



# ABAP CDS

Webinar, 30.09.2022, 10:00 Uhr



# Sören Schlegel

Managing Consultant at j&s-soft gmbh

eMail: [soeren.schlegel@js-soft.com](mailto:soeren.schlegel@js-soft.com)

Twitter: [@SoSchlegel87](https://twitter.com/SoSchlegel87)

## Themenschwerpunkte

- Konzeption & Management von Entwicklungsprojekten
- SAP Architektur moderner Anwendungen
- SAP Change Management and S/4HANA
- SAP CDS, oData und Cloud



# Johann Föbleitner

Senior Consultant at Cadaxo

eMail: [johann.foessleitner@cadaxo.com](mailto:johann.foessleitner@cadaxo.com)

Twitter: [@foessleitnerj](https://twitter.com/foessleitnerj)

## Beratungsschwerpunkte

- Konzeption & Management von Entwicklungsprojekten
- Qualitymanagement & Performanceoptimierung
- Clean Code
- Fiori, Fiori Elements & ABAP RESTful
- S/4HANA Custom Development
- SAP Champion 

# Domi Bigl

Senior Consultant at Cadaxo

eMail: [dominik.bigl@cadaxo.com](mailto:dominik.bigl@cadaxo.com)

Twitter: [@DomiBiglSAP](https://twitter.com/DomiBiglSAP)

## Beratungsschwerpunkte

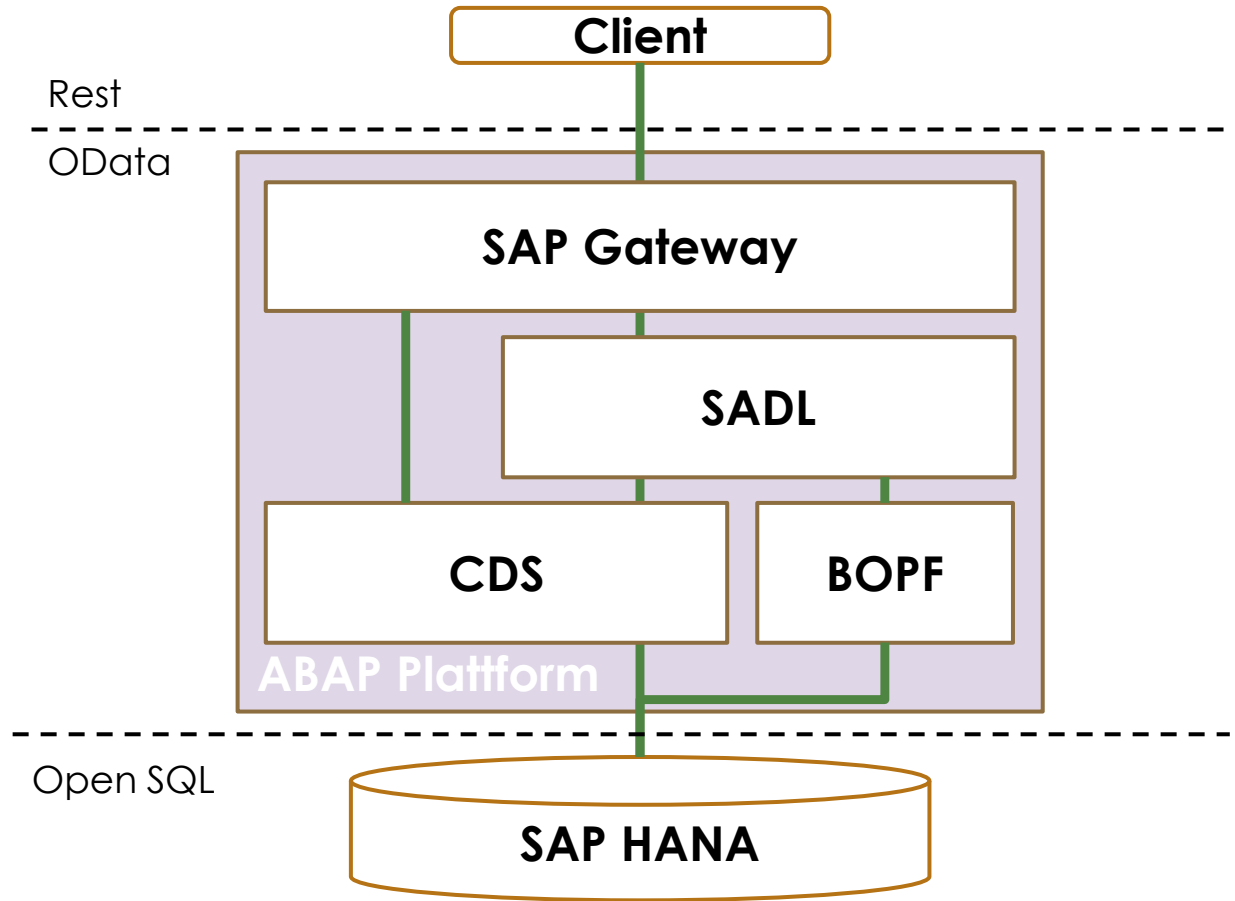
- Konzeption & Management von Entwicklungsprojekten
- Qualitymanagement & Performanceoptimierung
- ABAP Units
- SAP Fiori, SAP UI5



