

PUBLIC

# **Integrating SAP Analytics Cloud and SAP Data Warehouse Cloud**

Session ID: ANA361

Exercises / Solutions

Amogh Kulkarni, SAP

Ingo Hilgefort, SAP

## TABLE OF CONTENTS

1	REQUIRED RESOURCES FOR EXERCISES .....	3
1.1	Download Required Resources.....	3
1.2	System Details after your workshop .....	3
2	EXERCISE OVERVIEW .....	4
2.1	SAP Data Warehouse Cloud.....	4
2.2	SAP Analytics Cloud.....	4
2.3	SAP Data Warehouse Cloud & SAP analytics Cloud .....	4
3	SAP DATA WAREHOUSE CLOUD EXERCISES.....	5
3.1	SAP Data Warehouse Cloud – Exercise 01: First Log On.....	5
3.2	SAP Data Warehouse Cloud – Exercise 02: Create Your first Space.....	8
3.3	SAP Data Warehouse Cloud – Exercise 03: Importing Tables .....	11
3.4	SAP Data Warehouse Cloud – Exercise 04: Uploading Data .....	15
3.5	SAP Data Warehouse Cloud – Exercise 05: Creating Sales Hierarchy .....	17
3.6	SAP Data Warehouse Cloud – Exercise 06: Creating the View .....	19
3.7	SAP Data Warehouse Cloud – Exercise 07: Adding Hierarchy to the View .....	31
3.8	SAP Data Warehouse Cloud – Exercise 08: Configuring Label Columns.....	35
3.9	SAP Data Warehouse Cloud – Exercise 09: Adding Calculated Columns to the View.....	40
4	SAP ANALYTICS CLOUD EXERCISES .....	47
4.1	SAP Analytics Cloud - Exercise 01: Total Sales by Customer / Product / Sales Person .....	47
4.2	SAP Analytics Cloud - Exercise 02: Which Product receives the highest Discount? .....	53
4.3	SAP Analytics Cloud - Exercise 03: Which Sales Person is giving the most discount? .....	59
4.4	SAP Analytics Cloud - Exercise 04: Which Product is the most profitable product?.....	63
5	SAP DATA WAREHOUSE CLOUD – SELF SERVICE MODELLING .....	67
5.1	SAP Data Warehouse Cloud – Exercise 10: Uploading Additional Product Information.....	67
5.2	SAP Data Warehouse Cloud – Exercise 11: Adjusting View .....	71
5.3	SAP Analytics Cloud - Exercise 05: Adding Product Cost for 2020 .....	76

## 1 REQUIRED RESOURCES FOR EXERCISES

### 1.1 Download Required Resources

As part of the following set of exercises in SAP Data Warehouse Cloud as well as SAP Analytics Cloud, you will need a set of files. All relevant files can be downloaded from the GitHub site:

<https://github.com/SAP-samples/teched2020-ANA361>

Following the link above, you will find the exercise steps as well as a file named ANA361\_RESOURCES.ZIP.

Please download the file to your local machine and unzip the file into a separate folder.

The ZIP File contains 1 folder with CSV Files and 1 folder with JSON files.

### 1.2 System Details after your workshop

In case you are interested to follow the steps after the live online SAP TechED Sessions have finished, you can request your own SAP Data Warehouse Cloud tenant following this URL:

<https://saphanajourney.com/data-warehouse-cloud/trial/>

You will get a system with all components and can follow the step from the material outlined here.

Please note, that during the exercises mentioned here, you will see we talk about ANA361\_XX where XX is replaced with your user number. In case you requested your own trial system, you will get your own user account and you can then create the objects based on your own accounts / naming convention.

## **2 EXERCISE OVERVIEW**

In the following sections you will find exercises for SAP Data Warehouse Cloud and SAP Analytics Cloud. Some of the exercises are based on each other and some of the exercises are working on prebuilt / preconfigured materials to save you time. Below you will find some brief outlines on the exercises and which exercises are based on each other, so that you can decide which exercises you would like to focus on.

### **2.1 SAP Data Warehouse Cloud**

In section 4 of this document you start with the exercises in SAP Data Warehouse Cloud.

- SAP Data Warehouse Cloud – Exercise 01: First Log On
- SAP Data Warehouse Cloud – Exercise 02: Create Your first Space
- SAP Data Warehouse Cloud – Exercise 05: Importing Tables
- SAP Data Warehouse Cloud – Exercise 06: Uploading Data
- SAP Data Warehouse Cloud – Exercise 07: Creating Sales Hierarchy
- SAP Data Warehouse Cloud – Exercise 08: Creating the View
- SAP Data Warehouse Cloud – Exercise 08: Adding Hierarchy to the View
- SAP Data Warehouse Cloud – Exercise 09: Configuring Label Columns
- SAP Data Warehouse Cloud – Exercise 10: Adding Calculated Columns to the View

### **2.2 SAP Analytics Cloud**

The exercises in section 5 are based on the models that you created in SAP Data Warehouse Cloud in section 4.

### **2.3 SAP Data Warehouse Cloud & SAP analytics Cloud**

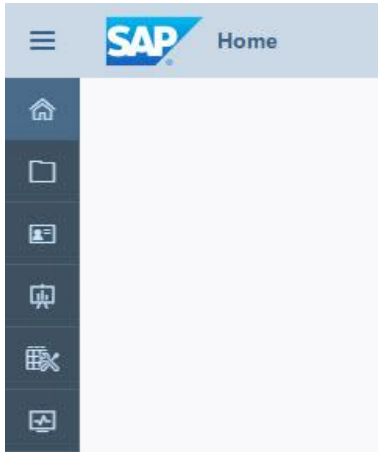
The exercises in section 6 are based on the models that you created in SAP Data Warehouse Cloud in section 4. You do not have to finish the exercises for SAP Analytics Cloud in section 5 to be able to use the updated model in section 6.

### 3 SAP DATA WAREHOUSE CLOUD EXERCISES

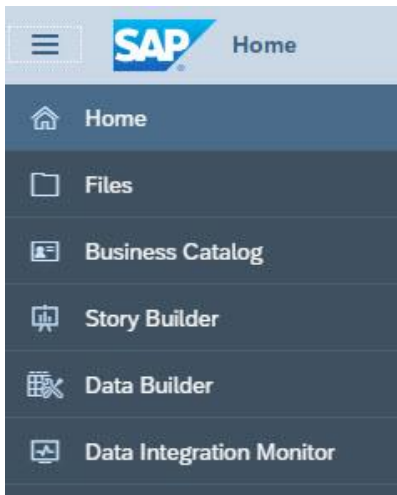
#### 3.1 SAP Data Warehouse Cloud – Exercise 01: First Log On

After you received your SAP Data Warehouse Cloud credential, please follow these steps:

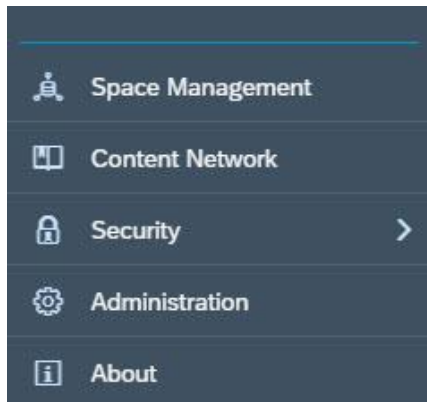
1. Open Google Chrome and log on to your SAP Data Warehouse Cloud trial.
2. In the top left of the start screen you will find the menu options.



3. By clicking on the menu in the top left (the icon with the three stripes), you can expand the menu to also show the menu text.



4. The bottom part of the menu shows the administrative functions.



5. Now let's clarify the different areas:

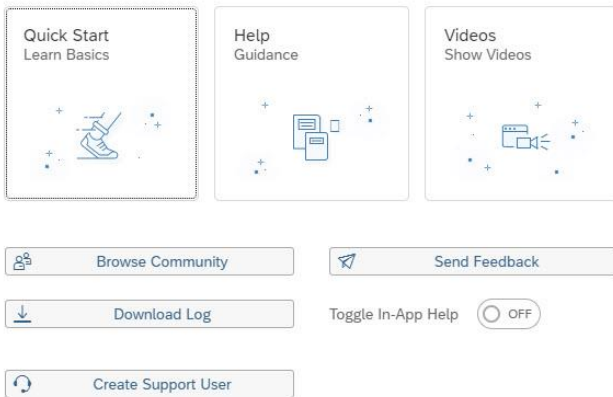
- Files: This is your repository where you will find all your assets that you created. Think off it as your Windows Explorer for your SAP Data Warehouse Cloud.
- Business Catalog: This gives you access to the Business Catalog of SAP Data Warehouse Cloud and you can search for the different assets in your SAP Data Warehouse Cloud based on descriptions and tags or assigned teams.
- Story Builder: This is the embedded version of SAP Analytics Cloud, which provides you access to all the augmented analytics capabilities.
- Data Builder: This is where you create all the different asset types, such as tables, views, and entity relationship models and where you do your data modeling.
- Data Integration Monitor: Here you receive an overview on all tables from remote data sources, such as your SAP HANA on-premise system, and you can configure if the table should be a remote data source, or if you would like to replicate (one or regular) the information into your SAP Data Warehouse Cloud.

6. Now let's take a look at the menu items from the Administrative section:

- Space Management: Spaces are a fundamental concept of SAP Data Warehouse Cloud and we will clarify the Spaces concept later on in this section.
- Content Network: The Content Network on the one hand provides you access to Sample content and Business Content from SAP and from SAP's partner network and on the other hand the Content Network is the area where you would export / import your own content for content lifecycle purposes.
- Security: This is the area where you create Users and Roles and also monitor traced activities of your system.
- Administration: This is the area where you configure your Data Provisioning Agents to gain access to your on-premise data sources, as well as the IP Whitelisting entries for the same purpose.

7. When you now navigate to the top right corner, please click on the  icon to launch the Help dialog.

What can we help you with today?

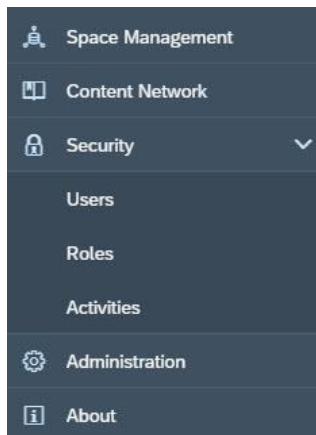


8. In the Help dialog you have multiple options from some Quick Start help to reach out to our support team.

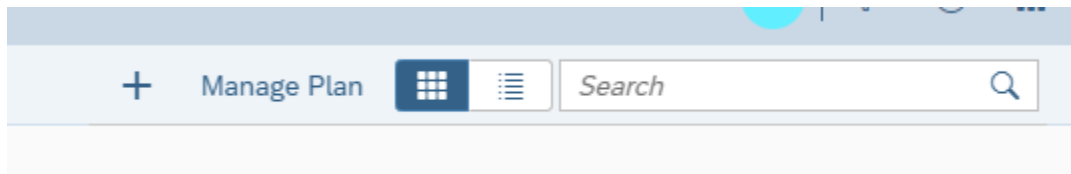
### 3.2 SAP Data Warehouse Cloud – Exercise 02: Create Your first Space

Spaces as part of the SAP Data Warehouse Cloud solution are virtual team environments where your administrator has the ability to assign users and roles, as well as additional resources, such as connections to data sources, and allocated space. In SAP Data Warehouse Cloud all data related workflows start with the selection of a Space, so you can see the Space is a fundamental concept and therefore we need to setup our Space as our first step.

1. Log On to your SAP Data Warehouse Cloud system.
2. In the menu on the left-hand side, select the option Space Management.



3. After you selected the menu item, you will be presented with a list of existing Spaces and you have the ability to create a new Space.



4. Use the “+” symbol to start the process to create a new Space (top right corner)





## Session ID: ANA361 - Integrating SAP Analytics Cloud and SAP Data Warehouse Cloud

5. Enter a Space Name – for our example use the following details:

- Space Name: ANA361\_XX (Replace XX with your assigned user number)
- Space ID: ANA361\_XX (Replace XX with your assigned user number)

6. The Space ID will be suggested based on your Space Name, but you have the option to change it as well.

7. Click Create.

8. You are now being presented with the properties of your new Space and you have the ability to configure the following options:

- As part of the Storage Assignment you can decide, how much storage space overall you will allocate to the Space and how much of the assigned storage space you will assign to the In-Memory allocation.
- You can assign the Space Priority, which will become relevant when multiple Spaces are sending requests to the system and the assigned priority will then help to decide which request takes priority.

9. In our example we will use the default assignments for the Storage Assignment as well as the Space Priority.

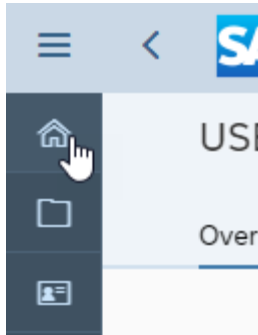
10. Now use the “Add” option in the Members area to assign your user to the Space.

11. Select the user matching your assigned user from the list of available users.

12. Click Add.

13. In the Connections area we can later assign connections to remote data sources.

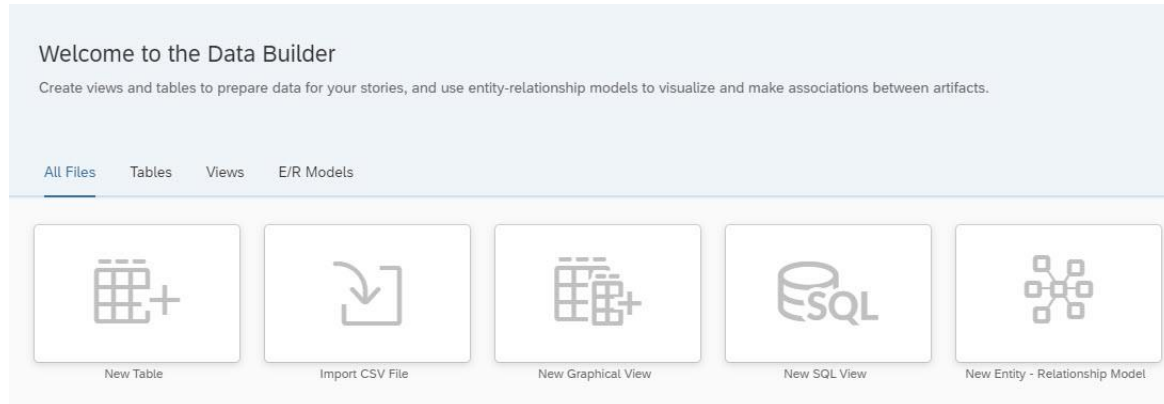
14. Click on the “Home” menu option to navigate back to the start screen.



You just created your first Space in SAP Data Warehouse Cloud and you can now start your next step and create your first table and model.

### 3.3 SAP Data Warehouse Cloud – Exercise 03: Importing Tables

Before we are going to start with our first exercise by importing a pre-defined set of tables, let's clarify the different asset types that you can create in SAP Data Warehouse Cloud. When you launch the Data Builder from the menu, you will be presented with this screen:

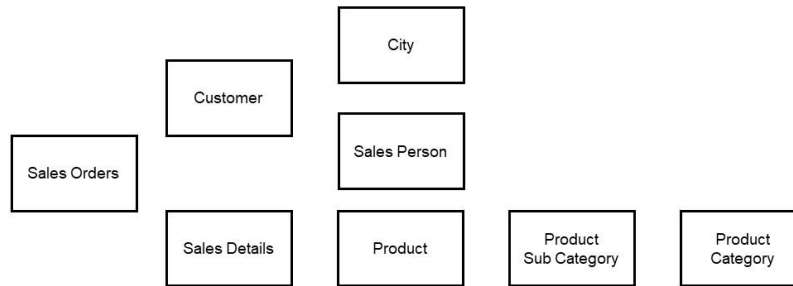


So, let's now look at the different asset types:

- **Table:** Here you basically define a new table from scratch, and you configure each field of the table and you do have the ability then to upload data to this table later on.
- **Import CSV File:** Here you also define a Table, but instead of defining the table and fields first, you are using a CSV file and upload the information in the CSV File as first step. You can then use the data wrangling capabilities with the information from the CSV File and then create the table with the data from the CSV File in one step.
- **Graphical View:** In the Graphical View you can leverage Tables and Views to create new Views using a visual interface.
- **SQL View:** In the SQL View you can leverage Tables and Views to create new Views by using SQL directly.
- **Entity Relationship Model:** Here you define the relationships between Tables or Views, which then are being leveraged when you create a new View based on the Tables or Views.

In this section we will start importing the tables for our sample models and then upload the raw data to those tables. In the first part of this overall section we will take a look at the sample model and which tables we will need. In the second part we will then import those tables in SAP Data Warehouse Cloud and finally in the third part, we will upload the raw data to those newly created tables, and we will also setup a hierarchy as part of the tables.

**SAP Data Warehouse Cloud**  
**Sample Model & Table**



© 2020 SAP SE or an SAP affiliate company. All rights reserved. | PUBLIC

1

The image above shows the sample model with the Sales Orders and the Sales Details being the two most important information, providing the header and detailed information for our transactions. In addition, we have the information for our customers, which then also allows us to join the information on the assigned Sales Person and the geographic information on each customers. Based on the Sales Details, we can then also bring in additional information for the Product, Product Sub Category, and the Product Category. After we created the model and uploaded the information in SAP Data Warehouse Cloud, we will then try to answer the following questions:

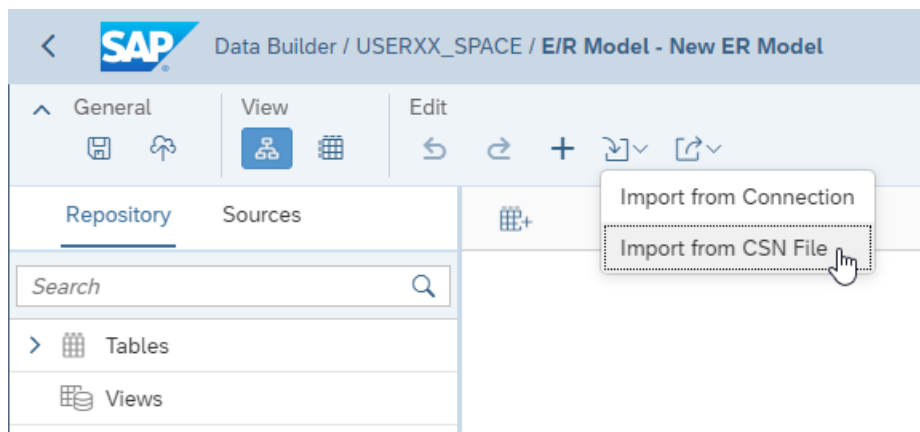
- What is the total sales by customer / by product / by sales person?
- Which product receives the highest discount?
- Which salesperson gives the most discount?
- Which product is the most profitable?
- Who is the most successful sales person? Sales manager?
- Who are our best customers?

## Session ID: ANA361 - Integrating SAP Analytics Cloud and SAP Data Warehouse Cloud

The files for these exercises are in the folder “Export\_JSON” from the ZIP file you downloaded.

We will now import the table definitions in form of a JSON file by importing an Entity Relationship Model.

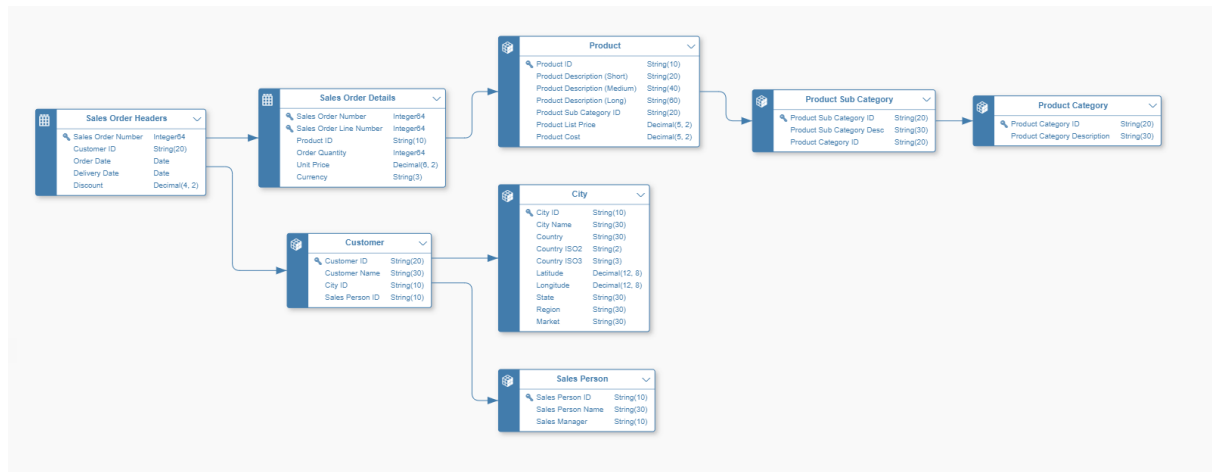
1. Log On to your SAP Data Warehouse Cloud tenant.
2. Select the menu option Data Builder on the left-hand side.
3. In case you are being asked, select your space that you created in the previous exercise.
4. Select the option New Entity – Relationship Model.
5. In the New Entity Relationship Model, navigate to the toolbar.
6. Navigate to the Edit menu.



