Paper 183-2008

Exploiting OLAP Cubes with SAS 9: A Hands on Workshop

Gregory S. Nelson
ThotWave Technologies, Chapel Hill, North Carolina

Abstract

In a related hands-on workshop (Nelson, 2008 - Building OLAP Cubes with SAS 9: A Hands on Workshop), we demonstrated how to use OLAP Cube Studio to build and manage an OLAP cube. In this workshop, we will take those cubes and learn how we can view them in a variety of SAS tools which might include: SAS Enterprise Guide, Microsoft Excel with the SAS OLAP Data Provider, and the SAS Information Delivery Portal. Finally, we will discuss best practices related to delivering cubes to your end users and we will learn how you can build web interfaces to exploit your OLAP cubes. In addition, we will discuss how you can secure your OLAP cubes so that you can restrict what data users/ group can see.

Keywords: Data warehouse, DI Studio, OLAP Cube Studio, star schema, MDDB, MDX, SAS Add-in for Microsoft Office, SAS Information Delivery Portal.

Introduction

This hands-on workshop is a continuation of a series of papers on data warehousing that began at SUGI 22 (1997) with a paper entitled "*Implementing a Dimensional Data Warehouse with the SAS® System*". Since then, we have shown how you can plan for, design and build data warehouses using SAS. In this workshop, we look specifically at how to exploit multi-dimensional data stores (or cubes as they are often referred to) using a variety of OLAP viewers in both SAS and non-SAS products.

As stated in the companion paper (Nelson, 2008 – Building OLAP Cubes with SAS 9: A Hands on Workshop), the purpose of an OLAP cube is to store data in such a way that an end user can slice and dice through data. The purpose of this workshop is to learn how to take the cube that we designed and built and actually make it come alive in some of these tools.

If you need a refresher on what an OLAP cube is or what products are needed to support this, please refer to the companion paper.

Scope of this Workshop

Just as we did in the previous papers, in this workshop, we are going to take some retail data from the Northwinds database that was converted to an OLAP cube and exploit that data using a variety of tools.

In this workshop, our goal is to demonstrate how easy it is to surface these OLAP cubes in Excel and on the web. We will start by beginning where we left off in the last workshop – showing the OLAP cube in SAS Enterprise Guide.

1. View the completed OLAP cube using SAS Enterprise Guide 3.0 OLAP Analyzer
2. View the completed OLAP cube using the Microsoft Excel
3. View the completed OLAP cube using the SAS Information Delivery Portal
4. View the completed OLAP cube using the SAS Web OLAP Viewer for Java
5. Discuss how we can implement the right security for our cube

The Workshop Data

Please note for continuity the data used, for the exercises below, is the same data used in the previous hands-on workshop. However, even if you did not attend the workshop you should be able to go back to your organization and use it today.

Data Used in this Workshop

The example we will be using throughout the paper will be an OLAP cube that was built in the previous workshop. This data was transformed into a star schema and then into an OLAP cube from the original source data which is the Northwinds Trading Company database that is distributed with Microsoft Access.

The OLAP cube that we will be working with is fairly straightforward. The ***measures*** or facts are simply the numbers that we have about the orders people placed for food. They include quantities of items purchased, the unit cost, applicable discounts and the dates the order was placed and shipped.

The ***dimensions*** include:

- Time (date, day of week, quarter and year)
- Customers (Name, city, region, country)
- Employees (so we can tell who sold what by region and country)
- Products (including which products and product categories)

If we were to visualize this as a dimensional model, we would first visualize the three primary subjects: customers, products and employees in a cube.

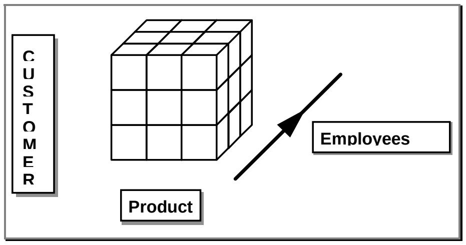


Figure 1. 3-D Model of our Main Subjects

Our three dimensional example can be extended to four dimensions by adding a time dimension to indicate the month of the year in which a sale was made. Visualizing a fourth dimension is more difficult than visualizing three. Imagine twelve boxes depicting the months of the year into which our cube is placed. When the box is placed in the JANUARY box, the cells contain information for JANUARY. When in the FEBRUARY box, the cells contain information for the month of FEBRUARY, and so on.

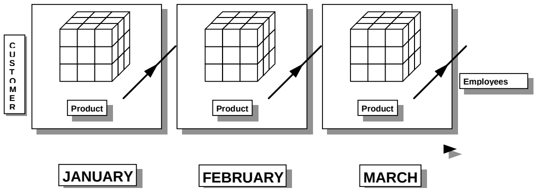


Figure 2. 4-D Model with the time dimension

This paradigm can be extended to five or more dimensions.

Let's take this opportunity to review our example. In our data, the metric is sales revenue (how much product was purchased by our customers.) We have four dimensions as depicted in Figure 4. At the intersection of two or more dimensions, we have a fact. For example, let's take the example of a two way intersection: *product B sold to customer 2*.

