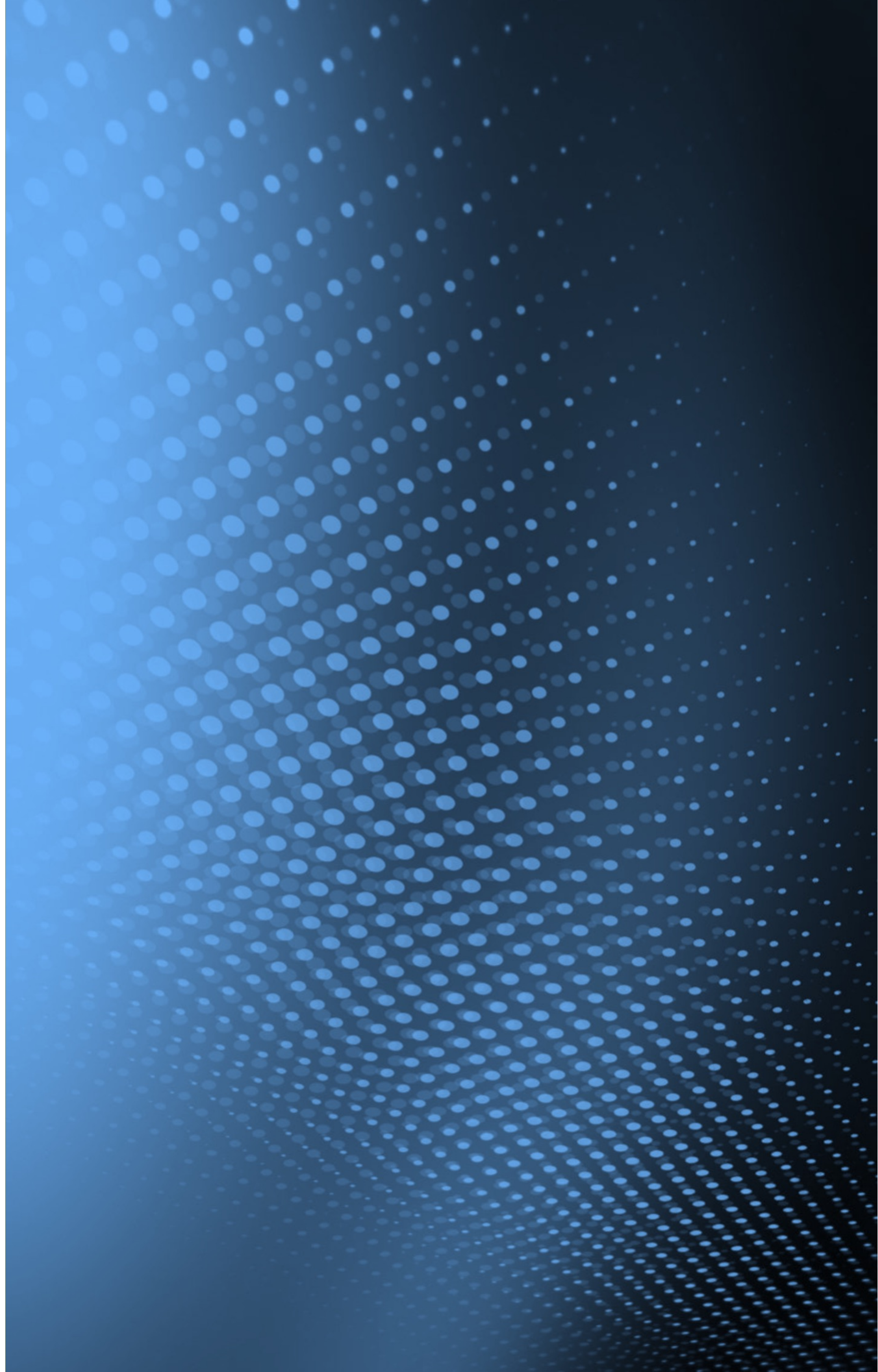
ABAP Core Data Services
Deep Dive on New Features and Capabilities

Konrad Gaerdes, Andrea Schlotthauer
User Assistance Developers at SAP BTP ABAP, SAP SE

June 6, 2024

Agenda



THE ROLE OF ABAP CDS IN THE ABAP CLOUD DEVELOPMENT MODEL

Use cases

ABAP CDS is part of the golden path technology recommendation for data modeling in the ABAP Cloud development model. Using ABAP Cloud means using ABAP CDS.

ABAP CDS IS THE FOUNDATION FOR …

SERVICE BASED USE CASES

- Transactional Apps with the ABAP RESTful Application Programming Model
- Multidimensional data models and analytical reports
- Data integration scenarios e.g., via SQL services
- Read-only scenarios based on CDS entities
- Business eventing
- Enterprise Search

LOCAL CONSUMPTION USE CASES VIA

- ABAP SQL
- other CDS objects
- ABAP Managed Database Procedures (AMDP)
- major ABAP frameworks

ABAP CDS – overview of supported CDS entity types[1]

Category	ABAP CDS Entity Type	ABAP CDS Statement

Standard view building	CDS View Entity
	CDS Projection View
	CDS Projection View – Analytical Query
	CDS DDIC-based View *(deprecated)*
		`DEFINE VIEW ENTITY`
		`DEFINE VIEW ENTITY AS PROJECTION ON`
		`DEFINE TRANSIENT VIEW ENTITY AS PROJECTION ON`
		`DEFINE VIEW` *(deprecated)*

Advanced view building – External implementation	CDS Custom Entity
	CDS Abstract Entity
		`DEFINE CUSTOM ENTITY`
		`DEFINE ABSTRACT ENTITY`

Advanced view building – SAP HANA breakout	CDS Table Function
	CDS Hierarchy
	CDS Scalar Function[2]
		`DEFINE TABLE FUNCTION`
		`DEFINE HIERARCHY`
		`DEFINE SCALAR FUNCTION`

Type definition	CDS Simple Type[2]
	CDS Enumerated Type[2]
		`DEFINE SIMPLE TYPE`
		`DEFINE TYPE ENUM`

Transactional behavior definition of business objects built with the ABAP RESTful Application Programming Model	CDS Behavior Definition (and Projection)	`DEFINE BEHAVIOR`

