



ABAP 7.53/7.54 Releaseabhängige Änderungen

Webinar, 27.3.2020, 10:00 Uhr



Johann Föbleitner

Senior Consultant at Cadaxo

eMail: johann.foessleitner@cadaxo.com

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

Beratungsschwerpunkte

- Konzeption & Management von Entwicklungsprojekten
- Qualitymanagement & Performanceoptimierung
- Clean Code
- S/4HANA Custom Development
- SAP Champion 



Domini Bigl

Senior Consultant at Cadaxo

eMail: dominik.bigl@cadaxo.com

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

Beratungsschwerpunkte

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

Online gegen Corona

Wir erzählen über unsere Erfahrungen und Ideen wie man sich auch im Home Office erfolgreich im SAP Umfeld weiterbilden kann.

Donnerstag 2. April 10:00 Uhr



Damir Majer



Gregor Wolf



Christian Drumm



Sören Schlegel



Domi Bigl



Johann Fößleitner

<https://www.eventbrite.com/e/online-gegen-corona-im-sap-umfeld-tickets-101196916702>



SAP/ABAP & Microsoft – made easy

Webinar, 29.5.2020, 10:00 Uhr

<https://www.eventbrite.com/e/sapabap-microsoft-tickets-101302538620>

- ABAP Dictionary
- Interne Tabellen
- Zuweisungen
- ABAP SQL
- AMDP
- ABAP CDS
- ABAP RESTful Programming Model
- Exceptionhandling
- ABAP Units
- ABAP Development Tools



ABAP Dictionary

- **Neue eingebaute Datentypen**
 - Gleitpunktzahlen
 - DECFLOAT16, DECFLOAT34
 - Datums-, Zeit- und Zeitstempelfelder
 - DATN, TIMN, UTCLONG
 - Geodaten
 - GEOM_EWKB
 - Derzeit **nur bei SAP HANA Datenbanken** möglich

- **Neuer eingebauter ABAP-Typ**

- UTCLONG

- 0001-01-01T00:00:00,0000000 -
9999-12-31T23:59:59,9999999

- Zeitstempelfunktionen

- utclong_current, utclong_add, utclong_diff

- Stringformatierung mit TIMESTAMP

- **POOL & CLUSTER Tabellen**

- Diese Tabellentypen sind ab 7.53 restlos abgeschafft (Halleluja!)
- Alle vorhandenen Tabellen werden in transparente Tabellen umgesetzt

Interne Tabellen

- **Interne Tabellen**

- In WHERE Bedingungen für LOOP, DELETE, MODIFY und FOR kann nun auch **IS [NOT] INSTANCE OF** verwendet werden
 - IS [NOT] INITIAL, IS [NOT] BOUND ging schon bisher
- Strengere Syntaxprüfung bei **COLLECT**
 - (btw - wer verwendet eigentlich noch collect?)

Zuweisungen

- **Berechnungszuweisungen (juhuuu!)**

- +=, -=, *=, /=, &&=
- ADD, SUBTRACT, MULTIPLY, DIVIDE sind jetzt aber wirklich obsolet

```
DATA field TYPE p decimals 2.
```

```
field += 5.
```

```
field -= 1.
```

```
field *= 2.
```

```
field /= 4.
```

~~OPEN~~ ABAP SQL

• Neue Datums-/Zeitfunktionen

- Zeitfunktion
 - TIMS_IS_VALID
- Zeitstempelfunktionen
 - TSTMP_IS_VALID, TSTMP_CURRENT_UTCTIMESTAMP, TSTMP_SECONDS_BETWEEN, TSTMP_ADD_SECONDS
- Datums-/Zeitkonvertierungen
 - TSTMP_TO_DATS, TSTMP_TO_TIMS, TSTMP_TO_DST, DATS_TIMS_TO_TSTMP
- Zeitzonefunktionen
 - ABAP_SYSTEM_TIMEZONE, ABAP_USER_TIMEZONE

https://help.sap.com/doc/abapdocu_latest_index_htm/latest/en-US/abenopen_sql_builtin_functions.htm

