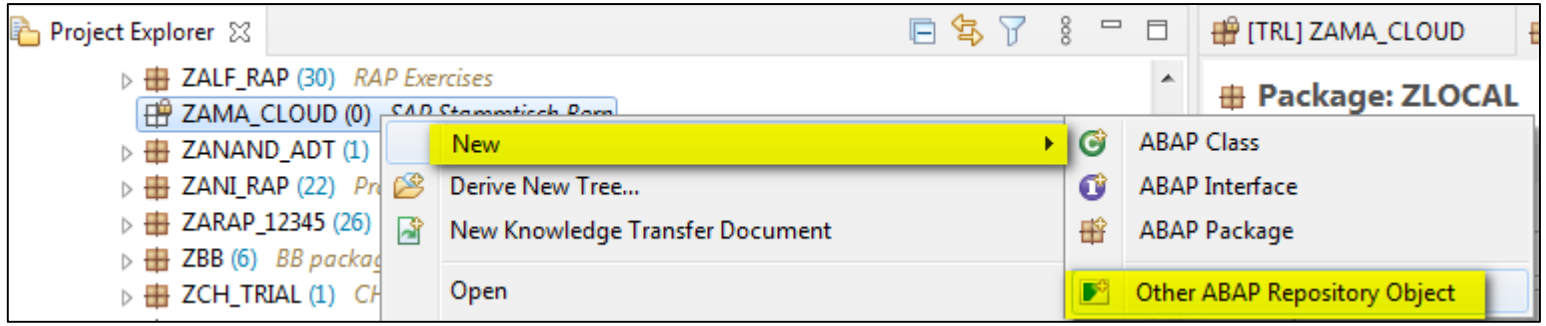
Core Data Services (CDS)

Tutorial 02: Use a Core Data Services for the access to a Join of two tables

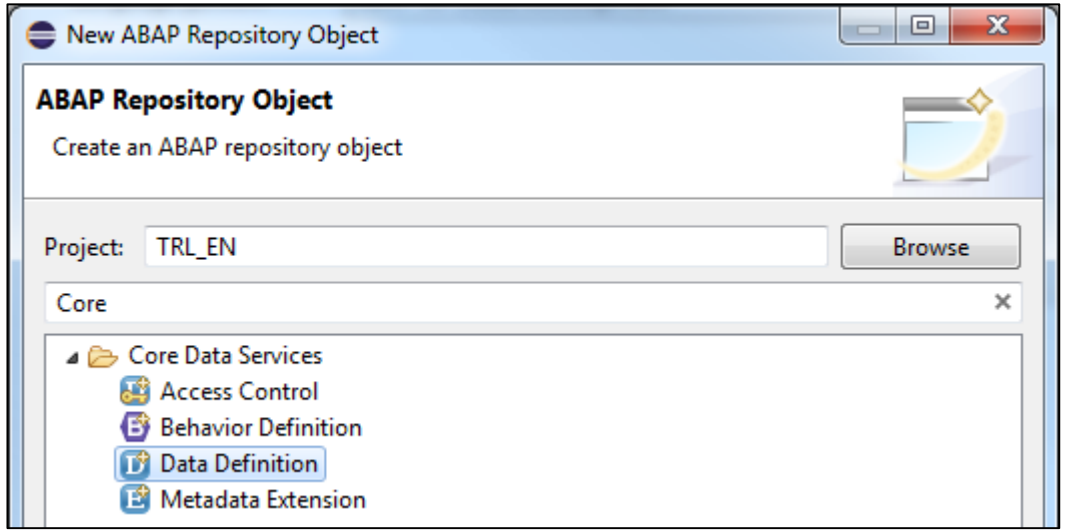
Alexander Maetzing, 13.11.2020

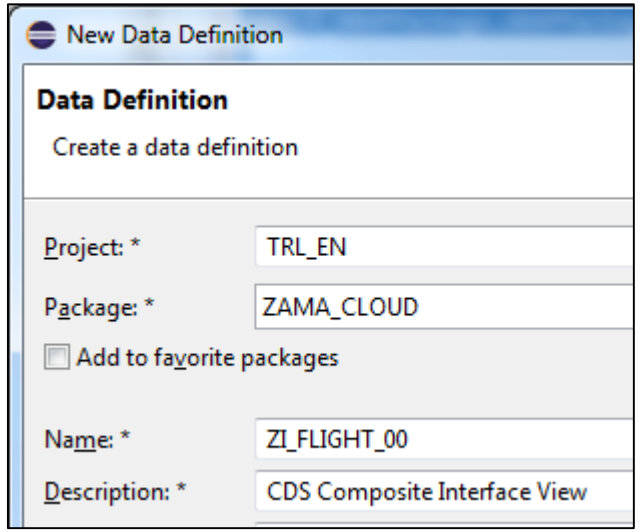
Step 01 - Create a new CDS-View (1)

- Create a new ABAP Repository Object.



