

Week 3 Unit 4: Core Data Services: View Definition



Core Data Services: View Definition

Outline

Content

- CDS View Definition Features
 - Projection List
 - Alias
- View-on-View Concept
- CDS View Extensions
- CDS Views with Input Parameters

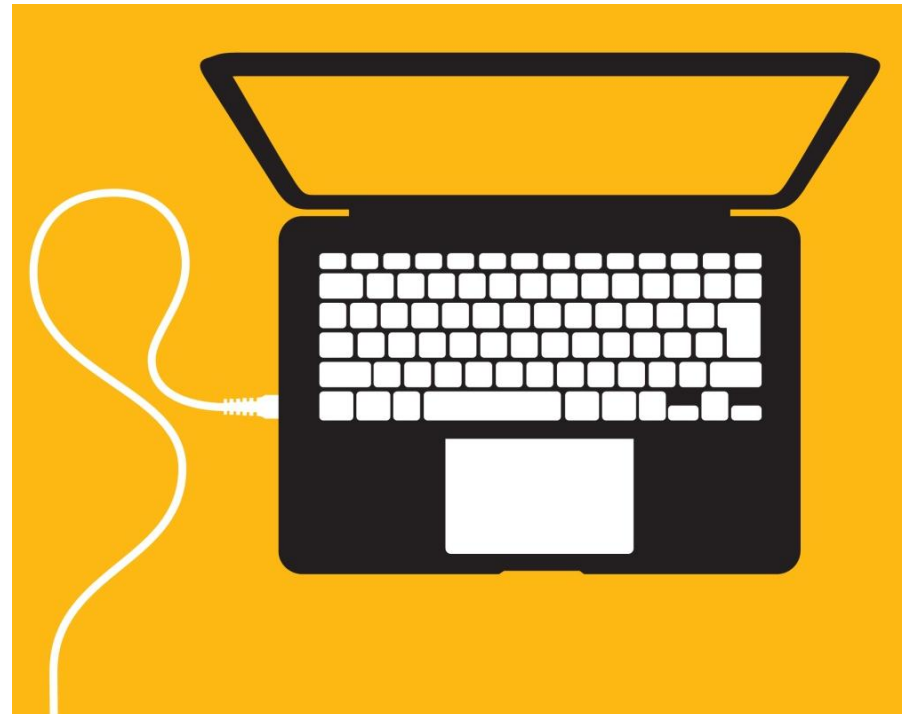


Core Data Services: View Definition

Demo

CDS View Definition Features

- Projection List:
 - Client Dependency
 - Semantic Information (Key)
 - Aliases
 - Aggregation
 - Literals
 - Arithmetic Expressions
 - Conditional Expressions
- GROUP BY & HAVING Clauses
- View-on-View Example
- CDS View Extensions
- CDS View with Input Parameters



Core Data Services: View Definition

Projection List (1)



Additional
Material

ABAP CDS View: Projection List

- Client-dependent view; no explicit client field necessary
- Semantic information (key field)
- Aliases
- Literal values:
 - C-sequence literals (Max length: 1333)
 - Signed integer literals (4-Byte)
- Aggregation functions:
 - **MIN, MAX, COUNT, AVG, SUM**
 - Alias required for function results

```
@AbapCatalog.sqlViewName: 'ZDDL5_CDS_10'  
define view zcdsv_aggregations as select  
  from snwd_so as so  
  inner join snwd_bpa as bpa  
    on so.buyer_guid = bpa.node_key  
{  
  key so_id as customer_id,  
    bpa.company_name,  
    'Literal' as string_literal,  
    42 as integer_literal,  
    so.currency_code,  
    sum( so.gross_amount ) as  
total_gross_amount  
}  
group by  
  bpa.bp_id,  
  bpa.company_name,  
  so.currency_code  
having sum( so.gross_amount ) > 1000
```



ABAP CDS View: Projection List

- Arithmetic expression:
 - Supported operators: +, -, * and unary –
 - Complex expressions and bracketing of sub-expressions possible
- Type casting:
 - Different operand types supported: Literal, column, path expression, build-in function, arithmetic expression
 - Various data types in ABAP namespace supported
 - Result length determined at activation time
 - No nesting of CAST expressions

Alias names required for resulting columns

```
@AbapCatalog.sqlViewName: 'ZDDL_S_CDS_11'  
define view zcdsv_arithmetics  
as select from snwd_so as so  
inner join snwd_bpa as bpa  
  on so.buyer_guid = bpa.node_key  
{  
  key bpa.bp_id as customer_id,  
  bpa.company_name,  
  so.currency_code,  
  ( so.gross_amount - so.net_amount )  
    as tax_amount,  
  0.85 * cast( so.gross_amount as  
  abap.fltp )  
    as reduced_gross_amount  
}
```



