



SAP HANA Database – Hands-on Workshop Labs

Document Version 3.1 – April 10, 2013

Date	Author	Notes	Revision
1 January 2012	Tom Ziek	Original Document	1.0
20 September 2012	Lele Palomo		2.0
5 February 2013	Chris Reed	SAP HANA SP5	3.0
10 April 2013	Walter Müllner	A4 Format	3.1



© 2012 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, System i, System i5, System p, System p5, System x, System z, System z10, System z9, z10, z9, iSeries, pSeries, xSeries, zSeries, eServer, z/VM, z/OS, i5/OS, S/390, OS/390, OS/400, AS/400, S/390 Parallel Enterprise Server, PowerVM, Power Architecture, POWER6+, POWER6, POWER5+, POWER5, POWER, OpenPower, PowerPC, BatchPipes, BladeCenter, System Storage, GPFS, HACMP, RETAIN, DB2 Connect, RACF, Redbooks, OS/2, Parallel Sysplex, MVS/ESA, AIX, Intelligent Miner, WebSphere, Netfinity, Tivoli and Informix are trademarks or registered trademarks of IBM Corporation.

Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

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

Oracle and Java are registered trademarks of Oracle and/or its affiliates.

UNIX, X/Open, OSF/1, and Motif are registered trademarks of the Open Group.

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

HTML, XML, XHTML and W3C are trademarks or registered trademarks of W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

SAP, R/3, SAP NetWeaver, Duet, PartnerEdge, ByDesign, SAP BusinessObjects Explorer, StreamWork, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and other countries.

Business Objects and the Business Objects logo, BusinessObjects, Crystal Reports, Crystal Decisions, Web Intelligence, Xcelsius, and other Business Objects products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Business Objects Software Ltd. Business Objects is an SAP company.

Sybase and Adaptive Server, iAnywhere, Sybase 365, SQL Anywhere, and other Sybase products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of Sybase, Inc. Sybase is an SAP company.

All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

The information in this document is proprietary to SAP. No part of this document may be reproduced, copied, or transmitted in any form or for any purpose without the express prior written permission of SAP AG.



SAP HANA Hands-on Workshop

Contents

SAP HANA Hands-on Workshop.....	3
SAP HANA Introduction	4
What is SAP HANA database?	4
About SAP HANA studio	6
Hands-on Labs	6
Connect to HANA Studio	6
Modeling HANA Views	11
Attribute Views.....	11
Hands-on Labs	11
Analytic Views.....	19
Hands-on Labs	20
Calculation Views	33
Hands-on Labs	34
Reporting from SAP HANA with SAP Lumira.....	56
SAP Lumira Overview.....	56
Appendix	71
TPC-H ERD Diagram.....	71
Weblinks	71



SAP HANA Introduction

SAP HANA is a modern platform for real-time analytics and applications. It enables organizations to analyze business operations based on large volume and variety of detailed data in real-time, without any delays.

HANA offers endless opportunity for business innovation within all industries and lines of business. The following three dimensions represent a model for understanding the potential of HANA:

- ✓ **Real-time information and insight into organizational data**
Gain immediate insight from operational or strategic data despite large data volumes, without information delays caused by nightly batch processes.
- ✓ **Real-time simulation & foresight to optimize business processes**
Simulate and foresee business situations like demand patterns or cash exposures based on current business conditions plus historical data and trends.
- ✓ **Real-time sense & response to capitalize on streaming data & events**
Process real-time streaming data like energy consumption with smart meters for driving immediate business decisions in an automated fashion.

What is SAP HANA database?

The SAP HANA database is a relational database that has been optimized to leverage state of the art hardware. It provides all of the SQL features of a standard relational database along with a feature rich set of analytical capabilities and an SAP specific programming language for stored procedures called SQL Script. With these facilities the SAP HANA database is capable of embedding application logic within the database itself. This allows complex queries to be executed directly inside the database, thus reducing the requirement of data transfer to and from the database. This enables SAP HANA based applications to process vast amounts of data whilst operating in a responsive, real-time manner.

From the administrators perspective the SAP HANA Database is conceptually about leveraging modern hardware system landscapes to increase database performance and reliability. Traditionally databases have been designed to operate in a situation where there are limited memory and CPU resources. Currently however, servers can provide in excess of 1TB of memory and up to 80 CPU cores on a single system.

To better understand how the SAP HANA database improves on the traditional database concepts, in the next section we will compare and contrast the two approaches to database systems.

Impact of Modern Hardware on Database System Architecture

Historically database systems were designed to perform well on computer systems with limited RAM, this had the effect that slow disk I/O was the main bottleneck in data throughput.

Consequently the architecture of these systems was designed with a focus on optimizing disk access, e. g. by minimizing the number of disk blocks (or pages) to be read into main memory when processing a query.

Computer architecture has changed in recent years. Now multi-core CPUs (multiple CPUs on one chip or in one package) are standard, with fast communication between processor cores enabling parallel processing. Main memory is no longer a limited resource, modern servers can have 1 TB of system memory and this allows complete databases to be held in RAM. Currently server processors have up to 80 cores, and 128 cores will soon be available. With the increasing number of cores, CPUs are able to process increased data per time interval. This shifts the performance bottleneck from disk I/O to the data transfer between CPU cache and main memory (see figure. 1).

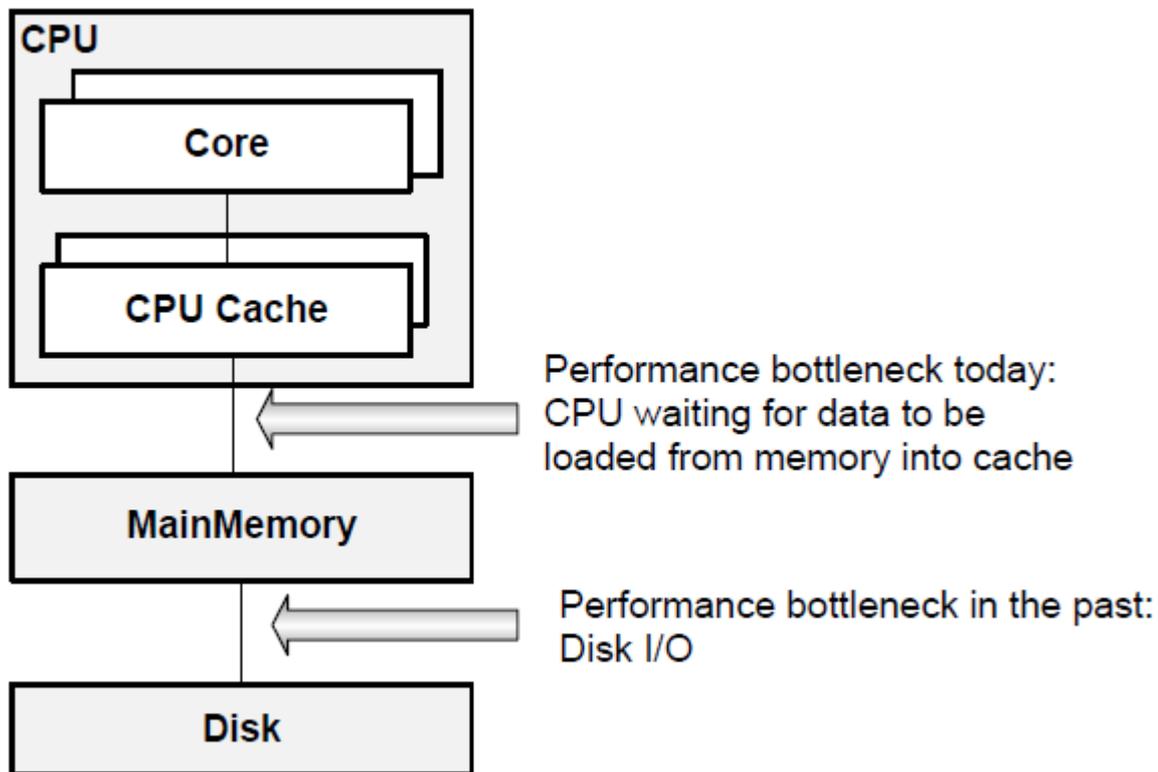


Figure 1: Hardware architecture: Current and past performance bottlenecks

Modern Database for Modern Hardware

From the discussion above it is apparent that traditional databases might not use current hardware most efficiently. So, what are the ideal characteristics of a database systems running on modern hardware?

In-memory database. All relevant data should be available in main memory, which characteristically avoids the performance penalty of disk I/O. To use the advantages of in-memory computation a cache-conscious implementation of data structures and algorithms is necessary.

Support for parallel execution. Higher CPU execution speeds are currently achieved by adding more cores to a CPU package. Multiple CPUs call for new parallel algorithms to be used in databases in order to fully utilize the computing resources available. SAP HANA addresses these requirements, by storing all of its data in RAM and also enabling queries to be split and optimized across multiple CPU cores and multiple SAP HANA servers.

Disk Storage. Disk storage is still required to ensure the ability to restart in case of power failure and for permanent persistency. This is not a performance issue however, as the required disk write operations can happen asynchronously as a background task. More over SAP HANA database tries to get the performance optimum from the different storage types, like main memory, solid state disks (SSD) and traditional mechanical hard drives (HDD).

About SAP HANA studio

The SAP HANA studio provides an administration console, an information modeler and a lifecycle management perspective. The SAP HANA studio runs on the Eclipse platform (For more information, see also <http://www.eclipse.org>).

The administration console of the SAP HANA studio allows technical users to manage the SAP HANA database as well as to create and manage user authorizations.

The information modeler allows technical users to create new or modify existing models of data.

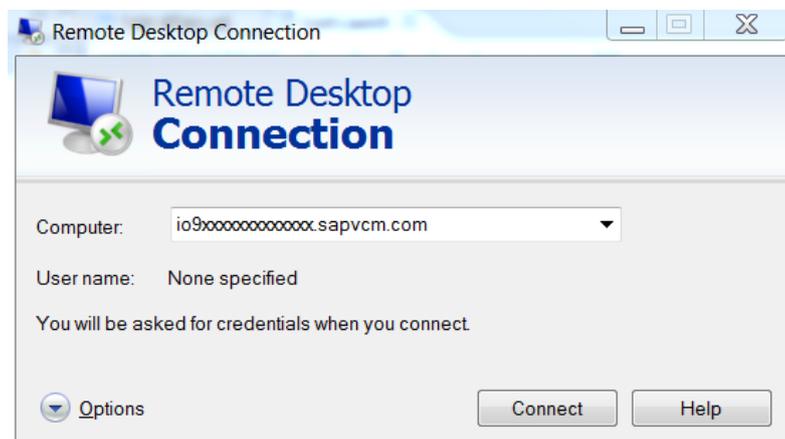
Hands-on Labs

Connect to HANA Studio

Your Instructor will provide connection and environment information. Our labs are being run from the Amazon Elastic Compute Cloud (Amazon EC2). It is comprised of two images:

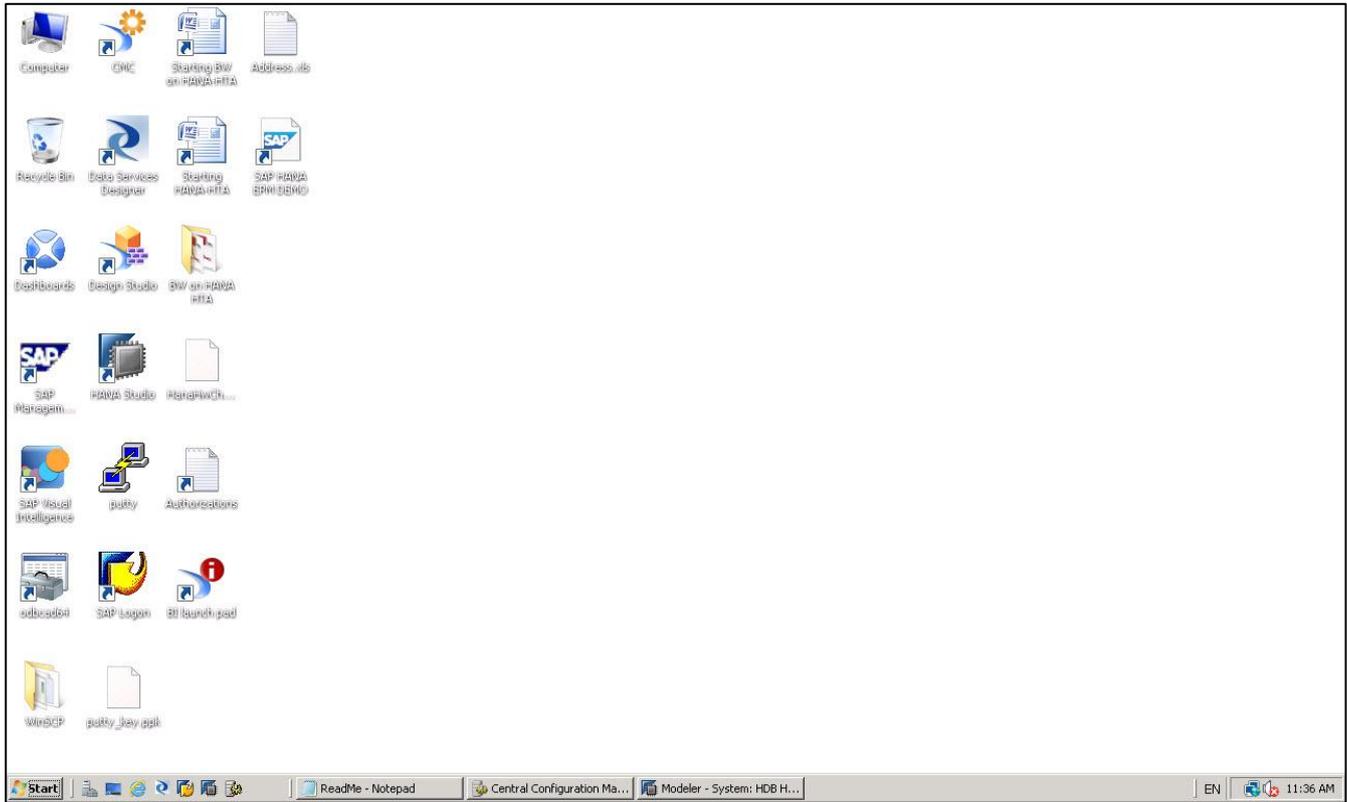
- A Windows-based image that contains the client tools used during the workshop labs. (We will only be connecting to this image directly today)
- A Linux-based image that houses the SAP HANA database environment.

Open the Remote Desktop Connection program (In Windows 7 got to Start => All Programs => Accessories => Remote Desktop Connection), enter the computer name provided by the instructor (it will be in the format as shown below) and then click Connect.

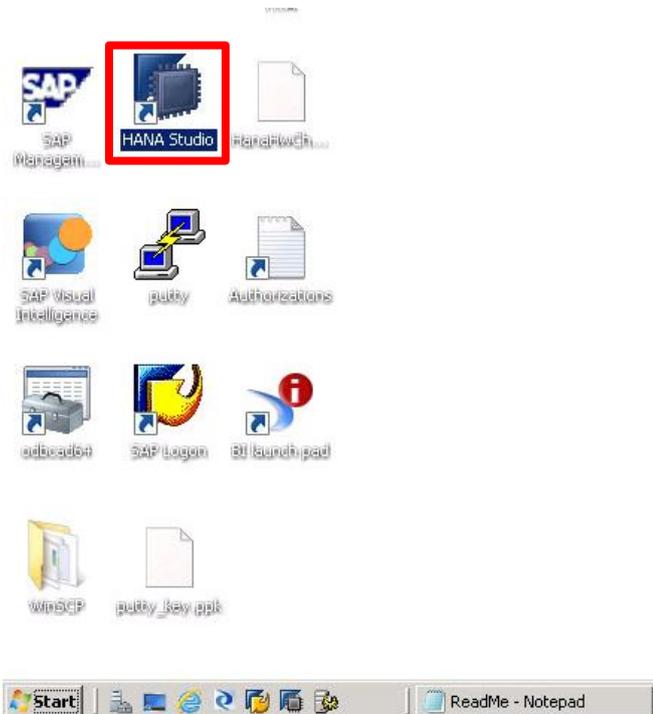


When prompted, log in as **Administrator** with a password of **Welcome1**

You will then be on the Windows client image as shown below.

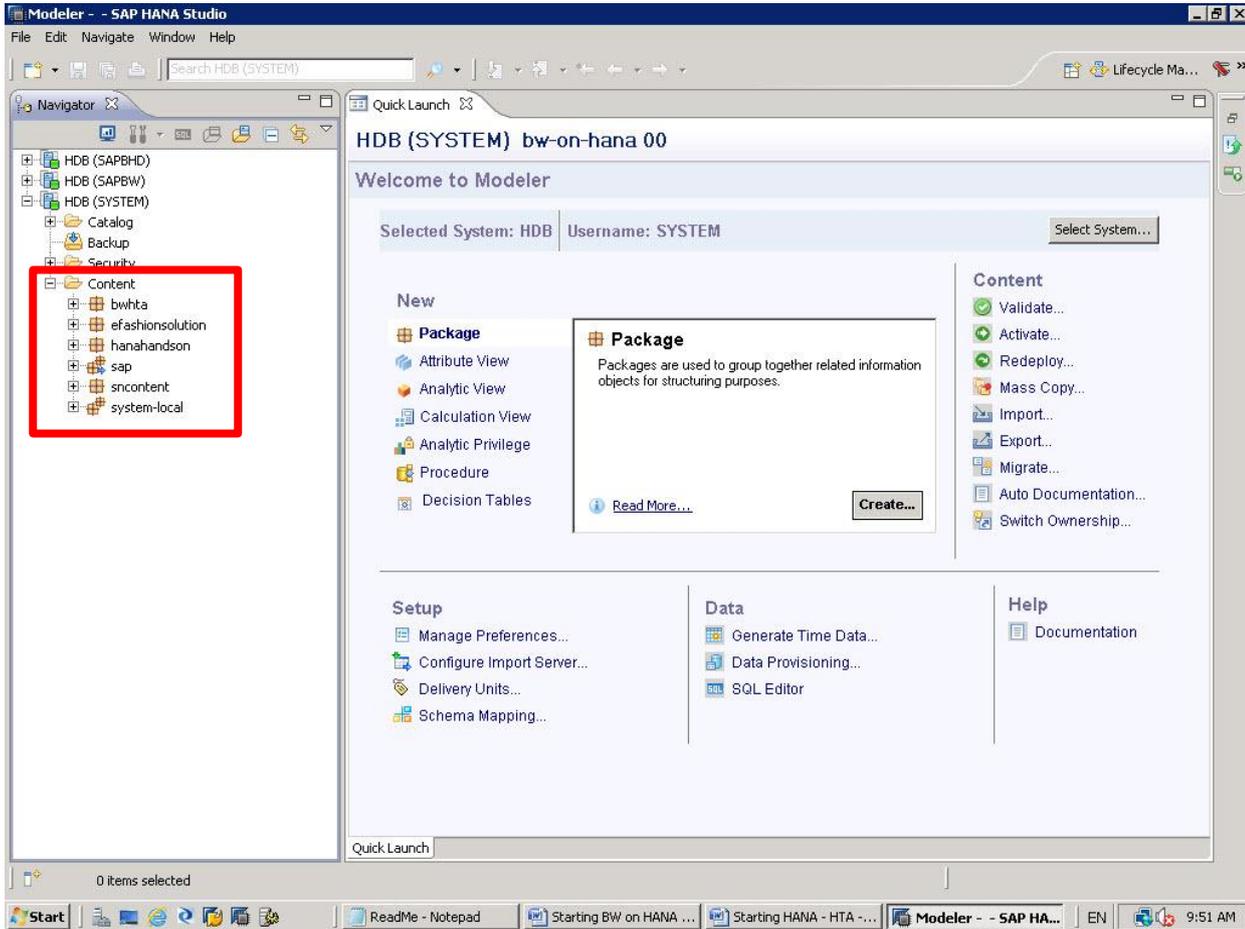


Start SAP HANA studio by double clicking on the hdbstudio.exe shortcut on the Windows desktop



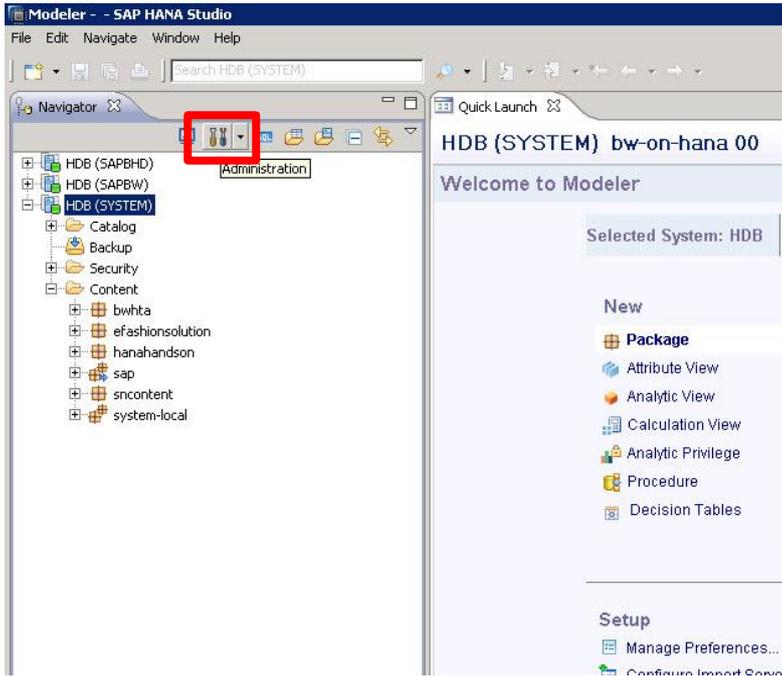
Once SAP HANA studio starts it will look like below with the Navigator on the left panel and the Quick Launch panel on the right. The Navigator shows the systems available at the highest level of the hierarchy. Our environment only shows one SAP HANA system instance setup called HDB. If you

expand HDB you will see the Catalog, which includes such items as authorization information, schemas and schema objects and data. Also under HDB will be Content, which includes the information models (such as Attribute Views and Analytic Views) that we will use to access the HANA database.

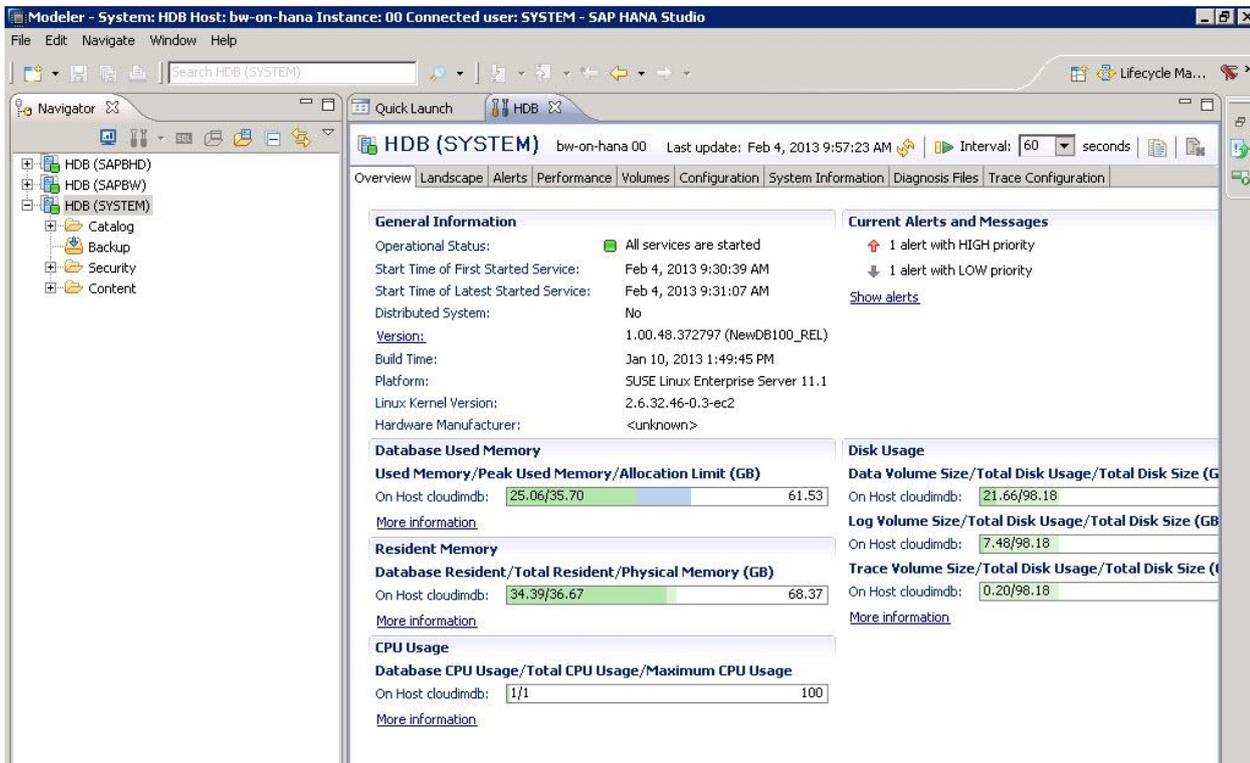


The Quick Launch panel is a good place to find the various objects you can create and actions you can perform on your SAP HANA instance. Before you start creating objects let's take a look at some of the administrative features that SAP HANA studio provides.

Highlight the HDB instance, then find and click the wrench icon on the toolbar right above the instance name.

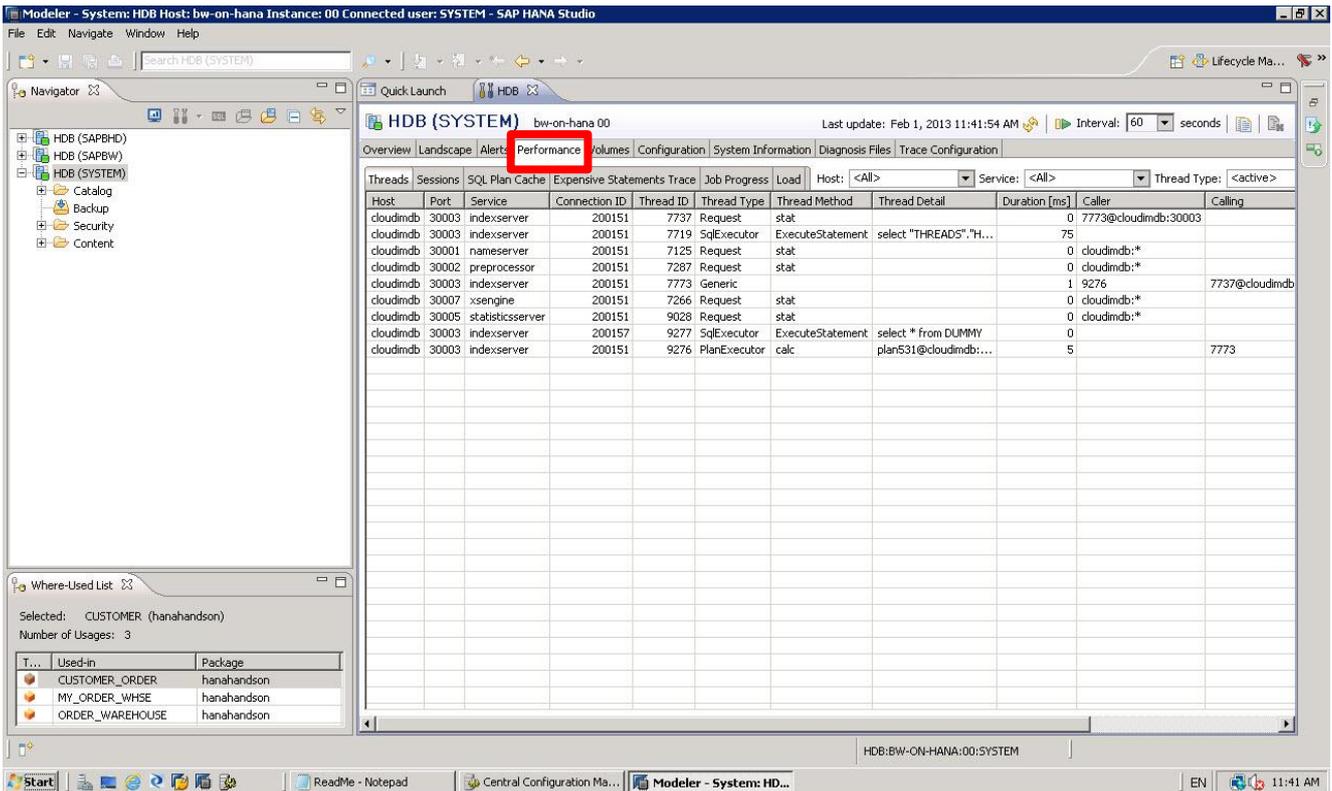


The panel to the right will show an overview of the SAP HANA instance.



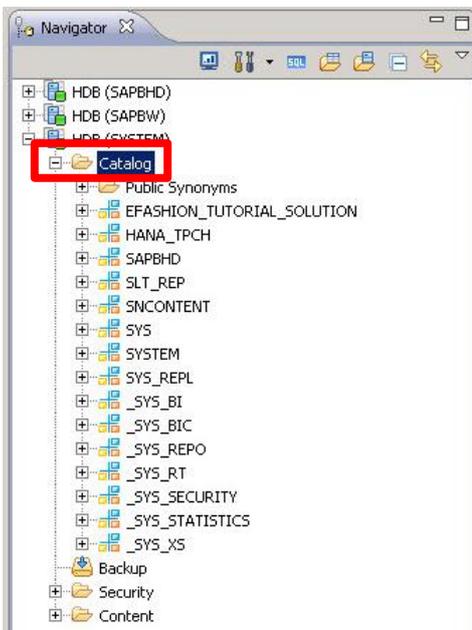
Click the Performance tab.

The Performance tab shows current load on such metrics as CPU, Memory, Disk usage, Network etc. The Configuration tab is where you can change the various settings of the environment without having to log into the Linux server to modify OS (Operating System) level configuration files. If you like check the other tabs on this panel before proceeding.

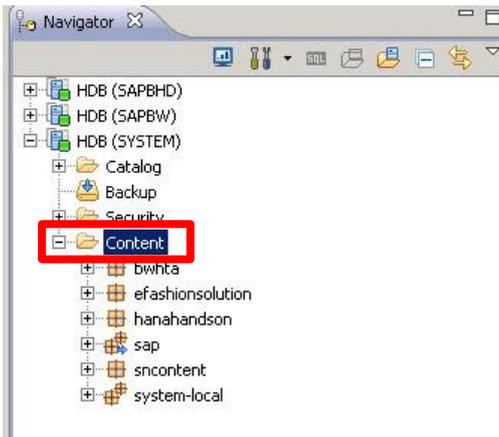


Now let's go back to the Navigator panel on the left and expand the HDB instance node on the hierarchy.

As we learned earlier Catalog contains the Authorization/Security information and the Schemas/Data.



Content is where the Informational Models/Views reside.



Modeling HANA Views

Attribute Views

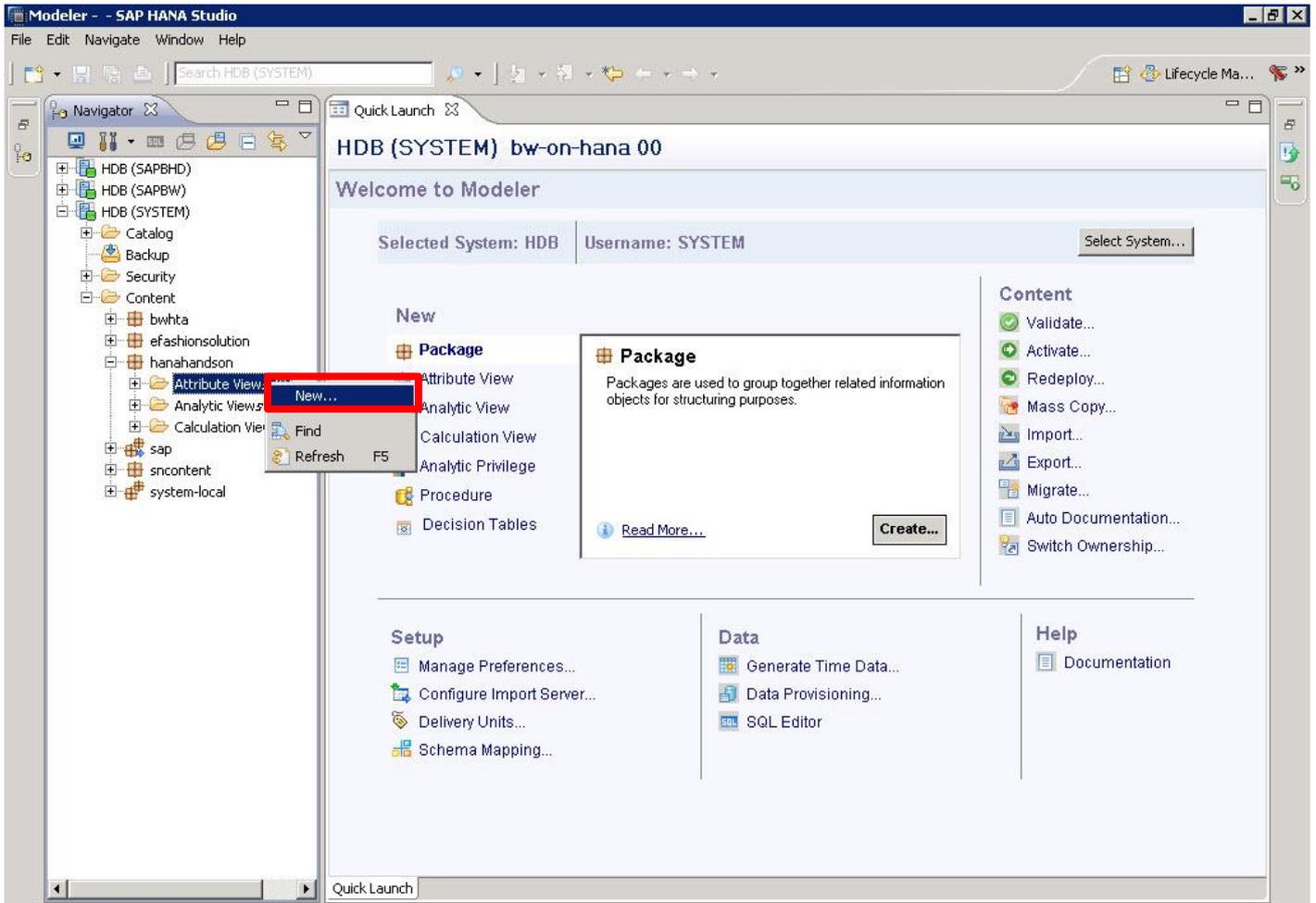
Attribute views are used to give master data tables context. This context is provided by text tables which give meaning to the master data. For example, if our fact table or analytic view only contains some numeric ID for each car dealer then we can link in information about each dealer using an attribute view. We could then display the dealers' names and addresses instead of their IDs thus providing the context for the master data table.

Attribute views are used to select a subset of columns and rows from a data table. As it is of little use to sum up attributes from master data tables there is no need to define measures or aggregates for attribute views.

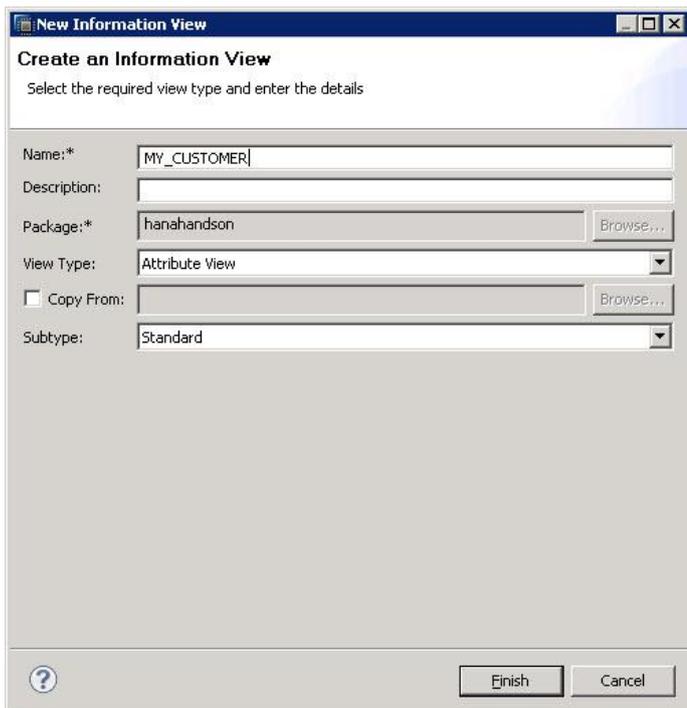
You can also use attribute views to join master data tables to each other, e. g. joining "Plant" to "Material".

