Table of Contents

6. SAP HANA – INFORMATION MODELER

SAP HANA Information Modeler; also known as HANA Data Modeler is heart of HANA System. It enables to create modeling views at the top of database tables and implement business logic to create a meaningful report for analysis.

Features of Information Modeler

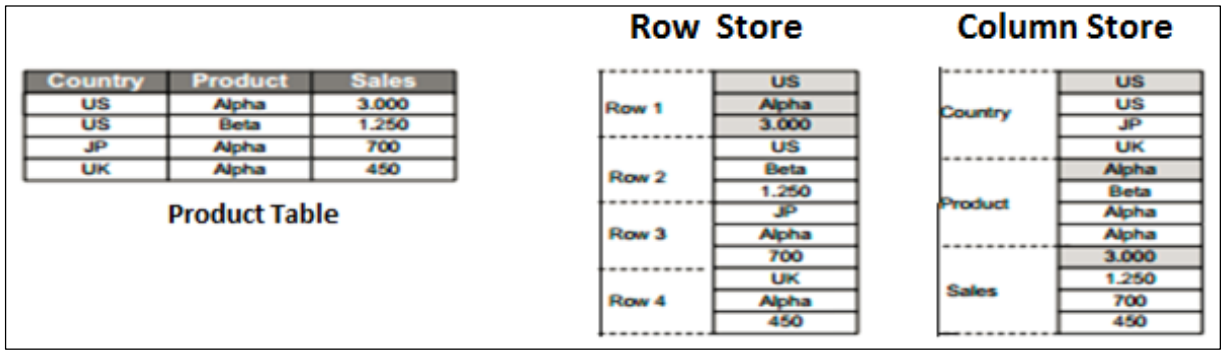
- Provides multiple views of transactional data stored in physical tables of HANA database for analysis and business logic purpose.
- Informational modeler only works for column based storage tables.
- Information Modeling Views are consumed by Java or HTML based applications or SAP tools like SAP Lumira or Analysis Office for reporting purpose.
- Also possible to use third party tools like MS Excel to connect to HANA and create reports.
- SAP HANA Modeling Views exploit real power of SAP HANA.

There are three types of Information Views, defined as:

- Attribute View
- Analytic View
- Calculation View

Row vs Column Store

SAP HANA Modeler Views can only be created on the top of Column based tables. Storing data in Column tables is not a new thing. Earlier it was assumed that storing data in Columnar based structure takes more memory size and not performance Optimized.



With evolution of SAP HANA, HANA used column based data storage in Information views and presented the real benefits of columnar tables over Row based tables.

PART 2: SAP HANA – MODELING

Snowflakes Schema

In Snowflakes schema, some of Dimension tables are further, normalized and Dim tables are connected to single Fact Table. Normalization is used to organize attributes and tables of database to minimize the data redundancy.

Normalization involves breaking a table into less redundant smaller tables without losing any information and smaller tables are joined to Dimension table.

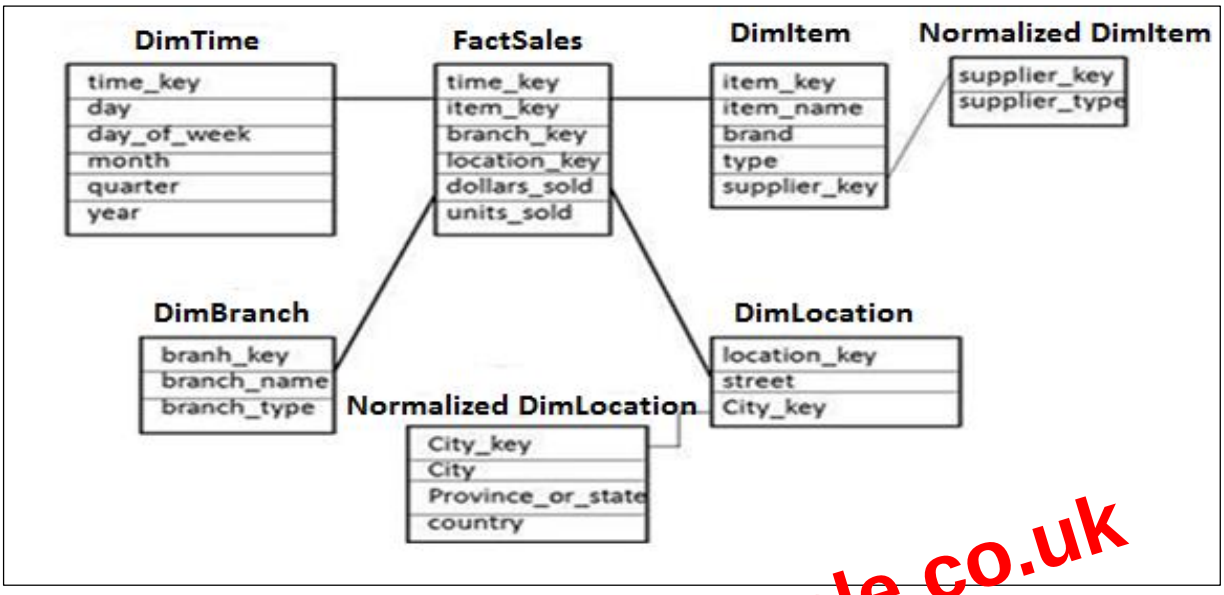


Figure: Snowflakes Schema: Dim, Normalized Dim and Fact Table

In the above example, DimItem and DimLocation Dimension tables are normalized without losing any information. This is called Snowflakes schema where dimension tables are further normalized to smaller tables.

Galaxy Schema

In Galaxy Schema, there are multiple Fact tables and Dimension tables. Each Fact table stores primary keys of few Dimension tables and measures/facts to do analysis.

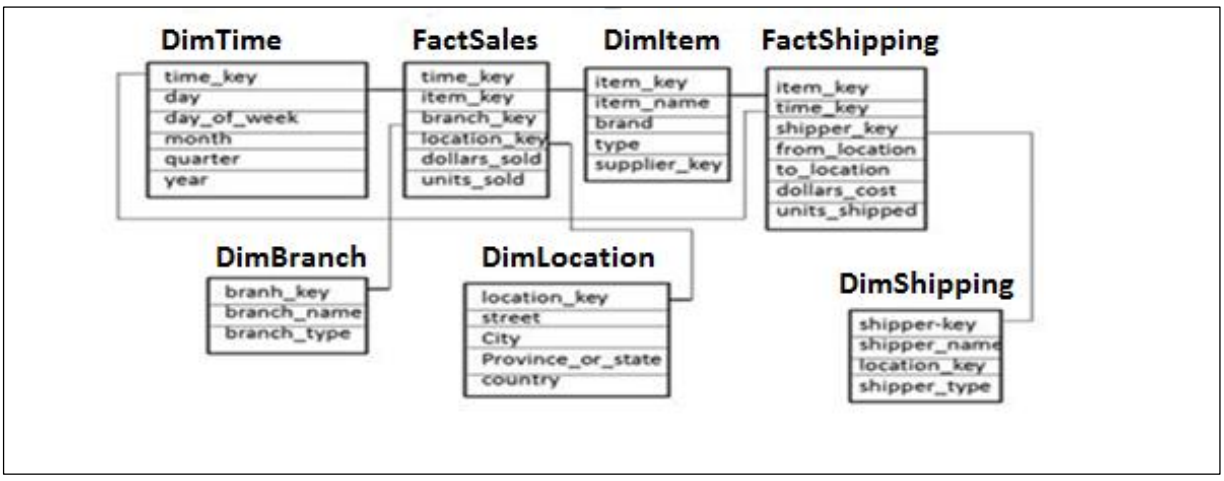
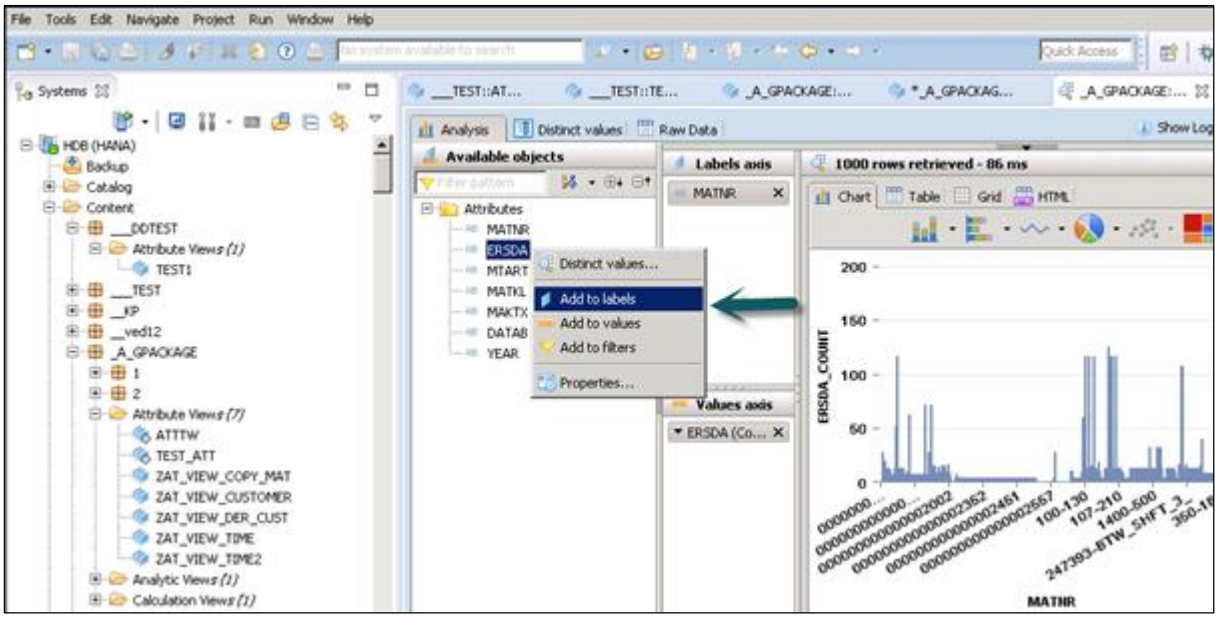


Figure: Galaxy Schema: Multiple Dim and Fact Tables



Attribute View: Adding Objects to Axis

You have four tables, two Dim tables, and two Fact tables. You have to find list of all employees with their Joining date, Emp Name, empId, Salary and Bonus.

