_STAFF_CODE
    SALES_BRANCH_CODE
    ORDER_DATE
    ORDER_CLOSE_DATE
    ORDER_METHOD_CODE
    ORDER_DETAIL_CODE
    SHIP_DATE
    PRODUCT_NUMBER
    PROMOTION_CODE
    Quantity
    Unit Cost
    Unit Price
    Unit Sale Price
    Revenue
    Gross Profit
    Margin
```

    The new calculations (Revenue, Gross Profit, and Margin) now appear as facts.

Task 2. Set formatting properties on query items in the Foundation Objects View.

1. Under **Sales Fact**, select **Revenue** and **Gross Profit**.
2. If necessary, from the **View** menu, click **Properties**.
3. In the **Properties** pane, for the **Revenue** query item, set the **Format type** to **Currency**, and the **Currency** property to **$ (USD) - United States of America, dollar**.
4. Apply the same formatting to the **Gross Profit** query item by dragging down the small black arrow.
5. Format **Margin** as a percentage.

Task 3. Update the Consolidation View Sales Fact query subject and test.

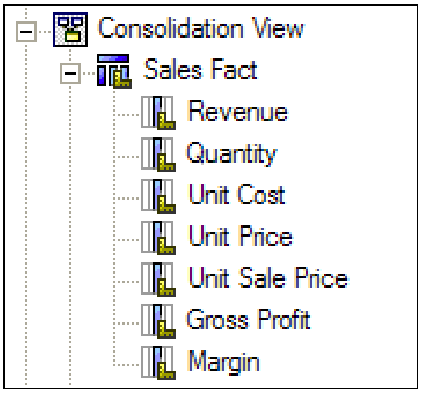
1. In the **Consolidation View**, double-click **Sales Fact**.
2. Add the three new calculations from **Foundation Objects View > gosales > Sales Fact**.
3. Click **Revenue** and then click **Top** to move it to the top of the list.
4. Click the **Test** tab, and then click **Test Sample**.

   Notice that Margin = Gross Profit / Revenue, as expected.

Task 3 Step 4: Caution: If you auto sum you will not get expected results for the Margin calculation. This is because the calculation is performed first and then aggregated (in this case summed). For this type of calculation, we would want to aggregate first and then perform the calculation.This can only be done with a stand alone calculation. This technique is covered towards the end of the course in the Employ Additional Modeling Techniques module..

5. Click **OK**.

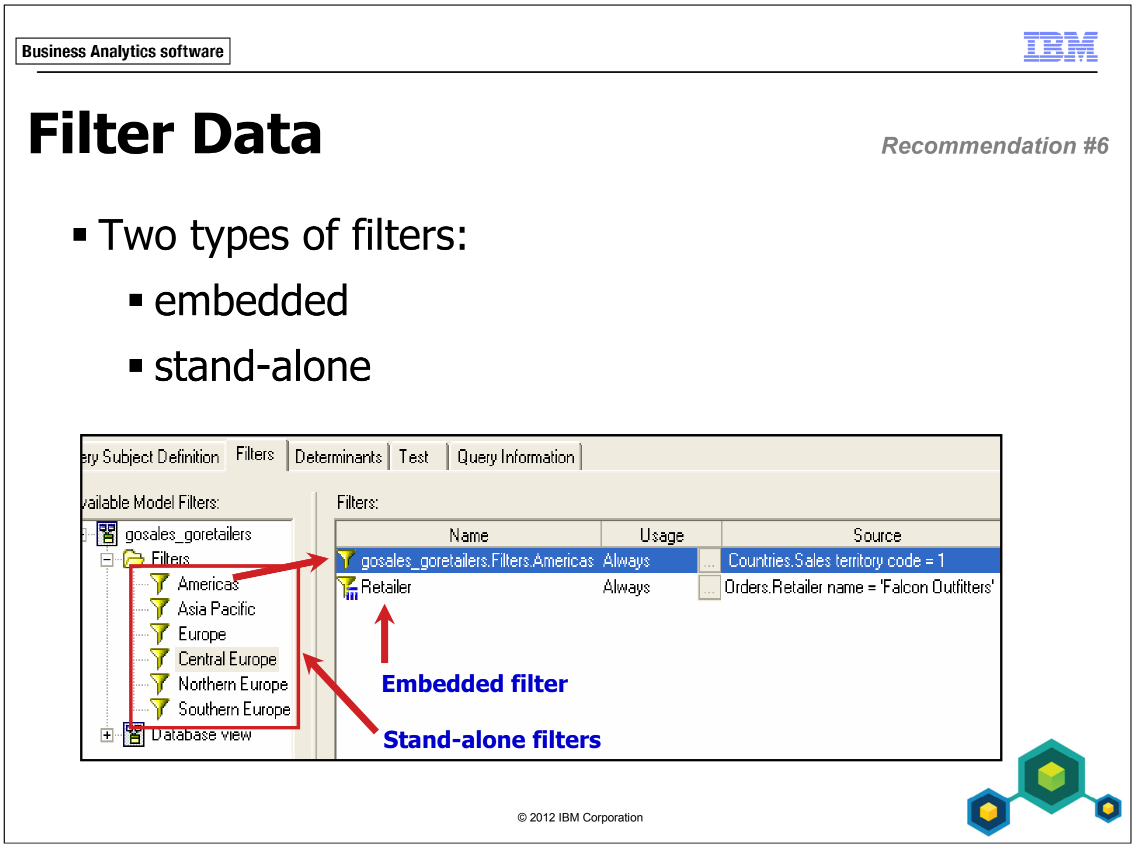
   The results appear as follows in the Project Viewer:



Consolidation View
Sales Fact
Revenue
Quantity
Unit Cost
Unit Price
Unit Sale Price
Gross Profit
Margin

6. Save the project.

Results:
You created and tested three embedded calculations within the Sales Fact query subject. This produced three new query items that report authors can exploit.


Business Analytics software

IBM

Filter Data

Recommendation #6

- Two types of filters:
  - embedded
  - stand-alone

ery Subject Definition | Filters | Determinants | Test | Query Information

vailable Model Filters:

- gosales_goretailers
  - Filters
    - Americas
    - Asia Pacific
    - Europe
    - Central Europe
    - Northern Europe
    - Southern Europe
  - Database view

Filters:

Name	Usage	Source
gosales_goretailers.Filters.Americas	Always	Countries.Sales territory code = 1
Retailer	Always	Orders.Retailer name = 'Falcon Outfitters'

© 2012 IBM Corporation

Embedded filters are appropriate when the filter is intended for just one query subject or dimension. Stand-alone filters are appropriate when required in multiple query subjects or dimensions, or to make commonly used filters readily available for authors.

Filters have a Usage setting with the following options:

- Always - the filter will always be applied, regardless of whether the filtered query item is in the query or not
- Optional - users may choose to enter a filter value or leave it blank (only applies to filters that use a prompt value or macro)
- Design Mode Only - Limits the amount of data that is retrieved when testing in Framework Manager or when authoring reports.

You can also restrict the data that a query retrieves by adding a WHERE clause to the SQL definition, or by setting governors. Governors are discussed in the Optimize and Tune Framework Manager Models module.

Demo 2: Create Embedded and Standalone Filters

Purpose:
Employee data includes historical job information that is beyond the scope of the reporting application you are building. So you will restrict EMPLOYEE_HISTORY to each employee's current record.

Report authors also want the ability to easily restrict retailer data to specific regions. To do this, you will create standalone filters that authors can apply during report creation.

Components:	**Framework Manager, IBM Cognos Workspace Advanced**
Project:	**GO Operational**
Package:	**GO Operational**

Task 1. Create an embedded filter to retrieve current employee records.

To restrict retrieval to an employee's current record, you can filter on EMPLOYEE_HISTORY.RECORD_END_DATE IS NULL. This signifies that the employee has not yet ended their current position.

You will place the filter on EMPLOYEE_HISTORY in the Foundation Objects View rather than on Staff by Location in the Consolidation View. This is because the filter causes the relationship between EMPLOYEE and EMPLOYEE_HISTORY to change from one-to-many to one-to-one, and all our relationships are in the Foundation Objects View. Changing this cardinality also prevents EMPLOYEE_HISTORY from acting as an ambiguous query subject.

1. In the **Project Viewer** pane, expand **Foundation Objects View>gosales**, and then double-click **EMPLOYEE_HISTORY**.

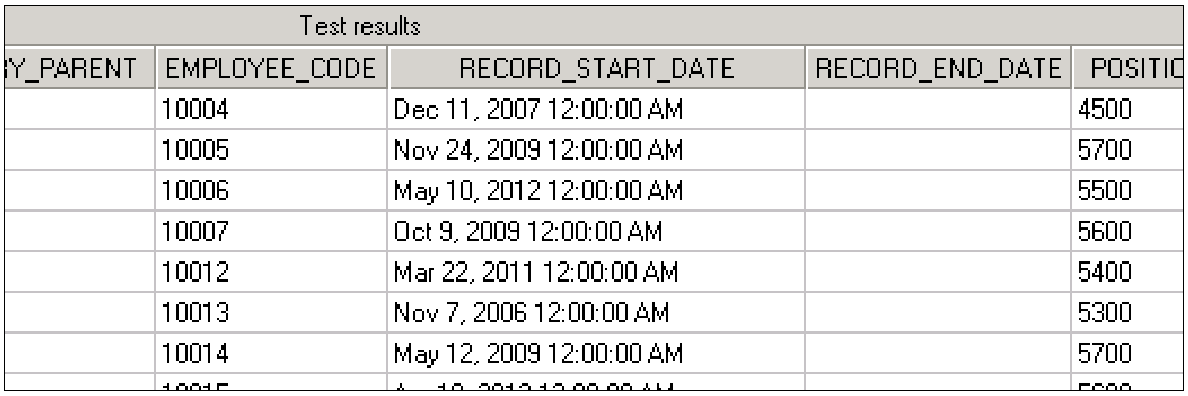
Task 1 Step 1: EMPLOYEE_HISTORY can act as a fact in certain query scenarios because of its cardinalities that can cause unwanted query splits.

2. Click the **Filters** tab, click **Add**, and then in the **Name** box, type **Current Employee History Record**.
3. In the **Available Components** pane, double-click **RECORD_END_DATE**.
4. Click in the **Expression Definition** pane at the end of the expression, and then type **IS NULL**.

   The results appear as follows:

   [gosales].[EMPLOYEE_HISTORY].[RECORD_END_DATE] IS NULL
5. Click **OK**, click the **Test** tab, and then click **Test Sample**.

   The results appear as follows:



Test results

Y_PARENT	EMPLOYEE_CODE	RECORD_START_DATE	RECORD_END_DATE	POSITIC
	10004	Dec 11, 2007 12:00:00 AM		4500
	10005	Nov 24, 2009 12:00:00 AM		5700
	10006	May 10, 2012 12:00:00 AM		5500
	10007	Oct 9, 2009 12:00:00 AM		5600
	10012	Mar 22, 2011 12:00:00 AM		5400
	10013	Nov 7, 2006 12:00:00 AM		5300
	10014	May 12, 2009 12:00:00 AM		5700

   Just one record for every employee is returned and each has a null RECORD_END_DATE.
6. Click **OK**.
7. Right-click **EMPLOYEE_HISTORY** and click **Launch Context Explorer**.
8. Click **Show Related Objects**, and then double-click the relationship between **EMPLOYEE_HISTORY** and **EMPLOYEE**.
9. Change the **EMPLOYEE_HISTORY** side to **1..1**, reflecting the impact of the new filter.
10. Click **OK**, close the **Context Explorer**, and then save the project.

Step 4: Be sure to type IS NULL and not =Null, the difference is subtle, but important.

Task 2. Create Standalone Retailer Location Filters.

1. In **Foundation Objects View>gosales**, right-click **SALES_REGION**, click **Test** and then click **Test Sample**.

   The five regions are as follows:
   710 = Americas
   740 = Asia Pacific
   750 = Northern Europe
   760 = Central Europe
   770 = Southern Europe

2. Click **Close**, and then, in the **Consolidation View**, create a new folder called **Model Filters**.
3. In the new **Model Filters** folder, create a new folder called **Retailer Location Filters**.
4. Right-click **Retailer Location Filters**, point to **Create**, and then click **Filter**.
5. In the **Name** box, type **Americas**.
6. In the **Available Components** pane, under **Consolidation View**, expand **Retailer by Location>Codes**.
7. Add **Retailer Site Region Code** to the **Expression definition** pane, and then type **= 710**.

   The results appear as follows:

   [Consolidation View].[Retailer by Location].[Retailer Site Region Code] = 710
8. Click **OK**.

Task 2 Step 2: We will be creating these filters in the Consolidation View rather than the Foundation Objects View because the filters are intended to be used in the final presentation view.

9. Repeat steps **4** to **8** to create new filters using the appropriate **Retailer Site Region Code**.

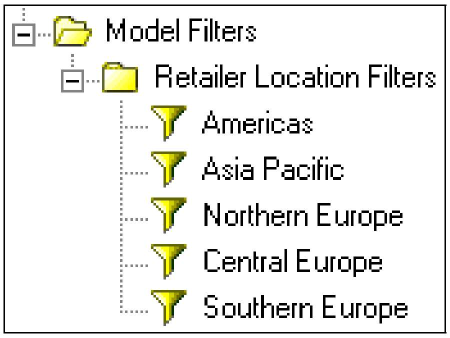
Retailer Site	Retailer Site Region Code
Asia Pacific	**740**
Northern Europe	**750**
Central Europe	**760**
Southern Europe	**770**

Tip: You can also copy the Americas filter and simply edit it to meet your needs.

The expression for Asia Pacific will appear as follows:

[Consolidation View].[Retailer by Location].Retailer Site Region Code] = 740

The results appear as follows:



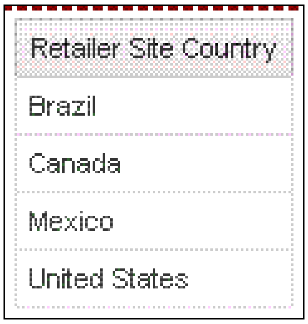
10. Save the project.

Task 3. Test the Standalone Filters.

1. Publish the **GO Operational** package.
2. In **IBM Cognos Connection**, log on, open **Cognos Workspace Advanced**, and then select the **GO Operational** package for a **List** report.

3. In the **Insertable Objects** pane, expand **Consolidation View**>**Retailer by Location**.
4. Add **Retailer Site Country** to the report.
5. Expand **Model Filters**>**Retailer Location Filters**, and then drag **Americas** onto the report.

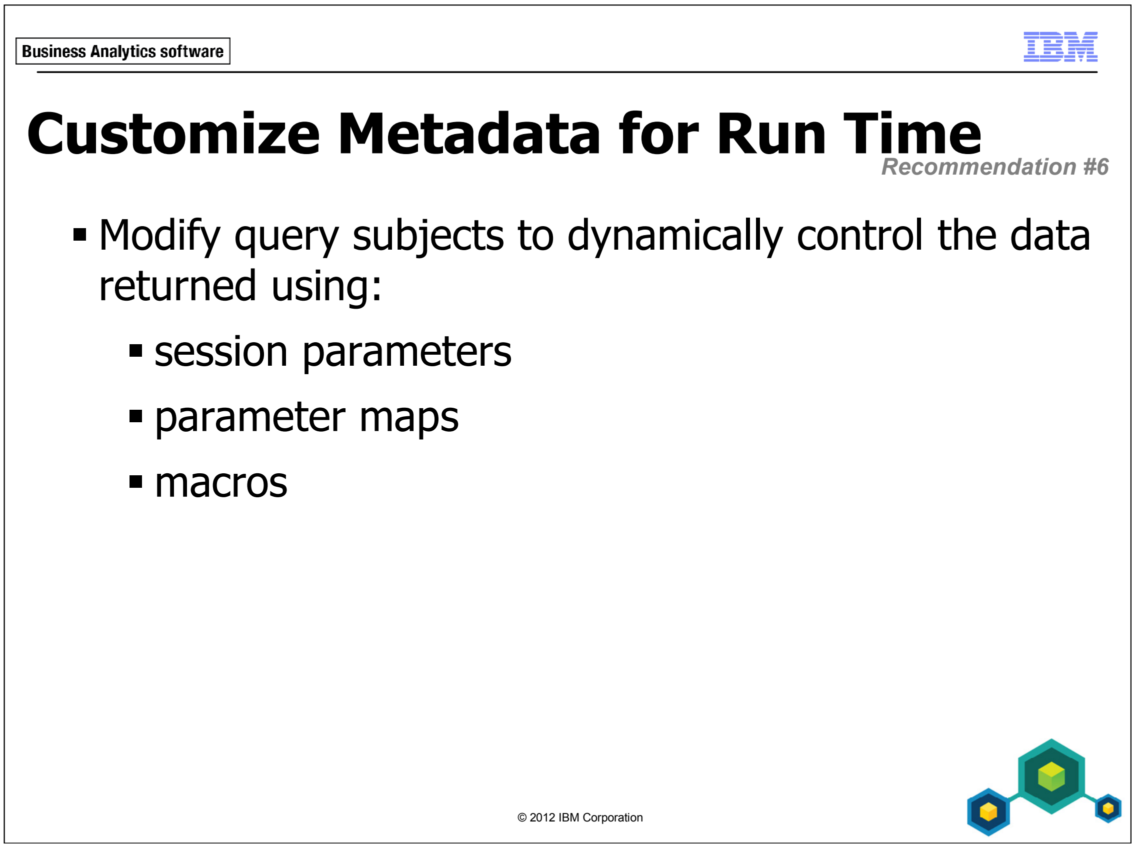
   The results appear as follows:

Retailer Site Country
Brazil
Canada
Mexico
United States

   The report is limited to records for the American continents.
6. Close your browser without saving the report, and leave Framework Manager open for the next demo.

Results:
You embedded a filter into EMPLOYEE_HISTORY to restrict the data to each employee's current record. You also created stand-alone filters for retailer locations that authors can apply during report creation.



Business Analytics software

Customize Metadata for Run Time

Recommendation #6

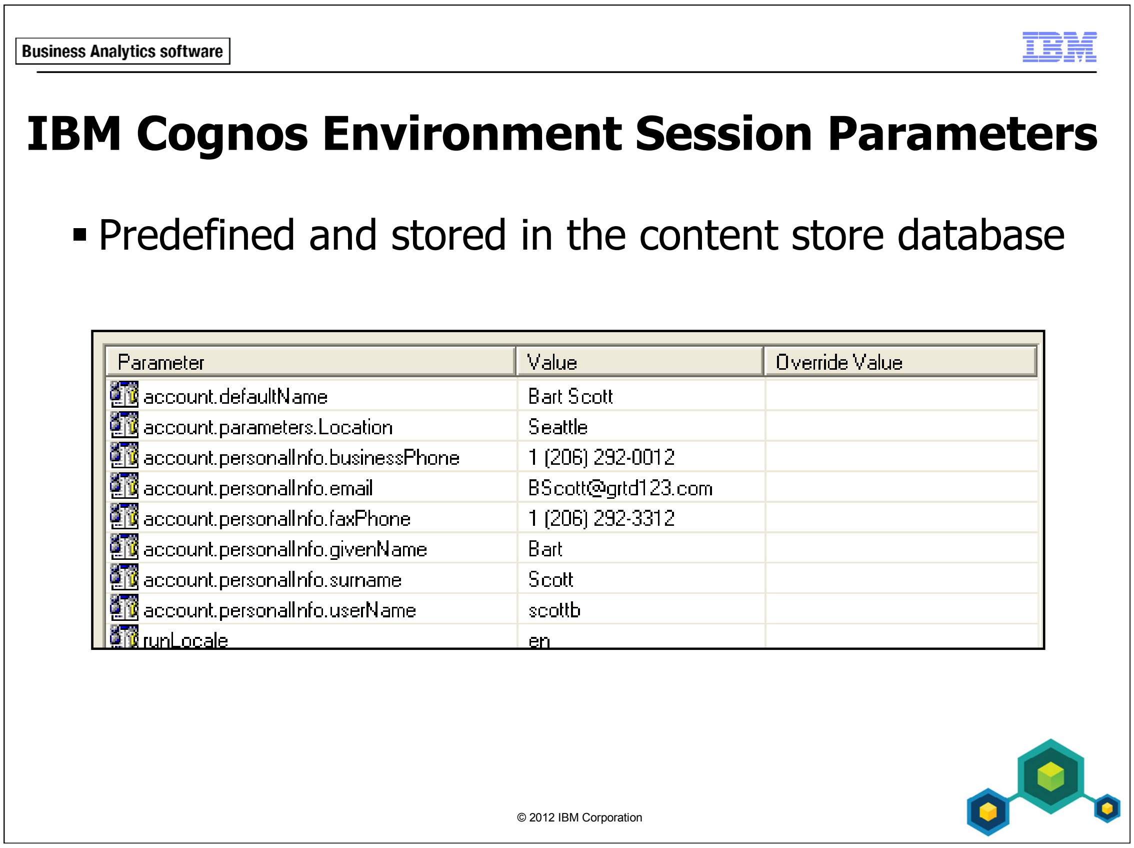
- Modify query subjects to dynamically control the data returned using:
  - session parameters
  - parameter maps
  - macros

© 2012 IBM Corporation

This concept can be used to dynamically return data from specific columns or rows, and even from specific data sources.

One example is to dynamically implement security by using a calculation to retrieve a user's account, group, or role information and then implement row-level security based on values stored in the data source.

Another example might be to retrieve location specific data for a particular user. For example, if a user works in Mexico, you can use a calculation to return sales values for Mexico by querying the Revenue_Mexico column rather than other columns in the table such as Revenue_Canada or Revenue_Germany.



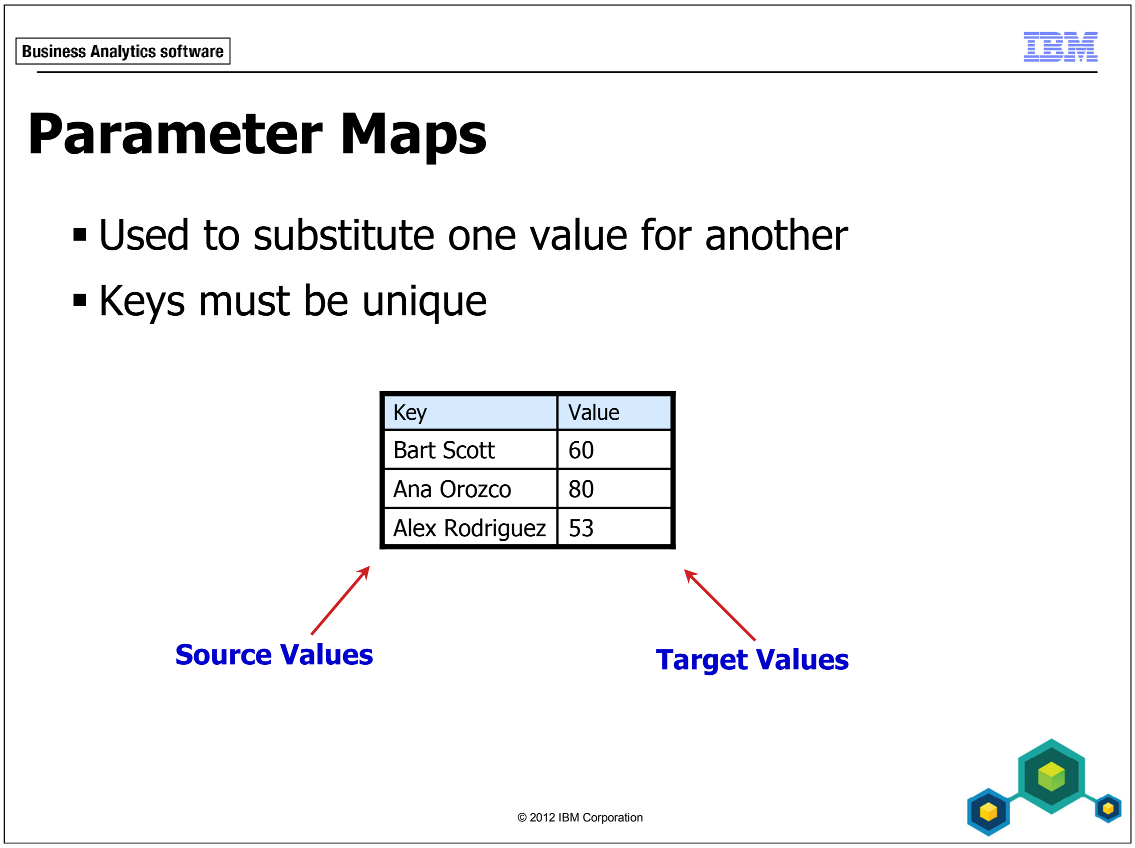
Business Analytics software

IBM Cognos Environment Session Parameters

- Predefined and stored in the content store database

Parameter	Value	Override Value
account.defaultName	Bart Scott	
account.parameters.Location	Seattle	
account.personalInfo.businessPhone	1 (206) 292-0012	
account.personalInfo.email	BScott@grtd123.com	
account.personalInfo.faxPhone	1 (206) 292-3312	
account.personalInfo.givenName	Bart	
account.personalInfo.surname	Scott	
account.personalInfo.userName	scottb	
runLocale	en	

A session parameter returns session information at run time (for example, runLocale or account.UserName).



