



FPT Software

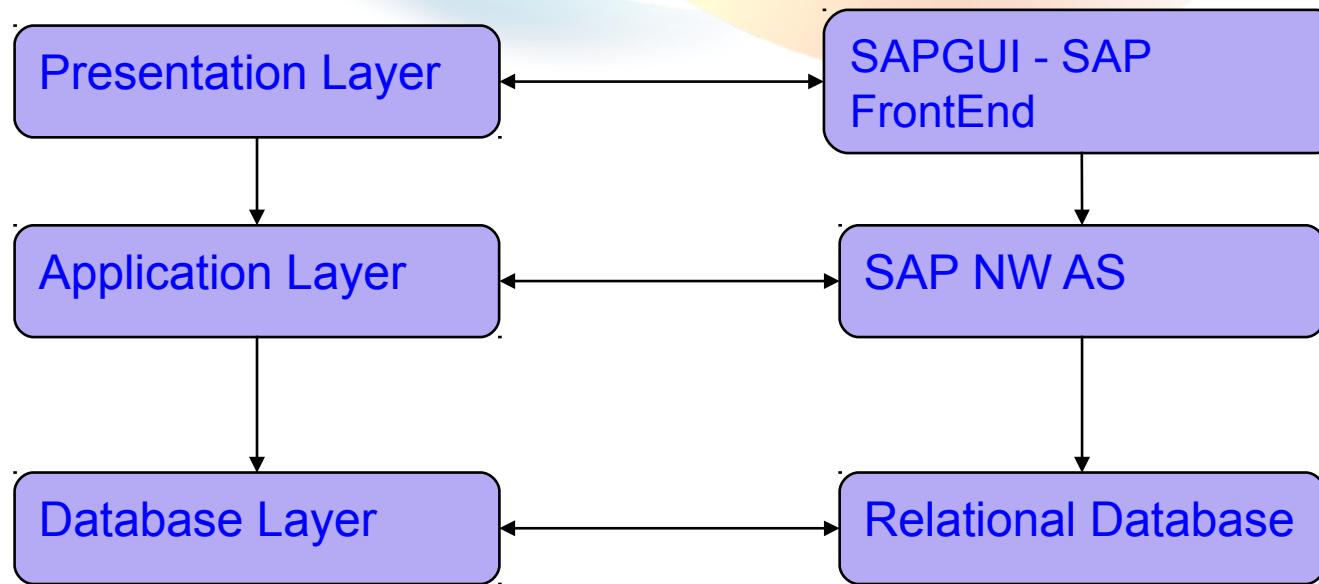
CDS View overview

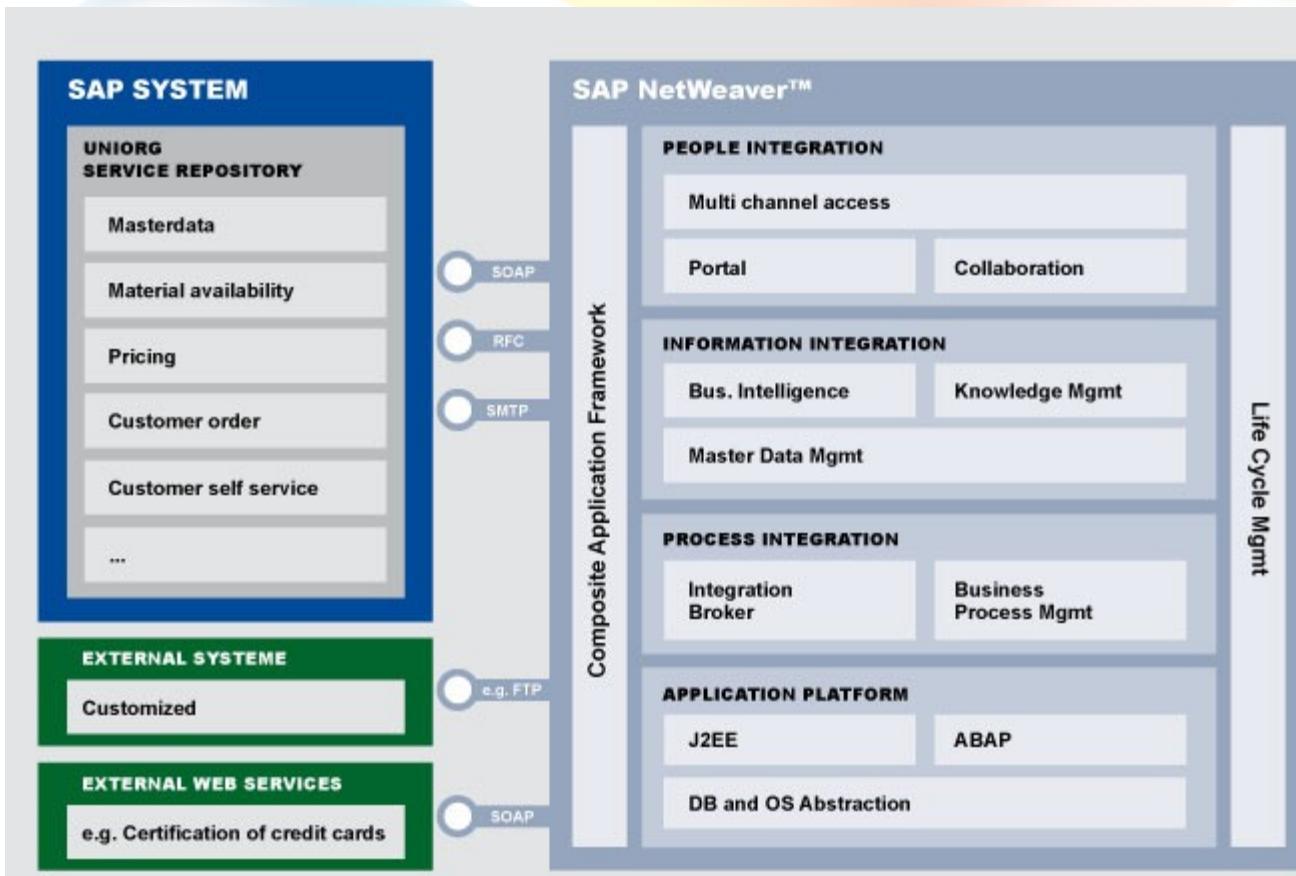
R&D Team

A large, stylized graphic element in the background features three overlapping, rounded rectangular shapes. The top shape is light green, the middle is light orange, and the bottom is light blue. They are positioned in a way that suggests depth, with the green shape at the top, the orange in the middle, and the blue at the bottom. A thin horizontal grey line runs across the bottom of the slide, intersecting the base of the blue shape.

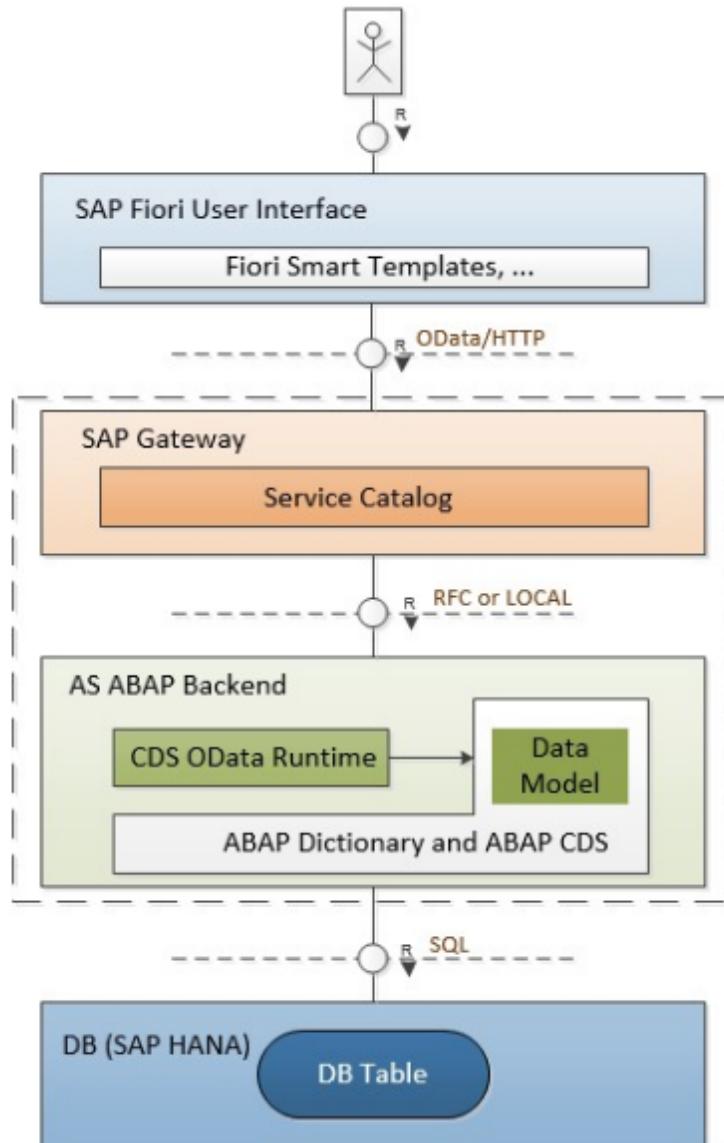
Report Date: 20.06.2018
Author: Nguyen Hoang Nam

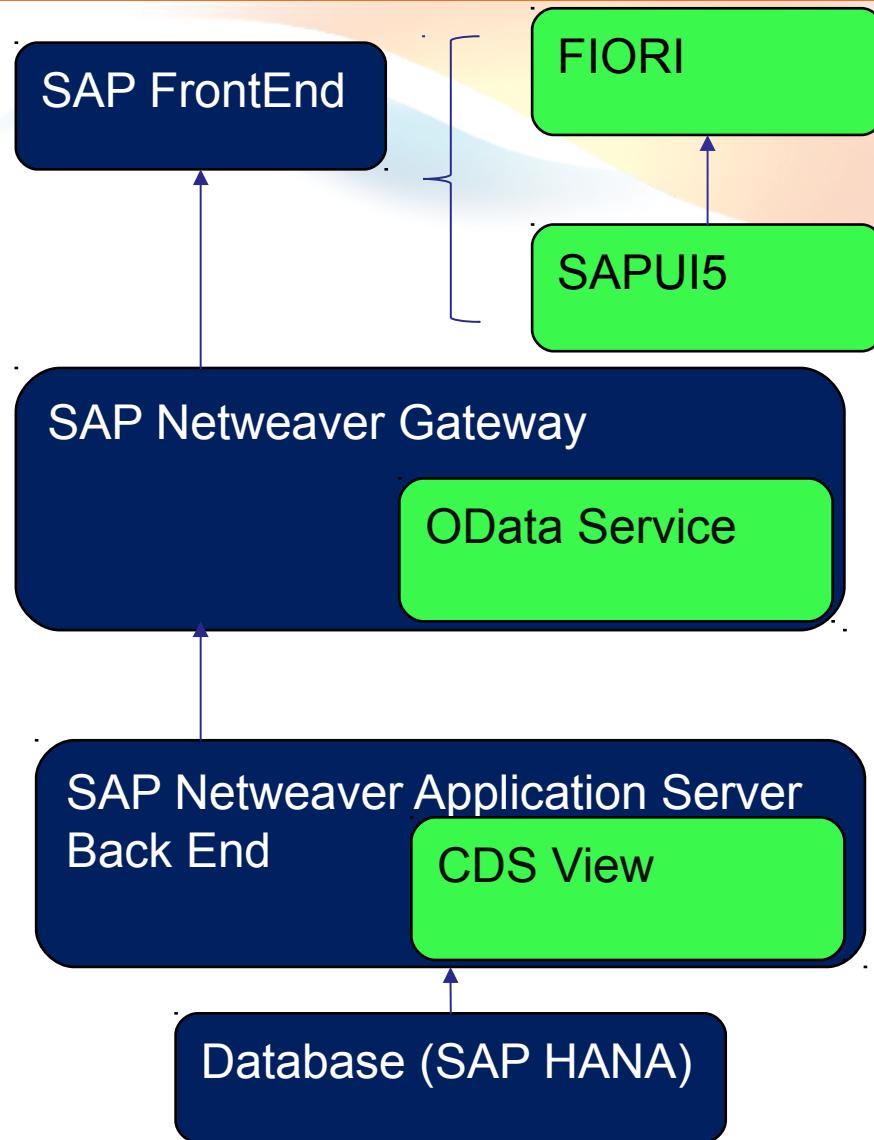
Internal use





Architecture of SAP Backend to FrontEnd



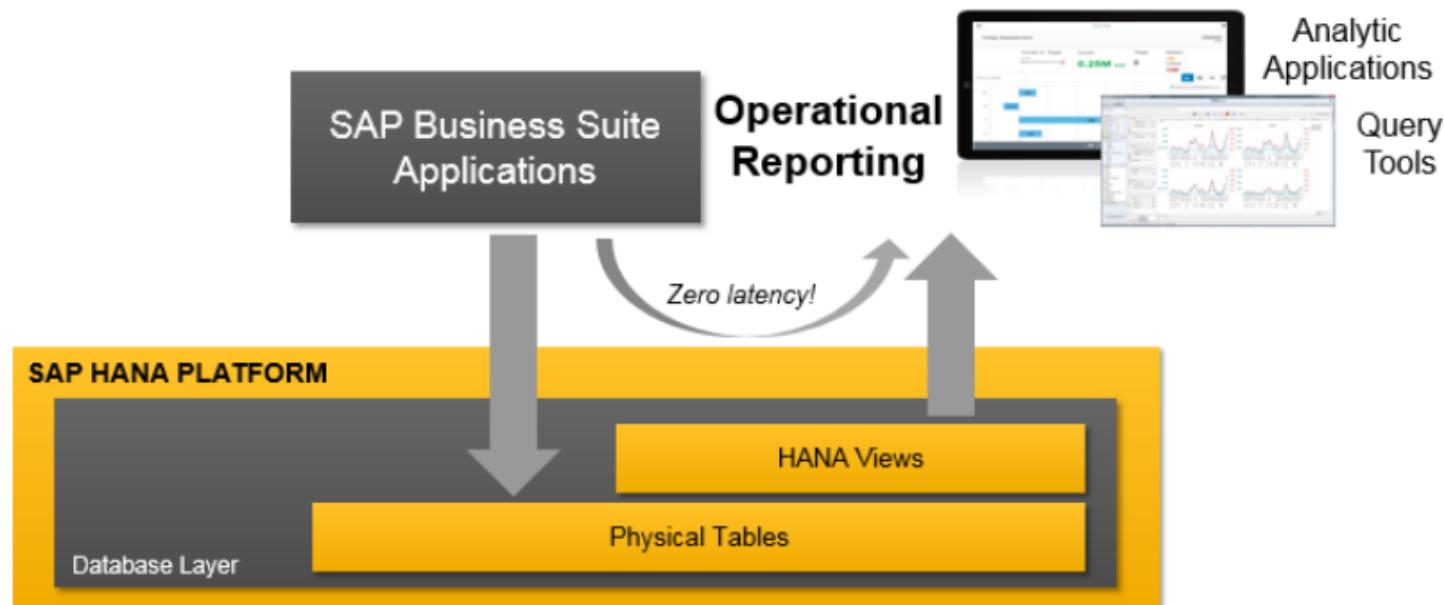


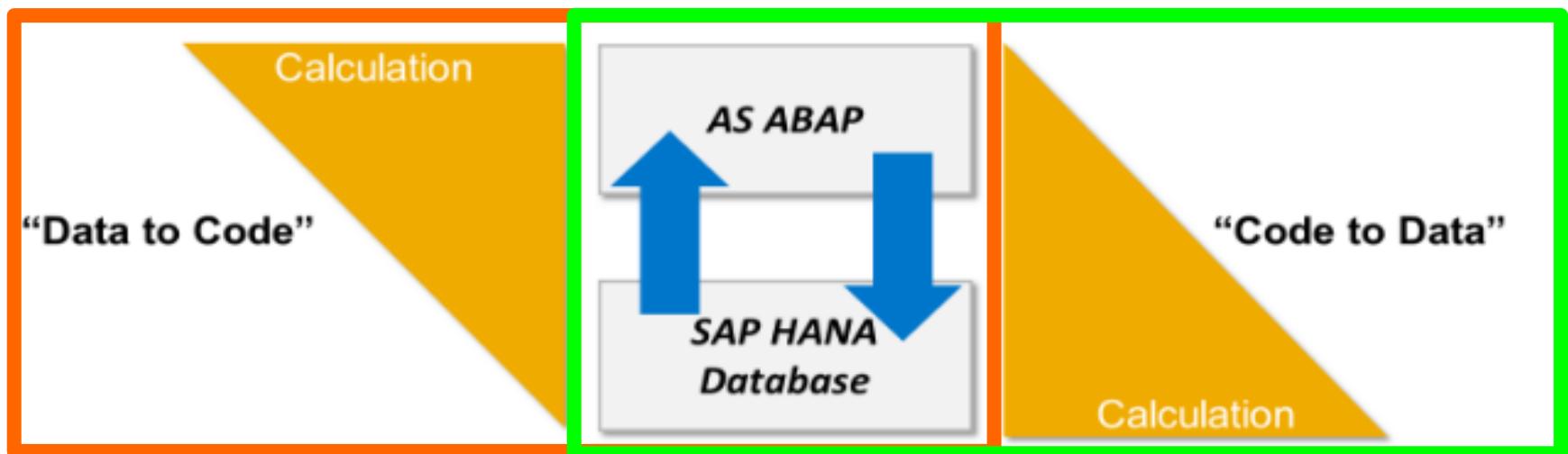
- THE RULE OF THUMB IS SIMPLE:

