Maksim Alyapyshev

February 17, 2017 | 7 minute read

How to use SQL window functions in ABAP CDS views

2 11 10,410

Follow

Like

RSS Feed

Hi!

In this blog I would like to consider a case of using **Windows Fuctions** in **ABAP CDS views** based on data in S/4HANA system.

I also take case of some features like:

- Main CDS view syntax and semantics
- Parameters in CDS views
- Consumption options of CDS views

Introduction

Currently SQL functions like LEAD, LAG, NTILE and other are not supported in CDS views.

There is a link to available CDS functions NW 7.51 documentation:

http://help-legacy.sap.com/abapdocu_751/en/abencds_language_elements.htm

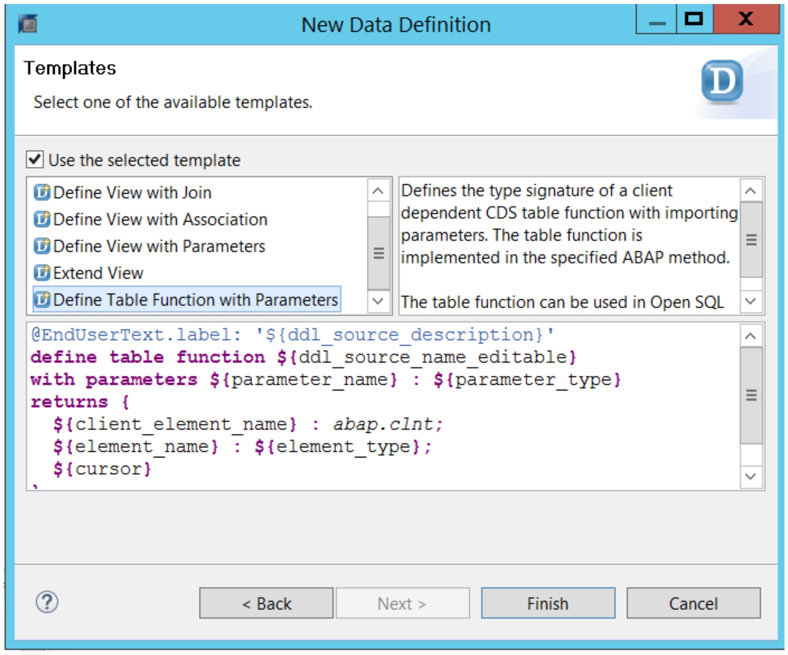
A roundabout solution is to use a CDS view based on Table Functions where you could use all power of SQL Script language.

There is a link to Windows Function documentation:

http://help-legacy.sap.com/saphelp_hanaplatform/helpdata/en/20/a353327519101495dfd0a87060a0d3/content.htm

My demo example is based on **SFLIGHT data model**, that is commonly used in ABAP courses and always available in ERP and S/4HANA system. ABAP CDS Views could be created in ABAP perspective of HANA studio.

1.Creating CDS view (Data definition) in our package with template:


New Data Definition

Templates

Select one of the available templates.

☑ Use the selected template

- Define View with Join
- Define View with Association
- Define View with Parameters
- Extend View
- Define Table Function with Parameters

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

The table function can be used in Open SQL

```
@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}
```

Define return structure and name of class and method with logic.

```
@EndUserText.label: 'CDS with TF'
define table function Z05_Cds_Tf
with parameters
    @Environment.systemField: #CLIENT
    p_clnt   :abap.clnt,
    p_carrid :s_carr_id,
    p_currency : s_currcode

returns {
    client   :s_mandt;
    carrid: s_carr_id;
    carrname :s_carrname;
    connid   :s_conn_id;
    fldate : s_date;
    flmonth: /BI0/OICALMONTH;
    paymentsum: s_sum;
    currency: s_currcode;
    paymentsumnew: s_sum;
  }
  implemented by method
    Z05_CL_DEMO_CDS=>GET_DATA_Z05_CDS_TF;
```

