SAP How-to Guide
Business Analytics

SAP HANA™ Appliance

# How to Enhance SAP HANA Live

**Applicable Releases:**

**SAP HANA DB 1.0 SPS05, SAP HANA Studio Version 1.0.60**

**Version 1.1**

**July 2013**

The Best-Run Businesses Run SAP™

© Copyright 2013 SAP AG. All rights reserved.

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.

SAP "How-to" Guides are intended to simplify the product implementation. While specific product features and procedures typically are explained in a practical business context, it is not implied that those features and procedures are the only approach in solving a specific business problem using SAP NetWeaver. Should you wish to receive additional information, clarification or support, please refer to SAP Consulting.

Any software coding and/or code lines / strings ("Code") included in this documentation are only examples and are not intended to be used in a productive system environment. The Code is only intended better explain and visualize the syntax and phrasing rules of certain coding. SAP does not warrant the correctness and completeness of the Code given herein, and SAP shall not be liable for errors or damages caused by the usage of the Code, except if such damages were caused by SAP intentionally or grossly negligent.

Disclaimer

Some components of this product are based on Java™. Any code change in these components may cause unpredictable and severe malfunctions and is therefore expressively prohibited, as is any decompilation of these components.

Any Java™ Source Code delivered with this product is only to be used by SAP's Support Services and may not be modified or altered in any way.

The Best-Run Businesses Run SAP™

## Document History

| Document Version | Description |
|---|---|
| 1.10 | Update the guide with layout from HANA Studio version 1.0.60 |
| 1.00 | First official release of this guide |

## Typographic Conventions

| Type Style | Description |
|---|---|
| *Example Text* | Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options.<br>Cross-references to other documentation |
| **Example text** | Emphasized words or phrases in body text, graphic titles, and table titles |
| `Example text` | File and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools. |
| **`Example text`** | User entry texts. These are words or characters that you enter in the system exactly as they appear in the documentation. |
| **`<Example text>`** | Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system. |
| `EXAMPLE TEXT` | Keys on the keyboard, for example, `F2` or `ENTER`. |

## Icons

| Icon | Description |
|---|---|
| | Caution |
| | Note or Important |
| | Example |
| | Recommendation or Tip |

The Best-Run Businesses Run SAP™

## Table of Contents


**1. Business Scenario ..... 1**

**2. Background Information ..... 1**

**3. Prerequisites ..... 3**

**4. Include Unexposed Field from a Standard Model – Step-by-Step Procedure ..... 4**

4.1 Browse standard SAP Hana Live content for standard query view ..... 4
4.2 Create a new query view in the customer namespace ..... 8
4.3 Enhance the views with relevant reuse views ..... 9
4.4 Activate and Test the view ..... 14

**5. Exclude unnecessary fields from Query view – Step-by-Step Procedure ..... 15**

5.1 Identify unnecessary fields ..... 15
5.2 Exclude the unnecessary fields from the query view ..... 15
5.3 Test the view ..... 15

**6. Include Custom Table(s) - Step-by-Step Procedure ..... 17**

6.1 Create a Custom Reuse View ..... 17
6.2 Create a new Query View ..... 18

**7. Include Custom Field(s) - Step-by-Step Procedure ..... 20**

7.1 Create a new Calculation View for the custom field(s) ..... 20
7.2 Locate and copy the appropriate Query view that needs to be extended ..... 22
7.3 Enhance the copied view to include the newly created custom view ..... 22
7.4 Test the Modified Query View ..... 24

# 1. Business Scenario

The SAP HANA Live scenario has been implemented in your SAP Suite on HANA server or in your SAP HANA Sidecar Server, depending on your current SAP HANA Database Server implementation.

You want to modify the delivered scenarios to reflect your reporting requirements and to include any custom fields that have been added to the standard SAP tables, as well as including custom tables that are part of your SAP implementation.

As an example, this guide uses the following scenario:

Customer IDES Company, have their own classification of countries (Developed, Emerging, Transitional) and their own classification of customers (Large, Medium, Small) that they use in their custom applications. They want to report Sales Orders based on these additional classifications as well. Since SAP Hana Live delivered by SAP already includes standard views for reporting on sales orders they have decided to extend these views to suit their requirements.

This guide consists of step-by-step instructions on the following 4 scenarios to satisfy the customer Sales Order reporting requirements.

- Include Unexposed Field from a Standard Model
- Exclude unnecessary fields from Query view
- Include Custom Table(s)
- Include Custom Field(s)

# 2. Background Information

The SAP HANA Live scenario as delivered contains “Private Views” which are the base views that directly use the SAP Tables.

The next level of views is the “Reuse Views” which consist of one or more “Private” views joined in combination to produce a composite view that forms a reusable unit related to a particular area or function within the system.

The top level of views is the “Query Views” which consist of one or more “Reuse” views joined in combination to expose the major attributes and measures that are needed to report on the various business scenarios. These “Query Views” are the only views that are exposed to the Client Tools to be used for reporting.

Since all of these different types of views are based on Standard SAP tables, they do not include any custom fields that may have been added to these tables in the SAP system, nor do they contain any custom tables that may have been added. The standard “Query Views” generally also contain more fields than are required for any particular reporting scenario.

The purpose of this guide is to show how to enhance the standard SAP HANA Live implementation to enable enterprise specific reporting which uses any custom data and caters for the specific requirements of the enterprise.

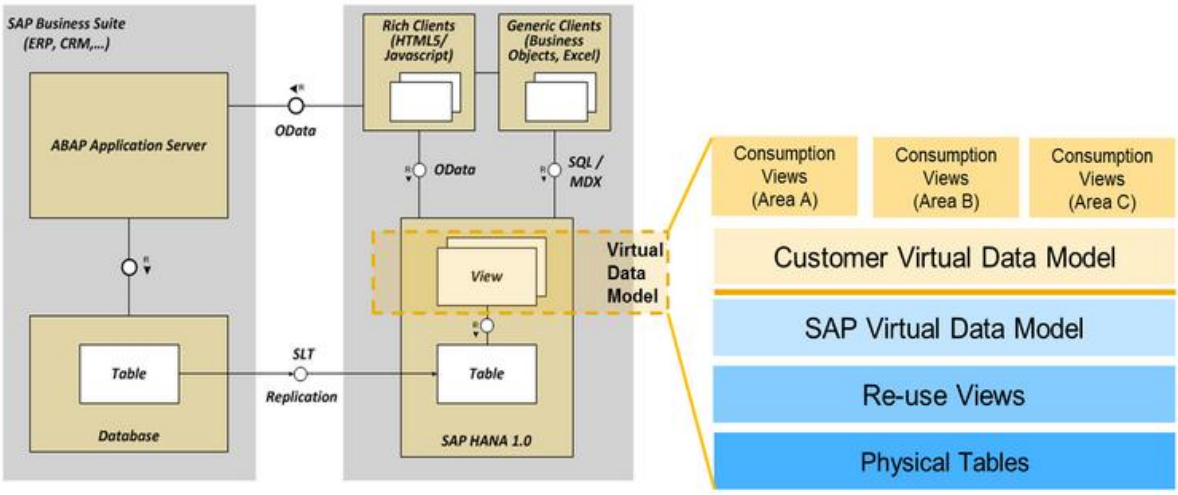



**Figure 1: SAP HANA Live deployment in a side by side manner**

**Important**

**Some of the important rules while modeling a query view:**

- Do not extend SAP delivered Views directly. Use Query/Reuse views for extensions by copying over to customer namespace
- To find a list of Query and Reuse views refer to http://help.sap.com/hba → SAP Hana Live for SAP ERP/CRM/SCM etc. Views that are not mentioned here are Private Views and should not be copied or modified
- For all graphical calculation views the flag 'Enforce SQL Execution' must be set to 'true'
- For all enhancements, graphical calculation views must be used. If they are not sufficient, use script based calculation views
- Avoid using Attribute and Analytical views unless there is a requirement to use features that are only available in these types of views

# 3. Prerequisites

The following prerequisites must be fulfilled before engaging in any enhancement of the supplied SAP HANA Live.

- The SAP HANA Server must be installed with the correct version of SAP HANA as per the requirements for the implemented release of SAP HANA Live.
- The appropriate standard SAP tables that are used by the SAP HANA Live implementation must exist and be populated, either directly by the SAP Application Server, or in a sidecar scenario, by an appropriate data replication mechanism. Refer to http://help.sap.com/hba for guides related to SAP HANA Live
- The database tables include the custom fields and tables, as well as custom data, required by the SAP HANA Application server.
- The SAP HANA Client and Studio software has been installed on the client systems that are being used to connect to the SAP HANA Database and manage the enhancement of the SAP HANA Live implementation, and the release versions of these applications are compatible with the release versions of the SAP HANA Database.

Before following any of the scenarios covered by this How-To guide, you must:

- Be familiar with the SAP HANA Live release notes.
- Have a connection from the client SAP HANA Studio to the SAP HANA Server.
- Understand the business requirements for the reporting scenarios.
- Be familiar with and have experience with the concepts and use of the SAP HANA Studio application.
- Have considered an appropriate naming convention, testing and transport mechanism to ensure a fully tested and controlled development process.

# 4. Include Unexposed Field from a Standard Model – Step-by-Step Procedure

In the Sales Order query view, the customer wants to see the Sales Organization, Sales District and also wants to see the 'Total Net Amount' based on these columns. Start by browsing through the SAP delivered SAP HANA Live (SHL) content for Business Suite. All the content can be found in the package: sap → hba

For the Sales Order query, the SalesOrderQuery (calculation view) in the 'ecc' folder of the SAP Hana Live content.

## 4.1 Browse standard SAP Hana Live content for standard query view

1. Locate the SAP Hana Live package using the SAP Hana Studio in the 'sap' package.
   Under sap → hba → ecc → Calculation Views, find the Sales Order related query view.
   Query views are easy to spot as they always end with the term 'Query'.
   For this scenario, SalesOrderQuery will be used.

