

SAP HANA - QUICK GUIDE

https://www.tutorialspoint.com/sap_hana/sap_hana_quick_guide.htm

Copyright © tutorialspoint.com

SAP HANA - OVERVIEW

SAP HANA is a combination of HANA Database, Data Modeling, HANA Administration and Data Provisioning in one single suite. In SAP HANA, HANA stands for High-Performance Analytic Appliance.

According to former SAP executive, Dr. Vishal Sikka, HANA stands for Hasso's New Architecture. HANA developed interest by mid-2011 and various fortune 500 companies started considering it as an option to maintain Business Warehouse needs after that.

Features of SAP HANA

The main features of SAP HANA are given below –

- SAP HANA is a combination of software and hardware innovation to process huge amount of real time data.
- Based on multi core architecture in distributed system environment.
- Based on row and column type of data-storage in database.
- Used extensively in Memory Computing Engine *IMCE* to process and analyze massive amount of real time data.
- It reduces cost of ownership, increases application performance, enables new applications to run on real time environment that were not possible before.
- It is written in C++, supports and runs on only one Operating System Suse Linux Enterprise Server 11 SP1/2.

Need for SAP HANA

Today, most successful companies respond quickly to market changes and new opportunities. A key to this is the effective and efficient use of data and information by analyst and managers.

HANA overcomes the limitations mentioned below –

- Due to increase in "Data Volume", it is a challenge for the companies to provide access to real time data for analysis and business use.
- It involves high maintenance cost for IT companies to store and maintain large data volumes.
- Due to unavailability of real time data, analysis and processing results are delayed.

SAP HANA Vendors

SAP has partnered with leading IT hardware vendors like IBM, Dell, Cisco etc. and combined it with SAP licensed services and technology to sell SAP HANA platform.

There are, total, 11 vendors that manufacture HANA Appliances and provide onsite support for installation and

configuration of HANA system.

Top few Vendors include –

- IBM
- Dell
- HP
- Cisco
- Fujitsu
- Lenovo *China*
- NEC
- Huawei

According to statistics provided by SAP, IBM is one of major vendor of SAP HANA hardware appliances and has a market share of 50-52% but according to another market survey conducted by HANA clients, IBM has a market hold up to 70%.

SAP HANA Installation

HANA Hardware vendors provide preconfigured appliances for hardware, Operating System and SAP software product.

Vendor finalizes the installation by an onsite setup and configuration of HANA components. This onsite visit includes deployment of HANA system in Data Center, Connectivity to Organization Network, SAP system ID adaption, updates from Solution Manager, SAP Router Connectivity, SSL Enablement and other system configuration.

Customer/Client starts with connectivity of Data Source system and BI clients. HANA Studio Installation is completed on local system and HANA system is added to perform Data modeling and administration.

SAP HANA - IN-MEMORY COMPUTING ENGINE

An In-Memory database means all the data from source system is stored in a RAM memory. In a conventional Database system, all data is stored in hard disk. SAP HANA In-Memory Database wastes no time in loading the data from hard disk to RAM. It provides faster access of data to multicore CPUs for information processing and analysis.

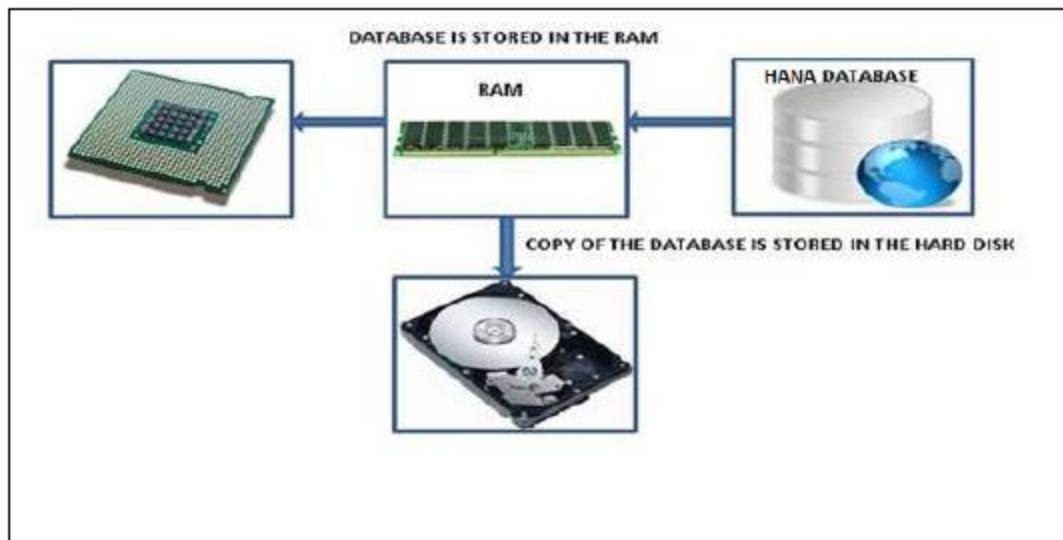
Features of In-Memory Database

The main features of SAP HANA in-memory database are –

- SAP HANA is Hybrid In-memory database.
- It combines row based, column based and Object Oriented base technology.
- It uses parallel processing with multicore CPU Architecture.
- Conventional Database reads memory data in 5 milliseconds. SAP HANA In-Memory database reads data

in 5 nanoseconds.

It means, memory reads in HANA database are 1 million times faster than a conventional database hard disk memory reads.



Analysts want to see current data immediately in real time and do not want to wait for data until it is loaded to SAP BW system. SAP HANA In-Memory processing allows loading of real time data with use of various data provisioning techniques.

Advantages of In-Memory Database

- HANA database takes advantage of in-memory processing to deliver the fastest data-retrieval speeds, which is enticing to companies struggling with high-scale online transactions or timely forecasting and planning.
- Disk-based storage is still the enterprise standard and price of RAM has been declining steadily, so memory-intensive architectures will eventually replace slow, mechanical spinning disks and will lower the cost of data storage.
- In-Memory Column-based storage provides data compression up to 11 times, thus, reducing the storage space of huge data.
- This speed advantages offered by RAM storage system are further enhanced by the use of multi-core CPUs, multiple CPUs per node and multiple nodes per server in a distributed environment.

SAP HANA - STUDIO

SAP HANA studio is an Eclipse-based tool. SAP HANA studio is both, the central development environment and the main administration tool for HANA system. Additional features are –

- It is a client tool, which can be used to access local or remote HANA system.
- It provides an environment for HANA Administration, HANA Information Modeling and Data Provisioning in HANA database.

SAP HANA Studio can be used on following platforms –

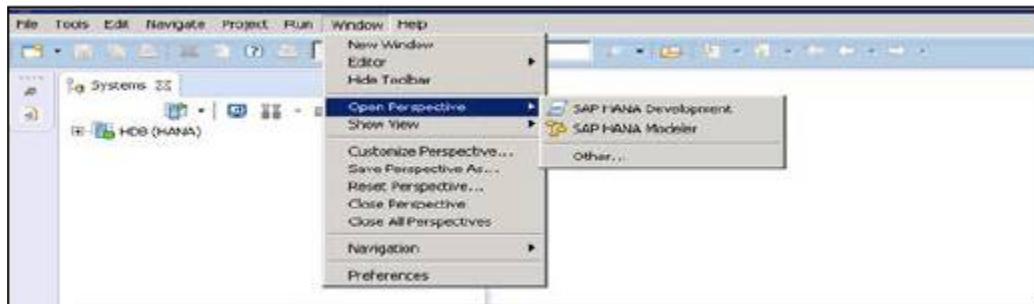
- Microsoft Windows 32 and 64 bit versions of: Windows XP, Windows Vista, Windows 7
- SUSE Linux Enterprise Server SLES11: x86 64 bit
- Mac OS, HANA studio client is not available

Depending on HANA Studio installation, not all features may be available. At the time of Studio installation, specify the features you want to install as per the role. To work on most recent version of HANA studio, Software Life Cycle Manager can be used for client update.

SAP HANA Studio Perspectives / Features

SAP HANA Studio provides perspectives to work on the following HANA features. You can choose Perspective in HANA Studio from the following option –

HANA Studio → Window → Open Perspective → Other



Sap Hana Studio Administration

Toolset for various administration tasks, excluding transportable design-time repository objects. General troubleshooting tools like tracing, the catalog browser and SQL Console are also included.

SAP HANA Studio Database Development

It provides Toolset for content development. It addresses, in particular, the DataMarts and ABAP on SAP HANA scenarios, which do not include SAP HANA native application development *XS*.

SAP HANA Studio Application Development

SAP HANA system contains a small Web server, which can be used to host small applications. It provides Toolset for developing SAP HANA native applications like application code written in Java and HTML.

By default, all features are installed.

SAP HANA - STUDIO ADMINISTRATION VIEW

To Perform HANA Database Administration and monitoring features, SAP HANA Administration Console Perspective can be used.

Administrator Editor can be accessed in several ways –

- **From System View Toolbar** – Choose Open Administration default button

- In System View – Double Click on HANA System or Open Perspective



HANA Studio: Administrator Editor

In Administration View: HANA studio provides multiple tabs to check configuration and health of the HANA system. Overview Tab tells General Information like, Operational Status, start time of first and last started service, version, build date and time, Platform, hardware manufacturer, etc.

Adding a HANA System to Studio

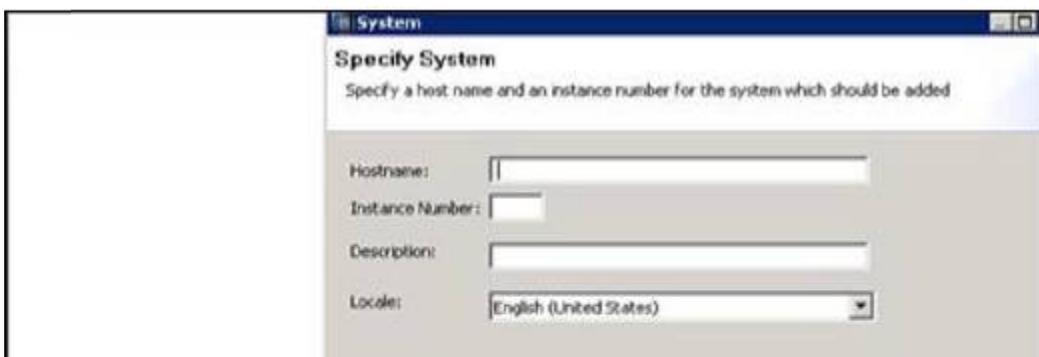
Single or multiple systems can be added to HANA studio for administration and information modeling purpose. To add new HANA system, host name, instance number and database user name and password is required.

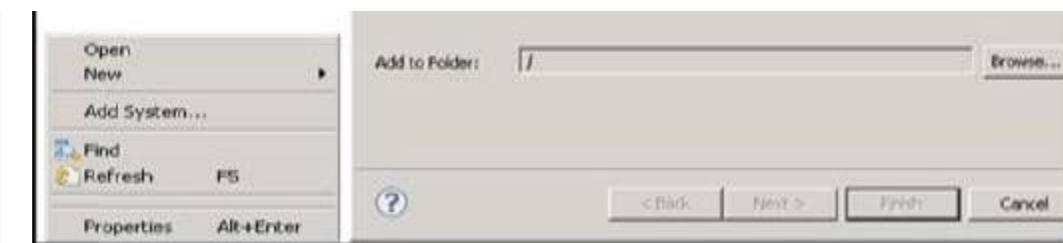
- Port 3615 should be open to connect to Database
- Port 31015 Instance No 10
- Port 30015 Instance No 00
- SSH port should also be opened

Adding a System to Hana Studio

To add a system to HANA studio, follow the given steps.

Right Click in Navigator space and click on Add System. Enter HANA system details, i.e. Host name & Instance number and click next.





Enter Database user name and password to connect to SAP HANA database. Click on Next and then Finish.



Once you click on Finish, HANA system will be added to System View for administration and modeling purpose. Each HANA system has two main sub-nodes, Catalog and Content.



Catalog and Content

Catalog

It contains all available Schemas i.e. all data structures, tables and data, Column views, Procedures that can be used in Content tab.

Content

The Content tab contains design time repository, which holds all information of data models created with the HANA Modeler. These models are organized in Packages. The content node provides different views on same physical data.

SAP HANA - SYSTEM MONITOR

System Monitor in HANA studio provides an overview of all your HANA system at a glance. From System Monitor, you can drill down into details of an individual system in Administration Editor. It tells about Data Disk, Log disk, Trace Disk, Alerts on resource usage with priority.

The following Information is available in System Monitor –

Column	Description
System ID	ID assigned to system when added
Operational State	Overall system status
Alerts	The system issues alerts when resource usage and statistical thresholds are violated. These alerts are categorized as low, medium, or high priority. There are also informational alerts. The number of alerts and their status is shown here
Data Disk (GB)	Size of the data volume on disk
Log Disk (GB)	Size of the log volume on disk
Trace Disk (GB)	Size of trace files on disk
Database Resident Memory (GB)	Size of resident memory at operating system level owing to SAP HANA database processes
System Resident Memory (GB)	Total size of resident memory in the operating system
Used Memory (GB)	Amount of physical memory used by the SAP HANA database
CPU (%)	Percentage of CPU used by the SAP HANA database
Hostname	Name of the server hosting the SAP HANA database
Instance Number	Instance number is the administrative unit that comprises the server software components
System Data Disk (GB)	Total disk space occupied on disk(s) containing data
System Log Disk (GB)	Total disk space occupied on disk(s) containing log files
Column	Description
System Trace Disk (GB)	Total disk space occupied on disk(s) containing trace files
System Physical Memory (GB)	Total amount of physical memory used
System CPU (%)	Overall CPU usage
Distributed	Indicates whether the system is running on a single host or it is a distributed system running on more than one host
Start Time First	Time that the first service started This value is updated when system is restarted for any reason
Start Time Latest	Time that the last service was started, if, for example, one of the services was restarted individually
Version	Software version number of the SAP HANA studio
Platform	Operating system on which the SAP HANA studio is running
Number of Crash Dump Files	The number of crash dump files in the trace directory of the system

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

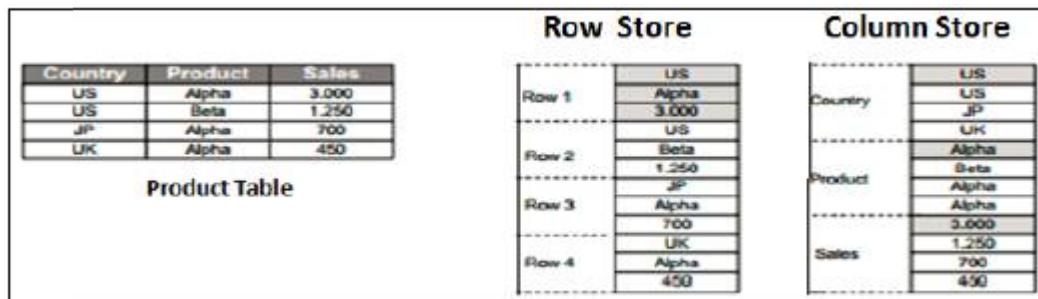
- Provides multiple views of transactional data stored in physical tables of HANA database for analysis and business logic purpose.
- Informational modeler only works for column based storage tables.
- Information Modeling Views are consumed by Java or HTML based applications or SAP tools like SAP Lumira 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.

Column Store

In a Column store table, Data is stored vertically. So, similar data types come together as shown in the example above. It provides faster memory read and write operations with help of In-Memory Computing Engine.

In a conventional database, data is stored in Row based structure i.e. horizontally. SAP HANA stores data in both row and Column based structure. This provides Performance optimization, flexibility and data compression in HANA database.

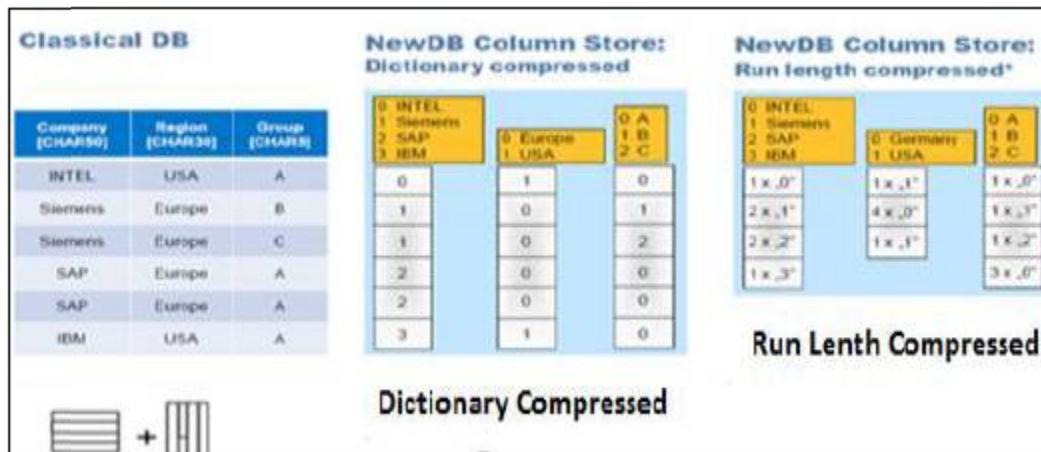
Storing Data in Columnar based table has following benefits –

- Data Compression
- Faster read and write access to tables as compared to conventional Row based storage
- Flexibility & parallel processing
- Perform Aggregations and Calculations at higher speed

There are various methods and algorithms how data can be stored in Column based structure- Dictionary Compressed, Run Length Compressed and many more.

In Dictionary Compressed, cells are stored in form of numbers in tables and numeral cells are always performance optimized as compared to characters.

In Run length compressed, it saves the multiplier with cell value in numerical format and multiplier shows repetitive value in table.



Functional Difference - Row vs Column Store

It is always advisable to use Column based storage, if SQL statement has to perform aggregate functions and calculations. Column based tables always perform better when running aggregate functions like Sum, Count, Max, Min.

Row based storage is preferred when output has to return complete row. The example given below makes it easy to understand.

Table - SALES				
	Date	Country	Product	Sales
Row 1	2013-01-01	India	Chocolate	1000
Row 2	2013-01-10	India	Ice-cream	2000
Row 3	2013-02-20	Germany	Chocolate	4000
Row 4	2013-03-01	US	Noodle	500

Row Store Column Store Row Store Column Store

Row Selection: 0<=T1> AND C1='INDIA' AND C2='CHOCOLATE'



In the above example, while running an Aggregate function *Sum* in sales column with Where clause, it will only use Date and Sales column while running SQL query so if it is column based storage table then it will be performance optimized, faster as data is required only from two columns.

While running a simple Select query, full row has to be printed in output so it is advisable to store table as Row based in this scenario.

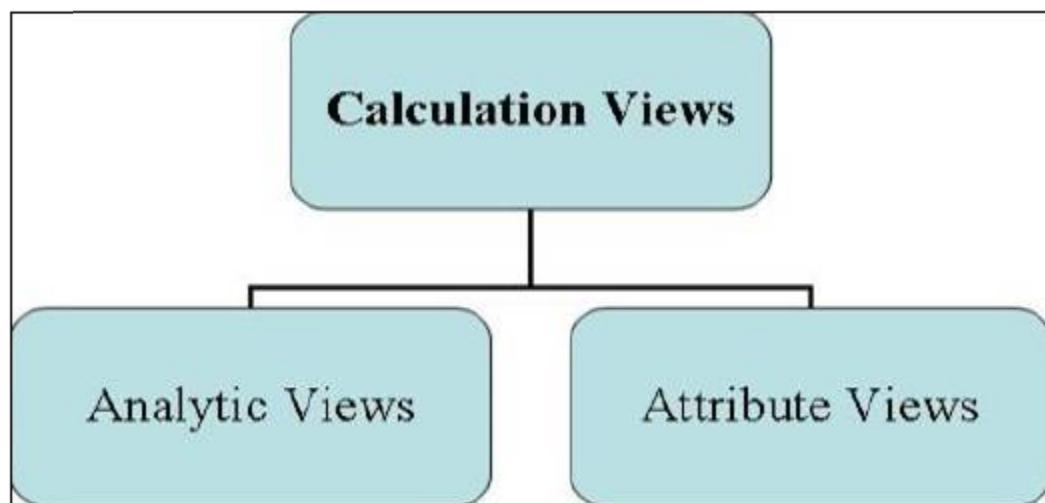
Information Modeling Views

Attribute View

Attributes are non-measurable elements in a database table. They represent master data and similar to characteristics of BW. Attribute Views are dimensions in a database or are used to join dimensions or other attribute views in modeling.

Important features are –

- Attribute views are used in Analytic and Calculation views.
- Attribute view represent master data.
- Used to filter size of dimension tables in Analytic and Calculation View.



Analytic View

Analytic Views use power of SAP HANA to perform calculations and aggregation functions on the tables in

database. It has at least one fact table that has measures and primary keys of dimension tables and surrounded by dimension tables contain master data.

Important features are –

- Analytic views are designed to perform Star schema queries.
- Analytic views contain at least one fact table and multiple dimension tables with master data and perform calculations and aggregations
- They are similar to Info Cubes and Info objects in SAP BW.
- Analytic views can be created on top of Attribute views and Fact tables and performs calculations like number of unit sold, total price, etc.

Calculation Views

Calculation Views are used on top of Analytic and Attribute views to perform complex calculations, which are not possible with Analytic Views. Calculation view is a combination of base column tables, Attribute views and Analytic views to provide business logic.

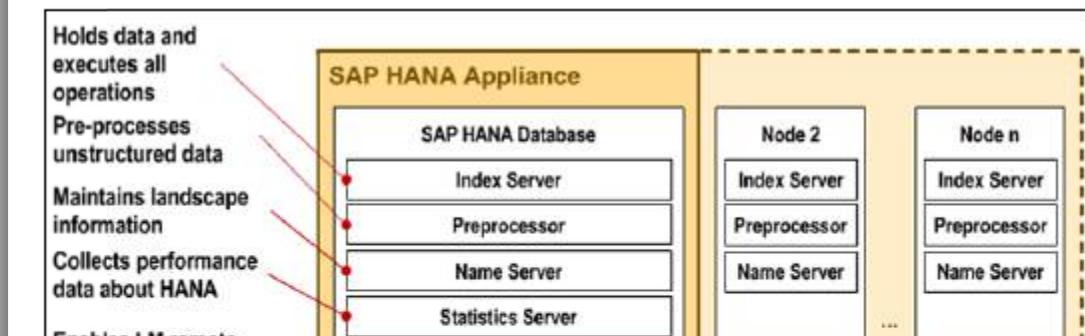
Important features are –

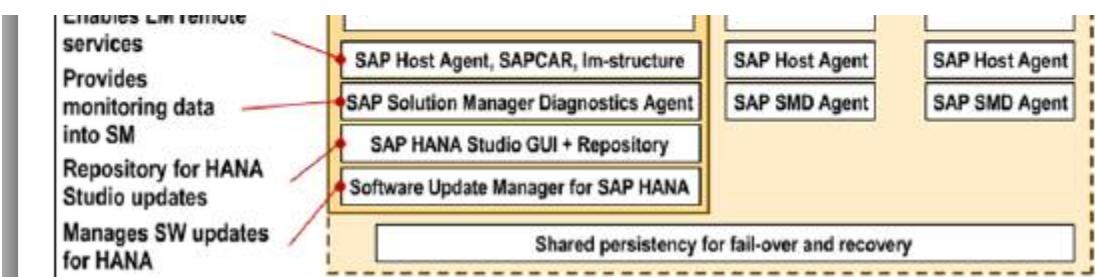
- Calculation Views are defined either graphical using HANA Modeling feature or scripted in the SQL.
- It is created to perform complex calculations, which are not possible with other views- Attribute and Analytic views of SAP HANA modeler.
- One or more Attribute views and Analytic views are consumed with help of inbuilt functions like Projects, Union, Join, Rank in a Calculation View.

SAP HANA - CORE ARCHITECTURE

SAP HANA was initially developed in Java and C++ and designed to run only Operating System Suse Linux Enterprise Server 11. SAP HANA system consists of multiple components that are responsible to emphasize computing power of HANA system.

- Most important component of SAP HANA system is Index Server, which contains SQL/MDX processor to handle query statements for database.
- HANA system contains Name Server, Preprocessor Server, Statistics Server and XS engine, which is used to communicate and host small web applications and various other components.



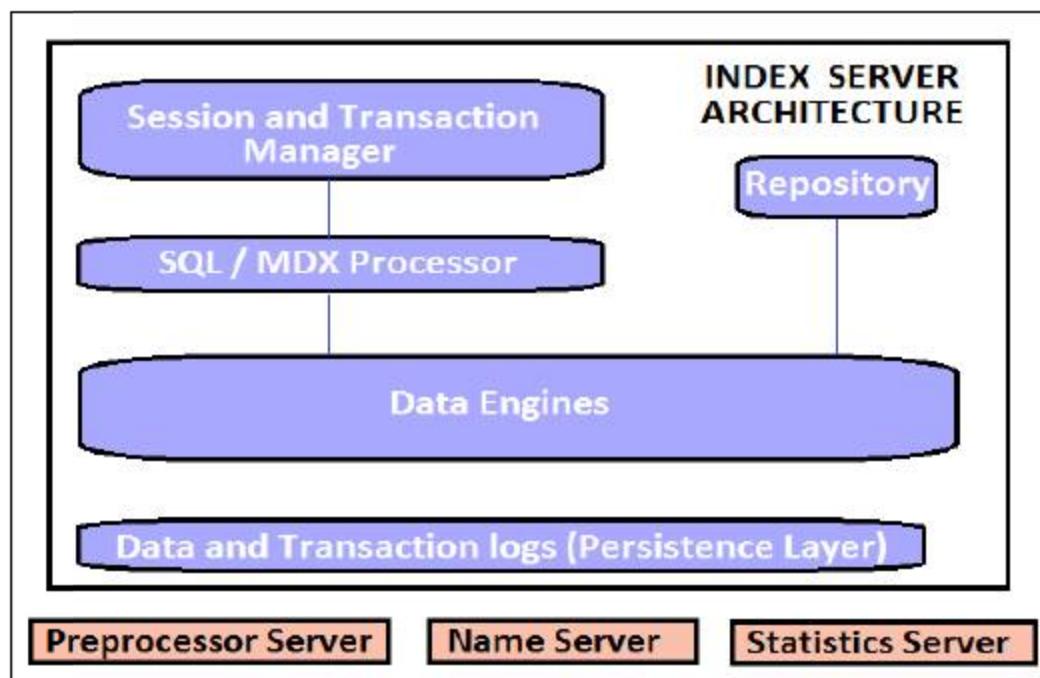


Index Server

Index Server is heart of SAP HANA database system. It contains actual data and engines for processing that data. When SQL or MDX is fired for SAP HANA system, an Index Server takes care of all these requests and processes them. All HANA processing takes place in Index Server.

Index Server contains Data engines to handle all SQL/MDX statements that come to HANA database system. It also has Persistence Layer that is responsible for durability of HANA system and ensures HANA system is restored to most recent state when there is restart of system failure.

Index Server also has Session and Transaction Manager, which manage transactions and keep track of all running and closed transactions.



Index Server – Architecture

SQL/MDX Processor

It is responsible for processing SQL/MDX transactions with data engines responsible to run queries. It segments all query requests and direct them to correct engine for the performance Optimization.

It also ensures that all SQL/MDX requests are authorized and also provide error handling for efficient processing

of these statements. It contains several engines and processors for query execution –

- MDX *MultiDimensionExpression* is query language for OLAP systems like SQL is used for Relational database. MDX Engine is responsible to handle queries and manipulates multidimensional data stored in OLAP cubes.
- Planning Engine is responsible to run planning operations within SAP HANA database.
- Calculation Engine converts data into Calculation models to create logical execution plan to support parallel processing of statements.
- Stored Procedure processor executes procedure calls for optimized processing; it converts OLAP cubes to HANA optimized cubes.

Transaction and Session Management

It is responsible to coordinate all database transactions and keep track of all running and closed transactions.

When a transaction is executed or failed, Transaction manager notifies relevant data engine to take necessary actions.

Session management component is responsible to initialize and manage sessions and connections for SAP HANA system using predefined session parameters.

Persistence Layer

It is responsible for durability and atomicity of transactions in HANA system. Persistence layer provides built in disaster recovery system for HANA database.

It ensures database is restored to most recent state and ensures that all the transactions are completed or undone in case of a system failure or restart.

It is also responsible to manage data and transaction logs and also contain data backup, log backup and configuration back of HANA system. Backups are stored as save points in the Data Volumes via a Save Point coordinator, which is normally set to take back every 5-10 minutes.

Preprocessor Server

Preprocessor Server in SAP HANA system is used for text data analysis.

Index Server uses preprocessor server for analyzing text data and extracting the information from text data when text search capabilities are used.

Name Server

NAME server contains System Landscape information of HANA system. In distributed environment, there are multiple nodes with each node has multiple CPU's, Name server holds topology of HANA system and has information about all the running components and information is spread on all the components.

- Topology of SAP HANA system is recorded here.
- It decreases the time in re-indexing as it holds which data is on which server in distributed environment.

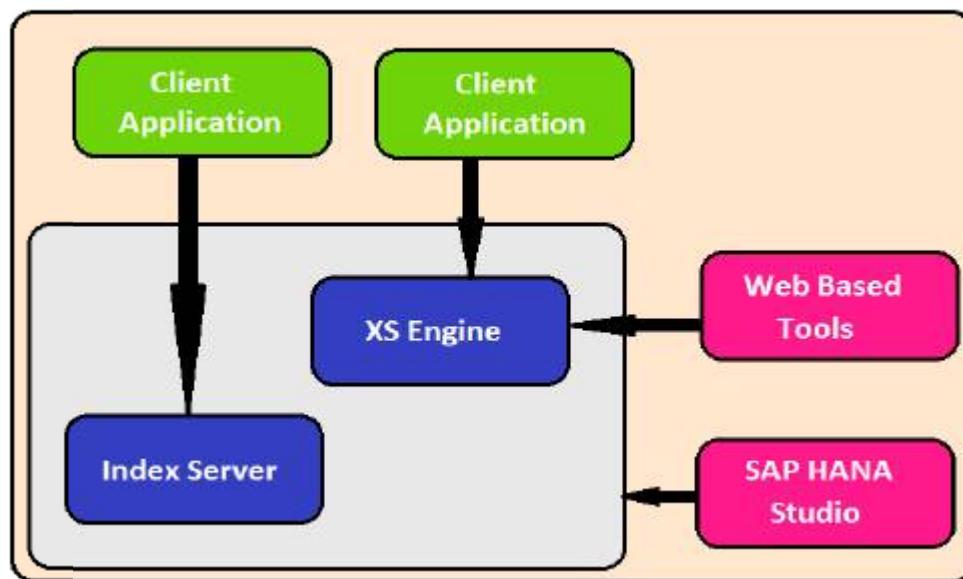
Statistical Server

This server checks and analyzes the health of all components in HANA system. Statistical Server is responsible for collecting the data related to system resources, their allocation and consumption of the resources and overall performance of HANA system.

It also provides historical data related to system performance for analyses purpose, to check and fix performance related issues in HANA system.

XS Engine

XS engine helps external Java and HTML based applications to access HANA system with help of XS client. As SAP HANA system contains a web server which can be used to host small JAVA/HTML based applications.



XS Engine transforms the persistence model stored in database into consumption model for clients exposed via HTTP/HTTPS.

SAP Host Agent

SAP Host agent should be installed on all the machines that are part of SAP HANA system Landscape. SAP Host agent is used by Software Update Manager SUM for installing automatic updates to all components of HANA system in distributed environment.

LM Structure

LM structure of SAP HANA system contains information about current installation details. This information is used by Software Update Manager to install automatic updates on HANA system components.

SAP Solution Manager *SAPSOLMAN* diagnostic Agent

This diagnostic agent provides all data to SAP Solution Manager to monitor SAP HANA system. This agent provides all the information about HANA database, which include database current state and general

information.

It provides configuration details of HANA system when SAP SOLMAN is integrated with SAP HANA system.

SAP HANA Studio Repository

SAP HANA studio repository helps HANA developers to update current version of HANA studio to latest versions. Studio Repository holds the code which does this update.

Software Update Manager for SAP HANA

SAP Market Place is used to install updates for SAP systems. Software Update Manager for HANA system helps in update of HANA system from SAP Market place.

It is used for software downloads, customer messages, SAP Notes and requesting license keys for HANA system. It is also used to distribute HANA studio to end user's systems.

SAP HANA - MODELING

SAP HANA Modeler option is used to create Information views on the top of schemas → tables in HANA database. These views are consumed by JAVA/HTML based applications or SAP Applications like SAP Lumira, Office Analysis or third party software like MS Excel for reporting purpose to meet business logic and to perform analysis and extract information.

HANA Modeling is done on the top of tables available in Catalog tab under Schema in HANA studio and all views are saved under Content table under Package.

You can create new Package under Content tab in HANA studio using right click on Content and New.

All Modeling Views created inside one package comes under the same package in HANA studio and categorized according to View Type.

Each View has different structure for Dimension and Fact tables. Dim tables are defined with master data and Fact table has Primary Key for dimension tables and measures like Number of Unit sold, Average delay time, Total Price, etc.

Fact and Dimension Table

Fact Table contains Primary Keys for Dimension table and measures. They are joined with Dimension tables in HANA Views to meet business logic.

Example of Measures – Number of unit sold, Total Price, Average Delay time, etc.

Dimension Table contains master data and is joined with one or more fact tables to make some business logic. Dimension tables are used to create schemas with fact tables and can be normalized.

Example of Dimension Table – Customer, Product, etc.

Suppose a company sells products to customers. Every sale is a fact that happens within the company and the fact table is used to record these facts.

TimeID	ProductID	CustomerID	Unit

4	17	2	1
8	21	3	2
8	4	1	1

CustomerID	Name	Gender	Income	Education	Region
1	Brian Edge	M	2	3	4
2	Fred Smith	M	3	5	1
3	Sally Jones	F	1	7	3

For example, row 3 in the fact table records the fact that customer 1 *Brian* bought one item on day 4. And, in a complete example, we would also have a product table and a time table so that we know what she bought and exactly when.

The fact table lists events that happen in our company *or at least the events that we want to analyze – No of Unit Sold, Margin, and Sales Revenue*. The Dimension tables list the factors *Customer, Time, and Product* by which we want to analyze the data.

SAP HANA - SCHEMA IN DATA WAREHOUSE

Schemas are logical description of tables in Data Warehouse. Schemas are created by joining multiple fact and Dimension tables to meet some business logic.

