



Important SAP Community Log-in Update!

Starting on September 5, the SAP Community will switch to SAP Universal ID as the only option to login. Create your SAP Universal ID now! If you have multiple S- or P- accounts, link your accounts to SAP Universal ID.

[Switch to SAP Universal ID Today!](#)[Ask a Question](#) [Write a Blog Post](#)[Login](#)

Technical Articles



Masaaki Arai

August 20, 2018 | 7 minute read

Cardinality of Association in CDS View

21 49 85,312

[Follow](#)[Like](#)

Purpose of this blog:

This blog explains how the Cardinality of Association in ABAP CDS View works.

Association is used to link a view (or table) to the source view (or table) in CDS View. In Association, Cardinality is defined. Cardinality is the relationship between the source and associated view in the form of [min .. max] (only the target



cardinality is stated). When Sales Order is the source and Sales Organization is the target, the cardinality is [0..1] (min .. max). This means if the record in which Sales Organization 1200 is there in Sales Order View, in Sales Organization View, there is up to one record (min = 0 / max=1) in which Sales Organization is 1200.

This blog explains how the internal behavior is changed by the setting of Cardinality in Association.

My motivation to write the blog:

Association is one of the main strong characteristics of CDS View and cardinality is set for the association. Many would wonder what it is, and I think more concrete explanation would be needed than the existing ones.

Summary:

- Joins and path expressions along associations can duplicate records of the result set in the query, if the join cardinality is >1.
 - In the CDS View (A), a table (or CDS View) (B) is used as the source and a table (or CDS View) (C) is associated to B. In case [0..1] or [1..1] is used in the association, when no attributes of C are used in SELECT Statement, JOIN to C is not happening and C is not accessed internally. It could optimize the response time.
 - If relationship between B and C is 1 : N, as JOIN does not happen, in the result set of the SELECT statement, records from B is not duplicated.
 - In case [0..*] or [1..*] is used in the association, JOIN happens and C is accessed internally, even when no attributes of C are used in SELECT Statement. As a result, the runtime would be longer and records from B would be duplicated if the relationship between B and C is 1:N,
- When the relationship of the source and joined view (or table) is 1 : N, if LEFT OUTER JOIN is used in the CDS View, JOIN happens, but if LEFT OUTER TO ONE JOIN is used, JOIN does not happen. When the relationship is N:1, JOIN does not happen even if LEFT OUTER JOIN is used.
- When [0..*] or [1..*] is used in Association, a field of the associated view (or table) cannot be used in WHERE condition and cannot be added using Extend View either.

Cardinality of association:

Cardinality is the relationship between the source and associated CDS View (or table) , included in the definition of the association as [min..max]. See [SAP Help](#) in detail.

To avoid undefined and platform-dependent behavior, the cardinality should always be defined to match the data in question.

The specified cardinality is evaluated by the syntax check for paths specified in the CDS DDL of CDS or in ABAP SQL. A non-matching cardinality usually produces a syntax check warning. For an association using the default foreign key (see below) the target cardinality must be [0..1] or [1..1].

The CDS compiler technically only distinguishes the following cardinalities:

- to-zero-or-one, which is specified as [0..1] or [1] or by omitting cardinality specification
- to-exactly-one, which is specified as [1..1]
- to-many, which is specified as [] or [*] or [n..*]

Providing more detailed information (like [2..3]) is possible, but the CDS compiler does not make use of it. It is simply a possibility for the developer to document specific knowledge about the relationship.

Normally association works like LEFT OUTER JOIN internally. If INNER JOIN should be used, [inner] should be used in the fields added, e.g. _VB[inner].DOCB. In this case, associated view is accessed in any cases.

Test:

[Test 1 Cardinality Test](#)

[Test 2 LEFT OUTER JOIN Test](#)

Test 1 Cardinality Test:

Case (go to [Result](#))

In the CDS View ZI_CDSVAB01, table ZDOCB is associated to the source table ZDOCA using association. Association key is DOCA.

In each test case, the cardinality of the association is changed and SELECT statement is executed to the CDS View ZI_CDSVAB01. In this SELECT statement, only the fields in ZDOCA are included.