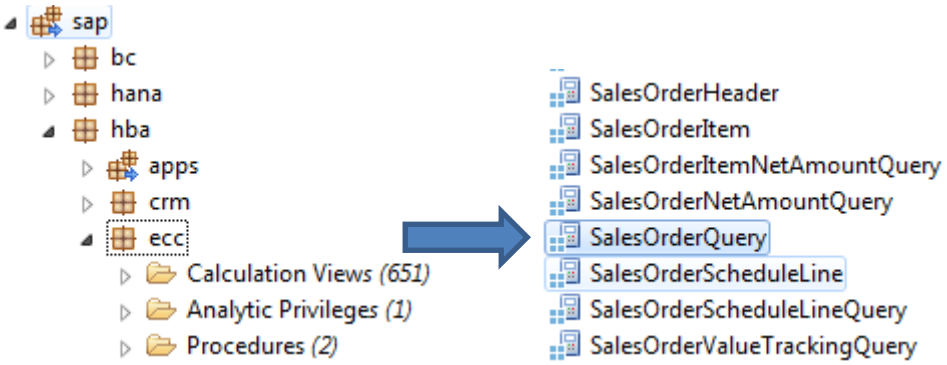
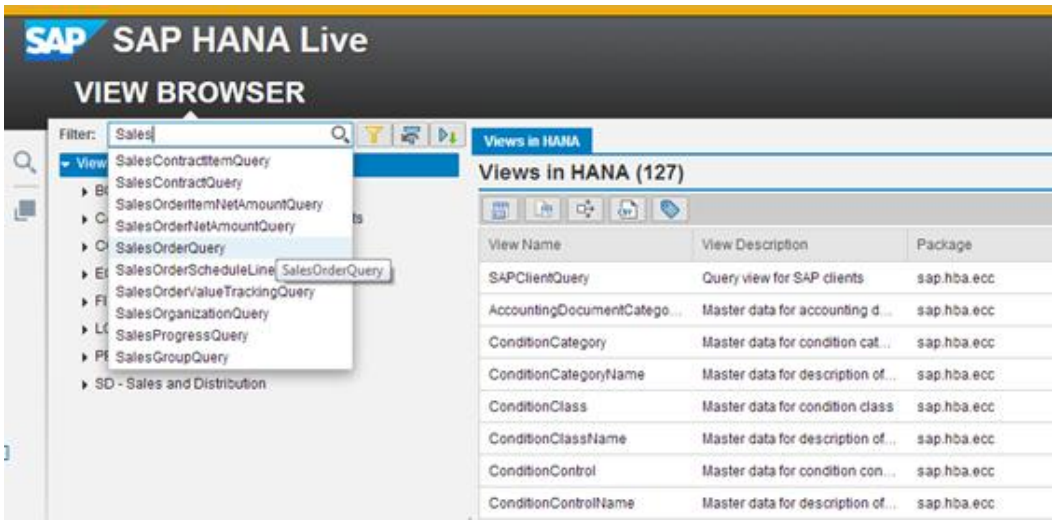



**Figure 2: SAP HANA Live content in the SAP Hana Studio**

2. There are various other calculation views in the package which are not 'Query views'. They are either 'Private views' or 'Reuse views'.
   To find out which ones are Private views and which others are Reuse views, refer to https://help.sap.com/hba → SAP Hana Live for ERP (or for other applications) → Choose the Functionality from the left side hierarchy

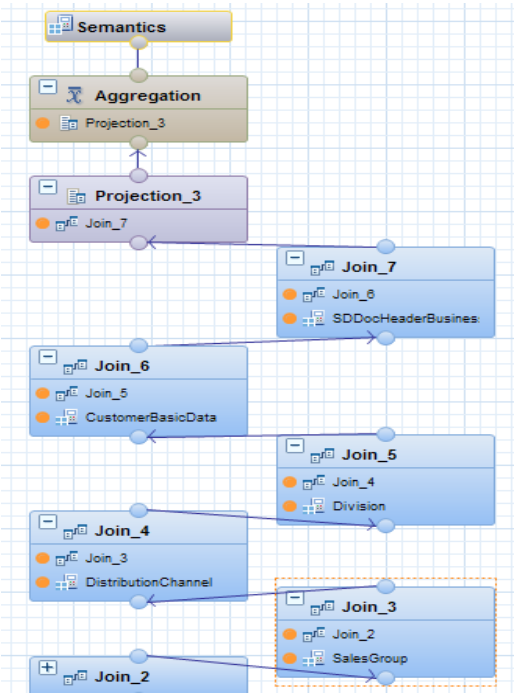
**Figure 3: SAP HANA Live view types in SAP Help documentation**

3. You can also browse the views using the SAP HANA Live View Browser
   To access the view browser, go to http://<server>:<port>/sap/hba/explorer, where port is 80<instance number>
   For more information, look at http://help.sap.com/hba -> SAP HANA Live




4. Double click and open the SalesOrderQuery view in the HANA modeler.
   Opening the SalesOrderQuery view shows that it uses many reuse views. Double-clicking on these will display the view. Query views are the only views that are exposed to your reporting tools.

   Do not reuse the standard 'Query views', but use a copy in your own namespace and make the necessary modifications there.




**List of reuse objects that the standard query view SalesOrderQuery view uses. These views can be reused and adapted to the new requirements.**

sap.hdba.ecc.SDDocHeaderBusinessPartner

sap.hdba.ecc.CustomerBasicData

sap.hdba.ecc.Division

sap.hdba.ecc.DistributionChannel

sap.hdba.ecc.SalesGroup

sap.hdba.ecc.SalesOffice

sap.hdba.ecc.SalesDocumentType

sap.hdba.ecc.SalesOrderHeader

**Figure 4: Query view example, SalesOrderQuery**

5. Look at the output to understand which fields are available from the SalesOrderQuery view. Check to make sure that all the fields that are needed for your reporting purpose exist.
   The view seems to have most of the fields that are needed for the report. But the SalesDistrict is missing. Since this is one of the fields that is needed for the report, we can check the reuse views to see whether this information is available in one of the views but is not exposed to the view output.

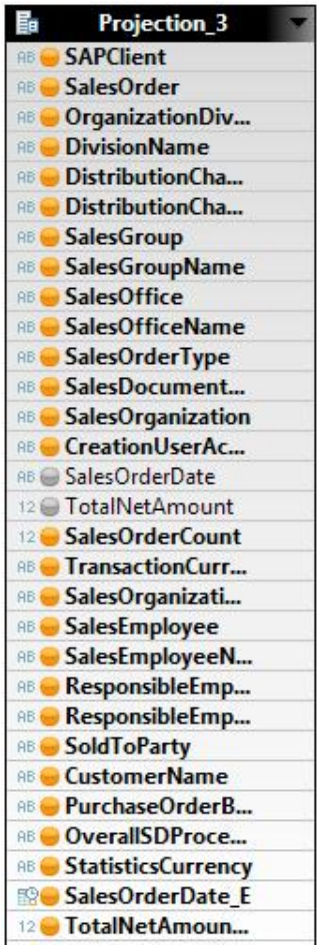



**Figure 5: Verify fields needed for output**

6. Looking at the SalesOrderHeader reuse view, it can be seen that the SalesDistrict column is available but is not exposed to the query view.

**Note**

Clicking on the 'projection' node will display all the fields available to that node. Any field that is **bolded** is exposed by the projection. Fields not bolded are available in the projection node, but are not exposed to any node further on.

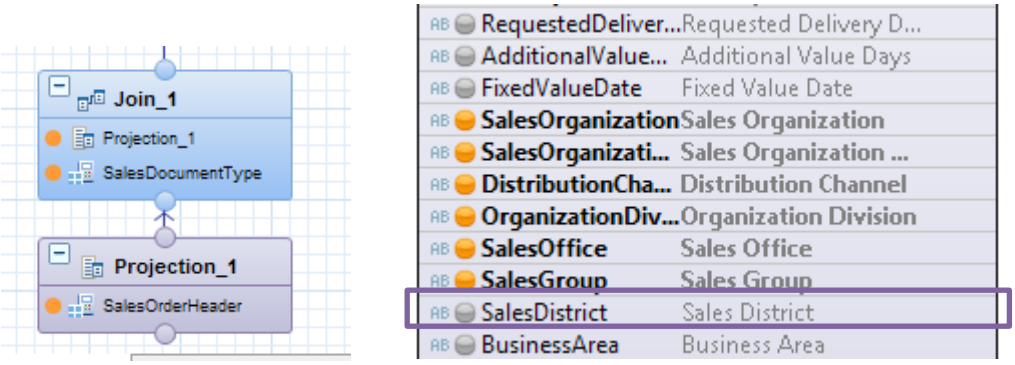



**Figure 6: Identify the reuse view to be used**

## 4.2 Create a new query view in the customer namespace

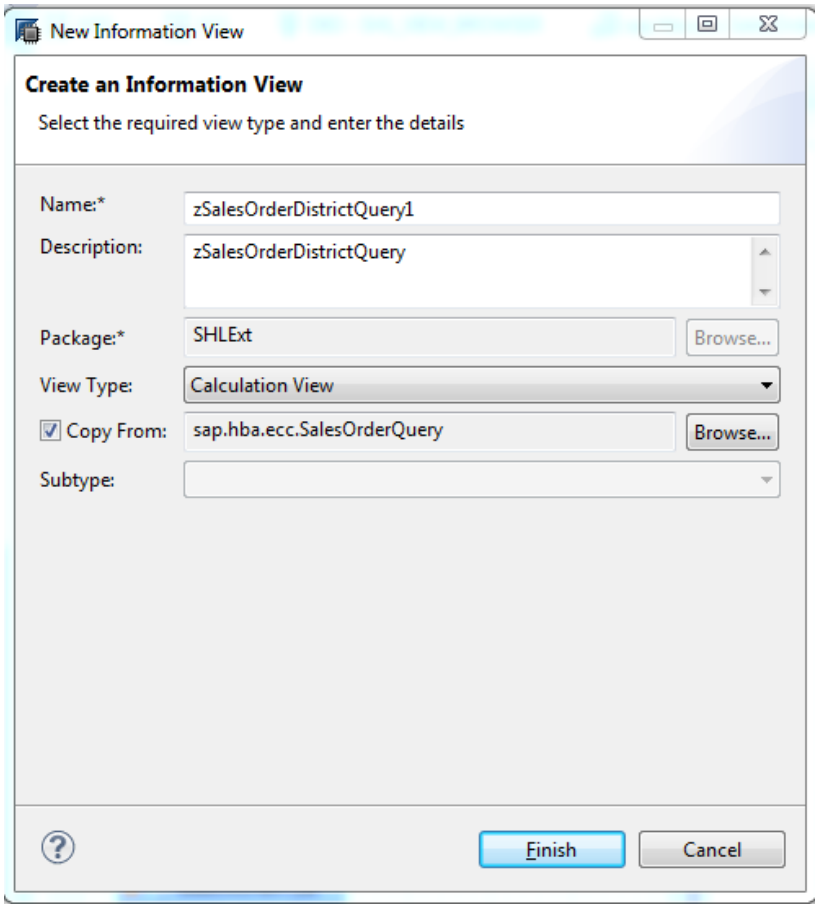



**Figure 7: New Calculation View as a Copy**

7. Create a new graphical calculation view “zSalesOrderDistrictQuery1” in your customer package, select the check box “Copy From”, and enter the name of the standard view “SalesOrderQuery”.