Hands-on Labs

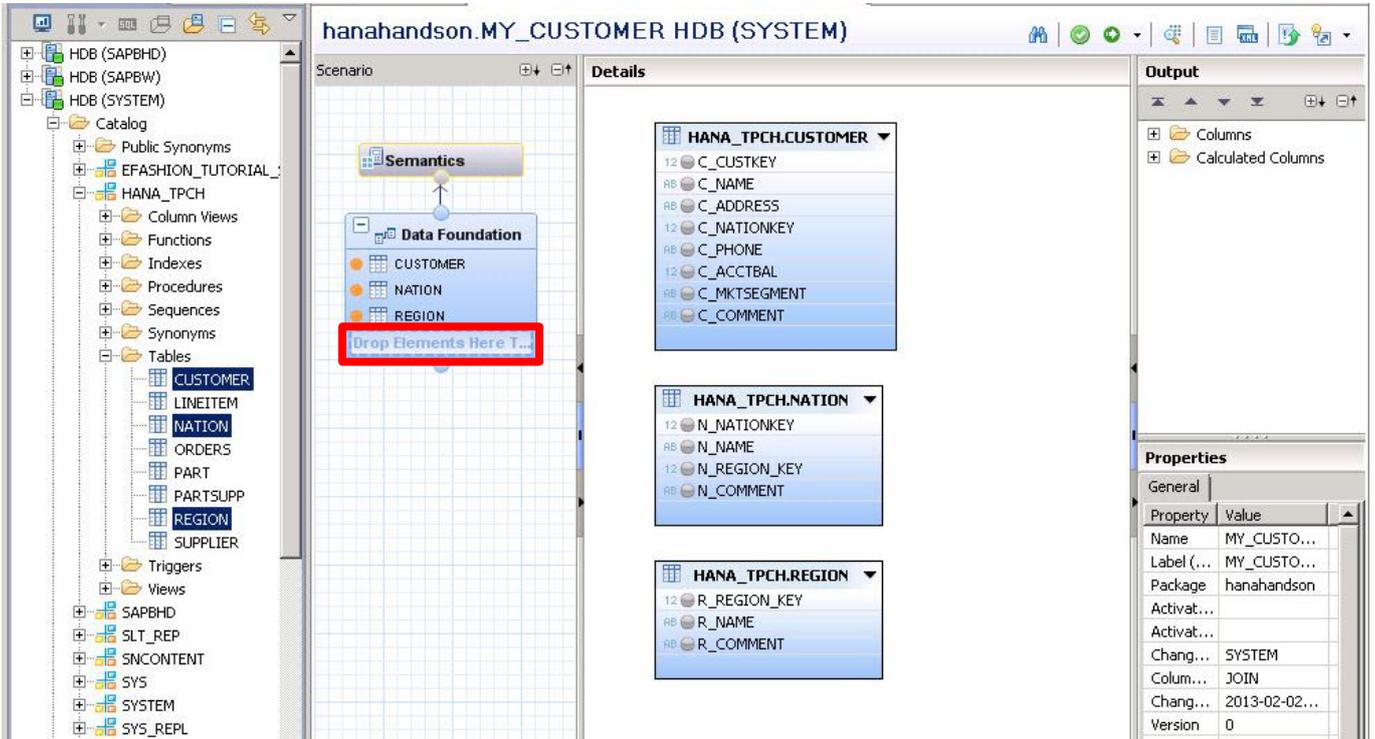
For Information Models we need to expand the IM2 instance => Content. Under Content are collections of views, privileges and procedures called Packages. Expand the hanahandson Package => Attribute Views. Right click on Attribute Views and choose New => Attribute View



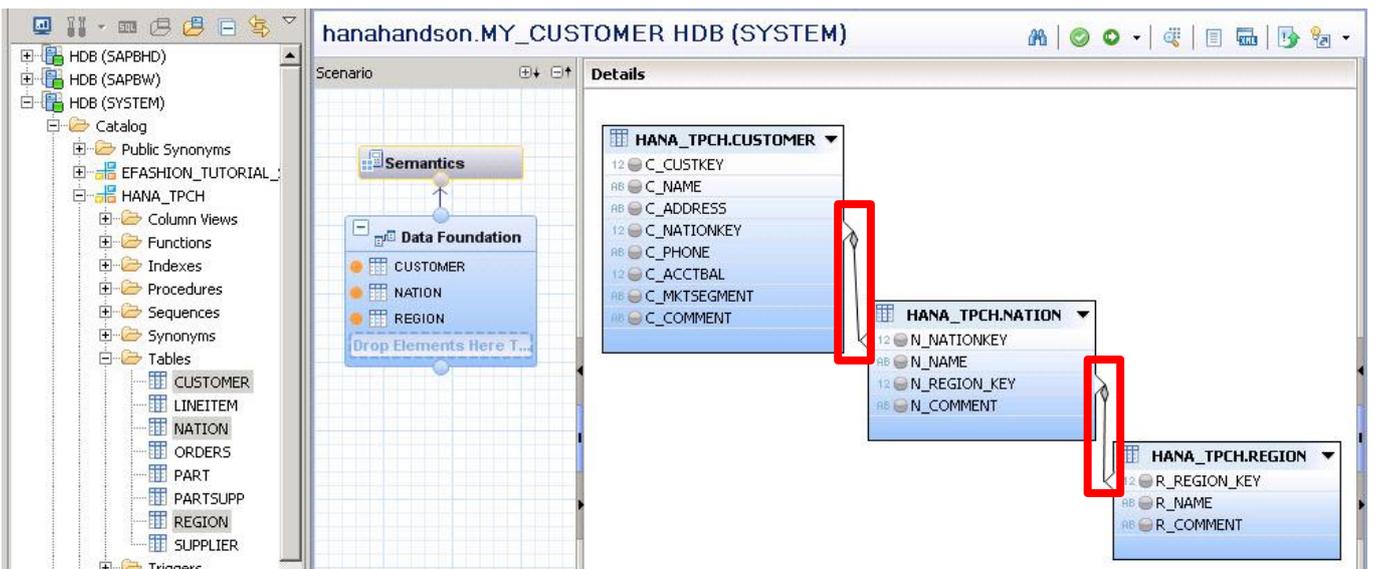
Name your Attribute View MY_CUSTOMER and click Finish (description is optional).



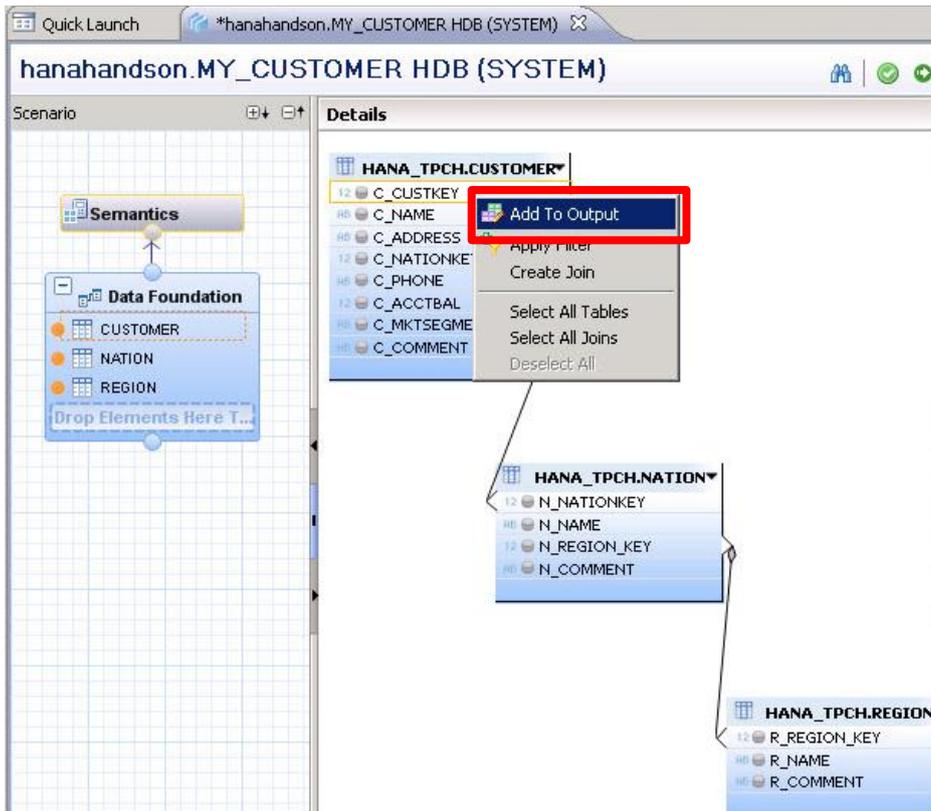
Choose three tables from the HANA_TPCH schema: Customer, Nation, Region (you can hold CTRL and select them together) and drag these tables into the Data Foundation area on the canvas. Your screen should now look like this:



Now join the tables by clicking on the key field and dragging to the corresponding keys. So in this example from *Region.R_Region_Key* => *Nation.N_Region_Key*, *Nation.N_Nationkey* => *Customer.C_Nationkey*.

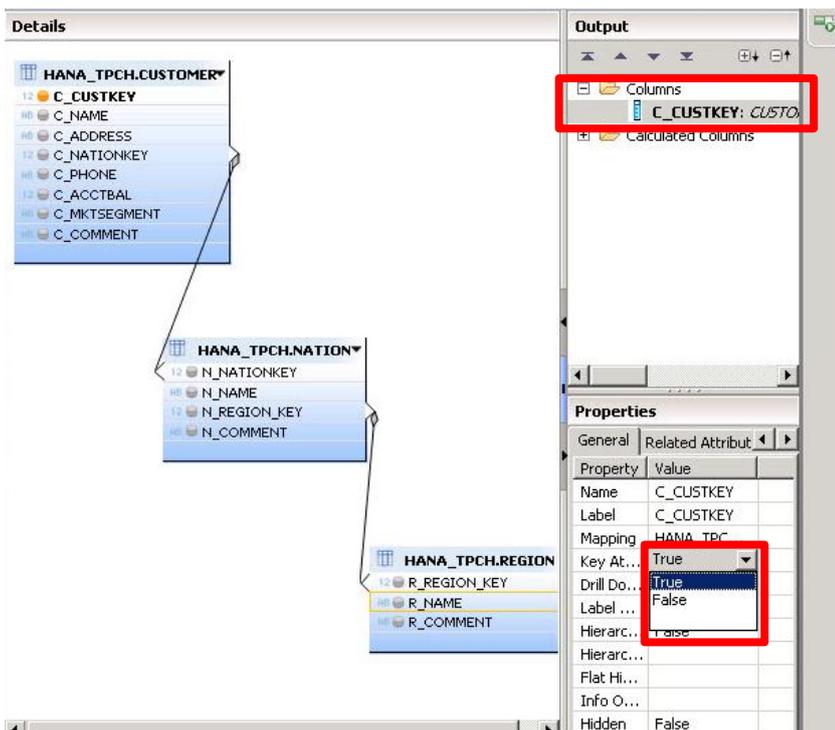


We will now specify the Key Attributes of our View. Select Customer.C_Custkey and right click 'Add To Output'.

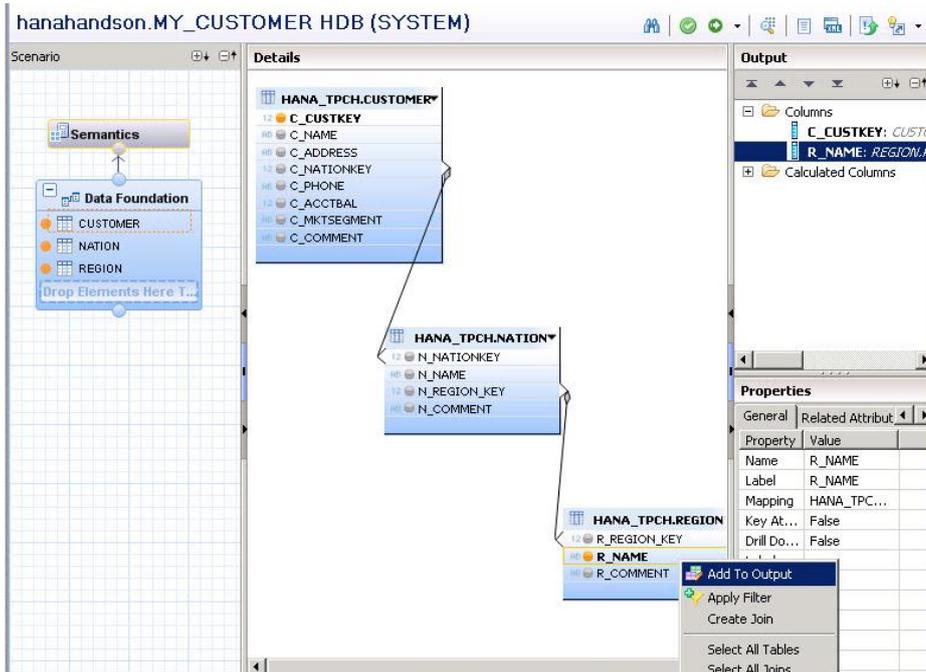


The attribute should now appear in the output on the right hand pane under Columns

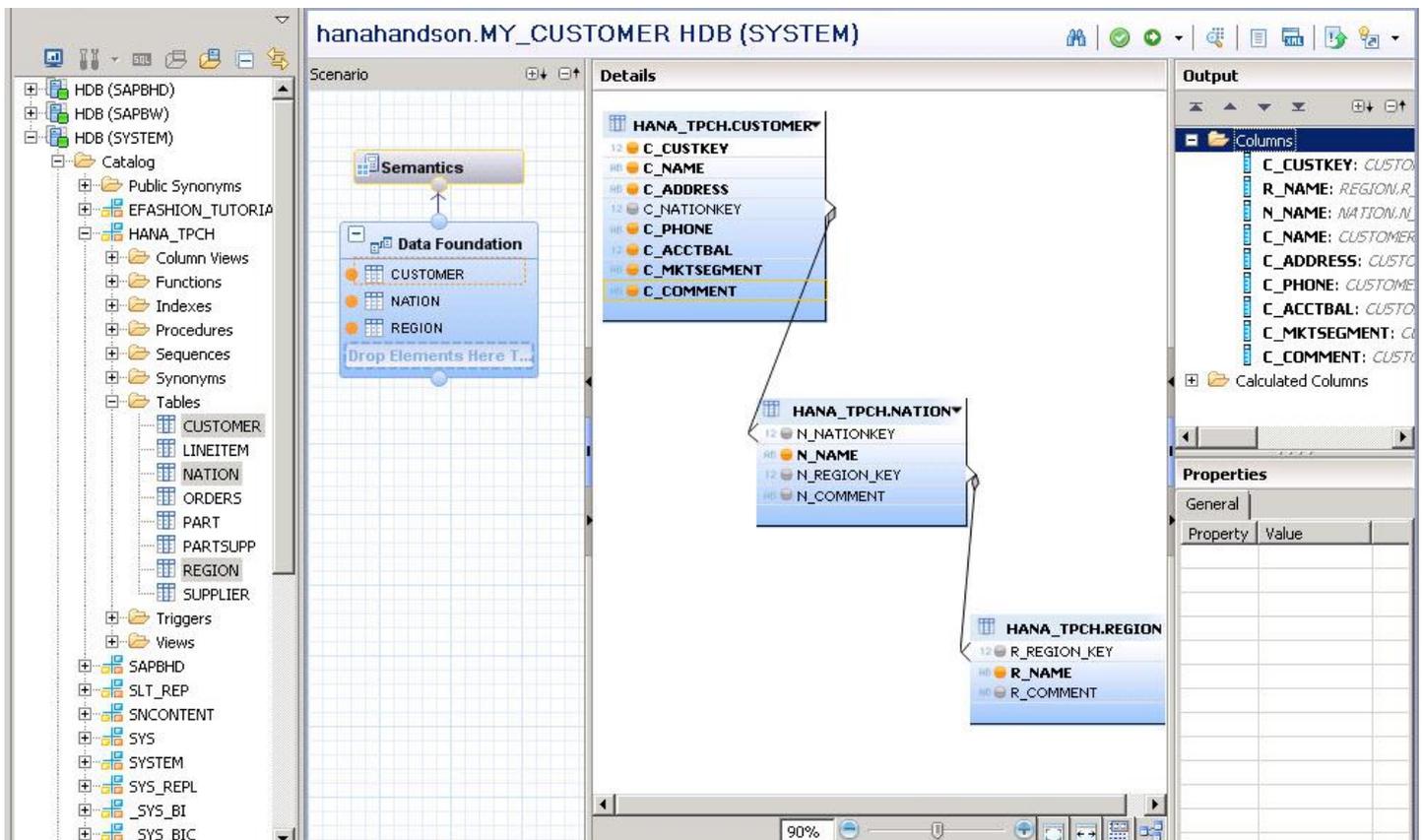
The attribute is visible in the Output panel and we now want to make this attribute a key attribute by first selecting it in the columns folder and then in the properties tab selecting the drop down menu and selecting True:



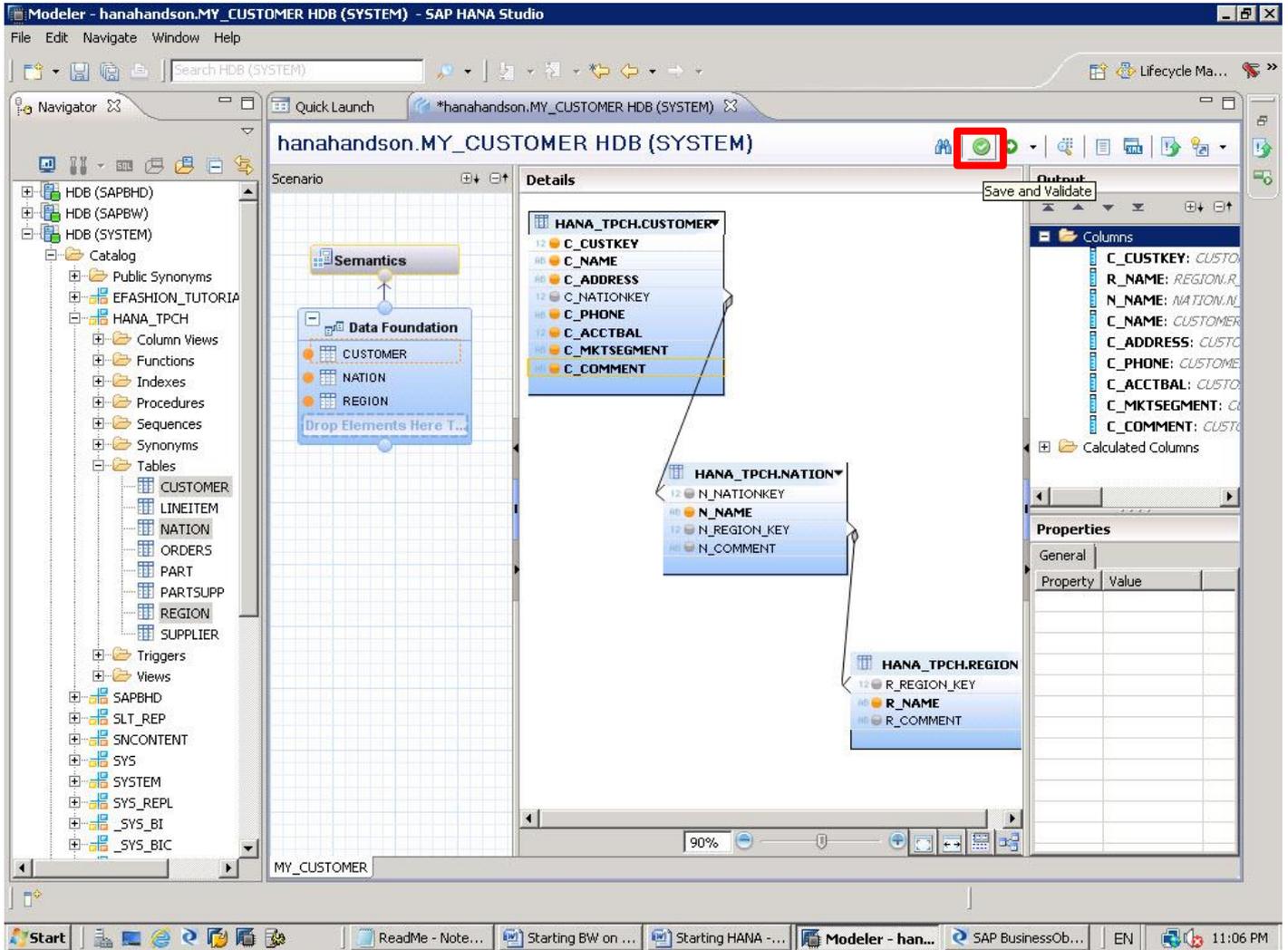
Now add another attribute. Right click on Region.R_Name and choose *Add as Attribute*, it should now look like the following:



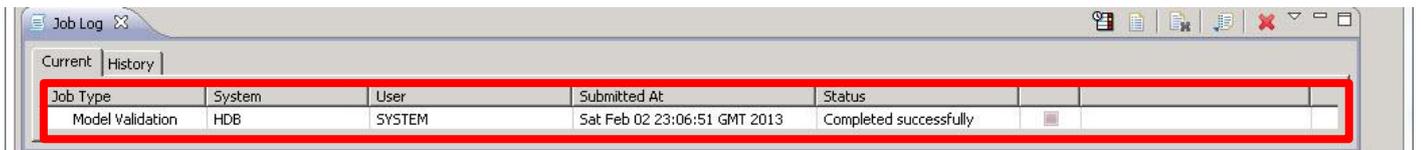
Add the remaining attributes in the same manner as the above: Nation.N_Name; Customer.C_Name; Customer.C_Address; Customer.C_Phone; Customer.C_Acctbal; Customer.C_Mktsegment; Customer.C_Comment. You should now have the below:



Let us validate the Attribute View - In the top right (above the modeling pane) select the green tick which will *Save and Validate*.

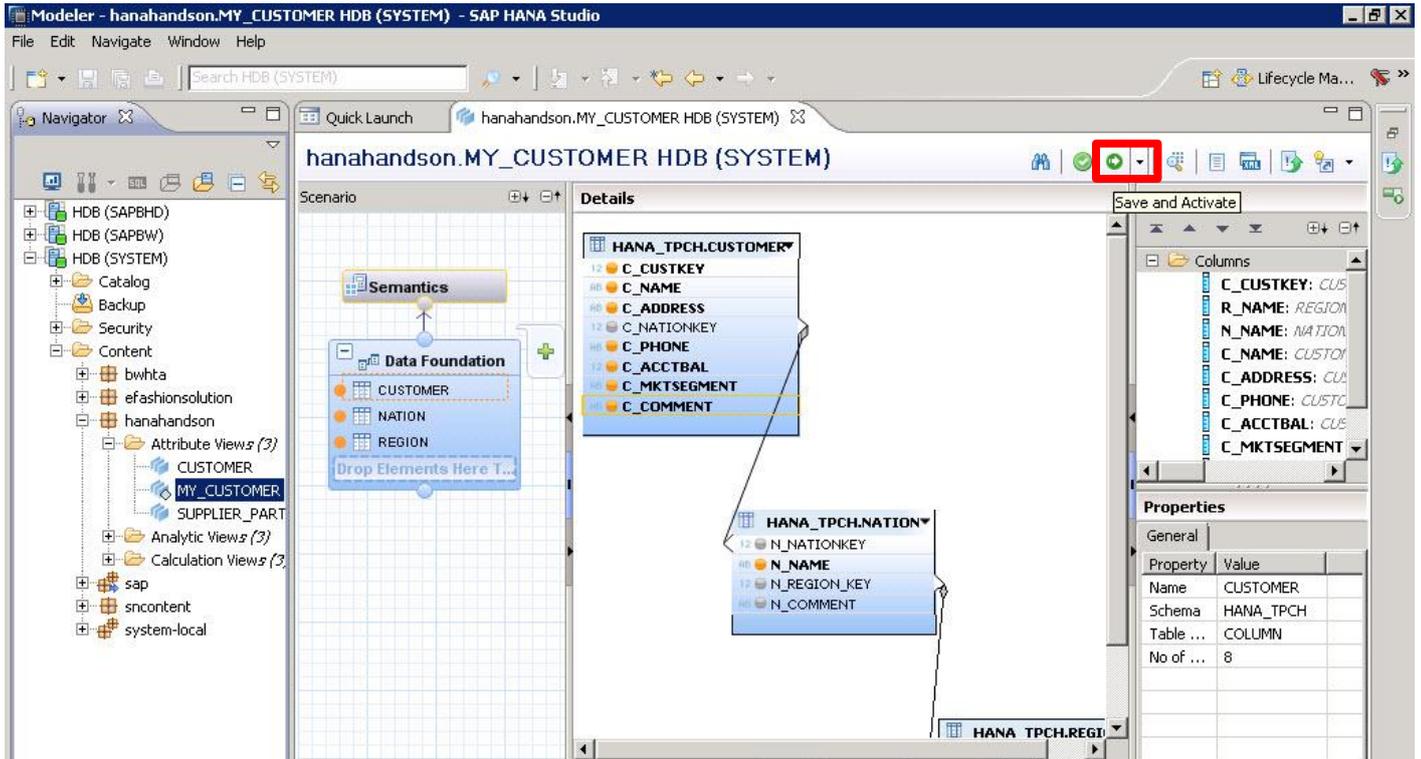


When the validation completes you will see the results in the Validation Log window at the bottom.

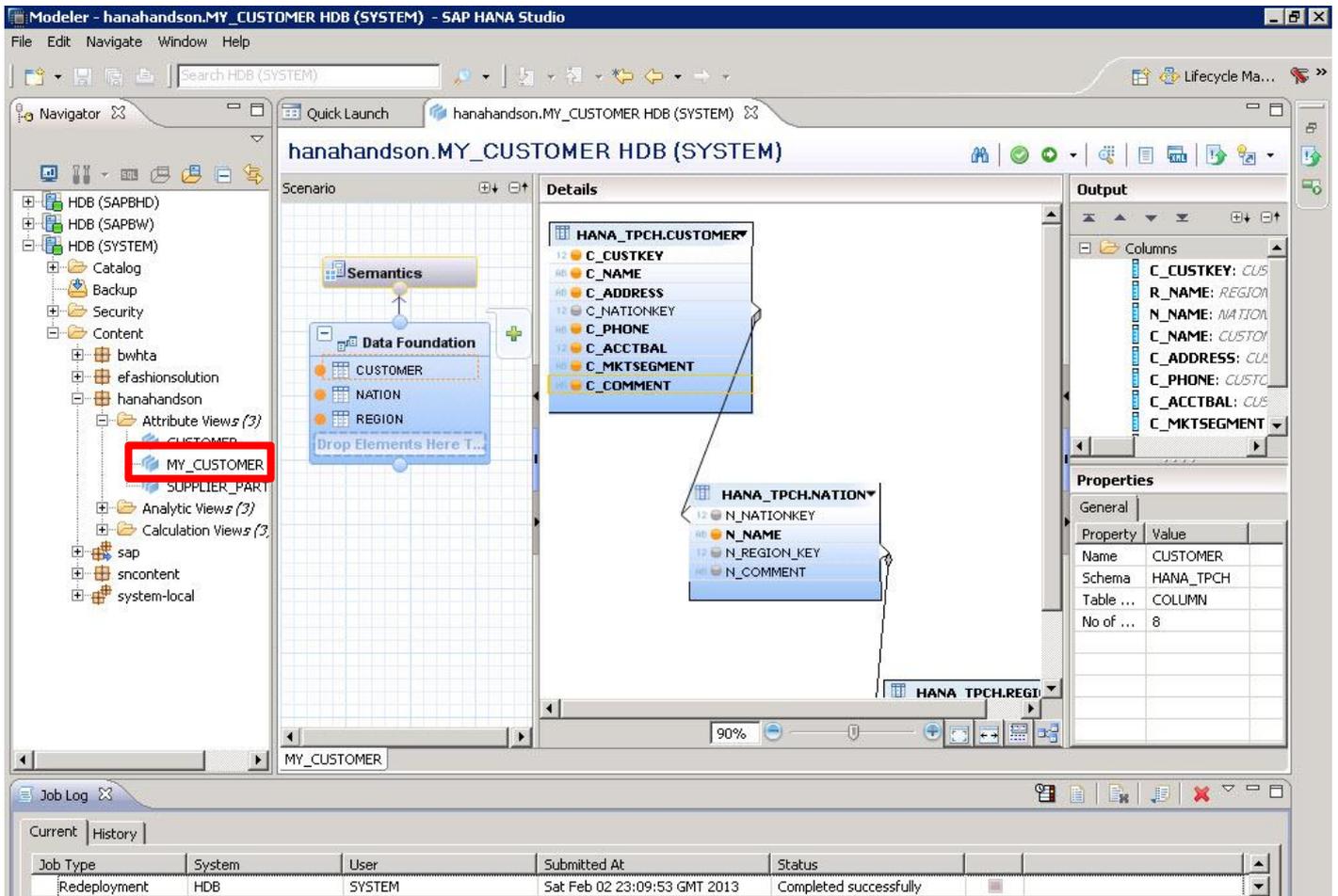


The next step is to activate the View for users to access it. You will notice that the My_Customer Attribute View has a diamond in its icon in the Navigator Panel. This indicates that you haven't activated the view.

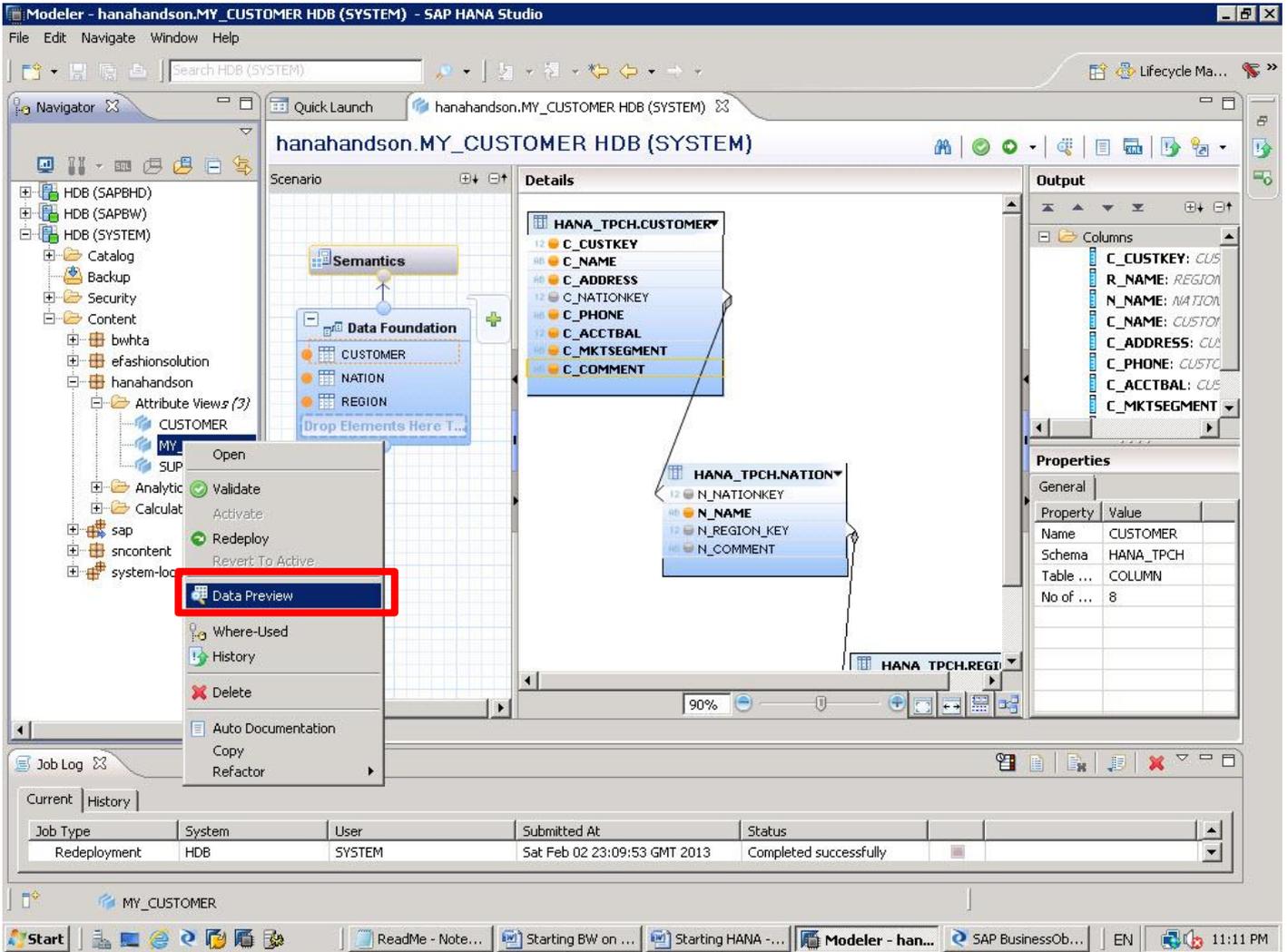
To activate the view select the icon next to where we just validated the view (a right facing arrow), click to activate:



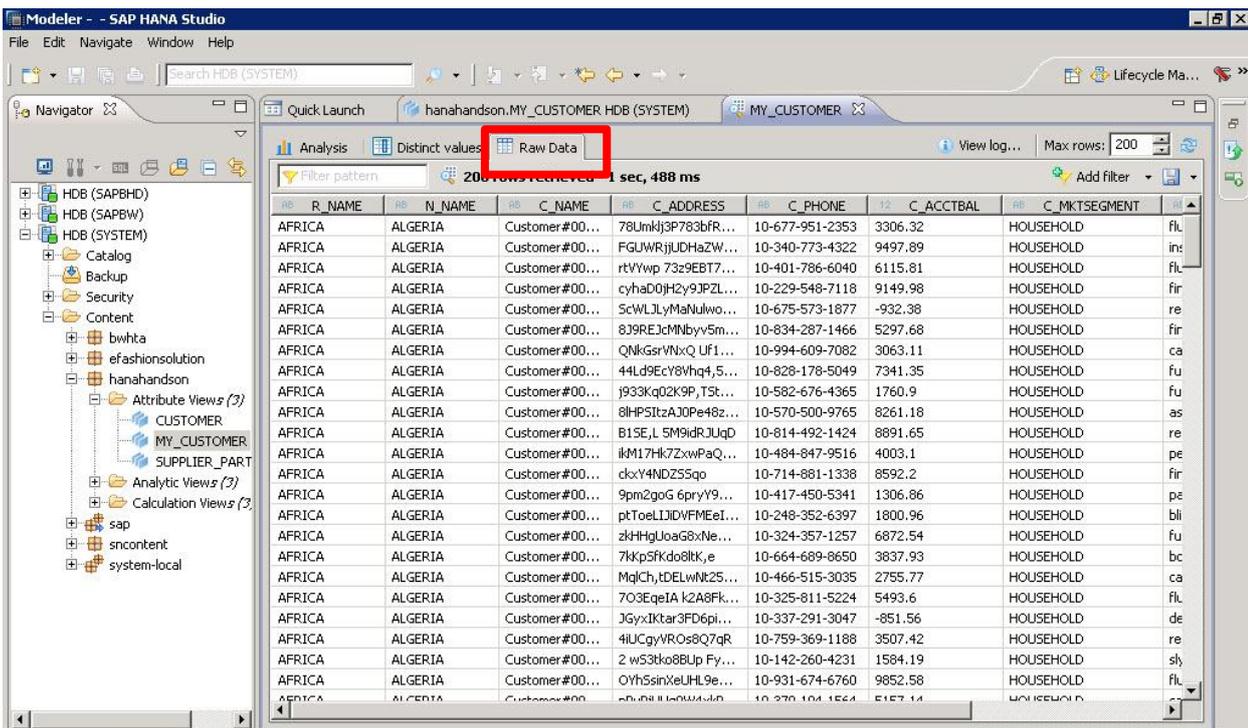
Your Attribute View is now activated:



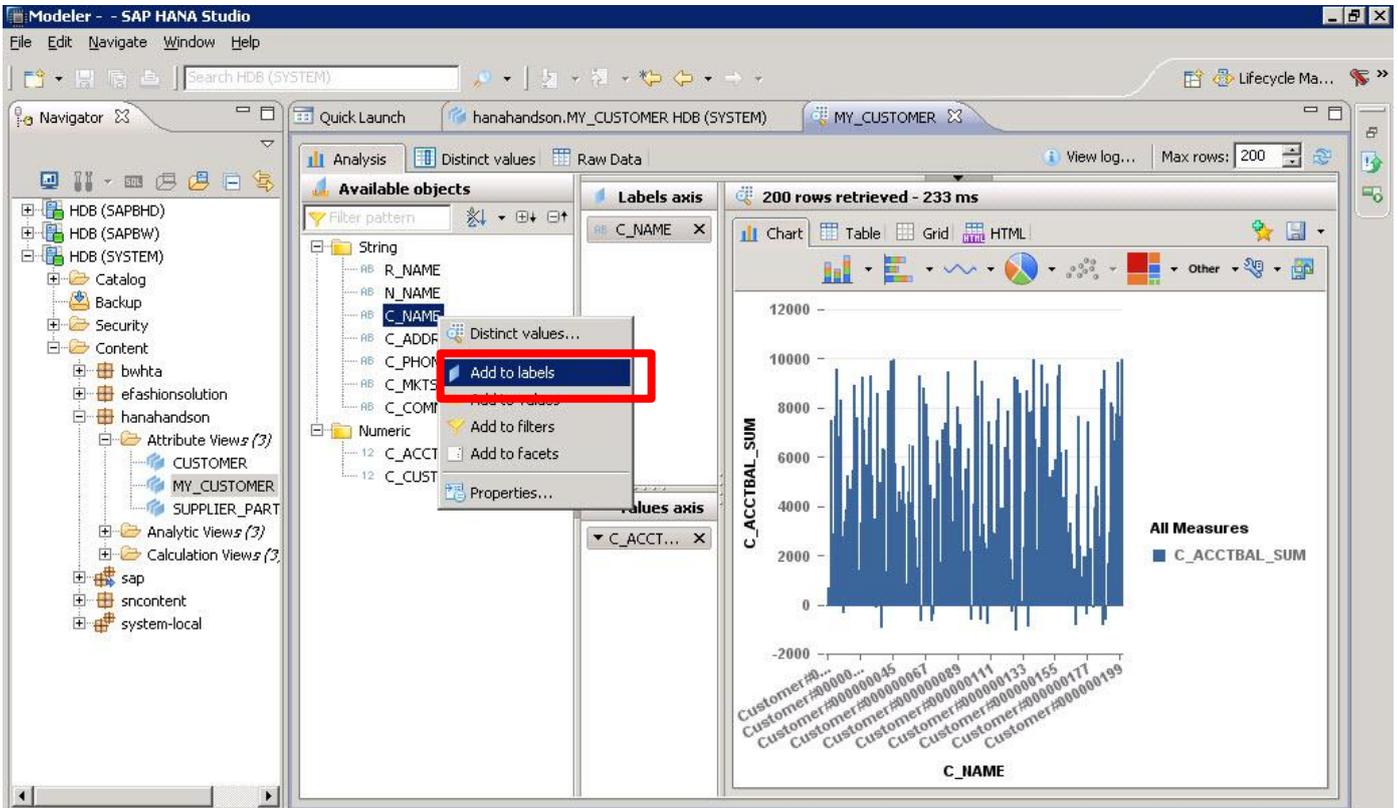
To test your view, right click on My_Customer and choose *Data Preview*



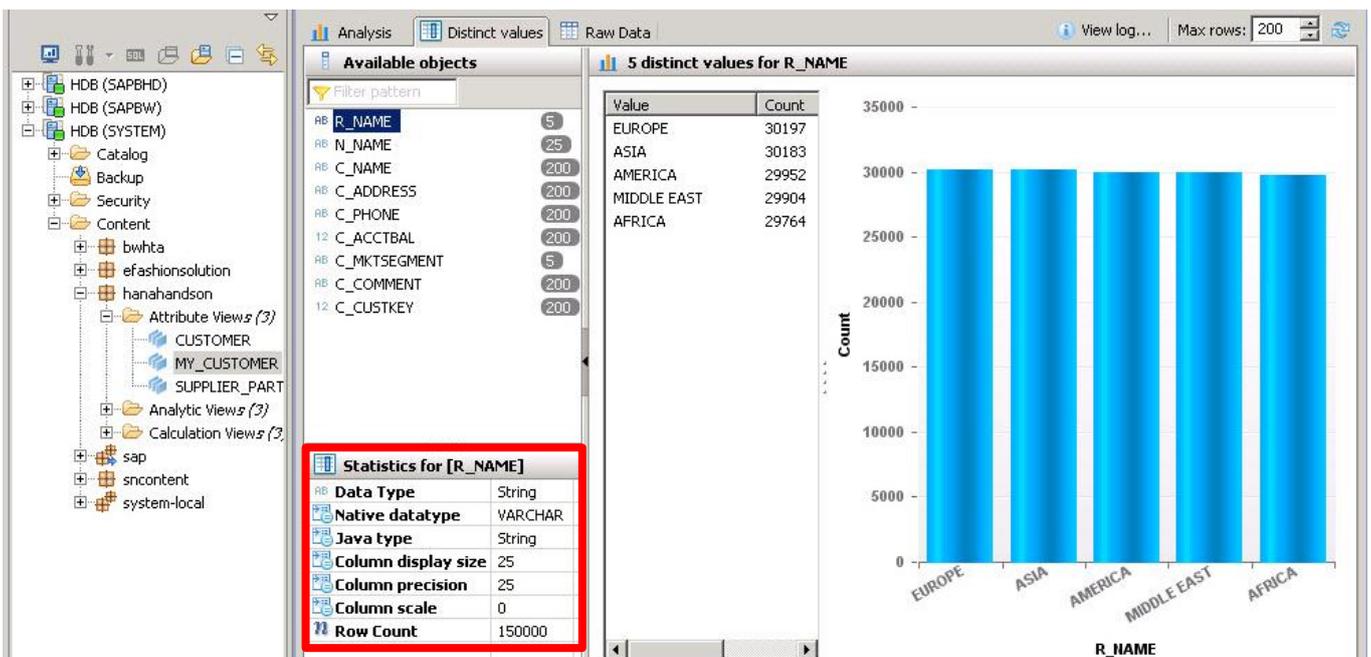
You should now see the data that your attribute view produces in the Raw Data tab.



