



Public

# **SAP Data Warehouse Cloud**

## **Beta Content Tutorial**

Version 1.0 – August 15, 2019

# TABLE OF CONTENTS

INTRODUCTION .....	3
STORY LINE .....	3
SPACES .....	5
PREPARE THE DATA MODEL .....	6
1. SAP HANA: Load the schema and tables including the sample data in your SAP HANA DB.....	6
2. Upload CSV files to SAP Data Warehouse Cloud .....	7
USE THE BEST RUN BIKES SALES DEMO CONTENT TO BUILD THE ANALYTICS .....	12
Use sample data .....	12
Connect to SAP HANA .....	12
Use CSV Files.....	15
Data Modeling – Basics.....	15
Data Modeling – enhanced Model .....	22
Master Data .....	22
Enhance Fact View .....	26
APPENDIX .....	34
Data Provisioning Agent Setup .....	34
Technical Details of the Bikes samples data model .....	40
Demo Content: Table Description.....	40
CSV Files: Data Type mapping.....	41

## 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](https://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

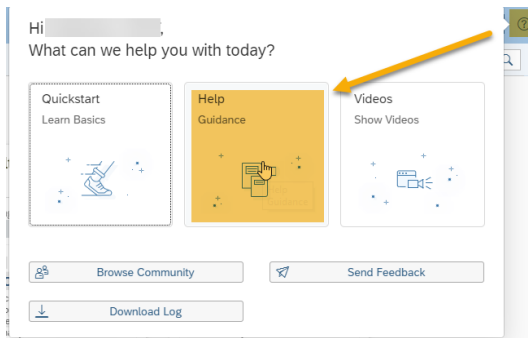
## SPACES

To work with SAP Data Warehouse Cloud, a Space is needed as a pre-requisite. Find more information on of Spaces [here](#).

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

To create a new space, navigate to  Space Management 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 a 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 CSV: Upload CSV files into SAP Data Warehouse Cloud to create local tables.

Please find the SAP HANA export and the CSVs 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`

Import Catalog Objects

Import objects from:

Local archive (\*.tar.gz):

<input checked="" type="checkbox"/>	Catalog Object	
<input checked="" type="checkbox"/>	DWC_DEMO	<input type="button" value="edit"/>
<input checked="" type="checkbox"/>	BusinessPartners	
<input checked="" type="checkbox"/>	Addresses	
<input checked="" type="checkbox"/>	Employees	
<input checked="" type="checkbox"/>	ProductCategories	
<input checked="" type="checkbox"/>	ProductCategoryTexts	
<input checked="" type="checkbox"/>	ProductTexts	
<input checked="" type="checkbox"/>	Products	
<input checked="" type="checkbox"/>	SalesOrderItems	

☒ Include dependencies ☒ Include table data ☐ Replace existing objects

Number of parallel threads:

- Click on Import

- Once imported, the tables will appear in the schema *DWC\_Demo*

	SYSTEM,DWC_Demo_...	
	Search Tables	
	DWC_Demo_Schema	
	Addresses	
	DWC_Demo_Schema	
	AddressesZwo	
	DWC_Demo_Schema	
	BusinessPartner	
	DWC_Demo_Schema	
	Employees	
	DWC_Demo_Schema	
	Products	
	DWC_Demo_Schema	
	SalesOrderItems	
	DWC_Demo_Schema	
	SalesOrders	


## 2. Upload CSV files to SAP Data Warehouse Cloud

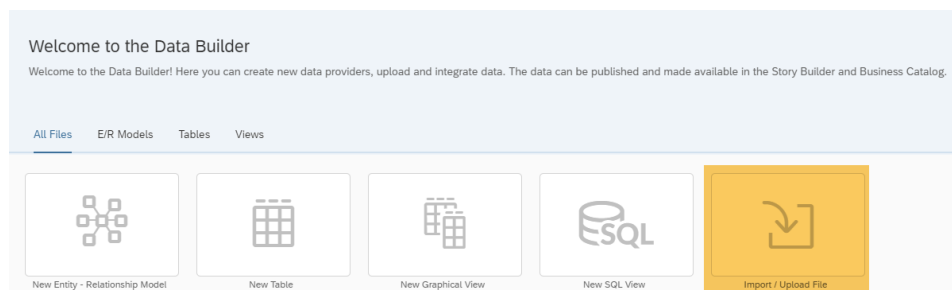
In this chapter you will learn how to upload the CSVs to a local table in SAP Data Warehouse Cloud to prepare the sample data model before you can start the modeling experience.

You can use this option as an alternative to using the SAP HANA remote connection described in the previous chapter. Or explore this option as an additional learning experience for the SAP Data Warehouse Cloud.

As shared in the introduction, a Space is a prerequisite to upload CSV files into a local table.

Follow these steps to upload the CSVs:

- Navigate to the  **Data Builder**
- 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 “Import / Upload File”



- In the upcoming pop-up click browse to select the CSV you want to upload

Upload File

File (max 20mb):

Browse

Upload your file to the server

Import Cancel

- The selected file will be uploaded, and a preview of the data is shown

Upload File

File (max 20mb):

Products.csv

Browse

Name:

Products\_table

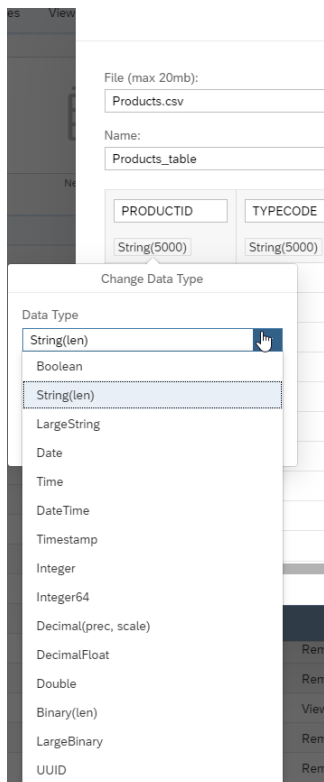
PRODUCTID	TYPECODE	PRODCATEGO...	CREATEDBY	CREATEDAT	CHANGEDBY	CHANGEDAT
String(5000)	String(5000)	String(5000)	String(5000)	String(5000)	String(5000)	String(5000)
RO-1001	PR	RO	0000000009	20181003	0000000009	20181003
RO-1002	PR	RO	0000000009	20181003	0000000009	20181003
RO-1003	PR	RO	0000000012	20181003	0000000012	20181003
BX-1011	PR	BX	0000000009	20181003	0000000009	20181003
BX-1012	PR	BX	0000000006	20181003	0000000006	20181003
BX-1013	PR	BX	0000000007	20181003	0000000007	20181003
BX-1014	PR	BX	0000000011	20181003	0000000011	20181003
BX-1015	PR	BX	0000000008	20181003	0000000008	20181003
BX-1016	PR	BX	0000000011	20181003	0000000011	20181003
CC-1021	PR	CC	0000000010	20181003	0000000010	20181003

Import Cancel

In this example we have upload the file "Products.csv"

- Before importing the file, we may now change the Name and the data types of the columns. The default data type currently is *String(5000)*.  
To change the data types just click on it for a column and chose an appropriate type:





Navigate outside of this window or press ESC to finalize this step and continue with the next column.

- In the Appendix you find an overview of all data types for each object. Please use this information to create the table with the same data types as being used from SAP HANA.

It is most important to adapt the non-string fields. This is how the mapping looks for the “Products\_table”

Upload File

File (max 20mb):  
 Browse

Name:

PRODUCTID	TYPECODE	PRODCATEGO...	CREATEDBY	CREATEDAT	CHANGEDBY	CHANGEDAT
String(10)	String(2)	String(2)	String(10)	Date	String(10)	Date
RO-1001	PR	RO	0000000009	20181003	0000000009	20181003
RO-1002	PR	RO	0000000009	20181003	0000000009	20181003
RO-1003	PR	RO	0000000012	20181003	0000000012	20181003
BX-1011	PR	BX	0000000009	20181003	0000000009	20181003
BX-1012	PR	BX	0000000006	20181003	0000000006	20181003
BX-1013	PR	BX	0000000007	20181003	0000000007	20181003
BX-1014	PR	BX	0000000011	20181003	0000000011	20181003
BX-1015	PR	BX	0000000008	20181003	0000000008	20181003
BX-1016	PR	BX	0000000011	20181003	0000000011	20181003
CC-1021	PR	CC	0000000010	20181003	0000000010	20181003

Import Cancel

Remark: Currently you need to change the data type during import here as they cannot be changed in the table maintenance with data loaded.

- Click Import. A progress indicator is shown while the upload is being executed:

Upload File

File (max 20mb):

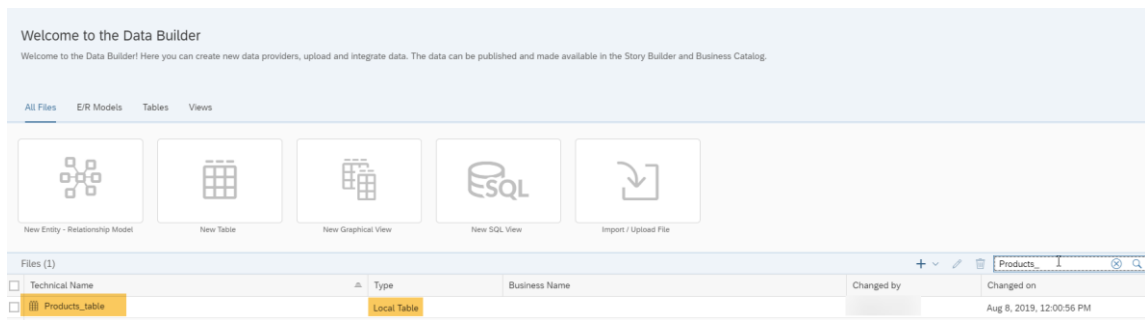
Name:

PRODUCTID	TYPECODE	PRODCATEGO...	CREATEDBY	CREATEDAT	CHANGEDBY	CHANGEDAT
String(10)	String(2)	String(2)	String(10)	Date	String(10)	Date
RO-1001	PR	RO			0000000009	20181003
RO-1002	PR	RO			0000000009	20181003
RO-1003	PR	RO			0000000012	20181003
BX-1011	PR	BX			0000000009	20181003
BX-1012	PR	BX			0000000006	20181003
BX-1013	PR	BX	0000000007	20181003	0000000007	20181003
BX-1014	PR	BX	0000000011	20181003	0000000011	20181003
BX-1015	PR	BX	0000000008	20181003	0000000008	20181003
BX-1016	PR	BX	0000000011	20181003	0000000011	20181003
CC-1021	PR	CC	0000000010	20181003	0000000010	20181003