8. Make sure that Multidimensional Reporting is set to ‘enabled’ and Enforce SQL Execution is set

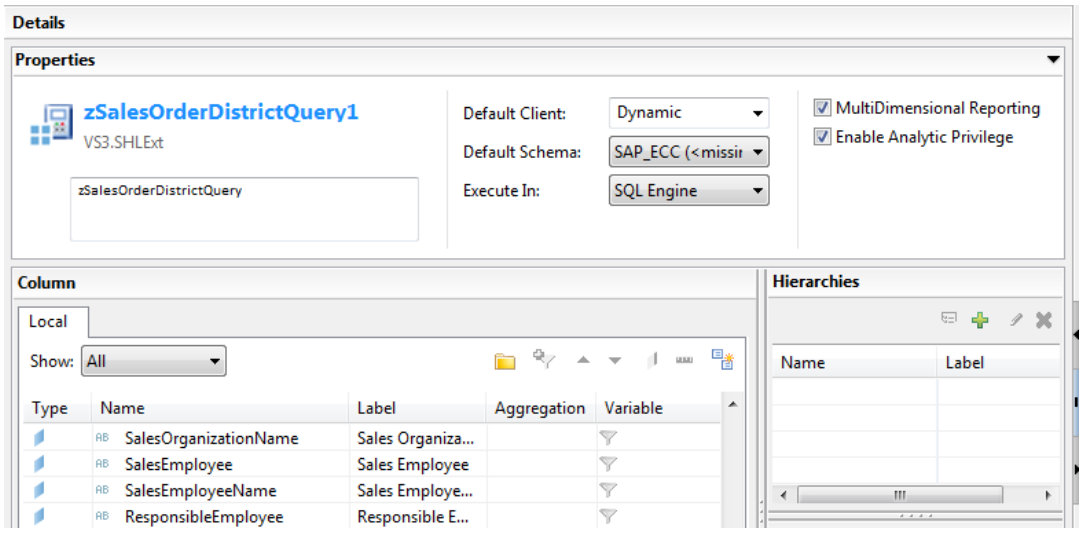



**Figure 8: Properties to be set for query views**

9. In zSalesOrderDistrictQuery1, select the Projection from the 'SalesOrderHeader' and send the SalesDistrict field to the output

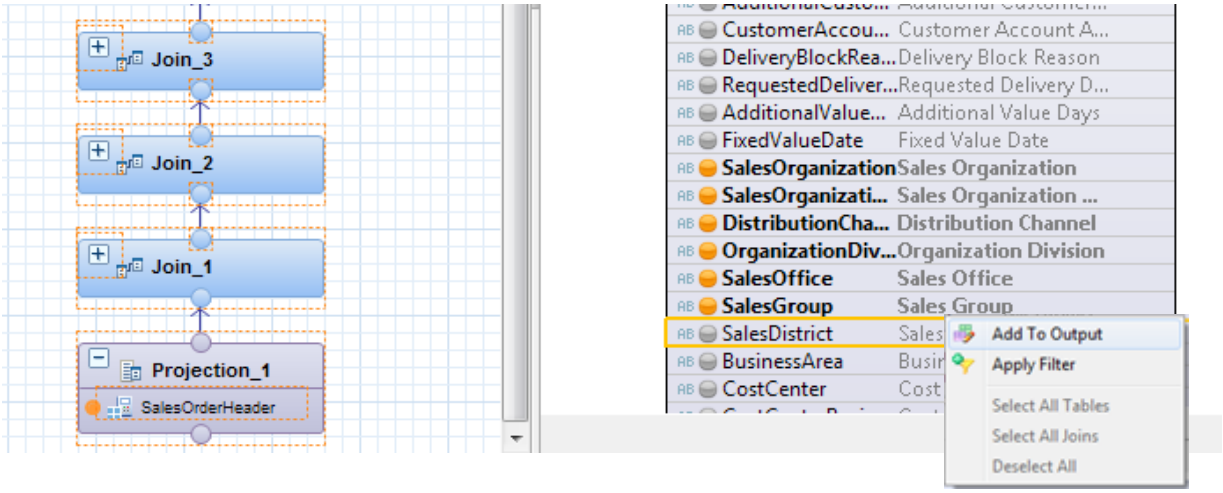



**Figure 9: Include Sales District in the Projection**

10. Select the "Join_1" node then add the SalesDistrict field to the output. At the Semantics level, set the column 'Type' as 'Attribute' for 'SalesDistrict'
11. Do the same thing for all the other Join nodes.

**Tip**

You can use the 'Data Preview' function by right clicking on any Join/Projection node to see intermediate results while you are adapting the views

## 4.3 Enhance the views with relevant reuse views

The SalesDistrict key has now been included, but to make the report usable the SalesDistrictName text needs to be exposed in the query view.

12. This information is available in another reuse view called "sap.hba.ecc.SalesDistrict"

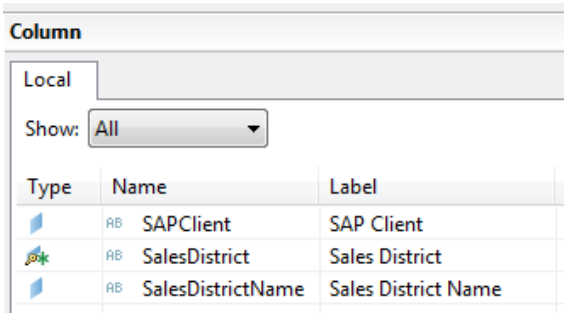



**Figure 10: Additional reuse view SalesDistrict**

13. In order to convert the SalesDistrict key into the SalesDistrictName text, this view needs to be added and joined into our query view.
To include the view, delete the link between the last Join node and the Projection node.

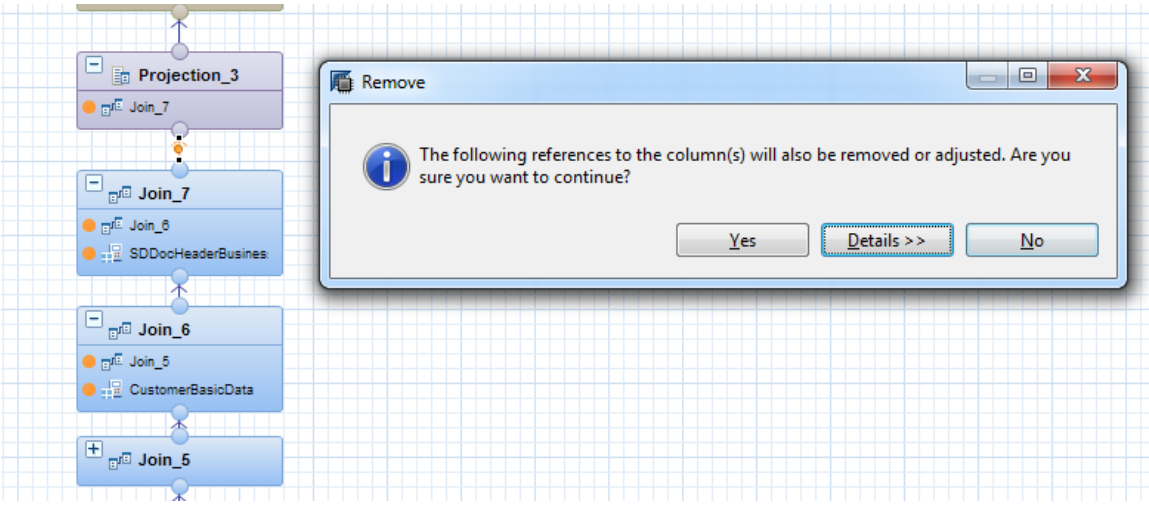



**Figure 11: Remove the link to join in the query view**

14. Add a Join node, and to remain consistent, rename it to “Join_8”.
15. Now drag and drop the reuse view “SalesDistrict” onto the screen as shown in figure 12.
16. Connect the nodes Join_7, Join_8, Projection_3 and SalesDistrict as shown in figure 12.

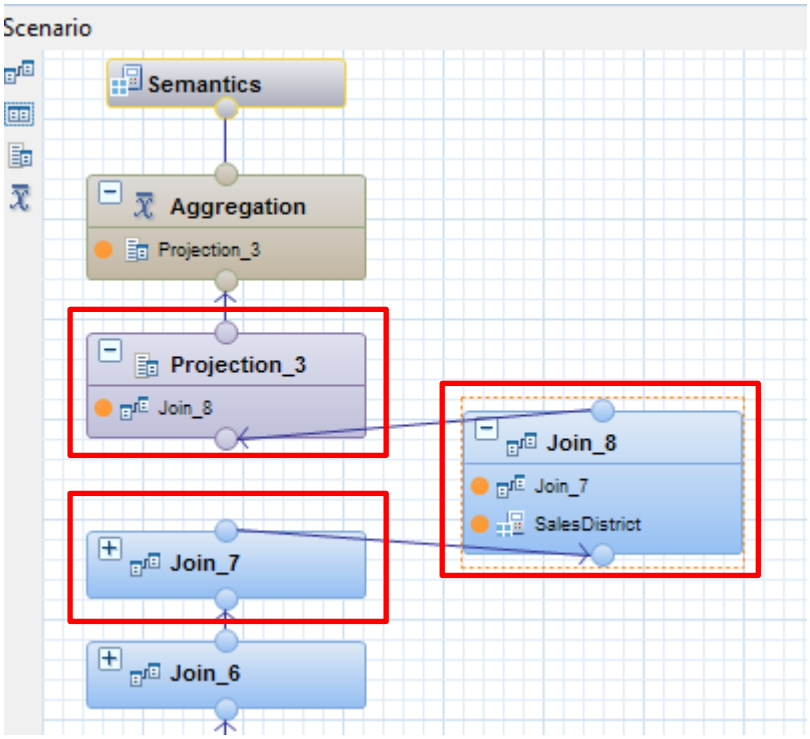



**Figure 12: Add Reuse View and Join node, and Link Nodes**

17. Select the Join_8 node, and then join the SAPClient and SalesDistrict fields from the Join_7 node to the same fields from the SalesDistrict reuse view. Then select one of the joins as shown in

figure 13 and change the Join Type to ‘Left Outer’ and Cardinality to ‘n..1’.