# CDS Views

# ● ABAP CDS Views

- Next Generation zum Definieren von Views
- CDS beinhaltet
  - DDL (Data Definition Language)
  - DCL (Data Control Language)
- 100% in ABAP integriert
  - ABAP Dictionary
  - Transportmanagement
- Pflege ausschließlich über ABAP in Eclipse
- Zugriff via Open SQL



# Traditionelle Datebankviews in SAP

Keine Outer  
Joins

Keine  
komplexen  
Joins

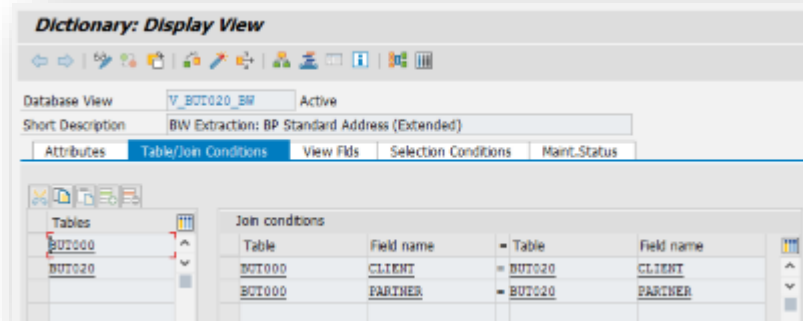
Keine  
Kommentare

Kein UNION

Kein View  
-> View

Keine impliziten  
Berechtigungen

...





## DDL

### Data Definition Language

- Definition von CDS Views

## DCL

### Data Control Language

- Definition von Zugriffskontrollen für CDS Views
- PFCG Rollen integrierbar

## ● ABAP CDS vs. HANA CDS

- Gleiche Syntax (DDL)
- Wer ABAP CDS kennt, findet sich auch in HANA CDS zurecht

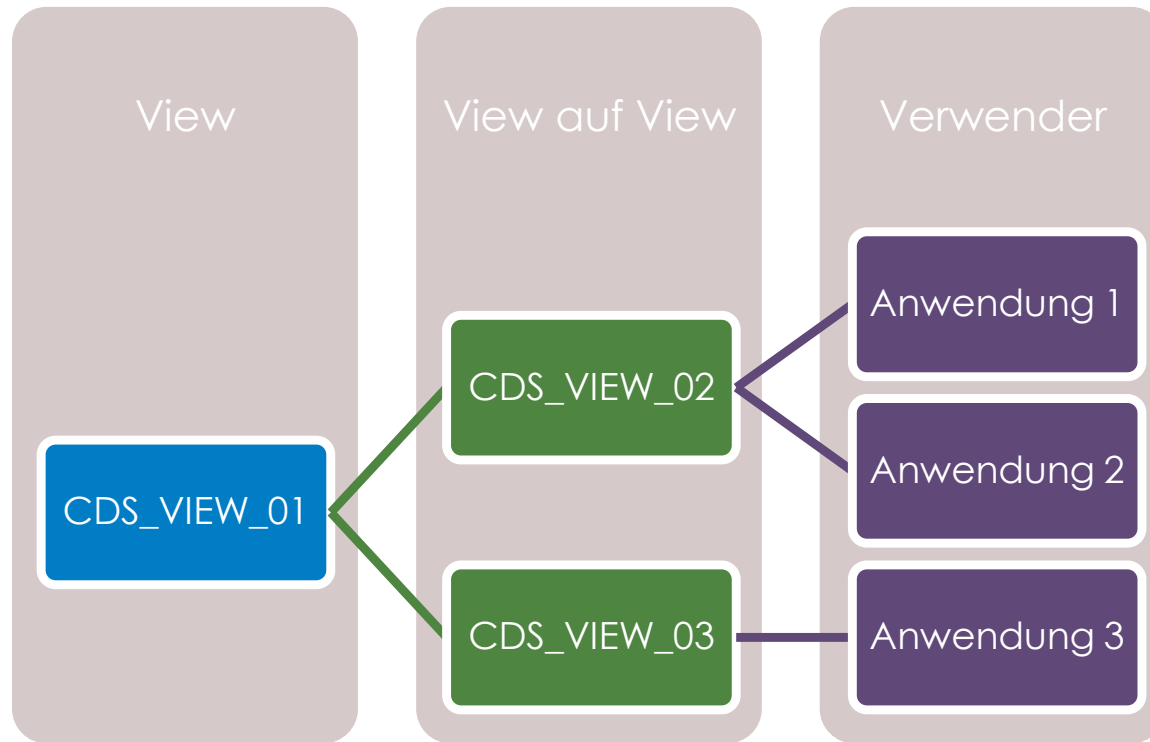
### ABAP CDS View

- DB Unabhängig -> Offen
- Tabellenbuffer
- ...



