

Overview about the Business Scenario & the Data Model

The Business Scenario

The sample data set for the session represents retail transactions from a number of outlet stores located in the United States. The transaction details include the store, the sold product, and the sales manager. In addition, the information on revenue, cost, discount, and profit is available for each transaction.

The sales department is looking for a few analytics they need:

- **Top 10 Revenue Generating Products**

They want to discover the top 10 revenue generating products.

- **Sales Per Region**

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

- **Best Sales Representative**

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

The exercises will walk you through the steps using SAP Datasphere and SAP Analytics Cloud to answer those open questions, using the following tables:

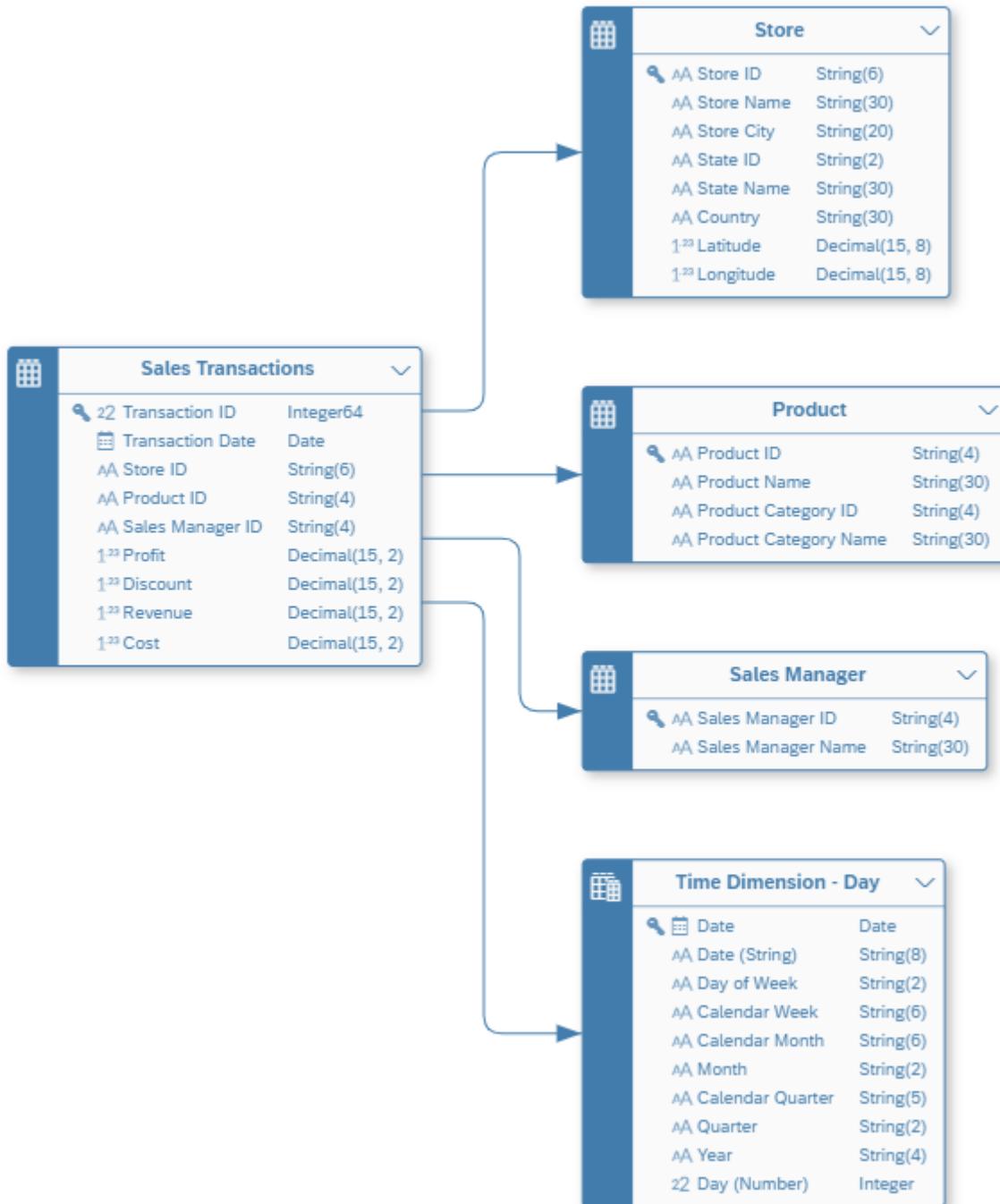
Table Name	Description	Model Type
Sales Transactions	Daily Retail Transactions per Store	Relational Dataset
Store	Details per Store Outlet	Dimension
Sales Manager	Details on all Sales Manager	Dimension
Product	Details on the Products being sold	Dimension

The data is provided in CSV files in this ZIP-file:  [DA180_Resources.zip](#)

The Data Model

These four tables consist of one fact table which includes the sales transactions details, and three dimension tables for more details about the product, the store and the sales manager.

The **Time Dimension** will be associated to the sales transaction table as shown in the Entity-Relationship model.



Summary

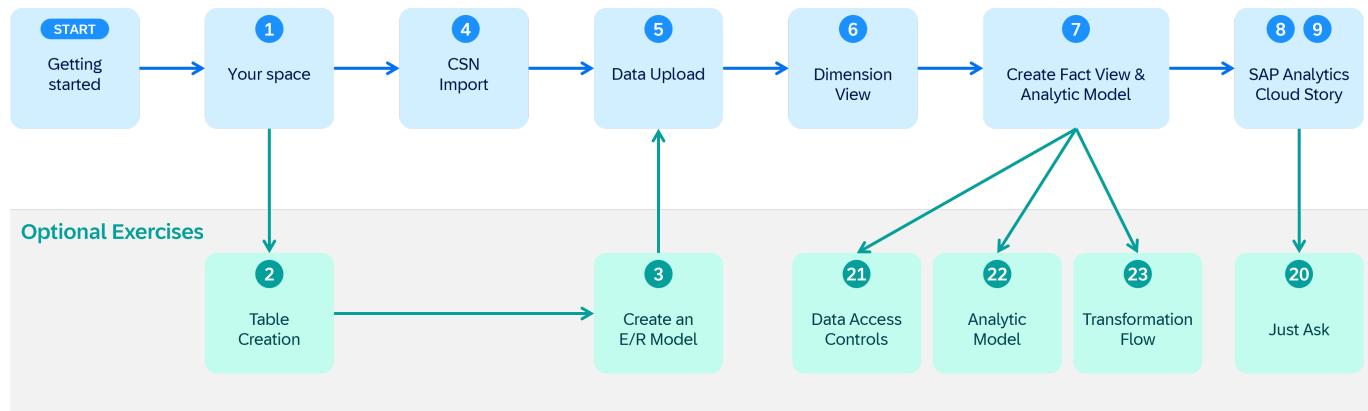
Now that you have a good understanding of the business scenario and the data model, you can start to get a basic trial access.

Continue with [Get your Basic Trial access](#)

Exercise Overview

We structured the overall exercise the following way:

- Use the blue path to finish early and enjoy exploring the system on your own
- You can learn more and use the **Optional Exercises** for deep dives on different topics



Let us give you a brief introduction on what can you expect in the different parts of the exercise.

1. Getting Started

In this area you get an [overview about the business scenario & the data model](#) used throughout the exercises.

We also give you guidance on how to [get your basic trial access](#) in case you do not have access to a SAP Datasphere system already.

If you have never touched an SAP Datasphere system you can get an overview about your [first log on](#) and how you navigate in the system.

2. SAP Datasphere

This section covers the main modeling exercises where you prepare tables, views and an analytic model to build the foundation for the stories in SAP Analytics Cloud.

In the first exercise [Get to know your own Space](#) you learn more about the concept of Spaces and the **Time Dimension** required for your modeling in later parts of the exercise.

You can [manually create the tables](#) and the [entity relationship model](#), or directly generate the tables and ER model by using the [importing the tables](#) provided with the [CSN file](#) called *Sales_ER_Model.json*.

Then you populate the tables with data by [uploading data files](#) using the CSV files from the ZIP file. Based on these tables you then [create a dimension view](#) as well as a [fact view and an analytic model](#).

3. SAP Analytics Cloud

In this part you will create simple stories in SAP Analytics Cloud to learn how to visualize your data based on a live connection to SAP Datasphere. The first story shows the [Top 10 Revenue Generating Products](#) and the second the [Revenue by Geography](#). Both require the previous exercises to be completed up to exercise 08.

4. Optional Exercises

These exercises are all optional and you can learn how to

- leverage Just Ask with SAP Analytics Cloud to [Identify Top Sales Managers with Just Ask](#)
- enable row-level security with Data Access Controls by [Creating Row-Level Permissions based on External Hierarchy](#)
- [explore more features of the Analytic Model](#)
- and learn about Transformation Flows by [Creating a Transformation Flow and the usage of delta tables](#)

Summary

Now that you have a good understanding of the exercises, lets get started with the hands-on part.

Continue with [Overview about the business scenario and the data model](#)

Exercise 1: Get to know your own Space



Note: This is an optional exercise.



Detour: SAP Datasphere - Spaces

Spaces as part of the SAP Datasphere solution are virtual team environments where your administrator has the ability to assign users and roles, as well as additional resources, connections to data sources, and allocated storage.

In SAP Datasphere all data related workflows start with the selection of a space, so you can see the space is a fundamental concept. Users can share tables and view to another space to allow users assigned to that space to use it as a source for their objects.

End of Detour

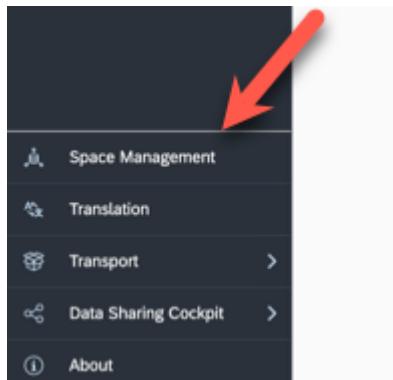
Get to know your own Space



Important:

In case you are not using a guided experience trial system for this hands-on training, then you require a space on the SAP Datasphere tenant you are using.

1. In the menu on the left-hand side, select the option **Space Management**.



2. After you selected the menu item, you will be presented with a list of existing spaces.

Spaces (1)

GE104421

GE104421

✓ Healthy

Deployed On Jul 18, 2024 11:32:56

Disk for Storage:
1.9 MB

Memory for Storage:
2.1 MB

SAP Datasphere

8 1 11 16 Edit

3. Click the **EDIT** button of your assigned space. The pre-defined spaces are usually named the same way as your user ID for example GE12345. The technical ID is identical to the space name.

Space Management | GE

Overview Workload Management Users Database Access Connections Time Data Auditing

General Settings

Space ID: GE Space Name: GE Space Status: Active Space Type: SAP Datasphere

Created By: [Redacted] Created On: Jul 18, 2024 11:24:43 Deployment Status: Deployed Deployed On: Jul 18, 2024 11:32:56

Description (4000 Characters Maximum):

Space Storage:
Enable Space Quota

Translation: ?
Source Language:

Workload Management

Priority
You can specify the prioritization of this space when querying the database. Enter a value from 1 (lowest priority) to 8 (highest priority). In a situation where spaces are competing for available threads, those with higher priorities are run before spaces with lower priorities.

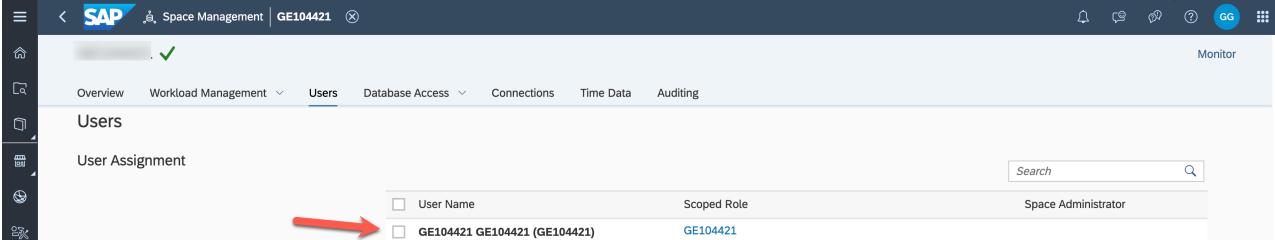
Space Priority: ?
1 (Low) [Slider] 8 (High)

Workload Configuration
Space Configuration: Default ?

4. You are now being presented with the properties of your space and the abilities to configure certain options.

Changing the storage assignment or workload management options is not possible in basic trial systems.

5. In the "Users" section you can see that your user (e.g. GE12345) is already assigned to the Space.

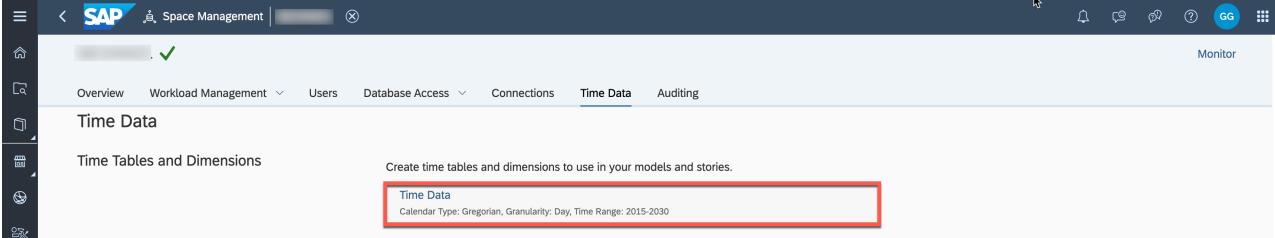


The screenshot shows the SAP Space Management interface for space GE104421. The 'Users' tab is selected. A red arrow points to the second row of the 'User Assignment' table, which lists 'GE104421 GE104421 (GE104421)' with a 'Space Administrator' role.

User Name	Scoped Role
GE104421 GE104421 (GE104421)	Space Administrator

6. The **Time Data** section shows that time table and dimensions have already been created in your space.

They provide standardized time data which we will use later as part of our model to create a date hierarchy. If you are using an own Datasphere tenant, please create the Time Tables and Dimensions.



The screenshot shows the SAP Space Management interface for space GE104421. The 'Time Data' tab is selected. A red box highlights the 'Time Data' section, which displays the message: 'Create time tables and dimensions to use in your models and stories.' Below this, it shows 'Time Data' with 'Calendar Type: Gregorian, Granularity: Day, Time Range: 2015-2030'.

Summary

You have explored the settings of your space in SAP Datasphere. You can now start your next step and create your first data model.

Continue to - [Exercise 4: Importing Tables](#) or [Exercise 02: Prepare Your Data \(optional\)](#)

Exercise 2 - Data Layer - Prepare Your Data



Note: This is an Optional Exercise



Detour: SAP Datasphere - Model Types

Before we are going to start with the first exercise in creating a table and creating the first model, let us have a look at the different asset types that you can create in the Data Builder of SAP Datasphere.

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

- **Table:** Create a Table to contain data by defining its column structure. You can configure each field of the Table, and already define semantics and associations. You have the ability to upload data to this Table later on.
- **Graphical View:** In the Graphical View you can leverage Tables and Views to prepare data and then create new Views using a graphical no-code/low-code editor. Join datasets as required, add other operators to remove or create columns and filter or aggregate data, and specify measures and other aspects of your output structure in the output node.
- **SQL View:** In the SQL View you can leverage Tables and Views to the create new Views by using SQL or SQL-script (table function) view in a powerful SQL editor.
Not used in the DA180 exercises.
- **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.
- **Analytic Model:** Analytic models are the analytical foundation for making data ready for consumption in SAP Analytics Cloud. They allow you to create and define multi-dimensional models to provide data for analytical purposes to answer different business questions.
- **Data Flow:** You define Data Flows to use data transformations and leverage the option to load data from a source system and persist it in SAP Datasphere.
Not used in the DA180 exercises.
- **Replication Flow:** Create a replication flow if you want to copy multiple data assets from the same source to the same target in a fast and easy way and do not require complex projections.
Not used in the DA180 exercises.
- **Transformation Flow:** Create a transformation flow if you want to load data from one or more source tables, apply transformations (such as a join), and output the result in a target table. You can load full or delta data sets of data from one or more source tables to a target table.
- **Intelligent Lookup:** Create an intelligent lookup to merge data from two entities even if they do not have a common identifier.
Not used in the DA180 exercises.
- **Task Chain:** Create a task chain that groups multiple tasks, which can be run in series manually or through a schedule.

Not used in the DA180 exercises.

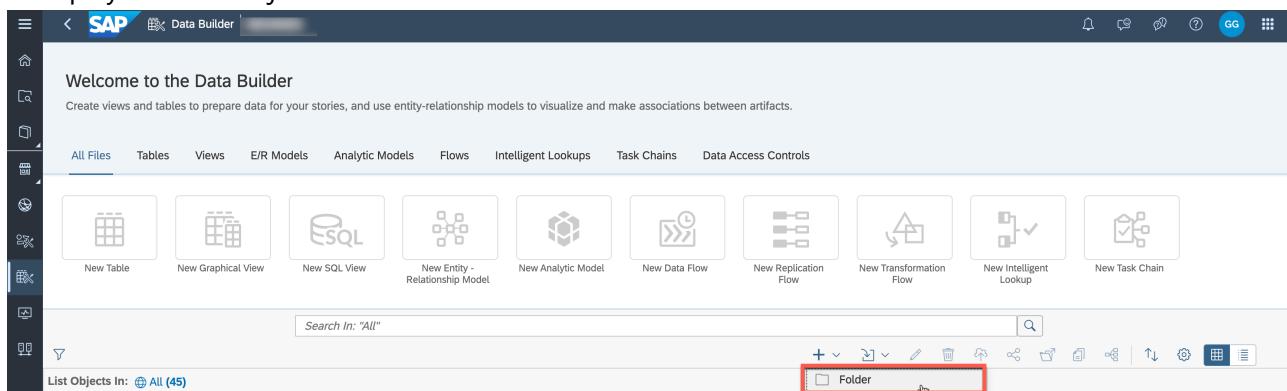
End of Detour

Start of the exercise

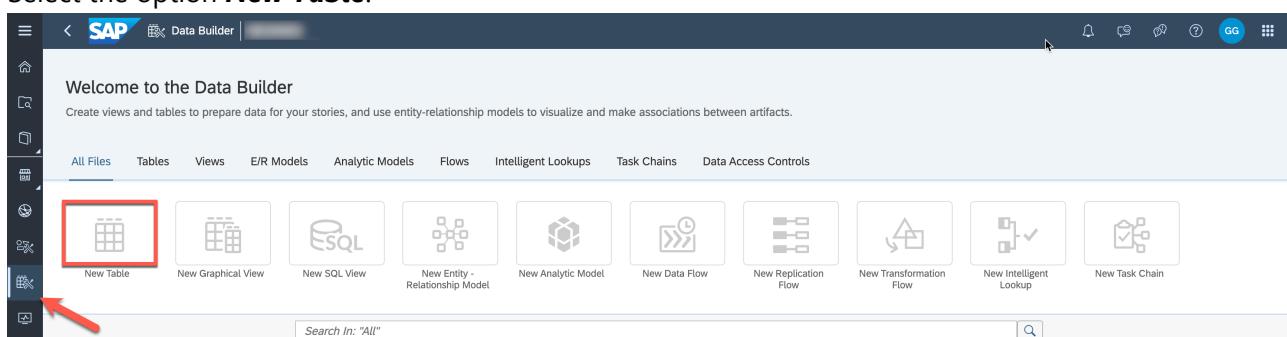
In this section we will start creating the tables for our sample models and then upload the raw data to those tables in [Exercise 5](#).

We will start to create the tables in SAP Datasphere.

1. Log On to your SAP Datasphere tenant.
2. Select the menu option Data Builder on the left-hand side. In case you are being asked, select your space (e.g. GE12345)
3. Create a new folder to organize the entities you will create in this jump-start session by navigating to the "+"-icon on the right and enter "TECHED2024-DA180" as business name and confirm. A new folder is displayed in the object list.



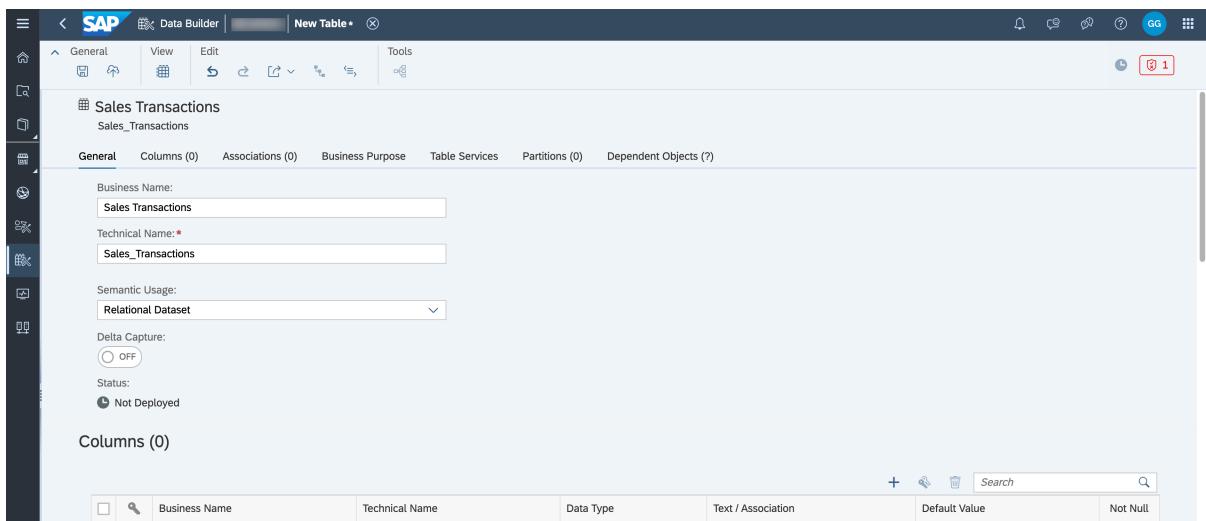
4. Select the option **New Table**.



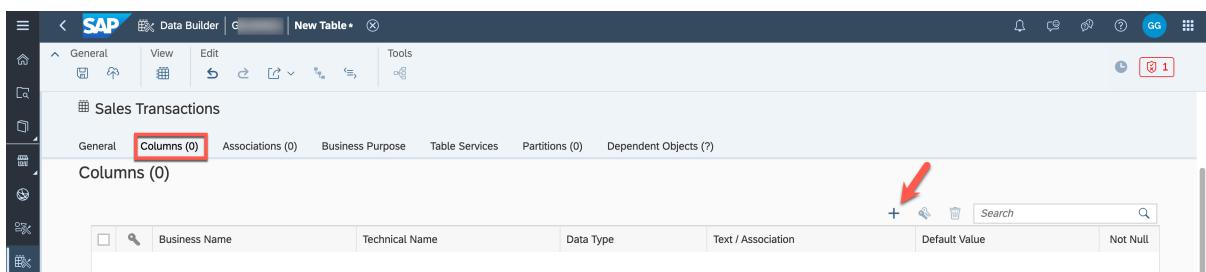
5. You are being presented with the details to create a new table.

6. Enter the following details:

- Business Name - Sales Transactions
- Technical Name - Sales_Transactions
- Semantic Usage - Relational Dataset



- Scroll down to the area **Columns** (you can also use the tabs in the page header for navigation). Here you define the structure of the table by adding the individual columns.
- Use the "+" sign in the top right corner of the **Columns** area to start the process of creating your first table column



- You now need to enter a business name, a technical name, and you need to configure the Data Type
- For the first column, enter the following details:
 - Business Name - Transaction ID
 - Technical Name - Transaction_ID
 - Data Type - Integer64



Tip: Please note, that you can change the **Data Type** simply by clicking on the default data type (e.g. String(100)) in the Data Type column.

-
11. After you entered the details for the first column, please enter the additional columns. All columns of the table are listed as follows:

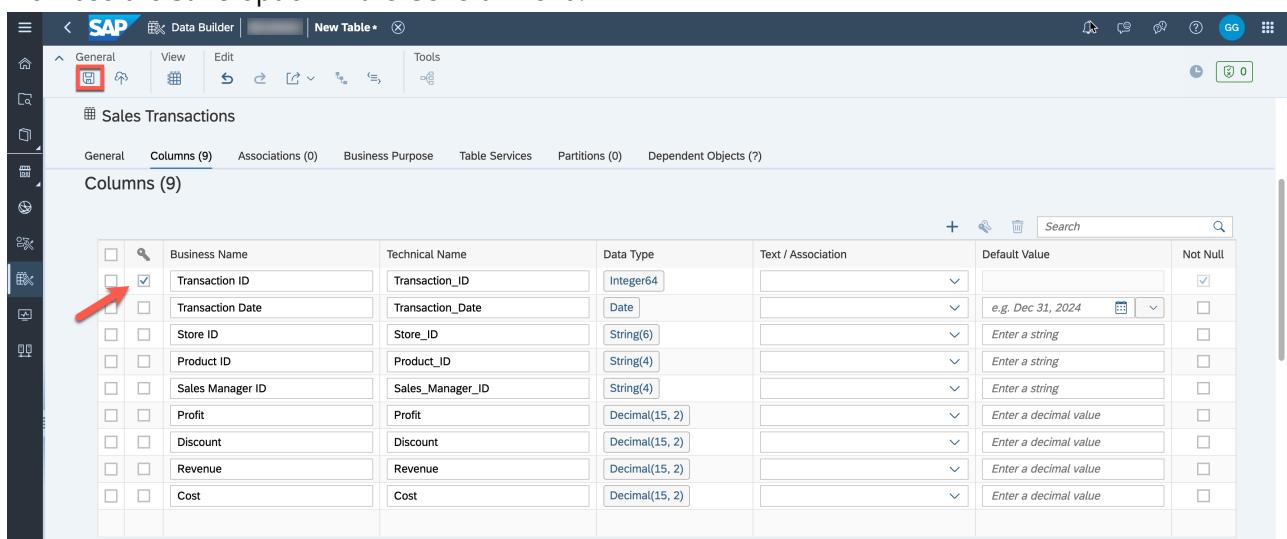
Key:	Business Name:	Technical Name:	Data Type:
X	Transaction ID	Transaction_ID	Integer64
	Transaction Date	Transaction_Date	Date
	Store ID	Store_ID	String (6)

Key: **Business Name:** **Technical Name:** **Data Type:**

Product ID	Product_ID	String (4)
Sales Manager ID	Sales_Manager_ID	String (4)
Profit	Profit	Decimal(15,2)
Discount	Discount	Decimal(15,2)
Revenue	Revenue	Decimal(15,2)
Cost	Cost	Decimal(15,2)

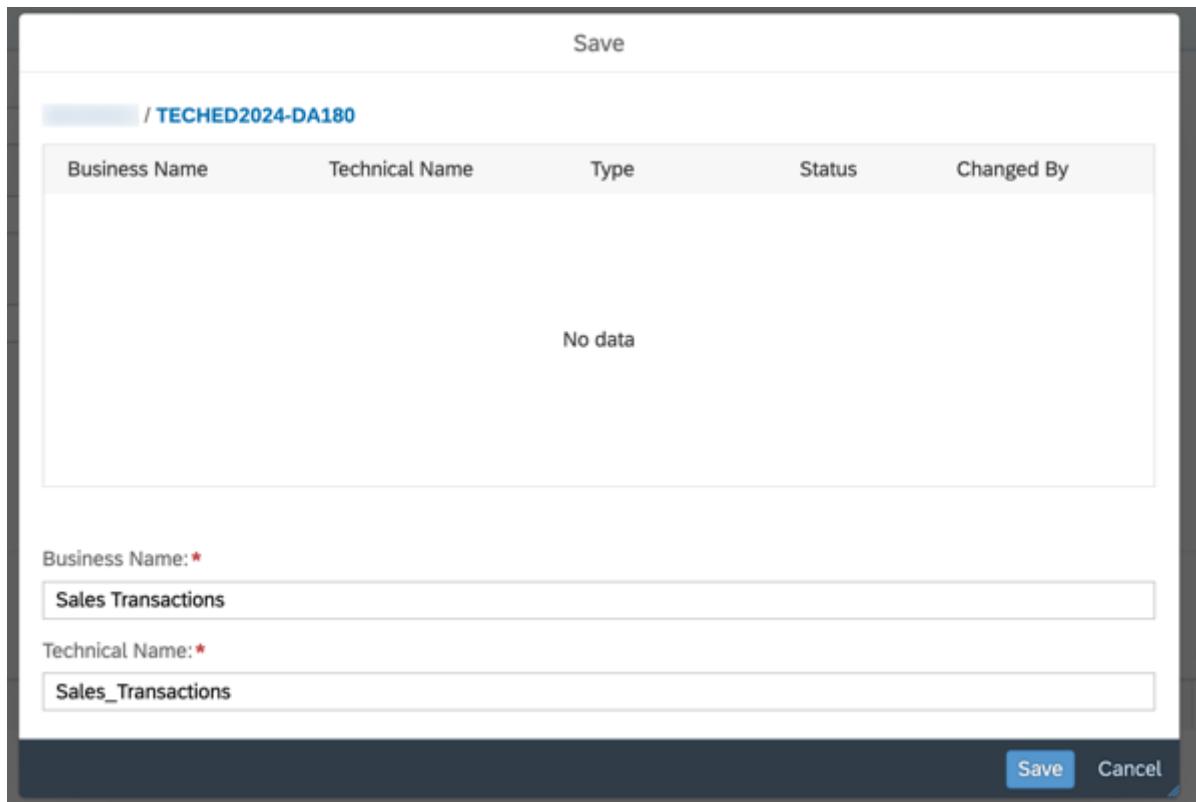
12. After you entered all columns for the table, ensure you enable the Key Column option for the column Transaction ID.

13. Now use the **Save** option in the General menu.



Business Name	Technical Name	Data Type	Text / Association	Default Value	Not Null
Transaction ID	Transaction_ID	Integer64			<input checked="" type="checkbox"/>
Transaction Date	Transaction_Date	Date		e.g. Dec 31, 2024	<input type="checkbox"/>
Store ID	Store_ID	String(6)		Enter a string	<input type="checkbox"/>
Product ID	Product_ID	String(4)		Enter a string	<input type="checkbox"/>
Sales Manager ID	Sales_Manager_ID	String(4)		Enter a string	<input type="checkbox"/>
Profit	Profit	Decimal(15, 2)		Enter a decimal value	<input type="checkbox"/>
Discount	Discount	Decimal(15, 2)		Enter a decimal value	<input type="checkbox"/>
Revenue	Revenue	Decimal(15, 2)		Enter a decimal value	<input type="checkbox"/>
Cost	Cost	Decimal(15, 2)		Enter a decimal value	<input type="checkbox"/>

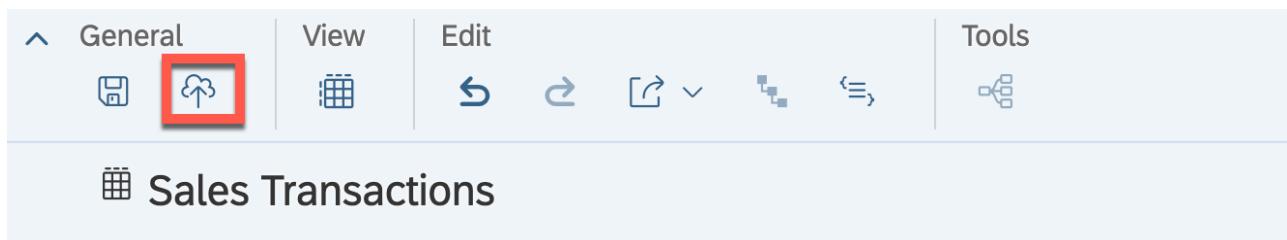
14. Select the folder "TECHED2024-DA180". On the first time you save the table, you will be asked to confirm the name and technical name.



15. Click **Save**.

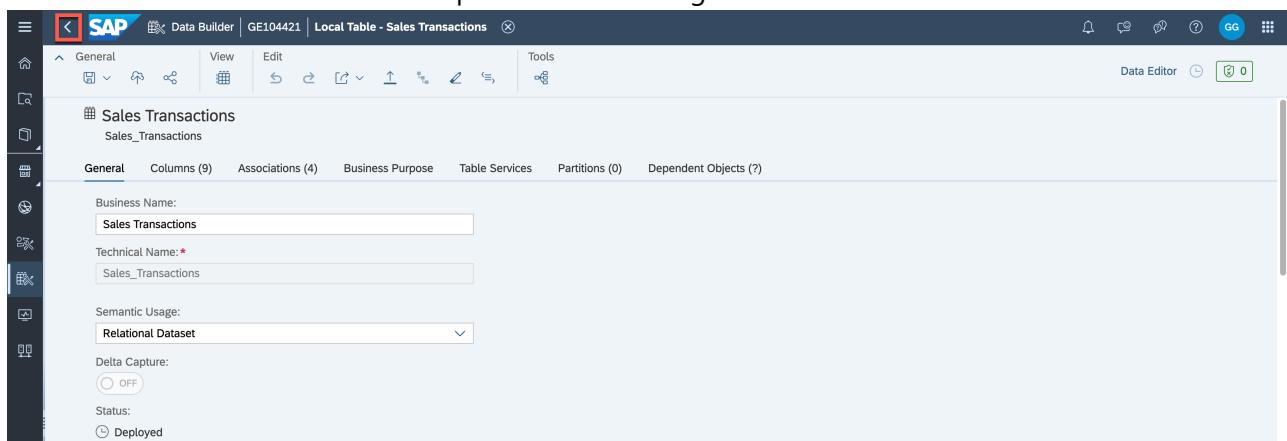
16. After you saved the changes, you also have to deploy the table, so that we can later on upload data to the table.

17. Use the Deploy option from the General menu.



18. We configured, saved, and deployed our first table.

19. Click on the "**back arrow**" on the top menu bar to navigate back to the list of tables.

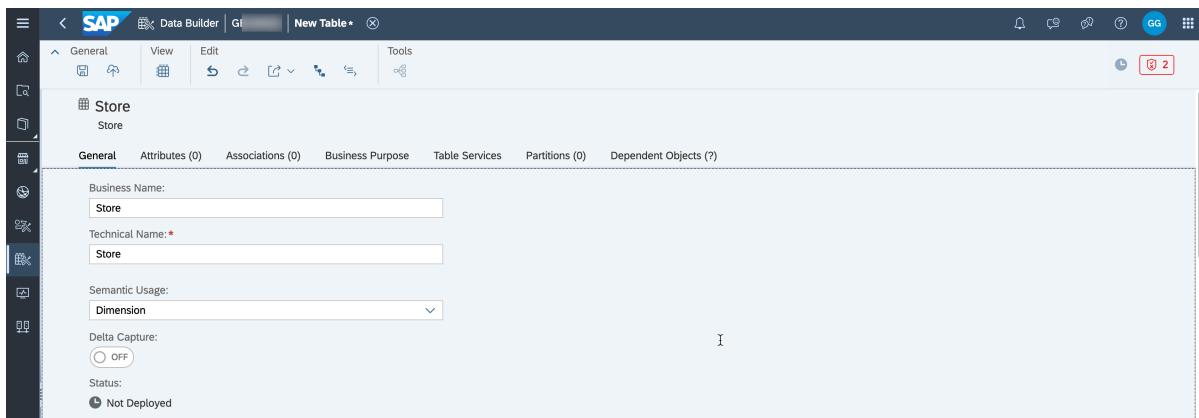


20. You are back at the home screen of the Data Builder and you should see your table in the list of objects.

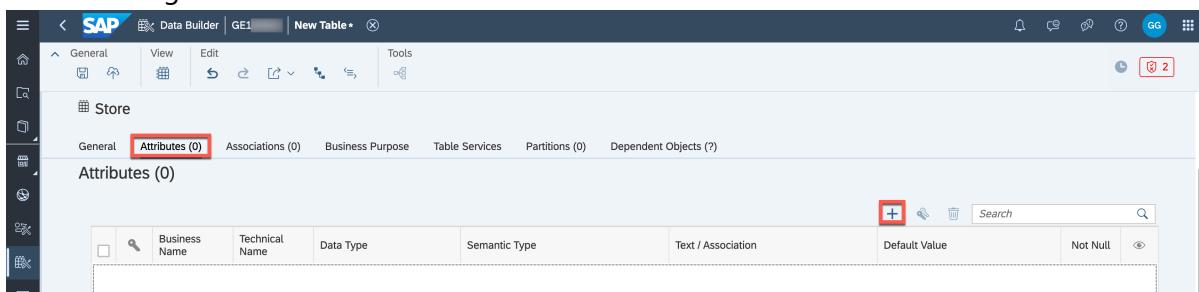
21. Use the option **New Table**.

22. Enter the following details:

- Business Name - Store
- Technical Name - Store
- Semantic Usage - Dimension



- Navigate to the **Attributes** area.
- Use the "+" sign to create new attributes.



Tip: You will notice, based on the Type Dimension, we now have two additional options for each Column: Semantic Type and Text/Association.

The column Text/Association allows you to specify the purpose of an attribute when your entity is consumed later. For example, you could have a Product ID and a Product Description in the table and use the Product Description column as text for the Product ID.

The Semantic Type option provides you with several options to choose from, so that you can configure an additional context for the column, such as the option to configure the column as a Currency column or a Language column.

25. Enter the following **Attributes** for the table:

Key:	Business Name:	Technical Name:	Data Type:	Semantic Type	Text/Association
X	Store ID	Store_ID	String (6)	None	Store Name
	Store Name	Store_Name	String (30)	Text	

Key:	Business Name:	Technical Name:	Data Type:	Semantic Type	Text/Association
	Store City	Store_City	String (20)		
	State ID	State_ID	String (2)		
	State Name	State_Name	String(30)		
	Country	Country	String(30)		
	Latitude	Latitude	Decimal(15,8)		
	Longitude	Longitude	Decimal(15,8)		

 **Tip:** Please note, that you can only select the Store Name for the column **Text/Association** after you entered the details for the Store Name into **Attributes** and select "Text" as the Semantic Type.

26. After you entered all columns for the table, ensure you enable the Key Column option for the column Store ID.
27. Click **Save** in the General menu.
28. Select the folder **TECHED2024-DA180**. You will be asked to confirm the business name as well as the technical name.
29. Click **Save**.
30. Click **Deploy** in the General Menu.
31. Navigate back to the list of tables.
32. Create a new table.
33. Enter the following details:
 - Business Name - Product
 - Technical Name - Product
 - Semantic Usage - Dimension
 - Navigate to the **Attributes** area.
 - Enter the following attributes for the table:

Key:	Business Name:	Technical Name:	Data Type:	Semantic Type	Text/Association
X	Product ID	Product_ID	String (4)	None	Product Name
	Product Name	Product_Name	String (30)	Text	

Key:	Business Name:	Technical Name:	Data Type:	Semantic Type	Text/Association
	Product Category ID	Product_Category_ID	String (4)	None	Product Category Name
	Product Category Name	Product_Category_Name	String (30)	Text	

36. After you entered all columns for the table, ensure you enable the Key Column option for the column Product ID.

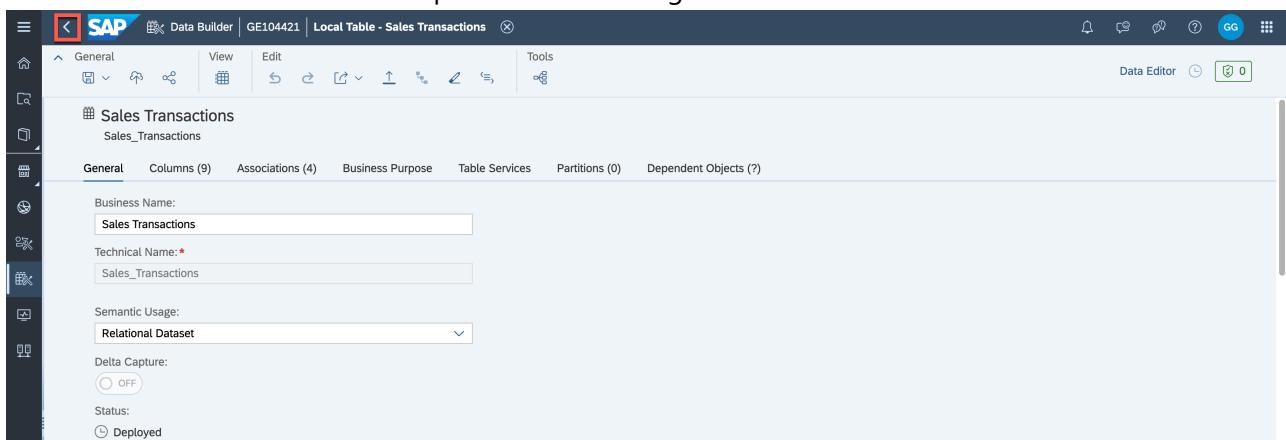
37. Click **Save** in the General menu.

38. Select the folder **TECHED2024-DA180**. You will be asked to confirm the business name as well as the technical name.

39. Click **Save**.

40. Click **Deploy** in the General Menu.

41. Click on the return arrow on the top menu bar to navigate back to the list of tables.



42. Use the option **New Table**.

43. Enter the following details:

- Business Name - Sales Manager
- Technical Name - Sales_Manager
- Semantic Usage - Dimension

- Navigate to the **Attributes** area.
- Enter the following attributes for the table:

Key:	Business Name:	Technical Name:	Data Type:	Semantic Type	Text/Association
X	Sales Manager ID	Sales_Manager_ID	String (4)	None	Sales Manager Name

Key:	Business Name:	Technical Name:	Data Type:	Semantic Type	Text/Association
Sales Manager Name		Sales_Manager_Name	String (30)	Text	

46. After you entered all columns for the table, ensure you enable the Key Column option for the column Sales Manager ID.
47. Click **Save** in the General menu.
48. Select the folder **TECHED2024-DA180**. You will be asked to confirm the business name as well as the Technical Name.
49. Click **Save**.
50. Click **Deploy** in the General Menu.
51. Use the Back option in top menu or as an alternative you can click on your Space name to navigate back to the home screen of the Data Builder.

Summary

You have now created all tables that we need for our model and the overview in the Data Builder when you filter on tables should look like this:

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 Files Tables Views E/R Models Analytic Models Flows Intelligent Lookups Task Chains Data Access Controls

New Table

Search In: "TECHED2024-DA180"

Business Name	Technical...	Type (Semantic Usage)	Space	Folder	Status	Changed On
Store	Store	Local Table (Dimension)	GE119057	TECHED2024-DA180	Deployed	Sep 25, 2024, 17:32:13
Sales Transactions	Sales_Tra...	Local Table (Relational D...	GE119057	TECHED2024-DA180	Deployed	Sep 25, 2024, 17:42:19
Product	Product	Local Table (Dimension)	GE119057	TECHED2024-DA180	Deployed	Sep 25, 2024, 17:32:13
Sales Manager	Sales_Ma...	Local Table (Dimension)	GE119057	TECHED2024-DA180	Deployed	Sep 25, 2024, 17:32:13

Continue to [Exercise 03 - Creating the Entity Relationship Model](#)

Exercise 3 - Creating the Entity-Relationship Model



Note: This is an Optional Exercise. You need to have finished [Exercise 02](#) before.



Detour: SAP Datasphere - Entity-Relationship Model

Let us clarify what an Entity-Relationship model is and why we are creating it, before we start the exercise.

An Entity-Relationship Model provides a variety of benefits:

- Depict the relationship of different entities
- Design physical or remote database models
- Re-use existing entities (table, view) from Data Builder
- Add new entities on-the-fly
- In-editor real time data preview
- Model Import / Export

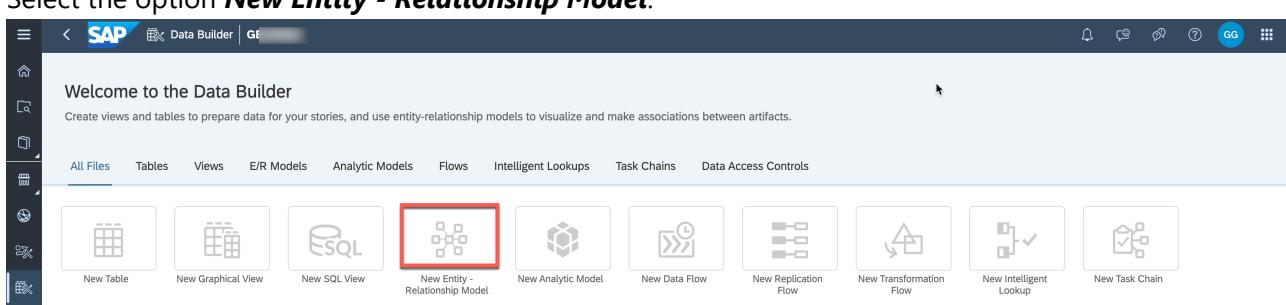
Basically, the Entity-Relationship Model is not a view that you would consume in SAP Analytics Cloud, but instead it represents the relationship between the tables or views, and it helps you to define the relationship once, so that you can reuse those when creating a new view.

Maintenance of associations and other capabilities of the E/R Model (like definition of semantic properties) can also be done in the table/view editor. Rules for creating associations depend on the Semantic Usage. The advantage of the E/R Model is that multiple entities can be modified at the same time while being visualized as diagram.

End of Detour

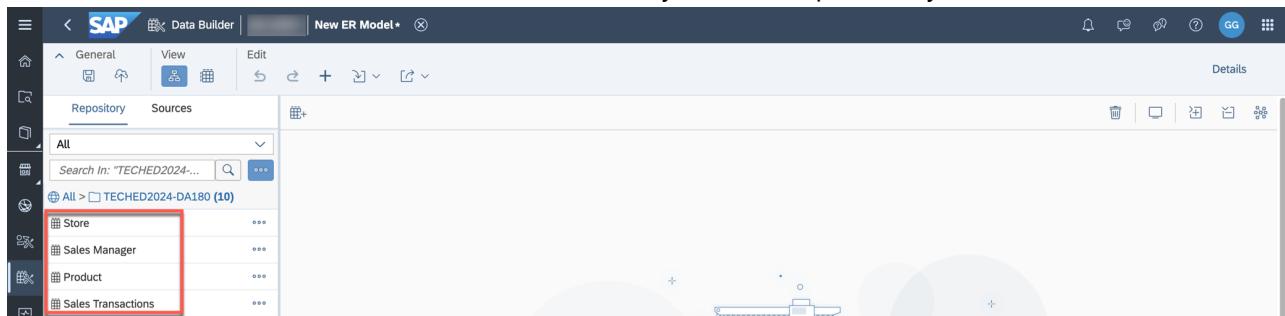
Create a new Entity-Relationship Model

1. Log On to your SAP Datasphere tenant.
2. Select the menu option Data Builder on the left-hand side
3. Select the option **New Entity - Relationship Model**.

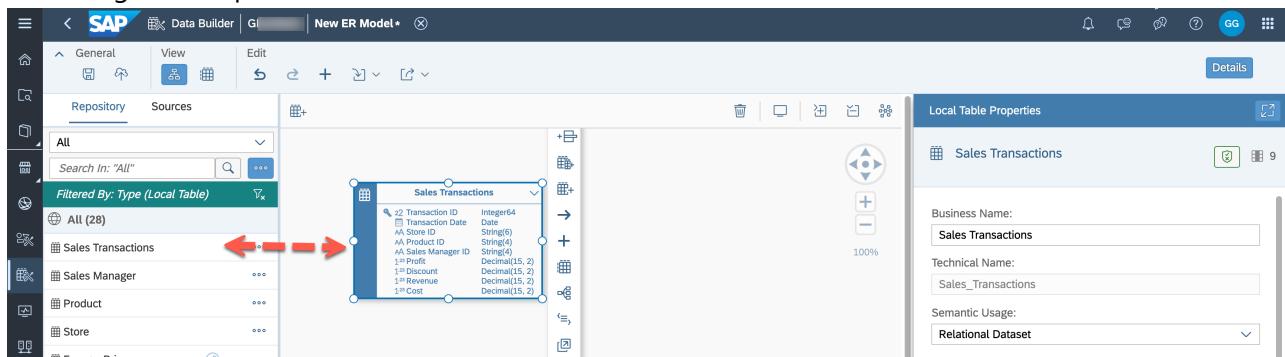


4. On the left-hand side you are presented with the local tables and views from your Repository. You also have the option to bring in tables from any remote sources configured.

5. Ensure to select the option **Repository** and the folder **TECHED2024-DA180**, so that you see the entities available. You should see the four tables that you created previously.



6. Now drag and drop the table **Sales Transactions** to the canvas.

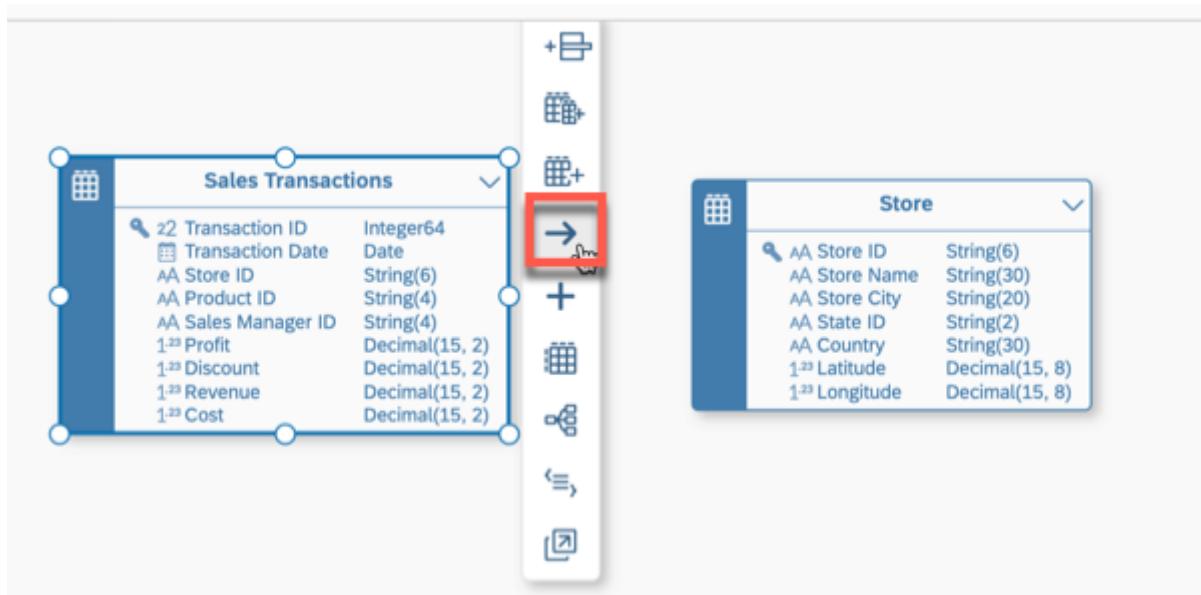


7. When you select the table **Sales Transactions** on the canvas, you are presented with various additional options. You could add a new column, create a new entity (view or table), create a new association, add related entities, preview the data, analyze impact and lineage, modify the custom entity annotations or open the table in the table editor.

