



DMM208 – New and Best Practices for Data Modeling with SAP HANA



Speakers

Las Vegas, Oct 19 - 23

- Werner Steyn

Barcelona, Nov 10 - 12

- Christoph Morgen



Agenda

- Data Modeling in SAP HANA
- New Approaches
- Best Practices

Disclaimer

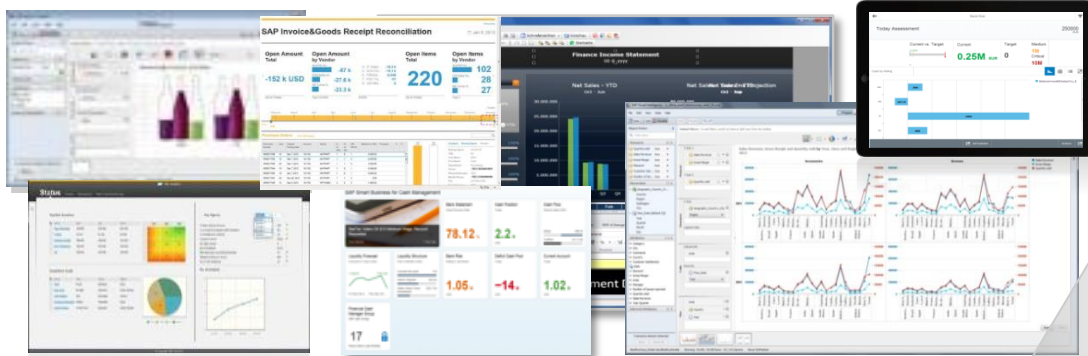
This presentation outlines our general product direction and should not be relied on in making a purchase decision. This presentation is not subject to your license agreement or any other agreement with SAP. SAP has no obligation to pursue any course of business outlined in this presentation or to develop or release any functionality mentioned in this presentation. This presentation and SAP's strategy and possible future developments are subject to change and may be changed by SAP at any time for any reason without notice. This document is provided without a warranty of any kind, either express or implied, including but not limited to, the implied warranties of merchantability, fitness for a particular purpose, or non-infringement. SAP assumes no responsibility for errors or omissions in this document, except if such damages were caused by SAP intentionally or grossly negligent.

Data Modeling with SAP HANA

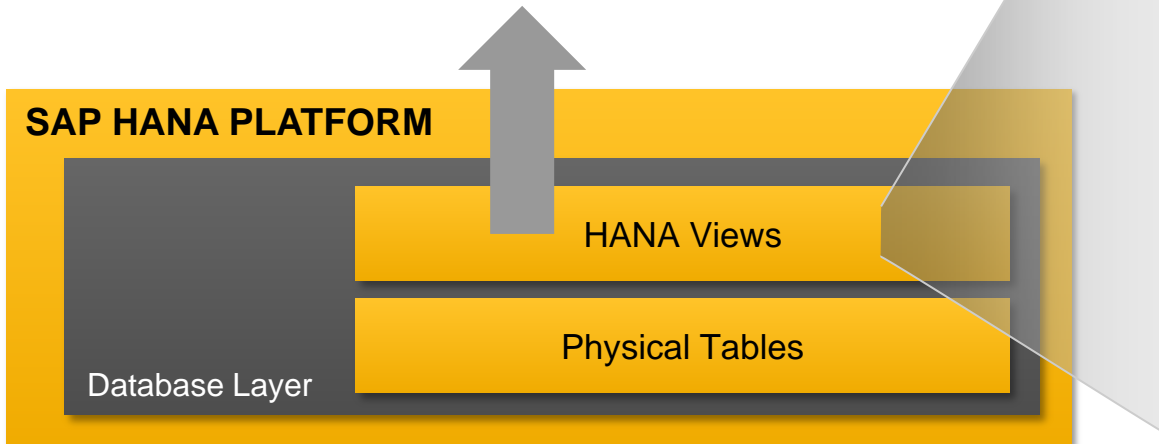
Overview SAP HANA Information Views

SAP HANA View Modeling – Overview

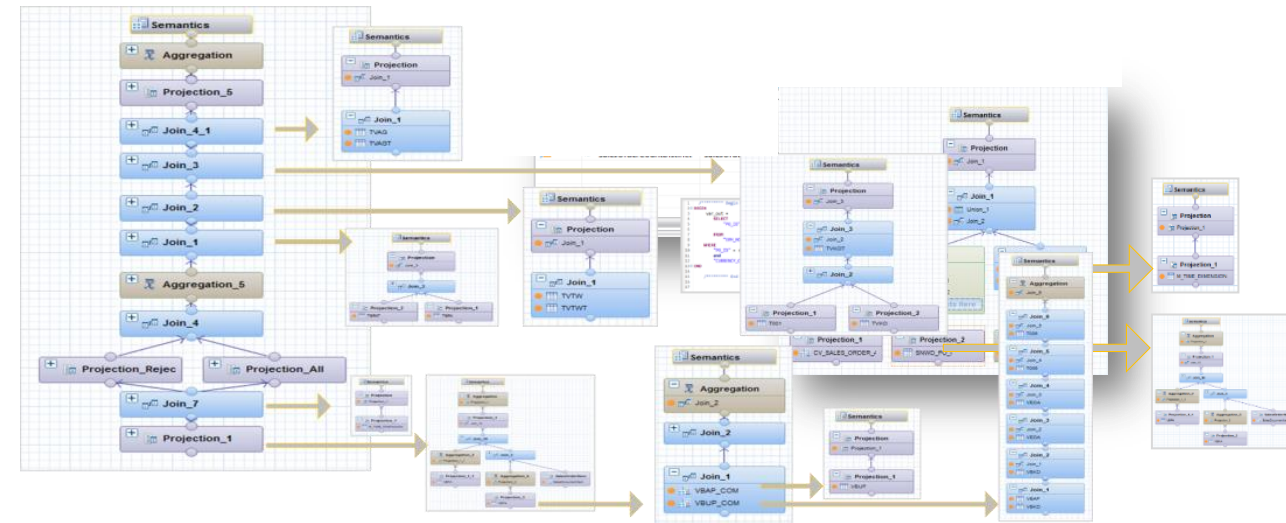
Virtual Data Modeling as a key SAP HANA concept



Operational Reporting | Applications | Analytics



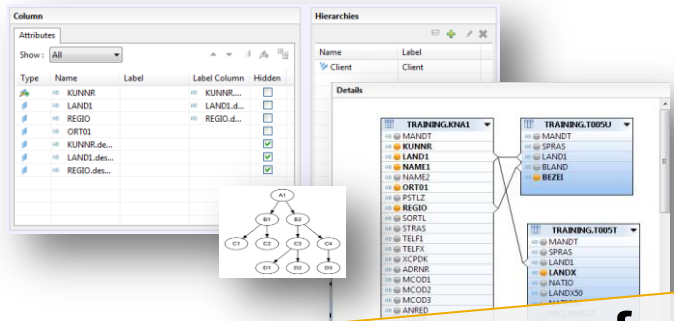
Virtual Data Flow Models



- No persistent Aggregations | single atomic copy of data
- In-Memory Engines | Performance
- Multidimensional Reporting Models
- Enterprise Applications Virtual Data Models

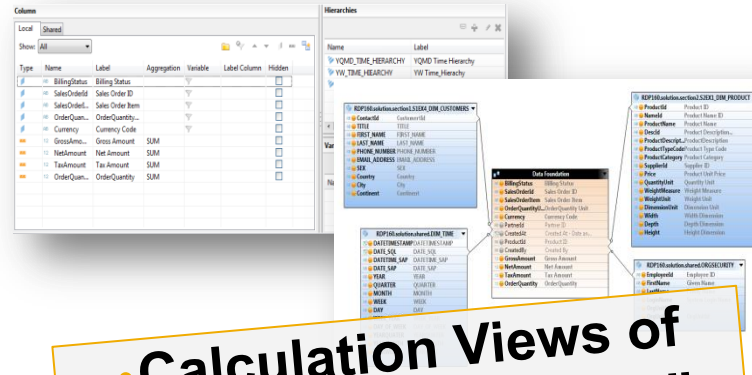
SAP HANA View Modeling – Overview

Flavors of SAP HANA View Modeling Approaches



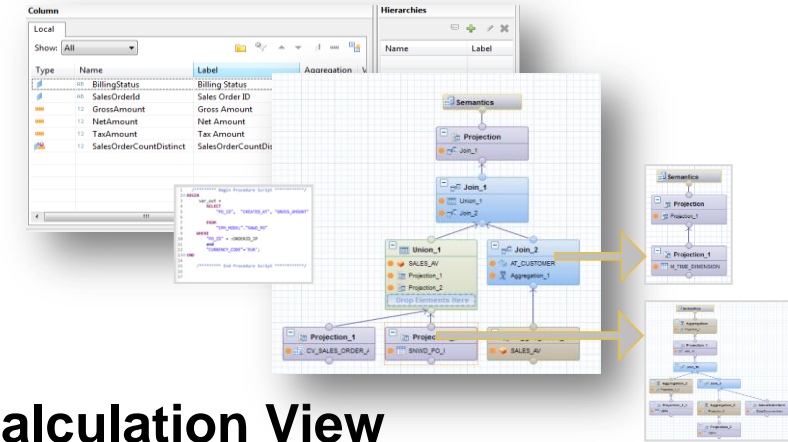
• Calculation Views of type “Dimension”

- Compose a dimensional view with a series of attributes derived from a collection of tables e.g. Master Data Views
- Highly re-used and shared in Analytic- and Calculation Views
- Used to build Hierarchies
- Hierarchies are key elements in use with Analytic View for multi-dimensional reporting



• Calculation Views of type “CUBE/StarJoin”

- Combines Fact-Tables with Attribute-Views to Star-Schema- or OLAP Cube-like objects for multidimensional reporting.
- Stores no aggregates and mass-aggregates on the fly
- Hierarchies are key for multi-dimensional access (navigation, filtering, slicing and aggregation)