To further explore the data we could look at the Analysis and Distinct values tabs. We can see the values are separated in to String and Numeric. Select C_NAME, right click and select 'Add to labels' – this will now appear in the Labels axis. Repeat the same for the C_ACCTBAL, this should now appear in the Value axis like below:



In the Distinct values tab we can see further details on the attributes stored in the SAP HANA database such as the; data type, column display size, java type and row count.



Analytic Views

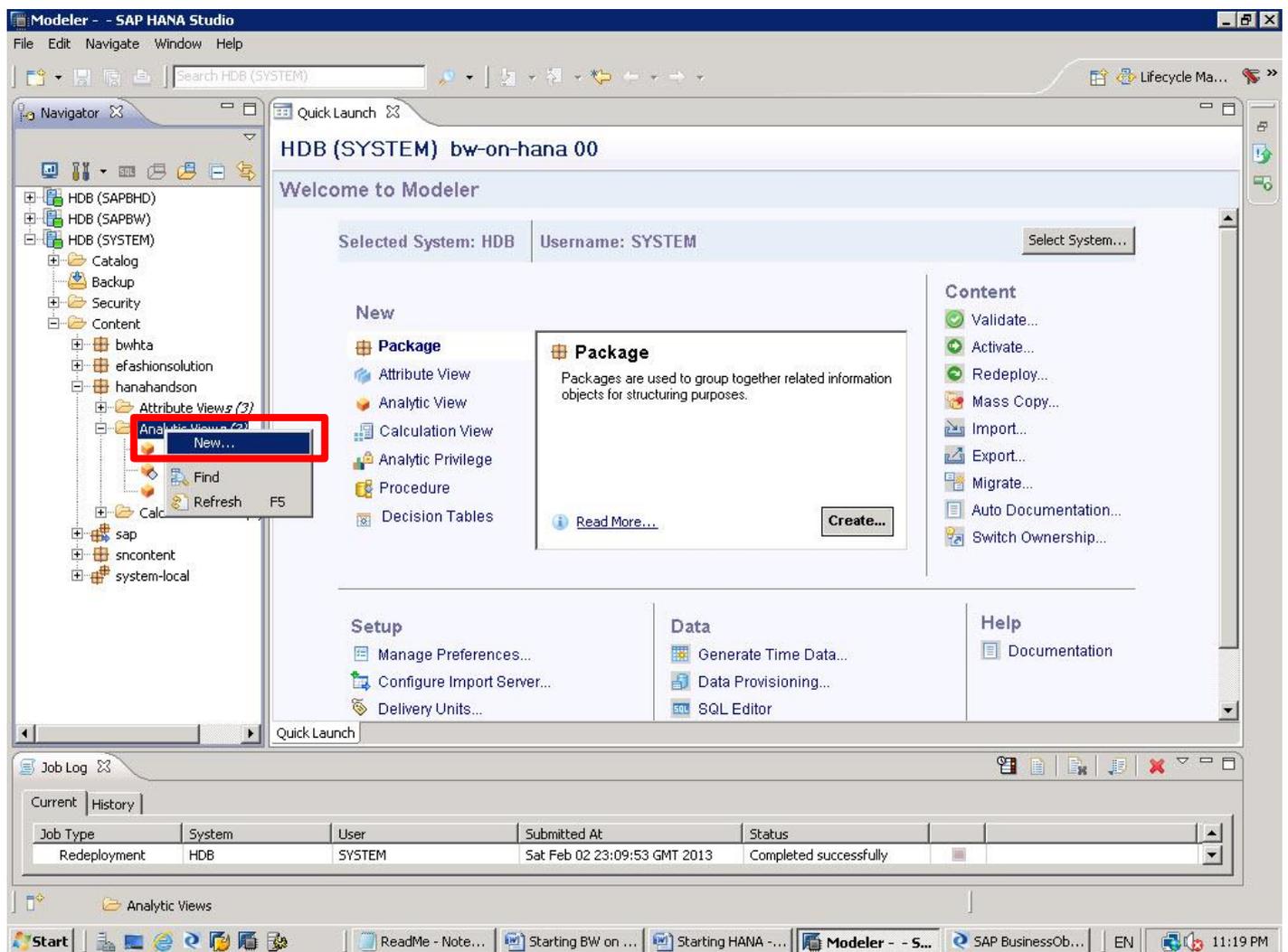
Analytic views are used to build a data foundation based on transactional tables. You can create a selection of measures (sometimes referred to as key figures), add attributes and join attribute views.

Analytic views leverage the computing power of SAP HANA to calculate aggregate data, e. g. the number of sold cars per country, or the maximum power consumption per day. They are defined on at least one *fact table*, i. e. a table which contains e. g. one row per sold car or one row per power meter reading, or more generally speaking, some form of business transaction records. Fact tables can be joined to allow access to more detailed data using a single analytic view. Analytic views can be defined on a single table, or joined tables.

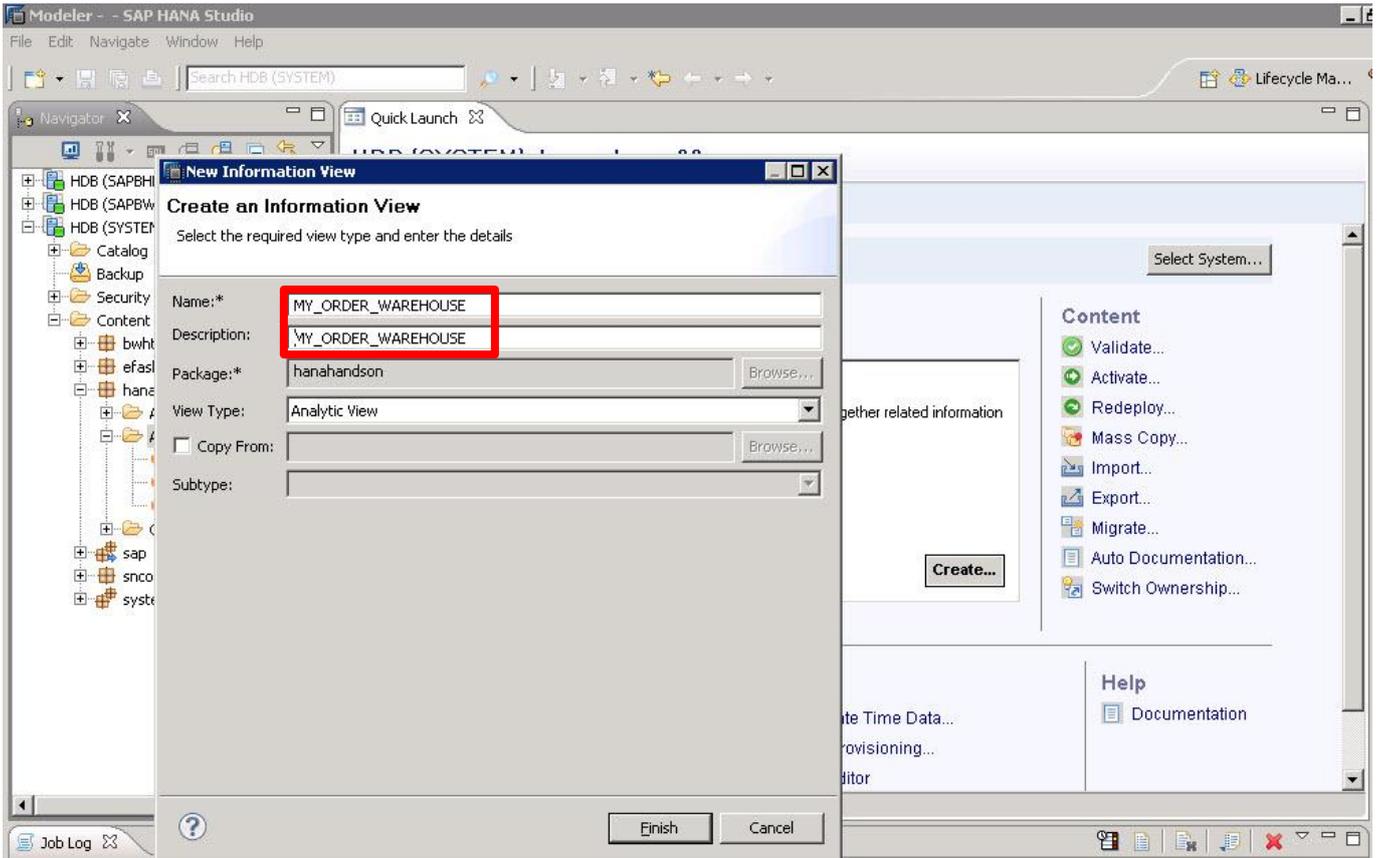
Analytic views can contain two types of attributes (or columns), so-called *measures* and *normal attributes*. Measures are attributes for which an aggregation must be defined. Normal attributes can be handled as regular columns and there is no need for aggregation.

Hands-on Labs

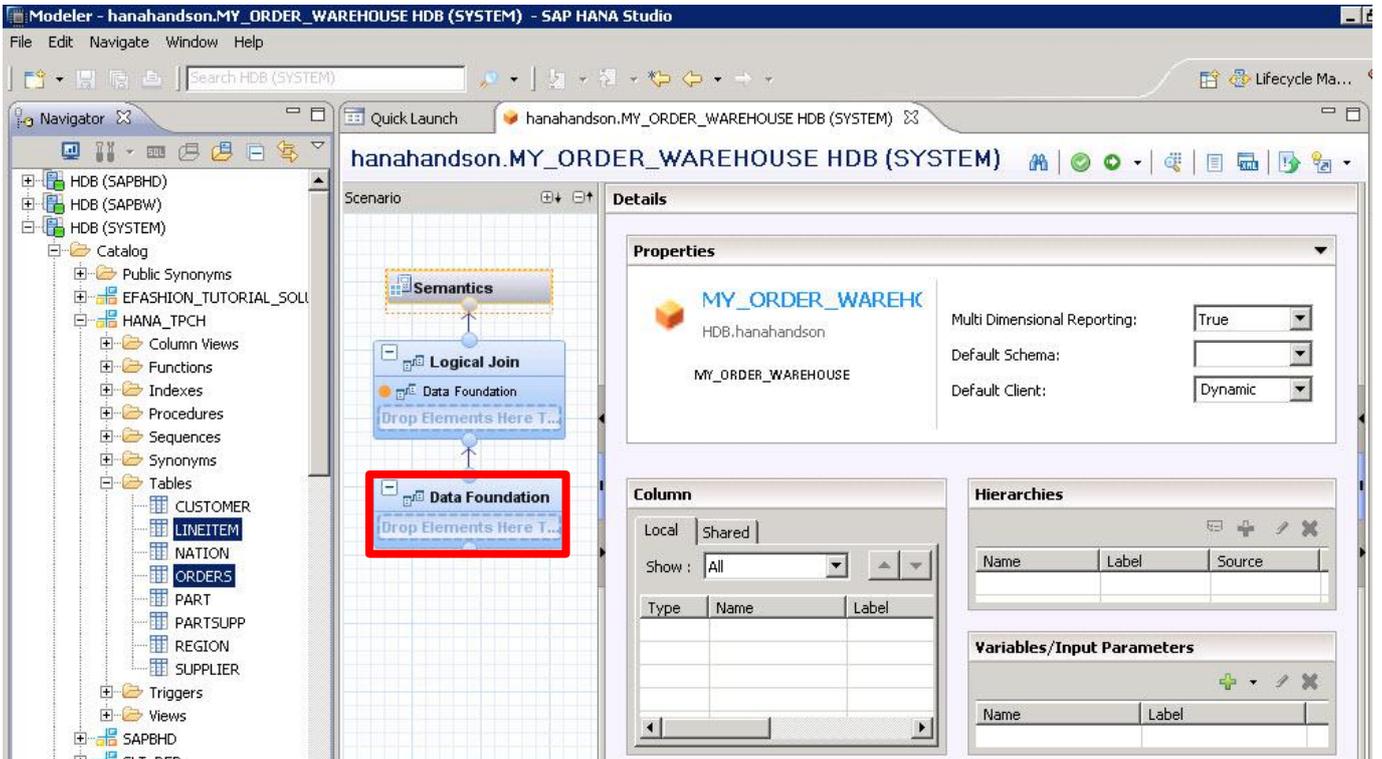
Now we will create an Analytic View. Right click on Analytic View under the hanahandson Package and choose New => Analytic View



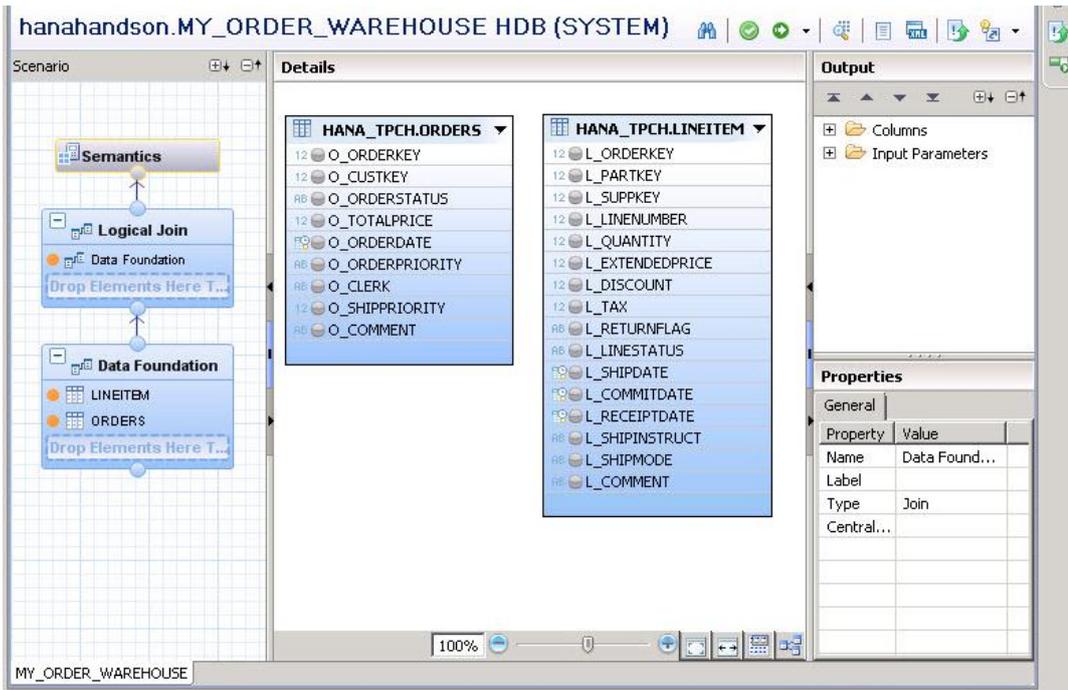
Name your Analytic View MY_ORDER_WAREHOUSE and click *Next*.



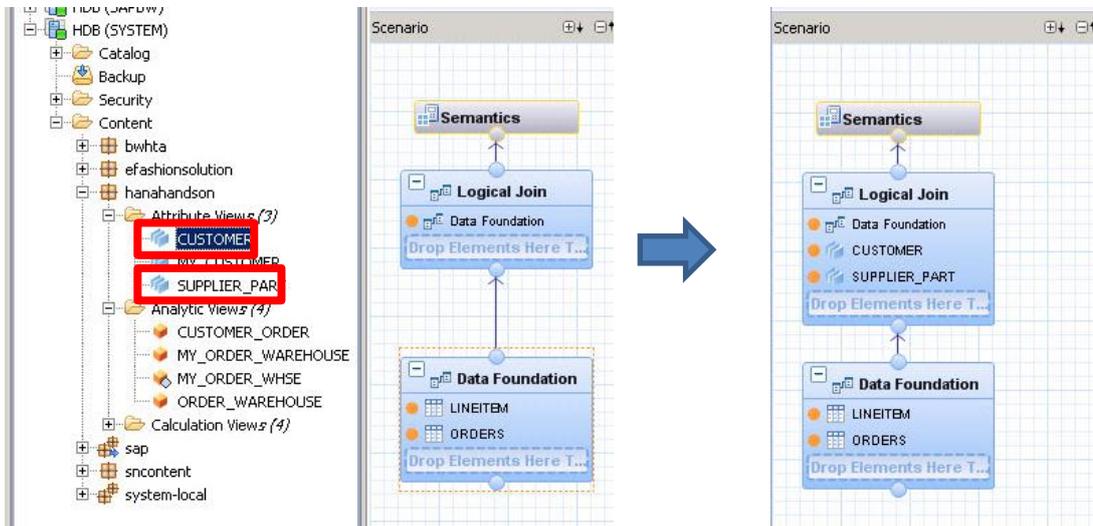
From the HANA_TPCH schema add the Orders and LINEITEM tables to your view by select either one by one or holding CTRL and then dragging them into the Data Foundation area on the canvas.



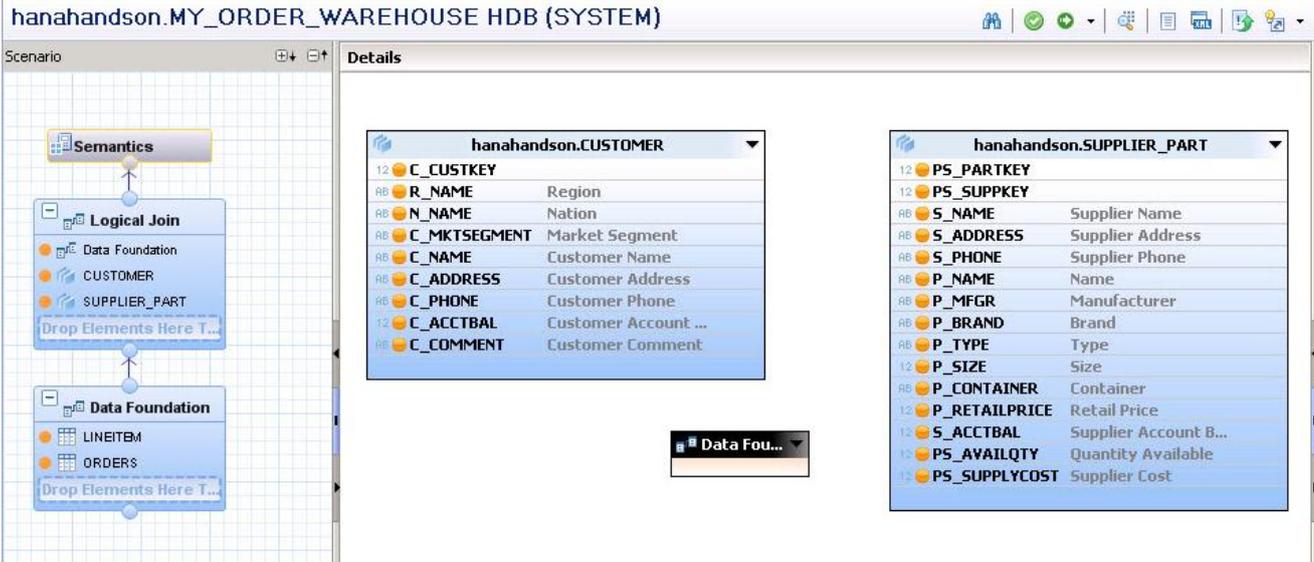
You should now have the below showing tables: ORDERS and LINEITEM:



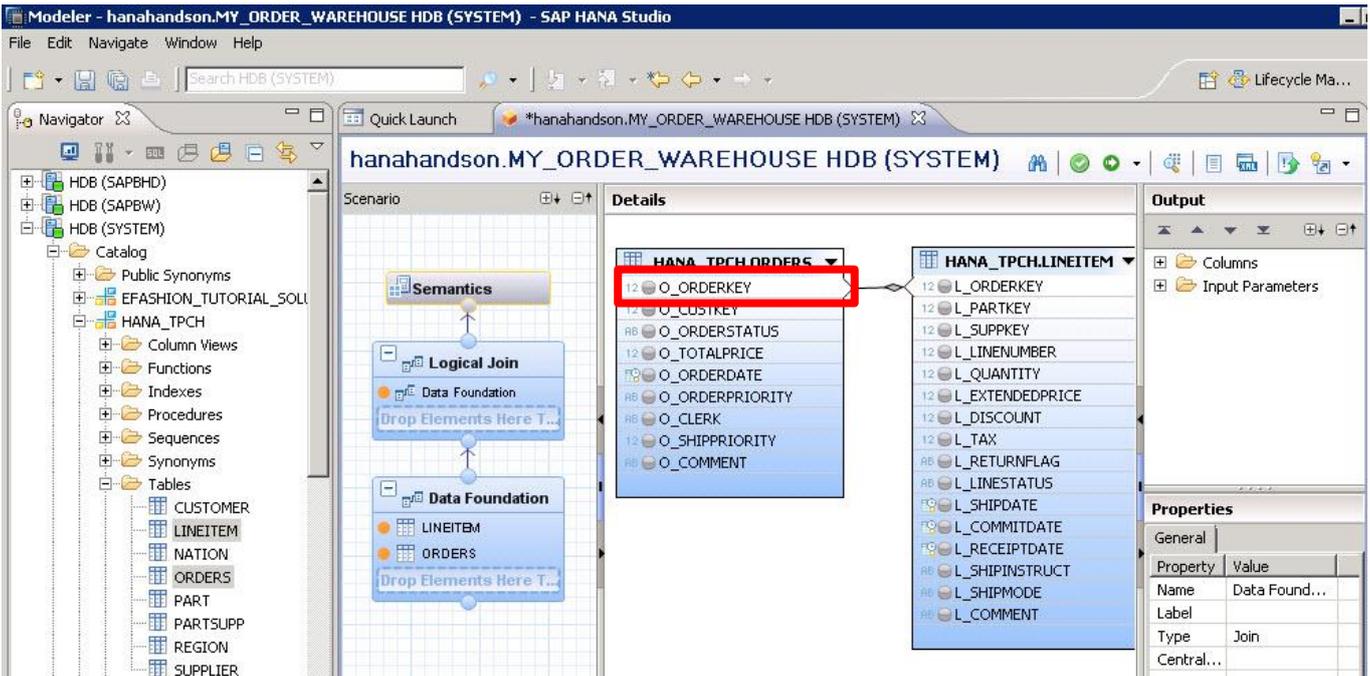
We will now add two Attribute Views: hanahandson.CUSTOMER and hanahandson.SUPPLIER_PART. To do this we will follow the same principal by expanding the content folder then dragging the Customer and Supplier attributes into logical join area.



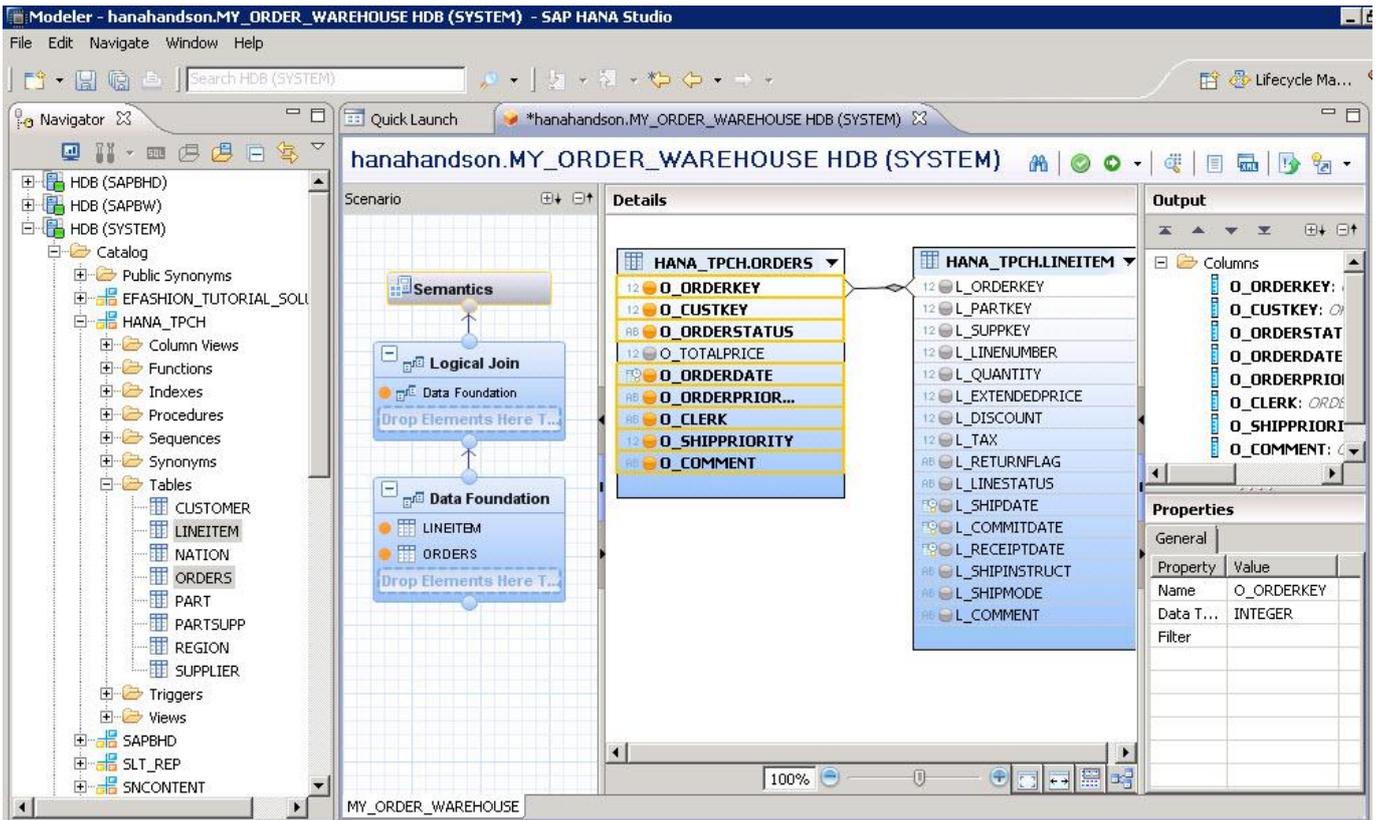
We should now see the following (notice currently the Data Foundation is unpopulated):



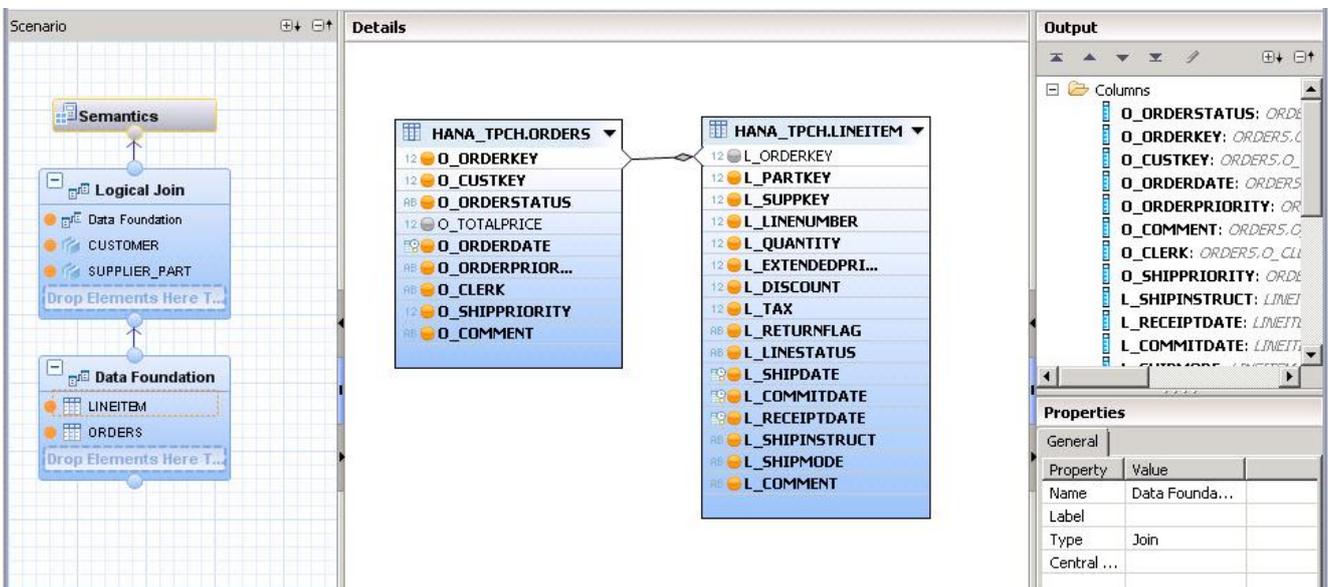
Selecting the Data Foundation area we should now see the tables presented in the canvas. Here we will join the two fact tables by clicking on Orders.O_ORDERKEY and dragging to Lineitem.L_ORDERKEY:



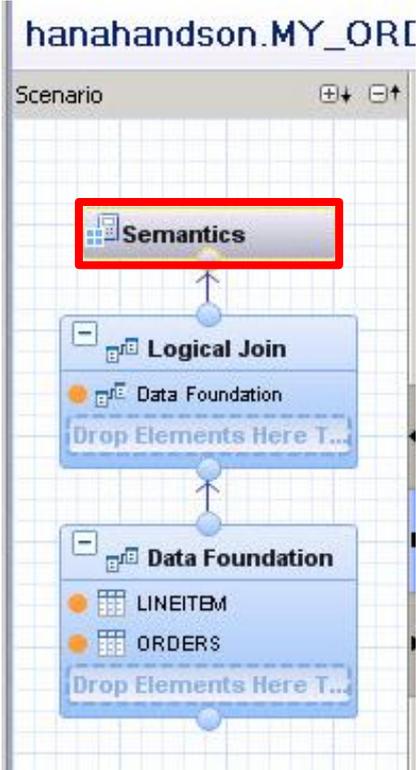
Now add the attributes to the output from Orders: O_Orderkey, O_Custkey, O_Orderstatus, O_Orderdate, O_Orderpriority, O_Clerk, O_Shippriority, O_Comment



And repeat the same for the whole of the LINEITEM table apart from L_ORDERKEY, you should now have the below:



Now we are going to define the attribute type. First select the Semantic level in our scenario:



You should then see the following screen where we will define our attributes:

ORDERS_COPY

Details

Properties

ORDERS_COPY
HDB.hanahandson

Multi Dimensional Reporting: True
Default Schema:
Default Client: Dynamic

Column

Type	Name	Label	Aggregation	Variab
RB	O_ORDERS...	O_ORDERSTATUS		
L2	O_ORDERKEY	O_ORDERKEY		
L2	O_CUSTKEY	O_CUSTKEY		
RB	O_ORDERD...	O_ORDERDATE		
RB	O_ORDERP...	O_ORDERPRIO...		
RB	O_COMMENT	O_COMMENT		
RB	O_CLERK	O_CLERK		
L2	O_SHIPPRI...	O_SHIPPRIORITY		
RB	L_SHIPINST...	L_SHIPINSTRUCT		
RB	L_RECEIPT...	L_RECEIPTDATE		
RB	L_COMMIT...	L_COMMITDATE		

Hierarchies

Name	Label	Source

Variables/Input Parameters

Name	Label

To change the type we will select the attribute and then a dropdown menu will appear prompting us to either select the field as an Attribute or Measure. Repeat this for the remainder (or alternatively for a faster way we could select multiple attributes by holding the shift key and select the relevant icon marked in the small red box). The fields: L_TAX, L_QUANTITY, L_DISCOUNT and L_EXTENDEDPRICE are to be selected as measures.

Scenario

- Semantics
- Logical Join
 - Data Foundation
 - CUSTOMER
 - SUPPLIER_PART
- Data Foundation
 - LINEITEM
 - ORDERS

Details

Properties

ORDERS_COPY
HDB.hanahandson

ORDERS_COPY

Multi Dimensional Reporting:

Default Schema:

Default Client:

Column

Local | Shared

Show: All

Type	Name	Label	Aggregation	Variable
Attribute	O_ORDERS...	O_ORDERSTATUS		
Measure			JE	
RB	O_ORDERP...	O_ORDERPRIO...		
RB	O_COMMENT	O_COMMENT		
RB	O_CLERK	O_CLERK		
12	O_SHIPPRI...	O_SHIPPRIORITY		
RB	L_SHIPINST...	L_SHIPINSTRUCT		
L	L_RECEIPT...	L_RECEIPTDATE		
L	L_COMMIT...	L_COMMITDATE		

Hierarchies

Name	Label	Source

Variables/Input Parameters

Name	Label

ORDERS_COPY

You should now have something similar to the below:

hanahandson.ORDERS_COPY HDB (SYSTEM)

Scenario

- Semantics
- Logical Join
 - Data Foundation
 - CUSTOMER
 - SUPPLIER_PART
- Data Foundation
 - LINEITEM
 - ORDERS

Details

Properties

ORDERS_COPY
HDB.hanahandson

ORDERS_COPY

Multi Dimensional Reporting:

Default Schema:

Default Client:

Column

Local | Shared

Show: All

Type	Name	Label	Aggregation	Variable
12	L_TAX	L_TAX	SUM	
12	L_QUANTITY	L_QUANTITY	SUM	
RB	L_RETURNF...	L_RETURNFLAG		
12	L_LINENUM...	L_LINENUMBER		
12	L_SUPPKEY	L_SUPPKEY		
RB	L_LINESTAT...	L_LINESTATUS		
12	L_PARTKEY	L_PARTKEY		
12	L_DISCOUNT	L_DISCOUNT	SUM	
RB	L_COMMENT	L_COMMENT		
12	L_EXTENDE...	L_EXTENDEDPR...	SUM	

Hierarchies

Name	Label	Source

Variables/Input Parameters

Name	Label

Switch to the Logical View so we can see the Attribute Views as well as the Data Foundation (which is the joined fact tables ORDERS and LINEITEM):

The screenshot shows the SAP HANA Logical View interface for the database 'hanahandson.MY_ORDER_WAREHOUSE HDB (SYSTEM)'. It displays three panels:

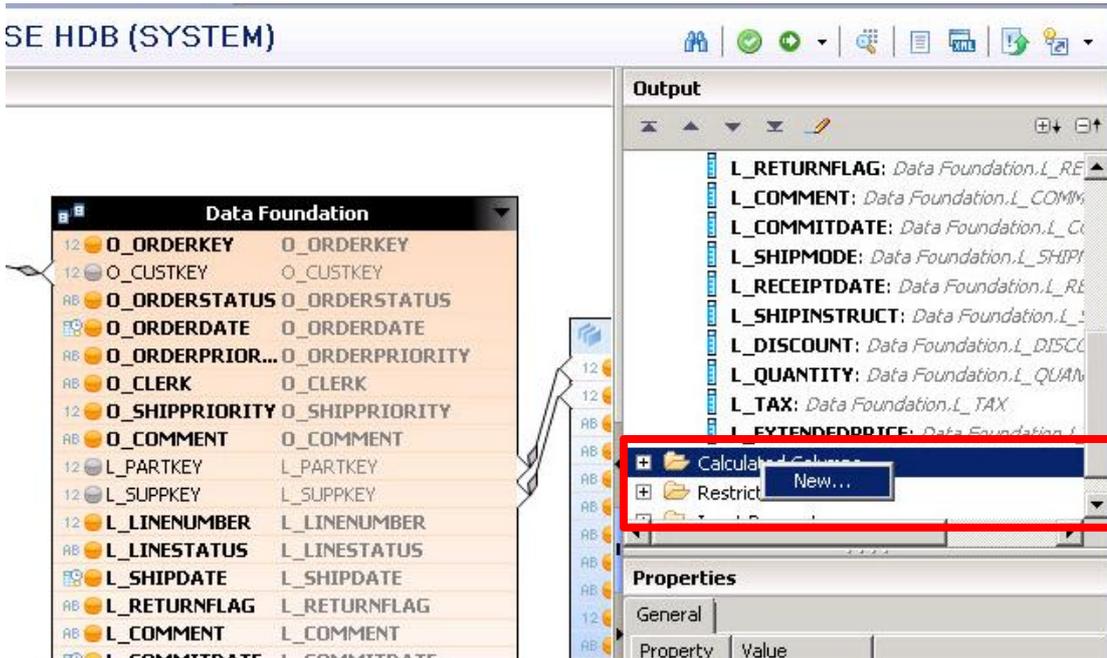
- hanahandson.CUSTOMER:**
 - C_CUSTKEY
 - R_NAME (Region)
 - N_NAME (Nation)
 - C_MKTSEGMENT (Market Segment)
 - C_NAME (Customer Name)
 - C_ADDRESS (Customer Address)
 - C_PHONE (Customer Phone)
 - C_ACCTBAL (Customer Account ...)
 - C_COMMENT (Customer Comment)
- Data Foundation:**
 - O_ORDERKEY (O_ORDERKEY)
 - O_CUSTKEY (O_CUSTKEY)
 - O_ORDERSTATUS (O_ORDERSTATUS)
 - O_ORDERDATE (O_ORDERDATE)
 - O_ORDERPRIOR... (O_ORDERPRIORITY)
 - O_CLERK (O_CLERK)
 - O_SHIPPRIORITY (O_SHIPPRIORITY)
 - O_COMMENT (O_COMMENT)
 - L_PARTKEY (L_PARTKEY)
 - L_SUPPKEY (L_SUPPKEY)
 - L_LINENUMBER (L_LINENUMBER)
 - L_LINESTATUS (L_LINESTATUS)
 - L_SHIPDATE (L_SHIPDATE)
 - L_RETURNFLAG (L_RETURNFLAG)
 - L_COMMENT (L_COMMENT)
 - L_COMMITDATE (L_COMMITDATE)
 - L_SHIPMODE (L_SHIPMODE)
 - L_RECEIPTDATE (L_RECEIPTDATE)
 - L_SHIPINSTRUCT (L_SHIPINSTRUCT)
 - L_DISCOUNT (L_DISCOUNT)
 - L_QUANTITY (L_QUANTITY)
 - L_TAX (L_TAX)
 - L_EXTENDEDPRI... (L_EXTENDEDPRI...)
- hanahandson.SUPPLIER_PART:**
 - PS_PARTKEY
 - PS_SUPPKEY
 - S_NAME (Supplier Name)
 - S_ADDRESS (Supplier Address)
 - S_PHONE (Supplier Phone)
 - P_NAME (Name)
 - P_MFGR (Manufacturer)
 - P_BRAND (Brand)
 - P_TYPE (Type)
 - P_SIZE (Size)
 - P_CONTAINER (Container)
 - P_RETAILPRICE (Retail Price)
 - S_ACCTBAL (Supplier Account B...)
 - PS_AVAILQTY (Quantity Available)
 - PS_SUPPLYCOST (Supplier Cost)

Join the Attribute Views to the Data Foundation: Customer.C_Custkey → Data Foundation.O_Custkey; Supplier_Part.PS_Partkey → Data Foundation.L_Partkey; Supplier_Part.PS_Suppkey → Data Foundation.L_Suppkey:

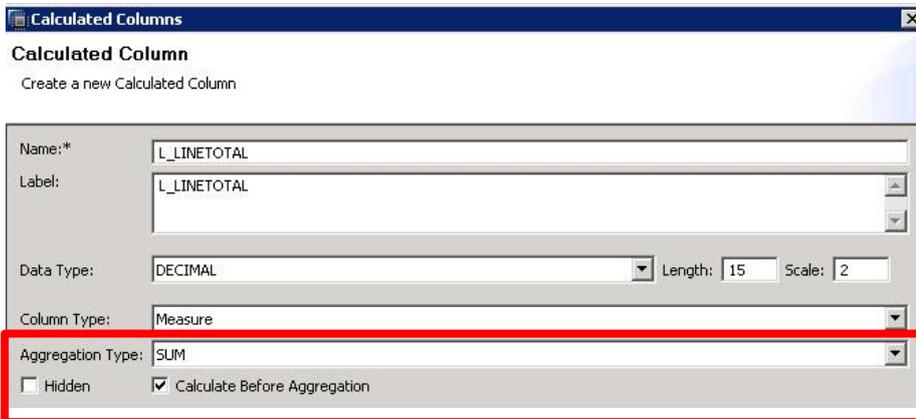
This screenshot shows the same SAP HANA Logical View interface as above, but with lines indicating the joins between the Attribute Views and the Data Foundation:

- A line connects **Customer.C_Custkey** to **Data Foundation.O_Custkey**.
- A line connects **Supplier_Part.PS_Partkey** to **Data Foundation.L_Partkey**.
- A line connects **Supplier_Part.PS_Suppkey** to **Data Foundation.L_Suppkey**.

