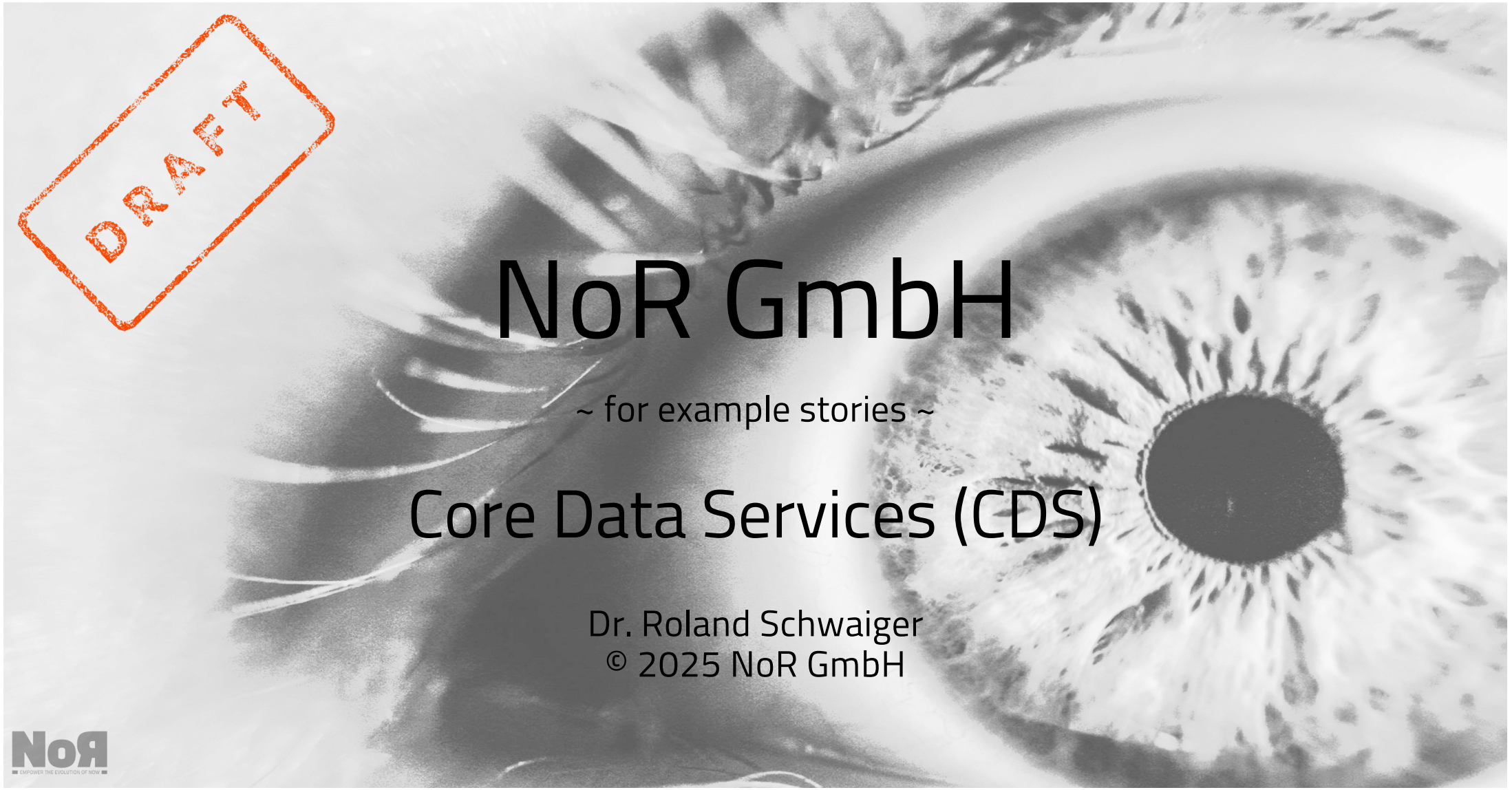


NoR GmbH

~ for example stories ~

Core Data Services (CDS)

Dr. Roland Schwaiger

Agenda

- System
- Part 2
  - CDS - Basics
    - VDM
    - CDS - Views
    - Data Preview
    - Association Navigation
    - SQL Console
    - Annotations
    - Dependencies
  - Creating CDS View
  - Inner Join, Projection, Selection
  - Annotations
  - Identifiers, translatable texts, methods
  - SQL Expressions
  - SQL Functions
  - Nested Views
  - Aggregations
  - Input parameter
  - Associations
  - ABAP Testing Program for CDS Usage
  - Access control
  - Table Functions (AMDP)

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Part 2

www.nor.gmbh office@nor.gmbh NoR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

CDS Data

CDS - Core Data Services

CDS is a collection of domain-specific languages and services for defining and consuming semantically rich data models

DDL = Data Definition Language
- Modeling and Retrieving Data
- Extension of native SQL

DQL = Data Query Language
- Use of CDS entities
- SQL Extension

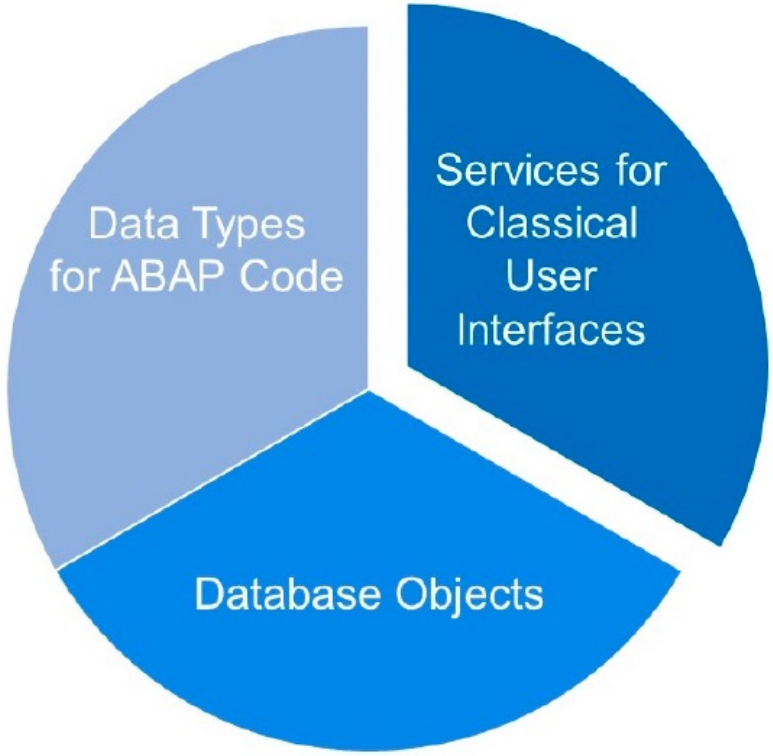
DCL = Data Control Language
- Defining Permissions for CDS Views
- Integrated into the classic SAP authorization concept

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

ABAP Dictionary

Tasks of the ABAP Dictionary

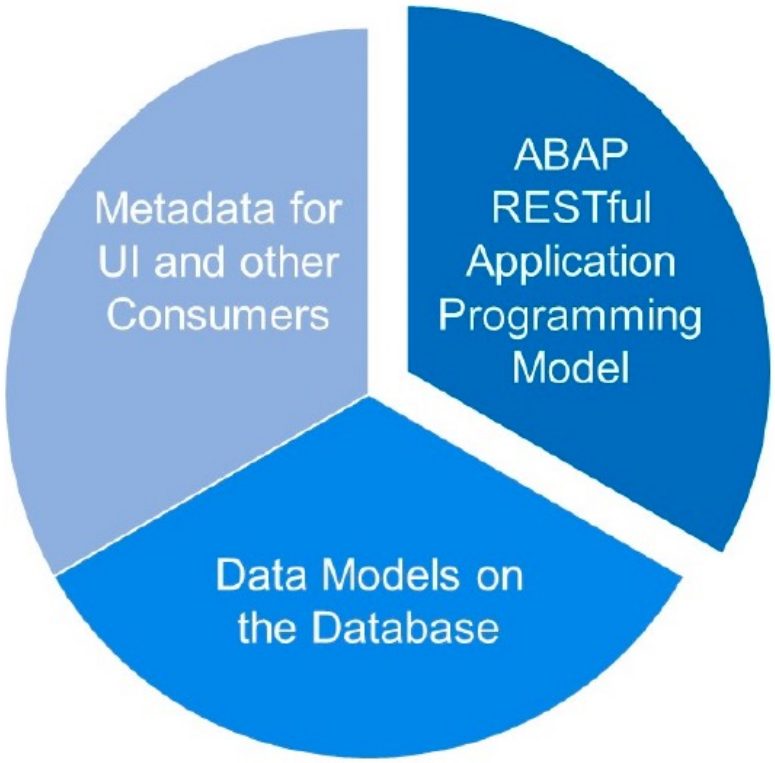
- **Defining Database Objects**
  - Database tables and views
  - Independent from actual database system
  - For use in ABAP SQL
- **Providing Data Types for ABAP Code**
  - Data elements, structures, table types
  - For use in data declarations and definition of interface parameters
  - globally visible in the system
- **Providing Services for Classical UIs**
  - Labels, documentation, value helps, and generated value checks, …



ABAP CDS

Tasks of the ABAP Core Data Services

- **Defining Data Models on the Database**
  - CDS objects (views and functions)
  - Including SQL logic for code-to-data
  - For use in ABAP SQL
- **Providing Metadata**
  - Semantically enriched data models
  - Support modern UI technologies (SAP Fiori, OData, and so on)
- **Support the ABAP RESTful Application Programming Model**
  - Special CDS objects
  - define the behavior of business objects



www.nor.gmbh office@nor.gmbh NoR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

CDS Objects

Development Object	Purpose	Remark
CDS Data Definition	Define a CDS Entity	CDS Entities are: • CDS View Entity • CDS DDIC-based Views (obsolete) • CDS Projection Views • CDS Abstract Entities • CDS Table Functions • CDS Custom Entities • CDS Hierarchies
CDS Access Control	Define a CDS Role	Control read access to data
CDS Metadata Extension	Add Metadata to a CDS Entity	Store metadata for a CDS entity outside its data definition
CDS Behavior Definition	Define the Behavior for a Business Object	Used for transactional processing in the ABAP RESTful Application Programming

Quick Reference
https://community.sap.com/t5/technology-blog-posts-by-members/cds-views-7-5x-quick-reference/ba-p/13978469

Name	Explanation
CDS DDIC-**Based View**	SQL access to data sources, technically implemented as a DB view.Introduced in ABAP 7.40 SP5. DEFINE VIEW ...
CDS Table Function	CDS entity implemented as an AMDP table function.Introduced in ABAP 7.50. DEFINE TABLE FUNCTION ...
CDS Abstract Entities	Defines an abstract CDS entity without a data source.Introduced in ABAP 7.53. DEFINE ABSTRACT ENTITY ...
CDS Hierarchies	CDS entity that realizes an SQL hierarchy on a data source.Introduced in ABAP 7.53. DEFINE HIERARCHY ...
CDS Custom Entities	Implemented as an ABAP class.Supports non-persistent data for UI scenarios.Introduced in ABAP 7.54. DEFINE CUSTOM ENTITY ...
CDS Projection Views	Access to a subset of a CDS view.Useful for service exposure with limited fields.Introduced in ABAP 7.54. DEFINE VIEW ENTITY ... AS PROJECTION
CDS View Entities	SQL access to data sources using the new CDS syntax.Modern replacement for DDIC-based views.Introduced in ABAP 7.55. DEFINE VIEW ENTITY ...

www.nor.gmbh office@nor.gmbh NoR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

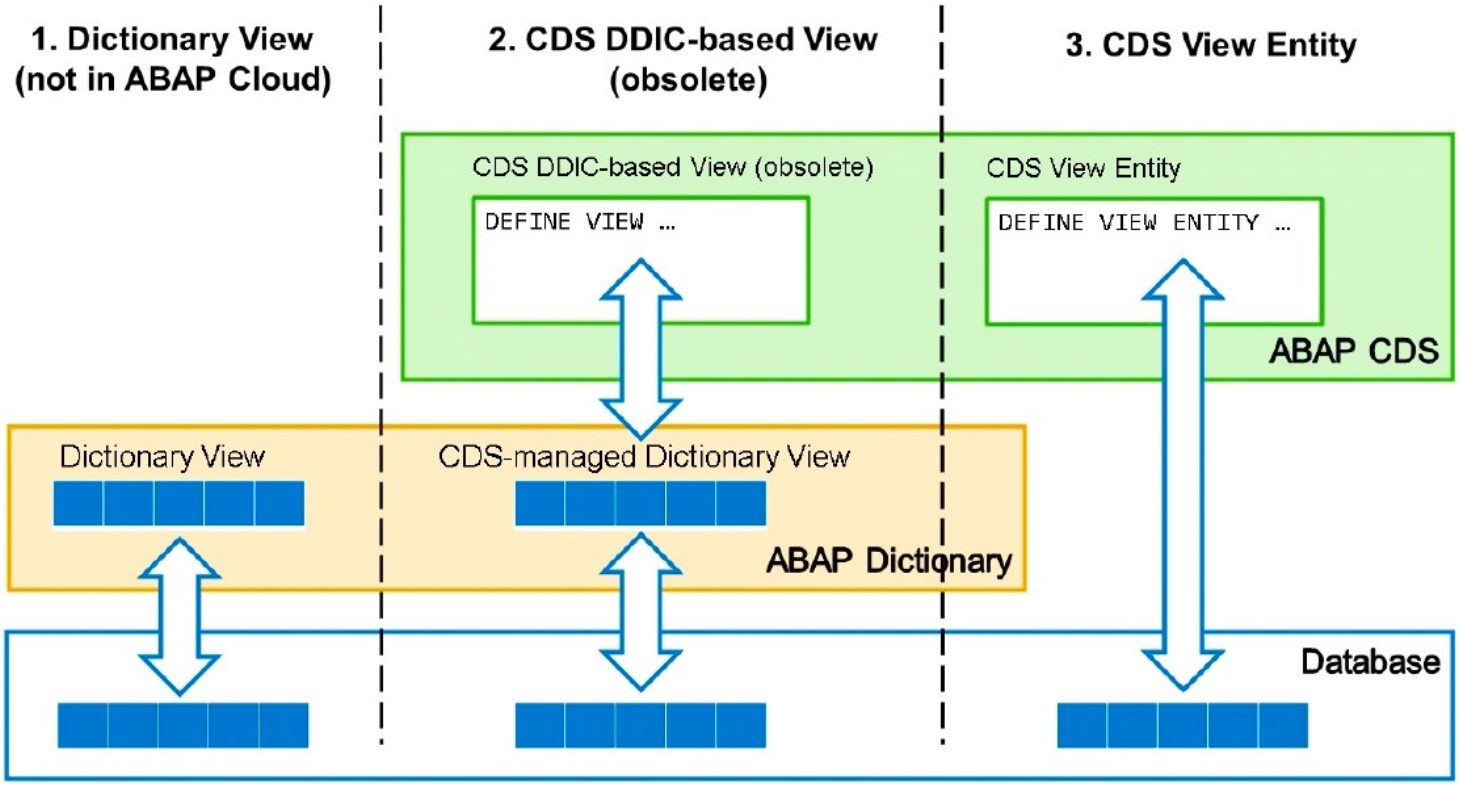
NoR
EMPOWER THE EVOLUTION OF NOW

CDS Subobjects in the Repository

- Access Control (DCL)
  - Must be maintained in Eclipse ADT
- Data Definition (DDL)
  - Must be maintained in Eclipse ADT
  - Defines CDS View or CDS Table Function
- Data Query Language (DQL)
  - Similar to Standard SQL-92
- Metadata Extension (@ Annotation)
  - Must be maintained in Eclipse ADT
  - Annotated CDS View or CDS Table Function

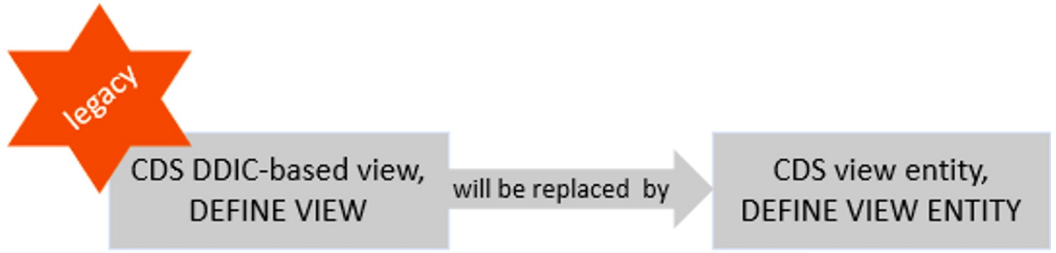
www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

CDS Views



www.nor.gmbh office@nor.gmbh NoR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

SAP CDS Innovations
CDS view entity



With ABAP release 7.55, a new type of CDS view is available: in official terminology, it's called CDS view entity. And it has come to replace the "classic" CDS DDIC-based views that have been around for years.

- No additional ABAP Dictionary view is created on activation.
- Improved performance during view activation.
- Optimization and simplification of syntax.
- Stricter syntax checks indicate problematic situations more explicitly, for example, annotation checks.

Naming conventions SAP

Name of the DDL Source:

- Max. 30 characters
- System-wide unique (Customer namespace!)
- Always uppercase

Name of the CDS View:

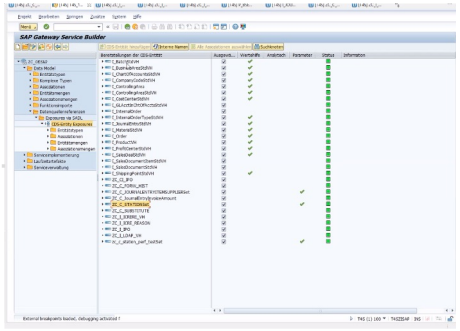
- Max. 30 characters
- System-wide unique (Customer namespace!)
- Unique in the ABAP Dictionary (global ABAP data type)
- Case-insensitive (upper/lowercase is not distinguished)
- Can differ from the name of the DDL source (not recommended!)

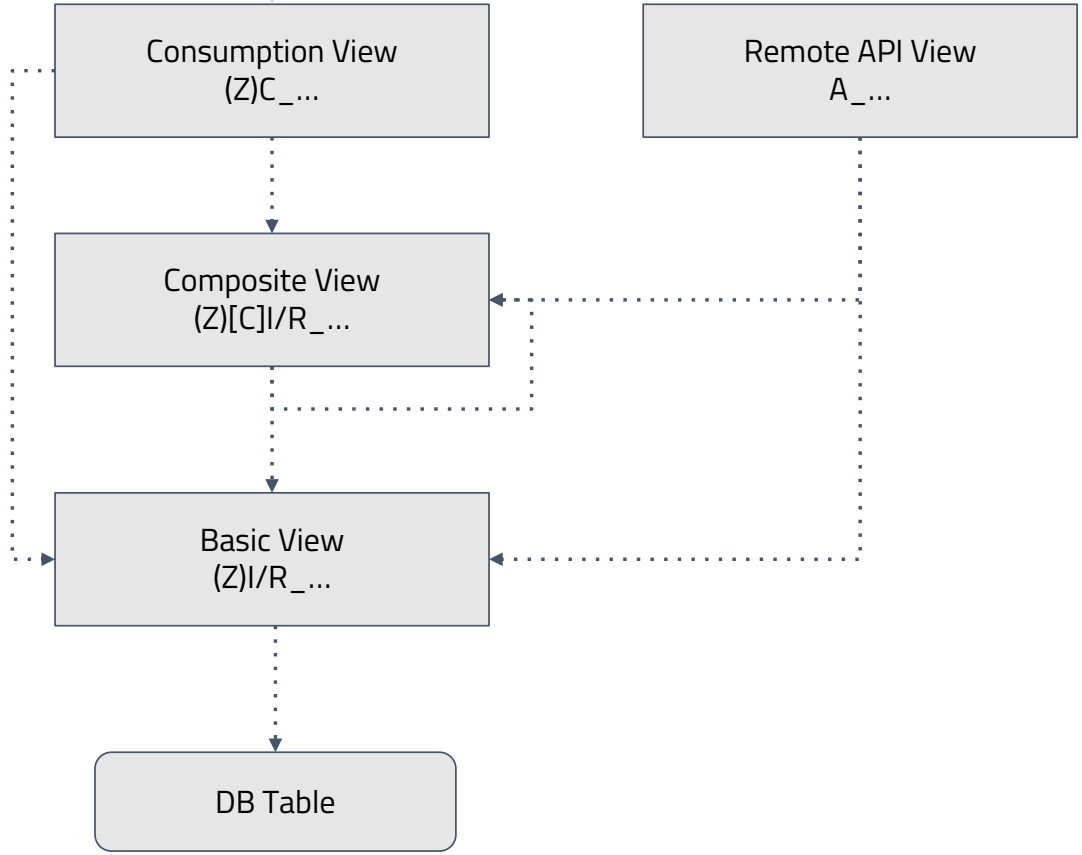
Name of the SQL View:

- Max. 16 characters (standard Dictionary object)
- System-wide unique (Customer namespace!)
- Unique in the ABAP Dictionary (global ABAP data type)
- Case-insensitive (will be transferred in uppercase)
- Must differ from the name of the CDS View

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

VDM - Virtual Data Model





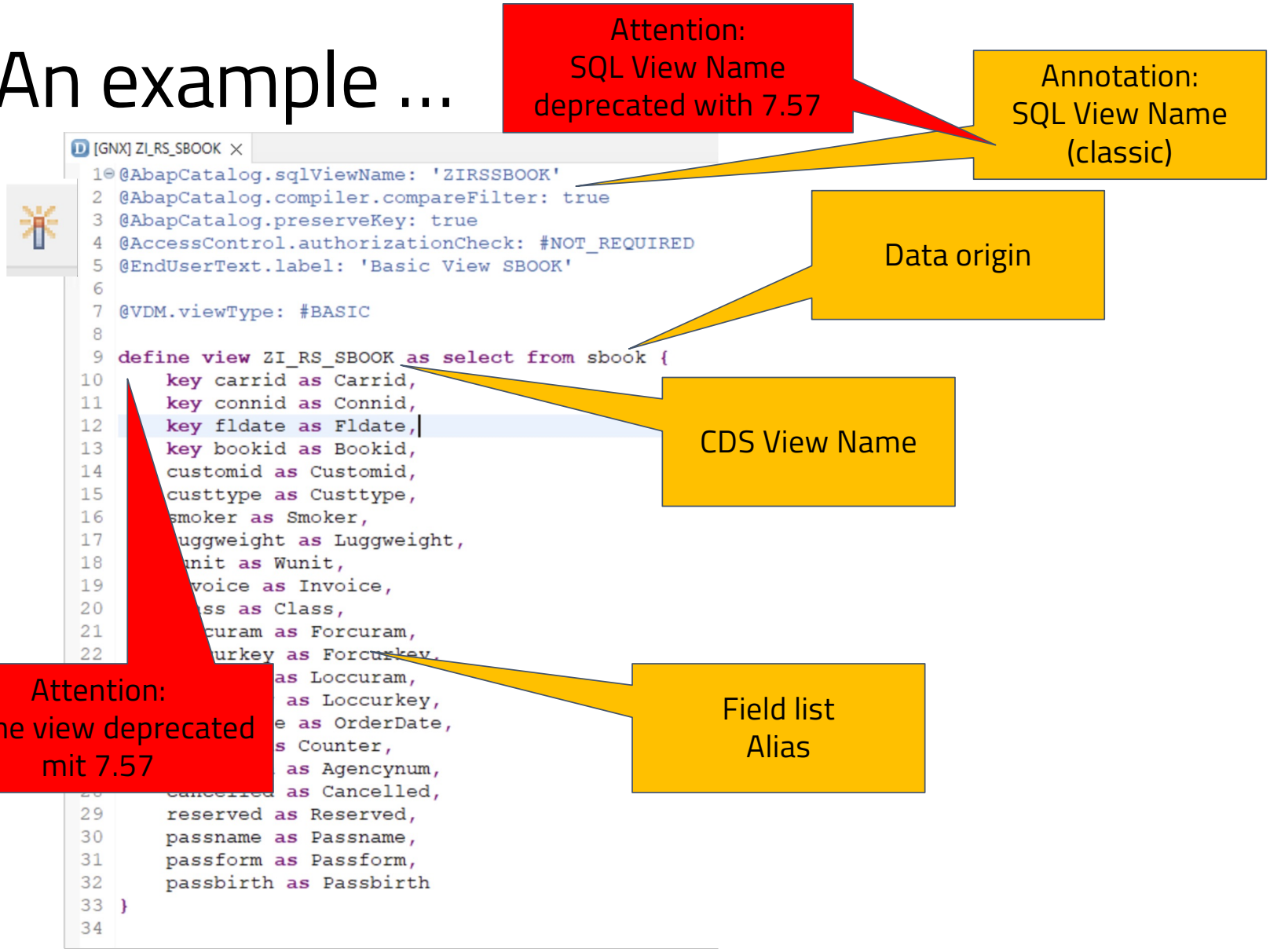
www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

VDM cont.

Identification
@VDM.viewType: #BASIC
@VDM.viewType: #COMPOSITE
@VDM.viewType: #TRANSACTIONAL
@VDM.viewType: #CONSUMPTION
@VDM.viewType: #CONSUMPTION @ObjectModel.transactional ProcessingDelegated:true
@VDM.lifecycle.contract.type: #PUBLIC_REMOTE_API
@VDM.private: true
@VDM.viewType: #EXTENSION
@VDM.viewExtension: true

www.nor.gmbh office@nor.gmbh NOR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

An example ...



