

SAP How-to Guide

SAP HANA Cloud

# Live Analytics Across All Your Enterprise Data Powered by SAP HANA Cloud & Analytics Cloud

*Provided by SAP Platform & Technology, Global Adoption and Consumption Centre of Excellence*

## **Applicable Releases:**

**SAP HANA Cloud**

**SAP Analytics Cloud**

**Version 1.00**

**June 2021**

**THE BEST RUN**









## Document History

Document Version	Description
1.00	First official release of this guide

## Typographic Conventions

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

## Icons

Icon	Description
	Caution
	Note or Important
	Example
	Recommendation or Tip

# Table of Contents

<b>1.</b>	<b>Prerequisites.....</b>	<b>1</b>
<b>2.</b>	<b>Introduction .....</b>	<b>2</b>
2.1	Goal .....	2
2.2	Background .....	2
2.2.1	Live Data Connectivity in SAP Analytics Cloud.....	2
2.2.2	The Challenge and the Solution .....	3
2.2.3	Connecting SAP HANA Cloud to Remote Data Sources.....	4
2.2.3.1	Virtualizing Data Using SDA .....	4
2.2.3.2	Virtualizing Data Using SDI .....	4
2.2.4	Advantages of Using SAP HANA Cloud as a Data Gateway.....	5
2.2.4.1	Central Solution to All Your Data .....	5
2.2.4.2	Live Analytics .....	5
2.2.4.3	SAP HANA Benefits .....	5
<b>3.</b>	<b>Setup Connection to Microsoft SQL Server .....</b>	<b>6</b>
3.1	Install and Configure SAP Data Provisioning Agent.....	7
3.2	Add SQL Server as a Remote Source .....	12
3.3	Create User-Provided Service (UPS) .....	17
3.4	Create SAP HANA Database Project with HDI Container .....	19
3.5	Create Calculation View.....	29
3.6	Create a story in SAP Analytics Cloud.....	35
<b>4.</b>	<b>Setup Connection to SAP ERP .....</b>	<b>41</b>
4.1	Register ABAP Adapter in SAP Data Provisioning Agent.....	41
4.2	Add SAP ERP as a Remote Source .....	42
4.3	Create Custom BAPIs in SAP ERP .....	45
4.4	Create Virtual Procedure.....	59
4.5	Create HANA Function.....	63
4.6	Create Calculation View.....	66
4.7	Create a story in SAP Analytics Cloud.....	69
<b>5.</b>	<b>Current Limitations.....</b>	<b>74</b>
5.1	Building Calculation Views in SAP HANA Cloud.....	74
5.2	Geospatial is not Supported .....	74
5.3	Creating Custom BAPIs in SAP NetWeaver AS ABAP .....	74
5.4	Known Limitations for Neo Environment.....	74

## 1. Prerequisites

This guide is written based on the following software versions. The steps for other software versions are mostly the same or similar, and the differences are pointed out wherever necessary.

- SAP HANA Cloud
- SAP Analytics Cloud
- Microsoft SQL Server 2017
- S/4HANA on premise 1809
- SAP Data Provisioning Agent 2.5.3 (SP05)

## 2. Introduction

### 2.1 Goal

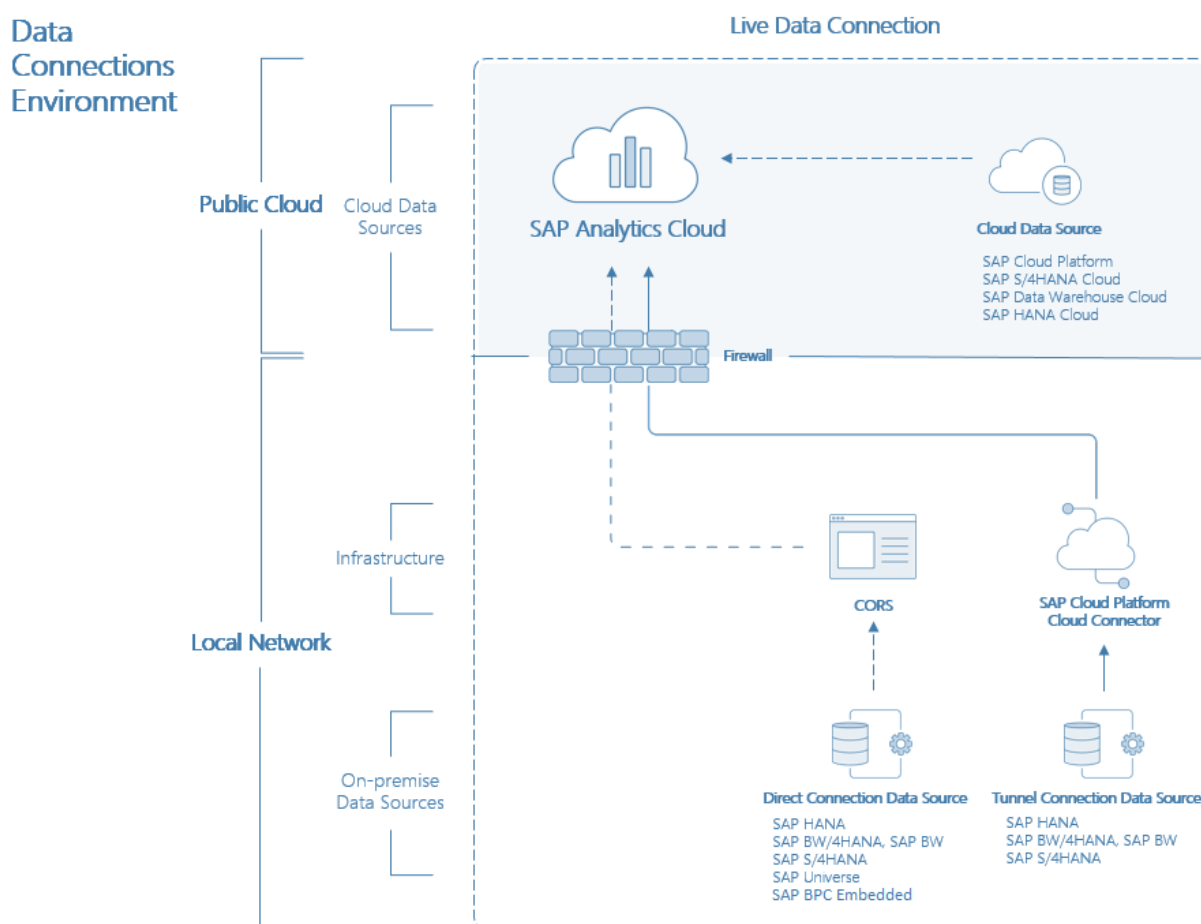
This document will help you run a real-time business with live analytics. Using SAP HANA Cloud as a data gateway, the live analytics capabilities of SAP Analytics Cloud will be expanded across heterogeneous data sources and SAP Analytics Cloud will be the central analytics solution for your corporation.

### 2.2 Background

#### 2.2.1 Live Data Connectivity in SAP Analytics Cloud

SAP Analytics Cloud provides live data connections to on-premise and cloud systems. Data is “live”, meaning that when a user opens a story in SAP Analytics Cloud, up-to-date data in the source system are retrieved immediately.

In SAP Analytics Cloud, you can create models from data sources in on-premise or cloud systems, build stories based on those models, and perform online analysis without data replication. The Live data connectivity allows SAP Analytics Cloud to be used in scenarios where data cannot be moved into the cloud for security or privacy reasons, or your data already exists on a different cloud system.



Using a live connection, SAP Analytics Cloud provides the business logic and builds the queries required to see your data in your browser. Your browser in turn sends those queries through a direct live connection to the on-premise data

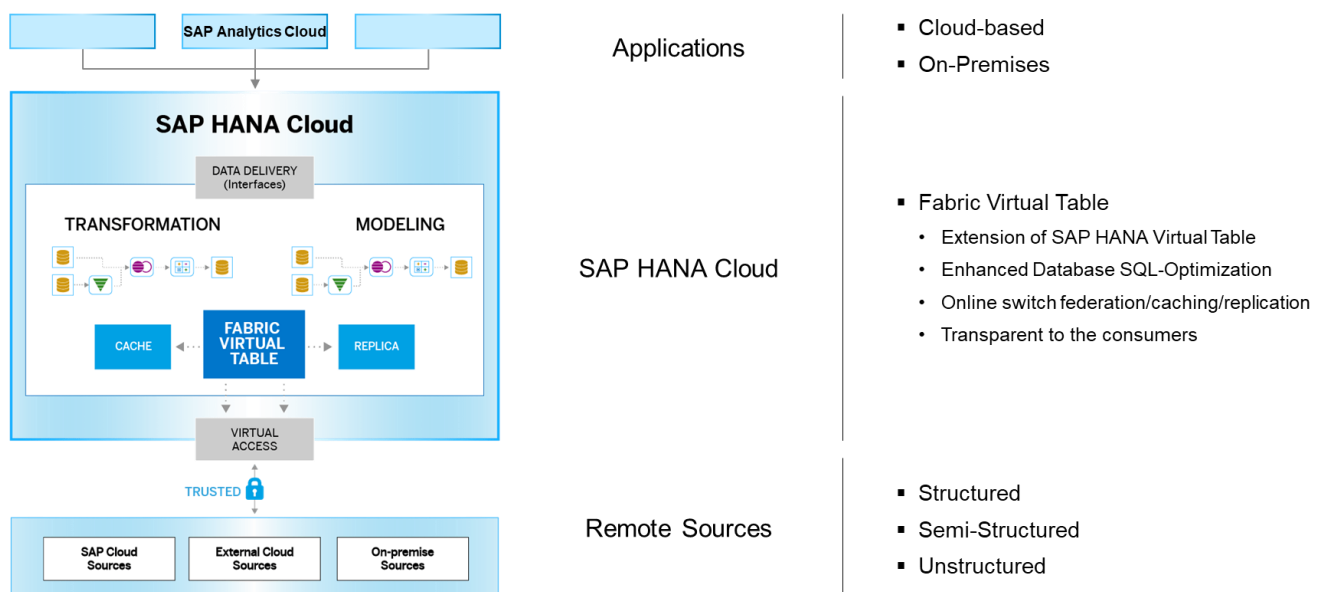
source. The results of those queries are returned to your browser, where visualizations are rendered. If your query was a list of profits per customer, for example, none of that information would return to or be stored in SAP Analytics Cloud. SAP Analytics Cloud stores queries for building the stories, measure names, columns names, filter values, and so on. Basically, the metadata lets SAP Analytics Cloud rebuild the query. But none of the payload data, not even the query results or part of the results, such as totals, are saved to SAP Analytics Cloud. Metadata is transferred to your browser and encrypted in memory.

Currently, SAP Analytics Cloud can connect to six types of live data sources: SAP HANA, SAP BW, SAP S/4HANA, SAP Universe, SAP BPC Embedded, and SAP Data Warehouse Cloud.

## 2.2.2 The Challenge and the Solution

The Live Data Connection is a preferred connection type for SAC to connect to remote sources, but it can only connect to six different data sources, specifically, six SAP systems. SAP Analytics Cloud doesn't provide live data connections to many other SAP solutions. Neither does SAC provide live data connections to non-SAP systems, such as Microsoft SQL Server, Oracle, MySQL, etc. These data sources are widely used by customers today. Moreover, this is not only about non-SAP systems, but also about those SAP systems such as SAP ECC 6.0 that cannot be connected live today. Although SAC provides acquired data connection to some other SAP solutions and non-SAP data sources, it has limitations on data volume of each data load, maintenance of delta load, replication and synchronization of authorization models. It would make SAP Analytics Cloud a broader adopted analytic application if it can provide live analytics to more SAP and non-SAP data sources.

To solve this challenge, SAP HANA Cloud can be used as a single gateway to connect to both on-premise and cloud data sources as shown in the following figure. With the data virtualization in SAP HANA Cloud, the analytical data won't be replicated from the remote source to SAP HANA Cloud or from SAP HANA Cloud to SAP Analytics Cloud. Hence, the customer can make decision for their organization based on the business insight in real-time. Moreover, this solution expands the Live Connections from SAP data to Non-SAP data, which makes SAP Analytics Cloud become an enterprise-wide analytics solution.





## 2.2.3 Connecting SAP HANA Cloud to Remote Data Sources

SAP HANA Cloud provides a single place to access, store, and process all enterprise data in real time. It is a cloud-native platform that reduces the complexity of multi-cloud or hybrid system landscapes. SAP HANA Cloud provides all of the advanced SAP HANA technologies for multi-model data processing in-memory or on disk. You can benefit from cloud qualities such as automatic software updates, elasticity, and low total cost of ownership by using SAP HANA Cloud either as a stand-alone solution or as an extension to your existing on-premise environment.

The SAP HANA Cloud provides simplified data access to connect all your information without the need to have all data loaded into a single storage solution. This capability provides operational and cost benefits and supports the development and deployment of next-generation analytical applications requiring the ability to access, synthesize, and integrate data from multiple systems in real time. In SAP HANA Cloud, you can create virtual objects using either SAP HANA Smart Data Access (SDA) or SAP HANA Smart Data Integration (SDI).

### 2.2.3.1 Virtualizing Data Using SDA

In SAP HANA Cloud, you create virtual tables, which point to remote tables in different data sources, and then write SQL queries to use these virtual tables. The SAP HANA query processor optimizes these queries by executing the relevant part of the query in the target database, returning the results of the query to SAP HANA, and then completing the operation. Physical data movement is not supported by SAP HANA Smart Data Access. Smart Data Access provides the ability to access and integrate data from multiple systems in real time regardless of where the data is located or what systems are generating the data. Thus, SDA enables SAP HANA Cloud to access heterogeneous sources in real-time.

The following figure shows the SDA supported databases as remote sources for SAP HANA Cloud. For more information, please see [SAP HANA Smart Data Access Supported Remote Sources](#).



### 2.2.3.2 Virtualizing Data Using SDI

Virtualizing data is also supported by SAP HANA Smart Data Integration (SDI). SDI uses SDA as a foundation and extends the capabilities of SDA. SAP HANA SDI also supports virtually accessing a variety of remote sources in real time using pre-built and custom adapters. The SAP HANA Data Provisioning Agent is required for connections with SDI. Virtual tables can be created for remote sources supported by SDI in the same way as for SDA.

SDI supports more remote sources and more versions of remote sources than SDA. The following figure shows some of the SDI supported remote sources for SAP HANA Cloud. For more information, please see [SAP HANA SDI and all its patches Product Availability Matrix \(PAM\)](#).



## 2.2.4 Advantages of Using SAP HANA Cloud as a Data Gateway

When using SAP HANA Cloud as a single gateway to all your data, it offers the following advantages:

- Manage data from both SAP and non-SAP landscape
- Gain real-time insight

### 2.2.4.1 Central Solution to All Your Data

SAP HANA Cloud helps organizations manage and access large volumes of data coming from multiple sources via data virtualization, offering simplicity and scalability. With data virtualization, SAP Analytics Cloud can live connect to heterogeneous data sources via SAP HANA Cloud. In other words, SAP HANA Cloud expands the Live Analytics capabilities of SAC from SAP-centric usage to enterprise-wide analytics tool. Therefore, SAP Analytics Cloud becomes a central analytics solution to all your enterprise data.

### 2.2.4.2 Live Analytics

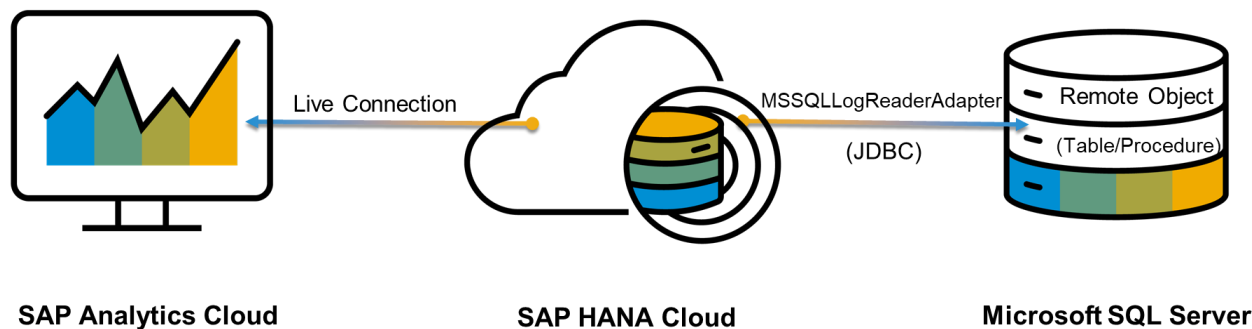
In the world of analytics, real-time reporting has been sought after for the last decade. With data virtualization, SAP Analytics Cloud performs advanced analytics on live transactional data, structured, in real-time and without data duplication. SAP HANA's in-memory analytics is the game changer that makes this possible, delivering rapid access to visually rich data that can be interactively explored and quickly analyzed. SAP HANA Cloud brings the simplicity and speed of SAP HANA to the cloud and stimulates the performance considering the whole thing is completed under the seamless platform. Hence, the business users can leverage SAP HANA Cloud to explore the accountable insights and make decisions with confidence in SAP Analytics Cloud.

### 2.2.4.3 SAP HANA Benefits

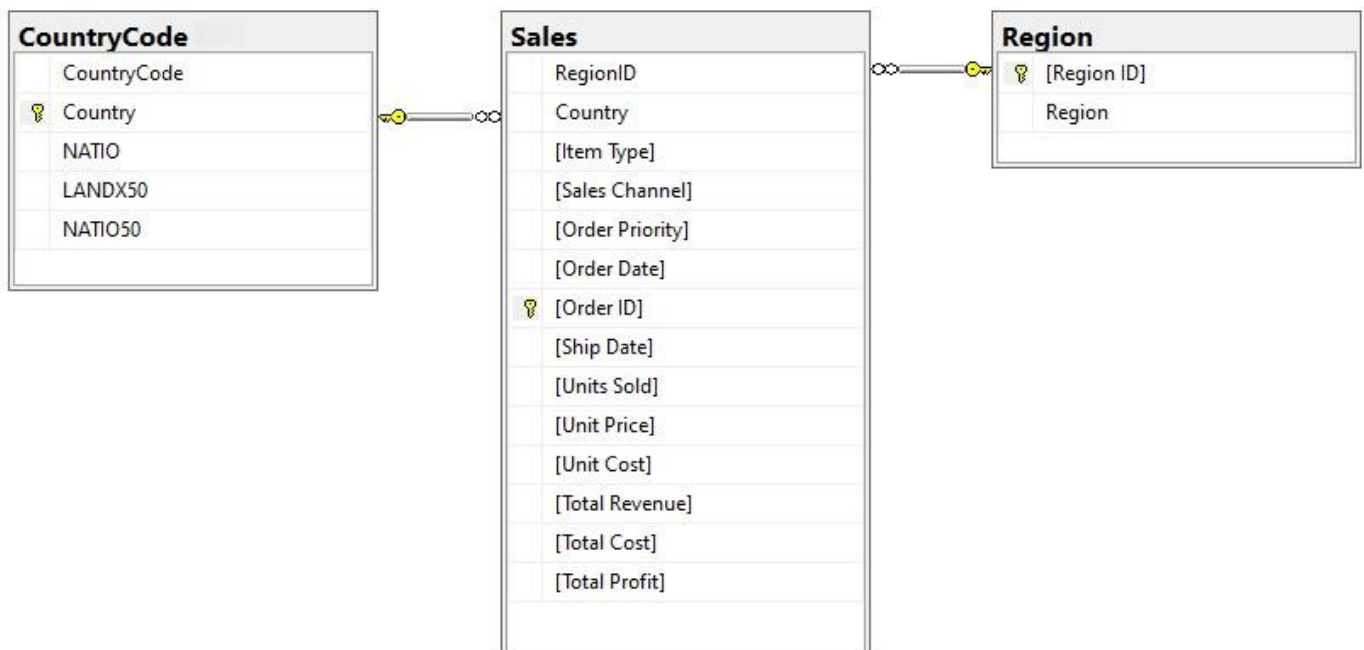
SAP HANA Cloud is not just a data gateway to remote sources. In SAP HANA Cloud, hierarchies can be added on flat data structure in source tables. This additional value is very important because it enables SAC to perform drill down analysis. SAP HANA Cloud supports both parent-child hierarchy and level-based hierarchy, and both can be consumed by SAC. Apart from exposing data from various remote sources in the form of virtual tables, data virtualization allows to combine these data with data resides in the SAP HANA physical tables. The feasibility to apply HANA planning, predictive as well as text search algorithm with the combined dataset provides an outstanding benefit to the customers and business.

### 3. Setup Connection to Microsoft SQL Server

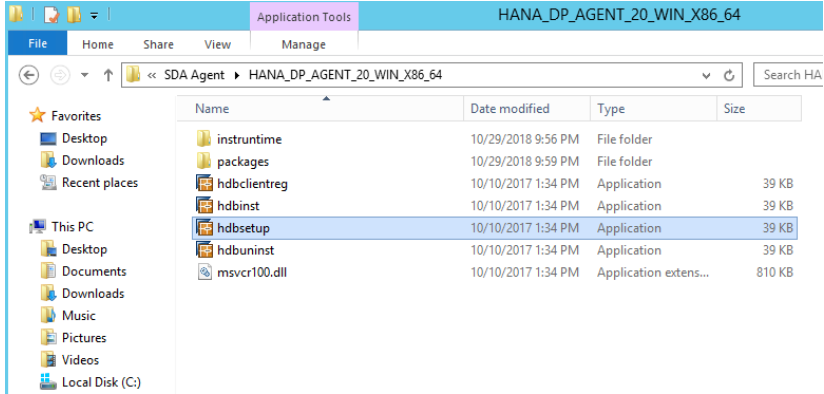
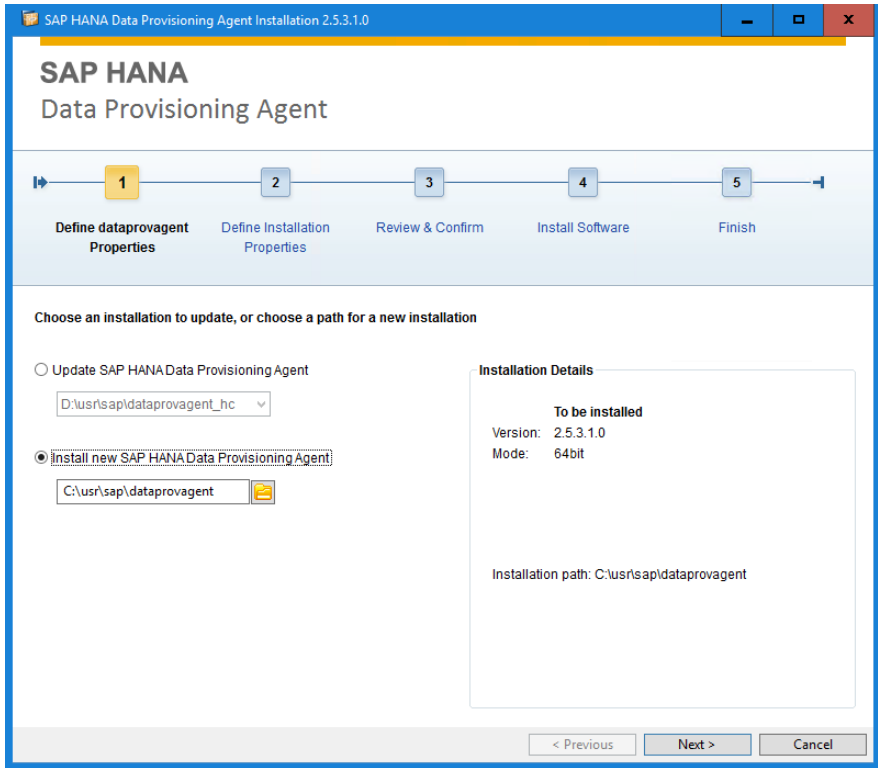
In this section, we will take Microsoft SQL Server as an example remote database and setup SAC live data connection to Microsoft SQL Server using SAP HANA SDI. Firstly, we need to setup SQL Server JDBC driver in Data Provisioning Agent to ensure that SAP HANA Cloud can access Microsoft SQL Server through JDBC protocol. Next, we create virtual tables that point to the tables in SQL Server, and then use the virtual tables to create calculation views, which can be consumed by SAC. Lastly, SAC establishes a live data connection to SAP HANA and eventually visualized the data in Microsoft SQL Server. The system landscape is setup as below.



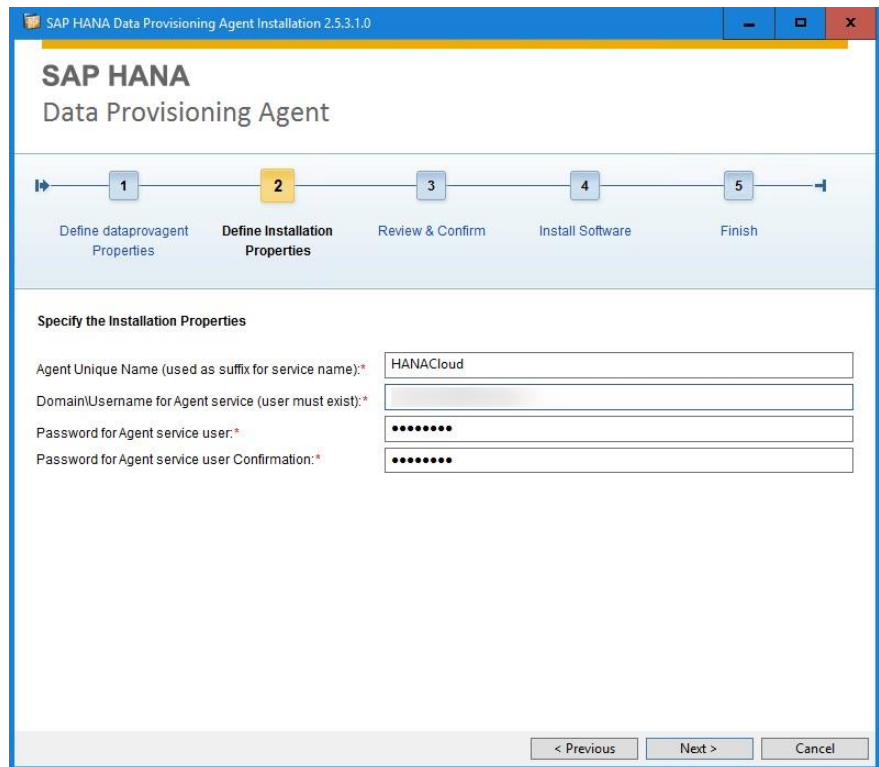
To test the Live Analytics to SQL Server, we created a sample database in SQL Server. The sample database contains three tables: **CountryCode**, **Sales**, and **Region** under the one schema. The following figure illustrates the sample database diagram. The following tables are not provided in this documentation. You may use any similar tables in your database to create a calculation view in this exercise.



## 3.1 Install and Configure SAP Data Provisioning Agent

Explanation	Screenshot
<ol style="list-style-type: none"> <li>1. Data Provisioning Agent Planning and Preparation. <ol style="list-style-type: none"> <li>a. <a href="#">Supported Platforms and System Requirements</a></li> <li>b. <a href="#">Software Download</a></li> </ol> </li> <li>2. Download <b>SAP HANA SDI 2.0</b> and extract the software to an empty directory</li> </ol>	
<ol style="list-style-type: none"> <li>3. Navigate to the directory where you unpacked the software and execute <b>hdbsetup.exe</b></li> <li>4. Assign a folder name for the new SAP HANA DATA Provisioning Agent and click <b>Next</b></li> </ol>	

5. Fill up all the mandatory fields and click **Next**

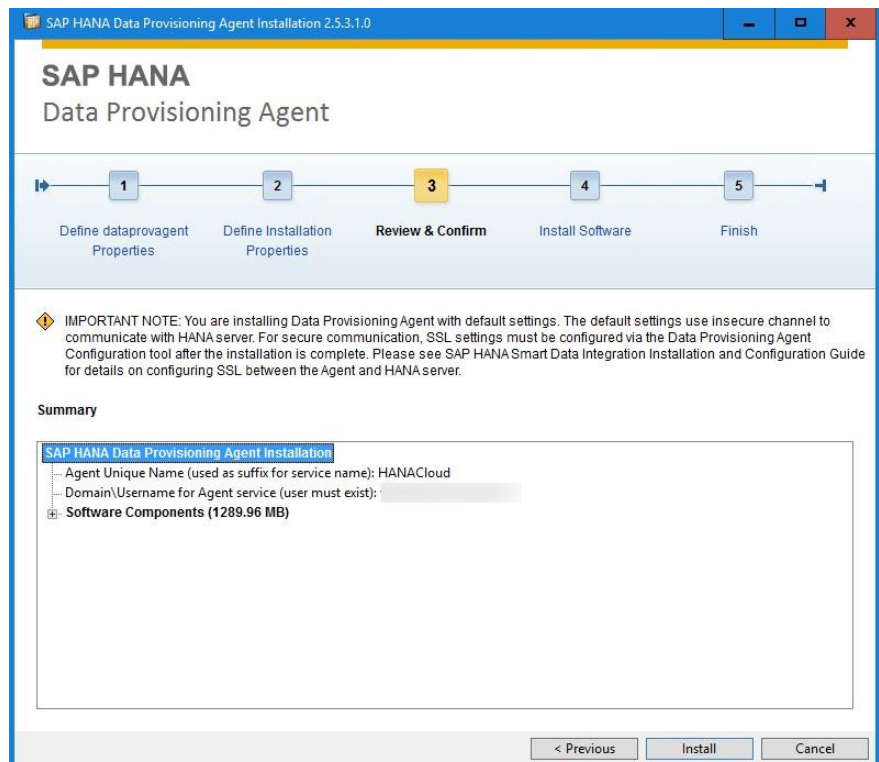


The screenshot shows the 'SAP HANA Data Provisioning Agent' installation window at step 2, 'Define Installation Properties'. The progress bar at the top indicates five steps: 1. Define dataprovagent Properties, 2. Define Installation Properties (current), 3. Review & Confirm, 4. Install Software, and 5. Finish. The main area is titled 'Specify the Installation Properties' and contains four input fields with red asterisks indicating they are mandatory:

- Agent Unique Name (used as suffix for service name):\*
- DomainUsername for Agent service (user must exist):\*
- Password for Agent service user:\*
- Password for Agent service user Confirmation:\*

At the bottom right, there are three buttons: '< Previous', 'Next >', and 'Cancel'.