Cardinality of association is changed in each case. 1) [0..1], 2) [1..1], 3) [0..*], 4) [1..*].

CDS View: ZI_CDSVAB01

AB	DOCA	AB	CA	AB	MA1
	1200000000		A00		120.000
	1200000010		A10		130.000

Table: ZDOCA

AB	DOCB	AB	DOCA	AB	CB	AB	MB1	AB	CKY	AB	MB2	AB	CKY2
	2200000000		1200000000		B00		250.000		JPY		300.00		JPY
	2200000010		1200000000		B10		260.000		JPY		310.00		USD
	2200000020		1200000010		B20		270.000		USD		320.00		JPY
	2200000030		1200000010		B30		280.000		KRW		330.00		KRW

Table: ZDOCB

Source Table:

Table ZDOCA

@EndUserText.label : 'Document A'

@AbapCatalog.enhancementCategory : #NOT_EXTENSIBLE

@AbapCatalog.tableCategory : #TRANSPARENT

@AbapCatalog.deliveryClass : #A

```
@AbapCatalog.dataMaintenance : #ALLOWED
```

```
define table zdoca {
```

```
key doca : abap.char(10) not null;
```

```
ca : abap.char(3);
```

```
ma1 : abap.dec(17,3);
```

```
}
```

Table ZDOCB

```
@EndUserText.label : 'Document B'
```

```
@AbapCatalog.enhancementCategory : #NOT_EXTENSIBLE
```

```
@AbapCatalog.tableCategory : #TRANSPARENT
```

```
@AbapCatalog.deliveryClass : #A
```

```
@AbapCatalog.dataMaintenance : #ALLOWED
```

```
define table zdocb {
```

```
key docb : abap.char(10) not null;
```

```
key doca : abap.char(10) not null;
```

```
cb : abap.char(3);
```

```
mb1    : abap.dec(17,3);

cky     : abap.cuky;

@Semantics.amount.currencyCode : 'zdocb.cky2'

mb2     : abap.curr(17,2);

cky2    : abap.cuky;

}
```

CDS View: ZI_CDSVAB01

```
@AbapCatalog.sqlViewName: 'ZICDSVAB01'

@AbapCatalog.compiler.compareFilter: true

@AccessControl.authorizationCheck: #CHECK

@EndUserText.label: 'TEST'

define view ZI_CDSVAB01 as select from zdoca as _VA

association [0..1] to zdocb as _VB on _VA.doca = _VB.doca

{

  _VA.doca,

  _VA.ca,
```

```

_VB.docb,

_VB.cb,

@DefaultAggregation: #SUM

_VA.ma1,


@DefaultAggregation: #SUM


_VB.mb1,





}

```

Data Preview of ZI_CDSVAB01:

►  ZI_CDSVAB01 ►

 Raw Data

 Filter pattern   **4 rows retrieved - 13 ms**  SC

AB doca	AB ca	AB docb	AB cb	AB ma1	AB mb1
1200000000	A00	2200000000	B00	120.000	250.000
1200000000	A00	2200000010	B10	120.000	260.000
1200000010	A10	2200000020	B20	130.000	270.000
1200000010	A10	2200000030	B30	130.000	280.000

SELECT statement:

```
SELECT  
  
ZI_CDSVAB01~DOCA,  
  
sum( ZI_CDSVAB01~MA1 ) as MA1  
  
FROM  
  
ZI_CDSVAB01  
  
GROUP BY  
  
ZI_CDSVAB01~DOCA
```

Result (go to [Case](#))

Case1 association [0..1]

```
@EndUserText.label: 'TEST'
```

```
define view ZI_CDSVAB01 as select from zdoca as _VA  
association [0..1] to zdocb as _VB on _VA.doca = _VB.doca  
{  
  _VA.doca,
```

Executing SELECT statement:

Values are not duplicated.