7. Open the Import option.
8. Select the option Import from CSN File.
9. Click Browse.
10. Select the file “Entity\_Relationship\_Model.json” from where you unzipped the file in the folder Export\_JSON.
11. Click Next.

Select Objects to Import			
Search			
<input type="checkbox"/> Business Name	Technical Name	Type	Status
<input type="checkbox"/> City	City	Dimension (Table)	Ready to Import
<input type="checkbox"/> Customer	Customer	Dimension (Table)	Ready to Import
<input type="checkbox"/> Product	Product	Dimension (Table)	Ready to Import
<input type="checkbox"/> Product Category	Product_Category	Dimension (Table)	Ready to Import
<input type="checkbox"/> Product Sub Category	Product_Sub_Category	Dimension (Table)	Ready to Import
<input type="checkbox"/> Sales Order Details	Sales_Order_Details	Local Table	Ready to Import
<input type="checkbox"/> Sales Order Headers	Sales_Order_Headers	Local Table	Ready to Import
<input type="checkbox"/> Sales Person	Sales_Person	Dimension (Table)	Ready to Import

12. Select all tables.
13. Click Import.
14. You will receive a message about the import being completed and the tables will be shown on the canvas in form of the imported Entity Relationship Model.



15. Save the changes to your Entity Relationship model using the Save icon in the General menu (top left)

16. Enter the following details:

- Business Name ANA361\_XX Entity Model (replace XX with your user number)
- Technical Name ANA361\_XX\_Entity\_Model (replace XX with your user number)

17. Click Save.

18. Deploy the model using the Deploy icon next to the Save icon (top left).

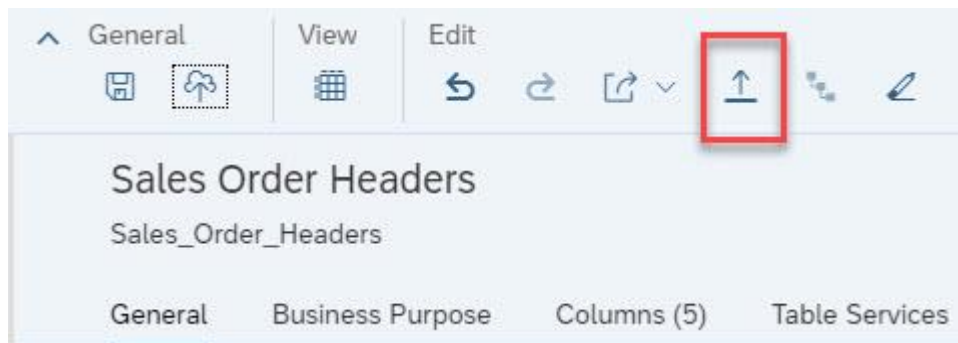
As part of the deployment of the Entity Relationship Model, also the underlying tables are being imported and deployed. When you navigate to the Data Builder screen, you should now see a set of 8 tables / dimensions and 1 Entity Relationship Model.

All Files   Tables   Views   E/R Models			
<div> <div>New Table</div> <div>Import CSV File</div> <div>New Graphical View</div> <div>New SQL View</div> <div>New Entity - Relationship Model</div> </div>			
Files (9)			
<input type="checkbox"/>	Business Name	Technical Name	Type
<input type="checkbox"/>	City	City	Dimension (Local Table)
<input type="checkbox"/>	Customer	Customer	Dimension (Local Table)
<input type="checkbox"/>	Product	Product	Dimension (Local Table)
<input type="checkbox"/>	Product Category	Product_Category	Dimension (Local Table)
<input type="checkbox"/>	Product Sub Category	Product_Sub_Category	Dimension (Local Table)
<input type="checkbox"/>	Sales Order Details	Sales_Order_Details	Relational Dataset (Local Table)
<input type="checkbox"/>	Sales Order Headers	Sales_Order_Headers	Relational Dataset (Local Table)
<input type="checkbox"/>	Sales Person	Sales_Person	Dimension (Local Table)
<input type="checkbox"/>	UserXX Entity Model	UserXX_Entity_Model	E/R Model

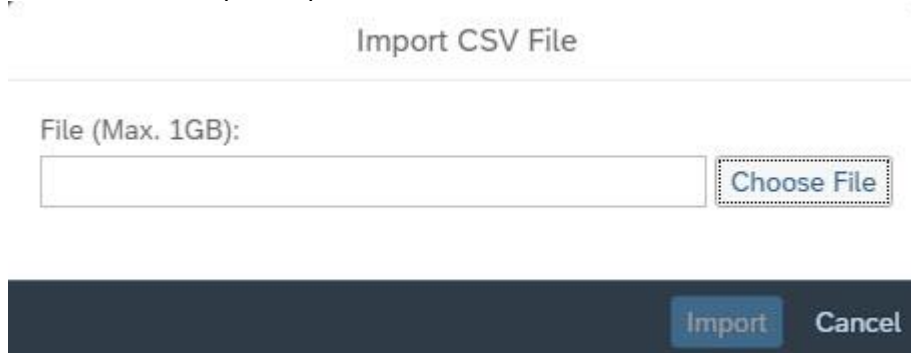
### 3.4 SAP Data Warehouse Cloud – Exercise 04: Uploading Data

All the required CSV files for the next set of steps are in the folder “CSV Source files” from the downloaded ZIP File. We will now upload the CSV files into the corresponding tables.

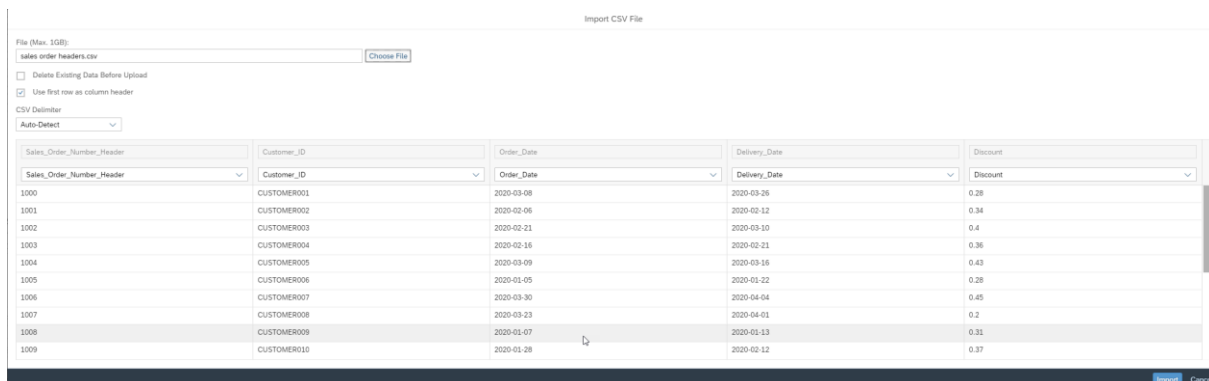
1. Log On to your SAP Data Warehouse Cloud tenant.
2. Select the menu option Data Builder on the left-hand side.
3. Use a double-click on the entry Sales Order Headers.



4. Select the menu option Upload Data from CSV File in the toolbar.



5. Click Choose File.
6. Navigate to where you unzipped the download.
7. Select the file “sales order headers.csv” from the folder CSV Source Files.



8. Ensure the option Use first row as column header is enabled.

9. Ensure the CSV Delimiter option is set to Auto-Detect.
10. Ensure that all columns of the table have a mapped column from the CSV File (2<sup>nd</sup> row in the screen).
11. Click Import.
12. You should receive a message about the successful import of the information.
13. There is no need to save or deploy the table after you imported the data.

You can now continue and repeat the steps for the other tables. Each table does have a corresponding CSV file as part of the download. Please upload the data for the following tables:

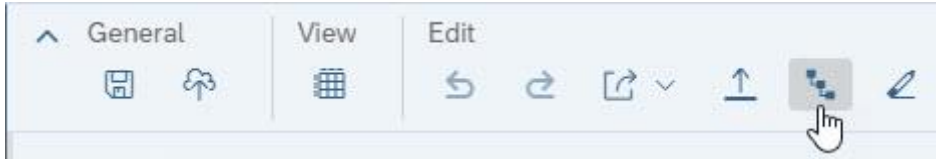
- City
- Customer
- Product
- Product Category
- Product Sub Category
- Sales Person
- Sales Order Details



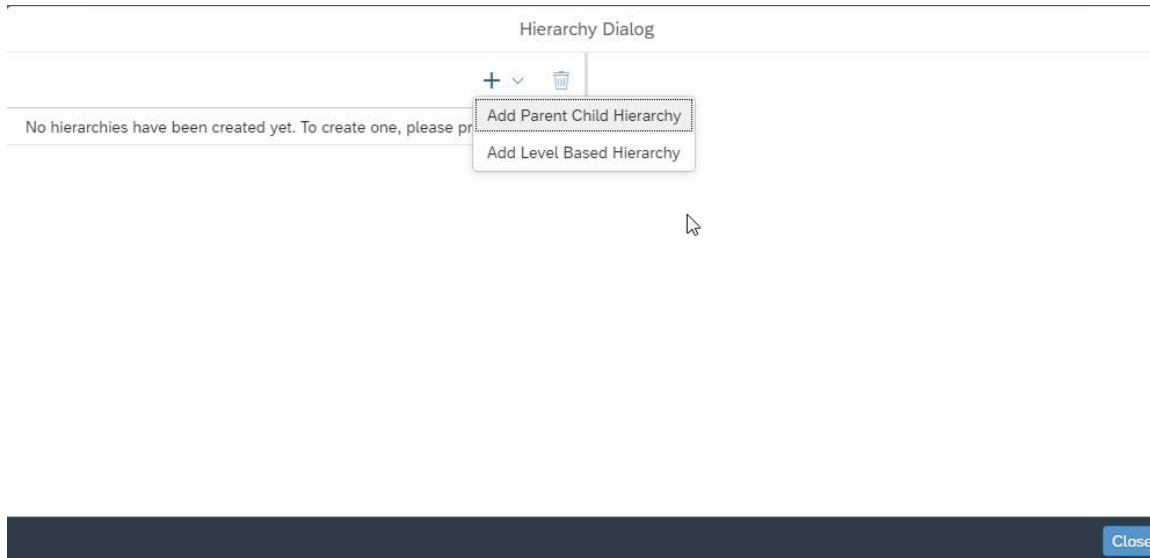
### 3.5 SAP Data Warehouse Cloud – Exercise 05: Creating Sales Hierarchy

After we uploaded all the data into the corresponding tables, we can now create our first hierarchy based on the information we just uploaded.

1. Log On to your SAP Data Warehouse Cloud tenant.
2. Select the menu option Data Builder on the left-hand side.
3. Select the entry Sales Person.
4. Use a double-click to open the details.
5. In the toolbar, select the option to create a new hierarchy.



6. In the next step, click the “+” icon and select the option Add Parent Child Hierarchy.



7. In the next screen you can enter the details for the hierarchy.

The screenshot shows the 'Hierarchy Dialog' interface. On the left, a tree view displays a hierarchy with 'Hierarchy' as the parent and 'Parent-Child Hierarchy' as the child. The right side of the dialog contains four input fields: 'Business Name' (filled with 'Hierarchy'), 'Technical Name' (filled with 'Hierarchy'), 'Parent Column' (filled with 'Sales Person ID'), and 'Child Column' (filled with 'Sales Person ID'). A 'Close' button is located at the bottom right of the dialog.

8. Enter the following details:

- Business Name Sales Manager Hierarchy
- Technical Name Sales\_Hry
- Parent Column Sales Manager
- Child Column Sales Person ID

9. Click Close.

10. Save your changes.

11. Deploy your changes.

We will use the hierarchy later as part of our views in SAP Data Warehouse Cloud as well as part of our dashboard in SAP Analytics Cloud.

### 3.6 SAP Data Warehouse Cloud – Exercise 06: Creating the View

In the previous exercise we created the tables and we created the Entity Relationship model. In the next steps we will now create our first view, combining the tables into an asset in SAP Data Warehouse Cloud, which then can be consumed in SAP Analytics Cloud.

A View in SAP Data Warehouse Cloud provides you with several benefits:

- Graphical or script-based editor
- Define views on top of remote, replicated data sources, or tables
- Define unions and joins, rename and remove columns, add calculations and filters
- Create Analytical Datasets, Dimensions or Relational Datasets
- Create Parent-Child or Level-based hierarchies in Dimension views
- Define measures & attributes in Analytical Datasets
- In the Graphical View Builder, you can compute and display the corresponding SQL Statement.

A View in SAP Data Warehouse Cloud allows you to leverage local tables, remote tables, or views and combine those into a new View. A View can also contain additional elements, such as filters and calculated columns and a view is consumable in SAP Analytics Cloud.

When you are creating the overall View, which you then will leverage as part of your analytics using SAP Analytics Cloud, you can follow two main approaches:

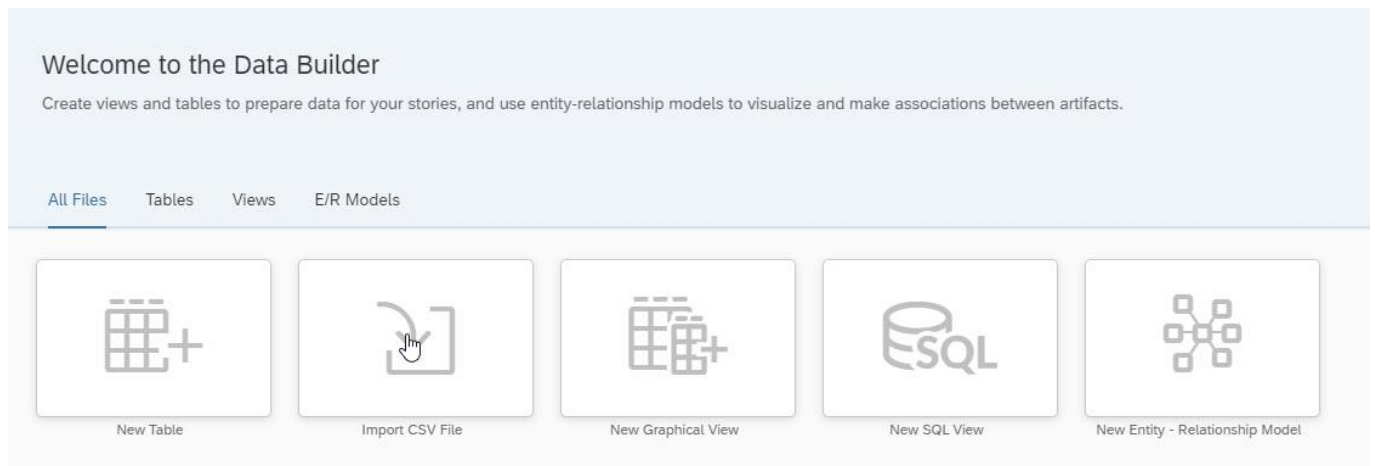
- You can bring in all tables into a single view and establish their relationships, add calculations, filter, and define the final output.
- or
- You can build a set of smaller views focused on a specific part of the overall output – for example Customer or Product – first and then combine those views into the final output.

Following the second approach will over time basically create a set of views focused on dimensions and you can re-use those views across multiple views and it creates less dependency between the final output and the actual physical data model.

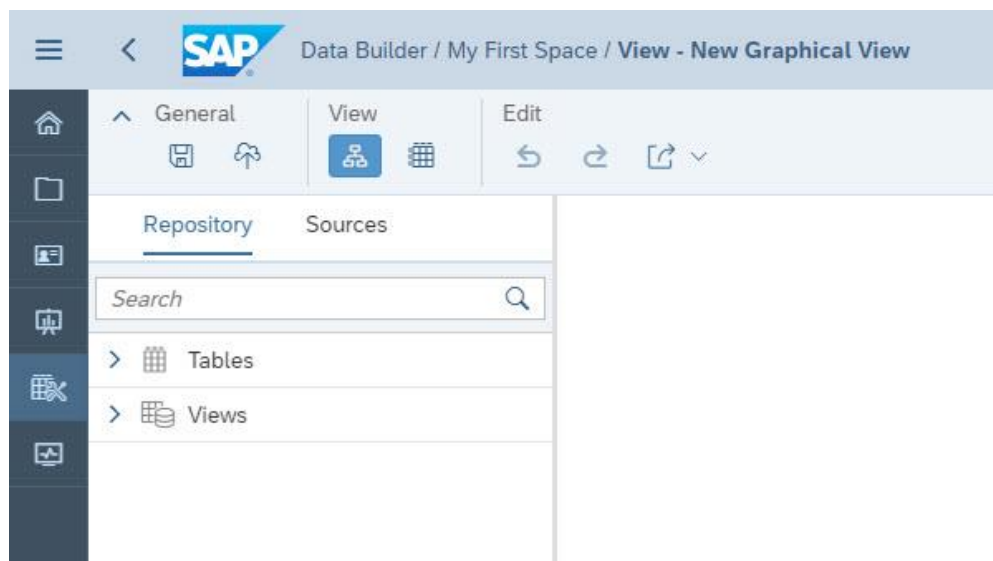
## Session ID: ANA361 - Integrating SAP Analytics Cloud and SAP Data Warehouse Cloud

In the next steps we will create the new View based on our previously created tables

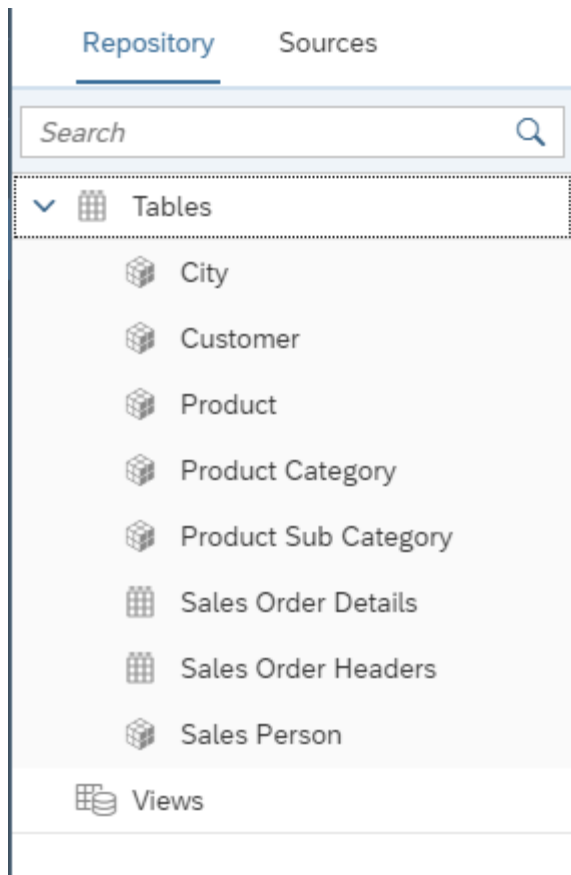
1. Log On to your SAP Data Warehouse Cloud tenant.
2. Select the menu option Data Builder on the left-hand side.



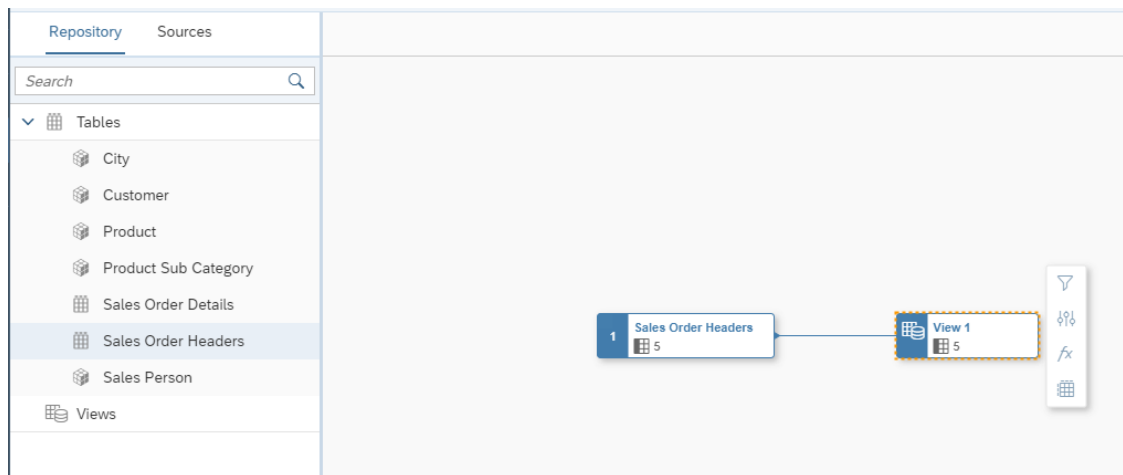
3. Click New Graphical View.