• UUID Funktion

- Die neue UUID Funktion hat keinen Parameter sondern erzeugt für jeden gelesenen Datensatz eine neue UUID vom Typ RAW 16

```
SELECT column1, column2, uuid( ) AS uuid  
FROM dbtable  
INTO TABLE @DATA(result).
```

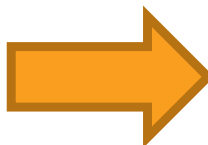
Demo

• Aggregatsfunktion **STRING_AGG**

- Mit **STRING_AGG** können zeichenartige Ergebnisse der Zeilen in eine Zeichenkette verkettet werden

```
select string_agg( city, ', ' order by city descending )
  from zfoe_test
 group by country
into table @data(result).
```

PK	CLIENT	PK	CITY	PK	COUNTRY
100			Wien		AT
100			Graz		AT
100			Salzburg		AT
100			Walldorf		DE
100			Weinheim		DE
100			New York		US
100			St. Augustine		US
100			Port Louis		ME



Ergebnis

Wien, Salzburg, Graz

Weinheim, Walldorf

St. Augustine, New York

Port Louis

Demo

- **Erweiterung der INTO Klausel: NEW**
 - Mit dem neuen Zusatz NEW können implizit anonyme Datenobjekte als Zielbereiche erzeugt werden

```
DATA dref TYPE REF TO data.  
  
SELECT FROM but000  
       FIELDS *  
       INTO TABLE NEW @dref.  
  
ASSIGN dref->* TO FIELD-SYMBOL(<fs>).
```

• Erweiterung der INTO Klausel: INDICATORS

- Mit dem neuen Zusatz INDICATORS können Indikatoren angegeben werden.
- Derzeit ist die Angabe eines Null-Indikators möglich.

```
select single from zfoe_test
  fields country,
         city,
         population
  into @data(result) indicators null structure null_ind.

out->write( |{ result-null_ind-country }, { result-null_ind-city }| ).
```

Demo

• Aggregatsfunktionen

- Aggregatsfunktionen können in SQL Ausdrücken verwendet werden

```
select from zfoe_test
fields country,
      sum( population ) as population,
      case when sum( population ) between 0 and 1000000 then 'unter 1 Mio'
            when sum( population ) between 1000000 and 5000000 then 'zwischen 1 und 5 Mio'
            else 'über 5 Mio'
      end as bereich
group by country
order by country
into table @data(result).
```

Demo

• GROUP BY / GROUPING SETS

- Mit den GROUPING SETS können Gruppierungsmengen unterhalb einer GROUP BY Klausel definiert werden

```
select from zfoe_test
  fields country,
         city,
         sum( population ) as population,
         grouping( country ) AS grouping_country,
         grouping( city ) AS grouping_city

group by grouping sets ( ( country ),
                        ( city ) )

into table @data(result).
```

COUNTRY	CITY	POPULATION	GROUPING_COUNTRY	GROUPING_CITY
AT		2650000.0	0	1
DE		60000.0	0	1
US		14850000.0	0	1
	Austin	350000.0	1	0
	Chicago	3000000.0	1	0
	Graz	500000.0	1	0
	Los Angeles	3000000.0	1	0
	New York	8500000.0	1	0
	Salzburg	150000.0	1	0
	Walldorf	15000.0	1	0
	Weinheim	45000.0	1	0
	Wien	2000000.0	1	0

Demo

ABAP SQL Windowing

- **ABAP SQL Windowing**
 - Ein Window definiert in ABAP SQL mit OVER ein Fenster als Teilmenge der Ergebnismenge

Country	City	Population	?
AT	Wien	2000000	
AT	Graz	500000	
AT	Salzburg	150000	
DE	Walldorf	15000	
DE	Weinheim	45000	
US	Austin	350000	
US	Chicago	3000000	
US	Los Angeles	3000000	
US	New York	8500000	

Window

Country	City	Population	Row Number
AT	Salzburg	150000	1
AT	Graz	500000	2
AT	Wien	2000000	3
DE	Walldorf	15000	1
DE	Weinheim	45000	2
US	Austin	350000	1
US	Chicago	3000000	2
US	Los Angeles	3000000	3
US	New York	8500000	4