Run Data Aging Max. Rows: 100

Raw Data

Filter pattern 2 rows retrieved - 1 ms

AB DOCA	AB MA1
1200000000	120.000
1200000010	130.000

```



SELECT
  ZI_CDSVAB01~DOCA,
  sum( ZI_CDSVAB01~MA1 ) as MA1
FROM
  ZI_CDSVAB01
GROUP BY
  ZI_CDSVAB01~DOCA
  
```

Explain:

(Take SQL trace with Transaction ST05 and display the trace. After aggregating the statement with the menu > Trace > Structure-Identical Statement, select the SELECT statement to access the CDS View, and run Edit > Display Execution Plan > For Recorded Statement.)

Only ZDOCA is accessed.

Display Execution Plan for SQL Statement

 Graphical Explain
  Execution Trace

SQL Statement

```

SELECT
  "DOCA" , SUM( "MA1" ) "MA1"
FROM
  /* Entity name: ZI_CDSVAB01 */ "ZICDSVAB01" "ZI_CDSVAB01"
GROUP BY
  "DOCA" LIMIT 100 WITH RANGE_RESTRICTION(' CURRENT' )
  
```

Execution Plan

Operation	Options	Object_name
PROJECT	ZICDSVAB01.DOCA, MA1	
LIMIT	NUM RECORDS: 100	
COLUMN TABLE		ZDOCA

Explain End

Case2 association [1..1]

```

~
@EndUserText.label: 'TEST'

define view ZI_CDSVAB01 as select from zdoca as _VA
association [1..1] to zdocb as _VB on _VA.doca = _VB.doca
{
  _VA.doca,
  _VA.ca,
  .
}
  
```

Executing SELECT statement:

Values are not duplicated.

Only ZDOCA is accessed.

11/39

Case3 association [0..*]

```
@EndUserText.label: 'TEST'
```

```
define view ZI_CDSVAB01 as select from zdoca as _VA  
association [0..*] to zdocb as _VB on _VA.doca = _VB.doca  
{  
  _VA.doca,  
  _VA.ca,
```

Executing SELECT statement:

Values are duplicated.

Run Data Aging Max. Rows: 100

Raw Data


Filter pattern 2 rows retrieved - 20 ms

AB	DOCA	AB	MA1
	1200000010		260.000
	1200000000		240.000



```
SELECT  
  ZI_CDSVAB01~DOCA,  
  sum( ZI_CDSVAB01~MA1 ) as MA1  
FROM  
  ZI_CDSVAB01  
GROUP BY  
  ZI_CDSVAB01~DOCA
```

Explain:

ZDOCA and ZDOCB are accessed.

Menu 

Display Execution Plan for SQL Statement

 Graphical Explain  Execution Trace

SQL Statement

```
SELECT
  "DOCA" , SUM( "MA1" ) "MA1"
FROM
  /* Entity name: ZI_CDSVAB01 */ "ZICDSVAB01" "ZI_CDSVAB01"
GROUP BY
  "DOCA" LIMIT 100 WITH RANGE_RESTRICTION(' CURRENT' )
```

Execution Plan

Operation	Options	Object_name
PROJECT	ZICDSVAB01.DOCA, MA1	
LIMIT	NUM RECORDS: 100	
AGGREGATION	GROUPING: ZICDSVAB01.DOCA, AGGREGATION: SUM(ZICDSVAB01.MA1)	
INDEX JOIN (LEFT OUTER)	INDEX JOIN CONDITION: _VA.DOCA = =A0.DOCA	ZDOCA
COLUMN TABLE		ZDOCB

Explain End

Case4 association [1..*]

@EndUserText.label: 'TEST'

```
define view ZI_CDSVAB01 as select from zdoca as _VA
association [1..*] to zdocb as _VB on _VA.doca = _VB.doca
{
  _VA.doca,
  _VA.ca,
```

Executing SELECT statement:

Values are duplicated.

The screenshot shows the SAP CDS View execution interface. On the left, the SQL editor contains the following SELECT statement:

```
SELECT
  ZI_CDSVAB01~DOCA,
  sum( ZI_CDSVAB01~MA1 ) as MA1
FROM
  ZI_CDSVAB01
GROUP BY
  ZI_CDSVAB01~DOCA
```


On the right, the 'Raw Data' tab is active, displaying the results of the query. The results are shown in a table with two columns: 'DOCA' and 'MA1'. The table contains two rows of data, indicating that the values are duplicated.

