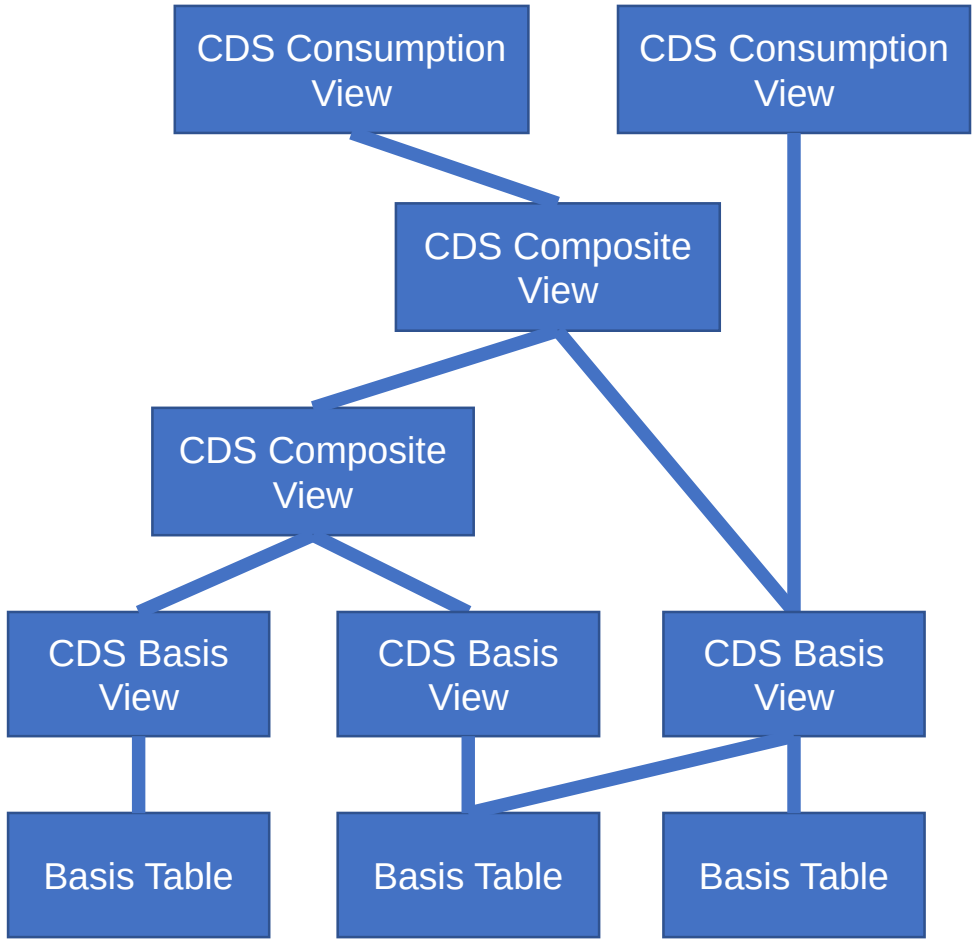
Core Data Services (CDS)

Tutorial 05: Call a CDS with Import Parameters

Alexander Maetzing, 13.11.2020

Repetition: CDS Viewtypes of SAP Virtual Data Model (VDM)



CDS Consumption View: Expose the data to the access of different consumers (ABAP reports, analytics tools, ...).

C_<CDS name>,
ZC_<CDS name>

CDS Composite (Interface) View: Middle layer to combine, manipulate and process data of different CDS Basic Views.

I_<CDS name>,
ZI_<CDS name>

CDS Basic (Interface) View: Fetching the raw data from the real database tables by filtering the access to columns and doing an initial processing.