[GNX] ZI_RS_SBOOK

```
@AbapCatalog.sqlViewName: 'ZIRSSBOOK'
@AbapCatalog.compiler.compareFilter: true
@AbapCatalog.preserveKey: true
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'Basic View SBOOK'

@VDM.viewType: #BASIC

define view ZI_RS_SBOOK as select from sbook {
    key carrid as Carrid,
    key connid as Connid,
    key fldate as Fldate,
    key bookid as Bookid,
    customid as Customid,
    custtype as Custtype,
    smoker as Smoker,
    luggweight as Luggweight,
    unit as Wunit,
    voice as Invoice,
    ss as Class,
    curam as Forcuram,
    urkey as Forcurkey,
    as Loccuram,
     as Loccurkey,
    e as OrderDate,
    s Counter,
     as Agencynum,
     as Cancelled,
    reserved as Reserved,
    passname as Passname,
    passform as Passform,
    passbirth as Passbirth
}
```

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

@Annotations

@AbapCatalog.sqlViewName: 'XXXX' — Auto creates SQL view corresponding to CDS view with given name 'XXXX'. (CDS DDIC-based View)

@AbapCatalog.preserveKey: true — Preserves the CDS key fields, note the Keyword 'Key' next to SCARR. Preserve key true ignores defaults keys from the database table/view and preserves keys defined explicitly in the View. Possible Values — true/false

@AccessControl.authorizationCheck: #CHECK — Tells whether an authorization check should be performed or not, how can we perform an authorization check in CDS? Check upcoming blogs. Possible Values — #CHECK, #NOT_ALLOWED, #NOT_REQUIRED, #PRIVLEDGE_ONLY.

@EndUserText.label: 'CARDINALITY' — It's a free text which provides info to the end user. This has its own advantage in FIORI, UI5 apps etc. can be used as field catalogues. 40 Characters is the max length.

@AbapCatalog.compiler.compareFilter: true — Defines the evaluation of filter conditions in path expressions of the CDS view. Possible Values — true/false

@ClientDependent: true — Defines Client handling when Open SQL is used to access the CDS View. Is optional and the default is "true". Possible Values — true/false.

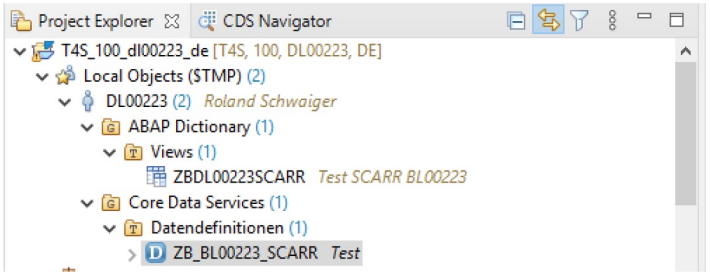
www.nor.gmbh office@nor.gmbh NoR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

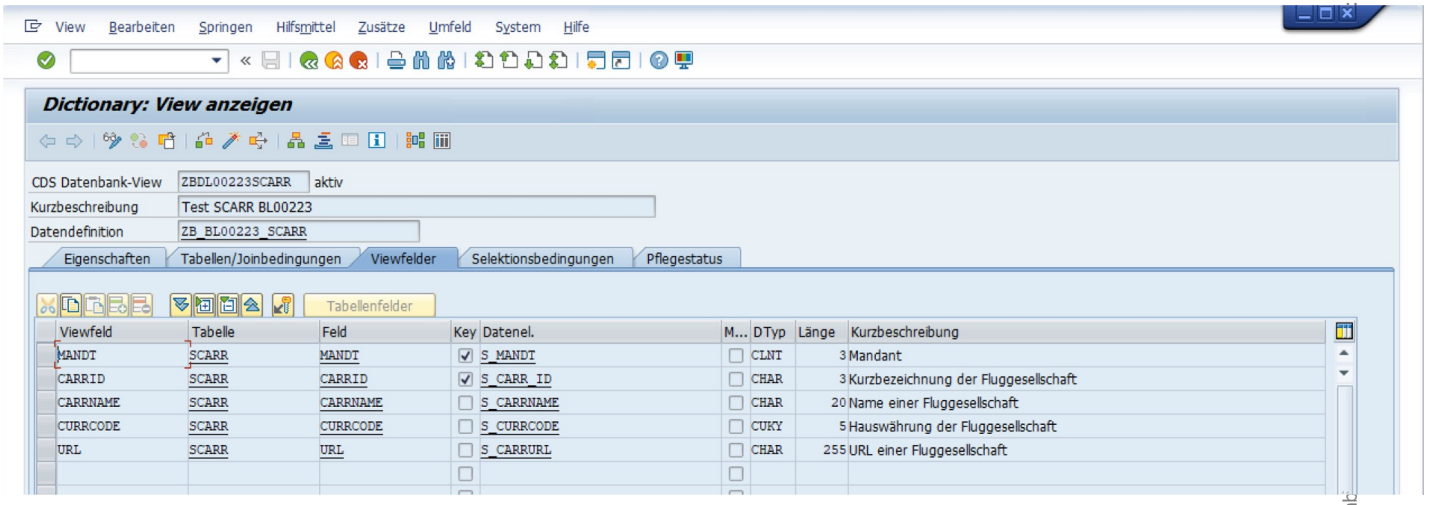
Annotations

	ABAP Annotations (Evaluated by ABAP runtime)	Framework-specific (Evaluated by Framework like OData, VDM, UI, Analytics, …) **
View Annotations	@AbapCatalog.viewEnhancementCategory @AccessControl.authorizationCheck @EndUserText.label	@VDM.viewType @UI.headerInfo.typeName @Analytics.query: true
Element Annotations	@Semantics.Amount.currencyCode @EndUserText.label	@UI.hidden: true @AnalyticsDetails.query.axis @Semantics.address.city
Parameter Annotations*	@Environment.systemField @EndUserText.label	…
Extension Annotations*	@AbapCatalog.sqlViewAppendName	…
Function Annotations*	@ClientDependent @EndUserText.label	…

www.nor.gmbh office@nor.gmbh NoR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

CDS DDIC-based View … SQL-View in DDIC





Views

SQL-View:

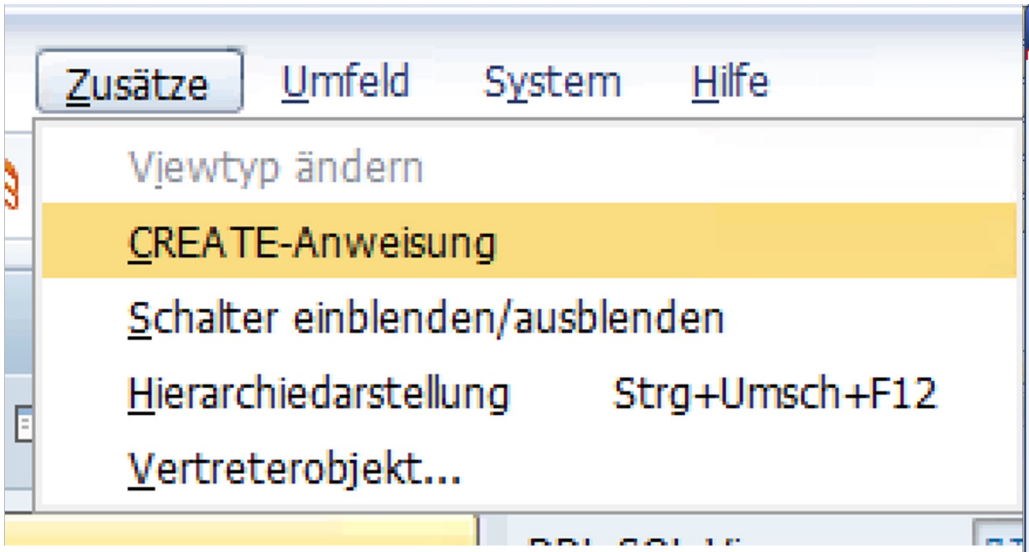
- Is Repository Object
- Is Database Object
- Is ABAP data type
- No additional semantics

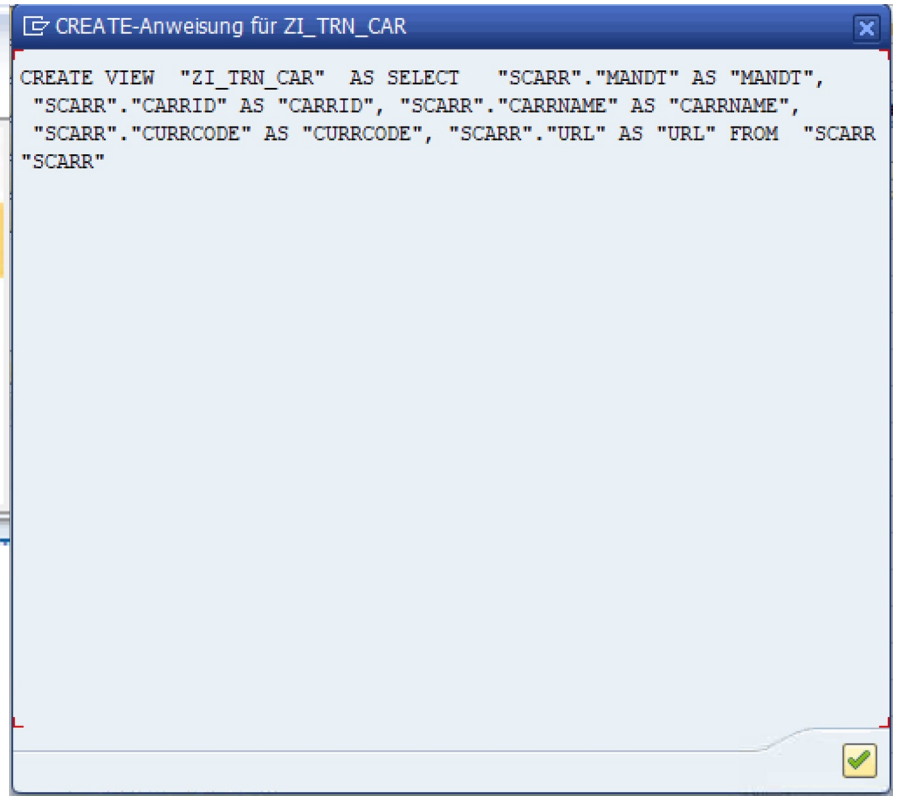
CDS-View:

- Not available in the repository
- Not in DB
- Is ABAP data type
- Additional semantics (annotations)

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

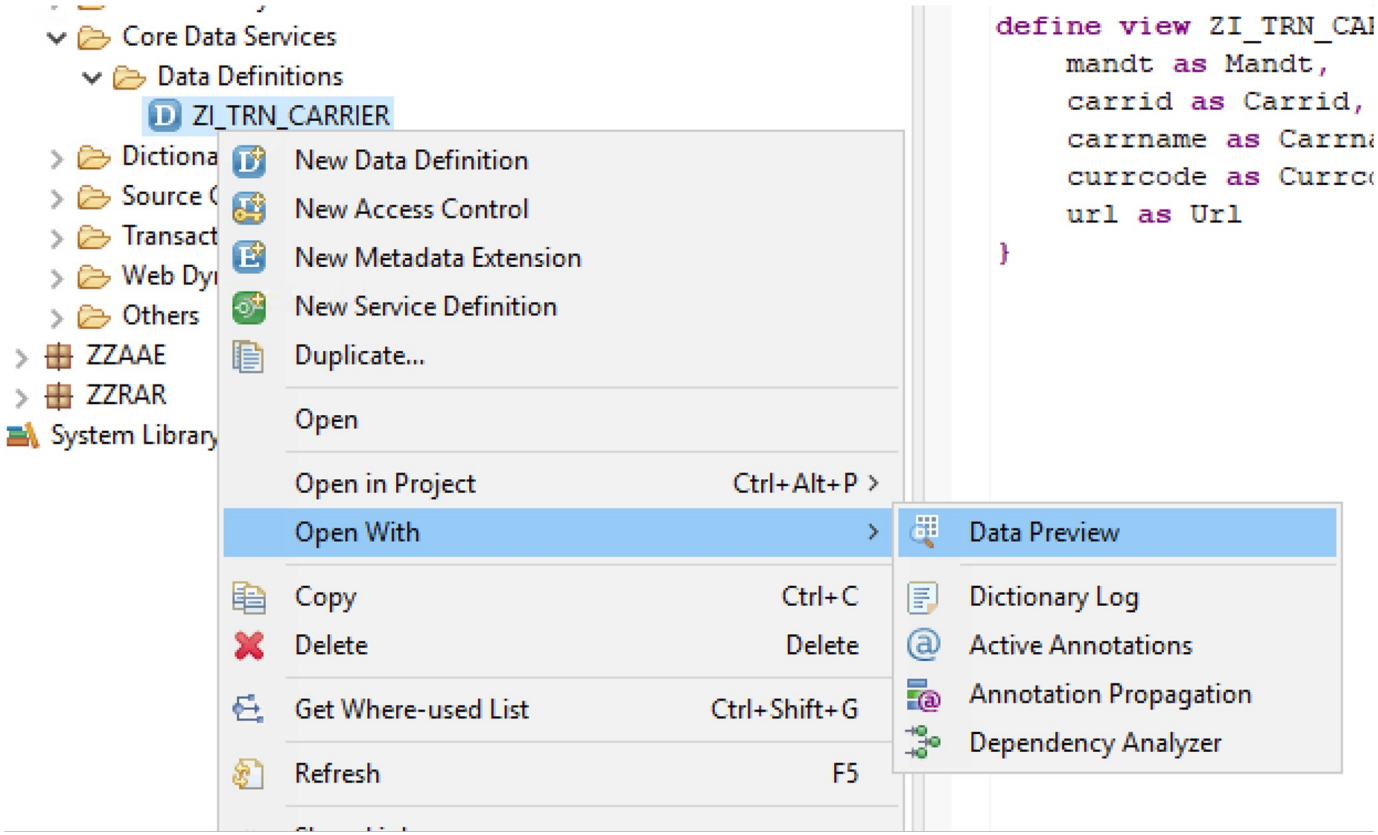
CREATE Instruction





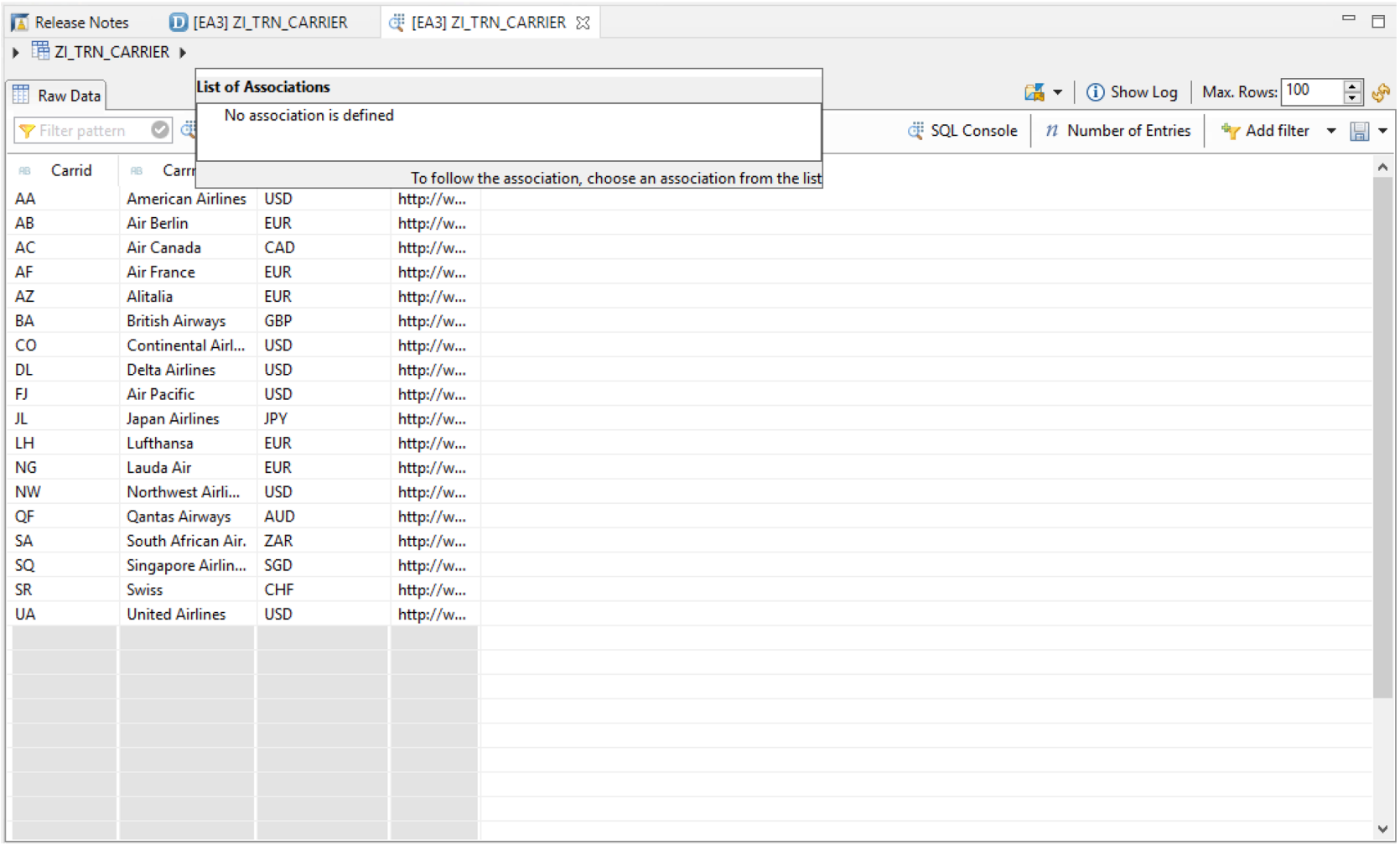
www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Data Preview



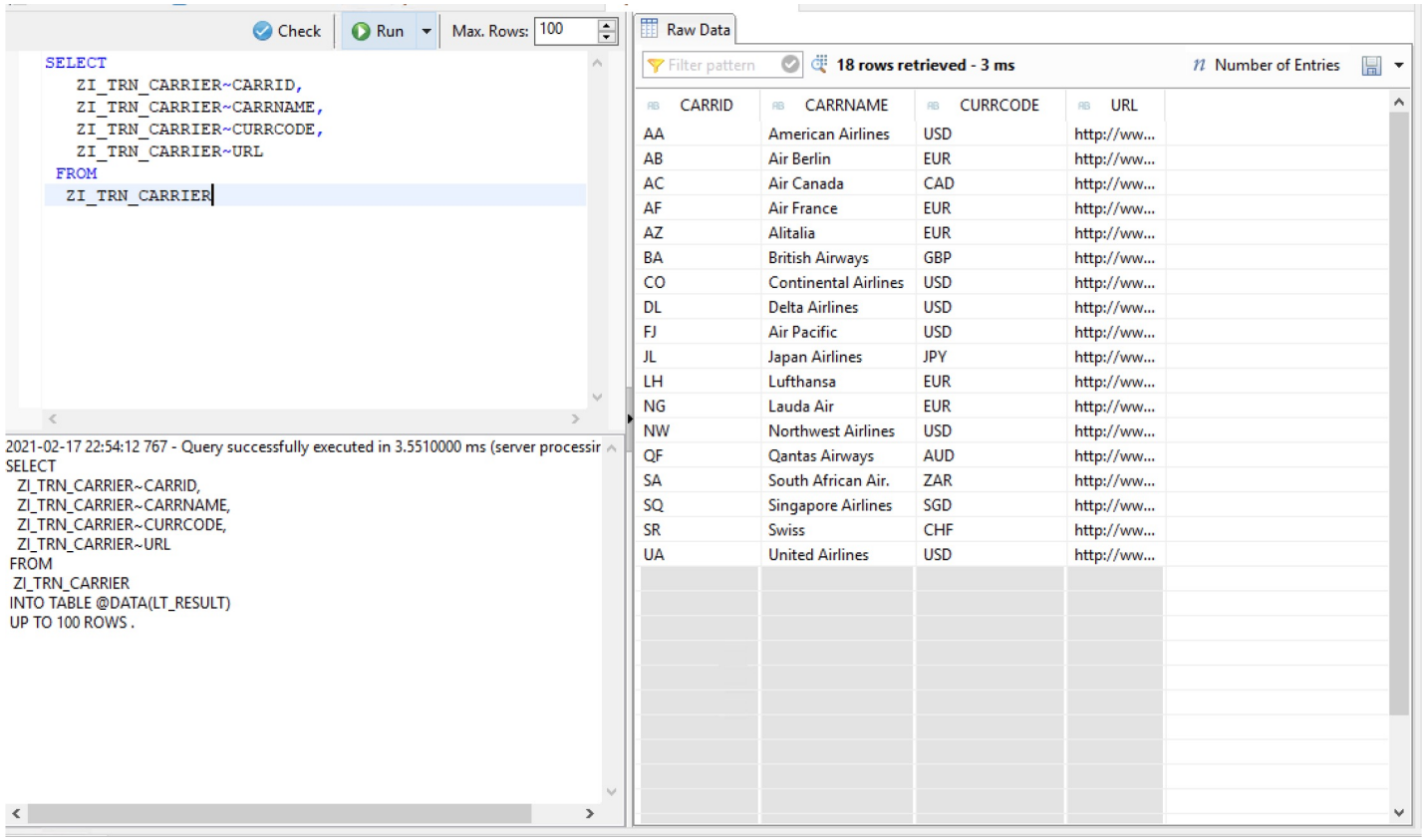
www.nor.gmbh office@nor.gmbh NOR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Association Navigation



www.nor.gmbh office@nor.gmbh NOR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

SQL Console



www.nor.gmbh office@nor.gmbh NoR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

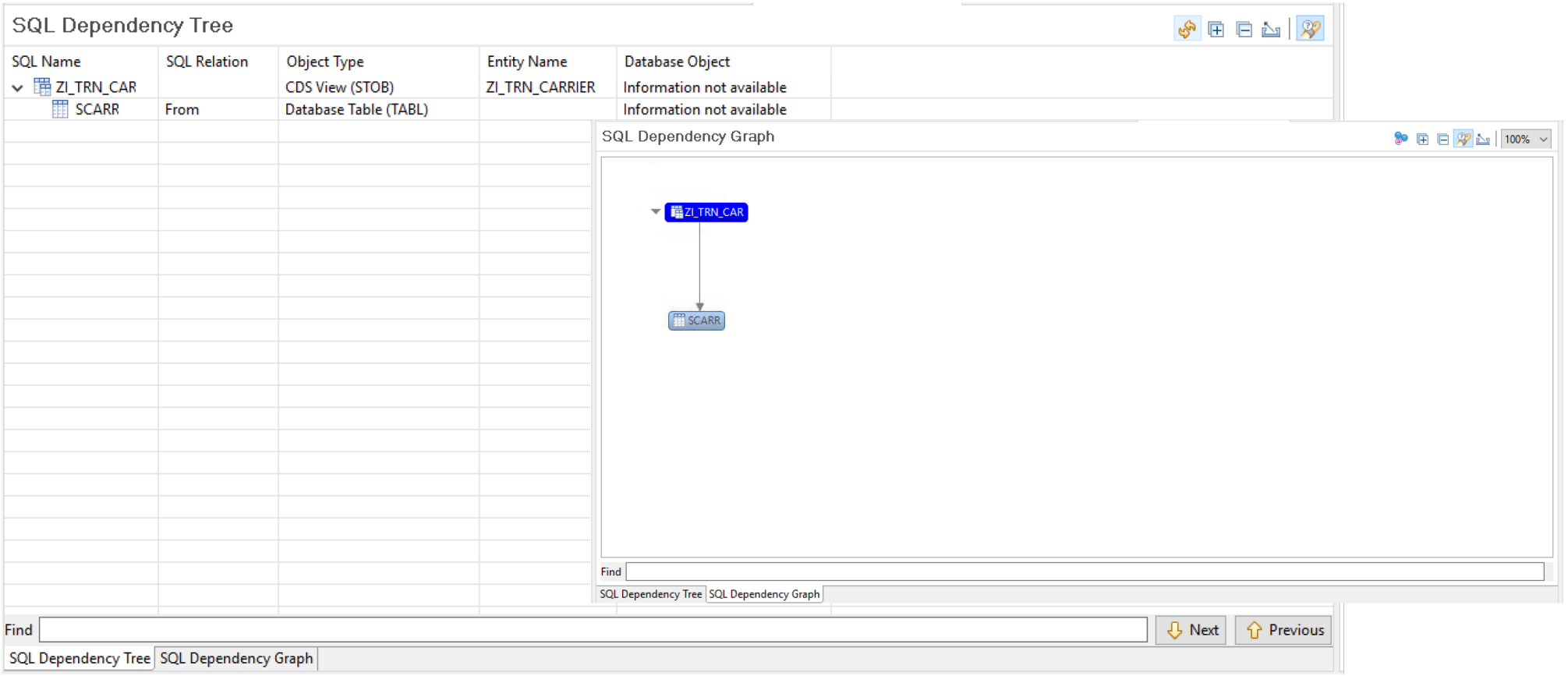
Active Annotations

Active Annotations for Entity ZI_TRN_CARRIER

type filter text

Annotated Elements	Annotation Value	Translated Text	Origin Data Source	Origin Data Element
ZI_TRN_CARRIER				
Entity annotations				
@AbapCatalog				
sqlViewName	'ZI_TRN_CAR'			
compiler				
compareFilter	true			
preserveKey	true			
@AccessControl				
authorizationCheck	#CHECK			
@EndUserText				
label	Carrier			
Carrid				
@EndUserText				
quickInfo	Airline Code		SCARR	S_CARR_ID
label	Airline		SCARR	S_CARR_ID
heading	ID		SCARR	S_CARR_ID
Carrname				

SQL Dependency Tree



www.nor.gmbh office@nor.gmbh NOR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

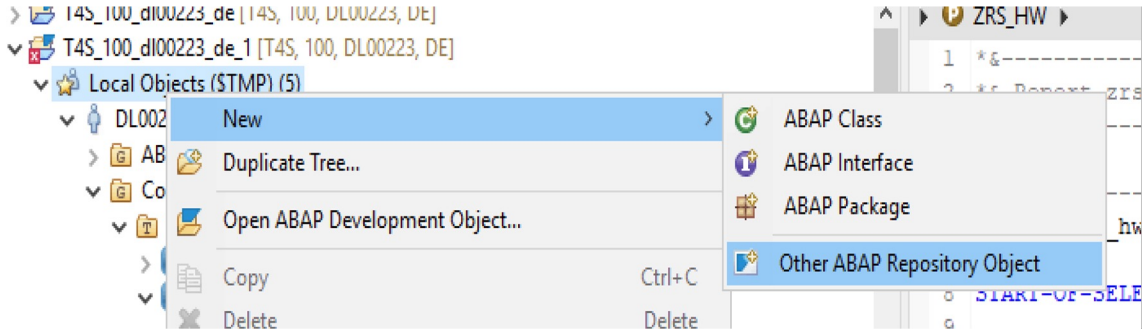
Understanding CDS View and Using it in Open SQL