4. On the left-hand side you can decide between:
  - Repository: Here you have access to the local tables (imported data) and Views.
  - Sources: Here you have access to your connections and the remote tables.
5. Ensure you select the option Repository.
6. Open the list of Tables.

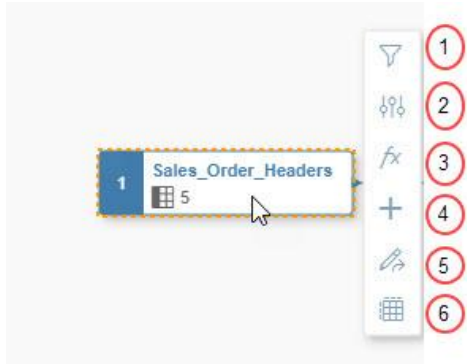


7. You are presented with the list of tables, which we created previously.
8. Drag and Drop the table Sales Order Headers to the canvas.



9. You automatically will – in addition to the table you dragged to the canvas – receive the output view as well, in our example called View 1.

10. Now click on the Sales Order Headers table on the canvas.



11. When you select the table on the canvas, you have the following options:

- (1) You can add Filter on top of the Table.
- (2) You can Rename or Hide columns.
- (3) You can add Calculated Columns.
- (4) You can add additional tables / views based on suggested joins, which are based on your Entity Relationship model.
- (5) You can open the table in the editor.
- (6) You can preview the data.

12. Now click on the output view that was added, in our example View 1.

13. Ensure the option Details (top right corner) is enabled.

14. Navigate to the Properties window.

Properties

View 1

Business Name:  
View 1

Technical Name:  
View\_1

Type:  
Relational Dataset

Allow Consumption:  
☐ OFF

Deployment Status:  
Not Deployed

Search

Columns (5)

22 Sales\_Order\_Number

AA Customer\_ID

Order\_Date

Delivery\_Date

1.23 Discount

Associations (0)

No data

Business Purpose

15. Here you can configure Properties for the final output:

- You can configure the Business Name as well as the Technical Name.
- You can configure the Type of Dataset
- You can decide, if the View can be consumed or not.
- You can choose which of the available Columns are shown or will be hidden.
- You can define additional Associations.
- You can provide details on the Business Purpose, which then will be available as part of the Business Catalog.

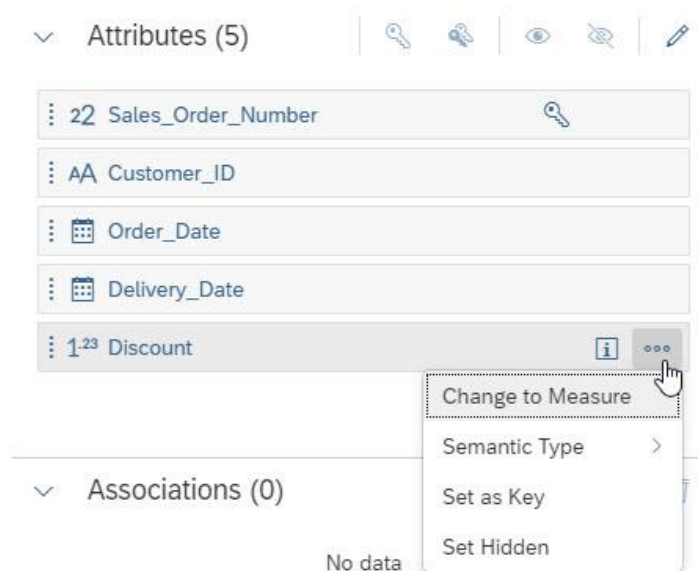
16. Enter Sales View as Business Name.

17. The Technical Name will be generated based on the Business Name, but you can also change it.

18. Set the Type to Analytical Dataset option.

19. Enable the option Exposing.

20. Now open the context menu for the Attribute Discount.



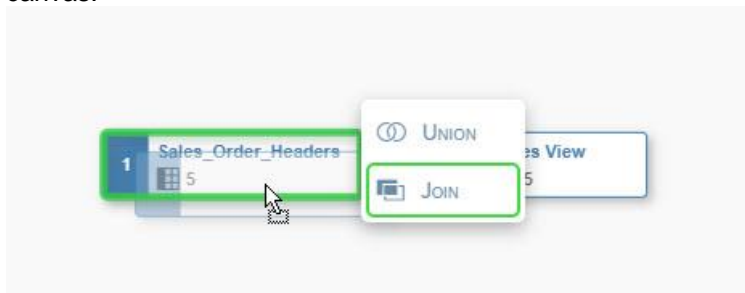
21. Select the option Change to Measure.

#### Average Aggregation

Please note, Discount is a measure that has to be aggregated using an Average and we will do so as part of our SAP Analytics Cloud story.

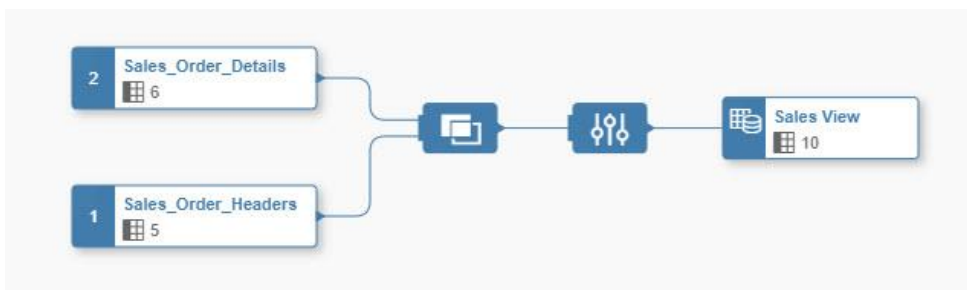


22. Now drag and drop the table Sales Order Details directly on top of the table Sales Order Headers on the canvas.



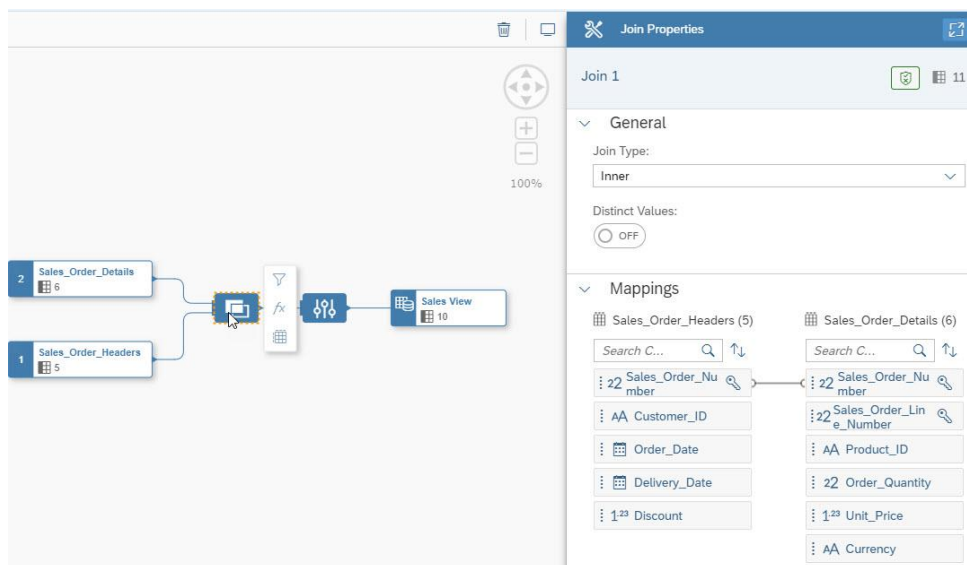
23. You are given the option to decide between a Union or a Join (keep the mouse button pressed).

24. Select the option Join.



25. Both tables have been joined based on the previously defined Entity Relationship Model.

26. Click on the first symbol next to the two tables for the Join Properties.



27. Here you can define which columns are being used for the join and you can define the Join Type.

28. Possible Join Types are:

- Inner Join: The Inner Join will select the set of records that match in both tables.
- Left Join: The Left join selects the complete set of records from the first table with the matching records from the second table.
- Right Join: The Right join selects the complete set of records from the second table with the matching records from the first table.
- Full Join: The Full join combines the results of the Left Join and Right Join and returns all rows from the tables on both sides of the join.
- Cross Join: The Cross join will create every possible combination (cartesian product) of rows from the first and second table. The joined table will contain a row of all columns in table 1 followed by all columns in table 2.
- Natural Join: The Natural join will create an implicit join based on the common columns in the two tables.

29. You also have the option to add additional Filter and add Calculated Columns.

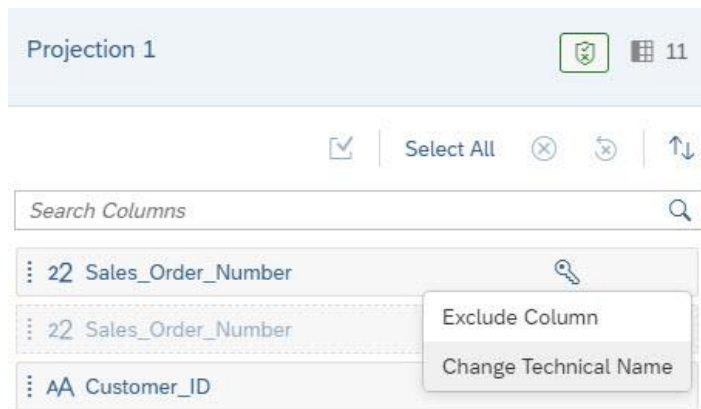
30. Set the Join Type to Inner.

31. Click on the second symbol for the Projection Properties.

The screenshot displays the SAP Analytics Cloud interface. On the left, a data model diagram shows two tables, 'Sales\_Order\_Details' (labeled 2) and 'Sales\_Order\_Headers' (labeled 1), connected by a join symbol. This is followed by a filter symbol and a calculated column symbol (fx), leading to a 'Sales View' (labeled 10). The right side of the interface features the 'Projection Properties' panel for 'Projection 1'. This panel includes a search bar labeled 'Search Columns' and a list of columns with their respective data types and key indicators:

Column Name	Data Type	Key Indicator
Sales_Order_Number	22	Key
Sales_Order_Number	22	Key
Customer_ID	AA	
Sales_Order_Line_Number	22	Key
Order_Date		
Product_ID	AA	
Delivery_Date		
Order_Quantity	22	
Discount	1 <sup>23</sup>	
Unit_Price	1 <sup>23</sup>	
Currency	AA	

32. In the Projection Properties you have the ability to Hide and Rename columns.



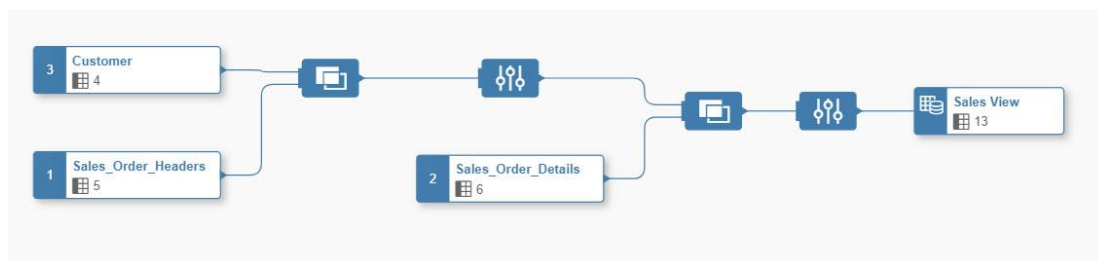
33. We will now add all the other Tables to our view.

34. Drag and Drop the table Customer on top of the table Sales Order Headers.

### Joining Additional Tables

Please note, that you can add additional tables on top of other tables, or you can add additional tables to the projections of previously added tables.

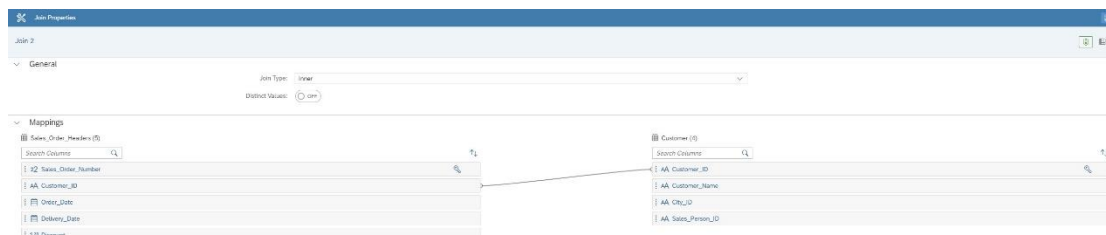
35. The table will then be joined with the Sales Order Headers table.



36. Click on the icon for the Join properties between table Customer and table Sales Order Headers.

37. Ensure the join type is set to Inner.

38. Ensure the join is based on the Customer ID column.



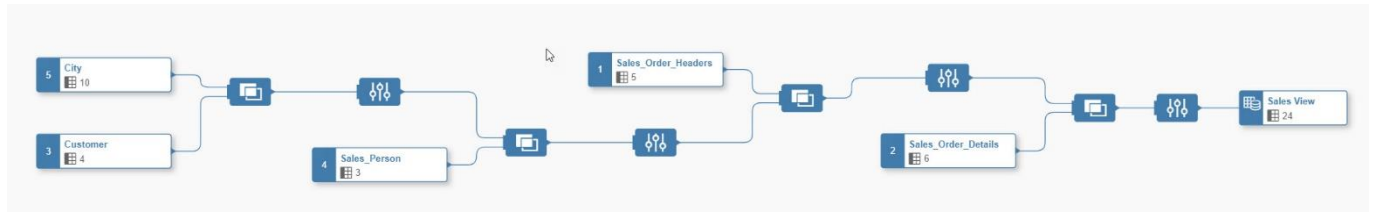
39. Now drag and drop the table Sales Person on top of table Customer.

40. Click on the icon for the Join properties between table Customer and table Sales Person.

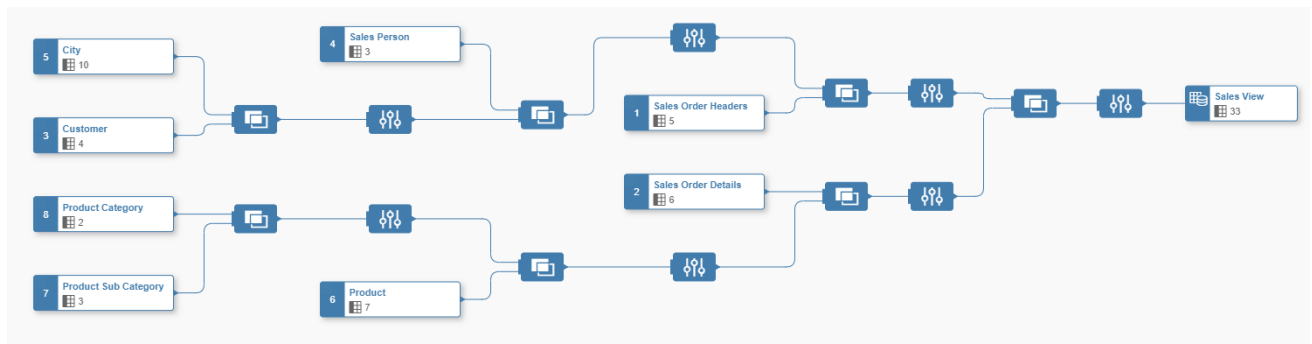
41. Ensure the join type is set to Inner.

## Session ID: ANA361 - Integrating SAP Analytics Cloud and SAP Data Warehouse Cloud

42. Ensure the join is based on the Sales Person ID column.
43. Now drag and drop the table City on top of table Customer.
44. Click on the icon for the Join properties between table Customer and table City.
45. Ensure the join type is set to Inner.
46. Ensure the join is based on the City ID column.
47. Your View should look similar to the figure shown below.



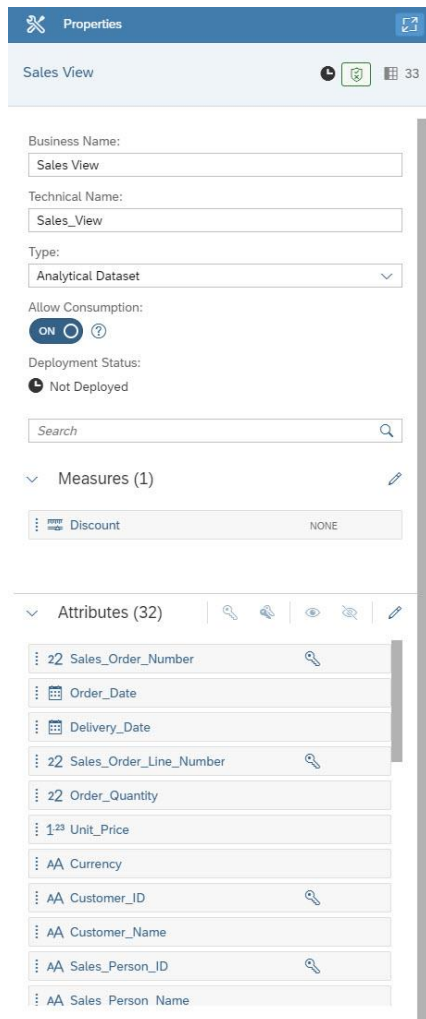
48. At this point we did add all the customer relevant tables and we can now add the product relevant information.
49. Now drag and drop the table Product on top of table Sales Order Details.
50. Click on the icon for the Join properties between table Sales Order Details and table Product.
51. Ensure the join type is set to Inner.
52. Ensure the join is based on the Product ID column.
53. Now drag and drop the table Product Sub Category on top of table Product.
54. Click on the icon for the Join properties between table Product Sub Category and table Product.
55. Ensure the join type is set to Inner.
56. Ensure the join is based on the Product Sub Category ID column.
57. Now drag and drop the table Product Category on top of Product Sub Category.
58. Click on the icon for the Join properties between table Product Sub Category and table Product Category.
59. Ensure the join type is set to Inner.
60. Ensure the join is based on the Product Category ID column.
61. Your overall View should look similar like the figure shown below.



## Joining Tables

Please note, that during this example we are not focused on ensuring that the tables are joined in the best possible scenario in regards to performance and data volume.

62. Now click on the overall output projection, in our example Sales View.
63. Ensure the Details option (top right corner) is enabled.
64. Navigate to the Properties of the Sales View.



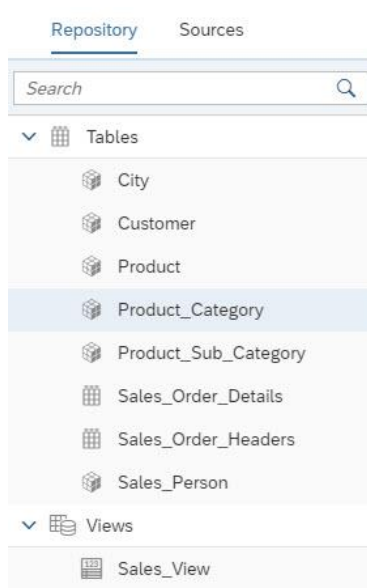
65. Our overall Sales View now contains all the columns from the joined tables as we have not removed or hidden any of them. In addition, we also will have to configure, which columns should be treated as Measures.
66. In the list of Attributes, select the following columns and use the context menu to change them to Measures:
  - Order Quantity
  - Unit Price
  - Product List Price
  - Product Cost

67. Regarding the list of columns available in the Attributes, we will leave all of them visible.
68. Save your View.
69. You will be asked to confirm the Business Name and Technical Name.
70. Click Save.
71. Deploy your View.

### 3.7 SAP Data Warehouse Cloud – Exercise 07: Adding Hierarchy to the View

At this point we created a View in SAP Data Warehouse Cloud, which combines all the tables that we created previously, and we could now go into SAP Analytics Cloud and start consuming the information. What we have to do, before going into SAP Analytics Cloud, is to add an additional View with an Association to – in our example – the table of the Sales Person, so that the hierarchy that we created previously is being added to the View and will be available within SAP Analytics Cloud.

1. Navigate to the Data Builder.
2. Select the option Views.
3. Click the option New Graphical View.
4. On the left-hand side open the list of Views from the Repository.



