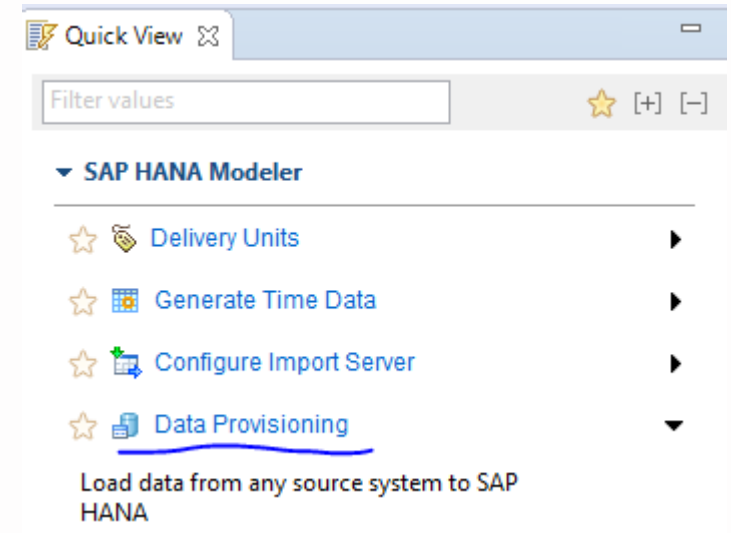


SAP HANA internal training – Session 6

March 2018

Data Provisioning

- Data Provisioning is a process of creating, preparing, and enabling a network to provide data to its user.
- Data needs to be loaded to SAP HANA before data reaches to the user via a front-end tool.
- Data Provisioning is done from SAP HANA Developer Perspective.
- Go to “Quick View” and select Data Provisioning:
- SAP HANA supports two type of Provisioning tool –
 - SAP HANA Built-In Provisioning Tool
 - Flat File
 - Smart Data Integration (SDI)
 - Smart Data Access (SDA)
 - Smart Data Streaming (SDS)



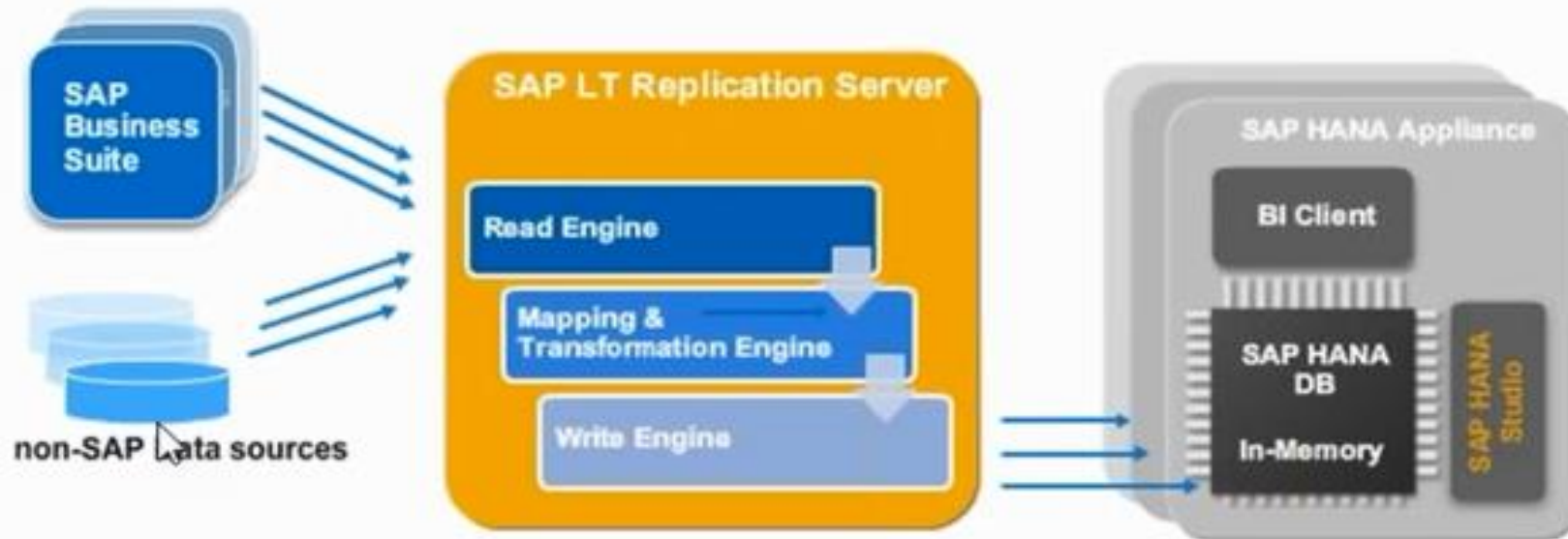
Data Provisioning

- External tool supported by SAP HANA
 - SAP Landscape Transformation (SLT)
 - SAP Business Objects Data Services (BODS)
 - SAP Direct Extractor Connection (DXC)
 - Sybase Replication Server (SRS)

Methods of Data Provisioning	Description
SLT	SLT ("SAP Landscape Transformation Replication Server") running on the SAP Net Weaver Platform. SLT is an ideal solution for Real-Time and Schedule time replication for SAP and non-SAP source system.
SAP Business Objects Data Services	SAP DATA Services is a platform for designing of ETL processes with a graphical user interface.
DXC	DXC stand for Direct Extractor Connect is a batch driven ETL tool.
Flat File Upload	This option used to Upload data (.csv, .xls, .xlsx) to SAP HANA.