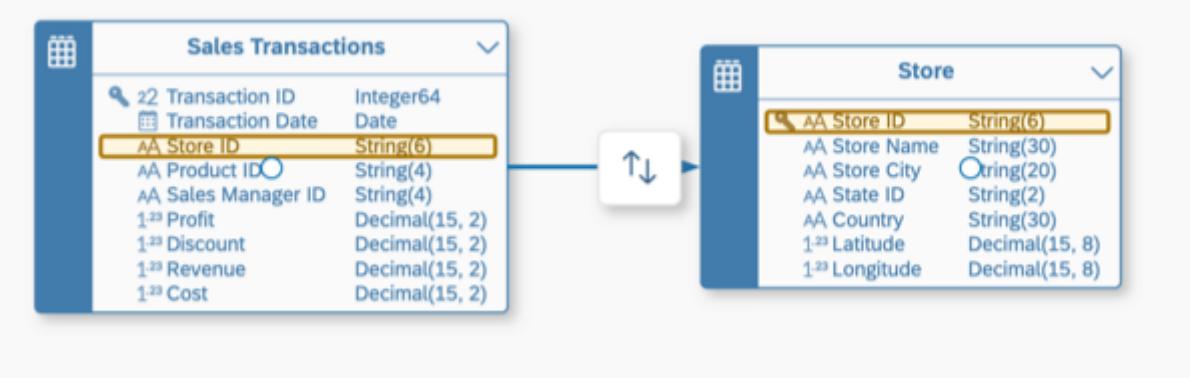
8. Now drag and drop the table **Store** to the canvas next to the table **Sales Transactions**.

9. Select the table **Sales Transactions**.

10. Use the "->" symbol to create a new association by clicking on it and holding onto the icon.



11. Drag the **arrow icon** to table **Store** to create a new association between the table **Sales Transactions** and the table **Store**.



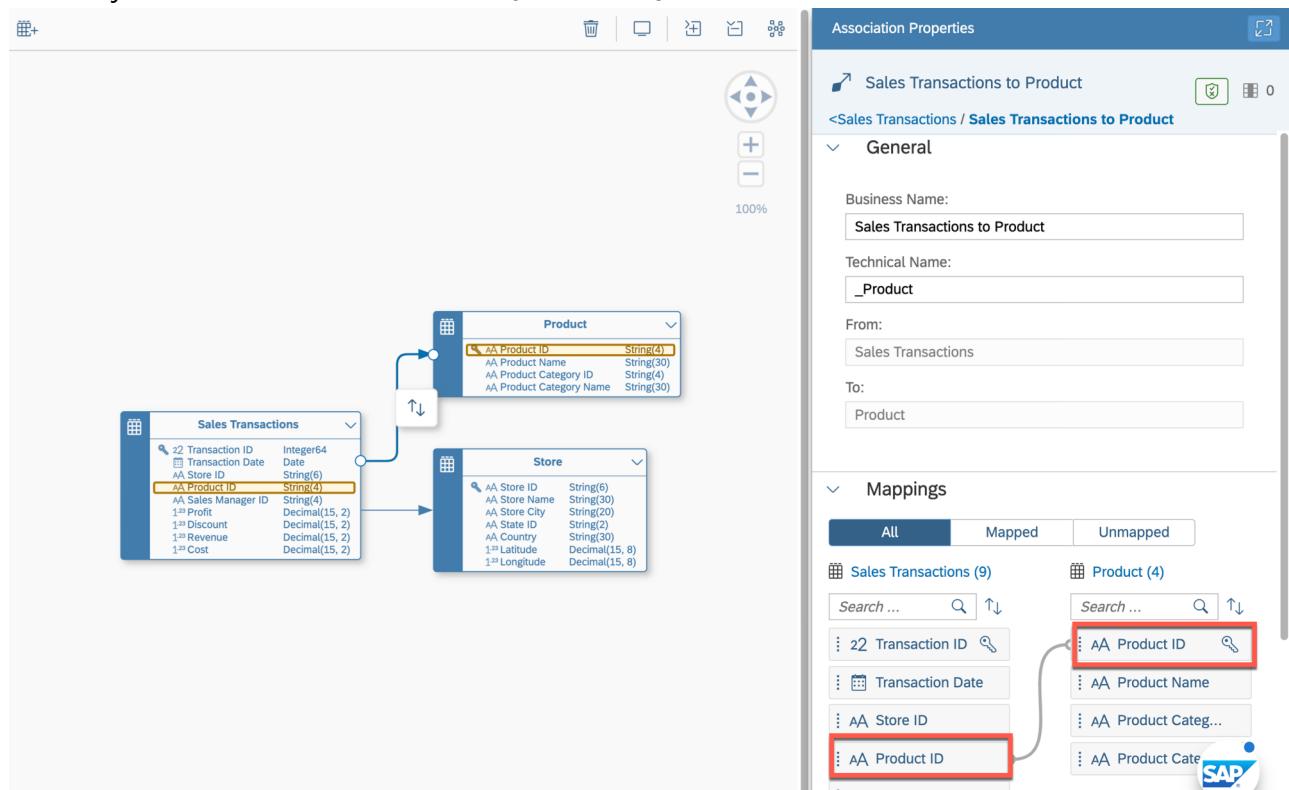
12. The **Association Properties** panel defines the mapping of the columns for the association. Ensure that the join between the table **Sales Transactions** and **Store** is based on column **Store ID (STORE_ID)**. The system suggests this based on matching columns, but you can also add / remove these joins manually.

The screenshot shows the SAP Data Modeler Association Properties panel. The "General" section contains fields for Business Name ("Sales Transactions to Store") and Technical Name ("_Store"). The "From:" field is set to "Sales Transactions" and the "To:" field is set to "Store". The "Mappings" section shows a list of mapped columns between "Sales Transactions (9)" and "Store (7)". The "AA_Store ID" column from "Sales Transactions" is mapped to the "AA_Store ID" column in "Store". A red arrow points to the "AA_Store ID" entry in the Mappings list.

13. Now drag the table **Product** onto the canvas.

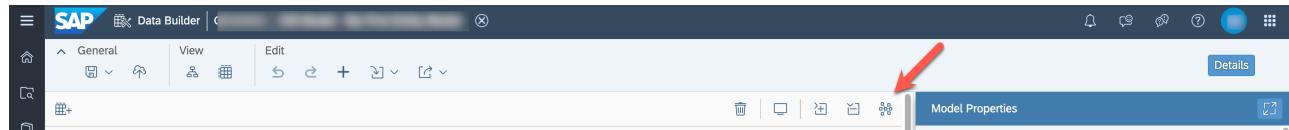
14. Select the table **Sales Transactions** on the canvas. Drag and drop the **arrow symbol** to the table **Products** to create a new association between the table **Sales Transactions** and table **Product**. Ensure

that the join is based on the **Product ID (Product ID)** column.

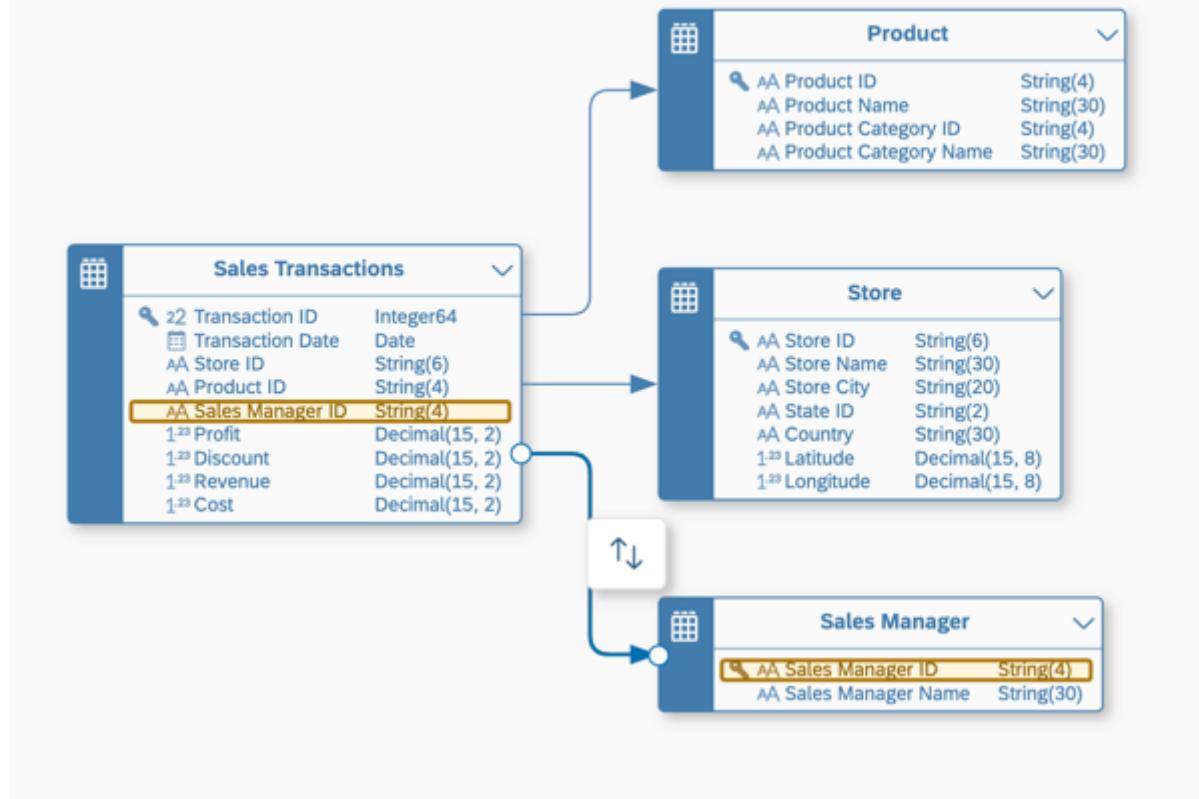


15. Now drag the table **Sales Manager** onto the canvas.

16. You can use the option **Auto Layout** in the toolbar to arrange all entities.



17. Create a new association between the tables **Sales Transactions** and **Sales Manager**. Ensure that the join is based on the Sales Manager ID (Sales Manager ID) column.



18. We want to associate our sales transactions with the time data created in the space so that we can later drill down and up in hierarchies (for example setting the granularity to day or quarter). On the left hand side, search for "Time Dimension Day". Select the dimension of your space (not the one shared). You can verify that by clicking the three dots ... and **Show Info** afterward.

The screenshot shows the SAP Data Builder interface with the following details:

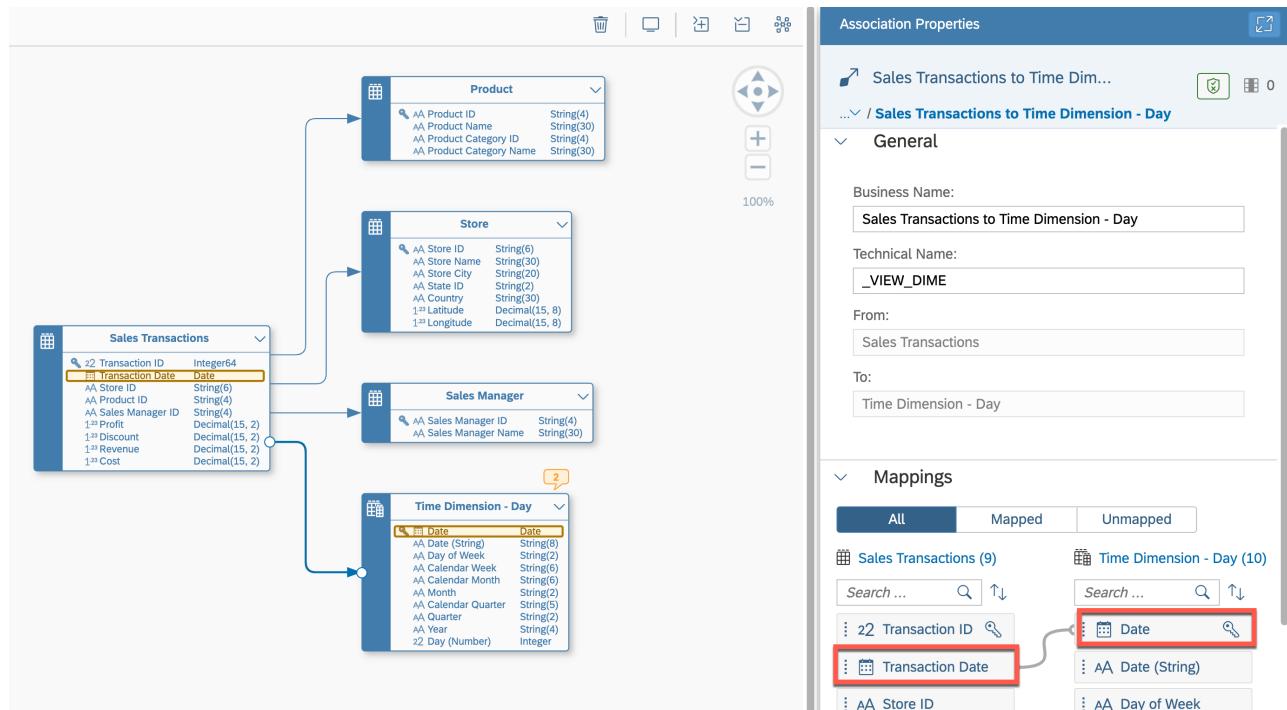
- Left Sidebar:** Shows a tree view with a red arrow pointing to the "Sharing" icon (a person icon).
- Central View:**
 - Sharing Status:** The "Sharing" section shows "Not Shared" (highlighted with a red box).
 - View Details:** Technical Name: SAP.TIME.VIEW_DIMENSION_DAY, Semantic Usage: Dimension, Type: View, Status: Deployed.
 - Associated Entities:** A diagram on the right shows associations from the Sales Transactions view to Product, Store, and Sales Manager entities.

19. Drag the identified view **Time Dimension – Day** to the canvas.

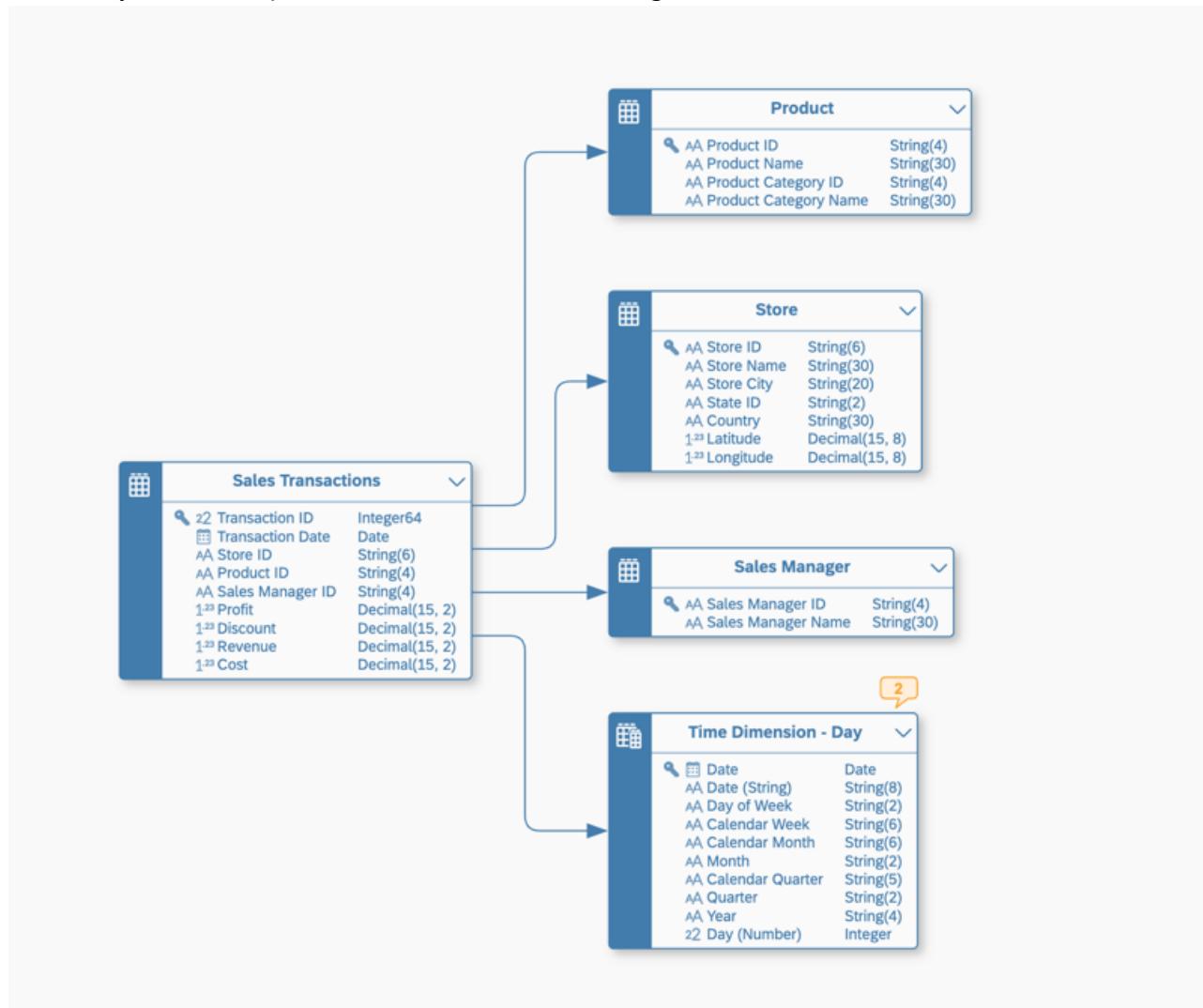
20. Select the table **Sales Transactions** on the canvas.

21. Select the **arrow symbol** and drag and drop the arrow to the view **Time Dimension - Day** to create a new association between the table **Sales Transactions** and the view **Time Dimension - Day**.

22. In the panel on the right-hand side (section **Mappings**), create a join between the columns **Transaction Date** and **Date**.



23. Your Entity-Relationship Model should look like the image shown below.



24. Save the changes to your Entity-Relationship Model by using the **Save option** in the General menu.

25. Enter the following details:

- Business Name - Sales Model - Entity-Relationship Model
- Technical Name - Sales_ER_Model
- Click **Save** and **Save Anyway** (the warning is not relevant).
- Deploy the model.

Summary

You've now created the Entity-Relationship Model to arrange the tables and the time dimension view in relation to one another. You could have also defined the associations in the individual table/view editors, but the E/R Model provides the advantage modifying multiple entities at the same time and seeing them in relation to each other. We will make use of the defined relations the next steps.

Continue to - [Exercise 05: Uploading Data](#)

Exercise 4 - Importing Tables

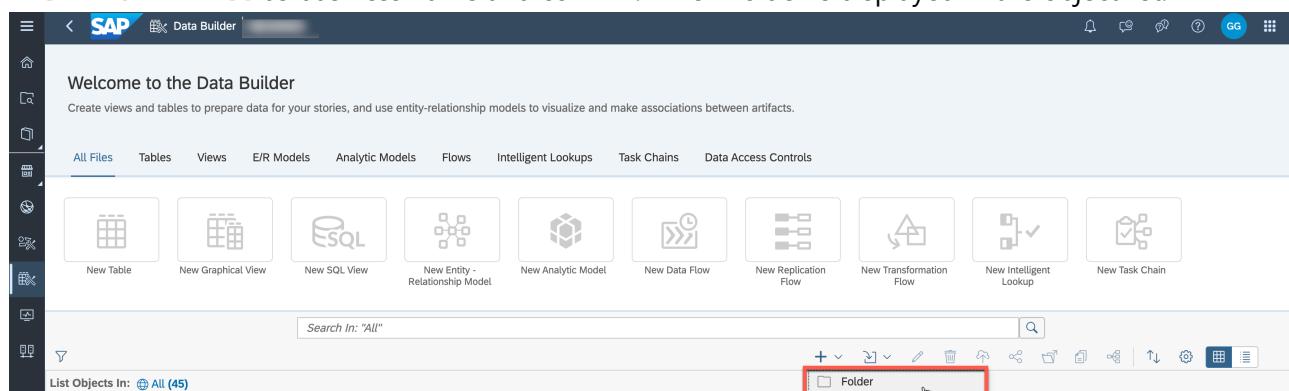
Important:

This exercise is only required in case you **did not** create the tables in [Exercise 02 - Prepare Your Data \(optional\)](#) and the Entity-Relationship Model for our sample model manually in [Exercise 03 - Creating the Entity-Relationship Model \(optional\)](#) and prefer to import them.

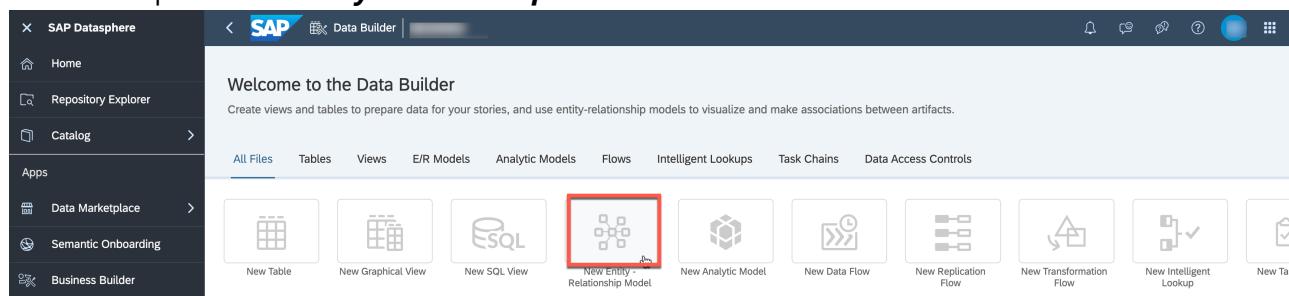
The file for this exercise is [Sales_ER_Model.json](#) and the file is part of the [ZIP file](#) you downloaded.

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

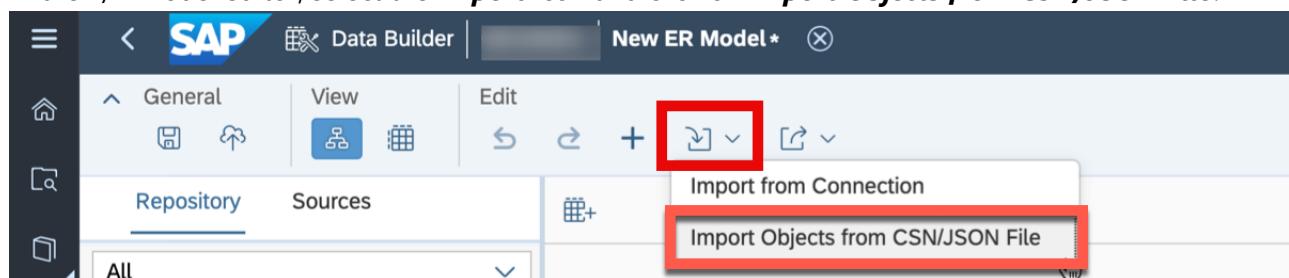
1. Log On to your SAP Datasphere tenant.
2. Select the menu option **Data Builder** on the left-hand side.
3. Create a new folder to organize the entities you will create in this jump-start session. Enter [TECHED2024-DA180](#) as business name and confirm. A new folder is displayed in the object list.



4. Select the option **New Entity-Relationship Model**.



5. In the E/R Model editor, select the **Import** icon and click on **Import Objects from CSN/JSON File**.



6. Select the file "**Sales_ER_Model.json**"

7. Click **Next**.

8. Select all entities for import.

Select Objects to Import			
	Technical Name	Type (Semantic Usage)	Status
<input checked="" type="checkbox"/>	Business Name	Product	Local Table (Dimension)
<input checked="" type="checkbox"/>	Product	Sales_Manager	Local Table (Dimension)
<input checked="" type="checkbox"/>	Sales Manager	Sales_Transactions	Local Table
<input checked="" type="checkbox"/>	Sales Transactions	Store	Local Table (Dimension)
<input checked="" type="checkbox"/>	Store		Ready to Import

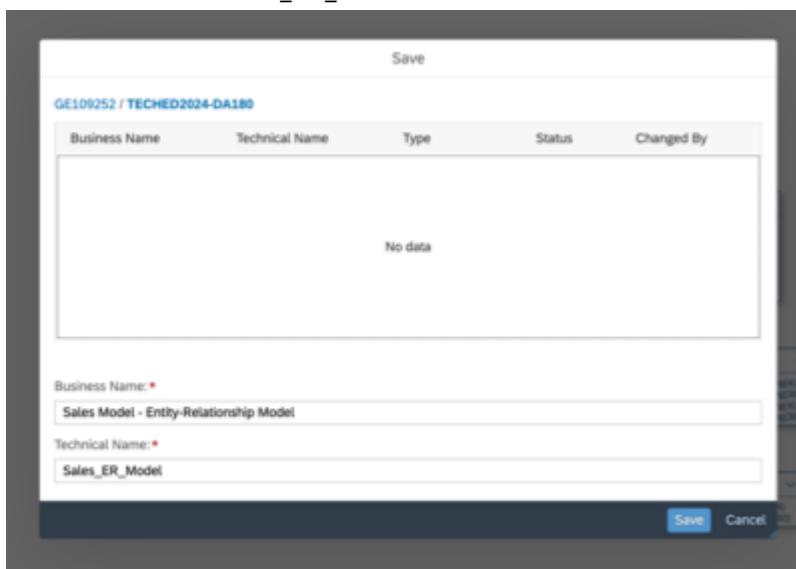
9. Click **Import CSN File**.

10. You will receive a message about the import being completed and the tables will be shown on the canvas arranged as an Entity-Relationship Model.

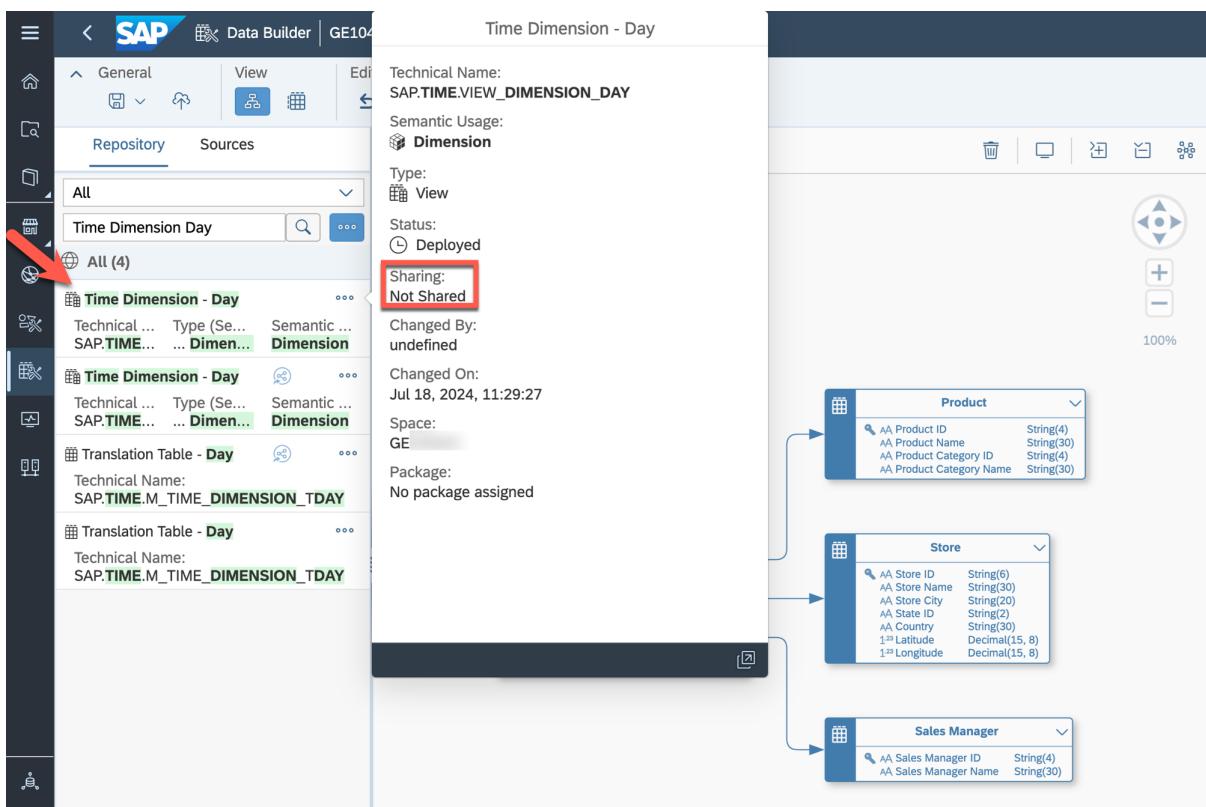
11. Save the changes to your Entity Relationship model using the **Save** option in the upper left corner.

12. Select the folder **TECHED2024-DA180** and enter the following details.

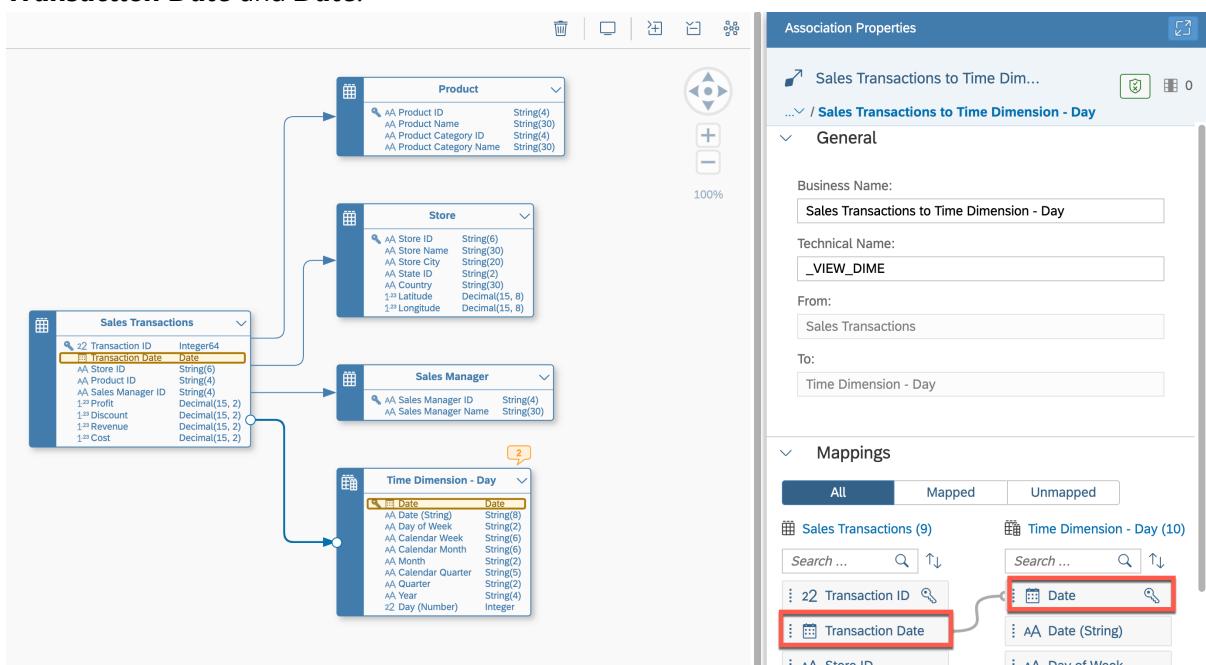
- Business Name: **Sales Model - Entity-Relationship Model**
- Technical Name: **Sales_ER_Model**



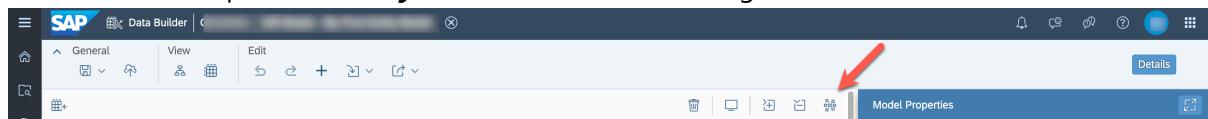
- Click **Save**.
- We want to associate our sales transactions with the time data created in the space so that we can later drill down and up in hierarchies (for example, setting the granularity to day or quarter). On the left-hand side, search for **Time Dimension - Day**. Select the dimension from your space (the one that is **Not Shared**). You can verify this by clicking the three dots (...) and then selecting **Show Info** afterward.



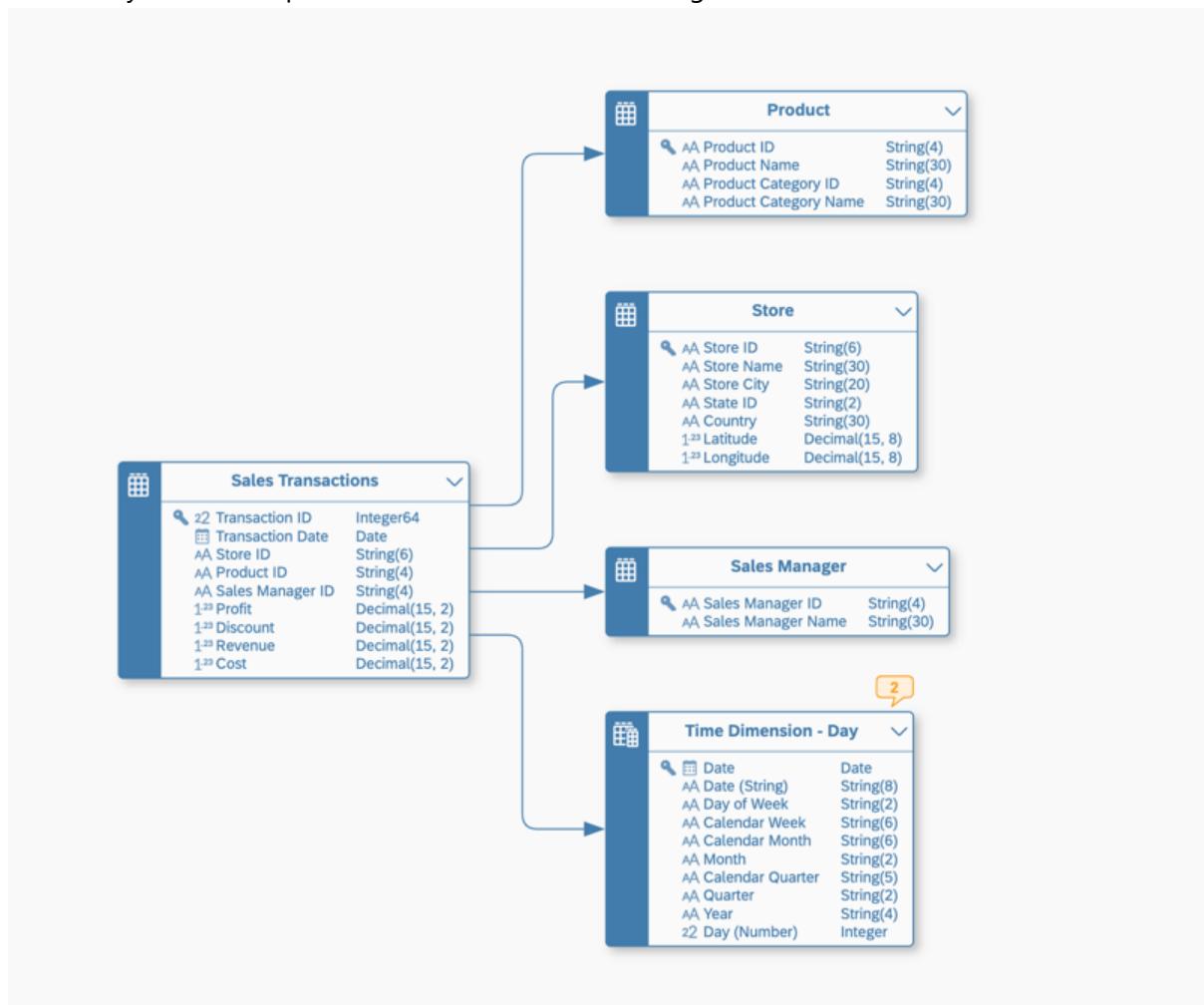
- Drag the identified view **Time Dimension – Day** to the canvas.
- Select the table **Sales Transactions** on the canvas.
- Select the **arrow symbol** and drag and drop the arrow to the view **Time Dimension – Day** to create a new association between the table **Sales Transactions** and the view **Time Dimension – Day**.
- In the panel on the right-hand side (section **Mappings**), create a mapping between the columns **Transaction Date** and **Date**.



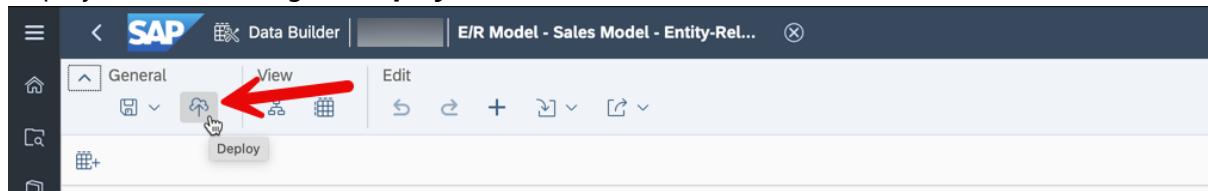
- You can use the option **Auto Layout** in the toolbar to arrange all entities.



- Your Entity-Relationship Model should look like the image shown below.



- Save the changes to your Entity-Relationship Model by using the **Save option** in the **General** menu. Select **Save Anyway** (the warning is not relevant).
- Deploy the model using the **Deploy** icon in the menu bar.



Summary

As part of the deployment of the Entity-Relationship Model, the underlying tables are also imported and deployed.

When you navigate to the **Data Builder** screen, you should now see:

- A set of four tables (three dimensions and one relational dataset)
- One Entity-Relationship Model

Continue with [Exercise 05: Uploading Data](#)

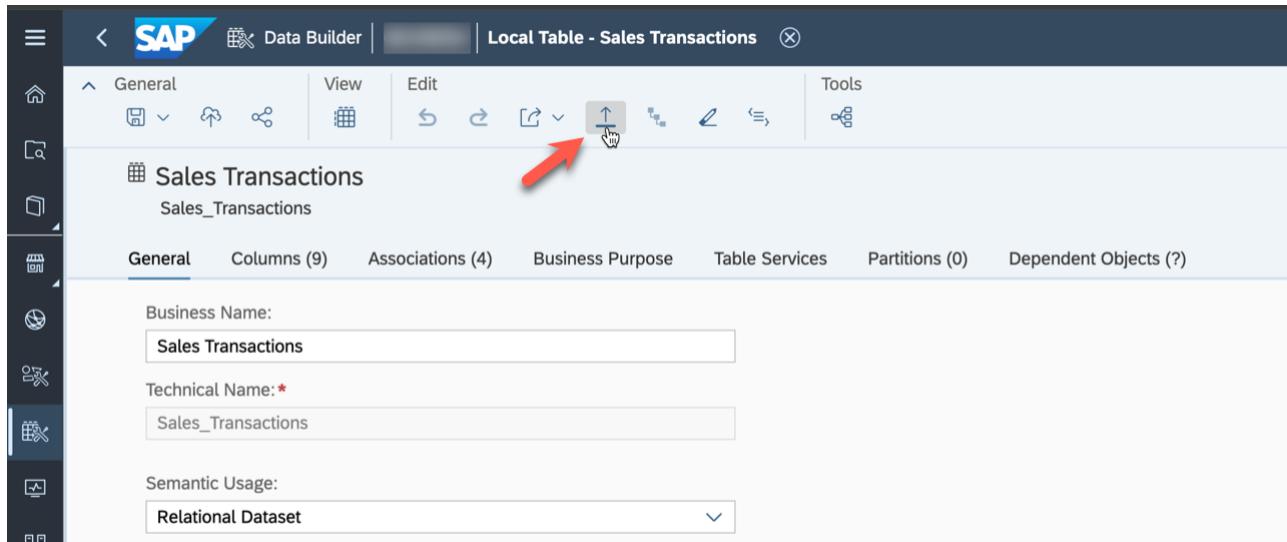
Exercise 5 - Uploading Data

In this exercise, we will populate the tables created in the previous steps, ensuring we have data for analysis. These exercises use data from CSV files to allow for easy repetition in any SAP Datasphere tenant.

To explore the various connectivity options of SAP Datasphere, refer to [the details available here](#) or explore the connections available in the app **Connections** (click on the corresponding icon on the left side).

The files for this exercise are part of the ZIP file you downloaded at the beginning. We will now upload the CSV files into the corresponding tables.

1. In your SAP Datasphere tenant select the menu option **Data Builder** on the left-hand side.
2. Click on your folder **TECHED2024-DA180**.
3. Click on the table **Sales Transactions**.
4. Select the menu option **Upload Data from CSV File** in the toolbar.



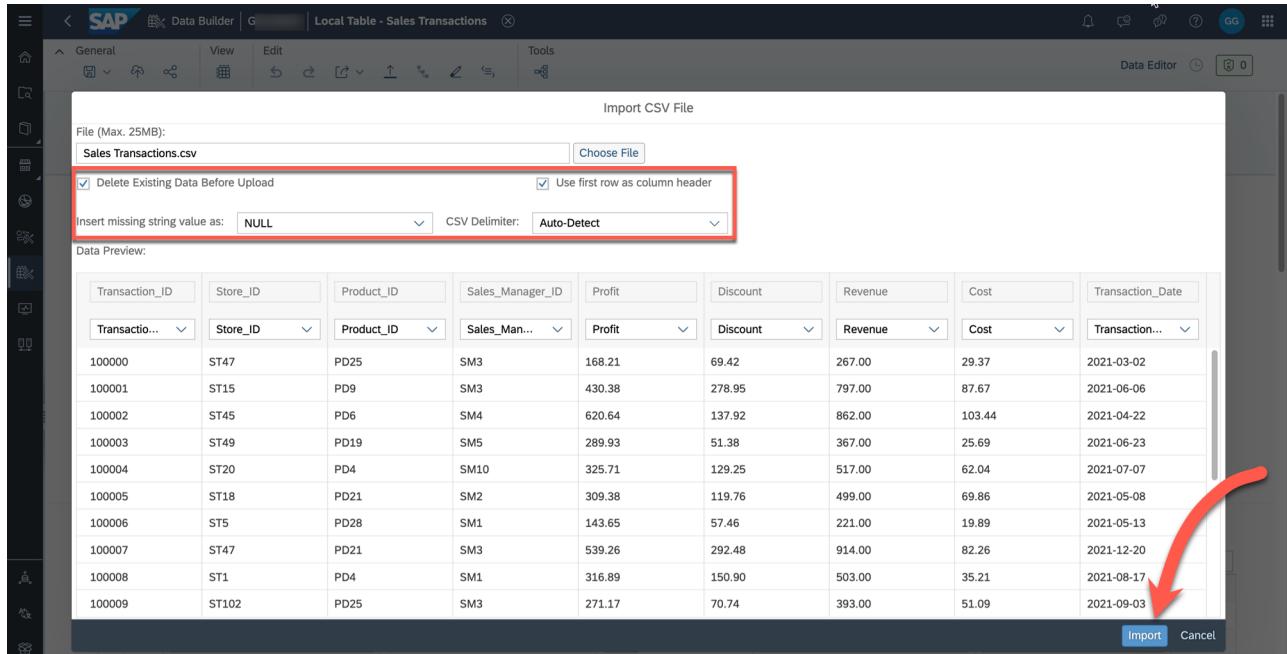
5. Click **Choose File**.
6. Navigate to where you unzipped the download.
7. Select the file **Sales Transactions.csv**.

Name	Date modified	Type	Size
Dimension Product.csv		Microsoft Excel Com...	1 KB
Dimension Store.csv		Microsoft Excel Com...	16 KB
Dimension Sales Manager.csv		Microsoft Excel Com...	1 KB
Sales Transactions.csv		Microsoft Excel Com...	5,806 KB

A red arrow points to the 'Sales Transactions.csv' file in the list.
8. Ensure the option **Delete Existing Data Before Upload** checkbox is ticked.
9. Ensure the option **Use first row as column header** checkbox is ticked.
10. Ensure the option **Insert missing string value as** is set to **NULL**.
11. Ensure the **CSV Delimiter** option is set to **Auto-Detect**.

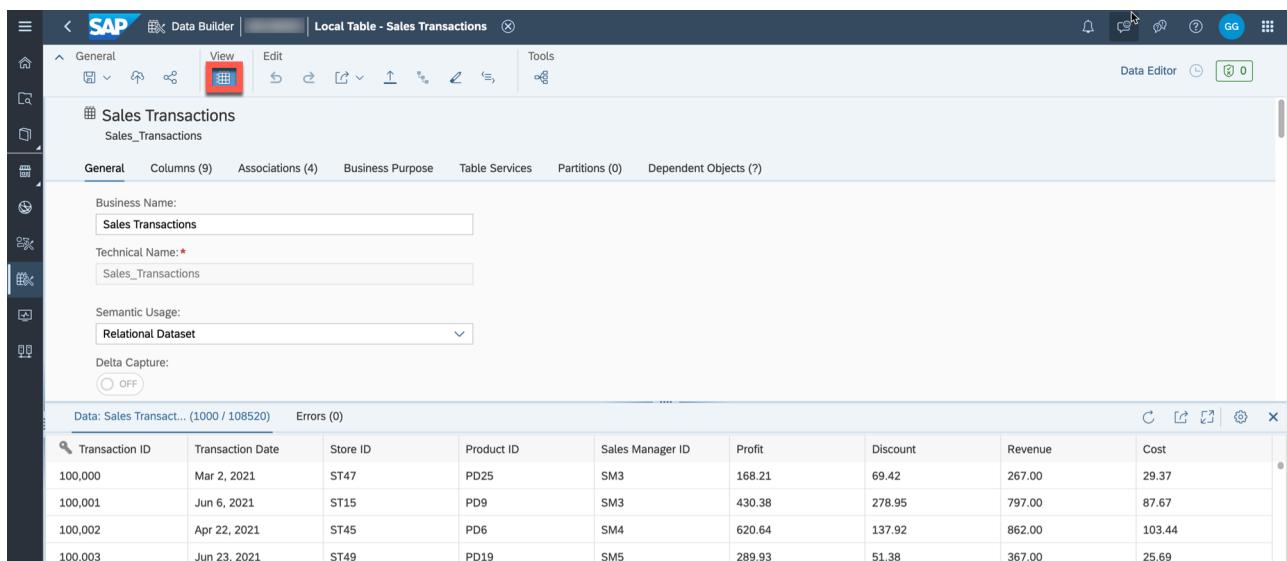
12. Ensure that all columns of the table have a mapped column from the CSV file.

13. Click **Import**.



14. You should receive a message about the successful import of the information.

15. You can use the **Data Preview** to check the data loaded into the table.



16. There is no need to save / deploy the table after you imported the data.

17. Continue and repeat steps 4–15 for the other tables. Each table has a corresponding CSV file included in the download.

Upload the data for the following tables:

- **Product** ([Dimension Product.csv](#))
- **Store** ([Dimension Store.csv](#))
- **Sales Manager** ([Dimension Sales Manager.csv](#))

Summary

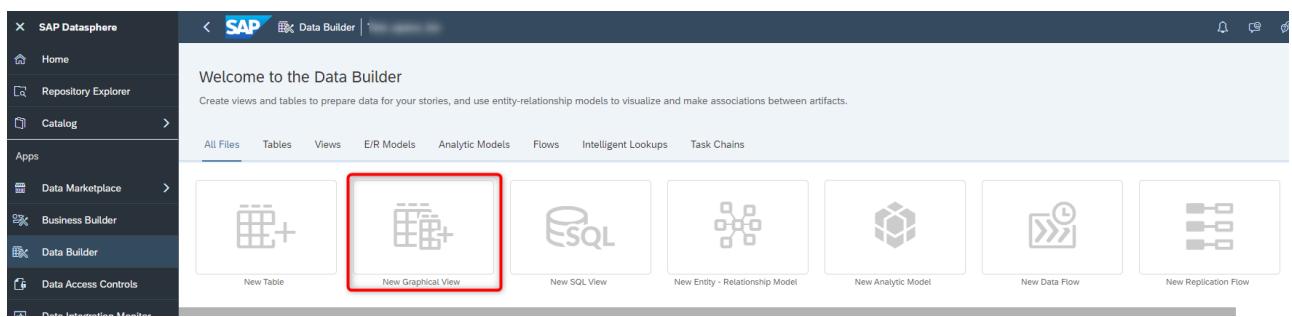
Now you have imported all the data files and you can start to build the data model.