***“DO AS MUCH AS YOU CAN IN
THE DATABASE TO GET THE
BEST PERFORMANCE”.***

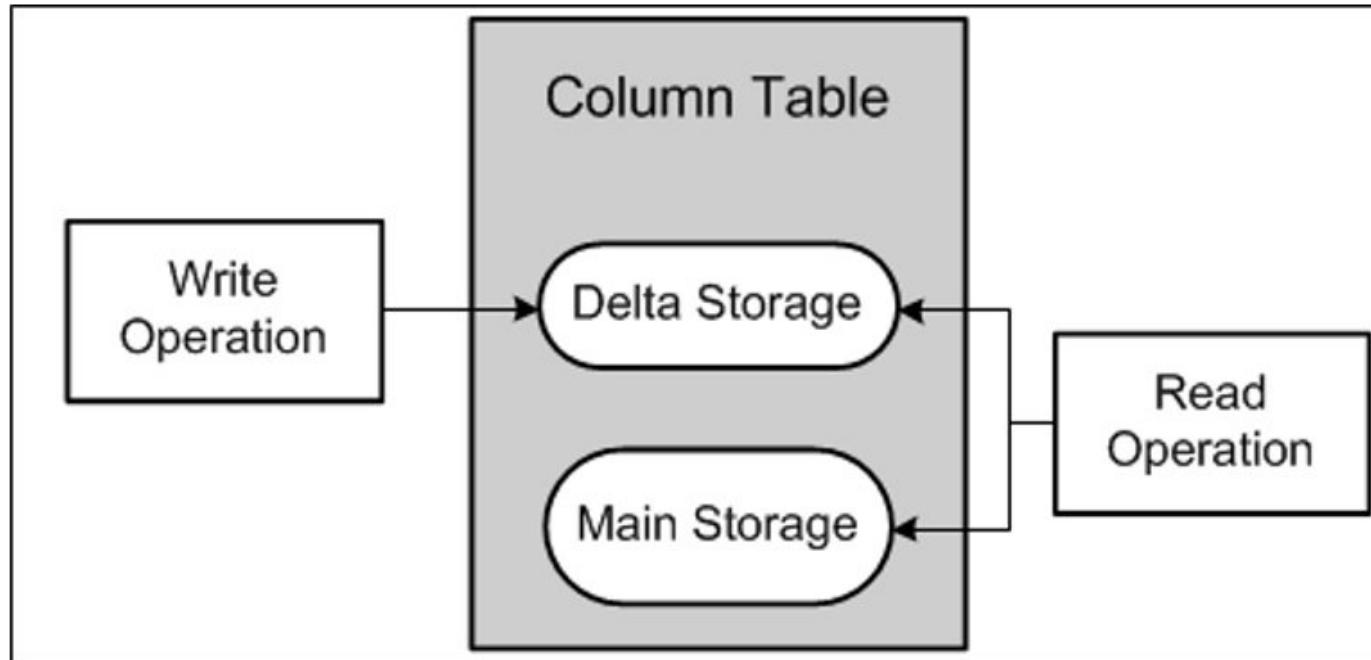
I. SAP HANA – New technology of SAP

- HANA – High-performance analytic appliance:
 - in-memory database technology
 - to process data stored in RAM as opposed to reading it from a disk





Row Store and Column Store



I. SAP HANA Technology

Traditional row database

ROW-BASED

<u>Name</u>	<u>Order</u>	<u>Material</u>
Tom	35	0123
Tom	47	0234
Matt	153	0234
Susan	278	1203
Susan	301	0123

SAP HANA database

Column-based

<u>Name</u>	<u>Order</u>	<u>Material</u>
Tom	35	0123
Tom	47	0234
Matt	153	0234
Susan	278	1203
Susan	301	0123

This data would be stored in a traditional database as follows:

001:Tom,35,0123;
002:Tom,47,0234;
003:Matt,153,0234;
004:Susan,278,1203;
005:Susan,301,0123;

In a column based system, the data would be stored as follows:

Tom:001, Tom:002, Matt:003, Susan:004, Susan:005;
35:001, 47:002, 153:003, 278:004, 301:005;
0123:001, 0234:002, 0234:003, 1203:004, 0123:005;

Convert our database table into two distinct tables:

Dictionary Data

ValueID	Value
1(00)	Tom
2(01)	Susan
3(10)	Matt
4(11)	(unused)

Attribute Vector :

Array Vector

Row	ValueID (2 bits)
1	00
2	00
3	10
4	01
5	01



What is CDS Views ?

Definition :

It is an infrastructure that can be used by database developers to create the underlying (persistent) data model which the application services expose to UI clients.

It is an SQL like language that provides an open-source Data Definition Language for the creation database artifacts

- Can combine the models in ABAP platform and very important component in current technology.
- The extension of SQL, allows optimal and extended use of the SAP HANA database.
- ABAP CDS and annotation technology can be used to model large scale information that can be optimally integrated into the interfaces for cloud applications or other user, since OData also uses a strongly model-based protocol.
- CDS View and programming model is the base of SAP Fiori applications.

DDL



Data Definition Language

Data modelling and retrieval on a higher semantic level

Extends native SQL means for higher productivity

QL



Query Language

Consume CDS entities via Open SQL in ABAP

Fully transparent SQL extensions

DCL



Data Control Language

Define authorizations for CDS views

Modelled and declarative approach

Integrates with classic authorization concepts

Built-in functions of CDS Views

Generic SQL Expressions

Simple- /
Searched-Case
Coalesce

Conversion Functions

Unit Conversion
Currency
Conversion
Cast
Hex2Bin
Bin2Hex
Decimal Shift

String Functions

Concat
Instr
Left / Right
Length
Lpad / Rpad
Ltrim / Rtrim
Replace

Arithmetic Functions

Abs
Ceil
Div (Integer)
Division (Dec)
Floor
Mod
Round

Date and Time Functions

Days between
Add Days
Add Month
Current Tmstp
Date is Valid
Tmstp is Valid
Seconds btw.
Tmstps
Add seconds



What is the function of CDS View?

- Like Database View in ABAP tcode SE11, but included more options :
 - Inner and outer joins
 - Associations
 - Union
 - Built-in functions
 - Annotation

Creating First CDS View

jon - 52.91.213.213 - Remote Desktop Connection

ABAP - zDemoABAP_Channel/WebContent/index.html - Eclipse

File Edit Source Navigate Search Project Run Window Help

Project Explorer

- A4H_001_bwdeveloper_en
- A4H_001_developer_en**
- zDemoABAP_Channel [System information unavailable]
- zDemoBOPF_Product [System information unavailable]

index.html

```

1 <!DOCTYPE HTML>
2 <html>
3   <head>
4     <meta http-equiv="X-UA-Compatible" content="IE=edge">
5     <meta http-equiv='Content-Type' content='text/html; charset=UTF-8'>
6
7     <script src="resources/sap-ui-core.js"
8           id="sap-ui-bootstrap"
9           data-sap-ui-libs="sap.m"
10          data-sap-ui-theme="sap_bluecrystal">
11   </script>
12   <!-- only load the mobile lib "sap.m" and the "sap_bluecryst
13
14   <script>
15     sap.ui.localResources("zappc_chat");
16     var app = new sap.m.App({initialPage:"idChat1"});
17     var page = sap.ui.view({id:"idchat1", viewName:"zappc_
18     app.addPage(page);
19     app.placeAt("content");
20   </script>
21
22   </head>
23   <body class="sapUiBody" role="application">
24     <div id="content"></div>
25   </body>
26 </html>

```

Feature Explorer

Welcome!

This interactive view allows you to explore the features most favored by the ABAP development team. These are features that will help you get stuff done faster! Interested?

Click here to begin!

Outline

- DOCTYPE:HTML
- html
 - head
 - meta http-equiv=X-UA-Compatible
 - meta http-equiv=Content-Type
 - script id=sap-ui-bootstrap
 - #comment
 - script
 - body class=sapUiBody

Properties

Property	Value
General	
accesskey	
async	
charset	
class	
contenteditable	
contextmenu	
data-sap-ui-libs	sap.m
data-sap-ui-theme	sap_bluecrystal
defer	
dir	
draggable	
dropzone	
event	
for	
hidden	



Creating First CDS View

workspace - ABAP - Eclipse

File Edit Navigate Search Project Run Window Help

Project Explorer

- > RS_BW_STRUPAK
- > S_ADDRESS_MANAGEMENT
- > S_APPLICATION_STRUCTURE
- > S_BUPA_SW
- > S_LCBN_OD
- > S_DMCM
- > S_NWDEM
- > S_QUERY_U
- > SA_COMM
- > SACMRT_L
- > SAJO
- > SAP_UI_RC
- > SAUP
- > SESH_SAP
- > SGW_BASI
- > SPAK_UNIT
- > SPAK_UNIT
- > SPRODUCT
- > SUI_BASIS
- > TEST
- > UACC_STR
- > UI2_ACH_S
- > VF_PRC_AN
- > VF_PRC_AN
- > WP_PI
- > ZDEMO_BI
- > ZDEMO_C
- ★ Add to Favorites
- > ZION_HANA_DEV

New

- ABAP Class
- ABAP Interface
- ABAP Program
- Other ABAP Repository Object

Open

Open in Project

Open With SAP GUI

Copy

Delete

Get Where-used List...

Export...

Refresh

F5

Activate

Run As

Debug As

Profile As

Coverage As

Add Bookmark...

Add to Favorites

Outline

An outline is not available.

Problems

Properties

Templates

Bookmarks

Task Repositories

Task List

Feed Reader

Transport Organizer

0 errors, 1 warning, 0 others

Description

> ⚠ Warnings (1 item)

Resource

Path

Location

Type

SAP HANA Administration Console

SAP HANA Development

SAP HANA Modeler

ABAP

Database Development

Feature Explorer

Welcome!

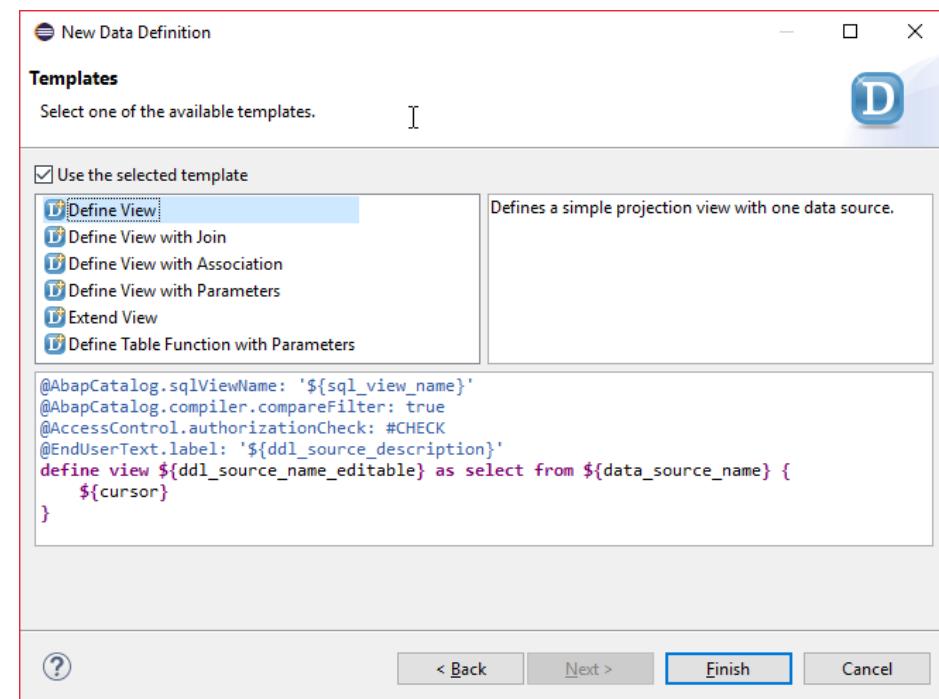
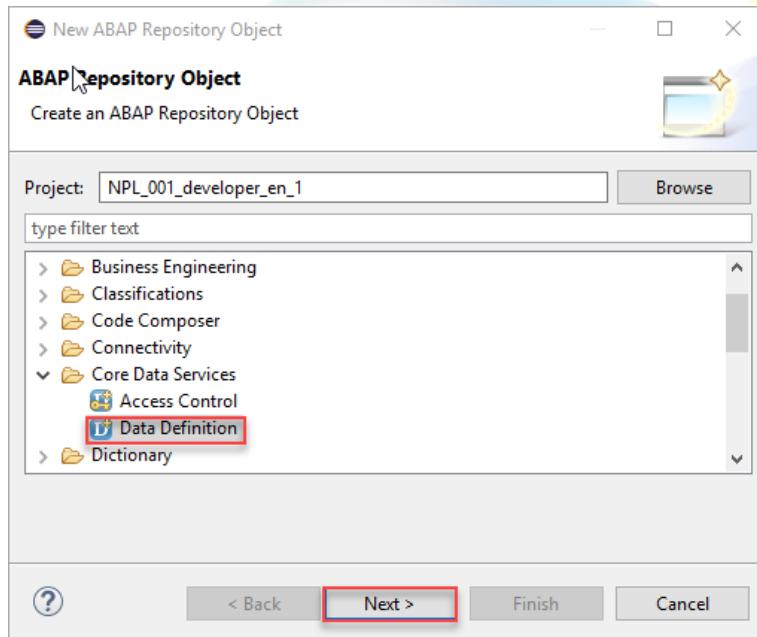
This interactive view allows you to explore the features most favored by the ABAP in Eclipse development team. These are features that will help you get stuff done faster! Interested?

Click here to begin!

tg

Samsung Will Love This iPhone 8 Rumor
Key feature just got stripped out.
tomsguide.onesignal.com

Creating First CDS View



```

[NPL] ZJON_CDS_VIEW_EXAMPLE [NPL] ZJON_CDS_VIEW_EXAMPLE
1 @AbapCatalog.sqlViewName: 'zjontestview1'
2 @AbapCatalog.compiler.compareFilter: true
3 @AccessControl.authorizationCheck:#NOT_REQUIRED
4 @EndUserText.label: 'Example ABAP CDS Views'
5 define view Zjon_Cds_View_Example as select from snwd_so {
6   //Sales order Header Table
7   key node_key as Node,
8   so_id as SalesOrder,
9   buyer_guid as buyer_id,
10  gross_amount,
11  net_amount,
12  tax_amount,
13  case billing_status
14  when 'P' then 'Paid'
15  when ' ' then 'Not Paid'
16  else '?' end as payment_status
17 }
18 
```

Opening Annotations

Datasource declarations

Field specification/declarations

SELECT conditions and restrictions

Opening Annotations – These declare high level data about the CDS view:

- @AbapCatalog.sqlViewName is the only mandatory annotation for a non-extending CDS View source file.
- @AccessControl.authorizationCheck specifies whether an authorization check should be performed for the current CDS view.

What happened

```

1@AbapCatalog.sqlViewName: 'zjontestview1'
2@AbapCatalog.compiler.compareFilter: true
3@AccessControl.authorizationCheck:#NOT_REQUIRED
4@EndUserText.label: 'Example ABAP CDS Views'
5define view Zjon_Cds_View_Example as select from snwd_so {
6  //Sales order Header Table
7  key node_key as Node,
8  so_id as SalesOrder,
9  buyer_guid as buyer_id,
10 gross_amount,
11 net_amount,
12 tax_amount,
13 case billing_status
14 when 'P' then 'Paid'
15 when '' then 'Not Paid'
16 else '?'
17 end as payment_status
18 }

```

Opening Annotations

Datasource declarations

Field specification/declarations

SELECT conditions and restrictions

- When you create an ABAP CDS View, a corresponding view is created on the HANA DB matching the CDS view created on the ABAP side.
- This view will have the SQL View name that you provided (after the `@AbapCatalog.sqlViewName` annotation) and will be generated automatically after you create the CDS view.
- The corresponding HANA view does not require a separate transport to move between HANA DBs. Once the view is transported and executed on SAP NetWeaver, the HANA View will be generated automatically in the corresponding HANA DB.
- You can view the create SQL view in Eclipse by using the SQL HANA Administrator Perspective (described on the next slide).

