

# Modelling in SAP HANA vs BW/BPC

## How to Benefit from Mixed Scenarios

### Table of Contents

What is a Mixed Scenario? .....	1
Add Calculations with HANA Views .....	2
Planning by Week Days .....	4
Addressing Data Protection (GDPR) Requirements.....	5
Final Words .....	5

For an organization involved in SAP HANA implementation it is hard to choose the right tools, skills, and resources needed for project's success. If we specifically look at existing Data Warehousing products from SAP, certain SAP BW components like master data capabilities, non-cumulative modelling, business planning functionality, analysis authorizations are not easy to design from scratch in HANA native. On the other hand, there is a lot of built-in functionality in SAP HANA, functions from business libraries (BFL, AFL, PAL), flexibility of calculation views, data masking features, etc. that does not really make sense to redesign on the ABAP side.

Below I review a few use cases and scenarios showcasing mixed modelling approaches. The focus is on so called "mixed scenarios" recommended by SAP where we can benefit from both BW and HANA native development.

### What is a Mixed Scenario?

According to SAP, a **Mixed Architecture** consists of an overall data model that is implemented at the same time by BW and native HANA tools. It combines processes, data and metadata of BW and native HANA, bringing the best capabilities of both worlds to gain flexibility and insight.

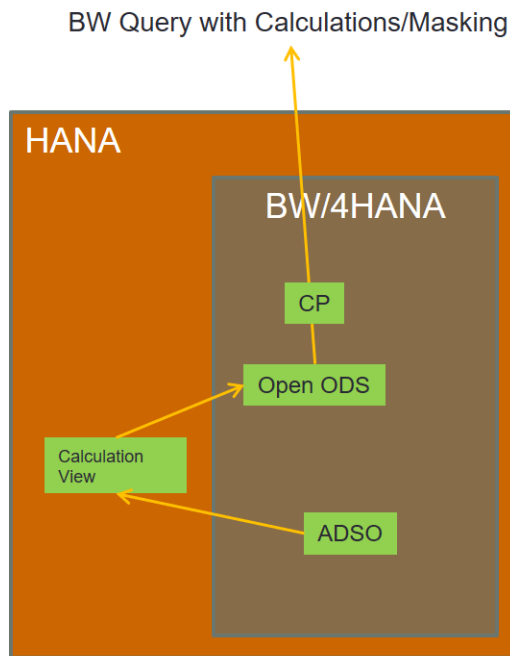
For more information on Mixed Architecture definition please check the following slide deck from SAP: <https://www.sap.com/documents/2016/06/d2cb3a52-767c-0010-82c7-eda71af511fa.html>

The reasoning behind applying mixed scenarios is that certain functionality is more efficient to implement in HANA native, as opposed to some other functionality would be easier to deliver via BW. For example, in favor of HANA native modelling there is a possibility of creating calculation views on top of multiple database tables with new columns added via complex calculations or formulas. BW (Bex) queries on one hand are missing certain flexibilities, but on the other hand provide features that are currently not available in HANA views, such as reusable structures, hierarchy variables, etc.

## Add Calculations with HANA Views

There are numerous examples of complex calculations to be done “on the fly” without involving additional persistency. In certain scenarios values for new fields may depend on a combination of fields and input parameters.

### Age Range based on DOB and Key Date



view when applying mixed architecture.

Let us look at a scenario where we have a person’s date of birth coming with one of the fields and additionally a key date parameter is passed to a query. Our objective is to produce a new calculated column (Age Range) classifying each person into the following age bucket:

- Younger than 15
- 16-25 years old
- 26-35 years old
- 36-55 years old
- 56 years and older

Obviously, SAP HANA view allows producing these kinds of calculations on the fly using calculated column formulas. In SAP BW it is a bit more complex, and previously using virtual characteristics was the only option to deliver on such requirements. Now we can add calculated columns (e.g. Age Range) in the HANA

### Derive Company Type based on If-Then Logic

Below you can find a few screenshots for an example of a HANA Calculation View with a newly calculated column (Company Type) that is derived “on the fly” at the query runtime using If-Then logic. These types of calculations are not easy to deliver within BW queries, therefore mixed scenarios with Calculation views can be useful.

The screenshot displays a data modeling tool interface with three main panels:

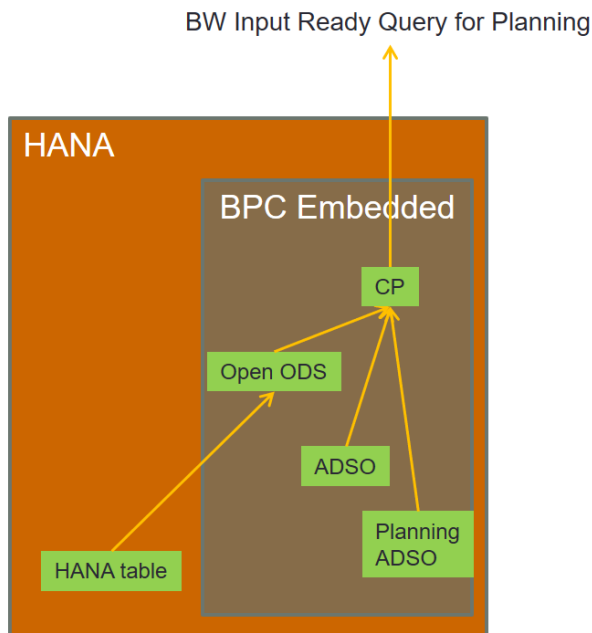
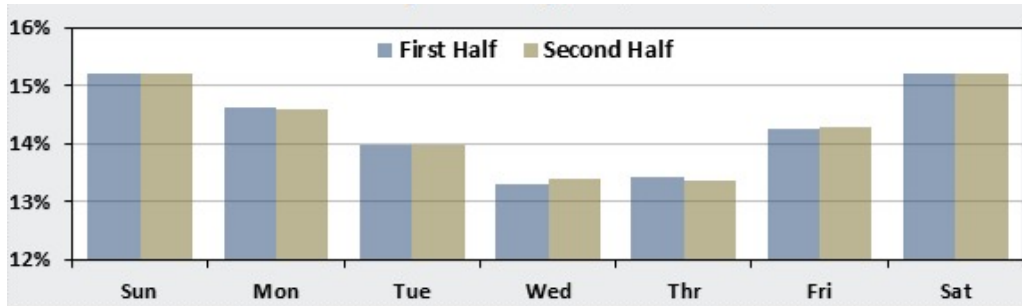
- Scenario:** Shows a flow diagram with three components: Semantics, Projection, and TBL\_CA\_CALC.
- Details:** Lists the columns for the table 'TBL\_CA\_CALC\_ALL\_TICKETS\_BASIC'. The columns include SystemType, PaymentKey, Hour, PlantKey, TicketNumber, SellerKey, TicketDate, TicketTime, TicketLineNumber, MaterialKey, ModifTypeKey, DiscountTypeKey, BGrossSaleVAT, GrossSalesVAT, NetSalesVAT, QtySold, PPlusCardNumber, NotPPlusCardNumber, PointsCount, SellerText, and TicketNumberInit.
- Output:** Lists the columns for the table. The 'CompanyType' column is highlighted with a red circle. The list includes PlantKey, TicketNumber, SystemType, PaymentKey, Hour, SellerKey, TicketDate, TicketTime, TicketLineNumber, MaterialKey, ModifTypeKey, DiscountTypeKey, BGrossSaleVAT, GrossSalesVAT, NetSalesVAT, QtySold, PPlusCardNumber, NotPPlusCardNumber, and PointsCount. Under 'Calculated Columns', 'CompanyType' is listed and circled in red.

The screenshot shows the 'Edit Calculated Column' dialog box with the following details:

- Name:** CompanyType
- Label:** Company Type
- Data Type:** VARCHAR, Length: 10, Scale:
- Column Type:** Attribute
- Client Aggregation:** None
- Hidden  Enable client side aggregation
- Expression Editor:** Contains the expression: `if(in("SystemType", 'STORE', 'ECOM'), 'Type A', if(in("SystemType", 'STOREB'), 'Type B', 'Type C'))`
- Language:** Column Engine
- Elements:** Filter pattern, Columns, Calculated Columns
- Operators:** +, -, \*, \*\*, /, %, (, )
- Functions:** Filter pattern, Conversion Functions, String Functions
- Buttons:** OK, Cancel

## Planning by Week Days

Let us imagine we have a dataset with daily granularity (by calendar day) and we want to do planning on top of it using calendar weeks and week days (Monday, Tuesday, etc.) This way we can easily apply week day dynamics in planning, for example, using statistics by shop/week day and applying proper ratios to the planned sales figures. Accumulated sales statistics may look as follows:



We want a relatively simple solution to derive week numbers and week days based on dates. Within SAP ABAP there is a function module that can be used (FM 'DATE\_COMPUTE\_DAY').

