Capgemini

SAP HANA

Lesson Name: HANA Modeling

Contents

Introduction

Before introducing modelling in SAP HANA, you must become familiar with some key concepts that are frequently used when reporting on financial or operational data.

- Measure
- Attribute
- Dimension
- Star Schema
- Hierarchy
- Semantics

Introduction

Measure and Attribute

Measure vs. Attribute

	Measure	Attribute
Definition	A numeric value, such as a price, quantity, volume, on which you can process arithmetic/statistics operations, such as sum, average, top n values, and calculations.	An element that is used to describe a measure.
Examples	■ Number of products sold ■ Unit Price ■ Total Price	■ Product ID ■ Product Name ■ Customer ID ■ Customer Name ■ Sales Organization ■ Sales Org. Country ■ Sales Org. Region ■ Currency

Dimension

- Analyzing measures are easier if you group attributes together by Dimension.
- Example: a Product ID dimension could be associated to several attributes, such as Product Name, Product Category, or Supplier.

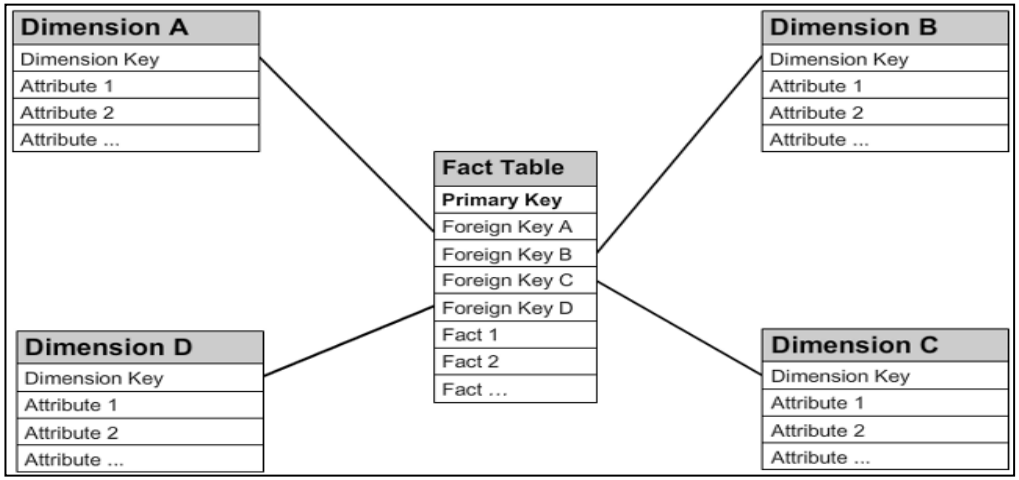
Dimension 1	Dimension 2	Dimension n
Product	**Sales Org.**	**Dim. name**
Product Key	Sales Org. Key	Key
Product Name	Sales Org. Name	Name
Product Category	Country	Attribute x
Supplier	Region	Attribute y

Introduction

Star Schema

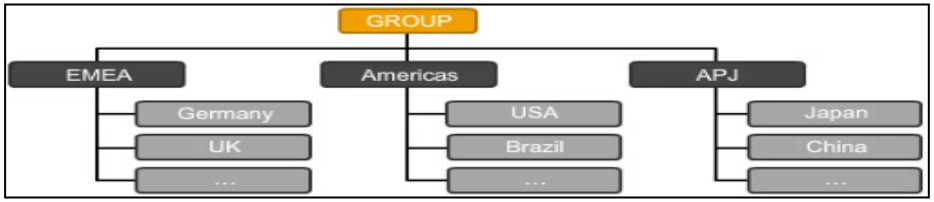
A star schema consists of one fact table that references one or several dimension tables.

The fact table contains Facts, or Measures, as well as the keys used to identify- for each dimension- which dimension member corresponds to each fact.



Hierarchy

A hierarchy is a structured representation of an organization, a list of products, the time dimension, and so on, by levels.



Introduction

Semantics

- Describes what a data means, or relate to

Example of Semantics:

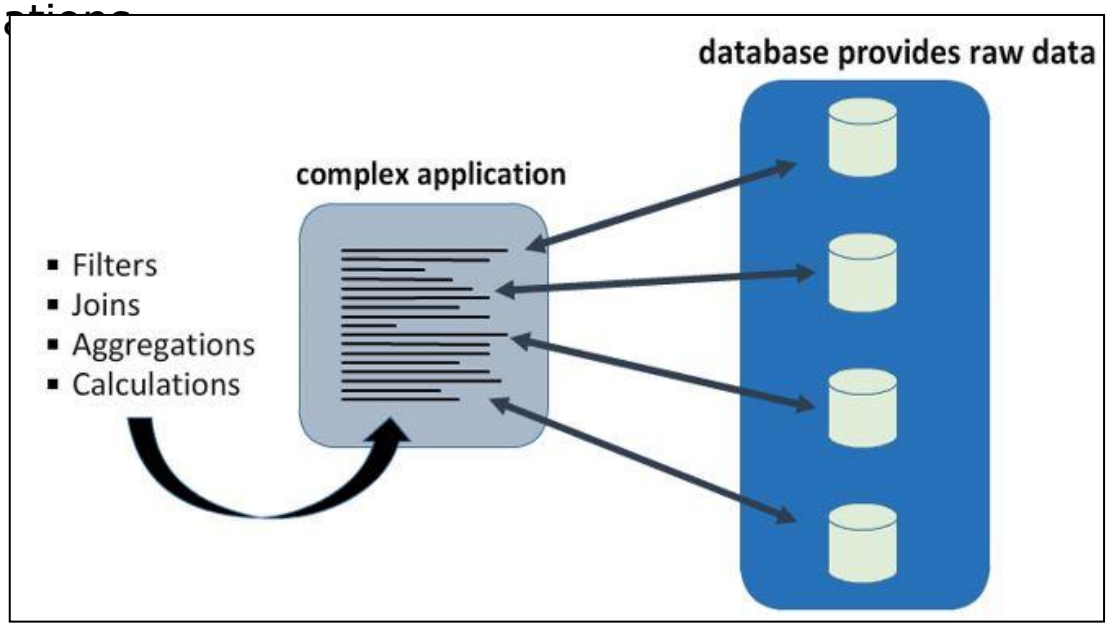
- A monetary value
- A number of items
- A weight, volume, distance or a compound of these measures.
- A Percentage

Real Time Example: Before summing up the amount of sales orders, you must make sure they are all expressed in the same currency.

Why Modeling

Traditional Approach

- The role of the database is to provide data.
- Application sends down Select statements to individual database tables.
- The raw data is sent from the database to the application.
- The application then begins to process the data by combining it, aggregating, and performing calculations



- Here the application does all the Hard Work

Why Modeling

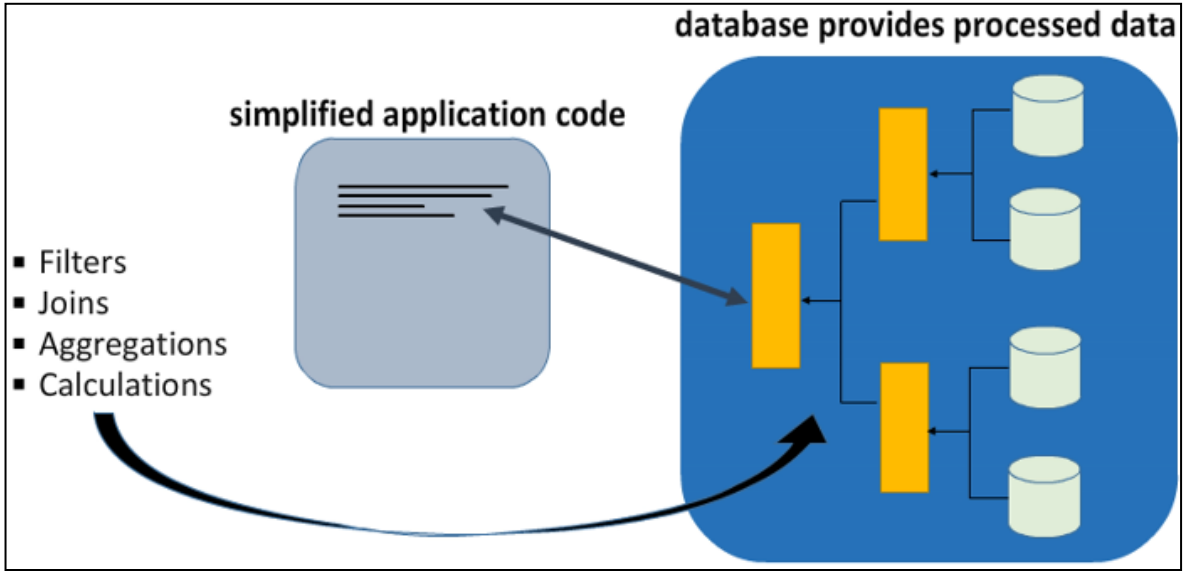
This results in :

- Moving a lot of raw data between the database and the application.
- Application code becomes complex.
- Performance degradation.

Why Modeling

HANA Approach

- One can build a modelling layer on top of the database tables, to provide data to the application in a ready to go, processed form.
- With SAP HANA, one can build Information Views that combines data from multiple tables and apply filters, conditions, calculations, aggregations.
- Information views are developed using easy to use modeling tools and are stored in SAP HANA alongside the database tables in the database catalog.
- Here Application is not the Hard Worker as it simply calls the required Information View and the processing is pushed down to the SAP HANA.



Why Modeling

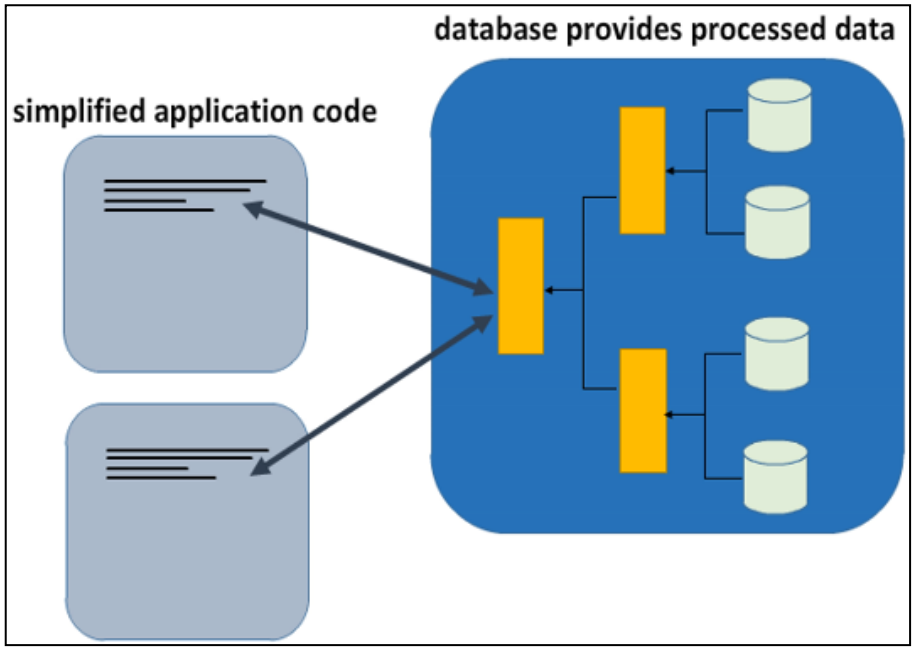
Pros:

- Application code is simplified.
- All Calculations are performed on the fly within the database engines.
- Only move the result instead of moving raw data.
- Very fast response time as processing is carried out in-memory.
- Information Views can be reused.
- Flexibility
- Adaptability
- Easy to transport

Why Modeling

Reuse of Models

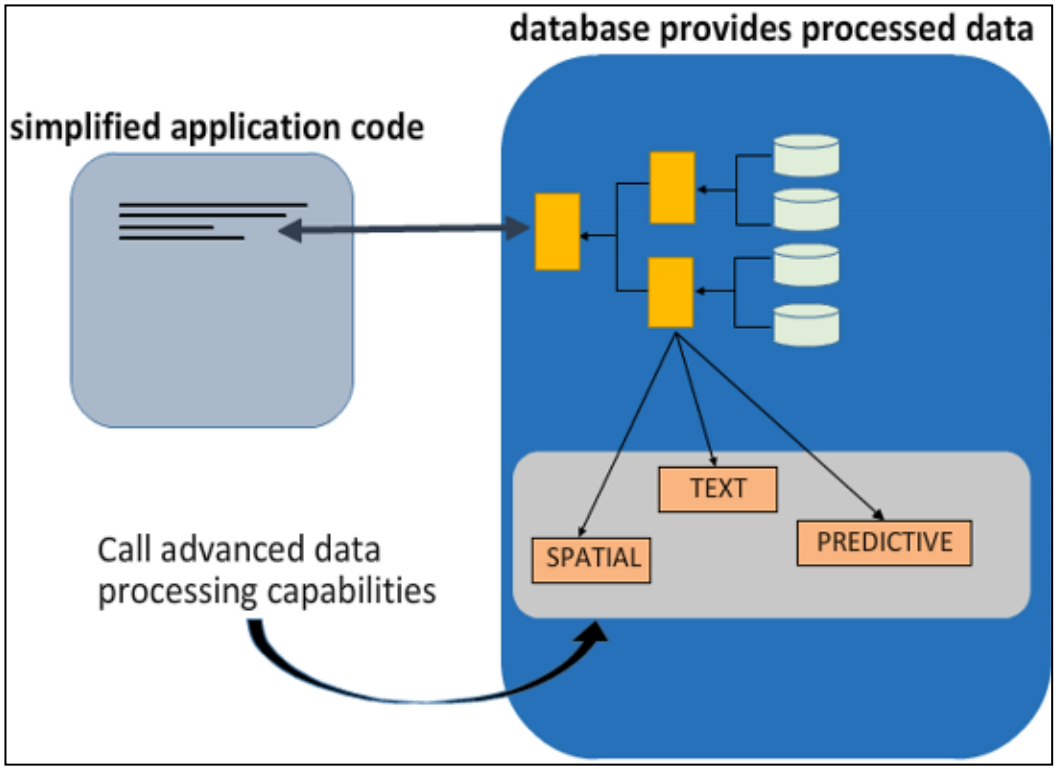
- With traditional approach there is a high degree of redundancy in the application code.
- This redundancy can be avoided by using SAP HANA views.
- SAP HANA views can contain dynamic placeholders.
- This means application can pass variables down to the view.
- Many of the views can also call procedures that have input parameters.
- These information views can consume other information views.