2. Create class/method defined in definition of table function.

In my example I use window function to calculate **running total** by amount fields. **Also you could use all other functions like LEAD, LAG, etc.**

Note that:

- You could include one select in another, in example I need to calculate calendar month for flight date first and use it in subsequent select ***in partition*** by statement.
- I am using tables sflight and scarr, but it is also possible to use ABAP views generated by standard or customer CDS view as an entry point for calculation.
- CDS view with table function could not be consumed directly.
- Semantics @VDM.viewType: #CONSUMPTION is forbidden there.

```
class Z05_CL_DEMO_CDS definition
  public
  final
  create public .

public section.
 "Include interface
 INTERFACES if_amdp_marker_hdb.

 CLASS-METHODS GET_DATA_Z05_CDS_TF
                  FOR TABLE FUNCTION Z05_Cds_Tf.

protected section.
private section.
ENDCLASS.



CLASS Z05_CL_DEMO_CDS IMPLEMENTATION.

METHOD GET_DATA_Z05_CDS_TF
         BY DATABASE FUNCTION FOR HDB
         LANGUAGE SQLSCRIPT
         OPTIONS READ-ONLY
         USING sflight scarr.
RETURN    SELECT  client,
                  carrid,
                  carrname,
                  connid,
                  fldate,
                  flmonth,
                  paymentsum,
                  currency,
                  sum(paymentsum) over (partition by carrid, co
                                        order by carrid, connid
    FROM
    (
    SELECT sf.mandt as client,
                  sf.carrid,
                  sc.carrname,
                  sf.connid,
                  sf.fldate,
                  left(sf.fldate,6) as flmonth,
                  sf.paymentsum,
                  sf.currency
                  FROM sflight AS sf
                    INNER JOIN scarr AS sc ON sf.mandt = sc.man
                                              sf.carrid = sc.ca
                    WHERE sf.mandt = :p_clnt and
                          sf.carrid = :p_carrid and
                          sf.currency = :p_currency
             )
     order by
          carrid,
          connid,
          fldate;


  ENDMETHOD.
ENDCLASS.
```

3. Create a consumption CDS view.

Note:

- I am using CUBE type for creating a analytical query later.

- Look at how parameters transferred between CDS views. **They must be filled and unfortunately could not be optional now.**
- You could include another join or associations if needed at this step. I include 2 assosiations to demonstrate a syntax. Fields from CDS views ***Z00_I_Airline*** and ***Z00_I_FlightConnection*** will be available in subsequent analytical query CDS views.
- Look at semantics of linking code field (***carrid***) and description field (**carrname**)
- Look at semantics of describing amount with currency fields.
- This CDS of CUBE type is generating transient provider. Which is visible for Analysis for Excel and BEx Query Designer for example.

```
@AbapCatalog.sqlViewName: 'Z05_CCDS'
@AbapCatalog.compiler.compareFilter: true
@VDM.viewType: #CONSUMPTION
@Analytics.dataCategory: #CUBE
@Analytics.dataExtraction.enabled: true
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'Demo consume CDS'

define view Z05_C_Cds
with parameters
    p_clnt   :abap.clnt,
    p_carrid :s_carr_id,
    p_currency : s_currcode
as select from z05_cds_tf (p_clnt:$parameters.p_clnt,p_carrid:$p
  association [0..1] to Z00_I_Airline as _Airline
      on $projection.carrid = _Airline.Airline
  association [0..1] to Z00_I_FlightConnection as _FlightConnect
    on $projection.carrid = _FlightConnection.Airline and
       $projection.connid = _FlightConnection.FlightConnection
{
    @ObjectModel.text.element:  [ 'carrname' ]
    @ObjectModel.foreignKey.association: '_Airline'
    key z05_cds_tf.carrid,
     @ObjectModel.foreignKey.association: '_FlightConnection'
    key z05_cds_tf.connid,
    key z05_cds_tf.fldate,

    @Semantics.text: true
    z05_cds_tf.carrname,
    z05_cds_tf.flmonth,
    @Semantics.amount.currencyCode: 'currency'
    @DefaultAggregation: #SUM
    Z05_Cds_Tf.paymentsum,
    @Semantics.amount.currencyCode: 'currency'
    @DefaultAggregation: #SUM
    Z05_Cds_Tf.paymentsumnew,
    @Semantics.currencyCode: true
    z05_cds_tf.currency,

     /* Associations */
    _Airline,
    _FlightConnection

}
```

