



## SAP BW on HANA 7.5 (BW310H)

Data Warehousing with SAP  
Business Warehouse powered by  
SAP HANA

People matter, results count.

# Agenda

- **Introduction to SAP HANA Architecture**
- **SAP Business Warehouse (BW)**
- **SAP HANA Studio**
- **Data Warehousing Workbench**
- **Info Objects like Characteristics and Key Figures using Eclipse**
- **Master Data Loading from SAP source system with Operational Data Provisioning .**
- **Generic Data Sources**
- **Transformation and Data Transfer Process**
- **Difference between Classic and Graphical Data Flow Modeling**
- **Modeling Data Store Object (Advanced)**
- **Transactional Data Loading from SAP source system with Operational Data Provisioning**
- **Data Loading from Flat file**
- **Modeling Composite Providers**
- **HANA Native Modeling with Calculation Views**
- **Modeling Open ODS Views**
- **Process Chains**

# Introduction to HANA

SAP HANA is a combination of HANA Database, Data Modeling, HANA Administration and Data Provisioning in one single suite. In SAP HANA, HANA stands for High-Performance Analytic Appliance.

The main features of SAP HANA are given below :

SAP HANA is a combination of software and hardware innovation to process huge amount of real time data.

- Based on multi core architecture in distributed system environment.
- Based on row and column type of data-storage in database.
- Used extensively in Memory Computing Engine (IMCE) to process and analyze massive amount of real time data.
- It reduces cost of ownership, increases application performance, enables new applications to run on real time environment that were not possible before.

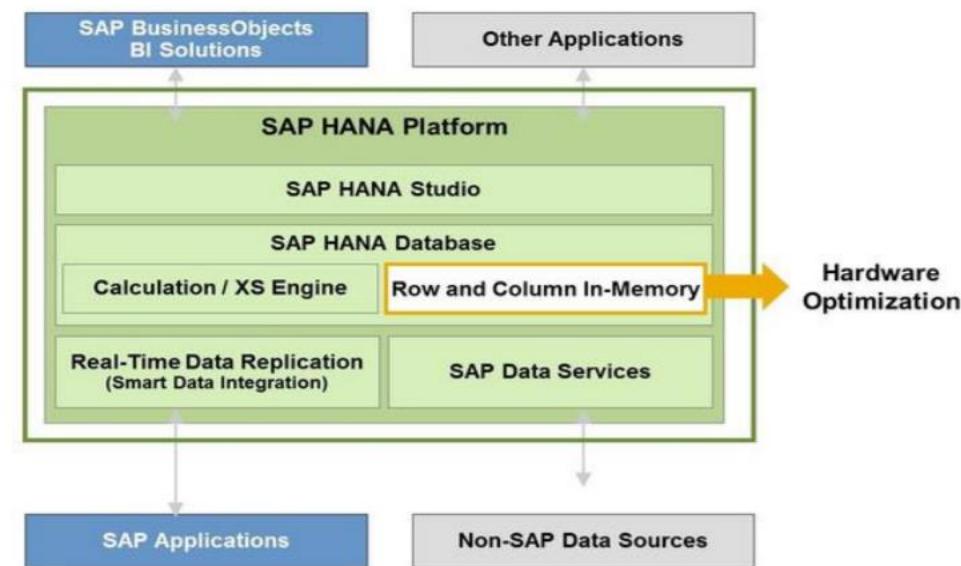
# SAP HANA Architecture

SAP HANA is an in-memory data platform that is deployable on premise or on demand and is useful as it's very fast due to all data loaded in-Memory and no need to load data from disk.

SAP HANA can be used for the purpose of OLAP (On-line analytic) and OLTP (On-Line Transaction) on a single database. SAP HANA Database consists of a set of in-memory processing engines.

**Calculation engine** is main in-memory Processing engines in SAP HANA. It works with other processing engine like Relational database Engine(Row and Column engine), OLAP Engine, etc.

**XS engine** It allows external application and developers to use SAP HANA database via the XS Engine client. The external client application can use HTTP to transmit data via XS engine for HTTP server.



# Agenda

- Introduction to SAP HANA Architecture
- SAP Business Warehouse (BW)
- SAP HANA Studio
- Data Warehousing Workbench
- Info Objects like Characteristics and Key Figures using Eclipse
- Master Data Loading from SAP source system with Operational Data Provisioning .
- Generic Data Sources
- Transformation and Data Transfer Process
- Difference between Classic and Graphical Data Flow Modeling
- Modeling Data Store Object (Advanced)
- Transactional Data Loading from SAP source system with Operational Data Provisioning
- Data Loading from Flat file
- Modeling Composite Providers
- HANA Native Modeling with Calculation Views
- Modeling Open ODS Views
- Process Chains

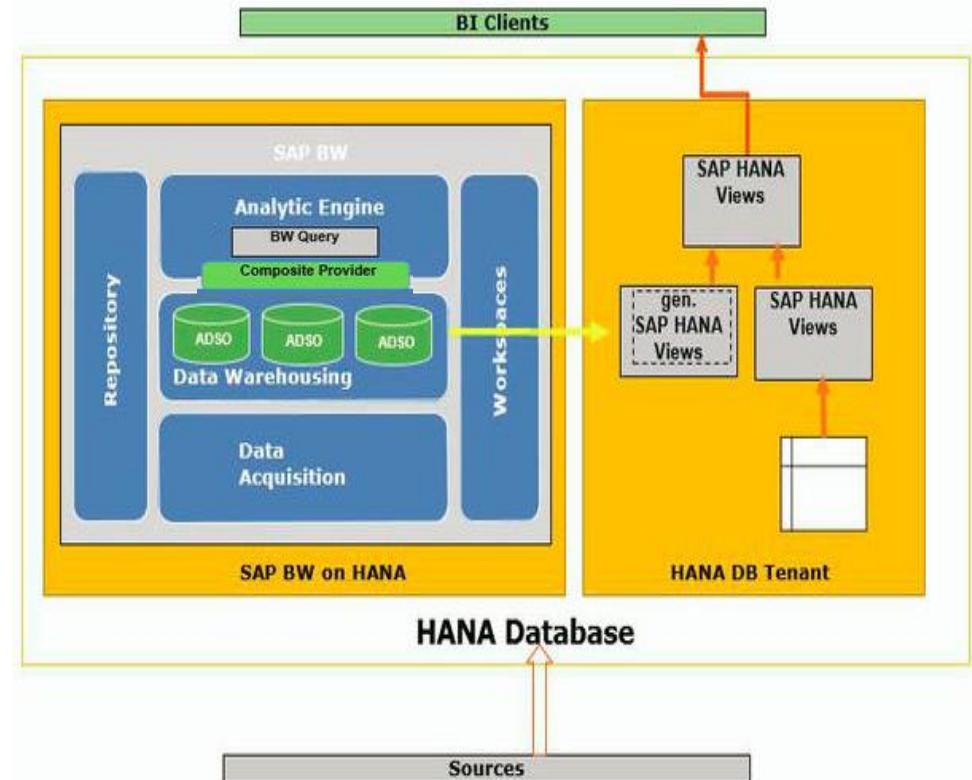
# Introduction to SAP BW Application on SAP HANA

SAP HANA is used as a database for a SAP system (non-BW), or even for a 3rd party development, like your fancy e-commerce system or your custom-made data warehouse.

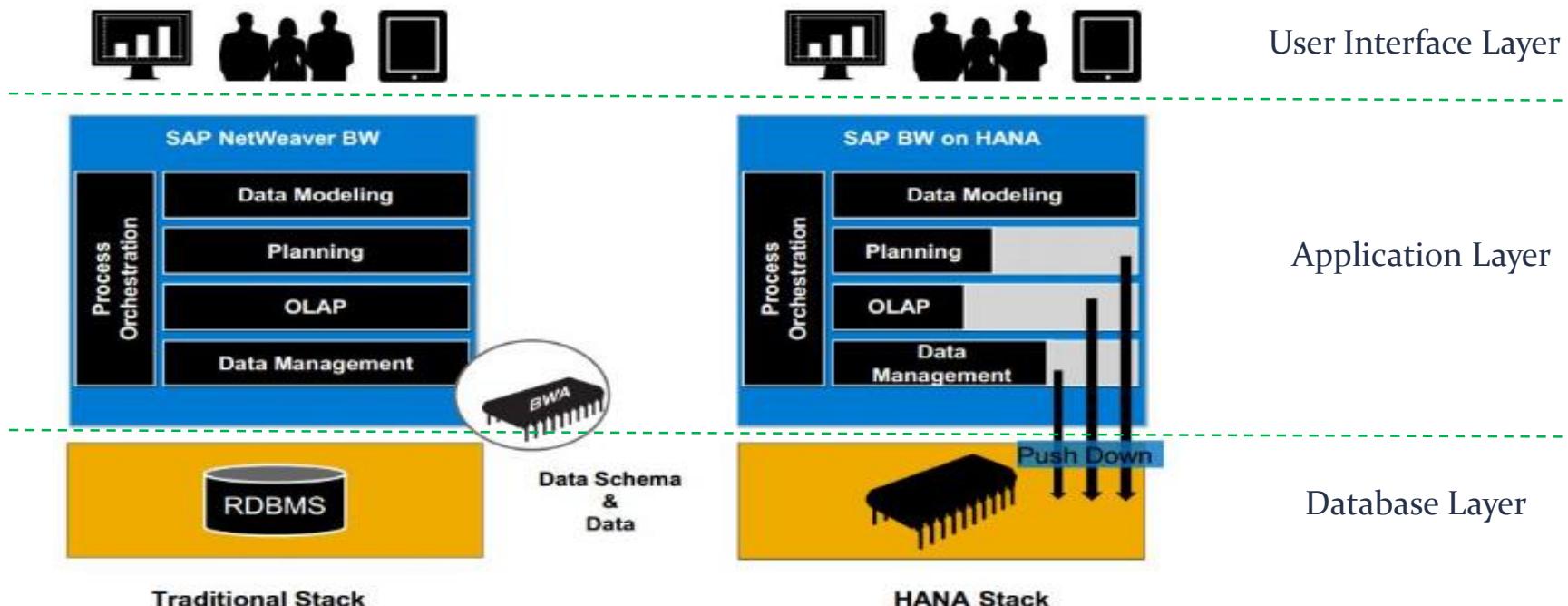
SAP BW, on the other hand, is a full-grown BI system, that includes tools for data modelling, ETL, data load management, OLAP engine, data mining, authorization and so on. SAP BW can work on different databases (Oracle, SQL Server) and also on HANA as underlying database.

SAP Business Warehouse (BW) on HANA combines the power of both the tools - BW Modeling and HANA in-memory computing engine to process huge amounts of data.

It helps you speed up data analysis by consuming data via a Data Warehouse (DW) for analytical reporting and data analysis. You can achieve key opportunities like real-time data integration and data modeling, and hence real-time BI reporting on large amount of data in the database.



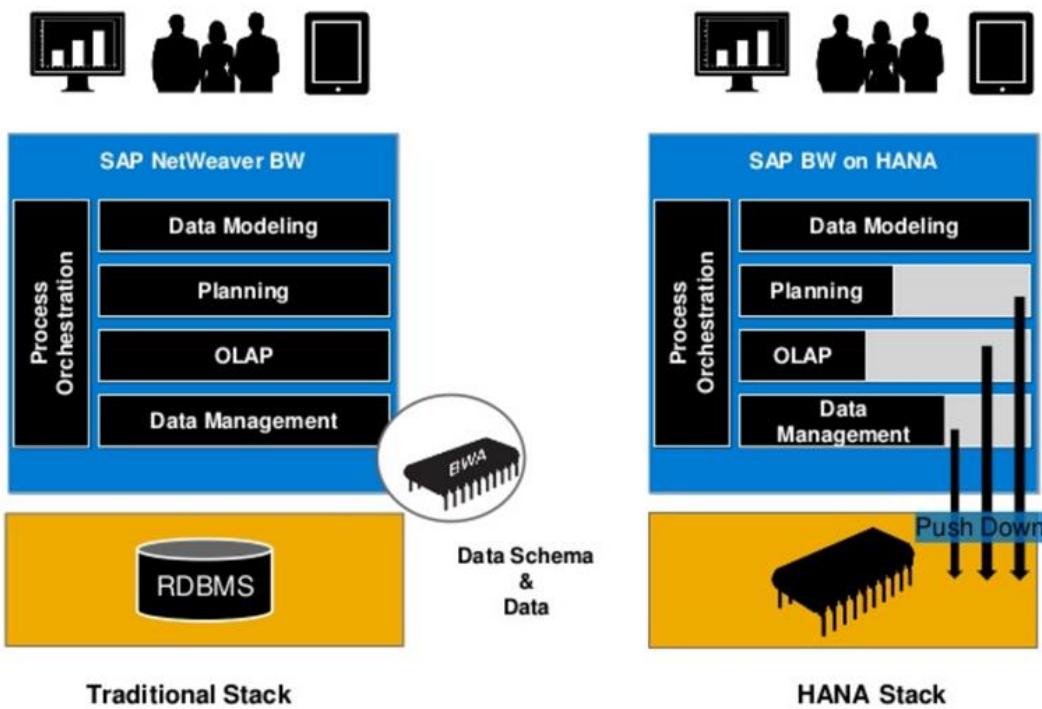
# SAP Netweaver BW Powered by SAP HANA



## HANA as Primary database for BW and Foundation for New application

1. HANA acts as in memory database used as a primary persistence for BW
2. BW cubes and DSOs are loaded and activated in HANA to dramatically improve Performance
3. BW continues to manage the analytic metadata and data provisioning processes.
4. Database migration Only! The BW application remains Intact
5. IT no longer needs to create and maintain aggregates, Indexes or spend valuable time on complex Data modelling in BW

# How Does BW running on RDBMS differ from BW running on HANA



## Customer Values of SAP BW powered by SAP HANA

- Excellent query performance for improved decision making
- Performance Boost for Data load processes for decreased data latency
- Accelerated In-memory planning capabilities for faster planning scenarios
- Flexible combine EDW with HANA Native data for real time insights and decision making
- Data persistency layers are cutoff and reduced administration efforts
- Simplified data modelling and remodeling

**Data intensive function are pushed down from BW to HANA layer.**

# SAP BW on HANA – Benefits

1. Excellent performance in analytical reporting and data loading using HANA in memory database capabilities. All BW functions performed in SAP HANA benefits from in-memory database and calculation engines for faster data processing.
2. With HANA optimized objects, you can perform complex queries, detailed analysis, high data volume, and aggregations efficiently.
3. All existing BI tools such as BEx, Business Objects BI reporting tools, and Microsoft Excel are directly supported by SAP BW on HANA.
4. SAP HANA provides high level of data compression. Column storage of tables requires less storage type and hence provides lower Total Cost of Ownership (TCO).
5. Business Warehouse Accelerator (BWA) is not required while using HANA underneath BW.
6. You don't need aggregated tables and HANA supports on-the-fly aggregations.
7. It has simplified data modeling by using in-memory-optimized objects. There is no need to load BWA index.
8. When you use SAP BW on HANA, the following processes are not required –
  - *Rolling Up Filled Aggregates*
  - *Filling of New aggregates*
  - *Adjust Time-Dependent Aggregates*
  - *Construct Database Statistics*
  - *Build Index*
  - *Delete Index*

# SAP BW on HANA – Architecture Column Vs Row Store

## Stores in SAP HANA

### Column Store



### Row Store



A	B	C
A1	B1	C1
A2	B2	C2
A3	B3	C3

In a row-based layout each row follows the next:

A1	B1	C1	A2	B2	C2	A3	B3	C3
----	----	----	----	----	----	----	----	----

While for a column-oriented layout it stores one column after the next:

A1	A2	A3	B1	B2	B3	C1	C2	C3
----	----	----	----	----	----	----	----	----

## Column Store (Advantages)

The table is searched based on values of a few columns

Calculations are typically executed on single or a few columns only

The table has a large number of columns

The table has a large number of rows and columnar operations are required

High compression rates can be achieved because the majority of the columns contain only few distinct values (compared to number of rows)

## Row Store (Advantages)

The application needs to only process a single record at one time (many selects and/or updates of single records)

The application typically needs to access a complete record (or row)

The columns contain mainly distinct values so that the compression rate would be low

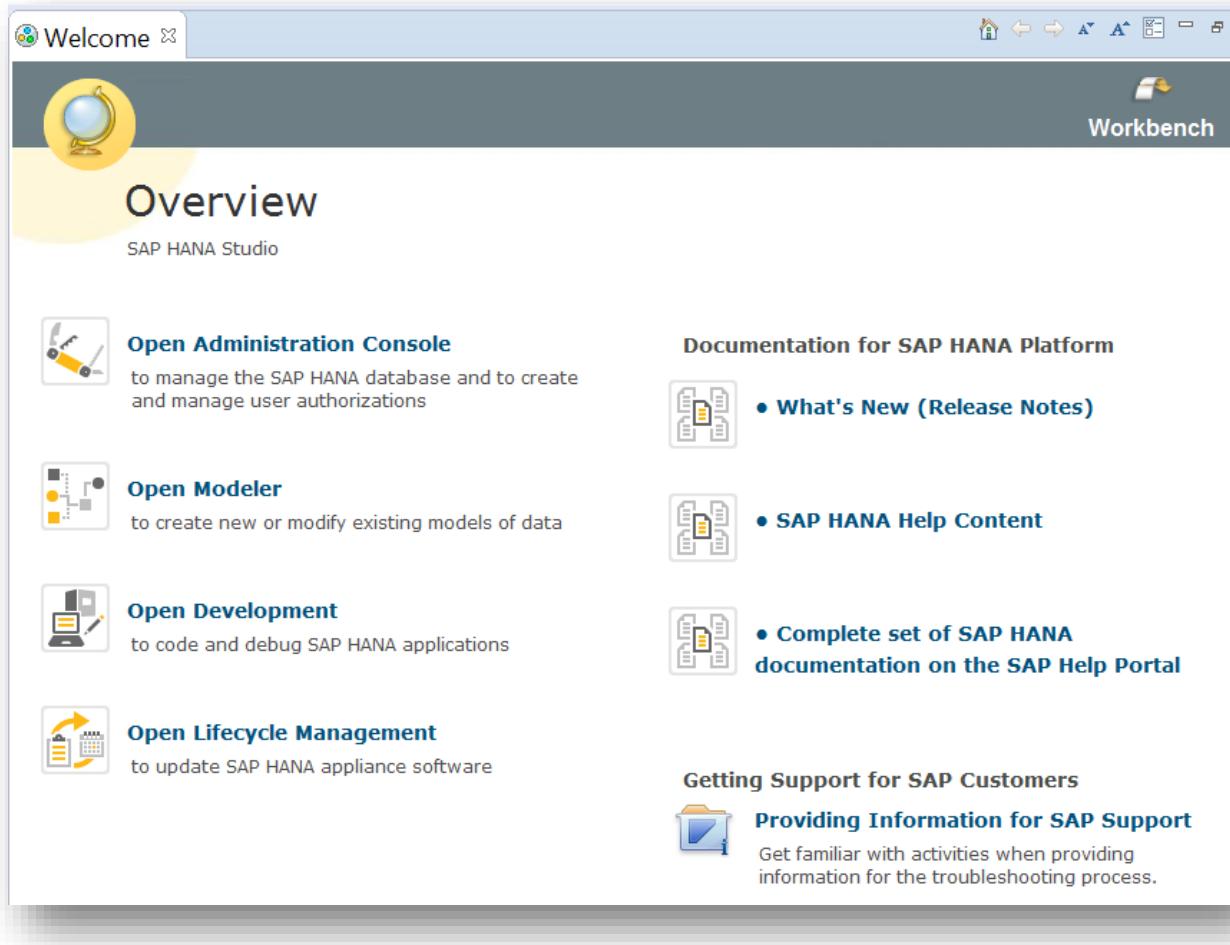
Neither aggregations nor fast searching are required

The table has a small number of rows (eg. configuration tables)

# Agenda

- Introduction to SAP HANA Architecture
- SAP Business Warehouse (BW)
- Data Warehousing Workbench
- SAP HANA Studio
- Info Objects like Characteristics and Key Figures using Eclipse
- Master Data Loading from SAP source system with Operational Data Provisioning .
- Generic Data Sources
- Transformation and Data Transfer Process
- Difference between Classic and Graphical Data Flow Modeling
- Modeling Data Store Object (Advanced)
- Transactional Data Loading from SAP source system with Operational Data Provisioning
- Data Loading from Flat file
- Modeling Composite Providers
- HANA Native Modeling with Calculation Views
- Modeling Open ODS Views
- Process Chains

# SAP HANA Studio - Overview



**SAP HANA studio consists of several applications / perspectives**

- Administration console
- Information modeler
- Lifecycle management

**Perspectives can be switched easily**

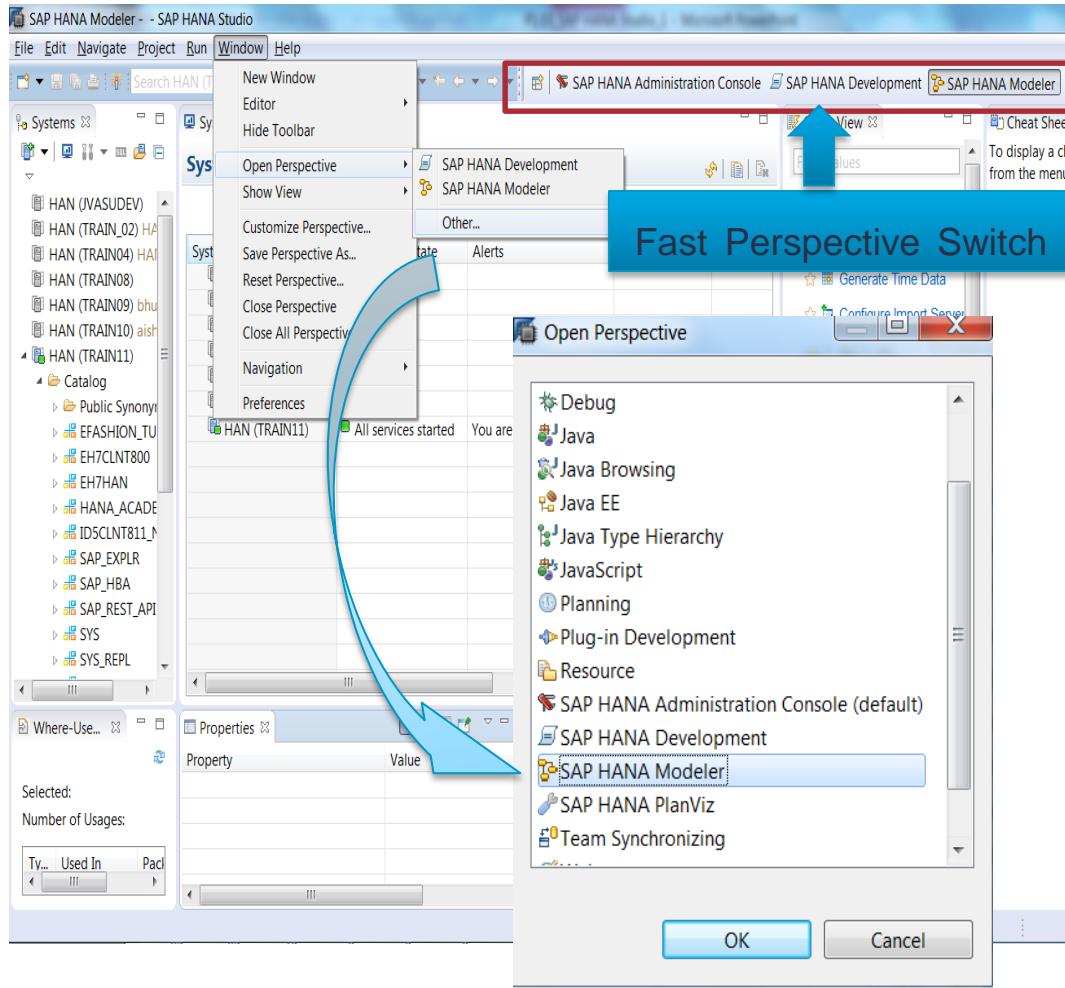
- Welcome screen
- Quick link buttons

# SAP HANA Studio - Overview

## SAP HANA Studio

- Provides an environment for administration , modeling and data provisioning
- SAP HANA studio is a Java based application that runs on the Eclipse platform
- It is a client tool, which can be used to access local or remote HANA system
- Perspectives with in HANA are predefined UI-layouts or views for several application uses
  - The Modeler Perspective is used by Data Architects to create Reporting Models
  - The Administration Console is used by SAP HANA administrators to control and monitor the engine
  - The Documentation view links to the current available documentation

