Virtual Characteristics and Key Figures Revisited: the Missing Part Debugger

Applies to:

SAP BW 3.5, SAP BI 7.0. For more information, visit the EDW homepage.

Summary

This document is extension of already submitted many documents on Virtual Characteristics and Key figures, here the main intention is to explain the missing part of implementing VCKF i.e. When, why and how to use debugger while implementing Virtual Characteristics and Key figures.

Author: Vipul Goyal

Company: Accenture

Created on: 1st June 2010

Author Bio

Vipul Goyal is a certified SAP BI Consultant working for Accenture Services Pvt. Ltd. He has more than 4 years of experince in SAP BI/BW

Table of Contents

Business Scenario 3
Background 3
Step by Step Guide 3
Related Content 13
Disclaimer and Liability Notice 14

Business Scenario

An organization has many contractors working for it with planned Start date and End date for their contact. Now the managers, need to know how many days are left for expiry of their contract, so that they can plan whether to extend the contract, or release him from his duties.

Now in this example, the use of Virtual Key figure would be shown to calculate number of Days for Expiry using End Date of Contract as Reference Characteristic.

Background

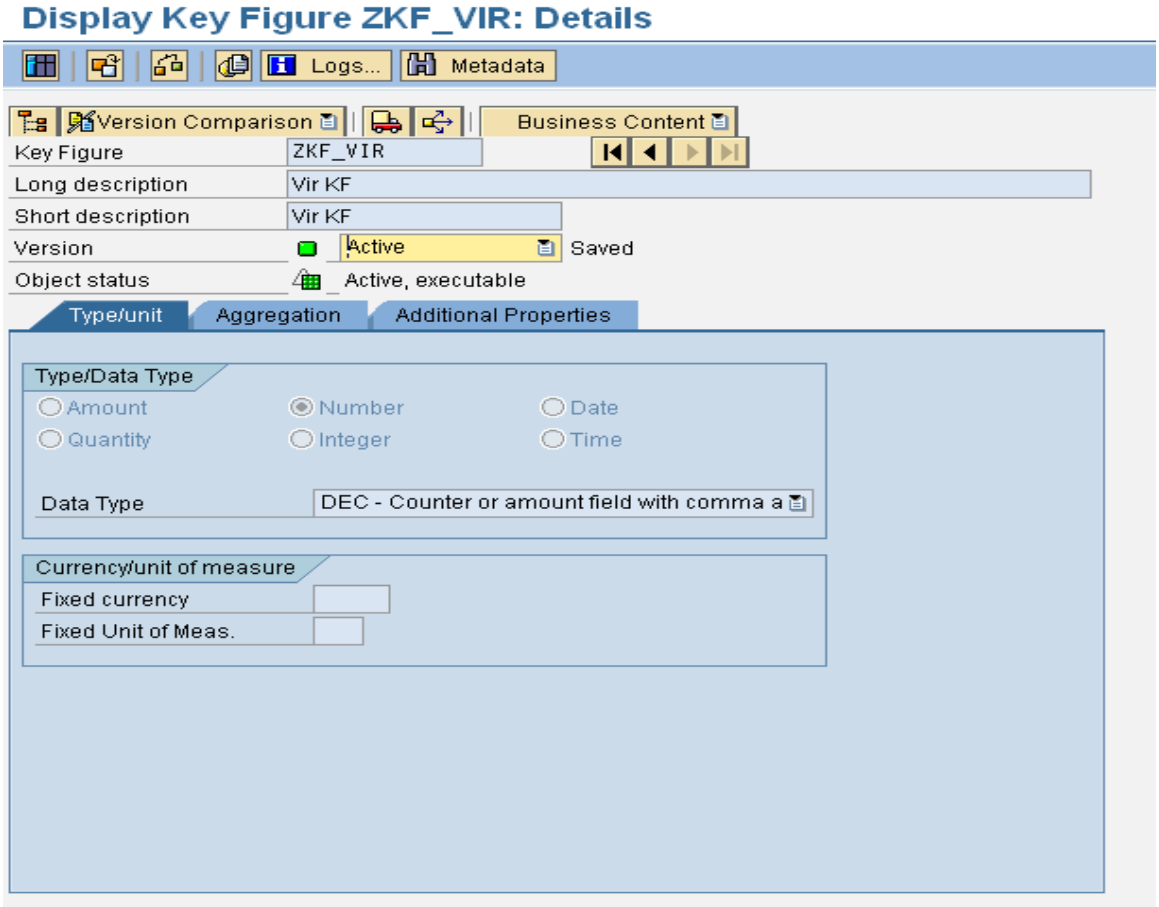
A virtual characteristic or a virtual key figure is an InfoObject, which is defined within the InfoProvider as metadata without having any data stored physically.

Virtual Key Figures (KF)/ Characteristics can be used to create the Query more dynamic. The value of Virtual KF/Characteristics does not store in the Data Target, it is calculated at runtime (At Query Execution time). Complex logic, database access is possible by using Virtual Key Figure and Characteristics.
The implementation part includes following steps, which also include Debugging part, which is used to identify position of referred Characteristic and Virtual Key Figure without which I was not able to implement Virtual concept.