DOCA	MA1
1200000010	260.000
1200000000	240.000



The interface also shows a 'Run' button, a 'Data Aging' tab, and a 'Max. Rows' dropdown set to 100. A status bar at the bottom right indicates '2 rows retrieved - 19 ms'.

Explain:

ZDOCA and ZDOCB are accessed.

Menu 

Display Execution Plan for SQL Statement

 Graphical Explain  Execution Trace

SQL Statement

```
SELECT
  "DOCA" , SUM( "MA1" ) "MA1"
FROM
  /* Entity name: ZI_CDSVAB01 */ "ZICDSVAB01" "ZI_CDSVAB01"
GROUP BY
  "DOCA" LIMIT 100 WITH RANGE_RESTRICTION(' CURRENT')
```

Execution Plan

Operation	Options	Object_name
PROJECT	ZICDSVAB01.DOCA, MA1	
LIMIT	NUM RECORDS: 100	
AGGREGATION	GROUPING: ZICDSVAB01.DOCA, AGGREGATION: SUM(ZICDSVAB01.MA1)	
INDEX JOIN (LEFT OUTER)	INDEX JOIN CONDITION: _VA.DOCA = =AO.DOCA	ZDOCA
COLUMN TABLE		ZDOCB

Explain End

Test 2 LEFT OUTER JOIN Test:

Case (go to [Result](#)):

In this test, tables are joined using LEFT OUTER JOIN instead of association. In case 1), ZDOCA is the source and ZDOCB is joined (ZI_CDSVAB02), and in case 3), LEFT OUTER TO ONE JOIN is used ((ZI_CDSVAB03). In case 3), ZDOCB is the source and ZDOCA is joined (ZI_CDSVAB04).

Source of 1) ZI_CDSVAB02:

```
@AbapCatalog.sqlViewName: 'ZICDSVAB02'

@AbapCatalog.compiler.compareFilter: true

@AccessControl.authorizationCheck: #CHECK

@EndUserText.label: 'TEST'

define view ZI_CDSVAB02 as select from zdoca as _VA

left outer join zdocb as _VB on _VA.doca = _VB.doca

{

_VA.doca,

_VB.docb,

_VA.ca,

_VB.cb,

@DefaultAggregation: #SUM

_VA.ma1,

@DefaultAggregation: #SUM

_VB.mb1
```



```
}
```

SELECT Statement for ZI_CDSVAB02:

```
SELECT  
  
ZI_CDSVAB02~DOCA,  
  
sum( ZI_CDSVAB02~MA1 ) as MA1  
  
FROM  
  
ZI_CDSVAB02  
  
GROUP BY  
  
ZI_CDSVAB02~DOCA
```

Source of 2) ZI_CDSVAB03:

```
@AbapCatalog.sqlViewName: 'ZICDSVAB03'  
  
@AbapCatalog.compiler.compareFilter: true  
  
@AccessControl.authorizationCheck: #CHECK  
  
@EndUserText.label: 'TEST'
```

```
define view ZI_CDSVAB03 as select from zdoca as _VA  
  
left outer to one join zdocb as _VB on _VA.doca = _VB.doca  
  
{  
  
_VA.doca,  
  
_VB.docb,  
  
_VA.ca,  
  
_VB.cb,  
  
@DefaultAggregation: #SUM  
  
_VA.ma1,  
  
@DefaultAggregation: #SUM  
  
_VB.mb1  
  
}
```

SELECT Statement for ZI_CDSVAB03:

```
SELECT  
  
ZI_CDSVAB03~DOCA,  
  
sum( ZI_CDSVAB03~MA1 ) as MA1  
  
FROM
```

```
ZI_CDSVAB03  
  
GROUP BY  
  
ZI_CDSVAB03~DOCA
```

