Carine Tchoutouo Djomo

March 30, 2015 2 minute read

Handling of SELECT-OPTIONS parameters within AMDP

The ABAP Managed Database Procedures (AMDP) framework provides the higher level of integration of the advanced HANA capabilities into ABAP applications. It allows creating and managing SQLScript-based DB procedures from the ABAP platform by using so called AMDP methods.

One difficulty faced by developers when working with AMDPs is the handling of `SELECT-OPTIONS` parameters (selection tables or range tables). The present blog will exactly tackle that topic.

Simply explained, the handling of `SELECT-OPTIONS` parameters in the context of AMDPs requires two steps:

1. Conversion of the selection tables into an SQL `WHERE` clause using method `CL_SHDB_SELTAB=>COMBINE_SELTABS( )`
2. Handling of dynamic WHERE clauses within the AMDP method using the function APPLY_FILTER

Step 1: Conversion of SELECT-OPTIONS parameters into an SQL WHERE clause

For those of you who have been generating dynamic `WHERE` clauses till now using the class **CL_LIB_SELTAB**: Do no longer use it again and if possible even replace such calls (as explained below) in your existing code!

The new class **CL_SHDB_SELTAB** – especially its static method **COMBINE_SELTABS( )**– shall be used for this purpose instead. It provides a comfortable coverage of the conversion functionality for SAP HANA (refer to SAP Note 2124672 – *SMP login required*). This conversion routine includes checks for SQL injections during the conversion of the selection tables into an SQL `WHERE` clause. ABAP 7.4 SP08 and higher is required in order to apply the above mentioned SAP Note.

Here is a simple demo report showing how to convert the SELECT-OPTIONS parameters into a `WHERE` clause:

```
REPORT zr_display_result.
DATA ls_bpa TYPE snwd_bpa.
* selection criteria
SELECT-OPTIONS: bp_id    FOR ls_bpa-bp_id,
                cur_code FOR ls_bpa-currency_code.

* convert the selection tables into a WHERE clause
DATA(lv_where) = cl_shdb_seltab=>combine_seltabs(
   it_named_seltabs  = VALUE #(
       ( name = 'BP_ID'          dref = REF #(  bp_id[] ) )
       ( name = 'CURRENCY_CODE'  dref = REF #(  cur_code[]  ) ) )
   iv_client_field   = 'CLIENT' ).

* call the AMDP method with the dynamic WHERE clause
zcl_amdp_select_options=>execute(
  EXPORTING
   iv_where_cond =  lv_where
  IMPORTING
   result_1   = DATA(lt_itab1)
   result_2   = DATA(lt_itab2) ).
* further processing... E.g. display
cl_demo_output=>display_data( value = lt_itab2 ).
```

As shown above, you just have to pass an internal table (*defined here using the new value operator `VALUE`*) filled with as many `SELECT-OPTIONS` parameters as required by your scenario. The name of the relevant field (`NAME`) and of the data reference to the corresponding `SELECT-OPTIONS`

table (DREF) is required for each entry. In case of relevance, it is recommended to specify the exporting parameter IV_CLIENT_FIELD with 'CLIENT' or 'MANDT' (depending on the related table field name) to ensure the addition of the client filter to the WHERE clause.

The method returns the dynamic WHERE condition as a string which can then be passed to the AMDP method. Let's now go to the next step.

Step 2: Handling of dynamic WHERE clauses within the AMDP method

What needs to be done is very simple: The SQLScript statement APPLY_FILTER is used to apply the selection criteria to the relevant dataset which can be a database table/view, a HANA view (except Analytical view) or an intermediate table variable.