4. Create a analytical query CDS view:

Note:

- OData service will be using for opening data of this query in **Fiori Query Browser** and **Smart Business KPI Tool.**
- Look at semantics definition of rows, columns and free characteristics of the query. Variables are also supported, but I didn't include in example.
- Look at exception aggregation semantics, that is useful, when flight date is excluded fro drill-down by user. It works only for formulas, that's why i define it like this.
- Look at totals semantics example.
- Last field show that fields from association CDS could be included is needed.
- Look at renaming of amount field with annotation **@EndUserText**. I need it to distinguish between my new running total field and standard amount field. This is also usefull think, cause this label could be translated at different languages (by SE63 t-code).

```
@AbapCatalog.sqlViewName: 'Z05_CCDSQ'
@AbapCatalog.compiler.compareFilter: true
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'CDS Demo Query'
@VDM.viewType: #CONSUMPTION
@Analytics.query: true
@OData.publish: true

define view Z05_C_Cds_Query
with parameters
    --@Consumption.defaultValue: '400'
    @Environment.systemField: #CLIENT
    p_clnt   :abap.clnt,
    @Consumption.defaultValue: 'AA'
    p_carrid :s_carr_id,
    @Consumption.defaultValue: 'USD'
    p_currency : s_currcode
    as select from Z05_C_Cds (p_clnt:$parameters.p_clnt,
                              p_carrid:$parameters.p_carrid,
                              p_currency:$parameters.p_currency)
              @AnalyticsDetails.query.axis: #ROWS
              @ObjectModel.text.element:  [ 'AirLineName' ]
              Z05_C_Cds.carrid,
              @AnalyticsDetails.query.axis: #ROWS
              Z05_C_Cds.connid,
              @AnalyticsDetails.query.axis: #ROWS
              Z05_C_Cds.flmonth,
              @AnalyticsDetails.query.axis: #ROWS
              @AnalyticsDetails.query.totals: #SHOW
              Z05_C_Cds.fldate,

              @AnalyticsDetails.query.axis: #COLUMNS
              Z05_C_Cds.paymentsum,

              @AnalyticsDetails.query.axis: #COLUMNS
              @EndUserText.label: 'Booking Total New'
              @DefaultAggregation: #FORMULA
              @AnalyticsDetails: { exceptionAggregationSteps:
             [{exceptionAggregationBehavior: #MAX,
             exceptionAggregationElements: ['carrid','connid','
              Z05_C_Cds.paymentsumnew + 0 as paymentsumnew,

              @AnalyticsDetails.query.axis: #COLUMNS
              Z05_C_Cds.currency,

              @Semantics.text: true
              Z05_C_Cds._Airline.AirlineName as AirLineName


}
```

