

SAP BW - Copy Query Between Mismatch Info Providers and Queries



Applies to:

SAP BI consultants who are accustomed with SAP ABAP Skills. For more information, visit the [EDW HomePage](#) .

Summary

Method to copy queries between two mismatch infoproviders. Process explains how to copy queries when from one Infoprovider to another when few of the Infoobjects are missing. Same can be applied if you don't want to copy few infoobjects, Calculated Key Figures, Restricted Key Figures from first query to be copied to another.

Author: Suraj Tigga

Company: Capgemini Consulting

Created on: 2 November 2010

Author Bio



Suraj Tigga is a Senior SAP BI / ABAP consultant at Capgemini Consulting, India. Suraj joined Capgemini Consulting in 2008 and has worked on multiple SAP BI implementation and support projects.

Table of Contents

Scenario	3
Step-By-Step Solution.....	3
Standard Error (Using Transaction RSZC).....	3
ABAP Code (<i>ZCOPY_QRY_TO_CUBE</i>).....	5
Copy Function Group (<i>RZIMPACT</i>)	7
Execution (Transaction ZRSZC)	14
Related Content	16
Disclaimer and Liability Notice.....	17

Scenario

Article would be helpful for SAP BW consultants for the below mentioned scenarios:

- Sometimes we have to copy almost similar queries between two infoproviders. For this we generally use transaction RSZC, but the transaction won't allow us to copy the queries because few infoobjects are missing in the other Infoproviders.
- Also using the same ABAP code we can copy the similar queries when we don't want few infoobjects, Calculated Key Figures, restricted Key Figures not to be available in the new query.

Step-By-Step Solution

Solution is explained in three steps:

Standard Error (Using Transaction RSZC): When trying to copy queries between different infoproviders.

ABAP Code (ZCOPY_QRY_TO_CUBE): Meant to trigger Function Module
'ZRSZ_I_COPY_QRY_TO_CUBE_POPUP'

Copy Function Group (RZIMPACT): Copy Function Group RZIMPACT to new 'ZRZIMPACT' and do the required modifications in the relevant Function Modules

Standard Error (Using Transaction RSZC)

In this particular example we have two dissimilar data targets as mentioned below:

DSO1: ZSD_O5 (First Query Built on this Data Target)

DataStore Object	Techn. name / value	Fu...	O...	Data...	L	Key Fi...	C...
Order: Condition Data	ZSD_O5						
Object Information							
Version	In Process						
Save	Saved						
Revised Version	Active Version						
Object Status	Active, executable						
Settings							
Key fields							
Data Fields							
Material	0MATERIAL			CHAR		18	
Sold-to party	0SOLD_TO			CHAR		10	
Customer group	0CUST_GROUP			CHAR		02	
Customer group 1	0CUST_GRP1			CHAR		03	
Customer group 2	0CUST_GRP2			CHAR		03	
Customer group 3	0CUST_GRP3			CHAR		03	
Customer group 4	0CUST_GRP4			CHAR		03	
Customer group 5	0CUST_GRP5			CHAR		03	
Distribution Channel	0DISTR_CHAN			CHAR		02	
Sales Representative	0SALESEMPLY			NUMC		08	
Sales Office	0SALES_OFF			CHAR		04	
Sales group	0SALES_GRP			CHAR		03	
Sales Organization	0SALESORG			CHAR		04	
Material group	0MATL_GROUP			CHAR		09	
Material group 1	0MATL_GRP_1			CHAR		03	
Material group 2	0MATL_GRP_2			CHAR		03	
Material group 3	0MATL_GRP_3			CHAR		03	
Material group 4	0MATL_GRP_4			CHAR		03	
Material group 5	0MATL_GRP_5			CHAR		03	
Product hierarchy	0PROD_HIER			CHAR		18	
Ship-To Party	0SHIP_TO			CHAR		10	
Division	0DIVISION			CHAR		02	
Payer	0PAYER			CHAR		10	
Local currency	0LOC_CURRCY			CUKY		05	

DSO2: YSD_O5 (Copied Query built on this Data Target)