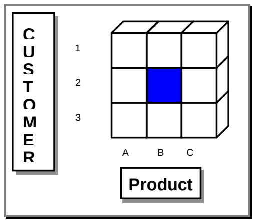


Figure 3. 3-D Model of our Main Subjects

If we calculate a summary statistic, such as sum of sales revenue, we could now answer: *how much money did customer 2 spend on product B?*

Obviously, we could come up with more complex questions for our warehouse which involves three or more dimensions. This is where the multi-dimensional database plays a significant role. The ability for users to slice-and-dice the data is done at a dimension level. *Ranging* is when users want to restrict their view to attributes on a dimension. In this scenario, a user might want just a few values from each dimension (e.g., I want to know how much revenue was generated by two of my employees from my two largest customers in the beverage and produce categories.) The figure below shows this situation.

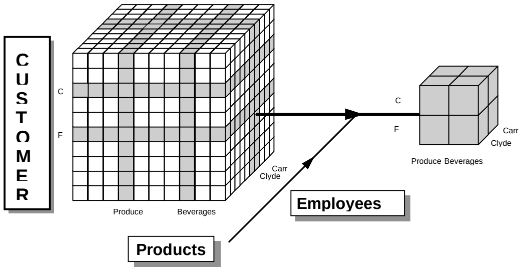


Figure 4. 3-D Model of our Main Subjects

In sum, the logical model helps us think about how our data is organized and whether or not we have the granularity in our data to answer the question(s).

Exercises

Task #1: Viewing the OLAP Cube in SAS Enterprise Guide

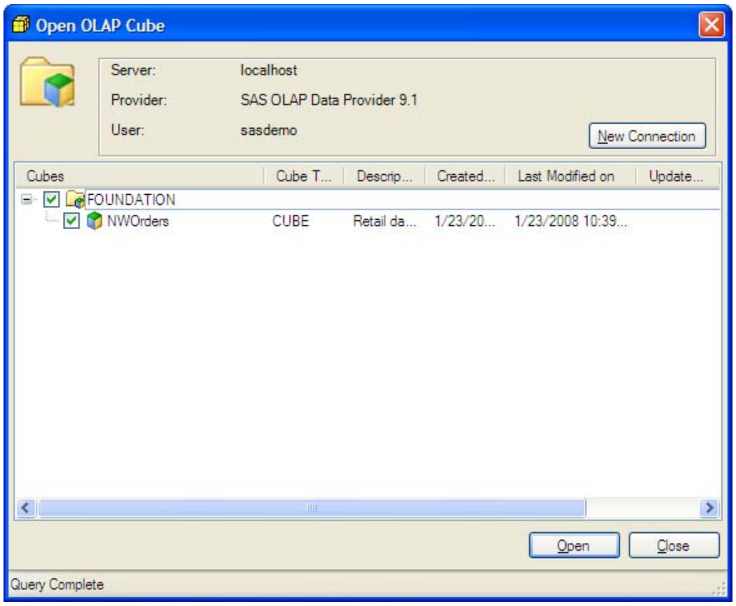
1. To open a SAS cube in Enterprise Guide, start Enterprise Guide and login (if required).

2. Select **File → Open → OLAP Cube**. The **OLAP Cube Login** window opens.

3. In the **OLAP Server Name** box, specify the name of the OLAP server that contains the cube that you want to open. (for this workshop, we will use the name of the machine or **localhost**.)

4. From the **Provider** drop-down list, select the provider. We will use : SAS OLAP Provider 9.1

5. Provide the userid and password (SASDEMO and the password SASPASS. After selecting OK, you should see the following screen.



Open OLAP Cube

Server: localhost

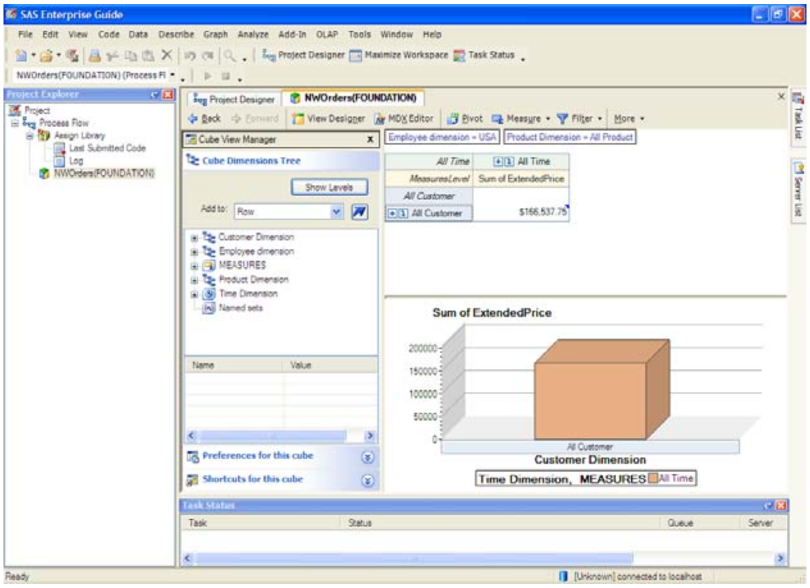
Provider: SAS OLAP Data Provider 9.1

User: sasdemo

Cubes	Cube T...	Descrip...	Created...	Last Modified on	Update...
☑ FOUNDATION					
☑ NWOrders	CUBE	Retail da...	1/23/20...	1/23/2008 10:39...	

Query Complete

6. In the **Open OLAP Cube** window, select the check box next to the cube that you want to open and click **Open**. The cube opens in the **OLAP Analyzer** window.