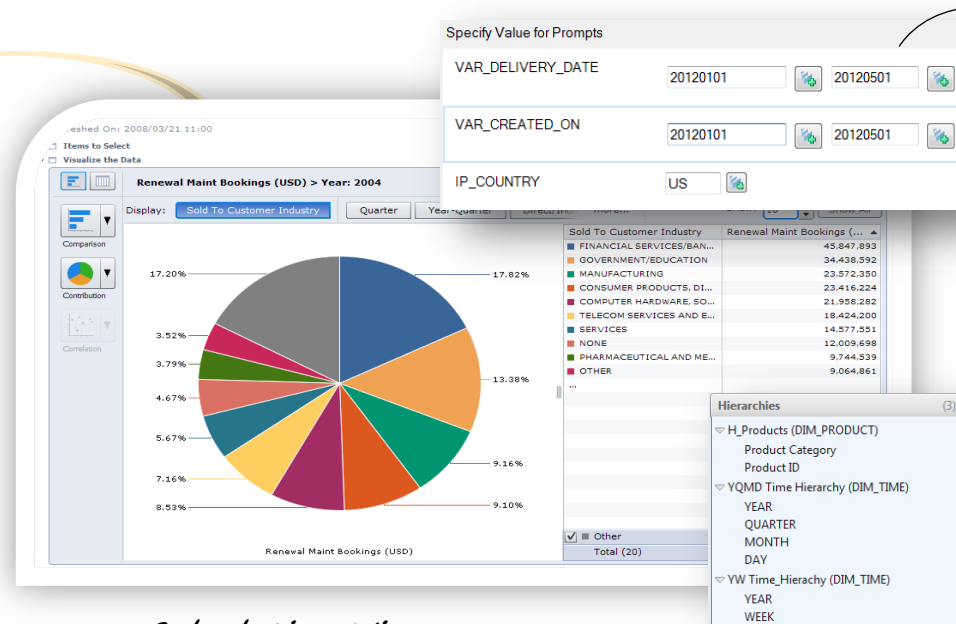
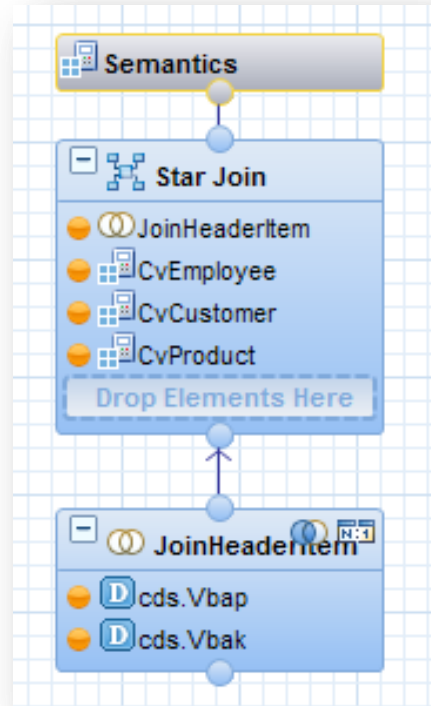
Calculation View

- Great flexibility for advanced use
- Approach to model custom scenarios like
 - Combined use of Multiple-Fact Tables/Analytics Views
 - Build Models on Normalized Data
 - Re-Use and stack views
 - Make use of custom scripted views

SAP HANA View Modeling – Overview

Virtual Data Models for Multidimensional Scenarios

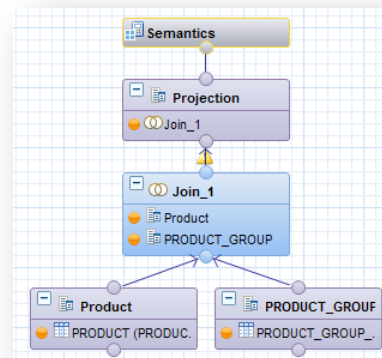
Calculation Views are usually build upon other Calculation Views and Column Tables



Reporting Tools can usually directly consume Calculation Views.

Multidimensional Tools support Hierarchies for Navigation, Filtering and Aggregation and HANA Prompts (Variables & Input Parameters) for efficient Pre-Filtering of Data.

Calculation Views



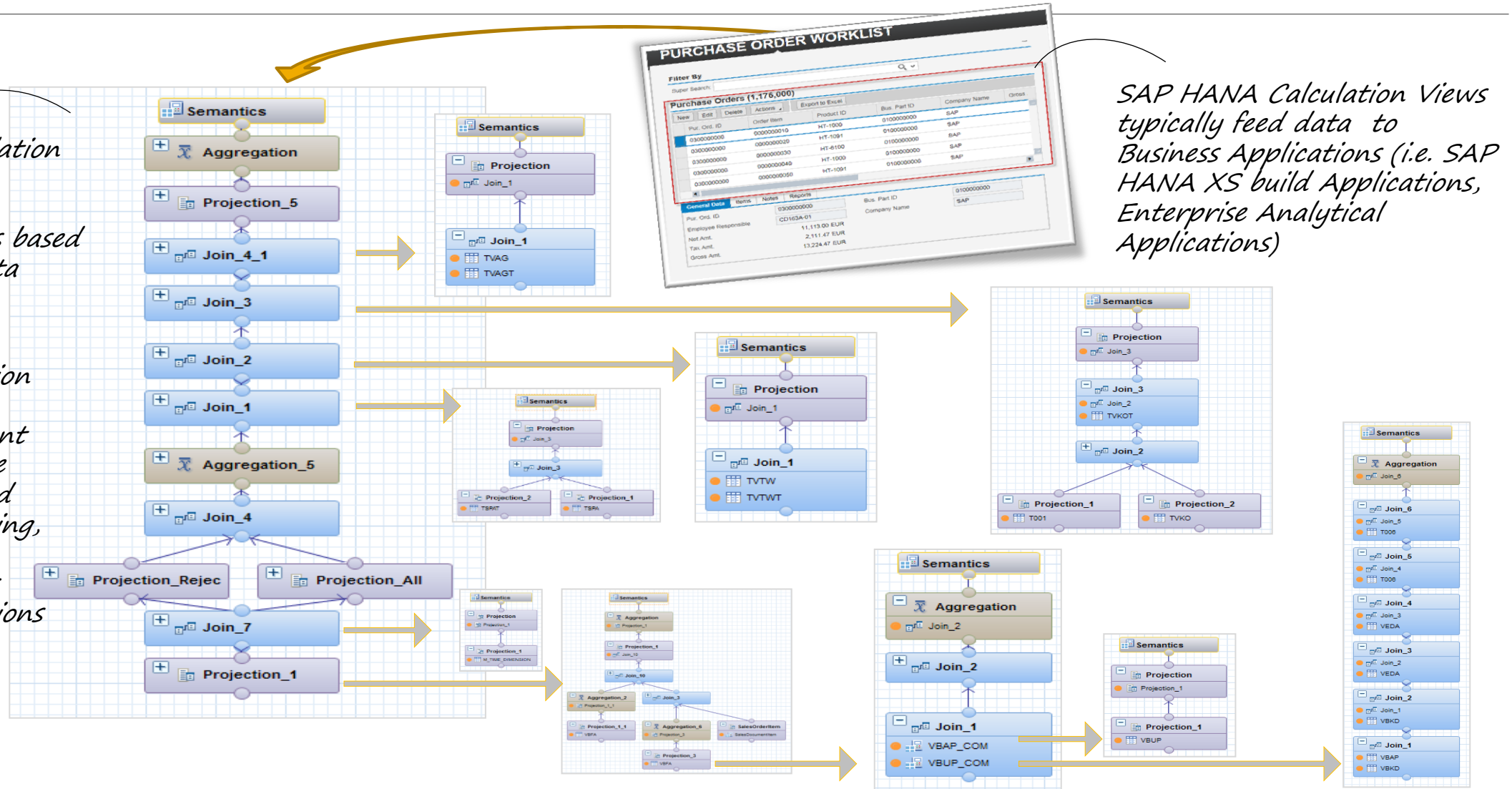
Column table

Table Name:			
MARA			
Columns	Indexes	Further Properties	Runtime Information
Name	SQL Data Type	Dim	
1 MANDT	NVARCHAR	3	
2 MATNR	NVARCHAR	18	
3 ERSDA	NVARCHAR	8	
4 ERNAM	NVARCHAR	12	
5 LAEDA	NVARCHAR	8	
6 AENAM	NVARCHAR	12	
7 VPSTA	NVARCHAR	15	
8 PSTAT	NVARCHAR	15	
9 LVORM	NVARCHAR	1	
10 MTART	NVARCHAR	4	

Virtual Data Models for Normalized Data Model Scenarios

SAP HANA Calculation Views provide the means to model sophisticated views based on normalized data structures.

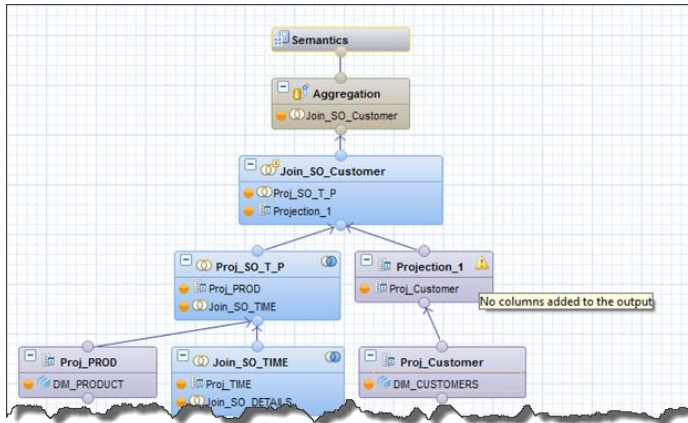
Complex Calculation Views demand a more explicit intent and control of the modeled set-based data flow, i.e. slicing, aggregation and filtering of sets as input to joins, unions etc.