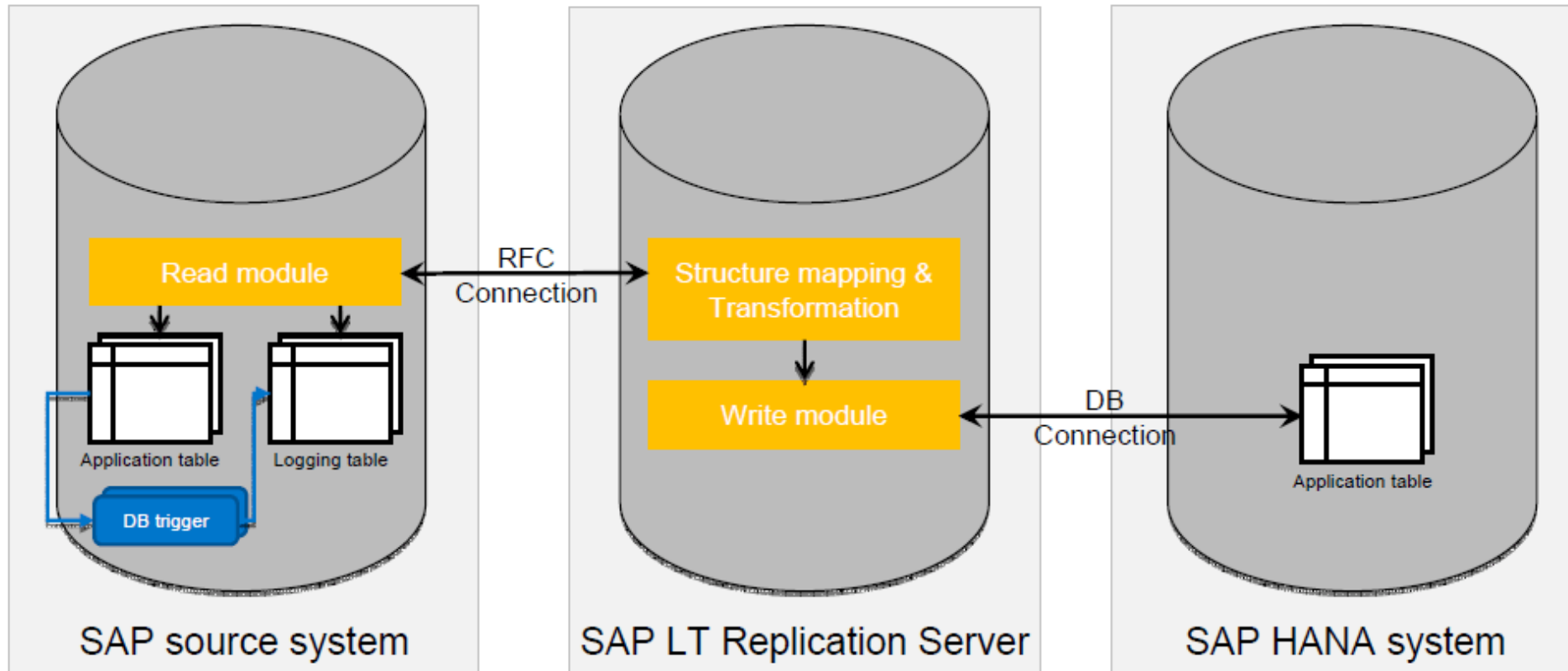
SLT - SAP Landscape Transformation

SAP Landscape Transformation Replication Server (aka "SLT") is the best choice for all SAP HANA customers who need real-time or scheduled data replication sourcing from SAP and NON-SAP sources with the option to accomplish even complex data transformations on the fly.



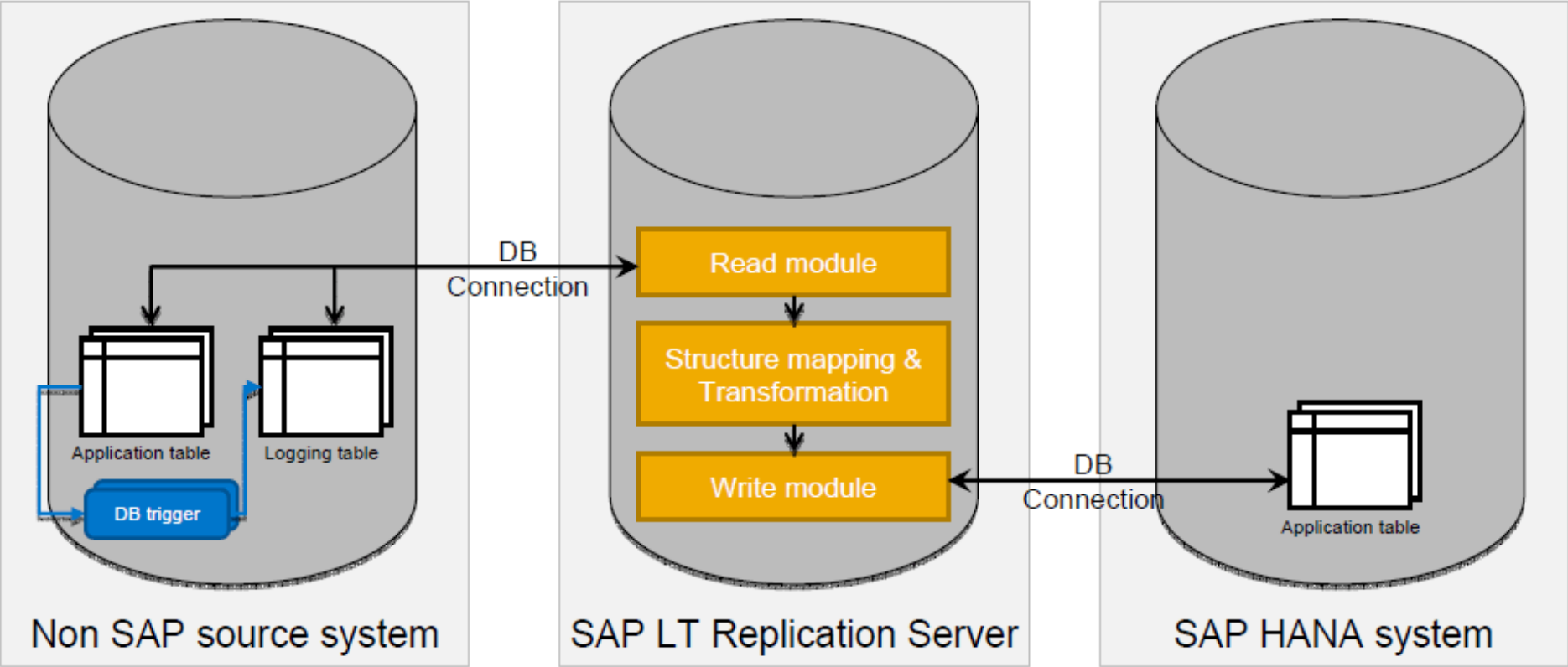
SLT - SAP Source

Read Module is within SAP source system



SLT - Non SAP Source

Read Module is in SAP SLT system



SAP BODS

SAP BODS is an ETL tool

The screenshot displays the SAP Data Services Designer interface for a data flow job named 'ZSFLIGHT_DataFlow'. The 'Project Area' on the left shows the job structure: 'ZSFLIGHT_Demo' > 'ZSFLIGHT_Job' > 'New_WorkFlow1' > 'ZSFLIGHT_DataFlow'. The 'Local Object Library' at the bottom left lists various transformation objects, with 'Map_CDC_Operation' highlighted. The main workspace shows a data flow diagram with a 'Query' object connected to a 'Map_CDC_Operation' object, which is then connected to a target table 'ZSFLIGHT_00_DEMO(SLT_Target_HANA.JASON_REG)'. The bottom right pane shows a preview of the data being loaded into the target table.

