Week 3 Unit 1: **Recent Open SQL Enhancements – Part 1**

SAP

openSAP

Recent Open SQL Enhancements – Part 1
Outline

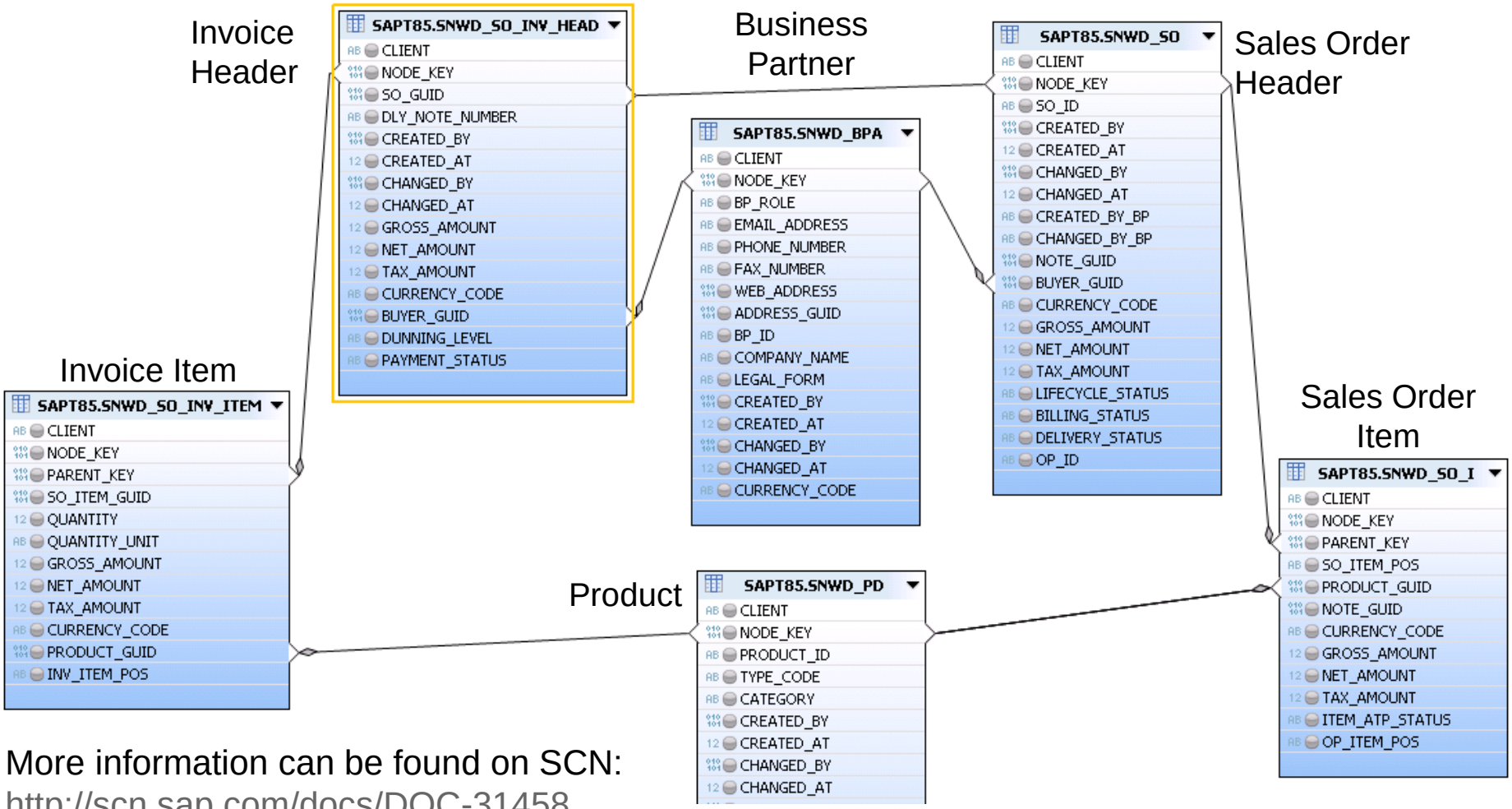
Content

- Enterprise Procurement Model
- Reminder: What is Open SQL?
- Open SQL evolution
- Open SQL enhancements
  - Syntax
  - SELECT list

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

Recent Open SQL Enhancements – Part 1

Enterprise Procurement Model (EPM)



More information can be found on SCN:
http://scn.sap.com/docs/DOC-31458

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

Recent Open SQL Enhancements – Part 1

Migration to SAP HANA

1. Detect

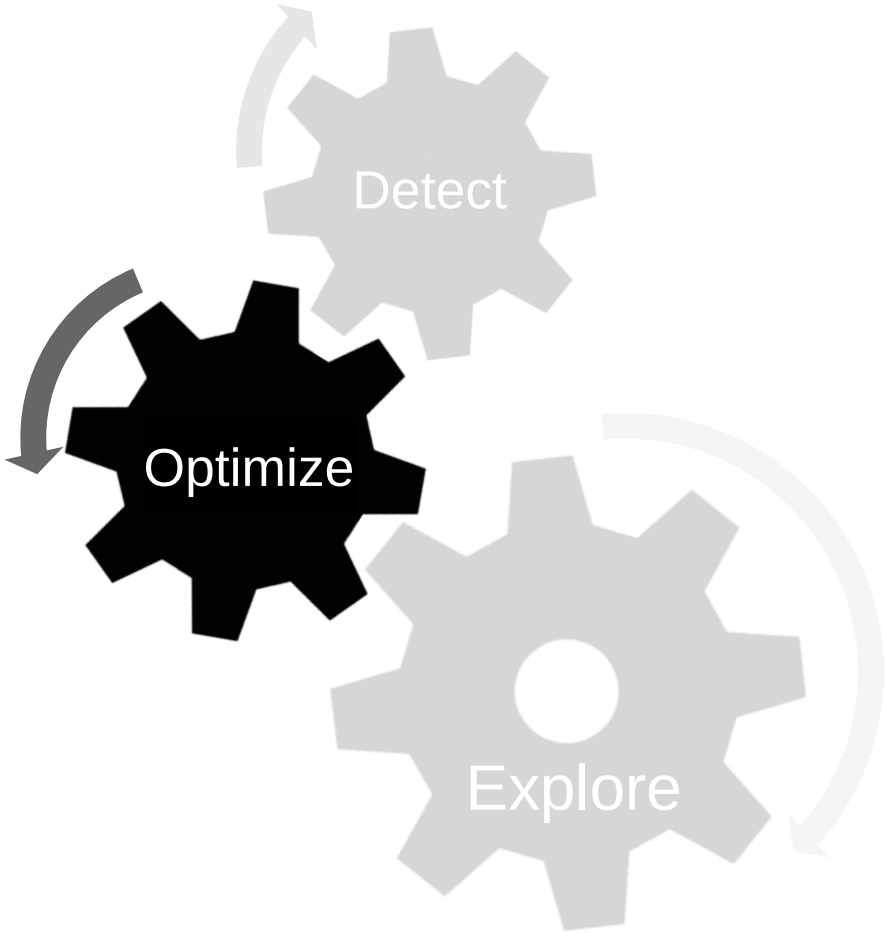
- Functional correctness
- Performance optimization potential

