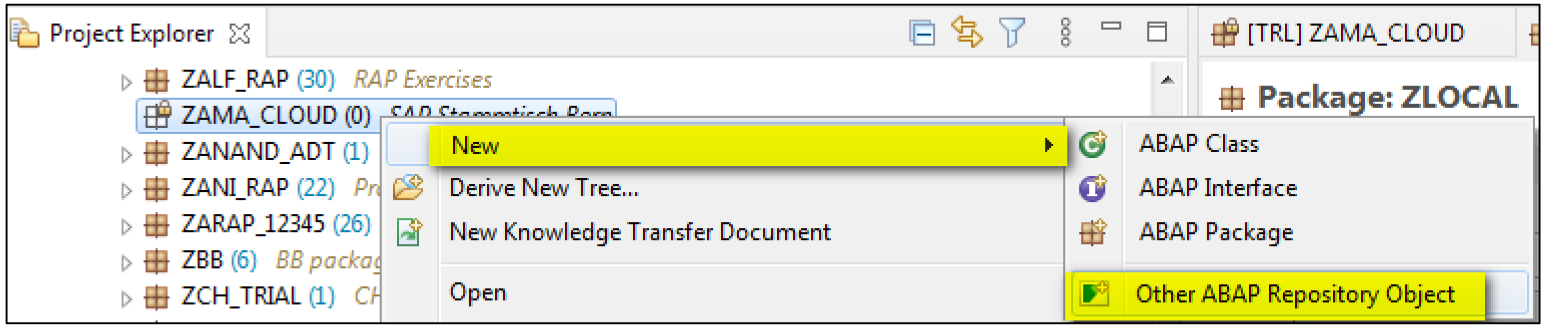
Core Data Services (CDS)

Tutorial 06: Extend an existing CDS and use CDS as data sources

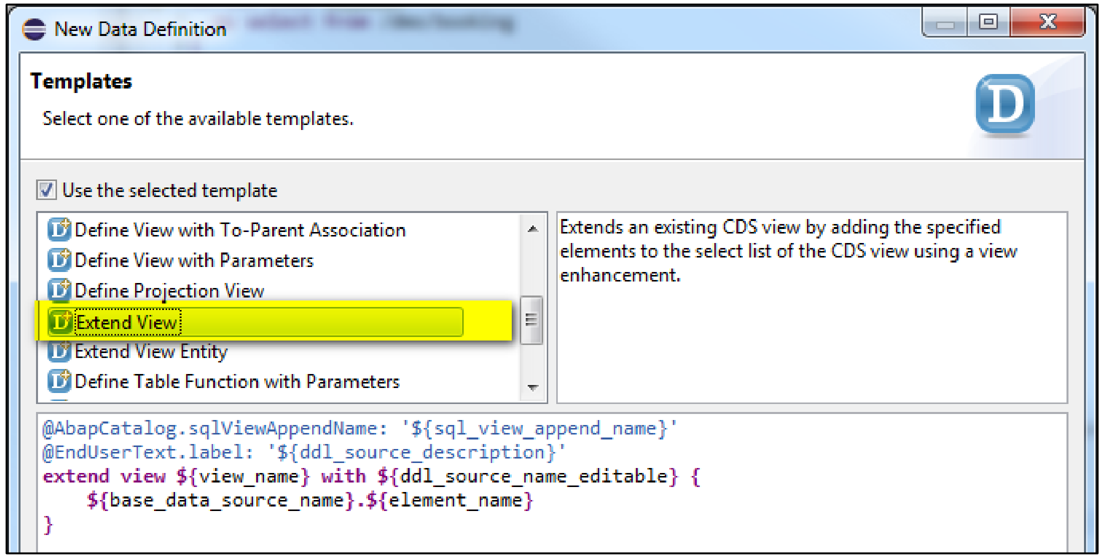
Alexander Maetzing, 13.11.2020

Step 01 - Use a CDS template to extend a CDS

- Create a new *Core Data Service Data Definition* as ABAP Repository Object.



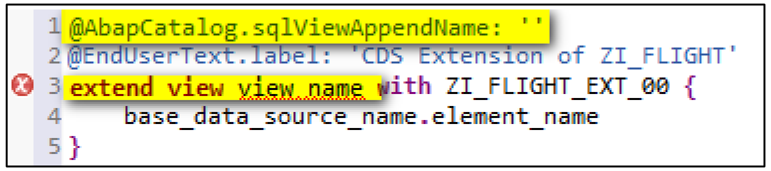
- Use the name ZI_FLIGHT_EXT_## and choose the template *Extend View*.


New Data Definition
Templates
Select one of the available templates.
Use the selected template
Define View with To-Parent Association
Define View with Parameters
Define Projection View
Extend View
Extend View Entity
Define Table Function with Parameters

[✓] Use the selected template

Extends an existing CDS view by adding the specified elements to the select list of the CDS view using a view enhancement.

```
@AbapCatalog.sqlViewAppendName: '${sql_view_append_name}'
@EndUserText.label: '${ddl_source_description}'
extend view ${view_name} with ${ddl_source_name_editable} {
    ${base_data_source_name}.${element_name}
}
```

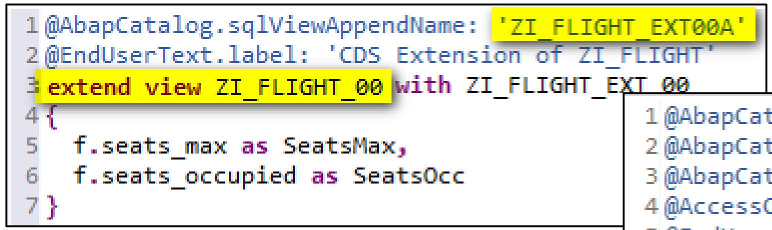


```
@AbapCatalog.sqlViewAppendName: ''
@EndUserText.label: 'CDS Extension of ZI_FLIGHT'
extend view view_name with ZI_FLIGHT_EXT_00 {
    base_data_source_name.element_name
}
```

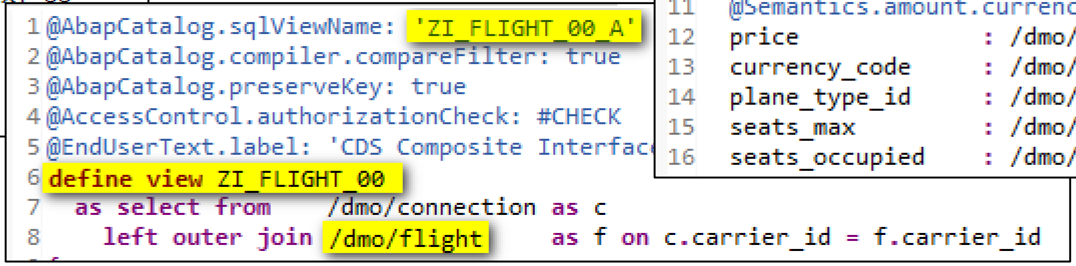
Note that SQL-View and Basis View have to be chosen properly in the next step.