6. Click **Install** and **Finish** the installation



The screenshot shows the 'SAP HANA Data Provisioning Agent' installation window at step 3, 'Review & Confirm'. The progress bar at the top indicates five steps: 1. Define dataprovagent Properties, 2. Define Installation Properties, 3. Review & Confirm (current), 4. Install Software, and 5. Finish. The main area contains an 'IMPORTANT NOTE' with a yellow warning icon:

**IMPORTANT NOTE:** You are installing Data Provisioning Agent with default settings. The default settings use insecure channel to communicate with HANA server. For secure communication, SSL settings must be configured via the Data Provisioning Agent Configuration tool after the installation is complete. Please see SAP HANA Smart Data Integration Installation and Configuration Guide for details on configuring SSL between the Agent and HANA server.

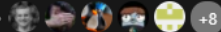
Below the note is a 'Summary' section with a blue header 'SAP HANA Data Provisioning Agent Installation'. It lists the following details:

- Agent Unique Name (used as suffix for service name): HANACloud
- DomainUsername for Agent service (user must exist):
- Software Components (1289.96 MB)

At the bottom right, there are three buttons: '< Previous', 'Install', and 'Cancel'.

7. Before registering the adapter with the SAP HANA system, ensure you have downloaded and installed the correct JDBC libraries. See the [SAP HANA smart data integration Product Availability Matrix \(PAM\)](#) for details.





## Download Microsoft JDBC Driver for SQL Server

03/02/2021 • 2 minutes to read •  +8

The Microsoft JDBC Driver for SQL Server is a Type 4 JDBC driver that provides database connectivity through the standard JDBC application program interfaces (APIs) available on the Java platform. The driver downloads are available to all users at no extra charge. They provide access to SQL Server from any Java application, application server, or Java-enabled applet.

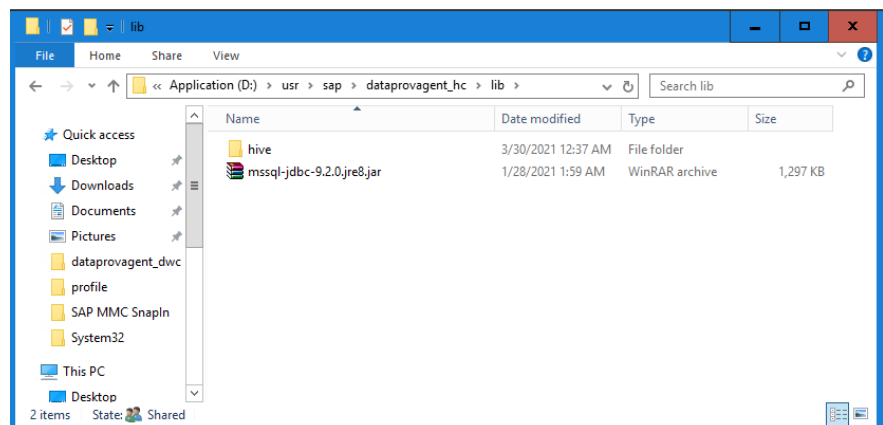
### Download

Version 9.2 is the latest general availability (GA) version. It supports Java 8, 11, and 15. If you need to use an older Java runtime, see the [Java and JDBC specification support matrix](#) to see if there's a supported driver version you can use. We're continually improving Java connectivity support. As such we highly recommend that you work with the latest version of Microsoft JDBC driver.

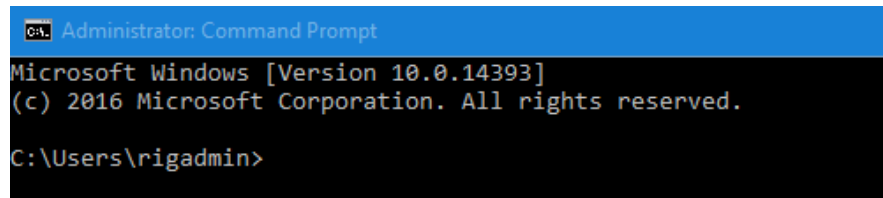
-  [Download Microsoft JDBC Driver 9.2 for SQL Server \(zip\)](#) 
-  [Download Microsoft JDBC Driver 9.2 for SQL Server \(tar.gz\)](#) 

8. Unzip the downloaded package and place the files in the <DPAgent\_root>/lib folder

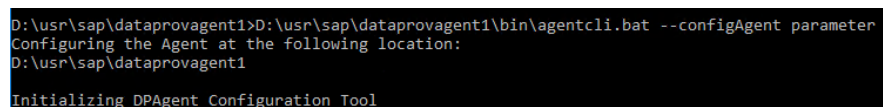
**Note:** <DPAgent\_root> is the directory where the DP agent was installed. By default, on Windows, this is C:\usr\sap\dataprovagent, and on Linux it is /usr/sap/dataprovgent.



9. Open an Administrator prompt on the Windows where the DP agent is installed



10. Start the Data Provisioning Agent: You must start the agent, if it has not been started already: Select option **2. Start or Stop Agent**, and then option **1. Start Agent**. Select **b. Back** to go back to the main menu.





<pre>&lt;DPAgent_root&gt;\bin\agentcli.bat --configAgent</pre>	<pre>***** DPAgent Configuration Tool ***** 1. Agent Status 2. Start or Stop Agent 3. Agent Preferences 4. Remote Source Credentials 5. SSL Keystores &amp; Settings 6. SAP HANA Connection 7. Agent Registration 8. Adapter Registration 9. Custom Adapters 10. Agent &amp; Adapter Versions q. Quit b. Back *****</pre>
<p>11. Select option <b>6. SAP HANA Connection</b></p>	<pre>Enter Option:6 ***** SAP HANA Connection ***** 1. Connect to SAP HANA Cloud via JDBC 2. Connect to SAP Data Warehouse Cloud via JDBC 3. Connect to SAP HANA service for SAP BTP, Cloud Foundry environment via JDBC 4. Connect to SAP HANA service for SAP BTP, Neo environment via HTTP 5. Connect to local SAP HANA (on-premises) via JDBC 6. Connect to local SAP HANA (on-premises) via TCP q. Quit b. Back ***** Enter Option: _</pre>
<p>12. Select option <b>1. Connect to SAP HANA Cloud via JDBC</b></p> <p>Specify the hostname, port, and Agent Admin HANA User credentials for the SAP HANA server as prompted.</p> <p>Select <b>b. Back</b> to go back to the main menu.</p>	<pre>Enter Option:1 ***** Connect to SAP HANA Cloud via JDBC ***** Press "Enter" button to keep default or skip an optional setting. If a setting cannot be skipped, that means it is required. ***** Enter Use encrypted JDBC connection[true]: Valid options: true false Enter Host Name[.hanacloud.d Enter Port Number[443]: Enter Agent Admin HANA User[DBADMIN]: Enter Agent Admin HANA User Password: Enter Agent Admin HANA User Password: (confirm) Enter Use Proxy Server[true]: Valid options: true false Enter Is Proxy Server HTTP? (Enter true for HTTP or false for SOCKS)[true]: Val Enter Proxy Host[proxy]: Enter Proxy Port[8080]: Enter Use Proxy Authentication[false]: Valid options: true false Enter HANA User Name for Agent Messaging[XADMIN]: Enter HANA User Password for Agent Messaging: (*****) Connecting to SAP HANA server via JDBC... Agent configuration tool is connected to SAP HANA server via JDBC.</pre>

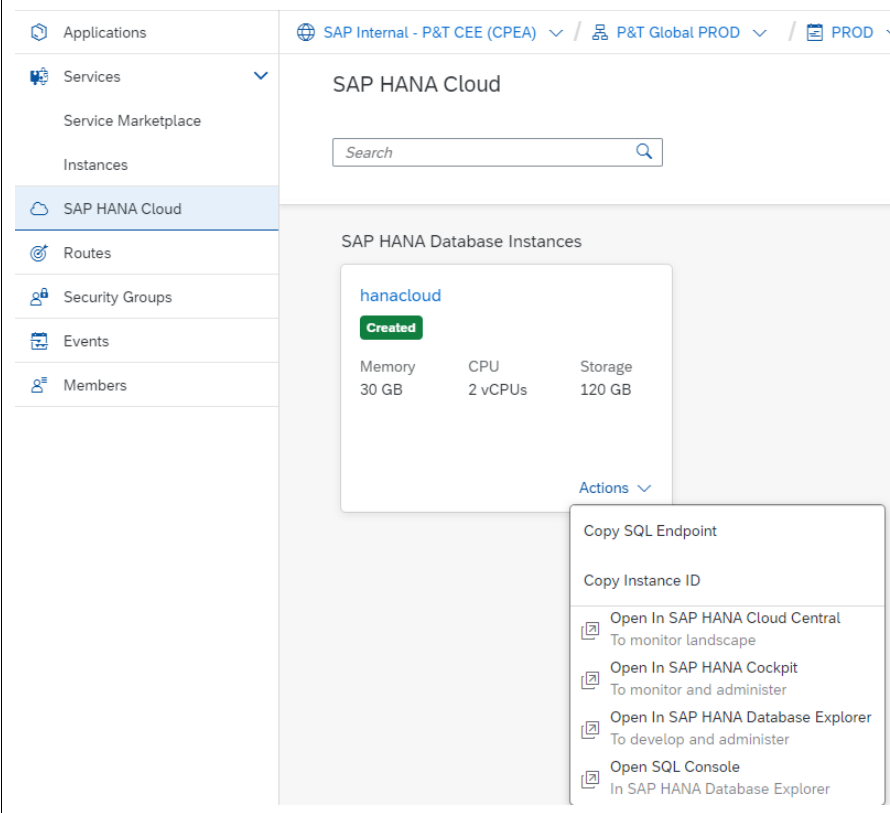
<p>Select option <b>7. Agent Registration</b></p>	<pre> ***** DPAgent Configuration Tool ***** 1. Agent Status 2. Start or Stop Agent 3. Agent Preferences 4. Remote Source Credentials 5. SSL Keystores &amp; Settings 6. SAP HANA Connection 7. Agent Registration 8. Adapter Registration 9. Custom Adapters 10. Agent &amp; Adapter Versions q. Quit b. Back ***** Enter Option:7 ***** Agent Registration ***** 1. Register Agent 2. Unregister Agent q. Quit b. Back </pre>
<p>13. Select option <b>1. Register Agent</b></p> <p>Specify the agent connection information as prompted.</p> <p>Select <b>b. Back</b> to go back to the main menu.</p>	<pre> Enter Option:1 ***** Register Agent ***** Press "Enter" button to keep default or skip an optional setting. If a setting cannot be skipped, that means it is required. ***** Enter Agent Name[dpagent_wdflbmt2652]: Enter Agent Host Name[wdflbmt2652]: Registering agent 'dpagent_wdflbmt2652' with SAP HANA. Agent 'dpagent_wdflbmt2652' successfully registered. </pre>
<p>14. Select option <b>8. Adapter Registration</b></p> <p>15. Select option <b>2. Register Adapter</b></p> <p>16. Enter adapter name: <b>MssqlLogReaderAdapter</b></p> <p><b>Note:</b> The adapter name must match the name displayed by the option <b>1. Display Adapters</b>.</p>	<pre> ***** Adapter Registration ***** 1. Display Adapters 2. Register Adapter 3. Unregister Adapter q. Quit b. Back ***** Enter Option:2 Enter adapter name: MssqlLogReaderAdapter Adapter 'MssqlLogReaderAdapter' successfully registered. </pre>



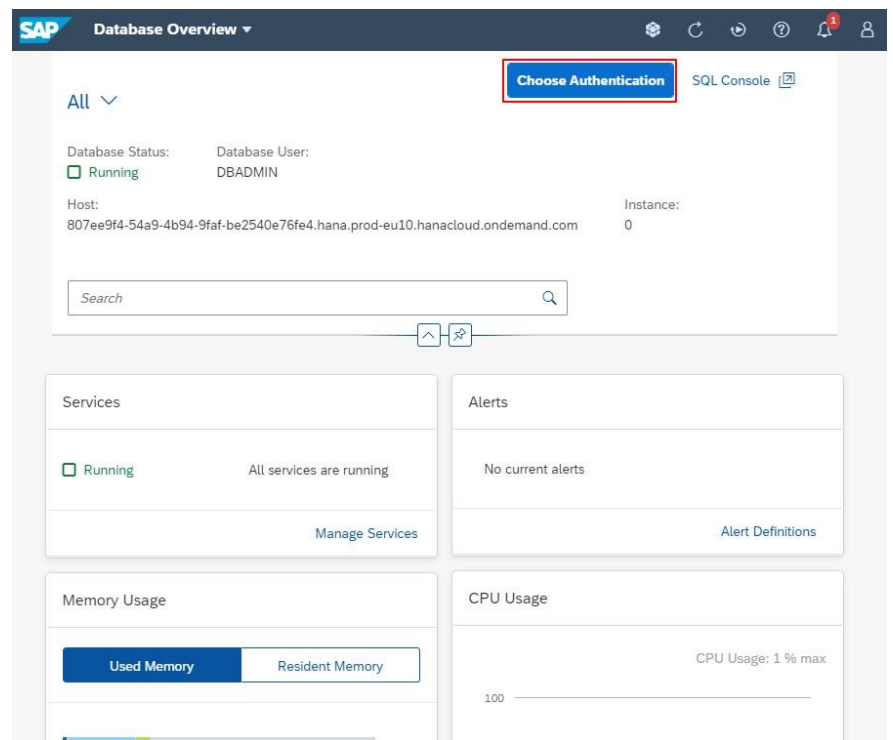
## 3.2 Add SQL Server as a Remote Source

Next, we will add the SQL Server as a remote source in SAP HANA Database Explorer. The following privileges are required to manage remote sources and virtual tables:

- System privileges: CREATE REMOTE SOURCE
- Object privilege: CREATE VIRTUAL TABLE on remote source SQLServer17

Explanation	Screenshot
17. Logon to SAP HANA Cloud instance and open the <b>SAP HANA Cockpit</b> .	

18. You are currently logging in as DBADMIN. You can switch to a technical user by clicking **Choose Authentication**



19. Enter **Username** and **Password**  
20. Click **OK**

**Credentials**

---

There are currently stored database logon credentials for hanacloud. What would you like to do?

☒ Log on with a different database user

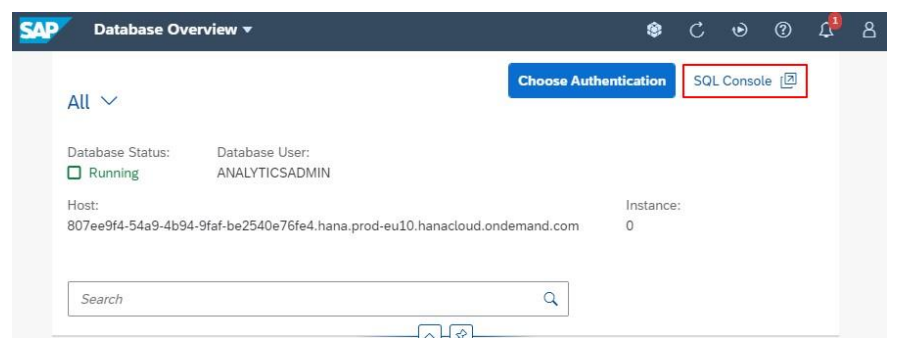
Username:

Password:

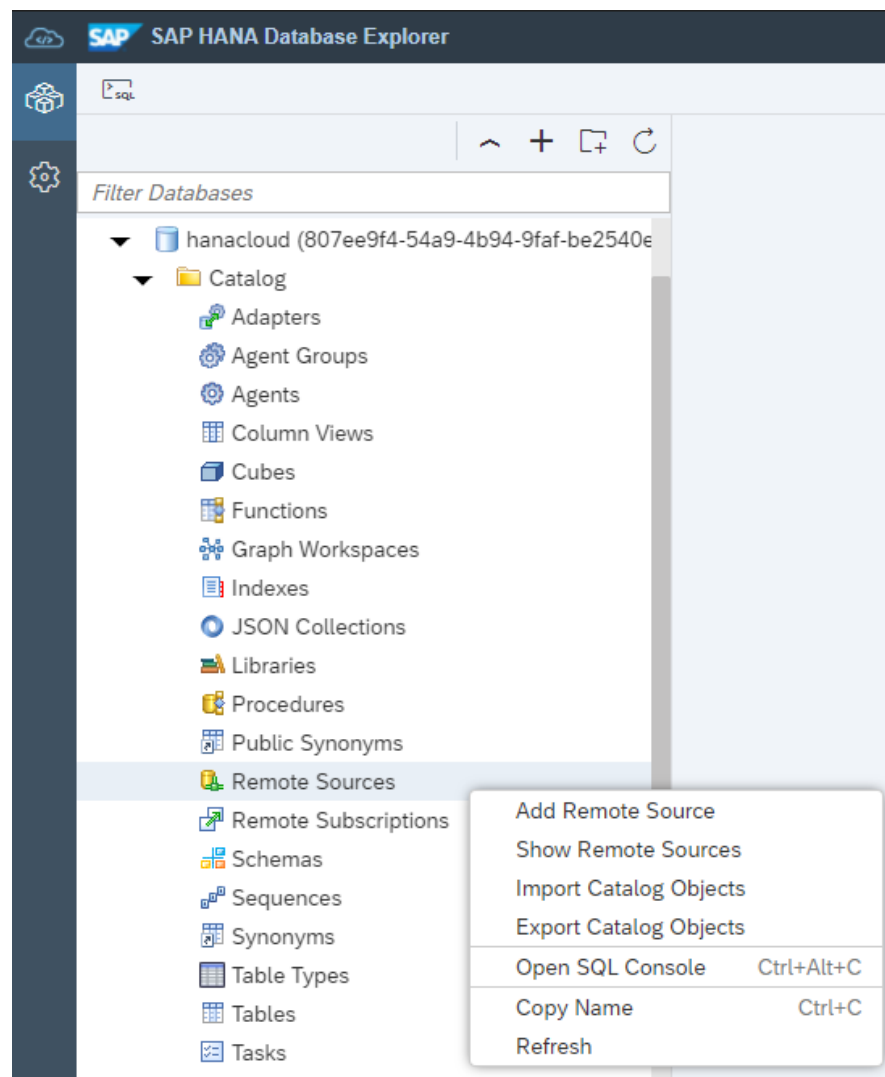
☐ Delete the stored credentials for this database

---

**OK**   **Cancel**

21. Open **SQL Console**

22. Go to **Catalog** -> **Remote Source**
23. Right click on the **Remote Source** and **Add Remote Source**



## 24. Fill in the remote source details:

Source Name  
 Adapter Name:  
 MssqlLogReaderAdapter  
 Data Server  
 Database Name  
 Credentials Mode: Technical  
 user  
 User Name  
 Password

**Create the remote source.**

Add Remote Source

\*Source Name

\*Adapter Name

Source Location

Property	Value
▼ Configurations	
▼ Generic	
Load and Replicate LOB colu...	<input type="text" value="true"/>
▼ Data Type Conversion	
Always Map Character Types ...	<input type="text" value="false"/>
Allow Map Character Types t...	<input type="text" value="true"/>
*Map SQL Server Data Type ...	<input type="text" value="false"/>
▼ Database	
*SQL Server Always On	<input type="text" value="false"/>
*Data Server (serverName[\\in...	<input type="text" value="www.yourservername.com"/>
Port Number	<input type="text" value="1433"/>

25. Now, you can see the SQL Server in the list of Remote Source, open the remote source **SQLServer17**

hanacloud (807ee9f4-54a9-4b94-9faf-be2540e76fe4.hana.prod-eu10.hanacloud.ondemand.com:443): SQLSen

SQLServer17 x

Filter Databases

- hanacloud (807ee9f4-54a9-4b94-9faf-be2540e76fe4.hana.prod-eu10.hanacloud.ondemand.com:443)
  - Catalog
    - Adapters
    - Agent Groups
    - Agents
    - Column Views
    - Cubes
    - Functions
    - Graph Workspaces
    - Indexes
    - JSON Collections
    - Libraries
    - Procedures
    - Public Synonyms
    - Remote Sources
    - Remote Subscriptions
    - Schemas
    - Sequences

Search RemoteSources

HI4

MySQL80

S4\_R14

SQLServer17

SQLServer17

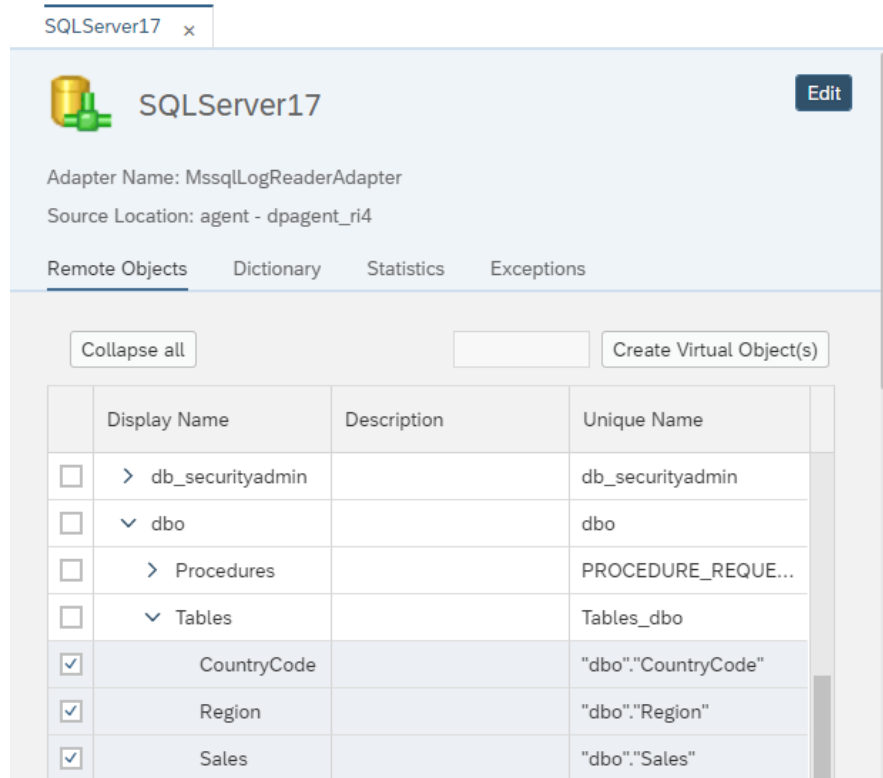
Adapter Name: MssqlLogReaderAdapter  
 Source Location: agent - dpagent\_r14

Remote Objects Dictionary Statistics Exceptions

Collapse all

	Display Name	Description	Unique Name
<input type="checkbox"/>	> INFORMATION_...		INFORMATION_SCHE...
<input type="checkbox"/>	> db_accessadmin		db_accessadmin
<input type="checkbox"/>	> db_backupopera...		db_backupoperator
<input type="checkbox"/>	> db_datareader		db_datareader
<input type="checkbox"/>	> db_datawriter		db_datawriter
<input type="checkbox"/>	> db_ddladmin		db_ddladmin
<input type="checkbox"/>	> db_denydatarea...		db_denydatareader
<input type="checkbox"/>	> db_denydatawriter		db_denydatawriter
<input type="checkbox"/>	> db_owner		db_owner
<input type="checkbox"/>	> db_securityadmin		db_securityadmin

26. Go to schema **dbo** -> **Tables** .  
27. Select all 3 tables and **Create Virtual Object(s)**



SQLServer17 x

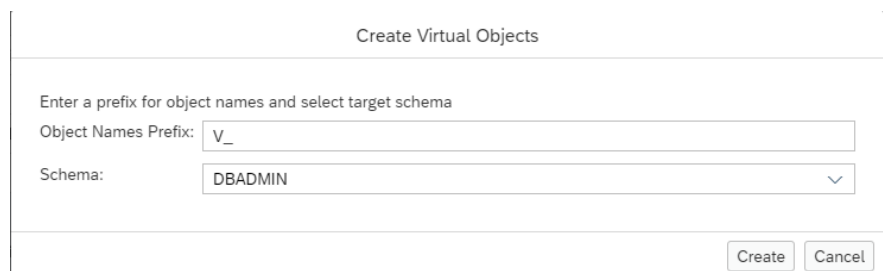
Adapter Name: MssqlLogReaderAdapter  
Source Location: agent - dpagent\_r4

Remote Objects Dictionary Statistics Exceptions

Collapse all Create Virtual Object(s)

	Display Name	Description	Unique Name
<input type="checkbox"/>	> db_securityadmin		db_securityadmin
<input type="checkbox"/>	▼ dbo		dbo
<input type="checkbox"/>	> Procedures		PROCEDURE_REQUE...
<input type="checkbox"/>	▼ Tables		Tables_dbo
<input checked="" type="checkbox"/>	CountryCode		"dbo"."CountryCode"
<input checked="" type="checkbox"/>	Region		"dbo"."Region"
<input checked="" type="checkbox"/>	Sales		"dbo"."Sales"

28. Optionally, give the Virtual Objects a prefix and select a schema to create the virtual tables.



Create Virtual Objects

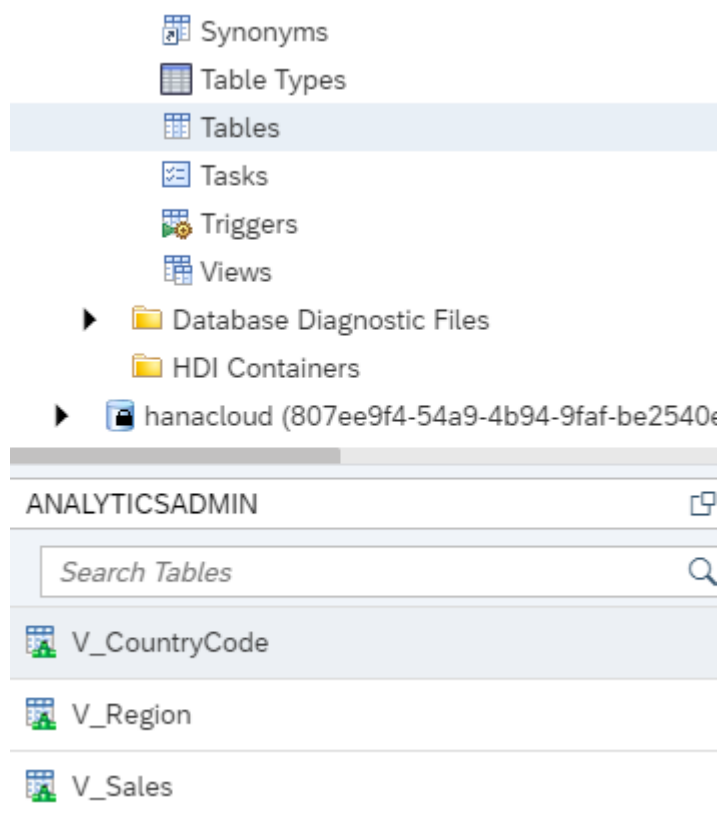
Enter a prefix for object names and select target schema

Object Names Prefix: V\_

Schema: DBADMIN

Create Cancel

29. Once the virtual tables are created, they will appear under the Tables of the selected schema.

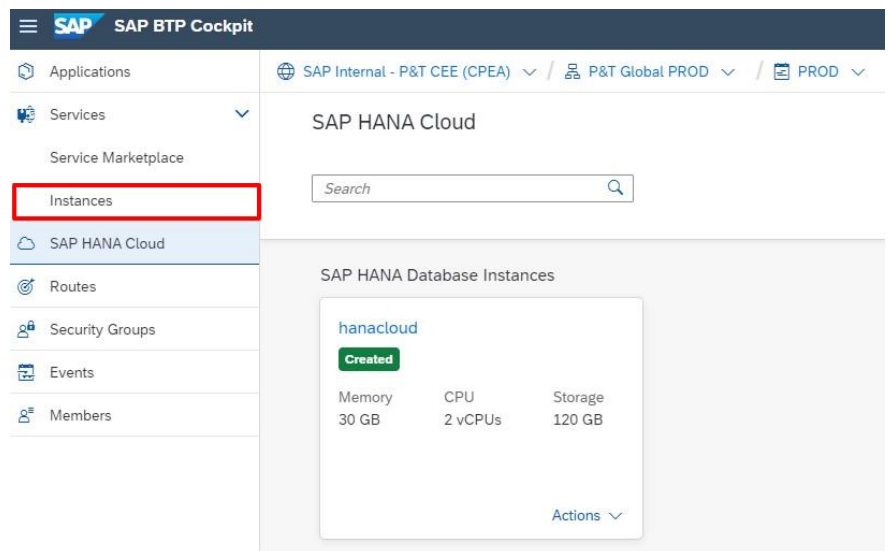


### 3.3 Create User-Provided Service (UPS)

Before building a SAP HANA Database project in SAP Business Application Studio, we need to create a User-Provided Service in SAP Business Technology Platform (BTP) subaccount. This USP will be used by the HANA project to connect to SAP HANA Cloud.