100 %

```
select from zfoe_test
  fields country, city, population,
  row_number( ) over( partition by country order by population, city ascending ) as row_number
group by country, city, population
order by country, population, city ascending
into table @data(result).
```

Window

Country	City	Population	Rank
AT	Salzburg	150000	1
AT	Graz	500000	2
AT	Wien	2000000	3
DE	Walldorf	15000	1
DE	Weinheim	45000	2
US	Austin	350000	1
US	Chicago	3000000	2
US	Los Angeles	3000000	2
US	New York	8500000	4

100 %

```
select from zfoe_test
  fields country, city, population,
  rank( ) over( partition by country order by population ) as rank_by_country_asc
group by country, city, population
order by country, population, city ascending
into table @data(result).
```

Window

Country	City	Population	Lead Population
AT	Salzburg	150000	500000
AT	Graz	500000	2000000
AT	Wien	2000000	0
DE	Walldorf	15000	45000
DE	Weinheim	45000	0
US	Austin	350000	3000000
US	Chicago	3000000	3000000
US	Los Angeles	3000000	8500000
US	New York	8500000	0

100 %

```
select from zfoe_test
  fields country, city, population,
  lead( population, 1 ) over( partition by country order by population, city ascending ) as lead_population
  group by country, city, population
  order by country, population, city ascending
  into table @data(result).
```

Window

Country	City	Population	Population/Total
AT	Wien	2000000	75,5 %
AT	Graz	500000	18,9 %
AT	Salzburg	150000	5,7 %
DE	Walldorf	15000	25 %
DE	Weinheim	45000	75 %
US	Austin	350000	2,4 %
US	Chicago	3000000	20,2 %
US	Los Angeles	3000000	20,2 %
US	New York	8500000	57.2 %

100 %

```
select from zfoe_test
  fields country, city, population,
  division( 100 * population, sum( population ) over( partition by country ), 1 ) as perc
  group by country, city, population
  order by country, population ascending
  into table @data(result).
```

Demo

• Überblick „Window“ Funktionen

- Alle klassischen Aggregatsfunktionen (SUM, MIN, MAX, ...)
- **RANK** erzeugt eine Rangfolge
- **DENSE_RANK** wie RANK aber ohne Lücken
- **ROW_NUMBER** nummeriert jede Ergebniszeile
- **LEAD** greift auf eine nachfolgende Zeile des Windows zu
- **LAG** greift auf eine vorhergehende Zeile des Windows zu

ABAP SQL Hierarchien

- **Hierarchiedaten als Datenquelle**
 - Hierarchien und Hierarchienavigatoren können angegeben werden
 - CDS-Hierarchie
 - Hierarchiegenerator HIERARCHY
 - CTE-Hierarchie

- **Tabelle ZFOE_WORLD**

- WORLD

- EUROPA

- AUSTRIA

- VIENNA

- ISCHGL

- GERMANY

- BAVARIA

- BERLIN

- AMERICA

- UNITED STATES

- CUBA

- ...

CDS Hierarchie

- CDS View mit Association
- CDS Hierarchie verwendet als Source den CDS View

```
1 @EndUserText.label : 'World'
2 @AbapCatalog.enhancementCategory : #NOT_EXTENSIBLE
3 @AbapCatalog.tableCategory : #TRANSPARENT
4 @AbapCatalog.deliveryClass : #A
5 @AbapCatalog.dataMaintenance : #LIMITED
6 define table zfoe_world {
7   key client : abap.clnt not null;
8   key area : abap.char(30) not null;
9   parent_area : abap.char(30);
10  population : abap.int8;
11 }
```

```
1 @AbapCatalog.sqlViewName: 'ZFOE_WORLD_CDSDB'
2 @AbapCatalog.compiler.compareFilter: true
3 @AbapCatalog.preserveKey: true
4 @AccessControl.authorizationCheck: #CHECK
5 @EndUserText.label: 'foe world cds'
6 define view zfoe_world_cds as select from zfoe_world
7 association [1] to zfoe_world_cds as _to_parent
8 on $projection.parent_area = _to_parent.area {
9   key area,
10  parent_area,
11  population,
12  _to_parent
13 }
14
```

```
1 define hierarchy ZFOE_WORLD_HIERARCHY
2 with parameters start_area : abap.char(20)
3 as parent child hierarchy (
4   source zfoe_world_cds
5   child to parent association _to_parent
6   start where parent_area = $parameters.start_area
7   siblings order by area ascending
8 )
9 {
10  area,
11  parent_area,
12  population,
13  _to_parent
14 }
```

```
81 select * from
82   zfoe_world_hierarchy( start_area = 'EUROPA' )
83   into table @data(result_cds_hierarchy).
84
85   out->write( result_cds_hierarchy ).
86
87 ENDMETHOD.
88
89 ENDCLASS.
```