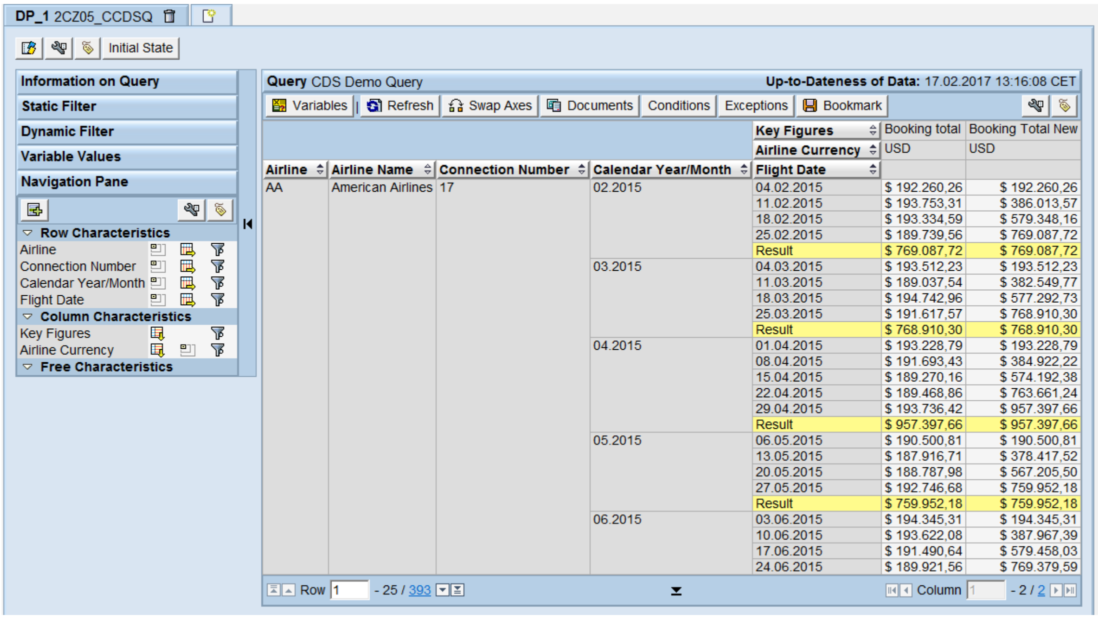
5. Consume query in RSRT:

Note:

- Booking total New is our running total key figure. Remember that it was calculated in CDS view with table function:

-
  ```
  ion by carrid, connid,flmonth
              order by carrid, connid, fldate) as paymentsumnew
  ```

- *Airline* and *Airline Name* are like BW characterictic key with text, because of correct semantic definition in previous point (point 4).
- Look at query name 2CZ05_CDSQ. 2C + sql view name from point 4.



DP_1 2CZ05_CCDSQ

Initial State

Information on Query
Static Filter
Dynamic Filter
Variable Values
Navigation Pane

Row Characteristics
Airline
Connection Number
Calendar Year/Month
Flight Date
Column Characteristics
Key Figures
Airline Currency
Free Characteristics

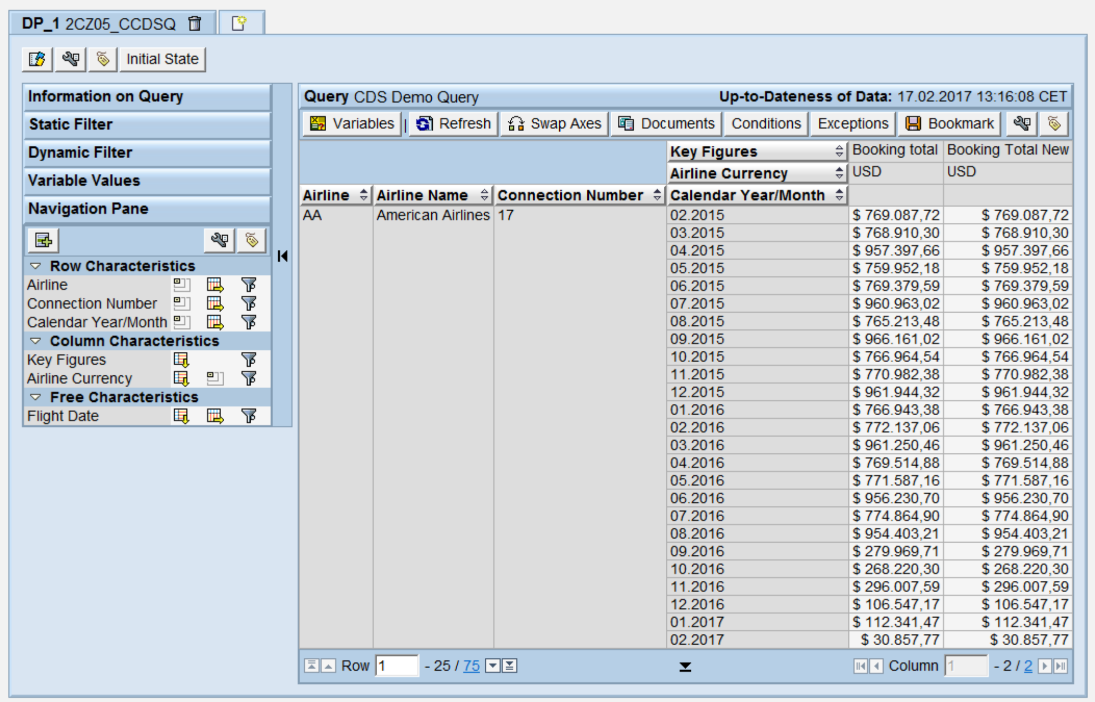
Query CDS Demo Query **Up-to-Dateness of Data:** 17.02.2017 13:16:08 CET