5. Drag and drop the previously created Sales View to the canvas.



6. Click on the output projection, in our example View 1.
7. Open the Properties.

8. Scroll down to the area Associations.

Properties

View 1

Business Name:  
View 1

Technical Name:  
View\_1

Type:  
Analytical Dataset

Allow Consumption:  
OFF

Deployment Status:  
Not Deployed

Search

Measures (5)

Attributes (28)

Associations (0)

No data

Business Purpose

9. Use the “+” sign to create a new Association.

10. You are being presented with the complete list of Tables and Views.

Select Association Target

Business Name	Technical Name	Type	Created By
Sales_Order_Details	Sales_Order_Details	Local Table	IHLGEFORT
Product	Product	Dimension (Table)	IHLGEFORT
Product_Sub_Category	Product_Sub_Category	Dimension (Table)	IHLGEFORT
Product_Category	Product_Category	Dimension (Table)	IHLGEFORT
Customer	Customer	Dimension (Table)	IHLGEFORT
City	City	Dimension (Table)	IHLGEFORT
Sales_Order_Headers	Sales_Order_Headers	Local Table	IHLGEFORT
Sales_Person	Sales_Person	Dimension (Table)	IHLGEFORT
Sales_View	Sales_View	Analytical Dataset (View)	IHLGEFORT

11. Select the previously created table Sales Person.



12. Click OK.

View 1 to Sales Person

View 1 / View 1 to Sales Person

General

Business Name:

View 1 to Sales Person

Technical Name:

\_Sales\_Person

From:

View 1

To:

Sales Person

Join

View 1 (33)

Search C...

22 Sales\_Order\_...

Order\_Date

Delivery\_Date

1<sup>23</sup> Discount

22 Sales\_Order\_...

22 Order\_Quantity

1<sup>23</sup> Unit\_Price

AA Currency

AA Customer\_ID

AA Customer\_Name

AA City\_ID

AA City\_Name

AA Country

Sales Person (3)

Search C...

AA Sales\_Person...

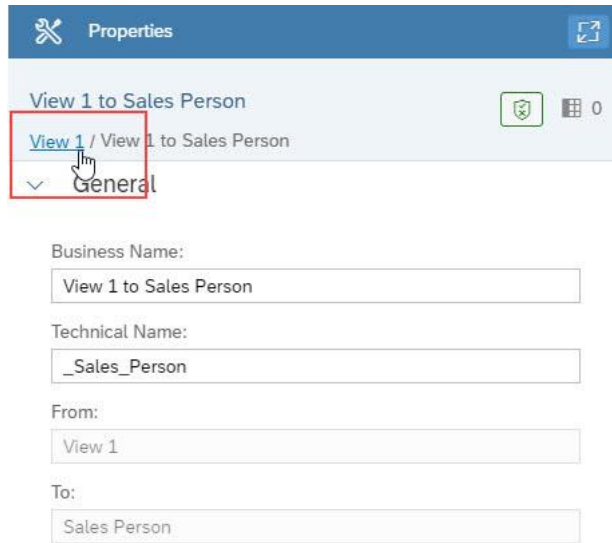
AA Sales\_Person\_Name

AA Sales\_Manager

13. You are being presented with the details on the Join between the Sales View and the table Sales Person.

14. Ensure the Join is defined based on the Sales Person ID.

15. After you defined the join, click on the name of the new View in the top left of the Properties to navigate back to the Properties of the View.



16. Enter ANA361\_XX Analytics View as Business Name and replace XX with your User Number.
17. Enter ANA361\_XX\_Analytics\_View as Technical Name and replace XX with your User Number.
18. Ensure the option Exposing is enabled.
19. Save the View
20. Deploy the View.

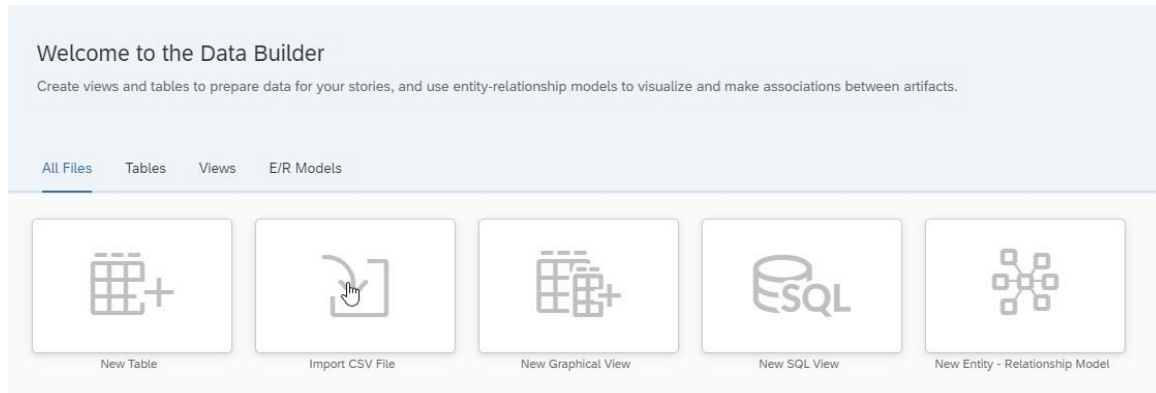
Now we created two Views based on the set of tables we created previously and by creating the second View we also made sure that the previously created hierarchy will be available when we consume the View in SAP Analytics Cloud.

### 3.8 SAP Data Warehouse Cloud – Exercise 08: Configuring Label Columns

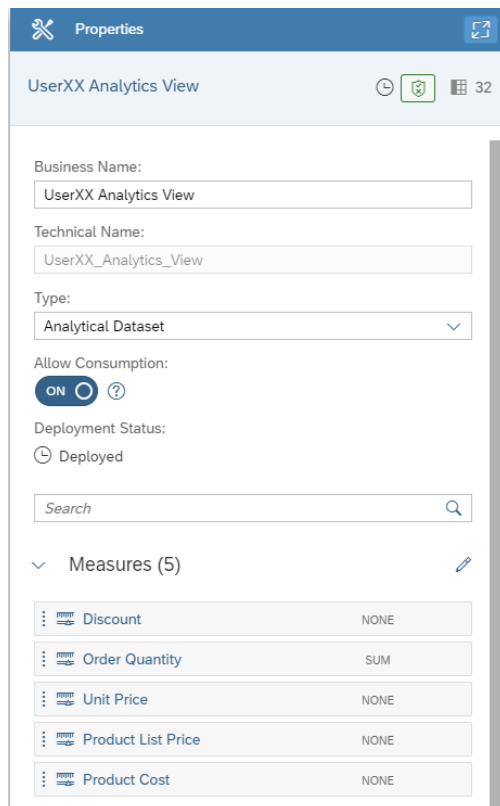
In the previous exercise we created the tables, the Entity Relationship model, and the views. Part of our view are currently columns which represent an ID and a description, for example for a Customer ID and a Customer Description. In this exercise we will configure those as Label Columns, so that in SAP Analytics Cloud the user can switch between the ID and Description.

In the next steps we will configure the Label Columns in our previously created View.

1. Log On to your SAP Data Warehouse Cloud tenant.
2. Select the menu option Data Builder on the left-hand side.



3. Open the View that we created in the previous exercise – in our example the ANA361\_XX Analytics View.
4. Select the overall output projection, in our example called ANA361\_XX Analytics View.
5. Open the Properties.



6. Click on the pencil symbol for the Attributes to open up the edit mode.

Attributes (27)

UserXX Analytics View

<input type="checkbox"/>	Business Name	Technical Name	Data Type	Semantic Type	Label Column	<input type="checkbox"/>
<input checked="" type="checkbox"/>	Sales Order Number	Sales_Order_Number	Integer64	None		<input checked="" type="checkbox"/>
<input type="checkbox"/>	Order Date	Order_Date	Date	None		<input checked="" type="checkbox"/>
<input type="checkbox"/>	Delivery Date	Delivery_Date	Date	None		<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Sales Order Line Number	Sales_Order_Line_Number	Integer64	None		<input checked="" type="checkbox"/>
<input type="checkbox"/>	Currency	Currency	String(3)	None		<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Customer ID	Customer_ID	String(20)	None		<input checked="" type="checkbox"/>
<input type="checkbox"/>	Customer Name	Customer_Name	String(30)	Text		<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	Sales Person ID	Sales_Person_ID	String(10)	None		<input checked="" type="checkbox"/>
<input type="checkbox"/>	Sales Person Name	Sales_Person_Name	String(30)	Text		<input checked="" type="checkbox"/>
<input type="checkbox"/>	Sales Manager	Sales_Manager	String(10)	None		<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	City ID	City_ID	String(10)	None		<input checked="" type="checkbox"/>
<input type="checkbox"/>	City Name	City_Name	String(30)	Text		<input checked="" type="checkbox"/>
<input type="checkbox"/>	Country	Country	String(30)	Geolocation - Normalized Name		<input checked="" type="checkbox"/>
<input type="checkbox"/>	Country ISO2	Country_ISO2	String(2)	None		<input checked="" type="checkbox"/>
<input type="checkbox"/>	Country ISO3	Country_ISO3	String(3)	None		<input checked="" type="checkbox"/>

Close

7. Now make the following changes:

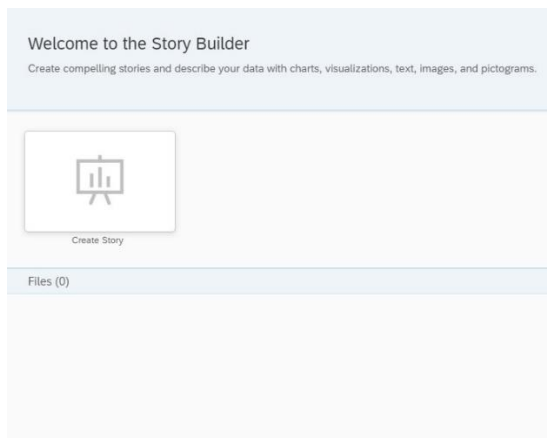
- Configure the Label Column for Customer ID with the value for Customer Name.
- Uncheck the Visible option (far right column) for the Customer Name.
- Configure the Label Column for City ID with the value for City Name.
- Uncheck the Visible option (far right column) for the City Name.
- Uncheck the Visible option (far right column) for the Country ISO2.
- Uncheck the Visible option (far right column) for the Country ISO3.
- Uncheck the Visible option (far right column) for the Latitude.
- Uncheck the Visible option (far right column) for the Longitude.
- Configure the Label Column for Sales Person ID with the value for Sales Person Name.
- Uncheck the Visible option (far right column) for the Sales Person Name.
- Configure the Label Column for Product ID with the value for Product Description (Short).
- Uncheck the Visible option (far right column) for the Product Description (Short).
- Configure the Label Column for Product Sub Category ID with the value for Product Sub Category Desc
- Uncheck the Visible option (far right column) for the Product Sub Category Desc
- Configure the Label Column for Product Category ID with the value for Product Category Description
- Uncheck the Visible option (far right column) for the Product Category Description

8. Click Close.

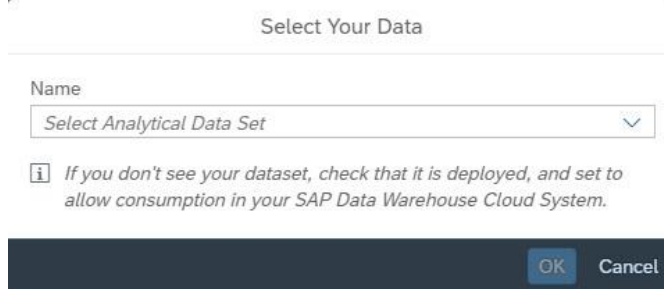
9. Save your changes.

10. Deploy your changes.

11. Open the Story Builder.



12. Click Create Story.



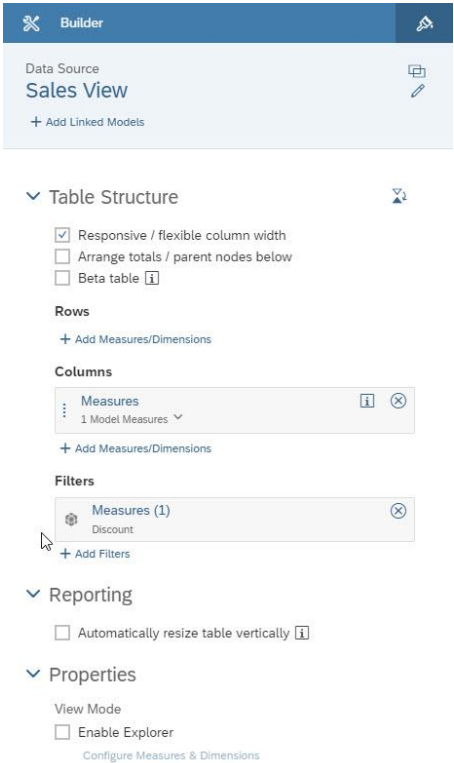
13. Open the list of available Analytical Data Sets and select the ANA361\_XX Analytics View.

14. Click OK.

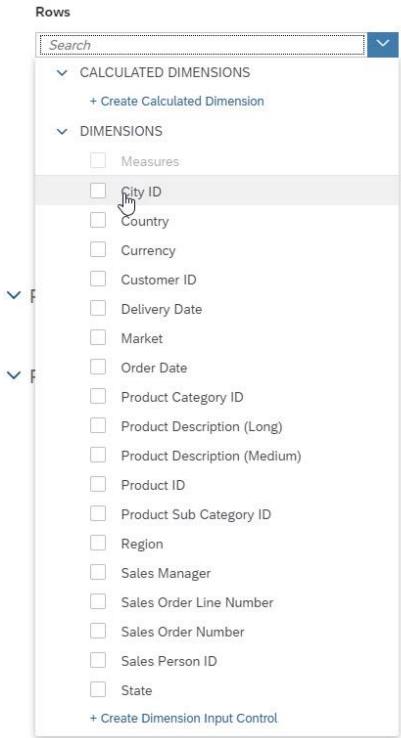


15. Click on the symbol for a new Table.

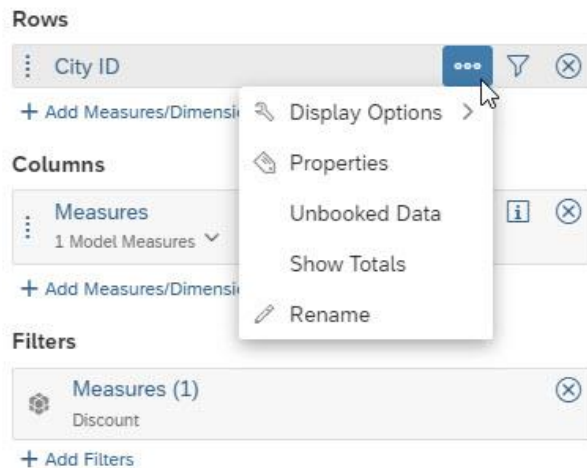
16. On the right-hand side navigate to the Builder panel.



17. Click Add Measures / Dimensions in the Rows area.



18. Select dimension City ID.
19. Now open the menu for dimension City ID in the Rows area.



20. Select the menu Display Options.
21. You should now have the option to choose between ID, Description, or ID & Description and the table in your Story should change the display the dimension members for dimension City ID accordingly.

City ID
CITY001 Tokyo
CITY002 Mexico City
CITY003 Paris
CITY004 London
CITY005 Washington
CITY006 Singapore
CITY007 Berlin
CITY008 Cape Town
CITY009 Lisbon
CITY010 Vienna
CITY011 Rabat
CITY012 The Hague
CITY013 Pretoria
CITY014 Ottawa
CITY015 Copenhagen
CITY016 Amsterdam
CITY017 Oslo
CITY018 Bloemfontein
CITY019 Canberra
CITY020 Bern

22. Navigate back to the Home screen of SAP Data Warehouse Cloud. In case you are being asked, if you would like to save the SAP Analytics Cloud story, select the option "Leave". We will create an SAP Analytics Cloud story in an exercise later on.

### 3.9 SAP Data Warehouse Cloud – Exercise 09: Adding Calculated Columns to the View

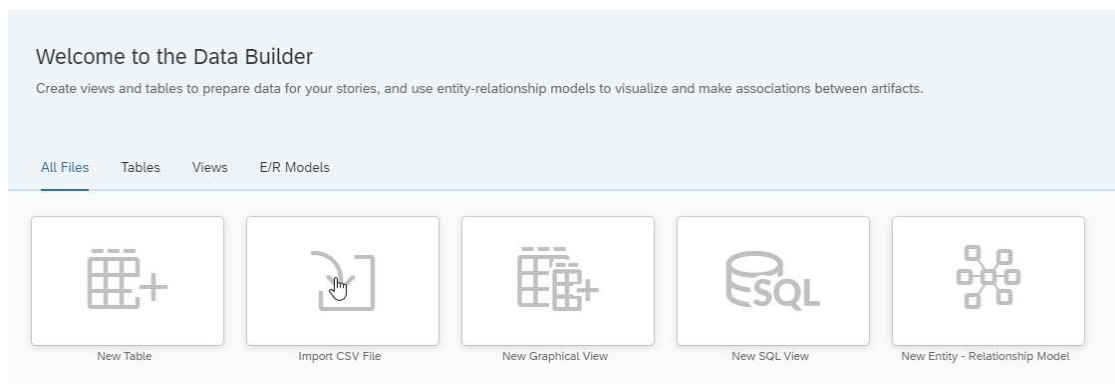
In the previous exercises we created the tables and the view, and we also configured the hierarchy as well as the label columns as part of the view. In this exercise we will add calculated columns as part of the View, so that we then can leverage those calculations as part of SAP Analytics Cloud.

We will add Calculated Columns to create the following values:

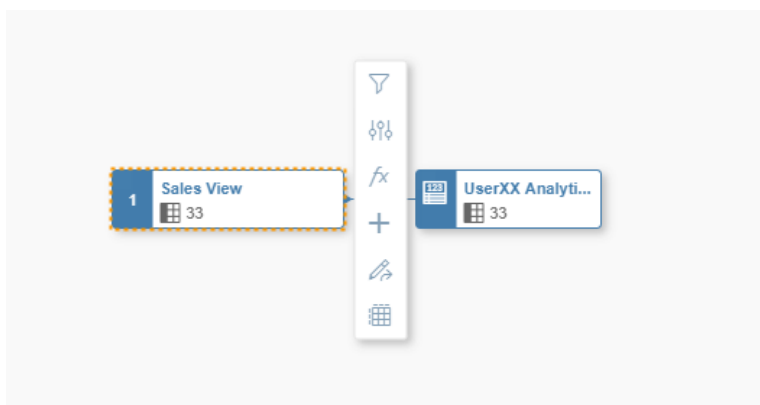
- Number of Days to deliver the product
- Profit per Product / Order Line Item as an absolute value
- Profit per Product / Order Line Item as a percentage value
- Profit per Product / Order Line Item as an absolute value with considering the Order discount
- Profit per Product / Order Line Item as a percentage value with considering the Order discount

In the next steps we will configure the Label Columns in our previously created View.

1. Log On to your SAP Data Warehouse Cloud tenant.
2. Select the menu option Data Builder on the left-hand side.

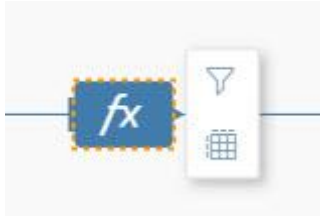


3. Open the View that we created in the previous exercise – in our example ANA361\_XX Analytics View.
4. Select the node for the Sales View.

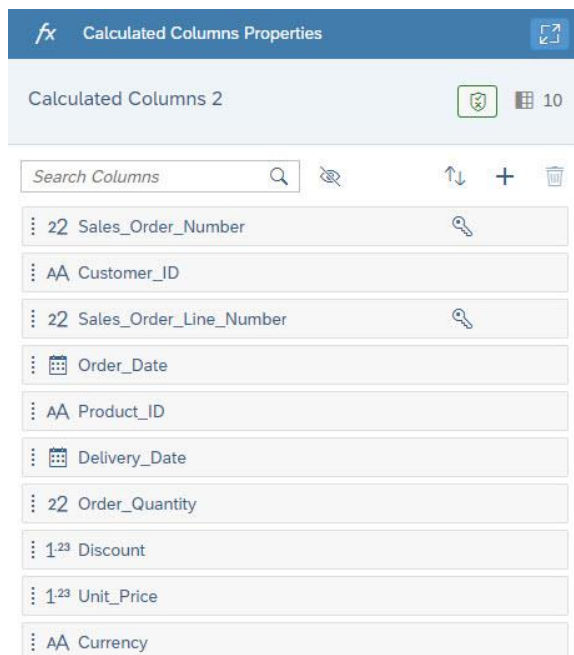




5. Click on the “fx” symbol to add the Calculated Column option.
6. You will receive an additional element as part of the view.



7. Click on the new element on the canvas for the Calculated Columns.
8. Ensure the Details option is enabled.
9. Navigate to the Calculated Column Properties.



10. Use the “+” icon in the top right corner to add a new Calculated Column.

Calculated Columns Properties

Calculated\_1

Technical Name: Calculated\_1

Data Type: AA String(len)

Length: 10

Expression

The expression of calculated column 'Calculated\_1' is invalid.