[NPL] ZJON_CDS_VIEW_EXAMPLE [NPL] ZJON_CDS_VIEW_EXAMPLE

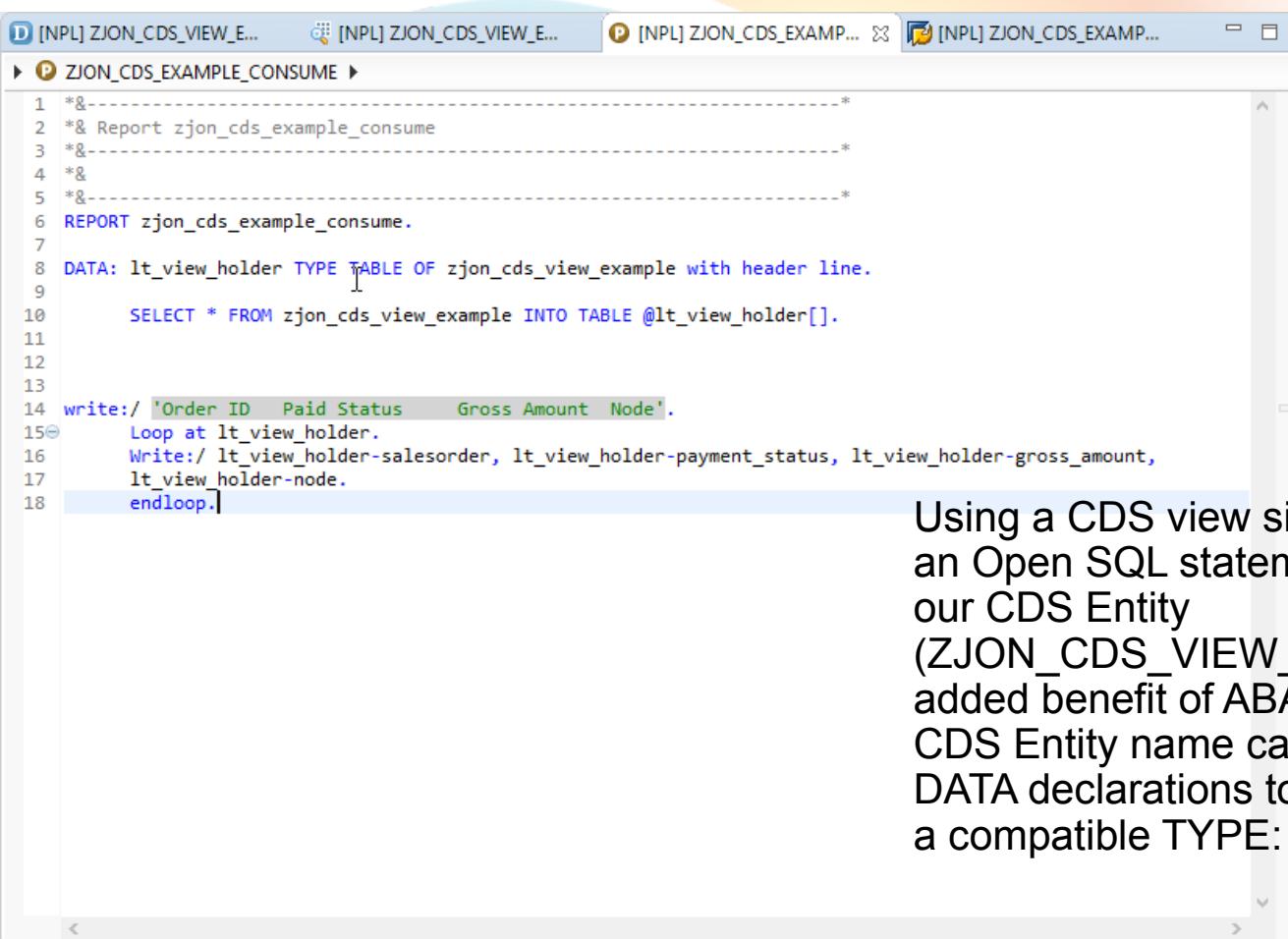
ZJON_CDS_VIEW_EXAMPLE ►

Raw Data | Show Log | Max. Rows: 1000

Filter pattern 701 rows retrieved - 12 ms SQL Console | Number of Entries | Add filter

AB	Node	AB	SalesOrder	AB	buyer_id	AB	gross_amount	AB	net_amount	AB	tax_amount	AB	payment_status
1244D0D392...	05000000000	1244D0D392B...	14385.85	12088.95	2296.90	Paid							
1244D0D392...	05000000001	1244D0D392B...	15117.76	12704.00	2413.76	Paid							
1244D0D392...	05000000002	1244D0D392B...	5631.08	4732.00	899.08	Paid							
1244D0D392...	05000000003	1244D0D392B...	1704.04	1431.97	272.07	Paid							
1244D0D392...	05000000004	1244D0D392B...	761.24	639.70	121.54	Paid							
1244D0D392...	05000000005	1244D0D392B...	101299.22	85125.40	16173.82	Paid							
1244D0D392...	05000000006	1244D0D392B...	250.73	210.70	40.03	Paid							
1244D0D392...	05000000007	1244D0D392B...	9715.16	8164.00	1551.16	Paid							
1244D0D392...	05000000008	1244D0D392B...	195.16	164.00	31.16	Paid							
1244D0D392...	05000000009	1244D0D392B...	3972.22	3338.00	634.22	Paid							
1244D0D392...	05000000010	1244D0D392B...	827.95	695.75	132.20	Not Paid							
1244D0D392...	05000000011	1244D0D392B...	325.94	273.90	52.04	Not Paid							
1244D0D392...	05000000012	1244D0D392B...	12704.40	10675.96	2028.44	Not Paid							
1244D0D392...	05000000013	1244D0D392B...	8996.40	7560.00	1436.40	Not Paid							
1244D0D392...	05000000014	1244D0D392B...	3459.33	2907.00	552.33	Not Paid							
1244D0D392...	05000000015	1244D0D392B...	862.73	724.98	137.75	Not Paid							
1244D0D392...	05000000016	1244D0D392B...	70.18	58.97	11.21	Not Paid							
1244D0D392...	05000000017	1244D0D392B...	178.14	149.70	28.44	Not Paid							
1244D0D392...	05000000018	1244D0D392B...	871.55	732.40	139.15	Not Paid							
1244D0D392...	05000000019	1244D0D392B...	1444.64	1213.99	230.65	Not Paid							
1244D0D392...	05000000020	1244D0D392B...	5357.97	4502.50	855.47	Not Paid							
1244D0D392...	05000000021	1244D0D392B...	158.98	133.60	25.38	Not Paid							
1244D0D392...	05000000022	1244D0D392B...	521.22	438.00	83.22	Not Paid							
1244D0D392...	05000000023	1244D0D392B...	411.50	345.80	65.70	Not Paid							
1244D0D392...	05000000024	1244D0D392B...	2225.22	2057.00	156.22	Not Paid							

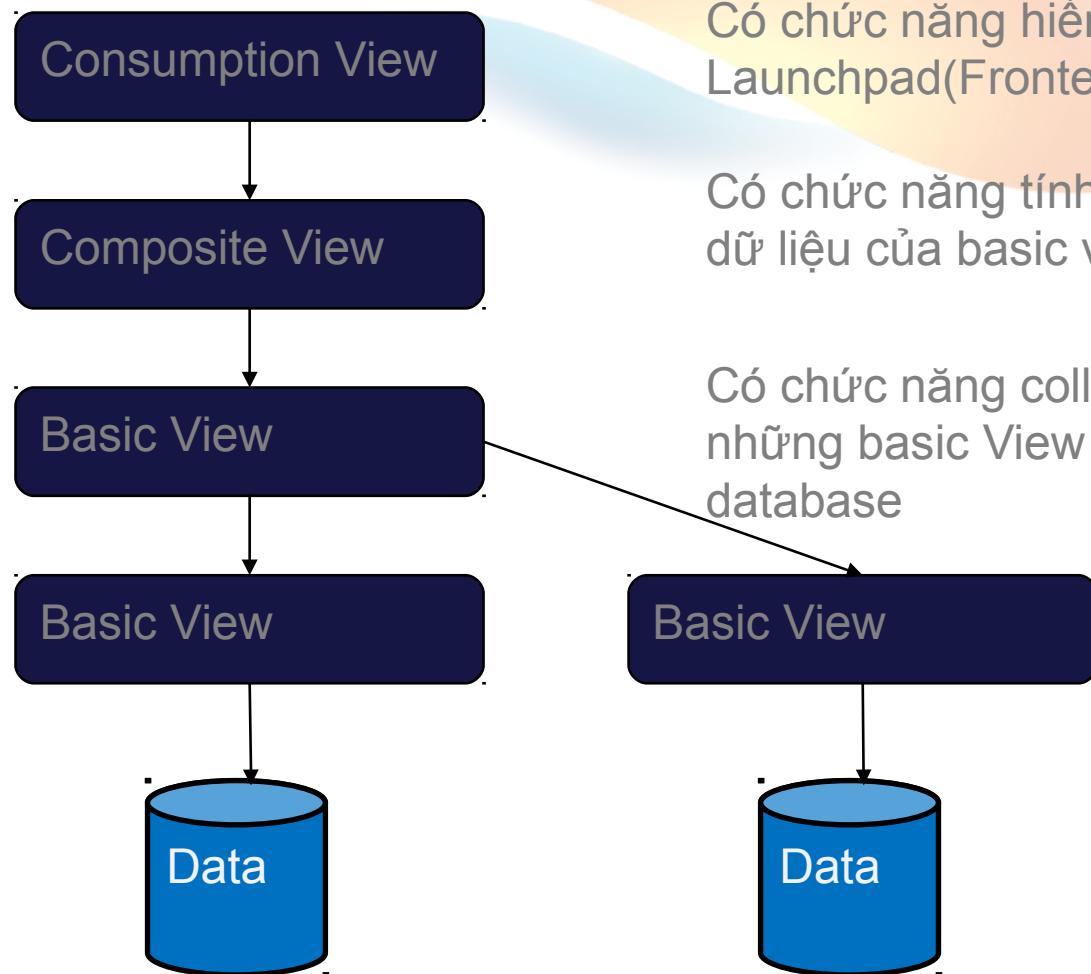
- Name space of CDS Model
- Definition of DDLS-Name
- Name after first transport.



The screenshot shows an SAP ABAP editor window with several tabs at the top. The active tab is titled 'P ZION_CDS_EXAMPLE_CONSUME'. The code in the editor is as follows:

```
1 *-<*>
2 *& Report zjon_cds_example_consume
3 *&-
4 *&
5 *&-
6 REPORT zjon_cds_example_consume.
7
8 DATA: lt_view_holder TYPE TABLE OF zjon_cds_view_example WITH HEADER LINE.
9
10    SELECT * FROM zjon_cds_view_example INTO TABLE @lt_view_holder[].
11
12
13
14 write:/ 'Order ID      Paid Status      Gross Amount      Node'.
15@ Loop at lt_view_holder.
16@   Write:/ lt_view_holder-salesorder, lt_view_holder-payment_status, lt_view_holder-gross_amount,
17@         lt_view_holder-node.
18 endloop.
```

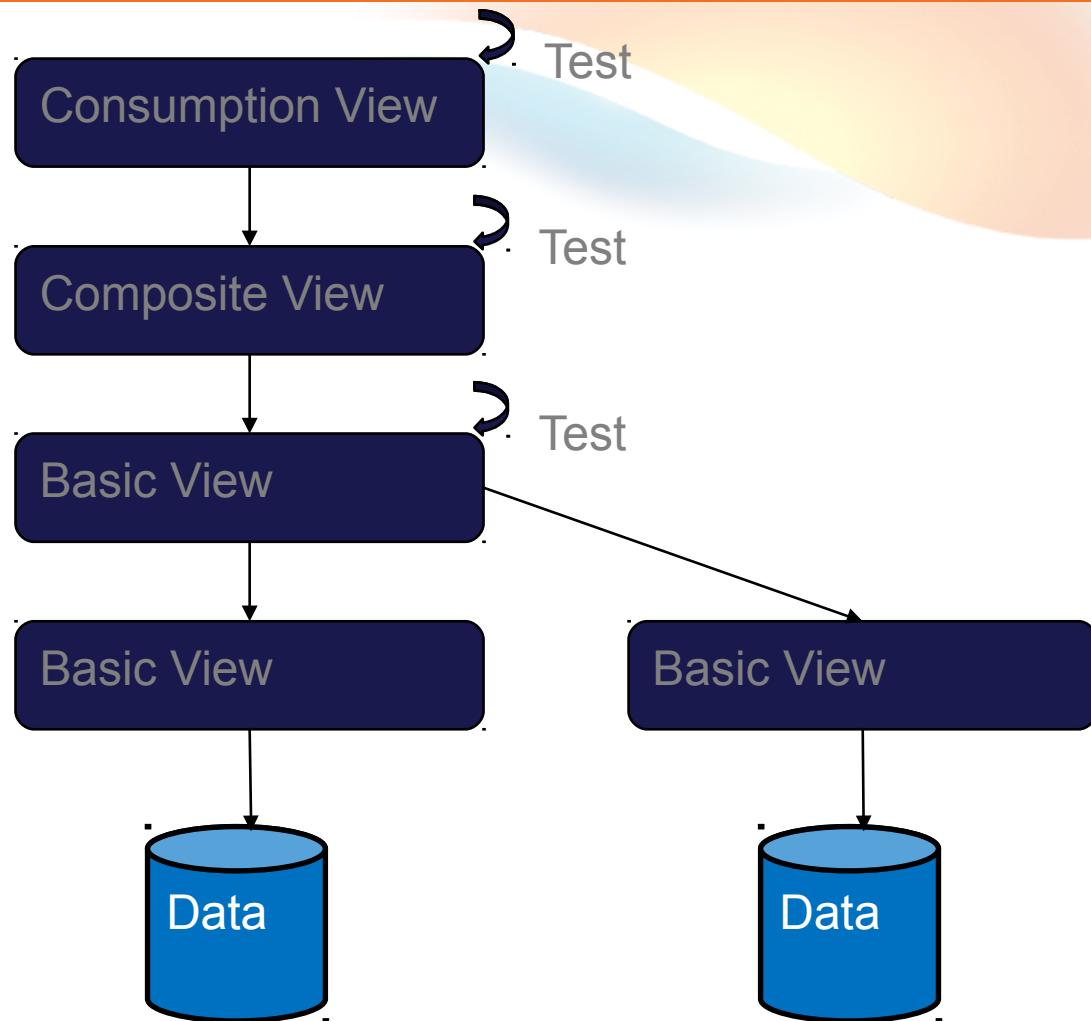
Using a CDS view simply requires using an Open SQL statement that selects from our CDS Entity (ZJON_CDS_VIEW_EXAMPLE). As an added benefit of ABAP CDS, a generated CDS Entity name can also be used in DATA declarations to create structures of a compatible TYPE:

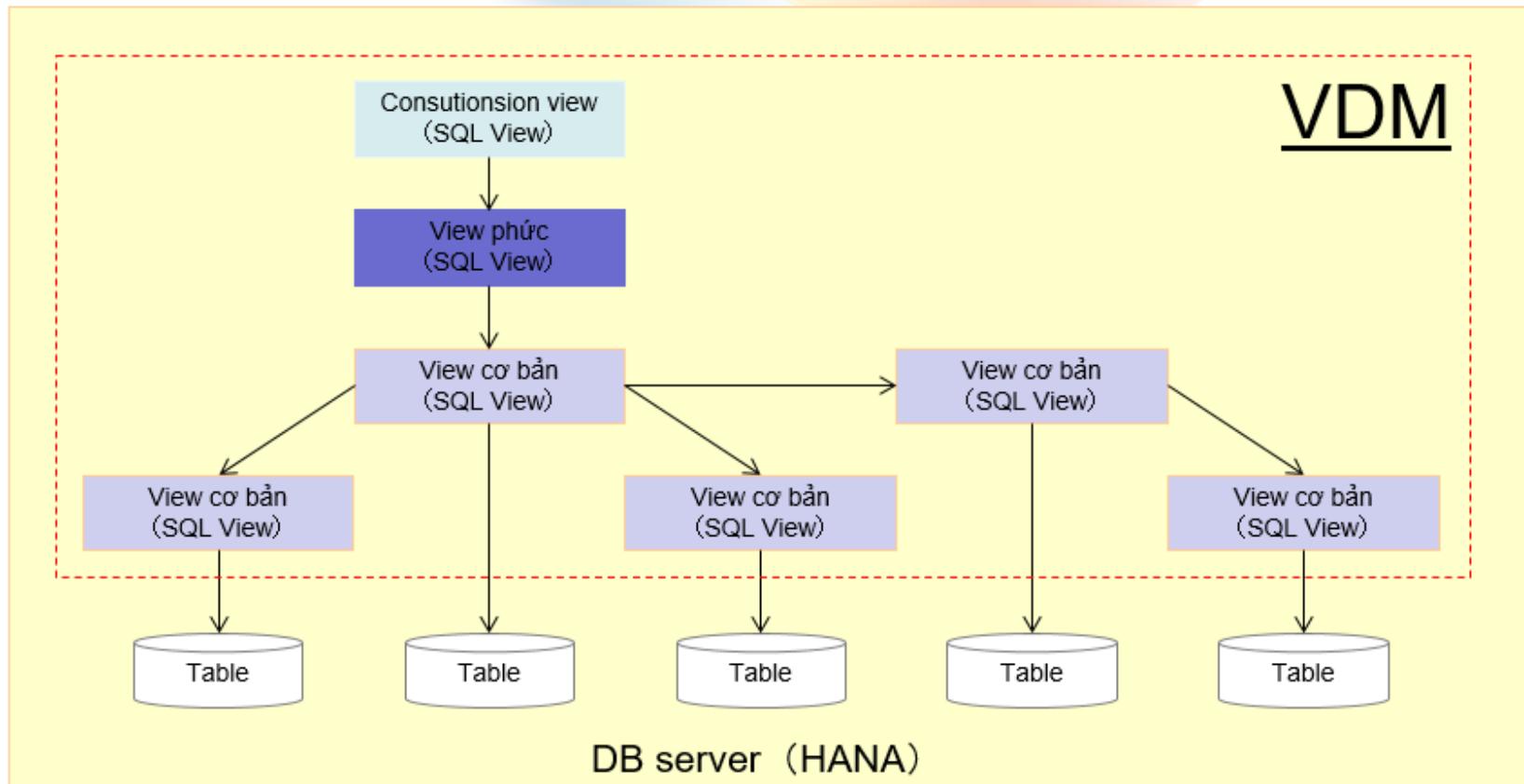


Có chức năng hiển thị lên trên Fiori Launchpad(Frontend)