2. Optimize

- Database-oriented programming

3. Explore

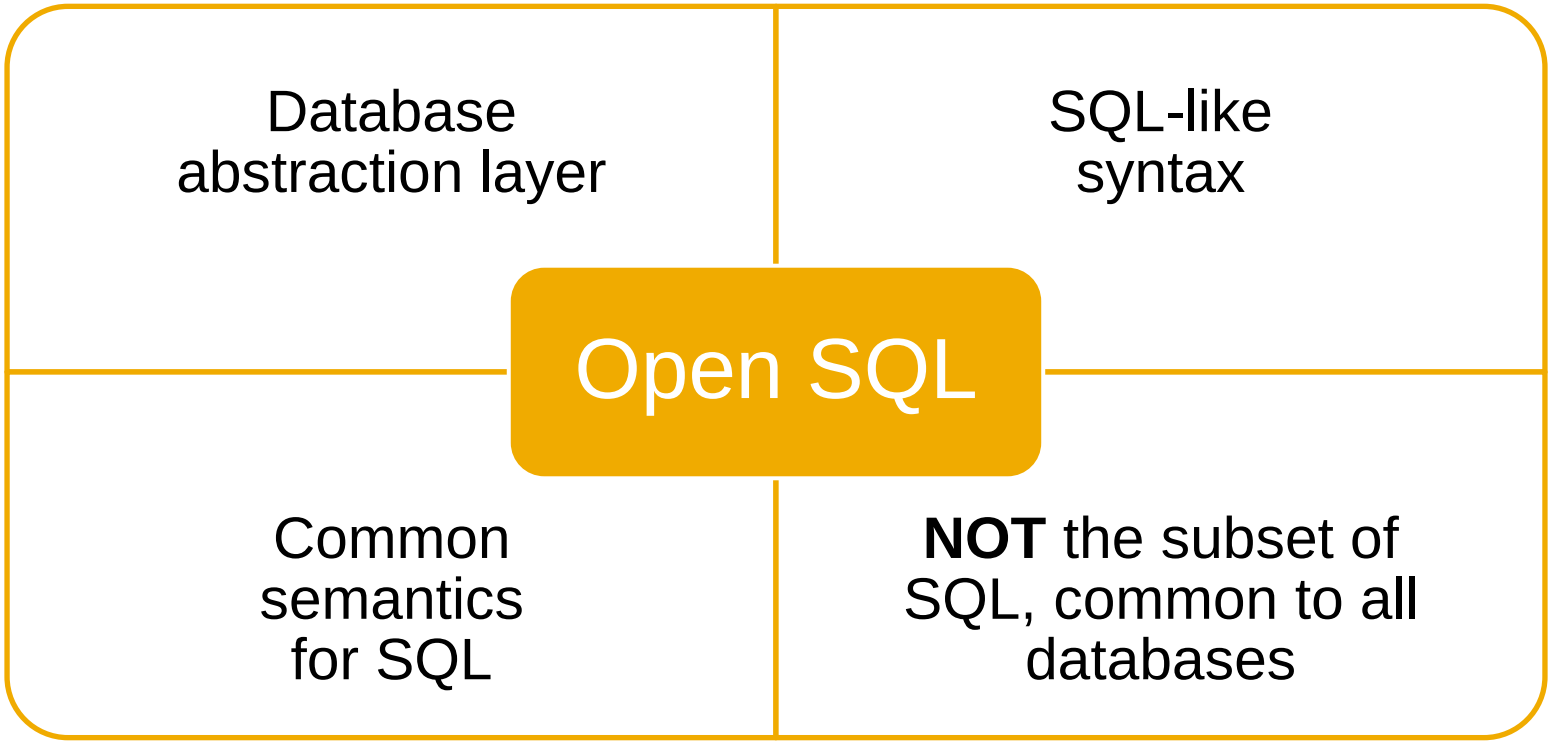
- Use SAP HANA-specific features
- Rethink & innovate



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

Recent Open SQL Enhancements – Part 1

Reminder: What Is Open SQL?



Open SQL is the only DB **abstraction layer** that defines a **common semantic** for all SAP-supported databases!

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

Recent Open SQL Enhancements – Part 1

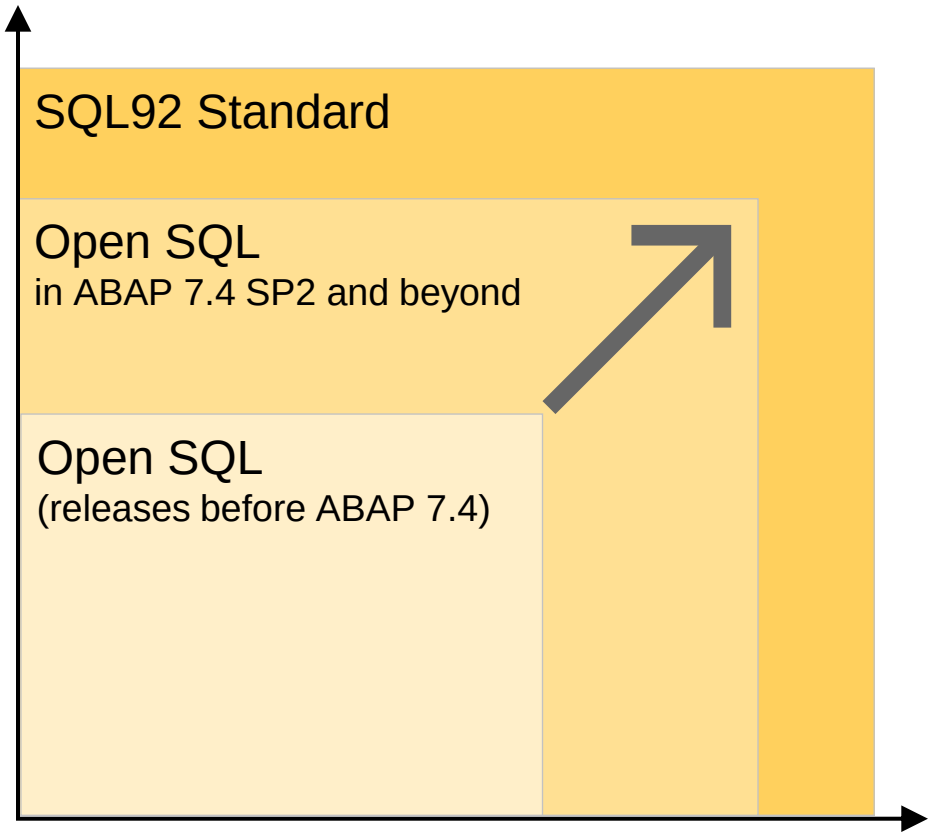
Open SQL in ABAP 7.4 SP2 and Beyond

Open SQL aspires to

- enable the application of the Code-to-Data paradigm
- provide (more) standard SQL features
- enable the consumption of SAP HANA-specific features