Copy and paste the below script in SQL editor and execute.

Dim Tables: Empdim and Empdate

```
Create column table Empdim (empId nvarchar(3),Empname nvarchar(100));
Insert into Empdim values('AA1','John');
Insert into Empdim values('BB1','Anand');
Insert into Empdim values('CC1','Jason');
```

```
Create column table Empdate (caldate  date, CALMONTH nvarchar(4) ,CALYEAR
nvarchar(4));
Insert into Empdate values('20100101','04','2010');
Insert into Empdate values('20110101','05','2011');
Insert into Empdate values('20120101','06','2012');
```

Fact Tables: Empfact1, Empfact2

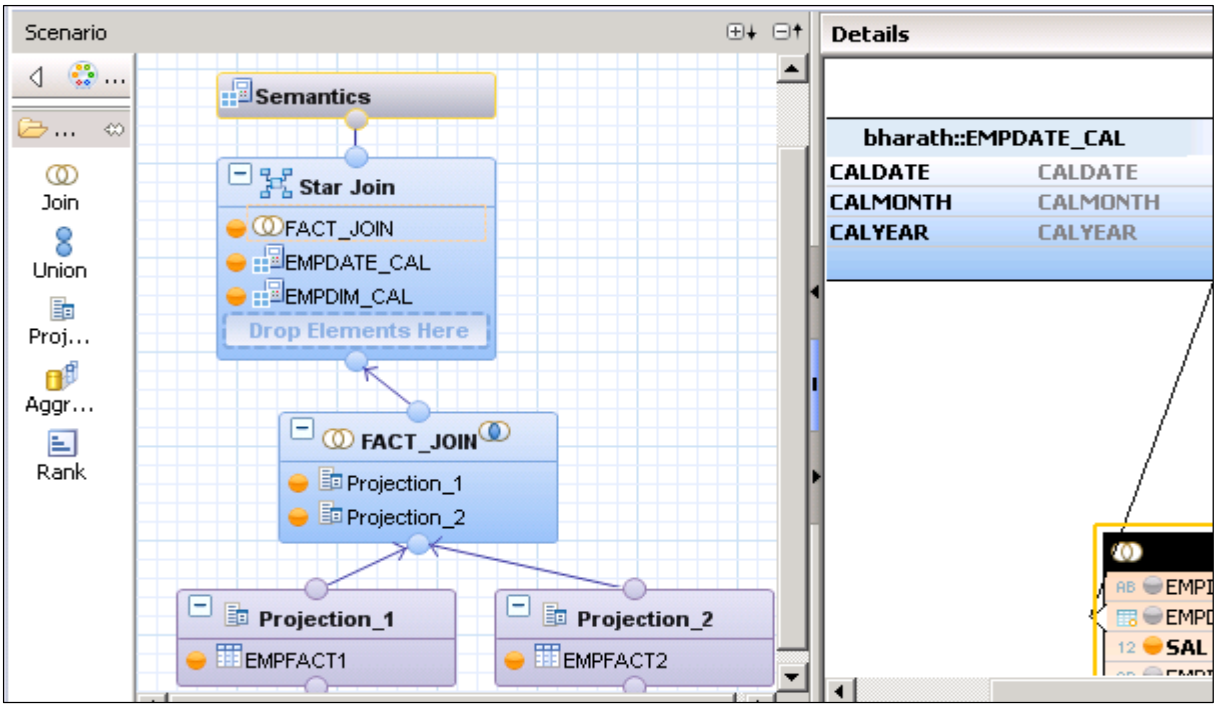
```
Create column table Empfact1 (empId nvarchar(3), Empdate date, Sal integer );
Insert into  Empfact1 values('AA1','20100101',5000);
Insert into  Empfact1 values('BB1','20110101',10000);
Insert into  Empfact1 values('CC1','20120101',12000);
```



```
Create column table Empfact2 (empId nvarchar(3), deptName nvarchar(20), Bonus
integer );
Insert into Empfact2 values ('AA1','SAP', 2000);
Insert into Empfact2 values ('BB1','Oracle', 2500);
Insert into Empfact2 values ('CC1','JAVA', 1500);
```

Now we have to implement Calculation View with Star Join. First change both Dim tables to Dimension Calculation View.

Create a Calculation View with Star Join. In Graphical pane, add 2 Projections for 2 Fact tables. Add both fact tables to both Projections and add attributes of these Projections to Output pane.



SAP HANA Calculation View: Star Join

Once view is activated successfully, right click on view name and click on Data Preview. Add attributes and measures to values and labels axis and do the analysis.

Benefits of using Star Join

It simplifies the design process. You need not to create Analytical views and Attribute Views and directly Fact tables can be used as Projections.

3NF is possible with Star Join.

Calculation View without Star Join

Create 2 Attribute Views on 2 Dim tables-Add output and activate both the views.

Create 2 Analytical Views on Fact Tables-> Add both Attribute views and Fact1/Fact2 at Data Foundation in Analytic view.

Now Create a Calculation View-> Dimension (Projection). Create Projections of both Analytical Views and Join them. Add attributes of this Join to output pane. Now Join to Projection and add output again.

Activate the view successful and go to Data preview for analysis.

15. SAP HANA – ANALYTIC PRIVILEGES

Analytic Privileges are used to limit access on HANA Information views. You can assign different types of right to different users on different component of a View in Analytic Privileges.

Sometimes, it is required that data in the same view should not be accessible to other users who do not have any relevant requirement for that data.

Example

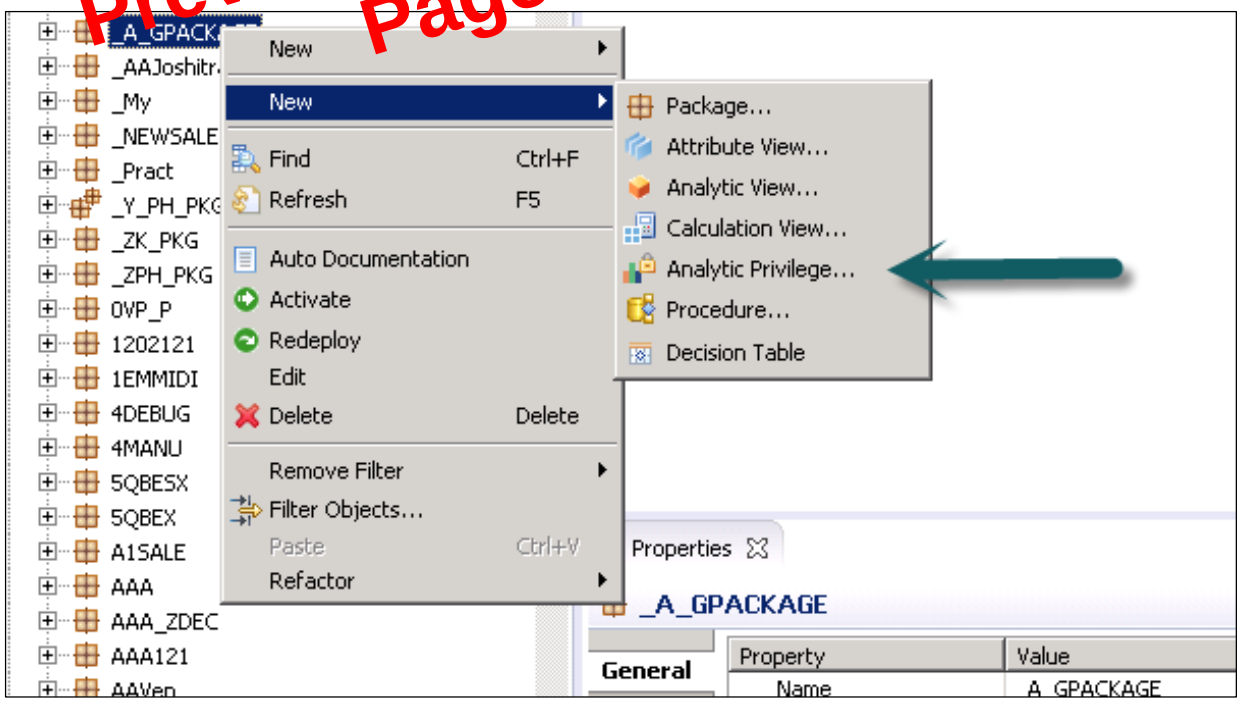
Suppose you have an Analytic view EmpDetails that has details about employees of a company- Emp name, Emp Id, Dept, Salary, Date of Joining, Emp logon, etc. Now if you do not want your Report developer to see Salary details or Emp logon details of all employees, you can hide this by using Analytic privileges option.