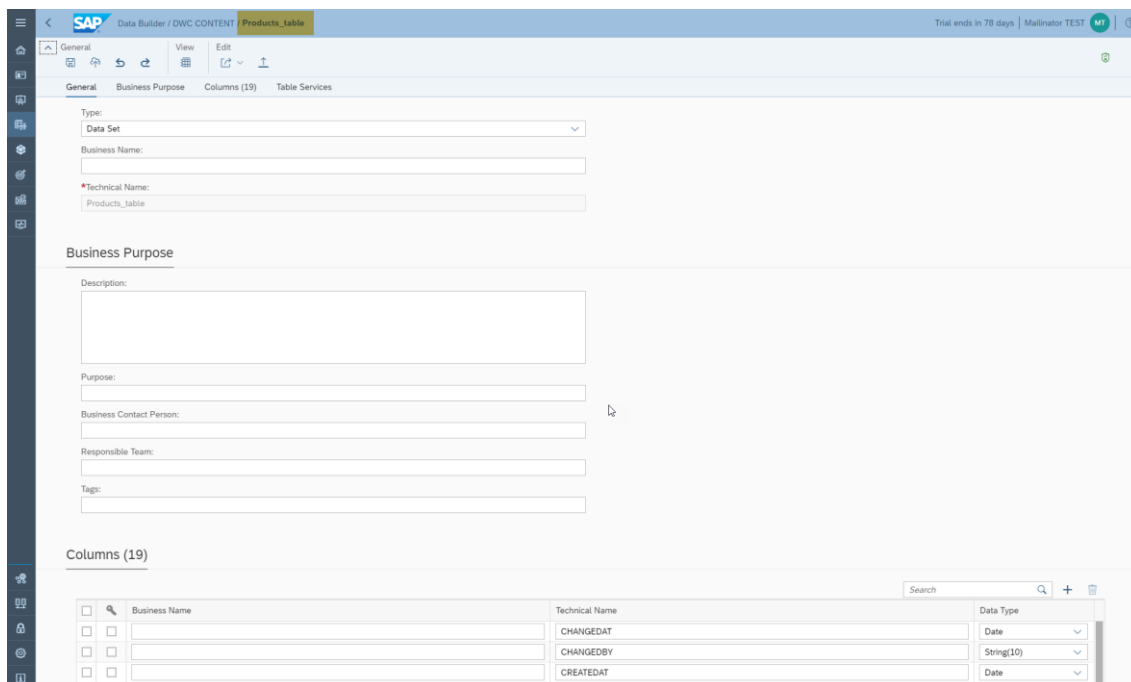
Uploading

Processing step 2 of 4  
Deploying table Products\_table to database

- Once the upload finished successfully, you will see the newly created table in the *Files* on the Data Builder screen with the type “Local Table”:

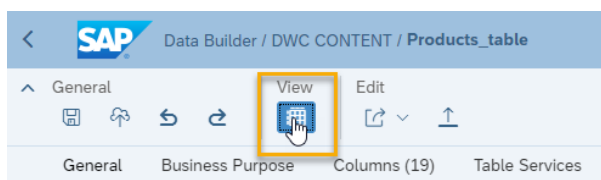


- Click on the table name to explore the table details:



Change this information where needed, e.g. add a *Business Name* of the table or add a *Business Name* to each column of the table, to have more speaking name.

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



- Similarly, upload all the provided CSV files into the system.

**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

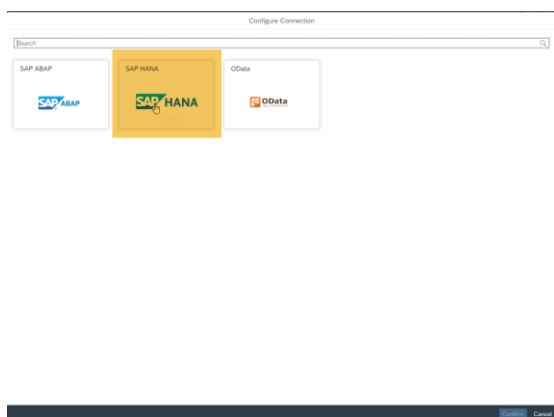
### Use sample data

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.

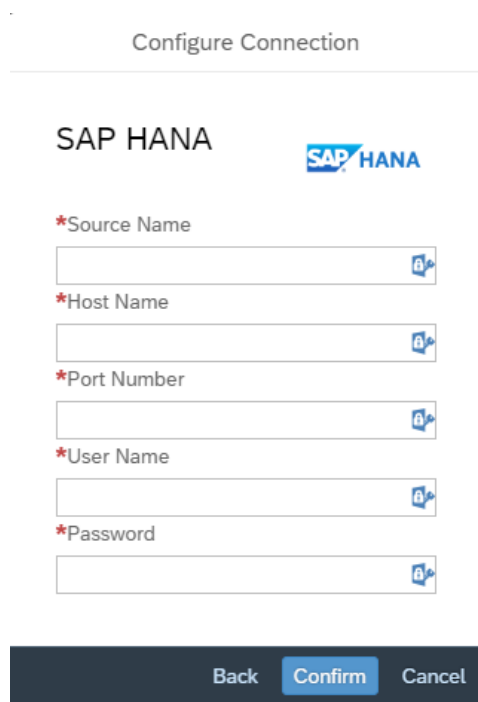
### Connect to SAP HANA

- 1.) Create a connection to your SAP HANA database

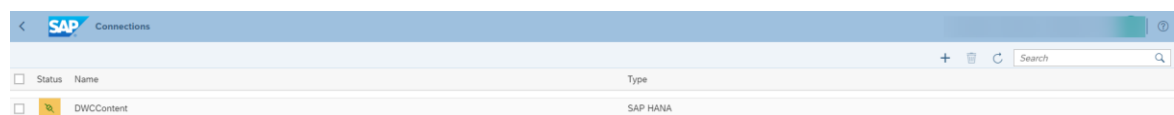
Navigate to  **Connections** and create a new connection to SAP HANA:




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:

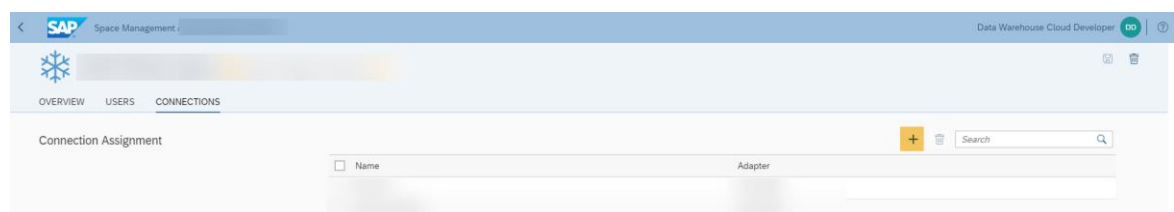


After the connection is created you can check if it is working by clicking the icon in the “Status” column next to your connection:



Status	Name	Type
	DWContent	SAP HANA

2.) Assign the connection to a space in

Name	Adapter

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

Select Connections

dwc\_content

Selected: 1

☒ DWC\_CONTENT



You have successfully assigned your connection:

Connection Assignment		+ Search	
<input type="checkbox"/> Name		Adapter	
<input type="checkbox"/> DWC_CONTENT		SAP HANA	

## 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.

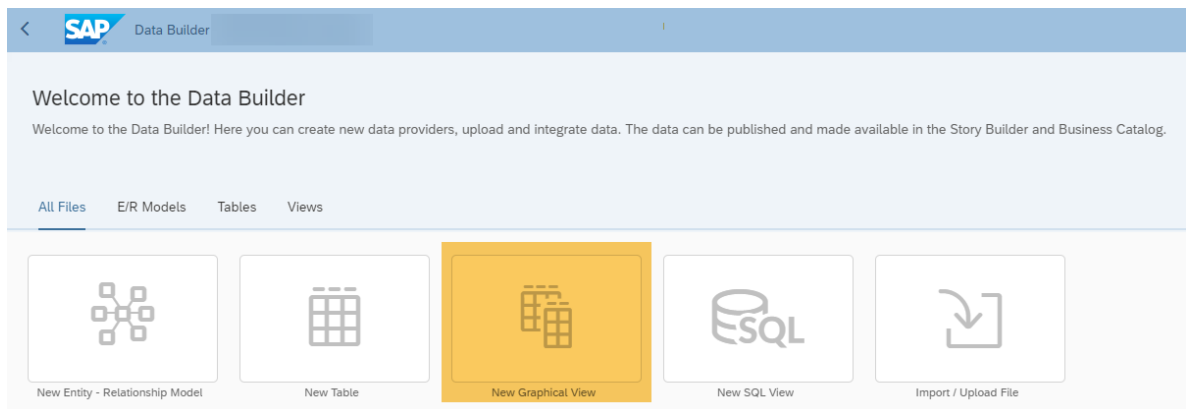
SAP HANA will be used as a source for the walkthrough and screenshots. It will be always mentioned how to apply a step using CSV files.

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.

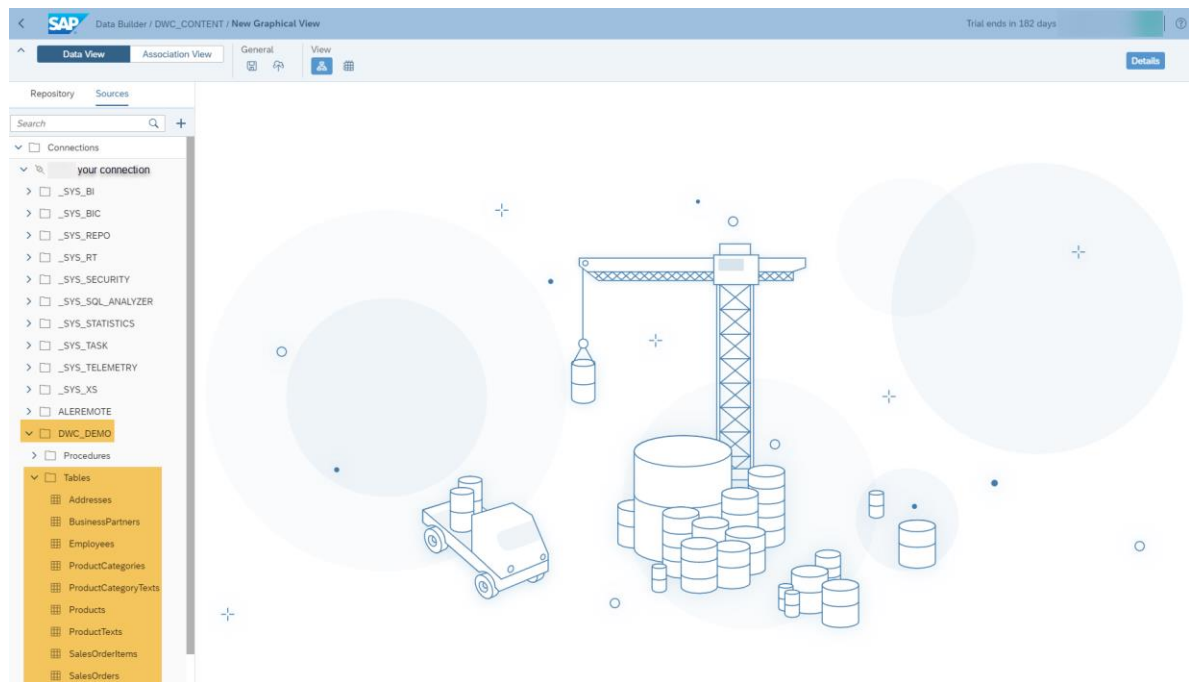
Subsequently we will enhance the model to create the reports laid out in the “Story Line” chapter at the beginning of this document in the chapter “Data Modeling – Full Modell” following this one.