Database uses relational model to store data. However, Data Warehouse use Schemas that join dimensions and fact tables to meet business logic. There are three types of Schemas used in a Data Warehouse –

- Star Schema
- Snowflakes Schema
- Galaxy Schema

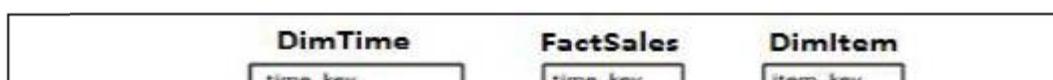
Star Schema

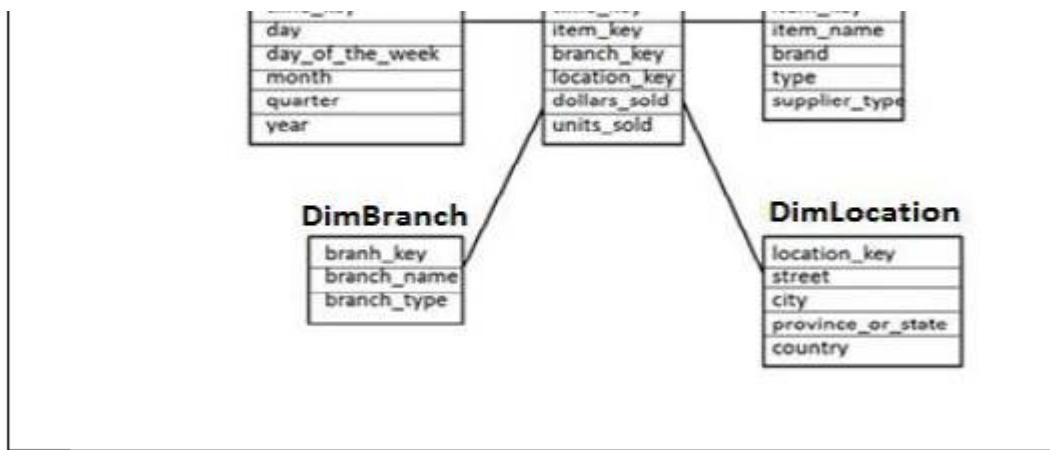
In Star Schema, Each Dimension is joined to one single Fact table. Each Dimension is represented by only one dimension and is not further normalized.

Dimension Table contains set of attribute that are used to analyze the data.

Example – In example given below, we have a Fact table FactSales that has Primary keys for all the Dim tables and measures units_sold and dollars_sold to do analysis.

We have four Dimension tables – DimTime, DimItem, DimBranch, DimLocation





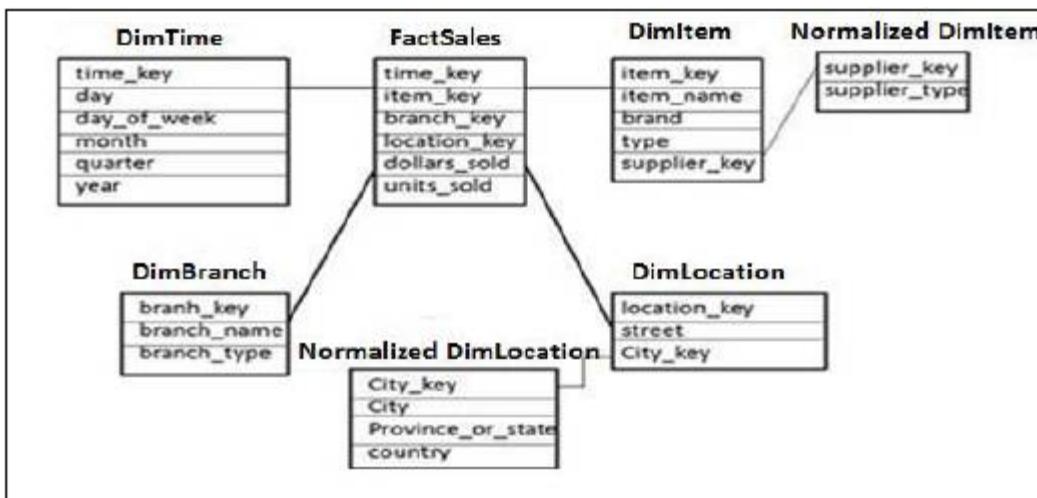
Each Dimension table is connected to Fact table as Fact table has Primary Key for each Dimension Tables that is used to join two tables.

Facts/Measures in Fact Table are used for analysis purpose along with attribute in Dimension tables.

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.

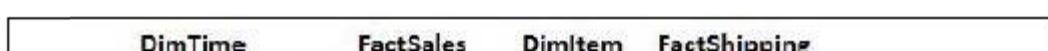
Normalization involves breaking a table into less redundant smaller tables without losing any information and smaller tables are joined to Dimension table.

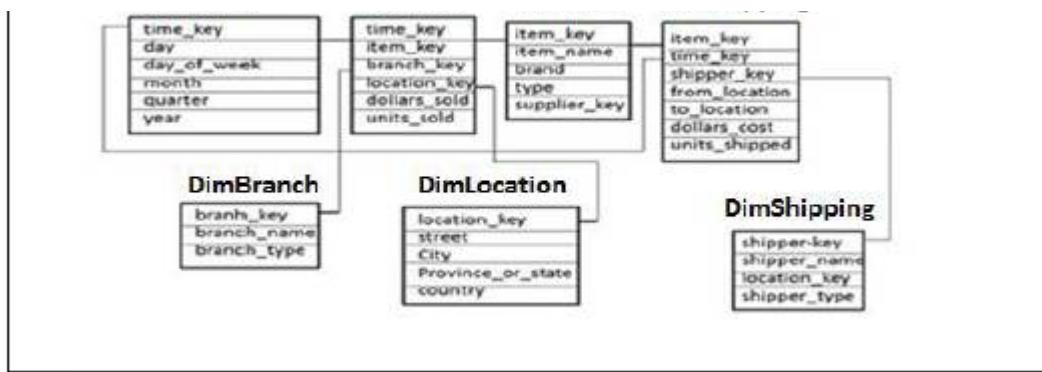


In the above example, DimItem and DimLocation Dimension tables are normalized without losing any information. This is called Snowflakes schema where dimension tables are further normalized to smaller tables.

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.





In the above example, there are two Fact tables FactSales, FactShipping and multiple Dimension tables joined to Fact tables. Each Fact table contains Primary Key for joined Dim tables and measures/Facts to perform analysis.

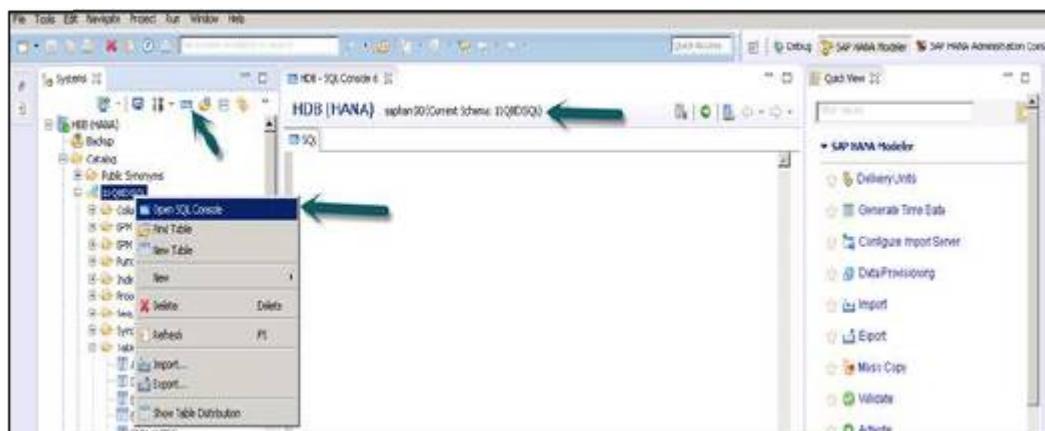
SAP HANA - TABLES

Tables in HANA database can be accessed from HANA Studio in Catalogue tab under Schemas. New tables can be created using the two methods given below –

- Using SQL editor
- Using GUI option

SQL Editor in HANA Studio

SQL Console can be opened by selecting Schema name, in which, new table has to be created using System View SQL Editor option or by Right click on Schema name as shown below –



Once SQL Editor is opened, Schema name can be confirmed from the name written on the top of SQL Editor. New table can be created using SQL Create Table statement –

```

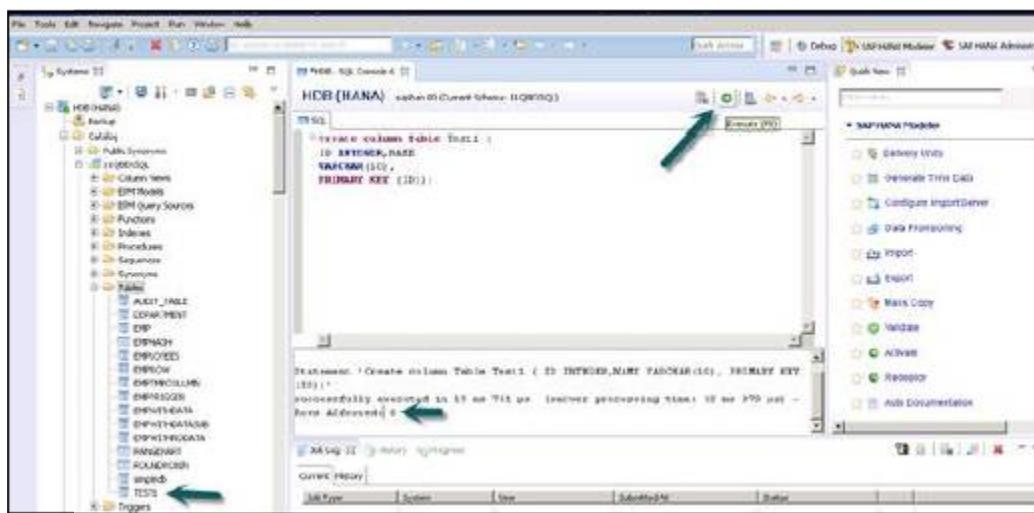
Create column Table Test1 (
    ID INTEGER,
    NAME VARCHAR(10),
    PRIMARY KEY (ID)
);
    
```

In this SQL statement, we have created a Column table "Test1", defined data types of table and Primary Key.

Once you write Create table SQL query, click on Execute option on top of SQL editor right side. Once the statement is executed, we will get a confirmation message as shown in snapshot given below –

Statement 'Create column Table Test1 IDINTEGER, NAMEVARCHAR(10, PRIMARY KEY ID)'

successfully executed in 13 ms 761 μs serverprocessingtime : 12ms979μs – Rows Affected: 0



Execution statement also tells about the time taken to execute the statement. Once statement is successfully executed, right click on Table tab under Schema name in System View and refresh. New Table will be reflected in the list of tables under Schema name.

Insert statement is used to enter the data in the Table using SQL editor.

```
Insert into TEST1 Values (1, 'ABCD')
Insert into TEST1 Values (2, 'EFGH');
```

Click on Execute.

You can right click on Table name and use Open Data Definition to see data type of the table. Open Data Preview/Open Content to see table contents.

Creating Table using GUI Option

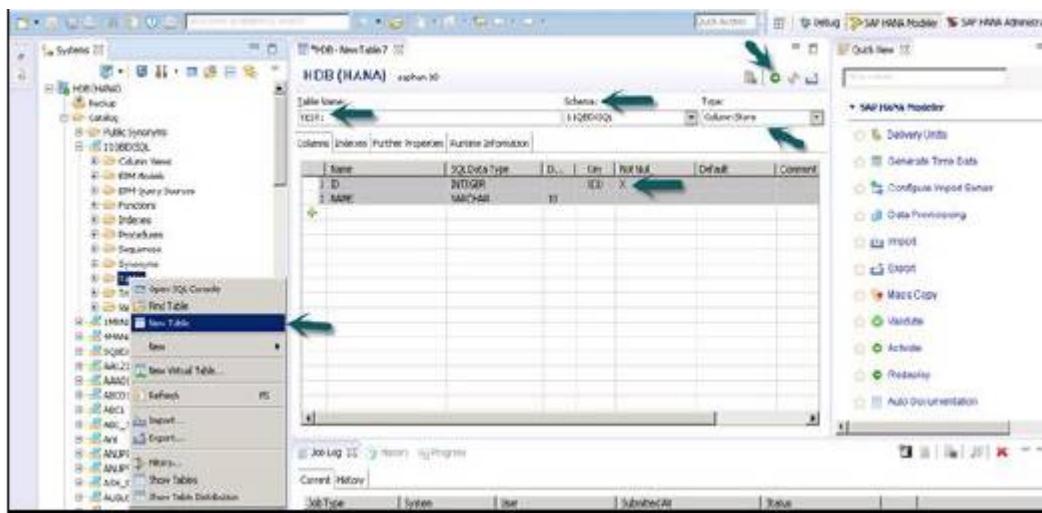
Another way to create a table in HANA database is by using GUI option in HANA Studio.

Right Click on Table tab under Schema → Select 'New Table' option as shown in snapshot given below.

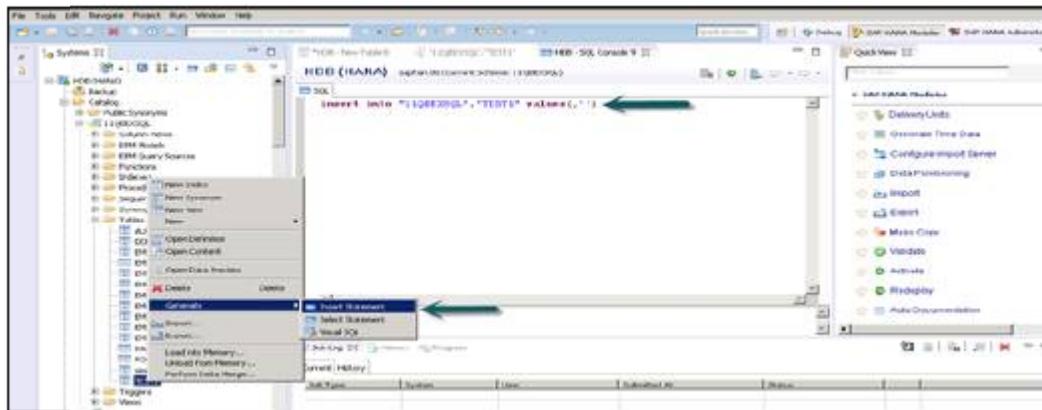
Once you click on New Table → It will open a window to enter the Table name, Choose Schema name from drop down, Define Table type from drop down list: Column Store or Row Store.

Define data type as shown below. Columns can be added by clicking on + sign, Primary Key can be chosen by clicking on cell under Primary key in front of Column name, Not Null will be active by default.

Once columns are added, click on Execute.



Once you Execute F8, Right Click on Table Tab → Refresh. New Table will be reflected in the list of tables under chosen Schema. Below Insert Option can be used to insert data in table. Select statement to see content of table.



Inserting Data in a table using GUI in HANA studio

You can right click on Table name and use Open Data Definition to see data type of the table. Open Data Preview/Open Content to see table contents.

To use tables from one Schema to create views we should provide access on the Schema to the default user who runs all the Views in HANA Modeling. This can be done by going to SQL editor and running this query –

GRANT SELECT ON SCHEMA "<SCHEMA_NAME>" TO _SYS_REPO WITH GRANT OPTION

SAP HANA - PACKAGES

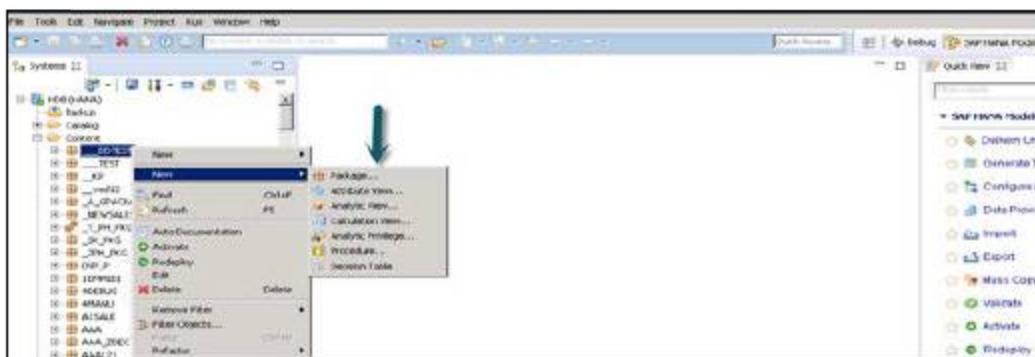
SAP HANA Packages are shown under Content tab in HANA studio. All HANA modeling is saved inside Packages.

You can create a new Package by Right Click on Content Tab → New → Package





You can also create a Sub Package under a Package by right clicking on the Package name. When we right click on the Package we get 7 Options: We can create HANA Views Attribute Views, Analytical Views, and Calculation Views under a Package.



You can also create Decision Table, Define Analytic Privilege and create Procedures in a Package.

When you right click on Package and click on New, you can also create sub packages in a Package. You have to enter Package Name, Description while creating a Package.

SAP HANA - ATTRIBUTE VIEW

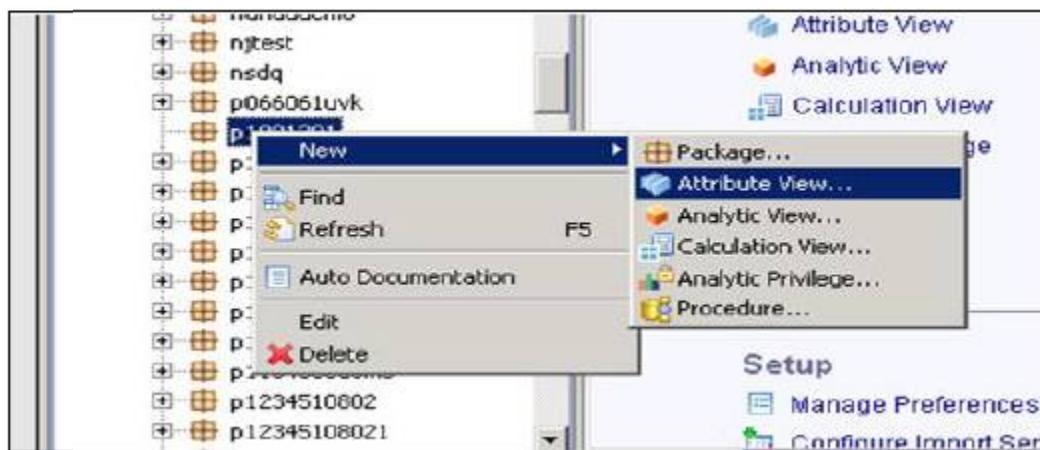
Attribute Views in SAP HANA Modeling are created on the top of Dimension tables. They are used to join Dimension tables or other Attribute Views. You can also copy a new Attribute View from already existing Attribute Views inside other Packages but that doesn't let you change the View Attributes.

Characteristics of Attribute View

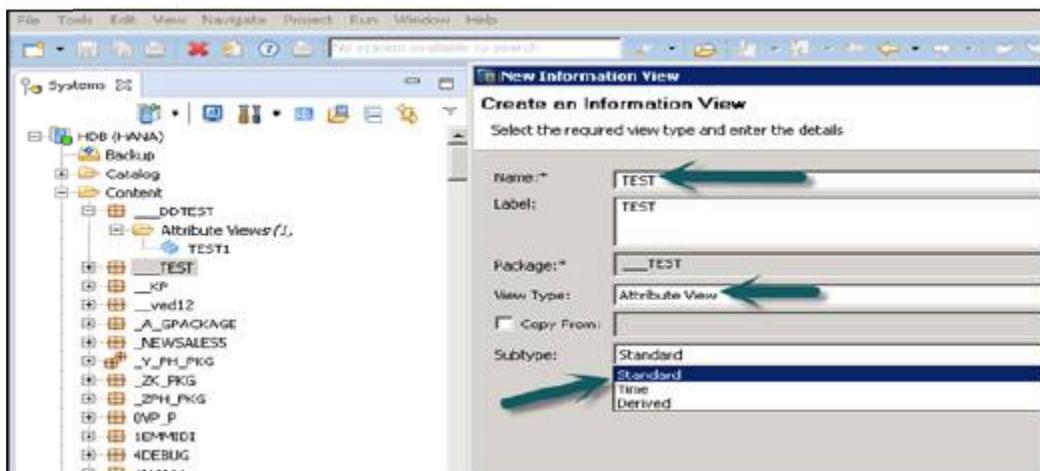
- Attribute Views in HANA are used to join Dimension tables or other Attribute Views.
- Attribute Views are used in Analytical and Calculation Views for analysis to pass master data.
- They are similar to Characteristics in BM and contain master data.
- Attribute Views are used for performance optimization in large size Dimension tables, you can limit the number of attributes in an Attribute View which are further used for Reporting and analysis purpose.
- Attribute Views are used to model master data to give some context.

How to Create an Attribute View?

Choose the Package name under which you want to create an Attribute View. Right Click on Package → Go to New → Attribute View



When you click on Attribute View, New Window will open. Enter Attribute View name and description. From the drop down list, choose View Type and sub type. In sub type, there are three types of Attribute views – Standard, Time, and Derived.

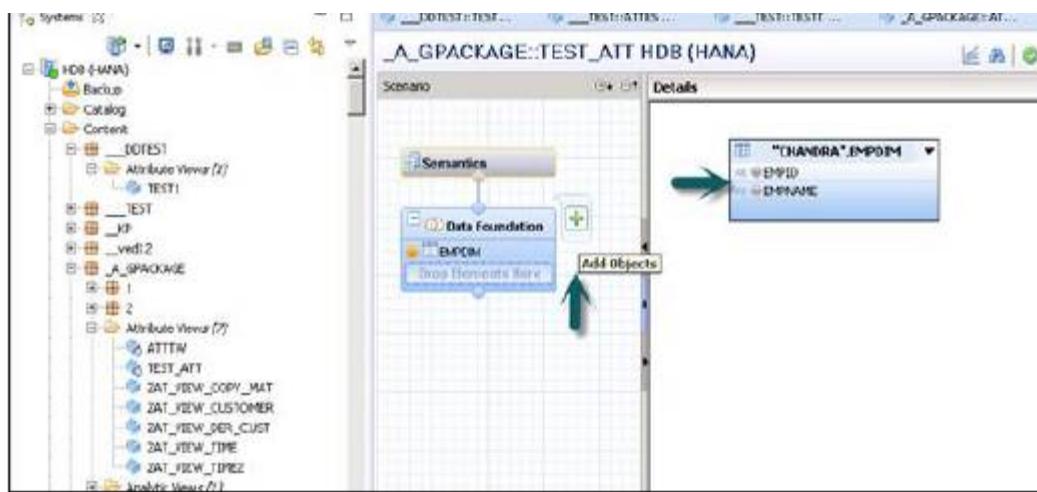


Time subtype Attribute View is a special type of Attribute view that adds a Time Dimension to Data Foundation. When you enter the Attribute name, Type and Subtype and click on Finish, it will open three work panes –

- Scenario pane that has Data Foundation and Semantic Layer.
- Details Pane shows attribute of all tables added to Data Foundation and joining between them.
- Output pane where we can add attributes from Detail pane to filter in the report.

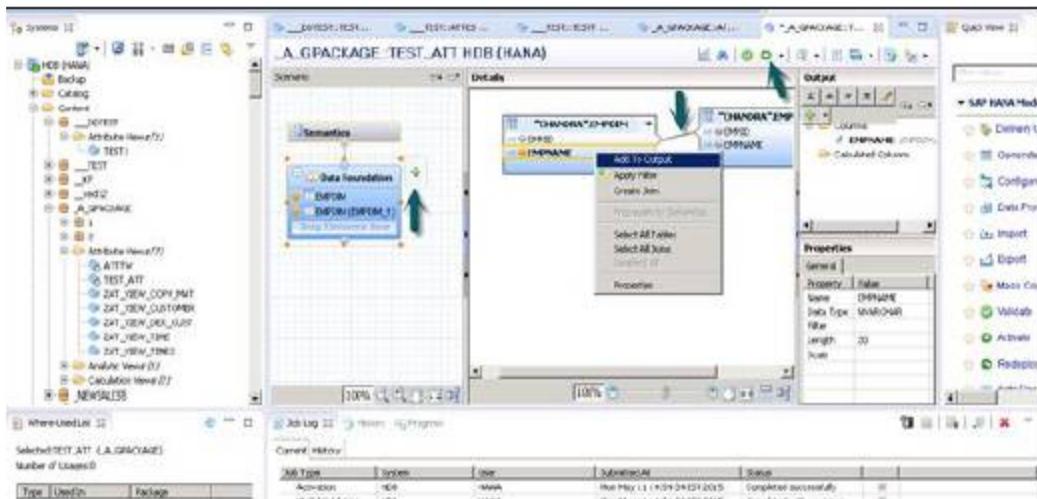
You can add Objects to Data Foundation, by clicking on '+' sign written next to Data Foundation. You can add multiple Dimension tables and Attribute Views in the Scenario Pane and join them using a Primary Key.

When you click on Add Object in Data Foundation, you will get a search bar from where you can add Dimension tables and Attribute views to Scenario Pane. Once Tables or Attribute Views are added to Data Foundation, they can be joined using a Primary Key in Details Pane as shown below.



Once joining is done, choose multiple attributes in details pane, right click and Add to Output. All columns will be added to Output pane. Now Click on Activate option and you will get a confirmation message in job log.

Now you can right click on the Attribute View and go for Data Preview.



Note – When a View is not activated, it has diamond mark on it. However, once you activate it, that diamond disappears that confirms that View has been activated successfully.

Once you click on Data Preview, it will show all the attributes that has been added to Output pane under Available Objects.

These Objects can be added to Labels and Value axis by right click and adding or by dragging the objects as shown below –





SAP HANA - ANALYTIC VIEW

Analytic View is in the form of Star schema, wherein we join one Fact table to multiple Dimension tables. Analytic views use real power of SAP HANA to perform complex calculations and aggregate functions by joining tables in form of star schema and by executing Star schema queries.

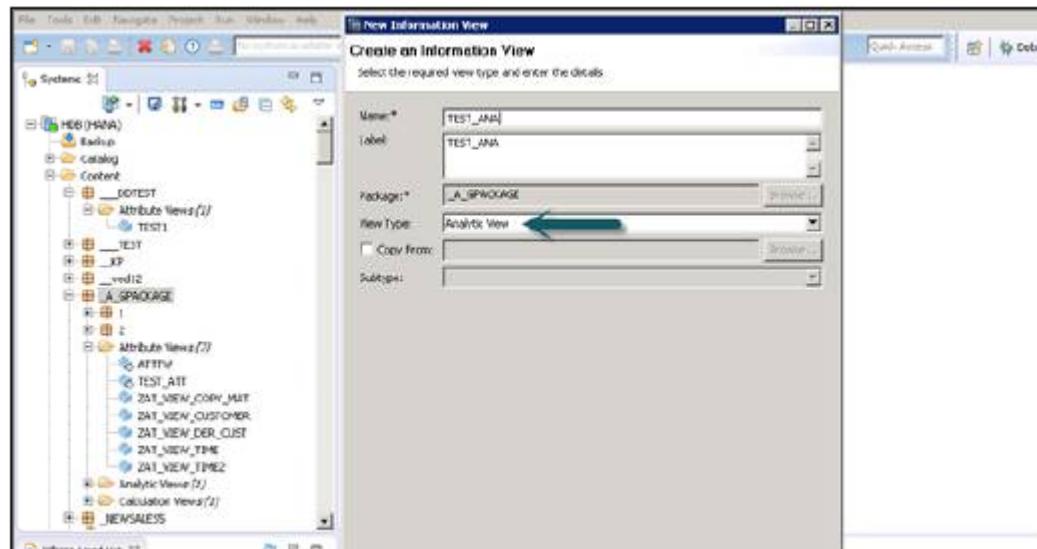
Characteristics of Analytic View

Following are the properties of SAP HANA Analytic View –

- Analytic Views are used to perform complex calculations and Aggregate functions like Sum, Count, Min, Max, Etc.
- Analytic Views are designed to run Start schema queries.
- Each Analytic View has one Fact table surrounded by multiple dimension tables. Fact table contains primary key for each Dim table and measures.
- Analytic Views are similar to Info Objects and Info sets of SAP BW.

How to Create an Analytic View?

Choose the Package name under which you want to create an Analytic View. Right Click on Package → Go to New → Analytic View. When you click on an Analytic View, New Window will open. Enter View name and Description and from drop down list choose View Type and Finish.

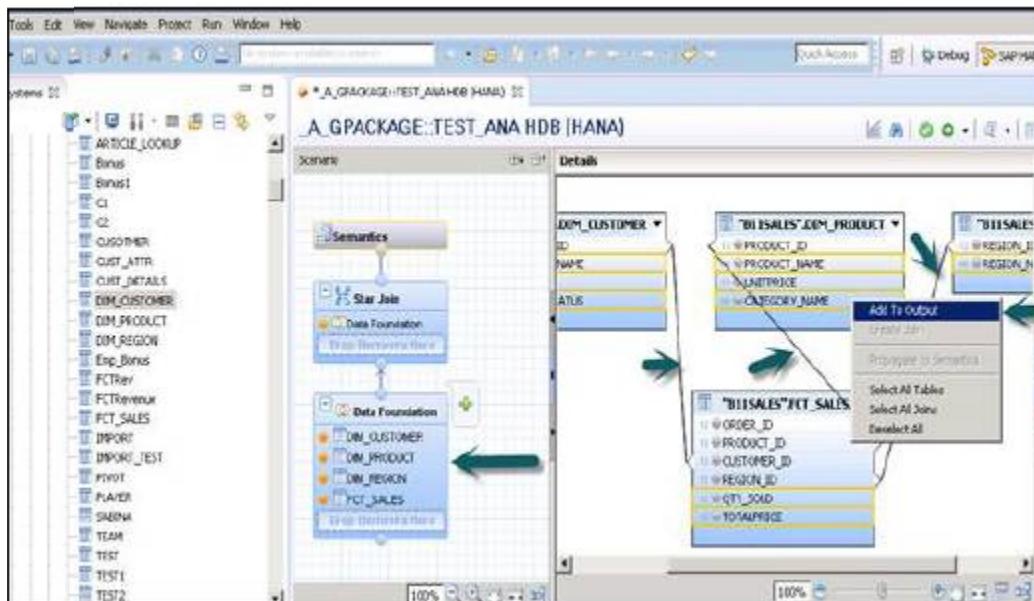




When you click Finish, you can see an Analytic View with Data Foundation and Star Join option.

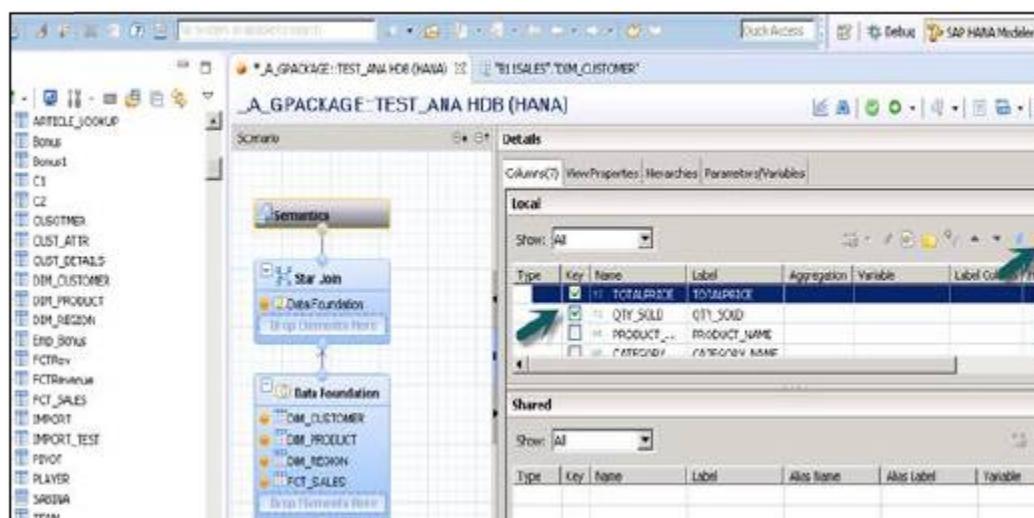
Click on Data Foundation to add Dimension and Fact tables. Click on Star Join to add Attribute Views.

Add Dim and Fact tables to Data Foundation using “+” sign. In the example given below, 3 dim tables have been added: DIM_CUSTOMER, DIM_PRODUCT, DIM_REGION and 1 Fact table FCT_SALES to Details Pane. Joining Dim table to Fact table using Primary Keys stored in Fact table.



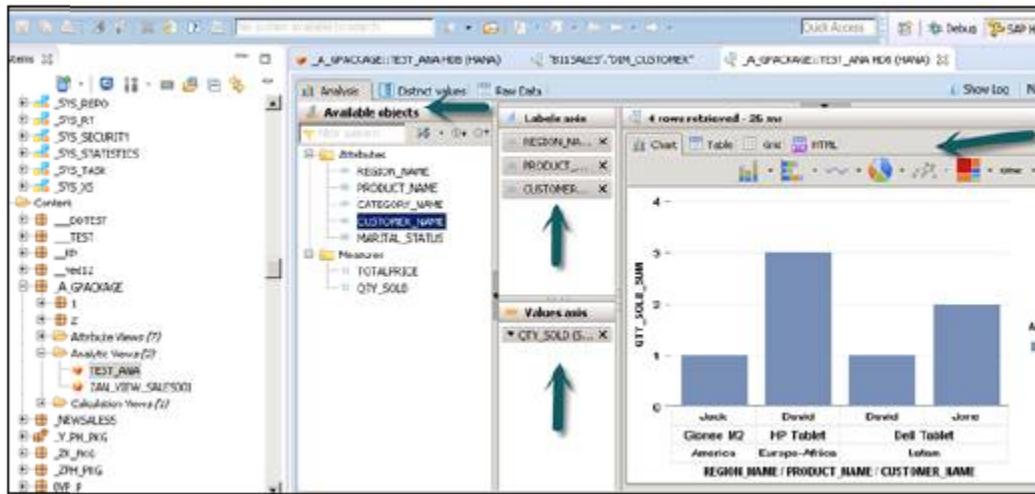
Select Attributes from Dim and Fact table to add to Output pane as shown in snapshot shown above. Now change the data type of Facts, from fact table to measures and Activate the View.

Click on Semantic layer, choose facts and click on measures sign as shown below to change datatype to measures and Activate the View.



Once you activate view and click on Data Preview, all attributes and measures will be added under the list of Available objects. Add Attributes to Labels Axis and Measure to Value axis for analysis purpose.