Source of 3) ZI_CDSVAB04:

```
@AbapCatalog.sqlViewName: 'ZICDSVAB04'  
  
@AbapCatalog.compiler.compareFilter: true  
  
@AccessControl.authorizationCheck: #CHECK  
  
@EndUserText.label: 'TEST'  
  
  
define view ZI_CDSVAB04 as select from zdocb as _Vb  
  
left outer join zdoca as _Va on _Va.doca = _Vb.doca  
  
{  
  
  _Va.doca,  
  
  _Vb.docb,  
  
  _Va.ca,  
  
  _Vb.cb,  
  
  @DefaultAggregation: #SUM
```

```
_Va.ma1,  
  
@DefaultAggregation: #SUM  
  
_Vb.mb1  
  
}
```

SELECT Statement for ZI_CDSVAB04:

```
SELECT  
  
ZI_CDSVAB04~DOCB,  
  
sum( ZI_CDSVAB04~MB1 ) as MB1  
  
FROM  
  
ZI_CDSVAB04  
  
GROUP BY  
  
ZI_CDSVAB04~DOCB
```

Environment:

AS ABAP 7.52 FPS00

HANA Release 2.00.021.00.15




Results (go to [case](#)):

Case1 Source : Target = 1:N / LEFT OUTER JOIN

Executing SELECT statement:


JOIN happens even when no attributes from the joined table ZDOCB are included in SELECT statement.

Values are duplicated.



 Run	Data Aging	Max. Rows: 100	 Raw Data
<pre>SELECT ZI_CDSVAB02~DOCA, sum(ZI_CDSVAB02~MA1) as MA1 FROM ZI_CDSVAB02 GROUP BY ZI_CDSVAB02~DOCA</pre>			
Filter pattern  2 rows retrieved - 6 ms			
AB	DOCA	AB	MA1
	1200000010		260.000
	1200000000		240.000

Explain:

ZDOCA and ZDOCB are accessed.

Menu 

Display Execution Plan for SQL Statement

 Graphical Explain  Execution Trace

SQL Statement

```
SELECT
  "DOCA" , SUM( "MA1" ) "MA1"
FROM
  /* Entity name: ZI_CDSVAB02 */ "ZICDSVAB02" "ZI_CDSVAB02"
GROUP BY
  "DOCA" LIMIT 100 WITH RANGE_RESTRICTION(' CURRENT' )
```

Execution Plan

Operation	Options	Object_name
PROJECT	ZICDSVAB02.DOCA, MA1	
LIMIT	NUM RECORDS: 100	
AGGREGATION	GROUPING: ZICDSVAB02.DOCA, AGGREGATION: SUM(ZICDSVAB02.MA1)	
INDEX JOIN (LEFT OUTER)	INDEX JOIN CONDITION: _VA.DOCA = _VB.DOCA	ZDOCA
COLUMN TABLE		ZDOCB

Explain End

Case2 Source : Target = 1:N / LEFT OUTER TO ONE JOIN

Executing SELECT statement:

JOIN does NOT happen when no attributes from the joined table ZDOCB are included in SELECT statement.

Values are not duplicated.

Run Data Aging Max. Rows: 100

```
SELECT
  ZI_CDSVAB03~DOCA,
  sum( ZI_CDSVAB03~MA1 ) as MA1
FROM
  ZI_CDSVAB03
GROUP BY
  ZI_CDSVAB03~DOCA
```

Raw Data

Filter pattern 2 rows

AB DOCA	AB MA1
1200000000	120.000
1200000010	130.000

Explain:

Only ZDOCA is accessed.

Display Execution Plan for SQL Statement
[Graphical Explain](#)
[Execution Trace](#)
SQL Statement

```

SELECT
  "DOCA" , SUM ( "MA1" ) "MA1"
FROM
  /* Entity name: ZI_CDSVAB03 */ "ZICDSVAB03" "ZI_CDSVAB03"
GROUP BY
  "DOCA" LIMIT 100 WITH RANGE_RESTRICTION(' CURRENT' )

```

Execution Plan

Operation	Options	Object_name
PROJECT	ZICDSVAB03.DOCA, MA1	
LIMIT	NUM RECORDS: 100	
COLUMN TABLE		ZDOCA

Explain End

See [SAP Help](#) about LEFT OUTER TO ONE in detail.