Let’s start:

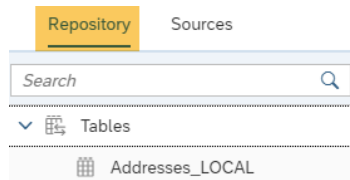
- 1.) To build a data model, navigate to the  **Data Builder**
- 2.) If this is your first visit to the Data Builder you need to select a Space. Select the Space you just have assigned the connection to – for this guide “DWC\_CONTENT” is used.
- 3.) In the entry screen different modeling options are 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:

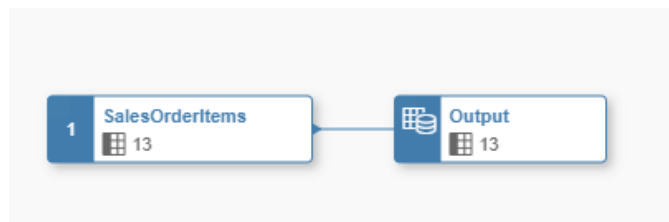


**CSV files:** In the left tab, navigate to *Repository* and select the tables you have created while uploading the CSV files:

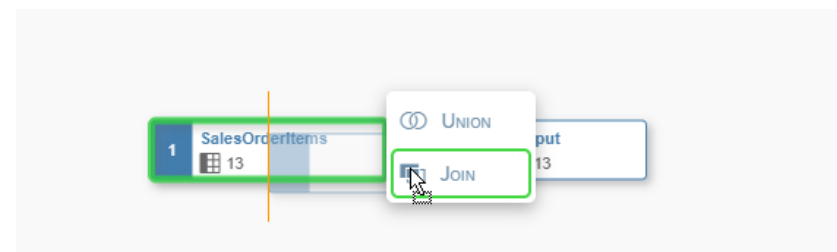


Remark: if you already have used an object, you can directly access it from the *Repository*.

- 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:

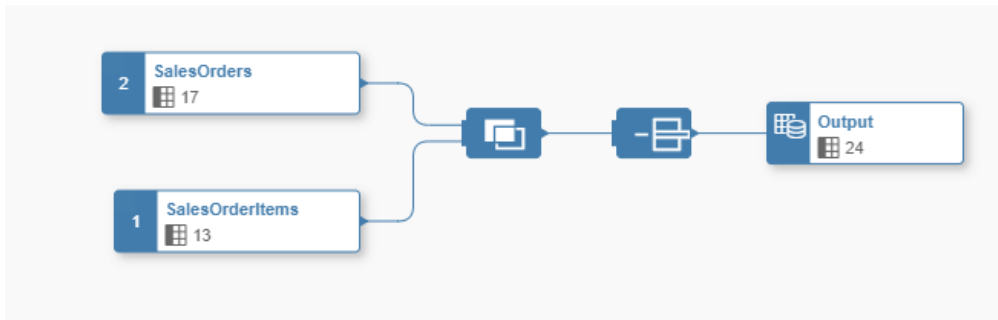


- 6.) Add the *SalesOrders* by dragging and dropping it on the *SalesOrderItems* in the canvas. An additional option to choose the join criteria appears, choose *Join*:





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:

Remove Columns Properties

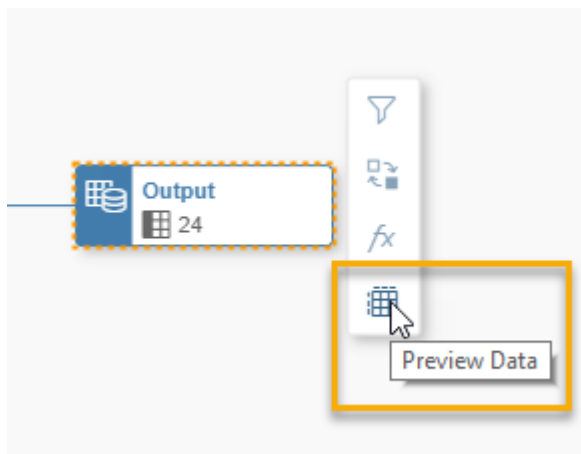
Remove Columns 1

Search Columns

Column Name	Keep
AA SALESORDERID	<input checked="" type="checkbox"/> <input type="radio"/>
AA SALESORDERITEM	<input checked="" type="checkbox"/> <input type="radio"/>
AA CREATEDBY	<input type="checkbox"/> <input type="radio"/>
AA PRODUCTID	<input type="checkbox"/> <input type="radio"/>
AA CREATEDAT	<input type="checkbox"/> <input type="radio"/>
AA NOTEID	<input type="checkbox"/> <input type="radio"/>
AA CHANGEDBY	<input type="checkbox"/> <input type="radio"/>
AA CURRENCY	<input type="checkbox"/> <input type="radio"/>
AA CHANGEDAT	<input type="checkbox"/> <input type="radio"/>
1-23 GROSSAMOUNT	<input type="checkbox"/> <input type="radio"/>

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:

SALESORDERID	SALESORDERITEM	PRODUCTID	NOTEID	CURRENCY	GROSSAMOUNT	NETAMOUNT
500000000	10	MB-1034		USD	2499.00	2186.63
500000000	20	CB-1161		USD	399.00	349.13
500000001	10	HB-1175		USD	899.00	786.63
500000001	20	RC-1056		USD	2499.00	2186.63
500000001	30	CC-1021		USD	1144.00	1001.00
500000001	40	CB-1161		USD	399.00	349.13
500000002	10	HB-1175		USD	899.00	786.63

- 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*

**Properties**

**Output** 13

Business Name:

Technical Name:

Output

Type:

Fact

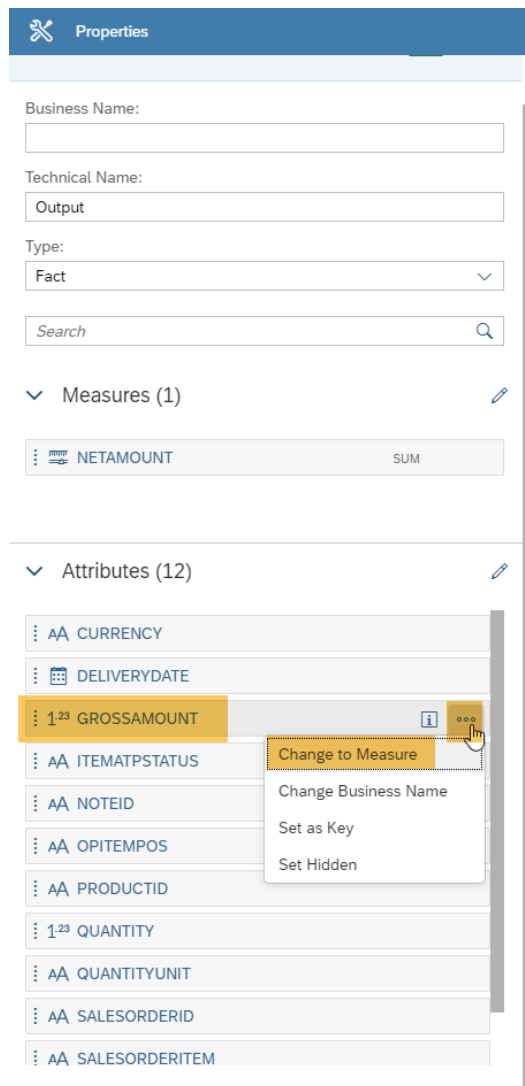
Search

Measures (0)

Drop attribute here to change it to a measure

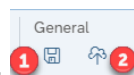
- 8.) After having changed the type to *Fact*, you need to move the columns that are measures from the *Attributes* to the *Measures*. You can either drag & drop a column or you use the context menu in the right area of a column (indicated by the three dots, once the mouse cursor hovers this area) and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

Make GROSSAMOUNT, NETAMOUNT, TAXAMOUNT and QUANTITY a measure.



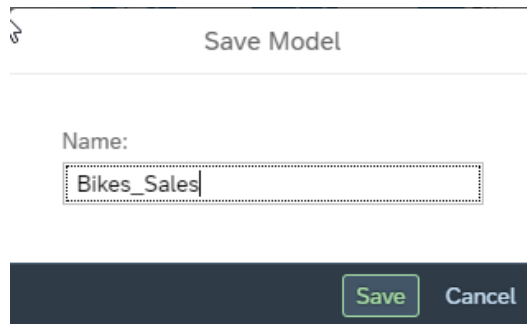
- 9.) Now let's save and the deploy the model. Either click the save button (1) first and then deploy (2), or

directly save and deploy clicking the deploy button (2):



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

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



A dialog box titled "Save Model" with a "Name:" label and a text input field containing "Bikes\_Sales". At the bottom are "Save" and "Cancel" buttons.

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

Model saved successfully.

After a successful deployment the following message will be shown

Deployment successful.

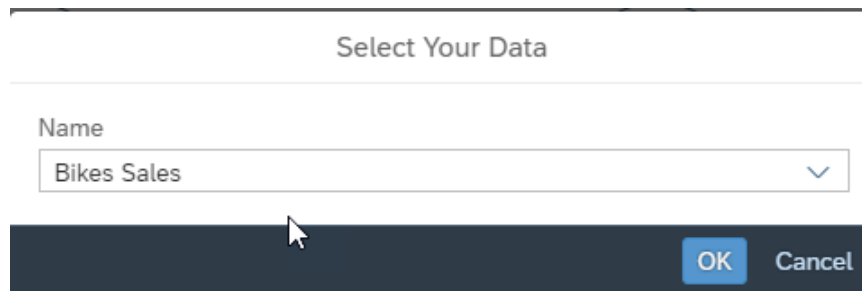
10.) Let's create the first visualization: navigate to the



11.) Create a new story:

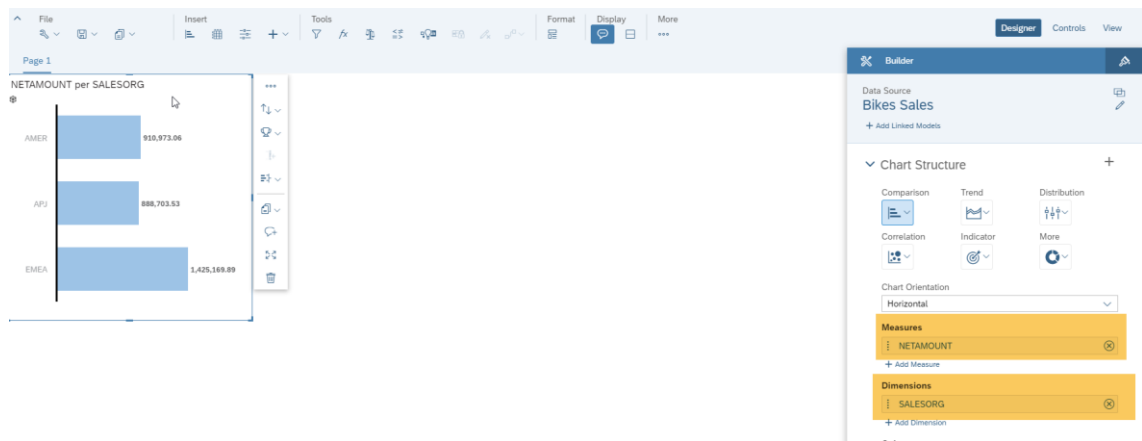


12.) Choose the Bikes Sales data model that you have just created and click OK:



A dialog box titled "Select Your Data" with a "Name" label and a dropdown menu showing "Bikes Sales". At the bottom are "OK" and "Cancel" buttons.