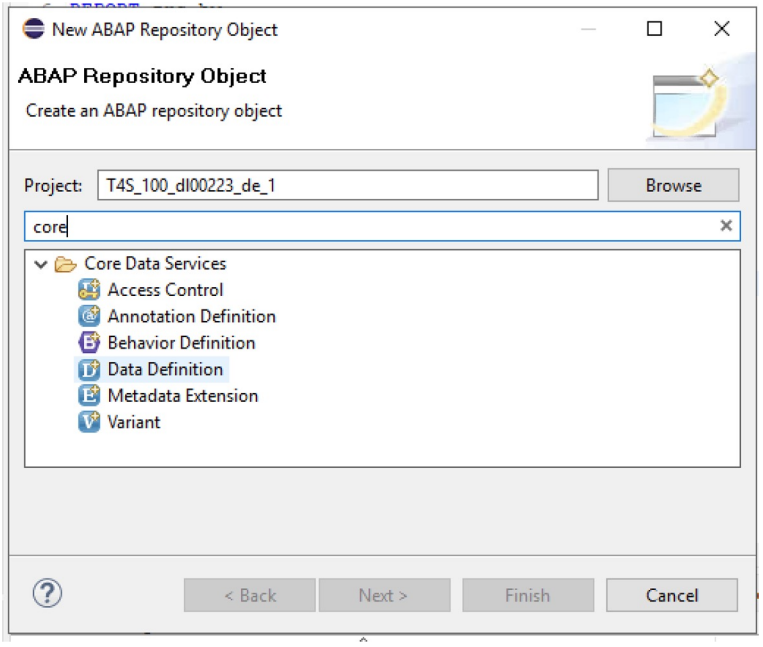
2

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

CDS Data Definition

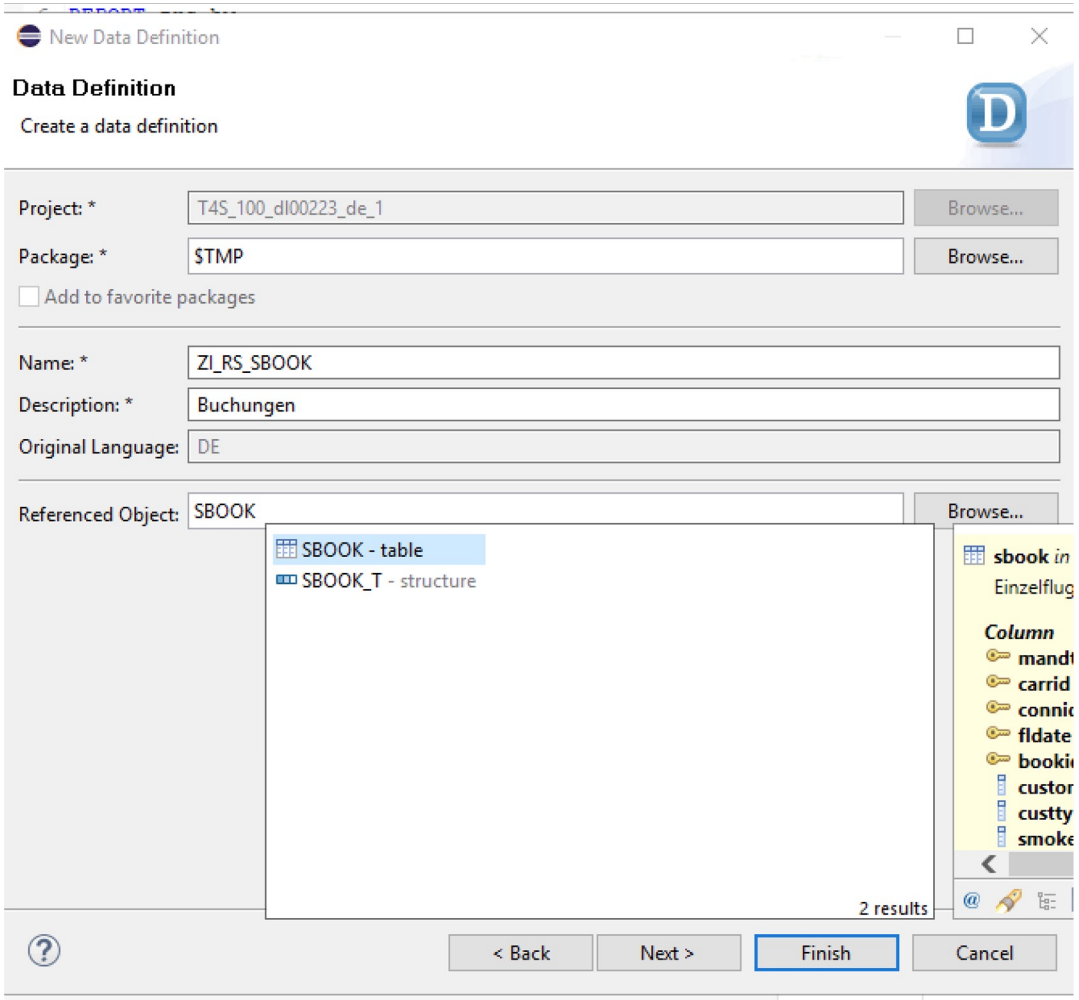
Creating CDS Data Definition (classic)





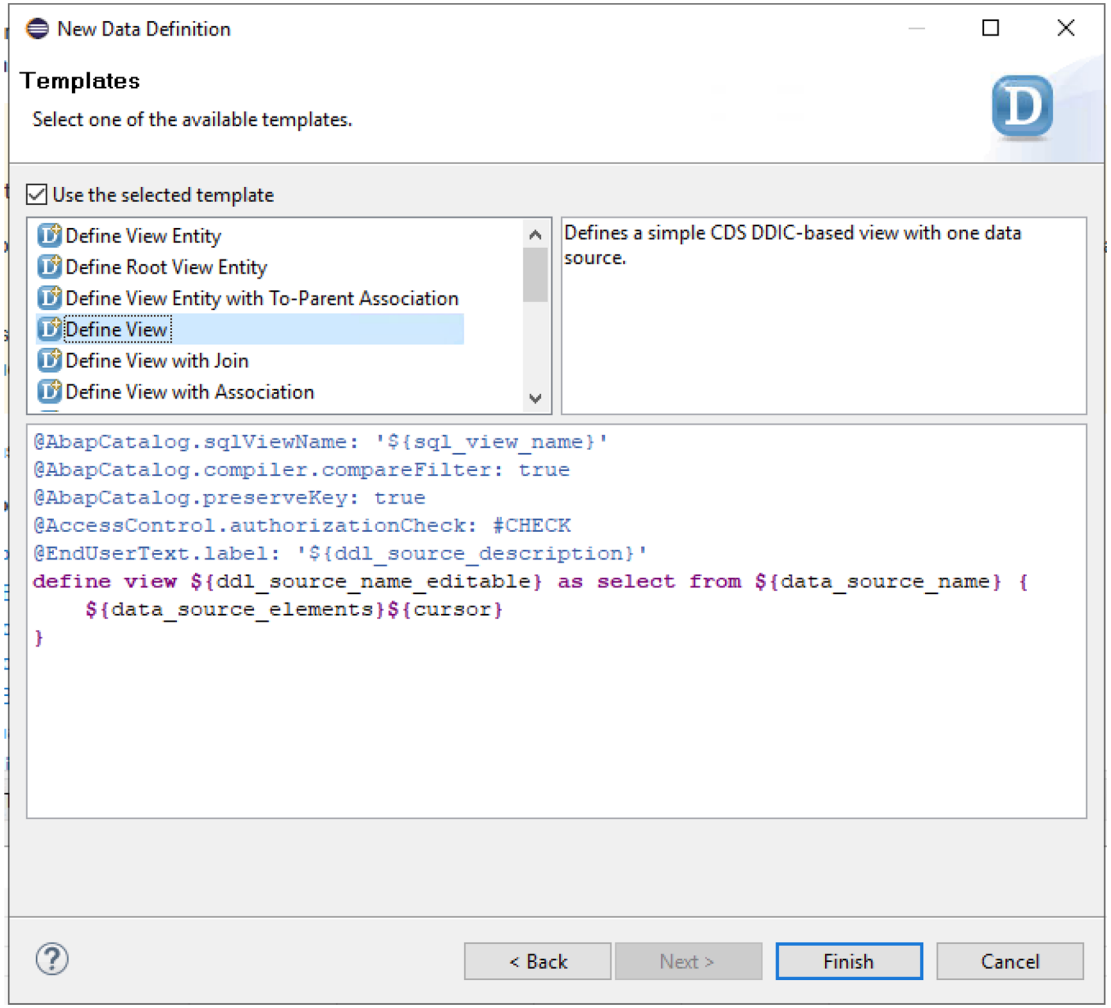
www.nor.gmbh office@nor.gmbh NOR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Create Data Definition



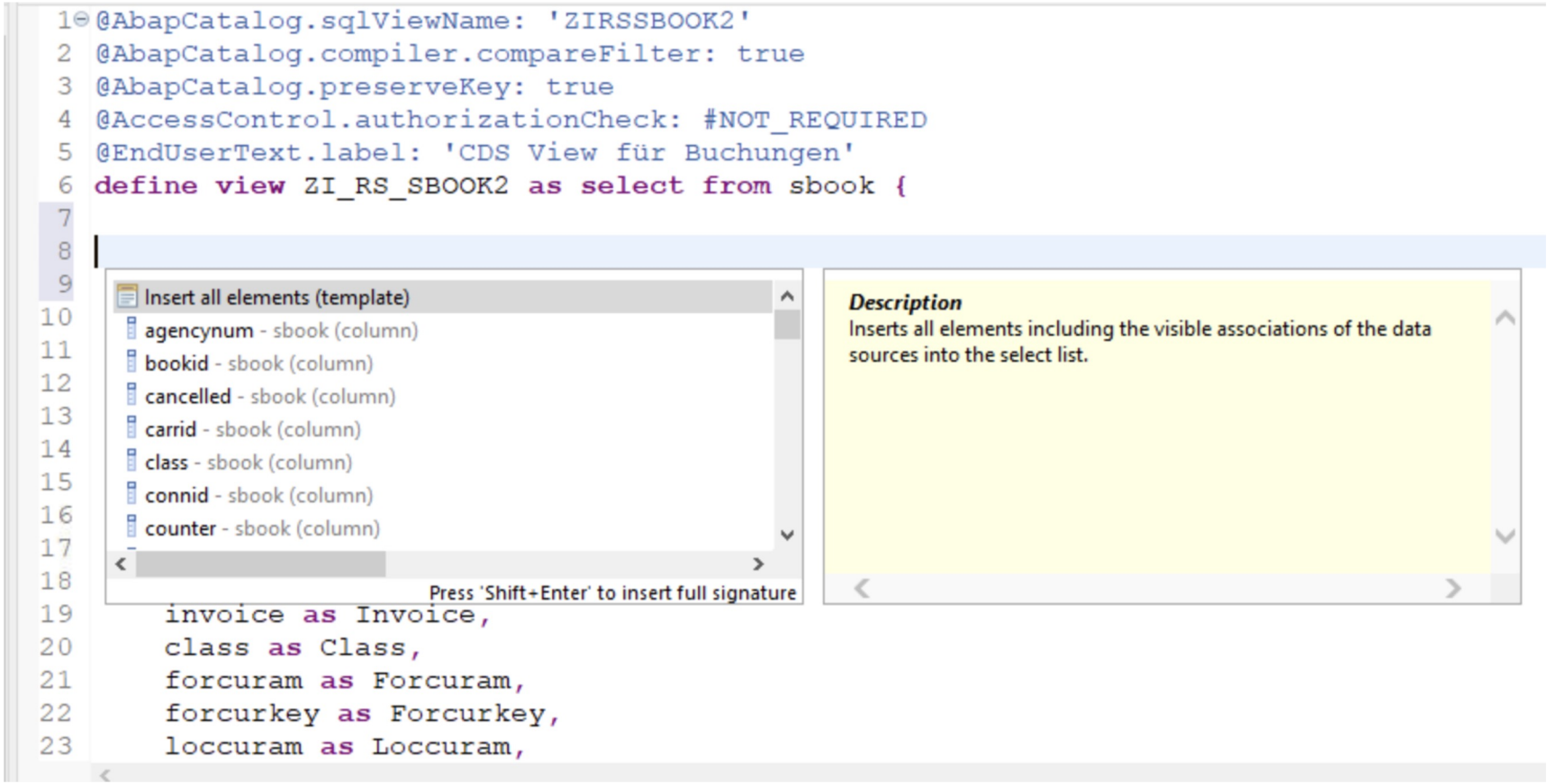
www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Data Definition Template
Define View (classic), Define View Entity



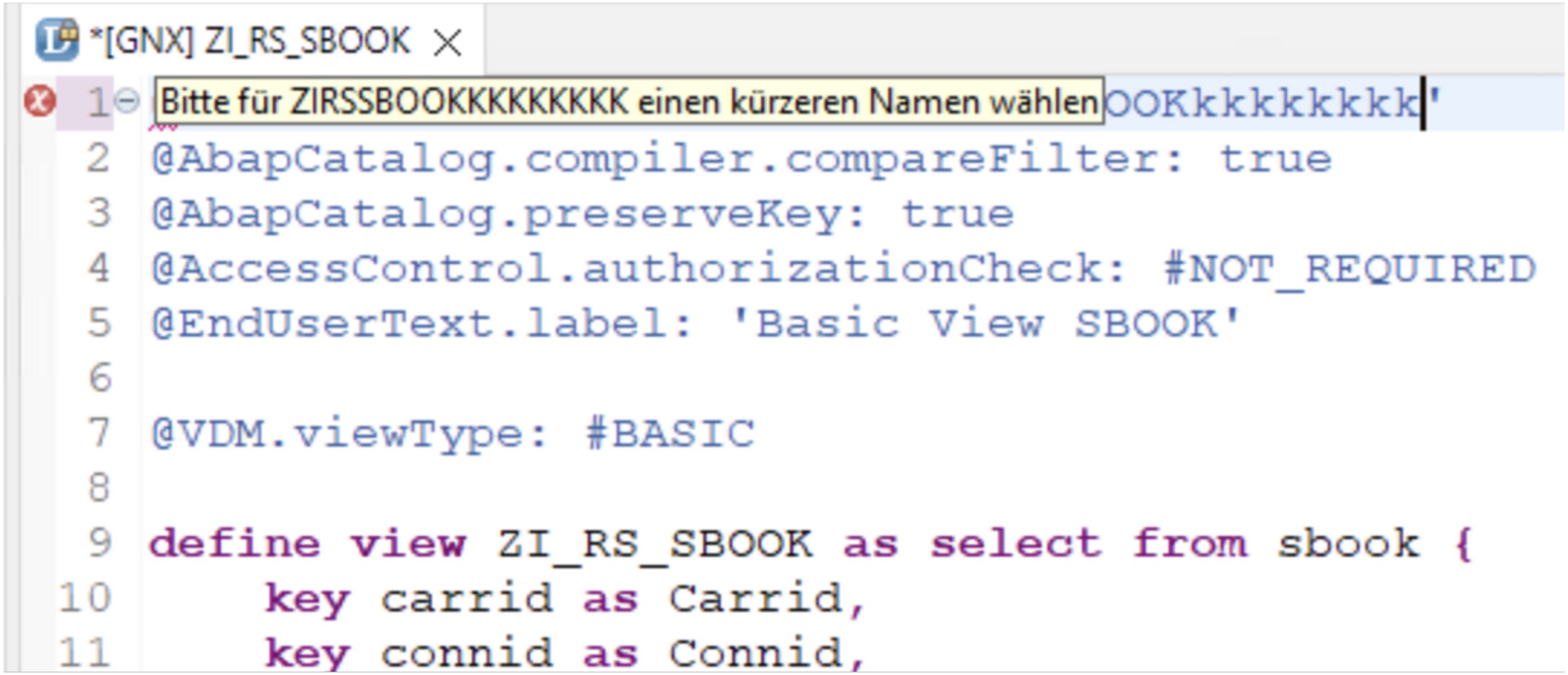
www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Inserting Data Definition Elements



www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

SQL View Name, like in DDIC



www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Data Definition

```
@AbapCatalog.sqlViewName: 'ZIRSSBOOK2'
@AbapCatalog.compiler.compareFilter: true
@AbapCatalog.preserveKey: true
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'CDS View für Buchungen'
define view ZI_RS_SBOOK2 as select from sbook {
    key carrid as Carrid,
    key connid as Connid,
    key fldate as Fldate,
    key bookid as Bookid,
    customid as Customid,
    custtype as Custtype,
    smoker as Smoker,
    luggweight as Luggweight,
    wunit as Wunit,
    invoice as Invoice,
    class as Class,
    forcuram as Forcuram,
    forcurkey as Forcurkey,
    loccuram as Loccuram,
    loccurkey as Loccurkey,
    order_date as OrderDate,
    counter as Counter,
```

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Create a legacy SQL-View based CDS

3

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

DEFINE VIEW -> DEFINE VIEW ENTITY

DEFINE VIEW

- Die *alte* Syntax für CDS-Views (bis ABAP 7.54 üblich).
- Erstellt im ABAP Dictionary ein **SQL View** (technischer Name über @AbapCatalog.sqlViewName).
- Es entsteht **zwei Objekte**:
  - CDS-View (Repository-Objekt im Paket)
  - SQL View im Dictionary (für DB und Open SQL)

DEFINE VIEW ENTITY

- Die *neue* Syntax (ab ABAP 7.55).
- Erstellt ein einziges Objekt: die **View Entity** im ABAP Dictionary.
- Kein zusätzliches SQL View Objekt mehr nötig → **einfacher, moderner, performanter**.
- Kein @AbapCatalog.sqlViewName mehr nötig.
- **Nur noch ein Objekt** im Dictionary (keine Verdoppelung).
- Zukunftssicher: **neue CDS-Features** (z. B. **CDS Table Functions mit View Entities**, RAP, Annotations) funktionieren nur mit view entity.

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

CDS View Create

4

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

CDS View Inner Join, Projection and Selection

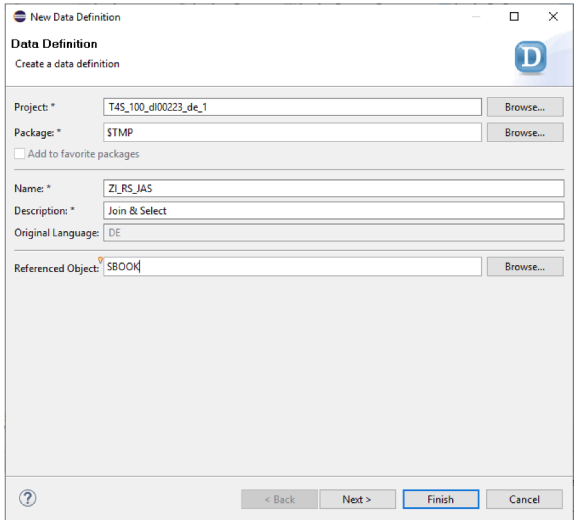
View Inner Join, Projection and Selection

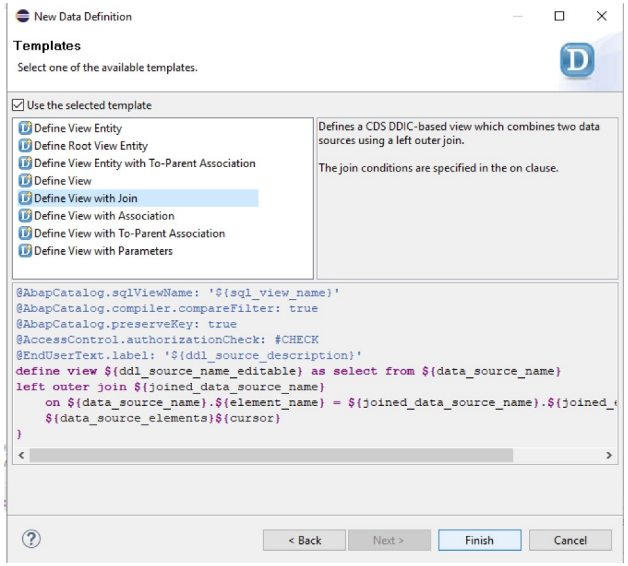
- An **INNER JOIN** combines rows from two (or more) data sources where the **join condition matches**.
  - If no match → row is excluded.
- **Projection** = selecting only the required fields instead of all columns.
- **Selection** = restricting data with WHERE clause.

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Inner Join, Projection and Selection - Example

```
define view entity ZC_FlightOverview
  as select from sflight as F
    inner join scarr   as C
      on F.carrid = C.carrid
{
  key F.carrid,
  key F.connid,
      F.fldate,
      F.price,
      C.carrname
}
where
      F.fldate between '20250101' and '20251231'
  and F.price > 200
```





www.nor.gmbh office@nor.gmbh NOR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

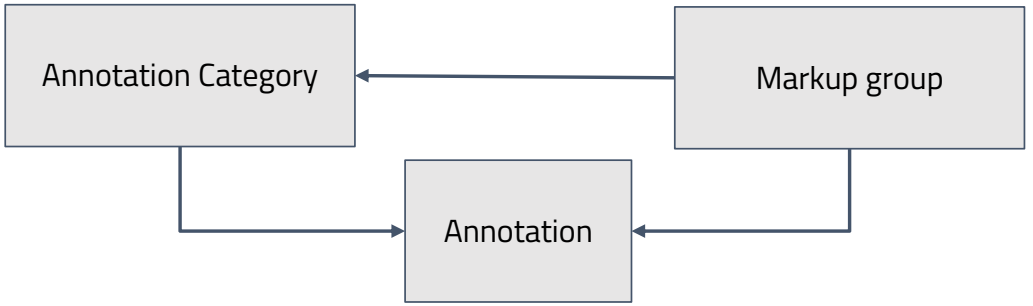
CDS View Definition Inner Join, Projection and Selection

5

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

DDL Source and Annotations

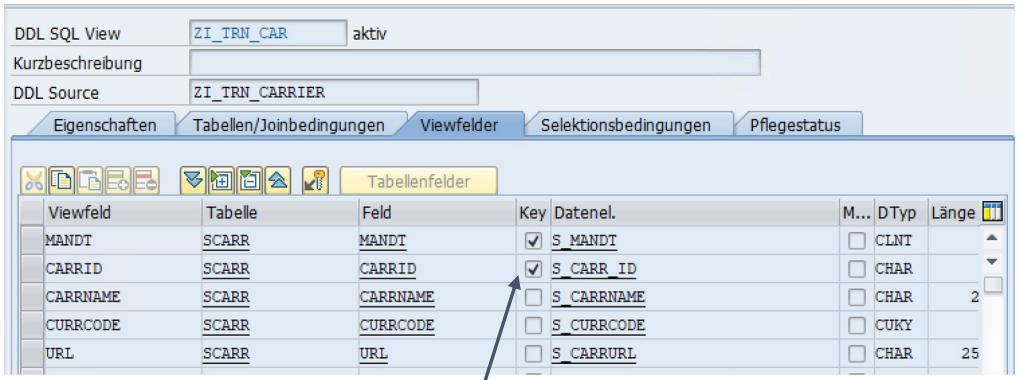
Annotationen



- View-Annotationen
  - @AbapCatalog.sqlViewName
- Element-Annotationen
  - @Semantics.unitOfMeasure
- Parameter-Annotationen
  - @Environment.systemField
- Erweiterungs-Annotationen
  - @AbapCatalog.sqlViewAppendName
- Funktions-Annotationen
  - @ClientDependent
- FIORI
  - @UI
- OData
  - @OData

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Annotations



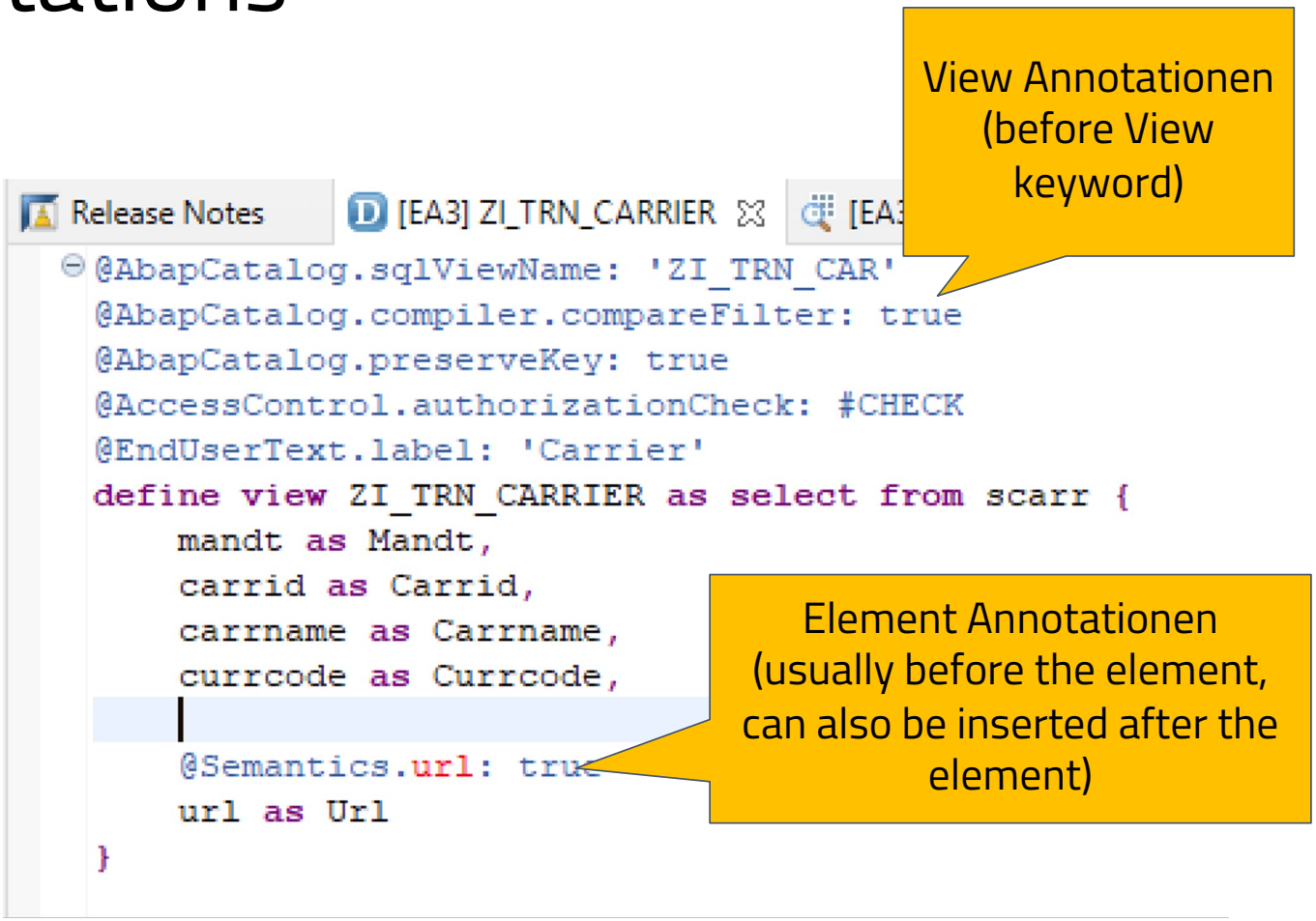
Properties

- Enriching the definition with metadata
- Starts with the @ sign
- Scope
  - View
  - Element
  - Parameter
  - …

```
@AbapCatalog.sqlViewName: 'ZI_TRN_CAR'
@AccessControl.authorizationCheck: #CHECK
@EndUserText.label: 'Carrier'
define view ZI_TRN_CARRIER as select from scarr
{
    key mandt as Mandt,
    carrid as Carrid,
    carrname as Carrname,

    @Semantics.currencyCode: true
    currcode as Currcode,

    @Semantics.url: true
    url as Url
}
```