- Analytic Privileges are only applied to attributes in an Information View. We cannot add measures to restrict access in Analytic privileges.
- Analytic Privileges are used to control read access on SAP HANA Information views.

So we can restrict data by Empname, EmpId, Emp logon or by Emp Dept and not by numerical values like salary, bonus.

Creating Analytic Privileges

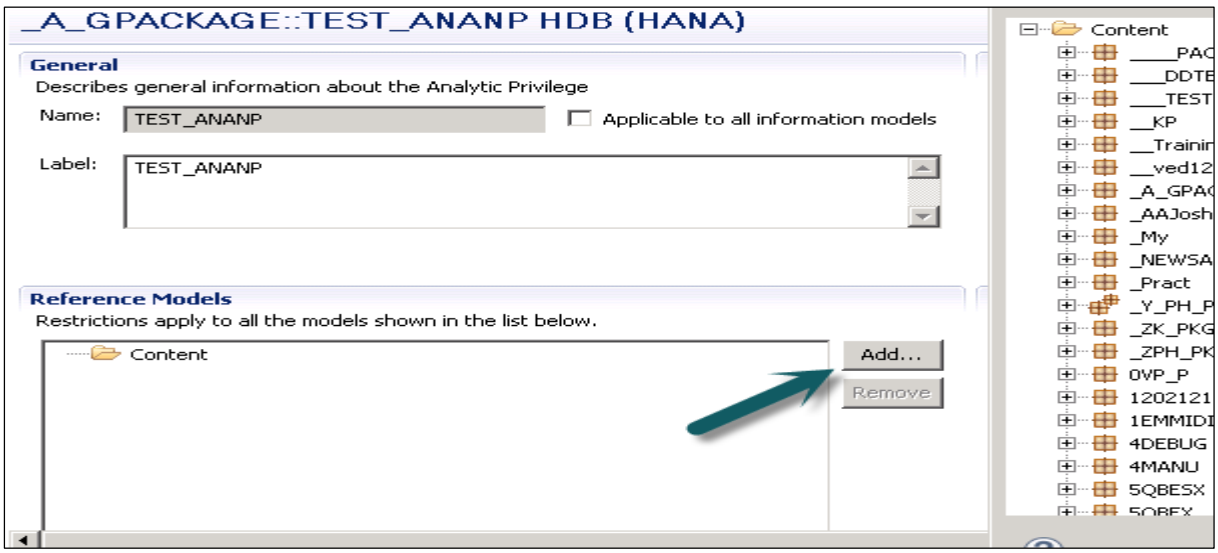
Right Click on Package name and go to new Analytic Privilege or you can open using HANA Modeler quick launch.



Enter name and Description of Analytic Privilege-> Finish. New window will open.

You can click on Next button and add Modeling view in this window before you click on finish. There is also an option to copy an existing Analytic Privilege package.

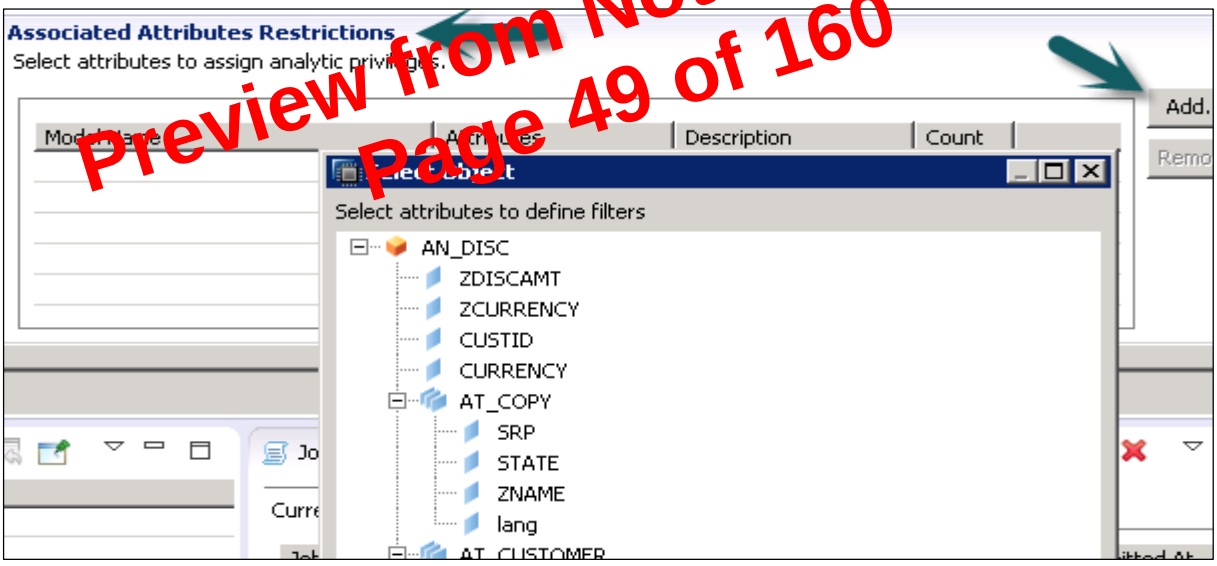
Once you click on Add button, it will show you all the views under Content tab.



SAP HANA: Adding Views to Analytic Privileges

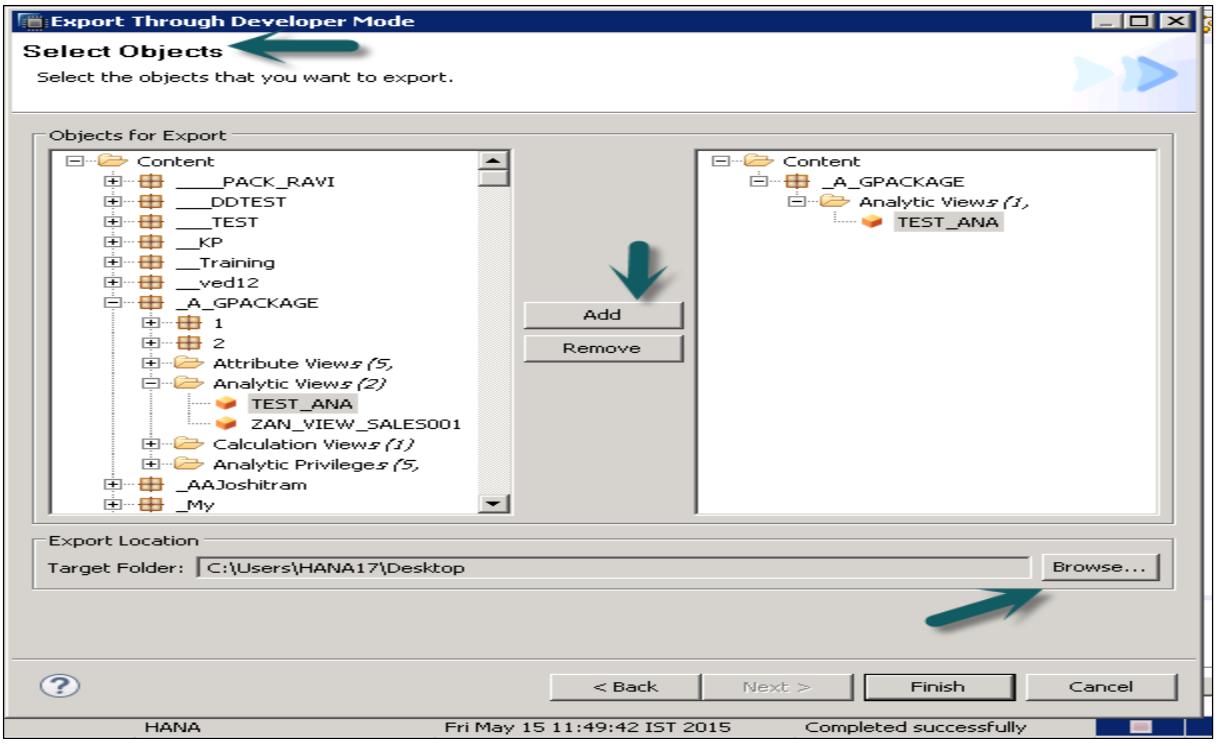
Choose View that you want to add to Analytic Privilege package and click OK. Selected View will be added under reference models.

Now to add attributes from selected view under Analytic Privilege, click on add button with Associated Attributes Restrictions window.



Add objects you want to add to Analytic privileges from select object option and click on OK.