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:

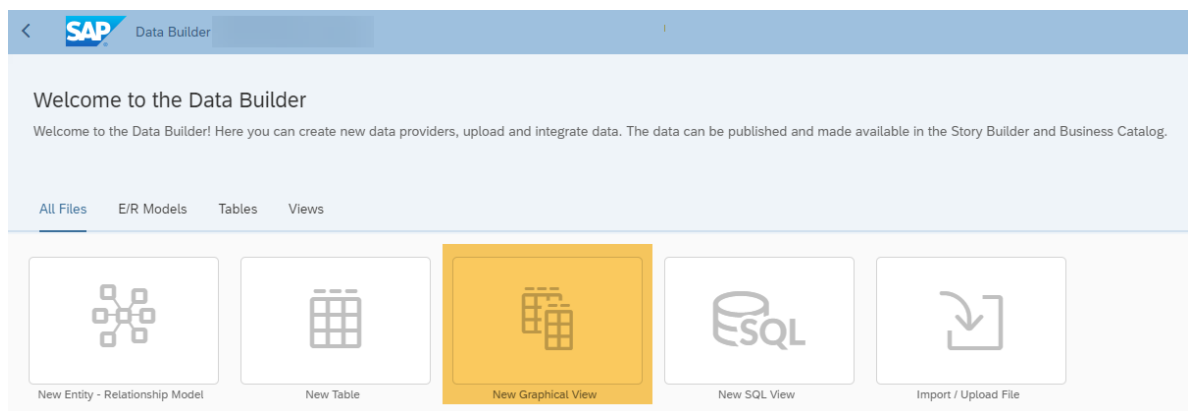
- master data objects for products, business partners, employees,
- add those master data objects to our Bikes Sales model
- rename measures and master data dimension that are needed for our reports to speaking names for the reports
- use further functionalities like calculated columns

### Master Data

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



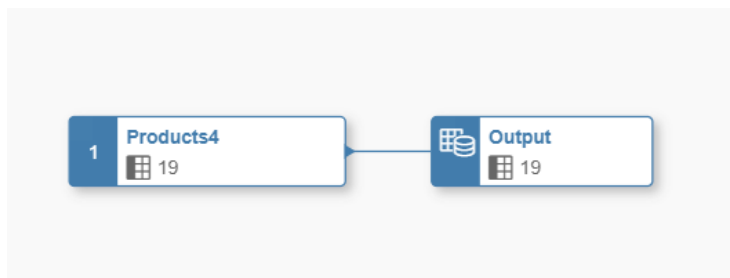
- Let's create a new graphical view.



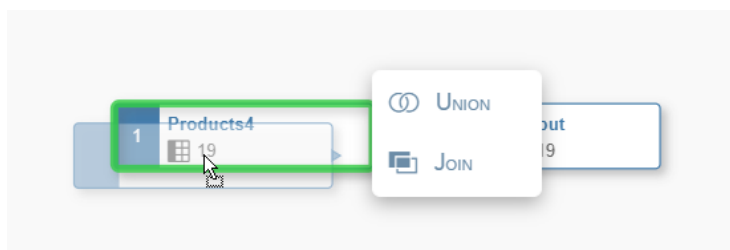
- Find your tables in the left panel. To access the data models from the remote HANA connection search in “Connections”, for CSV-Files search in “Repository”



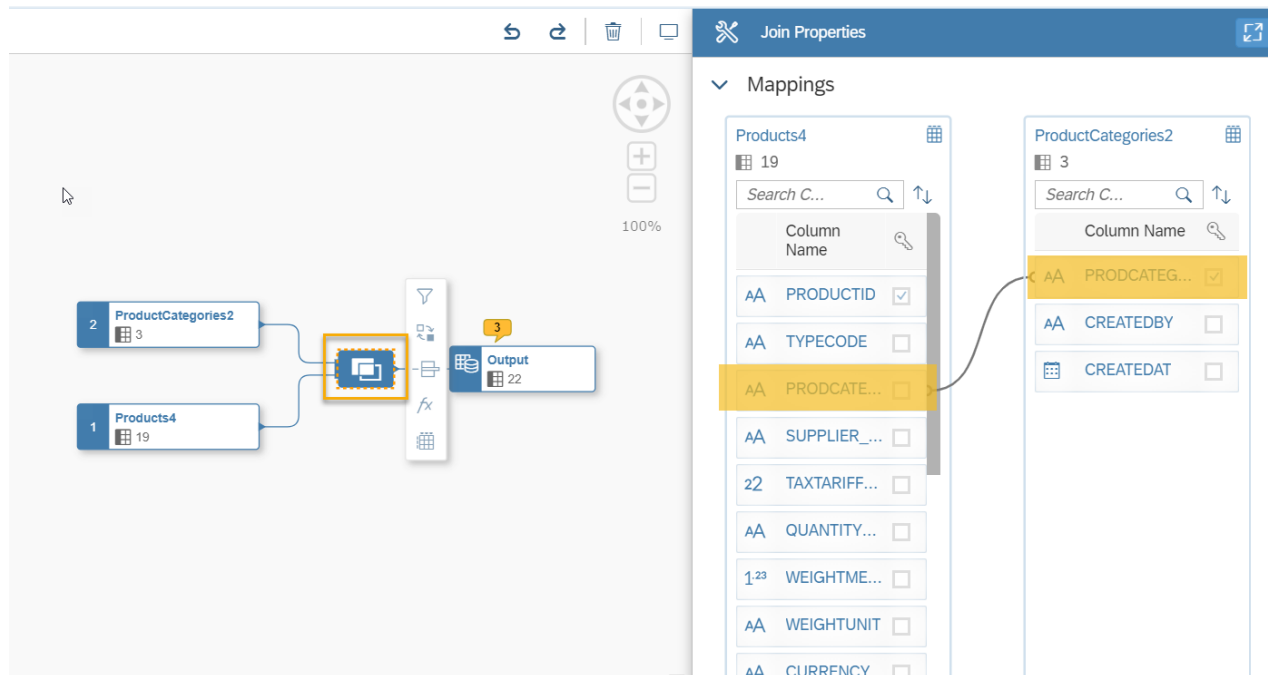
- Let's start with the products master data model. Therefore, you need the tables *Products*, *ProductCategories* and the corresponding *ProductTexts* and *ProductCategoryTexts*. You can find further information about the associations and attributes in the Appendix.
- To start drag and drop the *Products* table into the modeling area.



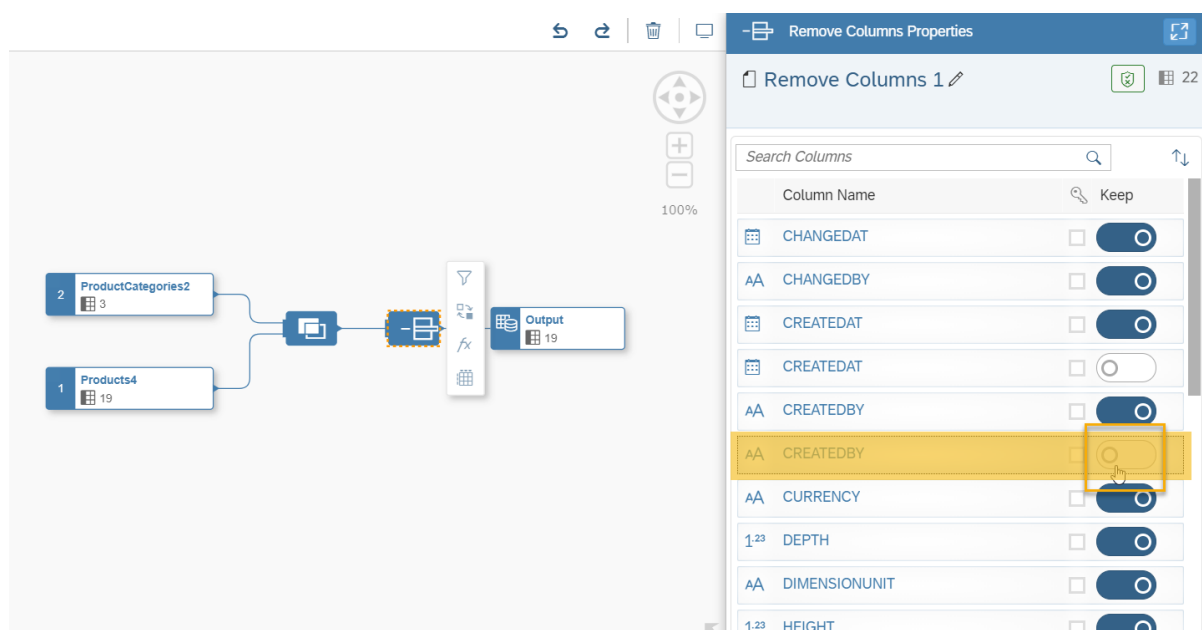
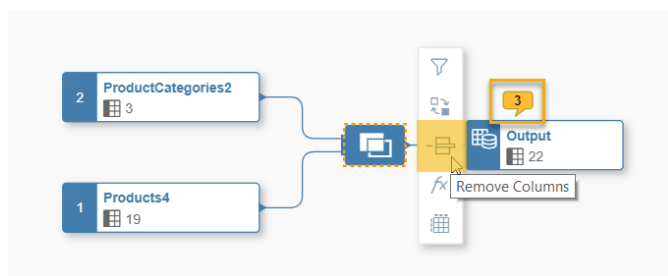
- To perform a join, drag the table *ProductCategories* onto the *Products* in the canvas.