Continue to - [Exercise 06: Creating the Dimension](#)

Exercise 6 - Creating the Dimension

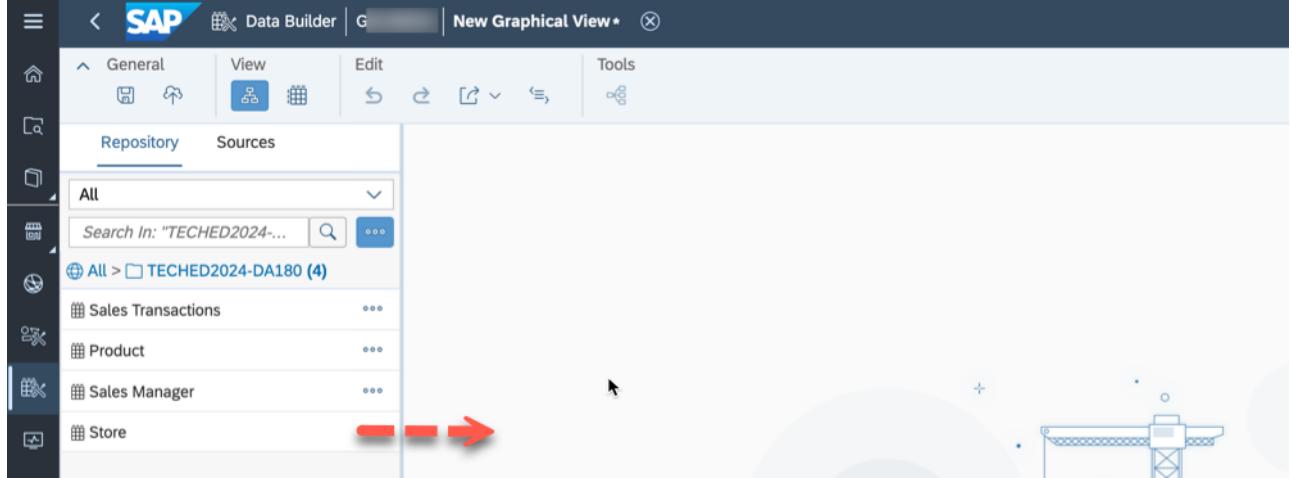
In this exercise, we will create a new view of the semantic type dimension based on our previously created table. We will enhance this data by configuring a geographic enrichment so that we can visualize the store location on a geo map later in SAP Analytics Cloud.

1. Log On to your SAP Datasphere tenant.
2. Select the menu option **Data Builder** on the left-hand side.
3. Click **New Graphical View**.

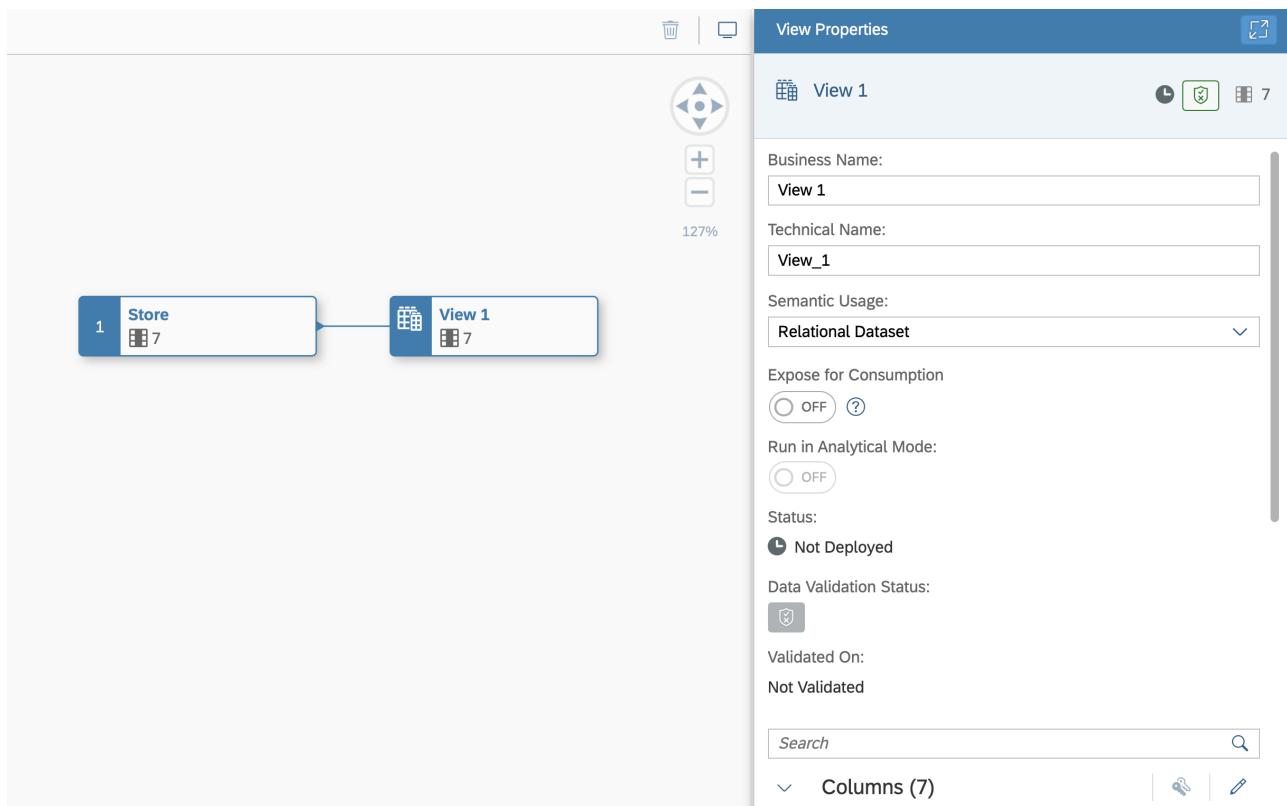


4. On the left-hand side you can decide between:
 - **Repository:** Here you have access to the local tables (imported data), Views, Intelligent Lookups and Shared Objects.
 - **Sources:** Here you have access to respective objects from your connections.
5. Ensure you select the option **Repository** and open the folder **TECHED2024-DA180**.

6. You are presented with the list of tables, which we created previously. Drag and drop the table **Store** to the canvas.

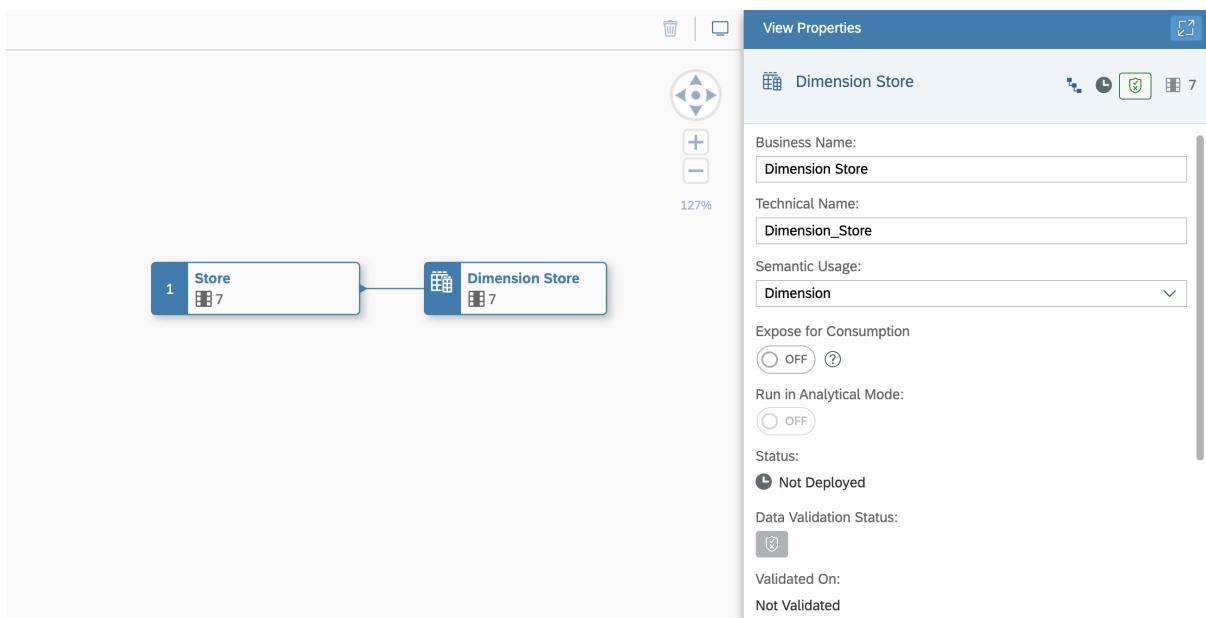


7. You automatically will – in addition to the table you dragged to the canvas – receive the output view, in the screenshot called **View 1**:
8. Navigate to the **View Properties** window by clicking on the **View 1** in the canvas.



9. Here you can configure properties for the view:

- **Business Name:** Dimension Store
- **Technical Name:** Dimension_Store
- **Semantic Usage:** Dimension



10. Now click on the table **Store** on the canvas.

11. When you select the table on the canvas, you have the following options (top to bottom)

- You can add filter on top of the source entity.

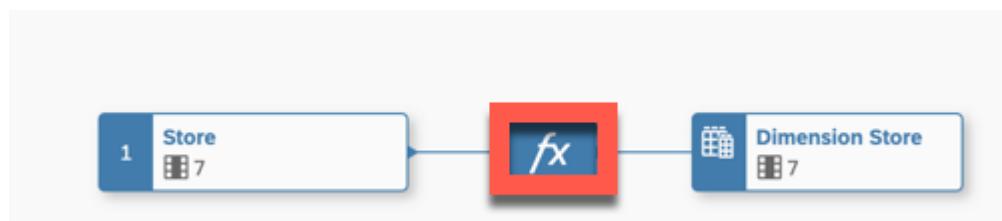
- You can rename or hide columns as part of a projection.
- You can add Calculated Columns.
- You can add an Aggregation View.
- You can add additional tables / views based on suggested joins, which are based on your Entity Relationship model.
- You can preview the data.
- You can open the Impact & Lineage Analysis.
- You can open the table in the editor in another browser tab.

12. Use the option to add a new calculated column. This option also includes the ability to configure the



geographic enrichment.

13. Now click on the new node **fx** on the canvas.



14. Navigate to the properties on the right-hand side.

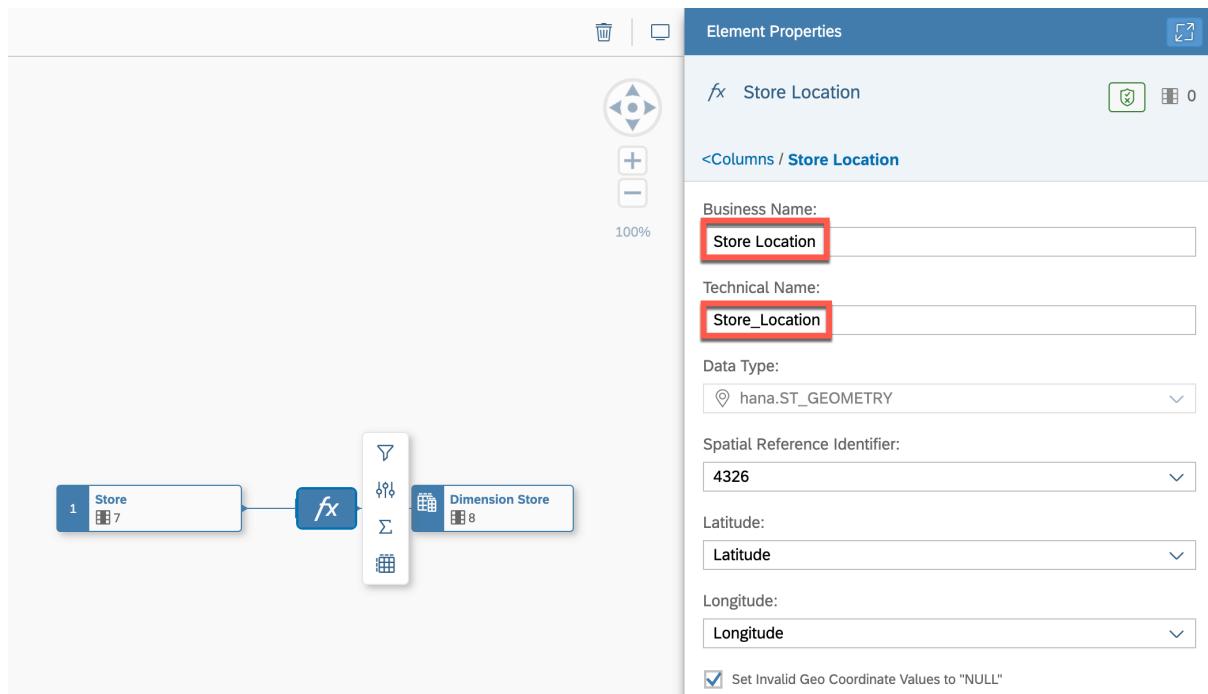
15. Click on the + sign and select the option **Geo-Coordinates Column**.

The screenshot shows the SAP Data Builder interface with the following details:

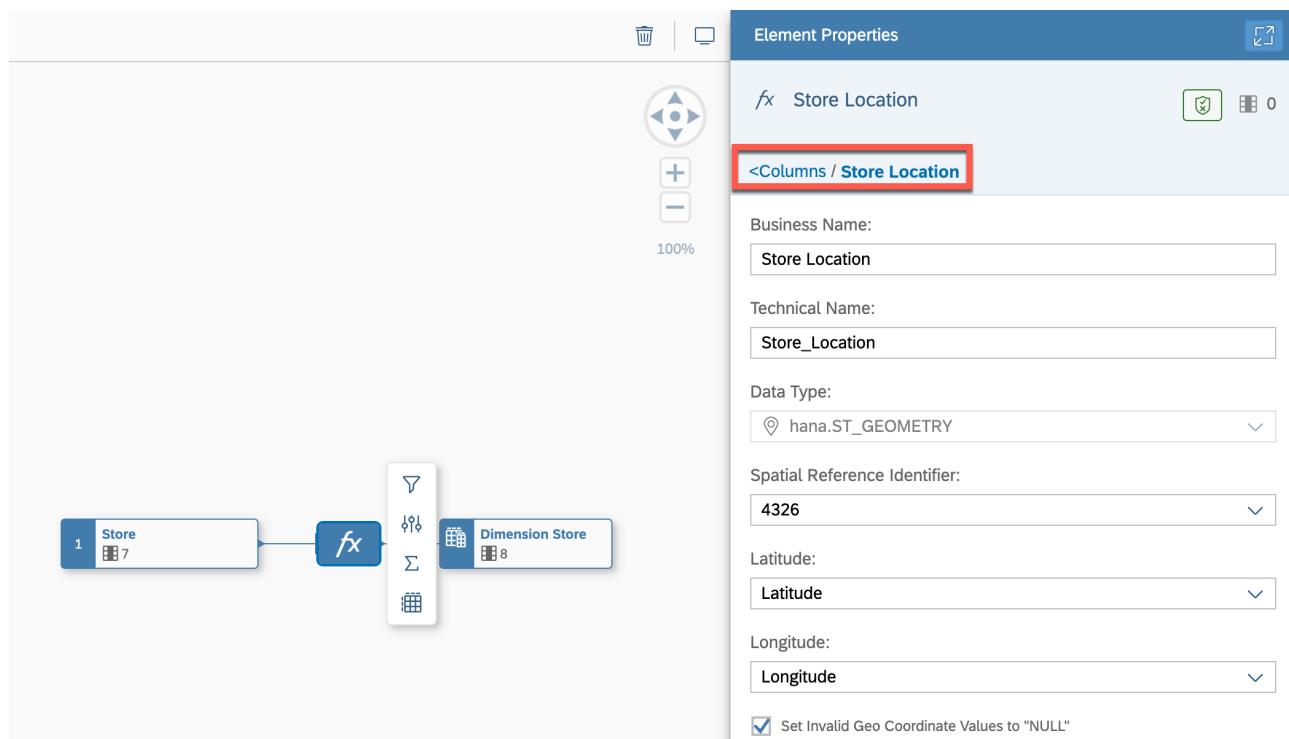
- Toolbar:** General, View, Edit, Tools.
- Repository:** Shows a tree structure with 'All', 'All > TECHED2024-DA180 (4)', 'Sales Transactions', 'Product', 'Sales Manager', and 'Store'.
- Sources:** Shows a connection between a 'Store' node (1) and a 'Dimension Store' node.
- Properties Panel:** 'CalculatedElements Properties' panel is open, showing 'Calculated Columns 1' with a list of columns:
 - Calculated Column (highlighted)
 - Geo-Coordinates Column (highlighted with a red box)
 - Currency Conversion Column

16. You are presented with the properties for the new column. Configure the following details:

- **Business Name:** Store Location
- **Technical Name:** Store_Location
- **Latitude:** Latitude
- **Longitude:** Longitude



17. After you configured the details, click on the <Columns / Store Location option in the properties window to go back.

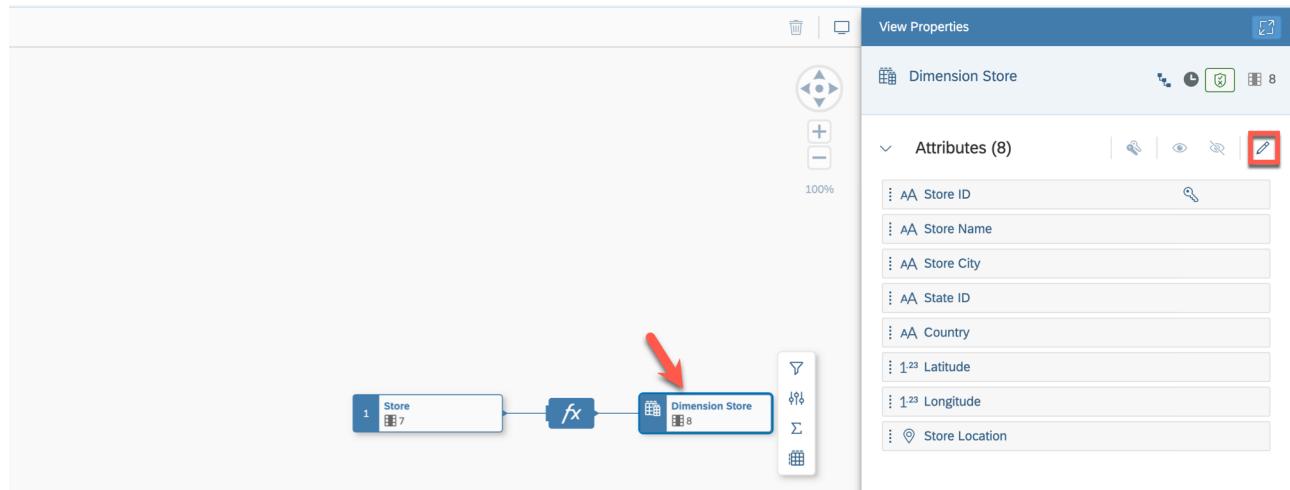


18. Select the final output node for the graphical view of semantic type dimension.

19. Navigate to the properties on the right-hand side.

20. Navigate to the **Attributes** area.

21. Use the pencil icon (top right area) to open the details for the Attributes.



22. Ensure that the semantic type for the attribute **Store Name** is set to **Text**.

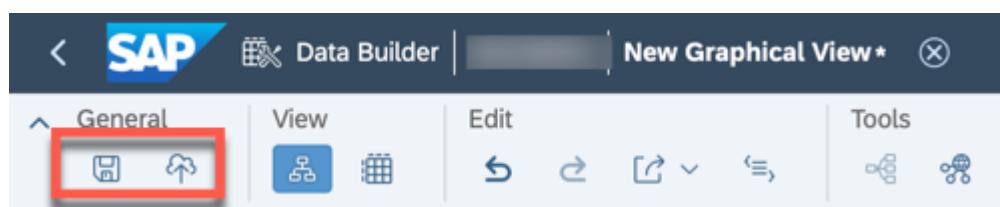
23. Set the **Label (Text / Association)** column for the line item **Store ID** to **Store Name**.

Attributes (9)					Search
Business Name	Technical Name	Data Type	Semantic Type	Label Column	
<input checked="" type="checkbox"/> Store ID	Store_ID	String(6)	None	Store Name	<input checked="" type="checkbox"/>
<input type="checkbox"/> Store Name	Store_Name	String(30)	Text	Store Name	<input checked="" type="checkbox"/>
<input type="checkbox"/> Store City	Store_City	String(20)	None		<input checked="" type="checkbox"/>
<input type="checkbox"/> State ID	State_ID	String(2)	None		<input checked="" type="checkbox"/>
<input type="checkbox"/> State Name	State_Name	String(30)	None		<input checked="" type="checkbox"/>
<input type="checkbox"/> Country	Country	String(30)	None		<input checked="" type="checkbox"/>
<input type="checkbox"/> Latitude	Latitude	Decimal(15, 8)	None		<input checked="" type="checkbox"/>
<input type="checkbox"/> Longitude	Longitude	Decimal(15, 8)	None		<input checked="" type="checkbox"/>
<input type="checkbox"/> Store Location	Store_Location	hana.ST_GeOMETRY(4326)	None		<input checked="" type="checkbox"/>

24. Click **Close**.

25. In the toolbar in the **General** menu, use the option to save your view in the folder **TECHED2024-DA180**.

26. After saving your dimension view, ensure you deploy the view.



Summary

You've now created your dimension view **Dimension Store**, including the added geo-coordinates column **Store Location**, which is required for using visualizations on geo maps within SAP Analytics Cloud.

Continue to - [Exercise 07: Creating the View](#)

Exercise 7 - Creating the View & Analytic Model



Detour: SAP Datasphere - Views & Analytic Models

A view in SAP Datasphere provides you with several benefits:

- No-code/low-code using the graphical editor as well as SQL and [SQLscript](#) editor,
- Foundation of data modeling on top of remote or local tables or views,
- Choose the semantic usage (e.g., Fact or Dimension) - enabling specific functionality and behavior depending on the semantic usage chosen,
- Define unions and joins, rename and remove columns, add calculations and filters.

A view in SAP Datasphere allows you to leverage local tables, remote tables, or views and combine those into a new view. Transformations defined in a Graphical or SQL View are executed during runtime when a view is accessed, for example, via the data preview option or as part of an Analytic Model. By default, views are not persisted, but there is the option to run and schedule snapshot replications.

Applying transformations to data is also possible in the entity **Transformation Flow**. In contrast to views, transformation flows load data from one or more sources and persist the result in a target table. This integration entity can detect delta changes when reading data from a local table that is enabled for delta.

An Analytic Model in SAP Datasphere provides the following benefits:

- Builds the analytical foundation to make data ready for consumption in SAP Analytics Cloud or exposure through the public OData API,
- Allows multi-dimensional and rich analytical modeling,
- Provides data for analytical purposes to answer different business questions by reusing predefined measures, hierarchies, filters, input parameters, and associations,
- Offers many features like calculated & restricted measures, exception aggregation, non-cumulative measures, pruning, variable support, analytical preview, multi-lingual metadata, etc.

The sources for analytic models are views or tables of the semantic type **Fact**, which can contain dimensions, texts, and hierarchies.

End of Detour

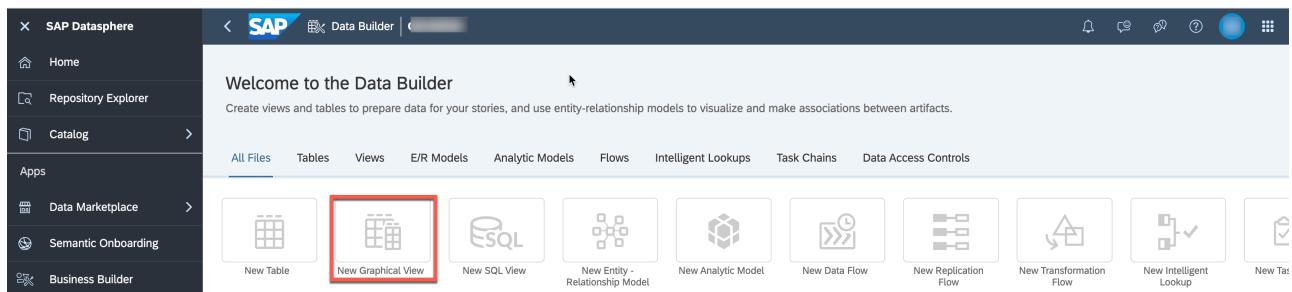
Start of the Exercise

In this exercise, we will create a view for the sales transaction table, which will serve as the base for the Analytic Model that can then be consumed in SAP Analytics Cloud.

The advantage of defining a view as a wrapper for the source table is that the data model becomes more flexible for future changes. By decoupling the view containing the semantics from the source object, it becomes possible to switch the source of the view from a local table to a remote table from a remote system.

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

3. Click **New Graphical View**.

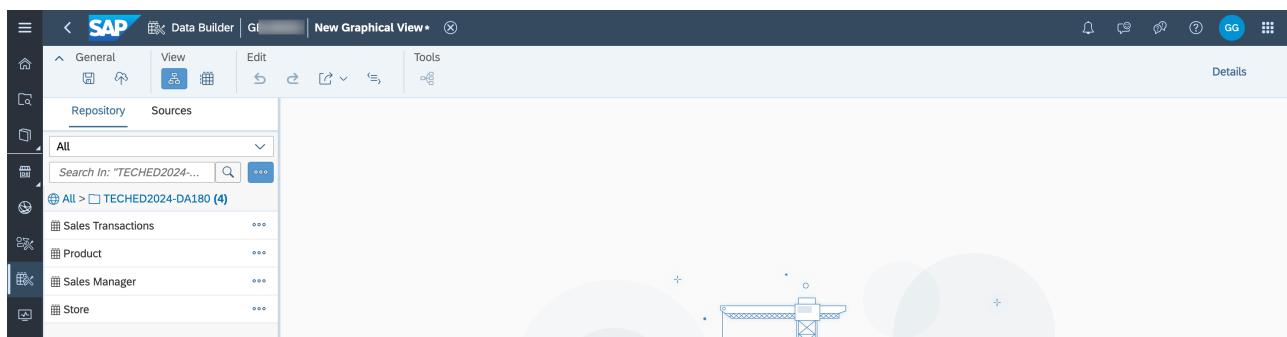


4. On the left-hand side you can decide between: - **Repository**: Here you have access to the local tables (imported data), views, Intelligent Lookups and Shared Objects. - **Sources**: Here you have access to the respective objects from your connections.

5. Ensure you select the option **Repository**.

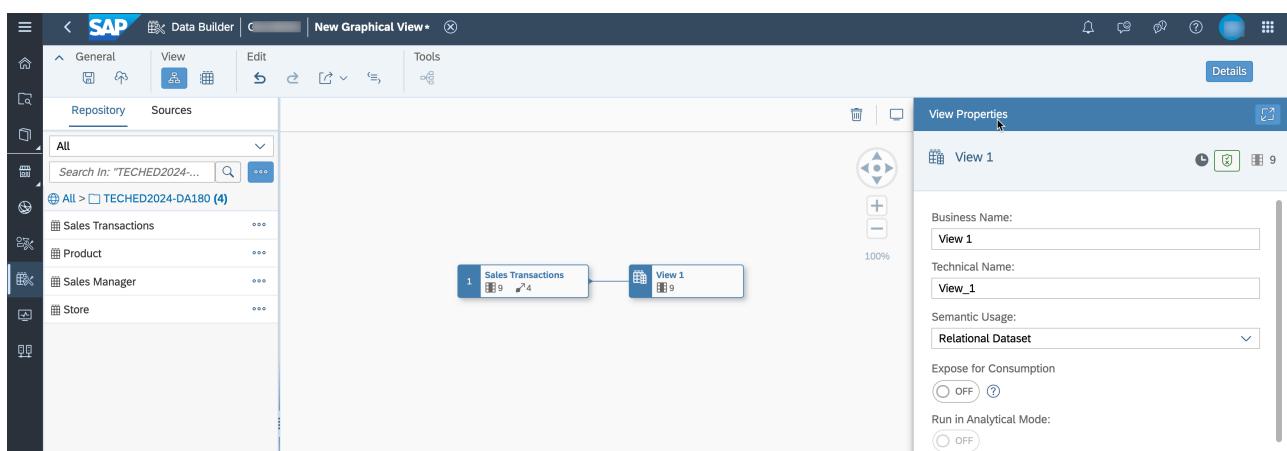
6. Open the folder **TECHED2024-DA180**.

7. You are presented with the list of tables, which we created previously.



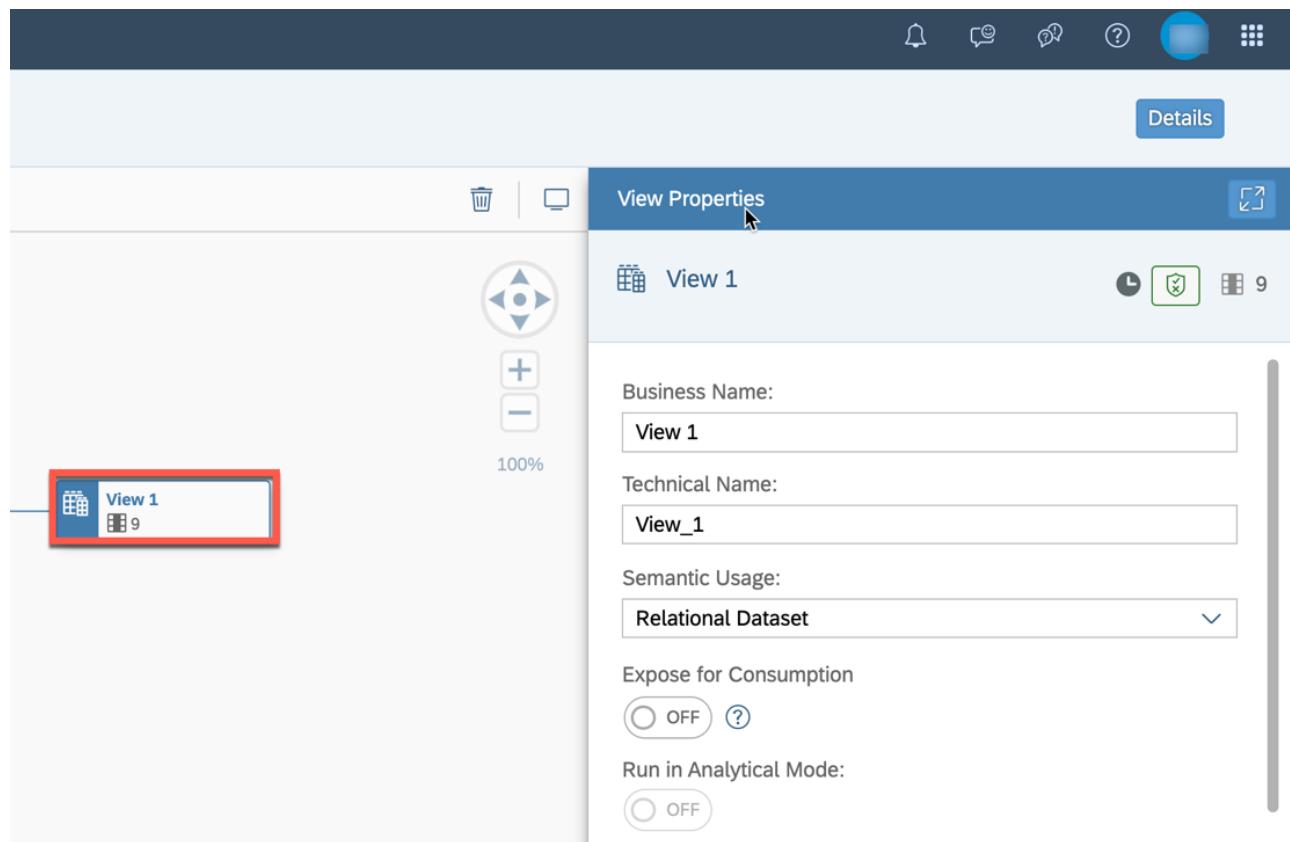
8. Drag and Drop the table **Sales Transactions** to the canvas.

9. When you drag the table to the canvas, you'll automatically get an output view as well. In this example, it's called **View 1**.



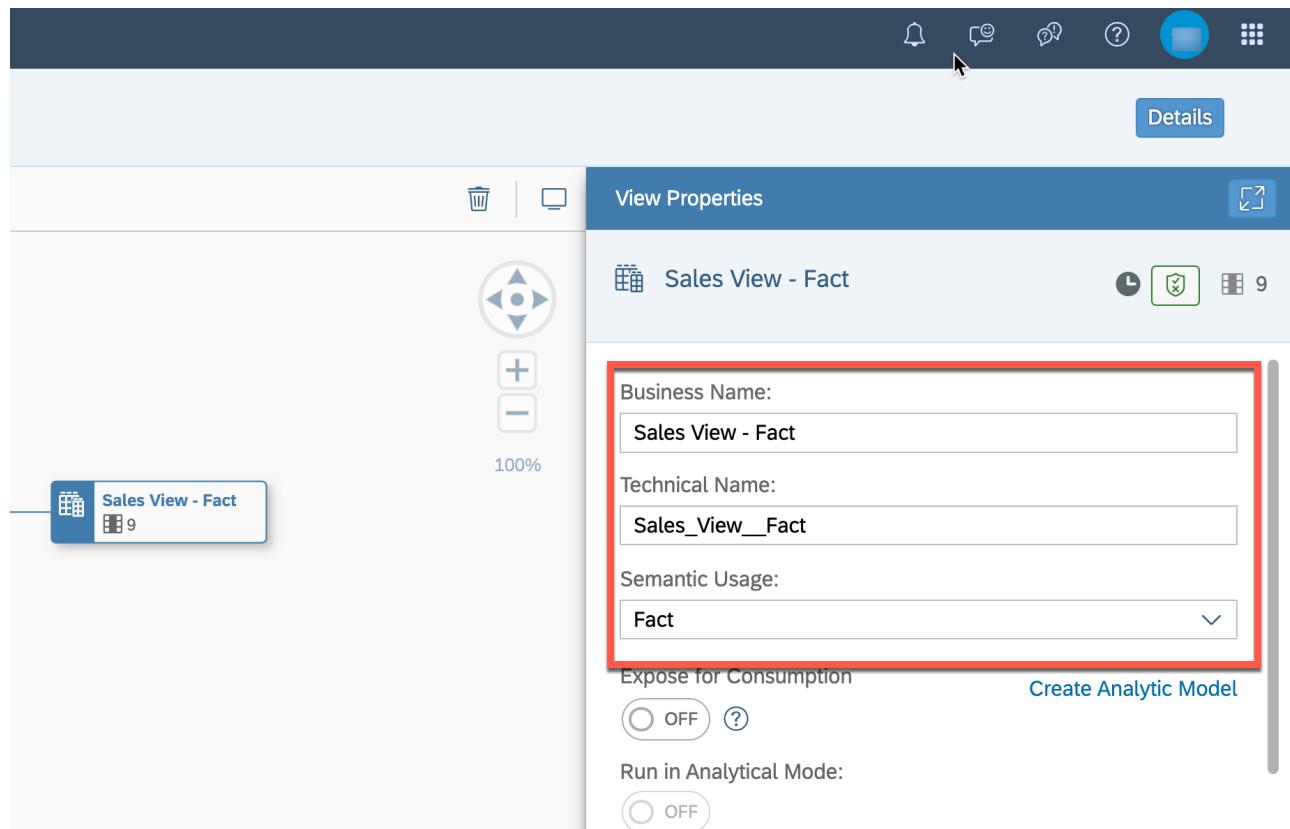
10. Ensure the option **Details** (top right corner) is enabled. Otherwise, click on the output view that was added, in our example **View 1**.

11. Navigate to the **Properties** window.



12. Here you should configure properties for the view:

- **Business Name:** Sales View – Fact
- **Technical Name:** Sales_View_Fact
- **Semantic Usage:** Fact

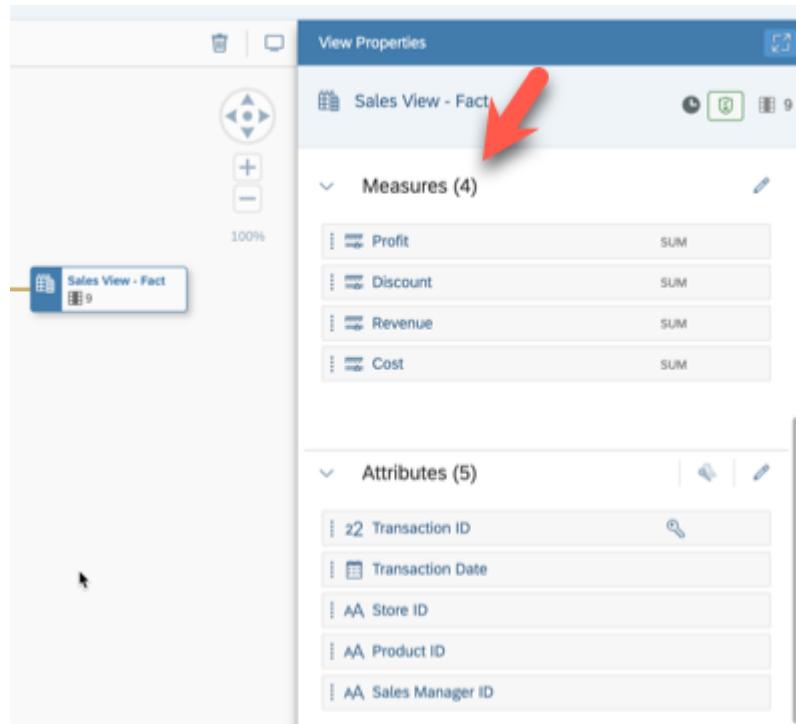


13. In the panel on the right-hand side, scroll down to the **Attributes** section.

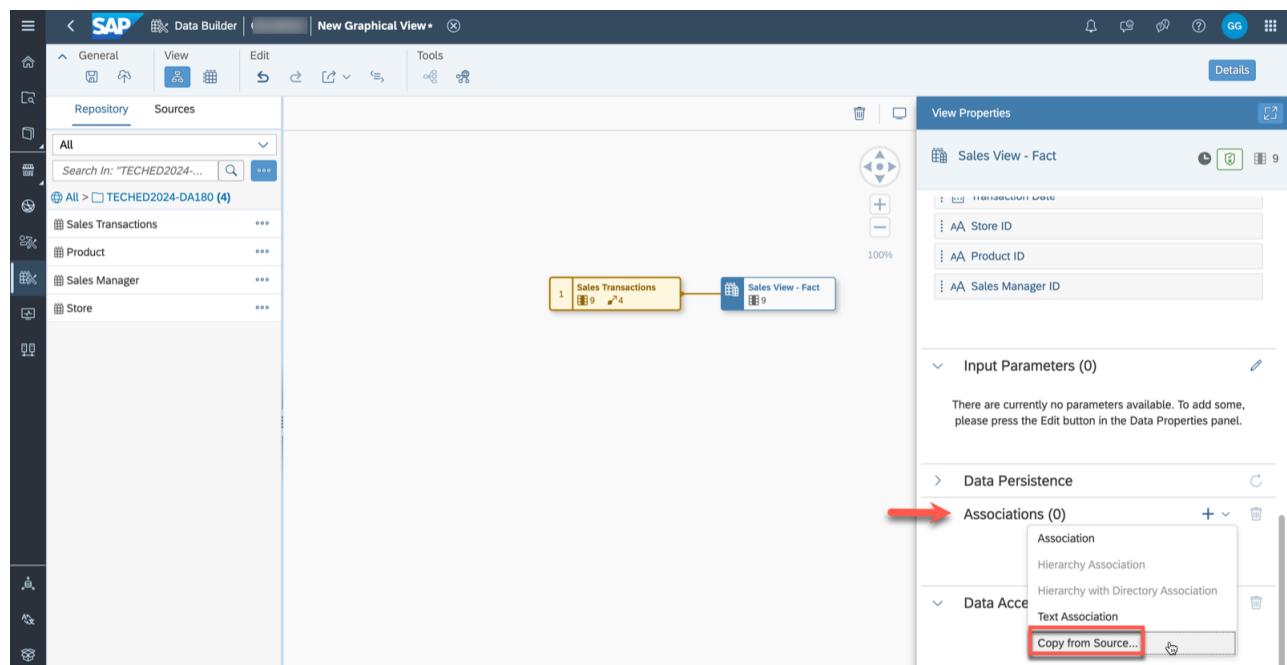
14. The columns **Profit**, **Discount**, **Revenue**, and **Cost** are measures (quantifiable values that refer to aggregatable fields of a model). Drag and drop these fields to the section **Measures** (multi-selection of measures is possible with keeping **CTRL** key pressed). You can also change the type by selecting the **Change to Measure** option.

The screenshot shows the 'View Properties' interface for a 'Sales View - Fact'. At the top, there are tabs for 'View Properties' and 'Edit Properties'. Below the tabs, the view title is 'Sales View - Fact' with a refresh icon, a shield icon, and a count of 9. There are three main sections: 'Measures (0)', 'Attributes (9)', and a central workspace. A red arrow points from the 'Measures (0)' section upwards towards the workspace. In the workspace, there is a placeholder text: 'Drop the attribute here to change it into a measure.' Below this, the 'Attributes (9)' section lists nine items. One item, '1.23 Discount', is highlighted with a red box and has a context menu open. The context menu options are: 'Change to Measure' (highlighted), 'Semantic Type >', 'Set as Key', and 'Add Text Association'. Other items in the 'Attributes' list include '22 Transaction ID', 'Transaction Date', 'AA Store ID', 'AA Product ID', 'AA Sales Manager ID', '1.23 Profit', '1.23 Revenue', and '1.23 Cost'.

15. Verify that your graphical view has four measures.



16. Scroll down to the section **Associations**. Click the + icon to add a new association. As the source of the view has associations itself, you can copy those instead of creating them again (indicated by the arrow symbol on the root view node). Select **Copy from Source**.



17. Select the available source associations for **Product ID**, **Sales Manager ID** and **Transaction Date**. Click **Create**.

18. As we created a new dimension view to add a column containing geographic coordinates, create a new association by selecting **Association** after clicking on the "+" icon next to the Associations section.
 19. Choose **Dimension Store** (View - Dimension) from the folder **TECHED2024-DA180**, and click on **Select**.

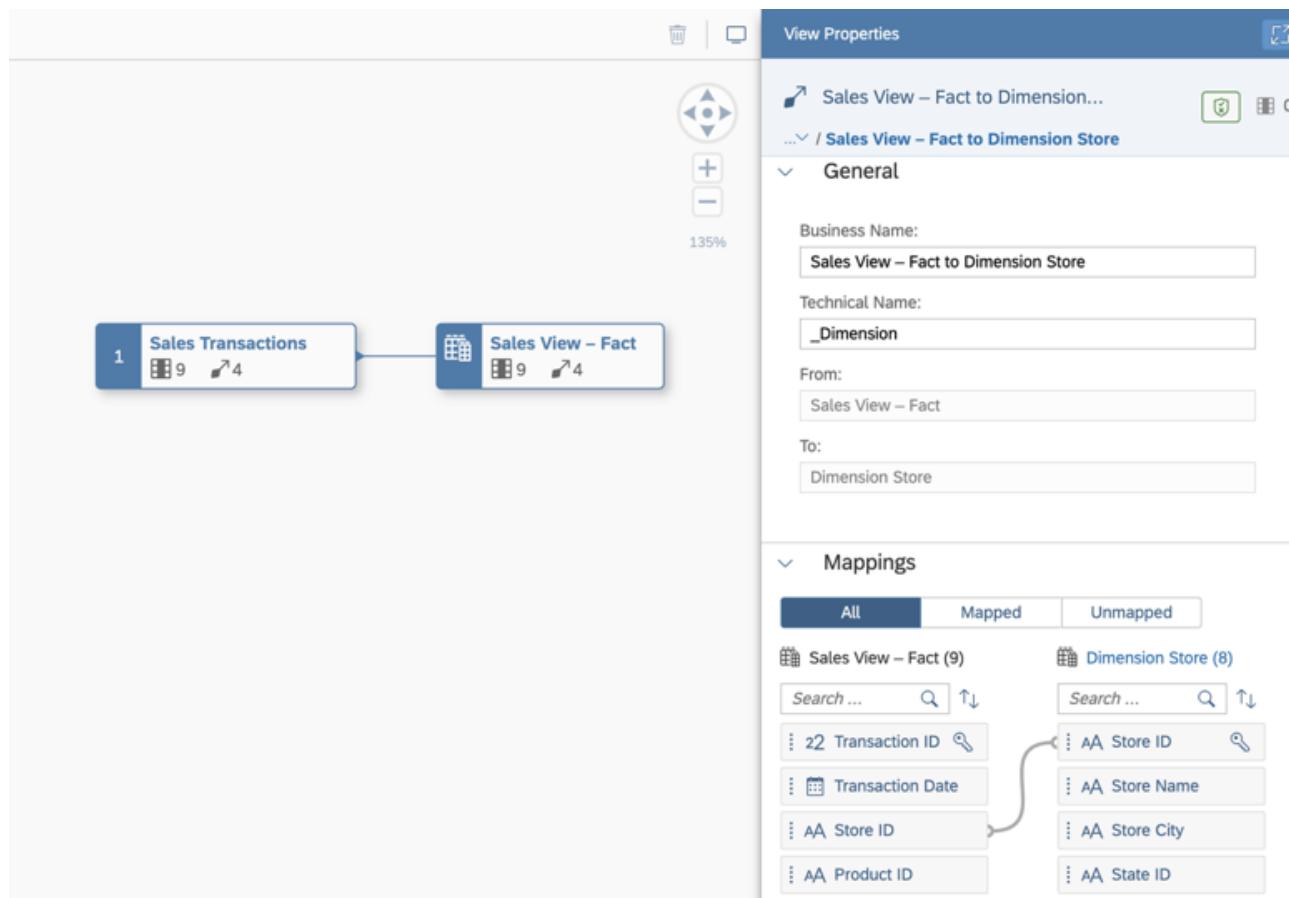
Select Association Target

Search In: "All"

List Objects In: [>All](#) > [TECHED2024-DA180 \(5\)](#)

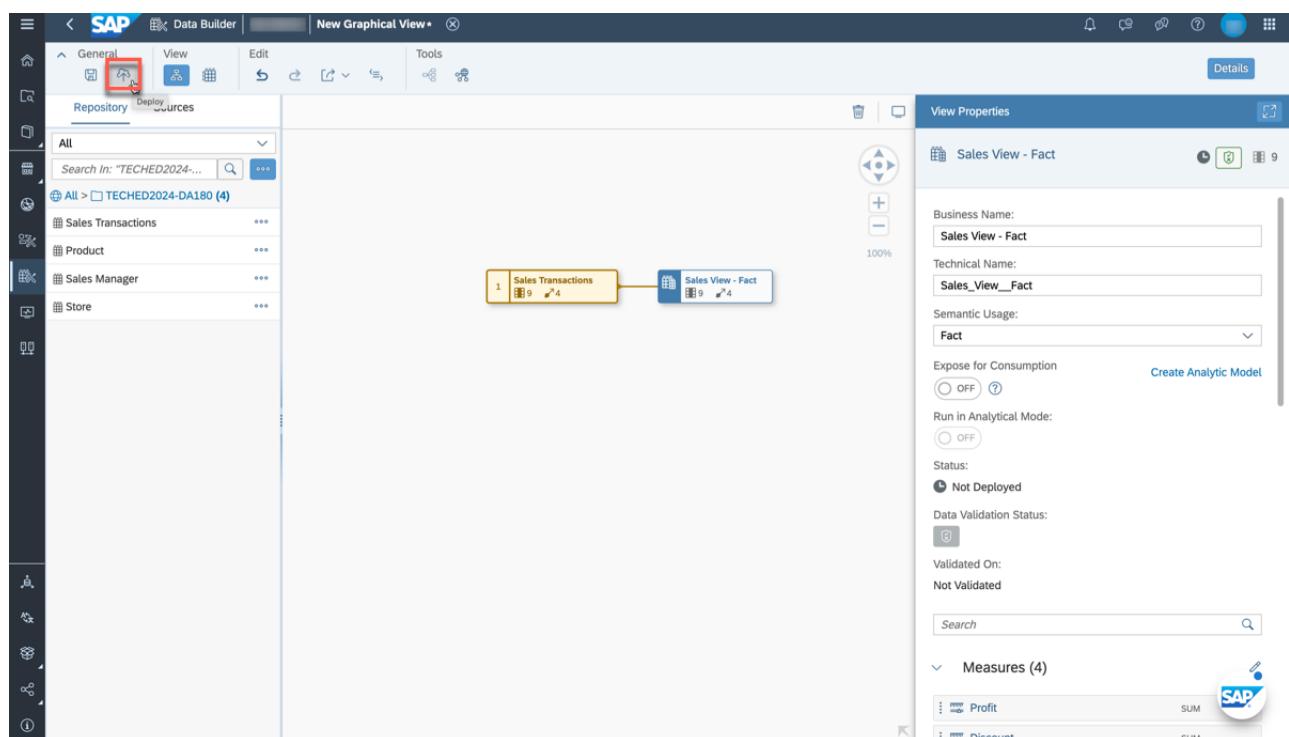
Business Name	Technical Name	Type (Semantic Usage)	Space ID	Folder Name	Object Status
Dimension Store	Dimension_Store	View (Dimension)	GE114169	TECHED2024-DA180	 Deployed
Sales Transactions	Sales_Transactions	Local Table (Relational D...)	GE114169	TECHED2024-DA180	 Deployed
Product	Product	Local Table (Dimension)	GE114169	TECHED2024-DA180	 Deployed
Sales Manager	Sales_Manager	Local Table (Dimension)	GE114169	TECHED2024-DA180	 Deployed
Store	Store	Local Table (Dimension)	GE114169	TECHED2024-DA180	 Deployed

20. Verify that the mapping is done correctly on the column ***Store ID***.

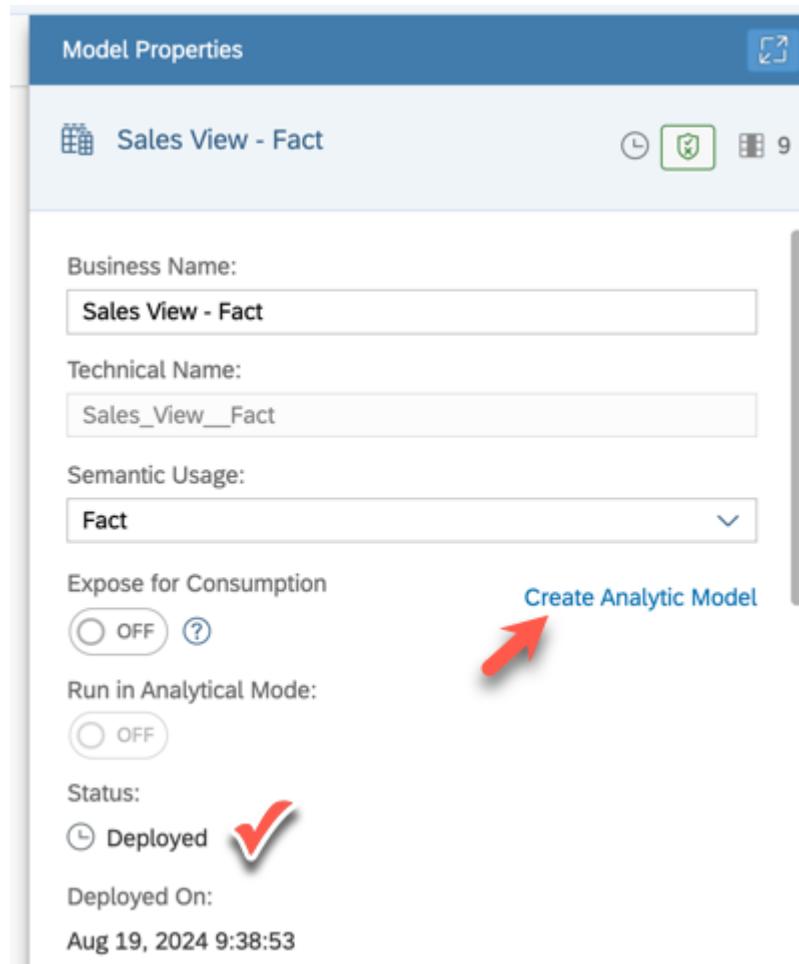


21. Save the view in the folder **TECHED2024-DA180**. You will be asked to confirm and **Save** the Business Name and Technical Name.