### HANA CDS View

- Nur für HANA
- Mehr Funktionen/Möglichkeiten



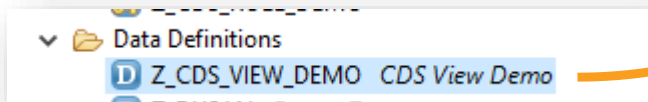
CDS View

Annotations

```
1 @AbapCatalog.sqlViewName: 'z_cds_demo_db'
2 @AbapCatalog.compiler.compareFilter: true
3 @AccessControl.authorizationCheck: #CHECK // #NOT_ALLOWED
4 @EndUserText.label: 'CDS View Demo'
5 define view Z_CDS_VIEW_DEMO as select from but000 {
6   key partner,
7   type,
8   mc_name1,
9   mc_name2,
10  augrp,
11  crusr
12 }
```

Viewfelder,  
Funktionen, ...

Tabelle der zu  
lesenden Daten



CDS Viewname

# Live Demo

## CDS View Entity

# CDS Entities

### CDS DDIC-Based View

SQL Zugriff auf Data Sources, technisch als DB View umgesetzt

DEFINE VIEW ...

ABAP 7.40 SP5

### CDS Table Function

CDS Entity welche als AMDP Funktion implementiert ist

DEFINE TABLE FUNCTION ...

ABAP 7.50

### CDS Abstract Entities

Definiert eine abstrakte CDS Entity ohne Datasource

DEFINE ABSTRACT  
ENTITY ...

ABAP 7.53

### CDS Hierarchies

CDS Entity welche eine SQL Hierarchy auf eine Data Source realisiert

DEFINE HIERARCHY ...

ABAP 7.53

### CDS Custom Entities

Realisiert als ABAP Klasse

DEFINE CUSTOM ENTITY ...

ABAP 7.54

### CDS Projection Views

Zugriff auf eine Teilmenge eines CDS Views

DEFINE VIEW ENTITY ... AS  
PROJECTION ...

ABAP 7.54

### CDS View Entities

SQL Zugriff auf Data Sources

DEFINE VIEW ENTITY ...

ABAP 7.55

# Eingebaute Funktionen



### Numerische Funktionen

ABS, CEIL,  
DIV, DIVISION,  
FLOOR, MOD,  
ROUND

### Zeichenketten-funktionen

CONCAT,  
CONCAT\_WITH\_SPACE,  
INSTR, LEFT, LENGTH,  
LOWER, LPAD,  
LTRIM, REPLACE,  
RIGHT, RPAD, RTRIM,  
SUBSTRING, UPPER

### Bytekettenfunktionen

BINTOHEX,  
HEXTOBIN

### Coalesce-Funktion

COALESCE

### Konvertierungs-funktionen

FLTP\_TO\_DEC,  
UNIT\_CONVERSION,  
CURRENCY\_CONVERSION,  
DECIMAL\_SHIFT

### Datumsfunktionen Zeitfunktionen

DATS\_IS\_VALID,  
DATS\_DAYS\_BETWEEN,  
DATS\_ADD\_DAYS,  
DATS\_ADD\_MONTHS,  
DATN\_DAYS\_BETWEEN,  
DATN\_ADD\_DAYS,  
DATN\_ADD\_MONTHS

### Zeitfunktion

TIME\_IS\_VALID

### Zeitstempelfunktionen

TSTMP\_IS\_VALID,  
TSTMP\_CURRENT\_UTCTIMEST  
AMP,  
TSTMP\_SECONDS\_BETWEEN,  
TSTMP\_ADD\_SECONDS,  
UTCL\_CURRENT,  
UTCL\_ADD\_SECONDS,  
UTCL\_SECONDS\_BETWEEN

### Zeitzonefunktionen

ABAP\_SYSTEM\_TIMEZONE,  
ABAP\_USER\_TIMEZONE

### Datums/Zeit- konvertierungen

TSTMP\_TO\_DATS,  
TSTMP\_TO\_TIMS,  
TSTMP\_TO\_DST,  
DATS\_TIMS\_TO\_TSTMP,  
TSTMP\_TO\_UTCL,  
TSTMP\_FROM\_UTCL,  
DATS\_TO\_DATN,  
DATS\_FROM\_DATN,  
TIMS\_TO\_TIMN,  
TIMS\_FROM\_TIMN