SAP HANA Modeling – Graphical Editors

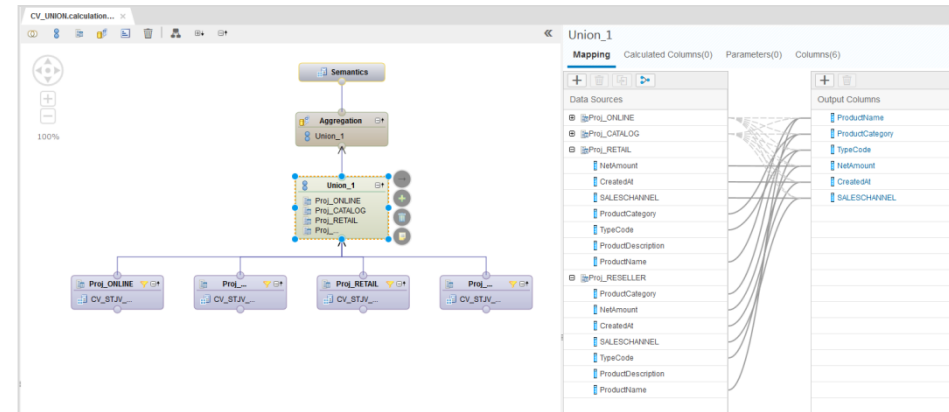
Eclipse-based & Web-based Development Workbench

SAP HANA Studio



- **Modeler Eclipse-perspective**
Systems view: Supports basic Modeling artifacts.
(no support for functions, roles, CDS objects, repository)
- **Development Eclipse-perspective**
Project/Navigator/Repository view: Support all design-time artifacts and repository functionality.

SAP HANA Web-based Development Workbench



Browser-based graphical View Editor

Browser → <http://hana-server:port/sap/hana/ide/editor/>

- No Analytic- and Attribute-view Modeling support
- Supports Calculation Models (table functions, roles, procedures, script based calc-models, analytical privileges etc.)
- SQL Editor support, Catalog, Security, Admin, etc.

Note: Web-based editors are future/mid-term innovation platform for SAP HANA development tools, however currently not yet feature complete (SAP HANA SPS10).

New Approaches

SAP HANA Information Models

SAP HANA View Modeling – New Approaches

New Approaches Overview

Key new approaches

- Calculation View Modeling
 - **Calculation views-Star-join** instead of Analytical/Attribute
- SQL-Script Integration
 - **Table functions** as data sources
(instead of script-based Calculation Views)
- Analytic Privileges
 - Support for SQL based analytic privileges
- Hierarchies
 - SQL enablement & Time Dependent hierarchies
- Data Sources
 - CDS entities, tenant container database tables
- Web-based Editor

Specific Enhancements

- Column lineage analysis
- **Comment** & View **Deprecation**
- **Copy & Replace** nodes!
- Multilanguage-label support in Views
- Performance Analysis / **Debugger** enhancements
- Support Labels/Descriptions in value help dialogs

General Enhancements

- Implicit **SQL Execution** - Model unfolding

SAP HANA View Modeling – New Approaches

Modeling multidimensional Scenarios in Calculation Views

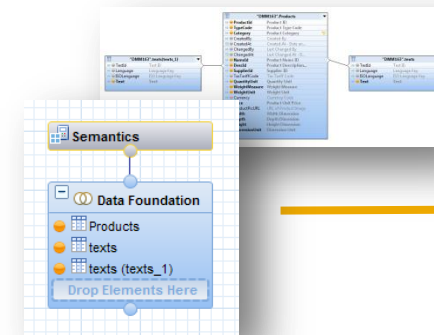
Multidimensional scenarios in Calculation Views

- Calculation Views of type Dimension
 - Composing Dimension-Attributes and -Hierarchies
- Calculation Views of type CUBE with StarJoin
 - Star-joins fact data with dimension-type calculation views like Analytic Views (with same performance guidance's)
 - Implicit Calculation View optimization will evaluate to delegate processing to same execution behavior as Analytic Views
- Note: Table functions can be used as input sources for either Dimensions or Facts

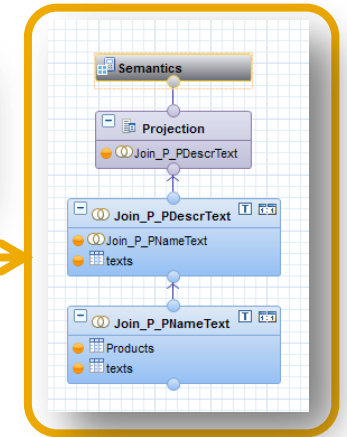
Usage Scenario

- Implement new multidimensional scenarios / new projects
- Seek to refactor Attribute-/Analytic View scenarios

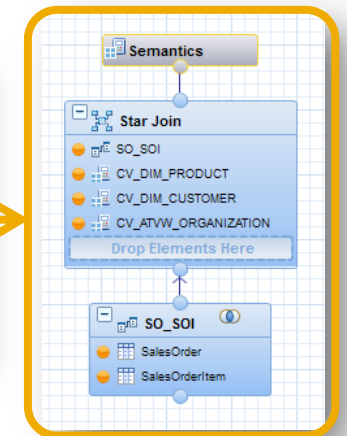
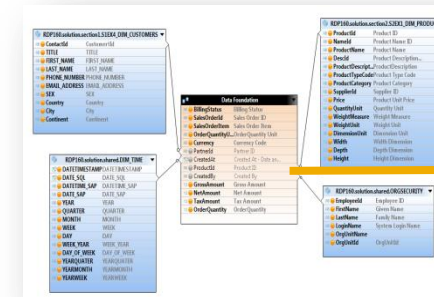
Attribute View



Calculation Views



Analytic View



Note: As of SAP HANA SPS10 Calculation Views are not yet functional feature equivalent to Attribute- and Analytic Views (i.e. Temporal Join & Fuzzy search).

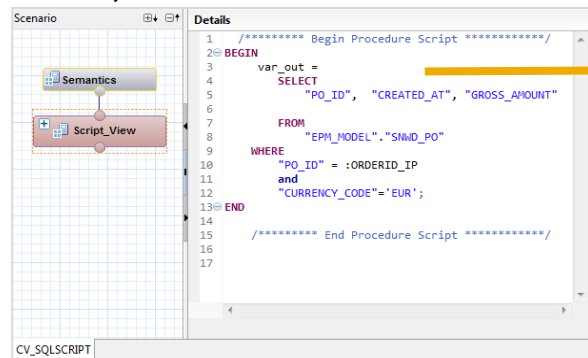
SAP HANA View Modeling – New Approaches

Scripting custom Data Flow Scenarios

Enabling SQL Script with Calculation views

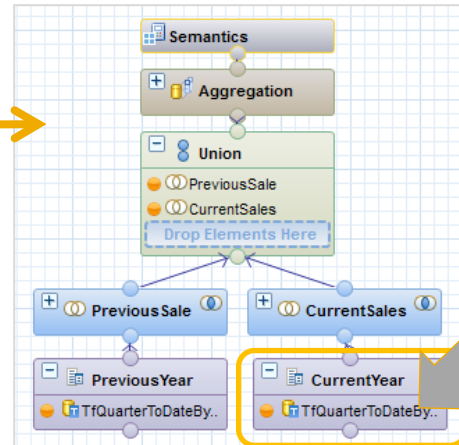
- Going forward use SQL Script **table functions** instead of script-based Calculation views
 - Side-effect free read-only functions using standard SQL
 - Consume data from tables, models, functions, predictive algorithms, etc.
 - Script based-calculation models can be re-factored into table functions

Historical
Script-based Calculation View



New!

Graphical Calculation View
referencing Table function



```
function QuarterToDateByYear(
    in ip_period varchar(1), in ip_date varchar(10)
)
returns series_data_t
language sqlscript sql security invoker as
begin
```

SQL

```
sd = select source_period_start, source_period_end
from series_disaggregate_date ('interval 3 month',
    'interval 1 day',:v_year,:v_next_year);

if :ip_period = 'c' then
    return select
        bb.generated_period_start as generatedperiodstart,
        aa.generated_period_start as factdate,
        bb.generated_period_start as datesql,
        'qtd' as "timekpi", 'cy' as cpy_flag
    from ...
```

SAP HANA View Modeling – New Approaches

Scripting custom Data Flow Scenarios

SQL Script approaches

- Standard SQL versus Calculation Engine (CE Functions) Plan Operators
 - Recommendation! When creating functions or procedures with SAP HANA SP09+ use standard SQL syntax instead of CE Functions. The SQL optimizer is able to leverage alternative execution engines and are not restricted only to the Calculation Engine

	<i>Recommended approach</i>	
	SQL	CE function
SELECT on column table	<code>out = SELECT A, B, C from "COLUMN_TABLE"</code>	<code>out = CE_COLUMN_TABLE("COLUMN_TABLE", [A, B, C])</code>
GROUP BY	<code>out = SELECT A, B, C, SUM(D) FROM "COLUMN_TABLE" GROUP BY A, B, C</code>	<code>col_tab= CE_COLUMN_TABLE("COLUMN_TABLE"); out = CE_AGGREGATION((col_tab, SUM(D), [A, B, C]);</code>