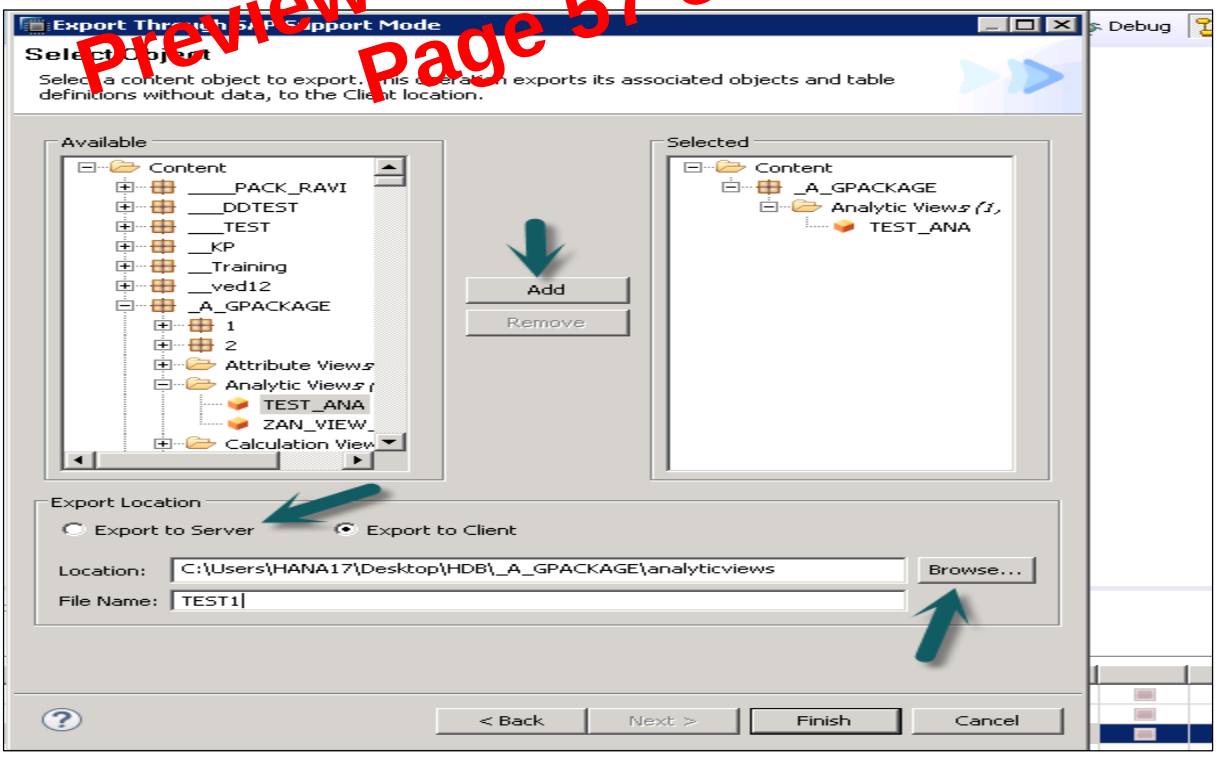
In Assign Restriction option, it allows you to add values you want to hide in Modeling View from specific user. You can add Object value that will not reflect in Data Preview of Modeling View.



Support Mode

This can be used to export the objects along with the data for SAP support purposes. This can be used when requested.

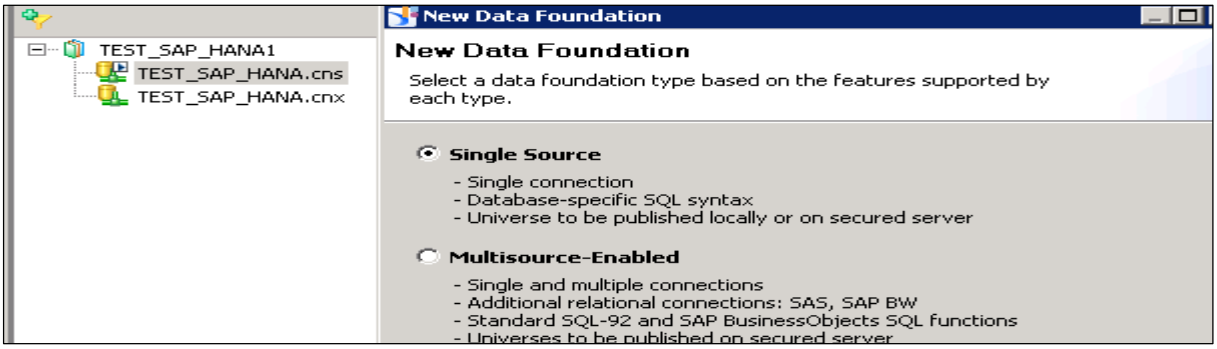
Example: User creates an Information View, which throws an error and he is not able to resolve. In that case, he can use this option to export the view along with data and share it with SAP for debugging purpose.



.cns- connection type represents secured Repository connection that should be used to create Data foundation.

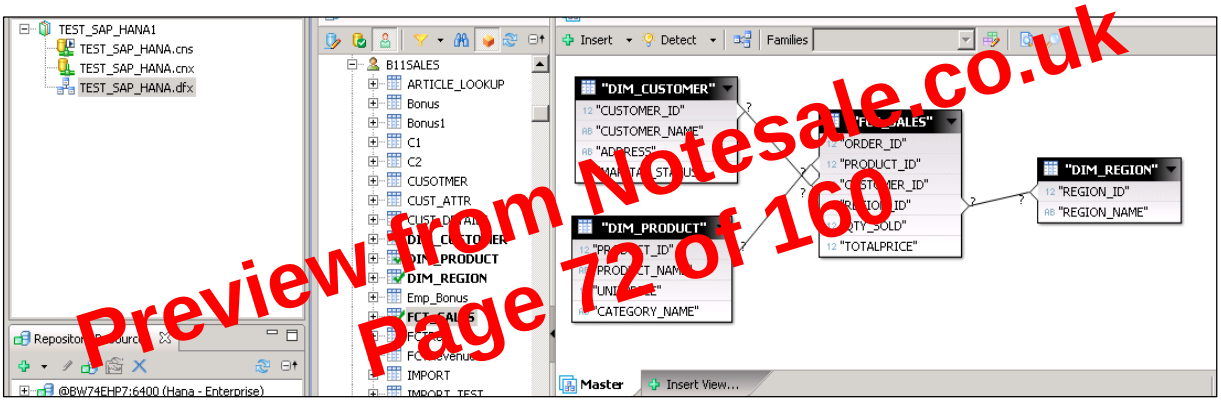
.cnx-represents local unsecured connection. If you use this connection while creating and publishing a Universe, it will not allow you to publish that to repository.

Choose .cns connection type->Right Click on this->click on New Data foundation->Enter Name of Data foundation->Next->Single source/multi source->click on Next->Finish.



It will show all the tables in HANA database with Schema name in the middle pane.

Import all tables from HANA database to master pane to create a Universe. Join Dim and Fact tables with primary keys in Dim tables to create a Schema.



Double Click on the Joins and detect Cardinality->Detect-> OK->Save All at the top. Now we have to create a new Business layer on the data foundation that will be consumed by BI Application tools.

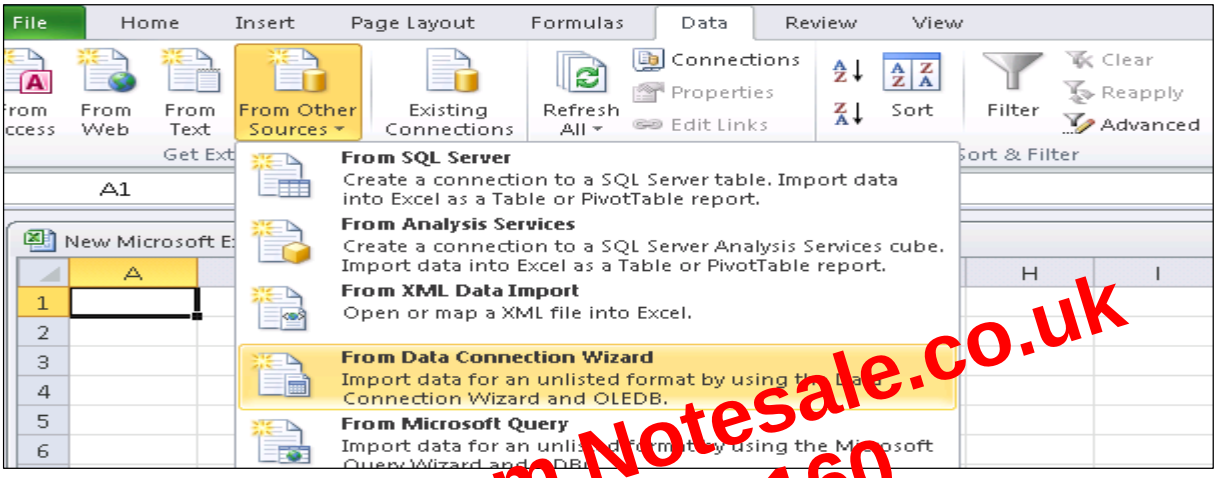
Right Click on .dfx and choose new Business Layer->Enter Name->Finish->. It will show all the objects automatically, under master pane->. Change Dimension to Measures (Type-Measure change Projection as required) ->Save All.