MANDT	CARRID	CONNID	FLDATE	PRICE
800	AA	17	2012.10.03	422.94
800	AA	17	2012.12.12	422.94
800	AA	17	2013.02.20	422.94
800	AA	17	2013.05.01	422.94
800	AA	17	2013.07.10	422.94
800	AA	17	2013.09.18	422.94
800	AA	17	2013.11.27	422.94
800	AZ	555	2012.10.03	185.00
800	AZ	555	2012.12.12	185.00
800	AZ	555	2013.02.20	185.00
800	AZ	555	2013.05.01	185.00
800	AZ	555	2013.07.10	185.00

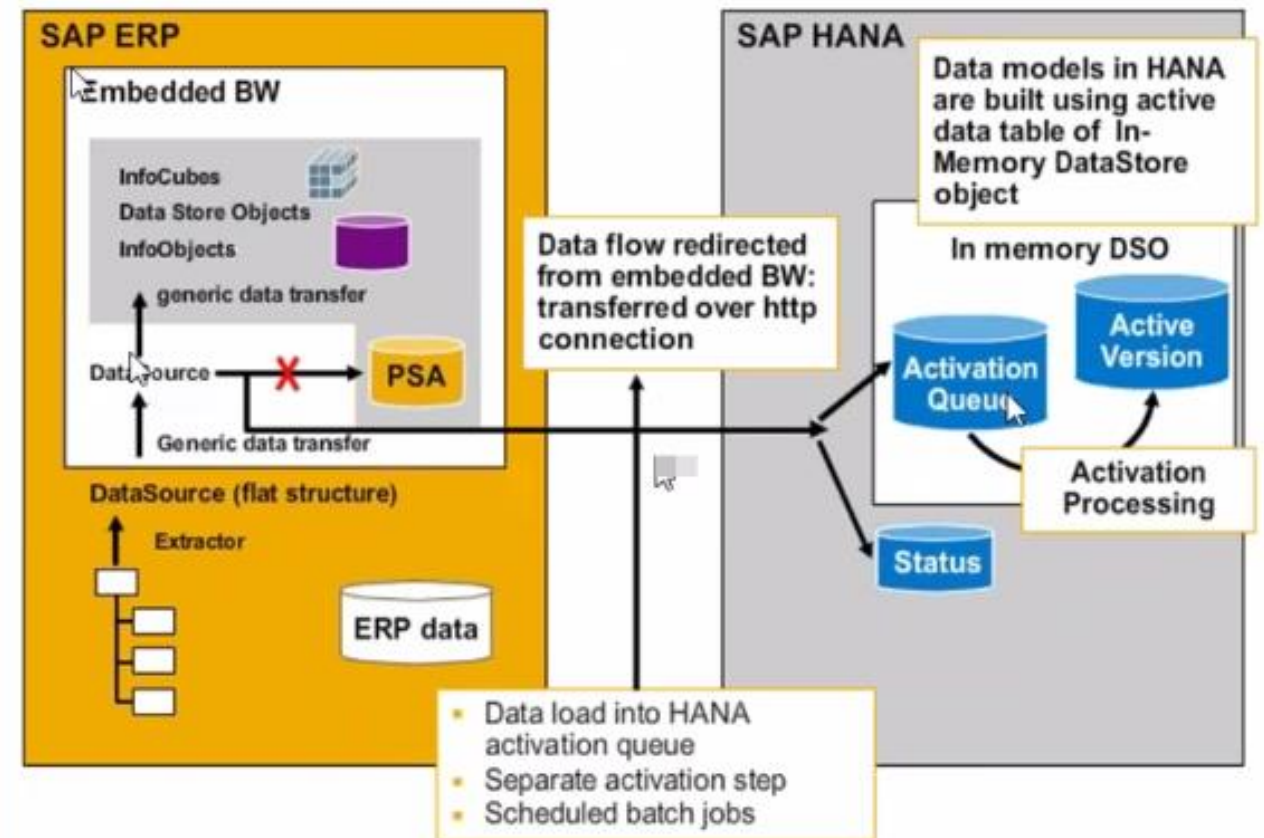
DXC

SAP HANA Direct Extractor Connection (DXC) is available as a simple option in ETL (batch) scenarios for data replication from existing SAP Data Source extractors into SAP HANA

Using DXC data can be extracted only from SAP business suite

DXC is best used in Sidecar scenario where data is loaded to HANA without disturbing existing BW flow

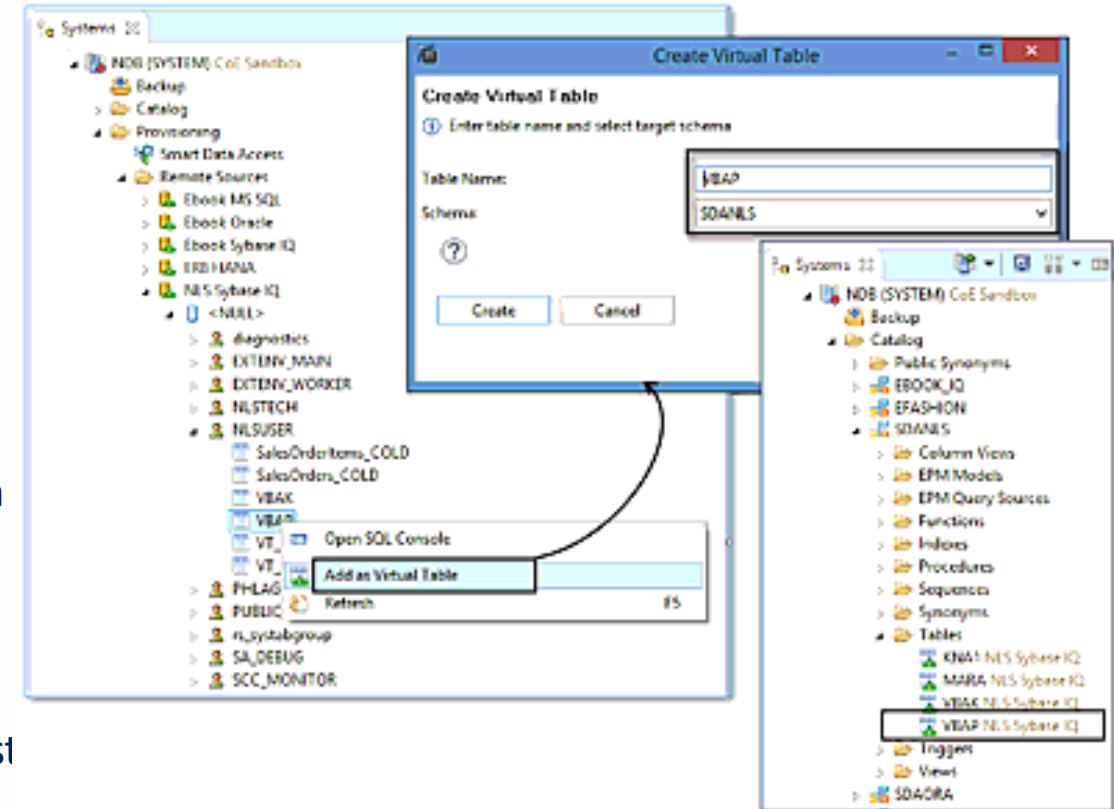
Overview – SAP HANA DXC Concept: Illustration