XXX - 7.40 SP8  
XXX - 7.50  
XXX - 7.51  
XXX - 7.54

# Live Demo

## Eingebaute Funktionen

# Annotations

# Was sind Annotations?

```
1 class PersonInfo {
2     int age = 25;
3     String name = "Joe";
4     public void show() {
5         System.out.println("Employee Name: " + name);
6         System.out.println("Employee Age: " + age);
7     }
8 }
9
10 public class AnnotationsExample extends PersonInfo {
11     int id = 12;
12     @Override
13     public void show() {
14         System.out.println("Employee id: " + id);
15     }
16
17     public static void main(String[] args) {
18         AnnotationsExample obj = new AnnotationsExample();
19         obj.show();
20     }
21 }
```

← Parent Class

Telling the compiler that subclass is overriding the show method of superclass

← Child Class Extending Parent Class

← Calling the show method using object of child class

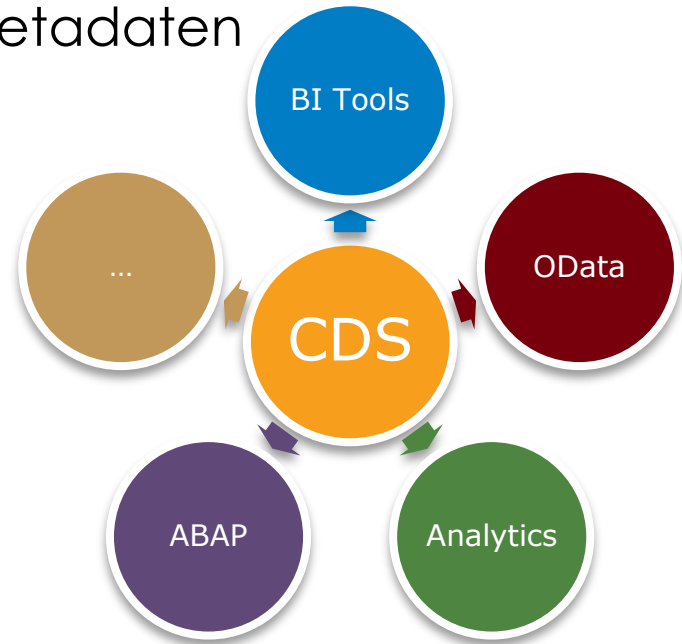
annotationsexample.AnnotationsExample >

Output - AnnotationsExample (run) x

run:  
Employee id: 12  
BUILD SUCCESSFUL (total time: 0 seconds)

## ● CDS View Annotations

- Ermöglicht das Anreichern von Metadaten direkt im View
  - Semantische Eigenschaften
  - Fiori Elements UIs
  - Analytische Annotations
  - ...



## ● CDS View - ABAP Annotations

- AbapCatalog Annotations
- AccessControl Annotations
- ClientHandling Annotations
- EndUserText Annotations
- Environment Annotations
- MappingRole Annotations
- Metadata Annotations
- Semantics Annotations

# ABAP Annotations

Annotation	Meaning	Scope	Type	Enums	Default	Text	MDE	API
AbapCatalog.buffering.numberOfKeyFields	Table buffering, number of key fields when buffering generic areas of DDIC based views	<a href="#">DDIC based View</a>	Integer	-	000	-	-	RELEASED_FOR_SAP_CLOUD_PLATFORM
AbapCatalog.buffering.type	Table buffering, defines the buffering type of DDIC based views	<a href="#">DDIC based View</a>	String(10)	#FULL #GENERIC #NONE #SINGLE	#NONE	-	-	RELEASED_FOR_SAP_CLOUD_PLATFORM
AbapCatalog.dbHints[ ].dbSystem	Obsolete: Replaced by Consumption.dbHintsCalculatedBy	<a href="#">DDIC based View</a>	String(3)	#ADA #ALL #ASE #DB2 #DB4 #DB6 #HDB #INF #MSS #ORA	-	-	-	RELEASED_FOR_SAP_CLOUD_PLATFORM
AbapCatalog.dbHints[ ].hint	Obsolete: Replaced by Consumption.dbHints[ ]	<a href="#">DDIC based View</a>	String(1298)	-	-	-	-	RELEASED_FOR_SAP_CLOUD_PLATFORM
AbapCatalog.entityBuffer.definitionAllowed	Table buffering, enables and disables buffering of view entities	View	Boolean	-	true	-	-	RELEASED_FOR_SAP_CLOUD_PLATFORM
AbapCatalog.extensibility.allowNewDatasources	Defines whether a CDS view entity extension is allowed to use new data sources, particularly newly defined associations.	View	Boolean	-	true	-	-	RELEASED_FOR_SAP_CLOUD_PLATFORM
AbapCatalog.extensibility.dataSources[ ]	Defines an allowlist for stable alias names of data sources and locally defined associations to be used in extensions.	View	LocalDefinitionRef	-	-	-	-	RELEASED_FOR_SAP_CLOUD_PLATFORM
AbapCatalog.extensibility.elementSuffix	Defines an element suffix which must be used for fields and associations when extending a released API.	<a href="#">Entity</a>	String(3)	-	-	-	-	RELEASED_FOR_SAP_CLOUD_PLATFORM
AbapCatalog.extensibility.extensible	CDS entity, controls CDS entity extensions	<a href="#">Entity</a> <a href="#">Service Definition</a>	Boolean	-	true	-	-	RELEASED_FOR_SAP_CLOUD_PLATFORM
AbapCatalog.extensibility.quota.maximumBytes	Defines the maximum number of bytes that can be added to a released API via extensions.	<a href="#">Entity</a>	Integer	-	-	-	-	RELEASED_FOR_SAP_CLOUD_PLATFORM

