International Journal on Science and Technology (IJSAT)

E-ISSN: 2229-7677 ● Website: www.ijsat.org ● Email: editor@ijsat.org

Analytical CDS Views for SAP S/4 HANA Reporting

Kumail Saifuddin Saif

kumail.saif@gmail.com
SAP Technical Architect & Projects Delivery Manager,
Accenture LLP, USA

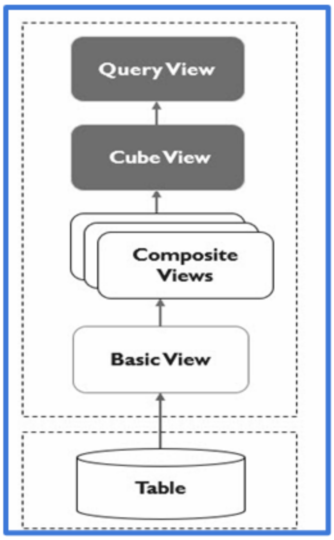
Abstract:

SAP S/4 HANA offers many advanced capabilities over its previous system versions, one of these is Embedded Analytics. The tools like ABAP CDS views, SET hierarchies and real time hierarchy replication i.e., HRRP_REP help to enable real time operational reporting. In this paper we will discuss the two types of ABAP CDS views which are essential for Reporting i.e., Cube views and Query views. Details regarding their place in the overall architecture, key annotations for their development and the structure of these CDS views will be discussed in detail.

Keywords: SAP S/4 HANA, Embedded Analytics, ABAP CDS Views, CDS CUBE, CDS Query.

1 Introduction:

SAP S/4HANA Embedded Analytics leverages Core Data Services (CDS) as a foundation for its Virtual Data Model (VDM). Within this framework, the Cube View and Query View play critical roles in enabling real-time reporting and analytics.



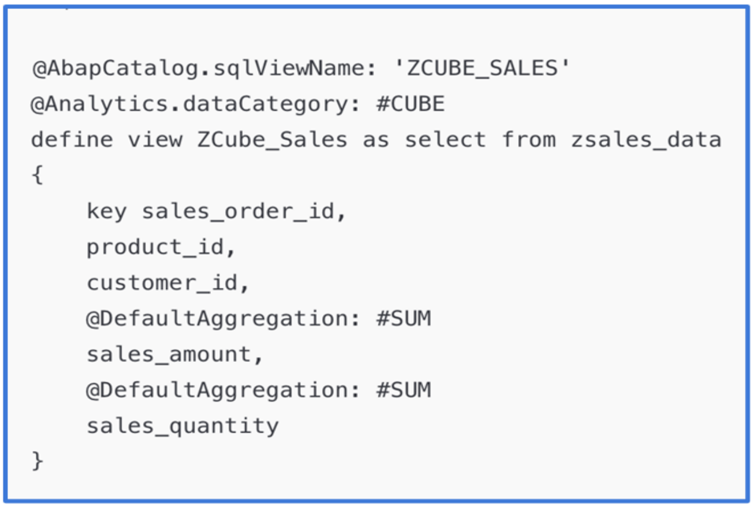
The **Cube View** is a data foundation layer in the VDM that aggregates transactional data into meaningful datasets. It acts as a multidimensional data model (similar to an OLAP cube) for analytics. It is built on top of Basic Views or other foundational CDS views. It prepares data for consumption by Query Views or external tools.

The **Query View** is a consumption layer designed for end-user reporting. It builds on Cube Views to create a reportable dataset with user-friendly formatting, filters, and parameters. It exposes only the fields relevant to the report, hiding technical or unused fields. It provides a ready-to-consume dataset for visualizations in reporting tools such as SAP Analysis for office (AFO) SAP Analytics CLoud (SAC) or SAP Fiori Apps to name a few. SAP provides many CDS views of the type Query and Cube that can be used as data sources to build Reports. However, there are times when you need to create a custom view. Some of the examples listed below:

- The SAP standard CDS view includes only the most commonly used fields and does not include all the fields required for your Business requirement.
- Custom fields may have been added to the SAP S/4HANA source tables and these need to be included in the views.
- You want to simplify the SAP-delivered cubes or queries so that they are easier to work with by removing unwanted fields or changing field descriptions.
- You want to apply fixed filters to the SAP-delivered views so the data is more relevant to the user.
- You want to offer users a prompt to allow them to choose their own filters at run-time.
- You want to create custom calculations as per the Business requirements.