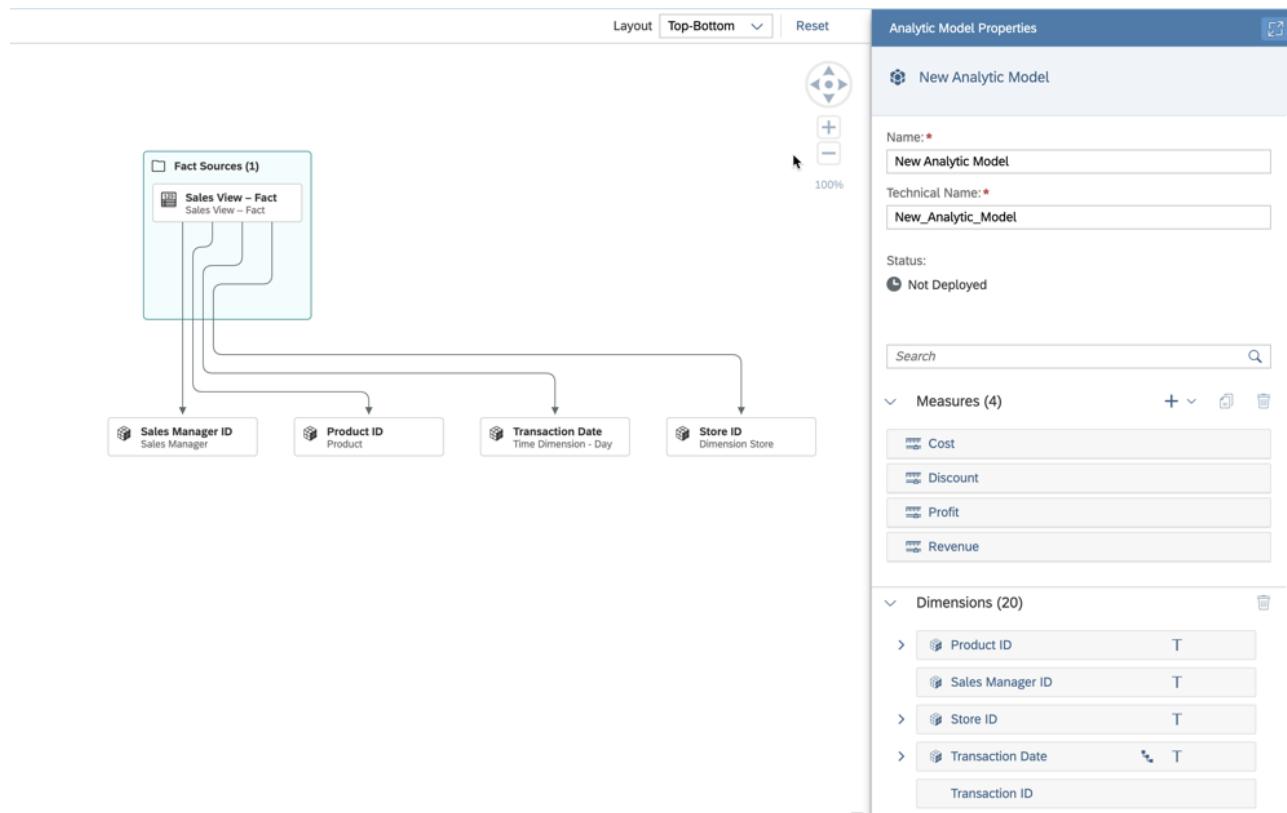
22. Deploy your view.



23. Once the view has been successfully deployed, you can create your Analytic Model by selecting the according option on the right-hand side.



24. You are now within your Analytic Model as depicted below. You are able to see the sales fact view and its associated dimensions that you added previously.



25. **Save** and **Deploy** your Analytic Model with the business name **Sales - Analytic Model** (the technical name is generated). Select the folder **TECHED2024-DA180**.

Summary

You have now created and deployed your first fact-based view and Analytic Model. The Analytic Model can now be consumed within SAP Analytics Cloud to answer different business questions.

Continue with - [Exercise 8 - Top 10 Revenue Generating Products](#)

Exercise 8 - Top 10 Revenue Generating Products

In this exercise, we will create a story in SAP Analytics Cloud (SAC), which allows us to analyze and identify the top 10 revenue generating products.

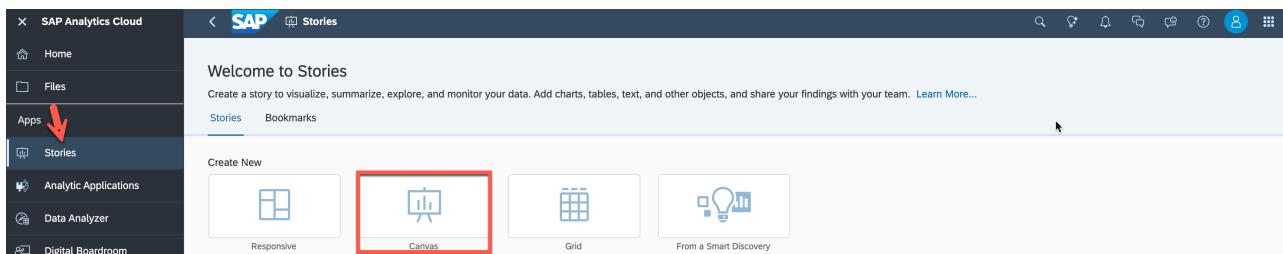
1. Log On to your SAP Analytics Cloud tenant.



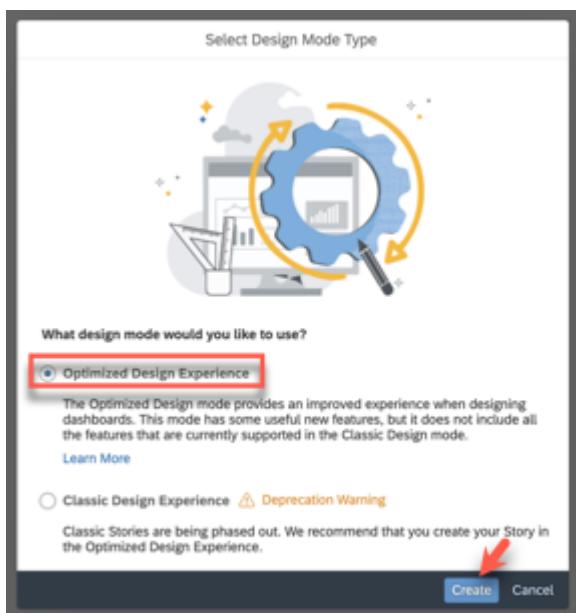
Note: The system might prompt you to sign in again. Use the same username and password for SAC as you do for Datasphere.

2. Select the menu **Stories** in the left-hand panel.

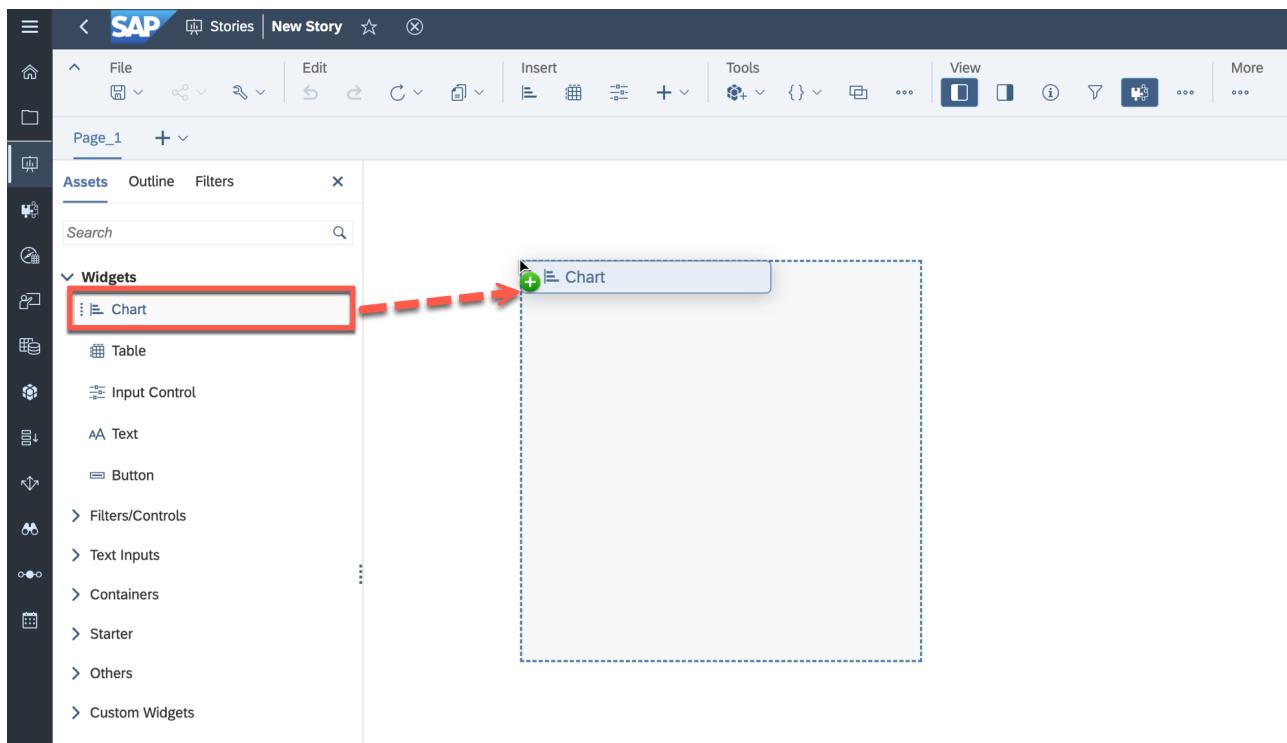
3. Select the option **Canvas** to create a new story.



4. Select **Optimized Design Experience** when asked which design mode to use. Click **Create**.

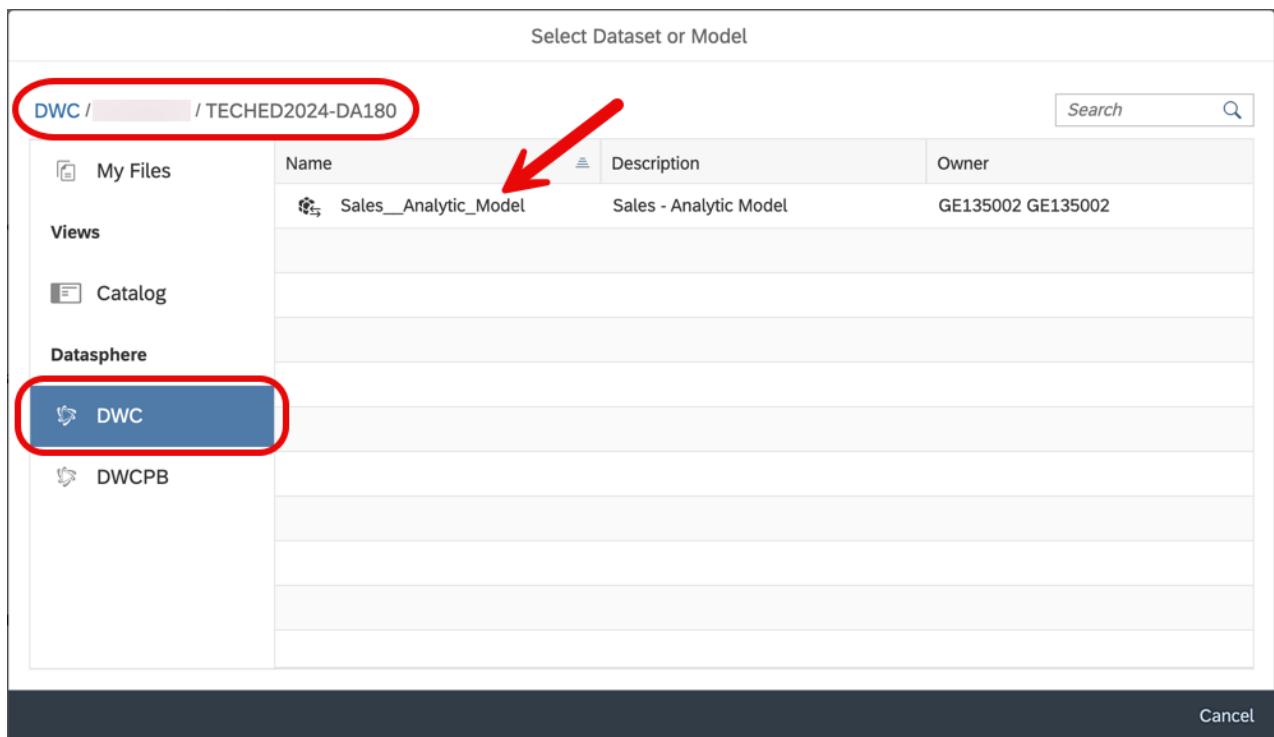


5. Under **Widgets**, select and drag **Chart** onto the canvas.



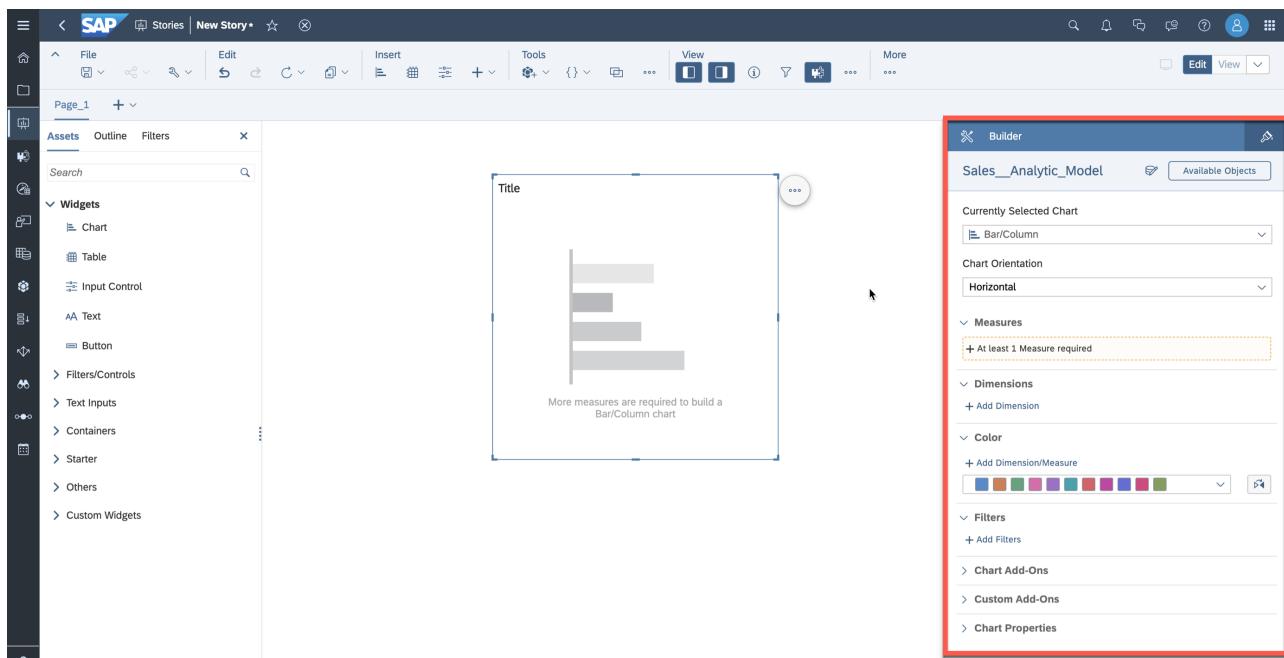
6. To select the model that you want to reference in your story:

- Select **DWC** as the connection on the left panel.
- Select your space, e.g., **GE123456**, and the folder **TECHED2024-DA180**.
- For our first example, select your **Sales - Analytic Model**.

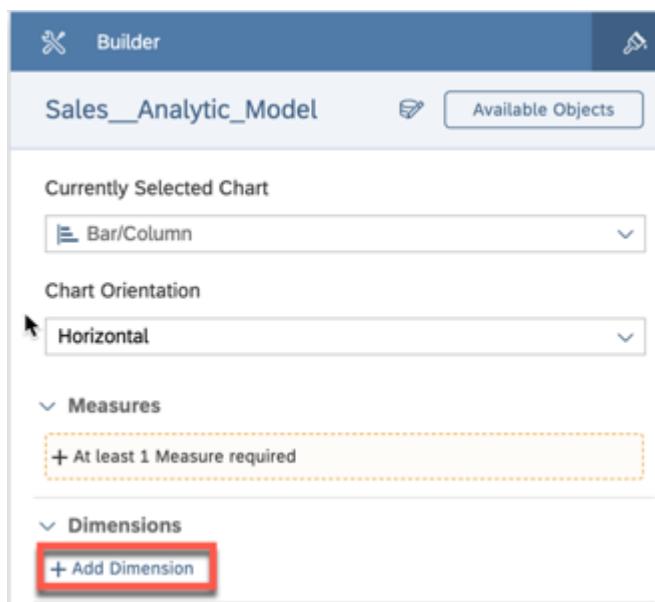


7. Now select the newly created empty chart on the canvas.

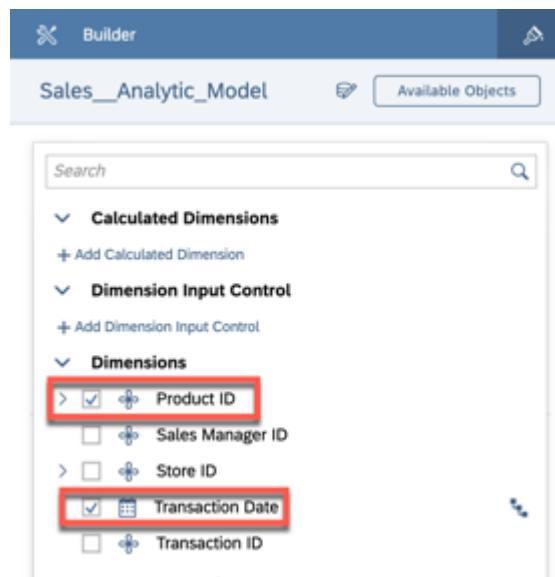
8. Navigate to the **Builder** panel on the right-hand side.



9. Click **Add Dimension** as part of the Dimensions section.

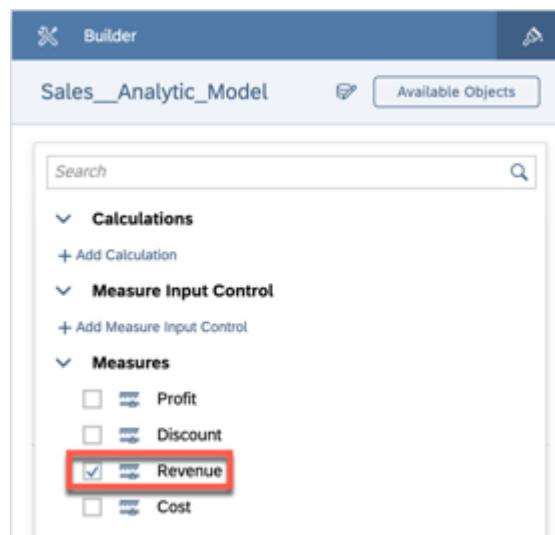


10. Select **Transaction Date** and **Product ID**. Afterwards, click anywhere outside the view for dimensions selection to return to the chart settings.

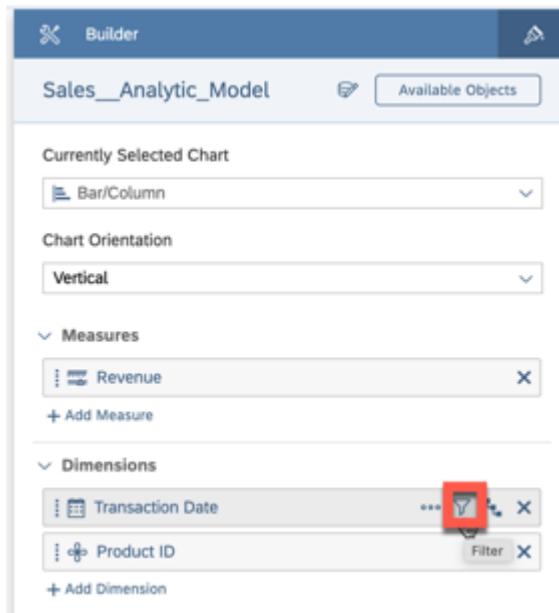


💡 Ensure the **Transaction Date** is positioned first within the dimensions section overview.

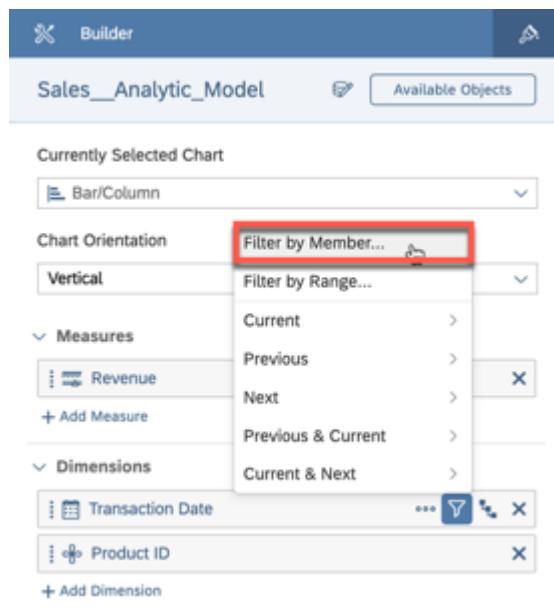
11. Add a measure by clicking on the **At least 1 Measure required** section.
12. Select the measure **Revenue**. Afterwards, click anywhere outside the view for measure selection to return to the chart settings.



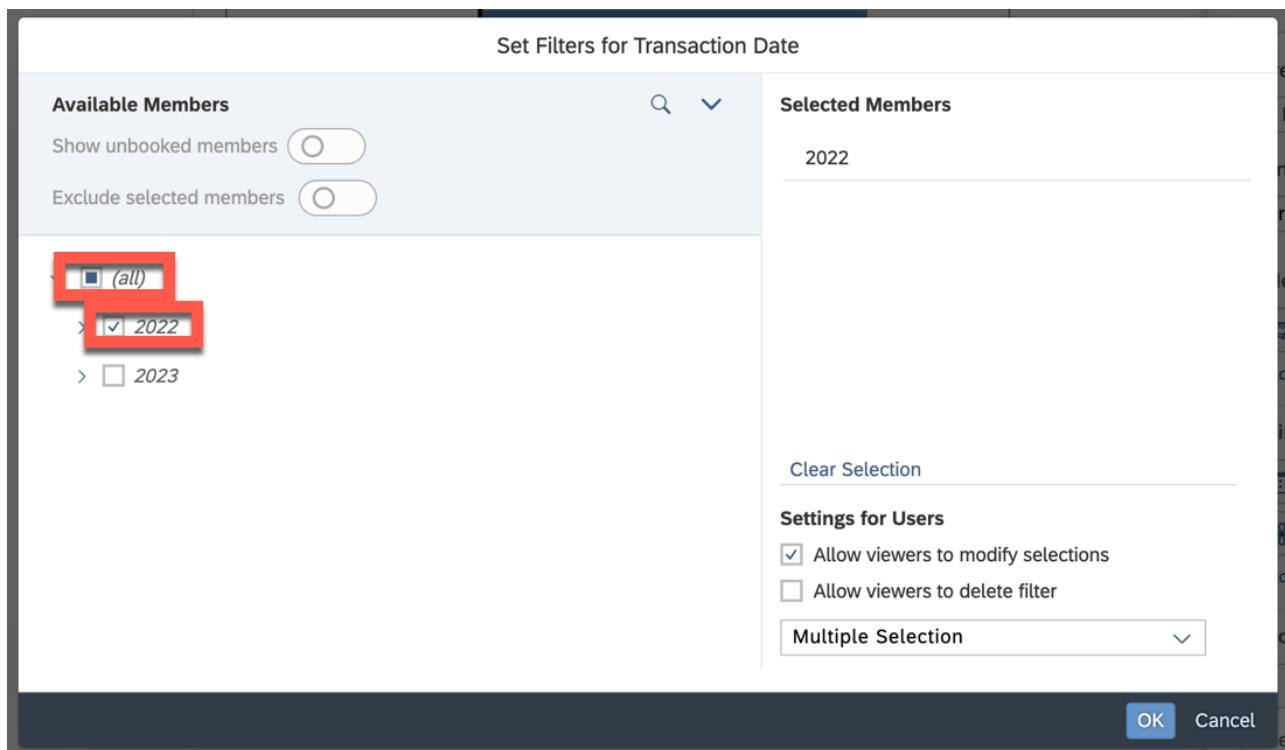
13. Click on the Filter icon for the dimension **Transaction Date**.



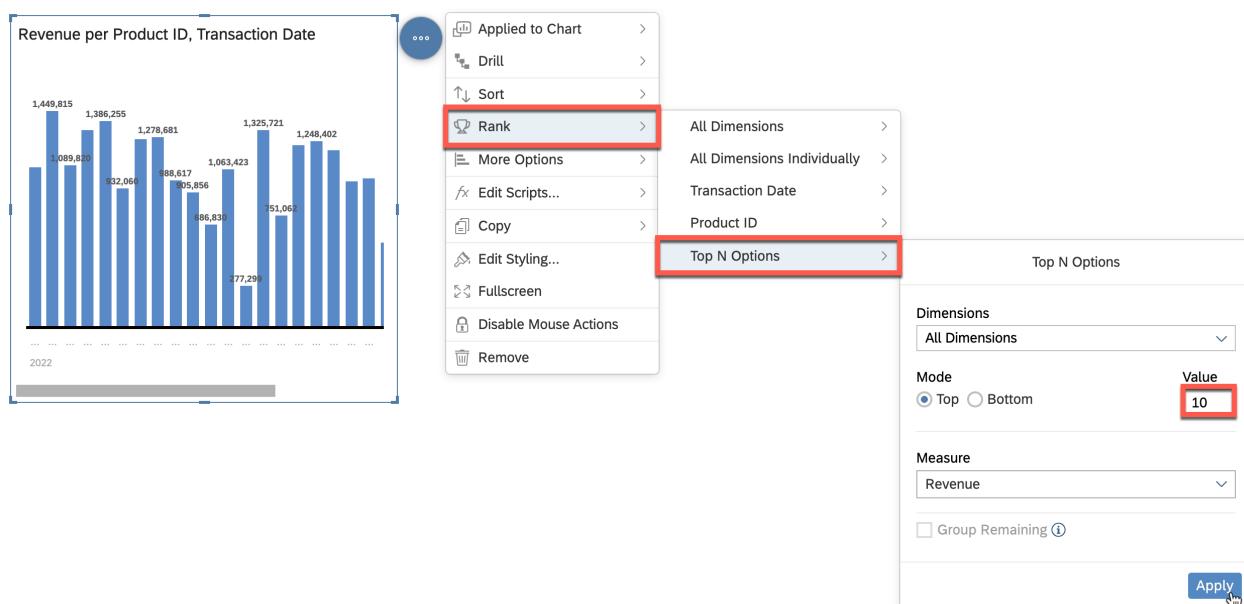
14. Select the option **Filter by Member**.



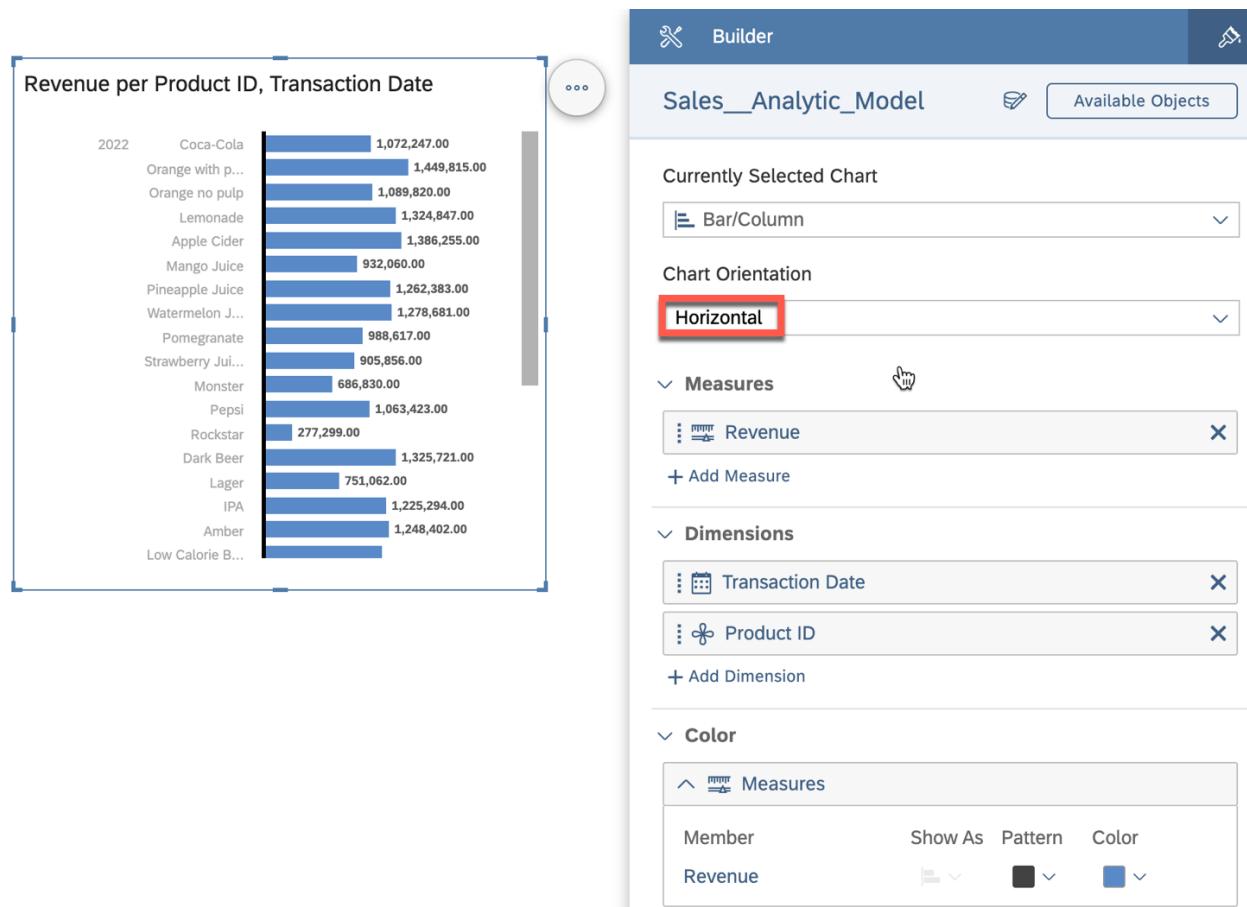
15. Open the list of members and select the year **2022**. Click **OK**.



16. Open the **More Actions** menu (...) for the chart (top right corner of chart). Within the menu, go to **Rank** > **Top N Options** > Update value to **10**. Click **Apply**.

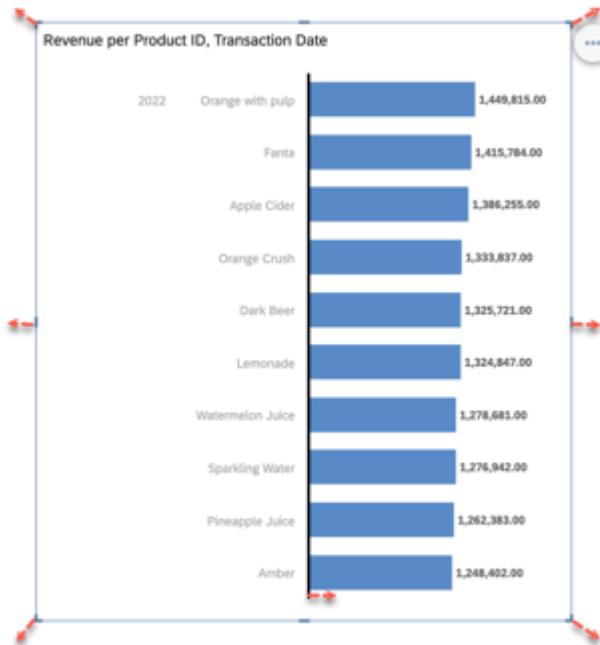


17. Update Chart Orientation to **Horizontal** in the Builder Panel.

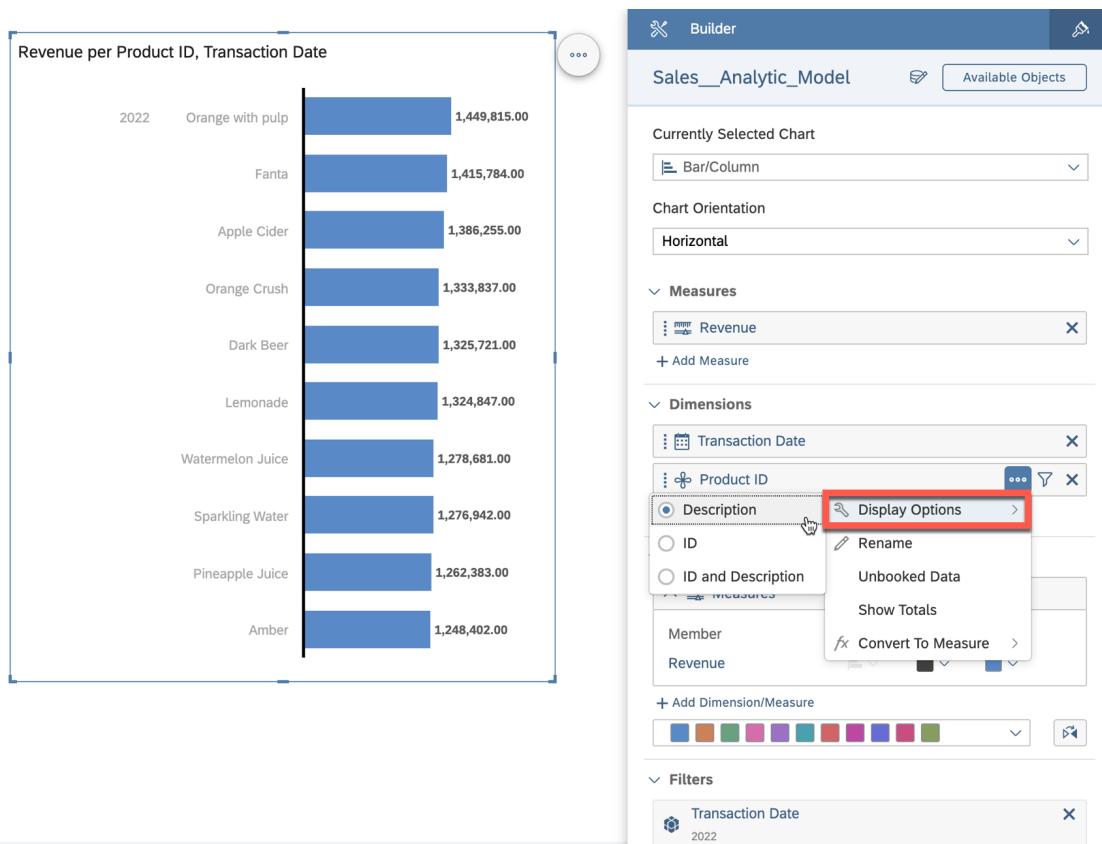


18. You can adjust the size of the chart by clicking and dragging the brackets outward or inward.

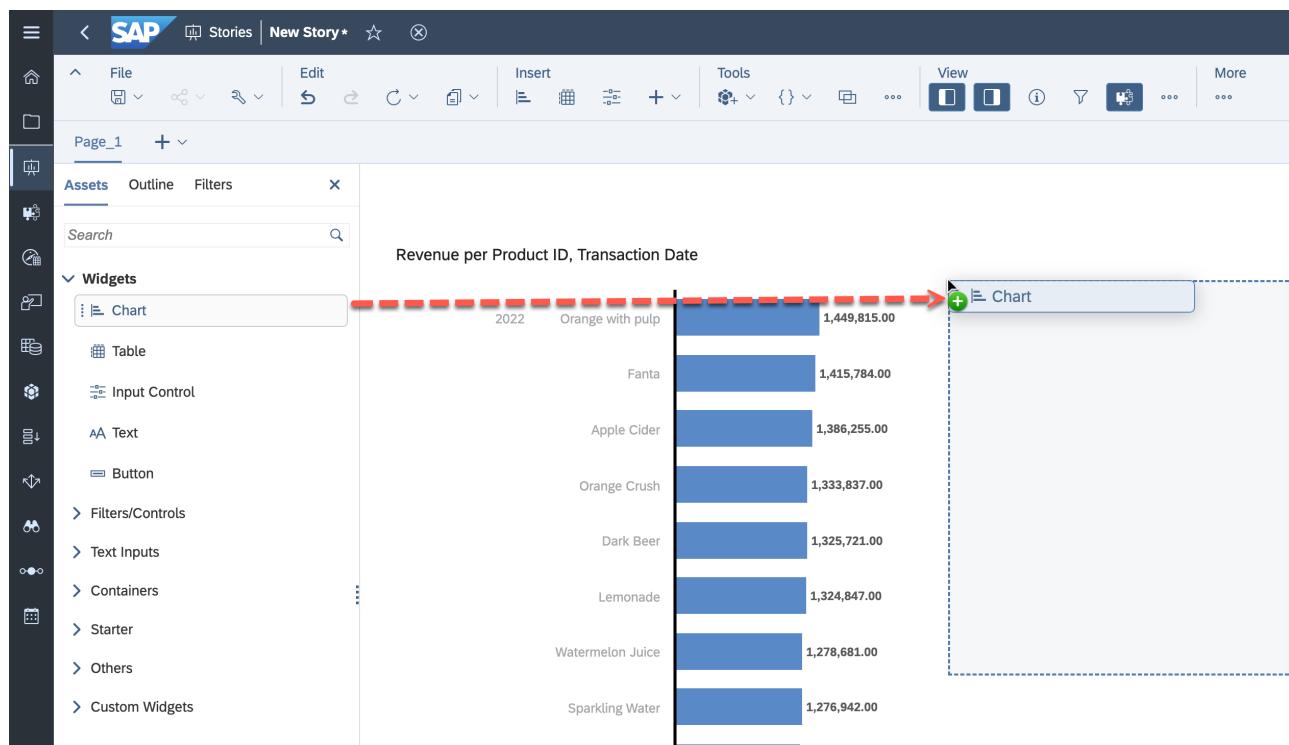
Additionally, you can move the y-axis to ensure the full product name is displayed.



19. Your chart should look like this. Note that only the description of the product is displayed, based on the configuration in **Display Options** for **Product ID**.

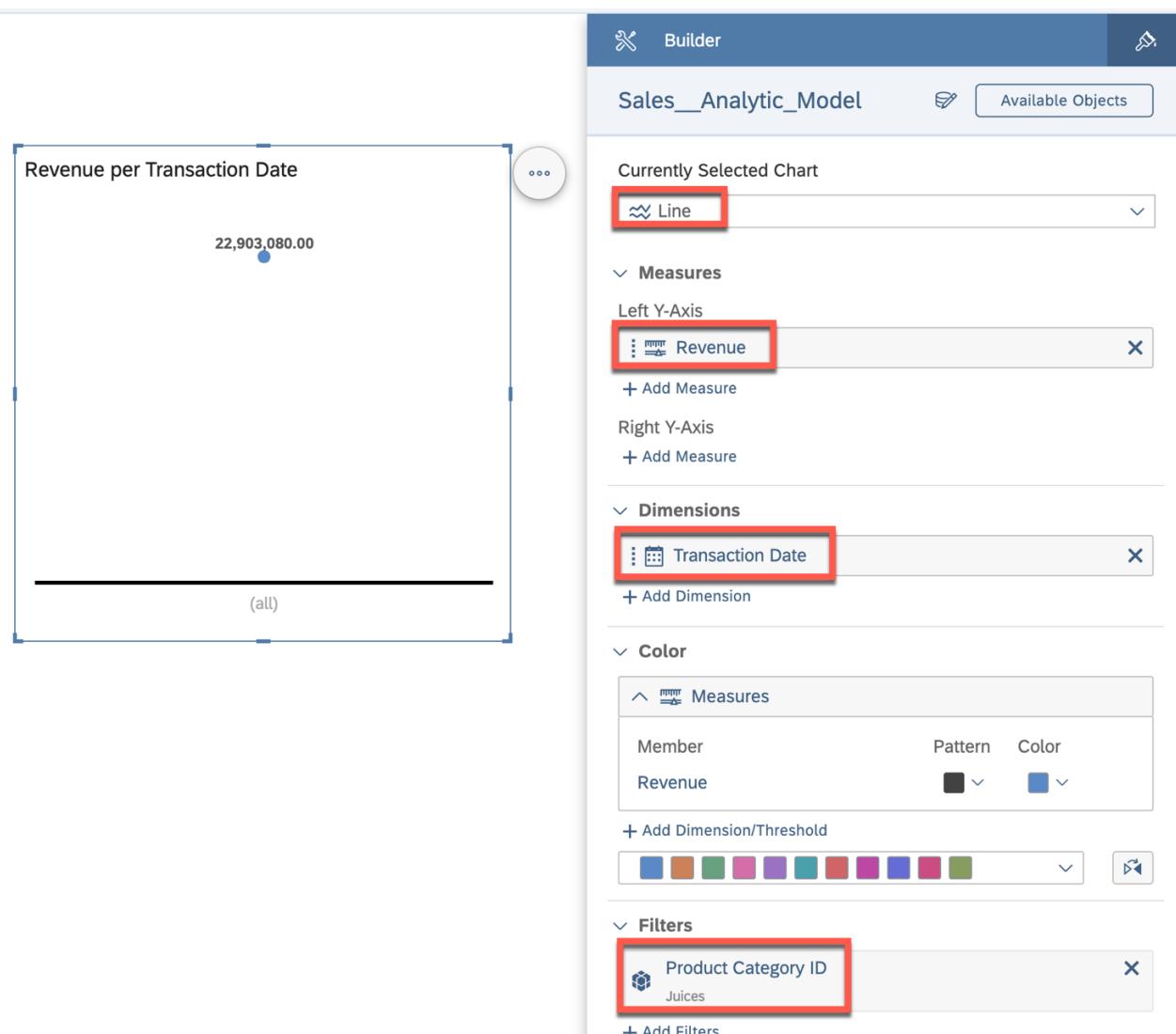


20. We want to visualize the change in revenue over time using an additional chart. Drag a new chart onto the canvas to get started.

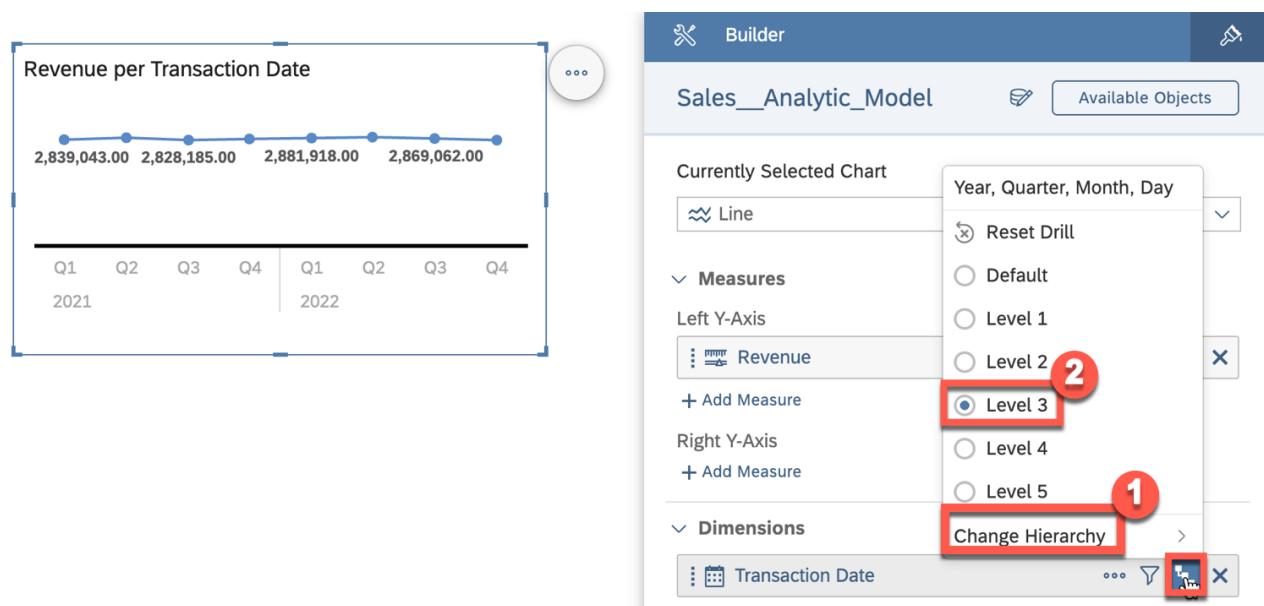


21. Set the following settings:

- Select **Line** as Currently Selected Chart.
- The measure for **Left Y-Axis** is **Revenue**.
- Select **Transaction Date** as Dimension.
- Filter **Product Category ID (Member)** to **Juices**.

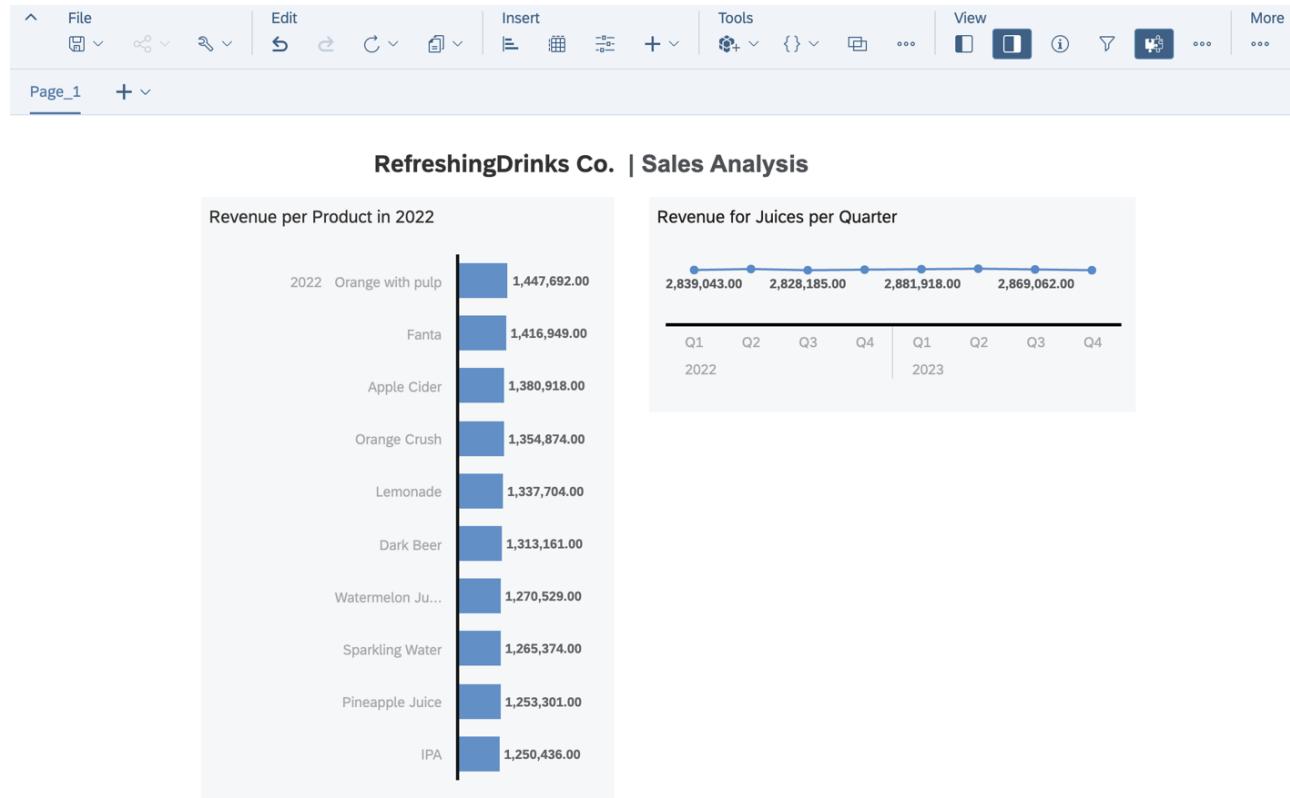


22. Use the time hierarchy defined for **Transaction Date** to perform a quarter-based analysis. Set the hierarchy to **Year, Quarter, Month, Day** by clicking on **Change Hierarchy**. Adjust the level to **Level 3** so that the revenue is displayed per quarter.