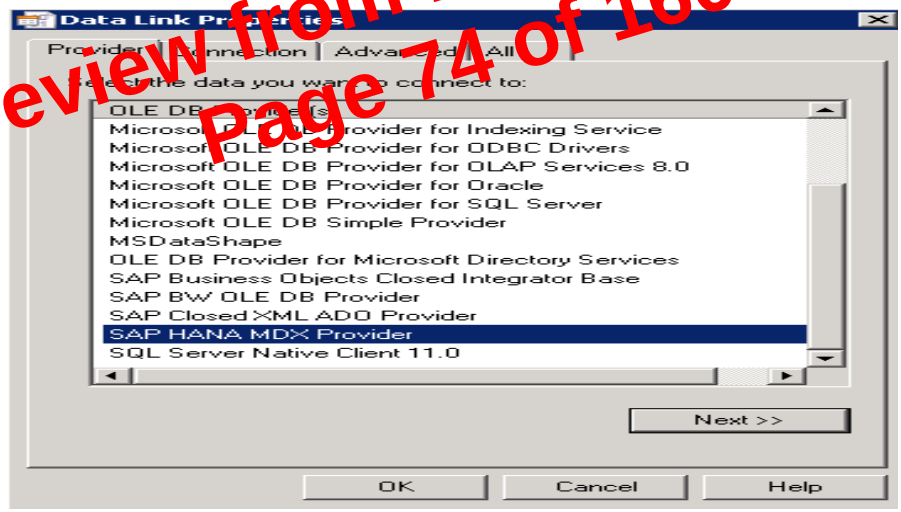
21. SAP HANA – EXCEL INTEGRATION

Microsoft Excel is considered the most common BI reporting and analysis tool by many organizations. Business Managers and Analysts can connect it to HANA database to draw Pivot tables and charts for analysis.

Connecting MS Excel to HANA

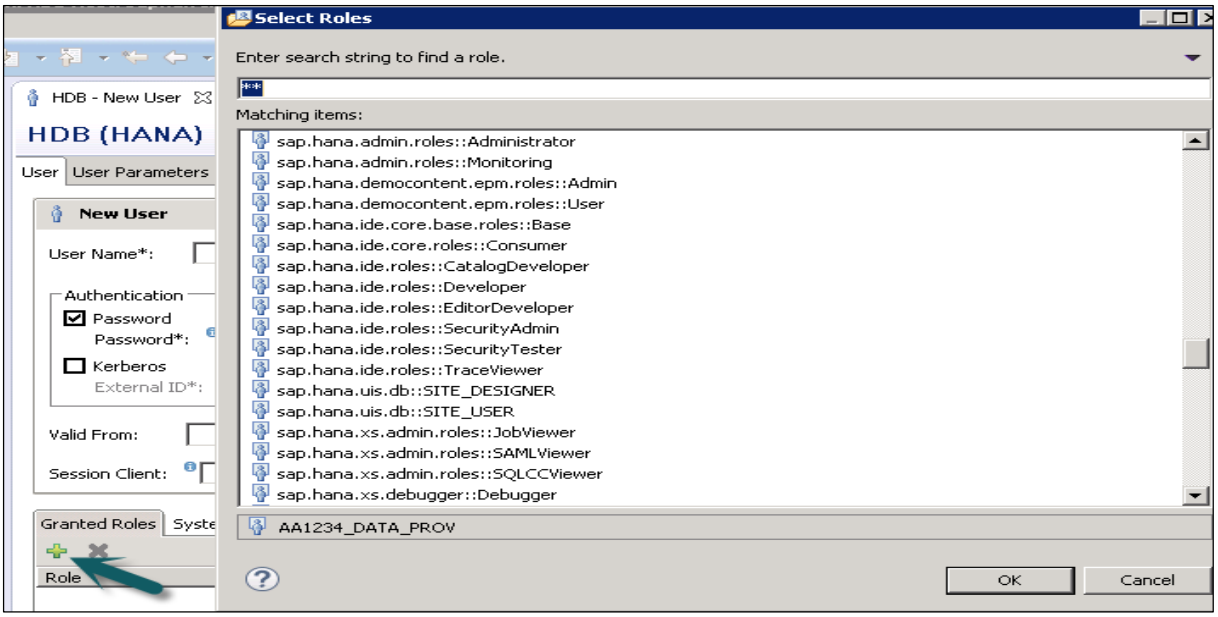
Open Excel and go to Data tab-> from other sources-> click on Data connection wizard-> Other/ Advanced and click on Next-> Data link properties will open.





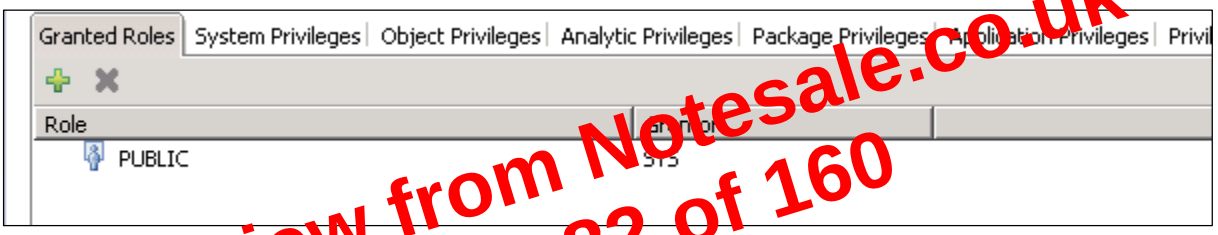
Choose SAP HANA MDX Provider from this list to connect to any MDX data source-> Enter HANA system details (server name, instance, user name and password)-> click on Test Connection-> Connection succeeded-> OK.

It will give you the list of all packages in drop down list that are available in HANA system. You can choose an Information view -> click Next-> Select Pivot table/others-> OK.



Granted role to Users

PUBLIC: This is Generic role and is assigned to all database users by default. This role contains read only access to system views and execute privileges for some procedures. These roles cannot be revoked.



Modeling

It contains all privileges required for using the information modeler in the SAP HANA studio.

System Privileges

There are different types of System privileges that can be added to a user profile. To add a system privileges to a user profile, click on + sign.

System privileges are used for Backup/Restore, User Administration, Instance start and stop, etc.

Content Admin

It contains the similar privileges as that in MODELING role, but with the addition that this role is allowed to grant these privileges to other users. It also contains the repository privileges to work with imported objects.

Object Privileges and their applicability on database objects:

Privilege	Schema	Table	View
ALL PRIVILEGES	---	YES	YES
ALTER	YES	YES	---
CREATE ANY	YES	---	---
DEBUG	YES	---	YES
DELETE	YES	YES	YES
DROP	YES	YES	YES
EXECUTE	YES	---	---
INDEX	YES	YES	---
INSERT	YES	YES	YES

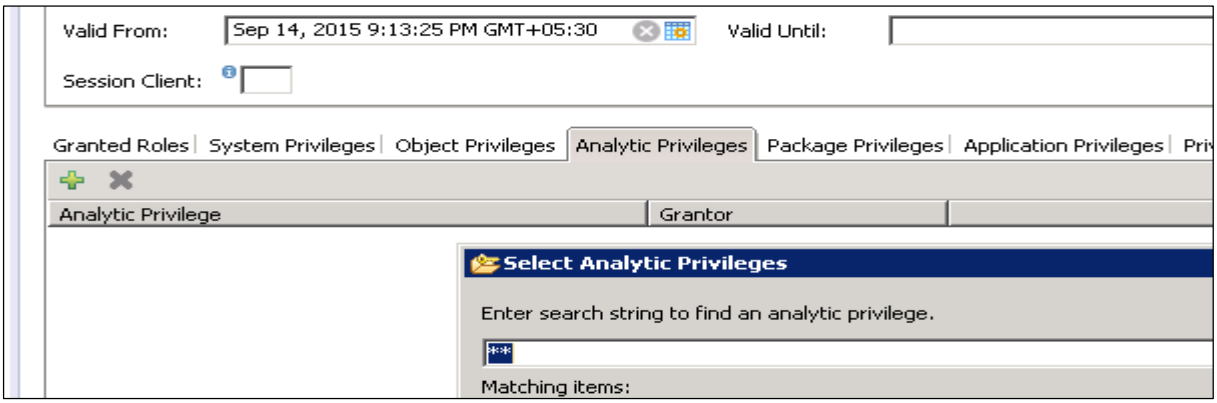
Analytic Privileges

Sometimes, it is required that data in the same view should not be accessible to other users who does not have any relevant requirement for that data.

Analytic privileges are used to limit the access on HANA Information Views at object level. We can apply row and column level security in Analytic Privileges.

Analytic Privileges are used for:

- Allocation of row and column level security for specific value range.
- Allocation of row and column level security for modeling views.



Analytic Privileges for Users and User Roles

24. SAP HANA – AUTHENTICATIONS

All SAP HANA users that have access on HANA database are verified with different Authentications method. SAP HANA system supports various types of authentication method and all these login methods are configured at time of profile creation.

Below is the list of authentication methods supported by SAP HANA:

- User name/Password
- Kerberos
- SAML 2.0
- SAP Logon tickets
- X.509

Authentication method	Available for access via JDBC/ODBC	Available for access via HTTP (SAP HANA XS)
User name/password Password policies, for example password length and complexity, can be defined	Yes	Yes (basic authentication, form-based login)
Kerberos	Yes	Yes (SPNEGO)
SAML 2.0	Yes (bearer token)	Yes
SAP logon/assertion tickets	Yes	Yes
X.509	-	Yes

User Name/Password

This method requires a HANA user to enter user name and password to login to database. This user profile is created under User management in HANA Studio-> Security Tab.