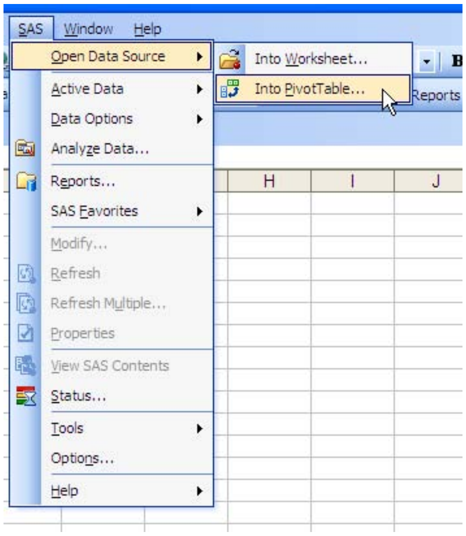


All Time	All Time
MeasuresLevel	Sum of ExtendedPrice
All Customer	
All Customer	$166,537.75

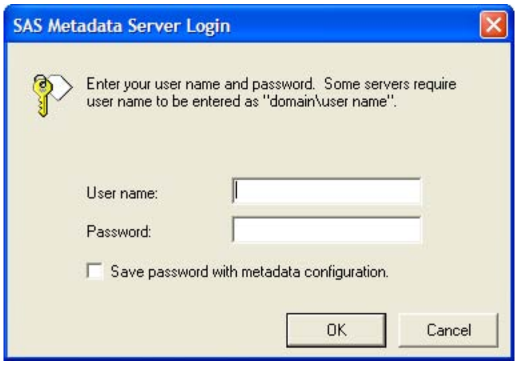
Task #2: Viewing the OLAP Cube in Microsoft Excel

1. To open a SAS cube in Microsoft Excel, start Excel.

2. Select **SAS → Open Data Source → Into PivotTable**.



3. The **Login** window opens.



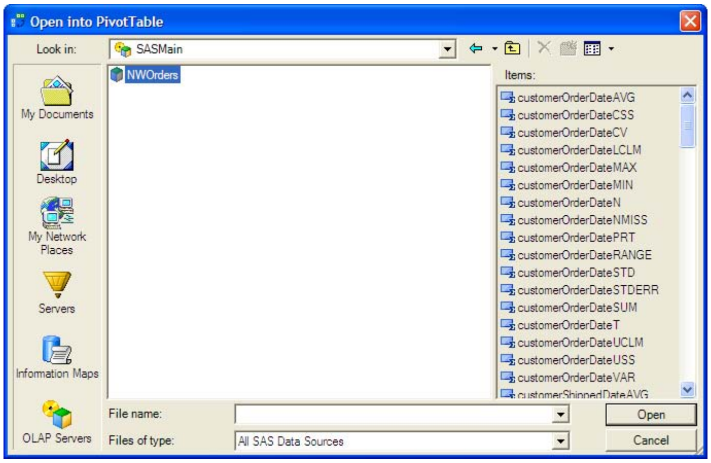
Enter your user name and password. Some servers require user name to be entered as "domain\user name".

User name:

Password:

□ Save password with metadata configuration.

4. In the **OLAP into Pivot Table** box, select **OLAP Servers** on the left (bottom), select **SASMain** and then **NWOrders**. Select **Open**.



Open into PivotTable

Look in: SASMain

NWOrders

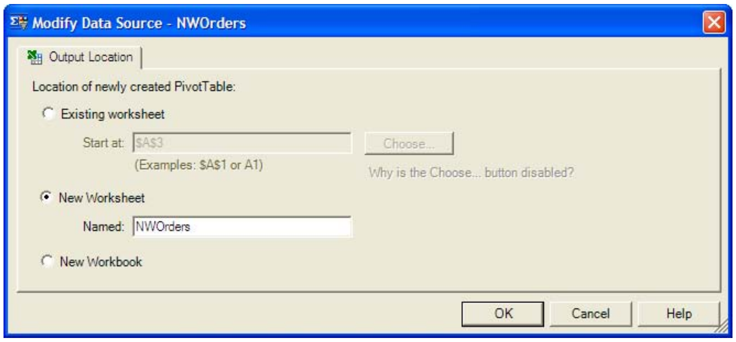
Items:

customerOrderDateAVG
customerOrderDateCSS
customerOrderDateCV
customerOrderDateLCLM
customerOrderDateMAX
customerOrderDateMIN
customerOrderDateN
customerOrderDateNMISS
customerOrderDatePRT
customerOrderDateRANGE
customerOrderDateSTD
customerOrderDateSTDERR
customerOrderDateSUM
customerOrderDateT
customerOrderDateUCLM
customerOrderDateUSS
customerOrderDateVAR
customerShippedDateAVG

File name:

Files of type: All SAS Data Sources

5. In the **Modify Data Source** box, select a new worksheet and click **Open**.



Modify Data Source - NWOrders

Output Location

Location of newly created PivotTable:

○ Existing worksheet

Start at: A3

(Examples: A1 or A1)