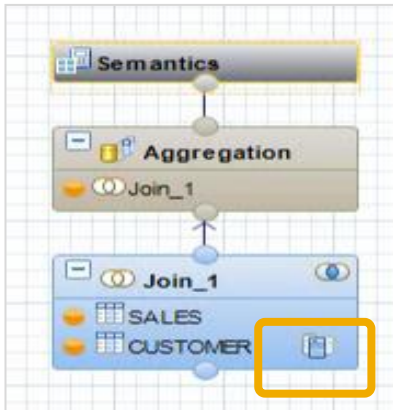
SAP HANA View Modeling – New Approaches

New Design-time Data Sources

New data sources for Calculation Views

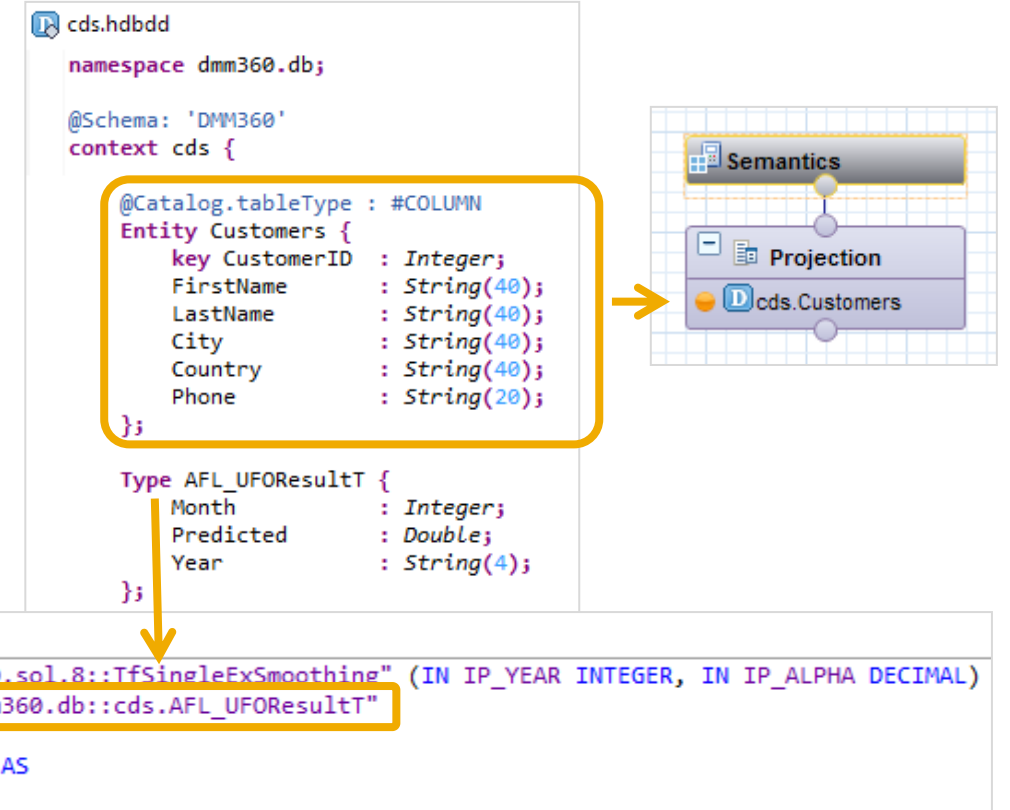
- **Multitenant database container tables**

- While modeling graphical calculation views, you can add data source from any of the isolated databases.



- **CDS (Entities and Types)**

- Graceful data structure lifecycle



SAP HANA View Modeling – New Approaches

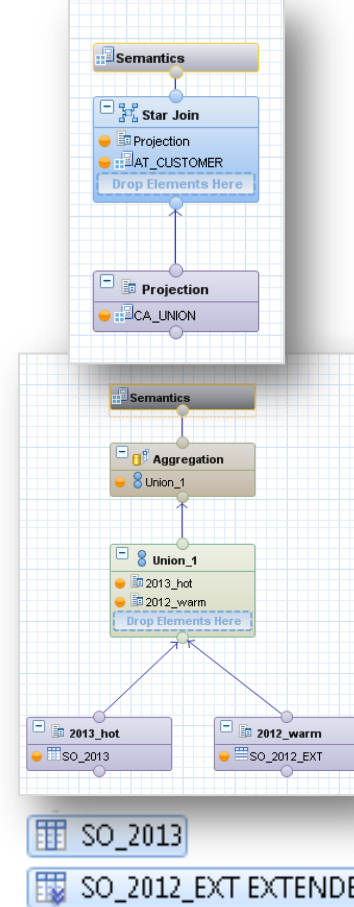
Modeling with Virtual and Extended Tables

Consuming non-In Memory data in Calculation views

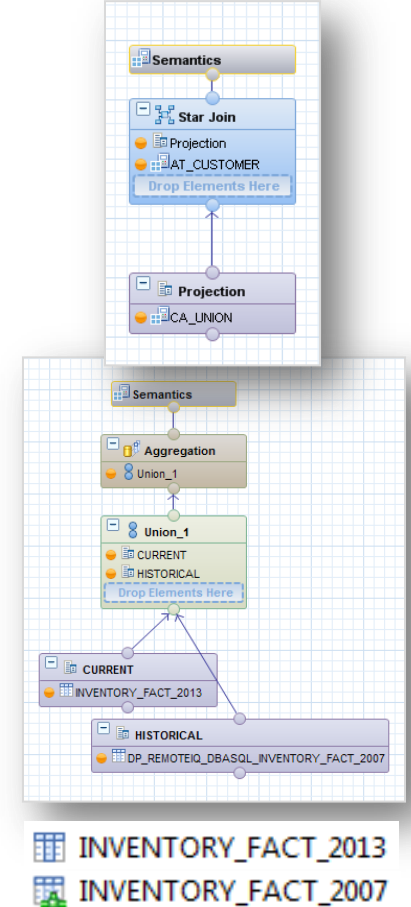
- Dynamic Tiering
 - Provides a disk-based columnar SAP HANA database extension using SAP HANA external tables (extended tables) *
- Smart Data Access
 - Provides a virtual access layer to outside SAP HANA data (e.g. other databases, Hadoop systems, etc.) using so-called virtual tables
- Model and approach these scenarios carefully and monitoring query performance
 - Remote data imposes a natural break in the execution
 - Ensure that filters and aggregation is pushed to remote sources
 - Use and observer join settings and optimizations
 - Implement input source pruning (explicit or implicit) techniques

*Supported external systems and restrictions are documented in SAP Note [1868209](#)

*Leveraging “warm data”
from SAP HANA
Dynamic Tiering
external tables*



*Leveraging “cold data”
via SAP HANA
Smart Data Access
virtual tables*



SAP HANA View Modeling – New Approaches

Calculation View Enhancements – Hierarchies

Hierarchy SQL Integration

- Hierarchy-based SQL processing capabilities enabled via SAP HANA View based hierarchies
 - SQL processing includes hierarchy-based filtering, aggregations as well as analytic privileges*.
- Usage scenario and enablement
 - Shared hierarchies from Dimension-Type and StarJoin-Type Calculation Views. Compound parent-child hierarchies are not allowed.
 - SQL use of hierarchies needs to be activated in the view properties (for all shared hierarchies). Once activated custom node-column name and hierarchy-expression parameter* can be given custom name

*Note: Use of hierarchy expression will be fully enabled, once hierarchy-expressions are documented in the SAP HANA SQL documentation and thus generally supported.

The screenshot shows the 'Hierarchy' dialog box in SAP HANA View Modeler. The 'General' tab is selected, displaying 'Shared Hierarchy Definition' for 'OrgHierarchy'. The 'Hierarchy Type' is set to 'Parent-Child Hierarchy'. The 'SQL Access' tab is highlighted with a yellow box, showing 'Enable SQL access' checked, 'Hierarchy Columns' set to 'SalesRepHierarchyNode', and 'Hierarchy Expression Parameter' set to 'SalesRepHierarchyExpression'. A yellow box also highlights the 'Enable Hierarchies for SQL access' checkbox in the top right corner.

SAP HANA View Modeling – New Approaches

Calculation View Enhancements – Hierarchies

Hierarchy SQL Integration

- **Hierarchy-aggregation**

– The **node column** can be used for **group by**:

SQL SQL Result

```
select "SalesRepHierarchyNode", sum("Revenue") as "Revenue",  
      sum("Cost") as "Cost", sum("Margin") as "Margin"  
FROM "_SYS_BIC"."mini/CvSalesCubeHier" group by "SalesRepHierarchyNode"
```

SalesRepHierarchyNode	Revenue	Cost	Margin
ASTERIX	12.360	3.057	75,26
JOHN	10.590	1.583	85,05
LISA	8.020	1.269	84,17
MAJESTIX	20.370	3.463	82,99
MIKE	2.570	314	87,78
OBELIX	7.310	304	95,84
THE_BIG_BOSS	30.960	5.046	83,7

Note, the Revenue and Cost column contain the hierarchical aggregated values, i.e. the Revenue for MAJESTIX is the sum of the revenue of all its child nodes.

The calculated column Margin (percentage) is calculated correctly after the aggregation on each hierarchy level.

Hierarchy Prompts

THE_BIG_BOSS
MAJESTIX
ASTERIX
OBELIX
JOHN
LISA

Name: Margin
Label: Margin in Percentage

Expression Semantics

Expression Editor

✓ Validate Syntax Language: Column Engine

$(\text{"Revenue"} - \text{"Cost"}) * 100 / \text{"Revenue"}$

- **Filtering on a hierarchy node value**

– The **node column** can be used for **filtering**

SQL SQL Result

```
select "SalesRep", sum("Revenue") as "Revenue",  
       sum("Cost") as "Cost", sum("Margin") as "Margin"  
FROM "_SYS_BIC"."mini/CvSalesCubeHier" where "SalesRepHierarchyNode" = 'MAJESTIX'  
group by "SalesRep"
```

SalesRep	Revenue	Cost	Margin	
ASTERIX	12.360	3.057	75,26	
MAJESTIX	700	102	85,42	
OBELIX	7.310	304	95,84	

Note, that the filter on the node column constrains to the hierarchy subtree. All the descendants of the node are selected

Analytic Privileges Editor – SQL-based Analytic Privileges

Create SQL-based Analytic Privileges

- Start with general Attribute based Analytic Privilege, then switch to SQL-based

Privilege Validity
Defines the Privilege Validity

Operator	Inclusion	From	To
Equal	<input checked="" type="checkbox"/>	2015-03-24	

Associated Attributes Restrictions
Select attributes to assign analytic privileges.