2 Cube View:

The annotation *@Analytics.dataCategory: #CUBE* designates a CDS view as a Cube View. The annotation is applied at the view's header to specify its role within the VDM. This informs the analytical engine of the view's purpose and how it should be processed. Below picture shows the Cube view definition using Analytics.datacategory annotation. As the cube view works as a star schema in the analytics model, it combines the transaction data (Facts) with master Data (Dimensions). This makes it one of the key requirements for a Cube CDS view to have at least one field of the type Fact or key figure. i.e., sales amount in this example. It is also to be noted that the annotation for the aggregation of the fact field is also required. *@DefaultAggregation: #SUM* annotation used in this example denotes that the SUM operation should be performed as aggregation behavior for the key figure sales amount. Other values can be specified such as AVG, MIN, MAX, COUNT, and NONE. Every amount field should also have its currency key field attached to it so that the correct currency key is displayed in the report. Following two annotations help define the same.



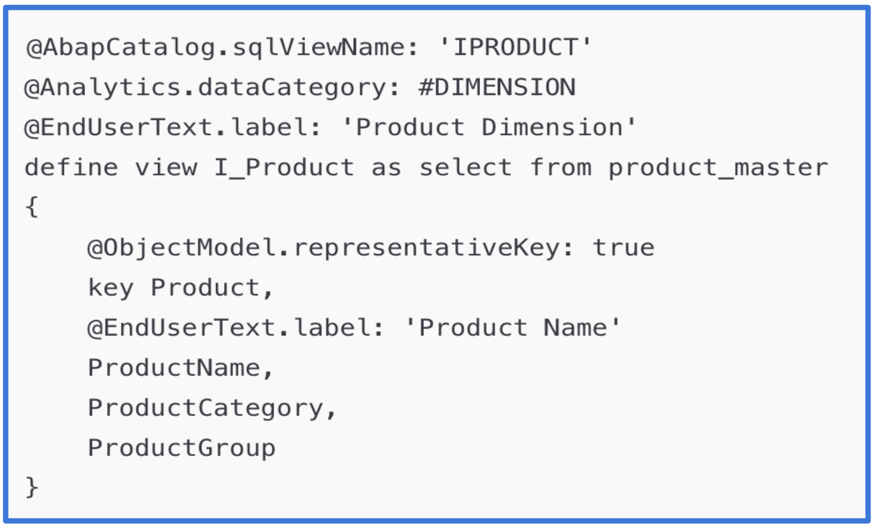
```
@AbapCatalog.sqlViewName: 'ZCUBE_SALES'
@Analytics.dataCategory: #CUBE
define view ZCube_Sales as select from zsales_data
{
    key sales_order_id,
    product_id,
    customer_id,
    @DefaultAggregation: #SUM
    sales_amount,
    @DefaultAggregation: #SUM
    sales_quantity
}
```

- *@Semantics.currencyCode:* This annotation tags a field containing a currency code.
- *@Semantics.amount.currencyCode:* The annotated field contains a monetary amount, and the corresponding currency code is contained in the referenced field.

Similarly, a Quantity field needs a unit of measure defined for its correct interpretation. Following two annotations help define the same.