Below you can see a code simple showing how to apply the dynamic WHERE clause in both cases; directly on a data source (table or view) [CASE 1] or on an intermediate dataset (table variable) [CASE 2].

```
CLASS zcl_amdp_select_options IMPLEMENTATION .
  METHOD execute BY DATABASE PROCEDURE FOR HDB LANGUAGE SQLSCRIPT
                 OPTIONS READ-ONLY USING snwd_bpa snwd_so.
*   CASE 1: apply filter options directly on dataset (view or table)
    result_1 = APPLY_FILTER(snwd_bpa, :iv_where_cond);
*   fill intermediate table variable
    itab  = SELECT bpa.client, bpa.bp_id, bpa.company_name, bpa.currency_code,
                   so.so_id, so.billing_status
              FROM snwd_bpa AS bpa
              LEFT OUTER JOIN snwd_so AS so
                ON so.buyer_guid = bpa.node_key
                AND so.client = bpa.client;
*   CASE 2: apply filter options on intermediate table variable
    result_2 = APPLY_FILTER(:itab, :iv_where_cond) ;
  ENDMETHOD.
ENDCLASS.
```

The APPLY_FILTER function expects two parameters. The first one is the dataset to which you want to apply the filter and the second one is the generated WHERE clause which is passed as a string argument. Find more information about the APPLY_FILTER function in the SAP HANA SQLScript reference.

Summary:

1. Static method `COMBINE_SELTABS( )` of the new class `CL_SHDB_SELTAB` shall be used for the conversion of `SELECT-OPTIONS` parameters (selection tables or range tables) into an SQL `WHERE` clause when running on HANA DB.
   - The optional parameter `IV_CLIENT` should be specified with `'CLIENT'` or `'MANDT'`) when applicable
   - This class implementation is provided for HDB (refer to SAP Note 2124672 – *SMP login required*)
   - ABAP 7.4 SP08 and higher is required in order to apply the above SAP Note
2. The class `CL_LIB_SELTAB` and its methods are obsolete
3. Use the SQLScript function `APPLY_FILTER` to apply the selection criteria to the selected data in the AMDP
   - The function can be applied on database tables/views, HANA views (except Analytical views) or table variables

That's it... bye!

Alert Moderator

Assigned tags

ABAP Development |

Related Blog Posts

How to extend an ABAP Managed Database Procedures using AMDP BADI's
By **Former Member** , Dec 26, 2016

Process complex rule tables direct in AMDP
By **Sean Holland** , Feb 02, 2018

AMDP Statement performance trace

By **Ankit Rastogi** , Feb 13, 2019

Related Questions

Modify data of a temp table in AMDP

By **Former Member** , May 13, 2016

AMDP Error Handling

By **Former Member** , May 26, 2017

AMDP database extract returning dubious records.

By **Diganto Goswami** , Jul 15, 2017

45 Comments

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

Amol Samte

March 30, 2015 at 11:56 am

Hello Carine,

Very nice blog.

1. Does apply filter may affect performance? Because we are fetching data and then we are applying filter to final data set.

2. How to handle cross client at the time of transporting in AMDP.

– Regards,

Amol

Like (0)

Thomas Gauweiler

March 30, 2015 at 12:10 pm

Hi Amol,

1) if possible apply filters to the table directly as shown with result_1.

If you need a to use a table variable like with result_2 I would apply all statically known filters directly with the first select. (e.g. filters on the client field).

2) you can omit the iv_client parameter and add a a condition with the client field on your own to the WHERE condition.

Best Regards, Thomas

Like (0)

Amol Samte

March 30, 2015 at 12:18 pm

Hi Thomas,

As of now we are doing in same pattern like writing if else code for MANDT. But it will be good if client handling dependency can be handled automatically.

Thanks,

Amol

Like (0)

Thomas Gauweiler

March 30, 2015 at 12:37 pm

Hi Amol,

I do not understand your requirement. In a scenario where I only use the current client, then I can use the iv_client parameter as shown (I have to pass the field name as the class cannot know on which table you want to do a SELECT).

In any other scenario I will anyway have to specify the needed client value explicitly.

What more automatism do you want?

Best Regards, Thomas

Like (0)

Amol Samte

March 30, 2015 at 1:04 pm

Hi Thomas,

Thanks for your reply.

Actually I have to fetch data from different schema.

e.g. In development we have schema RD2 and Production environment we have RP2.

In below code I have to put schema name RD2 for development server while transporting to production system I am changing to RP2.