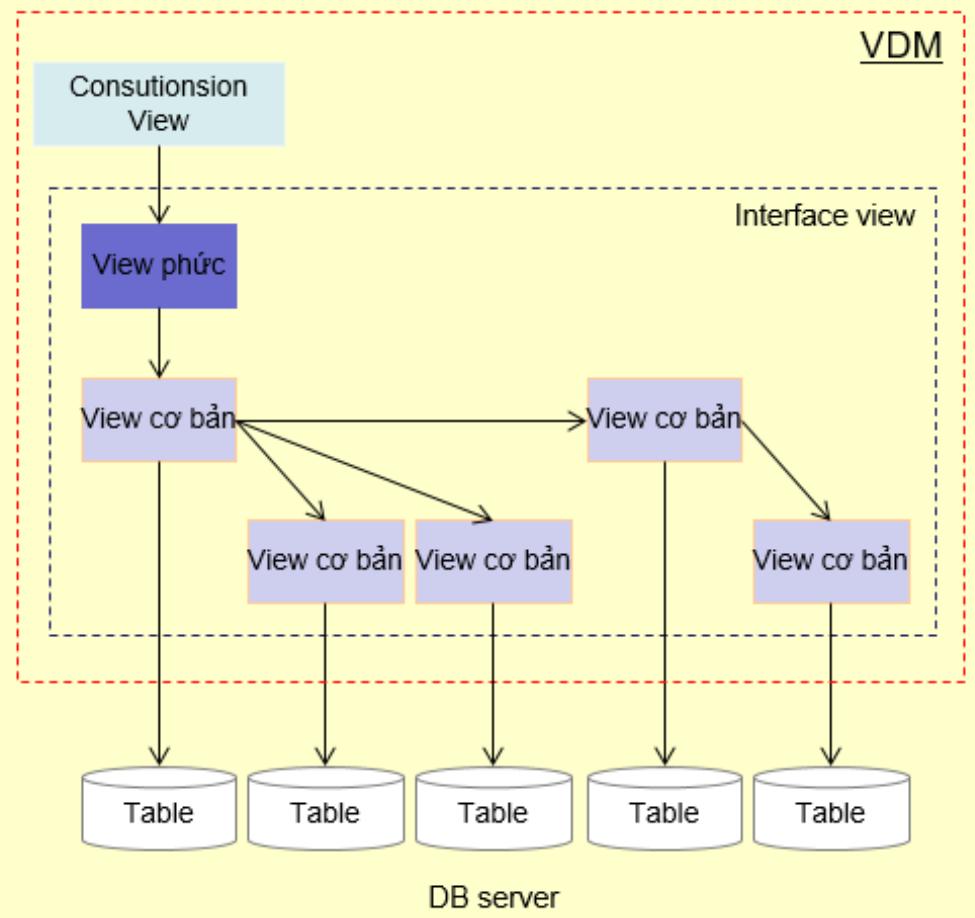
Có chức năng tính toán dựa trên dữ liệu của basic view

Có chức năng collect dữ liệu từ những basic View khác hoặc database





Structure of a CDS View program



Phân loại CDS View (SQL View)

Consutionsion view

View được sử dụng từ application phân tích
Mô tả cách tính toán cho các dữ liệu đã collect là chủ yếu.

Interface view(Tên 2 loại CDSView)

View collect/ tính toán dữ liệu theo trích xuất dữ liệu trực tiếp từ database, trích xuất dữ liệu đã sử dụng CDS View khác

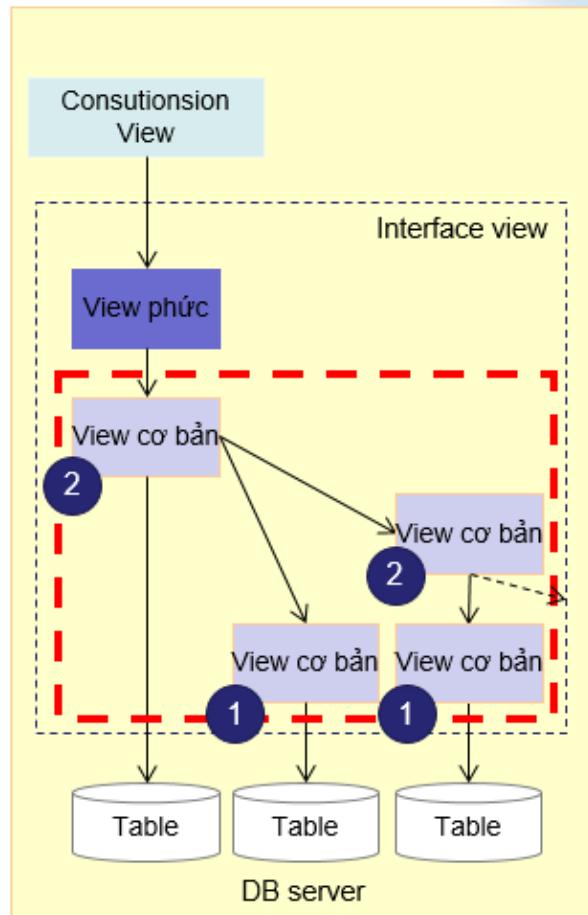
View phức

View trạng thái mới nhất khi sum data.
Định nghĩa phương pháp sum khi output.
View được sử dụng trực tiếp ở consutionsion view.

View cơ bản

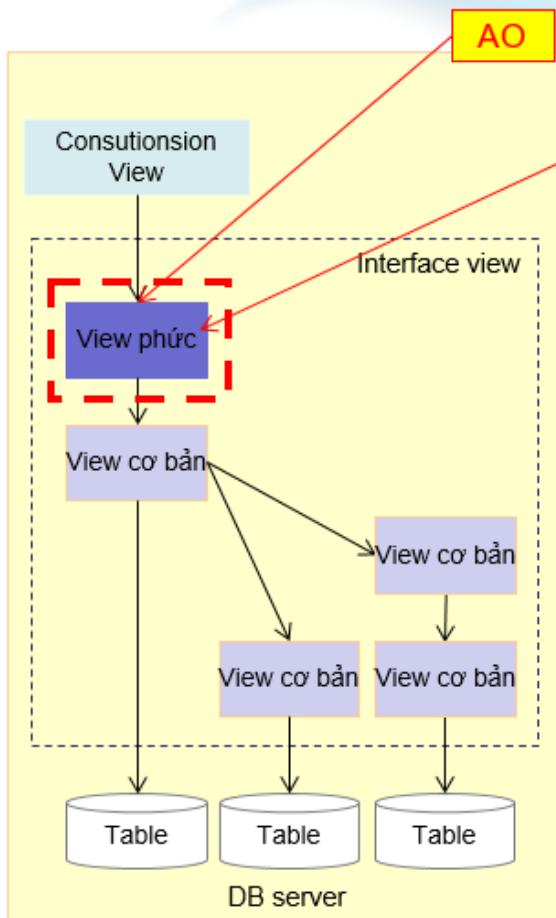
View collect data.

Tạo view cơ bản như là các phần riêng lẻ kết hợp chúng lại tạo mô hình dữ liệu ảo
View được sử dụng ở view phức hợp và các view cơ bản
View cơ bản truy cập trực tiếp vào database, và view liên kết (kết hợp) với các view cơ bản khác



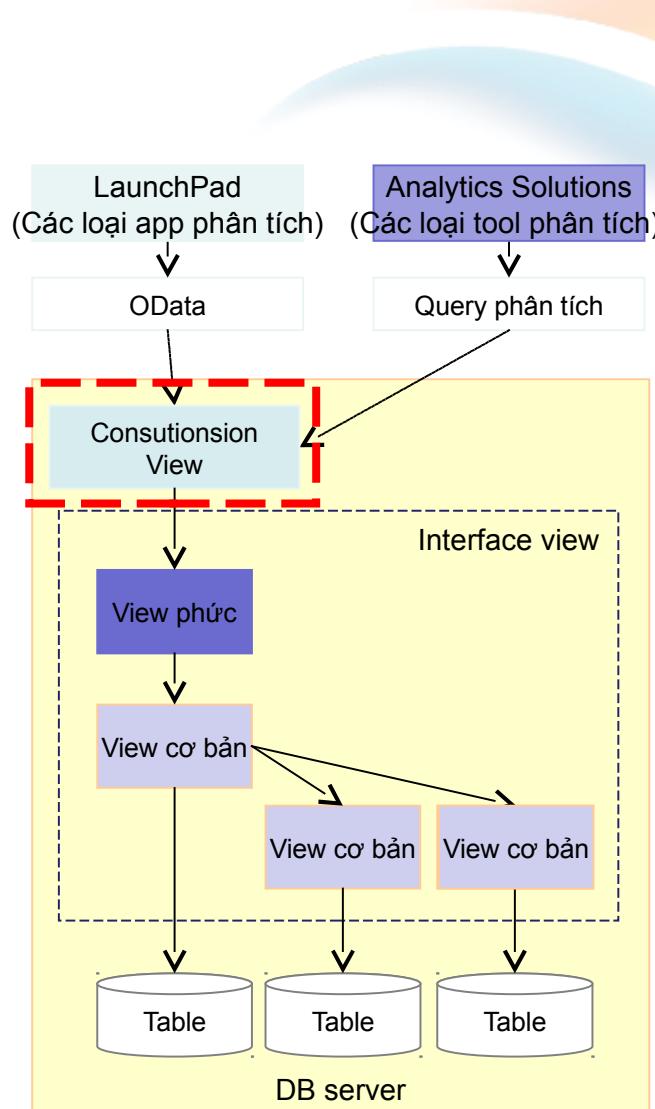
View cơ bản

- View được sử dụng ở view phức hợp.
- View collect data.
 - Tạo view cơ bản như là các phần riêng lẻ kết hợp chúng lại tạo mô hình dữ liệu ảo
 - View chỉ extract trực tiếp table(Không dài dòng) · · · Sơ đồ bên trái①
 - View liên kết (kết hợp) CDS View · · · Sơ đồ bên trái②
- Logic chủ yếu(@ là Annotation)
 - Chỉ định type VDM ở basic (@VDM.viewType : #basic)
 - SELECT từ table BD
 - SELECT từ CDS View
 - Association của CDS View
 - · · ·



View phức

- View được sử dụng ở consutionsion view.
- Là view base trên dữ liệu đã được thu thập bằng view cơ bản, thực hiện các loại tính toán
 - Có thể setting rule sum output ở từng item (Các giá trị SUM, giá trị MAX)
- Logic chủ yếu(@ là Annotation)
 - Chỉ định type VDM ở composite (@VDM.viewType : #composite)
 - SELECT CDS View (View cơ bản)
 - Gắn kết đơn vị cho item số lượng
Nơi tham khảo (@Semantics.quantity.unitofMeasure : '(Item tham khảo/参照項目)')
 - Gắn kết Currency 通貨 cho item số tiền
Nơi tham khảo (@Semantics.amount.currencyCode : '(Item tham khảo/参照項目)')
 - Setting Sum (@DefaultAggregation : #sum)
 - Setting Max (@DefaultAggregation : #max)
 - Setting Min (@DefaultAggregation : #min)
 - . . .



Nếu sử dụng Formula thì sẽ không thể tính toán theo item đã định nghĩa.
Đù không sử dụng Formula thì cũng có thể tính được.

Consumptionsion view

- View được sử dụng bởi các loại app và toll phân tích
- Được sử dụng từ các loại app phân tích của LaunchPad
- Được sử dụng từ các loại Analytics Solution
- Tính toán giữa các item, output new item (Formala, phép tính số học)
- Thiết lập nhánh, parameter khi được sử dụng bởi các loại app và tool phân tích

- Ví dụ app phân tích LaunchPad
Mô hình KPI, trình duyệt truy vấn, ...

- Ví dụ Analytics Solutions
Lumira 、 AnalysisOffice (AO) 、 GUI (RSRT) 、

- Logic chủ yếu(@ là Annotation)
 - Chỉ định VDM type ở consumption (@VDM.viewType : #consumption)
 - Gen OData (@OData.publish : true)
 - Gen Query phân tích (@Analytics.quely : true)
 - SELECT từ CDS View đối tượng
 - Setting parameter dùng để phân tích (Chỉ định parameter trong select)
 - Chỉ định công thức tính toán (@DefaultAggregation : #formula)
 - Trục ngang (@AnalyticsDetails. query .axis : #row)
 - Trục dọc (@AnalyticsDetails. query .axis : #column)
 - Setting trực dùng để phân tích

- Modelling is an activity in which user refine or slice data in the database table by creating information view based on the business scenario. This information views can be used for reporting and decision-making purpose.
- Information view is made from various combinations of content data to create a model for a business scenario.
- Content Data in information view are of two types :
 - **Attribute:** Descriptive and Non-Measureable Data. E.g. Vendor ID, Vendor Name, City, etc.
 - **Measure:** Data can be quantifiable and calculated. E.g. Revenue, Quantity Sold and Counters. The measure is derived from analytic and calculation view. The measure cannot be created in Attribute view.

Types of Attribute

SAP HANA Support three Type of attributes -

Types of Attributes	Activities
• Simple Attribute	It is derived from data foundation.
• Calculated Attribute	It is derived from one or more existing attributes and constants. E.g. Arithmetic calculation or derive the full name from the first and last name.
• Local Attribute	It is used inside modelling views (analytic View / calculation view) for Customize the behavior of attribute, so it is local to Modelling view and cannot access from outside of Modelling view.

- Defined with *key* syntax
- Must be set always at the begin of field list of a CDS Model
- Key should be defined not too long but keep the meaning to be able to understand
- The combination of the individual key fields must uniquely identify a data record as the selection result

```
@AbapCatalog.sqlViewName: 'ZISALESORDERITEM'
define view ZI_SalesOrderItem
    as select from zsalesorderitem
{
    key salesorder      as SalesOrder,
    key salesorderitem as SalesOrderItem,
    product          as Product
}
```

- To check the model consistency and have influence to CDS DCL(Data controlling language) and used by implementing frameworks
- To define the key of an OData entity set

Keys displayed in SQL View

Dictionary: Display View

View field	Table	Field	Key	Data elem.	Mod	DTyp	Length
MANDT	ZSALESORDERITEM	CLIENT	<input checked="" type="checkbox"/>		<input type="checkbox"/>	CLNT	3
SALESORDER	ZSALESORDERITEM	SALESORDER	<input checked="" type="checkbox"/>	VBELN	<input type="checkbox"/>	CHAR	10
SALESORDERITEM	ZSALESORDERITEM	SALESORDERITEM	<input checked="" type="checkbox"/>	POSNR	<input type="checkbox"/>	NUMC	1
PRODUCT	ZSALESORDERITEM	PRODUCT	<input type="checkbox"/>	MATNR	<input type="checkbox"/>	CHAR	40

- Cast statement to specify the type of calculated fields or to make type conversions of existing fields at the database level.
- CDS Models support both elementary ABAP types and data elements as target types for the cast operation.
- PRESERVING TYPE informs the compiler that there is no effective type change and that no cast is required in the database.
- FLTP_TO_DEC is a special CDS-function to converse Floating Point Values in a decimal number

```
define view Z_ViewWithCasts
    as select distinct from t000
{
    t000.logsys as ProjectedField,
    '20170809' as CharacterField,
    cast ( '20170809' as abap.dats ) as DateField,
    cast ( cast ( 'E' as abap.lang ) as sylangu preserving type )
        as LanguageField,
    1.2 as FloatingPointField,
    FLTP_TO_DEC(1.2 as abap.dec(4,2) ) as DecimalField
}
```

- Case statement is used to define a conditioned calculation directly into the CDS View logic.
- If no condition is met, they are given the value “zero”

```
define view Z_ViewWithCaseStatements
    as select from ZI_SalesOrder
{
    key SalesOrder,
        case (SalesOrderType)
            when 'TAF' then 'X'
            when 'OAF' then 'X'
            else ''
        end as IsStandardOrder,
        cast( case (SalesOrderType)
            when 'TAF' then 'X'
            when 'OAF' then 'X'
            else ''
        end as abap.char(3) ) as IsStandardOrderAsChar3,
        case when SalesOrderType = 'TAF' then 'X'
            when SalesOrderType = 'OAF' then 'X'
            else ''
        end as IsStandardOrder2
}
```

- With it you can access information about the runtime session in your CDS View logic

```
define view Z_ViewWithSessionVariables
    as select from t000
{
    $session.client           as ClientField,
    $session.system_date      as SystemDateField,
    $session.system_language   as SystemLanguageField,
    $session.user              as UserField
}
where
    mandt = $session.client
```

Session variable	Description	ABAP similar
\$session.client	Current mandt	Sy-mandt
\$session.system_date	System date of application server	sy-datum
\$session.system_language	Log in language	sy-langu
\$session.user	Current user	sy-uname

- By the standard query to database via open SQL Client-dependent data is automatically adjusted accordingly of the current client of the ABAP session.
- Multi-client Selection accesses must have one of the two additions USING CLIENT or CLIENT SPECIFIED.

```
SELECT *  
      FROM zsalesorder  
      CLIENT SPECIFIED  
      INTO TABLE @DATA(lt_zsalesorder)  
      WHERE client = '001'.
```

- But standard mandate handle :

```
@ClientHandling.type: #INHERITED  
@ClientHandling.algorithm: #AUTOMATED  
//or outdated: @ClientDependent: true  
define view ...
```

- Algorithm @ClientHandling.algorithm: #AUTOMATED more different than @ClientHandling.algorithm: #SESSION_VARIABLE. Session_variable is used to treat the consistent client. It filters the data with the client of the current session of user.
- The affected SQL view then has an additional one Where condition.

```
@AbapCatalog.sqlViewName: 'ZISALESORDER'  
@ClientHandling.algorithm: #SESSION_VARIABLE  
define view ZI_SalesOrder  
    as select from zsalesorder  
{  
    key salesorder      as SalesOrder,  
          salesordertype as SalesOrderType  
}
```

