SAP HANA Modelling Guide

1 SAP HANA MODELING DOCUMENTATION

1.1 Attribute Views

1.2 Analytic Views

1.3 Calculation views

1.6 Pre-Requisites for Creating Information Views

1.6.2 Roles to User

1.6.2.1 Creating Package:

1.6.3.1 Creating Database schema:

1.6.4 Privileges to _SYS_REPO

1.7 Creating Views

1.7.1 Creating Attribute view:

1.7.2 Creating Analytic View:

1.7.3 Creating calculation view:

1.7.4 Variables and Input Parameters in HANA:

1.8 Import/Export of Models

1.8.1 Developer mode

1.8.2 Delivery Unit

1.8.3 SAP Support Mode

1.9 SAP HANA Importing and Exporting Landscapes

1.11 Common Support Issues

1.11.1 Authorization Issues

1.11.2 Import or Export Issues

1.11.3 Schema Mapping

1.11.4 Information model Data Preview issues/Activation Fail

1.11.5 Studio Version Issues:

1.11.7 Performance Issues

1 SAP HANA MODELING DOCUMENTATION

SAP HANA Information Modeling or SAP HANA Data Modeling is used for HANA application development. Modeling views are created on top of database tables and implement business logic to create a meaningful report. These Information models are used to build reports in LUMRIA, SBEE, Webi etc. and can also be used for fulfilling business logic in Applications(XS based Apps,ABAP based applications..)

There are 3 types of views.
1) Attribute Views
2) Analytic Views
3) Calculation Views

1.1 Attribute Views

Attribute views are mainly used to model master data entities (like Product, Employee, Business Partner)

Attribute views are built by joining database tables.

Attribute views are used in Analytic and Calculation Views.

Attributes views are executed in Join Engine.

If there are massive Joins expected in the Model better to use Attribute Views for achieving better performance.

1.2 Analytic Views

Analytic views are like star schemas or fact tables surrounded by dimensions.

Analytic views use tables in data foundation and Attribute views in STAR join.

Calculated column can also be created in Analytic views.
Tables cannot be directly used in the star join of the Analytic view. Attribute views need to be built on the tables which are needed to be used in the Analytic view star join node.

### 1.3 Calculation views

Calculation views are required to deliver complex business requirement. Calculation views are simple and yet powerful because they mirror the functionality found in both attribute views and analytic views.

It can perform complex calculations not possible with other views.

Calculation views can be created as Graphical or scripted views depending on the business requirement.

Graphical views can be modeled using the graphical modeling features of the SAP HANA Modeler. Scripted views are created as sequences of SQL statements.

The owner of the calculation view created graphically is _SYS_REPO user (More discussion about _SYS_REPO user in the later part of the document)

If the calculation view is created through script then the owner of the view will be the user creating the calculation view. The privileges on this view to any other users including the SYSTEM user should be given by this user.

Calculation views can be referred as combination of tables, column views, attributes views and analytical views to deliver a complex business requirement. They offer to combine different analytical views into one source of data for reporting.

Calculation views consist of Join, Union, Projection and Aggregation node.

Based on the requirement these different nodes can be used in the calculation view during modelling.

### 1.4 Types of Engines used in Views

SAP HANA has 3 types of engines that are used based on the views required by the model.

Join Engine:
- Attribute View uses Join engine.
- SQL queries also use Join engine.

OLAP Engine:
- Used for Analytic view (without calculated columns). If Analytic views are based on attribute views, the joins in the attribute views get flattened and included in the analytic view run time object, so only the OLAP engine will be used for Analytic views.

Calculation Engine:
- Used for Analytic views with calculated attributes and Calculation views

Note: To check what engine is used to process a query: simply run the Plan Visualization on your query.

**Generate Plan Viz from Query**

Plan Viz is used to Analyze how the query execution is taking place in HANA (engines used, operators, Number of column queried from tables..) Select the query and right click and select explain plan viz.

Steps to Generate Planviz:
1) In the SQL console you can right click on the statement and select Planviz execution
The Generated Plan gives Information in 2 tabs

The Overview & executed Plan

Overview Page:

In this Page it provides Information about Operators used, number of tables used, memory Allocated

Executed Plan:

It will give the actual execution plan.
In the Screenshot Orange line is the Critical path, if we can optimize the Orange line nodes Query performance can be improved.