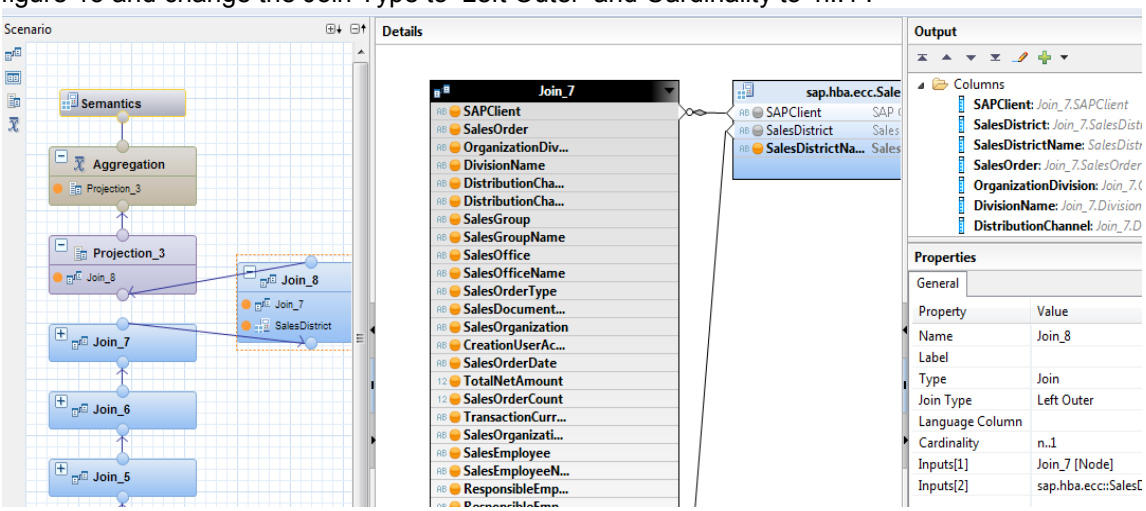



**Figure 13: Joins Between Join_7 node and SalesDistrict reuse view**

18. In order to make required columns available in Projection_3, Select the Projection_3 node and delete the calculate columns already present. Note that the rest of the columns are currently not available.

    Click on Yes for the warning question dialog.