Variables | Refresh | Swap Axes | Documents | Conditions | Exceptions | Bookmark

				Key Figures	Booking total	Booking Total New
				Airline Currency	USD	USD
Airline	**Airline Name**	**Connection Number**	**Calendar Year/Month**	**Flight Date**		
AA	American Airlines	17	02.2015	04.02.2015	$ 192.260,26	$ 192.260,26
				11.02.2015	$ 193.753,31	$ 386.013,57
				18.02.2015	$ 193.334,59	$ 579.348,16
				25.02.2015	$ 189.739,56	$ 769.087,72
				Result	$ 769.087,72	$ 769.087,72
			03.2015	04.03.2015	$ 193.512,23	$ 193.512,23
				11.03.2015	$ 189.037,54	$ 382.549,77
				18.03.2015	$ 194.742,96	$ 577.292,73
				25.03.2015	$ 191.617,57	$ 768.910,30
				Result	$ 768.910,30	$ 768.910,30
			04.2015	01.04.2015	$ 193.228,79	$ 193.228,79
				08.04.2015	$ 191.693,43	$ 384.922,22
				15.04.2015	$ 189.270,16	$ 574.192,38
				22.04.2015	$ 189.468,86	$ 763.661,24
				29.04.2015	$ 193.736,42	$ 957.397,66
				Result	$ 957.397,66	$ 957.397,66
			05.2015	06.05.2015	$ 190.500,81	$ 190.500,81
				13.05.2015	$ 187.916,71	$ 378.417,52
				20.05.2015	$ 188.787,98	$ 567.205,50
				27.05.2015	$ 192.746,68	$ 759.952,18
				Result	$ 759.952,18	$ 759.952,18
			06.2015	03.06.2015	$ 194.345,31	$ 194.345,31
				10.06.2015	$ 193.622,08	$ 387.967,39
				17.06.2015	$ 191.490,64	$ 579.458,03
				24.06.2015	$ 189.921,56	$ 769.379,59

Row 1 - 25 / 393 Column 1 - 2 / 2

If we delete flight date from drill-down we will see correct aggregation in **Booking Total New.** For this is become true we use **exception aggregation** in point 4.



DP_1 2CZ05_CCDSQ

Initial State

Information on Query
Static Filter
Dynamic Filter
Variable Values
Navigation Pane

Row Characteristics
Airline
Connection Number
Calendar Year/Month
Column Characteristics
Key Figures
Airline Currency
Free Characteristics
Flight Date