- 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 doubled columns





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

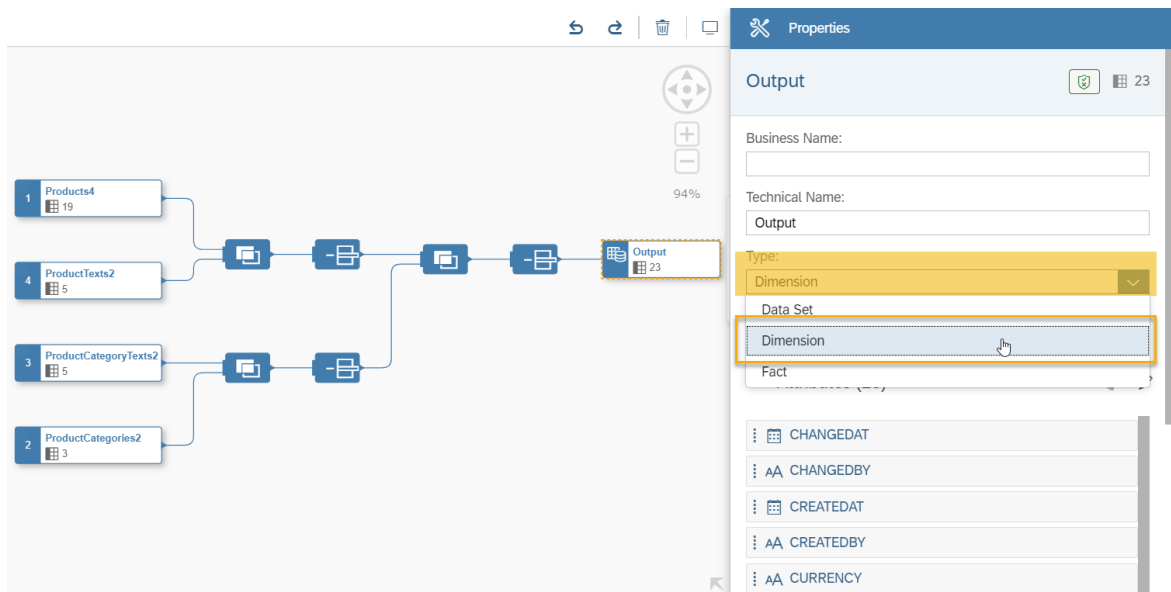
The screenshot shows a data flow diagram with four input tables: ProductCategories2 (2 rows), ProductCategoryTexts2 (4 rows), ProductTexts2 (3 rows), and Products4 (1 row). These are joined sequentially to an 'Output' table (23 rows). A 'Preview Data' button is highlighted. Below the diagram is a table titled 'Data Preview: Output (42)'.

PRODUCTID	TYPECODE	PRODCATEGORYID	SUPPLIER_PARTNE...	TAXTARIFFC
RO-1001	PR	RO	0100000000	1
RO-1002	PR	RO	0100000001	1
RO-1003	PR	RO	0100000002	1

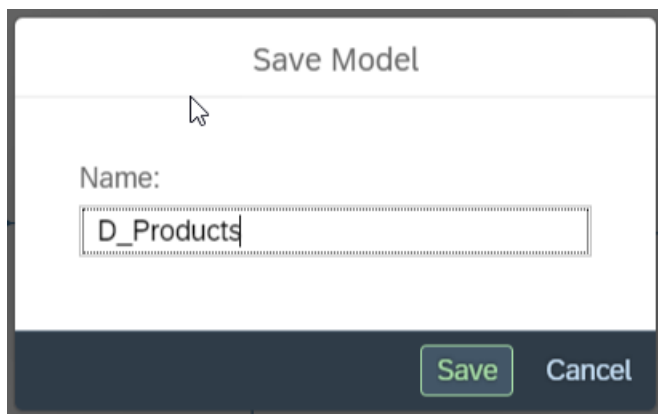
- To enable more speaking names, you can change the business names. Go to the “Properties” panel on the right and change the business name of the model as well as the properties.

The screenshot shows the 'Properties' panel for a data set. The 'Business Name' is 'Products Master Data'. The 'Technical Name' is 'D\_Products'. The 'Type' is 'Data Set'. Below this is a list of columns (23) with a search bar. A context menu is open over the 'CREATEDAT' column, showing options: 'Change Business Name', 'Set as Key', and 'Set Hidden'.

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



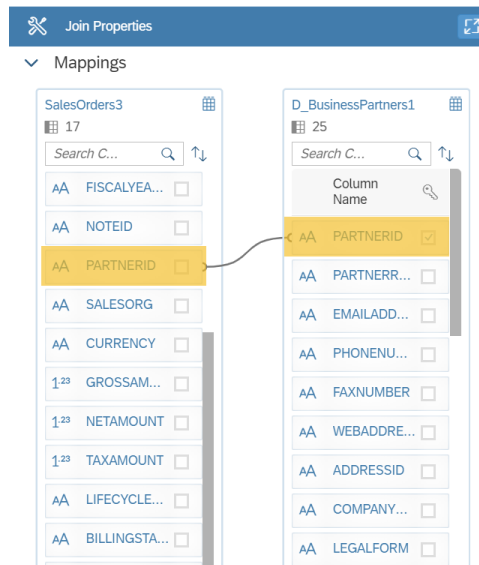
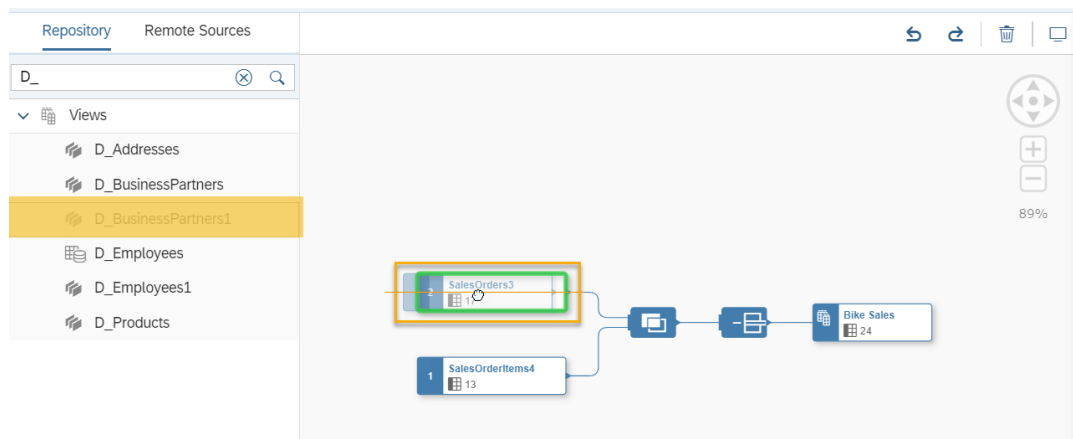
- Let's save and deploy the model. Name the new model “D\_Products”.



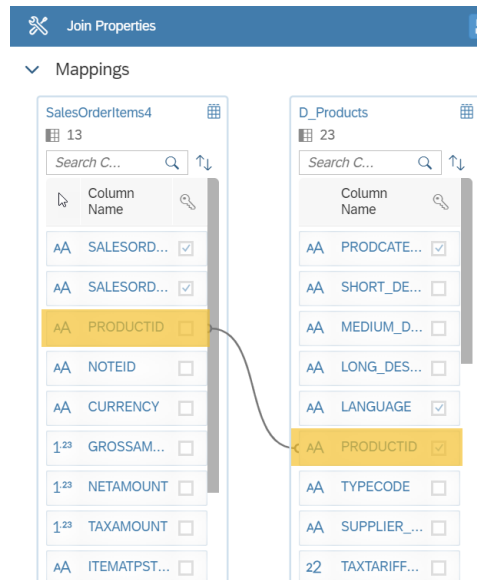
- To create the master data for the business partners, use the same schema. To join the tables *BusinessPartners* and *Addresses*, map the columns ADDRESSID. Save the model as a dimension with the name “D\_BusinessPartners”.

### Enhance Fact View

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



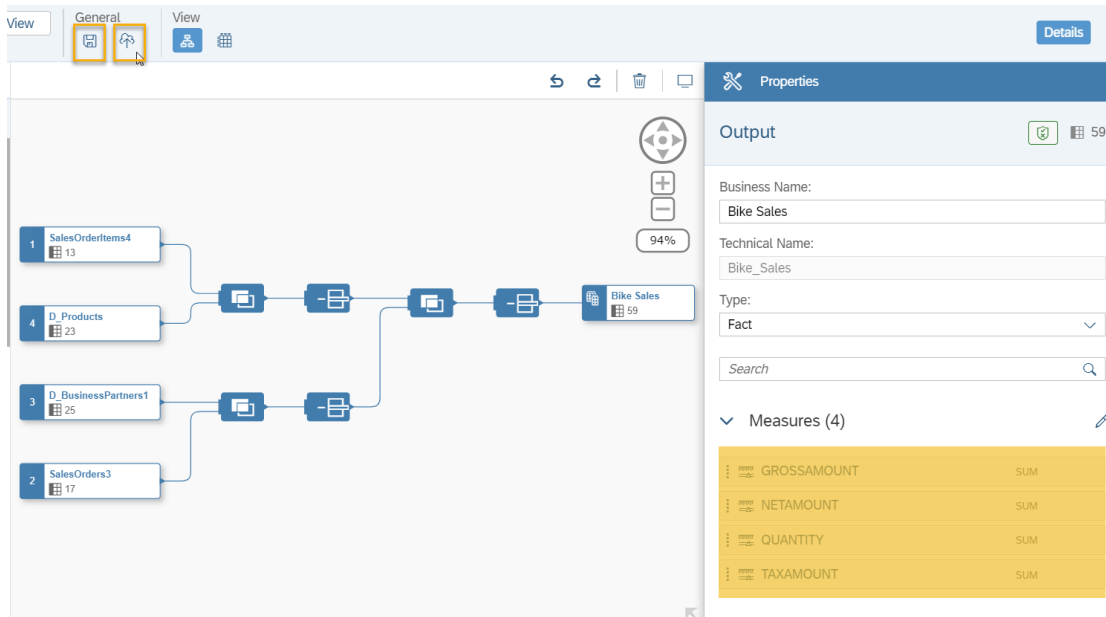
- The *D\_Products* model should be joined with the *SalesOrderItems* table. Therefore, map them on the



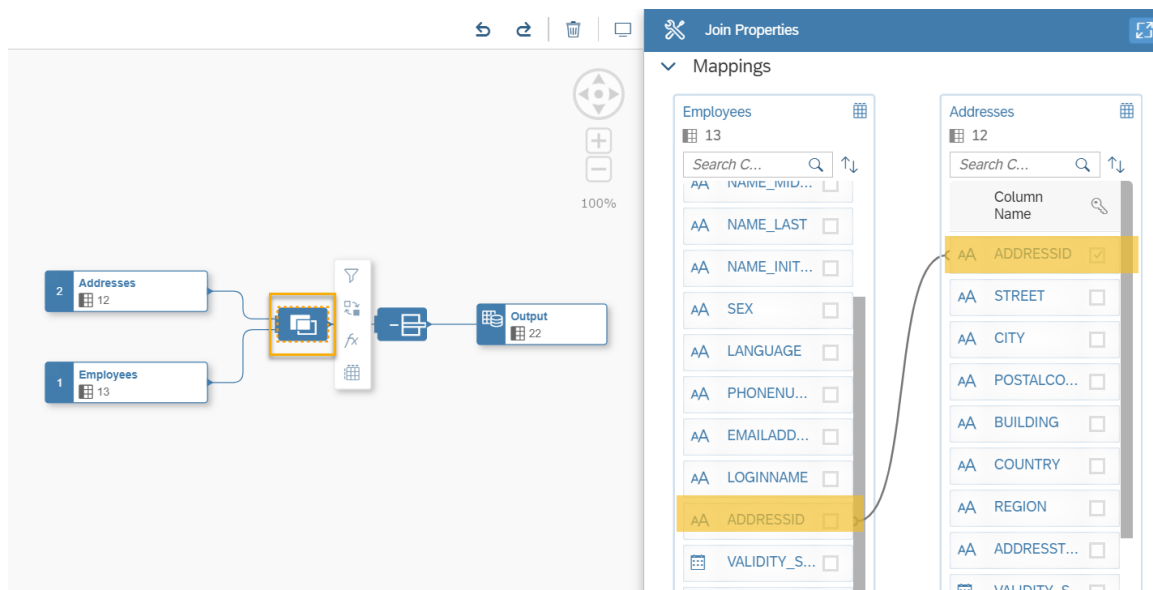
PRODUCTID as shown before.