- Is used to merge data set from different data source and unify it. The result of union is harmony result list with unique fields and associations.
- By selecting several selections statements chain together with Union function. Every single selection branch must have the same fields and associations in the same order.
- the respective same position elements of the Selection lists have the same definition.
- The name of the elements must be equal.
- the underlying association definitions must have the same on-conditions, cardinalities, and target entities.
- The typing of the individual fields of the union view, however, must not be identical for each branch. This type specification of the fields becomes rather derived from the first selection branch.

```
define view Z_UnionViewWithoutAssociation
    as select from Z_ViewAsDataSourceA
{
    key FieldA1 as UnionField1,
    key FieldA2 as UnionField2,
    key FieldA3 as UnionField3
}
union select from Z_ViewAsDataSourceB
{
    key FieldB2 as UnionField1,
    key FieldB1 as UnionField2,
    key ''      as UnionField3
}
```

- Union without duplicate
- Union with duplicate key is not available

```
define view Z_UnionViewWithoutDuplicate
  as select from Z_ViewAsDataSourceA
{
  key FieldA1
}
union select from Z_ViewAsDataSourceA
{
  key FieldA1
}
```

```
define view Z_UnionViewWithDuplicate
  as select from Z_ViewAsDataSourceA
{
  FieldA1
}
union all select from Z_ViewAsDataSourceA
{
  FieldA1
}
```

- With join you can model two data source with conditional join.
- The join condition describe criteria under which is the primary data, accessing to the record of secondary data source.
- You can use the data sources of your CDS view, which are linked by a join, to define elements in the projection list of your CDS view. moreover you can use the elements of the linked data sources to enrich where- conditions of your CDS view.
- Otherwise you should renounce the denormalization(*) of the CDS model, or merge data from different CDS models via join condition. This merge data is especially true for CDS view, which you want to reuse.
- Left outer join, inner join is the other statement

Left Outer Join Example

```
define view Z_ViewAsDataSourceD
  as select distinct from t000
{
  key cast( 'A' as abap.char(1) ) as FieldD1,
    cast( 'D' as abap.char(1) ) as FieldD2
}
  union select distinct from t000
{
  key cast( 'C' as abap.char(1) ) as FieldD1,
    cast( 'E' as abap.char(1) ) as FieldD2
}
```

```
define view Z_ViewAsDataSourceE
  as select distinct from t000
{
  key cast( 'D' as abap.char(1) ) as FieldE1,
    key cast( 'H' as abap.char(1) ) as FieldE2
}
  union select distinct from t000
{
  key cast( 'D' as abap.char(1) ) as FieldE1,
    key cast( 'I' as abap.char(1) ) as FieldE2
}
  union select distinct from t000
{
  key cast( 'F' as abap.char(1) ) as FieldE1,
    key cast( 'I' as abap.char(1) ) as FieldE2
}
```

Left Outer Join Example

CDS-View	FieldD1	FieldD2	FieldE1	FieldE2
Z_ViewAsDataSourceD	»A«	»D«	—	—
	»C«	»E«	—	—
Z_ViewAsDataSourceE	—	—	»D«	»H«
	—	—	»D«	»I«
	—	—	»F«	»I«
Z_ViewWithLeftOuterJoin	»A«	»D«	—	»H«
	»A«	»D«	—	»I«
	»A«	»E«	—	»null«
Z_ViewWithInnerJoin	»A«	»D«	—	»H«
	»A«	»D«	—	»I«

```
define view Z_ViewWithLeftOuterJoin
  as select from Z_ViewAsDataSourceD
    left outer to many join Z_ViewAsDataSourceE
      on Z_ViewAsDataSourceD.FieldD2 = Z_ViewAsDataSourceE.FieldE1
{
  key Z_ViewAsDataSourceD.FieldD1,
  key Z_ViewAsDataSourceD.FieldD2,
  key Z_ViewAsDataSourceE.FieldE2
}
```

- The number of records by the join becomes influenced by the cardinality of the join partner: There is a record of the primary data source has more than one corresponding record in the secondary data source.
- The first entry of the CDS View Z_ViewAsDataSourceD , which has the value "A" for the key field FieldD1, is linked with two records of the CDS view Z_ViewAsDataSourceE. Thus, the result list of the CDSView receives Z_ViewWithLeftOuterJoin two records for this entry. This (possible) cardinality jump is indicated by the addition TO MANY in the left- Outer join statement posted.
- Specify cardinality of the left outer join partner You should always set the maximum target cardinality (TO ONE or TO MANY) of a left outer join partner.
- This information serves on the one hand the Documentation of the structure of the CDS view. On the other hand, this can Information the processing of a selection request in the database optimize. However, if you maintain the cardinality, you should be careful that this is defined correctly, otherwise the selection result can be faulty.

- The second entry of the view Z_ViewAsDataSourceD, which returns the value "C" for the Key field FieldD1 has no join partner. According to the left-outer Join semantics, this record still remains in the result list. The linked field FieldE2 of the CDS view Z_ViewWithLeftOuterJoin receives in this case, at the database level, the null value.
- Necessary treatment of zero values At SQL level, the difference between initial values and null values is significant. As the rule, zero values from left outer join condition without join partner. When modeling your CDS view logic, you need to make that distinction observe and, if necessary, treat specifically.

```
define view Z_ViewWithJoinAndFilter
    as select from Z_ViewAsDataSourceD
        left outer to many join Z_ViewAsDataSourceE
            on Z_ViewAsDataSourceD.FieldD2 = Z_ViewAsDataSourceE.FieldE1
{
    key Z_ViewAsDataSourceD.FieldD1,
    key Z_ViewAsDataSourceD.FieldD2,
    key Z_ViewAsDataSourceE.FieldE2
}
where Z_ViewAsDataSourceE.FieldE2 is not null
```

- SQL aggregation functions allow you to perform the calculations of predefined aggregates efficiently in the database
- To do this, first define the aggregation level to which you want the result want to condense
- You enter the aggregation level with the Syntax element GROUP BY.
- The group-by statement is behind the Projection list and lists all the fields of the data sources that are influenced in the projection list.
- Avoid memory overflows: At the execution time of the selection query, a memory overflow when applying aggregation functions, using the check the expected value ranges and existing type of the fields and the type specifications, if necessary, by explicit type conversions expand.

SQL Aggregation function: Example

```
define view Z_ViewAggregationBase
  as select distinct from t000
{
  key 'A' as Field1,
  key 'A' as Field2,
    cast( 1 as abap.int1) as Field3
}
  union select distinct from t000
{
  key 'A' as Field1,
  key 'B' as Field2,
    cast( 2 as abap.int1) as Field3
}
  union select distinct from t000
{
  key 'A' as Field1,
  key 'C' as Field2,
    cast( 3 as abap.int1) as Field3
}
```

Field1	Field2	Field3
»A«	»A«	»1«
»A«	»B«	»2«
»A«	»C«	»3«

```
define view Z_ViewWithAggregations
  as select from Z_ViewAggregationBase
{
  key Field1,
    min(Field3) as FieldWithMin,
    max(Field3) as FieldWithMax,
    avg(Field3) as FieldWithAvg,
    cast( sum(Field3) as abap.int4 ) as FieldWithSum,
    count( distinct Field1 ) as FieldWithCountDistinct,
    count(*) as FieldWithCountAll
}
  group by Field1
```

- With associations you can direct relationships between CDS views capture.
- You can get the associations beyond that but also for external use by the users of your CDSViews expose. The possible external use includes both the reuse of those contained in the association definitions logic within consecutive built-in CDS views of a CDS view Stacks as well as the use of associations within open SQL selection requests of an ABAP application.
- In addition, associations are valuable as carriers of many metadata for the frameworks that use or interpret them, the numerous Derive functions from this metadata. For example, you can Associated dimension CDS views as the source of display attributes in analytical reports
- Fields and association are seen as CDS elements
- The namespace of the associations: The names of the CDS elements (fields and associations) and the parameter names share the same namespace. You have to be unique within a CDS model.

- The association definition describes the coupling of the definition source with an association goal.
- Association goal:
 - CDS-Views
 - CDS-Tabellenfunktionen
 - SQL-Views
 - Datenbanktabellen

- semantic names of the association should be written by an underscore prefix Ex: `_Item`.

```
define view ZI_SalesOrder  
...  
    association [0..*] to ZI_SalesOrderItem as _Item  
        on $projection.SalesOrder = _Item.SalesOrder  
...
```

Cardinality of Association

Cardinality	Minimal	Maximal
[]	0	1
[1]	0	1
[0..1]	0	1
[1..1]	1	1
[0..*]	0	Unlimited
[1..*]	1	Unlimited
No(default logic)	0	1

- External use :To make it accessible to users, this must be analogous to the fields in the projection list of the CDS view.
- Internal use: Without this exposure is the association definition only an internal implementation detail whose function you also over would have reached the definition of corresponding join coupling
- Association _Item is added to the projection list. It is thus part of the external signature of the CDS view mode

```
define view ZI_SalesOrder
  ...
    association [0..*] to ZI_SalesOrderItem as _Item ...
{
  key SalesOrder,
    _Item,
  ...
}
```

- Parameters are part of the signature of the CDS models.
- They represent input values used by the caller as data selection need to be supplied.
- CDS views without parameters are called SQL views on the SAP HANA Database created.
- CDS views with parameters are displayed analogous to the CDS table functions as SAP HANA table functions in the database.
- The type of parameter can be ABAP Type also data element. And it has prefix with “P_”

```
define view Z_ViewWithParameters  
with parameters  
    P_KeyDate : abap.dats,  
    P_Language : sylangu  
    ...
```

- Parameter can be used :
 - To formulate where condition
 - Define fields of the projection list of a CDS view
 - Provide values for parameters of other data sources
- Syntax element : \$parameters or colon

Parameter example

```

define view Z_ViewWithParameters
with parameters
  P_KeyDate : abap.dats,
  P_Language : sylangu
  as select from Z_ViewWithParametersDataSource
association [0..*] to Z_ViewWithParametersAscTrgt as _Target
  on $projection.KeyField = _Target.KeyField
association [0..1] to Z_ViewWithParametersAscTrgt
as _FilteredTarget
  on $projection.KeyField = _FilteredTarget.KeyField
  and $projection.Language = _FilteredTarget.Language
{
  key KeyField,
  ValidityEndDate,
  ValidityStartDate,
  :P_Language as Language,
  _Target(P_ValidityDate: :P_KeyDate)[1:Language=
    :P_Language].KeyField
  as TargetKeyField, _FilteredTarget
}
where ValidityEndDate >= :P_KeyDate
and ValidityStartDate <= :P_KeyDate
and Language      = :P_Language

```

```

define view Z_ViewWithParametersAscTrgt
with parameters
  P_ValidityDate : abap.dats
  as select from Z_ViewWithParametersDataSource
{
  key KeyField,
  key Language,
  ValidityEndDate,
  ValidityStartDate
}
where ValidityEndDate >= $parameters.P_ValidityDate
  and ValidityStartDate <= $parameters.P_ValidityDate

```

Parameter example: Explain

- When accessing the key field of the associated CDS View Z_ViewWithParameterAscTrgt via the following path expression _Target(P_validityDate::P_KeyDate)[1:Language =: P_Language].KeyField must check with parameter P_ValidityDate in View Z_ViewWithParameters.
- The next check is to check the parameter of P_Language, to apply as a filter to the associated field of Language. Through that, the key and the associated record of the target view is fixed.
- As a result, the maximal cardinality of the path is “1”

```
define view Z_ViewWithParametersConsumer  
as select from Z_ViewWithParameters(  
    P_KeyDate:  
        $session.system_date,  
    P_Language: 'E')  
{  
    key KeyField  
}
```

In ABAP

```
SELECT *  
FROM z_viewwithparameters( p_keydate = @sy-datum,  
    p_language = 'E' )  
INTO TABLE @DATA(lt_z_viewwithparameter).
```

In ABAP implicit parameter , however the parameter here is optional :

```
SELECT *  
FROM z_viewwithoptionalparameter  
INTO TABLE @DATA(lt_z_viewwithoptionalparameter).
```

In ABAP explicit parameter :

```
SELECT *  
FROM z_viewwithoptionalparameter( p_keydate = @sy-datum )  
INTO TABLE @DATA(lt_z_viewwithoptionalparameter).
```

The Open-SQL-Interface support to work automatically with parameter by using annotation @Environment.systemField. The values of this annotation can be
»CLIENT«, »SYSTEM_DATE«, »SYSTEM_TIME«, »SYSTEM_LANGUAGE«
und »USER« corresponding with the values of ABAP-System fields:
sy-mandt, sy-datum, sy-zeit, sy-langu und sy-uname

```
define view Z_ViewWithOptionalParameter
    with parameters
        @Environment.systemField: #SYSTEM_DATE
        P_KeyDate : abap.dats
    as select distinct from Z_ViewWithParametersDataSource
{
    key KeyField
}
where ValidityEndDate    >= $parameters.P_KeyDate
    and ValidityStartDate <= $parameters.P_KeyDate
```

Conversion function for currency and quantity unit

- The conversion function is used to calculate the currency and quantity unit from data records.
- The data records are influenced via process of conversion requirement in logic. Example : Calculate the time dependent with country code.
- CDS allows the conversion function directly in CDS data model

Conversion example of the quantity

```
define view Z_ViewWithUnitConversion
    with parameters
        P_DisplayUnit      : msehi
    as select from ZI_SalesOrderItem
    {
        key SalesOrder,
        key SalesOrderItem,
        @Semantics.quantity.unitOfMeasure: 'OrderQuantityUnit'
        OrderQuantity,
        @Semantics.unitOfMeasure: true
        OrderQuantityUnit,
        @Semantics.quantity.unitOfMeasure: 'OrderQuantityDisplayUnit'
        unit_conversion( quantity      => OrderQuantity,
                          source_unit   => OrderQuantityUnit,
                          target_unit   => :P_DisplayUnit,
                          error_handling => 'FAIL_ON_ERROR' )
        as OrderQuantityInDisplayUnit,
        @Semantics.unitOfMeasure: true
        :P_DisplayUnit as OrderQuantityDisplayUnit
    }
```

Conversion example of currency

```
define view Z_ViewWithCurrencyConversion
    with parameters
        P_DisplayCurrency : waers_circ,
        P_ExchangeRateDate : sydatum
    as select from ZI_SalesOrderItem
    {
        key SalesOrder,
        key SalesOrderItem,
        @Semantics.amount.currencyCode: 'TransactionCurrency'
        NetAmount,
        @Semantics.currencyCode: true
        TransactionCurrency,
        @Semantics.amount.currencyCode: 'DisplayCurrency'
        currency_conversion( amount          => NetAmount,
                             source_currency => TransactionCurrency,
                             target_currency  => :P_DisplayCurrency,
                             exchange_rate_date => :P_ExchangeRateDate,
                             exchange_rate_type => 'M',
                             round           => 'X',
                             decimal_shift   => 'X',
                             decimal_shift_back => 'X',
                             error_handling  => 'FAIL_ON_ERROR' )
        as NetAmountInDisplayCurrency,
        @Semantics.currencyCode: true
        :P_DisplayCurrency as DisplayCurrency,
    }
```

- SAP CDS annotations are evaluated by SAP frameworks and can be either ABAP annotations or framework-specific annotations.
- To enrich the SQL-logic of data model with extra metadata. During ABAP- runtime and using CDS Model CDS annotation can integrate extra information.
- CDS annotations that are evaluated by **ABAP runtime**:
 - AbapCatalog Annotations
 - AccessControl Annotations
 - ClientDependent Annotations
 - DataAging Annotations
 - EndUserText Annotations
 - Environment Annotations
 - MappingRole Annotations
 - Metadata Annotations
 - Semantics Annotations

- CDS annotations that (as a rule) are evaluated during runtime by specific frameworks such as **SADL**, **BOPF**, **Analytics**, or **Enterprise Search**:
 - Analytics Annotations
 - AnalyticsDetails Annotations
 - Consumption Annotations
 - DefaultAggregation Annotations
 - EnterpriseSearch Annotations
 - Hierarchy Annotations
 - ObjectModel Annotations
 - OData Annotations
 - Search Annotations
 - Semantics Annotations
 - UI Annotations
 - VDM Annotations
-

Domain	Description
ABAPCatalog	Control of the ABAP runtime environment and the ABAP dictionary
AccessControl	Documentation and control of authorization checks for CDS Models
Aggregation	Distinction of elements that can also be used as aggregating indicators
Analytics	Definition of analytical data models and applications
AnalyticsDetails	Definition of details of a analytical query, such as its default layout and the applied exception aggregations
ClientHandling	Control of client treatment
Consumption	Information for the users of the CDS Models, which are specially evaluated in framework implementation
EndUserText	Definition of translatable text of specified user
Environment	Control the default logic of parameters with system variables
Hierarchy	Definition of hierarchical relationships
Metadata	Control of annotation enhancements of CDS view by propagation and CDS Metadata extensions
ObjectModel	Describe the basic structural characteristics of the data models, including the labeling of their transactional aspects
OData	Auto exposure of CDS Models as OData Services
Search	Control of the search functionality
Semantics	Description of the basic semantics of elements and parameters
UI	Semantical definition of a UI display view that is independent with the specific UI implementation technology
VDM	Classification of the CDS models in the virtual data model

AbapCatalog-Annotations

Annotation	Meaning	Scope	Framework
AbapCatalog.buffering.numberOfKeyFields	SAP buffering, number of key fields when buffering generic areas	View	ABAP
AbapCatalog.buffering.status	SAP buffering, enables and disables buffering	View	ABAP
AbapCatalog.buffering.type	SAP buffering, defines the buffering type	View	ABAP
AbapCatalog.compiler.compareFilter	CDS view, specifies how filter conditions are evaluated in path expressions	View	ABAP
AbapCatalog.preserveKey	see CDS Annotations	View	ABAP
AbapCatalog.sqlViewAppendName	CDS view extension, name of the append view	View Extension	ABAP
AbapCatalog.sqlViewName	CDS view, name of the database view	View	ABAP
AbapCatalog.viewEnhancementCategory[]	see CDS Annotations	View	ABAP