Query CDS Demo Query **Up-to-Dateness of Data:** 17.02.2017 13:16:08 CET

Variables | Refresh | Swap Axes | Documents | Conditions | Exceptions | Bookmark

			Key Figures	Booking total	Booking Total New
			Airline Currency	USD	USD
Airline	**Airline Name**	**Connection Number**	**Calendar Year/Month**		
AA	American Airlines	17	02.2015	$ 769.087,72	$ 769.087,72
			03.2015	$ 768.910,30	$ 768.910,30
			04.2015	$ 957.397,66	$ 957.397,66
			05.2015	$ 759.952,18	$ 759.952,18
			06.2015	$ 769.379,59	$ 769.379,59
			07.2015	$ 960.963,02	$ 960.963,02
			08.2015	$ 765.213,48	$ 765.213,48
			09.2015	$ 966.161,02	$ 966.161,02
			10.2015	$ 766.964,54	$ 766.964,54
			11.2015	$ 770.982,38	$ 770.982,38
			12.2015	$ 961.944,32	$ 961.944,32
			01.2016	$ 766.943,38	$ 766.943,38
			02.2016	$ 772.137,06	$ 772.137,06
			03.2016	$ 961.250,46	$ 961.250,46
			04.2016	$ 769.514,88	$ 769.514,88
			05.2016	$ 771.587,16	$ 771.587,16
			06.2016	$ 956.230,70	$ 956.230,70
			07.2016	$ 774.864,90	$ 774.864,90
			08.2016	$ 954.403,21	$ 954.403,21
			09.2016	$ 279.969,71	$ 279.969,71
			10.2016	$ 268.220,30	$ 268.220,30
			11.2016	$ 296.007,59	$ 296.007,59
			12.2016	$ 106.547,17	$ 106.547,17
			01.2017	$ 112.341,47	$ 112.341,47
			02.2017	$ 30.857,77	$ 30.857,77

Row 1 - 25 / 75 Column 1 - 2 / 2

Important: You could say that it is possible to create things like this BEx query by Query Designer.

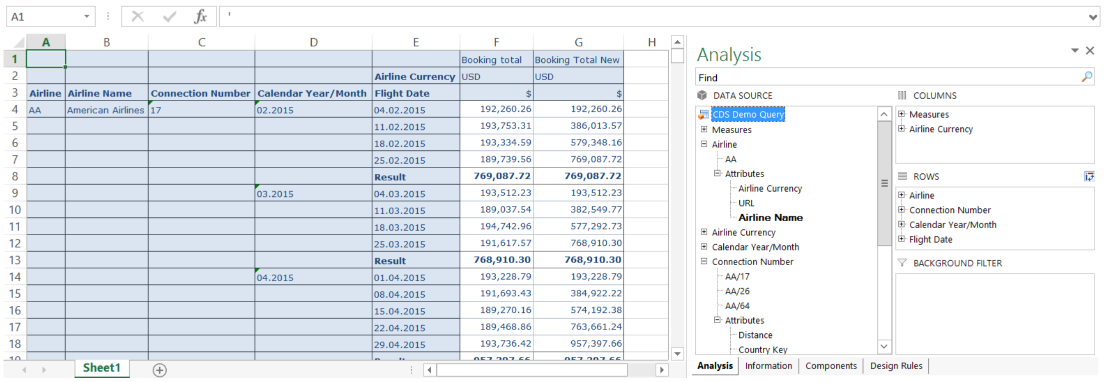
In most cases the answer is "Yes", but:

- BEx Query Designer could not be used at all with S/4HANA installation, and in S/4HANA Cloud scenarious.
- We couldn't consume the results very easy and simple by S/4HANA Fiori Interface (later will be shown)

6. Consumption in Analysis for Excel is similar:

Note:

- Association CDS view fields are shown as attributes which is important and usefull.
- At the variables screen (not shown) parameters we filed by default values we define earlier (point 4).