Introduction of new features

Reduction of existing limitations



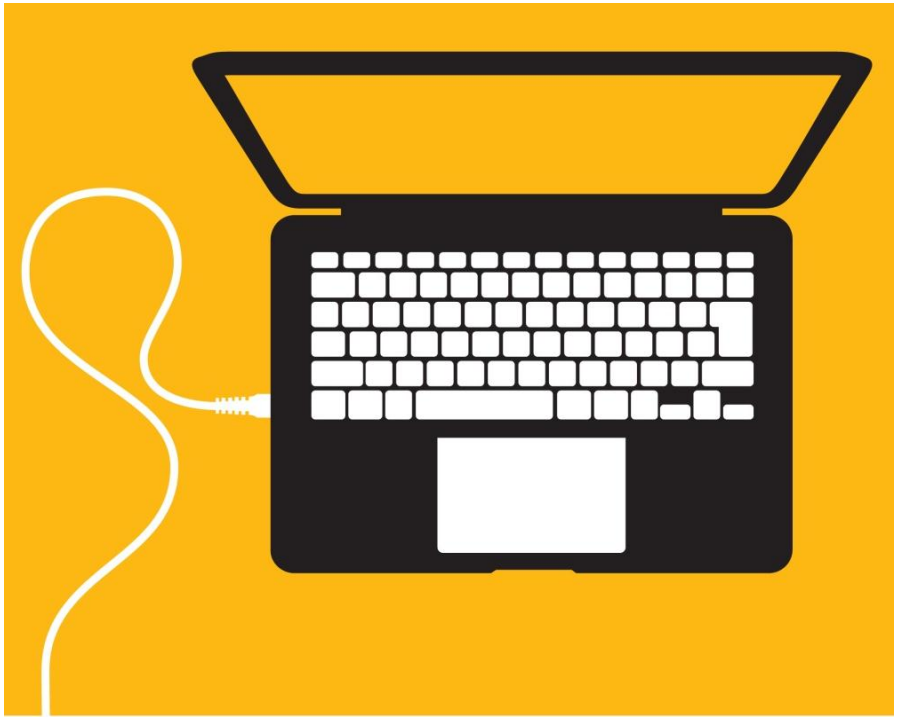
© 2014 SAP AG or an SAP affiliate company. All rights reserved

Recent Open SQL Enhancements – Part 1

Open SQL: Demo

Open SQL enhancements

- Syntax enhancements:
  - Escaping of host variables
  - Comma-separated select list
- SELECT list enhancements:
  - Aggregation functions
  - Literals
  - Arithmetical expressions



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

Recent Open SQL Enhancements – Part 1

What's New in Open SQL?

Additional Material

New Open SQL Syntax

- Escaping of host variables
- Comma separated element list

```
SELECT so_id,
       currency_code,
       gross_amount
  FROM snwd_so
  INTO TABLE @DATA(lt_result).
```

Target Type Inference

New SELECT List Features

- Aggregation functions
- Literal values (next slide)
- Arithmetic expressions (next slide)
- String expression (next slide)

```
SELECT bp_id,
       company_name,
       so~currency_code,
       SUM( so~gross_amount )
          AS total_gross_amount
  FROM snwd_so AS so
  INNER JOIN snwd_bpa AS bpa
          ON so~buyer_guid = bpa~node_key
  INTO TABLE @DATA(lt_result)
  GROUP BY bp_id, company_name,
           so~currency_code.
```

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

Recent Open SQL Enhancements – Part 1

Literal Values & Generic Existence Check

Additional Material

Literal Values

- Can now be used in the SELECT list
- Allow for a generic implementation of an existence check

```
SELECT so~so_id,
       'X' AS literal_x,
       42 AS literal_42
   FROM snwd_so AS so
   INTO TABLE @DATA(lt_result).
```

```
DATA lv_exists TYPE abap_bool
               VALUE abap_false.

SELECT SINGLE @abap_true
  FROM snwd_so
  INTO @lv_exists.

IF lv_exists = abap_true.
  "do some awesome application logic
ELSE.
  "no sales order exists
ENDIF.
```

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

Recent Open SQL Enhancements – Part 1
Expressions

Additional Material

Arithmetic Expressions

- **+, -, *, DIV, MOD, ABS, FLOOR, CEIL**
- **Remember:** Open SQL defines a semantic for these expressions common to all supported databases
- Refer to the ABAP documentation to see which expression is valid for which types

```
DATA lv_discount TYPE p LENGTH 1 DECIMALS 1
                 VALUE '0.8'.

SELECT ( 1 + 1 ) AS two,
       ( @lv_discount * gross_amount )
           AS red_gross_amount,
       CEIL( gross_amount )
           AS ceiled_gross_amount
  FROM snwd_so
  INTO TABLE @DATA(lt_result).
```

String Expressions

- Concatenate character columns with the && operator

```
SELECT company_name
    && ' (' && legal_form && ')'
  FROM snwd_bpa
  INTO TABLE @DATA(lt_result).
```

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

Recent Open SQL Enhancements – Part 1

What's Next?

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

Thank you

Contact information:

open@sap.com

openSAP

© 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.

Week 3 Unit 2: **Recent Open SQL Enhancements – Part 2**

SAP

openSAP

Recent Open SQL Enhancements – Part 2
Outline

Content

- Open SQL enhancements
  - SELECT list (cont.)
  - GROUP BY and HAVING clauses
  - JOIN statements
  - Client handling
- More information

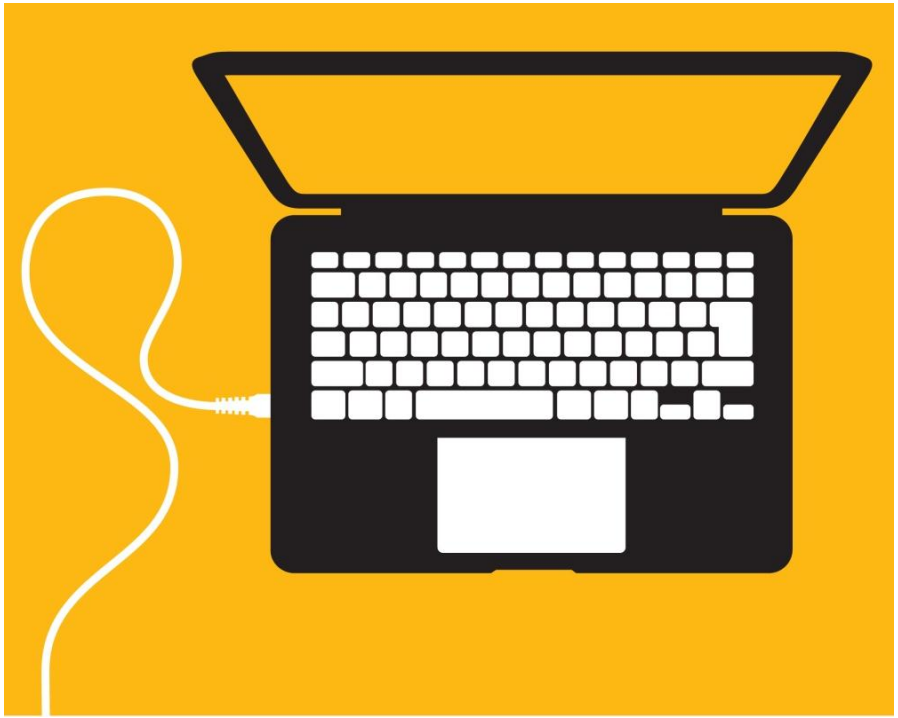
© 2014 SAP AG or an SAP affiliate company. All rights reserved

Recent Open SQL Enhancements – Part 2

Open SQL: Demo

Open SQL enhancements

- SELECT list enhancements:
  - Conditional expressions
- Expressions in
  - GROUP BY clause
  - HAVING clause
- JOIN statements
- Client handling



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

Recent Open SQL Enhancements – Part 2

Conditional Expressions

Additional Material

CASE Expression

```
"simple case
SELECT so_id,
       CASE delivery_status
         WHEN ' ' THEN 'OPEN'
         WHEN 'D' THEN 'DELIVERED'
         ELSE delivery_status
       END AS delivery_status_long
  FROM snwd_so
  INTO TABLE @DATA(lt_simple_case).

"searched case
SELECT so_id,
      CASE
        WHEN gross_amount > 1000
          THEN 'High volume sales order'
        ELSE ' '
      END AS volumn_order
   FROM snwd_so
   INTO TABLE @DATA(lt_searched_case).
```