Password should be as per password policy i.e. Password length, complexity, lower and upper case letters, etc.

You can change the password policy as per your organization's security standards. Please note that password policy cannot be deactivated.

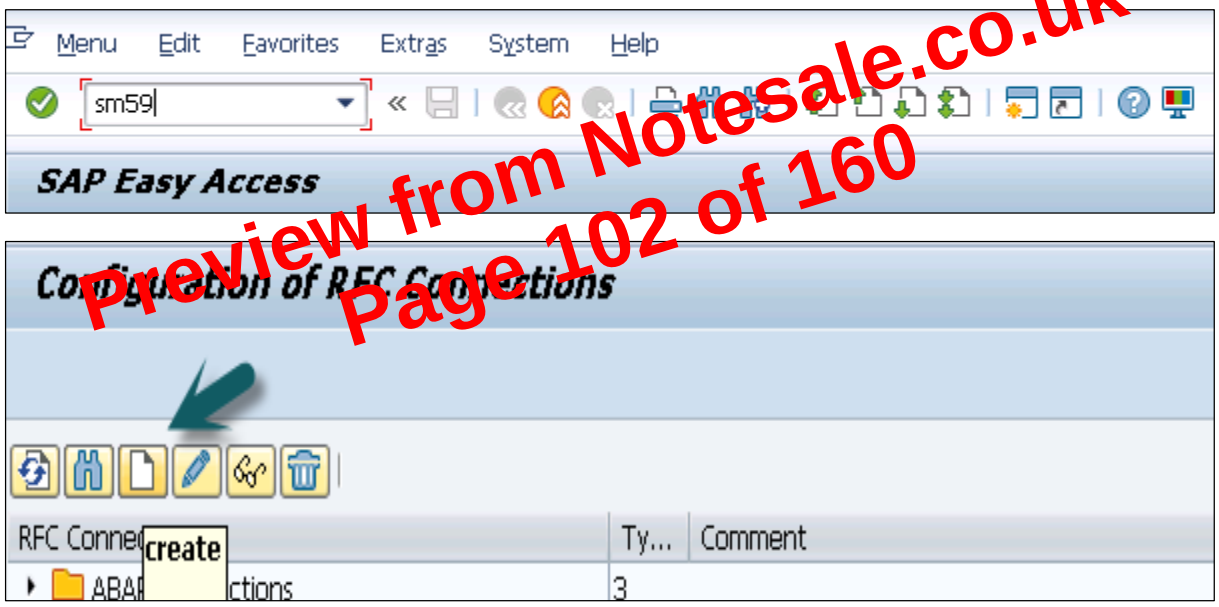
- SAP LT uses trigger based approach. It has no measureable performance impact in source system.
- It also provides data transformation and filtering capability before loading to HANA database.
- It allows real-time data replication, replicating only relevant data into HANA from SAP and non-SAP source systems.
- It is fully integrated with HANA System and HANA studio.

Creating a Trusted RFC Connection in ECC system

On your source SAP system AA1 you want to setup a trusted RFC towards target system BB1. When it is done, it would mean that when you are logged onto AA1 and your user has enough authorization in BB1, you can use the RFC connection and logon to BB1 without having to re-enter user and password.

Using RFC trusted/trusting relationship between two SAP systems, RFC from a trusted system to a trusting system, password is no required for logging on to the trusting system.

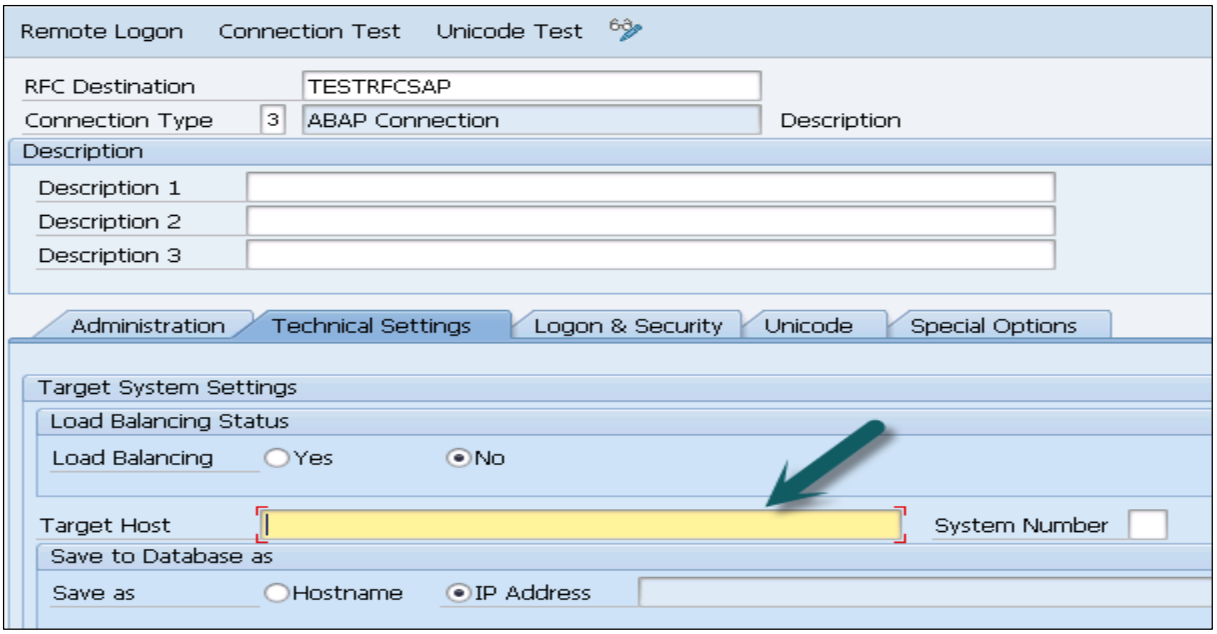
Open SAP ECC system using SAP logon. Enter transaction number sm59 -> this is transaction number to create a new Trusted RFC connection ->Click on 3rd icon to open a new connection wizard-> click on Create and new window will open.



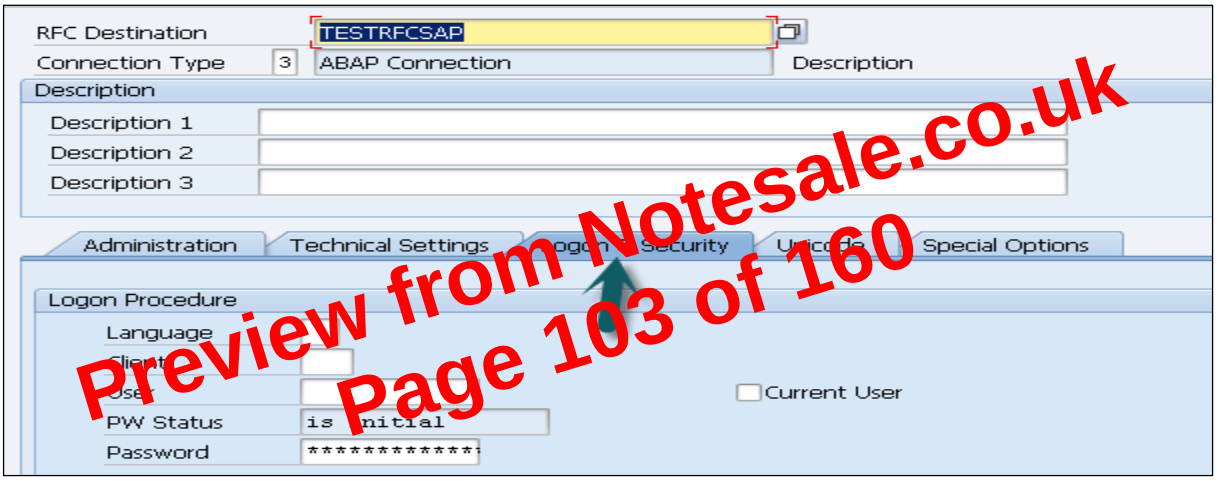
RFC Destination ECCHANA (enter name of RFC destination) Connection Type: 3 (for ABAP system)

Go to Technical Setting:

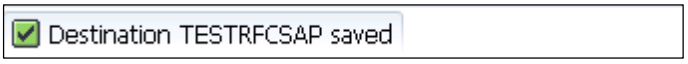
Enter Target host: ECC system name, IP and enter System number.



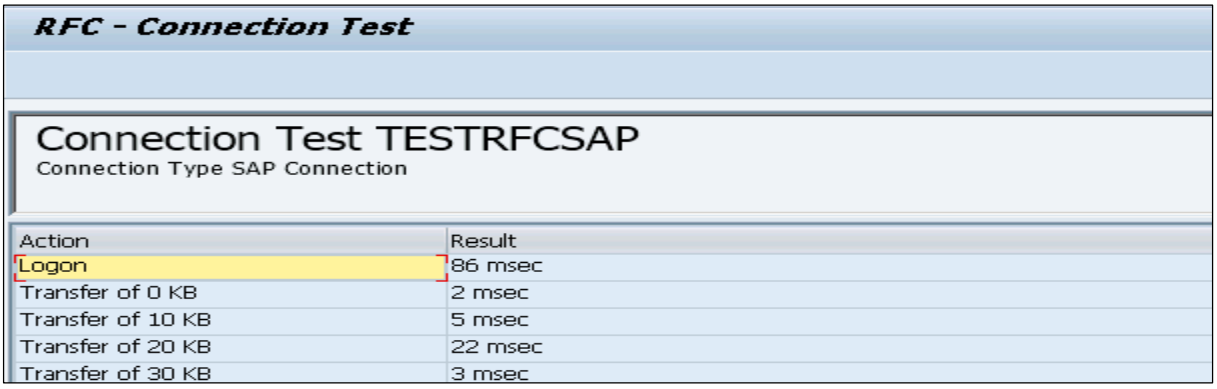
Go to Logon & Security tab, Enter Language, Client, ECC system user name and password.