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

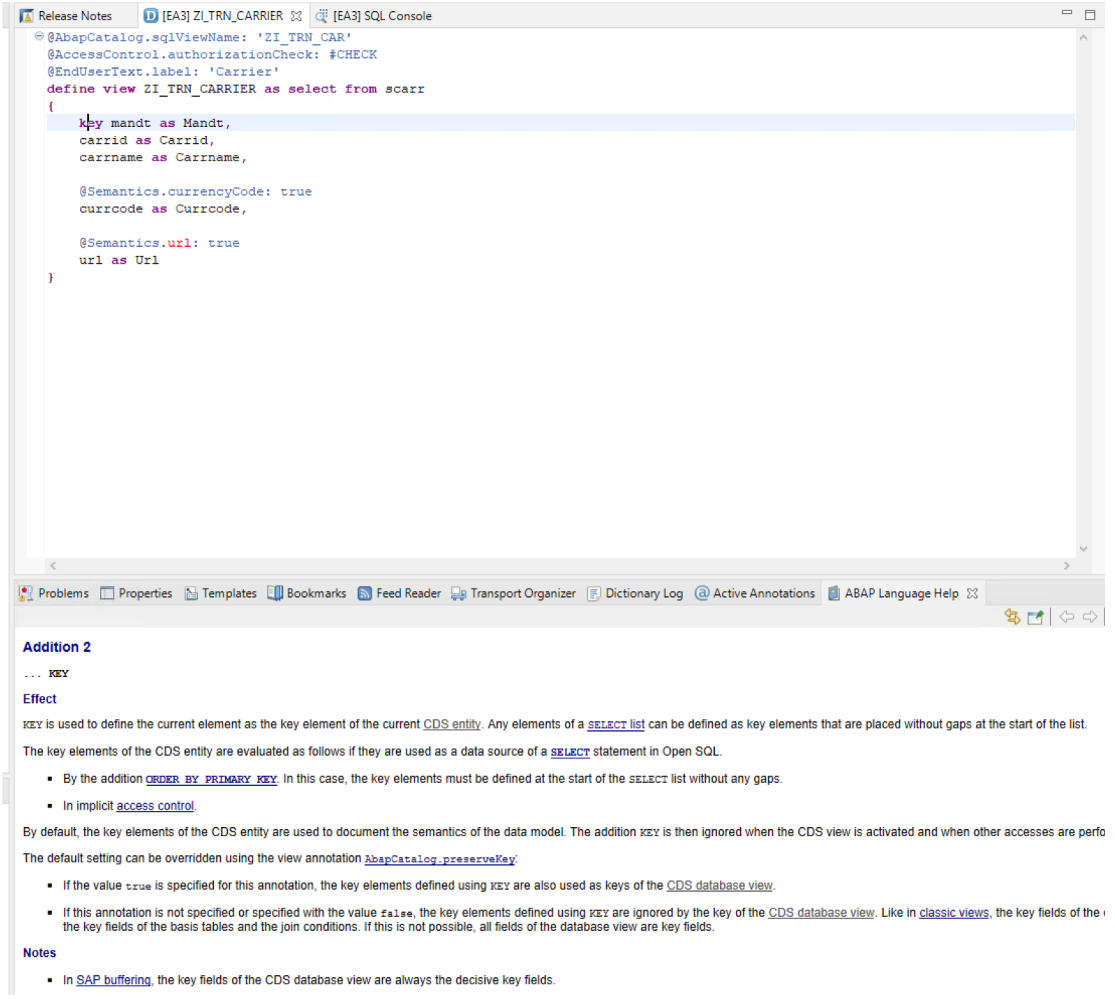
Annotations



www.nor.gmbh office@nor.gmbh NOR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Help - there is also as before ;-)

F1



Important annotations ABAP-View

Client handling controlled in Open SQL

```
@ClientHandling.type: #INHERITED
@ClientHandling.algorithm: #AUTOMATED
```

Access control

```
@AccessControl.authorizationCheck: #CHECK
```

Controlling the evaluation of filtered associations

```
@AbapCatalog.compiler.CompareFilter: true
```

Buffering SQL View in Open SQL

```
@AbapCatalog.Buffering.type: #GENERIC
@AbapCatalog.Buffering.numberOfKeyFields: 1
@AbapCatalog.Buffering.status: #ACTIVE
```

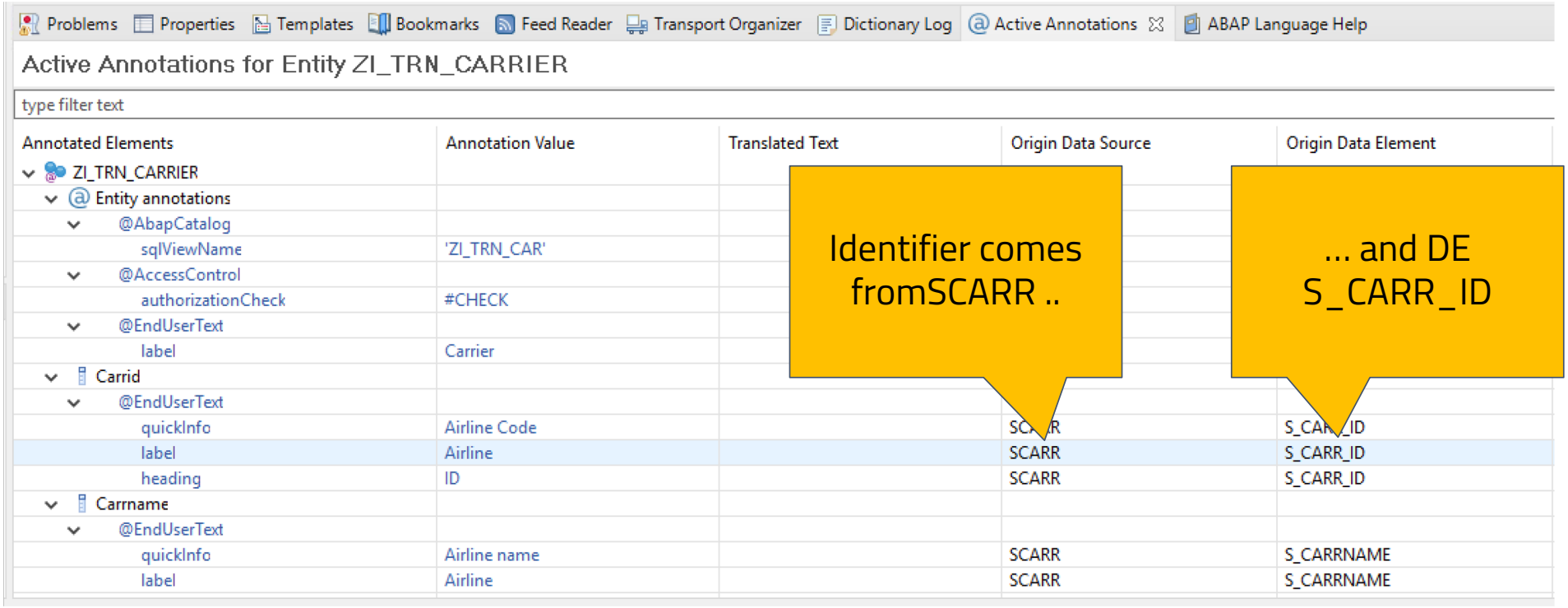
www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

ABAP Annotations for View Elements

```
Amount
        @Semantics.amount.currencyCode: 'Currency'
        price,
        Currency @<Semantics.currencyCode: true, //After element possible too
Quantity
        @Semantics.quantity.unitOfMeasure: 'Unit' //<-Use alias here
        distance,
        @Semantics.unitOfMeasure: true
        unit as Unit,
```

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Identifier



www.nor.gmbh office@nor.gmbh NöR GmbH – Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

ABAP annotations for translatable texts (SE63, short text, A5 User Interface Texts, DDLS CDS view)

View-Annotation

@EndUserText.label: ‘My Text’ //Max 60 Characters

Element-Annotation

@EndUserText.label: ‘Field Identifiers’ //Max 60 Characters

@EndUserText.quickInfo: ‘And another explanation’

afield,

www.nor.gmbh office@nor.gmbh NOR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Annotation texts with method in ABAP

To evaluate the semantic properties (annotations) of a CDS view (stored in system tables), a corresponding API (CL_DD_DDL_ANNOTATION_SERVICE, if available in your system) should be used.

Method Call

CL_DD_DDL_ANNOTATION_SERVICE=>GET_LABEL_4_ELEMENT(PA_CDS, PA_ELEM, PA_LANG)

Parameter

PA_CDS: DDSTRUCOBJNAME

PA_ELEM: DDFIELDNAME_L

PA_LANG: SY-LANGU

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

DDL Source and Annotation

5c

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Expressions

SQL Expressions

- **Expressions** are operations that can be used in the **field list**, **WHERE clause**, **GROUP BY**, or **HAVING** of a CDS view.
- They allow you to **calculate, transform, and conditionally derive values** at the database level.
- Expressions are **evaluated in the database** (pushdown principle).

Types of expressions:

- **Arithmetic Expressions**
- **String Expressions**
- **Date & Time Expressions**
- **CASE Expressions (Conditional Logic)**

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

SQL Expressions - Examples

- **Literals (characters, numbers)**
  - ‘Hello’ as Character1
  - 1 as Integer1

- **Type conversions (Target Type: Predefined Dictionary Types, ABAP Types: abap.int4, S_CARR_ID, ...)**
  - CAST (‘20250916’ AS /dmo/flight_date) as Fldate

- **CASE Expressions (Conditional Logic)**

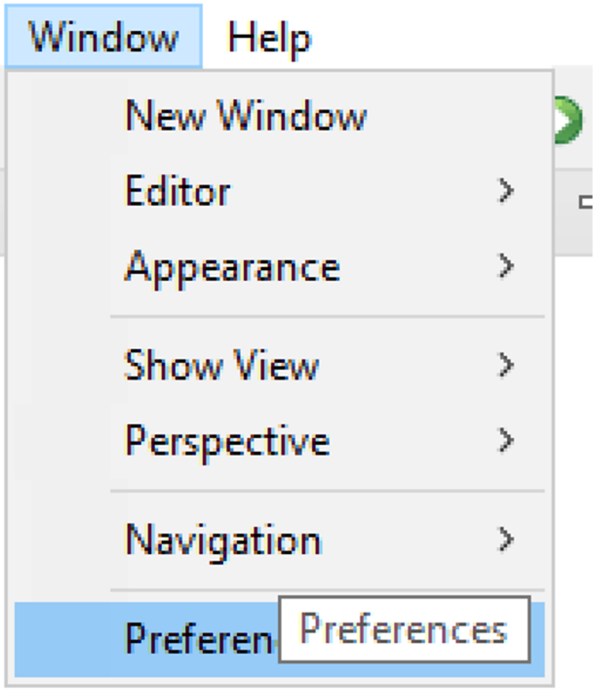
```
CASE operand // class
        WHEN operand1 THEN Value1 // WHEN ‘Y’ THEN ‘Yes’
        WHEN operand2 THEN Value2
        ...
        ELSE Valuen
END AS Classtext //Goal
```

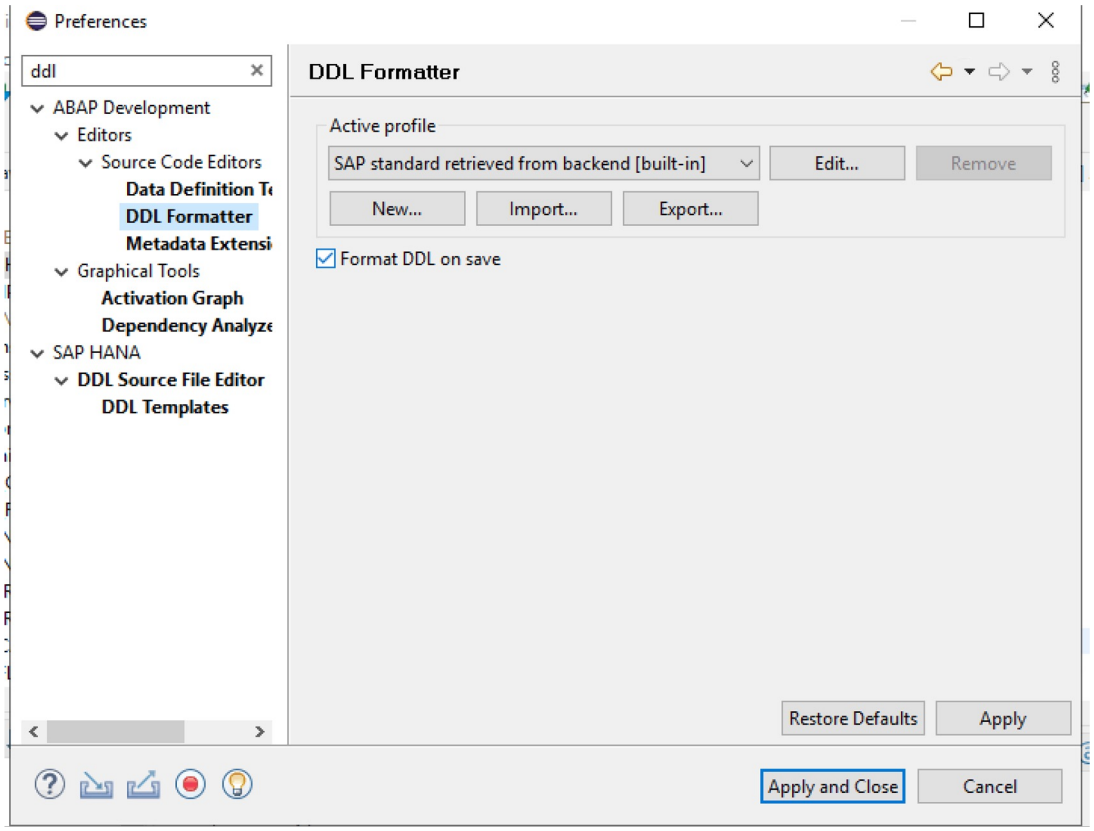
- **Arithmetic expressions (+,-,*,/)**
  - seatsmax - seatsocc as FreeSeats,
  - price * seatsocc as Revenue

Which types can be converted to which type(7.52) ABAP CDS - cast_expr - ABAP Keyword Documentation (sap.com) <https://help.sap.com/doc/abapdocu_752_index_htm/7.52/de-DE/abencds_f1_cast_expression.htm>

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Auto Format DDL on Save





www.nor.gmbh office@nor.gmbh NOR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Using Expressions

6

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Built in Functions

Built-in Functions

- CDS provides a large set of **built-in functions** that can be used directly in view definitions.
- They are evaluated on the **database level (pushdown)**, so they are highly efficient.

- We can group them into several categories:
  - **Arithmetic Functions**
  - **String Functions**
  - **Date & Time Functions**
  - **Conversion Functions**
  - **Boolean Functions**

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Built-in SQL functions

- Arithmetic Functions

div(ar1,arg2), mod(arg1,arg2), division(arg1,arg2,dec)

- Rounding Functions

abs(arg), floor(abs), ceil(arg), round(arg,pos)

- String functions

7.40: concat, replace, substring, ..

7.50: concat_with_space, length, left, right, ..

7.51: lower, upper, ..

- Currency and Unit Conversion

Unit_Conversion (DBTab T006), Currency_Conversion (DBTab TCUR)

- Calculations with dates

date_is_valid, dats_days_between, dats_add_days, dats_add_months

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Built-in Functions – Example CASE

```
define view entity ZC_Case
  as select from sflight
{
  carrid,
  price,
  case
    when price < 200 then 'CHEAP'
    when price between 200 and 500 then 'MEDIUM'
    else 'EXPENSIVE'
  end as PriceCategory
}
```

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Use built-in functions

7

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Nested Views

Nested Views – What?

- In ABAP CDS, a **nested view** is simply a CDS view that is based on another CDS view.
- Instead of building one very complex view, you can **layer views**:
- **Base view(s):** provide raw data selection.
- **Consumption view(s):** enrich or aggregate data for reporting, OData exposure, or analytics.

- This approach promotes **reusability, readability, and separation of concerns**.

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Nested Views – Why?

- **Reusability**: A base view can be used in multiple higher-level views.
- **Simplification**: Split complex logic into smaller, understandable parts.
- **Consistency**: KPIs or calculated fields defined once in a base view are reused consistently everywhere.
- **Performance**: Push-down to DB still applies — the database optimizer handles the nested queries efficiently.

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Nested View - Example

Interface View

```
@EndUserText.label: 'Flight Interface View'
define view entity I_Flight
  as select from sflight
{
  key carrid          as AirlineID,
  key connid          as ConnectionID,
  fldate              as FlightDate,
  price               as FlightPrice,
  currency            as CurrencyCode
}
```

- Acts as a clean, reusable **semantic layer**.
- No aggregations, no UI annotations — pure data access

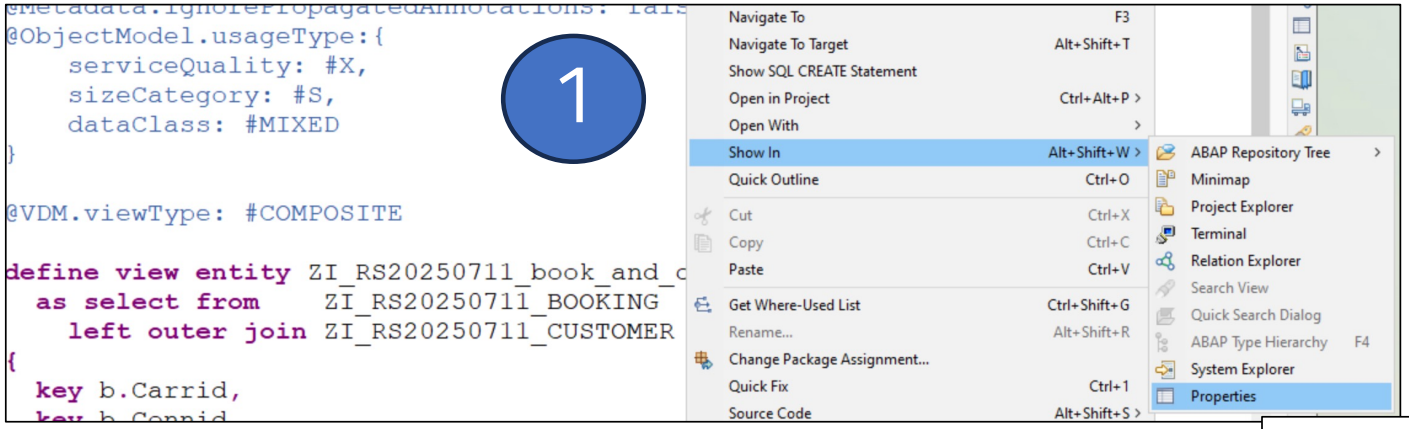
Composite View

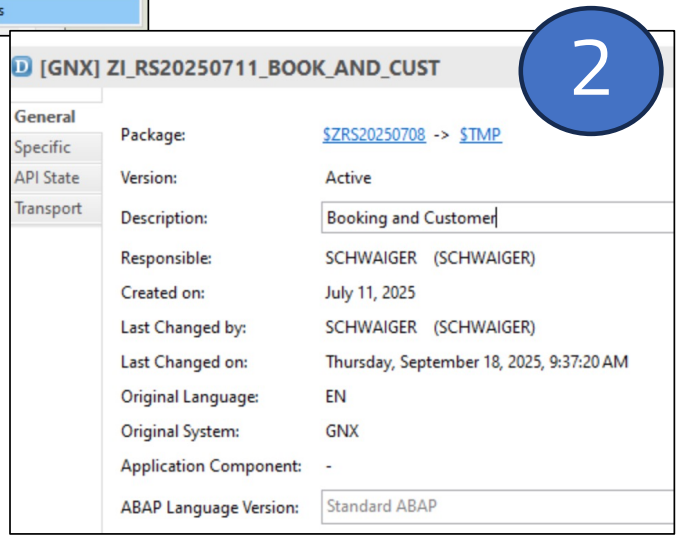
```
@EndUserText.label: 'Flight Aggregates by Airline'
define view entity C_FlightKPI
  as select from I_Flight
{
  key AirlineID,
  COUNT( * )     as NumFlights,
  AVG( FlightPrice )  as AvgPrice,
  SUM( FlightPrice )  as TotalRevenue,
  MIN( FlightPrice )  as MinPrice,
  MAX( FlightPrice )  as MaxPrice
}
group by AirlineID
```

- **Business logic layer**.
- Can be reused by multiple reports, apps, or analytics queries.

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Change Description of CDS DD





3

Activate!

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Nested Views

8

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Aggregations

Aggregation – What?

- In ABAP CDS (Core Data Services), aggregations are used to **summarize data across multiple rows**. Instead of returning all detailed records, aggregate functions compute a single value (or a smaller set of values) per grouping. They are very similar to SQL aggregates but integrated into the CDS modeling approach.

- Aggregations are typically used in **analytical scenarios**, **reporting views**, or whenever you need aggregated KPIs.

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Aggregations Functions

MIN, MAX

- Returns the **smallest/largest value** of an expression within the grouped set.
- Example: The earliest hire date in an organization.

SUM

- Returns the **total sum** of numeric values.
- Example: Total sales amount per customer.

AVG

- Returns the **average value** (arithmetic mean).
- Example: Average ticket price per airline.

COUNT

- Returns the **number of rows** in a group.
- Example: Number of flights per carrier.

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Aggregations - Example

Grouping and Aggregations

- Aggregations are only meaningful in combination with GROUP BY.
- Without GROUP BY, the entire dataset is treated as a single group.

Example with multiple aggregates:

```
define view entity ZSalesSummary
 as select from zsales
{
 customer_id,
 COUNT(*)     as NumOrders,
 SUM(amount)   as TotalAmount,
 AVG(amount)   as AvgOrder,
 MIN(amount)   as MinOrder,
 MAX(amount)   as MaxOrder
}
group by customer_id
```

Aggregations

9

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Input Parameters

Parameters – what for?

- Input for SQL expressions and functions
- Selection of data to be used in aggregations
- Mandatory selection criteria
- Filter on association

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Input parameter

```
Define view ...
with parameters
        par1: type, // abap.char(10), s_carr_id, ...
        par2: type,
         @Environment.systemField: #SYSTEM_DATE
        p_dats: syst_datum, // Optional Parameters
        @EndUserText.label: 'Personell Number'
        p_pernr: persno
        ...
$parameters.par1 //Acess, Usage
```

```
Check Feature Support (Before Release 7.50)
DATA: gt_feature_set TYPE CL_ABAP_DBFEATURES=>features_set_t.
INSERT CL_ABAP_DBFEATURES=>view_with_parameters INTO TABLE
gt_feature_set.
IF CL_ABAP_DBFEATURES=>use_feature( gt_feature_set ) = abap_false.
...
ENDIF.
```

Default Values

Value for the Annotation	Related ABAP system field
#CLIENT	client (sy-mandt)
#SYSTEM_DATE	system date (sy-datum)
#SYSTEM_TIME	system time (sy-uzeit)
#SYSTEM_LANGUAGE	language (sy-langu)
#USER	user name (sy-uname)
#USER_DATE	user date (sy-datlo)
#USER_TIMEZONE	user time zone (sy-zonlo)

Session Variables = ABAP system fields

Session Variable	Related ABAP system field
$session.client	sy-mandt
$session.system_date	sy-datum
$session.system_language	sy-langu
$session.user	sy-uname
$session.user_date	sy-datlo
$session.user_timezone	sy-zonlo

To access ABAP system fields in ABAP CDS,
prefer annotated input parameters (see next lesson)!

Access to input parameters

```
Define view ...
with parameters
        par1: type, // abap.char(10), s_carr_id, ...
        par2: type,
as select from ...{
$parameters.par1 as param1,
:par2 as param2,
}
```

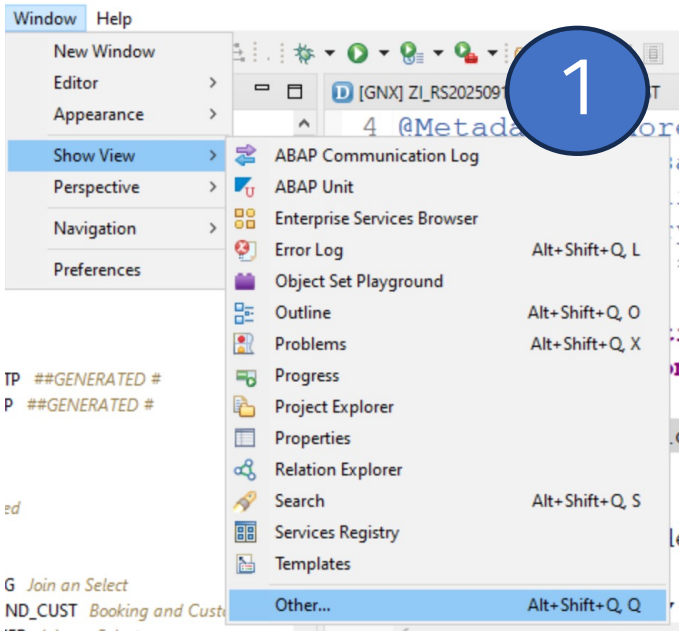
Where to use parameters in CDS?