# ● CDS View - Framework Annotations

- Aggregation Annotations
- AccessControl Annotations
- Consumption Annotations
- ObjectModel Annotations
- OData Annotations
- Search Annotations
- Semantics Annotations
- UI Annotations



# Analytic Annotations

```
@Analytics.query : true
define view financial as select from sales

{
  @AnalyticsDetails.query.axis : #ROWS
  product,
  @AnalyticsDetails.query.axis : #COLUMNS
  @AnalyticsDetails.query.formula : 'revenue - cost'

  1 as absolute_margin,

  @AnalyticsDetails.query.formula : 'NDIV0($projection.absolute_margin / revenue ) * 100'

  1 as relative_margin,

  @AnalyticsDetails.query.formula : 'CASE WHEN $projection.relative_margin > 20 THEN revenue ELSE 0 END'
  1 as revenue_for_margin_gt_20
}
```

# Semantic Annotations

```
DEFINE VIEW SalesOrderItem as select from ...  
{  
    ...  
  
    @Semantics.currencyCode  
    currency_code as CurrencyCode,  
  
    @Semantics.amount.currencyCode: 'CurrencyCode'  
    gross_amount as GrossAmount,  
  
    @Semantics.unitOfMeasure  
    unit_of_measure as UnitOfMeasure,  
  
    @Semantics.quantity.unitOfMeasure: 'UnitOfMeasure'  
    quantity as Quantity,  
  
    ...  
}
```

# ObjectModel Annotations

```
define view I_Material
  association [0..*] to I_MaterialText as _Text ... {
    @ObjectModel.text.association: '_Text'
    key Material,
    _Text, ...
  }
```

```
define view I_MaterialText ... {
  key Material,
  @Semantics.language: true
  key Language,
  @Semantics.text: true
  MaterialName,
  @Semantics.text: true
  MaterialDescription, ...
}
```

# Session Variables

# Session Variables

vname	Value when accessed using ABAP SQL
user	Current <u>user name</u> , nominal value of the ABAP system field <u>sy-uname</u> . The returned value has the data type CHAR.
client	Current <u>client</u> . The default value is the nominal value of the ABAP system field <u>sy-mandt</u> . In reads with an ABAP SQL statement (with the addition <u>USING CLIENT</u> ) and in calls of an <u>AMDP method</u> from ABAP (in whose declaration the addition <u>AMDP OPTIONS CDS SESSION CLIENT</u> is specified, the value specified here. The returned value has the data type CLNT, length 3.
system_language	<u>Text environment language</u> of the current <u>internal session</u> , nominal value of the ABAP system field <u>sy-langu</u> . The returned value has the data type LANG, length 1.
system_date	Current <u>system date</u> of the AS ABAP, nominal value of the ABAP system field <u>sy-datum</u> . The returned value has the data type DATS, length 8.
user_timezone	<u>User time zone</u> , nominal value of the ABAP system field <u>sy-zonlo</u> . The returned value has the data type CHAR, length 6.
user_date	Current <u>user date</u> , nominal value of the ABAP system field <u>sy-datlo</u> . The returned value is of data type DATS, length 8.

JDBC doesn't like CDS!

# Literale und Casts

# Literale

```
1 @AccessControl.authorizationCheck: #CHECK
2 @AbapCatalog.sqlViewName: 'YMDB_ROLECLASSIC'
3 @EndUserText.label: 'Data Definition Role'
4 @Metadata.allowExtensions: true
5 define view YMDB_I_Role_Classic
6   as select from ymdb_role as Role
7     association [0..1] to YMDB_TYPE_OF_ACTING_VH as _TypeofActing on $projection.TypeofActing = _TypeofActing.value_low
8 {
9
10   key role_id                                as RoleId,
11     actor_id                                as ActorId,
12     movie_id                               as MovieId,
13     role_name                              as RoleName,
14     @Consumption.valueHelpDefinition: [{ entity:{name: 'YMDB_IMPORTANCY_VH', element: 'value_low'} }]
15     importancy                              as Importancy,
16     @Consumption.valueHelpDefinition: [{ entity:{name: 'YMDB_TYPE_OF_ACTING_VH', element: 'value_low'} }]
17     type_of_acting                          as TypeofActing
18 }
19 where type_of_acting = #YMDB_TYPE_OF_ACTING.'SCREEN'
20 |
```

# Casts

## Syntax

```
... CAST( operand AS dtype [PRESERVING TYPE]) ...
```

1. Berechnungen und Typanpassungen
2. Expliziten Typ übergeben
  - a. CASE-Anweisung
  - b. Entfernen von Conversion-Exits



# ADT für CDS

# Live Demo

## ADT für CDS

# Associations

```

SELECT A~PARTNER,
       B~ADDRNUMBER
FROM BUT000 AS A
INNER JOIN BUT020 AS B
ON B~PATNER = A~PARTNER
ORDER BY A~PARTNER,
        B~ADDRNUMBER
INTO TABLE @DATA(LT_RESULTS) .