# SAP HANA Studio - Opening a Perspective



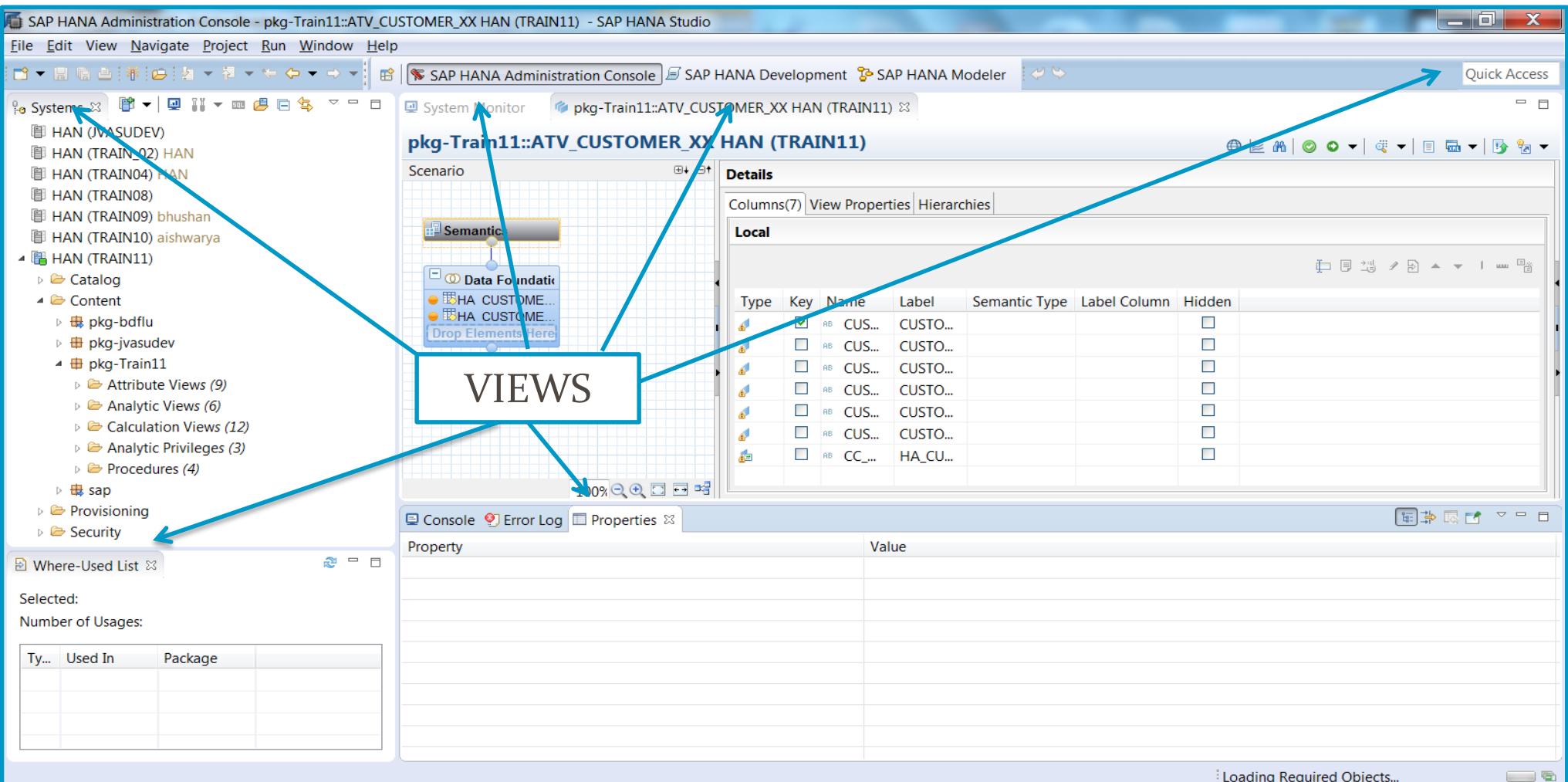
- To create data models in HANA switch to the modeler perspective in the Studio.
- Window -> Open Perspective -> Modeler you can switch to the Quick Launch of the Information Modeler.
- On the right top edge you find the fast perspective switch.
- The SAP HANA Studio offers other perspectives e.g. for debugging, resource and team synchronizing.

# SAP HANA Studio - BWMT Perspective

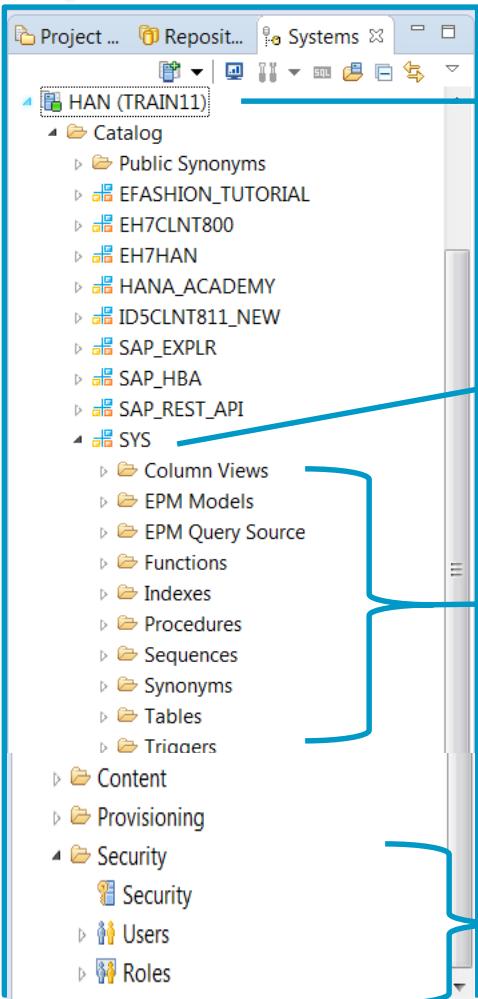
The screenshot illustrates the SAP HANA Studio interface with the BWMT Perspective selected. A central box labeled "VIEWS" is connected by arrows to several key components:

- Project Explorer:** Shows various projects and nodes, with a blue arrow pointing from the "Human Resources" node to the "VIEWS" box.
- Details View:** Displays the "TBWDEC" object, listing fields like KEY and DATA. A blue arrow points from the "TBWDEC" tab to the "VIEWS" box.
- Properties Panel:** Shows the configuration for a data type named "TBWCPIPSQ". A blue arrow points from the "Data Type" section to the "VIEWS" box.
- Problems View:** Shows a list of issues, with a blue arrow pointing from the "Problems" tab to the "VIEWS" box.

# SAP HANA Studio – SAP HANA Modeller Perspectives



# The Systems View-Modeling Content



The systems view lists all the systems that have been registered.

Hana instance - <user>

User database schema

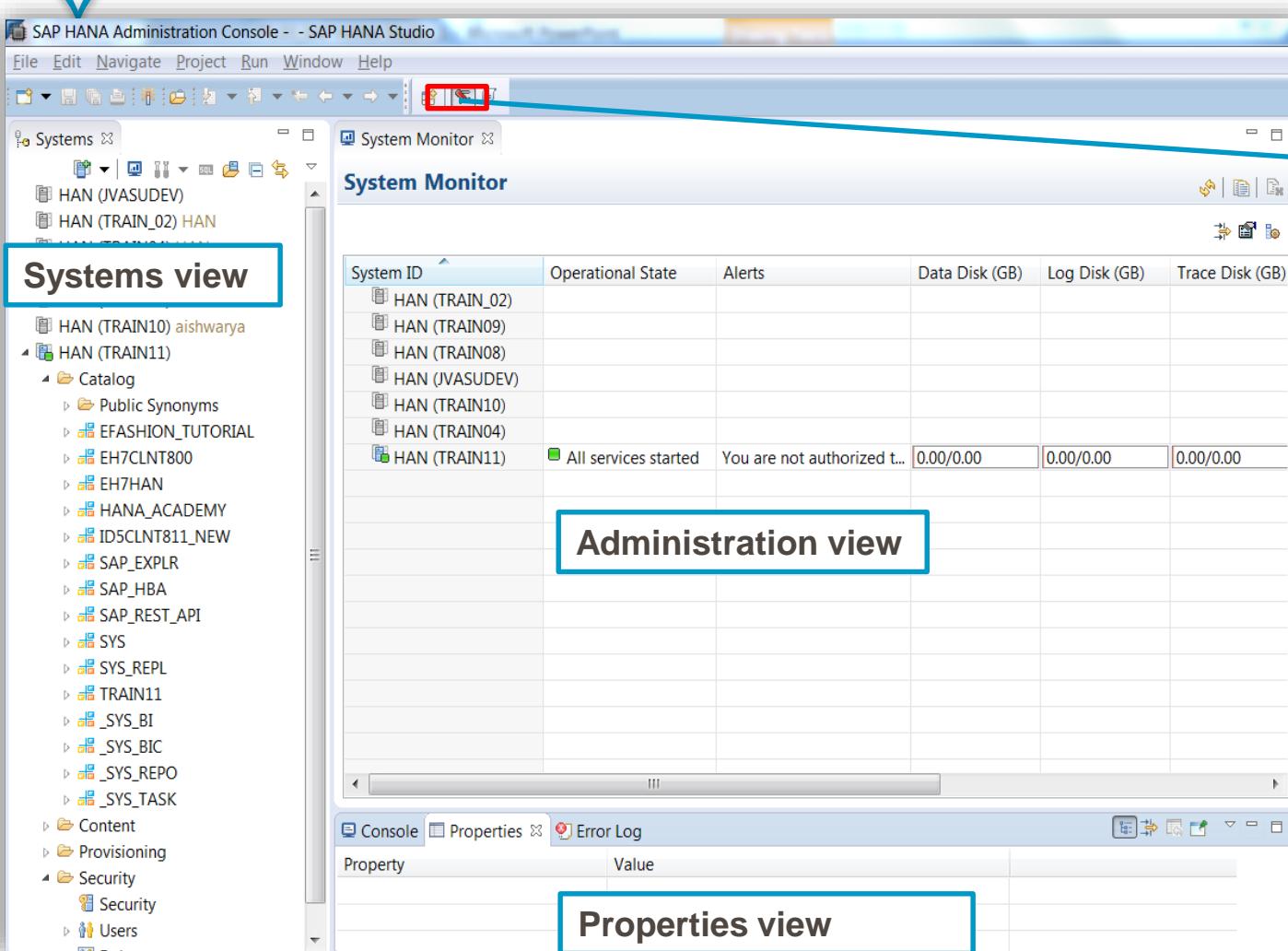
Schema Content : Column views, functions, indexes, procedures, sequences, synonyms, tables, triggers, views.

Users and roles

- The physical tables are located in the Navigation Tree under the Default Catalog node.
- The created column views are always located in schema \_SYS\_BIC, their meta data in schema \_SYS\_BI.

- The physical tables are the only storage area for data within SAP HANA. All the information models that will be created in the modeler will result in database views.

# SAP HANA Administration Perspective



You can access the administration console

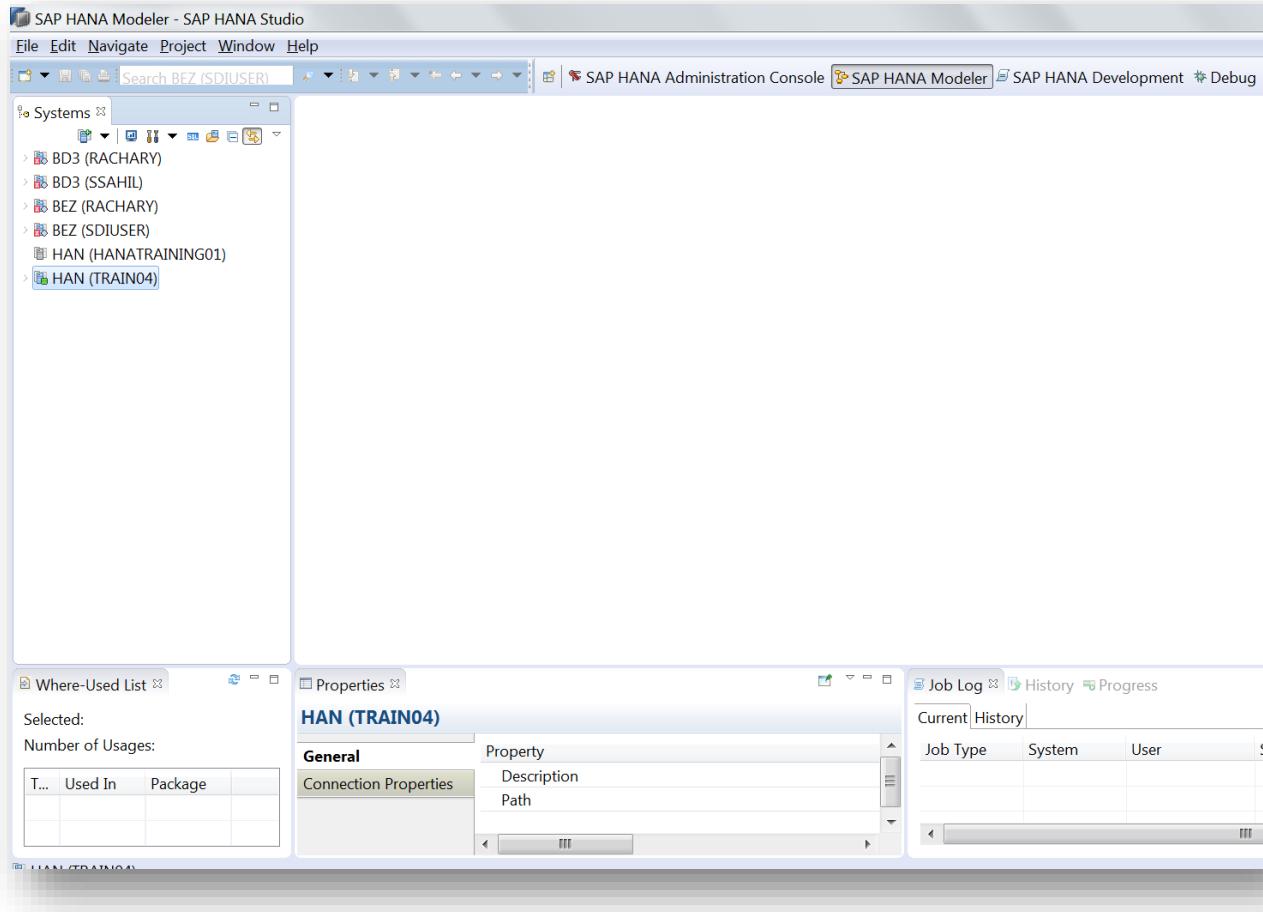
- By selecting the Administration icon in the top marked in red box.
- double click the **system** in the system monitor view

- This Perspective is Pre-delivered by SAP.
- In the administration console you are administrating HANA instances.

## Can do below tasks

- Starting and stopping the instance
- Backup/recovery of the instance
- Monitor the system
- Configure the engine instance
- Doing the problem analysis

# SAP HANA Modeler Perspective

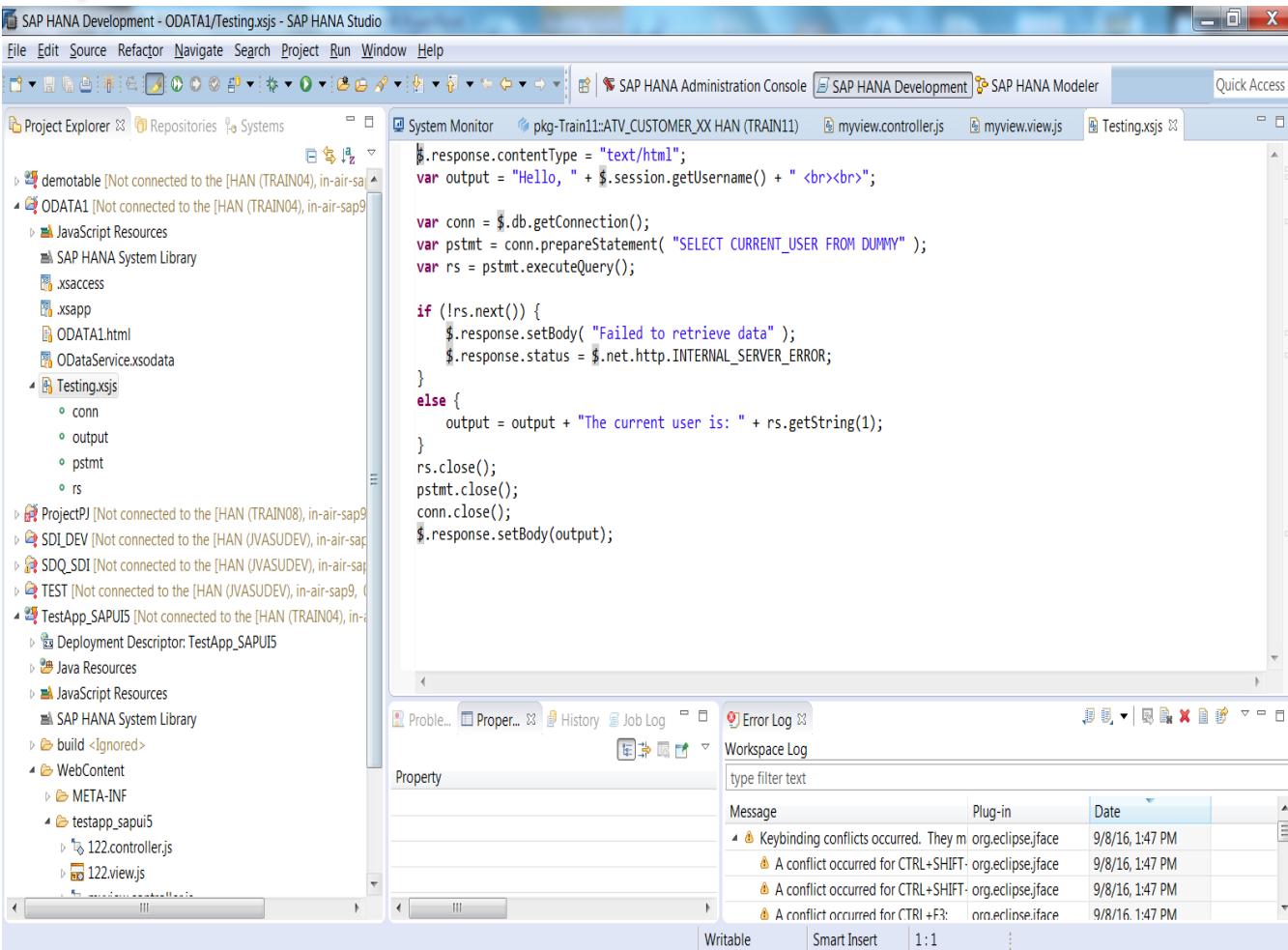


Two Methods to add system.

- **Manually**:-In the systems view ->Right click any blank area and choose Add System
- **By Importing a Landscape**:-Allows to connect to several HANA Systems at the same time by importing an XML file generated previously by a landscape export from the SAP HANA studio installed locally.

- Quick View is a practical starting point to the Modeler.
- From here you can create new views (Information Models) and analytic privileges ,Import and export source schemas, manage and transport information models, schema mapping.

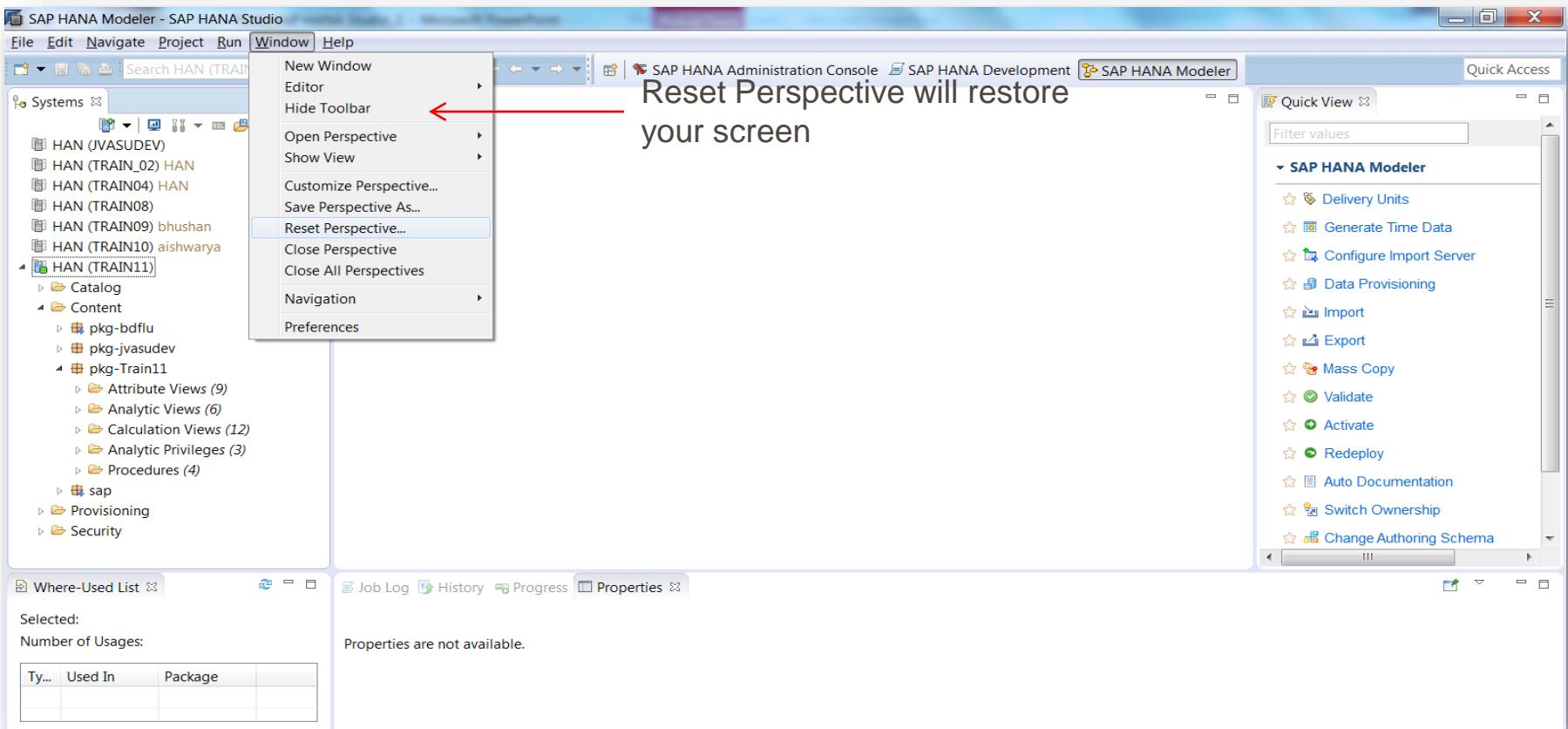
# SAP HANA Development Perspective



- The SAP HANA Studio consist of development perspective and used by application developers.
- This is the platform where we create JavaScript and HTML that will be used in the applications.
- There are plenty of tools to support the developer including trace, debug, code prompts, check-in, and check-out.

# Reset the perspective

Reset your perspectives will restore the screen to the default layout.



# SAP HANA Web-based Development Workbench

SAP HANA Web-based tools are **set of browser-based tools** which **allow users to develop and manage on HANA system from any device**.

SAP HANA Web-based Development Workbench contains four modules:

- **Editor:** Inspect, create, change, delete and activate SAP HANA repository objects.
- **Catalog:** Create, edit, execute and manage SQL catalog artifacts in the SAP HANA database.
- **Security:** Manage users and roles.
- **Trace:** View and download SAP HANA trace files and set trace levels (for example, info, error, debug).