AccessControl-Annotations

Annotation	Meaning	Scope	Framework
AccessControl.authorizationCheck	CDS authorizations, controls the authorization check	Table Function View	ABAP, SADL

Analytics-Annotations

Annotation	Meaning	Scope	Framework
Analytics.dataCategory	see CDS Annotations	Table Function View	see CDS Annotations
Analytics.dataExtraction.enabled	see CDS Annotations	Table Function View	see CDS Annotations
Analytics.hidden	see CDS Annotations	Table Function View	see CDS Annotations
Analytics.planning.enabled	see CDS Annotations	Table Function View	see CDS Annotations
Analytics.query	see CDS Annotations	Table Function View	see CDS Annotations
Analytics.writeBack.className	see CDS Annotations	Table Function View	see CDS Annotations

AnalyticsDetails-Annotations

Annotation	Meaning	Scope	Framework
AnalyticsDetails.exceptionAggregationSteps[].exceptionAggregationBehavior	see CDS Annotations	Element	see CDS Annotations
AnalyticsDetails.exceptionAggregationSteps[].exceptionAggregationElements[]	see CDS Annotations	Element	see CDS Annotations
AnalyticsDetails.planning.disaggregation	see CDS Annotations	Element	see CDS Annotations
AnalyticsDetails.planning.distribution	see CDS Annotations	Element	see CDS Annotations
AnalyticsDetails.planning.distributionReference	see CDS Annotations	Element	see CDS Annotations
AnalyticsDetails.planning.enabled	see CDS Annotations	Element	see CDS Annotations
AnalyticsDetails.query.axis	see CDS Annotations	Element	see CDS Annotations
AnalyticsDetails.query.decimals	see CDS Annotations	Element	see CDS Annotations
AnalyticsDetails.query.displayHierarchy	see CDS Annotations	Element	see CDS Annotations
AnalyticsDetails.query.formula	see CDS Annotations	Element	see CDS Annotations
AnalyticsDetails.query.hierarchyBinding[].type	see CDS Annotations	Element	see CDS Annotations
AnalyticsDetails.query.hierarchyBinding[].value	see CDS Annotations	Element	see CDS Annotations
AnalyticsDetails.query.hierarchyBinding[].variableSequence	see CDS Annotations	Element	see CDS Annotations
AnalyticsDetails.query.hierarchyInitialLevel	see CDS Annotations	Element	see CDS Annotations
AnalyticsDetails.query.scaling	see CDS Annotations	Element	see CDS Annotations
AnalyticsDetails.query.totals	see CDS Annotations	Element	see CDS Annotations
AnalyticsDetails.query.variableSequence	see CDS Annotations	Element	see CDS Annotations
AnalyticsDetails.resultValueSource	see CDS Annotations	Element	see CDS Annotations

ClientDependent-Annotations

Annotation	Meaning	Scope	Framework
ClientDependent	Client dependency, specifies how clients are handled	Table Function View	ABAP

Consumption-Annotations

Annotation	Meaning	Scope	Framework
Consumption.defaultValue	Default value	Parameter	Global
Consumption.derivation.binding[].targetElement	Parameters using an entity or procedure	Element Parameter	Global
Consumption.derivation.binding[].targetParameter	Parameters using an entity or procedure	Element Parameter	Global
Consumption.derivation.binding[].type	Parameters using an entity or procedure	Element Parameter	Global
Consumption.derivation.binding[].value	Parameters using an entity or procedure	Element Parameter	Global
Consumption.derivation.derivationFilter	Parameters using an entity or procedure	Element Parameter	Global
Consumption.derivation.lookupEntity	Parameters using an entity or procedure	Element Parameter	Global
Consumption.derivation.procedure	Parameters using an entity or procedure	Element Parameter	Global
Consumption.derivation.resultElement	Parameters using an entity or procedure	Element Parameter	Global
Consumption.filter.defaultValue	Filters specified for an element	Element	Global
Consumption.filter.hidden	Filters specified for an element	Element	Global
Consumption.filter.hierarchyBinding[].type	Filters specified for an element	Element	Global
Consumption.filter.hierarchyBinding[].value	Filters specified for an element	Element	Global
Consumption.filter.hierarchyBinding[].variableSequence	Filters specified for an element	Element	Global
Consumption.filter.mandatory	Filters specified for an element	Element	Global
Consumption.filter.multipleSelections	Filters specified for an element	Element	Global
Consumption.filter.selectionType	Filters specified for an element	Element	Global
Consumption.groupWithElement	Semantic dependency between elements	Element	Global
Consumption.hidden	Hidden element	Element Parameter	Global
Consumption.labelXElement	Assignment of the short text of an element	Element Parameter	Global
Consumption.quickInfoElement	Assignment of the tooltip of an element	Element Parameter	Global
Consumption.semanticObject	Assignment of semantics	Element Parameter Table Function View	Global
Consumption.valueHelp	Assignment of an input help	Element Parameter	Global

DefaultAggregation-Annotations

Annotation	Meaning	Scope	Framework
DefaultAggregation	see CDS Annotations	Element	see CDS Annotations

EndUserText-Annotations

Annotation	Meaning	Scope	Framework
EndUserText.label	Texts, short text connected to translation	Element Parameter Table Function View	ABAP
EndUserText.quickInfo	Texts, tooltip connected to translation	Element Parameter	ABAP

OData-Annotations

Annotation	Meaning	Scope	Framework
OData.publish	OData, generates a suitable OData service when the CDS entity is activated	Table Function View	SADL

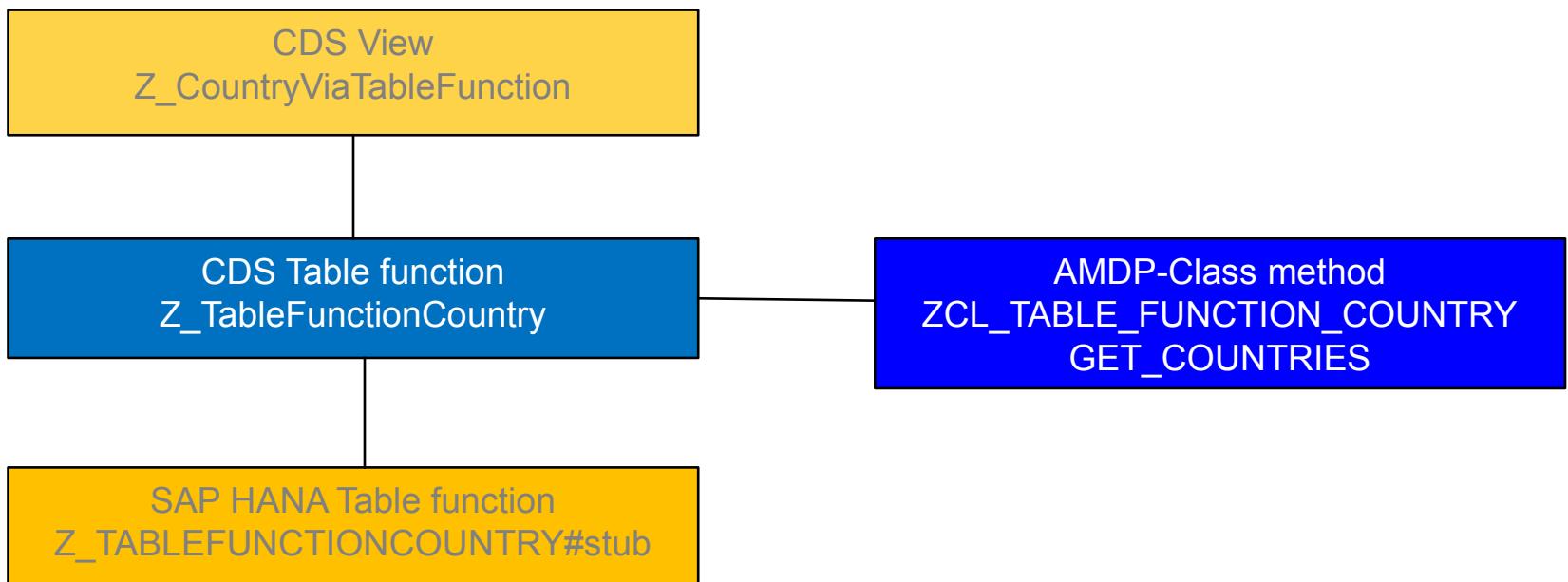
Semantics-Annotations

Annotation	Meaning	Scope	Framework
Semantics.address.city	Address, city	Element Parameter	see CDS Annotations
Semantics.address.country	Address, country	Element Parameter	see CDS Annotations
Semantics.address.label	Address, label	Element Parameter	see CDS Annotations
Semantics.address.postBox	Address, post office box	Element Parameter	see CDS Annotations
Semantics.address.region	Address, region	Element Parameter	see CDS Annotations
Semantics.address.street	Address, street	Element Parameter	see CDS Annotations
Semantics.address.subRegion	Address, subregion	Element Parameter	see CDS Annotations
Semantics.address.type[]	Address, type	Element Parameter	see CDS Annotations
Semantics.address.zipCode	Address, postal code	Element Parameter	see CDS Annotations
Semantics.amount.currencyCode	Currency field	Element	ABAP
Semantics.businessDate.at	Business date, date on which business data is valid	Element Parameter	see CDS Annotations
Semantics.businessDate.createdAt	Business data, date on which business data was created	Element Parameter	see CDS Annotations
Semantics.businessDate.from	Business date, date from which business data is valid	Element Parameter	see CDS Annotations
Semantics.businessDate.lastChangedAt	Business date, date on which business data was modified	Element Parameter	see CDS Annotations
Semantics.businessDate.to	Business date, date to which business data is valid	Element Parameter	see CDS Annotations
Semantics.calendar.dayOfMonth	Calendar, day of the month	Element Parameter	see CDS Annotations
Semantics.calendar.dayOfYear	Calendar, day of the year	Element Parameter	see CDS Annotations
Semantics.calendar.month	Calendar, month of the year	Element Parameter	see CDS Annotations
Semantics.calendar.quarter	Calendar, quarter of the year	Element Parameter	see CDS Annotations
Semantics.calendar.week	Calendar, week of the year	Element Parameter	see CDS Annotations
Semantics.calendar.year	Calendar, year	Element Parameter	see CDS Annotations
Semantics.calendar.yearMonth	Calendar, month of the year (including year)	Element Parameter	see CDS Annotations
Semantics.calendar.yearQuarter	Calendar, quarter of the year (including year)	Element Parameter	see CDS Annotations
Semantics.calendar.yearWeek	Calendar, week of the year (including year)	Element Parameter	see CDS Annotations
Semantics.calendarItem.categories	Calendary entry, appointment category	Element Parameter	see CDS Annotations

- CDS View use the ability of SAP HANA over SQL and table function is one of those abilities. Table function in CDS allows execution of SAP HANA SQLScript and can combine with CDS. You can use native SAP HANA functions for your CDS models.
- The SAP HANA database is not just a relational database system, which can be used via the Structured Query Language (SQL), but also offers tools and function libraries for a large number of areas, eg. Eg predictive analytics, financial mathematics, Data mining, full-text search and graph processing.
- It is through the technique of ABAP Managed Database Procedures (AMDP) possible to execute SAP HANA SQLScript from ABAP - with input parameters and the return of a result. This technique is used for CDS-table functions (Table Function). Technically it creates and executes an SAP HANA table function

- More complex CDS table functions use the imperative Pro-SQLScript style of SQLScript with classical control structures to control the processing flow explicitly.
- the AMDP technique and the CDS table functions are special useful, if special skills of SAP HANA are combined with the application data in ABAP.
- An example is Predictive Analytics, Predictive Analytics z. B. sales or liquidity forecasts or forecasts of contract consumption in shopping. This involves using native SAP HANA tools customer-specific analysis procedures defined and parameterized. As Interface to the ABAP world is a procedure in SAP HANA: the prediction procedure.
- SQL-query is only available in SAP HANA SQL :
 - functions for string processing (Strings), for example for regular expressions (REPLACE_REGEXPR etc.) or the Concatenation of strings of different data lines (STRING_AGG)
 - Factory calendar functions (ADD_WORKDAYS, WORKDAYS_BETWEEN)
 - Full text search with many options (CONTAINS)
 - Hierarchy functions

- Create a table function to display list of country. We need to implement three object in ABAP and then generate one HANA table function.



- Create new CDS DDL
- Choose the template for declaring table Function

```
@ClientHandling.type:      #CLIENT_DEPENDENT
@ClientHandling.algorithm: #SESSION_VARIABLE
define table function Z_TableFunctionCountry
    with parameters
        @Environment.systemField: #CLIENT
        P_SAPClient : vdm_v_sap_client
    returns
    {
        mandt           : vdm_v_sap_client;
        Country         : land1_gp;
        CountryCurrency : waers_005;
        IndexBasedCurrency : curin;
        HardCurrency    : curha;
        TaxCalculationProcedure : kalsm_d;
        CountryThreeLetterISOCode : intca3;
        CountryThreeDigitISOCode : intcn3;

    }
    implemented by method
        zcl_table_function_country=>get_countries
```

- Activate the table function

Implement method to use table function

- Create a ABAP class ZCL_TABLE_FUNCTION_COUNTRY and the method get_countries

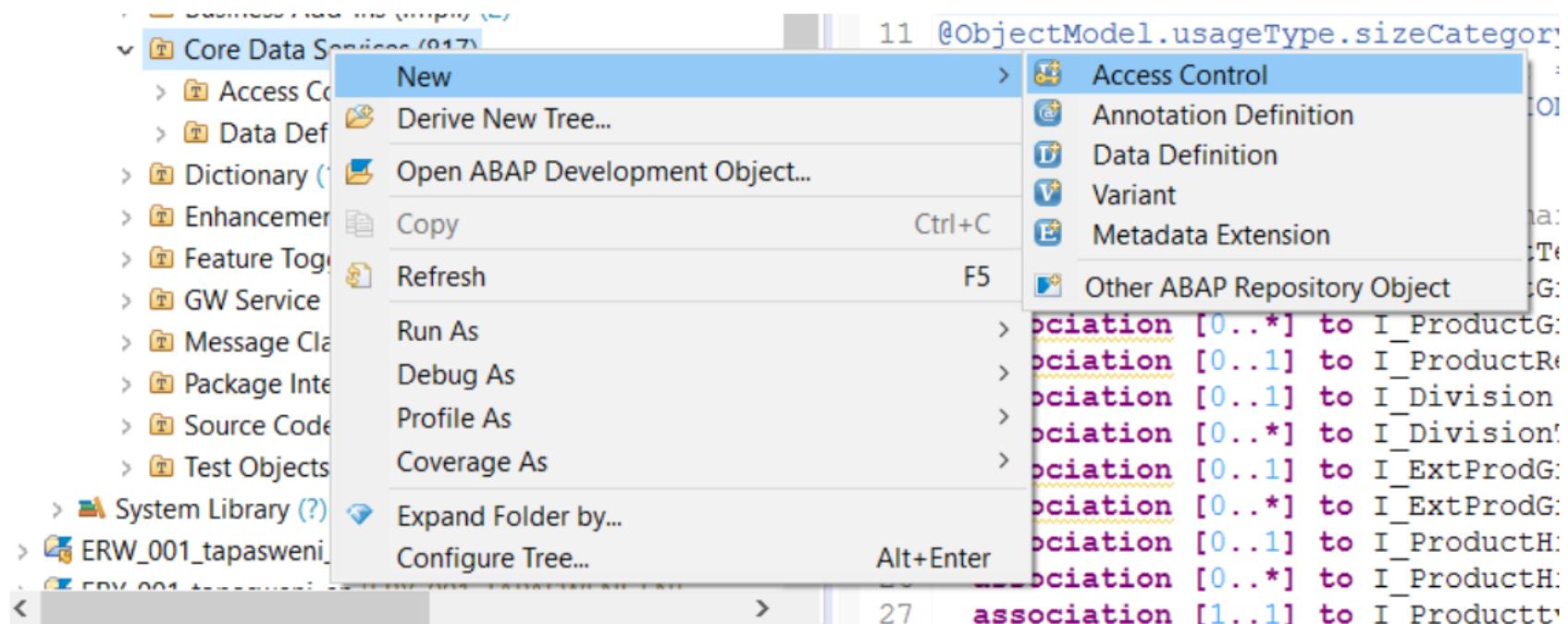
```

CLASS ZCL_TABLE_FUNCTION_COUNTRY DEFINITION
  PUBLIC
  FINAL
  CREATE PUBLIC .
  PUBLIC SECTION.
    INTERFACES if_amdp_marker_hdb.
    CLASS-METHODS get_countries
      FOR TABLE FUNCTION Z_TableFunctionCountry.
  PROTECTED SECTION.
  PRIVATE SECTION.
ENDCLASS.

CLASS ZCL_TABLE_FUNCTION_COUNTRY IMPLEMENTATION.
  METHOD get_countries
    BY DATABASE FUNCTION FOR HDB
    LANGUAGE SQLSCRIPT
    OPTIONS READ-ONLY
    USING IFICountry.
    RETURN
    SELECT
      :P_SAPClient as mandt,
      Country,
      CountryCurrency,
      IndexBasedCurrency,
      HardCurrency,
      TaxCalculationProcedure,
      CountryThreeLetterISOCode,
      CountryThreeDigitISOCode
    FROM
      IFICountry
    WHERE
      mandt = :P_SAPClient;
  ENDMETHOD.
ENDCLASS.
```

- From an authorization perspective, an end user should always only access the features and access data he is required to fulfill his operational requirements Tasks actually needed (Principle of least Privilege).
- Launch permission is also called a control that gives access to differentiated records based on the contained values (Instance authorization). This means, a user should in principle only have minimum permissions.
- Start and Instance permissions distinguish A system-based authorization control can both on functional level as well as at the level of individual data sets or on partial information of individual data records.