It also gives Actual records and expected records processed from a node.

Expected records are shown in the brackets.

This executed Planviz can be saved as *.plv file and can be send as an attachment to Support teams for further Analysis in case of any issue.

The *.plv can be opened in HANA studio.

### 1.5 How to Create Information Views

#### 1.6 Pre-Requisites for Creating Information Views

1. User in SAP HANA system
2. Roles to the User (Modeler role)
3. Data source (Schema where Data resides)
4. Privileges to _SYS_REPO user

#### 1.6.1 Users:
A user should be created or with any existing user already created we can log into the system.

The CREATE USER statement creates a new database user.

Only database users having the system privilege USER ADMIN are allowed to create another database user.

The specified user name must not be identical to the name of an existing user, role, or schema.

There are some users that are delivered with the SAP HANA database: SYS, SYSTEM, and a number of users with _SYS_ prefix.

Whenever a user is created, a schema is created with the same name.

For example when user HANA_TUTORIAL is created then HANA_TUTORIAL schema is automatically created along with the user. HANA_TUTORIAL user is the owner of HANA_TUTORIAL schema.

SYSTEM user has no rights to access the schema content unless until the schema owner HANA_TUTORIAL user, grants any privileges on the HANA_TUTORIAL schema.

There are 2 types of USERS
1) Restricted User
2) User

Users created without the option RESTRICTED are allowed to create any object in their own schema and they are granted the role PUBLIC allowing them to select any system view. Users created with the option RESTRICTED have no privileges, not even the allowance to create objects in their own schema. These users are meant for using specific applications only. All privileges needed for the application have to be granted to such users, preferably using a role combined with the application.

User can be created graphically or through the SQL statement

CREATE USER <username> PASSWORD <PASSWORD>

1.6.2 Roles to User

1.6.2.1 Creating Package:

Creation of views is done in the content folder.

To access the content folder REPOSITORY_REST object privilege is needed or else below error is encountered.
REPOSITORY_REST object privilege is must to create information views.

To access all the packages in the system root package privilege is necessary.

To create new package, right click the content folder -> new -> package.

Grant privileges on repository packages to authorize user to modify objects.

For more information on package privileges please look into below links:

Package Privilege Options
Package Privileges

After a package is created information views can be created in the package.

1.6.3 Data source (Schema where Data resides)

Data into Schema can come from either Applications or via Data provisioning Methods.

1.6.3.1 Creating Database schema:

If the user want to create a new schema, then system privilege CREATE SCHEMA is required.

Database schema can be created using the below SQL statement.

Create schema <schema_name>;

GRANT Statement (Access Control)

Link contains more information about System privileges.
Here MASTERDATA schema is created with user HANA_TUTORIAL. So HANA_TUTORIAL is the owner of MASTERDATA schema.

```sql
create schema MASTERDATA;
```

Once schema is created then tables and views can be created in this schema. Here I am creating PRODUCT, REGION and SALES tables.

We can use SQL script to create the tables or CDS tables. Generally Table should be created using CDS format, using WebIDE or in development perspective in Hana studio. This helps in transport of the tables between different landscapes.

A sample tutorial can be found at the below link on how to create CDS tables:

Get Started with CDS

If we use SQL script it is not a good idea as we do not create anything new in Production system.

Here I am creating tables using SQL script in SQL studio as we do not need to move the tables to a different system.

Script to create tables in SQL studio.

-- Create Product table

```sql
create column table "MASTERDATA"."PRODUCT"

    "PRODUCT_ID" INTEGER null,

    "PRODUCT_NAME" VARCHAR (100) null default "

);

insert into "MASTERDATA"."PRODUCT" values(1,'Shirts');
insert into "MASTERDATA"."PRODUCT" values(2,'Jackets');
insert into "MASTERDATA"."PRODUCT" values(3,'Trousers');
```
insert into "MASTERDATA"."PRODUCT" values(4,'Coats');
insert into "MASTERDATA"."PRODUCT" values(5,'Purse');