Step 02 - Complete the CDS extension

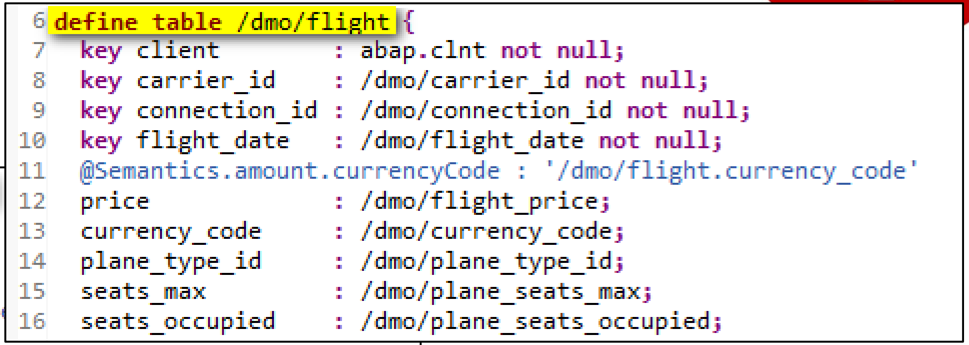
- Fill in the header and data retrieval information.



```
@AbapCatalog.sqlViewAppendName: 'ZI_FLIGHT_EXT00A'
@EndUserText.label: 'CDS Extension of ZI_FLIGHT'
extend view ZI_FLIGHT_00 with ZI_FLIGHT_EXT_00
{
  f.seats_max as SeatsMax,
  f.seats_occupied as SeatsOcc
}
```

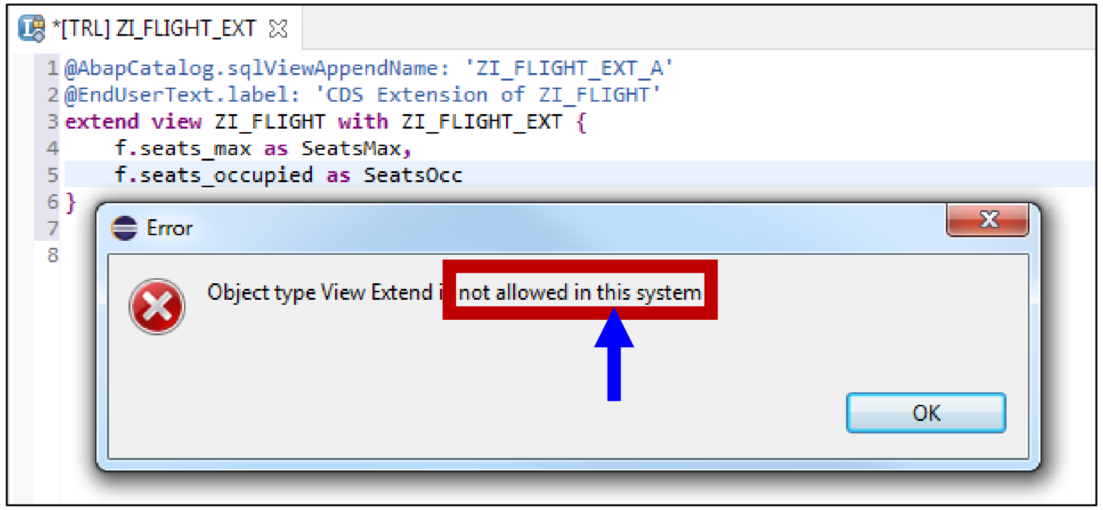


```
@AbapCatalog.sqlViewName: 'ZI_FLIGHT_00_A'
@AbapCatalog.compiler.compareFilter: true
@AbapCatalog.preserveKey: true
@AccessControl.authorizationCheck: #CHECK
@EndUserText.label: 'CDS Composite Interfac
define view ZI_FLIGHT_00
  as select from    /dmo/connection as c
    left outer join /dmo/flight     as f on c.carrier_id = f.carrier_id
```



```
define table /dmo/flight {
  key client         : abap.clnt not null;
  key carrier_id     : /dmo/carrier_id not null;
  key connection_id  : /dmo/connection_id not null;
  key flight_date    : /dmo/flight_date not null;
  @Semantics.amount.currencyCode : '/dmo/flight.currency_code'
  price              : /dmo/flight_price;
  currency_code      : /dmo/currency_code;
  plane_type_id      : /dmo/plane_type_id;
  seats_max          : /dmo/plane_seats_max;
  seats_occupied     : /dmo/plane_seats_occupied;
```

- **ATTENTION!** Be aware of some restrictions in the SAP ABAP Cloud and the Trial Version! Fortunately in any OnPremise system there are no constraints. Extending a CDS View is there an appropriate and easy way to reuse the SAP Virtual Data Model (VDM).