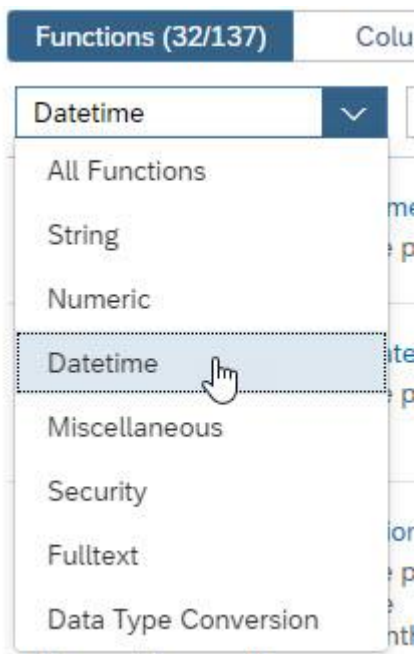
Functions (32/137) Columns (11) Operators

Search

22	Sales_Order_Number	Sales Order Number
AA	Customer_ID	Customer ID
22	Sales_Order_Line_Number	Sales Order Line Number
	Order_Date	Order Date
AA	Product_ID	Product ID
	Delivery_Date	

11. You are being presented with the option to use from a list of Functions and to add a new Calculated Column.
12. Enter Delivery Time in Days as Business Name.
13. Enter Delivery\_Time\_in\_Days as Technical Name.
14. Click on the tab Functions.

15. Select the category Datetime.



16. Scroll down to the function DAYS\_BETWEEN.

17. Click on the function entry (single click). The Function should now appear in the Expression box.

18. Now click on the tab Columns.

19. Place the cursor into the function on the left side of the comma.

20. Click on the column Order\_Date.

21. Place the cursor into the function on the right side of the comma.

22. Click on the column Delivery\_Date.

23. Your formula should look like this:

```
DAYS_BETWEEN(Order_Date,Delivery_Date)
```

24. Click Validate (above the Expression box).

25. Ensure the Data Type of the formula is set to Integer.

26. Click on the Columns option (top left of the Properties) to navigate back to the list of columns.



27. Navigate to the Calculated Columns node as part of your View.



28. Click on the Preview option.

ist Price	Product Cost	Product Sub Catego...	Product Sub Catego...	Product Category ID	Delivery Time in Days
	12.58	SUBCATEGORY002	-	PRODUCTCATEGOR...	17
	6.02	SUBCATEGORY002	-	PRODUCTCATEGOR...	17
	10.62	SUBCATEGORY002	-	PRODUCTCATEGOR...	17
	3.07	SUBCATEGORY003	-	PRODUCTCATEGOR...	17
	3.99	SUBCATEGORY003	-	PRODUCTCATEGOR...	17
	6.42	SUBCATEGORY001	-	PRODUCTCATEGOR...	18
	3.49	SUBCATEGORY001	-	PRODUCTCATEGOR...	18
	15.3	SUBCATEGORY001	-	PRODUCTCATEGOR...	18
	12.58	SUBCATEGORY002	-	PRODUCTCATEGOR...	18

29. You should see your newly created Calculated Column to the far right of the Preview.

30. Click on the element for the Calculated Columns on the canvas – the “fx” icon.

31. Ensure the Details option is enabled.
32. Navigate to the Calculated Column Properties.
33. Click on the “+” icon to add a new Calculated Column.
34. Enter Profit per Line Item without Discount as Business Name.
35. Enter Profit\_per\_LItem\_no\_Disc as Technical Name.
36. Enter the following formula:

$$\text{Order\_Quantity} * (\text{Unit\_Price} - \text{Product\_Cost} )$$

We are using the Unit Price per Product, subtract the Product Cost, and multiply it with the Order Quantity per Line item of the orders we have.

37. Click Validate.
38. Ensure the Data Type is set to Decimal with a Precision set to 6 and a Scale set to 2.
39. Click on the Columns option (top left of the Properties).
40. Click on the “+” icon to add a new Calculated Column.
41. Enter Profit % per Line Item without Discount as Business Name.
42. Enter PercProfit\_per\_LItem\_no\_Disc as Technical Name.
43. Enter the following formula:

$$(\text{Unit\_Price} - \text{Product\_Cost}) / \text{Unit\_Price}$$

44. Click Validate.
45. Ensure the Data Type is set to Decimal with a Precision set to 6 and a Scale set to 2.
46. Click on the Columns option (top left of the Properties).
47. Click on the “+” icon to add a new Calculated Column.
48. Enter Profit per Line Item with Discount as Business Name.
49. Enter Profit\_per\_LItem\_w\_Disc as Technical Name.
50. Enter the following formula:

$$\begin{aligned} &\text{Order\_Quantity} \\ &* ( \\ &(\text{Unit\_Price} * (1 - \text{Discount})) \\ &- \text{Product\_Cost} ) \end{aligned}$$

51. Click Validate.
52. Ensure the Data Type is set to Decimal with a Precision set to 6 and a Scale set to 2.
53. Click on the Columns option (top left of the Properties).
54. Click on the “+” icon to add a new Calculated Column.
55. Enter Profit % per Line Item with Discount as Business Name.

56. Enter PercProfit\_per\_LItem\_w\_Disc as Technical Name.

57. Enter the following formula:

```
(  
  (Unit_Price * (1-Discout))  
  - Product_Cost) / Unit_Price
```

58. Click Validate.

59. Ensure the Data Type is set to Decimal with a Precision set to 6 and a Scale set to 2.

60. Click on the Columns option (top left of the Properties).

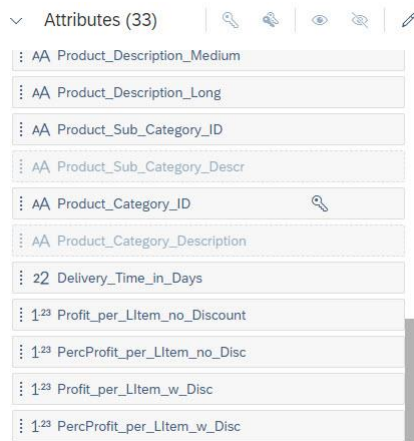
61. Click on the Preview option for the Calculated Column element as part of your view.

62. You should then see the Preview of your calculations.

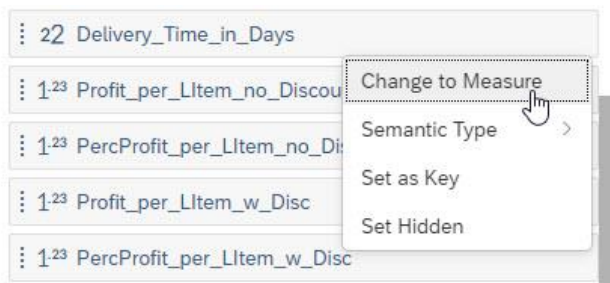
63. Now select the overall output element ANA361\_XX Analytics View.

64. Ensure the Properties are shown.

65. Scroll down in the list of Attributes.



66. Select each of the newly created Calculated Columns (they should be at the bottom of the list) and open the context menu and select the option Change to Measure.



67. Save your changes.

68. Deploy the changes to the Sales View.

We basically added a set of Calculated Columns to the Sales View and we will use those calculations when we create the story with the Story Builder.

## 4 SAP ANALYTICS CLOUD EXERCISES

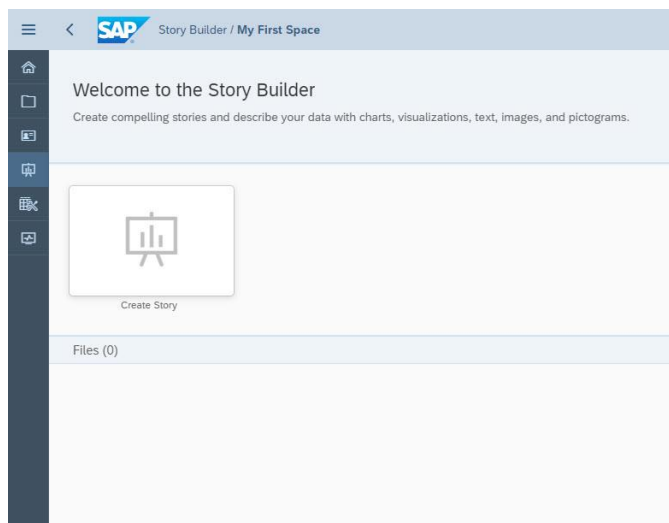
In the previous exercises we created the tables and the view, and we also configured the hierarchy as well as the label columns, and we reviewed all assets we created in the Business Catalog. In this exercise we will use the integration of SAP Analytics Cloud with SAP Data Warehouse Cloud to answer the following questions based on our data:

- What are the total sales by Customer / by Product / by Sales Person?
- Which product is receiving the highest discounts?
- Which Sales Person is giving the most discount?
- Which product is most profitable?
- Who is the most successful sales person / sales manager?

### 4.1 SAP Analytics Cloud - Exercise 01: Total Sales by Customer / Product / Sales Person

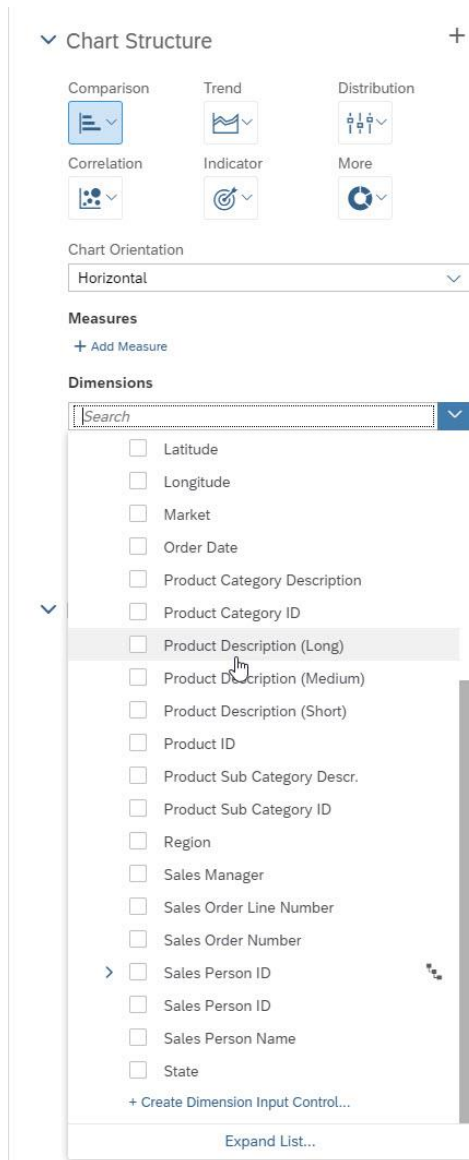
In the previous exercises we create the Sales View, which will be our source for the Story that we are going to create.

1. Log On to your SAP Data Warehouse Cloud tenant.
2. Select the menu option Story Builder on the left-hand side.
3. In case you are being asked, select the Space you created in the previous exercise.



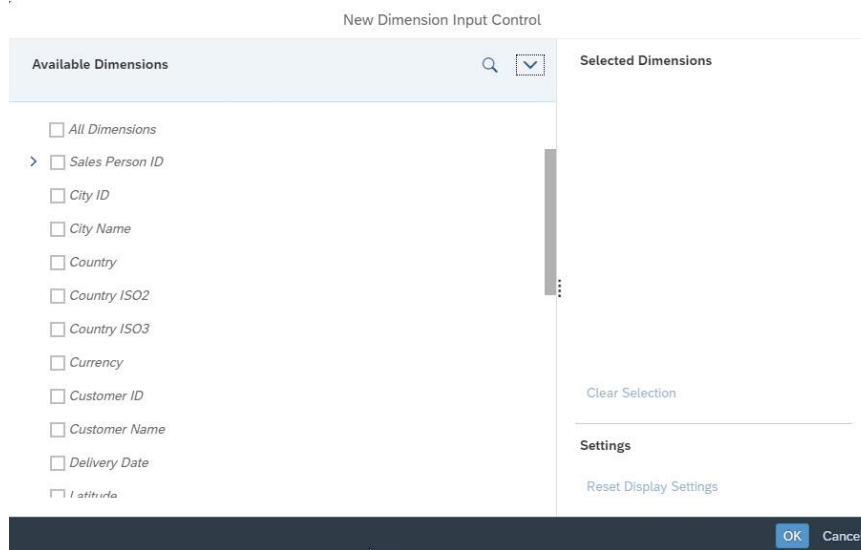
4. Click Create Story.
5. When being asked to select the data source, select the ANA361\_XX Analytics View.
6. Click OK.
7. Select the option to add a Chart.
8. In our example, we want to know the Total Sales Revenue by Customer and by Product and by Sales Person. So instead of creating three different charts, with the dimension that is used for aggregation being the only difference, we will setup a Dimension Input Control for the chart, so that the user would be able to choose.

9. Navigate to the Builder Panel on the right-hand side.
10. Click on Add Dimension for the Dimensions area and scroll down the list.





11. Select the option Create Dimension Input Control.



12. Select the dimensions: Customer ID, Product ID, and Sales Person ID.

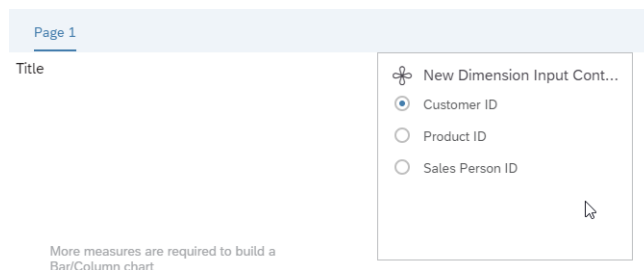
13. Click OK.

14. You will receive a new token as part of your page, and you can use the controls on the box to resize the Dimension Input Control.



More measures are required to build a Bar/Column chart

15. Resize the new Dimension Input Control, so that you can see all three options.



More measures are required to build a Bar/Column chart

16. Select the chart on your canvas.
17. Navigate to the Builder Panel on the right-hand side.
18. Click Add Measures for the Measures option.

In our example we don't have a measure for the Revenue, but we do have the Order Quantity, Unit Price, and the Discount – so we can create a calculation.

19. Select the option Create Calculation.
20. As Type select the option Calculated Measure.
21. For the Name enter Revenue (with Discount).
22. For the Formula enter the following:

```
(["ANA361_XX Analytics View":Order_Quantity] * ["ANA361_XX Analytics View":Unit_Price])  
*  
( 1 - ["ANA361_XX Analytics View":Discount])
```

You can start typing in the Formula box and the corresponding items will be listed, which you then can choose.

Please note, that the example above uses XX as user number and that in your example the XX should be replaced with your user number.

23. Click OK.

Because this formula calculates the values on a detailed level and because the values only exist on a product dimension level, we also need to create an aggregation, so that we can aggregate the value towards the customer, product, and sales person.

24. Select the Chart.
25. Navigate to the Builder Dimension.
26. Click Add Measure in the Measures option.
27. Click Create Calculation.
28. For the Type select the Aggregation option.

29. Enter Revenue Aggregation as Name.

30. As Operation make sure the option SUM is selected.

31. For the Measure select the previously created calculated measure Revenue (with Discount).

32. For the Aggregation Dimension select the dimension Sales Order Line Number.

33. Click OK.

34. Ensure you select the chart.

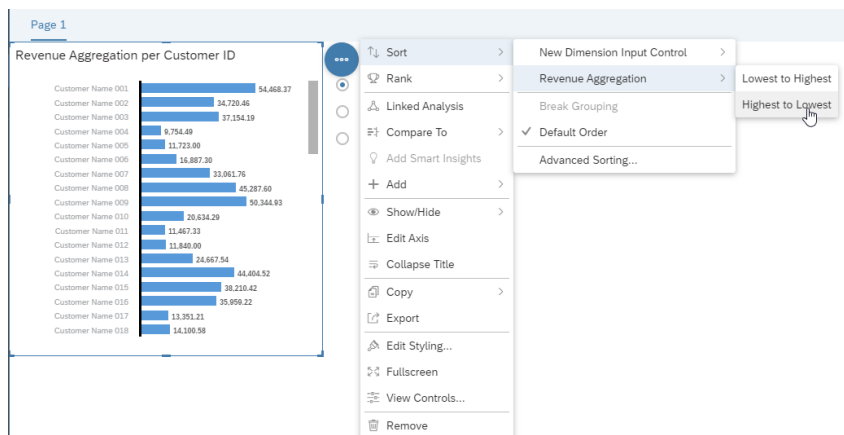
35. Navigate to the Builder Panel.

36. In the Measures area you should now have two measures:

- Revenue (with Discount)
- Revenue Aggregation

37. Remove the measure Revenue (with Discount) from the chart.

38. Now open the context menu of the chart (top right corner, three dots).

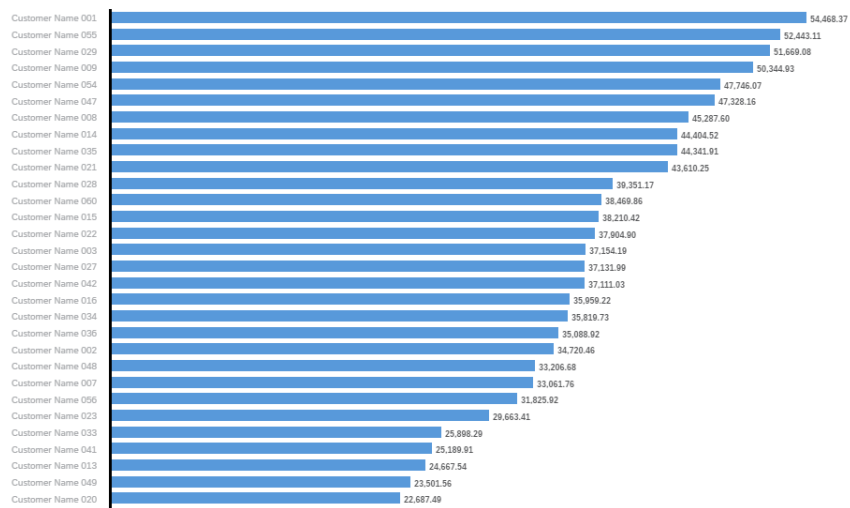


39. Select the menu Sort > Revenue Aggregation > Highest to Lowest.

You should now be able to use the Dimension Input Control and switch between the three dimensions (Customer ID, Product ID, and Sales Person ID) and show the aggregated values for each customer, product, and Sales Person.

## Session ID: ANA361 - Integrating SAP Analytics Cloud and SAP Data Warehouse Cloud

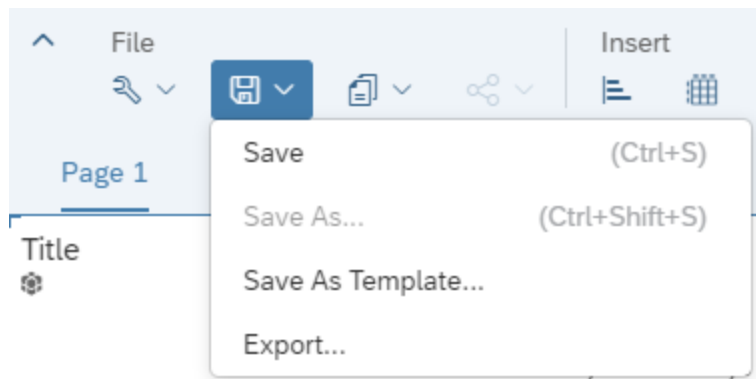
Revenue Aggregation per Customer ID



Please select the Dimension

- ☒ Customer ID
- ☐ Product ID
- ☐ Sales Person ID

40. Select the menu File in the toolbar and use the option to Save your story.



41. Select the option Save.

42. Enter ANA361 XX – Analytics Story as Name.

43. Enter ANA361 XX – Analytics Story as Description

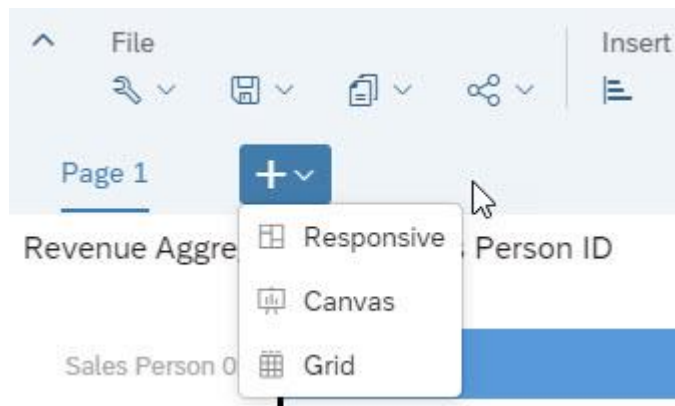
44. Click OK.

## 4.2 SAP Analytics Cloud - Exercise 02: Which Product receives the highest Discount?

In the following steps, we will use the previously created View to find out, which products received the highest discount rate. The steps are written as you would continue with the story from the previous exercise.

In this exercise we want to find out, which product received the highest discount. Based on our data model, the Unit Price and Quantity are part of the Sales Order Details for each product and the Discount rate is part of the Sales Order Headers. We are looking to find out the highest Discount Rate as a percentage value and the highest overall Discount amount as an absolute value. For the absolute value of the Discount value, we first need to calculate the discount value per line item of all the orders and then we need to create the aggregation on top of that calculation. For the discount rate we have to create an average aggregation on top of the discount rate of all the orders.

