



Deepa Rawat

# Practical Guide to Advanced DSOs in SAP®

- ▶ Fundamentals of Advanced Data Store Objects
- ▶ Modeling with SAP HANA Studio
- ▶ How to create an SAP HANA-optimized InfoProvider
- ▶ Examples and screenshots based on real-world scenarios

# Table of Contents

<b>Preface</b>	<b>7</b>
<b>1 Data warehousing and database technologies</b>	<b>9</b>
1.1 Starflake schema vs. snowflake schema	11
1.2 Relational databases and SAP HANA	12
1.3 SAP BW on SAP HANA	13
1.4 LSA and LSA++ architecture	15
<b>2 New objects in SAP BW and SAP HANA</b>	<b>19</b>
2.1 Open ODS views	19
2.2 Composite providers	28
2.3 Advanced DSOs	36
<b>3 Modeling with SAP HANA Studio</b>	<b>37</b>
3.1 SAP BW modeling perspective	37
<b>4 What is an Advanced DSO?</b>	<b>43</b>
4.1 Advanced DSO tables	44
4.2 Modeling properties	45
<b>5 Model template—data acquisition layer—including corporate memory</b>	<b>51</b>
<b>6 Model template—corporate memory—compression capabilities</b>	<b>61</b>
<b>7 Model template—corporate memory—reporting capabilities</b>	<b>73</b>
<b>8 Model template—data warehouse layer—delta calculation</b>	<b>85</b>
<b>9 Model template—data warehouse layer—datamart</b>	<b>95</b>

<b>10</b>	<b>Some more features of Advanced DSOs</b>	<b>105</b>
10.1	Field- and InfoObject-based modeling	105
10.2	Number of key fields	108
10.3	Remodeling with Advanced DSOs	108
10.4	Enhanced request management	108
10.5	Indexes and partitioning	113
10.6	Navigational attributes	114
10.7	aDSO lookup in transformation rules	114
10.8	External SAP HANA view	115
10.9	TCODE RSOADSO	115
<b>11</b>	<b>Comparison of classic DSO with Advanced DSO</b>	<b>119</b>
<b>12</b>	<b>Business example 1—analysis of customer spending pattern</b>	<b>121</b>
<b>13</b>	<b>Business example 2—actual customer spend compared with plan data</b>	<b>155</b>
<b>14</b>	<b>Some more features within SAP BW on HANA</b>	<b>171</b>
<b>15</b>	<b>Appendix</b>	<b>173</b>
15.1	File format	173
15.2	Customer data	173
15.3	Product data	173
15.4	Actuals data for orders	174
15.5	Plan data for advertisement spend	175
<b>A</b>	<b>References</b>	<b>178</b>
<b>B</b>	<b>The Author</b>	<b>179</b>
<b>C</b>	<b>Index</b>	<b>180</b>
<b>D</b>	<b>Disclaimer</b>	<b>182</b>

## 2 New objects in SAP BW and SAP HANA

Creating a data warehouse using SAP BW takes time because of InfoObject-based modeling technologies. In order to extract data into the older version of SAP BW (version 3.x or before), every source field that needed extraction into SAP BW had to be converted into an InfoObject in SAP BW before a simple report could be created. With SAP BW 7.4 and SAP BW 7.5 on SAP HANA, SAP has introduced field-based modeling and has brought in robust virtual objects. You can now implement reportable objects that do not need InfoObjects. This has changed the speed with which a data warehouse can be implemented.

Some of the new features that have been introduced with SAP BW 7.5 on SAP HANA are:

- ▶ Open ODS views
- ▶ Composite providers
- ▶ Advanced DSOs

There are many more new features, but these are the only objects discussed in this book.

### 2.1 Open ODS views

In SAP BW, modelers create reports that are meant to return results in a short amount of time. To do this, extensive modeling efforts are required to produce the reports. These modeling efforts require creating complex data models in SAP BW, which are time-consuming. Regular SAP BW modeling requires creating InfoObjects of type characteristics and key figures that are in turn used in InfoProviders and queries to return results.