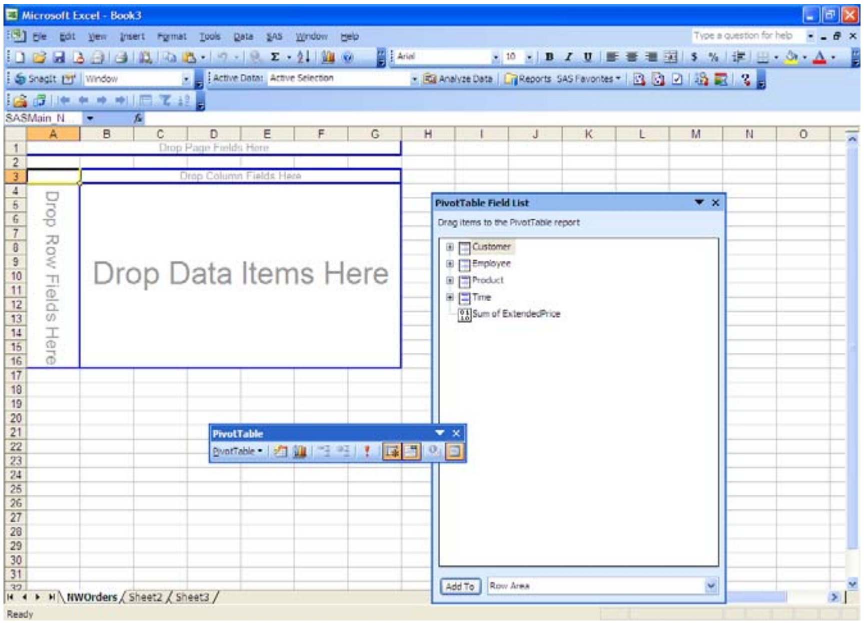
Why is the Choose... button disabled?

● New Worksheet

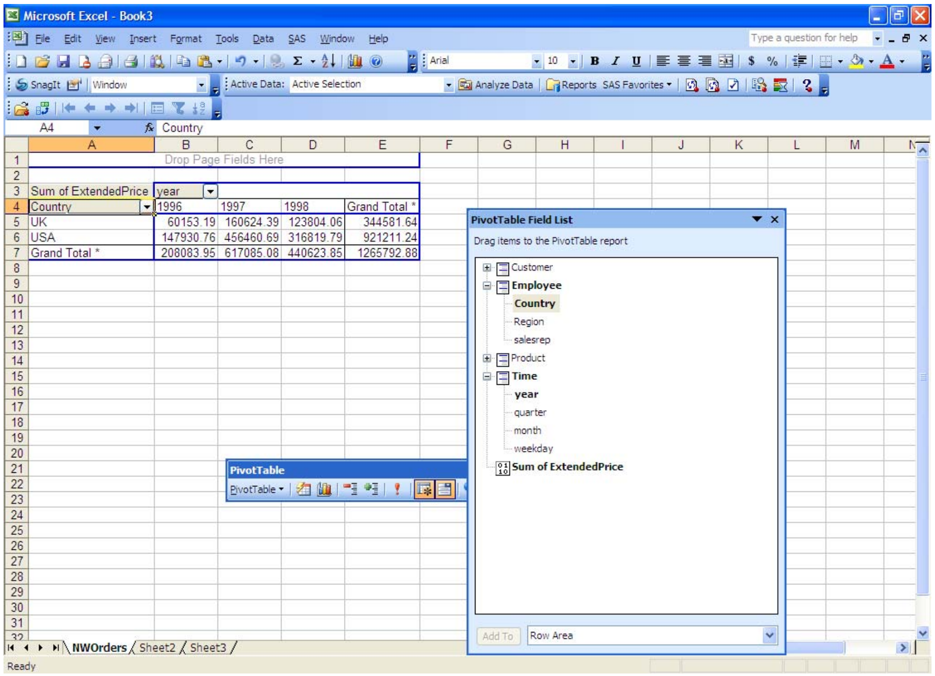
Named: NWOrders

○ New Workbook

6. You are now taken back to a new worksheet in Excel in the Pivot table wizard.



7. For this exercise we are simply going to report on sales by territory. For the measures (inside the table), we are going to select **Sum of Extended Price**. For the columns, we'll select Year and the rows, we'll select Employee Country (**Country**).



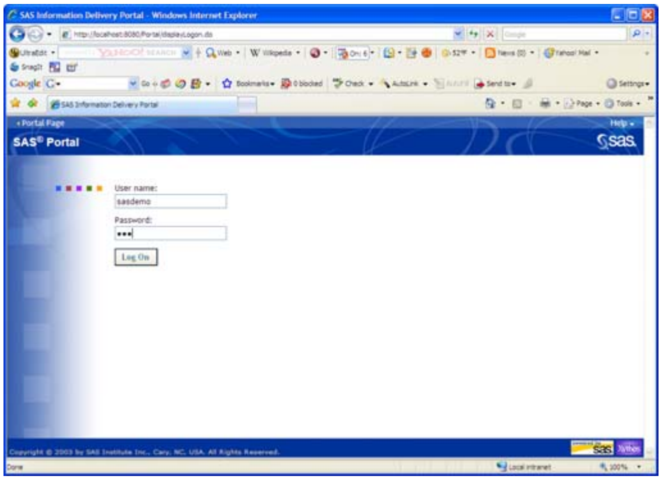
Sum of ExtendedPrice	year			
Country	1996	1997	1998	Grand Total
UK	60153.19	160624.39	123804.06	344581.64
USA	147930.76	456460.69	316819.79	921211.24
Grand Total	208083.95	617085.08	440623.85	1265792.88

8. We have now seen how to create an OLAP report in Microsoft Excel using the OLAP cube generated from SAS.

Task #3: Viewing the OLAP Cube in SAS Information Delivery Portal

1. To open a SAS cube in SAS Information Delivery Portal, start the portal application by pointing your browser to: **http://localhost:8080/Portal**