There is an option to choose different types of chart and graphs.



SAP HANA - CALCULATION VIEW

Calculation Views are used to consume other Analytic, Attribute and other Calculation views and base column tables. These are used to perform complex calculations, which are not possible with other type of Views.

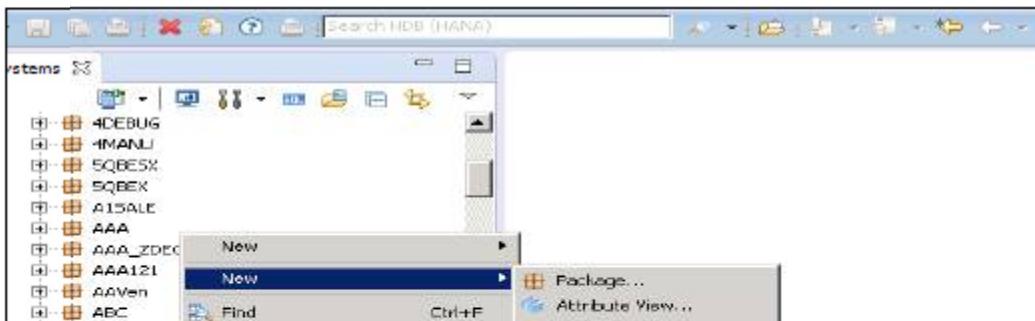
Characteristics of Calculation View

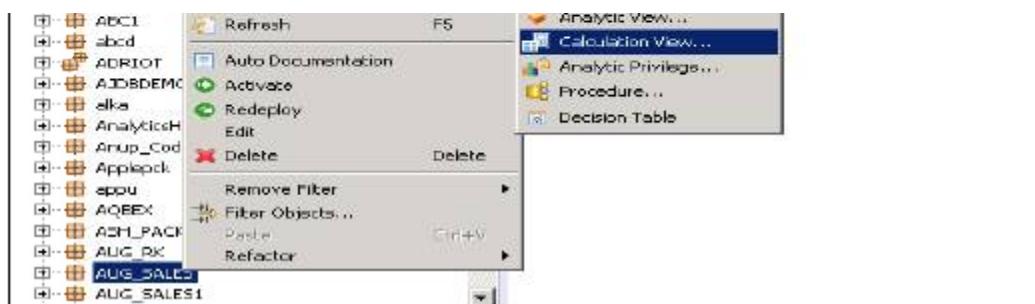
Below given are few characteristics of Calculation Views –

- Calculation Views are used to consume Analytic, Attribute and other Calculation Views.
- They are used to perform complex calculations, which are not possible with other Views.
- There are two ways to create Calculation Views- SQL Editor or Graphical Editor.
- Built-in Union, Join, Projection & Aggregation nodes.

How to create a Calculation View?

Choose the Package name under which you want to create a Calculation View. Right Click on Package → Go to New → Calculation View. When you click on Calculation View, New Window will open.





Enter View name, Description and choose view type as Calculation View, Subtype Standard or Time this is special kind of View which adds time dimension. You can use two types of Calculation View – Graphical and SQL Script.

Graphical Calculation Views

It has default nodes like aggregation, Projection, Join and Union. It is used to consume other Attribute, Analytic and other Calculation views.

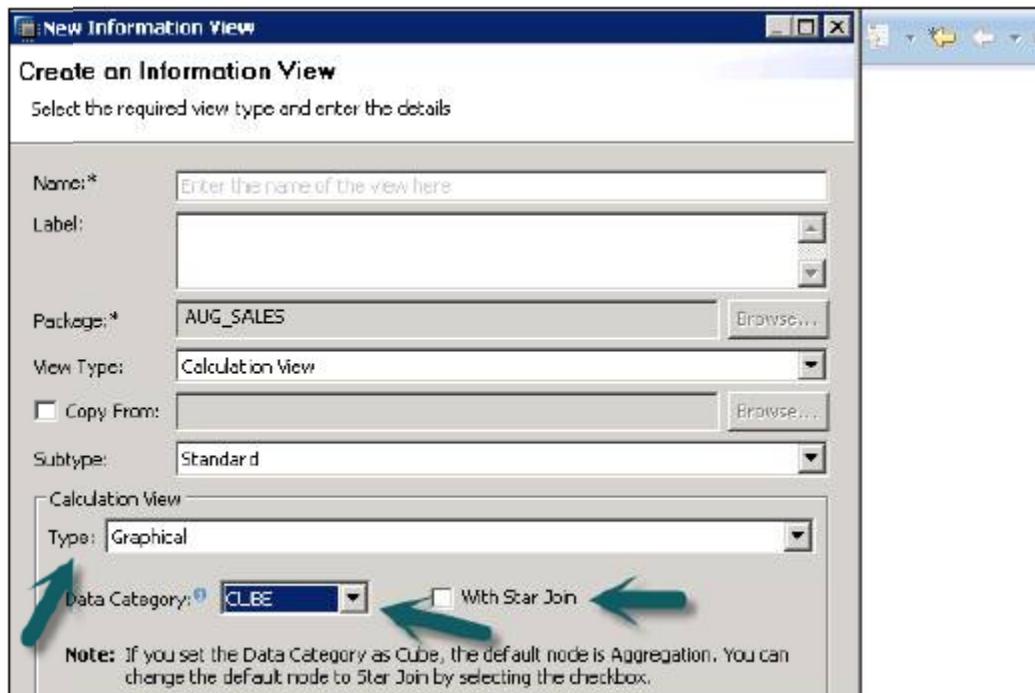
SQL Script based Calculation Views

It is written in SQL scripts that are built on SQL commands or HANA defined functions.

Data Category

Cube, in this default node, is Aggregation. You can choose Star join with Cube dimension.

Dimension, in this default node is Projection.



Calculation View with Star Join

It does not allow base column tables, Attribute Views or Analytic views to add at data foundation. All Dimension tables must be changed to Dimension Calculation views to use in Star Join. All Fact tables can be added and can use default nodes in Calculation View.

Example

The following example shows how we can use Calculation View with Star join –

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) ,CALLEYEAR 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

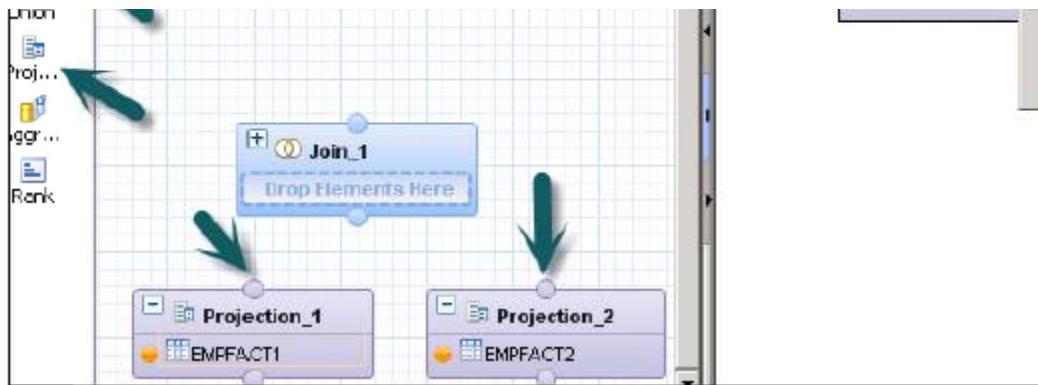
```
Create column table Empfact1 (empId nvarchar(3), Empdate date, Sal integer );
Insert into Empfact1 values('AA1','20100101',5000);
Insert into Empfact1 values('BB1','20110101',10000);
Insert into Empfact1 values('CC1','20120101',12000);
```

```
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);
```

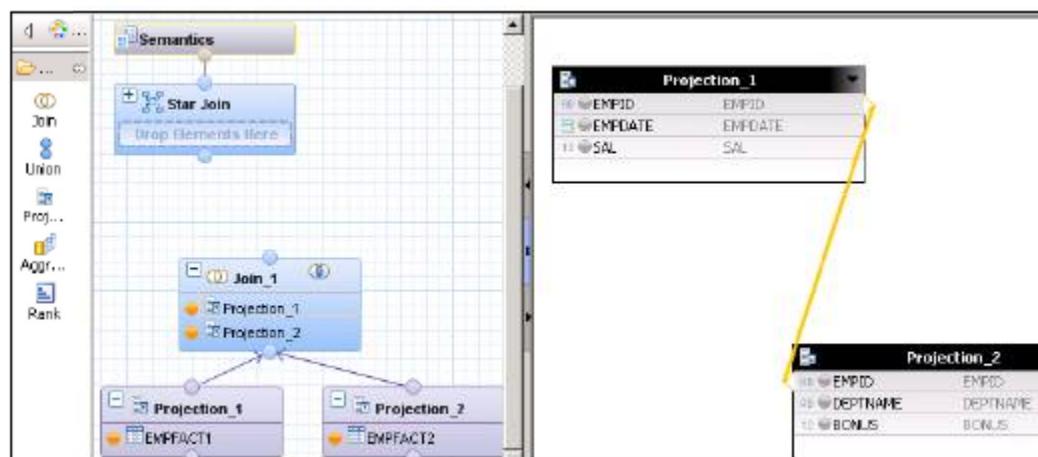
Now we have to implement Calculation View with Star Join. First change both Dim tables to Dimension Calculation View.

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.

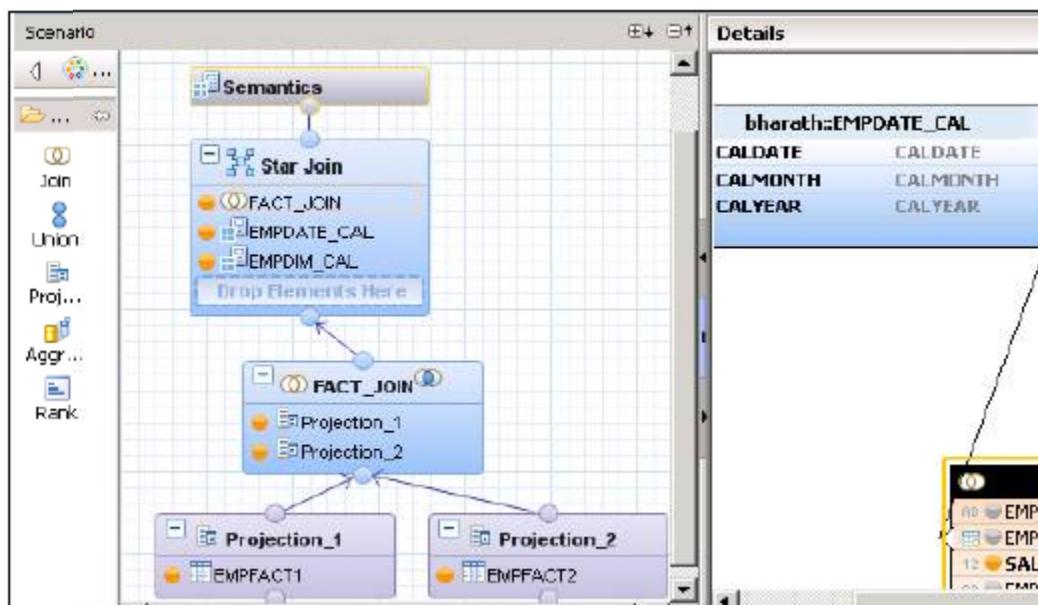




Add a join from default node and join both the fact tables. Add parameters of Fact Join to output pane.



In Star Join, add both- Dimension Calculation views and add Fact Join to Star Join as shown below. Choose parameters in Output pane and active the View.



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.

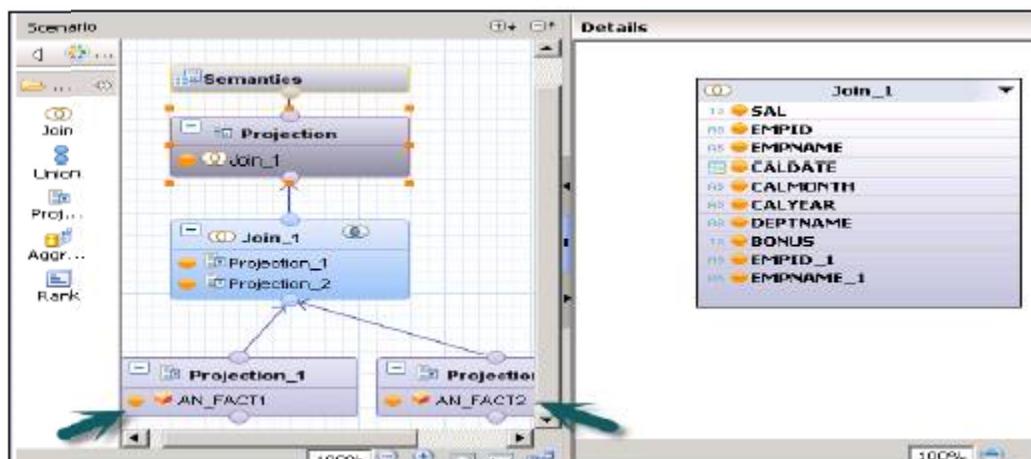
Calculation View without Star Join

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.



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

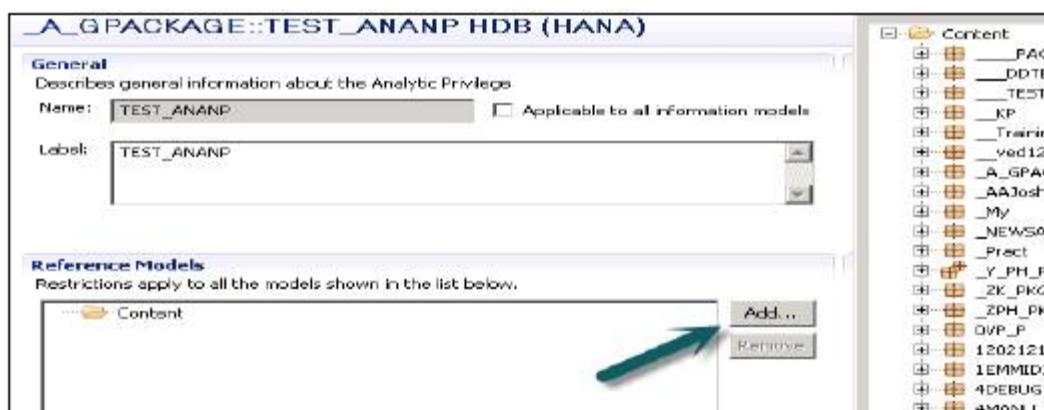
Right Click on Package name and go to new Analytic Privilege or you can open using HANA Modeler quick launch.



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.

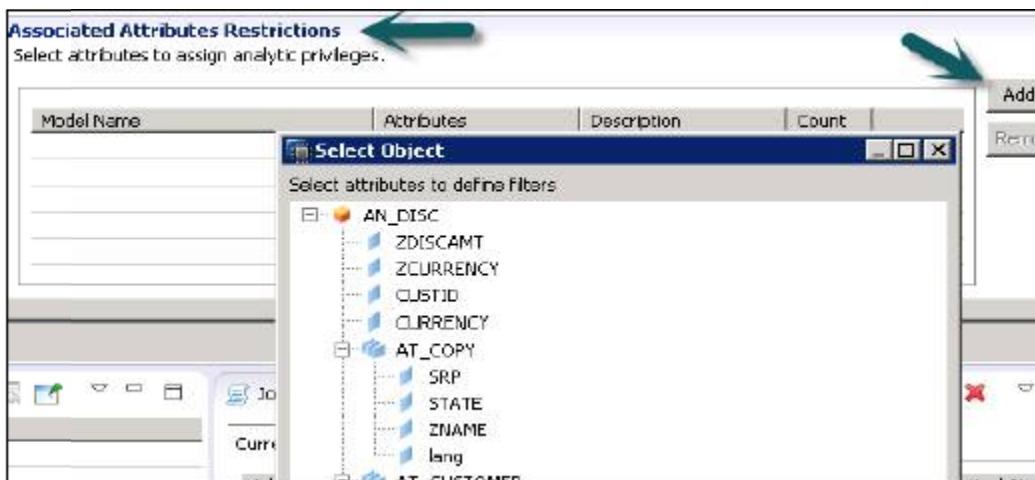
Once you click on Add button, it will show you all the views under Content tab.





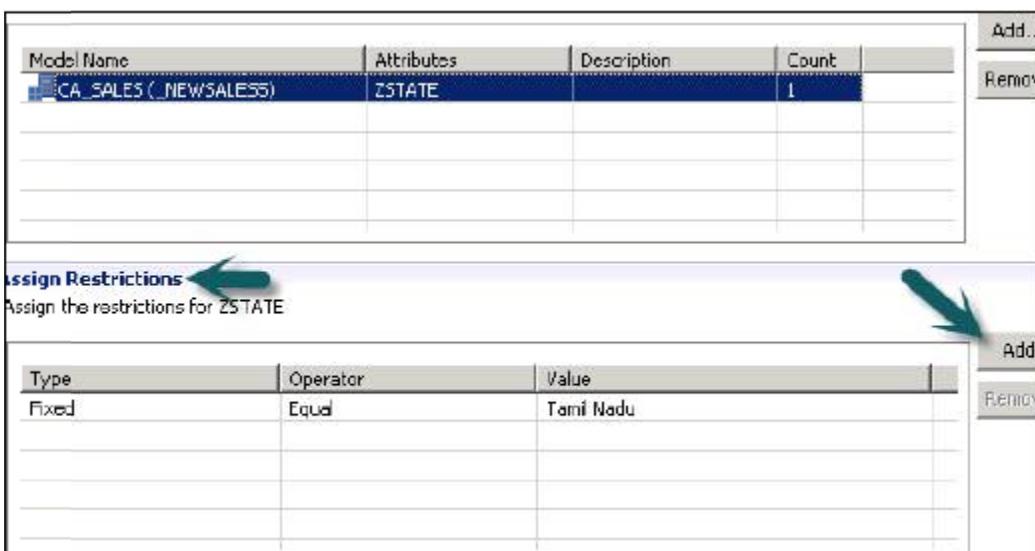
Choose View that you want to add to Analytic Privilege package and click OK. Selected View will be added under reference models.

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



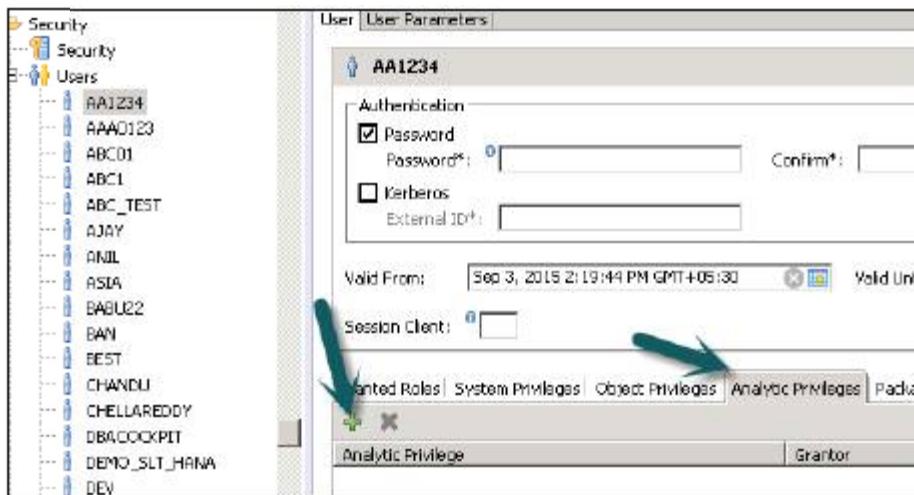
Add objects you want to add to Analytic privileges from select object option and click on OK.

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.



We have to activate Analytic Privilege now, by clicking on Green round icon at top. Status message – completed successfully confirms activation successfully under job log and we can use this view now by adding to a role.

Now to add this role to a user, go to security tab → User → Select User on which you want to apply these Analytic privileges.



Search Analytic Privilege you want to apply with the name and click on OK. That view will be added to user role under Analytic Privileges.

To delete Analytic Privileges from specific user, select view under tab and use Red delete option. Use Deploy arrowmark at top or F8 to apply this to user profile.

SAP HANA - INFORMATION COMPOSER

SAP HANA Information Composer is a self-service modeling environment for end users to analyze data set. It allows you to import data from workbook format *.xls*, *.csv* into HANA database and to create Modeling views for analysis.

Information Composer is very different from HANA Modeler and both are designed to target separate set of users. Technically sound people who have strong experience in data modeling use HANA Modeler. A business user, who does not have any technical knowledge, uses Information Composer. It provides simple functionalities with easy to use interface.

Features of Information Composer

- Data extraction** – Information Composer helps to extract data, clean data, preview data and automate the process of creation of physical table in the HANA database.
- Manipulating data** – It helps us to combine two objects *Physicaltables*, *AnalyticalView*, *attributeviewandcalculationviews* and create information view that can be consumed by SAP BO Tools like SAP Business Objects Analysis, SAP Business Objects Explorer and other tools like MS Excel.
- It provides a centralized IT service in the form of URL, which can be accessed from anywhere.

How to upload data using Information Composer?

It allows us to upload large amount of data *upto5millioncells*. Link to access Information Composer –

<http://<server>:<port>/IC>

Login to SAP HANA Information Composer. You can perform data loading or manipulation using this tool.

To upload data this can be done in two ways –

- Uploading .xls, .csv file directly to HANA database
- Other way is to copy data to clipboard and copy from there to HANA database.
- It allows data to be loaded along with header.

On Left side in Information Composer, you have three options –

Select Source of data → Classify data → Publish



Once data is published to HANA database, you cannot rename the table. In this case, you have to delete the table from Schema in HANA database.

“SAP_IC” schema, where tables like IC_MODELS, IC_SPREADSHEETS exists. One can find details of tables created using IC under these tables.

Raw Data					
Distinct values					
Analysis					
SPREADSHEET_ID	CREATED_BY	STATUS	TABLE_NAME	SCHEMA_NAME	
2864996dce011e186d1...	P1007547	PUBLISHED	IC/EMP	IC_TABLES	EMP
2864996dce011e1956f...	P1007547	PUBLISHED	IC/EMP1	IC_TABLES	EMP1
421adf81774c11e18c25...	P1007547	PUBLISHED	IC/COUNTRY	IC_TABLES	COUNT
5619b5b77cd11e191b0...	S0000012557	SAVED	IC/S0000012557/sa...	IC_TABLES	

Using Clipboard

Another way to upload data in IC is by use of the clipboard. Copy the data to clipboard and upload it with help of Information Composer. Information Composer also allows you to see preview of data or even provide summary of data in temporary storage. It has inbuilt capability of data cleansing that is used to remove any inconsistency in data.

Once data is cleansed, you need to classify data whether it is attributed. IC has inbuilt feature to check the data type of uploaded data.

Final step is to publish the data to physical tables in HANA database. Provide a technical name and description of table and this will be loaded inside IC_Tables Schema.

User Roles for using data published with Information Composer

Two set of users can be defined to use data published from IC.

- IC_MODELER is for creating physical tables, uploading data and creating information views.
- IC_PUBLIC allows users to view information views created by other users. This role does not allow the user to upload or create any information views using IC.

System Requirement for Information Composer

Server Requirements –

- At least 2GB of available RAM is required.
- Java 6 64 – bit must be installed on the server.
- The Information Composer Server must be physically located next to the HANA server.

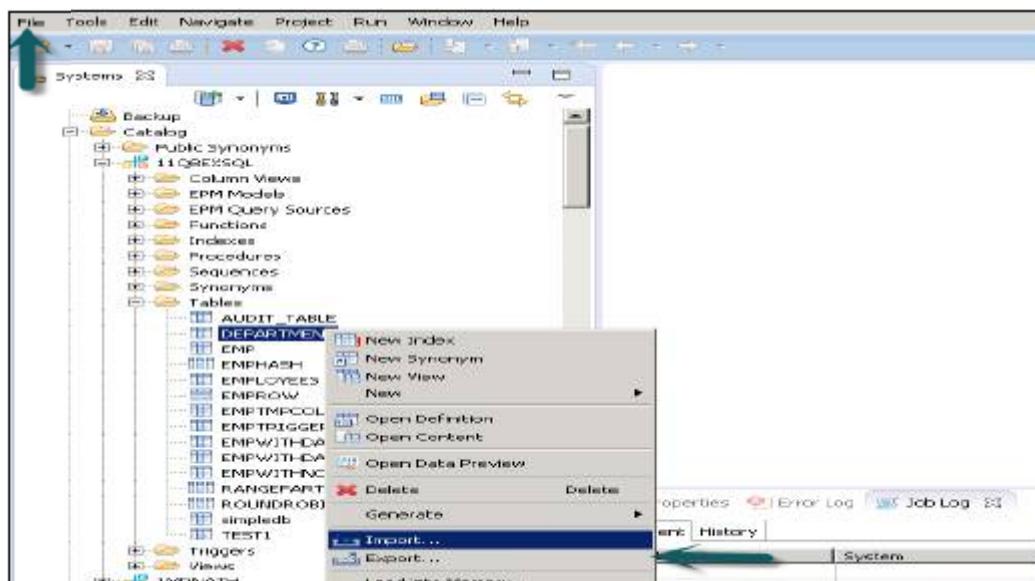
Client Requirements –

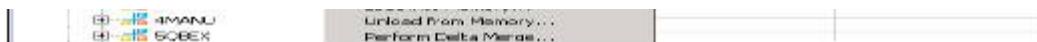
- Internet Explorer with Silverlight 4 installed.

SAP HANA - EXPORT & IMPORT

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 or Information model in HANA studio.





Exporting a table/Information model in HANA Studio

Go to file menu → Export → You will see options as shown below –



Export Options under SAP HANA Content

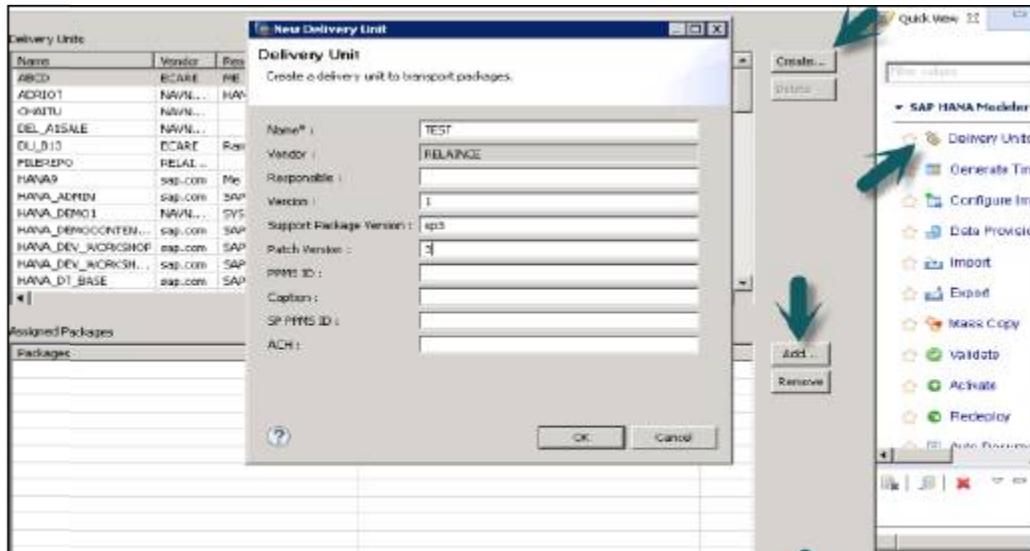
Delivery Unit

Delivery unit is 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.

Users 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.

The user should create Delivery Unit prior to using it.

This can be done through HANA Modeler → Delivery Unit → Select System and Next → Create → Fill the details like Name, Version, etc. → OK → Add Packages to Delivery unit → Finish



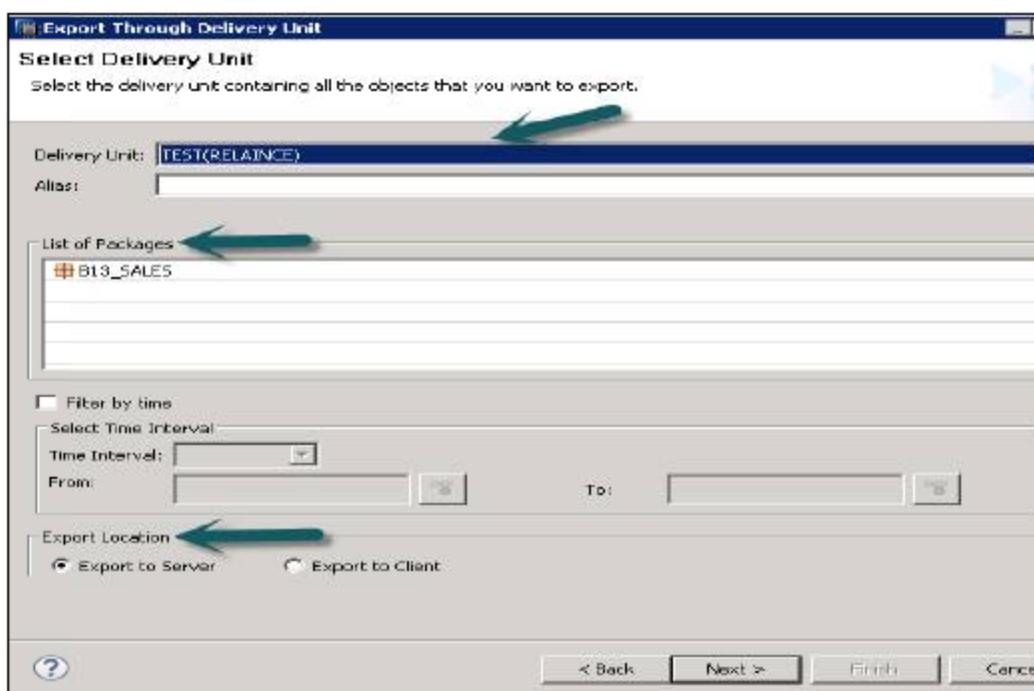


Once the Delivery Unit is created and the packages are assigned to it, user can see the list of packages by using Export option –

Go to File → Export → Delivery Unit → Select the Delivery Unit.

You can see list of all packages assigned to Delivery unit. It gives an option to choose export location –

- Export to Server
- Export to Client



You can export the Delivery Unit either to HANA Server location or to a Client location as shown.

The user can restrict the export through “Filter by time” which means Information views, which are updated within the mentioned time interval will only be exported.

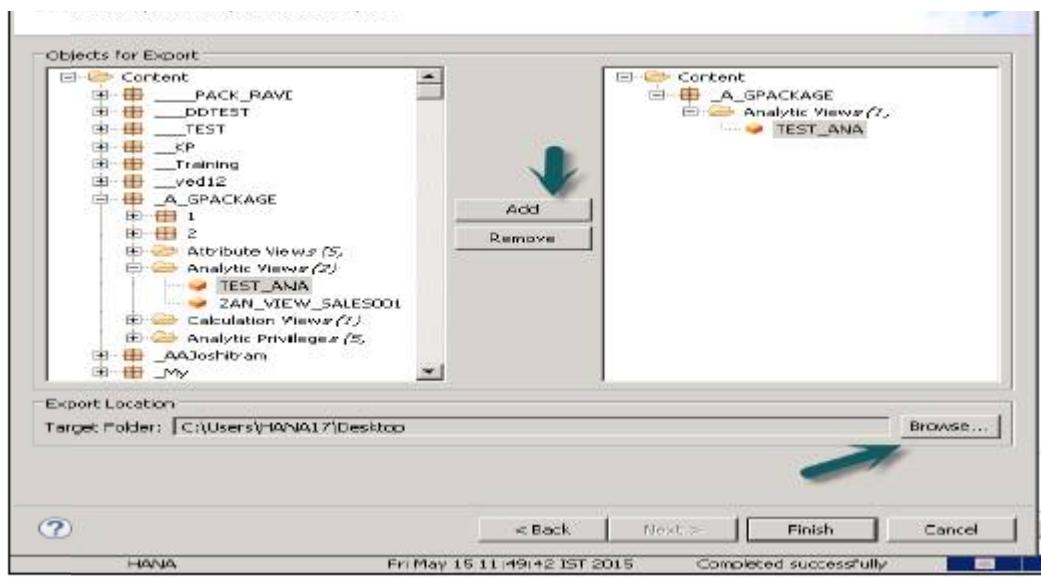
Select the Delivery Unit and Export Location and then Click Next → Finish. This will export the selected Delivery Unit to the specified location.

Developer Mode

This option can be used to export individual objects to a location in the local system. User can select single Information view or group of Views and Packages and select the local Client location for export and Finish.

This is shown in the snapshot below.

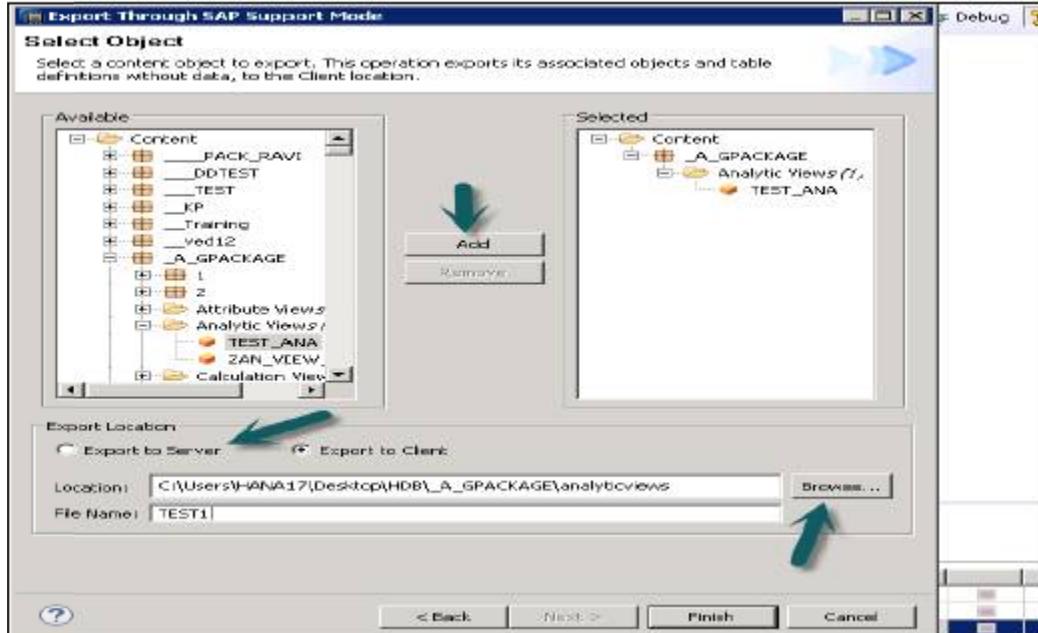




Support Mode

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.