Open ODS view was introduced when SAP released SAP BW 7.4 SP5 on SAP HANA. An Open ODS view allows SAP to integrate external tables and views (on external databases) with SAP BW's transaction and master data by utilizing a view concept. As this interface is a view-based interface, it means that an Open ODS view allows external data to be integrated within SAP BW without data persistency. An Open ODS view can provide three types of semantics within SAP BW. It can enable the external table structures within SAP BW as facts, attributes, or texts. Open ODS views help deliver reports in a shorter amount of time. Unlike other InfoProviders in SAP BW (classic DSOs, InfoCubes, etc.), they do not require InfoObjects for their creation; instead, they use fields for defining the construct of the view. Hence, they are field-based structures. The fields usually adopt the structure of the source table's fields. The field's properties can indicate the following:

- ▶ Whether the field is a key figure or characteristic
- ▶ If the field is a key figure, the aggregation behavior of the key figure
- ▶ Authorization relevance of the field

### Open ODS view restriction



Open ODS view can be used to indicate if a field contains a key figure or a characteristic, but it cannot be used to define hierarchies. As an SAP BW system uses a BW-specific internal format to specify a hierarchy structure, it is not possible to use an Open ODS view to identify a hierarchy.

A field can be associated with either an InfoObject or another Open ODS view. If a field is associated with another InfoObject, it can inherit all the attributes and text for that InfoObject. Similarly, if a field is associated with another Open ODS view, it can utilize the data of that other Open ODS view to enrich itself.

Once the Open ODS view is created, queries can be created to enable reporting.

An Open ODS view can accept data from four different types of sources.

## 2.1.1 Sources for Open ODS views

- ▶ **Datasources (BW)**—A datasource on a table helps expose an external table in a database to SAP BW’s Open ODS view. Datasources can be created via RSO2. If data needs to be extracted from a table, expose the table via a datasource. Some of the supported source system types are BW, SAP, ODP extraction technique, and DBCONNECT.
- ▶ **Database table or view**—Data in tables or views in any schema within SAP HANA can be viewed using this option.
- ▶ **Advanced Data Store Object (aDSO)**—Usually aDSOs can be exposed directly to SAP BW queries. The advantage of having an aDSO as a source to Open ODS views is to use a transaction datasource as master data or the other way around. All reporting takes place using the active table of the Advanced DSO.
- ▶ **Transformation**—A source table is usually interfaced directly to an Open ODS view, but in some situations, it might be required that some fields undergo some transformations before being exposed to the Open ODS view. In that case, it might be required that the source for an Open ODS view be a transformation attached to the datasource.

### Open ODS view availability



Open ODS view is available only on SAP BW systems running on SAP HANA databases.

## 2.1.2 Transporting Open ODS views

Transports for Open ODS view uses SAP’s TLOGO framework. The objects are transportable and the transportable object type is either FBPA or FBPD (SAP-delivered object). In order to transport the Open ODS view, all dependent objects should be available in the target system. If the Open ODS view is created on a source table, that source

table and its datasource should reside in the target system. The Open ODS view does not hold the connection information. The view is connected to the source system and the RSLOGSYSMAP table in the target system usually holds the source-to-target-system mapping details so that the transport can correctly recognize the target system and the table to which the Open ODS view is connected.

### 2.1.3 Advantages and disadvantages of Open ODS views

#### Advantages

1. Open ODS view is a view on the source table structures, so it does not persist data
2. Instead of using InfoObjects, Open ODS view utilizes field-based constructs. This helps a developer to quickly implement queries and do a quicker proof of concept.
3. Open ODS view can connect a field's definition to either an InfoObject or another Open ODS view. This enables a field to utilize an existing BW definition to enrich queries.

#### Disadvantage

1. Open ODS view should be created on smaller-sized source tables. Query performance gets degraded if the source table size is big.

### 2.1.4 An example utilizing Open ODS views

The next example shows how to do a quick proof of concept for an Open ODS view. The example utilizes the source system to be a datasource (BW). Custom `zzcus_reg` table has been created in SAP. It stores CUSTOMER ID, CUSTOMER STATE, and CUSTOMER REGION. Create an Open ODS view on this table and expose it to create a query on the basic table without creating InfoObjects.

1. Display the `zzcus_reg` Table—In order to display the `zzcus_reg` table, use the SE16 transaction code (TCODE), as shown in Figure 2.1.