Database Object	Technical Name / Value	PU	CU	Data	Key Part
Order Condition Data					
Object Information					
Version	In Process				
Save	Saved				
Revised Version	Active Version				
Object Status	Active, executable				
Settings					
Key fields					
Sales document	0DOC_NUMBER			CHAR	10
Sales document item	0S_ORD_ITEM			NUMC	06
Fiscal year variant	0FISCVARNT			CHAR	02
Condition type	0KNART			CHAR	04
Condition Counter	0KNCOUNTER			NUMC	02
Data Fields					
Material	0MATERIAL			CHAR	18
Sold-to party	0SOLD_TO			CHAR	10
Customer group	0CUST_GROUP			CHAR	02
Customer group 1	0CUST_GRP1			CHAR	03
Customer group 2	0CUST_GRP2			CHAR	03
Customer group 3	0CUST_GRP3			CHAR	03
Customer group 4	0CUST_GRP4			CHAR	03
Customer group 5	0CUST_GRP5			CHAR	03
Distribution Channel	0DISTR_CHAN			CHAR	02
Sales Representative	0SALESEMPLY			NUMC	08
Sales Office	0SALES_OFF			CHAR	04
Sales group	0SALES_GRP			CHAR	03
Sales Organization	0SALESORG			CHAR	04
Material group	0MATL_GROUP			CHAR	09
Material group 1	0MATL_GRP_1			CHAR	03
Material group 2	0MATL_GRP_2			CHAR	03
Material group 3	0MATL_GRP_3			CHAR	03
Material group 4	0MATL_GRP_4			CHAR	03
Material group 5	0MATL_GRP_5			CHAR	03
Product hierarchy	0PROD_HIER			CHAR	18
Ship-To Party	0SHIP_TO			CHAR	10
Local currency	0LOC_CURRCY			CUKY	05
Company code	0COMP_CODE			CHAR	04
Calendar Day	0CALDAY			DATS	08

Infoobjects 0DIVISION (Division) and 0PAYER (Payer) are absent in DSO YSD_O5 which is present in DSO ZSD_O5.

Step1:

Go to transaction RSZC to copy Query 'ZSD_O5_001' to a new query built on YSD_O5:

Query: ZSD_O5_001

BEx Query Designer - Query: [ZSD_O5_001] Order Conditions

Query Edit View Tools Help

InfoProvider

[ZSD_O5] Order: Condition Data

- Key Figures
 - [0KPRICE] Condition Rate
 - [0KNVAL] Condition Value
 - [0EXCHG_RATE] Exchange rate
 - [1ROWCOUNT] Number of Records
 - [ZVIRTKEY] Virtual Key Figure
- Dimensions
 - [DATA] Data part
 - [KEY] Key Part

Rows/Columns

Free Characteristics

Area for Dimensions

Columns

- Key Figures
 - [0EXCHG_RATE] Exchange rate

Area for Dimension

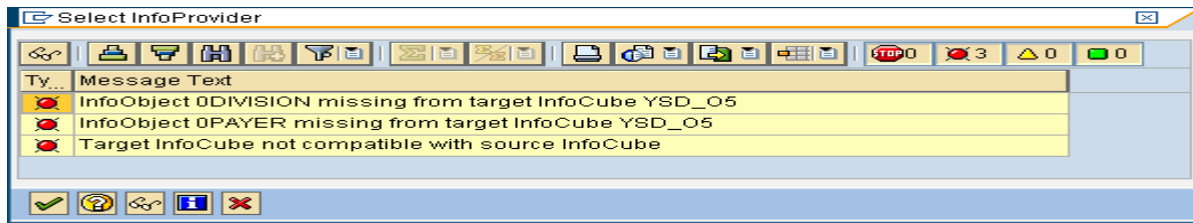
Rows

- [0KNTYP] Condition Category
- [0KNCLASS] Condition Class
- [0KN_USE] Condition Usage
- [0DIVISION] Division
- [0PAYER] Payer
- [0MATERIAL] Material

Preview

a-Condition	a-Condition	a-Condition	a-Divisic
a-Condition	a-Condition	a-Condition	a-Divisic
a-Condition	a-Condition	a-Condition	a-Divisic
a-Condition	a-Condition	a-Condition	a-Divisic

When we try to copy query between DSOs ZSD_05 and YSD_05, then we get the error message as below:



(Since 0DIVISION and 0PAYER are missing in DSO YSD_005).

ABAP Code (ZCOPY_QRY_TO_CUBE)