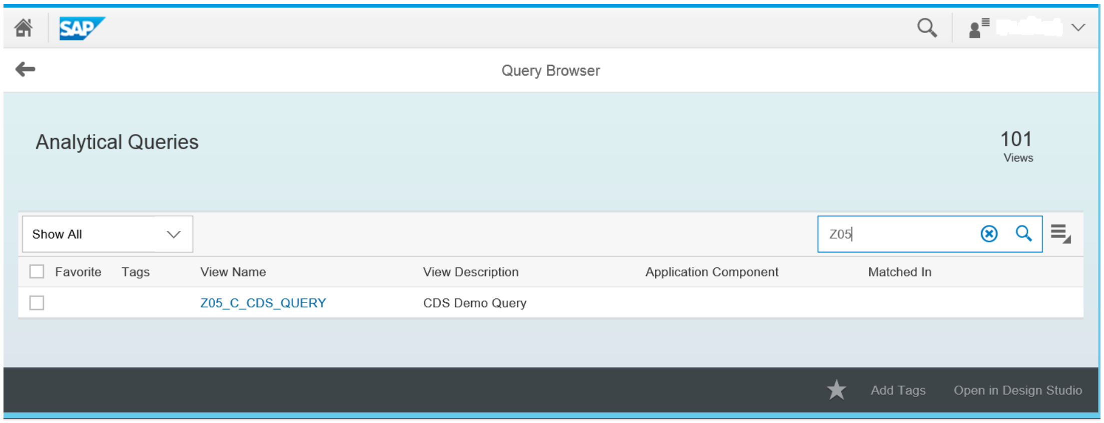
	A	B	C	D	E	F	G
1						Booking total	Booking Total New
2					Airline Currency	USD	USD
3	Airline	Airline Name	Connection Number	Calendar Year/Month	Flight Date	$	$
4	AA	American Airlines	17	02.2015	04.02.2015	192,260.26	192,260.26
5					11.02.2015	193,753.31	386,013.57
6					18.02.2015	193,334.59	579,348.16
7					25.02.2015	189,739.56	769,087.72
8					**Result**	**769,087.72**	**769,087.72**
9				03.2015	04.03.2015	193,512.23	193,512.23
10					11.03.2015	189,037.54	382,549.77
11					18.03.2015	194,742.96	577,292.73
12					25.03.2015	191,617.57	768,910.30
13					**Result**	**768,910.30**	**768,910.30**
14				04.2015	01.04.2015	193,228.79	193,228.79
15					08.04.2015	191,693.43	384,922.22
16					15.04.2015	189,270.16	574,192.38
17					22.04.2015	189,468.86	763,661.24
18					29.04.2015	193,736.42	957,397.66

7. Consumption in S/4HANA Fiori interface **Query Browser**.

Note:

- First of all needed basic setting of configuration back/front servers are done.
- OData service for Analytical Query is active. For activation go to (/IWFND/MAINT_SERVICE) at Front Server.

Open a **Query Browser App** and find our Query Z05_ and go to open in Design Studio. You could create a tile from this.



Query Browser

Analytical Queries

101 Views

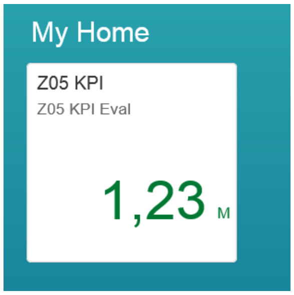
Search: Z05

□	Favorite	Tags	View Name	View Description	Application Component	Matched In
□			Z05_C_CDS_QUERY	CDS Demo Query		