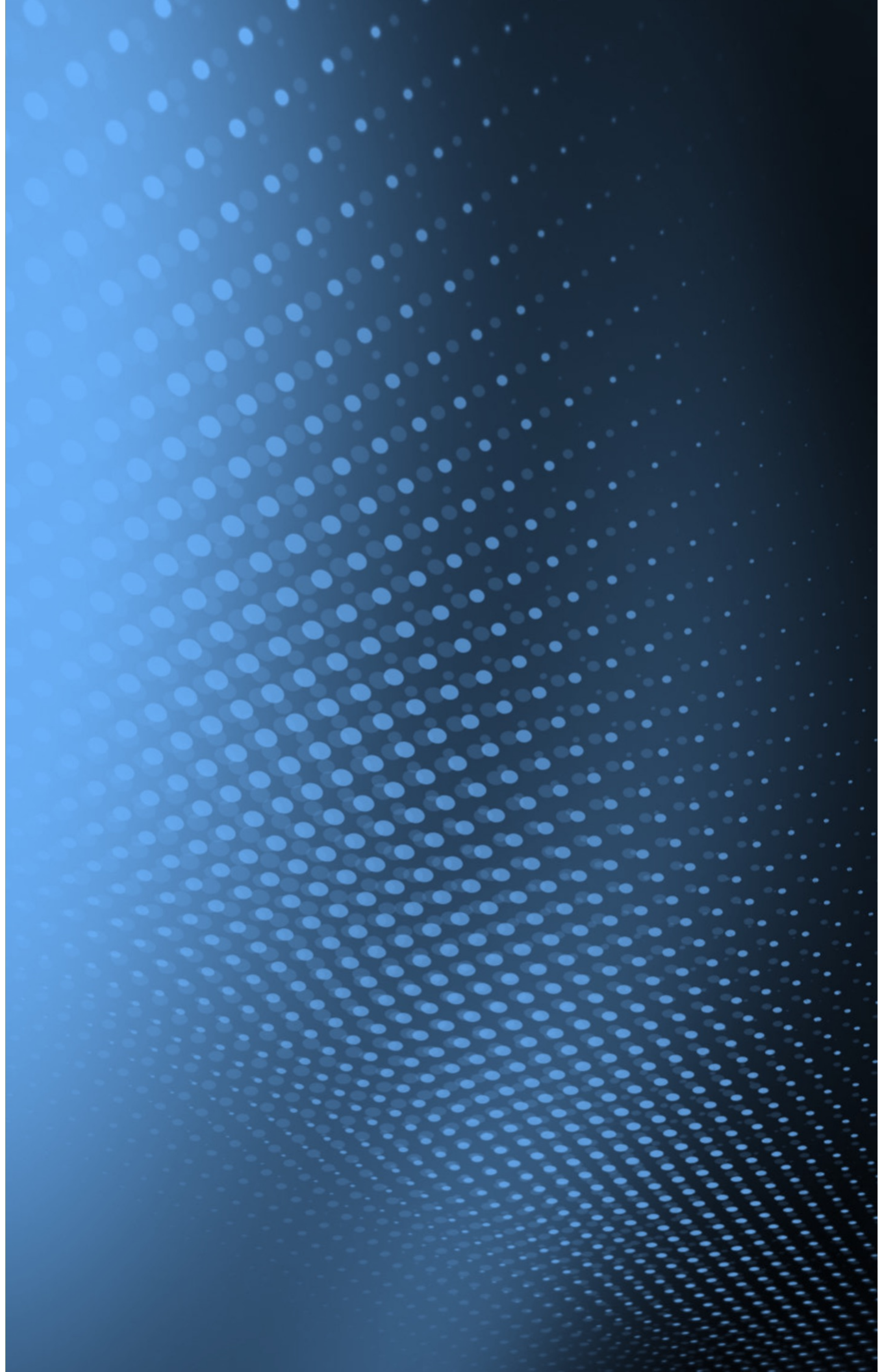
Modification-free extension	CDS View Entity Extension
	CDS Custom Entity Extension
	CDS Abstract Entity Extension
	CDS DDIC-based View Extension *(deprecated)*
	CDS Metadata Extension
	CDS Behavior Extension
		`EXTEND VIEW ENTITY`
		`EXTEND CUSTOM ENTITY`
		`EXTEND ABSTRACT ENTITY`
		`EXTEND VIEW` *(deprecated)*
		`ANNOTATE VIEW`
		`EXTEND BEHAVIOR`

Access control definition	CDS Access Control	`DEFINE ROLE`

[1] Status from Q2/2024 | [2] Available on-prem starting with Release 2023

NEW KIDS ON THE BLOCK

CDS SIMPLE TYPES



CDS simple types

CDS SIMPLE TYPES …

are user-defined elementary data types in ABAP CDS

can be used instead of a DDIC data element

are available as of 2302 in SAP BTP ABAP Environment, and SAP S/4HANA 2023

More information - Docu | Blog Post | Video

DDIC data elements and CDS simple types

	DDIC data element	CDS simple types
Availability	Typing in DDIC, CDS, and ABAP	Typing in CDS and in ABAP
Metadata	Technical settings in SE11	CDS annotations for **domain-specific metadata**
Definition	Form-based editor and DDIC DDL	Syntax-based definition in ADT
Framework support	Supported by technologies such as Web Dynpro and SAP GUI	Supported by frameworks such as the RAP Query Engine and the ABAP Analytical Engine

CDS simple types - definition

TYPING WITH A BUILT-IN DATA TYPE

```
@EndUserText.heading: 'First name'
define type demo_bt_first_name: abap.char( 30 );
```

STACKING OF SIMPLE TYPES

```
define type demo_simple_type_2 : demo_simple_type;
```

SIMPLE TYPE BASED ON A DATA ELEMENT

```
define type demo_simple_type_de : demo_destination;
```