SAP HANA SDA

Smart Data Access – Introduction

- SDA was introduced in SAP HANA SPS06.
- SDA is method of SAP HANA for accessing the data stored in remote data sources. With the help of SDA, SAP HANA can create so-called “virtual table” mapping to tables located in remote data sources, and then SAP HANA can access the data directly by accessing the “virtual table”.
- “virtual table” can be manipulated by SAP HANA just like an ordinary table, which means the operations, such as select, update, insert, delete, and so on, are all available for “virtual table”. Besides, join operation between local table and “virtual table” is supported. When such join operation taken, optimizer of SAP HANA sends the relevant operations to remote data source for processing, and then the result set would be sent back to SAP HANA for further processing.
- SDA virtual tables can’t be transported between systems, so they must be explicitly created in each environment. Therefore, if you have to create virtual tables for multiple remote objects, it easier to do it using SQLScript.

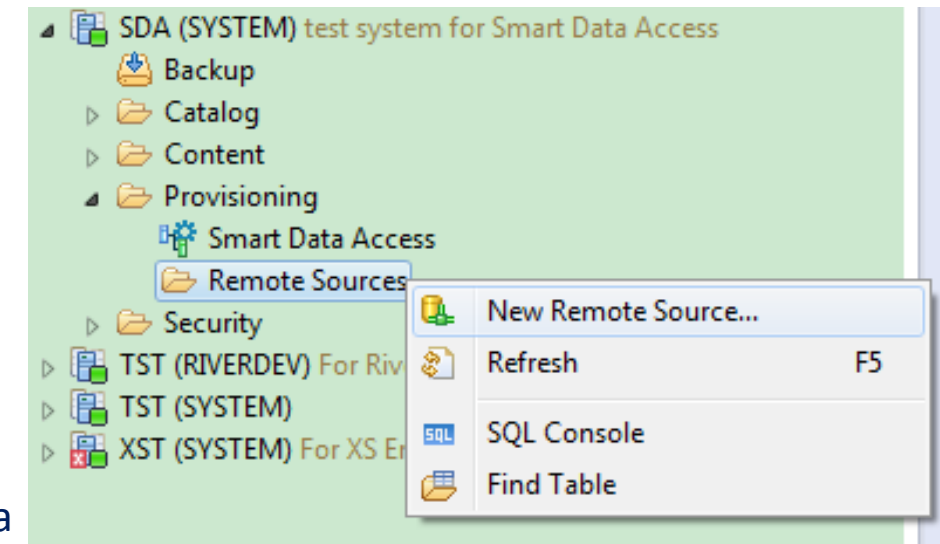


SAP HANA SDA

How to create remote system

Creating a remote data source in SAP HANA usually involves steps below :

1. Check whether SAP HANA provides specialized adapter for the data source, such as “ASEODBC”, “IQODBC”, “TDODBC”;
2. If specialized adapter is available, then just use it to create data source;
3. If specialized adapter is not available, then check whether there is a specialized property configuration template file, such as template for Oracle, MSSQL;
4. If specialized property configuration template exists, you can change the property configuration file according to your requirement, and then create data source using the modified file. For example, as long as the correctness of the modification ensured, you can disable the unnecessary functions, modify the mapping of data type or function based on your requirement.
5. If specialized property configuration template exists, you have to create a brand new property configuration file from scratch. To create such a file, you must be familiar with the properties of the data source and the driver it uses;
6. Create the data source in SAP HANA Studio using the specialized adapter or the common adapter (ie. ODBC adapter). When using common adapter, you need to specify the property configuration file for the data source.

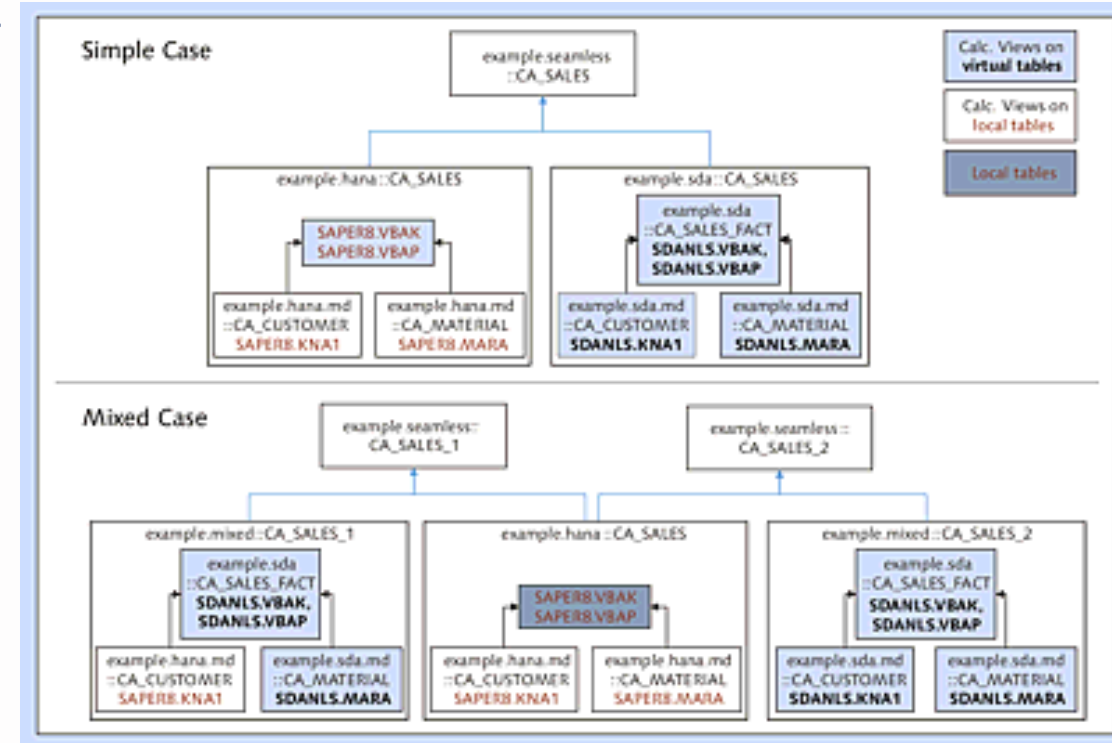


SAP HANA SDA

Smart Data Access – Best Practise

We can build SAP HANA information models on SDA virtual tables similar to how we would build local tables. However, there are differences between these processes that should be noted:

- Only calculation views are supported using SDA virtual tables. Attribute views and analytic views are not supported.
- One or more virtual tables must be isolated into one or more calculation views. These calculation views must not have SAP HANA native objects (i.e., tables, analytic views, attribute views, etc.).
- Calculation views on virtual tables must be set to execute in the SQL engine.
- One or more different calculation views may be created to combine calculation views on virtual tables with SAP HANA native objects (i.e., tables, analytic views, attribute views, etc.). These views do not need to be executed in the SQL engine.