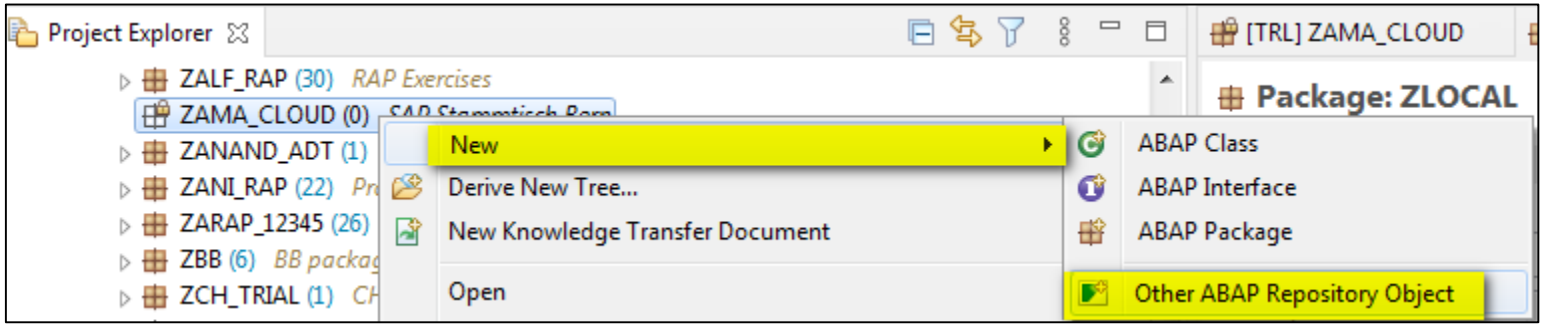
I_<CDS name>,
ZI_<CDS name>

Basis Tables: The database tables containing the raw data.

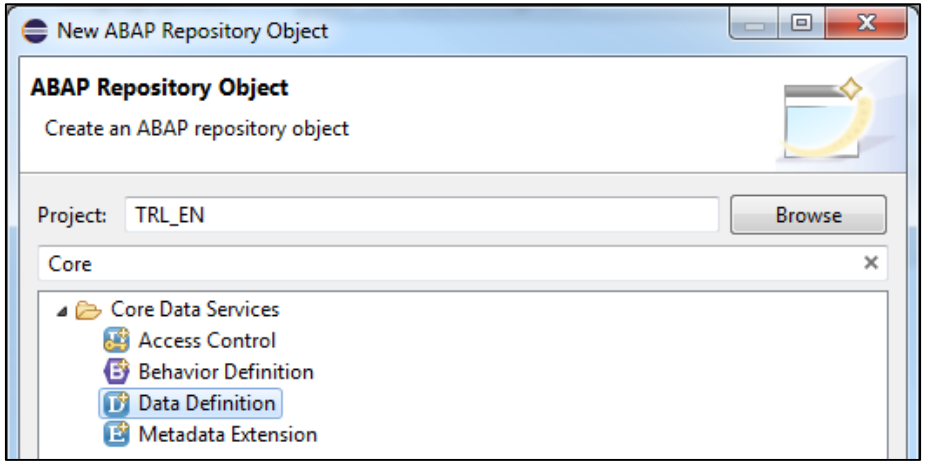
Naming conventions e.g. here:
https://blog.sap-press.com/how-to-name-a-virtual-data-model-in-sap-s4hana

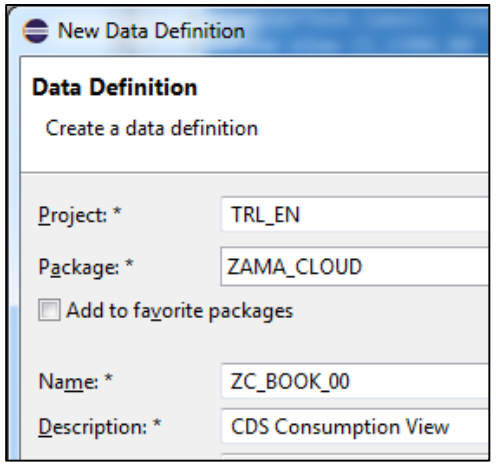
Step 01 - Create a new CDS-View (1)

- Create a new ABAP Repository Object.