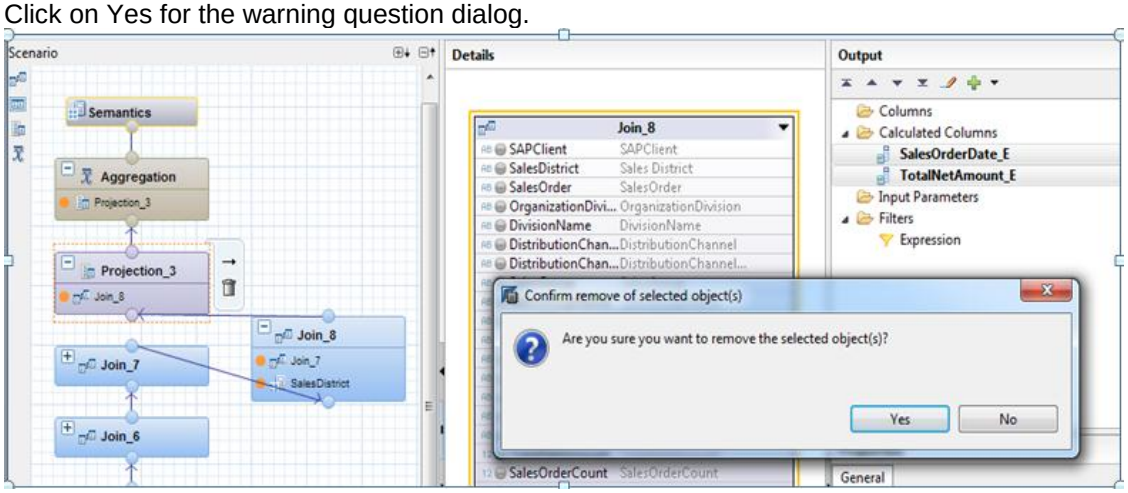



**Figure 14: Remove the old calculated columns from Projection_3**

19. Click on the Projection Node, and then add all the attributes to Output of Projection_3.

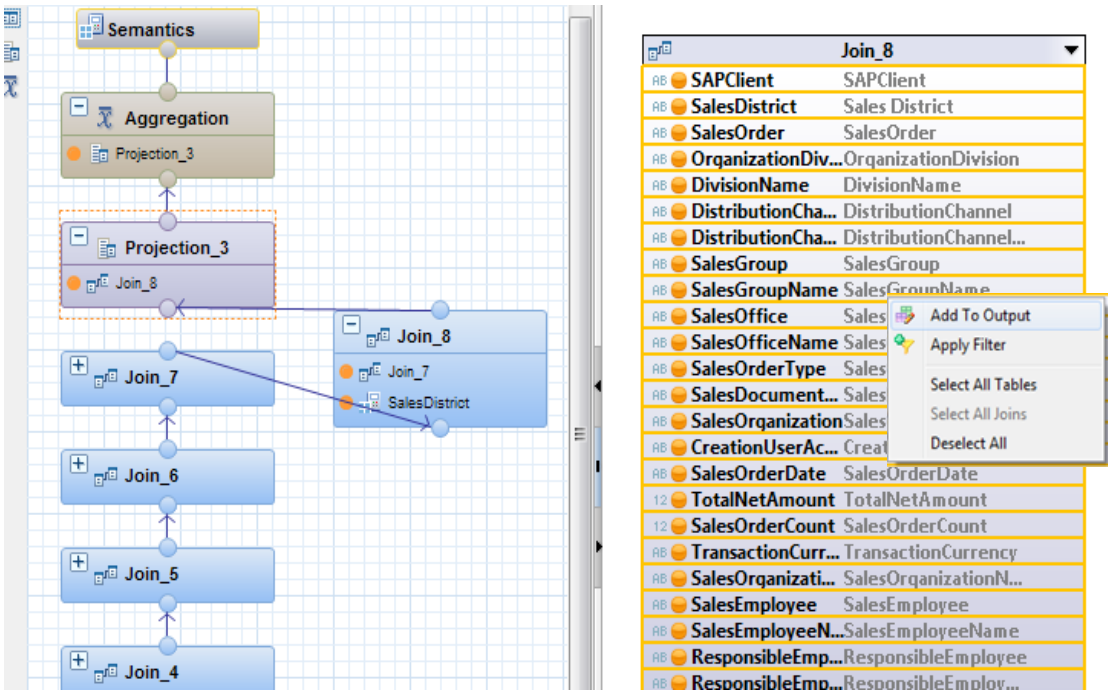



**Figure 15: Select and remove Projection Node Columns**

20. Open up the original SalesOrderQuery view, and use this view to recreate the two Calculated Columns.

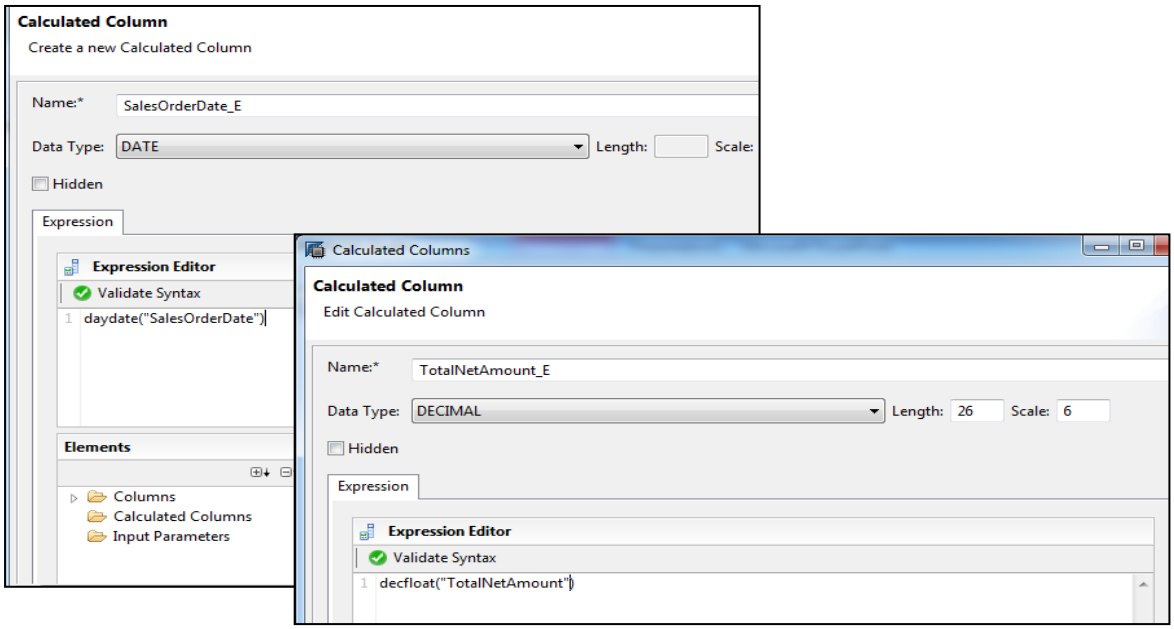



**Figure 16: Calculated Columns – SalesOrderDate_E and TotalNetAmount_E**

21. In the Aggregation node, include the necessary columns to output.

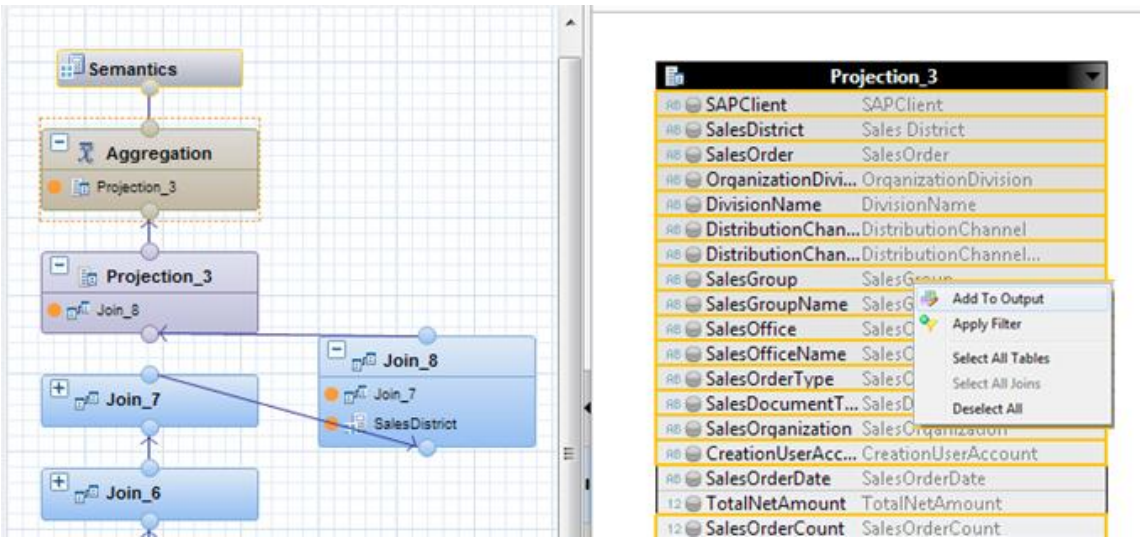



**Figure 17: Aggregation: Add necessary columns to output**

22. In the Semantic node, mark the columns as attributes or measures

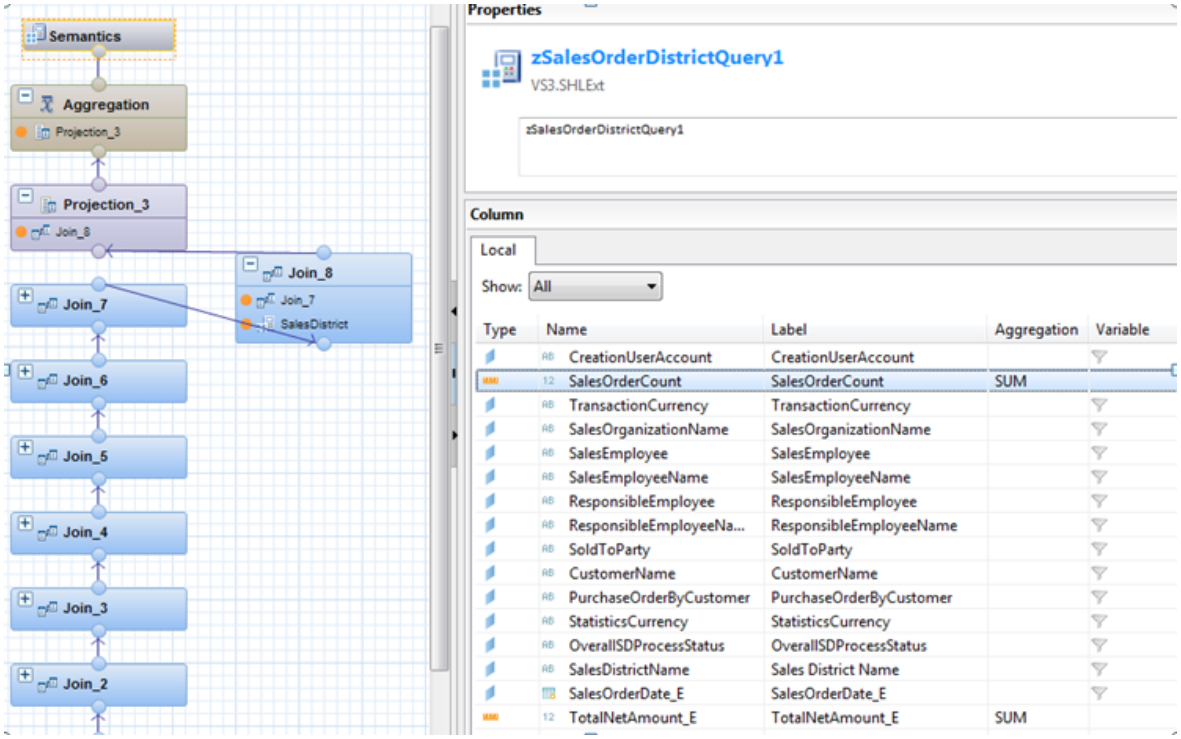



**Figure 18: Semantic: Mark attributes and measures**

## 4.4 Activate and Test the view

23. Save and activate the query view
24. Test the ‘zSalesOrderDistrictQuery1’ view is working as intended by creating an appropriate SQL Command to retrieve a few columns from the view, including the newly added SalesDistrict and SalesDistrictName columns.

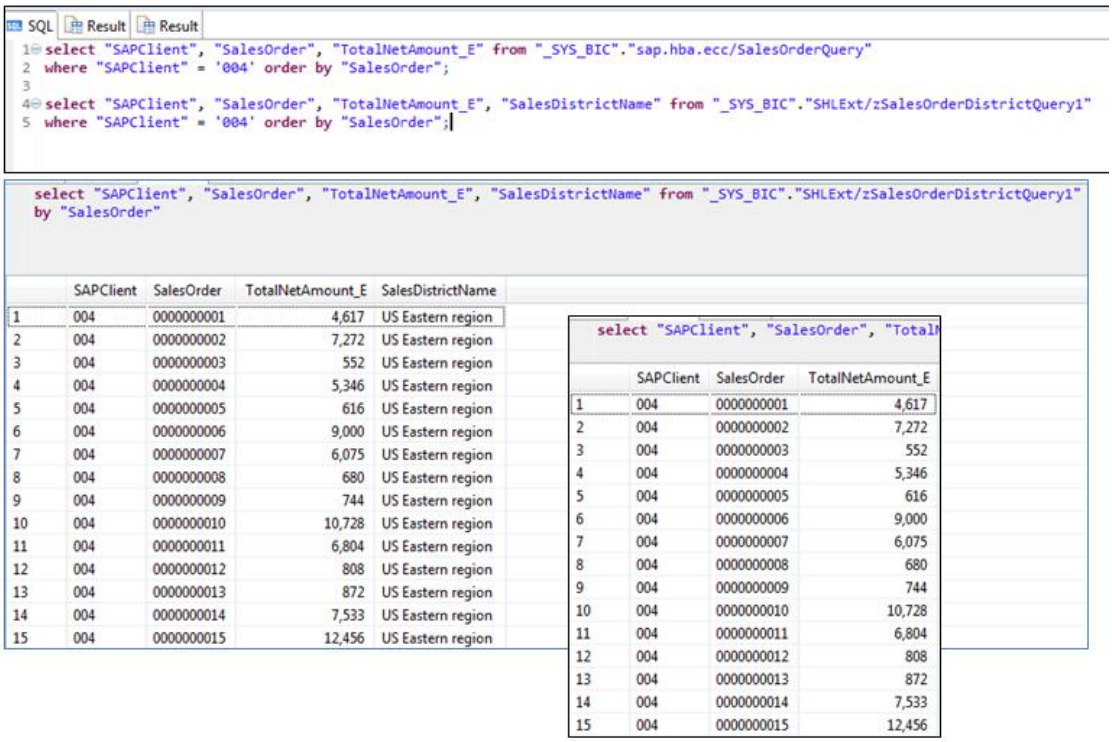



**Figure 19: Test result including SalesDistrict and SalesDistrictName**

# 5. Exclude unnecessary fields from Query view – Step-by-Step Procedure

Standard SAP Hana Live content contains query views which are designed as a reference for creating customer views for reporting. Depending on requirements, fields can be included or excluded by copying and modifying query views into the customer package and namespace. The example below details how unnecessary fields can be excluded from a view. It is always good practice to only expose the fields that are really needed for the reporting purpose.

In the sales order query view, there are some exposed fields such as SalesGroup and SalesOffice that the customer does not want to report on.

## 5.1 Identify unnecessary fields

Copy the zSalesOrderDistrictQuery query view and call it zSalesOrderReducedQuery, then review the view, and look at the output and identify the fields that are unnecessary for the report. For the example, SalesGroup and SalesOffice are not needed.

## 5.2 Exclude the unnecessary fields from the query view

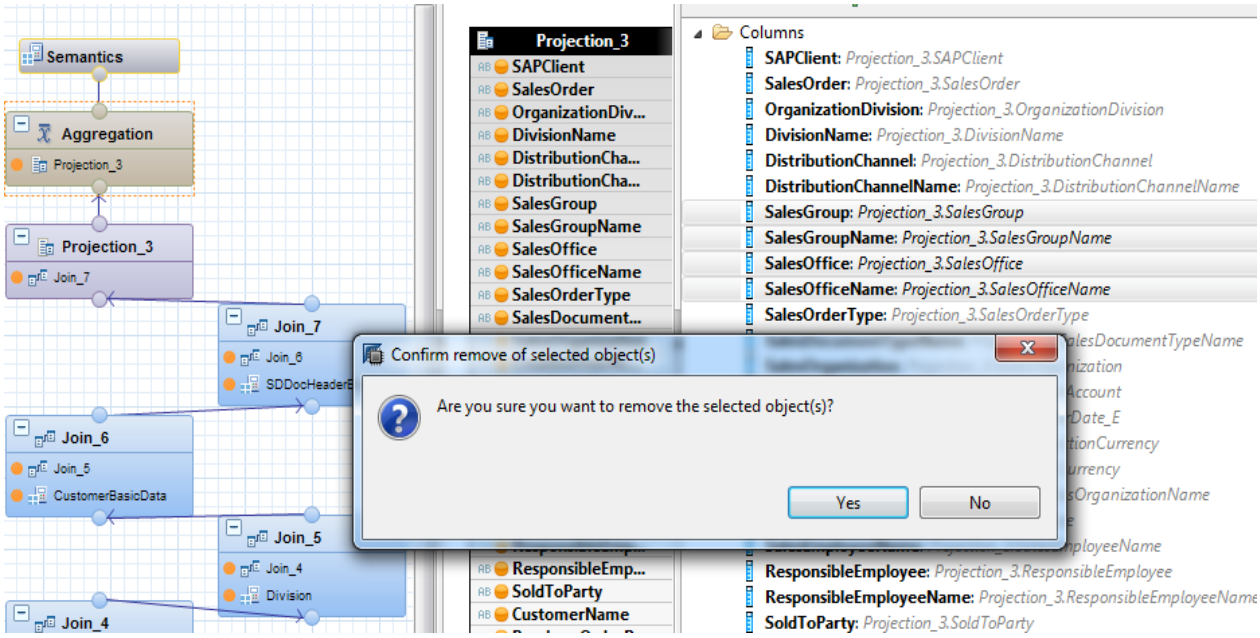



**Figure 20: Removing unnecessary fields from Query View**

1. Select the Aggregation node, then select the 4 unnecessary fields from the Output section and press <Delete>, click on 'Yes' and the fields are removed from the Query.
2. This still leaves these fields in nodes further down, but since they cannot be selected from any reporting tool that uses the view, and all the joins are Left Outer Joins, they should have no negative impact on performance.

## 5.3 Test the view

3. Save and activate the query view

4. Test the view using the SQL console to make sure that the 'zSalesOrderReducedQuery' view returns only the fields that were left in the output, and that the removed fields are not available.