COALESCE Expression

```
SELECT so_id,
       so~gross_amount
             AS so_amount,
       inv_head~gross_amount
             AS inv_amount,
       "potential invoice amount
       COALESCE( inv_head~gross_amount,
                 so~gross_amount )
             AS expected_amount
  FROM snwd_so AS so
  LEFT OUTER JOIN
       snwd_so_inv_head AS inv_head
    ON inv_head~so_guid = so~node_key
  INTO TABLE @DATA(lt_result).
```

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

Recent Open SQL Enhancements – Part 2

Expressions in GROUP BY & HAVING Clauses

Additional Material

GROUP BY Clause

```
SELECT bp_id,
       company_name,
       so~currency_code,
       SUM( so~gross_amount )
         AS total_amount,
       CASE
         WHEN so~gross_amount < 1000
           THEN 'X'
         ELSE ' '
       END AS low_volume_flag,
       COUNT( * ) AS cnt_orders
 FROM snwd_so AS so
 INNER JOIN snwd_bpa AS bpa
 ON bpa~node_key = so~buyer_guid
 INTO TABLE @DATA(lt_result)
 GROUP BY
   bp_id, company_name,
   so~currency_code,
   CASE
     WHEN so~gross_amount < 1000
       THEN 'X'
       ELSE ' '
     END
  ORDER BY company_name.
```

HAVING Clause

```
SELECT bp_id,
       company_name,
       so~currency_code,
       SUM( so~gross_amount )
        AS total_amount
   FROM snwd_so AS so
   INNER JOIN snwd_bpa AS bpa
   ON bpa~node_key = so~buyer_guid
   INTO TABLE @DATA(lt_result)
   WHERE so~delivery_status = ' '
   GROUP BY
     bp_id,
     company_name,
     so~currency_code
   HAVING SUM( so~gross_amount ) > 10000000.
```

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

Recent Open SQL Enhancements – Part 2

Support for JOIN Statements

Additional Material

Enhancements

- Now available: **RIGHT OUTER JOIN**
- Enhanced **bracketing** in JOIN expressions
- New functionality in **ON** conditions of JOIN statements like:
  - **Necessary requirement** of a field of the right table in the ON condition is dropped
  - Operators like **BETWEEN** or "**>**" can be used for comparisons
  - Possibility to use **fields of the right table** in the **WHERE** clause of LEFT OUTER JOINs
- Restriction of **maximum number of tables** in JOINs has been increased to 50

```
SELECT
    so_id,
    bp_id,
    gross_amount
 FROM snwd_so AS so
 RIGHT OUTER JOIN snwd_bpa AS bpa
   ON so~buyer_guid = bpa~node_key
  AND so~gross_amount > 100000
 INTO TABLE @DATA(lt_result).
```

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

Recent Open SQL Enhancements – Part 2

Automatic Client Handling

Additional Material

Automatic Client Handling

- Well known Open SQL client handling
- Client handling can be overruled with **USING CLIENT**
- Simplified/improved client handling in JOINs

```
SELECT
  bp_id,
  company_name,
  so~currency_code,
  so~gross_amount
FROM snwd_so AS so
INNER JOIN snwd_bpa AS bpa
  ON so~buyer_guid = bpa~node_key
  USING CLIENT '111'
INTO TABLE @DATA(lt_result).
```

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

Recent Open SQL Enhancements – Part 2

More Information

More information about recent Open SQL enhancements:

- Help Portal (http://help.sap.com/nw74): What‘s New – Release Notes for Support Package 05 & 08
- SCN blog: http://scn.sap.com/community/abap/blog/2014/02/06/abap-news-for-release-740-sp05

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

Recent Open SQL Enhancements – Part 2

What’s Next?

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

Thank you

Contact information:

open@sap.com

openSAP

© 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.

Week 3 Unit 3: **Core Data Services: What's in It for You?**

SAP

openSAP

Core Data Services: What's in It for You?

Outline

Content

- Introduction to Core Data Services (CDS)
- Core Data Services in ABAP
- Consumption of CDS views
- CDS vs. Open SQL

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

Core Data Services: What's in It for You?

Migration to SAP HANA

1. Detect

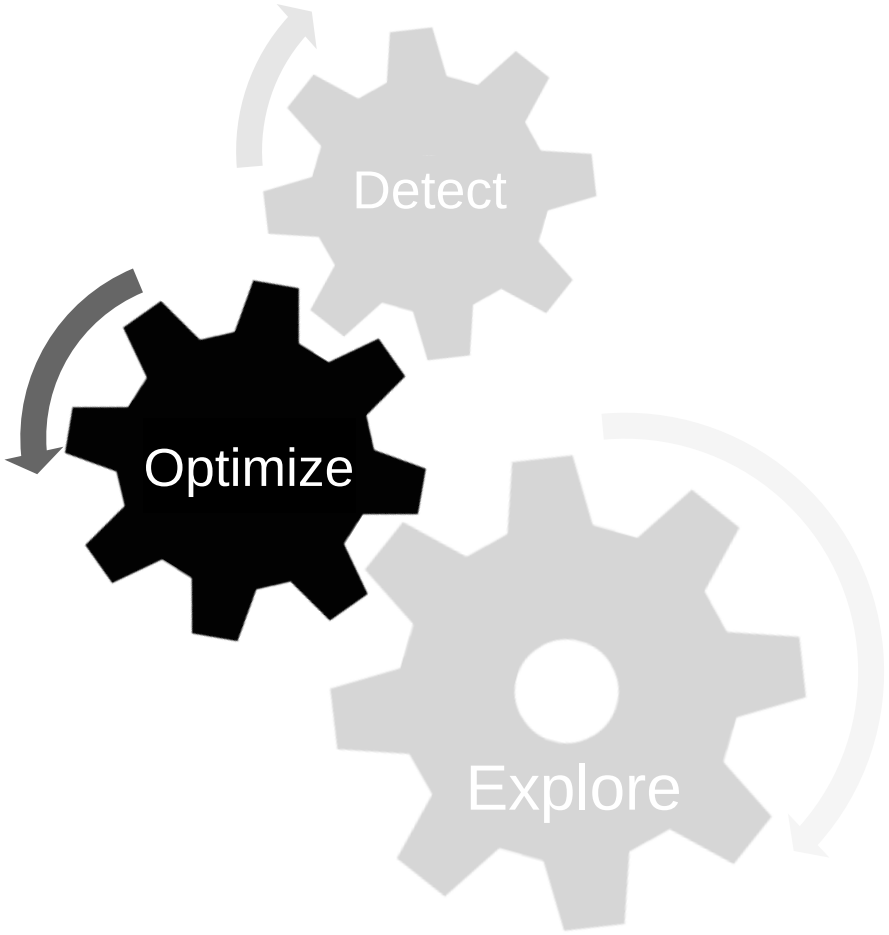
- Functional correctness
- Performance optimization potential

2. Optimize

- Database-oriented programming

3. Explore

- Use SAP HANA-specific features
- Rethink & innovate



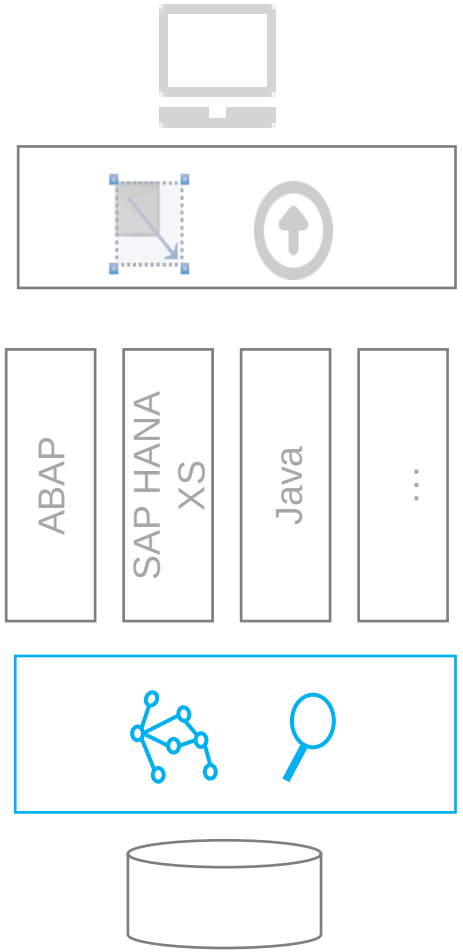
© 2014 SAP AG or an SAP affiliate company. All rights reserved

Core Data Services: What's in It for You?