Creating a new access control



```
@EndUserText.label: '${dcl_source_description}'
@MappingRole: true
define role ${dcl_source_name} {
    grant
        select
            on
                ${cds_entity}
                    where
                        ${condition};
    // -- Example WHERE condition
    // -- Two-field mapping to PFCG authorization with filter on read authorization
    // ( SalesOrderID, OrgID ) = aspect pfcg_auth( S_ACN_DEMO, SACMTSOID, SACMORGUID, ACTVT = '03' )
    // and
    // -- Equals-or-initial operator
    // ( CustomerCountry ) ?= aspect pfcg_auth( S_ACN_DEMO, SACMCNTRY )
    // or
    // -- Reference to the logged on users name
    // CreatedBy = aspect User
    // or
    // -- Literal condition
    // isPublic = 'X';
}
```

- Hierarchy of the evaluation of a PFCG condition :
 - The following rule applies with respect to the hierarchy of the evaluation of a PFCG condition:
 - If multiple authorizations are evaluated, the resulting conditions are joined using a logical “or”.
 - In the conditions of each authorization used, the values for the authorization fields in question are joined using a logical “and”.
 - If there are multiple values for an authorization field, they are joined using a logical “or”.

DDL:

```
@AbapCatalog.sqlViewName: 'DEMO_CDS_LITPFCG'  
@AccessControl.authorizationCheck: #CHECK  
define view demo_cds_auth_lit_pfcg  
as select from  
scarr  
{  
key carrid,  
carrname,  
currcode,  
url  
};
```

DCL:

```
@MappingRole: true  
define role demo_cds_role_lit_pfcg {  
grant select on demo_cds_auth_lit_pfcg  
where (carrid) =  
aspect pfcg_auth (s_carrid, carrid, actvty='03') and  
currcode = 'EUR'; }
```



BOPF- Business Object Processing Framework

- The Business Object Processing Framework is an ABAP OO-based framework that provides a set of generic services and functionalities to speed up, standardize, and modularize your development.
- BOPF manages the entire life cycle of your business objects and covers all aspects of your business application development. Instead of expending effort for developing an application infrastructure, the developer can focus on the individual business logic.
- Using BOPF, you get the whole application infrastructure and integration of various components for free. This allows you to rapidly build applications on a stable and customer-proved infrastructure.

Structure of BOPF

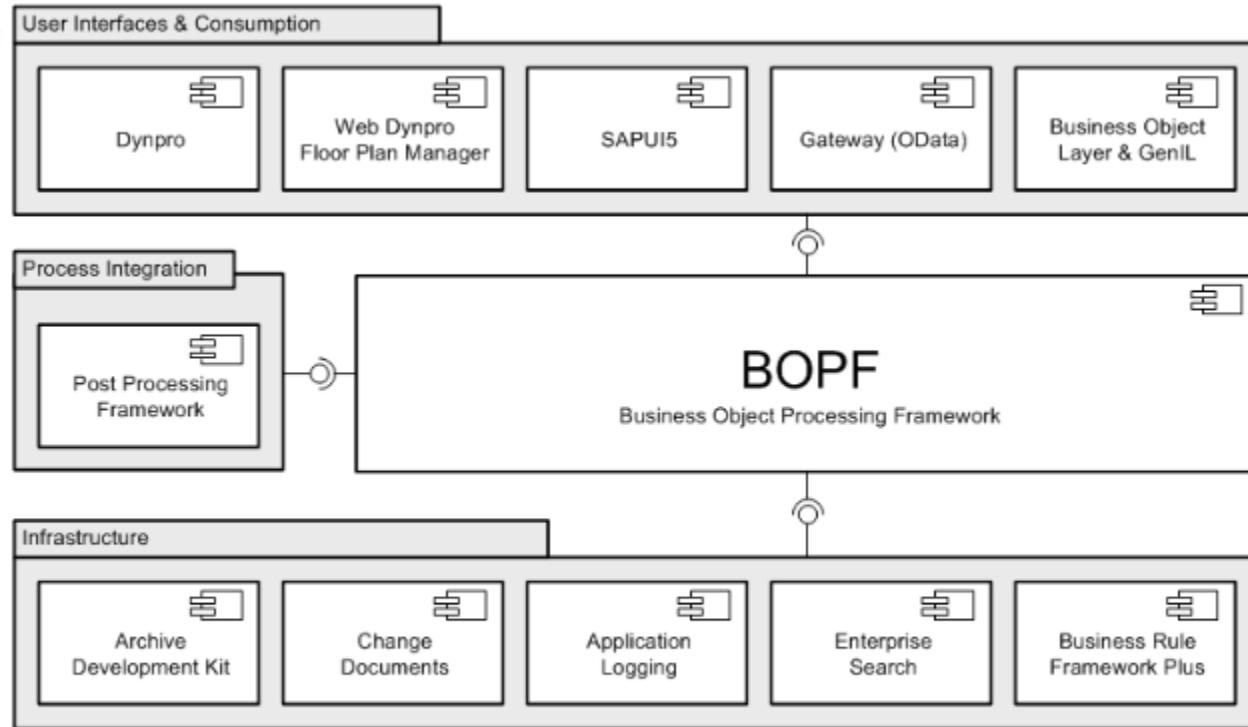


Figure: BOPF acts as a bridging unit between various components

Domain	BOPF provides a standard interface for consumption by the classic Dynpro UI
Web Dynpro / Floor Plan Manager (FPM)	The generation and configuration of complex user interfaces has never been as easy as it is today with the FPM. FPM is implemented as a Web Dynpro component and can be easily integrated with BOPF. BOPF provides configurable and codeless integration of FPM and enables you to seamlessly consume the services of BOPF Business Objects in a modification-free environment.
SAPUI5	SAPUI5 is designed for building lightweight UIs for casual use.
Gateway (OData)	SAP NetWeaver Gateway is a technology that provides a convenient way to connect devices, environments, and platforms to SAP software based on market standards. The BOPF integration of the Gateway is based on REST and OData standards.
Business Object Layer & GenIL	<p>The Business Object Layer (BOL) provides a generic API for accessing business data.</p> <p>The Generic Interaction Layer (GenIL) enables uniform access to business data using a stateless request/response format. BOPF provides adapters for BOL and GenIL integration.</p>

Example of BOPF



Hướng

- Code
- Tạo PCL
- Chụp evidence cho UT

- **Get Item**
 - Confirm Item có phải là key hay không
 - Confirm tất cả item đã được get đúng theo tài liệu mô tả hay không
 - Confirm định nghĩa annotation cho item get đã đúng như thiết kế chưa
- **Get Table**
 - Confirm có đang select đúng Table hay không
- **Điều kiện select**
 - Confirm điều kiện get có đúng và đủ như tài liệu hay chưa
- **Join**
 - Confirm Join có đúng như thiết kế chưa bao gồm: Inner join, left join
 - Điều kiện Join đúng và đủ chưa
- **Association**
 - Confirm association đã đúng như thiết kế chưa
 - Điều kiện Join đúng và đủ chưa
- **Confirm header annotation**
 - Confirm đã đúng như thiết kế chưa
- **Trường hợp View with parameter**
 - Confirm parameter đã đúng như thiết kế chưa

Tạo sheet PCL(tiếng Nhật)

【Sample】DD409_テスト仕様書(単体テスト)(ZGV8CO_R5105_CC機器マスク) - Excel

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
1	Project	経理高度化プロジェクト	Phase	詳細仕様	Business Process/ System	加工費分析/S4	Created	2017/12/25	Updated		K	L	M	N	O	P	Q	R	S	T
2	Deliverable	CDS_View定義書	DeliverableDetail	CC機器マスク	Form Number	DD409	Created by	日本 夏見	Updated by		K	L	M	N	O	P	Q	R	S	T
3	● テスト完了項目																			
4	● テスト未完了項目																			
5	10 目的: 目標選択																			
6	11 テストケース																			
7	12 No.	ケースNo.	ソース箇所	ソース箇所2	条件	予測結果	実施	担当	結果シート名	結果(1)	結果(1)	結果(2)	結果(2)	結果(2)	結果(3)	結果(3)	データ1	データ1	データ1	
8	13	001	BASIC (Z1_ZGV8COCDS0158)				合規	日 善	データ1	データ2	データ3	データ4	データ5	データ6	データ7	データ8	データ9	データ10	データ11	
9	14	001	管理領域	削除フラグ = ''		データプレビューでZGV8COT1003から条件に合致する値が取得できていること			○											
10	15	002	KOKRS	Key		Key設定されていること			○											
11	16	003	会計年度	削除フラグ = ''		データプレビューでZGV8COT1003から条件に合致する値が取得できていること			○											
12	17	004	GIAGR	Key		Key設定されていること			○											
13	18	005	会計期間	削除フラグ = ''		データプレビューでZGV8COT1003から条件に合致する値が取得できていること			○											
14	19	006	RFMAX	Key		Key設定されていること			○											
15	20	007	バージョン	削除フラグ = ''		データプレビューでZGV8COT1003から条件に合致する値が取得できていること			○											
16	21	008	VERSN	Key		Key設定されていること			○											
17	22	009	工場区分	削除フラグ = ''		データプレビューでZGV8COT1003から条件に合致する値が取得できていること			○											
18	23	010	ZFACTDIV	Key		Key設定されていること			○											
19	24	011	ラインコード(工場/C)	削除フラグ = ''		データプレビューでZGV8COT1003から条件に合致する値が取得できていること			○											
20	25	012	ZZLINECOFFC	Key		Key設定されないこと			○											
21	26	013	資産コード(機器)	削除フラグ = ''		データプレビューでZGV8COT1003から条件に合致する値が取得できていること			○											
22	27	014	ZKKSISAN	Key		Key設定されないこと			○											
23	28	015	I_ControlingArea	ASSOCIATION条件		結合条件がCOS View定義書と合致すること			○											
24	29	016	Z1_ZGV8COCDS0158	ASSOCIATION条件		結合条件がCOS View定義書と合致すること			○											
25	30	017	Z1_ZGV8			View定義書と合致すること			○											
26	31	Composite (Z1_ZGV8)																		
27	32	018	管理領域			データプレビューでZGV8COT1003から条件に合致する値が取得できているこ			○											
28	33	019	KOKRS			りうこと			○											
29	34	020	会計年度			ーでZ1_ZGV8COCDS0158から条件に合致する値が取得できているこ			○											
30	35	021				と														

Sheet thứ 3 của
file sample

Select destination and press ENTER or choose Paste



Tạo sheet PCL(tiếng Việt)

【Sample】DD409_テスト仕様書(単体テスト)(ZGV8CO_R5105_CC機番マスタ)_VN - Excel

File Home Insert Page Layout Formulas Data Review View Tell me what you want to do... Nguyen Hoang Nam (EKB.ERP) Share

F50

Project	経理高度化プロジェクト	Phase	詳細仕様	Business Process/ System	加工費分析/S4	Createdon	2017/12/25	Updatedon	
Deliverable	CDS_View定義書	Deliverable Detail	CC機番マスタ	FormNumber	DD409	Createdby	日立夏見	Updatedby	
• : Item hoàn thành test									
○ : Item chưa hoàn thành test									
Xem bảng mất : Item xem bảng mất									

Testcase

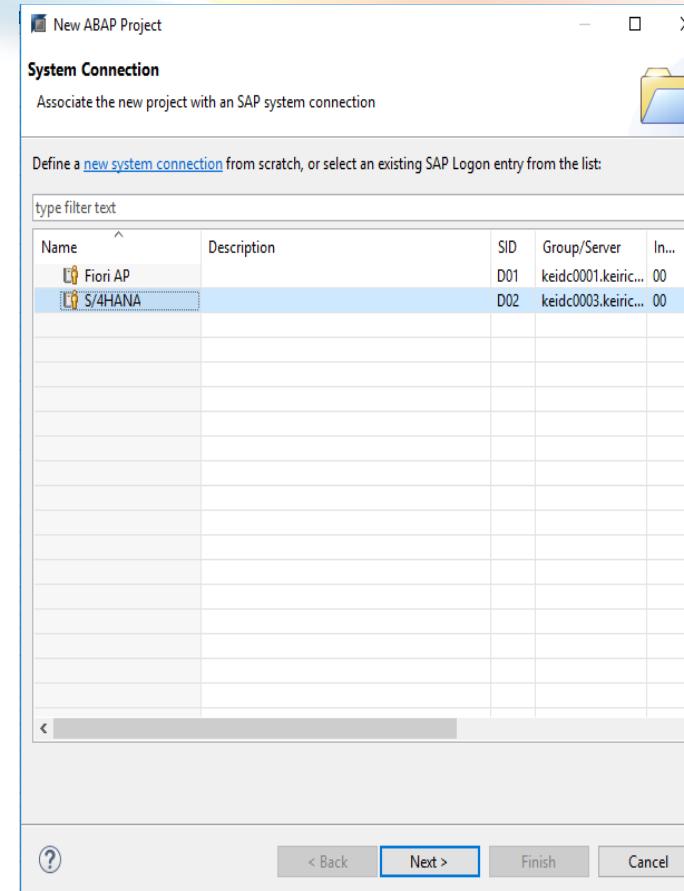
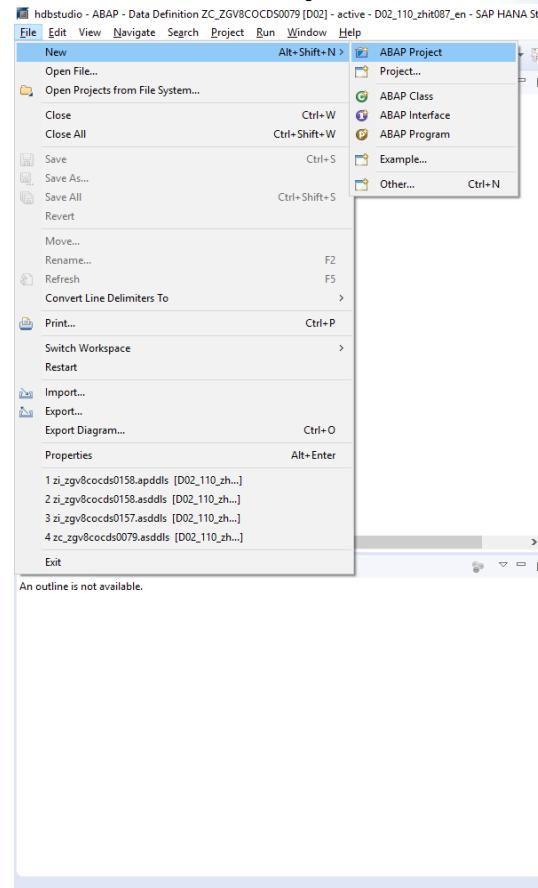
No	Case No	Vị trí source ¹	Vị trí source ²	Điều kiện lọc	Kết quả giả định	Tên sheet kết quả	Kết	Kết	Kết	Kết	Kết	Kết	Kết	Kết	Kết	Kết	Kết	Kết	Kết
						Ngày /Tháng	Ngu ồn thực	Ngu ồn phu	Data1	Data2	Data3	Data4	Data5	Data6	Data7	Data8	Data9	Data10	
14	001	BASIC	(ZI_ZGV8COCDS0158)																
15	1 001			Management area/管理領域	Flag xóa削除フラグ = "														
16	2 002			KOKRS	Key														
17	3 003			Fiscal year/会計年度	Flag xóa削除フラグ = "														
18	4 004			GJAHR	Key														
19	5 005			Fiscal period/会計期間	Flag xóa削除フラグ = "														
20	6 006			RPMAX	Key														
21	7 007			Version/バージョン	Flag xóa削除フラグ = "														
22	8 008			VERSN	Key														
23	9 009			Factory indicator/工場区分	Flag xóa削除フラグ = "														
24	10 010			ZZFACTDIV	Key														
25	11 011			Line code/ラインコード	Flag xóa削除フラグ = "														
26	12 012			ZZLINECDFCC	Key														
27	13 013			Code tài sản/資産コード	Flag xóa削除フラグ = "														
28	14 014			ZZKSISAN	Key														
29	15 015			I_ControllingArea	Điều kiệnASSOCIATION														
30	16 016			ZI_ZGV8COCDS0191	Điều kiệnASSOCIATION														
31	17 017			ZI_ZGV8COCDS0190	Điều kiệnASSOCIATION														
32				Composit	(ZI_ZGV8COCDS0157)														
33	18 018			Management area/管理領域		Phải get được giá trị theo điều kiện từ ZI_ZGV8COCDS0158 với data preview													
34	19 019			KOKRS	Key														

表紙 | 变更履歴 | 仕様書 | 結果 | 不具合一覧 | Header | 仕様書 (サンプル) | +

Ready 80%

SAP HANA Studio configuration

New -> ABAP Project



SAP HANA Studio configuration

- Next-> filled with following infos :

New ABAP Project

Connection Settings

Secure Network Communication (SNC) is disabled. For security reasons, you should enable SNC.

Connection Parameters

System ID: *	D02
Connection Type:	Custom Application Server
Message Server: *	
Group: *	
Message Server Port:	
Application Server: *	keidc0003.keiricloud.mx.toyota.co.jp
Instance Number: *	00
SAProuter String:	

Secure Network Settings

<input type="checkbox"/> Activate Secure Network Communication (SNC)	
SNC Level:	Highest available security level
SNC Name: *	
Single Sign-On (SSO):	Disabled

?

