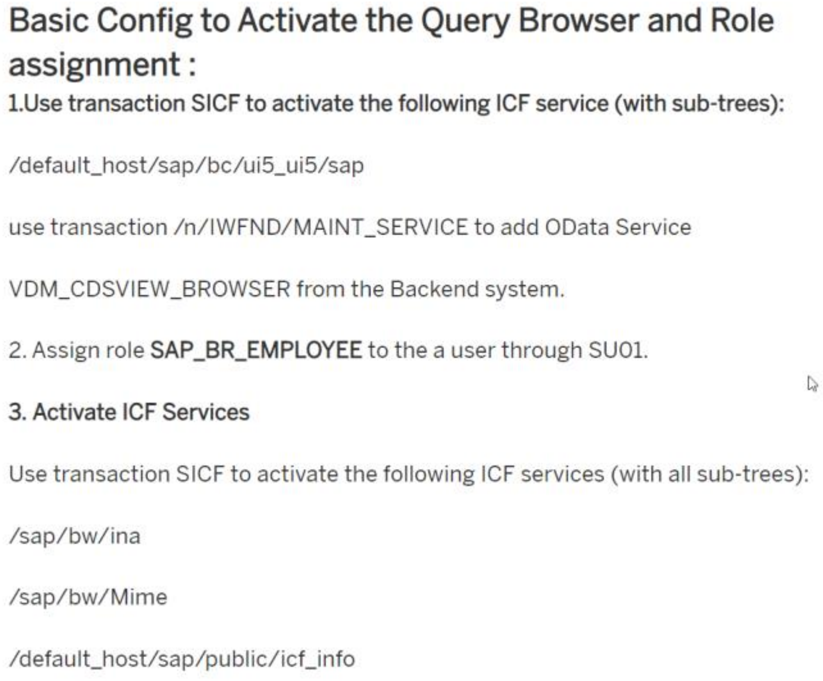
S12 - BW Modeling - ADSO

ADSO	1. Definition and Positioning 2. Types of ADSOs 3. Mapping to previous versions 4. Details on each type of ADSO 5. Active/Nonactive concept 6. Partitioning and Indexing	12th Sep: 10 AM - 12 Noon

Q & A - Pre Session

CDS View - Tools and Tables

Tcode SDDLAR	CDS analysis and Repair Tools
Table RSODPABAPCDSVIEW	The connection between DDLNAME and SQLVIEWNAME
Tcode RSRTS_ODP_DIS	Transaction RSRTS_ODP_DIS displays transient providers of different contexts. It can be used to check if the analytic engine is able to interpret the underlying metadata correctly.
CDS Query Browser	 **Basic Config to Activate the Query Browser and Role assignment :** **1.Use transaction SICF to activate the following ICF service (with sub-trees):** /default_host/sap/bc/ui5_ui5/sap use transaction /n/IWFND/MAINT_SERVICE to add OData Service VDM_CDSVIEW_BROWSER from the Backend system. 2. Assign role **SAP_BR_EMPLOYEE** to the a user through SU01. **3. Activate ICF Services** Use transaction SICF to activate the following ICF services (with all sub-trees): /sap/bw/ina /sap/bw/Mime /default_host/sap/public/icf_info

