Public

SAP Data Warehouse Cloud

Beta Content Tutorial

Version 1.1 – August 21, 2019

THE BEST RUN

TABLE OF CONTENTS

INTRODUCTION

The Best Run Bikes sample data model will help you explore and learn your way around in SAP Data Warehouse Cloud.

Please make use of the in-app help and available enablement materials on www.sapdatawarehouse.cloud in case you need further assistance beyond this document.

STORY LINE

The customer "Best Run Bikes" is looking for an analytics solution for their Sales department. As the name suggests they are experts in Bikes, but not so much in Data Warehousing and Analytics solutions.

These are the analytics they need:

1. Year-over-Year Sales Comparison

They want to compare the current years' company sales with the previous year. But they have all the data in Excel and visualizing this data with formulas is a painful task.

2. Sales Per Region

Due to an increase in the number of sales, the company wants to understand how the different regions are performing. Based on this visualization, the marketing team wants to identify the regions which are doing good as well as the regions which need attention or better marketing campaigns.

3. Best Sales Representative

It is time for the company to reward the best Sales Representative for all the hard-work that has resulted in a remarkable sales quota. For this purpose, the company needs to have a visualization that shows the sales revenue per sales representative.

SAP Data Warehouse Cloud and SAP Analytics Cloud will help solving the above challenges.

Follows these steps to create the above reports with SAP Data Warehouse Cloud using the Best Run Bikes data model:

1. Prepare data source in Data Warehouse Cloud
2. Create models using this data in Data Warehouse Cloud
3. Create SAP Analytics Cloud story using the models

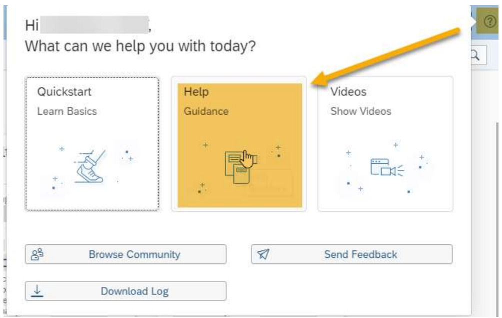
SPACES

To work with SAP Data Warehouse Cloud, a Space is needed as a pre-requisite. Find more information on Spaces here.

Either use a space you have already created or create a new space first.

To create a new space, navigate to 

and create a new space.
Please consult the help documentation if you require further assistance or use the in-app product help using “F1” anytime or use the following navigation path to access the help:



For our tutorial, we will use the Space *DWC_CONTENT*.

PREPARE THE DATA MODEL

You have two options to consume the sample content with SAP Data Warehouse Cloud:

1. Use an SAP HANA as a remote source: Import the schema including the tables into your SAP HANA database and access it through a remote connection. Currently only SAP HANA 2.0 is supported. To access the SAP HANA system from SAP Data Warehouse Cloud you need a Data Provisioning Agent. Find the setup instructions in the Chapter “Data Provisioning Agent Setup” in the appendix.

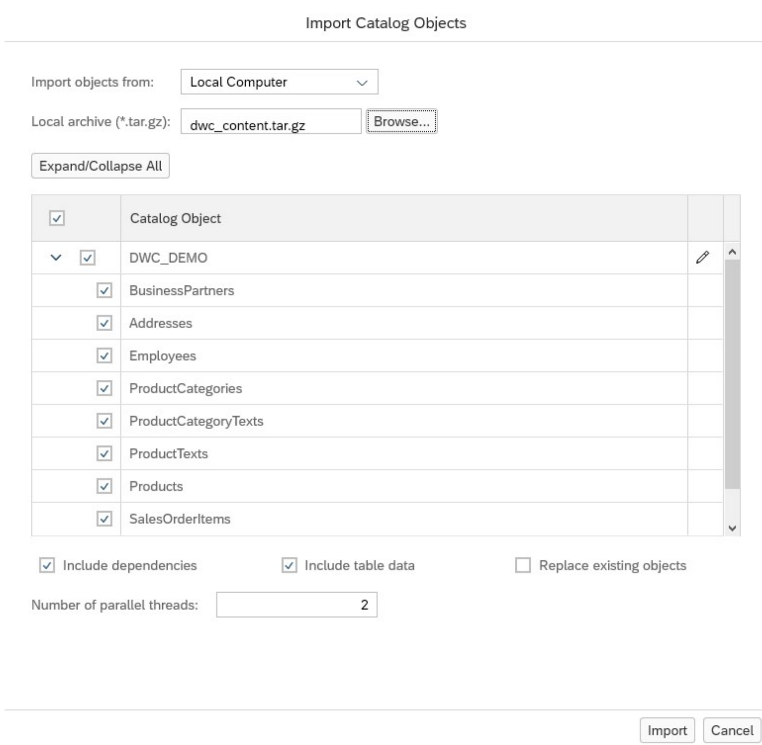
2. Use File upload: Upload the data model as well as the data into SAP Data Warehouse Cloud via flat files.

Please find the SAP HANA export and the files on https://github.sap.com/data-warehouse-cloud-content-beta.

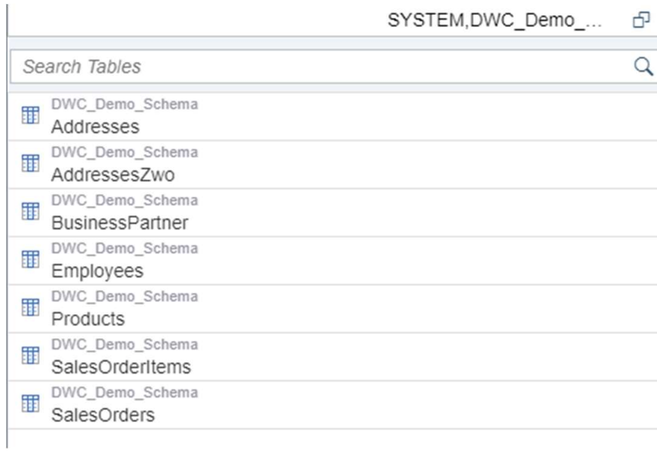
1. SAP HANA: Load the schema and tables including the sample data in your SAP HANA DB

The Data Provisioning Agent must be setup first, to be able to access the Bikes sample data models (tables) from SAP Data Warehouse Cloud to your SAP HANA as a remote source. Please read the chapter “Data Provisioning Agent Setup” in the appendix.

- Download and unzip dwc_content.zip from GitHub
- Logon to your SAP HANA system and import the package *dwc_content.tar.gz*



- Click on Import
- Once imported, the tables will appear in the schema *DWC_Demo*

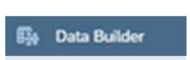


2. Upload local files to SAP Data Warehouse Cloud

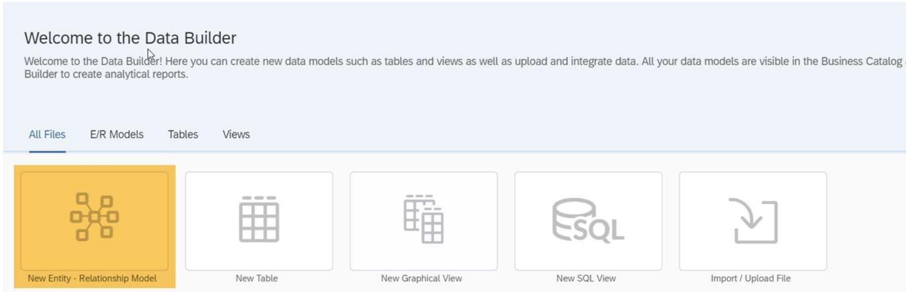
As alternative to accessing the data deployed in your SAP HANA, you can directly upload the data model and data into SAP Data Warehouse Cloud. Or you can explore this option as an additional learning experience for the SAP Data Warehouse Cloud.

Entity-Relationship Models in SAP Data Warehouse Cloud can be expressed in CSN (Core Schema Notation), stored in JSON format.
In this chapter you will learn how to upload a CSN file (BikeSalesModel.json), specifying the Bike Sales data model to an Entity–Relationship Model in SAP Data Warehouse Cloud. Deploying this Entity-Relationship Model will generate the corresponding local tables of the model for you, which you need as a base for the modeling experience.
After that you can upload the CSV files containing the data into these generated local tables.

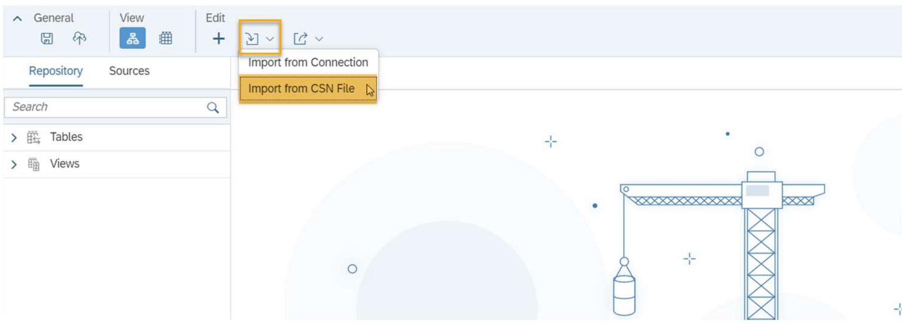
As shared in the introduction, a Space is a prerequisite to upload CSN files into a new Entity-Relationship model.

- Navigate to the 

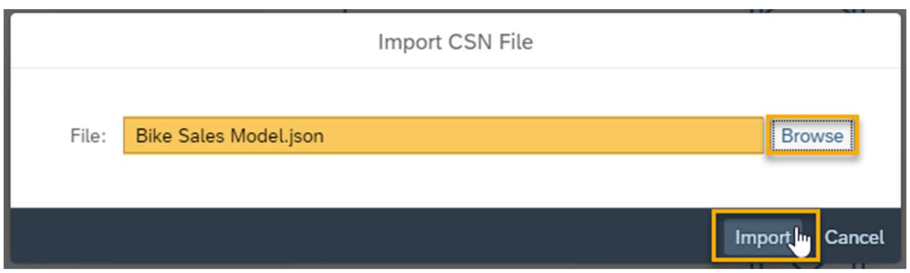
- If this is your first visit to the Data Builder, you need to select a Space. Select any space – for this guide “DWC_CONTENT” is used.
- Click on “New Entity – Relationship Model”



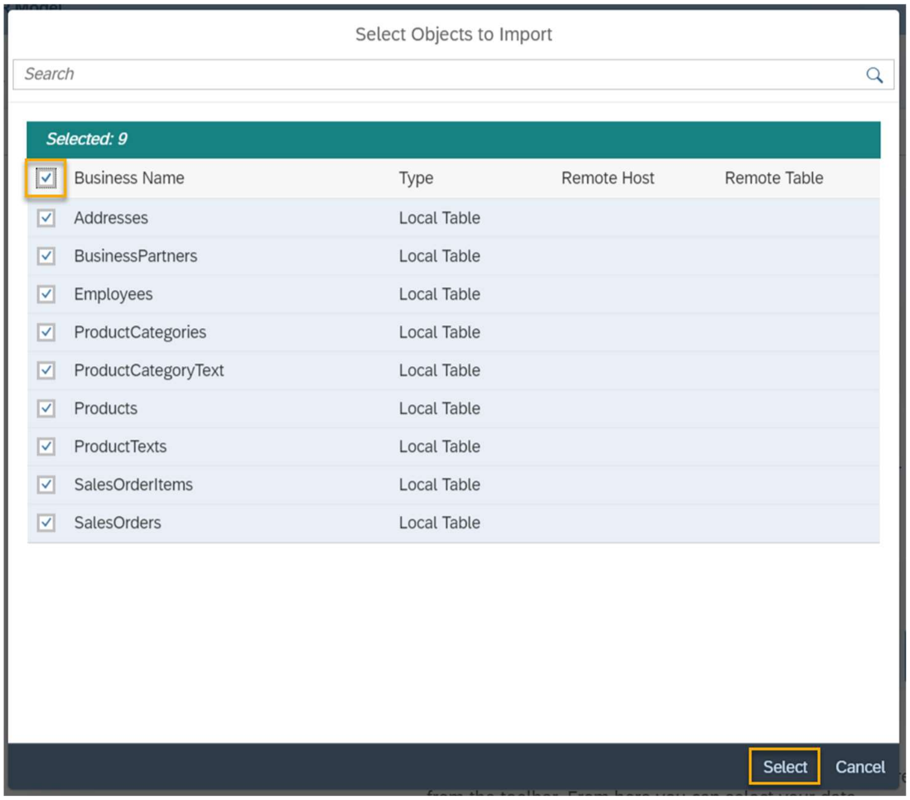
- To import the CSN file (Bike Sales Model.json) click on the import icon and chose *Import from CSN File*.