Business Analytics software

Parameter Maps

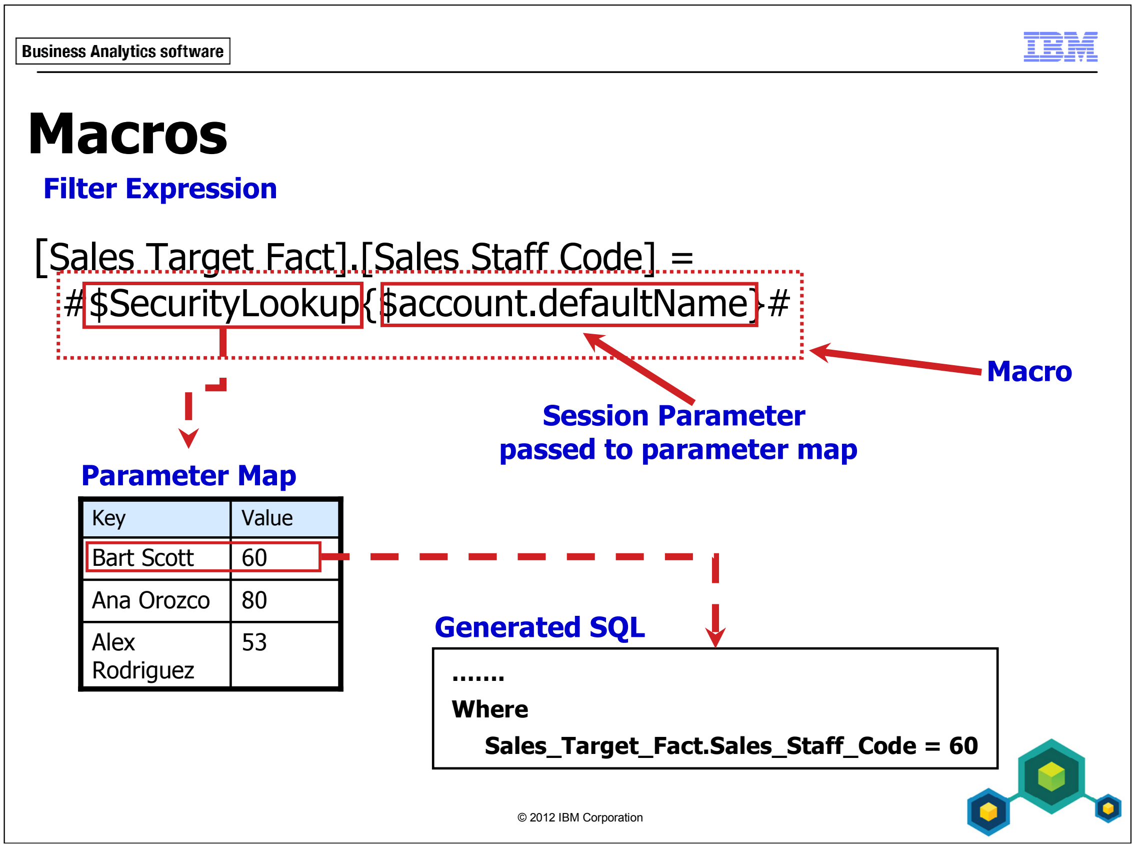
- Used to substitute one value for another
- Keys must be unique

Key	Value
Bart Scott	60
Ana Orozco	80
Alex Rodriguez	53

Source Values
Target Values

A parameter map is a two-column table, mapping a set of keys (source) to a set of substitution values (target).

The above example uses a parameter map to convert the account.DefaultName session parameter to a suitable value in the data, in this case a staff key value.


Business Analytics software
IBM

Macros

Filter Expression

```
[Sales Target Fact].[Sales Staff Code] =
  #$SecurityLookup{$account.defaultName}#
```

Macro
Session Parameter passed to parameter map

Parameter Map

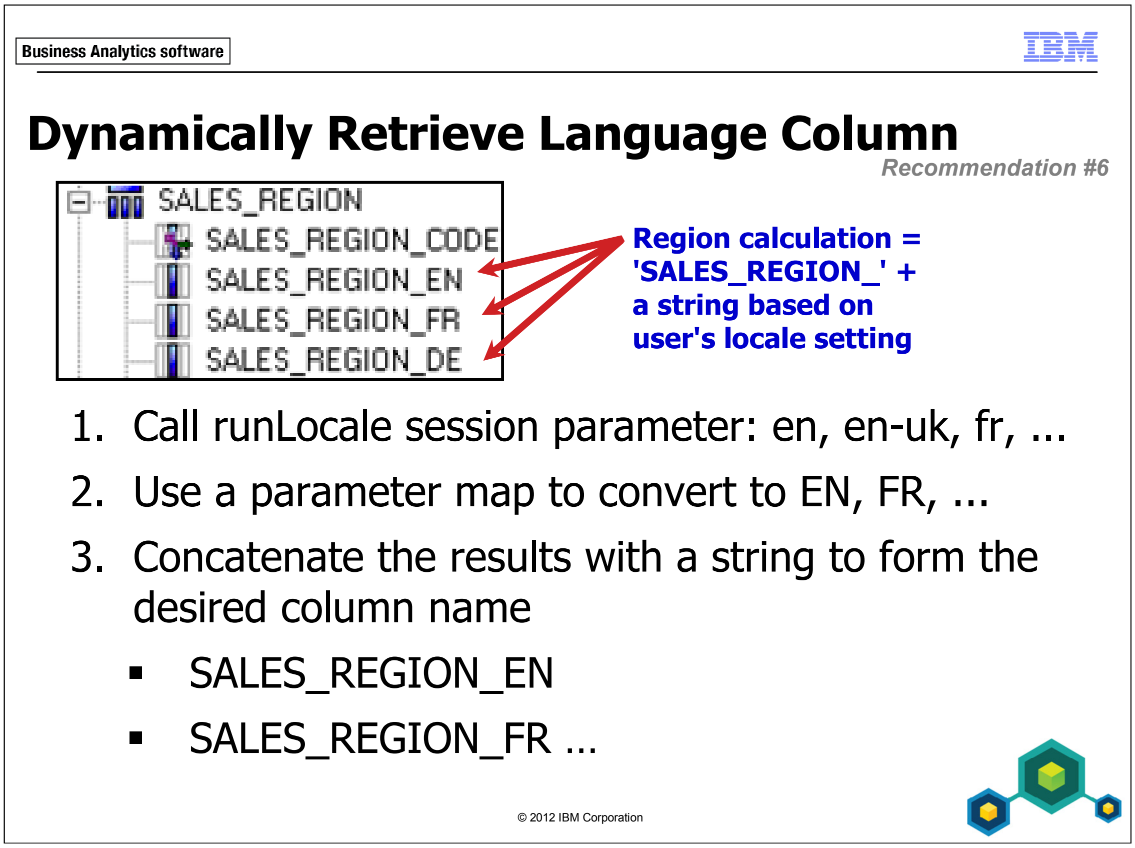
Key	Value
Bart Scott	60
Ana Orozco	80
Alex Rodriguez	53

Generated SQL

```
.......
Where
    Sales_Target_Fact.Sales_Staff_Code = 60
```

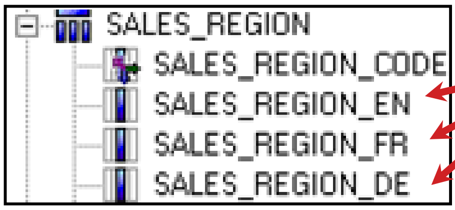
A macro is a fragment of code that you can insert within filters, calculations, properties, and so on, that are to be evaluated at run time. Macros are enclosed by the # character.

You will use the technique illustrated in the slide later in the course when applying security.


Business Analytics software

Dynamically Retrieve Language Column

Recommendation #6



Region calculation = 'SALES_REGION_' + a string based on user's locale setting

1. Call runLocale session parameter: en, en-uk, fr, ...
2. Use a parameter map to convert to EN, FR, ...
3. Concatenate the results with a string to form the desired column name
   - SALES_REGION_EN
   - SALES_REGION_FR …

© 2012 IBM Corporation

In the next demo, you will dynamically return a specific column in the data source based on a user's locale.

Instead of placing a macro in a calculation or filter, you can also edit the SQL directly in a query item. However, this may affect the query engine's ability to properly minimize the SQL. Modelers and Report Authors should be aware of this if they are examining the generated SQL in Framework Manager or Report Studio.

Demo 3: Calculate Based on Local Language

Purpose:
Authors at The Sample Outdoors Company would like report consumers to automatically view region names in the language of their locale. To do this, you will replace Retailer Site Region with a calculation that references a parameter map and a session parameter. You will then make this calculation standalone, so that you can reuse it for Branch Region, Staff Region, and Sales Target by Location.

Component:	**Framework Manager**
Project:	**GO Operational**

Task 1. Create a parameter map.

1. In the **Project Viewer** pane, **Foundation Objects View>gosales**, expand **SALES_REGION**.

   This query subject has column-based multilingual data. You pick the appropriate language column for the language you wish to view the data in.

2. Double-click **Retailer Site Region (alias)**.

   This model query subject was created with just one language query item, SALES_REGION_EN. You want to change this to pick the underlying query item based on a user's locale session parameter at run time. For example, the session locales for United States and France are en-us and fr-fr.

3. Click **Cancel**.

   You will now create a parameter map to substitute a locale session parameter to a value found in the data.

4. Right-click **Parameter Maps**, point to **Create**, and then click **Parameter Map**.

   The Create Parameter Map wizard opens. You can type in the values if you only support a small set of languages, or import the values from a file. You can also base the parameter map on query items within the model. For example, you may have a table in the database that is used to map language locales to language values matched by the data in other tables. If this is the case and the table is quite large, consider placing a filter on the query subject based on the session parameter you are using to look up the value (in this case runLocale). This reduces the record set down to one record rather than forcing a scan of the entire table at run time. For example, if the value is en-us at run time, then the query subject will only return one row consisting of en-us and EN (the key field and the value field) instead of all rows. In other words, you are dynamically reducing the parameter map list to one record before accessing it for a value.

   For this exercise there is no table that contains these mapping values, so you will use a text file that has the mappings entered for you.

5. In the **Name** box, type **Language_lookup**, and then click **Next**.
6. Click **Import File**.
7. Next to the **Filename** box, click **Browse**.
8. Navigate to <**IBM Cognos install drive> Program Files\IBM\cognos\c10\webcontent\samples\models**, click **Language_lookup.txt**, and then click **Open**.
9. Click **OK**.

   The values in the file are imported. Note that 'en-us' (and all other English variants) map to 'EN'. The same applies for other languages and their locales.

10. Click **Finish**.

Task 2. Create an embedded calculation for multilingual data.

1. In **Foundation Objects View>gosales>Retailer Site Region (alias)**, double-click **Retailer Site Region**.
2. In the **Expression definition** pane, to the left of **[gosales].[SALES_REGION].[SALES_REGION_EN]**, type **#'**
3. Replace **EN]** with **' +**

   The results appear as follows:

   #'[gosales].[SALES_REGION].[SALES_REGION_' +
4. Under **Available Components**, click the **Parameters** tab, expand **Parameter Maps**, and then double-click **Language_lookup**.
5. Collapse **Language_lookup**, expand **Session Parameters**, and then double-click **runLocale**.
6. At the end of the expression, type **+ ']'#**

   Delete any extra # symbols so the results appear as follows:

   #'[gosales].[SALES_REGION].[SALES_REGION_' + $Language_lookup{$runLocale} + ']'#

   The # character at the beginning and the end of the expression identify this expression as a macro.

   Note: This technique hard codes the path to your source query item. If you move or rename the source, you will have to update this calculation manually.
7. Click **Test Sample** ▶ to verify that the calculation works for your current locale, and then click **OK**.

   The Sales Region data will all display in the appropriate language.
8. Save the project.

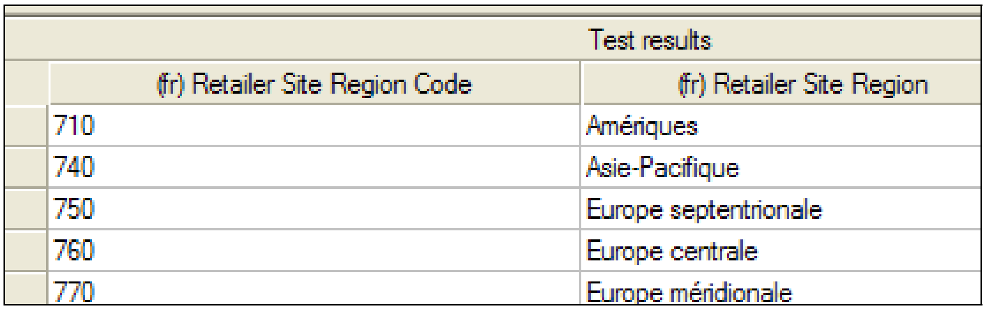
Task 2 Step 7: If the user's computer has the locale set to en_uk for United Kingdom, this string will be retrieved by the runLocale session parameter. The string will be converted to EN by the Language_lookup parameter map. And the EN string will be merged with the SALES_REGION_ prefix and the] suffix to create the desired query item name.

Task 3. Test the calculation under a different language.

1. From the **Project** menu, point to **Languages**, and then click **Define Languages**.
2. Scroll down to **French**, double-click **French** to add it to the list of **Project languages**, and then click **OK** twice.
3. If necessary, from the **View** menu, click **Tools**.
4. In the **Tools** window, if necessary, click the **Summary** tab, and then change **Active Language** to **French**.

   Notice that all object names in the Project Viewer are preceded by "(fr)". This is to remind you to translate the strings if required.
5. Right-click **(fr) Retailer Site Region (alias)**, click **Test**, and then click **Test Sample**.

   The results appear as follows:



Test results

(fr) Retailer Site Region Code	(fr) Retailer Site Region
710	Amériques
740	Asie-Pacifique
750	Europe septentrionale
760	Europe centrale
770	Europe méridionale

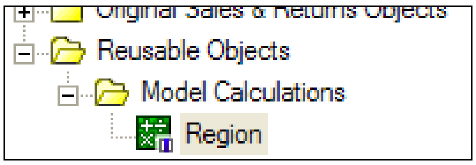
   The names appear in French based on the macro you created.
6. Click **Close**, and then change the **Active Language** back to **English**.

Task 4. Make the calculation reusable.

1. In **Foundation Objects View>gosales**, double-click **Retailer Site Region (alias)**.
2. In **Query Items and Calculations**, right-click **Retailer Site Region**, and then click **Convert to Stand-alone Calculation**.
3. Click **OK**, and then, double-click the new **Retailer Site Region** calculation at the bottom of the **gosales** namespace.

   The calculation has been preserved.
4. Click **Cancel**.
5. In the **gosales** namespace, create a folder called **Reusable Objects**.
6. In the **Reusable Objects** folder, create a folder called **Model Calculations**.
7. Drag the **Retailer Site Region** calculation into the **Model Calculations** folder.
8. Rename the calculation to **Region**.

   The results appear as follows:



Task 5. Apply the calculation to Branch Region (alias).

1. In the **Foundation Objects View**, expand **Branch Region (alias)**, and then double-click **Branch Region**.

   The expression uses the English region column. Rather than recreate the more generic language-based version, you will reuse the Region calculation.

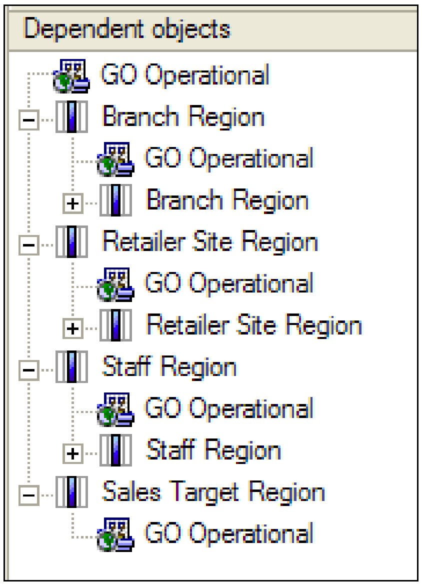
2. Delete the expression in the **Expression definition** box.
3. In the **Available Components** pane, expand **Reusable Objects**>**Model Calculations**, and then double-click **Region**.
4. Test the expression, and then click **OK**.
5. Repeat steps **1** to **3** to apply the **Region** calculation to the following query subjects:

   - **Foundation Objects View > Staff Region (alias)**
   - **Consolidation View > Sales Target by Location**

   To see if you applied the calculation in all the required locations, you can use the Show Object Dependencies feature.

6. Right-click the **Region** calculation, and then click **Show Object Dependencies**.

   In the Tools pane, a list of dependant objects is displayed.

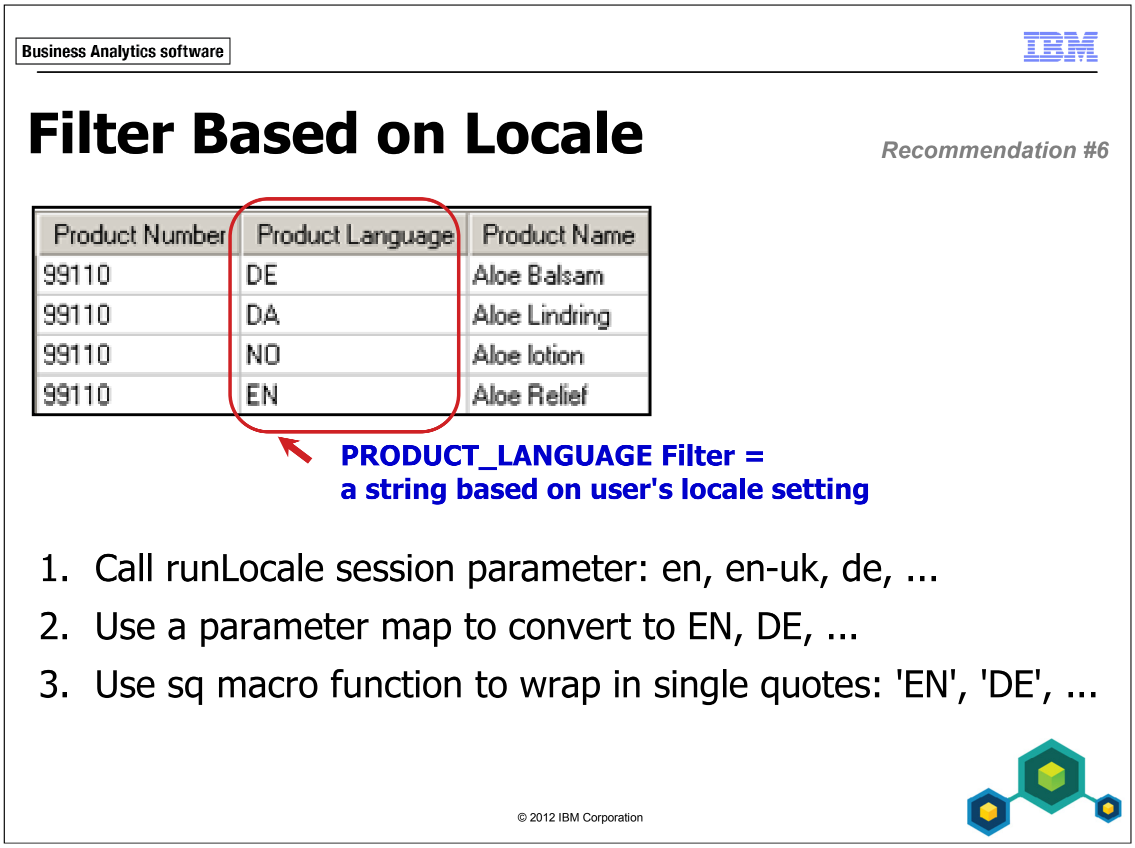


Dependent objects
- GO Operational
- Branch Region
  - GO Operational
  - Branch Region
- Retailer Site Region
  - GO Operational
  - Retailer Site Region
- Staff Region
  - GO Operational
  - Staff Region
- Sales Target Region
  - GO Operational

   You can click any of these objects to give them focus in the Project Viewer or Diagram pane. By doing this, you can verify you applied the calculation in all required areas.

Results:
You modified Retailer Site Region (alias) to automatically view region names in a language based on the report consumer's locale. You then made the calculation stand-alone, and reused it for the Branch Region (alias).

We could have placed the calculation directly in the SALES_REGION data source query subject, but calculations in data source query subjects will cause additional queries to the data source for metadata. For some data source vendors this can have a performance impact. This also affects portability of the model and change management.



Business Analytics software

Filter Based on Locale

Recommendation #6

Product Number	Product Language	Product Name
99110	DE	Aloe Balsam
99110	DA	Aloe Lindring
99110	NO	Aloe lotion
99110	EN	Aloe Relief

PRODUCT_LANGUAGE Filter =
a string based on user's locale setting

1. Call runLocale session parameter: en, en-uk, de, ...
2. Use a parameter map to convert to EN, DE, ...
3. Use sq macro function to wrap in single quotes: 'EN', 'DE', ...

© 2012 IBM Corporation

The same process that you used to create a calculation when the language selection is column-based can be used to create a filter when the language selection is row-based. Instead of building up a string for the column name, you simply create the string matching a PRODUCT_LANGUAGE row value and filter on this value.

Workshop 1: Filter Local Language

Each product has multiple records in the PRODUCT_NAME_LOOKUP table, one per language. Authors at the Sample Outdoors Company want report consumers to automatically view product names in the language of their locale. To do this, add a filter to PRODUCT_NAME_LOOKUP which equates PRODUCT_LANGUAGE to #sq($Language_lookup {$runLocale}) #

To do this:

- Use the runLocale session parameter to extract the locale.
- Convert the locale to the format of our PRODUCT_LANGUAGE column by using the Language_lookup parameter map that you created earlier.
- Wrap the string in single quotes by using a macro called sq.

Note that the filter changes the nature of the relationship between PRODUCT_NAME_LOOKUP and Product Type & Product to be 1..1 on both sides. Applying this filter and changing the cardinality will resolve unexpected results when querying Product Name.

Test your underlying changes in the Consolidation View using Product>Product Name and Sales Fact>Revenue.

Save the project when finished.

For more detailed information outlined as tasks, see the Task Table on the next page.

To see the desired filters, see the Workshop Results section that follows the Task Table.

The sq macro function is required because once the macro resolves to the appropriate value from the parameter map, it must be wrapped in quotes to be identified as a string by the filter.

If multi-lingual databases are not of interest, the module 9 solution file in C:\Edcognos\B5252\Solution Files\Module 9\Workshop 1 can be used to bypass the workshops.

Workshop 1: Task Table

Task	Where to Work	Hints
1. Add a filter to PRODUCT_ NAME_LOOKUP	Project Viewer, Foundation View, gosales	• Double-click the PRODUCT_ NAME_LOOKUP query subject. • Click the Filters tab. • Click Add. • Name it Language Filter.
2. Create the filter definition.	Available Components, Expression definition	• Add PRODUCT_LANGUAGE to the expression. • Type = at the end of the expression. • Select the Parameters tab, expand Macro Functions, and then double-click sq. • Expand Parameter Maps and double-click Language_lookup. • Expand Session Parameters and double-click runLocale. • Click OK then test the filter using the Test tab. All product names should be in your current language.

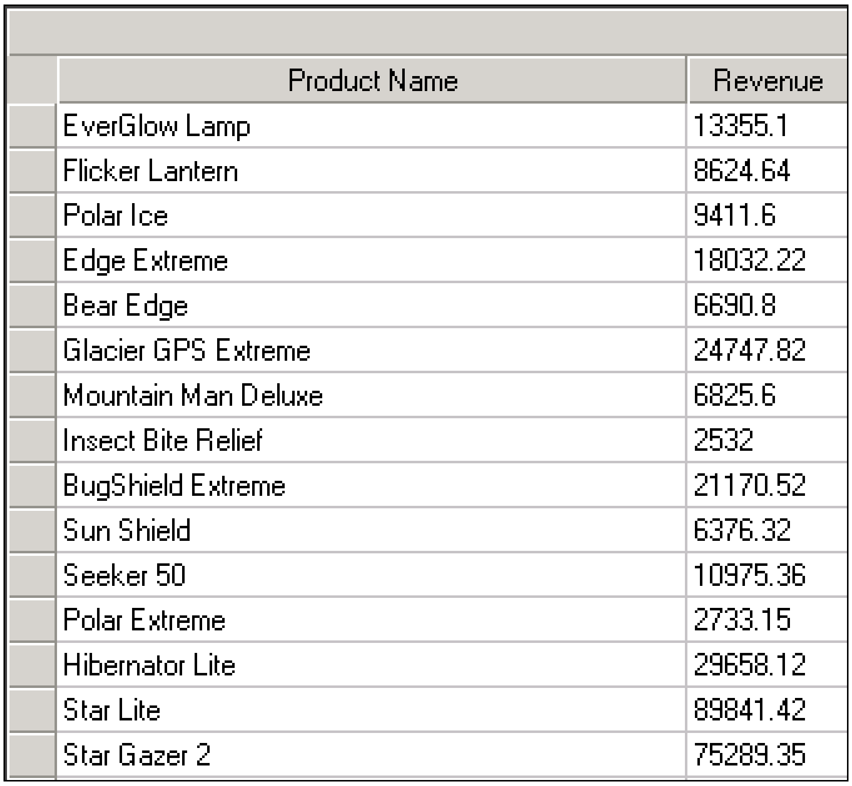
Task	Where to Work	Hints
3. Update relationship.	Project Viewer, Context Explorer	• Change the relationship between PRODUCT_NAME_LOOKUP and Product Type & Product to be 1..1 on both sides. • Test in the Consolidation View using Products>Product Name and Sales Fact>Revenue • Save the project.