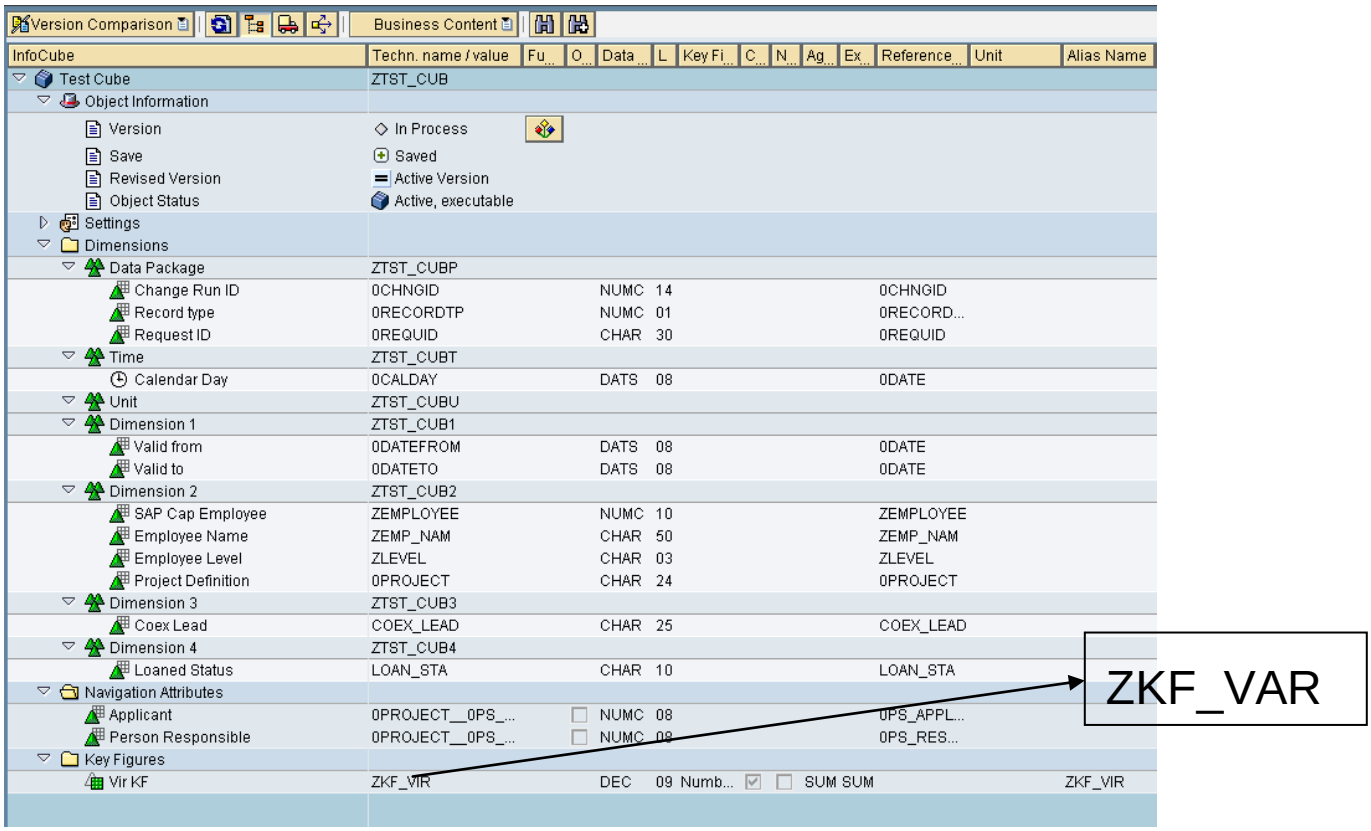
- Create InfoProvider with virtual Key figure
- Create a query and add virtual Key figure and other relevant Characteristics like Contract End Date
- Implement RSR_OLAP_BADI
- Identify position of referred Characteristic and Virtual Key Figure using debugger.