Export Options under SAP HANA Studio –

Landscape – To export the landscape from one system to other.

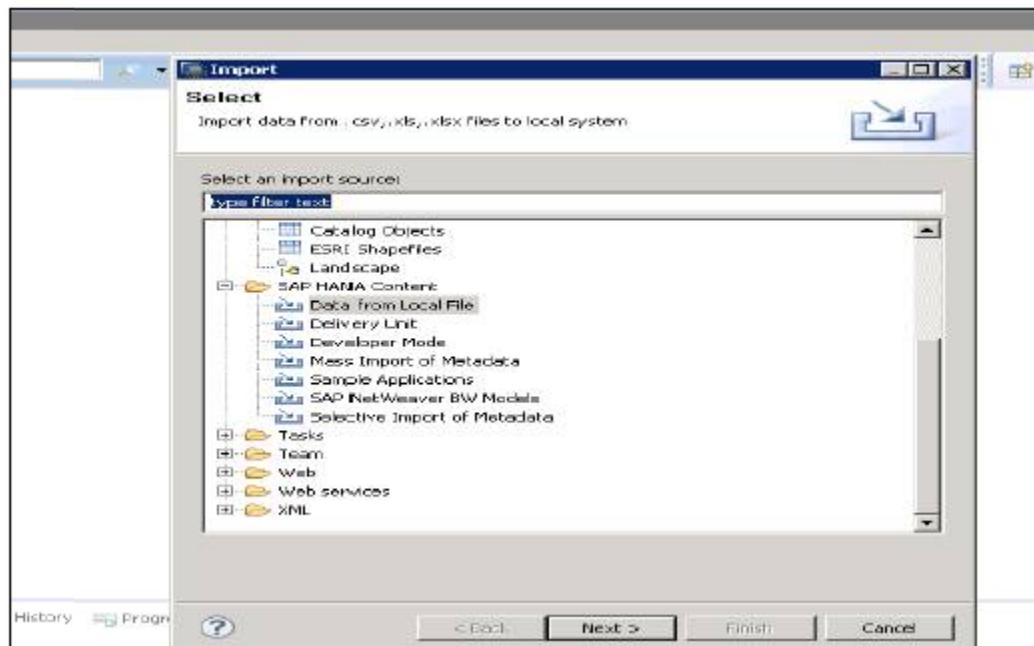
Tables – This option can be used to export tables along with its content.

Import Option under SAP HANA Content

Go to File → Import, You will see all the options as shown below under Import.

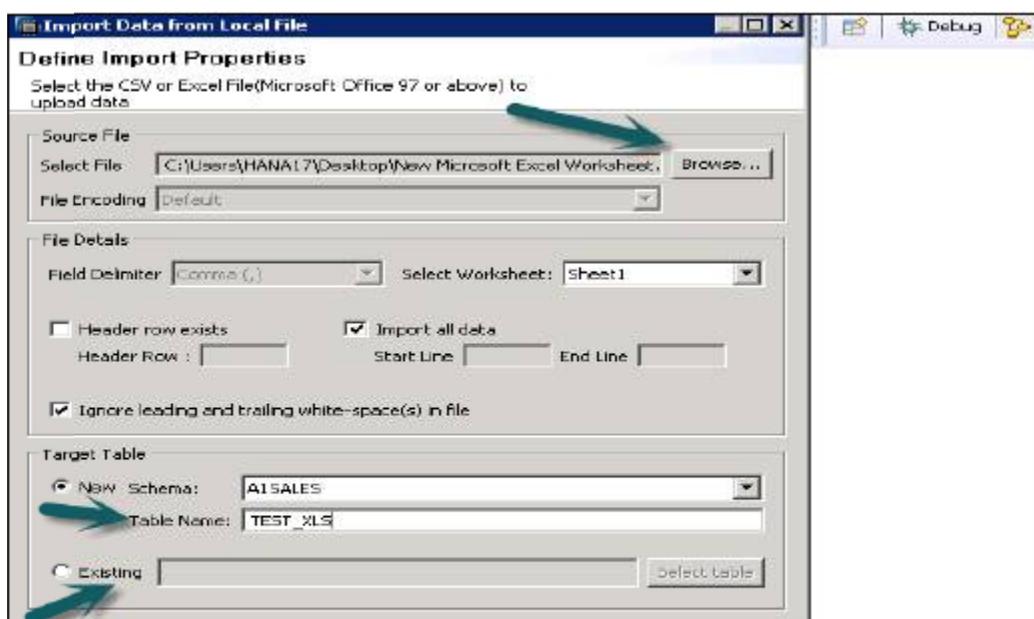
Data from Local File

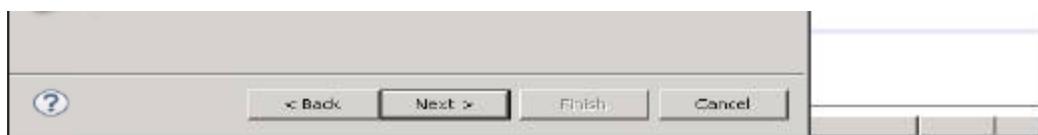
This is used to import data from a flat file like .xls or .csv file.



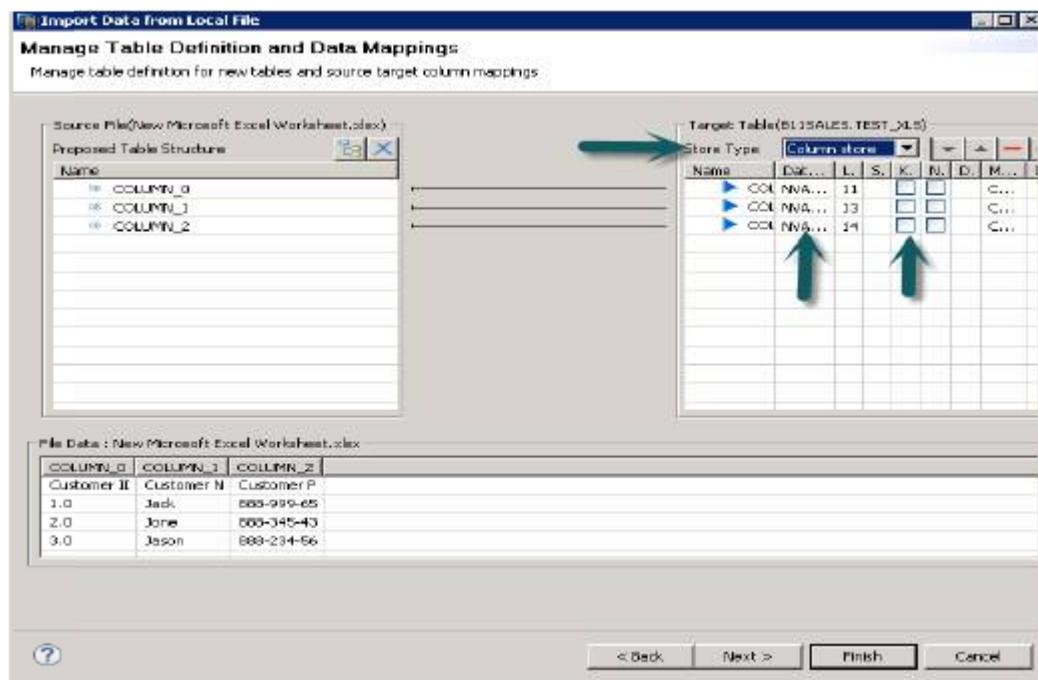
Click on Next → Choose Target System → Define Import Properties

Select Source file by browsing local system. It also gives an option if you want to keep the header row. It also gives an option to create a new table under existing Schema or if you want to import data from a file to an existing table.

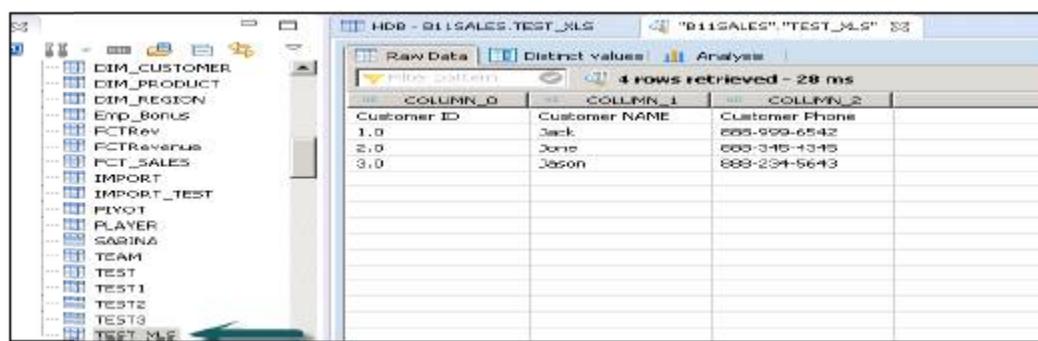




When you click on Next, it gives an option to define Primary Key, change data type of columns, define storage type of table and also, allows you to change the proposed structure of table.



When you click on finish, that table will be populated under list of tables in mentioned Schema. You can do the data preview and can check data definition of the table and it will be same as that of .xls file.

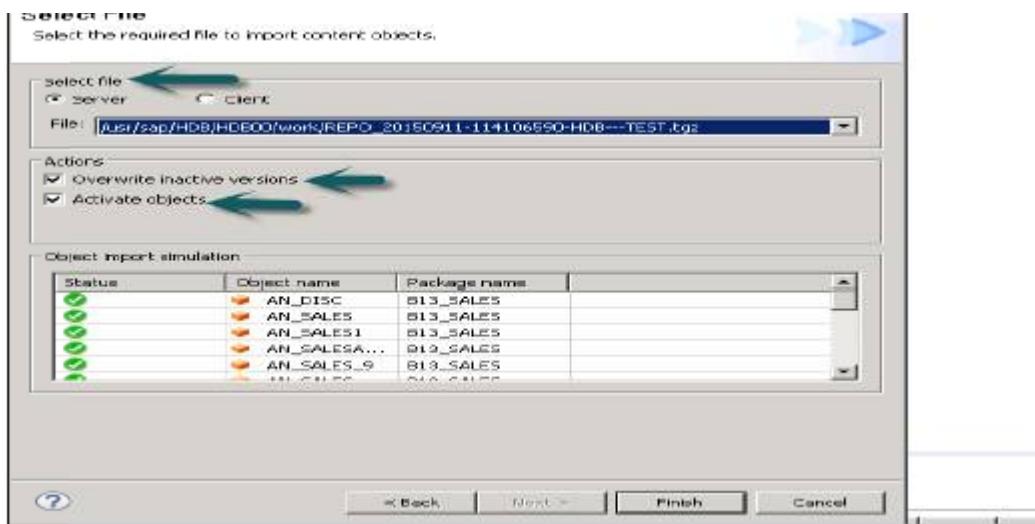


Delivery Unit

Select Delivery unit by going to File → Import → Delivery unit. You can choose from a server or local client.

You can select "Overwrite inactive versions" which allows you to overwrite any inactive version of objects that exist. If the user selects "Activate objects", then after the import, all the imported objects will be activated by default. The user need not trigger the activation manually for the imported views.





Click Finish and once completed successfully, it will be populated to target system.

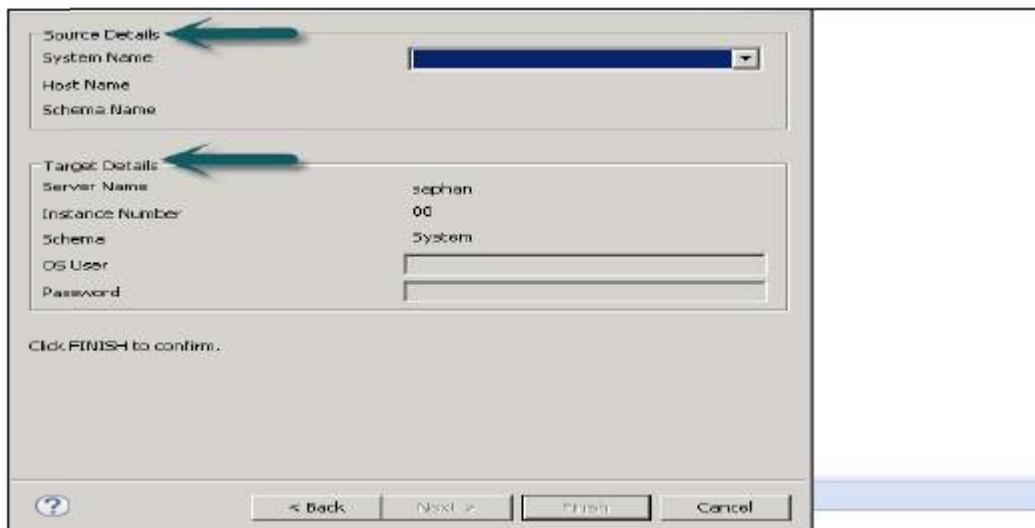
Developer Mode

Browse for the Local Client location where the views are exported and select the views to be imported, the user can select individual Views or group of Views and Packages and Click on Finish.

Mass Import of Metadata

Go to File → Import → Mass Import of Metadata → Next and select the source and target system.

Configure the System for Mass Import and click Finish.

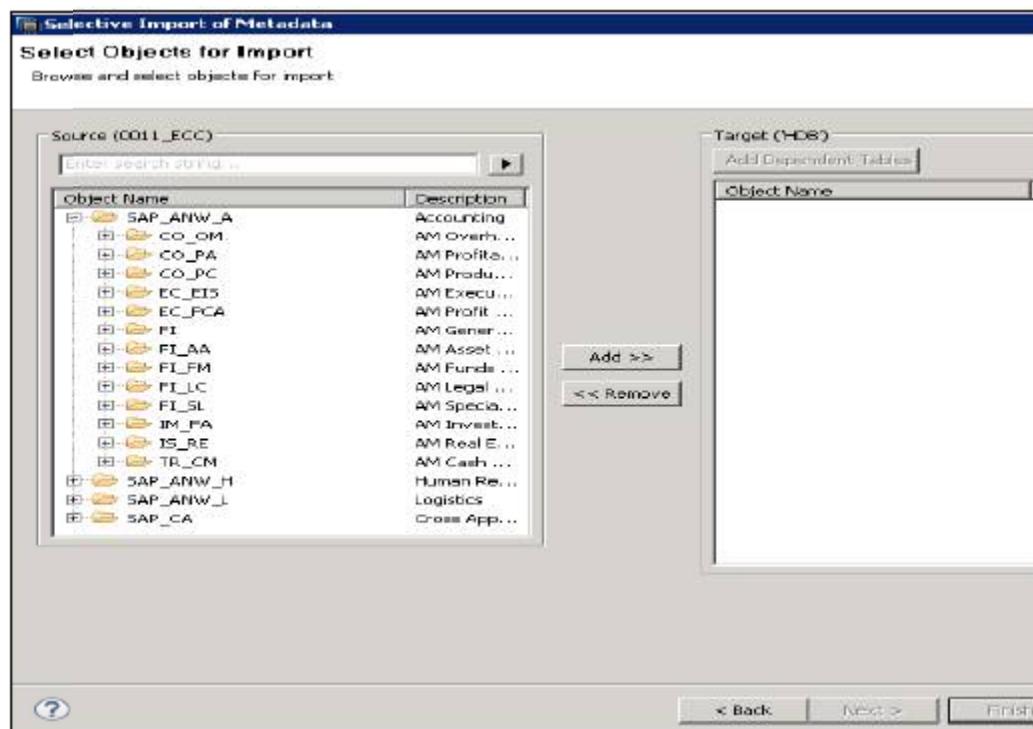
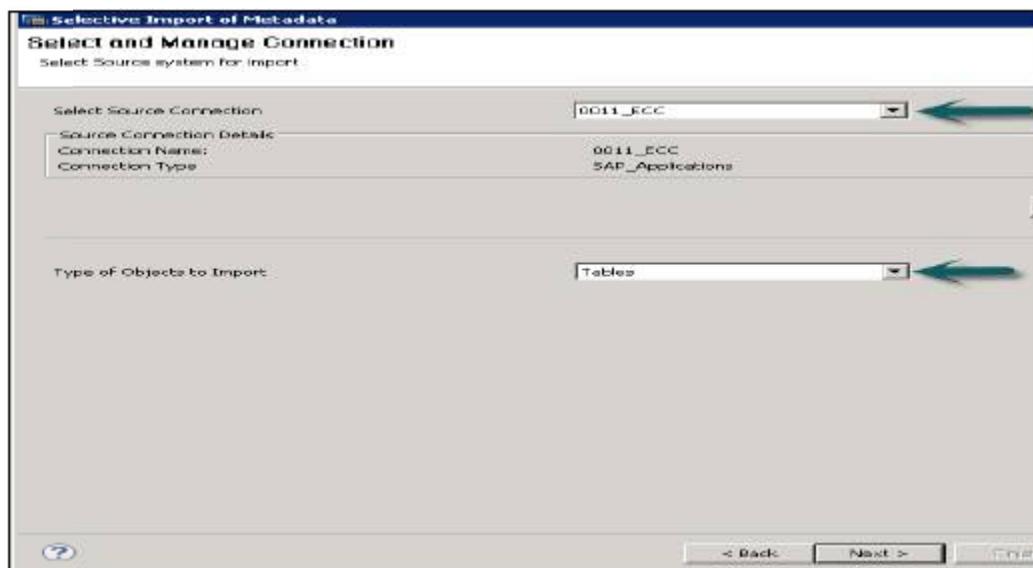


Selective Import of Metadata

It allows you to choose tables and target schema to import Meta data from SAP Applications.

Go to File → Import → Selective Import of Metadata → Next

Choose Source Connection of type "SAP Applications". Remember that the Data Store should have been created already of type SAP Applications → Click Next



Select tables you want to import and validate data if required. Click Finish after that.

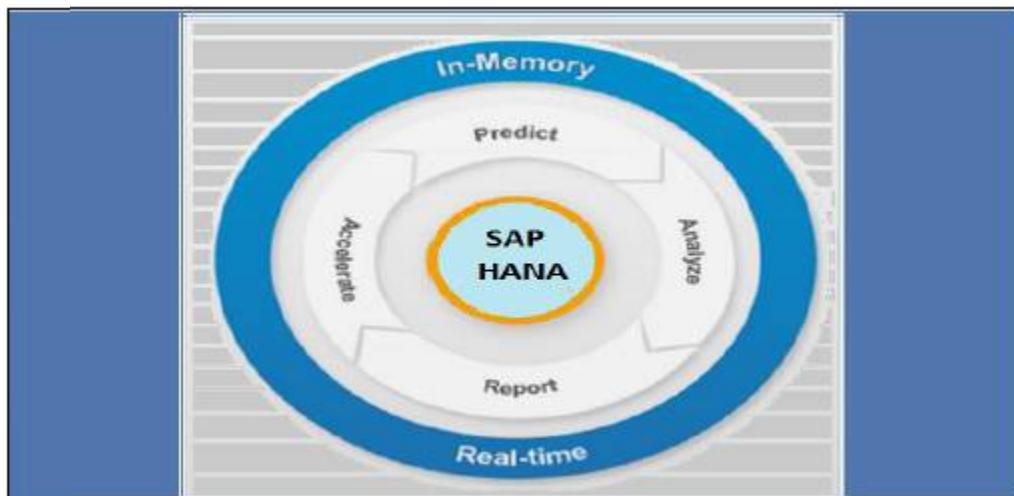
SAP HANA - REPORTING VIEW

We know that with the use of Information Modeling feature in SAP HANA, we can create different Information views Attribute Views, Analytic Views, Calculation views. These Views can be consumed by different reporting tools like SAP Business Object, SAP Lumira, Design Studio, Office Analysis and even third party tool like MS

Excel.

These reporting tools enable Business Managers, Analysts, Sales Managers and senior management employees to analyze the historic information to create business scenarios and to decide business strategy of the company.

This generates the need for consuming HANA Modeling views by different reporting tools and to generate reports and dashboards, which are easy to understand for end users.



In most of the companies, where SAP is implemented, reporting on HANA is done with BI platforms tools that consume both SQL and MDX queries with help of Relational and OLAP connections. There is wide variety of BI tools like – Web Intelligence, Crystal Reports, Dashboard, Explorer, Office Analysis and many more.

BI 4.0 CONNECTIVITY TO HANA VIEWS

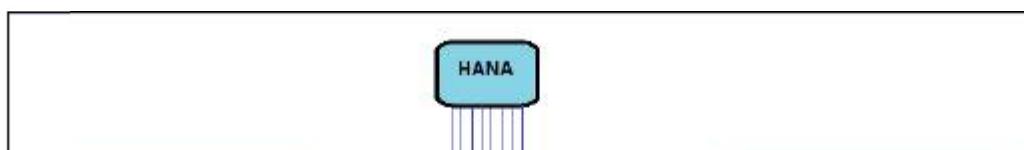
Reporting Tools

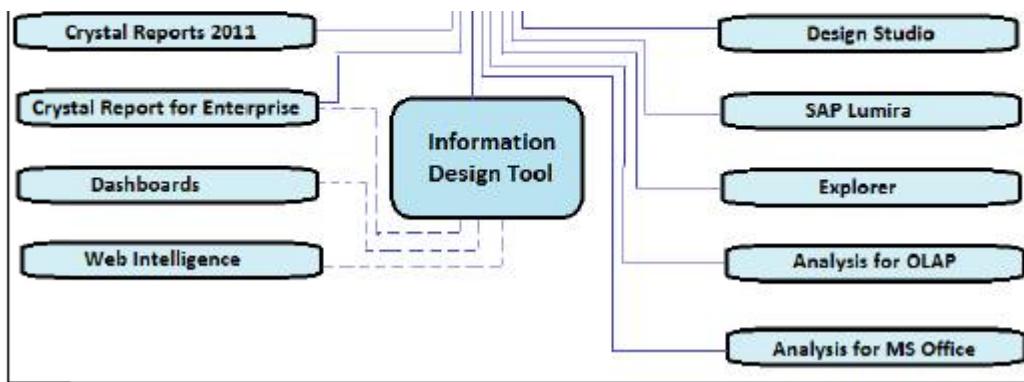
Web Intelligence and Crystal Reports are most common BI tools that are used for reporting. WebI uses a semantic layer called Universe to connect to data source and these Universes are used for reporting in tool. These Universes are designed with the help of Universe design tool UDT or with Information Design tool IDT. IDT supports multisource enabled data source. However, UDT only supports Single source.

Main tools that are used for designing interactive dashboards- Design Studio and Dashboard Designer. Design Studio is future tool for designing dashboard, which consumes HANA views via BI consumer Service BICS connection. Dashboard design *xcelsius* uses IDT to consume schemas in HANA database with a Relational or OLAP connection.

SAP Lumira has an inbuilt feature of directly connecting or loading data from HANA database. HANA views can be directly consumed in Lumira for visualization and creating stories.

Office Analysis uses an OLAP connection to connect to HANA Information views. This OLAP connection can be created in CMC or IDT.





In the picture given above, it shows all BI tools with solid lines, which can be directly connected and integrated with SAP HANA using an OLAP connection. It also depicts tools, which need a relational connection using IDT to connect to HANA are shown with dotted lines.

Relational vs OLAP Connection

The idea is basically if you need to access data from a table or a conventional database then your connection should be a relational connection but if your source is an application and data is stored in cube *multidimensional like Infocubes, Informationmodels* then you would use an OLAP connection.

- A Relational connection can only be created in IDT/UDT.
- An OLAP can be created in both IDT and CMC.

Another thing to note is that a relational connection always produces a SQL statement to be fired from report while an OLAP connection normally creates a MDX statement

Information Design Tool

In Information design tool *IDT*, you can create a relational connection to an SAP HANA view or table using JDBC or ODBC drivers and build a Universe using this connection to provide access to client tools like Dashboards and Web Intelligence as shown in above picture.

You can create a direct connection to SAP HANA using JDBC or ODBC drivers.

SAP HANA - CRYSTAL REPORTS

Crystal Reports for Enterprise

In Crystal Reports for Enterprise, you can access SAP HANA data by using an existing relational connection created using the information design tool.

You can also connect to SAP HANA using an OLAP connection created using information design tool or CMC.

Design Studio

Design Studio can access SAP HANA data by using an existing OLAP connection created in Information design tool or CMC same like Office Analysis.

Dashboards

Dashboards can connect to SAP HANA only through a relational Universe. Customers using Dashboards on top of SAP HANA should strongly consider building their new dashboards with Design Studio.

Web Intelligence

Web Intelligence can connect to SAP HANA only through a Relational Universe.

SAP Lumira

Lumira can connect directly to SAP HANA Analytic and Calculation views. It can also connect to SAP HANA through SAP BI Platform using a relational Universe.

Office Analysis, edition for OLAP

In Office Analysis edition for OLAP, you can connect to SAP HANA using an OLAP connection defined in the Central Management Console or in Information design tool.

Explorer

You can create an information space based on an SAP HANA view using JDBC drivers.

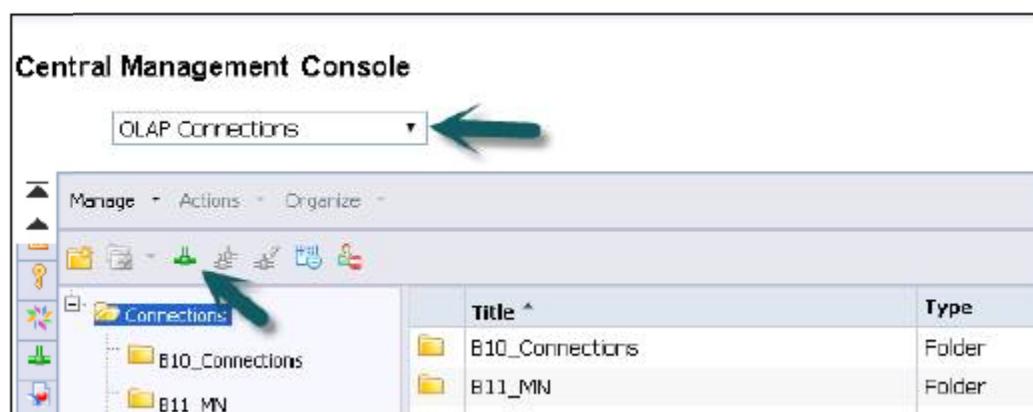
Creating an OLAP Connection in CMC

We can create an OLAP Connection for all the BI tools, which we want to use on top of HANA views like OLAP for analysis, Crystal Report for enterprise, Design Studio. Relational connection through IDT is used to connect Web Intelligence and Dashboards to HANA database.

These connection can be created using IDT as well CMC and both of the connections are saved in BO Repository.

Login to CMC with the user name and password.

From the dropdown list of connections, choose an OLAP connection. It will also show already created connections in CMC. To create a new connection, go to green icon and click on this.



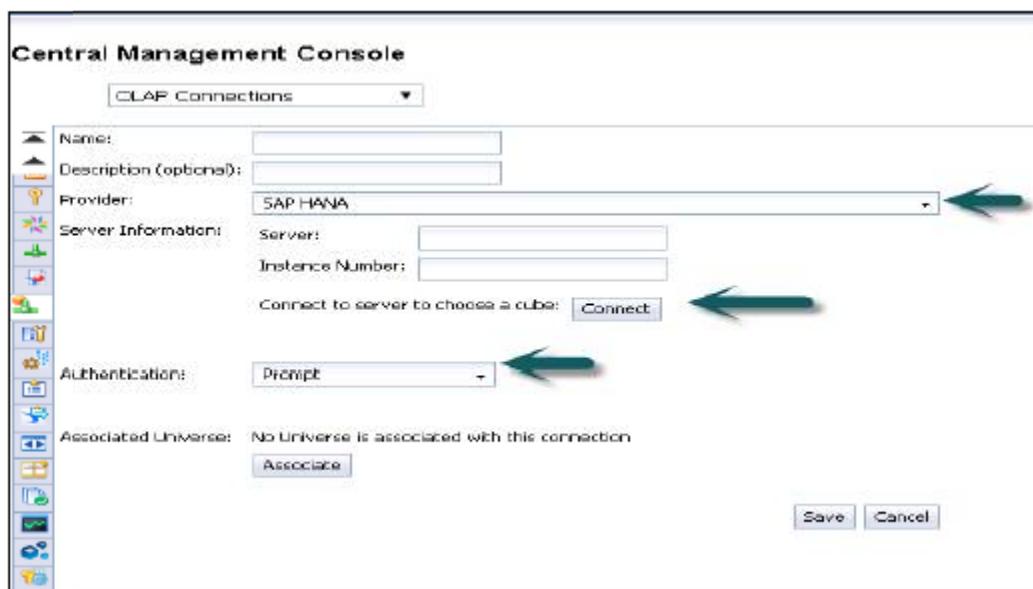
Enter the name of an OLAP connection and description. Multiple persons, to connect to HANA views, in different

BI Platform tools, can use this connection.

Provider – SAP HANA

Server – Enter HANA Server name

Instance – Instance number



It also gives an option to connect to a single Cube

You can also choose to connect to single Analytic or Calculation view or to the full HANA system.

Click on Connect and choose modeling view by entering user name and password.

Authentication Types – Three types of Authentication are possible while creating an OLAP connection in CMC.

- **Predefined** – It will not ask user name and password again while using this connection.
- **Prompt** – Every time it will ask user name and password
- **SSO** – User specific
- **Enter user** – user name and password for HANA system and save and new connection will be added to existing list of connections.

Now open BI Launchpad to open all BI platform tools for reporting like Office Analysis for OLAP and it will ask to choose a connection. By default, it will show you the Information View if you have specified it while creating this connection otherwise click on Next and go to folders → Choose Views *Analytic or Calculation Views*.

SAP Lumira connectivity with HANA system

Open SAP Lumira from Start Program, Click on file menu → New → Add new dataset → Connect to SAP HANA → Next



The screenshot shows the SAP Lumira interface with a sidebar containing links like Home, My Items, Documents, Visualizations, Datasets, Stories, SAP Lumira Cloud, SAP Lumira Edge Server, SAP BI, and Connections. The main area is titled 'Add new dataset' and lists several options for selecting a source. The 'Connect to SAP HANA' option is highlighted with a green arrow.

Difference between connect to SAP HANA and download from SAP HANA is that it will download data from Hana system to BO repository and refreshing of data will not occur with changes in HANA system. Enter HANA server name and Instance number. Enter user name and password → click on Connect.

This screenshot shows the 'Connect to SAP HANA' configuration dialog. It includes fields for Server (dropdown menu), Instance/Port (set to 00), Authenticate by Operating System (BAC) (unchecked), User (empty), Password (empty), Save Password (unchecked), and a Connect button. A large green arrow points upwards towards the Server dropdown.

It will show all views. You can search with the view name → Choose View → Next. It will show all measures and dimensions. You can choose from these attributes if you want → click on create option.

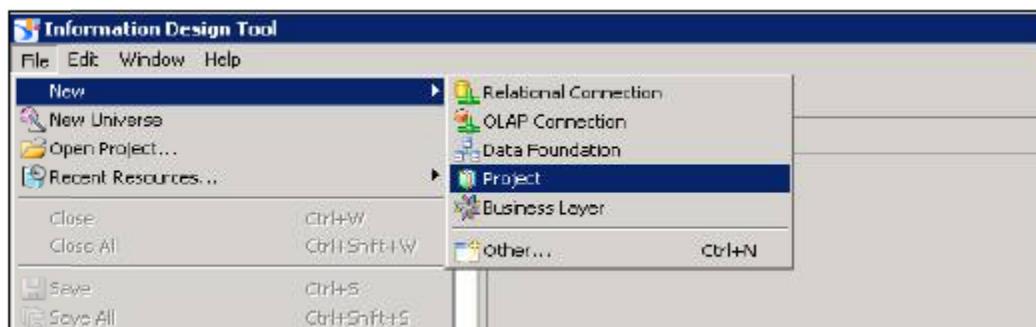
There are four tabs inside SAP Lumira –

- **Prepare** – You can see the data and do any custom calculation.
- **Visualize** – You can add Graphs and Charts. Click on X axis and Y axis + sign to add attributes.
- **Compose** – This option can be used to create sequence of Visualization *story* → click on Board to add numbers of boards → create → it will show all the visualizations on left side. Drag first Visualization then add page then add second visualization.
- **Share** – If it is built on SAP HANA, we can only publish to SAP Lumira server. Otherwise you can also publish story from SAP Lumira to SAP Community Network SCN or BI Platform.

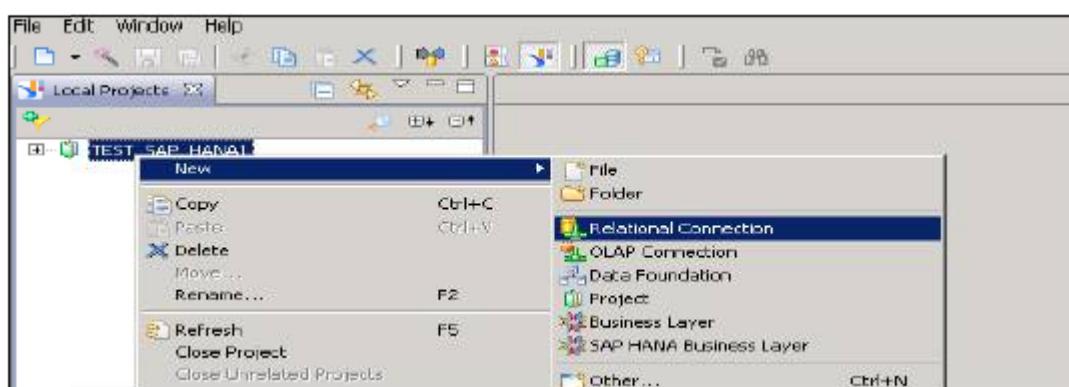
Save the file to use it later → Go to File-Save → choose Local → Save

Creating a Relational Connection in IDT to use with HANA views in WebI and Dashboard –

Open Information Design Tool → by going to BI Platform Client tools. Click on New → Project Enter Project Name → Finish.



Right-click on Project name → Go to New → Choose Relational Connection → Enter Connection/resource name → Next → choose SAP from list to connect to HANA system → SAP HANA → Select JDBC/ODBC drivers → click on Next → Enter HANA system details → Click on Next and Finish.



You can also test this connection by clicking on Test Connection option.



Test Connection → Successful. Next step is to publish this connection to Repository to make it available for use.

Right Click on connection name → click on Publish connection to Repository → Enter BO Repository name and password → Click on Connect → Next → Finish → Yes.





It will create a new relational connection with .cns extension.

