Technical Articles

Abhishek Patel

August 26, 2020 | 3 minute read

How to consume HANA Calculation views in S/4HANA CDS views

Follow

👍 Like

🔊 RSS Feed

In this blog you will learn how to consume the HANA Calculation Views inside a S/4HANA CDS views.

Consider a scenario where business already have report based Enterprise HANA calculation view and now they want to consume same view in S/4 HANA CDS view with additional functionalities and business logic. Below are the steps which we need to follow to achieve above use case.
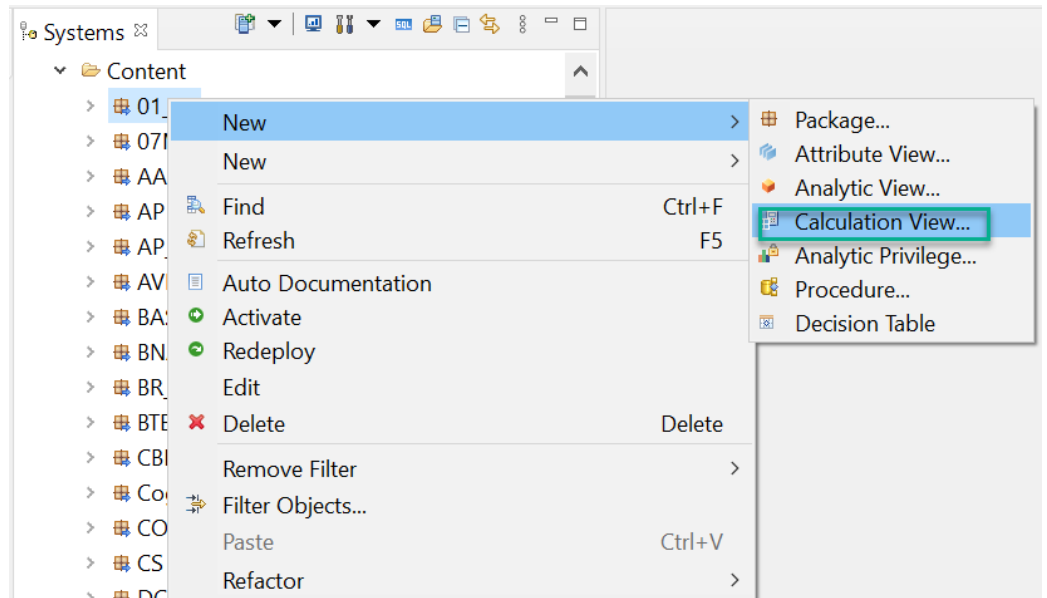
Steps to be followed:

1. Create Calculation view in E-HANA.
2. Test view output in HANA using native SQL.
3. Create CDS table function.
4. Create AMDP function for calling HANA native query on calculation view.
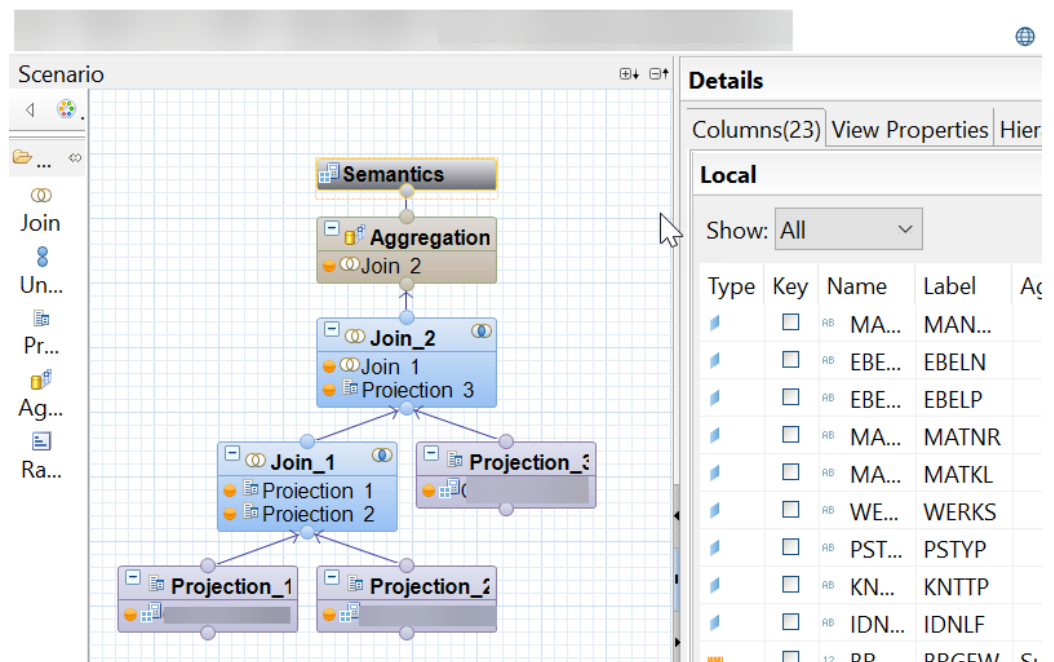5. Expose result to outer CDS with business logic.