Case3 Source : Target = N:1 / LEFT OUTER JOIN

Executing SELECT statement:

JOIN does NOT happen when no attributes are used from the joined table ZDOCA in SELECT statement.

Run Data Aging Max. Rows: 100 Raw Data



SELECT
 ZI_CDSVAB04~DOCB,
 sum(ZI_CDSVAB04~MB1) as MB1
FROM
 ZI_CDSVAB04
GROUP BY
 ZI_CDSVAB04~DOCB

Filter pattern 4 rows retrieved - 12 ms

AB	DOCB	AB	MB1
	2200000030		280.000
	2200000020		270.000
	2200000010		260.000
	2200000000		250.000

Explain:

Only ZDOCB is accessed.

Display Execution Plan for SQL Statement
 Graphical Explain
  Execution Trace
SQL Statement

```

SELECT
  "DOCB" , SUM( "MB1" ) "MB1"
FROM
  /* Entity name: ZI_CDSVAB04 */ "ZICDSVAB04" "ZI_CDSVAB04"
GROUP BY
  "DOCB" LIMIT 100 WITH RANGE_RESTRICTION(' CURRENT' )
  
```

Execution Plan

Operation	Options	Object_name
PROJECT	ZICDSVAB04.DOCB, MB1	
LIMIT	NUM RECORDS: 100	
AGGREGATION	GROUPING: ZICDSVAB04.DOCB, AGGREGATION: SUM(ZICDSVAB04.MB1)	
COLUMN TABLE		ZDOCB

Explain End

Others:

When cardinality is [n..*], it is not possible to use a field in the associated view (or table) in WHERE condition.

```

define view ZI_CDSVAB04 as select from zdoca as _VA
association [1..*] to ZI_CDSVAB as _VB on _VA.doca = _VB.doca
{
  _VA.doca,
  _VA.ca,
  _VB.docb,
  _VB.cb,
  @DefaultAggregation: #SUM
  _VA.ma1,
  @DefaultAggregation: #SUM
  _VB.mb1,

  --expose
  _VB
}
where _VB.cb = 'B10'
|

```

And it is not possible to add a field of associated view (or table) in Extend View.

```

> @AbapCatalog.sqlViewAppendName: 'ZEPURDOC'
@EndUserText.label: 'PurchasingDocument Extend'
extend view I_PurchasingDocument with ZE_PURCHASINGDOCUMENT
  association [0..*] to I_CalendarDate as _CalendarDate
  on $projection.PurchasingDocumentOrderDate = _CalendarDate.CalendarDate

{
  _CalendarDate.YearMonth,

  --expose
  _CalendarDate
}

```

Thanks for reading the blog!



Alert Moderator

Assigned Tags

ABAP Development

ABAP Extensibility

SAP S/4HANA

SAP S/4HANA Cloud, private edition

ABAP CDS view

embedded analytics

s/4 embedded analytics

[View more...](#) >

Similar Blog Posts

[Part#8. CDS Views - Joins and Associations](#)

By Sanjeev Kumar Oct 30, 2019

[New Cardinality Syntax for Performance Optimization in ABAP CDS and ABAP SQL](#)

By Andrea Schlotthauer May 10, 2023

[ABAP Core Data Services: New syntax for extending CDS entities](#)

By Andrea Schlotthauer Feb 17, 2022

Related Questions

[Fiori Elements: Filter based on association property](#)

By Sascha Weidlich Feb 21, 2019

Long runtime in CDS View using ASSOCIATION

By Enno Wulff Apr 06, 2023

cardinality in association in cds

By rajnish kumar Dec 17, 2020

21 Comments

You must be [Logged on](#) to comment or reply to a post.



Subhendu Ghanty

March 8, 2019 at 5:56 am

Very detailed and well explained article [Masaaki Arai](#) .

Is it possible to define right outer join using Association just by changing Cardinality ?