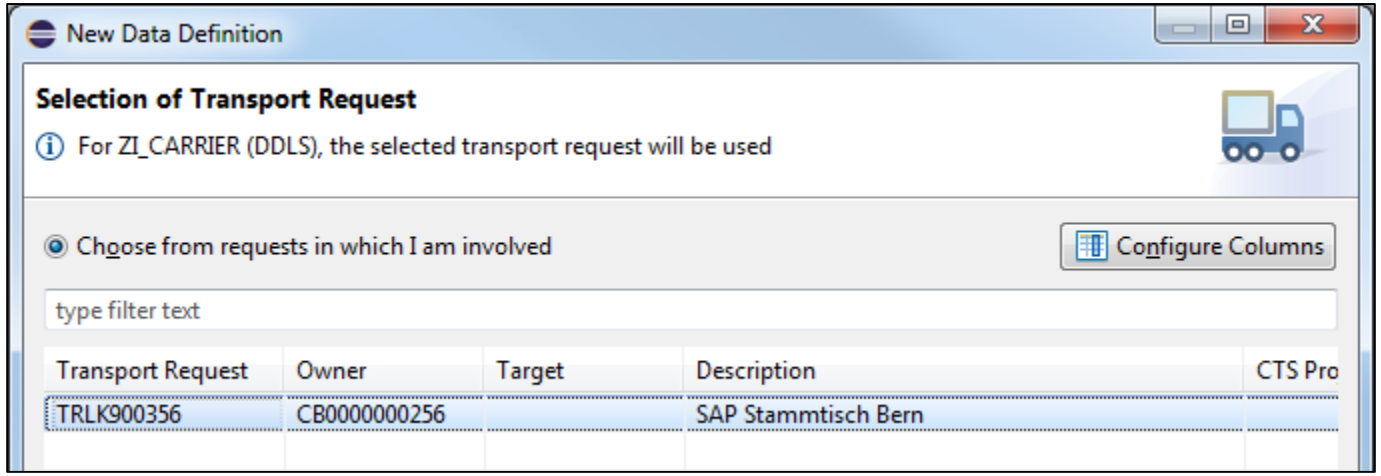
- Choose *Core Data Services -> Data Definition* and create a CDS Consumption View ZC_BOOK_##, following the naming convention.



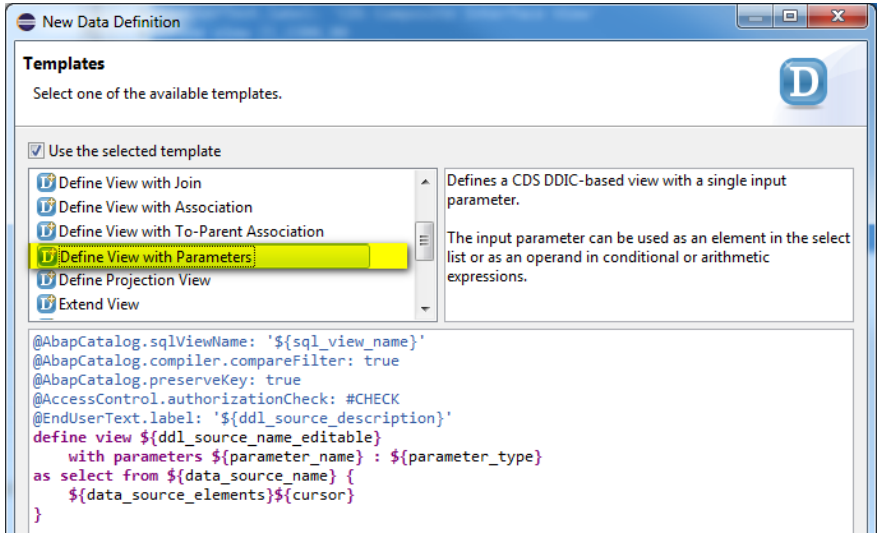


Step 01 - Create a new CDS-View (2)

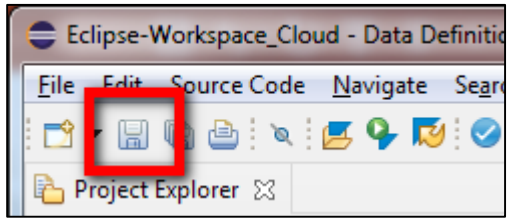
- Confirm the transport - and choose *Next*, **not** *Finish*!



- Select template *Define View with Parameters.*



- Don't forget to *Save* afterwards.



Step 02 - Fill header and field list

- Fill the header with an access to ZI_BOOK_## from the earlier tutorial.