So, let's start building:

Step 1: Create a Calculation View

1. Open Eclipse/ HANA Studio

2. Select system and expand content folder and choose package where calculation view needs to be created.

3. Right Click and Create New Calculation View.

4. Add required tables/ views and select columns as per logic. In the View Properties make Default Client = Cross Client and Apply privileges as blank if not required.

5. Now save and activate the view.

Step 2: Test view output in HANA using native SQL.
Open a SQL console by selecting a system and clicking on open a SQL console for selected system and write a query to fetch data from the view and hit a execute button to see the view's data.
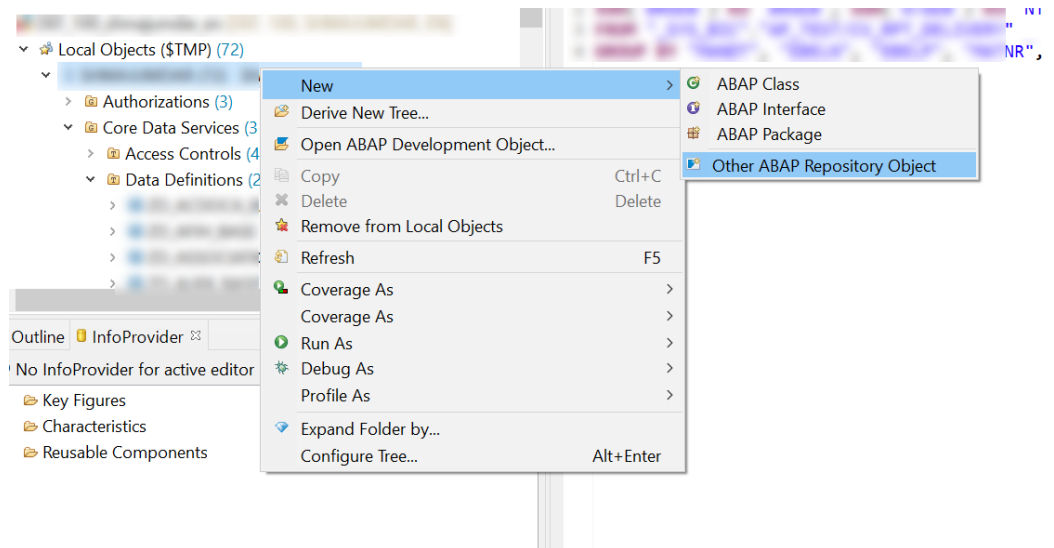
```
SQL  Result
SELECT "MANDT", "EBELN", "EBELP", "MATNR", "MATKL", "WERKS", "PSTYP", "KNTTP", "IDNLF", "MEINS", "BSART", "BEDAT",
  "EKORG", "EKGRP", "MEMORY", "DPDAT", "ETENR",
  sum("BRGEW") AS "BRGEW", sum("NTGEW") AS "NTGEW", sum("MENGE") AS "MENGE", sum("DPAMT") AS "DPAMT", sum("WEMNG") AS
  "WEMNG", sum("WAMNG") AS "WAMNG"
  FROM
  GROUP BY "MANDT", "EBELN", "EBELP", "MATNR", "MATKL", "WERKS", "PSTYP", "KNTTP", "IDNLF", "MEINS", "BSART", "BEDAT",
  "EKORG", "EKGRP", "MEMORY", "DPDAT", "ETENR"
```

MANDT	EBELN	EBELP	MATNR	MATKL	WERKS	PSTYP	KNTTP	IDNLF	MEINS	BSART	BEDAT	EKORG	EKGRP	MEMORY	DPDAT	ETENR	BF

```
Statement 'SELECT "MANDT", "EBELN", "EBELP", "MATNR", "MATKL", "WERKS", "PSTYP", "KNTTP", "IDNLF", "MEINS", ...'
successfully executed in 525 ms 783 μs  (server processing time: 195 ms 258 μs)
Fetched 3000 row(s) in 2.530 seconds (server processing time: 5 ms 106 μs)
Result limited to 3000 row(s) due to value configured in the Preferences
```

Step 3: Creating CDS Table Function

1. Open ABAP perspective from Eclipse/ HANA Studio
2. Select system and choose package where CDS data definition of type table function needs to be created.