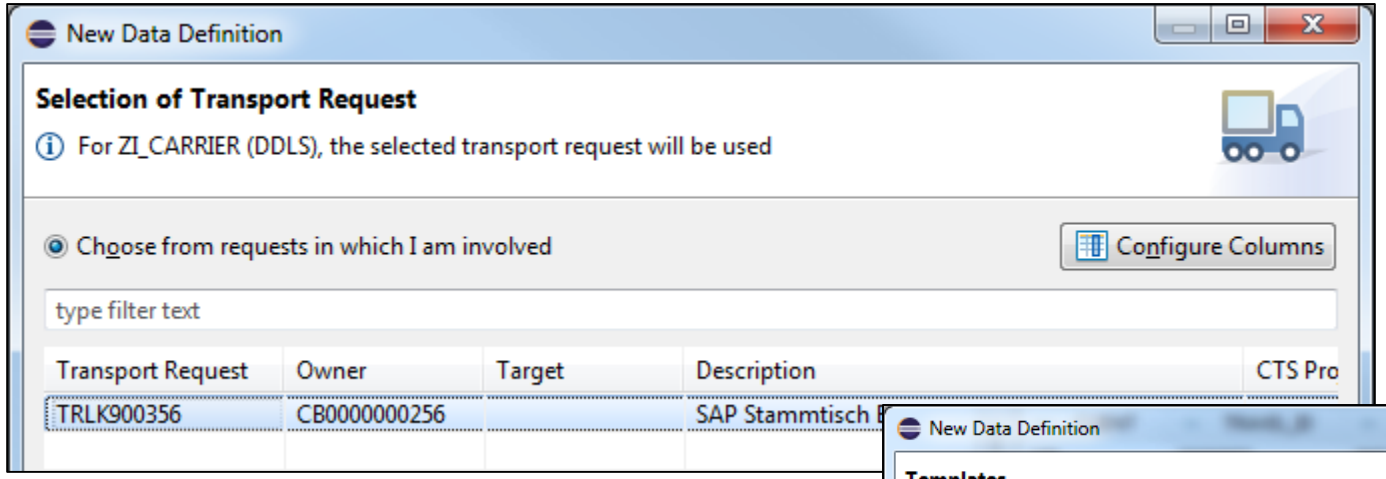
- Choose *Core Data Services -> Data Definition* and create a CDS View ZI_FLIGHT_##.