## How to open SAP HANA Web-based Development Workbench:

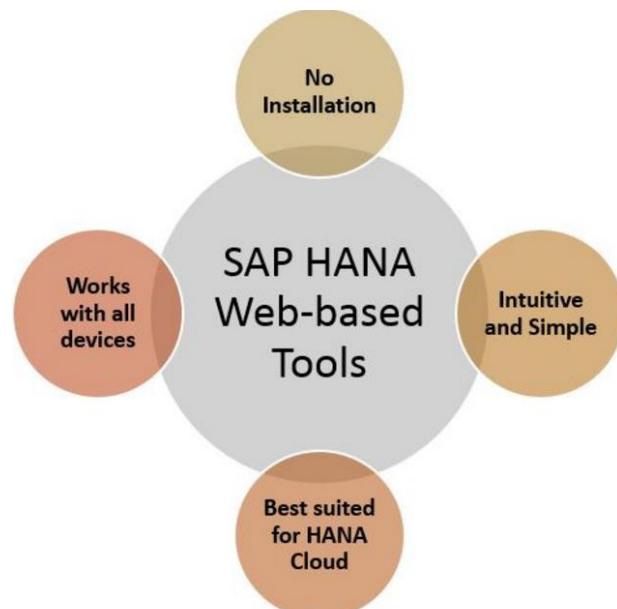
The SAP HANA Web-based Development Workbench is available on the SAP HANA XS Web server at the following URL:

<http://<WebServerHost>:80<SAPHANAINstance>/sap/hana/xs/ide>

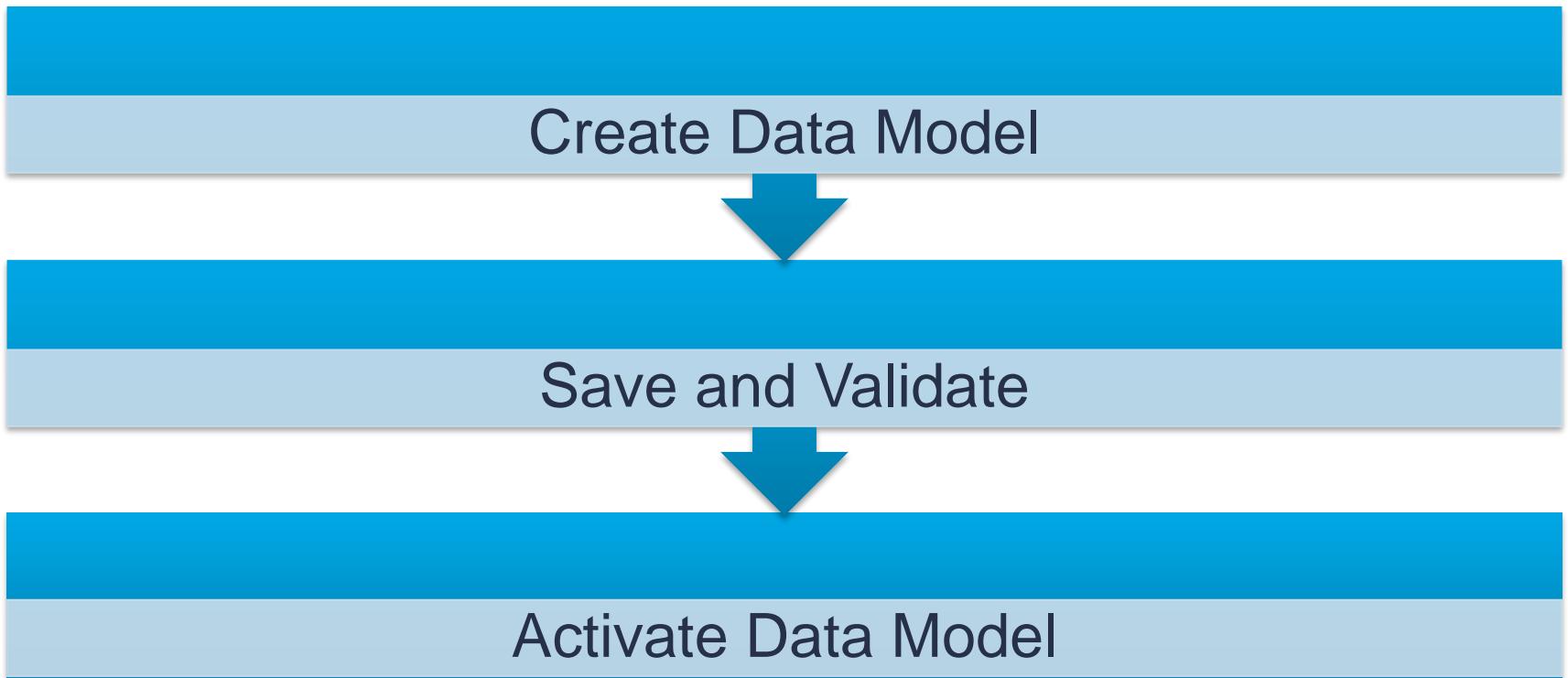
Suppose you HANA host id is “abc” and instance no is 02, then the URL will be

<http://abc:8002/sap/hana/xs/ide>

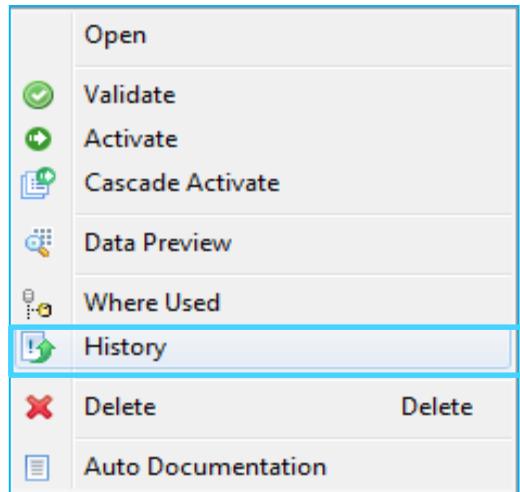
**Note:** The SAP HANA Web-based Development Workbench supports Microsoft Internet Explorer (10+), Mozilla Firefox, and Google Chrome Web browsers.



# Model Validation Processing Information Objects



# Object Versions – View Version History Processing Information Objects



The screenshot shows the SAP Fiori application interface for viewing object history. The title bar says "Where-Used List History". The details pane shows the object's name (AT\_TZH300\_PRODUCT), package (i065554), type (Attribute View), system (HDB), and two activation entries in the table:

Version	Activated By	Activation Date	Last Activated
2	SYSTEM	2012-01-11 11:10:18	14 days ago
1	SYSTEM	2012-01-11 11:06:24	14 days ago



Only Active Versions are displayed. Name of the user activating the view, Activation Date and Period from the last activation are available in Version History.

# Agenda

- Introduction to SAP HANA Architecture
- SAP Business Warehouse (BW)
- SAP HANA Studio
- Data Warehousing Workbench
- Info Objects like Characteristics and Key Figures using Eclipse
- Master Data Loading from SAP source system with Operational Data Provisioning .
- Generic Data Sources
- Transformation and Data Transfer Process
- Difference between Classic and Graphical Data Flow Modeling
- Modeling Data Store Object (Advanced)
- Transactional Data Loading from SAP source system with Operational Data Provisioning
- Data Loading from Flat file
- Modeling Composite Providers
- HANA Native Modeling with Calculation Views
- Modeling Open ODS Views
- Process Chains

# Info Objects like Characteristics and Key Figures using Eclipse

Info Objects are the smallest information units in BW, these are needed to create info providers.

## Types of Info-Object

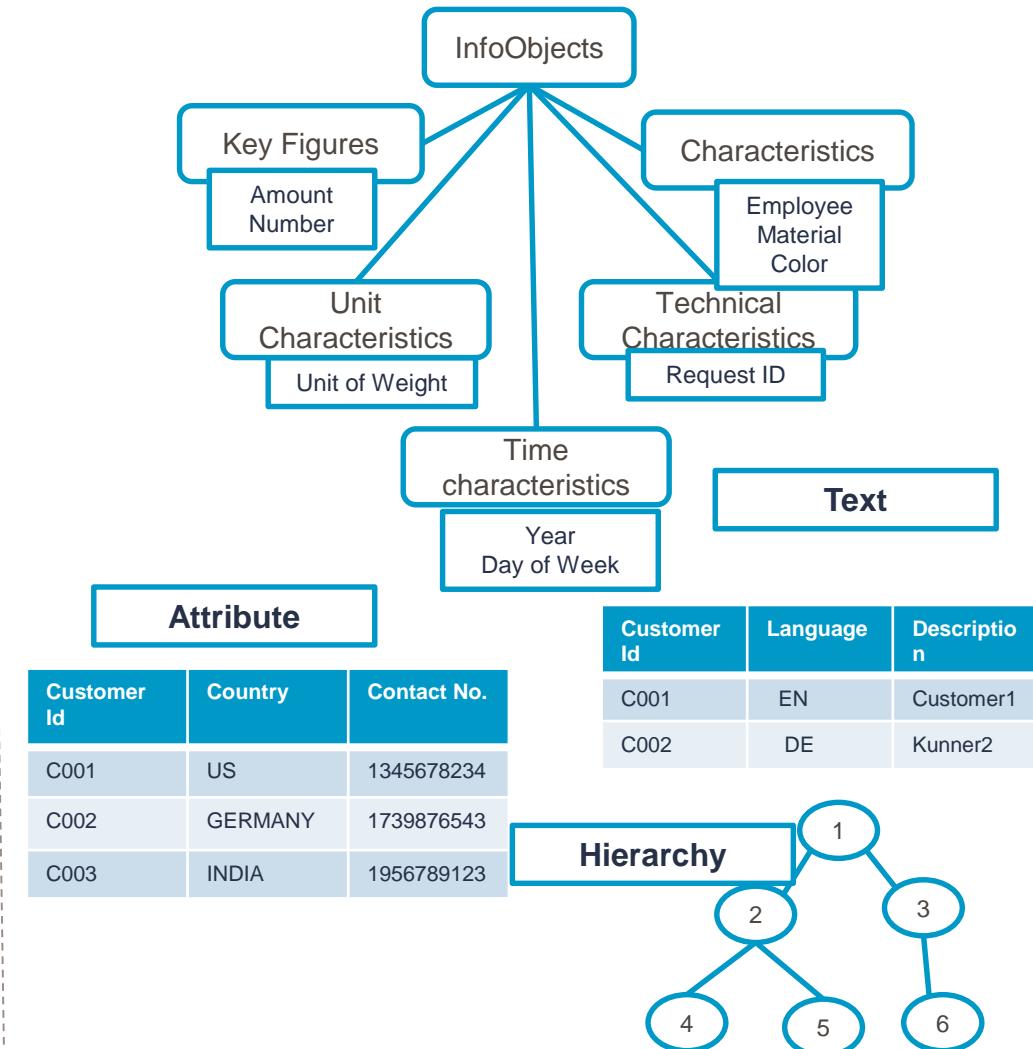
- 1) Characteristics are objects used to analyze key figures.
- 2) Key figures are numeric values with or without units.
- 3) Units are associated with key figure values. For example 100 EUR where 100 is key figure and EUR is unit.
- 4) Time characteristics give time reference to data.
- 5) Technical characteristics are provided by SAP and used for internal administrative purposes.

## Types of Master Data

- 1) Master data is data that does not change frequently. It is used as base for any transaction.

Example: Customer, Company Code, Material, etc.

- 1) Attribute can be created as a characteristic or key figure and there are two types of attributes: display and navigational attributes. e.g Address, Contact No
- 2) Text is the name associated with the key
- 3) Hierarchy is the parent-child relationship



# Info Objects like Characteristics and Key Figures using Eclipse (Contd..)

Beginning with SAP BW 7.5, modelling in the Data Warehousing Workbench (SAPGUI) has been replaced by Eclipse-based Modelling tools. Eclipse Modelling tools provide a unified modelling environment for configuration, management, and maintenance of BW and HANA metadata objects.

The screenshot shows the Eclipse BW Modeling interface with the project "BHT1\_YTEST" selected. The left sidebar shows various nodes under "Favorites" and "BW Repository". The main area displays the "General" tab for a characteristic named "YTEST". The "General" tab includes fields for Technical Name (YTEST), Description (Test), and Properties (Data Type: CHAR - Character String, Length: 10, etc.). Below this is a table for "Compounds" with columns for Superior Characteristic, Description, Referenced Characteristic, Constant Value, Data Type, and Length. A large blue box highlights the "General" tab title.

The screenshot shows the Eclipse BW Modeling interface with the project "BHT1\_YTEST" selected. The left sidebar shows various nodes under "Favorites" and "BW Repository". The main area displays the "Master Data/Texts" tab for a characteristic named "YTEST". The "Master Data/Texts" tab includes sections for "Read Access" (Access Type: Generic Access), "Data Target" (Application Component: BHT1\_YTEST, etc.), and "Miscellaneous" (Delete Master Data Using: RECORDMODE). A large blue box highlights the "Master Data/Texts" tab title.

General

Master Data/Text

The screenshot shows the Eclipse BW Modeling interface with the project "BHT1\_YCOSTCENT" selected. The left sidebar shows various nodes under "Favorites" and "BW Repository". The main area displays the "Attributes" tab for a characteristic named "YCOSTCENT". The "Attributes" tab includes a table for "Display and Navigation Attributes (8) / Col Attributes (0)" with columns for Name, Description, Time-Dependent, and Part. A large blue box highlights the "Attribute" tab title.

Attribute

The screenshot shows the Eclipse BW Modeling interface with the project "BHT1\_YTEST\_REF" selected. The left sidebar shows various nodes under "Favorites" and "BW Repository". The main area displays the "Hierarchies" tab for a characteristic named "YTEST". The "Hierarchies" tab includes sections for "Hierarchy Type" (Hierarchies), "Time-Dependent Properties" (Order in P4H: 1, etc.), and "External Characteristics in Hierarchies (0)". A large blue box highlights the "Hierarchy" tab title.

Hierarchy

# Info Objects like Characteristics and Key Figures using Eclipse (Contd..)

In SAP BW 7.5 on HANA, two additional tabs for maintaining Extended and Runtime Properties are introduced in the info object design/maintenance screen.

The screenshot shows the SAP BW Modeling interface for maintaining a characteristic. The left sidebar lists various BW objects. The main area is titled "BI Clients: Characteristic YTEST". It contains several sections: "General" (Display: Key, Selection: No selection restriction, Include Initial Value in Sort Sequence), "Query Filter Value" (Values in master data table, Only posted values for navigation, Representation of Filter Values During Query Execution: Selection area without values), and "Geographical" (Geographical Type: Initial Characteristic, Action:). A callout box labeled "BI Client" highlights the "General" section.

If you want the initial value in the query output to be displayed on top, select Include Initial Value in Sort Sequence

The screenshot shows the SAP BW Modeling interface for maintaining extended properties. The left sidebar lists various BW objects. The main area is titled "Extended: Characteristic YTEST". It contains sections for "Unit Conversion" (Miscellaneous, Document Property, Fixed Unit of Measure of Characteristic, Constant Value) and "Extended" (No values maintained). A callout box labeled "Extended" highlights the "Extended" section. Below the main area, a navigation bar includes tabs for General, Master Data/Texts, Attributes, BI Clients, Extended, and Runtime Properties.

CONFIDENTIAL – Please do not share

# Info Objects like Characteristics and Key Figures using Eclipse (Contd..)

- Modeling in the Data Warehousing Workbench (SAPGUI) has been replaced by Eclipse-based Modeling tools beginning from SAP BW 7.5
- Eclipse Modeling tools provide a unified modeling environment for configuration, management, and maintenance of BW and HANA metadata objects
- Following table shows difference between SAP BW7.4 and SAP BW 7.5 Edition for SAP HANA

Info Object Feature	SAP BW 7.4	SAP BW 7.5 (Edition for SAP HANA)
Time characteristics as attributes for 0CALMONTH, 0FISCPER etc.	No	Yes
Extended characteristics such as Unit Conversion, Miscellaneous, and Extended	No	Yes
View Data element, SID table, Master table, text table, and hierarchy table for info object	Yes, available at corresponding tabs at info object	Not available in Eclipse. Can be viewed in SAP GUI.
Support for INT8 Key Figures	No	Yes (overcome the 2 billion limitation for integer values)
Disable display attributes in Reference info object	No	We can hide the attributes of reference info objects in Attributes tab using a property called Visible to Consumers.
Additional runtime Properties	Available in Transaction Code RSRT	Available as Runtime Properties tab in Info object

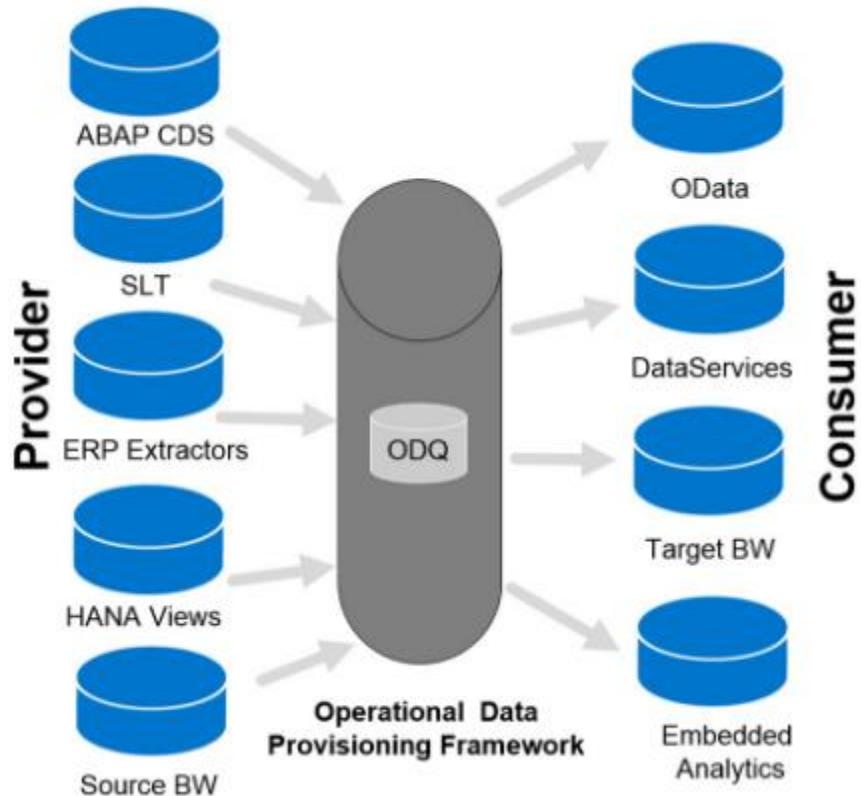
# Agenda

- Introduction to SAP HANA Architecture
- SAP Business Warehouse (BW)
- SAP HANA Studio
- Data Warehousing Workbench
- Info Objects like Characteristics and Key Figures using Eclipse
- Master Data Loading from SAP source system with Operational Data Provisioning .
- Generic Data Sources
- Transformation and Data Transfer Process
- Difference between Classic and Graphical Data Flow Modeling
- Modeling Data Store Object (Advanced)
- Transactional Data Loading from SAP source system with Operational Data Provisioning
- Data Loading from Flat file
- Modeling Composite Providers
- HANA Native Modeling with Calculation Views
- Modeling Open ODS Views
- Process Chains

# What is the ODP Framework

It is a infrastructure to unify data exchange between provider and consumers .

- Enables extract once deploy many architectures for sources
- Unified configuration and monitoring for all provider and subscriber types
- Time stamp based recovery mechanism for all provider types with configurable data retention periods
- Highly efficient compression enables data compression rates up to 90% in Operational Delta Queue (ODQ)
- Quality of service: „Exactly Once in Order“ for all providers
- Intelligent parallelization options for subscribers in high volume scenarios



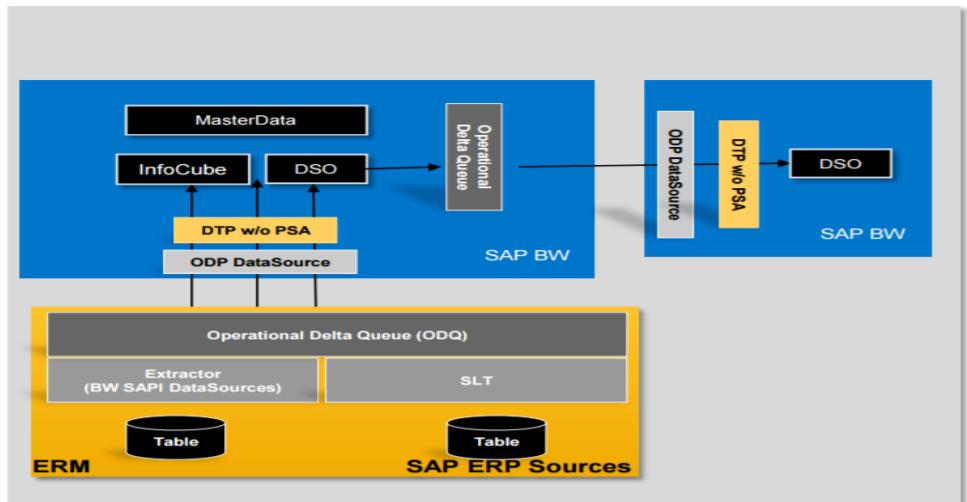
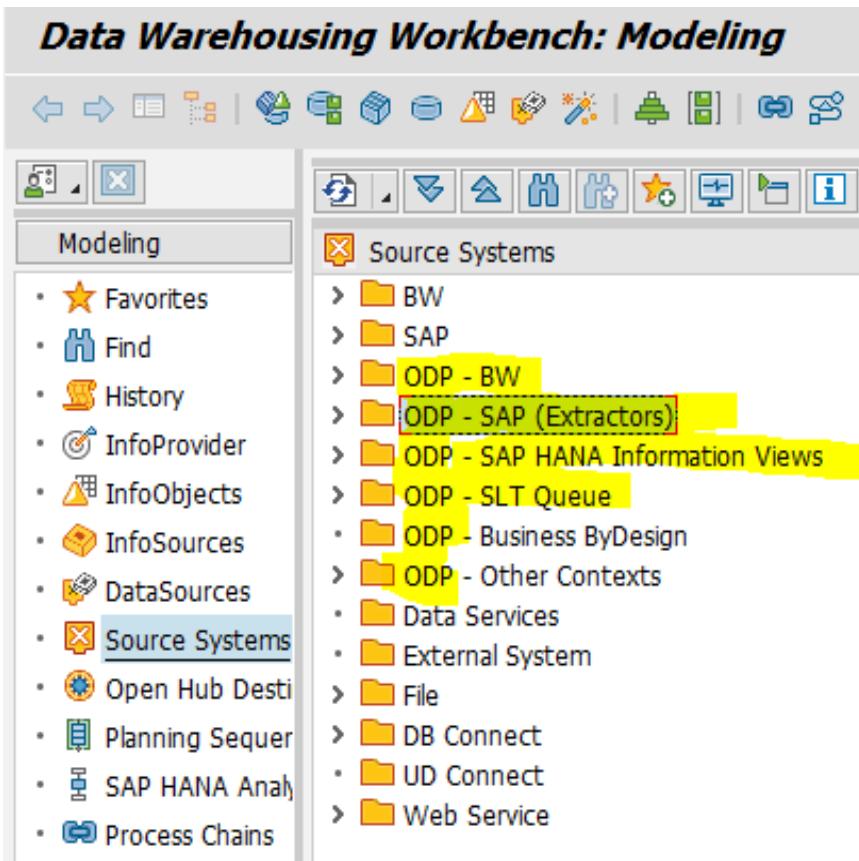
## Major use cases with ODP and BW

Data transfer of extractors SAP ERP (ODQ) a SAP BW

Real-time replication of tables and db-views via SAP SLT (ODQ) a SAP BW

Data transfer between SAP BW (ODQ) a SAP BW

# ODP Based Source Systems



## Main use cases available with ODP in BW 7.5:

- ODP based Data Provisioning Aspects for SAP ERP Sources
- SLT/ODP based real-time replication
- ODP based data transfer between BW systems
- ODP based consumption of SAP HANA Views

ODP is a Net Weaver based Framework, ODQ as persistency resides where the ODP is installed as source . Eg. If SLT is installed within the target BW the ODQ would of course also reside there

# BW Dataflow with DTP Extraction from ODP/ODQ

**Display DataStore Object YFIAR20**

Data Source: **YFIAR20** / **QT6MNDT002**

Extraction Mode: **Data**

Package Size: **200,000**

Adapter: **Extraction from SAP System by Operational Data Provider**

ODP Context: **DataSourcesWithoutESHModeling**

Operational Data Provider: **YFIAR20**

Customer Balances

**Monitor Delta Queues**

Provider: **BW DataSource**

Queue: **OFLIGHT\_BOOKING\_DATA**

Subscriber Type: **SAP BW**

ODP DataSource becomes visible in Operational Delta Queue with the request activation (InfoPackage/DTP) in the subscriber system

**DataSource**: **YFIAR20** / **Customer Balances**

Source System: **QT6MNDT002** / **QT6 Client 002**

Version: **Active**

Active Version: **Executable**

No. of Data Records: **20**

Customer Number	Company Code	Reconciliation Account in General Ledger	Period/year	Fiscal Year
0000000001	0001	0000140000	2010001	2010
0000000001	0001	0000140000	2010002	2010
0000000001	0001	0000140000	2010003	2010
0000000001	0001	0000140000	2010004	2010
0000000001	0001	0000140000	2010005	2010