*Global Class | Class-relevant Local Types | Local Types | Test Classes | Macros

Properties | Templates | Bookmarks | Task Repositories | Task List | Transport

ABAP Console

Demo: HIERARCHIES

Table	PARENT_AREA	POPULATION
AREA		
AUSTRIA	EUROPA	8823054
VIENNA	AUSTRIA	1889000
GERMANY	EUROPA	82800000
SWITZERLAND	EUROPA	8401120

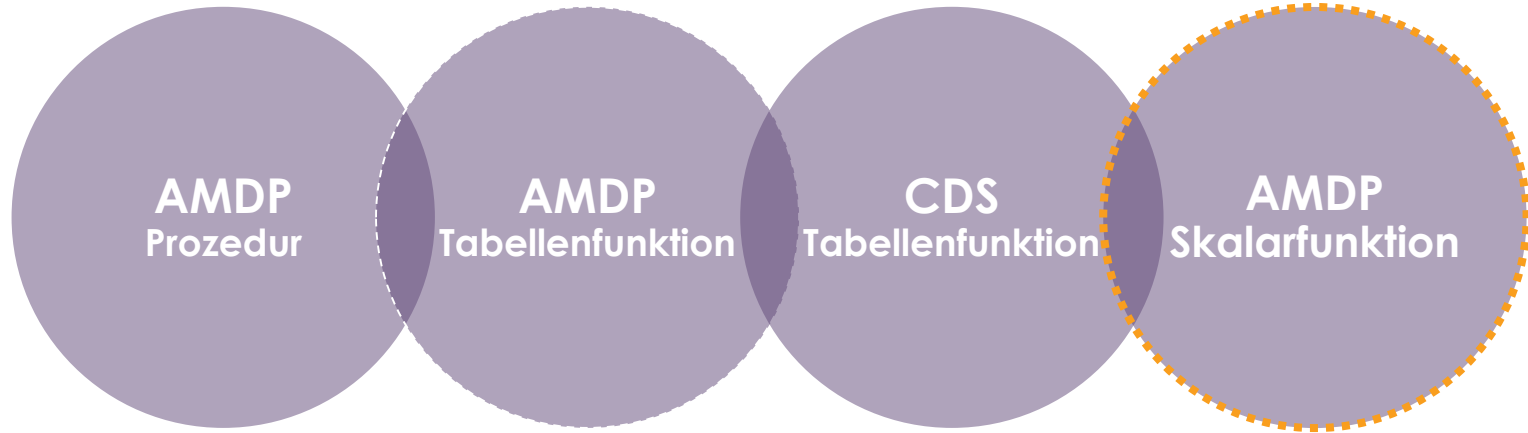
Demo

• Hierarchienavigatoren

- Knotennavigatoren
 - HIERARCHY_DESCENDANTS, HIERARCHY_ANCESTORS, HIERARCHY_SIBLINGS
- Aggregatnavigatoren
 - HIERARCHY_DESCENDANTS_AGGREGATE, HIERARCHY_ANCESTORS_AGGREGATE

```
SELECT FROM HIERARCHY_DESCENDANTS_AGGREGATE(  
    SOURCE zfoe_world_hierarchy( start_area = 'WORLD' )  
    MEASURES SUM( population ) AS total_population )  
FIELDS *  
INTO TABLE @DATA(result6).
```