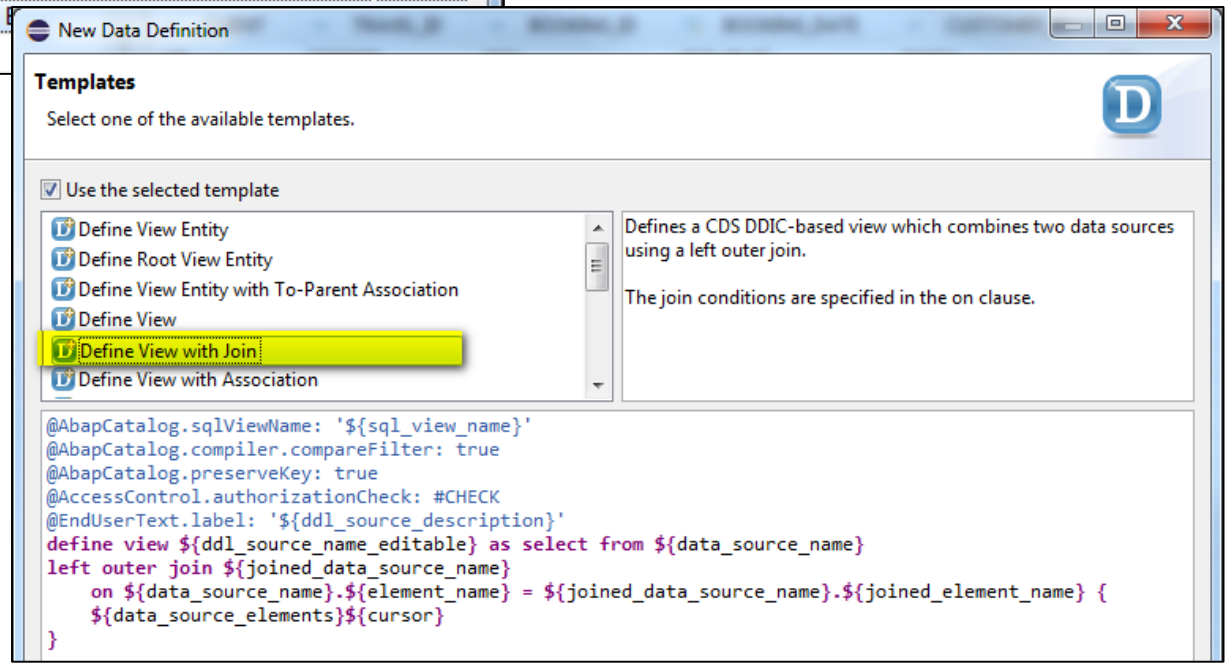


Step 01 - Create a new CDS-View (2)

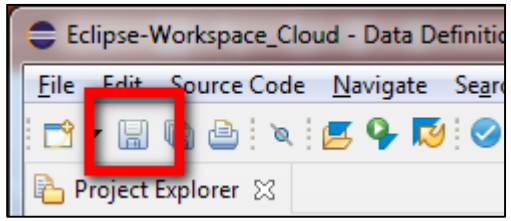
- Confirm the transport - and choose *Next*, **not** *Finish*!



- Select template *Define View with Join.*



- Don't forget to *Save* afterwards.



Step 02 - Adapt the name of the SQL-View

- Define the name of the SQL-View.

*[TRL] ZI_FLIGHT

```
1  @AbapCatalog.sqlViewName: 'ZI_FLIGHT_00_A'
2  @AbapCatalog.compiler.compareFilter: true
3  @AbapCatalog.preserveKey: true
4  @AccessControl.authorizationCheck: #CHECK
5  @EndUserText.label: 'CDS Composite Interface View'
6  define view ZI_FLIGHT as select from data_source_name
7  left outer join joined_data_source_name
8      on data_source_name.element_name = joined_data_source_name.joined_element_name {
9
10 }
11
```

Step 03 - Define the tables and the JOIN condition

- Enter the table names.

```
6  define view ZI_FLIGHT_00 as select from /DMO/CONNECTION
7  left outer join /DMO/FLIGHT
8      on data_source_name.element_name = joined_data_source_name.joined.element_name {
9
10     }
```

- Use aliases for the table names.

```
6  define view ZI_FLIGHT_00
7    as select from    /dmo/connection as c
8      left outer join /dmo/flight as f
9      on data_source_name.element_name = joined_data_source_name.joined_element_name {
10
11     }
```

- Enter the JOIN condition. Be aware of the absence of the JOIN condition for clients! And see that after the introduction of aliases the real table names may not be used anymore in the statement.

```
6  define view ZI_FLIGHT_00
7    as select from    /dmo/connection as c
8      left outer join /dmo/flight as f
9      on c.carrier_id = f.carrier_id {
10
11     }
```

Step 04 - Define the SQL SELECT statement

- Enter the SELECT statement via the Code Completion (*Insert all elements (template)*).

```
 6 define view ZI_FLIGHT_00
 7   as select from    /dmo/connection as c
 8     left outer join /dmo/flight as f
 9     on c.carrier_id = f.carrier_id {
10     key c.carrier_id as CarrierId,
11     key c.connection_id as ConnectionId,
12     key f.carrier_id as CarrierId,
13     key f.connection_id as ConnectionId,
14     key flight_date as FlightDate,
15     airport_from_id as AirportFromId,
16     airport_to_id as AirportToId,
17     departure_time as DepartureTime,
18     arrival_time as ArrivalTime,
19     distance as Distance,
20     distance_unit as DistanceUnit,
21     price as Price,
22     currency_code as CurrencyCode,
23     plane_type_id as PlaneTypeId,
24     seats_max as SeatsMax,
25     seats_occupied as SeatsOccupied
26 }
```

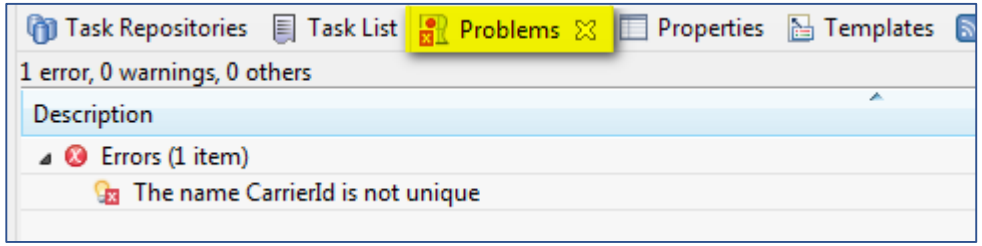
- Replace the full table names in the Join condition with the aliases.
- Note the usage of the aliases before each field. This is not necessary for fields which do not appear with the same name in both tables (as e.g. `carrier_id`). To prove this, remove some of the aliases.
- To prepare a later tutorial, also remove the lines 23 and 24 (`seats_max` and `seats_occupied` from the SELECT statement.