1. With your existing story open, click on the “+” sign next to the tab for Page 1.



2. Select the option to add a new Canvas Page.
3. Select the option to add a new Chart.
4. The Chart will leverage the ANA361\_XX Analytics View as default data source, based on our previous selection.
5. Select the chart on the canvas.
6. Navigate to the Builder Panel.
7. Click Add Dimension as part of the Dimensions section.
8. Select dimension Product ID.
9. Click on Add Measure as part of the Measures section.
10. Click on the option Create Calculation.

11. As Type for the calculation selection the option Calculated Measure.

Calculation Editor

Type: **Calculated Measure**

Name: **Calculated Measure 1**

Edit Formula

1

Format

Available Objects

Input Controls

+ Create New...

Formula Functions

Functions

- IF()
- ABS()
- LOG()
- LOG10()
- INT()
- FLOAT()
- DOUBLE()

OK Cancel

12. Enter Discount Value (Absolute) per Line item as Name.

13. Enter the following formula:

```
(["ANA361_XX Analytics View":Unit_Price] * ["ANA361_XX Analytics View":Order_Quantity] )  
*  
["ANA361_XX Analytics View":Discount]
```

You can start typing in the Formula Editor and then select the element.

Please note, that the example above uses XX as user number and that in your example the XX should be replaced with your user number.

14. Click OK.

This formula will calculate the absolute discount value on a per line item basis for the orders. Now we need to add an aggregation calculation.

15. Click on Add Measure as part of the Measures section.

16. Click on the option Create Calculation.

17. As Type for the calculation selection the option Aggregation.

The screenshot shows the 'Calculation Editor' dialog box. It has two main sections: 'Type' and 'Name' at the top, and 'Properties' below. In the 'Type' section, 'Aggregation' is selected in a dropdown menu. In the 'Name' section, 'Discount Value Aggregation' is entered in a text field. The 'Properties' section contains three rows: 'Operation' with 'SUM' selected, 'Measure' with 'Revenue (with Discount)' selected, and 'Aggregation Dimensions' with 'City ID' selected. Below these is a '+ Add a Dimension' button and a 'Use conditional aggregation' checkbox which is unchecked. At the bottom right, there are 'OK' and 'Cancel' buttons.

18. Enter Discount Value Aggregation as Name.

19. As Operation select the option SUM.

20. As Measure select the previously created Measure Discount Value (Absolute) per Line Item.

21. As Aggregation Dimension select the dimension Sales Order Line Number.

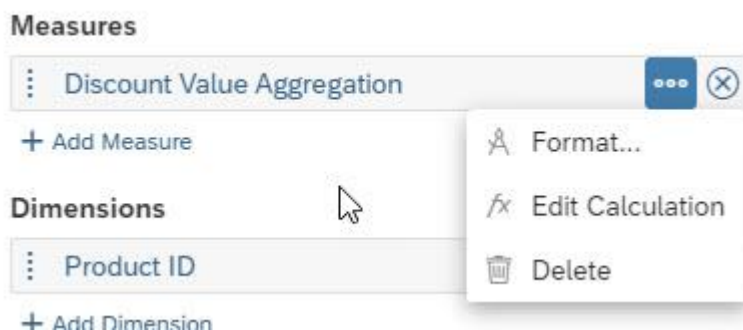
22. Click OK.

23. You should have now two measures as part of the Measures section of the chart.

24. Remove the measure Discount Value (Absolute) per Line Item from the Measures section.

25. Now navigate to the Measures section in the Builder Panel.

26. Open the context menu for the measure.



27. Select the option Format.

28. Set the Scale to Thousand.

29. Set the Scale Formatting to the option "k, m, bn".

30. Set the Decimal Places to 1.

Formatting

\*Scale  
Thousand

\*Scale Formatting  
k, m, bn

\*Decimal Places  
1

\*Show Sign As  
Default

Unit

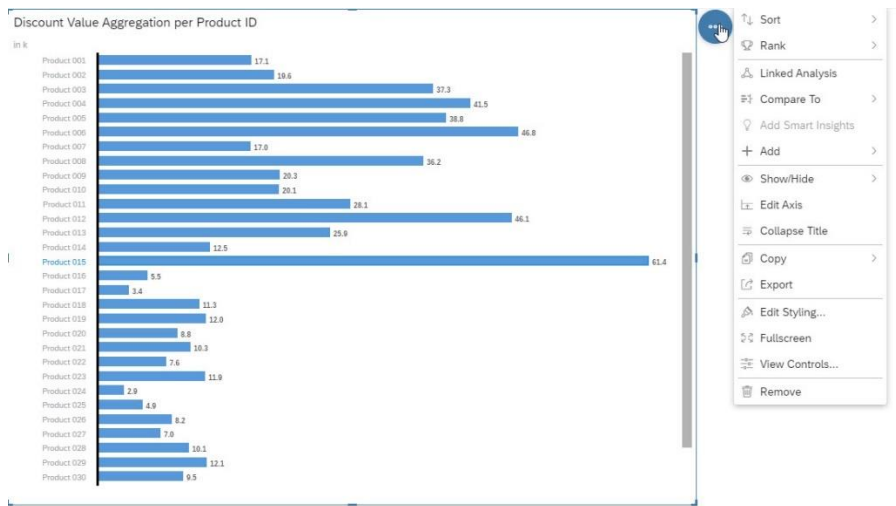
☒ Use unit of underlying measures

OK Cancel

31. Click OK.

32. Select the chart on the canvas.

33. Open the context menu in the top right corner.

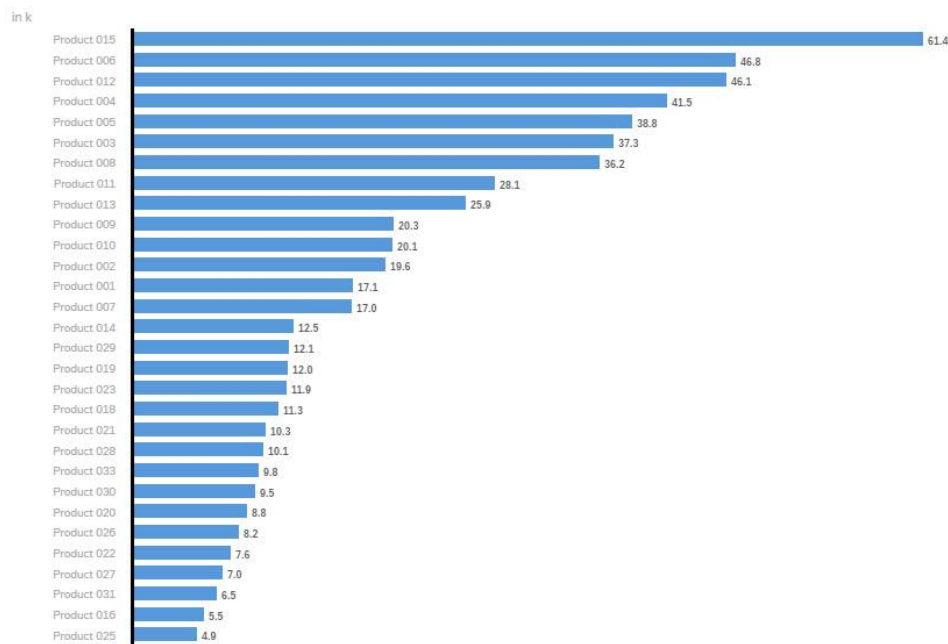


34. Select the menu Sort > Discount Value Aggregation > Highest to Lowest.



35. So we can clearly see that Product 015 has the highest absolute discount value.

Discount Value Aggregation per Product ID



36. As next step we need to find out, which product has the highest average discount rate in %.

37. In the toolbar, select the option to insert a new chart.



38. Select the new chart on the canvas.

39. Navigate to the Builder Panel.

40. Click Add Dimension in the Dimensions section.

41. Select dimension Product ID.

42. Click Add Measure in the Measures section.

43. Select the option Create Calculation.

44. As Type select the option Aggregation.

45. Enter Average Discount Rate as Name.

46. Select Average as Operation.

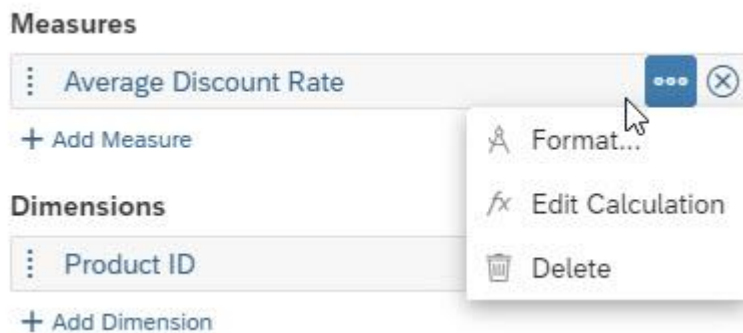
47. Select Discount as Measure.

48. Select Sales Order Line Number as Aggregation Dimension.

49. Click OK.

50. Select the measure Average Discount Rate in the Measures section.

51. Open the context menu.



52. Select the menu Format.

53. Disable the option Use unit of underlying measures.

54. Set the Scale to Percentage.

55. Set Decimal Places to 1.

56. Click OK.

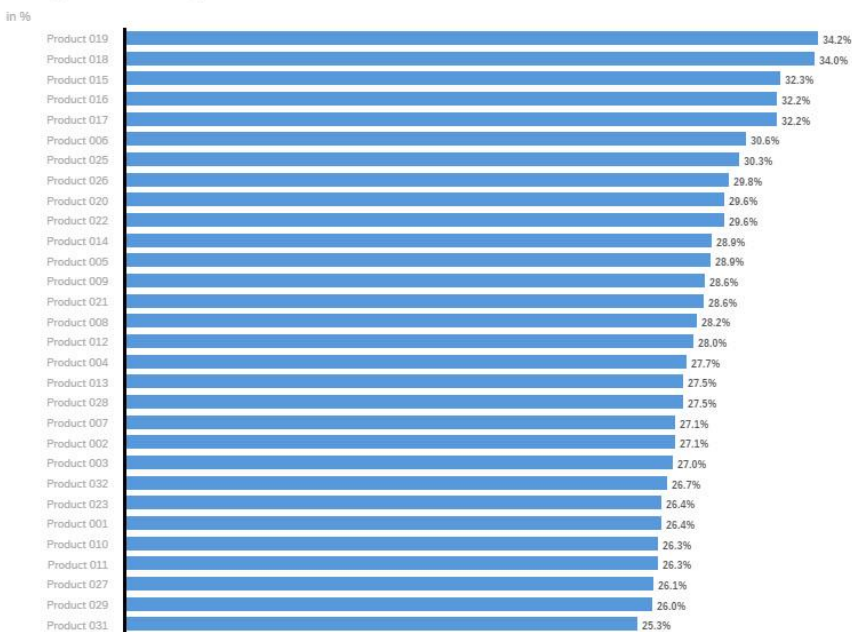
57. Select the chart on the canvas.

58. Open the context menu in the top right corner.

59. Select the menu Sort > Average Discount Rate > Highest to Lowest.

60. So we can clearly see that Product 019 has the highest average discount rate.

Average Discount Rate per Product ID

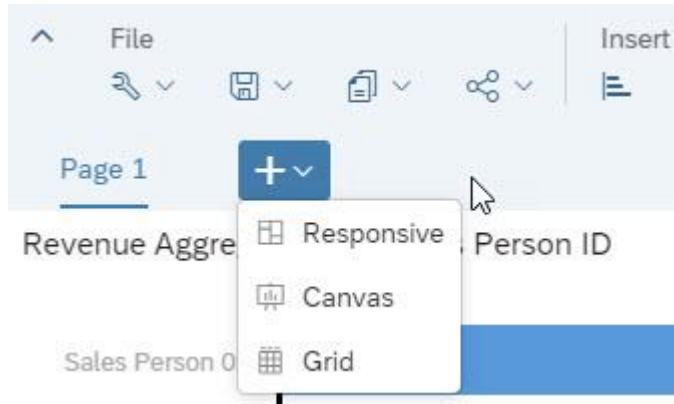


61. Save your changes.

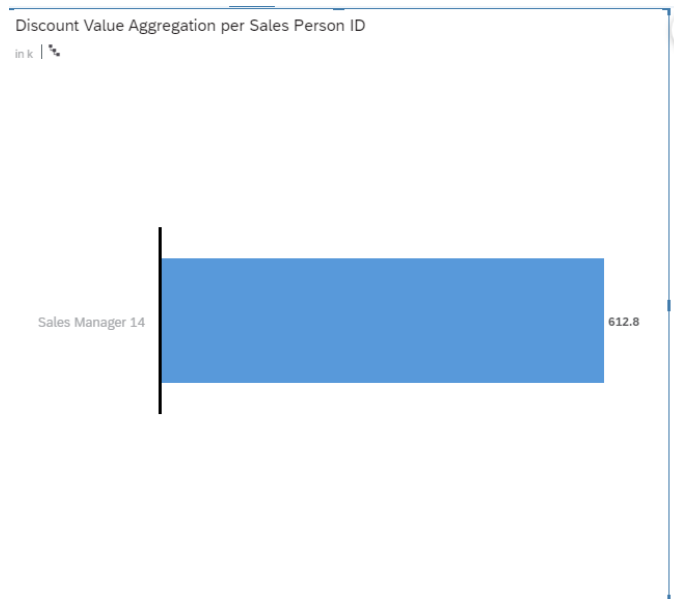
### 4.3 SAP Analytics Cloud - Exercise 03: Which Sales Person is giving the most discount?

In the following steps, we will use the previously created View to find out, which sales person has been giving the highest discount to our customers. In the previous steps we already created the necessary calculations, so we can re-use those to answer these questions.

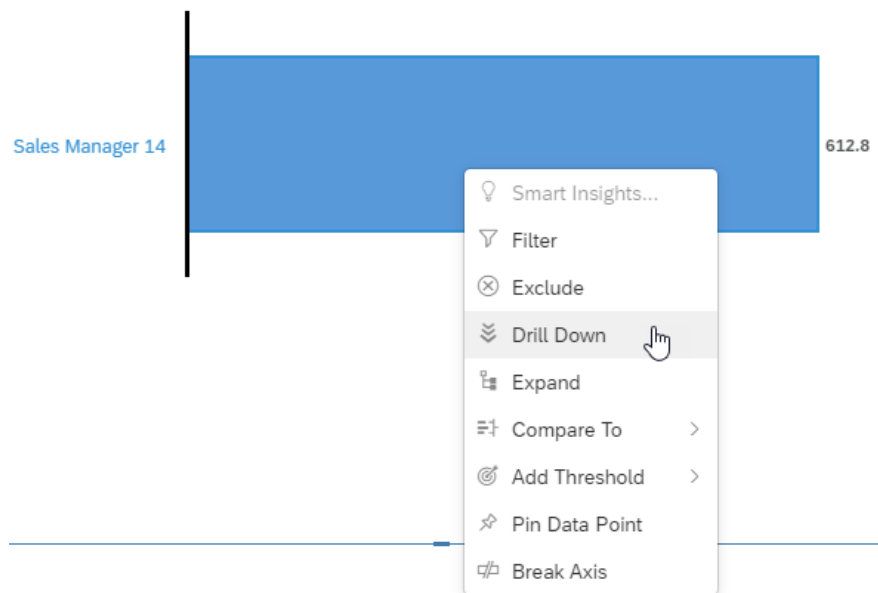
1. With your existing story open, click on the “+” sign next to the tab for Page 2.



2. Select the option to add a new Canvas Page.
3. Select the option to add a new Chart.
4. The Chart will leverage the ANA361\_XX Analytics View as default data source, based on our previous selection.
5. Select the chart on the canvas.
6. Navigate to the Builder Panel.
7. Click Add Dimension as part of the Dimensions section.
8. Select the dimension Sales Person ID.
9. Click Add Measures as part of the Measures section.
10. Select the measure Discount Value Aggregation.

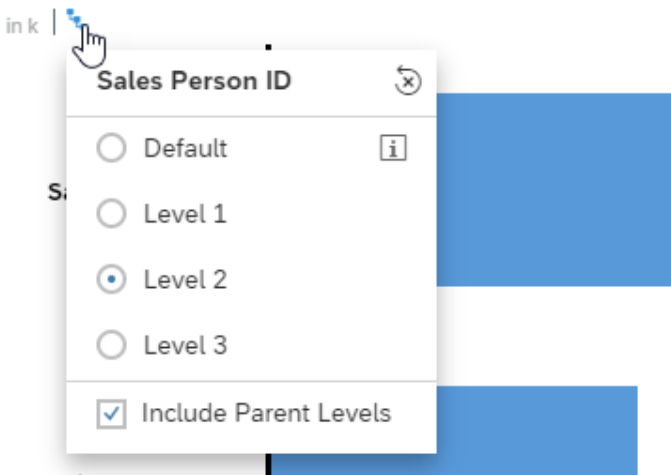


11. By default, the chart will leverage the hierarchy, which we added to our view.
12. Select the bar for the Sales Manager 14 on the chart.
13. Use a right-click.

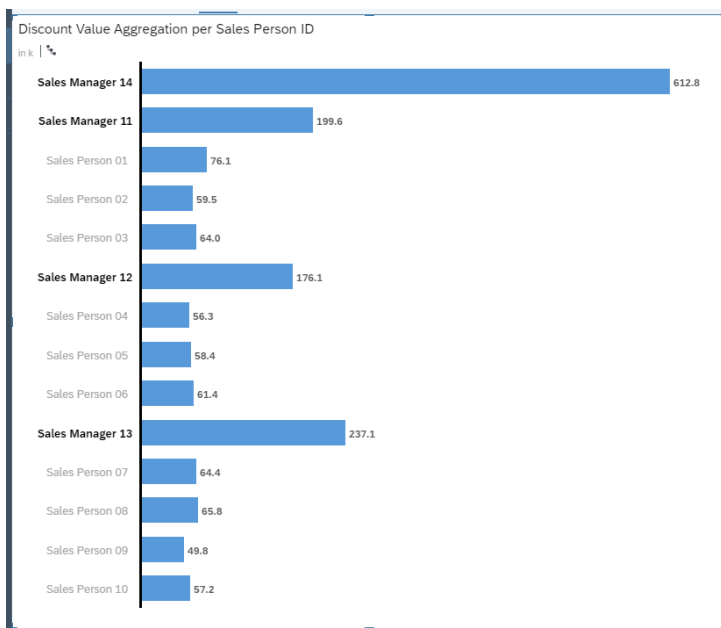


14. Select the menu option Expand and you will see the details for the next level.
15. In the chart title, you can also use the hierarchy symbol to expand the hierarchy.

## Discount Value Aggregation per Sales Person ID



16. Click on the icon in the chart title and select Level 03.



17. So we can see that Sales Person 01 has the highest absolute discount value.

18. As next step we need to find out, which sales person has the highest discount rate in %.

19. In the toolbar, select the option to insert a new chart.

20. Select the new chart on the canvas.

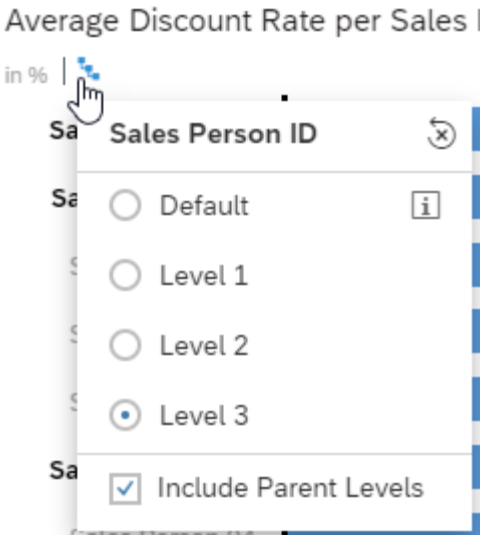
21. Navigate to the Builder Panel.

22. Click Add Dimension in the Dimensions section.

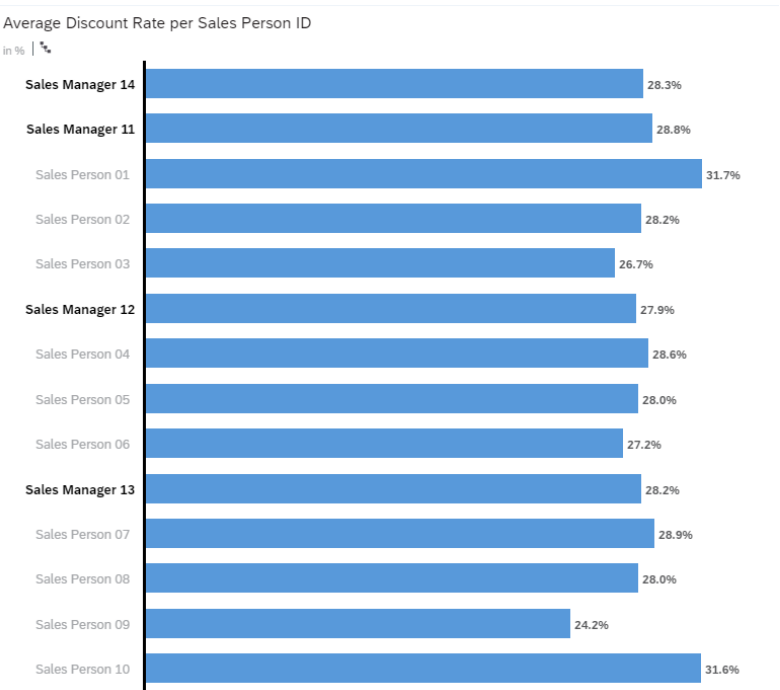
23. Select dimension Sales Person ID.