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

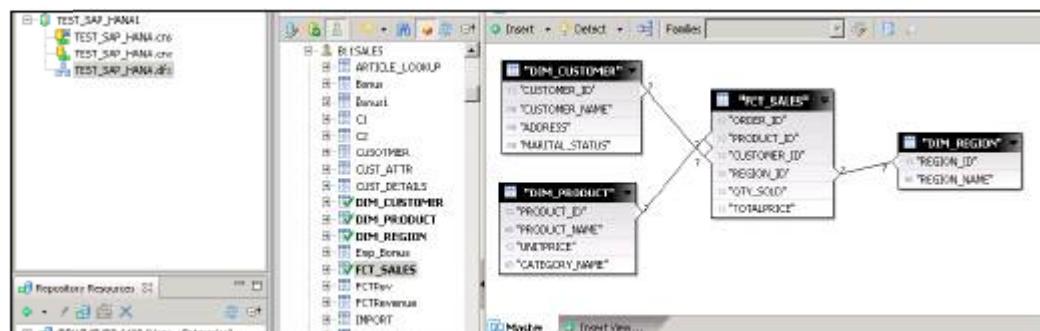
.cnx – represents local unsecured connection. If you use this connection while creating and publishing a Universe, it will not allow you to publish that to repository.

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.



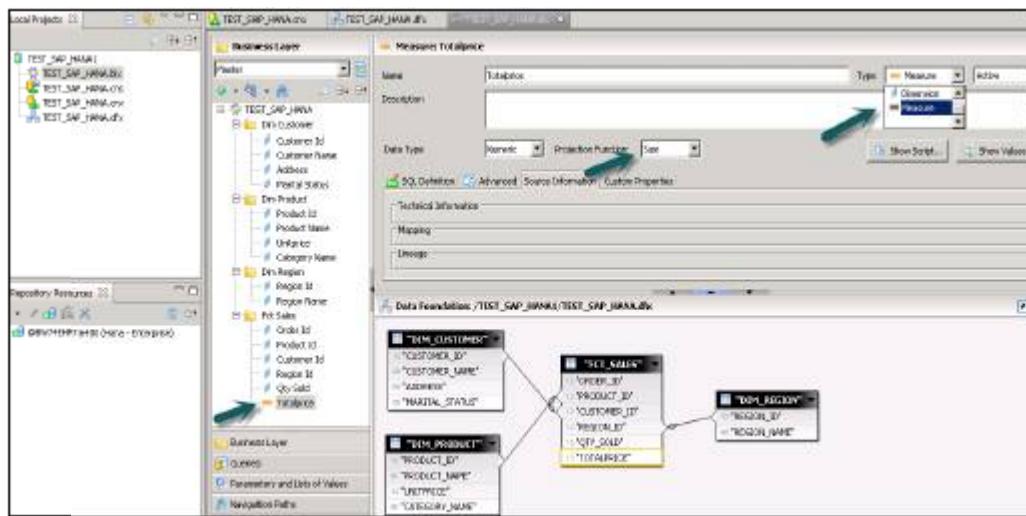
It will show all the tables in HANA database with Schema name in the middle pane.

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.



Right-click on .bfx file → click on Publish → To Repository → click on Next → Finish → Universe Published Successfully.

Now open WebI Report from BI Launchpad or Webi rich client from BI Platform client tools → New → select Universe → TEST_SAP_HANA → OK.



All Objects will be added to Query Panel. You can choose attributes and measures from left pane and add them to Result Objects. The **Run query** will run the SQL query and the output will be generated in the form of Report in WebI as shown below.

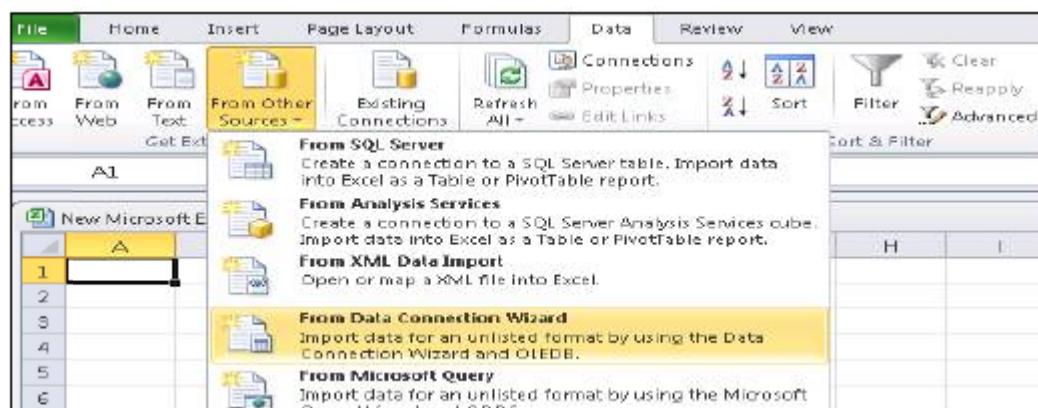
Customer No.	Marital Status	Product Name	Region Name	Qty Sold	Totalprice
David	Single	Dell Tablet	Lata m	1	10,000
David	Single	HP Tablet	Europe-Africa	3	22,000
Jack	Married	Orange M2	America	1	14,000
Jones	Married	Dell Tablet	Lata m	2	20,000

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.

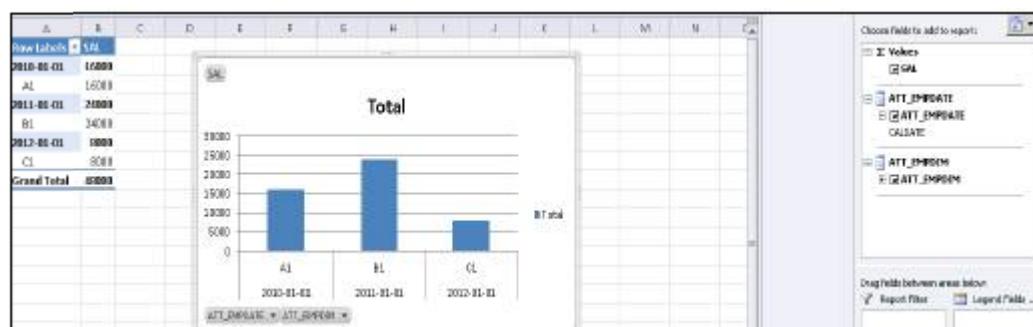
Connecting MS Excel to HANA

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 *servername, instance, username 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.



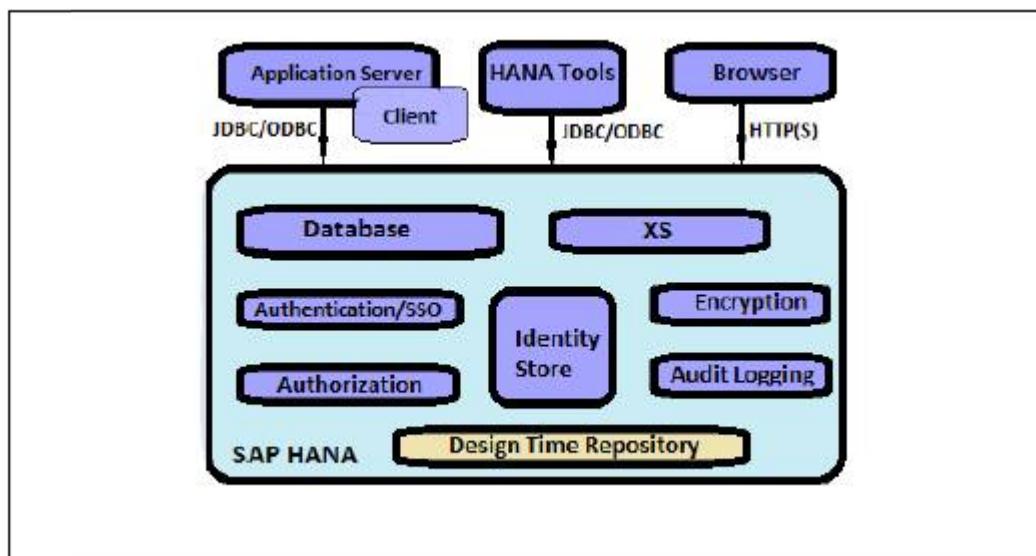
All attributes from Information view will be added to MS Excel. You can choose different attributes and measures to report as shown and you can choose different charts like pie charts and bar charts from design option at the top.

SAP HANA - SECURITY OVERVIEW

Security means protecting company's critical data from unauthorized access and use, and to ensure that Compliance and standards are met as per the company policy. SAP HANA enables customer to implement different security policies and procedures and to meet compliance requirements of the company.

SAP HANA supports multiple databases in a single HANA system and this is known as multitenant database containers. HANA system can also contain more than one multitenant database containers. A multiple container system always has exactly one system database and any number of multitenant database containers. An SAP HANA system that is installed in this environment is identified by a single system ID *SID*. Database containers in HANA system are identified by a SID and database name. SAP HANA client, known as HANA studio, connects to specific databases.

SAP HANA provides all security related features such as Authentication, Authorization, Encryption and Auditing, and some add on features, which are not supported in other multitenant databases.



Below given is a list of security related features, provided by SAP HANA –

- User and Role Management
- Authentication and SSO
- Authorization
- Encryption of data communication in Network
- Encryption of data in Persistence Layer

Additional Features in multitenant HANA database –

- **Database Isolation** – It involves preventing cross tenant attacks through operating system mechanism
- **Configuration Change blacklist** – It involves preventing certain system properties from being changed by tenant database administrators
- **Restricted Features** – It involves disabling certain database features that provides direct access to file system, the network or other resources.

SAP HANA User and Role Management

SAP HANA user and role management configuration depends on the architecture of your HANA system.

- If SAP HANA is integrated with BI platform tools and acts as reporting database, then the end-user and role are managed in application server.
- If the end-user directly connects to the SAP HANA database, then user and role in database layer of HANA system is required for both end users and administrators.

Every user wants to work with HANA database must have a database user with necessary privileges. User accessing HANA system can either be a technical user or an end user depending on the access requirement. After successful logon to system, user's authorization to perform the required operation is verified. Executing that operation depends on privileges that user has been granted. These privileges can be granted using roles in HANA Security. HANA Studio is one of powerful tool to manage user and roles for HANA database system.

User Types

User types vary according to security policies and different privileges assigned on user profile. User type can be a technical database user or end user needs access on HANA system for reporting purpose or for data manipulation.

Standard Users

Standard users are users who can create objects in their own Schemas and have read access in system Information models. Read access is provided by PUBLIC role which is assigned to every standard users.

The screenshot shows the SAP HANA Studio User Parameters dialog. The 'User' tab is selected. In the 'Authentication' section, the 'Password' checkbox is checked, and the 'Password*' and 'Confirm*' fields are populated. The 'Kerberos' checkbox is unchecked. Below this, there are fields for 'External ID*' and session client selection. Under 'Granted Roles', the 'PUBLIC' role is listed with a grantor of 'SYS'. A red arrow points to the 'PUBLIC' role entry.

Role	Grantor
PUBLIC	SYS

Restricted Users

Restricted users are those users who access HANA system with some applications and they do not have SQL privileges on HANA system. When these users are created, they do not have any access initially.

If we compare restricted users with Standard users –

- Restricted users cannot create objects in HANA database or their own Schemas.
- They do not have access to view any data in database as they don't have generic Public role added to profile like standard users.
- They can connect to HANA database only using HTTP/HTTPS.

USER ADMINISTRATION & ROLE MANAGEMENT

Technical database users are used only for administrative purpose such as creating new objects in database, assigning privileges to other users, on packages, applications etc.

SAP HANA User Administration Activities

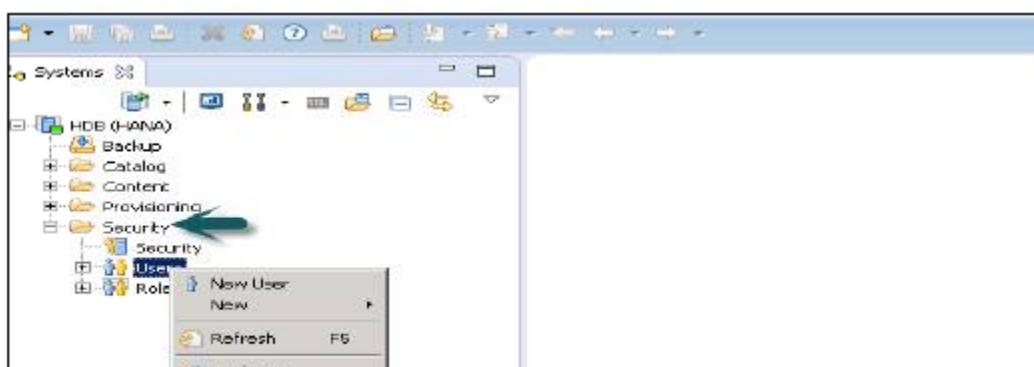
Depending on business needs and configuration of HANA system, there are different user activities that can be performed using user administration tool like HANA studio.

Most common activities include –

- Create Users
- Grant roles to users
- Define and Create Roles
- Deleting Users
- Resetting user passwords
- Reactivating users after too many failed logon attempts
- Deactivating users when it is required

How to create Users in HANA Studio?

Only database users with the system privilege ROLE ADMIN are allowed to create users and roles in HANA studio. To create users and roles in HANA studio, go to HANA Administrator Console. You will see security tab in System view –





When you expand security tab, it gives option of User and Roles. To create a new user right click on User and go to New User. New window will open where you define User and User parameters.

Enter User name *mandate* and in Authentication field enter password. Password is applied, while saving password for a new user. You can also choose to create a restricted user.

The specified role name must not be identical to the name of an existing user or role. The password rules include a minimal password length and a definition of which character types *lower*, *upper*, *digit*, *specialcharacters* have to be part of the password.

Different Authorization methods can be configured like SAML, X509 certificates, SAP Logon ticket, etc. Users in the database can be authenticated by varying mechanisms –

Internal authentication mechanism using a password.

External mechanisms such as Kerberos, SAML, SAP Logon Ticket, SAP Assertion Ticket or X.509.

A user can be authenticated by more than one mechanism at a time. However, only one password and one principal name for Kerberos can be valid at any one time. One authentication mechanism has to be specified to allow the user to connect and work with the database instance.

It also gives an option to define validity of user, you can mention validity interval by selecting the dates. Validity specification is an optional user parameter.

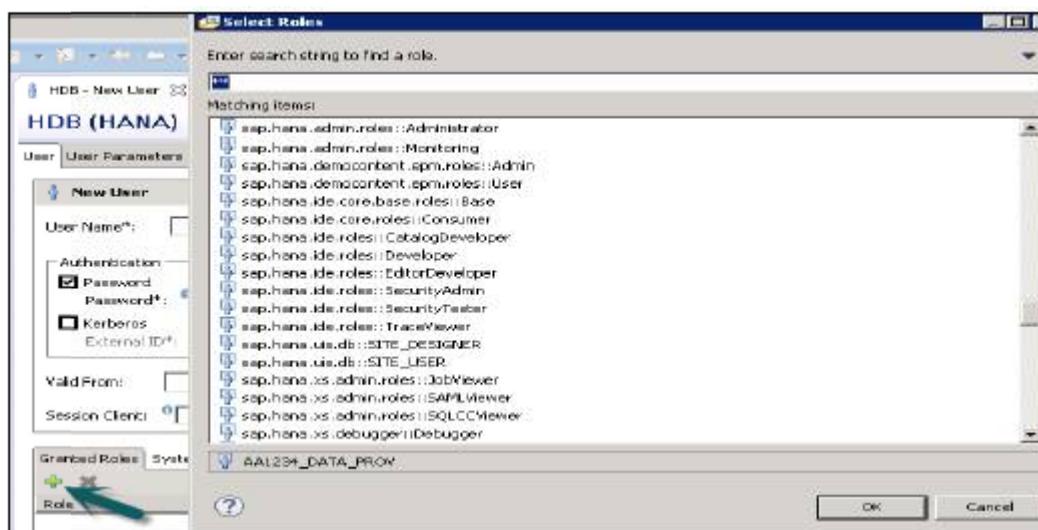
Some users that are, by default, delivered with the SAP HANA database are – SYS, SYSTEM, _SYS_REPO, _SYS_STATISTICS.

Once this is done, the next step is to define privileges for user profile. There are different types of privileges that can be added to a user profile.

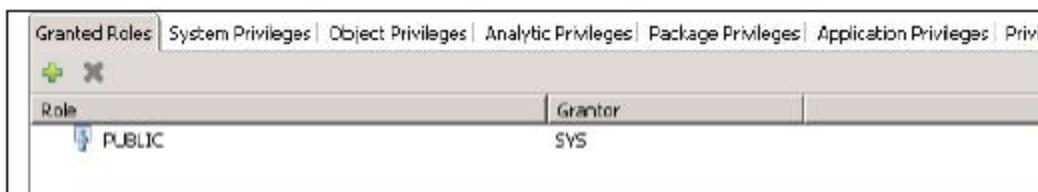
Granted Roles to a User

This is used to add inbuilt SAP.HANA roles to user profile or to add custom roles created under Roles tab. Custom roles allow you to define roles as per access requirement and you can add these roles directly to user profile. This

removes need to remember and add objects to a user profile every time for different access types.



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.

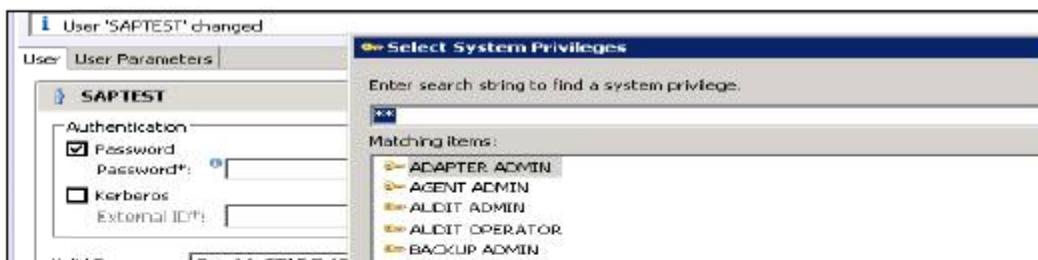
System Privileges

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

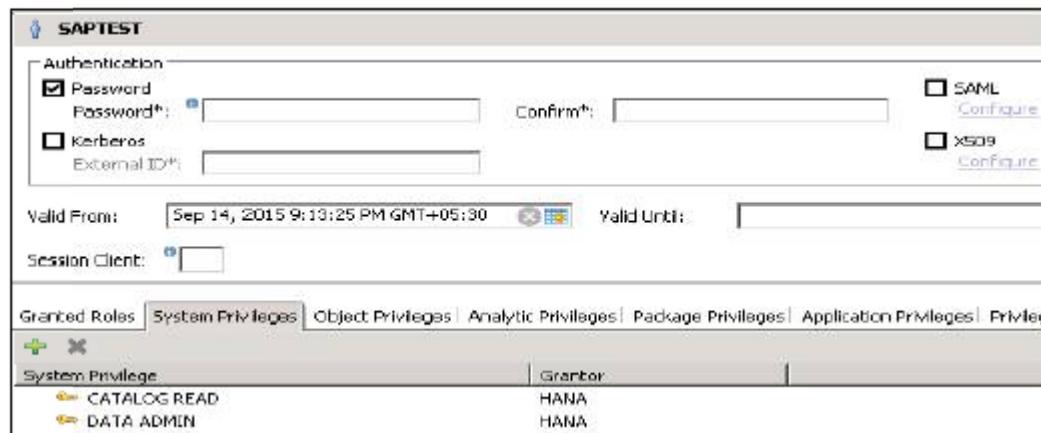
It contains the similar privileges as that in MODELING role, but with the addition that this role is allowed to grant these privileges to other users. It also contains the repository privileges to work with imported objects.





Data Admin

This is a type of privilege, required for adding Data from objects to user profile.



Given below are common supported System Privileges –

Attach Debugger

It authorizes the debugging of a procedure call, called by a different user. Additionally, the DEBUG privilege for the corresponding procedure is needed.

Audit Admin

Controls the execution of the following auditing-related commands – CREATE AUDIT POLICY, DROP AUDIT POLICY and ALTER AUDIT POLICY and the changes of the auditing configuration. Also allows access to AUDIT_LOG system view.

Audit Operator

It authorizes the execution of the following command – ALTER SYSTEM CLEAR AUDIT LOG. Also allows access to AUDIT_LOG system view.

Backup Admin

It authorizes BACKUP and RECOVERY commands for defining and initiating backup and recovery procedures.

Backup Operator

It authorizes the BACKUP command to initiate a backup process.

Catalog Read

It authorizes users to have unfiltered read-only access to all system views. Normally, the content of these views is filtered based on the privileges of the accessing user.

Create Schema

It authorizes the creation of database schemas using the CREATE SCHEMA command. By default, each user owns one schema, with this privilege the user is allowed to create additional schemas.

CREATE STRUCTURED PRIVILEGE

It authorizes the creation of Structured Privileges *AnalyticalPrivileges*. Only the owner of an Analytical Privilege can further grant or revoke that privilege to other users or roles.

Credential Admin

It authorizes the credential commands – CREATE/ALTER/DROP CREDENTIAL.

Data Admin

It authorizes reading all data in the system views. It also enables execution of any Data Definition Language *DDL* commands in the SAP HANA database

A user having this privilege cannot select or change data stored tables for which they do not have access privileges, but they can drop tables or modify table definitions.

Database Admin

It authorizes all commands related to databases in a multi-database, such as CREATE, DROP, ALTER, RENAME, BACKUP, RECOVERY.

Export

It authorizes export activity in the database via the EXPORT TABLE command.

Note that beside this privilege the user requires the SELECT privilege on the source tables to be exported.

Import

It authorizes the import activity in the database using the IMPORT commands.

Note that beside this privilege the user requires the INSERT privilege on the target tables to be imported.

Infile Admin

It authorizes changing of system settings.

License Admin

It authorizes the SET SYSTEM LICENSE command install a new license.

Log Admin

It authorizes the ALTER SYSTEM LOGGING [ON|OFF] commands to enable or disable the log flush mechanism.

Monitor Admin

It authorizes the ALTER SYSTEM commands for EVENTS.

Optimizer Admin

It authorizes the ALTER SYSTEM commands concerning SQL PLAN CACHE and ALTER SYSTEM UPDATE STATISTICS commands, which influence the behavior of the query optimizer.

Resource Admin

This privilege authorizes commands concerning system resources. For example, ALTER SYSTEM RECLAIM DATAVOLUME and ALTER SYSTEM RESET MONITORING VIEW. It also authorizes many of the commands available in the Management Console.

Role Admin

This privilege authorizes the creation and deletion of roles using the CREATE ROLE and DROP ROLE commands. It also authorizes the granting and revocation of roles using the GRANT and REVOKE commands.

Activated roles, meaning roles whose creator is the pre-defined user _SYS_REPO, can neither be granted to other roles or users nor dropped directly. Not even users having ROLE ADMIN privilege are able to do so. Please check documentation concerning activated objects.

Savepoint Admin

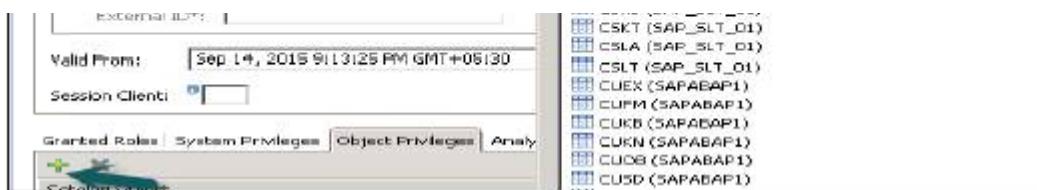
It authorizes the execution of a savepoint process using the ALTER SYSTEM SAVEPOINT command.

Components of the SAP HANA database can create new system privileges. These privileges use the component-name as first identifier of the system privilege and the component-privilege-name as the second identifier.

Object/SQL Privileges

Object privileges are also known as SQL privileges. These privileges are used to allow access on objects like Select, Insert, Update and Delete of tables, Views or Schemas.





Given below are possible types of Object Privileges –

- Object privilege on database objects that exist only in runtime
- Object privilege on activated objects created in the repository, like calculation views
- Object privilege on schema containing activated objects created in the repository,
- Object/SQL Privileges are collection of all DDL and DML privileges on database objects.

Given below are common supported Object Privileges –

There are multiple database objects in HANA database, so not all the privileges are applicable to all kinds of database objects.

Object Privilege	Command Types	Privilege Description	Comment
ALL PRIVILEGES	DDL & DML	This privilege is a collection of all Data Definition Language(DDL) and Data Manipulation Language(DML) privileges that the grantor currently possesses and is allowed to grant further. This privilege if grants is specific to the particular object being acted upon.	This privilege collection is dynamically evaluated for the given grantor and object. ALL PRIVILEGES is applicable to a table or view.
ALTER	DDL	Authorizes the ALTER command for the object	
DEBUG	DML	Authorizes debug-functionality for the procedure or calculation view or for the procedures and calculation views of a schema.	
DELETE	DML	Authorizes the DELETE and TRUNCATE commands for the object	
DROP	DDL	Authorizes the DROP commands for the object	
EXECUTE	DML	Authorizes the execution of an SQLScript function or a database procedure using the CALLS or CALL command respectively.	

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.



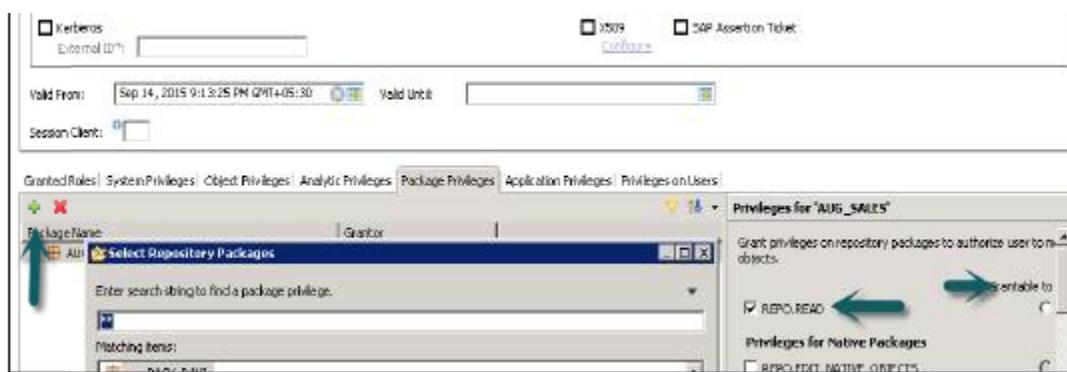
Package Privileges

In the SAP HANA repository, you can set package authorizations for a specific user or for a role. Package privileges are used to allow access to data models- Analytic or Calculation views or on to Repository objects. All privileges that are assigned to a repository package are assigned to all sub packages too. You can also mention if assigned user authorizations can be passed to other users.

Steps to add a package privileges to User profile –

- Click on Package privilege tab in HANA studio under User creation → Choose + to add one or more packages. Use Ctrl key to select multiple packages.
- In the Select Repository Package dialog, use all or part of the package name to locate the repository package that you want to authorize access to.
- Select one or more repository packages that you want to authorize access to, the selected packages appear in the Package Privileges tab.



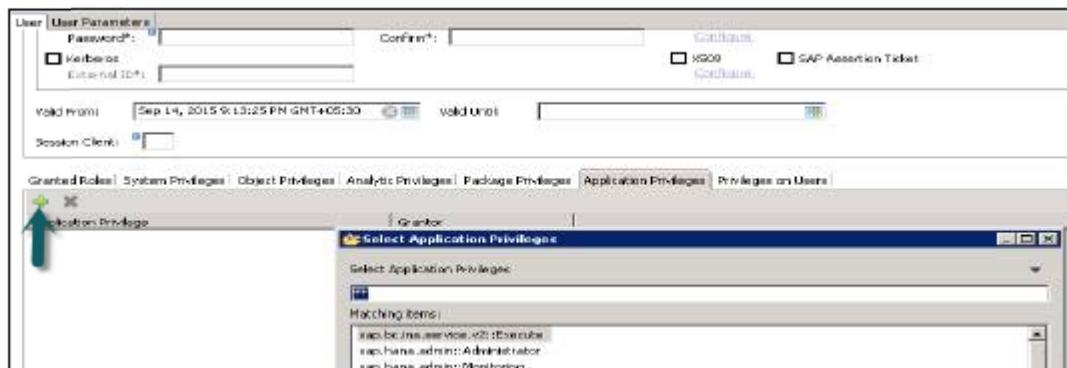


Given below are grant privileges, which are used on repository packages to authorize user to modify the objects –

- **REPO.READ** – Read access to the selected package and design-time objects *both native and imported*
- **REPO.EDIT_NATIVE_OBJECTS** – Authorization to modify objects in packages.
- **Grantable to Others** – If you choose ‘Yes’ for this, this allows assigned user authorization to pass to the other users.

Application Privileges

Application privileges in a user profile are used to define authorization for access to HANA XS application. This can be assigned to an individual user or to the group of users. Application privileges can also be used to provide different level of access to the same application like to provide advanced functions for database Administrators and read-only access to normal users.



To define Application specific privileges in a user profile or to add group of users, below privileges should be used –

- Application-privileges file *.xsprivileges*
- Application-access file *.xsaccess*
- Role-definition file < *RoleName* >. *hdbrole*

SAP HANA - AUTHENTICACTIONS

All SAP HANA users that have access on HANA database are verified with different Authenticactions 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

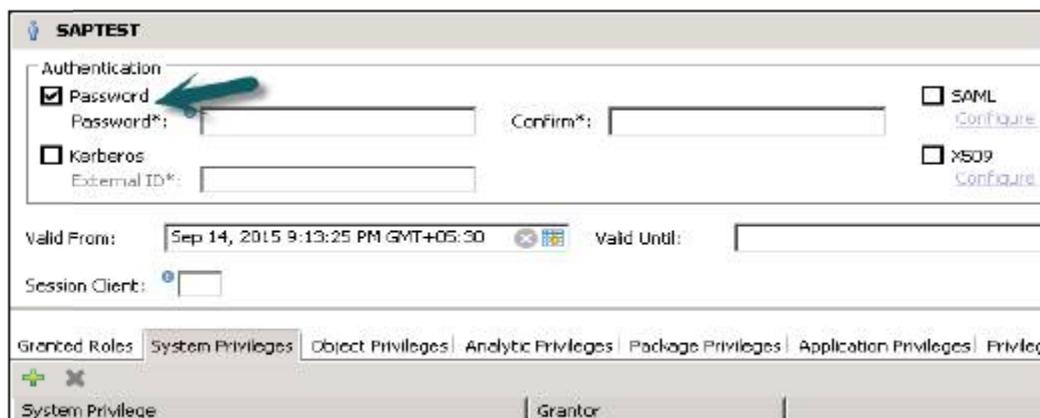
Authentication method	Available for access via JDBC/ODBC	Available for access via HTTP (SAP HANA XS)
User name/password Password policies, for example password length and complexity, can be defined	Yes	Yes (basic authentication, form-based login)
Kerberos	Yes	Yes (SPNEGO)
SAML 2.0	Yes (bearer token)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



The screenshot shows the SAP HANA Studio interface with the Security tab selected. In the Authentication section, there are several configuration fields:

- Password:** A checked checkbox with a red arrow pointing to it. Below it is a password input field labeled "Password*:".
- Kerberos:** An unchecked checkbox with a sub-field "External ID*:" and an input field.
- SAML Configure:** An unchecked checkbox with a link "Configure".
- X.509 Configure:** An unchecked checkbox with a link "Configure".

Below the authentication section, there are fields for "Valid From" (set to Sep 14, 2015 9:13:25 PM GMT+05:30) and "Valid Until" (empty), and a "Session Client" dropdown.

At the bottom, there are tabs for "Granted Roles", "System Privileges" (which is selected and highlighted in blue), and other privilege categories like "Object Privileges", "Analytic Privileges", "Package Privileges", "Application Privileges", and "Privileged Functions".

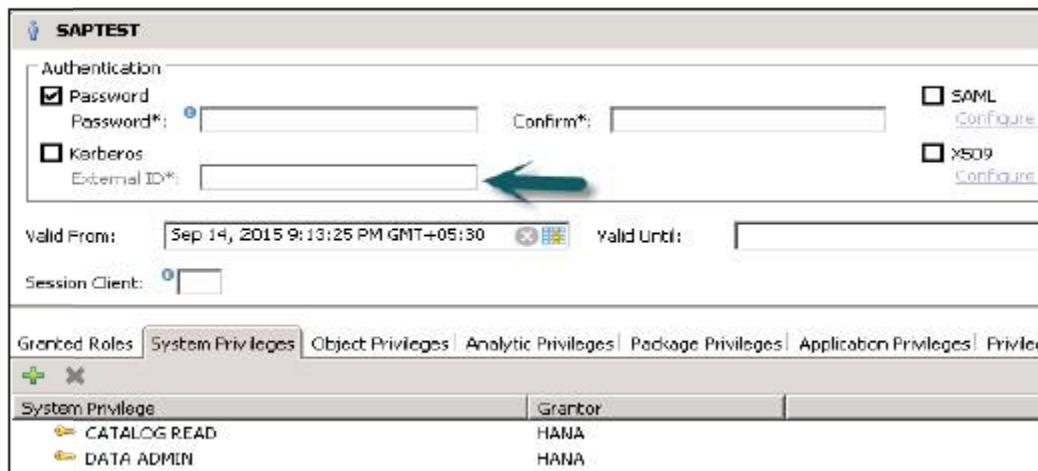
<input checked="" type="checkbox"/> CATALOG READ	HANA
<input checked="" type="checkbox"/> DATA ADMIN	HANA

Kerberos

All users who connect to HANA database system using an external authentication method should also have a database user. It is required to map external login to internal database user.

This method enables users to authenticate HANA system directly using JDBC/ODBC drivers through network or by using front end applications in SAP Business Objects.

It also allows HTTP access in HANA Extended Service using HANA XS engine. It uses SPENGO mechanism for Kerberos authentication.