Attributes	Origin	Description	Count
STORE	SPATIAL_DEMO (demo.sprint...		1

Assign the restrictions for STORE

Type	Operator	Value
Fixed	Equal	EBAY

☐ Attributes ☒ SQL Editor ☐ Dynamic
Define restrictions based on attributes

☐ Attributes ☒ SQL Editor ☐ Dynamic
Define restrictions based on SQL expression

```
(( "STORE" = 'EBAY' ) OR ( "STORE" = 'EDEKA' )) AND (( CURRENT_DATE BETWEEN 2015-03-24 00:00:00.000 AND 2015-03-24 23:59:59.999 ))
```

- Use SQL Hierarchies within SQL Analytical Privileges

☐ Attributes ☒ SQL Editor ☐ Dynamic
Define restrictions based on SQL expression

```
SalesHierarchyNode = MAJESTIX
```

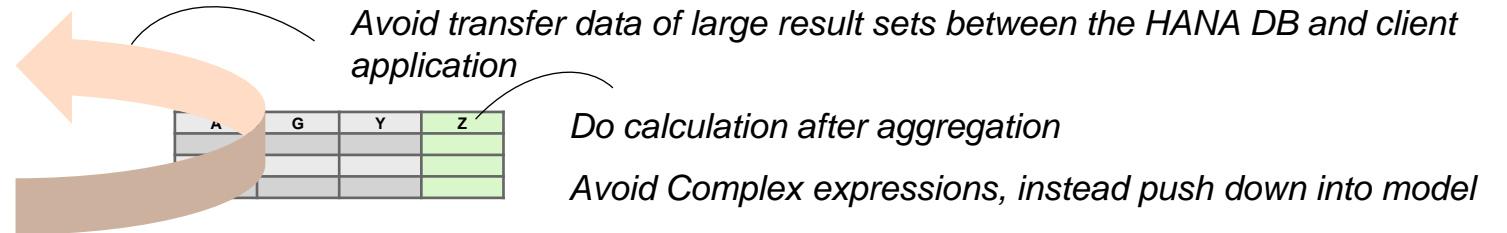
Best Practices

SAP HANA Information Models

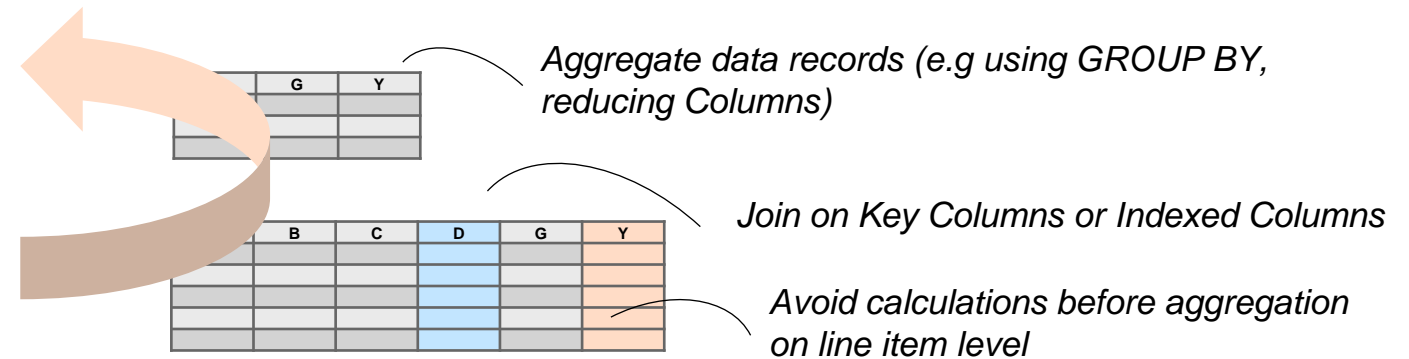
SAP HANA View Modeling – Best Practices

General Performance Guidelines

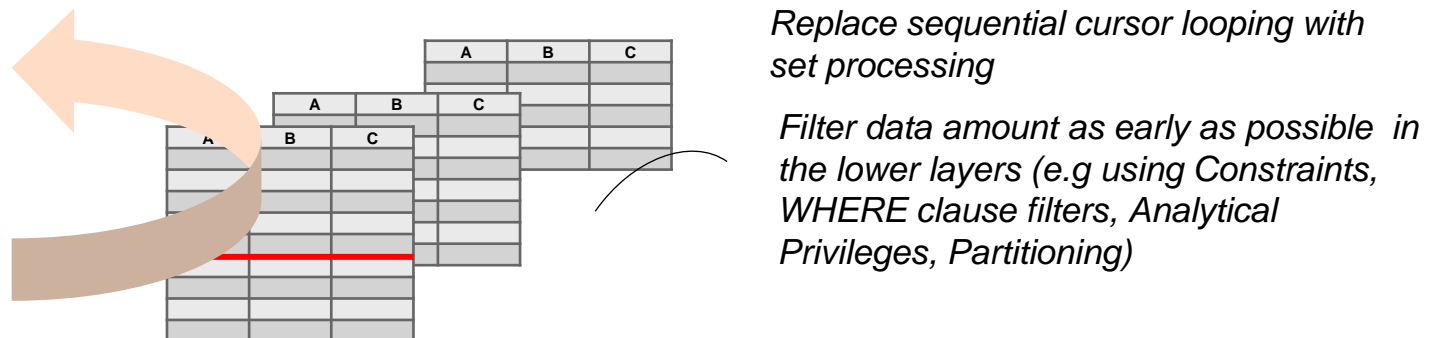
Client



Calculation Views
Procedures / Functions



Column
Store

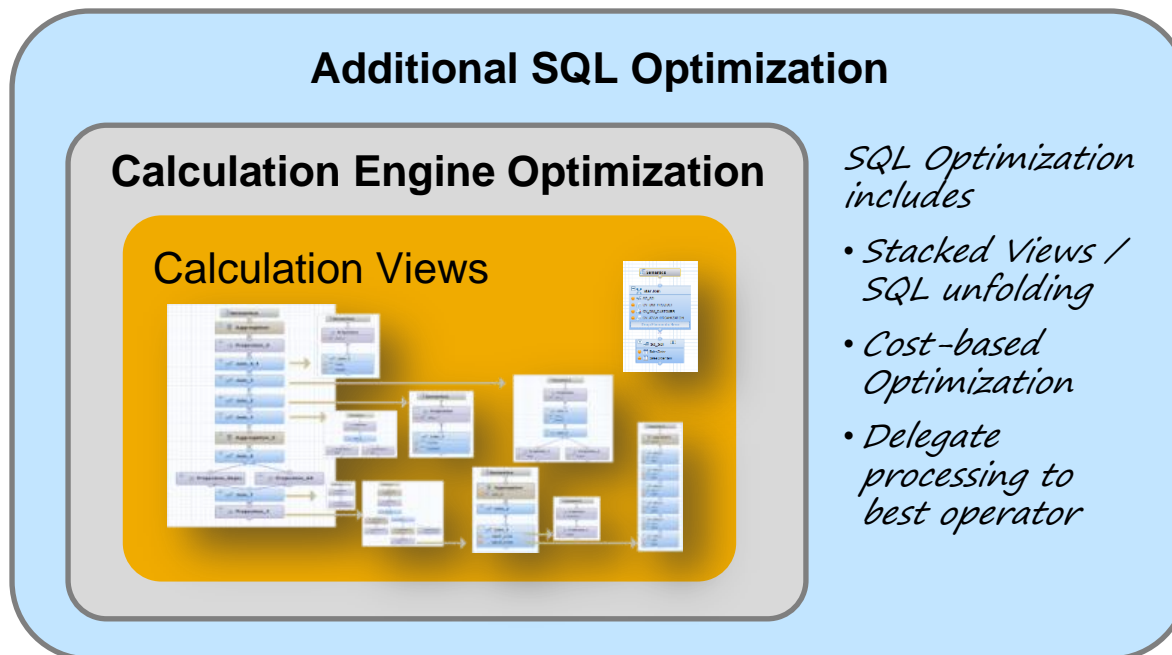


SAP HANA View Modeling – Best Practices

Optimized Calculation View Execution

Optimized execution for SQL-queries against Calculation Views

- The **initial Calculation Engine optimization** generates a single SQL statement across a stacked model, which is then passed to the SQL optimizer
- The **SQL optimizer adds additional optimizations** and delegates operations to the best database execution operator, e.g. delegate star join-nodes to the OLAP processing engine where possible



- Since SAP HANA Rev 62 the SQL optimization can explicitly be determined setting the “Execute In:” property to “SQL Engine”
- Since SAP HANA SPS9 the optimized execution behavior implicitly evaluates to leverage additional SQL optimization
- SAP HANA SPS10 optimization enhancements include implicit optimized Star Join processing

* Implicit SQL optimization details are documented in SAP Note [2223597](#).