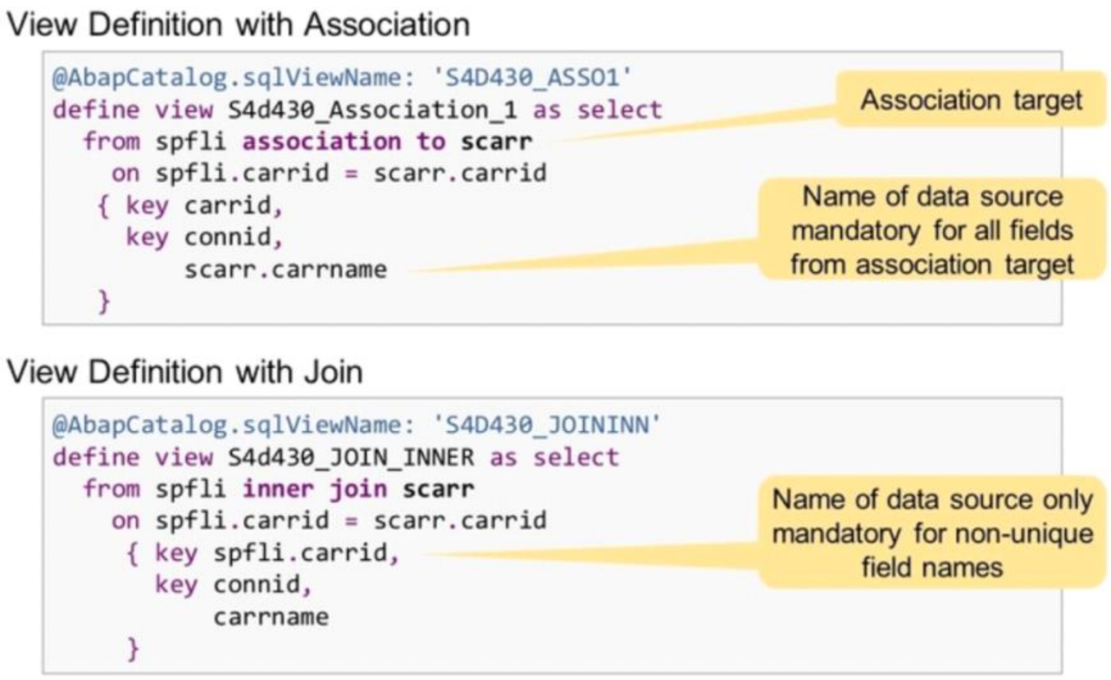
Associations in CDS Views

Associations

1. Join on Demand

Associations will only be triggered when user would access the required data which needs the Association of tables. For example, your CDS view has 4 Associations configured and user is fetching data for only 2 tables, the ASSOICATION on other 2 tables will not be triggered and the system would return the results quickly, so it enables really high turn-around time as compared to regular SQL JOINS.

2. Association vs Join



View Definition with Association

```
@AbapCatalog.sqlViewName: 'S4D430_ASSO1'
define view S4d430_Association_1 as select
  from spfli association to scarr
    on spfli.carrid = scarr.carrid
   { key carrid,
     key connid,
         scarr.carrname
   }
```

Association target

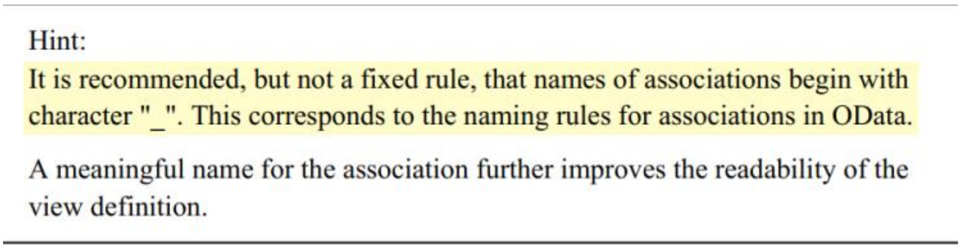
Name of data source mandatory for all fields from association target

View Definition with Join

```
@AbapCatalog.sqlViewName: 'S4D430_JOININN'
define view S4d430_JOIN_INNER as select
  from spfli inner join scarr
    on spfli.carrid = scarr.carrid
     { key spfli.carrid,
       key connid,
           carrname
     }
```