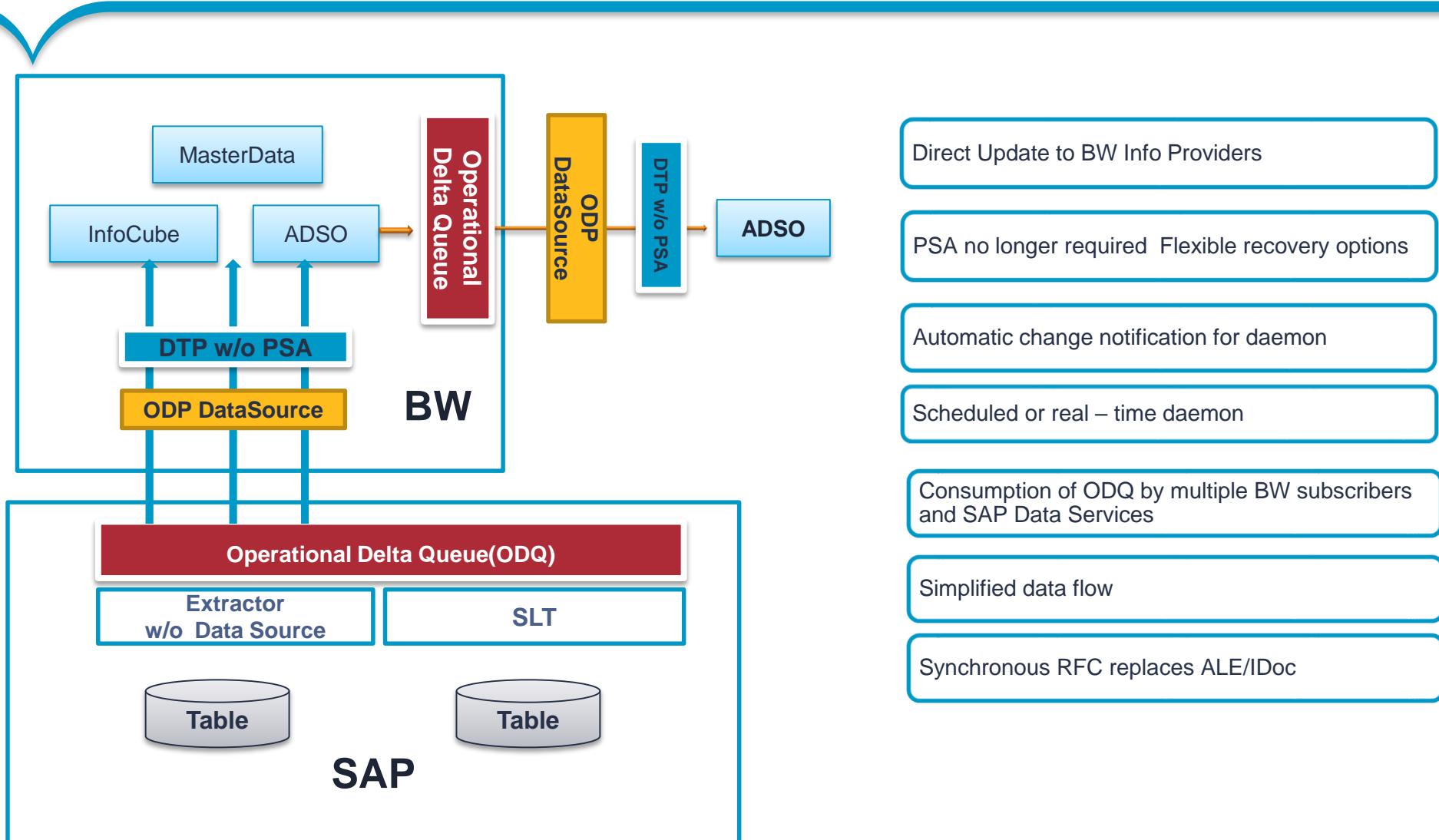
ODP capable ERP extractors can be replicated under source system type ODP – SAP (Extractors)

**Monitor Delta Queues**

Queue	Subscriptions	Requests	Units	Rows	Original Size in Bytes	Compressed Size in	Comp. %
OFLBL_4	1	39	1.081	43,913,163	46,824,320,344	1,626,776,068	96,5
2LBS_11_V_SSL	1	38	17,920	9,150,147	2,671,842,924	92,564,176	96,5
2LBS_11_VAHDR	1	38	18,743	742,030	2,116,269,560	139,457,257	93,4
2LBS_11_VAITM	1	38	18,928	11,209,404	79,341,935,784	1,634,280,475	97,9
2LBS_11_VAKON	1	1	43,375	207,654,781	1,563,677,706,112	15,978,100,390	99,0
2LBS_11_VASQL	1	38	18,289	8,688,192	65,196,192,768	1,219,600,770	98,1
2LBS_11_VASTH	1	38	18,252	918,756	66,150,432	11,138,413	83,2
2LBS_12_VOHDR	1	93	10,059	782,368	2,401,869,760	112,687,147	95,3
2LBS_12_VCTTM	2	92	10,659	7,643,541	48,490,624,104	935,558,525	98,1
2LBS_12_VCSQL	1	0	9,954	7,627,978	47,212,785,088	781,110,535	98,3
2LBS_13_VOHDR	1	194	4,259	457,387	852,569,368	38,761,222	95,5
2LBS_13_VDTTM	1	194	4,291	4,988,388	29,052,371,712	670,592,657	97,7
2LBS_13_VOKON	1	192	11,601	104,698,610	556,996,605,200	8,401,877,038	98,5

One common monitor for data provisioning information:  
Many beneficial information shown: number of units, records, requests, size, compression factor, number of subscriber per queue.

# Simplified data provisioning from SAP ERP and SAP BW



# ODP enabled Extractors

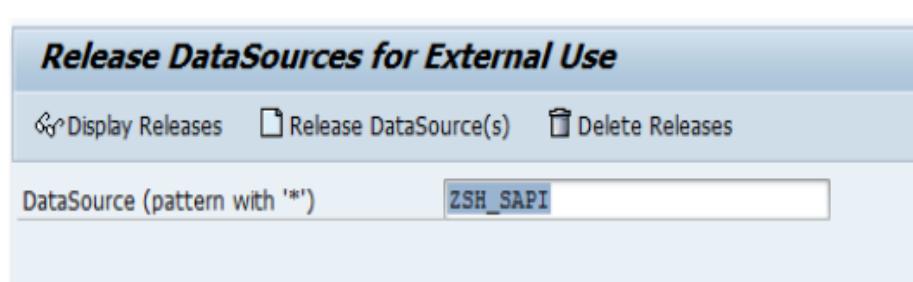
The ODP API does not show all Extractors, it only shows the released ones. The idea is that over the time multiple Extractors have been developed by SAP, some became obsolete, some might not work with this API. So along with the ODP API a new table is created in the dictionary called ROOSATTR containing all the Extractors in the API and hence Data Services 4.0 supports.

Initially these will be a very limited list focusing only on the most important Extractors and will grow over the time. And for sure it will not include customer written Extractors.

## To use the ODP data replication API for any generic Data Source

- Single Extractors can be released with RODPS\_OS\_EXPOSE.
- ODP API2.0 provides the list of ODP enabled extractors in ROOSATTR

OLTPSOURCE	EXPOSE_EXTERNAL
OACTTYPE_ATTR	X
OAUD_ACTION	X
OAUD_ACTION_ATTR	X
OAUD_ACTION_TEXT	X
OBPM_DEADLINES	X
OCOSTELMNT_ATTR	X
OCOSTELMNT_TEXT	X
OCO_OM_CCA_1	X
OCO_OM_CCA_40	X
OCO_OM_CCA_9	X
OCUSTOMER_TEXT	
ODF_MATL_ATTR	X
ODF_MATL_TEXT	X
OEPM_PO	X
OFI_AR_20	X
OFI_GL_1	X



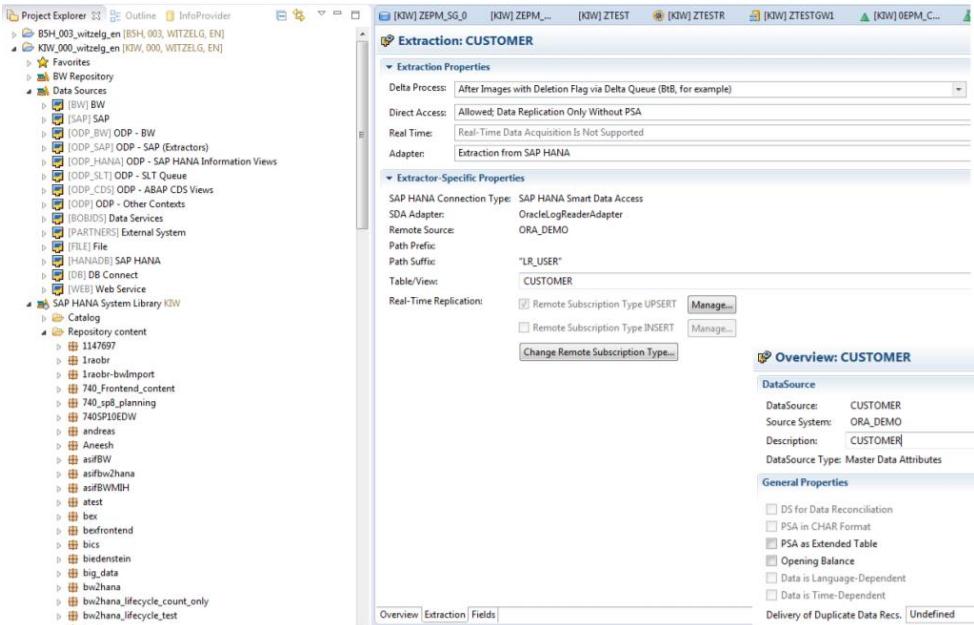
# Demo

1. Master Data Loading from SAP source system with Operational Data Provisioning .
2. Transactional Data Loading from SAP source system with Operational Data Provisioning

# Agenda

- Introduction to SAP HANA Architecture
- SAP Business Warehouse (BW)
- SAP HANA Studio
- Data Warehousing Workbench
- Info Objects like Characteristics and Key Figures using Eclipse
- Master Data Loading from SAP source system with Operational Data Provisioning .
- Generic Data Sources
- Transformation and Data Transfer Process
- Difference between Classic and Graphical Data Flow Modeling
- Modeling Data Store Object (Advanced)
- Transactional Data Loading from SAP source system with Operational Data Provisioning
- Data Loading from Flat file
- Modeling Composite Providers
- HANA Native Modeling with Calculation Views
- Modeling Open ODS Views
- Process Chains

# DataSource Maintenance in Eclipse



## New Eclipse-based DataSource Editor in BW Modeling Tools

Modern, intuitive Eclipse based UI

- Enhanced replication dialog
- New update dialog
- Remote repository based on information from source system (ODP only)

## Source Systems Available for Modeling in Eclipse

- SAP HANA, ODP, S-API (“extractors”)

# Agenda

- Introduction to SAP HANA Architecture
- SAP Business Warehouse (BW)
- SAP HANA Studio
- Data Warehousing Workbench
- Info Objects like Characteristics and Key Figures using Eclipse
- Master Data Loading from SAP source system with Operational Data Provisioning .
- Generic Data Sources
- Transformation and Data Transfer Process
- Difference between Classic and Graphical Data Flow Modeling
- Modeling Data Store Object (Advanced)
- Transactional Data Loading from SAP source system with Operational Data Provisioning
- Data Loading from Flat file
- Modeling Composite Providers
- HANA Native Modeling with Calculation Views
- Modeling Open ODS Views
- Process Chains

# Transformation and Data Transfer Process

The Transformation process is used to perform data consolidation, cleansing and data integration.

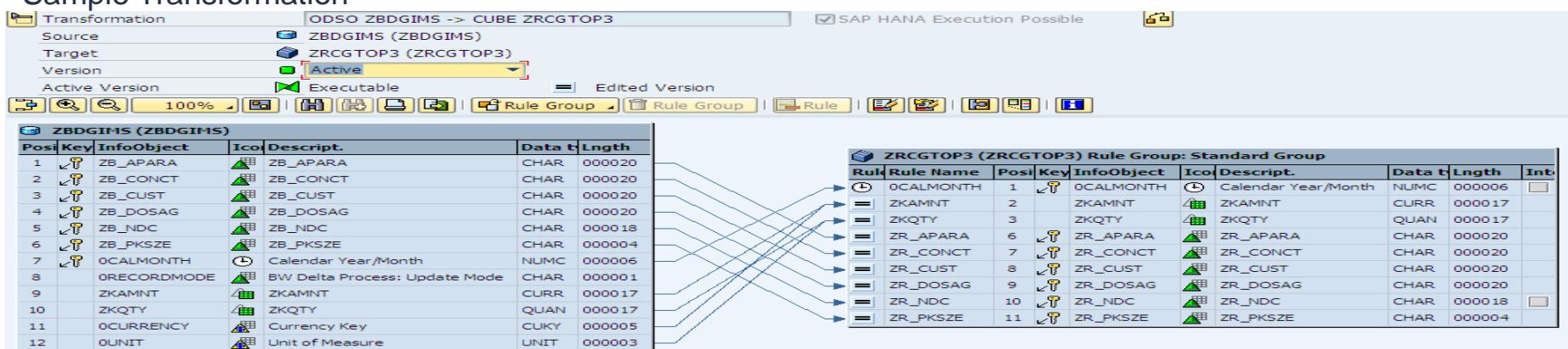
When data is loaded from one BI object to other BI object, transformation is applied on the data. Transformation is used to convert a field of source into the target object format.

Each Transformation consists of minimum one transformation rule.

As different rule types and routines are available that allows you to create simple to complex transformations.

Basically, Transformations are built up from **rules**, which bind one or more fields of the source structure to a given field of the target structure. Furthermore you can build up a complex logic for a Transformation using **additional routines**.

## Sample Transformation



# Transformation and Data Transfer Process

## Rules

Rules define the source fields which are used for loading a given target field, and the operation how the target value is calculated. Using rules, it is not possible to use several records from the source, or create several records in the target; Transformation rules are applied record-wise and create exactly one target record from exactly one source record.

Taking the given source fields as input, using rules you can choose from the following operations (called **Rules Types**)

**Constant:** assigns a predefined value regardless of the content of the input fields

**Direct Assignment:** copies the value of the input field without any conversion (only the possible conversion routines defined in the target InfoObject might apply)

**Formula:** using this rule type you can define basic logical operations to be performed on the input fields without having to know the ABAP programming language

**Initial:** assigns an initial value to the target field

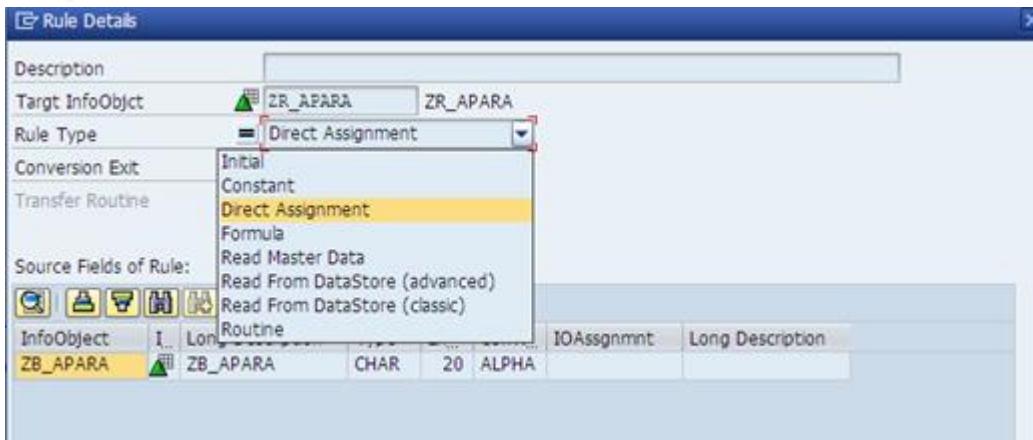
**Read Master Data:** using the input field value as a key, you can read the value of a Characteristic attribute belonging to a specified Characteristic; the InfoObject of the required Characteristic attribute must comply the InfoObject of the target field of the rule

**Read from DataStore:** using the input field value(s), you can read the data fields of a specified DataStore Object (DSO)

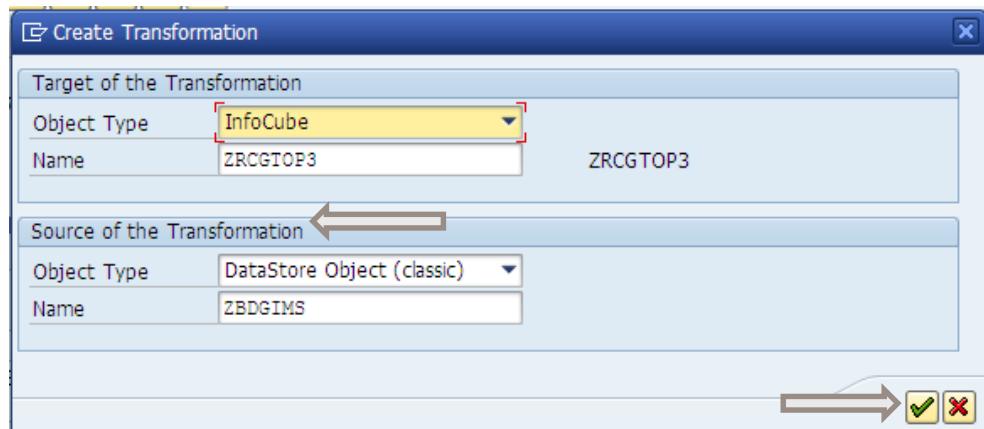
**Routine:** you can implement an ABAP method to calculate the value of the target field based on the values of the source fields;

# Transformation and Data Transfer Process

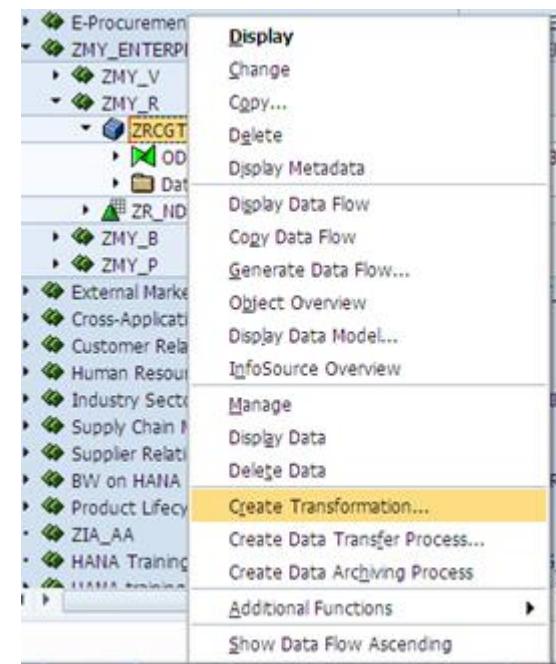
## Rule Type



Enter Source of the Transformation and click enter



Right Click on Info provider and select create transformation to create it



# Transformation and Data Transfer Process

## Additional Routines

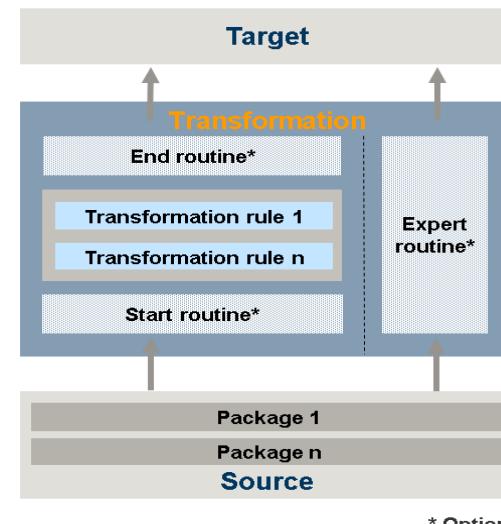
Using a start, end or expert routine you can define specific subroutines to be executed at certain points during the Transformation process.

These include the following:

**Start routine** is executed before applying the defined rules for the source dataset

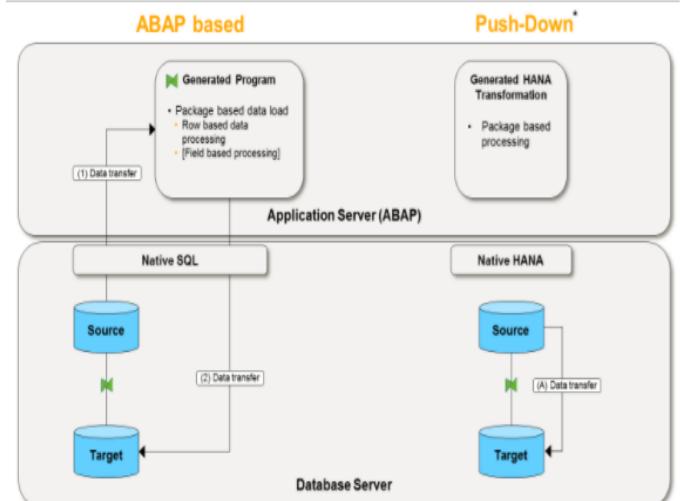
**End routine** is executed after applying the defined rules for the source dataset

**Expert routine** omits the defined rules and enables you to implement the entire Transformation process in code.



\* Optional

With support package stack (SPS) 05 for SAP NetWeaver 7.4, SAP has brought ABAP and SAP HANA together with features that enable developers to leverage the best of both worlds. One of the key differences for developing applications in ABAP for HANA is that you can push down data intense computations and calculations to the HANA DB layer instead bringing all the data to the ABAP layer and processing the data to do computations.



# Transformation and Data Transfer Process

DTP determines the process for transfer of data between two persistent/non persistent objects within BI.

As of SAP NetWeaver 7.0, InfoPackage loads data from a Source System only up to PSA. It is DTP that determines the further loading of data thereafter.

## Use

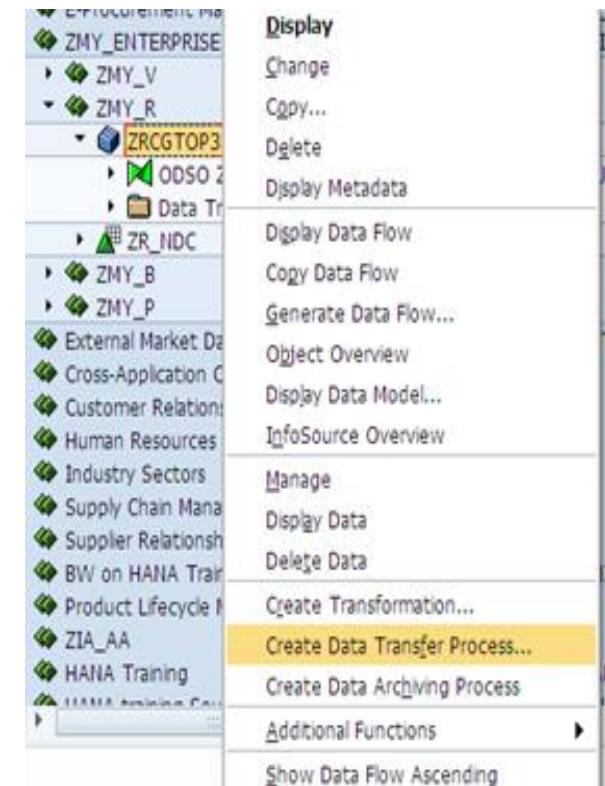
- Loading data from PSA to InfoProvider(s).
- Transfer of data from one InfoProvider to another within BI.
- Data distribution to a target outside the BI system; e.g. Open HUBs, etc.

In the process of transferring data within BI, the Transformations define mapping and logic of data updating to the data targets whereas, the Extraction mode and Update mode are determined using a DTP

## Key Benefits of using a DTP over conventional IP loading

DTP follows 1:1 mechanism between a source and a Target i.e. one DTP sources data to only one data target whereas, IP loads data to all data targets at once.

Better Error handling mechanism with the use of Temporary storage area, Semantic Keys and Error Stack.



# Transformation and Data Transfer Process

**Extraction :** There are two types of Extraction modes for a DTP – *Full* and *Delta*

- **Full** : It selects all the data available in the source based on the Filter conditions mentioned in the DTP
- **Delta** : When DTP is executed with Extraction mode Delta for the first time, all existing request till then are retrieved from the source and the delta is automatically initialized

❖ **Only get delta Once** : If this indicator is set, a snapshot scenario is built. The Data available in the Target is an exact replica of the Source Data

❖ **Get all new data request by request** : If you set this indicator in combination with ‘Retrieve Until No More New Data’, a DTP gets data from one request in the source. When it completes processing, the DTP checks whether the source contains any further new requests. If the source contains more requests, a new DTP request is automatically generated and processed.

NOTE: If ‘Retrieve Until No More New Data’ is unchecked, the above option automatically changes to ‘Get One Request Only’. This would in turn get only one request from the source.

Also, once DTP is activated, the option ‘Retrieve Until No More New Data’ no more appears in the DTP maintenance.