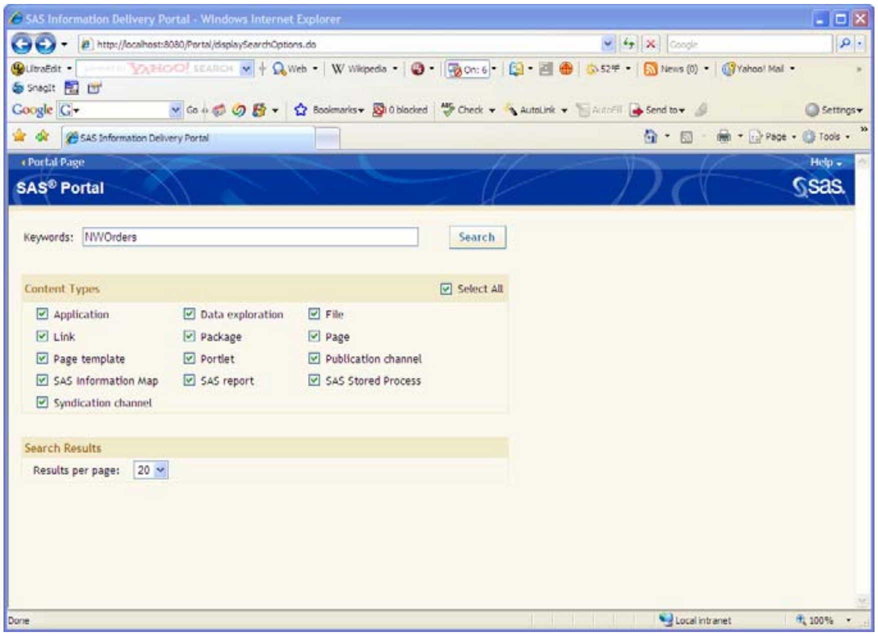
2. Log on using the SASDEMO userid and SASPASS password.


SAS Information Delivery Portal - Windows Internet Explorer
Portal Page
SAS® Portal
Help
sas
Log On
Copyright © 2003 by SAS Institute Inc., Cary, NC, USA. All Rights Reserved.
Done
Local intranet
100%

User name: sasdemo

Password: •••

3. Select **Search**. Type in NWOrders in the search box and select all items to search.



Keywords: NWOrders Search

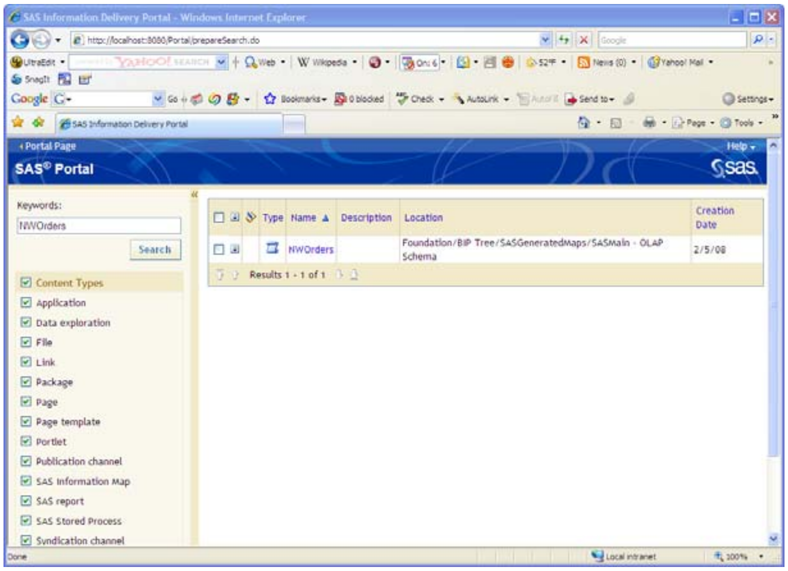
Content Types ☑ Select All

☑ Application ☑ Data exploration ☑ File
☑ Link ☑ Package ☑ Page
☑ Page template ☑ Portlet ☑ Publication channel
☑ SAS Information Map ☑ SAS report ☑ SAS Stored Process
☑ Syndication channel