Name of data source only mandatory for non-unique field names

3. Naming Convention



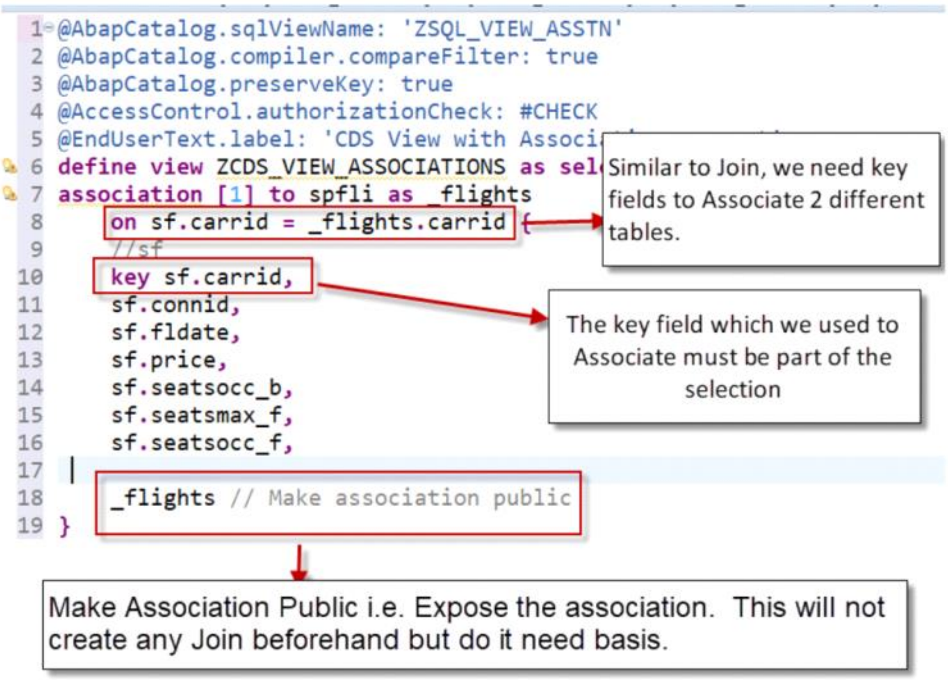
Hint:

It is recommended, but not a fixed rule, that names of associations begin with character "_". This corresponds to the naming rules for associations in OData.

A meaningful name for the association further improves the readability of the view definition.

Exposing Associations: (No join is executed at the first go)

-EXPOSED Association



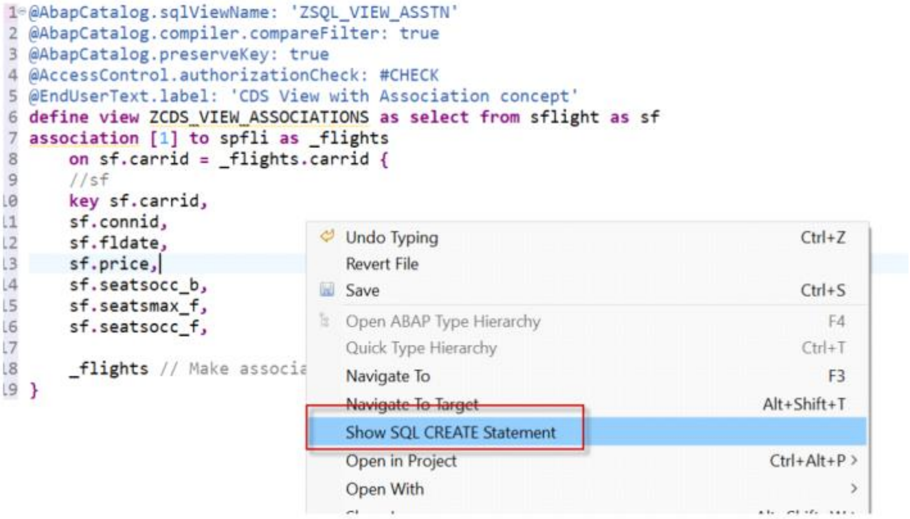
```
@AbapCatalog.sqlViewName: 'ZSQL_VIEW_ASSTN'
@AbapCatalog.compiler.compareFilter: true
@AbapCatalog.preserveKey: true
@AccessControl.authorizationCheck: #CHECK
@EndUserText.label: 'CDS View with Associ
define view ZCDS_VIEW_ASSOCIATIONS as sel
association [1] to spfli as _flights
    on sf.carrid = _flights.carrid {
    //sf
    key sf.carrid,
    sf.connid,
    sf.fldate,
    sf.price,
    sf.seatsocc_b,
    sf.seatsmax_f,
    sf.seatsocc_f,

    _flights // Make association public
}
```

Similar to Join, we need key fields to Associate 2 different tables.