```

Open SQL Join

```

@AbapCatalog.sqlViewName:
'Z_DEMOJOIN_DB'
@AccessControl.authorizationCheck:
#NOT_REQUIRED
define view z_demo_cds_join
as select from but000
inner join but020 on
    but000.partner = but020.partner
{
    but000.partner as partner,
    but020.addrnumber as addrnumber
}

```

CDS Join

```

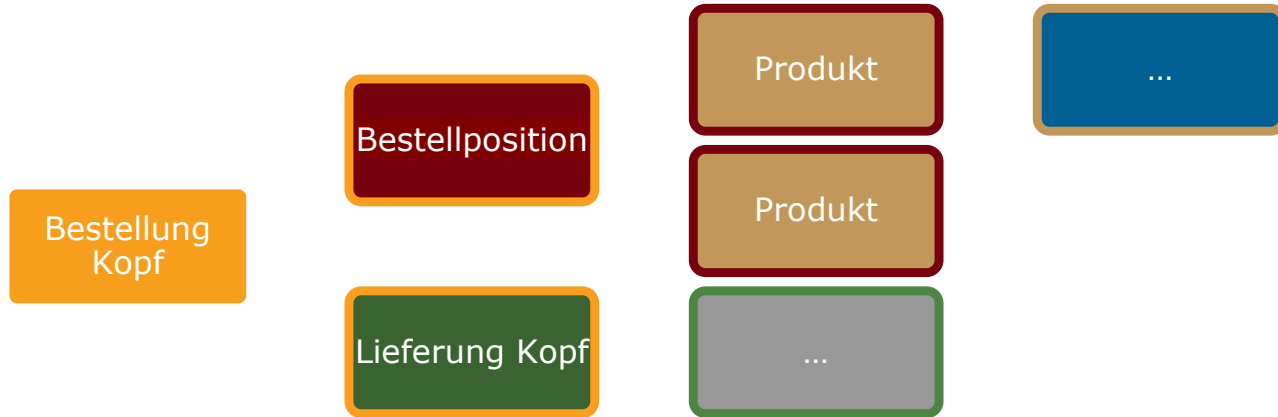
@AbapCatalog.sqlViewName:
'Z_DEMOJOIN_DB'
@AccessControl.authorizationCheck:
#NOT_REQUIRED
define view z_demo_cds_join
as select from but000
association [0..1] to but020 as
    _but020 on
    but000.partner = _but020-partner
{
    but000.partner,
    _but020[inner].addrnumber,
    _but020[1: ].addrnumber,
    _but020
}