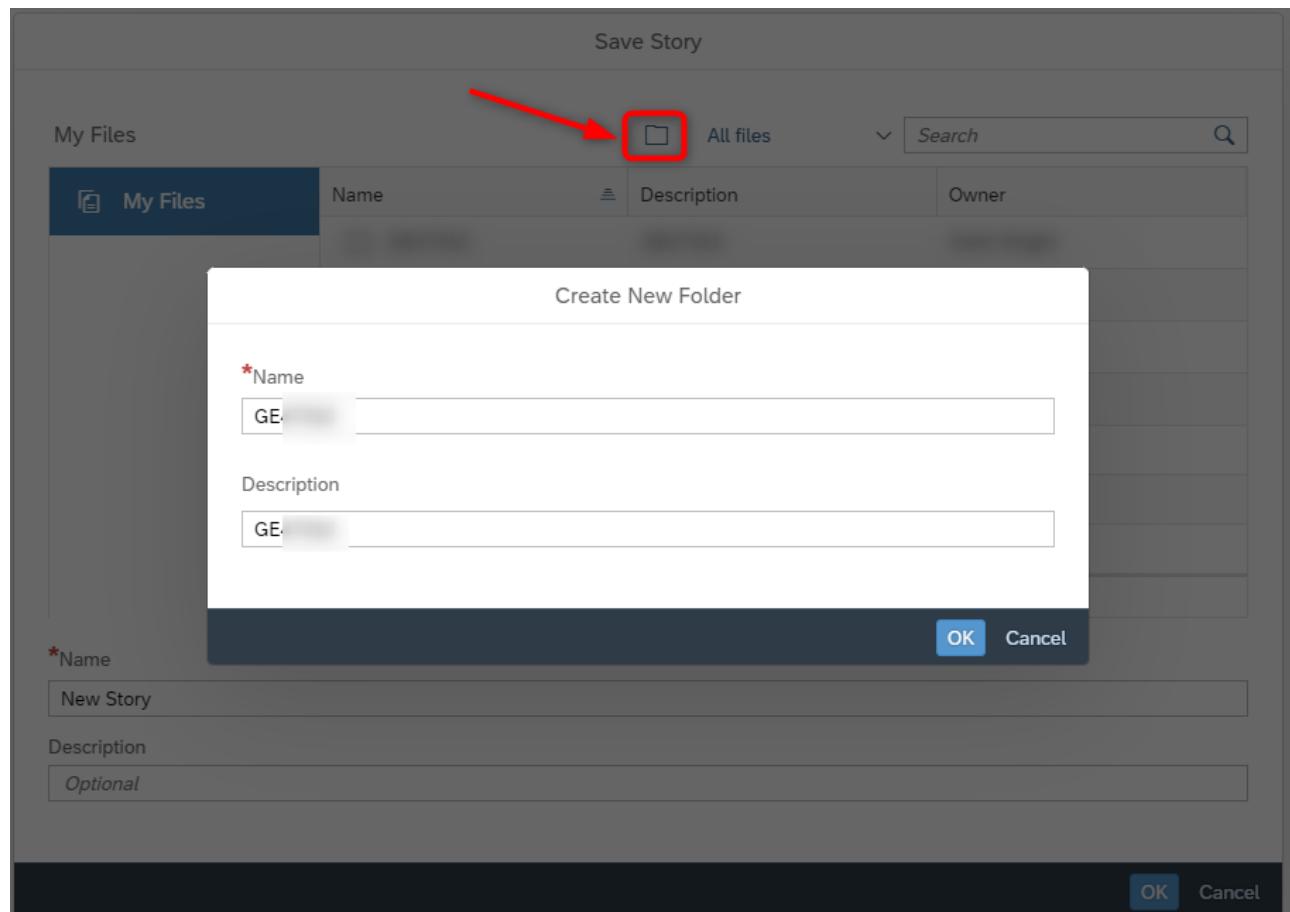
23. You can now modify the style of your story and add elements, headings, or adjust chart titles. If you select + (**Insert** section), you will see options to insert standard shapes or other entities, such as

dynamic text.



24. In the **File** menu, select the option **Save** to save your story.

25. Create a folder that matches your space name, e.g., **GE123456**.



26. Select the folder **GE123456**. Enter a name and a description, such as **Revenue Analysis - Products**.

27. Click **OK**.

Summary

You've now created your first SAP Analytics Cloud story based on the data model you built earlier.

Continue to - [Exercise 09: Revenue by Geography](#)

Exercise 9 - Geographic Revenue Distribution

In this exercise, we will set up a story in SAP Analytics Cloud that allows us to visualize the measures on a geographic map.

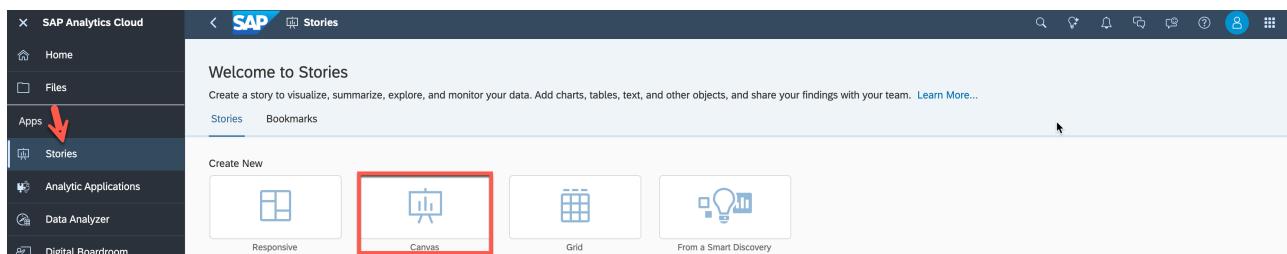
1. Log On to your SAP Analytics Cloud tenant.



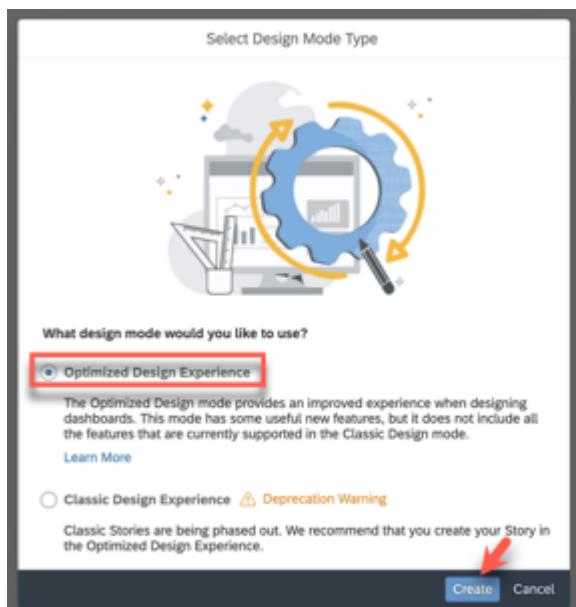
Note: The system may ask you to sign in again. Use the same user name and password for SAC as for Datasphere.

2. Select the menu **Stories** in the left-hand panel.

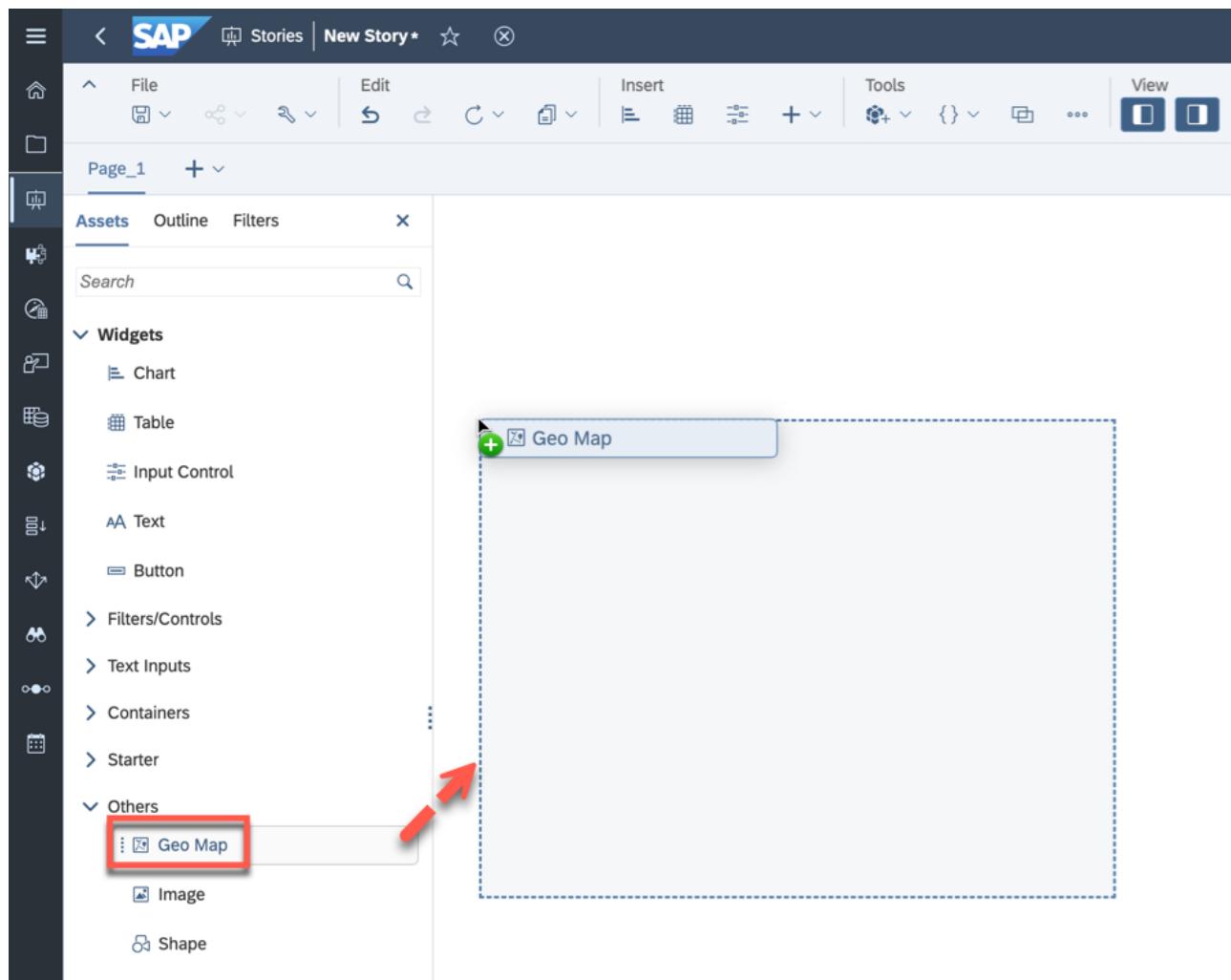
3. Select the option **Canvas** to create a new story.



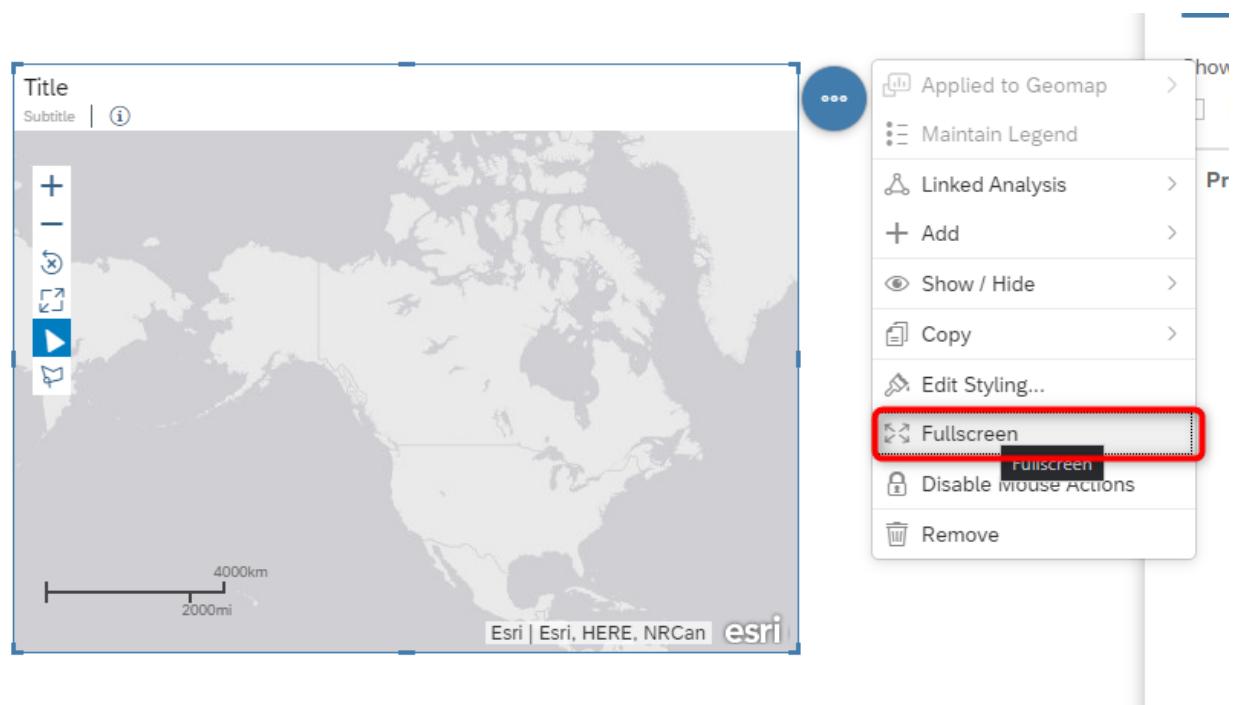
4. Select **Optimized Design Experience** when asked which design mode to use. Click **Create**.



5. Under **Others**, select and drag the **Geo Map** onto the canvas.



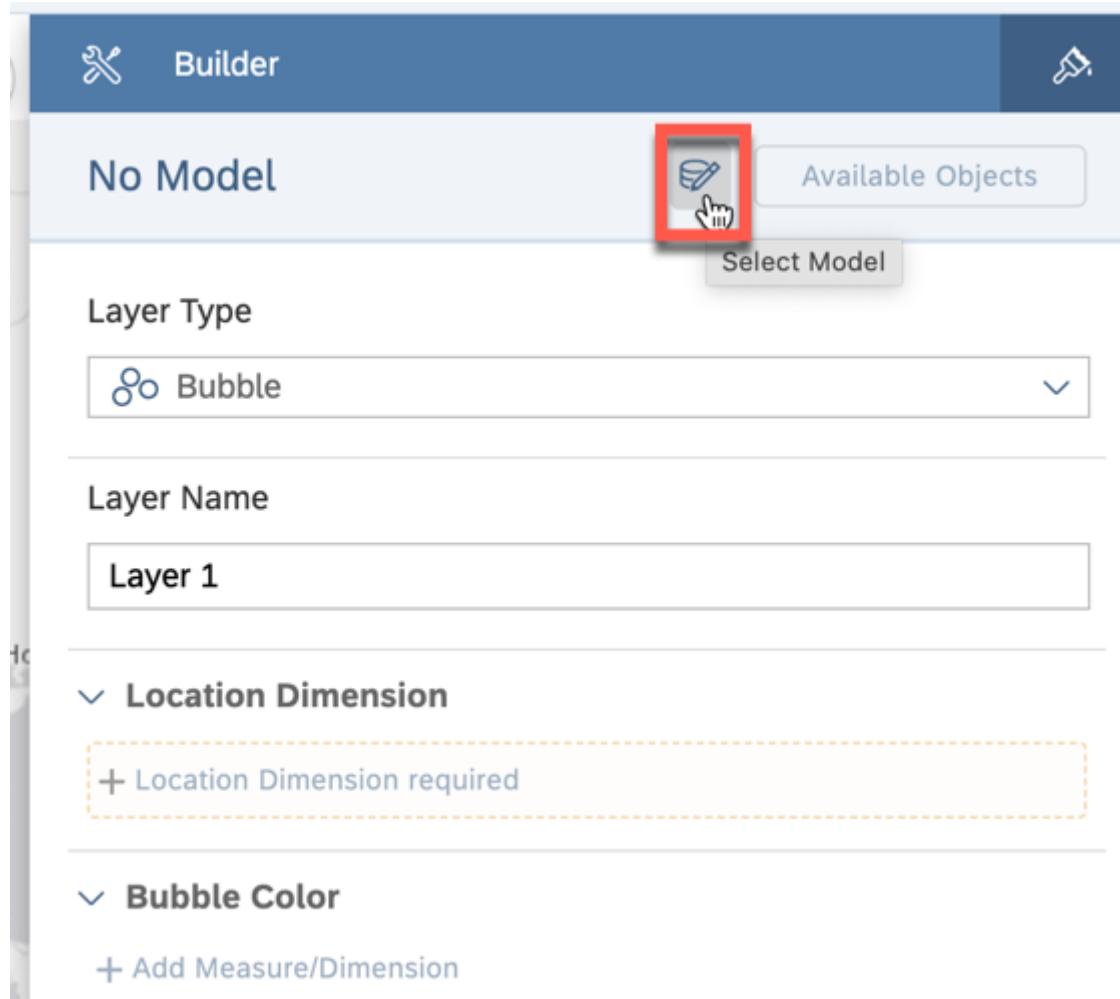
6. Resize the map so that it uses the complete canvas. You can accomplish this by opening the **More** menu (...) and selecting **Fullscreen**.



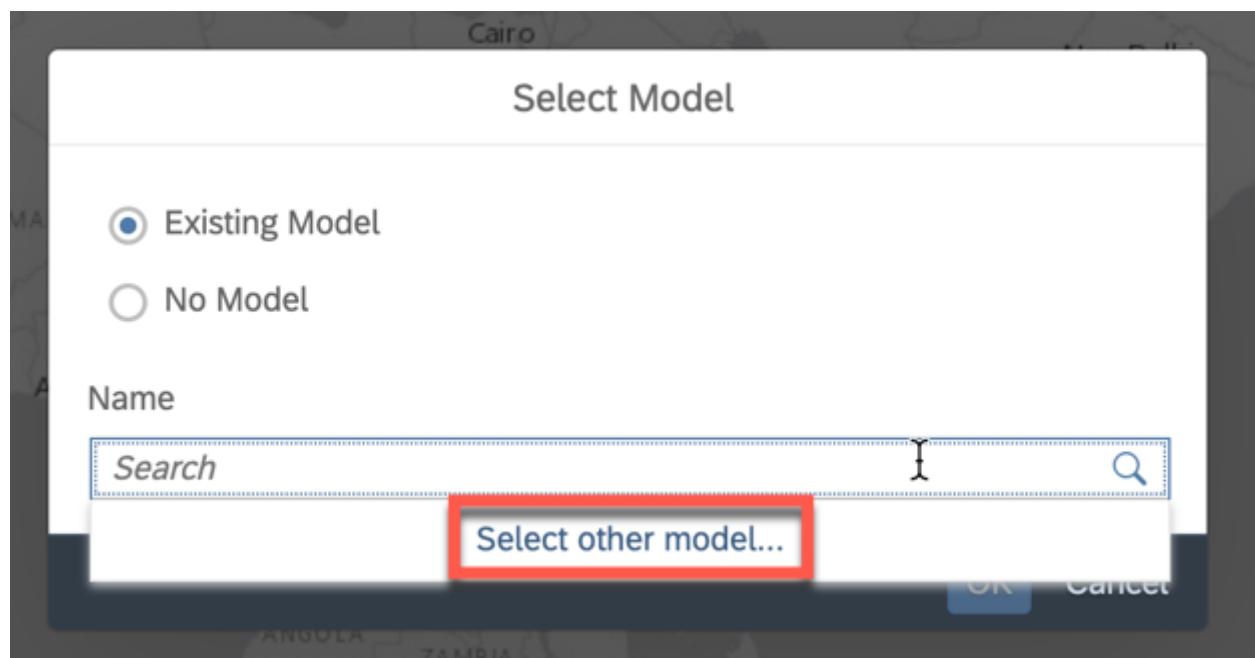
7. In the Builder panel on the right-hand side, select the option **Add Layer** for the Content Layer option.

The screenshot shows the 'Builder' interface with a blue header bar. On the left is a wrench icon, and on the right is a gear icon. The main title 'Geo Map Layers' is displayed in large blue text. Below it, a section titled 'Content Layers' has a red box around the '+ Add Layer' button. A mouse cursor is hovering over the 'Base Layer' section. Under 'Base Layer Style', there is a dropdown menu set to 'Light Gray'. To its right is an 'Opacity' input field containing '100' with a slider below it. In the 'Show/Hide' section, there is a checkbox labeled 'Hide Base Layer'. At the bottom, a section titled 'Properties' is partially visible.

8. Click on the model icon to choose your data model.



9. Click within the search field and choose **Select other model** option.



10. To select the model that you want to reference in your story:

- Select **DWC** as the connection on the left panel.
- Select your space, e.g., **GE123456**, and the folder **TECHED2024-DA180**.

- For our first example, select your **Sales - Analytic Model**.

Select Dataset or Model

Views	Name	Description	Owner
	Sales__Analytic_Model	Sales - Analytic Model	GE135002 GE135002

Cancel

11. In the Builder panel, click on **Location Dimension Required** for the **Location Dimension** area.

12. Select the option **Store Location**. This is the store location dimension we created previously based on the longitude and latitude values for the store dimension.

Builder

Sales__Analytic_Model Available Objects

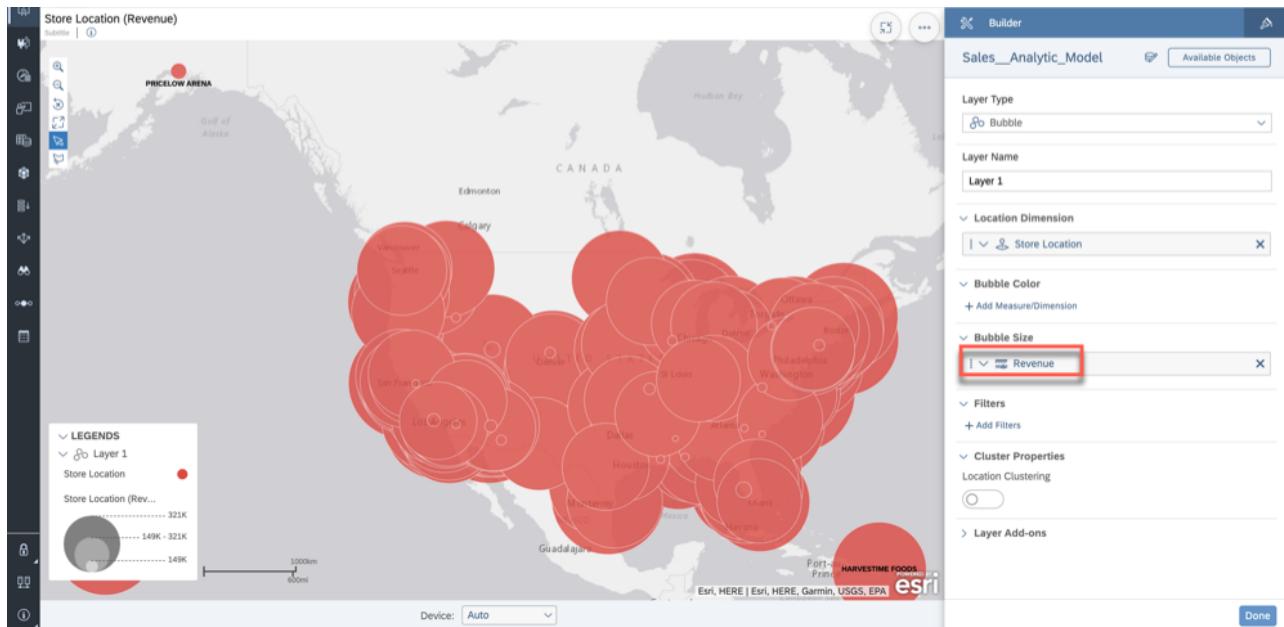
Search

Location Dimensions

Store Location

13. Click on **Add Measure** for the Bubble Size.

14. Select measure **Revenue**.



15. Click **Add Measure / Dimension** for the Bubble Color

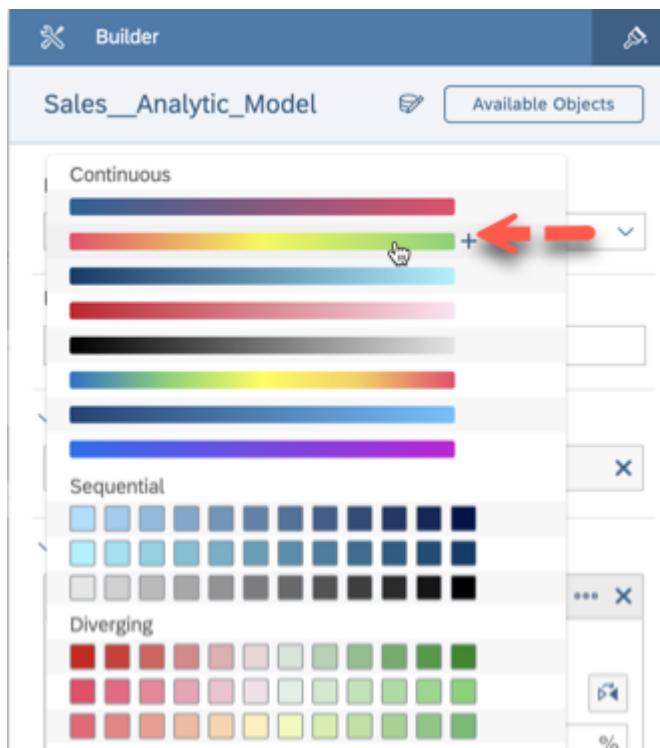
16. Select measure **Profit**.

17. Now open the details for the measure **Profit** as part of the Bubble Color.

18. Open the list of color **Palette**.

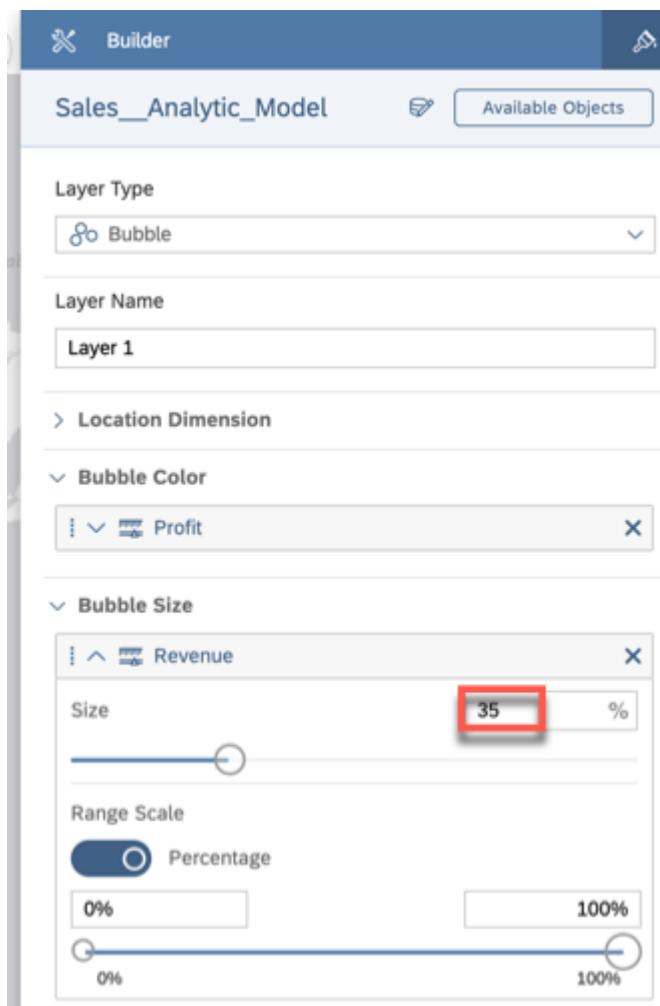
The screenshot shows the Tableau Builder interface with the project name "Sales_Analytic_Model". The "Available Objects" pane is open, showing a configuration for a layer named "Layer 1". Under "Location Dimension", "Store Location" is selected. In the "Bubble Color" section, the "Profit" field is chosen. The color palette is a continuous gradient from red to green. The second entry in the palette is highlighted with a red box. The "Ranges" dropdown is set to 1, and the opacity is set to 80%.

19. Select the second entry from the **continuous** category (red to green).



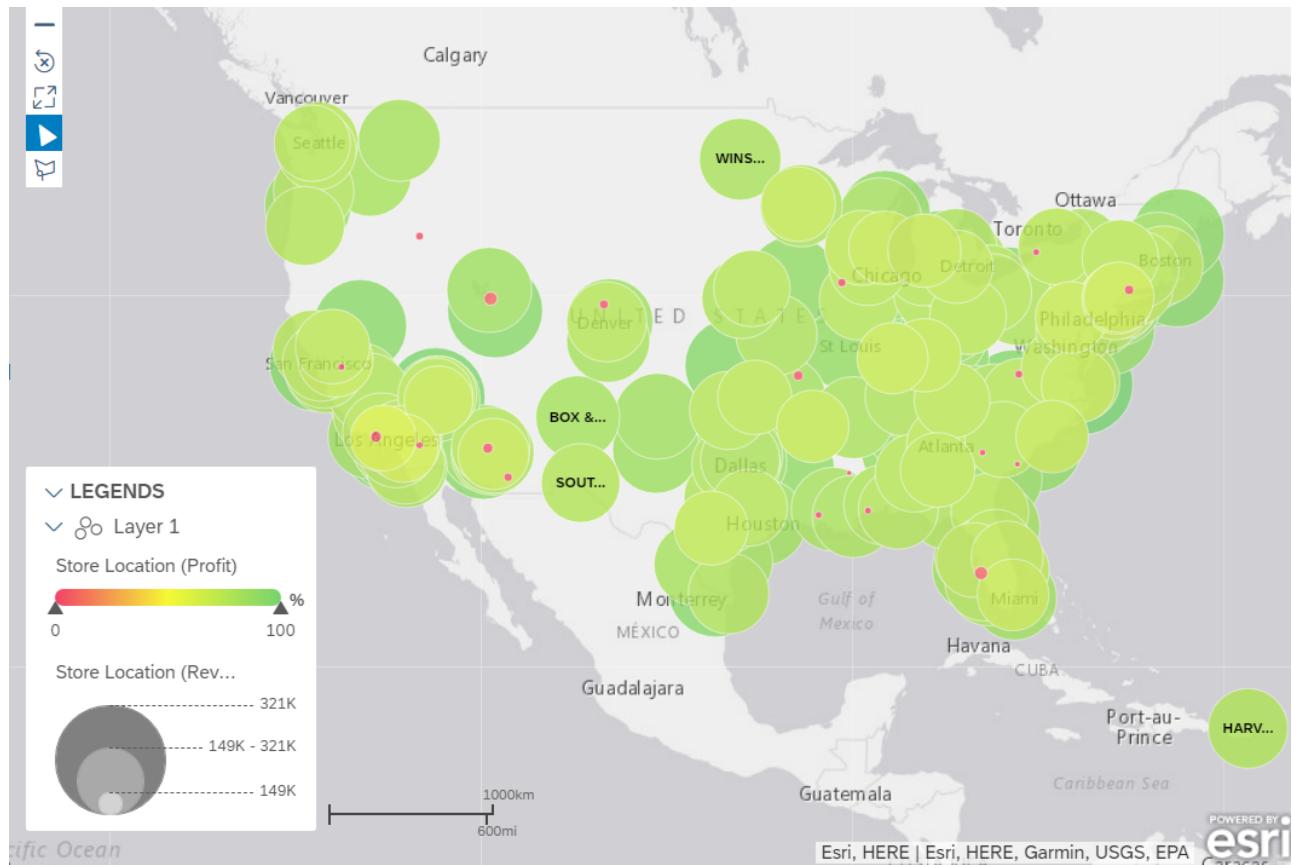
20. Now open the details for the **Bubble Size** definition (Revenue).

21. Set the size to **35%**.



22. Click **Done** to save your Layer.

23. Your map should look like this.

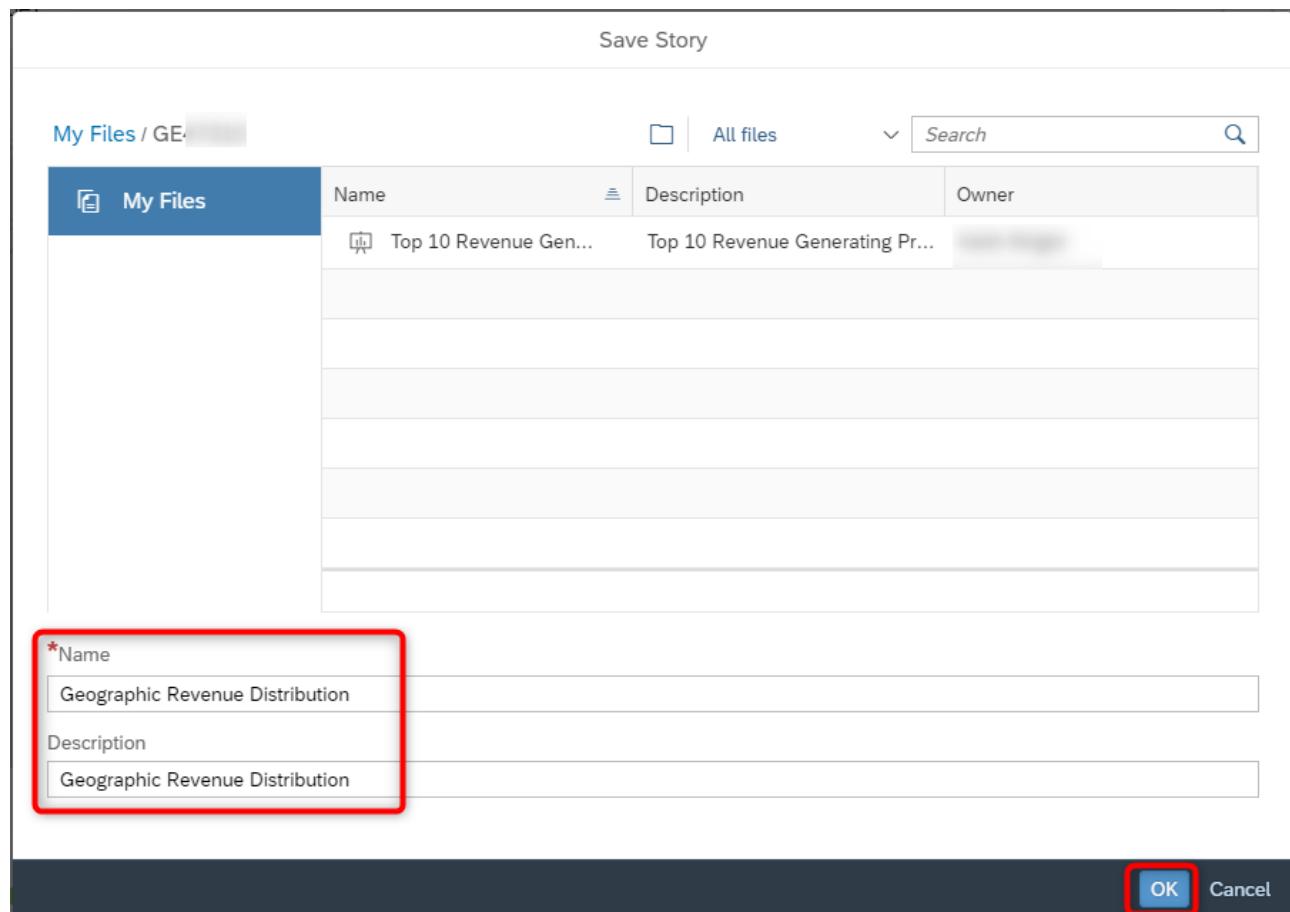


24. In the File menu, select **Save** to save your story.

25. Select the folder that matches your assigned user number, like **GE123456**.

26. Enter a name and description, like **Geographic Revenue Distribution**.

27. Click **Save**.



Summary

You've now created your second story in SAP Analytics Cloud using the preconfigured connection to your data models in SAP Datasphere.

You can continue with one of the *optional* exercises:

- [Exercise 20: Identify Top-Performing Sales Managers with Just Ask](#)
- [Exercise 21: Create Row-Level Permissions based on External Hierarchy\)](#)
- [Exercise 22: Explore the Analytic Model](#)
- [Exercise 23: Create a Transformation Flow\)](#)

Exercise 20 - Identify Top-Performing Sales Managers with Just Ask



Note: This is an OPTIONAL exercise.



Detour: SAP Analytic Cloud - Just Ask

Just Ask is the natural language query feature powered by artificial intelligence in SAP Analytics Cloud. It offers an easy and efficient way to ask questions about your data and provides fact-based answers in the form of tables and charts.

End of Detour

Start of Exercise

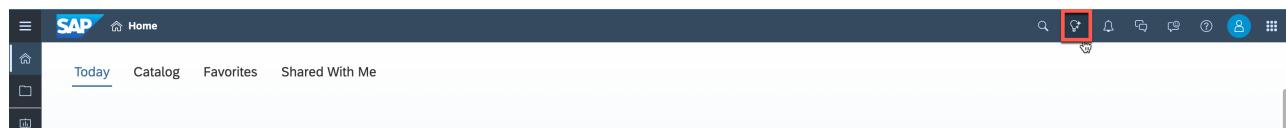
In this exercise, we will utilize Just Ask to query our data model by asking questions in everyday English. Our goal is to identify the best sales managers in our company.

1. Access the SAP Analytics Cloud tenant you got access to as part of the basic trial.

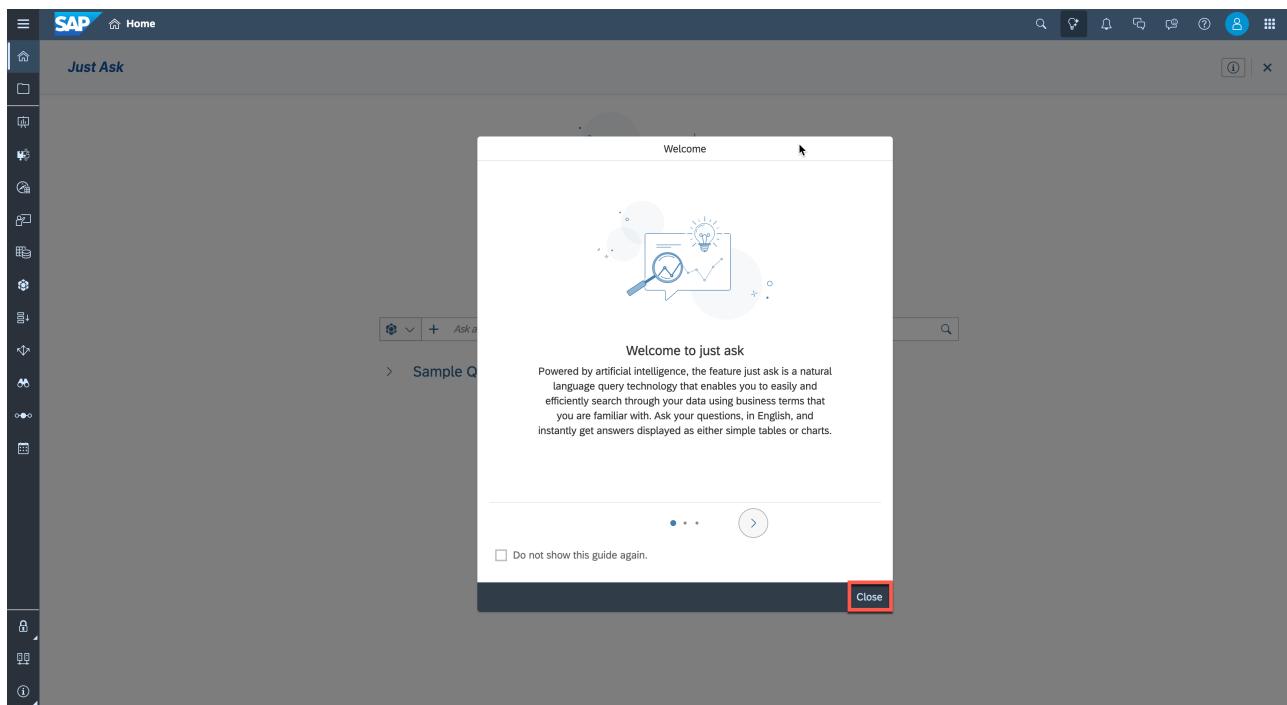


Note: The system may ask you to sign in again. Use the same user name and password for SAC as for Datasphere.

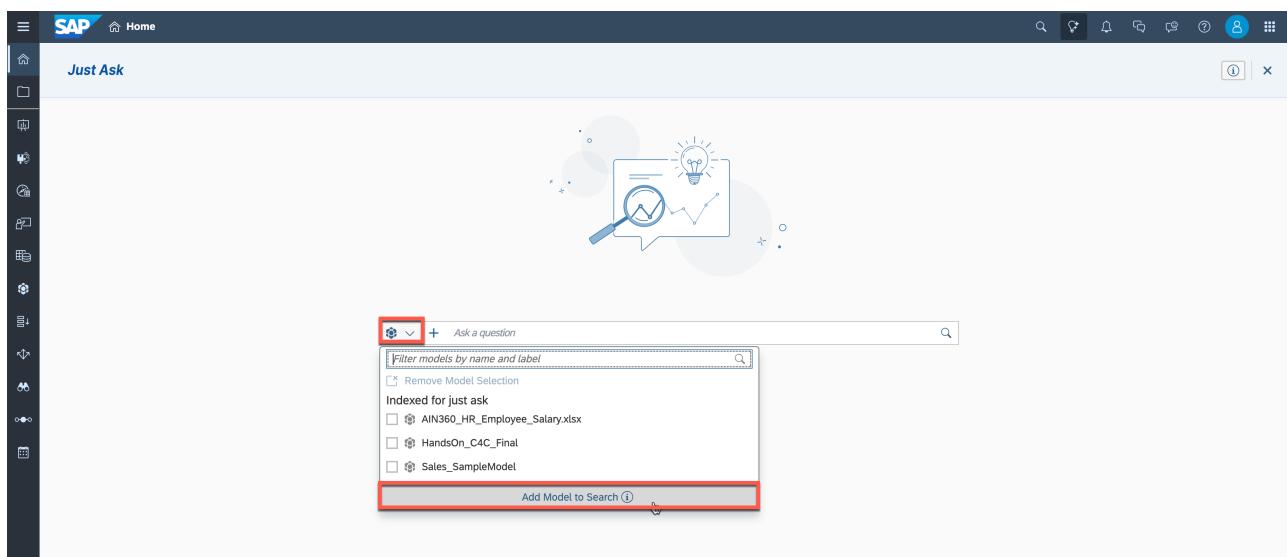
2. Select the menu **Just Ask** in the header.



3. Close the welcome guide, if opens.



4. Click the model icon left to the search bar and select "Add Model to Search".



5. Select **DSP** as the connection on the left panel, choose your space, and click on the previously created Analytic Model **Sales_Analytic_Model**.

The screenshot shows the SAP Datasphere interface with the title "Select Dataset or Model". The left sidebar has sections for "My Files", "Views", "Catalog", "Datasphere", and two items under "DWC": "DWC" and "DWCPB". The "DWC" item is highlighted with a red box and an arrow points to it from the text above. The main area displays a table with columns "Name", "Description", and "Owner". One row is visible: "Sales__Analytic_Model", "Sales - Analytic Model", and "GE135002 GE135002". A search bar at the top right contains the placeholder "Search".

6. The **Successfully Added** message will appear with the information that this model will only be available during the current sessions. Click **OK**.

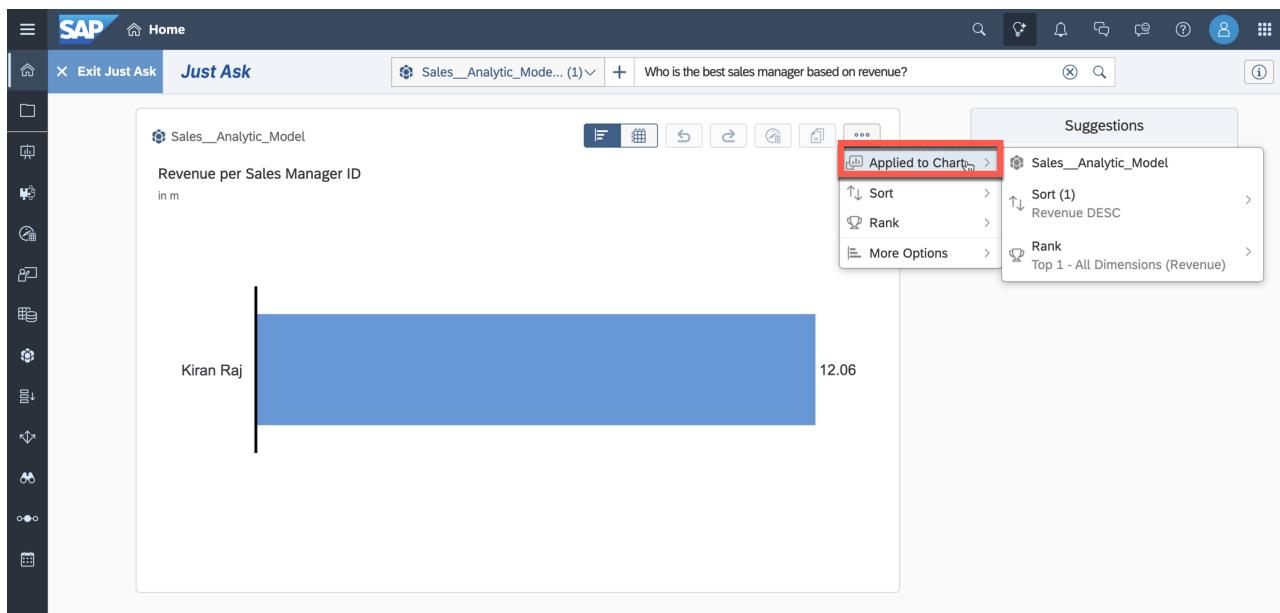
7. Now enter your question into the search bar: ***Who is the best sales manager based on revenue?***

To run the search, press enter or click on the magnifying glass symbol.

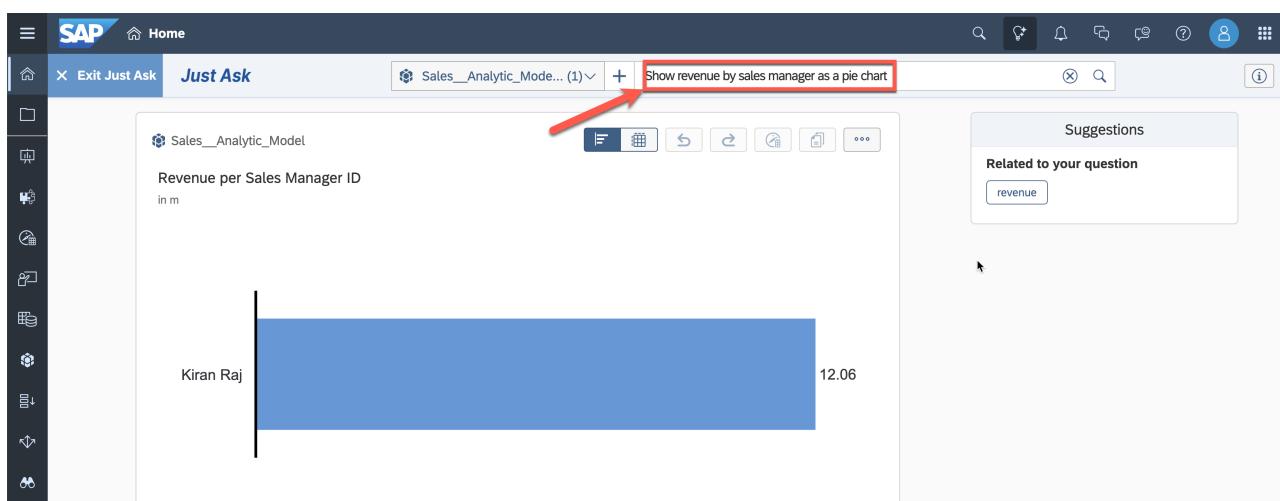
The screenshot shows the SAP Just Ask interface. The top navigation bar includes "SAP Home", "Just Ask" (which is highlighted), and other options like "Exit Just Ask". The main search bar contains the question "Who is the best sales manager based on revenue?". Below the search bar, there's a decorative graphic of a lightbulb and a magnifying glass over a chart. The left sidebar has various icons for navigation.