CDS simple types – usage in a CDS view entity

CDS VIEW ENTITY

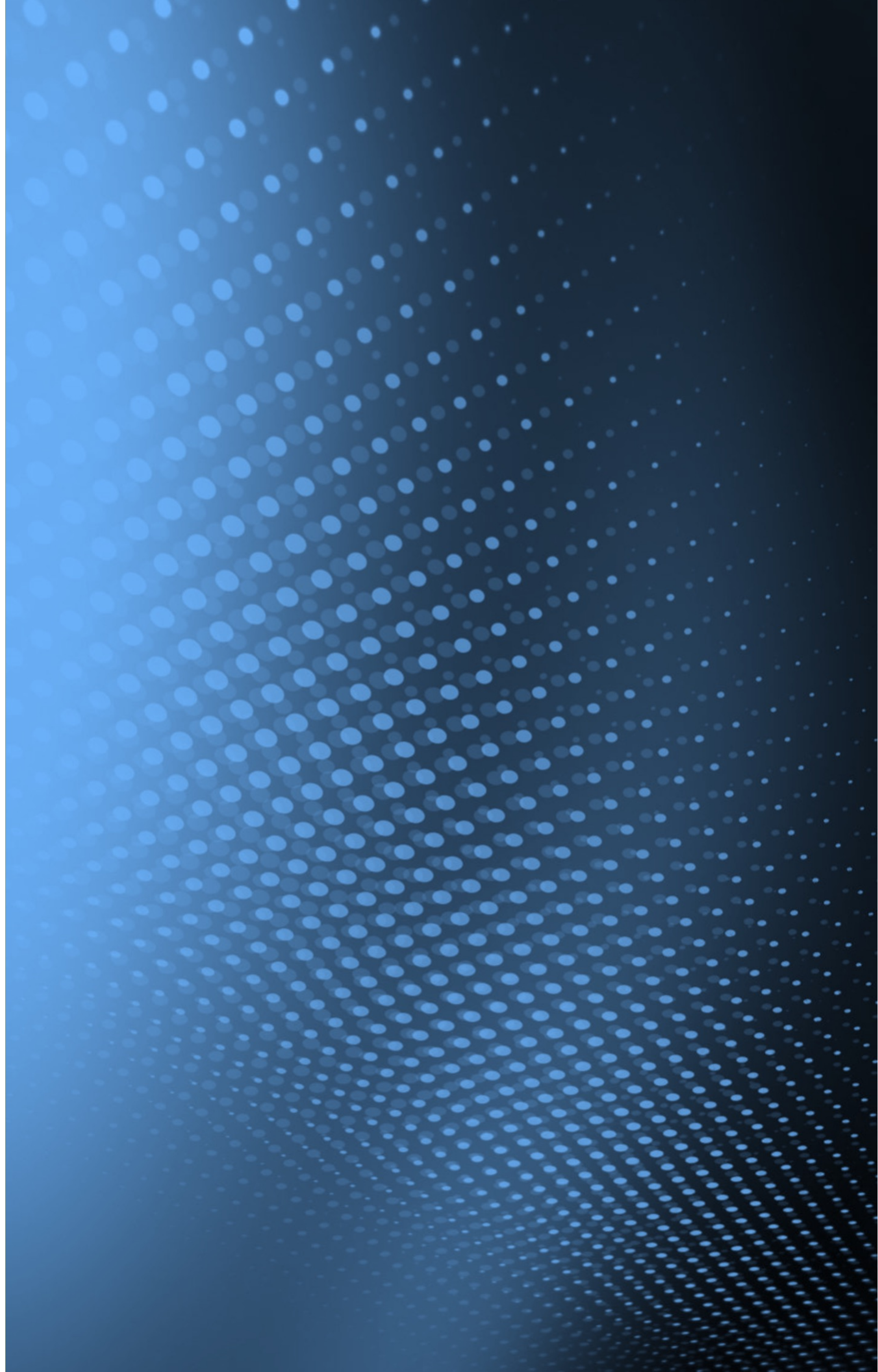
```
define view entity DEMO_CDS_SIMPLE_TYPE_USAGE
  with parameters
    p1 : demo_simple_type,
    p2 : abap.int4

  as select from demo_expressions
{
...
cast( char2 as demo_simple_type_de ) as cast_bt
}
```

CDS simple types – usage in ABAP

ABAP

```
DATA MyType TYPE demo_simple_type.
MyType = 1.
```

CDS ENUMERATED TYPES



What is an enumerated type?

An enumerated type is a data type that specifies a fixed set of values.
Objects typed with an enumerated type can have only one of the predefined values.

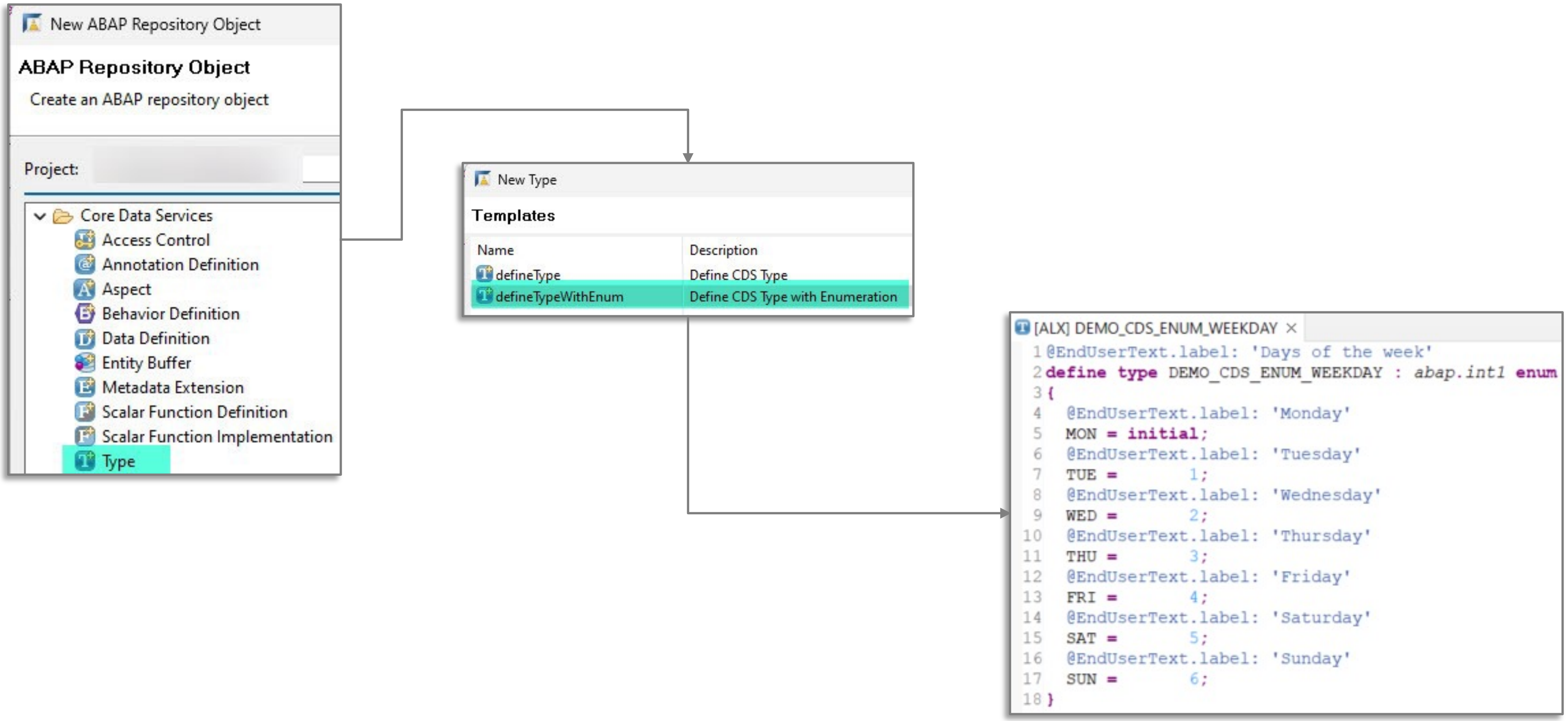
More information - Docu | Blog Post

CDS enumerated types - motivation

CDS ENUMERATED TYPES MAKES ENUMERATIONS GLOBALLY AVAILABLE AND REUSABLE IN DIFFERENT CONTEXTS

- CDS enumerated types vs. DDIC domains with fixed values
- CDS enumerated types vs. ABAP enumerations
- Leveraging CDS qualities

CDS enumerated types - definition example (1/2)



New ABAP Repository Object

ABAP Repository Object

Create an ABAP repository object

Project:

- Core Data Services
  - Access Control
  - Annotation Definition
  - Aspect
  - Behavior Definition
  - Data Definition
  - Entity Buffer
  - Metadata Extension
  - Scalar Function Definition
  - Scalar Function Implementation
  - Type