SQL | Result

```
Select top 100 *
from "_SYS_BIC"."SHLExt/zSalesOrderReducedQuery"
where "SAPClient" = '004'
order by "SalesOrder"
```

| | SAPClient | SalesOrder | OrganizationDivision | DivisionName | DistributionChannel | DistributionChannelName | SalesOrderType | SalesDocumentTypeName |
|---|---|---|---|---|---|---|---|---|
| 1 | 004 | 0000000001 | 00 | Cross-divisioN | 30 | Internet Sales | TA | Standard Order |
| 2 | 004 | 0000000002 | 00 | Cross-divisioN | 30 | Internet Sales | TA | Standard Order |
| 3 | 004 | 0000000003 | 00 | Cross-divisioN | 30 | Internet Sales | TA | Standard Order |
| 4 | 004 | 0000000004 | 00 | Cross-divisioN | 30 | Internet Sales | TA | Standard Order |
| 5 | 004 | 0000000005 | 00 | Cross-divisioN | 30 | Internet Sales | TA | Standard Order |
| 6 | 004 | 0000000006 | 00 | Cross-divisioN | 30 | Internet Sales | TA | Standard Order |
| 7 | 004 | 0000000007 | 00 | Cross-divisioN | 30 | Internet Sales | TA | Standard Order |
| 8 | 004 | 0000000008 | 00 | Cross-divisioN | 30 | Internet Sales | TA | Standard Order |
| 9 | 004 | 0000000009 | 00 | Cross-divisioN | 30 | Internet Sales | TA | Standard Order |
| 10 | 004 | 0000000010 | 00 | Cross-divisioN | 30 | Internet Sales | TA | Standard Order |
| 11 | 004 | 0000000011 | 00 | Cross-divisioN | 30 | Internet Sales | TA | Standard Order |
| 12 | 004 | 0000000012 | 00 | Cross-divisioN | 30 | Internet Sales | TA | Standard Order |
| 13 | 004 | 0000000013 | 00 | Cross-divisioN | 30 | Internet Sales | TA | Standard Order |
| 14 | 004 | 0000000014 | 00 | Cross-divisioN | 30 | Internet Sales | TA | Standard Order |
| 15 | 004 | 0000000015 | 00 | Cross-divisioN | 30 | Internet Sales | TA | Standard Order |

**Figure 21: Test results after the modifications**

# 6. Include Custom Table(s) - Step-by-Step Procedure

For this scenario the customer wishes to classify countries by using one of three classifications, where the classifications consist of 'Developed', 'Transitional', and 'Emerging'. Since the SalesOrderQuery view can be used to expose the country code, the requirement can be satisfied by using a custom table that contains the Country Code and the Classification, and joining that to a standard view that exposes the Country Name, then adding the resulting custom view into the query.

## 6.1 Create a Custom Reuse View

In order to enhance the existing Country reuse view to use the custom country classification table, the Country View needs to be copied and renamed into the customer namespace and customer package, and then modified.

1. Create a custom reuse view in the custom package and name it zCountryClassification.
2. Edit the newly created reuse view zCountryClassification and display the properties and ensure that the 'Multidimensional Reporting' is set to 'disabled' and 'Execute in' is 'set to 'SQL Engine'.
3. Locate the 'Country' reuse view in the package sap → hba → ecc, then drag and drop it onto the reuse view zCountryClassification, and add a Projection node and project all the fields from the 'Country' view.
4. Drag and drop the custom table onto the view, add a Projection node, then project all the fields from the custom table.
5. Create a Join, add the Country Projection and the Classification Projection then create a left outer join between the relevant key fields, in this case being the Country field. Rename the Classification field to a user friendly name such as CountryClassification.
6. Link the Join to the Projection node, and add the necessary fields to the output.
7. In the Semantics node, mark all the columns as attributes

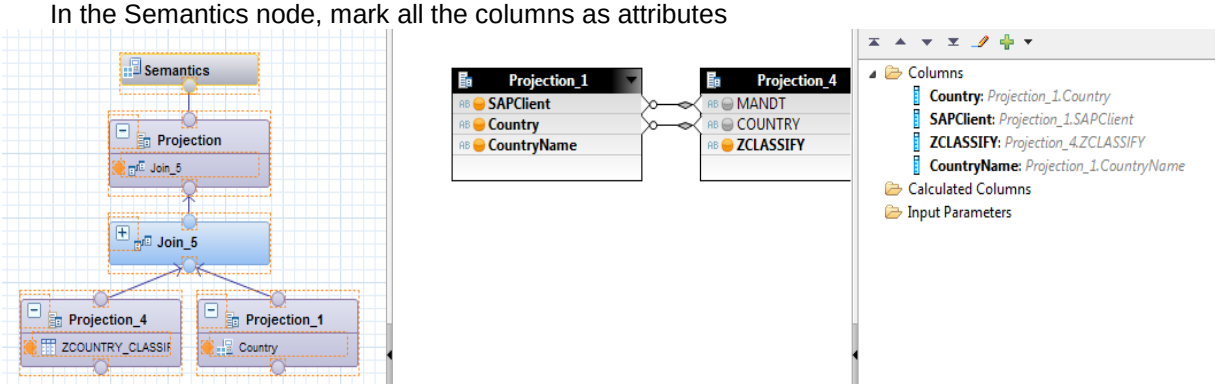



**Figure 22: Completed custom reuse view**

8. Activate and test the view using the SQL console

SQL | Result

```
select * from "_SYS_BIC"."SHLExt/zCountryClassification"
where "SAPClient" = '004'
```

| | SAPClient | Country | CountryName | CountryClassification |
|---|---|---|---|---|
| 1 | 004 | AE | Utd.Arab Emir. | Transitional |
| 2 | 004 | AR | Argentina | Transitional |
| 3 | 004 | AT | Austria | Transitional |
| 4 | 004 | AU | Australia | Transitional |
| 5 | 004 | BE | Belgium | Transitional |
| 6 | 004 | BR | Brazil | Developed |
| 7 | 004 | CA | Canada | Developed |
| 8 | 004 | CH | Switzerland | Emerging |
| 9 | 004 | CL | Chile | Transitional |
| 10 | 004 | CN | China | Emerging |
| 11 | 004 | CO | Colombia | Emerging |
| 12 | 004 | CU | Cuba | Developed |
| 13 | 004 | CZ | Czech Repu... | Transitional |

**Figure 23: Test results from Reuse View**

## 6.2 Create a new Query View

9. In order to expose the Country field and to add in the new Reuse View, copy the existing SalesOrderQuery view to the customer namespace and custom package – SHLExt.zSalesOrderClassificationQuery.
10. Edit the copied Query View and locate the node in the view that contains the Country field. For this Query View, it is the reuse view CustomerBasicData.

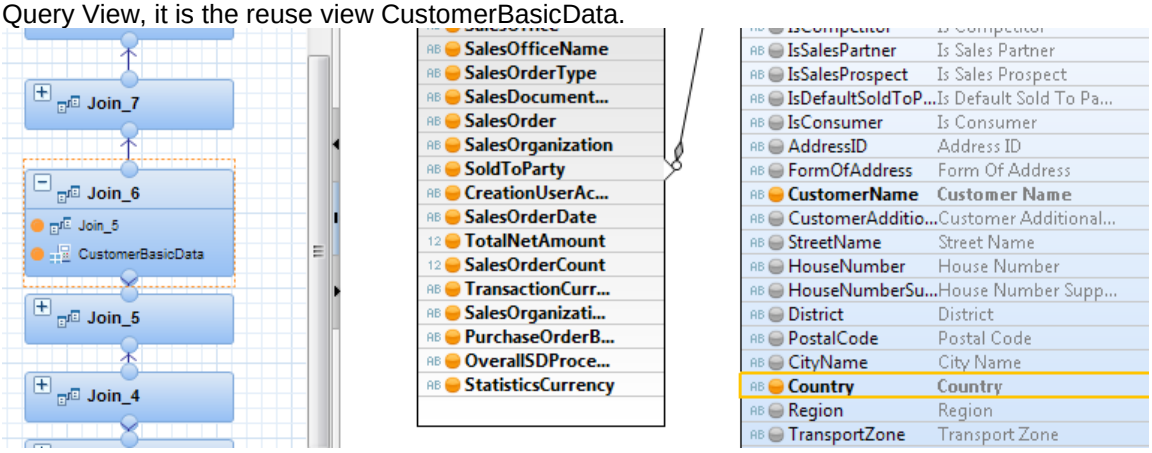



**Figure 164: Reuse view containing Country field**

11. Edit the Join_6 node to expose the Country field as an attribute, and then modify all the nodes above that node to ensure the field is exposed right up to the Output for the Query View.
12. Now break the link between the last Join node and the top Projection node, and add in a new Join node and connect it to the nodes above and below. Rename the new join to be consistent with the existing Join nodes.

13. Drag and drop the previously created Reuse View – zCountryClassification – beside the new Join node, and create a Left Outer join and add all the appropriate attributes to the output.

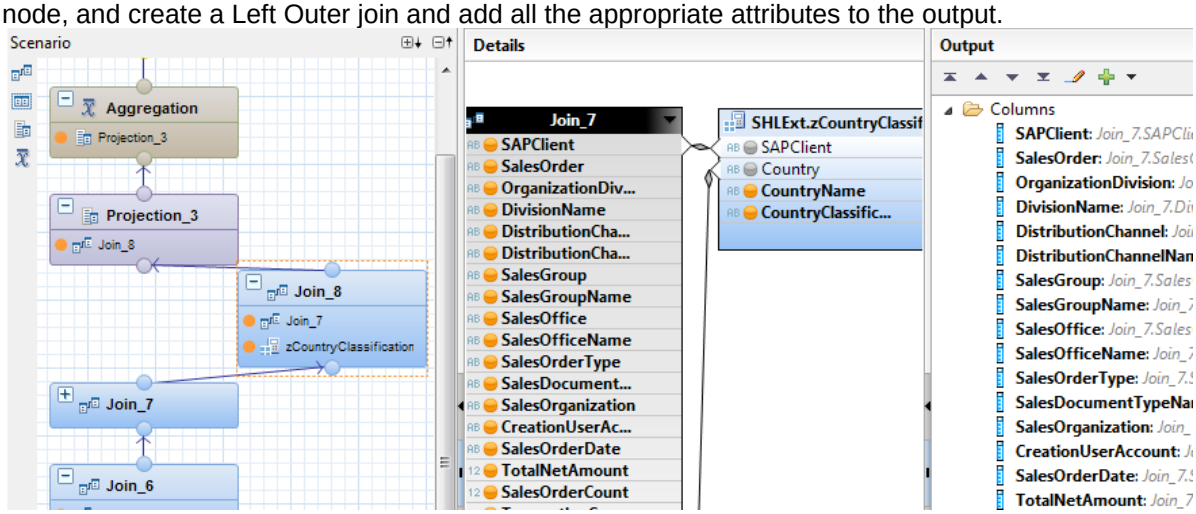



**Figure 25: New join and exposed fields**

14. Edit the top Output node and remove all the Output and Calculated Columns.
15. Edit the Projection in order to recreate the attributes and Calculated Columns. To do this, remove the two calculated columns from the output of the view, then add the columns again and recreate the Calculated Columns, by referring to the original Query View – SalesOrderQuery Projection node.
16. In the Aggregation node, add the necessary columns to the output
17. In the semantics node, mark columns and mark them as attributes or measures
18. Activate and test the view using the SQL console