Conditional Expressions

- Available CASE constructs
 - Simple CASE
 - Searched CASE
- CASE constructs can be nested
“CASE-in-CASE”
- Coalesce expression
 - Syntax short form for a CASE expression with two arguments
 - Returns the first argument if the value is not NULL, otherwise the second argument is returned

```
@AbapCatalog.sqlViewName: 'ZDDL_S_CDS_12'
define view zcdsv_cond_exp
as select from snwd_so as so
left outer join snwd_so_inv_head as inv_head
  on so.node_key = inv_head.so_guid
{
  key so.so_id,
  so.currency_code,
  so.gross_amount,
  case delivery_status
    when ' ' then 'OPEN'
    when 'D' then 'DELIVERED'
    else delivery_status
  end as delivery_status_long,

  case
    when so.gross_amount > 1000
    then 'High Volume Sales Order'
    else ' '
  end as high_volumne_text,

  coalesce( inv_head.payment_status,
    'Not yet invoiced') as payment_status
}
```

Core Data Services: View Definition

View-on-View Concept



Additional
Material

View-on-View

- View can have other views as data basis
- No restriction on the number of layers

```
@AbapCatalog.sqlViewName: 'ZDDL_S_CDS_13A'  
define view zcdsv_base as select  
from snwd_so as so  
{  
  key so.so_id as order_id,  
  so.buyer_guid,  
  so.currency_code,  
  so.gross_amount  
}
```

```
@AbapCatalog.sqlViewName: 'ZDDL_S_CDS_13B'  
define view zcdsv_view_on_view as select  
from zcdsv_base  
inner join snwd_bpa as bpa  
  on bpa.node_key = zcdsv_base.buyer_guid  
{  
  key bpa.bp_id,  
  bpa.company_name,  
  zcdsv_base.currency_code,  
  zcdsv_base.gross_amount  
}
```

Core Data Services: View Definition

CDS View Extensions



Additional
Material

Extend existing/delivered CDS view with:

- Table column
- Arithmetic & CASE expressions
- Literals

Extension “technique”:

- Append to base view

Not allowed on views including:

- Grouping – for example, aggregation
- UNION (ALL) statements

```
@AbapCatalog.sqlViewName: 'ZDDL_CDS_13A'  
define view zcdsv_base as select  
from snwd_so as so  
{  
  key so.so_id as order_id,  
  so.buyer_guid,  
  so.currency_code,  
  so.gross_amount  
}
```

```
@AbapCatalog.sqlViewAppendName: 'ZDDL_CDS_13C'  
extend view zcdsv_base with  
zcdsv_customer_extension  
{  
  so.delivery_status,  
  so.billing_status,  
  so.created_at,  
  so.created_by  
}
```


Core Data Services: View Definition

CDS View with Input Parameters



Additional
Material

CDS Views with Input Parameters

- Comma-separated list of scalar input parameters and corresponding type
- Supported parameter types:
 - Predefined data type like `abap.char(char_len)`
 - Name of a data element
- Parameter can be used in
 - the projection list as element or in arithmetic expressions
 - expressions in WHERE or HAVING clauses
 - expression in ON conditions of JOIN statements
 - ...

Not supported on all databases

→ DBSYS-dependent feature

```
@AbapCatalog.sqlViewName: 'ZDDL_S_CDS_14A'
define view zcdsv_with_input_parameters
  with parameters customer_name : abap.char(80)
as select
  from snwd_so as so
  join snwd_bpa as bpa
    on bpa.node_key = so.buyer_guid
  {
    key so.so_id as order_id,
    $parameters.customer_name as
    param_customer_name,

    case
      when bpa.company_name =
      $parameters.customer_name
      then 'Found it!'
      else 'Not found'
    end as found_customer
  }
  where bpa.company_name = parameters.customer_name
```

Core Data Services: View Definition

CDS View with Input Parameters: Consumption (1)



Additional
Material

Consumption in a CDS view

- Provide (mandatory) input parameter(s)

```
@AbapCatalog.sqlViewName: 'ZDDL_S_CDS_14B'  
define view zcdsv_consume_param_view as select from  
zcdsv with input parameters( customer name : 'SAP' ) as vwp  
{  
  vwp.param_customer_name  
}
```

```
@AbapCatalog.sqlViewName: 'ZDDL_S_CDS_14A'  
define view zcdsv with input parameters  
  with parameters customer name : abap.char(80)  
as select  
from snwd_so as so  
join snwd_bpa as bpa  
  on bpa.node_key = so.buyer_guid  
{  
  key so.so_id as order_id,  
  $parameters.customer_name as param_customer_name,  
  
  case  
    when bpa.company_name = $parameters.customer_name  
    then 'Found it!'  
    else 'Not found'  
  end as found_customer  
}  
where bpa.company_name = $parameters.customer_name
```