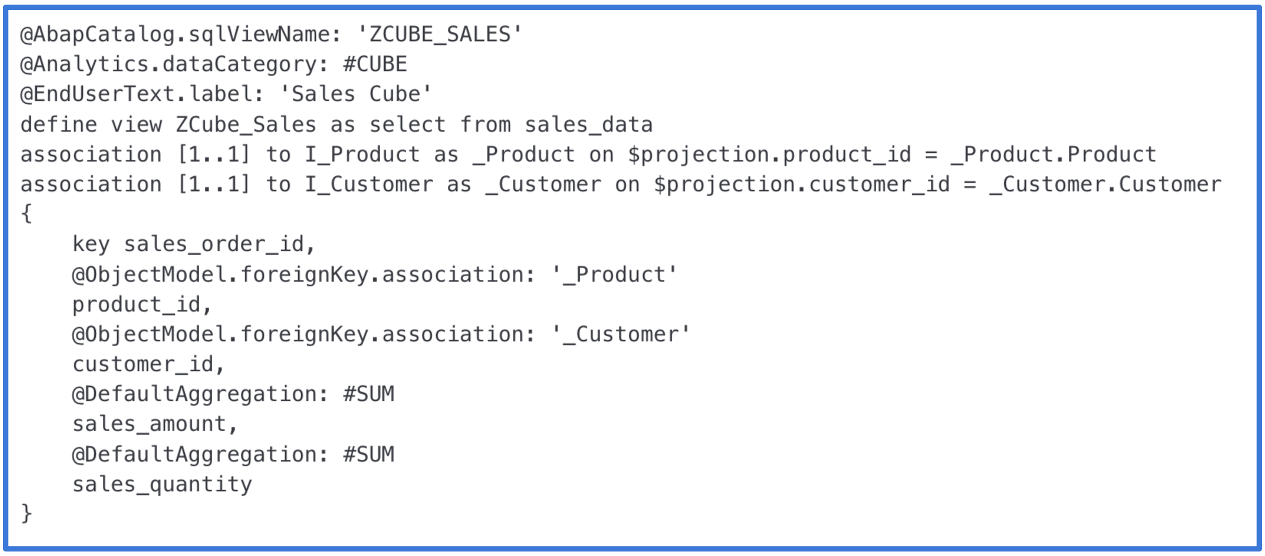
- *@Semantics.unitOfMeasure:* This annotation tags a field containing a unit of measure.
- *@Semantics.quantity.unitOfMeasure:* The annotated field contains a quantity, and the corresponding unit of measure is contained in the referenced field.

Now let us take a look at another important category of the fields in a cube view which is Dimensions. Dimensions classify measures into logical categories for reporting and analysis. Unlike measures, dimensions are not aggregated. They are used as keys or grouping criteria. Dimension fields can have additional details to be added such as Texts, Master data attributes and Hierarchies. These details can be added to the dimension fields by using associations to their corresponding text, master data or Hierarchy CDS views.



```
@AbapCatalog.sqlViewName: 'IPRODUCT'
@Analytics.dataCategory: #DIMENSION
@EndUserText.label: 'Product Dimension'
define view I_Product as select from product_master
{
    @ObjectModel.representativeKey: true
    key Product,
    @EndUserText.label: 'Product Name'
    ProductName,
    ProductCategory,
    ProductGroup
}
```

Dimension CDS views are defined using annotation *@Analytics.dataCategory: #DIMENSION* . Here is an example of the I_Product dimension CDS view used in the ZCube_Sales view.



```
@AbapCatalog.sqlViewName: 'ZCUBE_SALES'
@Analytics.dataCategory: #CUBE
@EndUserText.label: 'Sales Cube'
define view ZCube_Sales as select from sales_data
association [1..1] to I_Product as _Product on $projection.product_id = _Product.Product
association [1..1] to I_Customer as _Customer on $projection.customer_id = _Customer.Customer
{
    key sales_order_id,
    @ObjectModel.foreignKey.association: '_Product'
    product_id,
    @ObjectModel.foreignKey.association: '_Customer'
    customer_id,
    @DefaultAggregation: #SUM
    sales_amount,
    @DefaultAggregation: #SUM
    sales_quantity
}
```

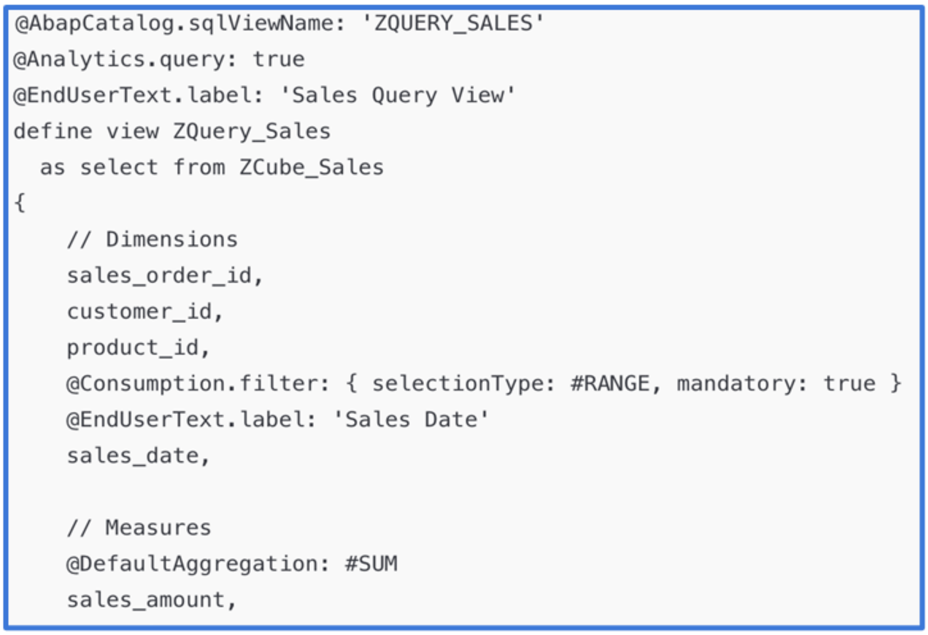
The annotation *ObjectModel.foreignKey.association* defines association to a view that represents a value list/check table of the annotated field. The annotated field must be valued as equal to the annotated representative key field of the target view. A hierarchy view of the dimension can also be associated using this annotation. This way the hierarchy as well as master data of the field are exposed during the query run time. Similar to this we can have text association of the field using annotation *ObjectModel.text.association*. This annotation defines the associated view (annotated with *@ObjectModel.dataCategory: #TEXT*), which provides textual descriptions for the annotated field. Additionally semantic annotation can also be added to a dimension field to provide semantic meaning, such as representing time, units, or currencies. Example *@Semantics.businessDate: true* to be added to a date field.