The key field which we used to Associate must be part of the selection

Make Association Public i.e. Expose the association. This will not create any Join beforehand but do it need basis.

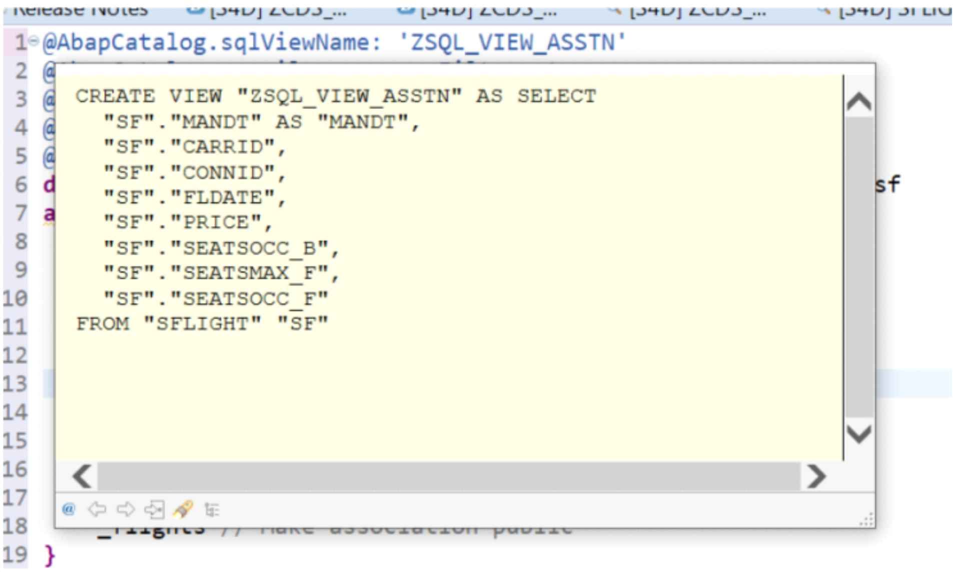
Right click and select 'Show SQL CREATE Statement;



```
@AbapCatalog.sqlViewName: 'ZSQL_VIEW_ASSTN'
@AbapCatalog.compiler.compareFilter: true
@AbapCatalog.preserveKey: true
@AccessControl.authorizationCheck: #CHECK
@EndUserText.label: 'CDS View with Association concept'
define view ZCDS_VIEW_ASSOCIATIONS as select from sflight as sf
association [1] to spfli as _flights
    on sf.carrid = _flights.carrid {
    //sf
    key sf.carrid,
    sf.connid,
    sf.fldate,
    sf.price,
    sf.seatsocc_b,
    sf.seatsmax_f,
    sf.seatsocc_f,

    _flights // Make associa
}
```

Undo Typing 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
Open With

As you can see, NO Join is created;