SAP HANA View Modeling – Best Practices

Performance Analysis & Query Insight Tools

Visualize Plan

Operator list

Operator	Input Rows
0 Result	12
0 Result	12
0 Result	0
10 Column Search	n/a
10 JEEvalPrecond	n/a
0 Dynamic Search	n/a
0 Search On Table	n/a
0 Basic Predicate	n/a
0 Basic Predicate	n/a
0 Search On Delta 1	n/a
0 Search On Delta 1	n/a
0 JEDistinctAttribute	n/a
0 JEAgregate	n/a
0 Result Assembly	n/a

Timeline – root cause analysis

Calculation View Debugger

Details

Node Query

```
SELECT "Year", "Month", "ProductID", "Name", "OrderID", "OrderDate", "Brand", "SalesRolling" FROM "dmm360.ws1.00/CvCumulativeSalesQuery/in_1"('PLACEHOLDER'=('$$language$$', 'E
```

Performance Analysis

Year	Month	ProductID	Name	OrderID
1	2013	1	Apples	1
2	2013	1	Apples	2
3	2013	1	Apples	3
4	2013	1	Apples	4
5	2013	1	Bana...	5
6	2013	2	Apples	6
7	2013	2	Apples	7
8	2013	2	Apples	8
9	2013	2	Apples	9
10	2013	2	Apples	10

Explain Plan

OPERATOR_NAME	EXECUTION...	TABLE_TYPE	TABLE_SIZE	OUTPUT...
ROW SEARCH	rdp361.solution.exercise5/SALESORDER_RAN...	ROW	?	45
WINDOW	WINDOW FUNC: DENSE_RANK() SORTING: ...	ROW	?	45
WINDOW	WINDOW FUNC: DENSE_RANK() SORTING: S...	ROW	?	45
COLUMN SE...	rdp361.solution.exercise5/SALESORDER_RAN...	COLUMN	?	45
AGGREGATI...	GROUPING: rdp361.solution.exercise5/SALES...	COLUMN	?	45
COLUMN ...	FILTER CONDITION: rdp361.solution.exercise...	COLUMN	OLAP VIEW	117,600

Essential Tips and Tricks - Filter & Aggregation Push Down

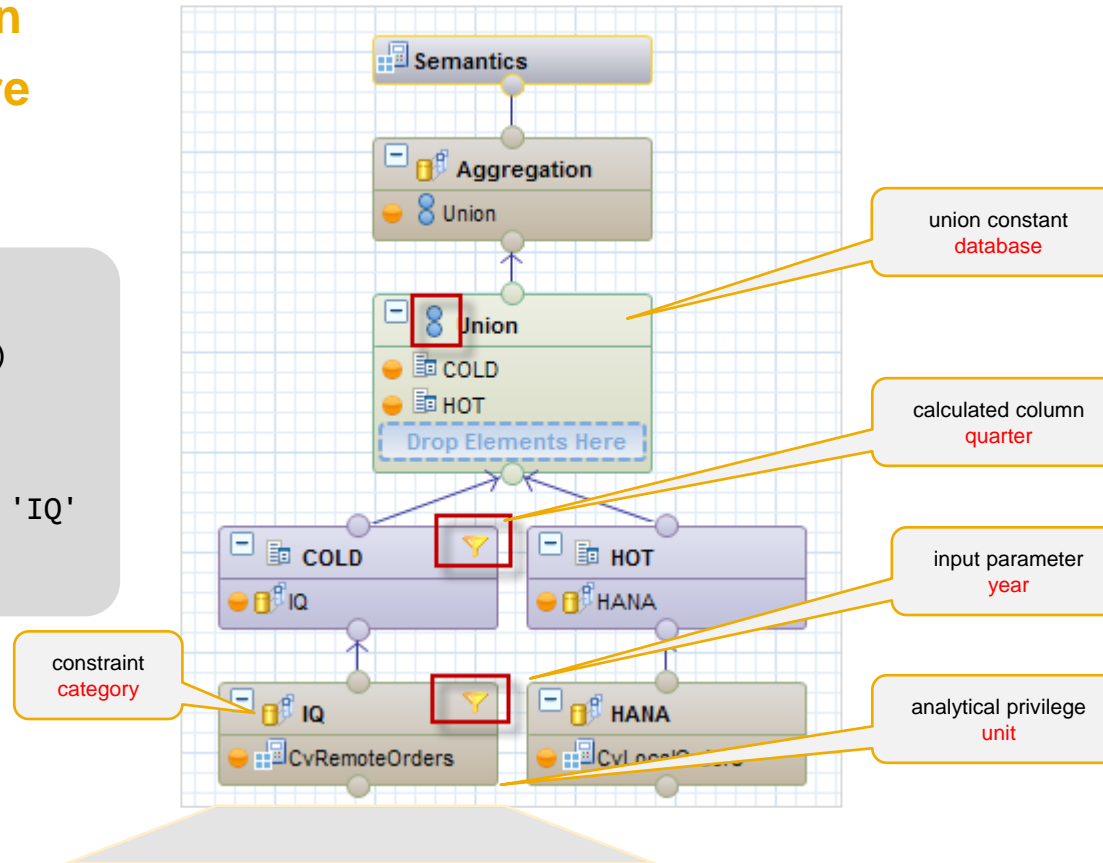
- Ensure that filters are pushed down
- Aggregate and ensure Group By are pushed down (i.e. SDA remote DB)

```
select
  division, sum(prevamount), sum(currentamount)
from CvOrdersQuery
  ('placeholder' = ('$$year$$', '2013'))
where
  country in ('USA', 'GERMANY') and database = 'IQ'
group by division
```

where clause
country

group by
division, quarter

```
group by division, quarter where
  country in ('USA','GERMANY') and year = '2013' and quarter = '4' and category= '100' and unit = 'XYZ'
```



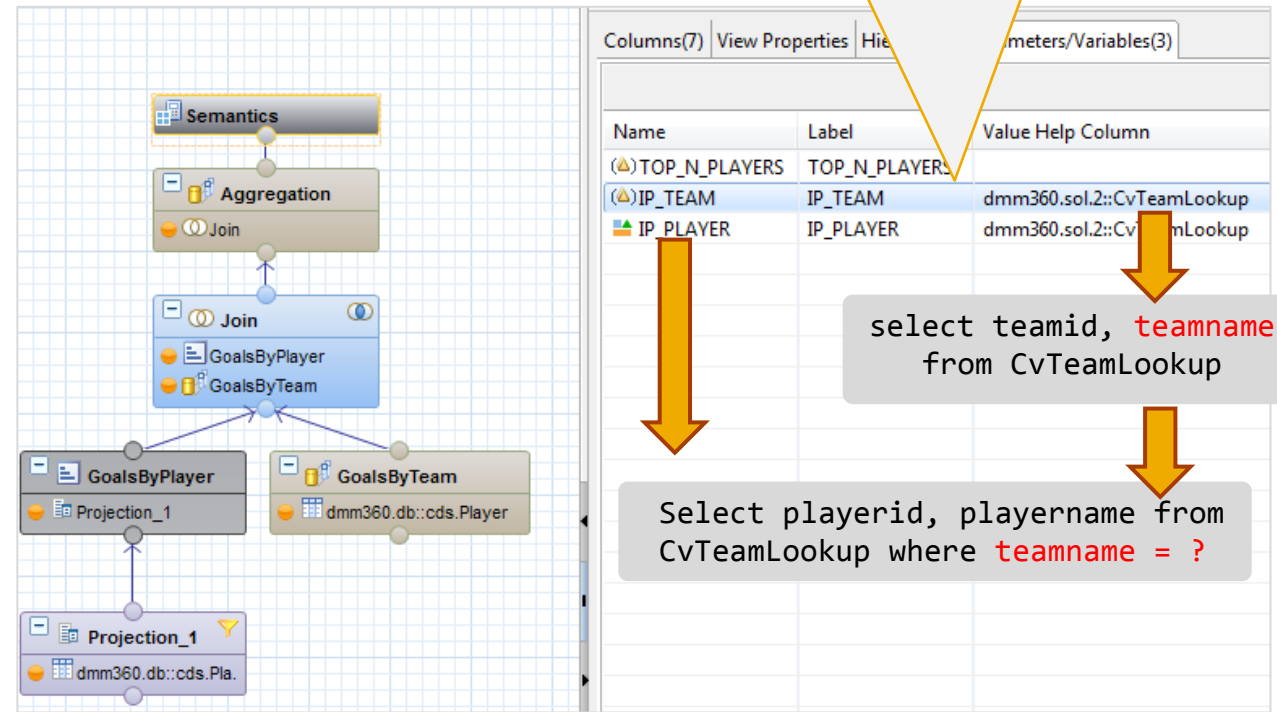
SAP HANA View Modeling – Best Practices

Essential Tips and Tricks - Value Help Lookup Models

Out-source Value Help Information to dedicated views

- **Benefit from faster value help dialogs**
- **Provide consistent LOVs across consuming views**
- **Dependencies between value help views**
- **Support Analytical Privileges**

```
select
  teamname, playername, rank,
  sum(playergoals), sum(teamgoals)
from
  CvSoccerPlayers(
    'placeholder'=('$$top_n_players$$','2'),
    'placeholder'=('$$ip_team$$','paderbom')
  )
group by teamname, playername, rank
order by teamname
```



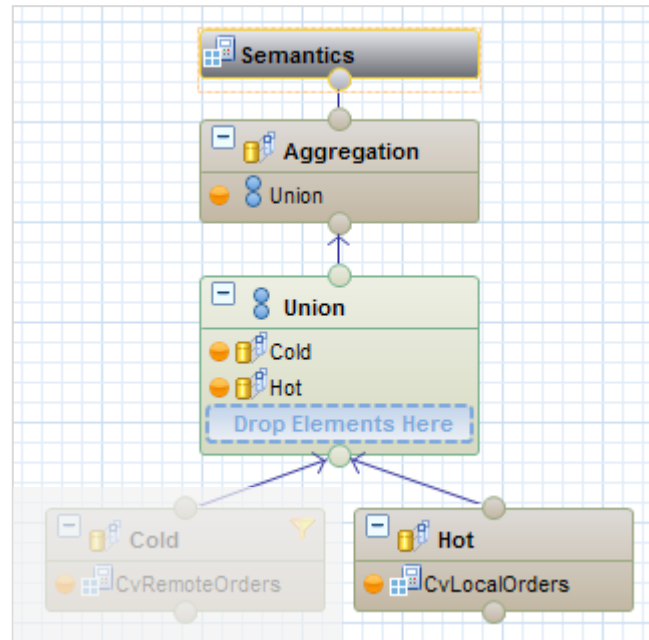
SAP HANA View Modeling – Best Practices

Essential Tips and Tricks - Input Source Pruning

Explicit pruning

```
select
  country, sum(previous), sum(current)
from CvSalesQuery
  where year = 2015 and
     country in ('usa', 'germany') and
     SOURCE = 'Hot'
group by country order by country
```

Query execution direction relies on the Union Constant filter



Implicit pruning

```
select
  country, sum(previous), sum(current)
from CvSalesQuery (
  'placeholder' = ('$$ip_year$$', '2015')
)
  where country in ('usa', 'germany')
group by country order by country
```


Query execution direction relies on the Date Input parameter

SAP HANA View Modeling – Best Practices

Essential Tips and Tricks – Keep Flag

Show Mouse sales in February!