So I am finding how to handle schema while transporting.

```
                FOR HDB
                LANGUAGE SQLSCRIPT
                OPTIONS READ-ONLY.
  EX_TSPAT = SELECT SPART, VTEXT FROM RD2.TSPAT
                                 WHERE SPRAS = 'E'
                                 ORDER BY SPART;

  EX_FINAL = APPLY_FILTER(:EX_TSPAT, :IP_WHERE);
```

Regards,

Amol

Like (0)

Thomas Gauweiler

March 30, 2015 at 1:35 pm

Hi Amol,

that is a completely different problem. You need to switch between different tables. The answer from a DB perspective would by synonyms. They could be used to point to the same table in different schemata.

But it is very difficult to use synonyms with AMDP. You have to maintain them on your own and you would have to ensure, that the synonym exists in each system of your landscape before the AMDP gets transported to it. Otherwise you risk syntax errors during import.

And you cannot use tables of your own ABAP schema, as these would to be declared in the USING clause.

As this is independent of APPLY_FILTERS and SELECT-OPTIONS we should stop the discussion within this topic here.

Best Regards, Thomas

Like (0)

Amol Samte

March 30, 2015 at 1:41 pm

Thanks Thomas.

-Amol

Like (0)

Sreehari V Pillai

March 30, 2015 at 12:08 pm

Thanks for the nice blog 🙂

Amol Samte : If you see the CASE 1, we are applying it directly on the table or view. But the second case, for sure it will degrade the performance. I used to handle the 'SELECT-OPTION' situation with alternate solution. I will issue one select 'COLUMN' from the tableusing the range variable from abap stack first and get the list of possible values. Then I pass the data to AMDP and then process it. You can pass the sy-mandt value as importing parameter to the AMDP method.

Sree

Like (0)

Former Member

April 5, 2015 at 4:10 pm

How to we handle below scenario ?

We have 2 tables to JOIN and also need to apply FILTER where they are a few fields on table 1 and rest on table 2 i.e filter has fields applicable for both the tables.

Appreciate if you give an example code.

Thanks in advance

Like (0)

Former Member

April 5, 2015 at 4:28 pm

Is it OK or not OK to use CE functions inside AMDP methods instead of regular SQL statements? Would it provide better performance if we use CE functions....How do we use Filter when we want to use CE Functions...

Like (0)

Jasmin Gruschke

April 7, 2015 at 7:57 am

Hi,
this is rather a query you'd like to put in the HANA development community (SAP HANA Developer Center). Or have a look at the HANA SQLScript reference (http://help.sap.com/hana/sap_hana_sql_script_reference_en.pdf) which gives more information about the Do's and Dont's, see in particular chapter 9 "Best Practices for Using SQLScript".
Cheers,

Jasmin

Like (0)

Former Member

April 5, 2015 at 7:10 pm

Excellent blog explaining the new features.

Like (0)

Amol Samte

April 15, 2015 at 12:40 pm

Hi,

We are on 7.4 SP 09 and class CL_SHDB_SELTAB is not available. Any suggestions.

-Amol

Like (0)

Jasmin Gruschke

April 15, 2015 at 12:50 pm

Hi Amol,
please check SAP note 2124672.
Cheers,
  Jasmin

Like (0)