Step1: Create a program ZCOPY_QRY_TO_CUBE which calls the function module ZRSZ_I_COPY_QRY_TO_CUBE_POPUP to copy the query:

```

Report      ZCOPY_QRY_TO_CUBE      Active
-----
1  *-----*
2  * Report  COPY_QRY_TO_CUBE      *
3  *-----*
4  * - start copy procedure : Copy the Query elements to other query *
5  *   where the few infoobjects are absent in the other infoproviders *
6  *   Also it takes care of the fact is someone doesn't needs some *
7  *   infoobjects in the other copied query *
8  *-----*
9
10 REPORT  zcopy_qry_to_cube      .
11
12 TABLES : rsdiobj.
13
14 DATA : t_iobj TYPE TABLE OF rsioobjnm.
15 * Selection of the Infoobjects not copied to the other query
16 SELECT-OPTIONS : s_iobj FOR rsdiobj-iobjnm.
17
18 IF s_iobj IS NOT INITIAL.
19 |
20 | LOOP AT s_iobj.
21 |   APPEND s_iobj-low TO t_iobj.
22 | ENDLOOP.
23
24 * FM called which copies the query
25 CALL FUNCTION 'ZRSZ_I_COPY_QRY_TO_CUBE_POPUP'
26   TABLES
27     t_iobj = t_iobj.
28
29 ENDIF.

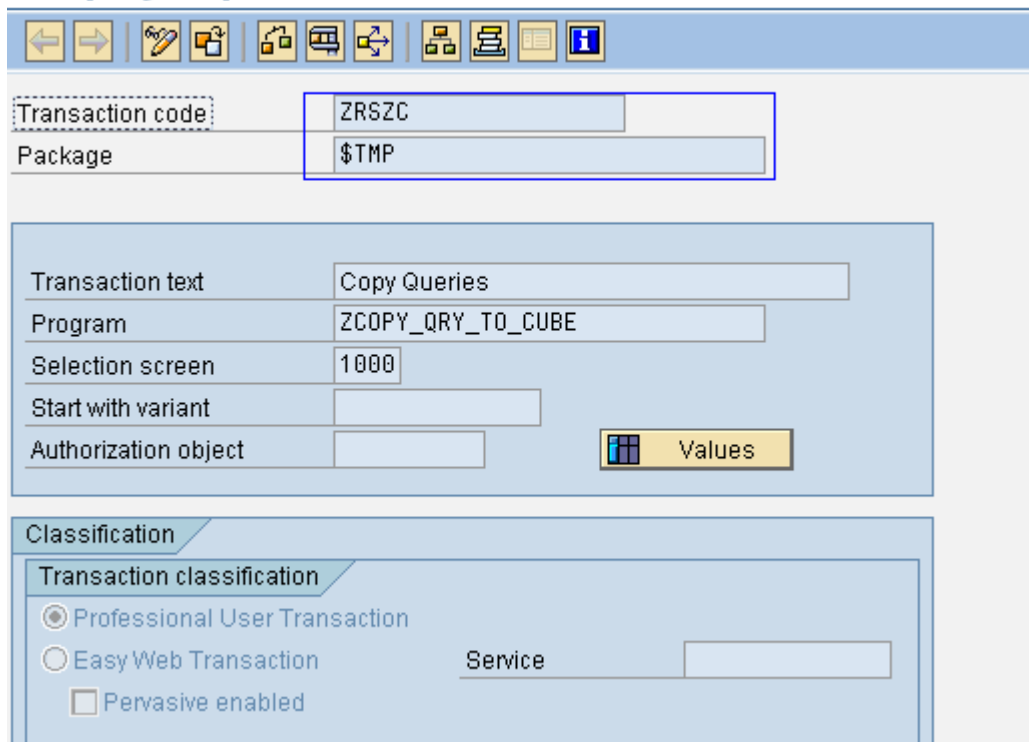
```

Execute the ABAP Code and enter the Infoobjects which are not required to be copied to the other query (Also if those Infoobjects are not available in the DSO YSD_05)

Selection Screen:**Copy Query (Where few infoobjects are absent Other Infoproviders)**


The screenshot shows the selection screen for the 'Copy Query' transaction. It features a blue header bar with a green checkmark icon. Below the header, there is a text input field labeled 'Infoobject(Not required)' followed by a yellow search button with a magnifying glass icon and a right-pointing arrow button.

Note: Internal table t_jobj contains the list of infoobjects which are entered in selection screen)

Step2:Create transaction Code for the above ABAP Code:**Display Report Transaction**


The screenshot displays the 'Display Report Transaction' screen in SAP. It includes a toolbar with various icons for navigation and actions. The main content area is divided into several sections:

- Transaction code:** ZRSZC
- Package:** \$TMP
- Transaction text:** Copy Queries
- Program:** ZCOPY_QRY_TO_CUBE
- Selection screen:** 1000
- Start with variant:** (empty field)
- Authorization object:** (empty field) with a 'Values' button.
- Classification:**
 - Transaction classification:**
 - Professional User Transaction
 - Easy Web Transaction
 - Pervasive enabled
 - Service:** (empty field)

Copy Function Group (RZIMPACT)

Step1: Copy the Function Group RZIMPACT to new 'ZRZIMPACT':

Copy Function Group: RZIMPACT

Old function group: RZIMPACT

New function group: ZRZIMPACT

Person resp.: DEVELOPER

Function group text: Impact Analysis

Copy Cancel

Copy the Subsequent Function Modules to new 'Z' Function Modules:

Standard Function Module	Z-Function Module(Copied)
RSZ_I_BASIC_CHA_WHERE_USED	ZRSZ_I_BASIC_CHA_WHERE_USED
RSZ_I_BASIC_KEYFIG_WHERE_USED	ZRSZ_I_BASIC_KEYFIG_WHERE_USED
RSZ_I_COPY_QRY_TO_CUBE	ZRSZ_I_COPY_QRY_TO_CUBE
RSZ_I_COPY_QRY_TO_CUBE_POPUP	ZRSZ_I_COPY_QRY_TO_CUBE_POPUP
RSZ_I_COPY_QRY_TO_CUBE_SINGLE	ZRSZ_I_COPY_QRY_TO_CUBE_SINGLE