-- Create Region table
create column table "MASTERDATA"."REGION"(
    "REGION_ID" INTEGER null,
    "REGION_NAME" VARCHAR (100) null default '',
    "SUB_REGION_NAME" VARCHAR (100) null default ''
);

insert into "MASTERDATA"."REGION" values(1,'Americas','North-America');
insert into "MASTERDATA"."REGION" values(2,'Americas','South-America');
insert into "MASTERDATA"."REGION" values(3,'Asia','India');
insert into "MASTERDATA"."REGION" values(4,'Asia','Japan');
insert into "MASTERDATA"."REGION" values(5,'Europe','Germany');

-- Create Sales table
create column table "MASTERDATA"."SALES"(
    "REGION_ID" INTEGER null,
    "PRODUCT_ID" INTEGER null,
    "SALES_AMOUNT" DOUBLE null);

insert into "MASTERDATA"."SALES" values(1,1,100);
insert into "MASTERDATA"."SALES" values(1,2,90);
insert into "MASTERDATA"."SALES" values(1,5,85);
insert into "MASTERDATA"."SALES" values(2,2,80);
insert into "MASTERDATA"."SALES" values(2,1,75);
insert into "MASTERDATA"."SALES" values(3,3,85);
insert into "MASTERDATA"."SALES" values(4,4,75);
insert into "MASTERDATA"."SALES" values(5,1,65);
insert into "MASTERDATA"."SALES" values(5,2,65);

PRODUCT, REGION and SALES tables are created after executing the above script in MASTERDATA schema.
1.6.4 Privileges to _SYS_REPO

The activation of modeling views are done in the name of user _SYS_REPO. 
_SYS_REPO user activates the models and creates the necessary runtime objects from them. Therefore _SYS_REPO user needs the allowance 
to select YOUR tables/views. (If _SYS_REPO user does not have select on the tables specified in the from-clause of the view-definition, it cannot 
define that view)

If other users need to select this view (obviously this is always the case, otherwise the views would not make sense), then _SYS_REPO needs to 
have the additional allowance to grant the select further (WITH GRANT OPTION).

Therefore after having activated all your models to access data in your schemas, _SYS_REPO wants to give you (and probably other users) read 
access to the activated models.

The user who is the owner of the schema, or the any other user who has select with grantable to others on the schema can give access on the 
schema to _SYS_REPO.

Apart from this the user need to have USER ADMIN System privilege to change the user.

1.7 Creating Views

1.7.1 Creating Attribute view:

To create an attribute view, right click the package->new->attribute view
Be sure to not forget granting SELECT privilege WITH GRANT option to _SYS_REPO user on the schema or objects used in Information views.

```sql
grant select on schema <SCHEMA> to _SYS_REPO with grant option
```

If the _SYS_REPO user is not given the SELECT privilege WITH GRANT option object privilege on the objects used in the creation of the Information views, the activation fails with below error:

```
Creating Analytic View:
```

To create the Analytic view right click the package->new->analytic view.
Analytic view is nothing but a star schema.

The transaction table is added to the data foundation and the dimensions are joined to the Transaction table.

Generally, the cardinality between the transaction table and the dimension tables is n to 1.

To add the tables to the data foundation, click on the plus sign.

In the star join node, only the Analytic view can add attribute views. We cannot add directly tables to the star join.

Always there should be at least one measure in the Analytic view. Or else the activation fails.

You should always select the type of the column whether attribute or measure.

If the type is not specified, then during activation, the system automatically assigns attribute to columns of type varchar or nvarchar and measure to columns of data type integer.

1.7.3 Creating calculation view:

To create the calculation view, select the package -> new -> calculation view.
Calculation view can be created as CUBE or Dimension. When the calculation view is created as CUBE you can select the final node before the semantics to be aggregation or star join.

Star join calculation almost same like Analytic view.

The calculation view is created as CUBE when the data need to be aggregated. When the data do not need to be aggregated the calculation view need to be created as Dimension.

Whenever a calculation view is created as a CUBE then there must be a measure defined, Or else the activation fails.

Restricted column cannot be created if the Calculation view is created as Dimension.

After creation of models to do data preview or run any select statements on the view, the user need to have select privilege on the run time view.