Now add a Calculated Measure to your Analytic View. From the Output window on the right, right click on Calculated Measures and choose New.



Call the new column L_LINETOTAL and use a SUM Aggregation type, by selecting the 'Calculate before Aggregation' check box and ensure the 'Hidden' box kept unchecked.



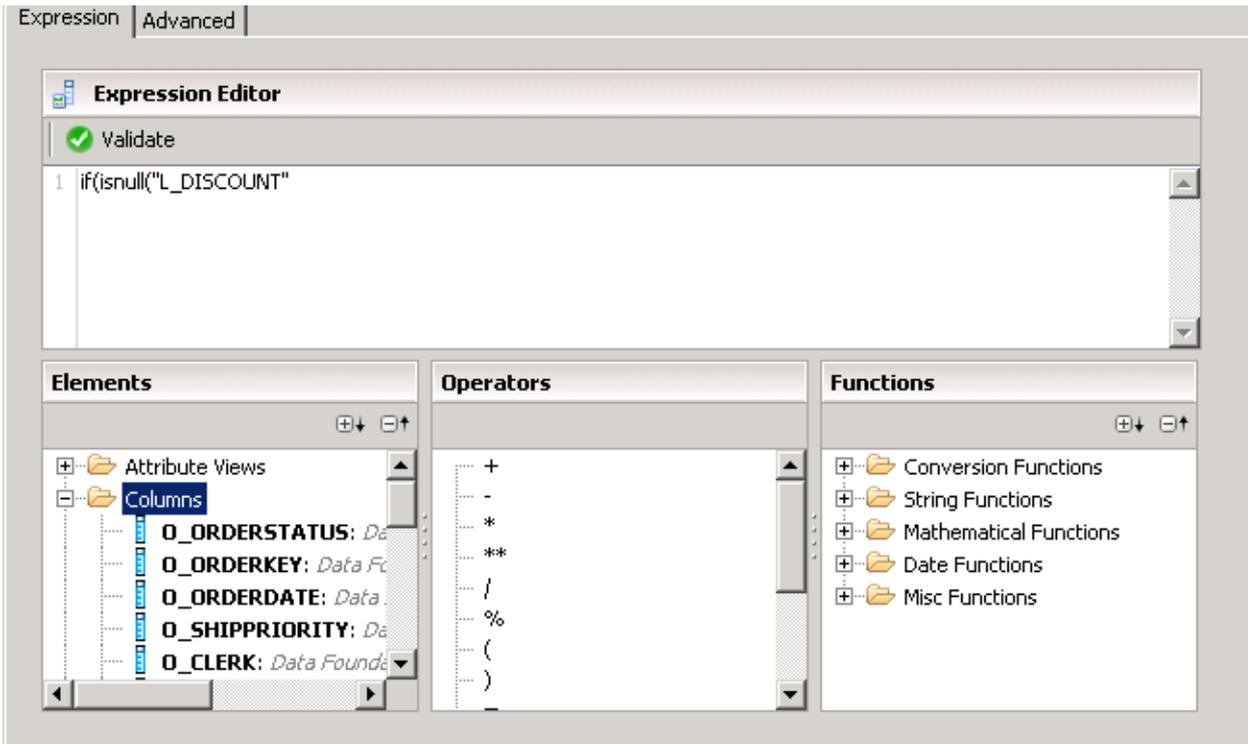
In the Expression Editor type: if(isnull(

Then expand the Columns folder in the Elements box and click on L_Discount to add it to the expression.

Then type:),0,((

And add the L_Extendedprice measure.

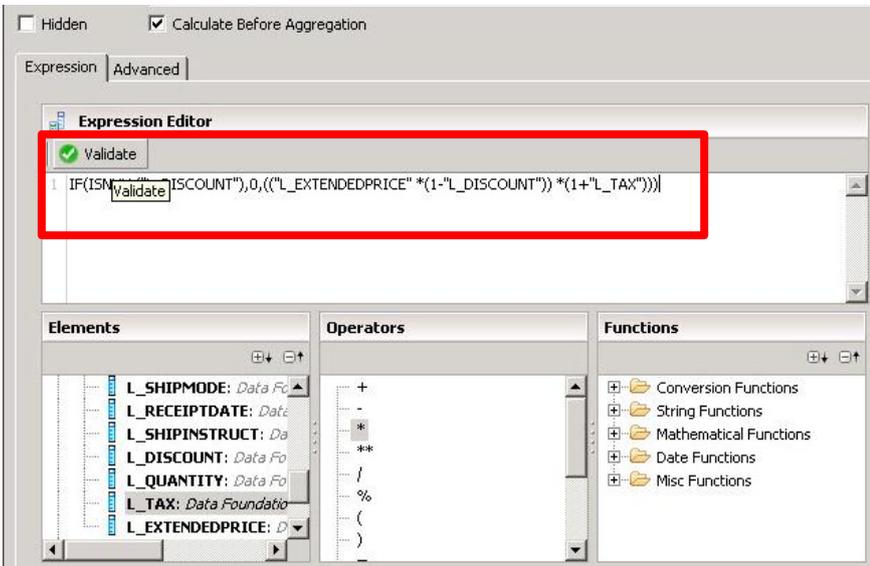
Next click the * from the Operators window:

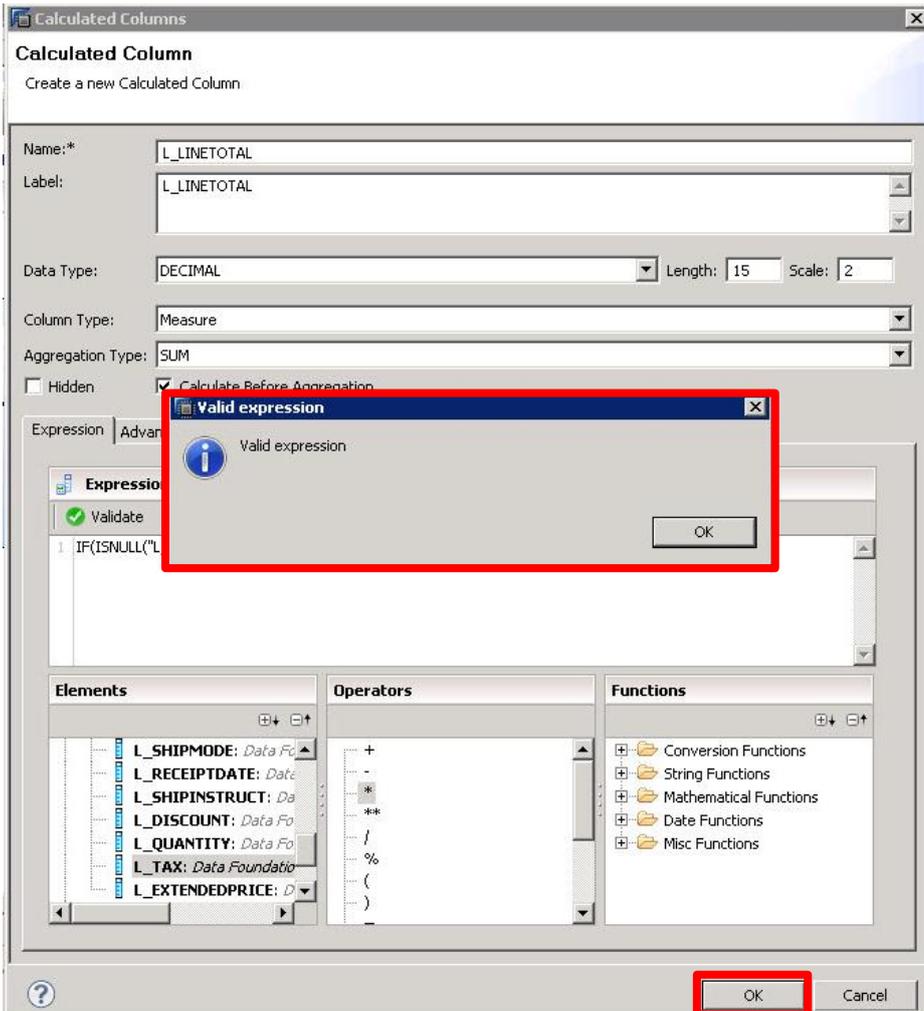


Continue in this until the formula in the Expression Editor reads:

`If(isnull("L_DISCOUNT"),0,((("L_EXTENDEDPRI" * (1 - "L_DISCOUNT")) * (1 + "L_TAX"))))`.

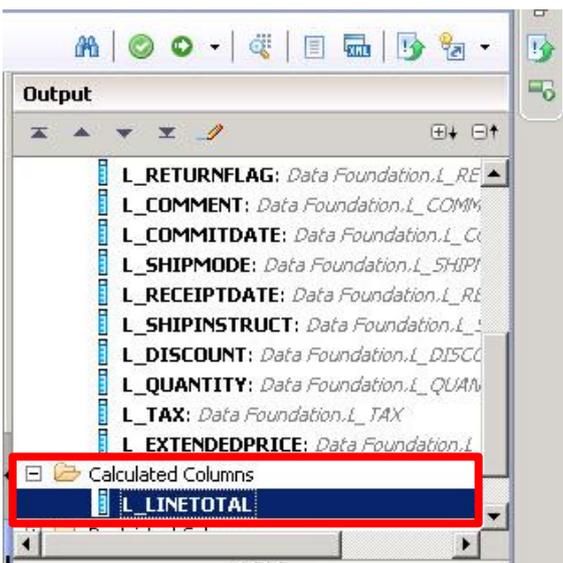
Once you have the below written correctly, check to validate it by selecting the validate icon:



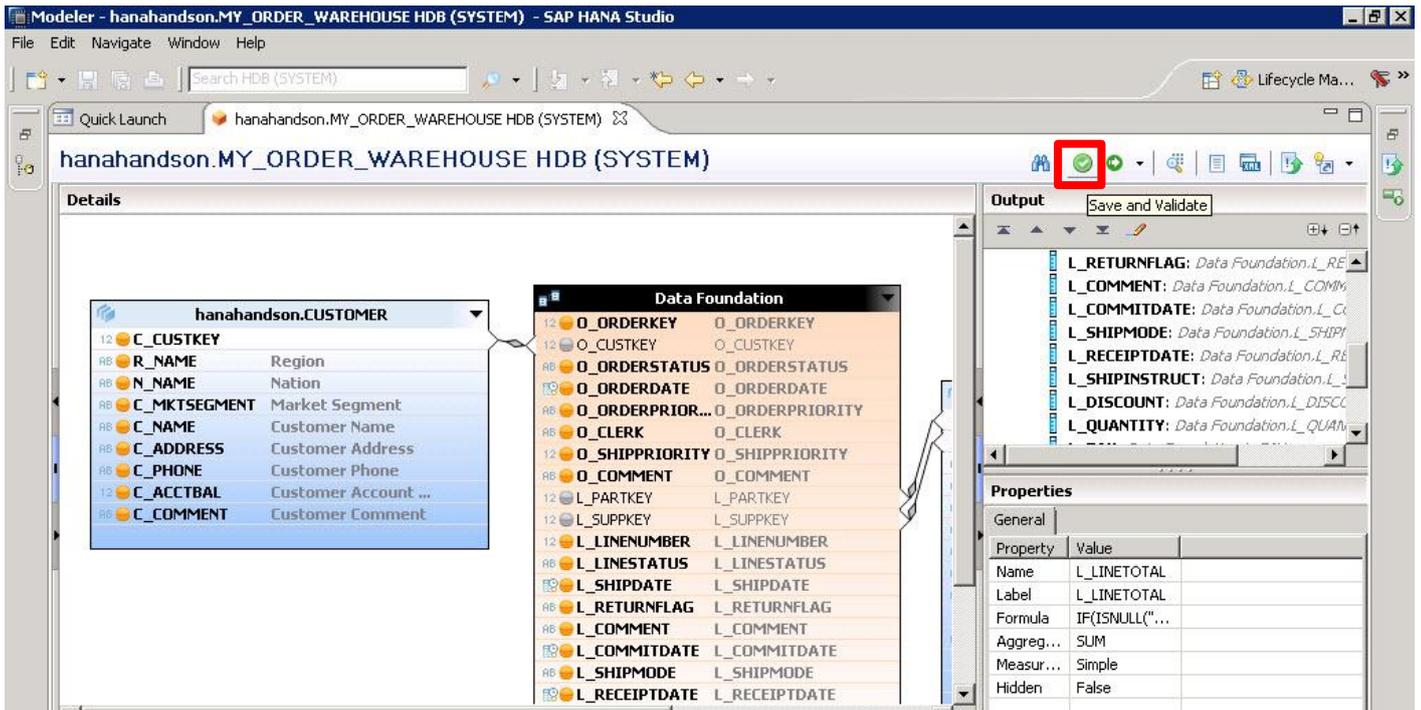


Click *Ok*

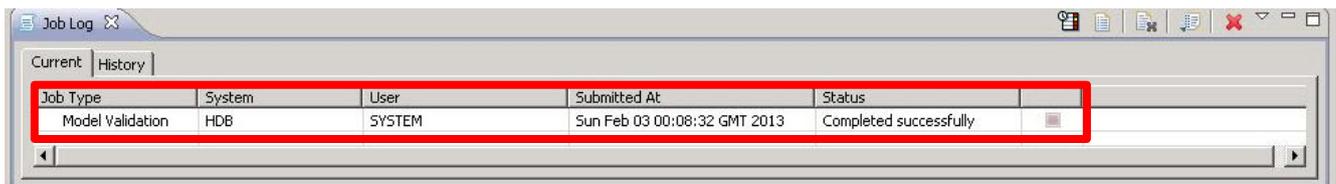
The Calculated Measure will now appear in the Output Window:



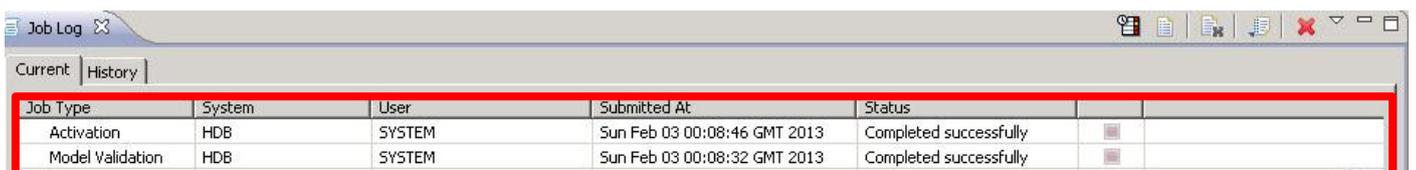
Save and Validate your view:



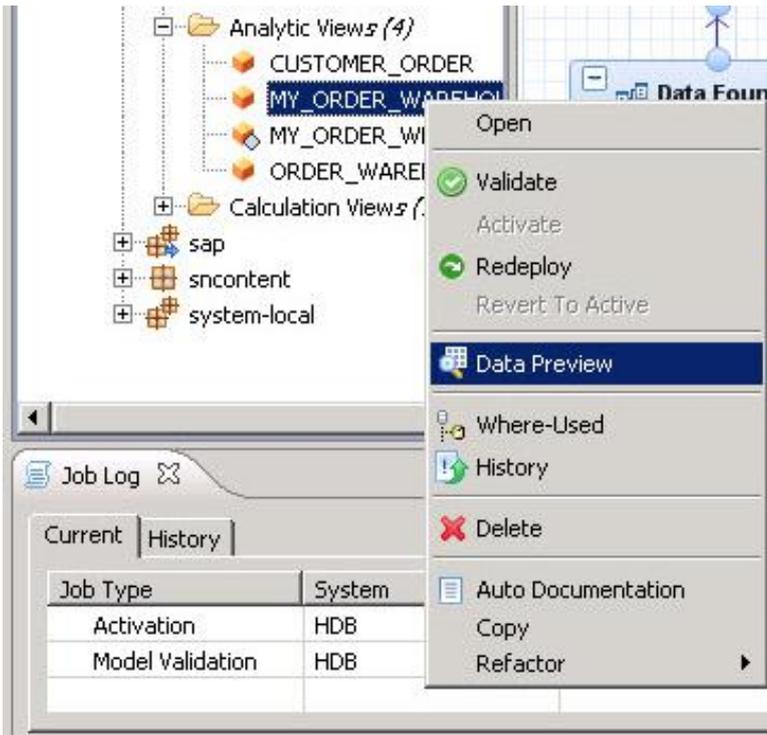
Confirm that it validated



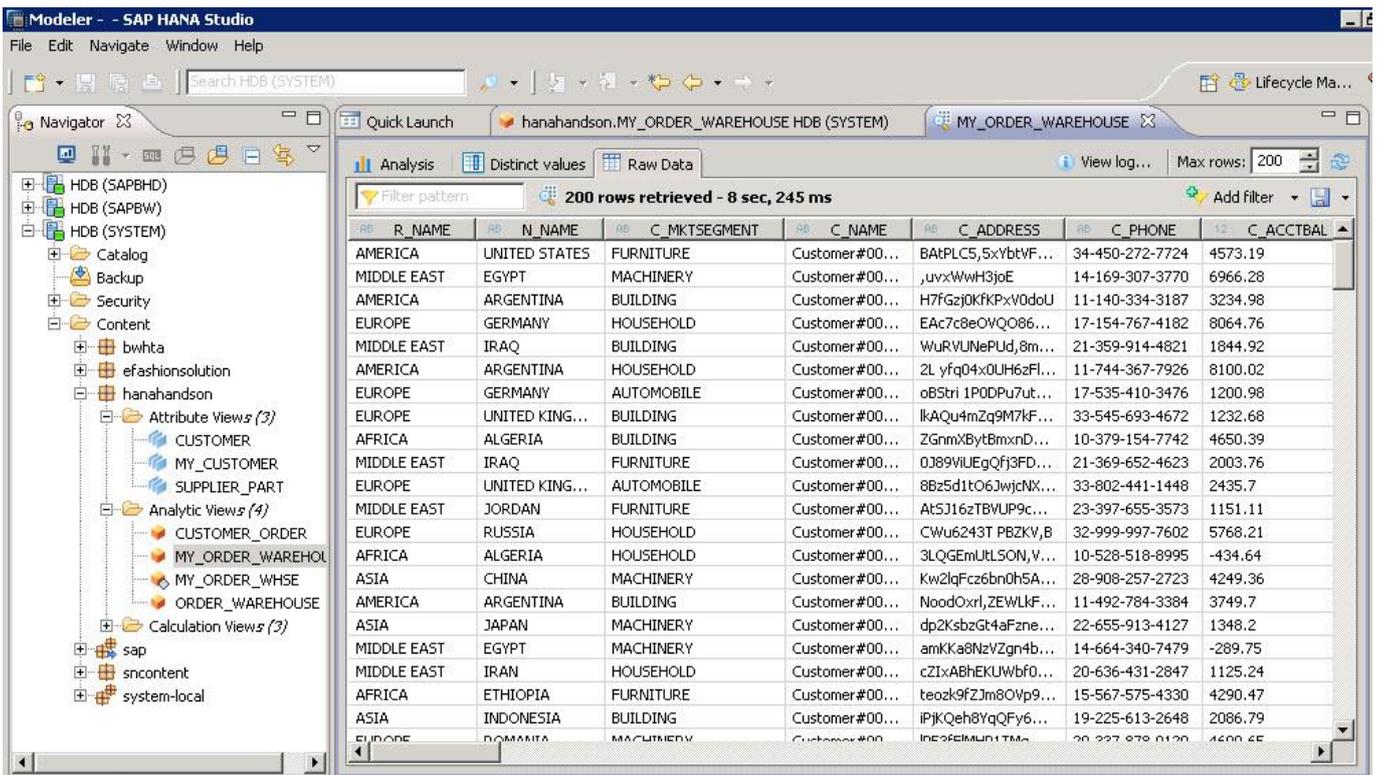
Activate your view – Click Activate and Confirm Activation:



Test your view by previewing the data. Right click My_Order_Warehouse → Data Preview

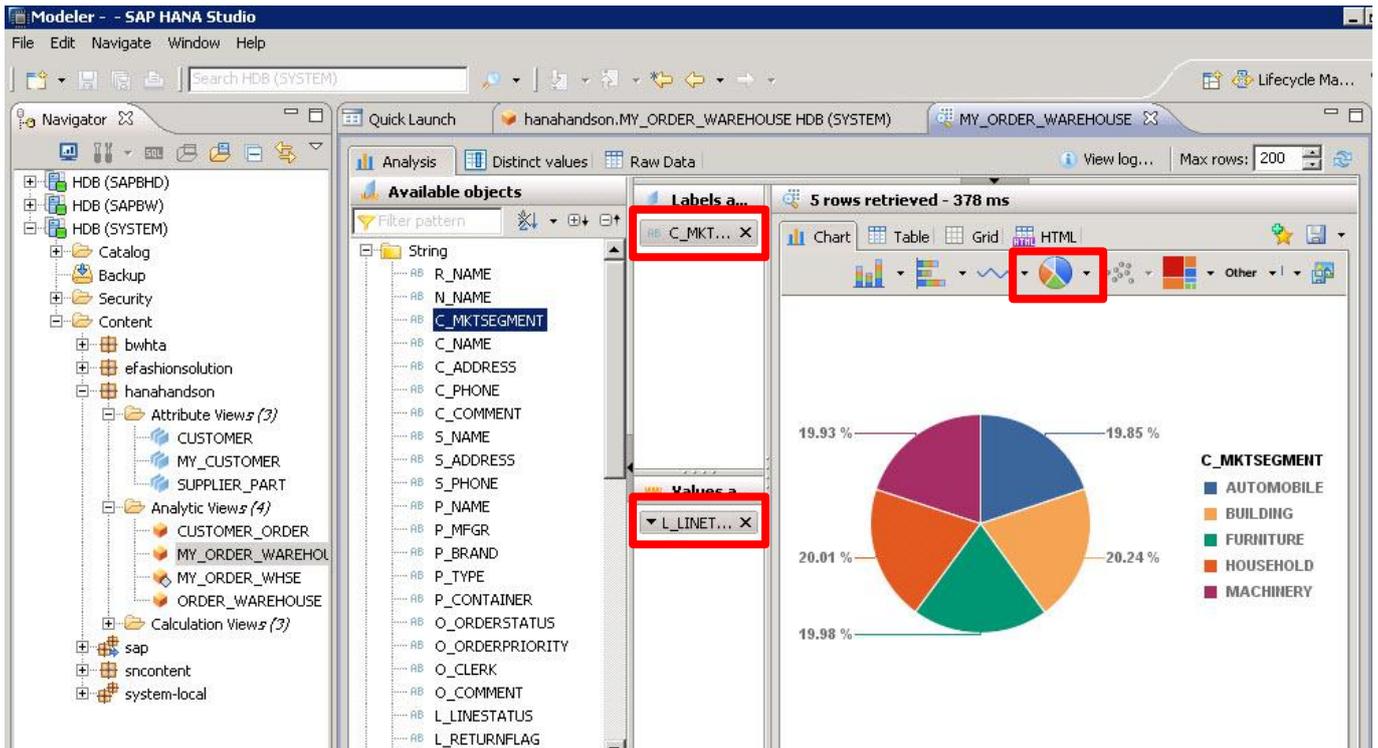


Data Preview of your Analytic View:



We can do some analysis on the data by selecting the Analysis tab. Click and drag C_MKTSEGMENT into the label axis and repeat the same for L_LINEITEM then drag the numeric attribute into the Value axis:

To see a different visualisation select the pie chart symbol.



Calculation Views

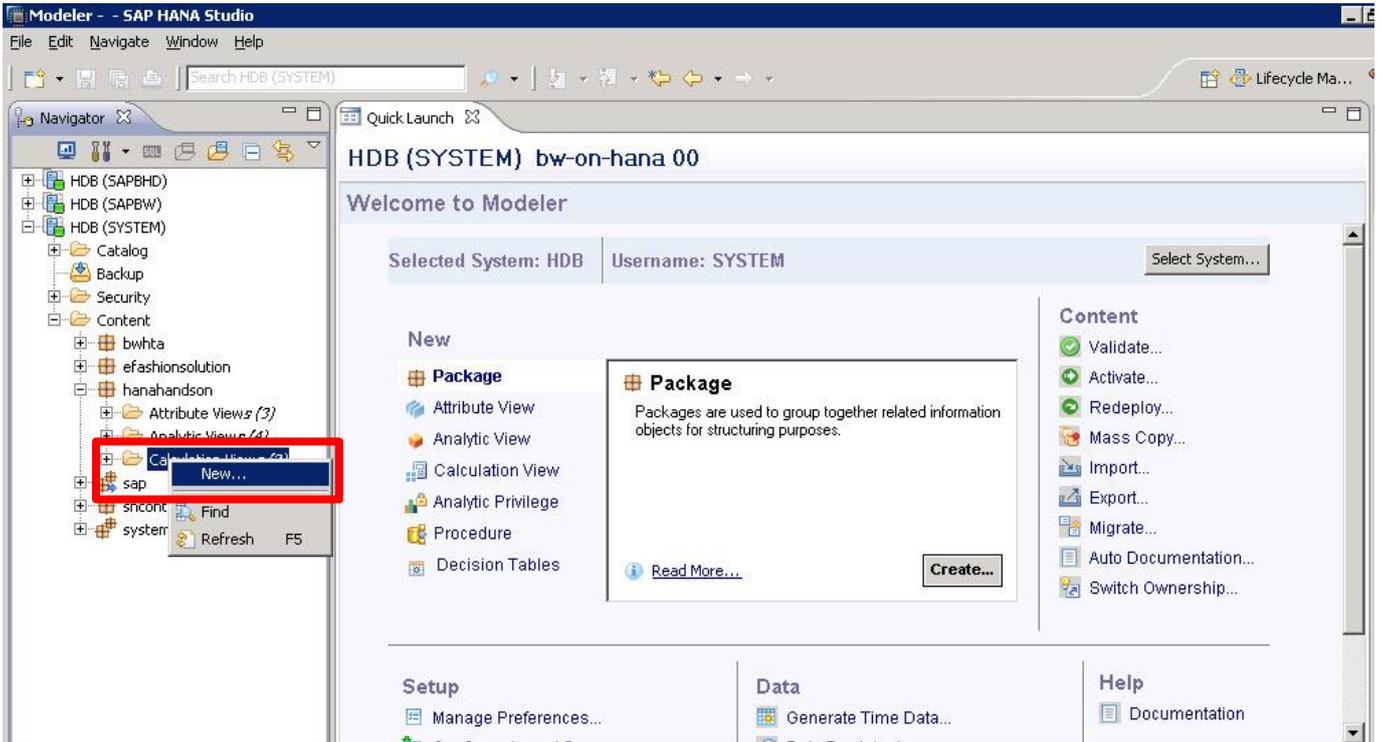
Calculation views are used to provide composites of other views. They are essentially a view which is based on the result from SQLScript. These scripts can join or union two or more data flows or invoke built-in or generic SQL functions.

Calculation views are defined as either *graphical views* or *scripted views* depending on how they are created. They can be used in the same way as analytic views, however, in contrast to analytic views it is possible to join several fact tables in a calculation view. Calculation views always have at least one measure.

Graphical views can be modeled using the graphical modeling features of the SAP HANA Information Modeler. Scripted views are created as sequences of SQLScript statements. In essence they are SQLScript procedures with certain properties.

Hands-on Labs

Right click on Calculation View and choose New => Calculation View



Name it MY_ORDERS_CY_PY. We will be making this as a graphical view so make sure that Graphical is the View Type. Click Next.

New Calculation View

Calculation View
This wizard creates a calculation view in the specified package.

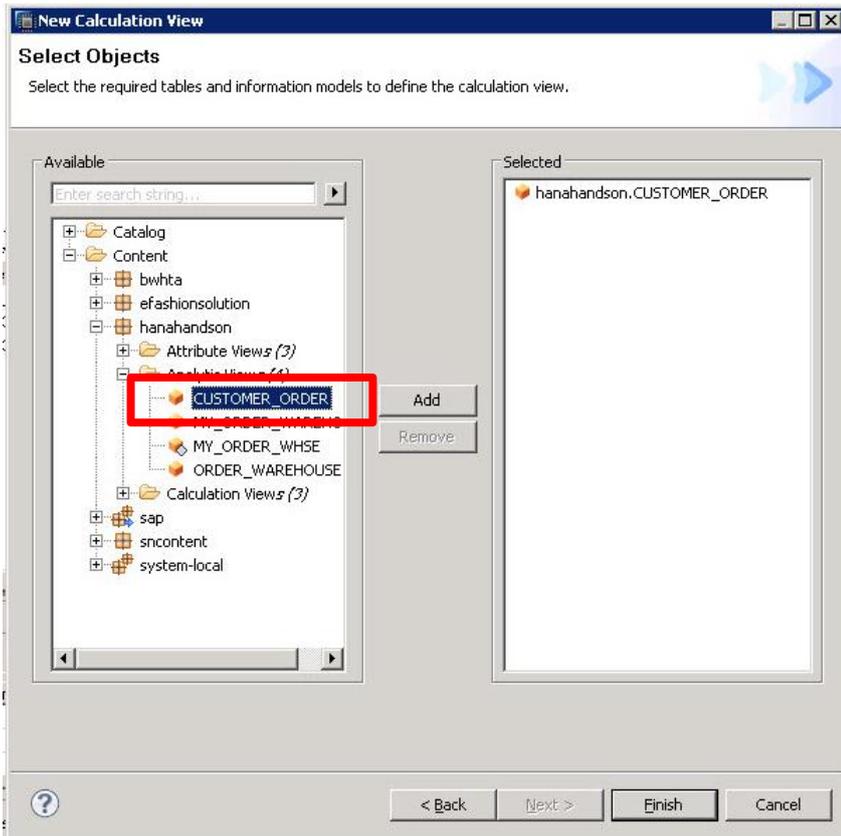
Name:* MY_ORDERS_CY_PY
Description MY_ORDERS_CY_PY
Package:* hanahandson

Create New
 Copy From

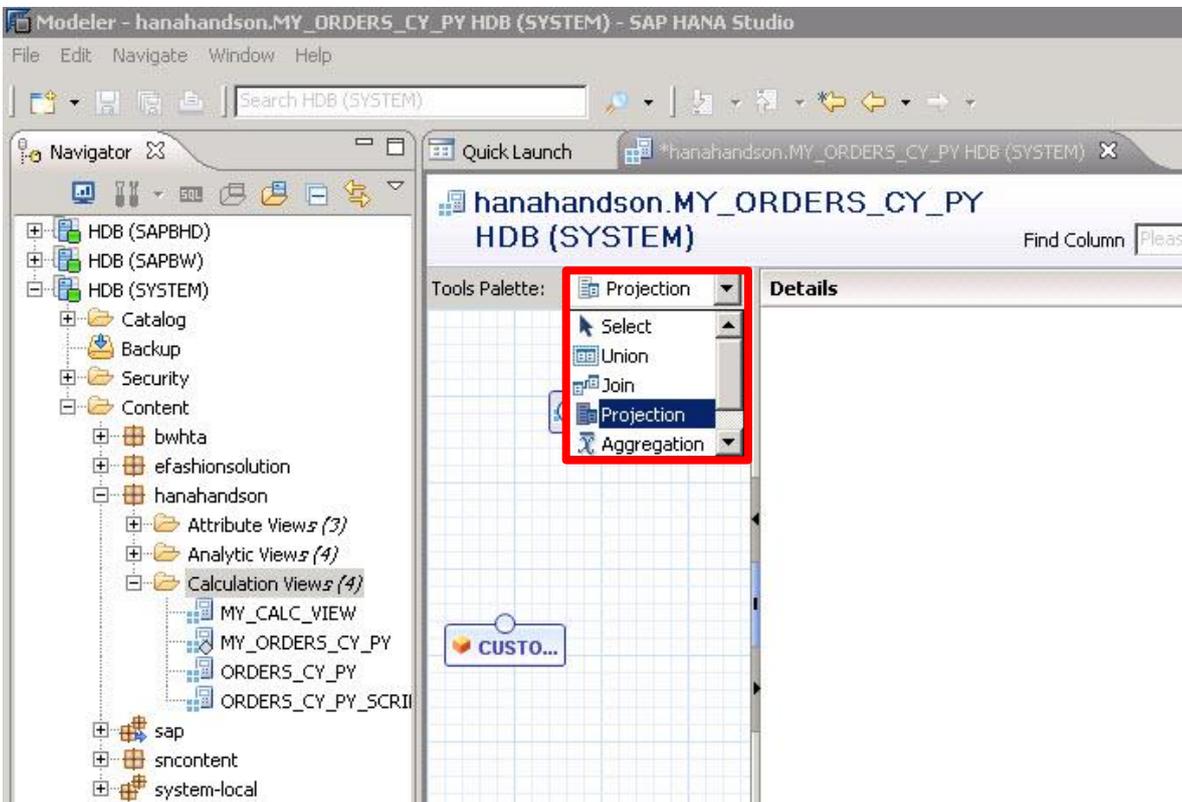
View Type
 Graphical
 SQL Script

Schema for conversion: SAPBHD
Run With Definer's Rights (User _SYS_REPO)

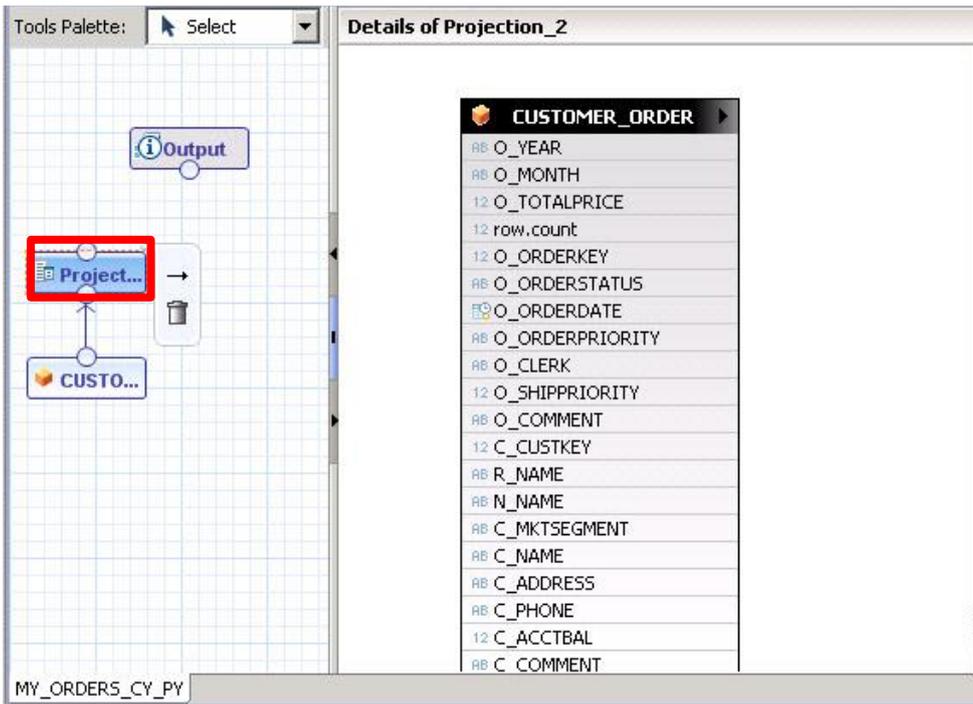
Expand the Content Folder. Expand the hanahandson Package and Analytic Views and add the Customer_order analytic view. Click Finish.



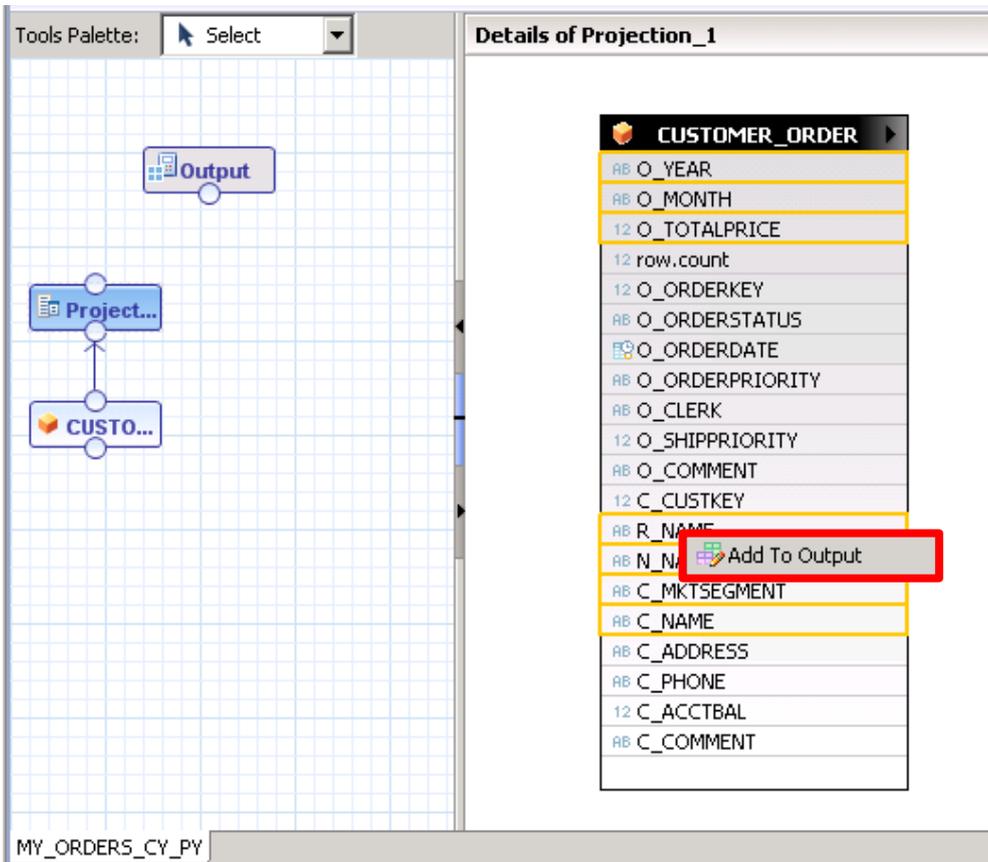
We need to add a Projection Component to our view.