Step 05 - Correct the error for the key columns

- The Code Completion functionality entered the key columns of the single tables as possible keys of the Join each. That leads to the usage of double (column-) aliases, hence to an error.

```
 6 define view ZI_FLIGHT_00
 7   as select from    /dmo/connection as c
 8     left outer join /dmo/flight as f
 9     on c.carrier_id = f.carrier_id {
10     key c.carrier_id as CarrierId,
11     key c.connection_id as ConnectionId,
12     key f.carrier_id as CarrierId,
13     key f.connection_id as ConnectionId,
14     key flight_date as FlightDate,
```

- See the additional (live!) error tab in the box (= Eclipse View) at the bottom.



- Note that although *CarrierId* and *ConnectionId* are both erroneous entries, only *CarrierId* is highlighted. Correct the error by deleting the usage of both columns from table `/DMO/FLIGHT` (line 11 and 12) from the CDS view.

Step 06 - Define a WHERE condition

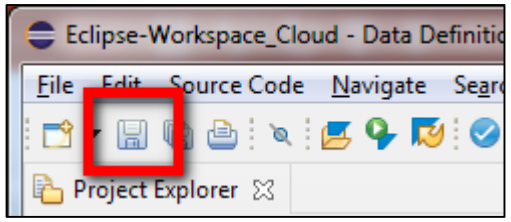
- Restrict the data to the carrier "Lufthansa" with `carrier_id` = 'LH'.

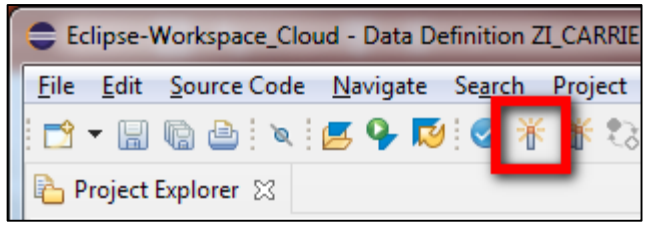
```
define view ZI_FLIGHT_00
  as select from    /dmo/connection as c
    left outer join /dmo/flight     as f on c.carrier_id = f.carrier_id
{
  key c.carrier_id    as CarrierId,
  key c.connection_id as ConnectionId,
  key flight_date     as FlightDate,
      airport_from_id as AirportFromId,
      airport_to_id   as AirportToId,
      departure_time  as DepartureTime,
      arrival_time    as ArrivalTime,
      distance        as Distance,
      distance_unit   as DistanceUnit,
      price           as Price,
      currency_code   as CurrencyCode,
      plane_type_id   as PlaneTypeId
}
where
  c.carrier_id = 'LH'
```

- The WHERE condition starts after the curly brackets of the field list.
- Note also here the absence of a semicolon at the end of the statement.
- The column in the WHERE condition must be prefixed with the alias, because `carrier_id` both appears in `/DMO/CONNECTION` and `/DMO/FLIGHT`.