AMDP



Shared Objects

• Invalidierung und Freigabe

- **Server-übergreifende** Invalidierung und Freigabe auch für nicht transaktionale Gebiete möglich
- Zusätzlicher Parameter **AFFECTED_SERVER** in der Gebietsklasse
 - INVALIDATE_INSTANCE
 - INVALIDATE_AREA
 - FREE_INSTANCE
 - FREE_AREA

```
free_instance
importing inst_name      type shm_inst_name default cl_shm_area=>default_instance
          terminate_changer type abap_bool default abap_true
          affect_server   type shm_affect_server default cl_shm_area=>affect_local_server
preferred parameter inst_name
returning value(rc)      type shm_rc
raising cx_shm_parameter_error
```

Documentation

Deletion of an Instance

Parameters

- ▢ **inst_name** Name of a Shared Object Instance of an Area
- ▢ **terminate_changer** Writing processes will be ended
- ▢ **affect_server** Servers on which the area is deleted or invalidated
- ▢ **rc** Return Value (Constants in CL_SHM_AREA)

ABAP CDS

- IS INITIAL
- Integer immer INT8
- DEFINE ABSTRACT ENTITY
 - „Struktur/Type“ Definition
- ANNOTATE ENTITY
 - Erweiterung ANNOTATE ENTITY
- DEFINE PFCG_MAPPING
 - Viewfelder => Auth Felder
 - Verwendet in DEFINE ROLE

```

1 @AbapCatalog.sqlViewName: 'ZC_DB_754_DEMO'
2 @AbapCatalog.compiler.compareFilter: true
3 @AbapCatalog.preserveKey: true
4 @AccessControl.authorizationCheck: #CHECK
5 @EndUserText.label: 'Demo CDS'
6 define view zi_db_750_demo
7   as select from zdb_754_demo
8 {
9   key partner,
10    name,
11    'Text'      as asCharacter,
12    '42'        as asNUMC,
13    42          as asINT,
14    0.22        as asFLTP,
15    :spras.'D'  as language
16 }
17 where
18   partner is initial
  
```

```

1 @AbapCatalog.sqlViewName: 'ZC_DB_754_DEMO'
2 @AbapCatalog.compiler.compareFilter: true
3 @AbapCatalog.preserveKey: true
4 @AccessControl.authorizationCheck: #CHECK
5 @EndUserText.label: 'Demo CDS'
6 define view zi_db_750_demo
7   as select from zdb_754_demo
8 {
9   key partner,
10    name,
11    'Text'      as asCharacter,
12    '42'        as asNUMC,
13    42          as asINT,
14    0.22        as asFLTP,
15    :spras.'D' as language
16 }
17 where
18   partner is initial
  
```

SELECT FROM zi_db_750_demo

FIELDS

INTO T

NDMETHOD.








zi_db_750_demo (View)

Demo CDS

Client Handling

- ☒ Client dependent
- ☐ Client session variable used

Column Data Element Data Type Description

	partner		char(10)	
	name		char(40)	
	asCharacter		char(4)	
	asNUMC		numc(2)	
	asINT		int1(3)	
	asFLTP		fltp(16,16)	
	language		lang(1)	



- @Consumption.dbHints
- @Consumption.dbHintsCalculatedBy
 - statt @AbapCatalog.dbHints
- @API.element.releaseState:#DEPRECATED
- @API.element.successor:
- DECFLOAT16 and DECFLOAT34 für CAST

Exceptionhandling

- NIE WIEDER An exception was raised - SY530
 - cl_message_helper=>get_latest_t100_exception
- RAISE **SHORTDUMP** TYPE cx_class | error_object
 - Mehr Infos in ST22
- IS_RESUMABLE

```
TRY.  
    "...  
  
    CATCH BEFORE UNWIND cx_abap_context_info_error INTO DATA(error_message).  
  
    DATA(protokoll) = cl_message_helper=>get_latest_t100_exception( error_message ).  
  
    IF error_message->is_resumable = abap_true.  
        RESUME.  
    ELSE.  
  
        RAISE SHORTDUMP error_message.  
  
    ENDIF.  
  
ENDTRY.
```

- MESSAGE error_object TYPE 'W'.
 - IF_T100_DYN_MSG

```
TRY.  
    "...  
  
    CATCH cx_abap_context_info_error INTO DATA(error_message).  
        MESSAGE error_message. " TYPE 'E'. "error_message->if_t100_dyn_msg~msgty  
ENDTRY.
```

ABAP Units

- Erweiterungen bei **ABAP Units!!!**

- ...