Step2: Declare a global internal table *it_obj* which holds the Infoobjects to be filtered (ABAP Code Selection Screen input):

```

Include LZRZIMPACTTOP Active
17 data:
18   g_target_cubetype type rsd_cubetype.
19
20   * Begin of change(Suraj)
21   DATA : it_obj TYPE TABLE OF zioobj.
22   * End of change(Suraj)
23
24   * Copy screen
25   selection-screen begin of screen 100 as window title g_title.
26
27   SELECTION-SCREEN BEGIN OF BLOCK ip WITH FRAME TITLE text-001.
28   PARAMETERS   g_p_ic1   LIKE v_compdic_compic-infocube OBLI
29   PARAMETERS   g_p_ic2   LIKE v_compdic_compic-infocube OBLI
30   SELECTION-SCREEN END OF BLOCK ip.

```

Step3: Declare Table *T_IOBJ* for the below mentioned Function Modules (Meant for holding the ABAP code selection screen details):

Function module: ZRSZ_I_COPY_QRY_TO_CUBE_POPUP Active

Attributes Import Export Changing Tables Exceptions Source code

Parameter Name	Typing	Associated Type	Optional	Short text
T_IOBJ	LIKE	ZIOBJ	<input type="checkbox"/>	Infoobject
			<input type="checkbox"/>	

Function module: ZRSZ_I_COPY_QRY_TO_CUBE_SINGLE Active

Attributes Import Export Changing Tables Exceptions Source code

Parameter Name	Typing	Associated Type	Optional	Short text
T_IOBJ	LIKE	ZIOBJ	<input type="checkbox"/>	Infoobject
			<input type="checkbox"/>	

Function module: ZRSZ_I_COPY_QRY_TO_CUBE Active

Attributes Import Export Changing Tables Exceptions Source code

Parameter Name	Typing	Associated Type	Optional	Short text
T_IOBJ	LIKE	ZIOBJ	<input type="checkbox"/>	Infoobject
			<input type="checkbox"/>	

Step4: Pass the internal table *t_iobj* to the Function Module *ZRSZ_I_COPY_QRY_TO_CUBE*:

Function module: ZRSZ_I_COPY_QRY_TO_CUBE_POPUP Active

Attributes Import Export Changing Tables Exceptions Source code

```

107 ELSE.
108 * Target InfoProvider is not available
109 MESSAGE A001 WITH G_P_IC2.
110 EXIT.
111 ENDIF.
112
113 *Everything is OK. Start copy procedure.
114 IF SY-SUBRC = 0.
115 * Begin of Change(Suraj)
116 CALL FUNCTION 'ZRSZ_I_COPY_QRY_TO_CUBE'
117 EXPORTING
118   i_source_infocube      = G_P_IC1
119   i_target_infocube     = G_P_IC2
120   I_DEFTP               = L_DEFTP
121   tables
122     t_iobj              = t_IOBJ
123   EXCEPTIONS
124     SELECTION_CANCELED  = 1
125     INHERITED_ERROR     = 2
126     OTHERS              = 3
127
128 * End of Change(Suraj)
129 ENDIF.
130
131
132 ENDFUNCTION.

```


Step5: Store the Selection Screen (ABAP Code) Infoobject details to globally declare internal table *it_iobj*.

Function module **ZRSZ_I_COPY_QRY_TO_CUBE** Active

Attributes Import Export Changing Tables Exceptions Source code

```

32 * Begin of Change (Suraj)
33   it_iobj[] = t_iobj[].
34 * End of Change(Suraj)

```

Step6: Delete the (ABAP Code Selection Screen) entries from internal table *l_t_source_cob_pro* which contains all the required details of the Source Query:

Include **LZRZIMPACTF01** Active

```

158 * target infocube
159   l_th_target_cob_pro = l_t_target_cob_pro.
160
161 * Begin of Change (Suraj)
162   LOOP AT l_t_source_cob_pro INTO wl_rsd_s_cob_pro.
163     READ TABLE it_iobj INTO wl_iobj
164       WITH KEY zicobj = wl_rsd_s_cob_pro-iobjnm.
165     IF sy-subrc EQ 0.
166       wl_iobjnm = wl_rsd_s_cob_pro-iobjnm.
167       DELETE l_t_source_cob_pro WHERE iobjnm = wl_iobjnm.
168     ENDIF.
169   ENDLOOP.
170 * End of Change (Suraj)
171
172   LOOP AT l_t_source_cob_pro
173     ASSIGNING <l_s_source_cob_pro>
174     WHERE chasel <> rsd_c_chasel-not_allowed