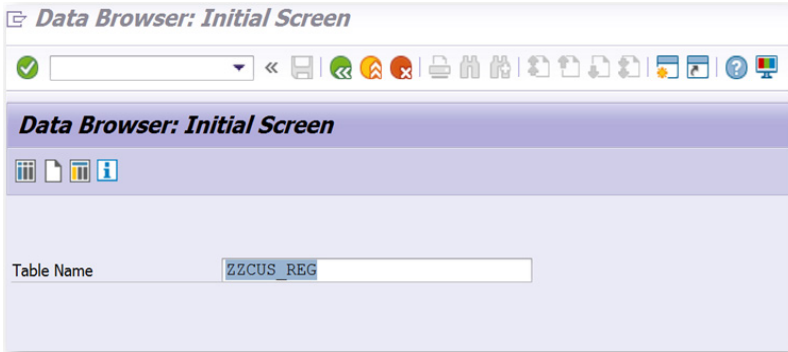


Figure 2.1: Table name being displayed via SE16

- When the data browser screen displays, click the NUMBER OF ENTRIES button and check how many rows are available. In this example, there are 12 rows in the table as shown in Figure 2.2.

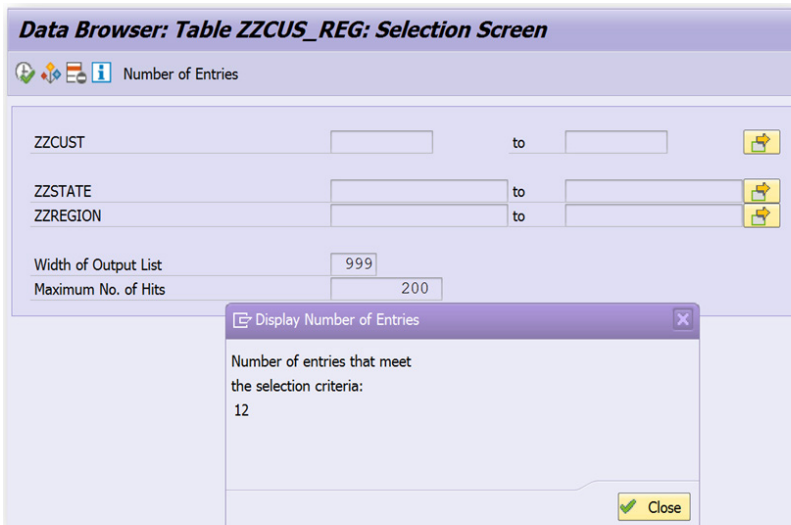


Figure 2.2: Display number of entries in a table

- Execute the TCODE RSO2 and create a datasource on the table zzcus\_reg. In this example, the datasource name is ZZCUS\_REG\_ATTR as shown in Figure 2.3.



# C Index

## A

- Advanced DSO 43
  - Comparison 119
  - Corporate Memory—
    - Compression Capabilities 61
  - Corporate Memory—Reporting Capabilities 73
  - Data Acquisition Layer—
    - including corporate memory 51
  - Data warehouse layer—data mart 95
  - Data warehouse layer—delta calculation 85
  - Enhanced Request Management 108
  - Example 121
  - Indexes and Partitioning 113
  - Model Template 47
  - Modelling Properties 45
  - Remodelling 108
  - Tables 44

## B

- Business transformation Layer 17

## C

- Column-oriented 13
- Composite Providers 28
  - Example 31
  - Inner Join 29
  - Outer Join 29
  - Scenarios 30
  - Supported Providers 29
- Corporate Memory 17

## D

- Data Acquisition 16
- Data Propagation Layer 16
- Data Warehousing 9
- Database 12
- Dynamic Tiering 13

## F

- Field-based Modelling 19, 105

## H

- Hot 13

## I

- InfoObject-based Modelling 19
- In-memory 13

## L

- LSA 15
- LSA++ 17

## M

- Modelling Perspective 37
  - BW Reporting Preview 41
  - History View 41
  - Master Data Maintenance 41
  - Problems View 41
  - Project View 38
  - Properties View 40

## O

- OLAP 9, 10
- OLTAP 12
- OLTP 9, 10

Open ODS Views 19  
  Example 22  
  Sources 21  
  Transporting 21  
Operational DataStore Layer 17

## **Q**

Quality and Harmonization  
  Layer 16

## **R**

Reporting Layer 17

## **S**

SAP BW 10  
  Classic DSOs 15  
  Composite Provider 15  
  Hybrid Provider 15

InfoCube 15  
InfoObject 14  
InfoSet 15  
MultiProvider 15  
PSA 15  
  Virtual Provider 15  
SAP HANA 12  
SAP HANA Studio 37  
Snowflake Schema 12  
SQL 12  
Starflake Schema 11

## **V**

Virtualization Layer 17

## **W**

Warm 13