3. Create a table function by right clicking on the Package and select New other ABAP Repository object-> Data Definition and provide name and description then select next
4. Choose the template "Define Table Function with Parameters" and say finish.

New Data Definition — □ ✕

Templates Ⓓ

Select one of the available templates.

☑ Use the selected template

Ⓓ Extend View
Ⓓ Extend View Entity
Ⓓ Define Table Function with Parameters
Ⓓ Define Abstract Entity with Parameters

Defines the type signature of a client dependent
CDS table function with importing parameters. Th
CDS table function is implemented in the specifie
ABAP method.

```
@EndUserText.label: '${ddl_source_description}'
define table function ${ddl_source_name_editable}
with parameters ${parameter_name} : ${parameter_type}
returns {
  ${client_element_name} : abap.clnt;
  ${element_name} : ${element_type};
  ${cursor}
}
implemented by method ${class_name}=>${method_name};
```

⑦   < Back   Next >   Finish   Cancel

5. Add required fields from HANA calculation view with data type as below and give AMDP class and function name which will create in next step and activate it.

```
 1 @EndUserText.label: '1
 2 define table function
 3 returns
 4 {
 5         MANDT: abap.clnt;
 6         EBELN: abap.char( 10 );
 7         EBELP: abap.int2;
 8         MATNR: abap.char( 18 );
 9         MATKL: abap.char( 9 );
10         WERKS: abap.char( 4 );
11         PSTYP: abap.char( 1 );
12         KNTTP: abap.char( 1 );
13         IDNLF: abap.char( 35 );
14         MEINS: abap.unit( 3 );
15         BSART: abap.char( 4 );
16         BEDAT: abap.dats;
17         EKORG: abap.char( 4 );
18         EKGRP: abap.char( 3 );
19         MEMORY: abap.char( 1 );
20         DPDAT: snwd_date;
21         ETENR: abap.int2;
22         BRGEW: abap.dec( 15, 3 );
23         NTGEW: abap.dec( 15, 3 );
24         MENGE: abap.dec( 15, 3 );
25         DPAMT: abap.dec( 11, 2 );
26         WEMNG: abap.dec( 15, 3 );
27         WAMNG: abap.dec( 15, 3 );
28     |
29 }                          AMDP CLASS    AMDP FUNCTION
30 implemented by method ZCL_AMDP_PUR=>ZF_PUR;
```

Step 4: Create AMDP function for calling HANA native query on calculation view

1. Select package where AMDP Class needs to be created.
2. Create AMDP Class by right clicking on the Package and select New ABAP Class and provide name and description then click finish.
3. Write AMDP Class as below by specifying "interfaces if_amdp_marker_hdb" statement with addition of defining function in public section. In the return statement of function definition use the SQL query that we used in step 2 for displaying calculation view's data and then activate AMDP ABAP Class.

```
▶ ⓒ ZCL_AMDP_PUR ▶
1⊖class ZCL_AMDP_PUR definition
2    public
3    final
4    create public .
5
6  public section.
7  interfaces if_amdp_marker_hdb.
8
9  class-methods: ZF_PUR for table function ZD_TABLE_FUNC_PUR.
10
11
12  protected section.
13  private section.
14  ENDCLASS.
15
16⊖CLASS ZCL_AMDP_PUR IMPLEMENTATION.
17
18⊖method ZF_PUR by database function for hdb language sqlscript
19  OPTIONS READ-ONLY.
20    return
21
22    SELECT "MANDT", "EBELN", "EBELP", "MATNR", "MATKL", "WERKS", "PSTYP", "KNTTP", "IDNLF", "MEINS", "BSART", "B
23    sum("BRGEW") AS "BRGEW", sum("NTGEW") AS "NTGEW", sum("MENGE") AS "MENGE", sum("DPAMT") AS "DPAMT", sum("WEMP
24    FROM "_SYS_BIC"."▓▓▓▓▓▓▓▓▓▓▓▓▓"
25    GROUP BY "MANDT", "EBELN", "EBELP", "MATNR", "MATKL", "WERKS", "PSTYP", "KNTTP", "IDNLF", "MEINS", "BSART",
26  endmethod.
27
28  ENDCLASS.
```