Workshop 1: Workshop Results

Your Local Language filter in PRODUCT_NAME_LOOKUP should appear as follows:

[gosales].[PRODUCT_NAME_LOOKUP].[PRODUCT_LANGUAGE] = #sq($Language_lookup{$runLocale})#

Your test on Product Name and Revenue should appear as follows:



Product Name	Revenue
EverGlow Lamp	13355.1
Flicker Lantern	8624.64
Polar Ice	9411.6
Edge Extreme	18032.22
Bear Edge	6690.8
Glacier GPS Extreme	24747.82
Mountain Man Deluxe	6825.6
Insect Bite Relief	2532
BugShield Extreme	21170.52
Sun Shield	6376.32
Seeker 50	10975.36
Polar Extreme	2733.15
Hibernator Lite	29658.12
Star Lite	89841.42
Star Gazer 2	75289.35

Product names are now only returned in one language.

Workshop 2: Calculate Based on Local Language

Recall that to meet business needs, you created a reusable Region calculation for both Retailer Site Region and Branch Region as well as the Sales Target by Location query subject in the Consolidation View. You will now do the same thing for Retailer Site Country, Branch Country, and Sales Target by Location with a Country calculation.

Create a calculation as follows:

- Create a new calculation in Reusable Objects>Model Calculations, called Country.
- Set its expression to:

  #'[gosales].[COUNTRY].[COUNTRY_' + $Language_lookup{$runLocale} + ']'#
- Change the definitions of the following query subjects to use this calculation:
  - Foundation Objects View > Branch Country (alias) > Branch Country
  - Foundation Objects View > Retailer Site Country (alias) > Retailer Site Country
  - Foundation Objects View > Staff Country (alias) > Staff Country
  - Consolidation View > Sales Target by Location > Sales Target Country
- Save and close Framework Manager.

For more detailed information outlined as tasks, see the Task Table on the next page.

This workshop can be made optional if you are pressed for time. We will be using a new starting point project at the start of the next module, since we need to add additional calculations for Product Line and Position.

Workshop 2: Task Table

Task	Where to Work	Hints
1. Create a Country calculation.	Project Viewer, Foundation View, gosales, Reusable Objects, Model Calculations	• Right-click the Model Calculations folder, Create > Calculation. • Name the calculation Country.
2. Create the expression.	Available Components, Expression definition	• Set its expression to #'[gosales].[COUNTRY].[COUNTRY_' + $Language_lookup{$runLocale} + ']'#
3. Change the four query subjects to use the calculation.	Project Viewer, Query Subject Definition dialog boxes for Retailer Site Country (alias) and Branch Country (alias)	• In the Branch Country (alias) query subject, edit Branch Country. • Delete the Expression definition. • In Available Components, locate the new calculation and add it to the Expression definition. • Test, and then click OK. • Repeat for Retailer Site Country (alias), Staff Country (alias), as well as Sales Target by Location (Consolidation View). • Save and close Framework Manager

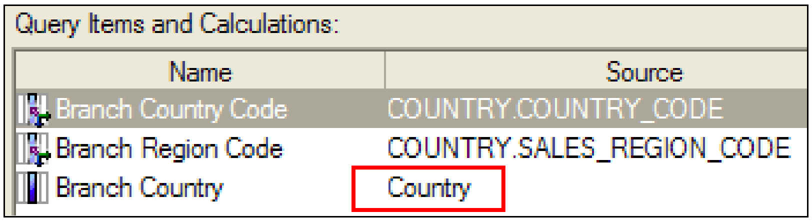
Workshop 2: Workshop Results

Your Country calculation expression should appear as shown below:

#'[gosales].[COUNTRY].[COUNTRY_' + $Language_lookup{$runLocale} + ']'#

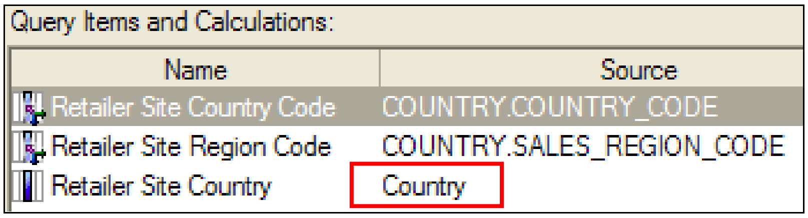
This new calculation should be referenced in the following:

Your Branch Country (alias) model query subject


Query Items and Calculations:

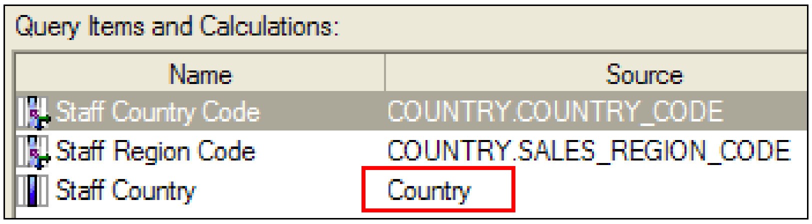
Name	Source
Branch Country Code	COUNTRY.COUNTRY_CODE
Branch Region Code	COUNTRY.SALES_REGION_CODE
Branch Country	Country

Your Retailer Site Country (alias) model query subject


Query Items and Calculations:

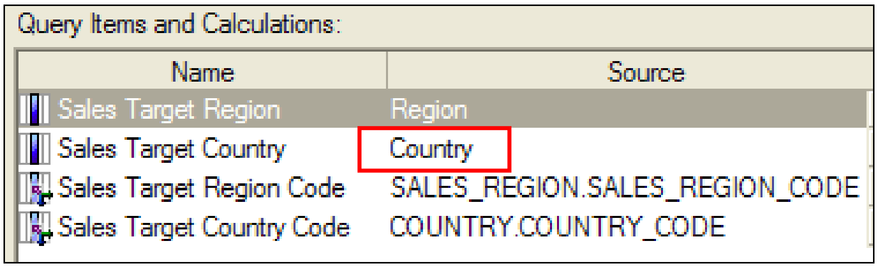
Name	Source
Retailer Site Country Code	COUNTRY.COUNTRY_CODE
Retailer Site Region Code	COUNTRY.SALES_REGION_CODE
Retailer Site Country	Country

Your Staff Country (alias) model query subject

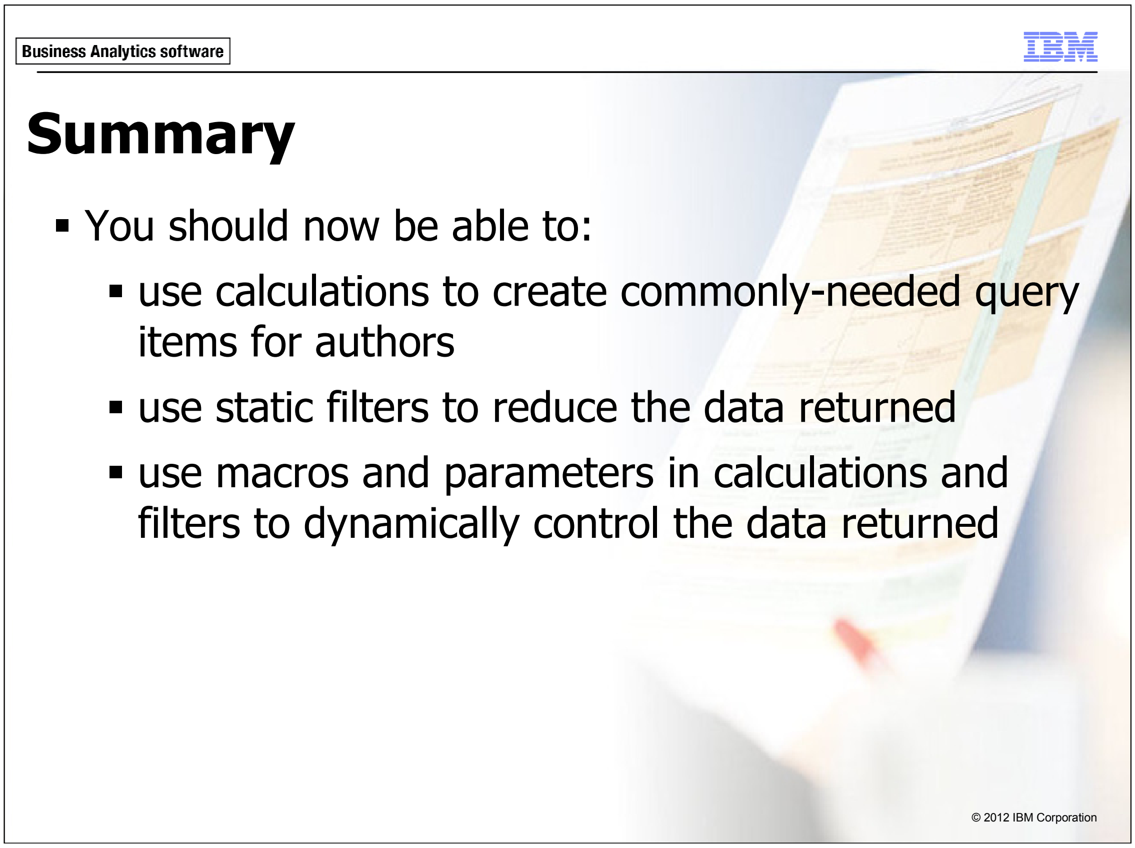

Query Items and Calculations:

Name	Source
Staff Country Code	COUNTRY.COUNTRY_CODE
Staff Region Code	COUNTRY.SALES_REGION_CODE
Staff Country	Country

Your Sales Target by Location model query subject


Query Items and Calculations:

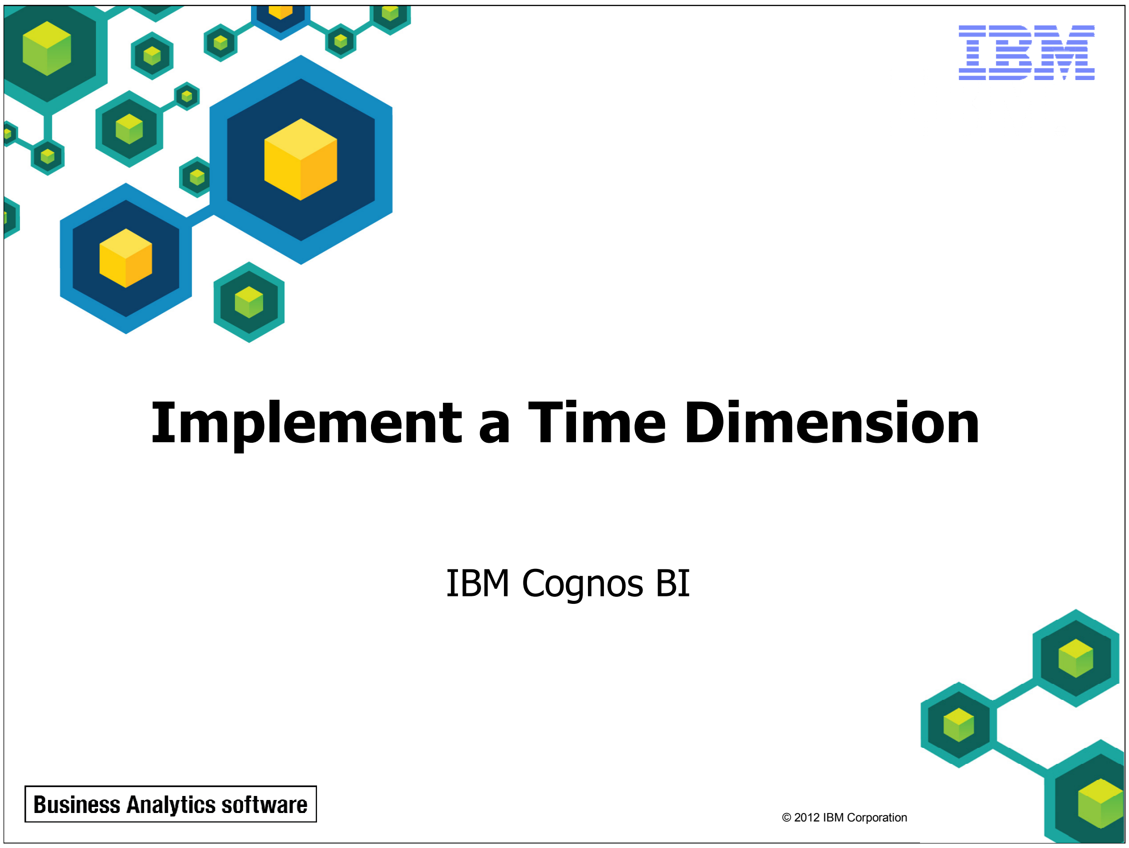
Name	Source
Sales Target Region	Region
Sales Target Country	Country
Sales Target Region Code	SALES_REGION.SALES_REGION_CODE
Sales Target Country Code	COUNTRY.COUNTRY_CODE


Business Analytics software

Summary

- You should now be able to:
  - use calculations to create commonly-needed query items for authors
  - use static filters to reduce the data returned
  - use macros and parameters in calculations and filters to dynamically control the data returned

© 2012 IBM Corporation



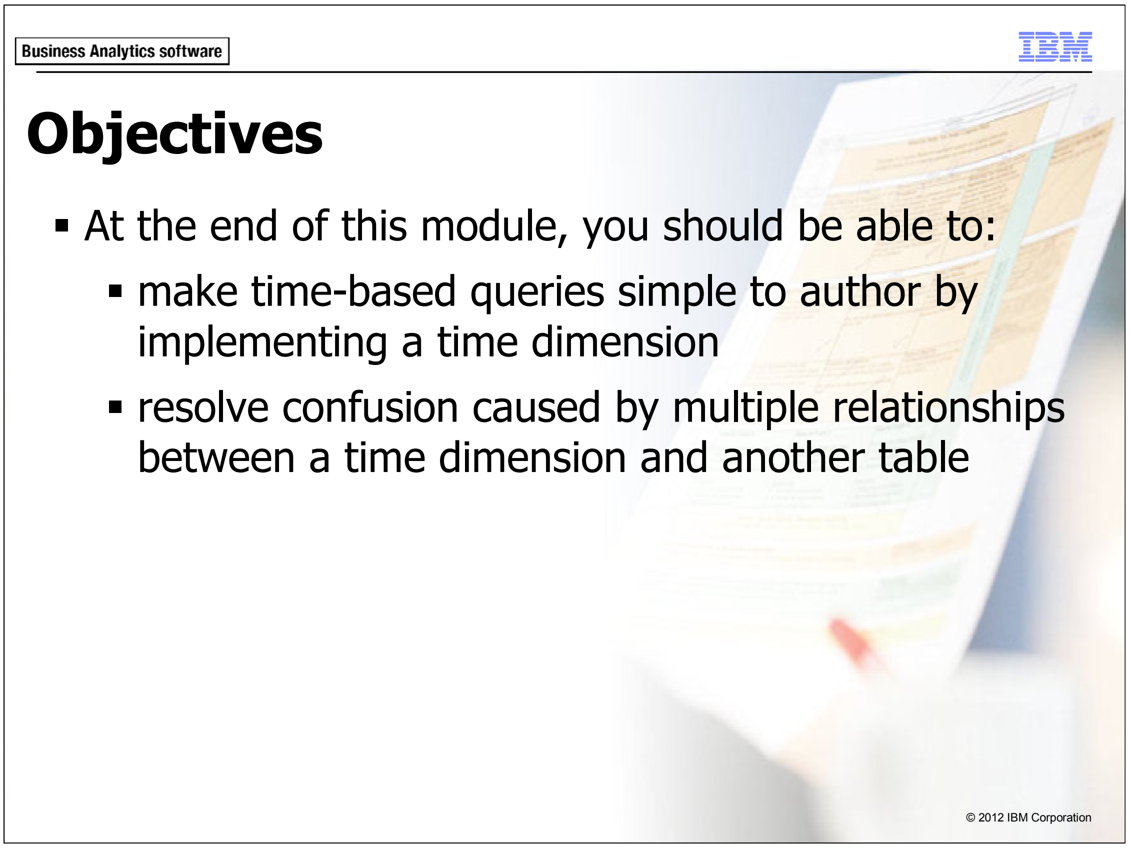
Implement a Time Dimension

IBM Cognos BI

Business Analytics software

Unless otherwise specified in demo or workshop steps, you will always log on to IBM Cognos in the Local LDAP namespace using the following credentials:

- User ID: admin
- Password: Education1



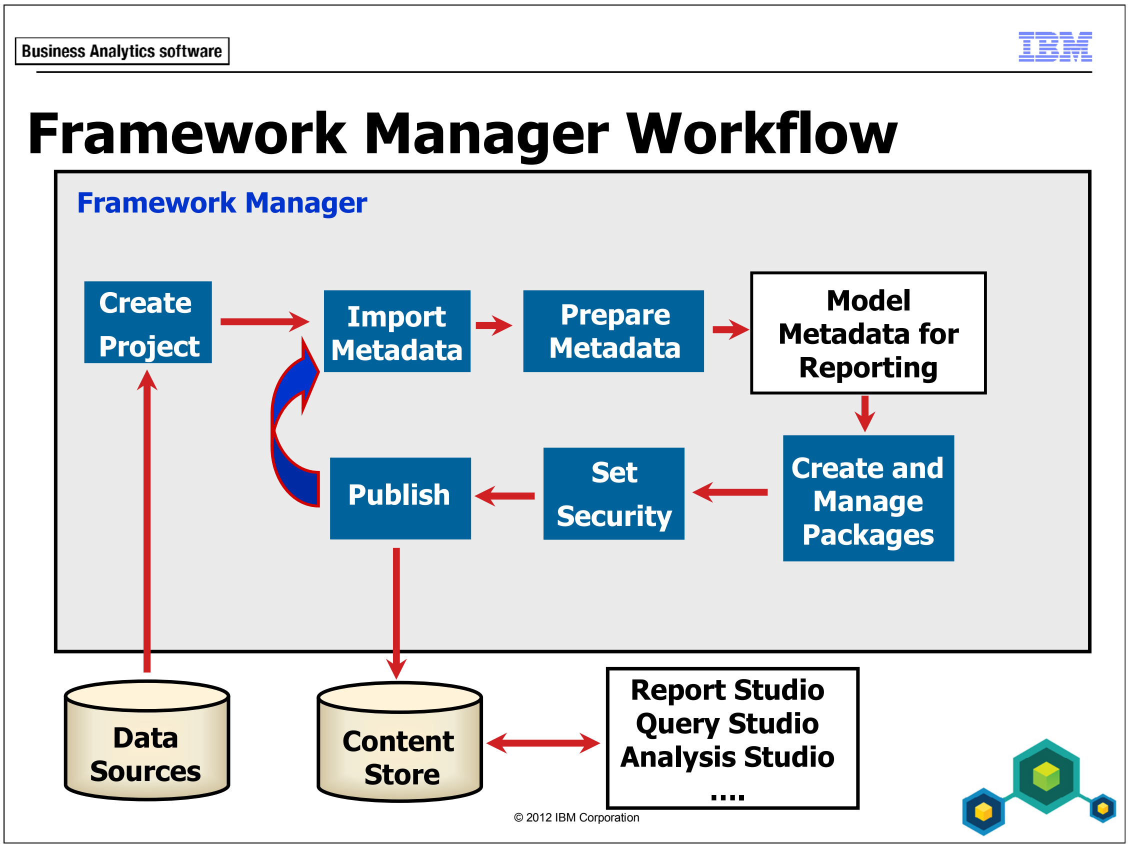
Business Analytics software

Objectives

- At the end of this module, you should be able to:
  - make time-based queries simple to author by implementing a time dimension
  - resolve confusion caused by multiple relationships between a time dimension and another table

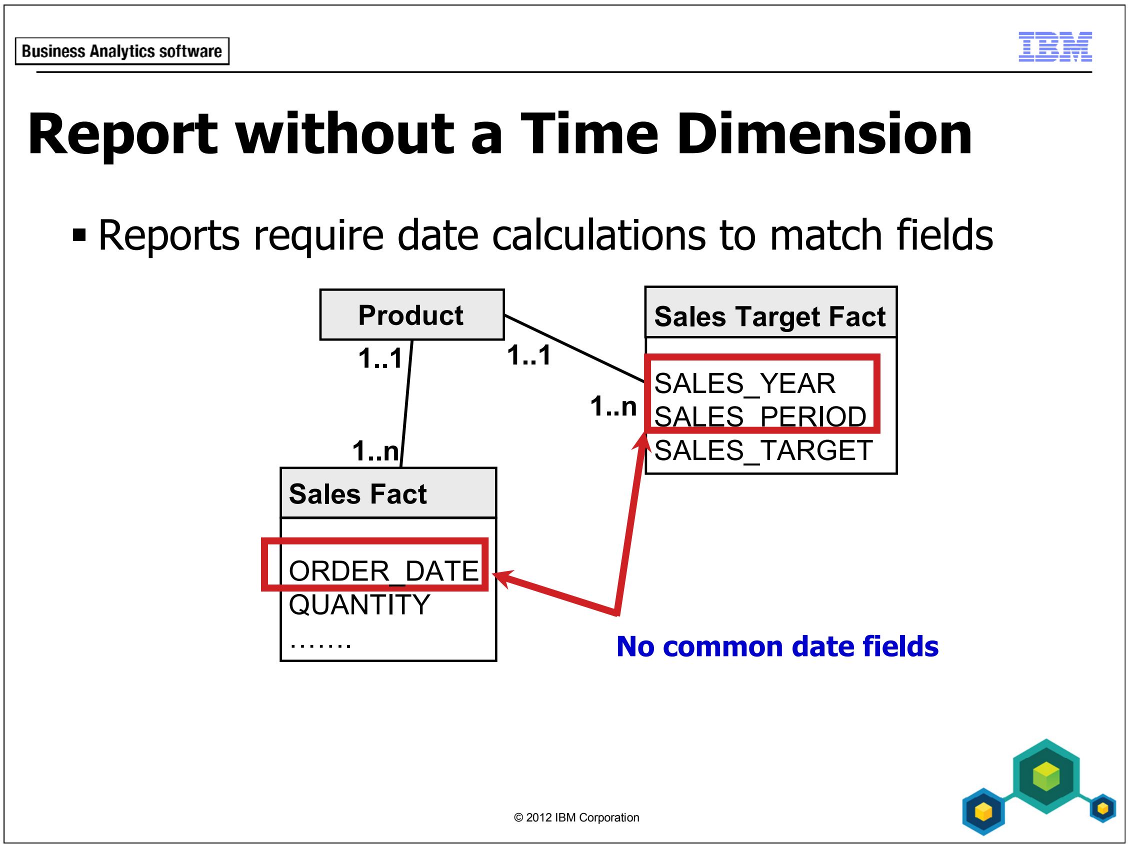
© 2012 IBM Corporation

Before reviewing this module, you should be familiar with IBM Cognos BI, IBM Cognos Connection, Query Studio and Report Studio. Suggested modules to reference:
• Overview of IBM Cognos BI
• Identify Common Data Structures
• Gather Requirements
• Create a Baseline Project
• Prepare Reusable Metadata
• Model for Predictable Results: Reporting Issues
• Model for Predictable Results: Virtual Star Schemas
• Model for Predictable Results: Consolidate Metadata
• Calculations and Filters



This module teaches the technique of implementing a time dimension in an operational model to simplify time-based queries across different facts.

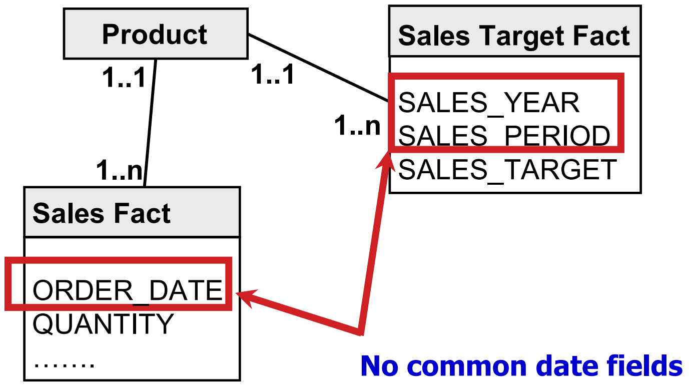
As for the SAP BW Time Dependent Hierarchies, this is more of something that will now be imported properly into Framework Manager where in previous version adjustments had to be made. As of v10.1, there is no modeling to do on these data items once imported.
Time-dependant hierarchies now automatically reflect hierarchy or structure changes. When a structure is imported into Framework Manager, each SAP BW time hierarchy is depicted as an individual level. Report Studio users can use these structures to report on and compare levels that are valid for a specific time period.



Business Analytics software

Report without a Time Dimension

- Reports require date calculations to match fields



© 2012 IBM Corporation

Either modelers or report authors must create calculations that manipulate date data to create matching values that can be used to compare data from different areas of the business. This can cause slower performance because you are now comparing data on non-indexed values.