Step by Step Guide

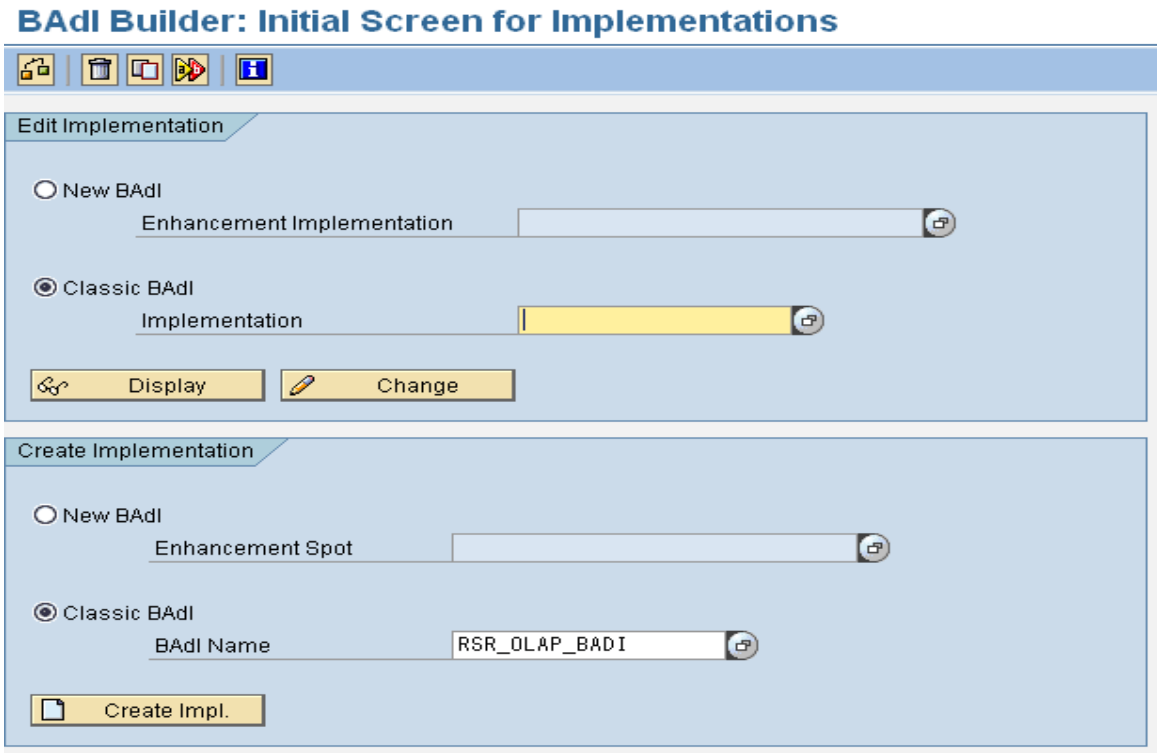
1. Create virtual KF in RSA1



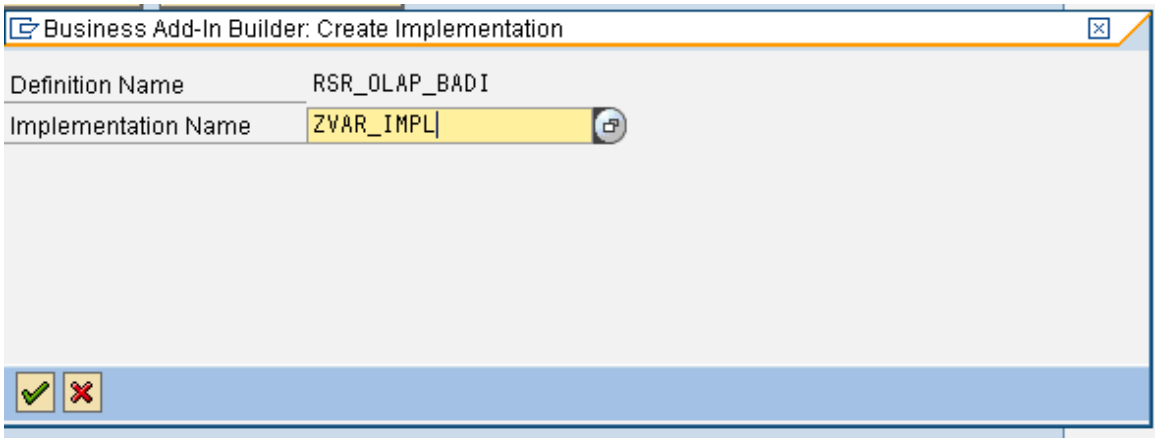
2. Include the char/kf in the cube/dso



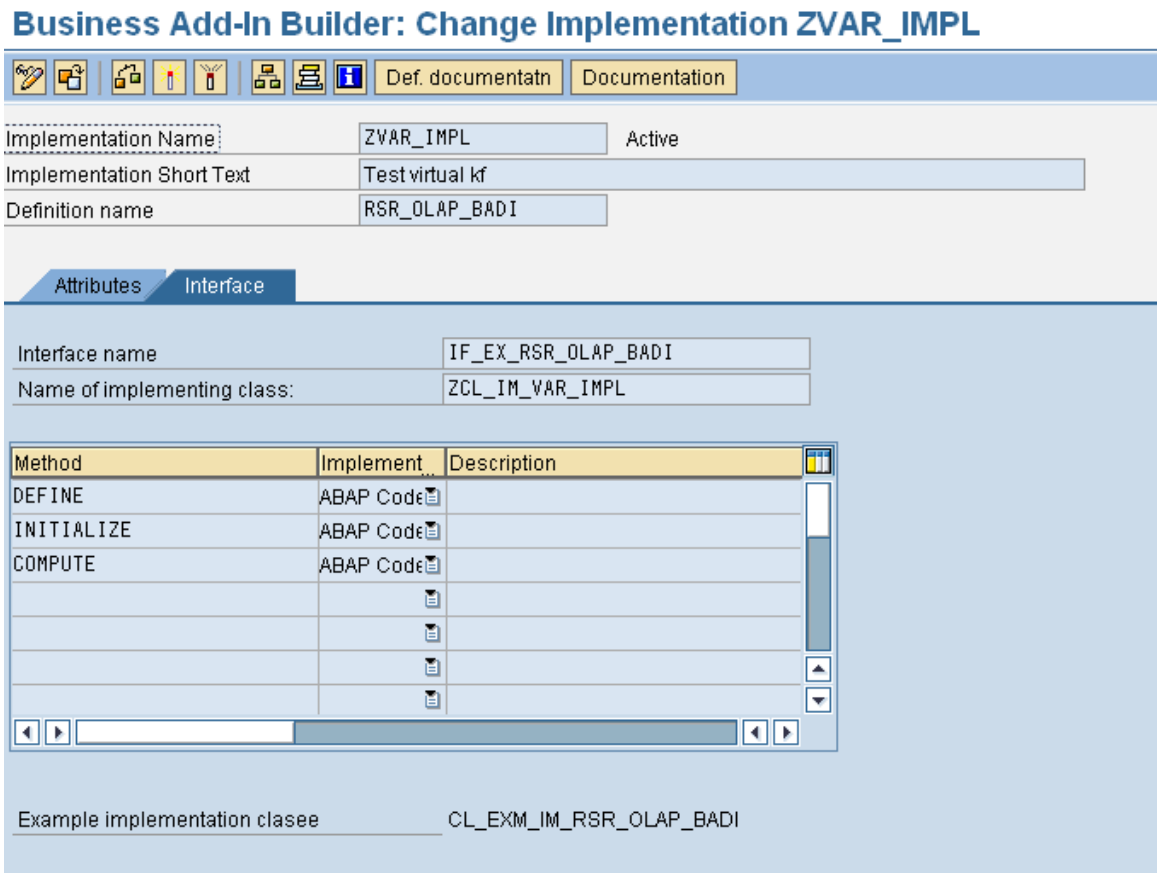
3. Go to SE19 Tcode and put RSR_OLAP_BADI under Create Implementation->Classical BAdi->BAdi Name