8. The search result is displayed as a card under the search field.

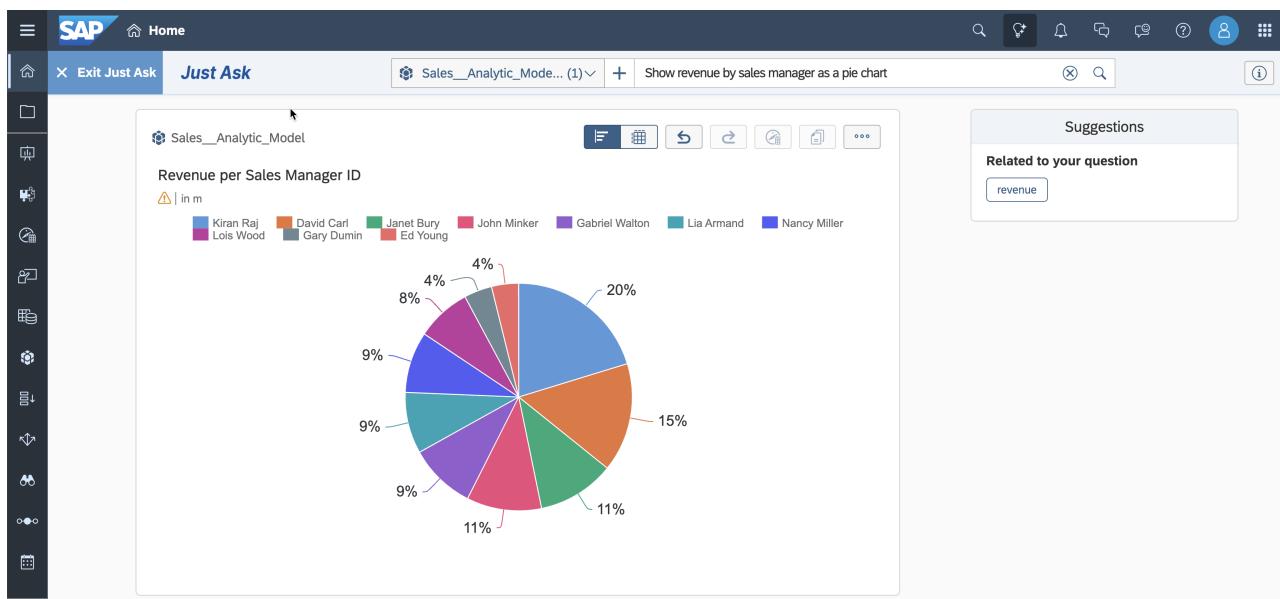
Kiran Raj is identified as the sales manager generating the most revenue. You can review the automatically applied settings, such as sorting and ranking, in addition to the correctly identified attribute and measure.



9. Compare him with the other sales managers. This time, the result should be displayed as a pie chart.
Enter the following in the search bar on the top and search: **Show revenue by sales manager as a pie chart**.

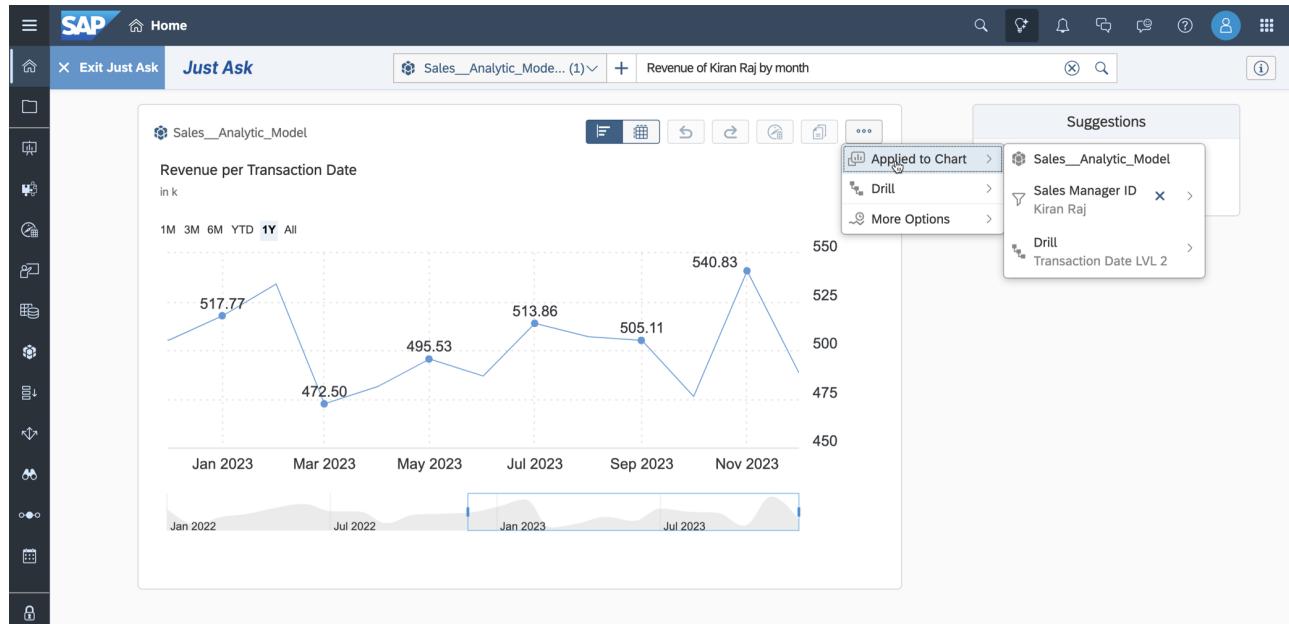


10. The pie chart below is displayed. Kiran Raj is responsible for 20% of the generated revenue.

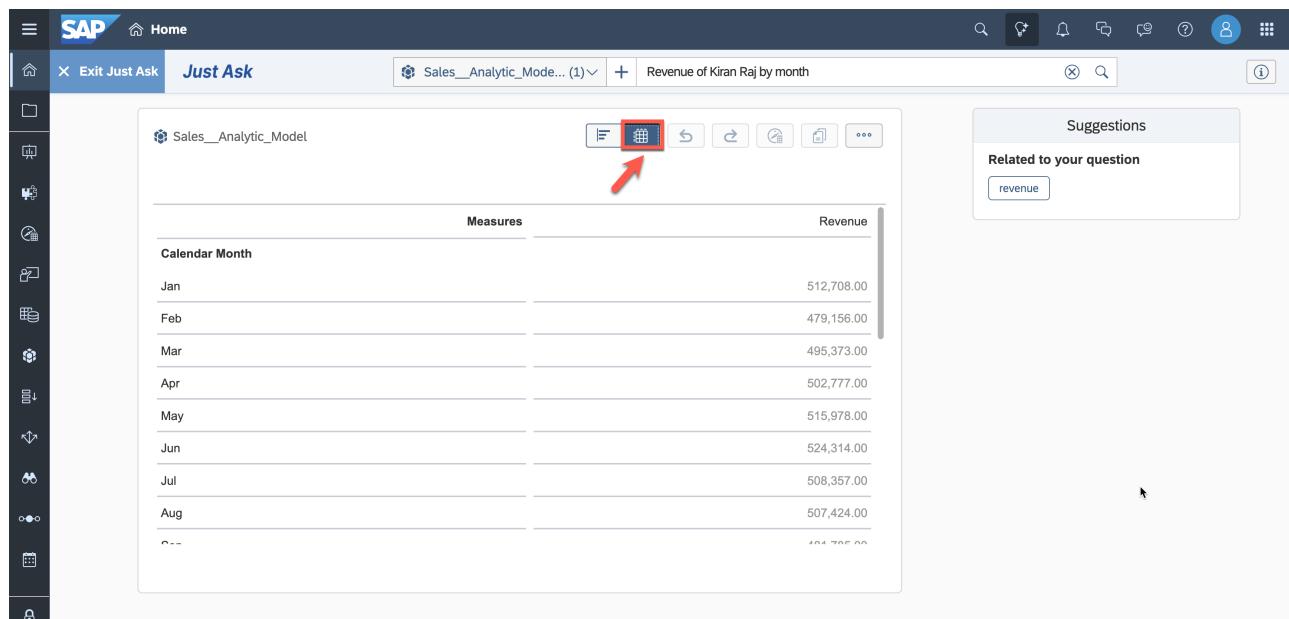


11. Can we see peaks for the revenue generated by Kiran Raj? Enter **Revenue of Kiran Raj by month** and search.

12. The chart visualizes the change of generated revenue. A filter is defined for sales manager and the drill level for the transaction date was adjusted to display months.



13. If you prefer to view the query result in a table format, you can change the display mode to a table.



Summary

Just Ask simplified and accelerated the process of finding answers to your reporting questions. Using the data model exposed by SAP Datasphere, you were able to obtain answers quickly without requiring any preconfigured data report.

You can continue with one of the *optional* exercises:

- [Exercise 21: Create Row-Level Permissions based on External Hierarchy](#)
- [Exercise 22: Explore the Analytic Model](#)

- Exercise 23: Create a Transformation Flow

Exercise 21 - Create Row-Level Permissions based on an External Hierarchy



Note: This is an OPTIONAL exercise.



Detour: SAP Datasphere - Hierarchies and Data Access Controls

You can specify the following [types of hierarchies](#):

1. **Parent-Child** - the hierarchy is recursive, may have any number of levels, and is defined by specifying a parent column and a child column within the dimension. For example, a departmental hierarchy could be modeled with the Parent Department ID and Department ID columns.
2. **Level-Based** - the hierarchy is non-recursive, has a fixed number of levels, and is defined by specifying two or more level columns within the dimension. For example, a time hierarchy could be modeled with the: Year, Quarter, Month, Week, and Day columns.
3. **External Hierarchy** - the parent-child hierarchy information is contained in a separate entity, which needs to be associated with the dimension (see [Create an External Hierarchy for Drill-Down](#)).
4. **Hierarchy with Directory** - your entity contains one or more parent-child hierarchies and has an association to a directory dimension containing a list of the hierarchies. These types of hierarchy entities can include nodes from multiple dimensions (for example, country, cost center group, and cost center) and are commonly imported from SAP S/4HANA Cloud and SAP BW (including SAP BW Bridge) systems.

Data access controls allow you to apply row-level security to your objects. There are different options for specifying criteria to determine which user is allowed to access specific data. One option is to define access based on a hierarchy. Each user can only view records that match the hierarchy values they are authorized for in the permissions entity, along with any of their descendants. Only external hierarchies with a single pair of parent-child columns are supported.

End of Detour

Create a Parent-Child Hierarchy

1. Log On to your SAP Datasphere tenant.
2. Select the menu option **Data Builder** on the left-hand side.
3. Select the option **New Table**.
4. Enter the following details:
 - **Business Name:** Hierarchy_ProductCategory
 - **Technical Name:** Hierarchy_ProductCategory
 - **Semantic Usage:** Hierarchy

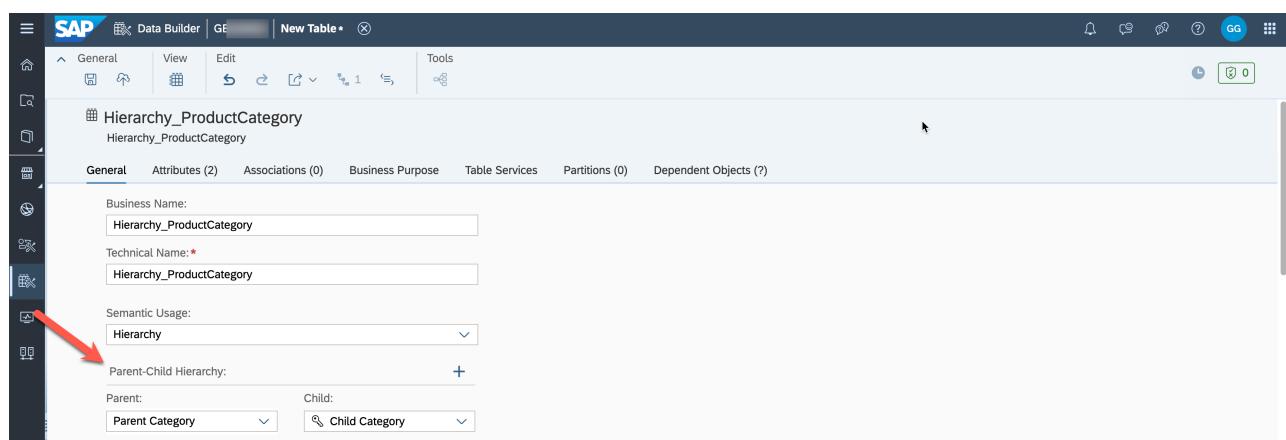
5. Warnings are displayed because no parent and child columns are defined yet to represent the hierarchy. The child column must be a key column, and both the parent and child columns must have the same data type.

6. Enter the following **Attributes** for the table:

Key:	Business Name:	Technical Name:	Data Type:
X	Child Category	Child_Category	String (30)
	Parent Category	Parent_Category	String (30)

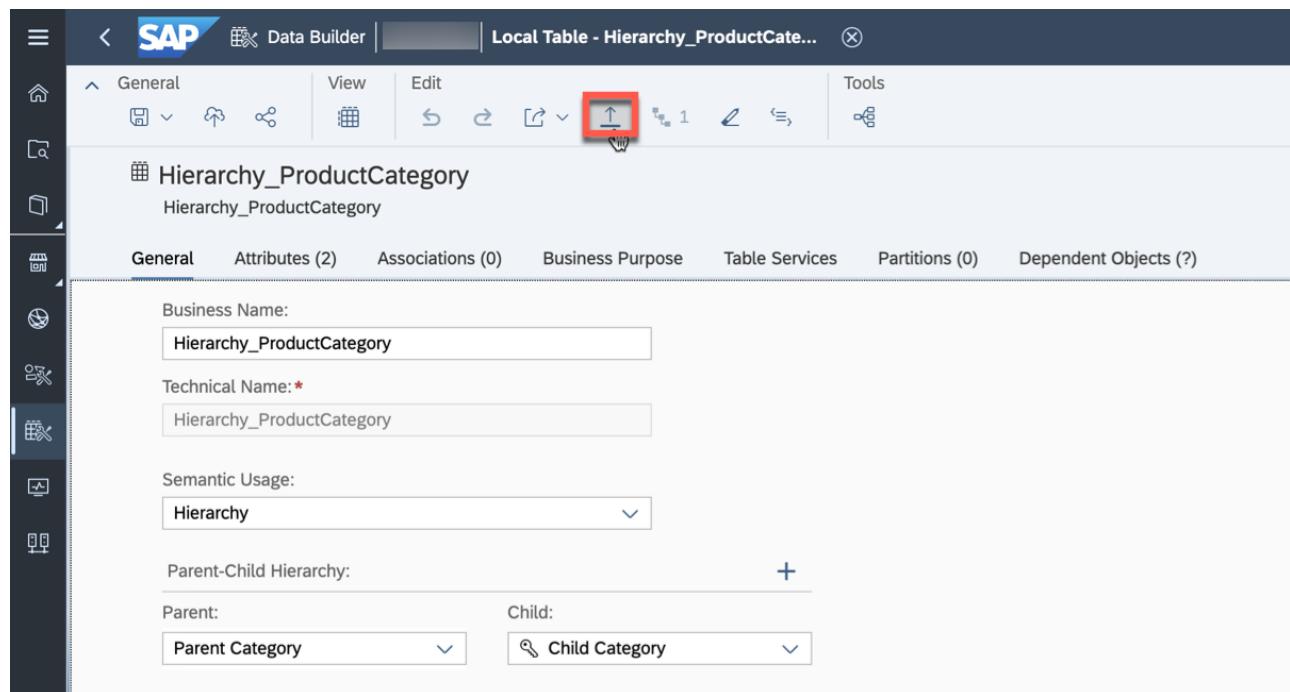
7. Ensure that **Child Category** is set as key.

8. Now select the column **Parent Category** as the parent and the column **Child Category** as the child. Notice that the validation icon in the upper-right corner turns green.



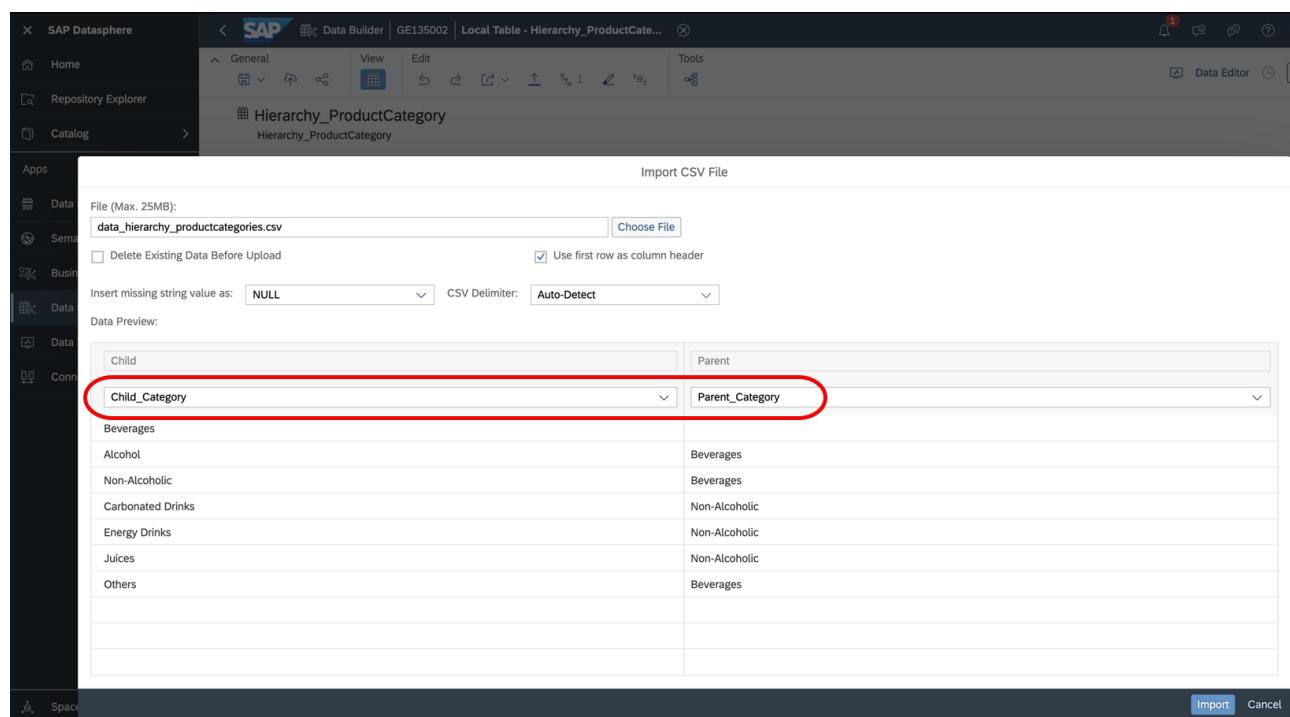
9. Save and deploy your hierarchy table in the folder **TECHED2024-DA180**.

10. Download the CSV file **data_hierarchy_productcategories.csv** and upload the data to the table.



The screenshot shows the SAP Data Builder interface with the title "Local Table - Hierarchy_ProductCategory". The "General" tab is selected. The "Business Name" field contains "Hierarchy_ProductCategory". The "Technical Name" field also contains "Hierarchy_ProductCategory". Under "Semantic Usage", the dropdown is set to "Hierarchy". In the "Parent-Child Hierarchy" section, "Parent" is set to "Parent Category" and "Child" is set to "Child Category".

Make sure to map the columns from the CSV file to the corresponding columns in the table.



The screenshot shows the SAP Datasphere interface with the title "GE135002 | Local Table - Hierarchy_ProductCategory". The "Import CSV File" dialog is open, showing a preview of data from "data_hierarchy_productcategories.csv". The "Child" and "Parent" columns are circled in red. The data preview shows the following rows:

Child	Parent
Beverages	Beverages
Alcohol	Beverages
Non-Alcoholic	Non-Alcoholic
Carbonated Drinks	Non-Alcoholic
Energy Drinks	Non-Alcoholic
Juices	Non-Alcoholic
Others	Beverages

11. Verify the data in the data preview.

The screenshot shows the SAP Data Builder interface with the following details:

- Title Bar:** SAP Data Builder - Local Table - Hierarchy_ProductCategory...
- Toolbar:** General, View, Edit, Tools.
- Table Structure:**

Business Name	Technical Name	Data Type	Text / Association	Default Value	Not Null
Child Category	Child_Category	String(30)			<input checked="" type="checkbox"/>
Parent Category	Parent_Category	String(30)		Enter a string	<input type="checkbox"/>
- Data View:** A grid showing the hierarchy data:

Child Category	Parent Category
Alcohol	Beverages
Beverages	NULL
Carbonated Drinks	Non-Alcoholic
Energy Drinks	Non-Alcoholic
Juices	Non-Alcoholic
Non-Alcoholic	Beverages
Others	Beverages

Create a Permission Entity

A permission entity (view or table) lists SAP Datasphere user IDs (in the form required by your identity provider) and assigns them to one or more criteria. In this example, you will define criteria for your own user so that you only see products belonging to the category or subcategories of non-alcoholic beverages.

1. Select the option **New Table** in the Data Builder.
2. Enter the following details:
 - **Business Name:** Permission Entity Product Category
 - **Technical Name:** PE_ProductCategory
 - **Semantic Usage:** Relational Dataset

3. Enter the following columns:

Key:	Business Name:	Technical Name:	Data Type:	Text/Association
X	Record ID	Record_ID	Integer	
	User ID	User_ID	String (50)	
	Product Category	Product_Category	String (30)	

4. Verify your configuration.

General

Business Name: Permission Entity Product Category

Technical Name: PE_ProductCategory

Semantic Usage: Relational Dataset

Delta Capture: OFF

Status: Deployed

Deployed On: Sep 25, 2024 22:16:38

Columns (3)

	Business Name	Technical Name	Data Type	Text / Association	Default Value	Not Null
<input type="checkbox"/>	Record ID	Record_ID	Integer		Enter a string	<input checked="" type="checkbox"/>
<input type="checkbox"/>	User ID	User_ID	String(50)		Enter a string	<input type="checkbox"/>
<input type="checkbox"/>	Product Category	Product_Category	String(30)		Enter a string	<input type="checkbox"/>

5. Save and deploy the table in the folder **TECHED2024-DA180**.

6. After the table is deployed, open the data editor to add records.

General

Business Name: Permission Entity Product Category

Technical Name: PE_ProductCategory

Semantic Usage: Relational Dataset

Delta Capture: OFF

Status: Deployed

7. Add the following record to the table. Replace the first part of the email address below for the user ID with the name of your user (same as the space name). Alternatively, you can copy your email address by going to **Profile** in the upper-right corner and selecting **Settings**, then **User Account**.

GE114169

Permission Entity Product Category

Record ID: 1, User ID: ge114169@sapexperienceacademy.com, Product Category: Non-Alcoholic

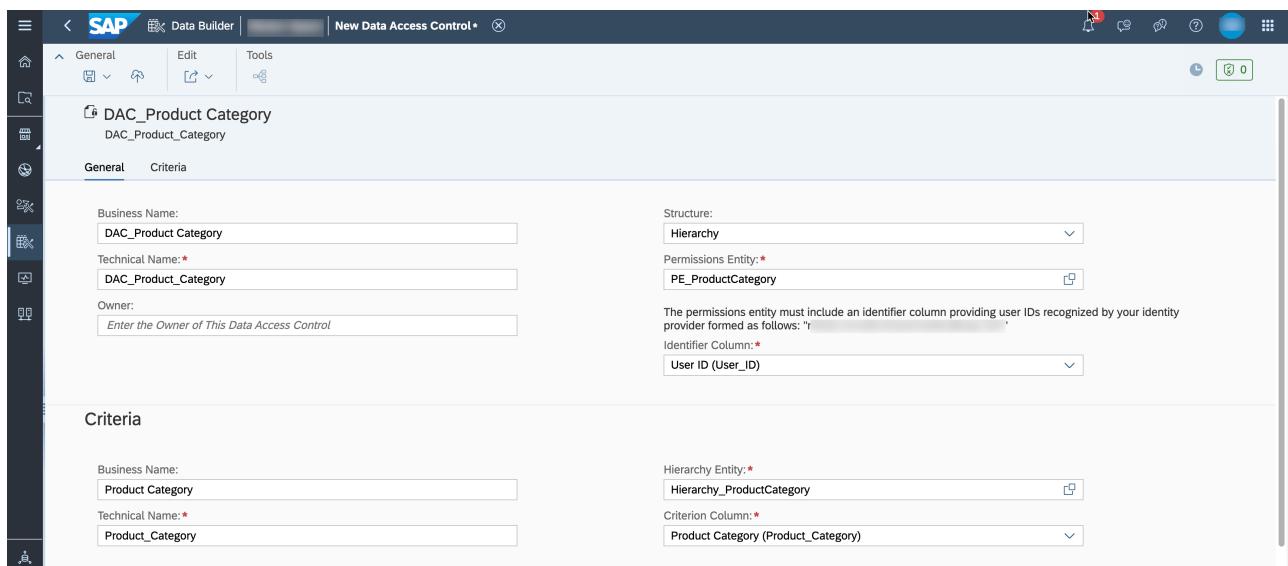
Record ID	User ID	Product Category
1	geXXXXXX@sapexperienceacademy.com	Non-Alcoholic

8. Save the added record.

Create a Data Access Control Entity

We want to protect the data of our sales transactions based on this parent-child hierarchy. Reporting users should only see sales for the product categories they have permissions to. Permissions will be defined for the nodes **Alcohol**, **Non-Alcoholic**, and **Others**.

1. In the side navigation area, select **Data Builder** and click **New Data Access Control** to open the editor.
2. Enter the following details in the **General** section and select the permission entity which you created in the previous step:
 - **Business Name:** DAC Product Category
 - **Technical Name:** DAC_ProductCategory
 - **Structure:** Hierarchy
 - **Permission Entity:** PE_ProductCategory
 - **Identifier Column:** User ID
3. Enter the following details in the **Criteria** section and select the hierarchy table which you created in the first step of this exercise:
 - **Business Name:** Product Category
 - **Technical Name:** Product_Category
 - **Hierarchy Entity:** Hierarchy_ProductCategory
 - **Criterion Column:** Product Category
4. Verify that your settings look like in the screenshot below:

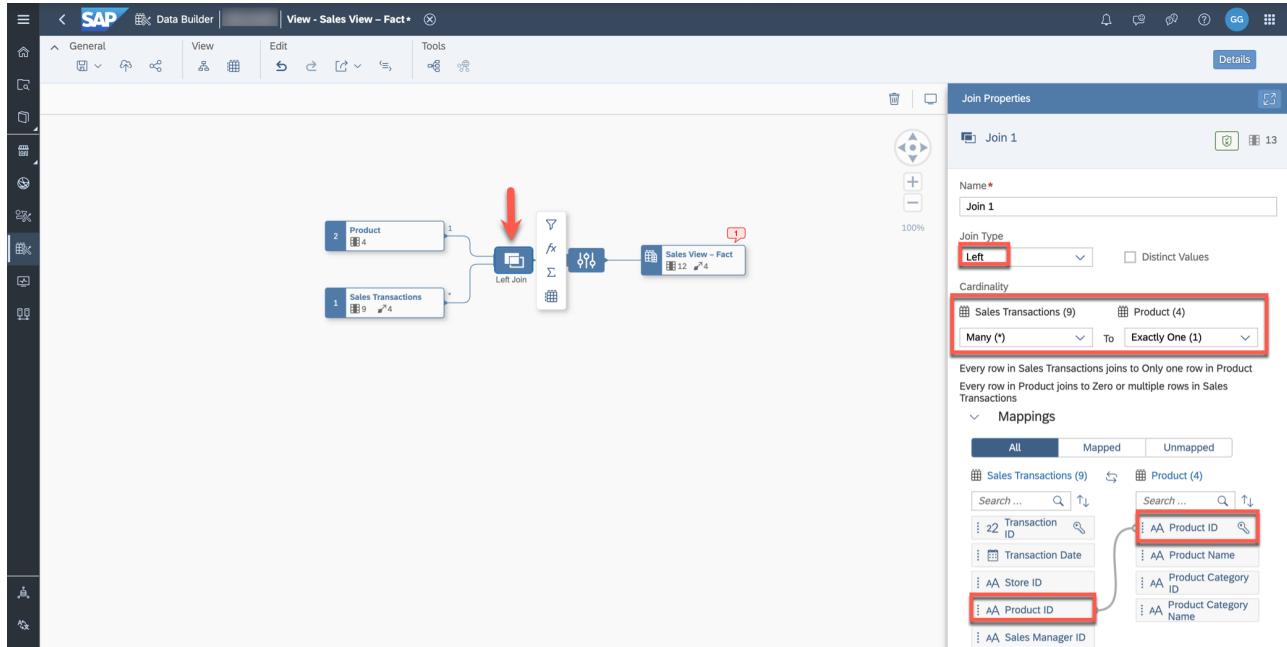


5. Save and deploy your new Data Access Control entity in the folder **TECHED2024-DA180**.

Apply the Data Access Control Entity to a View

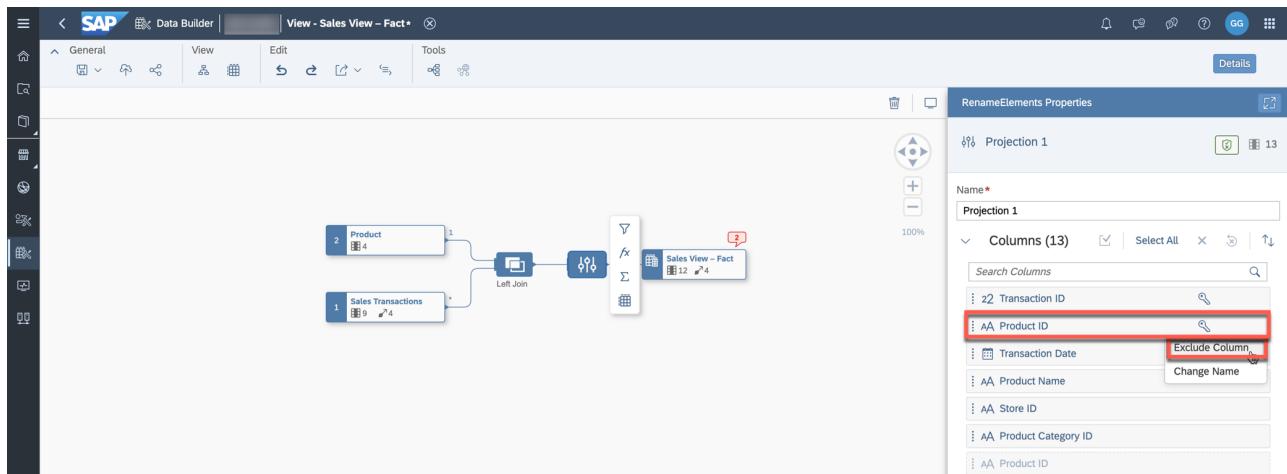
1. Open your previously created view **Sales View – Fact** (**Sales_View_Fact**).

2. The attributes which are authorization-relevant need to be part of the fact view. Drag the table **Product** into the editor over the ***Sales Transactions** node. The join should be created automatically.
3. Set the Join Type to **Left** and the cardinality to Many (**Sales Transactions**) to Exactly One (**Product**). Map the columns **Product ID**.



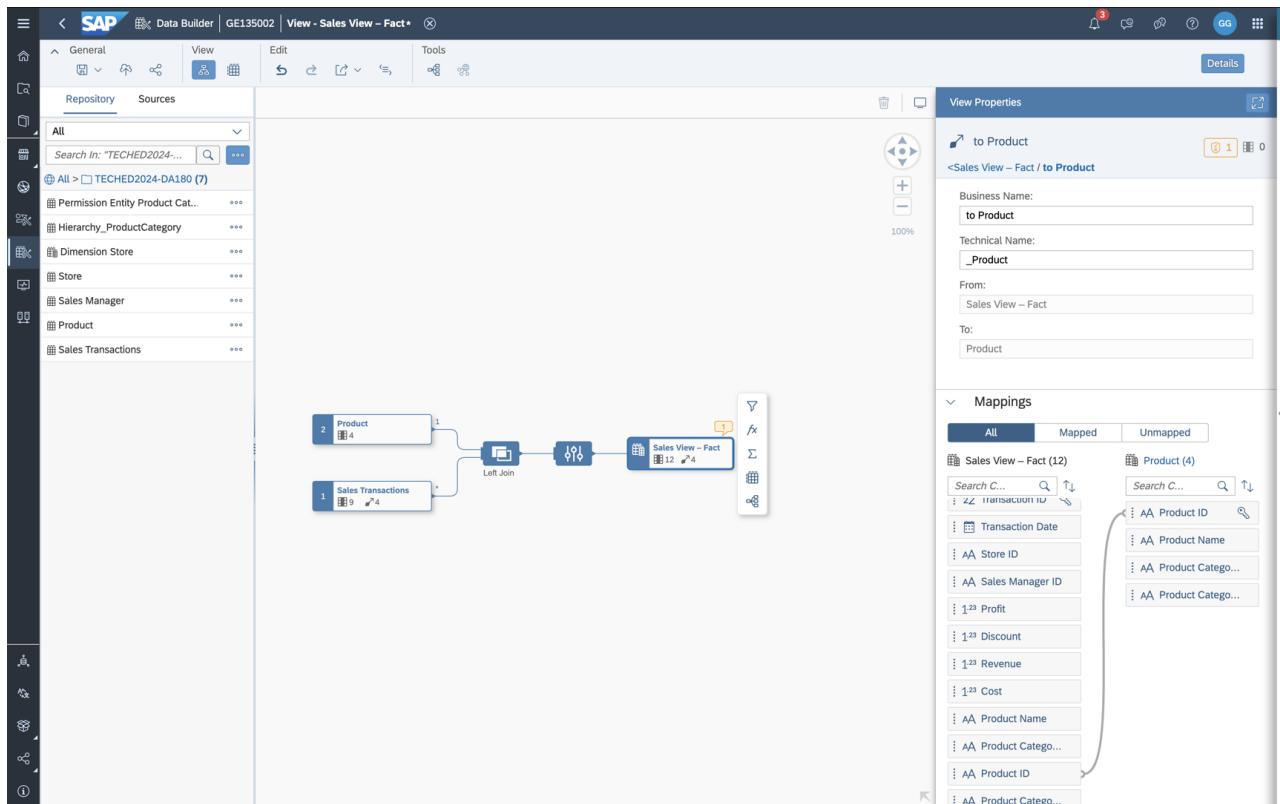
4. Open the Projection Node. The column **Product ID** of the table **Sales Transactions** is hidden while the column derived from the table **Product** (dimension) is visible.

We do not want to keep **Product ID** as a key, therefore exclude the column **Product ID** (Key from **Products**) and restore **Product ID (Sales Transactions)**.

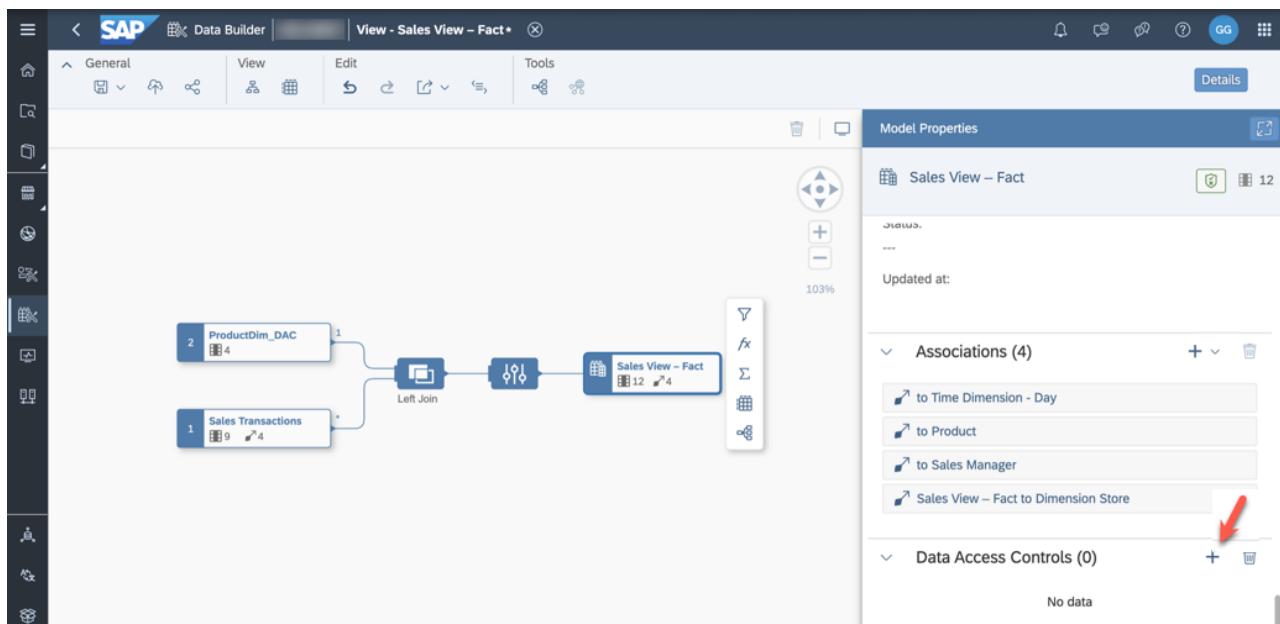


If you select the column in any diagram node you can trace the lineage back to the source.

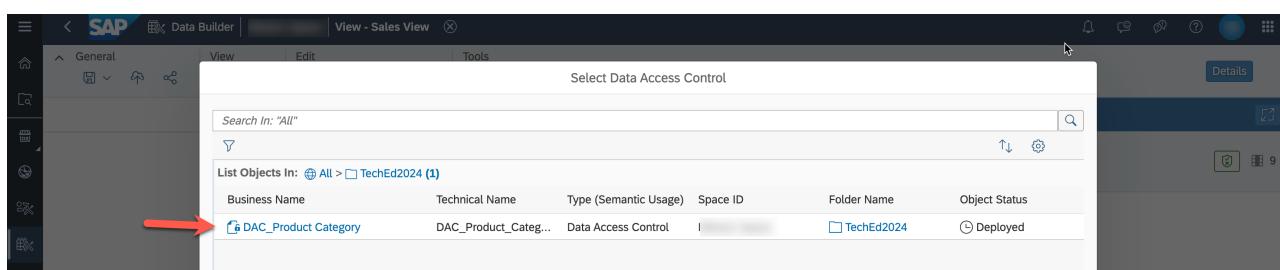
5. Open the node **Sales View - Fact**. Go to the association which points to the dimension **Product** and redo the mapping (**Product ID** to **Product ID**) as this got lost because of the join introduced above.



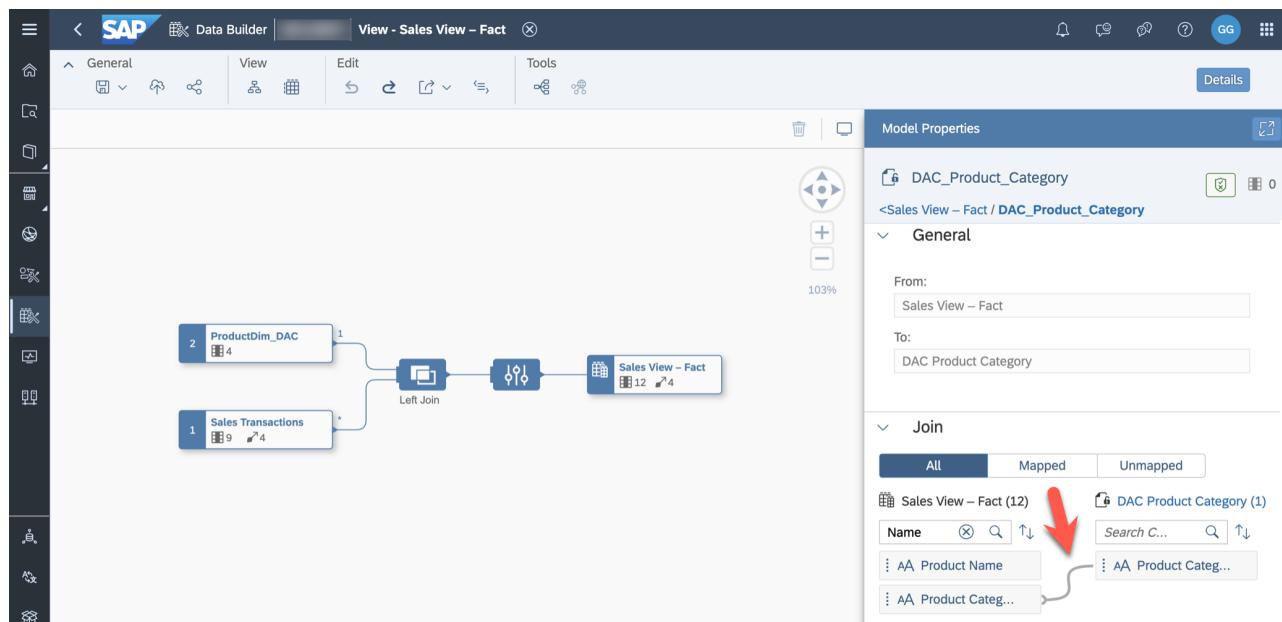
6. In the model properties, add a new Data Access Control.



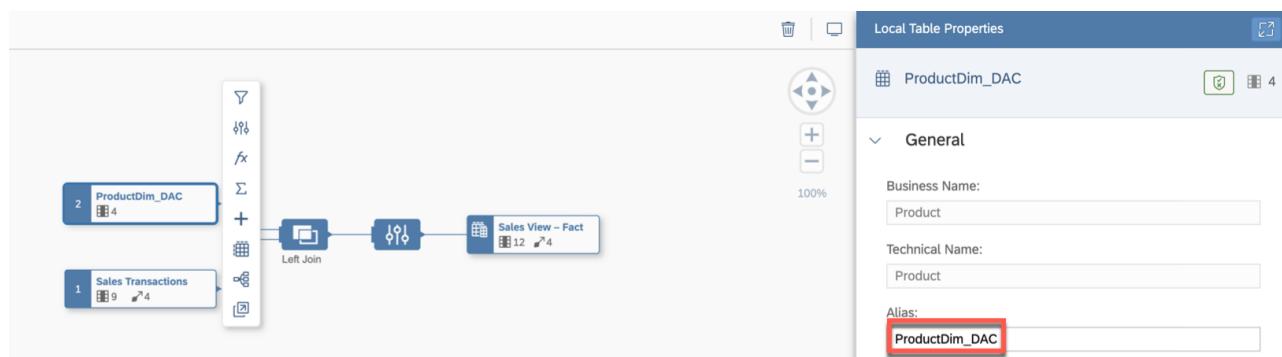
7. Select the previously created DAC **DAC_ProductCategory** (single click).



8. Map the two columns named **Product Category Name** (enter "Name" in the **Search** field). When you move your cursor over the columns, the complete name is displayed as tooltip.



9. Set the Alias of the node **Product** to **ProductDim_DAC**.



10. Save and deploy the view **Sales_View - Fact**. If you see the validation message warning that an association has been changed, click **Deploy Anyway**.

11. Open the data preview for the node **Sales View - Fact** and validate that you can only see sales for non-alcoholic data products.

The screenshot shows the SAP Data Builder interface with the title "View - Sales View - Fact". The left pane displays the data flow diagram. The right pane shows the "View Properties" for the view "Sales View - Fact". The "Semantic Usage" is set to "Fact". Below it, the "Data Validation" section shows a table with data. A red box highlights the "View Properties" tab and the data table. The data table contains the following rows:

Discount	Revenue	Cost	Product Name	Product Category ID	Product ID	Product Category Name
278.95	797.00	87.67	Apple Juice	PC2	PD9	Juices
137.92	862.00	103.44	Orange Crush	PC1	PD6	Carbonated Drinks
51.38	367.00	25.69	Monster	PC3	PD19	Energy Drinks
129.25	517.00	62.04	Ginger Ale	PC1	PD4	Carbonated Drinks
150.90	503.00	35.21	Ginger Ale	PC1	PD4	Carbonated Drinks
125.76	393.00	31.44	Apple Juice	PC2	PD9	Juices

12. Add a filter for **Product Category Name**.

The screenshot shows the SAP Data Builder interface. At the top, there's a toolbar with various icons. Below it is a navigation bar with 'General', 'View', 'Edit', and 'Tools' tabs. On the left, there's a sidebar with icons for file operations like 'New', 'Open', 'Save', etc. The main area displays a data flow diagram with three components: 'ProductDim_DAC', 'Sales Transactions', and 'Sales View - Fact'. A 'Left Join' is performed between 'ProductDim_DAC' and 'Sales Transactions'. The output of this join is then joined with 'Sales View - Fact'. The 'Sales View - Fact' component has four measures: Profit, Discount, Revenue, and Cost. In the bottom right corner of the main area, there's a 'View Properties' panel for 'Sales View - Fact'.

Discount	Revenue	Cost	Product Name	Product Category ID	Product ID	Product Category Name
278.95	797.00	87.67	Apple Juice	PC2	PD9	Sort Ascending
137.92	862.00	103.44	Orange Crush	PC1	PD6	Sort Descending
51.38	367.00	25.69	Monster	PC3	PD19	Filter
129.25	517.00	62.04	Ginger Ale	PC1	PD4	Carbonated Drinks
150.90	503.00	35.21	Ginger Ale	PC1	PD4	Carbonated Drinks
125.76	393.00	31.44	Apple Juice	PC2	PD9	Juices
88.90	635.00	69.85	Orange Crush	PC1	PD6	Carbonated Drinks
119.61	443.00	39.87	Monster	PC3	PD19	Energy Drinks
163.20	816.00	122.40	Ginger Ale	PC1	PD4	Carbonated Drinks
100.21	911.00	100.21	Ginger Ale	PC1	PD4	Carbonated Drinks

13. Filter on records which contain values equal to **Alcohol**. This search should not display any value.

The screenshot shows a 'Define Filters' dialog. Under the 'Include (1)' section, there is a dropdown for 'Product Category Name', a 'equal to' operator, and a text input field containing 'Alcohol'. Below this, there's a large callout box with the heading 'Important:' and the text: 'Please note that you will now only see a limited number of records due to the assigned data access control. You can modify the data record in the permission table **PE_ProductCategory** and replace the value in the column **Product Category** (Non-Alcoholic) with **Beverages** to see all records again when working through other exercises.'

14. You can open the previously created reports in SAP Analytics Cloud. You will only see the records that your user is authorized to access.

Summary

You've now protected the transactional records of the fact view with a Data Access Control based on product categories. If you want to recreate authorizations from the source system, you can import analysis authorizations from SAP BW and SAP BW/4HANA.

You can continue with one of the optional exercises:

- Exercise 20: Identify Top-Performing Sales Managers with Just Ask
- Exercise 22: Explore the Analytic Model
- Exercise 23: Create a Transformation Flow

Exercise 22 - Explore the Analytic Model



Note: This is an OPTIONAL exercise.



Detour: SAP Datasphere - Analytic Model

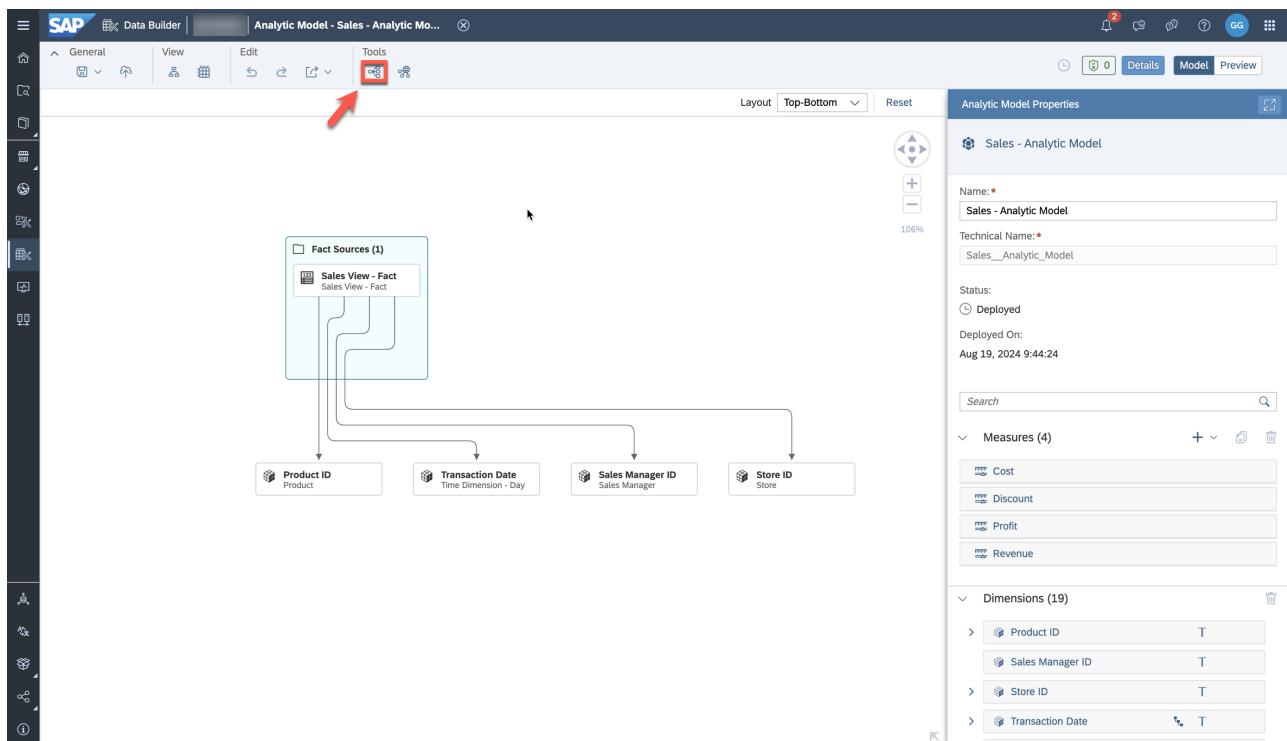
Analytic models form the analytical foundation for preparing data for consumption in SAP Analytics Cloud. They enable the creation and definition of multi-dimensional models, providing data for analytical purposes to answer various business questions. Predefined measures, hierarchies, filters, parameters, and associations offer flexible and straightforward navigation through the underlying data.

In this exercise, you will learn how to use the data preview of the Analytic Model and create different types of new measures to enhance the existing model.