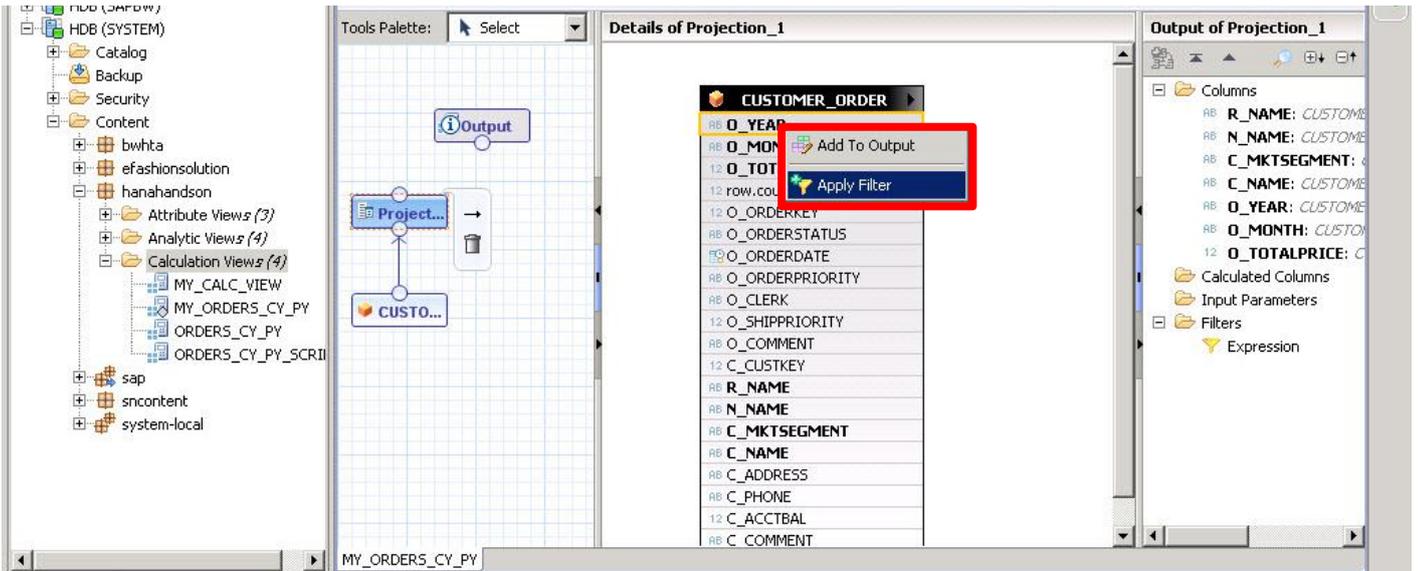
Link the Customer_Order to the Projection. Mouse over Customer_order and click and hold Create Connection and drag to the Projection component



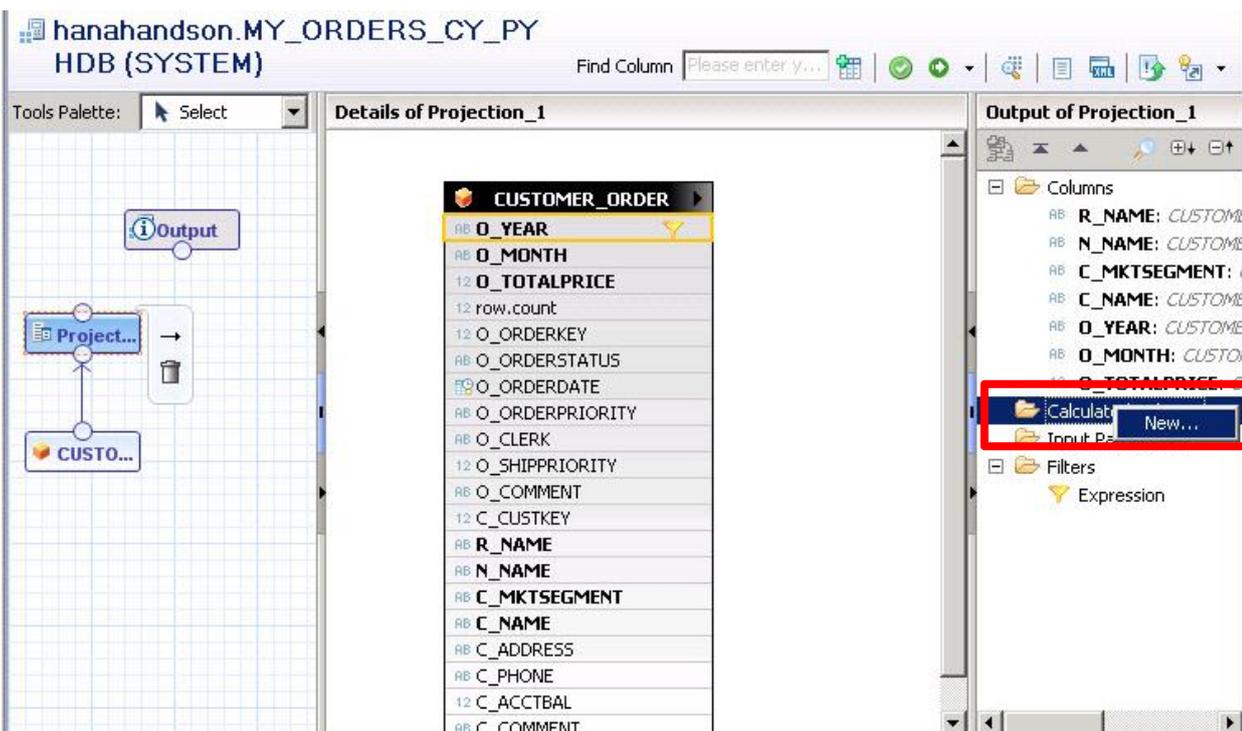
Click the Projection Component and add the following fields to the output: R_Name, N_Name, C_Mktsegment, C_Name, O_Year, O_Month, O_Totalprice



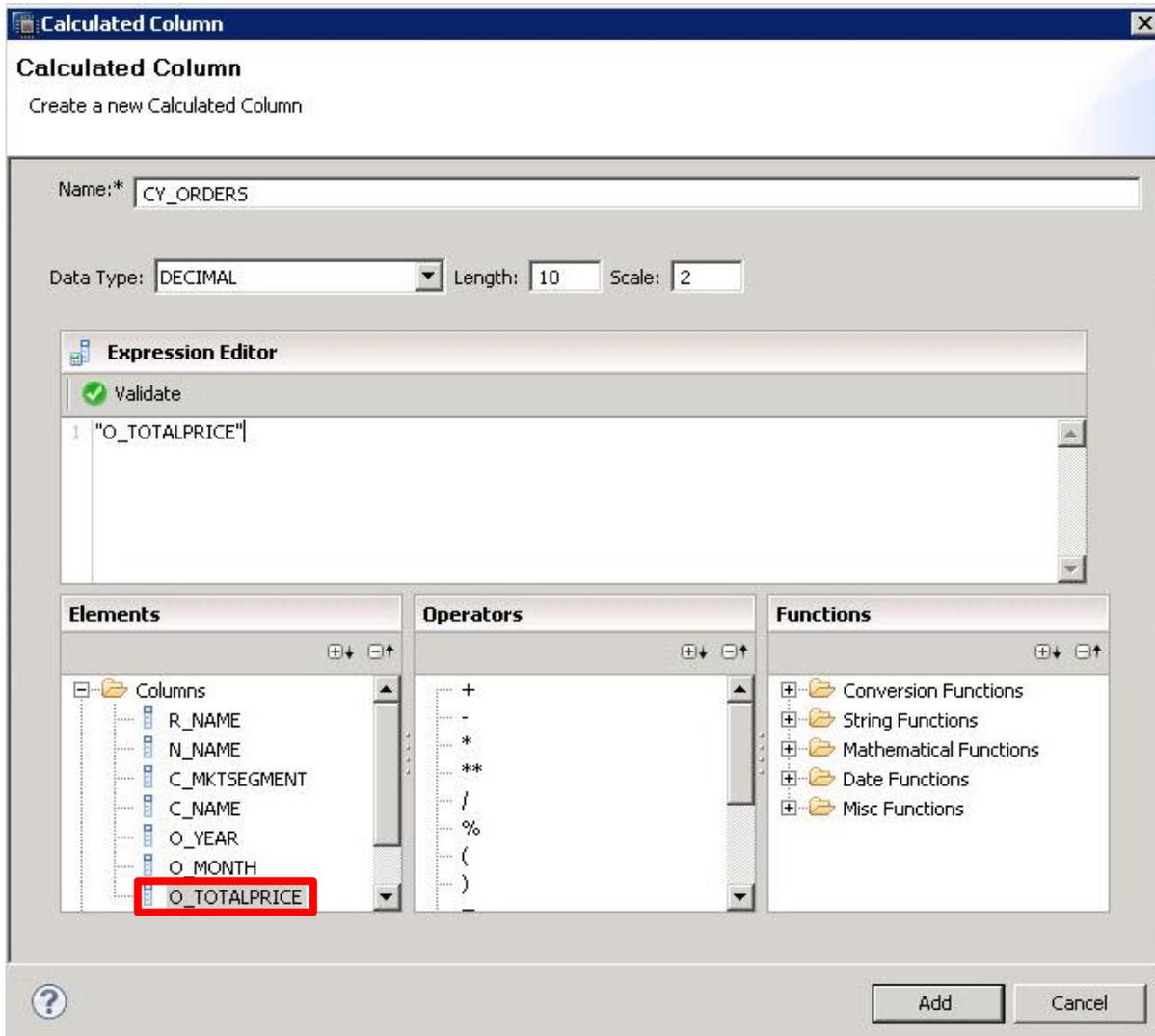
Apply a filter to O_Year so that Year = 1998. Right click on Customer_Order. O_Year → Apply Filter



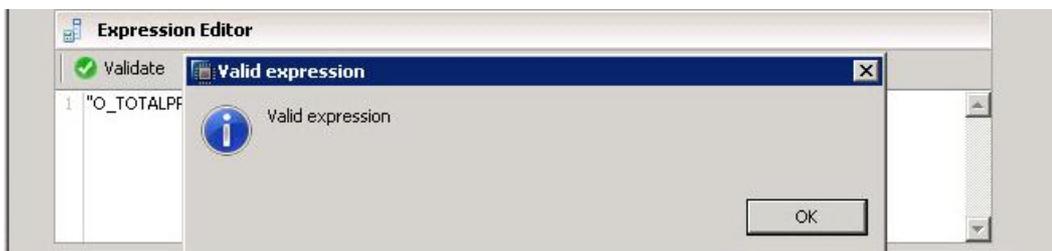
Create a Calculation Column for the Current Year. Right click on Calculation Column => New



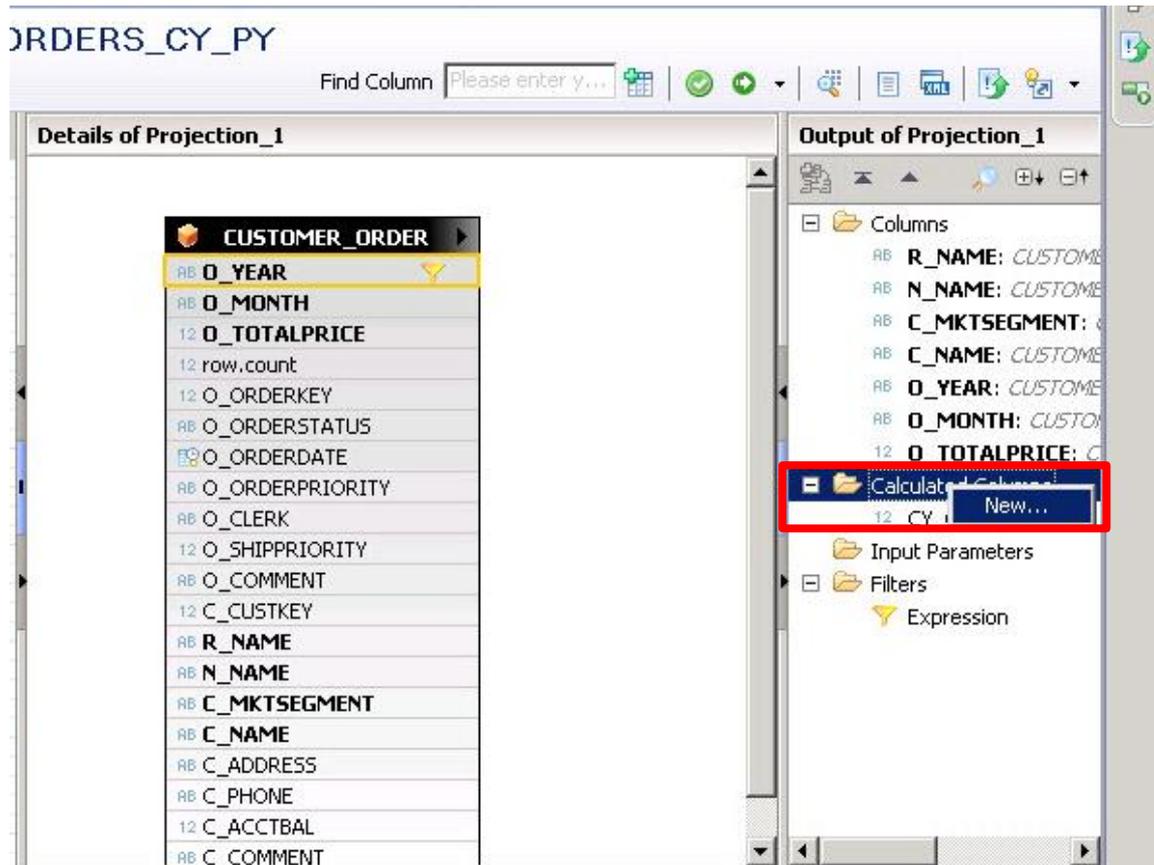
Name it CY_ORDERS, Data Type is Decimal Length 10 Scale 2, add O_TOTALPRICE to the Expression Editor.



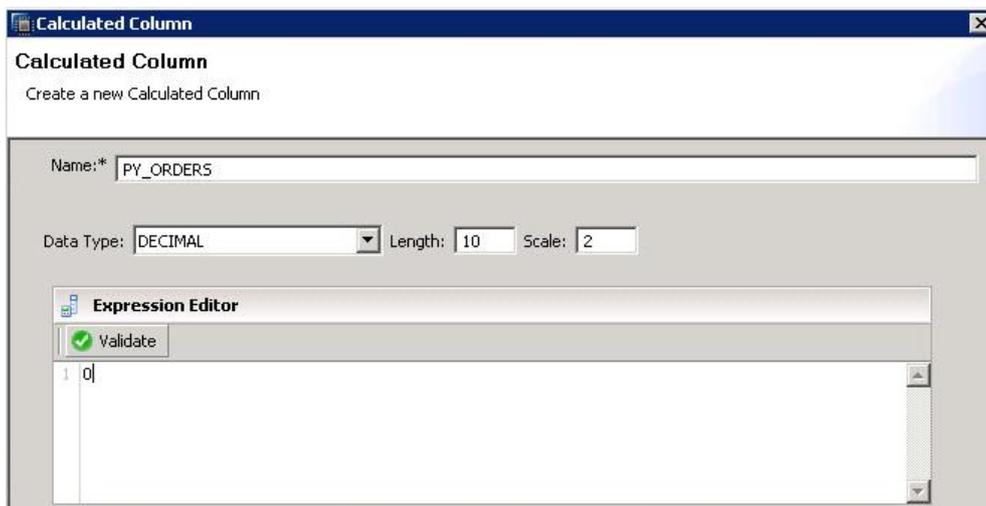
Validate the expression:



Create another Calculated Column for the Prior Year. Right click on Calculation Column → New



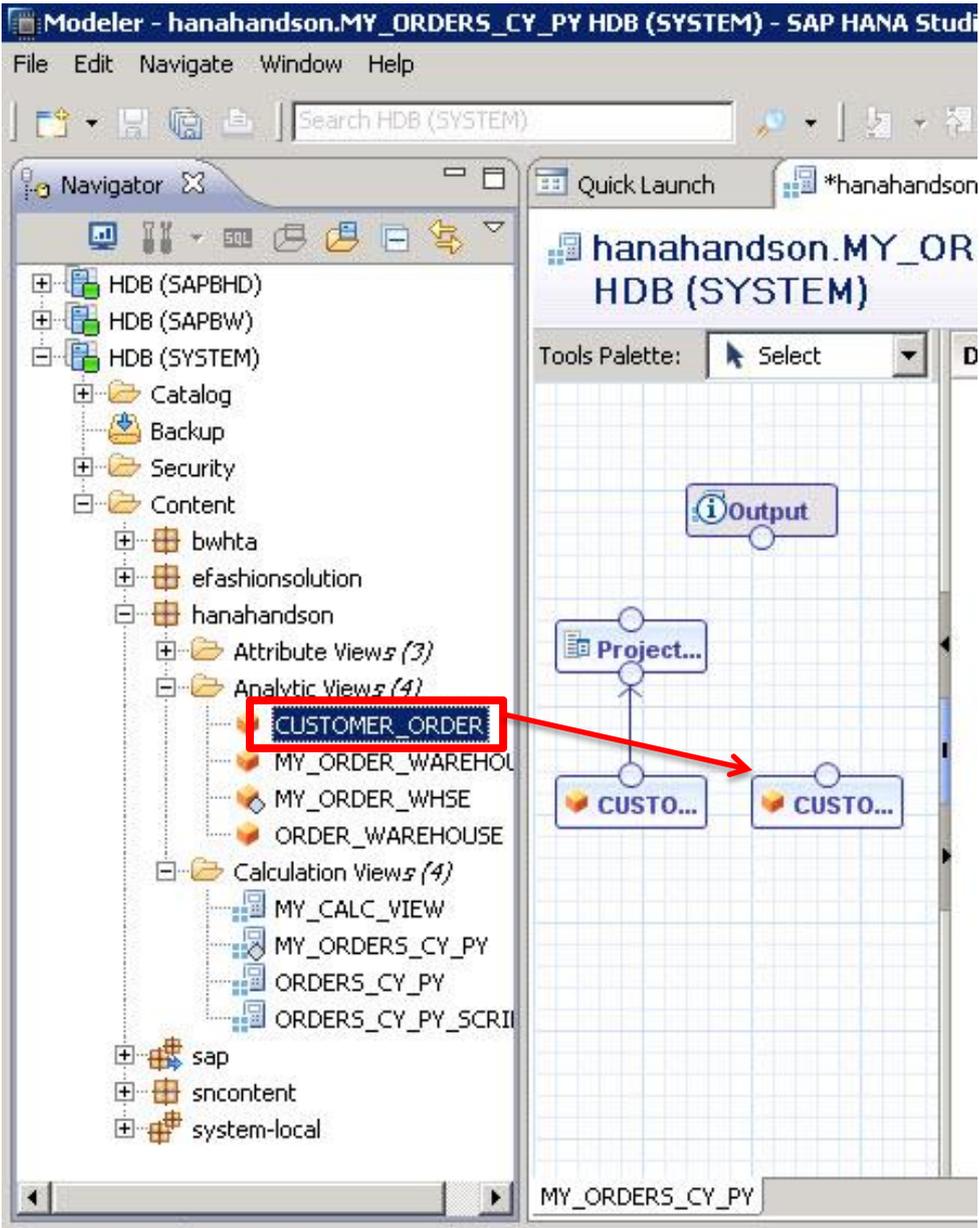
Name it PY_Orders, Decimal Length 10 Scale 2 and add the number 0 to the Expression Editor. (We need this field so that the union will function properly. All queries in a union need the same number of fields)



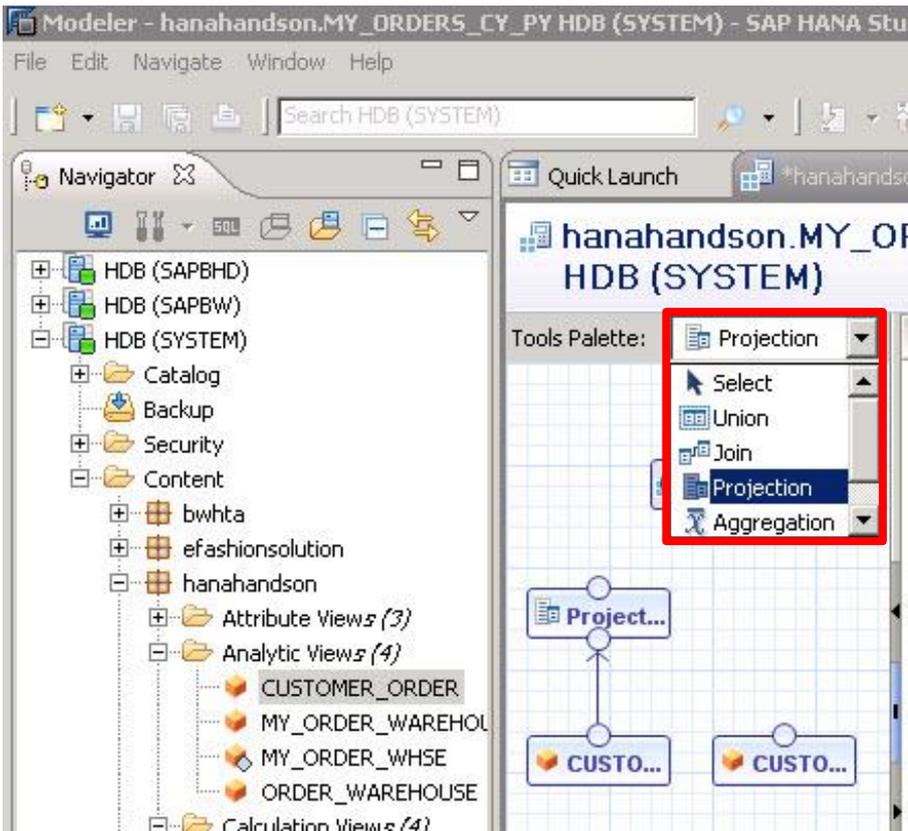
Validate expression and OK to close the editor:



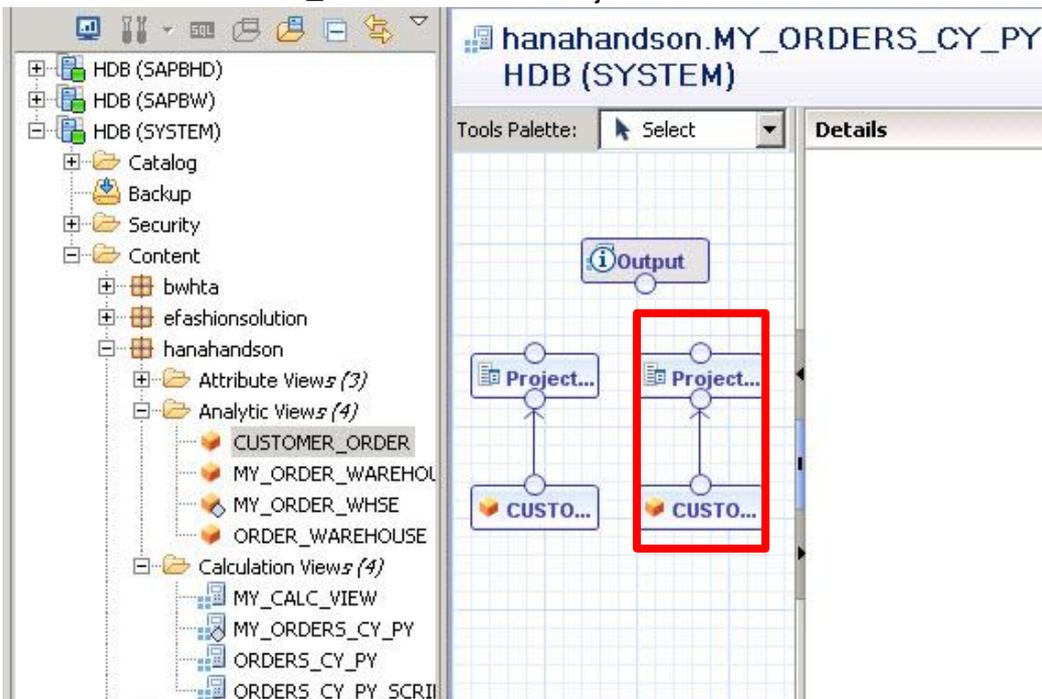
Now we will add the Prior Year to the Calculation View. Expand Analytic Views on the Navigator and drag Customer_Order onto the pallet.



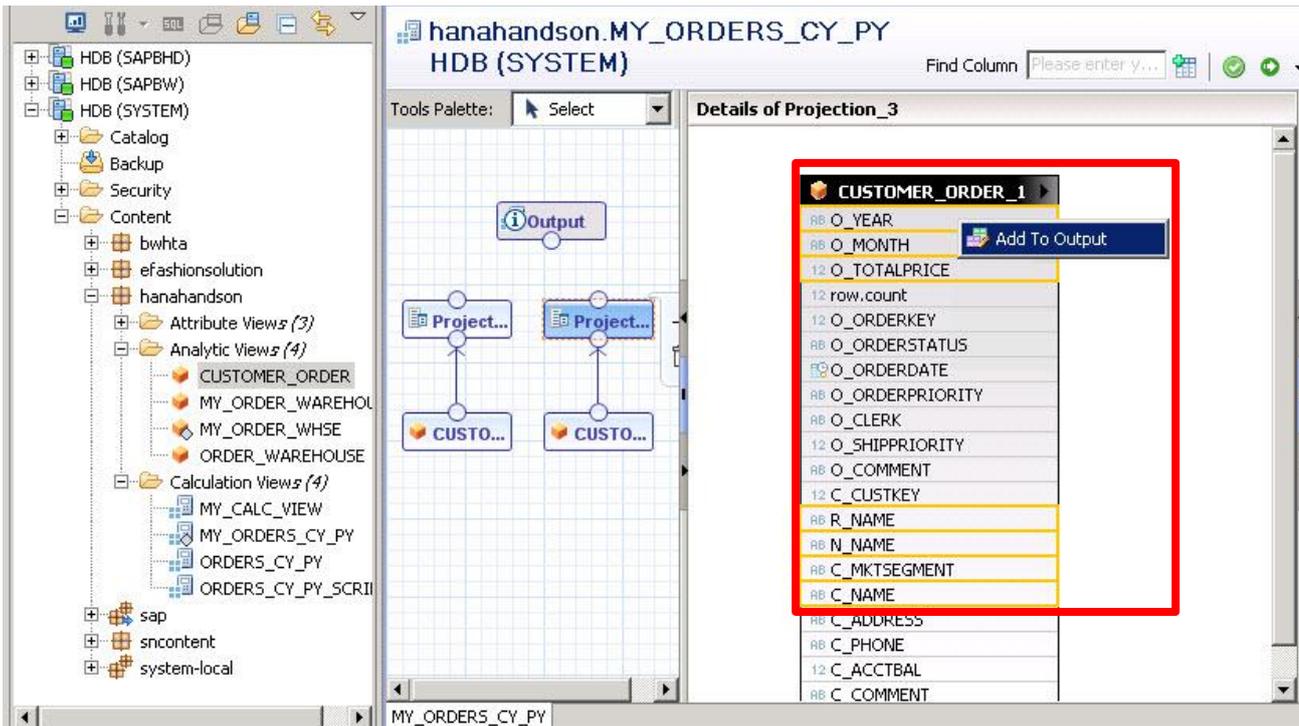
Add a Projection Component:



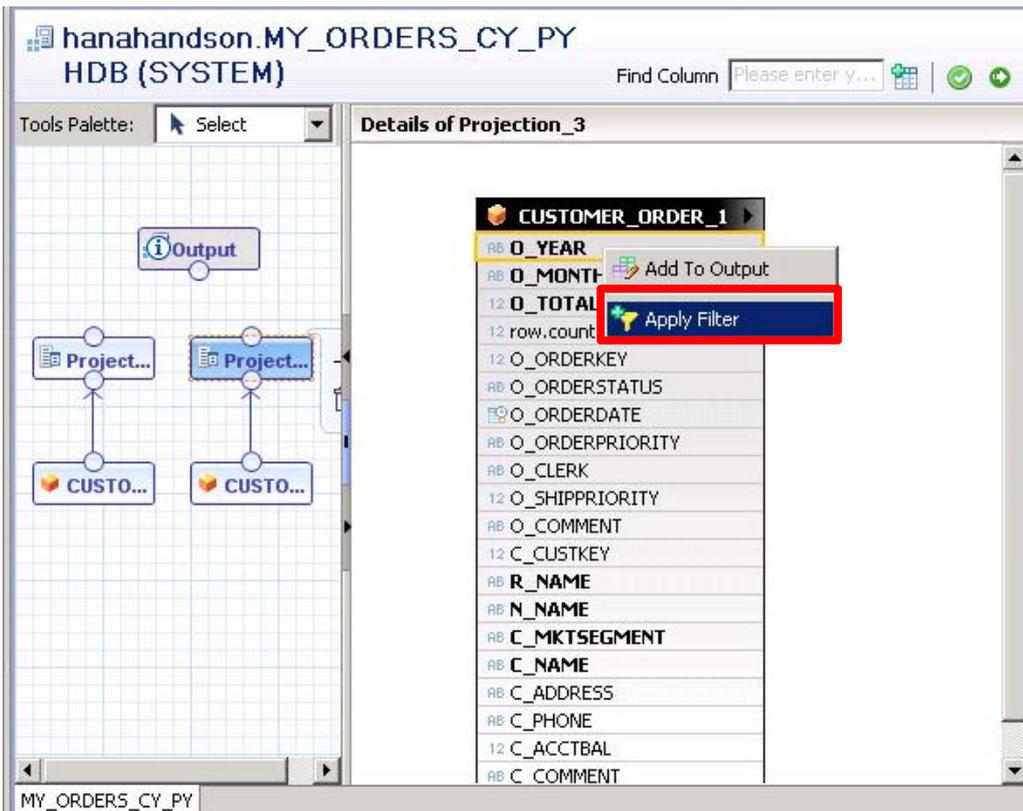
Link the new Customer_Order to the new Projection.

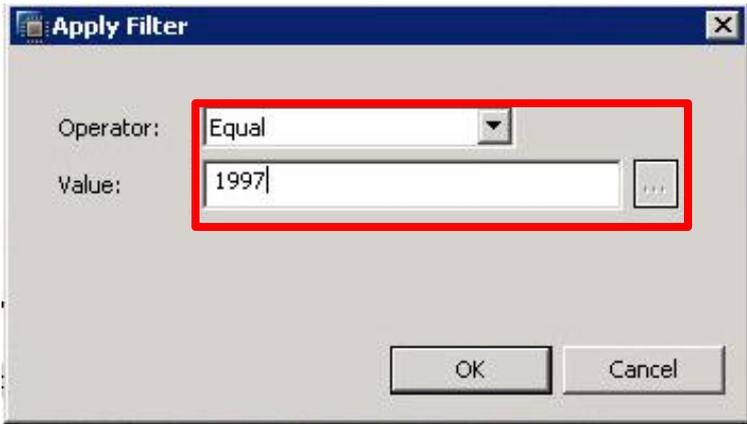


Select the new Projection Component and add the following fields to the output: R_Name, N_Name, C_Mktsegment, C_Name, O_Year, O_Month, O_Totalprice:

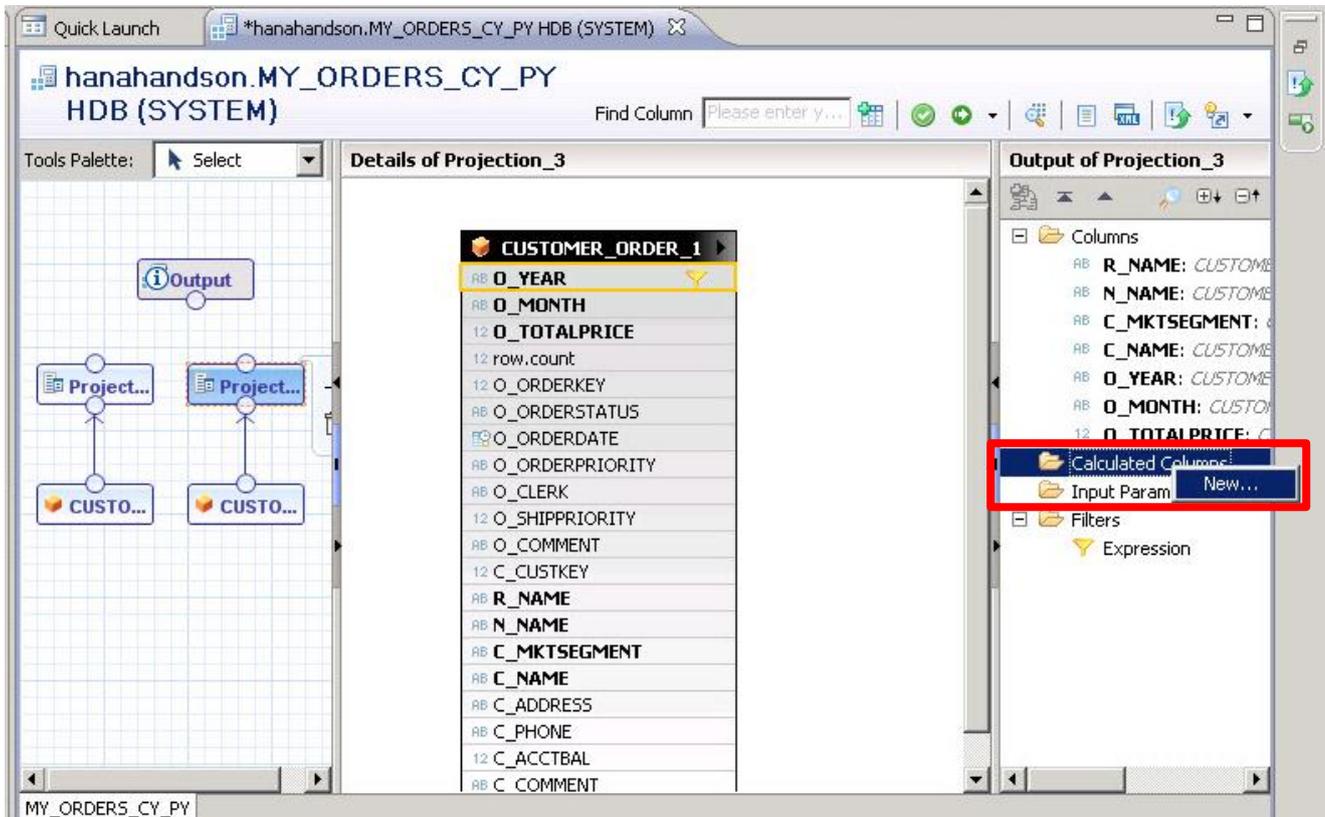


Apply a filter to O_Year so that Year = 1997. Right click on Customer_Order.O_Year → Apply Filter:

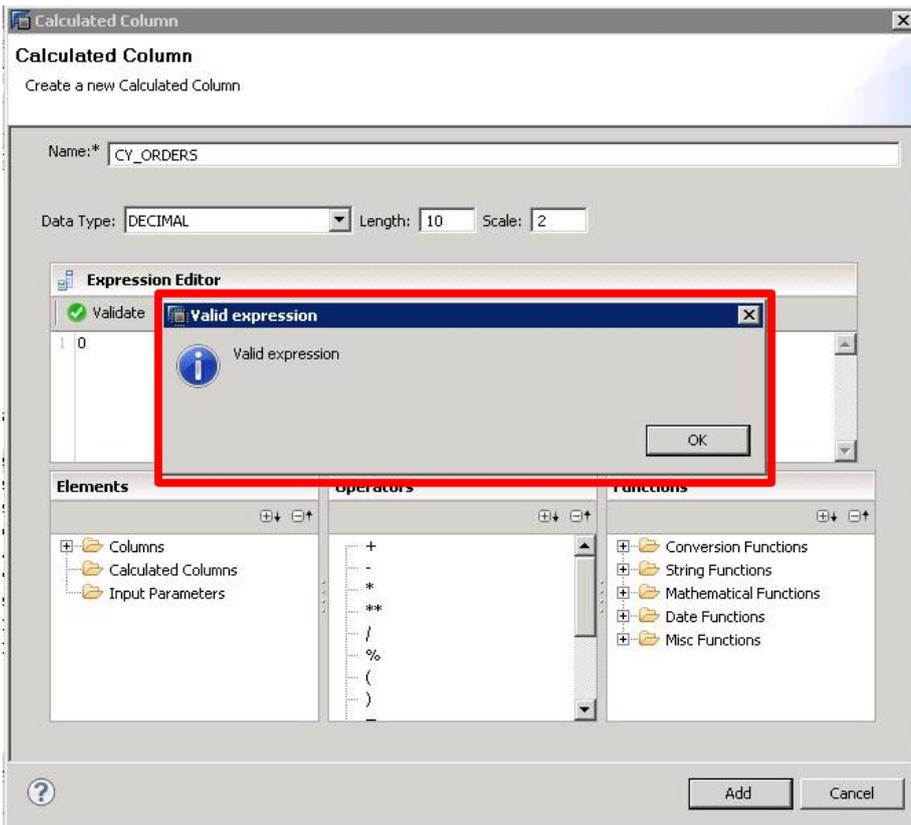




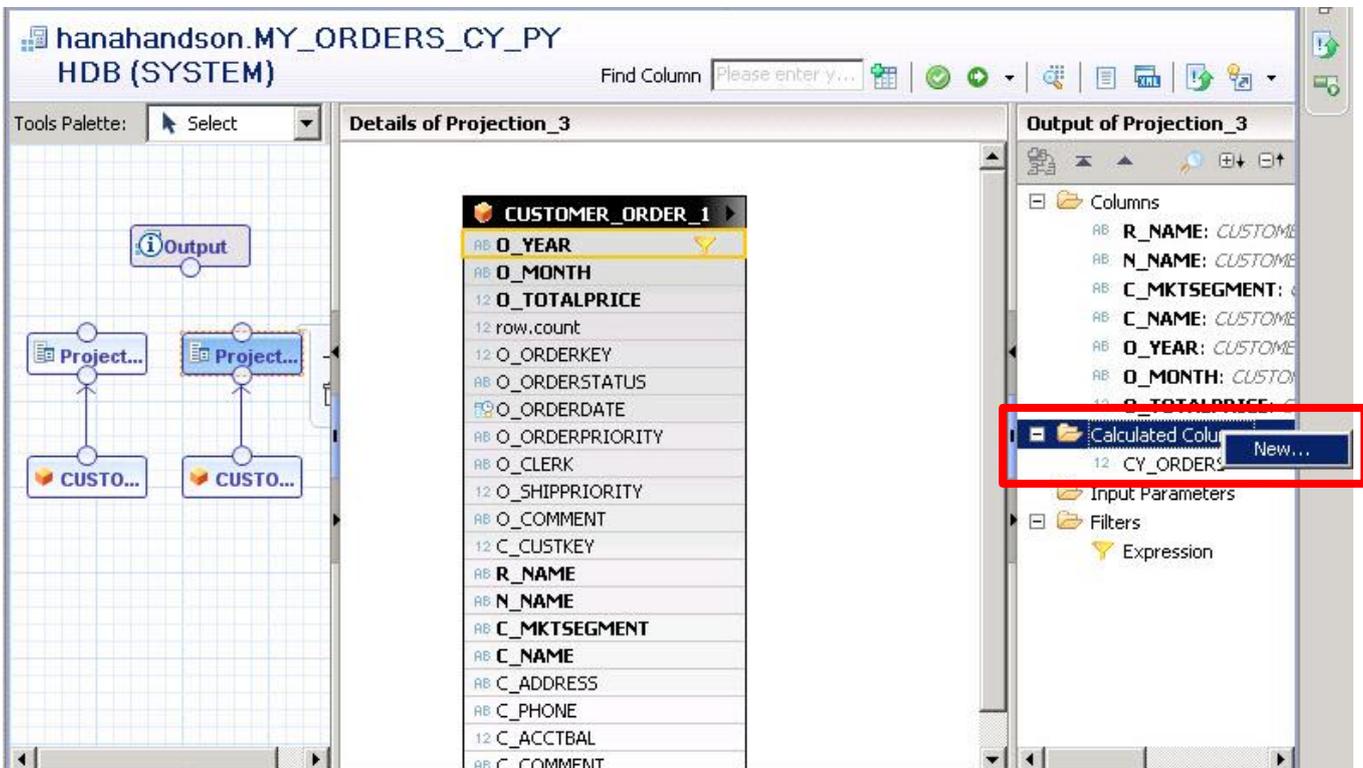
Create a Calculation Column for the Current Year. Right click on Calculation Column → New



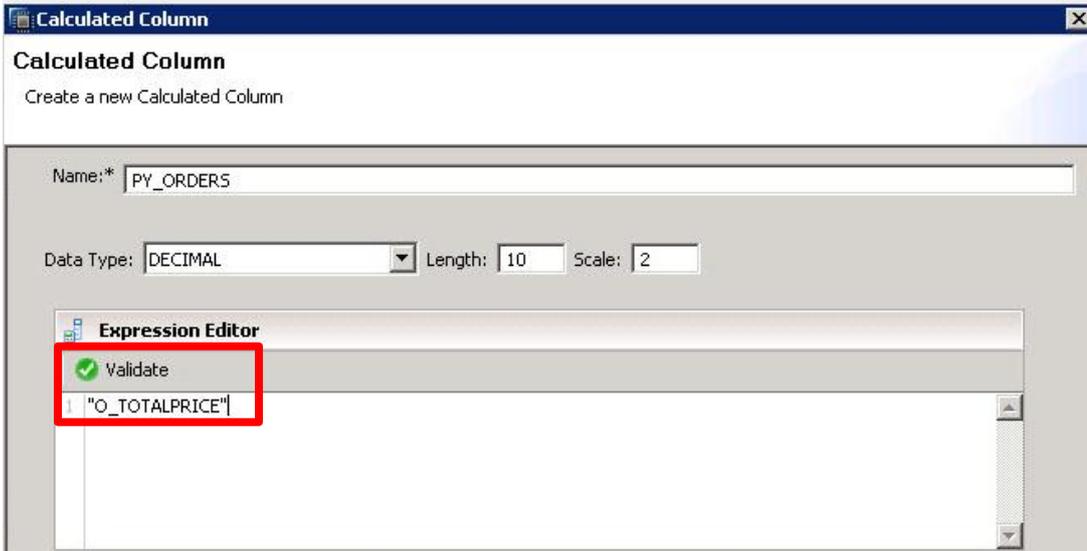
Name it CY_Orders, Data Type Decimal Length 10 Scale 2 and add the number 0 to the Expression Editor.