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

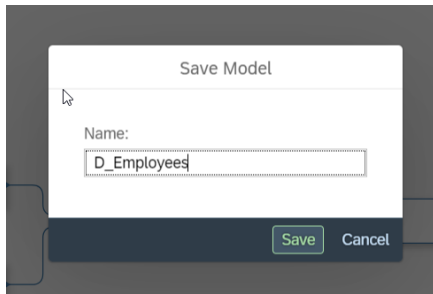
save and deploy.



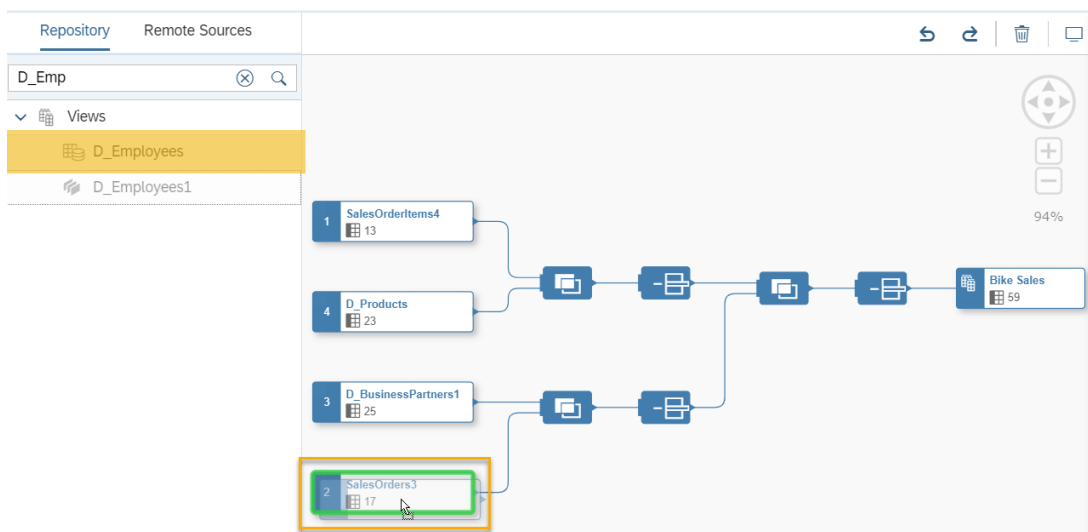
- For finding our Best Sales Representative, you first need to prepare the master data for the employees. The Employees table needs to be enriched with data about the addresses. For this reason, we need to create another new view, as explained in the steps before.
- This time join the Employees and the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



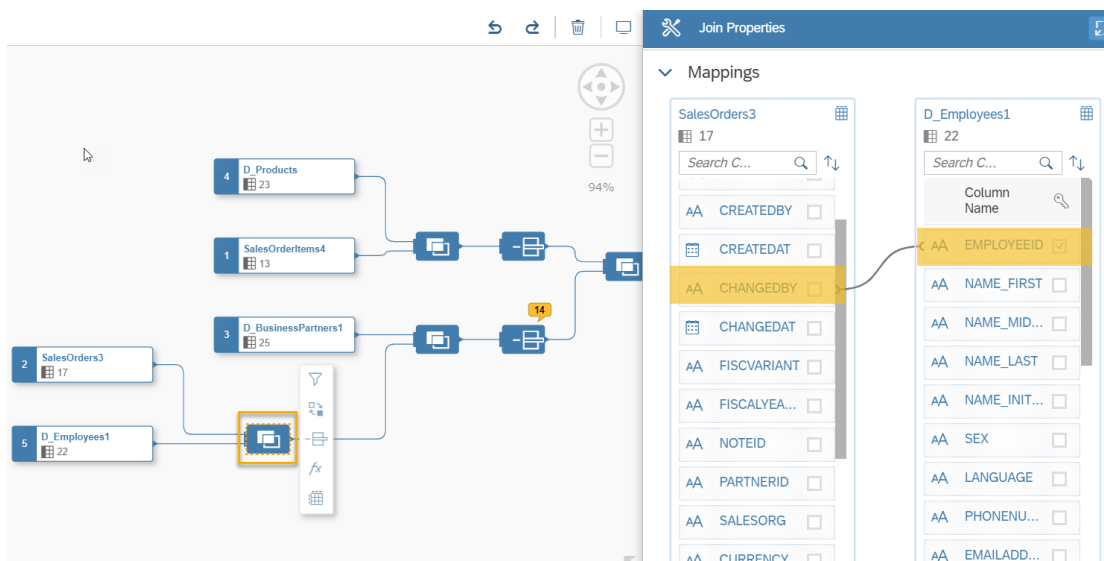
- When finished save and deploy the model under the name “D\_Employees”.



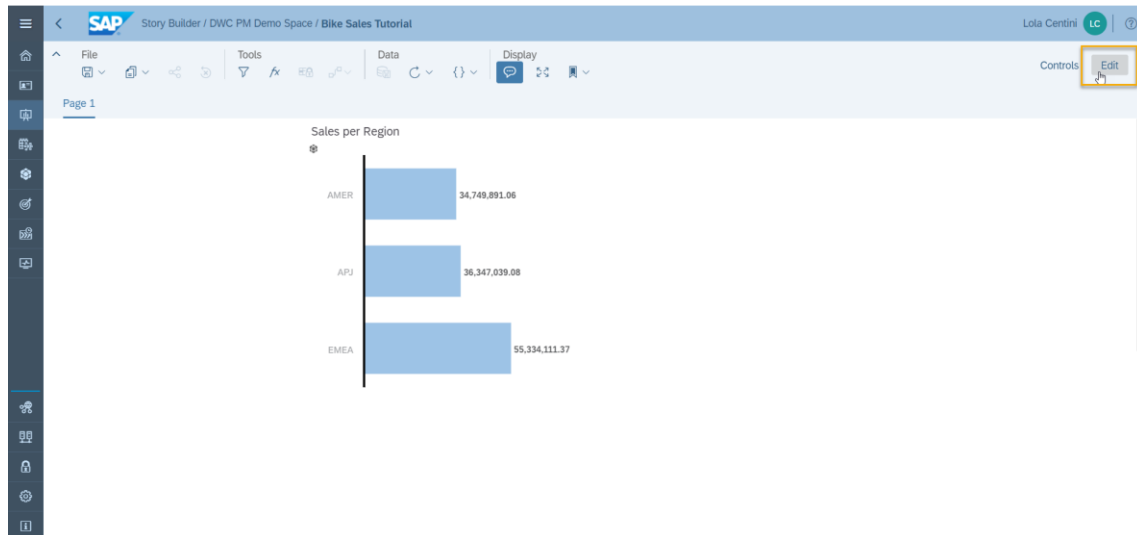
- To connect the employees to the sales orders, go back to the **Data Builder** and click on your *Bikes\_Sales* Fact to edit it.
- Let's add our Employees dimension. In the left panel go to “Repository” and find your *D\_Employees* in the “Views”. Drag and drop it onto the *SalesOrders* in the modeling area.



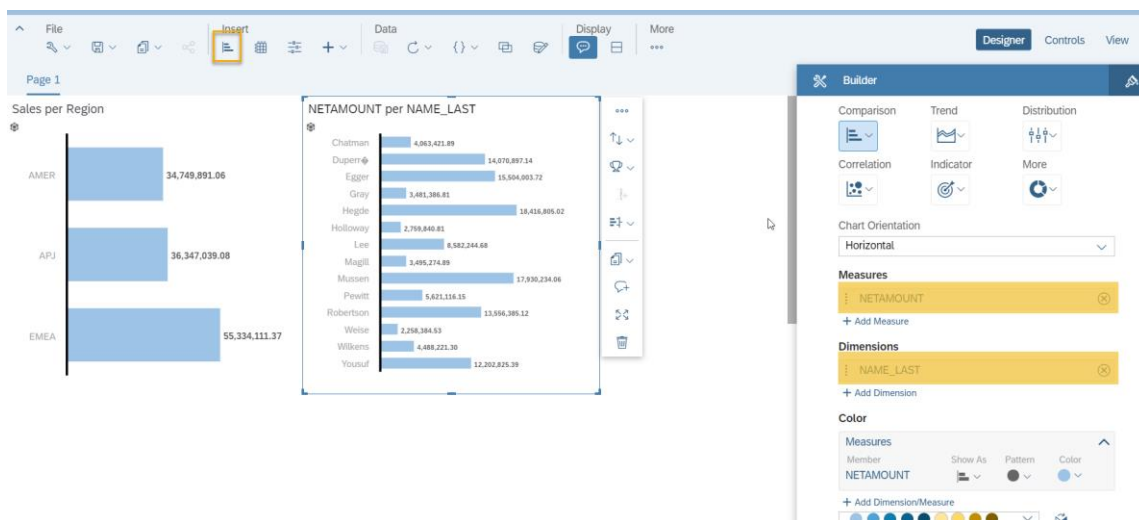
- This time map the EMPLOYEEID to the CREATEDBY column of the *SalesOrder* table. Don't forget to remove redundant columns from the model.