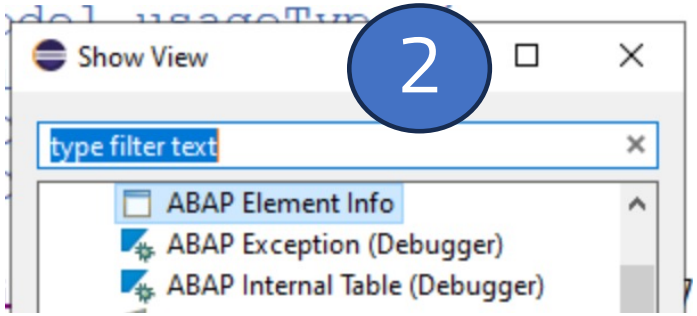
- in the element list
- in expressions
- in where or having clause
- in on condition of joins

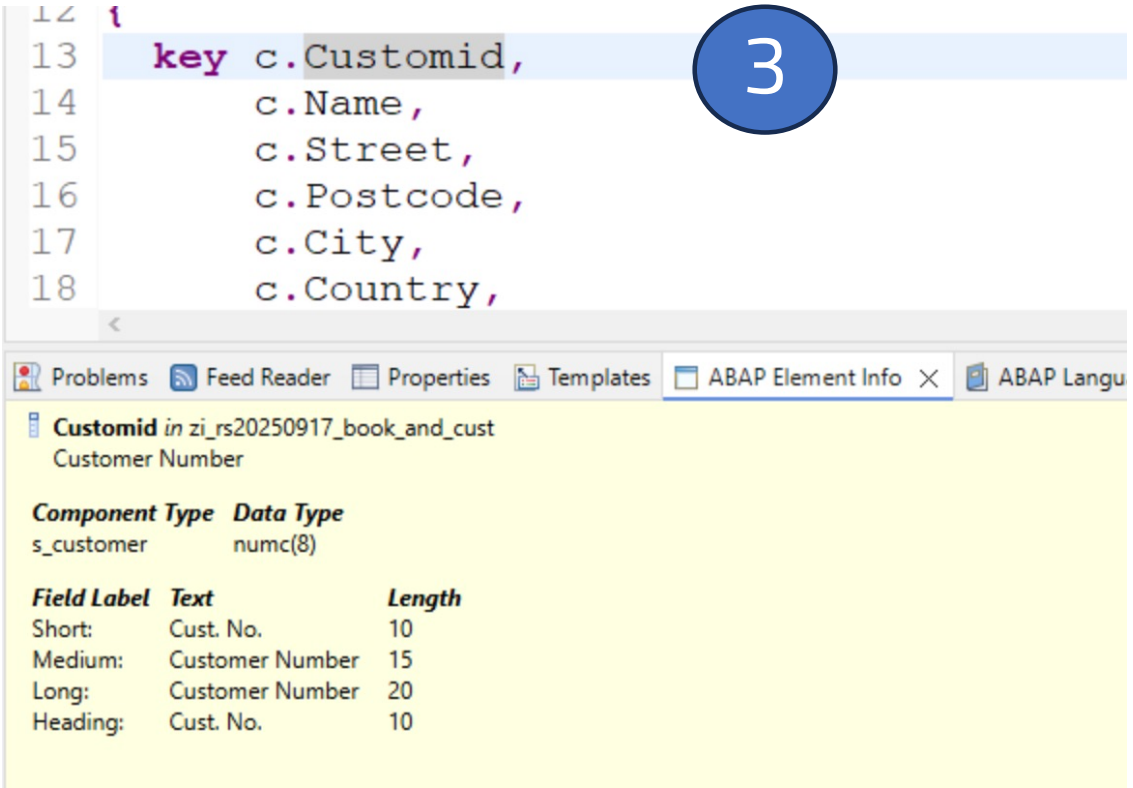
Where to use parameters in applications?
- in call of cds with parameters
as select from <cds>(par1: <value>, par2: <value>, ...)
- in ABAP SQL
SELECT FROM <cds>(par1 = <value>, par2 = <value>, ...) ...

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

ABAP Element Info







www.nor.gmbh office@nor.gmbh NOR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Input parameter

10

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

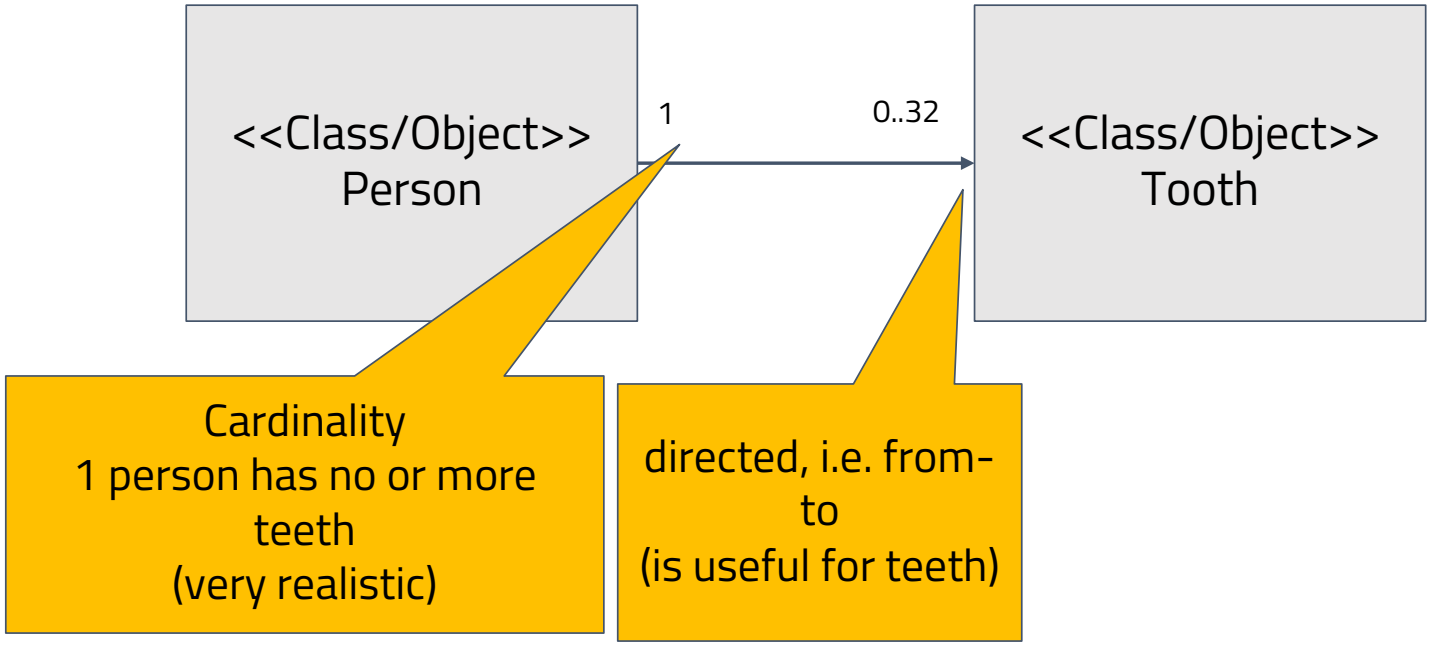
Associations

Relationships

- How to model relations between entity types?
- Where have you seen relationships in technical systems before?

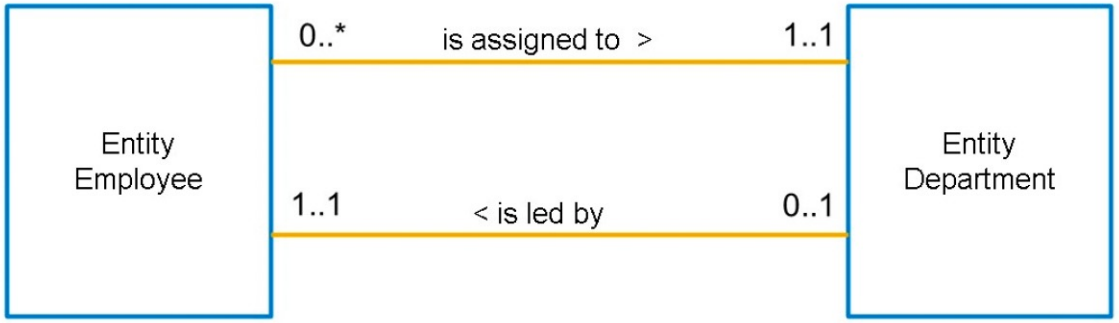
www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

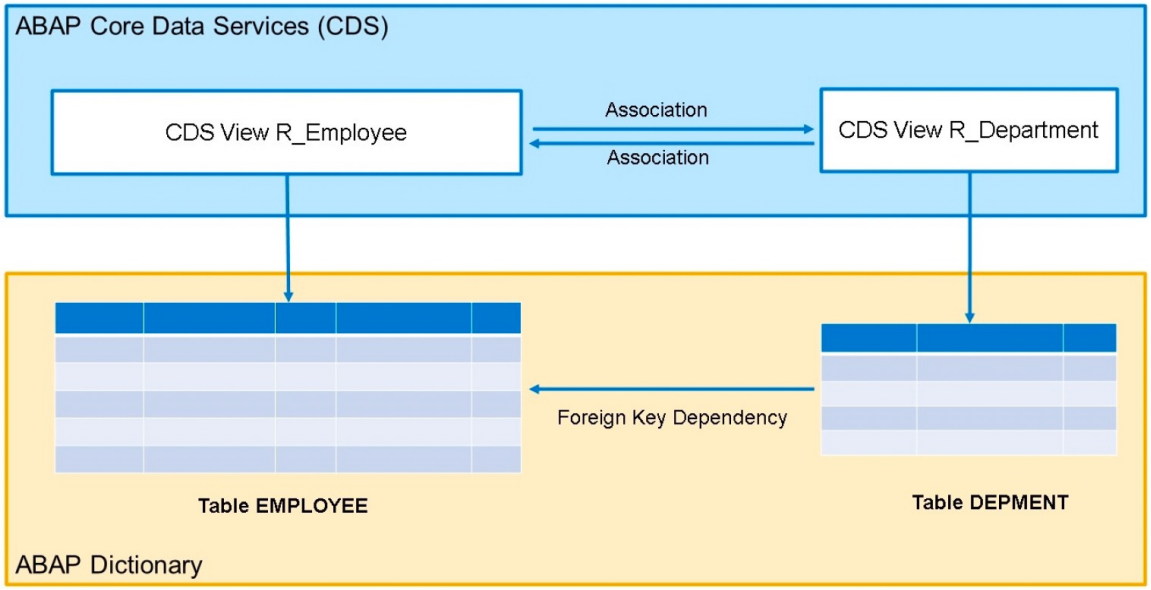
Associations - UML / ERM



www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Relationships between entity types



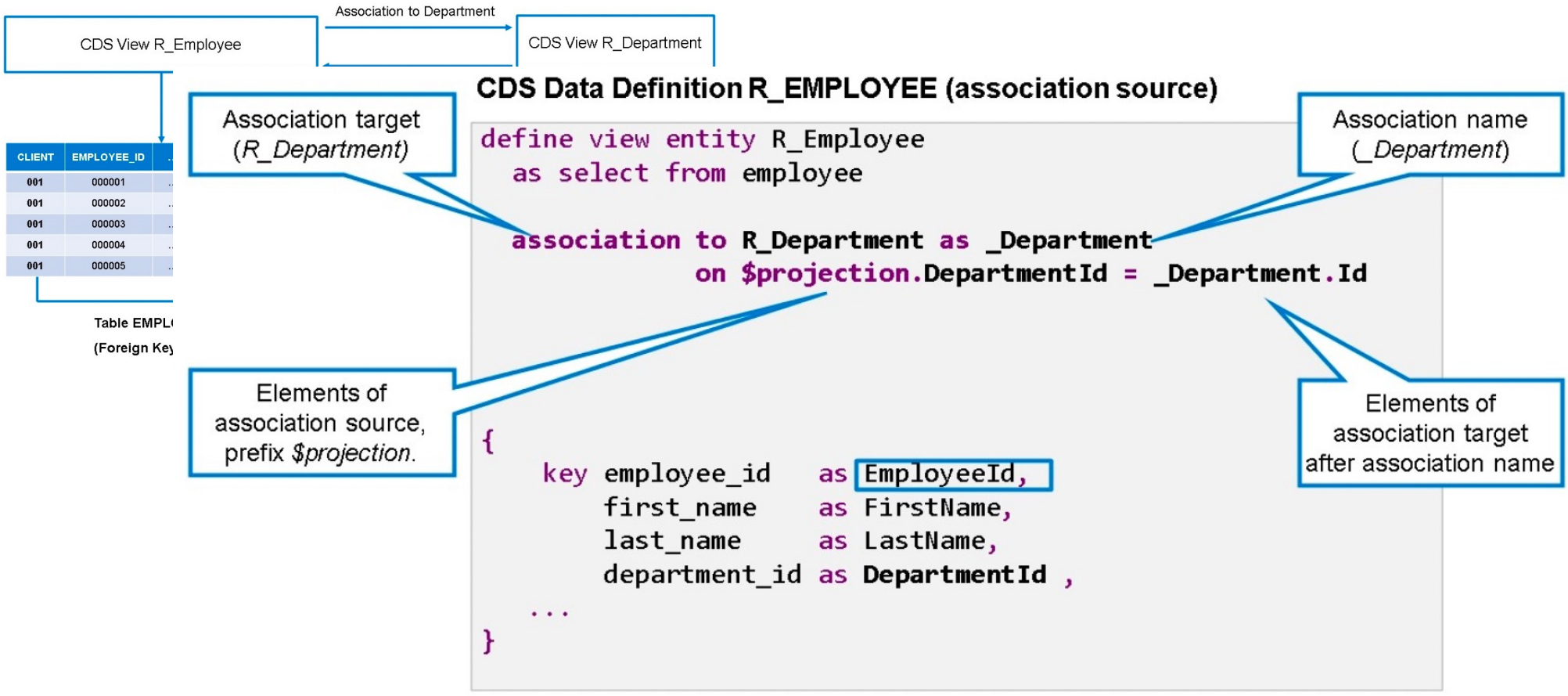


Associations CDS

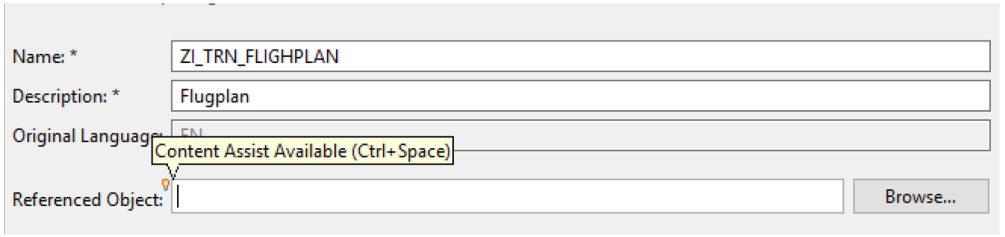
- Description of relationships
  - e.g. a join
  - Associations are technically realized as joins
- Associations are easier to read, just like before in the UML class diagram
  - Cardinalities
  - Paths as information about the data model and data origin
  - Filters as an alternative to WHERE or CASE
- Performance
  - Evaluation of on-demand ("JOIN on demand") of exposed associations

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

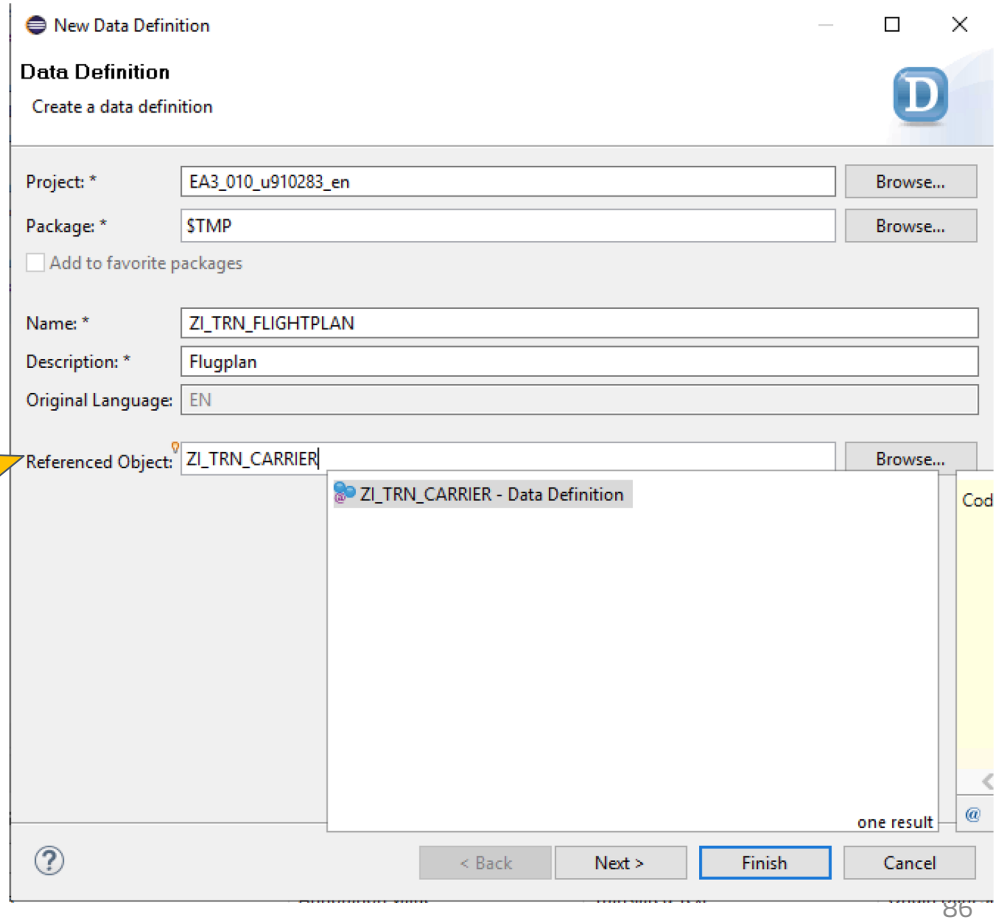
Association Syntax



Completion of the Association

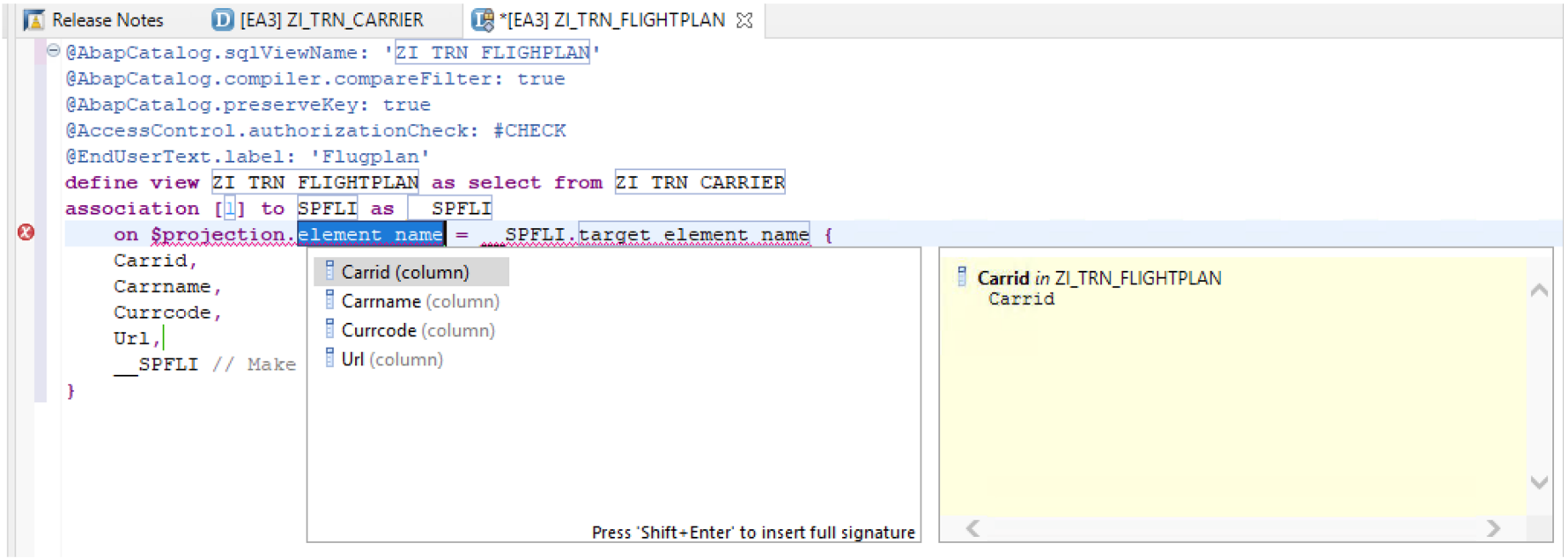


Efficient completion with CTRL + Space Bar



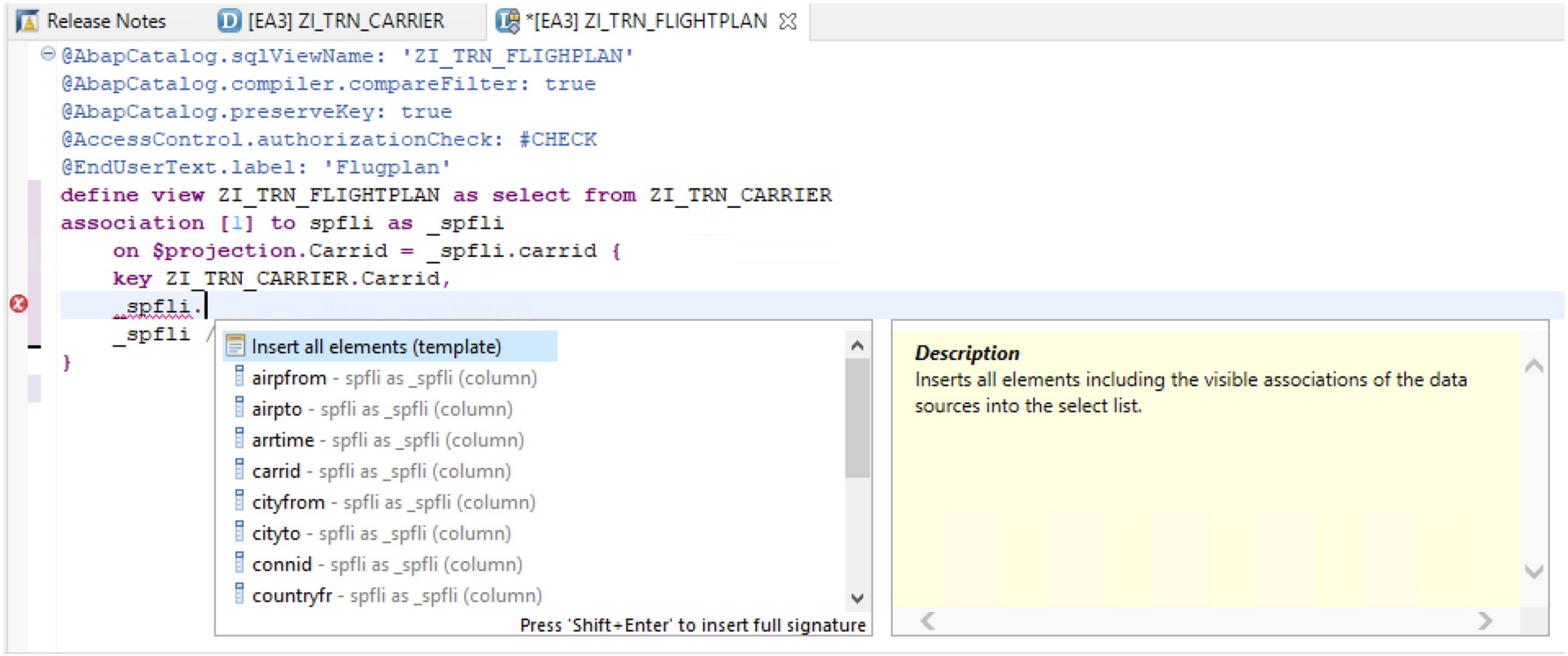
Associations

Link Criteria CDS View



www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Handling Associations



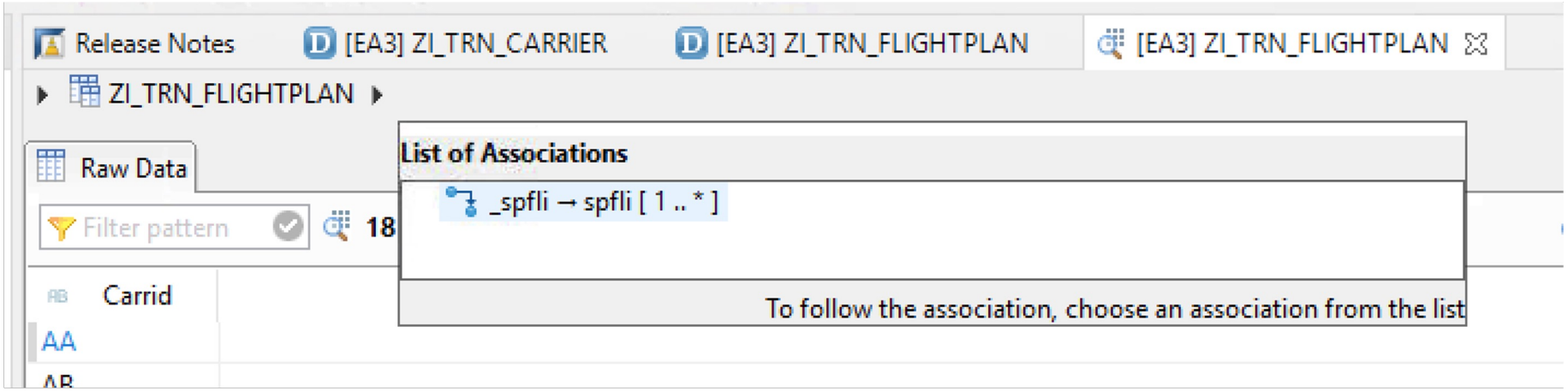
www.nor.gmbh office@nor.gmbh NOR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

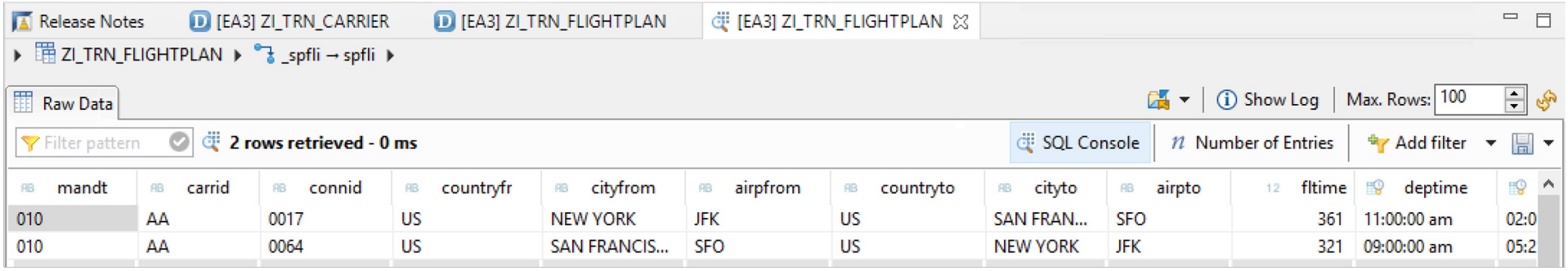
JOIN on Demand

- If association is just exposed with no usage in the field list then the actual join of the datasource is postponed until usage of a field.