*[TRL] ZI_FLIGHT_EXT

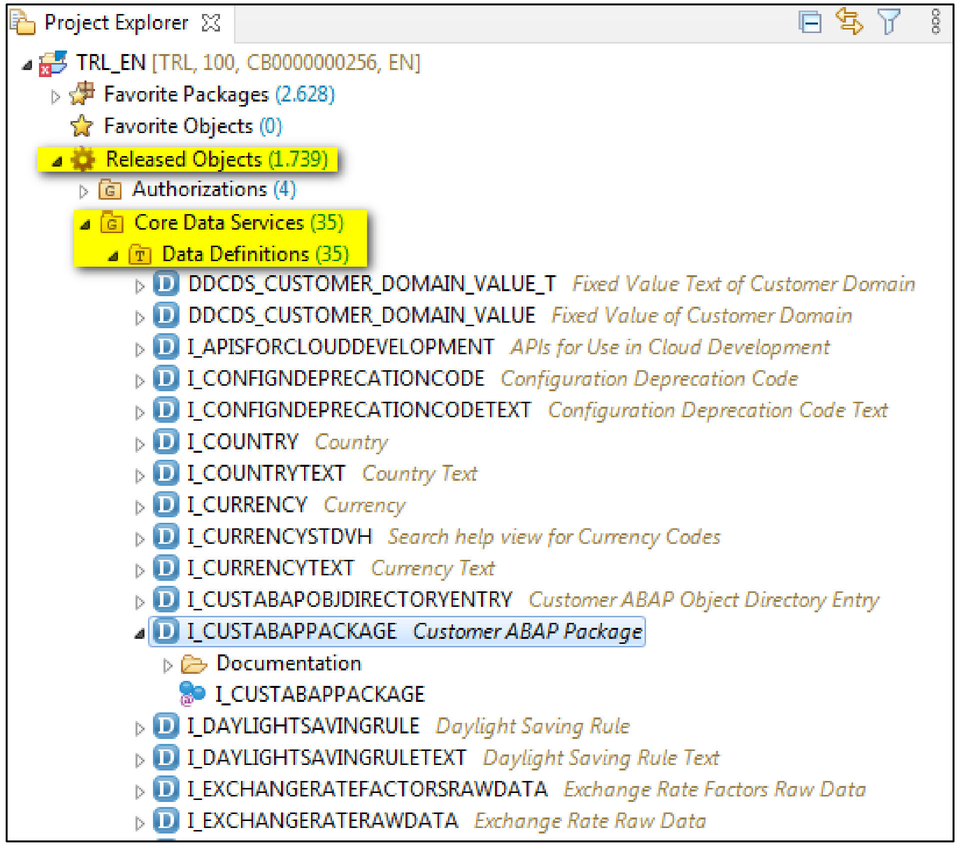
```
@AbapCatalog.sqlViewAppendName: 'ZI_FLIGHT_EXT_A'
@EndUserText.label: 'CDS Extension of ZI_FLIGHT'
extend view ZI_FLIGHT with ZI_FLIGHT_EXT {
    f.seats_max as SeatsMax,
    f.seats_occupied as SeatsOcc
}
```

Error

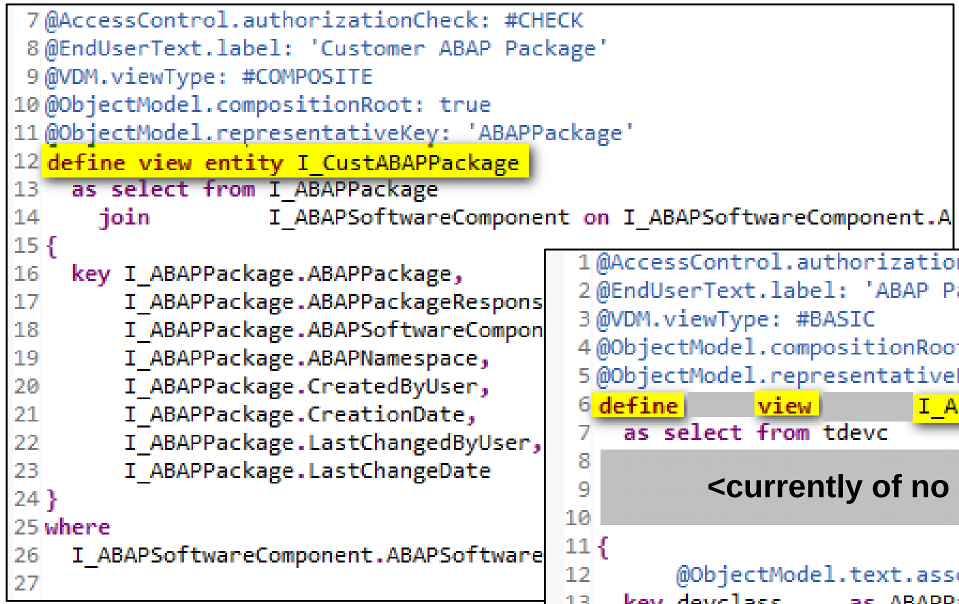
Object type View Extend i not allowed in this system