Why Modeling

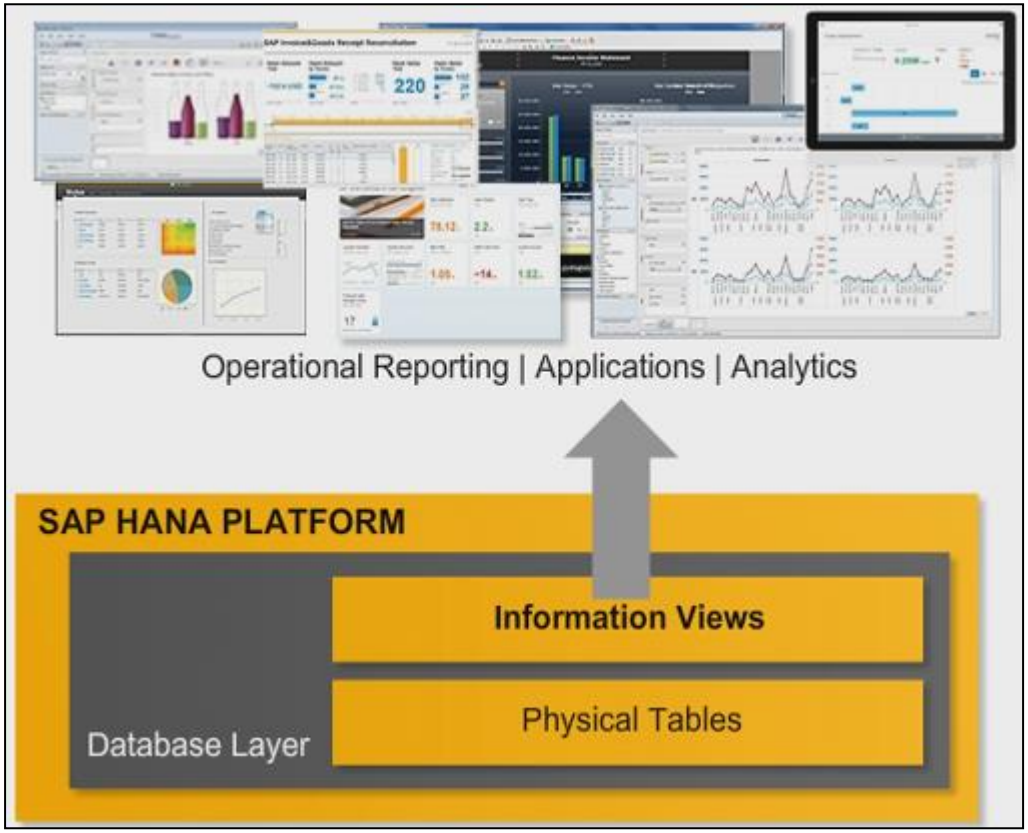
Advanced Data Processing Capabilities

- SAP HANA has built in advanced data processing capabilities.
- This includes textual, spatial and predictive functions.
- Information views can easily call these native SAP HANA functions and so the applications can leave all the complex processing to SAP HANA.



Information Views in SAP HANA

Information Views are used in SAP HANA to create a Virtual Data Model, based on the data that resides in the SAP HANA database schemas.



Information Views in SAP HANA

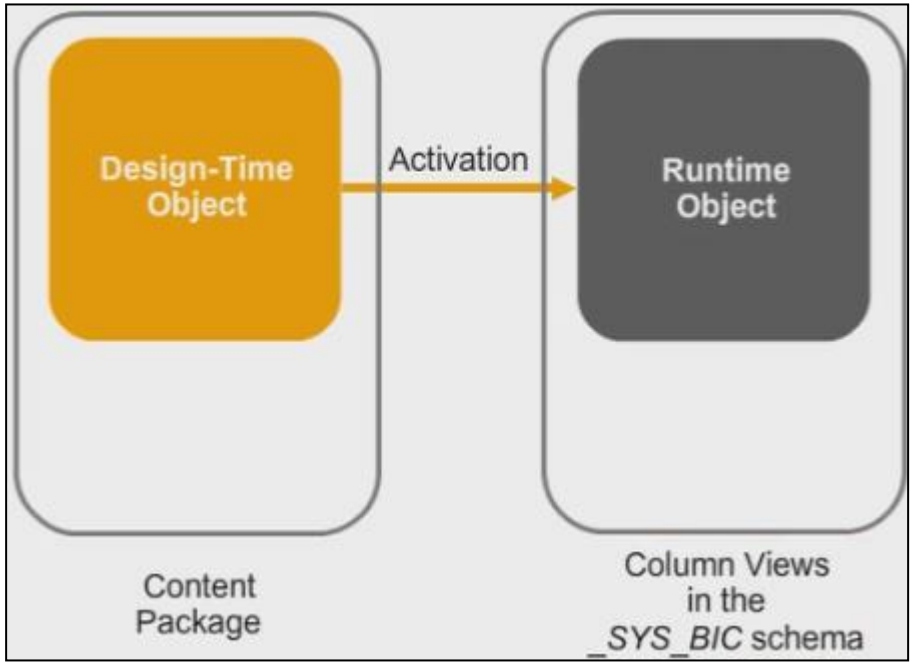
The purpose of information views is to:

- Organize the data from the individual transactional tables.
- To perform a variety of data calculation

Data can be made more meaningful than in the source tables by customizing the column names, assigning label columns to key columns, calculating additional attributes, and so on.

Design Time Vs Run Time Instances

When you create or modify an information view, with SAP HANA Studio or the Web Development Workbench, you work on a Design-Time Object, which is located in a package of the SAP HANA Content.



Design Time Vs Run Time Instances

Packages are organized into a hierarchy, and authorizations to access them can be defined at the package level.

Then, the activation of this Design-Time object creates a Runtime Object, which is a Column View located in a dedicated SCHEMA of the SAP HANA Database: the _SYS_BIC schema.

Demo

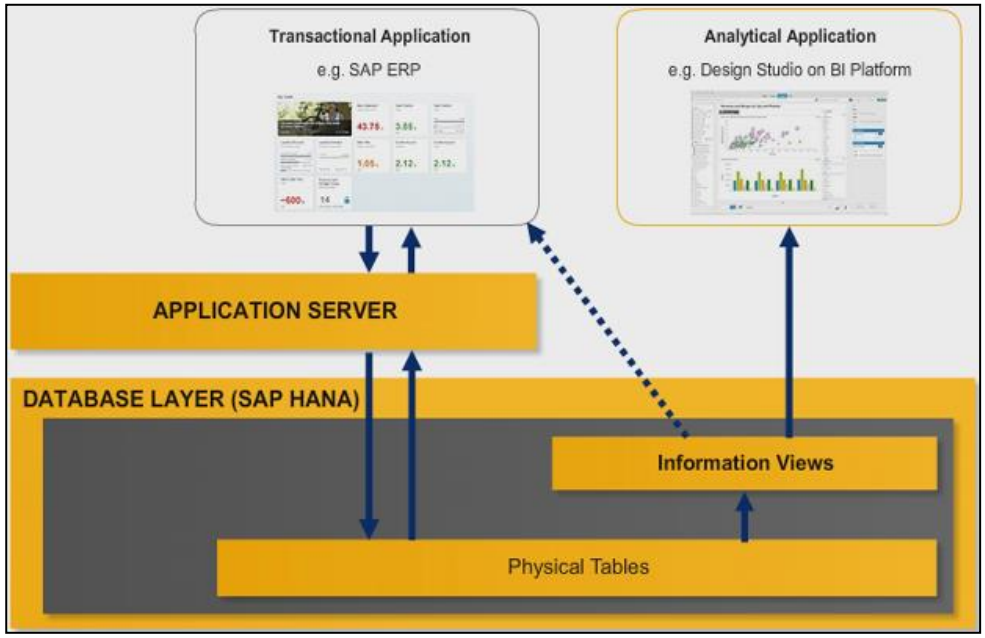
Demonstrate by creating an information view, how objects get created in _SYS_BIC schema.

Analytical Vs Transactional Requirements

In transactional applications, such as SAP ERP, the underlying data (stored in physical tables) are generally handled by the application server.

This layer is necessary to handle the business process, however complex it can be, and still ensuring data consistency at any time when updating multiple tables, checking the user's authorizations, and so on.

On the other hand, Information Views are used only to retrieve data, without making any changes to the source transactional data.

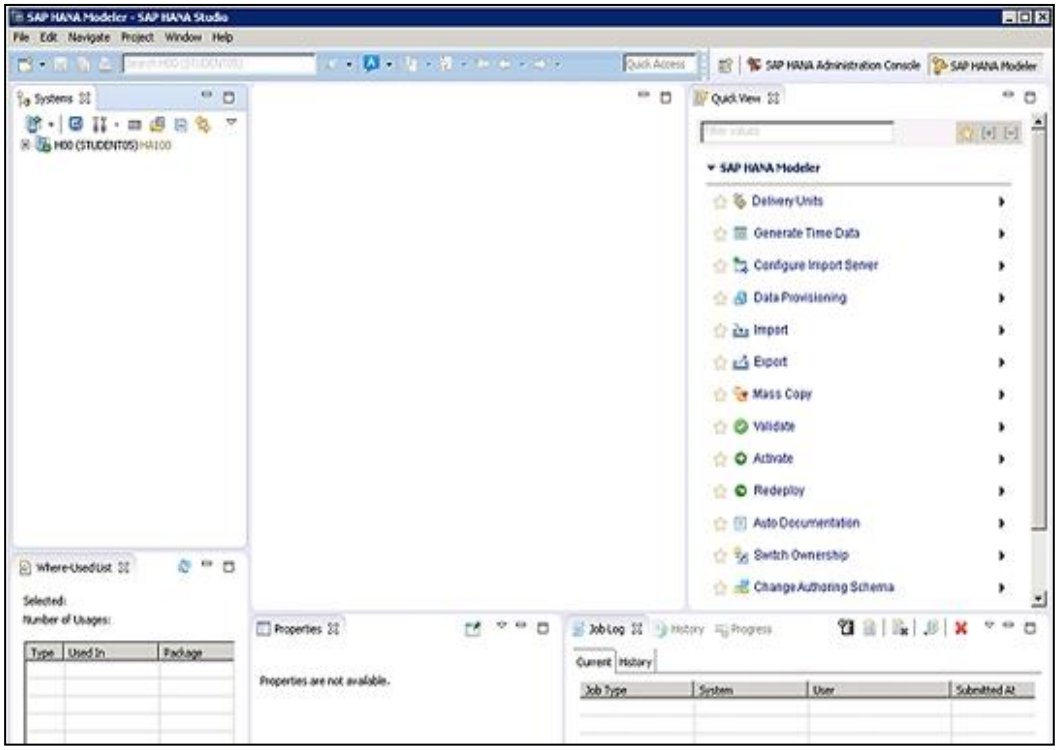


Working with SAP HANA Modeling Interface

Information views can be created in the SAP HANA Studio and also in the Web Workbench.

There are some limitations when using the Web Workbench.

Below Diagram shows the Modeler perspective In the SAP HANA Studio.



Working with SAP HANA Modeling Interface

Only calculation views — of any type — can be created and maintained in the Web Workbench whereas attribute, analytical and as well as calculation views can be maintained in the SAP HANA Studio.