Search Results

Results per page: 20

4. The **Search Results** window opens.



Keywords: NWOrders

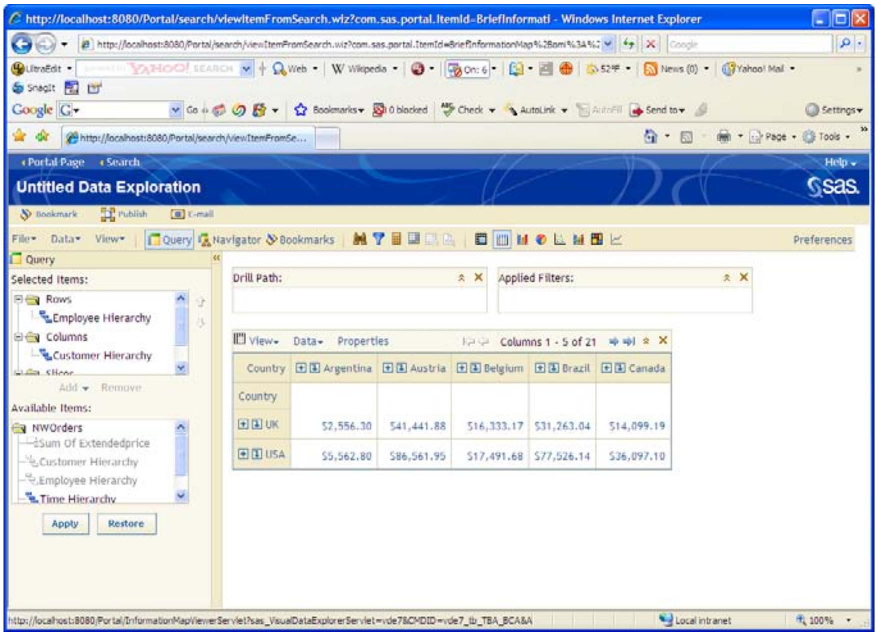
Search

☑ Content Types
☑ Application
☑ Data exploration
☑ File
☑ Link
☑ Package
☑ Page
☑ Page template
☑ Portlet
☑ Publication channel
☑ SAS Information Map
☑ SAS report
☑ SAS Stored Process
☑ Syndication channel

Type	Name ▲	Description	Location	Creation Date
	NWOrders		Foundation/BIP Tree/SASGeneratedMaps/SASMain - OLAP Schema	2/5/08

Results 1 - 1 of 1

5. Next, select the NW Orders OLAP cube found on the left. The OLAP cube will be brought into the main portal window.



http://localhost:8080/Portal/search/viewItemFromSearch.wiz?com.sas.portal.ItemId=BriefInformati - Windows Internet Explorer

Untitled Data Exploration

Selected Items: Rows: Employee Hierarchy; Columns: Customer Hierarchy

Available Items: NWOrders: Sum Of Extendedprice, Customer Hierarchy, Employee Hierarchy, Time Hierarchy

Drill Path:

Applied Filters:

Columns 1 - 5 of 21

Country	Argentina	Austria	Belgium	Brazil	Canada
Country					
UK	$2,556.30	$41,441.88	$16,333.17	$31,263.04	$14,099.19
USA	$5,562.80	$86,561.95	$17,491.68	$77,526.14	$36,097.10

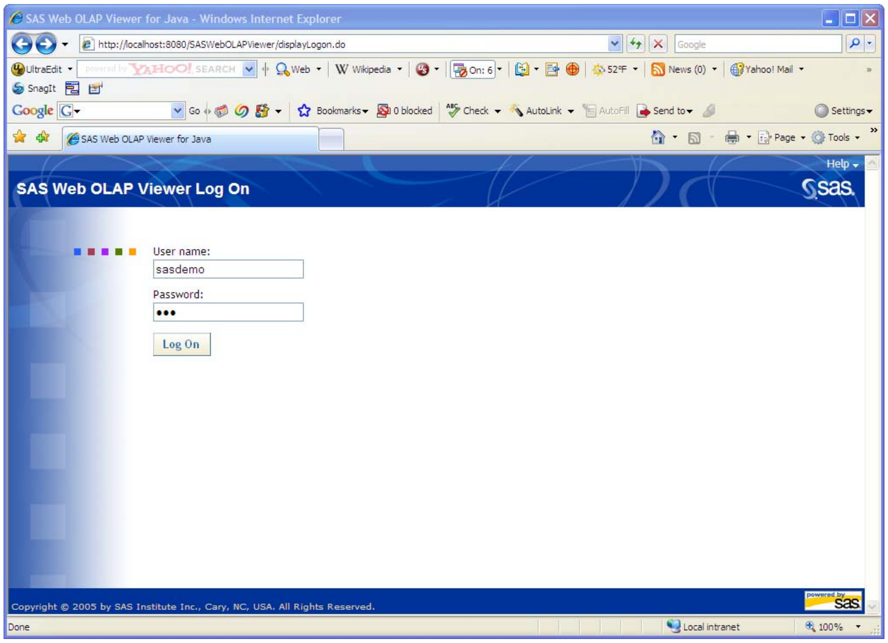
http://localhost:8080/Portal/InformationMapViewerServlet?sas_VisualDataExplorerServlet=vde7&CMDID=vde7_tb_TBA_BCA&A

6. From here, we can change the items that appear in the report by using the navigation on the left.

Task #4: Viewing the OLAP Cube in SAS Web OLAP Viewer for Java

1. To open a SAS cube in SAS Web OLAP Viewer for Java, start the application by pointing your browser to: **http://localhost:8080/SASWebOLAPViewer**