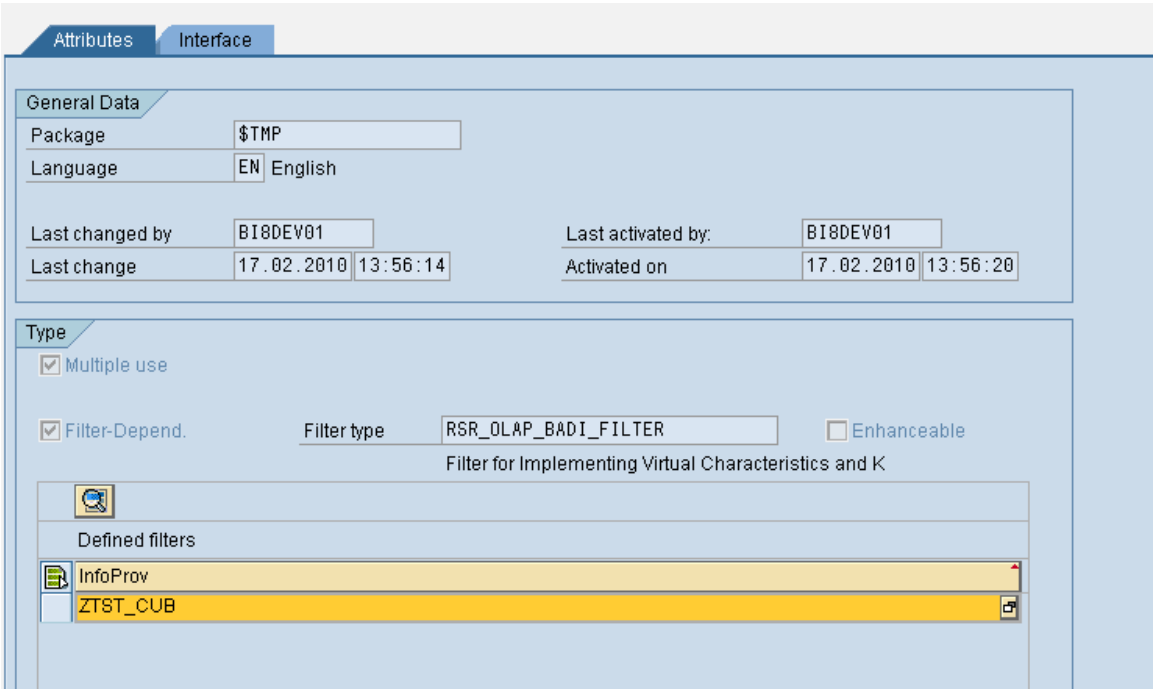
4. Enter your Implementation Name



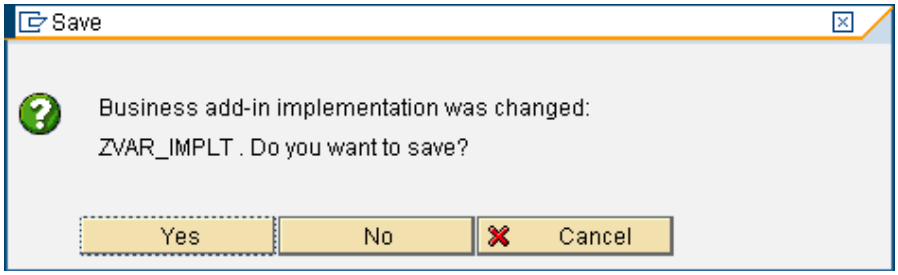
5. Give Description to your new implementation.



6. On attribute Tab, give Infoprovider name as filter.



7. Go to Interface tab and Double click on “Name of implementing class:” which is ZCL_IM_VAR_IMPLT, and say yes when following screen Pops up to save your implementation.

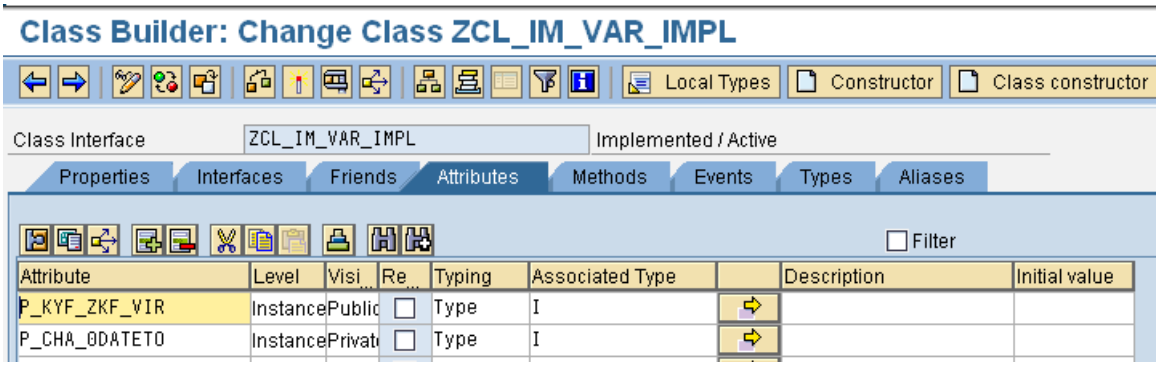


8. Add the attributes to the Implementation Class. The attributes must be of type I and should follow the standard naming convention.

In this example the Name of the implementing class is: ZCL_IM_VAR_IMPLT and add the attributes as shown in Screenshot below

Always use the convention P_KYF_<Key Figure InfoObject> if it is Key Figure and if it’s characteristic then use P_CHA_<Char. InfoObject>.

I have used Key Figure ZKF_VIR as virtual KF and it would be calculated based on Characteristic 0DATE_TO (End date of Contract)