Step 5: Expose table function results to outer CDS with business logic

1. Select package where CDS data definition needs to be created.
2. Create CDS View by right clicking on the Package and select New à other ABAP Repository object-> Data Definition and provide name and description then select next
3. Choose the template "Define view with join" and say finish.
4. Add required business logic to fetch data from table function and other views/ tables and activate it.

```
1⊖@AbapCatalog.sqlViewName: '▓▓▓▓▓▓▓'
2  @AbapCatalog.compiler.compareFilter: true
3  @AbapCatalog.preserveKey: true
4  @AccessControl.authorizationCheck: #CHECK
5  @ObjectModel.usageType.serviceQuality: #B
6  @Analytics.dataCategory: #CUBE
7  @EndUserText.label: 'PURCHASING'          Table function
8                                                 Name
9  define view ▓▓_ as select from ▓▓_▓_▓ as PUR
10
11  left outer join ▓▓▓▓A as MAKT on PUR.MATNR = MAKT.matnr
12
13  association [1] to t001w as _Plant on $projection.WERKS = _Plant.werks
14
15  association [1] to ▓▓▓▓▓: as _EKES on $projection.EBELN = _EKES.ebeln and
16                                          $projection.EBELP = _EKES.ebelp
17 {
18
19      key PUR.EBELN,
20      key PUR.EBELP,
21      PUR.MATNR,
22      _EKES.vbeln,
23      _EKES.vbelp,
24      _EKES.xblnr,
25⊖     @Consumption.labelElement: 'MATNR'
26      @Consumption.quickInfoElement: 'MAKTX' |
27      MAKT.maktx,
28      MATKL,
29      WERKS,
30⊖     @Consumption.labelElement: 'WERKS'
31      @Consumption.quickInfoElement: 'Plant_Name'
32      _Plant.name1 as Plant_Name,
33      PSTYP as Item_Category,
34      KNTTP as Assignment_Category,
35      IDNLF as Vendor_Material,
36      MEINS,
```

5. Run the CDS View and see the results.

This concludes the above blog post. Hope the blog helped you. Please feel free to give feedback and comments.

Happy Learning!!!

Alert Moderator

Assigned Tags

SAP HANA

S/4HANA CDS View

Similar Blog Posts ⌄

SAP S/4HANA Embedded Analytics vs. SAP HANA Native Modeling

By Johann Schedrin   Aug 19, 2021

Enabling Search in SAP HANA using CDS

By Mithun Sandeep Korlepara   Dec 04, 2014

SAP HANA Views

By Johann Schedrin   Feb 25, 2021

Related Questions

How to use CDS views in HANA Calculation view?

By Former Member   Aug 02, 2016

Differenece between HANA CDS and ABAP prospective CDS views

By ASHUTOSH SHRIVASTAVA   Mar 27, 2019

Best Practices HANA Modeling: Attr/An/Calc Views or CDS

By Former Member   Jan 27, 2016

Join the Conversation

SAP TechEd

Tune in for tech talk. Stay for inspiration. Upskill your future.

Coffee Corner

Join the new Coffee Corner Discussion Group.

7 Comments

You must be Logged on to comment or reply to a post.

Sven Knöpfler

September 1, 2020 at 7:29 am

Thanks for the How-to. That is especially interesting if you want to map input parameters from the calculation views. If you dont need that, you could also use an external view:

https://help.sap.com/viewer/6811c09434084fd1bc4f40e66913ce11/1709.000/en-US/4ec059036e391014adc9fffe4e204223.html

Like 0 | Share

Santosh Tankala
September 7, 2020 at 5:25 am

Thanks for great explanation!

I have a query, will S/4 system able to identify SYS_BIC schema directly when we are calling it AMDP class?

Regards,

Santosh T.

Like 0 | Share

> **Abhishek Patel** | Blog Post Author
> December 8, 2020 at 12:18 pm
>
> Yes, as system connection is already there S4 can access HANA database catalog schemas and objects.
>
> Like 0 | Share

prem kumar
June 26, 2021 at 1:52 am

Nice but how do we call calculation view with input parameter? The sql view of HANA model generates place holders for input parameter to which am unable to pass the importing parameter of the class.

Regards,

Prem

Like 0 | Share

> **Thiago Teixeira de Carvalho Dias**
> June 30, 2021 at 10:16 pm
>
> Hi prem kumar,
>
> FROM "_SYS_BIC"."<Calculation View Name>"(placeholder."$$P_MANDT$$"=>:p_client,
>
> placeholder."$$P_FILIAL$$"=>:p_filial,
>
> placeholder."$$P_DT_INI$$"=>:p_dt_ini,

```
placeholder."$$P_DT_FIN$$"=>:p_dt_fin,

placeholder."$$P_EMPRESA$$"=>:p_empresa);
```

Regards,
Thiago Dias.

Like 1 | Share

prem kumar
July 2, 2021 at 2:18 pm

Thank you Thiago Dias. Much appreciated.

Like 0 | Share

Jose Guilas
October 28, 2021 at 6:44 am

how and where did you get the 'ZD_TABLE_FUNC_PUR'

Like 0 | Share

Find us on

Privacy	Terms of Use
Legal Disclosure	Copyright
Trademark	Cookie Preferences
Newsletter	Support