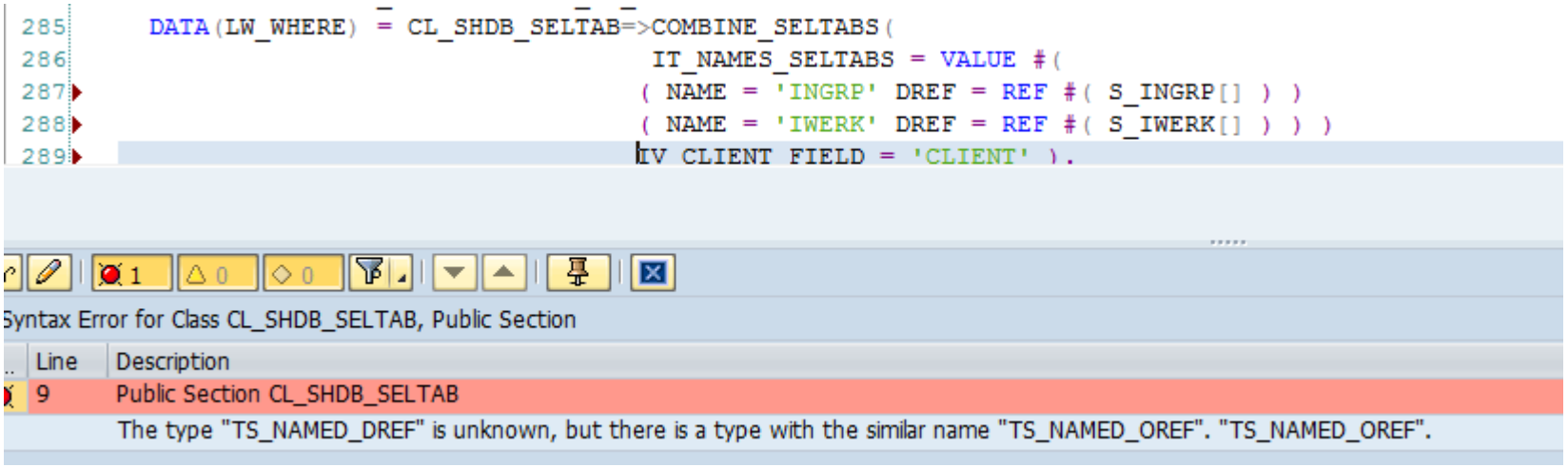
Amol Samte

April 16, 2015 at 9:03 am

Hi Jasmin,

We have implemented note 2124672.

In code we are getting below error.



Like (0)

Jasmin Gruschke

April 16, 2015 at 11:38 am

Hi Amol,
please have a look at the latest version of the note. There has been an issue in the version (I think it was V2) you applied which has been fixed yesterday.
Cheers,
Jasmin

Like (0)

Amol Samte

April 22, 2015 at 5:53 am

Hi Jasmin,

As you suggested we implemented latest version of note and its working.

Thanks,

Amol

Like (0)

Vipin Nagpal

May 5, 2015 at 1:13 pm

Hi,

I am trying to execute above example, while executing i have realized that BP_ID and COMPANY_NAME column does not exist in database table snwd_so. If i am trying to execute a report i am getting short dump with message “invalid column name”.

Please correct me if my observation is wrong.

Thanks

Like (0)

Christian Seitel

May 5, 2015 at 1:48 pm

Yes, adhering to the screenshots the where-clause is constructed for table “snwd_bpa” and then applied to table “snwd_so” resulting in a column name mismatch.

Like (0)

Carine Tchoutouo Djomo | Post author

May 5, 2015 at 6:35 pm

Hi Vipin,

you're fully right! Thanks for the feeback!

Let me just put it this way: I wanted to know who really implements the example... 😉 – Just kidding. I've updated the blog with the correct screenshots and it should work for you now.

Kind regards, Carine

Like (1)

Vipin Nagpal

May 6, 2015 at 6:03 am

Thanks a lot for consideration of my request.

Like (0)

Timothy Muchena

May 14, 2015 at 3:25 pm

Hi Carine

Can we use APPLY_FILTER and specify individual table columns at the same time?

Kind regards

Like (0)

Thomas Gauweiler

May 18, 2015 at 12:19 pm

Hi Mr Eli,

you want to use APPY_FLITER on a table, but return only some columns of it?

To my knowledge this is not psssible within a single statement. But the optimizer should be smart enough to avoid overhead in the two steps like:

```
tmp = APPLY_FILTER( table, ... );

result = SELECT ... FROM :tmp;
```

Best Regards, Thomas

Like (0)

Former Member

August 13, 2015 at 7:02 pm

Hi Carine...Thanx for this nice blog.

1) In Class Definition part of AMDP, what Type should be declaired for “iv_where_cond”.