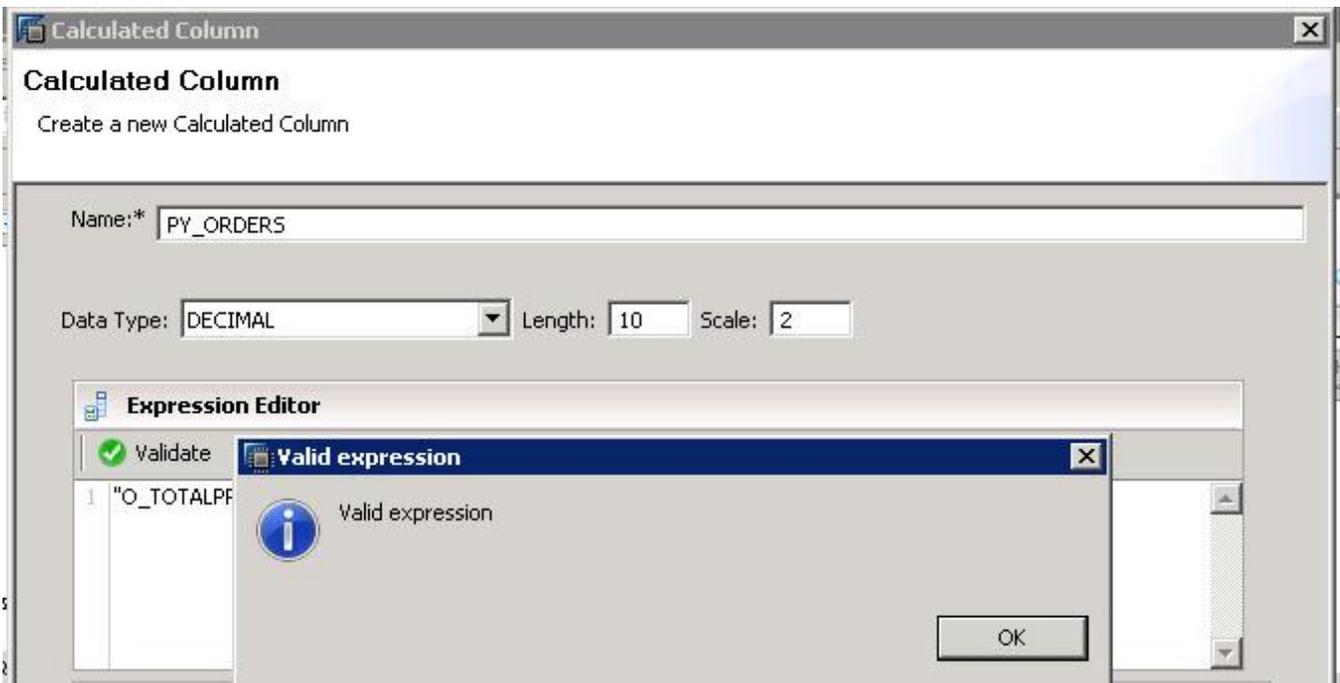
Create another Calculated Column for the Prior Year. Right click on Calculation Column → New



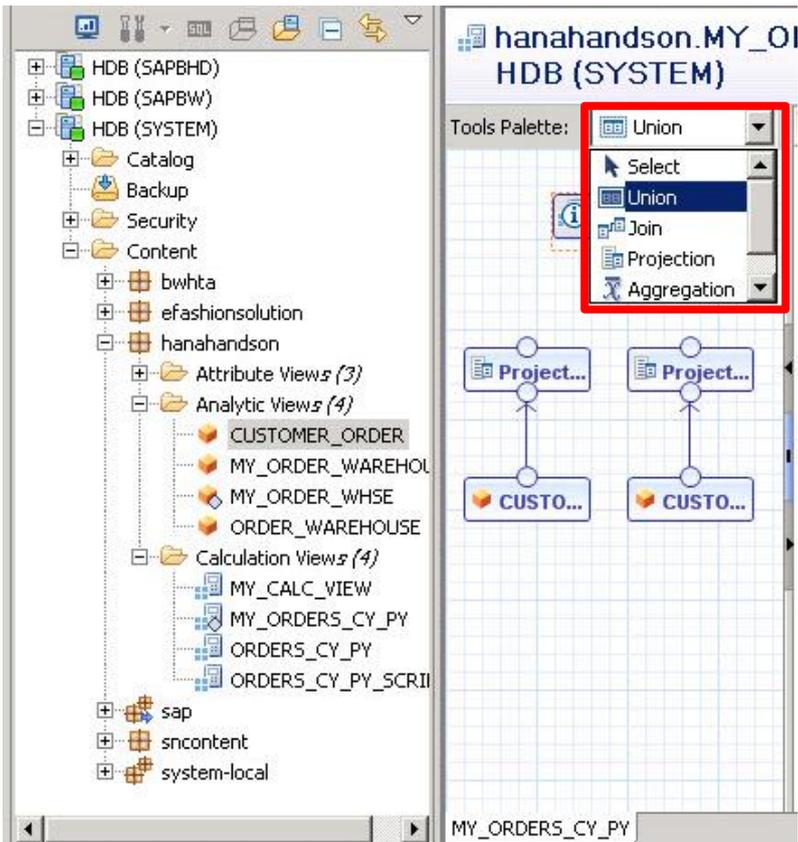
Name it PY_Orders, Data Type Decimal Length 10 Scale 2, and add O_Totalprice to the Expression Editor:



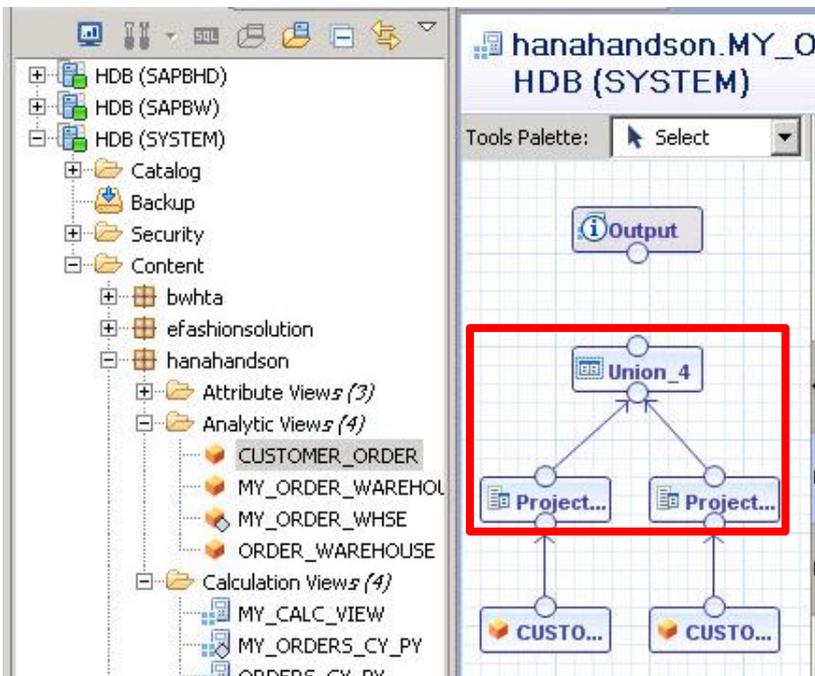
Validate the expression:



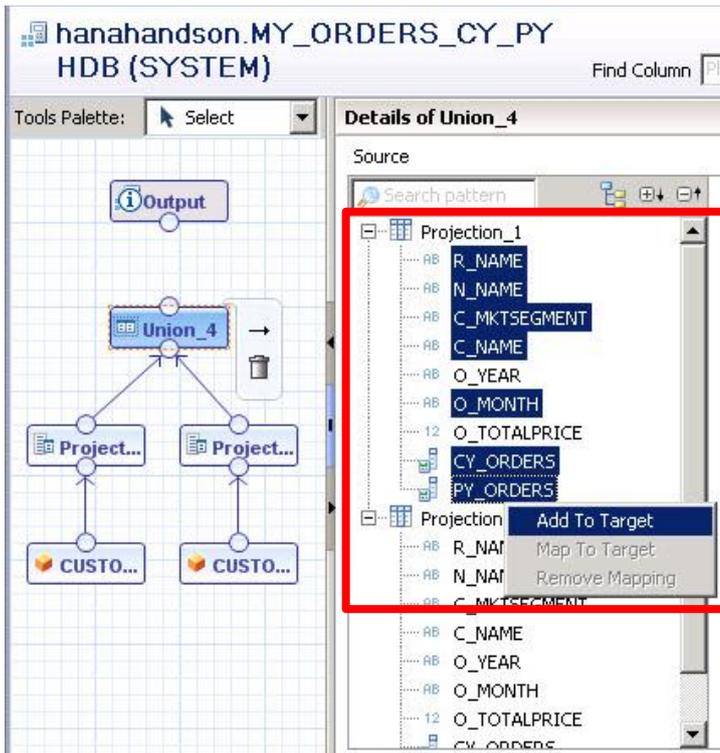
Now we will union the two Projections by adding a Union Component:



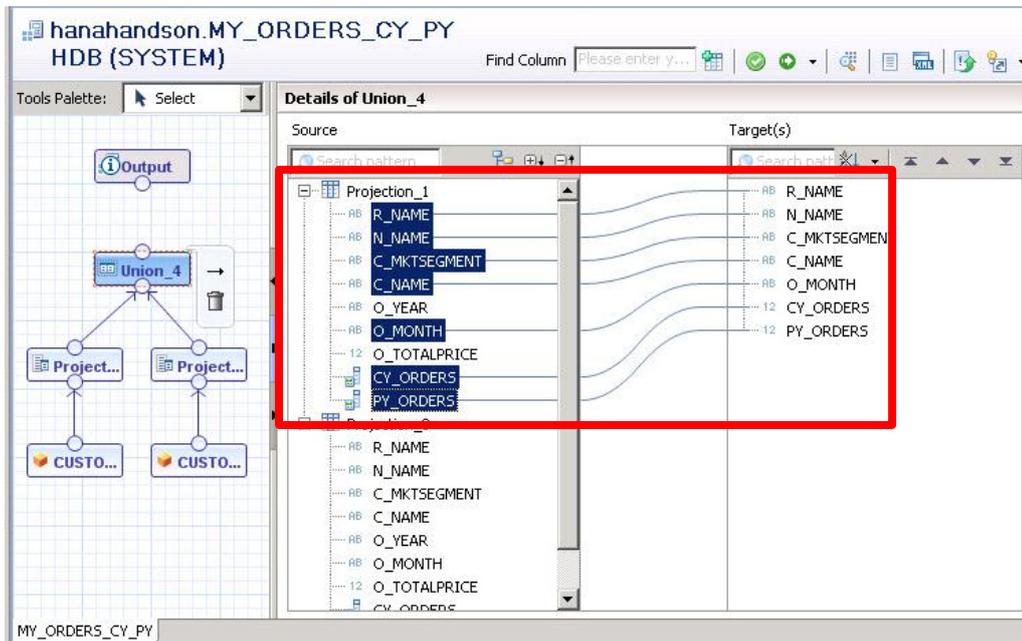
Connect the links from both Projections to the Union.



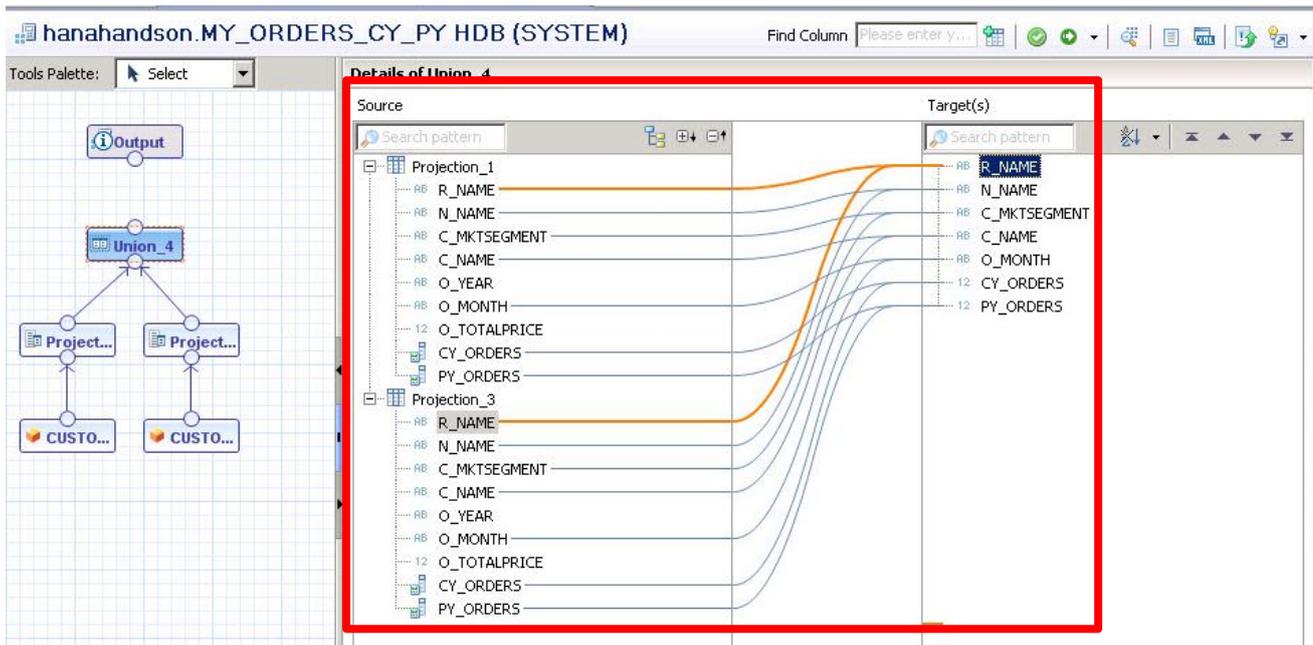
Select the Union Component and add all fields with the exception to O_Year and O_Totalprice from the Projection 1.



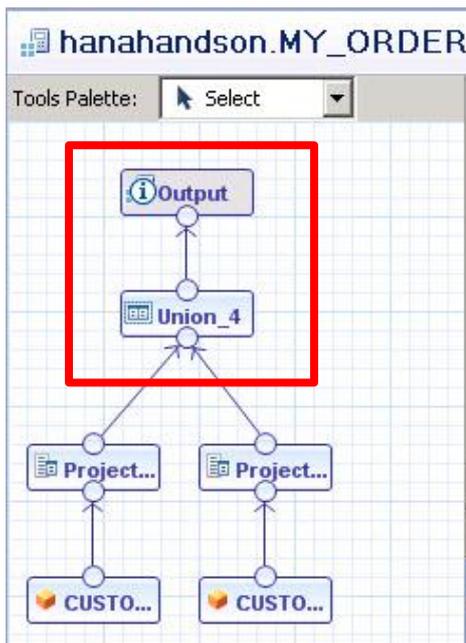
You should then see the below:



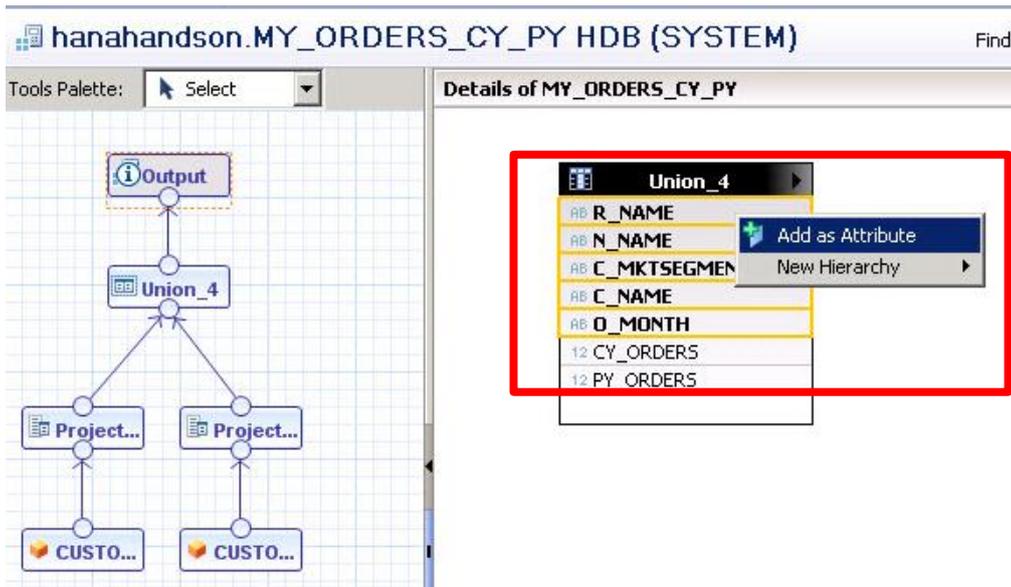
Map the corresponding fields from Projection 2 by dragging them to the target fields from projection 1:



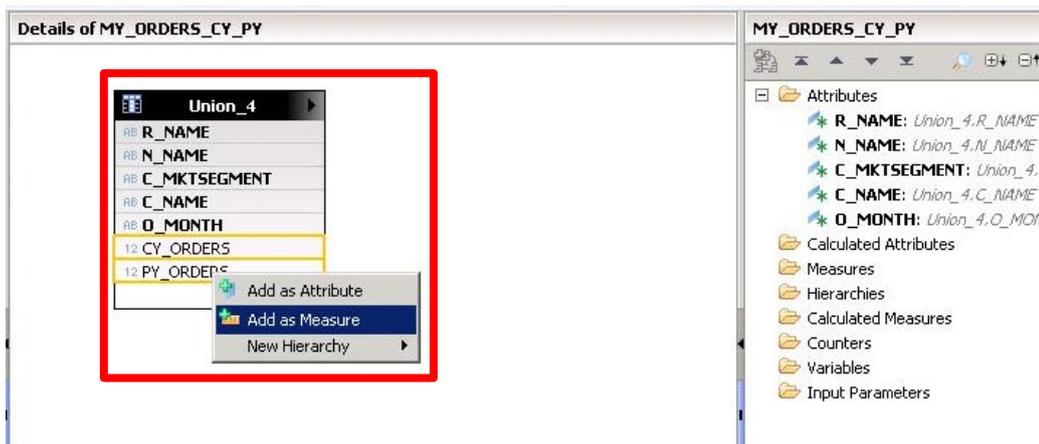
Link the Union Component with the Output Component



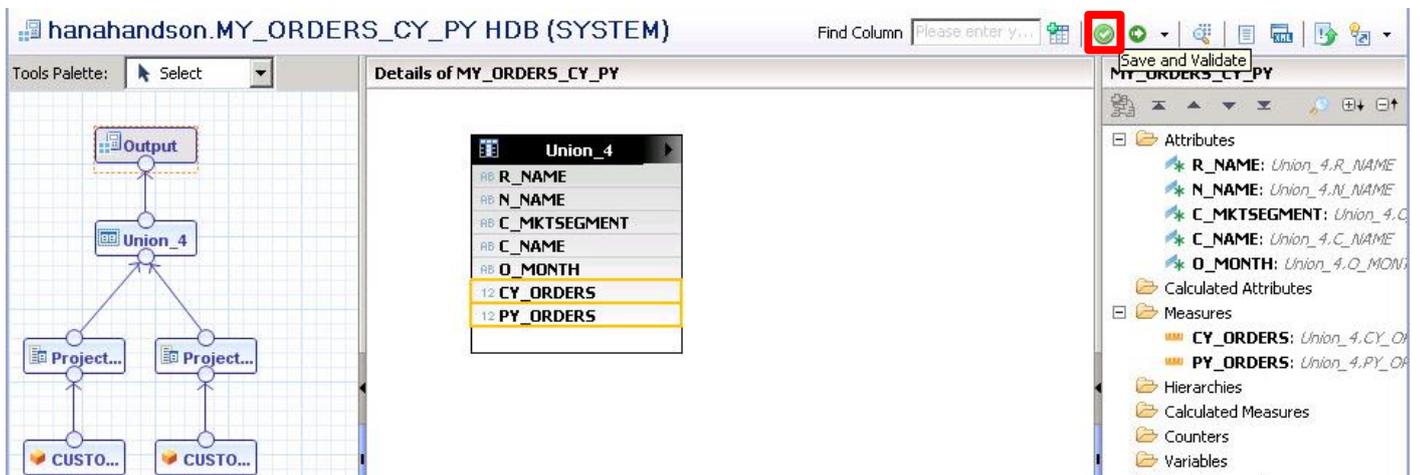
Click on Output and add the Attributes: R_Name, N_Name, C_Mktsegment, C_Name, O_Month:



Add the Calculation Columns as the Measures. CY_Orders, PY_Orders:



Save and Validate the Calculation View:



Job Type	System	User	Submitted At	Status
Model Validation	HDB	SYSTEM	Sun Feb 03 01:06:45 GMT 2013	Completed successfully

Save and Activate the Calculation View:

The screenshot shows the SAP Studio interface for a calculation view named 'MY_ORDERS_CY_PY'. The 'Save and Activate' button in the top right toolbar is highlighted with a red square. The main workspace displays a diagram of the calculation view structure, including an 'Output' node, a 'Union_4' node, and two 'Project...' nodes. The 'Details of MY_ORDERS_CY_PY' pane shows the 'Union_4' node with columns: R_NAME, N_NAME, C_MKTSEGMENT, C_NAME, O_MONTH, CY_ORDERS, and PY_ORDERS. The right-hand pane shows the 'MY_ORDERS_CY_PY' object with its attributes and calculated attributes.

Job Type	System	User	Submitted At	Status
Activation	HDB	SYSTEM	Sun Feb 03 01:06:52 GMT 2013	Completed successfully
Model Validation	HDB	SYSTEM	Sun Feb 03 01:06:45 GMT 2013	Completed successfully

Test the Calculation View by seeing the Data Preview – Right click the view and select Data Preview:

The screenshot shows the SAP Studio interface with a context menu open over the calculation view 'MY_ORDERS_CY_PY'. The 'Data Preview' option is highlighted with a red rectangle. The context menu includes options: Open, Validate, Activate, Redeploy, Revert To Active, Data Preview, Where-Used, History, Delete, Auto Documentation, Copy, and Refactor. The background shows the calculation view diagram and the Job Log table.

Output from the Data Preview:

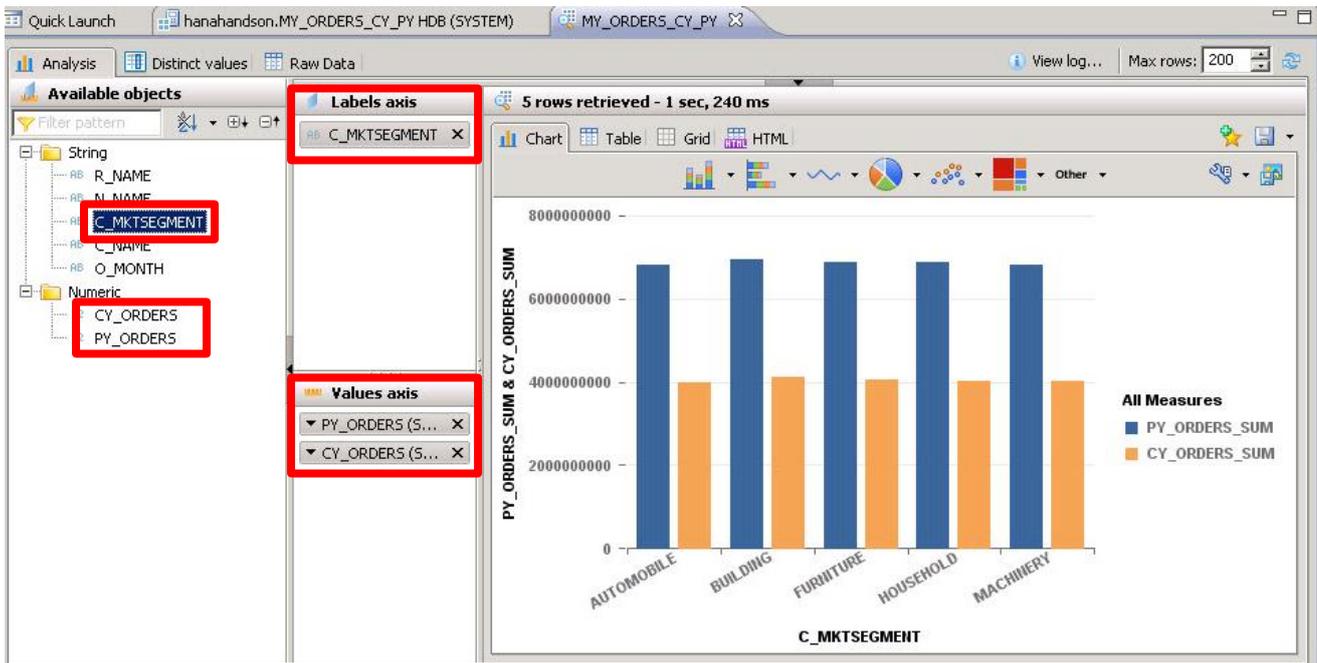
Quick Launch | hanahandson.MY_ORDERS_CY_PY HDB (SYSTEM) | MY_ORDERS_CY_PY

Analysis | Distinct values | Raw Data | View log... | Max rows: 200

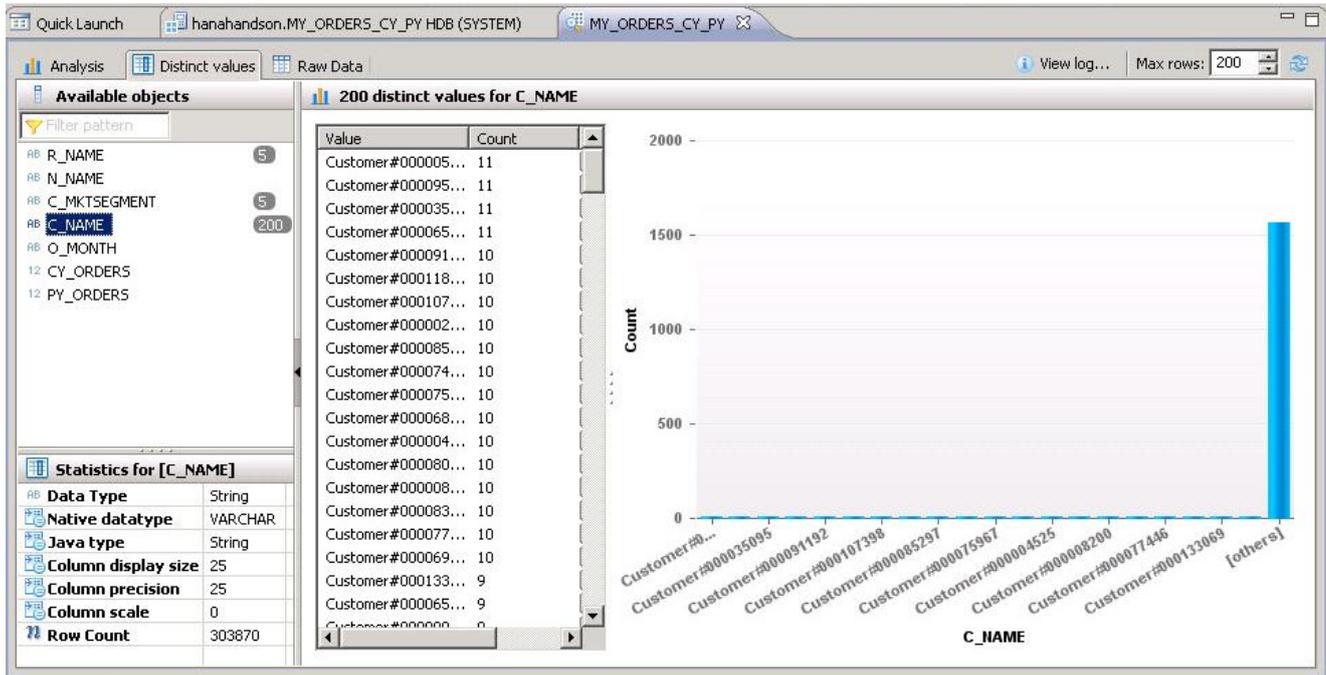
Filter pattern | 200 rows retrieved - 1 sec, 645 ms | Add filter

R#	R_NAME	N#	N_NAME	C#	C_MKTSEGMENT	C#	C_NAME	O#	O_MONTH	CY_ORDERS	PY_ORDERS
AMERICA	CANADA				AUTOMOBILE	Customer#00...		03		0	90246.29
EUROPE	RUSSIA				FURNITURE	Customer#00...		01		0	225313
MIDDLE EAST	EGYPT				FURNITURE	Customer#00...		07		0	164964.02
ASIA	CHINA				AUTOMOBILE	Customer#00...		08		0	244073.77
ASIA	INDONESIA				HOUSEHOLD	Customer#00...		10		0	233397.47
EUROPE	GERMANY				HOUSEHOLD	Customer#00...		11		0	83310.23
AFRICA	MOZAMBIQUE				BUILDING	Customer#00...		11		0	82950.71
EUROPE	FRANCE				HOUSEHOLD	Customer#00...		05		371561.8	0
MIDDLE EAST	SAUDI ARABIA				FURNITURE	Customer#00...		06		78194.43	77529.98
EUROPE	RUSSIA				BUILDING	Customer#00...		06		74523.59	0
EUROPE	FRANCE				MACHINERY	Customer#00...		03		366480.06	0
AMERICA	BRAZIL				AUTOMOBILE	Customer#00...		02		135889	0
EUROPE	RUSSIA				AUTOMOBILE	Customer#00...		08		0	233558.06
ASIA	VIETNAM				BUILDING	Customer#00...		05		0	58538.44
AMERICA	BRAZIL				AUTOMOBILE	Customer#00...		07		0	78079.77
ASIA	CHINA				MACHINERY	Customer#00...		02		23864.7	0
AMERICA	ARGENTINA				BUILDING	Customer#00...		04		53981.76	0
MIDDLE EAST	EGYPT				AUTOMOBILE	Customer#00...		10		0	312888.39
EUROPE	FRANCE				BUILDING	Customer#00...		06		0	72347.07
AFRICA	ALGERIA				HOUSEHOLD	Customer#00...		03		184280.26	182430.92
ASIA	JAPAN				HOUSEHOLD	Customer#00...		04		0	48089.12
EUROPE	ROMANIA				FURNITURE	Customer#00...		11		0	168342.63

Now let's take a different look at the data preview. Click on the Analysis tab and add C_MKTSEGMENT to the Label axis. With the same process add CY_ORDERS and PY_ORDERS to the Value axis:



Now with Data values tab you can see further details on the fields in the table.



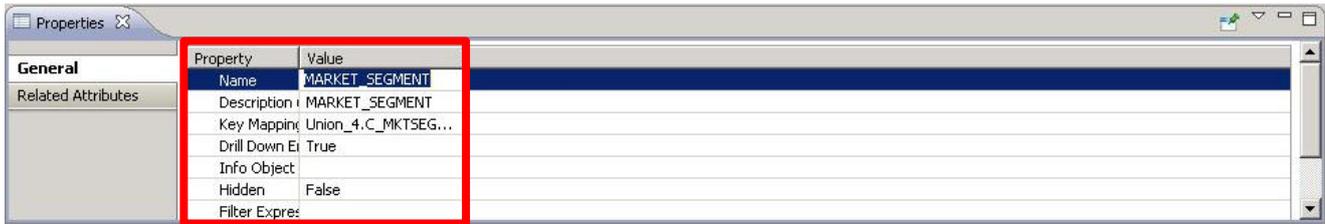
As you can see from the above some object names could be clearer and more meaningful so let's change one.

Ensure you have the properties tab opened and then click on the C_MKTSEGMENT object on the right from the attributes folder:

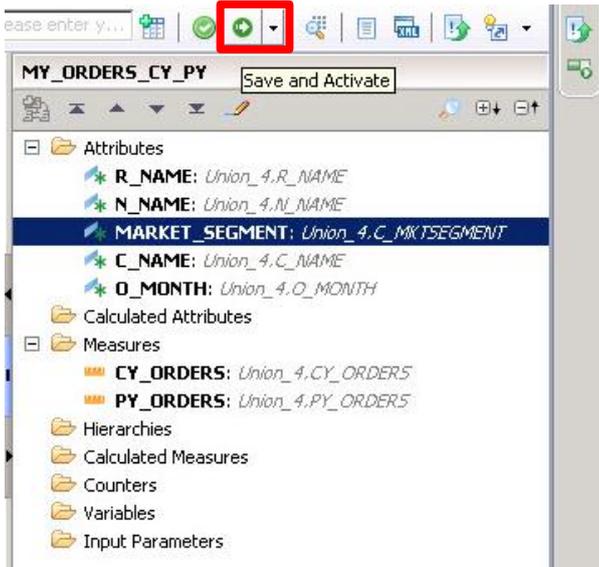
The screenshot shows the 'Details of MY_ORDERS_CY_PY' view. The 'C_MKTSEGMENT' attribute is highlighted in red in the 'Attributes' folder. The 'Properties' tab for this attribute is also highlighted in red and contains the following information:

Property	Value
Name	C_MKTSEGMENT
Description	
Key Mapping	Union_4.C_MKTSEG...
Drill Down Enabled	True
Info Object	
Hidden	False
Filter Expression	

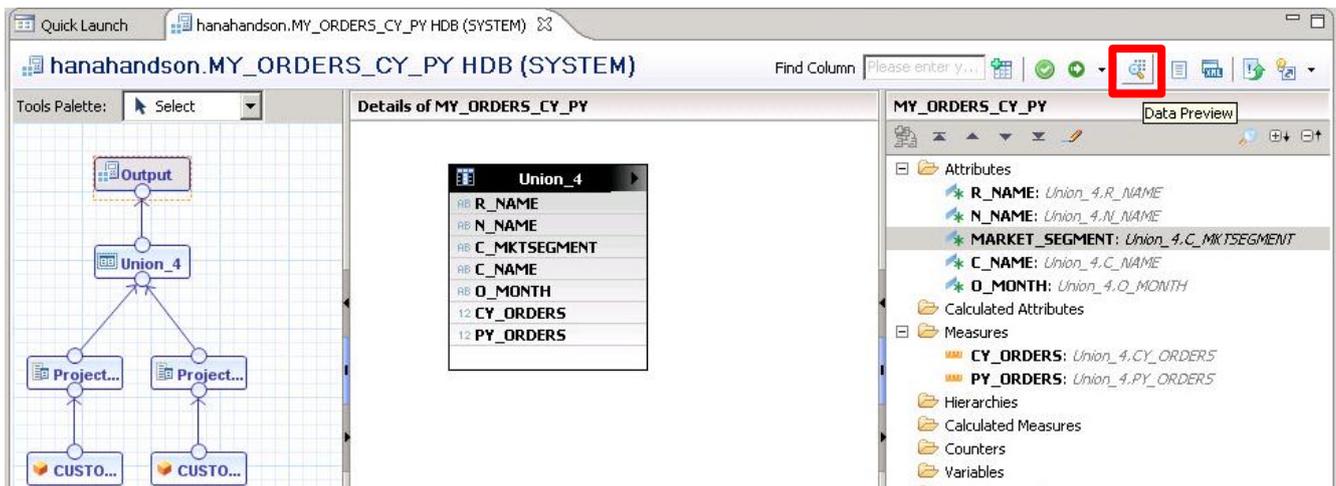
Change the Name and Description to MARKET_SEGMENT and click Save.



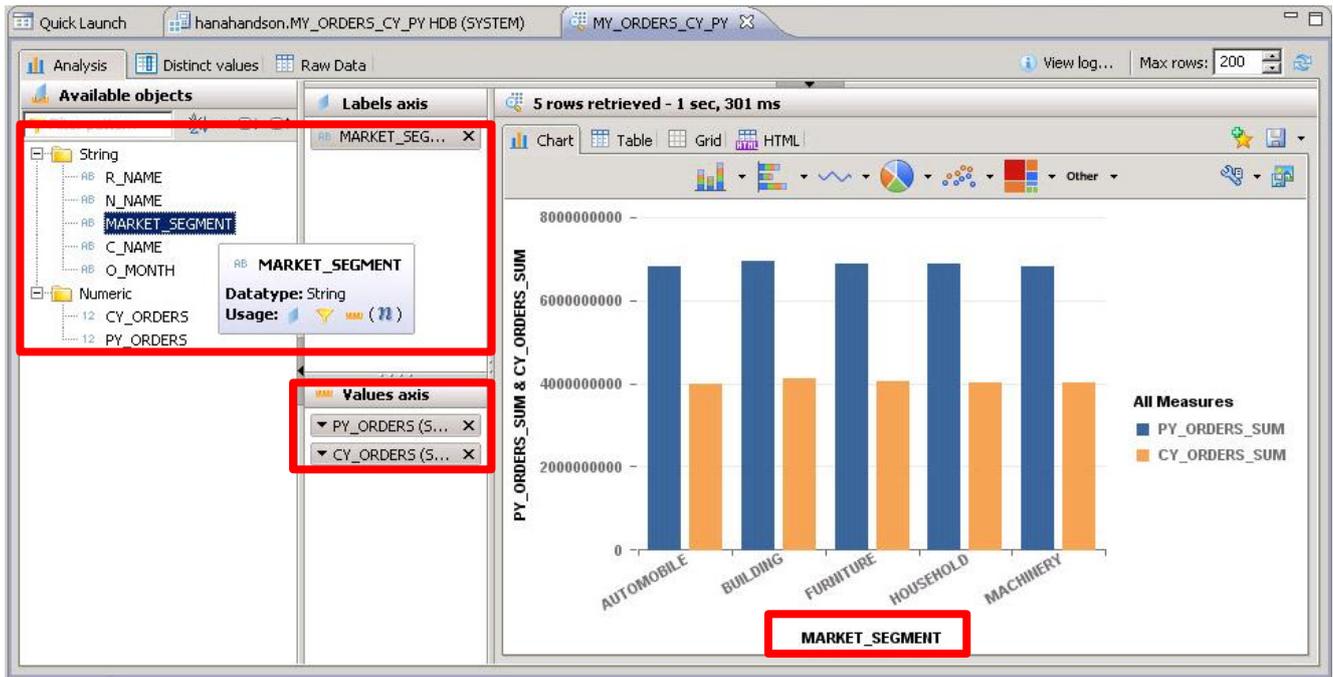
The modified name should now appear in the attributed folder - Validate and Activate it again:



Go to the Data Preview and the Analysis tab. You will now see the change you made to C_MKTSEGMENT:



We can now see the change is clearly displayed in the new data preview:





Reporting from SAP HANA with SAP Lumira

SAP Lumira Overview

SAP Lumira is the latest innovation in the SAP BusinessObjects Explorer solution family. It is a desktop-based visualization and data manipulation solution that allows business analysts to acquire data from a variety of corporate and personal data sources and manipulate without scripting. SAP Lumira produces beautiful visualizations and allows users to analyze data and quickly discover unique insight that can be easily shared throughout the organization.