*[TRL] ZC_BOOK_00

```
1  @AbapCatalog.sqlViewName: ''
2  @AbapCatalog.compiler.compareFilter: true
3  @AbapCatalog.preserveKey: true
4  @AccessControl.authorizationCheck: #CHECK
5  @EndUserText.label: 'CDS Consumption View'
6  define view ZC_BOOK_00
7      with parameters parameter_name : parameter_type
8  as select from ZI_BOOK_00 {
9
10 }
```

- Choose only some of the available columns for the field list.

```
10 {
11   key TravelId,
12   key BookingId,
13       BookingDate,
14       CustomerId,
15       CarrierId,
16       ConnectionId,
17       FlightDate,
18       FlightPrice,
19       CurrencyCode
20 }
```

Step 03 - Define a parameter and access it in the field list

- Define parameters after the *with parameters* key word in the header.

```
define view ZC_BOOK_00
  with parameters
    param1: abap.char(3),
    param2: abap_boolean
  as select from ZI_BOOK_00
{
```

  - Use the format *<parameter name> : <parameter type>.*
  - As type either DDIC types or data elements can be used. ATTENTION! In case of SAP ABAP Cloud only released data elements are allowed.
  - Different parameter/type pairs have to be separated by a comma (only the last may not use that comma at the end).

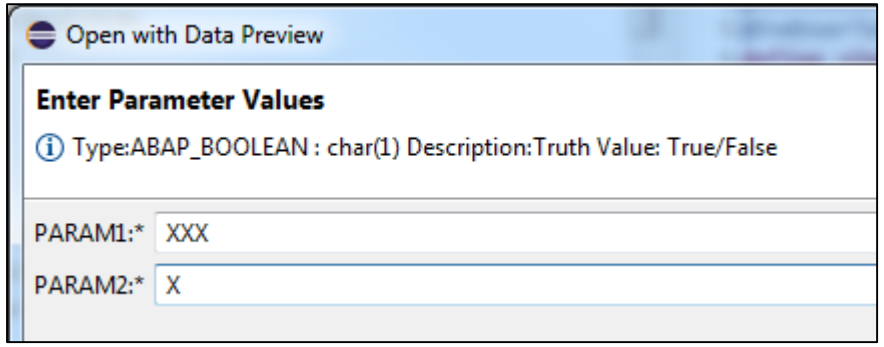
- Use parameters in two different ways.

```
{
  key TravelId,
  key BookingId,
      concat ($parameters.param1, ConnectionId) as ColPar1,
      :param2 as ColPar2,
      BookingDate,
      CustomerId,
```

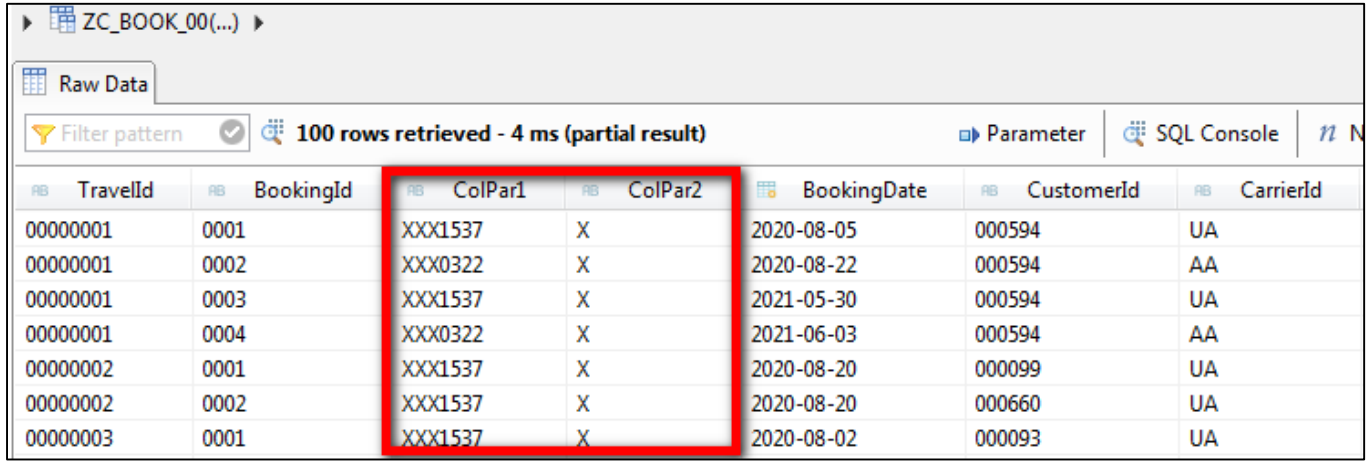
  - Use the prefix *$parameters.* and add the name of the parameter.
  - Use the prefix *:* and add the name of the parameter.