Like 0 | Share



Kaifa IS

August 5, 2020 at 9:44 am

HAVE U TRY ?WHAT'S THE CONCLUSION?PLS TELL ME

Like 0 | Share

**Jitendra Soni**

April 4, 2019 at 3:13 pm

Hello [Masaaki Arai](#),

Thanks for informative blog.

Could you please let me know what is the reason behind always using Join when executing [0..*] or [1..*] cardinality even no field is used in the select query from associated view?

Like 0 | Share

**Masaaki Arai** | Blog Post Author 

April 6, 2019 at 7:03 am

Hi Jitendra,

In my understanding, [0..*] or [1..*] is more normal behavior, and [0..1] or [1..1] is exceptional. [0..1] or [1..1] means the developer declare only one records are there on the right side so the records in the left side should not be duplicated, so the system can ignore the right side and do it for the efficiency.

Thanks, Masa

Like 0 | Share

**Jitendra Soni**

April 8, 2019 at 1:09 pm

Thanks [Masaaki Arai](#)

As per my understanding [0..*] or [1..*] would be normal if any field is used in selection list or in where clause from associated view

But even when no field is used from associated view and still join triggers, that i could not understand.

Like 0 | Share



Masaaki Arai | Blog Post Author 

April 10, 2019 at 11:05 am

Hi Jitendra,

If relationship between left and right is 1:n, even when no fields are selected, the record might have to be multiplied intentionally, so Join has to be happening. However, when using [1..1], as the developer declare the relationship should not be 1:n, the system can skip the join process.

Thanks, Masa

Like 1 | Share



Jitendra Soni

April 11, 2019 at 8:55 am

Thank you [Masaaki Arai](#)

Like 0 | Share



Pavlo Astashonok

October 26, 2019 at 9:35 pm

Too many typos, which makes it hard to understand the essence. In Test 2 description it should be

and in case 2), LEFT OUTER TO ONE JOIN is used

instead of

and in case 3), *LEFT OUTER TO NE JOIN* is used

Also it is better to place Test 2 definitions right before the test results, it will be more handy to compare them.

Like 1 | Share



Masaaki Arai | Blog Post Author 

October 29, 2019 at 8:58 am

Thanks, Pavel.

Corrected and try improving.

Like 2 | Share



Olivier Monteau

November 28, 2019 at 1:36 pm

Nice Blog!

Like 0 | Share



Jay Malla

January 29, 2020 at 2:58 pm

Hi [Masaaki Arai](#),

Nice article. Is there a way to search on an association with [1...*]. I have a scenario where we have a ListReport object page template. The ListReport has search for sales orders and the object page shows the line items. We want to have a search on the ListReport that searches through the line items for material number but just shows the sales orders.

So if we have

VBAK (sales order header table – where most of the List report searches are except for the material search)

VBAP (sales order line item table) – we need to expose the field Matnr

I have an association from VBAK to_items to VBAP. But I cannot do a search VBAK with to_items/matnr because of the 1 to many cardinality.

You have mentioned this above:

- *When [0..*] or [1..*] is used in Association, a field of the associated view (or table) cannot be used in WHERE condition and cannot be added using Extend View either.*

We do not have table functions on our system so I am trying to flatten the materials at the line item level into one view that I can link as an association to the header as [0..1] – but I am having some issues with that. I was wondering if you have any experience with flatten child records to the parent level to get around this issue.

Thanks,

Jay

Like 0 | Share

**Andrea Borgia**

April 21, 2021 at 4:26 pm

Masaaki Arai came across this blog looking for answers to [my question](#)

Have you ever found a case where the same CDS views would behave differently between HANA and earlier systems?

Like 0 | Share

**Masaaki Arai** | Blog Post Author 

April 21, 2021 at 6:47 pm

Hi Andrea,

Sorry, I have no experience to create CDS View on non-HANA DB.

I think SQL Trace (Tcd ST05) would help.

regards, Masa

Like 0 | Share

**Andrea Borgia**

April 23, 2021 at 2:21 pm

The limited information ST05 gives me is the same for both cardinalities on the older system.

I can't get more information because of this error:

[ASE Error SQL10331][SAP][ASE ODBC Driver][Adaptive Server Enterprise]Permission denied, database NPL, owner dbo.
You need the following permission(s) to run this command: SETUSER.