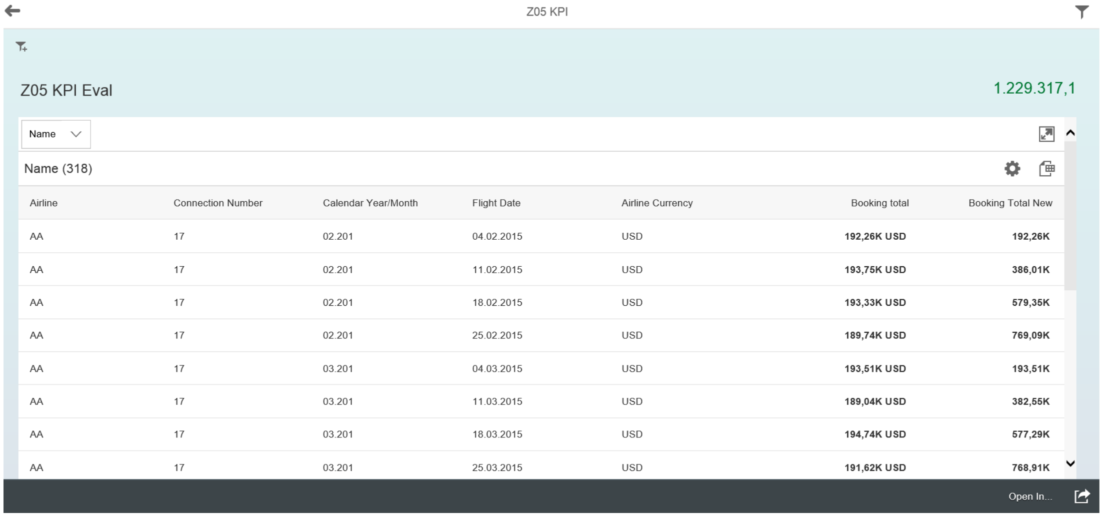
8. Consumption in S/4HANA **Smart Business KPI Modeler**.

You could create a KPI, evaluation, tile and drill-downs. I'm not going into details, because there are already good blogs on this thema. Only one thing parameters of this CDS view must be defined once more time, during creation of evaluation.

The Tile is:



After clicking it (just simple table as example) there could by bars, lines and so on.



Z05 KPI Eval 1.229.317,1

Name (318)

Airline	Connection Number	Calendar Year/Month	Flight Date	Airline Currency	Booking total	Booking Total New
AA	17	02.201	04.02.2015	USD	192,26K USD	192,26K
AA	17	02.201	11.02.2015	USD	193,75K USD	386,01K
AA	17	02.201	18.02.2015	USD	193,33K USD	579,35K
AA	17	02.201	25.02.2015	USD	189,74K USD	769,09K
AA	17	03.201	04.03.2015	USD	193,51K USD	193,51K
AA	17	03.201	11.03.2015	USD	189,04K USD	382,55K
AA	17	03.201	18.03.2015	USD	194,74K USD	577,29K
AA	17	03.201	25.03.2015	USD	191,62K USD	768,91K

Summary:

So we went through a long way actually:

- Creation of CDS based of Table Function.
- Creation of Class/Method with SQL Script logic included using of windows functions expressions.
- Creating of consumption Cube ABAP CDS view.
- Creation of consumption analytical query ABAP CDS view.
- Viewing results in RSRT, Analysis for Excel, S/4HAHA Fiori Query Browser.
- Viewing result of KPI tile (SmartBusiness KPI Modeler App) based on the same analytical query CDS view.

Thank you for attention!

See you at my later blogs.

Alert Moderator

Assigned tags

SAP S/4HANA

ABAP Development

SAP Fiori for SAP S/4HANA

SQL

abap cds views

cds

embedded analytics

Similar Blog Posts

Column Concatenation in ABAP CDS Views

By Muruga MuthuKrishnan Jun 11, 2019

Performance Optimization for ABAP CDS view

By Prosenjit Das Neogi Jul 03, 2018

Part#1. SAP CDS views Demystification

By Sanjeev Kumar Oct 21, 2019

Related Questions

HANA CDS View Annotations

By Sreekanth Surampally Sep 19, 2017

How does security work for S/4 HANA reporting?

By AJAYKUMAR SUGUMARAN Oct 05, 2019

ABAP CDS View: join tables on columns of different type

By Former Member Dec 15, 2015

2 Comments

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

Sergey Shablykin

February 22, 2017 at 7:05 am

Another great blog from Max!

Could you clarify is it possible to use ABAP-based method for table function?

Like 0 | Share

Maksim Alyapyshev | Blog Post Author

February 22, 2017 at 9:58 am

It is possible to call only HDB procedures methods as I know

Like 0 | Share