Analytical Privileges

- Analytic privileges are used to allow read access to data in SAP HANA information models (that is, analytic views, attribute views, and calculation views) depending on certain values or combinations of values.
- Analytic privileges are evaluated during query processing.
- Analytic privileges granted to users in a particular database authorize access to information models in that database only.
- Analytic privileges grant different users access to different portions of data in the same view based on their business role.
- Within the definition of an analytic privilege, the conditions that control which data users see is either contained in an XML document or defined using SQL.

The screenshot shows the SAP HANA Analytic Privilege configuration window for '10648923_1::AP_SAMPLE_1 HBC (TESTID1)'. The interface is divided into several sections:

- General:** Describes general information about the Analytic Privilege. It includes fields for 'Name' (AP_SAMPLE_1) and 'Label' (AP_SAMPLE_1). There is a checkbox for 'Applicable to all information models' which is currently unchecked.
- Secured Models:** Restrictions apply to all the models shown in the list below. It shows a tree view with 'Content' expanded, containing 'CA_SCRIPTEDVIEW (10648923_1)'. There are 'Add...' and 'Remove' buttons.
- Privilege Validity:** Defines the Privilege Validity. It contains a table with columns: Operator, Inclusion, From, and To.
- Associated Attributes Restrictions:** Select attributes to assign analytic privileges. It contains a table with columns: Attributes, Origin, Shared, and Description.
- Assign Restrictions:** A button at the bottom right of the Associated Attributes Restrictions section.

The bottom of the window shows a tabbed interface with 'Analytic Privilege' and 'Users/Roles' tabs.

Performance Analysis

Available tools

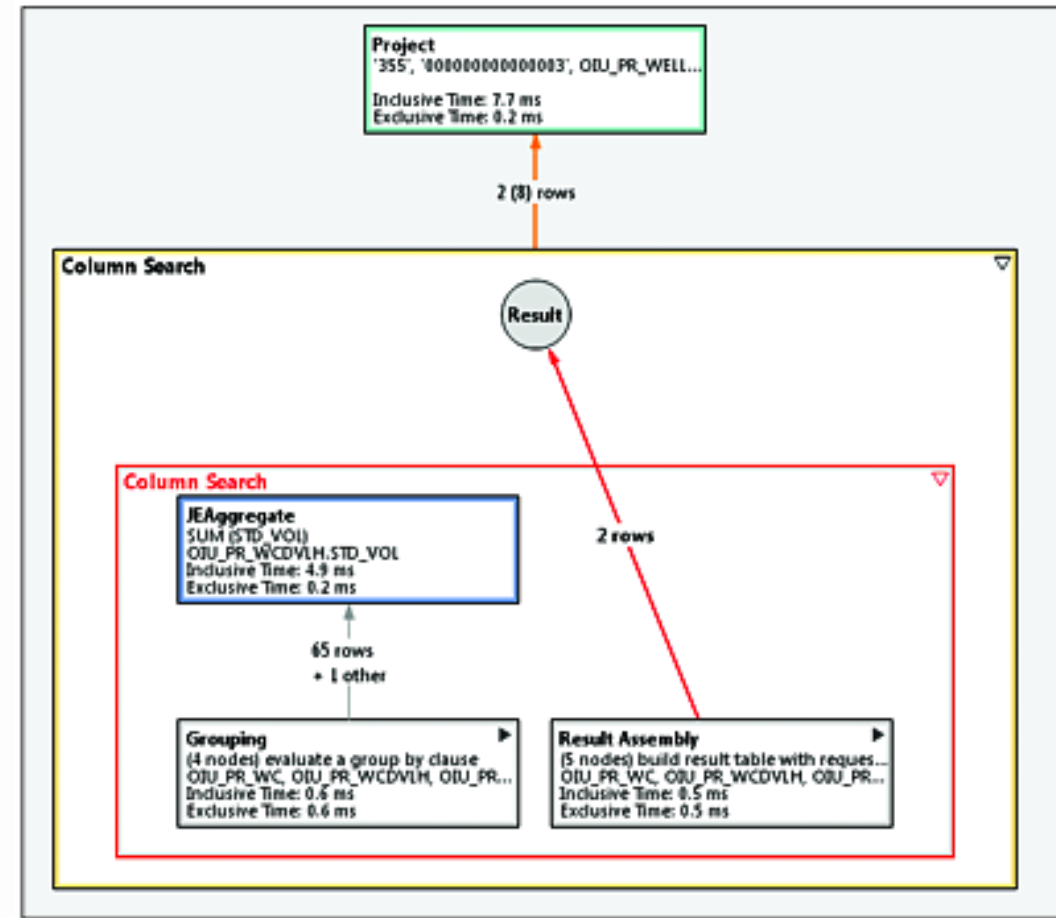
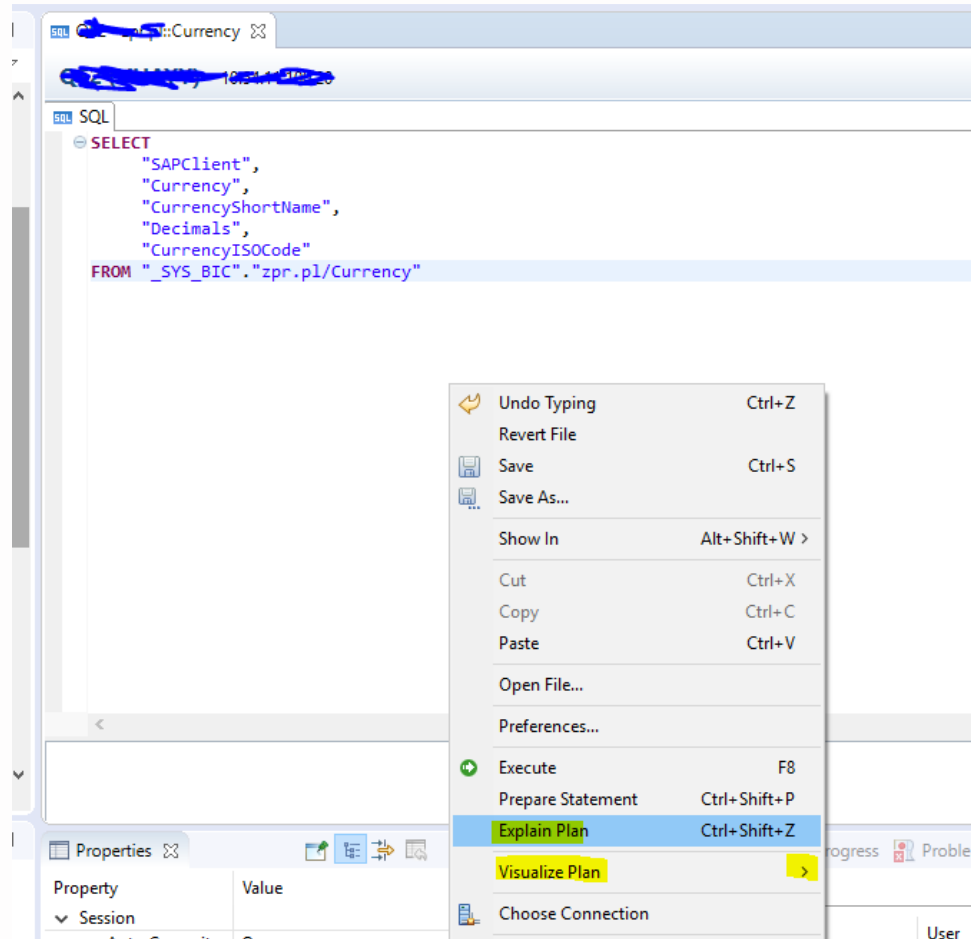
Below are details of some tools which help in monitoring performance of calculation views:

