**Date: 01/14/2021**

**Prepared By : Joseph Reddy Y**

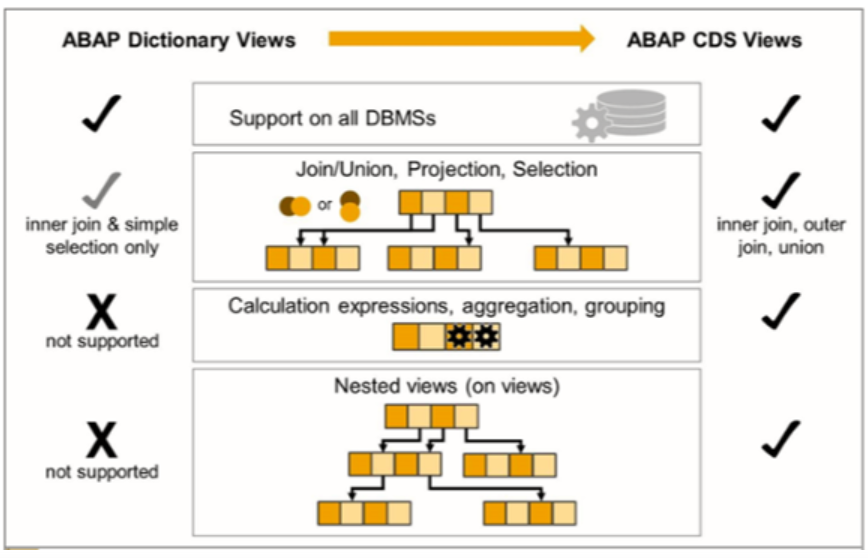
**Purpose:**

Using CDS View based approach to develop the BW Data sources on custom extractors instead of function module based data sources and make use of ABAP CDS views benefits and features.

**Introduction:**

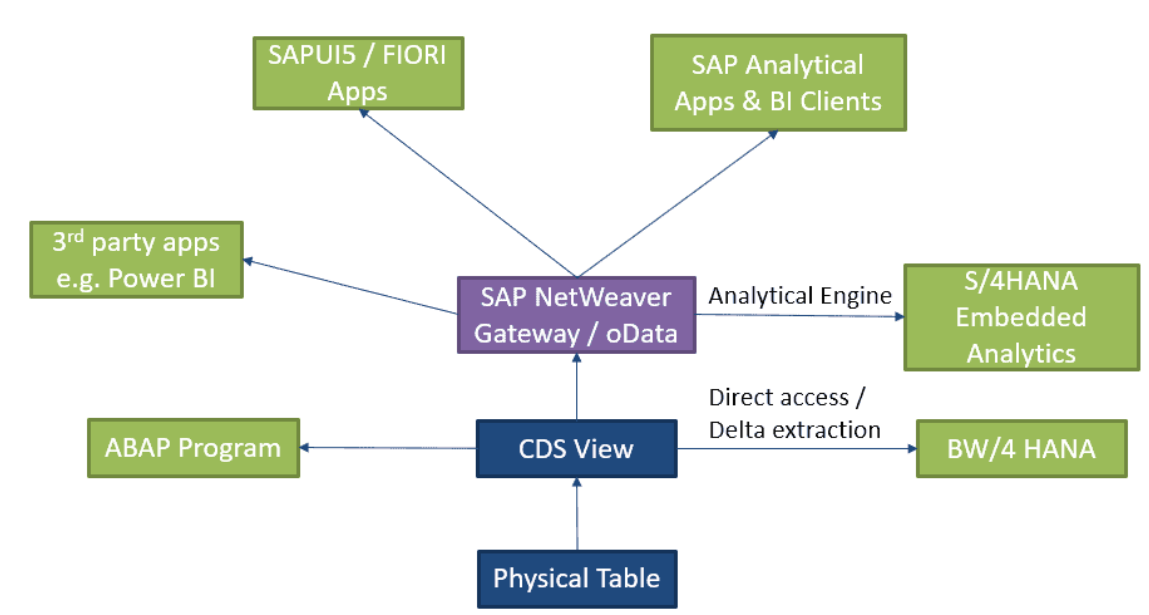
**ABAP Core Data Services** (CDS). The key benefits of the CDS framework are:

* **Many new SQL features:**
  + SQL joins: INNER JOIN, LEFT OUTER JOIN, …
  + SQL set operations: UNION, UNION ALL
  + SELECT clauses: GROUP BY, HAVING
  + Aggregate functions: AVG, MAX, SUM, …
  + Numeric functions: CEIL, MOD, ABS, …
  + String functions: SUBSTRING, LPAD, CONCAT, …
  + Date and time functions: DATS\_DAYS\_BETWEEN, DATS\_ADD\_DAYS, …
* **Easy-to-integrate services on-top:**
  + OData clients
  + SAP Fiori
  + Analytics
* **Simplification of application coding**