Introduction to Core Data Services (CDS)

Core Data Services

- Next generation of data definition and access for database-centric applications
- Optimized application programming model for all domains (transactional, analytical,…)
- Technically an extension to SQL:
  - Expressions
  - Domain-specific metadata
  - Associations
- CDS includes
  - Data Definition Language (**DDL**)
  - Query language (**QL**)
  - Data Manipulation Language (**DML**)
  - Data control language (**DCL**)



User Interface

UI Abstractions

Application Programming

Core Data Services

Database

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

Core Data Services: What's in It for You?

Core Data Services in ABAP

Code-to-Data paradigm

- Supported through extended view functionality

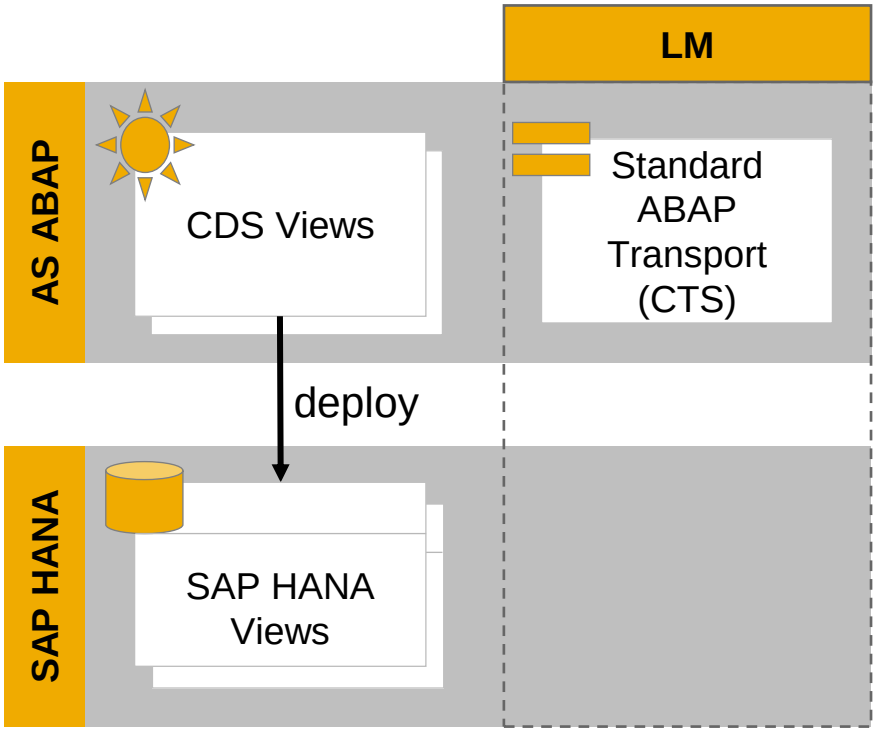
Definition of semantically rich data models in the ABAP Dictionary

- ABAP ‘view entities’ in DDL source objects (`R3TR DDLS`)

Fully integrated into the ABAP infrastructure

- Consistent lifecycle management with all other ABAP artifacts

Consumption via Open SQL on view entities



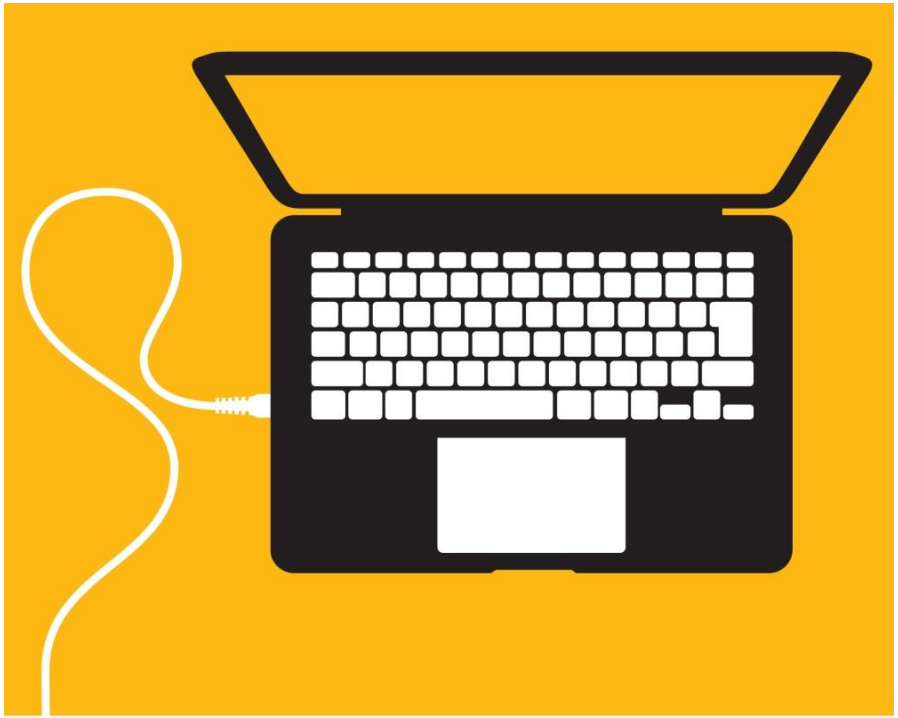
© 2014 SAP AG or an SAP affiliate company. All rights reserved

Core Data Services: What's in It for You?

Core Data Services: Demo

ABAP CDS View Demo

- Advanced View Definition in ABAP
- Data Preview
- Open SQL Consumption



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

Core Data Services: What's in It for You?

Simple CDS View & Open SQL Consumption

Additional Material

Definition & Consumption of an ABAP CDS View

- Definition in an ABAP DDL Source (R3TR DDLS)
- Definition only possible with ABAP Development Tools in Eclipse (**not** via transaction SE11)
- Consumption via
  - Open SQL
  - Data Preview (context menu in ADT)
  - SAP List Viewer
  - SAP NetWeaver Gateway (OData Model)
  - …

```
@AbapCatalog.sqlViewName:'ZDDLS_CDS_00'
define view zcdsv_simple as
  select from snwd_so
{
  key so_id as order_id,
  currency_code,
  gross_amount
}
```

```
DATA lt_cds TYPE
        STANDARD TABLE OF zcdsv_simple.

SELECT *
  FROM zcdsv_simple
  INTO TABLE @lt_cds.
```

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

Core Data Services: What's in It for You?

CDS Views vs. Open SQL

~~Competitors?~~

Not at all!