9. After you added the required attributes go to Methods Tab you will find three methods as below:

- IF_EX_RSR_OLAP_BADI~DEFINE
- IF_EX_RSR_OLAP_BADI~INITIALIZE
- IF_EX_RSR_OLAP_BADI~COMPUTE

10. IF_EX_RSR_OLAP_BADI~DEFINE

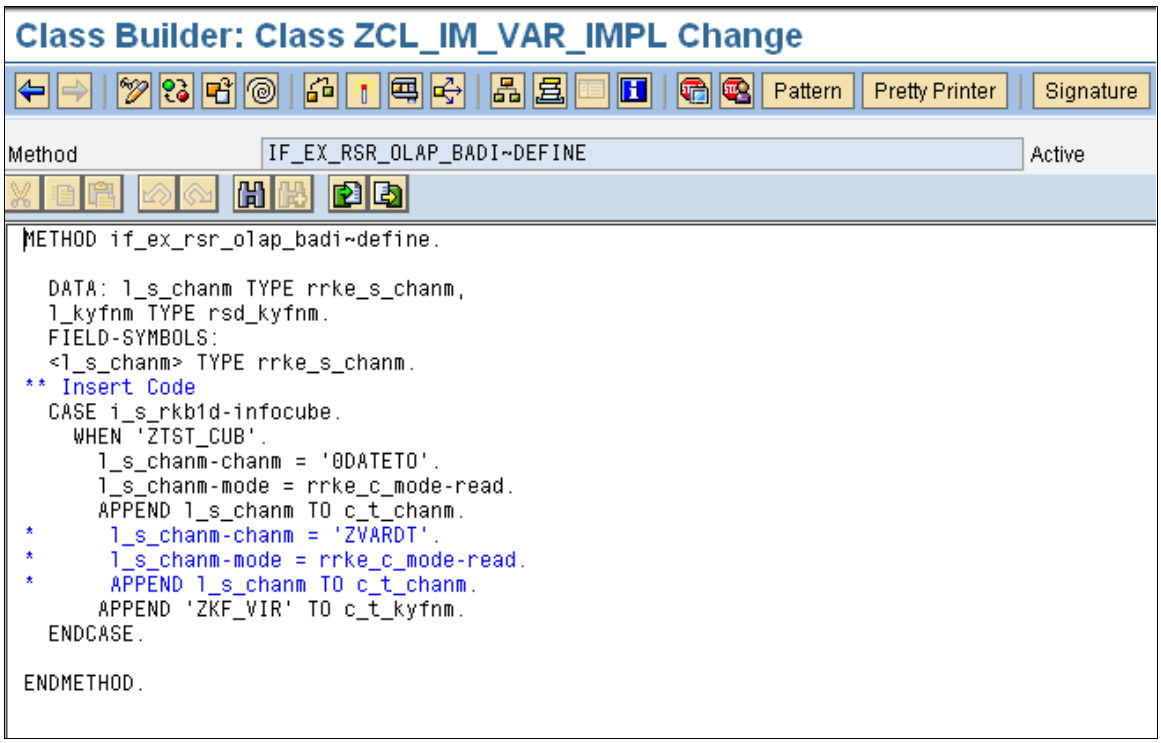
In define method, you define of characteristics and KFs involved in the calculations of VKF assign to system internal tables c_t_chanm and c_t_kyfnm and restrict this assignment in the case statement. to the cube used.

e.g. We are restricting following assignment to the cube 'ZTST_CUB'

Characteristic 0DATE_TO is assigned to c_t_chanm in read mode, as we have read data for this char. during runtime to calculate VKF.

And Key Figure 'ZKF_VIR' is assigned to internal table c_t_kyfnm.

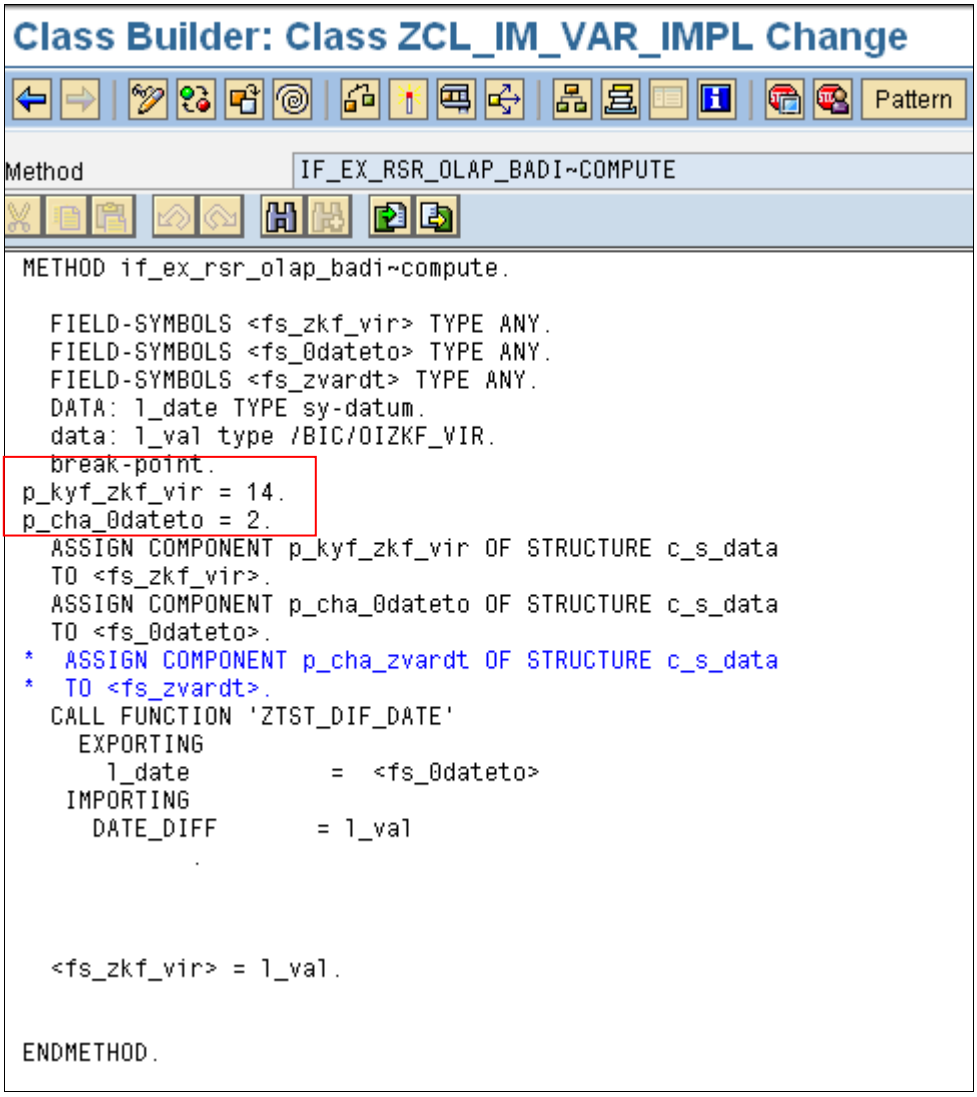
You can also define other Local declarations in this Method.



And then in Initialize Method, you can assign values to the Local Declarations if any, otherwise you can leave it as it, as done in this case.