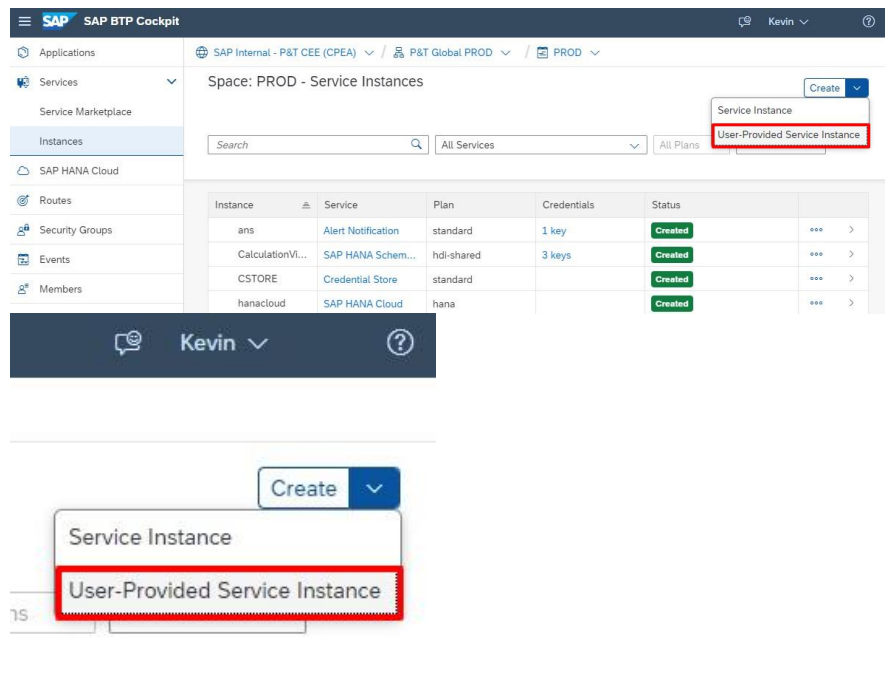
Explanation	Screenshot
-------------	------------

30. Navigate to **SAP BTP Cockpit**  
 31. Click on **Service** -> **Instances**



The screenshot shows the SAP BTP Cockpit interface. On the left, the navigation menu has 'Instances' highlighted with a red box. The main area displays 'SAP HANA Cloud' with a search bar. Below, 'SAP HANA Database Instances' are listed, showing a 'hanacloud' instance with a 'Created' status. The instance details include Memory (30 GB), CPU (2 vCPUs), and Storage (120 GB).

32. Create a **User-Provided Service Instance**



The screenshot shows the SAP BTP Cockpit interface for creating a service instance. The 'Create' dropdown menu is open, showing 'Service Instance' and 'User-Provided Service Instance', with the latter highlighted in red. Below, a table lists existing instances, including 'ans', 'CalculationVI...', 'CSTORE', and 'hanacloud'. The 'Create' button is also visible at the bottom.

Instance	Service	Plan	Credentials	Status
ans	Alert Notification	standard	1 key	Created
CalculationVI...	SAP HANA Schem...	hdi-shared	3 keys	Created
CSTORE	Credential Store	standard		Created
hanacloud	SAP HANA Cloud	hana		Created

33. Enter **Instance Name**

34. Specify the parameter in JSON format:

```
{
  "host": "<Your SAP HANA
Cloud SQL End>",
  "port": "443",
  "user": "ANALYTICSADMIN",
  "password": "Welcome1",
  "tags": ["hana"]
}
```

Update User-Provided Service Instance

Instance Name: \*

AnalyticsUPS

System Logs Drain URL:

Enter URL

Route Service URL:

Enter URL

Configure instance parameters

Upload a JSON file:

Please select a file Browse...

Specify the parameters in JSON format: Clear

```
1 {
2   "host": "hana.prod-eu10.hanacloud
   .ondemand.com",
3   "port": "443",
4   "user": "ANALYTICSADMIN",
5   "password": "Welcome1",
6   "tags": ["hana"]
7 }
```

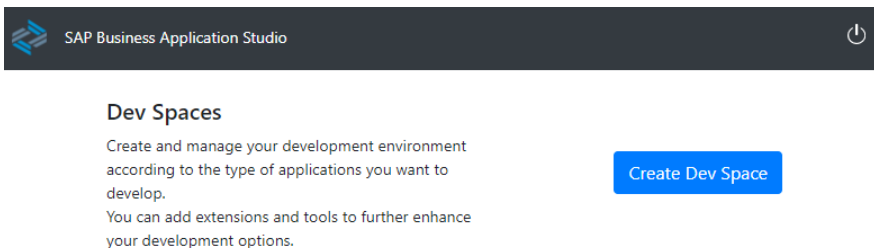
Save Cancel

35. The UPS instance is created in the list

Instance	Service	Plan	Credentials	Status	
AnalyticsUPS	User-Provided			Created	... >
ans	<a href="#">Alert Notification</a>	standard	1 key	Created	... >
CalculationVi...	<a href="#">SAP HANA Schem...</a>	hdi-shared	3 keys	Created	... >
CSTORE	<a href="#">Credential Store</a>	standard		Created	... >
hanacloud	<a href="#">SAP HANA Cloud</a>	hana		Created	... >

## 3.4 Create SAP HANA Database Project with HDI Container


In this section, we will create an SAP HANA Database Project in SAP Business Application Studio, deploy the project into SAP HANA Cloud as a HDI container.

Explanation	Screenshot
<p>36. Navigate to SAP Business Application Studio</p> <p>37. Create a new <b>Dev Space</b></p>	






38. Enter a Dev Space Name
39. Select **SAP HANA Native Application**
40. Click **Create Dev Space**

## Create a New Dev Space

What kind of application do you want to create?




- ☐  SAP Fiori
- ☐  Full Stack Cloud Application
- ☒  SAP HANA Native Application

## SAP HANA Native Application Dev Space

Build and deploy native SAP HANA applications or analytical models. This dev space contains a comprehensive set of editors to support the creation of database artifacts (calculation views, tables, SQLScript procedures, and more), as well as tools to enable an end-to-end development flow from project creation to the deployment to the SAP Cloud Platform.




## SAP Predefined Extensions

The following extensions are enabled by default.

-  **Basic Tools**  
Allows you to optimize your web development workflow. The... [more](#)
-  **SAP HANA Calculation View Editor**  
Allows you to edit and manage SAP HANA calculation views. The... [more](#)
-  **SAP HANA Database Explorer**

## Additional SAP Extensions






Select additional extensions to enhance your space.

- ☐  **SAPUI5 Adaptation Project**  
Allows to extend SAPUI5 applications using Adaptation project and Visual Editor
- ☐  **CDS Graphical Modeler**  
Allows you to design SAP core data services models in SAP... [more](#)
- ☐  **SAP HANA Performance**

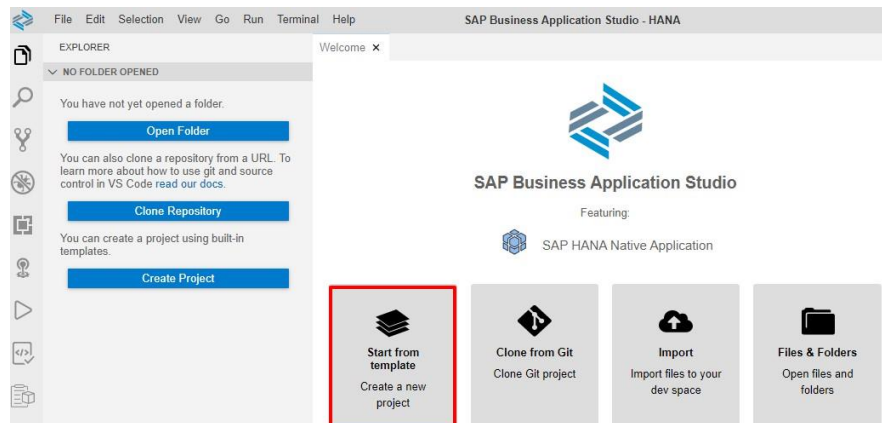
Cancel

Create Dev Space

41. Wait until the dev space status changed to **RUNNING**
42. Click on Dev Space Name

 <div> <b>HANA</b>  SAP HANA Native Application </div>	<div><b>RUNNING</b></div> <div>Created On: 03/25/2021 8:41 PM</div> <div>ID: ws-2fxz5</div> <div>     </div>
---	--

43. Create a new project by click on **Start from template** tile



The screenshot shows the SAP Business Application Studio interface. On the left is the Explorer panel with options: 'Open Folder', 'Clone Repository', and 'Create Project'. The main area displays the 'SAP HANA Native Application' project. Below the project name, there are four tiles: 'Start from template' (highlighted with a red box), 'Clone from Git', 'Import', and 'Files & Folders'.

44. Select **SAP HANA Database Project**  
45. Click **Start**

## New Project From Template

● Select Template and Target Location

### Select Template and Target Location

Select the template that best fits the type of application you want to develop.

The template will be generated in the folder specified below and it will contain the relevant files and configurations that will help you create a project quickly and easily.

Specify a target folder path [?](#) [Preferences](#)

/home/user/projects

Templates [?](#) \*

#### Basic Multitarget Application

Create a project for developing a multitarget application (MTA). MTAs are comprised of multiple

[More Information](#)



#### SAP HANA Database Project

Creates a new SAP HANA database project

[More Information](#)



**Start** >

46. Enter a Project Name  
47. Click **Next**

## New Project From Template



Select Template  
and Target  
Location



Add Basic  
Information



Set Basic  
Properties



Set Database  
Information



Bind to HDI  
Container  
service

### Add Basic Information

Provide basic project information

Project Name

[< Start Over](#)[Next >](#)

48. Click **Next**

## New Project From Template



Select Template  
and Target  
Location



Add Basic  
Information



Set Basic  
Properties



Set Database  
Information



Bind to HDI  
Container  
service

### Set Basic Properties

Enter the module name

[< Back](#)[Next >](#)

49. Select **HANA Cloud** as SAP HANA Database Version  
50. Click Next

## New Project From Template

● Select Template and Target Location

● Add Basic Information

● Set Basic Properties

● Set Database Information

○ Bind to HDI Container service

### Set Database Information

Define details of your application's database module.


Namespace

Schema Name

SAP HANA Database Version \*

HANA Cloud



Bind the database module to a Cloud Foundry service instance? 

☒ Yes ☐ No

< Back

Next >

51. Enter the e-mail address and Password
52. Select the Cloud Foundry Org and Space
53. Click Finish

## New Project From Template

● Select Template and Target Location

● Add Basic Information

● Set Basic Properties

● Set Database Information

● Bind to HDI Container service

### Bind to HDI Container service

Please provide your Cloud Foundry login credential:  
HDI container service binding if needed.

Enter Cloud Foundry endpoint

`https://api.cf.us10.hana.ondemand.com`

Enter e-mail address

`kevin.li03@sap.com`

Enter Password

.....



Choose Cloud Foundry Org \*

`4e198f75trial`



Choose Cloud Foundry Space \*

`dev`



Create a new HDI service instance?



Yes



No

Please enter a unique and non-existing service instance name

`CalculationViews-hdidb-ws-2fxz5`

Use the default database instance of the selected Cloud Foundry space?



Yes

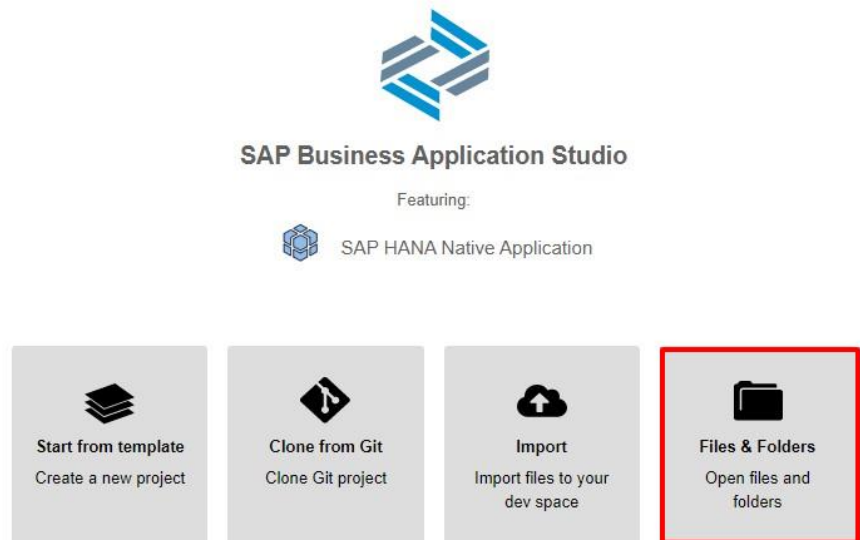


No

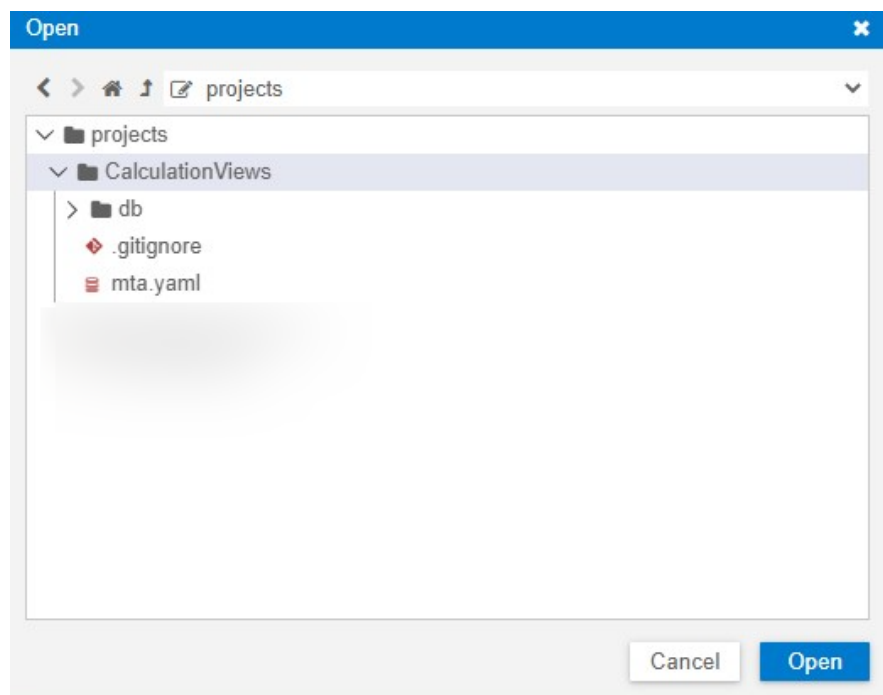
< Back

Finish

54. Click **Files & Folders** tile



55. Go to projects  
56. Select the project  
**CalculationViews**  
57. Click **Open**

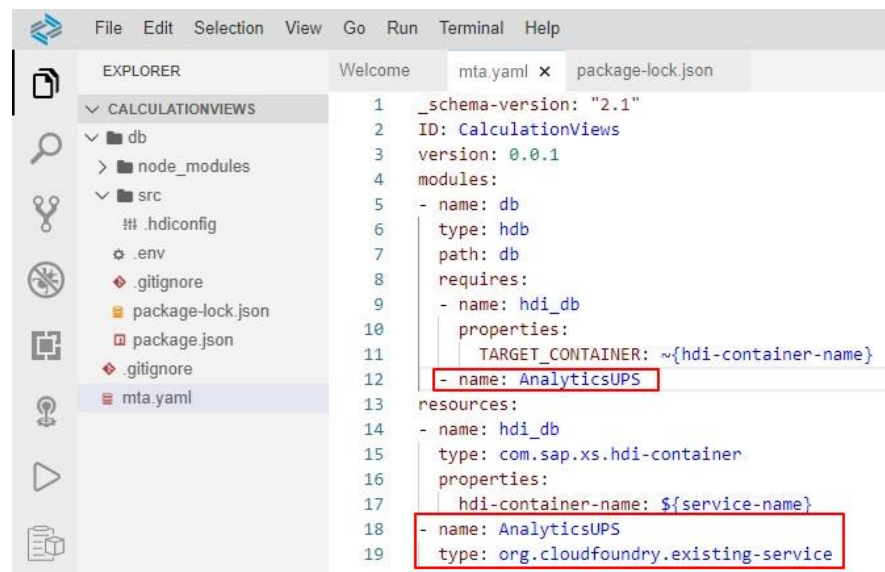


58. The project will get some basic structure as shown in the screenshot

59. Click on **mta.yaml**

60. In code editor, you need to add the User-Provided Service to the project using the following code:

- name: AnalyticsUPS  
type: org.cloudfoundry.existing-service



The screenshot shows the SAP HANA Cloud IDE interface. The Explorer on the left displays the project structure: CALCULATIONVIEWS, db, node\_modules, src, .hdi.config, .env, .gitignore, package-lock.json, package.json, .gitignore, and mta.yaml. The Editor on the right shows the content of mta.yaml with line numbers 1 through 19. The configuration includes a schema version, ID, version, and modules. A red box highlights the module definition for 'AnalyticsUPS' under the 'modules' section, and another red box highlights the resource definition for 'AnalyticsUPS' under the 'resources' section.

```

1  _schema-version: "2.1"
2  ID: CalculationViews
3  version: 0.0.1
4  modules:
5    - name: db
6      type: hdb
7      path: db
8      requires:
9        - name: hdi_db
10         properties:
11           TARGET_CONTAINER: ~{hdi-container-name}
12     - name: AnalyticsUPS
13  resources:
14    - name: hdi_db
15      type: com.sap.xs.hdi-container
16      properties:
17        hdi-container-name: ${service-name}
18    - name: AnalyticsUPS
19      type: org.cloudfoundry.existing-service

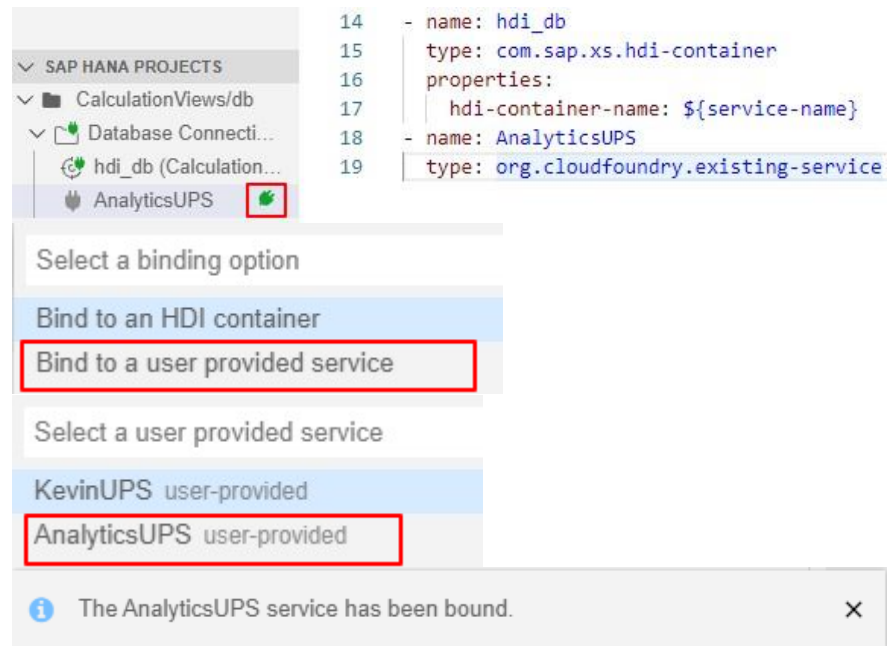
```

61. Expand the SAP HANA Projects -> CalculationViews/db -> Database Connections -> AnalyticsUPS

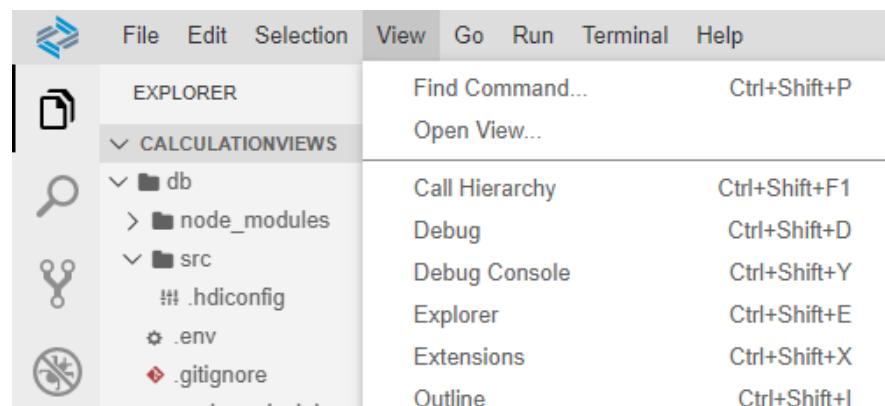
62. Click **Bind**

63. Select **Bind to a user provided service**

64. Select **AnalyticsUPS**



65. Go to **View -> Find Command...**



66. Search for **Create SAP HANA Database Artifact**  
67. Click on it

>create sap hana database

SAP HANA: **Create SAP HANA Database Artifact**

68. Select **Grants** as Artifact Type  
69. Enter **Artifact name**  
70. Click **Create**

Welcome

Create SAP HANA Database Artifact x

## Create SAP HANA Database Artifact

Choose artifact that you need for your project.

Choose path where you want to create the artifact.



\*

/home/user/projects/CalculationView 

Choose Database Version \*

HANA Cloud



Artifact Type  \*

Grants (hdbgrants)



Artifact name  \*

Analytics

Create

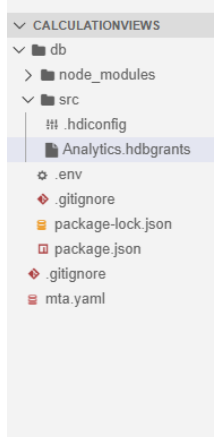
71. Click on **Analytics.hdbgrants**  
72. Delete the default grants in code editor, and enter the following JSON:

```
{
  "AnalyticsUPS": {
    "object_owner": {
      "schema_privileges": [
        {
          "reference": "ANALYTICSADMIN",
          "privileges_with_grant_option": ["SELECT", "EXECUTE"]
        }
      ]
    }
  }
}
```

EXPLORER

Welcome

Analytics.hdbgrants x

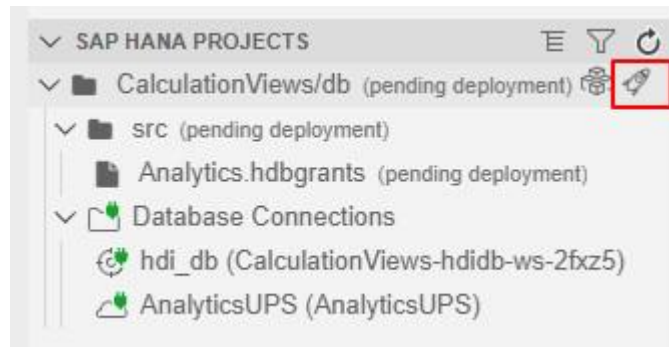


```
1 {
2   "AnalyticsUPS": {
3     "object_owner": {
4       "schema_privileges": [
5         {
6           "reference": "ANALYTICSADMIN",
7           "privileges_with_grant_option": ["SELECT", "EXECUTE"]
8         }
9       ]
10    },
11    "application_user": {
12      "schema_privileges": [
13        {
14          "reference": "ANALYTICSADMIN",
15          "privileges_with_grant_option": ["SELECT"]
16        }
17      ]
18    }
19  }
20 }
```



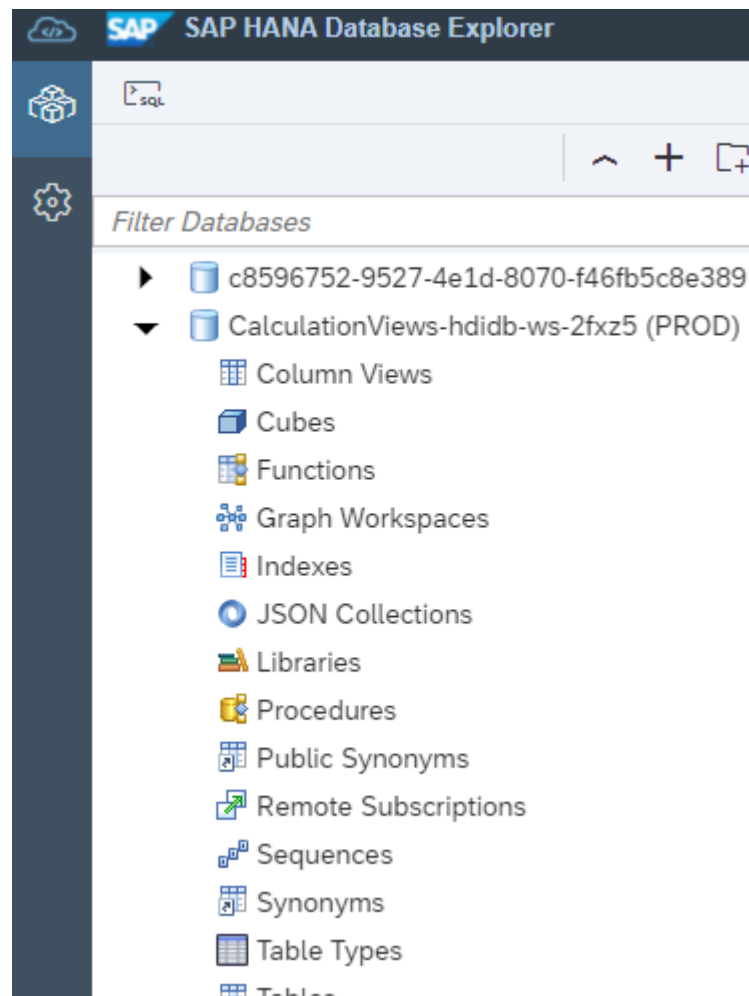
```
    ],  
    "application_user":  
{  
    "schema_privileg  
es": [  
    {  
    "reference": "ANA  
LYTICSADMIN",  
    "privileges_with  
_grant_option": ["SELECT"]  
    }  
    ]  
    }  
}
```

73. In SAP HANA PROJECTS, we can the project is in pending deployment. Let's deploy the HDI container by click the rocket button



74. Navigate to SAP HANA Database Explorer, the HDI container should show up in the list of databases.

Note: In case the HDI container didn't show up, you can manually add the HDI container.