Use "plain" Open SQL if you

- … only need the query **"once"**, that is, in the coding (which might still be executed several times)
- … need features only available in Open SQL like `FOR ALL ENTRIES`

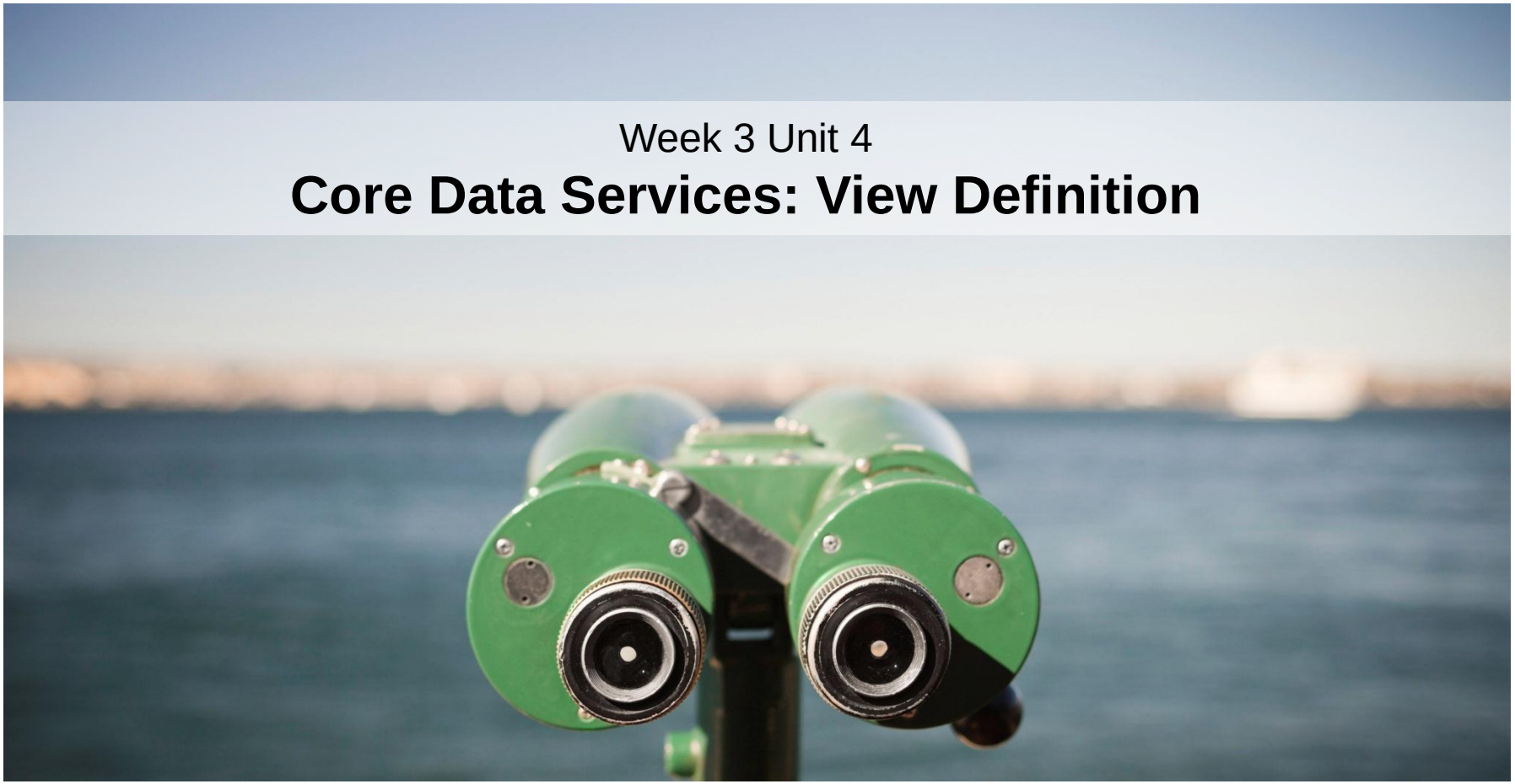
Use CDS views if you

- … have a real **re-use** case (same argument as for "old" dictionary views)
- … need features currently only available in DDL sources like `UNION`, `UNION ALL`, and so on
- … want to use features you will learn about in the upcoming units

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

Core Data Services: What's in It for You?

What’s Next?



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

Thank you

Contact information:

open@sap.com

openSAP

© 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.

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

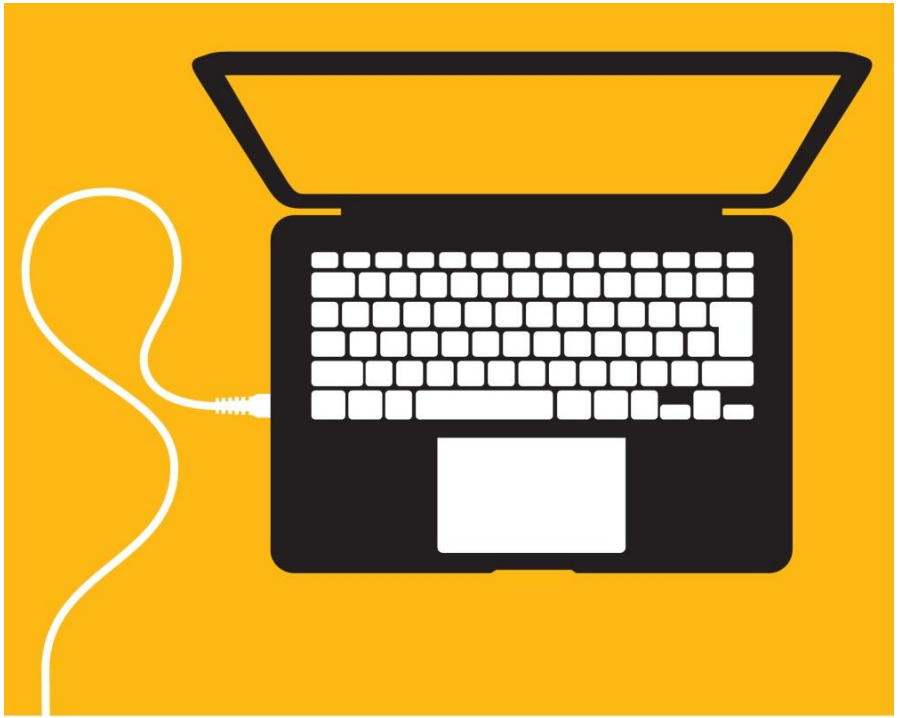
© 2014 SAP AG or an SAP affiliate company. All rights reserved

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



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

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: 'ZDDLS_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
```

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

Core Data Services: View Definition

Projection List (2)

Additional Material

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: 'ZDDLS_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
}
```

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

Core Data Services: View Definition

Projection List (3)

Additional Material

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: 'ZDDLS_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
}
```

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

Core Data Services: View Definition

View-Concept

Additional Material

View-on-View

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

```
@AbapCatalog.sqlViewName: 'ZDDLS_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: 'ZDDLS_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
}
```

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

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: 'ZDDLS_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: 'ZDDLS_CDS_13C'
extend view zcdsv_base with
zcdsv_customer_extension
{
  so.delivery_status,
  so.billing_status,
  so.created_at,
  so.created_by
}
```

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

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
  - …

```
@AbapCatalog.sqlViewName: 'ZDDLS_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
```