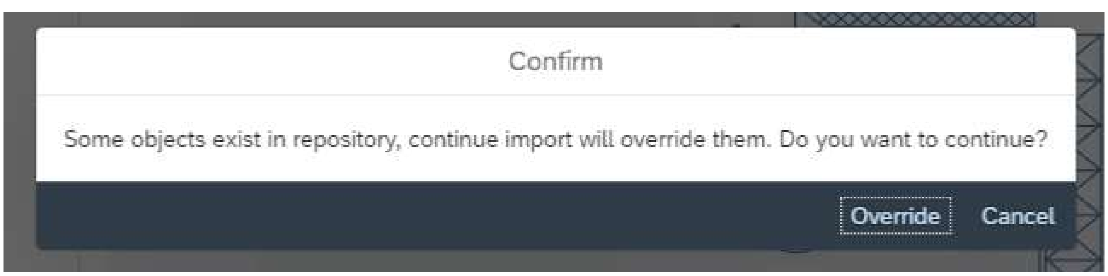
- In the opening dialogue click *Browse* to select the CSN file (Bike Sales Model.json) which you have downloaded from github.



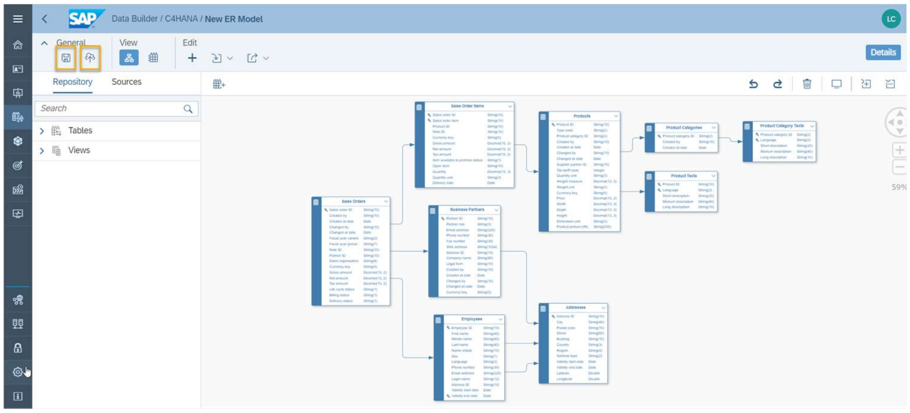
- In the next step select all tables and import them to the Entity – Relationship Modeler by pressing *Select*.

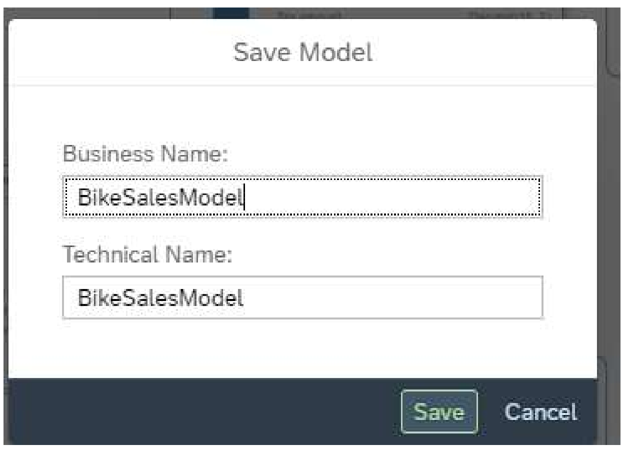


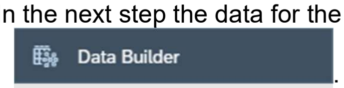
- In case the CSN file had been imported and deployed before, you will receive the following warning as the tables already exist in your system. In case the local table definitions have not been changed you can safely choose *Override*.



- Now you can see the complete model with all tables and associations. It should look like the following screen shot. Save the ER model providing a name, “BikeSalesModel”, and deploy it. By deploying it, the local tables are generated.

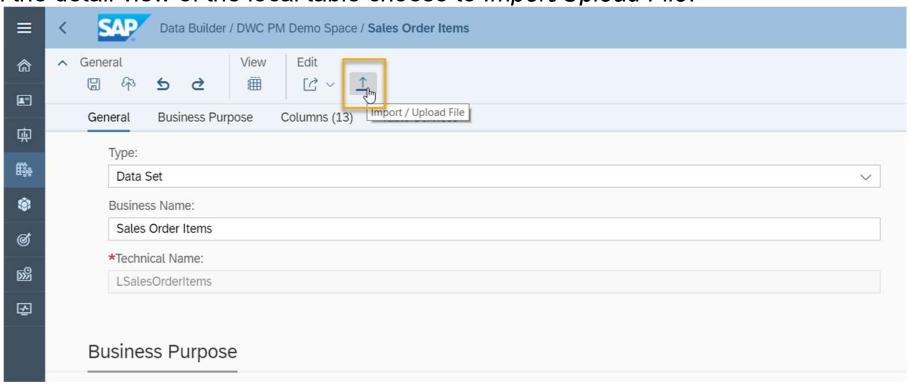




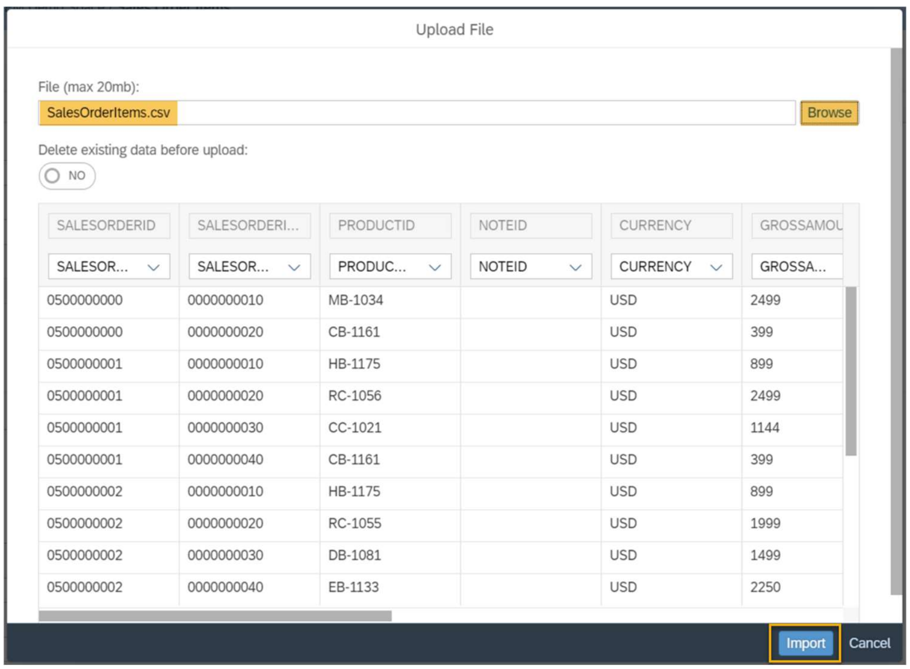
- In the next step the data for the tables needs to be uploaded. Therefore, go back to the 

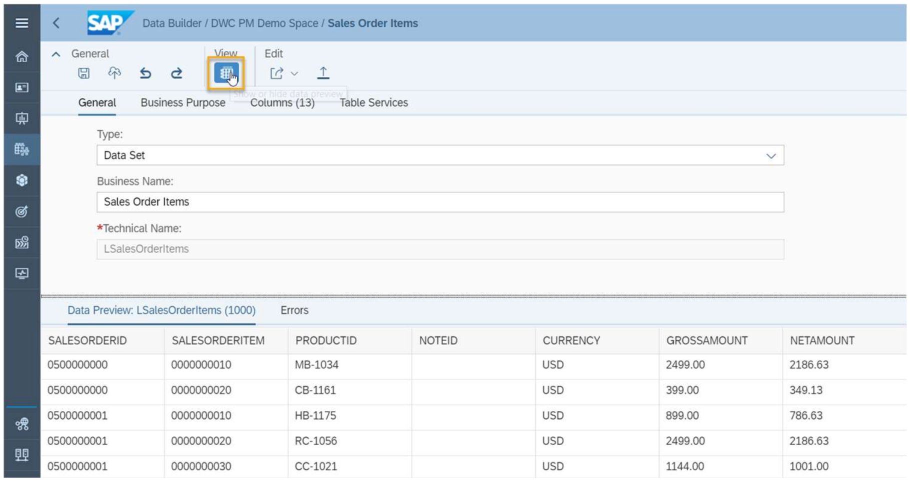
.
- Now search for your created local tables in the bottom panel. To start click on the *SalesOrderItems* table.

- In the detail view of the local table choose to *Import/Upload File*.



- Search for your CSV-file named *SalesOrderItems.csv*. The columns should be mapped automatically. Click *Import* to import the data.



- You can also preview the content of table. The preview pane will appear on the bottom part of the page displaying the data.
- 

- Repeat these steps for the tables *Addresses, BusinessPartners, Employees, ProductCategories, ProductCategoryTexts, Products, ProductTexts* and *SalesOrders*.

After all CSV files have been uploaded you have successfully prepared the data model.

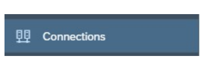
USE THE BEST RUN BIKES SALES DEMO CONTENT TO BUILD THE ANALYTICS

Use sample data

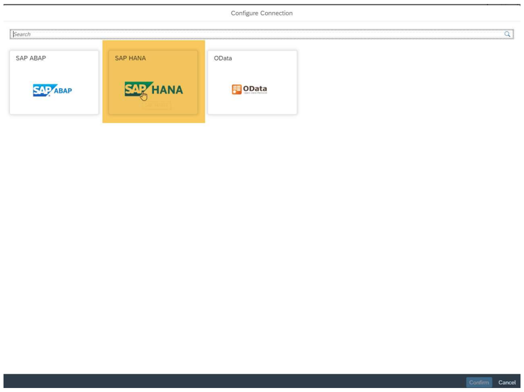
To use the Bikes Sales sample data model, a connection to SAP HANA from SAP Data Warehouse Cloud needs to be added. In case SAP HANA will not be used, the tables that have been created using "Import / Upload File" capabilities will be used instead.

Connect to SAP HANA

1.) Create a connection to your SAP HANA database

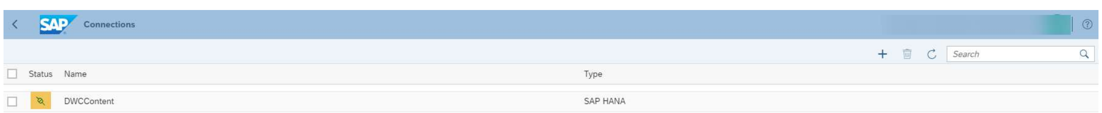
Navigate to 

and create a new connection to SAP HANA:


Configure Connection
Search
SAP ABAP
SAP ABAP
SAP HANA
SAP HANA
OData
OData
Confirm
Cancel

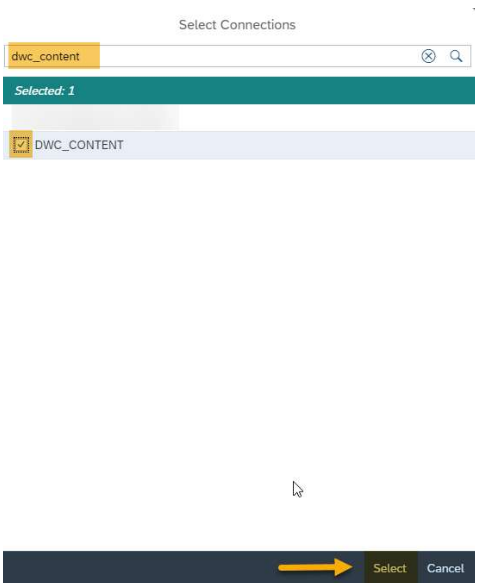
Enter your SAP HANA system details. You find this information in the SAP HANA Service Dashboard: Use the information provided under Endpoints – Direct SQL Connectivity, or in case you are using HANA Studio, find this information in the *Properties* of your SAP HANA system.
Click *Confirm* to create the connection:

After the connection is created you can check if it is working by clicking the icon in the "Status" column next to your connection:



2.) Assign the connection to a space in 

Add Connection: Press + and search for the connection you have just created. Mark the checkbox and click *Select* to assign the connection to your space



You have successfully assigned your connection:

Use CSV Files

As the CSV files have been uploaded to SAP Data Warehouse Cloud in the “Prepare the Data Model” step, there are no further steps necessary to start the data modeling.

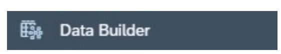
Data Modeling – Basics

Now as the data source is prepared either through having connected to the SAP HANA or through having uploaded the CSV files, let’s start the data modeling in SAP Data Warehouse Cloud.