Step 03 - SAP ABAP Cloud: Find an extendable CDS

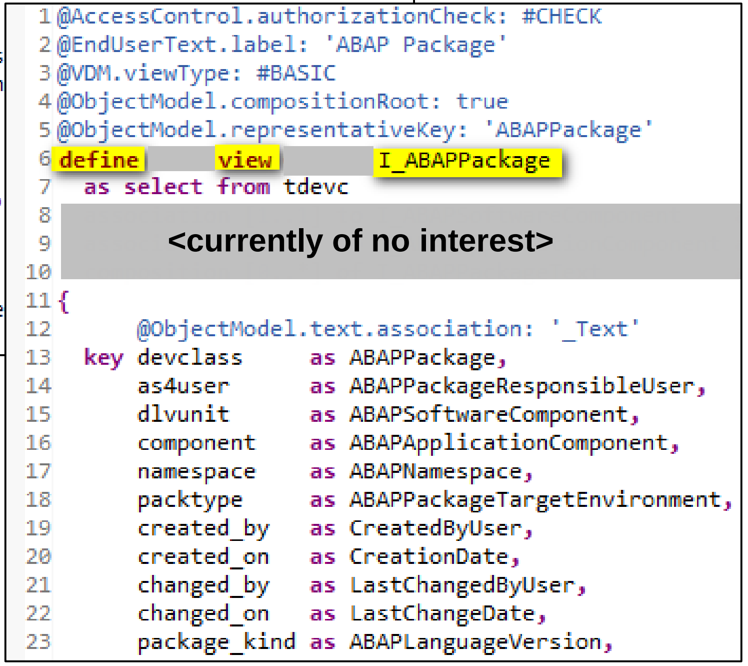
- Check the released Core Data Services in SAP ABAP Cloud. Use a double click (on `I_CUSTABAPPACKAGE`) to navigate in the Project Explorer. Use F3 to jump to a Repository Object definition inside the editor view (e.g. on `I_CustABAPPackage`).



```
Project Explorer
◢ TRL_EN [TRL, 100, CB0000000256, EN]
  ▷ Favorite Packages (2.628)
    Favorite Objects (0)
  ◢ Released Objects (1.739)
    ▷ Authorizations (4)
    ◢ Core Data Services (35)
      ◢ Data Definitions (35)
        ▷ DDCDS_CUSTOMER_DOMAIN_VALUE_T   Fixed Value Text of Customer Domain
        ▷ DDCDS_CUSTOMER_DOMAIN_VALUE   Fixed Value of Customer Domain
        ▷ I_APISFORCLOUDDEVELOPMENT   APIs for Use in Cloud Development
        ▷ I_CONFIGNDEPRECATIONCODE   Configuration Deprecation Code
        ▷ I_CONFIGNDEPRECATIONCODETEXT   Configuration Deprecation Code Text
        ▷ I_COUNTRY   Country
        ▷ I_COUNTRYTEXT   Country Text
        ▷ I_CURRENCY   Currency
        ▷ I_CURRENCYSTDVH   Search help view for Currency Codes
        ▷ I_CURRENCYTEXT   Currency Text
        ▷ I_CUSTABAPOBJDIRECTORYENTRY   Customer ABAP Object Directory Entry
        ◢ I_CUSTABAPPACKAGE   Customer ABAP Package
          ▷ Documentation
            I_CUSTABAPPACKAGE
        ▷ I_DAYLIGHTSAVINGRULE   Daylight Saving Rule
        ▷ I_DAYLIGHTSAVINGRULETEXT   Daylight Saving Rule Text
        ▷ I_EXCHANGERATEFACTORSRAWDATA   Exchange Rate Factors Raw Data
        ▷ I_EXCHANGERATERAWDATA   Exchange Rate Raw Data
```



```
@AccessControl.authorizationCheck: #CHECK
@EndUserText.label: 'Customer ABAP Package'
@VDM.viewType: #COMPOSITE
@ObjectModel.compositionRoot: true
@ObjectModel.representativeKey: 'ABAPPackage'
define view entity I_CustABAPPackage
  as select from I_ABAPPackage
    join          I_ABAPSoftwareComponent on I_ABAPSoftwareComponent.A
{
  key I_ABAPPackage.ABAPPackage,
      I_ABAPPackage.ABAPPackageRespons
      I_ABAPPackage.ABAPSoftwareCompon
      I_ABAPPackage.ABAPNamespace,
      I_ABAPPackage.CreatedByUser,
      I_ABAPPackage.CreationDate,
      I_ABAPPackage.LastChangedByUser,
      I_ABAPPackage.LastChangeDate
}
where
  I_ABAPSoftwareComponent.ABAPSoftware
```



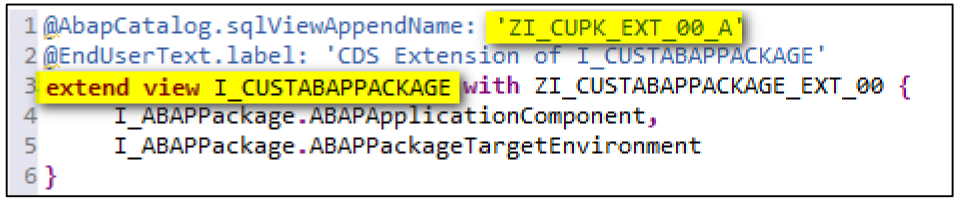
```
@AccessControl.authorizationCheck: #CHECK
@EndUserText.label: 'ABAP Package'
@VDM.viewType: #BASIC
@ObjectModel.compositionRoot: true
@ObjectModel.representativeKey: 'ABAPPackage'
define view I_ABAPPackage
  as select from tdevc

          <currently of no interest>

{
      @ObjectModel.text.association: '_Text'
  key devclass     as ABAPPackage,
      as4user      as ABAPPackageResponsibleUser,
      dlvunit      as ABAPSoftwareComponent,
      component    as ABAPApplicationComponent,
      namespace    as ABAPNamespace,
      packtype     as ABAPPackageTargetEnvironment,
      created_by   as CreatedByUser,
      created_on   as CreationDate,
      changed_by   as LastChangedByUser,
      changed_on   as LastChangeDate,
      package_kind as ABAPLanguageVersion,
```

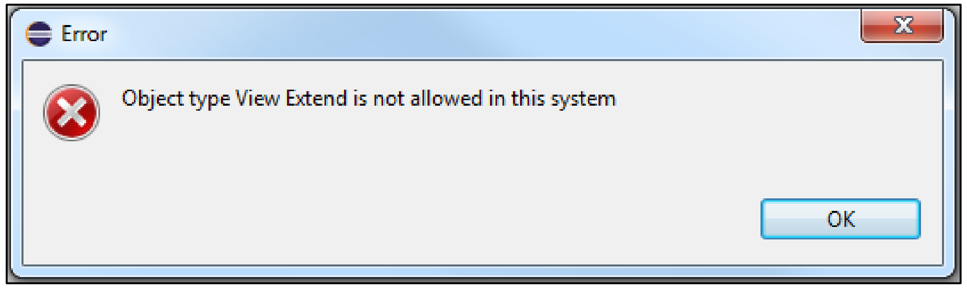
Step 04 - Create an extension for SAP ABAP Cloud

- Create an Extend View ZI_CUSTABAPPACKAGE_EXT_## based on `I_CUSTABAPPACKAGE`.



```
@AbapCatalog.sqlViewAppendName: 'ZI_CUPK_EXT_00_A'
@EndUserText.label: 'CDS Extension of I_CUSTABAPPACKAGE'
extend view I_CUSTABAPPACKAGE with ZI_CUSTABAPPACKAGE_EXT_00 {
      I_ABAPPackage.ABAPApplicationComponent,
      I_ABAPPackage.ABAPPackageTargetEnvironment
}
```