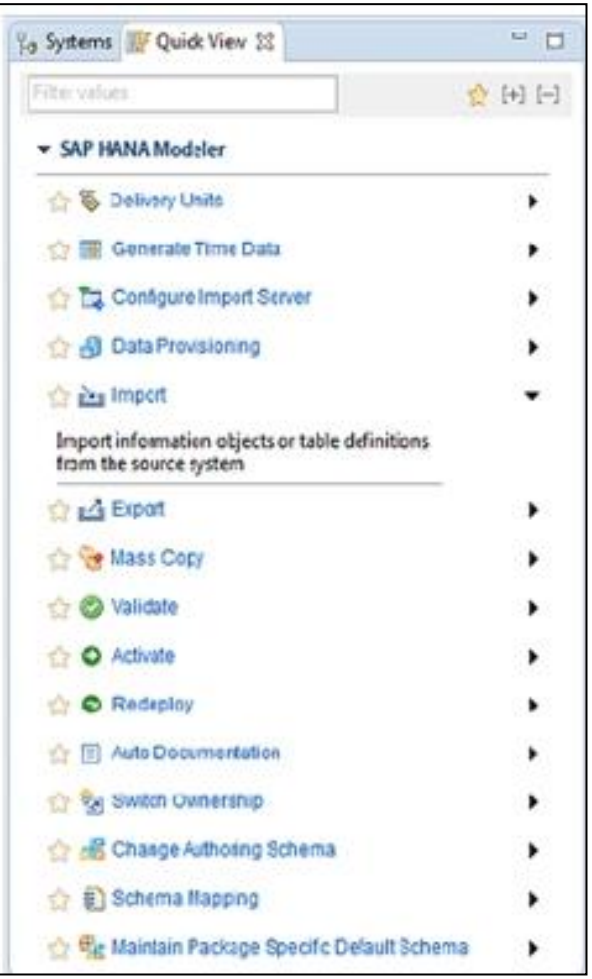
Following activities are supported by Modeling Perspective:

- Creating information objects such as attribute, analytic and calculation views, stored procedures and analytic privileges.
- Processing information models.
- Managing modeling content by performing multiple administration activities.
- Importing table definition from the source ERP system into SAP HANA studio.
- Loading data into these table definitions.

The Modeler Quick View

The Modeler Quick View provides easy access to frequently performed modeling tasks in the SAP HANA Studio. You access this view from the Help menu option.

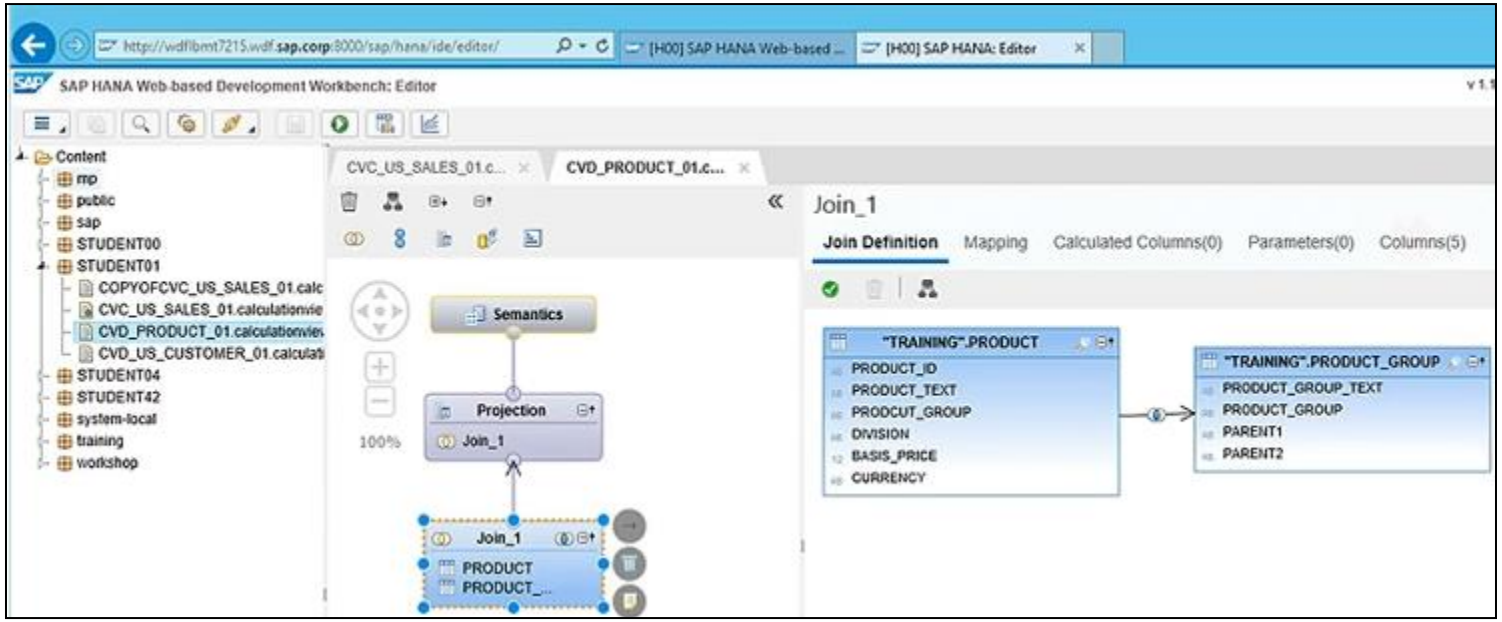
- Easy access to frequently performed tasks via the new *Quick* View pane (replacing *Quick Launch*)
- Available as a View in SAP HANA studio
- Allows you to quickly navigate and perform modeling tasks such as Validate, Activate, Schema Mapping, Import/Export…



Modeling with the SAP HANA Workbench

The SAP HANA Web Workbench can be used to create and maintain calculation views of any type.

One will not have access to many of the administration functions from the SAP HANA Web Workbench, such as schema mapping, and date generation. For those functions you should use the SAP HANA Studio.



Modeling with the SAP HANA Workbench

Key Benefit of using Web Workbench:

- You do not have to install and maintain any software locally.
- The SAP HANA Web Workbench is accessed through a browser.
- Layout of the SAP HANA Web Workbench is very similar to the SAP HANA Studio layout when modeling, so it is easy to jump between the two interfaces.
- Calculation views created with one interface can be accessed with the other.
- The URL for the Web Workbench is: http://<host>:80<instance>/sap/hana/xs/ide.
- Logon with the same user and password as you would with SAP HANA Studio.

Demo

Demonstrate how to start working with SAP HANA Studio and SAP Web Workbench if not covered in earlier sessions.

Demonstrate by creating an information view, information views are created in SAP HANA Studio and SAP Web Workbench.

Open and show the modeler quick view in SAP HANA Studio to show the frequently performed tasks in modeling.

Information Views in SAP HANA

The following three Information Views are available in SAP HANA:

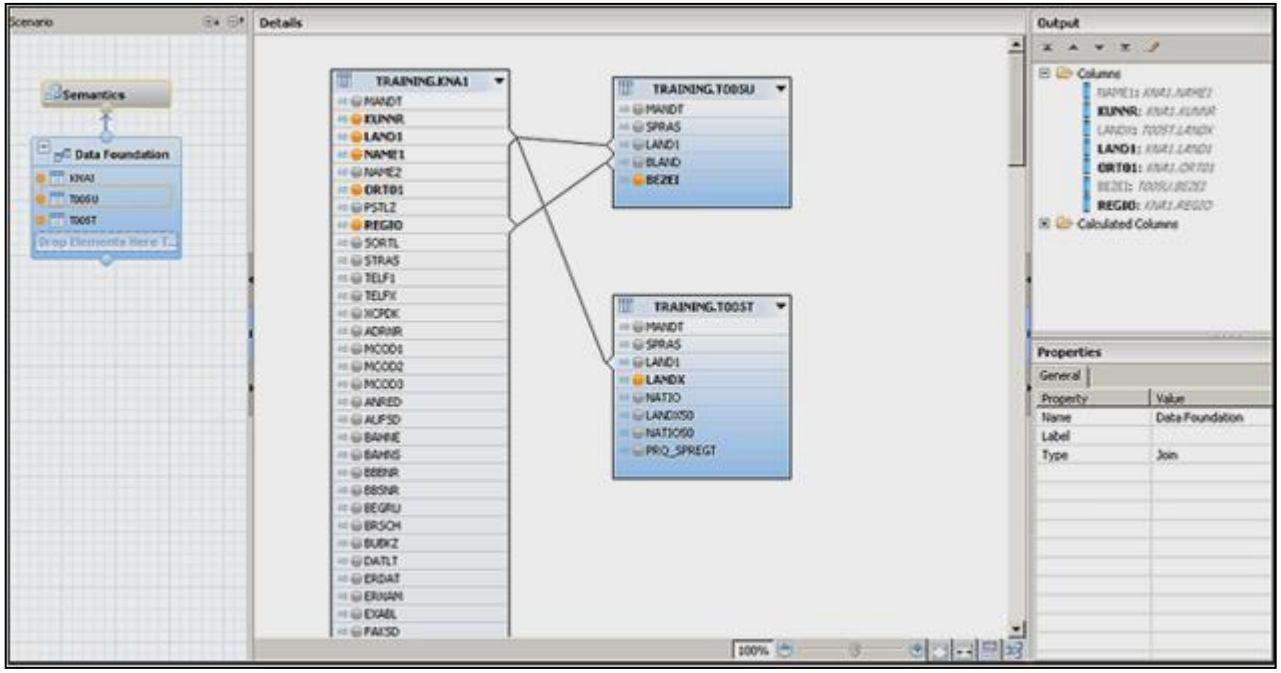
- Attribute View
- Analytic View
- Calculation View

Information Views in SAP HANA

- The most important type of view is calculation view.
- Attribute and analytic views have been available since the first release of SAP HANA, but have become less important as newer releases of SAP HANA deliver more powerful calculation views that can take over functionality of the other two views.
- However, there are just one or two modeling features in attribute and analytic views that, as of SPS11, are not yet available in the calculation view.
- This means that the attribute and analytic views are still needed for now.
- Migration tools are available to convert the existing attribute and analytic views to calculation views.
- But first we'll see attribute and analytic views.

Attribute View

- What is an Attribute View?
  - Attributes add context to data.
  - Can be regarded as Master Data tables.
  - Can be linked to fact tables in Analytical Views.

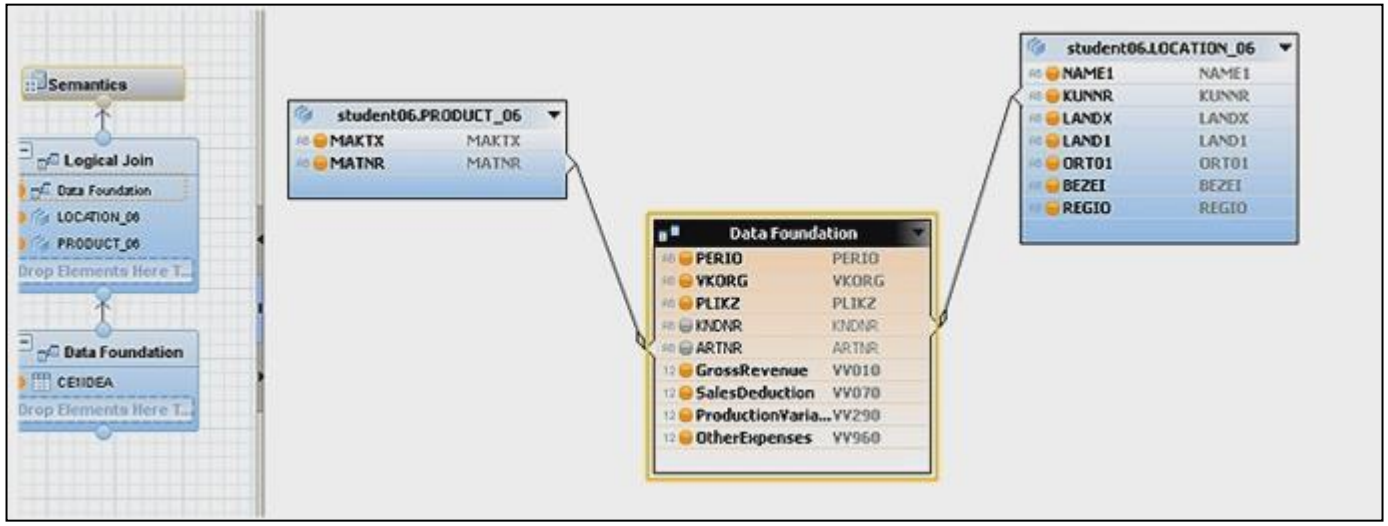


Attribute View

- An attribute view is used to group related attributes from one or more tables to form a list that can be accessed via SQL or consumed in another view.
- Since the arrival of the calculation view of type DIMENSION there is very little reason to create this type of view as all functionality has been included in the calculation view of type DIMENSION.
- But still there are certain functionalities that are not available in calculation view.
- SAP provide a tool for migration of attribute views to calculation views of type DIMENSION.

Analytic View

- An Analytic View can be regarded as a “cube”
- Multidimensional reporting tool.
- Fact table (data foundation) joined against modeled dimensions (attribute views).
- Analytic views do not store data
- Data is read from the joined database tables.
- Joins and calculated measures are evaluated at run time.
- Master data for MDX/BICS are stored in system tables.