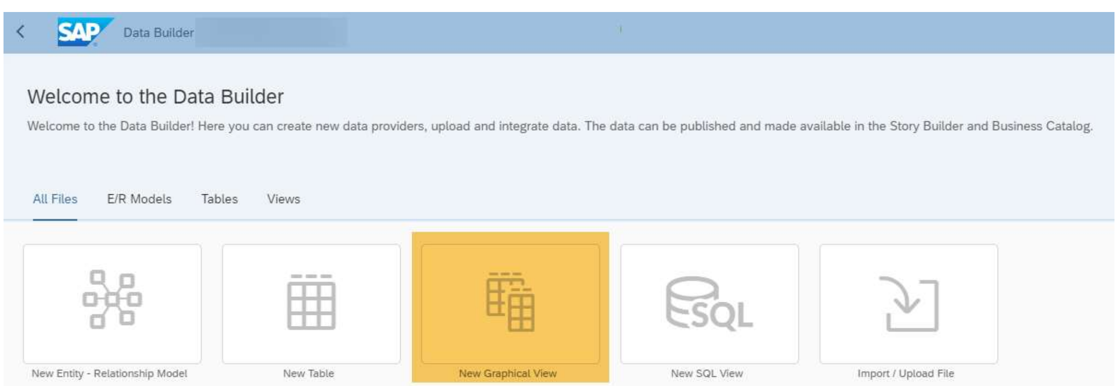
SAP HANA will be used as a source for the walkthrough and screenshots. It will be always mentioned how to apply a step using CSV files.

The following steps will navigate through the solution to introduce and learn the functionality to create a report with the minimum steps needed. This means we basically will use the facts *SalesOrders* and *SalesOrderItems* only.
Subsequently we will enhance the model to create the reports laid out in the “Story Line” chapter at the beginning of this document in the chapter “Data Modeling – Full Modell” following this one.

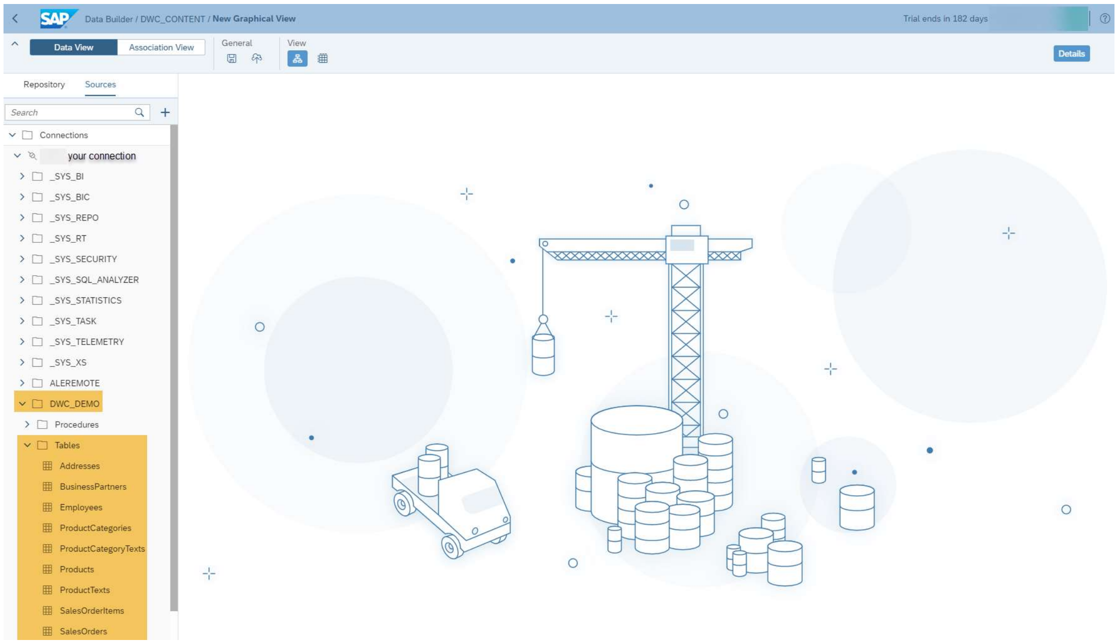
Let’s start:

1.) To build a data model, navigate to the 

2.) If this is your first visit to the Data Builder you need to select a Space. Select the Space you just have assigned the connection to – for this guide “DWC_CONTENT” is used.
3.) In the entry screen different modeling options are available: let’s start with the Graphical View:



4.) To access the sample data model in the remote HANA system navigate to “Sources” in the left panel and then open “Connections” and drill further to “DWC_CONTENT” in your HANA system:

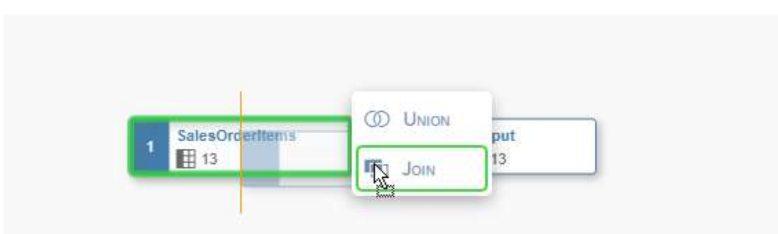


CSV files: In the left tab, navigate to *Repository* and select the tables you have created while uploading the CSV files:

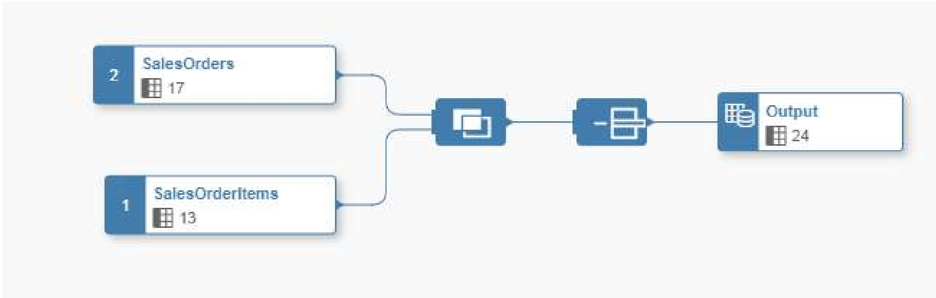
Remark: if you already have used an object, you can directly access it from the *Repository*.

5.) To start modeling, Drag and Drop a table. Build the data foundation with the *SalesOrderItems* and *SalesOrders* table: Drop the *SalesOrderItems* to the modeling area:

6.) Add the *SalesOrders* by dragging and dropping it on the *SalesOrderItems* in the canvas. An additional option to choose the join criteria appears, choose *Join*:

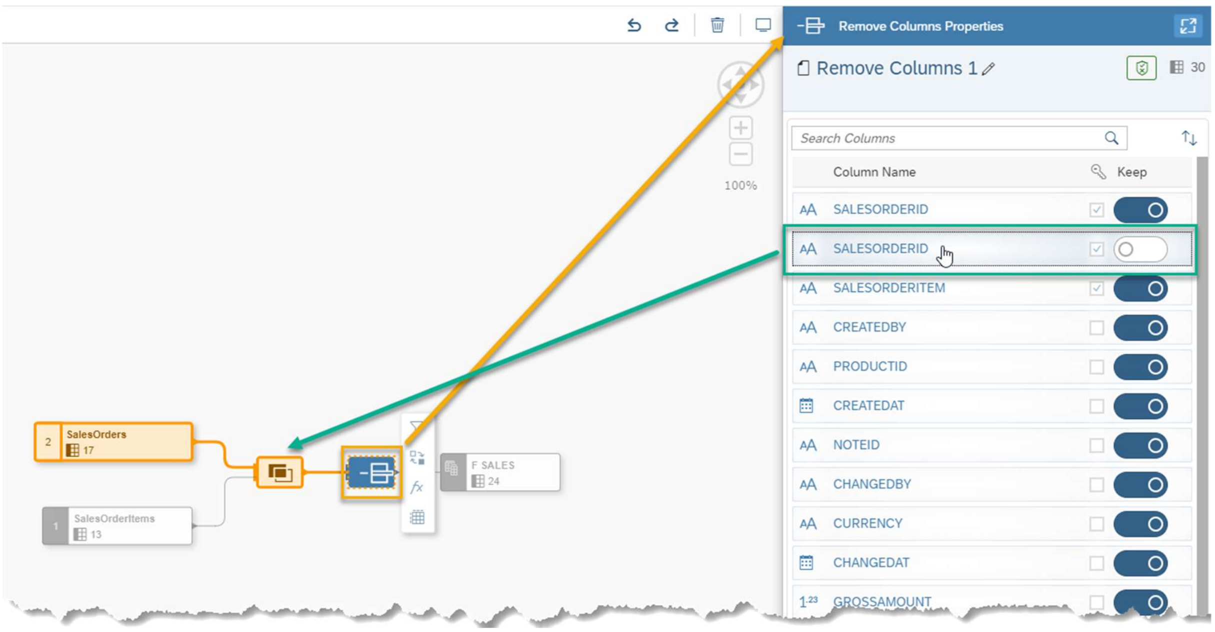


As a result you now created a first join:



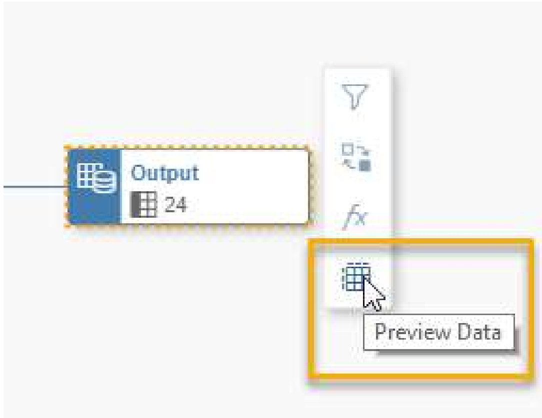
Next to the join, a "Remove Columns" properties has been added automatically. This has been added to remove duplicated columns from a join activity. This property can also always added manually.

In this case the "Remove Columns" looks like the following:

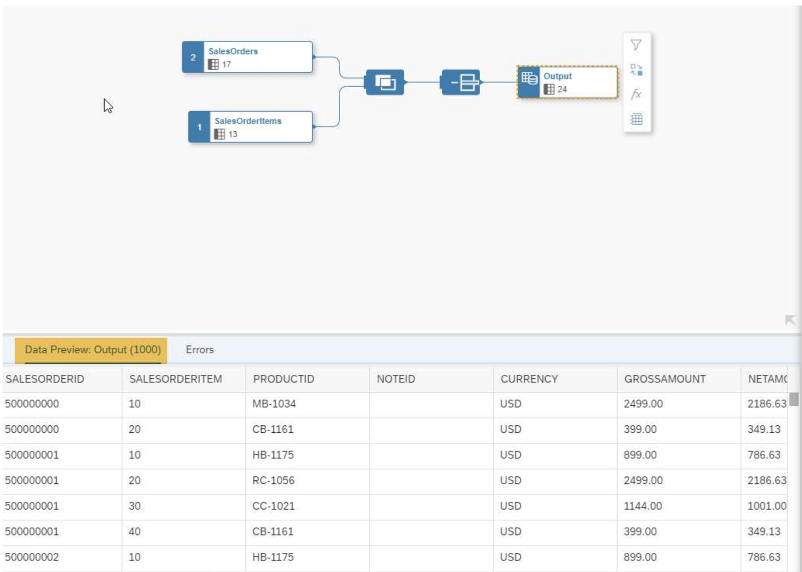


The screenshots shows the details of the "Remove Columns Properties". If you mark a column it is shown from which view a column is sourced.

You may already preview the data: click in the *Output* and in the now appearing menu, click *Preview Data*:

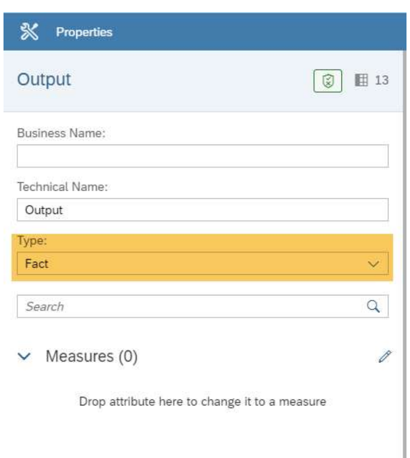


The preview is shown in the lower area of the screen:

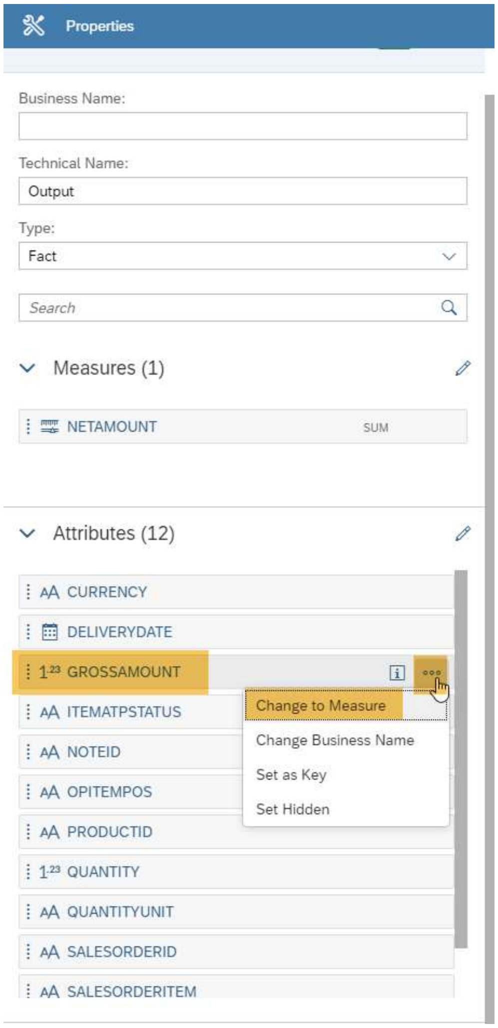


7.) Let’s already see how this looks in an report. Before being able to create the first report, the model needs to be saved and deployed. But before doing this, the view type needs to be set accordingly.