2) In “ABAP For HANA” course , it explained PARAMETERS for CDS Views. Similarly can you please explain how to use SELECT-OPTIONS for CDS View

Like (0)

Jasmin Gruschke

August 17, 2015 at 9:28 am

Hi Kunal,
1) you can use type string.
2) Carine’s article explains how to do “code pushdown” of select options to DB level (as the AMDPs respectively the DB procedures are directly executed on the database), while the “old” select options are rather ABAP language constructs. For CDS views, you can just use the SELECT-OPTIONS, when you consume the CDS View in an Open SQL statement.

Cheers,
Jasmin

Like (0)

Former Member

August 20, 2015 at 10:31 am

Thanks jasmin for ur reply.

-Kunal

Like (0)

Samuele Barzaghi

April 23, 2016 at 8:48 pm

Hi,

SAP note:

2124672 – Converting selection tables into SQL WHERE clause (HDB)

Seems backported to:

7.30 SP 7

7.31 SP 6

7.40 SP 5

Included in:

730	SAPKB73013
730	SAPKB73014
731	SAPKB73116
740	SAPKB74011

I don't know why 73013 and 73014 😉

Bye

Sam

Like (0)

Former Member

May 13, 2016 at 9:32 am

Carine Tchoutouo Djomo: Hi Carine , where good blog . let me know more information on APPLY FILTER as i have to pass two tables

```
SELECT a~partner, b~stat, b~udate, b~utime, b~inact

    INTO TABLE @lt_but000_crm_jcds

    FROM zbut000 AS a INNER JOIN zcrm_jds1 AS b

     ON a~partner_guid = b~objnr

     FOR ALL ENTRIES IN @lt_x
```

```
WHERE a~partner IN @s_partnr

 AND b~stat IN @s_estat

 AND b~udate IN @s_date

 AND b~udate = ( SELECT MAX( udate ) FROM zcrm_jds1 WHERE objnr = b~objnr )

 AND b~utime = ( SELECT MAX( utime ) FROM zcrm_jds1 WHERE objnr = b~objnr

                                  AND udate = ( SELECT MAX( udate ) FROM zcrm_jds1 WHERE objnr = b~objnr ) )

 AND b~inact = @lt_x-val.
```

so I am uisng AMDP with Select options , but getting error where i am passing the values in Apply filter.

Like (0)

 Amol Samte

May 16, 2016 at 5:21 am

Hi,

I guess you are writing above query in AMDP and which is an advanced open SQL thus it is not supporting to AMDP.

-Amol S

Like (0)

 Thomas Gauweiler

May 17, 2016 at 8:24 am

Usually you just do a JOIN in an AMDP instead of an FOR ALL ENTRIES in ABAP as you can pass the interal table as parameter into the AMDP.

But there is not dynamic version of a join. This only works for fixed join conditions.

Regards, Thomas

Like (0)

Pattanaik Satyaki

May 16, 2016 at 7:04 am

Hi,

How do I define the execute method import parameter?

```
PUBLIC SECTION.

  INTERFACES: IF_SHDB_DEF,

              if_amdp_marker_hdb.

  TYPES: ty_where_t type IF_SHDB_DEF=>TT_NAMED_DREF. "where clause

CLASS-METHODS execute

   IMPORTING

     value(iv_where_clause) TYPE ty_where_t.
```

system throws following syntax error.

The method "execute" contains a database procedure, which means that the row type of "IV_WHERE_CLAUSE" must be structured. All components of the row type must be elementary.

Thanks, Saty

Like (0)

Amol Samte

May 16, 2016 at 12:45 pm

You can use type string for where clause...