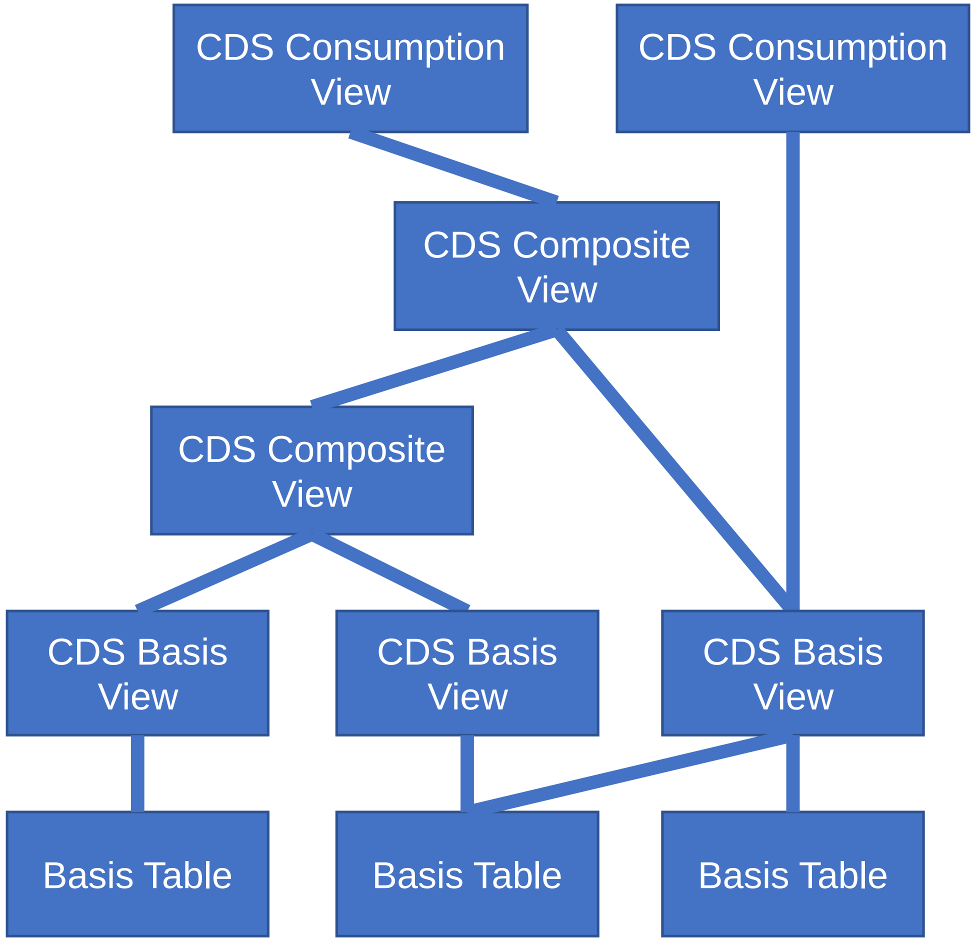
- ATTENTION! Also this CDS-View can't be stored in the Demo system (but - as mentioned earlier - of course in any other (e.g. OnPremise) system).



Error

Object type View Extend is not allowed in this system

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.

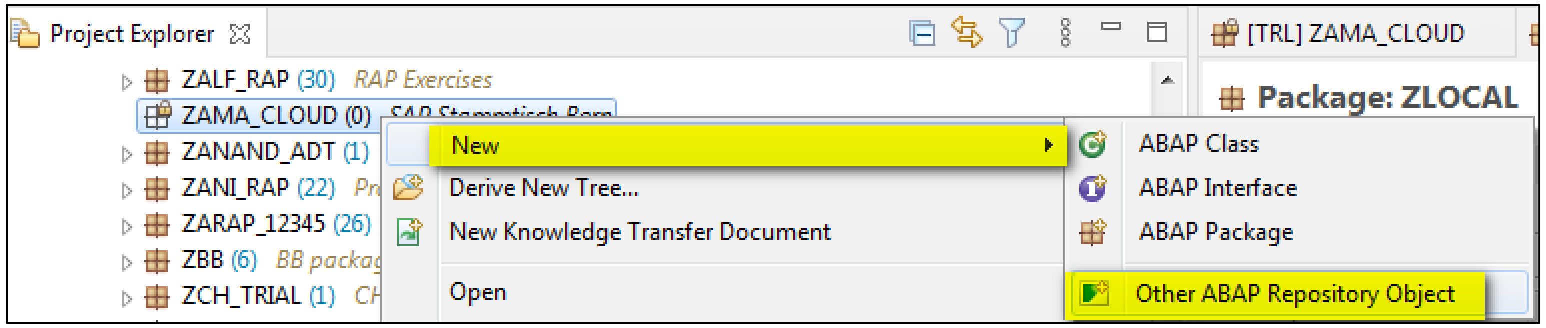
I_<CDS name>,
ZI_<CDS name>

Basis Tables: The database tables containing the raw data.

Naming convention
https://blog.sap-press.com/how-to-name-a-virtual-data-model-in-sap-s4hana

Step 05 - Creating a CDS Composite View

- Create a new *Core Data Service Data Definition* as ABAP Repository Object.