End of Detour

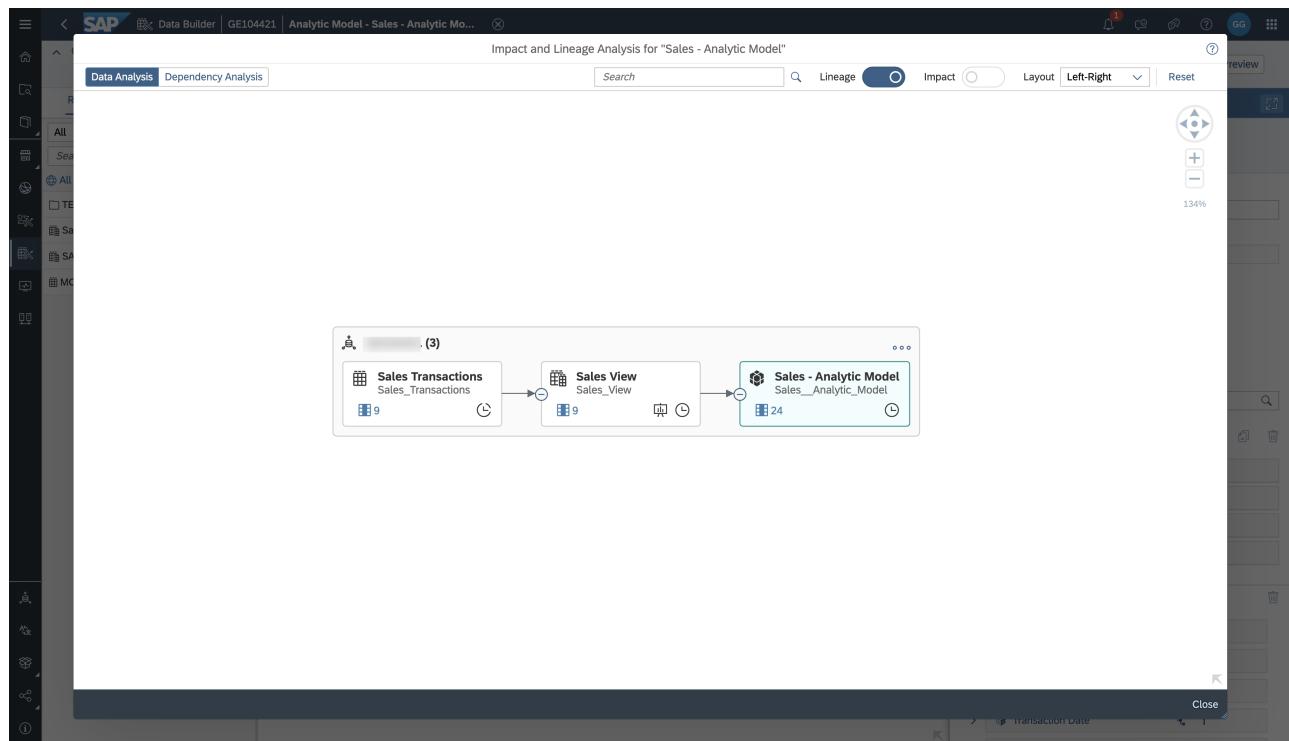
Start of the exercise

1. Log On to your SAP Datasphere tenant.
2. Select the menu option Data Builder on the left-hand side.
3. Open the previously created Analytic Model **Sales - Analytic Model**.
4. To get an overview of the data model as a reminder, open the **Impact and Lineage Analysis**.

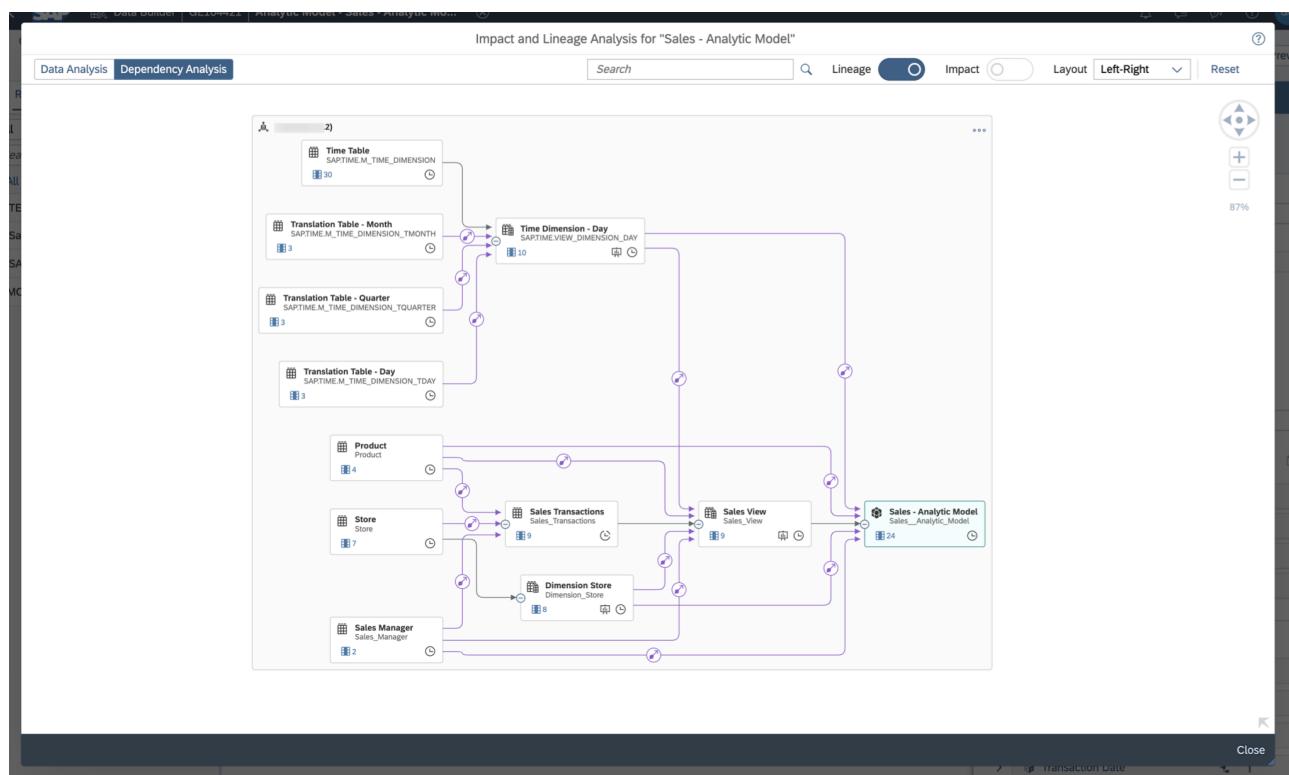


5. The **Data Analysis** diagram helps you understand the lineage of the Analytic Model.

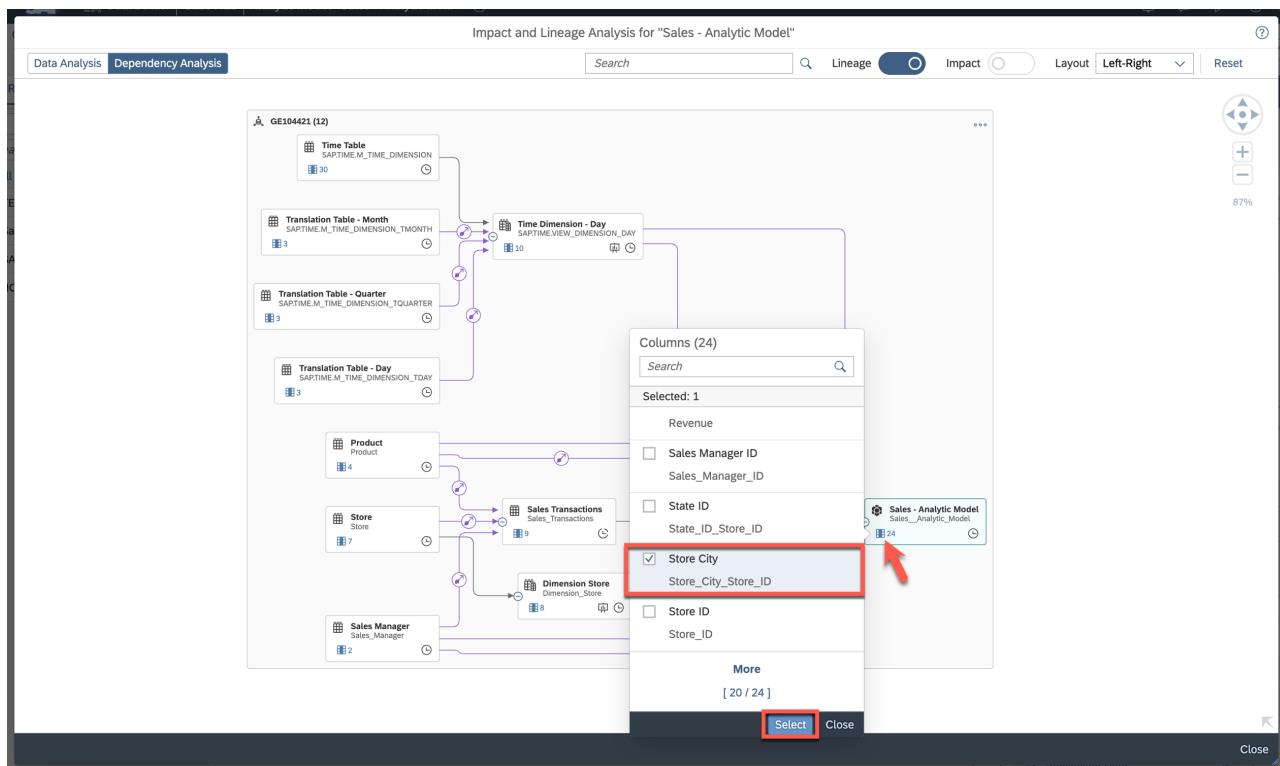
This view focuses on data movements and transformations, showing the flow from the local table containing sales transactions to the view and the Analytic Model.



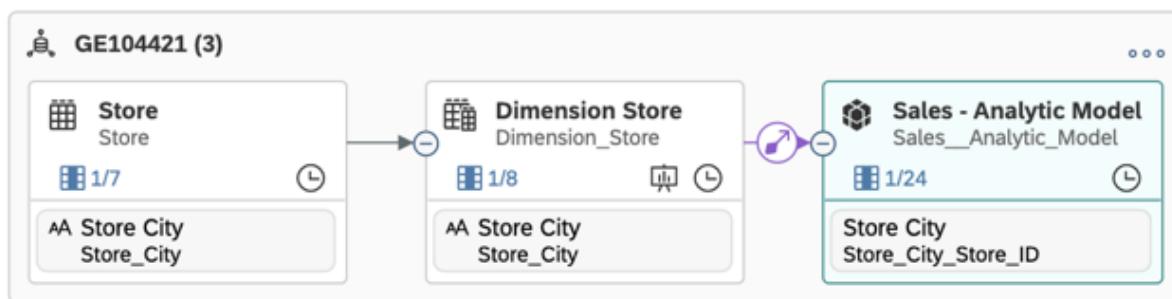
6. Switch to the **Dependency Analysis** diagram to display objects connected through associations (and data access controls, if defined). Here, you can view the lineage of the associations, such as the time dimension.



7. In addition to the lineage and impact of objects, you can analyze columns of tables, graphical and SQL views and Analytic Models. Click the columns icon on the Analytic Model to open the list of columns, select **Store City** and click **Select**.

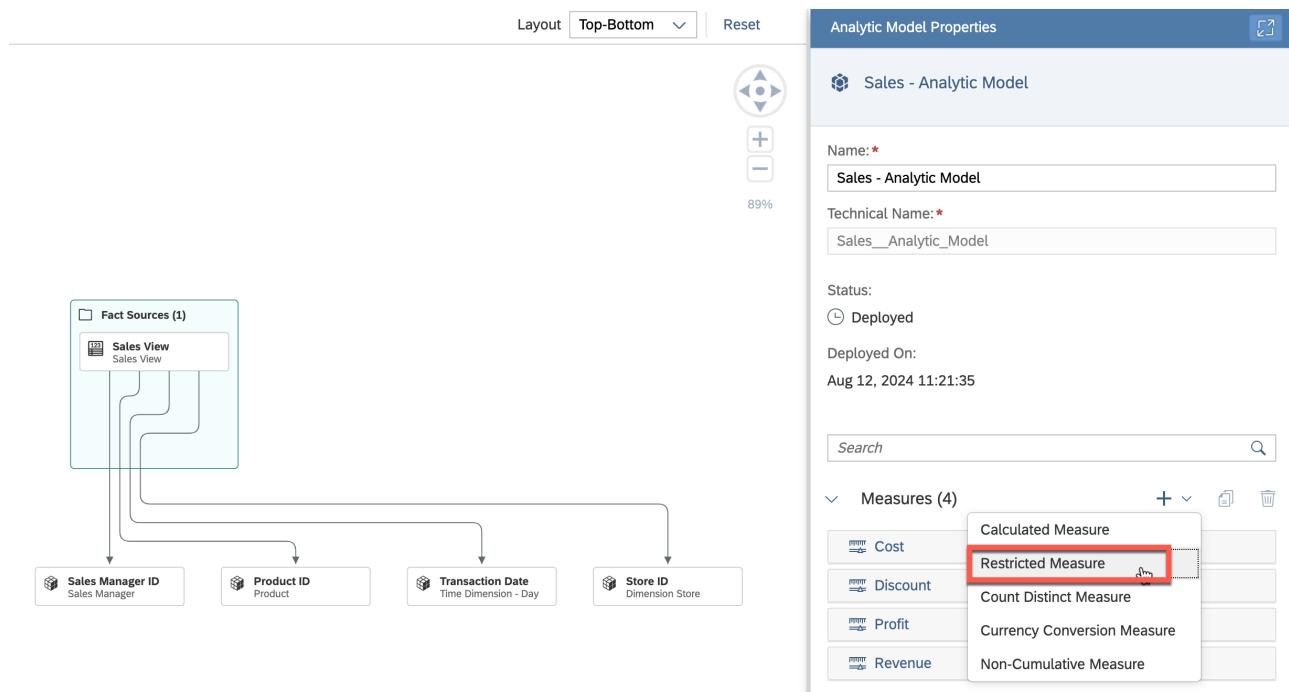


8. The diagram refreshes to show the lineage of the selected column under the analyzed object.



9. Select **Close** to return to the Analytic Model editor.

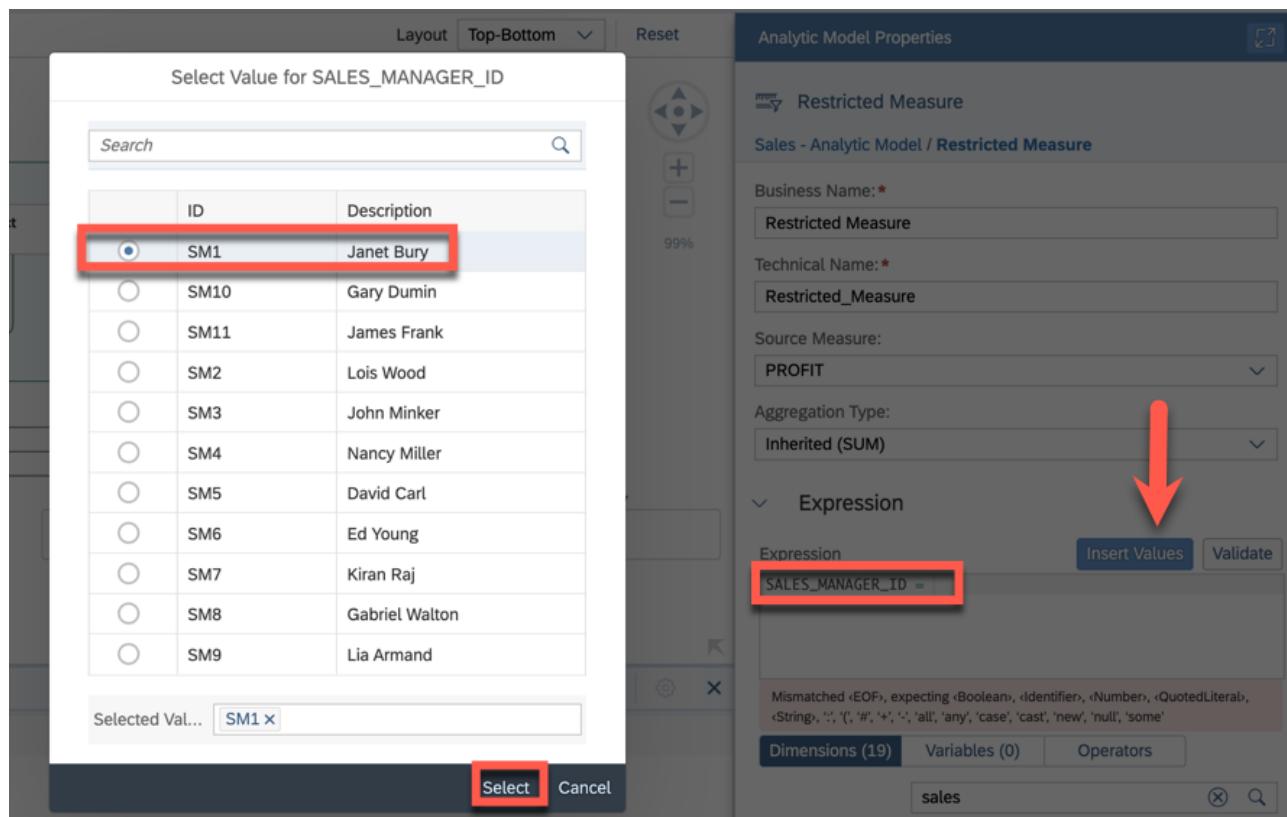
10. Create a new **Restricted Measure**. Restricted measures are based on existing measures but apply flexible filter expressions. They can use static filters (as done in this exercise) or refer to dimension attributes or variables.



11. We define a restricted measure to analyze the profit generated by the sales managers, specifically focusing on [Janet Bury](#).

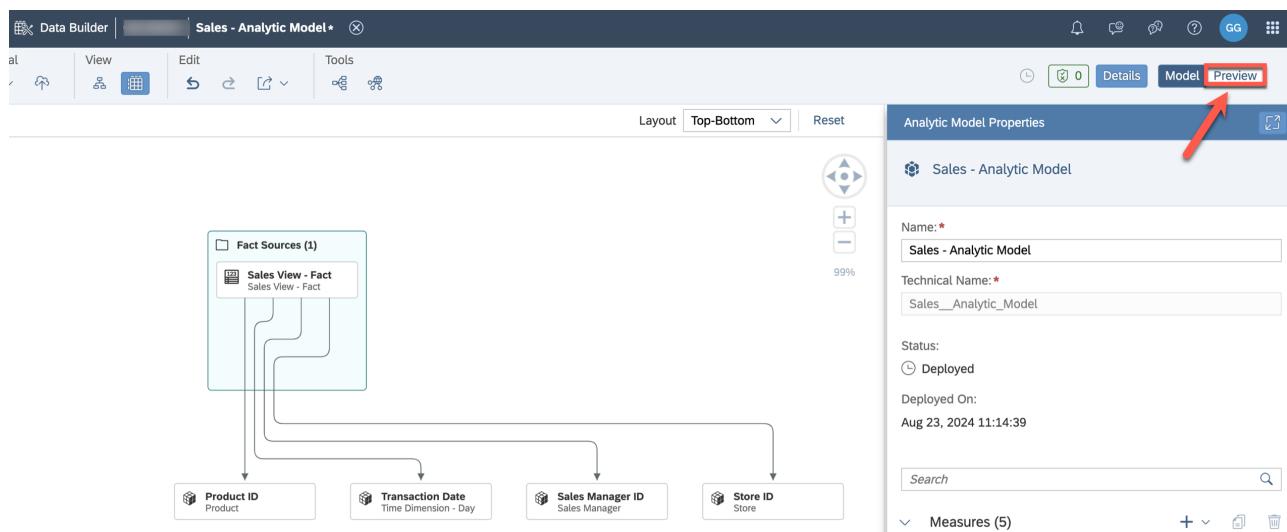
The screenshot shows the 'Restricted Measure' configuration dialog. It includes fields for 'Business Name' (set to 'Restricted Measure'), 'Technical Name' (set to 'Restricted.Measure'), 'Source Measure' (set to 'PROFIT'), and 'Aggregation Type' (set to 'Inherited (SUM)'). Under the 'Expression' section, the expression 'SALES_MANAGER_ID' is entered in the 'Expression' field. Below this, a list of dimensions is shown, with 'SALES_MANAGER_ID (Fact Sources)' highlighted and a red arrow pointing to it.

12. In the **Expression** editor, add = after **SALES_MANAGER_ID**. Click **Insert Values** and select [Janet Bury](#) (ID [SM1](#)).



13. Change the Business Name and the Technical Name to **Profit-JanetBury**.

14. Open the **Preview** in the upper-right corner. This is an analytical preview that allows you to navigate through various dimensions and view the aggregated data. It provides a glimpse of how the data will appear in an SAP Analytics Cloud story.



15. Open the Builder on the right side and select the measures **Profit** and **Profit-JanetBury**, as well as **Product Category ID** in the **Dimensions** section. The two measures allow you to compare the profit generated by Janet with the overall profit.

The screenshot shows the Analytic Model editor interface. On the left, a pivot table titled "Sales - Analytic Model" displays data for Product Category ID (PC1 to PC5) across measures PROFIT and Profit-JanetBury. The data is as follows:

Product Category ID		PROFIT	Profit-JanetBury
PC1	Carbonated Drinks	11,093,941.08	1,211,313.84
PC2	Juices	15,233,656.02	1,605,176.60
PC3	Energy Drinks	1,287,131.69	160,006.38
PC4	Alcohol	11,010,881.86	1,191,261.51
PC5	Others	1,690,971.56	186,323.08

On the right, the "Available Objects" panel is open, showing the configuration for the model. The "Measures" section has "PROFIT" and "Profit-JanetBury" selected (indicated by a red box). The "Dimensions" section has "Product Category ID" selected (also indicated by a red box). The "Columns" section lists "PROFIT" and "Profit-JanetBury".

16. Return to the Analytic Model editor by selecting **Model** in the upper right corner.

17. Save and deploy the model.

Summary

You have now explored more of the various features of the Analytic Model.

You can continue with one of the *optional* exercises:

- [Exercise 20: Identify Top-Performing Sales Managers with Just Ask](#)
- [Exercise 21: Create Row-Level Permissions based on External Hierarchy](#)
- [Exercise 23: Create a Transformation Flow](#)

Exercise 23 - Create a Transformation Flow



Note: This is an OPTIONAL exercise.



Detour: SAP Datasphere - Transformation Flows

Transformation Flows load data from one or more sources and persist the result in a target table. This integration entity can detect delta changes when reading data from a local table which is enabled for delta. Transformation Flows are also useful in scenarios when utilizing Replication Flows for Premium Outbound Integration: Replication Flows can access local tables (delta enabled) which are updated by a transformation flow and transfer the data records in a delta mode.



Detour: SAP Analytics Cloud - Replace Model in Stories

You can replace a model in your SAP Analytics Cloud story with another compatible model, for example an SAP Datasphere Analytic Model with a different Analytic Model. You don't have to recreate your full story if you want to replace the data source (model) with a compatible one. While some features may need to be recreated, the structure and formatting of your dashboard won't be affected.

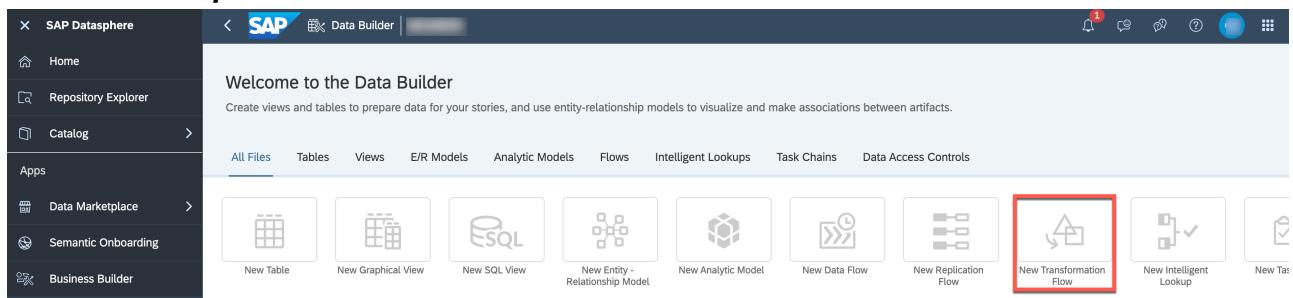
End of Detour

Start of the exercise

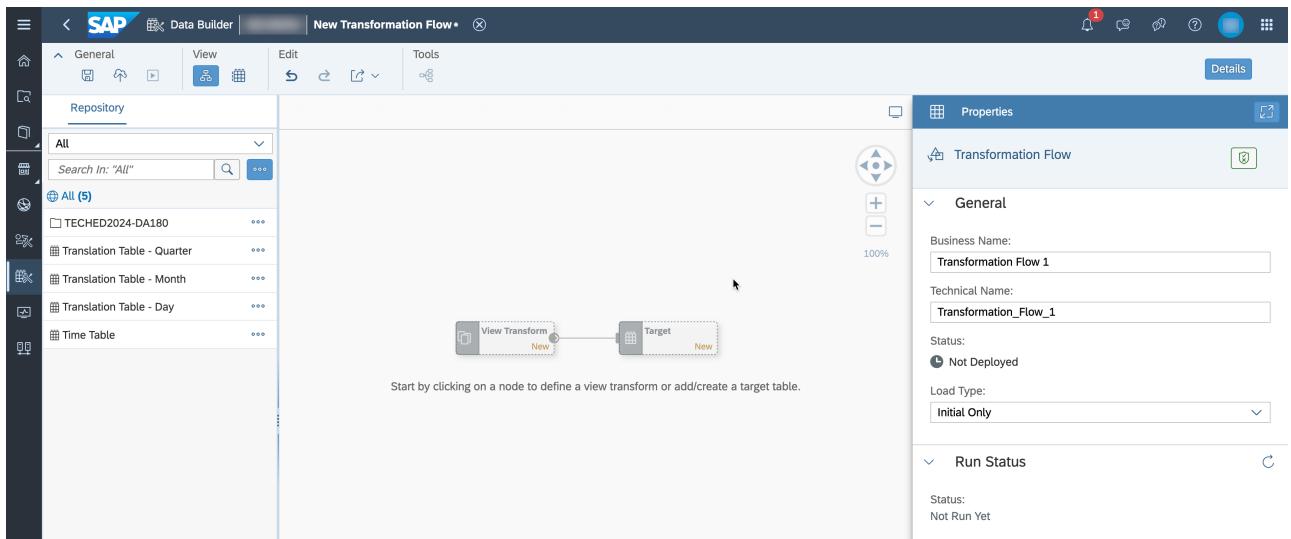
In this exercise, we will access a shared local table which is delta enabled for sales transactions. New sales transactions are generated in a source system and transferred to SAP Datasphere using a Replication Flow in a central admin space of the tenant. Assume that the store ID includes an additional prefix **US** which we need to remove so that we can map the store ID to the dimensions available in the view dimensions.

We will create a new Analytic Model which accesses the regularly updated data and use the replace model feature in SAC to have a second story accessing the new Analytic Model. This functionality is beneficial for development activities when you need to transition from one data model to another. Initially, we utilized CSV files, but now we are looking to map the data to a new data source for the sales transactions.

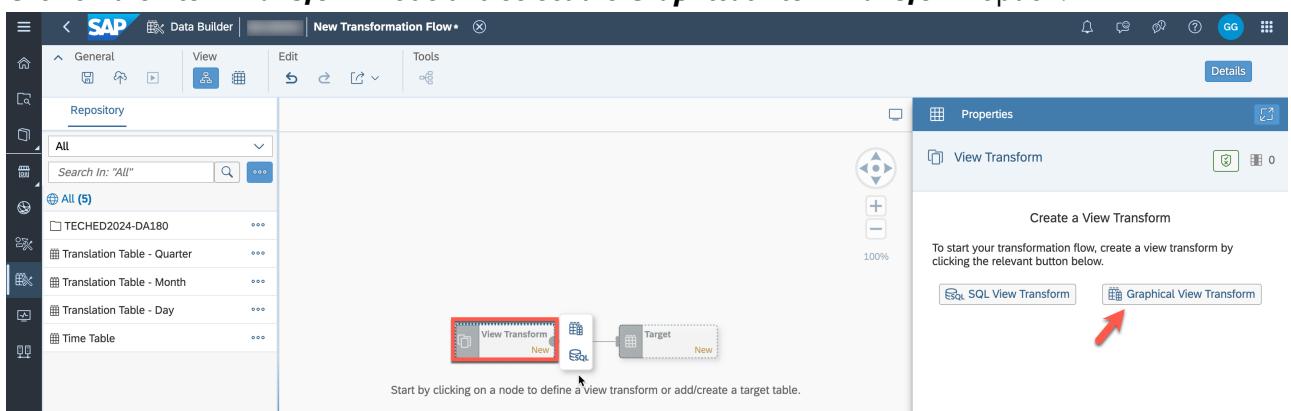
1. Log On to your SAP Datasphere tenant.
2. Select the menu option **Data Builder** on the left-hand side.
3. Click **New Transformation Flow**.



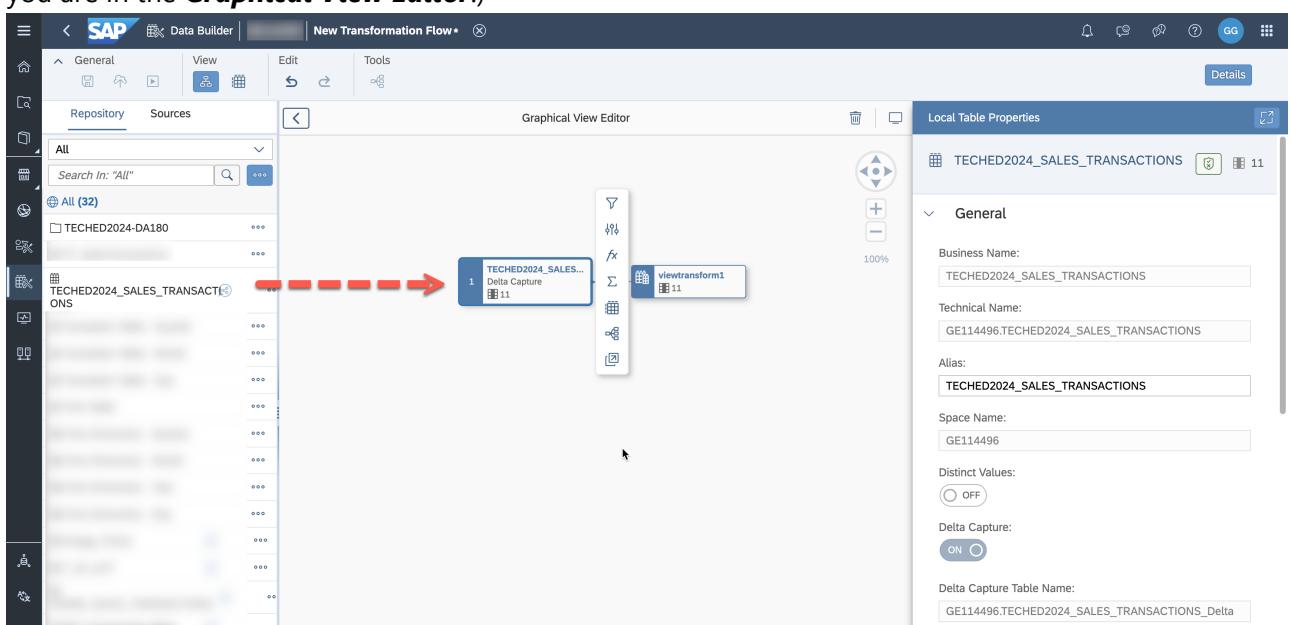
4. You see the Transformation Flow editor with options to define view transformation and create a new table.



5. Click on the **View Transform** node and select the **Graphical View Transform** option.



6. Drag and drop the table **TECHED2024_SALES_TRANSACTIONS** into the editor. The details on the right-hand side display that this shared table is capturing delta. (If you don't see this table, ensure that you are in the **Graphical View Editor**.)



7. If you scroll down, ensure that **Load from Table** is set to **Delta Capture**. As the delta capture setting is enabled for the source table, the columns **Change Date** and **Change Type** are automatically mapped

to these columns in the target table. Mapping these columns (or a calculated column that contains the content of these columns) to any other target column is not permitted.

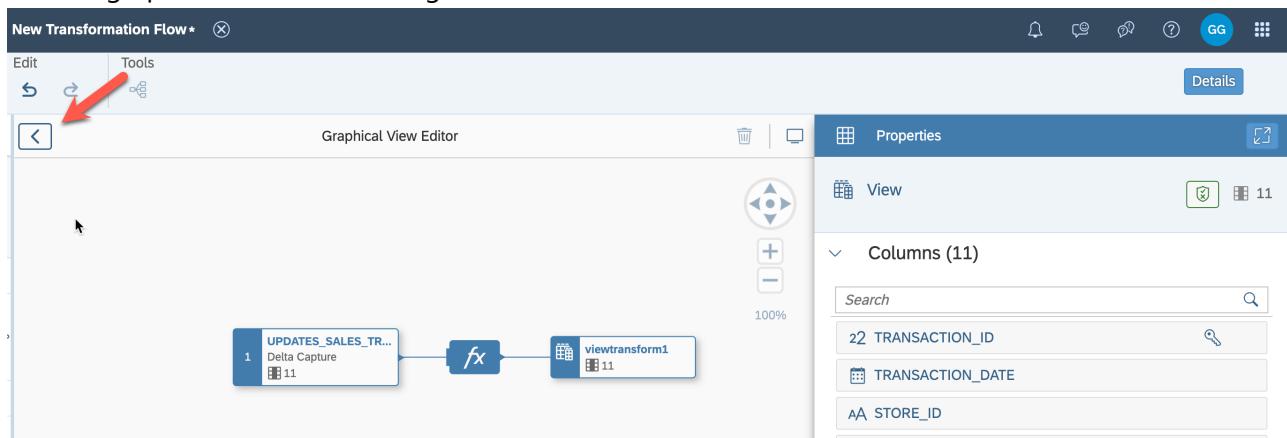
The screenshot shows the 'Local Table Properties' dialog. At the top, there is a 'Load from Table:' section with two radio button options: 'Delta Capture' (selected) and 'All Active Records'. Below this is a 'Columns' section with a dropdown menu. The listed columns are:

- 22 TRANSACTION_ID
- TRANSACTION_DATE
- AA STORE_ID
- AA PRODUCT_ID
- AA SALES_MANAGER_ID
- 1²³ PROFIT
- 1²³ DISCOUNT
- 1²³ REVENUE
- 1²³ COST
- AA Change Type
- Change Date

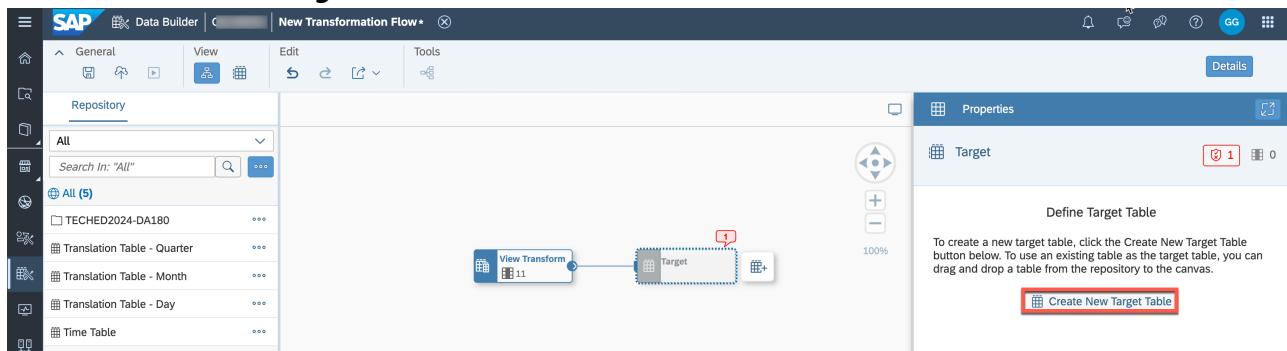
8. Add a new node for **Calculated Columns**. Choose the existing column **STORE_ID** and enter the expression **SUBSTR_AFTER(STORE_ID,'US')** as we will exclude the prefix of the STORE_ID so that it can be mapped to the dimension IDs. Adjust the length to 6 characters.

The screenshot shows the 'Graphical View Editor' interface. On the left, there is a flowchart with nodes: 'SHARE_SALES_TRAN...' (with a red arrow pointing to its output), 'fx' (a calculated column node), and 'viewtransform1'. The 'fx' node has a red box around its 'Expression' field. On the right, the 'Element Properties' panel is open for the 'STORE_ID' column. The 'Expression' field contains the value `RIGHT(STORE_ID,6)`, which is also highlighted with a red box. Other properties shown include Business Name: 'STORE_ID', Technical Name: 'STORE_ID', Data Type: 'AA String', and Length: '6'.

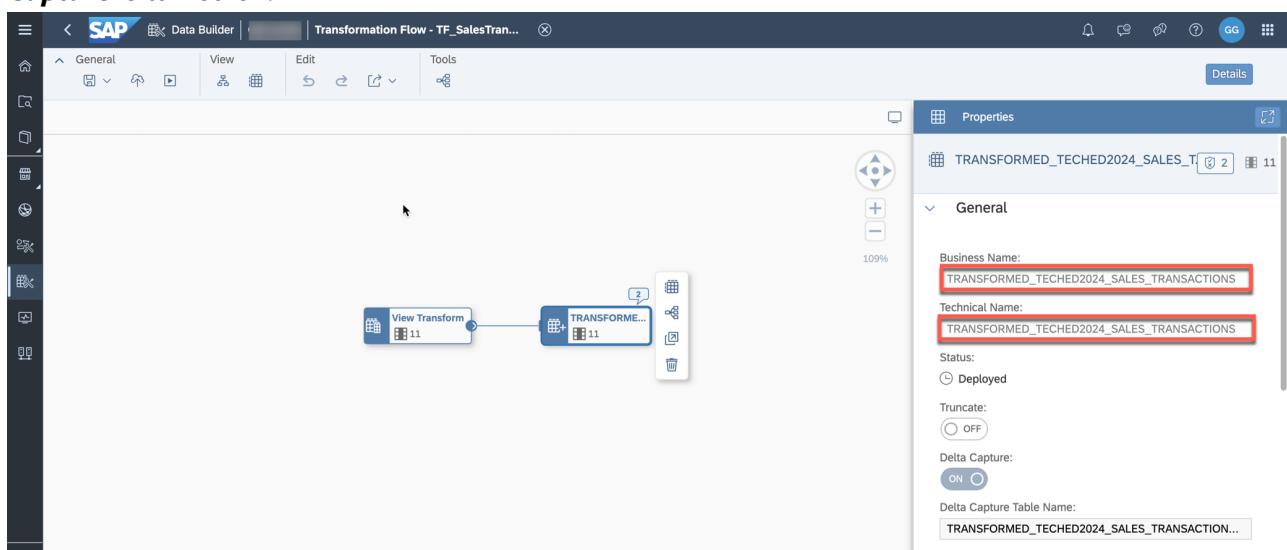
9. Exit the graphical view editor and go back to the transformation flow overview.



10. Click on **Create New Target Table**.

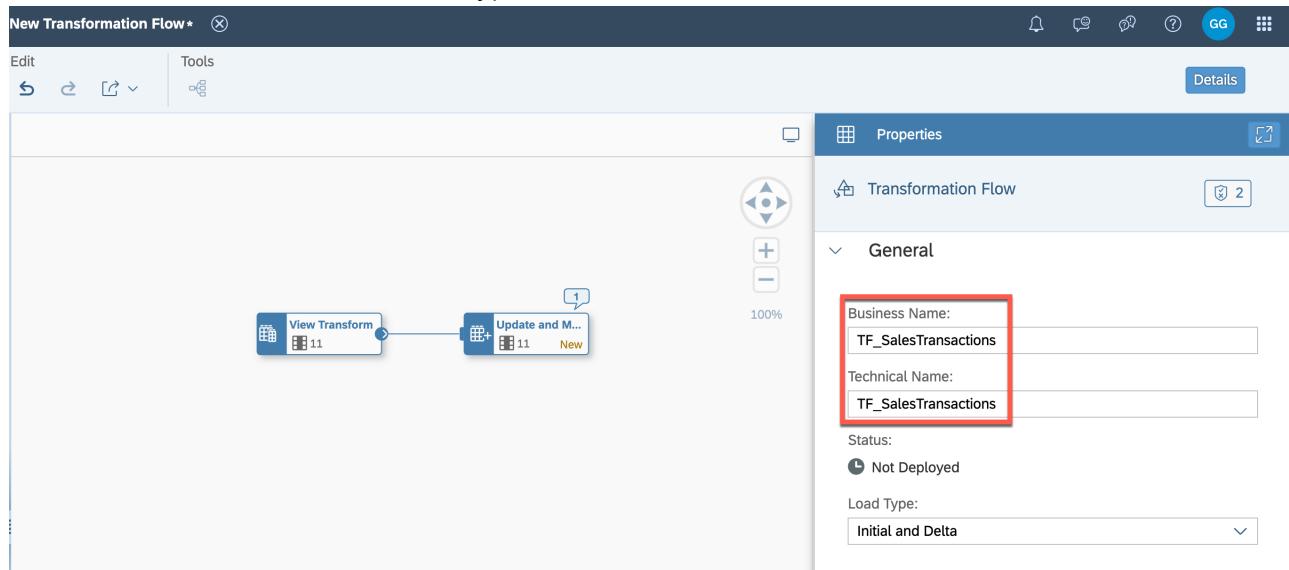


11. Set the business name to **TRANSFORMED_TECHED2024_SALES_TRANSACTIONS**. Verify that **Delta Capture** is turned on.

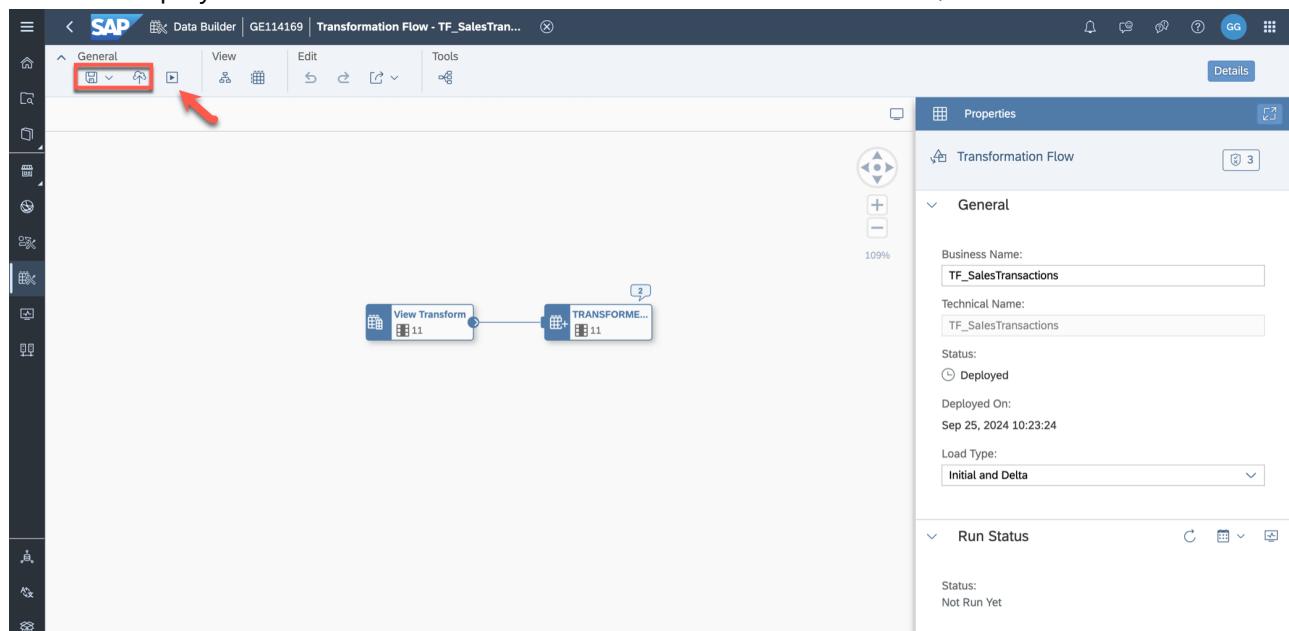


12. Deselect the target table by clicking somewhere else in the editor field. The properties for the Transformation Flow will appear on the right-hand side. Set the business name to

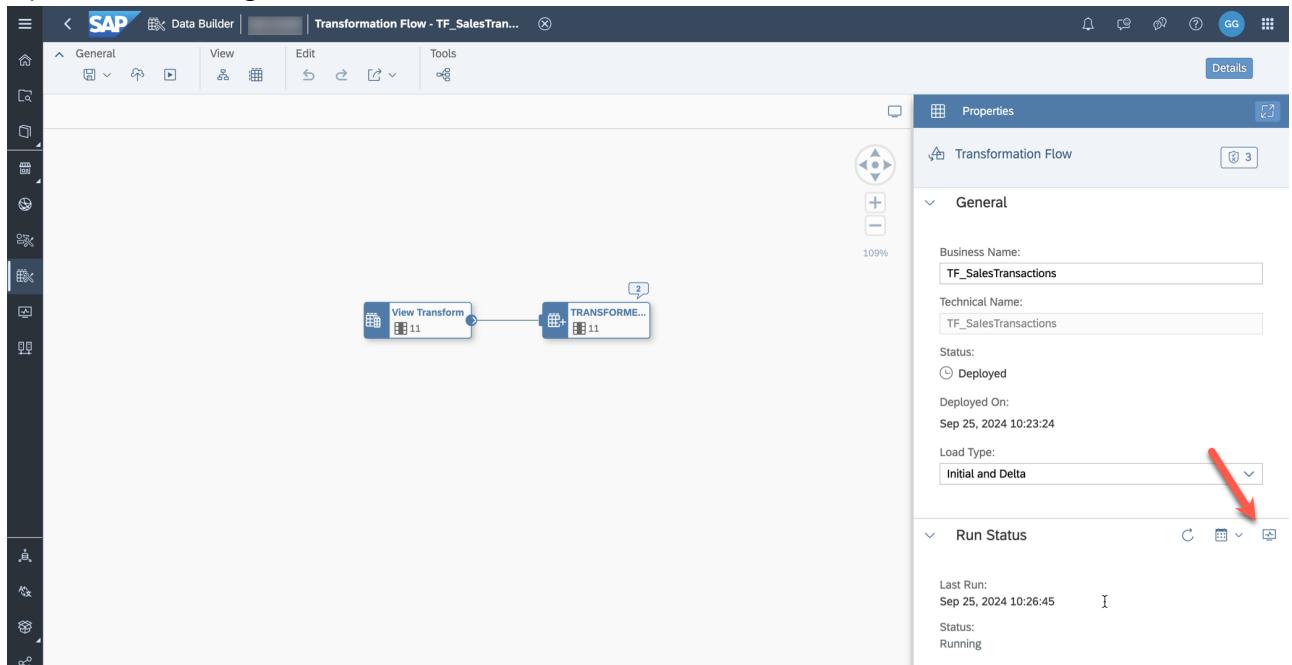
TF_SalesTransactions and the Load Type to **Initial and Delta**.



13. Save and deploy the Transformation Flow in the folder TECCHED2024-DA180, afterward click on **Run**.



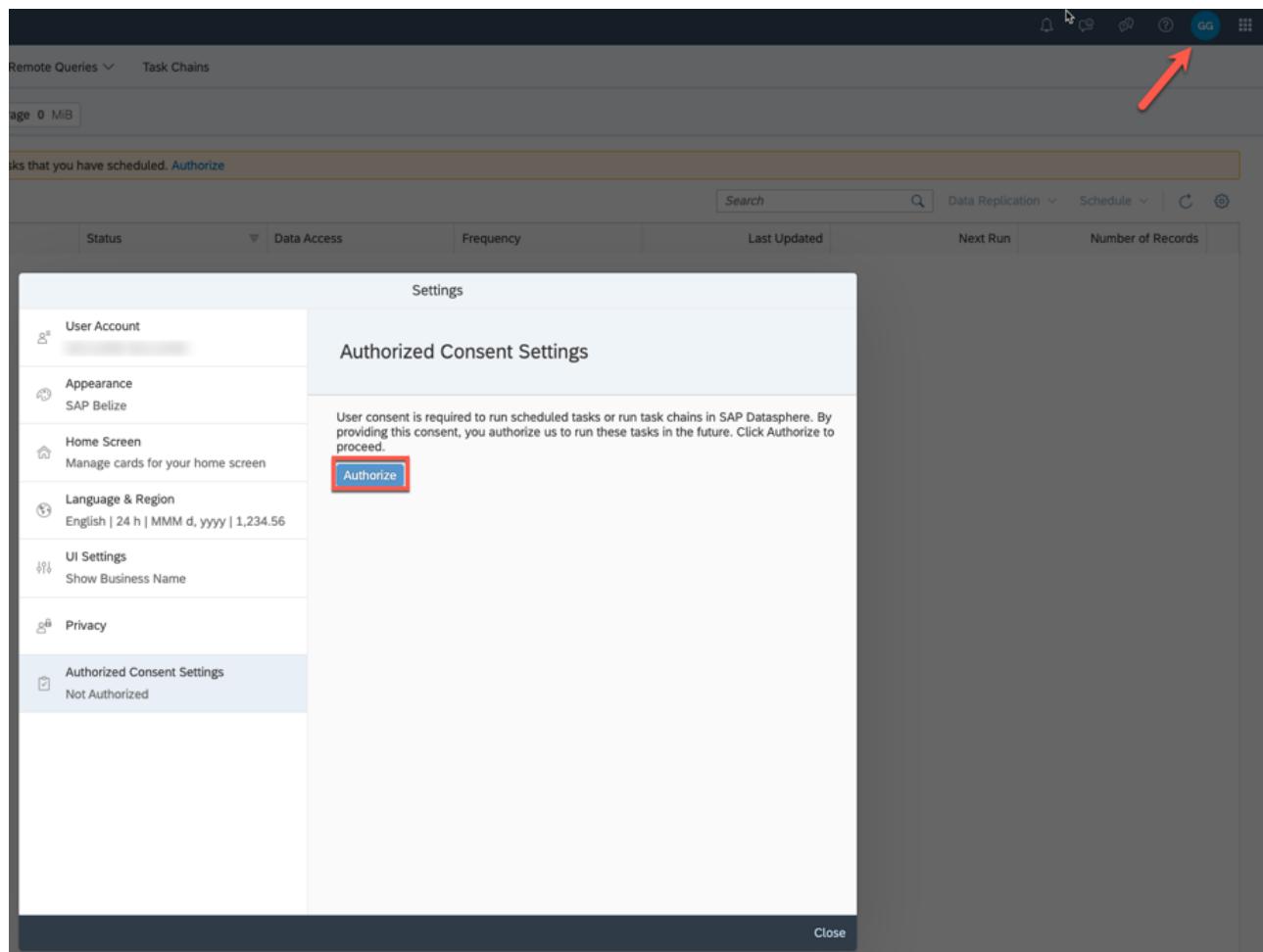
14. Open the Data Integration Monitor to have a look at the Transformation Flow run.