Like (0)

Former Member

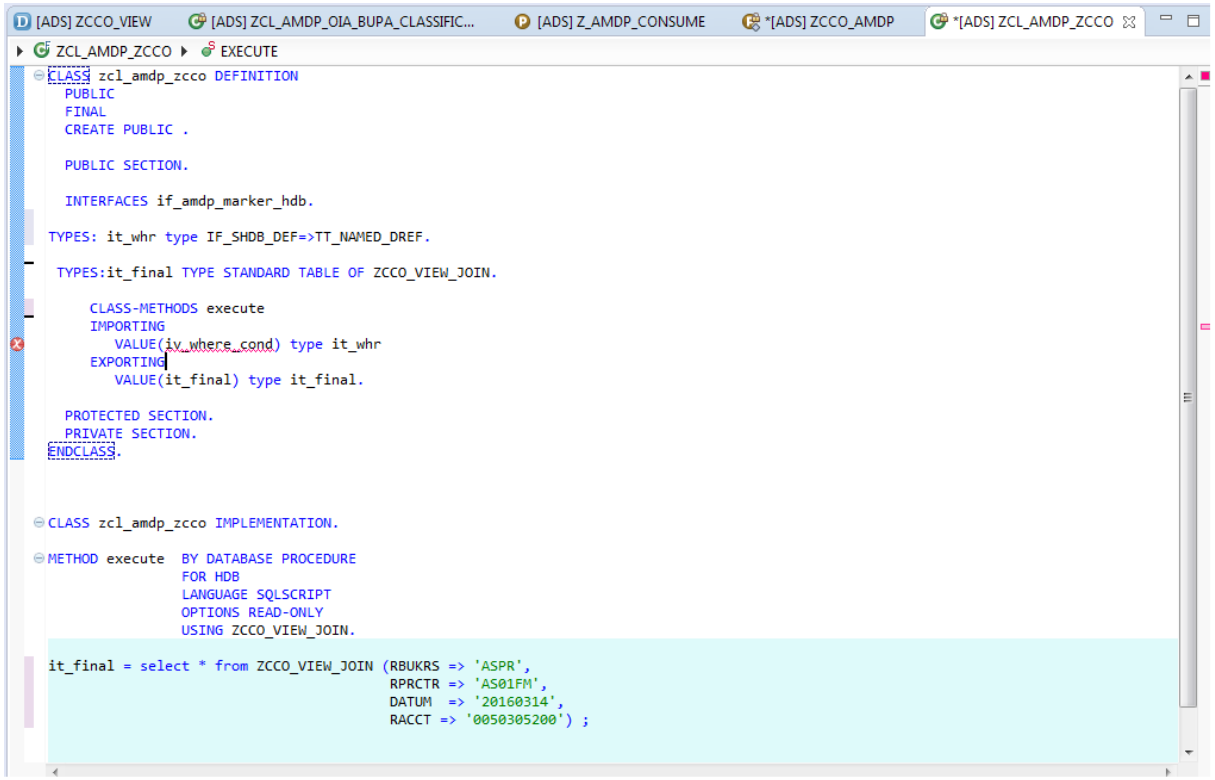
September 21, 2016 at 6:00 am

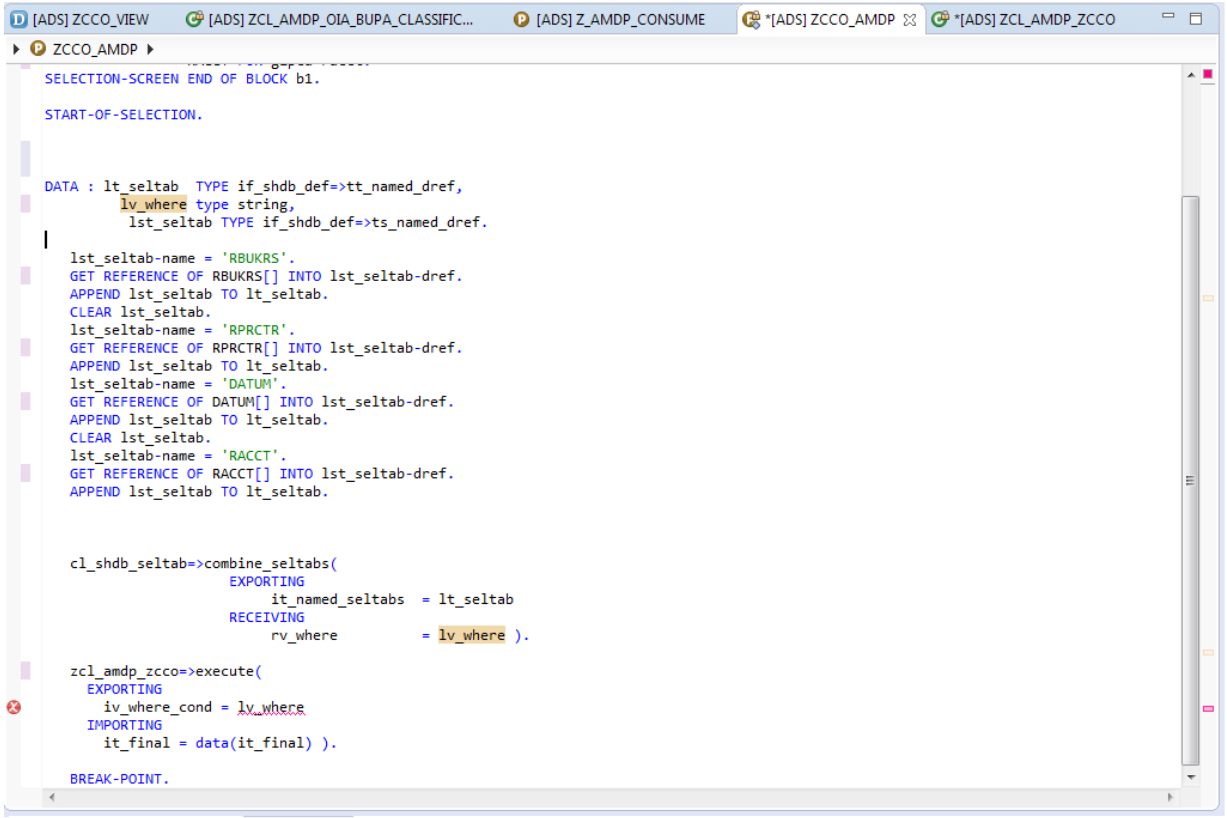
Dear Amol,

I am getting the same above error. Please help[me.

Regards,

Sr.s


[ADS] ZCCO_VIEW
[ADS] ZCL_AMDP_OIA_BUPA_CLASSIFIC...
[ADS] Z_AMDP_CONSUME
*[ADS] ZCCO_AMDP
*[ADS] ZCL_AMDP_ZCCO
ZCL_AMDP_ZCCO
EXECUTE
CLASS zcl_amdp_zcco DEFINITION
PUBLIC
FINAL
CREATE PUBLIC .
PUBLIC SECTION.
INTERFACES if_amdp_marker_hdb.
TYPES: it_whr type IF_SHDB_DEF=>TT_NAMED_DREF.
TYPES:it_final TYPE STANDARD TABLE OF ZCCO_VIEW_JOIN.
CLASS-METHODS execute
IMPORTING
VALUE(iv_where_cond) type it_whr
EXPORTING
VALUE(it_final) type it_final.
PROTECTED SECTION.
PRIVATE SECTION.
ENDCLASS.
CLASS zcl_amdp_zcco IMPLEMENTATION.
METHOD execute BY DATABASE PROCEDURE
FOR HDB
LANGUAGE SQLSCRIPT
OPTIONS READ-ONLY
USING ZCCO_VIEW_JOIN.
it_final = select * from ZCCO_VIEW_JOIN (RBUKRS => 'ASPR',
RPRCTR => 'AS01FM',
DATUM => '20160314',
RACCT => '0050305200') ;



Like (0)

Amol Samte

September 21, 2016 at 7:59 am

declare like below

VALUE(ip_where) TYPE string

Like (0)

Former Member

September 21, 2016 at 9:26 am

Dear Amol,

It is working..

Thank you very much. If I have any doubts I will contact you through mail.

can you give me your mail Id?

Regards,

DSR.

Like (0)

Amol Samte

September 26, 2016 at 6:11 am

Dear Srikant,

I guess providing e mail id here is against SCN rules of engagement...