For example if the user need to do data preview on CL_REGION view, then the user should have select privilege on 
"_SYS_BIC":"hanatutorial/CL_REGION"
1.7.4 Variables and Input Parameters in HANA:

Sometimes we need to filter the data based on user's input. This is where Input Parameters and Variables come into the pictures. Input parameters and variables are used to filter the result of modeling views.

**Variables:**

Variables are **bound to columns** and are **used for filtering** using WHERE clauses. As such, they can only contain the values available in the Columns they relate to.

**Important features of variables:**

- You use variables to filter data at runtime. You assign values to these variables by entering the value manually, or by selecting it from the drop-down list.
- You can also define whether the Variable is Mandatory or if it should have a Default Value.
- You can apply variables only in analytic and calculation views.

**Input Parameters:**

Sometimes you might not want a variable to just restrict the data of a view. But you also want to take input from the user and process it, returning dynamic data based on the user selection. Input Parameters makes this possible.

**Important Features of Input Parameters:**

- Input Parameters can contain any value the reporting user has to enter to parameterize the result. Therefore, a data type must be specified for each Input Parameter.
- Input Parameters are passed by Placeholders and used in Formulas.

**Input Parameter Types:**

The following types of Input variables are supported.

**Currency:**
Use this during currency conversion where the end user should specify a source or target currency.

**Date:**
Use this to retrieve a date from the end user using a calendar type input box.

**Static List:**
Use this when the end user should have a set list of values to choose from.

**Attribute Value:**
When an Input Variable has this type, it serves the same purpose as a normal Variable.

**None:**
If none of the above applies you do not have to specify an Input Variable type. The Type can be left blank.

For more information on how to create Variables and input parameters in Information models please look at below link.
1.8 Import/Export of Models

Models can be imported from one system to another systems in various Methods

1) Developer Mode
2) Delivery Unit
3) SAP Support Mode

The following SQL could be used to export information view:

EXPORT "_SYS_BIC"."<ViewName>" AS binary into '</path/to/export>);

If the issue is not data related or customer doesn’t allow the data to be exported, you may use the following SQL:

EXPORT "_SYS_BIC"."<ViewName>" AS binary into '</path/to/export>' with catalog only;

Then, the export will be saved at linux server under '/path/to/export/index' folder. If the system is a multi-node system, make sure the '/path/to/export/index' folder is on a shared file system, which all the hana nodes has access to. Compress that folder and send it to support.

For Support Purposes we can request customer to Export the Models in HANA Support Mode

This Mode exports HANA Views along with Dependents (required Tables...), so that we can replicate the issues on our end easily.

More details can be found in the document.

Import and Export in Hana Studio

Import and export can be done using developer mode, SAP support mode or through Delivery Unit.

HANA Export and Import option allows tables, Information models, Landscapes to move to a different or existing system. You do not need to recreate all tables and information models as you can simply export it to new system or import to an existing target system to reduce the effort.

This option can be accessed from File menu at the top or by right clicking on any table in HANA studio.

1.8.1 Developer mode

In Developer mode export and import, objects need to be activated manually after import and when we are importing multiple objects, activation should be done in a sequence.

Developer mode export:

Select the system for the objects need to be exported. Select the objects to be exported and select the target location and click on finish.

Developer Mode Import:

Select File ->import->SAP HANA Content ->Developer Mode ->select the target system into which the objects need to be imported.

Use case: Developer mode is used, when only few information models need to be exported and imported. The activation of the models need to be done manually.

1.8.2 Delivery Unit

A single Unit which can be mapped to multiple packages and can be exported as single entity. So that all the packages assigned to Delivery Unit can be treated as single unit. The user can use this option to export all the packages that make a delivery unit and the relevant objects contained in it to a HANA Server or to local Client location.

Use case: Delivery unit export and import is used when multiple packages need to be exported and imported

1.8.3 SAP Support Mode

This can be used to export the objects along with the data for SAP support purposes. This can be used when requested. For Eg, user creates a view which throws up error and he couldn’t able to resolve. In that case he can use this option to export the view along with data and share it with
Use case: SAP Support Mode is used when the dependent objects like tables need to be exported when exporting the information models.