Analytic View

- Analytic views are used to develop star schemas where a central fact table is surrounded by dimensions to create a multidimensional data set used for OLAP reporting.
- The only functionality that has not yet been included in calculation views of type CUBE, is the temporal join.
- Temporal joins provide joins between the fact table and the dimensions but also include extra date parameters so the fact row can be joined to the correct dimension row based on a specific date.
- Temporal joins are heavily used in data warehousing when dimensions contain a lot of historical attributes that need to be carefully joined to the facts at precise dates.
- Note: It can be expected that temporal joins will become available in the calculation view type CUBE in the very near future. This will truly render analytic views redundant.

Demo

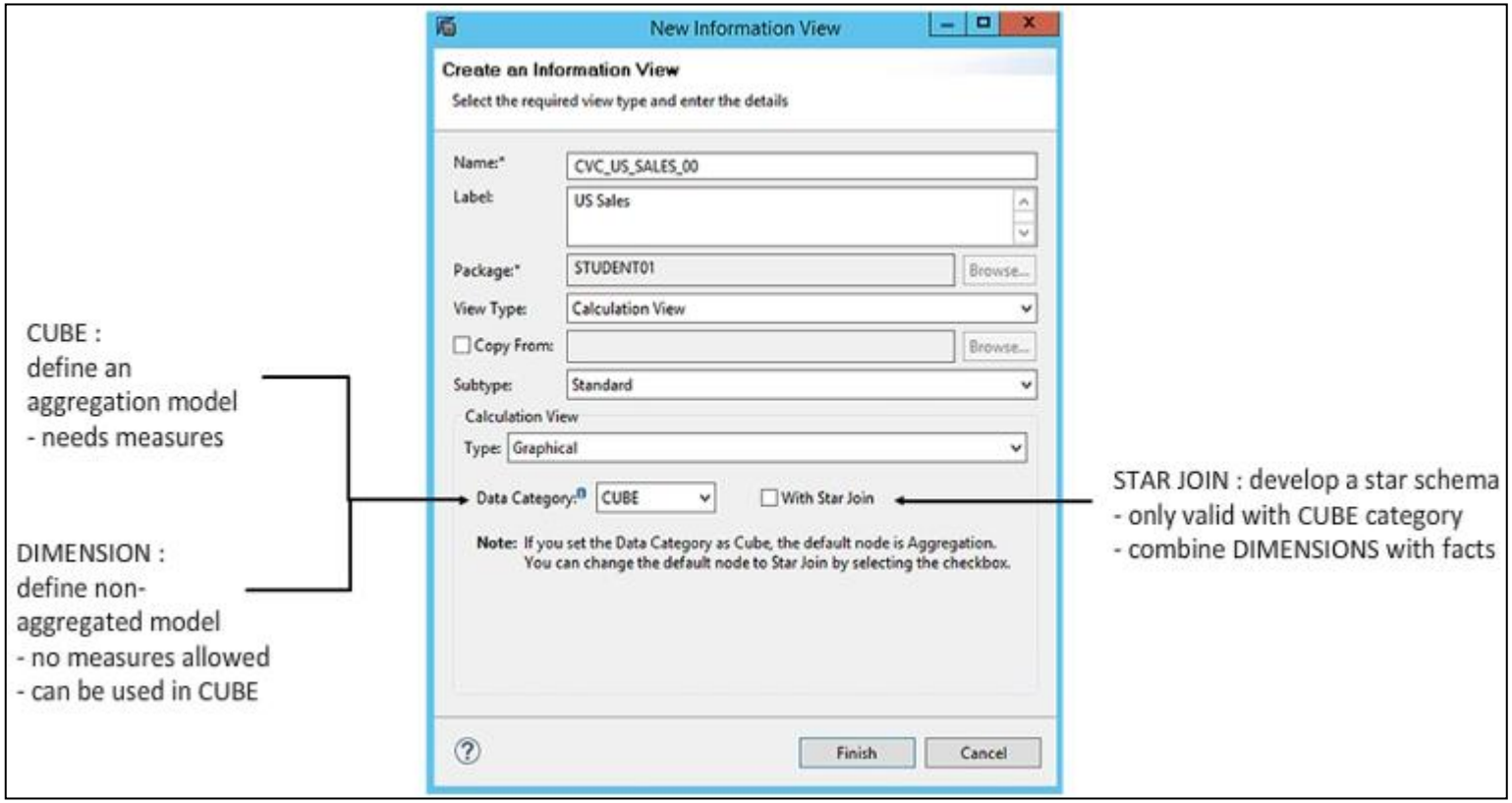
Creation of Attribute view.

Creation of Analytic view.

Information View

Types of Calculation View

- Dimension
- Cube without Star Schema
- Cube with Star Schema



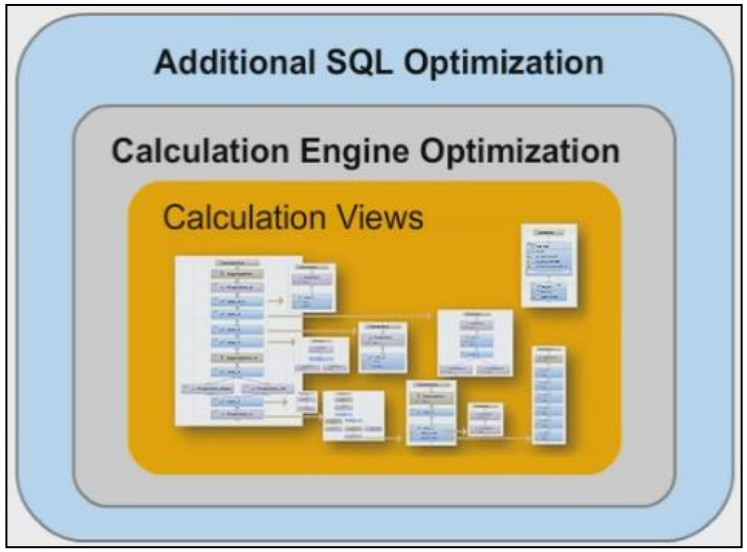
Graphical Calculation Views

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 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 (calculations 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.

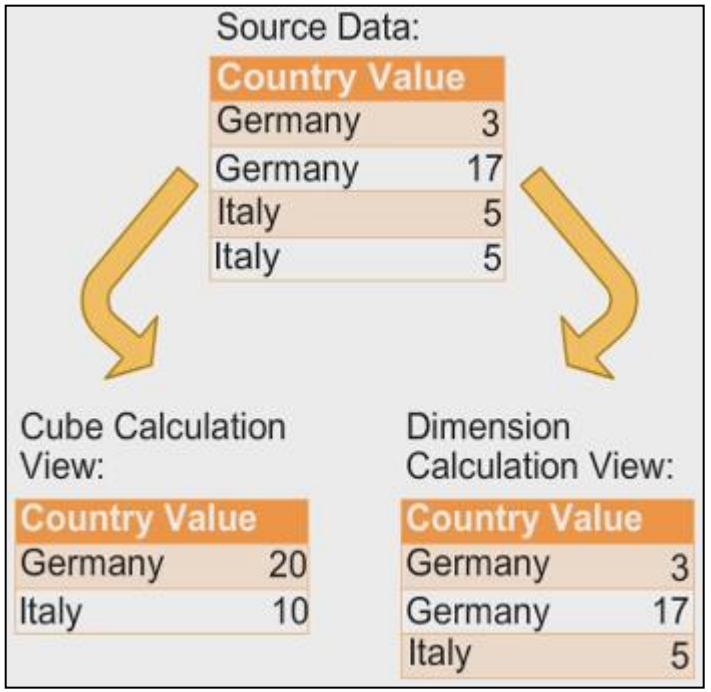
Overview of Calculation View

- In SAP HANA SPS10 and above, when you query a Calculation View, the optimization is made in two main steps:
  - 1. The initial Calculation Engine optimization generates a single SQL statement across a stacked model, which is then passed to the SQL optimizer.
  - 2. The SQL optimizer adds additional optimizations and delegates operations to the best database execution operator.
- For example, it delegates the execution of Star Join to the OLAP Engine whenever it is possible.



DIMENSION Calculation Views

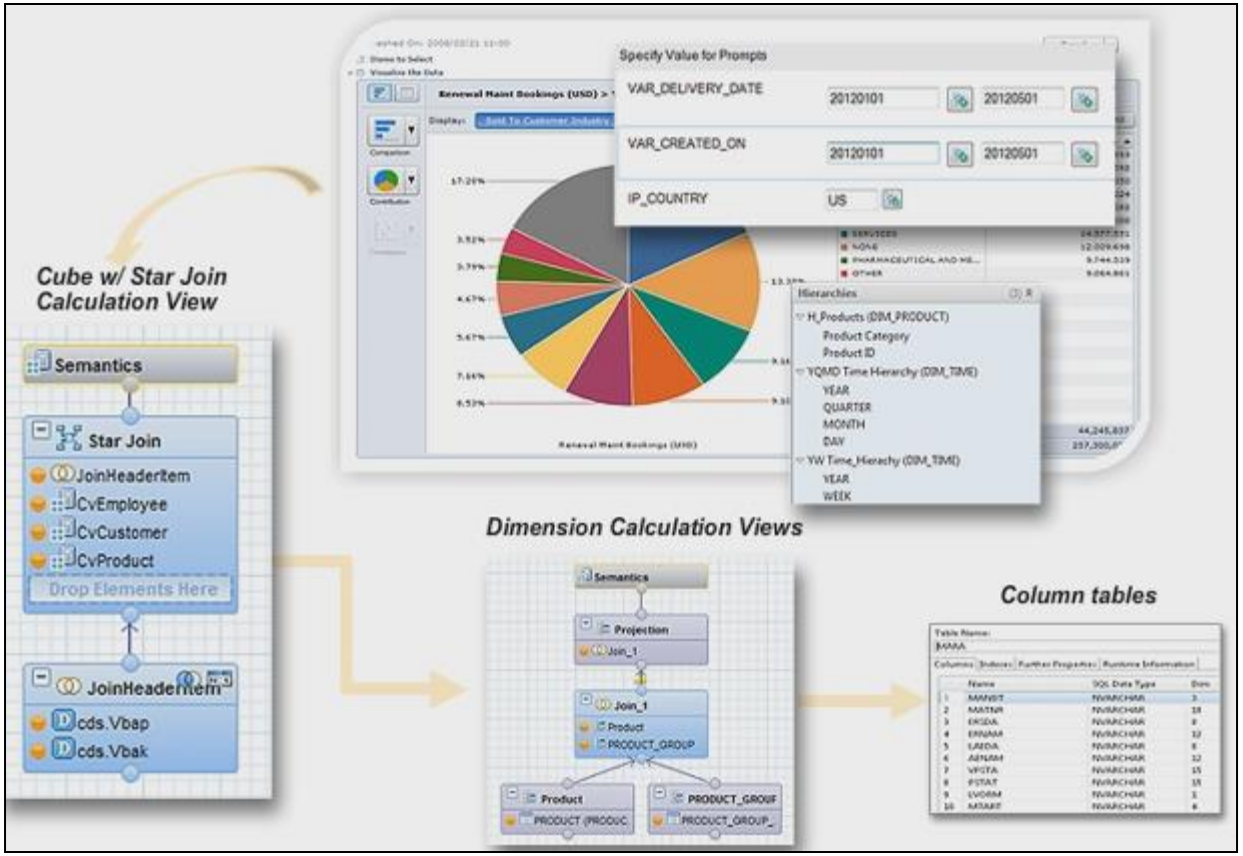
- A Dimension calculation view does not handle measures.
- Any column is always considered as an attribute.



- As a result, aggregation (if any) is always interpreted as a select distinct.
- Dimension calculation views do not allow measures: any numeric columns or values will be treated as attributes.

CUBE Calculation Views

- When you analyze data including measures, you use a Calculation View of type CUBE or ->CUBE with Star Join.



CUBE Calculation Views

- These types of views provide a number of features:
  - To filter data,
  - Control the aggregation of data,
  - Perform complex data calculations on measures,
  - Combine data from several data sets,
  - Retrieve sub-sets of data based on measure ranking (for example, the top n countries for Revenue or Margin), and so on.

- In this scenario, a CUBE with Star Join Calculation View enables a multidimensional reporting that leverages the source data (from the Vbap and ->Vbak tables) and dimension Calculation Views created for the main dimensions, such as ->Product, ->Employee or ->Customer.

Demo

Creation of Calculation View of type DIMESION.

Creation of Calculation View of type CUBE.

Creation of Calculation View of type CUBE with Star Join.