Not supported on all databases
→ DBSYS-dependent feature

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

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: 'ZDDLS_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: 'ZDDLS_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
```

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

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

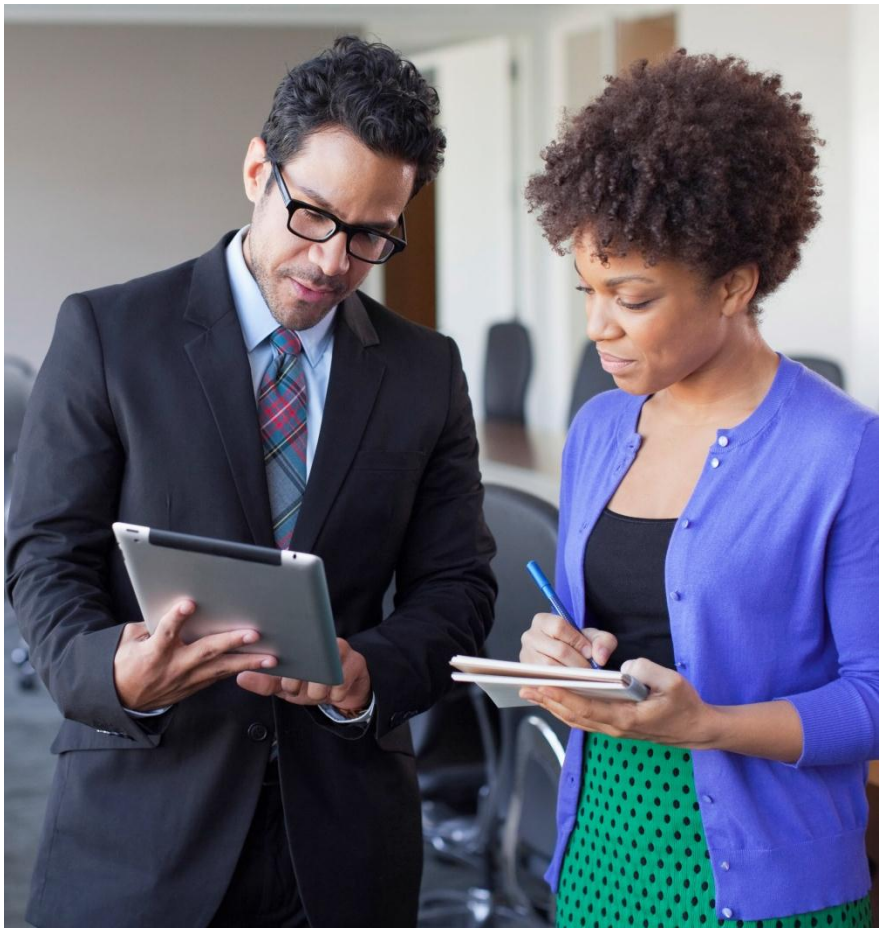
```
REPORT zr_cds_01_consumption_vwp.

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 ).
```

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

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)



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

Core Data Services: View Definition

What's Next?

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

Thank you

Contact information:

open@sap.com

openSAP

© 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.

Week 3 Unit 5: **Core Data Services: Associations**

Core Data Services: Associations
Outline

Content

- CDS Views: UNION and JOIN Support
- CDS Associations:
  - Definition
  - Consumption
- Association Types
  - Ad-Hoc Association
  - Exposed Association
- Filtered Associations
- Advantages of Associations

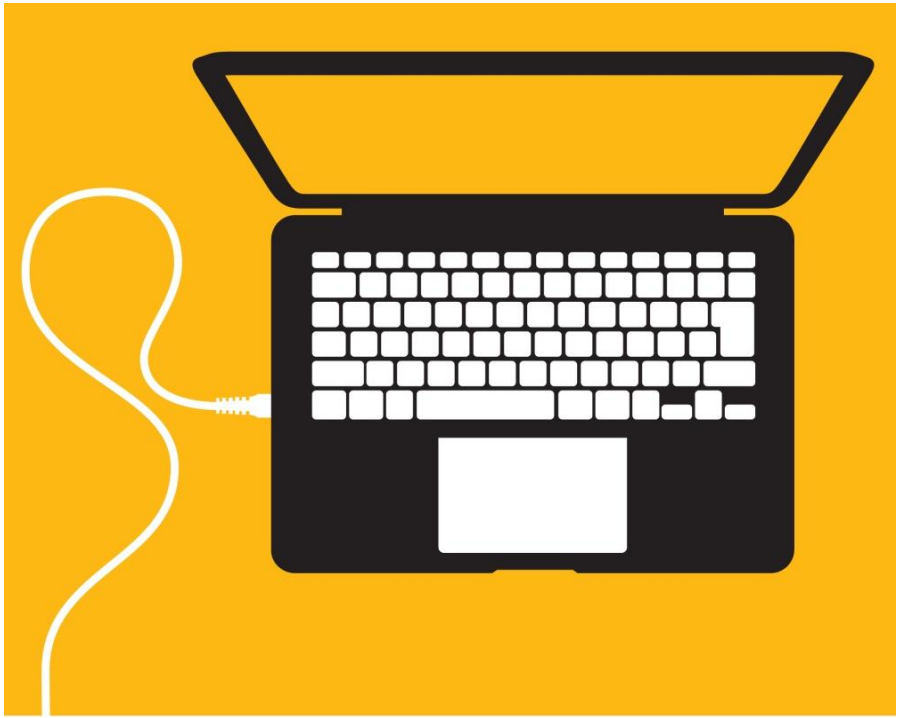
© 2014 SAP AG or an SAP affiliate company. All rights reserved

Core Data Services: Associations

CDS View Definition: Demo

CDS View Definition Features

- UNION (ALL)
- JOIN



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

Core Data Services: Associations

UNION & UNION ALL Statements

Additional Material

- Concatenation of different queries using **UNION (ALL)** construct
- Select lists of the different queries must
  - have the same number of columns
  - contain compatible types
- **UNION** implies a distinct semantic
- **UNION ALL** does not remove duplicates, so it does not imply the distinct semantic
- Do not mix **UNION** & **UNION ALL** in a CDS View

```
@AbapCatalog.sqlViewName: 'ZDDLS_CDS_20'
define view zcdsv_union as
select from snwd_so as so
  inner join snwd_bpa as bpa
     on so.buyer_guid = bpa.node_key
 {
    key bpa.bp_id,
    bpa.company_name,
    sum( gross_amount ) as total_gross_amount,
    'small' as category
 }
group by bpa.bp_id, bpa.company_name
having sum( gross_amount ) < 10000000

union all

select from snwd_so as so
  inner join snwd_bpa as bpa
     on so.buyer_guid = bpa.node_key
 {
   key bpa.bp_id,
   bpa.company_name,
   sum( gross_amount ) as total_gross_amount,
   'large' as category
 }
group by bpa.bp_id, bpa.company_name
having sum( gross_amount ) >= 10000000
```

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

Core Data Services: Associations

JOIN Statements

Additional Material

- Supported join types:
  - **INNER** join
  - **LEFT OUTER** join
  - **RIGHT OUTER** join
- Complex join operations using (…) are supported
- Arbitrary on-conditions (including >, >=, <, <=, like between, and, or, not)