In the slide example, you would need to create two calculations on ORDER_DATE in the Sales Fact query subject to match the SALES_YEAR and SALES_PERIOD fields in the Sales Target Fact query subject. You will see this in the next demo.

Other issues when using date fields from fact tables include the inability to properly stitch these queries together since there is no conformed dimension used.

Demo 1: Report without a Time Dimension

Purpose:
A typical business report compares the monthly sales targets to actual sales. You will see if this is easily accomplished with your current data.

Components:	**Framework Manager, IBM Cognos Workspace Advanced**
Project:	**GO Operational**
Package:	**GO Operational**

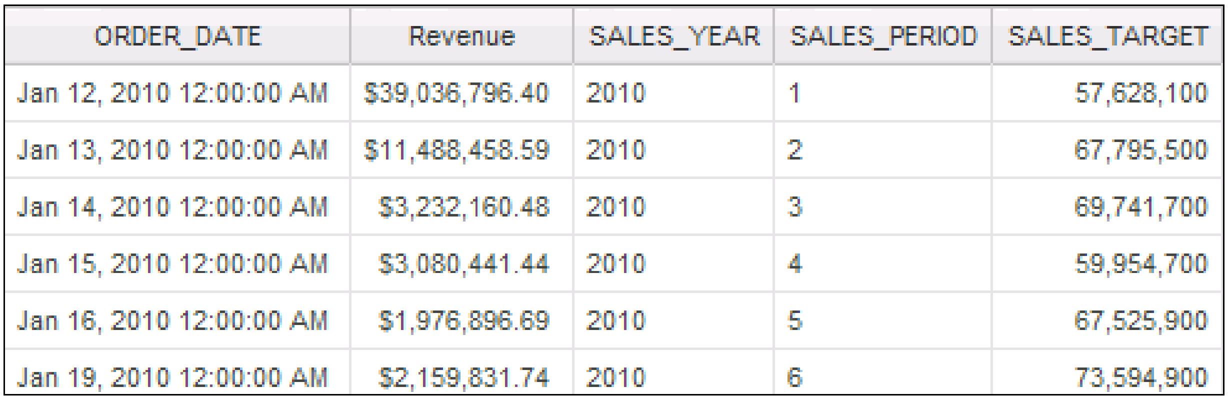
Task 1. Examine problems when a time dimension is absent.

1. In **Framework Manager**, open the **GO Operational** project located at **C:\Edcognos\B5252\CBIFM-Start Files\Module 10\GO Operational**.
2. If prompted, log in as User ID **admin**, and Password **Education1**.
3. Publish the **GO Operational** package.
4. In **IBM Cognos Connection**, open **Cognos Workspace Advanced**.
5. Select the **GO Operational** package, click **Create new**, and then double-click **List**.

6. Add the following items to the report from **Foundation Objects View**:

Query Subject	Query Item
Sales Fact	**ORDER_DATE** **Revenue**
SALES_TARGET	**SALES_YEAR** **SALES_PERIOD** **SALES_TARGET**

The results appear as follows:



ORDER_DATE	Revenue	SALES_YEAR	SALES_PERIOD	SALES_TARGET
Jan 12, 2010 12:00:00 AM	$39,036,796.40	2010	1	57,628,100
Jan 13, 2010 12:00:00 AM	$11,488,458.59	2010	2	67,795,500
Jan 14, 2010 12:00:00 AM	$3,232,160.48	2010	3	69,741,700
Jan 15, 2010 12:00:00 AM	$3,080,441.44	2010	4	59,954,700
Jan 16, 2010 12:00:00 AM	$1,976,896.69	2010	5	67,525,900
Jan 19, 2010 12:00:00 AM	$2,159,831.74	2010	6	73,594,900

The year and period (month) from SALES_TARGET do not match those in the ORDER_DATE from Sales Fact. You cannot get expected results without using a time dimension or creating calculations or modeling complex summary queries that manipulate date fields to tie the data together.

For example, you could extract the year and month from ORDER_DATE to match the fields in SALES_TARGET using functions such as:

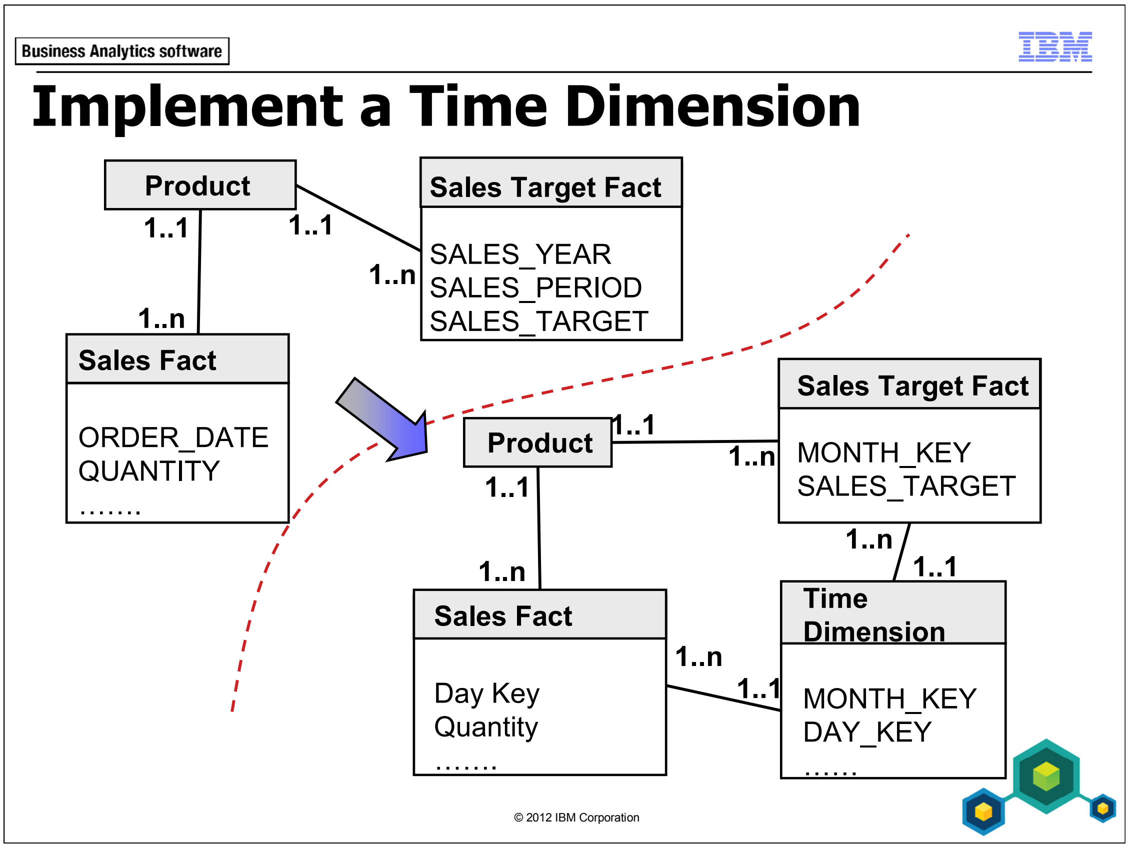
- Year(ORDER_DATE)
- Month(ORDER_DATE)

You could then sort in ascending order on the two new columns in order to match them to the SALES_YEAR and SALES_PERIOD columns from SALES_TARGET.

However, this cannot be accomplished in Query Studio and therefore requires the modeler to create the calculations or restricts these types of calculations to Report Studio authors. The results also do not properly stitch the two fact queries because the items used to compare the data do not come from a conformed source.

7. Return to **IBM Cognos Connection**, and do not save the report.

Results:
You identified the complexity of trying to create a time-based report without a time dimension. Without a conformed time dimension, you cannot achieve a proper stitch query and require calculations in order to match date fields from different sources.



Business Analytics software

Implement a Time Dimension

© 2012 IBM Corporation

A time dimension provides a conformed dimension to allow time-based queries across facts, either at the same or different levels of granularity. Currently the model being built in this course has various date formats in the fact tables (such as ORDER_DATE, SALES_PERIOD) that make time-based queries between facts difficult. Time dimensions also allow for easy rollups, for example, day-based facts can be rolled up to the month level.

A time dimension can be easily created by creating a table with all applicable dates for a time period, ensuring that it includes all required keys, date derivatives, and formats. There are several SQL scripts that can be found on the internet to build time dimensions. You can use any of the popular Internet search engines to find them.

You can import the time dimension into Framework Manager and create relationships as necessary. If the fact query subjects do not contain the required keys to relate to the time dimension, speak to the database administrator to see if they can be incorporated.

You can also create time keys in Framework Manager using calculations, but this may reduce performance since you are filtering on non-indexed fields.

Demo 2: Implement a Time Dimension

Purpose:
In order to perform predictable time-based queries, you will import and implement a time dimension.

Components:	**Framework Manager**, **Query Studio, Report Studio**
Project:	**GO Operational**
Package:	**GO Operational**

Task 1. Import Time Dimension metadata.

1. In **Framework Manager**, in the **Project Viewer** pane, expand **Foundation Objects View**, right-click the **gosales** namespace, and then click **Run Metadata Wizard**.
2. Ensure that **Data Sources** is selected, and then click **Next**.
3. Ensure that **GOSALES** is selected, and then click **Next**.
4. In the list of objects, expand **GOSALES>Tables**, and then select **TIME_DIMENSION**.
5. Click **Next**, and then click **Import**.
6. Click **Finish**.

Task 2. Test and adjust TIME_DIMENSION.

1. Drag **TIME_DIMENSION** below the **Staff Region (alias)** query subject to organize it alphabetically.
2. Right-click **TIME_DIMENSION**, click **Launch Context Explorer**, and then click **Show Related Objects**.

   TIME_DIMENSION is a generic query subject with no relationships to other query subjects.
3. Close **Context Explorer**.

4. Right-click **TIME_DIMENSION**, click **Test**, and then click **Test Sample**.

   This query subject is simply a calendar breakdown, giving the time over a certain range in multiple formats. It can be used to join two fact tables with different date formats. In this case, TIME_DIMENSION's DAY_DATE matches the format of Sales Fact's ORDER_DATE and TIME_DIMENSION's CURRENT_YEAR and CURRENT_MONTH match SALES_TARGET's SALES_YEAR and SALES_PERIOD.

5. Click **Close**.

6. Expand **TIME_DIMENSION**, click **CURRENT_YEAR**, and then, in the **Properties** pane beside **Format**, click **<Click to edit>**.

   Note: If your Properties pane is not visible, then from the View menu click Properties.

7. Under **Format type**, select **Number**, set **Use Thousands Separator** to **No**, and then click **OK**.

8. In the **TIME_DIMENSION**, change the **Usage** property of all query items identified as **Fact** to **Attribute**.

Task 3. Build Relationships to the Facts.

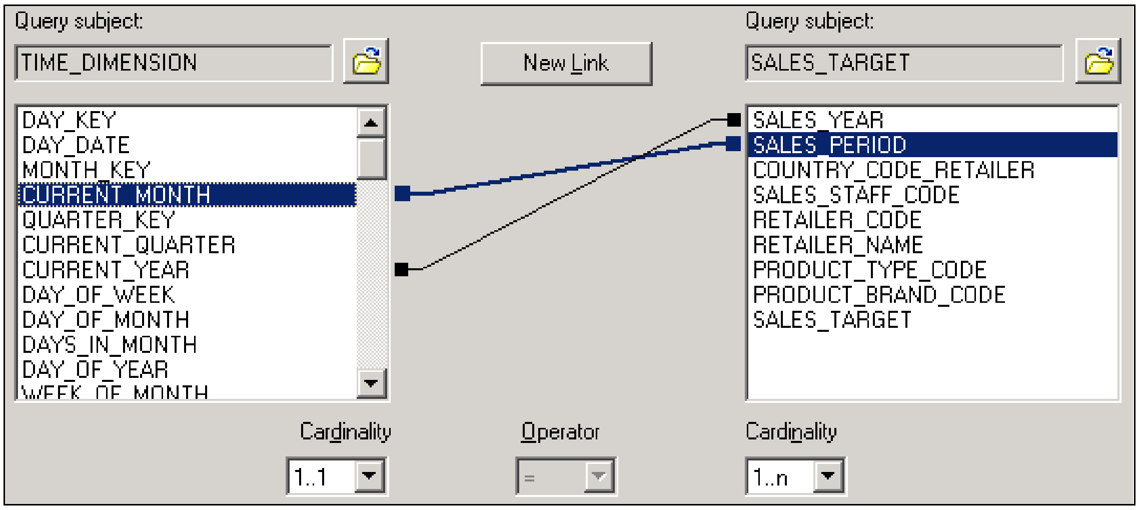
1. In the **Project Viewer** pane, in the **Foundation Objects View**, create the following relationships:

   - TIME_DIMENSION (**DAY_DATE**, **1..1**) to Sales Fact (**ORDER_DATE**, **1..n**)
   - TIME_DIMENSION (**DAY_DATE**, **1..1**) to Returns Fact (**RETURN_DATE**, **1..n**)

2. Create the following relationship but do not close the Relationship dialog box:

   - TIME_DIMENSION (**CURRENT_YEAR**, **1..1**) to SALES TARGET (**SALES_YEAR**, **1..n**)

3. Click **New Link** to add a second segment to this two-segment relationship:
   - TIME_DIMENSION (**CURRENT_MONTH**, **1..1**) to SALES TARGET (**SALES_PERIOD**, **1..n**)


Query subject: TIME_DIMENSION

New Link

Query subject: SALES_TARGET

DAY_KEY
DAY_DATE
MONTH_KEY
CURRENT_MONTH
QUARTER_KEY
CURRENT_QUARTER
CURRENT_YEAR
DAY_OF_WEEK
DAY_OF_MONTH
DAYS_IN_MONTH
DAY_OF_YEAR
WEEK_OF_MONTH

SALES_YEAR
SALES_PERIOD
COUNTRY_CODE_RETAILER
SALES_STAFF_CODE
RETAILER_CODE
RETAILER_NAME
PRODUCT_TYPE_CODE
PRODUCT_BRAND_CODE
SALES_TARGET

Cardinality: 1..1
Operator: =
Cardinality: 1..n

Note the Expression below the linkages:
TIME_DIMENSION.CURRENT_YEAR =
SALES_TARGET.SALES_YEAR
AND TIME_DIMENSION.CURRENT_MONTH =
SALES_TARGET.SALES_PERIOD

This relationship now consists of two values in TIME_DIMENSION that match time period values in SALES_TARGET. The fields used from the TIME_DIMENSION table, in this scenario, are not indexed in the database. This may affect performance. The ideal situation would be to have a month key in the SALES_TARGET table in the database. This would allow you to create your relationship based on one common indexed key in each table. If this is not an option, you should consider asking the database administrator to index the CURRENT_MONTH and CURRENT_YEAR fields in the TIME_DIMENSION table if performance is an issue.

4. Click **OK**.

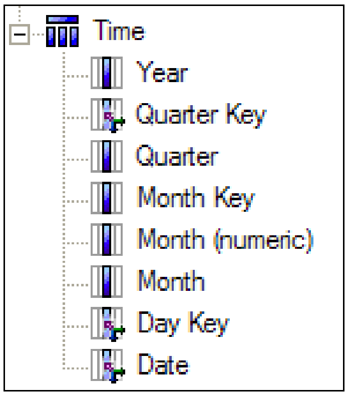
Task 4. Create the Time model query subject in the Consolidation View.

1. Right-click **Consolidation View**, point to **Create**, and then click **Query Subject**.
2. Name the query subject **Time**, and then, with **Model** selected, click **OK**.
3. In the **Available Model Objects** pane, expand **Foundation Objects View> gosales>TIME_DIMENSION**.
4. Add the following query items by double-clicking them:

   CURRENT_YEAR
   QUARTER_KEY
   CURRENT_QUARTER
   MONTH_KEY
   CURRENT_MONTH
   MONTH_EN
   DAY_KEY
   DAY_DATE
5. Click **OK**, and then rename the items as follows:

   CURRENT_YEAR to **Year**
   QUARTER_KEY to **Quarter Key**
   CURRENT_QUARTER to **Quarter**
   MONTH_KEY to **Month Key**
   CURRENT_MONTH to **Month (numeric)**
   MONTH_EN to **Month**
   DAY_KEY to **Day Key**
   DAY_DATE to **Date**

6. Organize the items as shown below:



7. Drag **Time** above the **Model Filters** folder.
8. Save the project.

Task 5. Prepare the model for multilingual data.

1. In **Foundation Objects View** > **gosales** > **Reusable Objects**, right-click **Model Calculations**, point to **Create** and click **Calculation**.
2. In the **Name** box, type **Month**.
3. In the **Available Components** box, navigate to **Foundation Objects View**> **gosales**>**TIME_DIMENSION** and drag the **MONTH_EN** data item into the **Expression definition** box.
4. In the **Expression definition** box, add **#'** in front of **[gosales]** and replace **EN]** with **' + $Language_lookup{$runLocale} + ']'#**.

   The completed expression should read as follows:

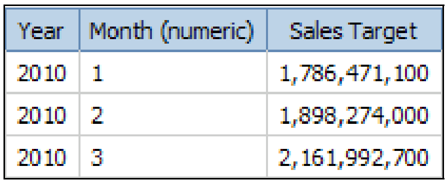
   #'[gosales].[TIME_DIMENSION].[MONTH_'+$Language_lookup {$runLocale}+']'#
5. Click **OK**, and then save the project.

Task 6. Publish and test package.

1. Publish the **GO Operational** package.
2. In **IBM Cognos Connection**, launch **Query Studio**, and then select the **GO Operational** package.
3. Create a new report with the following items from **Consolidation View**:

Query Subject	Query Item
Time	**Year** **Month (numeric)**
Sales Target Fact	**Sales Target**

The results appear as follows:



Year	Month (numeric)	Sales Target
2010	1	1,786,471,100
2010	2	1,898,274,000
2010	3	2,161,992,700

Knowing your business, and recalling the Sales Target values in the first demo, you can see that monthly sales targets are in the billions and this is too high. Double counting is occurring because there is not enough information specified on the time dimension to allow IBM Cognos to roll up values correctly at the month. You will need to specify determinants for the underlying TIME_DIMENSION query subject to resolve this issue.

You will specify determinants in the next module.

4. Save the query as **Test Time Dimension** to the default location.

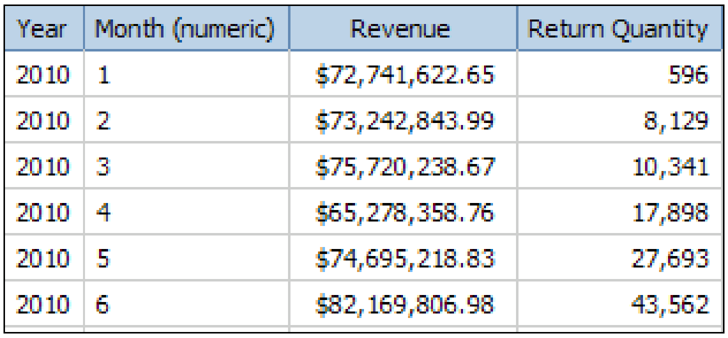
Note: It is important that you save this query, as it will be used in a subsequent demo.

You will now test a multi-fact query using the Time dimension as a conformed dimension.

5. Create a new report with the following items from **Consolidation View**:

Query Subject	Query Item
Time	**Year** **Month (Numeric)**
Sales Fact	**Revenue**
Returns Fact	**Return Quantity**

The results appear as follows:



Year	Month (numeric)	Revenue	Return Quantity
2010	1	$72,741,622.65	596
2010	2	$73,242,843.99	8,129
2010	3	$75,720,238.67	10,341
2010	4	$65,278,358.76	17,898
2010	5	$74,695,218.83	27,693
2010	6	$82,169,806.98	43,562

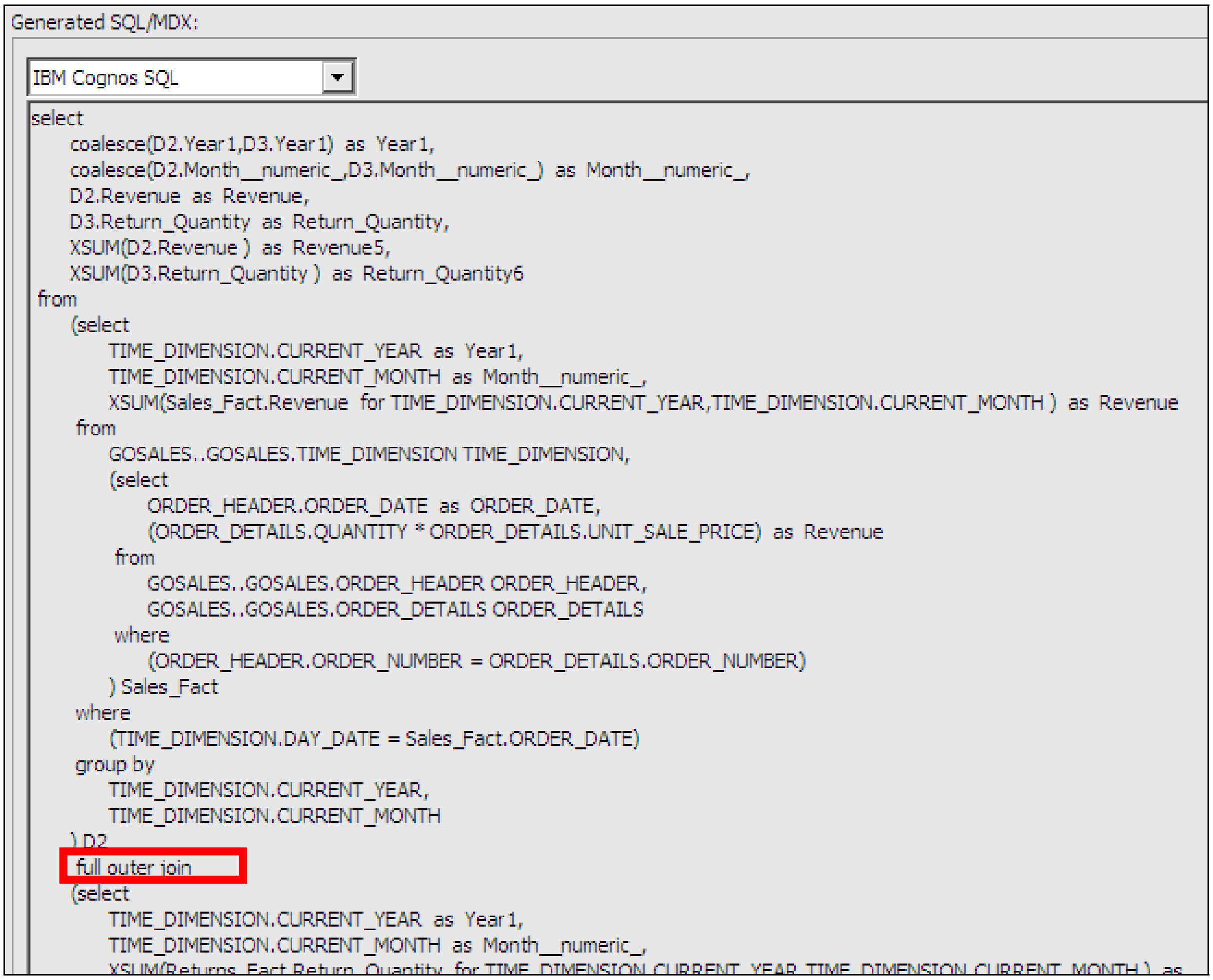
This multi-fact query was accomplished using Time as a conformed dimension to stitch the two fact queries together.

To verify this, you will open the report in Report Studio to view the SQL.