Features from Deprecated Views not Supported in Calculation View

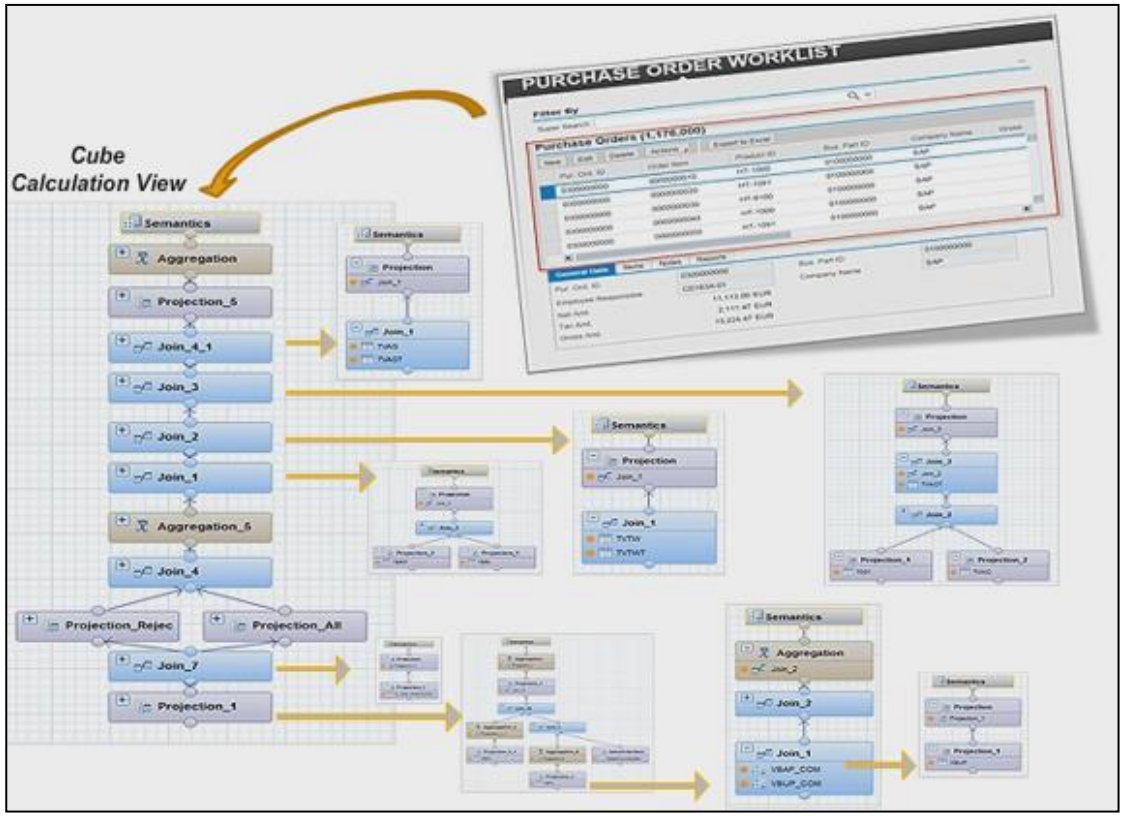
Functional Difference between Information Views Types

Feature	Available only in	Comments
Fuzzy Search	Attribute Views	It is possible to enable one or several columns of an Attribute Views for Fuzzy Search. This is not possible in Calculation Views
Temporal Join	*Star Join* node of Analytic Views	It is possible to join a fact table (Data Foundation) and the corresponding master data (Attribute View) based on temporal conditions.
Derived Attribute View	Attribute Views	If an Attribute View must be used several times in the same Analytic View, you can create a Derived Attribute View. This is a "dynamic" copy of the base Attribute View.

Apart from these limitations to their usage, Calculation Views are definitely the recommended graphical modeling object type.

Virtual Data Models for Normalized Data

- In this second scenario, SAP HANA Calculation Views typically feed data to Business Applications, such as SAP HANA XS build Applications or Enterprise Analytical Applications.



Virtual Data Models for Normalized Data

- They provide the means to model sophisticated views based on normalized data structures.
- To model the Purchase Order Worklist, a Cube Calculation View combines together a number of different source views, with Joins, Projections, Aggregations, and so on.
- Each of the different source views provide a custom set of measures and/or attributes.

Supported Data Source Types in Graphical Calculation Views

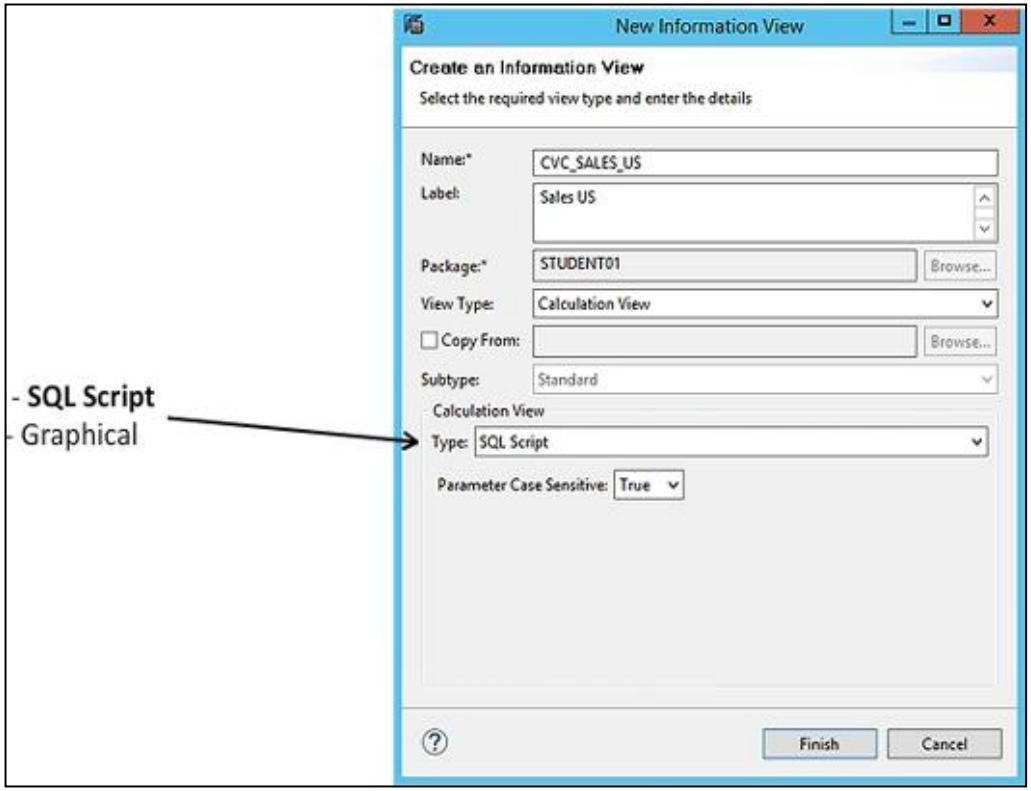
- Row and Column Tables from the same database.
- Row and Column tables from another database of the same HANA System, in a Multi Database Container configuration.
- Core Data Services (CDS) Entities.
- SQL Views.
- Graphical Calculation Views
- Note: Only Calculation Views of type DIMENSION are supported in the Star Join node of a Calculation View.
- Calculation views from another database of the same HANA System, in a MDC configuration.
- There are some limitations to this support. For example, “remote” DIMENSION Calculation Views cannot be used in the Star Join of a Calculation View.
- Table Functions

Supported Data Source Types in Graphical Calculation Views

- Decision Tables
- Virtual Tables(data provisioned through SAP HANA Smart Data Access)
- Analytic Views (deprecated).
- Script- based Calculation Views (deprecated).

Calculation View (Scripted)

- SQLScript is used to create scripted calculation view.
- SQLScript is an SAP HANA query language developed by SAP.
- It is based on standard SQL but includes many extra functions to allow the developer to include conditional flow control logic such as If, Then, Else and While.

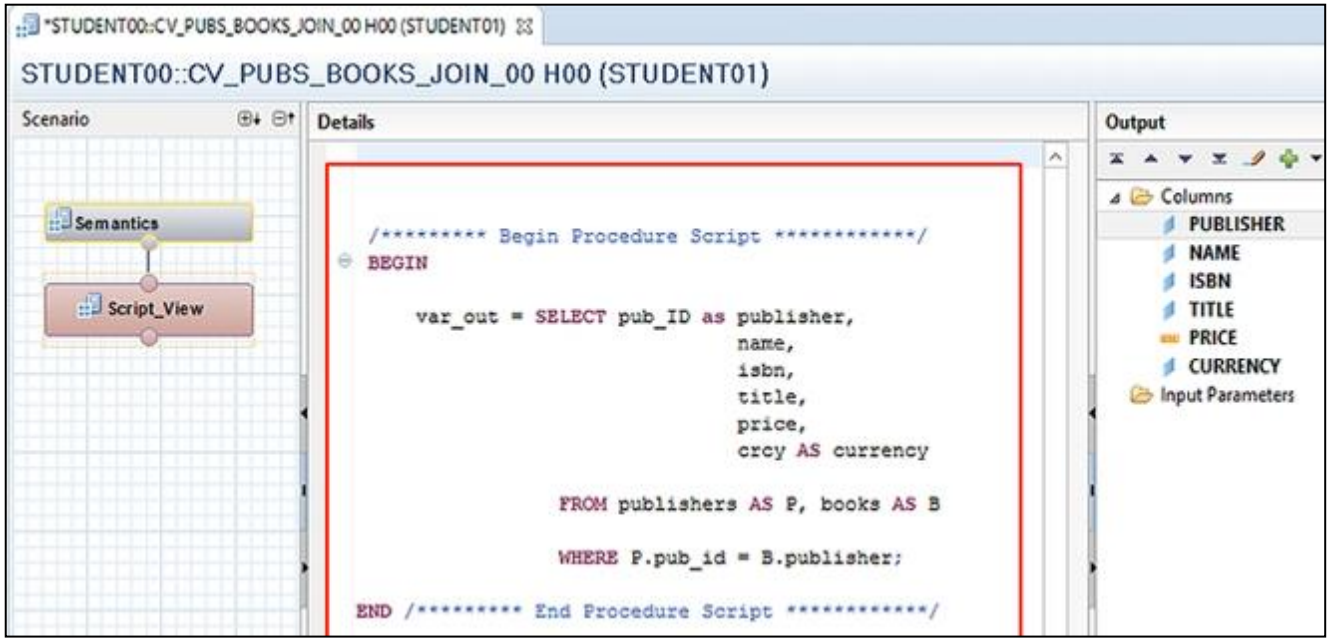


Calculation View (Scripted)

- To get started simply choose a calculation view but remember to select the subtype as SQL Script and not Graphical.
- Script based Calculation Views can be used when the graphical view types(DIMENSION and CUBE) do not fulfill the reporting requirement.
- In particular, they are useful when you need to apply a complex logic that is not supported by graphical information views. For example, when you want to use syntax such as conditions(IF…THEN…ELSE), loops (FOR or WHILE), and so on.

Writing the Script for Scripted Calculation View

- After selecting SQL Script as shown in the last slide, you would then define the output columns.
- Since SPS11 you can also have the system define the columns automatically by referring to an existing table or view.
- To write the script you must select the Script View node in the scenario pane.



Writing the Script for Scripted Calculation View

- The script editor now appears in the centre of the screen.
- Here you describe what you would like to happen using the language SQLScript.
- Note : It should be noted that from SPS9 onwards, SAP recommends that instead of creating scripted calculation views you should consider creating an SQL table functions.

Demo

Creation of Scripted Calculation View.

Querying Calculation Views in SAP HANA Studio

When building an Information View, SAP HANA Studio provides two ways to query the data of the view. It is important to understand between these approaches.

- i. Standard Data Preview
  - With the Standard Data Preview, you select all the columns that are included in the Semantics of the information view (provided that they are not hidden).
  - You can move the columns (drag and drop), and also order the result set by one (and only one) column.
- ii. Custom SQL Query
  - An alternative to the Standard Data Preview is to execute a Custom SQL Query. As shown in the figure, after generating the SQL Statement equivalent to the data preview, you can modify it, and for example change the selected columns or the group by clause, order the result set by several columns.