Project Explorer

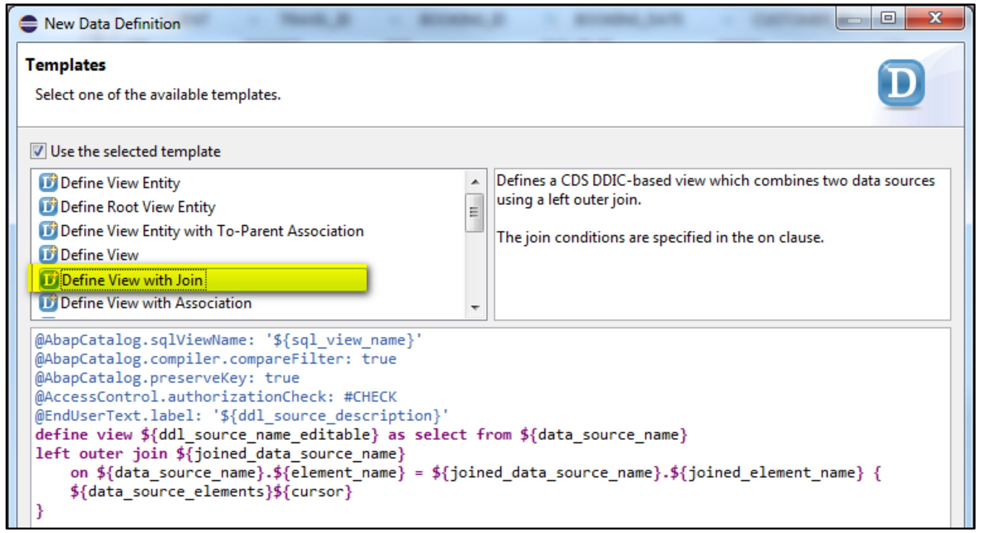
ZALF_RAP (30) RAP Exercises
ZAMA_CLOUD (0) SAP Stammtisch Bern
ZANAND_ADT (1)
ZANI_RAP (22) Pr
ZARAP_12345 (26)
ZBB (6) BB packag
ZCH_TRIAL (1) CH

New
Derive New Tree...
New Knowledge Transfer Document
Open

ABAP Class
ABAP Interface
ABAP Package
Other ABAP Repository Object

[TRL] ZAMA_CLOUD
Package: ZLOCAL

- Use the name ZI_CONN_## and choose the template *Define View with Join*



New Data Definition

Templates

Select one of the available templates.

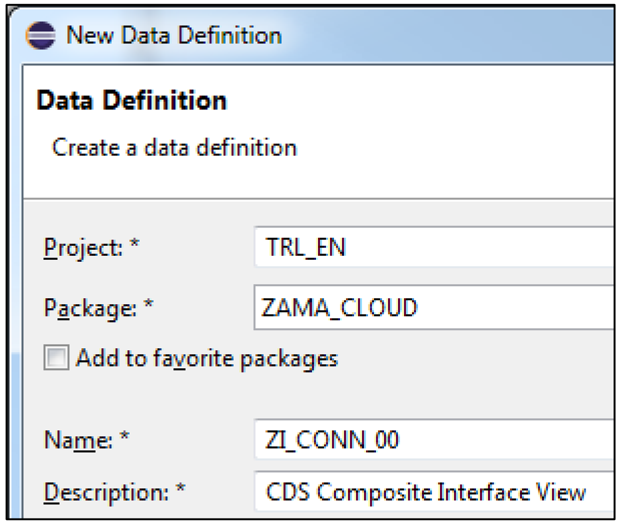
☑ Use the selected template

Define View Entity
Define Root View Entity
Define View Entity with To-Parent Association
Define View
Define View with Join
Define View with Association

Defines a CDS DDIC-based view which combines two data sources using a left outer join.

The join conditions are specified in the on clause.

```
@AbapCatalog.sqlViewName: '${sql_view_name}'
@AbapCatalog.compiler.compareFilter: true
@AbapCatalog.preserveKey: true
@AccessControl.authorizationCheck: #CHECK
@EndUserText.label: '${ddl_source_description}'
define view ${ddl_source_name_editable} as select from ${data_source_name}
left outer join ${joined_data_source_name}
    on ${data_source_name}.${element_name} = ${joined_data_source_name}.${joined_element_name} {
    ${data_source_elements}${cursor}
}
```



New Data Definition

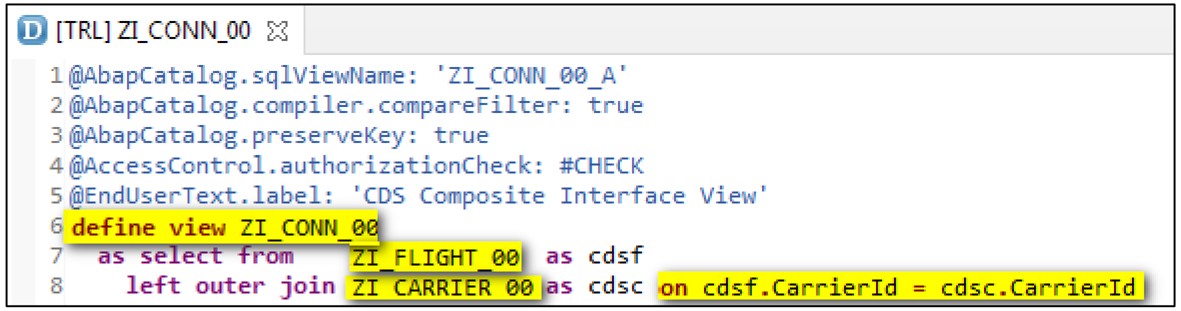
Data Definition

Create a data definition

Project: *	TRL_EN
Package: *	ZAMA_CLOUD
☐ Add to favorite packages	
Name: *	ZI_CONN_00
Description: *	CDS Composite Interface View

Step 06 - Fill the Core Data Service

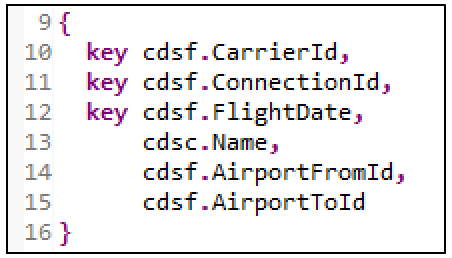
- Decide for a proper name of the SQL-View. Use the interface view ZI_FLIGHT_## and ZI_CARRIER_## as source of the Join. Use aliases for the view names. Set the Join condition. Please see, that for the Join condition the field names of the underlying CDS-Views have to be taken, not the field names of the database tables.