11. IF_EX_RSR_OLAP_BADI~COMPUTE

Now in compute method all the calculations are done, below is the sample code for the same.




```
METHOD if_ex_rsr_olap_badi~compute.

  FIELD-SYMBOLS <fs_zkf_vir> TYPE ANY.
  FIELD-SYMBOLS <fs_0dateto> TYPE ANY.
  FIELD-SYMBOLS <fs_zvardt> TYPE ANY.
  DATA: l_date TYPE sy-datum.
  data: l_val type /BIC/OIZKF_VIR.
  break-point.
p_kyf_zkf_vir = 14.
p_cha_0dateto = 2.
  ASSIGN COMPONENT p_kyf_zkf_vir OF STRUCTURE c_s_data
  TO <fs_zkf_vir>.
  ASSIGN COMPONENT p_cha_0dateto OF STRUCTURE c_s_data
  TO <fs_0dateto>.
*  ASSIGN COMPONENT p_cha_zvardt OF STRUCTURE c_s_data
*  TO <fs_zvardt>.
  CALL FUNCTION 'ZTST_DIF_DATE'
    EXPORTING
      l_date           =  <fs_0dateto>
   IMPORTING
     DATE_DIFF       = l_val
            .



  <fs_zkf_vir> = l_val.


ENDMETHOD.
```

Please be aware of the following points while writing this code:

A. Define field symbols in the same way as defined below i.e. <FS_KF name> and <FS_Char>

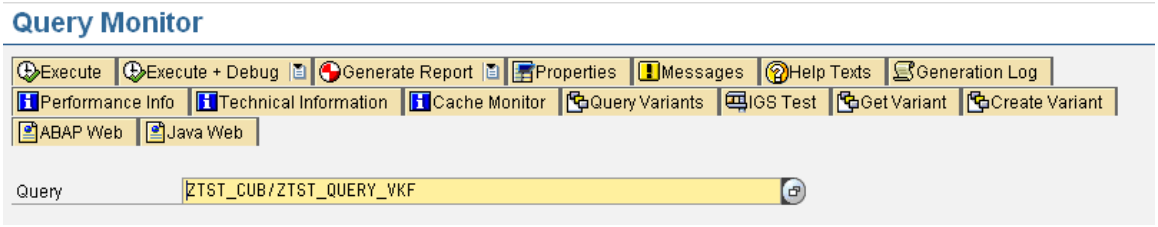
B. C_S_DATA is the table from which you need to read value and assign value of object involved e.g. 0DATE_TO value is assigned into <FS_0DATETO> and ZKF_VIR value assigned to <FS_ZKF_VIR>

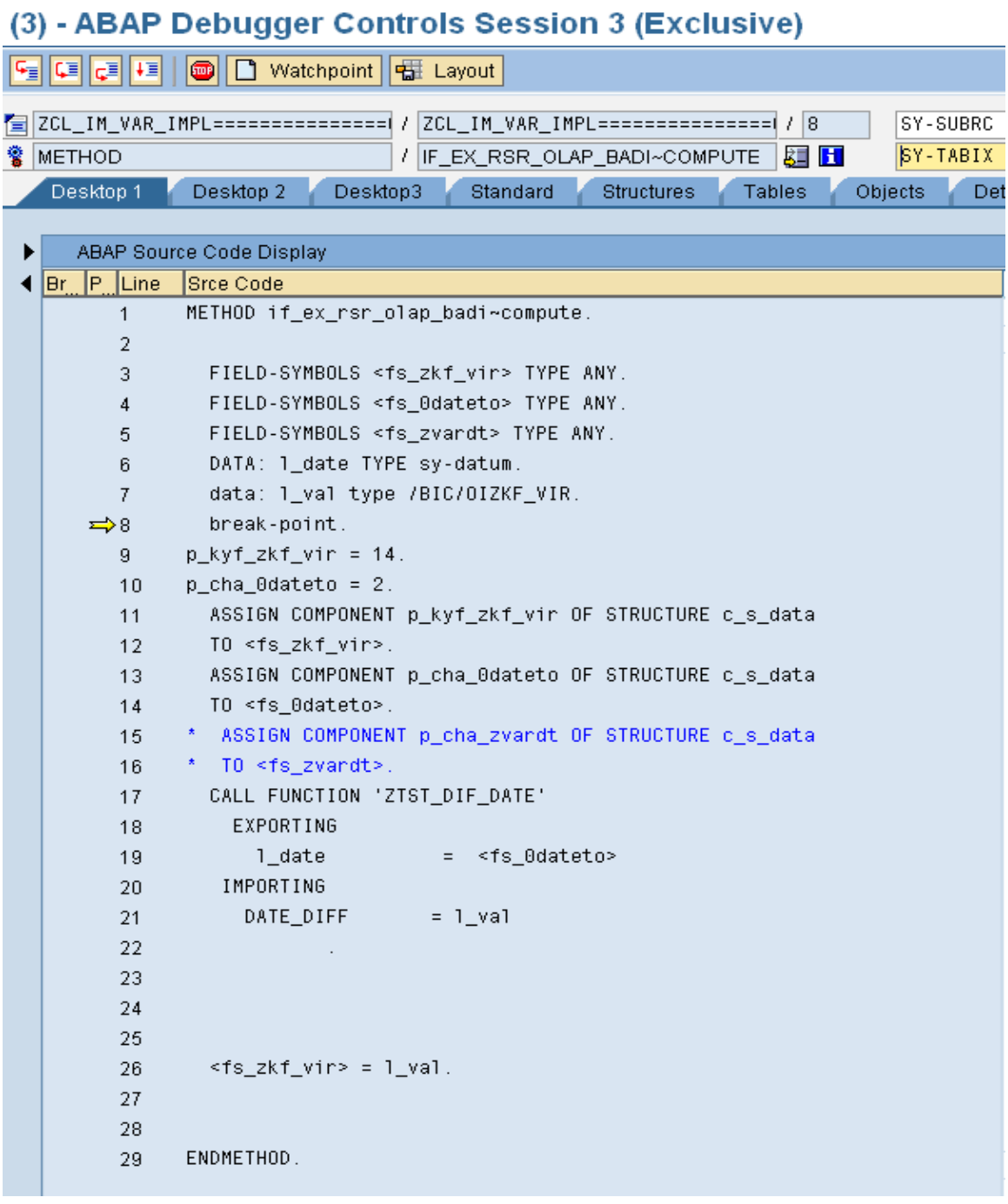
C. Above values are assigned to the Field symbols through components P_KYF_ZKF_VIR and P_CHA_0DATETO of C_S_DATA defined in the attributes tab.

D. But the problem with above assignment, its not returning any value for 0DATE_TO to respective Field symbol and thus VKF was not giving any value.