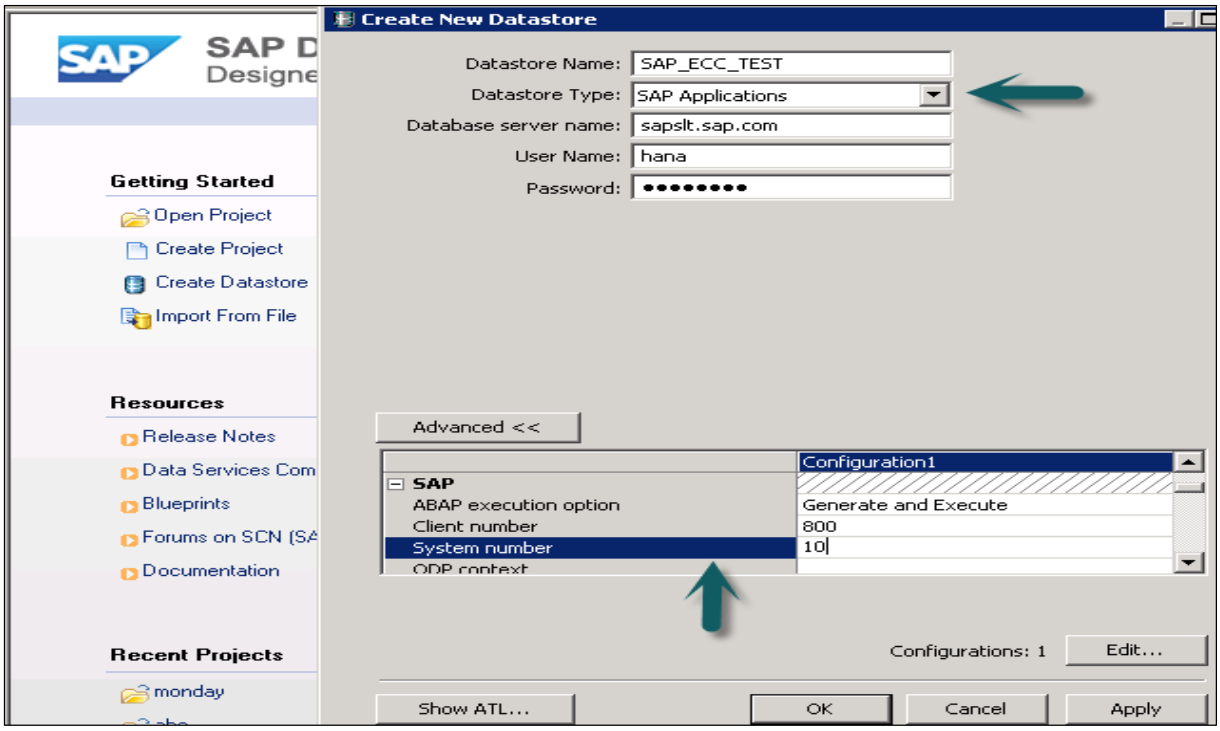


Click on the Save option at the top.

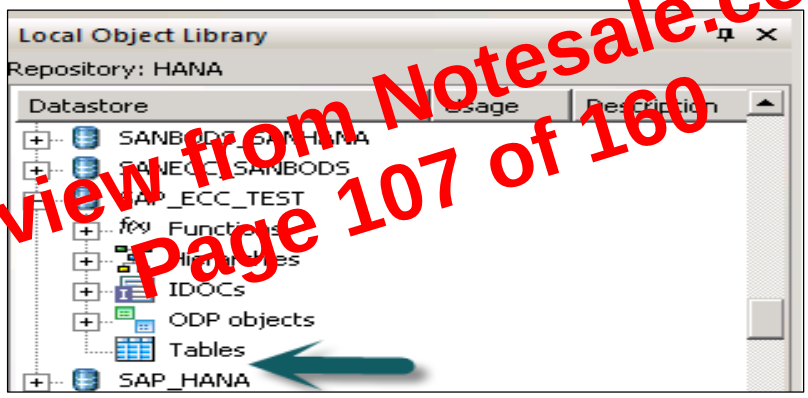


Click on Test Connection and it will successfully test the connection.





This data store will come under local object library, if you expand this there is no table inside it.



Right click on Table ->Import by name ->Enter ECC table to import from ECC system (MARA is default table in ECC system)-> Import -> Now expand Table -> MARA -> Right Click View Data. If data is displayed, Data store connection is fine.

Now, to choose target system as HANA database, create a new data store. Create Data store-> Name of data store SAP_HANA_TEST ->Data store type (database) -> Database type SAP HANA -> Database version HANA 1.x.

Enter HANA server name, user name and password for HANA system and OK.

PART 6: SAP HANA – MONITORING

35. SAP HANA – PERSISTENT LAYER

SAP HANA database persistence layer is responsible to manage logs for all the transactions to provide standard data back up and system restore function.

It ensures that database can be restored to the most recent committed state after a restart or after a system crash and transactions are executed completely or completely undone. SAP HANA Persistent Layer is part of Index server and it has data and transaction log volumes for HANA system and in-memory data is regularly saved to these volumes. There are services in HANA system that has their own persistence. It also provides save points and logs for all the database transactions from the last save point.

Why does SAP HANA database need a Persistent Layer?

- Main memory is volatile therefore data is lost during a restart or power outage.
- Data needs to be stored in persisted medium.
- Backup & Restore is available.
- It ensures that the database is restored to the most recent committed state after a restart and that transaction are either completely executed or completely undone.

Data and Transaction Log Volumes

Database can always be restored to its most recent state, to ensure these changes to data in the database are regularly copied to disk. Log files containing data changes and certain transaction events are also saved regularly to disk. Data and logs of a system are stored in Log volumes.

Data volumes stores SQL data and undo log information and also SAP HANA information modeling data. This information is stored in data pages, which are called Blocks. These blocks are written to data volumes at regular time interval, which is known as save point.

Log volumes store the information about data changes. Changes that are made between two log points are written to Log volumes and called log entries. They are saved to log buffer when transaction is committed.

Savepoints

In SAP HANA database, changed data is automatically saved from memory to disk. These regular intervals are called savepoints and by default they are set to occur every five minutes. Persistence Layer in SAP HANA database performs these savepoint at regular interval. During this operation changed data is written to disk and redo logs are also saved to disk as well.

The data belonging to a Savepoint tells consistent state of the data on disk and remains there until the next savepoint operation has completed. Redo log entries are written to the log volumes for all changes to persistent data. In the event of a database restart, data from the last completed savepoint can be read from the data volumes, and redo log entries written to the log volumes.

Frequency of savepoint can be configured by global.ini file. Savepoints can be initiated by other operations like database shut down or system restart. You can also run savepoint by executing the below command:

ALTER System SAVEPOINT

To save data and redo logs to log volumes, you should ensure that there is enough disk space available to capture these, otherwise the system will issue a disk full event and database will stop working.

During the HANA system installation, following default directories are created as the storage location for data and log volumes:

- /usr/sap/<SID>/SYS/global/hdb/data
- /usr/sap/<SID>/SYS/global/hdb/log

These directories are defined in global.ini file and can be changed at later stage.

Note that Savepoints do not affect the performance of transactions executed in HANA system. During a savepoint operation, transactions continue to run as normal. With HANA system running on proper hardware, impact of savepoints on the performance of system is negligible.

41. SAP HANA – SQL OPERATORS

An operator is a special character used primarily in SQL statement's with WHERE clause to perform operation, such as comparisons and arithmetic operations. They are used to pass conditions in a SQL query.

Operator types given below can be used in SQL statements in HANA:

- Arithmetic Operators
- Comparison/Relational Operators
- Logical Operators
- Set Operators

Arithmetic Operators

Arithmetic operators are used to perform simple calculation functions like addition, subtraction, multiplication, division and percentage.

Operator	Description
+	Addition - Adds values on either side of the operator
-	Subtraction - Subtracts right hand operand from left hand operand
*	Multiplication - Multiplies values on either side of the operator
/	Division - Divides left hand operand by right hand operand
%	Modulus - Divides left hand operand by right hand operand and returns remainder

43. SAP HANA – SQL EXPRESSIONS

An Expression is used to evaluate a clause to return values. There are different SQL expressions that can be used in HANA:

- Case Expressions
- Function Expressions
- Aggregate Expressions
- Subqueries in Expressions

Case Expression

This is used to pass multiple conditions in a SQL expression. It allows the use of IF-ELSE-THEN logic without using procedures in SQL statements.