Order	Month	Product	Store	Customer	Quantity	Price	Results
1	2	Ipod	TigerDirect	John	20	20	
2	2	Mouse	TigerDirect	Susan	2	5	10
3	2	Mouse	TigerDirect	John	2	5	10
4	2	Mouse	Amazon	John	2	5	10
5	2	Headset	Amazon	Susan	3	5	
6	2	Headset	Ebay	John	2	5	
7	2	Ipad	Amazon	Susan	3	250	
8	2	Ipad	Ebay	Susan	3	250	
9	2	Mouse	Amazon	Peter	2	5	10
10	3	Ipad	Amazon	John	3	250	



40

Regular SQL

SQL Result

```
SELECT "Month", "Product", sum("Quantity"), sum("Price")
FROM "DEMO"."demo.modeling.db::cds.StoreOrders"
WHERE ("Month" = '2' AND "Product" = 'Mouse')
group by "Month", "Product"
```

	Month	Product	SUM(Quantity)	SUM(Price)	Sales
1	2	Mouse	8	20	\$160

Generally the requested columns mandates the level of aggregation. Under certain circumstances this level of aggregation is too coarse and instead requires a finer level of aggregation.

- Setting the **Keep flag** will force additional columns into context altering the level of granularity.

Solution!

Within Calculation Model Set
Keep Flag = True on **Order ID**

SAP HANA View Modeling – Best Practices

Essential Tips and Tricks – Transparent Filter

Count the amount of stores that sold a Mouse to Johan and Susan!

```
select product, sum(quantity), sum(storecount) from model
where Customer in (john, susan) and product = mouse group by product
```

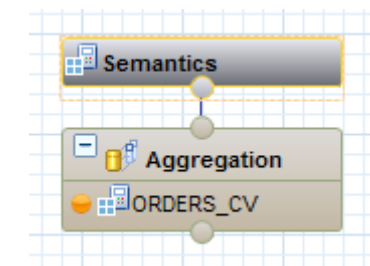
Order	Month	Product	Store	Customer	Quantity	Price
1	2	Ipod	TigerDirect	John	20	20
2	2	Mouse	TigerDirect	Susan	2	5
3	2	Mouse	TigerDirect	John	2	5
4	2	Mouse	Amazon	John	2	5
5	2	Headset	Amazon	Susan	3	5
6	2	Headset	Ebay	John	2	5
7	2	Ipad	Amazon	Susan	3	250
8	2	Ipad	Ebay	Susan	3	250
9	3	Ipad	Amazon	John	3	250
10	2	Mouse	Amazon	Peter	3	250

Results

Product	Quantity	StoreCount
Mouse	6	2

Stacked Calculation Model

AB	PR...	AB	CU...	12	StoreCount_SUM
			John		2
			Susan		1



Solution! Set the Transparency Filter for the column **Customer** to true, on all models and nodes that references the Customer

The Transparent Filter will omit columns from unnecessary being used part of Group By's. This Flag is required when using ...

- **Stacked** views where the lower views have count distinct measures
- Queries on the upper Calculation view contain **filters** on columns that are not projected

SAP HANA View Modeling – Best Practices

Essential Tips and Tricks – Dynamic Join

Show the Sales & Total Sales by **Region** & Product!

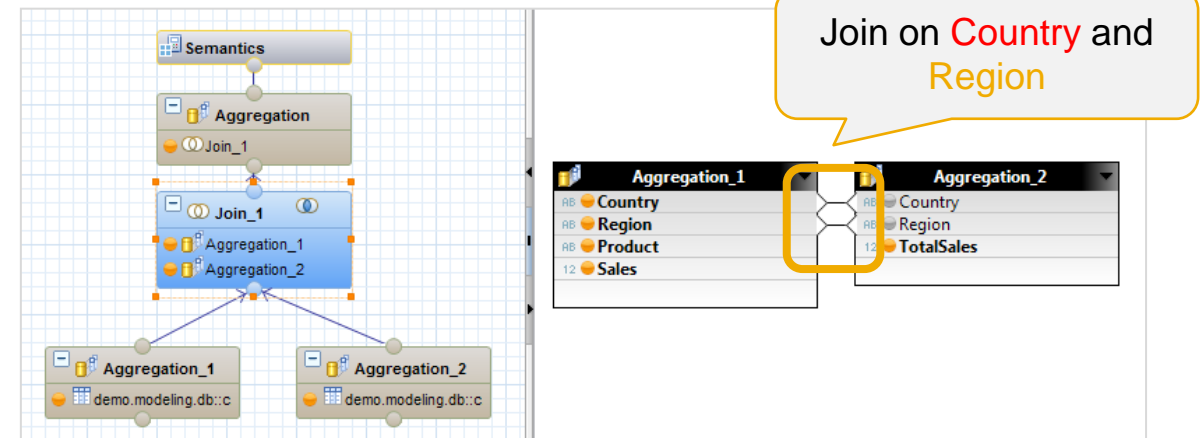
Region	Country	Product	Sale
APJ	JP	HT-1020	100
APJ	JP	HT-1001	100
APJ	AT	HT-1020	100

Region	Product	Sale
APJ	HT-1020	200
APJ	HT-1001	100

Region	Total Sales
APJ	300

Region	Product	Sales	Total Sales
APJ	HT-1020	200	300
APJ	HT-1001	100	300

Dynamic Join



Region	Country	Product	Sale
APJ	JP	HT-1020	100
APJ	JP	HT-1001	100
APJ	AT	HT-1020	100

Region	Country	Total Sales
APJ	JP	200
APJ	AT	100

Region	Country	Product	Sale	Total Sales
APJ	AT	HT-1020	100	100
APJ	JP	HT-1020	100	200
APJ	JP	HT-1001	100	200

Region	Product	Sale	Total Sales
APJ	HT-1001	100	200
APJ	HT-1020	200	300

Regular Join

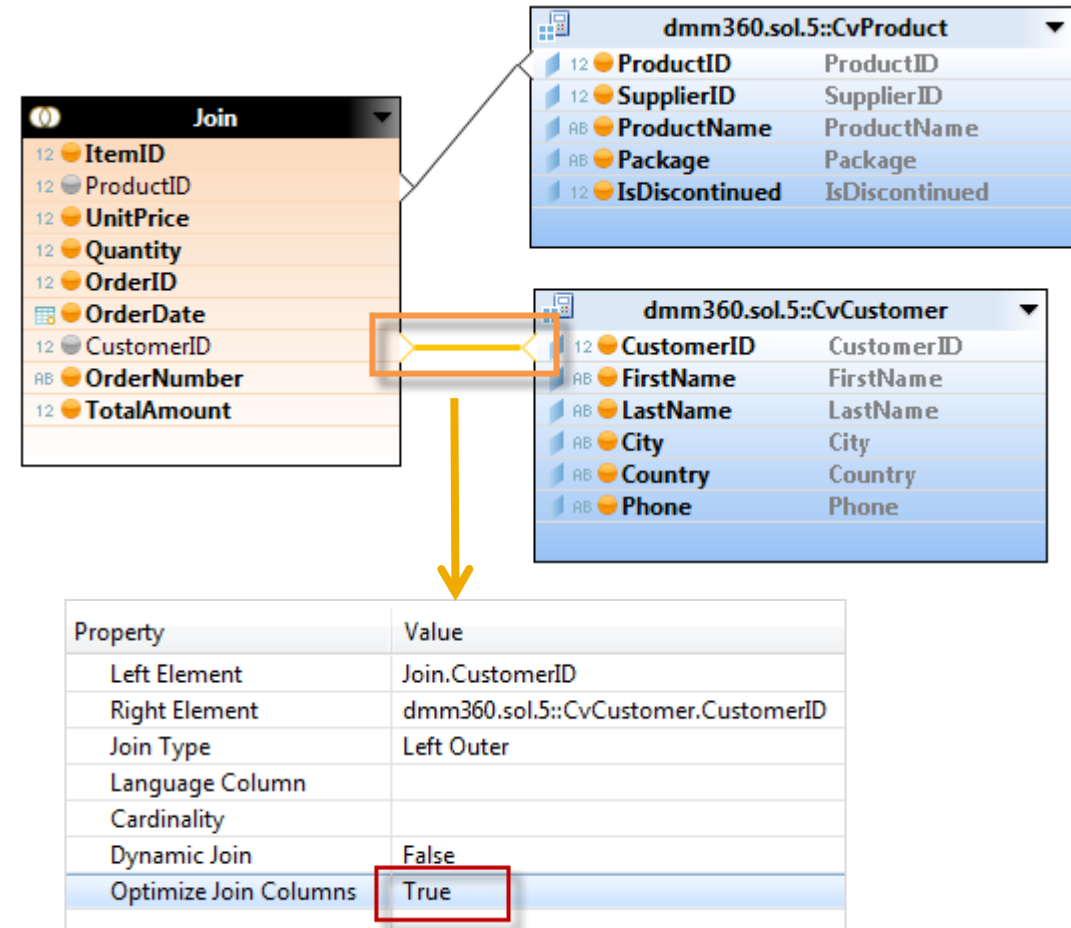
Only the Join columns requested in the query will be brought into context and play a part in the Join execution

Essential Tips and Tricks – Optimize Join Columns

Generally tables are pruned if no columns are requested and Join types such as Left-Outer or Referential are involved; under certain circumstances the Joined columns could be brought into context altering the level of aggregation.