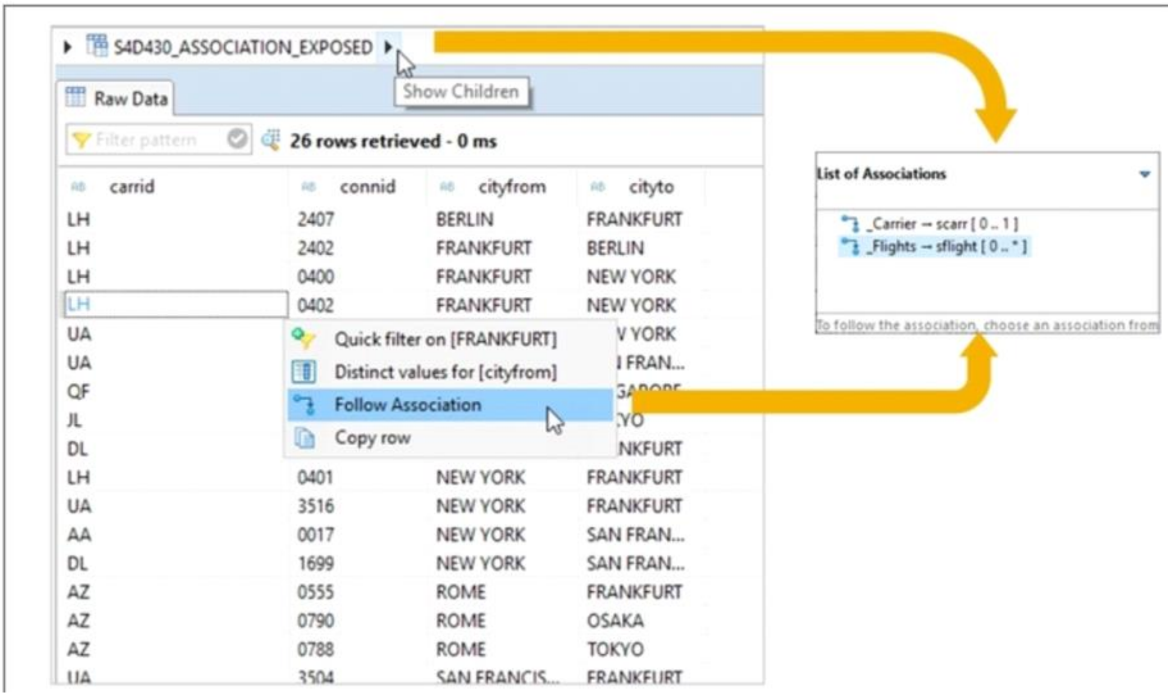


```
@AbapCatalog.sqlViewName: 'ZSQL_VIEW_ASSTN'
```

```
CREATE VIEW "ZSQL_VIEW_ASSTN" AS SELECT
  "SF"."MANDT" AS "MANDT",
  "SF"."CARRID",
  "SF"."CONNID",
  "SF"."FLDATE",
  "SF"."PRICE",
  "SF"."SEATSOCC_B",
  "SF"."SEATSMAX_F",
  "SF"."SEATSOCC_F"
FROM "SFLIGHT" "SF"
```

```
    _Flights // make association public
}
```

Exposed Associations in Data Preview


S4D430_ASSOCIATION_EXPOSED
Show Children
Raw Data
Filter pattern
Quick filter on [FRANKFURT]
Distinct values for [cityfrom]
Follow Association
Copy row
List of Associations

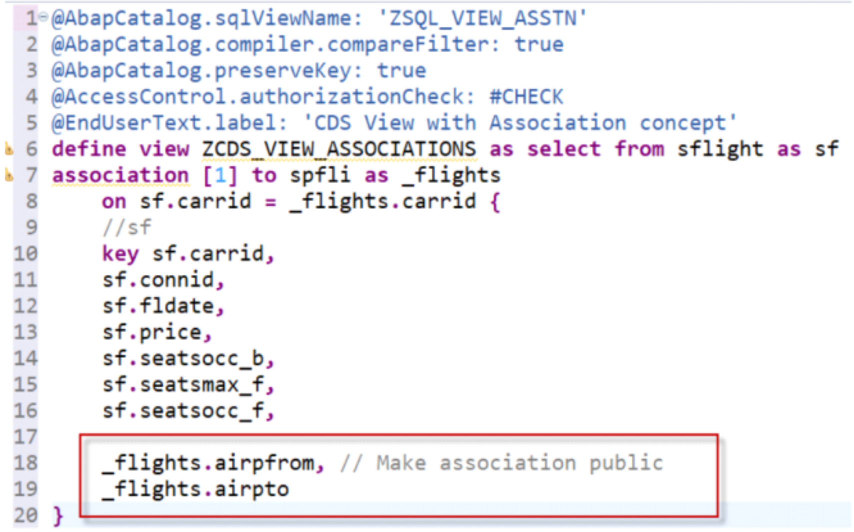
26 rows retrieved - 0 ms

carrid	connid	cityfrom	cityto
LH	2407	BERLIN	FRANKFURT
LH	2402	FRANKFURT	BERLIN
LH	0400	FRANKFURT	NEW YORK
LH	0402	FRANKFURT	NEW YORK
UA			V YORK
UA			I FRAN...
QF			GAPORE
JL			YO
DL			NKFURT
LH	0401	NEW YORK	FRANKFURT
UA	3516	NEW YORK	FRANKFURT
AA	0017	NEW YORK	SAN FRAN...
DL	1699	NEW YORK	SAN FRAN...
AZ	0555	ROME	FRANKFURT
AZ	0790	ROME	OSAKA
AZ	0788	ROME	TOKYO
UA	3504	SAN FRANCIS...	FRANKFURT