New Type

Templates

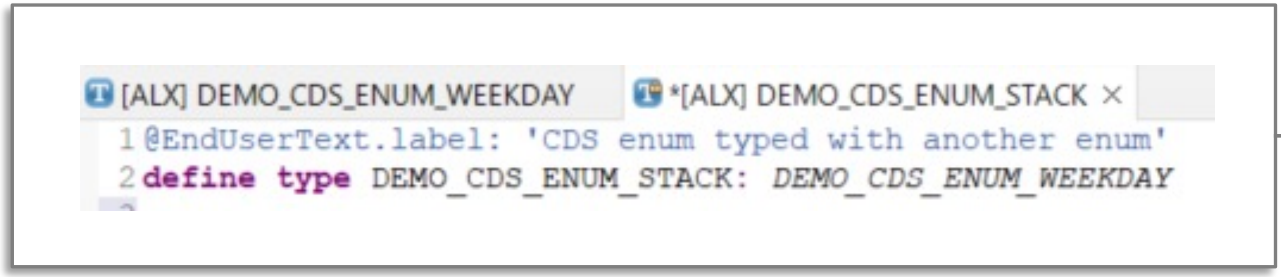
Name	Description
defineType	Define CDS Type
defineTypeWithEnum	Define CDS Type with Enumeration

[ALX] DEMO_CDS_ENUM_WEEKDAY

```
@EndUserText.label: 'Days of the week'
define type DEMO_CDS_ENUM_WEEKDAY : abap.int1 enum
{
  @EndUserText.label: 'Monday'
  MON = initial;
  @EndUserText.label: 'Tuesday'
  TUE =       1;
  @EndUserText.label: 'Wednesday'
  WED =       2;
  @EndUserText.label: 'Thursday'
  THU =       3;
  @EndUserText.label: 'Friday'
  FRI =       4;
  @EndUserText.label: 'Saturday'
  SAT =       5;
  @EndUserText.label: 'Sunday'
  SUN =       6;
}
```

CDS enumerated types - definition example (2/2)

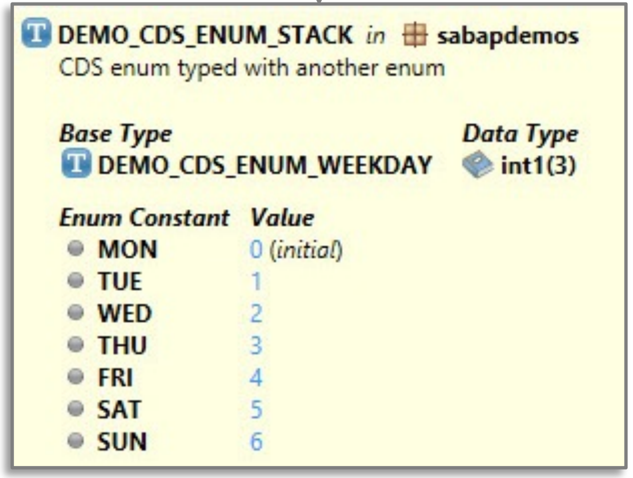
STACKING OF CDS ENUMERATED TYPES



[ALX] DEMO_CDS_ENUM_WEEKDAY   *[ALX] DEMO_CDS_ENUM_STACK

```
@EndUserText.label: 'CDS enum typed with another enum'
define type DEMO_CDS_ENUM_STACK: DEMO_CDS_ENUM_WEEKDAY
```

Code element info (F2)



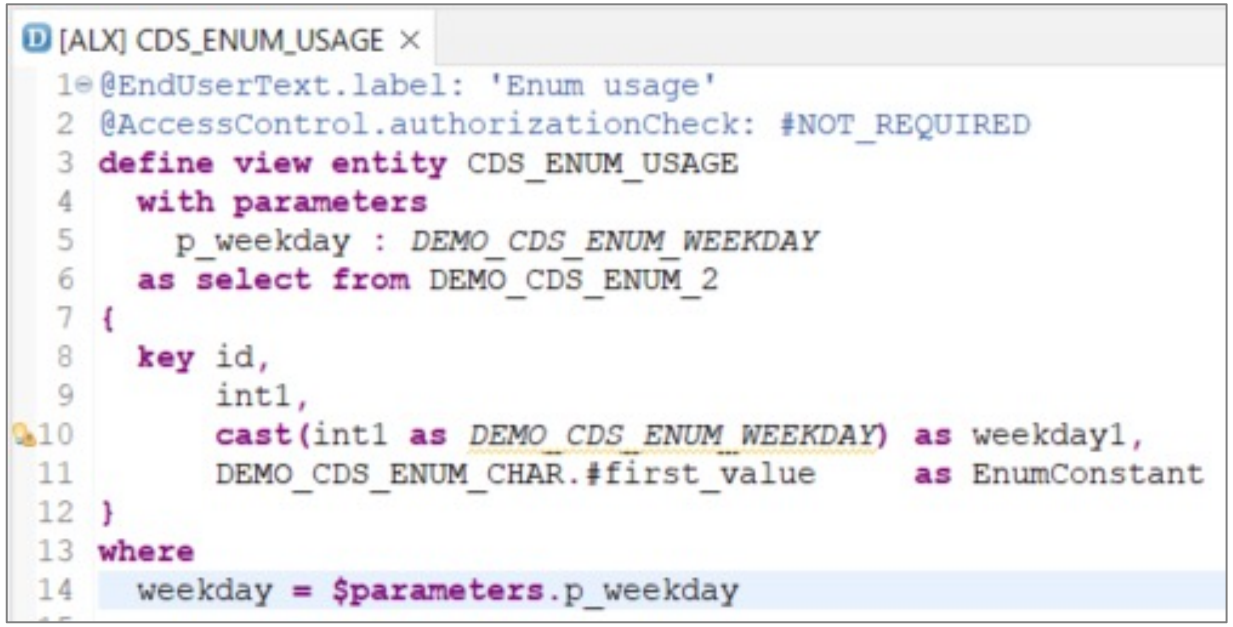
DEMO_CDS_ENUM_STACK *in* sabapdemos
CDS enum typed with another enum

Base Type	***Data Type***
DEMO_CDS_ENUM_WEEKDAY	int1(3)

Enum Constant	***Value***
MON	0 *(initial)*
TUE	1
WED	2
THU	3
FRI	4
SAT	5
SUN	6

CDS enumerated types - usage example (1/3)

USING CDS ENUMERATED TYPES IN CDS VIEW ENTITIES



```
@EndUserText.label: 'Enum usage'
@AccessControl.authorizationCheck: #NOT_REQUIRED
define view entity CDS_ENUM_USAGE
  with parameters
    p_weekday : DEMO_CDS_ENUM_WEEKDAY
  as select from DEMO_CDS_ENUM_2
{
  key id,
      int1,
      cast(int1 as DEMO_CDS_ENUM_WEEKDAY) as weekday1,
      DEMO_CDS_ENUM_CHAR.#first_value     as EnumConstant
}
where
  weekday = $parameters.p_weekday
```

