****

**ABAP Development for SAP HANA**

[**https://open.sap.com/courses/a4h1**](https://open.sap.com/courses/a4h1)

**Source Code of Week 3**



Version 1 - 15.10.2014

Only Demo content

Without any warranty

Contents

[CDS DDL Sources 3](#_Toc401149039)

[ZDDLS\_CDS\_00\_SIMPLE 3](#_Toc401149040)

[ZDDLS\_CDS\_10\_AGG\_AND\_LIT 3](#_Toc401149041)

[ZDDLS\_CDS\_11\_ARITH\_AND\_CAST 3](#_Toc401149042)

[ZDDLS\_CDS\_12\_CONDITIONAL\_EXP 4](#_Toc401149043)

[ZDDLS\_CDS\_13A\_BASE 4](#_Toc401149044)

[ZDDLS\_CDS\_13B\_VIEW\_ON\_VIEW 5](#_Toc401149045)

[ZDDLS\_CDS\_13C\_VIEW\_EXTENSION 5](#_Toc401149046)

[ZDDLS\_CDS\_14A\_INPUT\_PARAMETERS 5](#_Toc401149047)

[ZDDLS\_CDS\_14B\_CONSUMPTION 6](#_Toc401149048)

[ZDDLS\_CDS\_20\_UNION 6](#_Toc401149049)

[ZDDLS\_CDS\_21\_JOIN 6](#_Toc401149050)

[ZDDLS\_CDS\_30\_ASSOC\_SIMPLE 7](#_Toc401149051)

[ZDDLS\_CDS\_31A\_ASSOC\_TYPES 7](#_Toc401149052)

[ZDDLS\_CDS\_31B\_ASSOC\_TYPES 8](#_Toc401149053)

[ZDDLS\_CDS\_32A\_ASSOC\_FILTER 8](#_Toc401149054)

[ZDDLS\_CDS\_32B\_ASSOC\_FILTER 8](#_Toc401149055)

[ZDDLS\_CDS\_33A\_ADV\_ASSOC\_FILTER 9](#_Toc401149056)

[ZDDLS\_CDS\_33B\_ADV\_ASSOC\_FILTER 9](#_Toc401149057)

[ZDDLS\_CDS\_33C\_ADV\_ASSOC\_FILTER 9](#_Toc401149058)

[ABAP Reports 11](#_Toc401149059)

[ZR\_CDS\_00\_CONSUMPTION 11](#_Toc401149060)

[ZR\_CDS\_01\_CONSUMPTION\_VWP 11](#_Toc401149061)

[ZR\_OPENSQL\_00\_SYNTAX 12](#_Toc401149062)

[ZR\_OPENSQL\_01\_AGGREGATION 12](#_Toc401149063)

[ZR\_OPENSQL\_02\_LITERALS 13](#_Toc401149064)

[ZR\_OPENSQL\_03\_GENERIC\_EXIST 13](#_Toc401149065)

[ZR\_OPENSQL\_04\_ARITH\_EXPRESS 13](#_Toc401149066)

[ZR\_OPENSQL\_05\_CONCAT 14](#_Toc401149067)

[ZR\_OPENSQL\_10\_CASE 14](#_Toc401149068)

[ZR\_OPENSQL\_11\_COALESCE 15](#_Toc401149069)

[ZR\_OPENSQL\_12\_GROUP\_BY 15](#_Toc401149070)

[ZR\_OPENSQL\_13\_HAVING 16](#_Toc401149071)

[ZR\_OPENSQL\_14\_JOIN 16](#_Toc401149072)

[ZR\_OPENSQL\_15\_CLIENT\_HANDLING 16](#_Toc401149073)

# CDS DDL Sources

## ZDDLS\_CDS\_00\_SIMPLE

@AbapCatalog.sqlViewName: 'ZDDLS\_CDS\_00'

**define** **view** ZCDSV\_SIMPLE **as** **select** **from** snwd\_so **{**

snwd\_so**.**so\_id **as** order\_id**,**

snwd\_so**.**gross\_amount**,**

snwd\_so**.**currency\_code

**}**

## ZDDLS\_CDS\_10\_AGG\_AND\_LIT

@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** bpa**.**bp\_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

## ZDDLS\_CDS\_11\_ARITH\_AND\_CAST

@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

**}**

## ZDDLS\_CDS\_12\_CONDITIONAL\_EXP

@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

**}**

@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

**}**

## ZDDLS\_CDS\_13A\_BASE

@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

**}**

## ZDDLS\_CDS\_13B\_VIEW\_ON\_VIEW

@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

**}**

## ZDDLS\_CDS\_13C\_VIEW\_EXTENSION

@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

**}**

## ZDDLS\_CDS\_14A\_INPUT\_PARAMETERS

@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

## ZDDLS\_CDS\_14B\_CONSUMPTION

@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

**}**

## ZDDLS\_CDS\_20\_UNION

@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

## ZDDLS\_CDS\_21\_JOIN

@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

**}**

## ZDDLS\_CDS\_30\_ASSOC\_SIMPLE

@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

-- left outer join snwd\_so\_inv\_head as inv\_head

-- on so.node\_key = inv\_head.so\_guid

**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

## ZDDLS\_CDS\_31A\_ASSOC\_TYPES

@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

**}**

## ZDDLS\_CDS\_31B\_ASSOC\_TYPES

@AbapCatalog.sqlViewName: 'ZDDLS\_CDS\_31B'

**define** **view** zcdsv\_assoc\_types\_consumption **as**

**select** **from** zcdsv\_assoc\_types

**{**

zcdsv\_assoc\_types**.**business\_partners**.**bp\_id

**}**

## ZDDLS\_CDS\_32A\_ASSOC\_FILTER

@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

**}**

## ZDDLS\_CDS\_32B\_ASSOC\_FILTER

@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**

## ZDDLS\_CDS\_33A\_ADV\_ASSOC\_FILTER

@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

**}**

## ZDDLS\_CDS\_33B\_ADV\_ASSOC\_FILTER

@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

**}**

## ZDDLS\_CDS\_33C\_ADV\_ASSOC\_FILTER

@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

**}**

# ABAP Reports

## ZR\_CDS\_00\_CONSUMPTION

REPORT zr\_cds\_00\_consumption.