_Carrier → scarr [0 .. 1]
_Flights → sflight [0 .. *]

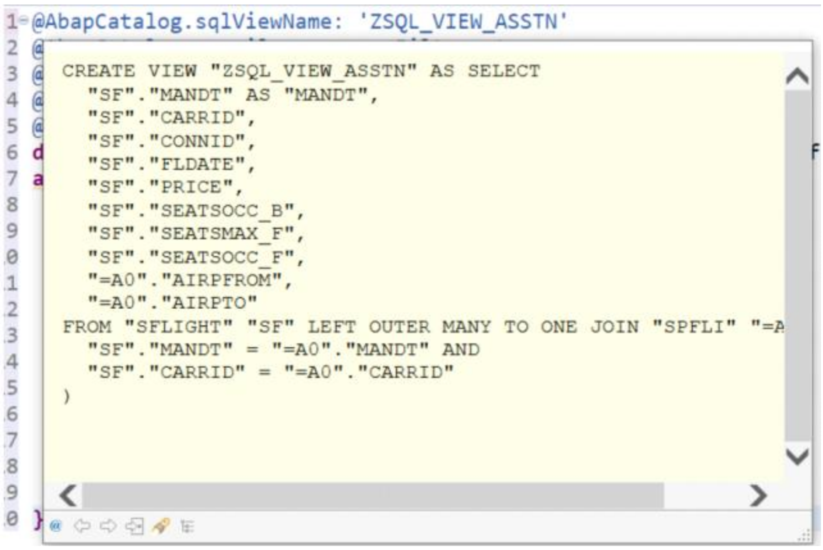
To follow the association, choose an association from

Now if we include fields from association:



```
@AbapCatalog.sqlViewName: 'ZSQL_VIEW_ASSTN'
@AbapCatalog.compiler.compareFilter: true
@AbapCatalog.preserveKey: true
@AccessControl.authorizationCheck: #CHECK
@EndUserText.label: 'CDS View with Association concept'
define view ZCDS_VIEW_ASSOCIATIONS as select from sflight as sf
association [1] to spfli as _flights
    on sf.carrid = _flights.carrid {
    //sf
    key sf.carrid,
    sf.connid,
    sf.fldate,
    sf.price,
    sf.seatsocc_b,
    sf.seatsmax_f,
    sf.seatsocc_f,

    _flights.airpfrom, // Make association public
    _flights.airpto
}
```