- Explain Plan tool : In this we can see which execution step uses which engine, through this we can avoid calling multiple engines by tweaking nodes in views, hence provides good performance.
- Plan Visualize tool: It gives the complete picture of how the data is flowing in a view, how much rows of data is moving from one node to another.
- Plan Execute tool : In this we can see execution time – which node is taking maximum time, how many tables were accessed, how many records, etc.
- Timeline View: This gives time stamps for each SQL execution step. It is also a good view to see which step is taking maximum time.

Performance Analysis

How to

To initialize the Plan/Visualize tool, open the generated SQL of the calculation view and right click in SQL editor. Tool opens up like below



Performance Analysis

Best Practices

- Validate Performance of Calculation Views - SAP HANA modeler provides you the following validation rules for performance validations. Execute these rules to identify the impact on the performance of the calculation view, and make corrections to the view accordingly.
 - Calculation in filter expression rule
 - Calculation in joins rule
 - Partition types in join rule.
- Use left outer joins/union maximum.
- Specify cardinality in joins (n:1 or 1:1) – only if sure
- Set optimize join = true (only if cardinality is set like above)
- Use table functions instead of scripted calculation views.
- Reduce the data set as early as possible. Use design time filters at the lowest level
- Using Filter is better than using Inner Join to limit the dataset.
- Avoid filters on calculated column (consider materializing these columns).