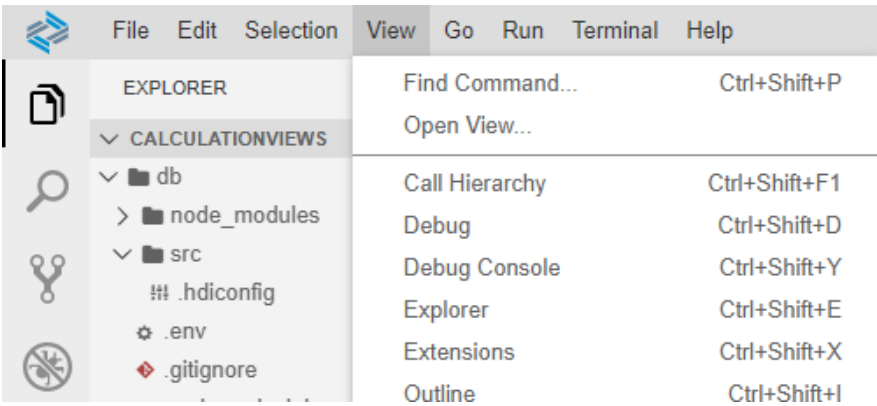
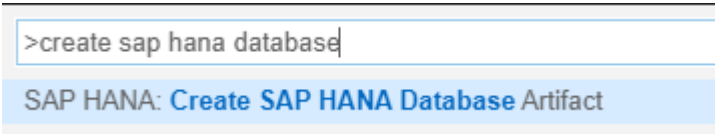
### 3.5 Create Calculation View

Now we can create a calculation view with the virtual tables in HDI container.

When Calculation Views are executed in SAP HANA Cloud, the query is first optimized in a special engine called "Calculation Engine". This has a drawback when the query is included in a larger SQL query, because in this case two different optimization processes are involved: One for the SQL query part and the other one for the Calculation View part. This can lead to inefficiencies between the different optimization processes.

To avoid these inefficiencies, you can set the parameter "Execute in" to "SQL Engine" in Calculation Views. This can push as much processing as possible to target data source. One exception is that this setting is active for Cubes with Star Joins automatically even though the "Execute in" option is not set to "SQL Engine". MDS queries often benefit from setting the flag to "SQL Engine" in the topmost Calculation Views that are queried by MDS. SAP Analytics Cloud utilizes MDS queries to retrieve the data from HANA Calculation Views.

Explanation	Screenshot
-------------	------------

<p>75. Go to <b>View</b> -&gt; <b>Find Command...</b></p>	 <p>The screenshot shows the SAP HANA Cloud IDE interface. The 'View' menu is open, displaying options: Find Command... (Ctrl+Shift+P), Open View..., Call Hierarchy (Ctrl+Shift+F1), Debug (Ctrl+Shift+D), Debug Console (Ctrl+Shift+Y), Explorer (Ctrl+Shift+E), Extensions (Ctrl+Shift+X), and Outline (Ctrl+Shift+I). The Explorer view on the left shows a project structure with folders 'db' and 'src', and files '.hdiconfig', '.env', and '.gitignore'.</p>
<p>76. Search for <b>Create SAP HANA Database Artifact</b> 77. Click on it</p>	 <p>The screenshot shows the search results in the SAP HANA Cloud IDE. The search bar contains the text '&gt;create sap hana database'. Below the search bar, a result is displayed: 'SAP HANA: Create SAP HANA Database Artifact'.</p>

78. Select **Calculation View** as  
Artifact Type

79. Enter **Artifact name**

80. Click **Create**

## Create SAP HANA Database Artifact

Choose artifact that you need for your project.

Choose path where you want to create the artifact.

ⓘ \*

/home/user/projects/CalculationView

Choose Database Version \*

HANA Cloud

Artifact Type ⓘ \*

Calculation View (hdbcalculationview) ▾

Artifact name ⓘ \*

Sales

Label ⓘ

Sales

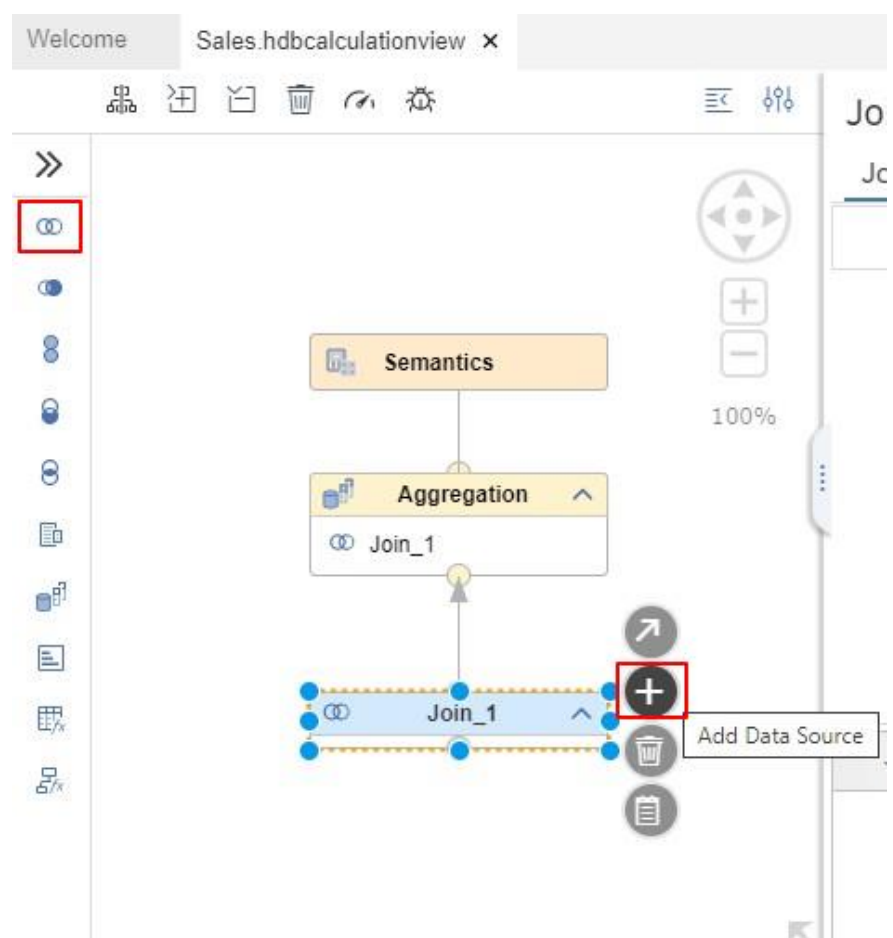
Data Category ⓘ \*

CUBE ▾

Use Star Join? ⓘ

Create

81. Create JOIN  
82. Click **Add Data Source**



83. Select **AnalyticsUPS** as the External Services  
84. Search "sales"  
85. Select virtual table **V\_Sales** from schema **ANALYTICSADMIN**  
86. Click **Create Synonym**



Find Data Sources

Search for an object name

All Types Selected  ⓧ

External Services:  ⌵

Results (2)      Selected Objects (1)

<input type="checkbox"/>	Type	Name	Schema	Synonym
<input type="checkbox"/>		V.Sales	DBADMIN	
<input checked="" type="checkbox"/>		V_Sales	ANALYTICSADMIN	
<input type="checkbox"/>				
<input type="checkbox"/>				
<input type="checkbox"/>				

Previous **Create Synonym** Finish Cancel

87. Enter a **Synonym Name**  
 88. Click **Finish**  
 89. Repeat the steps 82-88 to add another data source for virtual table **V\_Region**

**Note:** A Synonym file will be generated under src folder.

Find Data Sources

Synonym Name	Object Name	Schema	Use Ex...
calculationview.db::V_Sales	V_Sales	ANALYTICSA...	<input type="checkbox"/>

☐ Synonym for HDI Container    ☐ Generate .hdbgrants File

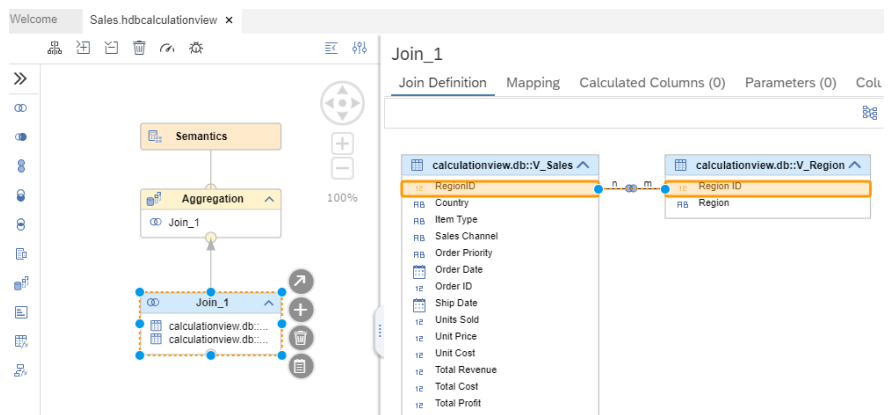
*i* Object Owner Role :

*i* Application User Role :

*i* Creating synonyms requires building hdbsynonym and hdbgrantor files after creating or modifying them.

90. JOIN 2 tables using column **Region ID**



91. Click on **Semantics** -> **View Properties** -> **Advanced**  
 92. Select Execute In **SQL Engine**

**Note:** The parameter Execute in 'SQL Engine' improves the performance of HANA calculation views.

Semantics ⌵ ✕

View Properties Columns (14) Hierarchies (0) Parameters (0)

General **Advanced** Static Cache

⚠ These properties may affect the output data. Set them cautiously.

☐ Propagate Instantiation to SQL Views

☐ Analytic View Compatibility Mode

☐ Ignore Multiple Outputs For Filter

Pruning Configuration Table:  🔍 + ✎

Execute In: **SQL Engine** ⌵

Execution Hints: ⌵ + 🗑 📄

<input type="checkbox"/>	Name	Value
--------------------------	------	-------

93. Click **Hierarchies**  
 94. Enter **Name**  
 95. Add 2 Nodes:  
     Level1: **Region**  
     Level2: **Country**

Semantics ⌵ ➤

View Properties Columns (14) **Hierarchies (1)** Parameters (0)

General

\*Name:

Label:

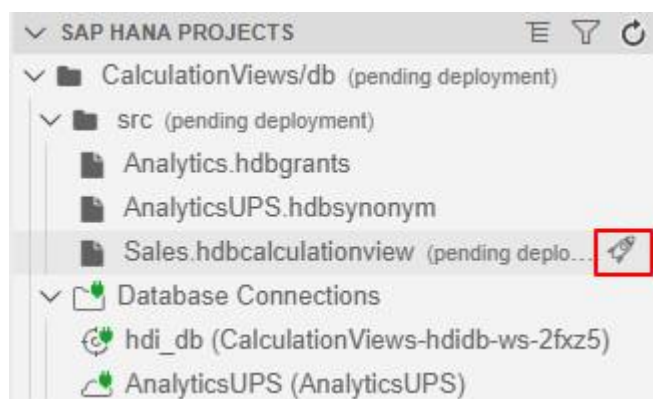
Notes:

Node Style:  ⌵

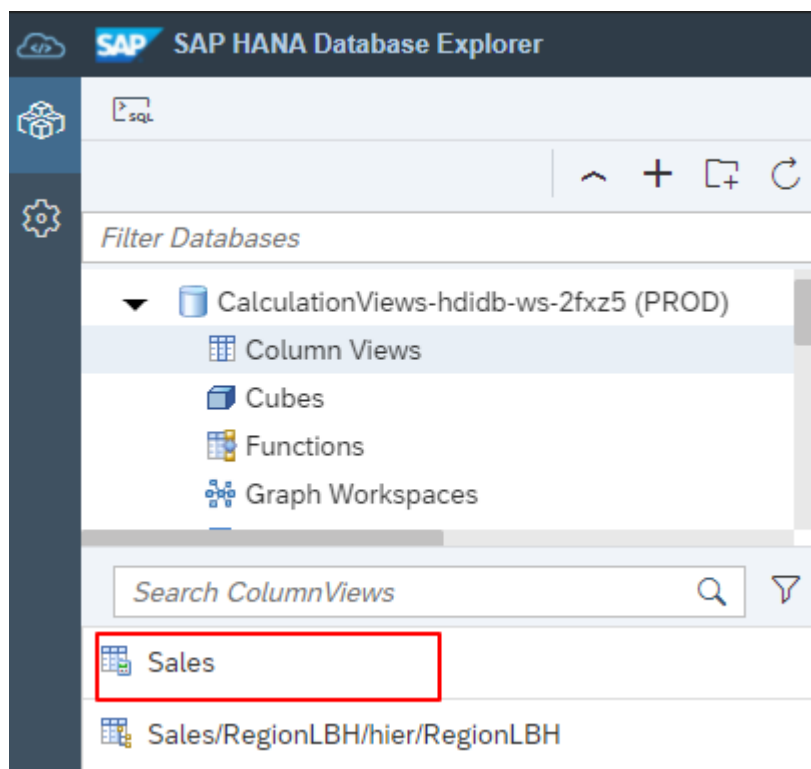
⌵ Nodes

<input type="checkbox"/>	Level	Column	Level Type	Order By	Sort Direction
<input type="checkbox"/>	1	Region <span>📄</span>	REGULAR <span>⌵</span>	Region <span>📄</span>	Ascending <span>⌵</span>
<input type="checkbox"/>	2	Country <span>📄</span>	REGULAR <span>⌵</span>	Country <span>📄</span>	Ascending <span>⌵</span>

96. **Deploy** the calculation view into HDI Container



97. Congratulations! You have created a calculation view in SAP HANA Cloud



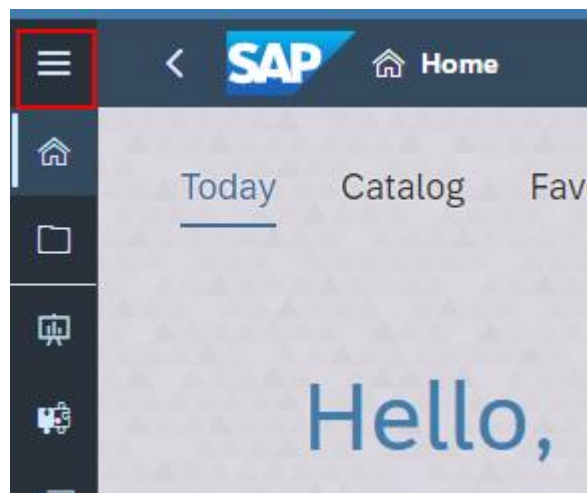
### 3.6 Create a story in SAP Analytics Cloud

Explanation	Screenshot
-------------	------------

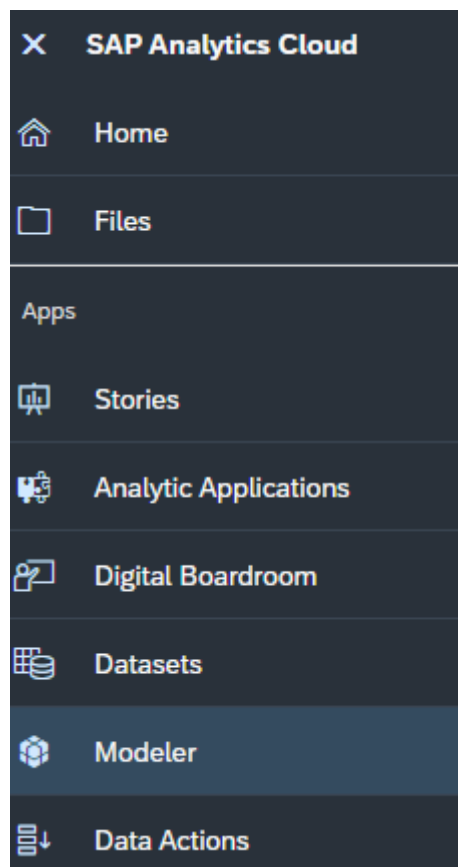


98. Logon to your SAP Analytics Cloud tenant.

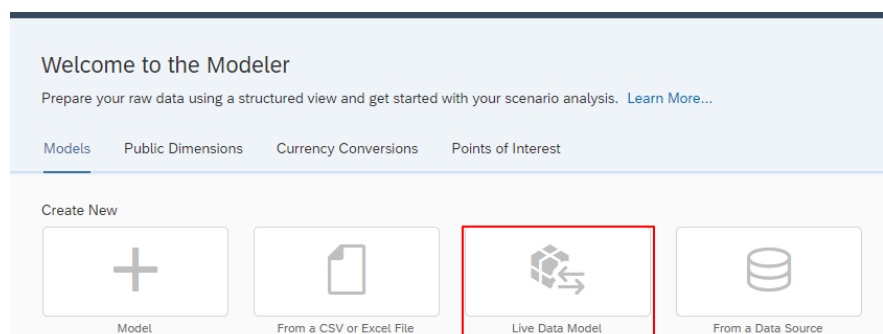
99. Click on the Main Menu.



100. Select **Modeler**



101. Click on **Live Data Model**



102. Enter the following information  
 System Type: **SAP HANA**  
 Connection: **HANACloud** or your HANA Cloud connection  
 Data Source: **Sales**

103. Click **OK**.

Create Model From Live Data Connection

Select Live Data Connection and Data Source





\*System Type:

\*Connection:

\*Data Source:

**OK** Cancel

104. **Save** the model.


Data sources     All Dimensions

**T**

	ID	Description	Aggregation Ty...	Exception Agg.
1	Total_Cost	Total_Cost	SUM	
2	Total_Profit	Total_Profit	SUM	
3	Total_Revenue	Total_Revenue	SUM	
4	Unit_Cost	Unit_Cost	SUM	
5	Unit_Price	Unit_Price	SUM	
6	Units_Sold	Units_Sold	SUM	
7				

105. Provide the new Model a name and save it in a directory.

Save Model

My Files / Public / KE / SQL Server  All files

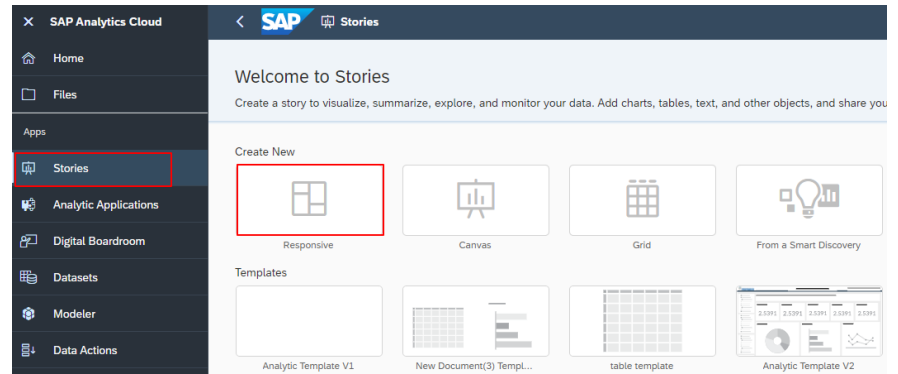
Name	Description	Owner
60MillionSales	-	Kevin Li
CopyofOneMillion	-	Kevin Li
LargeSalesDataSet	-	Kevin Li
SOLDPRICE	-	Kevin Li
SQLCALVIEW	-	Kevin Li
SQLUNIONVIEW	-	Kevin Li

\*Name

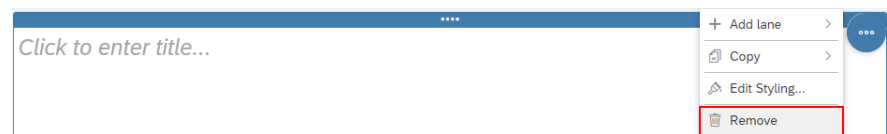
Description

**OK** Cancel

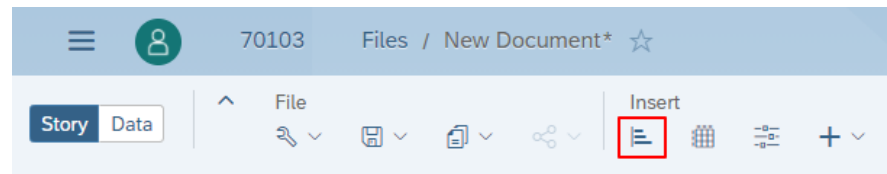
106. Select **Stories**  
107. Select **Responsive**



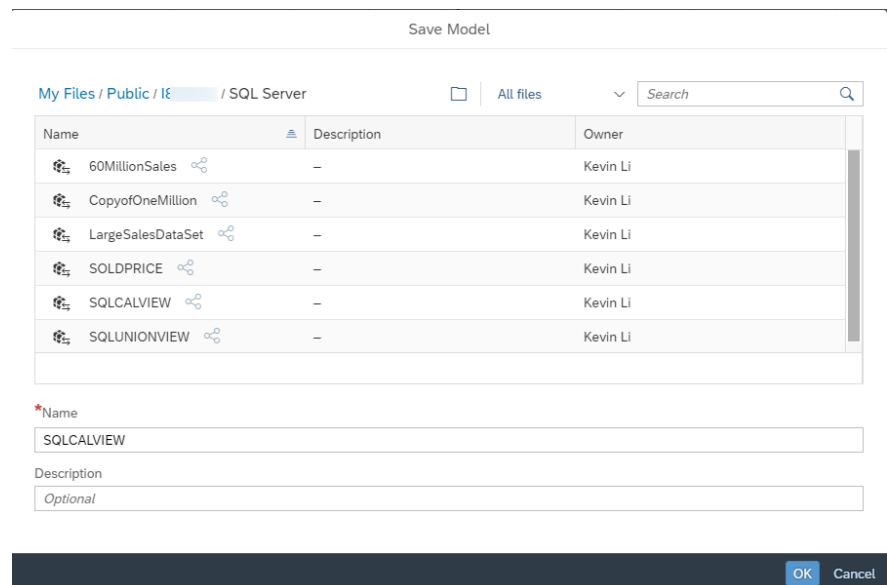
108. Remove one of the responsive pages.



109. Insert a Chart.

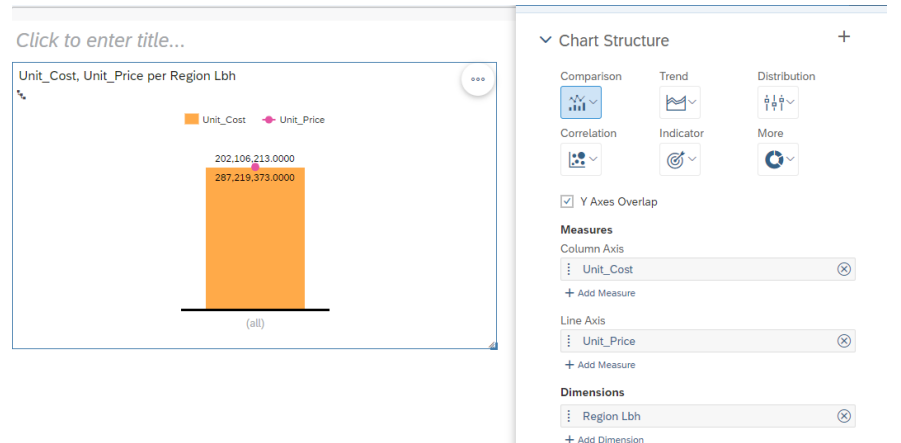


110. Select the model.



111. Select Unit\_Cost and Unit\_Price as the measures, and the hierarchy Region\_Lbh as the Dimension.

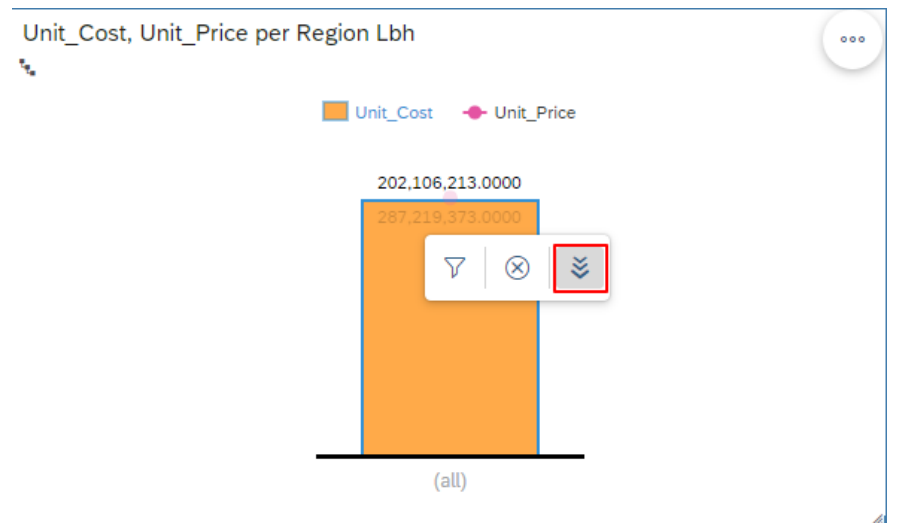
112. The chart looks like the screenshot.



113. Click on the bar.

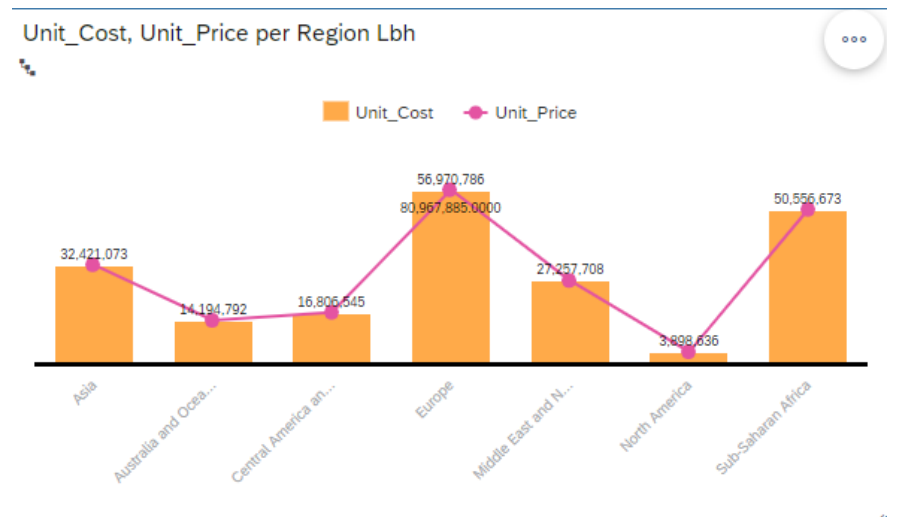
114. Click on the Drill Down.

115. **Note:** This drill-down analysis is enabled by the hierarchy created in HANA. This is an added value that HANA can build hierarchies on flat data structure in source tables.



116. The chart will show based on the regions.

117. Save the story.

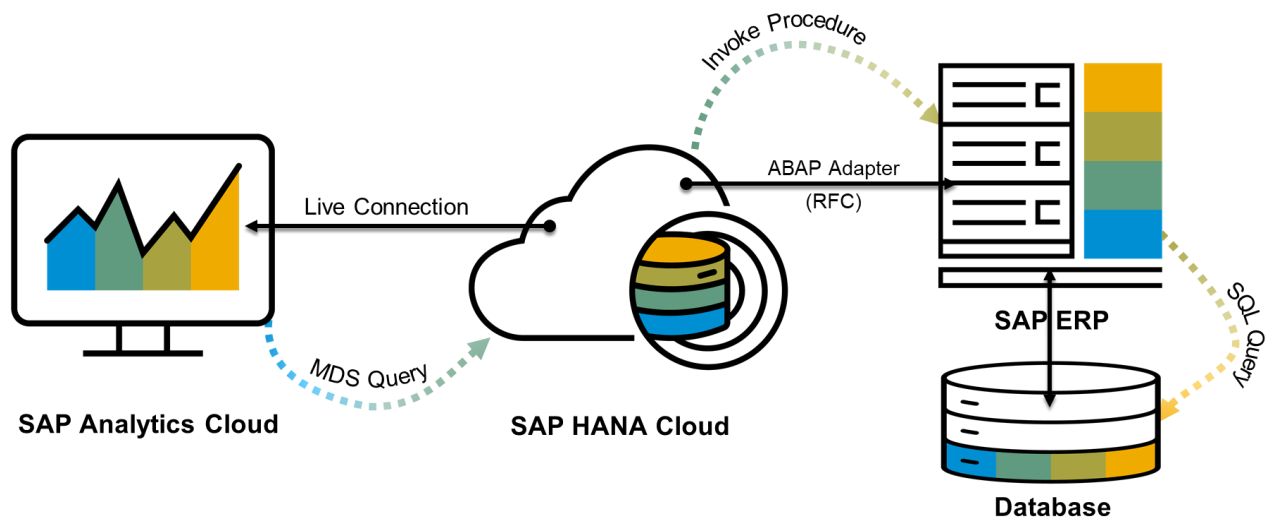




## 4. Setup Connection to SAP ERP

In this section, we will setup SAC live data connection to SAP ERP using SAP HANA SDI (Particularly, S/4HANA on-premise is used in this section, but the same setup can generally apply to SAP ECC 6.0). In the previous sections, we have configured SAC live data connection to Microsoft SQL Server using SAP HANA SDI. SDI can also connect to various types of SAP data, for example, SAP ERP systems. Here, the ABAP adapter is used to retrieve data from virtual tables through RFC for ABAP tables and ODP extractors.