- Es sind absolut keine Komponenten von Testklassen mehr im Produktivcode zugreifbar!
 - 😞 Ok, was war noch übrig? (CONST, TYPES, ???)

ABAP DEVELOPEMENT TOOLS - ADT

• Delclare ... explicitly

```
SELECT FROM zdb_754_demo
  FIELDS partner,
    name,
    'Text' AS asCharacter,
    '42' AS asNUMC,
    42 AS asINT,
    'D' AS language
  WHERE partner IS INITIAL
  INTO TABLE @DATA(db_partner_demos).
```

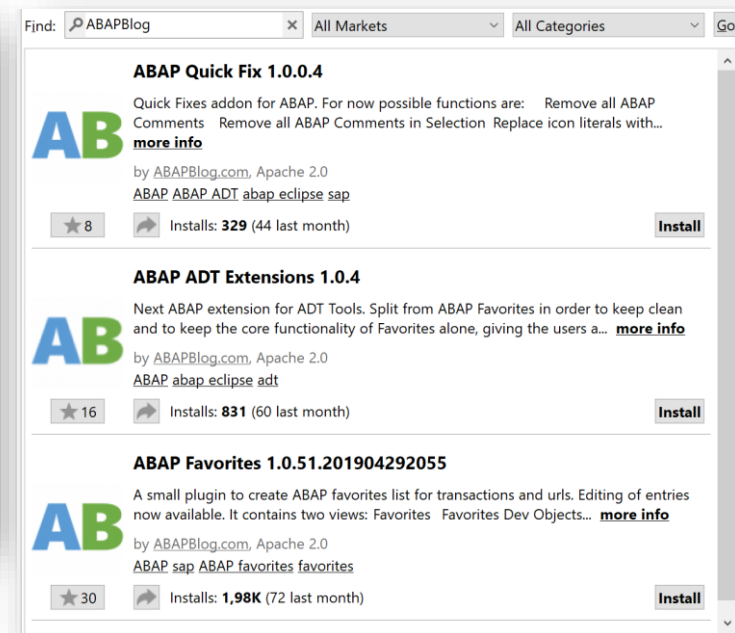
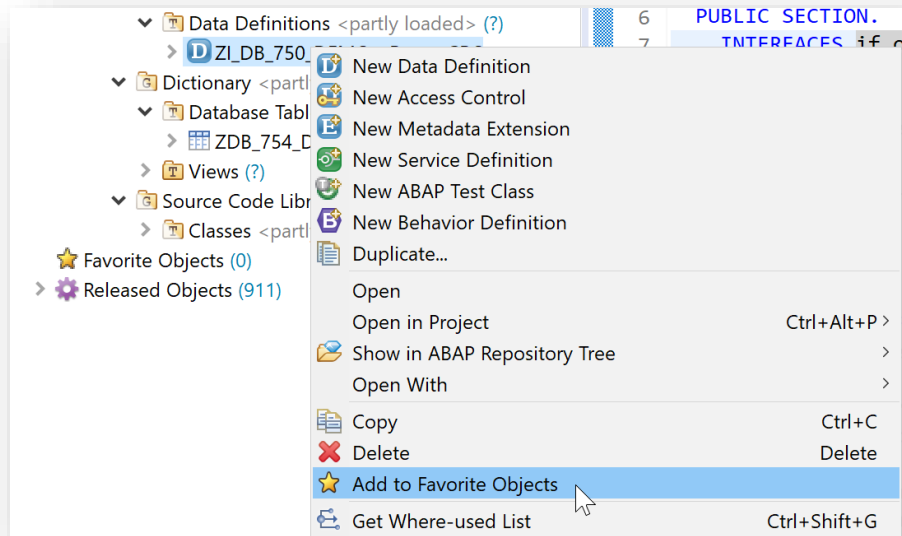
ENDMETHOD.