Despite not carrying the portfolio brand BusinessObjects, SAP Lumira is part of the SAP BusinessObjects portfolio and extends the value of SAP BusinessObjects Explorer. With the web and mobile experiences of Explorer, business users can quickly explore their data and build their own dashboards. SAP Lumira complements these two experiences while offering Business Analysts the ability to perform more advanced data transformation and analysis.

For business analysts: SAP Lumira is the fastest, most engaging way to find answers from data. With its engaging and easy-to-use interface, SAP Lumira redefines the notion of self-service information allowing users to bring together corporate information, their own personal data and enriched semantics from SAP to create new ways of viewing the business.

For IT: SAP Lumira allows IT to provide analysts a data discovery solution, enabling business users to work with both corporate information and personal data sources to answer the questions they are asked everyday without assistance from IT.

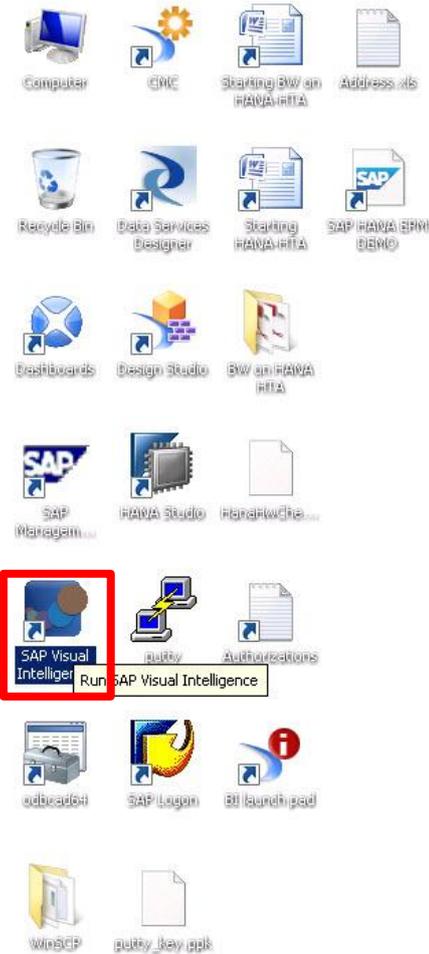
Further, SAP Lumira enables users to perform these functions without having to create predefined query, report, or dashboard and with the ability to manage SAP Lumira as part of their enterprise BI standard. From powerful data discovery to beautiful visual analytics, SAP Lumira is about bringing life to your data, finding the key information for better decisions.

With SAP Lumira you can:

- Consume data the way you process thoughts:
 - fast, interactive, and visual
- Connect, access and visualize data without a single line of code
- Get real-time answers on any volume of data
- View the big picture and drill down to details
- Share your findings instantly with others

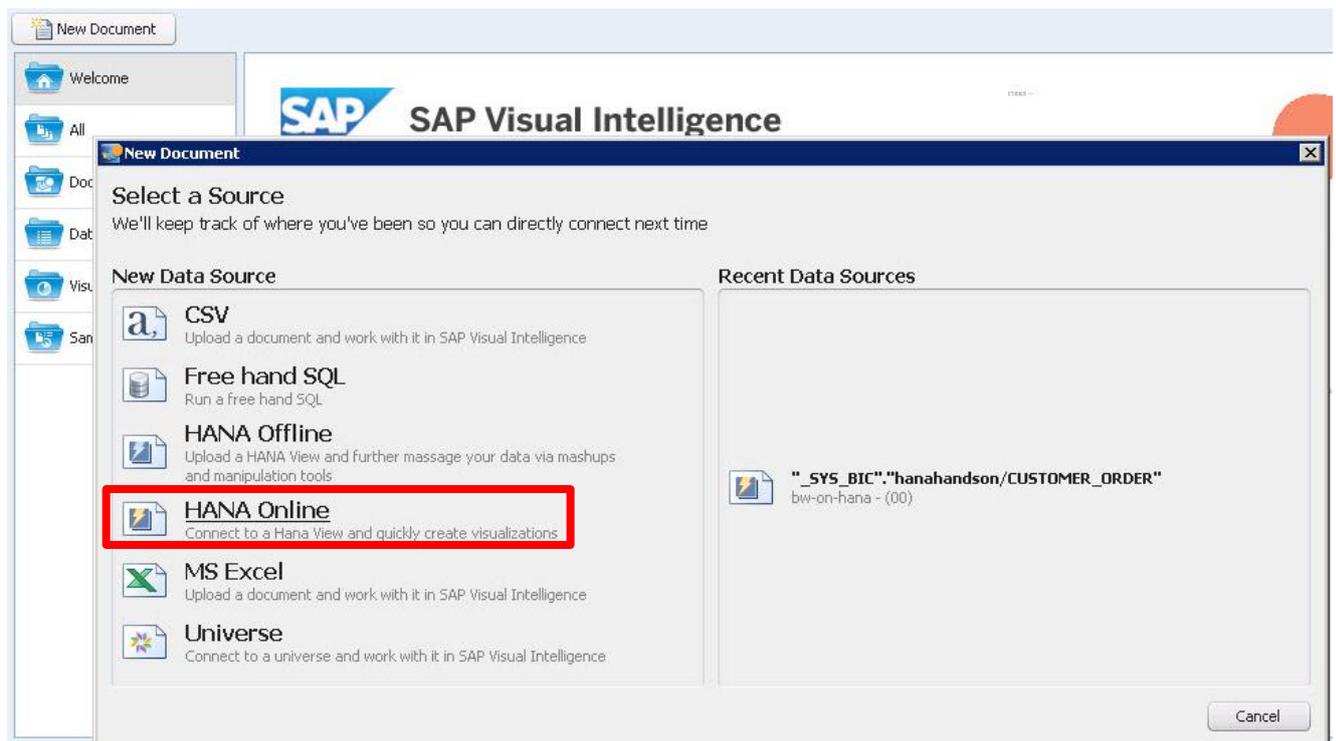
Hands-On Lab:

Open Lumira by double-clicking icon on your desktop:



Look up the SAP HANA server name and server instance in the modeling studio. Open the SAP HANA studio and in your modeling screen and look up the server and instance.

In Visual Intelligence select New Document and choose option “HANA Online”:



Enter credentials to the SAP HANA server:

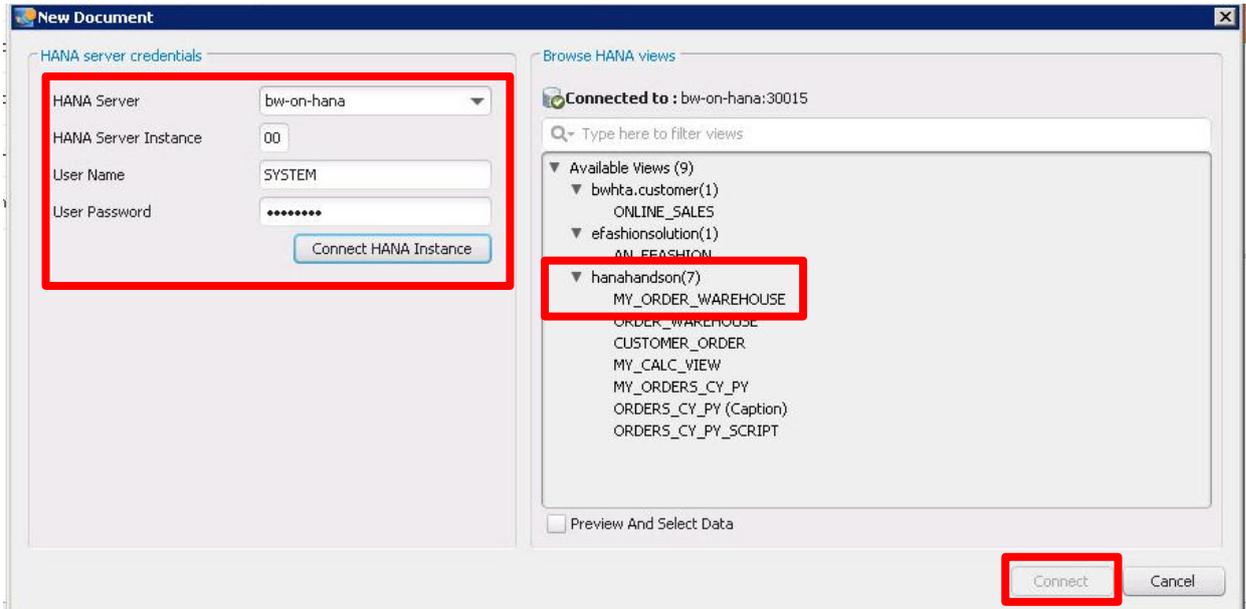
HANA Server: bw-on-hana

HANA Server Instance: 00

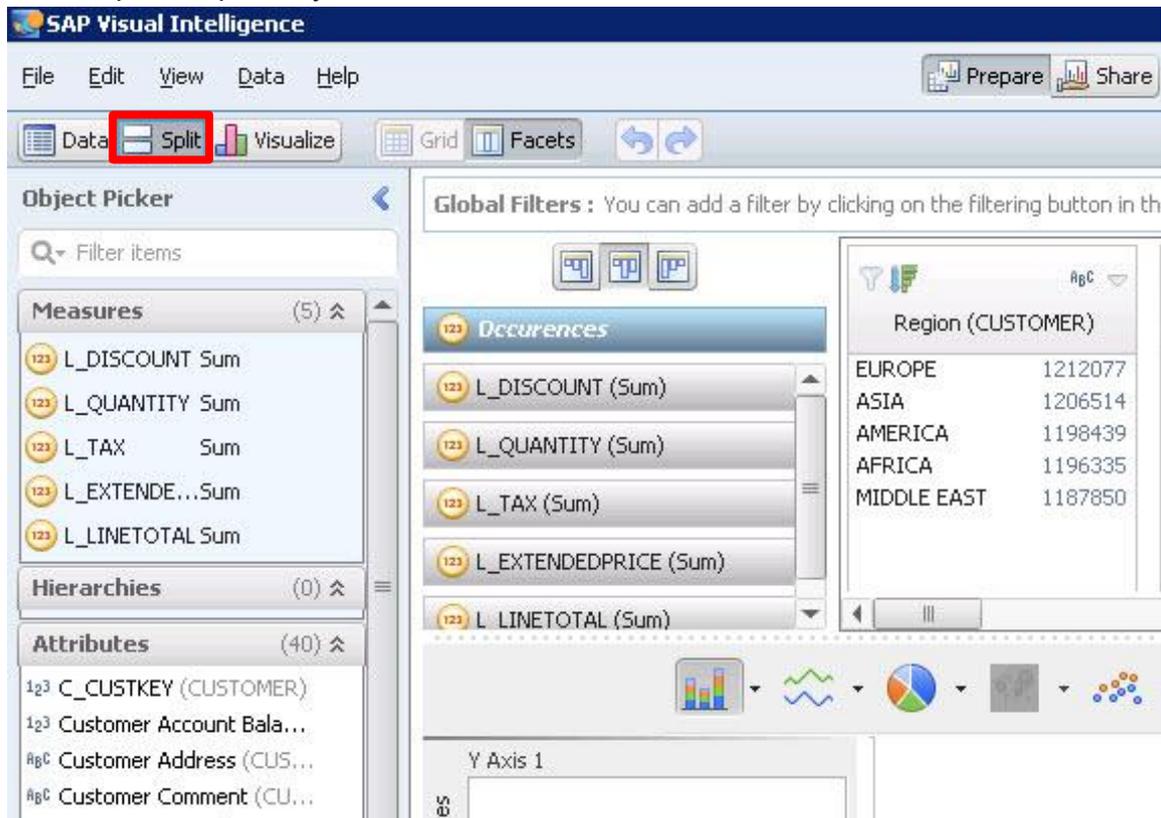
User Name: SYSTEM

User Password: Hana20!2

Then select Analytical View “MY_ORDER_WAREHOUSE” using following log in information, once selected click the connect button.

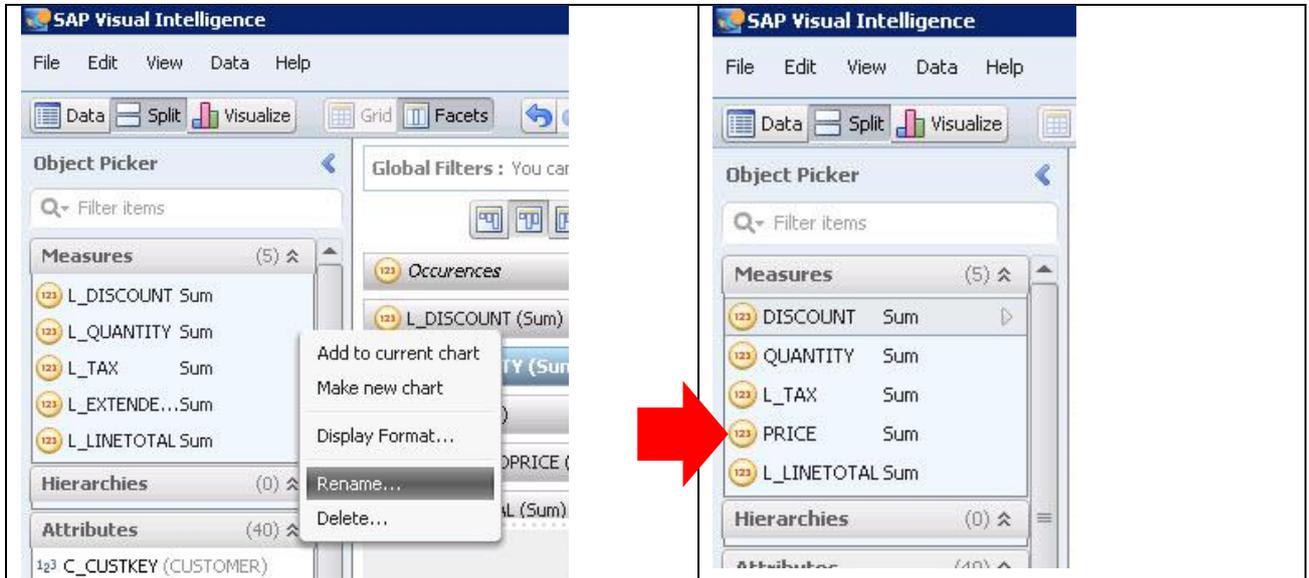


Select option “Split” so you can see all facets as well as the work area:

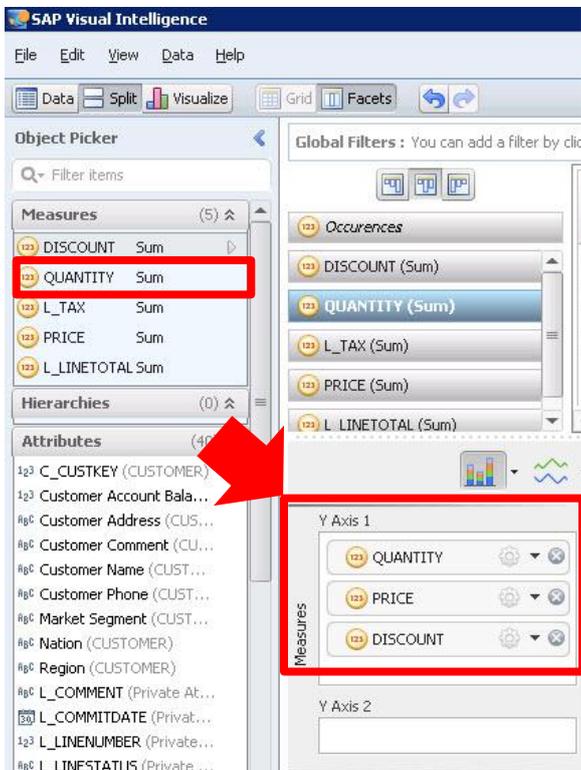


Rename your measures by right-clicking on each measure in the Measures area and choosing “Rename” option.

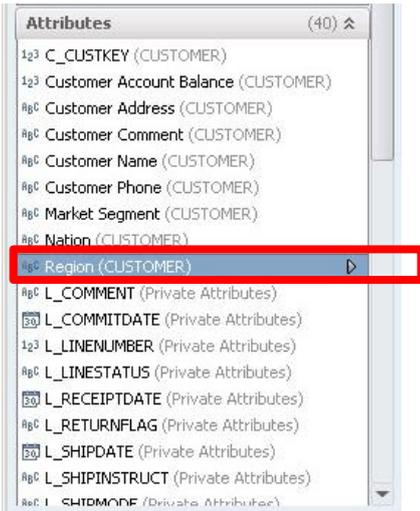
L_QUANTTITY = QUANTITY
 L_EXTENDED PRICE = EXTENDEDPRICE
 L_DISCOUNT = DISCOUNT



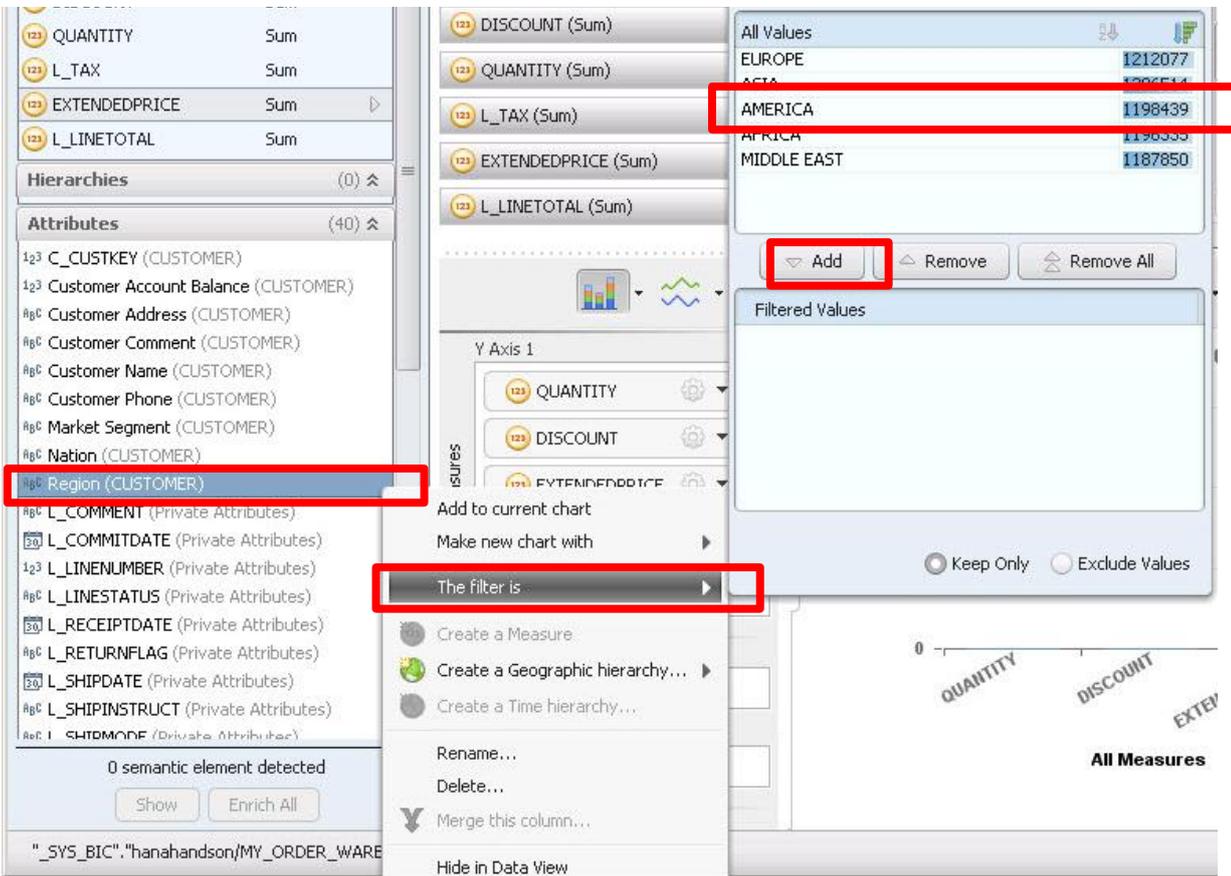
Select QUANTITY, EXTENDEDPRICE, DISCOUNT measures and drag and drop them one by one to Measures - Y Axis 1.



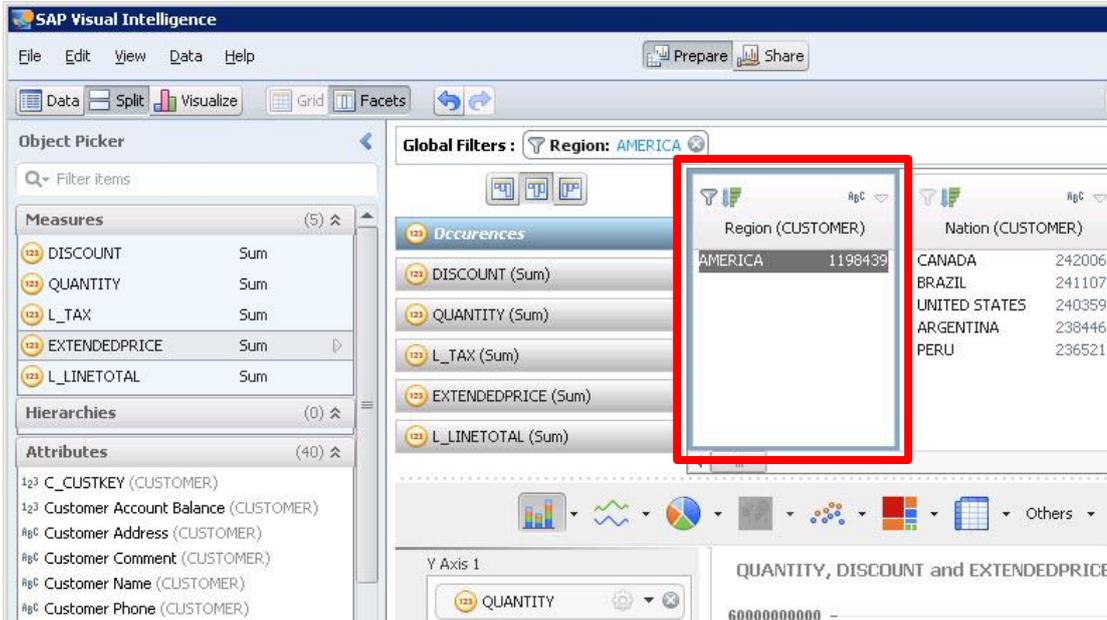
Create a filter for Region. Start by selecting Region in the attributes pane on the left and then select the small arrow to the right of the name:



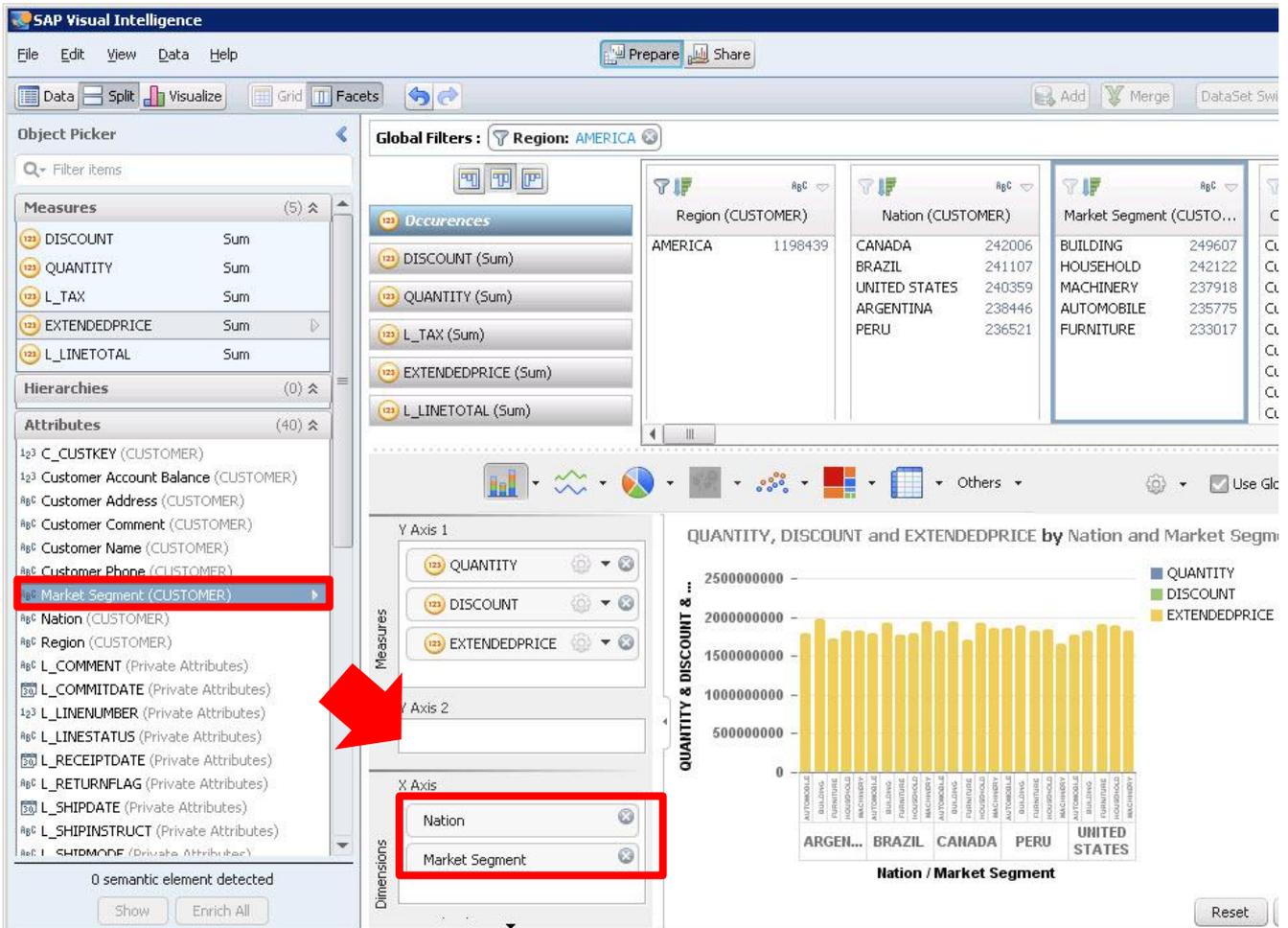
Once the arrow has been selected, navigate to 'The Filter Is' and select AMERICA from the list of values and Add to filterer values:



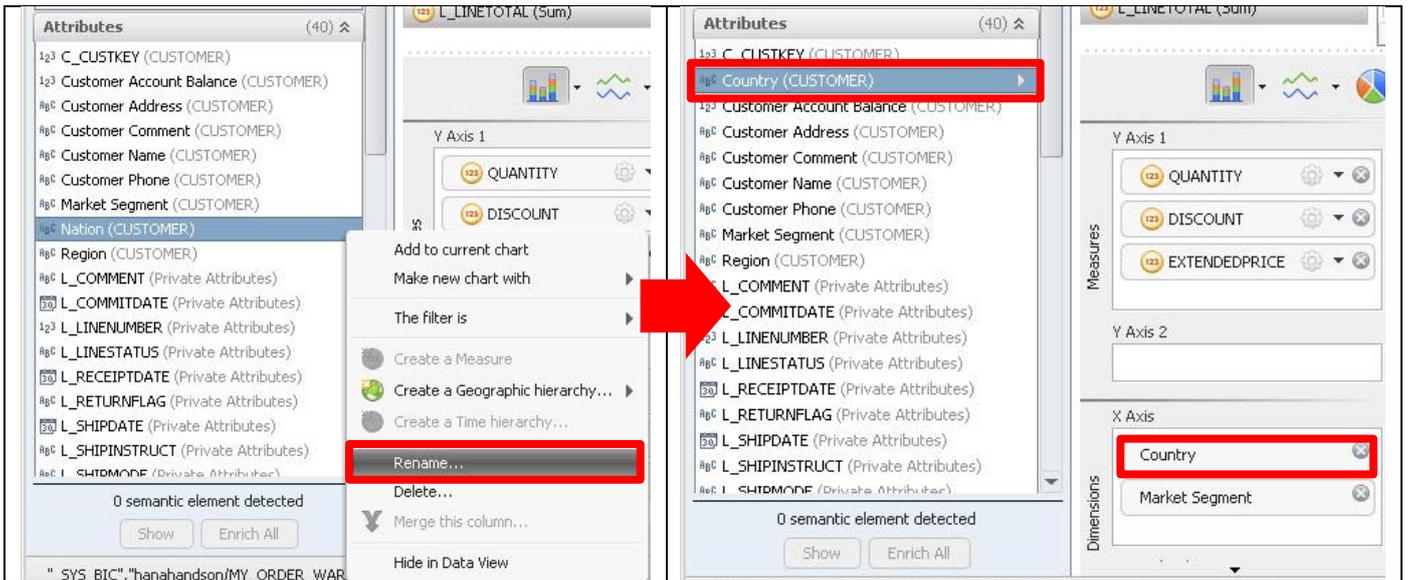
Once added, click away from the window to return back to the main screen, you should now have the below:



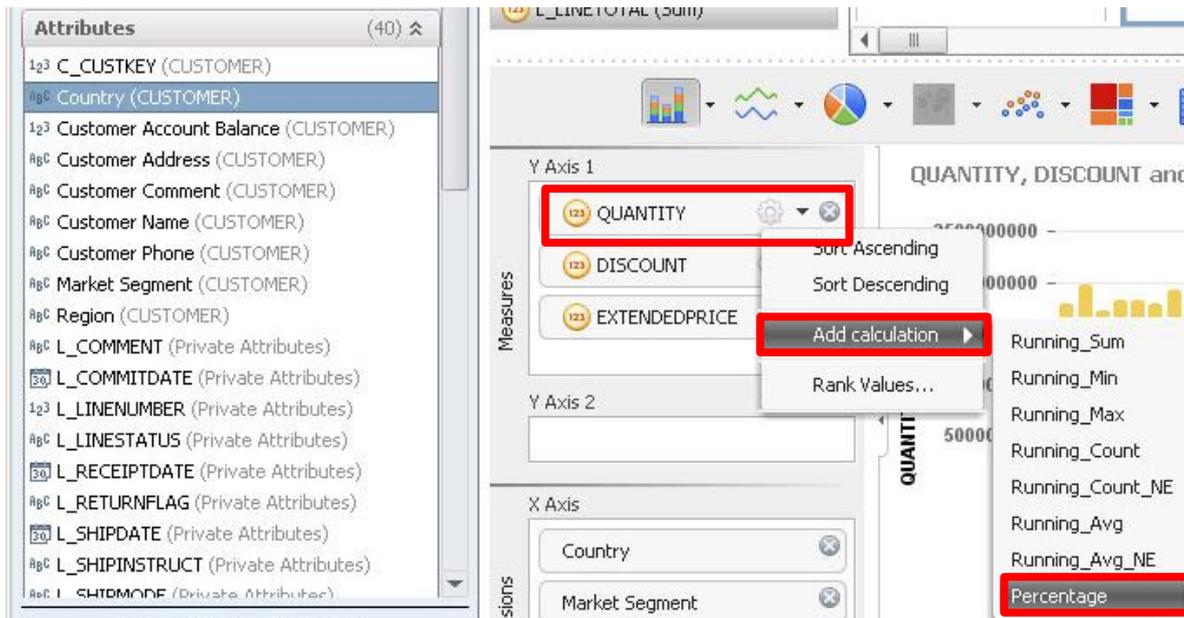
Select Nation and Market Segment and drag and drop these to the Dimensions area:



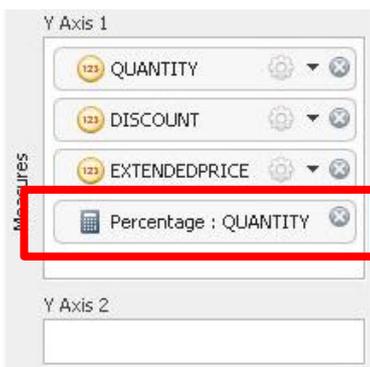
Rename Nation to Country, by right-clicking on Nation and then the small arrow to Rename:



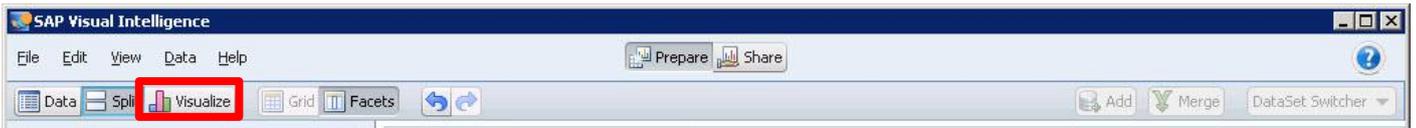
Create a calculated column which displays Percentage of Total for Quantity. Select the Quantity measure in Y Axis 1, select the triangle icon and choose Add calculation → Percentage.