In the *Properties* pane on the right change the *Type* to *Fact*



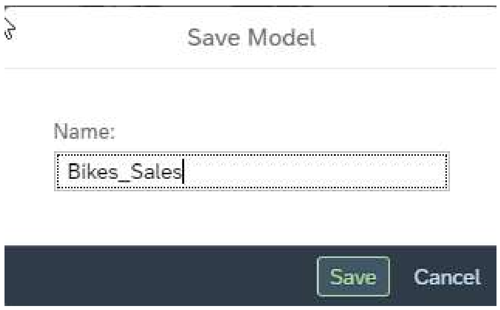
8.) After having changed the type to *Fact*, you need to move the columns that are measures from the *Attributes* to the *Measures*. You can either drag & drop a column or you use the context menu in the right area of a column (indicated by the three dots, once the mouse cursor hoovers this area) and click *Change to Measure*.
Make GROSSAMOUNT, NETAMOUNT, TAXAMOUNT and QUANTITY a measure.

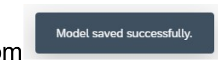


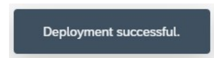
9.) Now let's save and the deploy the model. Either click the save button (1) first and then deploy (2), or directly save and deploy clicking the deploy button (2): 

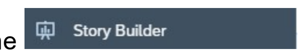
.
A model can only be used for reporting once it is deployed.

Your will be prompted to enter a *Model Name*:



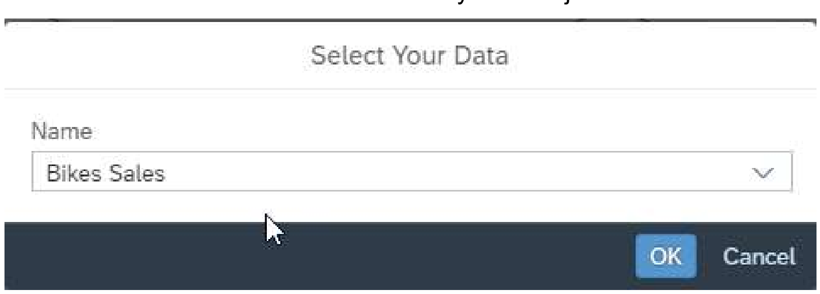
Click Save and wait until the following message appears at the bottom 

After a successful deployment the following message will be shown 

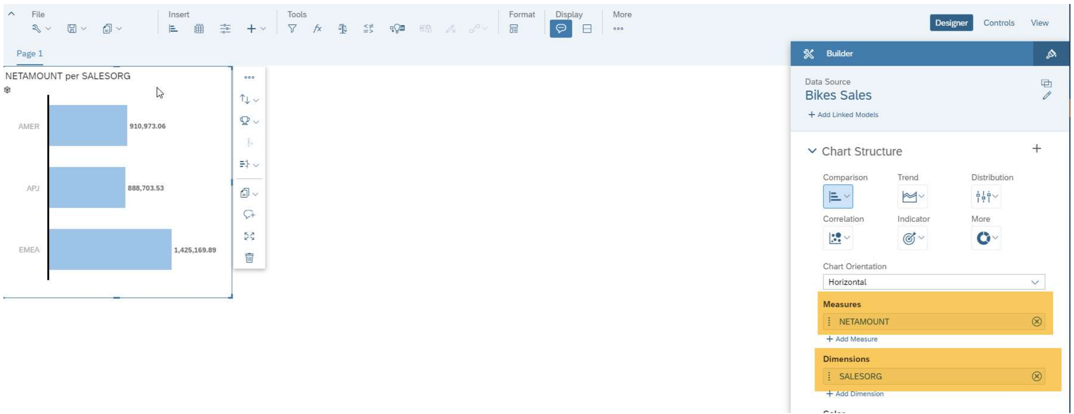
10.) Let's create the first visualization: navigate to the 

11.) Create a new story:

12.) Choose the Bikes Sales data model that you have just created and click *OK*:



13.) Add a chart and assign a measure and a dimension. Choose NETAMOUNT as Measure and SALESORG as a Dimension and save your story.



You have successfully created your first model and report in SAP Data Warehouse Cloud.

Data Modeling – enhanced Model

Now as the foundation is created and you have been introduced to the basics to connect to a data source, create a data model using a join and building an SAP Analytics Cloud story, this knowledge will be used to enhance the model and explore further functions and capabilities of SAP Data Warehouse Cloud.

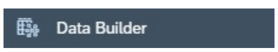
After these steps, all reports that have been requested by the Sales Department can be delivered.

Let's continue and enhance the model with the following:

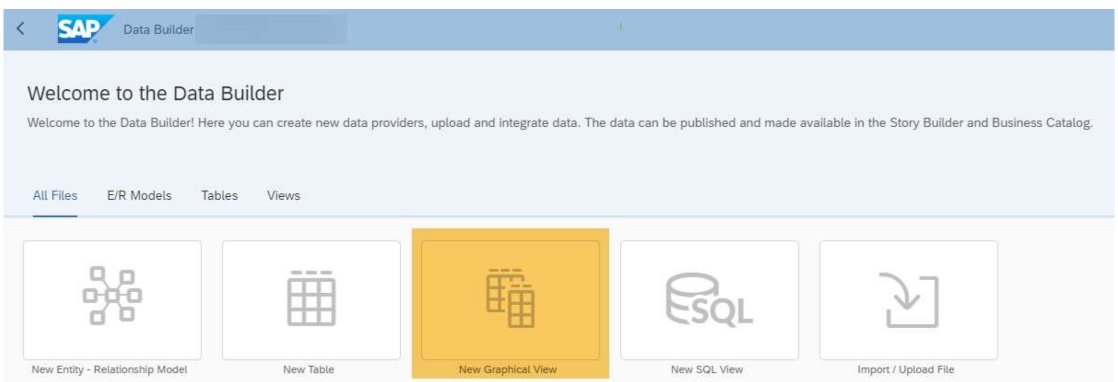
- master data objects for products, business partners, employees,
- add those master data objects to our Bikes Sales model
- rename measures and master data dimension that are needed for our reports to speaking names for the reports
- use further functionalities like calculated columns

Master Data

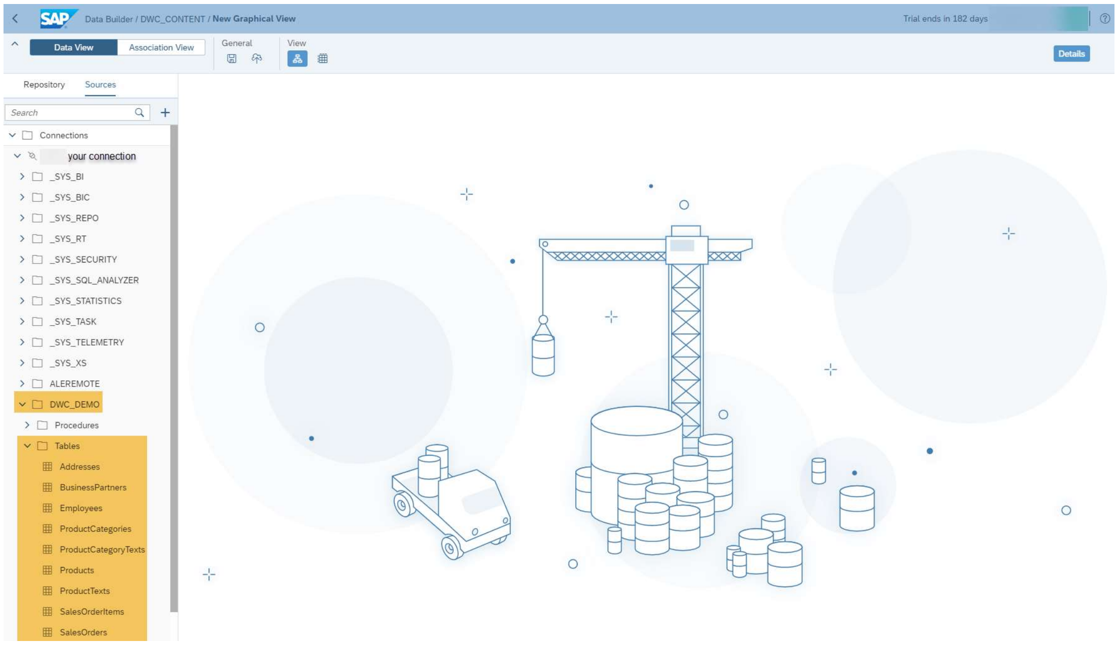
- To create master data objects for products, business partners and employees navigate to the



- Let's create a new graphical view.

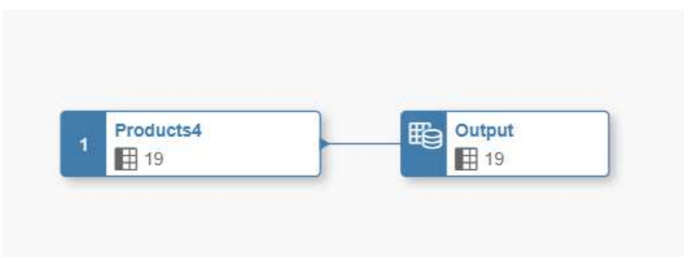


- Find your tables in the left panel. To access the data models form the remote HANA connection search in “Connections”, for CVS-Files search in “Repository”

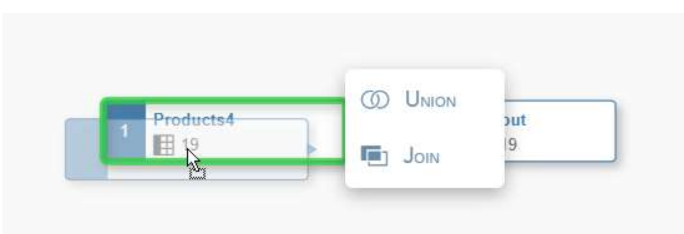


- Let’s start with the products master data model. Therefore, you need the tables *Products*, *ProductCategories* and the corresponding *ProductTexts* and *ProductCaregoryTexts*. You can find further information about the associations and attributes in the Appendix.

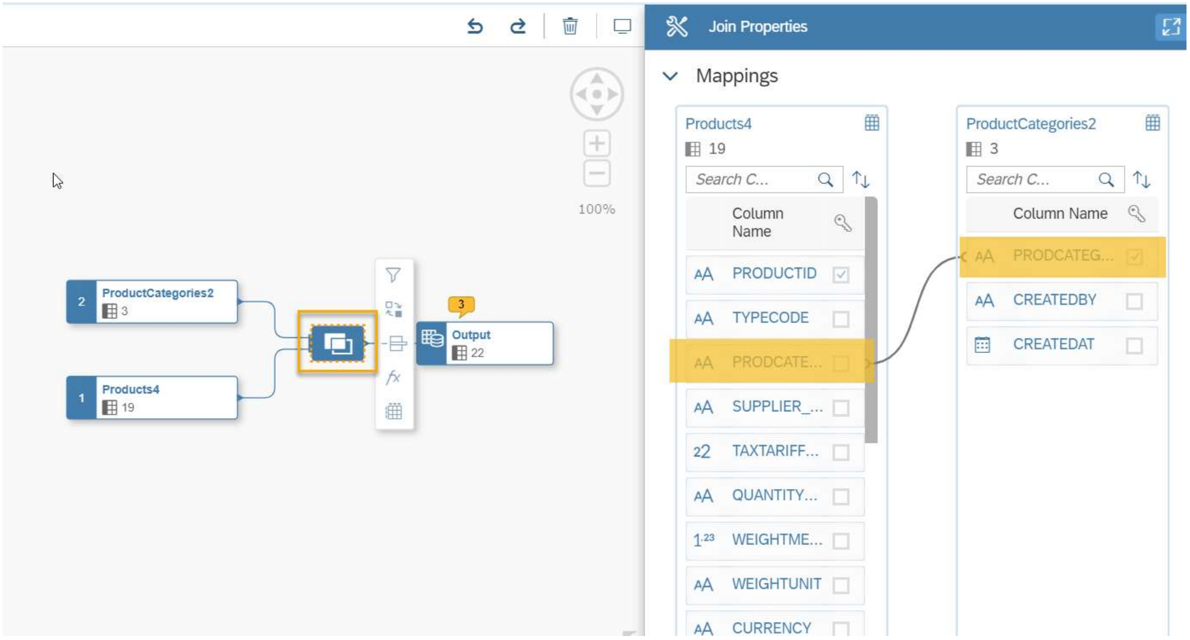
- To start drag and drop the *Products* table into the modeling area.