E. So, to resolve this we identified that the position of the above components is not fixed in the structure C_S_DATA, and we need to find the positions of the objects involved. For this we used debugger by inserting the break-point in the code as shown in Red Box above and identified the position as 14 and 2, by running the query in RSRT. I will try to explain this process in detail as this is main missing information which I didn't find in many of the blogs and white papers submitted.

12. After inserting the break-point in the above code, GOTO RSRT and execute the Report in Debug mode, the query will stop at the break-point just inserted as shown below:

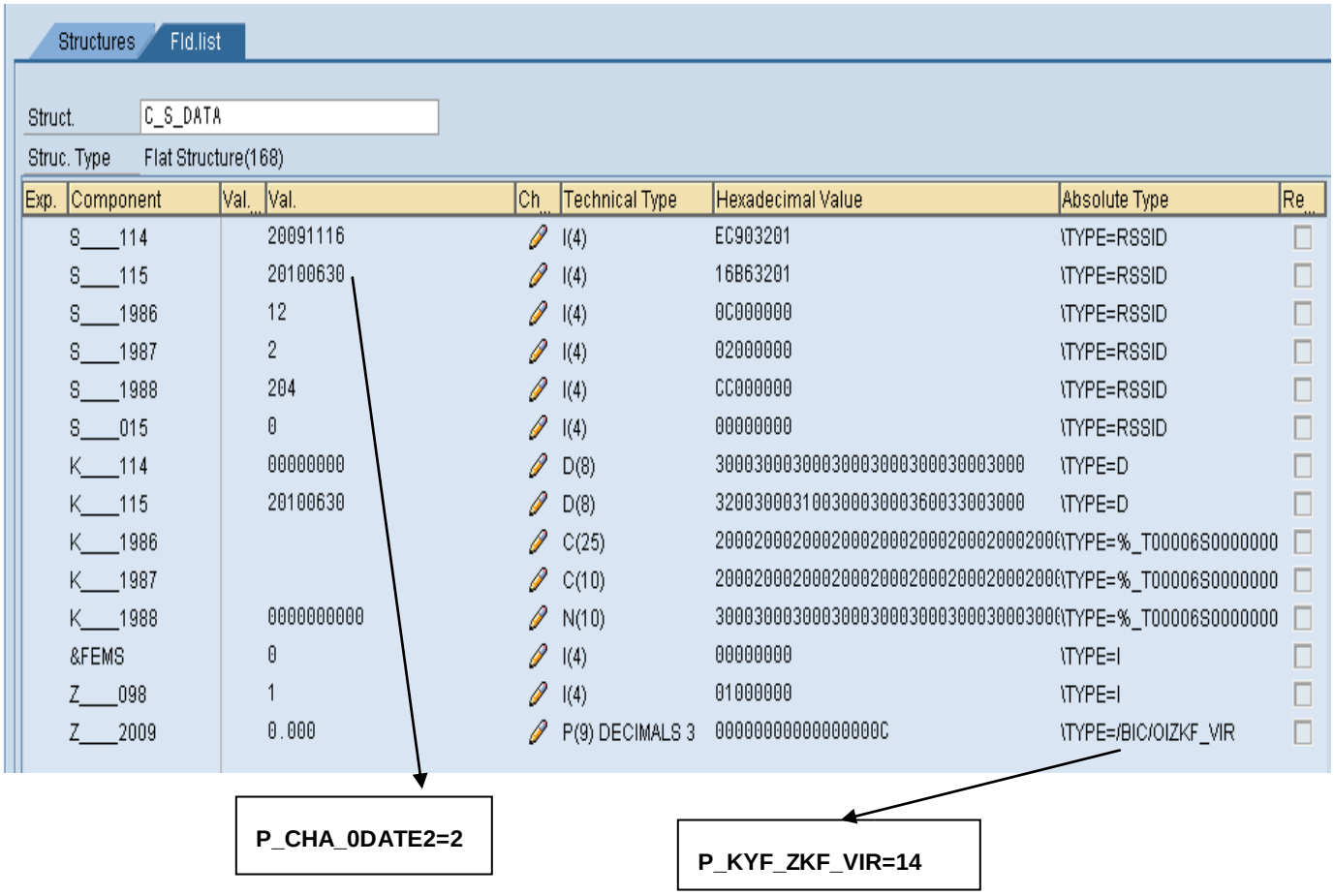




13. Double click on the structure C_S_DATA and following screen appears. Now, In case of characteristic 0DATE_TO I identified the position by seeing its value and for KF by seeing type as shown below.