****

**Value addition / Benefits:**

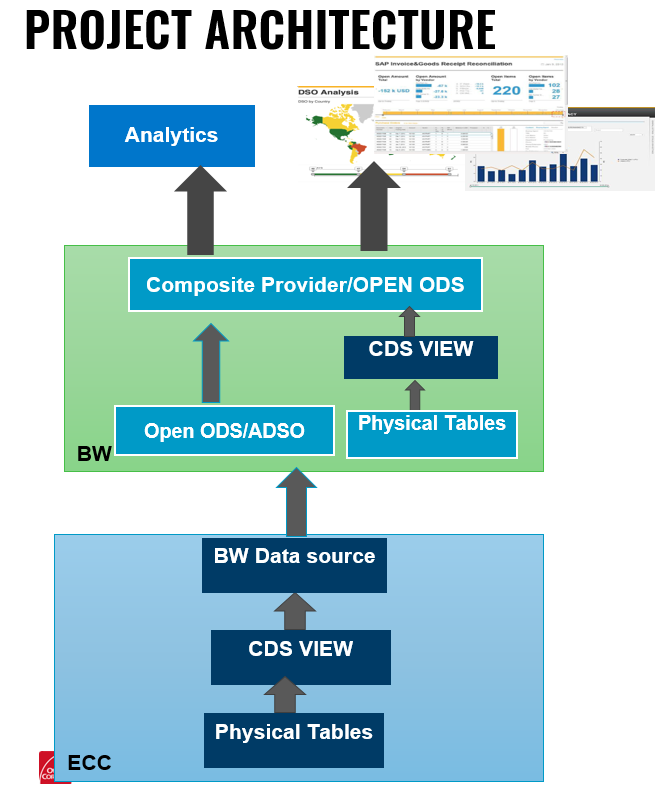
* ABAP CDS Views supports all kinds of joins , set operations, aggregate functions, string, date time functions, currency unit conversion functions. BW Data sources can be built on top of ABAP CDS Views to pull data into BW.
* Easy generation of full-fledged OData Services, Can be used in many places.
* Easy use of SAP UI5 Smart Controls and Fiori Elements, implicit authorization checks, and
* Performance boost, Rich Set of Built-in Functions and Code Pushdown.
* Saves the development time, increases productivity.
* Upward compatibility to S/4, Ready for future embeded analytics based on CDS Views in S/4 HANA.

****

**Project Architecture:**

**Steps:**

1. Develop ABAP CDS Views in ECC.
2. BW Data sources can be built on top of ABAP CDS Views to pull data into BW.
3. Consume them in OPEN ODS for real time reporting.
4. Or Load into ADSO for persistency and consume through composite for reporting.
5. CDS Views can be developed to Join BW Tables and expose for reporting through Open ODS/Composite.



**CDS View creation General guide lines**

1. Use SQL best practices when using CDS views : CDS is about view building. Improve your SQL skills before starting with CDS. Otherwise you will most likely create severe errors. Without SQL knowledge it is possible that you will get bad runtime results.
2. Build entity-relationship models before defining CDS views. Take a careful decision on normalized models vs. redundant storage of data, in particular with respect to avoiding JOIN operations on huge database tables.
3. implement CDS views with different purposes (operational reporting, OData..) in different ABAP packages. One reason for it is to control reuse.
4. Never Use cross joins between tables.
5. Avoid JOIN operations on huge database tables. Analyze the no of records on base tables and final view output , Large data sets will degrades the performance.
6. Avoid cyclic references in the definition of CDS views(to prevent problems during activation).
7. Avoid access to compatibility views for performance-intensive applications (if runtime is critical or if frequently executed).
8. Avoid calculated fields in WHERE-clauses and ON-clauses of CDS views, especially when tables with many entries are involved.
9. Calculated fields result from functions like string functions (CONCAT, RTRIM, ...)and CASE expressions
10. Note that all fields provided by a table function act as calculated. For a calculated field to be used (as filter or join condition) the values of this field for every row must be evaluated at this point of processing. That may be very time-consuming.
11. Avoid constants in ELSE branches within CASE especially for views to be reused in joins as it potentially limits DB
12. optimization due to not preserving the null value. The same limitation applies to fields defined as constant if the view in which they are defined is on the right side of a LEFT OUTER JOIN. Avoid cyclic associations or cyclic joins with involvement of left outer join, especially when tables with many entries are involved.
13. To achieve a certain business purpose, choose the simplest CDS view possible. Do not create the all-in-one complex view covering all demands.
14. Use associations to improve performance: If they are just defined and exposed within the projection list without direct exposition of fields, they are pure metadata; only if used in a path expression they become standard joins.
15. If CDS views contain LEFT OUTER JOIN relationships between tables, use the MANY TO ONE clause if possible.
16. Check if UNION clauses could be safely replaced by UNION ALL without changing the semantics.
17. Prefer UNION [ALL]over CASE expressions in the models as they allow more room for preferable choices for the database optimizer.