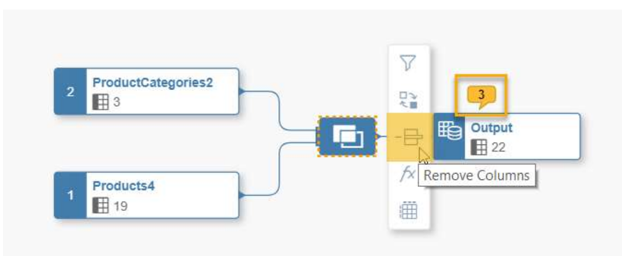
- To perform a join, drag the table *ProductCategories* onto the *Products* in the canvas.



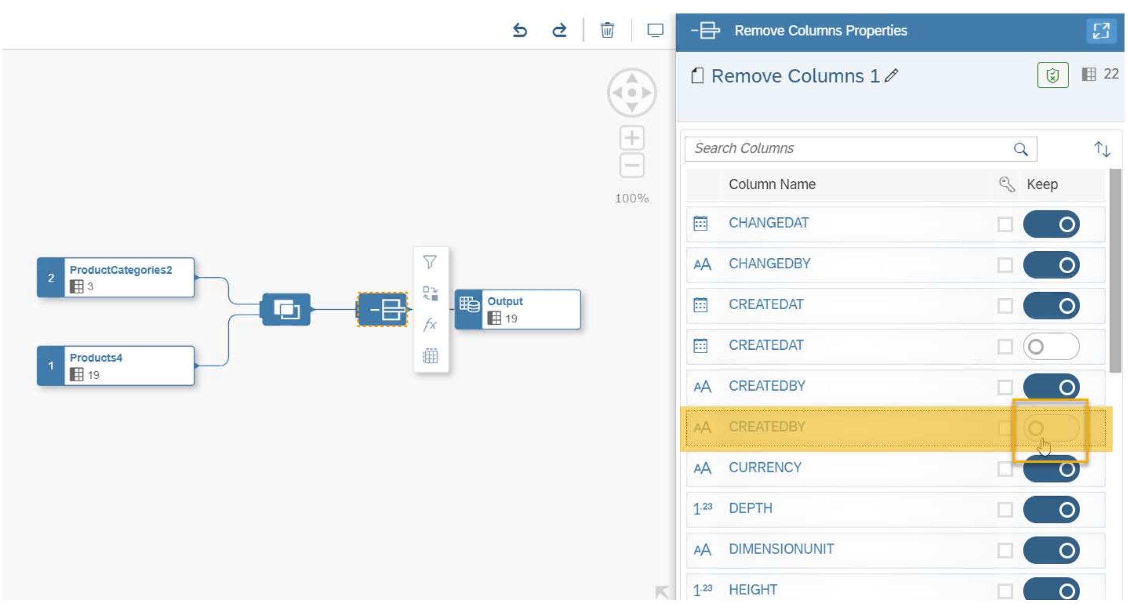
- Now you have created your join. The mapping has been made automatically, but you can also add it manually in the “Join Properties” panel on the right. Map the PRODUCTCATEGORYID by drag and drop.



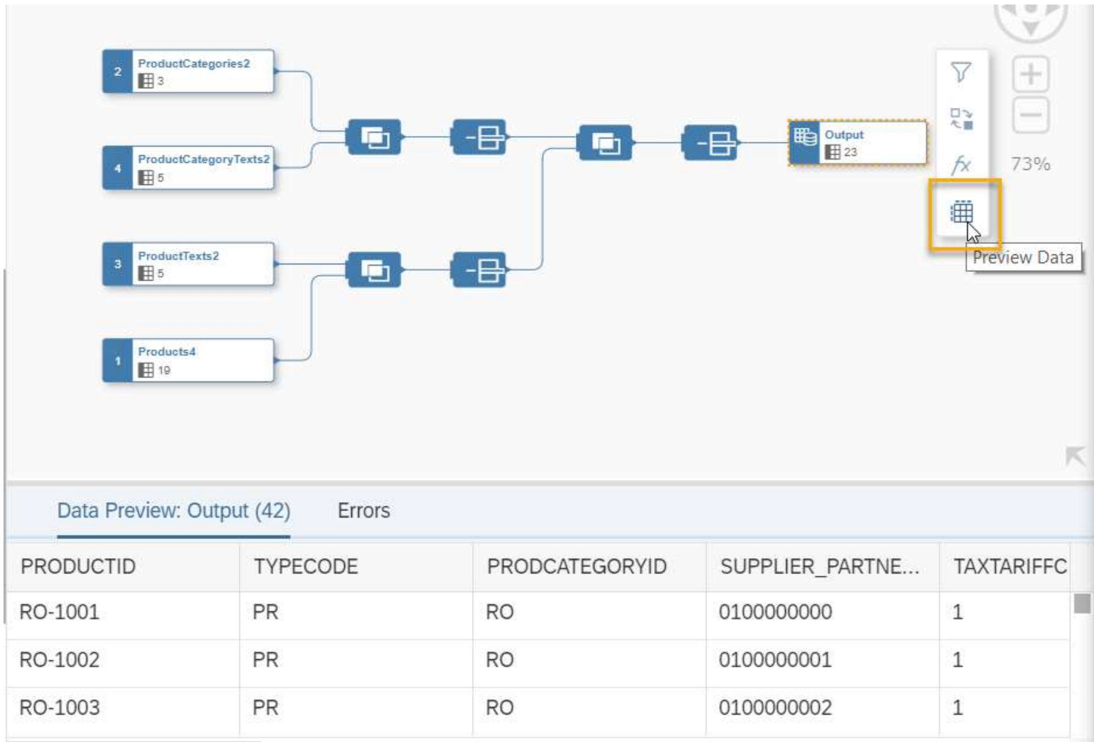
- Let’s remove redundant columns indicated by the text field. Click in the “Join” and in the now appearing menu, click “Remove Columns”. In the right panel “Remove Columns Properties” now disable the doubeled columns



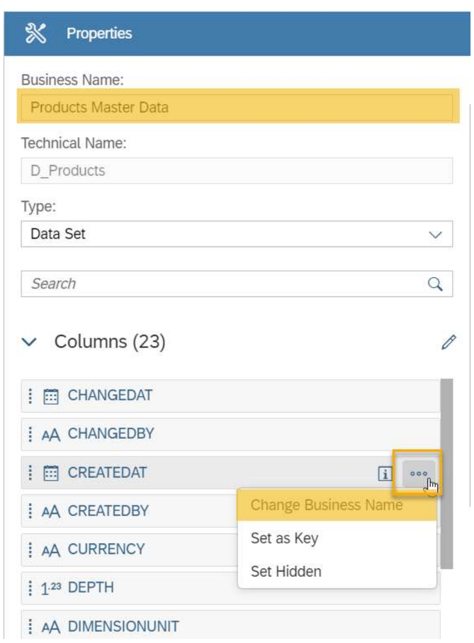
.



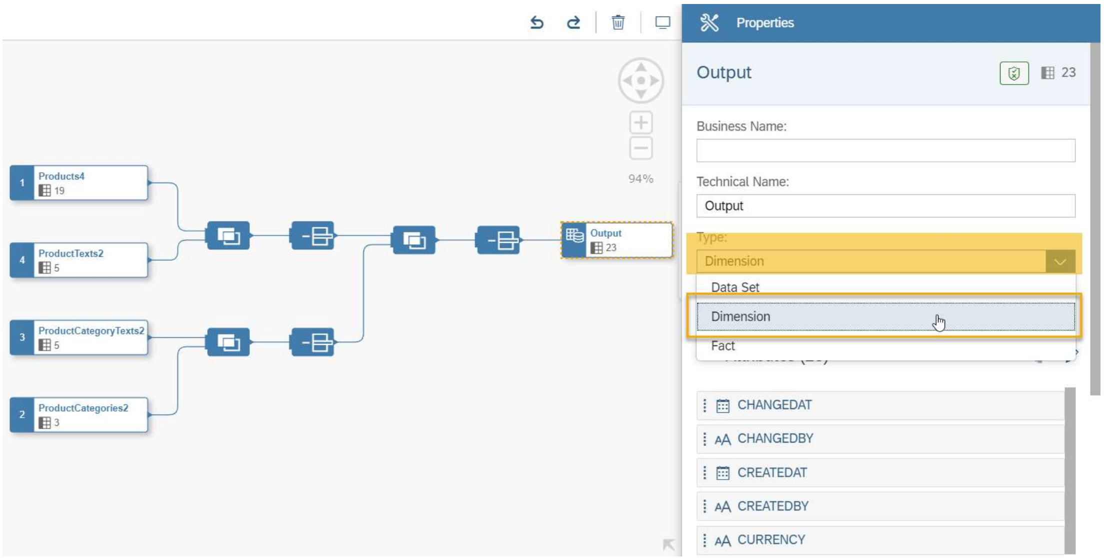
- Repeat the steeps to join the corresponding *ProductTexts* onto the *Products* and the *ProductCategoryTexts* onto *ProductCategories*. Then you can already preview your products master data.



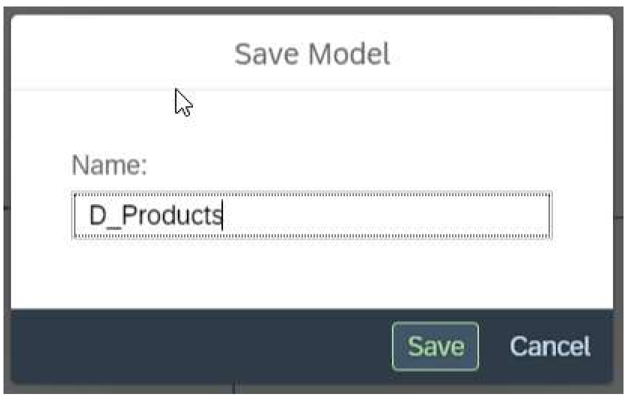
- To enable more speaking names, you can change the business names. Go to the "Properties" panel on the right and change the business name of the model as well as the properties.



- Before saving it, you need to make the model a dimension. In the right panel "Properties" change the type to "Dimension".

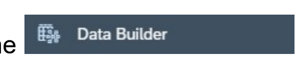


- Let's save and deploy the model. Name the new model "D_Products".



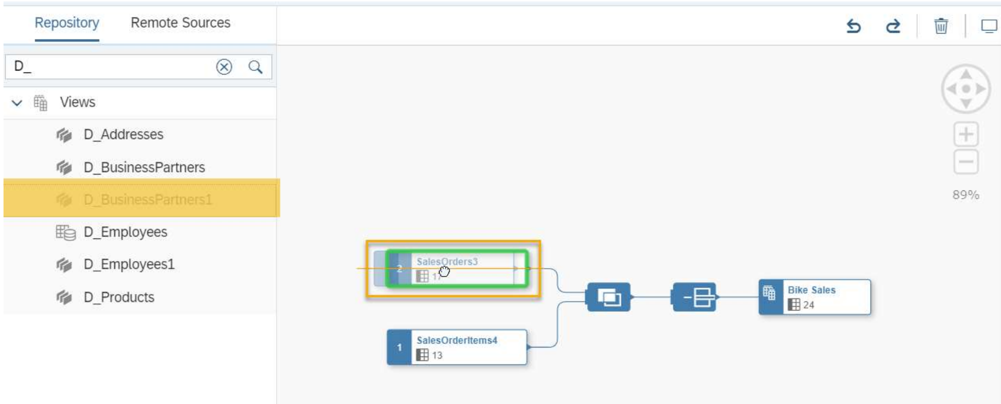
- To create the master data for the business partners, use the same schema. To join the tables *BusinessPartners* and *Addresses*, map the columns ADDRESSID. Save the model as a dimension with the name "D_BusinessPartners".