Read Class in Cube views:

By default, SAP S/4HANA and SAP BW/4HANA use predefined mechanisms to read data from a cube or other data source. However, sometimes you may need custom logic to fetch the data, e.g., for: Data transformation, Complex filtering, and Security restrictions. When you apply the *@ObjectModel.readClass* annotation to a CDS cube view, the system uses the specified ABAP class for reading data instead of directly querying the database tables or views. You define a custom ABAP class with a specific interface that implements the logic to retrieve data. The class is then referenced in the CDS view using the *@ObjectModel.readClass* annotation. This class must implement the required interface (if_sadl_data_provider), which contains methods for reading data from the data source.

3 Query View:

A Query View in SAP ABAP CDS is a consumption layer in the Virtual Data Model (VDM) used for reporting and analytical purposes. Annotation *@Analytics.query: true* is used to define the CDS view as a Query View. This is mandatory for all Query Views. Query views inherit the dimensions, measures and the hierarchies from the cube view. The Query View ZQUERY_SALES in the example provided is a consumption view built on top of the Cube View ZCube_Sales inheriting its associations and aggregations. Some of the used annotations of query view are listed below:

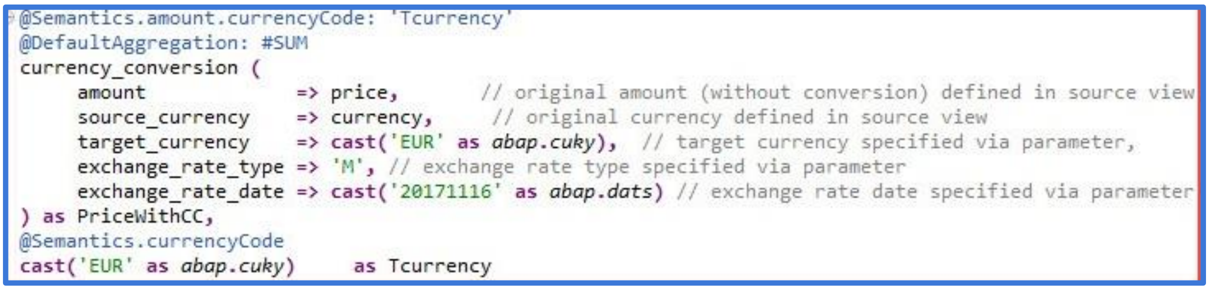


```
@AbapCatalog.sqlViewName: 'ZQUERY_SALES'
@Analytics.query: true
@EndUserText.label: 'Sales Query View'
define view ZQuery_Sales
  as select from ZCube_Sales
{
    // Dimensions
    sales_order_id,
    customer_id,
    product_id,
    @Consumption.filter: { selectionType: #RANGE, mandatory: true }
    @EndUserText.label: 'Sales Date'
    sales_date,

    // Measures
    @DefaultAggregation: #SUM
    sales_amount,
```