```

Step7: Function Module `RSZ_X_COMPONENT_GET` returns the source query data into respective internal tables. Delete all the respective entries as per the earlier filtered Infoobject details as stored in internal table `t_obj`:

```

Include      LZRZIMPACTF01      Active
289      CALL FUNCTION 'RSZ_X_COMPONENT_GET'
290      EXPORTING
291          i_compuid          = i_source_compuid
292          i_template_type   = i_defftp
293          i_language        = '*'
294          i_use_buffer      = 'A'
295          i_designtime_call = rs_c_false
296      IMPORTING
297          e_subrc           = c_subrc
298      TABLES
299          c_t_eltdir       = l_t_eltdir
300          c_t_eltprop      = l_t_eltprop
301          c_t_eltprio      = l_t_eltprio
302          c_t_eltattr      = l_t_eltattr
303          c_t_eltxref      = l_t_eltxref
304          c_t_compdire     = l_t_compdire
305          c_t_compic       = l_t_compic
306          c_t_select       = l_t_select
307          c_t_range        = l_t_range
308          c_t_calc         = l_t_calc
309          c_t_elttxt       = l_t_elttxt
310          c_t_cell         = l_t_cell.
311      IF c_subrc <> 0.
312          *   get and convert messages from msg server
313          PERFORM get_messages
314              CHANGING
315                  c_t_msg.
316          EXIT.

```

ABAP Code to delete the required entries (relevant to Infoobjects entered in Selection Screen):

Include	LZRZIMPACTF01	Active
319	* Begin of Change (Suraj)	
320	IF l_t_eltdir IS NOT INITIAL.	
321	SELECT seltuid	
322	teltuid	
323	FROM rszeltxref	
324	INTO TABLE t1_rszeltxref1	
325	WHERE seltuid EQ i_source_compuid	
326	AND objvers EQ 'A'.	
327	IF sy-subrc EQ 0.	
328	* Keep fetching records from table rszeltxref	
329	WHILE w_flag NE 'X'.	
330	SELECT seltuid	
331	teltuid	
332	FROM rszeltxref	
333	INTO TABLE t1_rszeltxref2	
334	FOR ALL ENTRIES IN t1_rszeltxref1	
335	WHERE seltuid EQ t1_rszeltxref1-teltuid	
336	AND objvers EQ 'A'.	
337	IF sy-subrc EQ 0.	
338	REFRESH t1_rszeltxref1.	
339	LOOP AT t1_rszeltxref2 INTO w1_rszeltxref2.	
340	MOVE-CORRESPONDING w1_rszeltxref2 TO wa_rszeltxref.	
341	MOVE-CORRESPONDING w1_rszeltxref2 TO w1_rszeltxref1.	
342	APPEND wa_rszeltxref TO t_rszeltxref.	
343	APPEND w1_rszeltxref1 TO t1_rszeltxref1.	
344	CLEAR: w1_rszeltxref1 , wa_rszeltxref ,w1_rszeltxref2.	
345	ENDLOOP.	
346	REFRESH t1_rszeltxref2.	
347	ELSE.	
348	w_flag = 'X'.	
349	ENDIF.	
350	ENDWHILE.	
351	ENDIF.	
352		
353	DELETE ADJACENT DUPLICATES FROM t_rszeltxref COMPARING ALL FIELDS.	
354		

```

* Begin of Change (Suraj)
IF l_t_eltdir IS NOT INITIAL.
    SELECT seltuid
        teltuid
    FROM rszeltxref
    INTO TABLE t1_rszeltxref1
    WHERE seltuid EQ i_source_compuid
    AND objvers EQ 'A'.
    IF sy-subrc EQ 0.
* Keep fetching records from table rszeltxref
    WHILE w_flag NE 'X'.
        SELECT seltuid
            teltuid
        FROM rszeltxref
        INTO TABLE t1_rszeltxref2
        FOR ALL ENTRIES IN t1_rszeltxref1
        WHERE seltuid EQ t1_rszeltxref1-teltuid
        AND objvers EQ 'A'.
        IF sy-subrc EQ 0.

```

```

REFRESH t1_rszeltxref1.
LOOP AT t1_rszeltxref2 INTO w1_rszeltxref2.
  MOVE-CORRESPONDING w1_rszeltxref2 TO wa_rszeltxref.
  MOVE-CORRESPONDING w1_rszeltxref2 TO w1_rszeltxref1.
  APPEND wa_rszeltxref TO t_rszeltxref.
  APPEND w1_rszeltxref1 TO t1_rszeltxref1.
  CLEAR: w1_rszeltxref1 , wa_rszeltxref ,w1_rszeltxref2.
ENDLOOP.
REFRESH t1_rszeltxref2.
ELSE.
  w_flag = 'X'.
ENDIF.
ENDWHILE.
ENDIF.