```
@AbapCatalog.sqlViewName: 'ZDDLS_CDS_21'
define view zcdsv_join as
    select from snwd_so as so
    inner join snwd_bpa as bpa
        on so.buyer_guid = bpa.node_key
    left outer join snwd_so_inv_head as
inv_head on so.node_key = inv_head.so_guid
 {
    key so.so_id,
    bpa.company_name,
    so.delivery_status,
    inv_head.payment_status
 }
```

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

Core Data Services: Associations
Concept

CDS Associations

- Definition of relationships between entities
- Association definition contains
  - Target entity
  - Cardinality […] (optional)
  - Alias name (optional)
  - ON condition representing the JOIN

Associations replace JOINs with simple path expressions

- Usage of aliases improves readability / model understanding
- Path expressions support
  - simplified consumption
  - easy refactoring of ON conditions

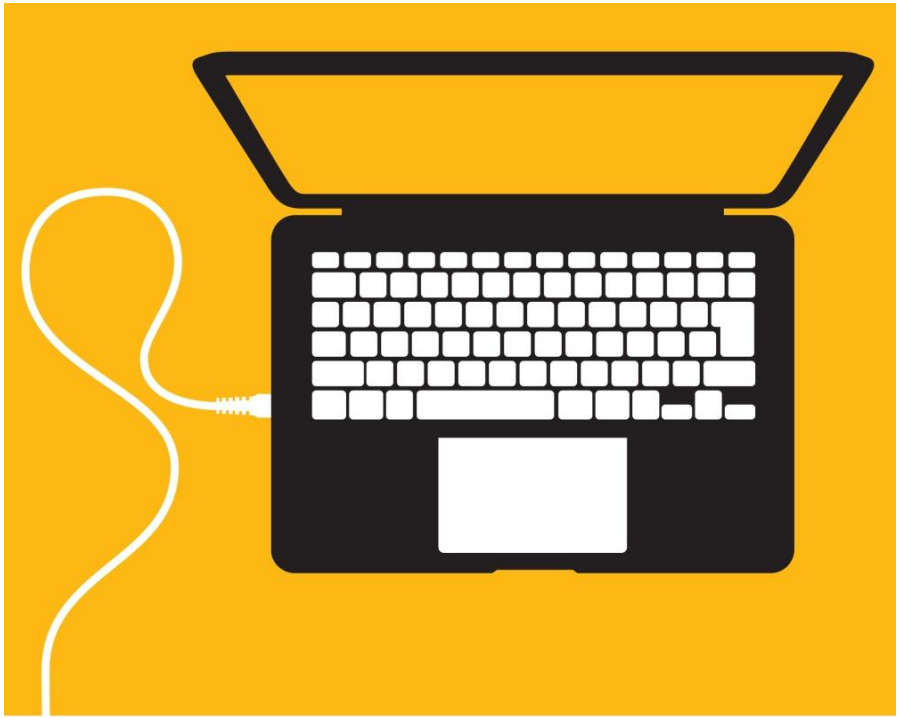
© 2014 SAP AG or an SAP affiliate company. All rights reserved

Core Data Services: Associations

Demo

CDS View with Associations

- CDS Association
  - Ad-Hoc Association
  - Exposed Association
- Consumption of an Association
  - Projection List
  - FROM Clause
  - Arithmetic Expressions
  - WHERE and HAVING Clauses
  - …
- Path Expression
  - Filter Conditions



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

Core Data Services: Associations

Simple Association

Additional Material

- Definition of associations within a CDS view
- Consumption of the association, for example in the projection list or in the WHERE clause
- Consumption of the association in an aggregation, the GROUP BY, and the HAVING clause

```
@AbapCatalog.sqlViewName: 'ZDDLS_CDS_30'
define view zcdsv_simple_assoc_examples as
    select from snwd_so as so
association [1] to snwd_bpa as business_partners
        on so.buyer_guid =
business_partners.node_key
association [0..1] to snwd_so_inv_head as
invoice_headers
        on so.node_key = invoice_headers.so_guid
 {
    key so.so_id as order_id,
    so.delivery_status,
    invoice_headers.payment_status,
    invoice_headers.currency_code,
    sum( invoice_headers.gross_amount )
      as total_gross_amount
 }
 where business_partners.company_name = 'SAP'
 group by so.so_id,
          so.delivery_status,
          invoice_headers.payment_status,
          invoice_headers.currency_code
 having sum( invoice_headers.gross_amount ) > 3000
```

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

Core Data Services: Associations

Association Types

Additional Material

Ad Hoc Associations

- Association definition and usage in the same CDS view

  → Association consumption constitutes a JOIN

Exposed Association

- Association definition, exposure of association, and exposure of fields used in the ON condition
  Consumption of association, for example in a view on the view
  → Exposure does not automatically lead to a JOIN: “JOINs on demand”

```
@AbapCatalog.sqlViewName: 'ZDDLS_CDS_31A'
define view zcdsv_assoc_types as
    select from snwd_so as so
association [1] to snwd_bpa as business_partners
         on so.buyer_guid = $projection.buyer_guid
association [0..1] to snwd_so_inv_head as
invoice_headers
         on so.buyer_guid = invoice_headers.so_guid
 {
    key so.so_id as order_id,
    so.delivery_status,
    -- ad hoc association
    invoice_headers.payment_status,

    --exposed association
    -- field used in the ON condition
    so.buyer_guid,
    -- exposing association business_partners
    business_partners
 }
```

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

Core Data Services: Associations

Filtered Associations

Additional Material

Filtered Association

- Base view: Definition of an association with a “to-n” cardinality
- Filter expression given in squared brackets
- Explicit cardinality of applied filter condition

```
@AbapCatalog.sqlViewName: 'ZDDLS_CDS_32A'
define view zcdsv_filter_example_base as
  select from snwd_so_inv_head as invoice_header
  association[1..*] to snwd_so_inv_item as invoice_items
   on $projection.header_guid = invoice_items.parent_key
 {
   invoice_header.so_guid  as order_guid,
   invoice_header.node_key as header_guid,
   invoice_items
 }
```

```
@AbapCatalog.sqlViewName: 'ZDDLS_CDS_32B'
define view zcdsv_filter_example_vov as
    select from snwd_so as so
association [1] to snwd_bpa as business_partners
        on so.buyer_guid = business_partners.node_key
association [0..1] to zcdsv_filter_example_base as invoice_headers
        on so.node_key = invoice_headers.order_guid
 {
    key so.so_id as order_id,

    -- value 01 means customer
    business_partners[ bp_role = '01' ].company_name as customer_name,

    -- filter 1..n association on first position
    invoice_headers.invoice_items[1: inv_item_pos =
'0000000010'].currency_code,
    invoice_headers.invoice_items[1: inv_item_pos =
'0000000010'].gross_amount
 }
 where invoice_headers.header_guid is not null
```

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

Core Data Services: Associations

Advanced Example for Filtered Associations

Additional Material

Filtered Association

- First view defines a normalized view on the EPM text table
- Second view defines an association between the product table and the text information encoded in the first view
- Third view consumes the association, filtering the language to the given input parameter value or a default value

```
@AbapCatalog.sqlViewName: 'ZDDLS_CDS_33A'
define view zcdsv_adv_filter_example_base
as select from snwd_texts
{
  parent_key as product_text_guid,
  language,
  text
}
```

```
@AbapCatalog.sqlViewName: 'ZDDLS_CDS_33B'
define view zcdsv_adv_filter_example_l1
as select from snwd_pd as pd
association [1..*] to zcdsv_adv_filter_example_base as texts
  on texts.product_text_guid = $projection.text_guid
{
 key pd.product_id,
 pd.desc_guid as text_guid,
 texts
}
```

```
@AbapCatalog.sqlViewName: 'ZDDLS_CDS_33C'
define view zcdsv_adv_filter_example_l2
with parameters lang : abap.lang
as select from zcdsv_adv_filter_example_l1 as product
{
  $parameters.lang as langu_filter_param,
  coalesce (
            product.texts[1: language = $parameters.lang ].text,
            product.texts[1: language = 'E'              ].text
          ) as product_description
}
```

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

Core Data Services: Associations

Advantages of Associations

Why would you use associations?

- Easy model consumption
  - Path expressions
  - Filter expressions
- Small(er) re-use views
- “JOINs on demand”:
  JOINs are only generated if the corresponding association is consumed

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

Core Data Services: Associations

What's Next?

Week 4 Unit 1

Native SAP HANA Usage in ABAP

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

Thank you

Contact information:

open@sap.com

openSAP

© 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.