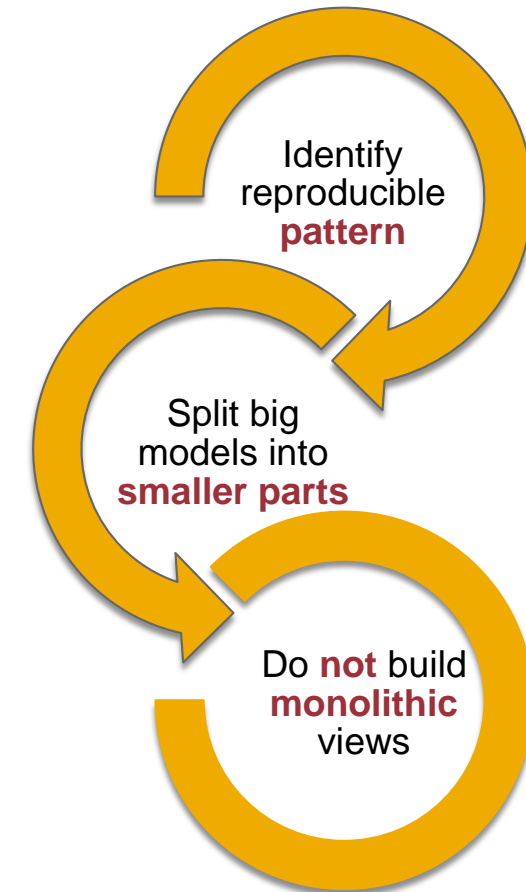
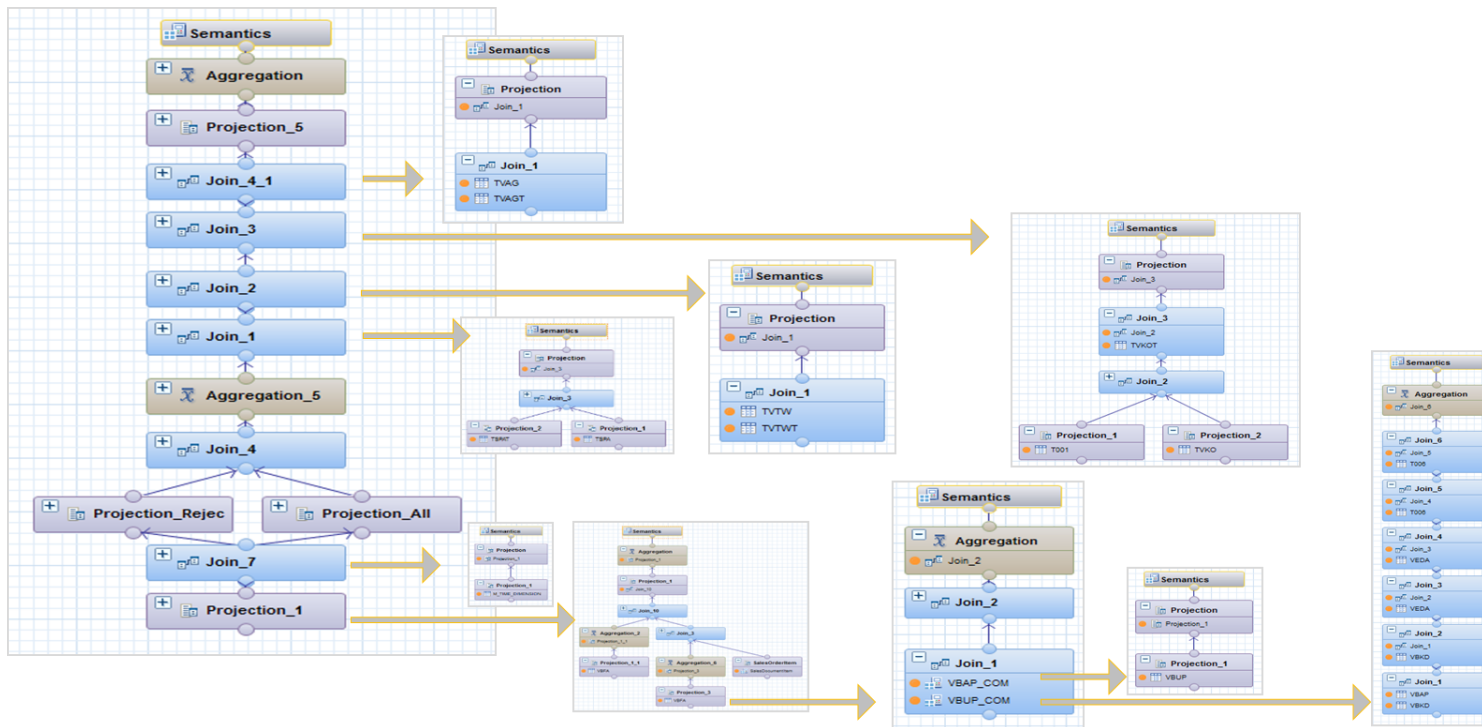
This setting will ensure that the joined columns are **omitted** and will lead to smaller intermediate result sets which can improve performance significantly depending on the type of query.

*Note: This feature generally applies to earlier revisions



Designing larger Virtual Data Models using HANA Views

Architecting complex SAP HANA View-based data models

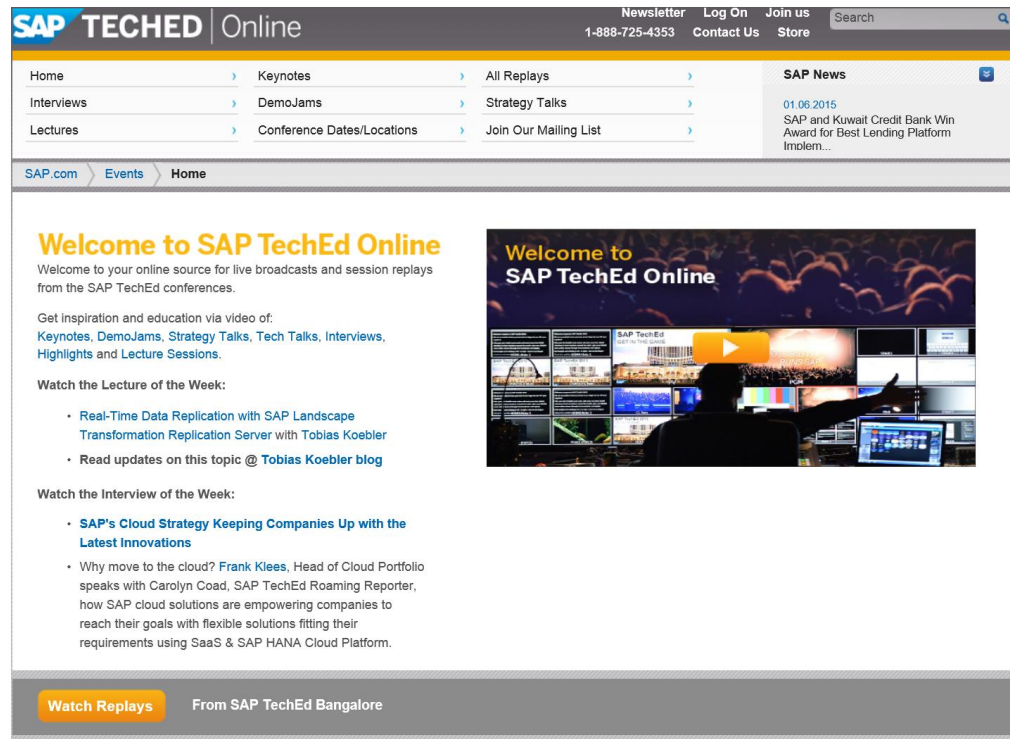


- Learn from SAP HANA Live! Layering approach.

SAP TechEd Online

Continue your SAP TechEd education after the event!

- Access replays of keynotes, Demo Jam, SAP TechEd live interviews, select lecture sessions, and more!
- Hands-on replays



<http://sapteched.com/online>

Further Information

Related SAP TechEd sessions:

DMM360 – Advanced Data Modeling in SAP HANA, Hands-On Workshop

DMM163 – Introduction to Data Modeling in SAP HANA, Hands-On Workshop

SAP Public Web

scn.sap.com <http://scn.sap.com/community/developer-center/hana> <http://scn.sap.com/community/hana-in-memory>

www.sap.com www.saphana.com

What's new in SAP HANA (incl. SAP HANA View Modeling)

<https://blogs.saphana.com/2014/10/21/what-is-new-in-sap-hana-sps-09/> <https://blogs.saphana.com/2015/06/16/new-sap-hana-sps10/>

SAP Education and Certification Opportunities

www.sap.com/education

<https://www.youtube.com/user/saphanaacademy> <https://open.sap.com/courses>

Watch SAP TechEd Online

www.saptech.com/online

Feedback

Please complete your session evaluation for
DMM208





Thank you!

Contact information:

Christoph Morgen

SAP HANA Product Management

SAP SE | Dietmar-Hopp-Allee 16 | 69190 Walldorf | Germany

christoph.morgen@sap.com | www.sap.com

Werner Steyn

SAP HANA Competence Center, SAP SE

werner.steyn@sap.com | www.sap.com

© 2015 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. Please see <http://global12.sap.com/corporate-en/legal/copyright/index.epx> for additional trademark information and notices.

Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors.

National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP SE or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP SE or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, which speak only as of their dates, and they should not be relied upon in making purchasing decisions.