```

DELETE ADJACENT DUPLICATES FROM t_rszeltxref COMPARING ALL FIELDS.

- * Retrieve records from table RSZELTDIR

```

SELECT eltuid
      defaultthint
FROM rszeltdir
INTO TABLE t_infoobject1
FOR ALL ENTRIES IN t_rszeltxref
WHERE eltuid EQ t_rszeltxref-teltuid.
IF sy-subrc EQ 0.
  LOOP AT t_infoobject1 INTO wa_infoobject1.
    READ TABLE it_iobj INTO w1_iobj
    WITH KEY ziobj = wa_infoobject1-defaultthint.
    IF sy-subrc EQ 0.
      APPEND wa_infoobject1 TO t_infoobject.
    ENDIF.
  CLEAR wa_infoobject1.
ENDLOOP.
ENDIF.

```
- * Delete record from l_t_eltdir

```

LOOP AT l_t_eltdir INTO w1_eltdir.
  READ TABLE t_infoobject INTO wa_infoobject WITH KEY
      eltuid = w1_eltdir-eltuid.

  IF sy-subrc EQ 0.
    w_eltuid = w1_eltdir-eltuid.
    DELETE l_t_eltdir WHERE eltuid = w_eltuid.
  ENDIF.
  CLEAR w_eltuid .
ENDLOOP.

```
- * Delete records from l_t_eltprop

```

LOOP AT l_t_eltprop INTO w1_eltprop.
  READ TABLE t_infoobject INTO wa_infoobject WITH KEY
      eltuid = w1_eltprop-eltuid.

  IF sy-subrc EQ 0.
    w_eltuid = w1_eltprop-eltuid.
    DELETE l_t_eltprop WHERE eltuid = w_eltuid.
  ENDIF.
  CLEAR w_eltuid.
ENDLOOP.

```

```

* Delete records from l_t_eltxref
LOOP AT l_t_eltxref INTO w1_eltxref.
  READ TABLE t_infoobject INTO wa_infoobject WITH KEY
                                eltuid = w1_eltxref-teltuid.
  IF sy-subrc EQ 0.
    w_eltuid = w1_eltxref-teltuid.
    DELETE l_t_eltxref WHERE eltuid = w_eltuid.
  ENDIF.
  CLEAR w_eltuid.
ENDLOOP.

* Delete records from l_t_select
LOOP AT l_t_select INTO w1_select.
  READ TABLE t_infoobject INTO wa_infoobject WITH KEY
                                eltuid = w1_select-eltuid.
  IF sy-subrc EQ 0.
    w_eltuid = w1_select-eltuid.
    DELETE l_t_select WHERE eltuid = w_eltuid.
  ENDIF.
  CLEAR w_eltuid.
ENDLOOP.

* delete records from l_t_range.
LOOP AT l_t_range INTO w1_range.
  READ TABLE t_infoobject INTO wa_infoobject WITH KEY
                                eltuid = w1_select-eltuid.
  IF sy-subrc EQ 0.
    w_eltuid = w1_select-eltuid.
    DELETE l_t_select WHERE eltuid = w_eltuid.
  ENDIF.
  CLEAR w_eltuid.
ENDLOOP.

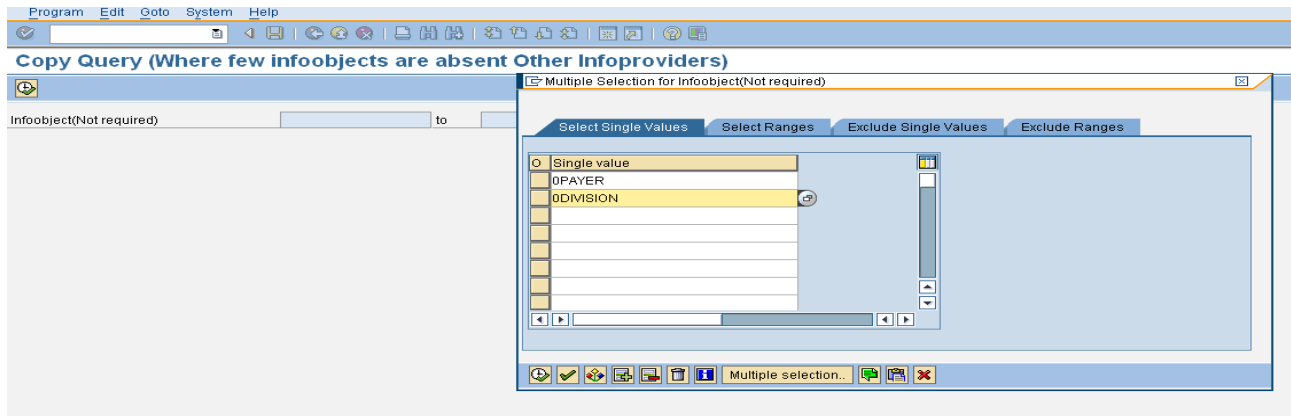
ENDIF.
* End of Change (Suraj)

```

Note: Read the table (DB) RSZELTDIR, RSZELTXREF to get the required details from the source query and compared with the internal table values returned from the Function Module RSZ_X_COMPONENT_GET and delete the relevant records from these internal tables with respect to the infoobjects entered in the ABAP code selection screen.