Currently, SAP Analytics Cloud only has acquired data connection to SAP ERP systems. With ABAP adapter, SAC can live connect to SAP ERP systems. The system landscape is started with installing and configuring Data Provisioning Agent. The ABAP adapter not only enables virtual tables mapping to the ABAP tables, but virtual procedures can also be mapped to the BAPIs in SAP ERP. Both virtual tables and virtual procedures can be used to create calculation views in SAP HANA Cloud, and then SAC consumes the calculation views through live data connection to SAP HANA Cloud. The system landscape is setup as below.



Since ABAP adapter exposes three types of data: ABAP tables, BAPIs, and ODP extractors to SAP HANA, it is important to decide which types of data can be used in this scenario. The ODP extractors includes CDS views, SAP BW, and SAP HANA, SAC already has live data connection to these data sources. Hence, ODP extractors can be excluded in this scenario. Again, the rule of thumb is to reduce the data transfer across the network. In this case, it is to push down as much processing as possible to SAP ERP's database. SAP AS ABAP cannot process SQL query from SAP HANA Cloud, so SAP HANA Cloud cannot push down query processing to SAP ERP. That means the raw query result on virtual tables mapped to ABAP tables need to be returned to SAP HANA Cloud, which will cause poor performance if the size of the ABAP tables is too large. Thus, the only option is to use BAPIs in SAP ERP. BAPIs can push the query down to SAP ERP's database if coded accordingly and return the aggregated result set to SAP HANA Cloud.

### 4.1 Register ABAP Adapter in SAP Data Provisioning Agent

The ABAP adapter retrieves data from virtual tables through RFC for ABAP tables and ODP extractors. The ABAPAdapter is a build-in adapter in Data Provisioning Agent, so we can directly register the adapter without downloading any drivers.

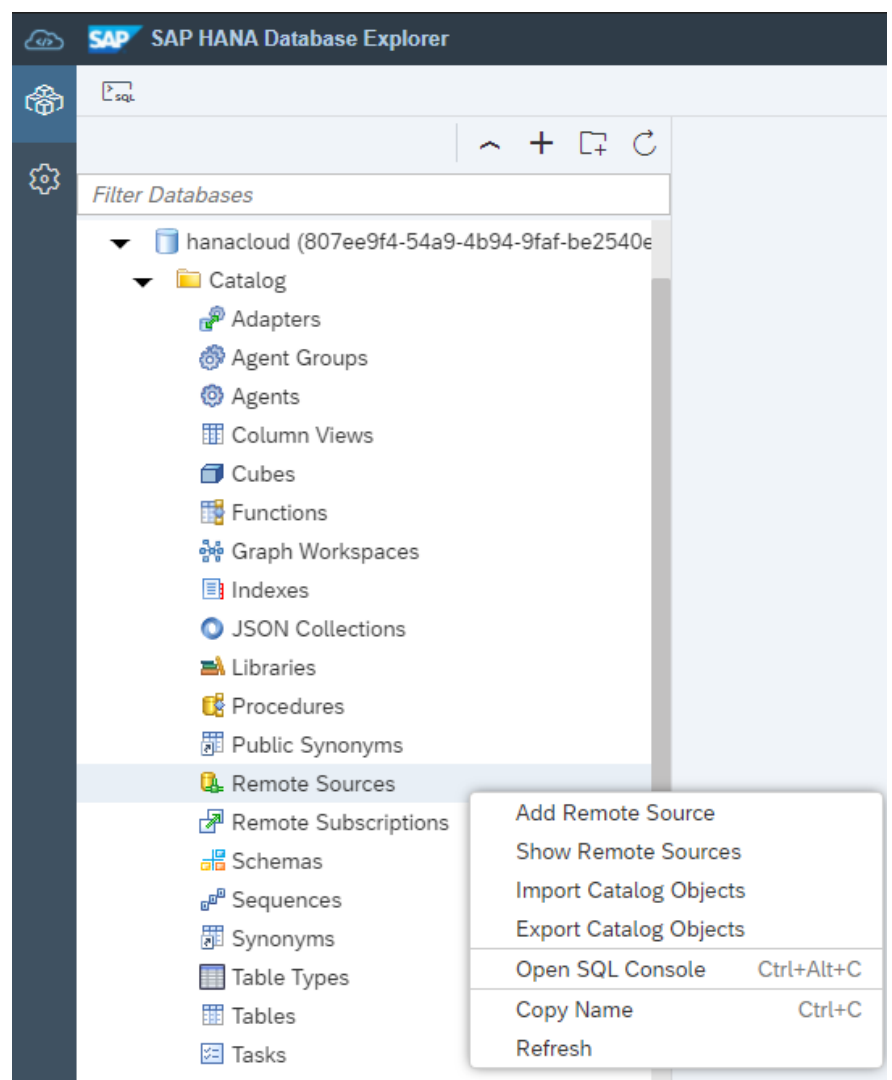
Explanation	Screenshot
118. Select option <b>8. Adapter Registration</b>	<pre> ***** DPAgent Configuration Tool ***** 1. Agent Status 2. Start or Stop Agent 3. Agent Preferences 4. Remote Source Credentials 5. SSL Keystores &amp; Settings 6. SAP HANA Connection 7. Agent Registration 8. Adapter Registration 9. Custom Adapters 10. Agent &amp; Adapter Versions q. Quit b. Back ***** Enter Option:7 ***** Agent Registration ***** 1. Register Agent 2. Unregister Agent q. Quit b. Back </pre>
119. Select option <b>2. Register Adapter</b> 120. Enter adapter name: <b>ABAPAdapter</b>  The adapter name must match the name displayed by the <b>Display Adapters</b> option.	<pre> ***** Adapter Registration ***** 1. Display Adapters 2. Register Adapter 3. Unregister Adapter q. Quit b. Back ***** Enter Option:2 Enter adapter name: ABAPAdapter Adapter 'ABAPAdapter' successfully registered. </pre>

## 4.2 Add SAP ERP as a Remote Source

Now, we will connect to SAP ERP using ABAPAdapter in Data Provisioning Agent. In this document, S/4HANA is used as an SAP ERP system, but an older SAP ERP system (SAP ECC) will also work in this case.

Explanation	Screenshot
-------------	------------

121. Navigate to **SQL Console**
122. Go to **Catalog** -> **Remote Source**
123. Right click on the **Remote Source** and **Add Remote Source**





124. Fill in the remote source details:

Source Name  
Adapter Name: ABAPAdapter  
Application Server  
Client  
Instance Number  
Credentials Mode: Technical user  
User Name  
Password

125. **Create** the remote source.

#### Add Remote Source

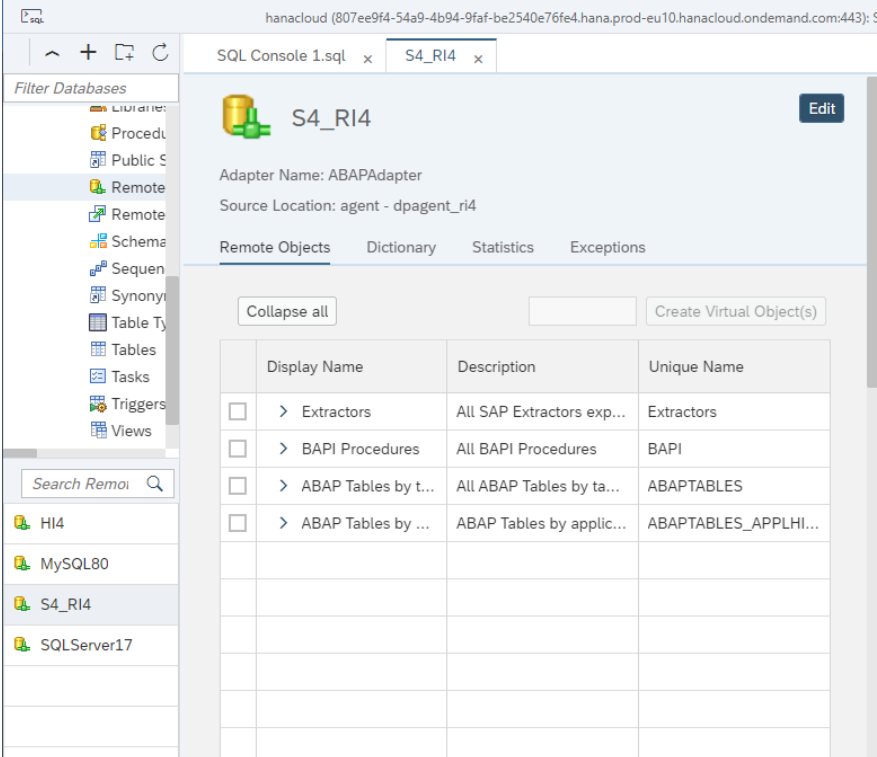
*Source Name	S4_RI4
*Adapter Name	ABAPAdapter
Source Location	AGENT: dpagent_ri4

Property	Value
▼ ConnectionInfo	
Connection Type	Custom Application Server
*Application Server	yourSAPERPhost
Message Server	
Message Server Port	
*Client	400
*Instance Number	00
System ID	
Server Group	
Connections Pool Size	10
Connections Limit	100

Create

Cancel

126. Now, you can see the SAP ERP system in the list of Remote Source, open the remote source **S4\_RI4**, BAPI Procedures, along with ABAP tables and ODP extractors are the data available from SAP S/4HANA



The screenshot shows the SAP HANA Cloud interface. On the left, a sidebar lists various database objects, with 'Remote Sources' selected. The main area displays the configuration for the remote source 'S4\_RI4'. The 'Adapter Name' is 'ABAPAdapter' and the 'Source Location' is 'agent - dpagent\_ri4'. Below this, the 'Remote Objects' tab is active, showing a table of objects available from the SAP S/4HANA system.

	Display Name	Description	Unique Name
<input type="checkbox"/>	> Extractors	All SAP Extractors exp...	Extractors
<input type="checkbox"/>	> BAPI Procedures	All BAPI Procedures	BAPI
<input type="checkbox"/>	> ABAP Tables by t...	All ABAP Tables by ta...	ABAPTABLES
<input type="checkbox"/>	> ABAP Tables by ...	ABAP Tables by applic...	ABAPTABLES_APPLHI...

## 4.3 Create Custom BAPIs in SAP ERP

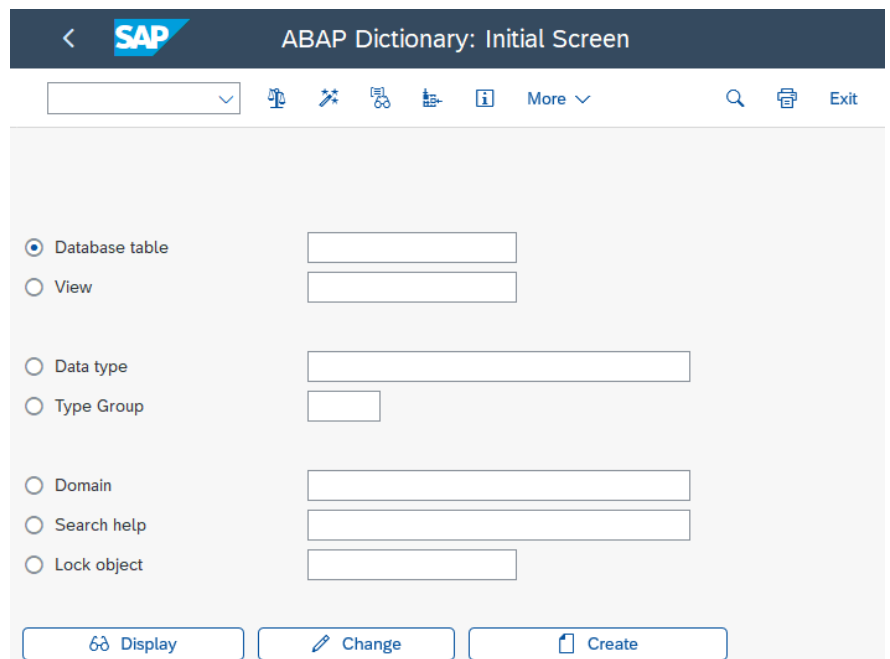
In this section, we will first create a structure in SAP NetWeaver AS ABAP, and then create a function module to return the aggregated data back from SAP ERP's database. Lastly, we will add this function module into a custom BAPI and publish the custom BAPI.

### Explanation

### Screenshot

127. Logon S/4HANA in  
SAPGUI.

128. Go to transaction SE11.



ABAP Dictionary: Initial Screen

Database table

View

Data type

Type Group

Domain

Search help

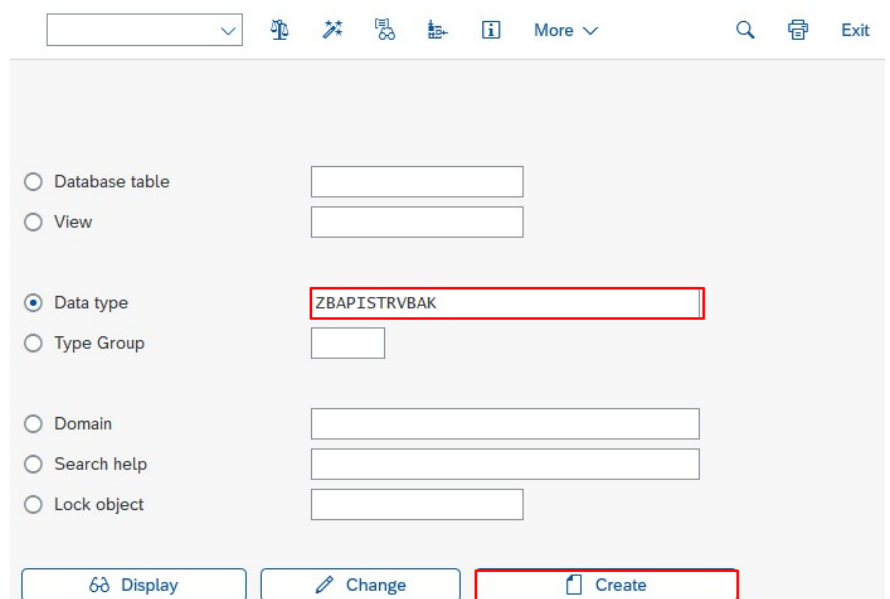
Lock object

Display Change Create

129. Select Data type.

130. Enter a custom BAPI name.

131. Click Create.



Database table

View

Data type

Type Group

Domain


Search help

Lock object

Display Change Create

132. Select Structure.

133. Click OK.



Create Type ZBAPISTRVBAK

Data element ☐

Structure ☒

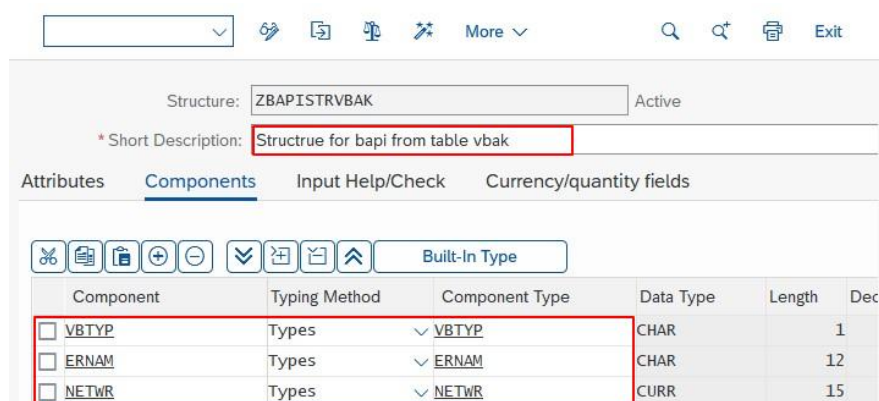
Table type ☐

OK X

134. Enter a Short Description.

135. Enter the Component (Field Name) and Component Type (Data Element) as shown in the screenshot.

**Note:** Here the component names start with Z as per the SAP recommendation. Let's use data elements that we have already created in the database table VBAK.



Structure: ZBAPISTRVBAK Active

\* Short Description: Structrue for bapi from table vbak

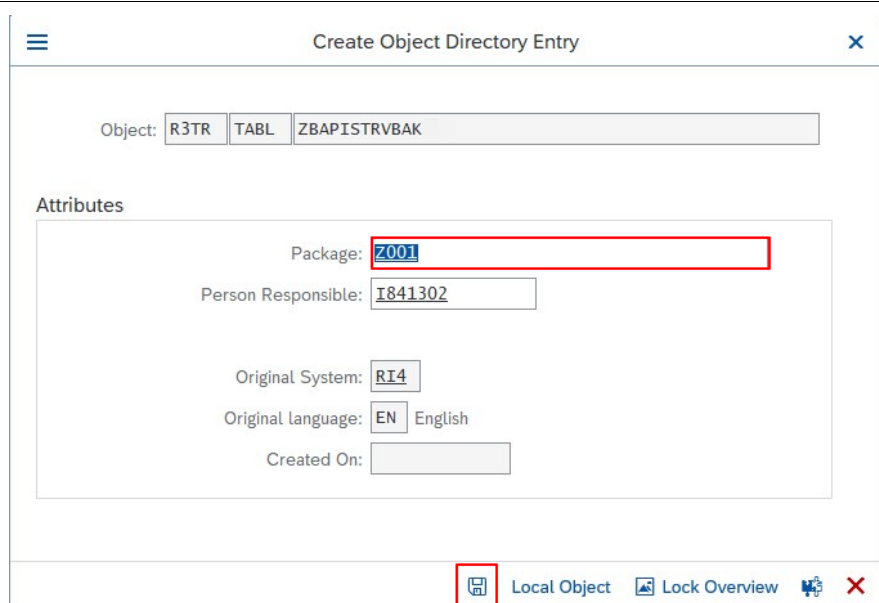
Attributes Components Input Help/Check Currency/quantity fields

Built-In Type

Component	Typing Method	Component Type	Data Type	Length	Dec
<input type="checkbox"/> VBTYP	Types	▼ VBTYP	CHAR	1	
<input type="checkbox"/> ERNAM	Types	▼ ERNAM	CHAR	12	
<input type="checkbox"/> NETWR	Types	▼ NETWR	CURR	15	

136. Enter the Package Z001.

137. Save the structure.



Create Object Directory Entry

Object: R3TR TABL ZBAPISTRVBAK

Attributes

Package: Z001

Person Responsible: I841302

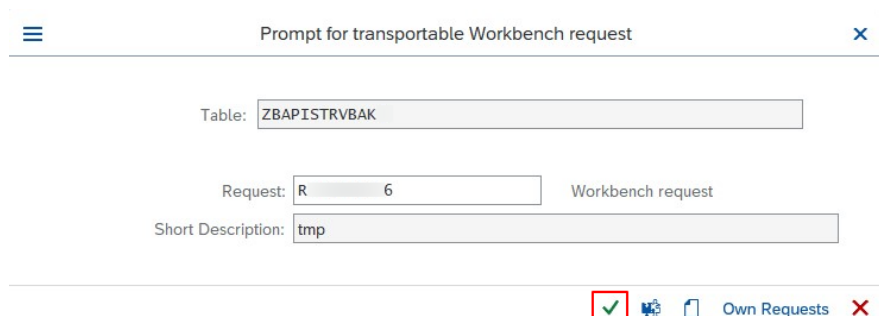
Original System: RI4

Original language: EN English

Created On:

Local Object Lock Overview

138. Click OK.



Prompt for transportable Workbench request

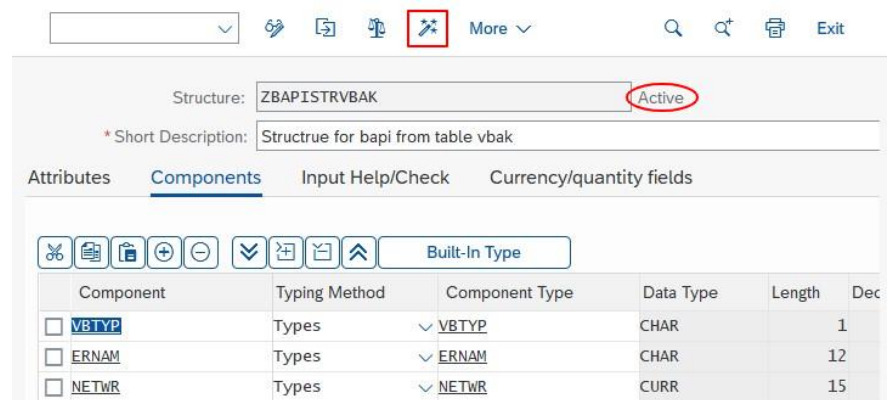
Table: ZBAPISTRVBAK

Request: R 6 Workbench request

Short Description: tmp

Own Requests

139. Activate the structure.



Structure: ZBAPISTRVBAK **Active**

\* Short Description: Structrue for bapi from table vbak

Attributes **Components** Input Help/Check Currency/quantity fields

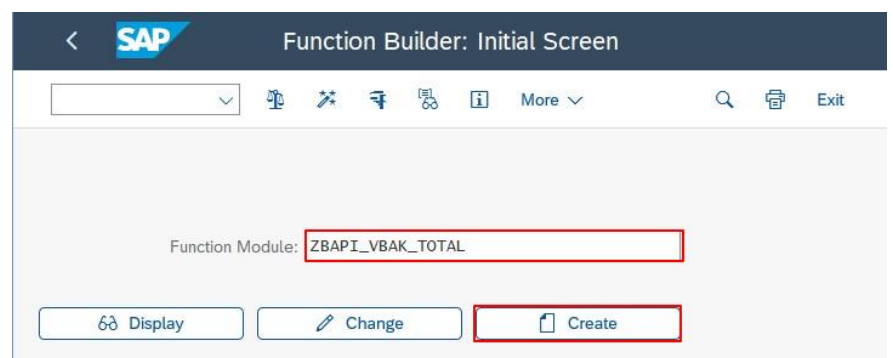
Component Typing Method Component Type Data Type Length Dec

<input type="checkbox"/> VB Typ	Types	VB Typ	CHAR	1	
<input type="checkbox"/> ERNAM	Types	ERNAM	CHAR	12	
<input type="checkbox"/> NETWR	Types	NETWR	CURR	15	

140. Go to transaction SE37.

141. Enter a new Function Module name.

142. Click Create.



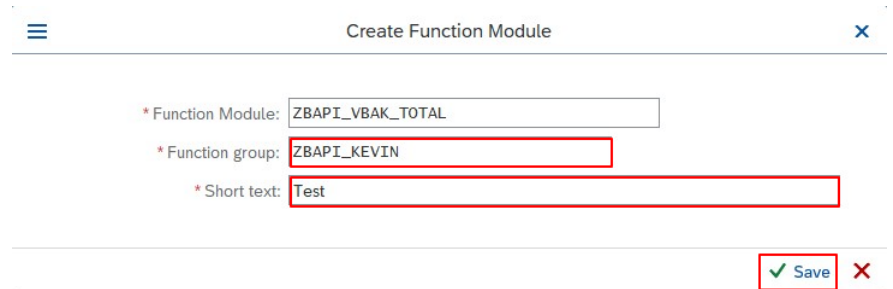
Function Module: ZBAPI\_VBAK\_TOTAL

Display Change **Create**

143. Enter Function group.

144. Enter a Short description.

**Note:** If function group is not already made, you could create it by following steps 39-41.



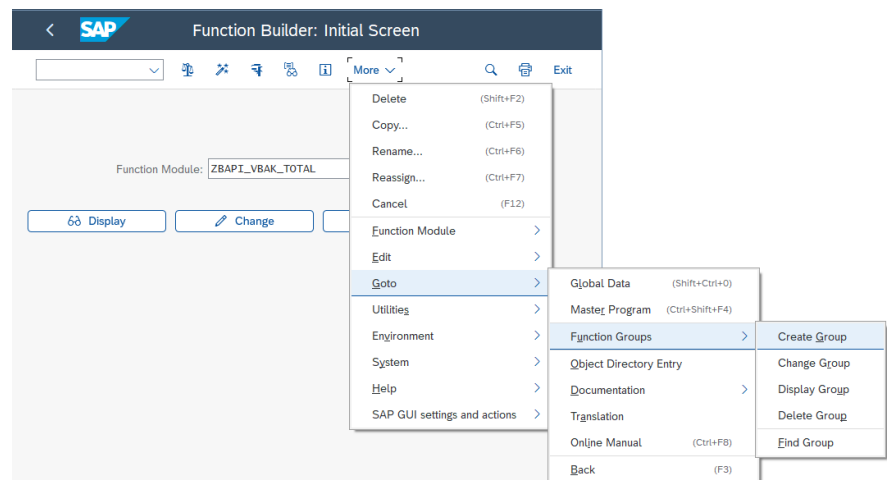
\* Function Module: ZBAPI\_VBAK\_TOTAL

\* Function group: ZBAPI\_KEVIN

\* Short text: Test

**Save**

145. Go to More -> Goto -> Function Groups -> Create Group.



Function Module: ZBAPI\_VBAK\_TOTAL

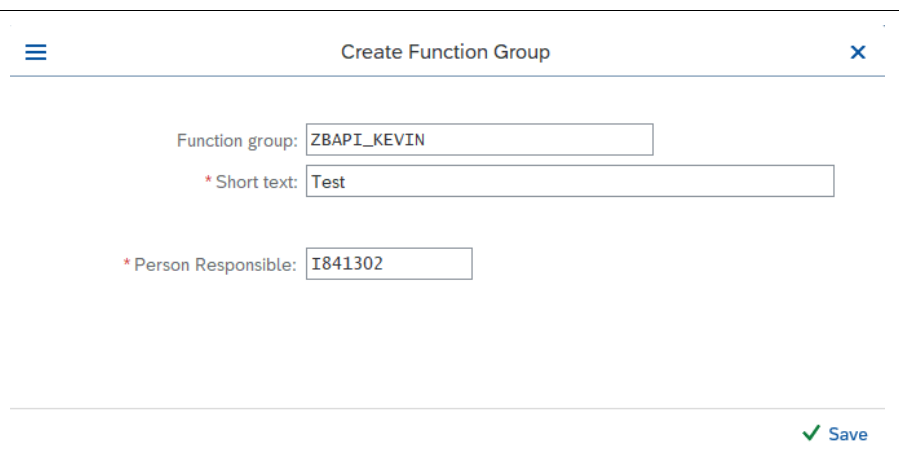
Display Change

More

- Delete (Shift+F2)
- Copy... (Ctrl+F5)
- Rename... (Ctrl+F6)
- Reassign... (Ctrl+F7)
- Cancel (F12)
- Function Module >
- Edit >
- Goto >**
  - Global Data (Shift+Ctrl+O)
  - Master Program (Ctrl+Shift+F4)
  - Function Groups >**
    - Create Group**
    - Change Group
    - Display Group
    - Delete Group
    - Find Group
  - Object Directory Entry >
  - Documentation >
  - Translation
  - Online Manual (Ctrl+F8)
  - Back (F3)
- Utilities >
- Environment >
- System >
- Help >
- SAP GUI settings and actions >

146. Enter a Short description.

147. Save the Function Group.



Function group: ZBAPI\_KEVIN

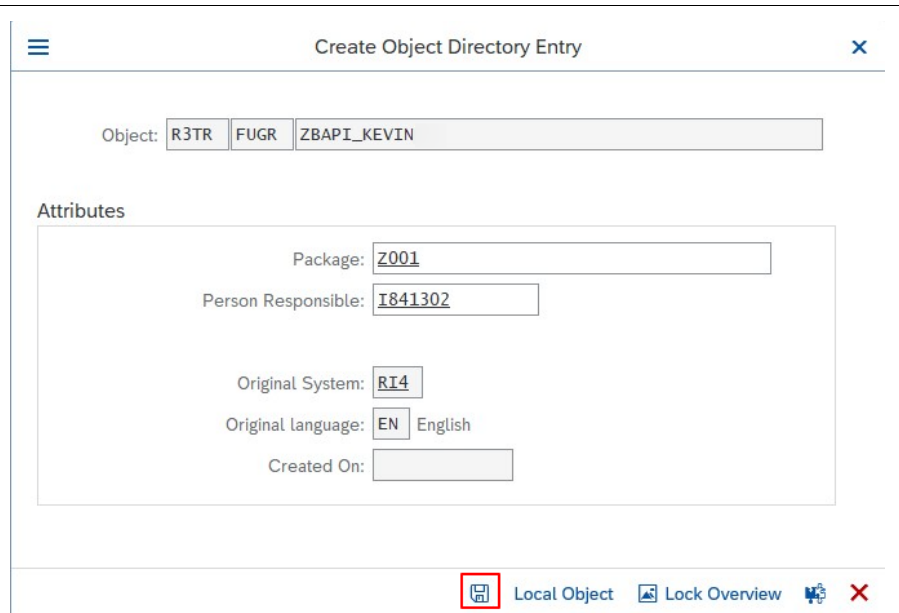
\* Short text: Test

\* Person Responsible: I841302

✓ Save

148. Enter a Package.

149. Click Save.



Object: R3TR FUGR ZBAPI\_KEVIN

Attributes

Package: Z001

Person Responsible: I841302

Original System: RI4

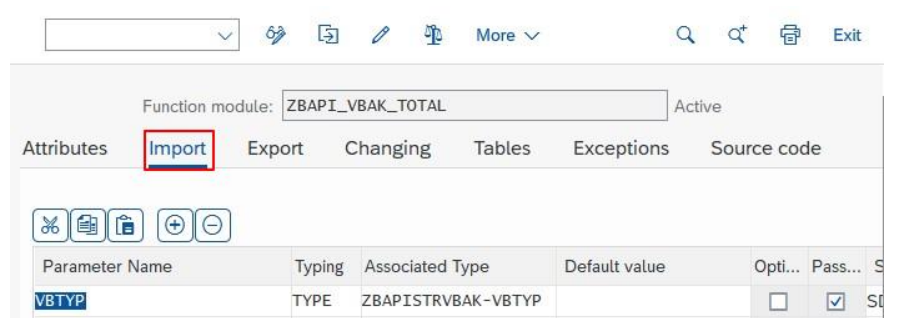
Original language: EN English

Created On:

Local Object Lock Overview

150. Click on Import.

151. Add VB Typ as a import parameter.



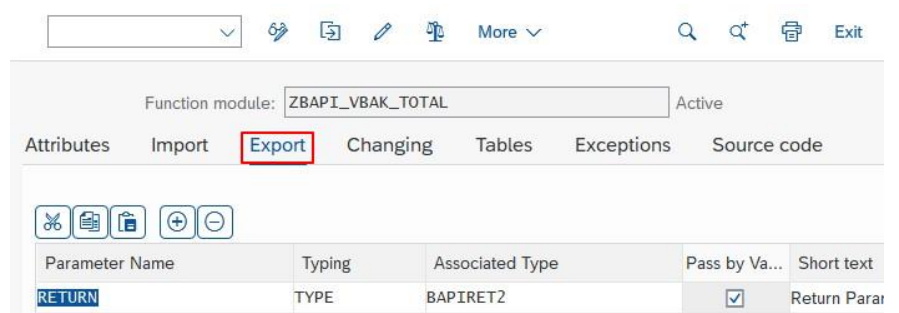
Function module: ZBAPI\_VBAK\_TOTAL Active

Attributes Import Export Changing Tables Exceptions Source code

Parameter Name	Typing	Associated Type	Default value	Opti...	Pass...	S
VB Typ	TYPE	ZBAPISTRVBAK-VB Typ		<input type="checkbox"/>	<input checked="" type="checkbox"/>	St

152. Click on Export.

153. Add RETURN as a export parameter.



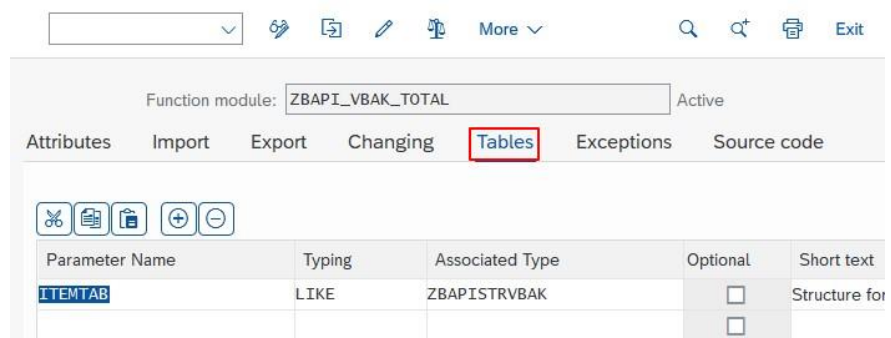
Function module: ZBAPI\_VBAK\_TOTAL Active

Attributes Import Export Changing Tables Exceptions Source code

Parameter Name	Typing	Associated Type	Pass by Va...	Short text
RETURN	TYPE	BAPIRET2	<input checked="" type="checkbox"/>	Return Par

154. Click on Table.

155. Add ITEMTAB as a table, the Associated Type is the structure ZBAPISTRVBAK.



Function module: ZBAPI\_VBAK\_TOTAL Active

Attributes Import Export Changing **Tables** Exceptions Source code

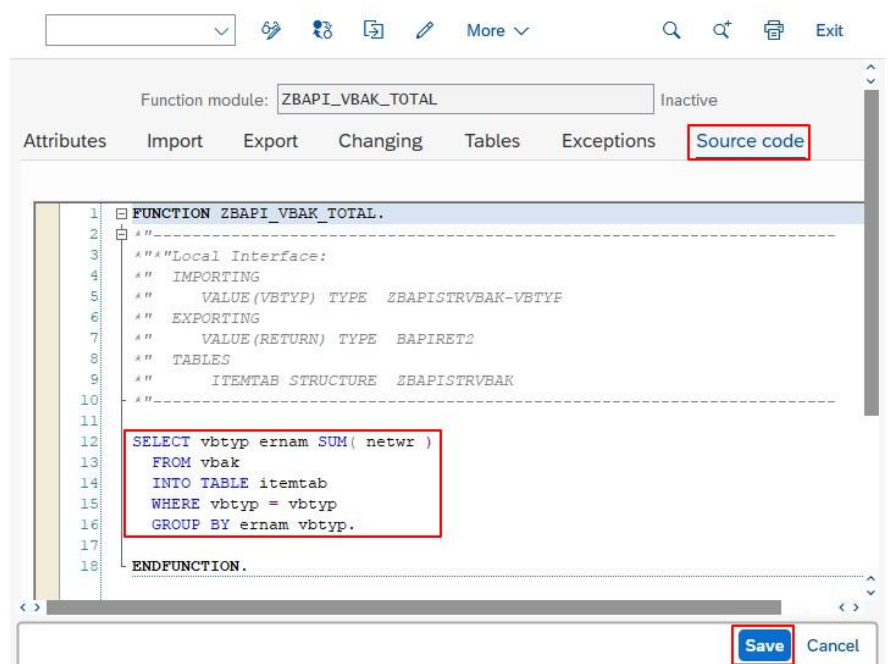
Parameter Name	Typing	Associated Type	Optional	Short text
ITEMTAB	LIKE	ZBAPISTRVBAK	<input type="checkbox"/>	Structure for

156. Click on Source code.

157. Enter the following code:

```
SELECT vbttyp ernam SUM(
  netwr )
FROM vbak
INTO TABLE itemtab
WHERE vbttyp = vbttyp
GROUP BY ernam vbttyp.
```

Note: The above ABAP code is to push the aggregation to the S/4HANA's database. Retrieving the aggregated result is critical for performance.



Function module: ZBAPI\_VBAK\_TOTAL Inactive

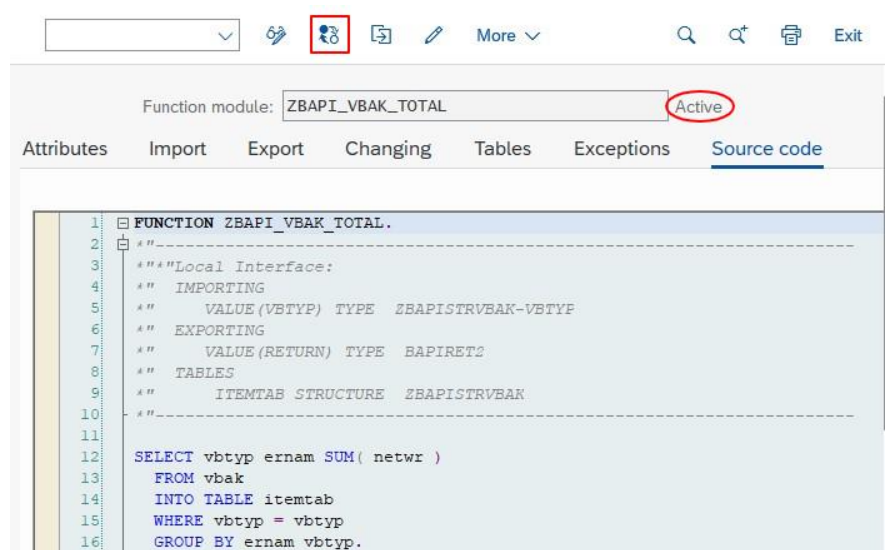
Attributes Import Export Changing Tables Exceptions **Source code**

```
1 FUNCTION ZBAPI_VBAK_TOTAL.
2
3   " "Local Interface:
4   " IMPORTING
5   "   VALUE(VBTYP) TYPE ZBAPISTRVBAK-VBTYP
6   " EXPORTING
7   "   VALUE(RETURN) TYPE BAPIRET2
8   " TABLES
9   "   ITEMTAB STRUCTURE ZBAPISTRVBAK
10
11
12   SELECT vbttyp ernam SUM( netwr )
13   FROM vbak
14   INTO TABLE itemtab
15   WHERE vbttyp = vbttyp
16   GROUP BY ernam vbttyp.
17
18 ENDFUNCTION.
```

Save Cancel

158. Click Save.

159. Activate the Function Module.



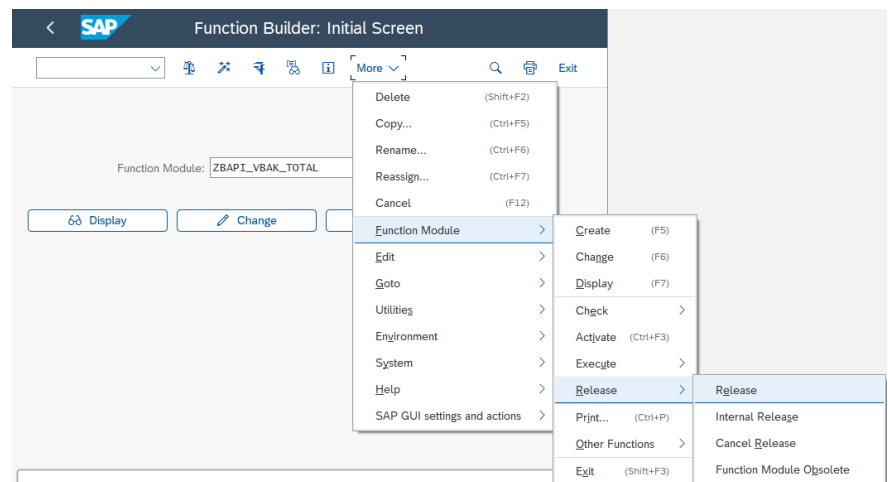
Function module: ZBAPI\_VBAK\_TOTAL **Active**

Attributes Import Export Changing Tables Exceptions **Source code**

```
1 FUNCTION ZBAPI_VBAK_TOTAL.
2
3   " "Local Interface:
4   " IMPORTING
5   "   VALUE(VBTYP) TYPE ZBAPISTRVBAK-VBTYP
6   " EXPORTING
7   "   VALUE(RETURN) TYPE BAPIRET2
8   " TABLES
9   "   ITEMTAB STRUCTURE ZBAPISTRVBAK
10
11
12   SELECT vbttyp ernam SUM( netwr )
13   FROM vbak
14   INTO TABLE itemtab
15   WHERE vbttyp = vbttyp
16   GROUP BY ernam vbttyp.
```

160. Go back.

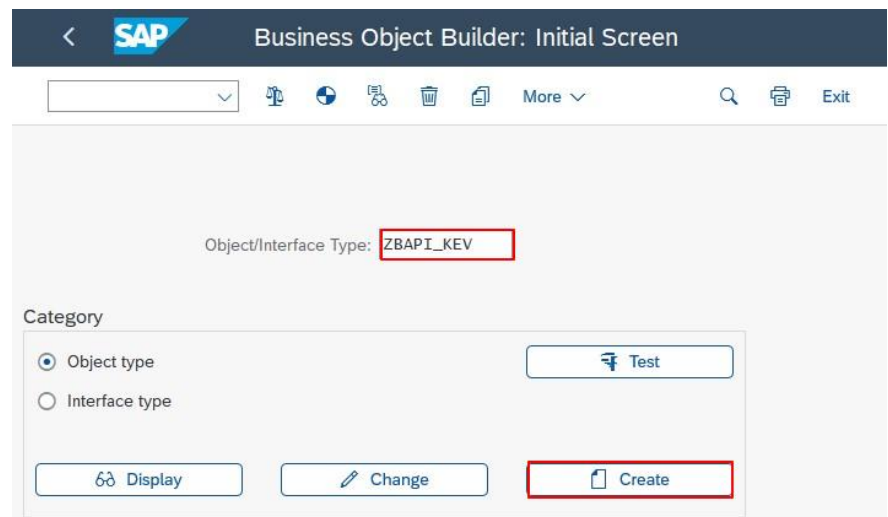
161. Release the function module by More -> Function Module -> Release -> Release.



162. Go to transaction SWO1.

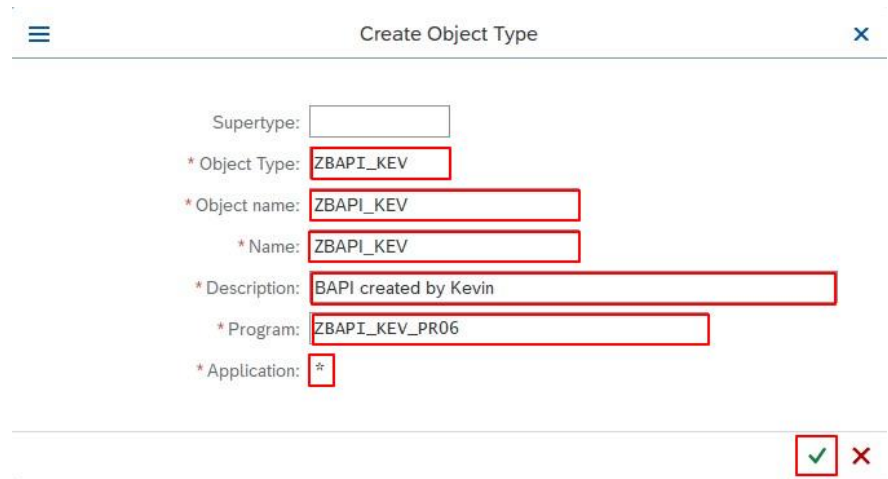
163. enter the name of the custom BAPI.

164. Click Create.



165. Enter the details as shown in the screenshot.

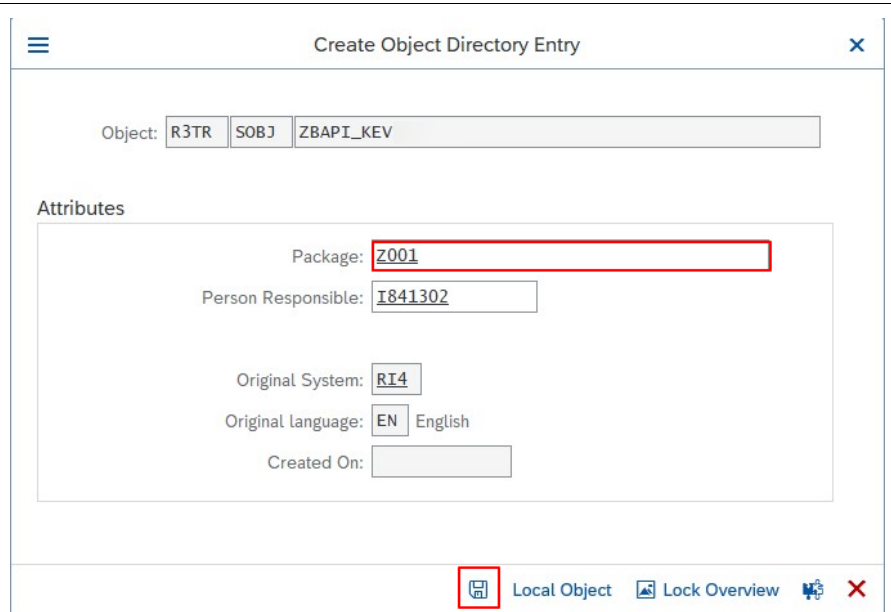
166. Click Save.



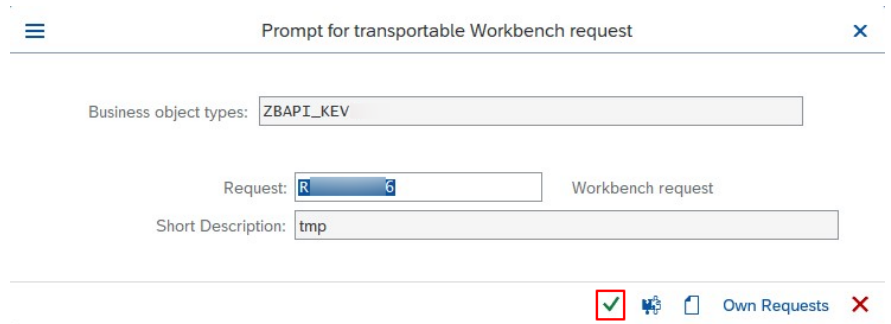


167. Enter a package Z001.

168. Click Save.

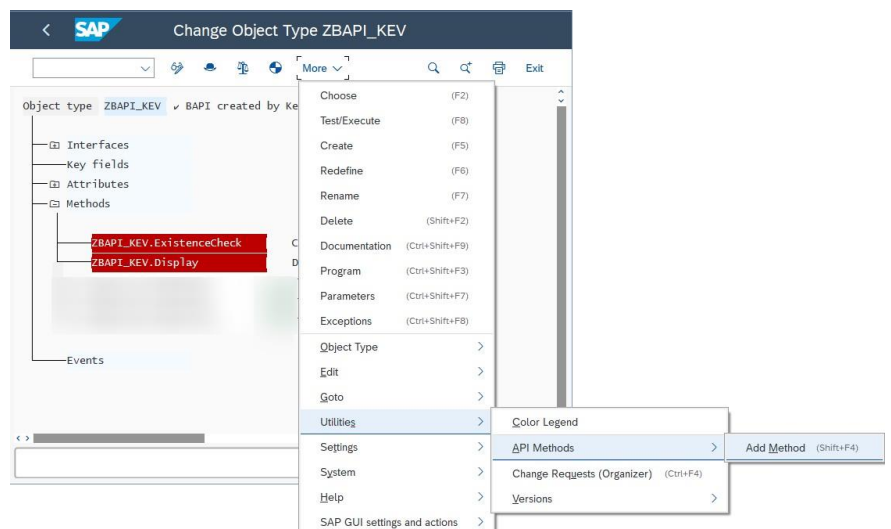


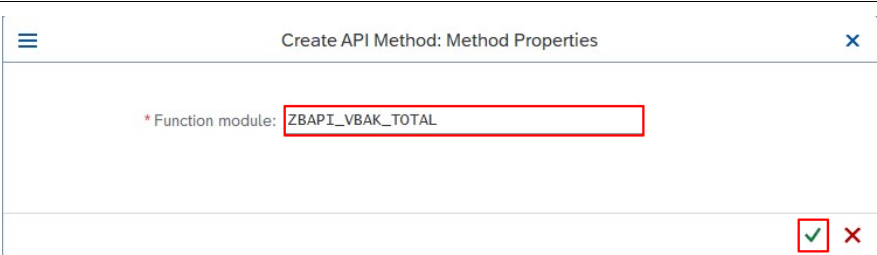
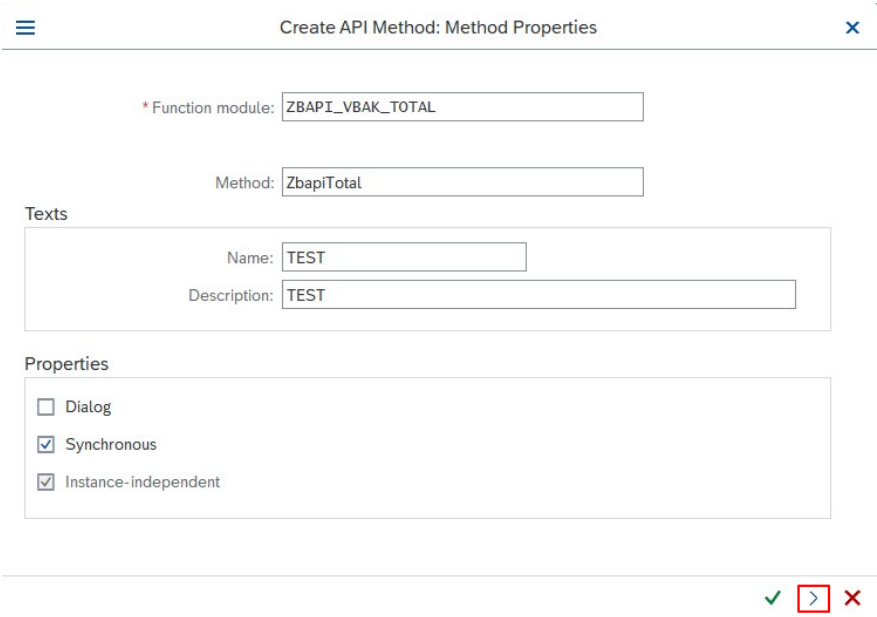
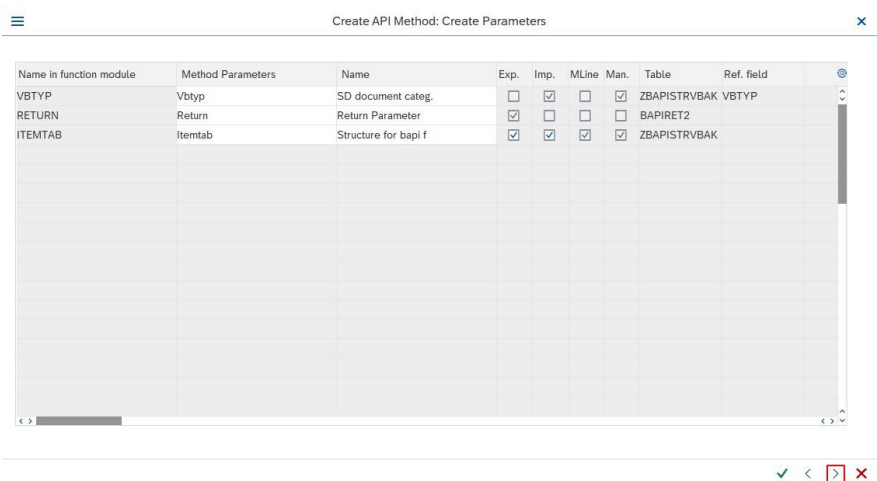
169. Click OK.



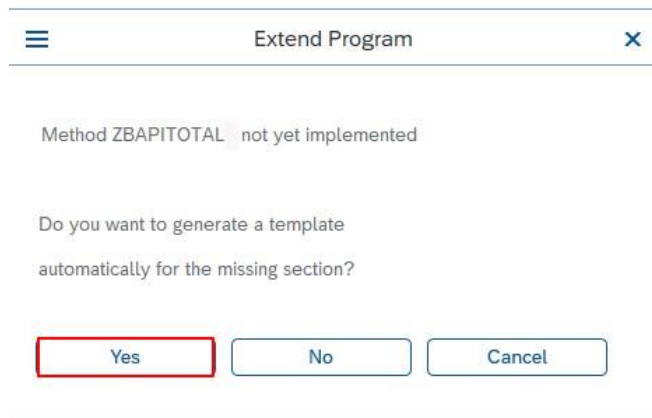
170. Select More -> Utilities -> API Methods -> Add Method.

**Note:** There would be two methods, showing in red color which comes by default while creating the BAPI.



171. Enter Function module name.	
172. Click OK.	
173. Click Next.	
174. Click Next.	

175. Click Yes.



Extend Program

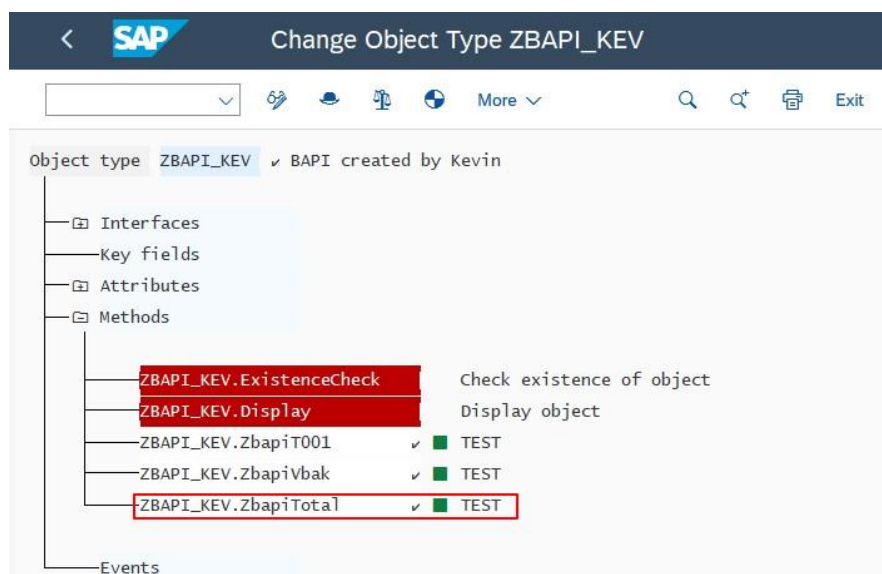
Method ZBAPITOTAL not yet implemented

Do you want to generate a template automatically for the missing section?