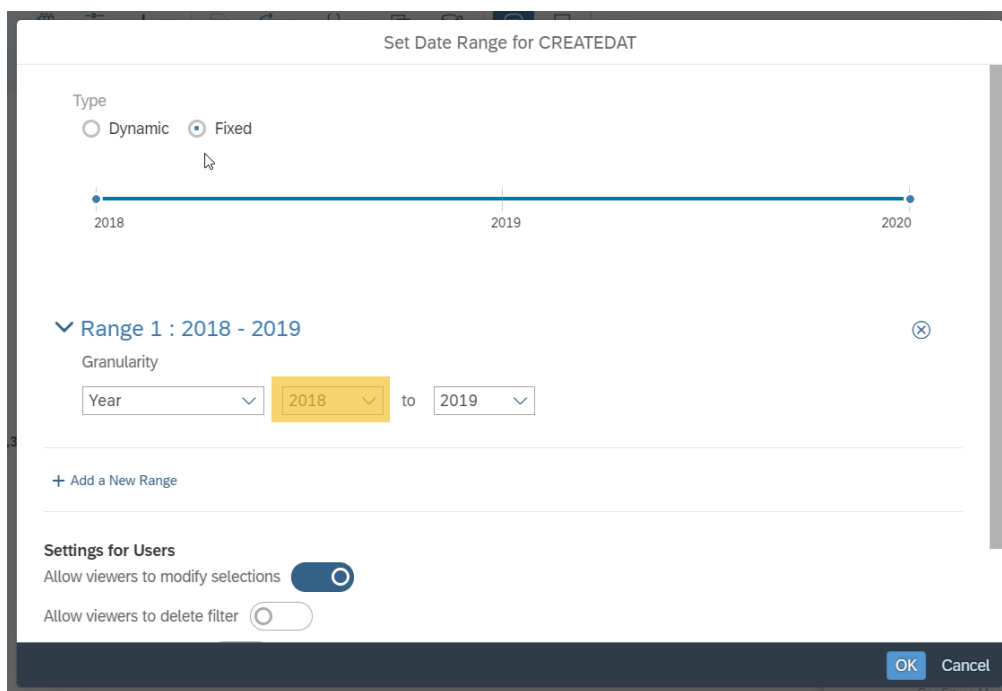
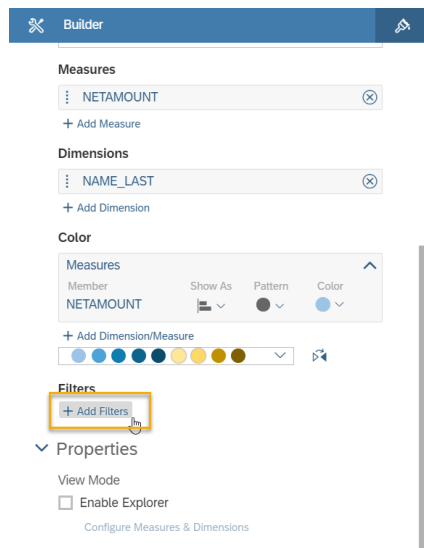
- Click in the output and check in the right panel, whether all measures are set to measures. You can already preview your data. When everything is correct, save and deploy.
- To continue with the visualization, navigate to the **Story Builder** and choose your space. Search for the “Bike Sales Tutorial” 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.

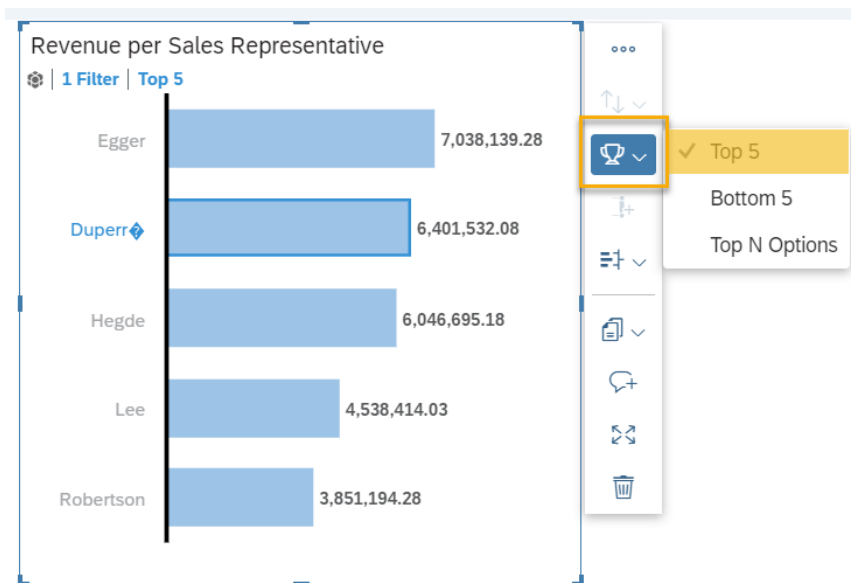


- Only the newer numbers are relevant for choosing the best sales representative. Filter for this year only. Choose the CREATEDAT (Range) for the filter. In the popup change the date range to 2019- 2019 to show this year's sales only.

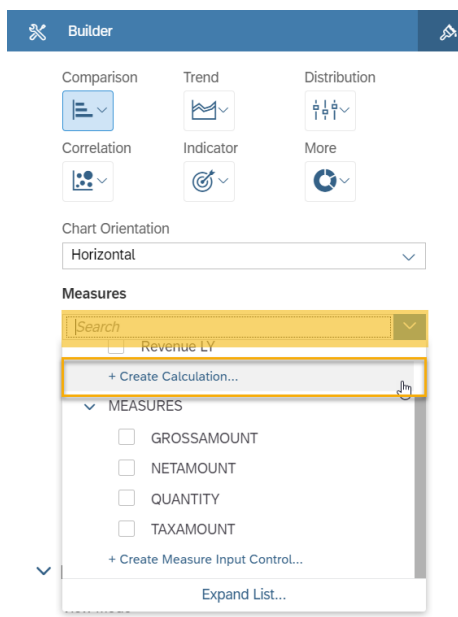


- To have a more speaking name change the title of the chart to “Revenue per Sales Representative”

- 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”.

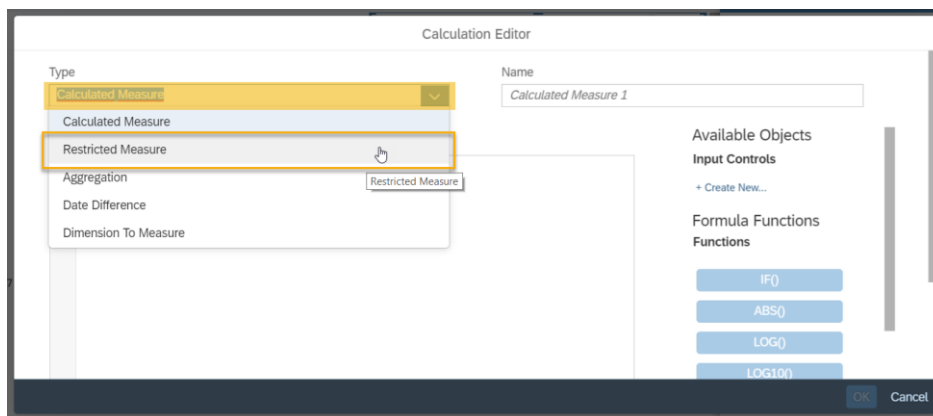


- To finish the requested reports the Year-over-Year Sales Comparison is needed. All the models are already prepared. In the SAC the numbers only need to be combined. To do so insert a new chart and then create a new calculated measure in the builder panel at the right.

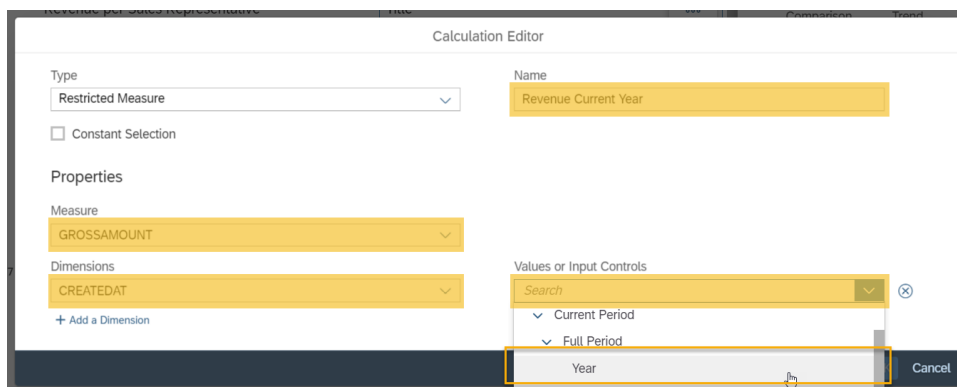




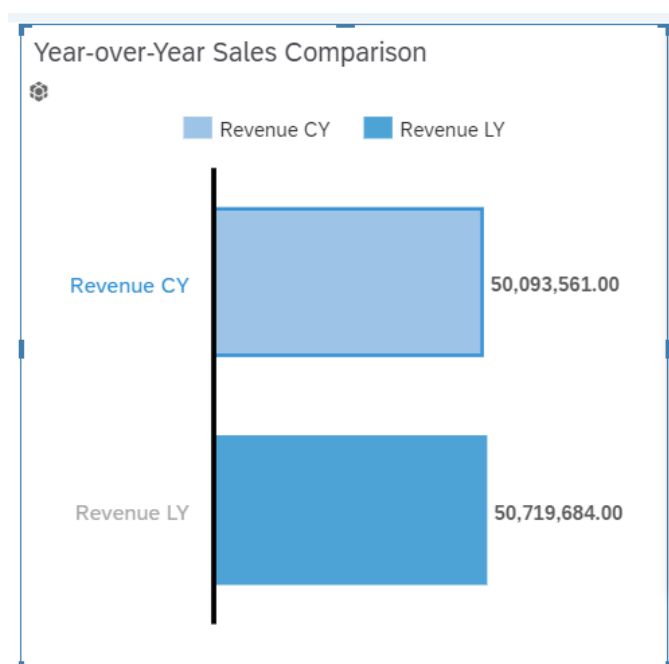
- In the following popup first change the type to a “Restricted Measure”.



- 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

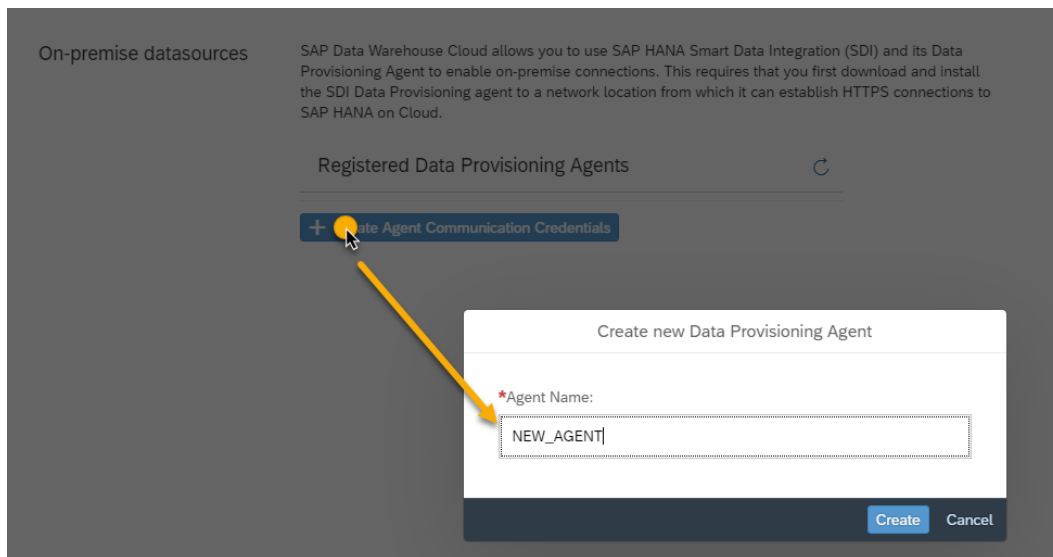
As a prerequisite to connect to your local HANA and other sources, the Data Provisioning Agent needs to be installed and configured. Follow the [installation guide](#) in the SAP Help Portal to install the Data Provisioning Agent. Install the latest version of HANA DP Agent 2.0 (as of Aug 6: SPS04)

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.