You can open thread here anytime we are here to support you 🙂

-Amol S

Like (0)

Thomas Gauweiler

May 17, 2016 at 9:24 am

The column names do not match: you do a

SELECT a.client, a.partner, b.stat, b.udate, b.utime, b.inact

so your columns are named CLIENT, PARTNER, STAT, UDATE, UTIME and INACT.

But in the call to COMBINE_SELTABS you name them as S_PARTNER, S_DATE and S_ESTAT. Here you have to use the same as in the SELECT (and as in the result table).

Regards, Thomas

Like (0)

Former Member

May 17, 2016 at 9:30 am

Thank you Thomas for your quick reply , I just fixed it . 🙂 thanks a lot . But not much difference in the performance after using amdp 🙁

Like (0)

Bilen Cekic

October 18, 2017 at 11:22 am

i am 2 years late to comment this blogpost but one of the biggest advantage of apply_filter is it supports SQL inline declarations.

in dynamic sql you cannot do;

```
exec 'lt_DAta = select * from.....'
```

but here you can simply do;

```
lt_Data = apply_Filter(table_name, :cond )
```

lets say you have a parallel processing in ABAP. with EXEC statement you need to use local temporary table to store the data. If 2 different process try to generate same internal table name at the same time, it will give “duplicate table name” error. But second example will just work fine.

Like (1)

rahul reddy

February 15, 2018 at 3:01 pm

why select-option is possible only with AMDP?why not with CDS view

Like (0)

Sreehari V Pillai

March 21, 2018 at 6:48 am

if you read the blog carefully, SELECT OPTIONS is not even available in AMDP too. But, conventionally, ABAP has the feature of select options to read user inputs. What matters here is , converting these range structures to meaningful where clause so that SQL can understand.
Coming to your question – select option is understood only by ABAP Open SQL. CDS is not ABAP. So select options is not understood by CDS. But, from the abap layer, while querying a CDS view, you can use the range variable.
Or, why don’t you use a table function ?

Read my Blog here –
https://blogs.sap.com/2018/03/21/select-options-in-cds-using-table-function/

Sreehari

Like (0)

Udayaditya Boruah

July 26, 2018 at 5:48 pm

Hello Experts,

Thanks for providing insights into using input parameters.

I have a scenario where there are multiple inputs (Date, MANDT and Store) passed as a string to the AMDP class to filter out records in the table. e.g when values are passed to Date, MANDT and Store, the filter1 value looks

FILTER1= RCLNT = '320' AND (BUDAT BETWEEN '20180401' AND '20180724') AND(WERKS = '4797')

Now I want to create a variable to handle the case when **WERKS =** blank.

How do I check whether FILTER1 has WERKS= **Some value** OR **is Blank?**

Any leads would be greatly appreciated.

Regards,

Uday

Like (0)

Thomas Gauweiler

July 31, 2018 at 10:51 am

I am not sure what you want to do.

Do you need a where condition that matches both emtpy values and some given value? That would be … **(WERKS = '4797' OR WERKS = ' ')**

Or do you want to know how to fill a selection table with this condition? Then you need two entries in the WERKS selection table:

WERKS EQ 4979

WERKS EQ

Or do you want to pass the information for searching with and without empty entries? Then you could pass a flag and do a IF statement in AMDP to switch between the condition with and without blank condition.

Regards, Thomas

Like (1)

Udayaditya Boruah

August 1, 2018 at 3:12 pm

Hello Thomas,

Thanks for providing detailed workarounds.

I was basically looking for the third option that you mentioned. I applied the same with an IF condition towards the end of AMDP and then applied the filter.

This way it worked.

Thank you,

Uday

Like (0)

Ashok Kumar

November 19, 2018 at 12:36 pm

I gave this one a try last week and got the AMDP execution failed error and invalid column name. Finally I managed to solve the error. The sample code is misleading. It should not be IV_CLIENT_FIELD = 'CLIENT'. but instead IV_CLIENT_FIELD = 'MANDT'

There was another dump after fixing this error as the system was running out of space when I filter after select query.

Conclusion: If you are working with a huge table like MARA then you need to filter it before the select query and not after.

Like (0)