SQL | Result

```
select sum("TotalNetAmount_E"), sum("NumberOfSalesOrders"), "Country", "CountryName",
"CountryClassification" from "_SYS_BIC"."SHLExt/zSalesOrderClassificationQuery"
where "SAPClient" = '004' group by "Country", "CountryName", "CountryClassification"
order by "Country"
```

| | SUM(TotalNetAmount_E) | SUM(NumberOfSalesOrders) | Country | CountryName | CountryClassification |
|---|---|---|---|---|---|
| 1 | 8,681,887.5 | 133 | AE | Utd.Arab Emir. | Transitional |
| 2 | 1,200 | 3 | AR | Argentina | Transitional |
| 3 | 654,836 | 12 | AT | Austria | Transitional |
| 4 | 56 | 2 | AU | Australia | Transitional |
| 5 | 1,000 | 1 | BR | Brazil | Developed |
| 6 | 32,230.87 | 5 | CA | Canada | Developed |
| 7 | 38,135.5 | 16 | CH | Switzerland | Emerging |
| 8 | 2,975 | 1 | CU | Cuba | Developed |
| 9 | 510,420,010.53 | 26,783 | DE | Germany | Transitional |
| 10 | 337,196 | 35 | EG | Egypt | Emerging |
| 11 | 164,255.55 | 21 | FR | France | Transitional |

**Figure 26: Test results for the new query view**

# 7. Include Custom Field(s) - Step-by-Step Procedure

When one or more custom fields are added to a standard table that is used in the SAP HANA Live Models and those custom fields need to be added to the Models, then the most straight-forward way of doing this is to create a custom Calculation View that exposes the custom fields and Master Data (if required) and then create a copy of the Query Calculation View and add in the newly created Calculation View.

## 7.1 Create a new Calculation View for the custom field(s)

1. Create a new Graphical Calculation View in the customer specific package with an appropriate unique name that conforms to your custom object naming convention.
2. Locate the table with the custom field(s) in the appropriate schema and drag and drop it into the new view.
3. Create a Projection node and link the table to the node, then select the key fields that are required to connect the table contents to the Query View as well as the custom field(s) and add them to the output.

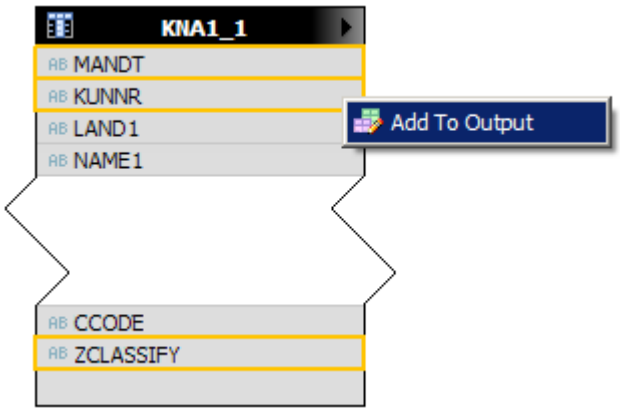



**Figure 27: Add Fields to Projection**

4. If there is a requirement for Master Data to be added for the custom field, then drag and drop the appropriate Master Data table onto the view.
5. Create a Join node and connect the Master Data table and the projection and join on the appropriate fields. The join type should be set to “Left Outer” and the cardinality should be set to “n..1”.

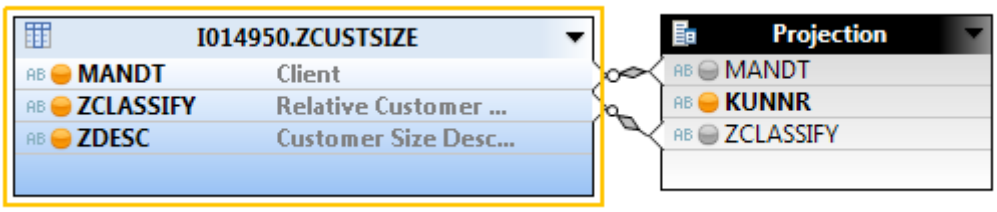



**Figure 28: Custom Field and Master Data Join**

6. Connect to the Projection node, and add all the required fields to Output, renaming the Columns as appropriate to User understandable terminology.
7. In the semantics node, mark the columns and set them as attributes

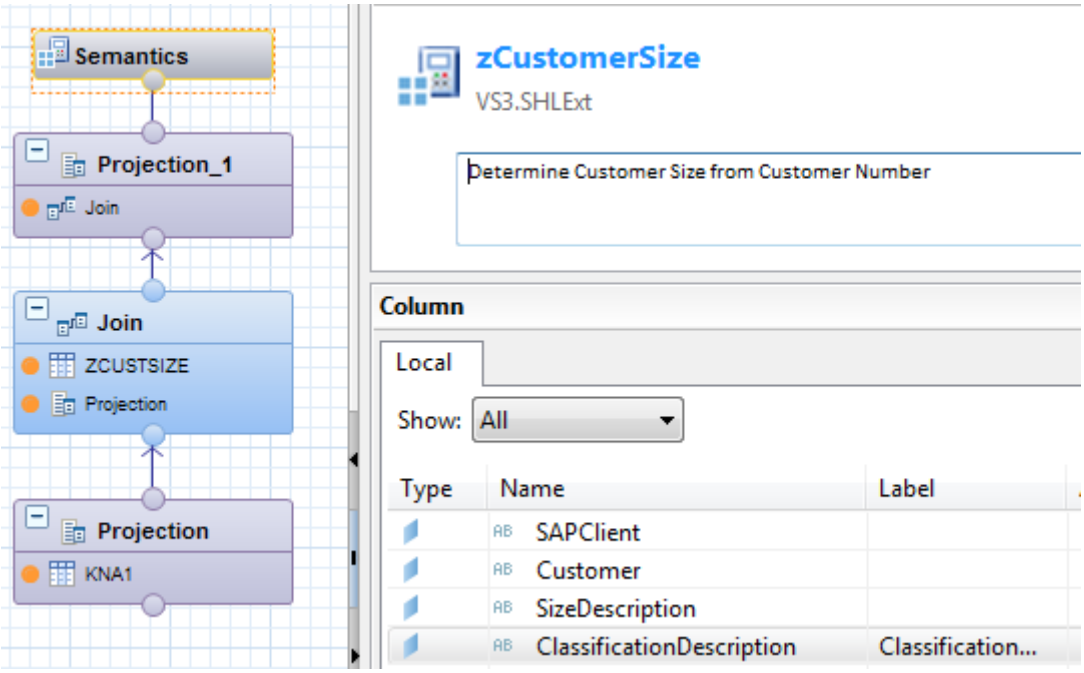
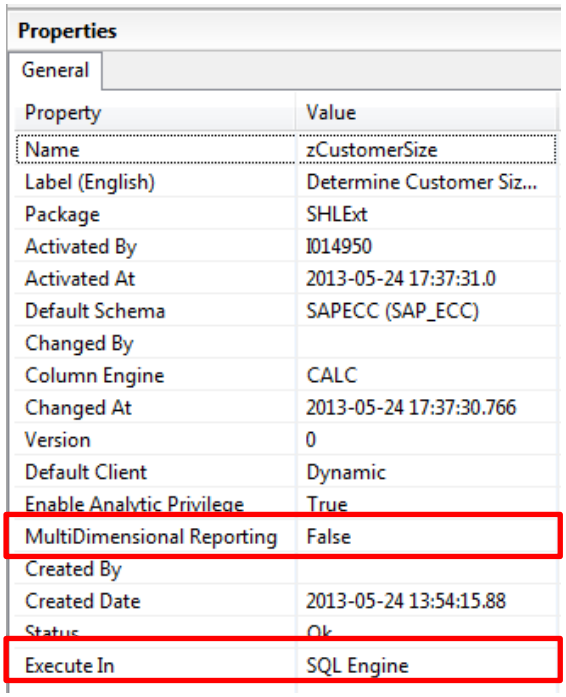



**Figure 29: After all the columns are added**

8. Display the properties of the new Calculation view by left-clicking in the checked background of the view, then selecting the "Properties" tab. Make sure that the property "Multidimensional Reporting" is set to "false" to prevent this view from being used by a client reporting tool, and also make sure that the property "Execute in" is set to "SQL Engine" which will normally boost the view performance.



Properties

General

| Property | Value |
|---|---|
| Name | zCustomerSize |
| Label (English) | Determine Customer Siz... |
| Package | SHLExt |
| Activated By | I014950 |
| Activated At | 2013-05-24 17:37:31.0 |
| Default Schema | SAPECC (SAP_ECC) |
| Changed By | |
| Column Engine | CALC |
| Changed At | 2013-05-24 17:37:30.766 |
| Version | 0 |
| Default Client | Dynamic |
| Enable Analytic Privilege | True |
| MultiDimensional Reporting | False |
| Created By | |
| Created Date | 2013-05-24 13:54:15.88 |
| Status | Ok |
| Execute In | SQL Engine |

**Figure 30: Check the View Properties**

9. Save and activate the view.

## 7.2 Locate and copy the appropriate Query view that needs to be extended

1. Locate the standard SAP HANA Live Query Calculation View in the SAP HANA Studio. The package that contains the view should exist in the appropriate package under **sap.hba**. This example will use the SalesOrderQuery Calculation View which can be found in the package **sap.hba.ecc**.
2. In order to keep Customer developments separate from the SAP HANA Live delivered content, the development objects should be added to a customer specific package with an appropriate unique name that conforms to your custom object naming convention.
3. To copy the original Calculation View, right-click on the view name and select **Copy**.

