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

Core Data Services: Associations
Outline

Content

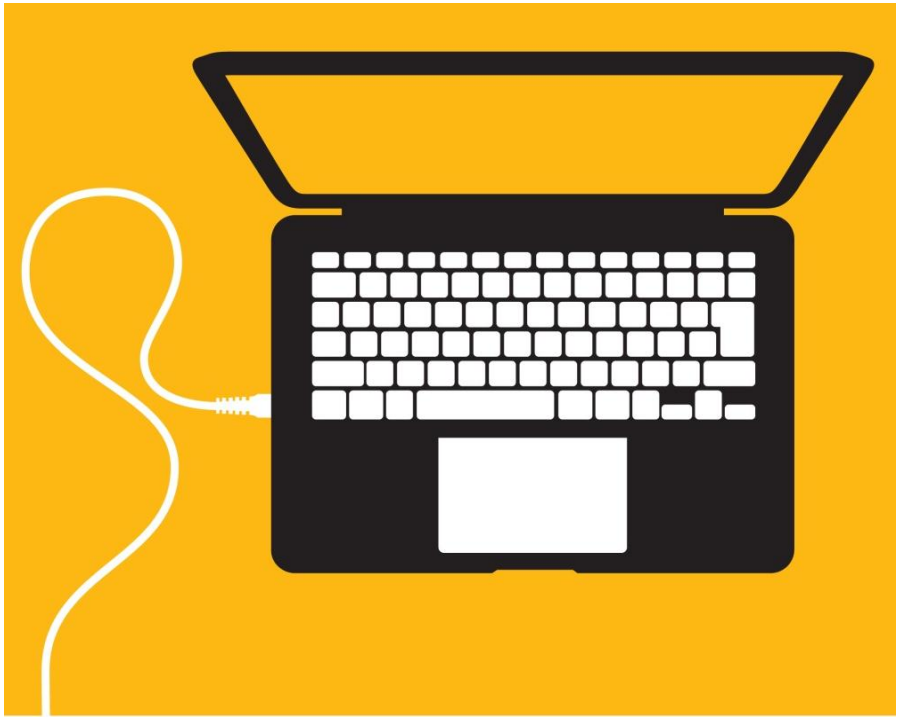
- CDS Views: UNION and JOIN Support
- CDS Associations:
  - Definition
  - Consumption
- Association Types
  - Ad-Hoc Association
  - Exposed Association
- Filtered Associations
- Advantages of Associations

Core Data Services: Associations

CDS View Definition: Demo

CDS View Definition Features

- UNION (ALL)
- JOIN



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
```

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
 }
```

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

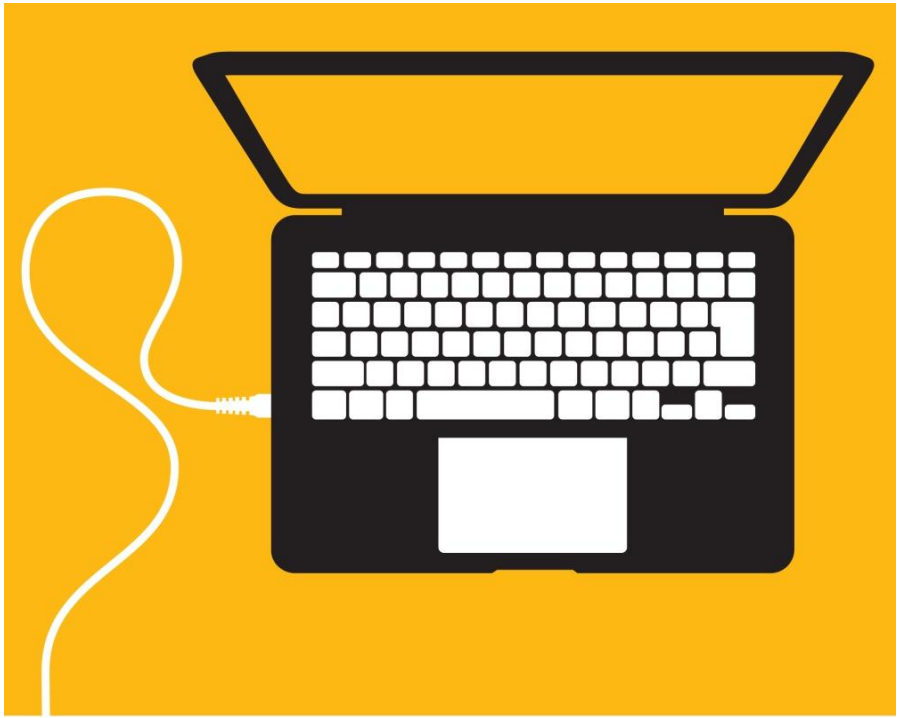
- Usage of aliases improves readability / model understanding
- Path expressions support
  - simplified consumption
  - easy refactoring of ON conditions

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



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
```

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
 }
```

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
```

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
}
```

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

Core Data Services: Associations

What's Next?

Week 4 Unit 1

Native SAP HANA Usage in ABAP

Thank you

Contact information:

open@sap.com

openSAP

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