The join will be executed



```
@AbapCatalog.sqlViewName: 'ZSQL_VIEW_ASSTN'
```

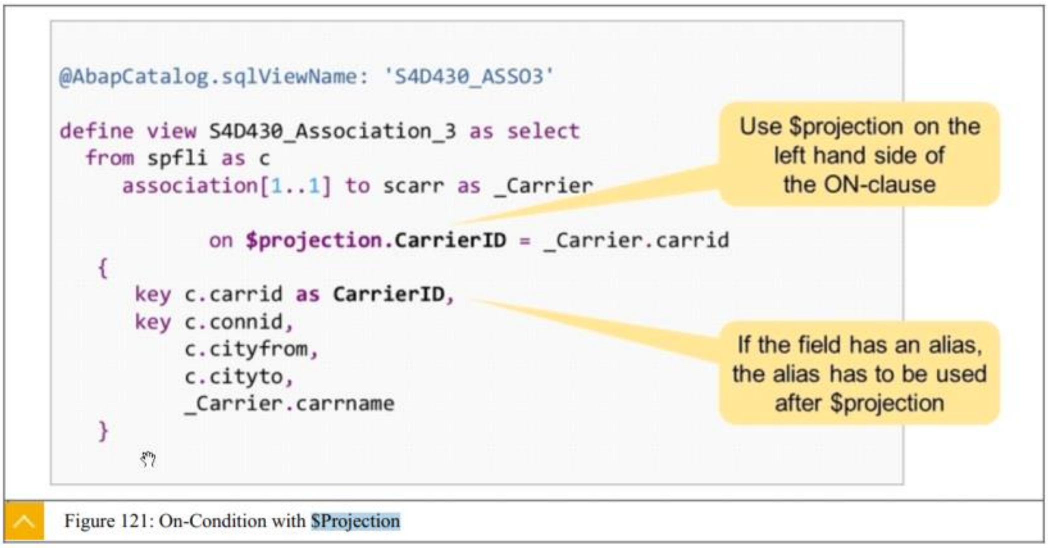
```
CREATE VIEW "ZSQL_VIEW_ASSTN" AS SELECT
  "SF"."MANDT" AS "MANDT",
  "SF"."CARRID",
  "SF"."CONNID",
  "SF"."FLDATE",
  "SF"."PRICE",
  "SF"."SEATSOCC_B",
  "SF"."SEATSMAX_F",
  "SF"."SEATSOCC_F",
  "=A0"."AIRPFROM",
  "=A0"."AIRPTO"
FROM "SFLIGHT" "SF" LEFT OUTER MANY TO ONE JOIN "SPFLI" "=A
  "SF"."MANDT" = "=A0"."MANDT" AND
  "SF"."CARRID" = "=A0"."CARRID"
)
```

$Projection in CDS Views

Definition

Can be used for 2 purposes:

- On Condition of an association:

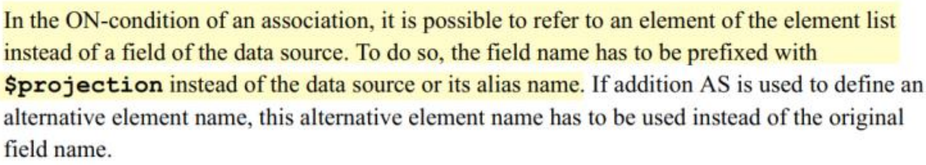


```
@AbapCatalog.sqlViewName: 'S4D430_ASSO3'

define view S4D430_Association_3 as select
  from spfli as c
     association[1..1] to scarr as _Carrier

            on $projection.CarrierID = _Carrier.carrid
   {
      key c.carrid as CarrierID,
      key c.connid,
          c.cityfrom,
          c.cityto,
          _Carrier.carrname
   }
```

Use $projection on the left hand side of the ON-clause

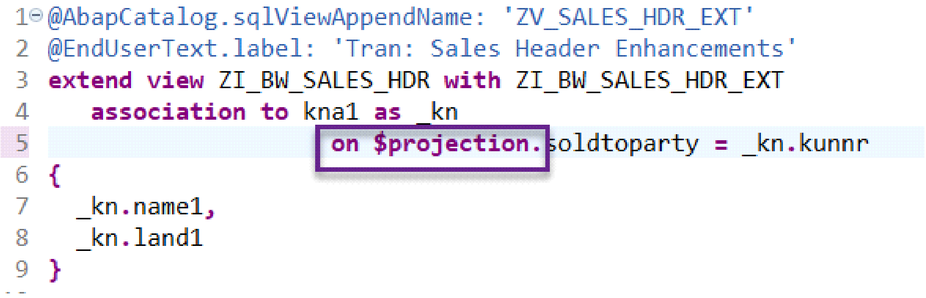
If the field has an alias, the alias has to be used after $projection

Figure 121: On-Condition with $Projection



In the ON-condition of an association, it is possible to refer to an element of the element list instead of a field of the data source. To do so, the field name has to be prefixed with **`$projection`** instead of the data source or its alias name. If addition AS is used to define an alternative element name, this alternative element name has to be used instead of the original field name.