2. Log on using the SASDEMO userid and SASPASS password.



SAS Web OLAP Viewer for Java - Windows Internet Explorer

http://localhost:8080/SASWebOLAPViewer/displayLogon.do

SAS Web OLAP Viewer Log On

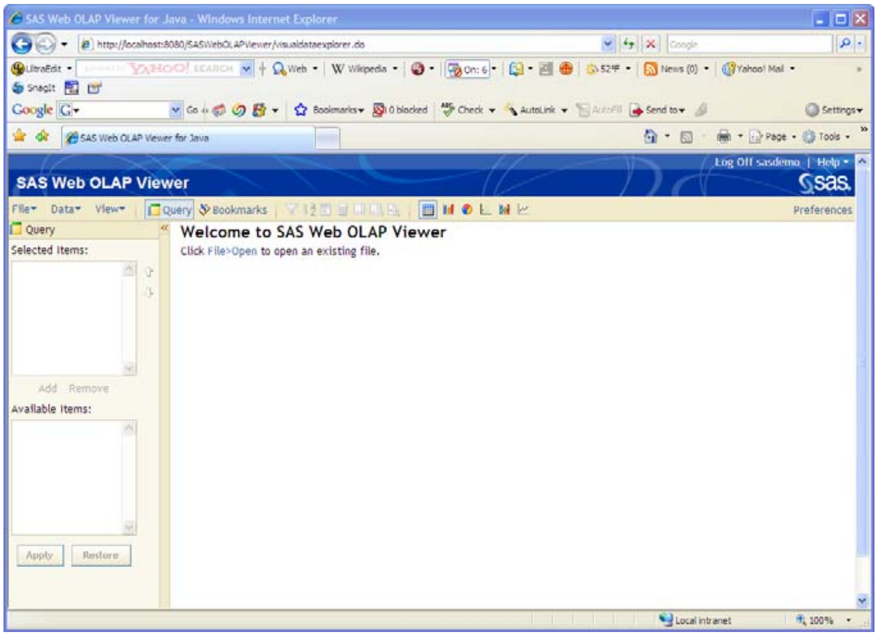
User name: sasdemo

Password: •••

Log On

Copyright © 2005 by SAS Institute Inc., Cary, NC, USA. All Rights Reserved.

3. You are presented with the main screen of the application.

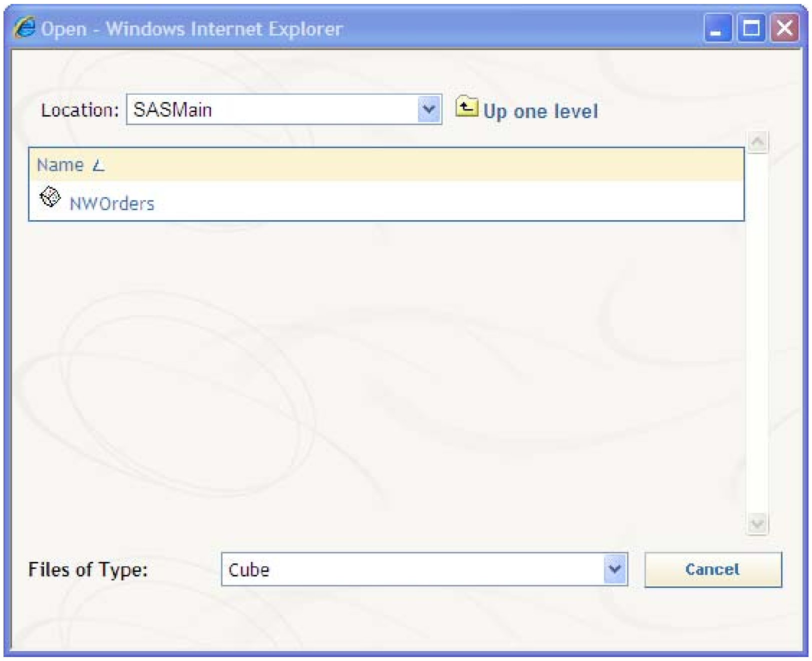

SAS Web OLAP Viewer for Java - Windows Internet Explorer
http://localhost:8080/SASWebOLAPViewer/visualdataexplorer.do
SAS Web OLAP Viewer
Log Off sasdemo | Help
File Data View Query Bookmarks
Preferences
Query
Selected Items:
Add Remove
Available Items:
Apply Restore
Local intranet
100%

Welcome to SAS Web OLAP Viewer

Click File>Open to open an existing file.

4. Select **File → Open** (or the link provided in the middle of the screen.)

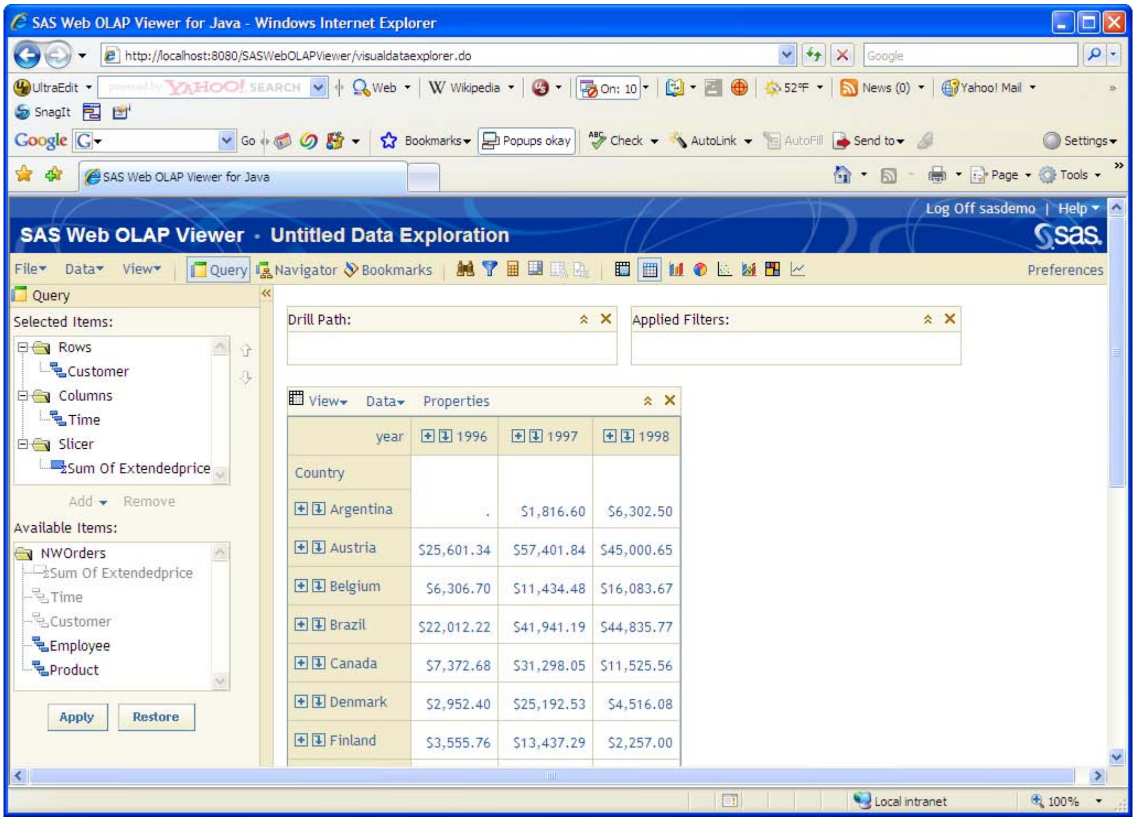
5. The Open dialog appears. Change the **Files of Type** dropdown list to **Cube** and then select **SASMain** in the location field. Select **NWOrders**.