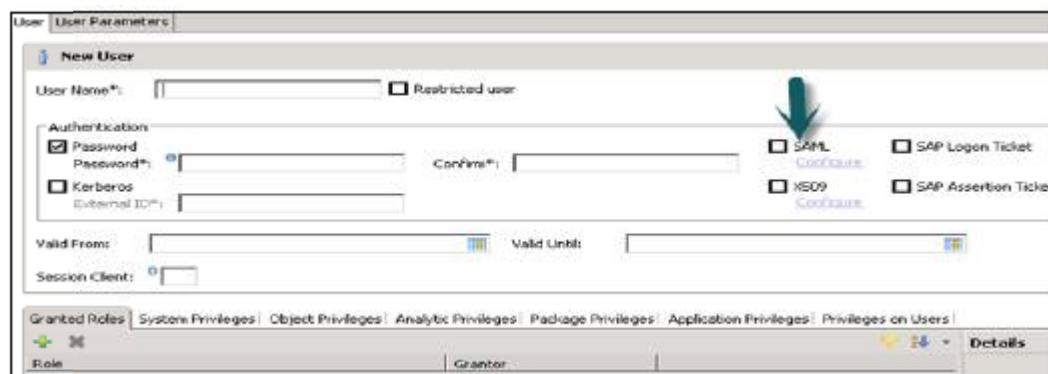


The screenshot shows the SAP HANA User Management interface for creating a new user named 'SAPTEST'. In the 'Authentication' section, the 'Kerberos' checkbox is selected, and the 'External ID*' field contains the value 'HANA'. A blue arrow points from the text 'External ID*' to the 'External ID*' field in the screenshot. Other authentication methods like 'Password' and 'SAML' are also listed. The 'Granted Roles' tab is selected, showing the roles 'CATALOG READ' and 'DATA ADMIN' assigned to the user.

SAML

SAML stands for Security Assertion Markup Language and can be used to authenticate users accessing HANA system directly from ODBC/JDBC clients. It can also be used to authenticate users in HANA system coming via HTTP through HANA XS engine.

SAML is used only for authentication purpose and not for authorization.



The screenshot shows the SAP HANA User Management interface for creating a new user. In the 'Authentication' section, the 'SAML' checkbox is selected. A blue arrow points from the text 'SAML' to the 'SAML' checkbox in the screenshot. Other authentication methods like 'Password' and 'Kerberos' are also listed. The 'Granted Roles' tab is selected, showing the roles 'CATALOG READ' and 'DATA ADMIN' assigned to the user.

SAP Logon and Assertion Tickets

SAP Logon/assertion tickets can be used to authenticate users in HANA system. These tickets are issued to users

when they login into SAP system, which is configured to issue such tickets like SAP Portal, etc. User specified in SAP logon tickets should be created in HANA system, as it does not provide support for mapping users.

User Parameters

New User

User Name*: [] Restricted user

Authentication

Password
Password*: [] Confirm*: [] SAML
[Configure](#) SAP Login Ticket
[Configure](#)

Kerberos
External ID*: [] X509
[Configure](#) SAP Assertion Ticket
[Configure](#)

Valid From: [] Valid Until: []

Session Client: []

Granted Roles System Privileges Object Privileges Analytic Privileges Package Privileges Application Privileges Privileges on Users

[Role](#) [Grantor](#) [Details](#)

X.509 Client Certificates

X.509 certificates can also be used to login to HANA system via HTTP access request from HANA XS engine. Users are authenticated by certificates that are signed from trusted Certificate Authority, which is stored in HANA XS system.

User in trusted certificate should exist in HANA system as there is no support for user mapping.

User Parameters

New User

User Name*: [] Restricted user

Authentication

Password
Password*: [] Confirm*: [] SAML
[Configure](#) SAP Login Ticket
[Configure](#)

Kerberos
External ID*: [] X509
[Configure](#) SAP Assertion Ticket
[Configure](#)

Valid From: [] Valid Until: []

Session Client: []

Granted Roles System Privileges Object Privileges Analytic Privileges Package Privileges Application Privileges Privileges on Users

[Role](#) [Grantor](#) [Details](#)

Single Sign On in HANA system

Single sign on can be configured in HANA system, which allows users to login to HANA system from an initial authentication on the client. User logs in at client applications using different authentication methods and SSO allows user to access HANA system directly.

SSO can be configured on below configuration methods –

- SAML
- Kerberos
- X.509 client certificates for HTTP access from HANA XS engine

- SAP Logon/Assertion tickets

SAP HANA - AUTHORIZATION METHODS

Authorization is checked when a user tries to connect to HANA database and perform some database operations. When a user connects to HANA database using client tools via JDBC/ODBC or Via HTTP to perform some operations on database objects, corresponding action is determined by the access that is granted to the user.

Privileges granted to a user are determined by Object privileges assigned on user profile or role that has been granted to user. Authorization is a combination of both accesses. When a user tries to perform some operation on HANA database, system performs an authorization check. When all required privileges are found, system stops this check and grants the requested access.

There are different types of privileges, which are used in SAP HANA as mentioned under User role and Management –

System Privileges

They are applicable to system and database authorization for users and control system activities. They are used for administrative tasks such as creating Schemas, data backups, creating users and roles and so on. System privileges are also used to perform Repository operations.

Object Privileges

They are applicable to database operations and apply to database objects like tables, Schemas, etc. They are used to manage database objects such as tables and views. Different actions like Select, Execute, Alter, Drop, Delete can be defined based on database objects.

They are also used to control remote data objects, which are connected through SMART data access to SAP HANA.

Analytic Privileges

They are applicable to data inside all the packages that are created in HANA repository. They are used to control modeling views that are created inside packages like Attribute View, Analytic View, and Calculation View. They apply row and column level security to attributes that are defined in modeling views in HANA packages.

Package Privileges

They are applicable to allow access to and ability to use packages that are created in repository of HANA database. Package contains different Modeling views like Attribute, Analytic and Calculation views and also Analytic Privileges defined in HANA repository database.

Application Privileges

They are applicable to HANA XS application that access HANA database via HTTP request. They are used to control access on applications created with HANA XS engine.

Application Privileges can be applied to users/roles directly using HANA studio but it is preferred that they should be applied to roles created in repository at design time.

Repository Authorization in SAP HANA Database

`_SYS_REPO` is the user owns all the objects in HANA repository. This user should be authorized externally for the objects on which repository objects are modeled in HANA system. `_SYS_REPO` is owner of all objects so it can only be used to grant access on these objects, no other user can login as `_SYS_REPO` user.

GRANT SELECT ON SCHEMA "<SCHEMA_NAME>" TO _SYS_REPO WITH GRANT OPTION

SAP HANA - LICENSE MANAGEMENT

SAP HANA License management and keys are required to use HANA database. You can install or delete HANA License keys using HANA studio.

Types of License keys

SAP HANA system supports two types of License keys –

- **Temporary License Key** – Temporary License keys are automatically installed when you install the HANA database. These keys are valid only for 90 days and you should request permanent license keys from SAP market place before expiry of this 90 days period after installation.
- **Permanent License Key** – Permanent License keys are valid only till the predefine expiration date. License keys specify amount of memory licensed to target HANA installation. They can be installed from SAP Market place under Keys and Requests tab. When a permanent License key is expired, a temporary license key is issued, which is valid for only 28 days. During this period, you have to install a permanent License key again.

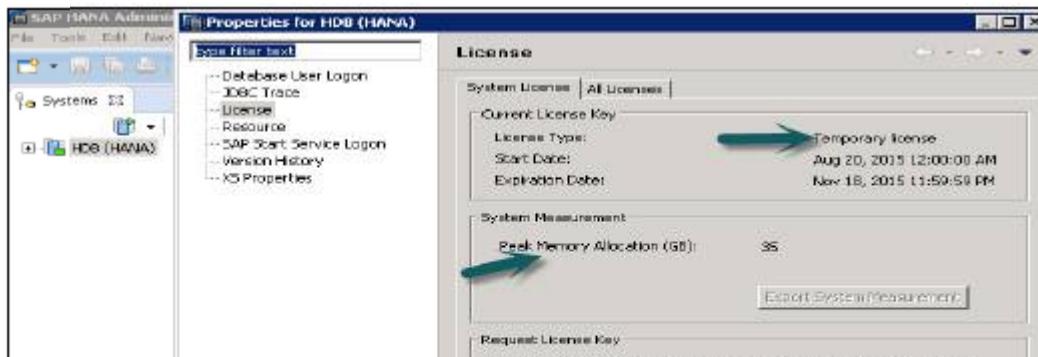
There are two types of permanent License keys for HANA system –

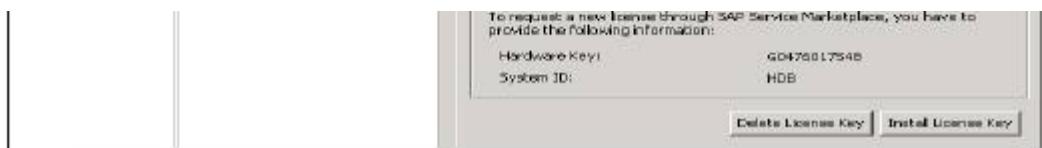
- **Unenforced** – If unenforced license key is installed and consumption of HANA system exceeds the license amount of memory, operation of SAP HANA is not affected in this case.
- **Enforced** – If Enforced license key is installed and consumption of HANA system exceeds the license amount of memory, HANA system gets locked. If this situation occurs, HANA system has to be restarted or a new license key should be requested and installed.

There are different License scenarios that can be used in HANA system depending on the landscape of the system *Standalone, HANACloud, BWonHANA, etc.* and not all of these models are based on memory of HANA system installation.

How to Check License Properties of HANA

Right Click on HANA system → Properties → License





It tells about License type, Start Date and Expiration Date, Memory Allocation and the information *HardwareKey*, *SystemId* that is required to request a new license through SAP Market Place.

Install License key → Browse → Enter Path, is used to install a new License key and delete option is used to delete any old expiration key.

All Licenses tab under License tells about Product name, description, Hardware key, First installation time, etc.

SAP HANA - AUDITING

SAP HANA audit policy tells the actions to be audited and also the condition under which the action must be performed to be relevant for auditing. Audit Policy defines what activities have been performed in HANA system and who has performed those activities at what time.

SAP HANA database auditing feature allows monitoring action performed in HANA system. SAP HANA audit policy must be activated on HANA system to use it. When an action is performed, the policy triggers an audit event to write to audit trail. You can also delete audit entries in Audit trail.

In a distributed environment, where you have multiple database, Audit policy can be enabled on each individual system. For the system database, audit policy is defined in nameserver.ini file and for tenant database, it is defined in global.ini file.

Activating an Audit Policy

To define Audit policy in HANA system, you should have system privilege – Audit Admin.

Go to Security option in HANA system → Auditing

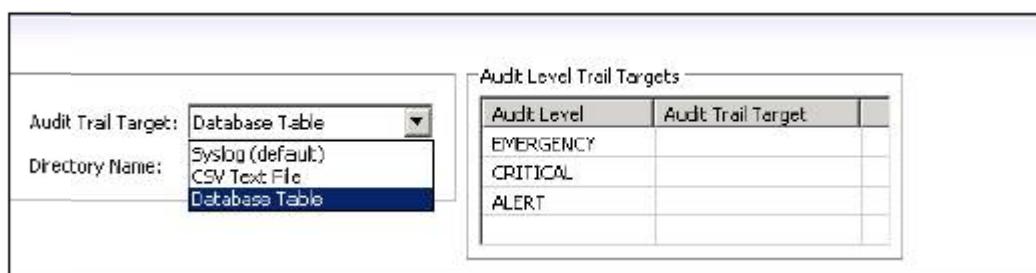


Under Global Settings → set Auditing status as enabled.

You can also choose Audit trail targets. The following audit trail targets are possible –

- **Syslog default** – Logging system of Linux Operating System.
- **Database Table** – Internal database table, user who has Audit admin or Audit operator system privilege he can only run select operation on this table.

- **CSV text** – This type of audit trail is only used for test purpose in a non-production environment.



You can also create a new Audit policy in the Audit Policies area → choose Create New Policy. Enter Policy name and actions to be audited.

Save the new policy using the Deploy button. A new policy is enabled automatically, when an action condition is met, an audit entry is created in Audit trail table. You can disable a policy by changing status to disable or you can also delete the policy.

SAP HANA - DATA REPLICATION OVERVIEW

SAP HANA Replication allows migration of data from source systems to SAP HANA database. Simple way to move data from existing SAP system to HANA is by using various data replication techniques.

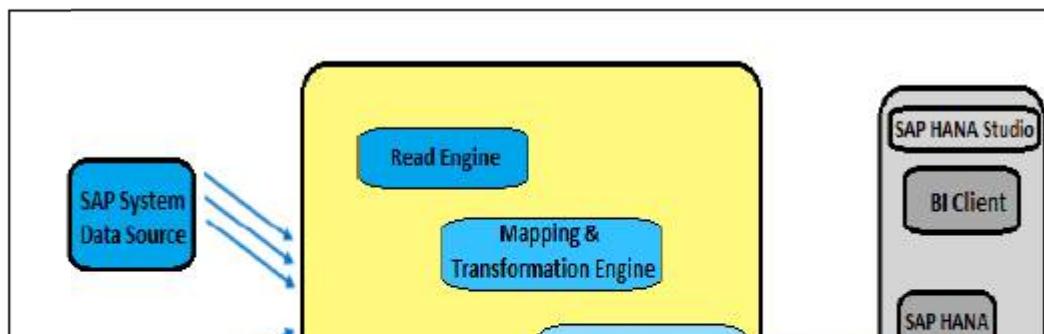
System replication can be set up on the console via command line or by using HANA studio. The primary ECC or transaction systems can stay online during this process. We have three types of data replication methods in HANA system –

- SAP LT Replication method
- ETL tool SAP Business Object Data Service *BODS* method
- Direct Extractor connection method *DXC*

SAP LT Replication Method

SAP Landscape Transformation Replication is a trigger based data replication method in HANA system. It is a perfect solution for replicating real time data or schedule based replication from SAP and non-SAP sources. It has SAP LT Replication server, which takes care of all trigger requests. Replication server can be installed as standalone server or can run on any SAP system with SAP NW 7.02 or above.

There is a Trusted RFC connection between HANA DB and ECC transaction system, which enables trigger based data replication in HANA system environment.





Advantages of SLT Replication

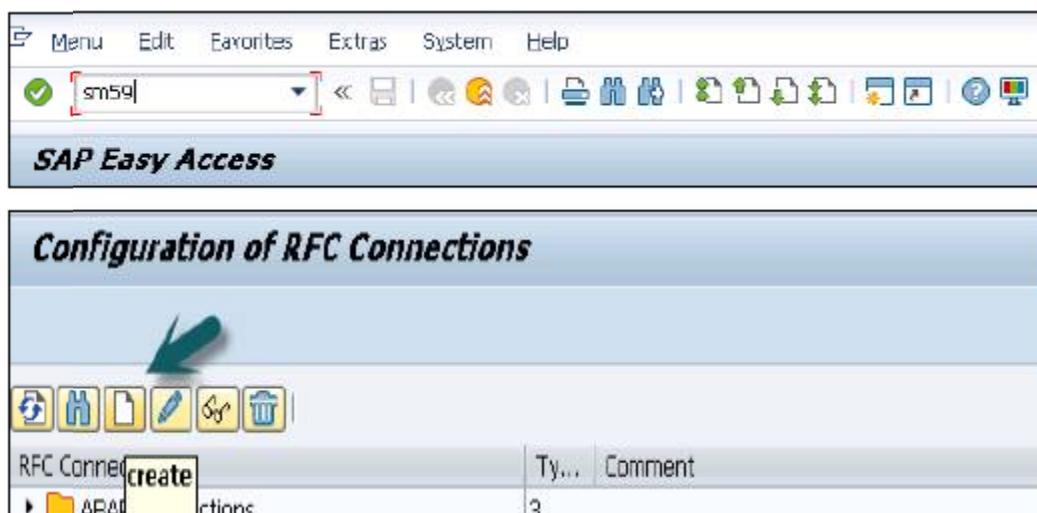
- SLT Replication method allows data replication from multiple source systems to one HANA system and also from one source system to multiple HANA systems.
- SAP LT uses trigger based approach. It has no measurable performance impact in source system.
- It also provides data transformation and filtering 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

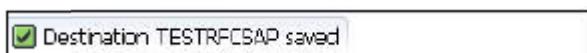
Enter Target host – ECC system name, IP and enter System number.

The screenshot shows the SAP HANA connection configuration interface. At the top, there are tabs for 'Remote Logon', 'Connection Test', and 'Unicode Test'. Below these, the 'RFC Destination' is set to 'TESTRFC5AP' and the 'Connection Type' is '3 ABAP Connection'. Under the 'Description' section, there are three fields: 'Description 1', 'Description 2', and 'Description 3'. A green arrow points to the 'Target Host' field, which contains a placeholder '1' and has a yellow background. To the right of this field is a 'System Number' checkbox. At the bottom of the screen, there are tabs for 'Administration', 'Technical Settings', 'Logon & Security' (which is selected), 'Unicode', and 'Special Options'. Below these tabs, there is a 'Target System Settings' section with options for 'Load Balancing Status' (radio buttons for 'Yes' and 'No'), 'Save to Database as' (radio buttons for 'Hostname' and 'IP Address'), and a 'Save as' dropdown menu.

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

The screenshot shows the SAP HANA connection configuration interface with the 'Logon & Security' tab selected. At the top, the 'RFC Destination' is 'TESTRFC5AP' and the 'Connection Type' is '3 ABAP Connection'. Under the 'Description' section, there are three fields: 'Description 1', 'Description 2', and 'Description 3'. Below these, there are tabs for 'Administration', 'Technical Settings', 'Logon & Security' (selected), 'Unicode', and 'Special Options'. In the 'Logon Procedure' section, there are fields for 'Language' (with a dropdown menu), 'Client' (with a dropdown menu), 'User' (with a dropdown menu), 'PW Status' (set to 'is initial'), and 'Password' (showing a masked password). A green arrow points to the 'Language' field.

Click on the Save option at the top.



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

The screenshot shows the SAP HANA connection configuration interface with the 'RFC - Connection Test' tab selected. At the top, the 'RFC Destination' is 'TESTRFC5AP' and the 'Connection Type' is 'SAP Connection'. Below these, there are tabs for 'Administration', 'Technical Settings', 'Logon & Security', 'Unicode', and 'Special Options'. In the 'Connection Test' section, it says 'Connection Test TESTRFC5AP' and 'Connection Type SAP Connection'. Below this, there is a table showing connection test results:

Action	Result
Logon	86 msec
Transfer of 0 KB	2 msec
Transfer of 10 KB	5 msec
Transfer of 100 KB	19 msec

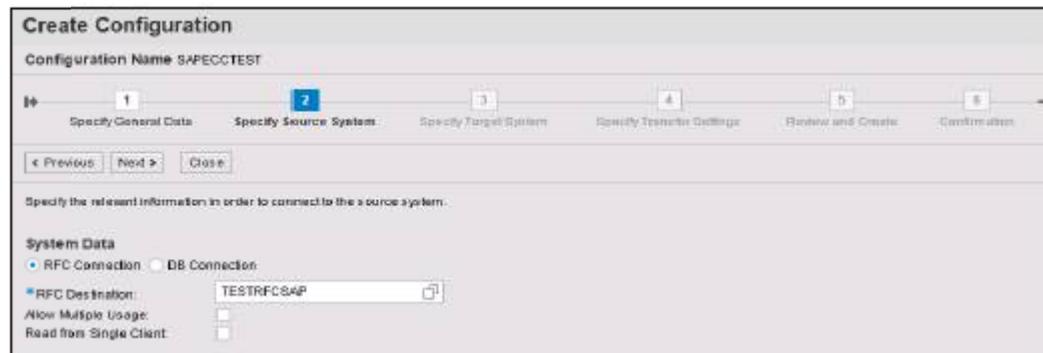


To Configure RFC connection

Run transaction – *litr toconfigureRFCconnection* → New browser will open → enter ECC system user name and password and logon.



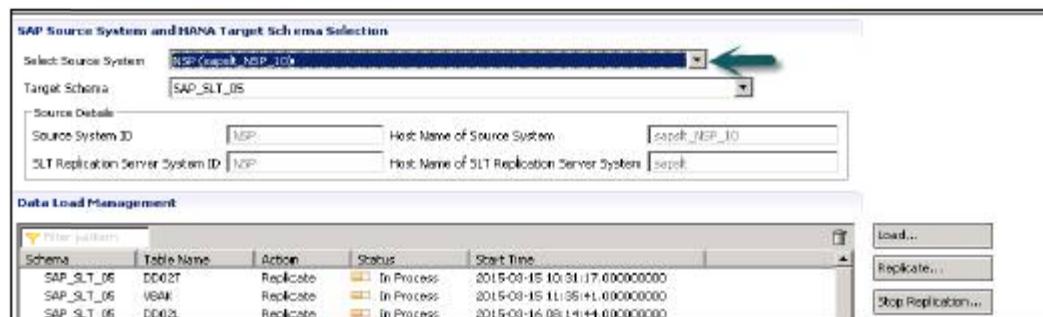
Click on New → New Window will open → Enter configuration name → Click Next → Enter RFC Destination connectionnamecreatedearlier, Use search option, choose name and click next.



In Specify Target system, Enter HANA system admin user name & password, host name, Instance number and click next. Enter No of Data transfer jobs like 007 it cannot be 000 → Next → Create Configuration.

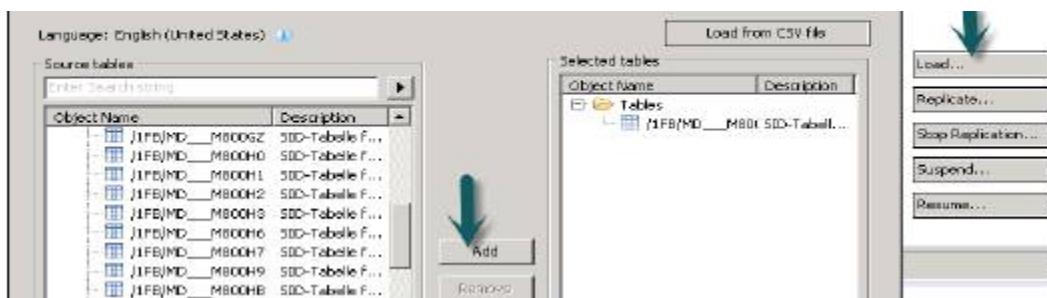
Now go to HANA Studio to use this connection –

Go to HANA Studio → Click on Data Provisioning → choose HANA system



Select source system *nameoftrustedRFCconnection* and target schema name where you want to load tables from ECC system. Select tables you want to move to HANA database → ADD → Finish.





Selected tables will move to chosen schema under HANA database.

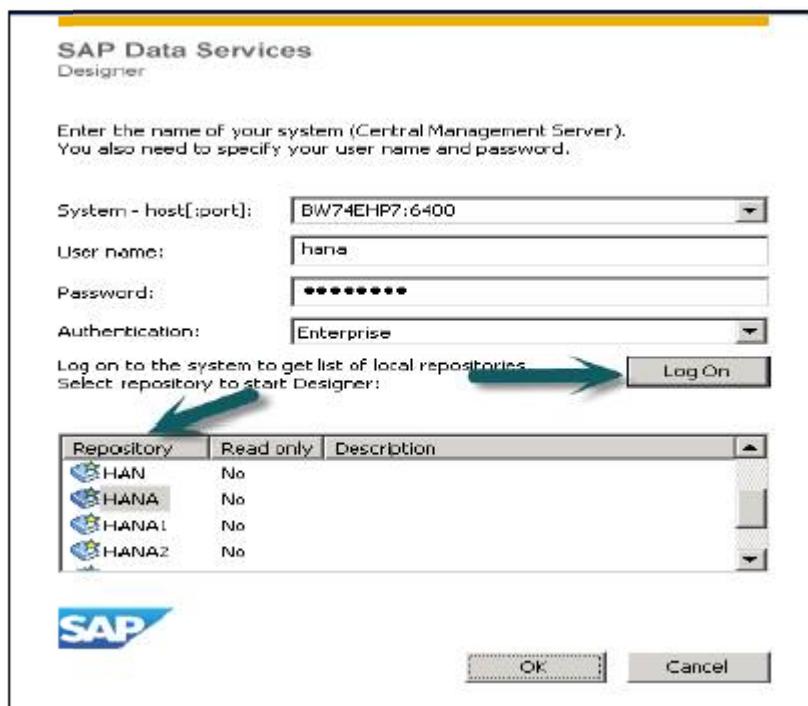
SAP HANA - ETL BASED REPLICATION

SAP HANA ETL based replication uses SAP Data Services to migrate data from SAP or non-SAP source system to target HANA database. BODS system is an ETL tool used to extract, transform and load data from source system to target system.

It enables to read the business data at Application layer. You need to define data flows in Data Services, scheduling a replication job and defining source and target system in data store in Data Services designer.

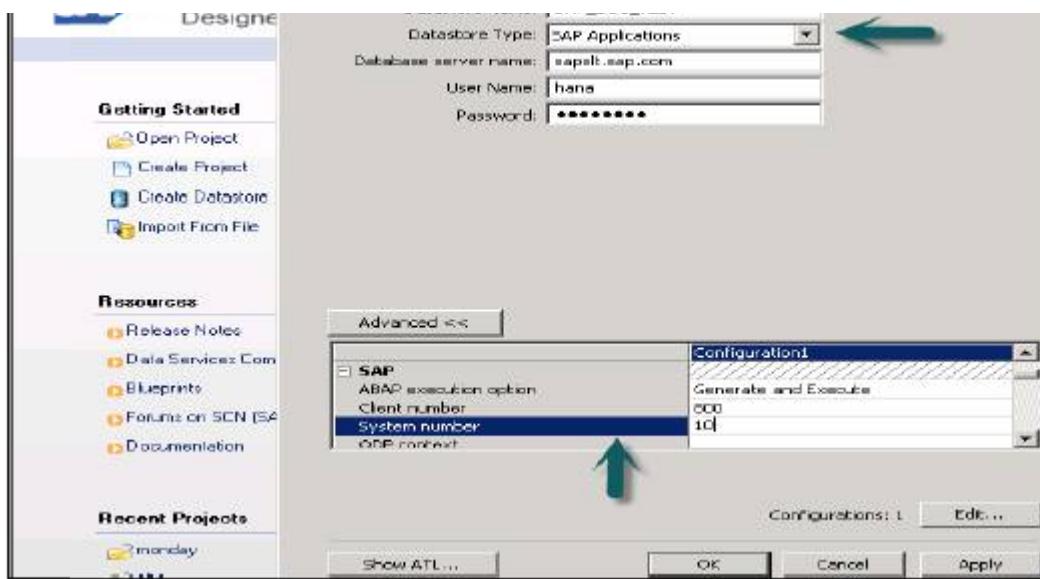
How to use SAP HANA Data Services ETL based Replication?

Login to Data Services Designer *chooseRepository* → Create Data store

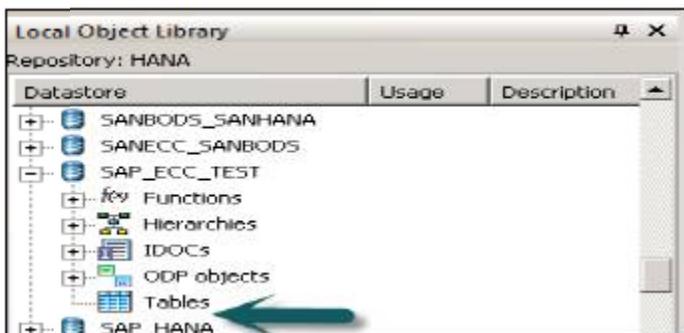


For SAP ECC system, choose database as SAP Applications, enter ECC server name, user name and password for ECC system, Advanced tab choose details as instance number, client number, etc. and apply.





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

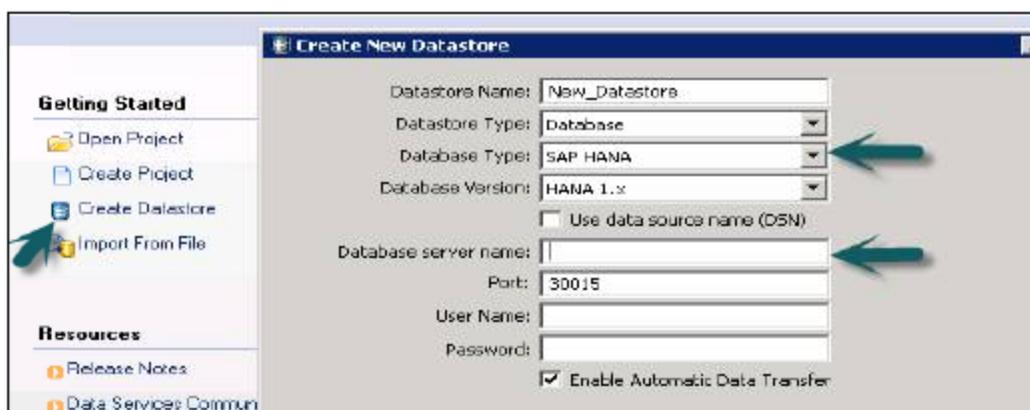


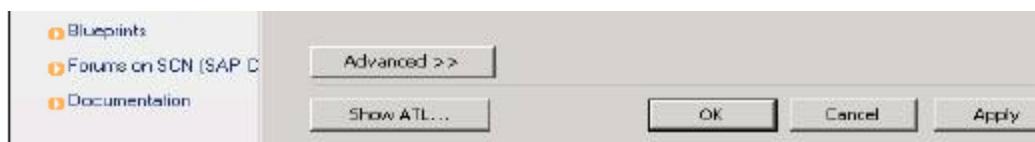
Right click on Table → Import by name → Enter ECC table to import from ECC system

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.

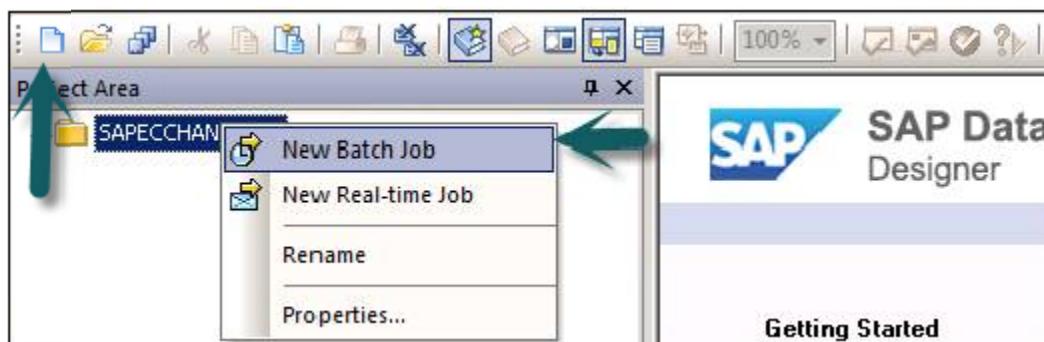




This data store will be added to Local Object Library. You can add table if you want to move data from source table to some specific table in HANA database. Note that target table should be of similar datatype as source table.

Creating a Replication Job

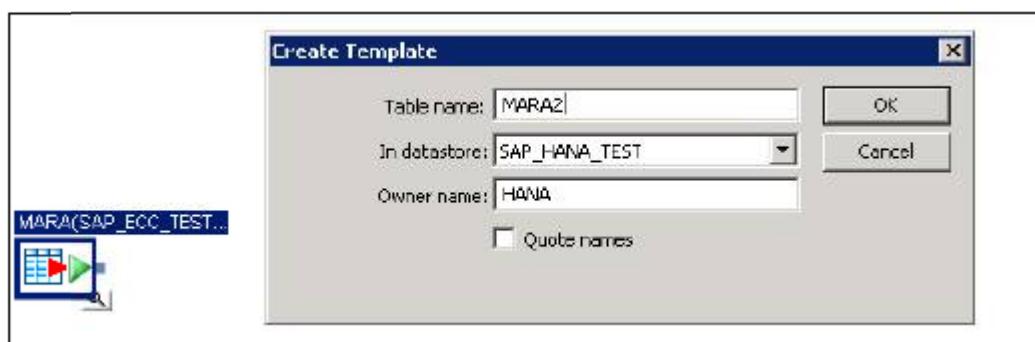
Create a new Project → Enter Project Name → Right Click on Project name → New Batch Job → Enter job name.



From right side tab, choose work flow → Enter work flow name → Double click to add it under batch job → Enter data flow → Enter data flow name → Double click to add it under batch job in Project area Save all option at top.



Drag table from First Data Store ECC *MARA* to work area. Select it and right click → Add new → Template table to create new table with similar data types in HANA DB → Enter table name, Data store ECC_HANA_TEST2 → Owner name *schemaname* → OK



Drag table to front and connect both the table → save all. Now go to batch job → Right Click → Execute → Yes → OK



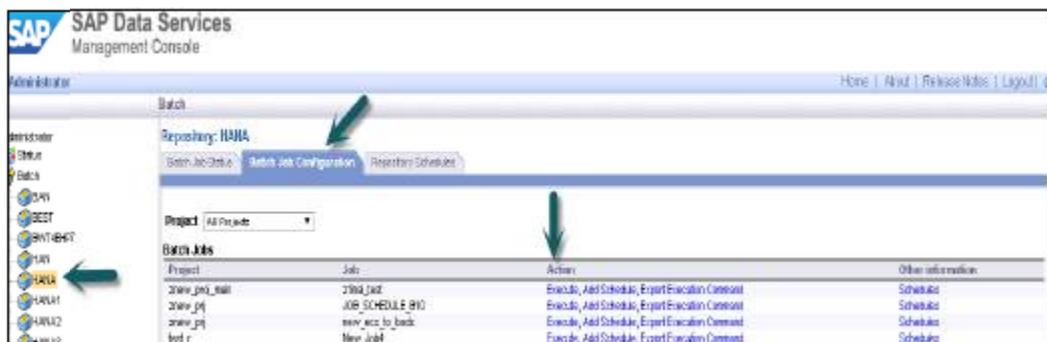
Once you execute the Replication job, you will get a confirmation that job has been completed successfully.

Go to HANA studio → Expand Schema → Tables → Verify data. This is manual execution of a batch job.

Scheduling of Batch Job

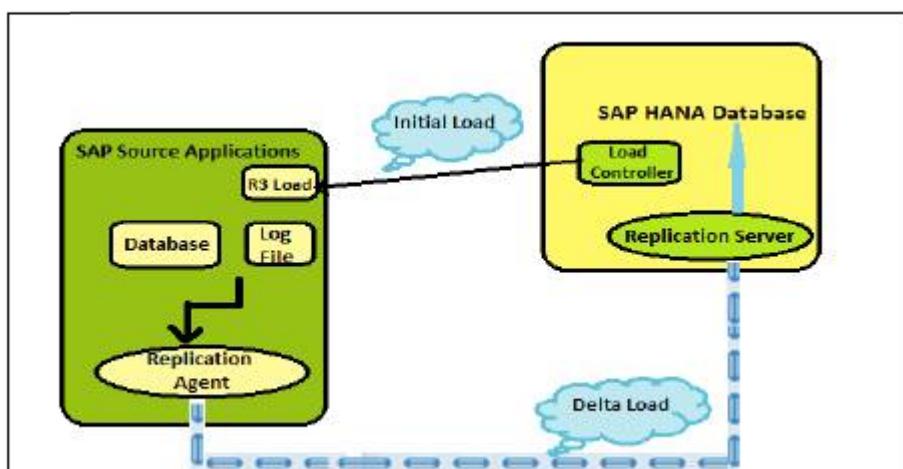
You can also schedule a batch job by going to Data Services Management console. Login to Data Services Management Console.