- Refer to another CDS view as an alias
  Extension CDS view



```
@AbapCatalog.sqlViewAppendName: 'ZV_SALES_HDR_EXT'
@EndUserText.label: 'Tran: Sales Header Enhancements'
extend view ZI_BW_SALES_HDR with ZI_BW_SALES_HDR_EXT
   association to kna1 as _kn
                  on $projection.soldtoparty = _kn.kunnr
{
  _kn.name1,
  _kn.land1
}
```

ADSO (Advanced DSO)

ADSO	OneDrive Link

Q & A - Post Session

Q	
A	

Code Blocks

<table>
<tr><td>CDS View</td><td><pre>
@AbapCatalog.sqlViewName: 'ZV_SALES_ITM'
@AbapCatalog.compiler.compareFilter: true
@AbapCatalog.preserveKey: true
@AccessControl.authorizationCheck: #CHECK
@EndUserText.label: 'Tran: SD Sales Item'

@Analytics: {dataCategory: #FACT,
                 dataExtraction: {
                          enabled: true,
                          delta.byElement: {
                                name: 'LastChangedAt',
                                maxDelayInSeconds: 1800,
                                detectDeletedRecords: true
                          }
                       }
                 }
define view ZI_BW_SALES_ITEM
  as select from   vbap as p
    left outer join vbak as k on p.vbeln = k.vbeln
{
  key p.vbeln   as SalesDoc,
  key p.posnr   as SalesItem,
      p.matnr   as Material,

      @Semantics.quantity.unitOfMeasure: 'SalesUnit'
      p.kwmeng   as OrderQty,

      @Semantics.unitOfMeasure: true
      p.vrkme   as SalesUnit,
      @Semantics.systemDateTime.lastChangedAt: true
      k.upd_tmstmp as LastChangedAt
}
</pre></td></tr>
<tr><td>Sales Header</td><td><pre>
@AbapCatalog.sqlViewName: 'ZV_SALES_HDR'
@AbapCatalog.compiler.compareFilter: true
@AbapCatalog.preserveKey: true
@AccessControl.authorizationCheck: #CHECK
@EndUserText.label: 'Tran: SD Sales Header'
@Analytics: {
    dataCategory: #FACT,
    dataExtraction.enabled: true
    }
define view ZI_BW_SALES_HDR
  as select from I_SalesDocument as SD //I_SalesDocument
{

    //I_SalesDocument

    key SD.SalesDocument,
    SD.SDDocumentCategory,
    SD.SalesDocumentType,
    SD.CreatedByUser,
    SD.LastChangedByUser,
    SD.CreationDate,
    SD.CreationTime,
    SD.LastChangeDate,
    cast(cast(substring(cast(SD.LastChangeDateTime as abap.char(25)),1,14) as abap.numc(14)) as abap.dec (15, 0))   as   LastChangedAt,
    SD.SalesOrganization,
    SD.DistributionChannel,
    SD.OrganizationDivision,
    SD.SoldToParty,
    SD.SalesDocumentDate,
    SD.IncotermsClassification,
    SD.IncotermsLocation1,
    SD.FiscalYear,
    SD.FiscalPeriod
  }
</pre></td></tr>
<tr><td>Header Extn</td><td><pre>
@AbapCatalog.sqlViewAppendName: 'ZV_SALES_HDR_EXT'
@EndUserText.label: 'Tran: SD Sales Header Extension'
extend view ZI_BW_SALES_HDR with ZI_BW_SALES_HDR_EXT
association to kna1 as _kn
                 on SD.soldtoparty = _kn.kunnr
{
  _kn.name1,
  _kn.land1
}
</pre></td></tr>
<tr><td>Item Extn</td><td><pre>
@AbapCatalog.sqlViewAppendName: 'ZV_SALES_ITM_EXT'
@EndUserText.label: 'Tran: Sales Item Enhancements'
extend view Zi_Bw_Sales_Item with ZI_BW_SALES_ITEM_EXT
association to mara as _m
             on $projection.matnr = _m.matnr
</pre></td></tr>
</table>

	{
	  _m.mtart
	}