**Update :** Error Handling in DTP - Erroneous records in a DTP load are written to a stack called Error Stack.

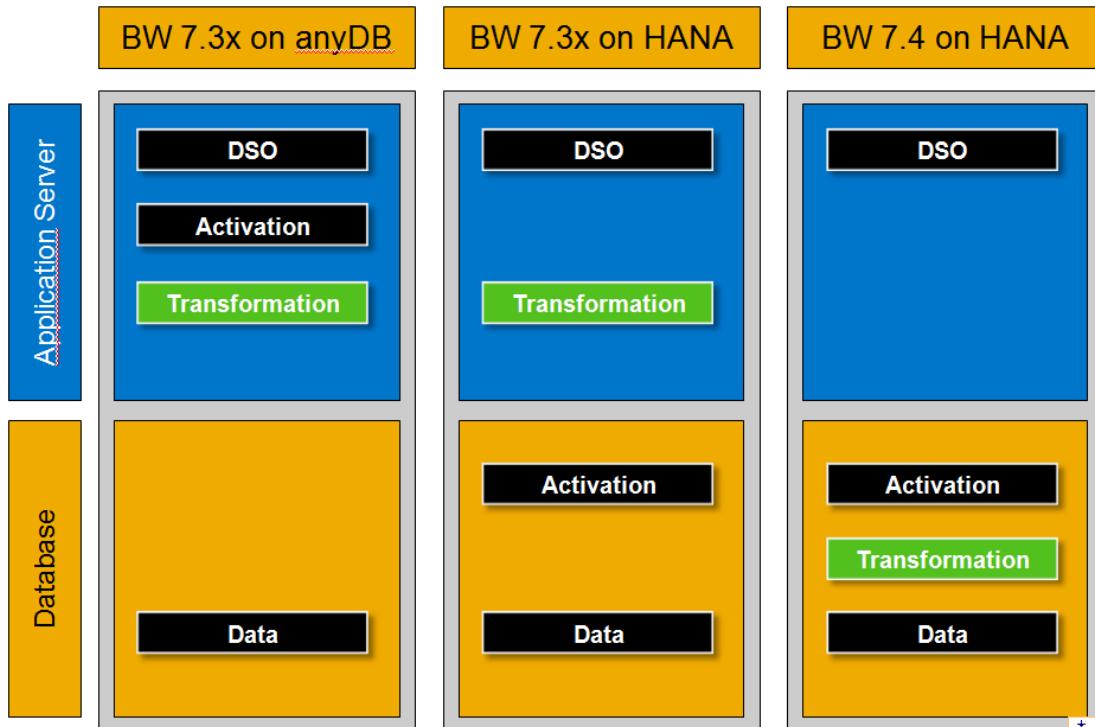
Error Stack is a request-based table (PSA table) into which erroneous data records from a data transfer process (DTP) are written. The error stack is based on the data source (PSA, DSO or Info Cube), that is, records from the source are written to the error stack.

In order to upload data to the Data Target, we need to correct the data records in the Error Stack and manually run the Error DTP.

**Execute :** Processing Mode:

- **Serial Extraction, Immediate Parallel Processing**
- **Serial in dialog process (for debugging)**
- **No data transfer; delta status in source: fetched:**

# SAP HANA optimized Transformations



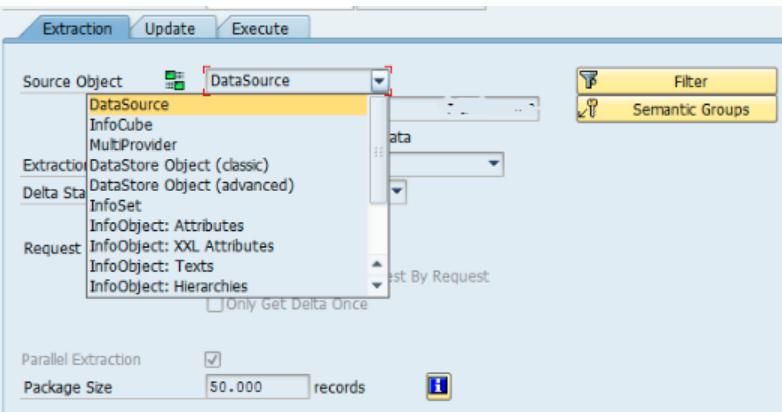
## Next level of performance in data staging for BW on HANA

- Standard Transformations are gradually optimized to be processed directly in SAP HANA by generating according DB procedure
- First set of Transformations optimized i.e.:
  - Mappings
  - Conversions (time, currency, units)
  - Formulas
  - Read Master Data, Read from DSO
  - Expert routine for HANA SQL script . .
- Sources:  
PSA, DSOs, InfoCubes, SPOs,  
CompositeProvider, MultiProvider
- Targets: DSO

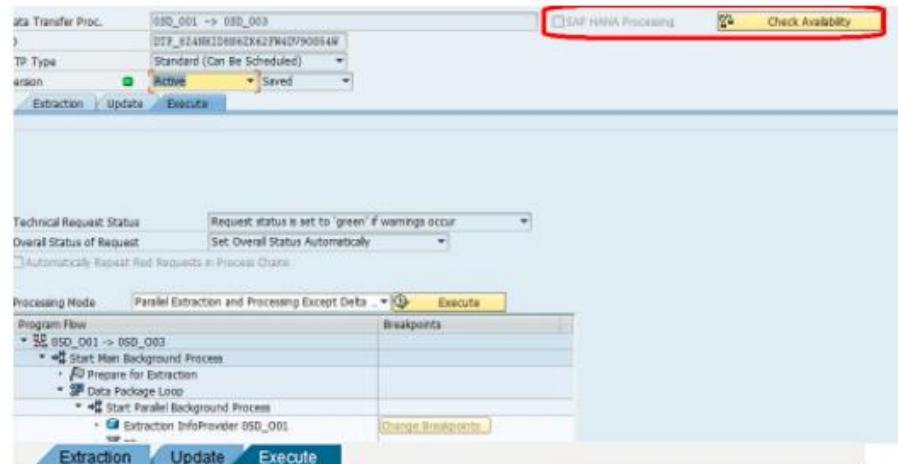
# Transformation and Data Transfer Process

## Some of the new Features of DTP

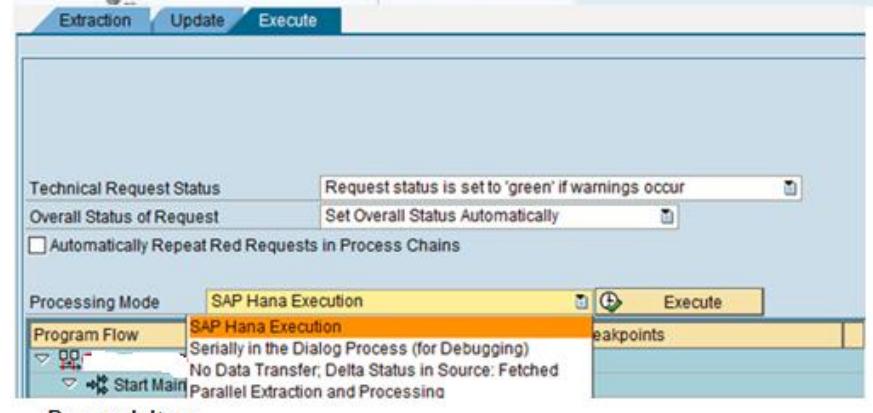
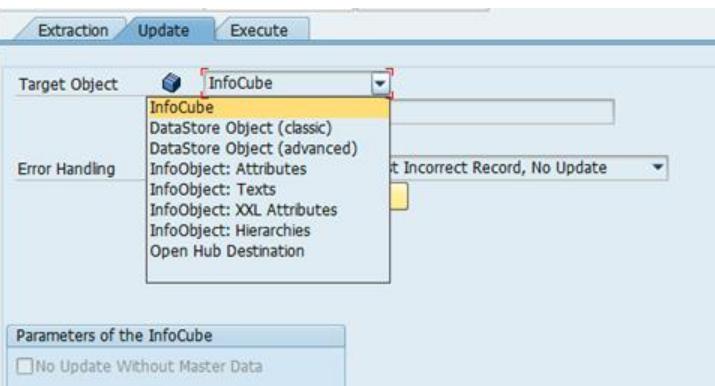
### More Options in Source Object in 'Extraction' Tab



### SAP HANA Processing Mode



### More Options in Target Object in 'Update' Tab



# Agenda

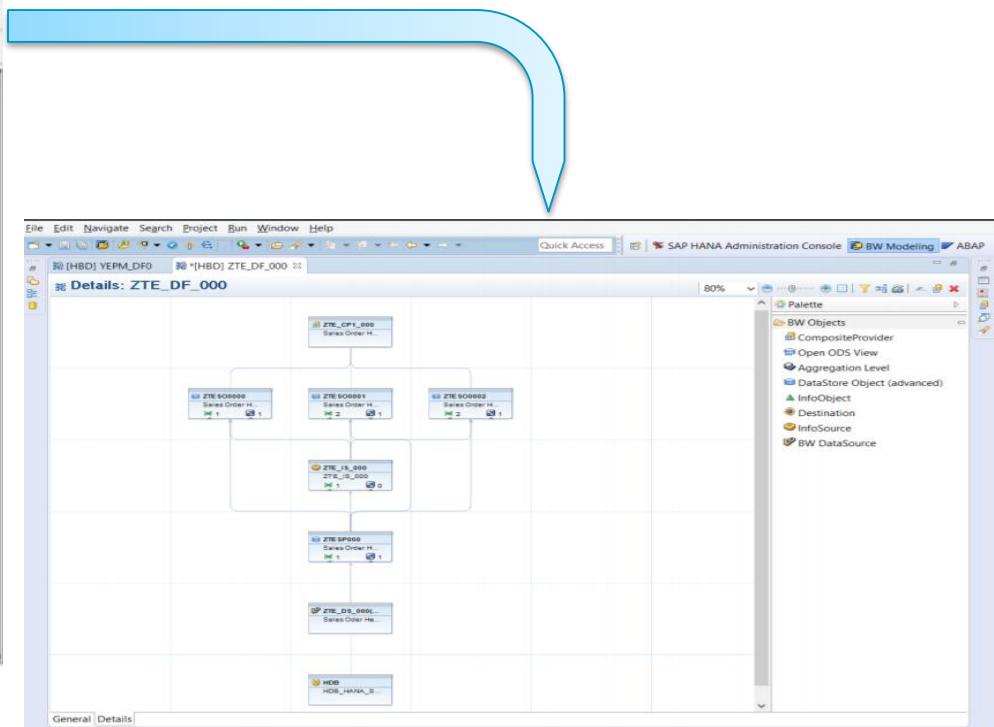
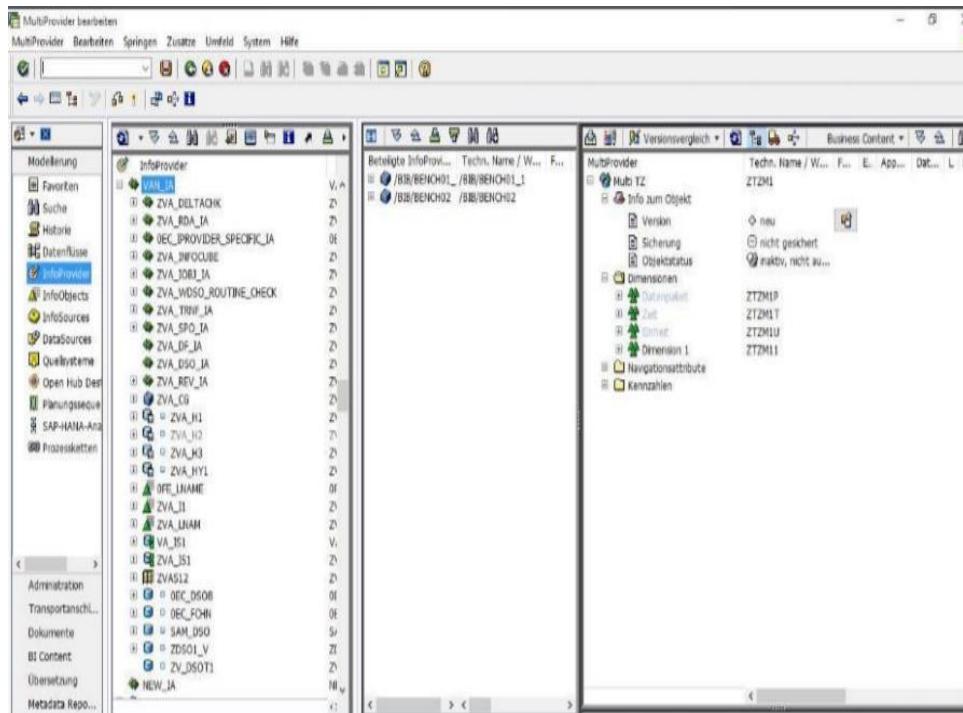
- Introduction to SAP HANA Architecture
- SAP Business Warehouse (BW)
- SAP HANA Studio
- Data Warehousing Workbench
- Info Objects like Characteristics and Key Figures using Eclipse
- Master Data Loading from SAP source system with Operational Data Provisioning .
- Generic Data Sources
- Transformation and Data Transfer Process
- Difference between Classic and Graphical Data Flow Modeling
- Modeling Data Store Object (Advanced)
- Transactional Data Loading from SAP source system with Operational Data Provisioning
- Data Loading from Flat file
- Modeling Composite Providers
- HANA Native Modeling with Calculation Views
- Modeling Open ODS Views Process Chains

# Difference between Classic and Graphical Data Flow Modeling

# Classic Vs Flow graph Modelling

The BW Modeling tools contain a graphical user interface. This provides you with a simple way of creating, editing and documenting data flows and objects in data flows. You can also display and modify the data flow, which corresponds to a specific object, on the graphical user interface.

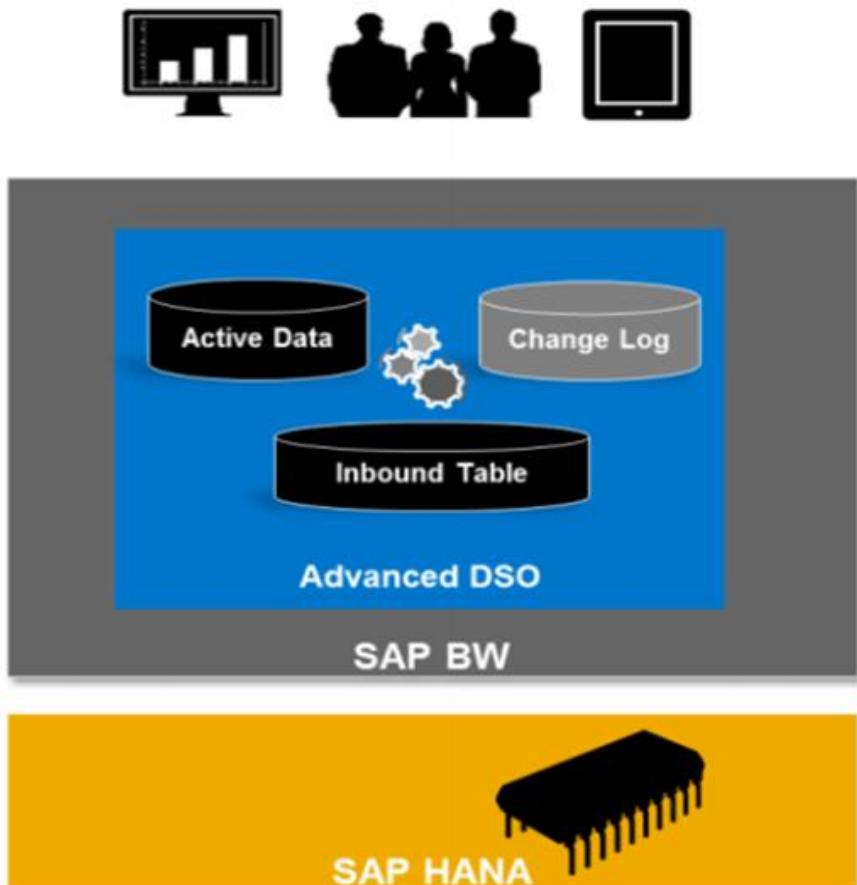
The data flow is the central entry point for modeling in SAP BW/4HANA. You can use the data flow to model objects and their relationships with each other for a particular scenario.



# Agenda

- Introduction to SAP HANA Architecture
- SAP Business Warehouse (BW)
- SAP HANA Studio
- Data Warehousing Workbench
- Info Objects like Characteristics and Key Figures using Eclipse
- Master Data Loading from SAP source system with Operational Data Provisioning .
- Generic Data Sources
- Transformation and Data Transfer Process
- Difference between Classic and Graphical Data Flow Modeling
- Modeling Data Store Object (Advanced)
- Transactional Data Loading from SAP source system with Operational Data Provisioning
- Data Loading from Flat file
- Modeling Composite Providers
- HANA Native Modeling with Calculation Views
- Modeling Open ODS Views Process Chains

# SAP BW advanced data store object (BW-ADSO)



**Advanced DSO have persistent object is combination of below objects**

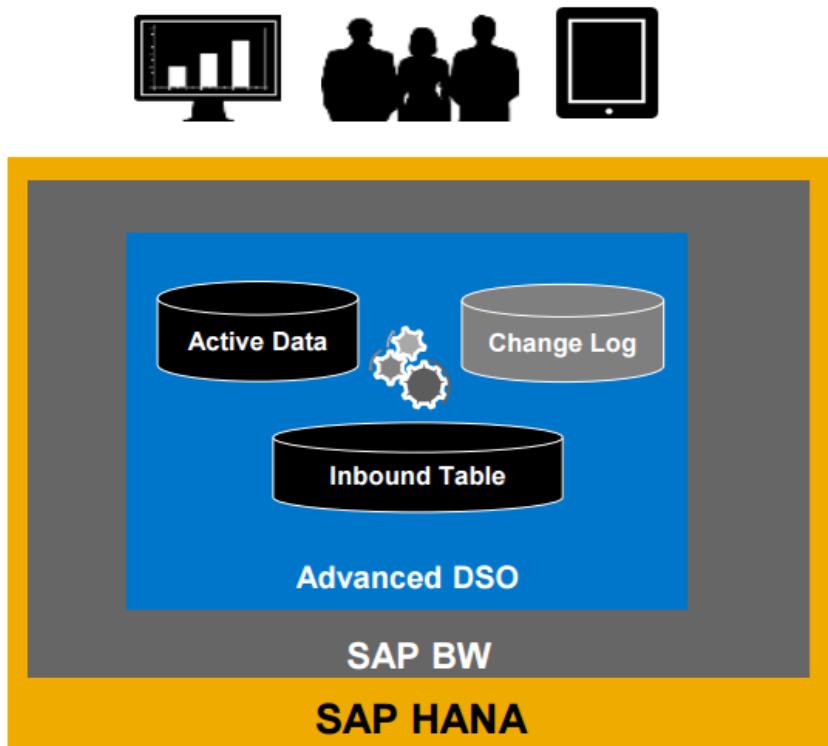
1. Field based structure of PSA (file based modeling)
2. No activation required for like WDSO (update property)
3. Three tables have (active, new, change log table) like standard DSO (over write property)
4. Info cube model – select setting – all char are keys (addition property )

## BackEnd Tables

ADSO can consist of up to three tables (similar to classic Standard DSO)

- a. Inbound table – Data is first loaded to Inbound table
- b. Change Log Table - provides the change history required for updating delta records in other InfoProviders.
- c. Active Data Table – On activation of request in DSO, records from Inbound tables are moved to Active Data table

# SAP BW advanced data store object (BW-ADSO)(Contd..)



**Next generation of Data Store Object simplifies data modeling in SAP BW**

- Consolidate DataStore Objects and Info Cubes  
Single type of BW object  
Settings to manage properties for certain services
- Combine Info Object and field based modeling
- New intuitive & modern modeling UI

## Optimized services

- High frequent data loads - based on optimized request management
- Change of usage scenario without deletion of data • Up to 120 key fields
- Optional SID entries for query performance optimization

# Semantic Group for Advanced DataStore Objects

A semantic group is used to generate semantically partitioned data models.

The screenshot shows the SAP GUI interface for creating a semantic group. The title bar says "Criteria: SEM\_GRP". The main area has a "Reference Structure" tab selected. A table lists "DataStore Object (advanced)" members: [SEM\_GRP0], [SEM\_GRP1], [SEM\_GRP2], [SEM\_GRP3], [SEM\_GRP4], [SEM\_GRP5], [SEM\_GRP6], and [ADSO\_EXT] ADSO added. The [ADSO\_EXT] row is selected. To the right, a "Detail" pane shows the "Technical Name" as "ADSO\_EXT" and the "Description" as "ADSO added to group". Below this, a "Direct Input" field contains "2008" with a "To" button and a "Add Criterion" button. A list box below shows "[2008]".

## New Highly Flexible Concept for Semantic Partitioning for Advanced DataStore Objects

### Tool to create semantical partitioned data models

- Based on a reference structure
- Each member can be changed independently

### Flexible definition of split criteria

- Pruning information is derived

### Generated advanced DataStore Objects remain individual objects including

- CompositeProvider
- DTP filters

### Only generated objects will be transported into subsequent systems

# Enhancement of Advanced DataStore Objects

The screenshot shows the configuration interface for a DataStore Object named ZSOHDR. It includes sections for General Properties, Modeling Properties, Activation, Model Template, and Storage Options.

- General Properties:** Technical Name: ZSOHDR, Description: Sales Order Header, External SAP HANA View checked.
- Modeling Properties:** Activation: Activate Data (checked), Write Change Log (checked), Keep Inbound Data, Extract from Inbound Table (unchecked), Unique Data Records (unchecked), Snapshot Support (unchecked). Special Types: Direct Update (unchecked), All Characteristics are Key, Reporting on Union of Inbound Tables (checked), Planning Mode (unchecked), Inventory (unchecked).
- Model Template:** A list of templates including Data acquisition layer (including corporate memory), Corporate memory - compression capabilities, Corporate memory - reporting capabilities, Data warehouse layer - delta calculation, Data warehouse layer - data mart, Planning (Planning on InfoCube-like, Planning on Direct Update), and Classic objects (Standard DataStore Object, Write-optimized DataStore Object, InfoCube). The "Standard DataStore Object" is selected.
- Storage Options:** SAP HANA Dynamic Tiering checked.

## Dynamic Tiering

Dynamic Tiering Based on Extension Nodes.

## Selective Deletion with Reverse Image

Propagate result of a selective deletion to BW targets

- Based on new change log request
- Roll back option

## Snapshot Enablement

- Differences between active table and newly loaded request written into change log as reverse images
- Propagate deletion of records in source to further BW targets (e.g. DataSource delivers full only)

## New Process Chain Types

- Clean up activities based on advanced DataStore Object type

# SAP BW Advanced DSO(Use Cases)

- Currently, the ADSO is used solely for reporting purposes.
- From BWOH 7.5 onwards, it is available for planning and non-cumulative scenarios as well.
- It can be deployed in various layers of the LSA++ architecture.
- Depending upon the use in LSA++ architecture, backend tables are created.
- To optimally complete the various tasks, you can specify modeling properties upon creation. You can adapt these properties later, if required.
- Model persistency of new end2end scenarios (staging and reporting) with the advanced DSO

# Understanding the Modeling Properties of aDSO

The aDSO physical structure, views and behaviour can be controlled using the modeling properties highlighted below. By selecting different property combinations, the aDSO can mimic the structure and functionality of any classic object.