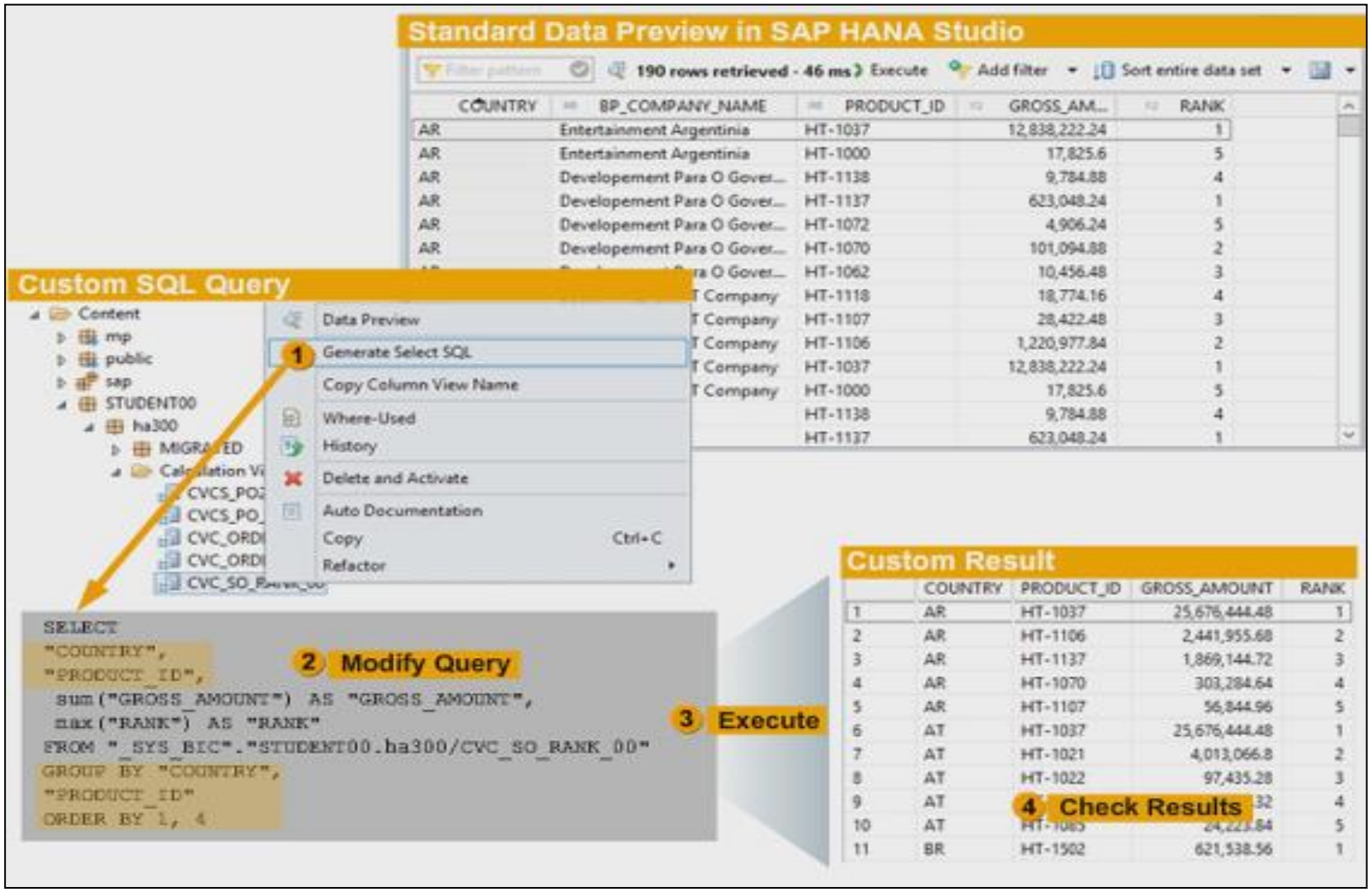
Querying Calculation Views in SAP HANA Studio

This is particularly useful to perform a thorough test of Calculation Views with a complex scenario like:

- Stacked calculation views,
- Counter measures,
- Join between several aggregation nodes with different GROUP BY columns.

Note: Indeed, Calculation Views behave differently depending on which columns are selected, on whether you explicitly define a group by or not. You must ensure that a View does not give wrong results if it is not correctly queried upon, or document the view so that it is correctly consumed by end-users with reporting applications.

Querying Calculation Views in SAP HANA Studio

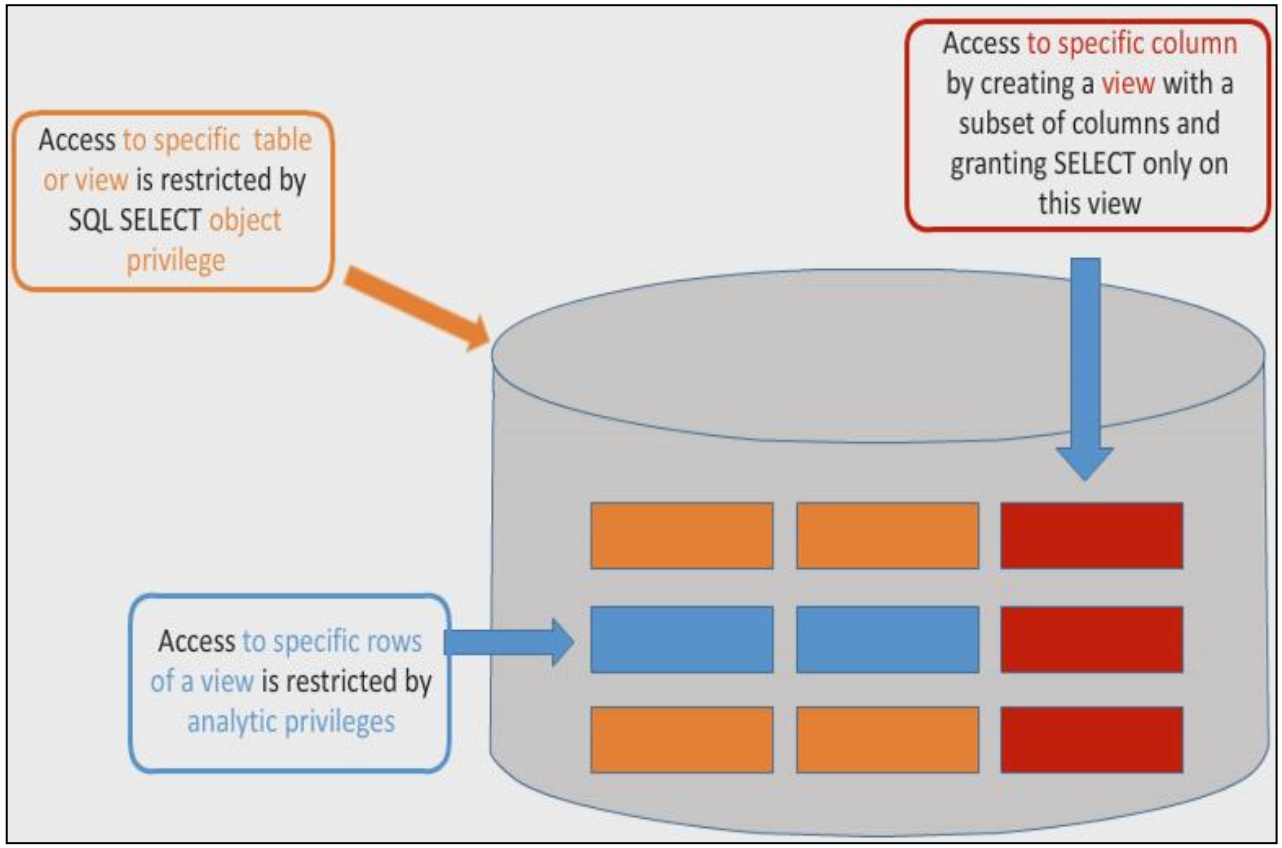


Demo

Creation of Calculation view in SAP HANA Studio and query it. Also modify the query and execute to illustrate other options too.

Security Consideration for Modelers

- Analytic Privilege
- Object Privilege
- Select Privilege



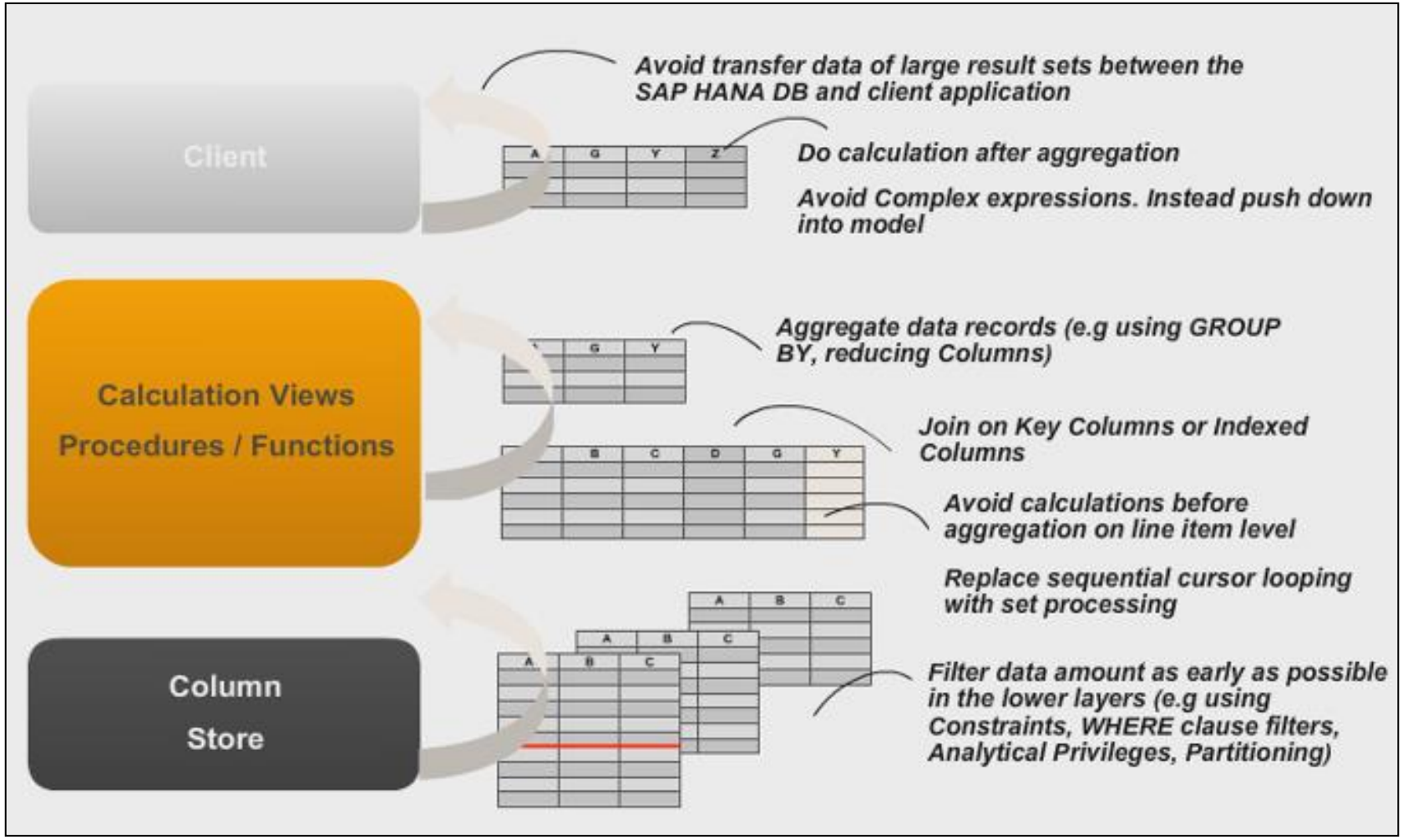
Security Consideration for Modelers

- Unless you grant access to users, they will see no data from your information views.
- A modeler needs to be aware of the different levels of security.
- The first level of access that is required is to the actual information view and also the source tables that are included in the view. This is achieved by granting a SELECT privilege on the view of table to the user or the role to which the user is assigned.
- Just because a user has access to the view does not mean they will see any data. Now they need to have access at the row level. This is achieved by defining an analytic privilege and then assigning this to the user or the role to which the user is assigned.
- Access to specific table or view is restricted by SQL SELECT object privilege.

Demo

Demonstrate by example how SELECT, Object and Analytic privileges can be applied.

Best Practices for Data Modeling

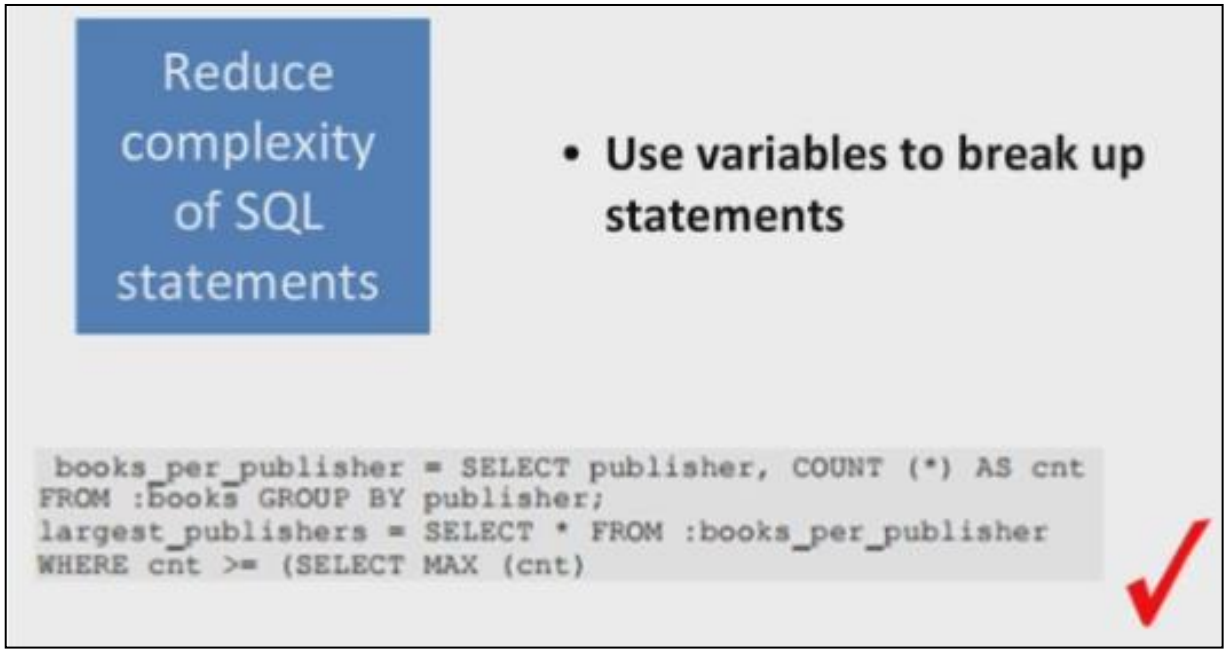


Best Practices when Writing SQL

Reduce complexity of SQL statements by using variables.