Thanks anyway.

Like 0 | Share



Masaaki Arai | Blog Post Author 

April 23, 2021 at 5:12 pm

Hello Andreas,

Sorry, I could not help enough.

But according to the error messages, it would be the permission issue in ASE layer. I found some SAP Notes with the term SETUSER, ASE.

I think it is efficient to ask help to ASE expert.

Thanks, Masa

Like 0 | Share



Andrea Borgia

April 24, 2021 at 8:44 am

Thanks, this is the [AS ABAP 7.52SP04 demo](#) from SAP, I guess I'll post a question shortly.

Like 0 | Share



Rohan Chavan

June 18, 2021 at 5:26 pm

Hi Masaaki,

My requirement is to add a Standard Field ANLA-IZWEK (Investment Reason) to the Asset History Sheet FIORI App.

ANLA-IZWEK Reason for Investment

• [Asset History Sheet \(F1615A\) - 5.8.5.8](#)

Std Field required in Filter n Dimension of Asset History Sheet app

I have added quite a lot Custom Fields to FIORI Apps using CFL, however, am clueless of how to add a Standard Field to filter Prompt & Dimensions of a Standard FIORI App.

Would really be obliged if you help throw some light on this.

Also, huge thanks for your very informative blog. Kudos!

Kind Regards,
Nida Patankar

Like 0 | Share



Masaaki Arai | Blog Post Author 

June 18, 2021 at 7:28 pm

Hi Nida,

You can associate ANLA to the source of Asset History with Extend View in the source of Asset balance.

<https://blogs.sap.com/2018/08/22/sample-custom-cds-view-using-predefined-virtual-data-model-and-extend-view-in-s4hana/>

or you can create wrapping view on top of the source of asset history VDM and associate ANLA in it.

<https://blogs.sap.com/2020/09/24/analytics-on-universal-journal-the-heart-of-sap-s-4hana/#How%20to%20add%20fields%20to%20VDMs?>

Thanks, Masa

Like 0 | Share

**Takayuki Sawada**

June 19, 2021 at 7:05 am

詳しい解説ありがとうございます。

本テーマと直接関係ないのですが、CDS Viewに関する質問です。

アドオンで作ったCDS ViewにFioriなどではなくSAP GUIからアクセスする

場合、一番ベストなやり方は何になるでしょうか？

SAPクエリで試してみたのですが、SAPクエリでインフォセットにCDS Viewを

割り当てた場合、アクセスコントロール(DCL)が効きませんでした。

ABAPでCDS Viewをアクセスする記述をすれば、アクセスコントロールが使われる

ことが確認できましたが、アドオンを各CDSビューごとに作ることは避けて今までの

クエリ的なツールでGUIも対応したいと考えています。

アドバイス頂けると幸いです。

Like 0 | Share

**Masaaki Arai** | Blog Post Author 

November 24, 2022 at 4:57 am

Sawada-san

CDS ViewはABAPプログラム内でSELECT文でアクセスが可能です。

SAPクエリでは、ご指摘の通り、SQL Viewからのアクセスとなり、アクセスコントロールはききません。

ベンダ的回答で恐縮ですが、SAPとしては、S4HANAのレポートは、Fioriベースのアプリで実現頂く事を推奨しています。例えば以下があります。

<https://launchpad.support.sap.com/#/notes/2579584>

(言い方を変えると、その方向で開発投資が行われている、という事になります。)

色々な状況があるかと思いますが、私個人もそれを推奨したいと思います。

SAPGUIメニューをFiori Launchpadで利用する事については、以下をご確認下さい。

<https://launchpad.support.sap.com/#/notes/2441101>

Like 0 | Share



Sneha Veerabhadrappa Dugani 

November 24, 2022 at 3:22 am

Awesome blog . Very well explained. This helped me a lot . Thanks a Million Masaaki Arai.

Like 0 | Share

Find us on



Privacy	Terms of Use
Legal Disclosure	Copyright
Trademark	Cookie Preferences
Newsletter	Support