6. Under **Menu**, click **Manage File**, and then click **Open in Report Studio**.
7. In **Report Studio**, from the **Tools** menu, click **Show Generated SQL/MDX**.

8. From the drop down list, select **IBM Cognos SQL**.

   The results appear as follows:

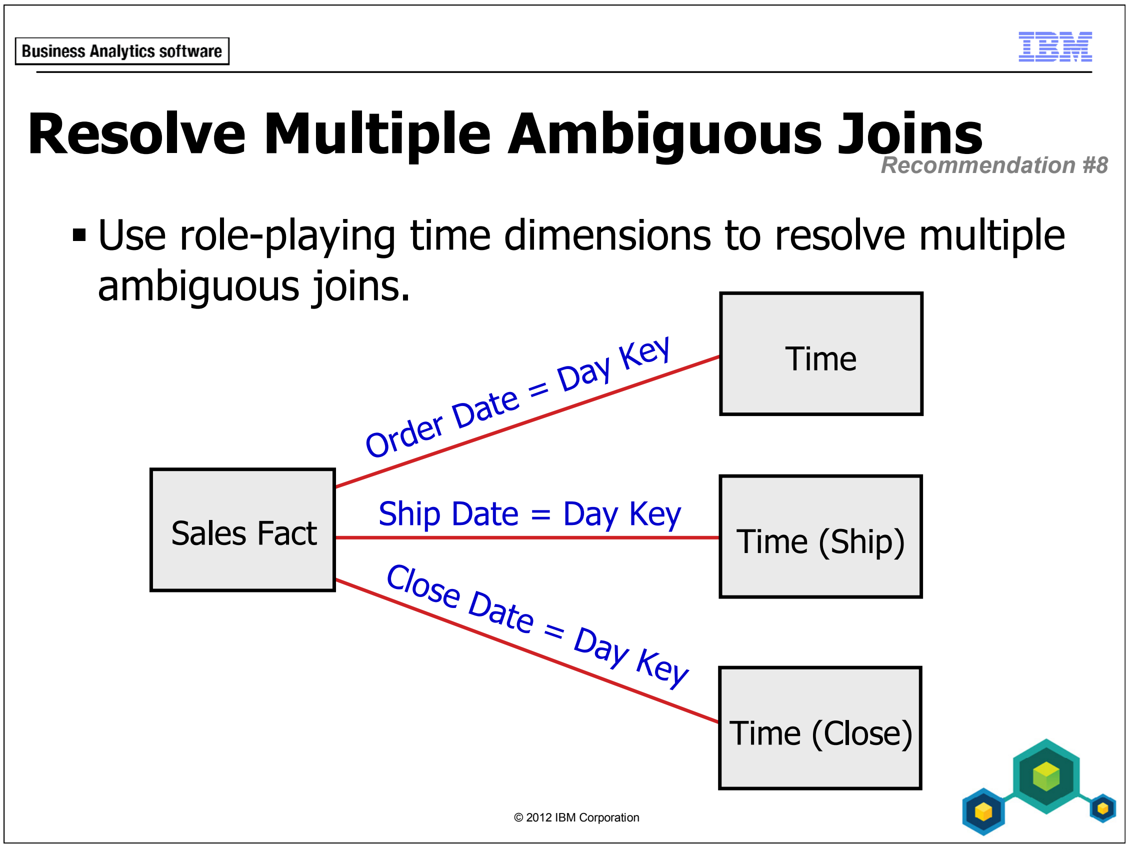


```
select
    coalesce(D2.Year1,D3.Year1)  as  Year1,
    coalesce(D2.Month__numeric_,D3.Month__numeric_)  as  Month__numeric_,
    D2.Revenue  as  Revenue,
    D3.Return_Quantity  as  Return_Quantity,
    XSUM(D2.Revenue )  as  Revenue5,
    XSUM(D3.Return_Quantity )  as  Return_Quantity6
 from
    (select
        TIME_DIMENSION.CURRENT_YEAR  as  Year1,
        TIME_DIMENSION.CURRENT_MONTH  as  Month__numeric_,
        XSUM(Sales_Fact.Revenue  for TIME_DIMENSION.CURRENT_YEAR,TIME_DIMENSION.CURRENT_MONTH )  as  Revenue
     from
        GOSALES..GOSALES.TIME_DIMENSION TIME_DIMENSION,
        (select
            ORDER_HEADER.ORDER_DATE  as  ORDER_DATE,
            (ORDER_DETAILS.QUANTITY * ORDER_DETAILS.UNIT_SALE_PRICE)  as  Revenue
         from
            GOSALES..GOSALES.ORDER_HEADER ORDER_HEADER,
            GOSALES..GOSALES.ORDER_DETAILS ORDER_DETAILS
         where
            (ORDER_HEADER.ORDER_NUMBER = ORDER_DETAILS.ORDER_NUMBER)
        ) Sales_Fact
     where
        (TIME_DIMENSION.DAY_DATE = Sales_Fact.ORDER_DATE)
     group by
        TIME_DIMENSION.CURRENT_YEAR,
        TIME_DIMENSION.CURRENT_MONTH
    ) D2
     full outer join
    (select
        TIME_DIMENSION.CURRENT_YEAR  as  Year1,
        TIME_DIMENSION.CURRENT_MONTH  as  Month__numeric_,
        XSUM(Returns_Fact.Return_Quantity  for TIME_DIMENSION.CURRENT_YEAR,TIME_DIMENSION.CURRENT_MONTH )  as
```

   If you scroll through the SQL, you will see that there are two queries merged together using a full outer join.

9. Click **Close**, close **Report Studio**, and then close your browser.

Results:
By implementing a Time dimension, you can now easily query across fact query subjects. You did however discover that Sales Target values are not correct and will correct this later in the modeling process using determinants.



Business Analytics software

Resolve Multiple Ambiguous Joins

Recommendation #8

- Use role-playing time dimensions to resolve multiple ambiguous joins.

Sales Fact
Order Date = Day Key
Time
Ship Date = Day Key
Time (Ship)
Close Date = Day Key
Time (Close)

© 2012 IBM Corporation

A table with multiple valid relationships between itself and another table presents a reporting trap. Sales Fact, for example, has three possible ways of joining to the Time dimension. To ensure that queries use the correct relationship, create two additional aliases for the Time dimension and assign each a role: one links to Sales Fact by Order Date, one by Ship Date, and one by Close Date.

Demo 3: Resolve Multiple Ambiguous Joins

Purpose:
Report authors require queries based on shipping and closing dates as well as the standard order date. However, while analyzing your model design, you see that this causes multiple relationships between Sales Fact and Time. The proper join may not be performed at run time. You will resolve this ambiguity by creating role-playing dimensions for each relationship.

Component:	**Framework Manager**
Project:	**GO Operational**

Task 1. Use Model Advisor to Identify the Issue.

1. In **Foundation Objects View>gosales**, right-click **Sales Fact** to launch the **Context Explorer**, and then double-click the relationship to **TIME_DIMENSION**.

   If necessary, click **Show Related Objects** .

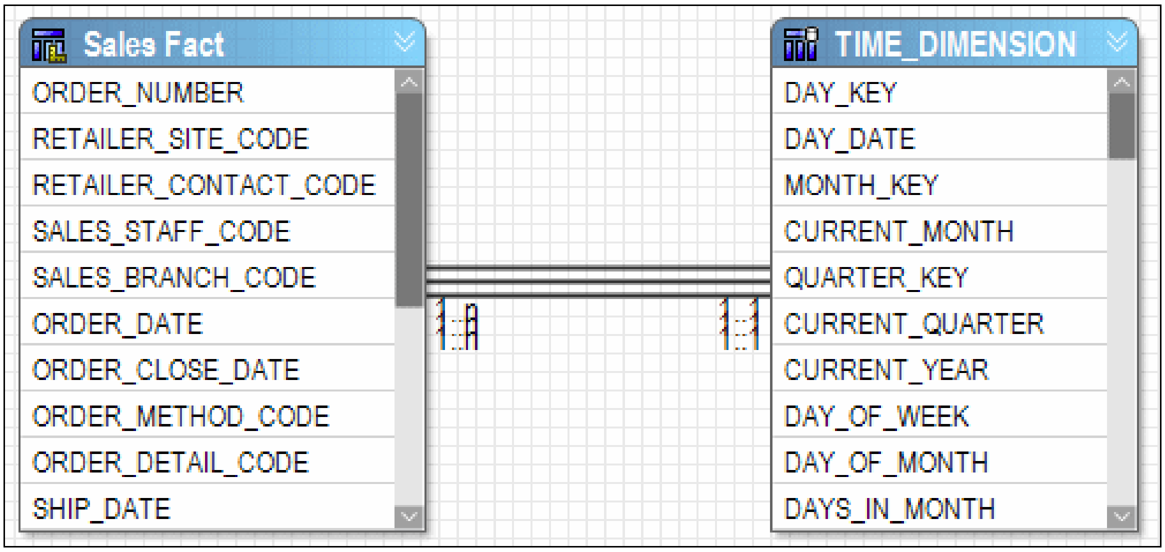
   Note that DAY_DATE is joined to ORDER_DATE, but should also have a relationship to ORDER_CLOSE_DATE and SHIP_DATE.

2. Click **Cancel**, close **Context Explorer**, and then create two more relationships between **TIME_DIMENSION** and **Sales Fact**:

   DAY_DATE (1..1) to **ORDER_CLOSE_DATE** (1..n)
   DAY_DATE (1..1) to **SHIP_DATE** (1..n)

3. Launch **Context Explorer** for **Sales Fact**, and then click **Show Related Objects**.

   The results appear as follows:



   Notice the multiple relationships between Sales Fact and TIME_DIMENSION. You cannot use the relationships in this state and get expected results. Which relationship should be used and when? There is not enough information for IBM Cognos to select the correct relationship.

   You will implement role playing time dimensions to handle ship dates and close dates.

4. Delete the two new relationships on **SHIP_DATE** and **ORDER_CLOSE_DATE**, and then close the **Context Explorer**.

Task 2. Create a Role-Playing Dimension.

1. In the **gosales** namespace, create a new model query subject called **Time (Close) (alias)**, and then click **OK**.
2. In the **Available Model Objects** pane, expand **Foundation Objects View> gosales>TIME_DIMENSION**, and then double-click to add the following query items to the definition:

   DAY_DATE

   CURRENT_YEAR
   QUARTER_KEY
   CURRENT_QUARTER
   MONTH_KEY
   CURRENT_MONTH
   MONTH_EN
   DAY_KEY
3. Click **OK**, and then drag **Time (Close) (alias)** below **TIME_DIMENSION**.
4. Create the following relationship, clicking **No** if prompted to recreate other relationships:

   Time (Close) (alias) (DAY_DATE, 1..1) to **Sales Fact** (ORDER_CLOSE_DATE, 1..n)
5. In **Time (Close) (alias)**, rename the following data items:

   DAY_DATE to **Close Date**

   CURRENT_YEAR to **Close Year**

   QUARTER_KEY to **Close Quarter Key**

   CURRENT_QUARTER to **Close Quarter**

   MONTH_KEY to **Close Month Key**

   CURRENT_MONTH to **Close Month (numeric)**

   MONTH_EN to **Close Month**

   DAY_KEY to **Close Day Key**

6. Create a new model query subject named **Time (Ship) (alias)**, with the following data items, renamed:

   DAY_DATE to **Ship Date**

   CURRENT_YEAR to **Ship Year**

   QUARTER_KEY to **Ship Quarter Key**

   CURRENT_QUARTER to **Ship Quarter**

   MONTH_KEY to **Ship Month Key**

   CURRENT_MONTH to **Ship Month (numeric)**

   MONTH_EN to **Ship Month**

   DAY_KEY to **Ship Day Key**

   Tip: You can make a copy of Time (Close) (alias) and edit it to save time.

7. Move **Time (Ship) (alias)** below **Time (Close) (alias**), and creating the following relationship:

   Time (Ship) (alias) (Ship Date, 1..1) to **Sales Fact** (SHIP_DATE, 1..n)

8. In the **Time (Close) (alias)** query subject, double-click **Close Month**, and delete the expression in the **Expression definition** box.

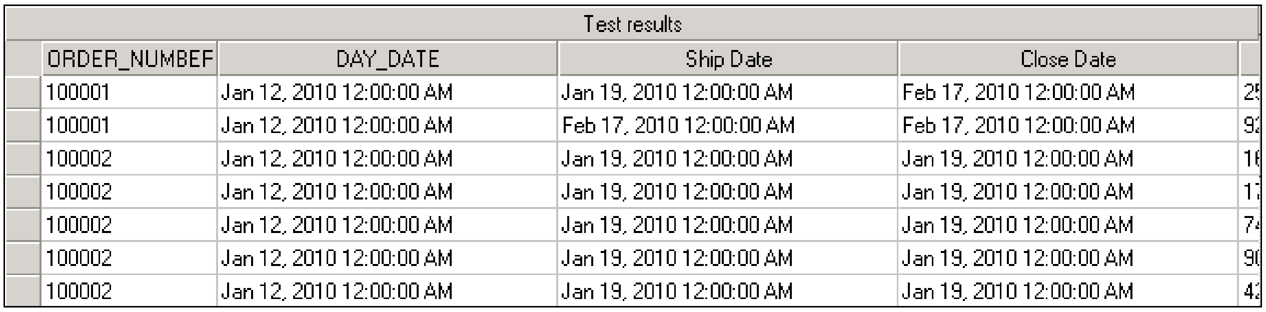
9. From **gosales>Reusable Objects>Model Calculations**, add the **Month** calculation.

10. In **Time (Ship) (alias)**, double-click **Ship Month**, and replace the existing expression with the **Month** calculation.

11. Select and test the following query items with Auto Sum enabled:

Query Subject	Query Item
Sales Fact	**ORDER_NUMBER**
TIME_DIMENSION	**DAY_DATE**
Time (Ship) (alias)	**Ship Date**
Time (Close) (alias)	**Close Date**
Sales Fact	**Quantity**

The results appear as follows:



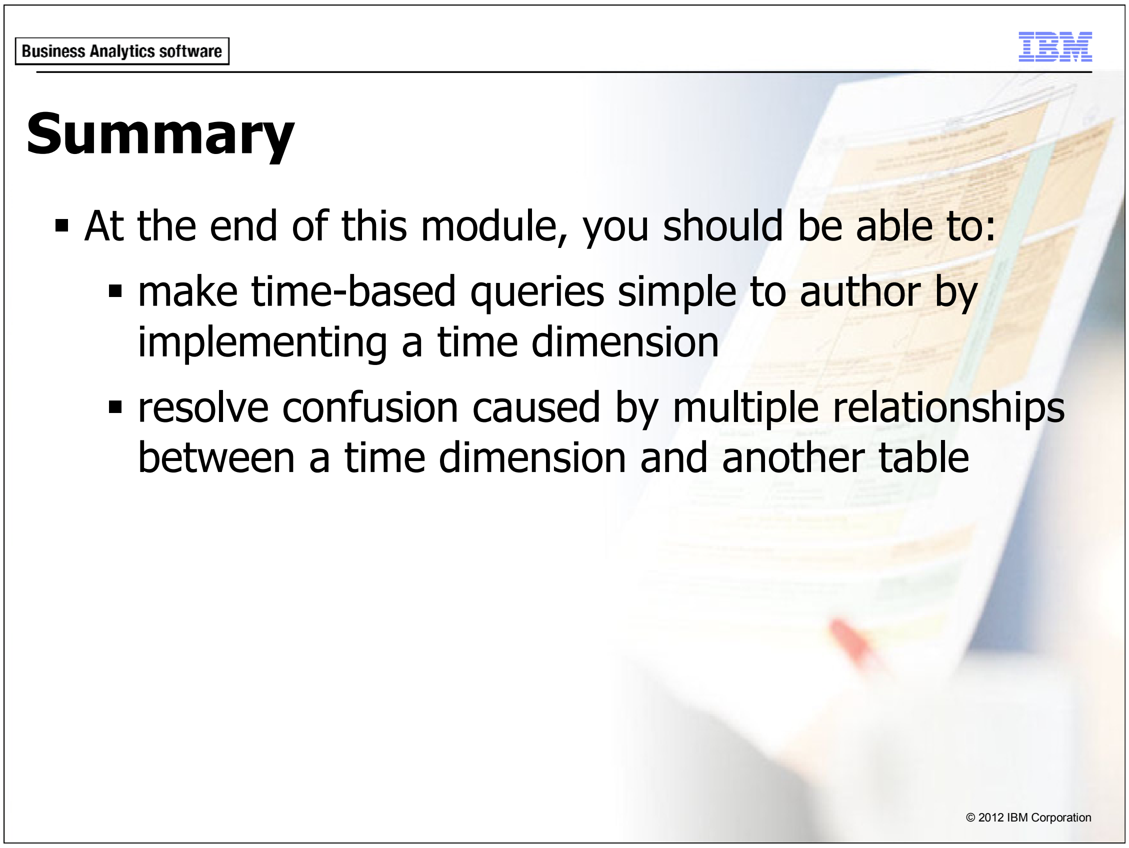
Test results

ORDER_NUMBEF	DAY_DATE	Ship Date	Close Date	
100001	Jan 12, 2010 12:00:00 AM	Jan 19, 2010 12:00:00 AM	Feb 17, 2010 12:00:00 AM	2
100001	Jan 12, 2010 12:00:00 AM	Feb 17, 2010 12:00:00 AM	Feb 17, 2010 12:00:00 AM	9
100002	Jan 12, 2010 12:00:00 AM	Jan 19, 2010 12:00:00 AM	Jan 19, 2010 12:00:00 AM	1
100002	Jan 12, 2010 12:00:00 AM	Jan 19, 2010 12:00:00 AM	Jan 19, 2010 12:00:00 AM	1
100002	Jan 12, 2010 12:00:00 AM	Jan 19, 2010 12:00:00 AM	Jan 19, 2010 12:00:00 AM	7
100002	Jan 12, 2010 12:00:00 AM	Jan 19, 2010 12:00:00 AM	Jan 19, 2010 12:00:00 AM	9
100002	Jan 12, 2010 12:00:00 AM	Jan 19, 2010 12:00:00 AM	Jan 19, 2010 12:00:00 AM	4

Notice that the dates in each column accurately represent the order dates, ship dates, and close dates. For most dates in this test result the close date is the same day as the ship date. But notice that the first record has a later close date. Larger orders do not always ship all products on the same day.

12. Click **Close**, save the project and then close Framework Manager.

Results:
You saw that multiple relationships between Sales Fact and Time meant that a proper join might not be performed at run time. You resolved this ambiguity by creating role-playing dimensions for the SHIP_DATE and ORDER_CLOSE_DATE relationships.

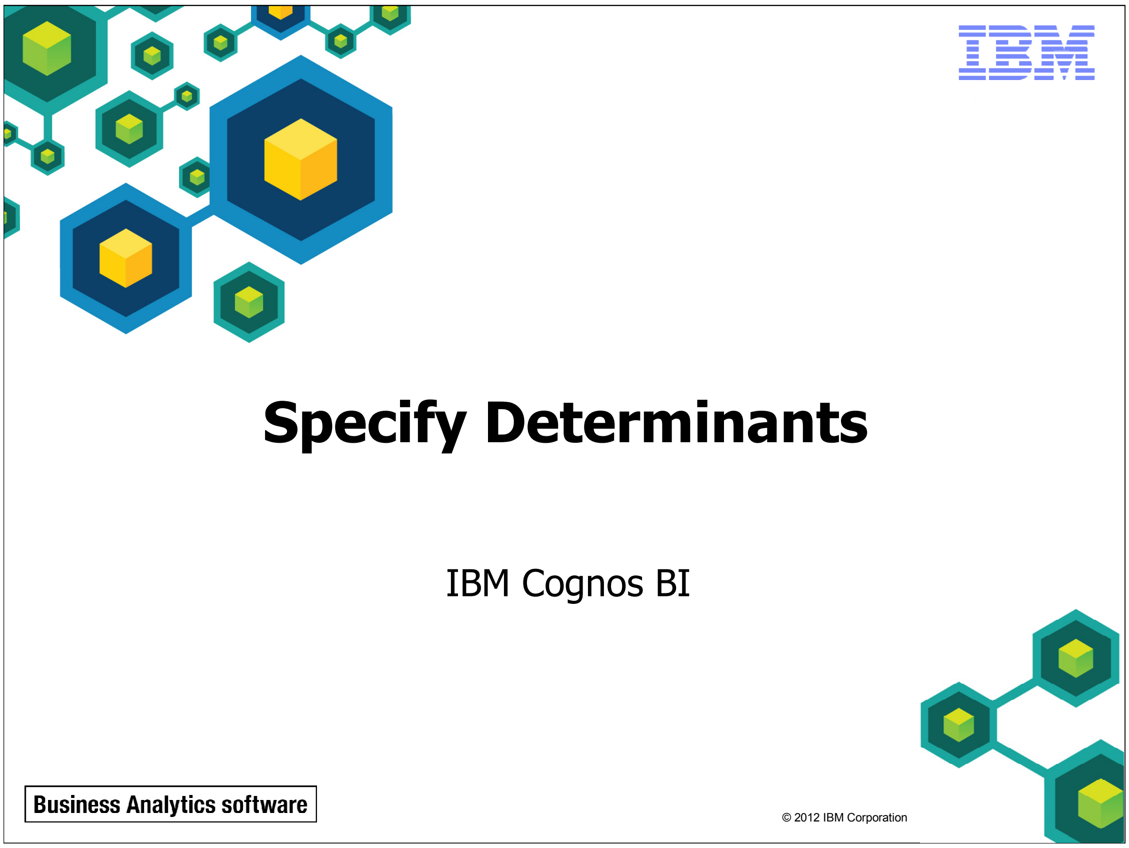


Business Analytics software

Summary

- At the end of this module, you should be able to:
  - make time-based queries simple to author by implementing a time dimension
  - resolve confusion caused by multiple relationships between a time dimension and another table

© 2012 IBM Corporation



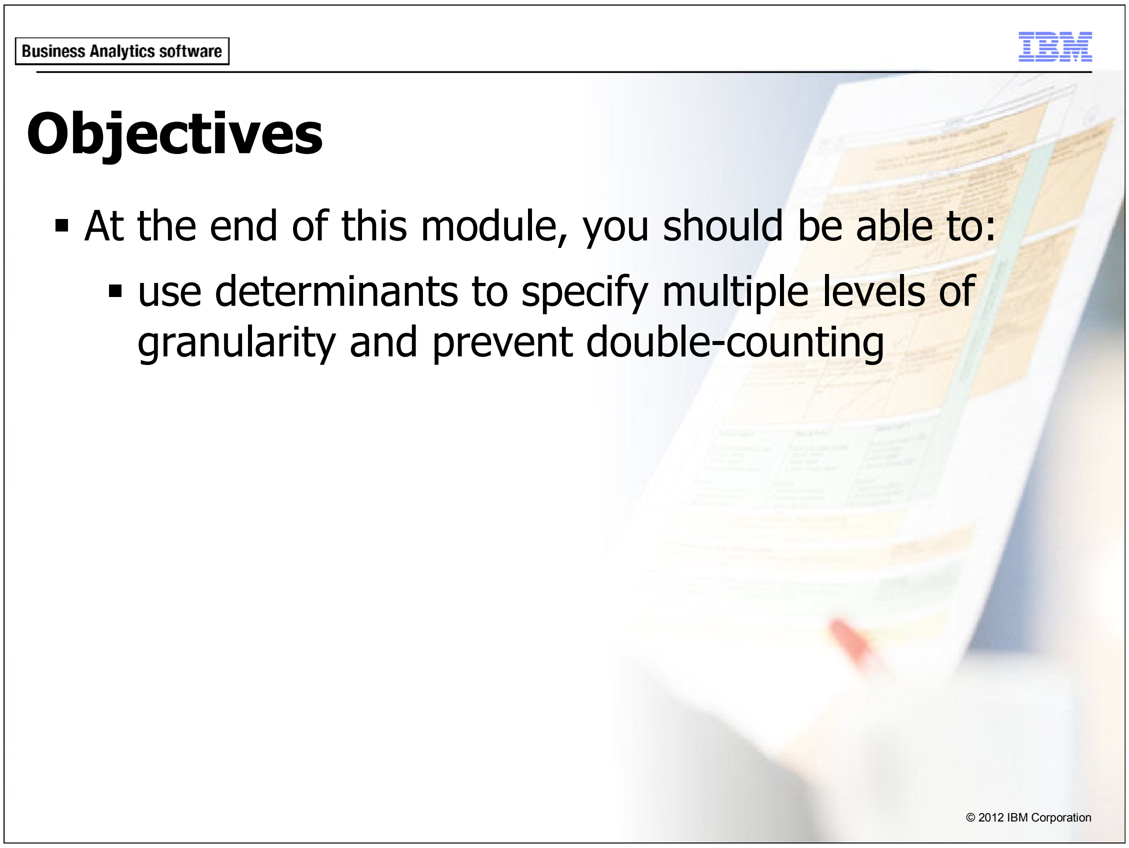
Specify Determinants

IBM Cognos BI

Business Analytics software

Unless otherwise specified in demo or workshop steps, you will always log on to IBM Cognos in the Local LDAP namespace using the following credentials:

- User ID: admin
- Password: Education1



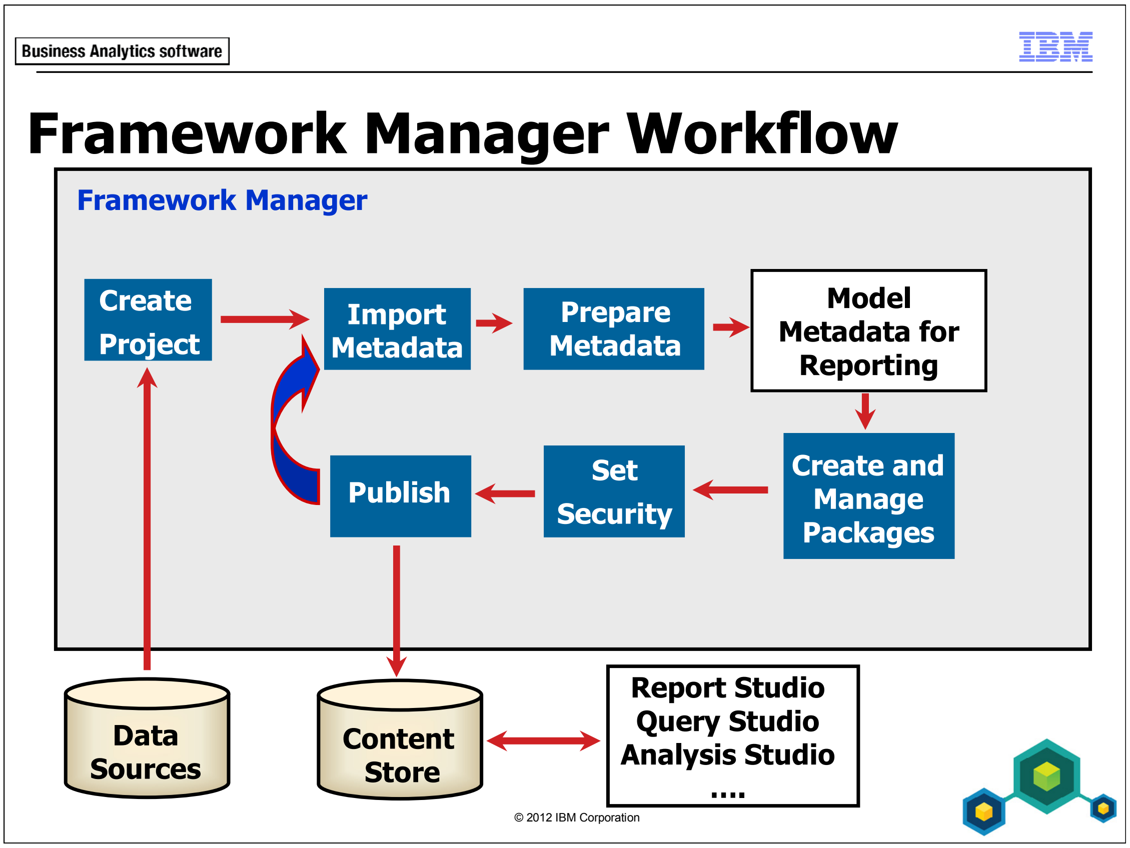
Business Analytics software

Objectives

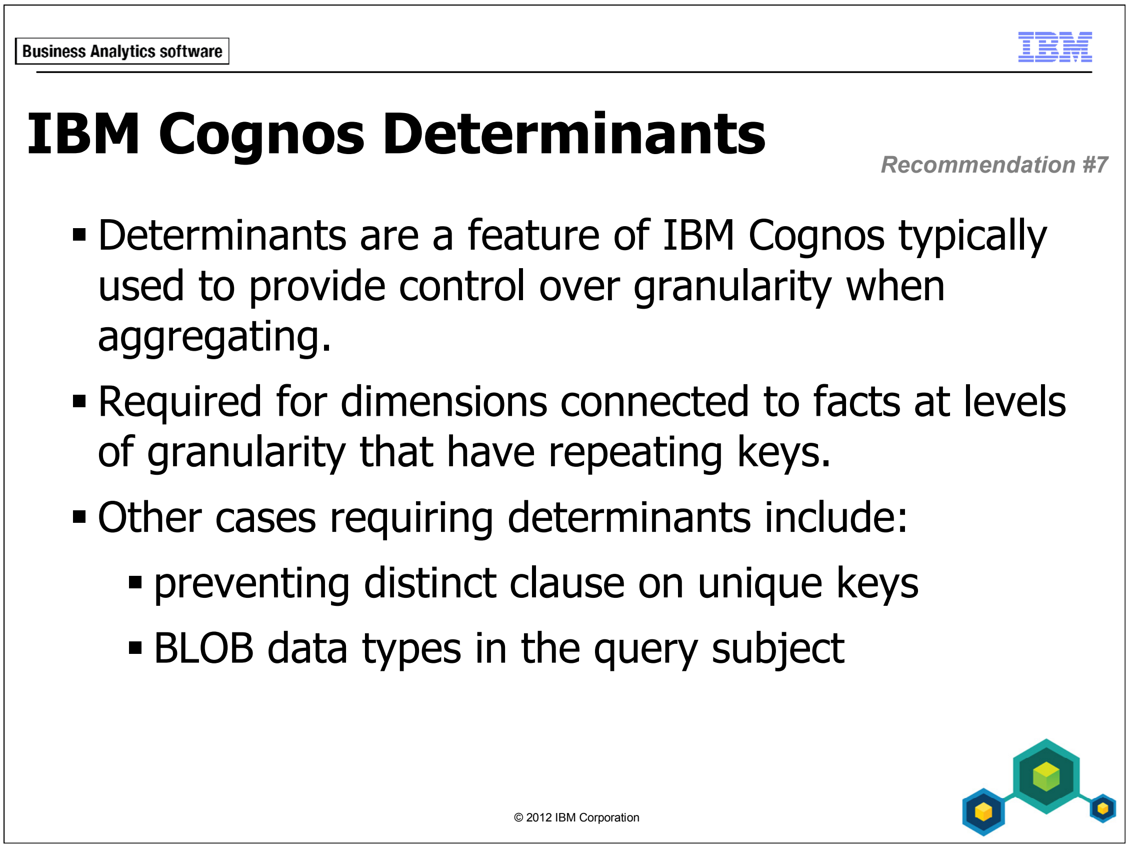
- At the end of this module, you should be able to:
  - use determinants to specify multiple levels of granularity and prevent double-counting

© 2012 IBM Corporation

Before reviewing this module, you should be familiar with IBM Cognos, IBM Cognos Connection, Query Studio and Report Studio. Suggested modules to reference:
• Overview of IBM Cognos
• Identify Common Data Structures
• Gather Requirements
• Create a Baseline Project
• Prepare Reusable Metadata
• Model for Predictable Results: Reporting Issues
• Model for Predictable Results: Virtual Star Schemas
• Model for Predictable Results: Consolidate Metadata
• Calculations and Filters
• Implement a Time Dimension



This module teaches the technique of specifying determinants to prevent double-counting.