Choose the repository from left side → Navigate to 'Batch Job Configuration' tab, where you will see the list of jobs → Against the job you want to schedule → click on add schedule → Enter the 'schedule name' and set the parameters like *time, date, reoccurring etc.* as appropriate and click on 'Apply'.



SAP HANA - LOG BASED REPLICATION

This is also known as Sybase Replication in HANA system. The main components of this replication method are the Sybase Replication Agent, which is part of the SAP source application system, Replication agent and the Sybase Replication Server that is to be implemented in SAP HANA system.



Initial Load in Sybase Replication method is initiated by Load Controller and triggered by the administrator, in SAP HANA. It informs R3 Load to transfer initial load to HANA database. The R3 load on source system exports data for selected tables in source system and transfer this data to R3 load components in HANA system. R3 load on target system imports data into SAP HANA database.

SAP Host agent manages the authentication between the source system and target system, which is part of the source system. The Sybase Replication agent detects any data changes at time of initial load and ensures every single change is completed. When there is a change, update, and delete in entries of a table in source system, a table log is created. This table log moves data from source system to HANA database.

Delta Replication after Initial Load

The delta replication captures the data changes in source system in real time once the initial load and replication is completed. All further changes in source system are captured and replicated from source system to HANA database using above-mentioned method.

This method was part of initial offering for SAP HANA replication, but not positioned/supported anymore due to licensing issues and complexity and also SLT provides the same features.

Note – This method only supports SAP ERP system as data source and DB2 as database.

SAP HANA - DXC METHOD

Direct Extractor Connection data replication reuses existing extraction, transformation, and load mechanism built into SAP Business Suite systems via a simple HTTPS connection to SAP HANA. It is a batch-driven data replication technique. It is considered as method for extraction, transformation, and load with limited capabilities for data extraction.

DXC is a batch driven process and data extraction using DXC at certain interval is enough in many cases. You can set an interval when batch job executes example: every 20 minutes and in most of cases it is sufficient to extract data using these batch jobs at certain time intervals.

Advantages of DXC Data Replication

- This method requires no additional server or application in the SAP HANA system landscape.
- DXC method reduces complexity of data modeling in SAP HANA as data sends to HANA after applying all business extractor logics in Source System.
- It speeds up the time lines for SAP HANA implementation project
- It provides semantically rich data from SAP Business Suite to SAP HANA
- It reuses existing proprietary extraction, transformation, and load mechanism built into SAP business Suite systems over a simple HTTPS connection to SAP HANA.

Limitations of DXC Data Replication

- Data Source must have a predefined mechanism for extraction, transformation and load and if not we need to define one.

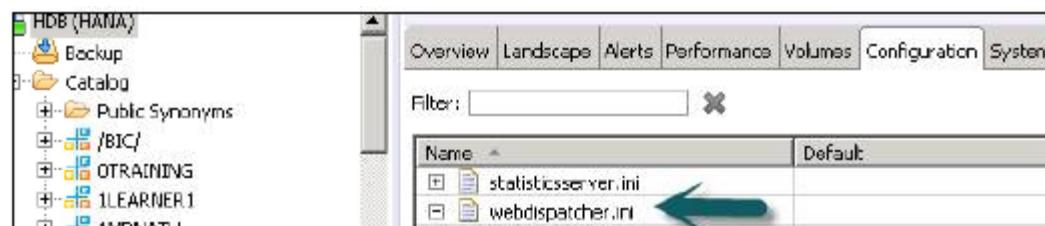
- It requires a Business Suite System based on Net Weaver 7.0 or higher with at least below SP: Release 700 SAPKW70021 SPstack19, from Nov 2008.

Configuring DXC Data Replication

Enabling XS Engine service in Configuration tab in HANA Studio – Go to Administrator tab in HANA studio of system. Go to Configuration → xsengine.ini and set instance value to 1.



Enabling ICM Web Dispatcher service in HANA Studio – Go to Configuration → webdispatcher.ini and set instance value to 1.



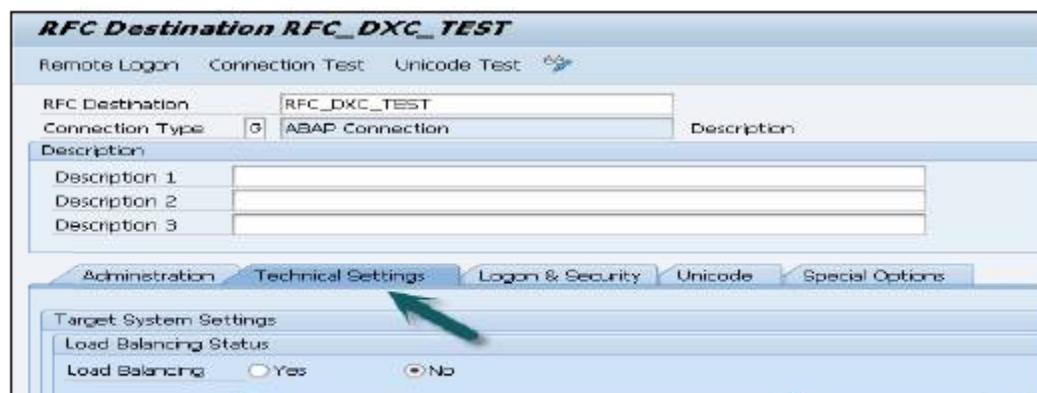
It enables ICM Web Dispatcher service in HANA system. Web dispatcher uses ICM method for data read and loading in HANA system.

Setup SAP HANA Direct Extractor Connection – Download the DXC delivery unit into SAP HANA. You can import the unit in the location /usr/sap/HDB/SYS/global/hdb/content.

Import the unit using Import Dialog in SAP HANA Content Node → Configure XS Application server to utilize the DXC → Change the application_container value to libxsdxc

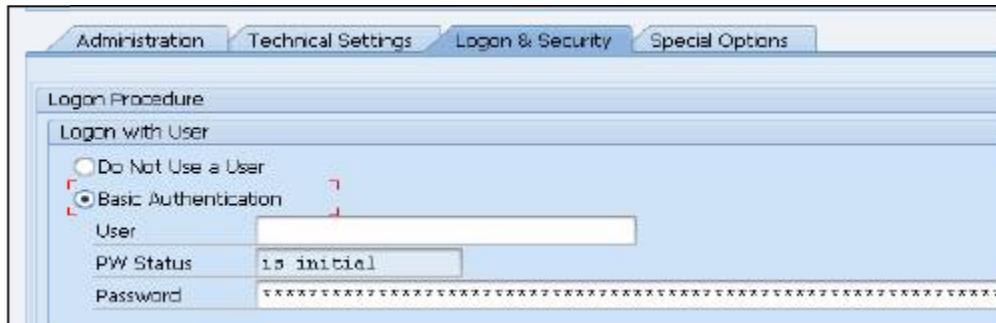
Creating a HTTP connection in SAP BW – Now we need to create http connection in SAP BW using transaction code SM59.

Input Parameters – Enter Name of RFC Connection, HANA Host Name and <Instance Number>





In Log on Security Tab, enter the DXC user created in HANA studio using basic Authentication method –



Setting up BW Parameters for HANA – Need to Setup the Following Parameters in BW Using transaction SE 38. Parameters List –

Data Browser: Table RSADMIN	
OBJECT	VAL
PSA_TO_HDB	DAT
PSA_TO_HDB_DATASOURCETABLE	DXC
PSA_TO_HDB_DESTINATION	DXC
PSA_TO_HDB_SCHEMA	DXC

- **PSA_TO_HDB_DESTINATION** – we need to mention where we need to move the Incoming data ConnectionName created using SM59
- **PSA_TO_HDB_SCHEMA** – To which Schema the replicated data need to assign
- **PSA_TO_HDB** – GLOBAL To Replicate All data source to HANA. SYSTEM – Specified clients to Use DXC. DATASOURCE – Only Specified Data Source are used for
- **PSA_TO_HDB_DATASOURCETABLE** – Need to Give the Table name having the List of data sources which are used for DXC.

Data Source Replication

Install data source in ECC using RSA5.

Replicate the metadata using specified application component (data source version Need to 7.0, if we have 3.5 version data source we need to migrate that. Active the data Source in SAP BW. Once data source is activated in SAP BW it will create the following Table in Defined schema –

- /BIC/A<data source>00 – IMDSO Active Table
- /BIC/A<data source>40 – IMDSO Activation Queue
- /BIC/A<data source>70 – Record Mode Handling Table

- /BIC/A<data source>80 – Request and Packet ID information Table
- /BIC/A<data source>A0 – Request Timestamp Table
- RSODSO IMOLOG - IMDSO related table. Stores information about all data sources related to DXC.

Now data is successfully loaded into Table /BIC/AOFI_AA_2000 once it is activated.

SAP HANA - CTL METHOD

Open SAP HANA Studio → Create Schema under Catalog tab. <Start here>

Prepare the data and save it to csv format. Now create file with “ctl” extension with following syntax –

```
-----
import data into table Schema."Table name"
from 'file.csv'
records delimited by '\n'
fields delimited by ','
 Optionally enclosed by '"'
error log 'table.err'
-----
```

Transfer this “ctl” file to the FTP and execute this file to import the data –

```
import from 'table.ctl'
```

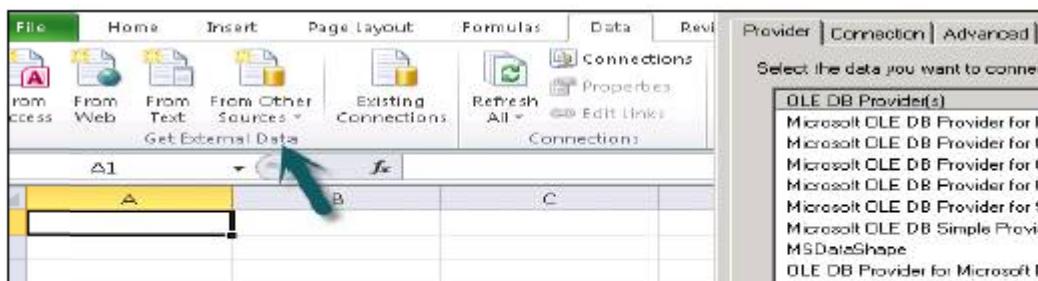
Check data in table by going to HANA Studio → Catalog → Schema → Tables → View Content

SAP HANA - MDX PROVIDER

MDX Provider is used to connect MS Excel to SAP HANA database system. It provides driver to connect HANA system to Excel and is further, used for data modelling. You can use Microsoft Office Excel 2010/2013 for connectivity with HANA for both 32 bit and 64 bit Windows.

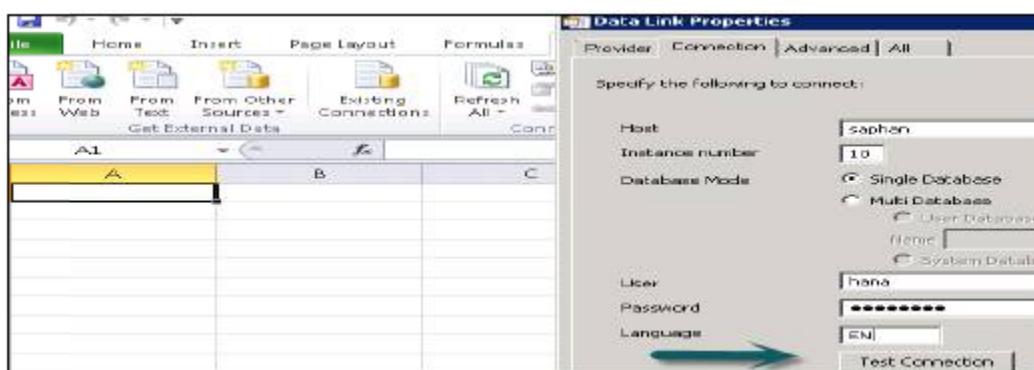
SAP HANA supports both query languages – SQL and MDX. Both languages can be used: JDBC and ODBC for SQL and ODBO is used for MDX processing. Excel Pivot tables use MDX as query language to read data from SAP HANA system. MDX is defined as part of ODBO *OLEDBforOLAP* specification from Microsoft and is used for data selections, calculations and layout. MDX supports multidimensional data model and support reporting and Analysis requirement.

MDX provider enables the consumption of Information views defined in HANA studio by SAP and non-SAP reporting tools. Existing physical tables and schemas presents the data foundation for Information models.

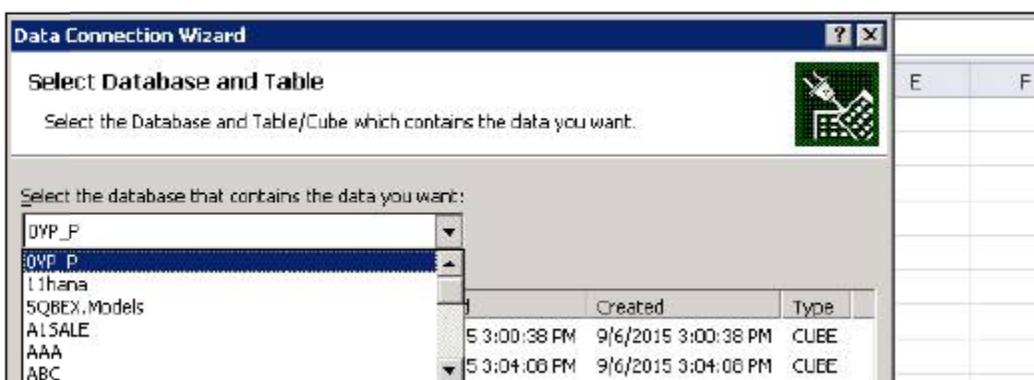




Once you choose SAP HANA MDX provider from the list of data source you want to connect, pass HANA system details like host name, instance number, user name and password.



Once the connection is successful, you can choose Package name → HANA Modeling views to generate Pivot tables.



MDX is tightly integrated into HANA database. Connection and Session management of HANA database handles statements that are executed by HANA. When these statements are executed, they are parsed by MDX interface and a calculation model is generated for each MDX statement. This calculation model creates an execution plan that generates standard results for MDX. These results are directly consumed by OLAP clients.

To make MDX connection to HANA database, HANA client tools are required. You can download this client tool from SAP market place. Once installation of HANA client is done, you will see the option of SAP HANA MDX provider in the list of data source in MS Excel.

SAP HANA - MONITORING & ALERTING

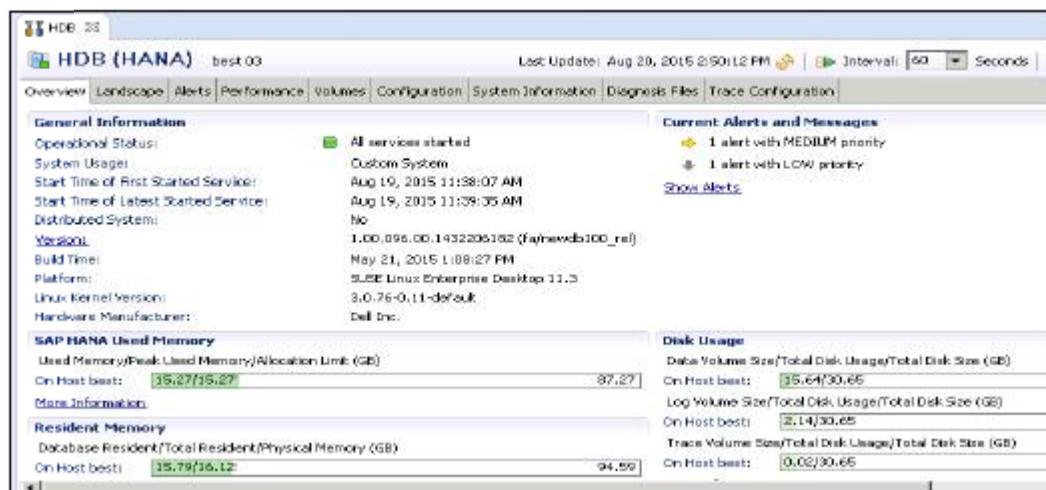
SAP HANA alert monitoring is used to monitor the status of system resources and services that are running in the HANA system. Alert monitoring is used to handle critical alerts like CPU usage, disk full, FS reaching threshold, etc. The monitoring component of HANA system continuously collects information about health, usage and performance of all the components of HANA database. It raises an alert when any of the component breaches the set threshold value.

The priority of alert raised in HANA system tells the criticality of problem and it depends on the check that is performed on the component. Example – If CPU usage is 80%, a low priority alert will be raised. However, if it reaches 96%, system will raise a high priority alert.

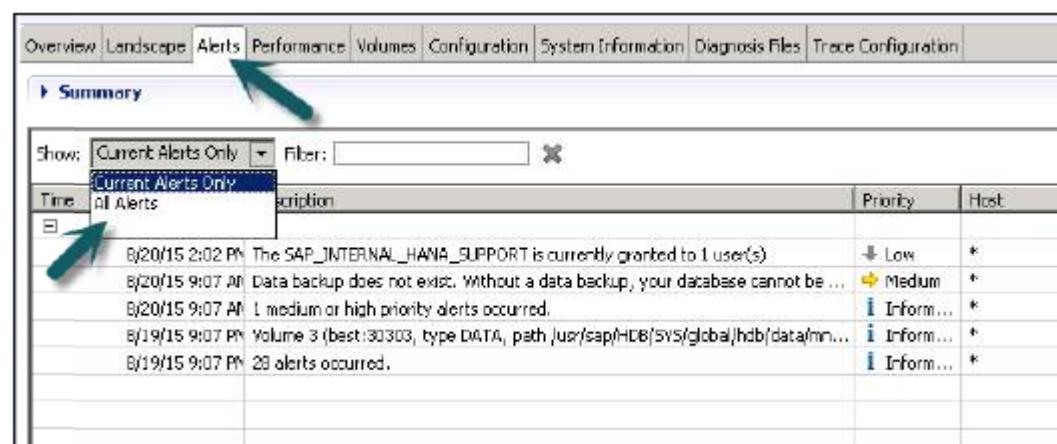
The System Monitor is the most common way to monitor HANA system and to verify the availability of all your SAP HANA system components. System monitor is used to check all key component and services of a HANA system.



You can also drill down into details of an individual system in Administration Editor. It tells about Data Disk, Log disk, Trace Disk, alerts on resource usage with priority.

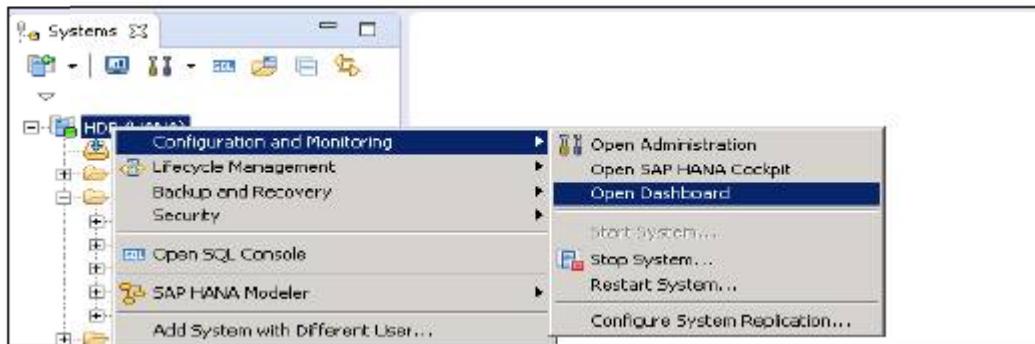


Alert tab in Administrator editor is used to check the current and all alerts in HANA system.



It also tells about the time when an alert is raised, description of the alert, priority of the alert, etc.

SAP HANA monitoring dashboard tells the key aspects of system health and configuration –



- High and Medium priority alerts.
- Memory and CPU usage
- Data backups

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.

SAP HANA - BACKUP & RECOVERY

SAP HANA backup and recovery is used to perform HANA system backups and recovery of system in case of any database failure.

Overview Tab

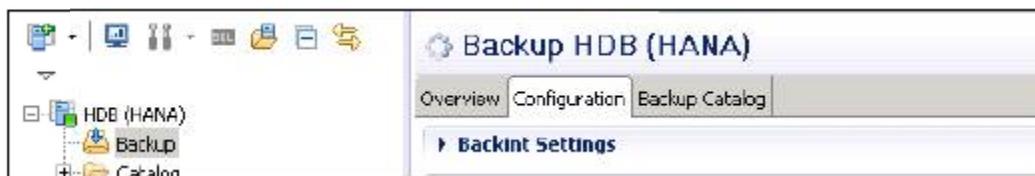
It tells the status of currently running data backup and last successful data backup.



Backup now option can be used to run data backup wizard.

Configuration Tab

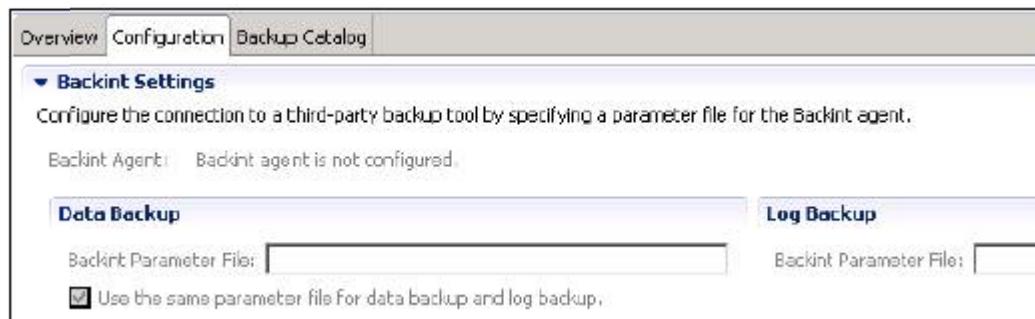
It tells about the Backup interval settings, file based data backup settings and log based data backup setting.



Backup Interval Settings

Backint settings give an option to use third party tool for data and log back up with configuration of backing agent.

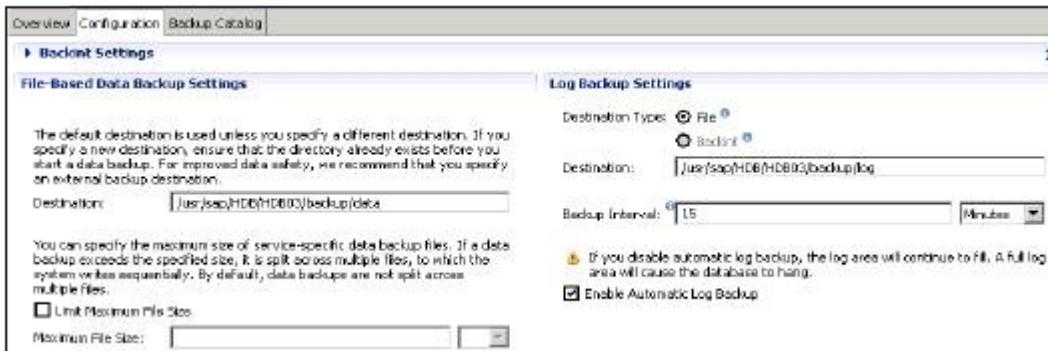
Configure the connection to a third-party backup tool by specifying a parameter file for the Backint agent.



File and Log Based Data Backup Settings

File based data backup setting tells the folder where you want to save the data backup on HANA system. You can change your backup folder.

You can also limit the size of data backup files. If system data backup exceeds this set file size, it will split across the multiple files.



Log backup settings tell the destination folder where you want to save log backup on external server. You can choose a destination type for log backup

File – ensures that sufficient space in system to store backups

Backint – is special named pipe exists on file system but require no disk space.

You can choose backup interval from drop down. It tells the longest amount of time that can pass before a new log backup is written. Backup Interval: It can be in seconds, minutes or hours.

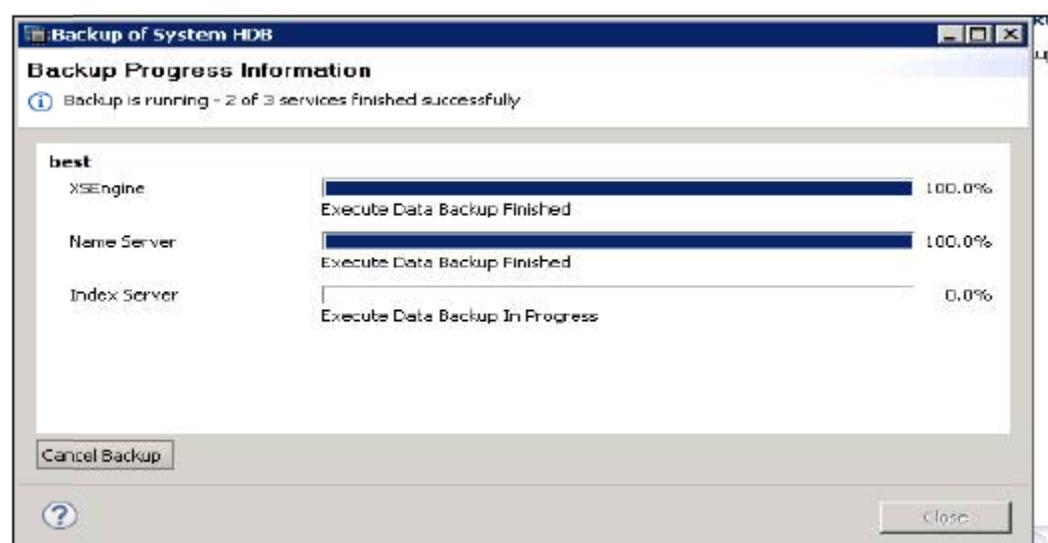
Enable Automatic log backup option: It helps you to keep log area vacant. If you disable this log area will continue to fill and that can result database to hang.

Open Backup Wizard – to run the backup of system.

Backup wizard is used to specify backup settings. It tells the Backup type, destination Type, Backup Destination folder, Backup prefix, size of backup, etc.



When you click on next → Review Backup settings → Finish



It runs the system backups and tells the time to complete backup for the each server.

Recovery of HANA System

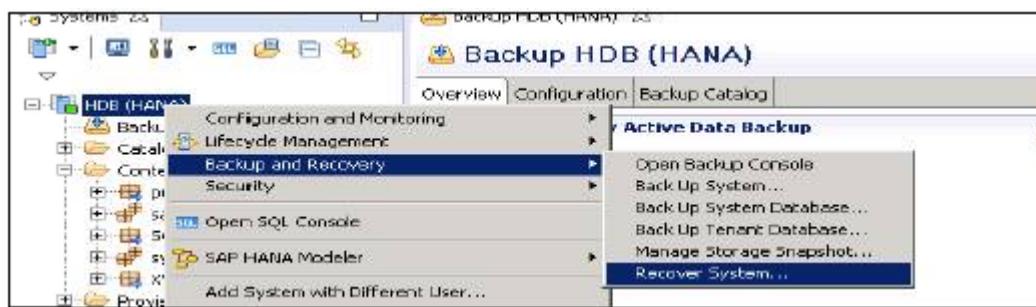
To recover SAP HANA database, the database needs to be shut down. Hence, during recovery, end users or SAP applications cannot access the database.

Recovery of SAP HANA database is required in the following situations –

- A disk in the data area is unusable or disk in the log area is unusable.
- As a consequence of a logical error, the database needs to be reset to its state at a particular point in time.
- You want to create a copy of the database.

How to recover a HANA system?

Choose HANA system → Right click → Back and Recovery → Recover System



Types of recovery in HANA system

Most Recent State – Used for recovering the database to the time as close as possible to the current time. For this recovery, the data backup and log backup have to be available since last data backup and log area are required to perform the above type recovery.

Point in Time – Used for recovering the database to the specific point in time. For this recovery, the data backup and log backup have to be available since last data backup and log area are required to perform the above type recovery

Specific Data Backup – Used for recovering the database to a specified data backup. Specific data backup is required for the above type of recovery option.

Specific Log Position – This recovery type is an advanced option that can be used in exceptional cases where a previous recovery failed.

Note – To run recovery wizard you should have administrator privileges on HANA system.

SAP HANA - HIGH AVAILABILITY

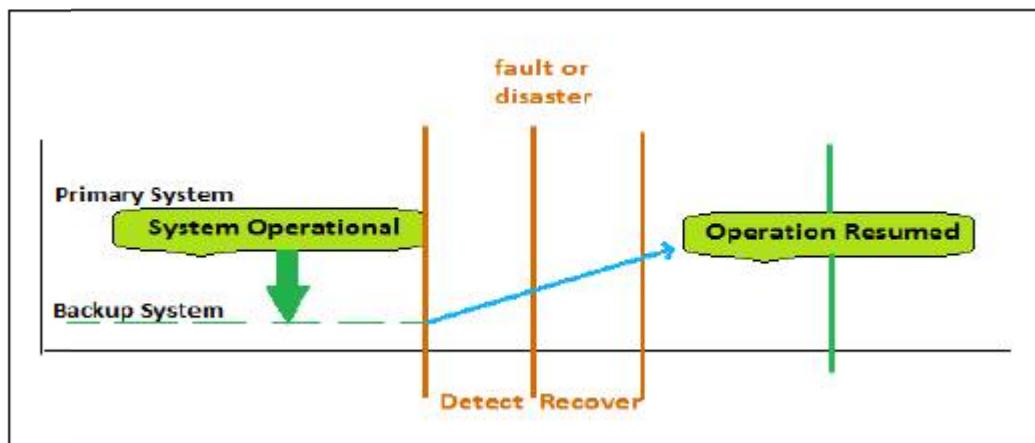
SAP HANA provides mechanism for business continuity and disaster recovery for system faults and software errors. High availability in HANA system defines set of practices that helps to achieve business continuity in case of disaster like power failures in data centers, natural disasters like fire, flood, etc. or any hardware failures.

SAP HANA high availability provides fault tolerance and ability of system to resume system operations after an

outage with minimum business loss.

The following illustration shows the phases of high availability in HANA system –

First phase is being prepared for the fault. A fault can be detected automatically or by an administrative action. Data is backed up and stand by systems take over the operations. A recovery process is put in action includes repair of faulty system and original system to be restored to previous configuration.



To achieve high availability in HANA system, key is the inclusion of extra components, which are not necessary to function and use in case of failure of other components. It includes hardware redundancy, network redundancy and data center redundancy. SAP HANA provides several levels of hardware and software redundancies as below

HANA System Hardware Redundancy

SAP HANA appliance vendors offer multiple layers of redundant hardware, software and network components, such as redundant power supplies and fans, error-correcting memories, fully redundant network switches and routers, and uninterrupted power supply *UPS*. Disk storage system guarantees writing even in the presence of power failure and use striping and mirroring features to provide redundancy for automatic recovery from disk failures.

SAP HANA Software Redundancy

SAP HANA is based on SUSE Linux Enterprise 11 for SAP and includes security pre-configurations.

SAP HANA system software includes a watchdog function, which automatically restarts configured services *indexserver*, *nameserver*, *andsoon*, in case of detected stoppage *killed or crashed*.

SAP HANA Persistence Redundancy

SAP HANA provides persistence of transaction logs, savepoints and snapshots to support system restart and recovery from failures, with minimal delay and without loss of data.

HANA System Standby and Failover

SAP HANA system involves separate standby hosts that are used for failover, in case of failure of the primary system. This improves the availability of HANA system by reducing the recovery time from an outage.

SAP HANA - LOG CONFIGURATION

The SAP HANA system logs all the transactions that change application data or the database catalog in log entries and stores them in log area. It uses these log entries in log area to roll back or repeat SQL statements. The log files are available in HANA system and can be accessed via HANA studio on Diagnosis files page under Administrator editor.

This screenshot shows the SAP HANA Studio interface. The title bar says 'HDB (HANA) best 03'. Below it, there's a navigation bar with tabs: Overview, Landscape, Alerts, Performance, Volumes, Configuration, System Information, Diagnosis Files (which is selected), and Trace Configuration. There's also a 'Last Update' timestamp. Underneath the tabs is a search bar labeled 'Filter:' and a dropdown for 'Host' set to '<All>'. Below these are two buttons: 'Merge Diagnosis Files' and 'Delete Trace'. A green arrow points to the 'Delete Trace' button. The main content area has a table with columns Host, Name, Type, Size (Byte), Modified, and a delete icon. One row is visible: 'best hdb.sopHDB_HD003' with Type 'File', Size 2,037,072, Modified '5/21/15 5:20 PM', and a delete icon.

During a log backup process, only the actual data of the log segments is written from the log area to service-specific log backup files or to a third-party backup tool.

After a system failure, you may need to redo log entries from log backups to restore the database to the desired state.

If a database service with persistence stops, it is important to ensure that it is restarted, otherwise recovery will be possible only to a point before service is stopped.

Configuring Log backup Timeout

The log backup timeout determines the interval at which the log segments are backed up if a commit has taken place in this interval. You can configure the log backup timeout using the Backup Console in SAP HANA studio –

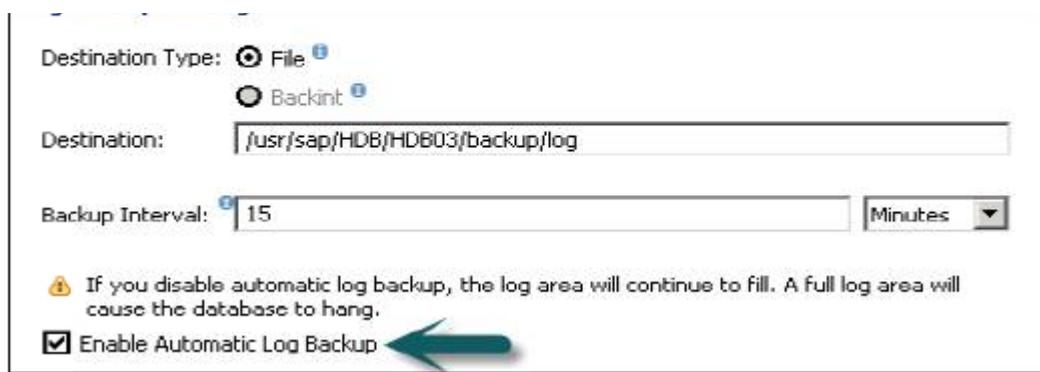
This screenshot shows the 'Backup HDB (HANA)' configuration screen. At the top, there are tabs: Overview, Configuration (which is selected), and Backup Catalog. Below the tabs, there's a sidebar with 'Backend Settings' and 'File-Based Data Backup Settings'. In the 'File-Based Data Backup Settings' section, there's a note about destination and a 'Destination' field containing '/user/sap/HDB/HD003/backup/dbs'. To the right, there's a 'Log Backup Settings' section with 'Destination Type' set to 'File' (selected with a radio button). The 'Destination' field contains '/user/sap/HDB/HD003/backup/log'. Below these, there's a 'Backup Interval' field with '35' and a dropdown menu set to 'Minutes'. A green arrow points to the 'Backup Interval' field.