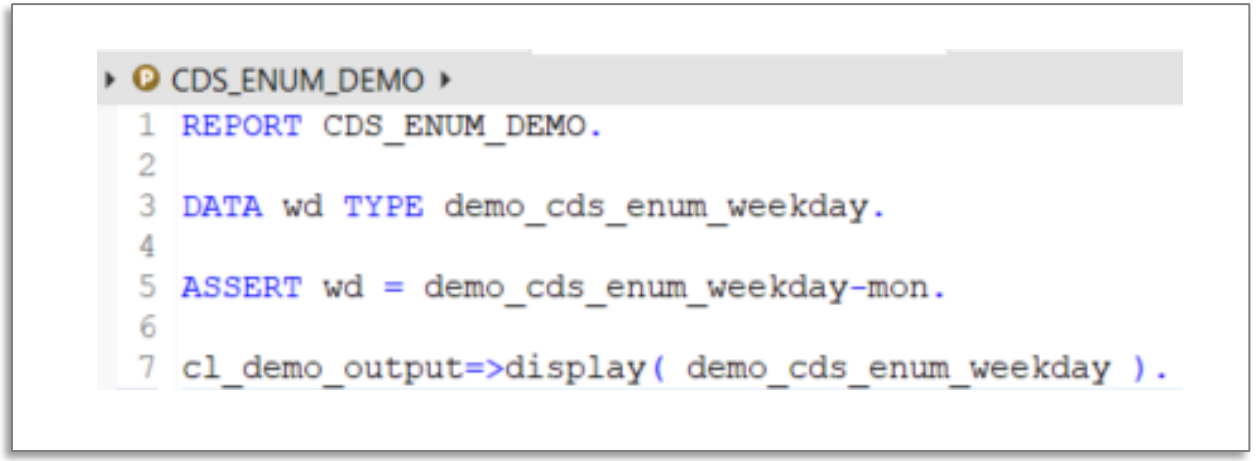
CDS ENUMERATED TYPES CAN ALSO BE USED …

in CDS projection views, CDS custom entities, and CDS abstract entities

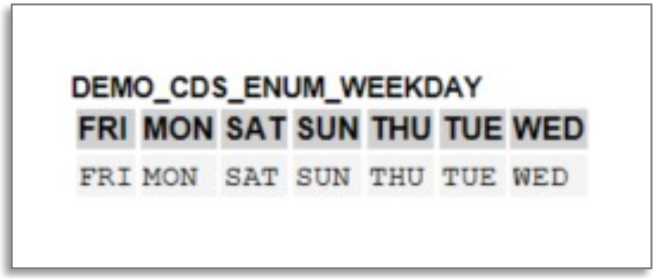
for casting, typing, parameter passing, as operand of a CASE expression, and operand in the SELECT list

CDS enumerated types - usage example (2/3)

USING CDS ENUMERATED TYPES IN ABAP


CDS_ENUM_DEMO

```
REPORT CDS_ENUM_DEMO.

DATA wd TYPE demo_cds_enum_weekday.

ASSERT wd = demo_cds_enum_weekday-mon.

cl_demo_output=>display( demo_cds_enum_weekday ).
```

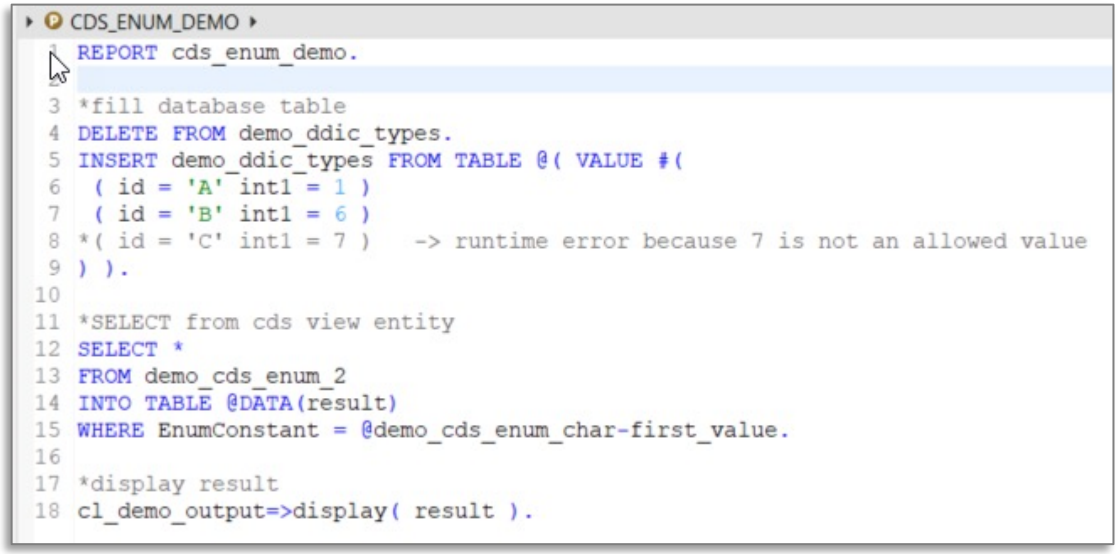

DEMO_CDS_ENUM_WEEKDAY

FRI	MON	SAT	SUN	THU	TUE	WED
FRI	MON	SAT	SUN	THU	TUE	WED

CDS enumerated types - usage example (3/3)