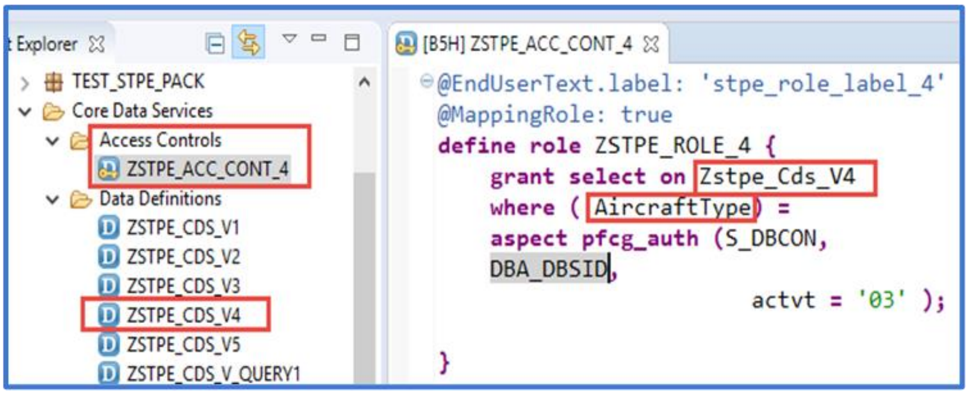
@EndUserText.label: - Provides a user friendly label for the field, such as "Sales Date"
@Consumption.:* - Used for filter definitions and user-specific customizations like mandatory filters, selection ranges, and hierarchies.
@Consumption.dynamicLabel: - Dynamic labels allow for user-friendly customization of field names using dynamic values from fields or calculations.
@AnalyticsDetails.query.formula - Used to perform real-time calculations using the available fields in the CDS view.
@Consumption.sortOrder: Used to define default sort behavior.
@Consumption.filter.defaultValue: It is used to populate the default value for a prompt.
@AnalyticsDetails.query.axis - used to define the default layout of the query fields. Fields can be defined as axis rows, columns, or free meaning not used in default layout.
@OData.Publish: true - Exposes the CDS query to be consumed via OData in tools like Fiori. OData service should be activated via transaction code /iwfnd/maint_service before it can be used.

Currency Conversion - Currency conversion function available in CDS view can help perform the conversion operation as seen in the example below. The function requires amount, source currency, target currency, exchange rate type and date to convert the amount in the required currency.



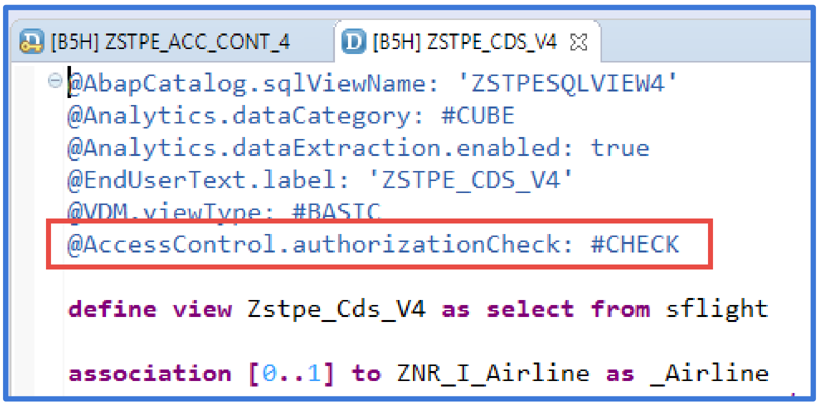
```
@Semantics.amount.currencyCode: 'Tcurrency'
@DefaultAggregation: #SUM
currency_conversion (
     amount            => price,       // original amount (without conversion) defined in source view
     source_currency   => currency,     // original currency defined in source view
     target_currency   => cast('EUR' as abap.cuky),  // target currency specified via parameter,
     exchange_rate_type => 'M', // exchange rate type specified via parameter
     exchange_rate_date => cast('20171116' as abap.dats) // exchange rate date specified via parameter
) as PriceWithCC,
@Semantics.currencyCode
cast('EUR' as abap.cuky)     as Tcurrency
```