You can also configure the log_backup_timeout_s interval in the global.ini configuration file.

The log backup to the "File" and backup mode "NORMAL" are the default settings for the automatic log backup function after installation of SAP HANA system. Automatic log backup only works if at least one complete data backup has been performed.

Once the first complete data backup has been performed, the automatic log backup function is active. SAP HANA studio can be used to enable/disable the automatic log backup function. It is recommended to keep automatic log backup enabled otherwise log area will continue to fill. A full log area can result a database freeze in HANA system.

This screenshot shows the 'Log Backup Settings' configuration screen. At the top, there's a header 'Log Backup Settings' with an information icon. Below the header, there's a table with two rows: 'Destination' and 'Backup Interval'. The 'Destination' row has a dropdown menu with 'File' selected. The 'Backup Interval' row has a dropdown menu with 'Minutes' selected. The entire configuration screen is enclosed in a light blue border.



You can also change the enable_auto_log_backup parameter in the persistence section of the global.ini configuration file.

SAP HANA - SQL OVERVIEW

SQL stands for Structured Query Language.

It is a standardized language for communicating with a database. SQL is used to retrieve the data, store or manipulate the data in the database.

SQL statements perform the following functions –

- Data definition and manipulation
- System management
- Session management
- Transaction management
- Schema definition and manipulation

The set of SQL extensions, which allow developers to push data into database, is called **SQL scripts**.

Data Manipulation Language *DML*

DML statements are used for managing data within schema objects. Some examples –

- **SELECT** – retrieve data from the database
- **INSERT** – insert data into a table
- **UPDATE** – updates existing data within a table

Data Definition Language *DDL*

DDL statements are used to define the database structure or schema. Some examples –

- **CREATE** – to create objects in the database
- **ALTER** – alters the structure of the database

- **DROP** – delete objects from the database

Data Control Language *DCL*

Some examples of DCL statements are –

- **GRANT** – gives user's access privileges to database
- **REVOKE** – withdraw access privileges given with the GRANT command

Why do We Need SQL?

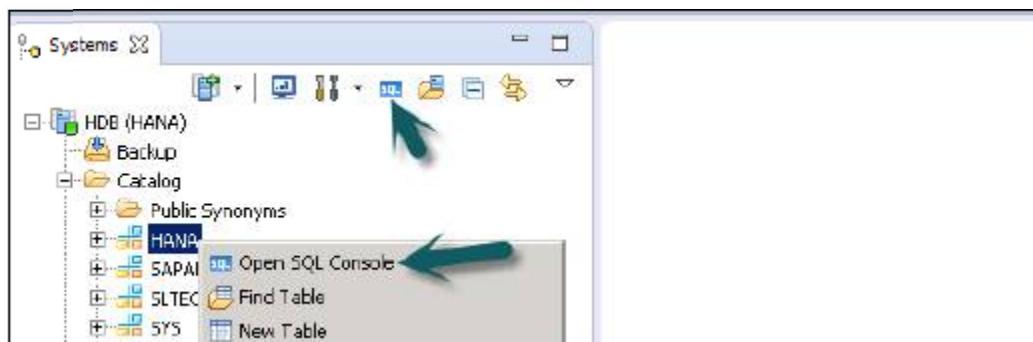
When we create Information Views in SAP HANA Modeler, we are creating it on top of some OLTP applications. All these in back end run on SQL. Database understands only this language.

To do a testing if our report will meet the business requirement we have to run SQL statement in database if Output is according to the requirement.

HANA Calculation views can be created in two ways - Graphical or using SQL script. When we create more complex Calculation views, then we might have to use direct SQL scripts.

How to open SQL console in HANA Studio?

Select the HANA system and click on SQL console option in system view. You can also open SQL console by right click on Catalog tab or any on any Schema name.



SAP HANA can act both as Relational as well as OLAP database. When we use BW on HANA, then we create cubes in BW and HANA, which act as relational database and always produce a SQL Statement. However, when we directly access HANA views using OLAP connection, then it will act as OLAP database and MDX will be generated.

SAP HANA - DATA TYPES

You can create row or Column store tables in SAP HANA using create table option. A table can be created by executing a data definition create table statement or using graphical option in HANA studio.

When you create a table, you also need to define attributes inside it.

SQL statement to create a table in HANA Studio SQL Console –

```
Create column Table TEST (
```

```

ID INTEGER,
NAME VARCHAR(10),
PRIMARY KEY (ID)
);

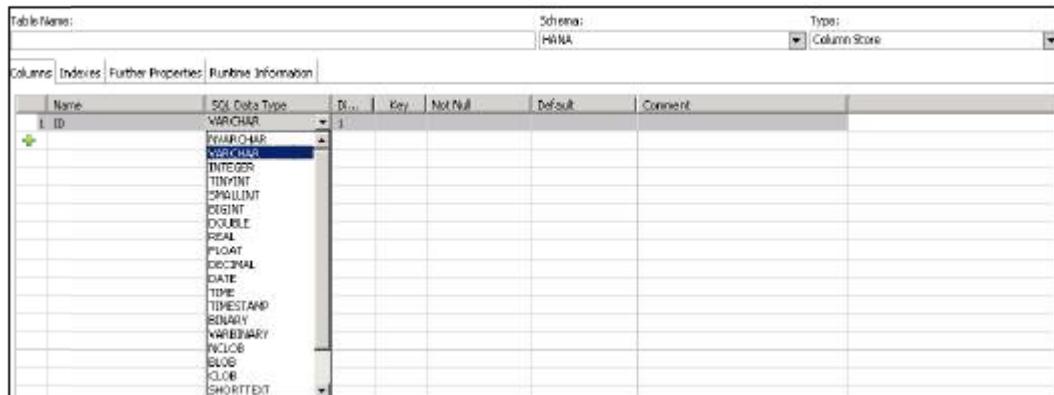
```

Creating a table in HANA studio using GUI option –



When you create a table, you need to define the names of columns and SQL data types. The Dimension field tells the length of value and the Key option to define it as primary key.

SAP HANA supports the following data types in a table –



SAP HANA supports 7 categories of SQL data types and it depends on the type of data you have to store in a column.

- Numeric
- Character/ String
- Boolean
- Date Time
- Binary
- Large Objects
- Multi-Valued

The following table gives the list of data types in each category –

Category	Data Type
Date Time	DATE, TIME, SECONDDATE, TIMESTAMP
Numeric	TINYINT, SMALLINT, INTEGER, BIGINT, SMALLDECIMAL, DECIMAL, REAL, DOUBLE

Boolean	BOOLEAN
Character	VARCHAR, NVARCHAR, ALPHANUM, SHORTTEXT
Binary	VARBINARY
Large Objects	BLOB, CLOB, NCLOB, TEXT
Multi-Valued	ARRAY

Date Time

These data types are used to store date and time in a table in HANA database.

- **DATE** – data type consists of year, month and day information to represent a date value in a column. Default format for a Date data type is YYYY-MM-DD.
- **TIME** – data type consists of hours, minutes, and seconds value in a table in HANA database. Default format for Time data type is HH: MI: SS.
- **SECOND DATE** – data type consists of year, month, day, hour, minute, second value in a table in HANA database. Default format for SECONDDATE data type is YYYY-MM-DD HH:MM:SS.
- **TIMESTAMP** – data type consists of date and time information in a table in HANA database. Default format for TIMESTAMP data type is YYYY-MM-DD HH:MM:SS:FFn, where FFn represents fraction of second.

Numeric

- **TinyINT** – stores 8 bit unsigned integer. Min value: 0 and max value: 255
- **SMALLINT** – stores 16 bit signed integer. Min value: -32,768 and max value: 32,767
- **Integer** – stores 32 bit signed integer. Min value: -2,147,483,648 and max value: 2,147,483,648
- **BIGINT** – stores 64 bit signed integer. Min value: -9,223,372,036,854,775,808 and max value: 9,223,372,036,854,775,808
- **SMALL** – Decimal and Decimal: Min value: $-10^{38} +1$ and max value: $10^{38} -1$
- **REAL** – Min Value:-3.40E + 38 and max value: 3.40E + 38
- **DOUBLE** – stores 64 bit floating point number. Min value: -1.7976931348623157E308 and max value: 1.7976931348623157E308

Boolean

Boolean data types stores Boolean value, which are TRUE, FALSE

Character

- **Varchar** – maximum of 8000 characters.
- **Nvarchar** – maximum length of 4000 characters
- **ALPHANUM** – stores alphanumeric characters. Value for an integer is between 1 to 127.
- **SHORTTEXT** – stores variable length character string which supports text search features and string

search features.

Binary

Binary types are used to store bytes of binary data.

VARBINARY – stores binary data in bytes. Max integer length is between 1 and 5000.

Large Objects

LARGEOBJECTS are used to store a large amount of data such as text documents and images.

- **NCLOB** – stores large UNICODE character object.
- **BLOB** – stores large amount of Binary data.
- **CLOB** – stores large amount of ASCII character data.
- **TEXT** – it enables text search features. This data type can be defined for only column tables and not for row store tables.
- **BINTEXT** – supports text search features but it is possible to insert binary data.

Multivalued

Multivalued data types are used to store collection of values with same data type.

Array

Arrays store collections of value with the same data type. They can also contain null values.

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

Comparison Operators

Comparison operators are used to compare the values in SQL statement.

Operator	Description
=	Checks if the values of two operands are equal or not, if yes then condition becomes true.
!=	Checks if the values of two operands are equal or not, if values are not equal then condition becomes true.
<>	Checks if the values of two operands are equal or not, if values are not equal then condition becomes true.
>	Checks if the value of left operand is greater than the value of right operand, if yes then condition becomes true.
<	Checks if the value of left operand is less than the value of right operand, if yes then condition becomes true.
>=	Checks if the value of left operand is greater than or equal to the value of right operand, if yes then condition becomes true.
<=	Checks if the value of left operand is less than or equal to the value of right operand, if yes then condition becomes true.
!<	Checks if the value of left operand is not less than the value of right operand, if yes then condition becomes true.
!>	Checks if the value of left operand is not greater than the value of right operand, if yes then condition becomes true.

Logical operators

Logical operators are used to pass multiple conditions in SQL statement or are used to manipulate the results of conditions.

Operator	Description
ALL	The ALL Operator is used to compare a value to all values in another value set.
AND	The AND operator allows the existence of multiple conditions in an SQL statement's WHERE clause.
ANY	The ANY operator is used to compare a value to any applicable value in the list according to the condition.
BETWEEN	The BETWEEN operator is used to search for values that are within a set of values, given the minimum value and the maximum value.
EXISTS	The EXISTS operator is used to search for the presence of a row in a specified table that meets certain criteria.
IN	The IN operator is used to compare a value to a list of literal values that have been specified.
LIKE	The LIKE operator is used to compare a value to similar values using wildcard operators.
NOT	The NOT operator reverses the meaning of the logical operator with which it is used. Eg – NOT EXISTS, NOT BETWEEN, NOT IN, etc. This is a negate operator.
OR	The OR operator is used to compare multiple conditions in an SQL statement's WHERE clause.
IS NULL	The NULL operator is used to compare a value with a NULL value.
UNIQUE	The UNIQUE operator searches every row of a specified table for uniqueness <i>noduplicates</i> .

Set Operators

Set operators are used to combine results of two queries into a single result. Data type should be same for both the tables.

- **UNION** – It combines the results of two or more Select statements. However it will eliminate duplicate rows.
- **UNION ALL** – This operator is similar to Union but it also shows the duplicate rows.
- **INTERSECT** – Intersect operation is used to combine the two SELECT statements, and it returns the records, which are common from both SELECT statements. In case of Intersect, the number of columns and datatype must be same in both the tables.
- **MINUS** – Minus operation combines result of two SELECT statements and return only those results,

which belong to first set of result and eliminate the rows in second statement from the output of first.

SAP HANA - SQL FUNCTIONS

There are various SQL functions provided by SAP HANA database –

- Numeric Functions
- String Functions
- Fulltext Functions
- Datetime Functions
- Aggregate Functions
- Data Type Conversion Functions
- Window Functions
- Series Data Functions
- Miscellaneous Functions

Numeric Functions

These are inbuilt numeric functions in SQL and use in scripting. It takes numeric values or strings with numeric characters and return numeric values.

- **ABS** – It returns the absolute value of a numeric argument.

```
Example – SELECT ABS (-1) "abs" FROM TEST;  
abs  
1
```

ACOS, ASIN, ATAN, ATAN2 These functions return trigonometric value of the argument

- **BINTOHEX** – It converts a Binary value to a hexadecimal value.
- **BITAND** – It performs an AND operation on bits of passed argument.
- **BITCOUNT** – It performs the count of number of set bits in an argument.
- **BITNOT** – It performs a bitwise NOT operation on the bits of argument.
- **BITOR** – It perform an OR operation on bits of passed argument.
- **BITSET** – It is used to set bits to 1 in <target_num> from the <start_bit> position.
- **BITUNSET** – It is used to set bits to 0 in <target_num> from the <start_bit> position.
- **BITXOR** – It performs XOR operation on bits of passed argument.
- **CEIL** – It returns the first integer that is greater or equal to the passed value.
- COS, COSH, COT (These functions return trigonometric value of the argument)

- **EXP** – It returns the result of the base of natural logarithms e raised to the power of passed value.
- **FLOOR** – It returns the largest integer not greater than the numeric argument.
- **HEXTOBIN** – It converts a hexadecimal value to a binary value.
- **LN** – It returns the natural logarithm of the argument.
- **LOG** – It returns the algorithm value of a passed positive value. Both base and log value should be positive.

Various other numeric functions can also be used – MOD, POWER, RAND, ROUND, SIGN, SIN, SINH, SQRT, TAN, TANH, UMINUS

String Functions

Various SQL string functions can be used in HANA with SQL scripting. Most common string functions are –

- **ASCII** – It returns integer ASCII value of passed string.
- **CHAR** – It returns the character associated with passed ASCII value.
- **CONCAT** – It is Concatenation operator and returns the combined passed strings.
- **LCASE** – It converts all character of a string to Lower case.
- **LEFT** – It returns the first characters of a passed string as per mentioned value.
- **LENGTH** – It returns the number of characters in passed string.
- **LOCATE** – It returns the position of substring within passed string.
- **LOWER** – It converts all characters in string to lowercase.
- **NCHAR** – It returns the Unicode character with passed integer value.
- **REPLACE** – It searches in passed original string for all occurrences of search string and replaces them with replace string.
- **RIGHT** – It returns the rightmost passed value characters of mentioned string.
- **UPPER** – It converts all characters in passed string to uppercase.
- **UCASE** – It is identical to UPPER function. It converts all characters in passed string to uppercase.

Other string functions that can be used are – LPAD, LTRIM, RTRIM, STRTOBIN, SUBSTR_AFTER, SUBSTR_BEFORE, SUBSTRING, TRIM, UNICODE, RPAD, BINTOSTR

Date Time functions

There are various Date Time functions that can be used in HANA in SQL scripts. Most common Date Time functions are –

- **CURRENT_DATE** – It returns the current local system date.
- **CURRENT_TIME** – It returns the current local system time.

- **CURRENT_TIMESTAMP** – It returns the current local system timestamp details *YYYY – MM – DDHH : MM : SS : FF*.
- **CURRENT_UTCDATE** – It returns current UTC *GreenwichMeanDate* date.
- **CURRENT_UTCTIME** – It returns current UTC *GreenwichMeanTime* time.
- **CURRENT_UTCTIMESTAMP**
- **DAYOFMONTH** – It returns the integer value of day in passed date in argument.
- **HOUR** – It returns integer value of hour in passed time in argument.
- **YEAR** – It returns the year value of passed date.

Other Date Time functions are – DAYOFYEAR, DAYNAME, DAYS_BETWEEN, EXTRACT, NANO100_BETWEEN, NEXT_DAY, NOW, QUARTER, SECOND, SECONDS_BETWEEN, UTCTOLOCAL, WEEK, WEEKDAY, WORKDAYS_BETWEEN, ISOWEEK, LAST_DAY, LOCALTOUTC, MINUTE, MONTH, MONTHNAME, ADD_DAYS, ADD_MONTHS, ADD_SECONDS, ADD_WORKDAYS

Data Type Conversion Functions

These functions are used to convert one data type to other or to perform a check if conversion is possible or not.

Most common data type conversion functions used in HANA in SQL scripts –

- **CAST** – It returns the value of an expression converted to a supplied data type.
- **TO_ALPHANUM** – It converts a passed value to an ALPHANUM data type
- **TO_REAL** – It converts a value to a REAL data type.
- **TO_TIME** – It converts a passed time string to the TIME data type.
- **TO_CLOB** – It converts a value to a CLOB data type.

Other similar Data Type conversion functions are – TO_BIGINT, TO_BINARY, TO_BLOB, TO_DATE, TO_DATS, TO_DECIMAL, TO_DOUBLE, TO_FIXEDCHAR, TO_INT, TO_INTEGER, TO_NCLOB, TO_NVARCHAR, TO_TIMESTAMP, TO_TINYINT, TO_VARCHAR, TO_SECONDDATE, TO_SMALLDECIMAL, TO_SMALLINT

There are also various Windows and other miscellaneous functions that can be used in HANA SQL scripts.

- **Current_Schema** – It returns a string containing the current schema name.
- **Session_User** – It returns the user name of current session

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

Subqueries in Expressions

A subquery as an expression is a Select statement. When it is used in an expression, it returns a zero or a single value.

A subquery is used to return data that will be used in the main query as a condition to further restrict the data to be retrieved.

Subqueries can be used with the SELECT, INSERT, UPDATE, and DELETE statements along with the operators like =, <, >, >=, <=, IN, BETWEEN etc.

There are a few rules that subqueries must follow –

- Subqueries must be enclosed within parentheses.
- A subquery can have only one column in the SELECT clause, unless multiple columns are in the main query for the subquery to compare its selected columns.
- An ORDER BY cannot be used in a subquery, although the main query can use an ORDER BY. The GROUP BY can be used to perform the same function as the ORDER BY in a subquery.
- Subqueries that return more than one row can only be used with multiple value operators, such as the IN operator.
- The SELECT list cannot include any references to values that evaluate to a BLOB, ARRAY, CLOB, or NCLOB.
- A subquery cannot be immediately enclosed in a set function.
- The BETWEEN operator cannot be used with a subquery; however, the BETWEEN operator can be used within the subquery.

Subqueries with the SELECT Statement

Subqueries are most frequently used with the SELECT statement. The basic syntax is as follows –

Example

```
SELECT * FROM CUSTOMERS
WHERE ID IN (SELECT ID
FROM CUSTOMERS
WHERE SALARY > 4500) ;
```

ID	NAME	AGE	ADDRESS	SALARY
4	Chaitali	25	Mumbai	6500.00
5	Hardik	27	Bhopal	8500.00
7	Muffy	24	Indore	10000.00

SAP HANA - SQL STORED PROCEDURES

A procedure allows you to group the SQL statement into a single block. Stored Procedures are used to achieve certain result across applications. The set of SQL statements and the logic that is used to perform some specific task are stored in SQL Stored Procedures. These stored procedures are executed by applications to perform that task.

Stored Procedures can return data in the form of output parameters *integerorcharacter* or a cursor variable. It can also result in set of Select statements, which are used by other Stored Procedures.

Stored Procedures are also used for performance optimization as it contains series of SQL statements and results from one set of statement determines next set of statements to be executed. Stored procedures prevent users to see the complexity and details of tables in a database. As Stored procedures contain certain business logic, so users need to execute or call the procedure name.

No need to keep reissuing the individual statements but can refer to the database procedure.

Sample Statement to Create Procedures

```
Create procedure prc_name (in inp integer, out opt "EFASION"."ARTICLE_LOOKUP")
as
begin
opt = select * from "EFASION"."ARTICLE_LOOKUP" where article_id = :inp ;
end;
```

SAP HANA - SQL SEQUENCES

A sequence is a set of integers 1, 2, 3, that are generated in order on demand. Sequences are frequently used in databases because many applications require each row in a table to contain a unique value, and sequences provide an easy way to generate them.

Using AUTO_INCREMENT column

The simplest way in MySQL to use sequences is to define a column as AUTO_INCREMENT and leave rest of the things to MySQL to take care.

Example

Try out the following example. This will create table and after that it will insert few rows in this table where it is not required to give record ID because it is auto-incremented by MySQL.

```
mysql> CREATE TABLE INSECT
-> (
-> id INT UNSIGNED NOT NULL AUTO_INCREMENT,
-> PRIMARY KEY (id),
-> name VARCHAR(30) NOT NULL, # type of insect
-> date DATE NOT NULL, # date collected
-> origin VARCHAR(30) NOT NULL # where collected
);

Query OK, 0 rows affected (0.02 sec)

mysql> INSERT INTO INSECT (id,name,date,origin) VALUES
-> (NULL,'housefly','2001-09-10','kitchen'),
-> (NULL,'millipede','2001-09-10','driveway'),
-> (NULL,'grasshopper','2001-09-10','front yard');

Query OK, 3 rows affected (0.02 sec)
Records: 3 Duplicates: 0 Warnings: 0
mysql> SELECT * FROM INSECT ORDER BY id;
```

+	-----	-----	-----	-----
---	-------	-------	-------	-------

```

| id | name          | date        | origin      |
+---+-----+-----+-----+
| 1  | housefly     | 2001-09-10 | kitchen    |
| 2  | millipede    | 2001-09-10 | driveway   |
| 3  | grasshopper  | 2001-09-10 | front yard |
+---+-----+-----+-----+
3 rows in set (0.00 sec)

```

Obtain AUTO_INCREMENT Values

LAST_INSERT_ID is a SQL function, so you can use it from within any client that understands how to issue SQL statements. Otherwise, PERL and PHP scripts provide exclusive functions to retrieve auto-incremented value of last record.

PERL Example

Use the mysql_insertid attribute to obtain the AUTO_INCREMENT value generated by a query. This attribute is accessed through either a database handle or a statement handle, depending on how you issue the query. The following example references it through the database handle –

```

$dbh->do ("INSERT INTO INSECT (name,date,origin)
VALUES('moth','2001-09-14','windowsill')");
my $seq = $dbh->{mysql_insertid};

```

PHP Example

After issuing a query that generates an AUTO_INCREMENT value, retrieve the value by calling mysql_insert_id –

```

mysql_query ("INSERT INTO INSECT (name,date,origin)
VALUES('moth','2001-09-14','windowsill')", $conn_id);
$seq = mysql_insert_id ($conn_id);

```

Renumbering an Existing Sequence

There may be a case when you have deleted many records from a table and you want to re-sequence all the records. This can be done by using a simple trick but you should be very careful to do so if your table is having join, with other table.

If you determine that resequencing an AUTO_INCREMENT column is unavoidable, the way to do it is to drop the column from the table, then add it again. The following example shows how to renumber the id values in the insect table using this technique –

```

mysql> ALTER TABLE INSECT DROP id;
mysql> ALTER TABLE insect
-> ADD id INT UNSIGNED NOT NULL AUTO_INCREMENT FIRST,
-> ADD PRIMARY KEY (id);

```

Starting a Sequence at a Particular Value

By default, MySQL will start sequence from 1 but you can specify any other number as well at the time of table

creation. Following is the example where MySQL will start sequence from 100.

```
mysql> CREATE TABLE INSECT
-> (
-> id INT UNSIGNED NOT NULL AUTO_INCREMENT = 100,
-> PRIMARY KEY (id),
-> name VARCHAR(30) NOT NULL, # type of insect
-> date DATE NOT NULL, # date collected
-> origin VARCHAR(30) NOT NULL # where collected
);
```

Alternatively, you can create the table and then set the initial sequence value with ALTER TABLE.

SAP HANA - SQL TRIGGERS

Triggers are stored programs, which are automatically executed or fired when some events occur. Triggers are, in fact, written to be executed in response to any of the following events –

- A database manipulation *DML* statement *DELETE, INSERT, or UPDATE*.
- A database definition *DDL* statement *CREATE, ALTER, or DROP*.
- A database operation *SERVERERROR, LOGON, LOGOFF, STARTUP, or SHUTDOWN*.

Triggers could be defined on the table, view, schema, or database with which the event is associated.

Benefits of Triggers

Triggers can be written for the following purposes –

- Generating some derived column values automatically
- Enforcing referential integrity
- Event logging and storing information on table access
- Auditing
- Synchronous replication of tables
- Imposing security authorizations
- Preventing invalid transactions

SAP HANA - SQL SYNONYM

SQL Synonyms is an alias for a table or a Schema object in a database. They are used to protect client applications from the changes made to name or location of an object.

Synonyms permit applications to function irrespective of user who owns the table and which database holds the table or object.

Create Synonym statement is used create a Synonym for a table, view, package, procedure, objects, etc.

Example

There is a table Customer of efashion, located on a Server1. To access this from Server2, a client application would have to use name as Server1.efashion.Customer. Now we change the location of Customer table the client application would have to be modified to reflect the change.

To address these we can create a synonym of Customer table Cust_Table on Server2 for the table on Server1. So now client application has to use the single-part name Cust_Table to reference this table. Now, if the location of this table changes, you will have to modify the synonym to point to the new location of the table.

As there is no ALTER SYNONYM statement, you have to drop the synonym Cust_Table and then re-create the synonym with the same name and point the synonym to the new location of Customer table.

Public Synonyms

Public Synonyms are owned by PUBLIC schema in a database. Public synonyms can be referenced by all users in the database. They are created by the application owner for the tables and other objects such as procedures and packages so the users of the application can see the objects.

Syntax

```
CREATE PUBLIC SYNONYM Cust_table for efashion.Customer;
```

To create a PUBLIC Synonym, you have to use keyword PUBLIC as shown.

Private Synonyms

Private Synonyms are used in a database schema to hide the true name of a table, procedure, view or any other database object.

Private synonyms can be referenced only by the schema that owns the table or object.

Syntax

```
CREATE SYNONYM Cust_table FOR efashion.Customer;
```

Drop a Synonym

Synonyms can be dropped using DROP Synonym command. If you are dropping a public Synonym, you have to use the keyword **public** in the drop statement.

Syntax

```
DROP PUBLIC Synonym Cust_table;  
DROP Synonym Cust_table;
```

SAP HANA - SQL EXPLAIN PLANS

SQL explain plans are used to generate detail explanation of SQL statements. They are used to evaluate execution plan that SAP HANA database follows to execute the SQL statements.

The results of explain plan are stored into EXPLAIN_PLAN_TABLE for evaluation. To use Explain Plan, passed SQL query must be a data manipulation language *DML*.

Common DML Statements

- **SELECT** – retrieve data from the a database
- **INSERT** – insert data into a table
- **UPDATE** – updates existing data within a table

SQL Explain Plans cannot be used with DDL and DCL SQL statements.

EXPLAIN PLAN TABLE in database

EXPLAIN PLAN_TABLE in database consists of multiple columns. Few common column names – OPERATOR_NAME, OPERATOR_ID, PARENT_OPERATOR_ID, LEVEL and POSITION, etc.

COLUMN SEARCH value tells the starting position of column engine operators.

ROW SEARCH value tells the starting position of row engine operators.

To create an EXPLAIN PLAN STATEMENT for a SQL query

```
EXPLAIN PLAN SET STATEMENT_NAME = 'statement_name' FOR <SQL DML statement>
```

To see values in EXPLAIN PLAN TABLE

```
SELECT Operator_Name, Operator_ID  
FROM explain_plan_table  
WHERE statement_name = 'statement_name';
```

To delete a statement in EXPLAIN PLAN TABLE

```
DELETE FROM explain_plan_table WHERE statement_name = 'TPC-H Q10';
```

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.

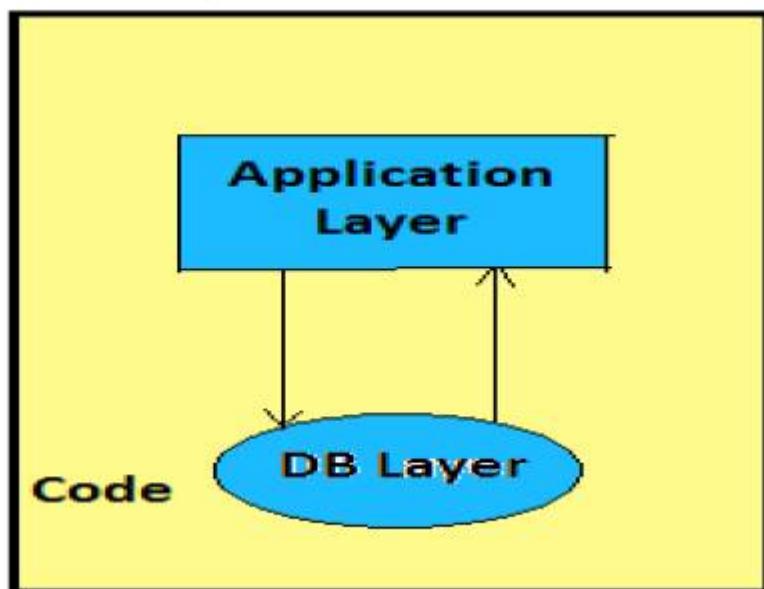
SAP HANA - SQL SCRIPT

Set of SQL statements for HANA database which allows developer to pass complex logic into database is called SQL Script. SQL Script is known as collections of SQL extensions. These extension are Data Extensions, Function Extensions, and Procedure Extension.

SQL Script supports stored Functions and Procedures and that allows pushing complex parts of Application logic to database.

Main benefit of using SQL Script is to allow the execution of complex calculations inside SAP HANA database. Using SQL Scripts in place of single query enables Functions to return multiple values. Complex SQL functions can be further decomposed into smaller functions. SQL Script provides control logic that is not available in single SQL statement.

SQL Script



SQL Scripts are used to achieve performance optimization in HANA by executing scripts at DB layer –

- By Executing SQL scripts at database layer, it eliminates need to transfer large amount of data from database to application.
- Calculations are executed at database layer to get benefits of HANA database like column operations, parallel processing of queries, etc.

Integration with Information Modeler

While using SQL scripts in Information Modeler, below given are applied to Procedures –

- Input parameters can be of scalar or table type.
- Output parameters must be of table types.
- Table types required for the signature are generated automatically.

SQL Scripts with Calculation Views

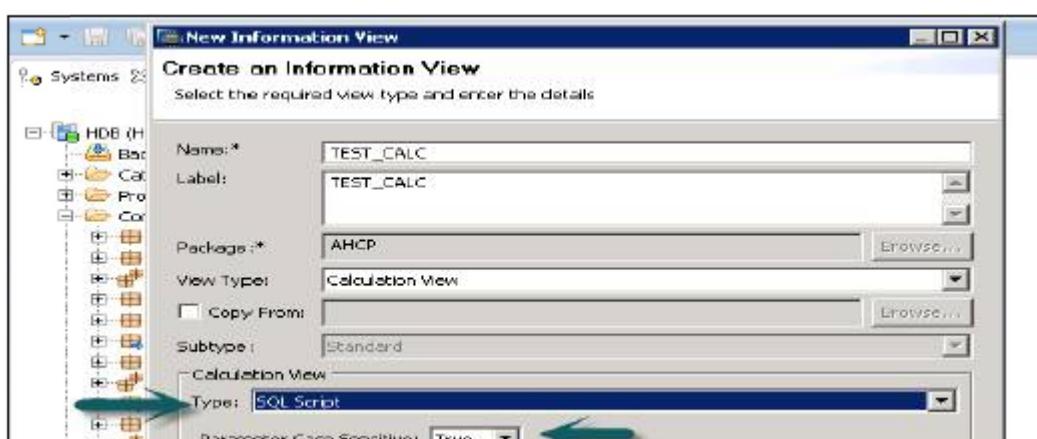
SQL script are used to create script based Calculation views. Type SQL statements against existing raw tables or column store. Define output structure, activation of view creates table type as per structure.

How to create a Calculation View with SQL Script?

Launch SAP HANA studio. Expand the content node → Select a package where you want to create the new Calculation view. Right Click → New Calculation View End of the navigation path → Provide name and description.



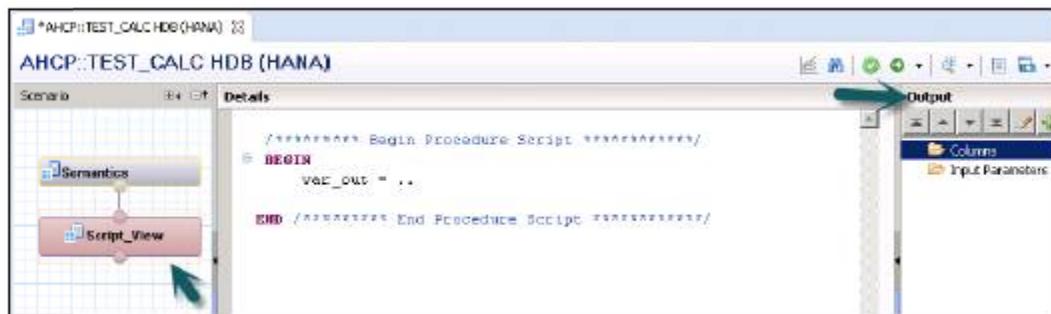
Select calculation view type → from Type dropdown list, select SQL Script → Set Parameter Case Sensitive to True or False based on how you require the naming convention for the output parameters of the calculation view → Choose Finish.



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.

In the Source pane, choose the columns that you want to add to the output → To add selective columns to the output, then select those columns and choose Add. To add all columns of an object to the output, then select the object and choose Add → Finish.

Activate the script-based calculation view – In the SAP HANA Modeler perspective – Save and Activate - to activate the current view and redeploy the affected objects if an active version of the affected object exists. Otherwise, only the current view is activated.

Save and activate all – to activate the current view along with the required and affected objects.

In the SAP HANA Development perspective – In Project Explorer view, select the required object. In the context menu, select Start of the navigation path Team Next navigation step Activate End of the navigation path.

SQL Scripting in HANA Information Modeler is used to create complex Calculation Views, which are not possible to create using GUI option.