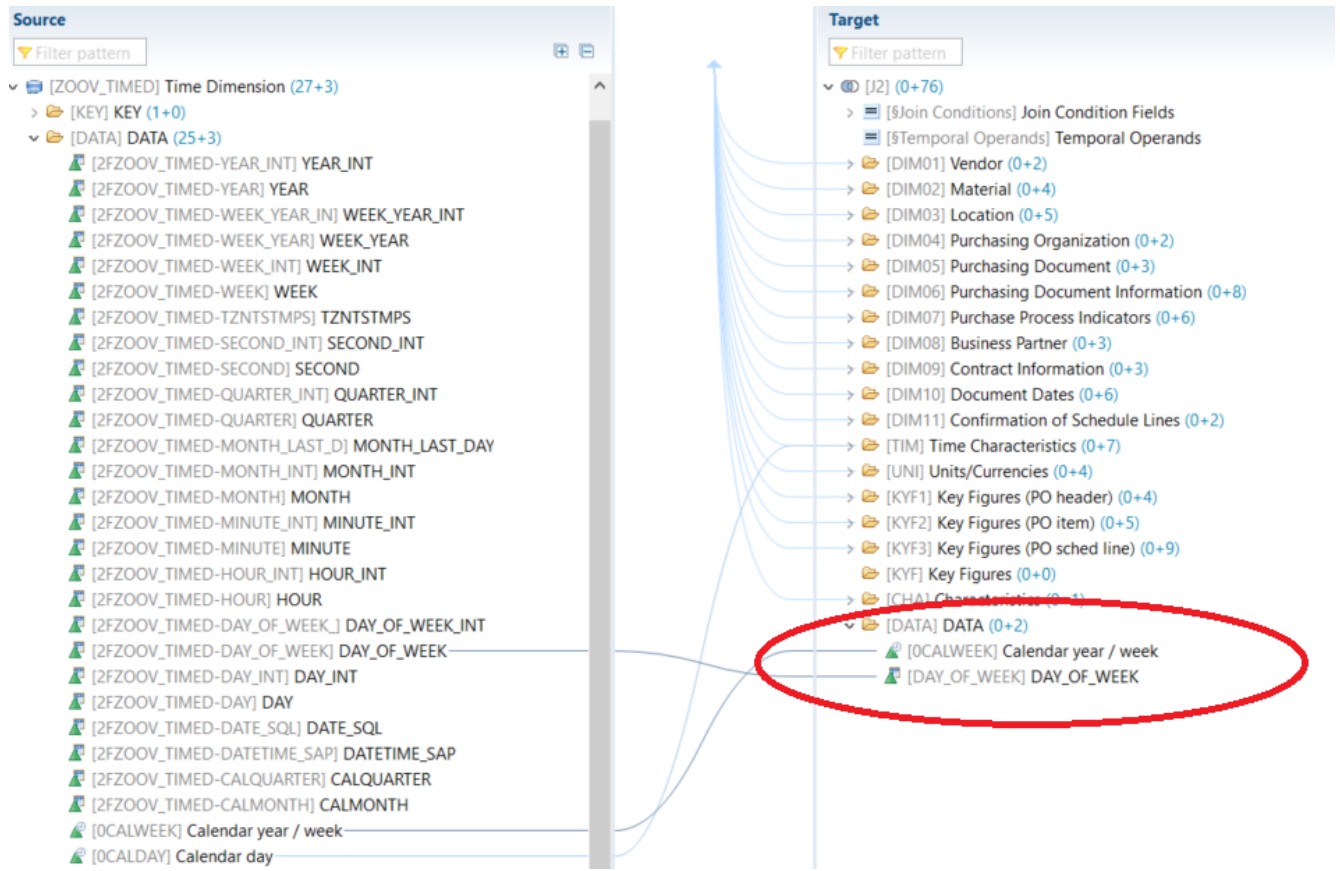
However, we do not want to use ABAP Function Modules, and instead, we want to implement a virtual solution without additional persistency objects.

In this case we will leverage a standard Time dimension from SAP HANA (M\_TIME\_DIMENSION). When using it for the first time we have to generate time data as described in the SAP Help below:

[https://help.sap.com/saphelp\\_hba/helpdata/en/oc/1bd75264e19d36e10000000a44538d/frameset.htm](https://help.sap.com/saphelp_hba/helpdata/en/oc/1bd75264e19d36e10000000a44538d/frameset.htm)

The generation must be done once. After that Time dimension can be consumed in any HANA based application, including HANA native, BW or BPC.

In the screenshot below you will see two extra fields added from the Time Dimension in the Composite Provider. These fields allow reporting and planning by week and week days on top of datasets where only calendar day is present.



## Addressing Data Protection (GDPR) Requirements

Data Masking feature in SAP HANA 2 helps addressing data protection requirements via Mixed scenarios. Data masking can be consumed both via HANA native models and in BW. This technical capability is especially relevant with the adoption of GDPR (The General Data Protection Regulation is a regulation in EU law on data protection and privacy for all individuals within the European Union).

Please see a separate post (white paper) on the Data Protection topic: <https://www.biportal.org/single-post/2018/04/11/Addressing-Data-Protection-GDPR-Requirements-in-SAP-HANA-and-BW4HANA>

## Final Words

There is a bunch of business applications and use cases where applying SAP HANA mixed scenarios can be crucial. Consuming HANA Calculation Views in BW and BPC brings benefits when designing analytical applications. HANA native modelling adds value on BW side when creating complex calculated columns, applying data masking, joining with HANA native tables. It is important to understand capabilities, benefits and value each product brings, and perform modelling in either SAP HANA native, BW or BPC whenever it makes sense and brings business value.



*Sergei Peleshuk has over 15 years of experience implementing BI technologies for global clients in retail, distribution, fast-moving consumer goods (FMCG), insurance, and energy industries. He has helped clients to design robust BI reporting and planning capabilities, leading them through all project phases: from analysis of requirements to building BI roadmaps, technical architecture, and efficient BI teams. Sergei is an expert in SAP Business Warehouse (SAP BW), SAP HANA, BPC, BusinessObjects, SAP Analytics Cloud, and SAP Lumira. You may contact Sergei at [peleshuk@biportal.org](mailto:peleshuk@biportal.org)*