Agent settings

---

NEW\_AGENT **INITIALIZING**

Last successful connection: Not available

HANA Server:

HANA Port:

HANA via JDBC:

HANA Use SSL:

HANA User Name for Agent Messaging:

HANA User Password for Agent Messaging:

Agent Adapters:

<input type="checkbox"/>	HANA Adapter
<input type="checkbox"/>	ABAP Adapter

**i** Adapters can only be registered or unregistered when the agent is connected.

**Close**

**Don't close this windows 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`

```
*****
DPAgent Configuration Tool
*****
1. Agent Status
2. Start or Stop Agent
3. Agent Preferences
4. Remote Source Credentials
5. SSL Keystores
6. SAP HANA Connection
7. Agent Registration
8. Adapter Registration
9. Custom Adapters
10. Agent & Adapter Versions
q. Quit
b. Back
*****
Enter Option:2
*****
Start or Stop Agent
*****
1. Start Agent
2. Stop Agent
3. Ping Agent
q. Quit
b. Back
*****
Enter Option:1
```

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

```
*****
Secure Storage Utility
*****
1. Set HANA XS Username
2. Set HANA XS Password
3. Set proxy username
4. Set proxy password
5. Set default keystore password (for Cloud Connectivity)
6. Set Adapter keystore password
7. Set Adapter truststore password
8. Set FileAdapter Access Token
9. Set ExcelAdapter Access Token
10. Set OutlookAdapter Access Token
11. Set Custom secure Property
12. Set Adapter Secure Property
q. Quit
b. Back
*****
Enter Option:
```

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

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



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

Agent settings

NEW\_AGENT **INITIALIZING**

Last successful connection: Not available

HANA Server:

HANA Port:

HANA via JDBC:

HANA Use SSL:

HANA User Name for Agent Messaging:


HANA User Password for Agent Messaging:

Agent Adapters:


☐ HANA Adapter


☐ ABAP Adapter



*Adapters can only be registered or unregistered when the agent is connected.*

 Close

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:  ):



Registered Data Provisioning Agents 

NEW\_AGENT **CONNECTED**  

10.) Now enable the Adapters “HANA Adapter” and “ABAP ADAPTER”:

Edit the Data Provisioning Agent you have just created:

Registered Data Provisioning Agents 

NEW\_AGENT **CONNECTED**  

Select the adapters and close the dialog afterwards:

Agent Adapters:

- ☒ HANA Adapter
- ☒ ABAP Adapter

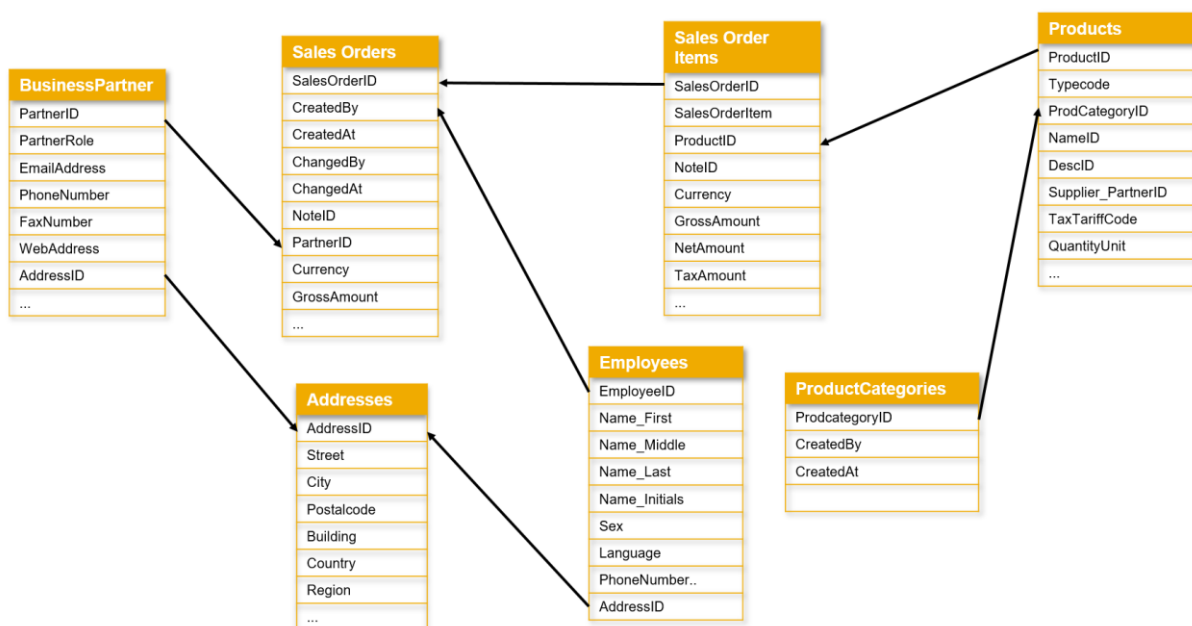
Close

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

## 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

## CSV Files: Data Type mapping

To adjust the data type while uploading CSV file, use the following information to have a proper data type assignment.

### 1.) Addresses

Key	Column	Data Type
X	ADDRESSID	String(10)
	CITY	String(40),
	POSTALCODE	String(10),

	STREET	String(60),
	BUILDING	String(10),
	COUNTRY	String(3),
	REGION	String(4),
	ADDRESSTYPE	String(2),
	VALIDITY_STARTDATE	DATE
	VALIDITY_ENDDATE	DATE
	LATITUDE	DOUBLE

## 2.) BusinessPartners

Key	Column	Data Type
X	PRODUCTID	String(10)
	TYPECODE	String(2)
	PRODCATEGORYID	String(2)
	CREATEDBY	String(10)
	CREATEDAT	DATE
	CHANGEDBY	String(10),
	CHANGEDAT	DATE
	SUPPLIER_PARTNERID	String(10)
	TAXTARIFFCODE	Integer
	QUANTITYUNIT	String(3)
	WEIGHTMEASURE	Decimal(13,3)
	WEIGHTUNIT	String(3)
	CURRENCY	String(5)
	PRICE	Decimal(15,2)
	WIDTH	Decimal(13,3)
	DEPTH	Decimal(13,3)
	HEIGHT	Decimal(13,3)
	DIMENSIONUNIT	String(3)
	PRODUCTPICURL	String(255)

## 3.) Employees

Key	Column	Data Type
X	EMPLOYEEID	String(10)
	NAME_FIRST	String(40)
	NAME_MIDDLE	String(40)
	NAME_LAST	String(40)
	NAME_INITIALS	String(10)
	SEX	String(1)
	LANGUAGE	String(2)
	PHONENUMBER	String(30)
	EMAILADDRESS	String(255)
	LOGINNAME	String(12)
	ADDRESSID	String(10)
	VALIDITY_STARTDATE	Date

	VALIDITY_ENDDATE	Date
--	------------------	------

#### 4.) ProductCategories

Key	Column	Data Type
X	PRODCATEGORYID	String(2)
	CREATEDBY	String(10)
	CREATEDAT	Date

5.) ProductCategoryText

Key	Column	Data Type
X	PRODCATEGORYID	String(2)
X	LANGUAGE	String(2)
	SHORT_DESCR	String(20)
	MEDIUM_DESCR	String(40)
	LONG_DESCR	String(10)

6.) Products

Key	Column	Data Type
X	PRODUCTID	String(10)
	TYPECODE	String(2)
	PRODCATEGORYID	String(2)
	SUPPLIER_PARTNERID	String(10)
	TAXTARIFFCODE	INTEGER
	QUANTITYUNIT	String(3)
	WEIGHTMEASURE	Decimal(13,3)
	WEIGHTUNIT	String(3)
	CURRENCY	String(5)
	PRICE	Decimal(15,2)
	WIDTH	Decimal(13,3)
	DEPTH	Decimal(13,3)
	HEIGHT	Decimal(13,3)
	DIMENSIONUNIT	String(3)
	CREATEDBY	String(10)
	CREATEDAT	Date
	CHANGEDBY	String(10)
	CHANGEDAT	Date
	PRODUCTPICURL	String(255)

7.) ProductTexts

Key	Column	Data Type
X	PRODUCTID	String(10)
X	LANGUAGE	String(2)
	SHORT_DESCR	String(20)
	MEDIUM_DESCR	String(40)

	LONG_DESCR	String(10)
--	------------	------------

8.) SalesOrderItems

Key	Column	Data Type
X	SALESORDERID	String(10)
X	SALESORDERITEM	String(10)
	PRODUCTID	String(10)
	NOTEID	String(10)
	CURRENCY	String(5)
	GROSSAMOUNT	Decimal(15,2)
	NETAMOUNT	Decimal(15,2)
	TAXAMOUNT	Decimal(15,2)
	ITEMATPSTATUS	String(1)
	OPITEMPOS	String(10)
	QUANTITY	Decimal(13,3)
	QUANTITYUNIT	String(3)
	DELIVERYDATE	Date

9.) SalesOrders

Key	Column	Data Type
X	SALESORDERID	String(10)
	CREATEDBY	String(10)
	CREATEDAT	Date
	CHANGEDBY	String(10)
	CHANGEDAT	Date
	FISCVARIANT	String(2)
	FISCALYEARPERIOD	String(7)
	NOTEID	String(10)
	PARTNERID	String(10)
	SALESORG	String(4)
	CURRENCY	String(5)
	GROSSAMOUNT	Decimal(15,2)
	NETAMOUNT	Decimal(15,2)
	TAXAMOUNT	Decimal(15,2)
	LIFECYCLESTATUS	String(1)
	BILLINGSTATUS	String(1)
	DELIVERYSTATUS	String(1)







[www.sap.com/contactsap](http://www.sap.com/contactsap)

© 2019 SAP SE or an SAP affiliate company. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP SE or an SAP affiliate company.

The information contained herein may be changed without prior notice. Some software products marketed by SAP SE and its distributors contain proprietary software components of other software vendors. National product specifications may vary.

These materials are provided by SAP SE or an SAP affiliate company for informational purposes only, without representation or warranty of any kind, and SAP or its affiliated companies shall not be liable for errors or omissions with respect to the materials. The only warranties for SAP or SAP affiliate company products and services are those that are set forth in the express warranty statements accompanying such products and services, if any. Nothing herein should be construed as constituting an additional warranty.

In particular, SAP SE or its affiliated companies have no obligation to pursue any course of business outlined in this document or any related presentation, or to develop or release any functionality mentioned therein. This document, or any related presentation, and SAP SE's or its affiliated companies' strategy and possible future developments, products, and/or platform directions and functionality are all subject to change and may be changed by SAP SE or its affiliated companies at any time for any reason without notice. The information in this document is not a commitment, promise, or legal obligation to deliver any material, code, or functionality. All forward-looking statements are subject to various risks and uncertainties that could cause actual results to differ materially from expectations. Readers are cautioned not to place undue reliance on these forward-looking statements, and they should not be relied upon in making purchasing decisions.

SAP and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP SE (or an SAP affiliate company) in Germany and other countries. All other product and service names mentioned are the trademarks of their respective companies. See <https://www.sap.com/copyright> for information and notices.

**THE BEST RUN**