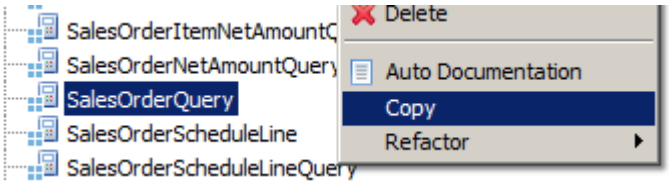



**Figure 31: Copy the Query View**

4. Navigate to the package where you are going to write the copy, right-click on the package name, and select **Paste**.

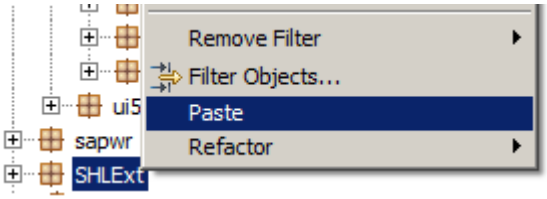



**Figure 32: Paste the Custom View**

5. In the dialog box, enter an appropriate new name of the custom view.

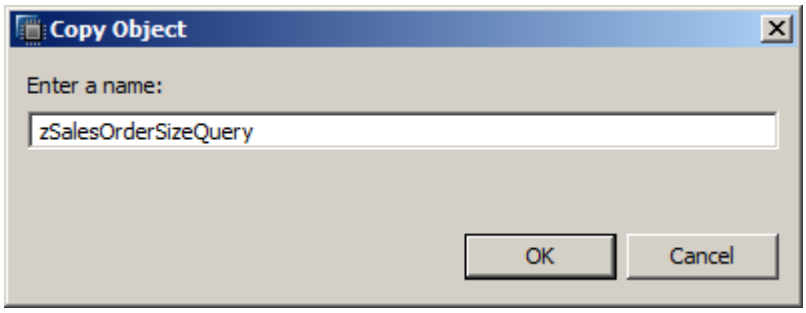



**Figure 33: Enter the Custom View Name**

## 7.3 Enhance the copied view to include the newly created custom view

1. Open the copied view in the editor.

2. Locate the appropriate place to join the new Custom view to the existing model. This needs to be where the keys required for the new view to link into the existing model are exposed. In the example, this can be done at the top level.
3. The two fields that are required to join the Custom view to the Query view are SAPClient and SoldToParty. Both of these fields are exposed at the node Join_7.

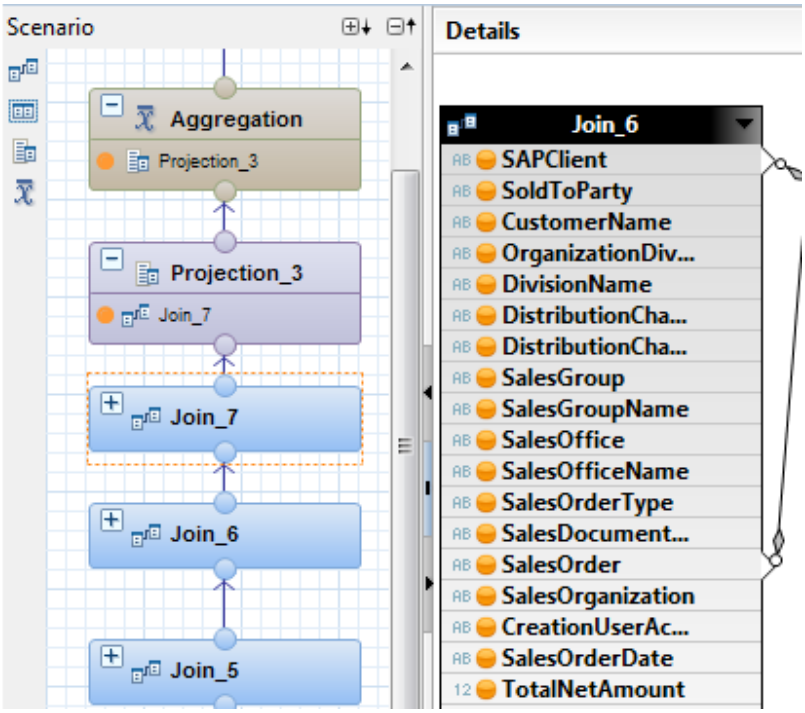



**Figure 34: Determine Node with Join Keys**

4. Delete the link between the Join_7 node and Projection node, then drag and drop the Custom view onto this view, and add in a Join node and ensure the new Join node is named ‘Join_8’.
5. Drag a connection from the Join_7 node to the new, Join_8 node, and add the Custom view to the Join_8 node.
6. Select the Join_8 node, and then in the ‘Details of Join_8’ panel, drag a join from the required key fields in Join_7 to the new view fields. The join type should be set to “Left Outer” and the cardinality should be set to “n..1”.

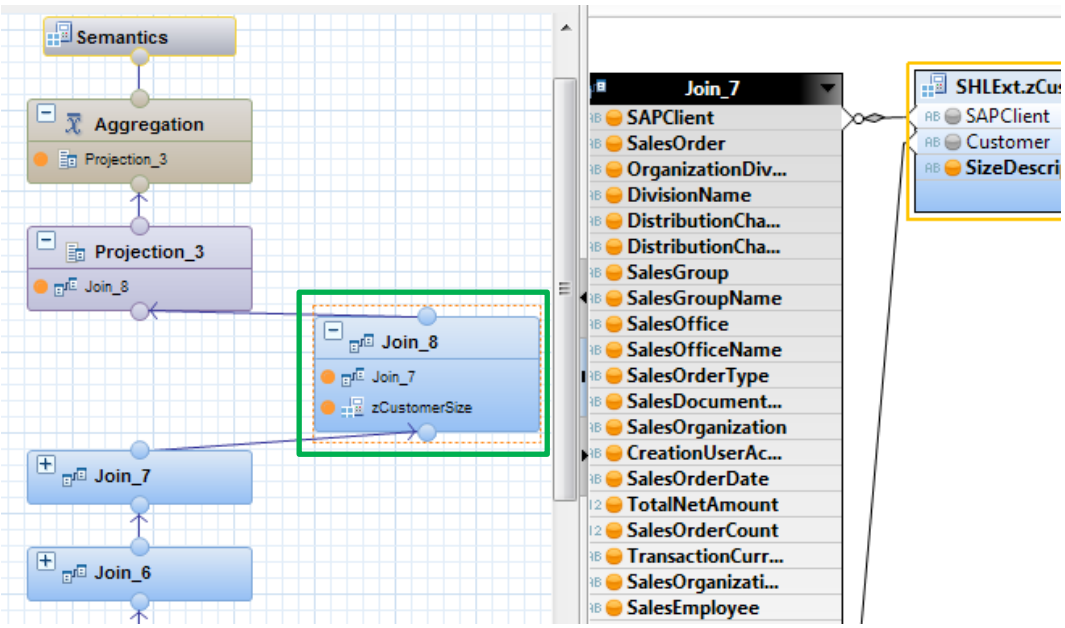



**Figure 35: New Node in Green Box**

7. Add in the connection from the Join_8 node to the Projection node. You now need to remove all the calculated columns from the Projection node. Now add all the required attributes and the newly added SizeDescription attribute to the Projection node making sure they appear in the Output of Projection pane as shown in Figure 9. Also recreate the calculation columns.

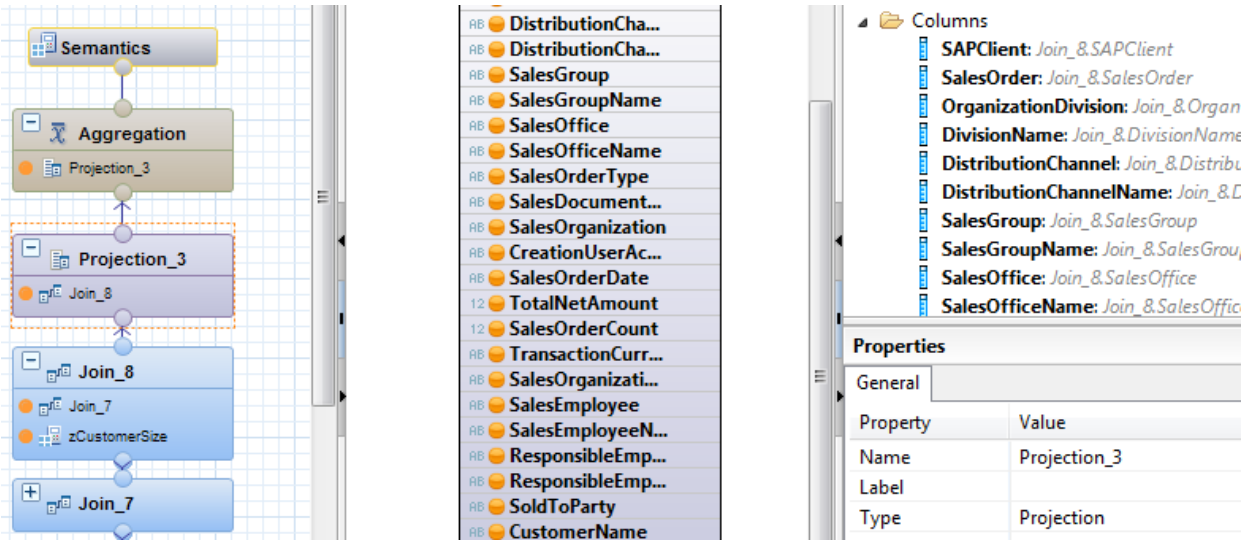



**Figure 36: New Attribute added to the Projection Node.**

8. In the Aggregation node, add all the necessary columns to the output.
9. In the Semantics node, mark the columns as attribute or measure.
10. Save and activate the view

## 7.4 Test the Modified Query View

1. Use a SQL command to test the output of the new Query View, ensuring that the new field(s) that were added are displaying correctly.

SQL | Result | Result

```
select "SoldToParty", "SizeDescription" from
"_SYS_BIC"."SHLExt/zSalesOrderSizeQuery" where "SAPClient" = '004' order
by "SoldToParty"
```

| | SoldToParty | SizeDescription |
|---|---|---|
| 1 | 0000000001 | Small |
| 2 | 0000000016 | Small |
| 3 | 0000000031 | Large |
| 4 | 0000000032 | Large |
| 5 | 0000000036 | Small |
| 6 | 0000000041 | Large |
| 7 | 0000000046 | Large |
| 8 | 0000000053 | Medium |
| 9 | 0000000055 | Medium |
| 10 | 0000000056 | Large |

**Figure 37: Test SQL Code and Output**

2. Compare the output from the new Query view to the output from the original Query view without including the newly added field(s) to ensure that the same results are being returned.

SQL | Result | Result

```
select "SoldToParty" from "_SYS_BIC"."sap.hba.ecc/SalesOrderQuery" where
"SAPClient" = '004' order by "SoldToParty"
```

| | SoldToParty |
|---|---|
| 1 | 0000000001 |
| 2 | 0000000016 |
| 3 | 0000000031 |
| 4 | 0000000032 |
| 5 | 0000000036 |
| 6 | 0000000041 |
| 7 | 0000000046 |
| 8 | 0000000053 |
| 9 | 0000000055 |
| 10 | 0000000056 |

**Figure 38: Matching Original Query Output**

3. If the results of the tests are positive, then congratulations, you have added new field(s) to a standard Query view.

www.sdn.sap.com/irj/sdn/howtoguides