Business Analytics software

IBM

IBM Cognos Determinants

Recommendation #7

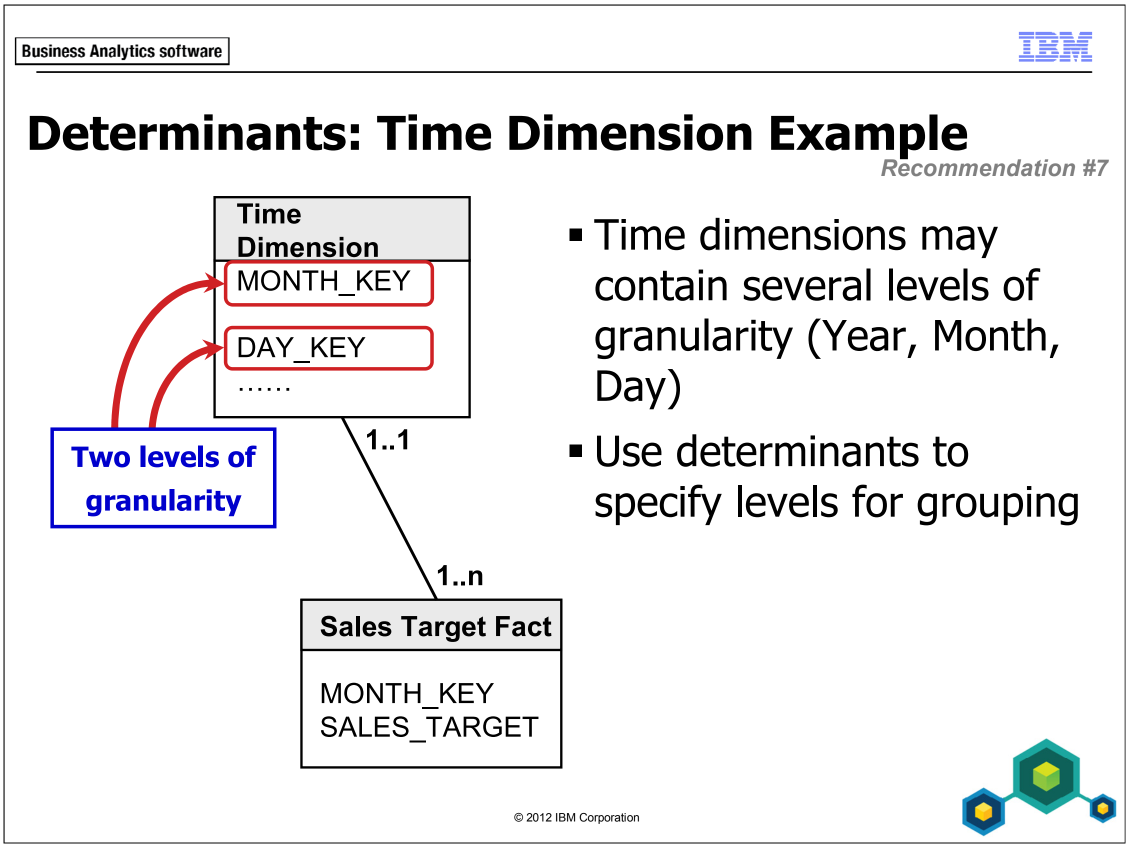
- Determinants are a feature of IBM Cognos typically used to provide control over granularity when aggregating.
- Required for dimensions connected to facts at levels of granularity that have repeating keys.
- Other cases requiring determinants include:
  - preventing distinct clause on unique keys
  - BLOB data types in the query subject

© 2012 IBM Corporation

Typically, determinants are specified to allow IBM Cognos to correctly aggregate facts in queries. Some designers also specify determinants proactively on any query subject with multiple levels of granularity, since you may link a fact at one of those levels of granularity in the future. Determinants are linked to the relationships their keys represent and are only used when the corresponding relationship is used in a query.

Determinants on unique data values (such as surrogate keys or dates), can prevent the generation of the distinct clause in the select statement when summarizing the data. Selecting distinct values unnecessarily can reduce performance. If the values are truly unique, then there is no need to scan for distinct values.

Querying Binary Large Objects (BLOB) requires additional key and index type information. If this information is not present in the data source, you can add it using determinants. To leverage indexes with DMR (especially when filtering on captions) associating your captions with the correct key in a determinant will improve query generation.



Business Analytics software

Determinants: Time Dimension Example

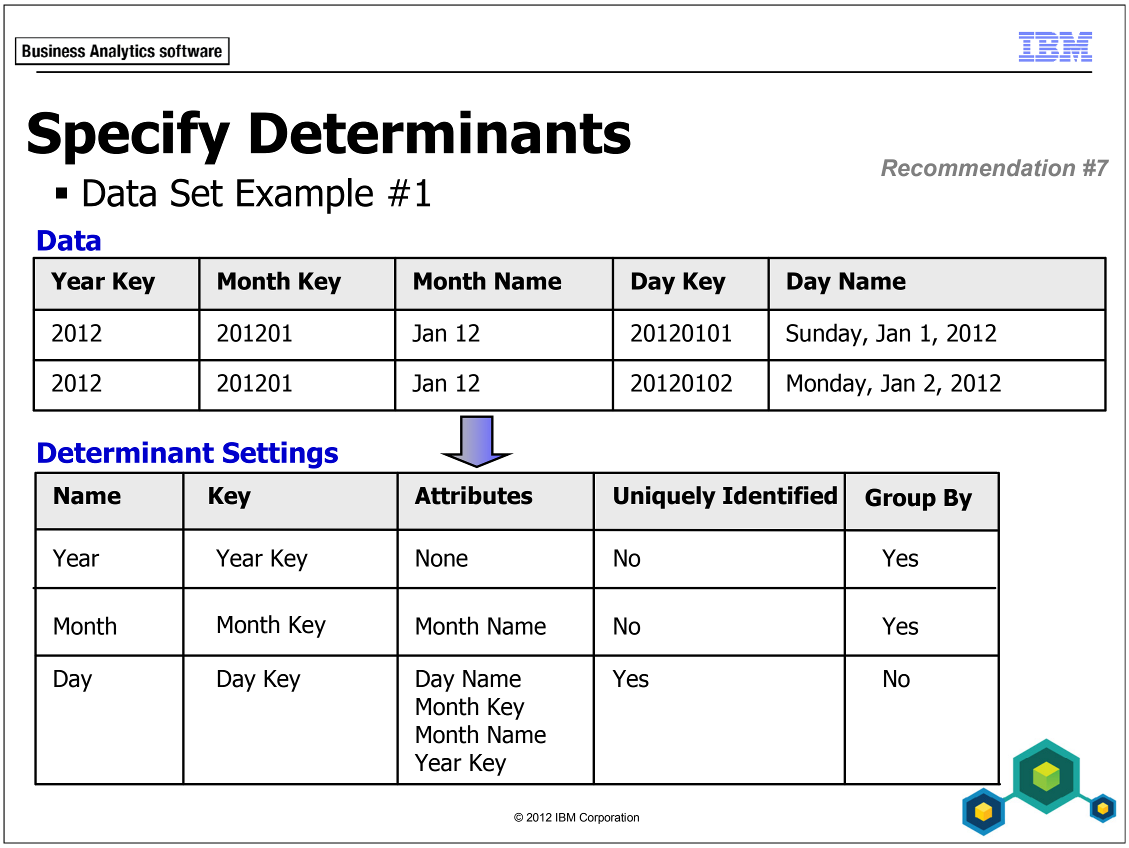
Recommendation #7

- Time dimensions may contain several levels of granularity (Year, Month, Day)
- Use determinants to specify levels for grouping

© 2012 IBM Corporation

In the slide example above, each unique instance of MONTH_KEY repeats once for every day in the month.

If you do not tell IBM Cognos that MONTH_KEY requires grouping at the month level, your queries will double-count SALES_TARGET for every day of the month. Setting a determinant at each level of granularity avoids any potential instances of double-counting.



Business Analytics software

IBM

Specify Determinants

Recommendation #7

- Data Set Example #1

Data

Year Key	Month Key	Month Name	Day Key	Day Name
2012	201201	Jan 12	20120101	Sunday, Jan 1, 2012
2012	201201	Jan 12	20120102	Monday, Jan 2, 2012

Determinant Settings

Name	Key	Attributes	Uniquely Identified	Group By
Year	Year Key	None	No	Yes
Month	Month Key	Month Name	No	Yes
Day	Day Key	Day Name Month Key Month Name Year Key	Yes	No

© 2012 IBM Corporation

For the data set above, it is possible to define three determinants, two non-unique determinants (based on Year Key and Month Key), and one unique determinant (based on Day Key).

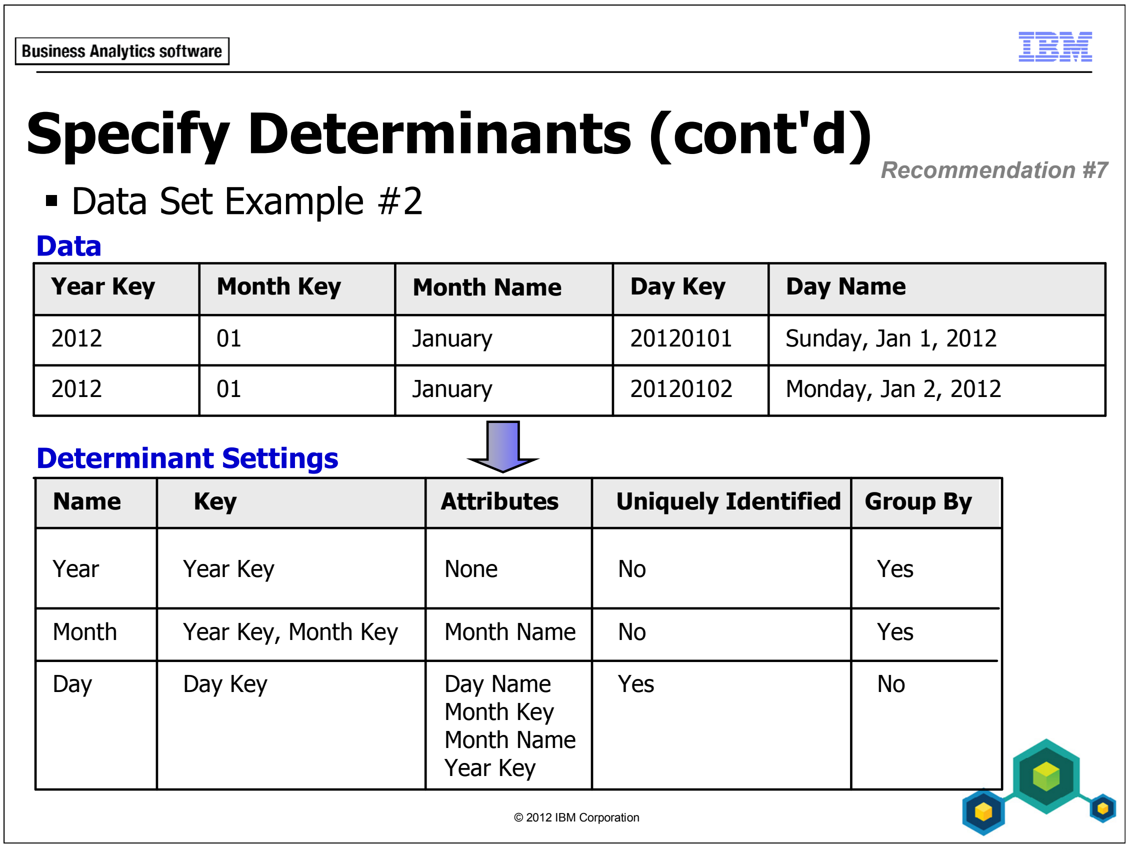
Day Key is the unique key of the table; therefore you can associate all the columns in the table to this key. Because it is a unique key at the lowest level of granularity, you check the Uniquely Identified setting and do not check the Group By setting.

Month Key is also a key but it is not unique since it repeats in the data (once for every day in the month). Uniquely Identified is not checked for this determinant. If you wanted to query month from this time table, you would write a query that used select min function syntax and group by the Month Key. That is why you check the Group By box in the determinant setting.

Similar logic is applied to the Year determinant.

When a query is created using a column from the table above, the IBM Cognos query engine evaluates the determinants, based on the relationship used in the query, one at a time until it finds the column reference (either the key or an attribute of the key) in a determinant and then stops when it finds it. For example, if you chose Month Name and Sales Target (Sales Target has a relationship to the Time dimension on the Month Key), IBM Cognos would evaluate the Year determinant and see that there is no reference to Month Name. It would then move on to the Month determinant and find the Month Name reference and stop evaluating. It would then generate the appropriate SQL and group on the Month Key.

Ensure that the key for each determinant includes the key or keys that identify the data set.



Business Analytics software

Specify Determinants (cont'd)

Recommendation #7

- Data Set Example #2

Data

Year Key	Month Key	Month Name	Day Key	Day Name
2012	01	January	20120101	Sunday, Jan 1, 2012
2012	01	January	20120102	Monday, Jan 2, 2012

Determinant Settings

Name	Key	Attributes	Uniquely Identified	Group By
Year	Year Key	None	No	Yes
Month	Year Key, Month Key	Month Name	No	Yes
Day	Day Key	Day Name Month Key Month Name Year Key	Yes	No

© 2012 IBM Corporation

In the event a key requires another key to provide uniqueness to identify the row of data, you need to nest the keys. For example, looking at the Month Key above, you can not identify which year it belongs to without looking at the Year Key. Therefore, the Month determinant nests the Year Key and the Month Key to generate the proper grouping in the SQL at run time. In this case the grouping would combine Year Key and Month Key.

Demo 1: Specify Determinants on the Time Dimension

Purpose:
A typical business report compares the monthly sales targets to actual sales. In a previous test you discovered that sales target values were being double-counted in report summary footers. You will specify determinants on TIME_DIMENSION to correct this problem.

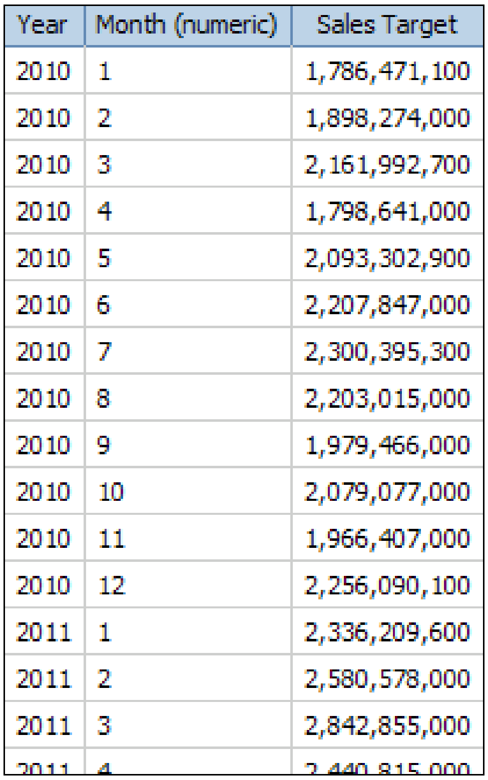
Components:	**Framework Manager**, **Query Studio**
Project:	**GO Operational**
Package:	**GO Operational**

Task 1. Review double-counting issue.

1. In **Framework Manager**, open the **GO Operational** project located at **C:\Edcognos\B5252\CBIFM-Start Files\Module 11\GO Operational**.
2. If prompted, log in as User ID **admin**, and Password **Education1**.
3. Launch **IBM Cognos Connection**, log on, and then click **IBM Cognos content**.

4. Click **GO Operational**, and then click **Test Time Dimension**.

   The results appear as follows:



Year	Month (numeric)	Sales Target
2010	1	1,786,471,100
2010	2	1,898,274,000
2010	3	2,161,992,700
2010	4	1,798,641,000
2010	5	2,093,302,900
2010	6	2,207,847,000
2010	7	2,300,395,300
2010	8	2,203,015,000
2010	9	1,979,466,000
2010	10	2,079,077,000
2010	11	1,966,407,000
2010	12	2,256,090,100
2011	1	2,336,209,600
2011	2	2,580,578,000
2011	3	2,842,855,000
2011	4	2,440,815,000

   Again, knowing your business, you can see that monthly sales targets are in the billions, which is too high. Double counting occurs because there is not enough information specified on the time dimension to allow IBM Cognos to roll up values correctly at the month. The target value is counted once for every day in a month rather than just once for each month. You will specify determinants for the underlying TIME_DIMENSION query subject to resolve this issue.

5. Click **Return** in the top right corner.

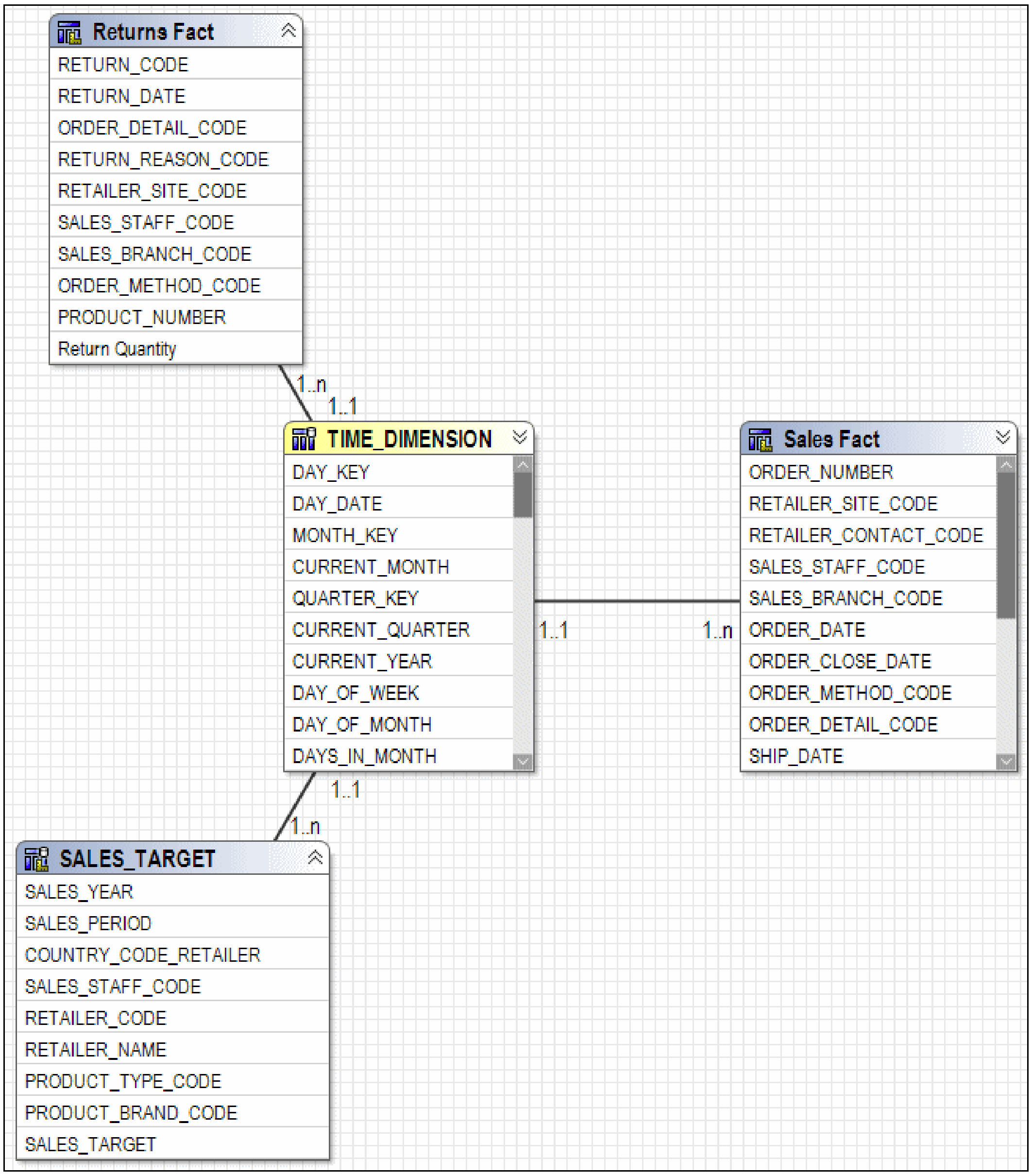
Task 2. Examine keys and unique values in TIME_DIMENSION.