Step 04 - Result in the Preview

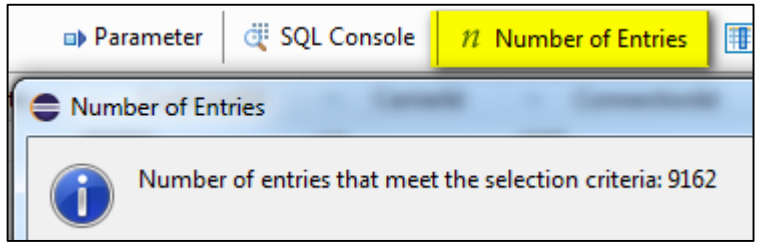
- Call the Preview functionality of the CDS-View from the context menue of the Project Explorer. Fill the values for the parameters as requested in the popup and note the different error-, warning- and information messages that appear.



- Evaluate the result in the preview.



TravelId	BookingId	ColPar1	ColPar2	BookingDate	CustomerId	CarrierId
00000001	0001	XXX1537	X	2020-08-05	000594	UA
00000001	0002	XXX0322	X	2020-08-22	000594	AA
00000001	0003	XXX1537	X	2021-05-30	000594	UA
00000001	0004	XXX0322	X	2021-06-03	000594	AA
00000002	0001	XXX1537	X	2020-08-20	000099	UA
00000002	0002	XXX1537	X	2020-08-20	000660	UA
00000003	0001	XXX1537	X	2020-08-02	000093	UA



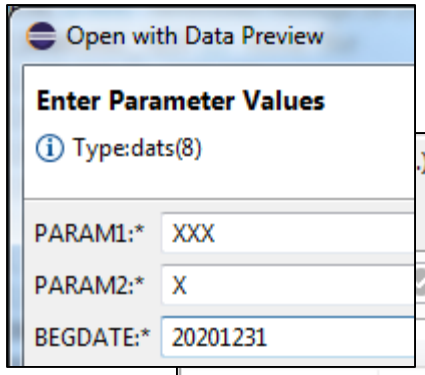
Step 05 - Access a parameter in the WHERE condition

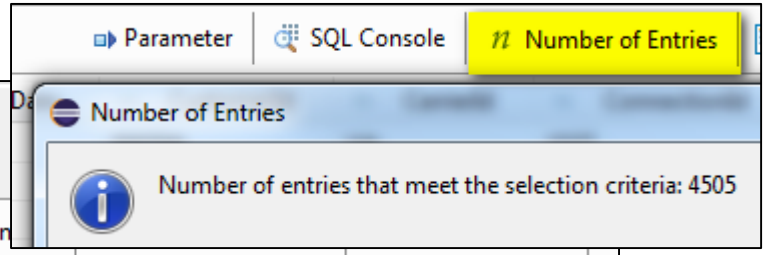
- There are some places where parameter can be used in an SQL View. Create a third parameter begDate as DDIC type `dats` and restrict the retrieved amount of records in the `WHERE` condition.

```
6 define view ZC_BOOK_00
7   with parameters
8     param1: abap.char(3),
9     param2: abap_boolean,
10    begDate: abap.dats
```

```
23      CurrencyCode
24 }
25 where BookingDate < :begDate
```

- Fill the values in the popup and evaluate the result.