24. Click Add Measure in the Measures section.

- 25. Select measure Average Discount Rate.
- 26. Select the chart on the canvas.
- 27. Use the hierarchy icon in the chart title and drill down to Level 03 of the hierarchy.
- 28. Ensure the option Include Parent Levels is enabled.



- 29. We can see that Sales Person 01 has the highest discount rate, closely follow by Sales Person 10.



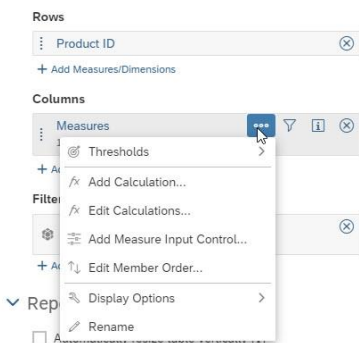
- 30. Save your changes.

#### 4.4 SAP Analytics Cloud - Exercise 04: Which Product is the most profitable product?

In the following steps, we will use the previously created View to find out, which product has the highest profitability (in %). We will consider the original unit price, the product cost, as well as the discounted price and we will show the average profitability, once calculated with the discount and once without.

For these types of calculations, we did already create the formulas as Calculated Columns in the View as part of the exercise for SAP Data Warehouse Cloud.

1. With your existing story open, click on the “+” sign next to the tab for Page 3.
2. Select the option to add a new Canvas Page.
3. Select the option to add a new Table.
4. The Table will leverage the ANA361\_XX Analytics View as default data source, based on our previous selection.
5. Navigate to the Builder Panel.
6. Click Add Measures / Dimensions for the Rows section.
7. Select dimension Product ID.
8. Open the context menu for the Measures in the Columns section in the Builder Panel.



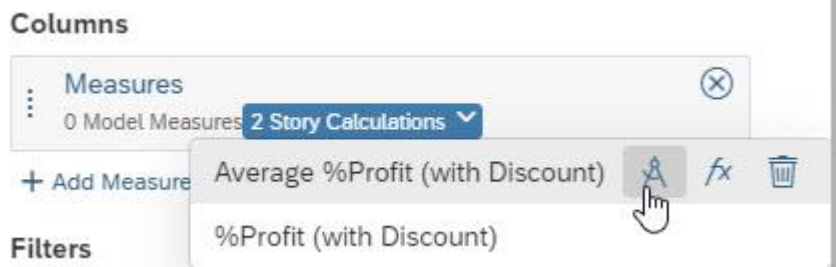
9. Select the option Add Calculation.
10. For the Type option set it to Aggregation.
11. As Name enter Average %Profit (with Discount).
12. Set the Operation to Average.
13. For the Measure select the measure Profit % per Line Item with Discount (which we created as part of the view as a Calculated Column).
14. For the Aggregation Dimensions select dimension Sales Order Line Number.
15. Click OK.
16. Select the table on the canvas.
17. Navigate to the Builder Panel.

18. Use the Filter option for the Measures.



19. Ensure only the Aggregation formula is displayed in the table.

20. In the Columns section open the list of Calculations.



21. Select the Formatting options for the Average %Profit with Discount measure.

22. Disable the option Use unit of underlying measure.

23. Set the Scale to Percentage.

24. Set the Decimal Places to 1.

25. Click OK.

26. Open the context menu for the Measures in the Builder Panel.

27. Select the option Add Calculation.

28. For the Type option set it to Aggregation.

29. As Name enter Average %Profit (without Discount).

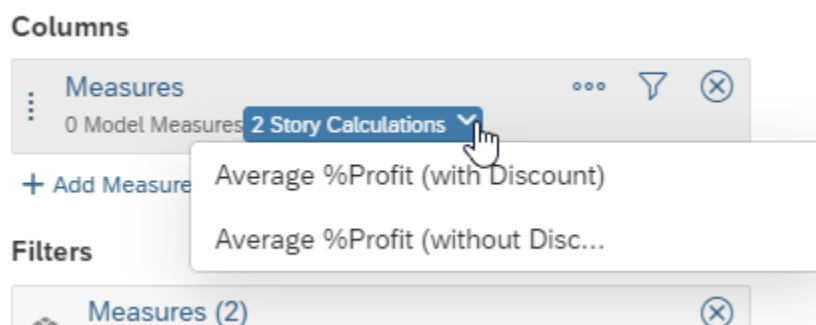
30. Set the Operation to Average.

31. For the Measure select the measure Profit % per Line Item without Discount.

32. For the Aggregation Dimensions select dimension Sales Order Line Number.

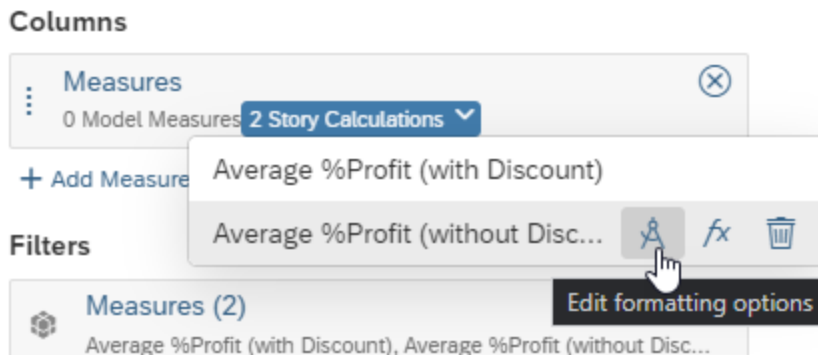
33. Click OK.

34. In the Columns section open the list of Calculations.





35. Select the Formatting options for the Average %Profit (without Discount) measure.



36. Disable the option Use unit of underlying measure.

37. Set the Scale to Percentage.

38. Set the Decimal Places to 1.

39. Click OK.

40. Your table should look similar to the image below:

Product ID	Measures	Average %Profit (with Discount)	Average %Profit (without Discount)
Product 001		24.5	51.0
Product 002		47.9	75.0
Product 003		15.0	42.0
Product 004		20.3	48.0
Product 005		44.1	73.0
Product 006		35.4	66.0
Product 007		49.9	77.0
Product 008		55.8	84.0
Product 009		-7.6	21.0
Product 010		56.7	83.0
Product 011		36.7	63.0
Product 012		21.0	49.0
Product 013		49.5	77.0
Product 014		35.0	64.0
Product 015		37.7	70.0
Product 016		1.9	34.0
Product 017		-12.0	20.2
Product 018		2.0	36.0
Product 019		-18.2	16.0
Product 020		43.5	73.1
Product 021		-12.6	16.0
Product 022		17.5	47.1
Product 023		32.6	59.0
Product 024		28.2	52.9
Product 025		-13.1	17.1
Product 026		-8.8	21.0
Product 027		-3.1	23.0
Product 028		10.5	38.0
Product 029		57.0	83.0
Product 030		26.3	51.0
Product 031		35.8	61.0
Product 032		-6.7	20.0
Product 033		40.8	66.0

41. You can now use the context menu to sort the columns in an ascending or descending way based on the two aggregation calculations.

Measures	Average %Profit (with Discount)	Average %Profit (without Discount)
Product ID		
Product 001		51.0
Product 002		75.0
Product 003		
Product 004		
Product 005		
Product 006		
Product 007		
Product 008		
Product 009		
Product 010		
Product 011		
Product 012		
Product 013	49.5	77.0
Product 014	35.0	64.0
Product 015	37.7	70.0
Product 016	1.9	34.0
Product 017	-12.0	20.2
Product 018	2.0	36.0

42. Use a right-click on the column header.

43. Select the menu Sort Options.

44. Select Sort Descending.

45. Based on those two columns we can see:

- Product 029 is the most profitable product with considering the discount with 56% profitability.
- Product 008 is the most profitable product without considering the discount with 84% profitability.

46. Save your changes.

## 5 SAP DATA WAREHOUSE CLOUD – SELF SERVICE MODELLING

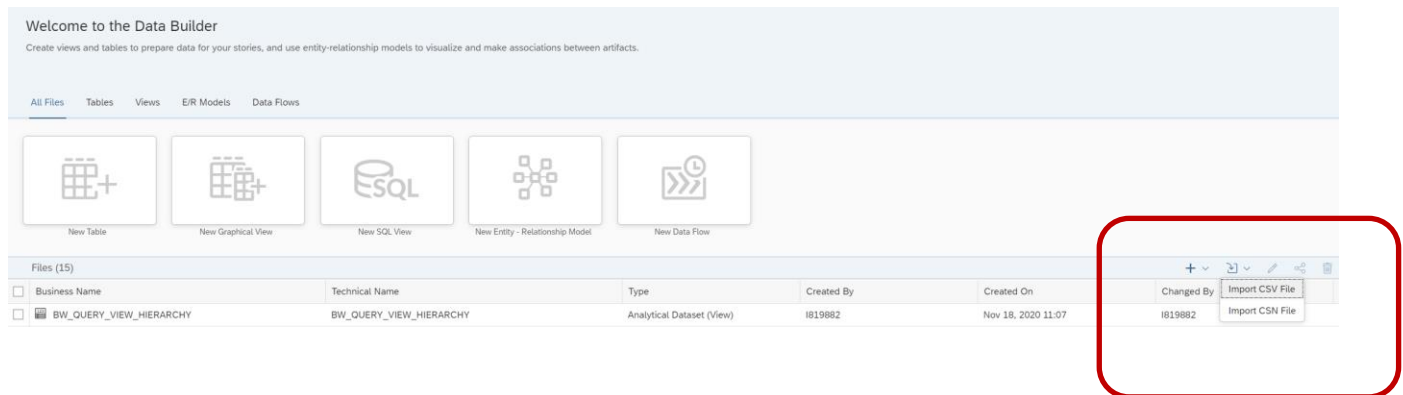
In the following steps, we will use the already existing views in SAP Data Warehouse Cloud and upload additional information in form of a CSV File and then combine the newly created information with the already existing view, so that in SAP Analytics Cloud we are presented with a combined view.

### 5.1 SAP Data Warehouse Cloud – Exercise 10: Uploading Additional Product Information

In the first step of uploading new information and joining the new information with the previously created model, we will upload the information in form of a CSV file.

In the next steps we will upload the CSV File and create a new table based on the file upload:

1. Log On to your SAP Data Warehouse Cloud tenant.
2. Select the menu option Data Builder on the left-hand side.
3. In case you are being asked, select your previously created Space.



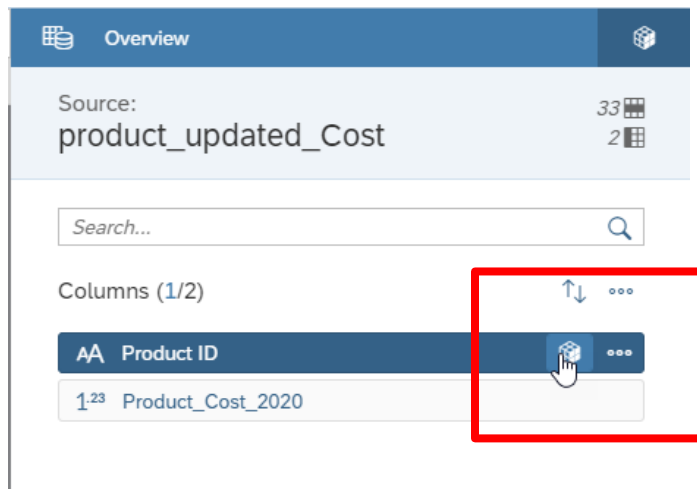
4. Select the option Import CSV File.
5. Click Select Source file.
6. Select the CSV file “product\_updated\_Cost.csv”.
7. Ensure the option Use first row as column header is enabled.
8. Set the CSV Delimiter to the option Auto-detect.
9. Click Upload.

Actions			
Create Transform			
AA	Product ID	1 <sup>st</sup>	Product_Cost_2020
1	PRODUCT001	20.808	
2	PRODUCT002	17.4862	
3	PRODUCT003	6.622	
4	PRODUCT004	12.213	
5	PRODUCT005	3.377	
6	PRODUCT006	4.8678	
7	PRODUCT007		
8	PRODUCT008		

## Session ID: ANA361 - Integrating SAP Analytics Cloud and SAP Data Warehouse Cloud

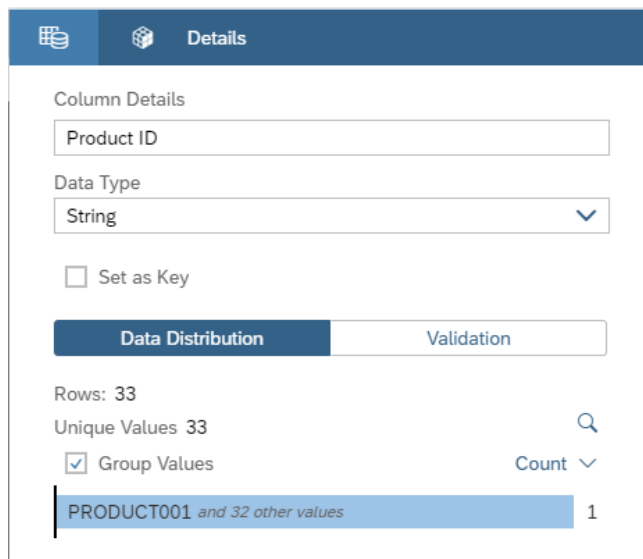
10. Click on the column header for the Product ID.

11. In the Overview panel on the right-hand side, click on the cube icon to see the details for the Product ID.



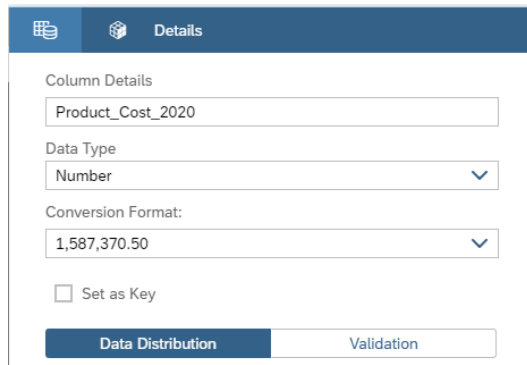
12. Ensure the Data Type is set to String.

13. Enable the option Set as Key.



14. Repeat the steps and take a look at the details for column Product Cost 2020.

15. Ensure the Data Type is set to Number.



Details

Column Details

Product\_Cost\_2020

Data Type

Number

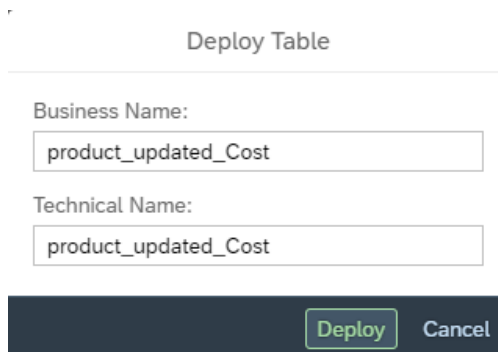
Conversion Format:

1,587,370.50

☐ Set as Key

Data Distribution Validation

16. Click Deploy (bottom right).



Deploy Table

Business Name:

product\_updated\_Cost

Technical Name:

product\_updated\_Cost

Deploy Cancel

17. Enter Product Cost 2020 as Business Name.

18. Enter Product\_Cost\_2020 as Technical Name.

19. Click Deploy.


20. You should receive a message about the successful deployment and the newly created table should be listed in the start screen for the Data Builder.


Session ID: ANA361 - Integrating SAP Analytics Cloud and SAP Data Warehouse Cloud


Welcome to the Data Builder


Create views and tables to prepare data for your stories, and use entity-relationship models to visualize and make associations between artifacts.


All FilesTablesViewsE/R Models

New Table

Import CSV File

New Graphical View

New SQL View

New Entity - Relationship Model

Files (11)

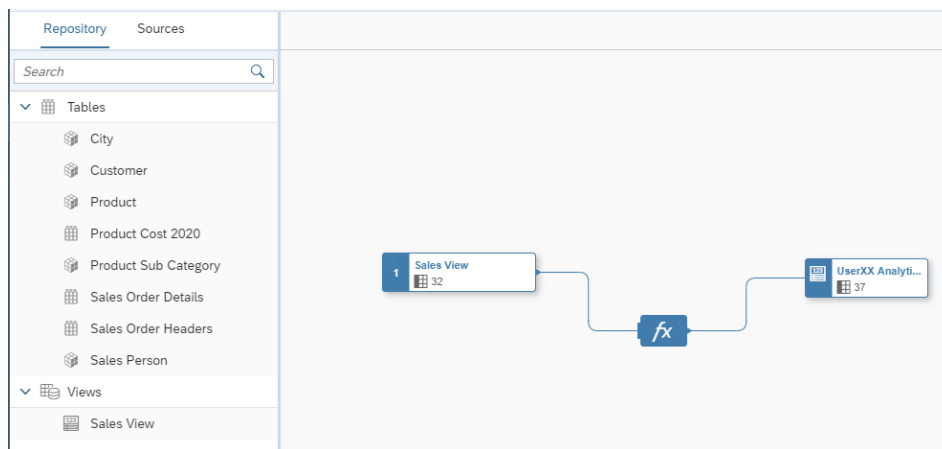
<input type="checkbox"/>	Business Name	Technical Name	Type
<input type="checkbox"/>	City	City	Dimension (Local Table)
<input type="checkbox"/>	Customer	Customer	Dimension (Local Table)
<input type="checkbox"/>	Product	Product	Dimension (Local Table)
<input type="checkbox"/>	Product Cost 2020	Product_Cost_2020	Relational Dataset (Local Table)
<input type="checkbox"/>	Product Sub Category	Product_Sub_Category	Dimension (Local Table)
<input type="checkbox"/>	Sales Order Details	Sales_Order_Details	Relational Dataset (Local Table)
<input type="checkbox"/>	Sales Order Headers	Sales_Order_Headers	Relational Dataset (Local Table)
<input type="checkbox"/>	Sales Person	Sales_Person	Dimension (Local Table)
<input type="checkbox"/>	Sales View	Sales_View	Analytical Dataset (View)
<input type="checkbox"/>	UserXX Analytics View	UserXX_Analytics_View	Analytical Dataset (View)
<input type="checkbox"/>	UserXX Entity Model	UserXX_Entity_Model	E/R Model

## 5.2 SAP Data Warehouse Cloud – Exercise 11: Adjusting View

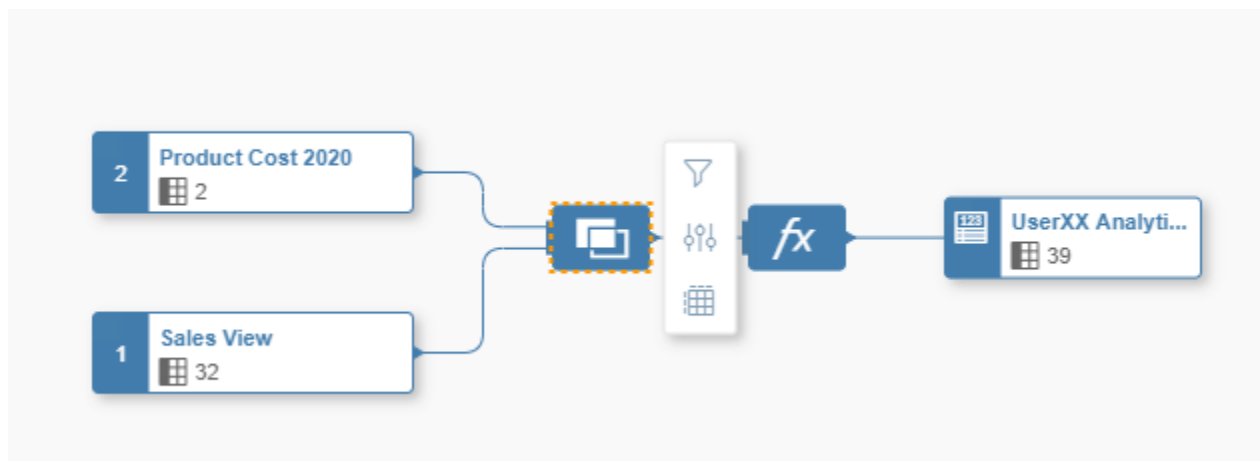
We uploaded the CSV file and created a new table in SAP Data Warehouse Cloud based on the CSV File. As next step we will now combine the information from the CSV File with our already existing view.