- Rename **db_partner_demos** (Ctrl+2, R)
- Declare local variable **db_partner_demos** explicitly
- Declare 6 local variables explicitly
- Convert **db_partner_demos** to attribute
- Convert **db_partner_demos** to returning parameter
- Convert **db_partner_demos** to changing parameter
- Convert **db_partner_demos** to exporting parameter

```
TYPES: BEGIN OF helper_type,
  partner      TYPE zdb_754_demo-partner,
  name         TYPE zdb_754_demo-name,
  ascharacter  TYPE c LENGTH 4,
  asnumc       TYPE c LENGTH 2,
  asint        TYPE i,
  language     TYPE c LENGTH 1,
END OF helper_type.

DATA: db_partner_demos TYPE STANDARD TABLE OF helper_type.
```

• Favorite Objects



ABAP RESTful Programming Model

Klassische ABAP Entwicklung

- Freestyle ABAP
Entwicklung

ABAP Plattform ≤ 7.50

ABAP Programmiermodell für SAP FIORI

- Gateway Builder
- OData
- BOPF
- Core Data Services

ABAP Plattform ≥ 7.51

ABAP RESTful Programmiermodell (ABAP RPM)

- Business Service
- Verhaltensdefinition
- Core Data Services

SAP Cloud Platform ABAP
Environment
1808 Cloud Edition
1909 On-Premise

- (Neue) Entwicklungen in S/4HANA und SAP Cloud Platform ABAP Environment
- CDS & ABAP
- Business Object
 - Datenmodellierung mit CDS
 - Verhaltensdefinition & Implementierung
 - CREATE/UPDATE/DEL...
 - Servicedefinition & Service Binding
 - Was gebe ich frei (zB Bestellkopf, Bestellposition, keine Produktdetails,...)
 - Mit welchem Namen (URL)
 - Managed / Unmanaged Zenario

```
managed implementation in class zbp_i_db_750_demo unique;

define behavior for zi_db_750_demo alias CDXDemo
persistent table ZDB_754_DEMO
lock master
{
    create;
    update;
    delete;
}
```

```
CLASS lcl_handler DEFINITION INHERITING FROM cl_abap_behavior_handler .
    PRIVATE SECTION.
        METHODS update_demo FOR MODIFY IMPORTING demo_data FOR CREATE CDXDemo.
ENDCLASS.
```


- Unmanaged Zenario
 - Bestehendes Coding, vorhandene APIs (Klassen, FM,...)
- Managed Zenario
 - SAP kümmert sich um CRUD
- MANAGED WITH – (Managed mit Erweiterung)
 - SAP, bitte kümmere dich um CRUD
 - ABER ich will was anpassen...

- Evolution of the ABAP Programming Model - Blog
 - <https://blogs.sap.com/2019/10/25/getting-started-with-the-abap-restful-programming-model/>

- ABAP RESTful Programming Model
 - <https://help.sap.com/viewer/fc4c71aa50014fd1b43721701471913d/201909.000/en-US/289477a81eec4d4e84c0302fb6835035.html>

- ABAP RESTful Programming Model
 - <https://blogs.sap.com/2019/05/23/sap-cloud-platform-abap-restful-programming-model-rap-for-beginners/>



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



<https://youtu.be/wxyZv1jPcHo>



https://youtu.be/nVtkLSI_oRY



<https://youtu.be/vZyojtKEtw4>

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

See you again!

Thank you for participating!



<https://twitter.com/foessleitnerj>



<https://www.linkedin.com/in/johann-fößleitner-a9851b2a>



https://www.xing.com/profile/johann_foessleitner



johann.foessleitner@cadaxo.com



<https://twitter.com/domibiglsap>



<https://www.linkedin.com/in/dominik-bigl-9b98b68b>



https://www.xing.com/profile/dominik_bigl



dominik.bigl@cadaxo.com