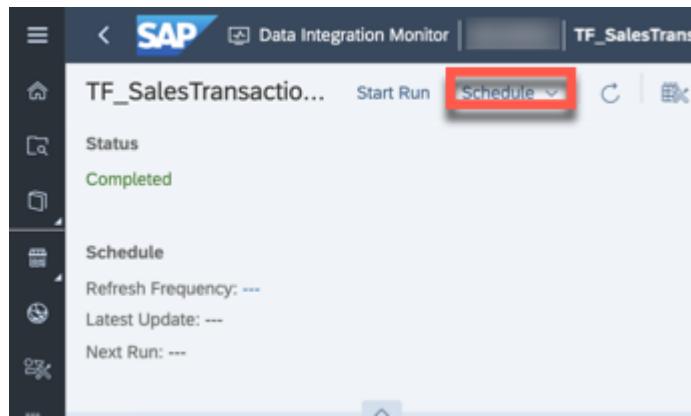
15. The Data Integration Monitor details display of the Transformation Flow run. Have a look at the messages and the section **Metrics** to check the number of processed records.

Timestamp	Category	Message
Aug 20, 2024 11:59:32	Information	Task 133384 started. View Details
Aug 20, 2024 11:59:34	Information	Starting transformation flow run.
Aug 20, 2024 11:59:36	Information	Preparing to transform data. View Details
Aug 20, 2024 11:59:36	Information	Reading data from the "View Transform" operation and writing it to the target table. View Details
Aug 20, 2024 11:59:37	Information	200 records written to the target table.
Aug 20, 2024 11:59:37	Information	Finalizing transformation flow run.
Aug 20, 2024 11:59:42	Information	Transformation flow run completed successfully. View Details
Aug 20, 2024 11:59:42	Information	Task 133384 has finished at 2024-08-20T09:59:37.290Z with status COMPLETED View Details

16. The Transformation Flow run should run scheduled. User consent is required to run scheduled tasks or task chains. Authorize SAP Datasphere to run these tasks.



17. Schedule the Transformation Flow to run once a day. It will retrieve the newest and changed records and add them to the local table. Click **Create Schedule**.



18. Define the scheduling as **Simple Schedule** which is repeated every 10 minutes. Choose a time and select **Create**.

Edit Schedule for "TF_SalesTransactions"

Schedules are created in Coordinated Universal Time (UTC). The current UTC time is Sep 23, 2024 8:37:48.

Frequency

Enter As: Simple Schedule

Repeat: Minutes

Every: 10 Minute(s)

Reset Every Hour

Time Range

Start Date: * Sep 22, 2024

End Date: No End Date

Ownership

Owner: GE114169

Show Runs In: UTC

At every 10 minute.

Pause Schedule: NO

Next Runs:

- Sep 23, 2024 8:40:00
- Sep 23, 2024 8:50:00
- Sep 23, 2024 9:00:00
- Sep 23, 2024 9:10:00
- Sep 23, 2024 9:20:00

Save Cancel

19. You can go back to this monitor after new runs to see that new records are written to the target table.

TF_SalesTransactions

Status: Completed

Schedule: Refresh Frequency: Scheduled

Latest Update: Sep 22, 2024 23:54:39

Next Run: Sep 23, 2024 10:50:00

Logs

Runs

Start	Activity	Status
Sep 23, 2024 10:40:33	Run	Completed
Sep 23, 2024 10:30:33	Run	Completed
Sep 23, 2024 10:20:33	Run	Completed
Sep 23, 2024 10:10:33	Run	Completed
Sep 23, 2024 10:00:34	Run	Completed
Sep 23, 2024 9:50:33	Run	Completed
Sep 23, 2024 9:40:33	Run	Completed

Run Details

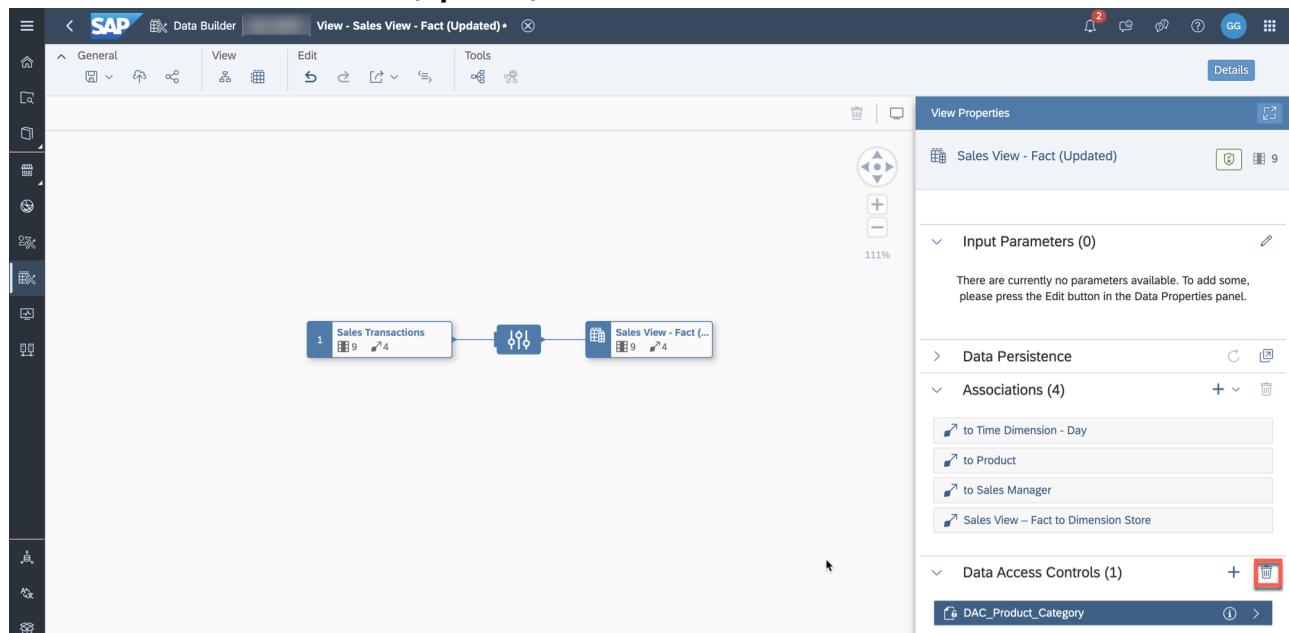
Activity:	Status:	Duration:	Start Date:	End Date:
Run	Completed	00:00:01	Sep 23, 2024 10:40:33	Sep 23, 2024 10:40:35
Run By:	Run Type:			
GE114169	Scheduled			

Message

Timestamp	Category	Message
Sep 23, 2024 10:40:33	Information	Task 141784 started. View Details
Sep 23, 2024 10:40:34	Information	Starting transformation flow run.
Sep 23, 2024 10:40:35	Information	Preparing to transform data. View Details
Sep 23, 2024 10:40:35	Information	Reading data from the "View Transform" operation and writing it to the target table. View Details
Sep 23, 2024 10:40:35	Information	10 records written to the target table.
Sep 23, 2024 10:40:35	Information	Finalizing transformation flow run.
Sep 23, 2024 10:40:46	Information	Transformation flow run completed successfully. View Details
Sep 23, 2024 10:40:46	Information	Task 141784 has finished at 2024-09-23T08:40:35.273Z with status COMPLETED

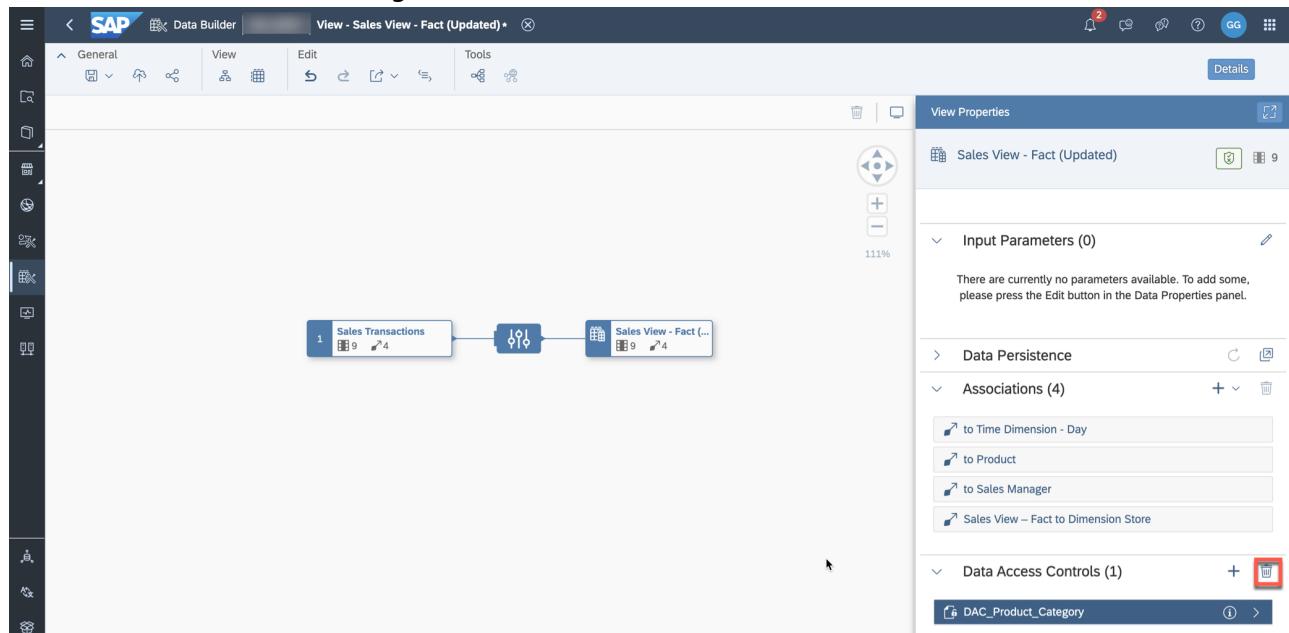
20. The Analytic Model used until now (Sales_Analytic_Model) is based on data of local tables. We will copy the fact model and replace the source so that the sales transaction data is coming from the table updated via the transformation flow instead of data imported once via CSV. Copy the view **Sales View** -

Fact. Name it **Sales View - Fact (Updated)**.

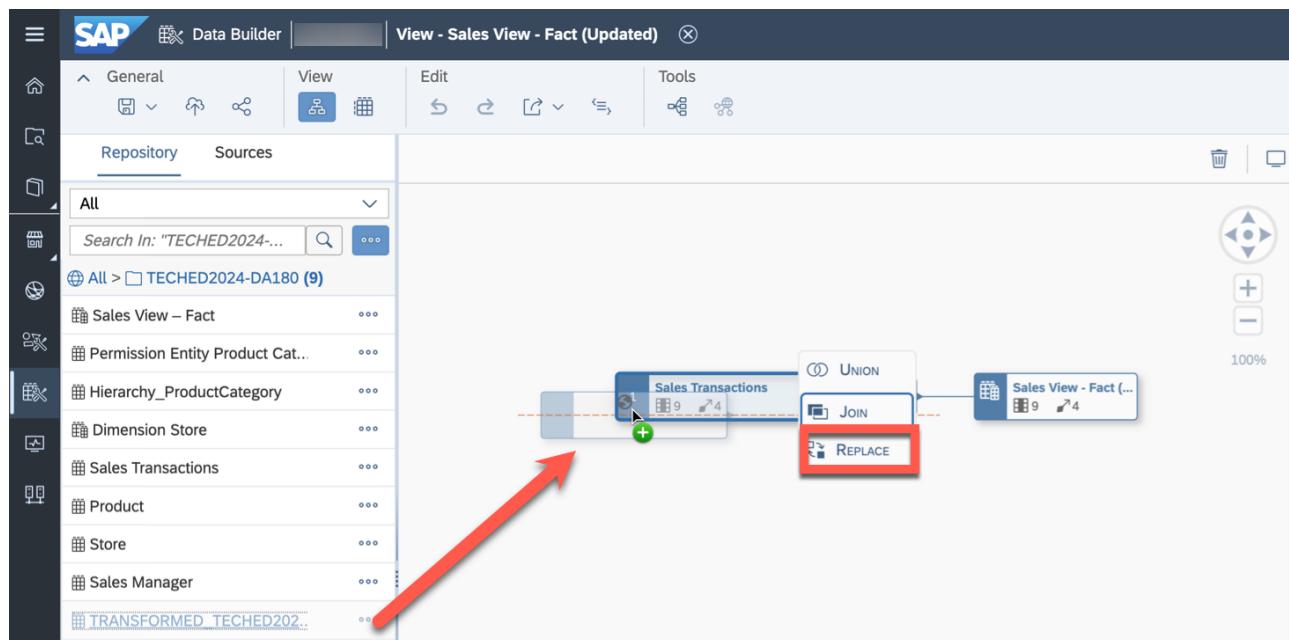


Tip: We create a copy of the fact view (and Analytic Model & SAC story) to preserve the previously created models based on the CSV files. You will copy the existing SAC story and replace the model so that you don't need to recreate a full story. Another option would also be switching the source table of the currently used fact view so that the updated records are displayed in the already existing SAC story.

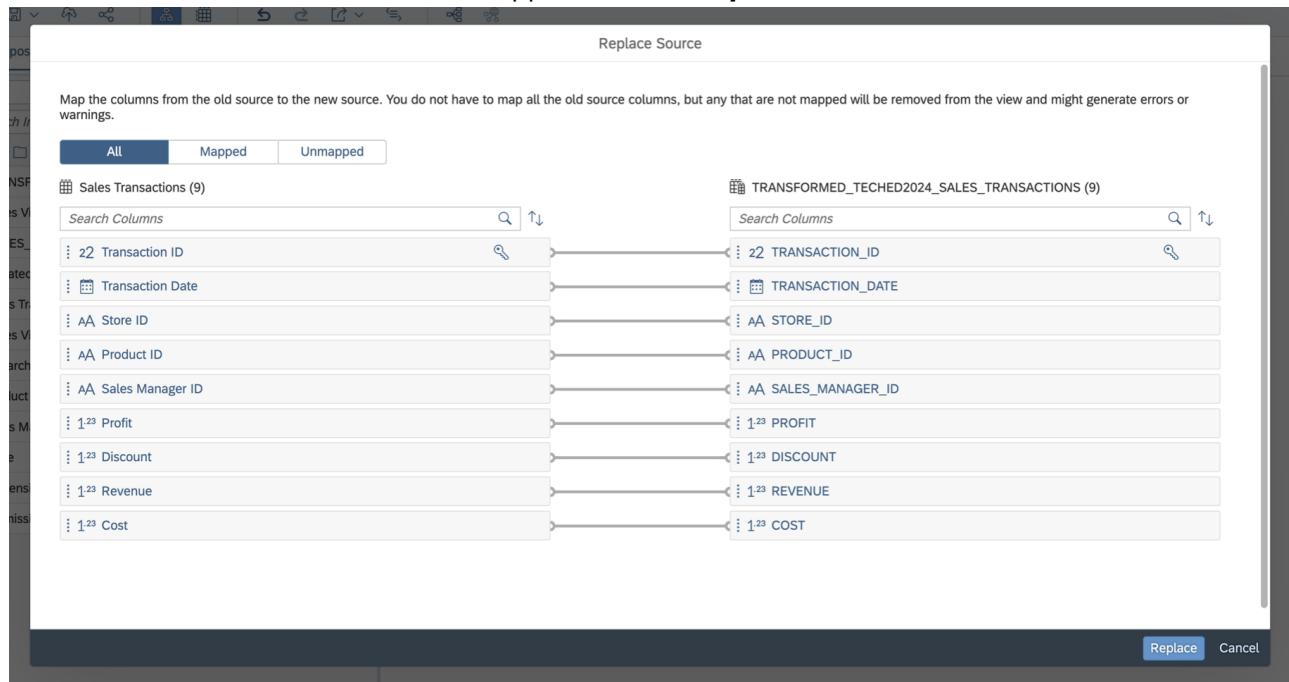
21. Open the copied view **Sales View - Fact (Updated)**. If you previously did the exercise about data access controls, the view does not look as displayed on the screenshot. Remove the join with **ProductDim_DAC** and the assigned DAC.



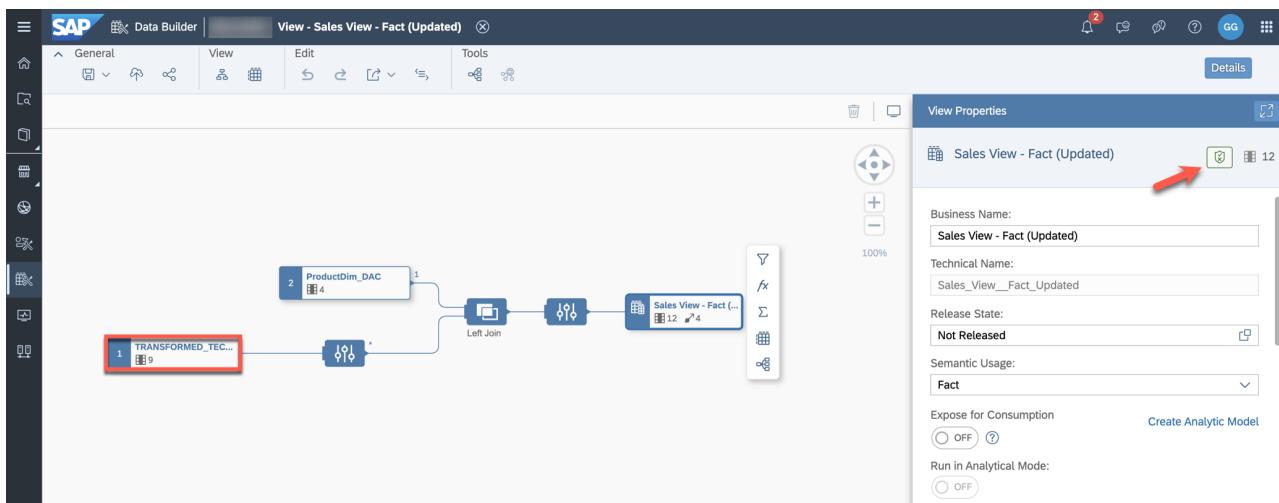
22. Drag and drop the table **TECHED2024_SALES_TRANSACTIONS** into the canvas and replace the original source table.



23. Map the columns from the old source to the new source by dragging them from the left to the right side. Ensure that all nine columns are mapped and click **Replace**.

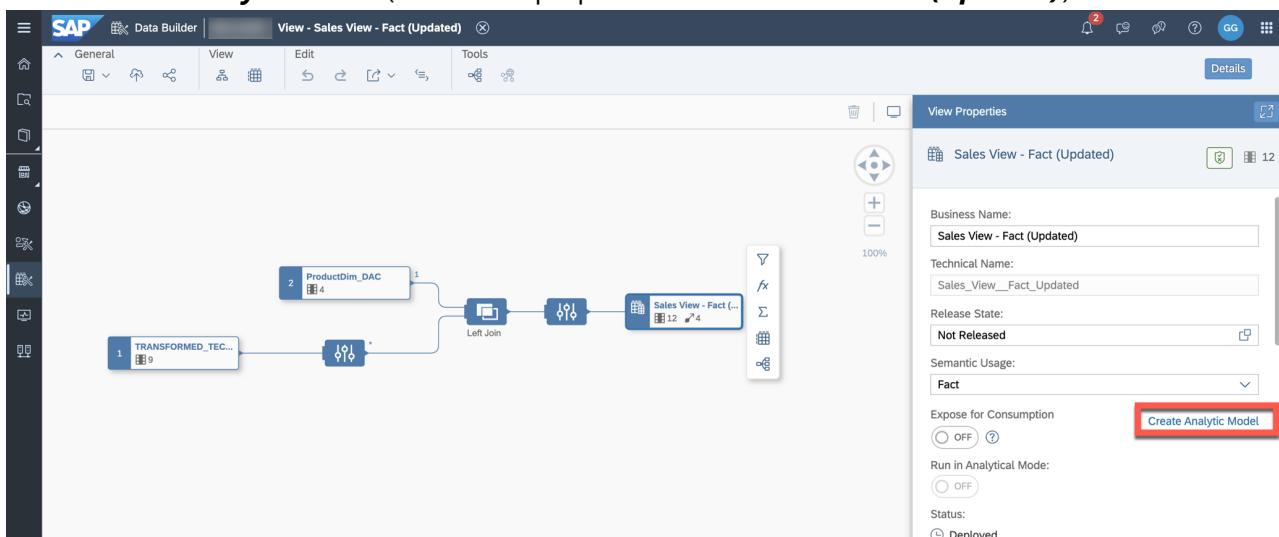


24. Now the source table is replaced with the local table which is enabled for delta capture. Ensure that the view validation of **Sales View - Fact (Updated)** displays a green status.

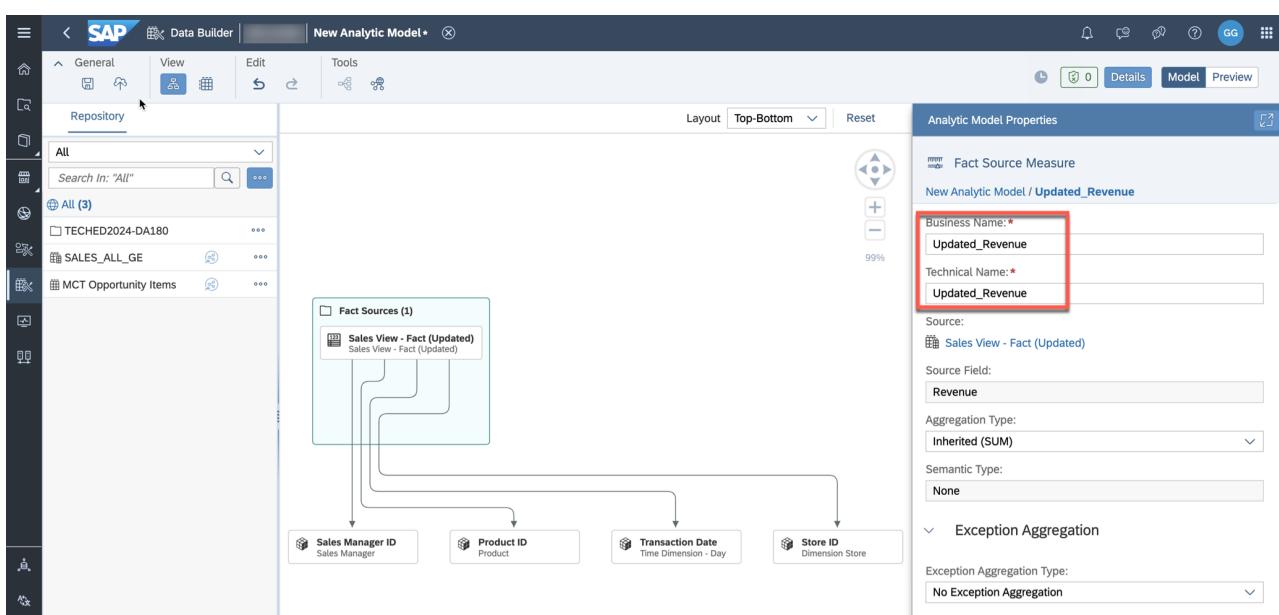


25. Save and deploy **Sales View - Fact (Updated)**.

26. Select **Create Analytic Model** (in the view properties of **Sales View - Fact (Updated)**).



27. When creating an updated data model, it is likely that measure or attribute names change. To simulate this, select the measure **Revenue** and adjust the business and technical name to **Updated_Revenue**.



28. Save the Analytic Model as **Sales - Analytic Model (Updated)** in the folder **TECHED2024-DA180**. You need to confirm the warning "Renaming measure technical name might affect existing stories" once before saving and deploying the model. The Analytic Model is now accessing delta enabled local table which is receiving updated records. We will reuse the previously created SAC Story and map it to the new updated Analytic Model so that the report displays the incoming sales transactions.

29. Use the product switch button in the upper right corner to switch to SAC.

The screenshot shows the SAP Data Builder interface. On the left, there's a navigation bar with 'General', 'View', 'Edit', and 'Tools'. Below it is a 'Repository' section with a search bar and a list of items under 'All' and 'TECHED2024-DA180'. In the center, there's a diagram showing 'Fact Sources (1)' (Sales View - Fact (Updated)), 'Dimensions' (Sales Manager ID, Product ID, Transaction Date, Store ID), and 'Measures'. On the right, there's a configuration panel for the 'Analytic Model' with fields for 'Name' (Sales - Analytic Model (Updated)), 'Technical Name' (Sales_Analytic_Model_Updated), 'Status' (Deployed), and 'Deployed On' (Sep 24, 2024 14:06:39). At the top right, there are two buttons: 'SAP Analytics Cloud' and 'SAP Datasphere'. A red arrow points to the 'SAP Analytics Cloud' button.

30. Switch to the **Files** application and copy the previously created story **Revenue Analysis - Products** to keep the previously created story as a backup. Copy it to the same folder and name it **Revenue Analysis - Products (Updated)**.

The screenshot shows the SAP Files application. On the left, there's a sidebar with 'My Files', 'Views' (which has a checked checkbox and a red arrow pointing to it), 'Owned by Me', 'Shared with Me', and 'Favorites'. The main area shows a list of files with columns for 'Name', 'Description', 'Type', 'Owner', 'Created On', 'Changed By', and 'Changed On'. A red arrow points to the 'Copy' icon in the toolbar at the top.

31. Open the story **Revenue Analysis - Products (Updated)** and switch to the **Edit** mode.

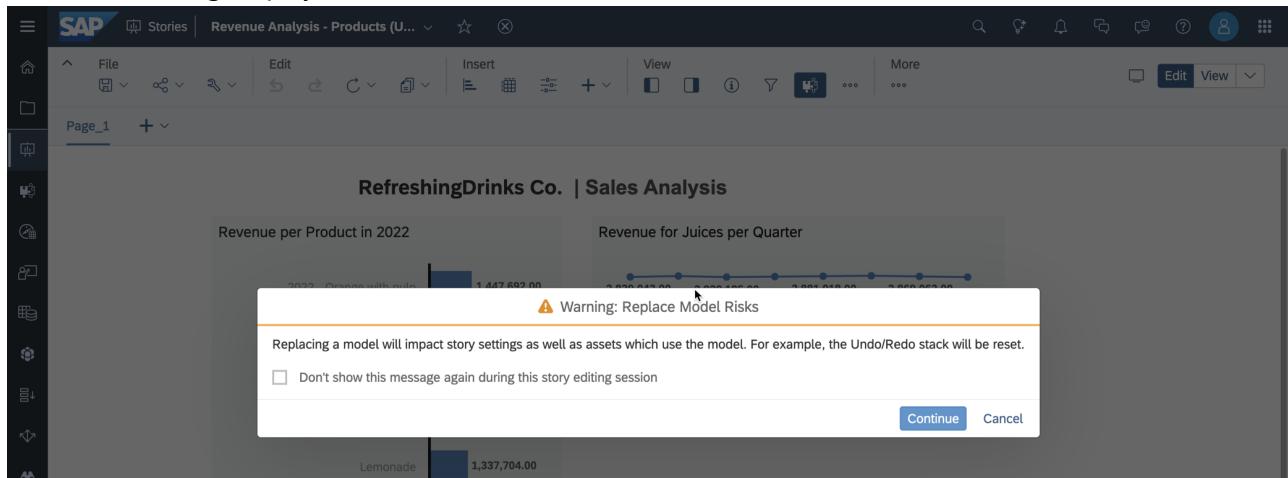
32. To replace the model, access ... -> **Add New Data** -> **Sales_Analytic_Model**

The screenshot shows the SAP Stories application. On the left, there's a sidebar with 'File', 'Insert', 'View', and 'More'. The main area shows a story named 'Revenue Analysis - Products (U...)'. A red arrow points to the 'More' button in the toolbar. A context menu is open with options like 'Add New Data...', 'Replace...', 'Edit Prompts', 'Link Dimensions', and 'Chart Scaling'. The 'Replace...' option is highlighted with a red box.

33. Select **Replace**.

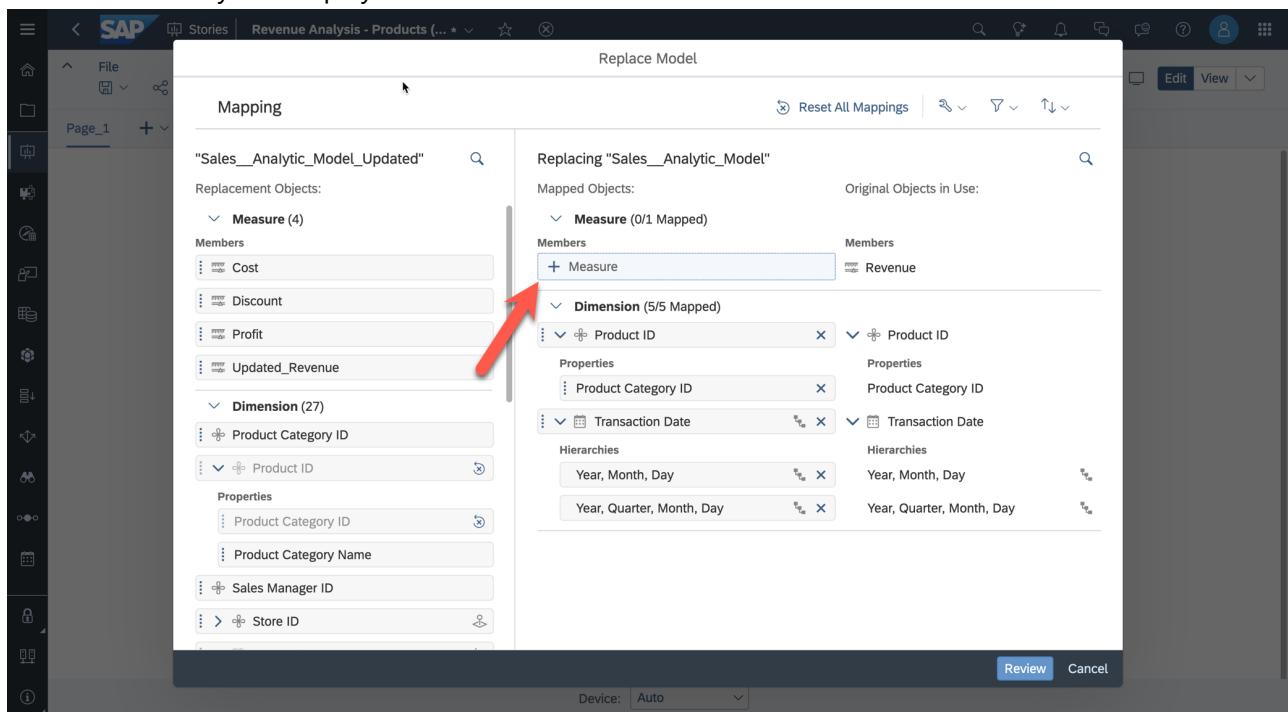
The screenshot shows the SAP Stories application with the same story and toolbar as the previous screenshot. A red box highlights the 'Replace...' option in the context menu.

34. Read the warning displayed and click **Continue**.

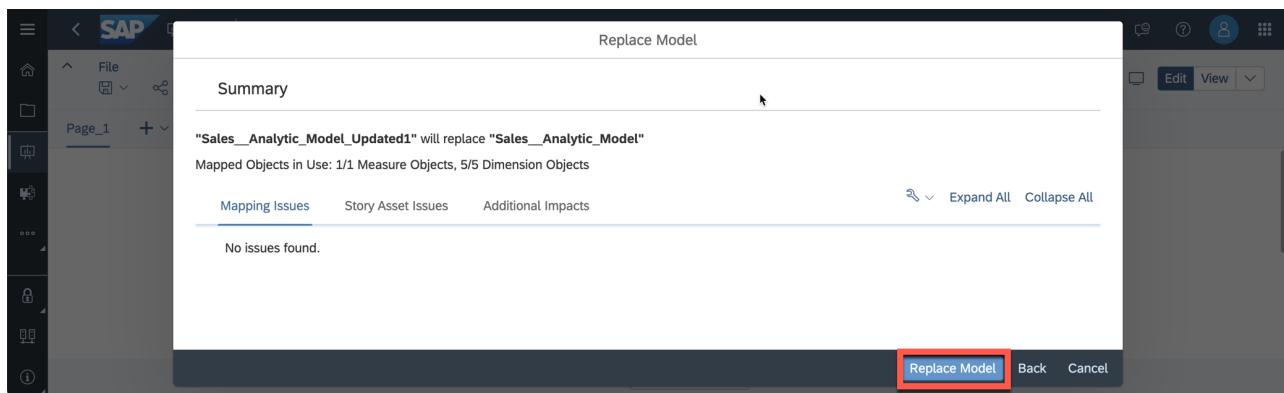


35. Select **select other model**. You can replace a model in your SAC story with another compatible model, an Analytic Model can be replaced with a different Analytic Model. Choose the connection **DATASPHERE** and click on the folder with your user's ID. Select the folder **TECHED2024-DA180** and click on the Analytic Model **Sales - Analytic Model (Updated)**.

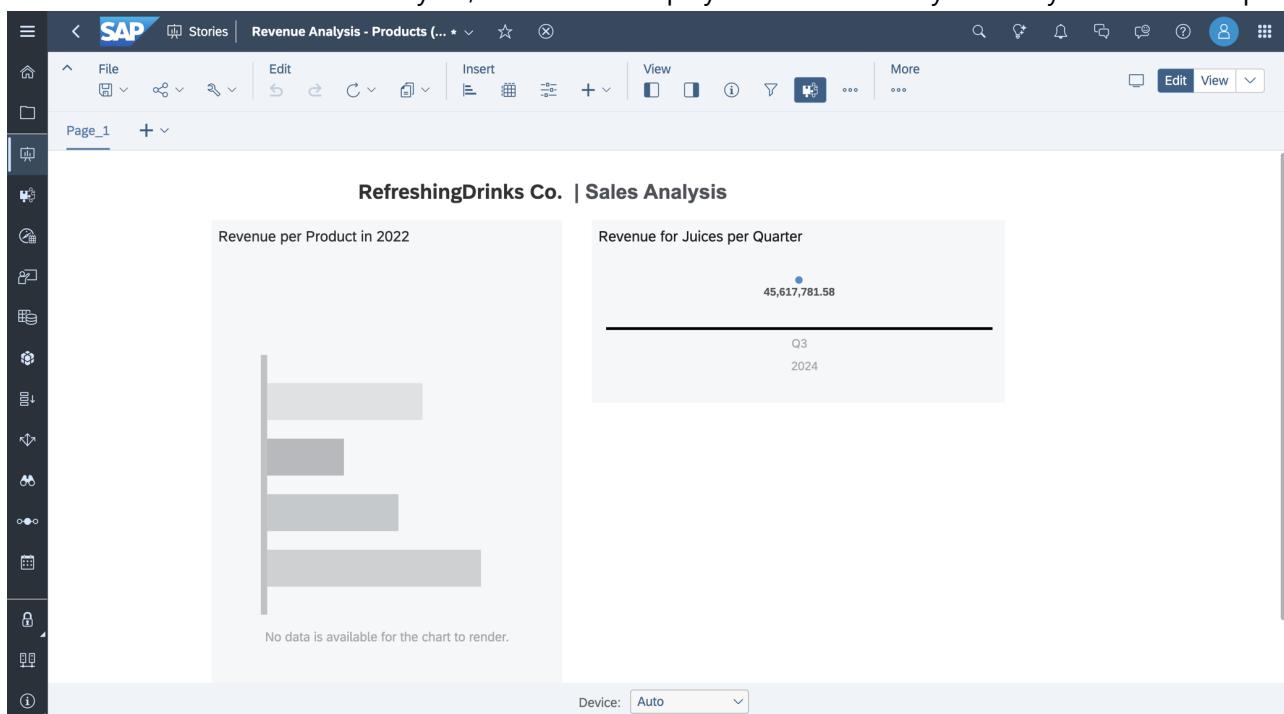
36. The replace model dialog displays the objects from the replacement model and the objects that need to be mapped (only objects used in the story are mapped). Some objects will automatically be mapped if they are similar to the original objects or if they are mandatory objects. Then they will be disabled (greyed out) in this panel. Map the measure **Updated_Revenue** to **Revenue**. Check that the mappings are done correctly like displayed in the screenshot.



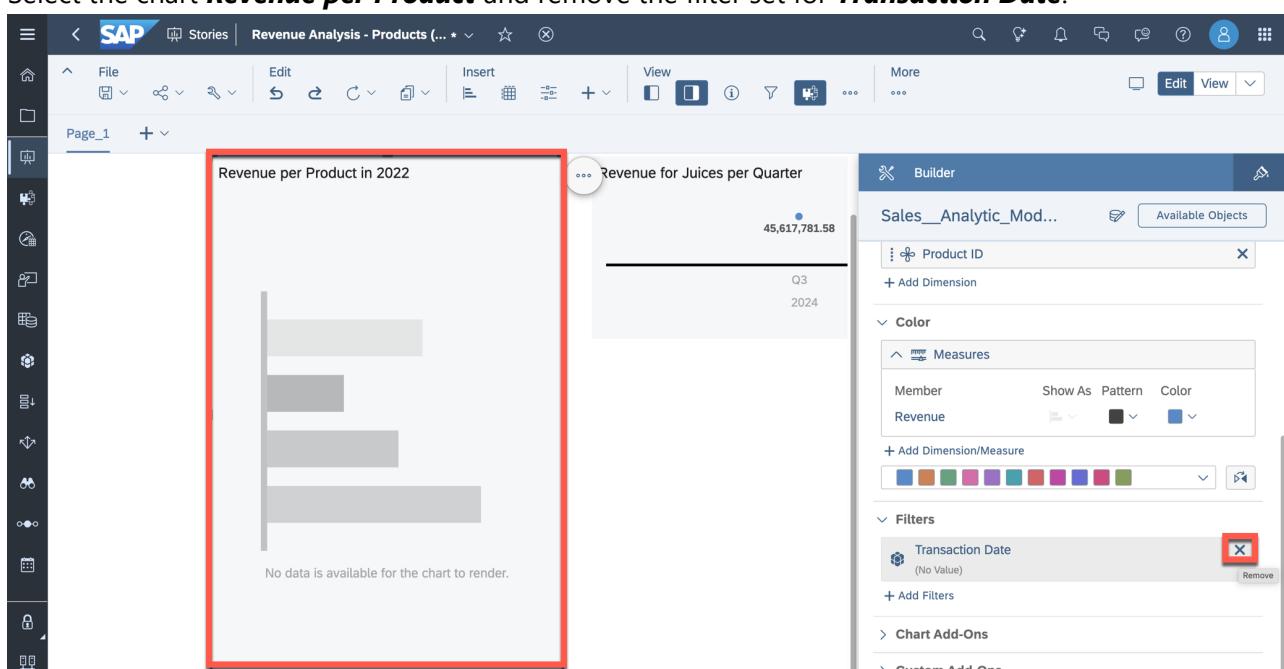
37. When you have finished mapping objects, select **Review**. Verify that no issues have been found and click **Replace Model**.



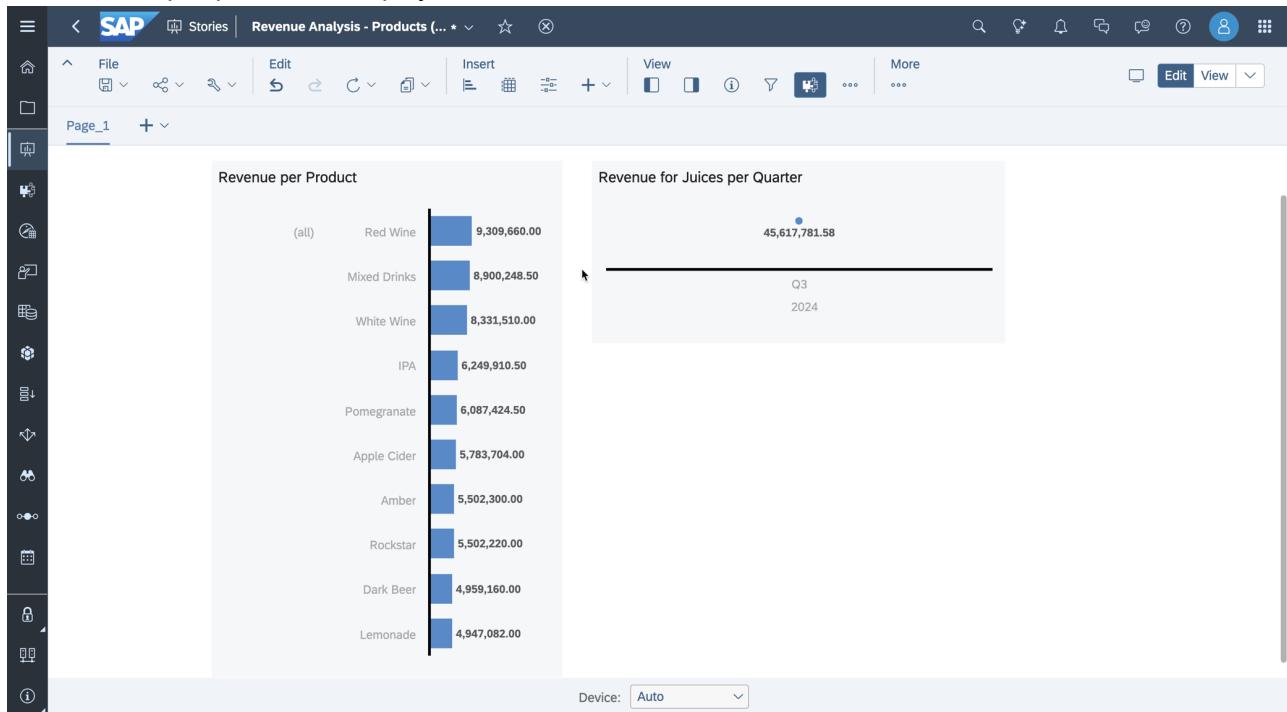
38. The story now accesses data from the new source. The filter for revenue per product is set to 2022. As there is no data available for this year, no data is displayed. We will modify the story in the next steps.



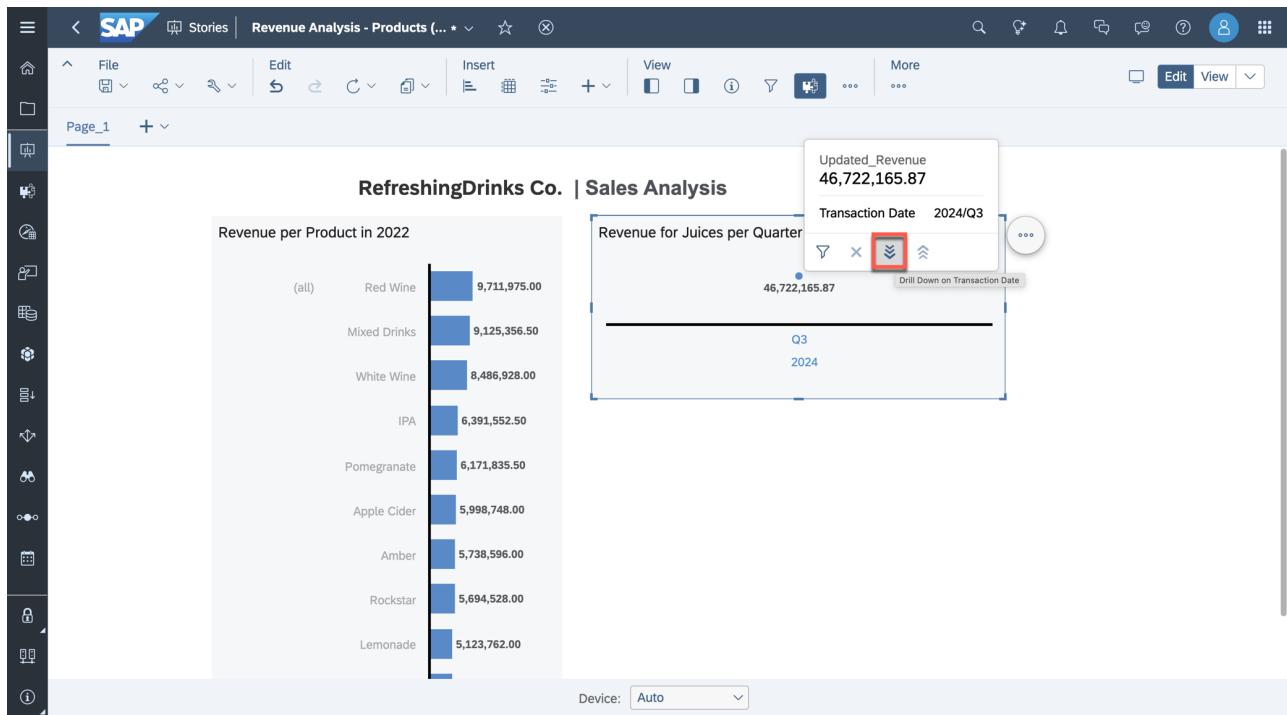
39. Select the chart **Revenue per Product** and remove the filter set for **Transaction Date**.



40. The revenue per product is displayed.

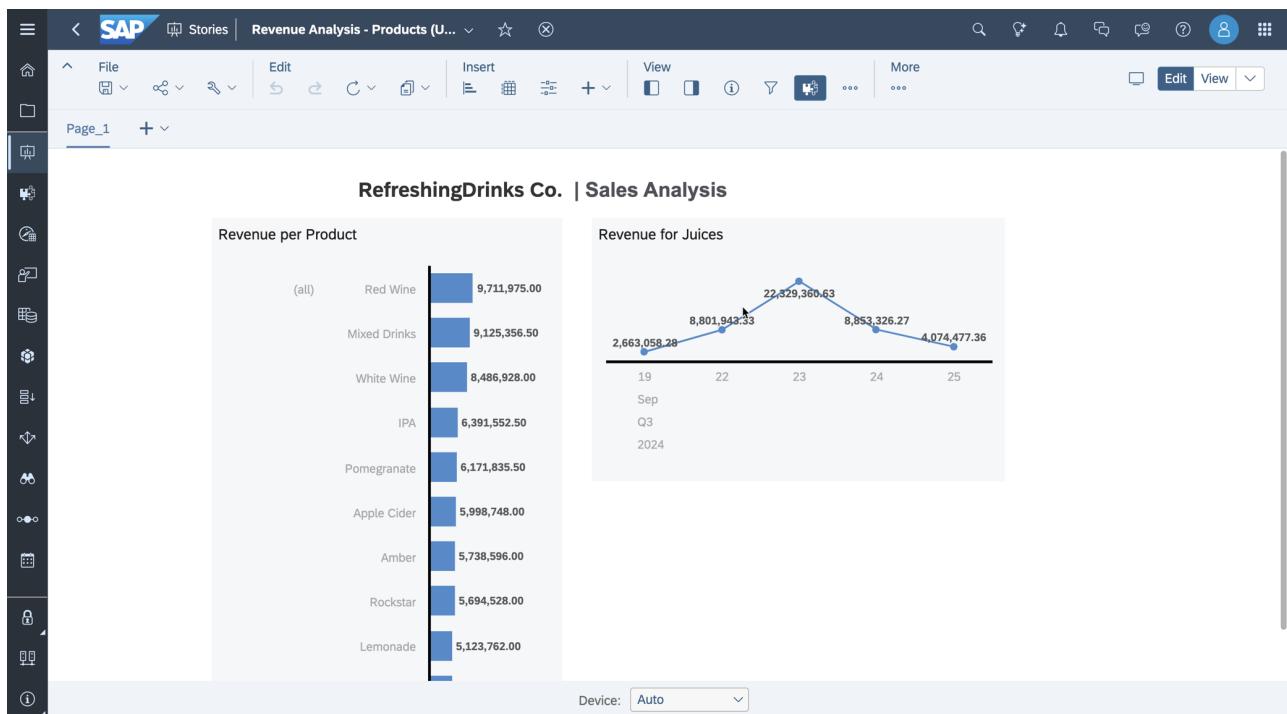


41. Now we focus on the chart on the right-hand side. We would like to receive a daily summary of recent sales of juices from the past few days. Open the wizard for drill-down when clicking onto the data point. Drill down on the transaction date until it is based on days (click twice on the **Drill Down** button).



42. Save the story. You are asked if you want to remove the unused model, select **Remove data source**.

43. Your final report dynamically displays the most recent beverage sales, allowing you to analyze the changes in revenue from sold juices over the past few days.



Summary

You have now created and deployed your first transformation flow to transform and access changing records in a scheduled approach. You replaced the data source while keeping the story structure and format intact.

You can continue with one of the optional exercises:

- [Exercise 20: Identify Top-Performing Sales Managers with Just Ask](#)
- [Exercise 21: Create Row-Level Permissions based on External Hierarchy](#)
- [Exercise 22: Explore the Analytic Model](#)