Access Control - CDS access controls are based on CDS roles that are defined in Data Control Language (DCL). When CDS entities that are assigned to a CDS role are accessed, additional access conditions are evaluated. In case of an Analytic Query (based on a CDS Cube with access controls) this means that during query runtime, the authorized values are added to the WHERE clause of the SQL statement with a logical AND.



```
@EndUserText.label: 'stpe_role_label_4'
@MappingRole: true
define role ZSTPE_ROLE_4 {
    grant select on Zstpe_Cds_V4
    where ( AircraftType) =
    aspect pfcg_auth (S_DBCON,
    DBA_DBSID,
                        actvt = '03' );

}
```



```
@AbapCatalog.sqlViewName: 'ZSTPESQLVIEW4'
@Analytics.dataCategory: #CUBE
@Analytics.dataExtraction.enabled: true
@EndUserText.label: 'ZSTPE_CDS_V4'
@VDM.viewType: #BASIC
@AccessControl.authorizationCheck: #CHECK

define view Zstpe_Cds_V4 as select from sflight

association [0..1] to ZNR_I_Airline as _Airline
```

Helpful transaction codes to check the CDS Views:

- Transaction RSRTS_ODP_DIS displays transient providers of different contexts e.g. CDS Queries. It can be used to check if the analytic engine is able to interpret the underlying metadata correctly.
- Transient query definition can be checked via transaction RSRTS_QUERY_CHECK.
- CDS authorization can be checked directly using the ABAP report R_ACMTOOL_SELECT
- Transaction SE11, display Database table by input SQLVIEWNAME, the sql view structure can be displayed. Then double click the DDLNAME there, CDS view definition can be displayed.

Conclusion:

SAP S/4HANA Embedded analytics uses CDS within its Virtual Data Model (VDM) to enable real-time reporting through Cube Views and Query Views. Cube Views, designated by @Analytics.dataCategory: #CUBE, serves as the foundational layer, aggregating transactional data with master data (dimensions) into a multidimensional model with facts and measures, while handling semantics for currency and quantity. Query Views, marked by @Analytics.query: true, build on Cube Views as the consumption layer, offering user-friendly datasets with labels, filters, calculations, and dynamic features for reporting tools like SAP Analysis for Office and SAP Analytics Cloud. Custom logic can be applied using read classes, and access controls via DCL ensure secure, role-based data access. Features like currency conversion, dynamic labels, and user prompts make this model adaptable to diverse business needs, supporting tailored and secure real-time analytics.

References

1 - What is SAP HANA? [Online]. Available at: https://www.ibm.com/topics/sap-hana

2 - SAP HANA Installing and administering. SAP TRAINING. [Online]. Available at: https://learning.sap.com/learning-journeys/installing-and-administering-sap-hana

3 - S/4HANA Embedded Analytics [Online]. Available at: https://help.sap.com/docs/SAP_S4HANA_ON-PREMISE/6b356c79dea443c4bbeeaf0865e04207/c53deb5765c7be12e10000000a44450e5.html

4 - Embedded Analytics based on ABAP CDS views [Online]. Available at: https://help.sap.com/docs/SUPPORT_CONTENT/bwplaolap/3361382575.html

5 - Replicate Runtime Hierarchy [Online]. Available at: https://help.sap.com/docs/SAP_S4HANA_ON-PREMISE/5e23dc8fe9be4fd496f8ab556667ea05/d477e555cd3b7b43e10000000a44450e5.html?locale=en-US&version=2020.latest

6 - SAP - ABAP CDS Development User Guide [Online]. Available at: https://help.sap.com/docs/SAP_NETWEAVER_AS_ABAP_752/f2e545608079437ab165c105649b89db/7c078765ec6d4e6b88b71bdaf8a2bd9f.html

7 - VDM Annotations [Online]. Available at: https://help.sap.com/doc/saphelp_nw75/7.5.5/en-US/ef/e9c80fc6ba4db692e08340c9151a17/content.htm?no_cache=true

8 - Financial Statement Versions [Online]. Available at: https://help.sap.com/docs/SAP_S4HANA_ON-PREMISE/651d8af3ea974ad1a4d74449122c620e/c064c2531bb9b44ce10000000a174cb4.html?locale=en-US&version=2020.latest

9 - Analytical Queries [Online]. Available at: https://help.sap.com/docs/abap-cloud/abap-data-models/cds-analytical-queries

10 - Virtual Data Model and CDS Views in SAP S/4HANA [Online]. Available at: https://help.sap.com/docs/SAP_S4HANA_ON-PREMISE/ee6ff9b281d8448f96b4fe6c89f2bdc8/8573b810511948c8a99c0672abc159aa.html

11 - Discovering SAP S/4HANA embedded analytics [Online]. Available at: https://learning.sap.com/learning-journeys/discovering-sap-s-4hana-embedded-analytics