100 rows retrieved - 27 ms (partial result)

BookingId	ColPar1	ColPar2	BookingDate	CustomerId	CarrierId	ConnectionId	FlightDate
00000001	0001	XXX1537	X	2020-08-05	000594	UA	1537	2020-08-22
00000001	0002	XXX0322	X	2020-08-22	000594	AA	0322	2020-08-24
00000002	0001	XXX1537	X	2020-08-20	000099	UA	1537	2020-08-22
00000002	0002	XXX1537	X	2020-08-20	000660	UA	1537	2020-08-22
00000003	0001	XXX1537	X	2020-08-02	000093	UA	1537	2020-08-22
00000003	0002	XXX1537	X	2020-08-02	000706	UA	1537	2020-08-22
00000003	0003	XXX0322	X	2020-08-05	000093	AA	0322	2020-08-24

Step 06 - Call a Consumption View with parameters (1)

- Use the ABAP class already created in an earlier tutorial and reuse the interface method `if_oo_adt_classrun~main()`.

```
CLASS zama_cds DEFINITION
  PUBLIC
  FINAL
  CREATE PUBLIC .

  PUBLIC SECTION.
    INTERFACES if_oo_adt_classrun.
  PROTECTED SECTION.
  PRIVATE SECTION.
ENDCLASS.
*****************************************
CLASS zama_cds IMPLEMENTATION.
  METHOD if_oo_adt_classrun~main.

  ENDMETHOD.
ENDCLASS.
```

Step 06 - Call a Consumption View with parameters (2)

- Write down the SELECT statement to ZC_BOOK_##. Use the code completion: Enter the SELECT statement and press after the table name Ctrl + blank.

```
select from zc_book_00
NDMETHOD.
CLASS.

out->write( 'Hello Wo

DATA ls_carrier TYPE
DATA lt_carrier TYPE

SELECT * FROM zi_carr
  WHERE CarrierId = 'LH'
  INTO TABLE @lt_carrier.
out->write( lt_carrier ).

DATA ls_carrier_a TYPE zi_carrier_00_a. "SQL-View
DATA lt_carrier_a TYPE STANDARD TABLE OF zi_carrier_00_a.
 SELECT * FROM zi_carrier_00_a
   WHERE CarrierId = 'LH'
   INTO TABLE @lt_carrier_a.
out->write( lt_carrier_a ).

SELECT FROM zi_flight_00
  FIELDS
    CarrierId AS Fluggesellschaft,
    ConnectionId AS Verbindung,
```

zc_book_00
zc_book_00_a

Press 'Shift+Enter' to insert full signature

ZC_BOOK_00 (View)
CDS Consumption View

Client Handling
- [x] Client dependent
- [ ] Client session variable used

Parameter	*Data Element*	*Data Type*
param1		char(3)
param2	abap_boolean	char(1)
begDate		dats(8)

Column	*Data Element*	*Data Type*	*Unit/Curre*
TravelId	/dmo/travel_id	numc(8)	
BookingId	/dmo/booking_id	numc(4)	
ColPar1		char(7)	
ColPar2		char(1)	
BookingDate	/dmo/booking_date	dats(8)	
CustomerId	/dmo/customer_id	numc(6)	
CarrierId	/dmo/carrier_id	char(3)	
ConnectionId	/dmo/connection_id	numc(4)	
FlightDate	/dmo/flight_date	dats(8)	
FlightPrice	/dmo/flight_price	curr(16,2)	currencyco
CurrencyCode	/dmo/currency_code	cuky(5)	

Step 06 - Call a Consumption View with parameters (3)

- Press shift + return to use the proposal. Due to an error in ADT, the set of parameters is copied twice into the coding. Delete one of the parameter sets. Then give some values to the formal parameters and optionally add a WHERE condition. Finally use method `out( )` to display the data in the console.

```
15    SELECT FROM zc_book_00( param1 = 'XXX'
16                          , param2 = 'X'
17                          , begdate = '20211231' )
18      FIELDS *
19      WHERE FlightPrice < 500
20      INTO TABLE @DATA(gt_zc_book_00).
21    out->write( lines( gt_zc_book_00 ) ).
22    out->write( gt_zc_book_00 ).
```

Problems | Properties | Templates | Bookmarks | Feed Reader | Transport Organizer | Console