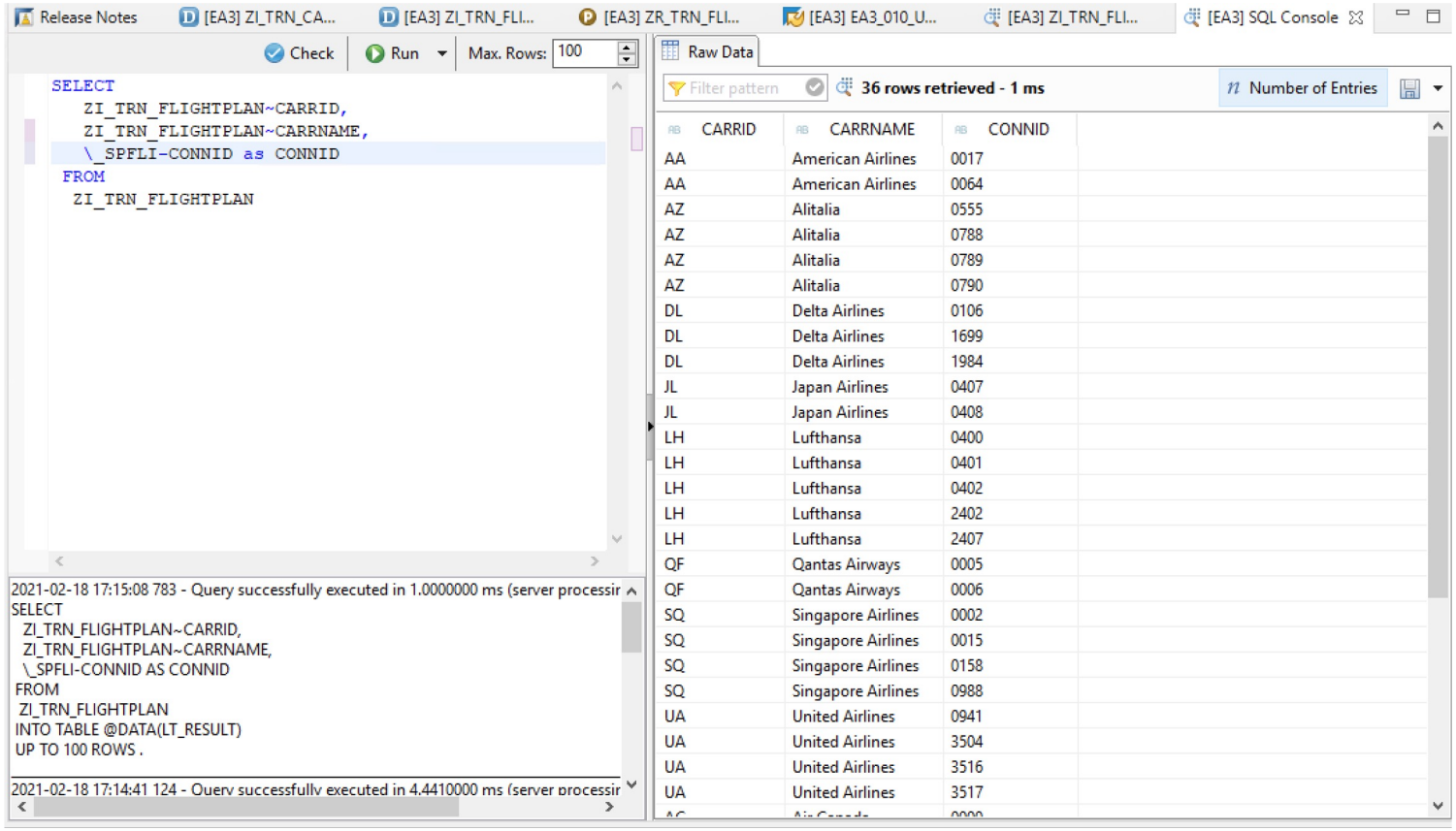
www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Association -> Navigation





CDS - Data Preview - SQL Console



www.nor.gmbh office@nor.gmbh NOR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

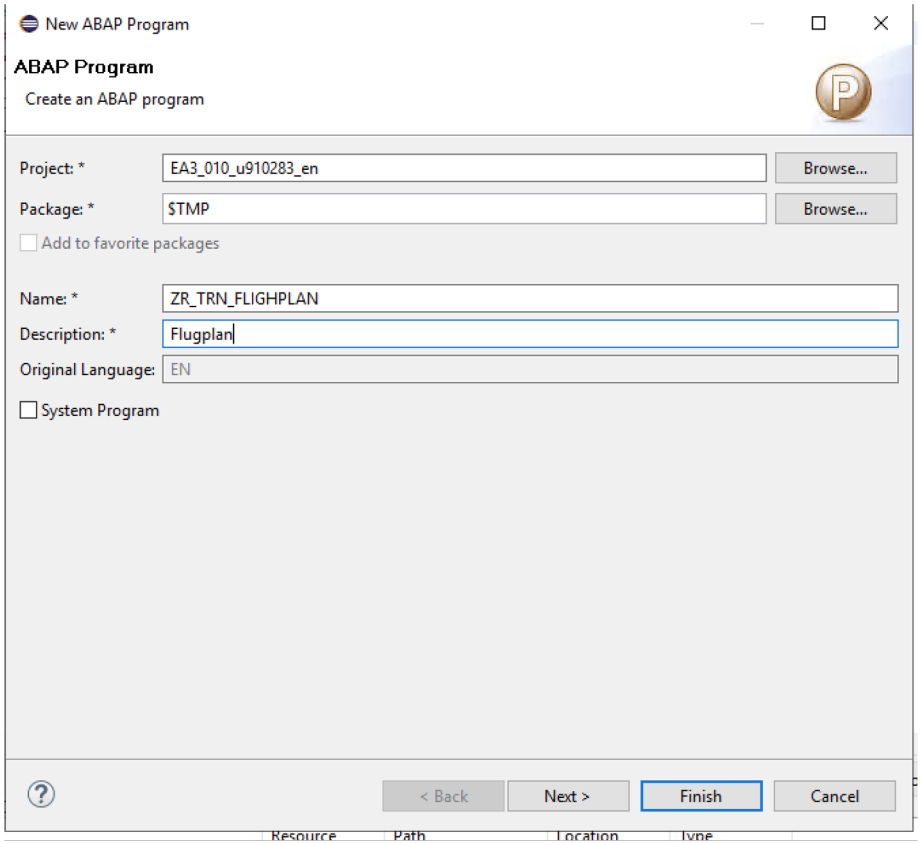
Associations
Association instead of Inner Join

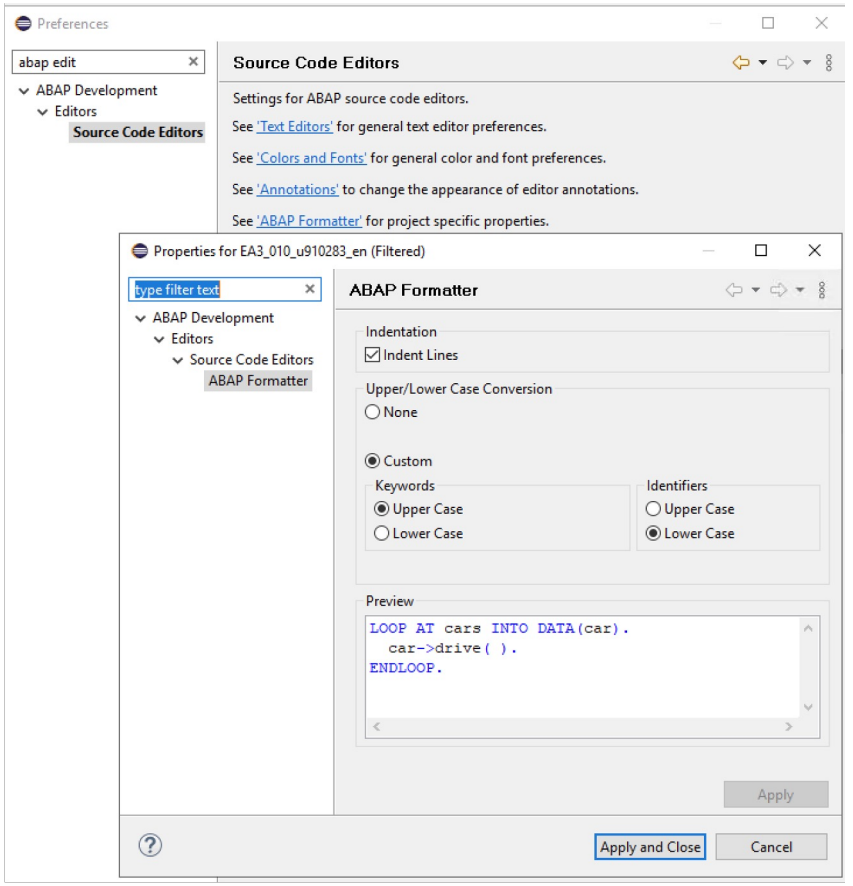
11

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Using Associations

ABAP and Pretty Printer





www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

CDS - ABAP Usage Exposed Association

Release Notes | [EA3] ZI_TRN_CARRIER | *[EA3] ZI_TRN_FLIGHTPLAN

```
@AbapCatalog.sqlViewName: 'ZI TRN FLIGHPLAN'
@AbapCatalog.compiler.compareFilter: true
@AbapCatalog.preserveKey: true
@AccessControl.authorizationCheck: #CHECK
@EndUserText.label: 'Flugplan'
define view ZI TRN FLIGHTPLAN as select from ZI TRN CARRIER
association [1] to SPFLI as   SPFLI
    on $projection.element name =   SPFLI.target element name
    Carrid,
    Carrname,
    Currcode,
    Url,
    __SPFLI // Make
}
```

Carrid (column)
Carrname (column)
Currcode (column)
Url (column)

In ABAP "\" for all association names

If an alias is used: a~_spfli-connid

Release Notes | [EA3] ZI_TRN_CARRIER | [EA3] ZI_TRN_FLIGHTPLAN | [EA3] ZR_TRN_FLIGHPLAN

ZR_TRN_FLIGHPLAN

```
*&---------------------------------------------------------------------*
*& Report zr_trn_flighplan
*&---------------------------------------------------------------------*
*& Verwendung von CDS Views im ABAP
*&---------------------------------------------------------------------*
REPORT zr_trn_flighplan.

DATA: BEGIN OF gs_flightplan,
        carrid TYPE s_carr_id,
        connid TYPE s_conn_id,
      END OF gs_flightplan,
      gt_flightplan2 LIKE TABLE OF

START-OF-SELECTION.

* Simple SELECT
  SELECT * FROM zi_trn_flightpl    TO TABLE @DATA(gt_flightplan1).

* Projection and Selection
  SELECT
    carrid,
    \_spfli-connid
   FROM zi_trn_flightplan
   where carrid = 'LH'
   and   \_spfli-connid = '0400'
   INTO TABLE @gt_flightplan2.

* Ausgabe
  cl_demo_output=>display( gt_flightplan2 ).
```

Im ABAP ist der Komponentenselektor "-" in der CDS-DDL ist es "."

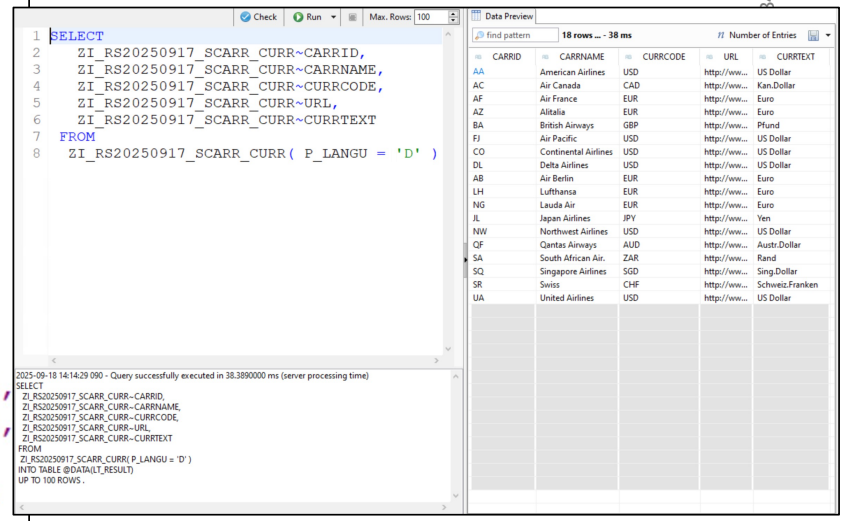
Filter in Associations

```
@AbapCatalog.viewEnhancementCategory: [#NONE]
@AccessControl.authorizationCheck: #NOT_REQUIRED
@EndUserText.label: 'Currency'
@Metadata.ignorePropagatedAnnotations: true
@ObjectModel.usageType:{
    serviceQuality: #X,
    sizeCategory: #S,
    dataClass: #MIXED
}
define view entity ZI_RS20250917_SCARR_CURR
  with parameters
    p_langu : syst_langu
  as select from scarr
  association [0..*] to tcurt as _Currency
  on $projection.Currcode = _Currency.waers
{
  key carrid                                    as Carrid,
      carrname                                  as Carrname,
      currcode                                  as Currcode,
      url                                       as Url,
      _Currency[spras = $parameters.p_langu].ktext as CurrText
}
```

Data Preview

find pattern — 18 rows retrieved - 46 ms

Carrid	Carrname	Currcode	Url	CurrText
AA	American Airlines	USD	http://w...	US Dollar
AC	Air Canada	CAD	http://w...	Kan.Dollar
AF	Air France	EUR	http://w...	Euro
AZ	Alitalia	EUR	http://w...	Euro
BA	British Airways	GBP	http://w...	Pfund



ABAP

- Using DDL Source in ABAP (Link Postings, Customers)
- Write a small program
- Activate, execute

12

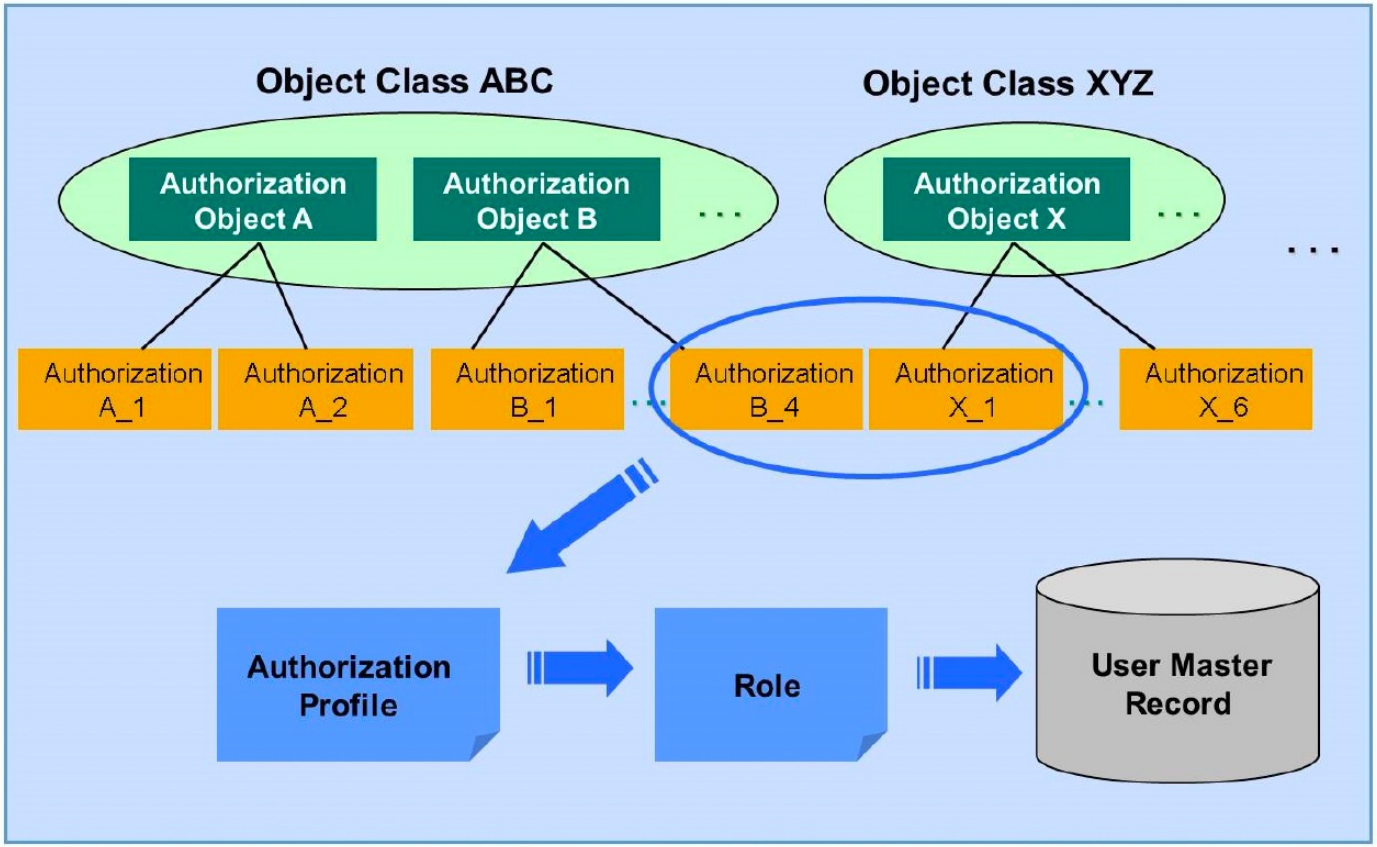
www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Access Control

Access control (Classical)

Authorization Object = SU21
Roles: PFCG
User: SU01

Check Auth User: SU53
User (Fiori): STAUTHTRACE



www.nor.gmbh office@nor.gmbh NOR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Auth Checks – Who is checking?

- Implicit Authorization Checks
  - Performed by the ABAP Framework
  - Start of a transaction
  - Start of a Web Dynpro application
  - Start of a report
  - Call of an RFC function module
  - Access to table content with data browser or maintenance dialogue

- Explicit Authorization Checks
  - Defined in the ABAP Coding (AUTHORITY-CHECK statement)
  - Access to specific functionality
  - Access to specific data

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Known Problems

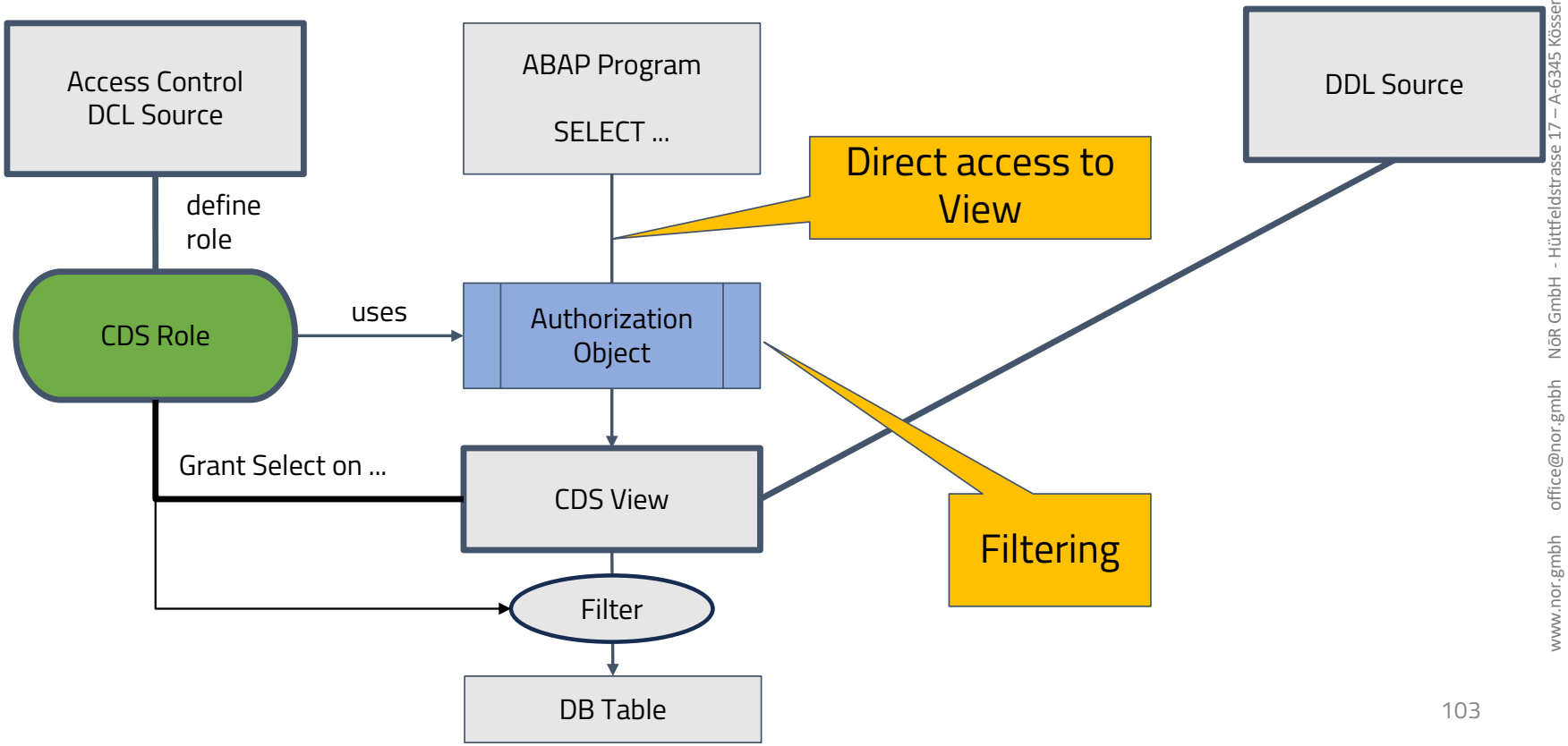
- General Problems:
  - No visible link between data and authorization objects
  - Developer can forget (or ignore) necessary checks
  - Users with debugging rights can bypass authorization checks
- Problems related to Code-to-Data:
  - All authorization checks on application level
  - Some calculations cannot be done on the database because they require an authorization check between steps

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Access control (CDS)

```
@EndUserText.label: 'Zugriffskontrolle Fluggesellschaft'
@MappingRole: true
define role ZI_TRN_CARRIER_AC {
    grant select on ZI_TRN_CARRIER
```

```
@AbapCatalog.sqlViewName: 'ZI_TRN_CAR'
@AccessControl.authorizationCheck: #CHECK
@EndUserText.label: 'Carrier'
define view ZI_TRN_CARRIER as select from scarr
{
```



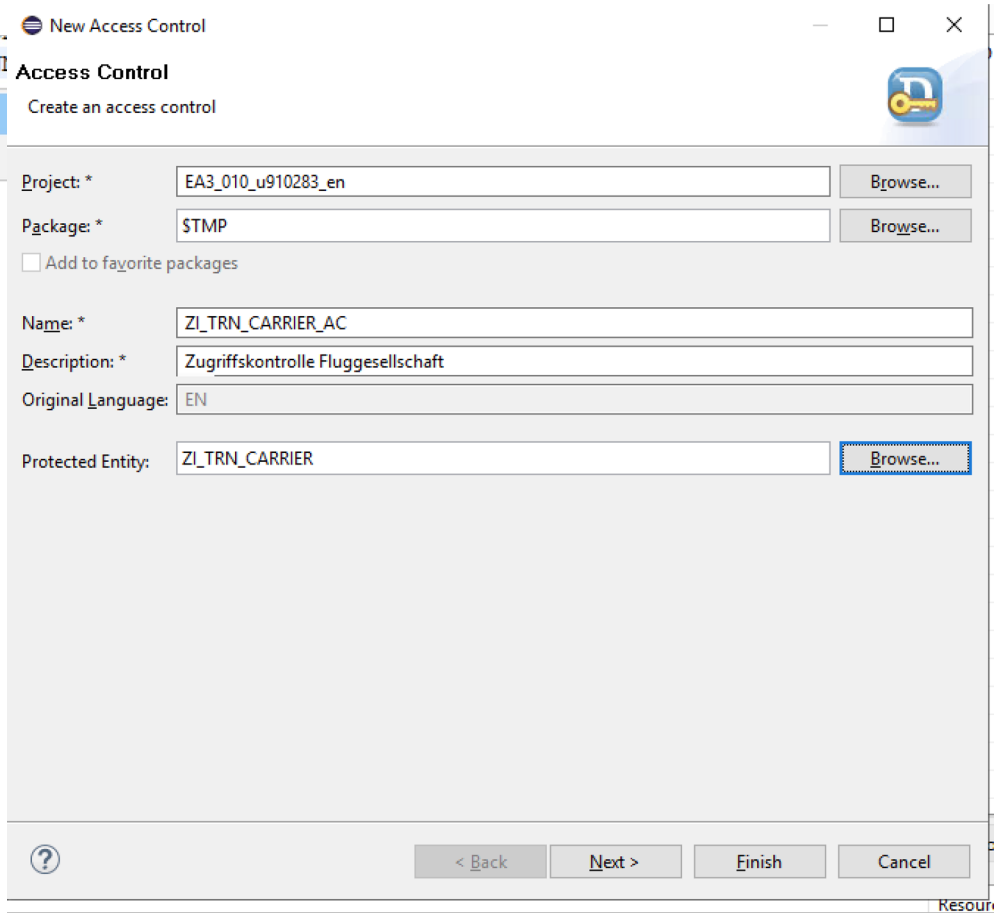
www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Example Code – AUTHORITY-CHECK

```
DATA gt_connections TYPE TABLE OF ZI_CONN.
DATA gs_connection LIKE LINE OF gt_connections.

PARAMETERS pa_citfr TYPE s_from_cit DEFAULT 'FRANKFURT'.

SELECT FROM ZI_CONN
 FIELDS carrid, connid, cityfrom, cityto
 WHERE cityfrom = @pa_citfr
 INTO TABLE @gt_connections.

LOOP AT gt_connections INTO gs_connection.


 AUTHORITY-CHECK OBJECT 'S_CARRID'
  ID 'CARRID' FIELD gs_connection-carrid
  ID 'ACTVT' FIELD '03'.
 IF sy-subrc <> 0.
  DELETE gt_connections INDEX sy-tabix.
 ENDIF.

ENDLOOP.
```