## Modeling Properties

Activate Data

Write Change Log

Keep Inbound Data, Extract from Inbound Table

Unique Data Records

All Characteristics are Key, Reporting on Union of Inbound and Active Table

Extended Storage

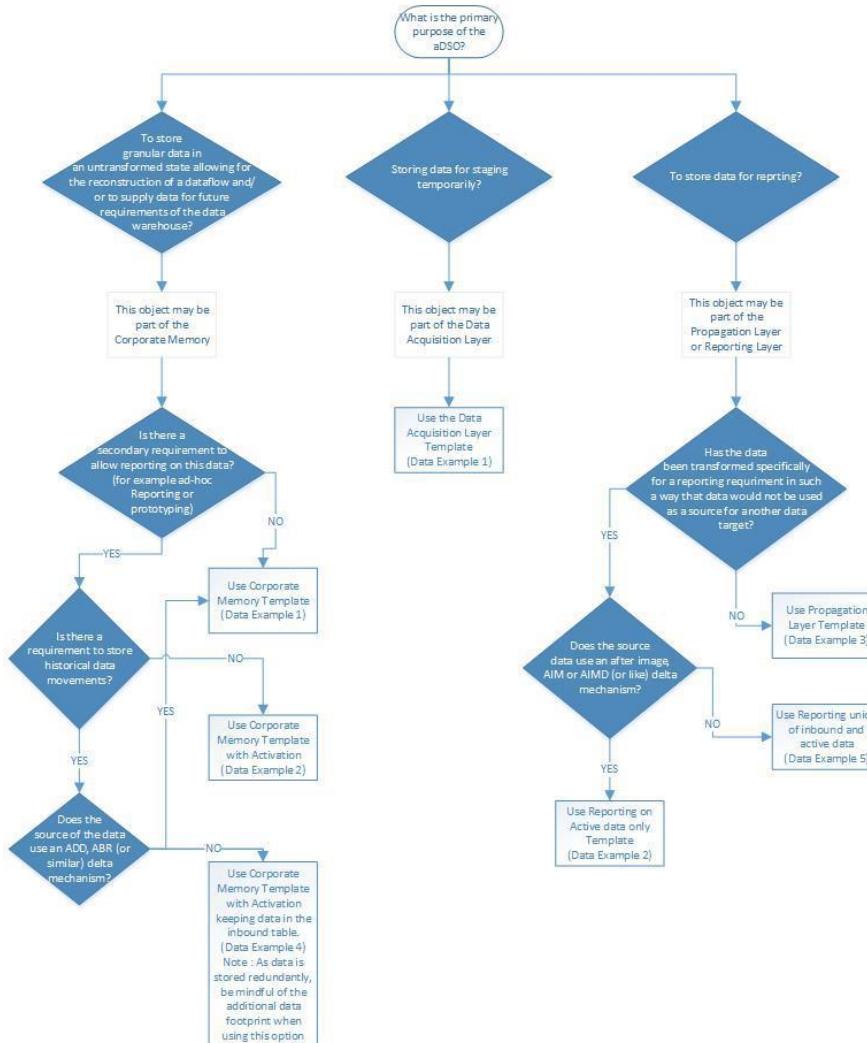
Direct Update

Planning Mode

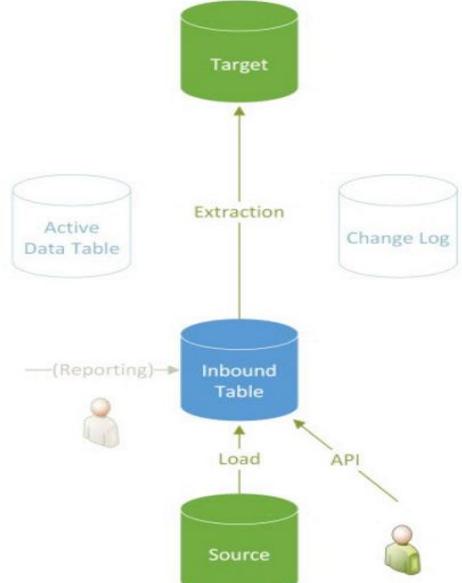
The extended table property alters the physical location of the data which does not cause the aDSO to behave differently

# Modeling properties using LSA++ templates

The flow chart below may assist in the selection of the modeling properties



# Data Acquisition Layer (Including Corporate Memory)



**Model Template**

- Enterprise data warehouse architecture
  - Data acquisition layer (including corporate memory)**
  - Corporate memory - compression capabilities
  - Corporate memory - reporting capabilities
  - Data warehouse layer - delta calculation
  - Data warehouse layer - data mart
- Planning
  - Planning on Cube-like
  - Planning on Direct Update
- Classic objects
  - Standard DataStore Object
  - Write-optimized DataStore Object**
  - InfoCube

## Modeling Properties

### Activation:

- Activate Data
  - Write Change Log
  - Keep Inbound Data, Extract from Inbound Table
  - Unique Data Records
  - Snapshot Support

### Special Types:

- Direct Update
- All Characteristics are Key, Reporting on Union of Inbound and Active Table
- Planning Mode
- Inventory

After the data loaded into the Provider, it can be found in the Inbound table:

Table: /BIC/AGA_ADS031 Displayed Fields: 7 of 7 Fixed Columns: [3] List Width 0250							
REQTSN	DATAPAKID	RECORD	RECORDMODE	/BIC/GA_EMP5	/BIC/GA_SAL	CURRENCY	
20151207123339000001000	000001	1		00000000000000000000000000000008	2.382,14	EUR	

The data doesn't get aggregated, neither by the transformation nor by the ADSO.

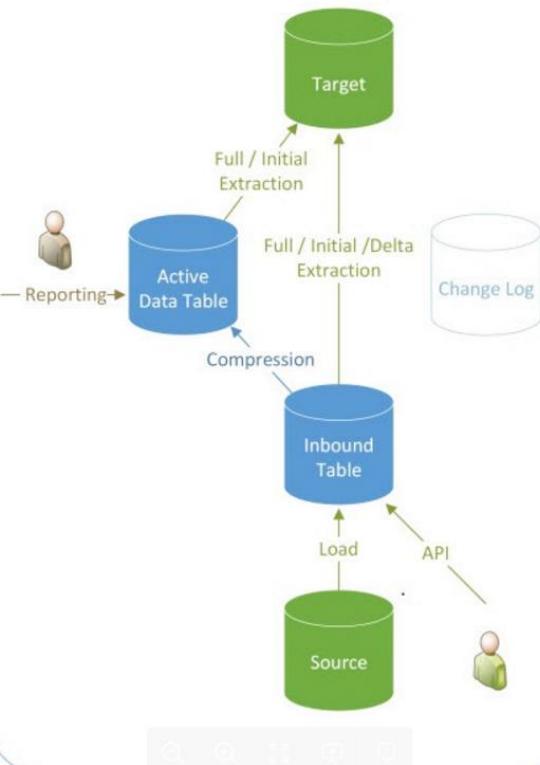
Used tables:

Inbound table /BIC/A\*1 - /BIC/AGA\_ADS031 in our particular case

No label

# Corporate memory with compression feature

Requests will still be loaded into the inbound table. Old requests that are no longer needed on detailed level can be compressed (aggregated according to the semantical key) into the active data table.



**Model Template**

- [-] Enterprise data warehouse architecture
  - [-] Data acquisition layer (including corporate memory)
  - [-] Corporate memory - compression capabilities**
  - [-] Corporate memory - reporting capabilities
  - [-] Data warehouse layer - delta calculation
  - [-] Data warehouse layer - data mart
- [-] Planning
  - [-] Planning on Cube-like
  - [-] Planning on Direct Update
- [-] Classic objects
  - [-] Standard DataStore Object
  - [-] Write-optimized DataStore Object
  - [-] InfoCube

**Modeling Properties**

**Activation:**

- Activate Data
- Write Change Log
- Keep Inbound Data, Extract from Inbound Table
- Unique Data Records
- Snapshot Support

**Special Types:**

- Direct Update
- All Characteristics are Key, Reporting on Union of Inbound and Active Table
- Planning Mode
- Inventory

## Example :

After the data loaded into the Provider, it can be found in the Inbound table:

REQUESTN	DATAPAKID	RECORD	RECORDMODE	/BIC/GA_EMPS	/BIC/GA_SAL	CURRENCY
20151207124758000008000	000001	1		00000000000000000000000000000006	5.325,60	EUR

The requests that are no longer needed on detailed level, can be compressed by activating the request itself.

DataStore Object (Advanced) GA\_ADS06  
Filter by Time from 01.11.2015 to 07.12.2015  
Filter by Status ■ ■ ■

REQUESTN	ACTIVATION REQUEST	ACTIVATED RECORDS	ERR. LOG
[2015-12-07 13:47:58 000008 CET]	1	1	0
✓ (2015-12-07 13:25:48 000008 CET)	1	1	1
✓ (2015-11-02 17:17:10 000030 CET)	1	1	1

After the request activation, the data can be found in the Active table:

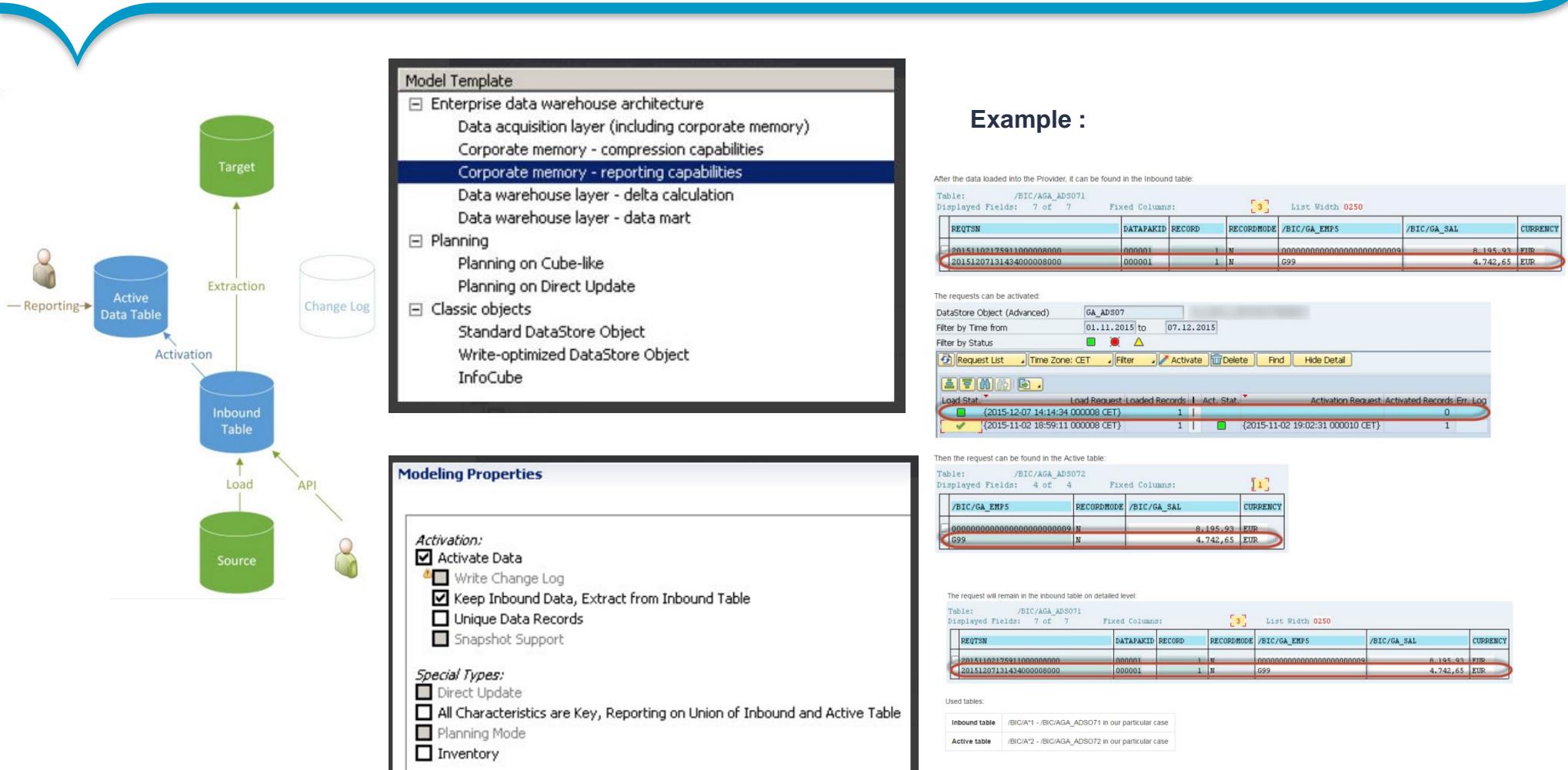
Table: /BIC/AGA_ADS062	Displayed Fields: 4 of 4	Fixed Columns: [1]	
/BIC/GA_EMPS	RECORDMODE	/BIC/GA_SAL	CURRENCY
0000000000000000000002	N	2.322,26	EUR
0000000000000000000006	N	5.325,60	EUR
0000000000000000000007	N	7.642,90	EUR

Used tables:

Inbound table	/BIC/A*1 - /BIC/AGA_ADS061 in our particular case
Active table	/BIC/A*2 - /BIC/AGA_ADS062 in our particular case

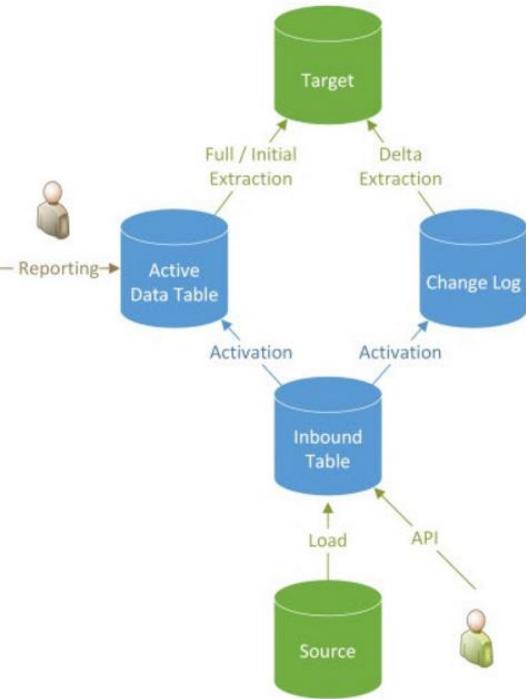
# Corporate memory with reporting option

The loaded requests can be activated but will remain in the inbound table on detailed level. This enables the user to report on data of the inbound layer without losing the detailed information.



# Datawarehouse layer- Delta Calculation

ADSO covering the "Standard" use-case of ODSOs. The following figure shows an ADSO covering the "Standard" use-case of ODSOs. Requests will be loaded into the inbound table. To report on the data, the user has to activate the loaded requests. The data is then transferred into the active data table and the history (delta) is stored in the change log. The change log is also used to rollback already activated request (recovery of the active data table)



**Model Template**

- Enterprise data warehouse architecture
  - Data acquisition layer (including corporate memory)
  - Corporate memory - compression capabilities
  - Corporate memory - reporting capabilities
  - Data warehouse layer - delta calculation**
  - Data warehouse layer - data mart
- Planning
  - Planning on Cube-like
  - Planning on Direct Update
- Classic objects
  - Standard DataStore Object
  - Write-optimized DataStore Object
  - InfoCube

**Modeling Properties**

**Activation:**

- Activate Data
- Write Change Log
- Keep Inbound Data, Extract from Inbound Table
- Unique Data Records
- Snapshot Support

**Special Types:**

- Direct Update
- All Characteristics are Key, Reporting on Union of Inbound and Active Table
- Planning Mode
- Inventory

## EXAMPLE :

After the data loaded into the Provider, the request can be found in the Inbound table:

Table: /BIC/AGAADS011						
Displayed Fields: 7 of 7 Fixed Columns: [3] List Width 0250						
REQTSN	DATAPAKID	RECORD	RECORDMODE	/BIC/GA_EMPS	/BIC/GA_SAL	CURRENCY
20151207133646000008000	000001	1	N	699		6.518,15- EUR

The request needed to be activated:

DataStore Object (Advanced)						
Filter by Time from 01.11.2015 to 07.12.2015						
Filter by Status						
Request List	Time Zone: CET	Filter	Activate	Delete	Find	Hide Detail
Load Stat	Load Request	Loaded Records: 1	CL Plan Act. Stat.	Activation Request	Activated Records: 0	Err. Log
(2015-12-07 14:36:46 000008 CET)		1				
(2015-11-03 19:09:55 000008 CET)		1				(2015-11-03 19:13:16 000014 CET) 1

The data is then transferred into the active table:

Table: /BIC/AGAADS012						
Displayed Fields: 4 of 4 Fixed Columns: [1]						
/BIC/GA_EMPS	RECORDMODE	/BIC/GA_SAL	CURRENCY			
000000000000000000000009	N	440,63-	EUR			
699	N	6.518,15-	EUR			

And to the Change Log table:

Table: /BIC/AGAADS013						
Displayed Fields: 7 of 7 Fixed Columns: [3] List Width 0250						
REQTSN	DATAPAKID	RECORD	RECORDMODE	/BIC/GA_EMPS	/BIC/GA_SAL	CURRENCY
20151103181316000014000	000001	1	N	000000000000000000000009	440,63-	EUR
20151207133946000014000	000001	1	N	699	6.518,15-	EUR

Used tables:

Inbound table	/BIC/A*1 - /BIC/AGAADS011 in our particular case
Active table	/BIC/A*1 - /BIC/AGAADS012 in our particular case
Change log table	/BIC/A*1 - /BIC/AGAADS013 in our particular case

# Agenda

- Introduction to SAP HANA Architecture
- SAP Business Warehouse (BW)
- SAP HANA Studio
- Data Warehousing Workbench
- Info Objects like Characteristics and Key Figures using Eclipse
- Master Data Loading from SAP source system with Operational Data Provisioning .
- Generic Data Sources
- Transformation and Data Transfer Process
- Difference between Classic and Graphical Data Flow Modeling
- Modeling Data Store Object (Advanced)
- Transactional Data Loading from SAP source system with Operational Data Provisioning
- Data Loading from Flat file
- Modeling Composite Providers
- HANA Native Modeling with Calculation Views
- Modeling Open ODS Views
- Process Chains

# Flat file loading through excel or csv files

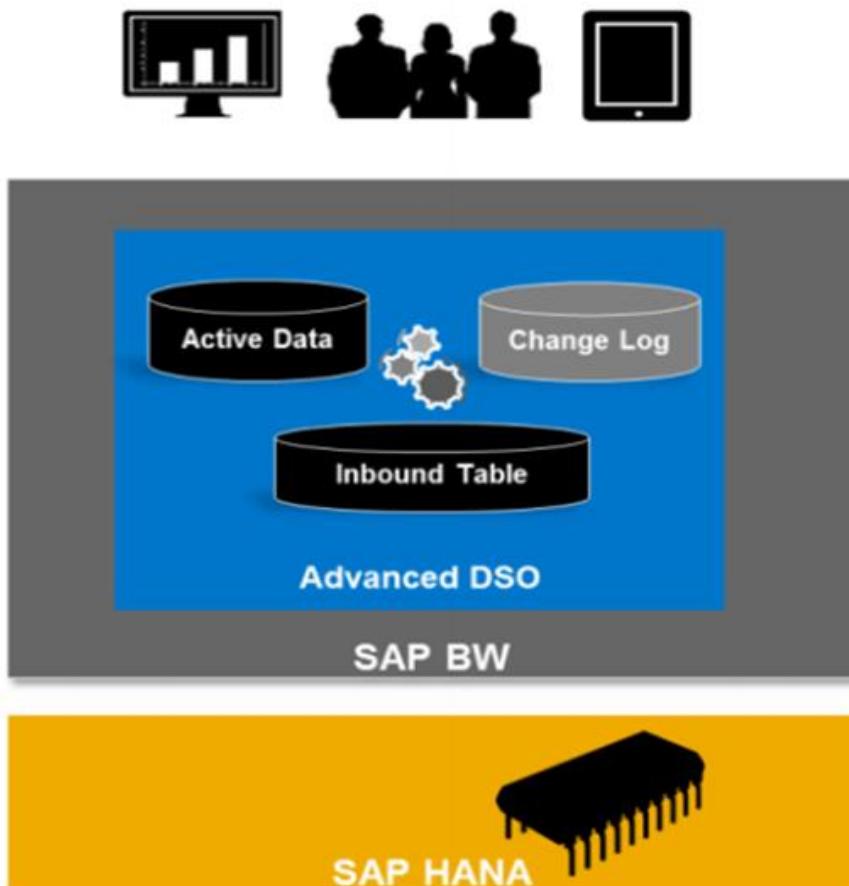
Below are steps involved in loading data from flat files (csv or excel) to SAP HANA database using SAP HANA Studio



# Agenda

- Introduction to SAP HANA Architecture
- SAP Business Warehouse (BW)
- SAP HANA Studio
- Data Warehousing Workbench
- Info Objects like Characteristics and Key Figures using Eclipse
- Master Data Loading from SAP source system with Operational Data Provisioning .
- Generic Data Sources
- Transformation and Data Transfer Process
- Difference between Classic and Graphical Data Flow Modeling
- Modeling Data Store Object (Advanced)
- Transactional Data Loading from SAP source system with Operational Data Provisioning
- Data Loading from Flat file
- Modeling Composite Providers
- HANA Native Modeling with Calculation Views
- Modeling Open ODS Views
- Process Chains

# SAP BW Composite Provider



The Composite Provider is a new InfoProvider type introduced with BW7.30. It offers the possibility to combine InfoProviders and HANA artefacts via UNION and JOIN.

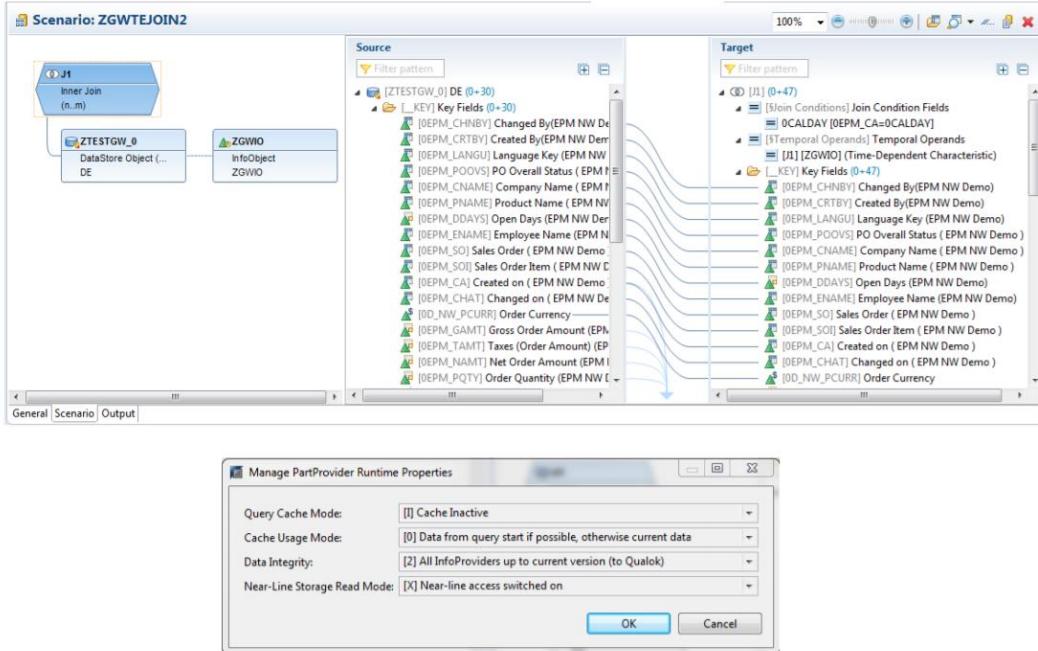
One characteristic of the Composite Provider is, that one “initial” Provider has to be taken over to the Composite Provider with a UNION connection as a basis.

Successor to BW MultiProvider, BW InfoSet, existing CompositeProvider, Transient Provider on HANA Models, Virtual Provider on HANA Models

UNION and JOIN operations are executed in HANA and not the application server.

BEx Queries can be created on Composite Providers as on any other BW InfoProvider.

# Simplification –Composite Provider New with SAP BW 7.5 SP1



## Simplified User Experience

- Easy creation process
- Conversion of existing MultiProvider and “old” CompositeProvider
- InfoArea Assignment
- Semantic grouping of output structures

## New with SAP BW 7.5 SP4\*

- Support of CompositeProvider with temporal joins to fully replace InfoSets
- Transfer tool from InfoSet to Composite Provider (incl. Queries)
- Runtime properties for Query Execution of each PartProvider available in Eclipse

# SAP BW Composite Provider(Use Cases)

Currently exist three major use cases for leveraging a Composite Provider

## I.BW Workspaces –

Composite Provider allows to quickly and easily combine central BW data with uploaded local data (Excel, csv, ...). This is done in the Workspace Designer

## II.Rapid - Prototype Models

– IT can quickly and flexible combine AnalyticIndexes created in APD or published SAP HANA Models). This is done in the transaction **RSLIMO**.

## III.JOINs between Info Provider

– Certain BW InfoProvider can be combined using the JOIN operation allowing IT to create new scenarios not possible or very expensive with standard techniques (MultiProvider, InfoSet). This is done in transaction **RSLIMOBW**

.

# Agenda

- Introduction to SAP HANA Architecture
- SAP Business Warehouse (BW)
- SAP HANA Studio
- Data Warehousing Workbench
- Info Objects like Characteristics and Key Figures using Eclipse
- Master Data Loading from SAP source system with Operational Data Provisioning .
- Generic Data Sources
- Transformation and Data Transfer Process
- Difference between Classic and Graphical Data Flow Modeling
- Modeling Data Store Object (Advanced)
- Transactional Data Loading from SAP source system with Operational Data Provisioning
- Data Loading from Flat file
- Modeling Composite Providers
- HANA Native Modeling with Calculation Views
- Modeling Open ODS Views
- Process Chains

# Calculation view

A Calculation View is a powerful and flexible information view, which you can use to define more advanced slices on the data available in the SAP HANA database. Calculation views are simple and they mirror the functionality found in both attribute views and analytic views and much more.