For more information on SAP HANA Studio and SAP HANA content export and import please look into below link:

Export and Import feature in HANA Studio

1.9 SAP HANA Importing and Exporting Landscapes

We can add multiple Hana instances in Hana studio. When we upgrade the studio or install a new studio version, instead of adding all the instances we can export the landscape and import the landscape to the newly installed studio.

For more information on Exporting and Importing landscapes in Hana studio please look into the below link.

Importing and exporting Landscapes in HANA Studio

1.10 Analytic Privileges:

Analytic privileges are intended to control read-only access to SAP HANA information models (attribute views, analytic views, and calculation views)

For more information on Analytic privileges, please look into below link:

Analytic Privileges

Example on Analytic Privilege can be found at:

Using Analytic Privileges

From SP10 onwards, a new way of creating Analytic Privilege has been introduced called SQL Analytic Privilege

Standard SQL-based privileges

In addition to the existing approach of creating classical analytic privileges, users can now create SQL based analytic privileges which provides more flexibility as you can use the functionalities available within the SQL environment.

First the authorization relevant view has to be marked that a SQL privileges should apply:
Afterwards the SQL-based privilege has to be created:

To finalize the Analytic Privilege (AP) you have to add the view which will be secured, switch to the SQL Editor and afterwards write your SQL coding into the SQL Editor. In the past this logic was placed within a procedure. You still have the possibility to use this functionality while switching to mode "Dynamic":

For more details on SQL Analytic Privilege, check the link below:

Analytic Privilege

1.11 Common Support Issues

Common Support Issues encountered by customers in Modelling area are:

1) Authorization Issues
2) Import/Export Issues
3) Schema Mapping
4) View Data Preview issues
5) Studio Version Issues
6) Performance Issues
7) Slow Activation of Objects

### 1.11.1 Authorization Issues

Generally if any authorizations are missing, then the view activation fail with insufficient privilege error.

To debug any authorization issues, User specific trace for authorization must be configured.

To activate an Authorization Trace in case of authorization Problems please look at below links:

**How to setup HANA authorization trace?**

- 2370781 - How to set the Database trace for Authorization to 'info'

To configure traces, you must have the system privilege TRACE ADMIN

To avoid authorization issues some rule of thumb

1) **_SYS_REPO** user must have SELECT with Grant permission, Execute to all the objects used to build information models.

2) This is because **_SYS_REPO** is the user who is activating all the information models.

3) The user doing the data preview or Writing any select queries on the Information models must have select rights on the run time model **_SYS_BIC/<viewname>**

4) If the information model contains any input parameters or variables, then the user doing the data preview must have select rights on the **_SYS_BI** schema as well.

It's always Advised to create Objects in HANA as design time Objects (repository Objects) to avoid issues with Object ownership and authorization issues.

If an individual create Objects they own the Objects and when they leave the organization it will be a problem to cascade that Ownership.

The Below Video explains more on that.

**Object Ownership in SAP HANA Database**

### 1.11.2 Import or Export Issues

To do import or export the user needs to have Import Export system privilege.

Import or export issues can be debugged by enabling the user specific trace for import or export.

To enable the import or export trace for user to debug the import export issues, the following are the steps to enable the trace.

Go to Hana System Administration, Trace Configuration:

User Specific Trace, select New Configuration (small Icon ‘Create’ upper right Corner of User-Specific Trace)

Context Name is a description for this user-defined-trace

Select Indexserver, select ‘Show all components’, and select Import export

System Trace level value can be set to ‘INFO’ (error is the default)

Click on Finish

Now trace is active

Now switch to the relevant user and produce the error.

From the diagnosis files you can find the user specific indexserver trace file with below naming convention.
Please make sure to delete the trace configuration enabled after collecting the trace.

This trace gives more information on which objects the user is missing the rights.

1.11.3 Schema Mapping

Schema mapping is required when moving SAP HANA objects from one system to another system, say from Development system to Test System, or Test System to Production System.

Schema Mapping issues also occur when Importing HANA Live content.

Suppose we are moving components from Development System (DEV) to Production System (PROD). The tables in DEV reside in DEV_SCHEMA and the same tables reside in PROD system in PROD_SCHEMA schema. If an attribute view is transported from DEV to PROD, it will not work because the schema name is referenced in the definition of attribute view. In order for the promoted objects to work in PROD, schema mapping needs to be set up in the target system.