Example

```
SELECT COUNT( CASE WHEN sal < 2000 THEN 1 ELSE NULL END ) count1,
COUNT( CASE WHEN sal BETWEEN 2001 AND 4000 THEN 1 ELSE NULL END ) count2,
COUNT( CASE WHEN sal > 4000 THEN 1 ELSE NULL END ) count3 FROM emp;
```

This statement will return count1, count2, count3 with integer value as per passed condition.

Function Expressions

Function expressions involve SQL inbuilt functions to be used in Expressions.

Aggregate Expressions

Aggregate functions are used to perform complex calculations like Sum, Percentage, Min, Max, Count, Mode, Median, etc. Aggregate Expression uses Aggregate functions to calculate single value from multiple values.

Aggregate Functions: Sum, Count, Minimum, Maximum. These are applied on measure values (facts) and It is always associated with a dimension.

Common aggregate functions include:

- Average ()
- Count ()
- Maximum ()
- Median ()
- Minimum ()
- Mode ()
- Sum ()

49. SAP HANA – SQL DATA PROFILING

SQL Data Profiling task is used to understand and analyze data from multiple data sources. It is used to remove incorrect, incomplete data and prevent data quality problems before they are loaded in Data warehouse.

Here are the benefits of SQL Data Profiling tasks:

- It helps is analyzing source data more effectively.
- It helps in understanding the source data better.
- It remove incorrect, incomplete data and improve data quality before it is loaded into Data warehouse.
- It is used with Extraction, Transformation and Loading task.

The Data Profiling task checks profiles that helps to understand a data source and identify problems in the data that has to be fixed.

You can use the Data Profiling task inside an Integration Services package to profile data that is stored in SQL Server and to identify potential problems with data quality.

Note: Data Profiling Task works only with SQL Server data sources and does not support any other file based or third party data sources.

Access Requirement

To run a package contains Data Profiling task, user account must have read/write permissions with CREATE TABLE permissions on the tempdb database.

Data Profiler Viewer

Data Profile Viewer is used to review the profiler output. The Data Profile Viewer also supports drilldown capability to help you understand data quality issues that are identified in the profile output. This drill down capability sends live queries to the original data source.

Data Profiling Task Setup and Reviewing

Setting up the Data Profiling Task

It involves execution of a package that contains Data Profiling task to compute the profiles. The task saves the output in XML format to a file or a package variable.

Reviewing the Profiles

To view the data profiles, send the output to a file and then use the Data Profile Viewer. This viewer is a stand-alone utility that displays the profile output in both summary and detail format with optional drilldown capability.

Data Profiling: Configuration Options

The Data Profiling task has these convenient configuration options:

Wildcard columns

While configuring a profile request, the task accepts '*' wildcard in place of a column name. This simplifies the configuration and makes it easier to discover the characteristics of unfamiliar data. When the task runs, the task profiles every column that has an appropriate data type.

Quick Profile

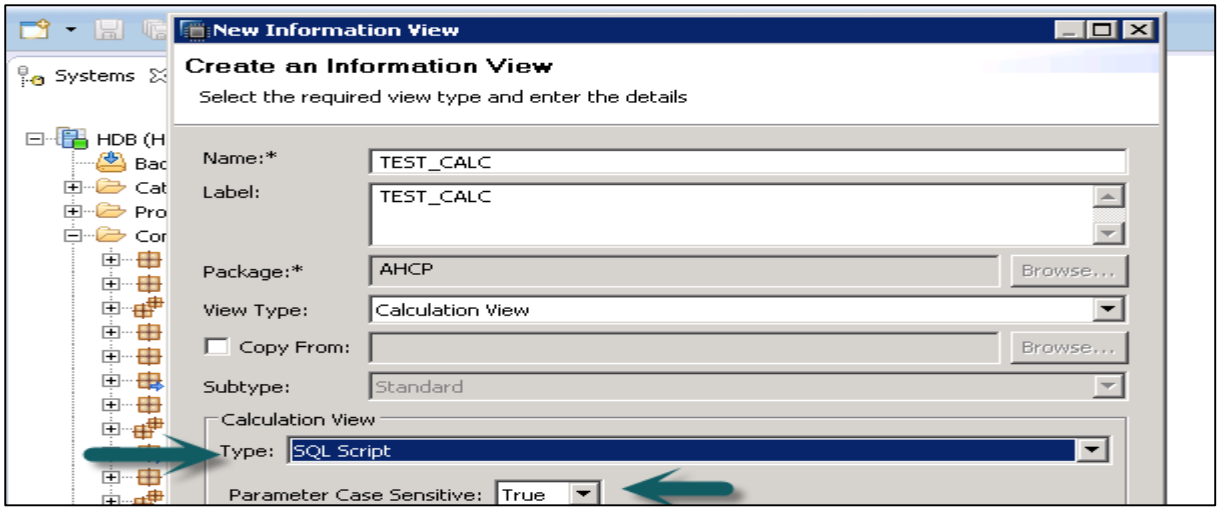
You can select Quick Profile to configure the task quickly. A Quick Profile profiles a table or view by using all the default profiles and settings.

The Data Profiling Task can compute eight different data profiles. Five of these profiles can check individual columns and the remaining three analyze- multiple columns or relationships between columns.

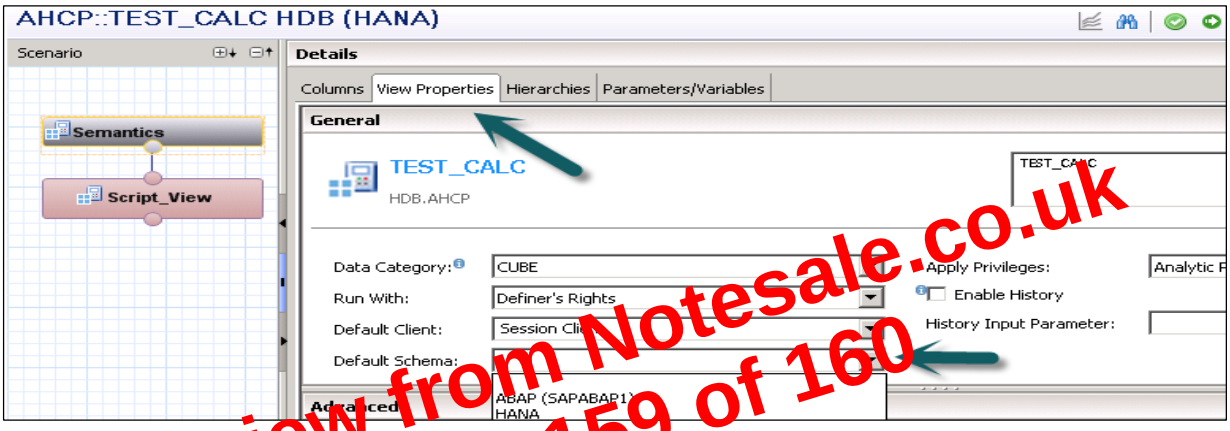
Data Profiling: Task Outputs

The Data Profiling task outputs the selected profiles into XML format that is structured like DataProfile.xsd schema.

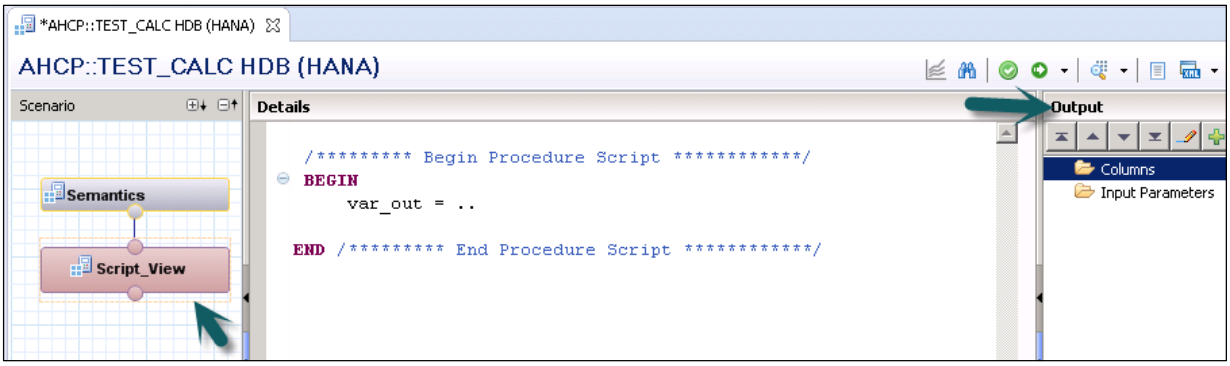
You can save a local copy of the schema and view the local copy of the schema in Microsoft Visual Studio or another schema editor, in an XML editor or in a text editor such as Notepad.



Select default schema: Select the Semantics node ->Choose the View Properties tab -> In the Default Schema dropdown list, select the default schema.



Choose SQL Script node in the Semantics node -> Define the output structure. In the output pane, choose Create Target. Add the required output parameters and specify its length and type.



To add multiple columns that are part of existing information views or catalog tables or table functions to the output structure of script-based calculation views:

In the Output pane, choose Start of the navigation path New Next navigation step Add Columns from End of the navigation path -> Name of the object that contains the columns you want to add to the output -> Select one or more objects from the dropdown list -> Choose Next.