ABAP Console

Field

670

Table

TRAVELID	BOOKINGID	COLPAR1	COLPAR2	COLPARTIME	BOOKINGDATE	CUSTOMERID	CARRIERID	CONNECTIONID	FLIGHTDATE	FLIGHTPRICE	CURRENCYCODE
00000001	0001	XXX1537	X	14:16:05	2020-08-05	000594	UA	1537	2020-08-22	438.0	USD
00000001	0002	XXX0322	X	14:16:05	2020-08-22	000594	AA	0322	2020-08-24	438.0	USD
00000001	0003	XXX1537	X	14:16:05	2021-05-30	000594	UA	1537	2021-06-18	438.0	USD
00000001	0004	XXX0322	X	14:16:05	2021-06-03	000594	AA	0322	2021-06-20	438.0	USD
00000002	0001	XXX1537	X	14:16:05	2020-08-20	000099	UA	1537	2020-08-22	438.0	USD
00000002	0002	XXX1537	X	14:16:05	2020-08-20	000660	UA	1537	2020-08-22	438.0	USD
00000003	0001	XXX1537	X	14:16:05	2020-08-02	000093	UA	1537	2020-08-22	438.0	USD

Step 07 - Use ABAP system parameters automatically (1)

- Allow the usage of some ABAP system variables (e.g. sy-datum) as parameter in a CDS-View using an annotation. Please note the two possibilities to define an annotation for a parameter / column.

```
define view ZC_BOOK_00
  with parameters
    param1   : abap.char(3),
    param2   : abap_boolean,
    begDate  : abap.dats,
    pa_date  : abap.dats @<Environment.systemField: #SYSTEM_DATE,
    @Environment.systemField: #SYSTEM_TIME
    pa_time  : abap.tims
  as select from ZI_BOOK_00
```

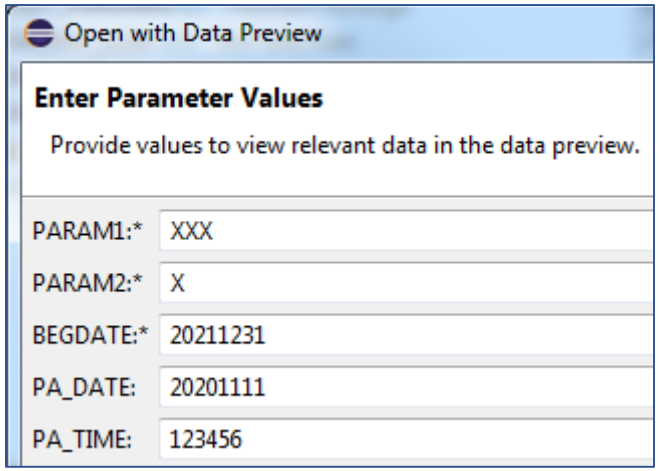
- The usage of the parameters is as defined previously.

```
      FlightPrice,
      CurrencyCode
}
where
      FlightDate between :pa_date and :begDate
```

```
{
  key TravelId,
  key BookingId,
      concat ($parameters.param1, ConnectionId) as ColPar1,
      :param2                                   as ColPar2,
      :pa_time                                  as ColParTime,
      BookingDate,
```

Step 07 - Use ABAP system parameters automatically (2)

- In the preview the data has to be given manually to the CDS-View.



ZC_BOOK_00(...)

Raw Data

100 rows retrieved - 27 ms (partial result)

Parameter | SQL Console | Number of Entries | Select Colu

TravelId	BookingId	ColPar1	ColPar2	ColParTime	BookingDate	CustomerId	CarrierId	ConnectionId	FlightDate
00000001	0003	XXX1537	X	12:34:56 PM	2021-05-30	000594	UA	1537	2021-06-18
00000001	0004	XXX0322	X	12:34:56 PM	2021-06-03	000594	AA	0322	2021-06-20
00000003	0005	XXX1537	X	12:34:56 PM	2021-06-14	000093	UA	1537	2021-06-18
00000003	0006	XXX1537	X	12:34:56 PM	2021-06-14	000706	UA	1537	2021-06-18
00000003	0007	XXX0322	X	12:34:56 PM	2021-06-07	000093	AA	0322	2021-06-20
00000003	0008	XXX0322	X	12:34:56 PM	2021-06-07	000706	AA	0322	2021-06-20
00000004	0003	XXX1537	X	12:34:56 PM	2021-06-07	000665	UA	1537	2021-06-18