"data definition using a CDS view as type

DATA lt\_cds TYPE STANDARD TABLE OF zcdsv\_simple.

"consumption of a CDS view

SELECT \* FROM zcdsv\_simple INTO TABLE @lt\_cds.

## ZR\_CDS\_01\_CONSUMPTION\_VWP

REPORT zr\_cds\_01\_consumption\_vwp.

DATA lv\_cust\_name TYPE c LENGTH 80 VALUE 'SAP'.

"plenty of ABAP application logic

"check if the feature is supported by the underlying database

DATA(lv\_feature\_supported) =

cl\_abap\_dbfeatures=>use\_features(

EXPORTING

requested\_features = VALUE #( ( cl\_abap\_dbfeatures=>views\_with\_parameters ) )

).

"if the feature is supported, select from the CDS view and

"provide a value for the input parameter

IF lv\_feature\_supported = abap\_true.

##DB\_FEATURE\_MODE[VIEWS\_WITH\_PARAMETERS]

SELECT \*

FROM zcdsv\_with\_input\_parameters( customer\_name = 'SAP' )

INTO TABLE @DATA(lt\_result).

ELSE.

"do some alternative coding here

ENDIF.

"even more awesome application logic

"display of results

cl\_demo\_output=>display\_data( lt\_result ).

## ZR\_OPENSQL\_00\_SYNTAX

REPORT zr\_opensql\_00\_syntax.

TYPES: BEGIN OF ty\_so\_amount,

so\_id TYPE snwd\_so-so\_id,

currency\_code TYPE snwd\_so-currency\_code,

gross\_amount TYPE snwd\_so-gross\_amount,

END OF ty\_so\_amount.

DATA lt\_so\_amount TYPE STANDARD TABLE OF ty\_so\_amount.

"using the "old" Open SQL syntax

SELECT so\_id

currency\_code

gross\_amount

FROM snwd\_so

INTO TABLE lt\_so\_amount.

"using the "new" Open SQL syntax

SELECT so\_id,

currency\_code,

gross\_amount

FROM snwd\_so

INTO TABLE @lt\_so\_amount.

"demo of the result set type inference

SELECT so\_id,

currency\_code,

gross\_amount,

delivery\_status

FROM snwd\_so

INTO TABLE @DATA(lt\_result).

## ZR\_OPENSQL\_01\_AGGREGATION

REPORT zr\_opensql\_01\_aggregation.

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.

cl\_demo\_output=>display\_data( value = lt\_result ).

## ZR\_OPENSQL\_02\_LITERALS

REPORT zr\_opensql\_01\_aggregation.

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.

cl\_demo\_output=>display\_data( value = lt\_result ).

## ZR\_OPENSQL\_03\_GENERIC\_EXIST

REPORT zr\_opensql\_03\_generic\_exist.

DATA lv\_exists TYPE abap\_bool VALUE abap\_false.

"generic check if a sales order exists

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.

## ZR\_OPENSQL\_04\_ARITH\_EXPRESS

REPORT zr\_opensql\_04\_arith\_express.

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).

## ZR\_OPENSQL\_05\_CONCAT

REPORT zr\_opensql\_05\_concat.

SELECT company\_name

&& ' (' && legal\_form && ')'

FROM snwd\_bpa

INTO TABLE @DATA(lt\_result).

## ZR\_OPENSQL\_10\_CASE

REPORT zr\_opensql\_10\_case.

"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).

"searched case - more advanced

DATA lv\_discount TYPE p LENGTH 1 DECIMALS 1 VALUE '0.8'.

SELECT so\_id,

company\_name,

gross\_amount AS orig\_amount,

CASE WHEN company\_name = 'SAP'

THEN ( gross\_amount \* @lv\_discount )

ELSE gross\_amount

END AS discount\_amount

FROM snwd\_so AS so

INNER JOIN snwd\_bpa AS bpa

ON so~buyer\_guid = bpa~node\_key

INTO TABLE @DATA(lt\_adv\_example\_searched\_case).

## ZR\_OPENSQL\_11\_COALESCE

REPORT zr\_opensql\_11\_coalesce.

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).

## ZR\_OPENSQL\_12\_GROUP\_BY

REPORT zr\_opensql\_12\_group\_by.

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.

cl\_demo\_output=>display\_data( value = lt\_result ).

## ZR\_OPENSQL\_13\_HAVING

REPORT zr\_opensql\_13\_having.

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.

cl\_demo\_output=>display\_data( value = lt\_result ).

## ZR\_OPENSQL\_14\_JOIN

REPORT zr\_opensql\_14\_join.

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).

## ZR\_OPENSQL\_15\_CLIENT\_HANDLING

REPORT zr\_opensql\_15\_client\_handling.

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

"client 111 does not exist in the system

USING CLIENT '111'

INTO TABLE @DATA(lt\_result).

cl\_demo\_output=>display\_data( value = lt\_result ).