Variables are used to capture the results of each SQL statement and can then be used as the inputs for subsequent steps.

This makes understanding the code much easier and does not sacrifice performance.



Demo

Show an example for Best Practices for Data Modeling

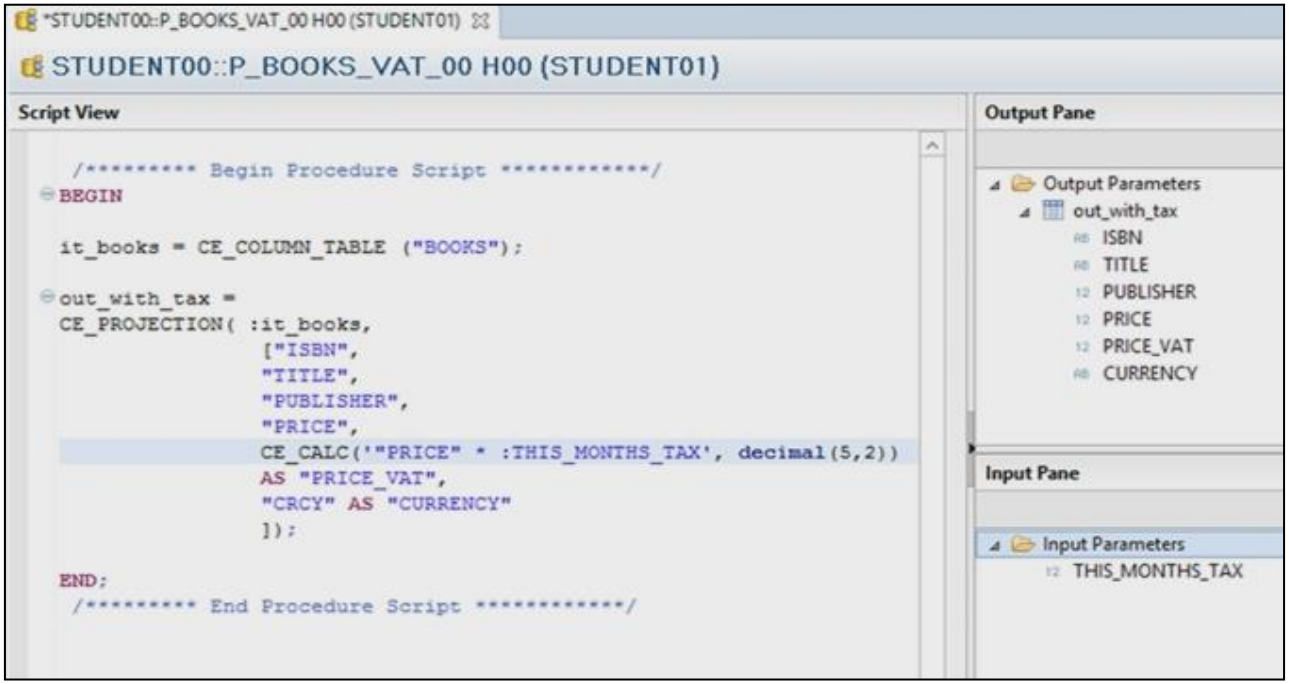
Best Practices when Writing SQL

Procedures

Procedures define reusable data processing functions.

Here you simply define a procedure and call this from the calculation view.

A procedure always has one or more output parameters but a procedure can also have one or more input parameters.



Procedures

For example, you could create a procedure that calculates the tax for each item sold. We would simply define the input parameter as a tax percentage within the procedure, and then define the output parameters as the columns we want to generate including the new column for calculated tax.

The actual procedure is written in SQLScript and reads the input parameters and then writes to the output parameters.

Procedures can be called from within calculation views or even standalone via SQL.

Procedures used within modeling are mostly used as read only.

In that case they are called stateless as they don't alter any data in the database. But procedures can also be used to update, insert and delete data. These are called stateful, but these type of procedures are not allowed when called from calculation views.

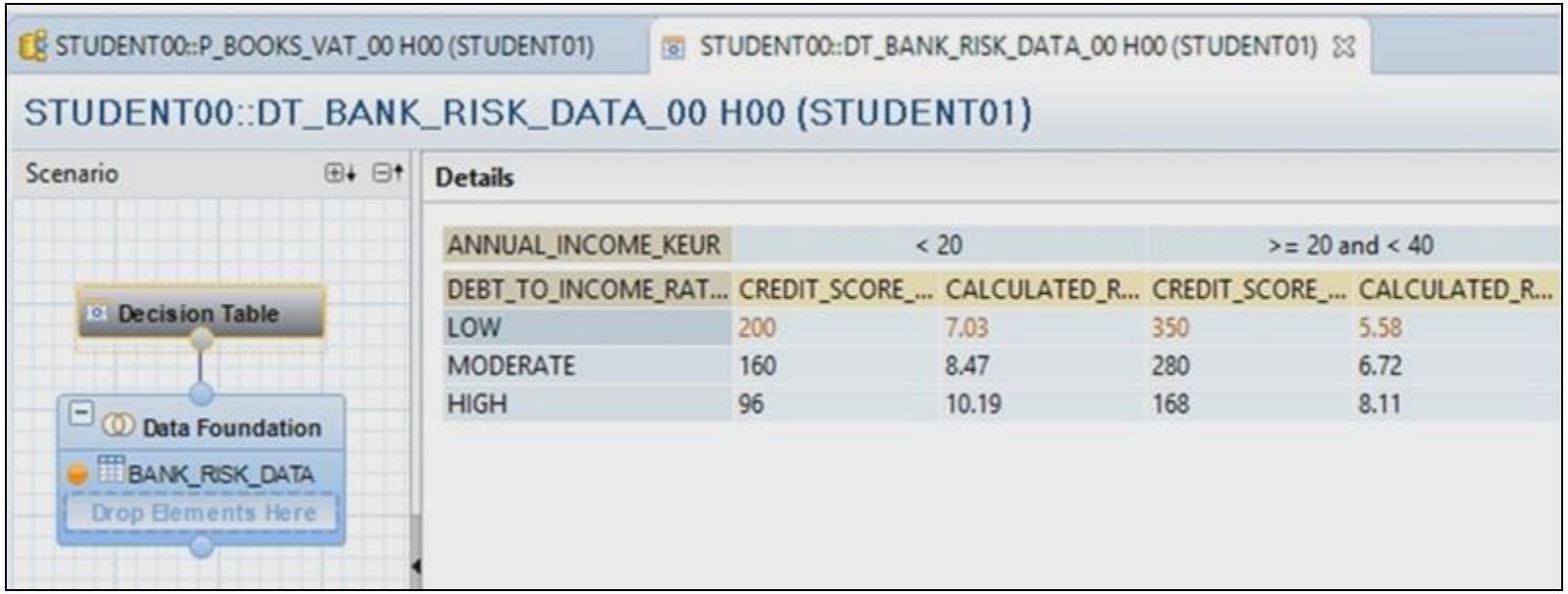
Stateful procedures are more likely to be used by developers who build applications.

Decision Table

A decision table can be used to store business rules that can be called by any calculation view or procedure.

Modelers and developers frequently use decision tables to ensure they build the most efficient objects.

Decision tables are useful when you find yourself developing calculation views or procedures where the business rules become complex and/or you know you will need to use the same rules again in other views.



Decision Table

The benefit of using decision tables is that you do not have to lock in any business rules into a calculation view or procedure that might need to be changed often.

For example, the target sales for each sales region. Imagine if these numbers changed each month and you had built multiple calculation views which included a calculated column that uses an If / Then /Else expression with hard coded sales percentages.

This hard coded condition would need to be maintained in each calculation view.

So a decision table can be used to store all business rules in a central place.

Maintenance of the rules is done in one place and the interface is very simple.

In fact you can even maintain and upload the rules from an Excel spreadsheet.

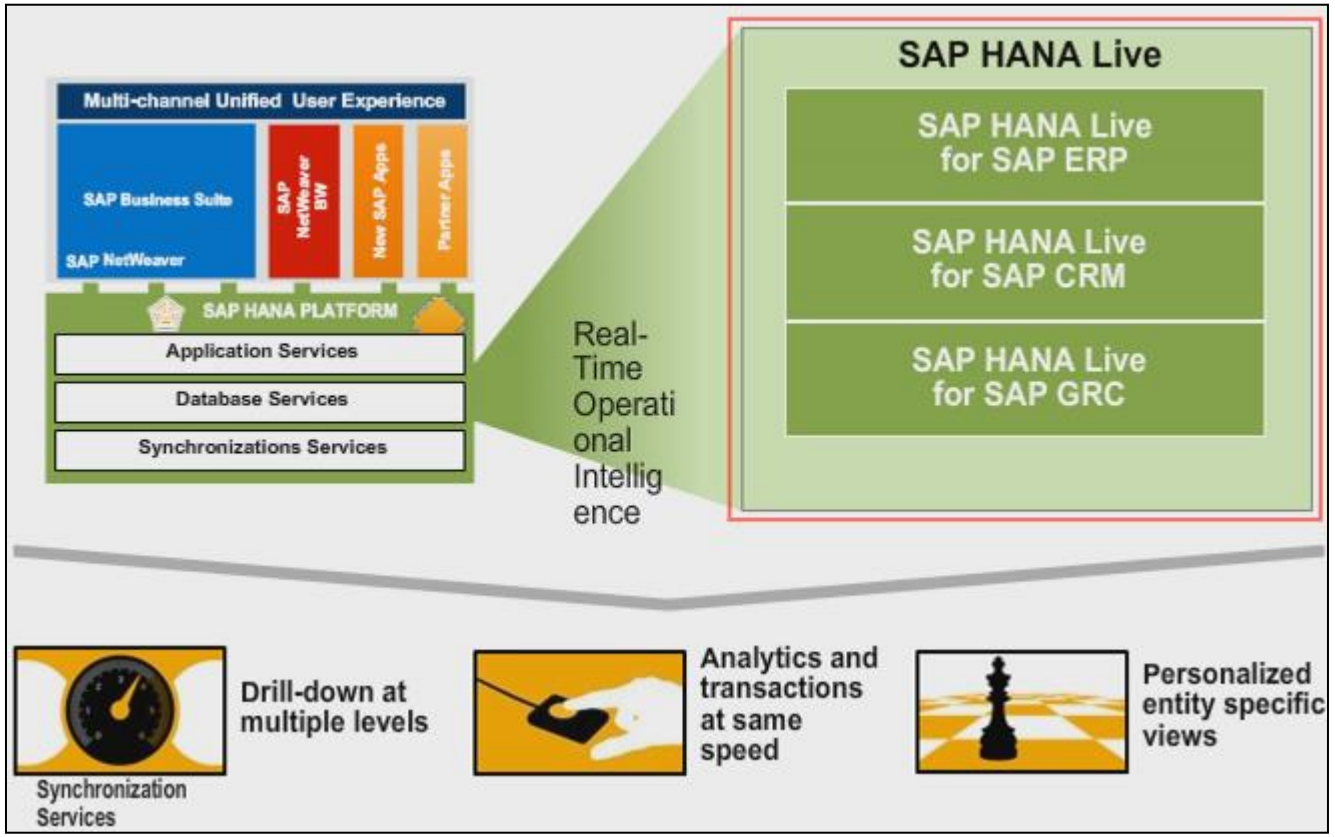
Demo

Show example for procedures and decision tables.

SAP HANA Live

SAP HANA Live is a comprehensive set of predefined SAP HANA information views based on SAP Business Suite tables.

SAP HANA Live is extremely important for SAP Business Suite.



SAP HANA Live

Due to the high degree of normalization of the database schema found in SAP Business Suite applications, tables can appear complex, fragmented and are often difficult to consume without a thorough understanding of each schema.

SAP HANA Live provides ready to use business views over the source tables so that report developers and application builders can easily consume business information in real time without being concerned about the underlying tables.