In the next steps we will extend our existing View with the newly added table based on the CSV file:

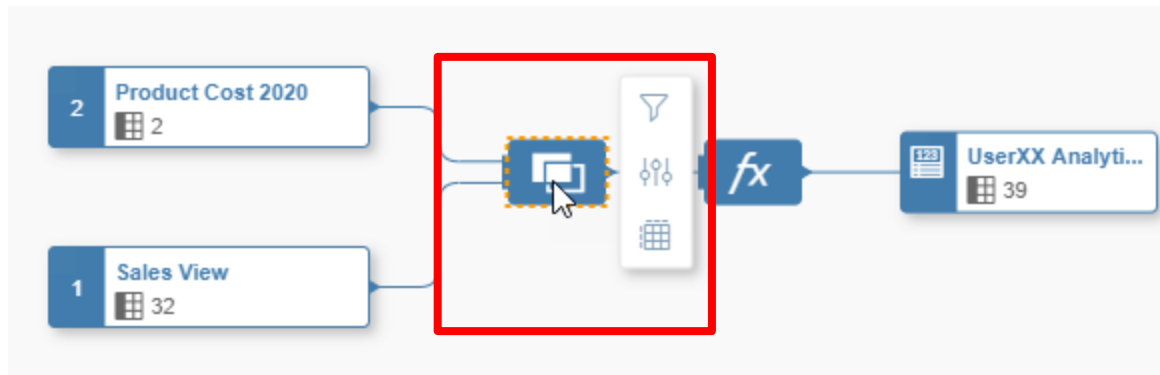
1. Log On to your SAP Data Warehouse Cloud tenant.
2. Select the menu option Data Builder on the left-hand side.
3. In case you are being asked, select your previously created Space.
4. Open the entry for our view – ANA361\_XX Analytics View.



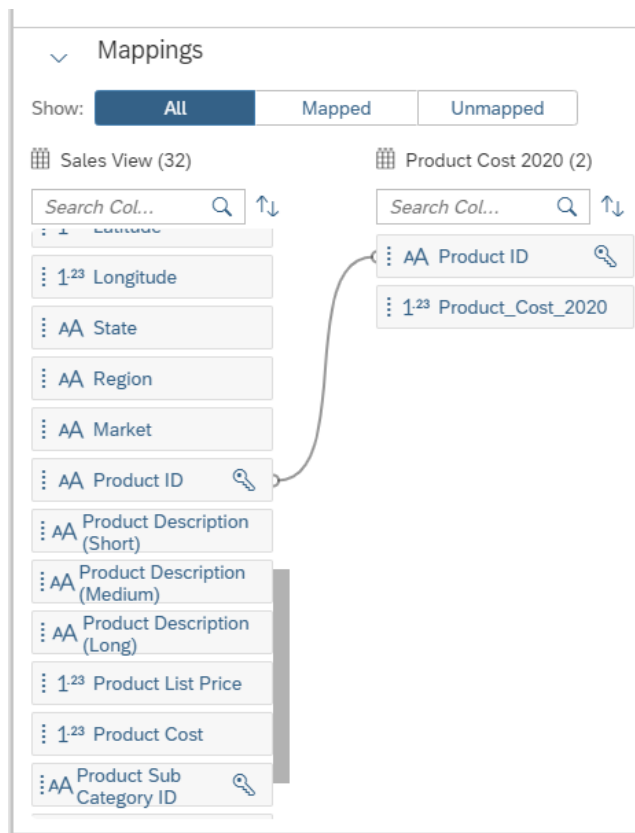
5. Now drag and drop the table Product Cost 2020 from the Repository on top of the Sales View node on the canvas.



6. The Table Product Cost 2020 will be joined with the Sales View.
7. Select the Join node on the canvas.



8. Ensure the Details are shown on the right hand side.
9. Ensure the Join is defined on the Product ID.



10. Ensure the Join Type is set to Inner.



At this point we joined the newly created table with the Product Cost for 2020 to our already existing view. What we can now add to our overall output is the calculation for the Profit and Profit in % based on the newly updated Product cost.

11. Click on the element for the Calculated Columns on the canvas – the “fx” icon.
12. Click on the “+” icon to add a new Calculated Column.
13. Enter Profit (Product Cost 2020) as Business Name.
14. Enter Profit\_Product\_Cost\_2020 as Technical Name.
15. Enter the following formula:

$\text{Order\_Quantity} * (\text{Unit\_Price} - \text{Product\_Cost\_2020})$

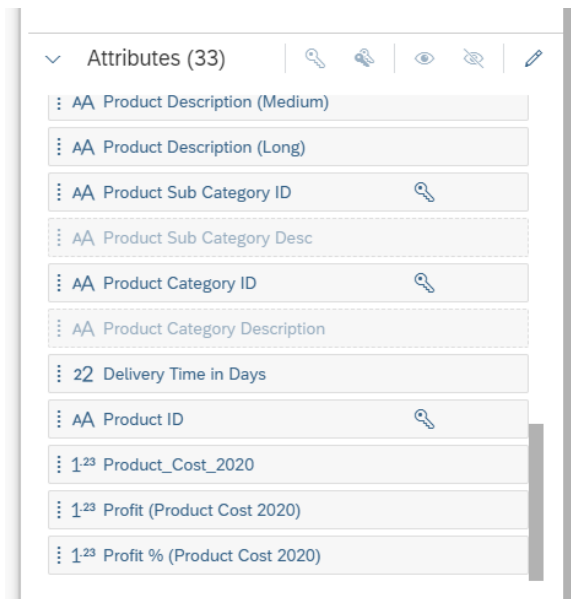
We are using the Unit Price per Product, subtracting the Product Cost for 2020, and multiply it with the Order Quantity per Line item of the orders we have.

16. Click Validate.
17. Ensure the Data Type is set to Decimal with a Precision set to 6 and a Scale set to 2.
18. Click on the Columns option (top left of the Properties).
19. Click on the “+” icon to add a new Calculated Column.
20. Enter Profit % (Product Cost 2020) as Business Name.
21. Enter ProfitPerc\_Product\_Cost\_2020 as Technical Name.
22. Enter the following formula:

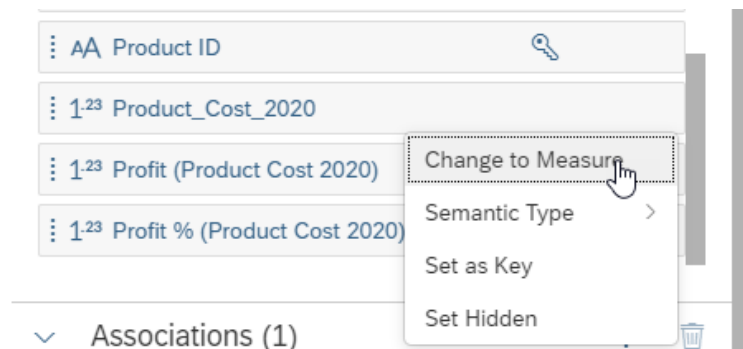
$(\text{Unit\_Price} - \text{Product\_Cost\_2020}) / \text{Unit\_Price}$

23. Click Validate.
24. Ensure the Data Type is set to Decimal with a Precision set to 6 and a Scale set to 2.
25. Click on the Columns option (top left of the Properties).
26. Now select the overall output element ANA361\_XX Analytics View.
27. Ensure the Properties are shown.

28. Scroll down in the list of Attributes.



29. Select each of the newly created Calculated Columns and the column for Product Cost 2020 (they should be at the bottom of the list) and open the context menu and select the option Change to Measure.



30. Now select the overall output element ANA361\_02 Analytics View.

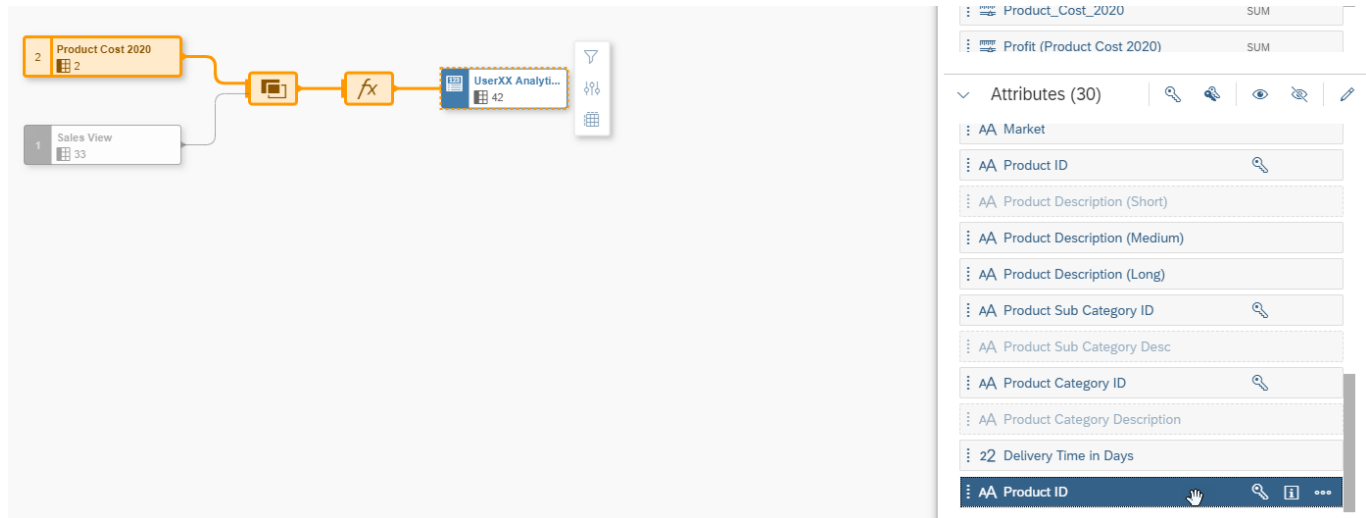
31. Ensure the Properties are shown.

32. Scroll down in the list of Attributes.

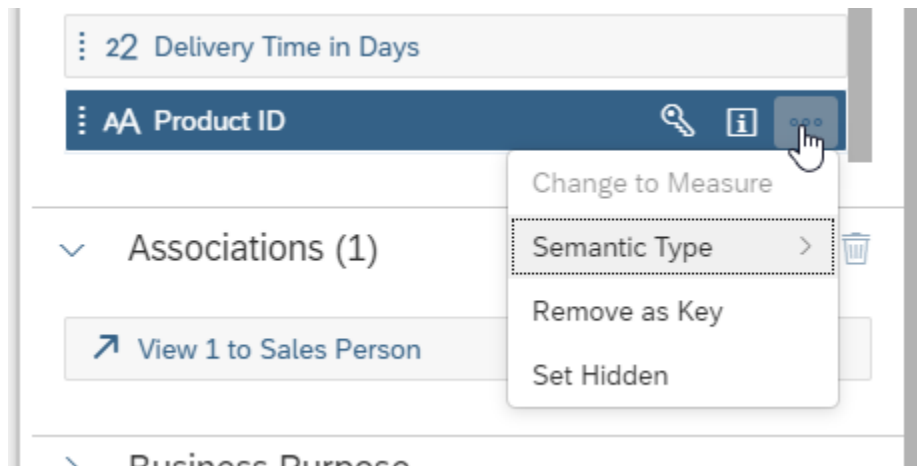
33. You will notice Product ID is showing twice in the list.

34. Click on each of the entries for Product ID in the list of Attributes.

35. Select the one that shows the origin to be the Product Cost 2020 table.



36. Open the context menu for the entry.



37. Select the option Set Hidden.

38. Save your changes.

39. Deploy the changes to the Sales View.

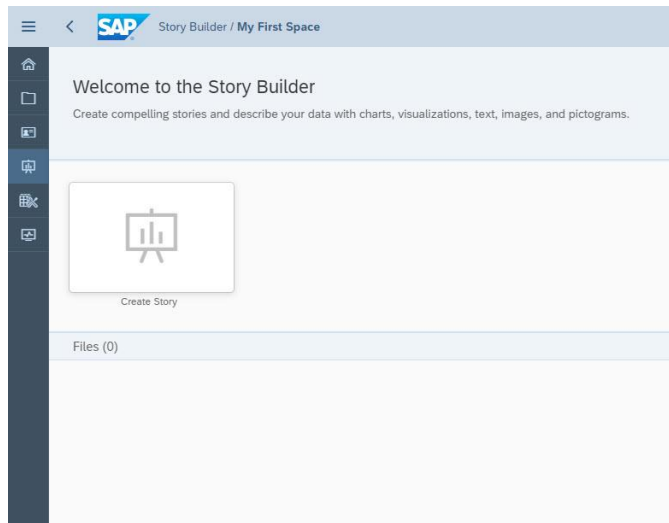
We basically added a set of Calculated Columns to the Sales View and we will use those calculations when we create the story with the Story Builder.

### 5.3 SAP Analytics Cloud - Exercise 05: Adding Product Cost for 2020

In the previous steps we uploaded the CSV File and we adjusted the model in SAP Data Warehouse Cloud to now also to include the information on the updated Product Cost. In the next steps we will then use the updated information and compare it to the original information, to see how much more or less profitable the products are.

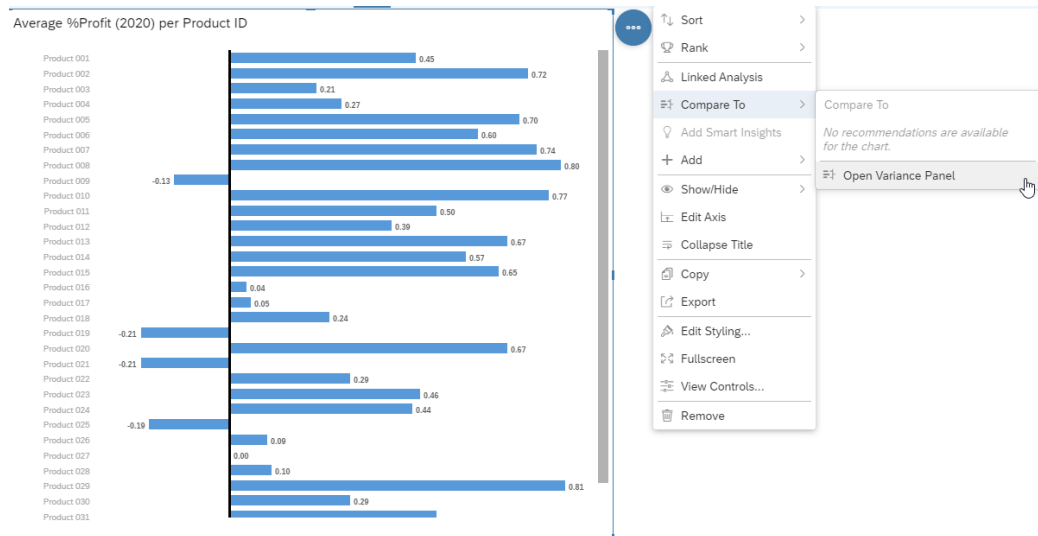
In the given example, we can re-use the calculations that we created as part of the view in SAP Data Warehouse Cloud, but we also need to then create an Average aggregation for the measure in SAP Analytics Cloud as we want to see the value across all products and the data provides the information on a Order Line Item level.

1. Log On to your SAP Data Warehouse Cloud tenant.
2. Select the menu option Story Builder on the left-hand side.
3. In case you are being asked, select the Space you created in the previous exercise.



4. Open the SAP Analytics Cloud story that we created in the previous steps. In case you didn't finish the SAP Analytics Cloud story from the previous exercises, please create a new story and skip step 5 and 6.
5. Click Edit in the top right corner.
6. With your existing story open, click on the "+" sign next to the existing tabs to create an additional page.
7. Select the option to add a new Canvas Page.
8. Select the option to add a new Chart.
9. The chart will leverage the ANA361\_XX Analytics View as default data source, based on our previous selection.
10. Navigate to the Builder Panel.
11. Click Add Measures / Dimensions for the Rows section.
12. Select dimension Product ID.
13. Open the context menu for the Measures in the Columns section in the Builder Panel.
14. Click Add Measures as part of the Measures section.
15. Select the option Create Calculation.
16. For the Type option set it to Aggregation.
17. As Name enter Average %Profit (2020).
18. Set the Operation to Average.

19. For the Measure select the measure Profit % (Product Cost 2020) (which we created as part of the view as a Calculated Column).
20. For the Aggregation Dimensions select dimension Sales Order Line Number.
21. Click OK.
22. Select the chart on the canvas.
23. Open the context menu of the chart.



24. Select the menu Compare To > Open Variance Panel.

Create Variance (A- B)

**COMPARE (A)**

Measure  
Average %Profit (2020)

+ Add Version

**TO (B)**

Measure  
Average %Profit (without Discount)

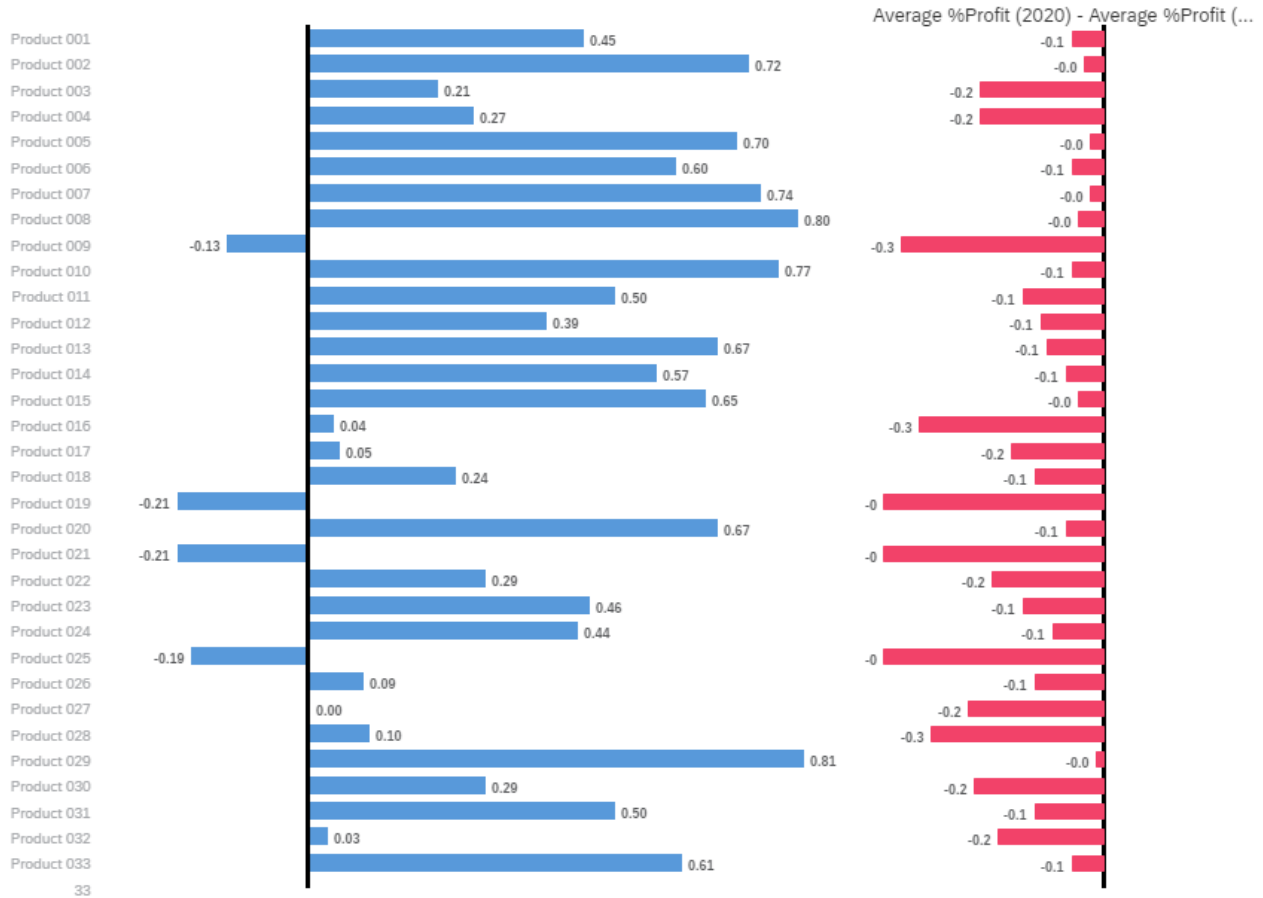
> Display Options

25. For the measure for Compare (A) select the measure Average %Profit (2020).
26. For the measure for Compare (B) select the measure Average %Profit (without Discount). You can open the "ALL" list to see all measures.
27. Click OK in the Variance Panel (bottom right).

You should now have a chart, which shows the %Profit based on the previous uploaded product costs compared to the %Profit based on the newly uploaded CSV file.

Average %Profit (2020) per Product ID

1 Variance



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

© 2020 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 [www.sap.com/copyright](http://www.sap.com/copyright) for additional trademark information and notices.