Yes No Cancel

176. Now the API method has been added.

177. Double click on the API method.



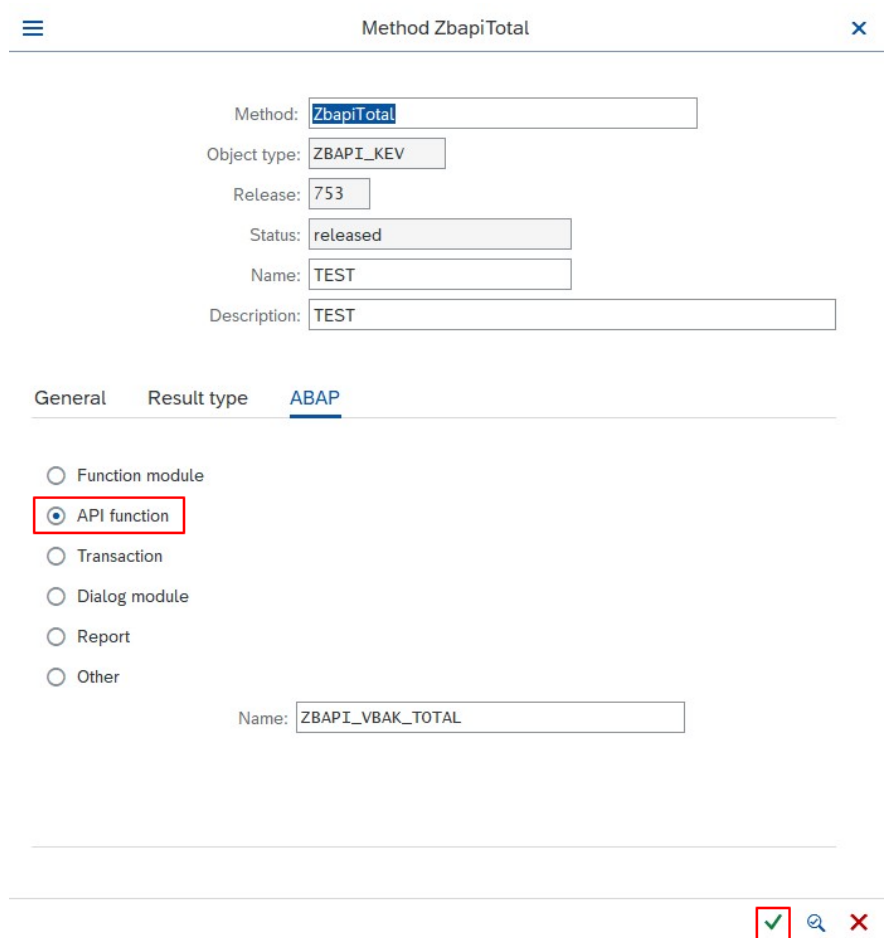
Change Object Type ZBAPI\_KEY

Object type ZBAPI\_KEY ✓ BAPI created by Kevin

- Interfaces
- Key fields
- Attributes
- Methods
  - ZBAPI\_KEY.ExistenceCheck Check existence of object
  - ZBAPI\_KEY.Display Display object
  - ZBAPI\_KEY.ZbapiT001 ✓ TEST
  - ZBAPI\_KEY.ZbapiVbak ✓ TEST
  - ZBAPI\_KEY.ZbapiTotal ✓ TEST
- Events

178. Select API Function.

179. Click Save.



Method ZbapiTotal

Method: ZbapiTotal

Object type: ZBAPI\_KEY

Release: 753

Status: released

Name: TEST

Description: TEST

General Result type **ABAP**

☐ Function module

☒ **API function**

☐ Transaction

☐ Dialog module

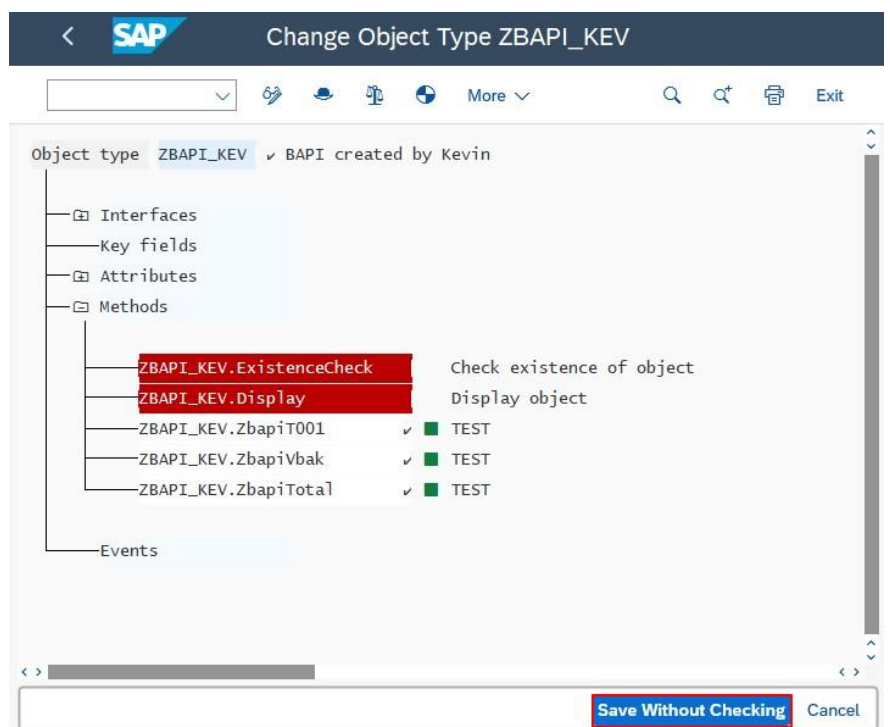
☐ Report


☐ Other

Name: ZBAPI\_VBAK\_TOTAL

✓ 🔍 ✕

180. Click Save.



<  Change Object Type ZBAPI\_KEY

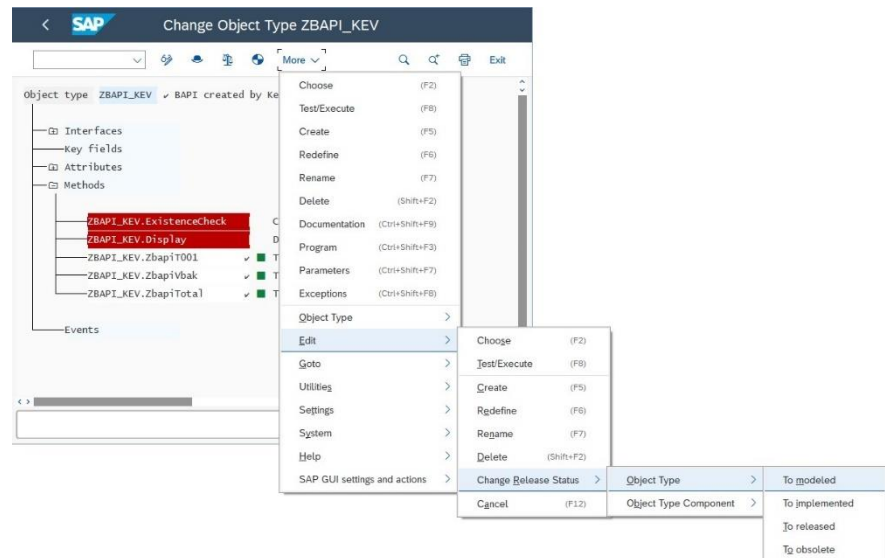
Object type ZBAPI\_KEY ✓ BAPI created by Kevin

- Interfaces
- Key fields
- Attributes
- Methods
  - ZBAPI\_KEY.ExistenceCheck Check existence of object
  - ZBAPI\_KEY.Display Display object
  - ZBAPI\_KEY.ZbapiT001 ✓ ■ TEST
  - ZBAPI\_KEY.ZbapiVbak ✓ ■ TEST
  - ZBAPI\_KEY.ZbapiTotal ✓ ■ TEST
- Events

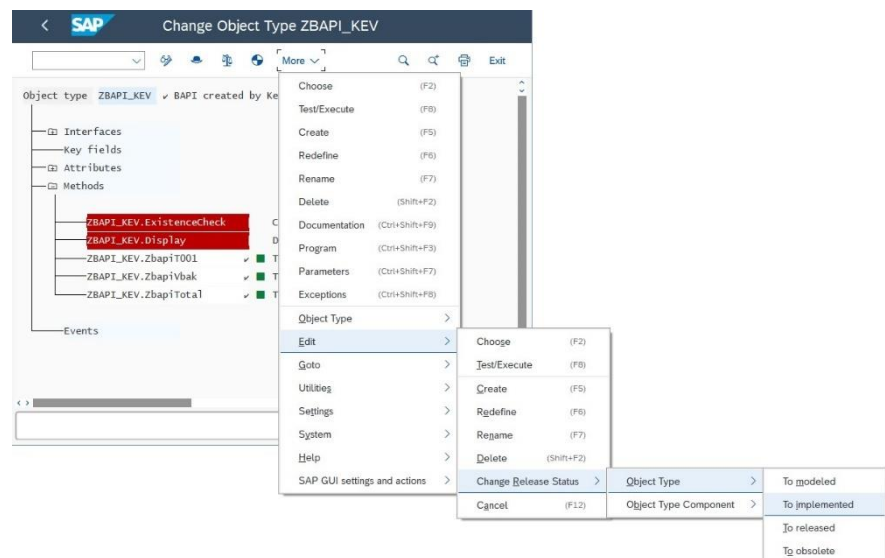
< > **Save Without Checking** Cancel

181. Select the Object  
"ZBAPI\_KEV".

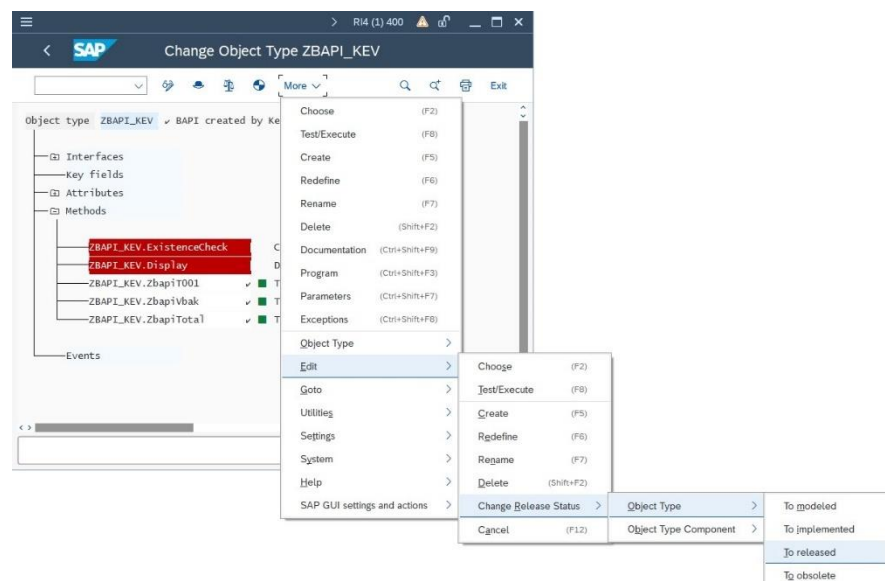
182. Go to More -> Edit ->  
Change Release Status ->  
Object type --> To modeled.



183. Go to More -> Edit ->  
Change Release Status ->  
Object type --> To implemented.

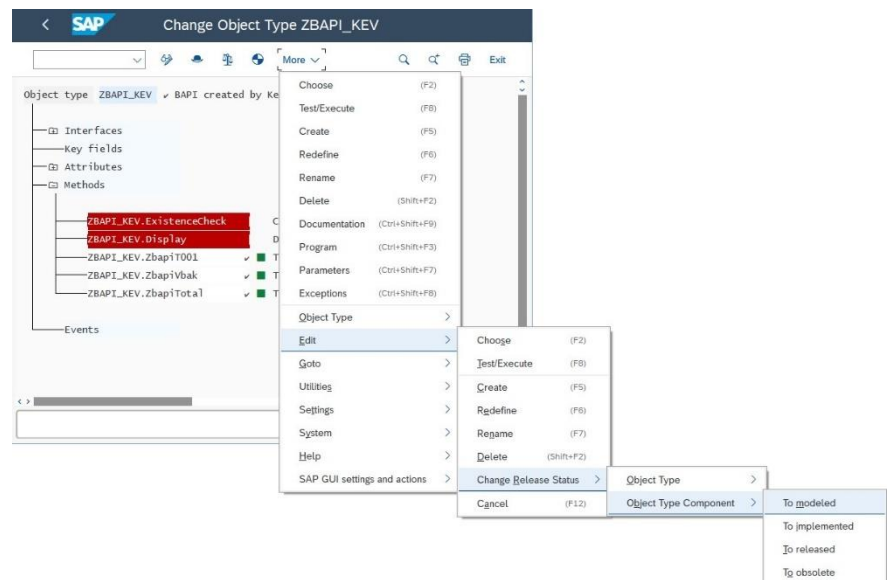


184. Go to More -> Edit ->  
Change Release Status ->  
Object type --> To released.

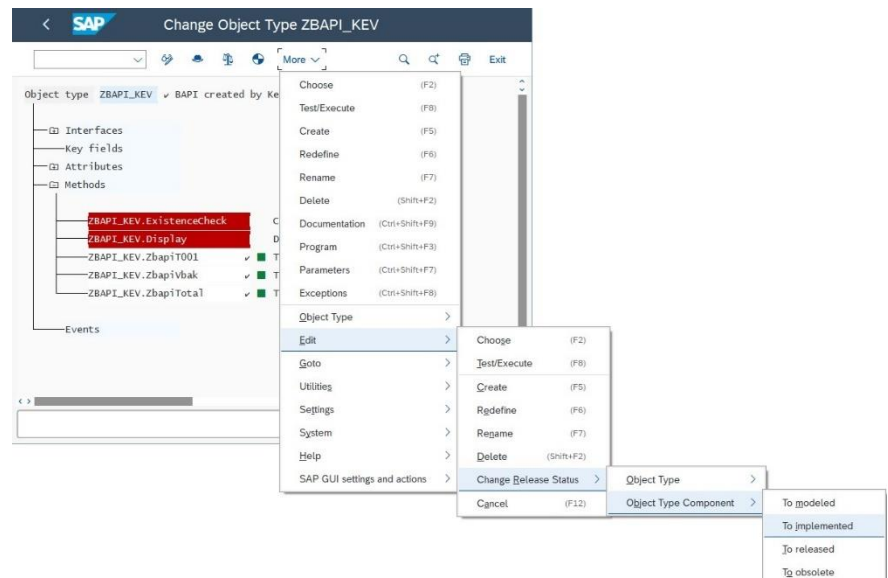


185. Select the API method.

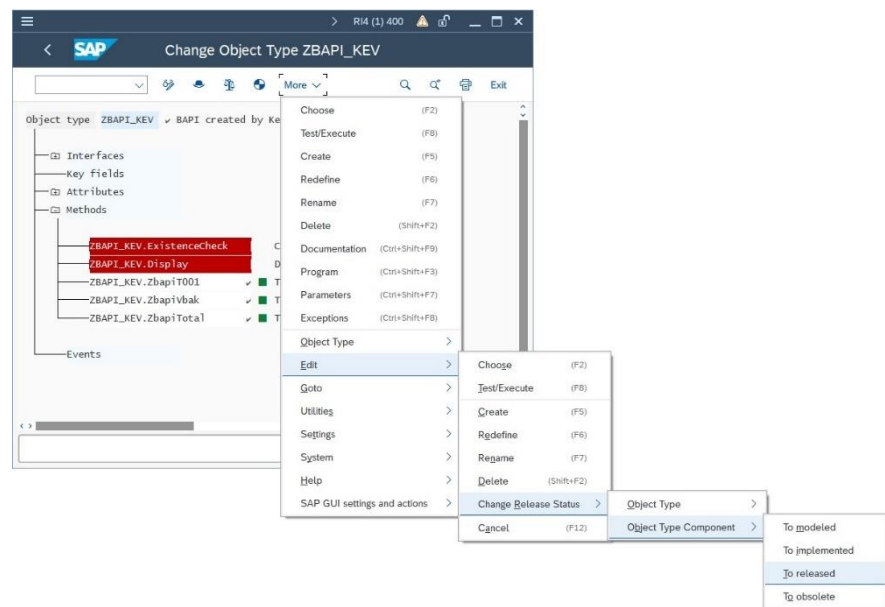
186. Go to More -> Edit -> Change Release Status -> Object type component -> To modeled.



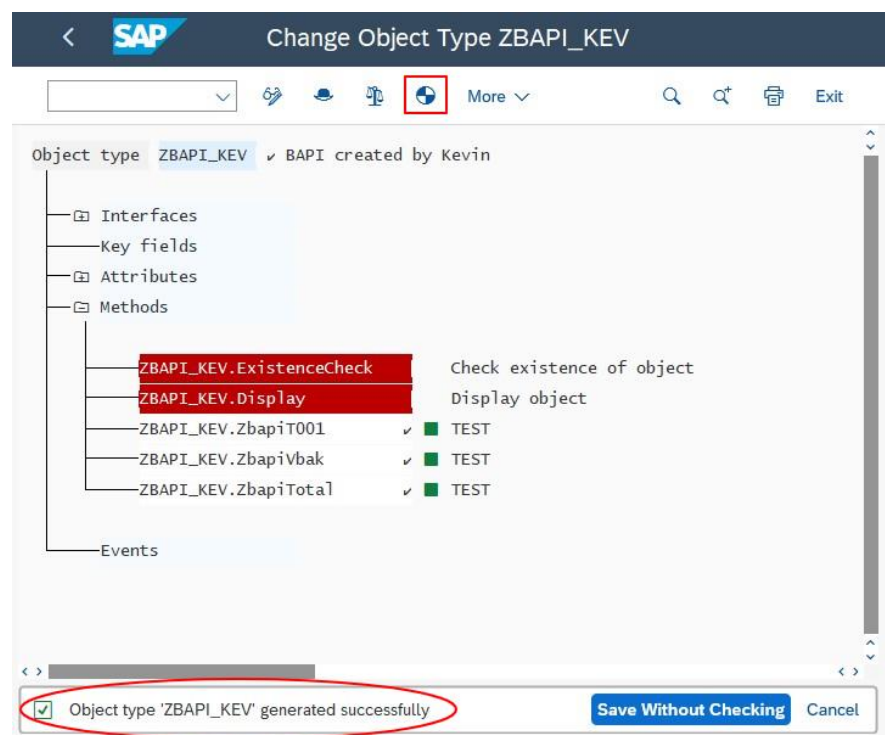
187. Go to More -> Edit -> Change Release Status -> Object type component -> To implemented.



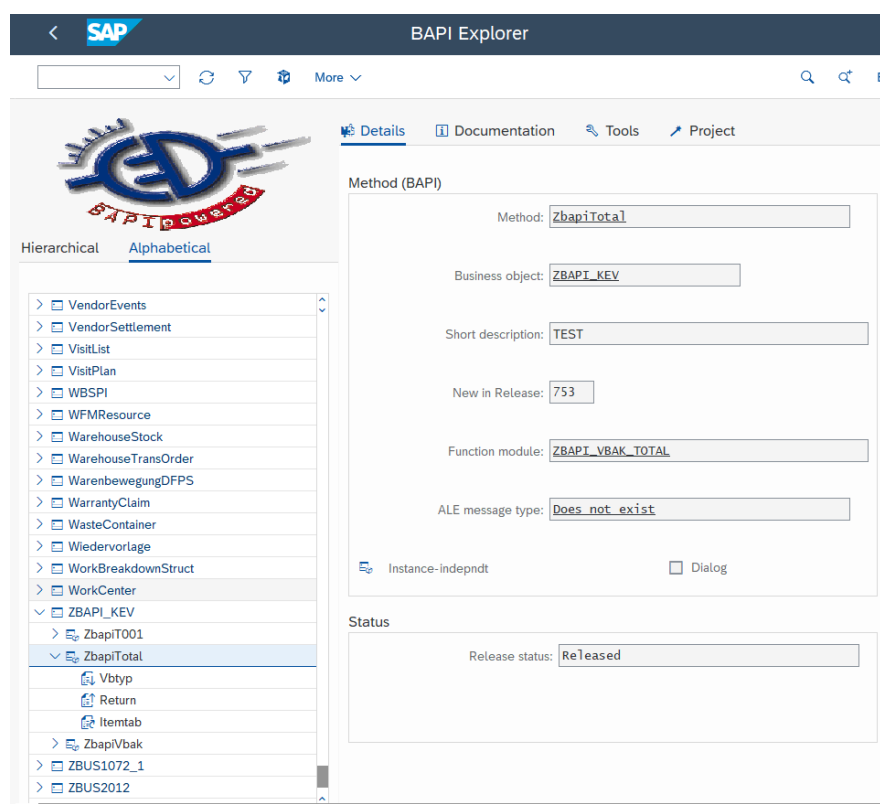
188. Go to More -> Edit -> Change Release Status -> Object type component -> To released.



189. Click Generate.



190. Go to transaction BAPI.
191. Click on Alphabetical.
192. Find your BAPI as shown in the screenshot.



## 4.4 Create Virtual Procedure

In HDI container, we can create a virtual procedure mapping to the custom BAPI in SAP ERP. Currently, the creation of virtual procedure still requires manual effort in SAP HANA Cloud. This [SAP Help](#) provides an example artifact code as following:

```
VIRTUAL PROCEDURE "com.sap.hana.example::REMOTE_PROCEDURE"
(
  IN INT_PARAM_AS_IN INT,
  OUT OUT_TABLE TABLE( INT_COLUMN INT, NVARCHAR_COLUMN NVARCHAR(2000))
)
CONFIGURATION '{}'
AT REMOTE_SOURCE
```

From the above code lines, there are 3 sections that needs to be replaced: parameters, configuration, and remote source. These information can be found in the function metadata by executing the following procedure:

```
call GET_REMOTE_SOURCE_FUNCTION_DEFINITION('REMOTE_SOURCE_NAME_HERE',
'BAPI.BAPI_NAME_HERE', ?, ?, ?)
```

In this documentation, the above SQL statement is executed as following:

```
call GET_REMOTE_SOURCE_FUNCTION_DEFINITION('S4_RI4', 'BAPI.ZBAPI_VBAK_TOTAL', ?, ?, ?)
```

In return, you can see 3 results and a message from the following screenshot. In Result 1, it show all the Parameters that need in this virtual procedure; In Result 3, it shows lines of code needs to be added as Configuration.



SQL Console 1.sql x

Analyze Current schema: DBA... Connected to: hanacloud (807ee9f4-54a9-4b94-9faf-be2540e7)

1 → call GET\_REMOTE\_SOURCE\_FUNCTION\_DEFINITION('S4\_RI4', 'ZBAPI\_ZBAPI\_VBAK\_TOTAL', ?, ?, ?)

Result 1 x Result 2 x Result 3 x Messages x History

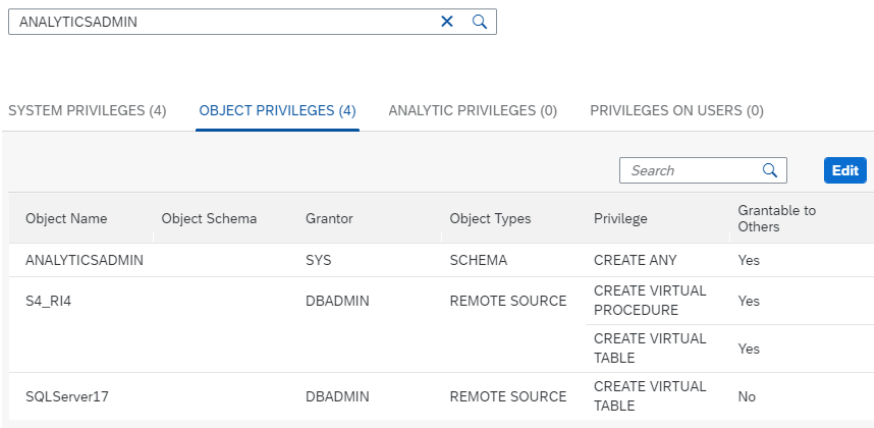

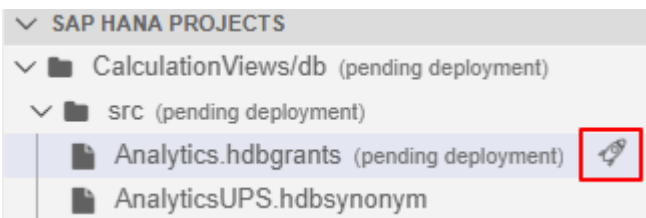
Rows (17)

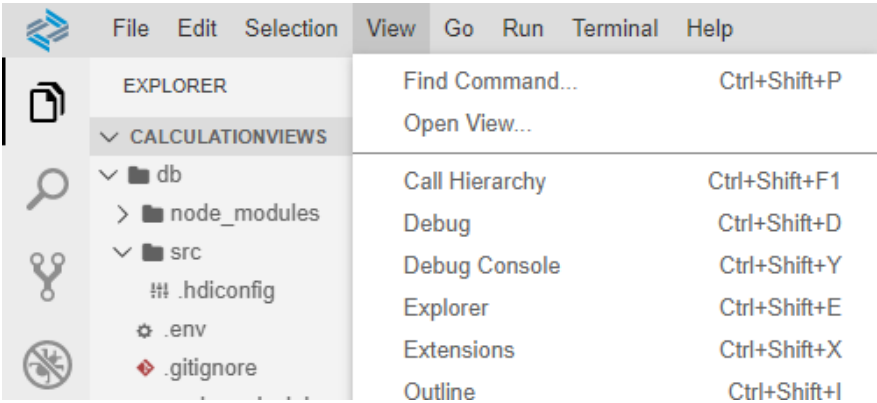
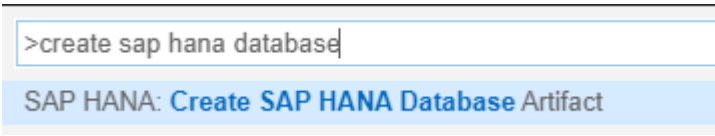
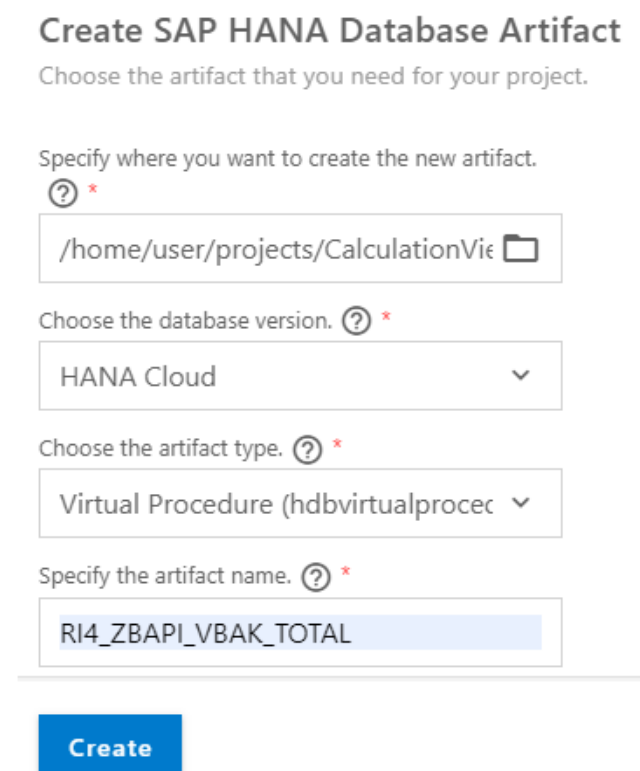
	PARAMETER_NAME	PARAMETER_TYPE	DATA_TYPE_NAME	LENGTH	SCALE	PARAMETER_POSITION
1	VBTYPE	IN	NVARCHAR	1	0	1
2	RETURN_TYPE	OUT	NVARCHAR	1	0	2
3	RETURN_ID	OUT	NVARCHAR	20	0	3
4	RETURN_NUMBER	OUT	NVARCHAR	3	0	4
5	RETURN_MESSAGE	OUT	NVARCHAR	220	0	5
6	RETURN_LOG_NO	OUT	NVARCHAR	20	0	6
7	RETURN_LOG_MSG_NO	OUT	NVARCHAR	6	0	7
8	RETURN_MESSAGE_V1	OUT	NVARCHAR	50	0	8
9	RETURN_MESSAGE_V2	OUT	NVARCHAR	50	0	9
10	RETURN_MESSAGE_V3	OUT	NVARCHAR	50	0	10
11	RETURN_MESSAGE_V4	OUT	NVARCHAR	50	0	11
12	RETURN_PARAMETER	OUT	NVARCHAR	32	0	12
13	RETURN_ROW	OUT	INTEGER	0	0	13

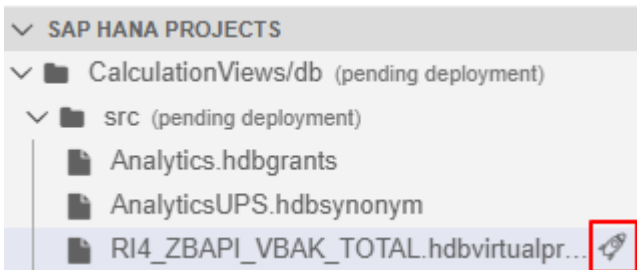
Therefore, the virtual procedure code can be constructed by the above information as followings:

```
VIRTUAL PROCEDURE "calculationview.db::RI4_ZBAPI_VBAK_TOTAL"(
  IN VBTYPE NVARCHAR(1),
  OUT TYPE NVARCHAR(1),
  OUT ID NVARCHAR(20),
  OUT NUMBER NVARCHAR(3),
  OUT MESSAGE NVARCHAR(220),
  OUT LOG_NO NVARCHAR(20),
  OUT LOG_MSG_NO NVARCHAR(6),
  OUT MESSAGE_V1 NVARCHAR(50),
  OUT MESSAGE_V2 NVARCHAR(50),
  OUT MESSAGE_V3 NVARCHAR(50),
  OUT MESSAGE_V4 NVARCHAR(50),
  OUT PARAMETER NVARCHAR(32),
  OUT ROW1 INTEGER,
  OUT FIELD NVARCHAR(30),
  OUT SYSTEM NVARCHAR(10),
  IN ITEM_TAB_IN TABLE (VBTYPE NVARCHAR(1), ERNAM NVARCHAR(12), NETWR DECIMAL(16, 2)),
  OUT ITEM_TAB_OUT TABLE (VBTYPE NVARCHAR(1), ERNAM NVARCHAR(12), NETWR DECIMAL(16, 2)))
CONFIGURATION '{
  "__DP_UNIQUE_NAME__": "ZBAPI_VBAK_TOTAL",
  "__DP_HAS_NESTED_PARAMETERS__": false,
  "__DP_USER_DEFINED_PROPERTIES__": {},
  "__DP_INPUT_PARAMETER_PROPERTIES__": [],
  "__DP_RETURN_PARAMETER_PROPERTIES__": [],
  "__DP_VIRTUAL_PROCEDURE__": true,
  "__DP_HAS_INTERNAL_OUTPUT_PARAMETER__": false,
  "__DP_DEFAULT_OUTPUT_PARAMETER_INDEX__": 0
}'
AT "S4_RI4"
```

We will use the above code to create a virtual procedure in SAP Business Application Studio.