**Naming Conventions:**

All the BW relevant CDS views need to be created in the package = Z\_BW\_CDS using ABAP Perspective in eclipse.

**CDS View Name**

**Max 30 characters**

**ZCDS<PREFIX>\_<Description><Type>** — for example, **ZCDSX\_DolphinInvoicesTP**.

Naming rules:

1. **ZCDS** – first 4 letters **registered namespace** of the company
2. **5&6th letter - I (interface view—basic & composite views)**, **C (consumption view)**, **A (remote API view)**,**P (private VDM view)**, **E (extension include view)**, **X (VDM view extension), BW – for BW**
3. Then Underscore **\_**
4. **Name of object – Description first letter in each word Upper case**
5. **Type 🡪 Last letters Vh-(Value Help), Text (Text) ,Hier(Hierarchy),Tp(Transactional),Func(table function),Cube(Cube),Q(Query),AUTH(Access control)**

#### SQL View Name

**Max - 16 characters**

**ZC<PREFIX>\_<DESCRIPTION><Type>** — for example, **ZCBW\_InvoicesTp**

Naming rules:

1. **ZC** – first two letters **registered namespace** of the company
2. **5&6th letter - - I (interface view—basic & composite views)**, **C (consumption view)**, **A (remote API view)**,**P (private VDM view)**, **E (extension include view)**, **X (VDM view extension), BW – for BW**
3. Then Underscore **\_**
4. **Name of object – Description first letter in each word Upper case**
5. **Type 🡪 Last letters Vh-(Value Help), Text (Text) ,Hier(Hierarchy),Tp(Transactional),Func(table function),Cube(Cube),Q(Query),AUTH(Access control)**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **View Type** | **CDS Definition tech name** | **CDS Definition tech name Ex(Max 30 Char)** | **SQL View Techname Ex (Max 16 Char)** | **Comments** |
| Dimension | ZCDSX\_<DimensionName>Dim | ZCDSBWW\_VendorDim | ZCBW\_VendorDim | for Attributes |
| Text | ZCDSX\_<DimensionName>Text | ZCDSBW\_VendorText | ZCBW\_VendorText | for Text |
| Transactional | ZCDSX\_<BusinessObjectName>Tp | ZCDSBW\_DolphinInvoicesTp | ZCBW\_InvoicesTp | for transactional data |
| Hierarchy | ZCDSX\_< DimensionName >Hier | ZCDSBW\_VendorHier | ZCBW\_VendorHier | for Vendor Hierarchy |
| Value Help | ZCDSX\_<DimensionName>Vh | ZCDSBW\_VendorVh | ZCBW\_VendorVh | for Value Help in S/4HANA and HANA DB |
| Table Function | ZCDSX\_<CubeName>Func | ZCDSBW\_VendorFunc | ZCBW\_VendorFunc | for Table Functions in S/4HANA and HANA DB |
| Cube | ZCDSX\_<CubeName>Cube | ZCDSBW\_VendorCube | ZCBW\_VendorCube | for star schema in S/4HANA and HANA DB |
| Query | ZCDSX\_<CubeName>Q | ZCDSBW\_VendorQ | ZCBW\_VendorQ | for query on star schema in S/4HANA and HANA DB |
| Access Control | ZCDSX\_<CubeName>Auth | ZCDSBW\_VendorAuth | ZCBW\_VendorAuth | to control access in ECC |

**Test case:**

|  |  |  |
| --- | --- | --- |
|  |  |  |
| **Data source type** | **CDS** | **Function module** |
| **Data source Name** | Z\_DS\_E6P\_DOL\_INVREC | Z\_DS\_E6P\_PTSAP\_STATUS\_CHANGES |
| **CDS View name** | ZDOL\_INVREC ( Dolphin Invoice reciepts ) | FM Name: ZBI\_DOL\_PTSAP\_INVOICES |
| **DB SQL View name** | ZDOL\_INVRECIEPTS |  |
| **No of records** | 134589 | 134589 |
| **Development time (Technical)** | 15 min | 30 min |
| **Extraction duration** | 1.27m/76.2sec | 1.712m/102.7 sec |
| **System** | extraction from E6V to B9D | extraction from E6V to B9D |
| **Exeecution Start time** | 1/15/2021 9:31 | 1/15/2021 9:321:42 AM |
|  | |  | | --- | |  | | |  | | --- | |  | |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

**References:**

[**https://www.sap.com/documents/2019/01/0e6d5904-367d-0010-87a3-c30de2ffd8ff.html**](https://www.sap.com/documents/2019/01/0e6d5904-367d-0010-87a3-c30de2ffd8ff.html)

[**https://blogs.sap.com/2016/02/01/getting-started-with-abap-core-data-services/**](https://blogs.sap.com/2016/02/01/getting-started-with-abap-core-data-services/)

[**https://inui.io/sap-abap-cds-views-best-practices/**](https://inui.io/sap-abap-cds-views-best-practices/)