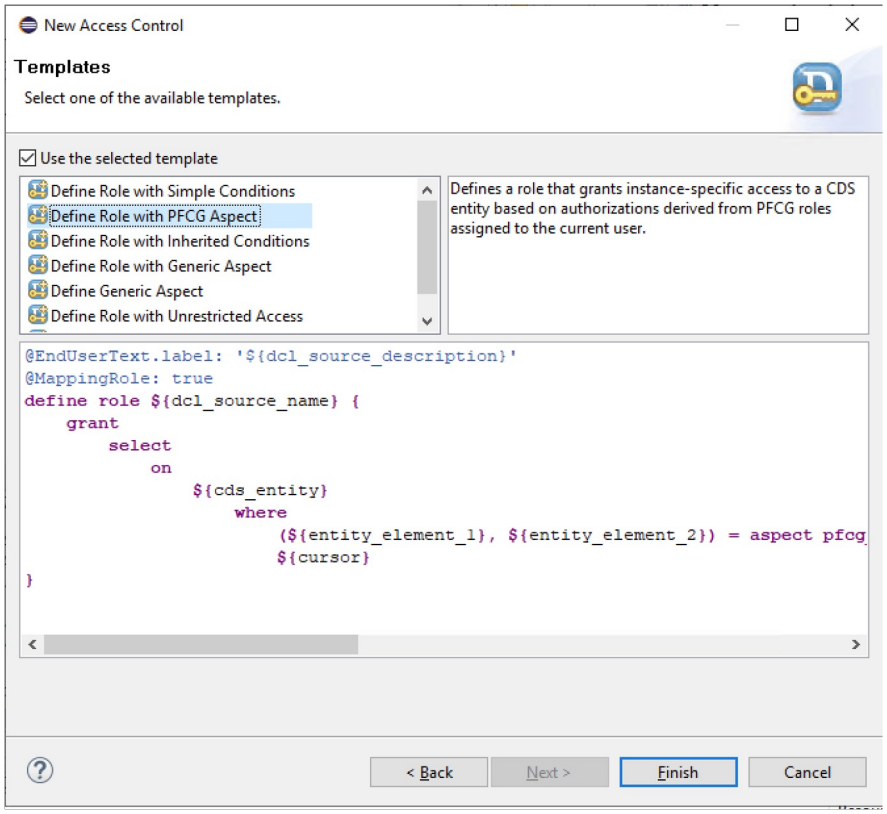
www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

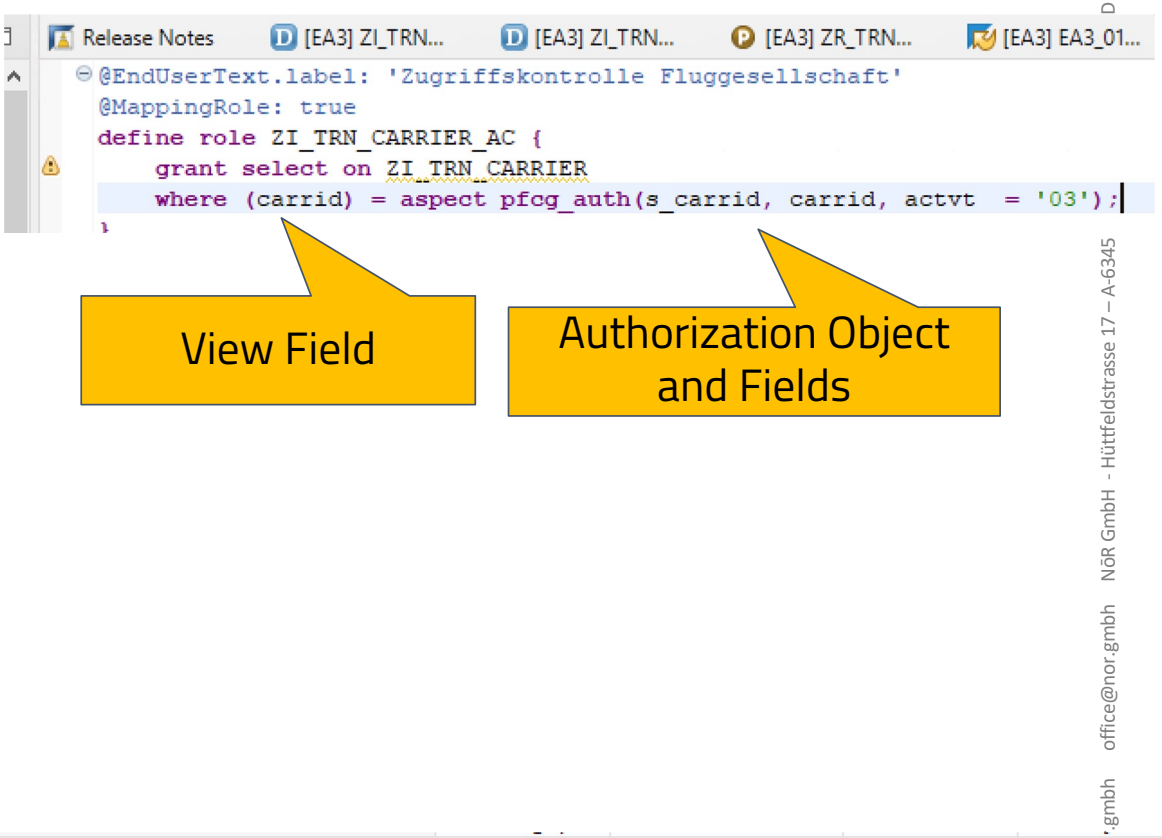
Create Access Control



www.nor.gmbh office@nor.gmbh NOR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Create Access Control





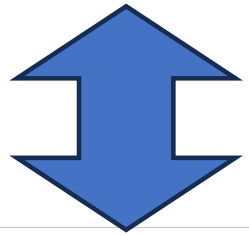
Infos (1 item)				
The entity ZI_TRN_CARRIER is used in DCL document ZI_TRN_CARRIER_AC [Access Control Management]	zi_trn_carrier....	/EA3_010_u910283_...	Unknown	ABAP Syntax ...

Example

```
@EndUserText.label: 'Demo: Authorization Check With Link to User Profile'
@MappingRole: true
define role ZA_CONN
{
   grant select
      on ZI_CONN
      where (carrid) = aspect pfcg_auth( S_CARRID,
                                 CARRID,
                                 ACTVT = '03' )

       and (carrid, counter) = aspect pfcg_auth( S_COUNTER,
                                       CARRID,
                                       COUNTNUM,
                                       ACTVT = '03' );
}
```

Mapping between classical and CDS Access Control

```
where (carrid)          = aspect pfcg_auth( S_CARRID,
                                            CARRID,
                                            ACTVT = '03'
                                          )
```



```
AUTHORITY-CHECK OBJECT 'S_CARRID'
           ID 'CARRID' FIELD view-carrid
           ID 'ACTVT'  FIELD '03'.
```


www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Annotation in Access Control

@EndUserText.label:

– Translatable short text for role (same as in Data Definition)

@MappingRole:

– Value true: Role is implicitly assigned to all users

– Value false: Currently not supported by ABAP CDS

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Annotation in Data Definition

@AccessControl.authorizationCheck:

#CHECK

Performs authorization check and provides warning if no role is assigned

#NOT_REQUIRED

Performs permission check and does not provide a warning if no role is assigned

#NOT_ALLOWED

Does not perform an authorization check and provides a warning if a role is assigned

www.nor.gmbh office@nor.gmbh NOR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Access Control

- Provide DDL Source (Link Bookings, Customers)
  - ZZI_xx_SBOOK_ASSO -> ZZI_xx_SBOOK_AUTH
- Define CDS Role
  - Name DCL Source and CDS Role Same
  - Template DefineMappingRole
  - Check CARRID
- Activate, view data

13

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Buffering CDS View Entities

Buffering CDS View Entities

- A CDS entity buffer temporarily caches the data retrieved from a CDS view entity into the shared memory of the current AS ABAP instance.

  Starting with the following ABAP releases, **buffering** of CDS view entities is supported:

  SAP BTP, ABAP Environment 2202

- ABAP platform cloud 2202

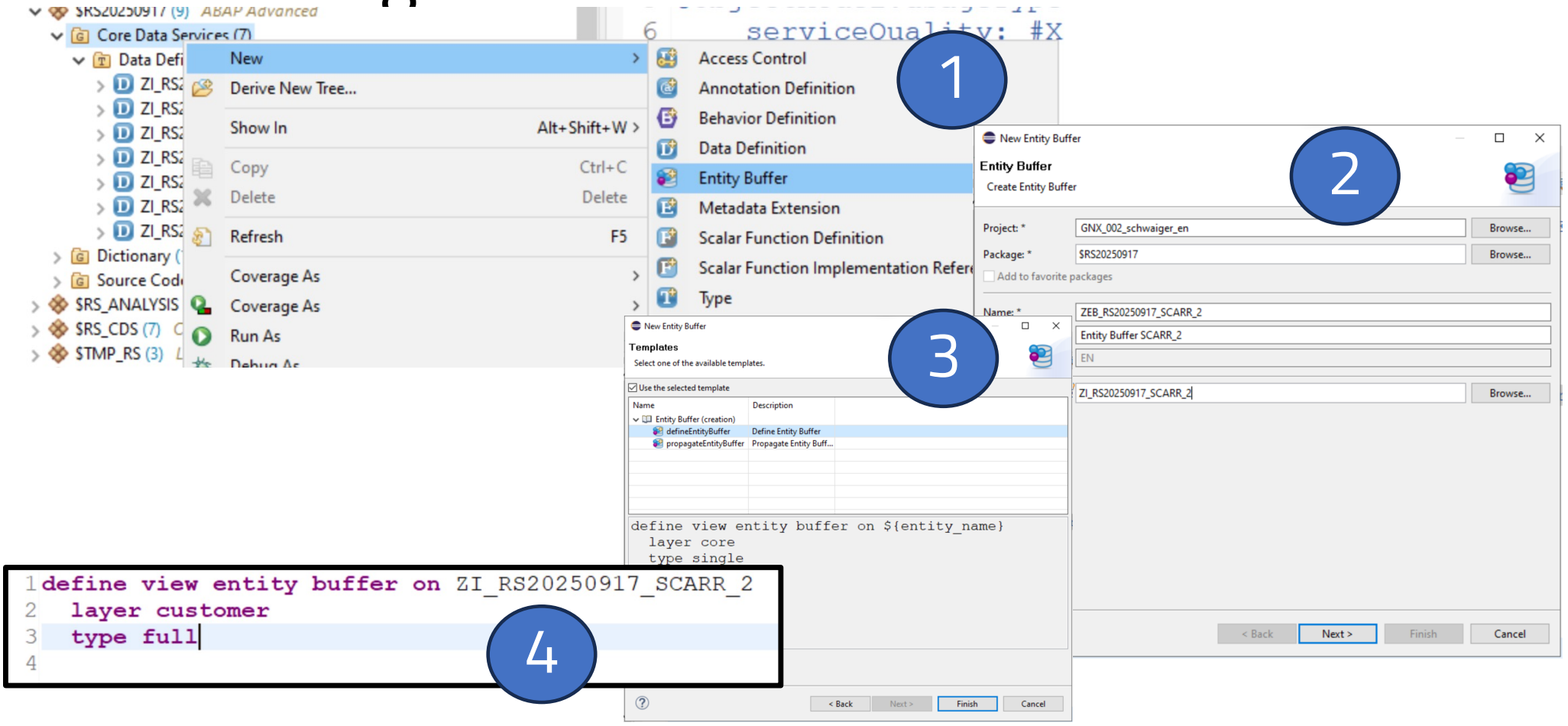
www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Buffering CDS View Entities
Preparation

- **Prepare CDS View Entity for Buffer Definitions**
  @AbapCatalog.entityBuffer.definitionAllowed: true
- The annotation itself does not lead to a buffer, but ensures the following restrictions apply:
  - Views with parameters are not supported
  - Views with unstable calculations are not supported (e.g. utcl_current, tstmp_current_utctimestamp)
  - Session variables (other than client) can not be used
  - Base objects have to be tables
  - Key length > 900 bytes is not supported
  - Views without key fields can not be buffered
  - Data aging in base tables is not allowed

www.nor.gmbh office@nor.gmbh NöR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

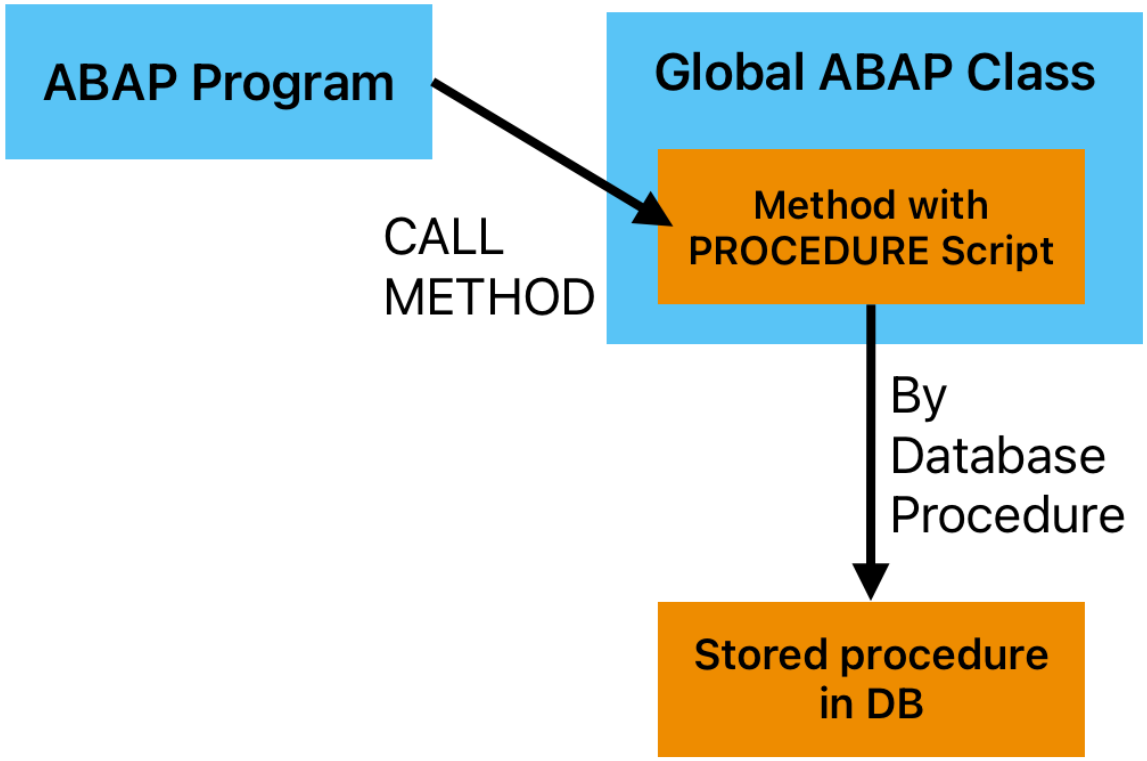
Buffering CDS View Entities



https://community.sap.com/t5/application-development-and-automation-blog-posts/buffering-cds-view-entities/ba-p/13520422

AMDP Procedures & Table Functions

AMDP


ABAP Program
Global ABAP Class
CALL
METHOD
Method with
PROCEDURE Script
By
Database
Procedure
Stored procedure
in DB

AMDP Code

Class Definition
- Implement the interface IF_AMDP_MARKER_HDB

Method Definition
- Add VALUE( ) for all parameters
- Only scalar types and table types allowed for parameters (no structures)

Supported Databases and Languages
- Only SAP HANA (currently)
- Only SAP HANA SQLScript (currently)

Use of Dictionary Objects
- Transparent tables and dictionary views
- Till 7.50: CDS views only via their CDS-SQL views
- Must be listed using the USING clause

AMDP Example – Fuzzy Search

ZCL_RS20250917_FUZZY

```
1  CLASS zcl_rs20250917_fuzzy DEFINITION
2    PUBLIC
3    FINAL
4    CREATE PUBLIC .
5
6    PUBLIC SECTION.
7      types: td_score type p length 3 DECIMALS 2.
8      types: begin of mts_carrier,
9               score type td_score,
10              carrid type scarr-carrid,
11              carrname type scarr-carrname,
12            end of mts_carrier,
13     mtt_carriers type standard table of mts_carrier.
14
15     INTERFACES if_amdp_marker_hdb.
16
17     class-methods: carrier_fuzzy
18                      importing value(iv_name) type scarr-carrname
19                      exporting value(et_carriers) type mtt_carriers.
20
21   PROTECTED SECTION.
22   PRIVATE SECTION.
23 ENDCLASS.
24
25 CLASS zcl_rs20250917_fuzzy IMPLEMENTATION.
26   METHOD carrier_fuzzy BY DATABASE PROCEDURE FOR HDB LANGUAGE SQLSCRIPT using Scarr.
27     et_carriers = select score( ) as score,carrid, carrname from scarr
28                   where CONTAINS (carrname, :iv_name, fuzzy( 0.8 ))
29                   and   mandt = session_context(  'CLIENT' ) ORDER BY score(  );
30   ENDMETHOD.
31
32 ENDCLASS.
```

TestObject->CARRIER_FUZZY()

Case-Sensitive

Runtime: 201.514 Microseconds

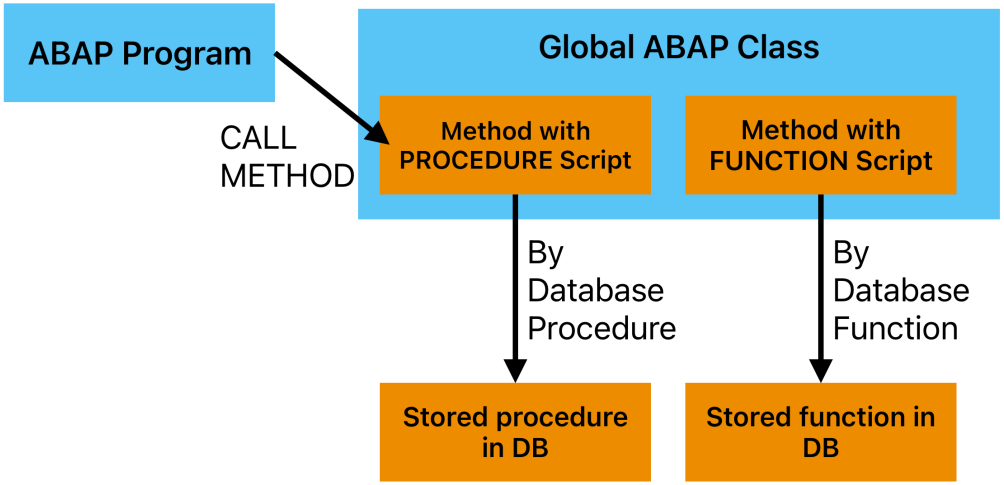
- CARRIER_FUZZY
  - Import Parameter
    - IM_CARRNAME — LUFTANSA
  - Export Parameter
    - EX_DETAILS — 1 Entry

1 Entry

SCORE	CAR	CARRNAME
0,91	LH	Lufthansa

order by score()

AMDP Function



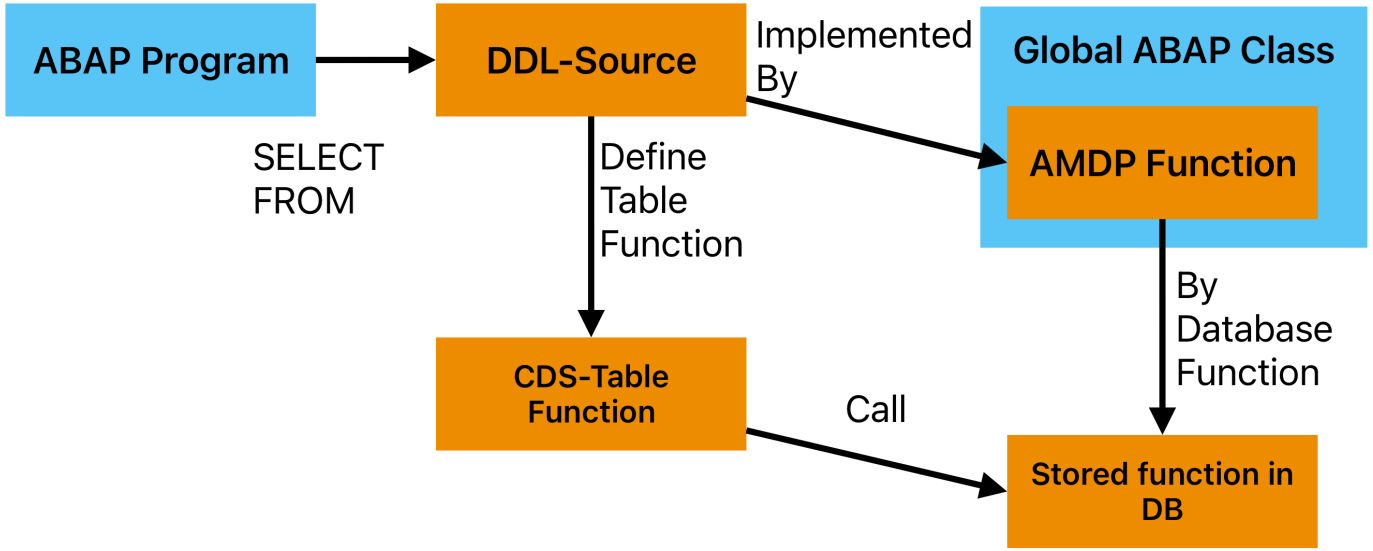
Unlike the implementation of **AMDP procedures**, the implementations of **AMDP functions** do not have export or changing parameters. They have exactly one table-like return value.

AMDP function implementations cannot be called directly from an ABAP program. However, they can be called from other AMDP functions or procedures.