1. In **Framework Manager**, in **Foundation Objects View>gosales**, right-click **TIME_DIMENSION**, and then click **Launch Context Explorer**.

2. Click **Show Related Objects**, click **Auto Layout**, ensure **Star** is selected, and then click **Apply**.

3. Click **Close**.

   The results appear similar to as follows:

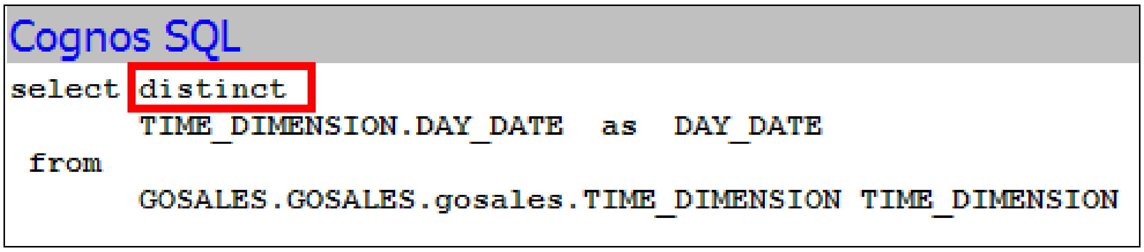


When specifying determinants, it is good practice to view the relationships to the object and examine the keys used in the relationships. In this case DAY_DATE, not DAY_KEY, is used to relate to Returns Fact and Sales Fact. This is an important piece of information for your determinant at the day level. Also, the relationship to SALES_TARGET is based on CURRENT_YEAR and CURRENT_MONTH. The determinants should reflect the relationships.

Before applying determinants on TIME_DIMENSION you will also quickly test effects of having no determinants on a unique value in the data. In this case DAY_DATE is unique.

4. Close **Context Explorer**, expand **TIME_DIMENSION** and then test the **DAY_DATE** query item with **Auto Sum** enabled.
5. Click the **Query Information** tab.

   The results appear as follows:



Cognos SQL

```
select distinct
       TIME_DIMENSION.DAY_DATE  as  DAY_DATE
 from
       GOSALES.GOSALES.gosales.TIME_DIMENSION TIME_DIMENSION
```

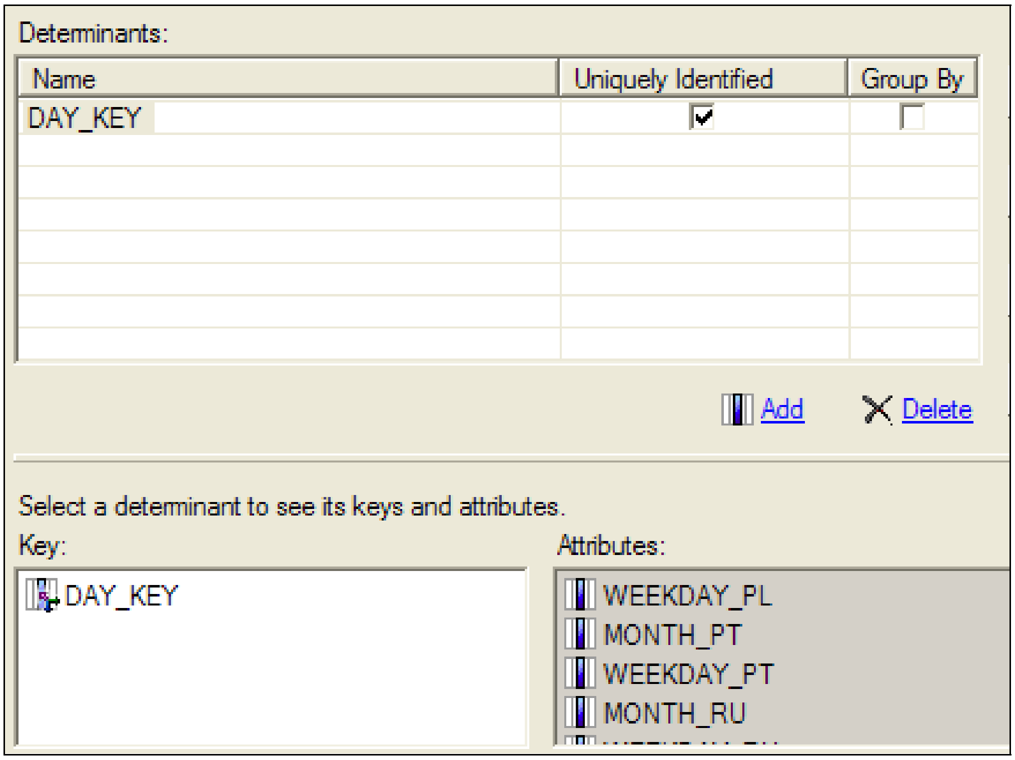
   Notice the distinct clause in the select statement. This is unnecessary as the DAY_DATE values in this table are all unique. You will now specify determinants to reflect your relationships and also observe that the distinct clause will not be applied when querying DAY_DATE.

6. Click **Close**.

Task 3. Specify determinants in TIME_DIMENSION.

1. In the **Project Viewer**, double-click **TIME_DIMENSION**, and then click the **Determinants** tab.

   The results appear as follows:



Determinants:

Name	Uniquely Identified	Group By
DAY_KEY	✓	☐

Select a determinant to see its keys and attributes.

Key:
- DAY_KEY

Attributes:
- WEEKDAY_PL
- MONTH_PT
- WEEKDAY_PT
- MONTH_RU

   During import, the primary key value is used to create a determinant for you. This value is unique and therefore all other values in the table can be associated with it. But in the case of your relationship to Sales Fact and Returns Fact, the DAY_KEY is not used in the relationship but rather DAY_DATE. You will alter this determinant to reflect your relationship.

2. Under **Key**, select **DAY_KEY**, and then press **Delete**.
3. Under **Available items**, drag **DAY_DATE** to the **Key** pane.

   You will now add the remaining determinants.

4. Under the **Determinants** pane, click **Add**.

   New Determinant appears below DAY_KEY in the Determinants pane.

5. Right-click **New Determinant**, and then click **Rename**.
6. Type **Year**, and then press **Enter**.
7. Click the **Up Arrow** key on the right to move **Year** above **DAY_KEY**.
8. From the **Available items** pane, drag **CURRENT_YEAR** to the **Key** pane.
9. Select the **Group By** check box beside **Year**.
10. Repeat steps **4** to **9** to create **Quarter**, with Key = **QUARTER_KEY**.
11. With the focus still on **Quarter**, drag **CURRENT_QUARTER** into the **Attributes** box.
12. Repeat steps **4** to **9** to create **Month**, with two Keys = **CURRENT_MONTH** and **CURRENT_YEAR**.

    CURRENT_MONTH and CURRENT_YEAR are used as the key for this determinant since the relationship to SALES_TARGET is based on those two query items. These two fields uniquely identify the relationship to SALES_TARGET and therefore will be used to correctly aggregate sales target values.

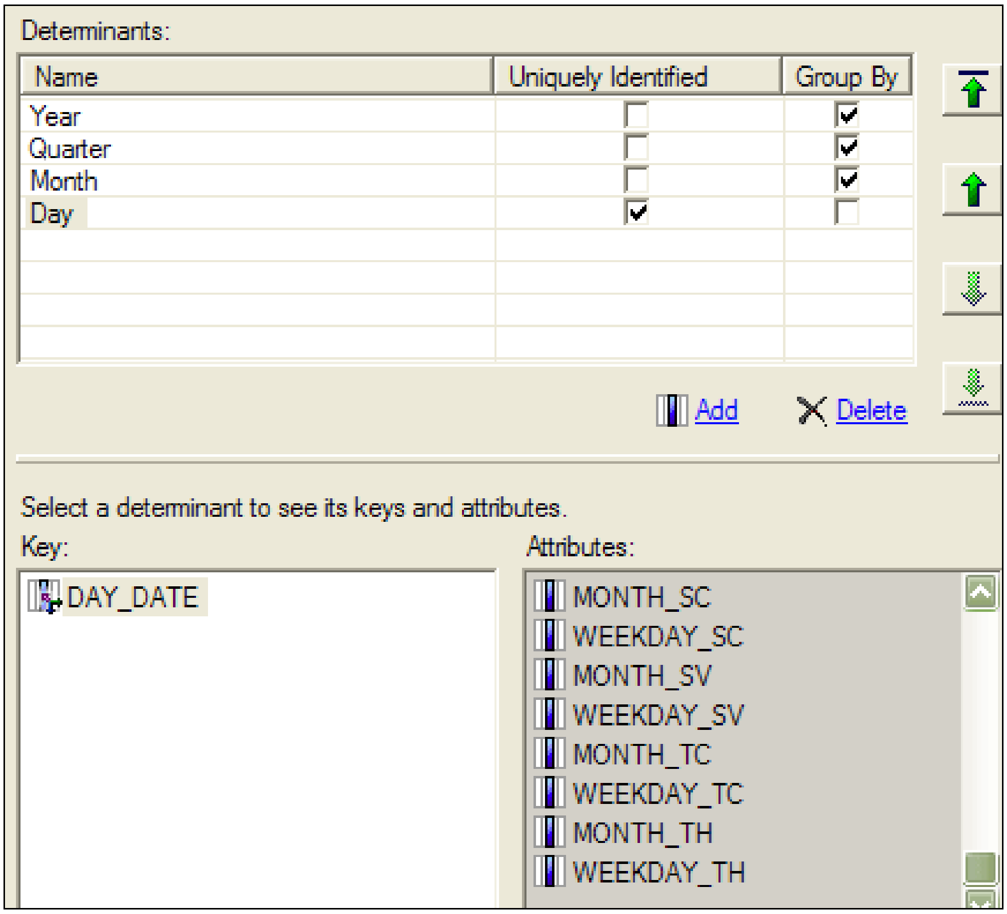
13. With the focus still on **Month**, drag the following items into the **Attributes** box: **MONTH_KEY** and all **MONTH_xx** items.

    Tip: Select the items using the Ctrl key. Then right-click one of the items and then click Add as Attributes.

    Typically MONTH_KEY would be used as the key for a determinant describing month data values. In this case however, MONTH_KEY is not used in any relationships, and therefore can act as an attribute of the Month determinant should the value be required in a report. If in the future you use this key in a relationship, you would need to reflect this in your determinants. You would need to create a new determinant to represent that relationship. Again, the ideal situation is to use a month key exclusively and consistently across all facts that report at the month level. These types of keys should be requested from the database administrator if they do not exist.

14. Rename the **DAY_KEY** determinant to **Day**.

    The results appear as follows:



Determinants:

Name	Uniquely Identified	Group By
Year	☐	✓
Quarter	☐	✓
Month	☐	✓
Day	✓	☐

Select a determinant to see its keys and attributes.

Key:

DAY_DATE

Attributes:

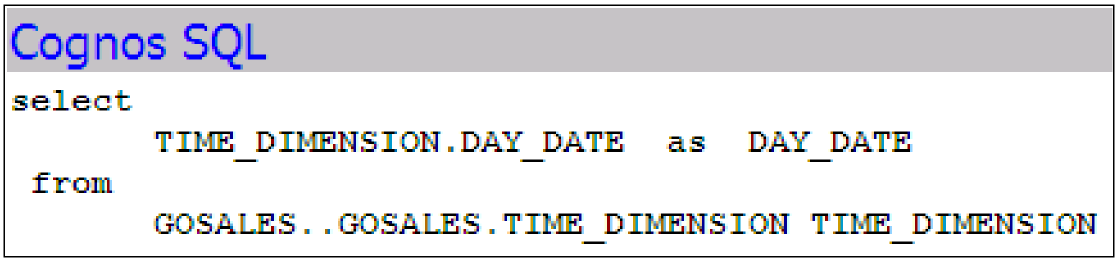
MONTH_SC
WEEKDAY_SC
MONTH_SV
WEEKDAY_SV
MONTH_TC
WEEKDAY_TC
MONTH_TH
WEEKDAY_TH

15. Click **OK**, and then save the project.

Task 4. Re-test DAY_DATE.

1. Under **TIME_DIMENSION**, test **DAY_DATE** with **Auto Sum** enabled, and then click the **Query Information** tab.

   The results appear as follows:



Cognos SQL

```
select
       TIME_DIMENSION.DAY_DATE  as  DAY_DATE
 from
       GOSALES..GOSALES.TIME_DIMENSION TIME_DIMENSION
```

   Notice that there is no longer a distinct clause in your select statement. This is because you have identified the DAY_DATE as the key and as being unique, therefore the IBM Cognos query engine has enough information to know it does not need to scan the table for distinct values. If you required another relationship later on using the DAY_KEY, then you would also specify a determinant on it and set it as unique. It is OK to have more than one unique determinant if they are truly unique. At query time, the relationship being used will determine which unique determent to use. If more than one relationship to a unique determinant is used, then each related determinant will be used in the query.

2. Click **Close**.

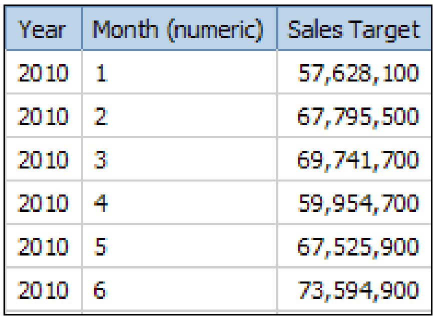
Task 5. Publish and Test the Package.

1. Publish the **GO Operational** package.
2. In **IBM Cognos Connection**, click **Test Time Dimension** to run the report again.

Task 4: Determinants and relationships work together. If a query uses Date (attribute of the Day determinant) with Revenue from Sales Fact (which is at the day level), then there will be no grouping for Date since the determinant is specified as unique. However, if you query Month (attribute of the Month determinant) and Sales Target (which is at the month level), then the query will be grouped by CURRENT_YEAR and CURRENT_MONTH as specified by the Month determinant.

3. When prompted to update the report, click **OK**.

   The results appear as follows:



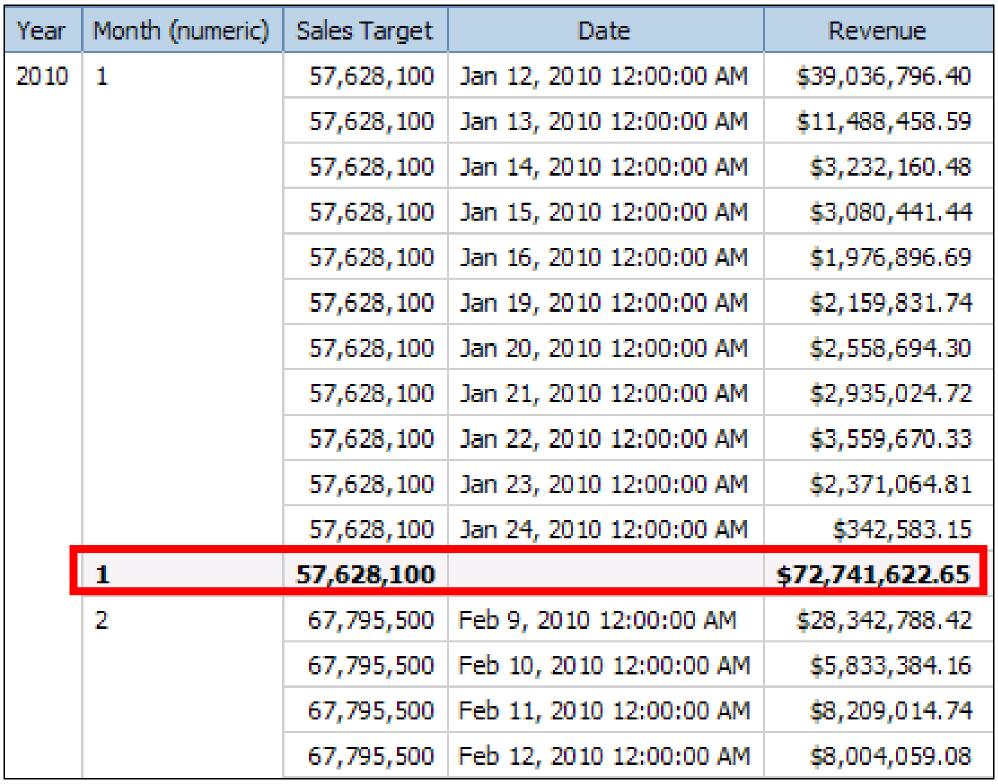
Year	Month (numeric)	Sales Target
2010	1	57,628,100
2010	2	67,795,500
2010	3	69,741,700
2010	4	59,954,700
2010	5	67,525,900
2010	6	73,594,900

   You now have accurate sales targets.

4. Under **Menu**, click **Insert Data**.
5. From the **Consolidation View**, add **Time**>**Date** and **Sales Fact**>**Revenue** to the report.

6. Select **Year** and **Month (numeric)** and click **Group** .

   The results appear as follows:



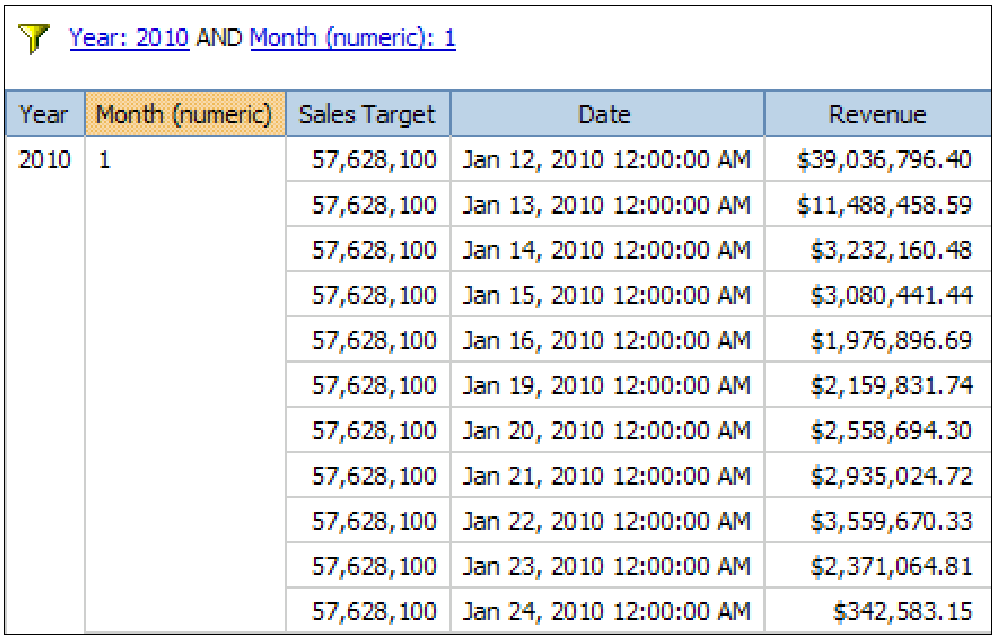
Year	Month (numeric)	Sales Target	Date	Revenue
2010	1	57,628,100	Jan 12, 2010 12:00:00 AM	$39,036,796.40
		57,628,100	Jan 13, 2010 12:00:00 AM	$11,488,458.59
		57,628,100	Jan 14, 2010 12:00:00 AM	$3,232,160.48
		57,628,100	Jan 15, 2010 12:00:00 AM	$3,080,441.44
		57,628,100	Jan 16, 2010 12:00:00 AM	$1,976,896.69
		57,628,100	Jan 19, 2010 12:00:00 AM	$2,159,831.74
		57,628,100	Jan 20, 2010 12:00:00 AM	$2,558,694.30
		57,628,100	Jan 21, 2010 12:00:00 AM	$2,935,024.72
		57,628,100	Jan 22, 2010 12:00:00 AM	$3,559,670.33
		57,628,100	Jan 23, 2010 12:00:00 AM	$2,371,064.81
		57,628,100	Jan 24, 2010 12:00:00 AM	$342,583.15
	1	**57,628,100**		**$72,741,622.65**
	2	67,795,500	Feb 9, 2010 12:00:00 AM	$28,342,788.42
		67,795,500	Feb 10, 2010 12:00:00 AM	$5,833,384.16
		67,795,500	Feb 11, 2010 12:00:00 AM	$8,209,014.74
		67,795,500	Feb 12, 2010 12:00:00 AM	$8,004,059.08

   Note that although the sales targets repeat (because they are not at the day level) they are not double counted at the Month grouping level.

7. Select the **Year** column, and then click **Filter**.
8. Set the **From** and **To** boxes to **2010**, and then click **OK**.
9. Select **Month (numeric)** column, and then click **Filter**.
10. Set the **From** and **To** boxes to **1**, and then click **OK**.

11. At the bottom of the window, click **Apply**.

    The results appear similar to the following:



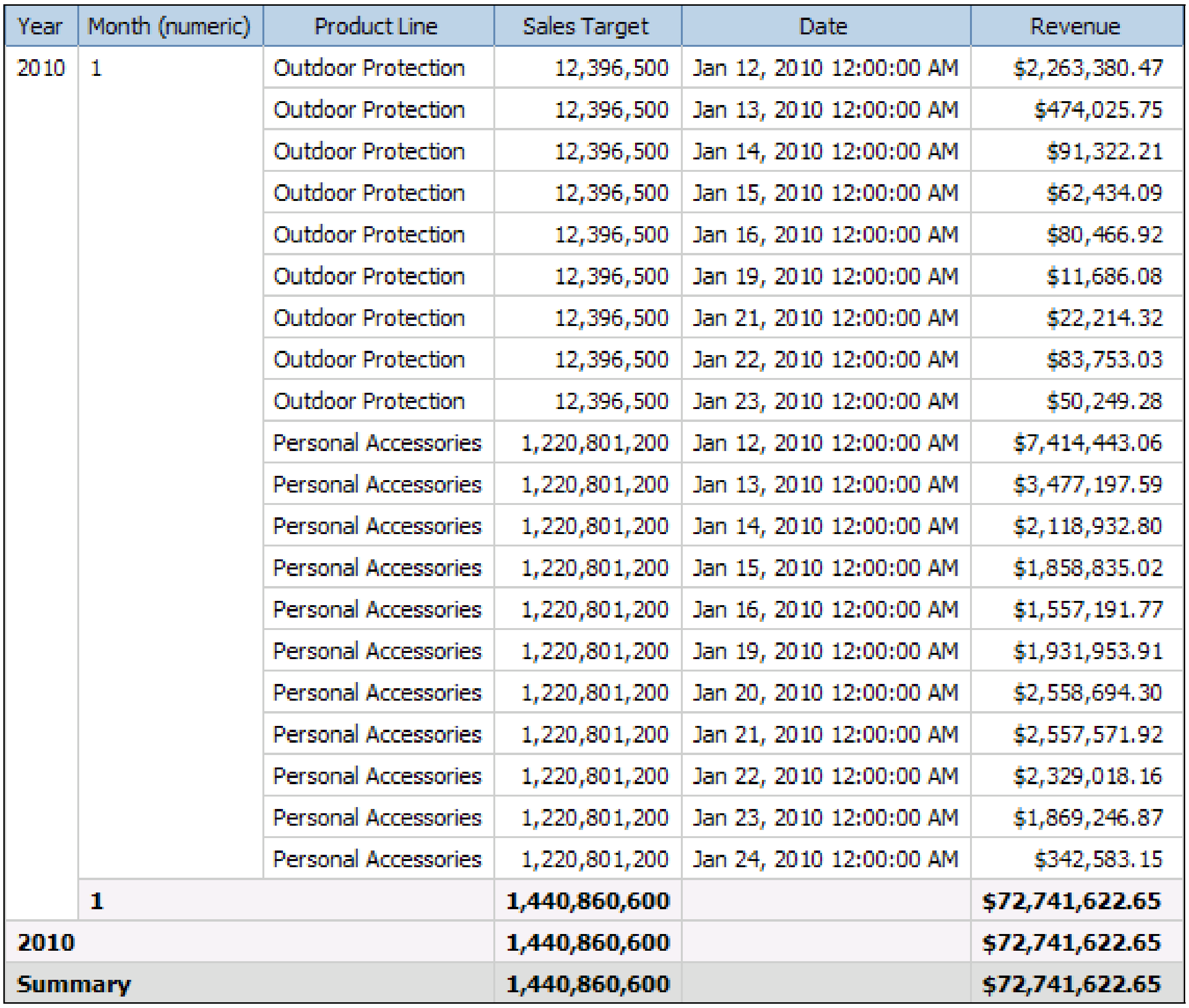
Year: 2010 AND Month (numeric): 1

Year	Month (numeric)	Sales Target	Date	Revenue
2010	1	57,628,100	Jan 12, 2010 12:00:00 AM	$39,036,796.40
		57,628,100	Jan 13, 2010 12:00:00 AM	$11,488,458.59
		57,628,100	Jan 14, 2010 12:00:00 AM	$3,232,160.48
		57,628,100	Jan 15, 2010 12:00:00 AM	$3,080,441.44
		57,628,100	Jan 16, 2010 12:00:00 AM	$1,976,896.69
		57,628,100	Jan 19, 2010 12:00:00 AM	$2,159,831.74
		57,628,100	Jan 20, 2010 12:00:00 AM	$2,558,694.30
		57,628,100	Jan 21, 2010 12:00:00 AM	$2,935,024.72
		57,628,100	Jan 22, 2010 12:00:00 AM	$3,559,670.33
		57,628,100	Jan 23, 2010 12:00:00 AM	$2,371,064.81
		57,628,100	Jan 24, 2010 12:00:00 AM	$342,583.15

This will give you a smaller data set to conduct your next test. You will now add Product Line from Products to see the results.