```

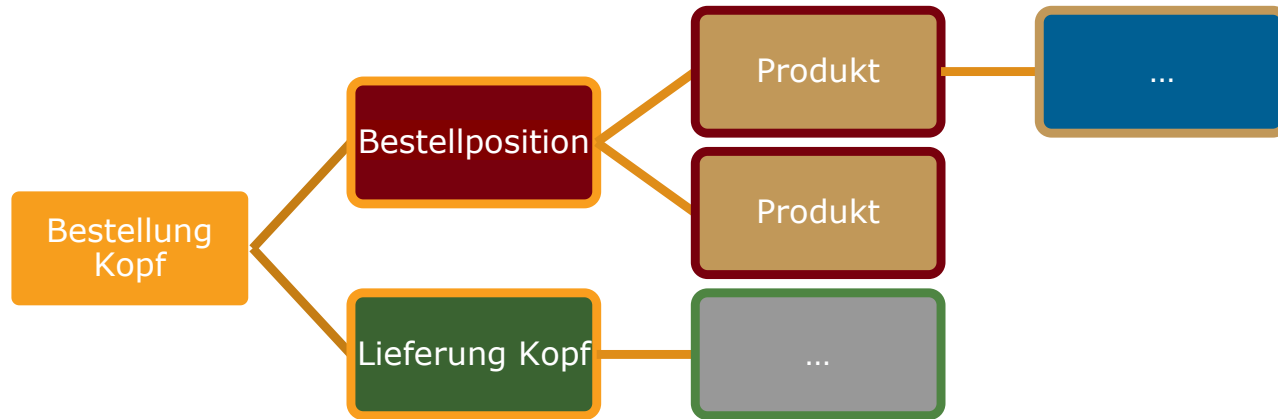
CDS Association

- **CDS Associations werden zur Laufzeit in JOINS übersetzt**

# ● Tables



## ● Datenmodell



— Associations

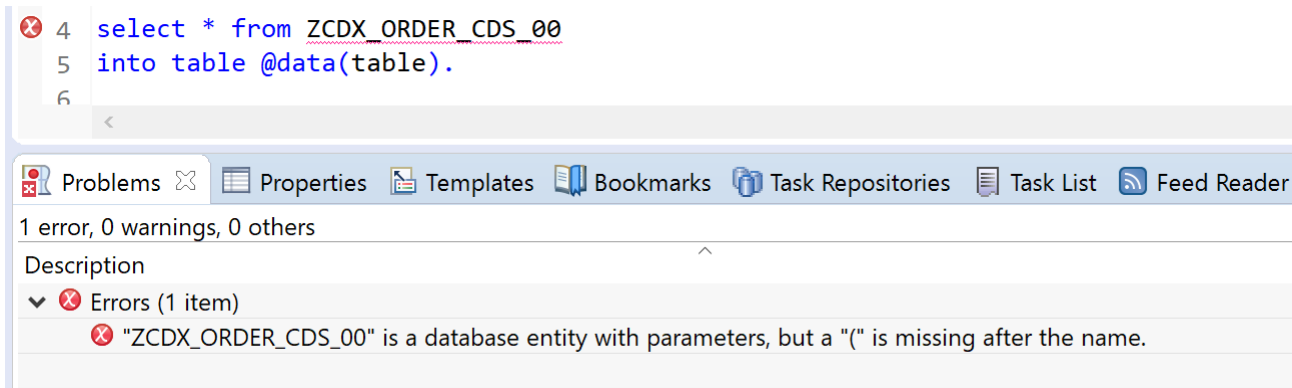
**Live Demo**

Accociations und Path Expressions

# Views mit Parametern



# ● Einschränkung/WHERE zu erzwingen



- Viewverhalten unabhängig von Daten steuern

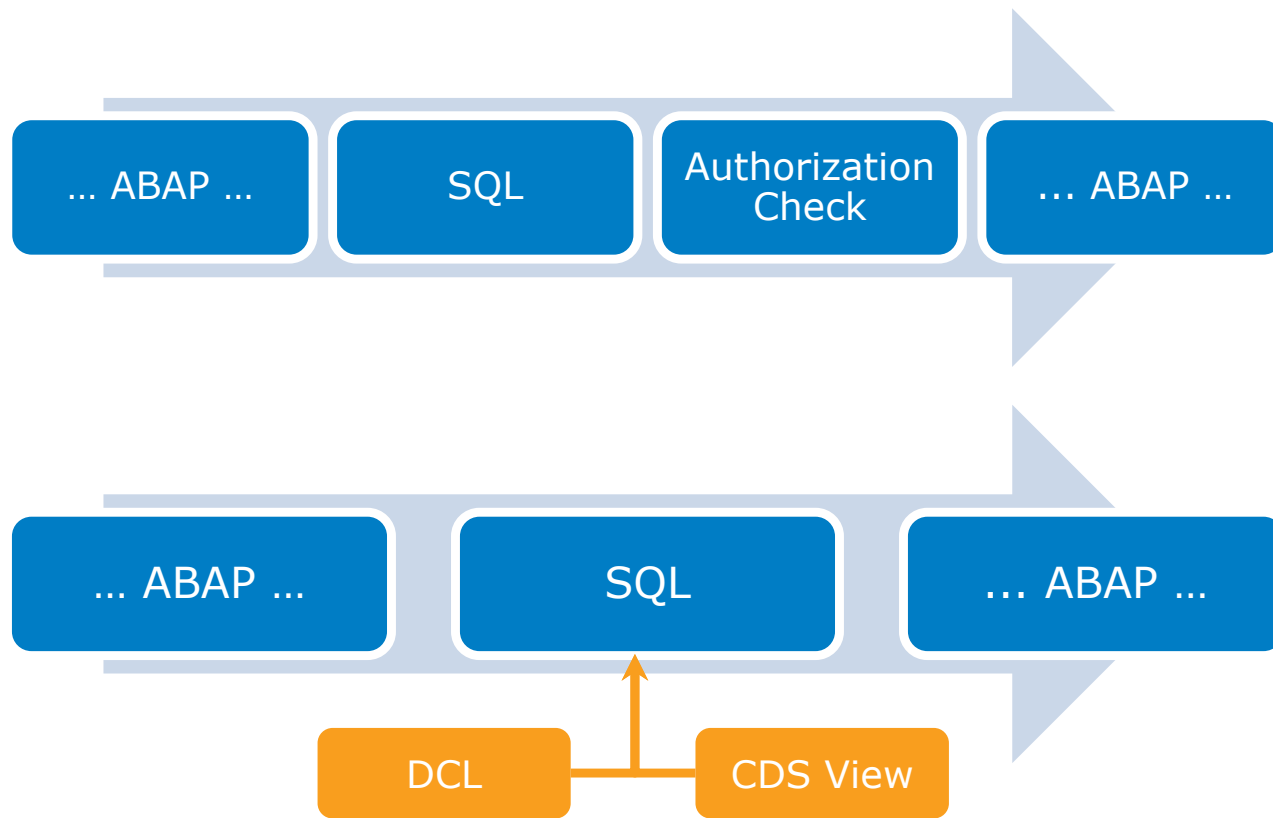
```
as select from but000
{
  case :p_name_case
    when 'FIRST' then name_first
    when 'LAST' then name_last
    else concat(name_first, concat(' ', name_last ) ) end as name
}
```

```
{
currency_conversion( amount => $parameters.p_Amount,
                    source_currency => cast( 'EUR' as abap.cuky( 5 ) ) ,
                    target_currency => cast( 'USD' as abap.cuky( 5 ) ) ,
                    exchange_rate_date => $parameters.p_ConvDate,
                    client => mandt ) as USD
}
```