HANA Metadata

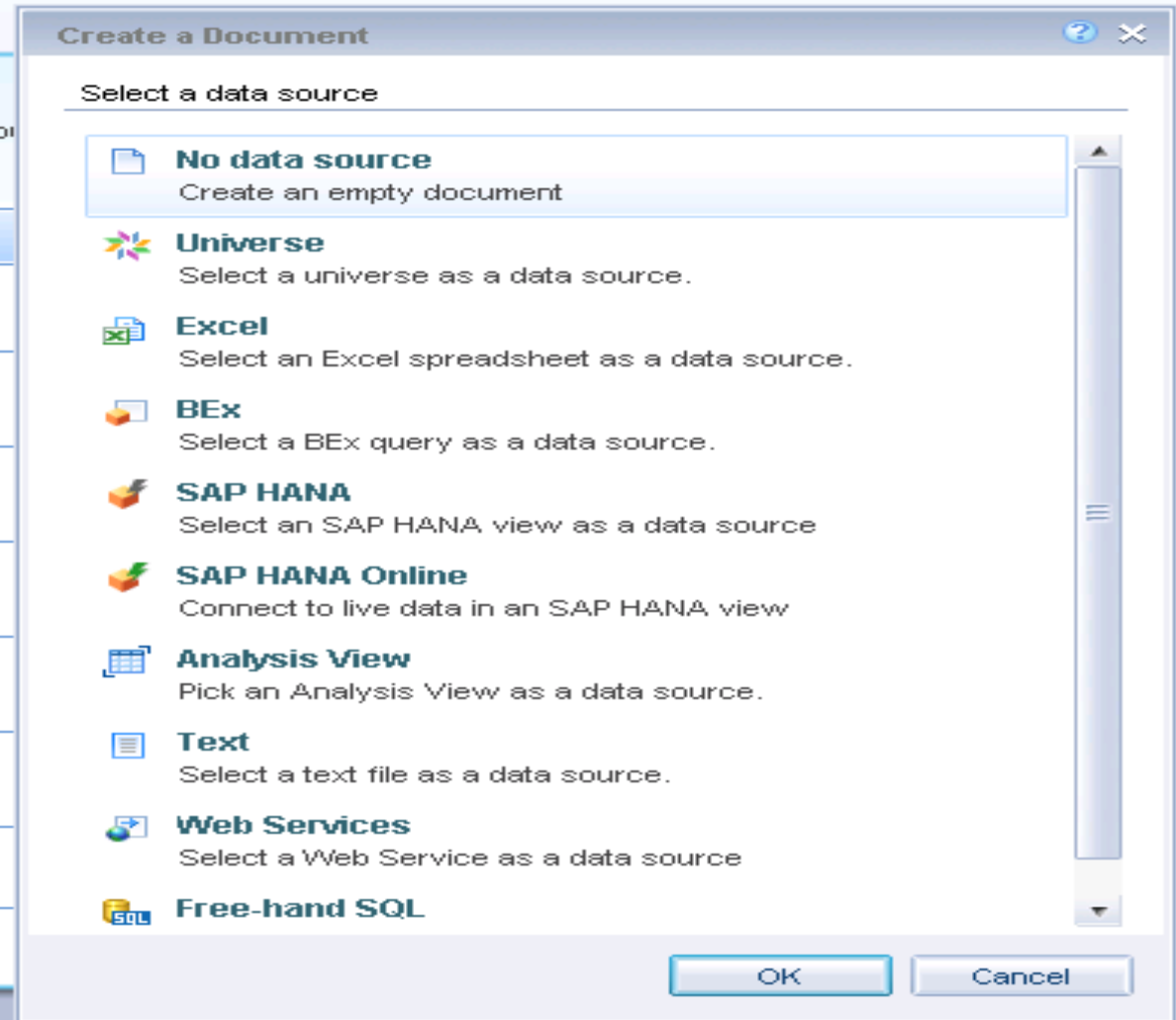
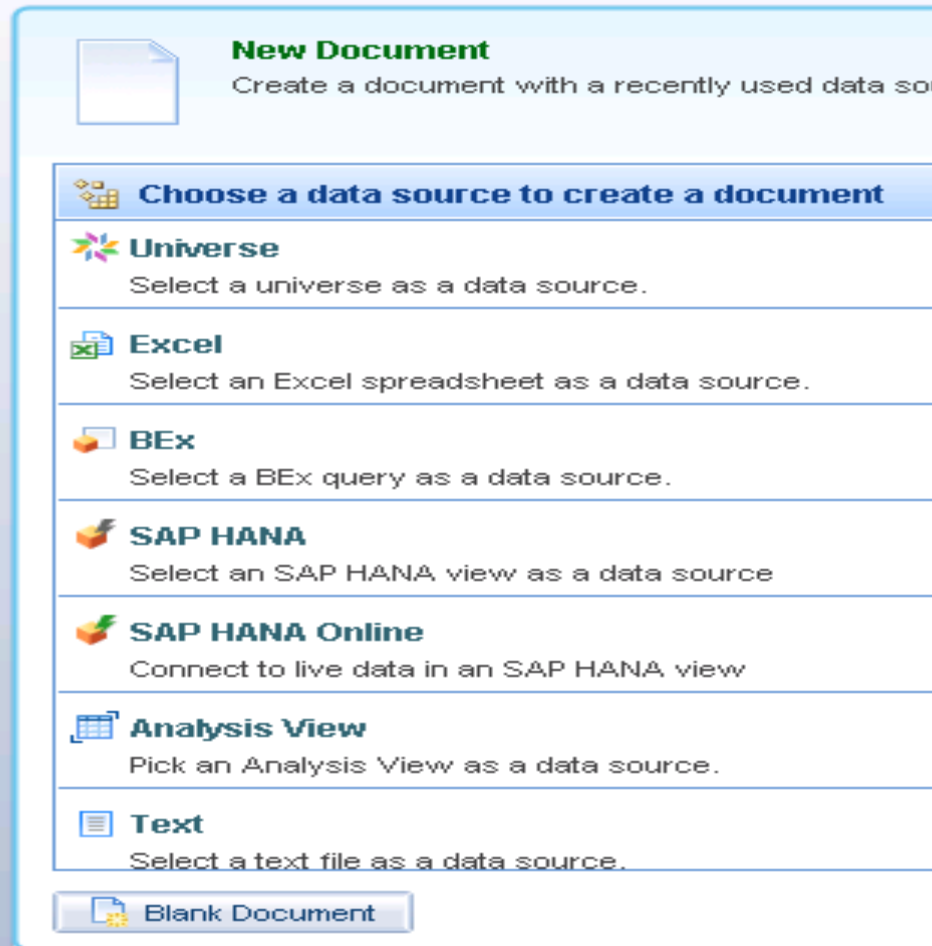
System views

System views allow to query for various information about the system state and metadata information. Some of the useful views are as below:

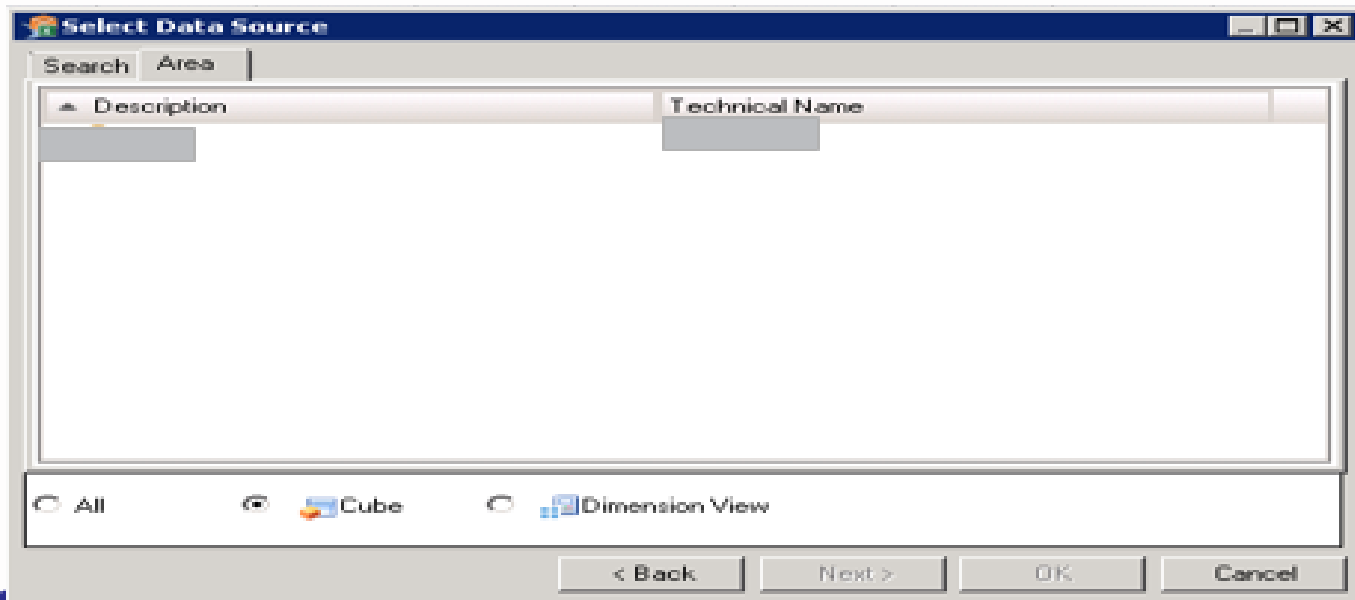
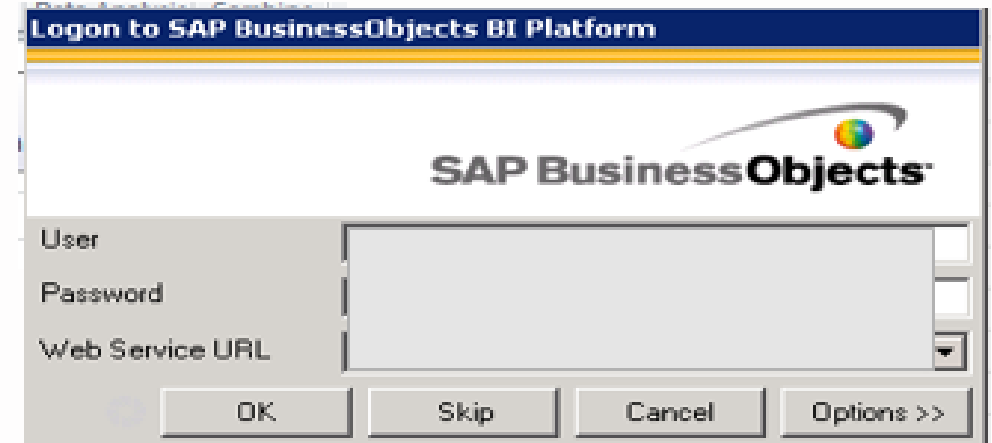
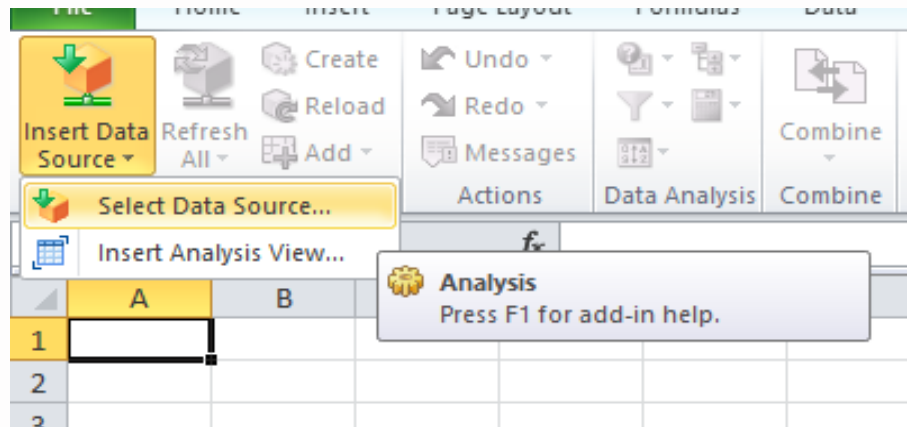
- CS_ALL_COLUMNS - Provides information from all columns of column tables, including internal ones.
- CS_JOIN_CONDITIONS - Provides join conditions for column store join views.
- CS_JOIN_TABLES - Provides information about the physical tables referred to by column store join views.
- CS_KEY_FIGURES - Provides information about the key figures defined for column store join views.
- CS_VIEW_COLUMNS - Provides information about the columns defined for column store join views.
- CS_VIEW_PARAMETERS - Provides a list of parameters of the objects in SAP HANA Database. For now, only calculation views are considered. The parameters of a view are parsed from the definition of the underlying scenario.
- EFFECTIVE_PRIVILEGES - Provides information about privileges of the specified user.
- INDEXES - Provides information about indexes on tables.
- M_CE_CALCVIEW_DEPENDENCIES - Provides all views that are referencing a Calculation Scenario.