## **There are two types of calculation views**

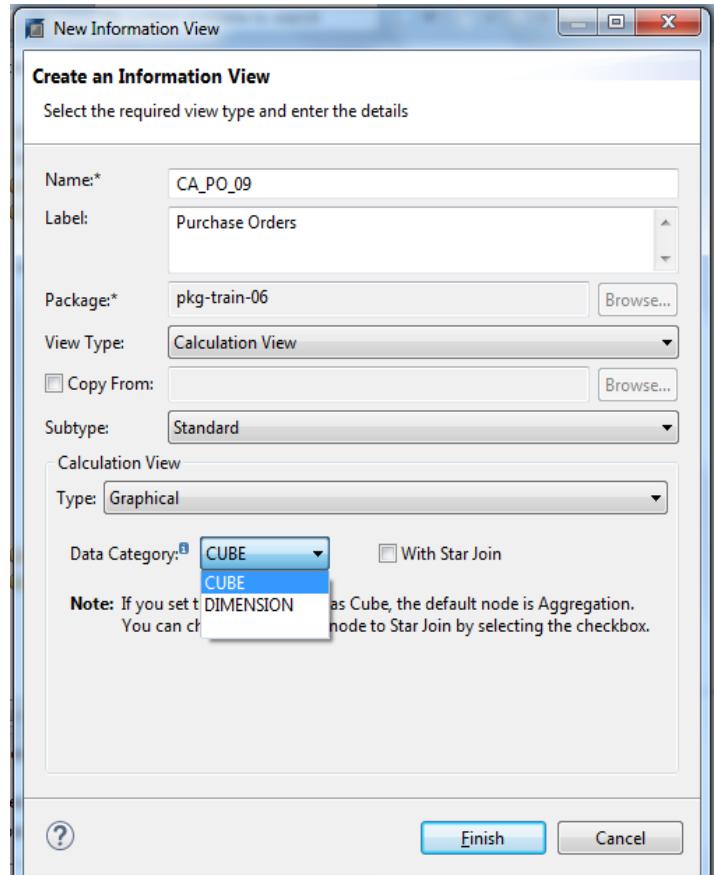
- Graphical calculation views, which can be created using a graphical editor.
- Script-based calculation views, which can be created by writing SQL scripts.

## **Characteristics of calculation views:**

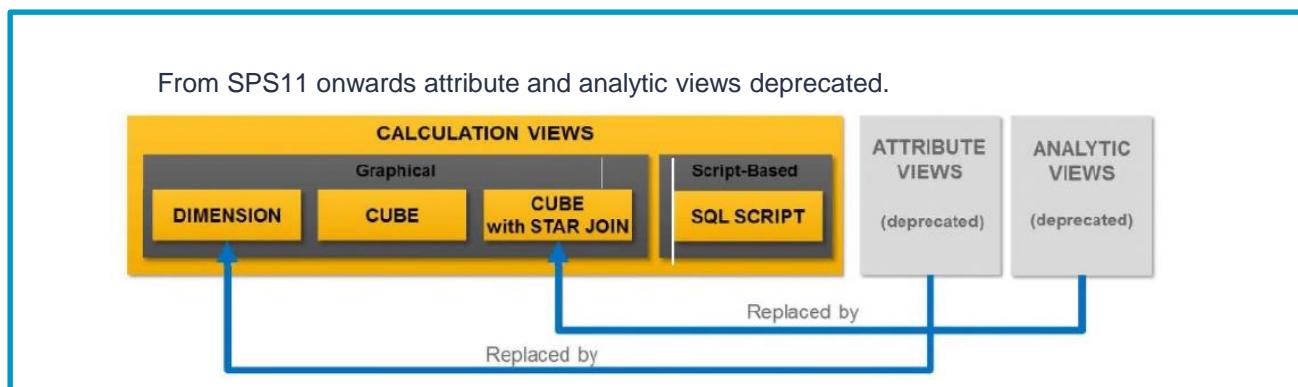
- Supports both OLAP and OLTP models.
- Supports complex expressions (i.e. IF, Case, Counter).
- Supports reusing Analytic views, Attribute views and other Calculation views (Graphical and Scripted).
- Supports analytic privileges (i.e. restricting a user for a certain cost center).
- Supports SAP ERP specific features (i.e. client handling, language, currency conversion).
- Provides ability to combine facts from multiple tables.
- Provides support for additional data processing operations, (i.e. Union, explicit aggregation).
- Provides ability to leverage specialized languages (i.e. R-Lang).
- Provides ability to leverage both Column and Row tables.

# Graphical calculation view

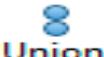
## Graphical Calculation view Data Category:



Calculation view type	Properties	Default Upper Node
[Blank]	No multidimensional support. Never exposed to any client tool	Projection
DIMENSION	No multidimensional support. Equivalent to an attribute view *	Projection
CUBE	Designed for data analysis with Multidimensional reporting	Aggregation
CUBE with Star Join	Similar to a CUBE Calculation view , but the upper node is a Star Join where you join all the attributes (calculation views of type DIMENSION *)	Star Join
* Only DIMENSION Calculation View can be used as a data source in the Star Join node of a CUBE With Star Join Calculation View.		



# Graphical calculation view

Icon	View Node	Description	Example
 Union	Union	Use union node to combine the result set of two or more data sources. Union nodes have two or more inputs.	For example, for retrieving the names of all employees of a store, which has different branches and each branch maintaining its own employee records table.
 Join	Join	Use join node to query data from two or more data sources, based on a specified condition. Join nodes have two inputs.	For example, for retrieving customer details and location based on the postal code column present in the two tables CUSTOMER and GEOGRAPHY. The CUSTOMER table has columns – Customer_ID, Customer_Name, Postal_Code, and GEOGRAPHY table has columns – Postal_Code, Region, Country.
 Proj...	Projection	Use projection node to filter or obtain a subset of required columns of a table or an information view. Projection nodes have one input.	For example, for selecting the employee name and employee department from a table consisting of many other columns.
 Aggr...	Aggregation	Use aggregation node to summarize data for a group of row values, by calculating values in a column. Aggregation nodes have one input.	For example, for retrieving total sales of a product in a month. The supported aggregation types are sum, min, and max.
 Rank	Rank	Use rank node to partition the data for a set of partition columns, and performs an order by operation on the partitioned data.	For example, consider a TRANSACTION table with two columns PRODUCT and SALES. If you want to retrieve the top five products based on its sales, then use a rank node.

# Graphical calculation view

## Main Supported Data Sources Types in Calculation Views

- Column Tables from the same database
- Column Tables from another database of the same HANA System, In a MultiDatabase Container configuration
- Core Data Services (CDS) Entities and SQL Views.
- Table Functions
- Virtual Tables (data provisioned through SAP HANA Smart Data Access)
- Analytic Views (deprecated)
- Script-based Calculation Views (deprecated)

# Graphical calculation view

**Data Category** : For Calculation Views, determines whether the view supports multidimensional reporting.

**Type**: Standard, Time or Derived \*

**Default Client**: Defines how to filter data by SAP client (MANDT).

**Default Schema** : Defines which schema must be used when the source schema of a catalog object is not specified.

**Apply Privileges**: Specifies the type of Analytic Privileges (Standard or SQL) to apply when executing a view.

**Enable History** : The value of this property determines whether your calculation view supports time travel queries.

**History Input Parameter** : Specifies which Input Parameter must be used to specify the timestamp in time travel queries.

**Column View Name**: This field contains the full name of the column view in the \_SV'S\_6/C schema. You can easily copy/paste this name to the code of a SQL statement.

**Deprecate** : Identifies views that are not recommended for reuse, though still supported in SAP HANA Modeler.

**Translate** : Determines whether the view labels can be translated into one or several additional languages.

**Execute in** : Determines whether the model must be executed by the SQL engine or column engine.

**Cache Invalidation Period** : If the view data are cached, determines whether the cache must be deleted on a daily or hourly basis, or after each transaction that modifies any of the underlying tables.

## Semantics node: View Properties tab

The screenshot shows the 'View Properties' tab for a Semantics node named 'CV\_SALES'. The node is associated with the package 'HAN.pkg-Train04'. The 'General' section contains the following properties:

Data Category:	CUBE	Enable History
Type:	STANDARD	History Input Parameter:
Default Client:	Session Client	Default Member:
Default Schema:	[...]	Column View Name: "SYS_BIC"."pkg-Train04/CV_SALES"
Apply Privileges:	Classical Analytic Privilege	Deprecate
		Translate

The 'Advanced' section contains the following properties:

Execute In:	[dropdown]	
Cache Invalidation Period:	[dropdown]	
Execution Hints		
Name	Value	Add
<Click to add>		Remove

A warning message in the 'Advanced' section states: "⚠ These properties may affect the output data. Please set them cautiously."

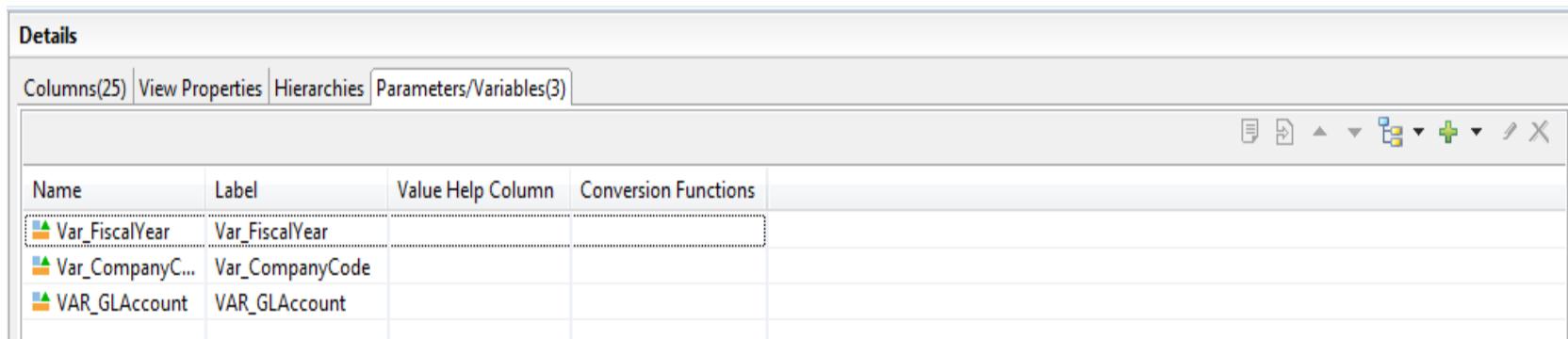
# Graphical calculation view

Semantics node: select the Hierarchies tab and click the '+' button in the Hierarchy pane:



The screenshot shows a software interface for managing hierarchies. At the top, there's a navigation bar with tabs: 'Columns(25)', 'View Properties', 'Hierarchies' (which is selected and highlighted in blue), and 'Parameters/Variables(3)'. Below the navigation bar is a section titled 'Local'. Under 'Local', there is a table with two columns: 'Name' and 'Label'. The table is currently empty. To the right of the table are several icons for managing data: a file icon, a copy icon, a plus sign icon (the 'create' button), a pencil icon, and a delete icon.

Semantics node: select the Parameters/variables tab and click the '+' button ,create variable and Input parameters.



The screenshot shows a software interface for managing parameters. At the top, there's a navigation bar with tabs: 'Columns(25)', 'View Properties', 'Hierarchies', and 'Parameters/Variables(3)' (selected and highlighted in blue). Below the navigation bar is a table with four columns: 'Name', 'Label', 'Value Help Column', and 'Conversion Functions'. There are three rows in the table, each representing a parameter:

Name	Label	Value Help Column	Conversion Functions
Var_FiscalYear	Var_FiscalYear		
Var_CompanyC...	Var_CompanyCode		
VAR_GLAccount	VAR_GLAccount		

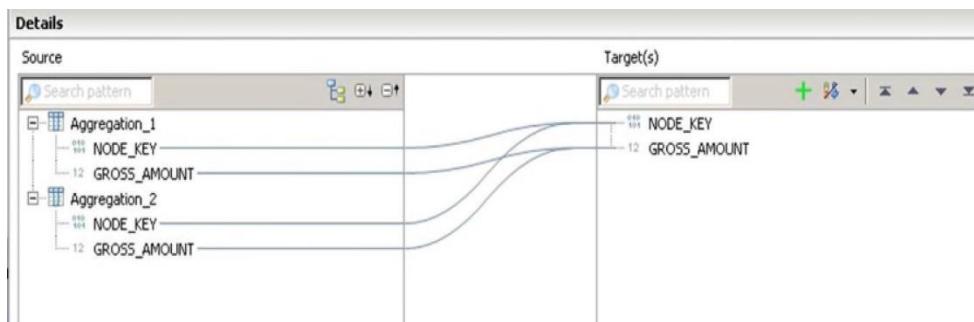
To the right of the table are icons for managing data: a file icon, a copy icon, up and down arrows for sorting, a plus sign icon (the 'create' button), a pencil icon, and a delete icon.

# Graphical calculation view – Union

## Union Node

- If you want to combine multiple result sets with identical structures into one result set, you can use a union node. Depending on how different the column names from the data source are, you can map the columns automatically by name (columns with identical names will be mapped automatically) or adjust the mapping manually.
- A mapping of the sources to the target is required and will allow you to adapt structural differences.

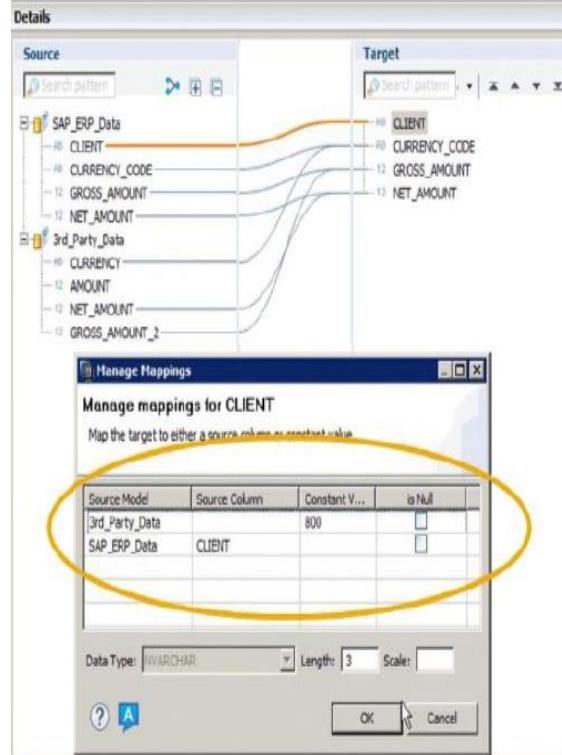
This can be done via a drag and drop interface.



Depending on the requirement, you can use one of the following approaches:

- A standard Union
- A Union with Constant Values

## Union with Constant Values : Unmapped Columns in a Union Node



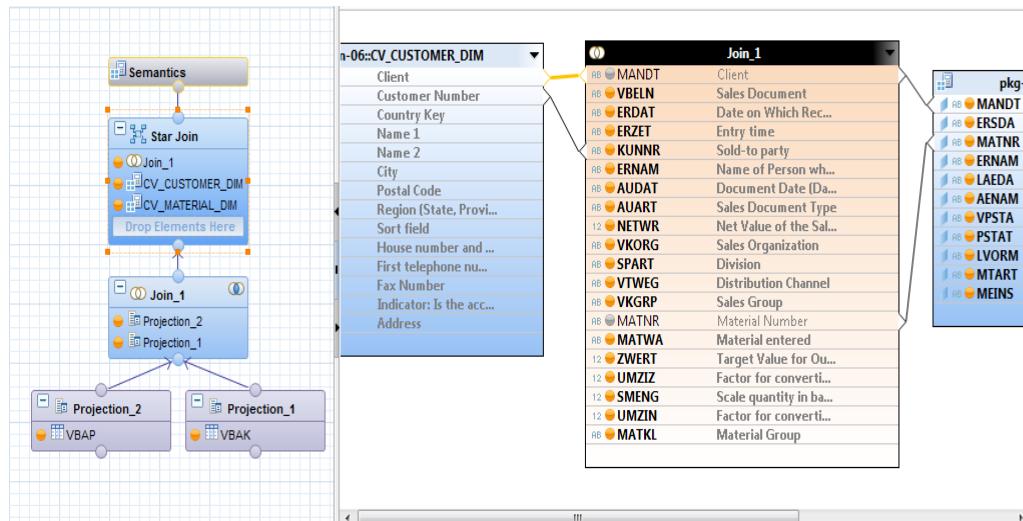
- There could be instances when a union needs to be performed where the sources have a different number of columns.
- You can then set a constant value for the source columns that do not have the target column.
- To set the constant value, right click the target column and choose "Manage Mappings".

# Graphical calculation view - Star Join

## Star join

The Star Join in Calculation Views of type *CUBE with Star Join* is a Node Type, rather than an Join type. It is used to structure data in a star schema. The fact table (data source) of a star join can be any type of input node. However, only Calculation Views of Data Category Dimension are allowed as input nodes for dimensions.

The type of joins between the fact and dimension tables within the star schema can be defined in the star join node.



The screenshot shows the "Edit Join" dialog for a join definition named "Join\_1".

**Properties:**

Property	Value
Left Element	Join_1.MANDT
Right Element	pkg-train-06::CV_CUSTOMER_DIM
Join Type	Referential
Language Col...	
Cardinality	
Dynamic Join	False
Optimize Join...	False

**Left Table:** join\_1

**Right Table:** pkg-train-06::CV\_CUSTOMER\_DIM

**Left Column:** MANDT, VBELN, ERDAT, KUNNR, ERZET, NETWR, VKORG, SPART, VTWEG, VKGRP, MATNR, AENAM, VPSTA, PSTAT, LVORM, MTART, MEINS

**Right Column:** MANDT, KUNNR, LAND1, NAME1, NAME2

**Properties:**

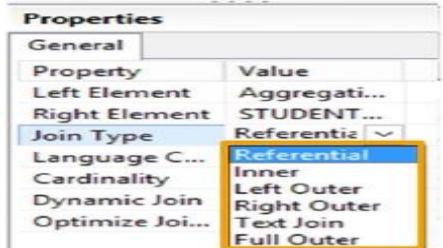
Property	Value
Left Element	Join_1.MANDT
Right Element	pkg-train-06::CV_CUSTOMER_DIM
Join Type	Referential
Language Column:	

**Temporal Properties:**

Temporal Column:	From Column:

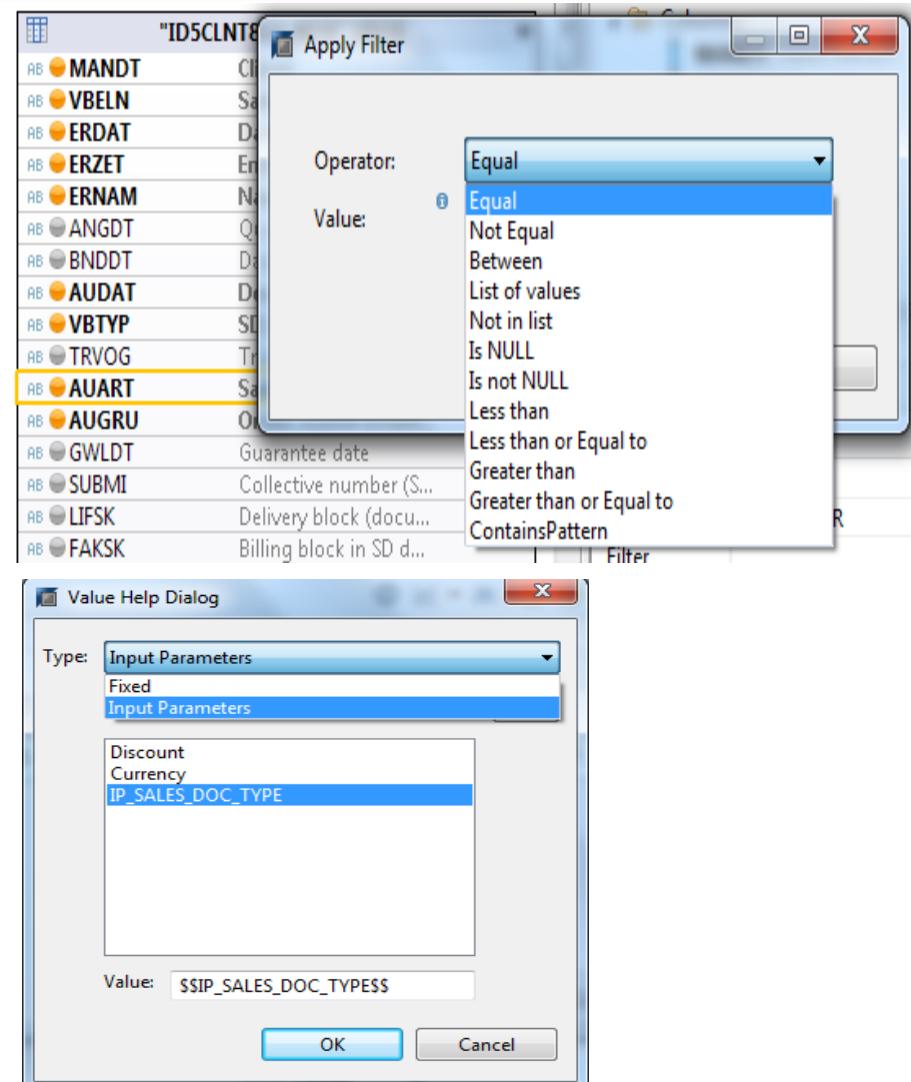
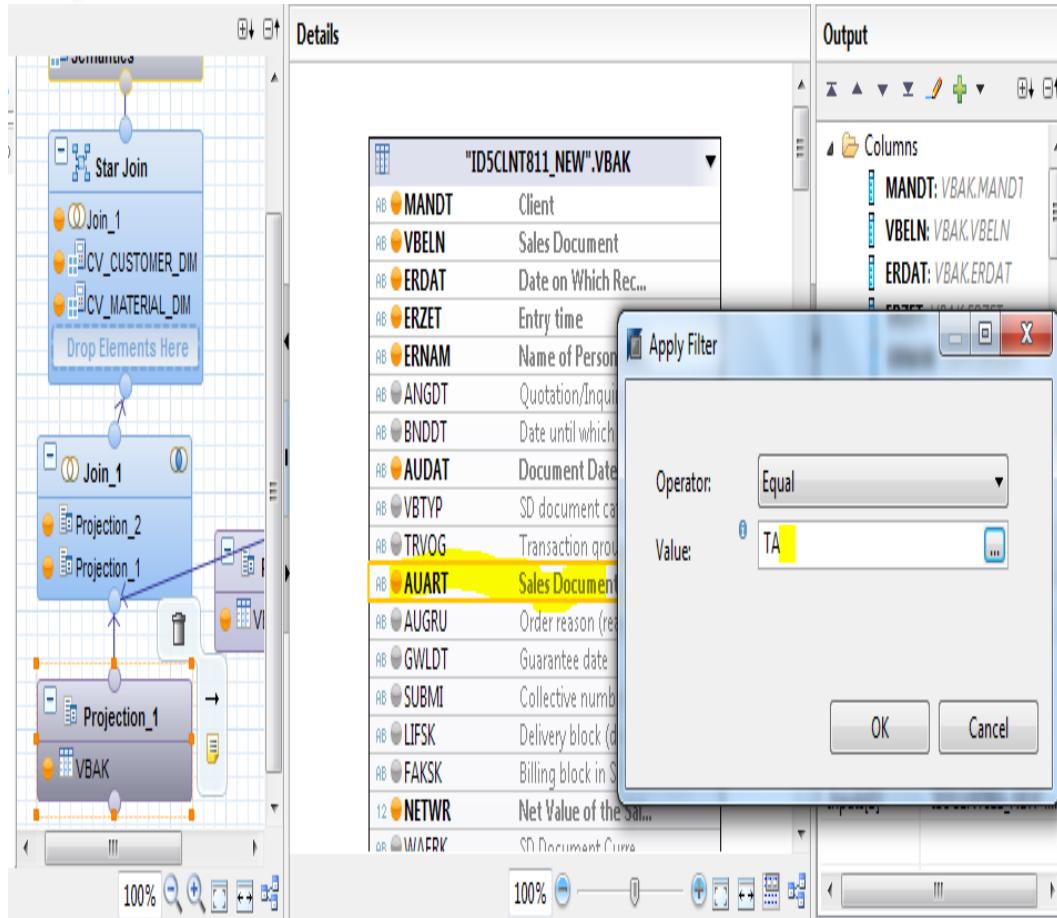
**Buttons:** Validate Join, OK, Cancel

Join type properties within the star join node:



# Graphical calculation view - Filter

## Filter



# Graphical calculation view - Calculated Column

## Calculated Column

A calculated column is defined within an information model and can use the string functions, mathematical functions etc. available in the editor. You can define a calculated column as measure or attribute. Double-click or drag-and-drop Elements, Operators and Functions to build the expression.