12. From **Products**, drag **Product Line** onto the report after **Month (numeric)**.
13. In the bottom of the window, click the **Bottom** link.

    The results appear similar as follows:

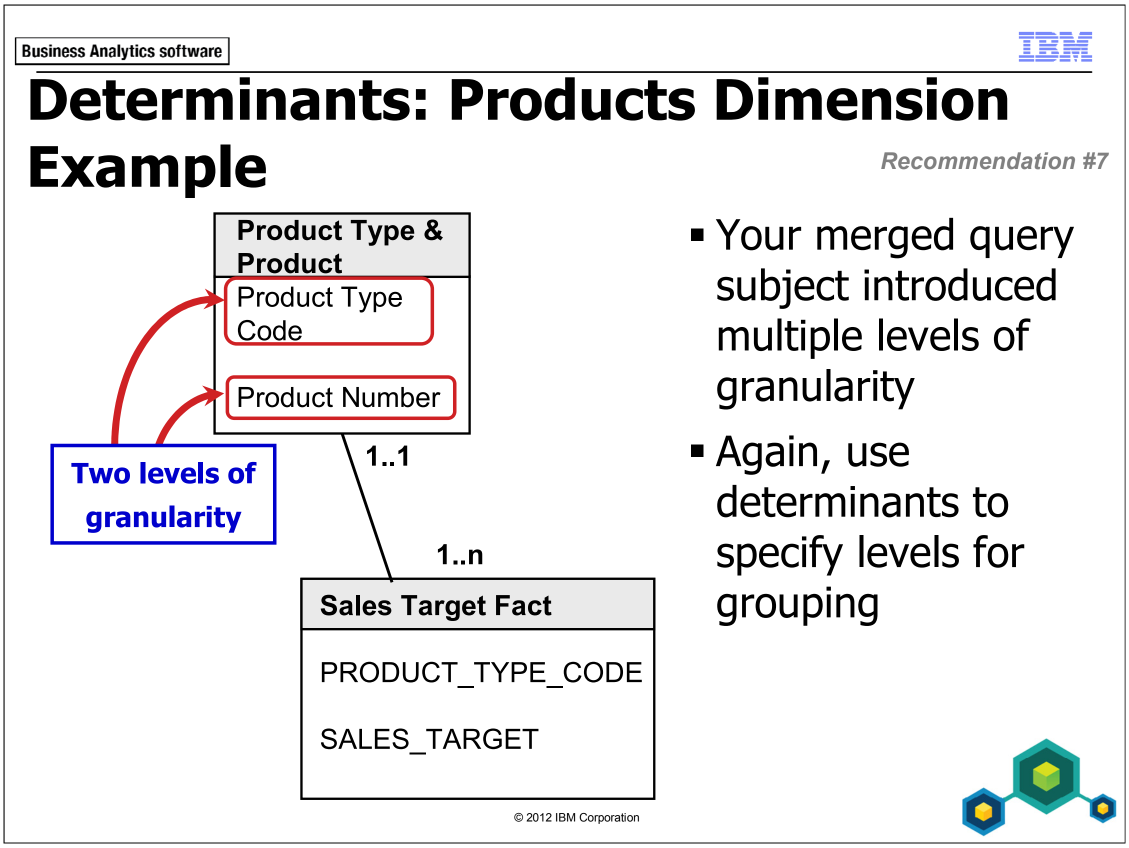


Year	Month (numeric)	Product Line	Sales Target	Date	Revenue
2010	1	Outdoor Protection	12,396,500	Jan 12, 2010 12:00:00 AM	$2,263,380.47
		Outdoor Protection	12,396,500	Jan 13, 2010 12:00:00 AM	$474,025.75
		Outdoor Protection	12,396,500	Jan 14, 2010 12:00:00 AM	$91,322.21
		Outdoor Protection	12,396,500	Jan 15, 2010 12:00:00 AM	$62,434.09
		Outdoor Protection	12,396,500	Jan 16, 2010 12:00:00 AM	$80,466.92
		Outdoor Protection	12,396,500	Jan 19, 2010 12:00:00 AM	$11,686.08
		Outdoor Protection	12,396,500	Jan 21, 2010 12:00:00 AM	$22,214.32
		Outdoor Protection	12,396,500	Jan 22, 2010 12:00:00 AM	$83,753.03
		Outdoor Protection	12,396,500	Jan 23, 2010 12:00:00 AM	$50,249.28
		Personal Accessories	1,220,801,200	Jan 12, 2010 12:00:00 AM	$7,414,443.06
		Personal Accessories	1,220,801,200	Jan 13, 2010 12:00:00 AM	$3,477,197.59
		Personal Accessories	1,220,801,200	Jan 14, 2010 12:00:00 AM	$2,118,932.80
		Personal Accessories	1,220,801,200	Jan 15, 2010 12:00:00 AM	$1,858,835.02
		Personal Accessories	1,220,801,200	Jan 16, 2010 12:00:00 AM	$1,557,191.77
		Personal Accessories	1,220,801,200	Jan 19, 2010 12:00:00 AM	$1,931,953.91
		Personal Accessories	1,220,801,200	Jan 20, 2010 12:00:00 AM	$2,558,694.30
		Personal Accessories	1,220,801,200	Jan 21, 2010 12:00:00 AM	$2,557,571.92
		Personal Accessories	1,220,801,200	Jan 22, 2010 12:00:00 AM	$2,329,018.16
		Personal Accessories	1,220,801,200	Jan 23, 2010 12:00:00 AM	$1,869,246.87
		Personal Accessories	1,220,801,200	Jan 24, 2010 12:00:00 AM	$342,583.15
	1		**1,440,860,600**		**$72,741,622.65**
2010			**1,440,860,600**		**$72,741,622.65**
Summary			**1,440,860,600**		**$72,741,622.65**

    The sales target values are double-counted. You will examine and correct this issue in the next workshop.

14. Save the report as **Products Dimension Test**, and then close your browser.

Results:
By specifying determinants on TIME_DIMENSION, you were able to correct a double-counting problem that occurred with sales targets.



Business Analytics software

Determinants: Products Dimension Example

Recommendation #7

Product Type & Product
Product Type Code
Product Number
Two levels of granularity
1..1
1..n
Sales Target Fact
PRODUCT_TYPE_CODE
SALES_TARGET

- Your merged query subject introduced multiple levels of granularity
- Again, use determinants to specify levels for grouping

© 2012 IBM Corporation

In the slide example above, the Product Type & Product query subject is a query subject you merged earlier to remove ambiguity from PRODUCT_TYPE. Each unique instance of Product Type Code repeats once for every related product. If you do not specify determinants, sales target values will be double-counted for every product belonging to a particular product type.

Again, you need to provide information to the IBM Cognos query engine that indicates Product Type Code requires grouping when queried against sales targets.

Workshop 1: Specify Determinants

Earlier in the modeling process, you merged some query subjects together to resolve ambiguous query subjects. You merged PRODUCT_TYPE and PRODUCT to create Product Type & Product, and you merged RETAILER and RETAILER_SITE to create Retailer & Retailer Site. Both these query subjects now present multiple levels of granularity in them and cause double-counting when querying against sales targets.

You will resolve this by specifying determinants on:

- Product Type & Product
    - PRODUCT_NUMBER is unique
    - Group on PRODUCT_TYPE_CODE and add appropriate attributes
- Retailer & Retailer Site
    - RETAILER_SITE_CODE is unique
    - Group on RETAILER_CODE and add appropriate attributes
- Save the project when you are finished and close Framework Manager.

Test the Product Type & Product determinants using the Product Dimension Test report in IBM Cognos Connection.

For more detailed information outlined as tasks, see the Task Table on the next page.

To see the desired filters, see the Workshop Results section that follows the Task Table.

Workshop 1: Task Table

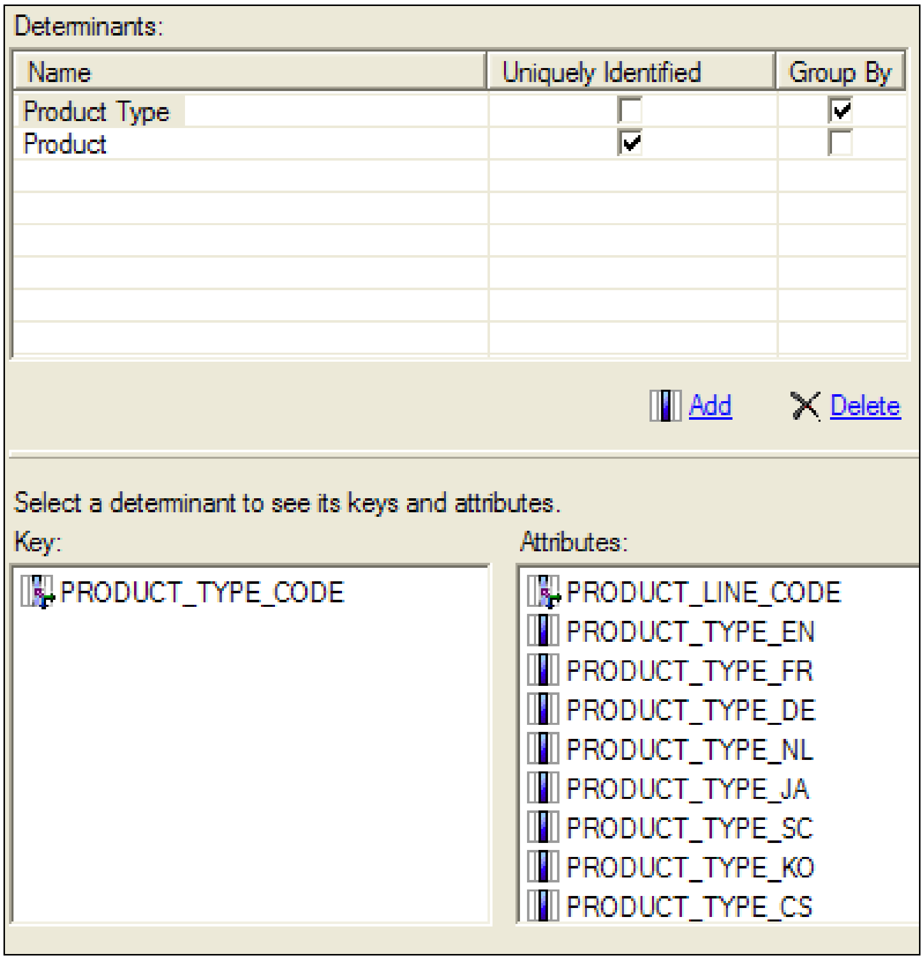
Task	Where to Work	Hints
1. Specify determinants for Product Type & Product.	Project Viewer, Foundation Objects view	• Double-click the Product Type & Product query subject • Click the Determinants tab • Click Add • Name the new determinant Product • Add PRODUCT_NUMBER as the key • Select Uniquely Identified checkbox • Add new determinant called Product Type • Add PRODUCT_TYPE_CODE as the key • Add PRODUCT_LINE_CODE and all PRODUCT_TYPE_xx items to the Attributes pane • Select Group By check box • Move Product Type determinant to the top

Task	Where to Work	Hints
2. Specify determinants for Retailer & Retailer Site.	Project Viewer, Foundation Objects view	• Double-click the Retailer & Retailer Site query subject • Click the Determinants tab • Click Add • Name determinant Retailer Site • Add RETAILER_SITE_CODE as the key • Select Uniquely Identified checkbox • Add new determinant called Retailer • Add RETAILER_CODE as the key • Add COMPANY_NAME, COMPANY_NAME_MB, RETAILER_TYPE_CODE, and RETAILER_START_DATE to the Attributes pane • Select Group By check box • Move Retailer determinant to the top • Save project

Task	Where to Work	Hints
3. Test Products Dimension Test report.	IBM Cognos Connection	• Publish Go Operational package • Launch IBM Cognos Connection and run Products Dimension Test • Close IBM Cognos Connection without saving and close Framework Manager, saving changes.

Workshop 1: Workshop Results

Your Product Type & Product determinants should appear as shown below:


Determinants:

Name	Uniquely Identified	Group By
Product Type	☐	✓
Product	✓	☐

Select a determinant to see its keys and attributes.

Key:

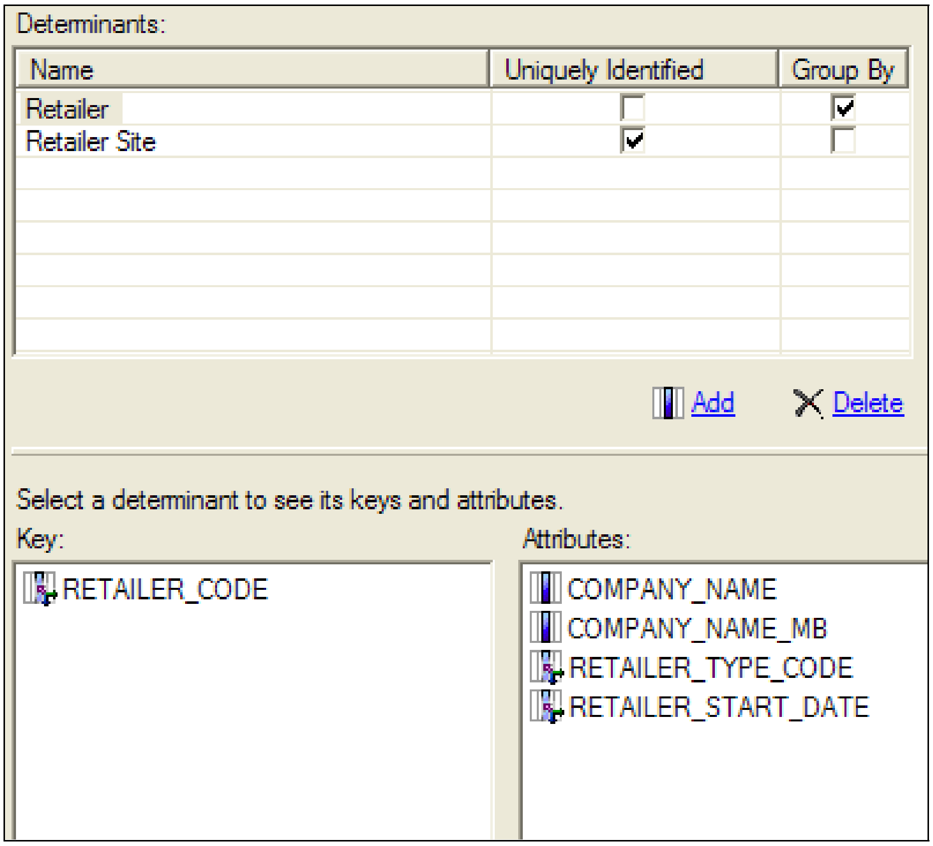
- PRODUCT_TYPE_CODE

Attributes:

- PRODUCT_LINE_CODE
- PRODUCT_TYPE_EN
- PRODUCT_TYPE_FR
- PRODUCT_TYPE_DE
- PRODUCT_TYPE_NL
- PRODUCT_TYPE_JA
- PRODUCT_TYPE_SC
- PRODUCT_TYPE_KO
- PRODUCT_TYPE_CS

Workshop 1: Workshop Results

Your Retailer & Retailer Site determinants should appear as shown below:



Determinants:

Name	Uniquely Identified	Group By
Retailer	☐	✓
Retailer Site	✓	☐

Select a determinant to see its keys and attributes.

Key:

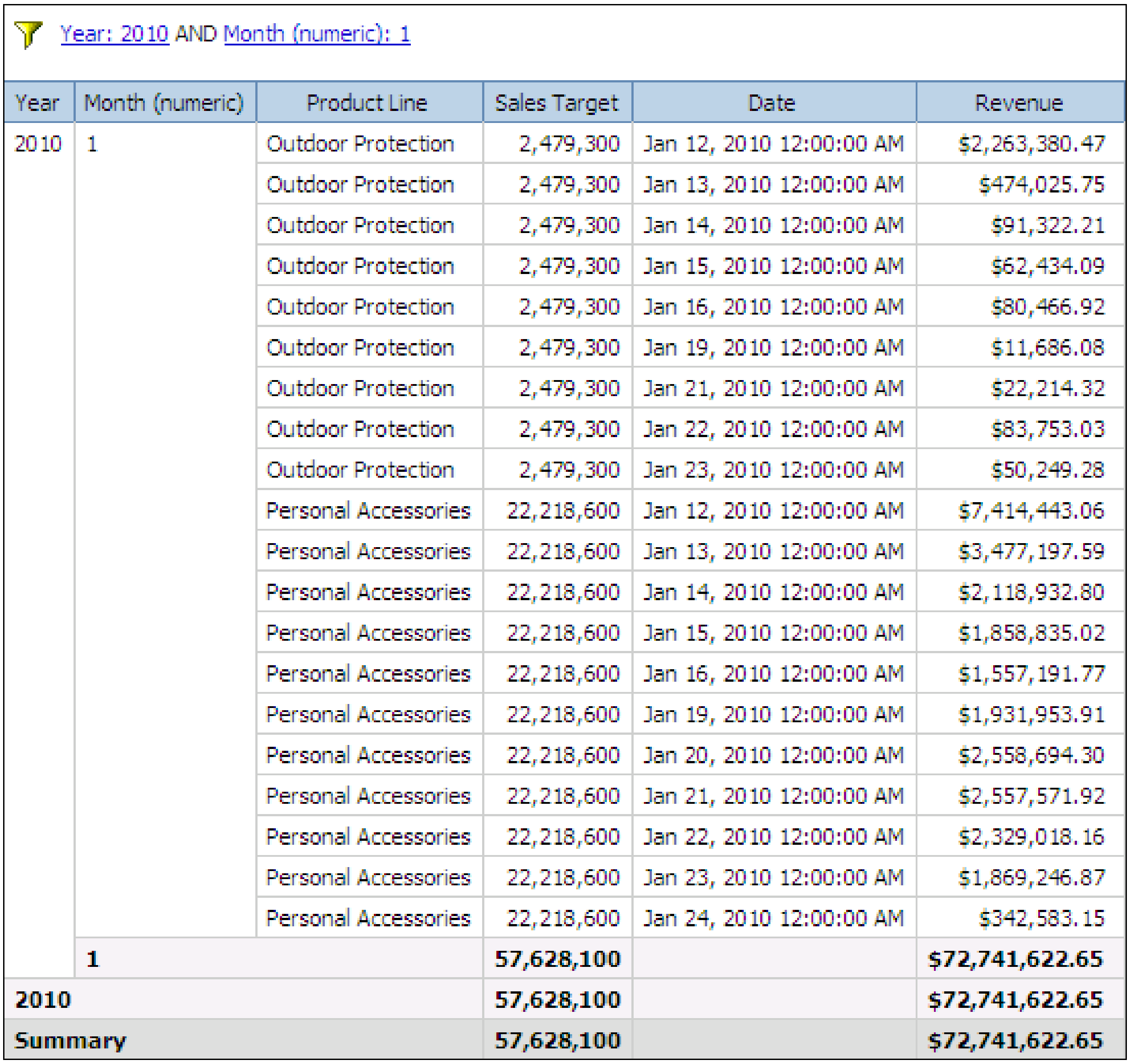
- RETAILER_CODE

Attributes:

- COMPANY_NAME
- COMPANY_NAME_MB
- RETAILER_TYPE_CODE
- RETAILER_START_DATE

Workshop 1: Workshop Results

After scrolling to the bottom of the Products Dimension Test report, the results should appear similar as shown below:

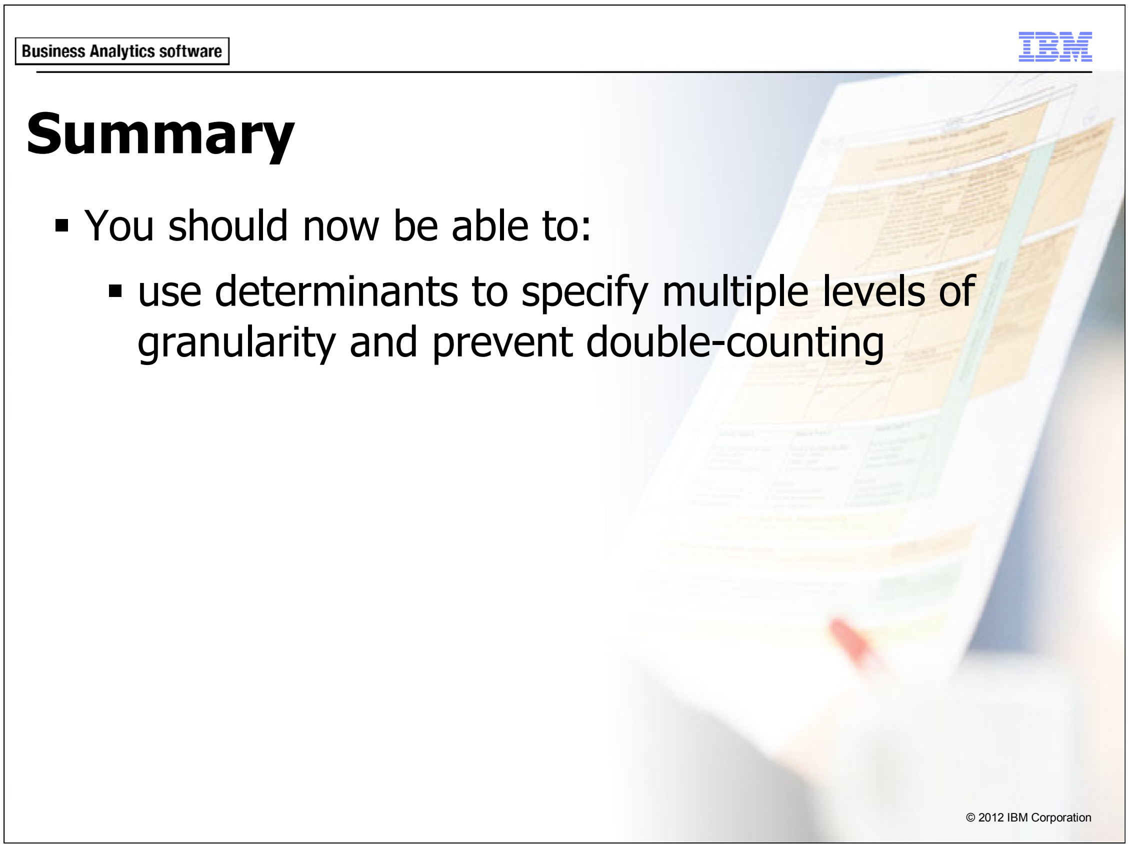


Year: 2010 AND Month (numeric): 1

Year	Month (numeric)	Product Line	Sales Target	Date	Revenue
2010	1	Outdoor Protection	2,479,300	Jan 12, 2010 12:00:00 AM	$2,263,380.47
		Outdoor Protection	2,479,300	Jan 13, 2010 12:00:00 AM	$474,025.75
		Outdoor Protection	2,479,300	Jan 14, 2010 12:00:00 AM	$91,322.21
		Outdoor Protection	2,479,300	Jan 15, 2010 12:00:00 AM	$62,434.09
		Outdoor Protection	2,479,300	Jan 16, 2010 12:00:00 AM	$80,466.92
		Outdoor Protection	2,479,300	Jan 19, 2010 12:00:00 AM	$11,686.08
		Outdoor Protection	2,479,300	Jan 21, 2010 12:00:00 AM	$22,214.32
		Outdoor Protection	2,479,300	Jan 22, 2010 12:00:00 AM	$83,753.03
		Outdoor Protection	2,479,300	Jan 23, 2010 12:00:00 AM	$50,249.28
		Personal Accessories	22,218,600	Jan 12, 2010 12:00:00 AM	$7,414,443.06
		Personal Accessories	22,218,600	Jan 13, 2010 12:00:00 AM	$3,477,197.59
		Personal Accessories	22,218,600	Jan 14, 2010 12:00:00 AM	$2,118,932.80
		Personal Accessories	22,218,600	Jan 15, 2010 12:00:00 AM	$1,858,835.02
		Personal Accessories	22,218,600	Jan 16, 2010 12:00:00 AM	$1,557,191.77
		Personal Accessories	22,218,600	Jan 19, 2010 12:00:00 AM	$1,931,953.91
		Personal Accessories	22,218,600	Jan 20, 2010 12:00:00 AM	$2,558,694.30
		Personal Accessories	22,218,600	Jan 21, 2010 12:00:00 AM	$2,557,571.92
		Personal Accessories	22,218,600	Jan 22, 2010 12:00:00 AM	$2,329,018.16
		Personal Accessories	22,218,600	Jan 23, 2010 12:00:00 AM	$1,869,246.87
		Personal Accessories	22,218,600	Jan 24, 2010 12:00:00 AM	$342,583.15
	1		**57,628,100**		**$72,741,622.65**
2010			**57,628,100**		**$72,741,622.65**
Summary			**57,628,100**		**$72,741,622.65**

Sales Target Values are not double-counted.

If your browser was left open from the last demo, the report may not work as expected due to caching and the determinants not being reflected. Simply close the browsers and then try again.



Business Analytics software

Summary

- You should now be able to:
  - use determinants to specify multiple levels of granularity and prevent double-counting