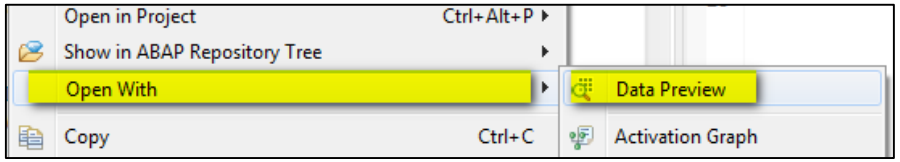
Step 07 - Preview the data

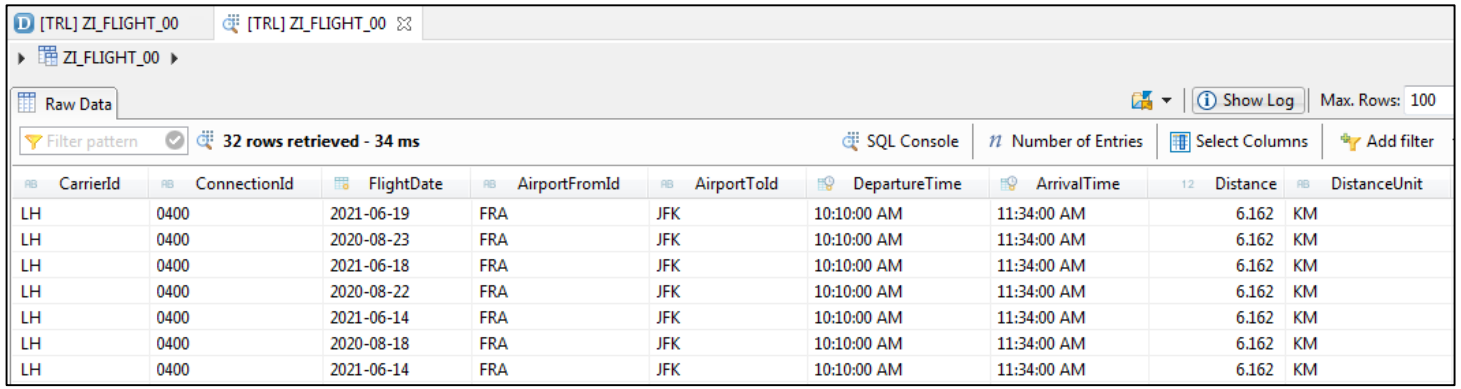
- Save and activate the CDS View.





- Do a preview of the data of the CDS.





CarrierId	ConnectionId	FlightDate	AirportFromId	AirportToId	DepartureTime	ArrivalTime	Distance	DistanceUnit
LH	0400	2021-06-19	FRA	JFK	10:10:00 AM	11:34:00 AM	6.162	KM
LH	0400	2020-08-23	FRA	JFK	10:10:00 AM	11:34:00 AM	6.162	KM
LH	0400	2021-06-18	FRA	JFK	10:10:00 AM	11:34:00 AM	6.162	KM
LH	0400	2020-08-22	FRA	JFK	10:10:00 AM	11:34:00 AM	6.162	KM
LH	0400	2021-06-14	FRA	JFK	10:10:00 AM	11:34:00 AM	6.162	KM
LH	0400	2020-08-18	FRA	JFK	10:10:00 AM	11:34:00 AM	6.162	KM
LH	0400	2021-06-14	FRA	JFK	10:10:00 AM	11:34:00 AM	6.162	KM

Note the QuickInfo for e.g. the column ConnectionId by hovering the mouse at the column's header.

Step 08 - Use Annotations to change the CDS behaviour (1)

- Annotations add metadata to the SELECT statement. Make an Annotation in the field list of the SELECT statement for a column. This allows e.g. to transfer DDIC properties to a CDS View

```
22        c.distance_unit    as DistanceUnit,
23        @Semantics.amount.currencyCode: 'CurrencyCode'
24        price              as Price,
25        @Semantics.currencyCode
26        currency_code      as CurrencyCode,
27        plane_type_id      as PlaneTypeId
```

- Make an Annotation for the whole CDS View in the header of the view. This allows e.g. a client dependency of an ABAP SELECT-statement with `USING CLIENT` accessing the CDS View.

```
1  @AbapCatalog.sqlViewName: 'ZI_FLIGHT_00_A'
2  @AbapCatalog.compiler.compareFilter: true
3  @AbapCatalog.preserveKey: true
4  @AccessControl.authorizationCheck: #CHECK
5  @EndUserText.label: 'CDS Composite Interface View'
6  @ClientHandling.type: #CLIENT_DEPENDENT
7  @ClientHandling.algorithm: #SESSION_VARIABLE
8  define view ZI_FLIGHT
9    as select from    /dmo/connection as c
10     left outer join /dmo/flight     as f on  c.carrier_id = f.carrier_id
11 {
```

- A direct result of these Annotations e.g. in the data preview is currently not visible.

Appendix - Sourcecode


```
@AbapCatalog.sqlViewName: 'ZI_FLIGHT_00_A'
@AbapCatalog.compiler.compareFilter: true
@AbapCatalog.preserveKey: true
@AccessControl.authorizationCheck: #CHECK
@EndUserText.label: 'CDS Composite Interface View'
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
{
  key c.carrier_id                      as CarrierId,
  key c.connection_id                   as ConnectionId,
  key f.flight_date                     as FlightDate,
      c.airport_from_id                 as AirportFromId,
      c.airport_to_id                   as AirportToId,
      c.departure_time                  as DepartureTime,
      c.arrival_time                    as ArrivalTime,
      c.distance                        as Distance,
      c.distance_unit                   as DistanceUnit,
      price                             as Price,
      currency_code                     as CurrencyCode,
      plane_type_id                     as PlaneTypeId
}
where
  c.carrier_id = 'LH'
```