In this scenario, the schema mapping to be created is

<table>
<thead>
<tr>
<th>Authoring Schema</th>
<th>Physical Schema</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEV_SCHEMA</td>
<td>PROD_SCHEMA</td>
</tr>
</tbody>
</table>

Note:
- Schema mapping is only relevant for modeling views. Not procedures or script-based calculation views. You need to map the references of the script-based calculation views and procedures manually, that is, by changing the script if the tables are qualified with the schema.
- You can map several authoring schemas to the same physical schema. For example, content objects delivered by SAP refer to different authoring schemas, whereas in the customer system, all these authoring schemas are mapped to a single physical schema where the tables are replicated.
- The mapping between authoring and physical schemas is stored in the configuration table ".SYS_BI"."M_SCHEMA_MAPPING"

How to create schema Mapping:

1. In Modeler Perspective, go to Help in Menu bar and select Quick Menu.
2. Click on Schema Mapping.
3. Add Authoring schema name. For example SAP_ERP.
4. Add Physical schema name. For example DE8_DH1.
1.11.4 Information model Data Preview issues/Activation Fail

Activation fail:

Generally the activation of view fails, when the
SYS_REPO user miss any SELECT with grant privilege on any objects on the under lying tables
or database views used in the information models.

Data Preview:

The Data preview on the view fails when the user doing the data preview do not have select rights on the runtime
objects(_SYS_BIC/<viewname>)

In the above two cases, the user specific trace can be configured and we can debug the issue.

1.11.5 Studio Version Issues:

It is always good to use the compatible studio version for the database.

For example Differential and incremental backups are introduced from SP10.

So the corresponding Hana studio version compatible with SP10 must be used to see the incremental and differential option in Hana studio.

This option is not be available in SP9, as this is a new feature introduced with SP10.

So always it is good to use the compatible studio version for Hana database.

1.11.6 Analytic Privileges:

If the Analytic Privilege is not added to the User role, query execution fails with Insufficient Privilege error.

Once Analytic Privilege is set in the Information View Properties, Analytic Privilege must be applied to the users/roles.

Some of the common issues encountered are, if an Analytic privilege is applied on the Information Model, to restrict the data ,then this Analytic
privilege must be added to the User as well.

For Example, in the information model AN_V3HN_LEAD we are applying Analytic privilege on the sales org Attribute to restrict the data for the
User, So that the user sees only the data of the sales org's which he is supposed to see.

And this Analytic privilege must be applied on the View AN_V3HN
And the user who is doing the data preview on the information model (AN_V3HN_LEAD) for example in this case TEST_USER must also have the Analytic privilege AP_TEST_CLASSIC assigned to him.

Or else it will throw the error as below.
1.11.7 Performance Issues

As there is no single way to create a Model for a requirement, it's always better to check from the beginning of the Model to check whether the Model is behaving as expected.

Planviz is great a tool to check the model is working as Intended. Best way to deal with performance issues is after every step of the Modeling try to execute the Plan viz and check its behaving as expected.

Common check to see in the Planviz are:

1) Filters are pushdown? (check to see are the Filters are pushdown to the lowest tables)
2) Is Join simplification is working? (in the Model there might be several tables, if the current query is requesting Fields from only 3 tables in the Planviz only those tables should appear)
3) Is Appropriate Engines/operators used

Some of the best practices can be checked at the following link:

SAP HANA Modeling good practices

1.11.8 Activation slow issues:

Activation time of a view is dependent on the number of the places the view is reused.

For example to use Master Data, Attribute views are created. These Attribute views are used in many other views like Analytical and Calculation views where ever the Master data is required.

When ever there is any change to the Base Master data Attribute view, then during activation or redeploying the view, all the other views which are consuming the base Master data view will be checked for any changes.

If the Base Master data view is used in 50 other views, then all the 50 views will be checked for any changes during activation or redeployment of the base view.

In most of the cases the activation slow is due the use of the view in many different places.

Additional Resources:

SAP HANA Modeling Guide for SAP HANA Studio