Consumption of HANA Views in Reporting Tools

Using HANA views in WEBI



Using HANA views in SAP Analysis Office



Using HANA views in Lumira

Add new dataset

Select a Source:

- Microsoft Excel**
Load an Excel Worksheet as a dataset
- Text**
Load a text file (*.csv, *.txt, *.log, *.prn, *.tsv) as a dataset
- Copy from Clipboard**
Copy from Clipboard
- Connect to SAP HANA**
Connect to data in an SAP HANA View
- Download from SAP HANA**
Download data from an SAP HANA view as a dataset
- Universe**
Connect to a Universe and download a dataset
- Query with SQL**
Run freehand SQL on a database to download a dataset
- Connect to SAP Business Warehouse**
Connect to a BEx Query or an InfoProvider

All Recently Used:

- New Microsoft Office Excel Worksheet.xlsx
C:\Users\hana17\Desktop
- New Microsoft Office Excel Worksheet.xlsx
C:\Users\hana17\Desktop
- ANA_TEST2
best_SYS_BIC/H118AL
- ANV_SALES
best_SYS_BIC/IA_SANJU
- SALE_VIEW_ANA
best_SYS_BIC/pras2
- SAP HANA database 1.0
- TEST_ANA
best_SYS_BIC/prashanthi
- DEMP_ANA
best_SYS_BIC/pras2_sai
- SHOW_ANA
best_SYS_BIC/prashanthi

Previous Next Create

Using HANA views in Lumira

 **Add new dataset**

Select a SAP HANA View

AN_PROMOTION

AN_SALES

CA_SALES_FACT

Dataset Name:

EMPLOYEE_ANV

EMPLOYEE_ANV



Select Measures and Dimensions

☐ Show only selected

Measures (2)

<input checked="" type="checkbox"/>	Measure Name +	
<input checked="" type="checkbox"/>	CAL_ALVY_AMT	Sum
<input checked="" type="checkbox"/>	CAL_ALVY_MON	Count

Dimensions (11)

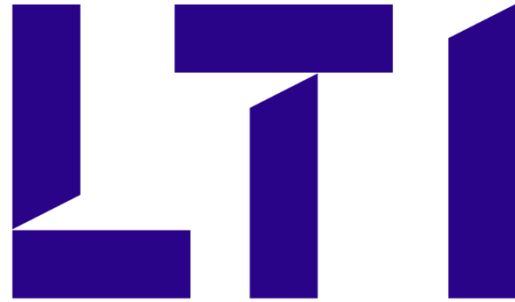
<input checked="" type="checkbox"/>	Dimension Name +		Values Preview
<input checked="" type="checkbox"/>	ABC CAL_ALV_FLAG		Click here to see sample values
<input checked="" type="checkbox"/>	123 CAL_BASIC		Click here to see sample values
<input checked="" type="checkbox"/>	123 CODE		Click here to see sample values
<input checked="" type="checkbox"/>	DOB		Click here to see sample values
<input checked="" type="checkbox"/>	ABC EMP_CODE		Click here to see sample values
<input checked="" type="checkbox"/>	ABC EMP_CODE (2)		Click here to see sample values
<input checked="" type="checkbox"/>	ABC ENAME		Click here to see sample values
<input checked="" type="checkbox"/>	123 NET_AMT		Click here to see sample values
<input checked="" type="checkbox"/>	ABC SAL_MON		Click here to see sample values
<input checked="" type="checkbox"/>	123 SALARY		Click here to see sample values
<input checked="" type="checkbox"/>	ABC TEXT1		Click here to see sample values

Previous

Next

Create

Cancel



Let's Solve