Step 07 - Use ABAP system parameters automatically (3)

- In the ABAP program reuse the method main() without any change to the SELECT statement. The usage of the system variables is completely triggeres by the CDS-View without any interference of the ABAP program. This is an option for a very restricted number of system variables only (`sy-mandt`, `sy-uname`, `sy-langu`, `sy-datum`, `sy-uzeit`).

```
15    SELECT FROM zc_book_00( param1 = 'XXX'
16                          , param2 = 'X'
17                          , begdate = '20211231' )
18      FIELDS *
19      WHERE FlightPrice < 500
20      INTO TABLE @DATA(gt_zc_book_00).
21    out->write( lines( gt_zc_book_00 ) ).
22    out->write( gt_zc_book_00 ).
```

Problems | Properties | Templates | Bookmarks | Feed Reader | Transport Organizer | Console

ABAP Console

Field

349

Table

TRAVELID	BOOKINGID	COLPAR1	COLPAR2	COLPARTIME	BOOKINGDATE	CUSTOMERID	CARRIERID	CONNECTIONID	FLIGHTDATE	FLIGHTPRICE	CURRENCYCODE
00000001	0003	XXX1537	X	14:19:02	2021-05-30	000594	UA	1537	2021-06-18	438.0	USD
00000001	0004	XXX0322	X	14:19:02	2021-06-03	000594	AA	0322	2021-06-20	438.0	USD
00000003	0005	XXX1537	X	14:19:02	2021-06-14	000093	UA	1537	2021-06-18	438.0	USD
00000003	0006	XXX1537	X	14:19:02	2021-06-14	000706	UA	1537	2021-06-18	438.0	USD
00000003	0007	XXX0322	X	14:19:02	2021-06-07	000093	AA	0322	2021-06-20	438.0	USD
00000003	0008	XXX0322	X	14:19:02	2021-06-07	000706	AA	0322	2021-06-20	438.0	USD
00000004	0003	XXX1537	X	14:19:02	2021-06-07	000665	UA	1537	2021-06-18	438.0	USD

Appendix - Sourcecode


```
@AbapCatalog.sqlViewName: 'ZC_BOOK_00_A'
@AbapCatalog.compiler.compareFilter: true
@AbapCatalog.preserveKey: true
@AccessControl.authorizationCheck: #CHECK
@EndUserText.label: 'CDS Consumption View'
define view ZC_BOOK_00
  with parameters
    param1  : abap.char(3),
    param2  : abap_boolean,
    begDate : abap.dats,
    pa_date : abap.dats @<Environment.systemField: #SYSTEM_DATE,
    @Environment.systemField: #SYSTEM_TIME
    pa_time : abap.tims
  as select from ZI_BOOK_00
{
  key TravelId,
  key BookingId,
      concat ($parameters.param1, ConnectionId) as ColPar1,
      :param2                                   as ColPar2,
      :pa_time                                  as ColParTime,
      BookingDate,
      CustomerId,
      CarrierId,
      ConnectionId,
      FlightDate,
      FlightPrice,
      CurrencyCode
}
where
      FlightDate between :pa_date and :begDate
//      FlightDate < :begDate
```

```
CLASS zama_cds DEFINITION
  PUBLIC
  FINAL
  CREATE PUBLIC .

  PUBLIC SECTION.
    INTERFACES if_oo_adt_classrun.
  PROTECTED SECTION.
  PRIVATE SECTION.
ENDCLASS.
**********************************************************
CLASS zama_cds IMPLEMENTATION.
  METHOD if_oo_adt_classrun~main.

    SELECT FROM zc_book_00( param1 = 'XXX'
                          , param2 = 'X'
                          , begdate = '20211231' )
      FIELDS *
      WHERE FlightPrice < 500
      INTO TABLE @DATA(gt_zc_book_00).
    out->write( lines( gt_zc_book_00 ) ).
    out->write( gt_zc_book_00 ).

  ENDMETHOD.
ENDCLASS.
```