Core Data Services: View Definition

CDS View with Input Parameters: Consumption (2)



Additional
Material

Consumption via Open SQL

- Check if the feature is supported
- Provide (mandatory) input parameter(s)
- Suppress syntax warning using the pragma
- Provide a “fallback” implementation / some error handling

```
REPORT zr_cds_01_consumption_vwp.

DATA lv_cust_name TYPE c LENGTH 80 VALUE 'SAP'.

"awesome application logic

DATA(lv_feature_supported) =
  cl_abap_dbfeatures=>use_features(
    EXPORTING
      requested_features =
        VALUE #( ( cl_abap_dbfeatures=>views_with_parameters ) )
  ).

IF lv_feature_supported = abap_true.
  SELECT *
  FROM zcdsv_with_input_parameters( customer_name = 'SAP' )
  INTO TABLE @DATA(lt_result)
  ##DB_FEATURE_MODE[VIEWS_WITH_PARAMETERS].
ELSE.
  "do some alternative coding here
ENDIF.

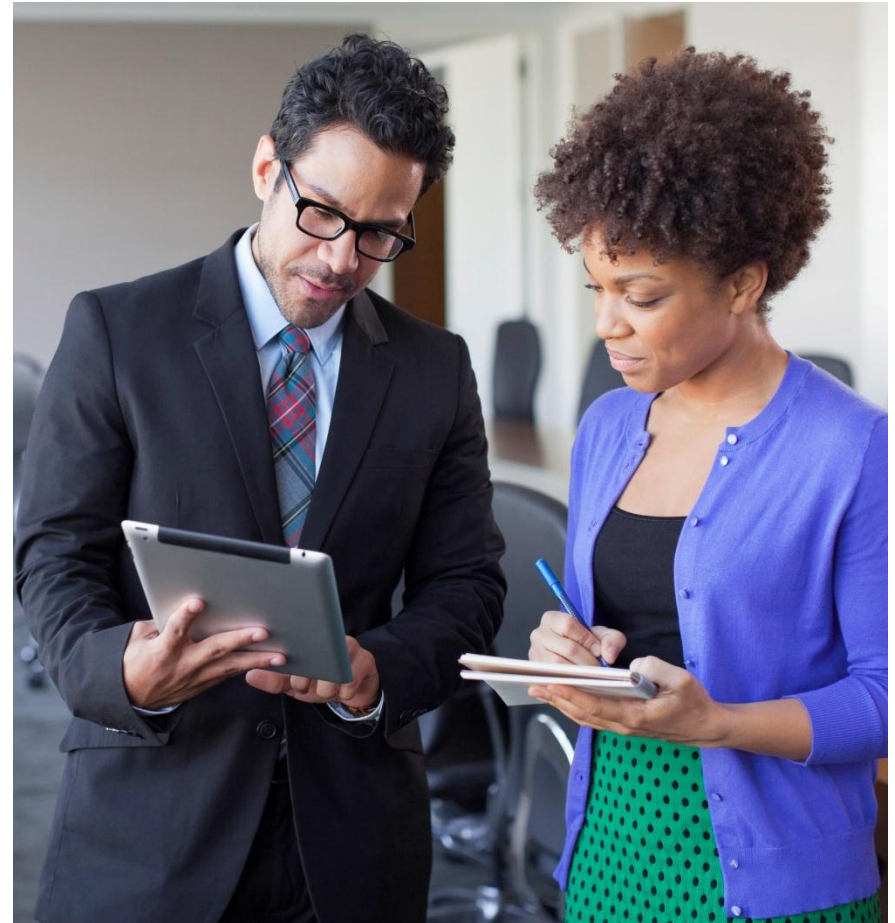
"even more awesome application logic
cl_demo_output=>display_data( lt_result ).
```

Core Data Services: View Definition

Conclusion

Key takeaways: CDS views...

- offer a rich set of features to follow the Code-to-Data paradigm
- can have other views as a data basis (View-on-View concept)
- can be extended
- can have scalar input parameters (DBSYS-dependent feature)



Core Data Services: View Definition

What's Next?

Week 3 Unit 5

Core Data Services: Associations





Thank you

Contact information:

open@sap.com

open**SAP**

© 2014 SAP AG or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG or an SAP affiliate company.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG (or an SAP affiliate company) in Germany and other countries. Please see <http://global12.sap.com/corporate-en/legal/copyright/index.epx> for additional trademark information and notices.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

National product specifications may vary.

These materials are provided by SAP AG or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP AG or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP AG or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP AG or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP AG's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP AG or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.