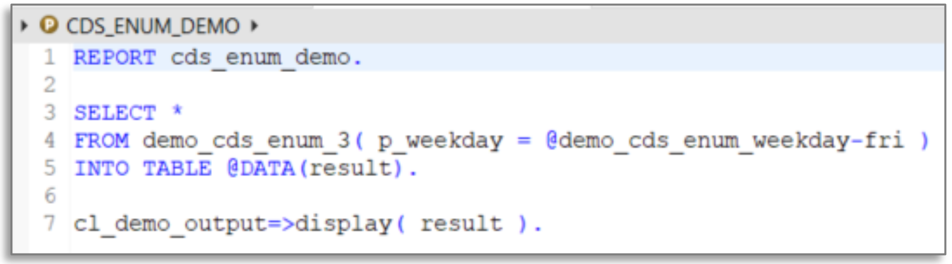
USING CDS ENUMERATED TYPES IN ABAP SQL

Comparison in a WHERE-clause



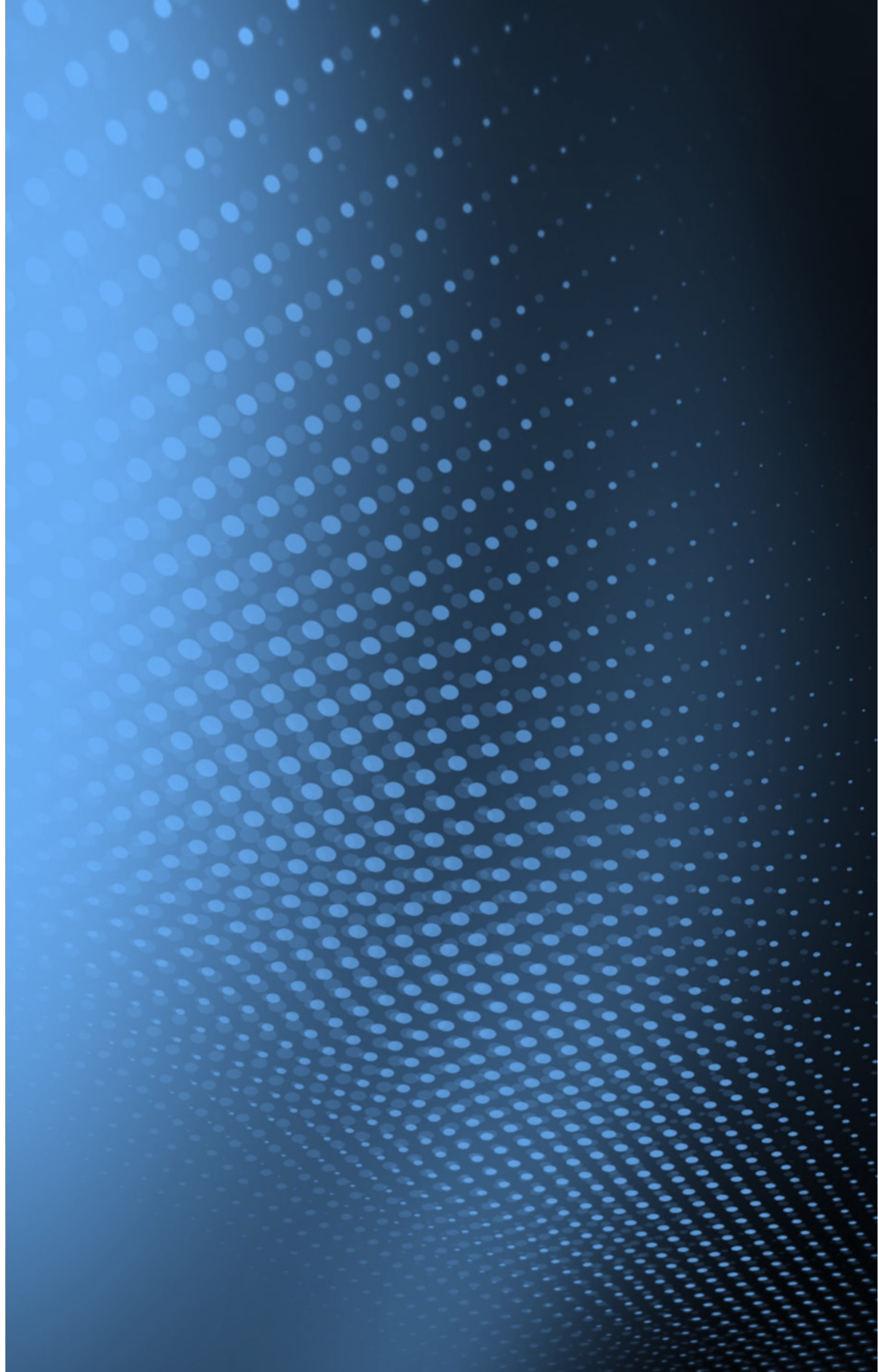
```
REPORT cds_enum_demo.

*fill database table
DELETE FROM demo_ddic_types.
INSERT demo_ddic_types FROM TABLE @( VALUE #(
 ( id = 'A' int1 = 1 )
 ( id = 'B' int1 = 6 )
*( id = 'C' int1 = 7 )   -> runtime error because 7 is not an allowed value
) ).

*SELECT from cds view entity
SELECT *
FROM demo_cds_enum_2
INTO TABLE @DATA(result)
WHERE EnumConstant = @demo_cds_enum_char-first_value.

*display result
cl_demo_output=>display( result ).
```

Parameter passing



```
REPORT cds_enum_demo.

SELECT *
FROM demo_cds_enum_3( p_weekday = @demo_cds_enum_weekday-fri )
INTO TABLE @DATA(result).

cl_demo_output=>display( result ).
```

CDS SCALAR FUNCTIONS



PUBLIC

CDS scalar functions

A SQL-based scalar function is a user-defined function that accepts multiple input parameters and returns exactly one scalar value

It allows developers to encapsulate complex algorithms into manageable, reusable code that can be used in CDS entities and in ABAP

A scalar function is linked with an AMDP function in which it is implemented using SQLScript

More information - Docu | Blog Post | Video

DEMO EXAMPLE

Is a delivery subject to import duties?

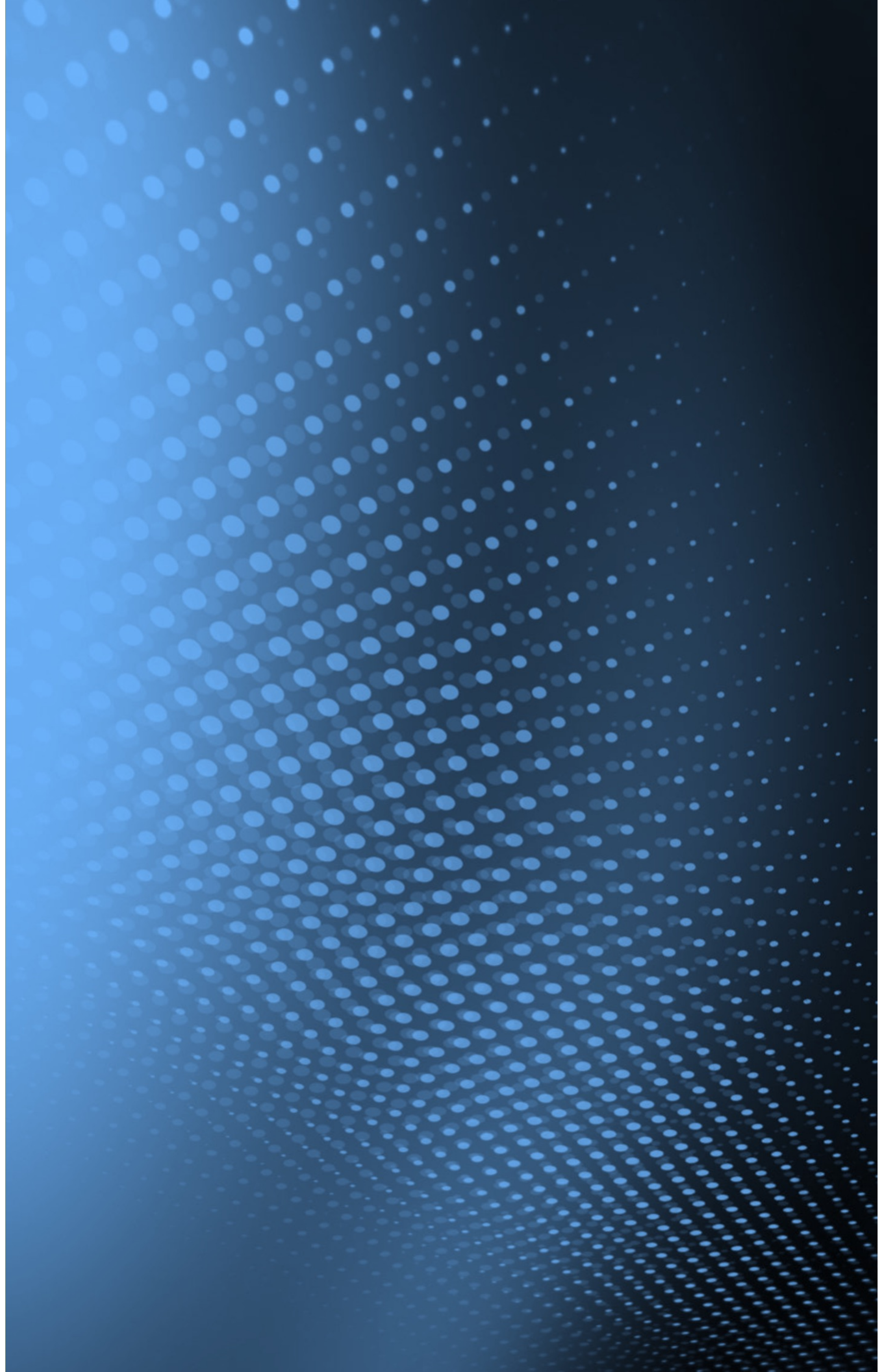
PACKAGING INFORMATION

```
distributor_id;
  customer_id;
  …
  import_duty;
  …
```