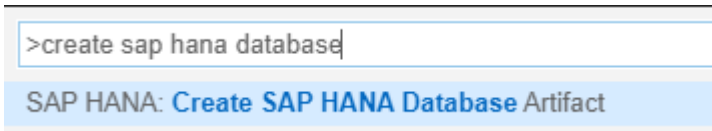
Explanation	Screenshot																														
<p>193. Go to <b>SAP HANA Cockpit</b> -&gt; <b>Privilege Assignment</b></p> <p>194. Add the privilege <b>CREATE VIRTUAL PROCEDURE</b> and <b>CREATE VIRTUAL TBALE ON</b> remote source <b>S4_RI4</b> for <b>ANALYTICSADMIN</b></p>	 <table><thead><tr><th>Object Name</th><th>Object Schema</th><th>Grantor</th><th>Object Types</th><th>Privilege</th><th>Grantable to Others</th></tr></thead><tbody><tr><td>ANALYTICSADMIN</td><td></td><td>SYS</td><td>SCHEMA</td><td>CREATE ANY</td><td>Yes</td></tr><tr><td>S4_RI4</td><td></td><td>DBADMIN</td><td>REMOTE SOURCE</td><td>CREATE VIRTUAL PROCEDURE</td><td>Yes</td></tr><tr><td></td><td></td><td></td><td></td><td>CREATE VIRTUAL TABLE</td><td>Yes</td></tr><tr><td>SQLServer17</td><td></td><td>DBADMIN</td><td>REMOTE SOURCE</td><td>CREATE VIRTUAL TABLE</td><td>No</td></tr></tbody></table>	Object Name	Object Schema	Grantor	Object Types	Privilege	Grantable to Others	ANALYTICSADMIN		SYS	SCHEMA	CREATE ANY	Yes	S4_RI4		DBADMIN	REMOTE SOURCE	CREATE VIRTUAL PROCEDURE	Yes					CREATE VIRTUAL TABLE	Yes	SQLServer17		DBADMIN	REMOTE SOURCE	CREATE VIRTUAL TABLE	No
Object Name	Object Schema	Grantor	Object Types	Privilege	Grantable to Others																										
ANALYTICSADMIN		SYS	SCHEMA	CREATE ANY	Yes																										
S4_RI4		DBADMIN	REMOTE SOURCE	CREATE VIRTUAL PROCEDURE	Yes																										
				CREATE VIRTUAL TABLE	Yes																										
SQLServer17		DBADMIN	REMOTE SOURCE	CREATE VIRTUAL TABLE	No																										
<p>195. Navigate to <b>SAP Business Application Studio</b></p> <p>196. Open <b>Analytics.hdbgrants</b></p> <p>197. Grant the container's object owner the privilege <b>CREATE VIRTUAL PROCEDURE ON REMOTE SOURCE</b> on the remote source.</p> <p><b>Note:</b> Don't forget to include the comma at the beginning!</p>	 <pre>1  { 2    "AnalyticsUPS": { 3      "object_owner": { 4        "schema_privileges": [ 5          { 6            "reference": "ANALYTICSADMIN", 7            "privileges_with_grant_option": ["SELECT", "EXECUTE"] 8          }, 9        ], 10     "global_object_privileges": [ 11       { 12         "name": "S4_RI4", 13         "type": "REMOTE SOURCE", 14         "privileges": [ "CREATE VIRTUAL TABLE" ], 15         "privileges_with_grant_option": [ "CREATE VIRTUAL PROCEDURE" ] 16       } 17     ], 18   }, 19   "application_user": { 20     "schema_privileges": [ 21       { 22         "reference": "ANALYTICSADMIN", 23         "privileges_with_grant_option": ["SELECT"] 24       } 25     ] 26   } 27 } 28 </pre>																														
<p>198. Deploy the new <b>Abalytics.hdbgrants</b></p>																															





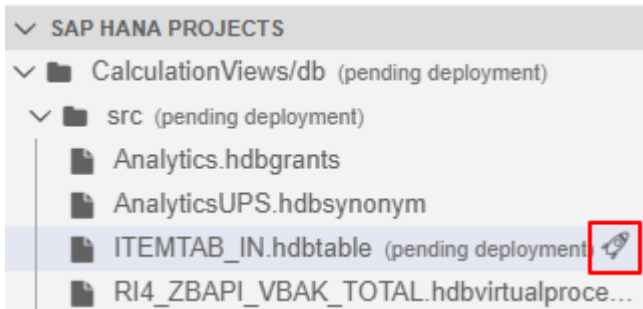
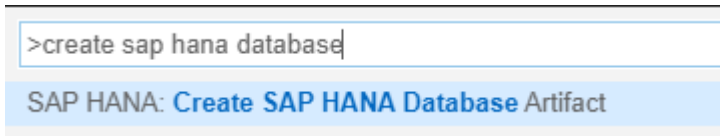
<p>199. Go to <b>View</b> -&gt; <b>Find Command...</b></p>	
<p>200. Search for <b>Create SAP HANA Database Artifact</b></p> <p>201. Click on it</p>	
<p>202. Select <b>Virtual Procedure</b> as Artifact Type</p> <p>203. Enter <b>Artifact name</b></p> <p>204. Click <b>Create</b></p>	 <p><b>Create SAP HANA Database Artifact</b></p> <p>Choose the artifact that you need for your project.</p> <p>Specify where you want to create the new artifact. <span>?</span> *</p> <p><input type="text" value="/home/user/projects/CalculationView"/></p> <p>Choose the database version. <span>?</span> *</p> <p><input type="text" value="HANA Cloud"/></p> <p>Choose the artifact type. <span>?</span> *</p> <p><input type="text" value="Virtual Procedure (hdbvirtualprocedure)"/></p> <p>Specify the artifact name. <span>?</span> *</p> <p><input type="text" value="RI4_ZBAPI_VBAK_TOTAL"/></p> <p><b>Create</b></p>
<p>205. Open the created <b>Virtual Procedure</b> and copy the code on the right side into the code editor</p>	<pre> VIRTUAL PROCEDURE "calculationview.db::RI4_ZBAPI_VBAK_TOTAL"(   IN VB_TYP NVARCHAR(1),   OUT TYPE NVARCHAR(1),   OUT ID NVARCHAR(20),   OUT NUMBER NVARCHAR(3),   OUT MESSAGE NVARCHAR(220),   OUT LOG_NO NVARCHAR(20),   OUT LOG_MSG_NO NVARCHAR(6),   OUT MESSAGE_V1 NVARCHAR(50),   OUT MESSAGE_V2 NVARCHAR(50),   OUT MESSAGE_V3 NVARCHAR(50), </pre>

	<pre> OUT MESSAGE_V4 NVARCHAR(50), OUT PARAMETER NVARCHAR(32), OUT ROW1 INTEGER, OUT FIELD NVARCHAR(30), OUT SYSTEM NVARCHAR(10), IN ITEMTAB_IN TABLE (VBTYPE NVARCHAR(1),ERNAM NVARCHAR(12),NETW R DECIMAL(16, 2)), OUT ITEMTAB_OUT TABLE (VBTYPE NVARCHAR(1),ERNAM NVARCHAR(12),NE TWR DECIMAL(16, 2))) CONFIGURATION '{   "__DP_UNIQUE_NAME__": "ZBAPI_VBAK_TOTAL",   "__DP_HAS_NESTED_PARAMETERS__": false,   "__DP_USER_DEFINED_PROPERTIES__": {},   "__DP_INPUT_PARAMETER_PROPERTIES__": [],   "__DP_RETURN_PARAMETER_PROPERTIES__": [],   "__DP_VIRTUAL_PROCEDURE__": true,   "__DP_HAS_INTERNAL_OUTPUT_PARAMETER__": false,   "__DP_DEFAULT_OUTPUT_PARAMETER_INDEX__": 0 }' AT "S4_RI4" </pre>
206. Deploy the virtual procedure	

## 4.5 Create HANA Function

In SAP HANA Cloud, we currently have to use HANA Function to make the use of the virtual procedures. Thus, the next step is to create a HANA function to call the virtual procedure in HDI container. In the definition of the virtual procedure, a HANA table is required as an input parameter, so we will start from creating a HANA table in HDI container.

Explanation	Screenshot
207. Search for <b>Create SAP HANA Database Artifact</b> 208. Click on it	

<p>209. Select <b>Table</b> as Artifact Type</p> <p>210. Enter <b>Artifact name</b></p> <p>211. Click <b>Create</b></p>	<h3>Create SAP HANA Database Artifact</h3> <p>Choose the artifact that you need for your project.</p> <p>Specify where you want to create the new artifact.  *</p> <input type="text" value="/home/user/projects/CalculationView"/> <p>Choose the database version.  *</p> <input type="text" value="HANA Cloud"/> <p>Choose the artifact type.  *</p> <input type="text" value="Table (hdbtable)"/> <p>Specify the artifact name.  *</p> <input type="text" value="ITEMTAB_IN"/> <p><b>Create</b></p>
<p>212. Open the created <b>Table</b> and copy the code on the right side into the code editor</p>	<pre>COLUMN TABLE "calculationview.db::ITEMTAB_IN" (   "VBTYPE" NVARCHAR(1),   "ERNAM" NVARCHAR(12),   "NETWR" DECIMAL(16,2) )</pre>
<p>213. Deploy the ITEMTABL_IN.hdbtable</p>	
<p>214. Search for <b>Create SAP HANA Database Artifact</b></p> <p>215. Click on it</p>	

216. Select **Function** as Artifact Type

217. Enter **Artifact name**

218. Click **Create**


## Create SAP HANA Database Artifact

Choose the artifact that you need for your project.

Specify where you want to create the new artifact.



/home/user/projects/CalculationView

Choose the database version.  \*

HANA Cloud

Choose the artifact type.  \*

Function (hdbfunction)

Specify the artifact name.  \*

ZBAPI\_VBAK\_TOTAL

Create

219. Open the created **Function** and copy the code on the right side into the code editor

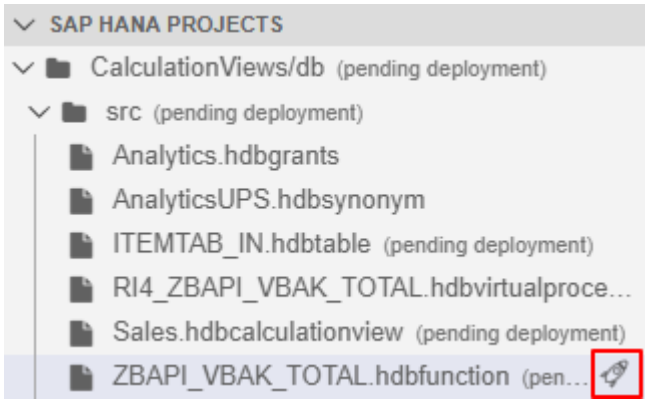
```
FUNCTION "ZBAPI_VBAK_TOTAL" ( )
  RETURNS TABLE (
    "VB Typ" NVARCHAR(1),
    "ERNAM" NVARCHAR(12),
    "NETWR" DECIMAL(16,2))
  LANGUAGE SQLSCRIPT
  SQL SECURITY INVOKER AS
BEGIN
  /*****
   Write your function logic
   *****/
  DECLARE i1 NVARCHAR(1) := 'C';

  DECLARE o1 NVARCHAR (1);
  DECLARE o2 NVARCHAR (20);
  DECLARE o3 NVARCHAR (3);
  DECLARE o4 NVARCHAR (220);
  DECLARE o5 NVARCHAR (20);
  DECLARE o6 NVARCHAR (6);
  DECLARE o7 NVARCHAR (50);
  DECLARE o8 NVARCHAR (50);
  DECLARE o9 NVARCHAR (50);
  DECLARE o10 NVARCHAR (50);
  DECLARE o11 NVARCHAR (32);
  DECLARE o12 INTEGER;
  DECLARE o13 NVARCHAR (30);
  DECLARE o14 NVARCHAR (10);

  ITEM_TAB_IN = select * from "calculationview.db::ITEMTAB_IN";

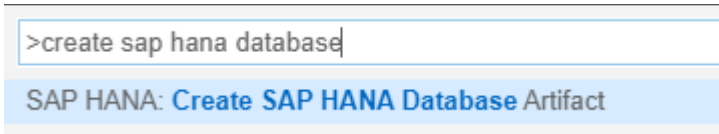
  call "calculationview.db::RI4_ZBAPI_VBAK_TOTAL"(i1,o1, o2, o3,
  o4, o5, o6, o7, o8, o9,o10,o11,o12,o13,o14, :ITEMTAB_IN,ITEMTAB_OUT);

  RETURN
  SELECT * FROM :ITEMTAB_OUT;
```

	END;
220. Deploy the ZBAPI_VBAK_TOTAL.hdbfunction	

## 4.6 Create Calculation View

Finally, we can create a calculation view using the above function in HDI container.

Explanation	Screenshot
221. Search for <b>Create SAP HANA Database Artifact</b> 222. Click on it	

223. Select **Calculation View** as  
Artifact Type

224. Enter **Artifact name**

225. Click **Create**

## Create SAP HANA Database Artifact

Choose the artifact that you need for your project.

Specify where you want to create the new artifact.

Choose the database version.

Choose the artifact type.

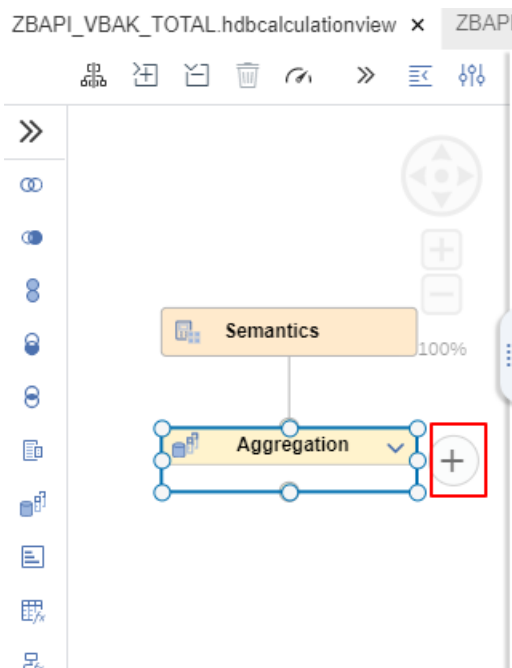
Specify the artifact name.

Specify the label.

**Create**

226. Click **Aggregation**

227. Click **Add Data Source**





228. Search Function  
**ZBAPI\_VBAK\_TOTAL**  
 229. Select the Function  
 230. Click **Finish**


Find Data Sources

Search for an object name

Services: 1 Services Selected

All Types Selected

Results (1)

Type	Name	Schema	Synonym
	calculationview.db::ZBAPI_VB...	CALCULATIONVIE...	

Create Synonym **Finish** Cancel

231. Create **Mapping** as shown  
 in the right screenshot

Aggregation

Mapping Calculated Columns (0) Restricted Columns (0) Parameters (0)

Data Sources

Output Columns

VBTYPE

NETWR

ERNAM

Drag & drop here

232. Deploy  
**ZBAPI\_VBAK\_TOTAL.hdbcal**  
**culatationview**

SAP HANA PROJECTS

CalculationViews/db (pending deployment)

src (pending deployment)


Analytics.hdbgrants

AnalyticsUPS.hdbsynonym

ITEMTAB\_IN.hdbtable (pending deployment)

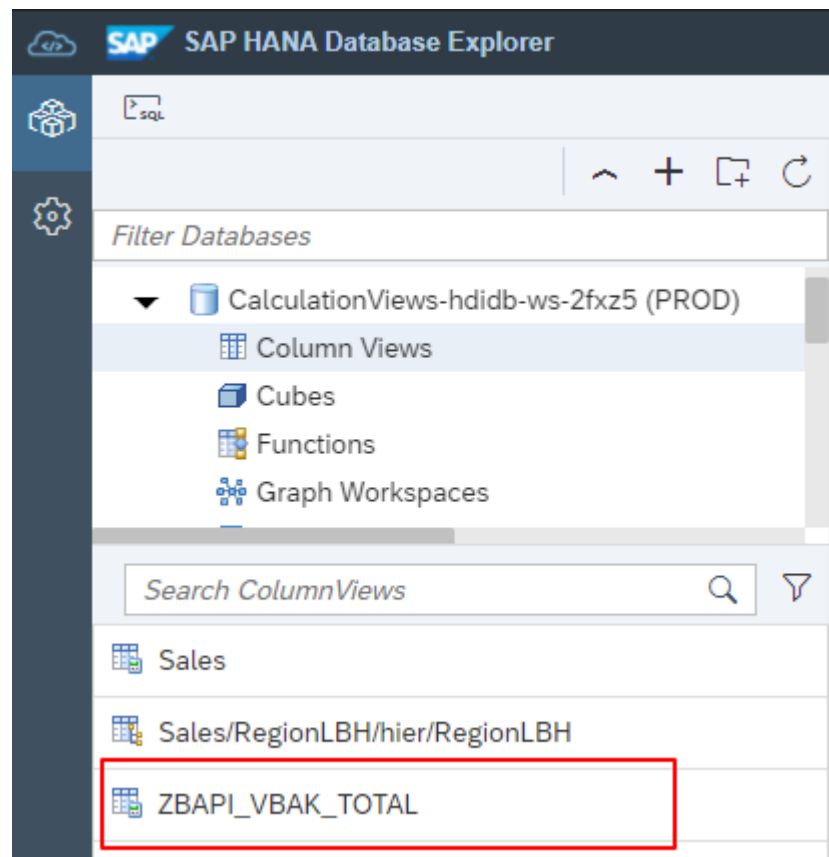
RI4\_ZBAPI\_VBAK\_TOTAL.hdbvirtualproce...

Sales.hdbcalculationview (pending deployment)

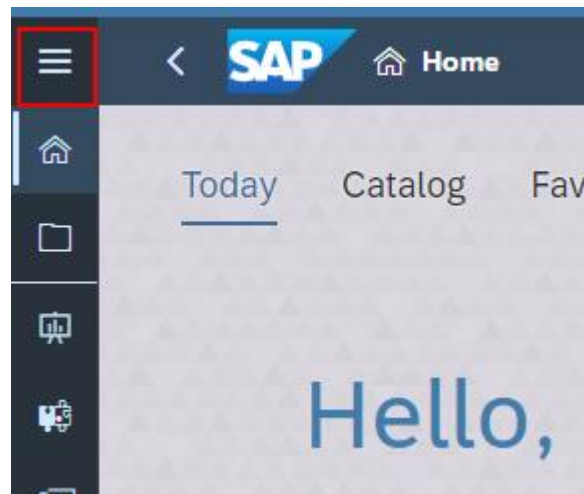
**ZBAPI\_VBAK\_TOTAL.hdbcalculationvi...** 

ZBAPI\_VBAK\_TOTAL.hdbfunction (pending ...)

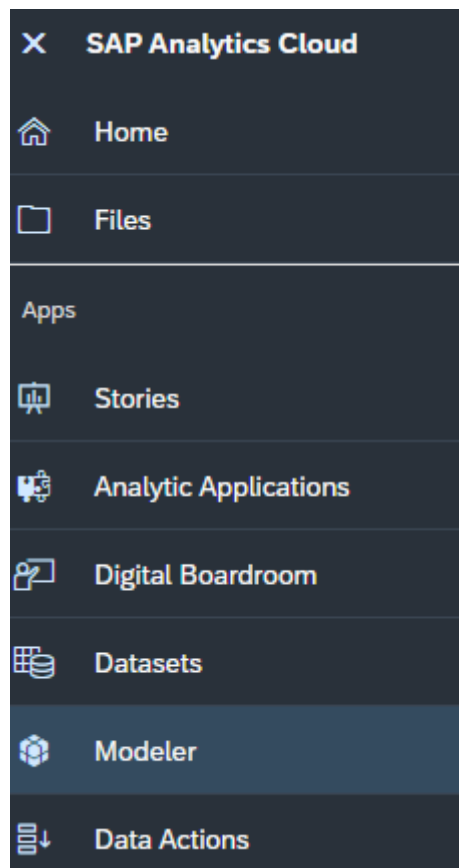
233. Congratulations! You have created a calculation view in SAP HANA Cloud



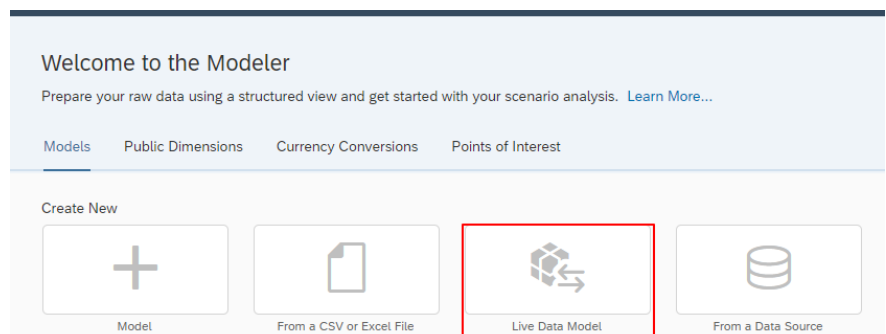
## 4.7 Create a story in SAP Analytics Cloud

Explanation	Screenshot
<p>234. Logon to your SAP Analytics Cloud tenant.</p> <p>235. Click on the Main Menu.</p>	

236. Select Modeler



237. Click on Live Data Model



238. Enter the following information  
 System Type: SAP HANA  
 Connection: HANACloud  
 Data Source:  
 ZBAPI\_VBAK\_TOTAL

239. Click OK.

Create Model From Live Data Connection

Select Live Data Connection and Data Source

\*System Type:  
 SAP HANA

\*Connection:  
 HANACloud

\*Data Source:  
 ZBAPI\_VBAK\_TOTAL

OK Cancel

240. Save the model.

70103 Files / New Model

Data sources

Save icon (highlighted in red)

ID	Description	Aggregation Ty...
1	NETWR	SUM
2		

241. Provide the new Model a name and save it in a directory.

Save Model

My Files / Public / t... / ABAP

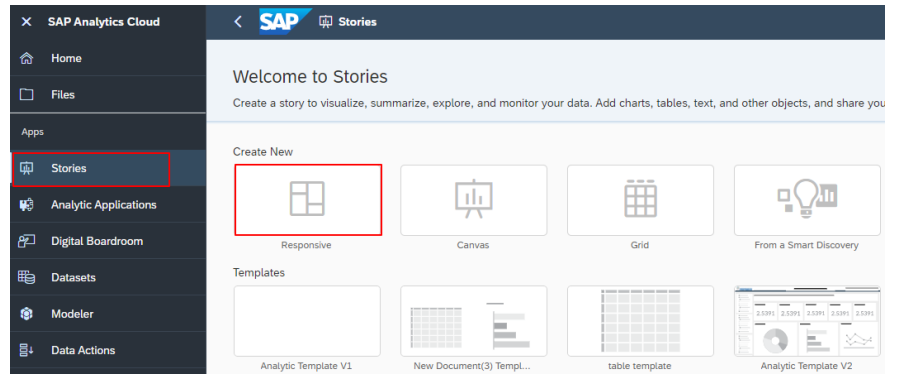
Name	Description	Owner
ABAPTABLES	-	Kevin Li
BAPI_COMPANY_LIST	-	Kevin Li
BAPI_SALESORDER_GET...	-	Kevin Li
ZBAPI_VBAK	-	Kevin Li
ZBAPI_VBAK_TOTAL	-	Kevin Li

\*Name  
 ZBAPI\_VBAK\_TOTAL

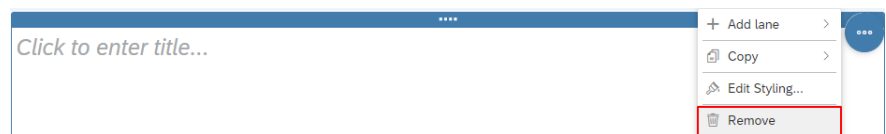
Description  
 Optional

OK Cancel

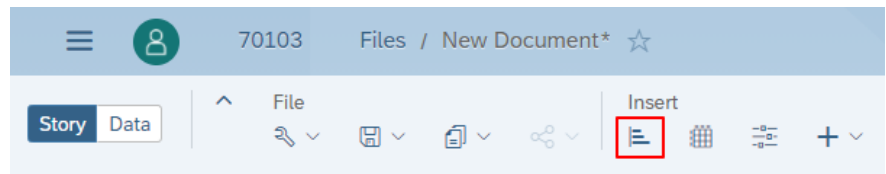
242. Select **Stories**  
243. Select **Responsive**



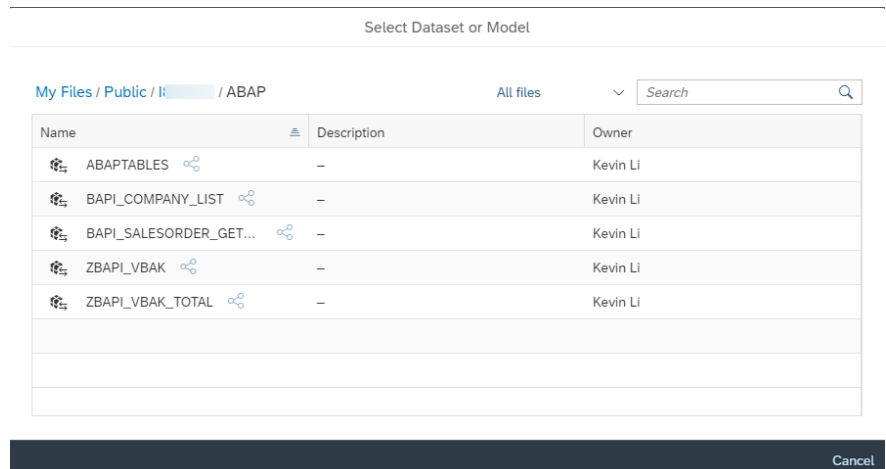
244. Remove one of the responsive pages.



245. Insert a Chart.



246. Select the model  
ZBAPI\_VBAK\_TOTAL.




247. Select NETWR as Measures.


248. Select ERNAM as Dimensions.


Data Source  
**ZBAPI\_VBAK\_TOTAL**


+ Add Linked Models


▼ Chart Structure +

Comparison  


Trend  


Distribution  


Correlation  


Indicator  



More  


Chart Orientation  
Vertical

Measures  
NETWR

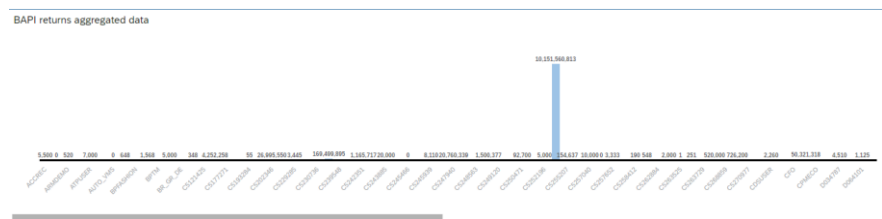
+ Add Measure

Dimensions  
ERNAM

+ Add Dimension

249. The chart look should like the screenshot.

250. Save the story.



## 5. Current Limitations

This chapter covers the current limitations of SAP Analytics Cloud using SAP HANA Cloud as a single gateway to heterogeneous remote sources.

### 5.1 Building Calculation Views in SAP HANA Cloud

Compare to the existing live data connectivity in SAC, HANA data modeling is an extra effort that must be invested by modelers. In order to have live analytics on the remote source, you must know how to build calculation views in SAP HANA Cloud using the virtual tables or virtual procedures.

### 5.2 Geospatial is not Supported

SAP HANA Spatial allows you to store and geo-process spatial data using various data types, algorithms, and constructors. However, Geospatial is not currently supported for Live Data Connections to SAP HANA HDI containers on SAPCP on Cloud Foundry (CF) systems.

### 5.3 Creating Custom BAPIs in SAP NetWeaver AS ABAP

Custom BAPIs need to be developed in most use cases. The performance best practice requires BAPIs to return aggregated results for optimal performance, however, very few existing BAPIs return aggregated results. Creating custom BAPIs requires some ABAP programming knowledge and effort. Whereas in general, it is not difficult to create custom BAPIs returning the aggregated results.

### 5.4 Known Limitations for Neo Environment

For SAP Analytics Cloud tenant provisioned in Neo environment, The following SAP Analytics Cloud features are not supported with [Live Data Connection to HANA Cloud using a Direct Connection](#) at this time:

- Enriched time dimensions
- Timestamp dimensions
- Dynamic image widgets
- Smart Insights
- Smart Assist/Changes Over Time in Smart Insights
- Smart Predict
- Story scheduling
- Android/iOS Mobile App:
- SAP Cloud Connector-based single sign-on
- Basic Authentication connections
- Geo maps
- OEM scenarios through “Extended Live Connections”



<https://www.sap.com/registration/contact.html>

<https://saphanajourney.com/hana-cloud/>

<https://saphanajourney.com/sap-analytics-cloud/>

[https://help.sap.com/viewer/product/HANA\\_CLOUD](https://help.sap.com/viewer/product/HANA_CLOUD)

[https://help.sap.com/viewer/product/SAP\\_ANALYTICS\\_CLOUD](https://help.sap.com/viewer/product/SAP_ANALYTICS_CLOUD)

© 2021 SAP SE or an SAP affiliate company. 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 SE or an SAP affiliate company. The information contained herein may be changed without prior notice.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary. These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies.

Please see <http://www.sap.com/corporate-en/legal/copyright/index.epx> for additional trademark information and notices.



The Best-Run Businesses Run SAP™