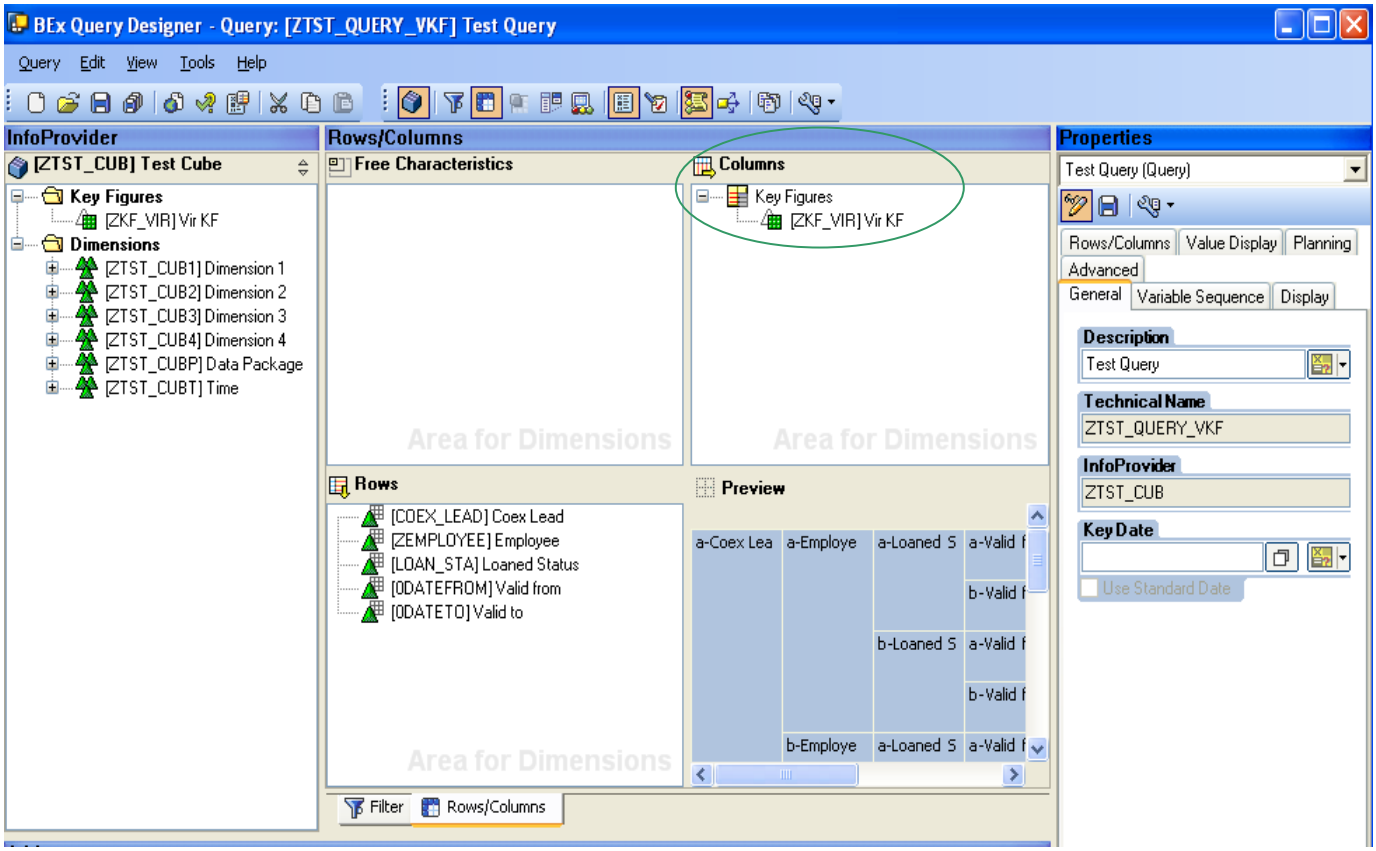
The value of S_115 i.e. 20100630 was same as of 0DATE_TO and it was at position 2 in the structure and Z_2009 has same as of Key Figure ZKF_VIR and it was at position 14 in the structure.

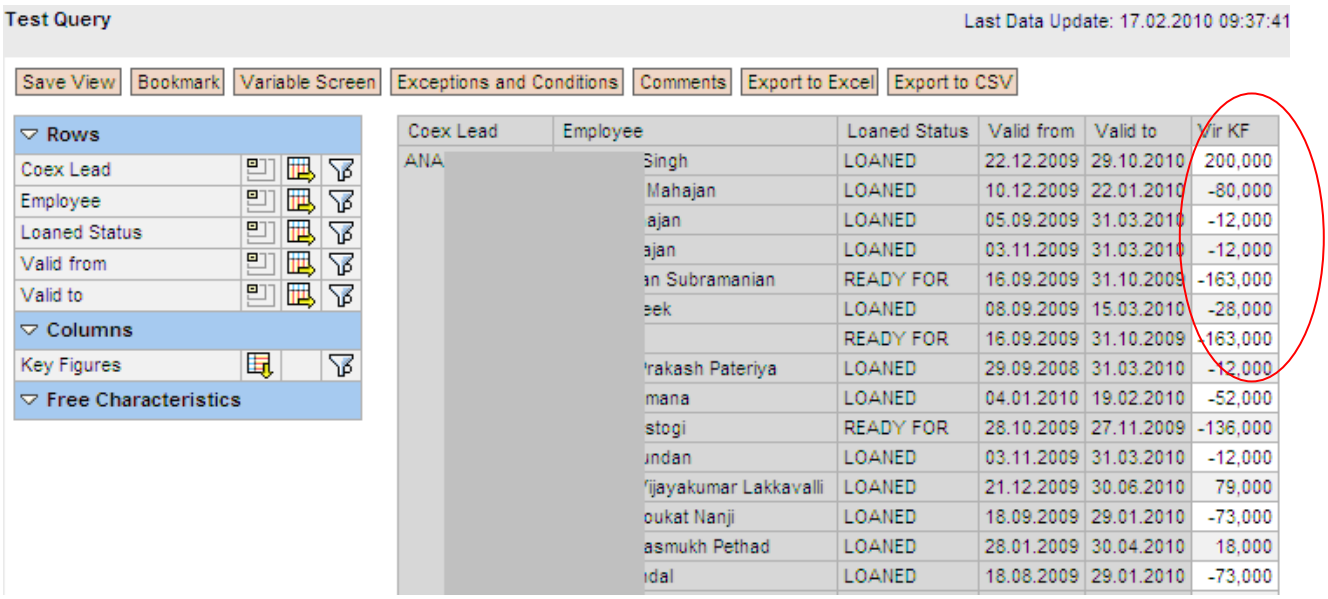
I was not able to identify any other method to get these positions, so I have to follow this manual method.



14. After identifying the positions, I have put this value in code given in Step 11, shown in red box. The same process need to be repeated while implementing Virtual Characteristics and Key Figures in your report as position of involved objects is not fixed.

15. Run the query and Virtual Key figure is calculated as shown below:





Related Content

Virtual Key Figures Characteristics Makes Query More Dynamic

Using the Cache Despite Virtual Characteristics/Key Figures

Step By Step Process for Virtual Key Figures and Characteristics through BAdi

For more information, visit the EDW homepage.

Disclaimer and Liability Notice

This document may discuss sample coding or other information that does not include SAP official interfaces and therefore is not supported by SAP. Changes made based on this information are not supported and can be overwritten during an upgrade.

SAP will not be held liable for any damages caused by using or misusing the information, code or methods suggested in this document, and anyone using these methods does so at his/her own risk.

SAP offers no guarantees and assumes no responsibility or liability of any type with respect to the content of this technical article or code sample, including any liability resulting from incompatibility between the content within this document and the materials and services offered by SAP. You agree that you will not hold, or seek to hold, SAP responsible or liable with respect to the content of this document.