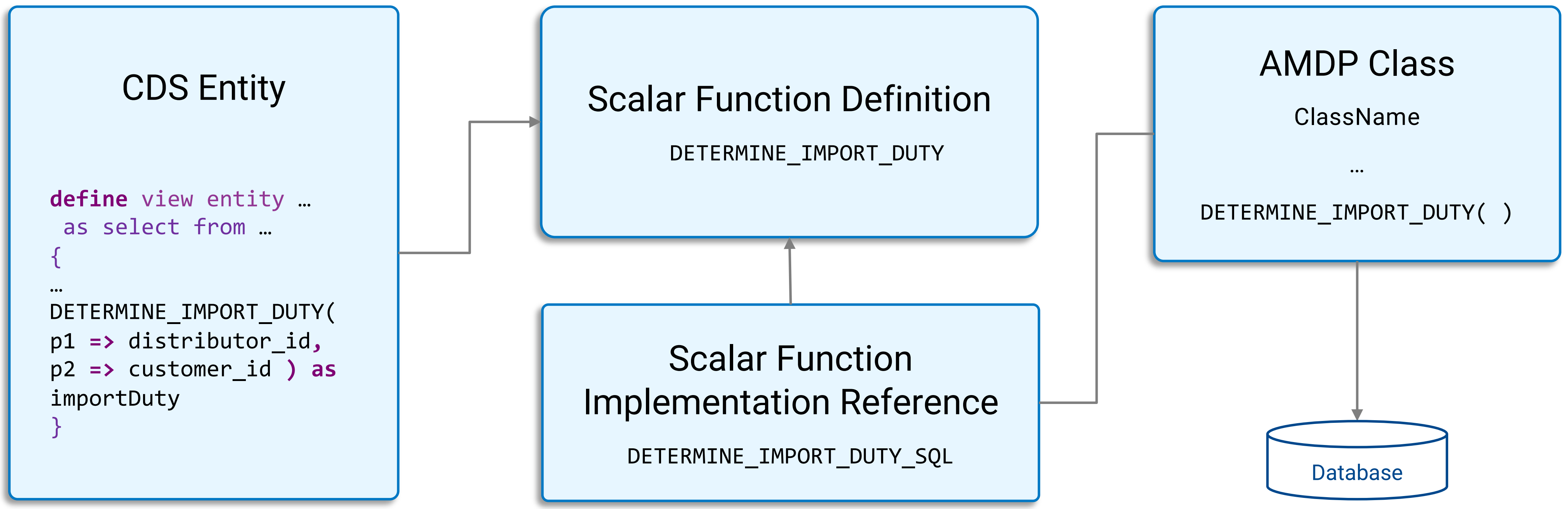
FUNCTION

```
determine_import_duty( distributor_id, customer_id )
```

PUBLIC



Design time



CDS Entity
Scalar Function Definition
DETERMINE_IMPORT_DUTY
AMDP Class
ClassName
…
DETERMINE_IMPORT_DUTY( )
Scalar Function Implementation Reference
DETERMINE_IMPORT_DUTY_SQL
Database

```
define view entity …
 as select from …
{
…
DETERMINE_IMPORT_DUTY(
p1 => distributor_id,
p2 => customer_id ) as
importDuty
}
```

CDS scalar functions - definition

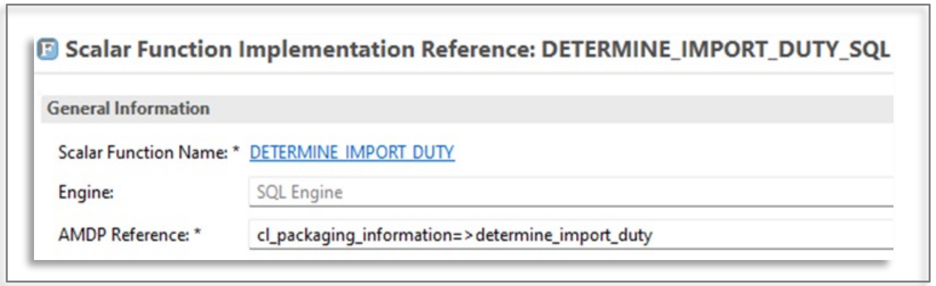
1

DEFINE THE CDS SCALAR FUNCTION DEFINITION

```
define scalar function DETERMINE_IMPORT_DUTY
      with parameters
        distributor_id: abap.int2,
        customer_id: abap.int2
      returns boolean;
```

2

DEFINE THE CDS SCALAR FUNCTION IMPLEMENTATION REFERENCE


Scalar Function Implementation Reference: DETERMINE_IMPORT_DUTY_SQL

General Information

Scalar Function Name: *	DETERMINE_IMPORT_DUTY
Engine:	SQL Engine
AMDP Reference: *	cl_packaging_information=>determine_import_duty

3

AMDP FUNCTION IMPLEMENTATION

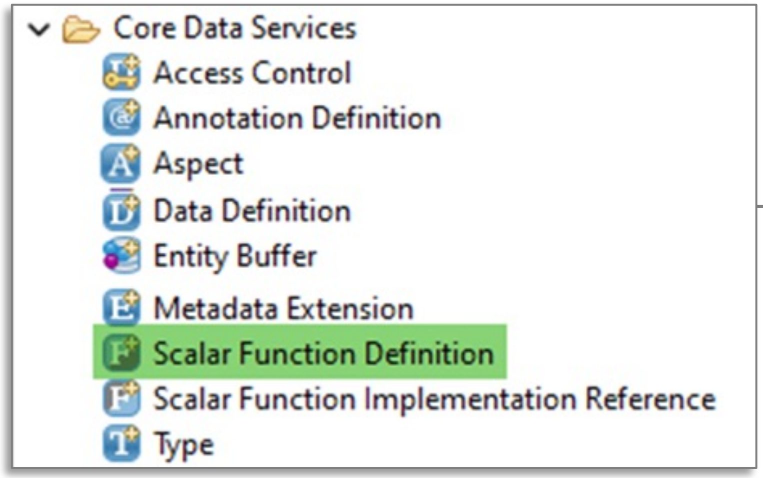
```
CLASS cl_packaging_information
IMPLEMENTATION.
   METHOD determine_import_duty BY DATABASE
FUNCTION
   FOR HDB
   LANGUAGE SQLSCRIPT
   OPTIONS READ-ONLY.
   //implementation goes here
   ENDMETHOD.
```