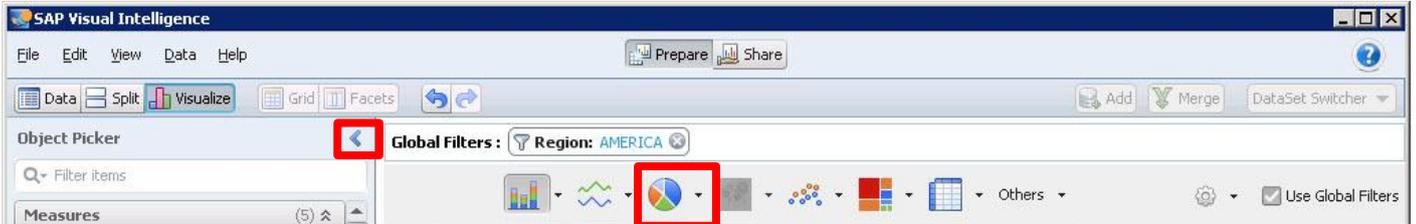
The new calculation should now be added below the existing measures:



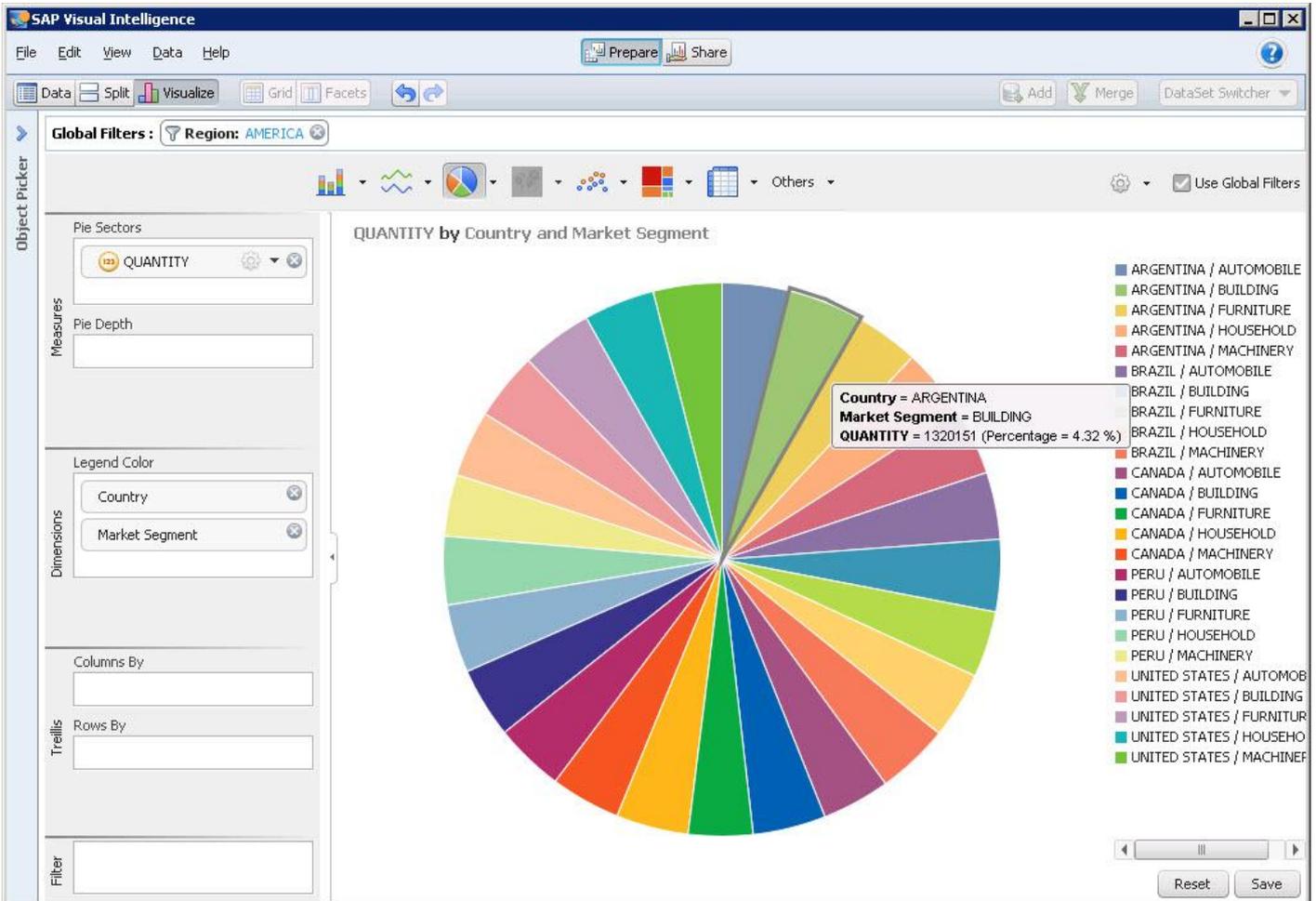
Select Visualize:



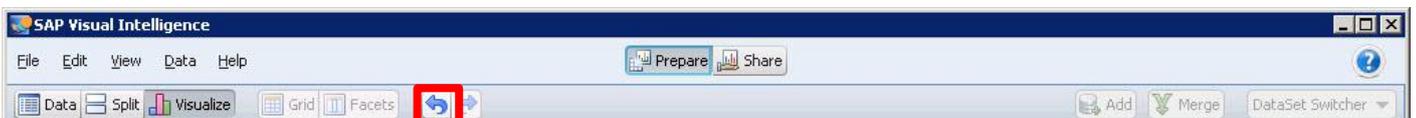
Then select the Pie Chart Icon and enlarge the window by hiding the left pane:



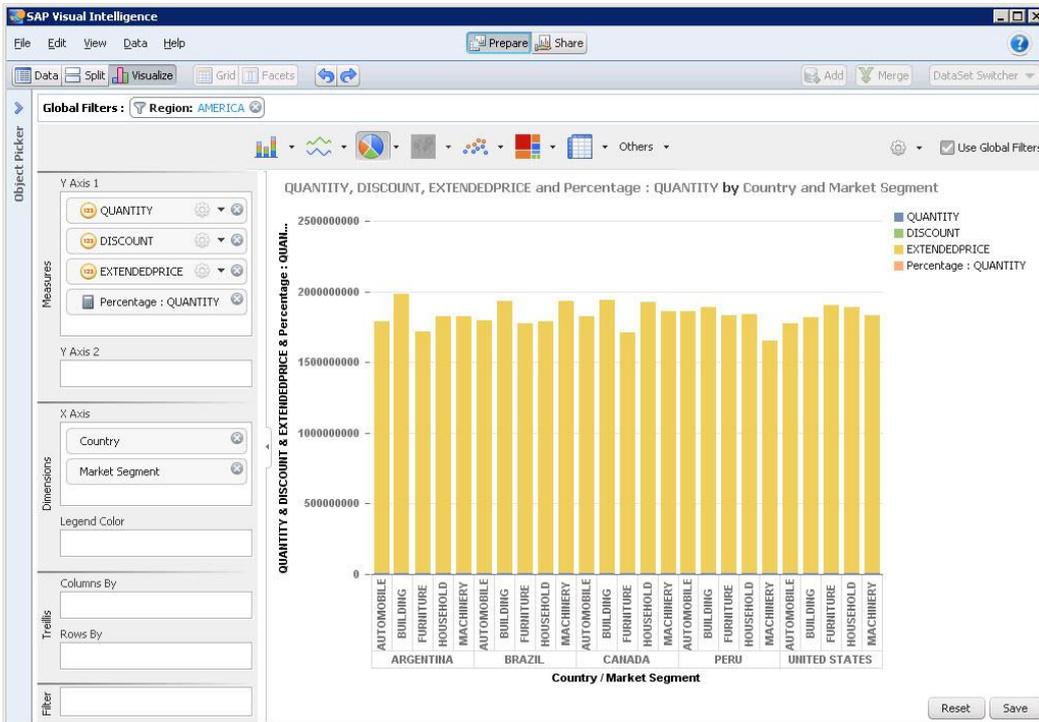
You should now see the below pie chart with the attributes arranged in alphabetical order:



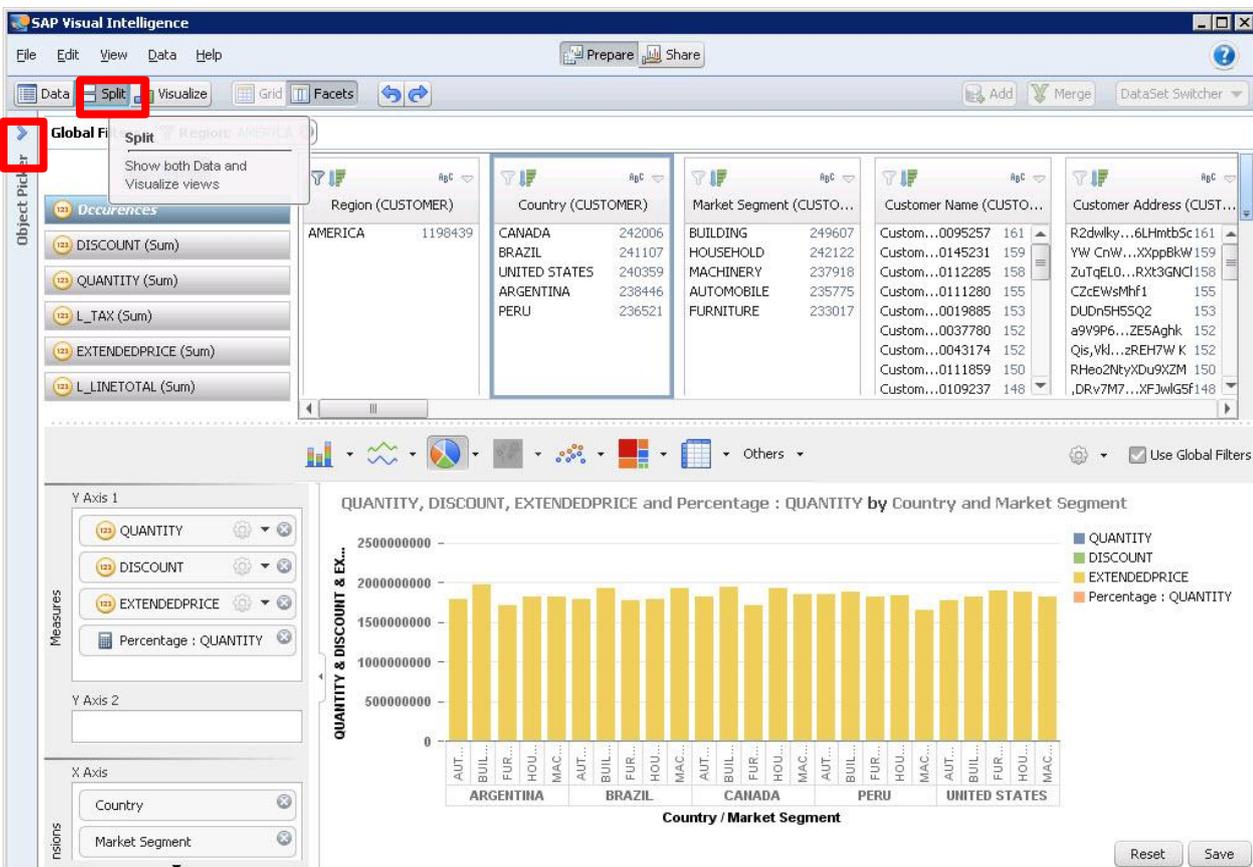
Now we are going to undo this to return to the previous state by selecting the undo icon:



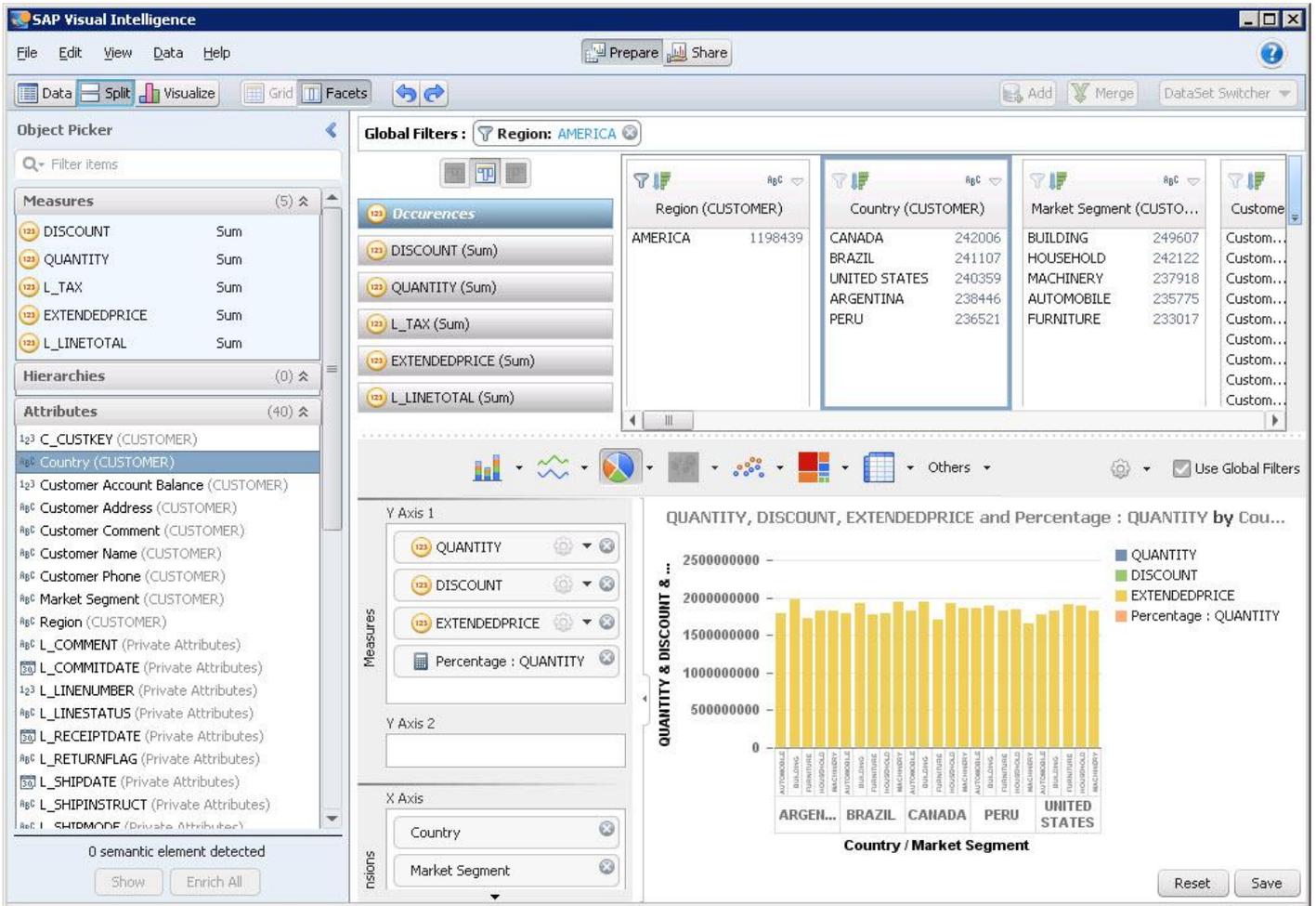
You should now see this:



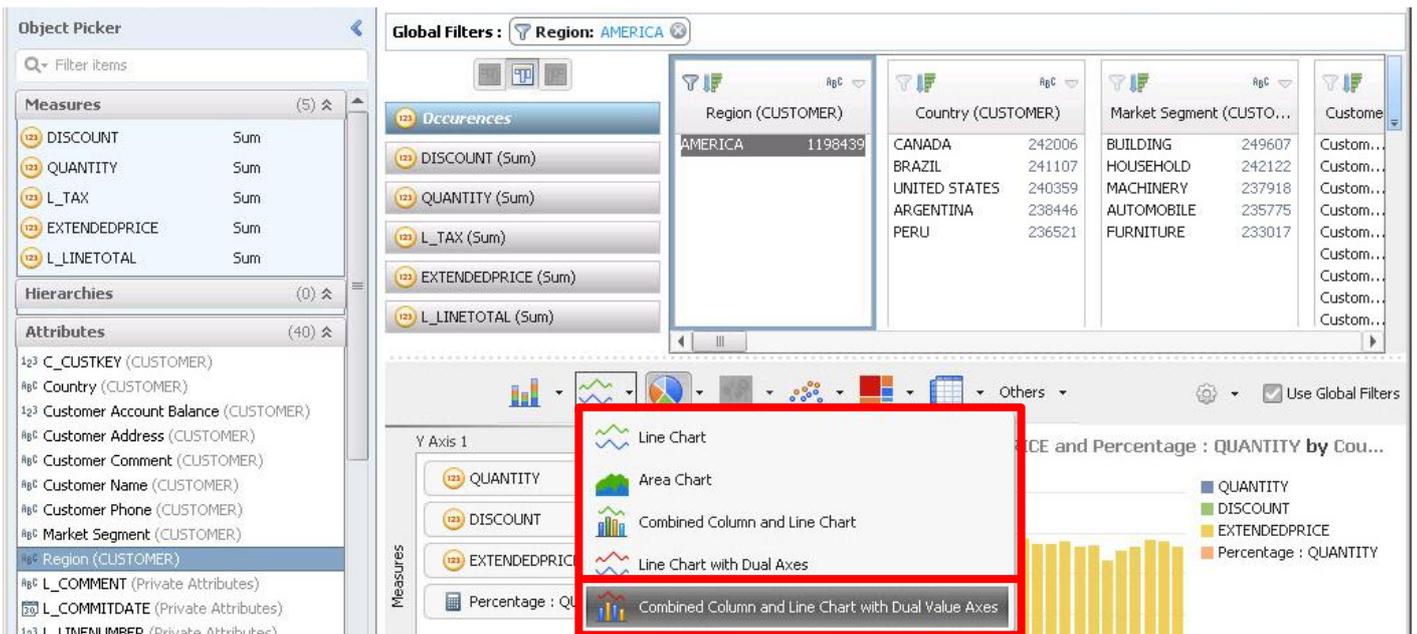
Select the Split Icon to see a more detailed view and expand the left pane by selecting the right pointing arrow:



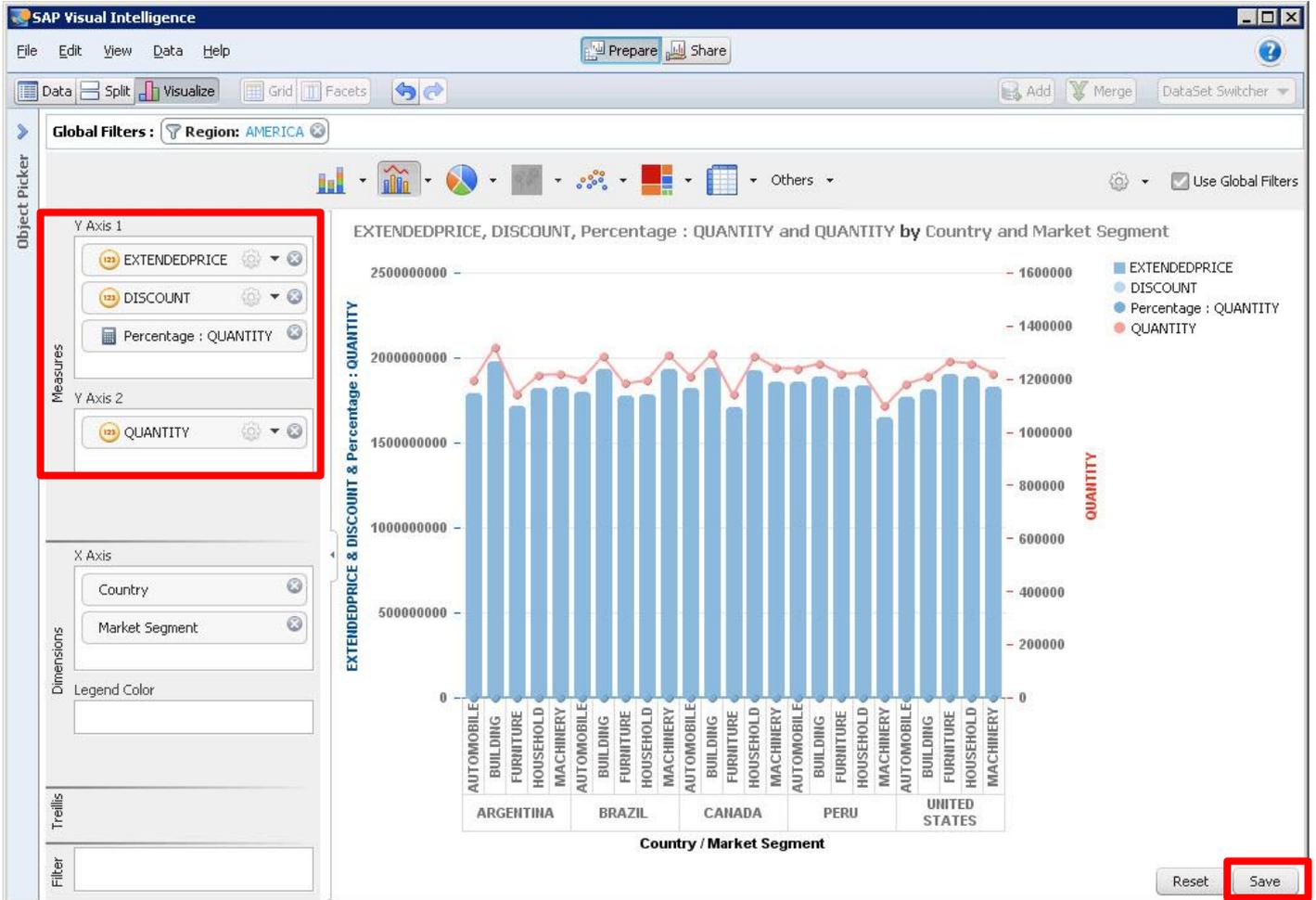
You should now have the following:



Switch to the Line Chart by selecting the icon and Combined Column and Line Chart with Dual Value Axes option.



Switch to the Visualize tab to analyze your chart, and re-arrange the measures so that EXTENDEDPRICE is first in the list followed by DISCOUNT and Percentage QUANTITY. Select the individual bars to view details about the data points.



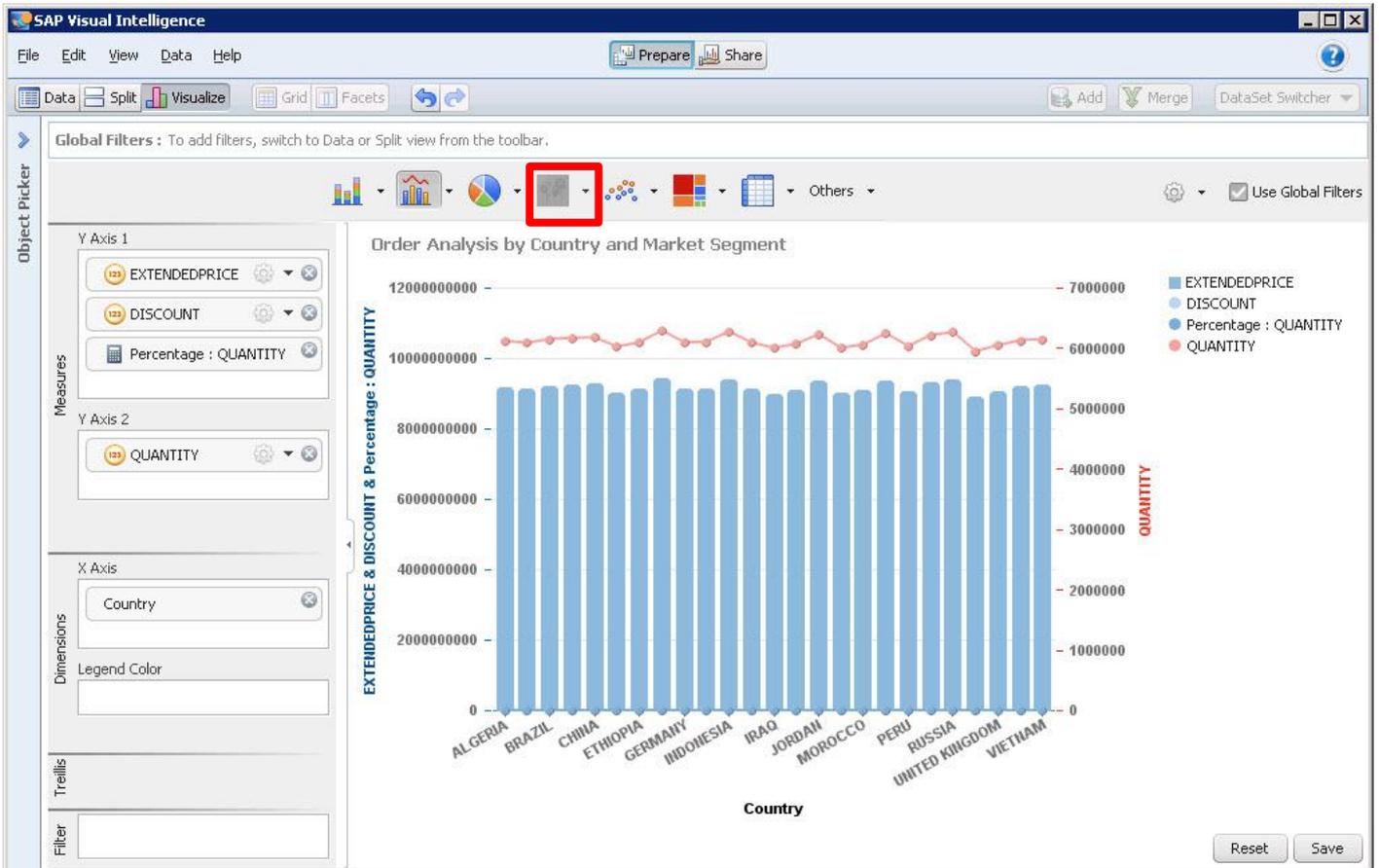
You can also double click on each data bar to zoom in to detail chart for each Country and Market Segment.

Rename your chart by double clicking on the title. Type "Order Analysis by Country and Market Segment":

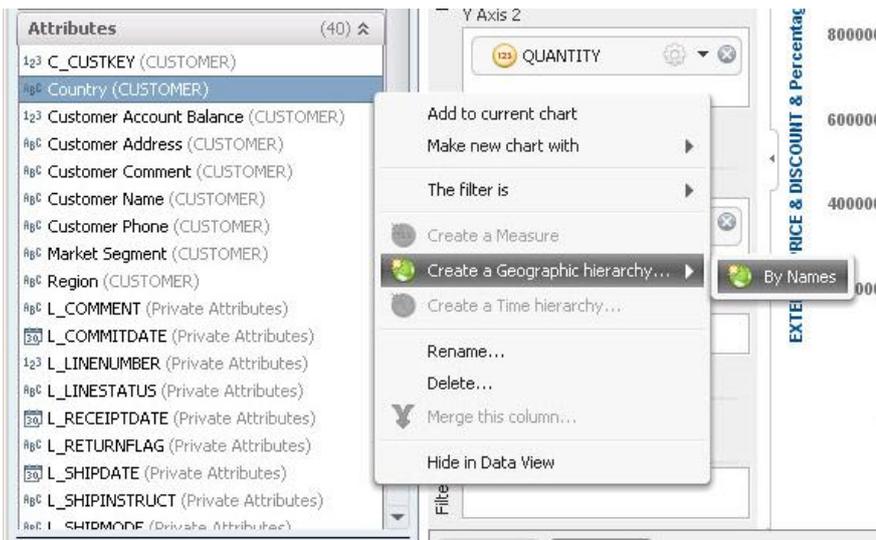


Save your Visualization.

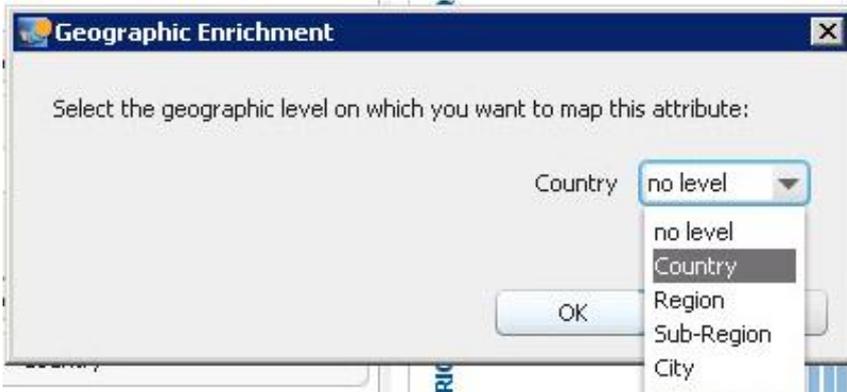
Notice that the Geographic Map is disabled. Clear all the selections by clicking on “X” (except Country):



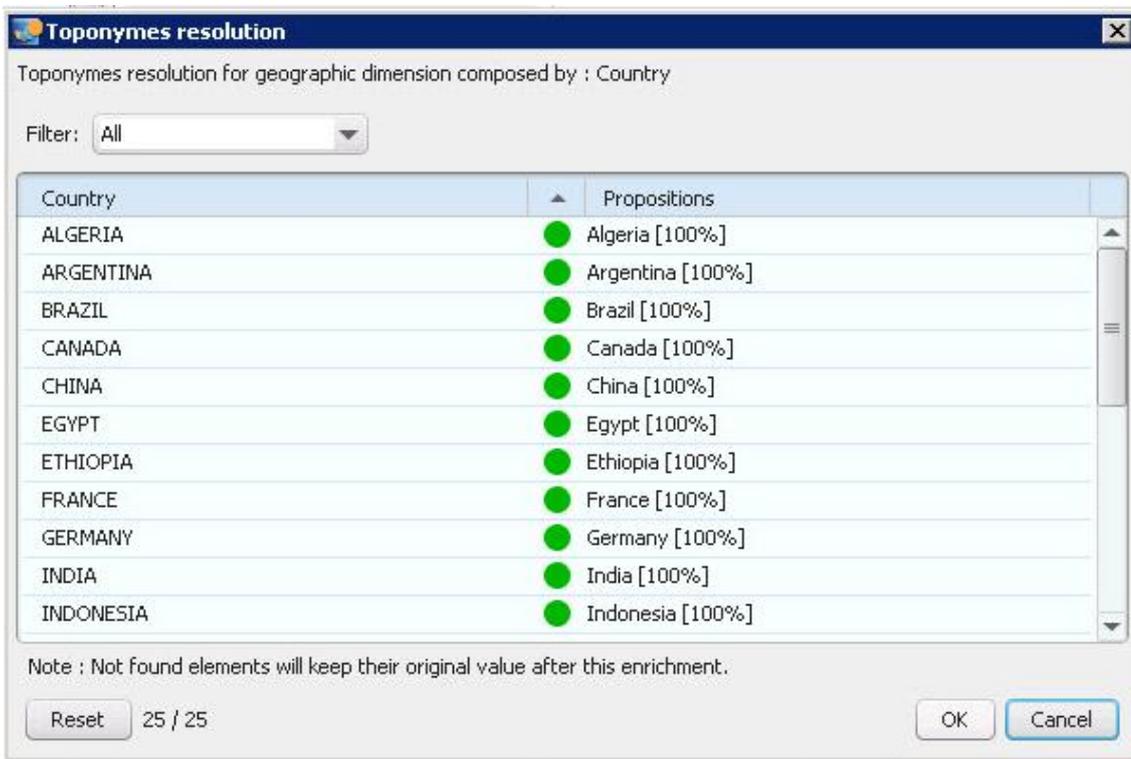
Create a geographic visual map by categorizing geographic dimension first. Select “Country” attribute and click on the arrow to the right. Click on “Create a Geographic hierarchy”:



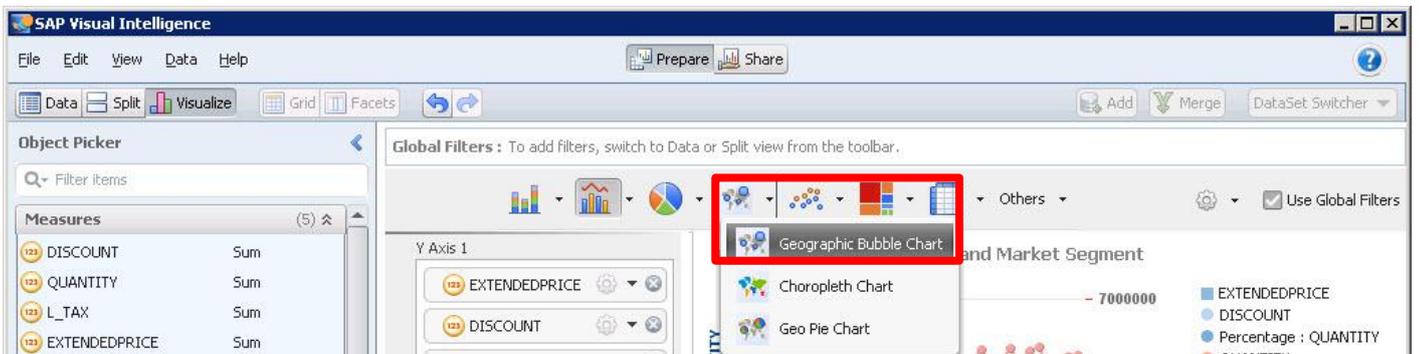
Select from the drop down Country:



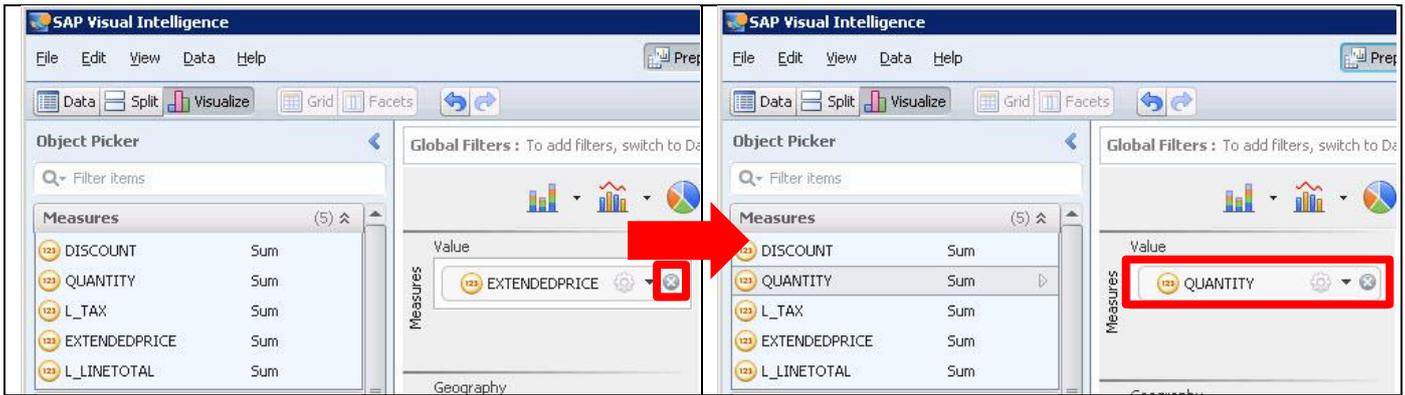
Confirm the elements identified are correct - This will create the geographic dimension. Click OK:



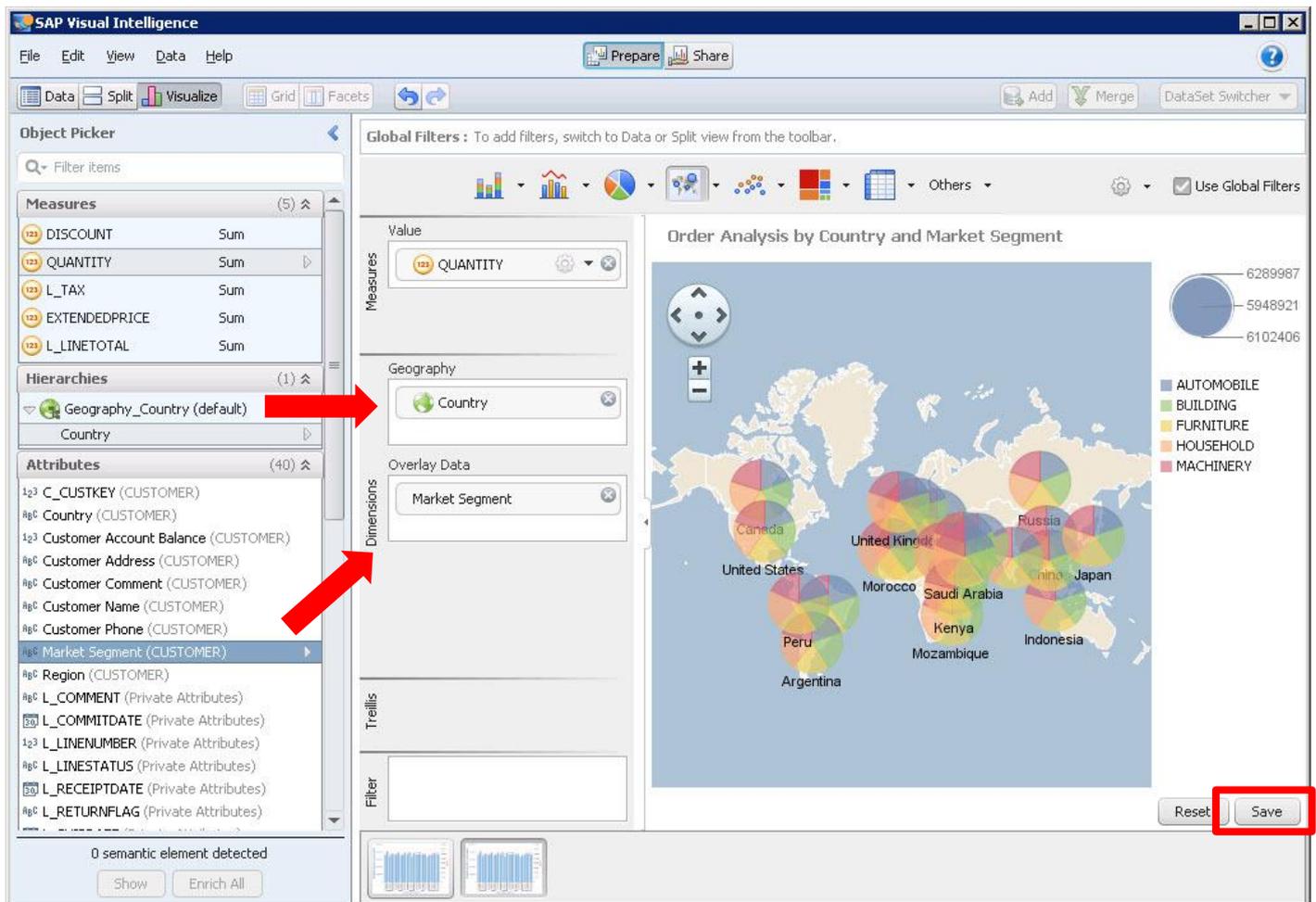
Select "GeographicBubble Chart":



Delete EXTENDEDPRICE using the delete icon and replace with QUANTITY by dragging it across.

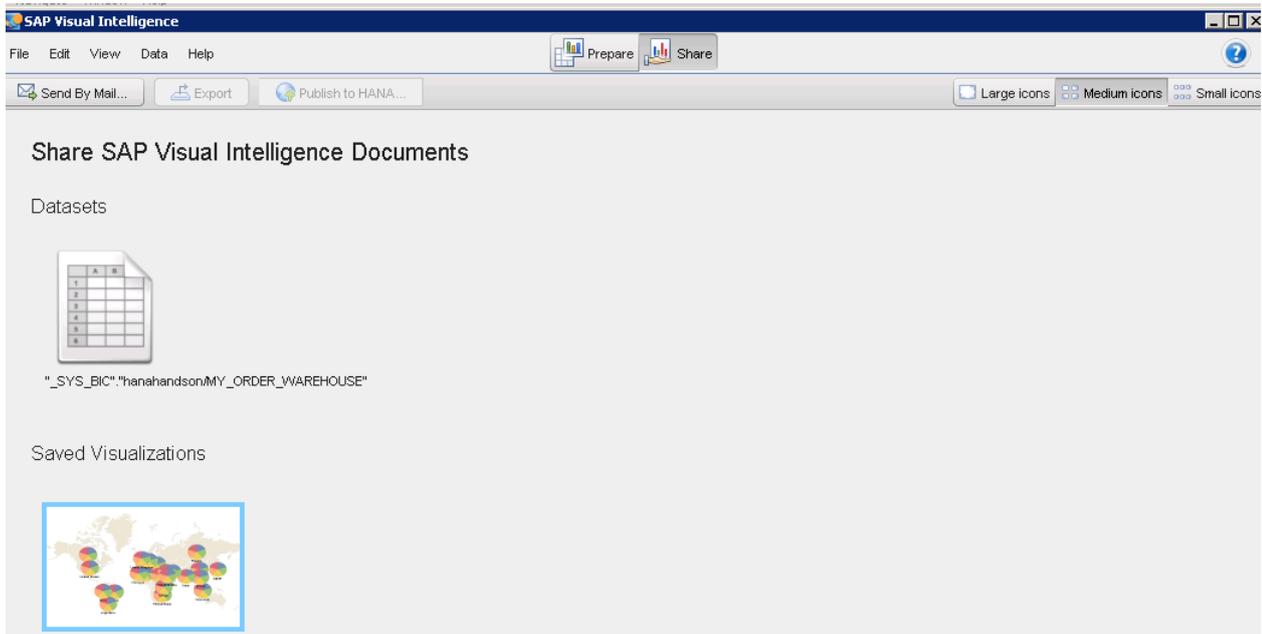


Select "Country" under hierarchies and drag it to Geography (if it is not there).
 Select "Market Segment" under attributes and drag it over to Overlay data.
 Click Save.



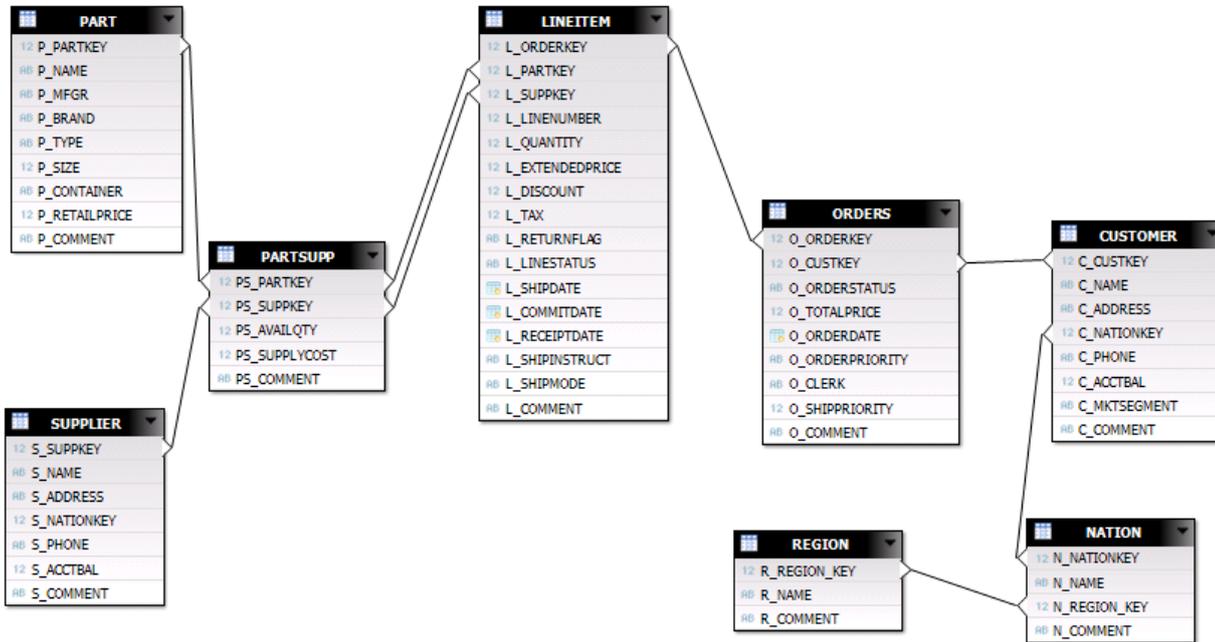


You can then share your charts with others via email or Streamwork, click “Share”.



Appendix

TPC-H ERD Diagram



Weblinks

<http://www.saphana.com>

<http://www.sap.com/hana>

<http://www.sap.com/pc/tech/in-memory-computing-hana.html>

<http://www.sdn.sap.com/irj/sdn/in-memory>

<http://help.sap.com/hana>