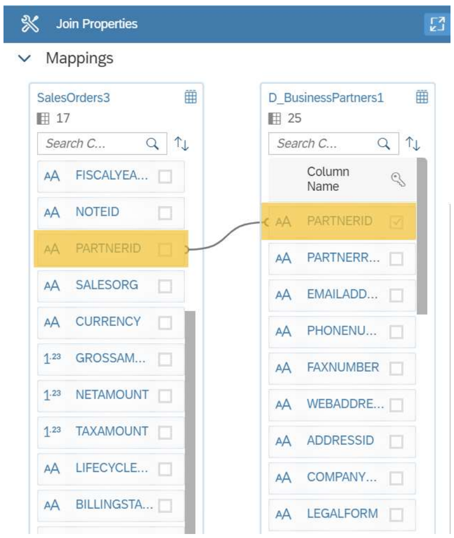
Enhance Fact View

- The new models now need to be added to the main model. Navigate to the 

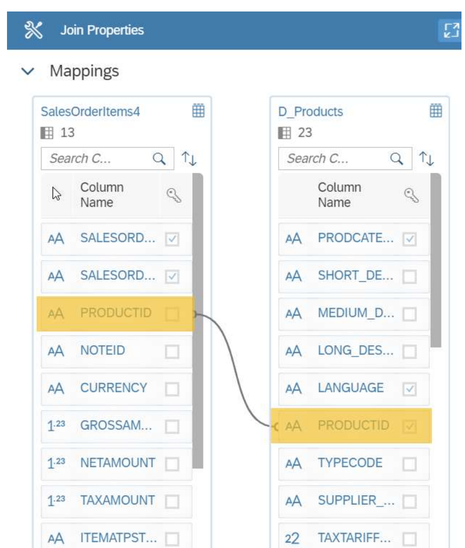
and choose the *Bikes_Sales* model. On the left side search for your created dimensions in "Repository", "Views".

- Drag and drop your *D_BusinessPartners* onto the *SalesOrders* in the modeling area. The join should be on the columns PARTNERID.

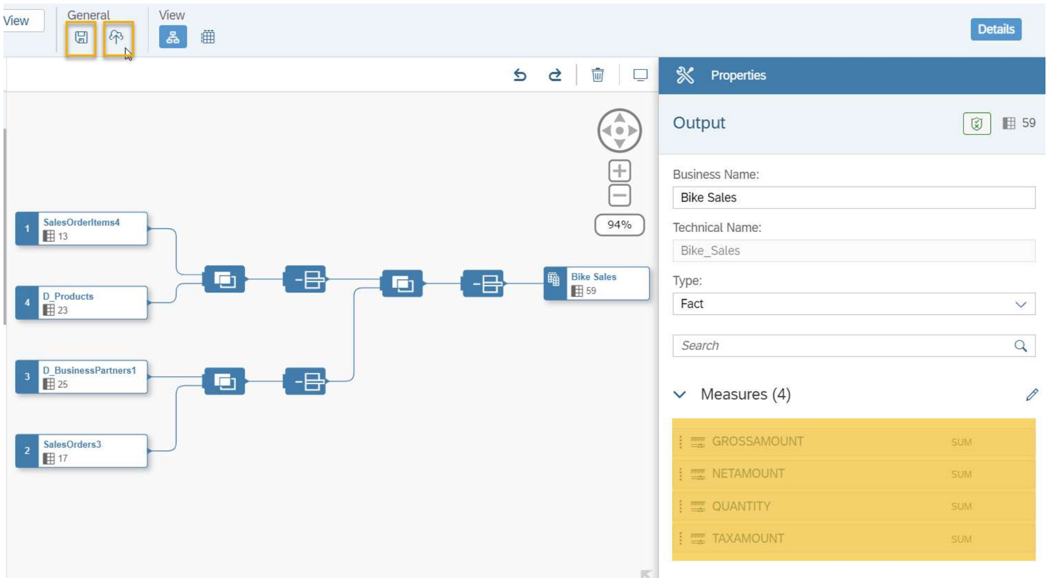




- The *D_Products* model should be joined with the *SalesOrderItems* table. Therefore, map them on the PRODUCTID as shown before.

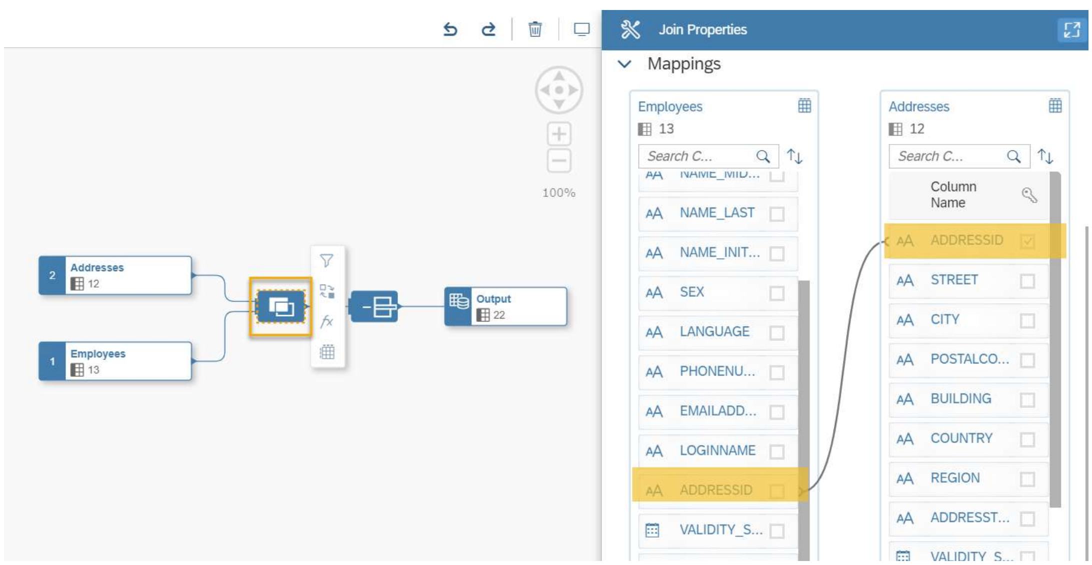


- Disable any redundant columns but make sure you disable the ones from the master data and not from the *SalesOders* or *SalesOrderItems*. Then control the measures. If your model looks like the following save and deploy.

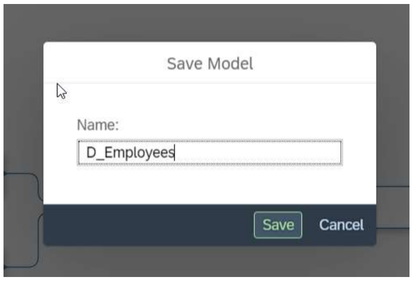


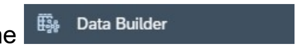
- For finding our Best Sales Representative, you first need to prepare the master data for the employees. The Employees table needs to be enriched with data about the addresses. For this reason, we need to create another new view, as explained in the steps before.

- This time join the Employees and the Addresses table on the column ADDRESSID.



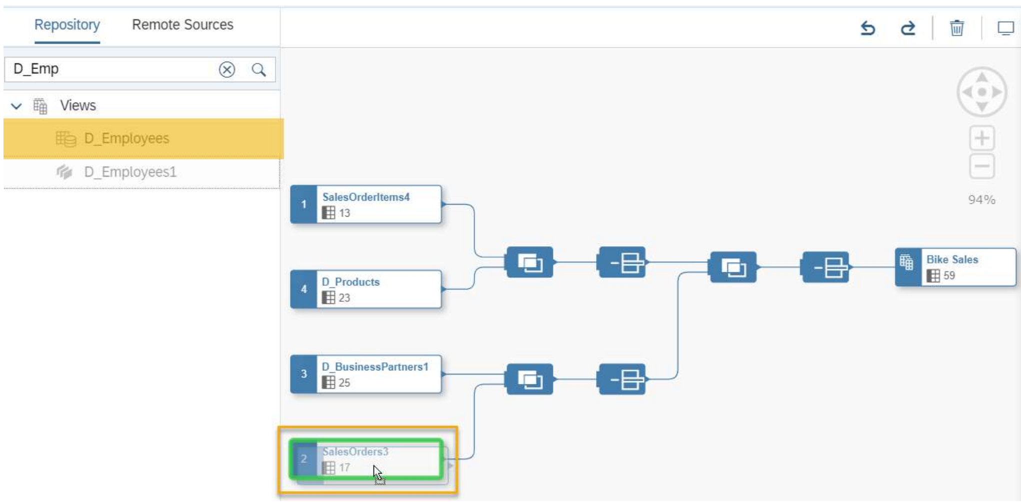
- When finished save and deploy the model under the name “D_Employees”.



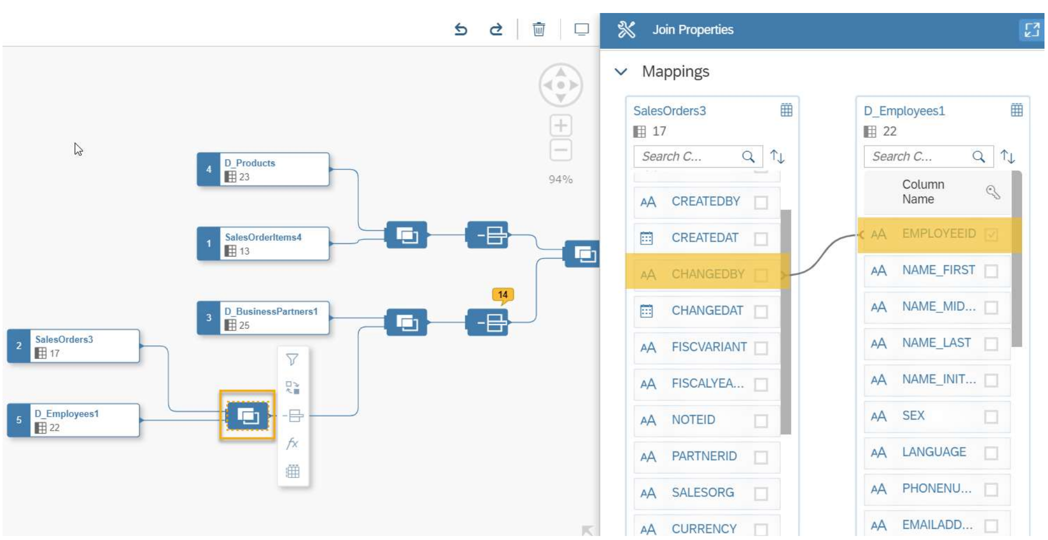
- To connect the employees to the sales orders, go back to the 

and click on your *Bikes_Sales* Fact to edit it.

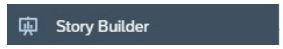
- Let’s add our Employees dimension. In the left panel go to “Repository” and find your *D_Employees* in the “Views”. Drag and drop it onto the *SalesOrders* in the modeling area.



- This time map the EMPLOYEEID to the CREATEDBY column of the *SalesOrder* table. Don’t forget to remove redundant columns from the model.

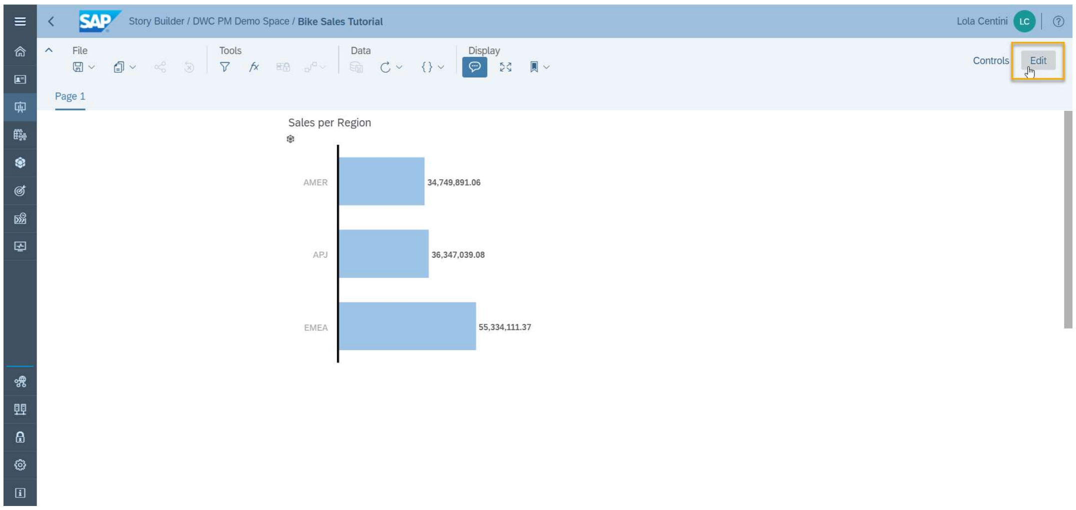


- Click in the output and check in the right panel, whether all measures are set to measures. You can already preview your data. When everything is correct, save and deploy.