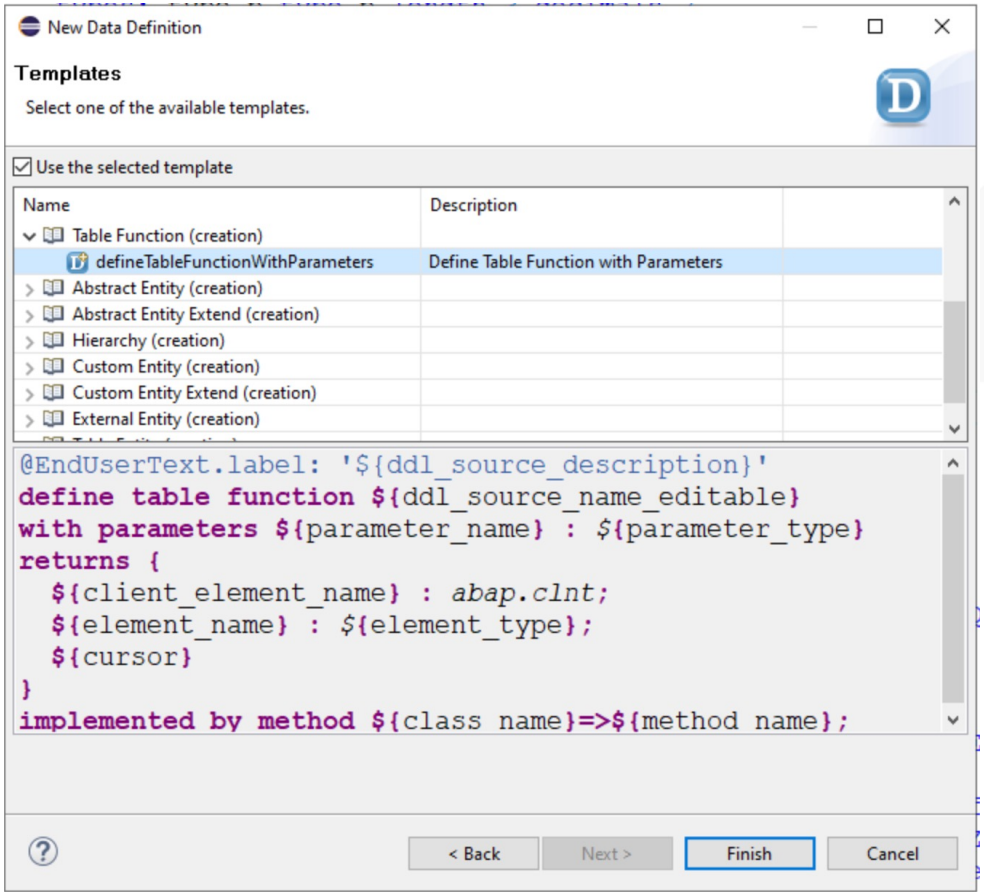
AMDP Function

```
CLASS zcl_rs20250716_tbl_fun_fuzzy DEFINITION
  PUBLIC
  FINAL
  CREATE PUBLIC.

  PUBLIC SECTION.
    INTERFACES if_amdp_marker_hdb.
    CLASS-METHODS: customer_fuzzy FOR TABLE FUNCTION
zi_rs20250716_tbl_fun_fuzzy.
  PROTECTED SECTION.
  PRIVATE SECTION.
ENDCLASS.

CLASS zcl_rs20250716_tbl_fun_fuzzy IMPLEMENTATION.
  METHOD customer_fuzzy BY DATABASE FUNCTION FOR HDB LANGUAGE SQLSCRIPT
OPTIONS READ-ONLY USING scustom.
    return select mandt, id, name, city, country from scustom where
contains( name,:name_in,fuzzy( 0.8 ) )| order by  score( ) desc;
  ENDMETHOD.
ENDCLASS.
```

CDS-Table Function



CDS-Table Functions




```
@EndUserText.label: 'Fuzzy'
define table function ZI_RS20250716_TBL_FUN_FUZZY
  with parameters
    name_in : s_custname
returns
{
  mandt   : abap.clnt;
  id      : s_customer;
  name    : s_custname;
  city    : s_city;
  country : s_country;

}
implemented by method
  ZCL_RS20250716_TBL_FUN_FUZZY=>customer_fuzzy;
```

```
CLASS zcl_rs20250716_tbl_fun_fuzzy DEFINITION
  PUBLIC
  FINAL
  CREATE PUBLIC .

  PUBLIC SECTION.
    INTERFACES if_amdp_marker_hdb .
    CLASS-METHODS: customer_fuzzy FOR TABLE FUNCTION zi_rs20250716_tbl_fun_fuzzy.
  PROTECTED SECTION.
  PRIVATE SECTION.
ENDCLASS.

CLASS zcl_rs20250716_tbl_fun_fuzzy IMPLEMENTATION.
  METHOD customer_fuzzy
  BY DATABASE FUNCTION
  FOR HDB
  LANGUAGE SQLSCRIPT
  OPTIONS READ-ONLY
  USING scustom.
    RETURN SELECT mandt, id, name, city, country
    FROM scustom
    WHERE contains( name,:name_in,fuzzy( 0.8 ) )
    order by  score( ) desc;
  ENDMETHOD.
ENDCLASS.
```

CDS-Table Functions in use

Use CDS-Table Function

```
* Data Retrieval
*********************************************************************
  SELECT FROM zI_rs20250716_tbl_fun_fuzzy( name_in = @pa_nam )
  FIELDS id, name, city, country ORDER BY country, city, name
  INTO TABLE @gt_customers.
```

Check feature availability

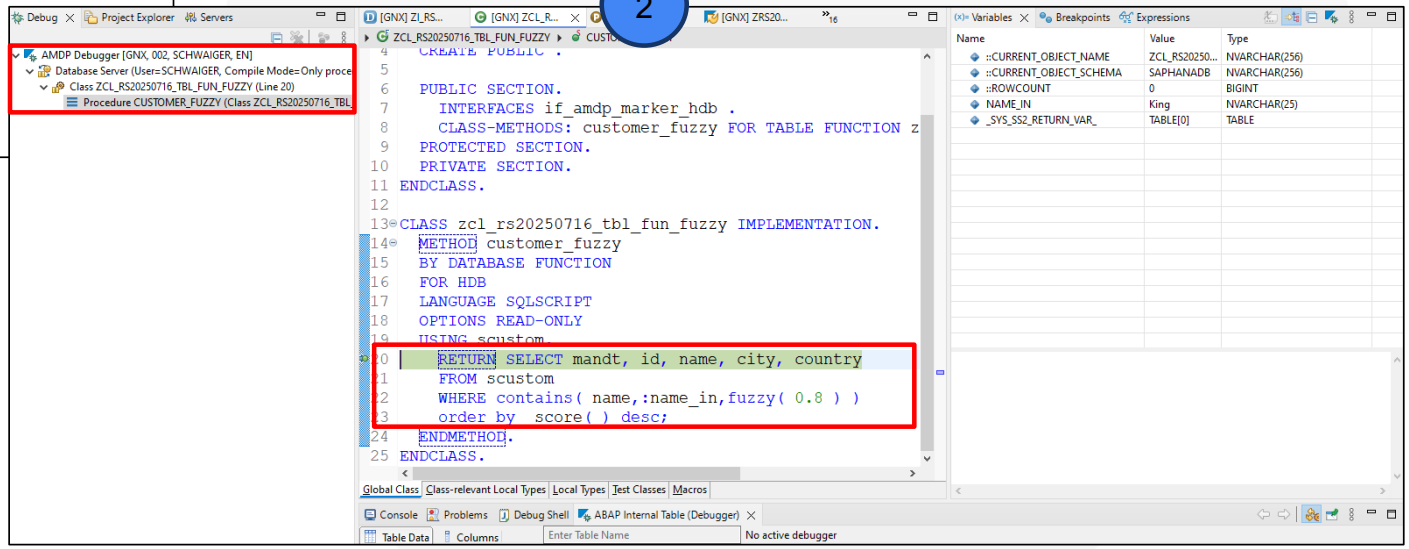
```
  INSERT cl_abap_dbfeatures=>amdp_table_function INTO TABLE gt_feature_set.
  IF cl_abap_dbfeatures=>use_features( gt_feature_set ) = abap_false.
    MESSAGE 'Fehler' TYPE 'E'.
  ENDIF.
* Alternative with VALUE expression (Rel. 7.40)
*IF cl_abap_dbfeatures=>use_features(
* VALUE #( ( cl_abap_dbfeatures=>amdp_table_function ) )
* ) = abap_false.
* MESSAGE e001(Z<Initials><ANSI Date>).
*ENDIF.
```

Eclipse Debugger for DB Procedures / Functions

```
13 CLASS zcl_rs20250716_tbl_fun_fuzzy IMPLEMENTATION.
14   METHOD customer_fuzzy
15   BY DATABASE FUNCTION
16   FOR HDB
17   LANGUAGE SQLSCRIPT
18   OPTIONS READ-ONLY
19   USING scustom.
20     RETURN SELECT mandt, id, name, city, country
21     FROM scustom
22     WHERE contains( name,:name_in,fuzzy(
23     order by  score( ) desc;
24   ENDMETHOD.
25 ENDCLASS.
```

1

2

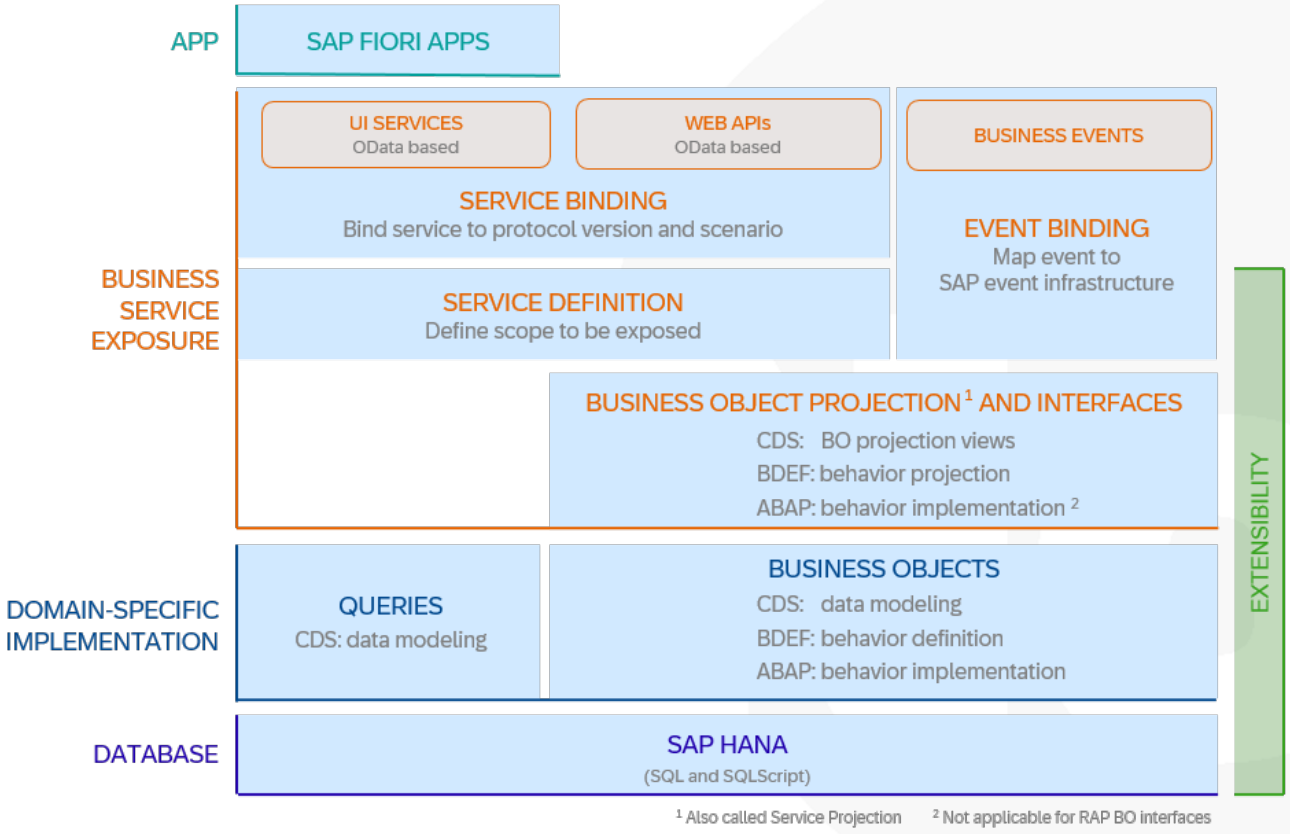


AMDP and CDS - Fuzzy, Linguistic Search

14

RAP in 20 mins

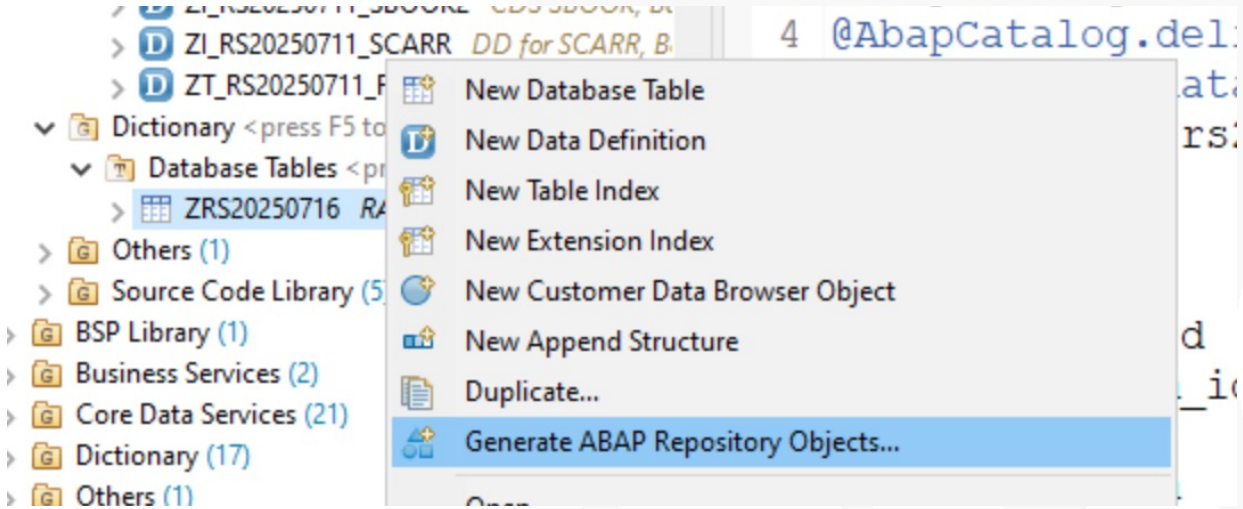
RAP Big Picture



Database Table

```
@EndUserText.label : 'RAP Table'
@AbapCatalog.enhancement.category : #NOT_EXTENSIBLE
@AbapCatalog.tableCategory : #TRANSPARENT
@AbapCatalog.deliveryClass : #A
@AbapCatalog.dataMaintenance : #RESTRICTED
define table zrs20250716 {
  key client             : abap.clnt not null;
  key uuid               : sysuuid_x16 not null;
  carrier_id             : /dmo/carrier_id;
  connection_id          : /dmo/connection_id;
  airport_from_id        : /dmo/airport_from_id;
  city_from              : city;
  country_from           : land1;
  airport_to_id          : /dmo/airport_to_id;
  city_to                : city;
  country_to             : land1;
  local_created_by       : abp_creation_user;
  local_created_at       : abp_creation_tstmpl;
  local_last_changed_by  : abp_locinst_lastchange_user;
  local_last_changed_at  : abp_locinst_lastchange_tstmpl;
  last_changed_at        : abp_lastchange_tstmpl;

}
```

Generate ABAP Repository Objects

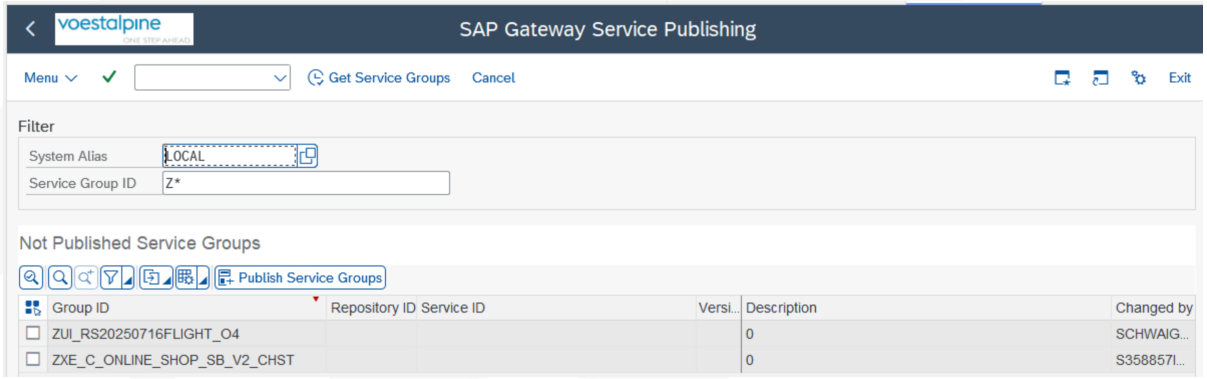


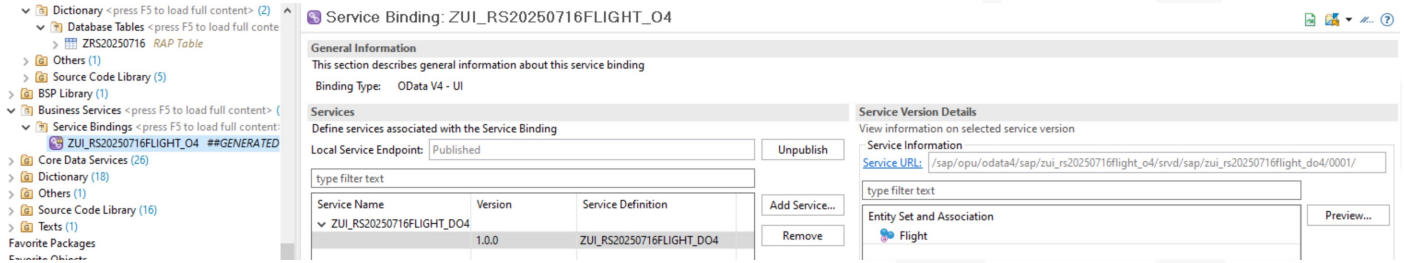
Data Generator
Naming suggestions

Field	Value
General → Referenced Object	`Z##FLIGHT`
General → Project Name	*<empty>*
General → Artifacts Prefix	*<empty>*
General → Artifacts Suffix	*<empty>*
Business Object → Data Model → CDS Entity Name	`ZR_##Flight`
Business Object → Data Model → CDS Entity Alias	`Flight`
Business Object → Behavior → Behavior Implementation Class	`ZBP_R_##FLIGHT`
Business Object → Behavior → Draft Table Name	`Z##FLIGHT_D`
Service Projection → Service Projection Entity → CDS Entity Name	`ZC_##Flight`
Service Projection → Service Projection Behavior → Behavior Implementation Class	`ZBP_C_##FLIGHT`
Business Service → Service Definition → Service Definition Name	`ZUI_##FLIGHT_O4`
Business Service → Service Binding → Service Binding Name	`ZUI_##FLIGHT_O4`
Business Service → Service Binding → Binding Type	`OData V4 - UI`

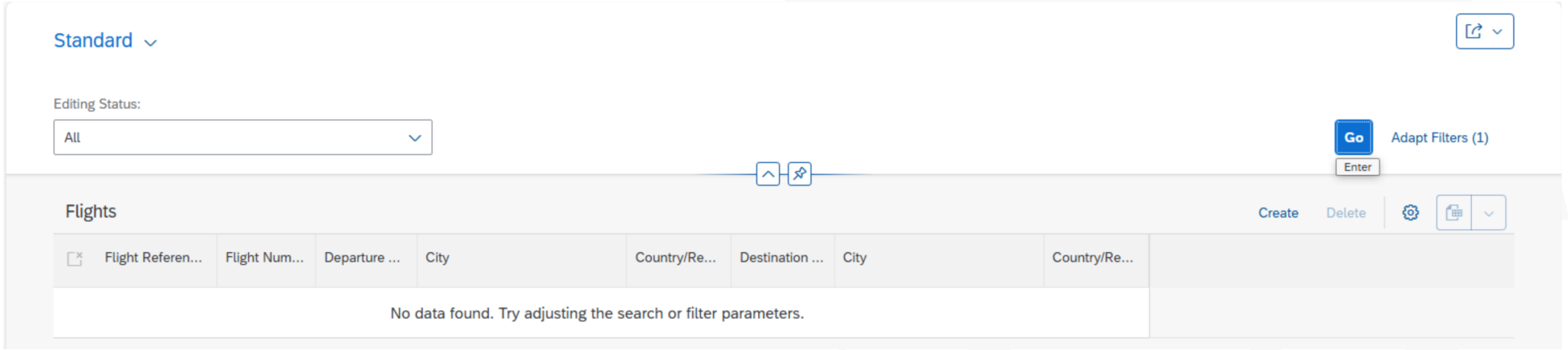
Register O4 Service

- /IWFND/V4_ADMIN (Transaction)
- Publish Service Groups
- Search Service Group: LOCAL, Z*
- Publish service group
- Refresh





Preview App

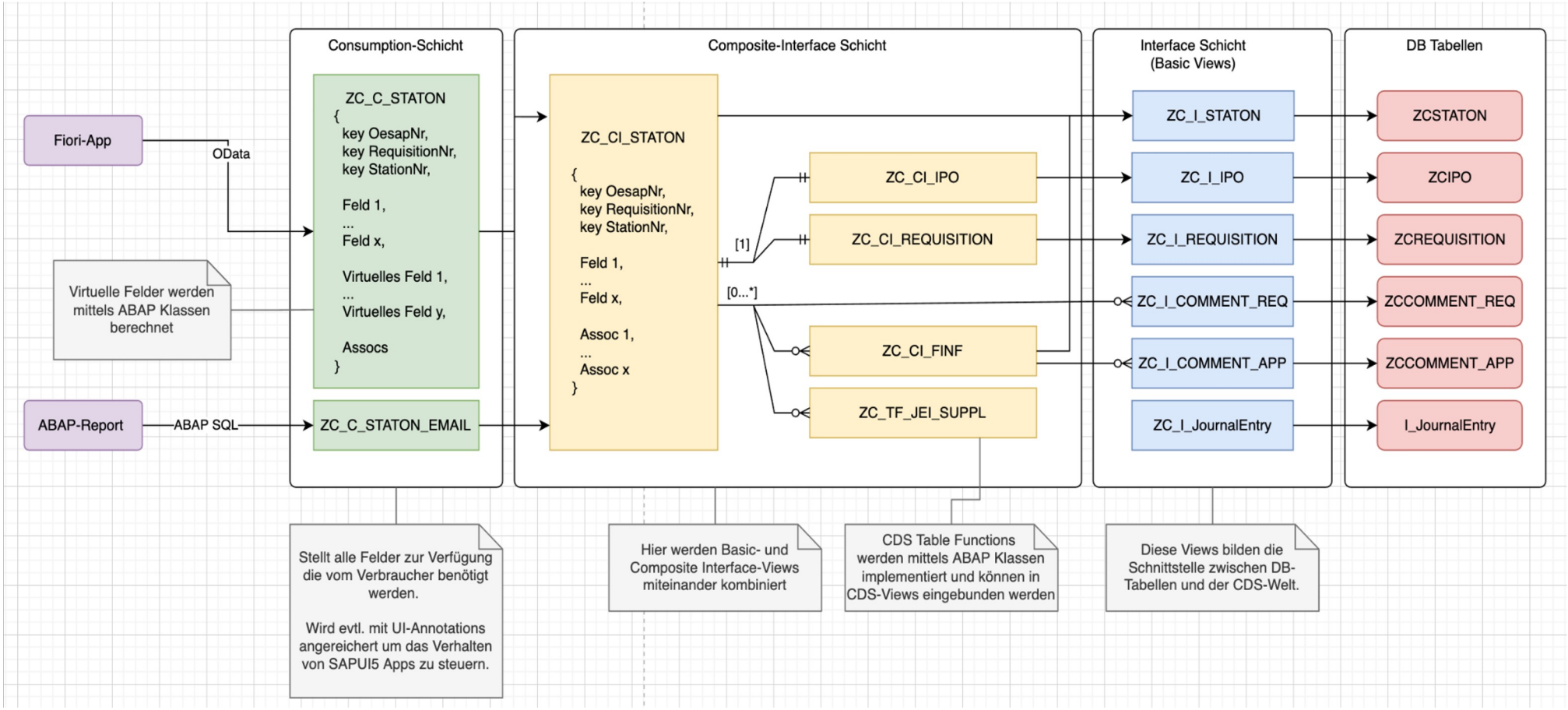

Standard
Editing Status:
All
Go
Enter
Adapt Filters (1)
Flights
Create
Delete
Flight Referen...
Flight Num...
Departure ...
City
Country/Re...
Destination ...
City
Country/Re...
No data found. Try adjusting the search or filter parameters.

RAP

- Create RAP

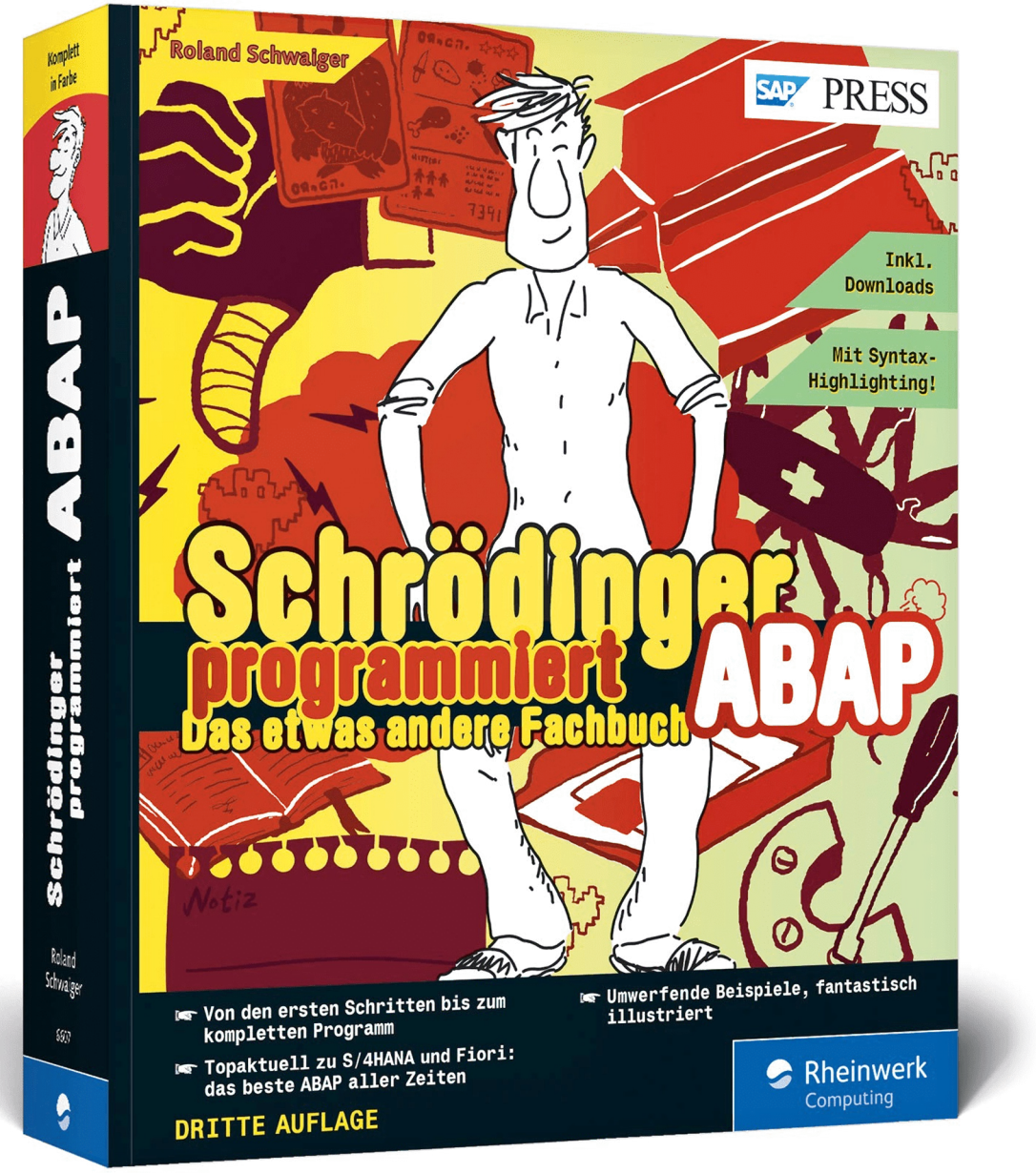
15

Real World Example



www.nor.gmbh office@nor.gmbh NOR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

DONE!



www.nor.gmbh office@nor.gmbh NoR GmbH - Hüttfeldstrasse 17 – A-6345 Kössen FN: 475603t UID: ATU72533127

Anhang