< Back **Next >** Finish Cancel

- System ID : D02
- Application Server : keidc0003.keiricloud.mx.toyota.co.jp
- Instance Number : 00
- Client : 110
- User & Password :
- Language : JA (= Japanese)

New ABAP Project

Logon to System

(i) Specify a value for field 'Password'

System ID: D02

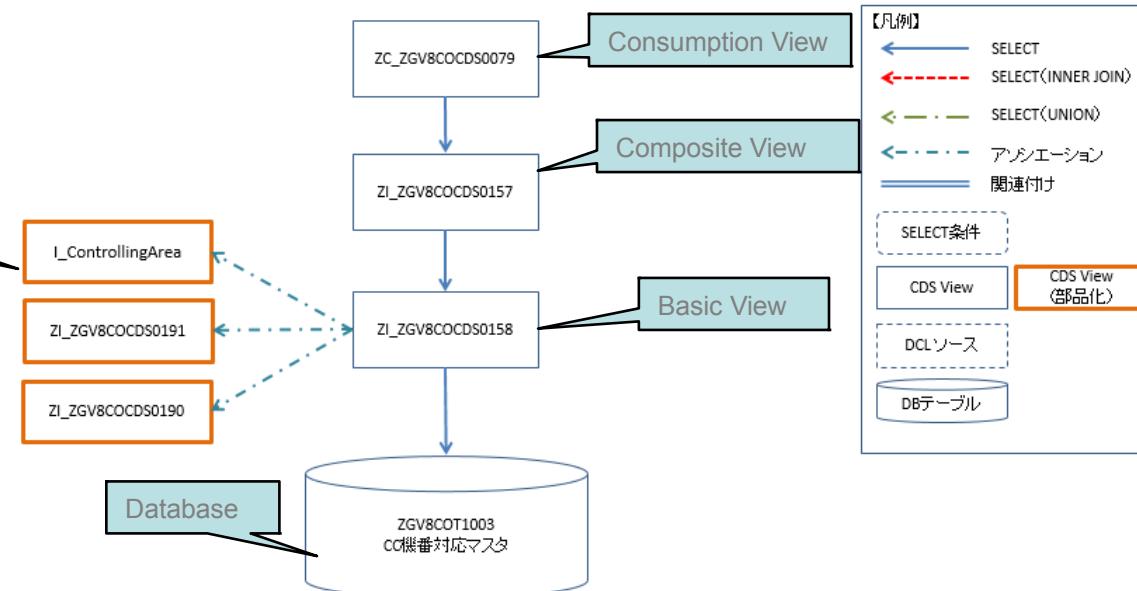
Client: *	110
User: *	ZHIT087
Password: *	
Language: *	JA

?

< Back Next > Finish Cancel

Demo Design of program

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
Project	経理高度化プロジェクト	Phase	詳細仕様	Business Process System	加工費分析/S4	Createdon	2017/11/27 <th>Updatedon</th> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td> <td></td>	Updatedon									
Deliverable	CDS_View定義書	Deliverable Detail	CC機番マスタ	Form Number	ED132	Createdby	日立 島田	Updatedby									



- Following Design

ED132_CDS_View定義書(CC機番マスタ) - Excel

File Home Insert Page Layout Formulas Data Review View Tell me what you want to do... Nguyen Hoang Nam (EKB.ERP) Share

A1

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	X	Y		
Project	経理高度化プロジェクト	Phase	詳細仕様	Business Process/ System	加工費分析/S4	Createdon	2017/11/27	Updatedon																	
Deliverable	CDS_View定義書	DeliverableDetail	CC機番マスタ	FormNumber	ED132	Createdby	日立 島田	Updatedby																	
9	ビューID	ZC_ZGV8COCDSD0079	ビュー名	CC機番マスタ																					
10	ビュータイプ	<input checked="" type="checkbox"/> Consumption <input type="checkbox"/> Interface <input type="checkbox"/> Private <input type="checkbox"/> Extended																							
11	アクセス	<input checked="" type="checkbox"/> 読込みのみ																							
12	No.	ヘッダーテーブル名	既定値																						
13	01	@AbapCataloge.sqlViewName	'ZCZGV8COCDSD0079'																						
14	02	@VDMviewType	#CONSUMPTION																						
15	03	@AccessControl.authorizationCheck	#NOT_REQUIRED																						
16	04	@EndUserTextLabel	'CC機番マスタ'																						
17	05	@odata.publish	TRUE																						
18	06	@Analytics.dataCategory	#CUBE																						
19	No.	ヘッダーテーブル名	既定値																						
20	21	ビューID	ZL_ZGV8COCDSD0157	ビュー名	CC機番マスタ Comp																				
22	ビュータイプ	<input type="checkbox"/> Consumption <input checked="" type="checkbox"/> Interface <input type="checkbox"/> Private <input type="checkbox"/> Extended																							
23	アクセス	<input checked="" type="checkbox"/> 読込みのみ																							
24	No.	ヘッダーテーブル名	既定値																						
25	01	@AbapCataloge.sqlViewName	'ZIZGV8COCDSD0157'																						
26	02	@VDMviewType	#COMPOSITE																						
27	03	@AccessControl.authorizationCheck	#CHECK																						
28	04	@EndUserTextLabel	'CC機番マスタ Comp'																						
29	05	@Analytics.dataCategory	#CUBE																						
30	06	@ClientDependent	TRUE																						
31	No.	ヘッダーテーブル名	既定値																						
32	33	ビューID	ZL_ZGV8COCDSD0158	ビュー名	CC機番マスタ Basic																				
34	ビュータイプ	<input type="checkbox"/> Consumption <input checked="" type="checkbox"/> Interface <input type="checkbox"/> Private <input type="checkbox"/> Extended																							
35	アクセス	<input checked="" type="checkbox"/> 読込みのみ																							
36	No.	ヘッダーテーブル名	既定値																						
37	01	@AbapCataloge.sqlViewName	'ZIZGV8COCDSD0158'																						
38	02	@VDMviewType	#BASE																						
39	03	@AccessControl.authorizationCheck	#NOT_REQUIRED																						
40	04	@EndUserTextLabel	'CC機番マスタ Basic'																						
41	05	@Analytics.dataCategory	#CUBE																						
42	06	@ClientDependent	TRUE																						
43	No.	ヘッダーテーブル名	既定値																						
44																									
45																									
46																									
47																									

変更履歴 | Basic ZL_ZGV8COCDSD0158 | Composit ZL_ZGV8COCDSD0157 | Consumption ZC_ZGV8COCDSD0079 | ビューデザイン | アクション | header | + 100% 97

ED132_CDS_View定義書(CC機番マスタ) - Excel

File Home Insert Page Layout Formulas Data Review View Tell me what you want to do... Nguyen Hoang Nam (EKB.ERP) Share

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	Project	経理高度化プロジェクト	Phase	詳細仕様	Business Process/ System	加工費分析/S4	Createdon	2017/11/27	Updatedon															
2	Deliverable	CDS_View定義書	Deliverable Detail	CC機番マスタ	Form Number	ED132	Createdby	日立 島田	Updatedby															
3	ビューアイド	Z1ZGV8COCD0158	ビュータイプ	CC機番マスタ Basic	Private	Extended				16														
4	アクセス	読み込みのみ																						
5	テーブルID/データ名	結合条件	選択条件																					
6	データソース	リレーデー/項目名/名称	= テーブルID/データ/項目名/名称	AND/OR																				
7	ZGV8COT1003	CC機番対応マスタ	ZGV8COT1003 ZZDELFLG/削除フラグ	= ''																				
8	ASSOCIATION 1	as J_DONTROLLINGAREA \$projection	kokrs	= J_DONTROLLINGAREA ControllingArea																				
9	ASSOCIATION 2	as Z1ZGV8COCD0191 \$projection	kkrs	= Z1ZGV8COCD0191 kkrs																				
10	ASSOCIATION 3	as Z1ZGV8COCD0190 \$projection	versn	= Z1ZGV8COCD0190 versn																				
11	1. ビュー定義	テーブル名/データ名	結合条件	選択条件																				
12	2. ZGV8COCD0190	as Z1ZGV8COCD0190 \$projection	kkrs	= Z1ZGV8COCD0190 kkrs																				
13	3. ZZFACTDIV	as ZGV8COT1003 \$projection	zzfactdiv	= ZGV8COT1003 zzfactdiv																				
14	4. ZZLINEOPFC	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
15	5. ZZKSISAN	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
16	6. ZZFACTDIV	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
17	7. ZZLINEOPFC	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
18	8. ZZKSISAN	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
19	9. ZZFACTDIV	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
20	10. ZZLINEOPFC	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
21	11. ZZKSISAN	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
22	12. ZZFACTDIV	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
23	13. ZZLINEOPFC	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
24	14. ZZKSISAN	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
25	15. ZZFACTDIV	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
26	16. ZZLINEOPFC	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
27	17. ZZKSISAN	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
28	18. ZZFACTDIV	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
29	19. ZZLINEOPFC	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
30	20. ZZKSISAN	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
31	21. ZZFACTDIV	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
32	22. ZZLINEOPFC	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
33	23. ZZKSISAN	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
34	24. ZZFACTDIV	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
35	25. ZZLINEOPFC	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
36	26. ZZKSISAN	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
37	27. ZZFACTDIV	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
38	28. ZZLINEOPFC	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
39	29. ZZKSISAN	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
40	30. ZZFACTDIV	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
41	31. ZZLINEOPFC	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
42	32. ZZKSISAN	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
43	33. ZZFACTDIV	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
44	34. ZZLINEOPFC	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
45	35. ZZKSISAN	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
46	36. ZZFACTDIV	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
47	37. ZZLINEOPFC	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
48	38. ZZKSISAN	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
49	40. ZZFACTDIV	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
50	41. ZZLINEOPFC	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
51	42. ZZKSISAN	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
52	43. ZZFACTDIV	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
53	44. ZZLINEOPFC	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
54	45. ZZKSISAN	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
55	46. ZZFACTDIV	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
56	47. ZZLINEOPFC	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
57	48. ZZKSISAN	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
58	49. ZZFACTDIV	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
59	50. ZZLINEOPFC	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
60	51. ZZKSISAN	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
61	52. ZZFACTDIV	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
62	53. ZZLINEOPFC	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
63	54. ZZKSISAN	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
64	55. ZZFACTDIV	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
65	56. ZZLINEOPFC	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
66	57. ZZKSISAN	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
67	58. ZZFACTDIV	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
68	59. ZZLINEOPFC	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
69	60. ZZKSISAN	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
70	61. ZZFACTDIV	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
71	62. ZZLINEOPFC	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
72	63. ZZKSISAN	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
73	64. ZZFACTDIV	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
74	65. ZZLINEOPFC	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
75	66. ZZKSISAN	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
76	67. ZZFACTDIV	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
77	68. ZZLINEOPFC	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
78	69. ZZKSISAN	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
79	70. ZZFACTDIV	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
80	71. ZZLINEOPFC	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
81	72. ZZKSISAN	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
82	73. ZZFACTDIV	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
83	74. ZZLINEOPFC	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
84	75. ZZKSISAN	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				
85	76. ZZFACTDIV	as ZGV8COT1003 \$projection	char	= ZGV8COT1003 char																				

hbstudio - ABAP - Data Definition ZI_ZGV8COCD50158 [D02] - active - D02_110_zhit087_en - SAP HANA Studio

File Edit Source Code Navigate Search Project Run Window Help

Project Explorer CDS Navigator

D [D02] ZI_ZGV8COCD50158

```

1 @AbapCatalog.sqlViewName: 'ZI_ZGV8COCD50158'
2 @OML.viewType: #BASIC
3 @AccessControl.authorizationCheck: #NOT_REQUIRED
4 @EndUserText.label: 'CC帳番又はBasic'
5 @Analytics.dataCategory: #CUBE
6 @ClientDependent:true
7 /*+hidewarning { "IDS": [ "KEY_CHECK" ] }
8 define view ZI_ZGV8COCD50158
9   as select from zgv8cot1003
10
11   association [0..1] to I_ControlingArea as _I_CONTROLLINGAREA //Controlling Area
12     on $projection.kokrs = _I_CONTROLLINGAREA.ControllingArea
13
14   association [0..*] to ZI_ZGV8COCD50191 as _ZI_ZGV8COCD50191 on $projection.kokrs = _ZI_ZGV8COCD50191.kokrs
15     and $projection.versn = _ZI_ZGV8COCD50191.versn
16
17   association [0..*] to ZI_ZGV8COCD50190 as _ZI_ZGV8COCD50190 on $projection.kokrs = _ZI_ZGV8COCD50190.kokrs
18     and $projection.zzfactdiv = _ZI_ZGV8COCD50190.zzfactdiv
19
20   key kokrs, // 管理領域
21
22   key gjahr, // 会計年度
23
24   key rppmax, // 会計期間
25
26   key versn, // バージョン
27
28   key zzfactdiv, // 工場区分
29
30   zgv8cot1003.zzlinecdfcc, // ラインコード(工場C/C)
31
32   zgv8cot1003.zzsisan, // 貢産コード(機番)
33   /*ASSOCIATION*/
34
35   _I_CONTROLLINGAREA, // Controlling Area
36   _ZI_ZGV8COCD50191, // バージョンをリストする
37   _ZI_ZGV8COCD50190 // 工場区分テキストビュー
38 }
39
40 where
41   zzdelfig = ''

```

Outline

ZI_ZGV8COCD50158

- select
 - from
 - associations
 - select list
 - where

1 2 3 4 5 6

Read-Only Smart Insert 8:29 D02_110_keidc003_ZHIT087_EN

【Sample】DD409_テスト仕様書(単体テスト)(ZGV8CO_RS105_CC機種マスタ) - Excel

File Home Insert Page Layout Formulas Data Review View Tell me what you want to do... Nguyen Hoang Nam (EKB.ERP) Share

BJ54

52 BASIC (ZI_ZGV8COCCDS0158)

53 <入力>

54 [ZI_ZGV8COCCDS0158] ビュー名: CC機種マスタ Basic
ビュータイプ: Consumption Interface Private Extended
読み込みのみ
アクセス:

No.	AbapCatalogEntry.viewName	固定値
01	@VDM.viewType	#BASIC
02	@AccessControl.authorizationCheck	#NOT_REQUIRED
03	@EndUserTextLabel	CC機種マスタ Basic
04	@Analytics.dataCategory	#CUBE
05	@ClientDependent	TRUE

55 [D02] ZI_ZGV8COCCDS0158 3
 1 @AbapCatalog.sqlViewName: 'ZIZGV8COCCDS0158'
 2 @VDM.viewType: #BASIC
 3 @AccessControl.authorizationCheck: #NOT_REQUIRED
 4 @EndUserTextLabel: 'CC機種マスタ Basic.'
 5 @Analytics.dataCategory: #CUBE
 6 @ClientDependent: true
 7 /*+[hideWarning] { "IDS" : ["KEY_CHECK"] } */

56 <結果>

57

58 テーブルID/デ_名称 結合条件 準拠条件
 59 テーブル名/名称 = テーブルID/データ_項目名/名称 テーブルID/データ_項目名/名称 比較値 AND/OR

60 ZGV8COT1003 CC機種対応マスター ZGV8COT1003 ZGV8COT1003 = ''

61 ASSOCIATION(0..1)
 62 ControllingArea as J_DCNTROLLINGAREA \$projection kokrs = J_DCNTROLLINGAREA ControllingArea
 63 ASSOCIATION(0..*)
 64 ZI_ZGV8COCCDS0191 as ZI_ZGV8COCCDS0191 \$projection versn = ZI_ZGV8COCCDS0191 versn
 65 AND
 66 ZI_ZGV8COCCDS0190 as ZI_ZGV8COCCDS0190 \$projection kokrs = ZI_ZGV8COCCDS0190 kokrs
 67 AND
 68 ZI_ZGV8COCCDS0190 \$projection lasteditd = ZI_ZGV8COCCDS0190 lasteditd
 69
 70 No. ビュー項目 テーブルID/データ_項目名/名称 Key データ型 データエレメント 内容説明 備考
 71 1 KOKRS ZGV8COT1003 管理用域 X CHAR
 72 2 GUARH ZGV8COT1003 会計年度 X NUMC
 73 3 RMAX ZGV8COT1003 会計月期 X NUMC
 74 4 LTYPE ZGV8COT1003 会計年月 X DATE
 75 ZZFACTDV ZGV8COT1003 工場区分 X CHAR
 76 ZLINECODEQ ZGV8COT1003 ラインコード(工場/C/C) X CHAR
 77 ZKXSSAN ZGV8COT1003 活用コード機種 X CHAR
 78
 79 ASSOCIATION
 80 8 J_DCNTROLLINGAREA
 81 9 ZI_ZGV8COCCDS0191
 82 10 ZI_ZGV8COCCDS0190
 83
 84 Key 設定されていること
 85 組合せ条件がCDS View定義書と合致すること
 86 8 /*+[hideWarning] { "IDS" : ["KEY_CHECK"] } */
 87 define view ZI_ZGV8COCCDS0158
 88 as select from ZGV8COT1003
 89 association [...] to J_DCNTROLLINGAREA as _J_DCNTROLLINGAREA

106 <結果>

107 Key 設定されていること
 108 組合せ条件がCDS View定義書と合致すること
 109 10 /*+[hideWarning] { "IDS" : ["KEY_CHECK"] } */
 110 define view ZI_ZGV8COCCDS0158
 111 as select from ZGV8COT1003
 112 association [...] to J_DCNTROLLINGAREA as _J_DCNTROLLINGAREA

Ready 85%

- To be continued....



OPEN DISCUSSION



THANK YOU