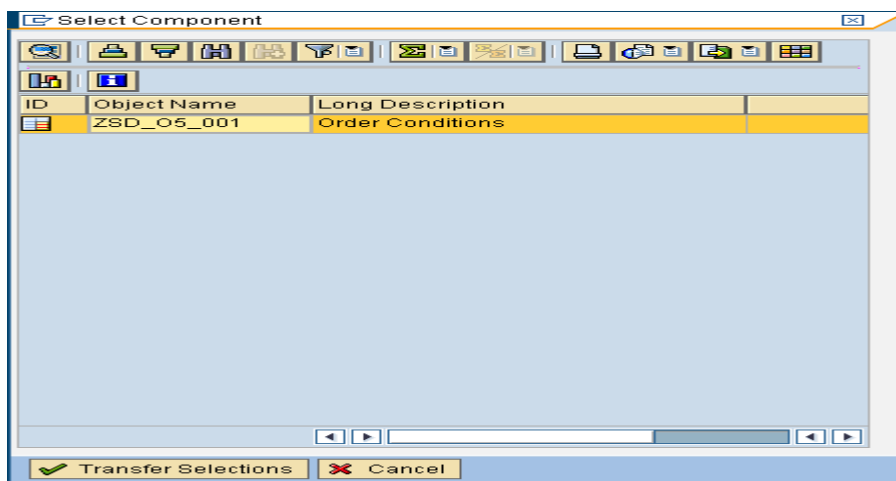
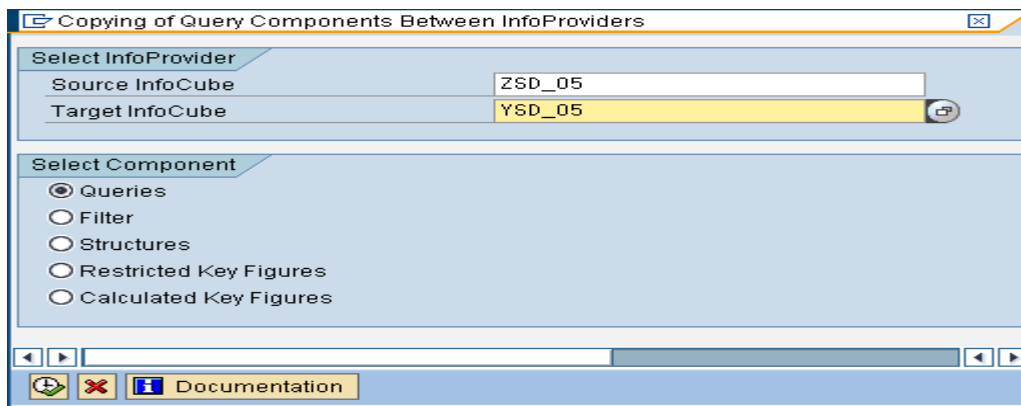
Execution (Transaction ZRSZC)

Step1: Go to SE38, execute the code `ZCOPY_QRY_TO_CUBE` and enter the Infoobject values not required in the copied query:

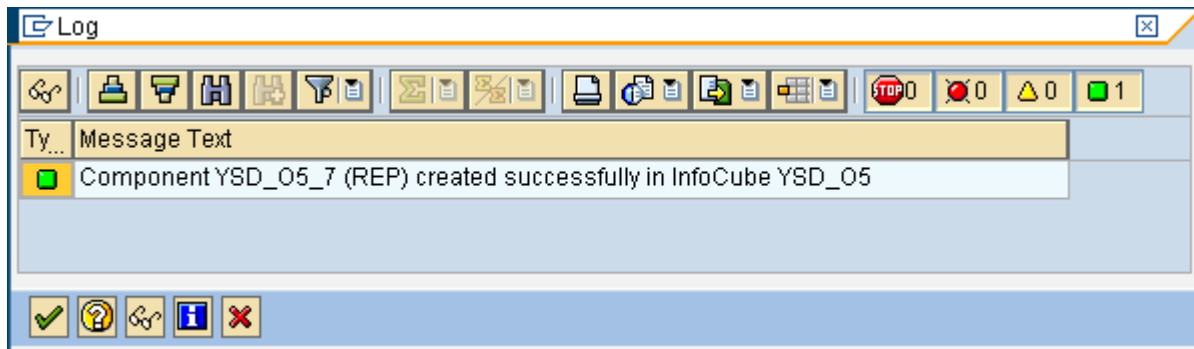
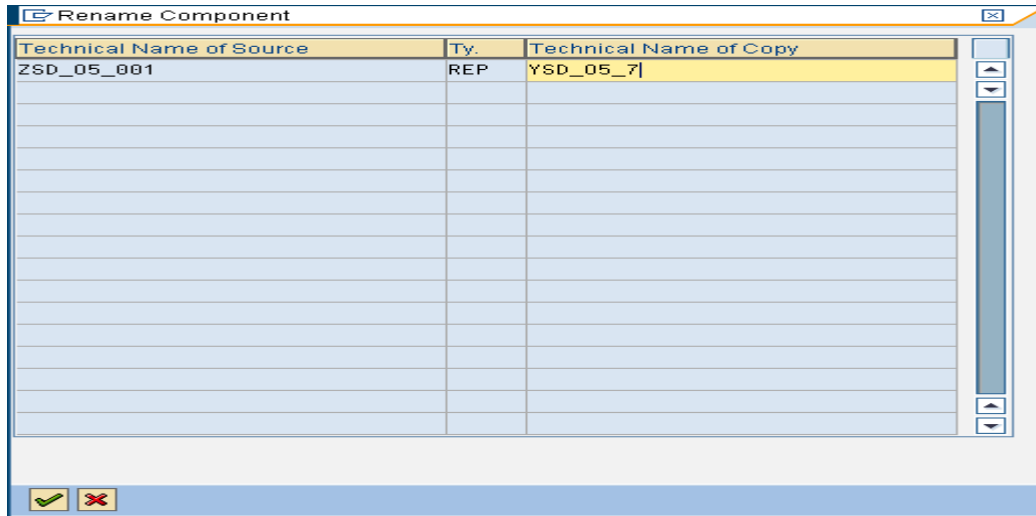


It triggers the transaction ZRSZC.

Step2: Enter Source and Target DSO names:

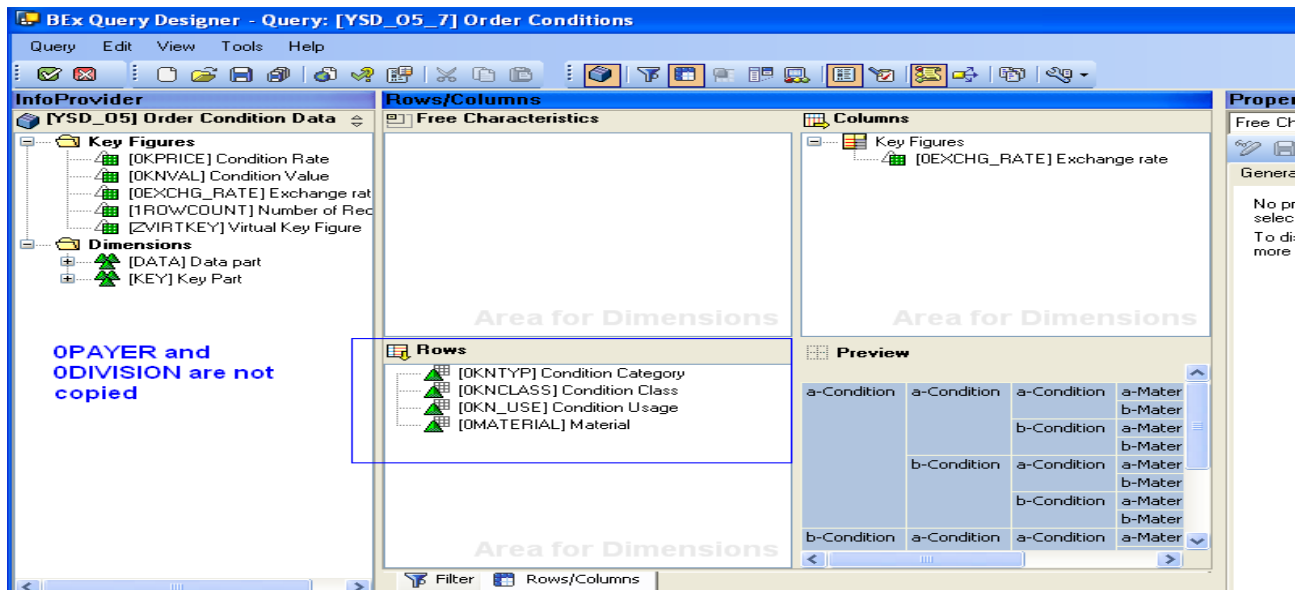


Step3: Enter the new Query Name ‘



Successfully the Query is copied.

Step4: Verify the query – 0PAYER and 0DIVISION are absent in the new query copied:



Related Content

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.