# Live Demo

## View mit Parameter

# DCL Berechtigungen



# Zugriffsregeln

## Bedingte Zugriffsregel

- ... GRANT SELECT ON ...
- Schränkt den Zugriff mit Hilfe von Zugriffsbedingungen (WHERE) ein
  - Literalbedingungen (Country = 'DE')
  - Benutzerbedingungen (
  - PFCG-Bedingungen

## Vollzugriffsregel

- ... GRANT SELECT ON ...
- Ohne den Zusatz einer Zugriffsbedingung (WHERE) wird ein Zugriff bedingungslos freigegeben

## Geerbte Zugriffsregeln

- ... GRANT SELECT ON ... INHERIT ...
- Übernimmt die Zugriffsregeln einer bereits vorhandenen CDS Rolle

# Live Demo

## Daten Filtern mit DCL

```

1 @EndUserText.label: 'DCL Demo 01'
2 @MappingRole: true
3 define role Z_DEMO_CUSTOMER_01 {
4     grant
5         select
6             on
7                 /CADAXO/UI38D_Demo_Customer
8                 //where
9                     // Country = 'DE' // Demo 1
10                    // Uname = aspect user // Demo 2
11                    // ( Country ) = aspect pfcg_auth( ZCUSCOUNT, COUNTRY, ACTVT = '03' ) //Demo 3 - DE, AT
12                    // or Country = 'US' // Demo 4
13                    // Country is null // Demo 5
14                ;
15 }

```



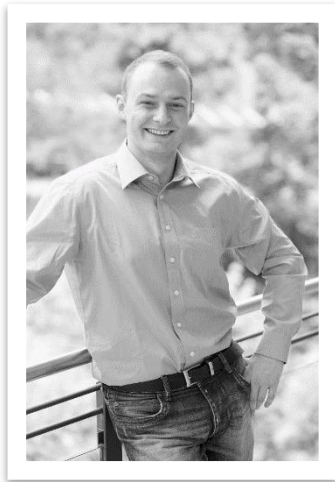


## ● ABAP CDS 1

- CDS Views Einführung (Föß)
- CDS Entities Überblick (Föß)
- Expressions und Funktionen (Föß)
- Access Control (Föß)
  
- Annotations (Sören)
- Session Variablen (Sören)
- Literale / Cast (Sören)
- CDS in ADT (Sören)
  
- Associations (Domi)
- CDS Views mit Parametern (Domi)
- Select Path Zugriff, Parameter (Domi)
  
- Metadata Extensions (Sören)

## ● ABAP CDS 2

- CDS View Table Function
- CDS Custom Entities
- CDS View Hierarchy
- Annotations als CDS Objekte (DEFINE ANNOTATION)
- Migration CDS DDIC Views -> CDS Entities
- CDS View Erweitern
- ...



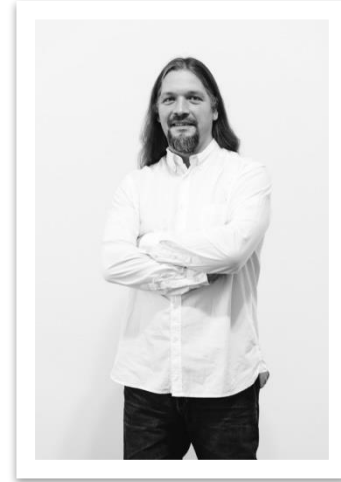
[soeren.schlegel@js-soft.com](mailto:soeren.schlegel@js-soft.com)

@SoSchlegel87



[johann.foessleitner@cadaxo.com](mailto:johann.foessleitner@cadaxo.com)

@foessleitnerj



[dominik.bigl@cadaxo.com](mailto:dominik.bigl@cadaxo.com)

@DomiBiglSAP



<http://www.cadaxo.com/blog/>