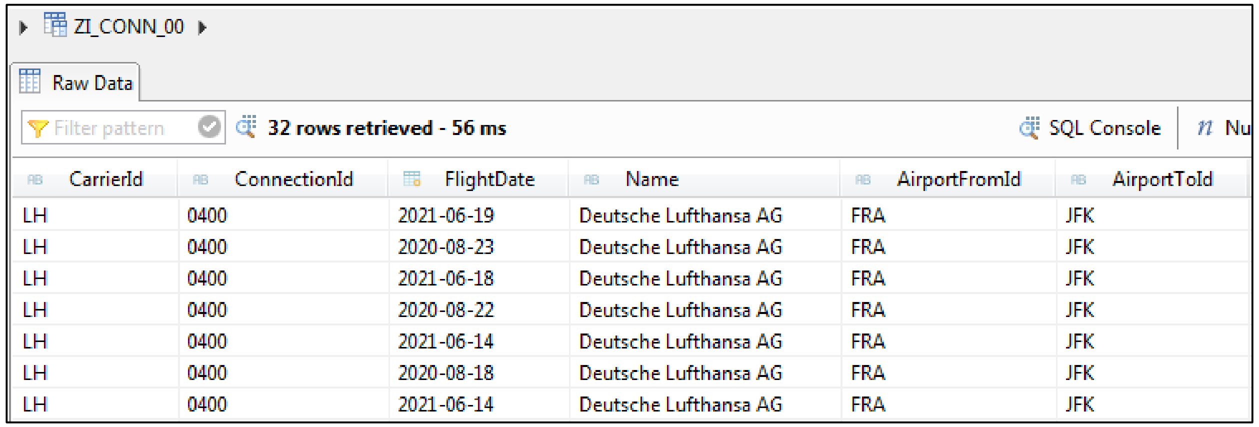
[TRL] ZI_CONN_00

```
@AbapCatalog.sqlViewName: 'ZI_CONN_00_A'
@AbapCatalog.compiler.compareFilter: true
@AbapCatalog.preserveKey: true
@AccessControl.authorizationCheck: #CHECK
@EndUserText.label: 'CDS Composite Interface View'
define view ZI_CONN_00
  as select from     ZI_FLIGHT_00  as cdsf
    left outer join ZI_CARRIER_00 as cdsc on cdsf.CarrierId = cdsc.CarrierId
```

- Fill the field list with some columns of the underlying CDS-Views and check the resulting records.



```
{
  key cdsf.CarrierId,
  key cdsf.ConnectionId,
  key cdsf.FlightDate,
      cdsc.Name,
      cdsf.AirportFromId,
      cdsf.AirportToId
}
```



ZI_CONN_00

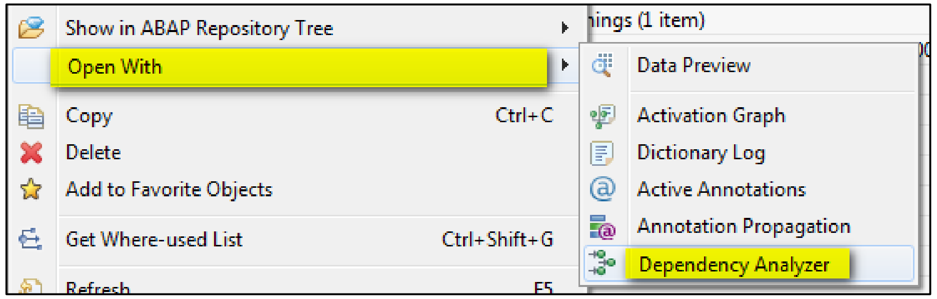
Raw Data

Filter pattern   32 rows retrieved - 56 ms   SQL Console   Nu

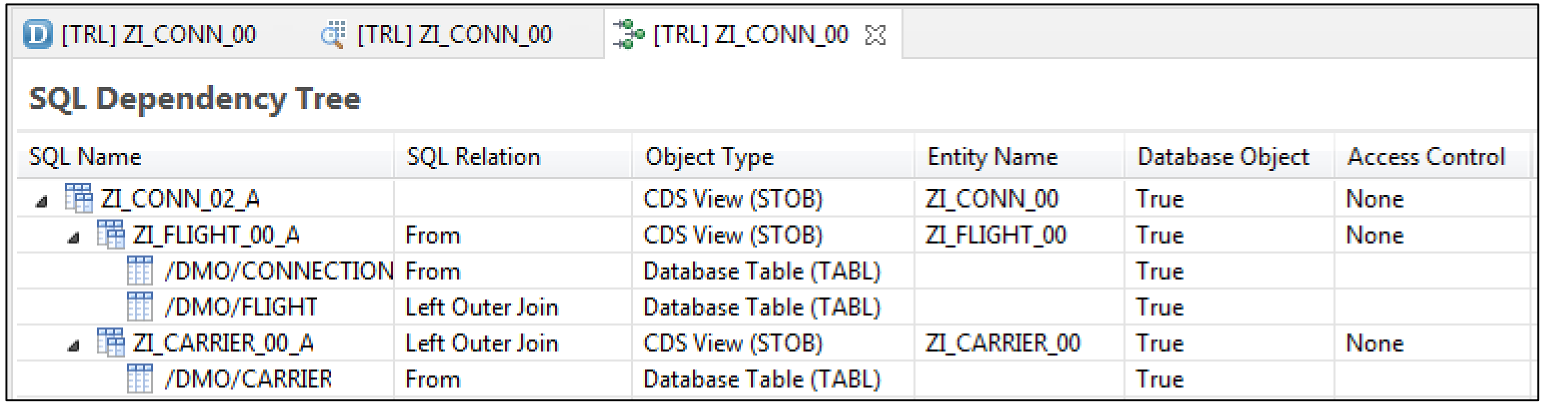
CarrierId	ConnectionId	FlightDate	Name	AirportFromId	AirportToId
LH	0400	2021-06-19	Deutsche Lufthansa AG	FRA	JFK
LH	0400	2020-08-23	Deutsche Lufthansa AG	FRA	JFK
LH	0400	2021-06-18	Deutsche Lufthansa AG	FRA	JFK
LH	0400	2020-08-22	Deutsche Lufthansa AG	FRA	JFK
LH	0400	2021-06-14	Deutsche Lufthansa AG	FRA	JFK
LH	0400	2020-08-18	Deutsche Lufthansa AG	FRA	JFK
LH	0400	2021-06-14	Deutsche Lufthansa AG	FRA	JFK

Step 07 - Evaluate dependencies

- Choose in the context menue of the CDS-View in the Project Explorer *Open With -> Dependency Analyzer*.



- Examine the result.