The screenshot shows the SAP BusinessObjects Data Services graphical interface. On the left, there is a navigation tree with nodes like 'Semantics', 'Aggregation', 'Projection\_3', 'Join\_1', 'Projection\_1', and 'Projection\_2'. In the center, there is a list of attributes from an information model, including VTWEG, KUNNR, POSNR, MATNR, MATWA, PMATN, CHARG, MATKL, ARKTX, PSTV, POSAR, LFRREL, ZWERT, ZMENG, UMZIZ, UMZIN, SMENG, NETWK\_1, NETPR, SKTOF, and UMREF. The 'SKTOF' attribute is selected. To the right, a 'Create a Calculated Column' dialog box is open. The dialog shows the following details:

Name*	Total_Price
Label:	Total_Price
Data Type:	DECIMAL
Column Type:	Measure
Client Aggregation:	Formula
Hidden:	<input type="checkbox"/> Enable client side aggregation

The 'Expression Editor' section contains the expression: "SMENG""NETPR". Below it, the 'Elements' panel lists 'MANDT: Projection\_3.MANDT' and 'VBELN: Projection\_3.VBELN'. The 'Operators' panel includes standard arithmetic operators (+, -, \*, /, %, (), \*\*). The 'Functions' panel includes 'Conversion Functions', 'String Functions', and 'Mathematical Functions'. At the bottom right of the dialog are 'OK' and 'Cancel' buttons.

# Graphical calculation view - Restricted Column

## Restricted Column

The restricted column is restricted based on one or more attributes. These columns can be anything in the base table or view that the modeler defines in order to help reporting or further modeling. A restricted column can be created in an aggregation node of a calculation view. You assign a measure to the restricted column, and also which columns define the restriction.

The screenshot shows two identical dialog boxes for 'Edit Restricted Column Definition'. Both dialogs have a title bar 'Edit Restricted Column' and a main area with the following fields:

- Name:** Restricted\_Price
- Label:** Restricted\_Price
- Column:** NETPR
- Hidden:** (checkbox)

Below these fields is a section titled 'Restrictions' with the following settings:

- Column:** (radio button selected)
- Expression:** (radio button)

Column	Operator	Value	Include
VBTYP	Equal	C	<input checked="" type="checkbox"/>
VTWEG	Equal	10	<input checked="" type="checkbox"/>
SPART	Equal	90	<input type="checkbox"/>

At the bottom of each dialog are 'OK' and 'Cancel' buttons.

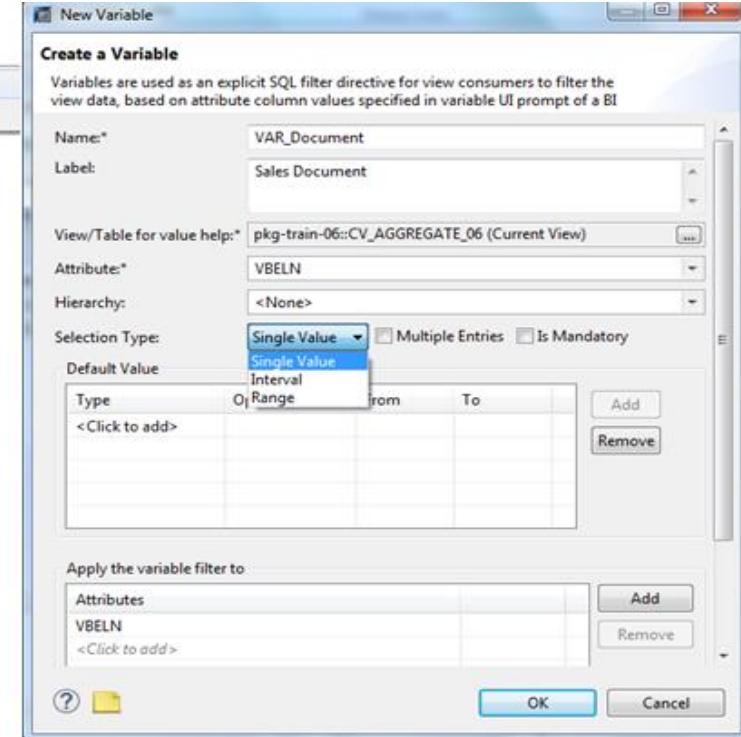
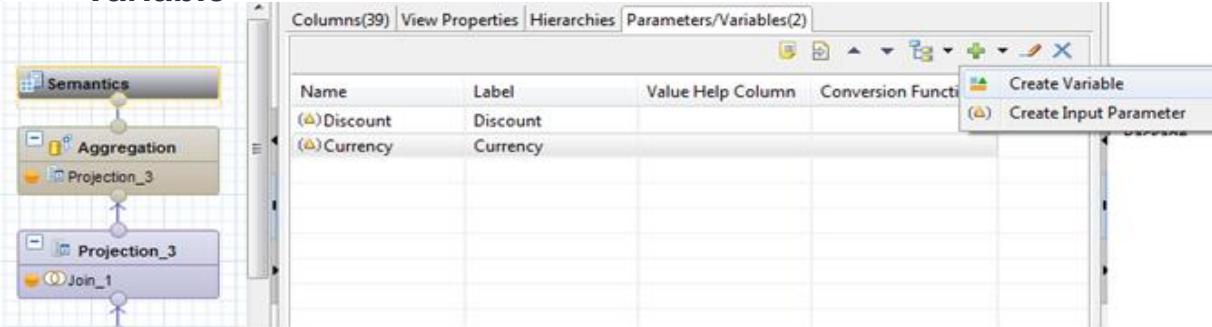
CONFIDENTIAL – Please do not share

Presentation Title | Date

Copyright © Capgemini 2017. All Rights Reserved

# Graphical calculation view - Variable

## Variable



**View/Table for Value Help and Attribute:** These settings define which view/table and which attribute from this view/table is used as a reference to provide a list of values at runtime.

**Selection Type:** Whether selections should be based on intervals, ranges or single values.

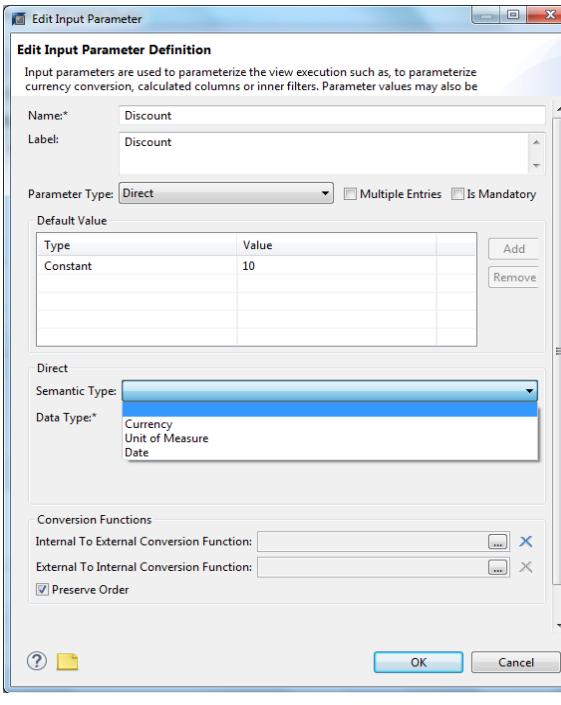
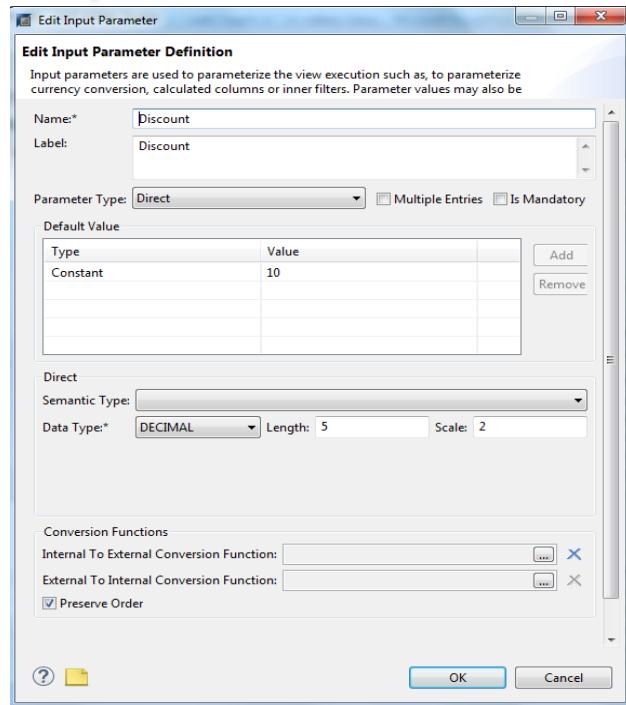
**Multiple Entries:** Whether multiple occurrences of the selection type are allowed.

**Is Mandatory:** Check if the variable is Mandatory or Optional

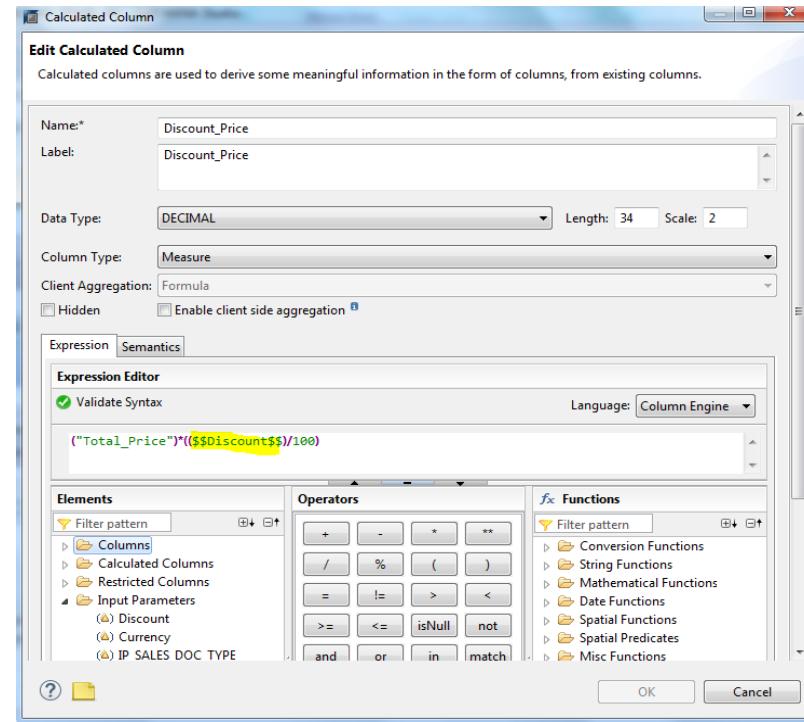
Variable/Input Parameter	Operator	From	To	Value Type	Selection Type
(*) Discount	Equal	10		Attribute Value	Single Value
(*) VAR_Document	Equal			Attribute Value	Single Value

# Graphical calculation view - Input Parameter

## Input Parameter



## Input Parameter used in Calculated Column



**Direct:** For currency conversion, when you want the end user to specify a source or target currency.

For Date, To retrieve data based on a date entered by the end user (or chosen in a calendar type input box).

For Unit of Measure, To retrieve data based on a unit of measure chosen by the end user.

**Static List:** To provide the end user with a predefined list of values in which he/she chooses one or several items.

**Column:** To provide the end user with a list of values from a column of the information model

**Derived From Table:** When you want the end user to have a set list of values from a table (not necessarily included in the view)

**Derived From Procedure:** When you want the parameter value to be passed to the information model based on the scalar output of a stored procedure.

**Direct (without semantic type):** When none of the above applies and/or when you want the user to enter a parameter without choosing it from a predefined list.

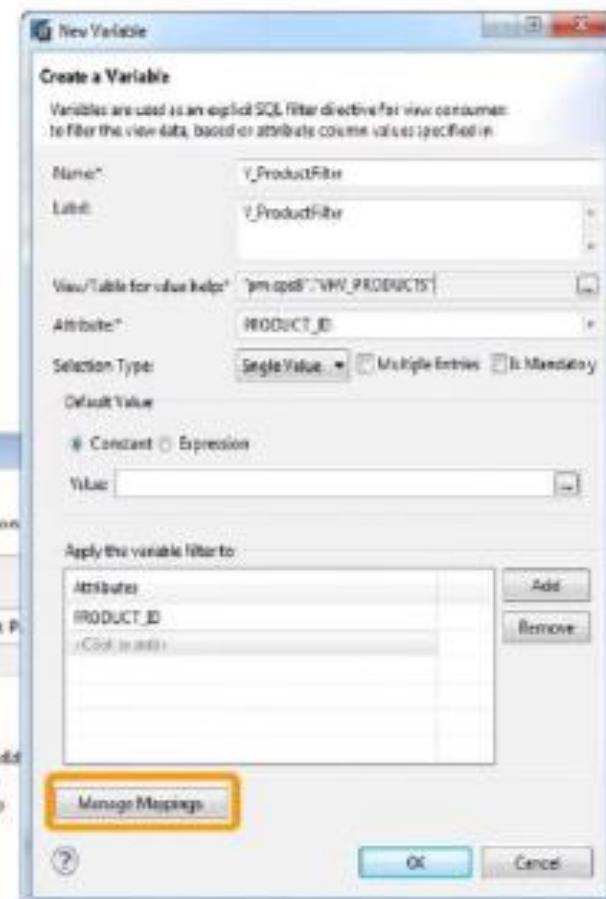
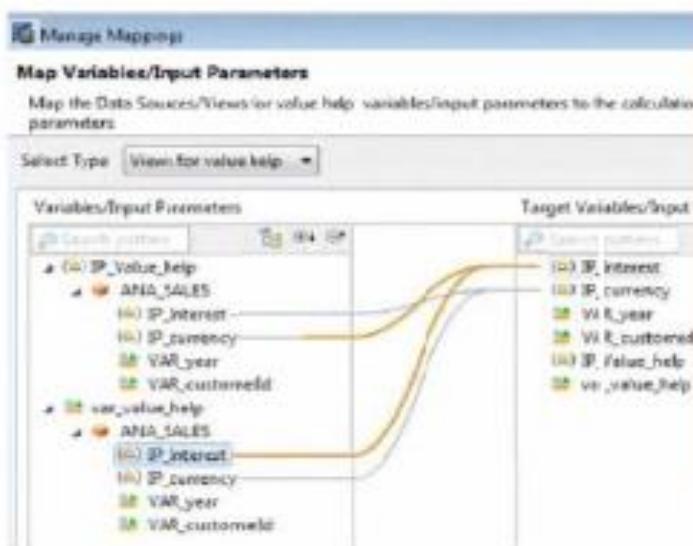
# Graphical calculation view

## Passing Parameters to External Views for Value Help

- Variables and Input Parameter can be mapped to variables and input parameters from external views.
- Allows filtering and customizing value help lists from external views.
- Supported with Analytic and Calculation Views (Graphical and Script).

A Manage Mapping dialog box can be opened from:

- The variable/input parameter creation dialog box
- The "Semantics" node



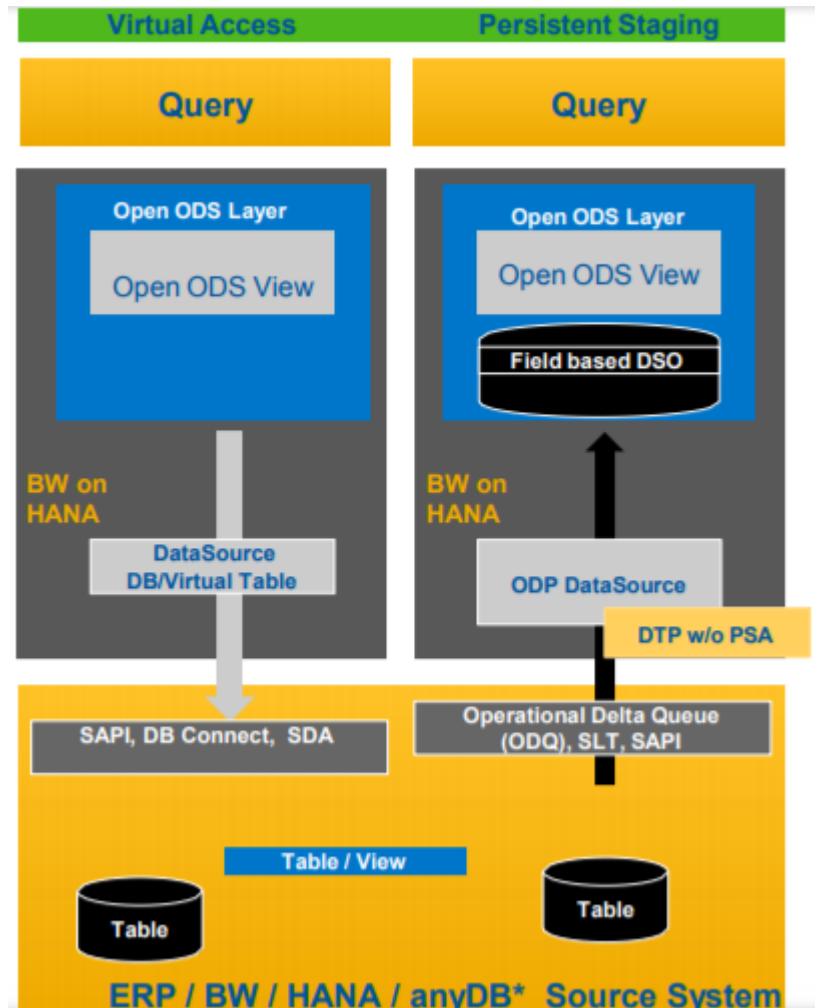
# Agenda

- Introduction to SAP HANA Architecture
- SAP Business Warehouse (BW)
- SAP HANA Studio
- Data Warehousing Workbench
- Info Objects like Characteristics and Key Figures using Eclipse
- Master Data Loading from SAP source system with Operational Data Provisioning .
- Generic Data Sources
- Transformation and Data Transfer Process
- Difference between Classic and Graphical Data Flow Modeling
- Modeling Data Store Object (Advanced)
- Transactional Data Loading from SAP source system with Operational Data Provisioning
- Data Loading from Flat file
- Modeling Composite Providers
- HANA Native Modeling with Calculation Views
- Modeling Open ODS Views
- Process Chains

# Modeling Open ODS Views

Open ODS views allow the flexible integration of external data sources into SAP BW on HANA, without the need to create InfoObjects. Open ODS views enable you to define data models for objects like database tables, database views or Data Sources (for direct access).

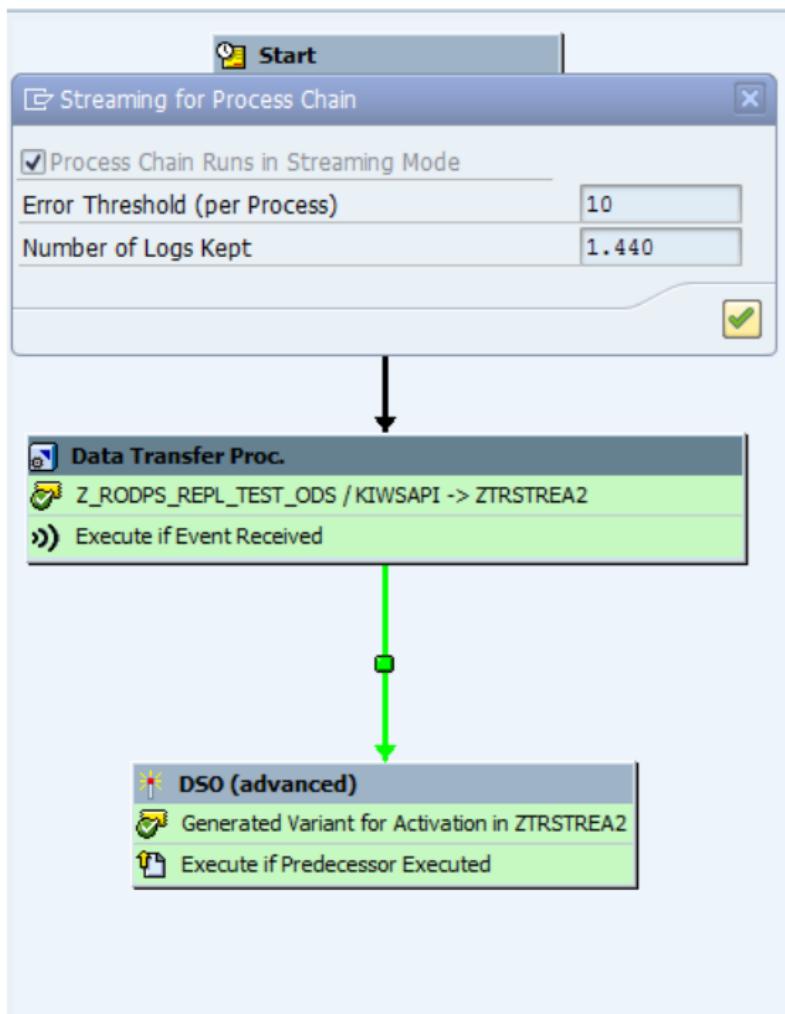
- Represents a view on a source and adds analytic metadata to it.
- Does not have separate storage for transaction data or master data.
- You can specify whether a specific field should be interpreted as a key figure or characteristic
- Open ODS views are available if the BW system is running on the SAP HANA database.
- Open ODS views cannot be defined for hierarchies.



# Agenda

- Introduction to SAP HANA Architecture
- SAP Business Warehouse (BW)
- SAP HANA Studio
- Data Warehousing Workbench
- Info Objects like Characteristics and Key Figures using Eclipse
- Master Data Loading from SAP source system with Operational Data Provisioning .
- Generic Data Sources
- Transformation and Data Transfer Process
- Difference between Classic and Graphical Data Flow Modeling
- Modeling Data Store Object (Advanced)
- Transactional Data Loading from SAP source system with Operational Data Provisioning
- Data Loading from Flat file
- Modeling Composite Providers
- HANA Native Modeling with Calculation Views
- Modeling Open ODS Views
- Process Chains

# Streaming Process Chains



## Streaming is a New Operation Mode of a Process Chain

- Process data in real-time
- High frequent execution

## Streaming Mode as Central Property Per Process Chain

Activated via Process Chain -> Attributes -> Streaming Mode  
Process types that can trigger the chain in streaming mode:

*DTP (Hana Data Source, ODP) or START*

Process types that can be used and run in streaming mode:  
*ADSO Activation, ABAP Program, Decision, AND, OR, EXOR*

## Improved Streaming Mode Monitoring

## Specific Behavior for Failed Processes and Logging

The background of the slide features a large, dense school of fish swimming in a circular pattern, creating a sense of movement and depth. The fish are silvery-blue and appear to be Barracudas.

# Questions?



People matter, results count.

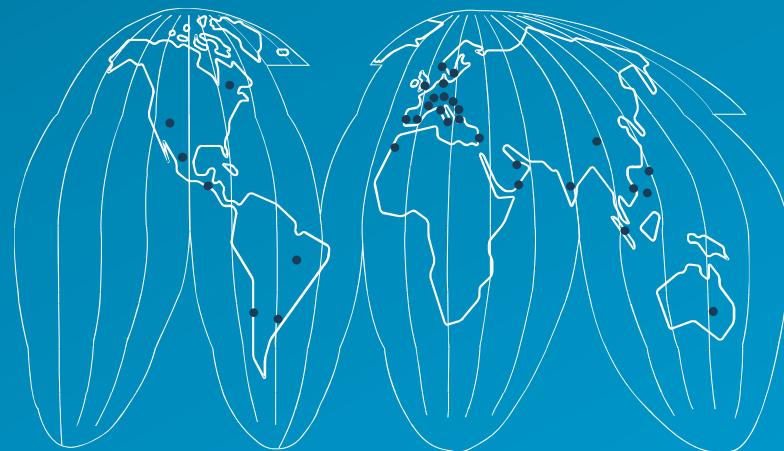


## About Capgemini

With more than 120,000 people in 40 countries, Capgemini is one of the world's foremost providers of consulting, technology and outsourcing services. The Group reported 2011 global revenues of EUR 9.7 billion.

Together with its clients, Capgemini creates and delivers business and technology solutions that fit their needs and drive the results they want. A deeply multicultural organization, Capgemini has developed its own way of working, the Collaborative Business Experience™, and draws on Rightshore®, its worldwide delivery model.

*Rightshore® is a trademark belonging to Capgemini*



**[www.capgemini.com](http://www.capgemini.com)**