CDS scalar functions - usage example

USING CDS SCALAR FUNCTIONS IN A CDS VIEW ENTITY

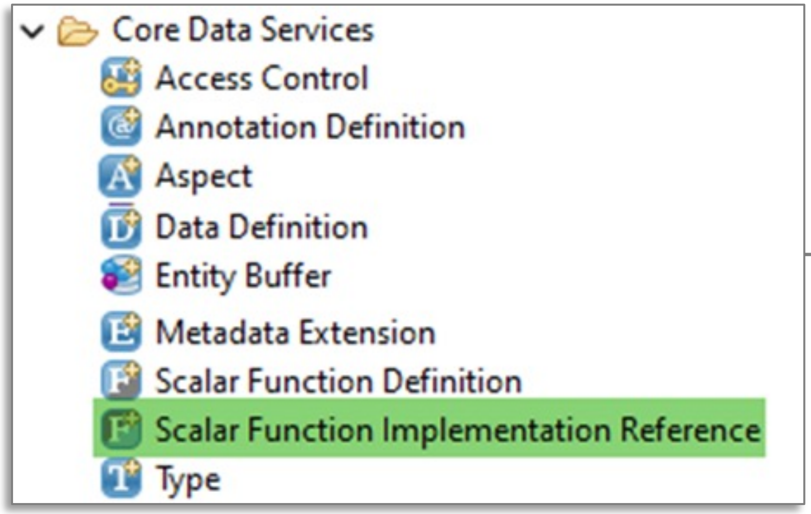
```
define view entity abapconf_packaging_information
  with parameters
    distributor_id : abap.int2,
    customer_id : abap.int2
  as select from abapconf_I_CUSTOMER  as Customer
    inner join abapconf_I_DISTRIBUTOR as Distributor on
      Distributor.DistributorId = $parameters.distributor_id
{
…
determine_import_duty( distributer_id => $parameters.distributor_id,
                          customer_id => $parameters.customer_id ) as
                             ImportDuty

}
```

CDS scalar function definition

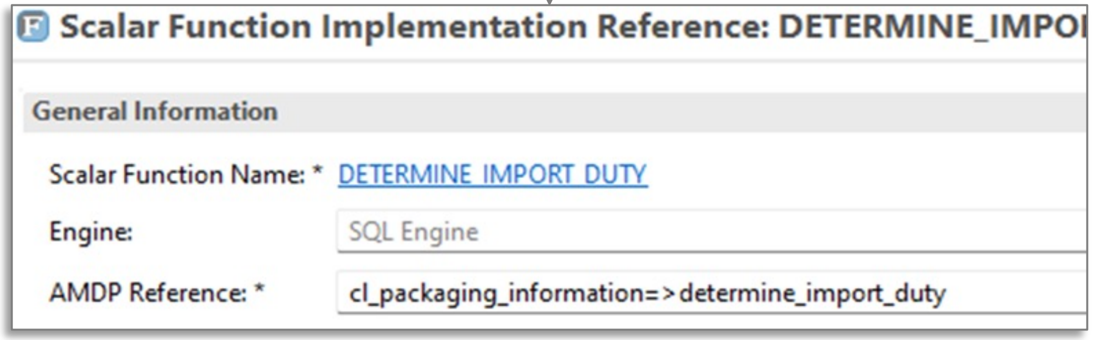

Core Data Services
Access Control
Annotation Definition
Aspect
Data Definition
Entity Buffer
Metadata Extension
Scalar Function Definition
Scalar Function Implementation Reference
Type

```
define scalar function determine_import_duty
  with parameters
    distributer_id: abap.int2,
    customer_id: abap.int2
  returns Boolean
```

CDS scalar function implementation reference



- Core Data Services
  - Access Control
  - Annotation Definition
  - Aspect
  - Data Definition
  - Entity Buffer
  - Metadata Extension
  - Scalar Function Definition
  - Scalar Function Implementation Reference
  - Type



Scalar Function Implementation Reference: DETERMINE_IMPOI

General Information

Scalar Function Name: *	DETERMINE_IMPORT_DUTY
Engine:	SQL Engine
AMDP Reference: *	cl_packaging_information=>determine_import_duty

Implementation in an AMDP method

IMPLEMENTING A CDS SCALAR FUNCTION IN AN AMDP METHOD

```
METHOD determine_import_duty BY DATABASE FUNCTION
                             FOR HDB
                             LANGUAGE SQLSCRIPT
                             OPTIONS READ-ONLY
    USING abapconf_i_customer
          abapconf_i_distributor
          abapconf_i_country.
  declare is_distr_country_eu_member, is_cust_country_eu_member char( 1 );
  declare distr_country_code, cust_country_code char( 2 );
  SELECT country INTO distr_country_code FROM abapconf_i_distributor WHERE distributorid = distributer_id;
  SELECT country INTO cust_country_code FROM abapconf_i_customer WHERE customerid = customer_id;

  SELECT iseuropeanunionmember INTO is_distr_country_eu_member FROM abapconf_i_country WHERE country
    =    distr_country_code;
  SELECT iseuropeanunionmember INTO is_cust_country_eu_member FROM abapconf_i_country WHERE country = cust_country_code;
  IF NOT ( is_distr_country_eu_member = 'X' AND is_cust_country_eu_member = 'X' )
  then
    result = 'X';
  END if;
ENDMETHOD.
```

ROADMAP

ABAP CDS roadmap

CDS aspects

CDS external entities

CDS buffer propagation and buffer delegation

CDS table entities

Writable views

This is the current state of planning and may be changed by SAP at any time without notice.

KNOWLEDGE RESOURCES FOR ABAP CDS

Further information

- ABAP Data Models | SAP Help Portal
- ABAP Core Data Services in the ABAP Keyword Documentation (sap.com)
- ABAP CDS Development Tools: User Guide | SAP Help Portal
- ABAP CDS Feature Tables
- ABAP CDS Glossary
- Getting Started with ABAP Core Data Services (CDS) - SAP Community
- ABAP CDS Cheat Sheet
- Many of the demonstrations shown are delivered as part of the SABAPDEMOS package in all on-premise SAP systems and they are part of the ABAP Keyword Documentation

Thank you.

Contact information:
Andrea.Schlotthauer@sap.com
Konrad.Gaerdes@sap.com