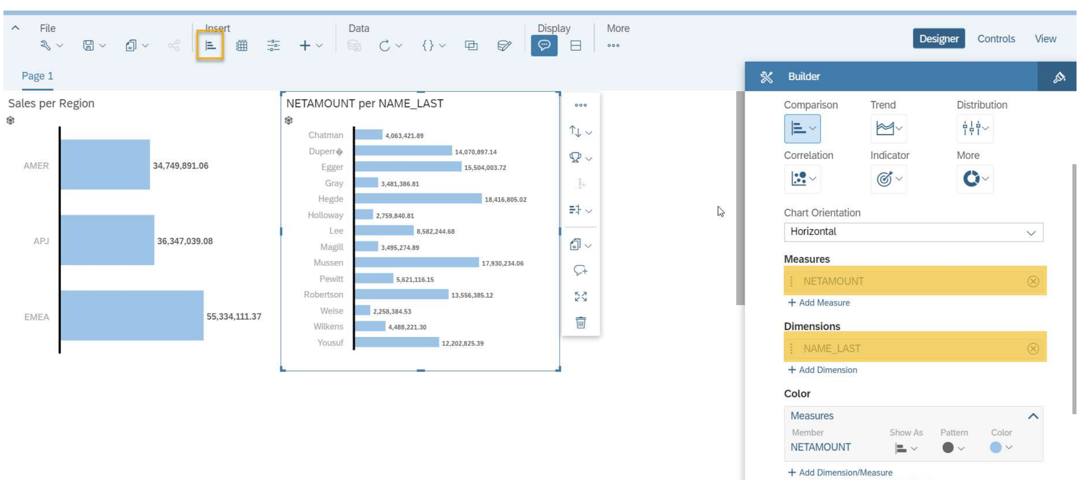
- To continue with the visualization, navigate to the 

and choose your space. Search for the “Bike Sales Tutorial” you created earlier.

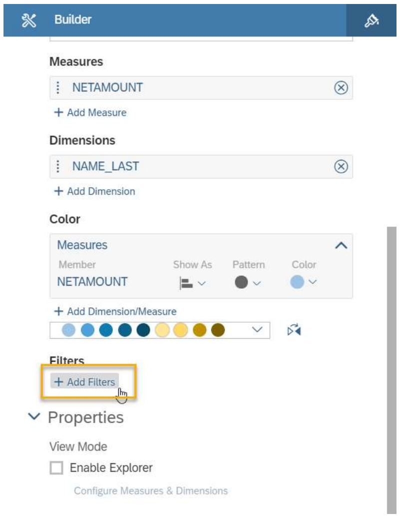
- To start enable editing in the upper right corner.

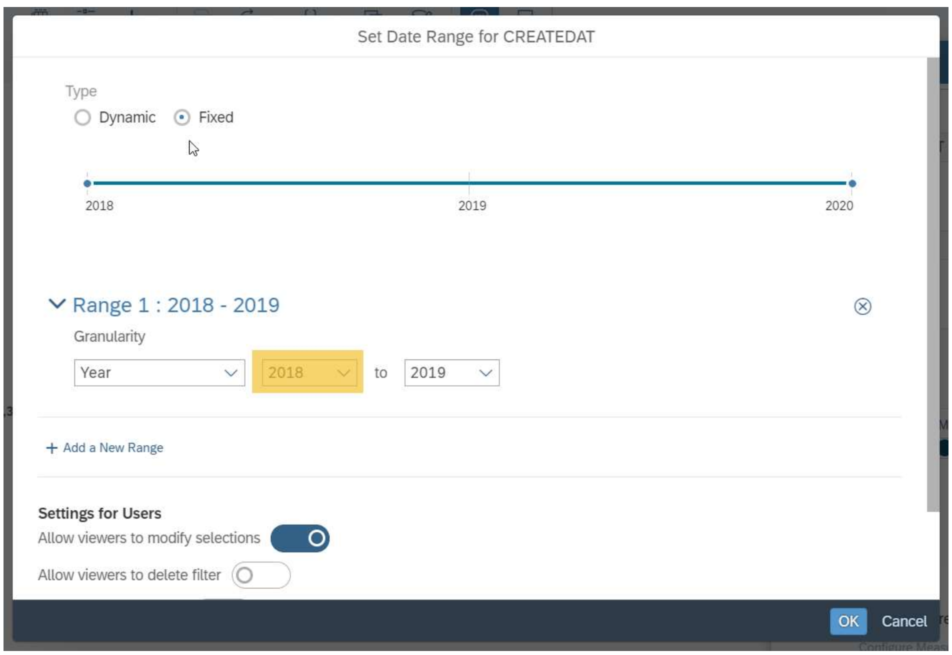


- Insert a new chart and assign a measure and a dimension. Choose NETAMOUNT as Measure and NAME_LAST as a Dimension.



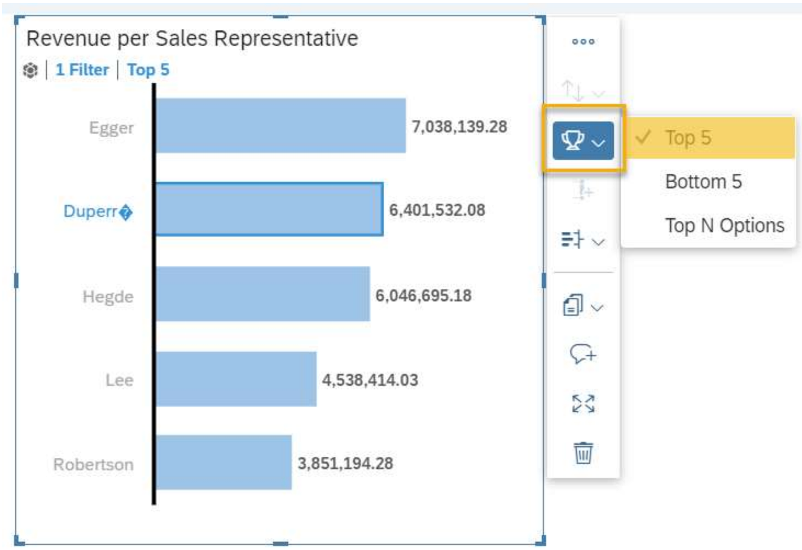
- Only the newer numbers are relevant for choosing the best sales representative. Filter for this year only. Choose the CREATEDAT (Range) for the filter. In the popup change the date range to 2019- 2019 to show this year’s sales only.



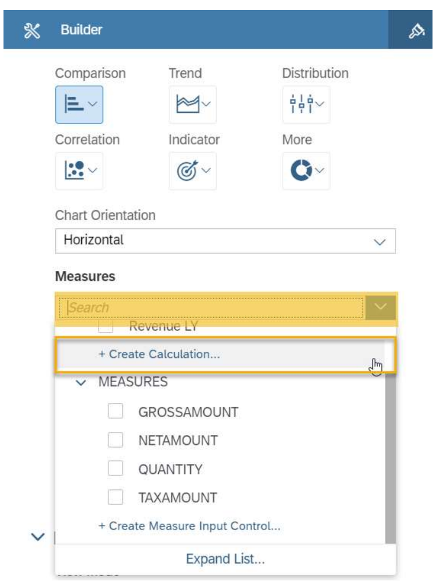


- To have a more speaking name change the title of the chart to “Revenue per Sales Representative”

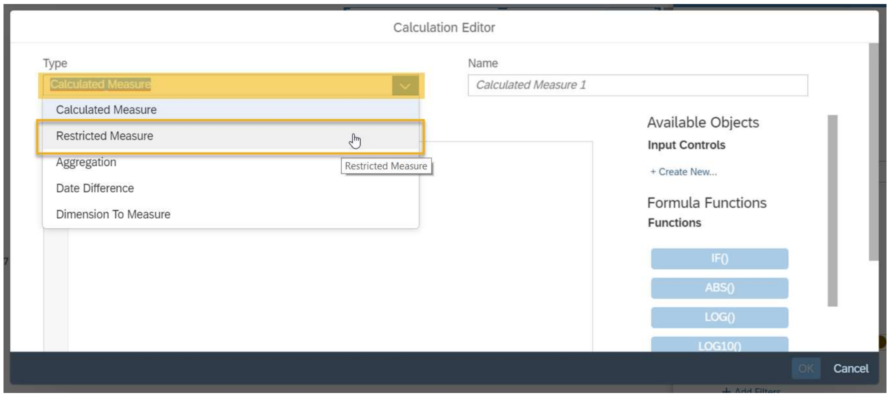
- Possible options are to show only a selection for example the top 5 representatives. In the menu next to the chart choose the option “Top 5”.



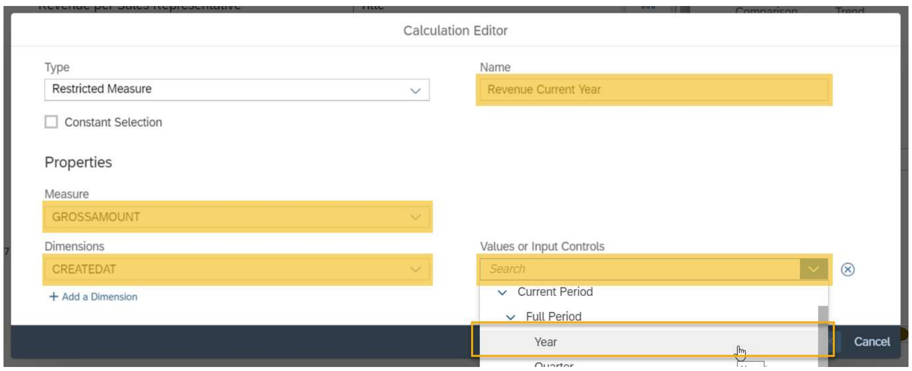
- To finish the requested reports the Year-over-Year Sales Comparison is needed. All the models are already prepared. In the SAC the numbers only need to be combined. To do so insert a new chart and then create a new calculated measure in the builder panel at the right.



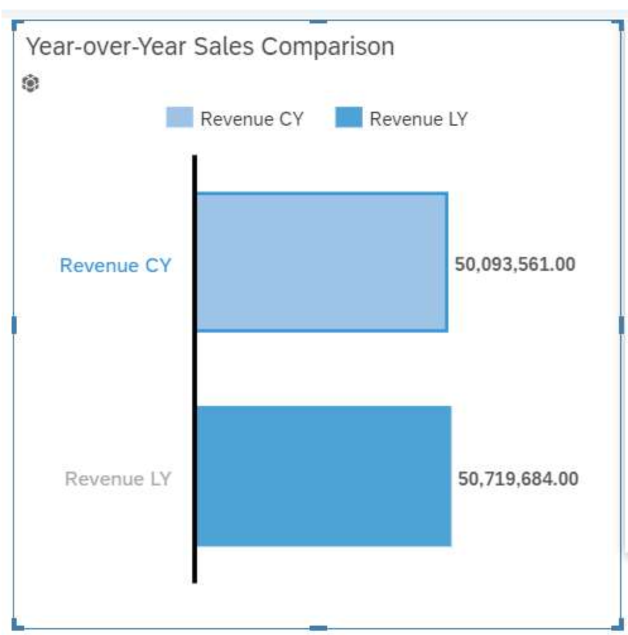
- In the following popup first change the type to a “Restricted Measure”.



- In the editor now change the name to “Revenue Current Year”. Set the measure to GROSSAMOUNT and the dimension to CREATEDAT. Then choose the value for the creation date. For the current year go to “Current Period”, “Full Period” and choose “Year”.