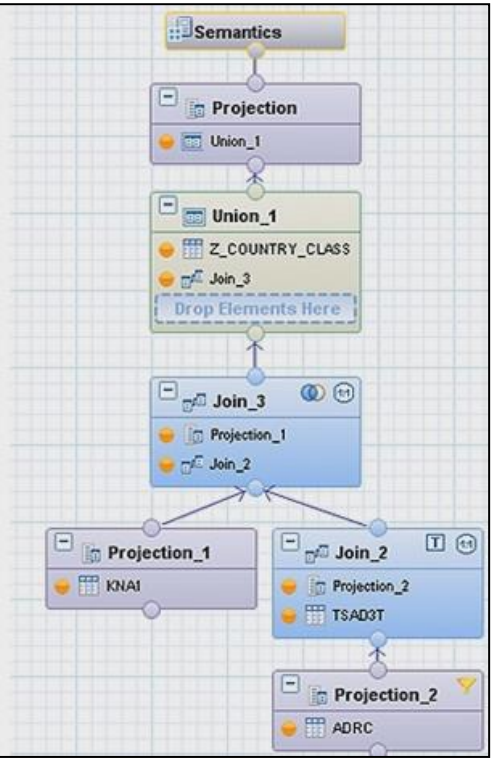
The information views are created by SAP but can be modified by customers using standard SAP HANA modeling tools.

Key Aspects of SAP HANA Live

SAP HANA Live models contain all necessary joins, filters, aggregations and transformations to turn the data in the raw Suite tables into meaningful information.

SAP HANA Live models are compiled into column views

- 100% use of standard HANA modelling techniques
- Built entirely with HANA calculation views
- Uses joins to combine re-use views to a consumption query view
- Adds calculated attributes along the way
- Adds input parameters and variables too
- Mostly graphical type calculation models



Key Aspects of SAP HANA Live

SAP HANA Live views can be consumed by any SAP Business Objects reporting tool, or by any custom development where a column view is required.

Most SAP HANA Live tools are integrated into SAP HANA studio; however, some SAP HANA Live tools are standalone.

SAP HANA Live has its own release cycle so check the versions available and especially to evaluate which version will work with your SAP Business Suite and SAP HANA solution combination.

SAP HANA Live Deployment Scenarios

Ideally, all operational data should be replicated in real time, hence the name SAP HANA Live.

With the integrated deployment scenario, SAP HANA live is installed inside SAP HANA that is powering the suite application.

This means the tables used by the suite application are also now shared by the SAP HANA live views.

No data movement takes place, there is zero redundancy and this means that the SAP HANA live views always reflect real- time information.

Side-by-side deployment requires the implementation of a data loading tool to move data from the tables in the source Suite application to SAP HANA. SAP Landscape Transformation Replication Server (SLT) is recommended, but other data provisioning tools can be used.

SAP HANA Live Deployment Scenarios

The key objective is to ensure operational tables are replicated to SAP HANA.

One of the most powerful tools used to explore SAP HANA Live models is the SAP HANA Live Browser.

The SAP HANA Live Browser is able to generate a list of tables required to support any SAP HANA Live model. One of the major benefits of using SLT for table replication is that it can consume the list of tables produced by SAP HANA Live Browser.

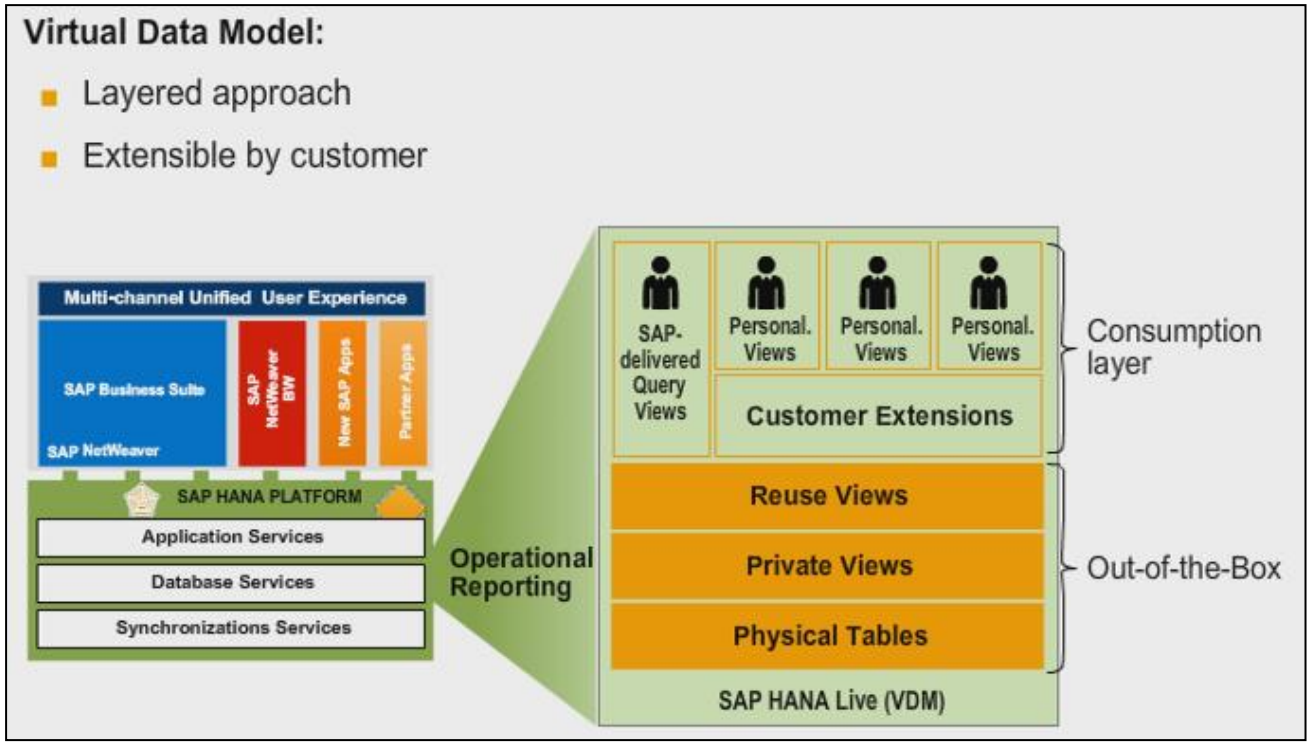
This makes setting up replication much easier.

The alternative is to enter all tables manually.

Virtual Data Model

SAP HANA Live relies on a Virtual Data Model (VDM).

The Virtual Data Model is a structured representation of HANA database views used in the SAP HANA Live for SAP Business Suite following consistent modeling rules.



Virtual Data Model

SAP HANA Live is built in layers to promote a high degree of reuse.

The lower layers combine data, effectively de-normalizing the ERP model.

These lower layers must not be changed as they have been carefully created by SAP to provide a business view of the data using all required tables and relevant joins.

The highest layer (consumption) can be changed by customers, in fact the highest layer supplied by SAP is really an optional quick start layer so customer can use SAP HANA Live while they consider customizations.

View in the Virtual Data Model

Query Views:

- These are designed for direct consumption by an analytical application(for example based on HTML5) or a generic analytical tool(for example SAP BusinessObjects tools).

Reuse Views:

- These are designed for reuse by other views and must not be consumed directly by analytical tools.

Private Views:

- Private views encapsulate certain SQL transformations on one or several database tables or even other views.
- A Private view may be based on database tables, other private views or on reuse views.

Opportunities for New Applications

Business Applications

- Transactional data
- Master data
- Analytical data

Geographic Information Systems (GIS)

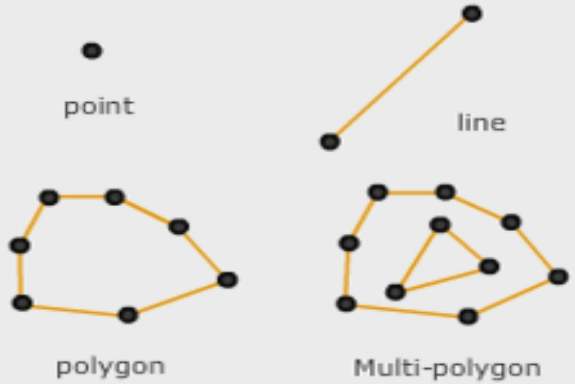
- Geographical data
- Location-based data
- Maps and topologies

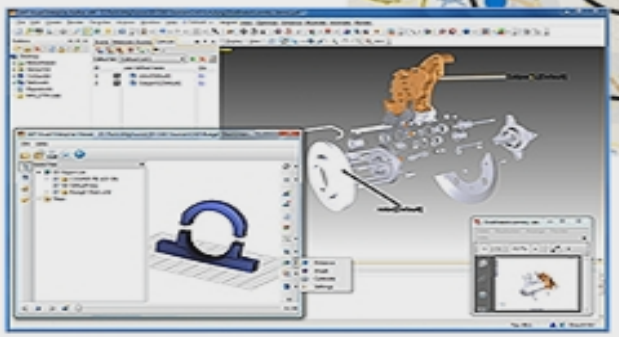
Engineering Systems Diagrams

- 2D/3D graphs
- Animations

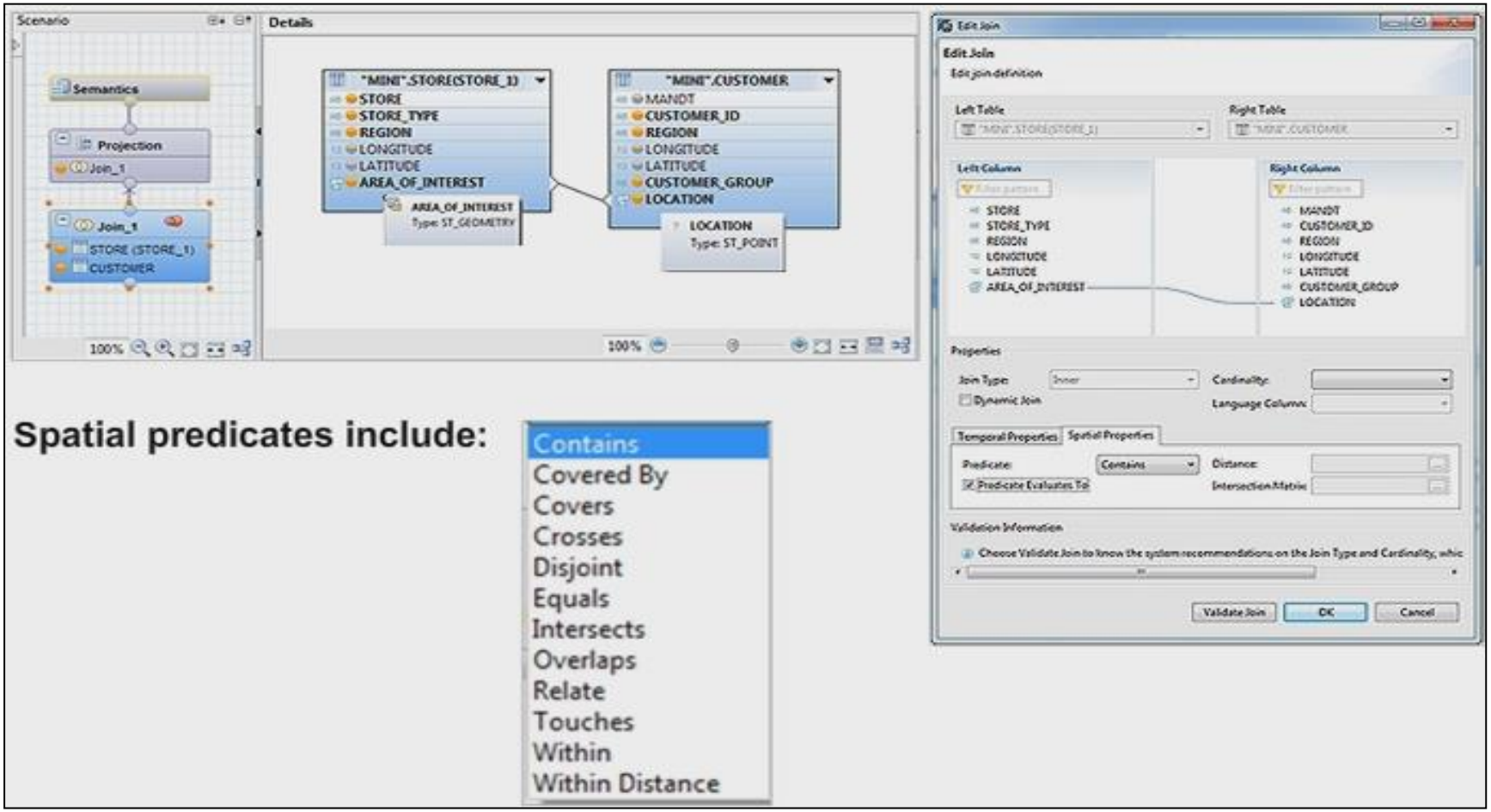
SAP HANA Spatial Processing

- **Store, process, manipulate, share, and retrieve spatial data directly in the database**
- **Process spatial vector data with spatial analytic functions:**
  - Measurements – distance, surface, area, perimeter, volume
  - Relationships – intersects, contains, within, adjacent, touches
  - Operators – buffer, transform
  - Attributes – types, number of points
- **Store and transform various 2D coordinate systems**
- **Process vector data**
- **Implements the ISO/IEC 13249-3 standard and Open Geospatial Consortium (1999 SQL/MM standard)**





Calculation View- Spatial Operation



Demo

Example to extend an existing HANA Live query view.

Demonstrate Calculation View- Spatial Operation.