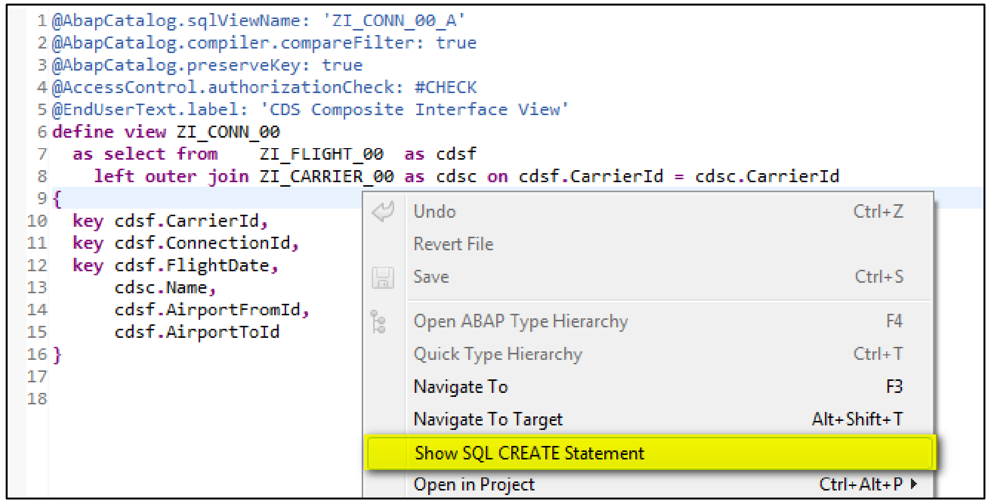


SQL Dependency Tree

SQL Name	SQL Relation	Object Type	Entity Name	Database Object	Access Control
ZI_CONN_02_A		CDS View (STOB)	ZI_CONN_00	True	None
ZI_FLIGHT_00_A	From	CDS View (STOB)	ZI_FLIGHT_00	True	None
/DMO/CONNECTION	From	Database Table (TABL)		True	
/DMO/FLIGHT	Left Outer Join	Database Table (TABL)		True	
ZI_CARRIER_00_A	Left Outer Join	CDS View (STOB)	ZI_CARRIER_00	True	None
/DMO/CARRIER	From	Database Table (TABL)		True	

Step 08 - Evaluate the SQL CREATE statement

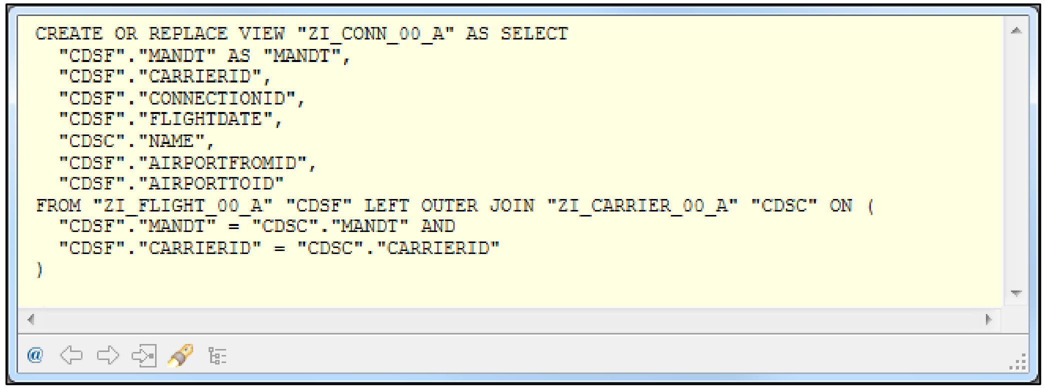
- Use the editor view of the CDS View ZI_CONN_## to call the context menue. Choose *Show SQL CREATE Statement* to have a look how the statement looks like, that creates the CDS-View in HANA.



```
@AbapCatalog.sqlViewName: 'ZI_CONN_00_A'
@AbapCatalog.compiler.compareFilter: true
@AbapCatalog.preserveKey: true
@AccessControl.authorizationCheck: #CHECK
@EndUserText.label: 'CDS Composite Interface View'
define view ZI_CONN_00
  as select from    ZI_FLIGHT_00  as cdsf
    left outer join ZI_CARRIER_00 as cdsc on cdsf.CarrierId = cdsc.CarrierId
{
  key cdsf.CarrierId,
  key cdsf.ConnectionId,
  key cdsf.FlightDate,
      cdsc.Name,
      cdsf.AirportFromId,
      cdsf.AirportToId
}
```

Undo Ctrl+Z
Revert File
Save Ctrl+S
Open ABAP Type Hierarchy F4
Quick Type Hierarchy Ctrl+T
Navigate To F3
Navigate To Target Alt+Shift+T
Show SQL CREATE Statement
Open in Project Ctrl+Alt+P

- Examine the result.



```
CREATE OR REPLACE VIEW "ZI_CONN_00_A" AS SELECT
  "CDSF"."MANDT" AS "MANDT",
  "CDSF"."CARRIERID",
  "CDSF"."CONNECTIONID",
  "CDSF"."FLIGHTDATE",
  "CDSC"."NAME",
  "CDSF"."AIRPORTFROMID",
  "CDSF"."AIRPORTTOID"
FROM "ZI_FLIGHT_00_A" "CDSF" LEFT OUTER JOIN "ZI_CARRIER_00_A" "CDSC" ON (
  "CDSF"."MANDT" = "CDSC"."MANDT" AND
  "CDSF"."CARRIERID" = "CDSC"."CARRIERID"
)
```

Appendix - Sourcecode

```
@AbapCatalog.sqlViewName: 'ZI_CONN_00_A'
@AbapCatalog.compiler.compareFilter: true
@AbapCatalog.preserveKey: true
@AccessControl.authorizationCheck: #CHECK
@EndUserText.label: 'CDS Composite Interface View'
define view ZI_CONN_00
  as select from    ZI_FLIGHT_00  as cdsf
    left outer join ZI_CARRIER_00 as cdsc on cdsf.CarrierId =
cdsc.CarrierId
{
  key cdsf.CarrierId,
  key cdsf.ConnectionId,
  key cdsf.FlightDate,
      cdsc.Name,
      cdsf.AirportFromId,
      cdsf.AirportToId
}
```