- For comparison to the last year repeat the steps and name the measure “Revenue Last Year”. This time choose the “Previous Period” as the value. Now your report for the Year-over-Year Sales Comparison should look like this:



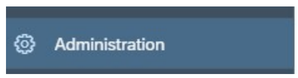
APPENDIX

Data Provisioning Agent Setup

As a prerequisite to connect to your local HANA and other sources, the Data Provisioning Agent needs to be installed and configured. Follow the installation guide in the SAP Help Portal to install the Data Provisioning Agent. Install the latest version of HANA DP Agent 2.0 (as of Aug 6: SPS04)

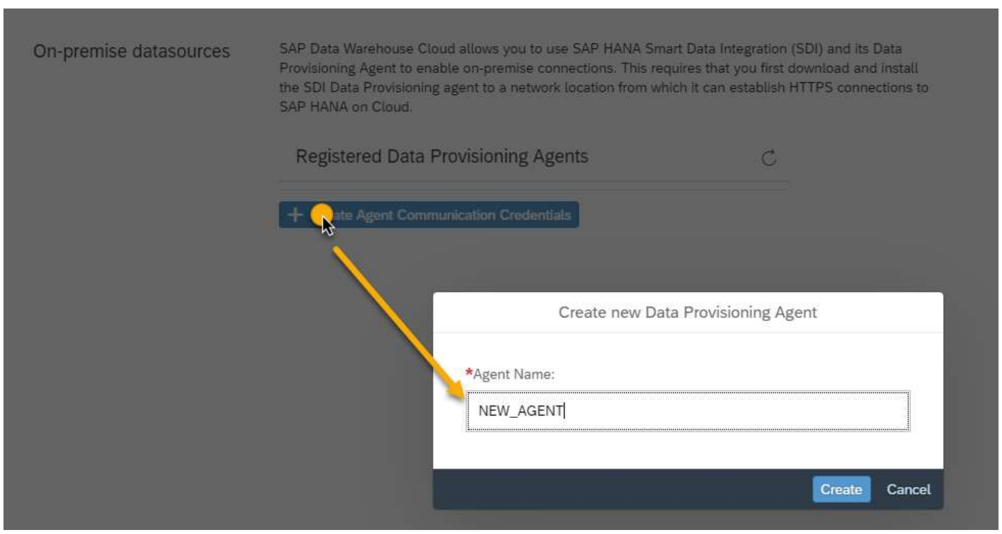
After you have successfully installed the Data Provisioning Agent, follow these steps to finalize the configuration with SAP Data Warehouse Cloud:

1.) Check if the user that will configure the Data Provisioning Agent in SAP Data Warehouse Cloud has the role "Data Warehouse Cloud Administrator"

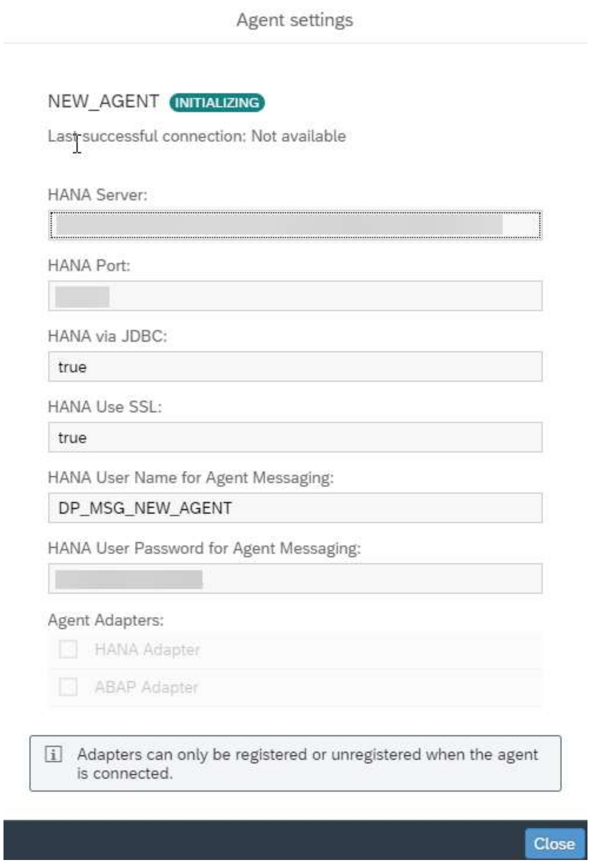
2.) Navigate to "Administration" 

in the navigation bar.

3.) Register a new Data Provisioning Agent: click "Create Agent Communication Credentials" and enter the new Agent Name:



Click “Create”. As a result, HANA Server Information, user and password to register the Data Provisioning Agent with SAP Data Warehouse Cloud will be shown.
The numbering on the following screenshot will help you with the configuration in the next step as they map to a setting in the DP Agent configuration.



Don’t close this window yet, as you need this information to configure the agent in the next step!

4.) Navigate to the Data Provisioning Agent installation <DP Agent root> and configure the agent as follows:
Edit the file *dpagentconfig.ini* and check / change the following values according to the values from 3.) above (Agent settings)

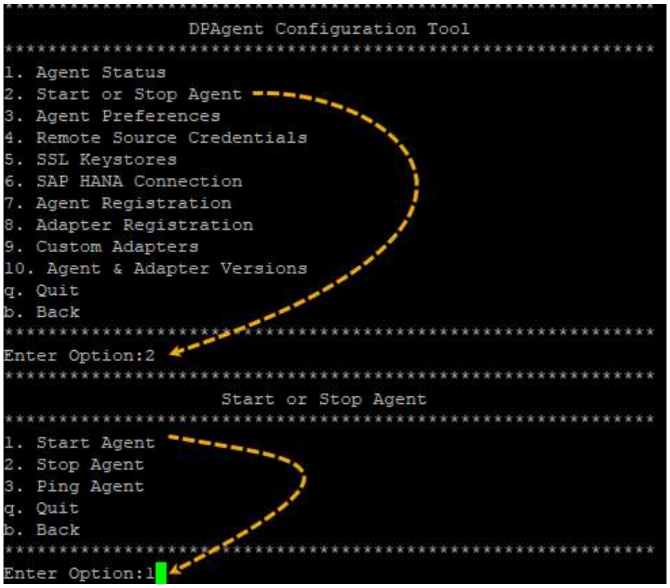
1. agent.name=<DP Agent Name> (case sensitive!)
2. hana.port=<HANA Port>
3. hana.onCloud=false
4. hana.server=<HANA Server>
5. hana.useSSL=true
6. jdbc.enabled=true
7. jdbc.host=<HANA Server>
8. jdbc.port=< HANA Port>
9. jdbc.encrypt=true

5.) Start the Data Provisioning Agent once:

Navigate to <DP Agent root>/bin and execute the following

Linux: ./agentcli.sh --configAgent

Windows: agentcli.bat --configAgent



Wait a few seconds and check if the agent is running using the “Ping Agent” option (3).

Exit the DP Agent Configuration tool: “q. Quit”.

6.) Configure HANA XS Username and Password. Navigate to <DP Agent root>/bin and execute the following

Linux: ./agentcli.sh --setSecureProperty

Windows agentcli.bat --setSecureProperty

Set the user and password from the registration dialog in 3.) using the Option 1 and 2 in the following screen with these values:

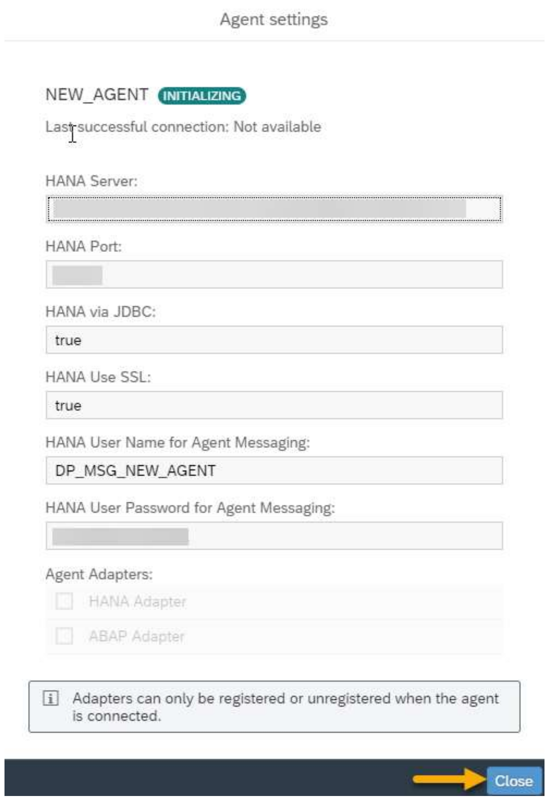
HANA XS Username = HANA User for Agent Messaging

HANA XS Password = HANA User Password for Agent Messaging

Exit the DP Agent Configuration tool: “q. Quit”.

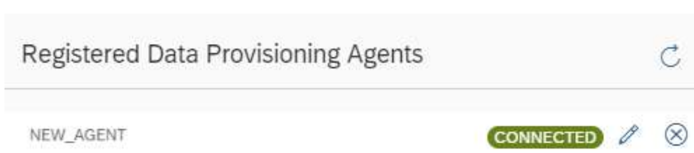
7.) Stop and restart the Data Provisioning agent – see step 5.)

8.) Now you can close the Data Provisioning Agent setting window:

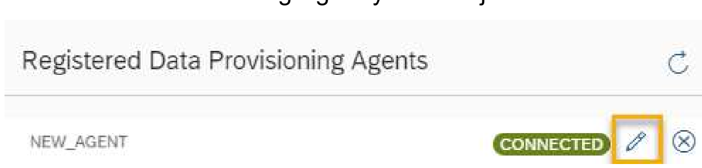


9.) As a result, the Data Provisioning Agent is now connected

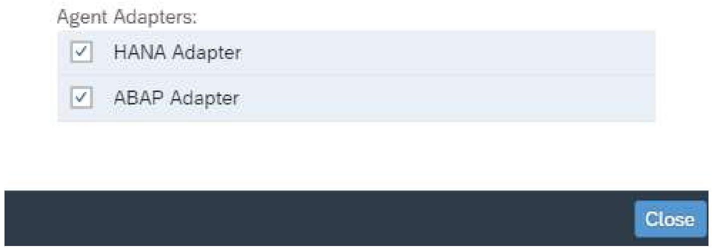
(you might need to refresh the view in case the status is not yet changed: ):



10.) Now enable the Adapters "HANA Adapter" and "ABAP ADAPTER":
Edit the Data Provisioning Agent you have just created:



Select the adapters and close the dialogue afterwards:

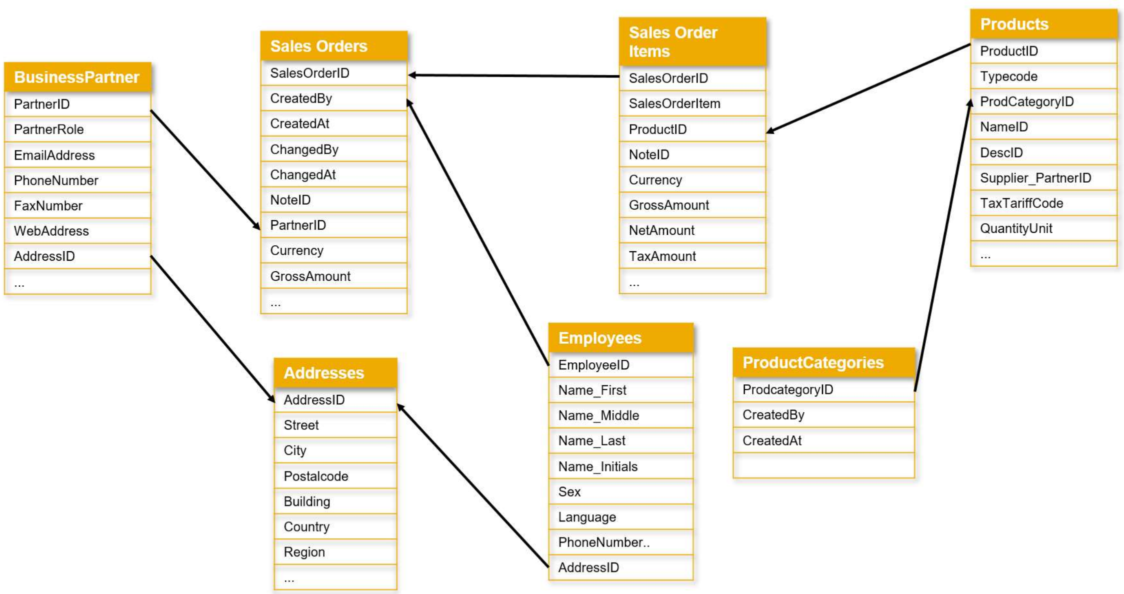


You have successfully setup the Data Provisioning Agent. You may now create connections to remote SAP HANA and SAP ABAP systems.

Technical Details of the Bikes samples data model

Demo Content: Table Description

The demo content mentioned in this document is based on a Sales Order scenario. The model is developed in a way that it covers the basic scenarios as well as a few advanced scenarios.



SalesOrders

This dataset contains consolidated order details with Gross Amount, Net Amount, and Tax Amount aggregated for each order, i.e. one row per order. It also contains Partner information, creating employee, Sales Organization for region details (for e.g. EMEA, APJ, etc.) and most importantly, the three status flags for an order – BillingStatus, DeliveryStatus, LifecycleStatus. These status flags have three possible values – I (In Progress), C (Completed) and X (Canceled).

Foreign keys in this dataset – CreatedBy and ChangedBy (EmployeeID from Employees), PartnerID (PartnerID from BusinessPartner)

SalesOrderItem

This dataset contains the breakdown of each order from SalesOrders, therefore, redundant data for SalesOrderID is included. This is the right place to find the products that have been ordered, quantity of the products, and pricing details for each product. The DeliveryDate column in this dataset impacts the DeliveryStatus flag in SalesOrders. In case of a canceled order, the date in this column is set to “99991231”.

Foreign keys in this dataset – ProductID (ProductID from Products)

Following table displays detailed information about all the tables:

Table Name		Semantics	Modeling Information
Addresses	address data	Address data of employees and business partners	Dimension
BusinessPartners	business partner data	customers who place orders ordering bikes	Dimension
Employees	employee data	employees create and change sales orders. They belong to a specific sales organization	Dimension
ProductCategories	Product Categories (+ extra text table)	All products belong to a certain product category	Dimension
Products	products (+ extra text table)	Products are being sold by the company. Table contains product specs like price, weight, product category	Dimension
SalesOrders	sales order header	Sales orders header part containing e.g. status information, business partner, when it was created/changed etc.	Fact data
SalesOrderItems	sales order items	Sales order item information on which products have been sold, quantity and amounts	Fact data