Location: SASMain

Files of Type: Cube

6. The default view is presented.



year	1996	1997	1998
Country			
Argentina	.	$1,816.60	$6,302.50
Austria	$25,601.34	$57,401.84	$45,000.65
Belgium	$6,306.70	$11,434.48	$16,083.67
Brazil	$22,012.22	$41,941.19	$44,835.77
Canada	$7,372.68	$31,298.05	$11,525.56
Denmark	$2,952.40	$25,192.53	$4,516.08
Finland	$3,555.76	$13,437.29	$2,257.00

7. From here, we can change the items that appear in the report by using the navigation on the left, add additional report items such as charts and graphs and filter our data.

Security: Controlling Access to your OLAP Cubes

Now that we have seen how we can view our cubes in a variety of tools, you may (should) be asking yourself, *what if I don't want everyone to see the entire cube?* OLAP cubes can be protected at three levels – the entire cube, on dimensions, hierarchies and level objects and at a member level.

The entire cube can be protected through the *SAS Management Console* by granting or denying access to the **ReadMetadata** and **Read** permissions for the cube.

You can also specify read permissions for an individual dimension, level, or hierarchy for a cube. To set these permissions, you also use the *SAS Management Console* and select **Environment Management** → **Authorization Manager** → **Resource Management** → **By Location**. Select a cube and the individual dimension, level, or hierarchy that you want to modify the permissions for. Select the **Properties** dialog box and the **Authorization tab**. You can grant or deny Read access to the selected dimension, level, or hierarchy for specific users or user groups.

If you have more than one measure, you may also want to set the authorization at the measure level. That is, you may want some users to see Extended Price and others to see say your margin or profit. You do this as you did above, but select your cube and the needed measure or calculated measure that you want to modify the permissions for. Select the **Properties** dialog box and the **Authorization** tab. You can grant or deny **Read** access to the selected measure or calculated measure for specific users or user groups.

The SAS OLAP server monitors the permission settings for measures, calculated measures, dimensions, levels, and hierarchies and enforces those permissions at query time.

Exercise Summary

In the steps above, we took an OLAP Cube built using SAS OLAP Cube Studio and surfaced it using a variety of SAS and non-SAS tools (or clients.) As you might guess, this is just the beginning of what you can really do. We followed the simplest case regarding just bringing those cubes to life and we encourage you to learn more about what wonderful things you can do with this technology. For example, combing geospatial maps and data tables in the same dashboard with an OLAP cube is a fairly common thing. Exploiting these OLAP cubes with MDX queries is also another area that we didn't touch upon here, but there may be applications that take you beyond the basics we've covered.

If you want to expose your cubes in traditional SAS client tools such as web report studio or the information delivery portal, you can also use SAS Information Map Studio to create an information map from the cube and they will be available from those tools as tables (information maps.)

We hope that you continue your education by trying to repeat the process at home and playing around with all of the options.

Biography

Greg Nelson, President and CEO

Greg has recently started his third decade in the SAS eco-system as a programmer, analyst, architect, and teacher. Greg is the President and CEO of ThotWave Technologies where he supports customers in a variety of industries. Prior to ThotWave, Mr. Nelson spent several years in consulting, media and marketing research, database marketing, and large systems support. Mr. Nelson holds a B.A. in Psychology and PhD level work in Social Psychology and Quantitative Methods.

About ThotWave

ThotWave Technologies, LLC is a Cary, NC-based consultancy and a market leader in real-time decision support, specializing in regulated industries, such as life sciences, energy and financial services. ThotWave works at the juncture of business and technology to help companies improve their operational and strategic performance and recognizes the difference between simply accessing data and making data work for business. Through products, partnerships and services, ThotWave enables businesses to leverage data for faster, more intelligent decision making.

Contact information

Your comments and questions are valued and encouraged. Contact the authors at:

Greg Nelson greg@thotwave.com

ThotWave Technologies, LLC

2504 Kildaire Farm Road

Cary, NC 27511

(800) 584 2819

http://www.thotwave.com